

# **The Complete x86**

**The Definitive Guide to  
386, 486, and Pentium-Class  
Microprocessors**

**Volume I**

**Edited by John Wharton**

**MICRODESIGN  
RESOURCES**

MicroDesign Resources  
874 Gravenstein Hwy. So.  
Sebastopol, CA 95472  
707.824.4004, fax: 707.823.0504  
email: info@mdr.ziff.com

Copyright © 1994 MicroDesign Resources  
All rights reserved.  
Printed in the United States of America

ISBN 1-885330-02-2

No part of this report may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without prior written permission.

MicroDesign Resources is a trademark of Ziff-Davis Publishing L.P.

This report contains and analyzes information from publicly available sources and from industry contacts. Although we consider these sources reliable, we cannot guarantee the accuracy, whether past, present or future, of the information or analyses contained herein. Readers assume full responsibility for any use made of this report or its contents, in whole or in part.

Trademark names are used throughout this report. These trademarks are the property of their respective owners.

Please inform us of any errors of fact or omission so they may be corrected in future editions of the report.



Printed on recycled paper.

# Table of Contents

<b>Foreword</b> .....	xxxiii
<b>Executive Summary</b> .....	xxxv
<i>Why this Report is Needed</i> .....	xxxv
<b>Preface</b> .....	xxxvii
<i>How this Report is Organized</i> .....	xxxvii
<i>Terminology</i> .....	xxxviii
<i>Acknowledgments</i> .....	xxxix
<i>Feedback and Updates</i> .....	xl
<b>About the Authors</b> .....	xli
<b>About the Publisher</b> .....	xliii

## Volume I

<b>Part I: Preliminaries</b> .....	1
<b>1 The x86 Business Climate</b> .....	3
1.1 The Explosion of Design Alternatives .....	6
1.2 Forestalling the Blood Bath .....	8
1.3 Dimensions of Differentiation .....	8
1.4 Commentary .....	13
1.5 For More Information.....	14
<i>Microprocessor Report Articles</i> .....	14
<i>Other Technical References</i> .....	15
<i>Other Periodicals</i> .....	15
<b>2 x86 Family Heritage</b> .....	17
2.1 Jurassic Parts .....	17
<i>The 8086</i> .....	19
<i>The 8088</i> .....	22
<i>The 80186</i> .....	23
<i>The 80286</i> .....	23
2.2 The 386 Family .....	24
2.3 The 486 Family .....	25
2.4 The Explosion of Third-Party CPUs .....	26
<i>AMD Becomes an Industry Force</i> .....	26
<i>Chips and Technologies Crashes and Burns</i> .....	27
<i>Cyrix Joins the Fray</i> .....	27
<i>TI Plays Copy Cat</i> .....	28
<i>IBM Pulls an End Run</i> .....	28
<i>Late Arrivals and Also-Rans</i> .....	29

## Table of Contents

---

2.5 Intel Strikes Back .....	30
<i>“Second-Generation” 386 and 486 Designs</i> .....	30
<i>Pentium</i> .....	31
2.6 Commentary .....	32
<i>The Intel Technology Treadmill</i> .....	32
<i>The Future of the x86 Market</i> .....	33
2.7 For More Information.....	34
<i>Vendor Publications</i> .....	34
<i>Microprocessor Report Articles</i> .....	34
<i>Other Technical References</i> .....	34
<b>3 The x86 Microprocessor Architecture</b> .....	<b>35</b>
<i>Common Ground</i> .....	35
3.1 Programming Model .....	36
<i>Working Registers</i> .....	36
<i>Program Control Registers</i> .....	38
<i>System Registers</i> .....	39
<i>Control Register Functions</i> .....	40
<i>Breakpoint Registers</i> .....	41
3.2 Integer Instructions .....	41
<i>Memory Addressing</i> .....	43
<i>Segmentation</i> .....	44
<i>Memory Paging</i> .....	46
3.3 Floating-Point Architecture .....	47
<i>FPU Instruction Set</i> .....	48
<i>FPU Operand Registers</i> .....	49
<i>FPU Control and Status Registers</i> .....	50
<i>FPU Exception and Trace Registers</i> .....	50
<i>FPU Memory-Based Data Formats</i> .....	50
3.4 CPU Operating Modes .....	52
<i>Interrupt and Exception Processing</i> .....	53
3.5 For More Information.....	53
<i>Vendor Publications</i> .....	53
<i>Microprocessor Report Articles</i> .....	53
<i>Other Technical References</i> .....	54
<b>Part II: The Players</b> .....	<b>55</b>
<b>4 Vendor Profiles</b> .....	<b>57</b>
4.1 Intel .....	57
4.2 Advanced Micro Devices .....	62
4.3 Chips and Technologies .....	66
4.4 Cyrix .....	68
4.5 Texas Instruments .....	72



4.6	IBM	76
4.7	NexGen	79
4.8	For More Information	80
	<i>Vendor Publications</i>	80
	<i>Microprocessor Report Articles</i>	80
	<i>Other Technical References</i>	81
	<i>Other Periodicals</i>	81
<b>Part III: The Products</b>		<b>83</b>
<b>5</b>	<b>Intel 386 Microprocessors</b>	<b>85</b>
	<i>Intel 386 Family Overview</i>	85
5.1	Intel 386 Core Technology	86
	<i>Core Design</i>	87
	<i>Low-Level Instruction Timing</i>	88
	<i>Clock Timing</i>	89
5.2	The Intel i386DX Microprocessor	90
	<i>Features</i>	90
	<i>Cache Support</i>	91
	<i>Floating-Point Support</i>	91
	<i>System Interface</i>	91
	<i>Package and Frequency Options</i>	95
	<i>Vital Statistics</i>	95
5.3	The Intel i386SX Microprocessor	98
	<i>Background</i>	98
	<i>Features</i>	99
	<i>Cache Support</i>	100
	<i>Coprocessor Support</i>	101
	<i>System Interface</i>	101
	<i>Package and Frequency Options</i>	102
5.4	The Intel 80376 Microprocessor	104
	<i>Background</i>	104
	<i>Features</i>	105
	<i>Architecture Modifications</i>	105
	<i>Package and Frequency Options</i>	106
5.5	The Intel i386SL "SuperSet" Microprocessor	107
	<i>Features</i>	108
	<i>System Overview</i>	108
	<i>Cache Support</i>	109
	<i>Floating-Point Support</i>	110
	<i>System Interface Description</i>	110
	<i>Package and Frequency Options</i>	111
5.6	Futures	112

	<i>Geopolitical Pawns?</i> .....	112
5.7	For More Information.....	113
	<i>Vendor Publications</i> .....	113
	<i>Microprocessor Report Articles</i> .....	113
	<i>Other Technical References</i> .....	114
<b>6</b>	<b>Intel 486 Microprocessors</b> .....	117
	<i>Intel 486 Family Overview</i> .....	117
	<i>Architecture Extensions</i> .....	118
	<i>Execution Pipeline</i> .....	121
	<i>Instruction / Data Cache</i> .....	127
	<i>System Interface</i> .....	130
	<i>“SL-Enhanced” Processors</i> .....	133
6.1	The Intel i486DX Microprocessor .....	135
	<i>Floating-Point Unit</i> .....	136
	<i>Processor Clock</i> .....	136
	<i>System Interface</i> .....	137
	<i>Deadlock Backoff</i> .....	141
	<i>SL-Enhancements</i> .....	142
	<i>Vital Statistics</i> .....	142
6.2	The Intel i486DX-50 Microprocessor .....	147
	<i>Background</i> .....	147
	<i>Clock Synthesis Circuit</i> .....	148
	<i>System Interface</i> .....	149
	<i>Vital Statistics</i> .....	149
6.3	The Intel i486DX2 Microprocessor .....	150
	<i>Clock-Doubler Circuitry</i> .....	151
	<i>Relative Performance</i> .....	152
	<i>System Upgrade Good News / Bad News</i> .....	152
	<i>System Interface</i> .....	153
	<i>Vital Statistics</i> .....	153
6.4	The Write-Back-Enhanced IntelDX2 Microprocessor .....	154
	<i>Overview</i> .....	154
	<i>System Interface</i> .....	155
	<i>Performance</i> .....	156
6.5	The IntelDX4 Microprocessor .....	157
	<i>Clock-Multiplier Options</i> .....	158
	<i>Other Enhancements</i> .....	159
	<i>System Interface</i> .....	159
	<i>Vital Statistics</i> .....	160
6.6	The Intel i486SX Microprocessor .....	161
	<i>System Interface</i> .....	162
	<i>Relative Performance</i> .....	163
	<i>Vital Statistics</i> .....	164

6.7	The Intel i486SX2 Microprocessor .....	165
	<i>Vital Statistics</i> .....	165
6.8	The Intel i487SX Microprocessor .....	166
	<i>Features</i> .....	166
	<i>System Interface</i> .....	167
	<i>Vital Statistics</i> .....	167
6.9	The IntelDX2 OverDrive Microprocessor .....	168
	<i>Features</i> .....	168
	<i>Pinout</i> .....	169
	<i>Vital Statistics</i> .....	170
6.10	The IntelDX4 OverDrive Microprocessor .....	171
6.11	The IntelSX2 OverDrive Microprocessor .....	172
6.12	The Intel i486SL Microprocessor .....	173
	<i>Background</i> .....	174
	<i>"The Best Laid Plans..."</i> .....	175
	<i>Vital Statistics</i> .....	176
6.13	The Intel "RapidCAD" 386 Microprocessor .....	177
	<i>Features</i> .....	177
	<i>Vital Statistics</i> .....	178
6.14	Futures .....	179
	<i>The "P24T"</i> .....	179
6.15	Commentary .....	179
6.16	For More Information.....	182
	<i>Vendor Publications</i> .....	182
	<i>Microprocessor Report Articles</i> .....	182
	<i>Other Technical References</i> .....	185
	<i>Other Periodicals</i> .....	185
<b>7</b>	<b>AMD 386 and 486 Microprocessors</b> .....	<b>187</b>
	<i>Background</i> .....	187
7.1	Core Design .....	189
	<i>Design Methodology</i> .....	189
	<i>Compatibility and Performance</i> .....	190
	<i>Availability</i> .....	191
7.2	The AMD Am386SX and Am386SXL Microprocessors .....	192
	<i>Features</i> .....	192
	<i>System Interface</i> .....	193
	<i>Vital Statistics</i> .....	193
7.3	The AMD Am386SXLV Microprocessor .....	195
	<i>Architecture Extensions</i> .....	195
	<i>System Interface</i> .....	196
	<i>Vital Statistics</i> .....	197
7.4	The AMD Am386DX and Am386DXL Microprocessors .....	198
	<i>System Interface</i> .....	198

## Table of Contents

---

<i>Vital Statistics</i> .....	199
7.5 The AMD Am386DXLV Microprocessor .....	200
<i>Features</i> .....	200
<i>Vital Statistics</i> .....	201
7.6 The AMD Am386SC300 “Elan” Microprocessor .....	202
<i>Features</i> .....	203
<i>System Interface</i> .....	204
7.7 The AMD Am486SX and Am486SX2 Microprocessors .....	206
<i>System Interface</i> .....	207
<i>Vital Statistics</i> .....	208
7.8 The AMD Am486SXLV Microprocessor .....	209
<i>Architecture Extensions</i> .....	209
<i>System Interface</i> .....	210
<i>Vital Statistics</i> .....	211
7.9 The AMD Am486DX Microprocessor .....	212
<i>Vital Statistics</i> .....	212
7.10 The AMD Am486DX2 Microprocessor .....	213
<i>Features</i> .....	213
<i>Vital Statistics</i> .....	213
7.11 The AMD Am486DXL and Am486DXLV Microprocessors .....	214
<i>System Interface</i> .....	215
<i>Vital Statistics</i> .....	215
7.12 The AMD Am486DX4 Microprocessor .....	216
7.13 Futures .....	217
7.14 Commentary .....	218
<i>Technical Comparisons</i> .....	219
<i>Legal Entanglements</i> .....	219
<i>Business Strategy</i> .....	221
<i>Production Limitations</i> .....	222
7.15 For More Information.....	223
<i>Vendor Publications</i> .....	223
<i>Microprocessor Report Articles</i> .....	224
<i>Other Technical References</i> .....	226
<i>Other Periodicals</i> .....	226
<b>8 C&amp;T 386 Microprocessors</b> .....	<b>227</b>
<i>Overview</i> .....	228
<i>Core Design</i> .....	228
8.1 The C&T 38600DX Microprocessor .....	230
<i>System Interface</i> .....	230
<i>Package and Frequency Options</i> .....	231
<i>Relative Performance</i> .....	231
8.2 The C&T 38605DX Microprocessor .....	232
<i>Cache Design</i> .....	232

	<i>System Interface</i> .....	233
	<i>Relative Performance</i> .....	234
8.3	Commentary .....	235
	<i>Too Little, Too Different, Too Late?</i> .....	235
8.4	For More Information.....	236
	<i>Vendor Publications</i> .....	237
	<i>Microprocessor Report Articles</i> .....	237
<b>9</b>	<b>Cyrix 486 Microprocessors</b> .....	<b>239</b>
9.1	Core Design.....	239
	<i>Pipeline Performance</i> .....	241
	<i>Architecture Enhancements</i> .....	243
	<i>System Management and Standby Modes</i> .....	243
9.2	The Cyrix Cx486SLC and Cx486SLC/e Microprocessors.....	245
	<i>Features</i> .....	246
	<i>Cache Configuration</i> .....	246
	<i>Instruction Set Additions</i> .....	248
	<i>Configuration Registers</i> .....	249
	<i>System Interface</i> .....	252
	<i>Clocking Regimes</i> .....	254
	<i>Relative Performance</i> .....	254
	<i>Package and Pinout</i> .....	255
	<i>Vital Statistics</i> .....	255
9.3	The Cyrix Cx486SLC/e-V Microprocessor .....	257
	<i>Vital Statistics</i> .....	257
9.4	The Cyrix Cx486SLC2 Microprocessors.....	258
	<i>Vital Statistics</i> .....	258
9.5	The Cyrix Cx486DLC Microprocessor .....	259
	<i>System Interface</i> .....	259
	<i>Relative Performance</i> .....	260
	<i>Mortal Statistics</i> .....	261
9.6	The Cyrix Cx486SRx2 Microprocessor.....	262
	<i>Product Overview</i> .....	263
	<i>Frequency Options</i> .....	264
9.7	The Cyrix Cx486DRx2 Microprocessor .....	265
	<i>Frequency Options</i> .....	266
9.8	The Cyrix Cx486S and Cx486S2 Microprocessors.....	267
	<i>Cache Characteristics</i> .....	268
	<i>Floating-Point Strategy</i> .....	268
	<i>System Interface</i> .....	269
	<i>Clocking Regimes</i> .....	272
	<i>Instruction Set Additions</i> .....	273
	<i>Configuration Registers</i> .....	273
	<i>Vital Statistics</i> .....	276

9.9	The Cyrix Cx486DX and Cx486DX2 Microprocessors .....	277
	<i>Cache Design</i> .....	278
	<i>Floating-Point Unit</i> .....	278
	<i>System Interface</i> .....	279
	<i>Power Management</i> .....	279
	<i>Relative Performance</i> .....	280
	<i>Vital Statistics</i> .....	280
9.10	Commentary .....	282
	<i>What's in a Name?</i> .....	282
	<i>Business Issues</i> .....	283
	<i>Legal Issues</i> .....	284
	<i>Compatibility</i> .....	284
9.11	For More Information.....	285
	<i>Vendor Publications</i> .....	285
	<i>Microprocessor Report Articles</i> .....	285
<b>10</b>	<b>IBM 386 and 486 Microprocessors</b> .....	<b>287</b>
	<i>Dysfunctional Corporate Relations</i> .....	287
	<i>Creatively Licensed</i> .....	288
	<i>Competitive Thrusts</i> .....	289
10.1	The IBM 386SLC Microprocessor .....	290
	<i>Cache Configuration</i> .....	291
	<i>System Interface</i> .....	291
	<i>Programming Model Extensions</i> .....	296
	<i>Instruction Set Extensions</i> .....	298
	<i>Vital Statistics</i> .....	299
10.2	The IBM BL486SLC2 Microprocessor .....	300
	<i>Cache Configuration</i> .....	300
	<i>Clocking Regimes</i> .....	301
	<i>System Interface</i> .....	301
	<i>Architecture Extensions</i> .....	303
	<i>Vital Statistics</i> .....	303
10.3	The IBM BL486SX2/SX3 "Blue Lightning" Microprocessor .....	305
	<i>System Interface</i> .....	306
	<i>Vital Statistics</i> .....	306
10.4	The IBM BL486DX and BL486DX2 Microprocessors .....	308
	<i>System Interface</i> .....	309
	<i>Vital Statistics</i> .....	310
10.5	Futures .....	311
10.6	Commentary .....	312
	<i>Strategic Direction</i> .....	312
	<i>Terminology Footnote</i> .....	313
10.7	For More Information.....	313
	<i>Vendor Publications</i> .....	314

	<i>Microprocessor Report Articles</i> .....	314
	<i>Other Periodicals</i> .....	315
<b>11</b>	<b>Texas Instruments 486 Microprocessors</b> .....	<b>317</b>
11.1	The TI486SLC/E and TI486SLC/E-V Microprocessors .....	318
	<i>System Interface</i> .....	319
	<i>Compatibility</i> .....	319
	<i>Vital Statistics</i> .....	321
11.2	The TI486DLC/E and TI486DLC/E-V Microprocessors .....	322
	<i>System Interface</i> .....	323
	<i>Vital Statistics</i> .....	323
11.3	The TI486SXLC and TI486SXLC2 Microprocessors .....	325
	<i>Clock Circuitry</i> .....	326
	<i>Vital Statistics</i> .....	326
11.4	The TI486SXL and TI486SXL2 Microprocessors .....	327
	<i>Vital Statistics</i> .....	328
11.5	The TI "Rio Grande" Processor Chip Set .....	329
	<i>Support Logic</i> .....	331
	<i>Vital Statistics</i> .....	332
11.6	Commentary .....	333
	<i>The i486SL Redux?</i> .....	334
11.7	For More Information .....	335
	<i>Vendor Publications</i> .....	335
	<i>Microprocessor Report Articles</i> .....	335
<b>Part IV:</b>	<b>Pentium-Class Processors</b> .....	<b>337</b>
<b>12</b>	<b>The Intel Pentium Family</b> .....	<b>339</b>
	<i>Overview</i> .....	339
	<i>Pipeline Operation</i> .....	341
	<i>Instruction Issue Rules</i> .....	346
	<i>Instruction Cache and TLB</i> .....	349
	<i>Data Cache and TLBs</i> .....	352
	<i>D-Cache Snooping</i> .....	354
	<i>Branch Prediction Logic</i> .....	355
	<i>Floating-Point Unit</i> .....	358
	<i>Architecture Extensions</i> .....	363
	<i>Software Optimization</i> .....	367
12.1	The Intel 0.8 $\mu$ Pentium "P5" .....	368
	<i>System Interface</i> .....	369
	<i>System Management Functions</i> .....	378
	<i>Bus Sizing</i> .....	379
	<i>Functional Redundancy Checking</i> .....	380
	<i>Vital Statistics</i> .....	382

## Table of Contents

---

12.2	The Intel 0.6 $\mu$ Pentium “P54C”	383
	<i>Overview</i>	384
	<i>Bus Interface</i>	386
	<i>Multiprocessor Support</i>	387
	<i>Interrupt Control Logic</i>	388
	<i>Clock-Generator Circuitry</i>	389
	<i>Vital Statistics</i>	389
12.3	Futures	390
	<i>The Intel “P24T”</i>	390
12.4	Commentary	390
	<i>Competition with RISC</i>	394
12.5	For More Information	396
	<i>Vendor Publications</i>	396
	<i>Microprocessor Report Articles</i>	396
	<i>Other Technical References</i>	398
	<i>Other Periodicals</i>	399
<b>13</b>	<b>NexGen Microprocessors</b>	<b>401</b>
13.1	The NexGen Nx586 Microprocessor	401
	<i>Core Design</i>	402
	<i>Is it Superscalar Yet?</i>	406
	<i>Cache Logic</i>	406
	<i>Execution Timing</i>	407
	<i>Branch Prediction</i>	409
	<i>Floating-Point Unit</i>	410
	<i>System Interface</i>	411
	<i>Performance</i>	412
	<i>Vital Statistics</i>	413
13.2	Commentary	413
13.3	For More Information	415
	<i>Vendor Publications</i>	415
	<i>Microprocessor Report Articles</i>	415
	<i>Other Periodicals</i>	416

## Volume II

<b>Part V: Perspective</b>	417	
<b>14 Core Design and Implementation</b>	<b>419</b>	
	<i>How Device Design Affects Product Selection</i>	419
14.1	Overview of Core Functional Differences	420
14.2	Implementation Technology Alternatives	423
	<i>Complexity in the x86 Architecture</i>	424
14.3	Microprogrammed vs Pipelined Designs	427
	<i>Microprogramming Basics</i>	428



	<i>Prelude to Pipelining – A Hardwired Machine</i> .....	435
	<i>Pipelining for Performance</i> .....	439
	<i>Combining Microprogramming and Pipelining</i> .....	444
14.4	<b>Pipeline Design Comparison</b> .....	446
	<i>The Intel/AMD 486 Pipeline</i> .....	447
	<i>The Cyrix/TI/IBM 486 Pipeline</i> .....	450
	<i>The Intel Pentium Pipeline</i> .....	452
14.5	<b>Instruction Execution Timing Comparisons</b> .....	454
	<i>Register-to-Register ALU Operations</i> .....	455
	<i>Memory-to-Register Load Operations</i> .....	456
	<i>Memory-to-Register ALU Operations</i> .....	458
	<i>Register-to-Memory ALU Operations</i> .....	459
	<i>Jump and Branch Operations</i> .....	462
	<i>Integer Multiplication Operations</i> .....	468
14.6	<b>Core Performance Summary</b> .....	469
	<i>Micro-performance Estimates</i> .....	471
14.7	<b>On-Chip Cache</b> .....	474
	<i>Cache Configuration</i> .....	475
	<i>Cache Size</i> .....	476
	<i>Line Replacement Policies</i> .....	477
	<i>External Cache Interface</i> .....	478
14.8	<b>Post-Pentium Implementations</b> .....	481
	<i>Register Renaming</i> .....	481
	<i>Multiple Execution Units and Reservation Stations</i> .....	483
	<i>Reorder Buffer</i> .....	483
	<i>Next-Generation Implementations</i> .....	484
14.9	<b>Device Design and Layout Comparison</b> .....	485
	<i>Intel i386DX Die Analysis</i> .....	485
	<i>AMD Am386SX/DX Die Analysis</i> .....	487
	<i>Intel i486SX/DX Die Analysis (1.0<math>\mu</math>)</i> .....	490
	<i>Intel i486DX/DX2 Die Analysis (0.8<math>\mu</math>)</i> .....	492
	<i>IntelDX4 Die Analysis</i> .....	494
	<i>AMD Am486SX/DX Die Analysis</i> .....	496
	<i>Cyrix Cx486SLC/DLC Die Analysis</i> .....	498
	<i>Pentium Die Analysis (0.8<math>\mu</math>)</i> .....	500
	<i>Pentium Die Analysis (0.6<math>\mu</math>)</i> .....	502
	<i>NexGen Nx586 Die Analysis</i> .....	505
14.10	<b>For More Information...</b> .....	506
	<i>Microprocessor Report Articles</i> .....	506
	<i>Other Technical References</i> .....	507
<b>15</b>	<b>Manufacturing Costs</b> .....	509
15.1	<b>Manufacturing Cost: Theory</b> .....	509
	<i>Wafer and Die Cost</i> .....	510

## Table of Contents

---

	<i>Manufacturing Defects and Die Yield</i> .....	512
	<i>Die Testing</i> .....	515
	<i>Packaging</i> .....	516
	<i>Final Test</i> .....	517
	<i>Other Costs</i> .....	518
15.2	<b>Manufacturing Cost: Practice</b> .....	523
	<i>Process Technology Comparison</i> .....	523
	<i>Wafer Costs</i> .....	525
	<i>Die Area and Net Yield</i> .....	526
	<i>Packaging Costs</i> .....	528
	<i>Design Cost Comparison</i> .....	529
15.3	<b>Manufacturing Cost: Summary</b> .....	530
15.4	<b>Commentary</b> .....	532
	<i>Intel vs RISC Costs</i> .....	533
	<i>Price vs Cost</i> .....	534
15.5	<b>For More Information</b> .....	536
	<i>Vendor Publications</i> .....	536
	<i>Microprocessor Report Articles</i> .....	536
	<i>Other Technical References</i> .....	536
<b>16</b>	<b>Legal Issues</b> .....	537
	<i>Issues Affecting the x86 Business</i> .....	537
16.1	<b>Patents</b> .....	539
	<i>Value of Patent Protection</i> .....	539
	<i>Criteria for a Patent</i> .....	540
	<i>Patent Infringement and Licensing</i> .....	542
16.2	<b>Copyrights</b> .....	544
	<i>Software Copyright Protection</i> .....	544
	<i>Applying Software Copyrights to Microprocessors</i> .....	545
	<i>Clean-Room Product Engineering</i> .....	546
16.3	<b>Trademarks</b> .....	549
16.4	<b>Trade Secrets</b> .....	550
16.5	<b>x86-Related Litigation</b> .....	551
	<i>Intel v. AMD</i> .....	554
	<i>AMD v. Intel</i> .....	562
	<i>Intel v. Cyrix</i> .....	563
	<i>Intel v. ULSI System Technology</i> .....	567
	<i>Intel v. Chips and Technologies</i> .....	568
	<i>Texas Instruments v. Cyrix</i> .....	569
16.6	<b>Commentary</b> .....	571
16.7	<b>For More Information</b> .....	572
	<i>Microprocessor Report Articles</i> .....	572
<b>17</b>	<b>Compatibility Issues</b> .....	577

17.1	Overview .....	578
	<i>Degrees of Compatibility</i> .....	579
	<i>Hardware Compatibility Challenges</i> .....	580
	<i>Software-Compatibility Challenges</i> .....	581
17.2	Compatibility Assurance Methodologies .....	583
	<i>The Intel Approach</i> .....	583
	<i>The AMD Approach</i> .....	584
	<i>The C&amp;T Approach</i> .....	584
	<i>The Cyrix Approach</i> .....	585
	<i>The TI Approach</i> .....	586
	<i>The IBM Approach</i> .....	586
17.3	Case Studies .....	587
	<i>Pushing the Envelope of Market Acceptance</i> .....	587
	<i>Aftershocks</i> .....	589
17.4	Commentary .....	592
17.5	For More Information.....	593
	<i>Vendor Publications</i> .....	593
	<i>Microprocessor Report Articles</i> .....	593
	<i>Other Technical References</i> .....	594
	<i>Other Periodicals</i> .....	594
<b>18</b>	<b>Future Directions</b> .....	<b>595</b>
18.1	Evolutionary Futures .....	595
	<i>x86 Generations</i> .....	595
	<i>386-Class Derivatives</i> .....	596
	<i>486-Class Derivatives</i> .....	597
	<i>Pentium Derivatives</i> .....	598
18.2	AMD's K5 .....	599
	<i>Tackling the x86 Bottleneck</i> .....	600
	<i>Dispatching Four Instructions Per Cycle</i> .....	601
	<i>Dispatch Capability</i> .....	603
	<i>Caches and Memory Management</i> .....	606
	<i>Design Technology</i> .....	607
	<i>Competing with Pentium</i> .....	608
18.3	Cyrix's M1 .....	609
	<i>Pipeline Design</i> .....	610
	<i>Register Renaming</i> .....	612
	<i>Memory Bypassing</i> .....	612
	<i>Branch Logic</i> .....	612
	<i>Die Size</i> .....	613
	<i>Advanced Microarchitecture</i> .....	614
	<i>Avoiding Appendix H</i> .....	615
	<i>Pinout and Bus Interface</i> .....	615
	<i>Competing Against K5</i> .....	616

## Table of Contents

---

<i>Cyrix's M9</i> .....	617
18.4 The Intel "P6" and "P7" .....	617
<i>Pipelined Designs</i> .....	618
<i>Early P6 Details</i> .....	619
<i>Hybrid Architectures</i> .....	619
<i>Intel/HP Partnership</i> .....	620
18.5 Commentary .....	621
18.6 For More Information... ..	622
<i>Microprocessor Report Articles</i> .....	622
<i>Other Technical References</i> .....	624
<i>Other Periodicals</i> .....	625
<b>Part VI: Price and Performance</b> .....	<b>627</b>
<b>19 Pricing Structures</b> .....	<b>629</b>
19.1 Price and Availability .....	630
<i>Intel 486 and Pentium Products</i> .....	630
<i>AMD 386 and 486 Products</i> .....	632
<i>Cyrix 486 Products</i> .....	633
<i>Texas Instruments 486 Products</i> .....	633
<i>IBM 486 Products</i> .....	633
<i>NexGen 586 Products</i> .....	634
19.2 Long-Term Trends .....	634
19.3 Commentary .....	635
19.4 For More Information... ..	637
<i>Microprocessor Report Articles</i> .....	637
<i>Other Technical References</i> .....	638
<i>Other Periodicals</i> .....	638
<b>20 Performance Measurement and Analysis</b> .....	<b>639</b>
20.1 Benchmarking Issues .....	639
<i>The Need for Good PC Benchmarks</i> .....	640
<i>Benchmark Design</i> .....	640
<i>Binary vs Source Benchmarks</i> .....	641
<i>Performance as a Surface</i> .....	642
20.2 Non-PC-Specific Benchmarks .....	644
<i>MIPS and MFLOPS</i> .....	644
<i>Whetstone</i> .....	645
<i>Dhrystone</i> .....	646
<i>Linpack</i> .....	647
<i>SPEC</i> .....	647
20.3 PC CPU Benchmarks .....	655
<i>ZD Labs PC Bench and CPUmark</i> .....	655
<i>WinBench</i> .....	657

<i>Norton SI</i> .....	657
<i>Landmark</i> .....	658
<i>Power Meter</i> .....	658
<i>iCOMP</i> .....	659
<i>CPU Benchmark Results</i> .....	661
20.4 Application-Based PC Benchmarks .....	668
<i>BAPCo</i> .....	668
<i>Winstone</i> .....	672
<i>Performance Effects of Write-Back Caches</i> .....	674
20.5 Commentary .....	677
20.6 References .....	678
<i>Vendor Publications</i> .....	678
<i>Microprocessor Report Articles</i> .....	678
<i>Other Technical References</i> .....	679
<i>Other Periodicals</i> .....	679
<b>Appendix A: Vendor Contacts</b> .....	A-1
<i>Advanced Micro Devices</i> .....	A-1
<i>Chips &amp; Technologies</i> .....	A-1
<i>Cyrix</i> .....	A-2
<i>IBM</i> .....	A-2
<i>Intel</i> .....	A-2
<i>NexGen</i> .....	A-2
<i>Texas Instruments</i> .....	A-3
<b>Appendix B: Summary of Processor Specifications</b> .....	B-1
<i>Appendix B Table of Contents</i> .....	B-1
<i>General</i> .....	B-2
<i>Features</i> .....	B-3
<i>Cache</i> .....	B-3
<i>System Interface</i> .....	B-4
<i>Power Characteristics</i> .....	B-4
<i>Performance</i> .....	B-5
<i>Technology</i> .....	B-5
<i>Price</i> .....	B-5
<b>Appendix C: System Benchmark Data</b> .....	C-1
<b>Appendix D: Intel x86 Patents</b> .....	D-1
<b>Appendix E: Floating-Point Bug Strikes Pentium</b> .....	E-1
E.1 Testing for the Bug .....	E-2
E.2 Anatomy of the Bug .....	E-2
E.3 Probability of Occurrence .....	E-3
E.4 Impact on Applications .....	E-6

## Table of Contents

---

E.5 IBM Claims.....	E-7
E.6 Effect on Users.....	E-8
E.7 Intel's Response.....	E-9
E.8 For More Information.....	E-10

# List of Figures

## Volume I

### 1 The x86 Business Climate

Figure 1-1. x86 processor introductions: the early years.....	6
Figure 1-2. x86 processor introductions: now.....	7
Figure 1-3. Intel 386/486 product price trends.....	8
Figure 1-4. x86 product differentiation: the early years. ....	9
Figure 1-5. x86 product differentiation: now. ....	9
Figure 1-6. x86 core frequency design options. ....	10
Figure 1-7. x86 core microarchitecture design options. ....	10
Figure 1-8. x86 cache size design options. ....	11
Figure 1-9. x86 power strategy design options.....	11

### 2 x86 Family Heritage

Figure 2-1. 8086 programming model.....	20
---	----

### 3 The x86 Microprocessor Architecture

Figure 3-1. The 386 user programming model.....	37
Figure 3-2. The 386 system programming model.....	39
Figure 3-3. The 386 control register field definitions. ....	40
Figure 3-4. Variable-length instruction formats. ....	43
Figure 3-5. Memory operand address components.....	44
Figure 3-6. Segmentation base and offset address computation.....	45
Figure 3-7. PMMU address translation mechanism. ....	47
Figure 3-8. Floating-point programming model.....	49
Figure 3-9. Memory-based data formats for floating-point data types.....	51

### 4 Vendor Profiles

Figure 4-1. Texas Instruments' 1993 product sales ratios. ....	73
Figure 4-2. IBM 1993 product sales ratios. ....	77

### 5 Intel 386 Microprocessors

Figure 5-1. Intel i386DX system interface. ....	91
Figure 5-2. Intel i386DX PGA pinout. ....	96
Figure 5-3. Intel i386DX PQFP pinout.....	97

## List of Figures

---

Figure 5-4. Intel i386SX system interface. . . . .	100
Figure 5-5. Intel i386SX PQFP pinout. . . . .	103
Figure 5-6. Intel i386SL functional system partitioning. . . . .	108
Figure 5-7. Intel 386SL direct cache support. . . . .	110
Figure 5-8. Intel i386SL 10-chip minimum system design. . . . .	111
<b>6 Intel 486 Microprocessors</b>	
Figure 6-1. Intel 486 programming model. . . . .	119
Figure 6-2. Intel 486 programming model additions. . . . .	120
Figure 6-3. Intel 486 microarchitecture. . . . .	122
Figure 6-4. Intel 486 pipeline stages. . . . .	122
Figure 6-5. Intel 486 pipeline timing for simple operations. . . . .	125
Figure 6-6. Intel 486 pipeline timing for reg-to-mem operations. . . . .	126
Figure 6-7. Intel 486 pipeline timing for branch operations. . . . .	127
Figure 6-8. Intel i486DX block diagram. . . . .	136
Figure 6-9. Intel i486DX system interface. . . . .	137
Figure 6-10. Intel i486DX system deadlock avoidance. . . . .	142
Figure 6-11. Intel i486DX PGA pinout. . . . .	144
Figure 6-12. Intel i486DX PQFP pinout. . . . .	145
Figure 6-13. Intel i486DX SQFP pinout. . . . .	146
Figure 6-14. Intel i486DX-50 PLL clock synthesis circuit. . . . .	148
Figure 6-15. Intel i486DX2 PLL clock-doubler circuitry. . . . .	151
Figure 6-16. Upgrade socket interface to i486SX (PGA). . . . .	163
Figure 6-17. Upgrade socket interface to i486SX (PQFP/SQFP). . . . .	163
Figure 6-18. Intel i486SL system partitioning. . . . .	174
<b>7 AMD 386 and 486 Microprocessors</b>	
Figure 7-1. AMD Am386SXLV system interface. . . . .	197
Figure 7-2. AMD Am386DXLV system interface. . . . .	201
Figure 7-3. AMD Am386SC300 “Elan” Block diagram and system interface. . . . .	203
Figure 7-4. AMD Am486SX and Am486SX2 system interface. . . . .	207
Figure 7-5. AMD Am486SXLV system interface. . . . .	210
Figure 7-6. AMD Am486DXL/Am486DXLV system interface. . . . .	215



**8 C&T 386 Microprocessors**

Figure 8-1. C&amp;T 38605DX system interface..... 233

**9 Cyrix 486 Microprocessors**

Figure 9-1. Cyrix 486 core microarchitecture..... 240

Figure 9-2. Cyrix core Suspend Mode state transition diagram..... 244

Figure 9-3. Cyrix Cx486SLC system registers..... 250

Figure 9-4. Cyrix Cx486SLC/e configuration control register 0. .... 251

Figure 9-5. Cyrix Cx486SLC/e configuration control register 1. .... 251

Figure 9-6. Cyrix Cx486SLC and Cx486SLC/e system interface. .... 252

Figure 9-7. Cyrix Cx486SLC/e PQFP pinout. .... 255

Figure 9-8. Cyrix Cx486DLC system interface. .... 260

Figure 9-9. Cyrix Cx486S and Cx486S2 system interface..... 270

Figure 9-10. Cyrix Cx486S and Cx486S2 system registers..... 274

Figure 9-11. Cyrix Cx486S and Cx486S2 configuration control register 1..... 275

Figure 9-12. Cyrix Cx486S and Cx486S2 configuration control register 2..... 275

Figure 9-13. Cyrix Cx486S and Cx486S2 configuration control register 3..... 276

Figure 9-14. Cyrix Cx486DX and Cx486DX2 system interface..... 279

**10 IBM 386 and 486 Microprocessors**

Figure 10-1. IBM 386SLC system interface..... 292

Figure 10-2. IBM 386SLC system register model. .... 296

Figure 10-3. IBM 386SLC model-specific register 1000H..... 297

Figure 10-4. IBM BL486SLC2 system interface. .... 302

Figure 10-5. IBM BL486SLC2 model specific register 1002H..... 303

Figure 10-6. IBM BL486SX2/SX3 system interface..... 306

Figure 10-7. IBM BL486DX and BL486DX2 system interface. .... 309

**11 Texas Instruments 486 Microprocessors**

Figure 11-1. TI486SLC/E and TI486SLC/E-V system interface..... 319

Figure 11-2. TI486DLC/E and TI486DLC/E-V system interface. .... 323

Figure 11-3. TI "Rio Grande" CPU block diagram..... 330

Figure 11-4. TI "Rio Grande" system interface. .... 331

Figure 11-5. Divergence between TI and Cyrix product strategies..... 333

**12 The Intel Pentium Family**

Figure 12-1. Intel Pentium microprocessor block diagram. . . . . 341

Figure 12-2. Intel Pentium integer unit pipeline operation. . . . . 342

Figure 12-3. Pentium integer unit data-path pipeline stages. . . . . 343

Figure 12-4. Pentium dual-instruction issue efficiency. . . . . 350

Figure 12-5. Pentium instruction cache hit rates. . . . . 350

Figure 12-6. Pentium split-fetch instruction cache operation. . . . . 351

Figure 12-7. Pentium data cache hit rates. . . . . 352

Figure 12-8. Pentium data cache interleaved bank partitioning. . . . . 353

Figure 12-9. Pentium interleaved data cache operation. . . . . 354

Figure 12-10. Pentium dual-access D-cache efficiency. . . . . 354

Figure 12-11. Pentium branch target buffer organization. . . . . 356

Figure 12-12. Pentium branch history bit state transitions. . . . . 357

Figure 12-13. Branch-prediction logic accuracy. . . . . 359

Figure 12-14. CPU registers after invoking CPUID with EAX = 0. . . . . 365

Figure 12-15. CPU registers after invoking CPUID with EAX = 1. . . . . 365

Figure 12-16. Back-to-back Pentium cache-fill timing. . . . . 372

Figure 12-17. Pentium cache-line invalidation sequence timing. . . . . 376

Figure 12-18. Packing/unpacking logic for partial-word transfers. . . . . 379

Figure 12-19. Functional redundancy checking interconnections. . . . . 380

Figure 12-20. 0.8 $\mu$  and 0.6 $\mu$  Pentium die size comparison. . . . . 385

Figure 12-21. 0.6 $\mu$  Pentium multiprocessing system architecture. . . . . 387

**13 NexGen Microprocessors**

Figure 13-1. NexGen Nx586 microarchitecture. . . . . 403

Figure 13-2. NexGen Nx586 execution pipeline timing. . . . . 408

Figure 13-3. NexGen Nx586 system partitioning. . . . . 412

**Volume II**

**14 Core Design and Implementation**

Figure 14-1. Instruction encoding formats for x86 processors. . . . . 425

Figure 14-2. Simple microprogrammed machine organization. . . . . 430

Figure 14-3. Nonpipelined (386-class) microarchitecture. . . . .	433
Figure 14-4. Machine organization with hardwired-control. . . . .	436
Figure 14-5. Hardwired machine organization with enhanced datapath. . . . .	438
Figure 14-6. Simple pipelined microarchitecture. . . . .	441
Figure 14-7. Microarchitecture with microcoded and hardwired control. . . . .	445
Figure 14-8. Intel 486 pipeline organization. . . . .	448
Figure 14-9. Cyrix 486 pipeline organization. . . . .	450
Figure 14-10. Intel Pentium pipeline organization (one of two pipelines). . . . .	453
Figure 14-11. Intel 486 pipeline timing for reg-to-reg ops. . . . .	455
Figure 14-12. Cyrix 486 pipeline timing for reg-to-reg ops. . . . .	456
Figure 14-13. Intel 486 pipeline timing for memory loads. . . . .	456
Figure 14-14. Cyrix 486 pipeline timing for memory loads. . . . .	457
Figure 14-15. Intel 486 pipeline timing for mem-to-reg ops. . . . .	458
Figure 14-16. Cyrix 486 pipeline timing for mem-to-reg ops. . . . .	459
Figure 14-17. Pentium pipeline timing for mem-to-reg ops. . . . .	459
Figure 14-18. Intel 486 pipeline timing for reg-to-mem ops. . . . .	460
Figure 14-19. Cyrix 486 pipeline timing for reg-to-mem ops. . . . .	460
Figure 14-20. Pentium pipeline timing for reg-to-mem ops. . . . .	462
Figure 14-21. Source code for jump and branch timing examples. . . . .	463
Figure 14-22. Intel 486 pipeline timing for branch operations. . . . .	464
Figure 14-23. Cyrix 486 pipeline timing for branch operations. . . . .	465
Figure 14-24. Pentium pipeline mispredicted branch timing. . . . .	467
Figure 14-25. A decoupled, out-of-order superscalar processor organization. . . . .	482
Figure 14-26. Intel i386DX die photo (6.2 × 6.9 mm at 1.0μ). . . . .	486
Figure 14-27. AMD Am386SX/DX die photo (6.5 × 7.5 mm at 0.9μ). . . . .	488
Figure 14-28. i486DX die photo (10.5 × 15.7 mm <sup>2</sup> at 1.0 μ). . . . .	490
Figure 14-29. Intel i486DX/DX2 die photo (6.9 × 11.9 mm at 0.8μ). . . . .	493
Figure 14-30. IntelDX4 die photo (8.6 × 8.9 mm at 0.6μ). . . . .	494
Figure 14-31. AMD Am486SX/DX/DX2 die photo (9.1 × 9.7 mm at 0.7μ). . . . .	497
Figure 14-32. Cx486SLC/DLC die photo (10.4 × 10.4 mm at 0.8μ). . . . .	499
Figure 14-33. Pentium “P5” die photo (16.7 × 17.6 mm at 0.8μ). . . . .	500

## List of Figures

---

Figure 14-34. Pentium “P54C” die photo (13.3 × 12.3 mm at 0.6μ).....	503
Figure 14-35. NexGen Nx586 die photo (0.5μ, size unknown). ....	506
<b>15 Manufacturing Costs</b>	
Figure 15-1. Division of costs in a typical 0.8-micron IC fab.....	510
Figure 15-2. Microprocessor die placement on a circular wafer. ....	511
Figure 15-3. The defect-density learning curve. ....	513
Figure 15-4. Good-die yield vs die size and defect density. ....	513
Figure 15-5. Fully utilized vs pad-limited die layouts.....	514
Figure 15-6. Standard package types and sizes (actual size). ....	517
Figure 15-7. Microprocessor yield vs frequency.....	521
<b>16 Legal Issues</b>	
Figure 16-1. The Intel 386 MMU architecture patent, number 4,972,338.....	541
<b>17 Compatibility Issues</b>	
Figure 17-1. Dimensions of microprocessor product compatibility.....	579
<b>18 Future Directions</b>	
Figure 18-1. Division of x86 product line shipments vs time, in millions of units.....	596
Figure 18-2. Block diagram of AMD’s K5.....	601
Figure 18-3. K5 instruction translation process.....	602
Figure 18-4. K5 pipeline timing. ....	605
Figure 18-5. Cyrix “M1” device microarchitecture.....	610
Figure 18-6. Cyrix “M1” and Pentium pipeline structures. ....	611
Figure 18-7. Anticipated Intel product introductions (as of 6/92).....	618
<b>19 Pricing Structures</b>	
Figure 19-1. Long term Intel price trends. ....	634
Figure 19-2. Low end 386 and 486 price trends.....	636
<b>20 Performance Measurement and Analysis</b>	
Figure 20-1. Performance of various 486DX2-66 systems on SYSmark93 .....	643
Figure 20-2. SPEC benchmark results for x86 and low-end to midrange RISC processors .	653
Figure 20-3. SPEC benchmark results for x86 and midrange to high-end RISC processors	654
Figure 20-4. Correlation among Intel’s integer benchmarks.....	664
Figure 20-5. Correlation among Intel’s floating-point benchmarks. ....	665

Figure 20-6. CPUmark16 results for various x86 processors. . . . . 666

Figure 20-7. Variation in CPUmark16 among systems of each processor type. . . . . 667

Figure 20-8. Variation in SYSmark92 ratings among systems of each processor type. . . . . 670

Figure 20-9. Variation in SYSmark93 ratings among systems of each processor type. . . . . 671

Figure 20-10. Variation in Winstone94 ratings among systems of each processor type. . . . . 673

Figure 20-11. Benchmark results illustrating source of Winstone variation. . . . . 675

**Appendix E: Floating-Point Bug Strikes Pentium**

Figure E-1. The probability of an error affecting a given bit position. . . . . E-4



# List of Tables

## Volume I

### 1 The x86 Business Climate

Table 1-1. Intel 386 and 486 product line summary.....	3
Table 1-2. Alternate vendor 386 and 486 product line summary.....	4
Table 1-3. Pentium- and post-Pentium product line summary.....	5

### 3 The x86 Microprocessor Architecture

Table 3-1. The 386 integer instruction set summary.....	42
Table 3-2. Memory operand addressing modes.....	44
Table 3-3. Floating-point instruction set summary.....	48
Table 3-4. CPU operating modes.....	52

### 4 Vendor Profiles

Table 4-1. Intel company profile.....	58
Table 4-2. Intel financial results ('89-'93).....	59
Table 4-3. AMD company profile.....	62
Table 4-4. AMD financial results ('89-'93).....	64
Table 4-5. Chips and Technologies company profile.....	66
Table 4-6. Chips and Technologies financial results ('89-'93).....	67
Table 4-7. Cyrix company profile.....	68
Table 4-8. Cyrix financial results ('89-'93).....	69
Table 4-9. Texas Instruments company profile.....	72
Table 4-10. Texas Instruments financial results ('89-'93).....	74
Table 4-11. IBM company profile.....	76
Table 4-12. IBM financial results ('89-'93).....	77
Table 4-13. NexGen company profile.....	79

### 5 Intel 386 Microprocessors

Table 5-1. Intel i386DX feature summary.....	90
Table 5-2. Intel i386DX address and data bus signals.....	92
Table 5-3. Intel i386DX system bus control and status signals.....	93
Table 5-4. Intel i386DX transfer cycle encoding.....	93
Table 5-5. Intel i386DX device control and status signals.....	94
Table 5-6. Intel i386SX feature summary.....	98
Table 5-7. Intel i386SX address and data bus signals.....	101

## List of Tables

---

Table 5-8. Intel i386SX system control and status signals. . . . .	101
Table 5-9. Intel i386SX device control and status signals. . . . .	102
Table 5-10. Intel 80376 feature summary. . . . .	104
Table 5-11. Intel i386SL feature summary. . . . .	107

### 6 Intel 486 Microprocessors

Table 6-1. Differences between Intel 386 and 486 microprocessors. . . . .	118
Table 6-2. Intel 486 instruction set additions. . . . .	121
Table 6-3. Intel 486 burst-mode-transfer address sequence. . . . .	131
Table 6-4. Intel 486 “SL-enhanced” instructions. . . . .	134
Table 6-5. Intel i486DX feature summary. . . . .	135
Table 6-6. Intel i486DX address and data bus signals. . . . .	138
Table 6-7. Intel i486DX bus control and status signals. . . . .	139
Table 6-8. Intel i486DX cache control and status signals. . . . .	140
Table 6-9. Intel i486DX device control and status signals. . . . .	141
Table 6-10. Intel SL-enhanced 486 device control and status signals. . . . .	143
Table 6-11. Intel i486DX-50 feature summary. . . . .	147
Table 6-12. Intel i486DX-50 JTAG boundary-scan signals. . . . .	149
Table 6-13. Intel i486DX2 feature summary. . . . .	150
Table 6-14. Write-Back-enhanced IntelDX2 feature summary. . . . .	154
Table 6-15. WB-enhanced IntelDX2 revised interface signals. . . . .	156
Table 6-16. IntelDX4 feature summary. . . . .	157
Table 6-17. IntelDX4 and i486DX2 core clock multiplier factors. . . . .	158
Table 6-18. Intel i486SX feature summary. . . . .	161
Table 6-19. Intel i486SX signals. . . . .	162
Table 6-20. Intel i486SX2 feature summary. . . . .	165
Table 6-21. Intel i487SX feature summary. . . . .	166
Table 6-22. Intel i487SX interface signals. . . . .	167
Table 6-23. IntelDX2 OverDrive feature summary. . . . .	168
Table 6-24. IntelDX4 OverDrive feature summary. . . . .	171
Table 6-25. IntelSX2 OverDrive feature summary. . . . .	172
Table 6-26. Intel i486SL feature summary. . . . .	173
Table 6-27. Intel “RapidCAD” 386 feature summary. . . . .	177
Table 6-28. Intel 486 product feature comparison. . . . .	180
Table 6-29. Intel 486 product PGA pinout comparison. . . . .	181



**7 AMD 386 and 486 Microprocessors**

Table 7-1. AMD Am386SX/Am386SXL feature summary. . . . .	192
Table 7-2. AMD Am386SXLV feature summary. . . . .	195
Table 7-3. AMD Am386SXLV new instructions. . . . .	196
Table 7-4. AMD Am386SXLV interface signals. . . . .	196
Table 7-5. AMD Am386DX and Am386DXL feature summary. . . . .	198
Table 7-6. AMD Am386DXLV feature summary. . . . .	200
Table 7-7. AMD Am386SC300 “Elan” feature summary. . . . .	202
Table 7-8. AMD Am486SX and Am486SX2 feature summary. . . . .	206
Table 7-9. AMD Am486SX interface signals. . . . .	207
Table 7-10. AMD Am486SXLV feature summary. . . . .	209
Table 7-11. AMD Am486SXLV new instructions. . . . .	210
Table 7-12. AMD Am486SXLV interface signals. . . . .	211
Table 7-13. AMD Am486DX feature summary. . . . .	212
Table 7-14. AMD Am486DX2 feature summary. . . . .	213
Table 7-15. AMD Am486DXL and Am486DXLV feature summary. . . . .	214
Table 7-16. AMD Am486DX4 feature summary. . . . .	216
Table 7-17. AMD 386 and 486 product feature comparison. . . . .	220

**8 C&T 386 Microprocessors**

Table 8-1. C&T 38600DX feature summary. . . . .	230
Table 8-2. C&T 38600DX special interface signals. . . . .	230
Table 8-3. C&T 38605DX feature summary. . . . .	232
Table 8-4. C&T 38605DX special interface signals. . . . .	234

**9 Cyrix 486 Microprocessors**

Table 9-1. ALU instruction core cycle count comparison. . . . .	241
Table 9-2. Data-transfer instruction core cycle counts. . . . .	242
Table 9-3. Protected-mode control instruction core cycle counts. . . . .	242
Table 9-4. Cyrix Cx486SLC and Cx486SLC/e feature summary. . . . .	245
Table 9-5. Cyrix Cx486SLC/e instruction set additions. . . . .	249
Table 9-6. Cyrix Cx486SLC/e special interface signals. . . . .	253
Table 9-7. Cyrix Cx486SLC-V feature summary. . . . .	257
Table 9-8. Cyrix Cx486SLC2 feature summary. . . . .	258
Table 9-9. Cyrix Cx486DLC feature summary. . . . .	259

## List of Tables

---

Table 9-10. Cyrix Cx486DLC special interface signals. . . . .	260
Table 9-11. Cyrix Cx486SRx2 feature summary.. . . . .	262
Table 9-12. Cyrix Cx486DRx2 feature summary. . . . .	265
Table 9-13. Cyrix Cx486S and Cx486S2 feature summary. . . . .	267
Table 9-14. Cyrix Cx486S and Cx486S2 special interface signals. . . . .	269
Table 9-15. Cyrix Cx486S and Cx486S2 instruction set additions. . . . .	273
Table 9-16. Cyrix Cx486DX and Cx486DX2 feature summary. . . . .	277
Table 9-17. Cyrix Cx486DX/Cx486DX2 special interface signals. . . . .	280
<b>10 IBM 386 and 486 Microprocessors</b>	
Table 10-1. IBM 386SLC feature summary. . . . .	290
Table 10-2. IBM 386SLC special interface signals.. . . . .	293
Table 10-3. IBM 386SLC cache-line fill order. . . . .	295
Table 10-4. IBM 386SLC instruction set additions. . . . .	298
Table 10-5. IBM BL486SLC2 feature summary. . . . .	300
Table 10-6. IBM BL486SLC2 special interface signals. . . . .	302
Table 10-7. IBM BL486SX2/SX3 “Blue Lightning” feature summary.. . . . .	305
Table 10-8. IBM BL486SX2/SX3 special interface signals. . . . .	307
Table 10-9. IBM BL486SX2/SX3 cache-line fill order.. . . . .	307
Table 10-10. IBM BL486DX and BL486DX2 feature summary.. . . . .	308
Table 10-11. Neophyte’s IBM-to-English phrase book. . . . .	313
<b>11 Texas Instruments 486 Microprocessors</b>	
Table 11-1. TI486SLC/E and TI486SLC/E-V feature summary.. . . . .	318
Table 11-2. TI486SLC/E special interface signals. . . . .	320
Table 11-3. TI486DLC/E and TI486DLC/E-V feature summary. . . . .	322
Table 11-4. TI486DLC/E special interface signals. . . . .	324
Table 11-5. TI486SXLC and TI486SXLC2 feature summary.. . . . .	325
Table 11-6. TI486SXL and TI486SXL2 feature summary. . . . .	327
Table 11-7. TI “Rio Grande” CPU feature summary.. . . . .	329
<b>12 The Intel Pentium Family</b>	
Table 12-1. “Simple” Pentium instruction formats and operands.. . . . .	347
Table 12-2. Serialization of accesses to D-cache.. . . . .	348
Table 12-3. Pentium FPU instruction latency and throughput.. . . . .	361
Table 12-4. Pentium-specific x86 instruction set extensions. . . . .	364

Table 12-5. Intel 0.8-micron Pentium "P5" feature summary. . . . .	368
Table 12-6. Pentium address and data bus signals. . . . .	369
Table 12-7. System bus cycle control and status signals. . . . .	370
Table 12-8. Cache control and status signals. . . . .	370
Table 12-9. Miscellaneous device control and status signals. . . . .	371
Table 12-10. Performance monitoring and tracing signals. . . . .	371
Table 12-11. Pentium bus transfer cycle-type definitions. . . . .	373
Table 12-12. Pentium burst-mode transfer order. . . . .	373
Table 12-13. Special Pentium bus cycle encodings. . . . .	374
Table 12-14. Intel 0.6-micron Pentium "P54C" feature summary. . . . .	383
Table 12-15. "Top Ten Reasons" why Intel delayed announcing Pentium. . . . .	391

### 13 NexGen Microprocessors

Table 13-1. NexGen Nx586 CPU feature summary. . . . .	402
Table 13-2. NexGen FPU instruction execution times. . . . .	411
Table 13-3. NexGen announcement chronology. . . . .	414

## Volume II

### 14 Core Design and Implementation

Table 14-1. Core characteristics of x86 microprocessors. . . . .	422
Table 14-2. Conceptual steps needed to increment program counter. . . . .	431
Table 14-3. Micro-operations needed to increment program counter. . . . .	431
Table 14-4. Micro-operations for complete register-to-register ADD. . . . .	432
Table 14-5. Register-to-register ADD microroutine using more powerful core. . . . .	435
Table 14-6. Pipeline execution cycle comparison for arithmetic instructions. . . . .	469
Table 14-7. Pipeline execution cycle comparison for various control instructions. . . . .	470
Table 14-8. Typical weighted-average cycles per instruction. . . . .	472
Table 14-9. Multiply-intensive weighted-average cycles per instruction. . . . .	473
Table 14-10. Cache characteristics of x86 microprocessors. . . . .	476
Table 14-11. External cache control signal definitions. . . . .	479

### 15 Manufacturing Costs

Table 15-1. Estimated 386/486 wafer cost and defect density. . . . .	526
Table 15-2. Die area and yield estimates for 386/486 processors. . . . .	527
Table 15-3. Cost estimates for various package types. . . . .	529
Table 15-4. Cost estimate summary for 386/486 processors. . . . .	531

## List of Tables

---

Table 15-5. Intel die yield and cost vs competing RISC CPUs. . . . .	533
Table 15-6. Intel cost summaries vs competing RISC CPUs. . . . .	534
Table 15-7. Estimated gross margins for selected x86 and RISC CPUs. . . . .	535
<b>16 Legal Issues</b>	
Table 16-1. Summary of key x86-related litigation. . . . .	552
<b>19 Pricing Structures</b>	
Table 19-1. Intel 486 price and availability. . . . .	631
Table 19-2. IntelDX4 and Pentium price and availability. . . . .	631
Table 19-3. Intel upgrade processor price and availability. . . . .	631
Table 19-4. AMD 386 and 486 price and availability. . . . .	632
Table 19-5. Cyrix 486 price and availability. . . . .	633
Table 19-6. Texas Instruments 486 price and availability. . . . .	633
Table 19-7. IBM 486 price and availability. . . . .	634
Table 19-8. NexGen 586 price and availability. . . . .	634
<b>20 Performance Measurement and Analysis</b>	
Table 20-1. Full SPEC92 benchmark detail for selected Intel processors. . . . .	649
Table 20-2. Components of SPECint92 benchmark suite. . . . .	651
Table 20-3. Components of SPECfp92 benchmark suite. . . . .	652
Table 20-4. Components of Intel's iCOMP performance metric, version 1.0 . . . . .	659
Table 20-5. UNIX CPU benchmark results for Intel's x86 microprocessors. . . . .	661
Table 20-6. DOS CPU benchmark results for Intel's x86 microprocessors . . . . .	662
Table 20-7. CPU-intensive benchmark results for non-Intel x86 processors. . . . .	663
Table 20-8. Components of SYSmark92 benchmark suite. . . . .	668
Table 20-9. Components of SYSmark93 benchmark suite. . . . .	669
Table 20-10. Components of Winstone94 benchmark suite. . . . .	672
Table 20-11. Cyrix's Cx486DX2 benchmark comparisons. . . . .	675
Table 20-12. Performance for Intel's write-through and write-back i486DX2-66. . . . .	676
<b>Appendix E: Floating-Point Bug Strikes Pentium</b>	
Table E-1. Binary representation of several failing divisors. . . . .	E-5

# Foreword

---

Since I started the *Microprocessor Report* newsletter in 1987, the microprocessor industry has changed dramatically. The emergence of five major RISC architectures and a proliferation of x86 designs, fueled by the explosive growth of the PC industry, have made the microprocessor marketplace vastly more complex.

*Microprocessor Report* is well known for its up-to-the-minute coverage of new microprocessors. In the course of doing this work, we collect far more information than we can ever include in the newsletter. We have frequently heard from our readers that a more comprehensive, all-in-one reference would be invaluable. Based on this input, we set out two years ago to produce a series of in-depth reports, called the MicroDesign Resources Technical Library.

Our first Technical Library report, *New DRAM Technologies*, was created by Steven Przybylski and published in April 1994. Now, after several man-years of effort, we have completed the first two microprocessor volumes in the series: *The Complete x86*, by John Wharton, and *RISC on the Desktop*, by Linley Gwennap. John and Linley have invested enormous amounts of time to make these reports the most comprehensive resources available. Many others have made significant editorial contributions, including Brian Case, Rich Belgard, Nick Tredennick, Ivy Lui, and myself.

These reports have been a mammoth undertaking, and the results speak for themselves. I believe that they are destined to become the "bibles" of the microprocessor industry. Not only do

they describe a wide range of products in great depth, but they put the technology in the context of the marketplace.

We will be updating and enhancing these reports periodically, and I would appreciate your feedback on them. The best way to contact me is by email ([m Slater@mdr.ziff.com](mailto:m Slater@mdr.ziff.com)); you can also reach me by phone at 707.824.4004. I trust you will find the reports valuable, and I look forward to your feedback.

*Michael Slater  
Sebastopol, California  
December, 1994*

# Executive Summary



---

It is the best of times. It is the worst of times. It is a time of rapidly expanding options as new x86 devices burst on the scene. It is a time of consolidation as old products are displaced or withdrawn from the market. Never before have PC designers been faced with so many attractive alternatives, and never before has the processor selection process been more tough.

## **Why this Report is Needed**

It used to be, if you were building a 386-class PC, your choices were simple. Midrange desktop systems chose an i386DX device for a reasonable mix of price and performance. Low-cost or portable PCs used an i386SX to reduce expense and part count. Vendors of high-end desktops and "tower" process-servers selected the i486DX for absolute maximum performance. You could pick any supplier you wanted, Henry Ford might have said, as long as it was Intel. The hardest decision a designer faced was the choice of what CPU frequency best matched the target system price and performance.

Times have changed. In just three years an explosion of new x86 alternatives has rocked the market. The number of functionally different 32-bit x86-compatible microprocessors has grown from three to three dozen, not counting assorted frequency and voltage options. At least six vendors now vie with Intel for x86-class sockets. At the 486 level, processors are available in more than 20 functionally different pinouts, with or without on-chip floating-point units, with on-chip caches of various configurations ranging from 1K to 16K bytes, and with at least six different clocking schemes. The earlier "standard" devices have been supplemented by higher-integration, lower-power, and lower-voltage variations. Furthermore, Intel has begun flooding the

market with its long-stalled and eagerly-awaited Pentium, the first CISC microprocessor able to execute multiple instructions per clock cycle, and is just now beginning to drop hints about what's likely to come next.

Alas, flexibility breeds confusion. Deciding which device to use for a given application has become a daunting task, requiring the knowledge of everything from supplier track records and nuances of various architectural extensions to the merits of internal pipeline timing and competing cache coherency protocols. This special report, *The Complete x86—The Definitive Guide to 386, 486, and Pentium-Class Microprocessors*, will help clarify and simplify that task.



# Preface

---

## How this Report is Organized

Since the process of evaluating and selecting a microprocessor is itself a multifaceted task, this report is divided into six major parts.

- **Part I: Preliminaries** contains background information about the 386 microprocessor family, including a summary of its development history and a brief review of certain aspects of the architecture that distinguish alternative implementations.
- **Part II: The Players** introduces the major suppliers competing for shares of the x86 market, with a brief business profile of each.
- **Part III: The Products** discusses the forty-something separate 386- and 486-class microprocessor products now on the market, including a brief review of the unique features, benefits, resources, pinouts, bus interfaces, and internal implementations of each.
- **Part IV: Pentium-Class Processors** contains an in-depth discussion of the technical merits, implementation details, system-design issues, and business strategies surrounding the Pentium microprocessor, the first member of the x86 family to deliver superscalar execution, as well as derivative designs and competing products from NexGen.
- **Part V: Perspective** compares competing product implementations on technical issues such as core microarchitecture, pipeline design, cache efficiency, manufacturability, and software compatibility. This side-by-side comparison

gives you the technical meat you need to understand the key distinctions within an otherwise confusing morass of similar looking and sounding devices.

- **Part VI: Price and Performance** contains price and availability tables for each vendor's x86 product line, discusses the challenging field of processor benchmarks, and summarizes the relative performance of various 386- and 486-class microprocessors running an assortment of industry-standard benchmark programs.
- The **Appendices** contain vendor contact data and summaries of the technical specifications, performance data, and technology patents presented elsewhere in the report.

At the beginning of each part is a brief description of the chapters it contains and the subjects they cover. At the end of each chapter is a personal commentary to put the material in perspective. Each chapter also includes a list of reference manuals, articles from past issues of *Microprocessor Report*, and other sources that provide further information on the chapter topics.

### **Terminology**

Much of the confusion that arises in surveying an industry comes from the fact that different vendors often use different terminology, symbols, or conventions to represent analogous concepts. Some vendors refer to "copy-back" caches, while others use "write-back" to mean the same thing. Likewise with "cache coherency" vs "cache consistency" or "bus snooping" vs "inquiry cycles." Intel and AMD follow one convention for denoting hexadecimal numbers, while IBM follows another. (It seems that whenever the industry is on the verge of reaching a standard on some new terminology, IBM finds its own aberrant word for the same thing: ROS for ROM, RWM for RAM, etc. The entity that every other company in the world calls a "motherboard" is, in IBMspeak, designated a "planar.") To avoid this confusion, we try to be consistent in our terminology, the better to focus on differences in concepts rather than differences in etymology.

Littered throughout this report are various vendors' claimed and registered trademarks. Rather than place a trademark symbol at every occurrence, we hereby state that we are using product names only in an editorial fashion with no intention of infringement of the trademark. We have, however, tried to avoid using any trademarked terms except in reference to a specific vendor's products. For example, we use the letters "i386SX"

only in reference to a specific Intel product and “Am486DX” only in the context of a specific AMD device.

Vendor-specific prefixes are omitted (as in “486SX”) when describing generic capabilities of a device produced by multiple vendors. Except where otherwise noted, “x86” or “386” with no suffix alludes to fundamental architectural features or capabilities shared by the entire 32-bit x86 product line, including various flavors of the 386, 486, and Pentium families.

## Acknowledgments

The preparation of any report of this size and scope is necessarily a group effort, especially in a field this dynamic. No one person can master all the subtle nuances of competing processor architectures, core implementation technologies, and pipeline timing, much less competitive business strategies and intellectual property law. Just keeping abreast of the rapid changes in each field is a full-time job.

Fortunately, the contributing editors and staff of MicroDesign Resources and *Microprocessor Report* have collective expertise across a broad cross-section of microprocessor and technology-related topics. I would like to acknowledge those who prepared many of the more specialized chapters of this report, and express my appreciation for their help.

Michael Slater provided editorial guidance and overview for the entire report. Michael Feibus assembled the business profiles of microprocessor vendors that appear in **Chapter 4**. Brian Case wrote the better part of **Chapter 12**, detailing the Pentium microprocessor, as well as the technical analyses of various CPU implementations contained in **Chapter 14**. Linley Gwennap created the manufacturing cost model and analyses that appear in **Chapter 15**. Michael Slater and Richard Belgard prepared the survey of legal and intellectual property issues contained in **Chapter 16** and **Appendix D**.

Michael Slater and Nick Tredennick were responsible for the performance analysis topics covered in **Chapter 20**, from collecting data and preparing the comparison tables to formatting the graphs and drafting the text. Linley Gwennap also prepared the Pentium floating-point discussion in **Appendix E**. I was responsible for (and shoulder the blame for any inaccuracies in) the remaining chapters, including the x86 overview material, most of the product descriptions, and miscellaneous topics covered in the other chapters of this report.

I am also especially grateful to Deena Waters and Judi Clark for contributing their artistic and organizational skills to this project, for their many helpful review comments, and for coaching me through to completion; to Ellen Clements, for her diligence in reviewing and copy editing early drafts; to Brian Case and Marianne Mueller for ongoing moral support; and to Ron Liskey, Melanie Sanborn, Brian Hunziker, and the entire gang at MicroDesign Resources for collecting raw data, coordinating vendor contacts, and taking care of the myriad of other production details.

Thanks are also due to all who took the time to review drafts of this report in all its many incarnations, especially Michael Slater, Brian Case, Martin Reynolds, Bernard Peuto, and John Novitsky, and to the vendor representatives who provided the background information, data sheets, and other technical support, and who reviewed various drafts for technical accuracy.

### **Feedback and Updates**

Much of the information in this report is extremely time sensitive; new microprocessor products and design variations are being announced monthly, and vendor pricing seems to be in a state of continuous flux. Supplements and updates to this report will be published periodically in an effort to keep the most time-sensitive information up to date.

My goal is for this report and its updates to be as complete and accurate as possible. If you spot any errors, discrepancies, or areas you feel need further clarification, or if you have any background information, industry anecdotes, or unsubstantiated rumors you'd like to see included in future editions—anon-ymously or with attribution—please feel free to drop me a line.

*John H. Wharton  
Palo Alto, California  
415.856.8051  
jwharton@mdr.ziff.com  
December, 1994*

# About the Authors

---

One of the industry's top analysts and an inveterate Intel watcher, John Wharton has over twenty years' experience in microcomputer design and implementation. As an applications engineer at Intel, he was responsible for defining the architecture of the 8051 family, Intel's highest volume processor to date. Since 1987 he has been a contributing editor and columnist for *Microprocessor Report*, a lecturer in Computer Science at Stanford University, and a consultant in computer system design and architecture.

Michael Slater is president of MicroDesign Resources, editorial director of *Microprocessor Report*, and organizer of the Microprocessor Forum. He is author of the widely used text, *Microprocessor-Based Design*, and began his career as an R&D engineer at HP.

Brian Case is a respected authority on microprocessor architecture and implementation and previously worked as an architect and system designer at AMD, Apple, and Sun. At AMD, he helped define the architecture and structure of the 29000, one of the industry's highest-volume RISC CPUs. He has been a contributing editor for *Microprocessor Report* since 1988.

Dr. Nick Tredennick has established a reputation as one of the industry's leading advocates of CISC architectures. Among other accomplishments, he developed the original microcode for the Motorola 68000 and the IBM Micro/370. He has been a contributing editor for *Microprocessor Report* since 1990, and is creator of the "Tredennick Awards," presented each year at the Microprocessor Forum.



# About the Publisher

---

MicroDesign Resources is an information services company focusing on microprocessors and related technologies. Its principal products are the *Microprocessor Report* newsletter, the Microprocessor Forum conference, and the MicroDesign Resources Technical Library. The MicroDesign Resources staff also provides technical seminars on high-performance microprocessors, which it offers on both a public and an in-plant basis, and provides strategic consulting to help vendors evaluate and refine product definitions, plan introduction strategies, and understand the competitive microprocessor landscape.

In 1987 the company began publishing *Microprocessor Report*, the leading technical publication for the microprocessor industry. Published every three weeks, *Microprocessor Report* is exclusively subscriber supported and is dedicated to providing unbiased, in-depth, and critical analysis of new, high-performance microprocessor developments. In addition to covering the chips themselves, the newsletter covers the microprocessor implications of emerging platforms, emerging personal computer technologies, workstation designs, mobile computing devices, embedded processors, DSP technology, and intellectual property issues. In mid-1992, MicroDesign Resources was acquired by Ziff Communications and is now a division of Computer Intelligence InfoCorp, operating as a wholly independent subsidiary.

*Microprocessor Report* is written by Editorial Director Michael Slater, who also serves as president of the company, Editor-in-Chief Linley Gwennap, and Senior Analyst Jim Turley, and is reviewed by the 13-member editorial board. All members of the

editorial staff are distinguished by extensive technical backgrounds. Contact *mpr-info@mdr.ziff.com* for more information, or call 707.824.4001.

MicroDesign Resources also sponsors the Microprocessor Forum. An annual conference now in its eighth year, the Forum features announcements of many of the coming year's most significant new microprocessors. Presentations are technically oriented and are generally given by the microprocessor's chief architect. The conference program is supplemented by in-depth technical seminars presented by the *Microprocessor Report* editorial staff. The 1994 Forum was a sellout with more than 800 attendees.

MicroDesign Resources recently introduced the MicroDesign Resources Technical Library, a series of technical reports that provide comprehensive surveys and in-depth analyses of specific product areas. There are MicroDesign Resources reports on several categories of microprocessors; titles include *RISC on the Desktop: A Comprehensive Analysis of RISC Microprocessors for PCs, Workstations, and Servers*; *The Complete x86: The Definitive Guide to 386, 486, and Pentium-Class Microprocessors*; *New DRAM Technologies: A Comprehensive Analysis of the New Architectures*; and *High-Performance Embedded Processors: The Comprehensive Directory and Selection Guide* (to be published Spring, 1995).



# Part I: Preliminaries

Before attempting to study the various flavors of 386 and 486 microprocessors sold by Intel and its competitors, it's helpful to have at least some preliminary background on what's happening in the 386/486 marketplace, how the x86 product line got where it is, and some of the difficulties posed by the x86 architecture on device designers in creating newer and more efficient implementations.

**Part I** of this report contains a quick review of this information. It consists of three chapters:

**Chapter 1: The x86 Business Climate**

**Chapter 2: x86 Family Heritage**

**Chapter 3: The x86 Microprocessor Architecture**



# 1

## The x86 Business Climate

The past few years have redefined the face of the 386/486 microprocessor market. Intel has pursued a program of wholesale proliferation of 386- and 486-family microprocessors, rolling out more than a score of different products with different levels of integer performance, FPU capability, clocking regimes, and pinouts, as listed in Table 1-1.

Vendor	Device	Features
Intel	i386DX	First 32-bit member of the x86 microprocessor family
	i386SX	i386DX core with 16-bit bus interface in a lower-cost package
	80376	De-DOSed i386SX targeted for embedded applications
	i386SL	High-integration/low-power 386-derivative for notebook applications (deceased)
	i486DX	Pipelined implementation of 386 architecture with on-chip FPU and 8KB cache (note 1)
	i486DX-50	50-MHz i486DX redesigned for 0.8 $\mu$ three-layer-metal process (note 1)
	i486DX2	i486DX with on-chip clock doubler, marketed for OEM applications (note 1)
	WB-enhanced IntelDX2	i486DX2 with on-chip cache enhanced to support copy-back operation
	IntelDX4	Clock-tripled, lower-power 3.3V i486DX with 16KB on-chip cache
	i486SX	i486DX with FPU removed and cosmetic pinout changes (note 1)
	i486SX2	Clock-doubled version of the i486SX
	i487SX	i486SX/SX2 adjunct with rehabilitated FPU and yet another unique pinout
	IntelSX2 OverDrive	Upgrade processor based on i486SX2 core, packaged for the retail masses
	IntelDX2 OverDrive	Upgrade processor based on i486DX2 core, packaged for the retail masses
	IntelDX4 OverDrive	Upgrade processor based on IntelDX4 core, packaged for the retail masses
	i486SXLP, i486DXLP	i486SX and i486DX with direct 2x clock input and reduced Fmin (special-order only)
	i486SL	Static, high-integration i486DX for notebook applications (moribund)
RapidCAD 386	i386DX/i387DX replacement chip set with 486-class FPU performance (deceased)	

Table 1-1. Intel 386 and 486 product line summary.

(note 1: "SL-enhanced" variation has replaced original design.)

Since 1991 at least five other vendors have leapt into (and in one case retreated from) the 386/486 arena. All told, more than 30 competing 386- and 486-class products have been introduced or preannounced, including those listed in Table 1-2.

Vendor	Device	Features
AMD	Am386SX	AMD version of i386SX with Intel-compatible specs and pinout
	Am386SXL	Low-power static version of Am386SX (deceased)
	Am386SXLV	Low-voltage Am386SXL with SMM extensions (deceased)
	Am386DX	AMD version of i386DX with Intel-compatible specs and pinout
	Am386DXL	Low-power static version of Am386DX (deceased)
	Am386DXLV	Low-voltage Am386DXL with SMM extensions (deceased)
	Am386SC300 ("Elan")	Highly-integrated single-chip CPU and PC chip-set for ultrasmall systems
	Am486SX	AMD version of i486SX with AMD $\mu$ code, compatible specs and pinout
	Am486SXLV	Low-voltage, static Am486SX with SMM extensions (deceased)
	Am486SX2	Clock-doubled i486SX with Intel $\mu$ code, compatible specs and pinout
	Am486DX	AMD version of i486DX with Intel $\mu$ code, compatible specs and pinout
	Am486DXL	Low-power, static Am486DX with SMM extensions (deceased)
	Am486DXLV	Low-voltage, low-power, static Am486DX with SMM extensions (deceased)
	Am486DX2	AMD version of i486DX2 with Intel $\mu$ code, compatible specs and pinout
	Am486DX4	Bond-out option of Am486DX2 die with clock-tripling circuitry, still with 8KB cache
C&T	38600DX	C&T optimized, pin-compatible version of i386DX (deceased)
	38605DX	Pinout-extended version of 38605DX + 512B I-cache (deceased)
Cyrix	Cx486SLC/e	486-class static integer core with 1KB cache, i386SX pinout, and SMM
	Cx486SLC/e-V	Low-voltage version of Cx486SLC/e
	Cx486SLC, SLC2-V	Clock-doubled version of Cx486SLC/e and 3-V variation
	Cx486DLC	i386DX pinout version of Cx486DLC (deceased)
	Cx486SRx <sup>2</sup>	Clock-doubled Cyrix 486 core in 386SX pinout for end-user upgrades
	Cx486DRx <sup>2</sup>	Clock-doubled Cyrix 486 core in 386DX pinout for end-user upgrades
	Cx486S, Cx486S2	Conventional and clock-doubled Cyrix cores with 2KB cache in i486SX pinout
	Cx486DX, Cx486DX2	Conventional and clock-doubled cores with FPU, 8KB cache, and i486DX pinout
IBM	386SLC	Enhanced 486SX-like integer core with 8KB cache in 386SX pinout
	BL486SLC2	Clock-doubled 386SLC core with 16KB cache in 386SX pinout
	BL486SX2/SX3	Clock-tripled BL486SLC2 with 16KB cache in 386DX pinout
	BL486DX2	IBM second-source equivalent of Cx486DX2
Texas Instruments	TI486SLC/E	TI version of Cx486SLC/e
	TI486SLC/E-V	TI version of Cx486SLC/e-V
	TI486DLC/E	TI version of Cx486DLC, enhanced with Cyrix-style SMM capabilities
	TI486SXLC	TI derivative of Cx486SLC core with 8KB cache and enhanced 386SX pinouts
	TI486SXLC2	Clock-doubled version of the TI486SXLC
	TI486SXL	TI variant of the TI486SXLC with burst-mode-challenged 486SX-style bus interface
	TI486SXL2	Clock-doubled version of the TI486SXL
	"Rio Grande"	Highly integrated integer CPU plus system logic derived from Cyrix core (stillborn)

Table 1-2. Alternate vendor 386 and 486 product line summary.

Intel's first 386 competitor, AMD, managed not only to establish itself as a viable supplier but quickly took the majority of the 386 business from Intel before beginning to compete head-on with Intel in the 486 market. Chips and Technologies came and went as a 386 supplier, unable to face the rigors of an intensely competitive marketplace.

Cyrix burst on the scene with its Cx486SLC and Cx486DLC, quickly gaining many second-tier design wins; it has since broadened its product line to include products that match (and in some ways exceed) Intel product specs. In 1993, IBM began competing in the x86 component business, albeit indirectly. In 1994, the gloves came off, as IBM began selling Cyrix-designed chips directly to OEMs. Texas Instruments has also begun marketing Cyrix-designed processors under its own name, and has recently begun spinning off (and in some cases withdrawing) devices with larger caches, increased system integration, and revised bus interfaces, all based on the original Cyrix core.

Not to be outdone, Intel responded with a proliferation of Pentium-family microprocessors with different performance levels, supply voltages, and pinouts, and has recently begun dropping hints about what lies on beyond Pentium. Even NexGen, that erstwhile proponent of system architectures yet-to-come, has begun shipping its Pentium competitor, the Nx586, and plans to provide floating-point support in 1995 (see Table 1-1).

Cyrix is starting to reveal more detailed plans for the "M1" product it plans to introduce in 1995, and AMD in 4Q94 began revealing plans for its upcoming "K86" family.

Vendor	Device	Features
Intel	0.8 $\mu$ Pentium ("P5")	Superscalar CPU with separate 8KB I and D caches, 64-bit bus, branch prediction, fast FPU, and on-chip functional verification logic
	0.6 $\mu$ Pentium ("P54C")	Lower-cost Pentium redesigned for 3.3-V operation and reduced power requirements, with on-chip support for multiprocessor interrupt logic
	Pentium Overdrive ("P24T")	Upcoming Pentium-based upgrade chip with 32-bit bus and i486DX2-like pinout
	"P6"	Future successor to Pentium family, likely with Pentium-like macroarchitecture
	"P7"	Long-range future Intel CPU, possibly with new, VLIW influenced architecture.
NexGen	Nx586	Highly optimized superscalar Pentium-class integer unit with 32KB cache
	"Nx586 with FPU"	Upcoming Nx586-based two-chip module with on-board pipelined FPU
Cyrix	"M1"	Upcoming Pentium-class machine with improved, superpipelined, superscalar execution
AMD	"K5"	Upcoming superscalar design intended to exceed Pentium performance
	"K6"	Next-generation AMD processor

Table 1-3. Pentium- and post-Pentium product line summary.

This shows a remarkable contrast to the period up through early 1991, when Intel was the sole supplier of 386 and 486 processors. Systems designers have more choices than ever before, filling nearly every conceivable price/performance niche, and competition is driving prices ever lower. Prices for some midrange 386 and 486SX chips have fallen as much as 25% per quarter, fueling sharp drops in system prices.

Through all the bedlam, Intel has kept its profits high. Though hit by huge 386 market-share losses and steep price cuts in the low-end and midrange markets, Intel has succeeded in moving a large part of the market to its high-end—and high-margin—i486DX2, IntelDX4, and Pentium-class products.

## 1.1 The Explosion of Design Alternatives

Life used to be simple. More than five years after the 1985 introduction of the Intel 80386 (now called the i386DX), there were still only four products competing for the 32-bit 386/486 market (see Figure 1-1). You could choose any vendor you wanted, Henry Ford might have said, as long as it was Intel.

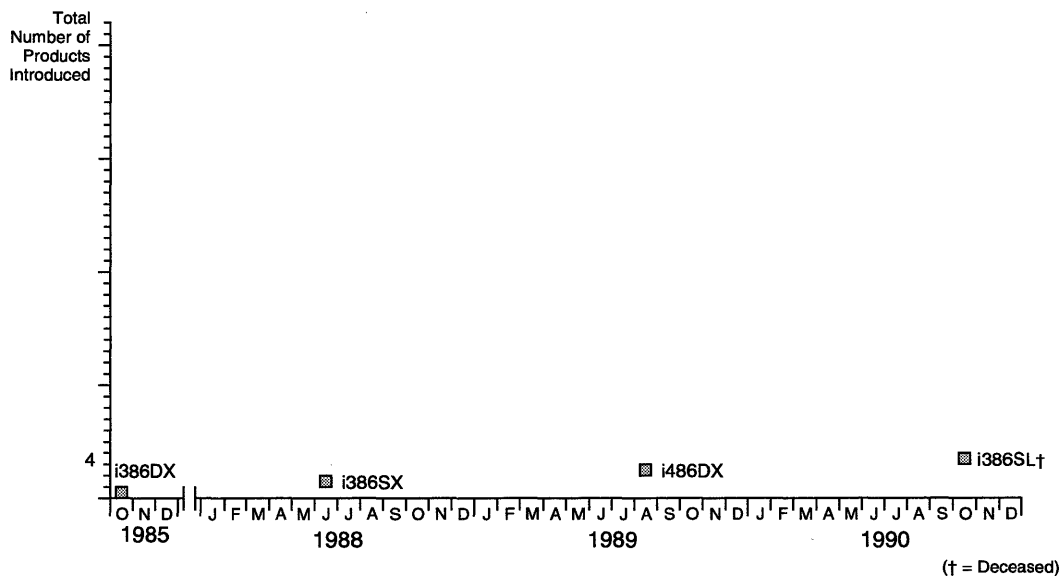


Figure 1-1. x86 processor introductions: the early years.

Times have changed. Over the last four years an explosion of new x86 alternatives has rocked the market. More than 60 different 386- and 486-compatible products exist, including functionally different devices as well as functionally equivalent parts from different vendors (see Figure 1-2). (Note that this total does *not* include assorted frequency, voltage, and packaging options for otherwise similar products.) In addition, Intel is continuing aggressively to ramp-up production volumes of its long-stalled Pentium, and in 1994 began proliferating the family by introducing the “P54C,” the first of an expected salvo of new parts derived from the Pentium core.

As the supply of x86 processors rose, the conventional wisdom was that prices would fall. With more vendors, more products, and more production volume coming on line steadily since 1992, the law of supply and demand said Intel would be in for some hard times indeed.

And yet, the law of supply and demand appears to have been overturned. Except for some steep price declines in mid-1991—when AMD first entered the 386 market—prices on Intel’s mainstream products held surprisingly steady throughout 1993 and well into 1994. It wasn’t until 2Q94 that the prices for certain x86 products began to collapse. This can be seen in

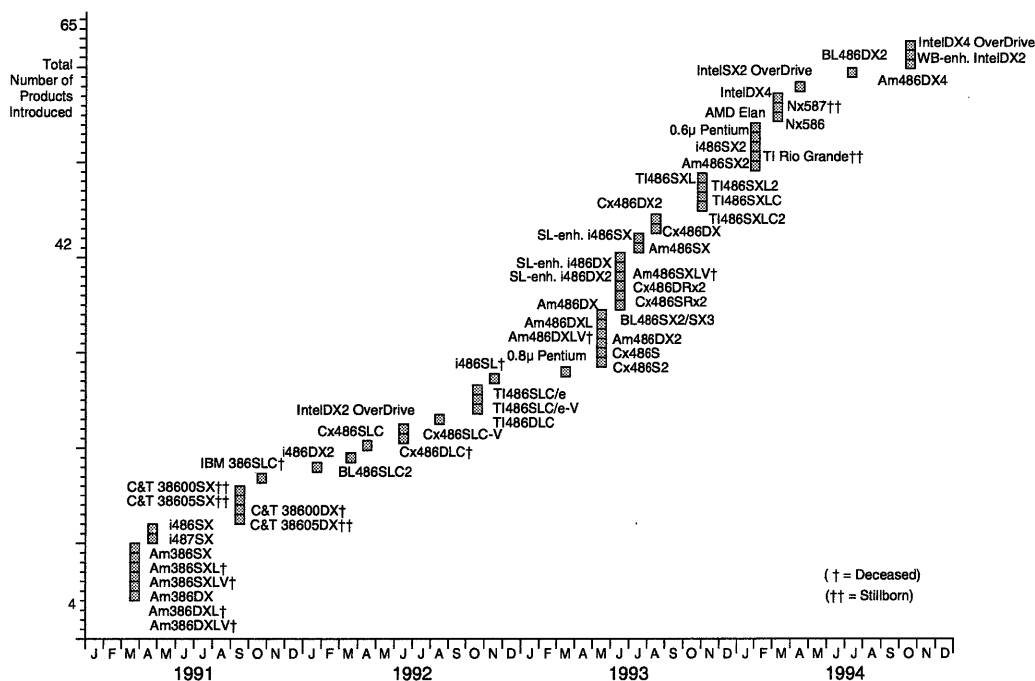


Figure 1-2. x86 processor introductions: now.

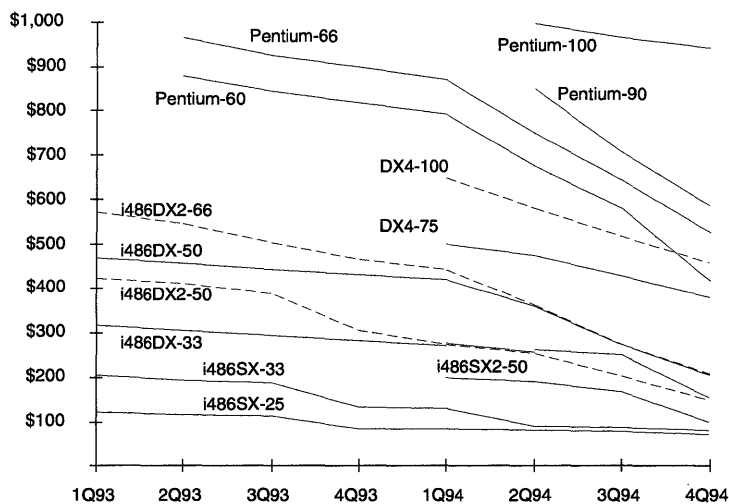


Figure 1-3. Intel 386/486 product price trends.

Figure 1-3, which plots Intel's published 486 and Pentium prices (in 1000-unit quantities) throughout 1993 and 1994.

## 1.2 Forestalling the Blood Bath

What gives? What delayed the blood bath so many analysts had predicted? Where was the flaw in conventional wisdom?

The answer appears to be threefold. First, there was indeed a blood bath in 1993, but it took place in the systems arena, not at the chip level. Prices did plummet for 386 and 486 portable and desktop systems. The 80286 became quite profoundly dead, creating new demand for 386- and 486-type devices, which neatly matched the industry-wide increase in supply.

More important, though, is the fact that the conventional wisdom's assumptions were wrong. The dozens of new products weren't really competing for the same x86 market. Instead, dozens of products were competing for *dozens* of x86 markets.

## 1.3 Dimensions of Differentiation

Back when only four x86 processors existed, if you were building a 386-class PC, your choices were straightforward. Midrange desktop systems chose an i386DX device for a reason-



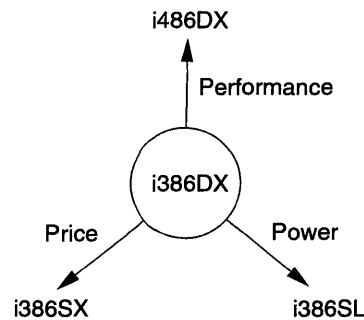


Figure 1-4. x86 product differentiation: the early years.

able mix of price and performance. High-end desktops or tower-style servers selected the i486DX to achieve absolute maximum performance—and cost be damned! Low-cost, high-volume PCs used an i386SX to reduce expense and part count. Portable PCs generally selected an i386SL to reduce power requirements and board area (see Figure 1-4). The toughest decision a designer faced was which CPU frequency to select for a particular target performance level and price.

With the explosion of new devices, though, the situation changed. Instead of just price, performance, and power-optimization issues, designers face not just more options, but a plethora of whole new dimensions of product options (see Figure 1-5).

The original “standard” 386/486 devices have been joined by higher- and lower-integration, higher- and lower-speed, and lower-power and -voltage variations. At the 486 level alone, pro-

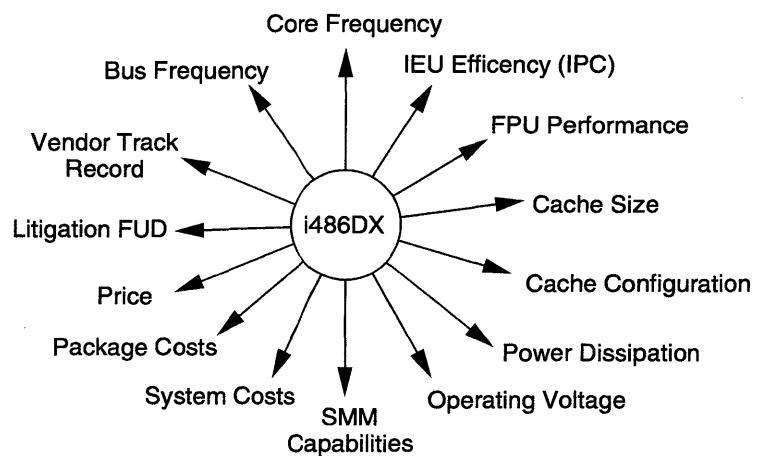


Figure 1-5. x86 product differentiation: now.

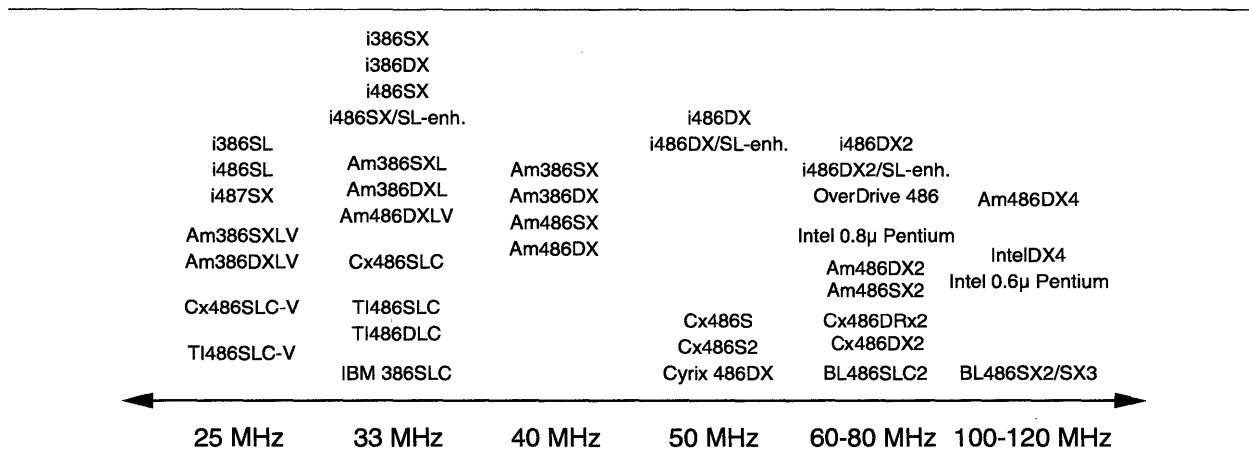


Figure 1-6. x86 core frequency design options.

processors are now available with at least eight different pinouts, with or without on-chip floating-point units, with on-chip caches ranging from 1K to 16K bytes, different set associativity, write-through or copy-back configurations, and at least four different clocking schemes.

As shown in Figure 1-6, devices are now available with maximum core clock rates that range from 25 MHz to 100 MHz. The technologies used within the processor execution pipelines range from simple, sequential, microcoded execution to state-of-the-art superscalar and superpipelined designs (see Figure 1-7).

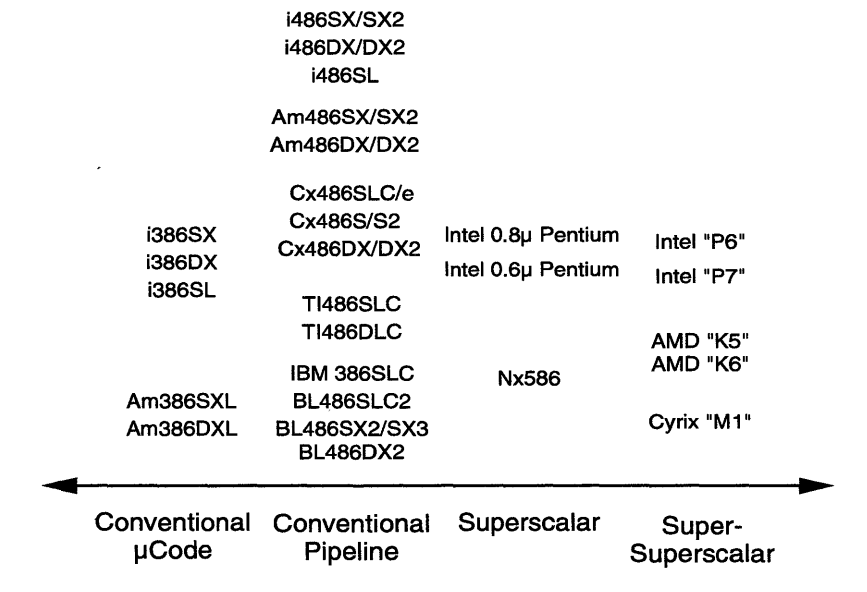


Figure 1-7. x86 core microarchitecture design options.

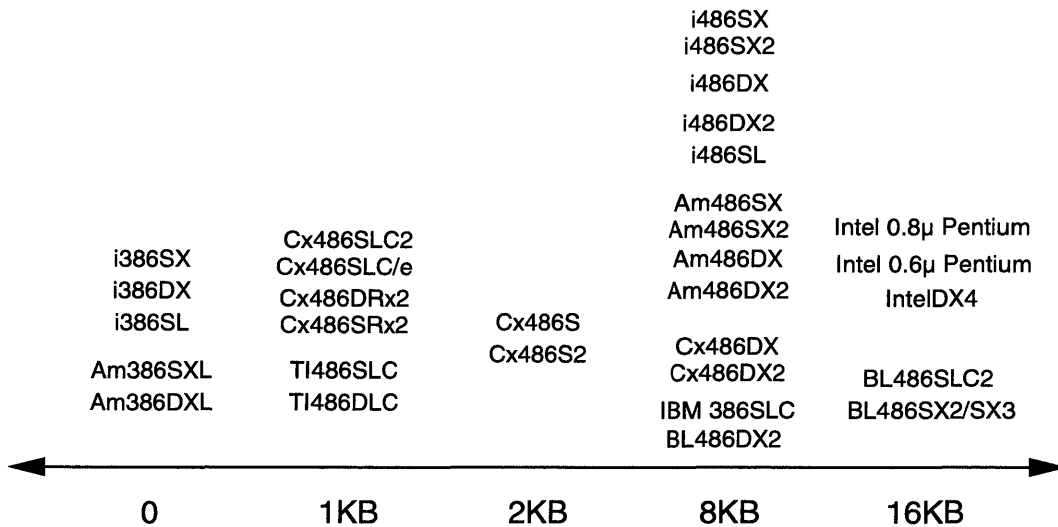


Figure 1-8. x86 cache size design options.

Products have on-chip caches that vary in size from zero to 16K bytes, including instruction-only caches, those that combine instructions and data, and split I/D organizations, either direct-mapped, two-way, or four-way set associative (Figure 1-8). Several different approaches have been developed for reducing power at the low end (Figure 1-9)—and for dissipating more heat at the high end.

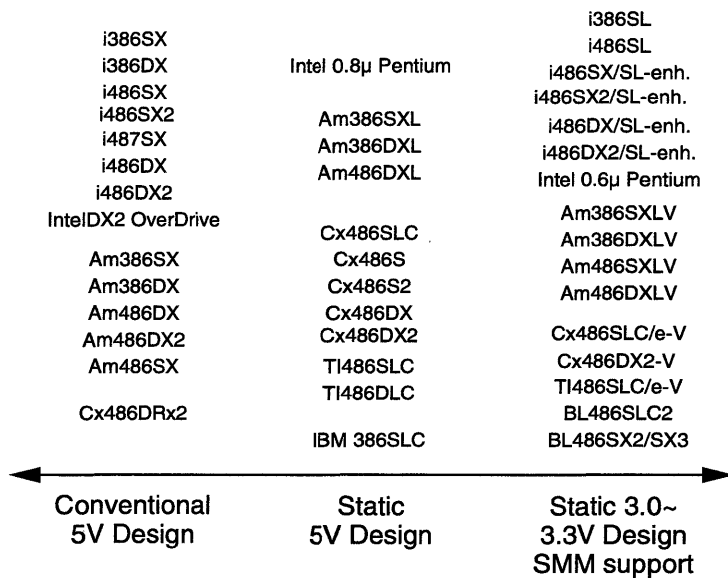


Figure 1-9. x86 power strategy design options.

And as a rule, each new product has staked out a unique combination of characteristics along the various price/performance/capability continua. Pentium, with its dual pipelines, large caches, wide buses, and high-frequency clock, initially required a 5-V supply, and burned wattage galore. And at least until recently, the devices with the most aggressive power-down modes often had performance that was really lame.

With so many dimensions along which to distinguish themselves, products faced surprisingly little direct competition. Through late '93, there were never more than two vendors offering interchangeable pinouts and comparable combinations of features. AMD was an unlicensed second source of certain Intel parts, while TI second-sourced the considerably more capable Cyrix 386-compatible pinout devices. Chips and Technologies and IBM pursued their own unique visions, while NexGen continued marching to the beat of its own strategic drummer.

The microprocessor industry can tolerate second sourcing—indeed, until the 386 came along system vendors often refused to design in a new processor until it had a licensed second source. It takes three or more vendors, all jockeying for a bigger slice of the pie, for things to get deliciously nasty.

With two sources for an interchangeable product, the industry leader may typically capture 70% to 90% of the unit shipments. Meanwhile, the second-source vendor can often garner 10% to 30%—nothing spectacular, but enough to pay its bills. (Indeed, when a second source first comes on line with a new product, even a 10% market share may be enough to saturate its production lines.) Under such conditions, neither the primary nor secondary vendor has much motivation to start a price war.

The situation changes when three or more vendors all jockey for a fixed market pie. In that case, it may happen that no one company controls even half the market, and the smallest of the vendors may have a 5% share or less. Under *this* scenario, it makes eminent sense for competitors to start undercutting each other's pricing structures: the runt of the litter may feel its very survival depends on buying market share through rock-bottom pricing, and even the company with the fattest slice of the pie may still see considerable room for market-share growth.

Moreover, it's the newest, highest-performance, highest-profit-margin segment of the market that inherently draws the most attention from alternate-sources, and that's precisely the segment of the market most susceptible to severe price reductions.

But throughout 1993 and well into 1994, industry demand for 486-class devices continued to outstrip supply. When competition for a particular socket became too intense, one or more of the competing vendors gracefully withdrew, refocusing their finite production capacity on more lucrative markets.

## 1.4 Commentary

Alas, good things (from the chip vendors' perspective) never last. In 1994 x86 prices did indeed start to fall. Pundits' and analysts' predictions of CPU price wars did prove to be correct, if somewhat premature. Indeed, during 3Q94 Intel's official price for the i486DX-33 fell by 40%, and at times the street price of some products has fallen by as much as 30% in a single month. (Extrapolate *that* trend line and in 100 days the parts would be free!) And the reasons for these price declines relate to the same issues that supported chip price in the past.

First, while the early 486-clone devices from AMD, Cyrix, TI, C&T, and IBM included 386SX, 386DX, and various unique-pinout devices, recent announcements have focused primarily on products that are compatible with the standard 486 pinout and bus structure. Moreover, these products now offer a wide range of frequencies, clock-multiplier options (including 2x, 3x, and fractional values), different amounts of on-chip cache, write-through or copy-back modes, and a range of different power-optimization techniques. Since all such products have essentially the same pinout, there's more opportunity for competition for the same socket solely on the issue of price.

Second, production limitations are becoming a thing of the past. AMD is converting its sub-micron development and prototyping facility to full-scale production, and has a new \$1 billion facility in the works. Cyrix actively courted new foundries to supplant its original production partners, and in 1994 secured a significant chunk of IBM's formidable excess production capacity. (IBM is also building the NexGen processor, lending greater credence to that design.)

And Intel is well into a program to single-handedly bring \$5 billion worth of new fab capacity on-line by the middle of the decade, primarily in 0.6-micron (and smaller) 8"-wafer fab plants. Intel's investments have produced a triple-whammy to the industry. The smaller-geometry process both lowers the cost and radically increases the supply of Intel's newest, highest-

performance products, such as the write-back enhanced IntelDX2, the clock-tripled IntelDX4, and the 0.6 $\mu$  Pentium "P54C." Moreover, the migration of existing high-end products to newer facilities frees up existing 0.8-micron capacity, thus ending the production limitations that previously faced lower-end 486-class commodity products.

So—will these price declines continue? Clearly not; it is the duty of any self-respecting research report to look for discontinuities in *any* ongoing trend—preferably with radical changes that start *now*. One way chip vendors can offset steep price declines is by redefine the battlefield, and there is evidence this may be starting to happen. Device pricing structures were once absolutely critical in system vendors' buying decisions, but once prices have fallen sufficiently, performance becomes paramount. And as more chip vendors abandon the bottom-feeding marketplace and move to the high end, head-to-head competition will decrease, and the average device sale prices may tend to stabilize.

## 1.5 For More Information...

Additional information on the state of the x86 microprocessor market may be found in the following publications:

### **Microprocessor Report Articles**

- 1: *Processors Battle for PC Market\**. Michael Slater, MPR vol. 2 no. 11, 11/88, pg. 1. (*Cover story.*)
- 2: *My Klone86 Will Be Out Very Soon*. Nick Tredennick, MPR vol. 4 no. 17, 10/3/90, pg. 3. (*Editorial.*)
- 3: *Intel's Dominance to Begin Fading in 1991\**. Michael Slater, MPR vol. 5 no. 1, 1/23/91, pg. 14.
- 4: *1991: The Year of RISC*. Nick Tredennick, MPR vol. 5 no. 2, 2/6/91, pg. 3.
- 5: *MIPS and Sunset*. Nick Tredennick, MPR vol. 5 no. 12, 6/26/91, pg. 12.
- 6: *Can the 386 Architecture be an Open Standard?\**. Michael Slater, MPR vol. 5 no. 15, 8/21/91, pg. 3. (*Editorial.*)
- 7: *1984 Revisited*. John Wharton, MPR vol. 5 no. 15, 8/21/91, pg. 15. (*Oblique Perspective column.*)

- 8: *Pinouts and Performance for 386/486-Compatible Microprocessors\**. Michael Slater, MPR vol. 5 no. 17, 9/18/91, pg. 3. (Editorial.)
- 9: *IBM to Make 386SX Variant with Cache*. MPR vol. 5 no. 17, 9/18/91, pg. 5. (Most Significant Bits item.)
- 10: *IBM and Intel To Jointly Develop x86 Chips\**. Michael Slater, MPR vol. 5 no. 22, 12/4/91, pg. 18. (Most Significant Bits item.)
- 11: *Proliferation of 386/486-Compatible Microprocessors to Accelerate in '92\**. Michael Slater, MPR vol. 6 no. 1, 1/22/92, pg. 1. (Cover story.)
- 12: *Die Like a Man*. Nick Tredennick, MPR vol. 6 no. 6, 5/6/92, pg. 18.
- 13: *The Incredible Shrinking PC\**. Michael Slater, MPR vol. 6 no. 13, 10/7/92, pg. 3. (Editorial.)
- 14: *Multivendor 386/486 Market Burgeoning\**. Michael Slater, MPR vol. 7 no. 1, 1/25/93, pg. 1. (Cover story.)
- 15: *Chip Developers Eager to Share Plans*. Linley Gwennap, MPR vol. 7 no. 1, 1/25/93, pg. 3.
- 16: *Readers Pick AMD as Top Processor Vendor*. Linley Gwennap, MPR vol. 7 no. 2, 2/15/93, pg. 15. (Feature article.)
- 17: *Cyrix IPO Reveals Fab Issues*. MPR vol. 7 no. 9, 7/12/93, pg. 19. (Most Significant Bits item.)
- 18: *Putting Windows NT in Perspective*. Michael Slater, MPR vol. 7 no. 13, 10/4/93, pg. 3.
- 19: *PPC 604 Powers Past Pentium*. Linley Gwennap, MPR vol. 8 no. 5, 4/18/94, pg. 1. (Cover story.)
- Other Technical References**
- 20: *Marketing High Technology*. William Davidow, Free Press, 1986. (Case histories of Intel marketing strategies.)
- 21: *Microprocessors: A Programmer's View*. Robert Dewar and Matthew Smosna, McGraw-Hill, Inc., 1990, ISBN 0-07-016638-2.
- Other Periodicals**
- 22: *Computer Revolution*. Stratford Sherman, Fortune Magazine, vol. 127 no. 12, 6/14/93, pg. 56.
- 23: *80x86 Wars*. Tom Halfhill, Byte, vol. 19 no. 6, 6/94, pg. 74. (Cover Story about Intel and its strongest x86 and RISC competition.)

24: *Compaq Rocks Corporate World with AMD Chip*. Brooke Crothers and Bob Francis, *Info World*, vol. 16 no. 38, 9/19/94, pg. 1.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



# 2

## x86 Family Heritage

---

Chip designers must always contend with the technological limitations of the day. Early microprocessor architectures were often compromised by restrictions imposed by transistor budgets, design tools, and packaging technology. Even now, such factors as die size and allowable power dissipation may directly affect the performance of a given processor implementation.

### 2.1 Jurassic Parts

The first microprocessors were never intended to serve as general-purpose devices. The four-bit 4004, introduced by Intel in 1971, was designed for use in simple desk calculators built by Busicom, a Japanese office-equipment company. The eight-bit Intel 8008, introduced the next year, was intended to replace the control logic and reduce the cost of a line of commercial CRTs built by Datapoint.

It was not until 1974 that a microprocessor appeared that provided sufficient resources to build a simple, general-purpose computer. This device, the Intel 8080, contained an eight-bit ALU, accumulator, six working registers, and rudimentary support for a 16-bit (64K-byte) memory address space. Nevertheless, programs for the 8080 and its Intel 8085 and Zilog Z80 derivatives had to be small and simple, arithmetic precision was limited, and address space was constrained.

In 1975 work began on the successor to the 8008 and 8080, a 16-bit microprocessor to be called the 8800. At the time, in the

era of the DEC VAX (Digital was called “DEC” back then) and its supersophisticated auto-incrementing indirect memory-addressing modes, complexity was seen as good. Processor architects and software scientists spoke of a need to decrease a perceived “semantic gap” between what high-level programs could specify in a single statement and what machine-language programs could accomplish in a single instruction. The 8800 was supposed to close the semantic gap.

Alas, the 8800 fell victim to creeping elegance: its architecture grew to include an ALU that could perform 8-, 16-, or 32-bit integer arithmetic and 32-, 64-, or 80-bit floating-point math. (This was the first microprocessor of any type to include floating-point variables as a fundamental data type. The floating-point formats and semantics defined for the 8800 project would later evolve into the IEEE-754 floating-point standard now supported by essentially all new microprocessor designs.) Direct hardware support was added for object-oriented software, per-access privilege verification, and fault-tolerant multiple-processor systems.

The project relocated from Santa Clara to Oregon, the better to isolate its key engineers from Silicon Valley-based competition. The move also had the effect of isolating the key designers (some thought) from reality. Microcoded primitives were added to support software multitasking and communications between processors and processes. The good news was that software could suspend operation of one process and initiate an entirely new task in a sterile, protected execution environment, all with a single machine-language instruction. The bad news was that single instruction might be hundreds of bits long and could require several tens of milliseconds (!) to complete.

In time it became necessary to split the 8800 design into a two-chip set; later, a third chip would become necessary when system designers discovered the original architects had neglected to provide any mechanism for transferring data between memory and the outside world.

The device’s nomenclature would evolve, too: by the time it was finally introduced in 1981, it was destined to acquire the moniker “iAPX432,” a so-called micromainframe computer. Its level of sophistication would by then prove to be too high for the marketplace, and its performance would prove to be far too low. A few years later, work on the 432 would be quietly discontinued.

**The 8086** As the sophistication of the 8800 grew, its introduction schedule slipped. Meanwhile, back in Santa Clara, Intel had decided in 1977 to develop a simpler interim processor to fill the latest one-year slip in the 8800 design schedule. An emergency rush project was begun to develop a new 16-bit successor to the 8080. Only two constraints were imposed on this device:

- its architecture had to be close enough to the 8080's that assembly language programs written for the earlier device could be automatically translated to the new one, and
- the entire design process had to be completed, and 1,000 working parts had to be on Intel's distributors' shelves, within 12 months of project initiation.

According to industry lore, two Intel engineers, Steve Morse and Bruce Ravenel, were relieved of their other duties and given the task of developing a complete software specification for the new, stop-gap machine. After holing up for two weeks in an empty conference room, they were joined by a third designer named Bill Pohlman. One week later the architects completed a preliminary spec. The architects requested a fourth week to define memory-management facilities; management refused. There just wasn't time, it was decided, for the crash development program to concern itself with such nuances.

The device they invented was called the 8086, a part number selected to hasten its acceptance by giving customers the impression it was designed to be a natural progression from the recently introduced 8085. The 8086 architecture was essentially a "stretched" version of the 8080. Its ALU supported both 8- and 16-bit operations, its data bus was widened to 16 bits, and its address bus was expanded to 20 bits, allowing a memory space up to one megabyte.

The 8086 programming model (Figure 2-1) defined fourteen 16-bit registers. Four of these (designated AX, BX, CX, and DX) were used for general-purpose arithmetic. Four more registers (SI, DI, BP, and SP) served as "index" or "pointer" registers to simplify address calculations. Four others (CS, DS, ES, and SS) designated various memory segments to be used for accessing instruction code, data values, or the stack. The final two registers held the instruction pointer (IP) and status flags.

In the interest of time, the 8086 specification chose not to define any floating-point operations; instead, the architects decided floating-point support would be provided in the form of a collec-

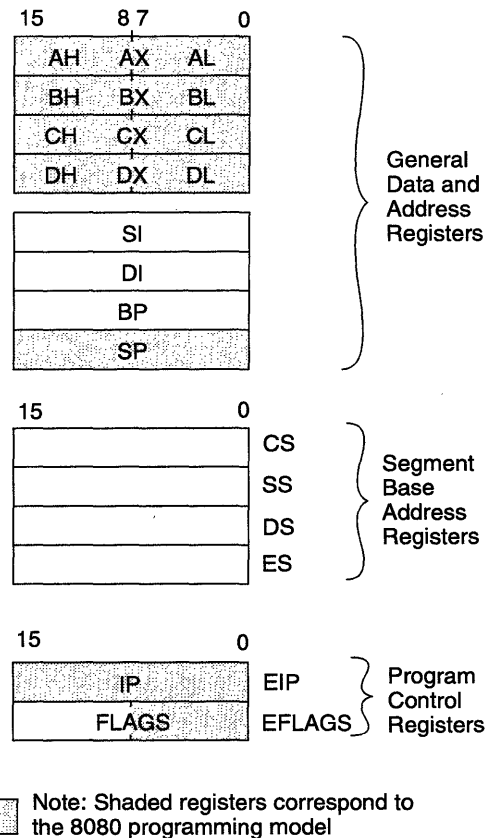


Figure 2-1. 8086 programming model.

tion of escape instructions that would operate as no-ops in the 8086 instruction set. A second, auxiliary, floating-point “coprocessor” would monitor the 8086 instruction stream and detect and interpret the instruction whenever one of these reserved opcodes was executed. In the case of the 8086, this FPU coprocessor was dubbed the 8087.

The 8086 provided compatibility with 8080 software by mapping each of the earlier device’s working registers onto the high- or low-order bytes of the 8086’s four general-purpose registers. Each instruction in the 8080 repertoire was likewise supported by an equivalent eight-bit 8086 operation with identical semantics, and the settings of each of the 8080’s status flags were mimicked by the corresponding bit in the 8086 FLAGS register. Thus, with the proper choice of operations and operands, it was possible to emulate the exact semantics of every instruction in an existing 8080 program with an equivalent (albeit generally longer and typically less efficient) 8086 operation.

In order to expand the 8080 address space beyond 16 bits (64K bytes), the 8086 adopted a system of memory-space segmentation. Segmentation can be thought of as a form of glorified bank-swapping. Each code, data, or stack reference may be configured to access a different region (“segment”) of memory. Before each memory access, the value in a 16-bit register that specifies the corresponding segment is shifted left four bits to form a 20-bit “segment base address,” which is added to the 16-bit “offset” computed by the instruction-addressing mode.

Segmentation provided a quick and dirty method to expand the physical address space beyond 64K bytes while working within the constraints of a 16-bit CPU. The downsides were increased complexity in the programming model, incompatibilities between software modules, inefficiencies in execution time, and added hardware cost.

(A bit of industry arcana: the original Morse-Pohlman-Ravenel specification had included four additional “limit” registers that defined the size of each of the four segments. Their intent was that these registers would provide a rudimentary form of memory protection. Errant software that attempted to access memory above its allocated segment size could thus be intercepted before any damage could be done.

(When the chip was built, these registers were left off to save transistors and design time. “What good would they do?” the chip designers asked. By the time a software product ships, they figured, any errant accesses should have been located and fixed! Had the architects’ intent been carried through to the final design, however, software vendors might have used segmentation more in the way it was intended. Software compatibility problems that arose as the segmentation model changed in future-generation products might thus have been avoided.)

The point of this discussion is to show that the 8086—the grand patriarch of the x86 dynasty—was born disadvantaged. Its architecture was compromised to preserve compatibility with a product line that began as a cost-reduced CRT controller. Its definition process was compromised to save specification time. And its implementation was further compromised to save cost.

A thousand working 8086 processors did indeed make it onto distributor shelves within a week of the 1978 target date, largely due to the wholesale compromises struck throughout the part’s gestation. Customers were uniformly underwhelmed, and disappointingly slow to accept the new part. Months passed,

then quarters, then years, while Intel marketers tried to convince themselves that sales would begin to soar, just as soon as the extended design-in cycles ended.

Customers who needed 16-bit processing power were far more interested in the 68000 being developed by Motorola, or the upcoming Z8000 from Zilog. The consensus among system designers was that these other devices would provide a more regular, orthogonal, true minicomputer-like architecture.

The 8086, in comparison, had an architecture deemed arcane by programmers, a memory segmentation scheme that was cumbersome to use, and a bus interface that was expensive to design for. While a wealth of byte-wide RAMs, EPROMs, and peripherals had by then been introduced to support the 8080 and 8085, and while new peripherals were allegedly being designed for 16-bit buses, no such devices existed yet.

### **The 8088**

Its primary strength being in the area of hardware, Intel responded to these concerns by attacking the hardware problem first. Intel quickly developed a new device that was fully software compatible with the 8086, and contained essentially the same 16-bit core, but required only an eight-bit interface to memory and I/O. This device (called the 8088) could thus be used with inexpensive support components developed for the 8085. While the 8088 attracted considerable interest among designers of high-end embedded-control applications, even here the part was slow to be accepted.

Its secondary strength being marketing, Intel then responded with a number of intensive marketing programs. Most notable was “Operation Crush,” which included saturation ads, high-pressure customer visits, and promises to the field sales force of generous cash rewards and paid vacations to Tahiti and other exotic locales if certain aggressive sales goals were met. (See reference 10 and **Chapter 4: Vendor Profiles** or further information on Operation Crush.)

All this mattered little until 1980, when a small team of renegade IBM engineers from Boca Raton, Florida decided to build a new “personal computer.” The CPU they selected was the Intel 8088. Intel’s 16-bit processors had been in production somewhat longer than Motorola’s, and the compatibility of the 8086 instruction set with existing 8080 programs (they thought) would hasten the availability of third-party software. Moreover, the 8088’s byte-wide bus would be cheaper to design a system around.

Thus the 8088-based IBM PC came to be introduced in the fall of 1981. IBM approached Bill Gates, who had formed a small company to sell BASIC interpreters, about acquiring an operating system. (The president of IBM had served on the national board of United Way with Bill's mom.) Microsoft in turn bought the rights to a self-proclaimed "quick and dirty" operating system developed by an even smaller company, renamed it MS-DOS, and sub-licensed it to IBM under a sweetheart deal. The rest, as they say, is history. (See references 8 and 9.)

**The 80186** The desire to further reduce system costs led Intel to develop the 80186, a highly integrated processor that combined an 8086-like core with address decoders, timers, an interrupt controller, and direct-memory access (DMA) logic comparable to those found in an IBM PC. Just for good measure, a handful of new instructions were added to the CPU in the process.

Unfortunately, the peripherals built into the 80186 did not strictly match those of the PC, either in the functions they performed or the mechanisms for accessing them. PCs built with the 80186 were thus unable to execute certain DOS programs. The 80186 became a highly successful product, with derivatives still being introduced, but only in the realm of embedded control. I/O-related software compatibility problems prevented the part from being able to run many MS-DOS applications, and the 80186 failed totally as a processor for desktop computers.

**The 80286** Even a one-megabyte address space proved insufficient, as software sophistication and the users' need for power grew. Next came the Intel 80286, a device with a faster clock, a 24-bit (16-MB) physical address space, on-chip memory management and protection, and yet a few more specialized instructions.

Following reset, an 80286 defaults to a mode in which it has the same 1MB address space and the same memory-segmentation scheme as the 8086, and is fully compatible with 8086 software, and incidentally delivers three times the performance. The 80286 architects intended that 8086-mode software would simply initialize a series of tables used by the memory-protection logic and then switch the processor into a "native" operating mode in which the full 16-MByte address space and the new improved 286 memory-management logic would be enabled.

The 80286 was quickly adopted for use in the IBM PC/AT, a successor to the original PC. Unfortunately, the 80286 architecture suffered from two problems. First, in native mode the semantics of the segmentation registers were different from those of the

8086: the earlier part used the contents of the segment registers directly as a base address, while in the 80286 these registers determined base addresses indirectly via a memory-based segment-attribute table. Because of this change, much existing MS-DOS software would not run in 80286 protected mode.

Second, in the interest of software security, once the switch was made to native mode, there was no way (short of a full hardware reset) to switch back to a full, less-secure, 8086-compatibility mode. This, naturally, didn't stop IBM. PCs built using the 80286 included a mechanism by which protected-mode 80286 software could indeed return to 8086-compatibility mode by issuing a command to the keyboard controller chip, thereby causing the controller to toggle an unused I/O pin, which would then reset the main 80286 CPU, which could then examine coded data in memory to determine whether it had experienced a cold or warm start, and could thus reinitialize the entire system either from scratch, or could resume execution of an earlier program based on process state variables stored in memory. (I'm not making this up.)

Needless to say, 80286 task-state-switching times were somewhat slow. As a result, 80286-based personal computers were seldom used as anything more than accelerated 8086 boxes. Even without these flaws, though, the 80286 would still have been fundamentally limited by its 16-bit ALU and register set. Even with a 24-bit address space, memory segments were still held to 64K bytes. Thus even native-mode 80286 application programs had to battle all the same architectural limitations as the original 8086.

## 2.2 The 386 Family

With the introduction in 1985 of the 32-bit Intel 80386 microprocessor (later to be renamed the i386, and later still the i386DX), all internal registers, buses, and arithmetic operations were expanded to 32 bits. Physical memory addressability was expanded to  $2^{32}$  bytes (four gigabytes). A number of new instructions and new addressing modes were added, and the orthogonality of existing operations was enhanced.

Program size became essentially unbounded. A virtual-memory paging system was added that provided a logical address space up to  $2^{46}$  bytes (64 terabytes). New process-management data structures, instructions, and interprocess protection mecha-



nisms were added to let multiple users run multiple programs, even multiple operating systems, all at the same time, with each task and context fully protected from the software malfunctions of the others.

A new mode of emulating the 8086 was also added, making it possible to run multiple DOS, UNIX, or OS/2 applications simultaneously. It thus became practical to preserve a user's investment in 16-bit applications software without precluding the use of new, native-mode 386 capabilities.

The original segmentation model was improved. Two new segment registers were added, and the offsets allowed within each grew from 16 to 32 bits. Each code segment or data structure could thus use as much of the physical address space as it needed, up to a full 4GB.

But just as the 16-bit bus of the original 8086 was wider than what was needed for many applications, the 32-bit bus of the i386DX increased system costs for certain low-priced applications. Following the precedent of the lower-cost, 8-bit-bused 8088, Intel later introduced a cost-reduced member of the 386 family called the i386SX: a lower-cost 16-bit-bus device intended to compete directly with the 80286 on system costs.

## 2.3 The 486 Family

In 1989 Intel introduced the i486DX microprocessor, a faster and more highly-integrated addition to the x86 family. In addition to its 32-bit, 386-compatible CPU, the 486 included eight kilobytes of on-chip cache, an on-chip floating-point coprocessor, and significantly faster external bus protocols.

In keeping with the precedent of the 16-bit-bus 8086 being followed by the lower-cost 8-bit-bus 8088, and the 32-bit-bus i386DX being followed by the lower-cost i386SX, Intel (true to form) introduced in 1991 a lower-cost version of the i486DX device, again with an "SX" suffix. Instead of reducing its bus width, however, the i486SX provided essentially the same system interface as the i486DX, but eliminated the on-chip FPU as a justification for its lower price.

One area in which the 486 family departed from the tradition of its predecessors was in *not* reopening the underlying architecture. Whereas the 8080, 8085, 8086, 80186, 80286, and 386 had

each augmented the instruction set, registers, and other capabilities of their predecessors, the 486 was essentially just a faster implementation of the architecture solidified by the 386. Operating systems and applications software originally developed for 386 CPUs could thus run unchanged on systems built with a 486. Indeed, aside from minor revisions to BIOS firmware to configure the cache and FPU, the 486 instruction set did not change sufficiently to justify the development of customized software, so the market had little excuse not to quickly accept the new part.

The 386 and 486 processors departed from industry tradition in another major way. Microprocessors had typically been multiple-sourced. Whenever a new device was introduced, the company that developed it would immediately license its design to competing vendors. The availability of alternate sources was thought to increase the perceived viability of a new product. Moreover, the resulting competition for market share would reassure OEM system vendors that prices for the part would continue to fall steadily. By the time the 386 was announced, half a dozen vendors were competing for 80286 market share. And true to the laws of supply and demand, 286 prices had eroded to the point of vanishingly small profit margins.

The 386 and 486, on the other hand, were the first Intel processors that were *not* second-sourced. System vendors that wanted the latest and greatest chips had no choice but to buy from Intel. The resulting monopoly kept margins high, Intel's profitability soared, and a window of opportunity opened for chip vendors wishing to share in the wealth.

## 2.4 The Explosion of Third-Party CPUs

Into the alternate sourcing void stepped a number of vendors, each with its own strategy on how best to tap into the Intel profit pool.

### **AMD Becomes an Industry Force**

AMD was first to arrive, introducing its own version of the 386DX in 1991. Soon the AMD product line grew to include more than a dozen alternate implementations of Intel's mainstream 386 and 486 processors. AMD parts are generally pin- and spec-compatible with Intel's, but with higher-frequency operation and improved electrical characteristics.

In its first two years of participation in the 386 market, AMD became remarkably well established. Its customers include dozens of third-tier companies, most of the second tier, and a growing number of the first. Even IBM began using AMD processors in low-end machines sold in Europe, and major companies such as Digital Equipment Corp. and Compaq are featuring AMD-based products. By the end of 1992, AMD had captured (or acquired by default) more than half of Intel's 386 unit production. By the end of 1993, Intel had withdrawn almost completely from the competition for 386-based PCs.

### **Chips and Technologies Crashes and Burns**

Chips and Technologies was next to announce a broad range of planned 386 derivatives in 1991. Six products were announced and more promised. Some were said to be compatible with Intel devices, while others were to have enhanced pinouts and improved functionality.

Alas, compatibility problems forced C&T to go through several iterations of its design before the chip was adequately debugged, which eroded customer confidence. The company also encountered repeated production delays, and the devices needed an extremely large die that made it hard for C&T to compete with more established products on price. Facing tough competition and dwindling cash reserves, C&T bailed out of the 386 market and canceled further 486 developments.

### **Cyrix Joins the Fray**

Cyrix, a small company that had previously made only Intel-compatible math coprocessors, joined the fray in 1992. Cyrix's Cx486SLC and Cx486DLC combined a 486-class CPU core with a 1K cache in an i386SX- or i386DX-compatible package, and quickly became quite successful in the notebook market and in some desktop sockets. Since then Cyrix has continued to innovate, introducing devices that could match (and in some cases exceed) the features of Intel's comparable 486s.

Cyrix's initial success throughout 1993 and 1994 came primarily from second- and third-tier system suppliers. By mid-1994, Cyrix had staked its claim in the rapidly developing notebook PC market and was shipping about 3% to 4% of all 386 and 486 chips—not bad for a tiny company battling better-established Intel and AMD with its first microprocessor products. Considering that even 4% of the x86 chip market is more than two million chips, a small market share can still be very attractive to a young chip supplier. Within a short period of their introduction, Cyrix had easily shipped far more 486 CPUs than the total unit sales of any RISC processor company for desktop systems.

**TI Plays Copy Cat**

In contrast to Intel and AMD, Cyrix does not own its own fabrication facilities; instead, it contracted with SGS-Thomson and Texas Instruments to build the chips Cyrix would sell. In return, TI secured the right to build and sell devices based on the Cyrix core. TI began selling its own versions of the Cyrix 486SLC and 486DLC designs in 1992.

During 1993 the Cyrix-TI relationship dissolved into litigation, leaving TI without an outside source of next-generation x86 designs. Still, in 2Q94 TI began announcing custom designs derived from the Cyrix CPU core but with additional on-chip functions. The most aggressive of these (code named “Rio Grande”) was later discontinued, after TI failed to find any customers interested in using the part. Unless TI succeeds in developing more sophisticated x86 core technology of its own, the company is unlikely to remain a long-term player in the market.

**IBM Pulls an End Run**

The surprise entrant in the x86 sweepstakes turned out to be IBM—all the more a surprise because IBM had worked closely with Intel for years, and at one point had even owned 12% of the company’s stock. Under various technology exchange agreements, IBM was prohibited from selling any x86-architecture microprocessors of its own design on the merchant market if it used intellectual property gained from Intel.

In 1993 IBM attempted to exploit a loophole in the Intel technology license. Apparently, the Intel agreement let IBM sell microprocessor subsystems or daughterboards, but never spelled out what level of complexity was required for a “chip” to be deemed a “system.” Moreover, the agreement apparently let IBM provide CPU chips to third-party manufacturers contracted to assemble IBM boards or systems, and to sell those finished boards and assemblies to OEM system vendors.

Apparently, though, there was nothing in the Intel deal to prevent the company doing subcontracted assembly work from being the same company that bought the finished boards when they were done. In other words, IBM could (it felt) provide raw CPU components to an established PC vendor—Compaq, say, for the sake of discussion—that would then assemble motherboards using the devices, and pay IBM not for the *chips* but rather for the *value* those chips added to the *boards* they were in! (While IBM representatives at one point openly solicited business under these terms, there’s no evidence that Compaq or any other system vendor attempted to exploit this loophole.)

Thus IBM was allowed to market its 386SLC, 486SLC2 (later to be renamed the BL486SLC2), and “Blue Lightning” (later to be renamed the BL486SX2/SX3) processors *directly* to system vendors as part of a CPU daughterboard—even if these consisted of no more than a chip or two on a small circuit board—and hoped to sell parts to system motherboard builders *indirectly* through subcontracted assembly and licensing arrangements.

IBM’s chips are technologically impressive. They are conceptually similar to the Cyrix Cx486SLC and Cx486DLC, but with a faster core and much larger caches. They are also remarkably small, thanks to IBM’s advanced IC processes. IBM’s BL486SX2/SX3 chip could also serve as the heart of an end-user upgrade product for existing 386 systems. Nevertheless, IBM wasn’t likely to play a major role in the microprocessor business as long as its chips could only be sold within subassemblies.

In 1994 IBM became an alternate source for Cyrix-designed 486-class products, thus opening up a channel for direct chip-level sales to system OEMs. Perhaps more important, IBM has acquired the rights to sell current and future leading-edge designs developed by Cyrix and NexGen.

### **Late Arrivals and Also-Rans**

In addition to these players, several others have made moves on the 386/486 business. In Japan, V.M. Technology (VMT)—a company founded by Masatoshi Shima, a contributor to the 4004 and Intel’s lead designer on the 8080—has spent years developing a 386SX-equivalent processor for ultralow-power Asian-language word processors and hand-held digital appliances. The device was finally announced in 1993. As the market moves to the 486, it’s not clear if VMT will join the majors.

United Microelectronics Corp. (UMC), one of the largest Taiwanese semiconductor makers, licensed a high-speed 386 design from Irvine-based Meridian Semiconductor, and the company said it would market a 486SX-compatible processor in 1993. These chips were announced and reportedly began sampling in Asia during 2Q94. The company claims it will also introduce a 486DX2-caliber product in 1995.

Back in the U.S.A., Integrated Information Technologies (IIT) has a 486-compatible development program under way, but no details have been released. IIT has entered into a partnership with National Semiconductor for undisclosed future products, possibly including microprocessors.

NexGen, whose products have been delayed many times, finally began sampling product in 1993. In 1Q94 the company announced a non-pin-compatible two-chip set said (by NexGen) to compete favorably with Pentium on both price and performance. NexGen's plan originally called for it to sell only systems, but it has abandoned this plan and is focusing on being a chip-level microprocessor supplier.

Undoubtedly, numerous other companies have 386/486 processor developments under way; this market is just too big and too profitable to ignore. It seems that it is only a matter of time before one of the major Japanese semiconductor makers breaks into the business, though legal questions may delay this until late 1995 or later.

## 2.5 Intel Strikes Back

Despite the near total loss of the 386 market to AMD—or possibly *because of* the loss—Intel has continued to report steadily improving financial results. As AMD began encroaching on 386 sales, the average selling price (ASP)—and potential profit—of the devices quickly fell to well under \$100. The ASP of a 486 at the time remained several times higher, and freed of its 386 production commitments, Intel could devote more of its production capacity to higher-margin 486s.

So as AMD 386 unit volume grew, so did Intel 486 production, and Intel profits soared. This is a familiar Intel strategy: when prices on older chips drop, Intel relinquishes the chore of servicing that market to a competitor and reallocates its fab capacity and other finite resources to newer, higher-margin chips.

### **“Second-Generation” 386 and 486 Designs**

In order to counter AMD's low-voltage and low-power 386 devices, and in an effort to bolster 386 ASPs, Intel announced in 4Q90 a fully redesigned, static, low-voltage, and highly integrated version of the i386SX CPU. This device, called the i386SL, had a brief but brilliant career: it was quickly selected by virtually every first-tier vendor of 386 lap-top PCs and sold in extremely high volumes until early 1993, when the device's high manufacturing costs and Intel's desires to migrate the customer base away from the 386 led the part to be discontinued. A fully redesigned, static, low-voltage, and highly integrated version of the i486SX, called the i486SL, was also introduced in 1992, but active promotion of the part for new designs ceased a year later.

Other than price cuts and the usual clock-rate increases, the biggest and most lasting news in Intel's 486 product line was the introduction of "clock-doubler" versions of the 486, known as the i486DX2 in the OEM market and as the OverDrive chip in the retail market. These chips match higher on-chip clock rates with slower motherboard designs for a happy compromise between system cost and performance. While a 50-MHz i486DX2 isn't quite as fast as a non-doubled i486DX running flat-out at 50-MHz, it does provide 85–90% of full performance with a much lower system cost. A clock-doubled i486DX2-66 roughly matches the performance of the i486DX-50, depending on the application, while allowing a significantly simpler and less costly 33 MHz motherboard design.

(Other vendors have begun playing clock-multiplier games as well. IBM's 486SLC2 has a 16K on-chip cache and a clock-doubler in a 386SX pinout. IBM's "Blue Lightning," has a clock-tripler to match its 60-, 75-, and 100-MHz internal rate to 20-, 25-, and 33-MHz system designs. AMD, Cyrix, and IBM began shipping clock-doubler versions of 486SX- and 486DX-class machines, prompting Intel to quickly introduce its own i486SX2 and the 100-MHz clock-tripled IntelDX4.)

It's not clear how successful Intel has been with its OverDrive upgrade processors (the retail version of the i486DX2). Cutting the cost of a CPU upgrade sounds great, and Intel's shareholders should like the idea of selling two processors for every system, but it remains to be seen how many system owners will bother to upgrade. One estimate is that between three million and four million units were sold during 1993, which would account for as much as one billion dollars of retail sales. Even if no one were to buy upgrade processors, however, Intel still benefited from the OverDrive campaign because it gave consumers another reason to buy an Intel 486 system instead of a competitor's high-end 386.

**Pentium** The formal announcement of Pentium occurred in March of 1993, though only relatively small (by x86 standards) numbers of devices were delivered through fourth quarter 1993. While the floodgates on Pentium production began opening in the first quarter of 1994, Intel thinks it will likely be mid-1995 before Pentium shipments approach current 486 volumes.

Because of constrained availability and other factors, Pentium systems carried premium prices through much of 1993. With no direct competition at that performance level, Intel had little motivation to price the part aggressively. System makers also

try to charge the highest markup on systems using the latest processor, hoping to recapture whatever profit they can while demand for the new technology is high.

## 2.6 Commentary

The 386/486 market has changed dramatically and is continuing to change. Straight Intel clones, such as AMD's 386, will become a thing of the past as each vendor develops its own implementation of the architecture. While Intel likes to call the compatible chip vendors "imitators," they are imitating less and less.

With Pentium-class designs, no vendor can afford the delay incurred by waiting for Intel's chip to ship and then analyzing it before developing a competitive product. Instead, multiple vendors are creating their own independent implementations of a de facto-standard instruction-set architecture. The result will be a wider array of offerings than ever before, giving system designers and users more choices and more competitive pricing.

### **The Intel Technology Treadmill**

As superscalar implementations of the x86 architecture become common, Intel will gain a new advantage over its competitors. Pentium requires a significantly different code-generation strategy (in the compiler) to produce the fastest programs. Pentium-aware compilers are essential to getting the most performance out of that processor. Intel is working closely with compiler vendors to make sure that they produce code that is well optimized for Pentium. Developers of performance-critical applications software are more likely to provide recompiled versions as Pentium systems become more widely available.

Now consider the plight of a company such as Cyrix, which is developing its own superscalar x86 processor that it expects will be faster than Pentium. Cyrix has little chance of getting compiler developers to produce Cyrix-specific code optimizers, and application developers wouldn't be interested in any case because Cyrix, a newer, smaller and less established player, can't guarantee the market demand. Because of this, Cyrix is finding it necessary to design processors to deliver comparably high performance, whether or not the software it's running has been recompiled according to new rules.

Vendors of Intel-compatible processors must therefore design their processors to make do with the compiler optimization



strategies dictated by Intel for the 486 family or for Pentium. This can be achieved, in part, by using similar design techniques. Another approach is to design the processor hardware so that code-generation strategies are less critical by allowing speculative and out-of-order instruction execution. These techniques allow the processor to effectively reorganize the program as it is being executed, reducing the dependence on compiler optimizations, but they also make the processor design considerably more complex and harder to debug.

### **The Future of the x86 Market**

The battle of the RISCs, while obviously critical to RISC vendors, is nearly irrelevant to x86 vendors. Even if PowerPC sales were to take off or if Windows NT on RISC platforms were to be a wild success, RISC processors would likely sell no more than a few million chips per year into the desktop market. The effect of this on x86 sales might be to cut annual shipments from 60 million units per year to, say, just 55 million devices—barely a dent in the success of the architecture. Even as early as 1992, Cyrix—the youngest and smallest of the alternate-source x86 vendors—was shipping more CPUs for the desktop market than all RISC workstation vendors combined.

The real battle for volume in the desktop microprocessor market, at least until late in the decade, will be among the vendors of x86 microprocessors. Intel is clearly in a commanding position. Pentium is still Intel-proprietary and has been nearly immune from the price compression that has consolidated the low-end and midrange processors.

Unless Intel stumbles badly with Pentium, that company will retain its performance lead among x86 implementations and have a new core processor from which to spin derivatives—while its competitors slog it out in the 386 and 486 trenches and attempt to establish new Pentium-class competition. Pricing of 486DX chips will surely follow the trend of the 386 and 486SX, eventually falling under \$50. The big price cuts will occur as AMD, Cyrix, TI, IBM, and probably others ramp up production. By then, Intel will have once again established the high ground with its Pentium-family derivatives and may willingly relinquish the low-end 486 market, just as it did with the 8086, 80286, and 386 markets before it.

Intel still holds the keys to the kingdom in one important sense: it is the only company with the market influence to evolve the architecture. Perhaps with the “P6,” and surely with the “P7,” new features—and possibly even an entire alternate instruction set—will be added to the architecture. Intel has the leadership

role that allows it to add new capabilities and gain broad industry support for them, while other vendors will have to follow Intel's lead.

## 2.7 For More Information...

Additional information on the x86 microprocessor family heritage may be found in the following publications:

- |                                       |   |
|---------------------------------------|---|
| <b>Vendor Publications</b>            | 1: <i>Intel 8086 Programmer's Reference Manual</i> . Intel Corporation, 1989, order #270710-001.  |
| <b>Microprocessor Report Articles</b> | 2: <i>386-Real-Mode-Compatible Chip from Japan</i> . MPR vol. 2 no. 5, 5/88, pg. 2. ( <i>Most Significant Bits item.</i> )                                |
|                                       | 3: <i>Japanese Startup Develops "Virtual Microprocessors"</i> *. MPR vol. 3 no. 9, 9/89, pg. 8. ( <i>Feature article.</i> )                               |
|                                       | 4: <i>Startup Reveals Design for Superscalar 386-Compatible Processor</i> *. Michael Slater, MPR vol. 5 no. 4, 3/6/91, pg. 7. ( <i>Feature article.</i> ) |
|                                       | 5: <i>Football and Microprocessors</i> . Nick Tredennick, MPR vol. 6 no. 1, 1/22/92, pg. 19. ( <i>Editorial.</i> )  |
|                                       | 6: <i>VLSI Previews x86 PDA Processors</i> . MPR vol. 7 no. 9, 7/12/93, pg. 4. ( <i>Most Significant Bits item.</i> )                                     |
|                                       | 7: <i>VLSI Integrates 486SL Power Management</i> . Linley Gwennap, MPR vol. 7 no. 9, 7/12/93, pg. 16. ( <i>Feature article.</i> )                         |
| <b>Other Technical References</b>     | 8: <i>Hard Drive: Bill Gates and Making of the Microsoft Empire</i> . James Wallace and Jim Erickson, Harper Business, 1993, ISBN 0-88730-629-2.          |
|                                       | 9: <i>Gates</i> . Steven Manes and Paul Andrews, Simon & Schuster, 1993, ISBN 0-671-88074-8.  |
|                                       | 10: <i>Marketing High Technology</i> . William Davidow, Free Press, 1986. ( <i>Case histories of Intel marketing strategies.</i> )                        |

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)

# 3

## The x86 Microprocessor Architecture

---

A microprocessor's architecture and its implementation are two very different things. Architecture relates to the collection of instructions a computer can execute, the registers within the computer on which the instructions operate, and the memory-based data structures interpreted and maintained by the CPU. These are the resources with which assembly language programmers and compiler writers must be concerned when constructing new operating system software and application programs.

Implementation relates to the internal hardware resources used to perform the operations determined by the architecture, resources such as arithmetic-logic units, register files, address adders, caches, and internal data buses. Any given architecture may be implemented in many different ways, depending on the relative importance of such factors as cost, performance, and complexity.

Most of the remaining sections of this report relate to the technical details of various implementations of the same x86 architecture. Unfortunately, certain aspects of a computer's architecture may have considerable long-term ramifications on its implementations, affecting how cheaply or how efficiently high-performance versions can be built.

### **Common Ground**

To understand some of the issues affecting various implementations of x86-family devices, it is therefore useful to understand the underlying architecture. This chapter gives a quick overview of the architecture common to each of the 386, 486, and Pentium processors now in production.

Specifically, this chapter summarizes the 386 architecture, including the application- and system-level programming models, the instruction set, memory-addressing modes, and software emulation and compatibility issues. The details presented here describe the capabilities shared by all members of the 386 and 486 product lines made by Intel, AMD, Cyrix, TI, and others. Processor-specific extensions to the architecture, including new instructions implemented by the 486 and Pentium product lines, and the new “System-Management Mode” facilities added to recent power-conscious 386 and 486 devices, are described within the respective product descriptions in **Part III** and **Part IV** of this report.

Be aware, though, that the 386 architecture is complex in every sense of the word. The bad news is that it’s beyond the scope of this report to cover the 386 architecture at more than a superficial level. The good news is that the 386 has the most heavily documented architecture in computer history. Consult the **For More Information...** section at the end of this chapter for a list of additional references relating to the 386 programming model and instruction set.

## 3.1 Programming Model

The user programming model of 386-family microprocessors involves a number of different aspects, including the user-visible programming model, instruction set, and addressing modes that reference external memory-based operands.

### Working Registers

Figure 3-1 shows the registers that are visible to 386 and 486 applications programs: eight 32-bit working registers, six memory-segment selector registers, and the instruction-pointer and flags registers.

The 386 architecture is register based; most instructions operate on eight working registers designated EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP. Each is 32 bits wide, and each can contain raw data, address components, or a full memory address, according to the needs of the software.

By convention, the first four working registers most often manipulate data, and the latter four are generally involved in computing memory addresses. Several of the working registers perform dedicated functions, and special instructions operate on these registers by default. For example, EAX is the

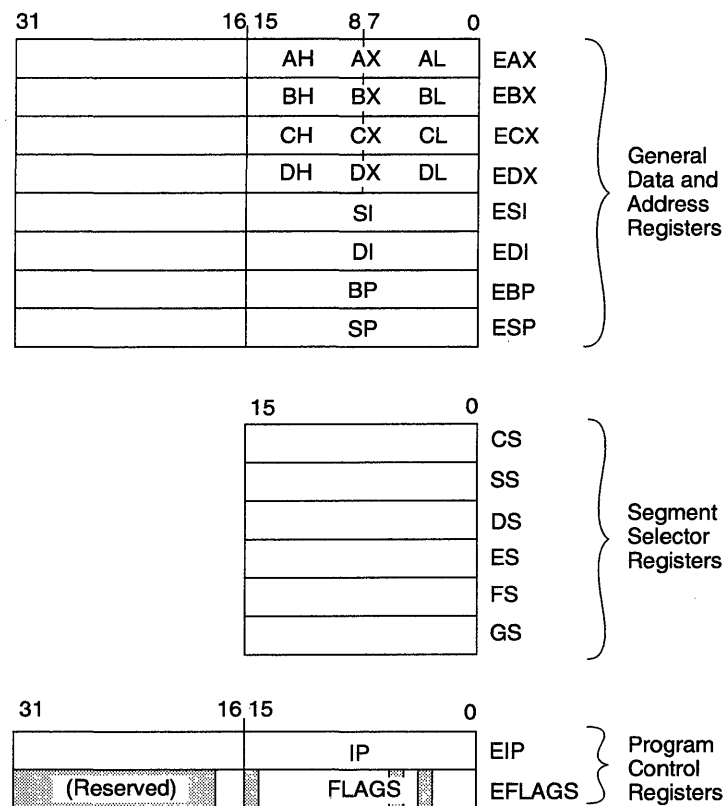


Figure 3-1. The 386 user programming model.

destination or source for instructions affecting input or output ports, and sometimes acts as a data accumulator or temporary register. The ECX register holds loop-count values for iterated byte-string moves or searches. The EDX register operates as a pointer for indirect I/O port operations.

The ESI and EDI registers are source and destination index pointers, respectively, for string operations. Each pointer is automatically incremented or decremented by the number of bytes involved following each iteration. The ESP register acts as an implicit stack pointer.

A number of instructions affect operands in the system-stack region of memory. These instructions read the ESP register to determine the current top-of-stack, and increment or decrement ESP accordingly. EBP is a base pointer, used in referencing stack-based variables. These include parameters passed to a subroutine, as well as local variables allocated within the subroutine stack.

In some cases, allocating particular registers to dedicated hardware functions improves the performance of time-critical operations. Dedicated logic and special data buses connected to the ESP, for example, can retrieve and update its contents without disrupting data transfers between the register file and the ALU.

In other cases, use of a default register increases the density of 386 microprocessor programs. Most arithmetic and logical instructions come in two types. The generic form can specify any register as its destination operand, while a shorter version implicitly affects the EAX register. A special conditional branch instruction tests whether the loop count (ECX) register is zero.

It should be noted, though, that each of the working registers is indeed general purpose. Despite the usage conventions described above, and in addition to performing functions imposed by the hardware, each register can be a source or destination operand for all register operations.

Programs developed for the 386 architecture generally make use of the full precision of each register. It is often necessary, however, for a 32-bit microprocessor to perform operations with less precision, such as when manipulating 8-bit ASCII character strings or when interacting with byte-wide peripherals or memory devices. A 32-bit microprocessor may also have to limit its precision to share database files or memory-based data structures with a less capable processor. A special case of the latter situation arises when a 386 microprocessor executes programs originally developed for (16-bit) 8086- or 80286-based systems.

When necessary, all arithmetic and logical operations of the 386 instruction set can manipulate partial-width subfields within the general-purpose register set. Instructions performed to 16 bits of precision can reference just the low-order half of each of the registers, designated by the symbols AX, BX, CX, DX, SI, DI, BP, and SP. Instructions that operate in 8-bit mode reference only the 8-bit data register fields labeled AH, AL, BH, BL, CH, CL, DH, and DL. (The 8- and 16-bit register fields and their names duplicate the programming model of the 8088, 8086, and 80286 microprocessors.)

### **Program Control Registers**

The 32-bit extended instruction pointer (EIP) indicates the instruction currently being executed. Values derived from the EIP control the instruction prefetch unit and keep track of

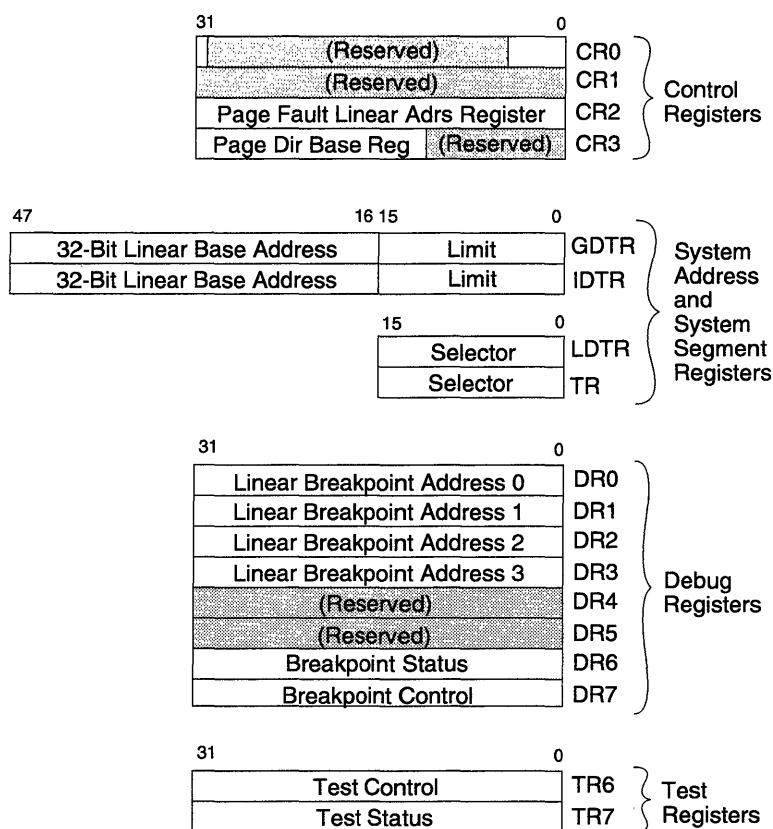


Figure 3-2. The 386 system programming model.

instructions in earlier stages of the prefetch pipeline. Program-redirection instructions, including jumps, calls, and conditional branches, all allow target addresses to lie anywhere within the full 4-Gbyte processor address space.

The extended flags register (EFLAGS) contains a combination of status flags and control fields. This register is a superset of the program status word implemented by previous members of the product line, starting with the 8088 and 8086. Certain fields within the EFLAGS register are available for use by all software; protection mechanisms make other fields visible only to software operating at the highest privilege level.

### System Registers

In addition to the user-accessible registers described above, the 386 architecture defines a number of system registers and memory-based data structures. These resources are not normally accessible to applications programs, and can be considered part of a separate, system-level programming model. These registers are shown in Figure 3-2.

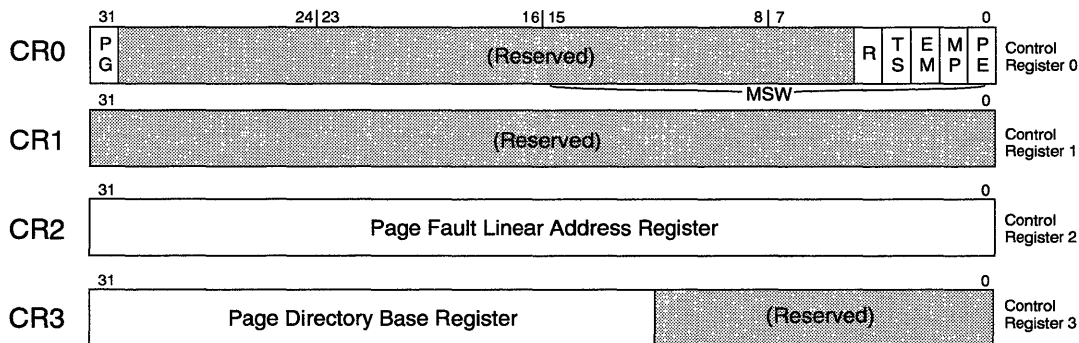


Figure 3-3. The 386 control register field definitions.

### Control Register Functions

Control registers CR0 through CR3 determine the CPU operating mode and execution options and the base address within system memory of various critical data structures, as detailed in Figure 3-3.

Control register CR0 contains five control and status flags. Functions controlled by these bits are as follows:

- PG and PE control the processor operating mode and enable 386 memory paging and protection features. To provide full compatibility with operating system software developed for the 8088, 8086, and 80286 processors, these bits can be initialized to limit processor operation to the simpler instruction set, reduced precision, and less capable addressing modes implemented by earlier devices.
- EM and MP control the handling of floating-point operations in the instruction stream.
- TS is set when task switches occur, and can be used in conjunction with bits EM and MP to reduce the time needed to save the state of tasks that don't use the floating-point unit.

Control register CR2 preserves the 32-bit linear address of the attempted memory reference that produced the most recent page fault. Operating system page-fault handlers use this address to locate the offending page, swap a new page into physical memory from disk, and update the memory-based page tables.

Control register CR3 contains the physical base address of the page directory table for the currently executing task. Page tables must be aligned at 4-Kbyte page boundaries, so only the high-order 20 bits need to be specified.



**Breakpoint Registers** To allow full-speed, noninvasive software instrumentation and debugging, the 386 architecture defines four hardware address comparator registers, labeled DR0 through DR3 in Figure 3-2. Each can be programmed with an arbitrary linear memory address. When executing the instruction at that address, or referencing the data value at that address, an immediate hardware trap is invoked. Two additional registers (DR6 and DR7) contain control and status information for the breakpoint logic. This facility is especially powerful, since it does not affect program execution speed, and it can detect instruction execution events not visible from outside the device.

## 3.2 Integer Instructions

The 386 architecture defines a fairly extensive integer instruction set. Integer and control operations divide into eight categories:

- Data transfer
- Arithmetic and logic
- Program control
- Operating system and high-level language support
- String operations
- Bit manipulation
- Shift/rotate
- Machine control

These operations are summarized in Table 3-1. (Floating-point instructions are discussed later in this chapter.) Many of the table entries identify a whole class of individual instructions. A single mnemonic can produce several different instruction forms, depending on the size of the operands and whether the source and destination parameters occupy a register, memory location, or field within the instruction itself.

Instructions can operate on 0, 1, 2, or 3 operands, and vary in length depending on the operation performed, its addressing modes, and the precision of any embedded constants. Instructions with implicit operands (i.e., set or clear a bit in the EFLAGS register, or return from subroutine) take just a single byte. Simple program branches take just two bytes, and stack-based memory accesses usually need only three.

DATA TRANSFER	PROGRAM CONTROL	STRING OPERATIONS
Move operand Move w/ zero extension Move w/ sign extension Convert operand width Exchange operands Load effective address Load segment pointer Load flags register Store flags register Push operand Pop operand Push flags register Pop flags register Push all registers Pop all registers Input port value Output port value Table look-up	Unconditional jump Jump if above Jump if above or equal Jump if below Jump if below or equal Jump if greater Jump if greater or equal Jump if less Jump if less or equal Jump if positive Jump if negative Jump if equal Jump if not equal Jump if carry set Jump if carry clear Jump if overflow set Jump if overflow clear Jump if parity even Jump if parity odd Jump if count equals zero Loop Loop while equal Loop while not equal Call procedure or task Interrupt Interrupt if overflow Return from procedure/task Return from interrupt	Move string Input string Output string Compare string Scan string Load string Store string Repeat string instruction Repeat if equal Repeat if not equal
		BIT MANIPULATION
		Bit test Bit test and set Bit test and reset Bit test and complement Bit scan forward Bit scan reverse Insert bit string Extract bit string
		SHIFT/ROTATE
		Logical shift left Logical shift right Arithmetic shift left Arithmetic shift right Double shift left Double shift right Rotate left Rotate right Rotate left w/ carry Rotate right w/ carry
		MACHINE CONTROL
		Load machine status Store machine status Enable interrupts Disable interrupts Set direction flag Clear direction flag Coprocessor escape Await FPU completion Lock bus Halt
ARITHMETIC & LOGIC	OS & HIGH-LEVEL LANGUAGE SUPPORT	
Add operands Add with carry Increment operand ASCII adjust for addition Decimal adjust for addition Subtract operands Subtract with borrow Compare operands Decrement operand Negate operand ASCII adjust for subtract Decimal adjust for subtract Multiply ordinal Multiply integer ASCII adjust after multiply Divide ordinal Divide integer ASCII adjust before divide Logical AND operands Logical OR operands Logical XOR operands Logical NOT operand Set carry flag Clear carry flag Complement carry flag Test operands Copy condition codes	Store global descriptor Store local descriptor Store interrupt descriptor Store task register Load global descriptor Load local descriptor Load interrupt descriptor Load task register Adjust privilege level Load access rights Load segment limit Verify segment rights Check array bounds Setup param block Leave procedure	

Table 3-1. The 386 integer instruction set summary.

More complex forms (i.e., those that override default segment and operand-size assumptions and contain embedded constants and address-offset fields) can take as many as 15 instruction bytes. In most programs, instructions average between three and four bytes in length. Programs for the 386 architecture have no alignment restrictions, so instructions can begin at any byte address.

A generic 386 instruction appears as shown in Figure 3-4. Each begins with a series of up to four optional “prefix” bytes.

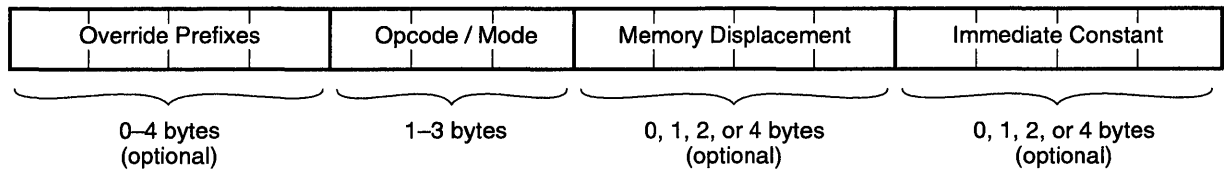


Figure 3-4. Variable-length instruction formats.

Next comes the primary opcode field, containing one, two, or three contiguous bytes. Various fields within the opcode bytes determine the size and format of the opcode itself, the operation to be performed, its precision, any working registers used as operands, addressing modes and pointers needed for memory-based operands, and the size and format of any remaining instruction fields.

After the opcode comes an optional address displacement constant, from one to four bytes long. If the instruction uses a constant as a source operand, the initial fields are followed by a numeric constant up to four bytes long. Instructions have no alignment restrictions: any prefix, opcode, or constant field can begin at any memory address.

### Memory Addressing

No matter how many registers an architecture provides, most variables referenced by any computer program must reside in external system memory. The number of variables that fit on chip is obviously limited. Operating systems and application programs spend most of their time processing memory-based operands. Character strings, data arrays, linked lists, stacks, and other complex structures must reside in memory, so that they can be referenced indirectly, with address calculations performed at run time.

The more sophisticated the addressing modes defined by an architecture, and the greater the amount of on-chip hardware devoted to performing such address calculations, the more effectively the processor will be able to deal with such structures, and the smaller and quicker its programs.

The 386 architecture allows a wide variety of operand-addressing modes that allow efficient yet flexible access to memory-based data. Each instruction that references memory may specify up to five components—a memory segment selector, an optional working register, a second optional working register shifted by from zero to three bits, and an optional in-line signed 8-, 16-, or 32-bit constant (see Figure 3-5).

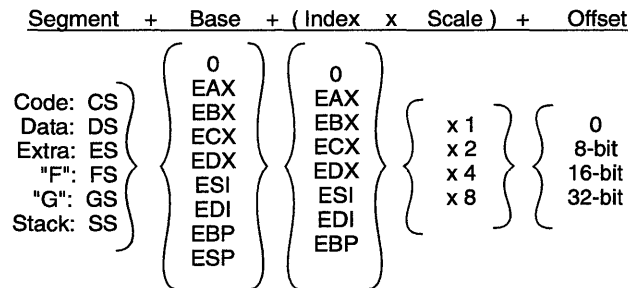


Figure 3-5. Memory operand address components.

These five components can be mixed and matched in various ways; Table 3-2 illustrates several examples, with an example of the standard assembly language syntax used for each.

Not only are these modes available for memory load and store instructions, but they can also be incorporated into most arithmetic and logical operations, and can specify either a memory-based source operand, or a memory operand used as both source and destination. A single 386 instruction can thus combine operations that would consume a series of discrete instructions on reduced-instruction-set machine.

Mode Designation	Address Components	Intel Assembly Language Syntax
Direct	Displacement only (8-, 16-, or 32-bit literal)	dword ptr 100
Register-indirect	Base only (16- or 32-bit register variable)	byte ptr [ebx]
Indexed	Scaled variable index only (scale factor = 1, 2, 4, or 8)	word ptr [ebx*4]
Based	Displacement + base	byte ptr [ebp - 30]
Indexed-displacement	Displacement + scaled index	dword ptr 100[ebx - 4]
Based-indexed	Base + scaled index	dword ptr [eax + ecx*8]
Based-indexed with displacement	Displacement + base + scaled index	dword ptr 200[eax + ecx*8]

Table 3-2. Memory operand addressing modes.

**Segmentation** The 386 architecture supports an optional memory address segmentation scheme. The processor address space is divided into a set of discrete regions, each of which is allocated to code (program instructions), data, or stack storage. Each segment

has associated with it a 32-bit base address and a 32-bit segment size, allowing each segment to begin anywhere within the 4-Gbyte address space and to fill any amount of memory.

Whenever an executing program references memory—be it for an instruction, stack operand, or data—the address value defined by the instruction is added to the base address value corresponding to the memory segment being referenced (see Figure 3-6). Programs that use segmentation thus need only concern themselves with offset of a memory value within a segment, not its full absolute address.

There are six segment selector registers, each 16 bits wide. An executing task may access up to 8K global segments for system-wide code and data structures plus an additional 8K local segments for task-specific data resources. Whenever a program modifies any of the segment registers, the processor automatically loads the internal base, limit, and attribute registers corresponding to the new segment-selector value.

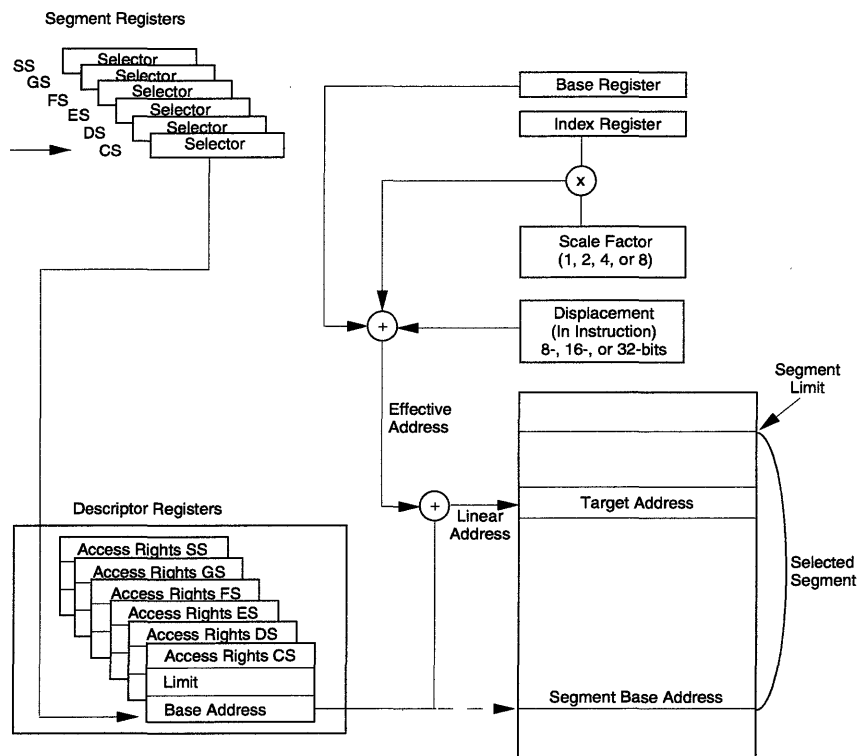


Figure 3-6. Segmentation base and offset address computation.

With a different value in each segment selector, up to six segments can be active at a time. The segment involved in each data transfer is selected automatically, depending on the transfer type. Instruction code is always retrieved from memory selected by the code segment (CS) register. Stack-based parameters and local variables always occupy memory determined by the stack segment (SS) register. By default, data addresses act as offsets into memory selected by the data segment (DS) register, though instructions with explicit segment overrides can let data reside in the code segment, stack segment, or any of three alternate data segments selected by registers ES, FS, and GS.

Operating systems use segmentation in different ways. Sixteen-bit operating systems such as MS-DOS, Windows, and OS/2 use segmentation to extend program address space. Protected multitasking operating systems such as OS/2 use the full power of segments to share the same memory image of an application program among several users, reducing system RAM requirements.

Some 32-bit operating systems, notably UNIX, prefer a large, “flat” (nonsegmented) address space. In such situations, each of the segment registers can be initialized to enable the same base address, with a segment size equal to the total available memory. This effectively disables the segmentation hardware.

### **Memory Paging**

The 386 architecture supports an optional built-in memory paging system enabled by setting the PG bit in control register CR0. When paging is enabled, the processor automatically translates every linear instruction or data address to a physical address, according to translation values stored in memory-based tables. Following the standard conventions of supermini and mainframe computers, page tables are arranged in a two-level hierarchy, as shown schematically in Figure 3-7.

Each task can have a separate page-table directory; control register CR3 indicates the base address, or root, of the directory for the current task. Each directory contains entries for up to 1,024 page tables, each of which may in turn describe the physical address mapping of up to 1,024 pages. Memory pages are 4K bytes long, so each page table can map up to four megabytes, and each page directory controls mapping for the full four-gigabit linear address space.

Various fields of a linear address are interpreted as offsets into page directories, tables, and pages. In practice, the trans-

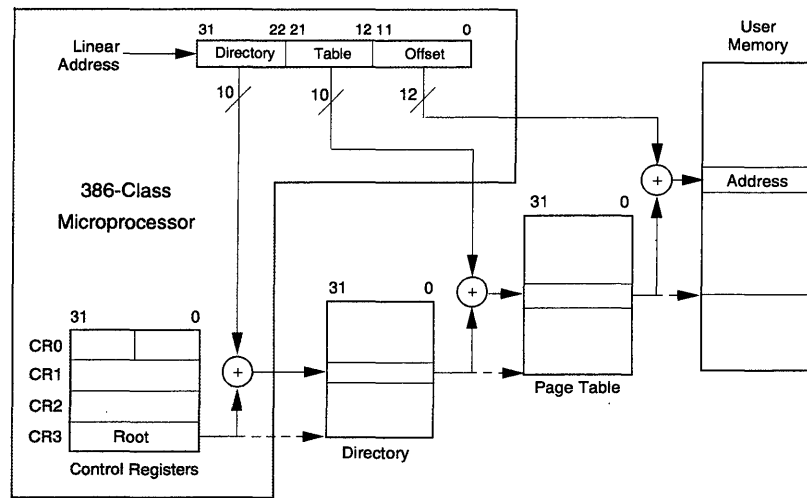


Figure 3-7. PMMU address translation mechanism.

lation tables in memory are only rarely consulted. Mapping information for previously accessed pages is cached in a special Translation Lookaside Buffer (TLB). The size of the TLB is generally different for each implementation; see **Chapters 6 through 13** for details on specific devices.

When the data needed for a memory address translation cannot be found in the TLB, microcoded control routines automatically retrieve directory and page-table data as necessary to update the least recently used TLB entry.

### 3.3 Floating-Point Architecture

Early personal computers were used primarily for word processing, accounting, database retrieval, and applications for which simple integer arithmetic was perfectly adequate. Requirements for floating-point arithmetic operations were simulated by slower integer math.

While engineering simulations, 3-D graphics, and scientific visualization have always demanded fast floating-point arithmetic, even standard business applications now rely more and more on floating-point performance. Desktop publishing programs use trigonometry and fractional arithmetic to rotate and scale typeface fonts and graphics. Financial analysis software needs the speed of a math coprocessor to recompute large, complex spreadsheets and accelerate compound inter-

est projections. Even operating systems now demand floating-point hardware to support graphical user interfaces.

Among 386-class CPUs, floating-point capabilities are supported by an off-chip coprocessor such as the i386SX or i387DX. Certain 486- and Pentium-class CPUs support comparable capabilities via an on-chip FPU (see **Part III** and **Part IV** of this report for details).

### FPU Instruction Set

The floating-point instruction set used by 387-series FPUs directly supports a variety of basic data-transfer, format-conversion, and arithmetic operations, as well as transcendental operations such as sine, cosine, tangent, log, power, and arctangent, all with 80 bits of precision. These instructions are summarized in Table 3-3.

Floating-point instructions occupy the same instruction stream as conventional integer instructions. Execution of floating-point operations proceeds in parallel with integer operations, so the computation of new operand addresses and decisions concerning program flow can continue unimpeded.

DATA TRANSFER	BASIC ARITHMETIC	TRANSCENDENTAL FUNCTIONS
Load real operand Load integer operand Load BCD operand Store real operand Store integer operand Store BCD operand Store real and pop Store integer and pop Store BCD and pop	Add real operands Add integer operand Subtract real operands Subtract integer operand Multiply real operands Multiply integer operand Divide real operands Divide integer operand Subtract real reversed Subtract integer reversed Divide real reversed Divide integer reversed Add real and pop Add integer and pop Subtract real and pop Subtract integer and pop Multiply real and pop Multiply integer and pop Divide real and pop Divide integer and pop Sub real reversed and pop Sub int reversed and pop Divide real reversed and pop Divide int reversed and pop Partial remainder Partial remainder (IEEE) Round real to integer Absolute value Change value sign Square root real operand Scale real operand Examine operand Extract real components	Cosine real Sine real Sine and cosine Partial tangent Partial arctangent $(2^x) - 1$ $(x) * \log_2(y)$ $(x) * \log_2(y+1)$
CONSTANT INITIALIZATION Load +0.0 Load +1.0 Load Load $\log_2(10)$ Load $\log_2(e)$ Load $\log_{10}(2)$ Load $\log_n(2)$		FPU MACHINE CONTROL Initialize FPU Store status word Load control word Store control word Clear exceptions Store environment Load environment Save state Restore state Increment stack pointer Decrement stack pointer Free stack operand No-op
DATA COMPARISONS Compare with real Compare with integer Compare with zero Unordered compare Compare real and pop Compare real and pop twice Compare integer and pop Compare int. and pop twice Unordered compare and pop Unord comp and pop twice		

Table 3-3. Floating-point instruction set summary.



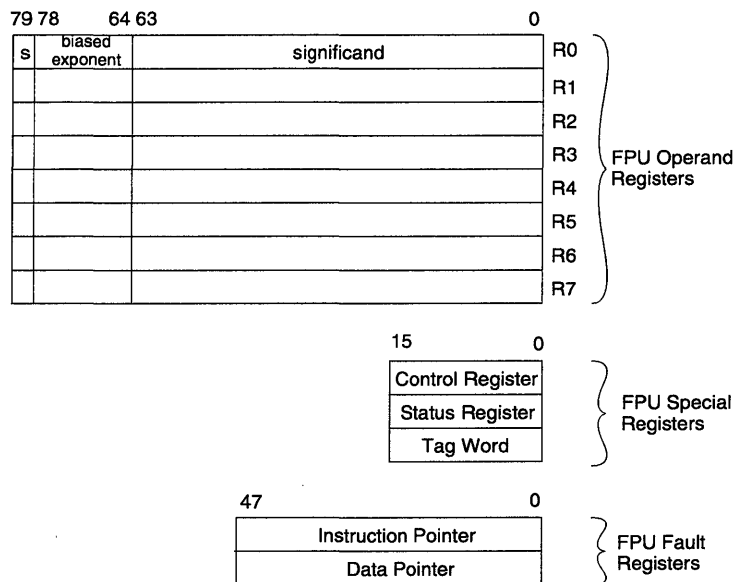


Figure 3-8. Floating-point programming model.

### FPU Operand Registers

The 387 floating-point model defined by 386, 486, and Pentium processors establishes a separate set of registers for floating-point operands as well as separate control, status, and register-tag words (see Figure 3-8). The eight floating-point operand registers are each 80 bits wide, organized as a sign bit, a 15-bit biased exponent, and a 64-bit significand. This format can represent numbers between  $\pm 10^{4932}$ . Specially coded data patterns represent plus and minus infinity, zero, and undefined values (not-a-number, or NaN).

Operand registers are organized such that they can be referenced two ways. The “natural” register organization is as a circular stack. A special top-of-stack counter (TOP) keeps track of which registers are in use and is automatically updated by operand push and pop instructions. Stack-based addressing allows a short form of each FPU operation. These instructions implicitly operate on the top one or two stack elements and return the result to the top-of-stack.

A three-bit field within the status register lets application programs and the operating system determine the internal stack state. Stack overflows and underflows cause an FPU exception trap, so software can maintain an arbitrarily large stack in memory, saving and retrieving operands as needed.

Alternatively, the eight operand registers can be treated as a conventional set of registers, each with an explicit three-bit selector code. Programs may use the top-of-stack register as an implicit accumulator, and explicitly designate which other register to use as the second source, much like the integer execution unit's register-addressing model.

### **FPU Control and Status Registers**

The FPU control word adjusts operation of certain aspects of the floating-point computations. Option fields in the control register regulate such issues as the rounding modes and precisions used, and how the FPU should treat each of several types of exceptions that might occur. In addition to the operand stack pointer, the status register contains exception flags that indicate the results of previous computations. The register-tag word shows whether each operand register is empty or contains valid or faulted data.

### **FPU Exception and Trace Registers**

Since execution of FPU instructions proceeds in parallel with integer operations, execution faults can occur long after the offending instruction has been initiated. In the meantime, the main instruction pointer may have changed considerably. In multiprogramming environments, the process containing the offending instruction may even have been switched out. Debugging such programs and dealing with exceptions that arise at run time can therefore be quite complicated.

Two additional status registers inside the 386 CPU help keep track of such errors. The FPU instruction pointer retains a copy of the full 48-bit virtual address of each floating-point instruction as it executes. The data pointer keeps the 48-bit virtual address of any related memory-based operand. Examination of these registers by a floating-point exception- or trap-handler makes it much easier for an operating system to back out of such errors when they occur.

### **FPU Memory-Based Data Formats**

Within system memory, FPU variables may reside in a variety of formats, according to the needs of the program.

- Single-precision (32-bit) and double-precision (64-bit) floating-point formats are compatible with IEEE STD 754.
- Extended-precision (80-bit) floating-point format matches the exact representation of the internal FPU registers. By including extra "guard bits" for both the exponent and mantissa, this format eliminates rounding errors that might otherwise occur when saving intermediate calculation results in memory.

- Various binary integer formats (16, 32, and 64 bits) provide a compact representation for integer constants and simplify data exchanges between the FPU and the main integer execution unit.
- Packed binary-coded decimal (18-digit BCD) format is often used within the financial processing community to represent monetary amounts, since the accumulation of floating-point rounding errors can inadvertently lead to the creation and annihilation of money. BCD strings are the standard internal data representation used by COBOL interpreters, and the use of BCD formats can speed conversion between ASCII and floating-point representations.

Figure 3-9 shows the layout of each of these memory formats. Whatever the external memory representation, however, each of these formats exists internally as an 80-bit floating-point variable. The FPU instruction that loads each operand type automatically converts it from its memory-based format to the extended register precision. The conversion is always exact,

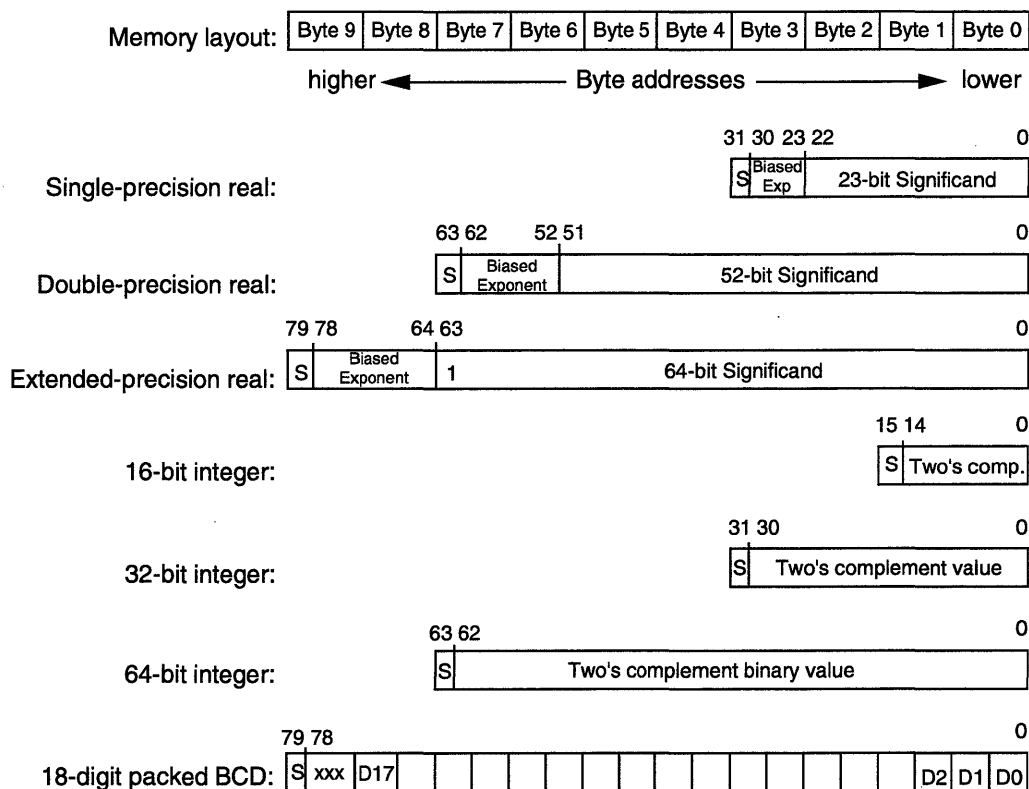


Figure 3-9. Memory-based data formats for floating-point data types.

so no rounding errors occur. Each of the binary and BCD integer formats “fits” within the significand field of the registers.

Instructions that store floating-point registers in external memory automatically convert data from the internal format to the appropriate memory-based format, performing rounding operations as necessary. Rounding errors may occur when 80-bit register values are truncated to fit within 32-bit or 64-bit memory formats, but register-based data may safely be saved and restored in extended-precision format with no loss of precision.

### 3.4 CPU Operating Modes

A 386-based microprocessor can operate in several different modes (see Table 3-4). Following a hardware reset, the device starts out in Real Mode, in which it behaves like a fast, unprotected 8086 with a one-megabyte address space. 32-bit-savvy operating systems use Real Mode just to build the required operating system data structures in memory and then switch to Protected Mode for 80286 emulation (how *passé!*) or to the 32-bit Paged-Protected “native” mode (now *de rigueur*).

Mode	Processor Operation Semantics
Real Mode	Exact 8088/8086 emulation
Protected Mode	Exact 80286 emulation, plus 32-bit extensions via prefix codes and code-segment descriptor settings
Paged Protected Mode	Protected mode plus paging enabled underneath segmentation
Virtual 8086 Mode	8088/8086 emulation within virtual memory paging and protection system

Table 3-4. CPU operating modes.

Within the 386 architecture’s protected multitasking environment, an individual task may be dispatched by the operating system to run in Virtual 8086 Mode. This establishes an environment in which the processor behaves like an 8088 or 8086, with the same precision of operation, addressing modes, and memory segmentation scheme as its 16-bit forerunner.

Virtual memory paging is still in effect, so each Virtual 8086 Mode program may be assigned its own local address space anywhere within the 4-gigabyte physical address space. The operating system may allocate a full megabyte of physical

memory to a Virtual 8086 Mode task, or may assign it a much smaller area and swap code and data pages into physical memory as needed.

The processor reenters the native, 32-bit protected mode when a task signals an exception or when a hardware or software interrupt occurs, making the full resources of the architecture available to interrupt and exception handlers provided by the operating system. The exception handler determines how to process the offending interrupt and I/O instructions encountered within a Virtual 8086 Mode task.

An arbitrary number of tasks can thus execute simultaneously in a 386-based system, some in native mode, others in Virtual 8086 Mode. Each would execute within its own memory region, with each prevented from corrupting the memory spaces of the others or of the operating system.

### **Interrupt and Exception Processing**

The 386 interrupt mechanism supports 256 vectors, each of which may be invoked by external hardware, internal software, or both. External control circuitry can detect and prioritize service requests from peripheral components and I/O subsystems. In addition, the exception-handling facilities of the 386 architecture operate by invoking predefined interrupt-handler routines in response to unexpected or irregular processor conditions.

## **3.5 For More Information...**

More detailed technical information on the x86 microprocessor architecture may be found in the following publications:

### **Vendor Publications**

- 1: *80386 System Software Writer's Guide*. Intel Corporation, 1988, order #231499-001.
- 2: *Microprocessors Data Book Volume I: Intel386, 80286, and 8086 Microprocessors*. Intel Corporation, 1994, order #230843-011.

### **Microprocessor Report Articles**

- 3: *Instruction Set Design Is Crucial*. Brian Case, MPR vol. 2 no. 7, 7/88, pg. 8. (Editorial.)
- 4: *Tredennick Presents the Case for CISC*. MPR vol. 2 no. 11, 11/88, pg. 16. (Feature article.)

- 5: *It's Not RISC vs. CISC—It's New vs. Old*. Nick Tredennick, MPR vol. 3 no. 2, 2/89, pg. 12. (*Editorial.*)
- 6: *A Tale of Two Architectures\**. Michael Slater, MPR vol. 4 no. 3, 2/21/90, pg. 12. (*Feature article.*)
- 7: *Why Programmers Hate the 8086 and 286\**. John Levine, MPR vol. 4 no. 13, 8/8/90, pg. 10. (*Feature article.*)
- 8: *386 Architecture Overcomes 286 Defects\**. John Levine, MPR vol. 4 no. 14, 8/22/90, pg. 6. (*Feature article.*)
- 9: *Intel Lays Out x86 Roadmap\**. MPR vol. 5 no. 13, 7/24/91, pg. 10. (*Feature article.*)
- 10: *Meridian Strikes Deal With UMC*. MPR vol. 5 no. 17, 9/18/91, pg. 4. (*Most Significant Bits item.*)
- 11: *Intel Announces New Interrupt Controller*. MPR vol. 6 no. 15, 11/18/92, pg. 5. (*Most Significant Bits item.*)

#### Other Technical References

- 12: *80386 Technical Reference*. Edmund Strauss, Brady Books, 1987, ISBN 0-13-246893-X.
- 13: *Computer Architecture: A Quantitative Approach*. John Hennessy and David Patterson, Morgan Kaufmann Publishers, 1990, ISBN 1-55860-069-8. (*The definitive textbook on modern computer architecture design methodologies.*)
- 14: *Microcomputer Systems: The 8088 Family*. Yu-cheng Liu and Glenn Gibson, Prentice-Hall, 1986. (*textbook on Intel's 8086, 80186, 80286.*)
- 15: *Microprocessor-Based Design*. Michael Slater, Prentice-Hall, 1989, ISBN 0-13-582248-3.
- 16: *Microprocessors: A Programmer's View*. Robert Dewar and Matthew Smosna, McGraw-Hill, Inc., 1990, ISBN 0-07-016638-2.
- 17: *PCI System Architecture*. MindShare Press.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)

# Part II: The Players

---

While this report is certainly technology-intensive, technical merit alone will not guarantee a product's success. Business issues related to a product's supplier are as important as its transistor count, process geometry, and line pitch. A microprocessor vendor's track record, history, sales volume, profitability, and stability should all be considered in selecting which device to use for an anticipated design.

Consider, for example, the sad case history of Chips & Technologies. In 1988 C&T undertook a massive project to develop a line of 386-compatible microprocessors. Three years and \$50 million later, C&T took the wraps off a number of new devices; unfortunately, the market winds had shifted, the chip-set business had gone south, and C&T no longer felt it was in its best interest to actively pursue the 386 CPU market.

Several of C&T's preannounced parts failed to materialize; while the others were sampled but quietly withdrawn from the market. Designers who committed to C&T devices soon found themselves hastily redesigning their systems to use parts from more stable vendors.

**Part II** of this report examines the seven announced vendors of x86 processors from a business perspective. All are covered in a single chapter:

## **Chapter 4: Vendor Profiles**





# 4

## Vendor Profiles

---

This chapter profiles each of the major vendors competing in the 386 and 486 marketplace, including their history, business strategies, financial health, and probable future directions.

### 4.1 Intel

It is no understatement to say that Intel Corp. is by far the most powerful force today in PC hardware. The company's ability to adapt to the prevailing market conditions has been a key factor in attaining its status as the predominant microprocessor supplier (see Table 4-1)—and that ability should help the company continue its success into the next century.

During 1994 Intel completed its tenth consecutive quarter of growth, each setting a new record for both quarterly sales and income. As a result, the company is known for its financial stability (see Table 4-2), and in 1993 became the largest semiconductor vendor in the world.

The 25-year-old firm has its origins in the R&D labs. The company was founded in 1968 by Robert Noyce, general manager of Fairchild Semiconductor, and Gordon Moore, Fairchild's R&D director. Robert Noyce is one of the two people (along with TI's Jack Kilby) credited with inventing the integrated circuit. Noyce and Moore quickly brought in Andrew Grove, the current president and CEO, as the first operations manager.

Company	Intel Corporation
Year Founded	1968
Headquarters	Santa Clara, CA
Stock Exchange/Symbol	OTC (NASDAQ): INTC
Number of Employees	29,500
Revenues <sup>1</sup>	\$9.4 billion
Net Income <sup>1</sup>	\$2.4 billion
Net Profit Margin <sup>1</sup>	25.1%
Total Assets <sup>2</sup>	\$11.3 billion
Total Liabilities <sup>2</sup>	\$3.8 billion
Shareholders' Equity <sup>2</sup>	\$7.5 billion
Stock Value (per share) <sup>3</sup>	\$61.5
Shares Outstanding <sup>3</sup>	419 million
Total Market Valuation <sup>3</sup>	\$25.7 billion
Payout Ratio (dividends/earnings) <sup>1</sup>	3.7%

Table 4-1. Intel company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 12/93

<sup>2</sup> as of 12/25/93

<sup>3</sup> as of 9/30/94

R&D was the basis of the company's initial success. A host of "firsts" credited to the company include the first commercially viable DRAMs, SRAMs, PROMs, and EPROMs in the industry. As a sign of its technological diversity, Intel also developed the Schottky bipolar process technology used for high-speed TTL, the first high-capacity magnetic bubble memories, the first custom graphics chip for video games, and the first single-chip digital signal processor. Microma, an Intel subsidiary, was even the first company to build digital watches with LCD displays; company officers decided to liquidate Microma when they realized they had never really intended to get into the jewelry business.

In recent years, however, Intel has devoted its emphasis to various high-performance MOS and CMOS processes. Of course, the company was a pioneer in the microprocessor arena as well, but its early days were dominated by memory devices. In fact, about 80% of Intel's sales in the late '70s were memory devices. Today, the numbers are basically reversed, with microprocessors and related products representing more than 80% of sales.

Although the company's success in its early days was due to innovations in the R&D lab, the key to its fortunes in the '80s and '90s was a chip crafted not by R&D but by marketing: the

	'89	'90	'91	'92	'93
Revenues (millions)	\$3,127	\$3,921	\$4,779	\$5,844	\$8,782
Change in Revenues	8.8%	25.4%	21.9%	22.3%	50.3%
Net Income (millions)	\$391	\$650	\$819	\$1,067	\$2,295
Change in Net Income	-13.7%	66.3%	25.9%	30.3%	115.2%
Profit Margin	12.5%	16.6%	17.1%	18.2%	26.1%
R&D Expenditures (millions)	N.A.	N.A.	\$619	\$780	\$970
Return on Equity	15.3%	18.1%	18.5%	19.6%	30.6%
Dividends per Share	\$0.00	\$0.00	\$0.00	\$0.05	\$0.20
Earnings per Share	\$1.04	\$1.60	\$1.96	\$2.49	\$5.20
Book Value per Share	\$6.91	\$8.99	\$10.83	\$13.01	\$17.01
Price/Earnings Ratio (high)	17.3	16.3	15.1	18.4	14.3
Price/Earnings Ratio (low)	11.0	8.8	9.6	9.3	8.2
Share Price High	\$17.88	\$26.00	\$29.63	\$45.75	\$74.25
Share Price Low	\$11.63	\$14.00	\$18.88	\$23.25	\$42.75
# of Shares Outstanding (millions)	369.0	399.3	407.8	418.6	441.0

Table 4-2. Intel financial results ('89-'93). (Source: company reports; Schwab investment reports)

8086. The travails of designing the chip and "Operation Crush," pivotal marketing campaign to convince major companies (including IBM) to adopt the part in the years that followed, has been well documented. (Reference 24 at the end of this chapter gives former Intel Vice President William Davidow's perspective of the 8086 marketing wars.)

Operation Crush is key to any discussion of Intel, not only because of its impact on the company but also because of its impact on the entire PC industry that sprouted up around the x86 architecture. The strategy and tactics of Operation Crush remain strongly entrenched at Intel, more than a decade later. And they prove useful in explaining and predicting Intel's actions as it advances its efforts to maintain its market dominance and extend the reach of its computer component sales.

With so much at stake in the CPU market, Intel never hesitates to spend money to isolate and crush the competition. For example, Intel established 11 alternate sources of the 8086 and 8088 to help garner the IBM account. Once the PC market was firmly dependent on the architecture, however, Intel cut the number of authorized second sources for the 80286 to just three—and to zero for the 386 and beyond.

When AMD, one of the three approved sources of the 80286, tried to extend the life of the chip by introducing 80286s with clock rates faster than Intel's stopping point of 12 MHz, the company introduced the i386SX—the so-called “AMD killer.” The i386SX eventually did kill the 80286, but not AMD.

Intel currently perceives two threats to its dominance of the PC microprocessor market: x86 clones from below and RISC microprocessors from above.

Intel has spent the past three years trying to isolate and destroy its x86 competitors. Actually, on the legal battleground, the company has been doing it for much longer (see **Chapter 16: Legal Issues**). During this time, Intel's tactics have served to dramatically up the ante for competitors to join in this market while at the same time limiting competitors' chances for success.

Examples include:

- Direct-marketing campaigns targeting end users, increasing retail demand for Intel-based PCs. Intel has reportedly spent more than one billion dollars to date to establish greater brand-name recognition.
- The “Intel Inside” campaign, which sweetens the pot with advertising kickbacks for all-Intel OEMs.
- The “Intel Inside” campaign, part 2: Flooding the marketing communications channels with the “Intel Inside” campaign gives a none-too-subtle message to PC OEMs that Intel has virtually limitless money and muscle to throw behind its architecture—and OEMs have to question whether competitors can match it.
- OverDrive, which adds upgradability to OEM's CPU purchase checklist.
- An expanded number of parts for niche markets, thereby limiting the success of any one competitor's part.
- Announcements of massive R&D budgets, capital expansions, and planned x86 introductions to instill fear in competitors and foster questions about competitors' commitment to this market in the minds of OEMs.

RISC processors have been around in the workstation market for more than five years. So far these processors have had virtu-

---

ally no impact on Intel's bottom line. But today there is a new development that changes the equation: the Windows NT operating system from Microsoft. The portability of the OS and applications written for it could (in theory) open the NT PC market to RISC competition.

The inability to trademark numbers and the ensuing confusion caused by competitors' appropriation of Intel's 486 product numbering scheme for less capable products led Intel to name its latest microprocessor Pentium rather than the 586, as had been expected. The name also helps to distance the chip from earlier members of the x86 family, which had been perceived as inferior to RISC competition, and lets Intel align itself more readily with competing RISC processors that sport sexy-sounding names like PowerPC, Alpha, MIPS, and SPARC. IBM, Digital Equipment, Silicon Graphics, and Sun are clearly gearing up to give Intel a run for the desktop market. And Intel, too, is readying for battle.

## 4.2 Advanced Micro Devices

Like Intel, AMD spun off from Fairchild Semiconductor. Unlike Intel, however, the company was born more out of sales and marketing than raw technology. AMD built its reputation and sales as an alternate-source supplier of industry-standard components (see Table 4-3).

Company	Advanced Micro Devices, Inc.
Year Founded	1969
Headquarters	Sunnyvale, CA
Stock Exchange/Symbol	NYSE: AMD
Number of Employees	12,000
Revenues <sup>1</sup>	\$1.75 billion
Net Income <sup>1</sup>	\$252 million
Net Profit Margin <sup>1</sup>	14.4%
Total Assets <sup>2</sup>	\$1.93 billion
Total Liabilities <sup>2</sup>	\$0.58 billion
Shareholders' Equity <sup>2</sup>	\$1.35 billion
Stock Value (per share) <sup>3</sup>	\$29.75
Shares Outstanding <sup>3</sup>	92 million
Total Market Valuation <sup>3</sup>	\$2.7 billion
Payout Ratio (dividends/earnings) <sup>1</sup>	0%

Table 4-3. AMD company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 12/93

<sup>2</sup> as of 12/26/93

<sup>3</sup> as of 9/30/94

AMD quickly developed a reputation as an aggressive marketer with quality parts at competitive prices. The company grew relatively unconstrained until the DRAM glut that began in late 1984. Unlike Intel, the technology powerhouse that turned to marketing for expansion in the late 1970s, the marketing-oriented AMD turned to technology to turn its fortunes around during its troubled times in the mid-'80s.

In late 1985, the company unveiled the Liberty Chip Campaign, which showcased an accelerated R&D effort with the end goal of introducing a new product every week for a year. Few of those products amounted to much in the marketplace, although some formed the foundation for parts that are successful today, including Ethernet and SCSI chips.

In 1984, AMD finally realized CEO Jerry Sanders' dream of becoming a \$1 billion company. But the glow of success was short lived, as the bottom fell out of the DRAM market in '85: sales plummeted 29% from \$1.1 billion to \$795 million. AMD surpassed \$1 billion again in '88, but only after acquiring Monolithic Memories Inc., the market leader in programmable logic devices. After the buyout, sales again contracted slowly back toward the \$1 billion range until 1991, when AMD introduced its Am386 family, Intel's first competition in the 32-bit PC microprocessor market.

AMD did not initially intend to design and make its own 386; it had expected Intel to give it the rights to the 386 design database as part of a 1982 cross-licensing agreement between the two companies. Intel terminated its ties with AMD a few months before the Intel 386 was introduced. In the ensuing arbitration case, AMD fought bitterly for 386 production rights. Although Intel was censured in ensuing rulings, AMD was ultimately left to develop a 386 on its own.

AMD built the 386, but used Intel microcode and logic design. (The legal battle surrounding that issue ultimately has forced AMD to write its own microcode for 486-class products and beyond. See **Chapter 16: Legal Issues** for further details.)

AMD pioneered the market for 386-compatible processors with the Am386SX and Am386DX. Because each chip was compatible with Intel parts—the use of Intel microcode helped allay fears of incompatibility in the marketplace—AMD paved the way for Cyrix and other entrants. If Intel had been able to exploit any compatibility problems with the AMD 386 family, AMD's fortunes—and those of other vendors that followed—likely would have been much less rosy.

AMD did, however, reap benefits by being first to compete with Intel. The company arrived on the scene before Intel began carving up the x86 market into microslices, a strategy that limits the potential success of any one competitor's part. AMD announced in October of 1992 that it had shipped its 10-millionth 386. It is unlikely that any Intel competitor will be able to repeat that feat without developing a multitude of niche products. Cyrix's sales, which are significantly lower than AMD's, are a preliminary indication of that (see **Chapter 9: Cyrix 486 Microprocessors**).

AMD is clearly committed to this market. That is understandable, given the tremendous impact the parts have had on its for-

tunes. Until it introduced the 386, AMD's sales were making a slow transition to growth products, with net overall sales stuck at about \$1 billion. In the years since its 386 and 486 families were introduced, the company's annual sales have skyrocketed to more than \$1.6 billion (see Table 4-4).

	'89	'90	'91	'92	'93
Revenues (millions)	\$1,105	\$1,059	\$1,227	\$1,515	\$1,648
Change in Revenues	-1.9%	-4.1%	15.8%	23.5%	8.8%
Net Income (millions)	\$46.1	\$-53.6	\$145.3	\$245.0	\$228.8
Change in Net Income	138.9%	-216.3%	371.1%	68.6%	-6.6%
Profit Margin	4.2%	NE†	11.8%	16.2%	13.9%
R&D Expenditures (millions)	\$262	\$227	\$213	\$203	\$201
Return on Equity	6.7%	NE†	18.5%	23.4%	16.9%
Dividends per Share	\$0.00	\$0.00	\$0.00	\$0.75	\$0.00
Earnings per Share	\$0.44	\$-0.78	\$1.53	\$2.57	\$2.30
Book Value per Share	\$8.51	\$7.73	\$9.32	\$11.86	\$14.63
Price/Earnings Ratio (high)	23.9	NE†	11.6	8.4	14.3
Price/Earnings Ratio (low)	16.2	NE†	2.6	2.9	7.4
Share Price High	\$10.50	\$11.38	\$17.75	\$21.50	\$32.88
Share Price Low	\$7.13	\$3.63	\$4.00	\$7.38	\$17.00
# of Shares Outstanding (millions)	81.1	82.3	84.0	88.2	92.4

Table 4-4. AMD financial results ('89-'93). (Source: company reports; Schwab investment reports)

†NE = Negative earnings invalidates calculation

The company clearly has the marketing savvy to tough it out in this market for the long haul. AMD has been able to maintain its visibility as a player amid a sea of Intel messages appearing everywhere from billboards to computer stores to TV.

But does the company have the technological muscle to remain a long-term competitor? AMD for years has outpaced the industry in R&D expenditures. But until the x86—which represents far more D than R—sales growth has been underwhelming. In this respect, the AMD 486 family represents a critical juncture for AMD. AMD 386 sales, which have grown to represent more than one-third of AMD revenues, likely peaked during 1992 or 1993, and have undoubtedly fallen as industry demand for the 386 family evaporates.



Regarding AMD's prospects in the 486 market, the company introduced its first parts in mid-1993. That occurred somewhat earlier in the product life cycle than did the introduction of the Am386 family. This is a strong point in AMD's favor. However, Intel has subdivided the 486 market so many ways that AMD may find it difficult to compete successfully in each niche.

A key determination will depend on the level of success AMD achieves for parts it bring to market, how quickly and to what level AMD can ramp up its production volume, and whether AMD can repeat its feat of 10 million 386 units sold. AMD sold approximately 4 million units of 486-class products in 1994 and received a strong vote of confidence from Compaq in the form of an exclusivity agreement for the Am486SX2-66. Indications are that AMD understands what it takes to compete in this marketplace, though, and is expecting to be a long-term competitor.

## 4.3 Chips and Technologies

Chips and Technologies (see Table 4-5), founded in 1985, followed AMD into the 386 market in 1992. Several months later, the San Jose, Calif.-based company became the market's first casualty when it ran out of cash to support the venture.

Company	Chips and Technologies, Inc.
Year Founded	1985
Headquarters	San Jose, CA
Stock Exchange/Symbol	OTC (NASDAQ): CHPS
Number of Employees	N.A.
Revenues <sup>1</sup>	\$79.6 million
Net Income <sup>1</sup>	\$8.3 million loss
Net Profit Margin <sup>1</sup>	-10.4%
Total Assets <sup>2</sup>	\$64.8 million
Total Liabilities <sup>2</sup>	\$45.1 million
Shareholders' Equity <sup>2</sup>	\$19.7 million
Stock Value (per share) <sup>3</sup>	\$5.00
Shares Outstanding <sup>3</sup>	14 million
Total Market Valuation <sup>3</sup>	\$70 million
Payout Ratio (dividends/earnings) <sup>1</sup>	0%

Table 4-5. Chips and Technologies company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 6/93

<sup>2</sup> as of 6/30/93

<sup>3</sup> as of 9/30/94

The bottom line for C&T was that the company offered the marketplace too little too late. The company's tardiness was compounded by manufacturing problems. As well, because of the company's financial woes, PC OEMs questioned its long-term staying power—a concern that ultimately proved itself out. As a result, C&T sales and profits fell sharply after reaching their peak in 1989 and 1990 (see Table 4-6).

Chips and Technologies originally announced plans to build six devices. At the low end would be the 38600DX and 38600SX "Super386" processors that planned to be pin-compatible with the i386SX and i386DX but with an improved pipeline that promised to boost performance by 10-15% over comparable Intel chips. In addition, C&T announced the 38605DX and 38605SX, which were to feature a 512-byte instruction cache for still

	'89	'90	'91	'92	'93
Revenues (millions)	\$217.6	\$293.4	\$225.1	\$141.1	\$97.9
Change in Revenues	53.8%	34.8%	-23.3%	-37.3%	-30.6%
Net Income (millions)	\$33.0	\$29.3	\$-9.6	\$-57.4	\$-49.1
Change in Net Income	49.3%	-11.2%	-132.8%	-497.9%	14.5%
Profit Margin	15.2%	10.0%	-4.3%	-40.7%	-50.2%
R&D Expenditures (millions)	N.A.	N.A.	\$52	\$45	\$22
Return on Equity	31.3%	21.9%	NE†	NE†	NE†
Dividends per Share	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Earnings per Share	\$2.23	\$1.88	\$-.71	\$-4.00	\$-3.13
Book Value per Share	\$7.30	\$9.31	\$8.52	\$5.04	\$1.23
Price/Earnings Ratio (high)	11.4	14.0	NE†	NE†	NE†
Price/Earnings Ratio (low)	4.8	7.7	NE†	NE†	NE†
Share Price High	\$25.75	\$23.50	\$13.50	\$14.13	\$6.88
Share Price Low	\$13.75	\$5.25	\$6.50	\$3.25	\$2.75
# of Shares Outstanding (millions)	14.4	14.4	13.4	14.3	16.0

Table 4-6. Chips and Technologies financial results ('89-'93). (Source: company reports; Schwab investment reports)

†NE = Negative earnings invalidates calculation

higher performance. The latter devices were not pin-compatible with Intel parts, so system OEMs would be forced to redesign boards to make use of the parts—a factor that forced OEMs to scrutinize C&T's long-term commitment to the market more closely than they might have with pin-compatible replacements to Intel's line.

## 4.4 Cyrix

If history is any indication, Cyrix Corp. will be in the x86 arena for the long haul (see Table 4-7). Only six years old, Cyrix has already built a substantial and profitable business by competing with Intel and succeeding where others have failed. The company faces some tests, though, with its follow-on and next-generation products.

Company	Cyrix Corporation
Year Founded	1988
Headquarters	Richardson, TX
Stock Exchange/Symbol	NASDAQ: CYRX
Number of Employees	250
Revenues <sup>1</sup>	\$125 million
Net Income <sup>1</sup>	\$19.6 million
Net Profit Margin <sup>1</sup>	15.7%
Total Assets <sup>2</sup>	\$115 million
Total Liabilities <sup>2</sup>	\$31 million
Shareholders' Equity <sup>2</sup>	\$84 million
Stock Value (per share) <sup>3</sup>	\$45.25
Shares Outstanding <sup>3</sup>	18.6 million
Total Market Valuation <sup>3</sup>	\$841.6 million
Payout Ratio (dividends/earnings) <sup>1</sup>	0%

Table 4-7. Cyrix company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 12/93

<sup>2</sup> as of 12/31/93

<sup>3</sup> as of 9/30/94

Cyrix was cofounded by Jerry Rogers, president and CEO, formerly head of Texas Instruments' microprocessor division, and Tom Brightman, VP of systems engineering, who previously worked at TI, Atari, and Commodore. The VP of engineering and head of the chip design team is Kevin McDonough, a former TI Fellow.

Jim Chapman, VP of marketing, is a 10-year Intel veteran who most recently served as director of marketing for the i386SX and i386SL. Berry Cash, who was a founder of Mostek and is now a general partner of InterWest Partners III, is chairman of the board. Other board members include L.J. Sevin, also a former Mostek executive and now a partner in Sevin Rosen

Management, and Melvin Sharp, an attorney who led TI's intellectual property efforts for over a decade.

The company was privately held until 1Q93. In its first year of reporting financial results, the company posted an impressive net profit of more than \$8 million on sales of nearly \$73 million (see Table 4-8). In the first six months of 1994, net profit rose to \$14.4 million on sales of \$105 million.

	'89	'90	'91	'92	'93
Revenues (millions)	0†	25†	\$55	\$73	\$125
Change in Revenues	N.A.†	N.A.†	21%†	133%	171%
Net Income	2.9†	9.85†	\$12.7	\$8.4	\$19.6
Change in Net Income	N.A.†	N.A.†	30%†	-34%	233%
Net Profit	N.A.†	39%†	23.1%	11.5%	13.9%
R&D Expenditures (millions)	\$2.1	\$1.9	\$4.4	\$8.3	\$15.7
Return on Equity	N.A.†	99.3†	55.4†	26.7†	23.3%
Dividends per Share	\$0.00†	\$0.00†	\$0.00†	\$0.00†	\$0.00
Earnings per Share	\$8.86†	\$0.70†	\$0.78	\$0.49	\$1.06
Book Value per Share	N.A.†	N.A.†	N.A.†	N.A.†	\$14.63
Price/Earnings Ratio (high)	N.A.†	N.A.†	N.A.†	N.A.†	N.A.
Price/Earnings Ratio (low)	N.A.†	N.A.†	N.A.†	N.A.†	N.A.
Share Price High	N.A.†	N.A.†	N.A.†	N.A.†	\$40.50
Share Price Low	N.A.†	N.A.†	N.A.†	N.A.†	\$19.25
# of Shares Outstanding (millions)	3.4†	14.1†	16.3	17.2	18.4

Table 4-8. Cyrix financial results ('89-'93). (Source: company reports; Schwab investment reports)

† Estimate; company privately held until 1993  
N.A. = Data not available

The company cut its teeth against Intel with one of the first unauthorized Intel coprocessors. Cyrix's first product, an 80387-compatible floating-point unit, came to market in early 1990. Cyrix actually was the second competitor to enter the coprocessor market, but the first one—Integrated Information Technology (IIT) of Santa Clara, Calif.—encountered compatibility problems that Intel managed to exploit enough to keep IIT from becoming a significant competitor.

Cyrix's parts offered superior performance and—unlike IIT's—left no kinks in the armor for Intel to exploit. As a result of that and savvy channel manipulation, Cyrix proved to be a

formidable competitor with its 80387 work-alikes, taking significant share from Intel in that highly profitable marketplace.

Cyrix introduced the Cx486SLC and Cx486DLC in April '92. The parts were pin-compatible with Intel's i386SX and i386DX processors but offered 486-like features beyond the 386 architecture that enabled the company to market the chips as 486-class products.

Both C&T and Cyrix followed AMD to the market by nearly a year. Cyrix, however, enhanced the 386 architecture with its pin-compatible parts enough to call them 486s. The Cx486SLC and Cx486DLC feature a full 486 instruction set and a 1K cache—small compared to Intel's 486 cache, which is 8K. However, no pin-compatible 386-class processor featured cache at that time.

Intel's marketing moves to accelerate the transition from the 386- to 486-class processors actually helped Cyrix's cause, because Intel didn't have enough capacity to meet the demand it created for 486s. PC OEMs, desperate to meet the demand for 486 PCs, saw Cyrix's parts as an interim fix. They snapped up Cyrix's 486-labeled parts and thrust them into hastily reworked 386 computer designs.

As sales of the Cx486SLC and Cx486DLC began declining, Cyrix introduced the Cx486S and Cx486S2 processors, which neatly filled the price/performance gap between conventional 386 devices and the i486SX—a gap that quickly vanished when Intel slashed i486SX prices.

In 4Q93 Cyrix introduced the Cx486DX and Cx486DX2—the first Cyrix devices to be essentially equivalent in pinout and functionality to parts first introduced by Intel. Along the way, Cyrix also tested the retail aftermarket with the Cx486SRx2 and Cx486DRx2 386 system upgrade processors.

The Cyrix road map continues to include new, innovative designs. The company's real test will come with the M1, a Pentium-class superscalar processor due in early 1995. The industry is watching anxiously to see whether the company can bring the M1 into production on schedule. Cyrix's commitment to this market is clear: the company is basing its very existence on its x86 product line.

Unfortunately, Cyrix's biggest sales successes to date have come primarily from third-tier customers. The largest PC vendor to

---

adopt Cyrix products so far is AST—a large company, to be sure, but still somewhat lower in volume than IBM or Compaq.

The company's fortunes may change following the 2Q94 announcement of a cross-licensing agreement with IBM by which the Armonk behemoth would manufacture and become a second-source marketing channel for current and planned Cyrix designs. The agreement assures Cyrix a near-infinite supply of high-quality, leading-edge fab capacity and will likely prove critical to establishing the credibility of both companies' chips.

## 4.5 Texas Instruments

If any competitor can match Intel on technology breadth, manufacturing ability, intellectual property portfolio, and staying power, it is Texas Instruments (see Table 4-9). Ironically, TI is the vendor with the most to explain regarding long-term commitment to this marketplace.

Company	Texas Instruments, Inc.
Year Founded	1930
Headquarters	Dallas, TX
Stock Exchange/Symbol	NYSE: TXN
Number of Employees	59,000
Revenues <sup>1</sup>	\$8.5 billion
Net Income <sup>1</sup>	\$0.73 billion
Net Profit Margin <sup>1</sup>	8.6%
Total Assets <sup>2</sup>	\$6.0 billion
Total Liabilities <sup>2</sup>	\$3.7 billion
Shareholders' Equity <sup>2</sup>	\$2.3 billion
Stock Value (per share) <sup>3</sup>	\$68.00
Shares Outstanding <sup>3</sup>	93.6 million
Total Market Valuation <sup>3</sup>	\$6.3 billion
Payout Ratio (dividends/earnings) <sup>1</sup>	13.0%

Table 4-9. Texas Instruments company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 12/93

<sup>2</sup> as of 12/31/93

<sup>3</sup> as of 9/30/94

Texas Instruments has a rich history in semiconductors, ever since Jack Kilby was co-credited with inventing the integrated circuit, along with Robert Noyce, then working at Fairchild.

The company has stayed true to its research and development roots in the IC business ever since—even if its ability to market its creations is repeatedly called into question. The company has a full portfolio of intellectual property, including many basic IC and PC design and manufacturing patents.

TI's track record for marketing technology is as checkered as its technology foundation is rich. Few can forget the debacle surrounding the TI PC, a technological jewel in which industry standards were barely an afterthought. In the PC graphics



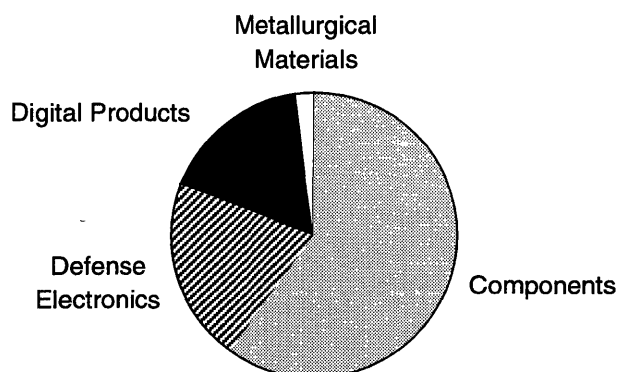


Figure 4-1. Texas Instruments' 1993 product sales ratios. (Source: company reports)

arena, TI was first to market with Microsoft Windows acceleration by adapting its 340 graphics processors to the environment—only to be driven out by optimized accelerators that cut graphics costs tremendously while offering comparable or superior performance.

For such a technology-driven company, it is curious that TI opted to buy into the x86 arena rather than designing a processor in-house, a decision that some say calls into question its long-term commitment to this market. TI negotiated foundry deals with both Chips and Technologies and Cyrix that gave them rights to market either product line and adapt the designs for future products.

TI opted for the Cyrix part—a move that in hindsight proved to be the right call—and entered the market in 1992 with TI-marked versions of the Cx486SLC and Cx486DLC. It was not until early 1994, however, that TI was able to adapt the Cyrix core for use in a design of its own.

Unfortunately for TI, the relationship with Cyrix dissolved into litigation, and TI has apparently lost the rights to new Cyrix designs, including the upcoming “M1,” and other follow-on products. The company claims to have a next-generation x86 core under development, but has yet to establish a track record for being able to successfully complete CISC processor designs of such complexity.

There is another reason to call into question TI's commitment to the market. Of all the competitors in this roundup, TI's broad range of business interests (see Figure 4-1) makes it by far the least dependent on the PC industry for revenue.

The company's product mix has in the past ranged from discrete transistors and small-scale TTL logic to consumer calculators, watches, and learning toys for kids. TI has also been very active in developing custom electronics systems for the government, and—as a point of special interest to numismatists the world over—produces the copper-clad sheet-metal stock used to mint “sandwich” coins for the U.S. and many foreign countries.

Excluding memory—PC memory sales shouldn't be affected by its x86 market presence—TI plays a small role in the PC motherboard-related components market with parts such as its PC chip sets. It does have sales to hard-disk drive, modem, graphics, and networking product manufacturers, but these—as well as sales into other PC peripherals—are largely transparent to PC OEMs.

	'89	'90	'91	'92	'93
Revenues (millions)	\$6,522	\$6,567	\$6,784	\$7,440	\$8,523
Change in Revenues	3.6%	0.7%	3.3%	9.7%	14.6%
Net Income	\$292	\$-39	\$-409	\$247	\$472
Change in Net Income	-20.4%	-113.4%	-948.7%	160.4%	92.7%
Profit Margin	4.5%	NE†	NE†	3.3%	5.6%
R&D Expenditures (millions)	N.A.	N.A.	\$527	\$470	\$590
Return on Equity	14.9%	NE†	NE†	14.3%	20.6%
Dividends per Share	\$0.72	\$0.72	\$0.72	\$0.72	\$0.72
Earnings per Share	\$3.04	\$-0.92	\$-5.40	\$2.50	\$5.07
Book Value per Share	\$24.10	\$22.46	\$19.36	\$20.92	\$25.49
Price/Earnings Ratio (high)	15.4	NE†	NE†	20.9	16.6
Price/Earnings Ratio (low)	9.3	NE†	NE†	12.0	9.0
Share Price High	\$46.75	\$44.00	\$47.63	\$52.25	\$84.25
Share Price Low	\$28.13	\$22.50	\$26.00	\$30.00	\$45.75
# of Shares Outstanding (millions)	85	82	82	85	94

Table 4-10. Texas Instruments financial results ('89-'93). (Source: company reports; Schwab investment reports)

†NE = Negative earnings invalidate calculation

More significant to TI are sales to Sun Microsystems, Hewlett-Packard, and other workstation vendors. Combine TI's nominal dependence on PC OEMs with the rising costs of competing in the x86 market—à la Intel marketing—and it's reasonable to conclude that TI probably won't be a major long-term player. The costs of competing in this market are rising, and after

recording losses in 1990 and 1991 (see Table 4-10), TI is anything but swimming in cash.

Regarding legal issues, the company found itself sucked into lawsuits filed by Intel against TI's x86 partners, Cyrix, and C&T. TI may be in the strongest position to indemnify its customers from Intel's latest legal tactic: asking for royalties on basic PC-design intellectual property from OEMs using non-Intel x86s. Due to TI's strong legal ground in technology, it is unlikely that Intel would care to attack TI in the courtrooms.

## 4.6 IBM

The biggest wild card in the x86 market is IBM (see Table 4-11). Founded in 1914, IBM is one of the world's largest and most established corporations. As a computer and peripherals manufacturer, the company is both one of the largest manufacturers and one of the largest consumers of integrated circuits in the world—and of x86 microprocessors in particular.

Company	International Business Machines Corp.
Year Founded	1914
Headquarters	Armonk, NY
Stock Exchange/Symbol	NYSE: IBM
Number of Employees	250,000 and falling
Revenues <sup>1</sup>	\$63 billion
Net Income <sup>1</sup>	\$7.3 billion loss
Net Profit <sup>1</sup>	-11.6%
Total Assets <sup>2</sup>	\$81 billion
Total Liabilities <sup>2</sup>	\$61 billion
Shareholders' Equity <sup>2</sup>	\$20 billion
Stock Value (per share) <sup>3</sup>	\$69.625
Shares Outstanding <sup>3</sup>	571 million
Total Market Valuation <sup>3</sup>	\$39.7 billion
Payout Ratio (dividends/earnings) <sup>1</sup>	-10.0%

Table 4-11. IBM company profile. (Source: company reports; Schwab investment reports)

<sup>1</sup> 12 months ended 12/93

<sup>2</sup> as of 12/31/93

<sup>3</sup> as of 6/8/94

While IBM is a truly huge conglomerate, it has fallen on hard times of late. Revenues have been essentially flat for five years, consistently hovering between \$63 billion and \$69 billion (see Table 4-12). Since 1990 the company has lost over \$20 billion.

The problem seems to be that IBM draws most of its income from markets not directly related to the microprocessor or PC industries (see Figure 4-2). As computer buyers move away from centralized mainframe computer centers to workstations and desktop PCs, sales of IBM's older product lines continue to wane, as are the ancillary services (financing, software, and maintenance contracts) that have bolstered IBM's sales and profits in the past.

	'89	'90	'91	'92	'93
Revenues (billions)	62.7	69.0	64.8	64.5	62.7
% Change in Revenues	5.1%	10.1%	-6.1%	-0.4%	-2.8%
Net Income (billions)	3.76	6.02	-5.64	-6.87	-7.99
% Change in Net Income	-31.6%	60.2%	-194%	NE†	NE†
Profit Margin	6.0%	8.7%	NE†	NE†	NE†
Return on Equity	9.8%	14.1%	NE†	NE†	NE†
Dividends per Share	\$4.73	\$4.84	\$4.84	\$4.84	\$1.58
Earnings per Share	\$6.47	\$10.51	-\$0.99	-\$12.03	-\$14.02
Book Value per Share	\$67.01	\$74.96	\$64.81	\$48.34	\$34.11
Price/Earnings Ratio (high)	20.2	11.7	NE†	NE†	NE†
Price/Earnings Ratio (low)	14.4	9.0	NE†	NE†	NE†
Share Price High	\$130.88	\$123.13	\$139.75	\$100.38	\$59.88
Share Price Low	\$93.38	\$94.50	\$83.50	\$48.75	\$40.63
# of Shares Outstanding (millions)	575	571	571	571	579

Table 4-12. IBM financial results ('89-'93). (Source: company reports; Schwab investment reports)

†NE = Negative earnings invalidate calculation

It would therefore seem that IBM and Intel would have a lot to gain from a productive relationship with each other. Indeed, the two companies have had a long and varied relationship. Big Blue began to build its own x86 processors in 1991, thanks to an agreement hammered out with Intel nearly two years before Intel introduced the 80386, and shortly before Intel announced it would not authorize alternate sources for the 386 product line. IBM negotiated for the right to make a percentage

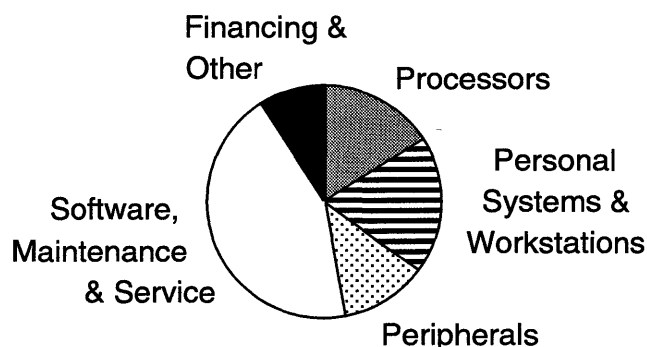


Figure 4-2. IBM 1993 product sales ratios. (Source: Schwab investment reports)

(rumored to be in the neighborhood of 20%, and increasing over time) of its own 386s in-house. IBM didn't exercise its option until nearly six years later.

IBM's x86 production has taken a new twist of late, with the company offering its latest varieties to the open market. IBM is currently promoting its 386SLC and BL486SLC2 products as well as the BL486SX/SX3, a clock-tripled x86 first introduced under the code name "Blue Lightning." IBM is limited by that same 386 agreement crafted eight years ago. For example, IBM can sell its chips only as parts of boards and subsystems—although IBM interprets the word "subsystem" to include modules as simple as a one-chip circuit board.

More important, if IBM's production is limited by the original 386 agreement, it means the company can't supply more than 20% of what it buys from Intel. In other words, it has to buy four from Intel to supply one to the open market—and this calculation incorrectly assumes that IBM isn't using any internally produced 386s in its own systems.

IBM's x86 activity has been seriously expanded during 1994 as the result of several agreements. IBM's original agreement with Intel gave it the ability to produce derivative products and a given percentage of 486s for its own use. IBM has since renegotiated its agreement with Intel whereby it can now supply a larger percentage of 486s using Intel designs for internal use only. In return, IBM gave up any claims to manufacture Pentia for its own use. IBM now offers a full line of motherboard products using the Intel-derived design.

IBM also announced alliances with both Cyrix and NexGen during 1994, which significantly expands its product menu. These agreements are similar in that they allow IBM to manufacture and sell chip-level products to customers outside IBM. These agreements should allow IBM Microelectronics to expand its marketing reach and develop new sales channels for PowerPC.

## 4.7 NexGen

It's hard to know what to make of NexGen (see Table 4-13). Although the company was founded in 1986—two years before Cyrix—it was not until mid 1994 that the company shipped its first product. But the device it finally did ship was technologically quite impressive: a high-end binary-compatible x86 microprocessor, designed completely independently of Intel, that was able to compete favorably with Pentium.

Company	NexGen
Year Founded	1986
Headquarters	Milpitas, California
Stock Exchange/Symbol	(privately held)
Revenues	N.A.
Net Income	N.A.
Net Profit	N.A.
Total Assets	N.A.
Total Liabilities	N.A.

Table 4-13. NexGen company profile. (Source: company reports)

NexGen spent its youth doing pure research and development. According to company spokespersons, the first two years were spent studying the principles of x86 architecture, and in 1988 they began designing what has become the Nx586. After eight years of effort, NexGen finally succeeded in bringing their first product—the Nx586—to market. The general market response to the Nx586 in 1995 should be interesting to watch.

The company is still privately held, so detailed income and expense statements are unavailable. Since 1986, NexGen has reportedly received over \$90 million in funding. Principal investors include Kleiner, Perkins, Caufield and Byers, Paine Webber Inc., ASCII Corporation, Compaq Computer Corporation, Olivetti Corporation, and Harvard University.

## 4.8 For More Information...

Additional business information on the various x86 microprocessor vendors may be found in the following publications:

### Vendor Publications

- 1: *Advanced Micro Devices 1993 Annual Report*. Advanced Micro Devices, 3/94, order #90180.
- 2: *AMD's Impact on Personal Computers*. Advanced Micro Devices, 9/94, order #18457B.
- 3: *Chips and Technologies, Inc. 1993 Annual Report*. Chips and Technologies.
- 4: *Cyrix Corporation 1993 Annual Report*. Cyrix Corporation.
- 5: *Defining Intel: 25 Years / 25 Events*. Intel Corporation, 6/93, order #241730. (*An interesting compilation of achievements in business and technology, published to commemorate the 25th anniversary of Intel's founding.*)
- 6: *Intel Corporation 1993 Annual Report*. Intel Corporation, 3/94, order #241941-001.
- 7: *International Business Machines – 1993 Annual Report*. International Business Machines.
- 8: *Texas Instruments 1993 Annual Report*. Texas Instruments, order #TI-29387.

### Microprocessor Report Articles

- 9: *Survey of Semiconductor Companies*. MPR vol. 5 no. 11, 6/12/91, pg. 16.
- 10: *Buy Intel Because, Well, It's Intel\**. Michael Slater, MPR vol. 5 no. 13, 7/24/91, pg. 3. (*Editorial.*)
- 11: *Intel Declares Victory in the Mother of All Demos\**. John Wharton, MPR vol. 5 no. 21, 11/20/91, pg. 11. (*Oblique Perspective column.*)
- 12: *Proliferation of 386/486-Compatible Microprocessors to Accelerate in '92\**. Michael Slater, MPR vol. 6 no. 1, 1/22/92, pg. 1. (*Cover story.*)
- 13: *A New World for Intel\**. Michael Slater, MPR vol. 6 no. 6, 5/6/92, pg. 3. (*Editorial.*)
- 14: *Gonzo Marketing\**. John Wharton, MPR vol. 6 no. 9, 7/8/92, pg. 20. (*Oblique Perspective column.*)
- 15: *Semiconductor Company Profiles*. MPR vol. 6 no. 11, 8/19/92, pg. 20.



- 16: *Who Drives the PC Industry?*. Michael Slater, MPR vol. 6 no. 16, 12/9/92, pg. 3. (Editorial.)
  - 17: *Intel Continues Record Spending*. MPR vol. 6 no. 17, 12/30/92, pg. 4. (Most Significant Bits item.)
  - 18: *Readers Pick AMD as Top Processor Vendor*. Linley Gwennap, MPR vol. 7 no. 2, 2/15/93, pg. 15. (Feature article.)
  - 19: *What's Next for Intel?*. Michael Slater, MPR vol. 7 no. 6, 5/10/93, pg. 3. (Editorial.)
  - 20: *x86 Vendors Unveil New Slogans, Not Chips*. MPR vol. 7 no. 16, 12/6/93, pg. 5.
  - 21: *Number Two Doesn't Always Try Harder*. Linley Gwennap, MPR vol. 8 no. 3, 3/7/94, pg. 3. (Editorial.)
  - 22: *Intel's Predicament*. Michael Slater, MPR vol. 8 no. 6, 5/9/94, pg. 3. (Feature article.)
- Other Technical References**
- 23: *Aspects of Cache Memory and Instruction Buffer Performance*. M. D. Hill, U.C. Berkeley, 1987. (Ph.D. dissertation.)
  - 24: *Marketing High Technology*. William Davidow, Free Press, 1986. (Case histories of Intel marketing strategies.)
  - 25: *Profiles—A Worldwide Survey of IC Manufacturers*. Integrated Circuit Engineering, 1994.
- Other Periodicals**
- 26: *Rethinking IBM*. Judith Dobrzynski, Business Week, 10/4/93, pg. 86. (Business viewpoint of Lou Gerstner's first six months.)
  - 27: *Inside Intel*. Robert Hof, Business Week, 6/1/92, pg. 86.
  - 28: *Video, Flash Memory – The 'Other' Intel is Cooking*. Robert D. Hof, Business Week, 6/1/92, pg. 90.
  - 29: *Computer Revolution*. Stratford Sherman, Fortune Magazine, vol. 127 no. 12, 6/14/93, pg. 56.
  - 30: *Products That Make Markets*. Belinda Luscombe, Fortune Magazine, vol. 127 no. 12, 6/14/93, pg. 82.
  - 31: *Business Week 1000, America's Most Valuable Companies*. Business Week, 6/22/93.
  - 32: *Will We Keep Getting More Bits for the Buck?*. Otis Port, Neil Gross, Robert Hof, Richard Brandt, and Peter Burrows, Business Week, 7/4/94, pg. 90.

33: *Wonder Chips*. Otis Port, Neil Gross, Robert Hof, and Gary MacWilliams, *Business Week*, 7/4/94, pg. 86.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)

# Part III: The Products

---

By mid-1994, at least six major vendors had begun competing for slices of the x86 pie, and at least one had thrown in the towel. More than 40 different products had been announced, counting functionally different devices in each vendor's product repertoire and functionally compatible parts produced by different vendors.

**Part III** of this report describes briefly each of the 386- and 486-class microprocessor products announced through 4Q94. It is divided into separate chapters that discuss each vendor's product lines:

**Chapter 5: Intel 386 Microprocessors**

**Chapter 6: Intel 486 Microprocessors**

**Chapter 7: AMD 386 and 486 Microprocessors**

**Chapter 8: C&T 386 Microprocessors**

**Chapter 9: Cyrix 486 Microprocessors**

**Chapter 10: IBM 386 and 486 Microprocessors**

**Chapter 11: TI 486 Microprocessors**



# 5

## Intel 386 Microprocessors

Since the beginning, Intel has been the first vendor to introduce products at each new generation of technology. As a result, the on-chip resources, pinouts, and bus protocols defined by Intel have become de facto standards for the industry, adopted or adapted by competing vendors.

The Intel 386 family in particular served to define architectural and electrical capabilities that have reappeared in many other products. In order to understand the design techniques used by various flavors of competing 386 and 486 devices, it's therefore useful first to understand the Intel 386 product line.

### **Intel 386 Family Overview**

When it was introduced in 1985, the 80386 was seen as a "80286 stretch," the next in an ongoing series of enhancements to Intel's microprocessor product line. Its instruction set, architecture, and on-chip resources—ALU, register file, memory management system, bus interface, and so forth—were functionally quite similar to its predecessors'.

Where there were differences, they were generally quantitative, not qualitative. The ALU and working registers, which had been 8 bits each on the 8080 and 16 bits on the 8086, 80186, and 80286, grew to 32 bits on the 80386. Whereas the 8080 and 8088 had 8-bit data buses, and the 8086, 80186, and 80286 data buses had grown to 16 bits, the 80386 grew its data bus to 32 bits.

The address bus, which had supported 16 bits in the 8080, 20 bits for the 8088, 8086, and 80186, and 24 bits for the 80286, grew to 32 bits for the 80386 as well. A few new instructions

were added to the 80386 instruction set—just as the 8086, 80186, and 80286 had each expanded the instruction set of its predecessor. But no new working registers were added to the 386 programming model, and no new types of instructions were introduced.

In practice, though, the 80386 set a new standard for microprocessor functionality. Whereas a 16-bit CPU and data bus constituted a clear compromise between desired functionality and the reality of then-current technology, the 32-bit ALU, registers, and buses of the 80386 seemed, if anything, to exceed market demands.

New memory paging support, improved mechanisms for executing existing 8086 programs, and the elimination of the need for memory segmentation all helped overcome many of the weaknesses of the 8086 and 80286 designs. (Ironically, mainstream users were unable to take advantage of these capabilities until Windows 3.1 became available in 1992—seven years after the 80386 was introduced!) The 80386 was also the first microprocessor to include built-in breakpoint and debug registers, which greatly simplified software development and testing.

Thanks to its faster clock, higher resolution, and ability to emulate DOS-based 8086 object code from within protected mode, the 80386 could significantly outperform the 80286. Thus it soon became clear that the 80386 architecture had the potential to obsolete its humble 16-bit predecessors.

In time, Intel reworked the 80386 nomenclature to reflect the quantum leap in its capabilities. The leading “80” was dropped and the “i” prefix added when it became clear that companies could not copyright simple numbers. The part designation was retrofitted with “DX” and “SX” suffixes as versions with different pinouts were introduced.

## 5.1 Intel 386 Core Technology

By today’s standards the Intel 386 core is quite spartan. Its 32-bit integer execution unit implements (by definition) the base-level 386 architecture defined in **Chapter 3** of this report, including an ALU, working register file, and various test and debug registers. In addition, the device contains a paged memory-management unit with a translation lookaside buffer

(TLB), and a system bus interface with 32 bits each of address and data.

While its predecessors were initially implemented using NMOS process technology and later redesigned for CMOS, the 80386 was the first Intel microprocessor of any type to be designed for CMOS technology from the start. This was done not so much to minimize power consumption as to reduce internal heat dissipation. Following the design conventions of the day, the internal logic of the device used a two-phase timing regime, in which buses, PLAs, and other signal nodes are “precharged” to one logic state during the first phase, and then conditionally “discharged” to the other state during the second.

If such a CPU spends too much time in its second phase, the charge stored on these nodes will dissipate, whether or not the chip logic intended it too. Such nodes are thus considered “dynamic,” in much the same sense as the dynamic memory cells of a conventional DRAM. Because of this, the original 80386 core could not operate below a certain minimum clock frequency. This minimum frequency leads in turn to a relatively high minimum current drain ( $I_{cc}$ ), which effectively precludes the device from being used in small, battery-powered applications.

(In 1990, Intel redesigned the 80386 core to eliminate dynamic nodes, so newer embedded-control products based on this core may indeed allow static operation.)

### **Core Design**

When the 80386 was designed, Intel was undoubtedly more concerned just with getting the device to work than with the aesthetic details of its implementation. Most of its increased transistor and die size budget relative to the 80286 went into widening the ALU, CPU registers, and internal buses, adding new segmentation registers and paged memory management, and enhancing the instruction set, memory-management model, and compatibility modes.

The time and material budget left little room for design sophistication or performance optimization. Obtaining absolute maximum throughput in terms of number of clock cycles per instruction (CPI) was of lesser importance. Instead of the sleek, efficient (but silicon-hungry) multiple-stage execution pipelines so common among today’s high-end processors, the 386 core makes extensive use of microcoded execution logic. Compared to more modern devices, then, the 80386 core seems almost gla-

cially slow, although it was still considerably faster than the chips that came before.

Instead of a conventional pipeline, the 386 design is built from a series of interconnected special-function units. A semi-autonomous instruction prefetch unit retrieves instruction bytes from external memory into a 16-byte queue that feeds the instruction decode logic. (Later design reduced the queue length to 12 bytes.) Decode logic extracts the opcode, register, offset, and immediate operand fields as needed and then transmits complete instruction words in parallel into a three-level queue of ready instructions.

A microcoded execution unit then interprets each of these instructions sequentially until it is done. The elasticity of the instruction-prefetch and assembled-instruction queues accommodates breaks in the flow of execution that occur when instruction fetches contend for use of the external bus.

Unfortunately, while the prefetch unit does endeavor to retrieve, align, and begin decoding ensuing instructions during slow ALU operations, this rudimentary form of instruction overlapping does not accelerate branch processing, and at times it can create contention for the system bus, delaying other operations.

### **Low-Level Instruction Timing**

The rate at which instructions execute is limited by several bottlenecks within the Integer Execution Unit (IEU). Instruction execution is not pipelined, so even the simplest register-to-register MOV and ADD operations require at least two clock cycles: one to read the operands, and a second to perform the operation and store the result.

More typically, though, instructions require additional clock cycles to complete. Jumps, calls, and returns consume at least 8, 10, and 13 cycles, respectively. Simple integer multiplies involve an iterative shift-and-add process that consumes up to 41 clock cycles.

Memory transfer instructions—loads, stores, and ALU operations that use a memory-based source or destination operand—require extra clock cycles to retrieve address register contents and compute the effective operand address. The 386 Address Generation Unit (AGU) contains a single 32-bit two-input adder for all code and data address computations, so memory address computations that involve more than two com-



ponents must be performed serially. Index-register scaling (when used) can add still more delays.

Reading or writing each memory-based operand adds at least two additional clock cycles—more, if wait states are needed. Operands not aligned on natural memory-word boundaries require additional transfers to retrieve or write back. If, for example, a memory system requires two wait states per transfer, even a “simple” register-to-memory ADD using a four-part indexed address to a misaligned address can require up to 15 CPU cycles to complete. The most complex microcoded instructions can take dozens of cycles.

The 386 core provides no direct on-chip support for floating-point operations. While floating-point instructions may be detected and interpreted to some degree by 386-family microprocessors, the actual floating-point circuitry is contained within a separate floating-point “coprocessor” device designated the i387SX or i387DX.

### **Clock Timing**

The clock circuitry within the 386 core is fairly unsophisticated. An externally generated signal is divided by two internally to produce the two non-overlapping square-wave phase signals that synchronize both internal operations and the bus interface. By convention, specifications of a particular device’s operating frequency refer to both the internal, subdivided core frequency and the external bus interface, such that a 20-, 25-, or 33-MHz processor, for example, requires a 40-, 50-, or 66-MHz external oscillator.

Each instruction consumes multiple cycles of the subdivided internal clock signal. A two-clock NOP (no operation) or register-to-register ADD instruction, for example, consumes two CPU clock cycles, corresponding to four oscillations of the external input.

## 5.2 The Intel i386DX Microprocessor

The Intel i386DX is the oldest member of the 32-bit x86 dynasty. While its age places it among the least sophisticated designs, it has nevertheless become a standard against which the features and performance of other 386 and 486 processors are often measured. Table 5-1 summarizes the general features and specifications of the i386DX microprocessor.

Product Name	Intel i386DX
Introduction Date	October 1985
Prognosis	Targeting embedded-control and third-world PCs
Device Integration Level	Microcoded 32-bit integer execution unit Paged memory-management unit
CPU Architecture Level	De facto standard 386 integer instruction set
Core Technology	De facto standard Intel 386 core
Pinout	De facto standard 386DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4 GB (Address A31..A2 plus BE3#.BEU#)
Data-Transfer Modes	Two cycles minimum per 32-bit transfer One-half cycle address pipelining optional Dynamic bus resizing for 16-bit transfers
Cache Support	Optional external 82385DX cache controller or 82395DX integrated cache peripheral
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	20-, 25-, or 33-MHz core operation
Clocking Regime	Operating frequency = Clkin freq + 2
Active Power Dissipation	1.95 W @ 5.0 V and 33 MHz (worst case)
Power-Control Features	None
Process Technology	Initially 1.5 $\mu$ two-layer-metal CMOS Redesigned for 1.0 $\mu$ two-layer-metal CMOS
Die Size	404 $\times$ 379 mils (1.5 $\mu$ design) 270 $\times$ 244 mils (1.0 $\mu$ design)
Transistor Count	275,000 transistors
Package Options	132-pin "standard" PGA or 132-pin PQFP
Notes	✓ First x86 CPU to implement the 32-bit architecture

Table 5-1. Intel i386DX feature summary.

**Features** The i386DX contains a full 32-bit integer unit, a 4-Gbyte logical address space, and a paged virtual-memory management unit (PMMU). The device implements (by definition) and complies with the full 386 architecture, i.e., programming model, register set, instruction set, binary encodings, and so forth.

**Cache Support** The i386DX contains no direct support for cached memory systems. An Intel 82385DX cache controller, an 82395DX cache controller/RAM combination device, or similar products from other vendors may optionally be used in i386DX-based designs. Many integrated chip sets also contain cache control logic.

**Floating-Point Support** The i386DX provides no direct on-chip support for floating-point operations. Initially, floating-point support for the i386DX could be provided using either an Intel 80287—the FPU initially designed for the 80286—or a newer, more efficient design designated the 80387, which was later renamed the i387DX. In time, i386DX support for the 80287 was phased out, and a variety of alternative coprocessor devices were developed by such third-party semiconductor vendors as Weitek, Cyrix, and IIT.

**System Interface** As the first implementation of the 80386 family, the system interface defined by the i386DX device became an industry standard and served as the starting point as new functions were added in follow-on devices. The device provides separate 32-bit buses for address bits and data in order to support the full 4-gigabyte physical address space defined by the architecture. Figure 5-1 illustrates a basic i386DX system interface.

The standard 132-pin PGA package includes 83 signal pins, 41 power and ground pins, and 8 no-connect pins. Since the con-

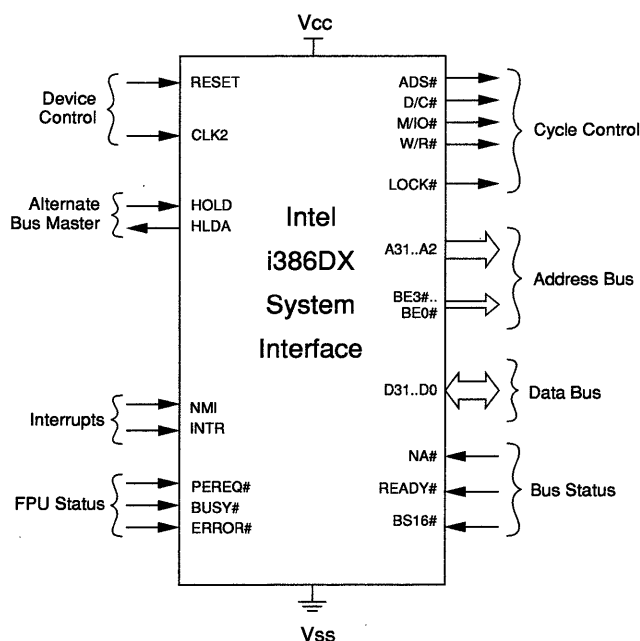


Figure 5-1. Intel i386DX system interface.

ventions followed by the i386DX bus interface have been adopted throughout the 386 and 486 product lines, the name and function of each signal pin is described in detail below.

**Address and Data Bus.** Table 5-2 describes the i386DX address- and data-bus signals. Pins D31..D0 form the bidirectional data bus; D0 denotes the least-significant bit. Each of the four bytes that make up a 32-bit data word has its own address. The x86 architecture is “little-endian,” which means the least-significant byte within each word is has the lowest address.

Signal	Direction	Function
D31..D0	I/O	Data bus (D31=MSB, D0=LSB)
A31..A2	Out	Address bus (A31=MSB)
BE3#..BE0#	Out	Byte enable controls (BE3# enables D31..D24)

Table 5-2. Intel i386DX address and data bus signals.

Note that the address pins A31..A2 provide only the 30 highest-order bits as external signals. These represent the address of an aligned four-byte word of physical memory or an I/O port. Four separate “byte-enable” control signals (BE3#..BE0#) indicate which of the bus’s one-byte subfields is active during each transfer. Pins D31..D24 are enabled by BE3#, while BE0# enables D7..D0.

The byte-enable pins serve to encode both the two lowest-order address bits and the number of bytes involved in a given transfer. In effect, A31..A2 identify one of one billion 32-bit words in memory, while BE3#..BE0# indicate which combinations of bytes within that word are involved in each transfer.

**Bus Control and Status.** Table 5-3 describes the i386DX bus control and status pins. Output pin ADS# (address strobe) goes low during the first clock cycle of each new bus cycle to indicate that a new transfer operation has begun, and to indicate that the various other address and control signals are valid.

Output pins M/IO# (memory/IO), D/C# (data/code), and W/R# (write/read) define the type of bus cycle being performed. These signals are encoded as shown in Table 5-4.

LOCK# (bus lock) is an output signal that indicates the external memory system must complete the current data transfer cycle in “locked” mode: if main memory is currently in use, the transfer must be delayed, and once the transfer begins, no other bus master may initiate any transfers until the locked transfer is

Signal	Direction	Function
ADS#	Out	Address strobe (indicates start of new bus cycle)
M/IO#	Out	Memory vs I/O cycle (indicates operand type)
D/C#	Out	Data vs Code cycle (indicates operand usage)
W/R#	Out	Write vs read cycle (indicates transfer direction)
LOCK#	Out	Locked bus cycle (indicates indivisible operation)
NA#	In	Next address (enables pipelined transfers)
READY#	In	Ready (transfer data accepted/available)
BS16#	In	Bus size 16 (splits word transfers into two cycles)
HOLD	In	Hold request (external bus master request)
HLDA	Out	Hold acknowledge (bus available to other master)

Table 5-3. Intel i386DX system bus control and status signals.

complete and LOCK# is deasserted. LOCK# is asserted automatically by the CPU during page-table accesses and atomic (non-divisible) read-modify-write instructions; it may be explicitly requested for any other memory operations by inserting a LOCK prefix before the instruction opcode.

The NA# (next address) input signal lets the external memory system acknowledge that it has latched or no longer needs the values on A31..A2. If additional memory cycles are pending, the i386DX can respond to this signal by presenting the address and control signals needed for the ensuing transfer before the outstanding data transfer is done.

The READY# (ready) input synchronizes data transfer completion, and allows slow memory systems to request wait states as necessary until read data is valid or write data has been accepted.

M/IO#	D/C#	W/R#	Transfer Cycle Type
Low	Low	Low	Interrupt acknowledge cycle
Low	Low	High	(does not occur)
Low	High	Low	Read data from I/O port
Low	High	High	Write data to I/O port
High	Low	Low	Fetch instruction from memory
High	Low	High	If BE2# is low: System halt cycle if BE0# is low: System shutdown cycle
High	High	Low	Read data operand from memory
High	High	High	Write data operand to memory

Table 5-4. Intel i386DX transfer cycle encoding.

BS16# (bus size 16) is an input signal that may be asserted if the external memory system can not support 32-bit transfers to the requested address. If so, the i386DX will complete the transfer or latch data using pins D15..D0 only, and then initiate a second transfer cycle to read or write the high-order half of the initial operand. This facility allows BIOS EPROMs, for example, to be just 16 bits wide.

The HOLD (hold request) input and HLDA (hold acknowledge) output signals provide a mechanism by which the i386DX can share use of a private local bus with other processors or DMA controllers. When HOLD is requested, the i386DX disables its address, data, and status output signals and asserts HLDA in response.

**Device Control and Status.** Table 5-5 describes the i386DX device control and status pins. CLK2 (2× clock) is the system clock input signal. An externally generated clock signal of twice the desired core frequency must be driven onto pin CLK2. Bus logic runs at the same frequency and is controlled by the same phase signals as the core.

Signal	Direction	Function
CLK2	In	Processor clock input (CPU freq = CLK2 freq ÷ 2)
RESET	In	Processor reset
INTR	In	Maskable interrupt request
NMI	In	Non-maskable interrupt
PEREQ	In	Processor extension (FPU) service request
BUSY#	In	Busy (FPU coprocessor status)
ERROR#	Out	Floating-point error detected
FLT#	In	Float (disables all outputs for board-level testing) (Note: not provided by Intel PGA packages)

Table 5-5. Intel i386DX device control and status signals.

The RESET (reset) input pin is asserted to reset the device. The INTR (interrupt request) pin is asserted to initiate a vectored CPU interrupt sequence. The NMI (non-maskable interrupt request) input pin invokes a non-vectored interrupt service routine that is always enabled and always takes precedence over all other interrupt service routines.

PEREQ (processor-extension request), BUSY# (busy), and ERROR# (error) are three signals that coordinate communication between the i386DX and an external floating-point math coprocessor, as described below.

The FLT# (float) is an input signal present on PQFP-packaged versions of the i386DX that disables the output drivers of all other pins. Normally, in order to perform test or debug a circuit board, the CPU must be removed to keep it from interfering with other devices on the circuit board. PQFP devices, however, are typically soldered directly to a circuit board. Asserting this signal disables all the CPU's outputs, and has the same effect as removing the chip from the circuit. The FLT# signal is not supported by PGA versions of the i386DX, since motherboards that contain PGA sockets may have board-level testing performed before the CPU is inserted, and PGA devices may be removed from their sockets should further system debugging be needed.

### **Package and Frequency Options**

The i386DX was initially offered only in a 132-pin ceramic pin-grid-array (PGA) package. In recent years Intel has begun offering the part in a lower-cost 132-lead plastic quad flat pack (PQFP) package in response to competition from AMD (see **Chapter 7**). Figures 5-2 and 5-3 depict the pinout assignments for each package type.

### **Vital Statistics**

The i386DX contains 275,000 transistors, and was originally fabricated using a 1.5-micron, two-layer-metal CMOS process. Initial parts allowed operating frequencies of up to 16 or 20 MHz, and were housed in a 132-pin ceramic pin-grid-array (PGA) package. In time, Intel redesigned the part for 1.0 micron design rules, and raised its maximum clock frequency to 33 MHz.

Intel currently offers the i386DX in 20-, 25-, and 33-MHz versions, although the lower-speed parts generally have the same price as the faster ones. The design of the i386DX device is not static; standard devices specify a minimum input frequency of 16 MHz, corresponding to an 8-MHz minimum core frequency.

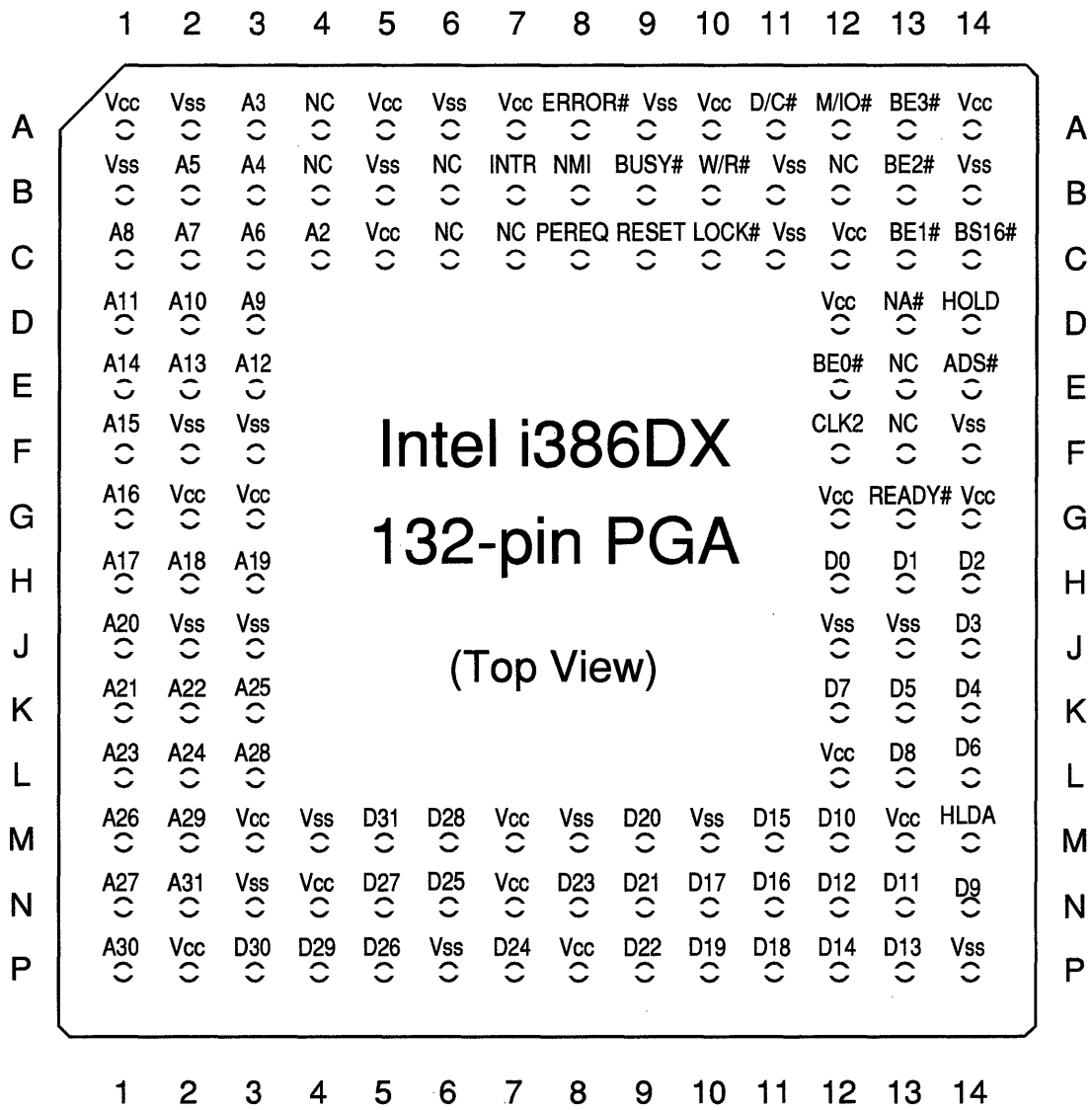


Figure 5-2. Intel i386DX PGA pinout.



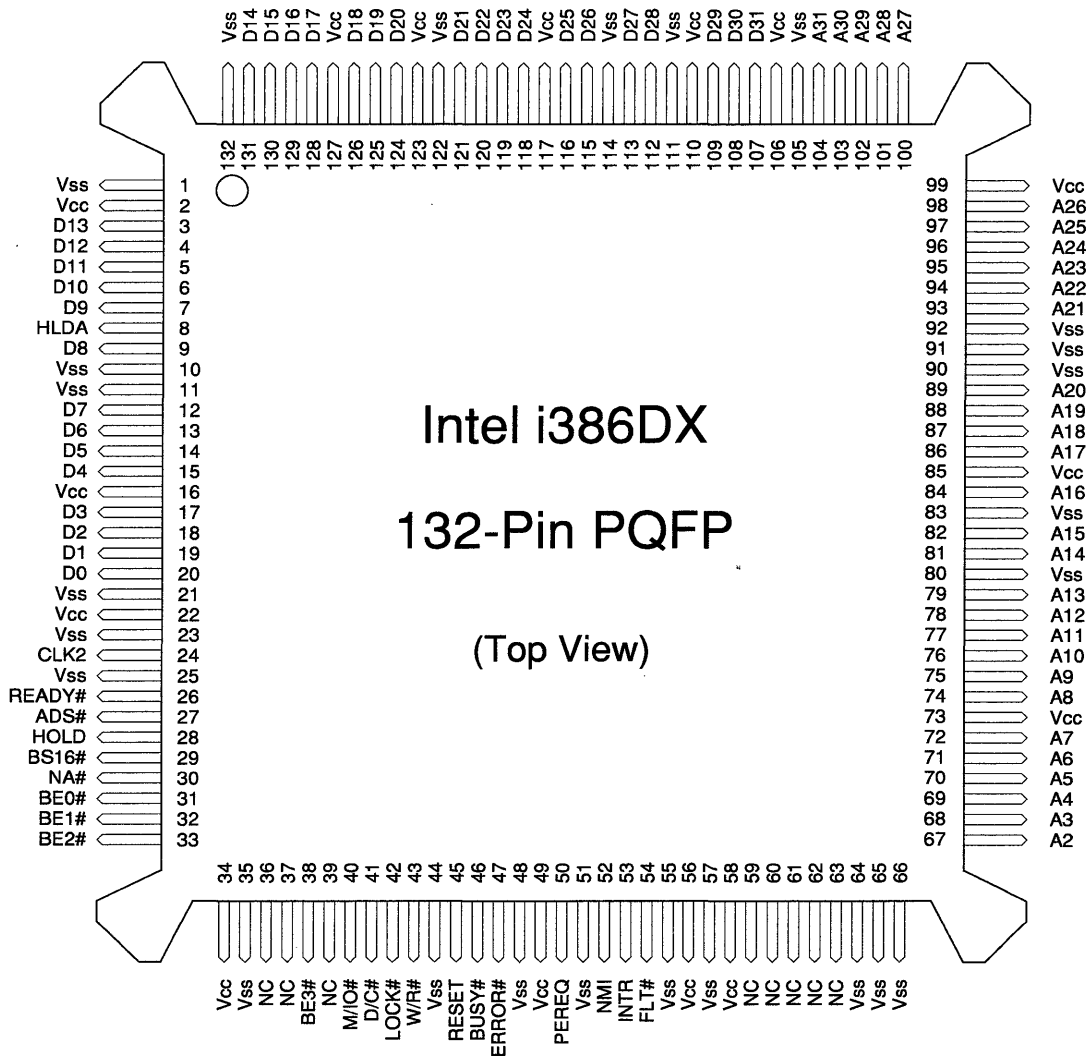


Figure 5-3. Intel i386DX PQFP pinout.

## 5.3 The Intel i386SX Microprocessor

The i386SX is a lower-cost derivative of the basic i386DX design. It is fully software-compatible—upward, downward, and sideways—with the i386DX device. All that's different is its physical manifestation: the data bus is limited to 16 bits, and the physical address bus to 24 bits, allowing the device to be sold in a lower-cost pin-reduced package. Features of the i386SX are summarized in Table 5-6.

Product Name	Intel i386SX
Introduction Date	June 1989
Prognosis	Being de-emphasized except within third-world markets
Device Integration Level	Same as i386DX
CPU Architecture Level	Same as i386DX
Core Technology	Same as i386DX
Pinout	De facto standard 386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Two cycles minimum per 16-bit transfer One-half cycle address pipelining optional
Cache Support	Optional external 82385SX cache controller or 82396SX integrated cache controller/RAM
Floating-Point Support	Optional external 387SX-class FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	20-, 25-, or 33-MHz core operation
Clocking Regime	Core operating frequency = $1/2 \times \text{Clkin}$
Active Power Dissipation	1.9 W @ 5.0 V and 33 MHz (worst case)
Power-Control Features	None
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Die Size	242 $\times$ 269 mils
Transistor Count	275,000 transistors
Package Options	100-pin plastic QFP
Notes	Smaller, lower-priced variation on 386 core

Table 5-6. Intel i386SX feature summary.

**Background** From a software perspective, the 386 architecture had many advantages over the 80286, including increased arithmetic precision, expanded addressability, paged memory management, and better emulation capabilities. From a hardware perspective, these advantages came at some cost.

The 80286 device itself was significantly less expensive, due to its smaller die and price competition from many alternate

sources. Expanding the i386DX address and data buses to 32 bits required a package with extra pins for the wider buses, new bus-control signals, and the additional power and ground pins needed to drive them. These extra pins in turn mandated a larger PGA package, which cost more to build.

Moreover, the i386DX indirectly increased system costs in other ways. Its wider buses required more interface circuitry for address buffers, bus transceivers, decoders, and the like. The 32-bit memory bus required twice as many DRAMs to populate a minimum system, and SIMMs (byte-wide memory modules) had to be added or replaced in groups of four rather than two. These extra components, and the i386DX's own physically larger package, consumed more real estate on the motherboard. An i386DX-based system drew considerably more power than a 80286 box, with possibly adverse effects on heat dissipation and power supply design.

Thus, despite the architectural advantages of the i386DX, sales of the 80286 remained high for years thereafter. This phenomenon spoiled Intel's plans two ways. As long as the 80286 continued to sell well, software designers would continue to view the 8086 and 80286 as the least-common denominators in the system market and might never migrate their application programs to the 32-bit x86 world, over which (coincidentally) Intel was the sole-source supplier. And even though Intel continued to be the world's largest single manufacturer of 80286 devices, competition was starting to eat into Intel's market share, and had driven its ASP (average selling price) and margins into the mud.

Thus was the i386SX born in 1989, its primary purpose to kill off the multiple-sourced 80286 and restore Intel's monopoly in the x86 market.

**Features** Like its big brother, the i386SX contains a full 32-bit integer unit, a 4-Gbyte logical address space, and paged virtual-memory management. Both have the same programming model, register set, instruction set, binary encodings, and so forth. The cores of both processors were designed using the same basic implementation technology and microarchitecture, and have essentially the same transistor count and internal timing.

Because the i386SX has a narrower bus interface, however, instruction timings often differ between the two devices. In general, the i386SX device makes less efficient use of the system

bus interface. When an i386DX device needs to read or write an aligned 32-bit value, the entire transfer can complete in two internal clock cycles. An i386SX requires at least two transfers—four internal clock cycles, or eight CLK2 oscillations—to transfer the same operand in two 16-bit parts, least-significant part first.

Likewise, an i386DX can retrieve 32-bit values aligned on an odd byte address in four internal clock cycles, while an i386SX requires at least six. Moreover, the i386SX instruction prefetch logic must generally perform nearly twice as many 16-bit reads to retrieve the instructions for a given code sequence, consuming greater bus bandwidth and increasing the likelihood that data transfers will have to contend for system bus usage. Because of this last phenomenon, the i386SX design shortened the instruction prefetch queue from 16 bytes to 12.

### Cache Support

The i386SX contains no direct support for cached memory systems. An Intel 82385SX cache controller, 82396SX integrated cache controller/RAM device, or similar products from other vendors may optionally be used in i386SX-based designs, as can the cache-control logic within many integrated chip sets.

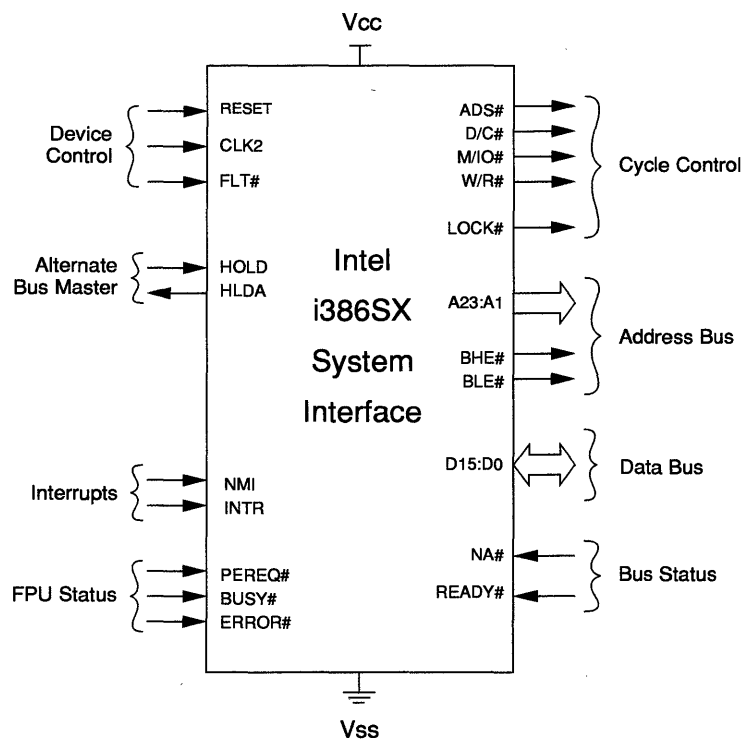


Figure 5-4. Intel i386SX system interface.

## Coprocessor Support

Because of its narrower data bus, the i386SX is unable to connect to a standard i387DX FPU. Intel developed a separate device, designated the i387SX coprocessor, for i386SX-based designs. Other vendors, including Weitek, Cyrix, and IIT, likewise have slightly different designs for 386SX-class systems.

## System Interface

From a practical perspective, the only difference between the i386SX and i386DX processors is the interface between each device and its system. Figure 5-4 illustrates a basic i386SX system interface.

The standard 100-pin PQFP package includes 58 signal pins, 32 power and ground pins, and 10 no-connect pins. Tables 5-7 through 5-9 define the names and functions of the i386SX signal pins.

Signal	Direction	Function
A23..A1	Out	Address bus (A23 = MSB)
D15..D0	I/O	Data bus (D15 = MSB)
BHE#, BLE#	Out	Byte high enable and byte low enable controls

Table 5-7. Intel i386SX address and data bus signals.

Most of these signals have the same name and perform the same function as comparable signals defined by the i386DX. The chief differences are that i386DX address pins A31..A24 and data pins D31..D16 have been eliminated. The narrower data bus requires only two byte-enable signals, now designated BHE# (byte high enable) and BLE# (byte low enable) in lieu of BE3#..BE0#. Since the memory system now consists of 16-bit words, signal BS16# is no longer needed, and an extra low-order address pin (A1) has been added.

Signal	Direction	Function
ADS#	Out	Address strobe (start of new bus cycle)
M/IO#	Out	Memory vs I/O bus cycle
D/C#	Out	Data vs code bus cycle
W/R#	Out	Write vs read bus cycle
LOCK#	Out	Locked (indivisible) bus cycle
NA#	In	Next address (enables pipelined transfers)
RDY#	In	Ready (transfer data accepted/available)
HOLD	In	Bus hold request (external master request)
HLDA	Out	Bus hold acknowledge (bus available)

Table 5-8. Intel i386SX system control and status signals.

Signal	Direction	Function
CLK2	In	Processor clock input (CPU freq.=1/2 CLK2)
RESET	In	Processor reset
INTR	In	Maskable interrupt request
NMI	In	Non-maskable interrupt
PEREQ	In	Processor extension (FPU) service request
BUSY#	In	Busy (FPU coprocessor status)
ERROR#	Out	Floating-point error detected
FLT#	In	Float (disables all outputs for board testing)

Table 5-9. Intel i386SX device control and status signals.

### Package and Frequency Options

The i386SX is offered only in a 100-pin PQFP package (see Figure 5-5), and versions are currently available with core frequencies up to 20, 25, and 33 MHz. The minimum core operating frequency is specified to be 4 MHz, though specially selected “low-power” versions can be ordered that allow the core frequency to be as low as 2 MHz. These slower parts generally carry a slight price premium—partly because they take longer to test, increasing the manufacturing cost, and partly because their improved specifications make them worth more to users.

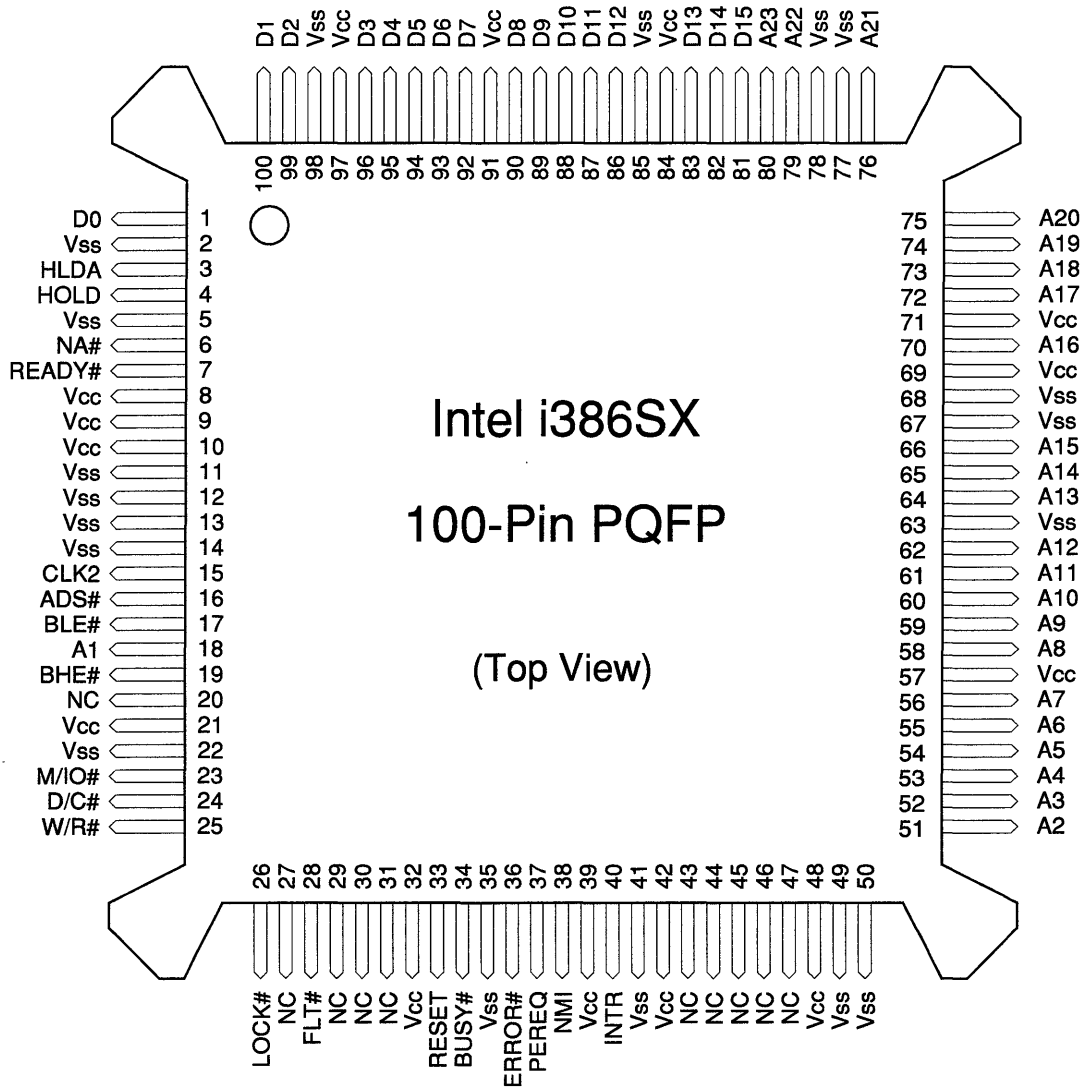


Figure 5-5. Intel i386SX PQFP pinout.

## 5.4 The Intel 80376 Microprocessor

The Intel 80376 microprocessor is a version of the i386SX with real-mode operation and the on-chip PMMU disabled. The 80376 is intended for embedded computing rather than PC-class applications and is included in this report chiefly for historical and comparative reasons. Table 5-10 summarizes the general features and specifications of the 80376 microprocessor.

Product Name	Intel 80376
Introduction Date	April 1989
Prognosis	Positions strictly for embedded applications
Device Integration Level	Same as i386SX but with PMMU disabled
CPU Architecture Level	Same as i386SX but with real-mode operation disabled
Core Technology	"De-DOSed" Intel 386SX die
Pinout	Same as i386SX
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as i386SX
Cache Support	Optional external 82385SX cache controller or 82396SX integrated cache controller/RAM
Floating-Point Support	Optional external i387SX FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	16- or 20-MHz core operation
Clocking Regime	Same as i386SX
Active Power Dissipation	1.25 W @ 5.0 V and 20 MHz (worst case)
Power-Control Features	None
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Transistor Count	275,000 transistors
Die Size	269 $\times$ 242 mils
Package Options	100-pin plastic QFP
Notes	Modified version of the i386SX die

Table 5-10. Intel 80376 feature summary.

### Background

When the i386SX was still in its planning and design stages, Intel had great hopes that this new, low-cost device would open new markets for the 386 family, not just in lower-cost desktop PCs, but in laser printers, factory automation, network controllers, and the like. In response to extremely low price projections from Intel, Xerox and others began designing embedded systems based on i386SX hardware.



By the time the i386SX was introduced, however, company strategy had shifted. Intel decided the initial target prices were lower than necessary; the i386SX was simply worth more to PC vendors than the prices that had already been promised to embedded-system vendors. Thus was born the 80376, a device that could deliver the same performance as the i386SX, but that would be uniquely suited for the embedded world, and thus would not (i.e., *could* not) compete for PC sockets.

**Features** The 80376 contains a slightly modified version of the standard i386SX die. It is delivered in the same 100-pin PQFP as the i386SX, uses the same pinout, and has the same system interface with respect to signal definitions, timing, and electrical characteristics. It connects to the same i387SX floating-point coprocessor and other peripherals as is the i386SX, as well as 387SX-class FPUs and peripherals from other vendors. (See the discussion of the i386SX system interface above for details.)

**Architecture Modifications** For nearly all practical purposes, the 80376 architecture is nearly identical to that of the standard 386. The user-mode programming model, addressing modes, and instruction set and encodings are identical, as are most of the system-mode control registers and instructions.

However, there are two critical differences between the 80376 and the i386SX devices. The first is that the on-chip PMMU—present on all other members of the 386/486 product line—and all registers that relate to it have been disabled. This is purportedly because embedded applications generally have no need for memory paging. Application code for laser printers, network hubs, and the like is typically resident in on-board EPROMs or ROMs, rather than in DRAM; thus there is no secondary storage device, such as a disk drive, from which code is loaded, nor is there any need to swap pages in and out of RAM.

Second, real-mode operation has also been disabled. Whereas other 386 and 486 family members begin operating in 16-bit “real mode” following reset, and require software intervention to switch into 32-bit “native” mode, the 80376 powers up in native mode directly. This change was purportedly made to simplify the programming interface and save the user from having to understand the different mode semantics.

In truth, both changes were made so Intel could play marketing games with the part. Real-mode operation, in which all other 386-class CPUs emulate a high-speed 8086—is necessary for running standard DOS software. Memory paging is necessary

for Unix. By omitting these modes, Intel was able to assure that the resulting device could not be induced to execute the established base of DOS and Unix applications, and thus would not be suitable for desktop computer systems.

As a result, Intel was able to introduce the 80376 at approximately half the price of its i386SX cousin. Price pressure on the i386SX later caused its price to fall, however, until at this point the price differential is quite small. For many applications, this difference is not enough to justify the (albeit minor) software differences between the parts, or the lack of second-source channels.

**Package and  
Frequency Options**

The 80376 is available only in a 100-pin PQFP package and is currently offered with core frequencies of 16 or 20 MHz. It has the same execution timing as the i386SX, so if both chips were able to execute the same software at the same clock rate, the performance of the two would be the same.

## 5.5 The Intel i386SL “SuperSet” Microprocessor

The i386SL is a fully static derivative of the 386 family for power-conscious applications in portable lap-top, notebook, and subnotebook (“palm-top”) PCs. Table 5-11 summarizes the features and specifications of the i386SL microprocessor.

Product Name	Intel i386SL
Introduction Date	October 1990
Prognosis	New design activity being discouraged
Device Integration Level	Static i386SX integer unit core and PMMU; On-chip cache tags and control logic; Direct system memory controllers and drivers; Direct ISA backplane controllers and drivers; On-chip power-management logic
CPU Architecture Level	Full 386 integer instruction set plus Intel SMM (System Management Mode) extensions
Core Technology	386 core redesigned for static and low-voltage operation
Pinout	Custom
Data Bus Width	ISA-compatible 16-bit system data bus Separate 16-bit cache data bus
Physical Addressability	16 megabytes accessible via ISA bus Separate interface for local DRAM and cache
Data-Transfer Modes	Supports multiple transfer types including standard ISA bus, high-speed local bus, system SRAM and DRAM control sequencing
Cache Support	Internal control logic and tags for optional off-chip cache Configurable for 16K, 32K, or 64K bytes One-, two-, or four-way set associative Write-through operation only
Floating-Point Support	Optional external i387SX FPU
Voltage Options	3.0 V to 3.6V or 4.5 V to 5.5 V
Frequency Options	20- or 25-MHz core operation @ 5 V 16- or 20-MHz core operation @ 3.3 V
Clocking Regime	Core operating frequency equals a programmable fraction of clock input
Active Power Dissipation	3.5 W @ 5.0 V & 25 MHz; 1.1 W @ 3.3 V & 20 MHz
Power-Control Features	Static operation; programmable frequency subdivider
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Transistor Count	850,000 transistors (CPU); 260,000 (I/O)
Die Size	508 $\times$ 516 mils (CPU); 416 $\times$ 508 mils (I/O chip)
Package Options	196-pin PQFP or 227-lead land grid array
Other Features	First Intel processor to support SMM I/O drivers directly compatible with ISA bus On-chip support for LIM memory paging

Table 5-11. Intel i386SL feature summary.

**Features** The i386SL was part of a family of devices that crammed the processor and peripheral circuitry of an entire ISA-compatible PC into a highly integrated chip set. While these features made the part attractive for ultrasmall, low-power battery-operated computers, they were of minimal value in desktop applications.

As even the low-power portable PC market began shifting toward 486-class processors, Intel began de-emphasizing the i386SL in favor of static implementations of the 486 family (see **Chapter 6: Intel 486 Microprocessors**). While some i386SL-based notebook computers were still being sold during 1994, new design activity has ceased. The device is included in this report for historical and comparison purposes.

**System Overview** The i386SL chip-set partitions an entire generic 386-based personal computer system into a handful of dedicated chips. Figure 5-6 shows how system functions divide among chip-set elements.

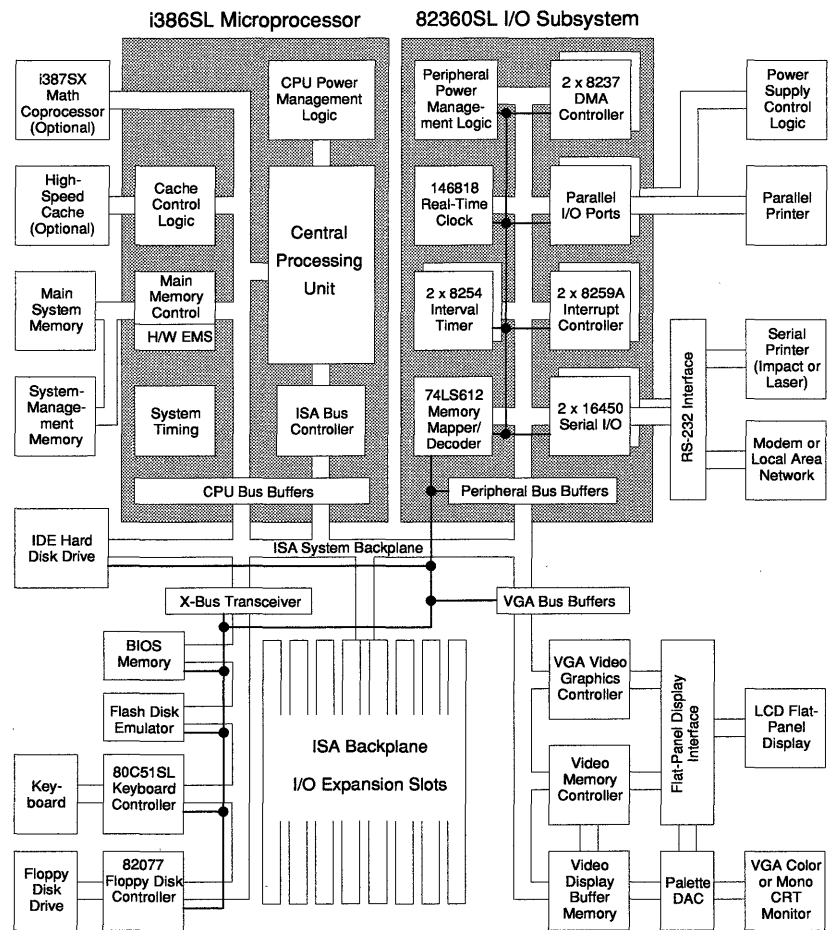


Figure 5-6. Intel i386SL functional system partitioning.

The i386SL device itself contains a static 386-class CPU core plus control logic and high-current drivers for an ISA-compatible system bus interface; control logic and tags for an optional external cache memory; and interfaces for an SRAM- or DRAM-based main memory system.

An array of software-configurable variable-frequency clock generators within the i386SL allows the CPU and its peripheral chips to run at a variety of speeds, allowing system software to fine-tune system power consumption under a variety of situations. An array of programmable counters monitors software I/O activity to system peripherals, making it possible for system software to intelligently decide when it is safe and prudent to disable or re-enable display back lights, disk drives, and other power-hungry peripheral subsystems.

Numerous system peripheral and I/O devices are incorporated into an auxiliary support chip designated the 82360SL. These include all the DMA controllers, timer/counters, interrupt controllers, serial ports, parallel I/O ports, and decoding logic found in a standard ISA-based PC clone.

A third chip, designated the 80C51SL, performs custom keyboard interface functions. This device contains a low-power eight-bit general-purpose microcontroller based on the venerable old 8051 (motto: "Fifteen years without a major redesign, and still going strong"); an interface port through which the 8051 core can receive commands from and return data to the i386SL CPU; a program ROM for user-defined control algorithms; and a gate array that may be configured as needed to perform I/O and control logic functions.

A fourth chip typically provides a standard VGA interface to an LCD display or CRT. Because the interface requirements for different displays depend greatly on the display type and size selected, this last chip generally varies from one application or system architecture to another.

### **Cache Support**

The i386SL takes a novel approach to cache design. The presence of cache in a battery-operated system actually tends to preserve battery life, since the same effective performance can be obtained using a correspondingly slower system clock. Moreover, it takes a considerable amount of power to access system memory continuously; to the extent that a cache allows system memory to remain idle, power consumption will fall.

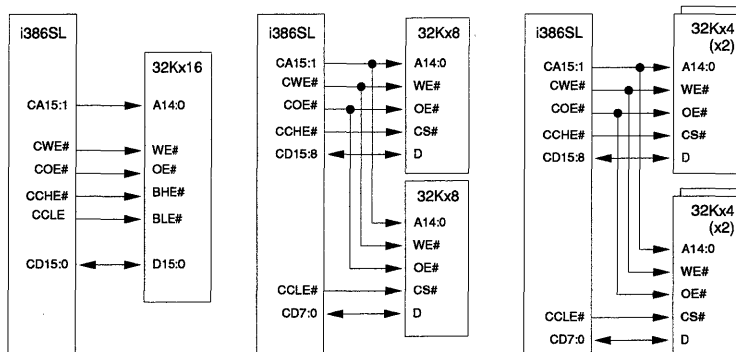


Figure 5-7. Intel 386SL direct cache support.

The i386SL contains cache control logic and tag memory but does not contain any cache data arrays. Attaching one, two, or four external SRAM devices (see Figure 5-7) enables the i386SL to support cache arrays as large as 64K bytes. The cache is unified (i.e., it buffers both instructions and data), has a two-byte line size, and can be configured to be direct-mapped, two-way, or four-way set associative.

Note that no random-logic “glue” is required for any of the cache configurations shown in Figure 5-7; the i386SL CPU’s control and data pins can be configured through software to connect directly to corresponding pins on the SRAM chips.

### Floating-Point Support

The i386SL does not contain any direct support for floating-point operations; however, it may be used with an optional 386SX-class floating-point coprocessor. The i386SL CPU generates the external clock signal needed by the off-chip i387SX FPU. The i386SL minimizes FPU power consumption by automatically slowing or stopping the FPU clock except when floating-point operations are in progress.

### System Interface Description

Much of the complexity of the i386SL family comes from the plethora of system architectures, memory configurations, I/O options, and backplane driver requirements. The CPU contains software-configurable control logic to support any of a wide range of design alternatives.

System memory can be up to a total of 32M bytes and may be built from SRAM or DRAM devices of varying capacity. Depending on the memory devices installed, certain i386SL pins are software configurable to emit demultiplexed address lines, read and write control strobes, and DRAM RAS and CAS signals.

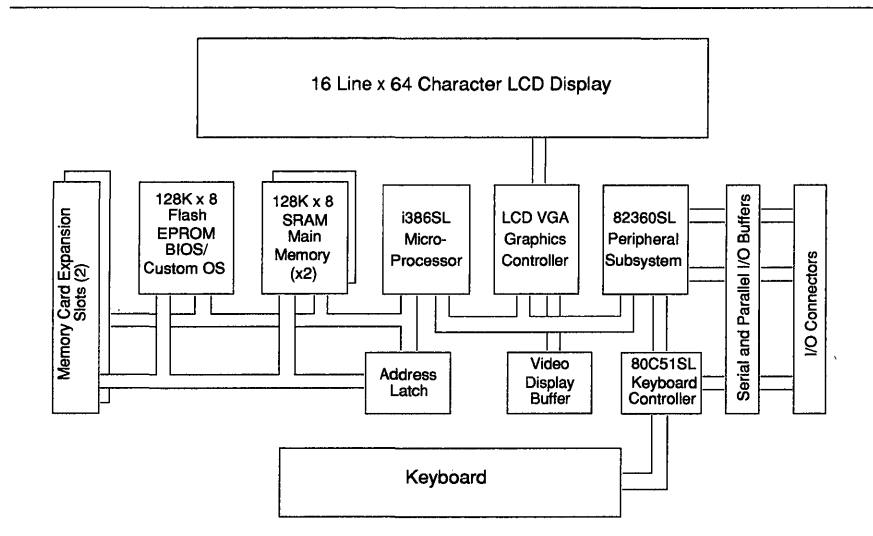


Figure 5-8. Intel i386SL 10-chip minimum system design.

The upshot of all this configurability is that systems may be designed with an exceedingly low chip count and no external glue. Figure 5-8 shows a simple checkbook-size computer that incorporates 384 Kilobytes of system memory, can run standard DOS applications, supports both ISA-standard and PCMCIA-type expansion boards, and can be built with a total of just 10 ICs—including system memory.

It's beyond the scope of this report to describe the i386SL system interface, signal names, and configuration options; suffice it to say that the Intel data sheet that summarizes the hardware interface of the i386SL and 82370SL devices is 150 pages long. A table that identifies simply the name, location, and I/O attributes of each signal pin runs more than 10 pages, while another table containing a brief summary of each pin's function consumes more than 15 pages.

### Package and Frequency Options

Intel offered the i386SL in either a 196-lead PQFP package or a 227-lead land-grid-array (LGA) package. The 82360SL support chip was available only in a 196-pin PQFP. Each chip was offered in 5-V versions that supported CPU core frequencies up to 20 or 25 MHz, or in 3.3-V versions that supported core frequencies up to 16 or 20 MHz. Just to make the purchasing-decision process even more convoluted, lower-cost versions of each CPU were available in which the cache-control circuitry was disabled.

## 5.6 Futures

It has long been part of Intel's official corporate charter that the company will not compete in a market unless it can either dominate the industry or run a strong second, with opportunities to advance. A corollary of this policy is that when market conditions change and Intel no longer finds it sufficiently lucrative to sell an aging product, the company retreats gracefully from the market.

As competition began to develop for 386-class processors from AMD, Cyrix, et al (see following chapters), Intel appeared to withdraw from the desktop 386 market. No new design activity is under way for 386-based desktop or portable PCs, although Intel entered into a (now-defunct) cross-licensing agreement with VLSI Technology for the i386SL processor core.

This is not to say Intel has discontinued 386 production. Instead, it has begun pursuing new 386 markets outside the conventional PC arena. One of these is in 32-bit embedded computing. While the 386 architecture has no inherent advantages over competing processors for embedded systems—including Intel's own i960 family—the widespread availability of software tools, compilers, debuggers, operating systems, utility libraries, and the like makes it relatively easy for embedded system suppliers to design with these parts. Also, any of the 110 million or so 386- and 486-based PCs now in use can serve double duty as a development system, software testbed, and debugger for 386-based designs.

### **Geopolitical Pawns?**

Another new market for 386 processors may be opening in the Far East. In April of 1994 the government of China announced that the Intel 386 microprocessor had been selected as the core of the next generation of small business and consumer computers. To save manufacturing and transportation costs (!), Intel will be shipping huge volumes of bare 386 die overseas, to be packaged and assembled into systems on the Chinese mainland. As of this writing, anticipated production volumes and other details of this deal had not been divulged. One has to wonder whether third-world markets are attractive to Intel on their own merit, or merely as a way to keep AMD from exploiting Asian markets to recoup its 386 development costs.



## 5.7 For More Information...

Additional technical information on the Intel 386 product lines may be found in the following publications:

### Vendor Publications

- 1: *386 SL Microprocessor SuperSet Programmer's Reference Manual*. Intel Corporation, 1990, order #240815-001.
- 2: *80386 System Software Writer's Guide*. Intel Corporation, 1988, order #231499-001.
- 3: *Intel386 SL Microprocessor SuperSet Data Book*. Intel Corporation, 1992, order #240814-004.
- 4: *Introduction to the Intel386 SL Microprocessor SuperSet: Technical Overview*. Intel Corporation, 1991, order #240852-002.
- 5: *Microprocessors Data Book Volume I: Intel386, 80286, and 8086 Microprocessors*. Intel Corporation, 1994, order #230843-011.

### Microprocessor Report Articles

- 6: *Intel's P9 Could Make 286 Architecture Obsolete\**. MPR vol. 1 no. 1, 9/1/87, pg. 1. (Cover story.)
- 7: *Details of 80376 Begin to Emerge*. MPR vol. 1 no. 4, 12/87, pg. 3. (Most Significant Bits item.)
- 8: *Intel's 80376 Provides Lower-Cost 386 Replacement for Embedded Control\**. MPR vol. 2 no. 4, 4/88, pg. 9. (Feature article.)
- 9: *Intel Christens P9 the 80386SX\**. MPR vol. 2 no. 6, 6/88, pg. 1. (Cover story.)
- 10: *Intel Drops SX Price to Crush 286*. MPR vol. 3 no. 2, 2/89, pg. 2. (Most Significant Bits item.)
- 11: *386SX Price Drops, but 286 Sales Remain Strong*. MPR vol. 3 no. 10, 10/89, pg. 2. (Most Significant Bits item.)
- 12: *More Bugs in the 486*. MPR vol. 4 no. 2, 2/7/90, pg. 4. (Most Significant Bits item.)
- 13: *Intel Finally Moves 386SX to 20 MHz*. MPR vol. 4 no. 2, 2/7/90, pg. 5. (Most Significant Bits item.)
- 14: *More 386 Family Parts Coming*. MPR vol. 4 no. 3, 2/21/90, pg. 4. (Most Significant Bits item.)
- 15: *"Smart Cache" Reduces 386 Cache to Single Chip\**. MPR vol. 4 no. 10, 5/30/90, pg. 6. (Feature article.)

- 16: *Processors and PC Chip Sets Merge\**. MPR vol. 4 no. 13, 8/8/90, pg. 1. (Cover story.)
- 17: *386SL Brings 386 Power to Notebook Computers\**. Michael Slater, MPR vol. 4 no. 18, 10/17/90, pg. 1. (Cover story.)
- 18: *SuperSet Provides Transparent Power Management*. Michael Slater, MPR vol. 4 no. 19, 10/31/90, pg. 12. (Feature article.)
- 19: *Intel Licenses Power Management Chip*. MPR vol. 4 no. 21, 11/14/90, pg. 4. (Most Significant Bits item.)
- 20: *Intel's 386SL Will Not Support SRAM Initially*. MPR vol. 4 no. 21, 11/14/90, pg. 4. (Most Significant Bits item.)
- 21: *Intel Loses 386 Trademark\**. Michael Slater, MPR vol. 5 no. 5, 3/20/91, pg. 1. (Cover story.)
- 22: *Intel offers New 386SL, 486SX Versions*. MPR vol. 5 no. 18, 10/2/91, pg. 4. (Most Significant Bits item.)
- 23: *Intel Claims Am386 Infringes PLA Copyright\**. Michael Slater and Rich Belgard, MPR vol. 5 no. 20, 10/30/91, pg. 11. (Feature article.)
- 24: *Intel Counters with SL*. MPR vol. 6 no. 2, 2/12/92, pg. 5. (Most Significant Bits item.)
- 25: *The Intel System Management Mode\**. Simon Ellis, MPR vol. 6 no. 2, 2/12/92, pg. 16. (Feature article.)
- 26: *Intel Announces Its First 3.3-V Processors*. MPR vol. 6 no. 5, 4/15/92, pg. 4. (Most Significant Bits item.)
- 27: *Intel Samples 3.3-V 386SL*. MPR vol. 6 no. 8, 6/17/92, pg. 4. (Most Significant Bits item.)
- 28: *Intel Forges 386SL Deal With VLSI Technology*. MPR vol. 6 no. 10, 7/29/92, pg. 4. (Most Significant Bits item.)
- 29: *Intel Slashes 386SL Prices*. MPR vol. 6 no. 11, 8/19/92, pg. 4. (Most Significant Bits item.)
- 30: *Intel Redesigns 386 for Embedded Market*. Linley Gwennap, MPR vol. 7 no. 14, 10/25/93, pg. 22. (Feature article.)
- 31: *PDAs Begin Shipping in 1993*. Linley Gwennap, MPR vol. 8 no. 1, 1/24/94, pg. 18. (Feature article.)
- 32: **Other Technical References** 80386 *Technical Reference*. Edmund Strauss, Brady Books, 1987, ISBN 0-13-246893-X.

33: *Marketing High Technology*. William Davidow, Free Press, 1986. (Case histories of Intel marketing strategies.)

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



# 6

## Intel 486 Microprocessors

---

Whereas the Intel i386SX and i386DX devices were tremendously successful from a business perspective, and while the 32-bit architecture they embodied overcame the limitations of their 16-bit forebears, and while the complexity and performance of each part was quite impressive for its day, both devices left much to be desired in terms of their implementations. As semiconductor technology advanced and the number of transistors available to chip designers increased, it became possible to build processors with both better performance and higher integration levels than the original 386 family members.

A completely new implementation of the 386 architecture resulted in the i486DX device, introduced in 1989. In the intervening years Intel has introduced more than a dozen major derivatives of the i486DX core, and another dozen minor updates. Since they and many of the other processors described in later chapters of this report take advantage of a number of 486 implementation techniques, this chapter contains a detailed review of the basic 486 core design followed by a description of each of the current Intel 486 family members.

### **Intel 486 Family Overview**

In addition to being able to run 386 programs three to five times faster than an i386DX, the i486DX contains an 8-Kilobyte on-chip instruction and data cache, a complete 387-class floating-point unit, and a more efficient system interface.

The 486 family is characterized by a number of improvements and enhancements over 386-class products. Chief among these are the 486 family's higher levels of integration and performance, and several minor additions to the 386 architecture.

Feature	Device Comparison
Integration Level	The 486 combines the integer and FPU facilities of the i386DX and i387DX onto one chip, along with an 8-KB cache.
Architecture	The 486 adds eight new instructions for CPU configuration and control and multiple-processor communications to the 386 repertoire, plus the 387 instruction set.
Control Registers and Flags	New control register bits and previously undefined bits in the memory descriptor tables configure processor mode, cache operation, and external memory cacheability.
Memory Management Unit	A new page-protection feature improves support for Unix and other multitasking OSs. A new control flag optionally traps execution on suboptimally aligned data objects.
Execution Pipeline	The 486 contains a heavily pipelined integer execution unit that typically requires one-third to one-half as many clock cycles as the 386 for most integer instructions.
Instruction/Data Cache	The 486 contains an 8-KB instruction/data cache on-chip. Instruction requests and data loads and stores that hit within the cache can complete in a single clock cycle.
Floating-Point Unit	The 486 includes a full 387-compatible FPU on chip. Data passes between the IEU and FPU through dedicated buses for somewhat better performance.
System Interface	The 486 supports efficient burst-mode instruction fetches and data loads, support for a second-level cache, optional parity on the databus, and several features to simplify PC design.
Power Management	Since late 1993, all new Intel 486 processors have provided power-management features including a static core, low-power modes, and system management mode.

Table 6-1. Differences between Intel 386 and 486 microprocessors.

Table 6-1 lists the general areas in which 386- and 486-class processors differ. These areas are discussed below.

### Architecture Extensions

The 486 instruction set and architecture are a superset of those originally defined by the i386DX microprocessor and the i387DX FPU, enhanced to include a number of new registers, new instructions, and new operating modes.

The user-mode programming model for the Intel 486 family includes each of the integer working registers, control and status registers, and FPU registers originally defined by 386-family products. These registers are shown in Figure 6-1.

**Programming Model Extensions.** The 486 architecture also extends the original 386 system-mode register set by implementing several new control and status registers and several new bits and bit fields within existing system registers and memory-based data structures. Two previously reserved bits in control register CR0 now enable the cache replacement and write-through facilities. Five new 32-bit test registers have also

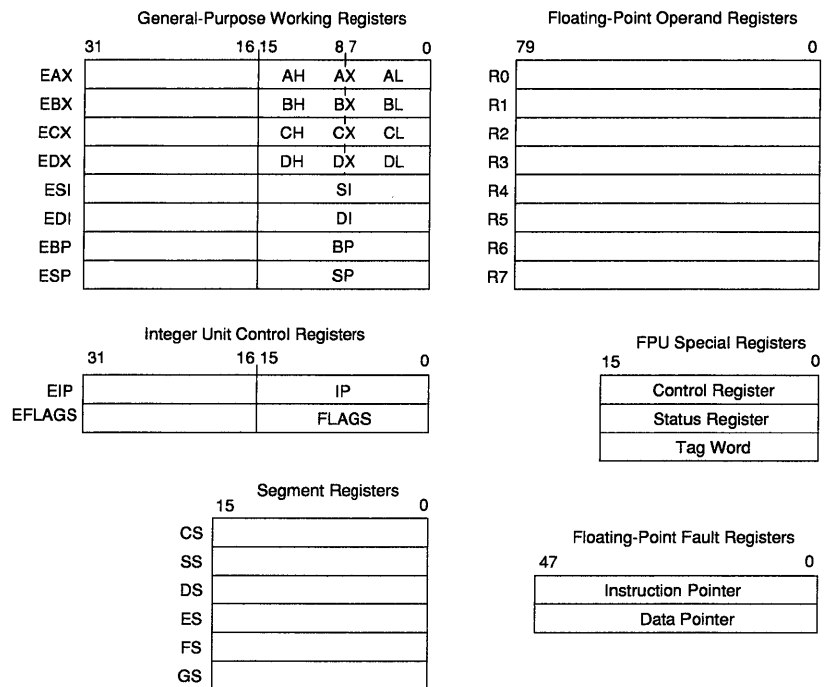


Figure 6-1. Intel 486 programming model.

been added to let the OS test the operation of the cache memory and tag data arrays. New and revised control registers appear in Figure 6-2. Light gray shading indicates register fields defined by the 386 architecture. Darker gray fields are reserved by Intel for future expansion.

A newly defined bit in each page-table entry controls cacheability on a per-page basis. If the PCD (page cache disable) bit for a particular page is set, internal caching of data from that page will not be allowed. If PCD is cleared, internal caching is enabled. The state of the PCD bit for the referenced page is copied out to an external pin during every external memory access. Off-chip logic can monitor this pin to prevent a second-level cache from collecting noncacheable data.

A second bit in each memory page-table entry controls whether a second-level cache implements a write-through or write-back policy for each page. The PWT (page write-through) bit is copied to an output pin during every memory cycle. PWT is not monitored by the internal cache, since all writes are write-through.

Since the instruction set and programming model of the 486 encompass all of the instructions, registers, and system data

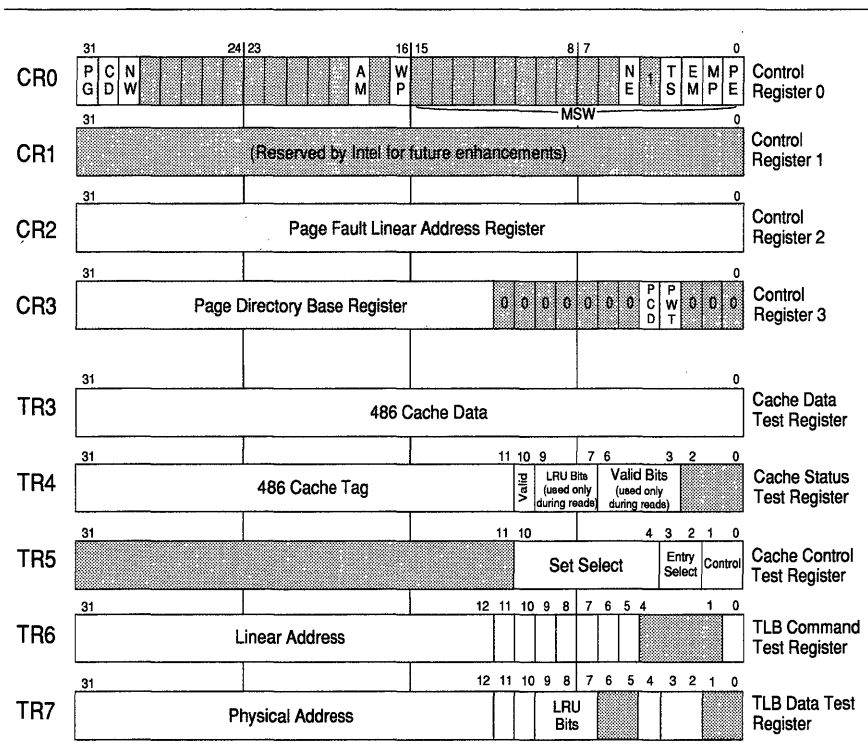


Figure 6-2. Intel 486 programming model additions.

structures of its predecessors, existing 8086, 80286, and 386 operating systems and application programs generally run unmodified on compatible 486-based hardware. Each of the new register fields and data structures is visible only to protected-mode operating system code and is thus transparent to applications programs. System-initialization code must be revised, however, to enable cache operation, and programs that contain software delay loops may need to be adjusted to compensate for the faster instruction-execution rate.

**Instruction Set Additions.** Six new instructions in the 486 improve the performance of multiple-processor-based system designs and control the new on-chip cache and optional external caches. SL-enhanced family members implement a seventh instruction for power-management software. Table 6-2 describes these instructions.

The BSWAP instruction reverses the order of the four bytes in a 32-bit register so a 486 can share data structures and on-line databases more easily with “big-endian” processors in networked installations.



Instruction	Mode	Operation
BSWAP	User/ System	Byte swap. Reverse byte order within register
XADD	User/ System	Atomic (indivisible) exchange and add
CMPXCHG	User/ System	Atomic (indivisible) compare and exchange
INVD	System	Invalidate data cache
WBINVD	System	Perform write-back cycle and invalidate cache
INVLPG	System	Invalidate TLB page entry

Table 6-2. Intel 486 instruction set additions.

The XADD (exchange-and-add) and CMPXCHG (compare-and-exchange) instructions perform atomic (indivisible) memory read/modify/write sequences in order to simplify software semaphores in multiprocessing applications without having to invoke OS functions or disable interrupt processing.

INVD invalidates the internal program and data cache for software testing or system verification purposes and initiates a special bus cycle to flush any external cache systems.

WBINVD invalidates the internal cache and initiates two special bus cycles. The first instructs an external copy-back cache (if present) to write any dirty (modified) cache lines back to main memory; the second flushes the external cache.

The INVLPG instruction invalidates the entry for a specific page within the on-chip TLB.

## Execution Pipeline

The 486 was the first x86 microprocessor to contain a pipelined instruction execution unit, on-chip cache, and several enhancements to the 386 architecture, as shown in Figure 6-3. This section describes the functional units that make up the 486 execution pipeline and how they interact to achieve single-cycle execution of many instruction types.

**Pipeline Overview.** The 486 pipeline includes five stages: prefetch (PF), two decode stages (D1 and D2), execution (EX), and register-file write-back (WB). A series of single-cycle instructions will be fully overlapped as they pass through the pipeline, as shown in Figure 6-4.

Unlike early pipelined microprocessors, the 486 is not restricted to the simple, lock-step progression of instructions through the pipeline. Instructions may consume a varying number of clock cycles in each stage. Interlocks prevent each stage of the pipeline from advancing unless later stages will be ready to absorb the resulting data when it arrives. Conversely, later pipeline

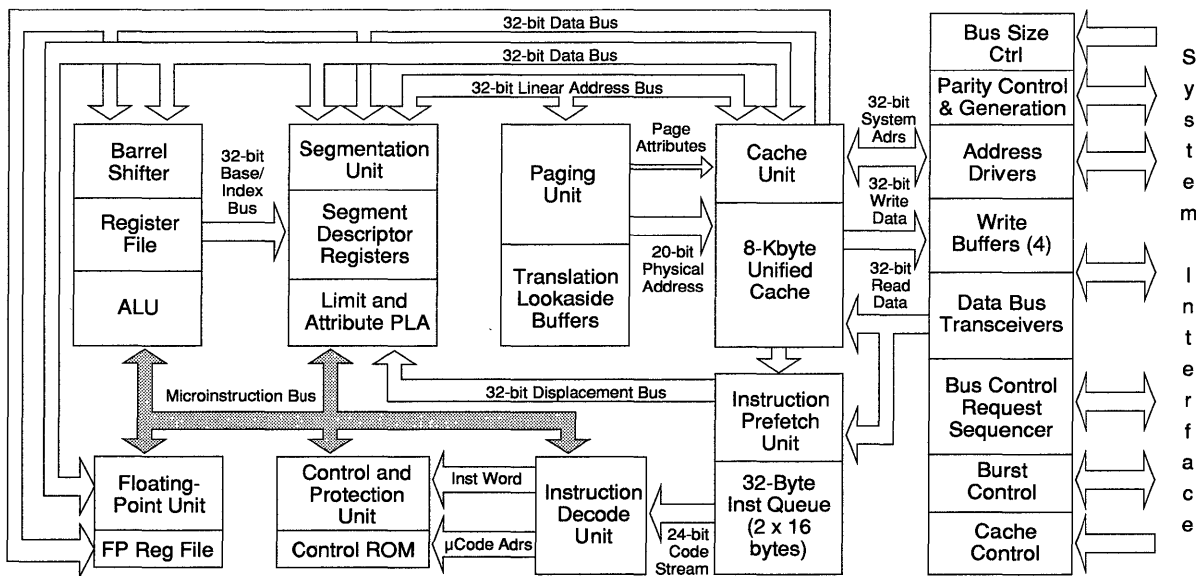


Figure 6-3. Intel 486 microarchitecture.

stages may continue to advance while earlier stages are busy or blocked.

The PF stage prefetches instructions from the cache or main memory into two 16-byte instruction prefetch buffers organized as a 32-byte circular queue. The PF stage tries to stay several cycles ahead of the execution unit, so each instruction will generally be retrieved several clock cycles before it is due to begin executing. The instruction buffers are physically implemented as a strip of silicon between the two halves of the I/D cache. Each holds an entire 16-byte cache line, so together they can hold between four and ten full instructions.

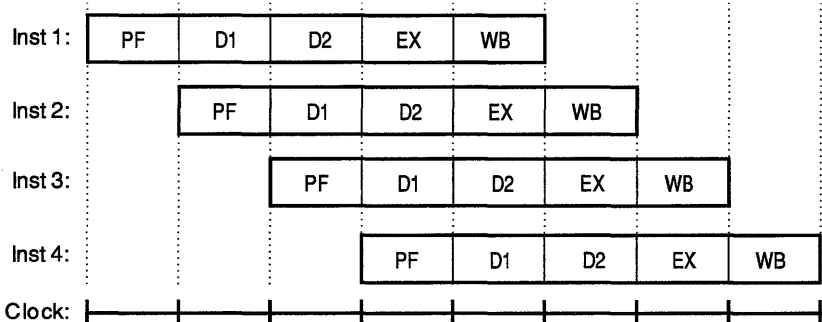


Figure 6-4. Intel 486 pipeline stages.

The D1 pipeline stage “cracks” the instruction encoding. Logic attached to the instruction prefetch buffers extracts the opcode, constant, and displacement fields of each instruction in parallel as needed, regardless of their alignment. D1 logic examines the instruction opcode, determines the instruction class into which it falls, and determines what operation will later be performed by the execution stage. D1 also determines the entry point within a microinstruction ROM that contains the control word for the first execution cycle; if the instruction requires a memory address calculation, then D1 also retrieves the information needed to compute the address for use by the segmentation unit.

The D2 stage expands each instruction into the appropriate control signals for the ALU. For single-cycle instructions, this is simply a function of the original opcode bits. The D2 stage also controls the computation of the more complex addressing modes.

During the EX stage, the integer unit ALU performs the appropriate calculation. The 486 pipeline may take multiple EX clocks to complete a complex macroinstruction or to manipulate complex data structures.

The WB stage dispenses with data produced during the preceding EX stage. If the current instruction modifies memory, the computed value is sent to the cache and to the bus interface write buffers. On cache misses, the internal cache is left unchanged.

The 486 register file has six separate ports, so different pipeline stages may retrieve the data they need without interfering with each other. The design also includes logic for register bypassing. Hard-wired comparators detect whether either of an instruction’s source-register operands was modified or loaded during the preceding instruction, in which case the register file input bus is routed (“bypassed”) directly to the ALU. This eliminates clock cycles that would otherwise be consumed writing data to the register file.

The pipeline treats override prefix instructions differently from “real” instructions. When the D1 stage detects a prefix instruction, it sets a corresponding flag and begins decoding the next instruction. Each prefix byte therefore adds one extra D1 clock to the instruction it modifies. When the primary opcode field is detected, the override flags are passed on down the decode/execute pipe and cleared.

However, prefix instructions do not require any processing in the D2 and EX stages. As a result, D1 can absorb a series of prefix bytes while D2 completes an earlier multicycle instruction. In such cases, prefix codes execute in effectively zero clocks, since they do not delay the time at which the instruction they modify can begin.

**Data Retrieval Pipeline.** The execution unit can perform register-to-register operations in a single clock. The more challenging task is to incorporate complex address calculations, virtual memory translation, and data retrieval into the pipeline without slowing it down. These functions are performed by a second two-stage data-retrieval pipeline—involving the segmentation unit, paging unit, and cache—that operates in parallel with the decode and execution units described above.

The data-retrieval pipeline contains logic to compute virtual and physical memory addresses, access the cache, and control the external bus. The address calculation unit has dedicated ports into the register file and can retrieve index register values without disrupting arithmetic instructions. A dedicated port for the stack pointer reduces the clock count for subroutine linkage and common stack instructions.

By the time an instruction leaves the D1 pipeline stage, memory addressing information has been passed to the Segmentation Unit. Resident copies of the segment descriptors supply segment base and limit values. A dedicated port from the register file provides the base or index register contents. The displacement constant, if needed, is extracted from the instruction stream.

During the execution pipeline D2 stage, the segmentation unit combines base register and displacement components to determine a segment-offset value, which is compared to a segment-size register to detect limit violations. A separate adder simultaneously computes the full 32-bit linear address, i.e., base register plus displacement plus segment-base. Four-part addressing modes and those that combine a base register with a shifted index register consume an extra clock cycle in the D2 pipeline stage as the second register is retrieved.

While the main instruction pipeline is in its first execution cycle, the data retrieval system comes into play. If paging is enabled, the 32-bit linear address produced by the Segmentation Unit must be interpreted as a virtual address. During the EX clock, the high-order bits of the linear address

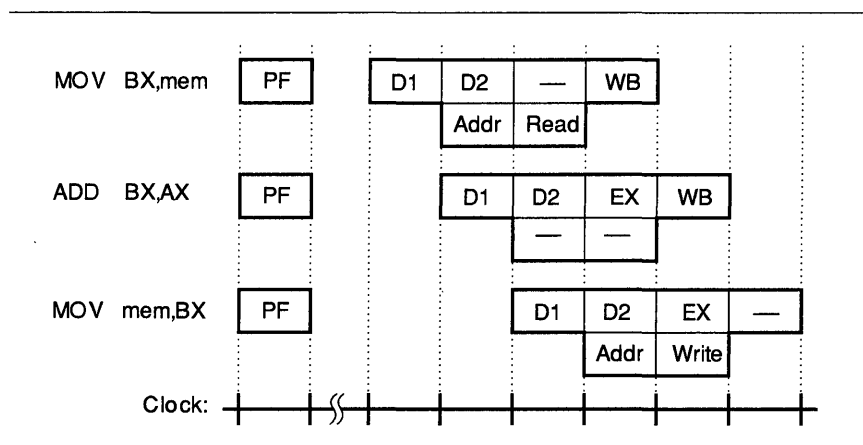


Figure 6-5. Intel 486 pipeline timing for simple operations.

computed during D2 are sent to the paging unit and compared in parallel to the tag bits of the TLB entries. Assuming a page hit, the TLB returns the corresponding physical address bits of the corresponding page during the EX clock.

Meanwhile, the linear address computed during D2 is sent to the cache, and the four sets of cache tags enabled by address bits A10..A4 are retrieved. The four words of data in the selected cache line are retrieved, and comparators check whether the tag bits for any of the cache lines match the corresponding bits of the physical address. If so, the corresponding cache data passes through another multiplexer, and the properly aligned data emerges.

The end of the EX cycle marks the start of the WB cycle. If a load instruction had initiated a data access, the cache data will be saved during the WB cycle. Thus, a load instruction completes with the same timing as a simple register-to-register add. If the next instruction uses that register as a source operand, bypass gates send the cache data directly to the ALU. The next instruction can use the fetched data immediately, without having to perform a register file lookup cycle.

**Execution Timing.** Figure 6-5 shows the respective 486 pipeline stages for a series of three instructions. The first is a simple memory load of data assumed to be present in the cache. The second performs a register-to-register add, using the just-loaded data. The third instruction stores the computed result to memory. All three instructions are prefetched together, and each requires a single clock cycle in each pipeline stage.

Figure 6-6 shows a register-to-memory ADD instruction with the same overall effect as the sequence in Figure 6-5. While the single-instruction form still takes three clock cycles to complete, it requires just four instruction bytes rather than ten, does not corrupt any temporary registers, and frees up the D1 stage early to begin decoding the next instruction.

**Branch Processing.** Control transfer instructions (i.e., jumps, calls, and conditional branches) are detected in the D1 stage of the main execution pipeline. The segmentation unit computes the target address during the D2 stage and retrieves the cache line containing the target instruction during the first EX stage. Meanwhile, the opcode multiplexer in the IPU adjusts its shift-position count so opcode bytes of the target instruction will emerge from the IPU, fully aligned, and enter the first decoder stage at the start of what would otherwise be the WB cycle of the branch instruction. Jumps and calls thus consume three cycles in the execution pipeline.

Conditional branch instructions present a challenge to heavily pipelined machines, since CPU flag settings may be affected by earlier instructions that have not yet completed when the branch instruction begins. When the D1 stage decodes a conditional branch instruction, the 486 core initiates a “speculative” prefetch of the target instruction on the assumption that the branch will indeed be taken.

Once previous instructions have completed, if the state of the CPU flags does indeed match the branch condition anticipated, the instruction that was the target of the speculative prefetch will already have been retrieved. The branch instruction can

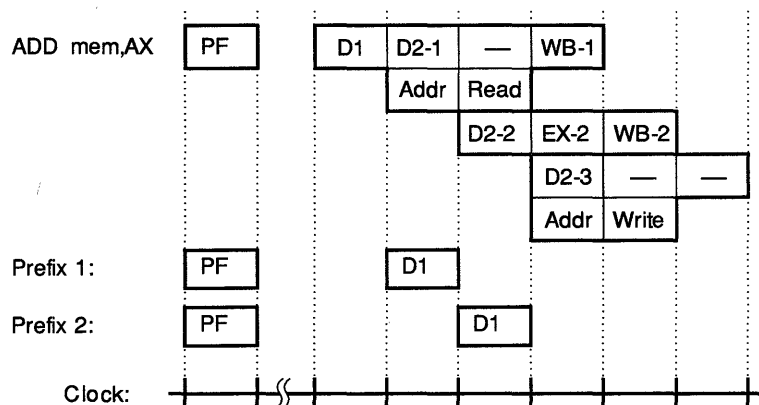


Figure 6-6. Intel 486 pipeline timing for reg-to-mem operations.

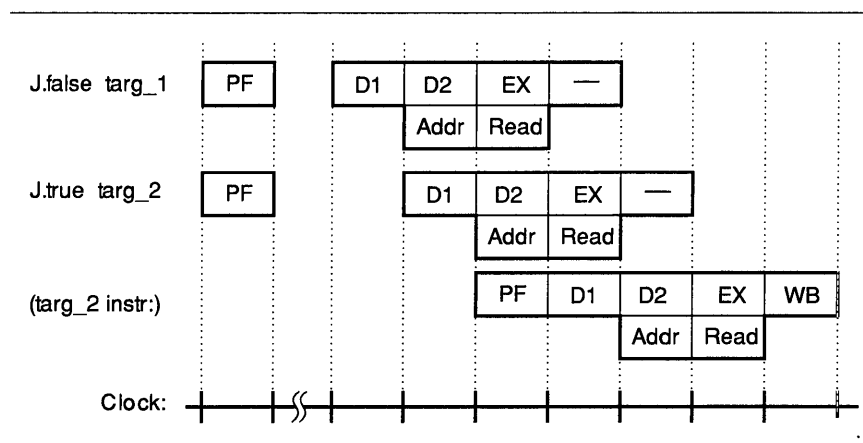


Figure 6-7. Intel 486 pipeline timing for branch operations.

then complete in three clock cycles with the same timing as a simple jump instruction.

If the condition tested proves false, the prefetched instructions will be abandoned, and the instructions immediately following the branch—generally still present in the prefetch queue—continue through the pipeline. Untaken branches therefore execute in a single clock cycle. Figure 6-7 shows the execution of two back-to-back conditional branches. The first branch falls through, while the second branch is taken.

As sequential instruction execution proceeds, each half of the prefetch queue will periodically empty itself. Prefetch logic attempts to refill empty buffers with the next sequential instruction block. If the cache misses, prefetch logic requests a burst of instruction fetch cycles from external memory. Sequential prefetches are performed in ascending order, with each word written to both the prefetch buffer and the cache as it is received. In the meantime, the IEU can generally keep busy processing instructions that remain in the alternate prefetch buffer. This means performance is minimally impaired, even when external prefetch cycles are required.

### Instruction/Data Cache

The standard 486 processor core contains an 8-Kilobyte unified instruction and data cache. The cache has a four-way, set-associative organization, with 128 sets. The line size is 16 bytes. Cache accesses generally overlap other aspects of instruction execution such that memory operations seldom stall pipeline operation.

**Unified Design.** The i486DX cache stores both code and data in the same physical array. Intel claims this provides more efficient cache utilization than separate 4-Kilobyte code and data caches, for example. Programs that deal with large data structures may use more of the cache for data storage, while those with minimal data leave more cache available for code.

Unifying the cache also ensures compatibility with existing 8088 and 386 software. While the practice of an application program modifying its own code is not recommended, the 386 architecture does not prohibit it. In fact, many standard operating systems rely on run-time code modification for added flexibility and speed. For example:

- Program overlay loaders, used to overcome the 640K address limitation of 8088- and 80286-based PCs, must adjust program and data address references to match the program's location in main memory.
- Programs that perform floating-point arithmetic often contain operating system trap instructions in lieu of floating-point opcodes. As each trap is encountered at run time, the OS backfills the program with either a coprocessor instruction or a call to an equivalent floating-point library subroutine, depending on whether an FPU is available.
- Microsoft Windows and other OSs with graphical user interfaces (GUIs) often build small, highly optimized graphics routines on the stack, and then call them to produce the fastest possible screen updates.

Unified cache designs, on the other hand, are less effective if code and data fetches must compete for accesses to the cache, stalling instruction execution. On average, instruction prefetches occur only every 5 to 10 clock cycles, and data accesses occur only every third cycle or so; simultaneous requests for instructions and data seldom occur.

When simultaneous instruction prefetches and data transfers do collide for use of the cache, the data access is given higher priority. The execution unit can generally continue processing instructions from the prefetch buffer for a cycle or two until the data access is completed.

The cache is physically mapped. The segmentation and memory-paging mechanisms of the 8086 and 386 architectures allow operand "aliasing," in which different linear addresses



may access the same physical location. Physical cache mapping guarantees that existing software that modifies memory-based variables will also update aliased copies of the same datum.

**Write-Through Operation.** Operations that modify memory write through the cache. On cache hits, the cache and system memory are both updated with new data; on a cache miss, only the system memory is updated. The cache is therefore never “dirty,” that is, cached data values always match those in main memory.

While write-through caches are generally thought to be less efficient than copy-back designs, they do provide several mitigating advantages, especially in single-processor systems. They are simpler to design and help avoid several potential memory system bottlenecks. An entire cache line is guaranteed to be valid or invalid collectively. The processor need not allocate a new line on data writes, nor must it fill a partial line by reading system memory before writing new data. Flushing the cache consists of marking all tags as invalid; it is not necessary to copy modified cache locations back to main memory when a process context switch occurs. And compatibility issues involving memory-mapped peripherals are simplified.

While write-through operation is sufficient for personal computers and other single-processor system designs, it may be less efficient in a multiple-486 system that shares a single memory system or I/O bus. The shared bus could then be saturated by the write operations of the various CPUs.

Such configurations—minicomputers, process servers, etc.—would likely have a second-level write-back cache between each 486 subsystem and the system backplane. Processors in the 486 family provide instructions, data structures, and control signals to support second-level caches with both write-through and copy-back allocation policies. See the individual device descriptions below for further details.

The cache uses a simplified least recently used (LRU) replacement policy. Logic splits the four candidates for replacement into two pairs. Status flags keep track of which of the two pairs and which line within each pair was least recently used.

**Cache Efficiency.** Intel claims most programs have a measured hit rate of about 96% for both instructions and data, depending on program size and the complexity of the program mix. In large multitasking systems, the hit rate drops to about

92%, since cached instructions and data for a given task tend to get corrupted whenever a task is swapped out.

Microprocessors with on-chip cache pose a particular challenge during the program development and debugging phases. The vast majority of all program and data fetches are satisfied by the internal cache, so traditional system-level debugging techniques based on logic analyzers and bus-trace collection logic are largely ineffective in debugging 486-based systems.

Software can therefore configure a 486 to disable its internal cache, in which case all program or data references are forced to the external bus in much the same manner as a 386 microprocessor. This allows external logic to trace program execution, albeit at a greatly reduced speed.

### **System Interface**

Like the i386DX, the 486 system interface connects to the rest of its system via 32-bit parallel address and data buses. The control signals, cycle types, and transfer timing are very similar. Compared to the 386 family, though, the 486 enhances its system interface in several respects. The 486 supports a multiple-word burst-transfer mode for instruction and data fetches, automatic parity generation, and multiple-processor cache coherency protocols.

**Burst-Mode Transfers.** Memory operations that “hit” the 486 cache do not produce any bus traffic. Those that “miss” are transformed into external memory bus cycles. The bus interface tries to fill an entire cache line with a single four-word “burst-mode” transfer. With sufficiently fast memory, i.e., assuming zero-wait-state transfers, all four words can be transferred in five clock cycles total.

The order in which data words are retrieved depends on the original target address. For sequential instruction prefetches, all four words are retrieved in ascending order. Otherwise, the order in which burst transfers are performed is designed to make efficient use of interleaved (two-bank) memory systems.

This order is somewhat nonintuitive. As shown in Table 6-3, the first cycle always transfers the word containing the target data value, which is immediately passed directly to the unit initiating the request. Instruction execution can then continue with the shortest possible delay. The second cycle of a burst transfer reads the other half of the 64-bit-aligned memory word containing the target value. The third and fourth cycles retrieve the remaining values in the corresponding order.

Target Address	First Transfer	Second Transfer	Third Transfer	Fourth Transfer
xxxxxx0H	xxxxxx0H	xxxxxx4H	xxxxxx8H	xxxxxxCH
xxxxxx4H	xxxxxx4H	xxxxxx0H	xxxxxxCH	xxxxxx8H
xxxxxx8H	xxxxxx8H	xxxxxxCH	xxxxxx0H	xxxxxx4H
xxxxxxCH	xxxxxxCH	xxxxxx8H	xxxxxx4H	xxxxxx0H

Table 6-3. Intel 486 burst-mode-transfer address sequence.

The four words fetched during each burst transfer are held temporarily in a 16-byte holding register. If all four words are cacheable—the most common case—the entire cache will be updated at once. If not, data in the holding register is cleared and the cache is left unchanged.

The 486 also works with main memory systems that do not support burst operation, in which case the bus controller provides a separate address/data bus cycle for each word needed. If the memory region addressed is noncacheable, i.e., if it represents a memory-mapped I/O device or is part of a shared data structure, only the data word requested is retrieved. Ensuing cycles are aborted, and the state of the cache is left unchanged.

**Write Buffers.** Computers built with the 386 and other non-cached CPUs use the address and data buses primarily to read data into the CPU, so most of the bus cycles perform instruction fetches, and the remaining transfers mostly perform data reads. In i486DX-based systems, traffic on the external bus is reversed. Most program fetches and data reads are satisfied by the cache, so they do not involve the bus. Data writes, on the other hand, pass through to the external bus, so the majority of all bus traffic in i486DX-based systems is outbound. In systems with slow main memory, write operations could become a bottleneck.

The 486 uses internal write buffers to “decouple” the CPU from main memory so slow main memory write cycles won’t impede execution. If the external bus is available, write operations initiate an immediate data transfer. If the bus is busy, write operations save the destination address and data in an internal “write buffer” instead, and the CPU may continue executing ensuing instructions while the write operation is pending. When the bus later becomes available, internally buffered data is written to main memory. The bus interface contains four such buffers. If all four are in use, the write operation stalls until a write cycle completes and a buffer becomes available.

**Cache Coherency.** In many system configurations, all locations in main memory may be cached as needed. However, caching certain types of data can create hazards:

- System designs with memory-mapped I/O ports should not cache port values. Input port values can change spontaneously. The CPU should reread memory-mapped input ports each time the designated location is referenced.
- Multiprocessing systems may use main memory for communications buffers. Locations within this region should not be cached, since they may change at the whim of an attached processor.
- Even simple desktop PCs often have direct memory access (DMA) controllers on their hard disks and network interface boards that bypass the main CPU when they load programs or data into main memory. If the overwritten locations had previously been read by the main CPU, data in the cache would be invalid.

The potential mismatch between external and cached data is called cache inconsistency. The 486 has both hardware and software solutions to avoid this hazard.

The first hardware solution uses a cache-enable input pin. External address decoders can be designed to detect references to specific, noncacheable memory regions. If this pin is negated during the transfer sequence, the value fetched will not be cached. Further references to the same location will generate cache misses and force additional external memory fetches. This technique is best suited for handling memory-mapped I/O situations. (See device descriptions below for details.)

The second solution involves a technique known as “bus snooping.” The higher-order 486 address bus pins are bidirectional. When an auxiliary processor or DMA controller modifies main system memory, external logic drives the affected address onto the address pins. Logic within the cache compares the address of the location being modified against the internal tags. If a match is detected, the affected cache line is marked invalid. Later references to the same address will detect a cache miss and generate an external fetch. Cache tags are single ported, so snoop cycles that begin at the same time as an internal cache operation cause the internal instruction pipeline to stall for one clock cycle.

The third hardware solution is the most drastic. External logic can assert an input pin that immediately invalidates all internal cache tags. This is the most effective way to invalidate a large block of memory at once; for example, if an entire memory bank must be disabled due to hardware failure, or for bus-master operations that access main memory but cannot be snooped.

**On-Chip Self-Test.** To verify system integrity at run-time, the i486DX includes built-in self-test circuitry. Following system reset, an automatic self-test routine can optionally be invoked. The routine takes about  $2^{20}$  clock cycles to complete and confirms proper operation of most of the ALU, control microcode, cache, and virtual memory TLB cache. Fault coverage for the self-test is approximately 80%. Software-accessible test registers also allow the cache to be exercised and verified under program control.

### **“SL-Enhanced” Processors**

When it was introduced, and for several years thereafter, the 486 product line was not especially sensitive to power conservation. The core logic contained dynamic nodes that did not allow the CPU clock to be stopped or to run any slower than 8MHz, wasting power.

In late 1993, as battery-operated laptops and “green” PCs were coming increasingly into vogue, Intel announced that all of its future microprocessors would include power-management features such as a static core design, support for stopped-clock operation, system-management mode (SMM), and other features reminiscent of the i386SL chip discussed in **Chapter 5: Intel 386 Microprocessors**. Moreover, existing 486 products would also be modified to support these power-saving functions.

The newer versions of the 486 family were said to be “SL-enhanced.” After a short transition period all 486 production shifted to the enhanced design. Chips that support the new features retain the same part numbers as their predecessors, but have the characters “&E” stamped on the package. Note that while the original i386SL device also included a formidable amount of on-chip system logic, high-current I/O drivers, and the like, the SL-enhanced 486 chips include no such logic.

In addition to the generic 486 instructions listed in Table 6-2, SL-enhanced members of the Intel 486 family support the two special instructions shown in Table 6-4.

Instruction	Mode	Operation	Opcode
CPUID	User/ System	Head processor identification data	0FA2H
RSM	System	Resume normal execution following an SMI service routine	0FAAH

Table 6-4. Intel 486 "SL-enhanced" instructions.

The CPUID instruction gives software a mechanism for determining certain characteristics of the CPU on which it is running.

The RSM instruction terminates a system-management interrupt service routine and reloads prior CPU status.

SL-enhanced devices supported a number of new clocking modes and instructions. Executing a conventional HALT instruction places an SL-enhanced CPU into "Auto Halt" mode, greatly reducing power consumption. Any interrupt or reset will return the processor to normal operation. Asserting a special input signal can also put the chip into a new "Stop Grant" mode, with the same low power rating as the Auto Halt mode, until the signal is deasserted or the chip is reset. In either of these two modes, the processor will automatically power up for one cycle, as necessary, to service cache snoop requests.

Once in Stop Grant mode, the external clock input can be switched to the desired frequency, but the CPU will be unavailable for about one millisecond while its oscillator circuitry stabilizes. After that period, the CPU re-enters Stop Grant mode and can be returned to normal operation at the new clock speed. In effect, one must hold down the clutch long enough to cleanly shift gears.

Or, once in Stop Clock mode, the clock input can be stopped completely, reducing power requirements to about 1 mW. In "Stop Clock" mode, however, the processor cannot respond to snoop requests or interrupts.

For systems that wish to change clock speed "on the fly," certain members of the 486 family are available in a slightly modified version that eliminates the on-chip oscillator stabilizer circuit and accepts its clock input directly from two input pins. These parts can change clock speeds at any time, without using Stop Grant mode. Such chips use the same part numbers as standard 486 CPUs but must be identified by a special ordering code.

## 6.1 The Intel i486DX Microprocessor

The i486DX is the workhorse of Intel's original 486 product line. It combines a much more efficient implementation of the 386 integer core with a complete floating-point unit and 8 Kilobytes of unified instruction and data cache. Table 6-5 summarizes the general features and specifications of the i486DX microprocessor. A block diagram of the part appears in Figure 6-8.

Product Name	Intel i486DX
Introduction Date	April 1989
Prognosis	Approaching dotage
Device Integration Level	Pipelined 32-bit IEU and PMMU 8K-byte unified instruction/data cache Microcoded 80-bit floating-point unit
CPU Architecture Level	Standard 386 integer instruction set plus standard 387 floating-point instruction set plus six new instructions for cache control and multiprocessor support
Core Technology	De facto standard Intel 486 core
Pinout	De facto standard 486DX pinout
Data Bus Width	32 bits with parity (D31..D0 plus DP3..DP0)
Physical Addressability	4 GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Four-word (16-byte) burst-mode transfers Dynamic resizing for 8- or 16-bit transfers
Cache Support	8K bytes unified I- and D-cache Four-way set associative write-through operation only
Floating-Point Support	On-chip 80-bit microcoded FPU
Operating Voltage	4.75 V to 5.25 V (5-V version); 3.0 V to 3.6V (3.3-V version)
Frequency Options	20-, 25-, or 33-MHz core operation
Clocking Regime	Core operating frequency = 1 × CLK input
Active Power Dissipation	3.15 W @ 5.0 V and 33 MHz (worst case) 1.37 W @ 3.3 V and 33 MHz (worst case)
Power-Control Features	Standard Intel "SL-Enhanced" feature set
Process Technology	Originally 1.0 $\mu$ two-layer-metal CMOS; Redesigned for 0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	414 × 619 mils (165 mm <sup>2</sup> ) (1.0 $\mu$ technology) 273 × 468 mils (81 mm <sup>2</sup> ) (0.8 $\mu$ technology)
Package Options	168-pin PGA or 196-lead PQFP (5.0 V parts) 208-lead SQFP (3.3 V parts)

Table 6-5. Intel i486DX feature summary.

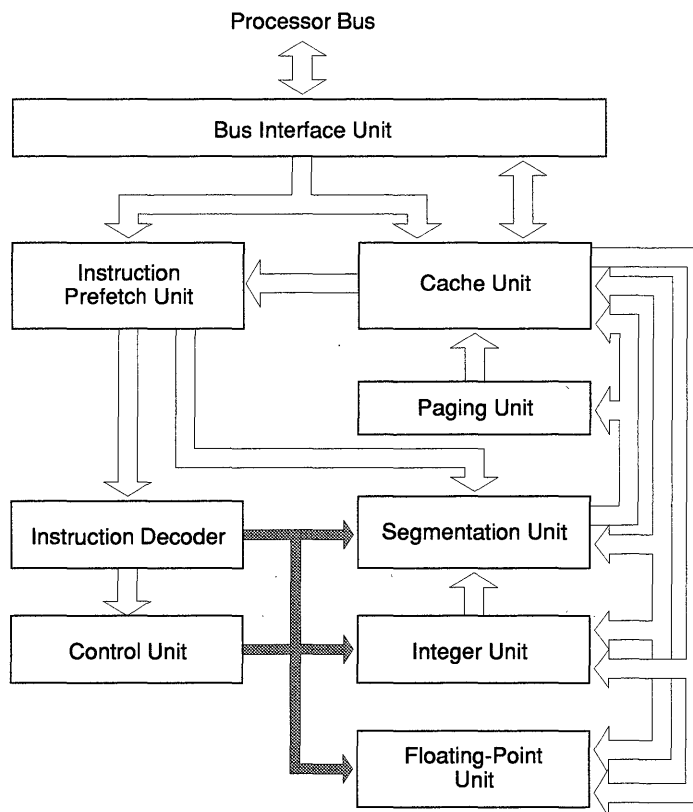


Figure 6-8. Intel i486DX block diagram.

### Floating-Point Unit

The design of the floating-point unit contained in the i486DX was inherited in large part from the 80387 FPU. Its programming model and instruction set are identical, and most arithmetic operations take essentially the same number of clock cycles to complete as they do on an i387DX. The 486 FPU does enjoy a moderate performance advantage over 387-based systems, though, due to reduced communications overhead in passing commands and data between the integer core and the FPU logic.

### Processor Clock

Microprocessors of the 386-family require an external clock input at a frequency two times higher than the internal clock frequency. The i486DX device implements a 1× system clock, so a 33-MHz processor uses a 33-MHz oscillator. Eliminating the need for a higher-frequency signal simplifies system design and helps meet FCC radio-frequency emission standards.



## System Interface

The i486DX supports a variety of system interface architectures. A total of 99 pins carry address, data, and control information. An additional 52 pins are dedicated to Vcc and Vss power distribution. Figure 6-9 shows a functional grouping of these pins.

Like the 386, total physical memory in a 486-based system can be up to four gigabytes. Unlike the 386, memory arrays and I/O buses can be 8, 16, or 32 bits wide, in any combination. Transfers can occur individually or in four-transfer bursts. On-chip logic can generate or optionally check the parity of each byte transferred on the data bus. Multiprocessing systems allow a hierarchy of bus levels, with contention resolution, second-level caches, and cache consistency protocols supported in hardware.

The names and functions of signals used by the i486DX are summarized in Tables 6-6 through 6-9. Its basic memory bus interface is patterned after that of the 386 device. Many of the pins described in this section perform essentially the same functions as their counterparts on the 386 microprocessor, though signal timing and electrical characteristics may differ.

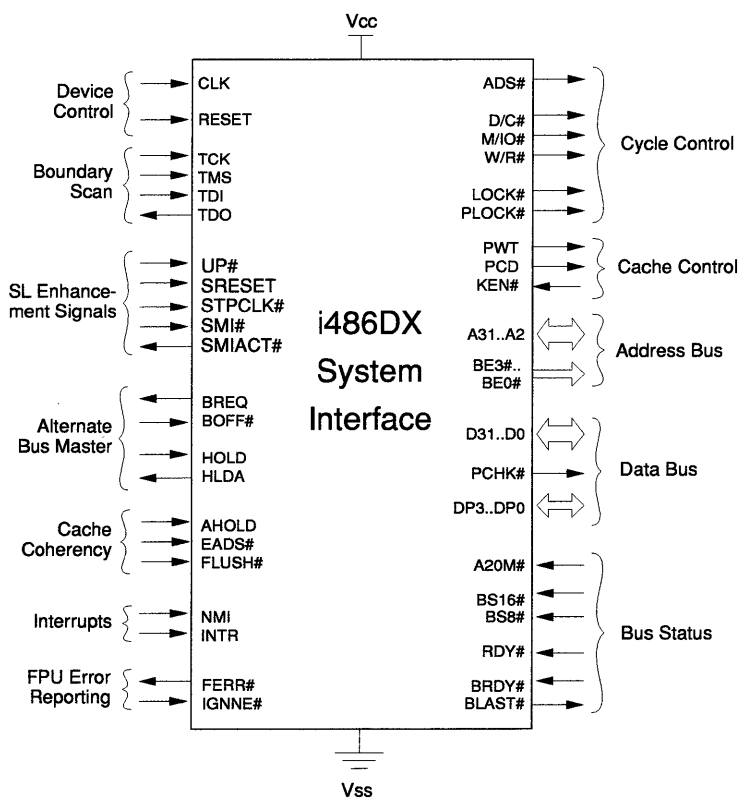


Figure 6-9. Intel i486DX system interface.

Symbol	Direction	Signal Name/Function
A31..A4	I/O	Address output bus/cache-line snoop input bus
A3..A2	Out	Address output bus LSBs
BE3#..BE0#	Out	Data bus byte enable controls
D31..D0	I/O	Data I/O bus (D31 = MSB)
DP3..DP0	I/O	Data bus byte parity bits (even parity)
PCHK#	Out	Data bus parity error detected
A20M#	In	Address bit 20 mask

Table 6-6. Intel i486DX address and data bus signals.

Address pins A31..A2, byte-enable pins BE3#..BE0, and data pins D31..D0 generally operate as on the i386DX, except that A31..A4 can also serve as address inputs during cache snoop cycles.

Bidirectional pins DP3..DP0 produce and verify parity for each byte of the data bus. On-chip parity logic eliminates the cost and real estate consumed by off-chip parity logic. More important, it increases the effective memory access time in the critical timing path of most external memory systems.

When parity errors are detected, the CPU asserts the PCHK# output pin on the next clock cycle, but CPU operation is otherwise unaffected. External logic can decide whether parity errors should signal a normal interrupt or a nonmaskable interrupt, or whether they should be ignored, depending on the characteristics of the memory in which the error occurred.

The A20M# pin compensates for an anomaly in the way the 8086 and 80286 microprocessors handle memory address overflows. Since the 8086 has just a 20-bit physical address bus, address computations that overflow the one-megabyte address boundary are effectively aliased to the very bottom region of memory. MS-DOS and DOS-based software exploit this feature to access both the very top and the very bottom of the 1MB address space using the same segment base register.

In contrast, the 80286 and 386 architectures allow larger physical address spaces, so address calculations that overflow one megabyte do *not* access low-order addresses, causing such software to malfunction. IBM-compatible 80286- and 386-based systems were therefore designed to include external logic that forces address pin A20 low externally under software control in order to emulate 8086 behavior. Low-order addresses and addresses that lie just over the 1MB boundary are thus aliased to the same physical memory location.

This trick doesn't work with microprocessors that have on-chip cache, since the aliased values would be cached internally as different locations. Instead, input pin A20M# on the i486DX masks address-bit 20 internally. This simplifies system logic slightly, ensures that internal cache tags always match external memory addresses, and removes a critical propagation delay from the external address timing path.

Symbol	Direction	Signal Name/Function
ADS#	Out	Address strobe (initiates new bus cycle)
M/IO#	Out	Memory vs I/O bus cycle
D/C#	Out	Data vs code bus cycle
W/R#	Out	Write vs read bus cycle
LOCK#	Out	Locked (indivisible) bus cycle
PLOCK#	Out	Pseudo lock (multiple-transfer transaction)
BS16#	In	Bus size 16; 32-bit transfers require two cycles
BS8#	In	Bus size 8; 32-bit transfers require four cycles
RDY#	In	Ready (transfer data accepted/available)
BRDY#	In	Burst-mode transfer ready
BLAST#	In	Burst last (final cycle of burst-mode transfer)
BOFF#	In	Back off (abort all outstanding bus cycles)
HOLD	In	Bus hold request (external master request)
HLDA	Out	Bus hold acknowledge (bus available)
BREQ	Out	Bus request (internal bus cycle pending)

Table 6-7. Intel i486DX bus control and status signals.

The i486DX ADS#, M/IO#, D/C#, W/R#, LOCK#, BS16#, RDY#, HOLD, HLDA, and BREQ signals perform generally the same functions as their 386 counterparts. For further details see **Chapter 5: Intel 386 Microprocessors**.

The PLOCK# output signal is asserted by the processor any time a single data element (such as an 80-bit floating-point variable) requires more than one bus cycle to load or store. This is to ensure that no other bus master will be allowed to gain control of the bus in mid-transfer.

Inputs BS8# and BS16# control an enhanced dynamic bus-sizing facility. On any memory cycle, system logic can indicate if the addressed device is just one or two bytes wide, rather than four, by driving the corresponding input. The bus controller in the i486DX will then immediately issue up to three additional bus cycles, as needed, to retrieve the higher-order bytes.

This facility makes it possible for an i486DX to boot itself from a single byte-wide EPROM, and can simplify peripheral interfacing, for example. It can also simplify and eliminate the external state machine that would otherwise be required to perform byte-wide and double-byte transfers on the ISA, EISA, and Micro Channel buses.

On the first cycle of an instruction fetch or cache-line fill, bus-control logic will attempt to initiate a burst-mode transfer. If main memory supports burst-mode transfers from the memory region addressed, external circuitry asserts the BRDY# input pin, and an entire sequence of instruction or data words can be transferred on successive clock cycles.

External logic should assert BLAST# to inform the i486DX when a burst-mode transfer sequence is completed.

Asserting the BOFF# input pin causes the i486DX to abort and reinitiate any data-transfer cycles currently in progress. This gives i486DX-based systems a graceful way to escape from potential bus deadlock situations, as detailed below.

Symbol	Direction	Signal Name/Function
PCD	Out	Page cache disable bit for requested data
PWT	Out	Page write-through bit for requested data
KEN#	In	Cacheability enabled for requested data
AHOLD	In	Address hold (float address bus next cycle)
EADS#	In	External snoop address driven to bus
FLUSH#	In	Flush cache data

Table 6-8. Intel i486DX cache control and status signals.

The PCD and PWT output signals control the cacheability and write-through policy of external second-level caches under control of the memory descriptor tables.

The KEN# input pin determines the cacheability of external memory regions. If the address of a transfer corresponds to a cacheable region in main memory, external circuitry should assert the KEN# input pin when the data is returned. Otherwise, KEN# should be deasserted. Asserting KEN# gives memory-mapped I/O ports, shared memory regions, and other configuration-dependent resources a way to let the data they contain be retrieved without consuming internal cache space, and ensures the processor reads a shared variable or memory-mapped port it will retrieve the most current value.

The AHOLD and EADS# input signals are used to perform cache snoop cycles that invalidate internal cache lines if an external copy of the same data is modified by external logic. Asserting the FLUSH# input simultaneously invalidates the data in all internal cache lines.

Symbol	Direction	Signal Name/Function
CLK	In	Processor clock input
RESET	In	Reset processor
INTR	In	Maskable interrupt request
NMI	In	Non-maskable interrupt
FERR#	Out	Floating-point error detected
IGNNE#	In	Ignore numeric (FPU) errors

Table 6-9. Intel i486DX device control and status signals.

CLK is the system clock input and provides the fundamental timing and internal operating frequency for the i486DX. The internal i486DX CPU core runs at the same frequency as the CLK input. All external timing parameters are specified with respect to the rising edge of CLK. Its voltage levels are compatible with standard TTL signals.

RESET, NMI, and INTR perform the same functions as the identically named pins on the i386DX. Refer back to **Chapter 5: Intel 386 Microprocessors** for details.

FERR# is asserted when the on-chip FPU encounters a floating-point exception. System designers may choose to process these exceptions entirely within the 486 CPU, or the FERR# output may be connected to an external interrupt controller in order to preserve full PC hardware and software compatibility.

The IGNNE# input may be asserted externally to cause numeric errors to be ignored.

### Deadlock Backoff

In systems with multiple bus masters, it's possible for a situation to arise in which two separate bus masters are each in control of some system resource, and each is attempting to gain control of some other resource. If the resource on which each is waiting is already controlled by the other bus master, and neither bus master can release the resource it controls until it completes the transaction it has begun, then neither bus master can complete the transaction it has begun until the other master releases the resources *it* controls.

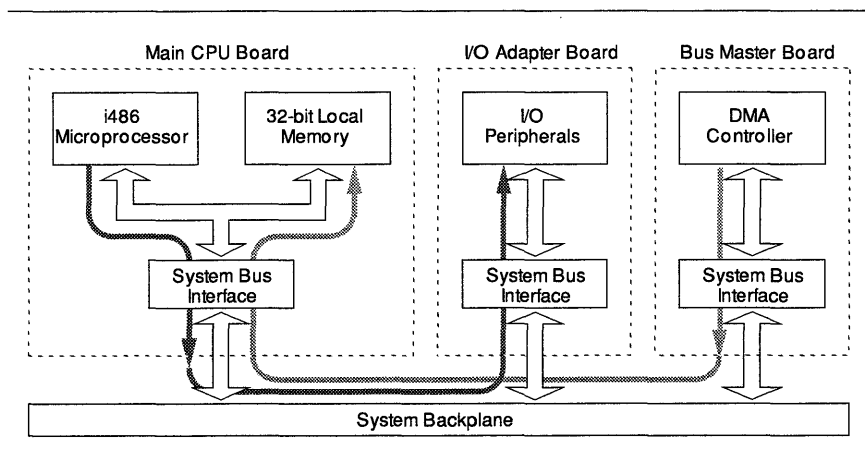


Figure 6-10. Intel i486DX system deadlock avoidance.

Figure 6-10 shows one such situation. The i486DX is attempting to gain control of the system bus interface logic from the local-bus side in order to read an I/O port, while a DMA controller is attempting gain control of the same interface from the system-bus side in order to read data from memory on the CPU board.

This situation can lead to a system deadlock unless one of the bus masters contains the logic to allow it to “back off” from its request. This logic is built into the i486DX. If external circuitry determines that a deadlock situation has occurred (for example, by ANDing together control signals that indicate both sides of the system-bus interface are busy), the BOFF# input pin is asserted. The i486DX will then immediately float its address, data and status pins until BOFF# is deasserted, letting the other bus master complete its transfer and release any resources it was holding.

### SL-Enhancements

SL-enhanced versions of the i486DX add the control and status signals described in Table 6-10 to those described earlier in this section.

TMS, TCK, TDI, and TDO provide the interface to JTAG-compliant on-chip boundary-scan test logic. TMS enables the JTAG test mode, TCK is the test-mode clock input, and TDI and TDO are the serial input and output data pins, respectively.

### Vital Statistics

The original i486DX had a die size of  $414 \times 619$  mils ( $165 \text{ mm}^2$ ), using a 1.0-micron two-layer-metal CMOS process. Devices currently in production measure  $273 \times 468$  mils ( $81 \text{ mm}^2$ ), with a 0.8-micron, three-layer-metal die. The 1.0-

Symbol	Direction	Signal Name/Function	PGA Pin	Prior Signal
TCK	In	JTAG boundary scan clock	A3	N.C.
TMS	In	JTAG boundary scan mode select	B14	N.C.
TDI	In	JTAG boundary scan test data in	A14	N.C.
TDO	Out	JTAG boundary scan test data out	B16	N.C.
UP#	In	Upgrade processor present	C11	N.C.
SRESET	In	System management reset	C10	N.C.
STPCLK#	In	Stop clock	G15	N.C.
SMI	In	System management interrupt	B10	N.C.
SMIACT	Out	System management mode active	C12	N.C.

Table 6-10. Intel SL-enhanced 486 device control and status signals.

micron device requires a supply voltage between 4.75 V and 5.25 V, but the 0.8-micron device can operate with a supply voltage of either 3.0 V to 3.6 V or 4.75 V to 5.25 V.

The 5.0-V version is available in either a 168-pin PGA or 196-lead PQFP package and runs at speeds up to 33 MHz. Pinout diagrams for each package appear in Figures 6-11 and 6-12. The device dissipates 3.15 W (worst case) at 5.0 V and 33 MHz.

The 3.3 V version uses a 208-lead SQFP package and also runs at up to 33 MHz. Its pinout diagrams appear in Figure 6-13. The device dissipates 1.37 W (worst case) at 3.3 V and 33 MHz.

The i486DX was originally offered in 20- and 25-MHz flavors and is currently available in 25-MHz and 33-MHz variations. (A faster, redesigned 50-MHz version is described in the next section.) At 5.0 V and 33 MHz, the i486DX dissipates approximately 4.5 W (worst case).

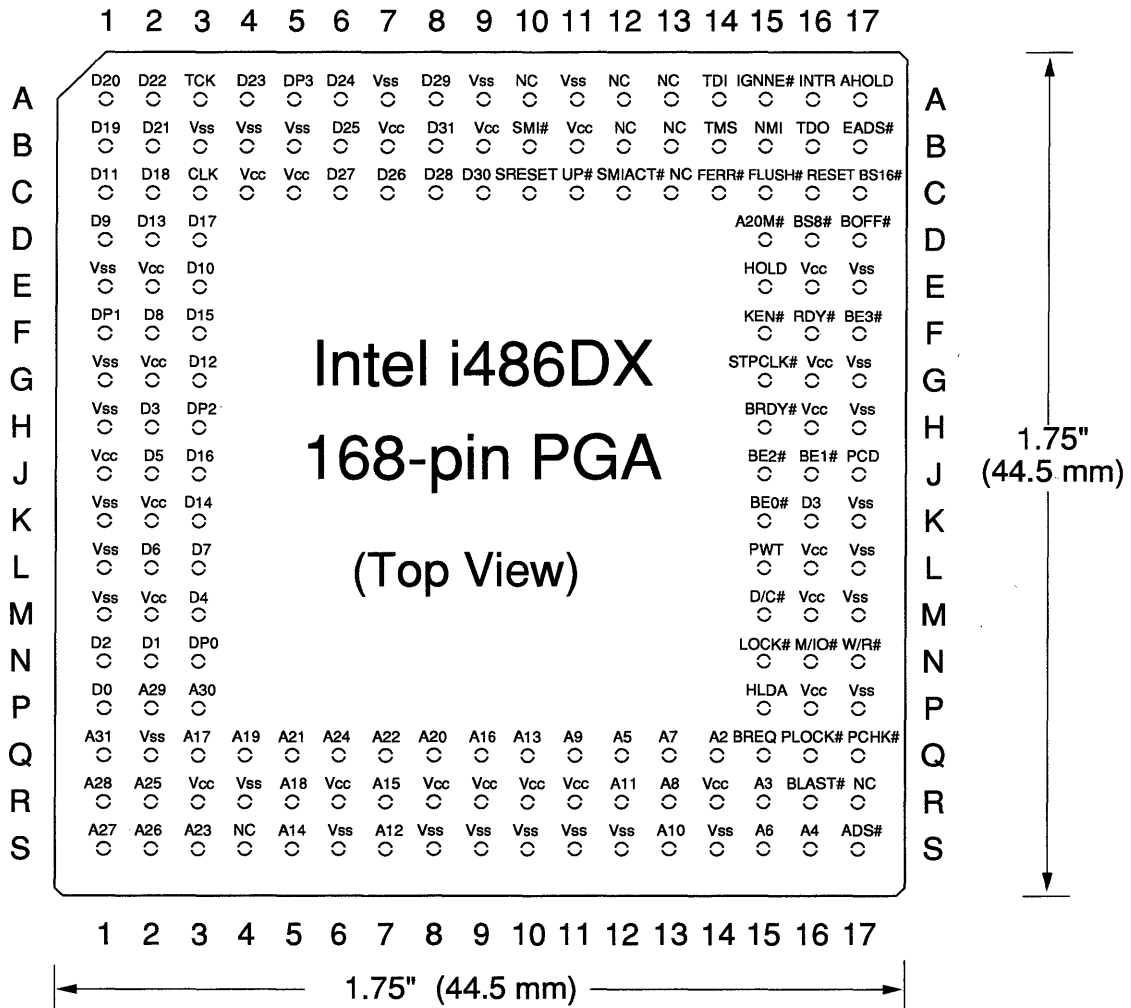


Figure 6-11. Intel i486DX PGA pinout.



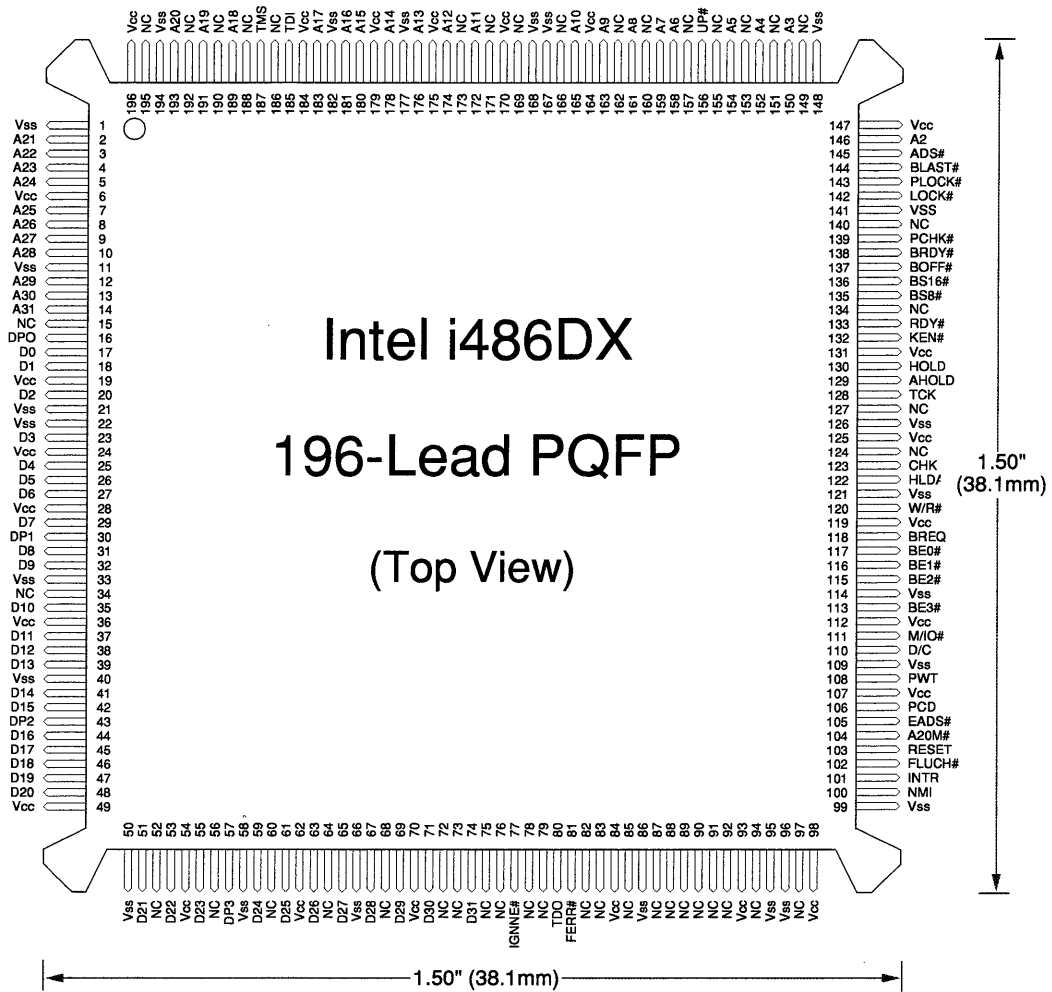


Figure 6-12. Intel i486DX PQFP pinout.

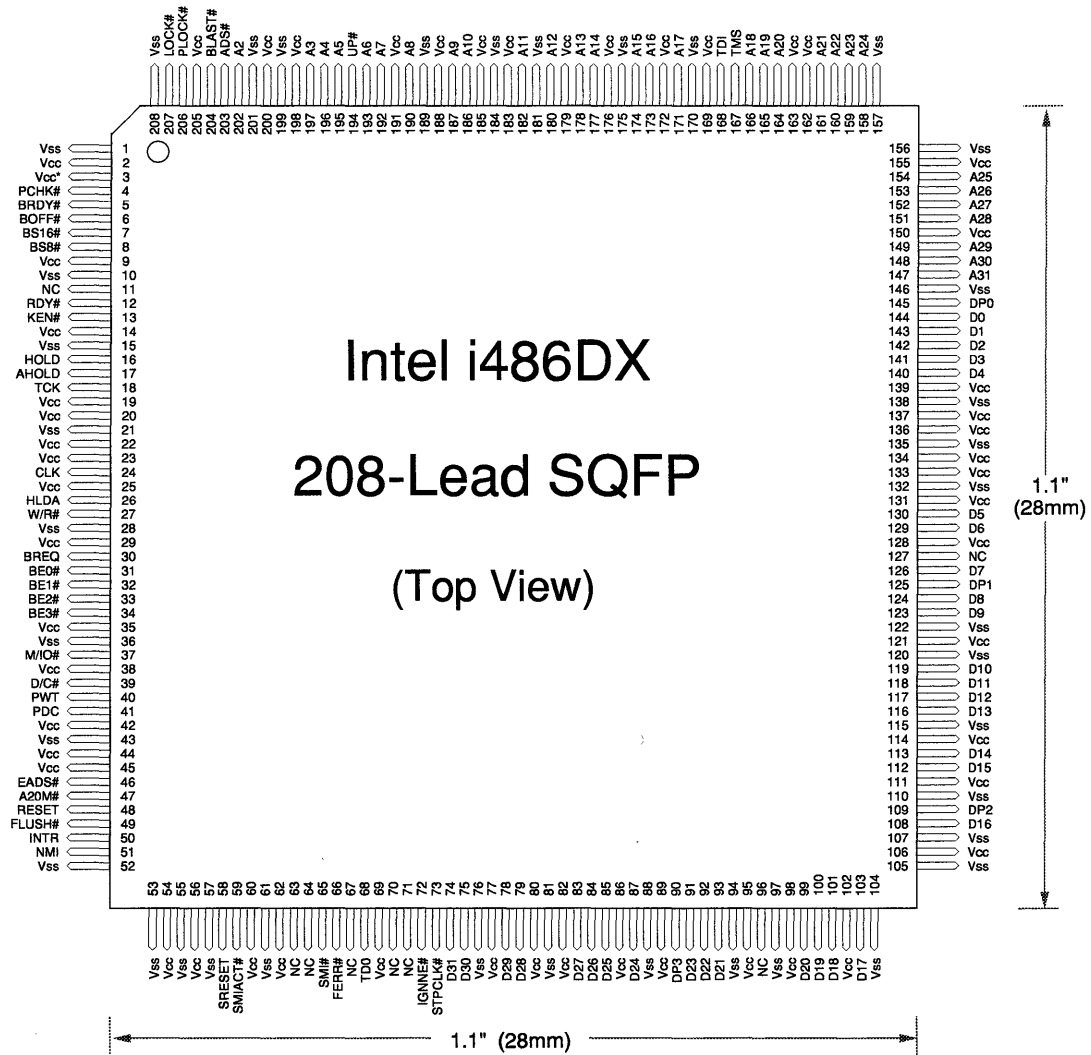


Figure 6-13. Intel i486DX SQFP pinout.

## 6.2 The Intel i486DX-50 Microprocessor

The i486DX-50 microprocessor is a faster incarnation of the original i486DX, based on a new implementation of the core that was designed to take advantage of faster process technology. It is functionally compatible with the original i486DX device, with the inclusion of the JTAG system test circuitry. Table 6-11 summarizes the general features and specifications of the i486DX-50 microprocessor.

Product Name	Intel i486DX-50
Introduction Date	June 1991
Prognosis	On decline (replaced by SL-enhanced i486DX)
Device Integration Level	Same as i486DX
CPU Architecture Level	Same as i486DX
Core Technology	Redesigned Intel 486 core
Pinout	Standard i486DX pinout augmented with JTAG boundary-scan interface
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	4.75 V to 5.25 V
Frequency Options	50-MHz core operation
Clocking Regime	Core freq = CLK input frequency
Active Power Dissipation	4.75 W @ 5.0 V and 50 MHz (worst case)
Power-Control Features	None
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	273 $\times$ 468 mils (84 mm <sup>2</sup> )
Package Options	168-pin ceramic PGA
Other Features	Phase-locked-loop (PLL) clock synthesizer H/W programmable output drive levels JTAG boundary-scan logic

Table 6-11. Intel i486DX-50 feature summary.

**Background** The 50-MHz redesign resulted from a two-year effort by a circuit design team at Intel's Portland Technology Development group. The team's charter was to take the existing 486 logic design, implement it in Intel's new 0.8-micron (drawn) three-level-metal process, and tune it for the highest possible speed.

The combination of smaller geometry, three-layer metal, and redesign of critical circuit elements reduced the die-size by more than 50% and potentially doubled the maximum clock rate.

According to Intel, switching from two-layer to three-layer metal without changing any other features would have reduced the die area by 25%. This, combined with the gains from a special router Intel developed for use with this process, increased speed by 25%. Another 15% speed gain resulted from analyzing 7,000 potentially speed-limiting paths and redesigning those with the longest delays. The remainder of the speed and die-size improvements come from the inherent advantages of the finer-geometry process.

The new design does not include any architectural changes. The only externally visible change is the inclusion of three programmable drive levels for the buses. This allows the system designer to match the driver impedance to the system configuration. Outputs provide the full CMOS rail-to-rail voltage swing, and the bus inputs allow TTL or CMOS thresholds to be selected.

### Clock Synthesis Circuit

The i486DX-50 clock generator was redesigned to improve performance. Instead of deriving the internal clock phase signals directly from the external input as was done with previous family members, the i486DX-50 includes an internal phase-locked-loop (PLL) clock generator, as shown in Figure 6-14.

The circuit shown in Figure 6-14 causes the Voltage Controlled Oscillator to generate whatever high-frequency clock signal is required to satisfy the requirements of the other elements of the circuit. In this case, the frequency produced will be such that, when divided by two to produce the internal phase signals, it will yield the same frequency supplied to the external CLK pin. In effect, CLK acts as a simple input signal that regulates the internal clock frequency and phase.

The phase signals used to coordinate system operation and timing are thus derived entirely within the chip, minimizing the

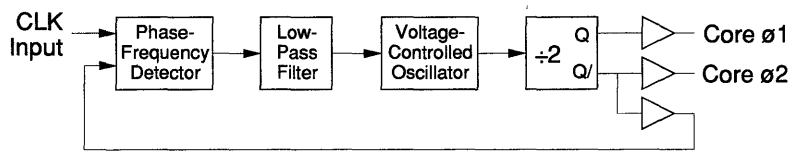


Figure 6-14. Intel i486DX-50 PLL clock synthesis circuit.

propagation delays and timing skews that would otherwise result from an off-chip clock. As a result, the i486DX-50 specifications reduce input signal setup times from 3 ns to 1.5 ns and hold time from 2.5 ns to 1.0 ns.

### System Interface

The i486DX-50 implements the same system interface as the original (non-SL-enhanced) i486DX devices, with the addition of the signals that make up the JTAG boundary-scan logic interface. The names and functions of the JTAG interface signals are summarized in Table 6-12.

Symbol	Direction	Signal Name/Function	PGA Pin #	i486DX Signal
TCK	In	JTAG test clock	A3	N.C.
TMS	In	JTAG test mode select	B14	N.C.
TDI	In	JTAG test data in	A14	N.C.
TDO	Out	JTAG test data out	B16	N.C.

Table 6-12. Intel i486DX-50 JTAG boundary-scan signals.

### Vital Statistics

The i486DX-50 has a die size of  $273 \times 468$  mils ( $81 \text{ mm}^2$ ) using a 0.8-micron three-layer-metal CMOS process and requires a supply voltage between 4.75 V and 5.25 V. It is available only in a 168-pin PGA package and runs (naturally) at 50 MHz. The device dissipates 4.75 W (worst case) at 5.0 V and 50 MHz. Its pinout matches that of the original i486DX devices illustrated in the previous section, with the additional JTAG interface pins listed above.

## 6.3 The Intel i486DX2 Microprocessor

The Intel i486DX2 microprocessor is a version of the 486 family that uses on-chip “clock-doubling” circuitry to improve the performance of the processor by 60% to 100% without increasing the cost or complexity of external system logic. Table 6-13 summarizes the general features and specifications of the i486DX2 microprocessor.

Product Name	Intel i486DX2
Introduction Date	February 1992
Prognosis	Healthy
Device Integration Level	Same as i486DX with on-chip PLL clock-frequency doubler
CPU Architecture Level	Same as i486DX
Core Technology	Clock-doubled 486 core
Pinout	Augmented, rearranged i486DX pinout
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX except that bus operates at one-half core frequency
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	4.75 V to 5.25 V (5-V version) 3.0 V to 3.6 V (3.3-V version)
Frequency Options	25 or 33 MHz (50- or 66-MHz core freq) @ 5.0 V 20 or 25 MHz (40- or 50-MHz core freq) @ 3.3 V
Clocking Regime	Core operating frequency = 2 × CLK input Bus operating frequency = CLK input freq
Active Power Dissipation	6.0 W @ 5.0 V and 66 MHz core (worst case) 1.85 W @ 3.3 V and 50 MHz core (worst case)
Power-Control Features	Standard Intel “SL-Enhanced” feature set plus “Auto Idle” clock-reduction mode
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	6.9 mm × 11.9 mm (81 mm <sup>2</sup> )
Package Options	168-pin PGA
Other Features	Higher-frequency devices are available with built-in heat sinks

Table 6-13. Intel i486DX2 feature summary.

The i486DX2 operates from a 25- or 33-MHz external clock and appears nearly identical to a standard chip from a hardware perspective, but it operates internally at 50 or 66 MHz as long as its memory needs are satisfied by the on-chip cache.

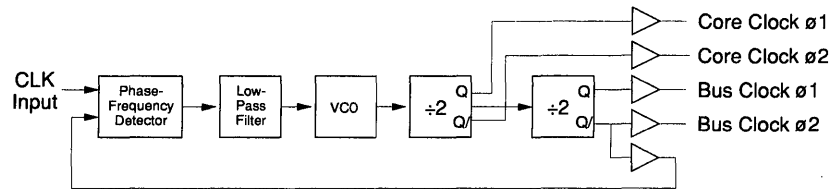


Figure 6-15. Intel i486DX2 PLL clock-doubler circuitry.

The i486DX2 provides system makers with a very easy way to introduce a new model simply by replacing the i486DX in existing 25-MHz systems with an i486DX2. These “pseudo-50-MHz” systems have displaced the i486DX-33 and i486DX-50 as the most popular system for power users. True 50-MHz systems have been too expensive to become mainstream products and have been popular primarily as servers.

Systems based on an i486DX2-50 are significantly less expensive than true 50-MHz systems because slower cache memories and other system components can be used. The design task is also much easier; while designing a true 50-MHz system is difficult, an i486DX2-50 allows a no-brainer upgrade to any 25-MHz design. This enables every clone vendor to offer this configuration, whereas many of them avoided building 50-MHz systems.

### Clock-Doubler Circuitry

The i486DX2 was the first x86 microprocessor to include an on-chip clock-frequency doubler to enhance core performance. The clock-synthesizer circuit shown in Figure 6-15 is derived from that of the i486DX-50, with the addition of an extra divider stage and separate phase signals for the CPU core and the system bus interface.

In effect, this circuit causes the voltage-controlled oscillator to adjust itself as needed in order to produce an internal clock signal of the proper frequency and timing such that, when the internal clock is twice divided by two, the resulting ø2 signal used by the system-interface signals matches the frequency and phase of the CLK input pin.

Whenever the i486DX2 core is waiting for a memory or I/O cycle to complete, the core clock frequency is cut in half. This feature, called “Auto Idle” mode, automatically reduces power consumption by up to 10%, according to Intel estimates. Transitions into and out of Auto Idle mode are fully software-transparent and do not have any effect on performance or system design.

(Aside from the i486DX2, Auto Idle mode is supported only on the IntelDX4 processor discussed later in this chapter. Coincidentally, these are the only two SL-enhanced devices that include clock-multiplier circuitry. Presumably the Auto Idle feature works by switching the core logic clocks to the outputs of the second of the two divide-by-two counters in the PLL feedback loop.)

### **Relative Performance**

Assuming cache hits for all instruction and data accesses, software performance would, of course, be exactly two times that of a standard i486DX at the same clock rate. In practice, the performance gain seen by the user depends strongly on the application, and varies from as little as 10% for I/O-bound or “cache-buster” programs to nearly 100% for programs that spend most of their time performing repetitive operations on small data sets.

According to Intel’s benchmark data, the performance of the i486DX2-50 comes surprisingly close to a “straight” i486DX-50, that is, a device with the same core frequency and a full-speed bus. Dividing the bus speed by two reduces throughput on the Norton SI and Dhrystone benchmarks—benchmarks which generally fit in the on-chip cache—by less than 2% in a system with an external 128-Kilobyte write-through cache.

On the SPEC integer benchmarks, the i486DX2-50 is just 11% slower than the i486DX-50, and on the SPEC floating-point benchmarks, it is 13% slower when run on systems with an external 256-Kilobyte copy-back cache.

### **System Upgrade Good News/Bad News**

The i486DX2 can theoretically upgrade any i486DX system simply by replacing the original CPU with an i486DX2, but several potential problems may arise:

- The chip’s power consumption is substantially higher, so the cooling in some systems may not be adequate. The i486DX-25 dissipates 2.75 W typical and 3.5 W maximum, while the i486DX2-50 draws 3.875 W typical and 4.75 W maximum.
- While the interface timing specifications are identical, the actual timing is slightly different. This can cause problems in some marginal system designs.
- Some BIOS programs include speed-dependent timing loops.



Intel says its initial testing found about one system in four that encountered problems. Making a list of systems that can be safely upgraded isn't as easy as it might seem, since it sometimes depends on which revision of the system board and BIOS is present. Computer dealers may offer unauthorized upgrades, and sophisticated end-users may be willing to try the upgrade themselves, but the potential for problems is significant.

### **System Interface**

Since the i486DX2 is socket-compatible to the i486DX, all system-interface signals have (by definition) the same names as, perform the same functions as, use the same pin locations as, and match the timing of the corresponding signals of the original product.

Because the i486DX2 has a much higher bus utilization than the standard i486DX, it is more sensitive to the performance of the external cache and memory system. Cacheless i486DX-25 PCs with a good memory-system design perform nearly as well as i486DX-25 PCs that do include a cache. If a system is based on a higher-end processor such as the i486DX2-66, adding a fast cache to its main memory system can improve its performance dramatically.

Intel's tests show that adding a 256K-byte write-through cache to an i486DX2-66 system increases performance by an average of 10% for DOS applications and 16% for Windows applications. Even with the cache, reducing the DRAM write latency by one clock cycle boosted Dhrystone performance by 24% and SPECint89 performance by 13%, illustrating the importance of an optimized memory system.

### **Vital Statistics**

The i486DX2 has a die size of  $273 \times 468$  mils ( $81 \text{ mm}^2$ ) using a 0.8-micron three-layer-metal CMOS process, the same as the i486DX-50. Whereas the i486DX2 includes a small amount of additional logic to handle the clock-doubling circuitry and half-speed bus interface, this did not affect the part's die size. The device allows a supply voltage of either 3.0 V to 3.6 V or 4.75 V to 5.25 V.

The 3.3-V version is supplied in a 208-lead SQFP package with core frequencies of either 40 MHz or 50 MHz. The part dissipates 1.85 W (worst case) at 3.3 V and 50 MHz. The 5.0-V version is available in a 168-pin PGA and is offered in versions with core frequencies of either 50 MHz or 66 MHz. The device dissipates 6 W (worst case) at 5.0 V and 66 MHz. The i486DX2 pinouts are the same as those of the i486DX devices discussed earlier in this section.

## 6.4 The Write-Back-Enhanced IntelDX2 Microprocessor

The Write-Back-Enhanced IntelDX2 microprocessor is a variation on the i486DX2, with the on-chip cache redesigned to support copy-back as well as conventional (write-through) operation. Table 6-13 summarizes the general features of the WB-enhanced IntelDX2 microprocessor.

Product Name	Write-Back-Enhanced IntelDX2
Introduction Date	October 1994
Prognosis	Promising
Device Integration Level	Same as i486DX
CPU Architecture Level	Same as i486DX
Core Technology	Clock-doubled 486 core
Pinout	Augmented i486DX pinout
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX with copy-back support added
Floating-Point Support	Same as i486DX
Operating Voltage	4.75 V to 5.25 V
Frequency Options	25 or 33 MHz (50- or 66-MHz core freq)
Clocking Regime	Core operating frequency = $2 \times$ CLK input Bus operating frequency = CLK input freq
Active Power Dissipation	6.0 W @ 5.0 V and 66 MHz core (worst case)
Power-Control Features	Standard Intel "SL-Enhanced" feature set
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	6.9 mm $\times$ 11.9 mm (81 mm <sup>2</sup> )
Package Options	168-pin PGA

Table 6-14. Write-Back-enhanced IntelDX2 feature summary.

### Overview

Prior to the WB-enhanced IntelDX2, the caches contained in all members of the Intel 486 family operated in write-through mode only. Whenever the CPU altered or updated a memory location the CPU wrote the new value directly through to system memory. If the memory location to be changed was already present in the on-chip cache, both external memory and the internal cache value would be updated with the same value.

Write-through caches have a very curious effect on the nature of the bus traffic between the CPU and system memory. In a pro-

cessor with no internal caching, or with an internal cache disabled, the majority of all bus transactions flow from system memory into the CPU. Every instruction executed must first be fetched, and every memory-based operand must be loaded into the CPU before it can be used in a calculation. In comparison, new values are written to memory relatively infrequently.

When an on-chip cache is present the situation changes. Most instruction fetches and operand reads are then satisfied by the cache, eliminating the need for perhaps 90% of all memory read cycles. But if the cache operates only in write-through mode, all bus write cycles must still be performed. The majority of bus transactions that remain are therefore writes.

This does not make very effective use of either the processor bus or of system memory. As processor cores run faster and faster, the bus can quickly saturate with unnecessary write operations—unnecessary because most of the values written to memory will not be read before the same location is modified again. In tightly-coupled multiprocessing systems, shared buses can quickly saturate as well.

Write-back (or copy-back) caches avoid these bottlenecks by circumventing most unnecessary write operations. Only when a cache line must be used to buffer a different memory location will the values previously saved in that line be copied back to memory.

## **System Interface**

The system interface of the WB-enhanced IntelDX2 is a superset of the interface implemented by the earlier members of the Intel 486 family. Seven new signals have been defined, or have been redefined to support additional capabilities. These signals are listed in Table 6-6.

The WB/WT# signal allows external hardware to control the mode in which the internal cache logic operates.

The CACHE# output is active on read cycles to indicate that the memory location being accessed is internally cacheable. On write cycles CACHE# is active to indicate that a burst-mode write will be performed.

When a cache snoop cycle is executed, the HITM# signal is asserted to indicate whether modified data with the address being snooped is present in cache.

Symbol	Direction	Signal Name/Function	PGA pin	Prior function
WB/WT#	In	Write-back/write-through mode control	B13	N.C.
CACHE#	Out	Cacheability status on reads or burst-mode writes	B12	N.C.
HITM#	Out	Hit/Miss detected to a Modified cache line	A12	N.C.
INV	In	Cache line invalidation request	A10	N.C.
FLUSH#	In	Flush (write back) modified cache lines to system memory	C15	FLUSH#
SRESET	In	Soft reset	C10	SRESET
PLOCK#	Out	Pseudo-bus lock	Q16	PLOCK#

Table 6-15. WB-enhanced IntelDX2 revised interface signals.

When the INV input is asserted during a snoop cycle, the address specified will be invalidated, but any modified data within that line will first be written back to system memory.

When the FLUSH# input signal is asserted, any cache lines that contain modified data are written back to memory.

The SRESET input provides a mechanism by which the CPU can be reset quickly without losing or corrupting any modified data within the cache.

The PLOCK# signal is asserted by non-WB-enhanced 486 devices to prevent other bus masters from acquiring the bus. Write-back protocols eliminate this hazard, so the signal is never asserted while WB-enhanced operation is enabled.

**Performance** See **Chapter 20: Performance Issues** for an analysis of the performance effects of the write-back cache.

## 6.5 The IntelDX4 Microprocessor

The IntelDX4 microprocessor is a 100-MHz variation of the i486DX2 that provides twice the on-chip cache and uses “clock-tripling” circuitry to further enhance system performance at relatively modest system frequencies. Table 6-16 summarizes the general features and specifications of the IntelDX4 microprocessor.

Product Name	IntelDX4
Introduction Date	March 1994
Prognosis	Thriving
Device Integration Level	Same as i486DX with 16K-byte cache and programmable on-chip PLL clock-frequency tripler
CPU Architecture Level	Same as i486DX
Core Technology	Clock-tripled 486 core
Pinout	Augmented, rearranged i486DX pinout
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Four-word burst-mode transfers Dynamic resizing for 8-bit or 16-bit transfers Bus operates at one-third the core frequency
Cache Support	16K-byte I/D cache
Floating-Point Support	Same as i486DX
Operating Voltage	3.0 V to 3.6 V
Frequency Options	75-MHz or 100-MHz core frequency
Clocking Regime	Core freq = 2× or 3× CLK input Bus frequency = CLK frequency
Active Power Dissipation	4.3 W @ 3.3 V and 100 MHz (worst case)
Power-Control Features	Standard Intel “SL-Enhanced” feature set plus “Auto Idle” clock reduction mode
Process Technology	0.6 $\mu$ four-layer-metal BiCMOS
Transistor Count	1.6 million transistors
Die Size	339 × 351 mils (77 mm <sup>2</sup> )
Package Options	168-pin PGA

Table 6-16. IntelDX4 feature summary.

The first surprise of this product is its name. That’s right, there’s no “486” in the name, just “IntelDX4.” Intel claims it dropped the “486” designation because Cyrix and IBM, by selling parts that use a 486 part number but fall short of i486DX performance, have made the 486 appellation meaningless. More likely, Intel is motivated by the fact that the new name is trademarkable, whereas the digits “486” are not. (Never fear; AMD

soon began offering a slightly less capable product with the Am486DX4 name and others are likely to follow suit.)

The second surprise is that the “DX4” name seems to imply that the chip supports clock-quadrupling, whereas in fact it does not. The internal clock can be programmed to run at 2× or 3× the external clock rate, but not at 4×. (Intel initially planned also to allow a 2.5× configuration, but the part is thus far unable to support this option.)

The third surprise is that whereas much of the early interest in the IntelDX4 focused on its role on the desktop, Intel began repositioning the part primarily for notebook systems almost immediately after introduction. The device includes power-management features useful to notebook vendors, and its lower supply voltage and reduced power consumption make it a good fit in battery-powered portable system.

### Clock-Multiplier Options

Table 6-17 shows a variety of bus-clock/core-clock combinations that may now be obtained using i486DX2 and IntelDX4 devices.

CPU	Bus Frequency	Clock Multiplier	Core Frequency
i486DX2	25 MHz	2×	50 MHz
	33 MHz	2×	66 MHz
IntelDX4	25 MHz	3×	75 MHz
	33 MHz	3×	100 MHz
	50 MHz	2×	100 MHz

Table 6-17. IntelDX4 and i486DX2 core clock multiplier factors.

In order to obtain 100-MHz operation, system designers must switch to a 33-MHz (or faster) bus. “Merely” tripling the core clock pretty well saturates the system bus, causing system performance to max out. Cranking up the clock another notch to 100 MHz while keeping a 25-MHz bus would not deliver noticeably better performance than a 25/75-MHz configuration. Intel would rather see notebook vendors move to a 33-MHz system bus to improve the performance of the IntelDX4.

The seemingly redundant plethora of frequency options is best understood by considering the needs of notebook vs desktop markets. Most notebook vendors have found it advantageous power-wise to stick with a 25-MHz bus, even as most desktop vendors moved to a 33-MHz bus. Thus, even though a 25/75-MHz IntelDX4 will have performance similar to a 33/66-MHz

i486DX2, notebook vendors will likely prefer the former part, while desktop systems will use the latter.

### Other Enhancements

The IntelDX4 increases the size of the on-chip cache to 16KB, twice the size of earlier family members'. The cache is otherwise identical to its predecessors': it uses the same 16-byte lines, the same four-way set-associative organization, and the same write-through policy as the original i486DX.

The larger cache does help offset the extra cycles lost due to cache misses which result when the CPU clock frequency is raised without a corresponding boost in bus speed. In fact, Intel rates its 100-MHz IntelDX4 more than 50% faster than a 66-MHz i486DX2, even though both use the same 33-MHz system bus.

Despite the 44% reduction in die area that would normally result from the smaller process, the larger cache means the IntelDX4 die is just 5% smaller than the 0.8 micron i486DX2. According to the MPR Cost Model (see **Chapter 15: Manufacturing Costs**), the IntelDX4 will likely cost about 25% more to build than the i486DX2 because of its more expensive process. As the new process matures and defect densities decline, the manufacturing cost of the IntelDX4 will approach that of the i486DX2.

The IntelDX4 includes one other minor addition: an enhanced virtual-8086 mode also supported by the Pentium microprocessor (see **Chapter 12: Intel Pentium Microprocessors**). Intel would like operating-system vendors to take advantage of this new feature, but is having little success. A write-back cache might have improved performance for a much broader range of applications, but Intel appears to be focusing on architectural features that are harder for other x86 vendors to copy.

### System Interface

The IntelDX4 runs internally at 3.3 V; the 0.6-micron transistors in the core cannot tolerate the stress of 5 V operation. To connect to existing system-logic chip sets and standard memory chips, the device has "5-V tolerant" I/O buffers. The system must provide a 3.3-V supply to the chip, however, so the parts cannot be dropped directly into existing 5-V sockets. The lower internal voltage keeps the power consumption reasonable even at the higher clock rates; at 100 MHz, the IntelDX4 is rated at 4.3 W (worst case), 28% less than a 5 V i486DX2 at 66 MHz.

Otherwise, the IntelDX4 is similar to Intel's other 486 chips. It has the same packaging options and uses the same pinout as

the i486DX and i486DX2. It supports the Intel “SL-Enhanced” power-management features and has a 208-pin SQFP packaging option.

Desktop system vendors may support both the i486DX2-66 and the IntelDX4-100 with the same 33-MHz system motherboard and bus. While the IntelDX4-100 may be configured with an external clock of 50 MHz, few PC vendors want to introduce a 486 with a system bus at that speed. The 50/100-MHz IntelDX4 is likely to see little usage initially, although it may become more popular as chip set vendors begin to support 50-MHz devices.

**Vital Statistics**

The IntelDX4 is currently offered in a 168-pin ceramic pin-grid array (PGA) package with the same pinout as that defined for the i486DX PGA package described above.

Intel initially planned to introduce an IntelDX4 in late 1994 that would run with an 83-MHz core frequency obtained by multiplying a 33-MHz external clock by 2.5. This chip would fill the gap between the 66-MHz i486DX2 and the 100-MHz IntelDX4 for desktop systems. At this writing, though, Intel was still working on the clock “two-and-a-halfing” circuit.

The IntelDX4 has a die size of  $339 \times 351$  mils ( $77 \text{ mm}^2$ ) using a 0.6-micron four-layer-metal CMOS process and requires a supply voltage of 3.0 V to 3.6 V.

It is also supplied in a 208-lead SQFP package and is offered in versions with core frequencies of either 75 MHz or 100 MHz. The device dissipates 4.3 W (worst case) at 3.3 V and 100 MHz.



## 6.6 The Intel i486SX Microprocessor

The Intel i486SX microprocessor is a lower-cost version of the i486DX from which the FPU has been removed. Table 6-18 summarizes the general features and specifications of the i486SX microprocessor.

Product Name	Intel i486SX
Introduction Date	April 1991
Prognosis	In production
Device Integration Level	Same as i486DX with floating-point unit disabled or removed
CPU Architecture Level	486 integer-unit instruction set only
Core Technology	Intel 486 core with FPU disabled or removed
Pinout	Subsetted, modified i486DX pinout
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	None (requires auxiliary i487SX or OverDrive)
Operating Voltage	4.75 V to 5.25 V (i486SX)
Frequency Options	25- or 33-MHz core operation @ 5.0 V 25- or 33-MHz core operation @ 3.3 V
Clocking Regime	Core operating frequency = 1 × CLK input
Active Power Dissipation	3.42 W @ 5.0 V and 33 MHz (worst case) 1.27 W @ 3.3 V and 33 MHz (worst case)
Power-Control Features	Standard Intel "SL-Enhanced" feature set
Process Technology	Originally 1.0 $\mu$ two-layer-metal CMOS 0.8 $\mu$ three-layer-metal CMOS
Transistor Count	900,000 transistors
Die Size	270 × 410 mils (72 mm <sup>2</sup> )
Package Options	168-pin ceramic PGA, 196-lead PQFP, or 208-lead SQFP

Table 6-18. Intel i486SX feature summary.

The i486SX extends the 486 integer core architecture to the low-cost/low-performance end of the PC spectrum, and is intended for use in low-cost systems that might previously have chosen a 386-class processor. To minimize production costs and eliminate the need for an expensive PGA socket on the motherboard, the i486SX is optionally available in both standard and "slim" plastic quad flat pack (PQFP and SQFP) packages.

Typical 386-based systems did not, of course, include built-in floating-point capability; this could be added at a later date by inserting an optional 387-class math coprocessor. In the case of the i486SX, floating-point capability can be restored only by removing the defeatured CPU and replacing it with a 486-family device that does not disable its FPU, or by disabling the i486SX and adding a second, full-featured processor such as the Intel i487SX (described in the following section) or an “OverDrive” 486 (described later in this chapter) to the “upgrade coprocessor” socket provided on most i486SX motherboards.

### System Interface

The i486SX in a PGA package has essentially the same pinout as the i486DX, with a few minor changes. Since the FPU has been disabled, FPU-related pins are not provided. One new signal has been defined, two other signals have been disabled, and a fourth was arbitrarily repositioned in order to prevent end users from upgrading an i486SX system to use an i486DX device by merely changing the CPU. The names and functions of pins that changed for the i486SX appear in Table 6-19.

Symbol	Direction	Signal Name/Function	PGA Pin #	i486DX Signal
NMI	In	Non-maskable interrupt	A15	IGNNE#
UP#	In	Upgrade processor present	C11	N.C.
N.C.	—	No connect	B15	NMI
N.C.	—	No connect	C14	FERR#

Table 6-19. Intel i486SX signals.

Floating-point capability can be restored to an i486SX-based system in the field by adding an “OverDrive” upgrade processor (discussed later in this chapter). In order to simplify field upgrades, most i486SX-based motherboards provide a “ZIF” (zero-insertion-force) socket that allows ICs to be easily inserted or removed. Inserting an auxiliary processor in this socket has the effect of fully disabling the original “host” processor.

The interconnections between the i486SX host CPU and the upgrade socket vary, depending on the host revision level and package type. Early i486SX devices in PGA packages were derived from standard i486DX die simply by disabling the on-chip FPU, and several discrete external components were required to disable the part when an upgrade processor was installed. Figure 6-16 illustrates the recommended circuit.

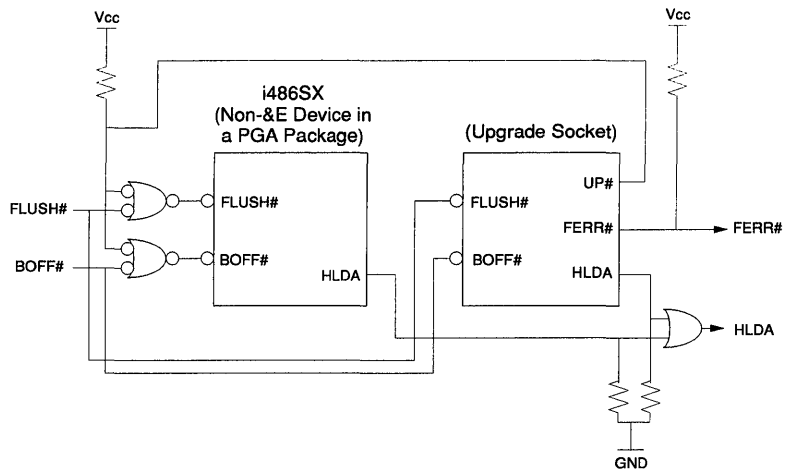


Figure 6-16. Upgrade socket interface to i486SX (PGA).

Later, when the i486SX device was redesigned to actually remove the FPU circuitry, the logic needed to eliminate these discrete components was built into the chip. Later still, the circuitry was added each of the “SL-enhanced” 486 devices, including those supplied in PGA packages. Figure 6-17 shows the simplified upgrade-socket interface allowed by newer i486SX hosts.

Intel currently offers several products that are compatible with the upgrade processor pinout. These include the i487SX and various OverDrive processors described later in this chapter.

**Relative Performance**

Since their CPU cores, caches, and bus interfaces are identical, the performance of integer-only code on the i486SX is essentially identical to that of an i486DX at the same frequency. Nevertheless slight performance discrepancies do seem to arise in

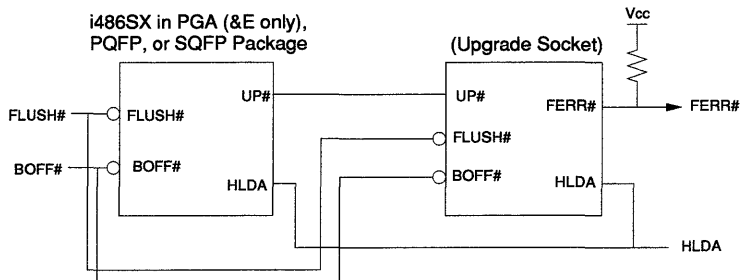


Figure 6-17. Upgrade socket interface to i486SX (PQFP/SQFP).

system-level integer benchmarks. This may reflect the fact that interrupt service routines for the two devices are different, depending on whether or not OS calls and interrupt service routines attempt to save and restore the FPU state.

On floating-point intensive applications, the lack of FPU hardware on an i486SX-only-based system dramatically degrades its performance vs an i486DX, due to the necessity of emulating all floating-point operations in software.

In an i486SX-based system with an upgrade processor installed, the i486SX itself is disabled, so system performance depends entirely on the performance of the upgrade CPU.

**Vital Statistics**

The i486SX has a die size of  $270 \times 410$  mils ( $72 \text{ mm}^2$ ) using a 0.8-micron three-layer-metal CMOS process and allows a supply voltage of either 3.0 V to 3.6 V or 4.75 V to 5.25 V.

The 5.0-V version is available in either a 168-pin PGA or a 196-lead PQFP package and runs at speeds up to 33 MHz. The device dissipates 3.42 W (worst case) at 5.0 V and 33 MHz.

The 3.3-V version uses a 208-lead SQFP package and also runs at up to 33 MHz. The device dissipates 1.27 W (worst case) at 3.3 V and 33 MHz.

## 6.7 The Intel i486SX2 Microprocessor

The Intel i486SX2 microprocessor is, as the name suggests, a clock-doubled version of the 486 family from which the FPU has been removed. Table 6-20 summarizes the general features and specifications of the i486SX2 microprocessor.

Product Name	Intel i486SX2
Introduction Date	March 1994
Prognosis	Stable
Device Integration Level	Same as i486DX but with FPU removed and with an on-chip PLL clock-frequency doubler
CPU Architecture Level	Same as i486DX with FPU removed
Core Technology	Clock-doubled standard 486 core
Pinout	Standard i486SX pinout
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX Bus operates at one half core frequency
Cache Support	Same as i486DX
Floating-Point Support	None; requires auxiliary upgrade processor
Operating Voltage	4.75V to 5.25V
Frequency Options	25 (50-MHz core freq)
Clocking Regime	Core operating frequency = $2 \times$ CLK input Bus operating frequency = CLK input
Active Power Dissipation	6.0W @ 5.0 V and 66 MHz (worst-case)
Power-Control Features	None
Process Technology	0.8 $\mu$ three-layer metal CMOS
Transistor Count	1.0 million transistors
Die Size	270 $\times$ 410 mils (72 mm <sup>2</sup> )
Package Options	168-pin PGA

Table 6-20. Intel i486SX2 feature summary.

### Vital Statistics

The i486SX2 has a die size of 270  $\times$  410 mils (72 mm<sup>2</sup>) using a 0.8-micron three-layer-metal CMOS process and requires a supply voltage of 4.75 V to 5.25 V. The part is available in either a 168-pin PGA or a 196-lead PQFP package and is offered in versions with core frequencies of either 50 MHz or 66 MHz. The device dissipates 6.0 W (worst case) at 5.0 V and 66 MHz.

The system interface and pinout of the i486SX2 are the same as those of the i486SX device discussed earlier in this chapter.

## 6.8 The Intel i487SX Microprocessor

The Intel i487SX microprocessor is the upgrade processor that restores the floating-point capability excised from the i486SX. Table 6-21 summarizes the general features and specifications of the i487SX microprocessor.

Product Name	Intel i487SX
Introduction Date	April 1991
Prognosis	Poor
Device Integration Level	Same as the i486DX
CPU Architecture Level	Same as the i486DX
Core Technology	Same as the i486DX
Pinout	Augmented, rearranged i486DX pinout
Data Bus Width	Same as the i486DX
Physical Addressability	Same as the i486DX
Data-Transfer Modes	Same as the i486DX
Cache Support	Same as the i486DX
Floating-Point Support	Same as the i486DX
Operating Voltage	4.75 V to 5.25 V
Frequency Options	20-, 25-, or 33-MHz core operation
Clocking Regime	Core operating frequency = 1 × CLK input
Active Power Dissipation	3.4 W @ 5.0 V and 33 MHz (worst case)
Power-Control Features	None
Process Technology	1.0 $\mu$ two-layer-metal CMOS 0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	6.9 × 11.9 mils (81 mm <sup>2</sup> )
Package Options	169-pin PGA
Notes	Same die as i486DX with modified pinout and different CPU identification code at reset.

Table 6-21. Intel i487SX feature summary.

**Features** The i487SX is promoted by Intel as being a floating-point “math coprocessor,” and its nomenclature was chosen to promulgate the 8086/8087, 80286/80287, 386/387 numbering sequence. In fact, the i487SX actually contains a standard, fully functional i486DX die in a PGA package, with a slightly modified pinout, one new signal, and an additional “key” pin to assure proper orientation in the motherboard socket. The only modification made to the die itself is that it provides a different CPU identification code following reset.

The i487SX is not an OEM product; most often it is sold directly to end users via Intel's retail distribution channels as an after-market upgrade.

### System Interface

The i487SX provides essentially the same pin functions as the i486DX, with the addition of one new output signal, an alignment key, and the arbitrary repositioning of one other pin. Table 6-22 summarizes the names and functions of i487SX pins that have changed relative to the i486DX pinout.

Symbol	Direction	Signal Name/Function	PGA Pin #	i486DX Signal
UP#	Out	Upgrade processor present; disables host CPU. Internally bonded to Vss (Gnd)	B14	N.C.
FERR#	Out	Floating-point error	A13	N.C.
KEY	—	Key pin; assures proper alignment in PGA socket	D4	—
N.C.	—	Not connected	C14	FERR#

Table 6-22. Intel i487SX interface signals.

### Vital Statistics

The i487SX has a die size of  $6.9 \times 11.9$  mils ( $81 \text{ mm}^2$ ) using a 0.8-micron three-layer metal CMOS process and requires a supply voltage of 4.5 V to 5.5 V. It is supplied in a 169-pin PGA package and runs at speeds up to 33 MHz. The device dissipates 3.4 W (worst case) at 5.0 V and 33 MHz.

## 6.9 The IntelDX2 OverDrive Microprocessor

The IntelDX2 OverDrive microprocessor is a device that lets end users upgrade their 486-based PCs to achieve i486DX2 levels of performance. Table 6-23 summarizes the general features and specifications of the IntelDX2OverDrive microprocessor.

Product Name	IntelDX2 OverDrive
Introduction Date	June 1992
Prognosis	In production
Device Integration Level	Same as i486DX2
CPU Architecture Level	Same as i486DX
Core Technology	Clock-doubled 486 core
Pinout	Both standard i486DX and i487SX pinouts
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	4.75 V to 5.25 V
Frequency Options	40-, 50-, or 66-MHz core operation
Clocking Regime	Core operating frequency = 2 × CLK input Bus operating frequency = CLK input
Active Power Dissipation	6.0 W @ 5.0 V and 66 MHz (worst case)
Power-Control Features	None
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.185 million transistors
Die Size	273 × 468 mils (81 mm <sup>2</sup> )
Package Options	168-pin PGA (i486DX-compatible version) or 169-pin PGA (i487SX-compatible version)
Other Features	Faster versions offer an integrated heat sink
Notes	End-user retail version of the i486DX2. Pinouts available to match either a standard i486DX PGA or an i487SX upgrade socket.

Table 6-23. IntelDX2 OverDrive feature summary.

**Features** The i486DX2 silicon lives a double life—one as a performance enhancer for OEM systems and another as an end-user upgrade product. This version, called the IntelDX2 OverDrive processor, is pin-compatible with the i487SX upgrade for i486SX systems.



When an IntelDX2 OverDrive processor is installed, the original processor is electrically disabled. Intel discourages users from physically removing the original processor, in part because of potential damage to the system board and in part because it doesn't want to create a supply of used 486 chips. Once an OverDrive processor is installed, however, it should be possible to remove the original CPU without affecting system operation.

Just to make things messy, Intel actually offers two versions of the IntelDX2 OverDrive processor: one for 16- and 20-MHz systems, and another for 25-MHz systems. Confusingly, IntelDX2 OverDrive processors are rated by the clock rate of the system they plug into, whereas i486DX2 processors are rated by their internal clock rate. Thus, a 25-MHz IntelDX2 OverDrive corresponds to an i486DX2-50.

The apparent reasoning here is that OverDrive devices are marketed to end users who specify a device based on the CPU frequency originally used by the system being upgraded, whereas the i486 DX2 is marketed to OEM designers who specify a device based on the approximate core throughput.

The biggest difference between the i486DX2 and the IntelDX2 OverDrive processor is the sales channel: the i486DX2 is an OEM product, while the OverDrive processor is an end-user upgrade, to be sold through retail channels.

**Pinout** The IntelDX2 OverDrive pinout matches that defined by the i487SX, which differs from the standard i486DX in that an alignment pin has been included to make it harder to insert the chip incorrectly, a new "upgrade present" signal has been defined, and one signal has been arbitrarily repositioned to a different pin. Intel apparently wants the various 486 versions not to be pin-compatible so that it can more easily pursue different pricing and marketing strategies for the different versions.

The IntelDX2 OverDrive processor effectively obsoletes the i487SX, since it provides considerably higher performance at a slightly higher price. The exact performance boost provided by an OverDrive processor depends on the application's cache performance. On trivial benchmarks such as Landmark, it provides a 100% performance increase; other small benchmarks, such as Dhrystone, show a 90–95% increase. Application-level performance is typically boosted 40–70%, according to Intel's benchmarks. In a 25-MHz system with a 64-Kbyte external cache, SPECmark89 performance was increased 66%, from 8.8 to 14.6 (21.3 SPECint89, 11.3 SPECfp89).

Because the CPU core is running at twice the clock rate of a standard 486 processor, the IntelDX2 OverDrive processor has higher bus utilization and is therefore more sensitive to memory-system performance. Thus, a system with a fast DRAM system and a second-level cache will benefit more from an OverDrive processor than will a system without cache or with slow DRAM.

Intel would like to see *all* system vendors—even those using the i486DX and i486DX2—begin putting OverDrive sockets on their motherboards, which would increase Intel's potential aftermarket and allow the same OverDrive processors to be used in i486SX and i486DX systems. In this context, the only new device needed is a 33-MHz OverDrive processor. Intel does not yet offer an OverDrive processor to beef up 50-MHz systems; see the description of future Pentium derivatives (**Chapter 12: Intel Pentium Microprocessors**) for information on expected developments in this arena.

Intel also offers OverDrive processors for i486DX systems that don't have an OverDrive socket. In this case, the i486DX must be removed from its socket and replaced with the OverDrive processor. If this sounds a lot like an i486DX2, it is: the only difference is the name and the fact that Intel sells the i486DX2 primarily to OEM system makers, and sells the equivalent OverDrive processor directly to users through retail channels.

**Vital Statistics**

The IntelDX2 OverDrive has a die size of  $273 \times 468$  mils ( $81 \text{ mm}^2$ ) using a 0.8-micron three-layer-metal CMOS process and requires a supply voltage of 4.75 V to 5.25 V.

The 5-V version is supplied in a 168-pin PGA package and is offered in versions with core frequencies of either 50 MHz or 66 MHz. The device dissipates 6.0 W (worst case) at 5.0V and 66 MHz. Its pinout matches the i487SX devices discussed earlier in this chapter.

## 6.10 The IntelDX4 OverDrive Microprocessor

The IntelDX4 OverDrive microprocessor is a device that lets end users upgrade their 486-based PCs to achieve IntelDX4 levels of performance. Table 6-23 summarizes the general features and specifications of the IntelDX4 OverDrive microprocessor.

Product Name	IntelDX4 OverDrive
Introduction Date	October 1994
Prognosis	In production
Device Integration Level	Same as IntelDX4
CPU Architecture Level	Same as IntelDX4
Core Technology	Clock-tripled 486 core
Pinout	Both standard i486DX and i487SX pinouts
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as IntelDX4
Floating-Point Support	Same as i486DX
Operating Voltage	4.75 V to 5.25 V to package; operates at 3.3 V internally
Frequency Options	100-MHz core operation
Clocking Regime	Core operating frequency = 2 × CLK input Bus operating frequency = CLK input
Active Power Dissipation	6.5 W @ 5.0 V and 66 MHz (worst case)
Power-Control Features	None
Process Technology	0.6 $\mu$ four-layer-metal BiCMOS
Transistor Count	1.6 million transistors
Die Size	339 × 351 mils (77 mm <sup>2</sup> )
Package Options	168-pin PGA (i486DX-compatible version) or 169-pin PGA (i487SX-compatible version)
Other Features	Package-mounted voltage regulator Integrated heat sink
Notes	End-user retail version of the IntelDX4. Pinouts match a standard i487SX upgrade socket.

Table 6-24. IntelDX4 OverDrive feature summary.

## 6.11 The IntelSX2 OverDrive Microprocessor

The IntelSX2 OverDrive microprocessor is a device that lets end users upgrade their 486-based PCs to achieve IntelDX4 levels of performance. Table 6-23 summarizes the general features and specifications of the IntelSX2 OverDrive microprocessor.

Product Name	IntelSX2 OverDrive
Introduction Date	October 1994
Prognosis	In production
Device Integration Level	Same as i486SX2
CPU Architecture Level	Same as i486SX2
Core Technology	Clock-doubled i486SX core
Pinout	Same as i487SX
Data Bus Width	Same as i486DX
Physical Addressability	Same as i486DX
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486SX
Operating Voltage	4.75 V to 5.25 V
Frequency Options	40-, 50-, or 66-MHz core operation
Clocking Regime	Core operating frequency = $2 \times$ CLK input Bus operating frequency = CLK input
Active Power Dissipation	4.1 W @ 5.0 V and 66 MHz (worst case)
Power-Control Features	None
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.0 million transistors
Die Size	270 $\times$ 410 mils (72 mm <sup>2</sup> )
Package Options	168-pin PGA (i486DX-compatible version) or 169-pin PGA (i487SX-compatible version)
Other Features	Faster versions offer an integrated heat sink
Notes	End-user retail version of the i486SX2 Pinouts match a standard i487SX upgrade socket

Table 6-25. IntelSX2 OverDrive feature summary.

## 6.12 The Intel i486SL Microprocessor

The Intel i486SL is designed for low-power notebook and subnotebook-class PCs. It combines the integer pipeline, cache, FPU, and other on-chip resources of a standard 486 with the power-conservation features and higher system integration of the i386SL. Table 6-26 summarizes the general features and specifications of the i486SL.

Product Name	Intel i486SL
Introduction Date	November 1992
Prognosis	On life support
Device Integration Level	Standard 486-class CPU and MMU with 8KB on-board combined Instruction/Data cache and FPU Direct interconnections to system DRAM Direct ISA backplane control logic and drivers Automatic power-management logic
CPU Architecture Level	Complete 486 integer and floating-point ISA with Intel SMM architecture extensions
Core Technology	Static redesigned Intel 486 core
Pinout	Custom
Data Bus Width	ISA bus: 16 data bits plus two parity bits Local DRAM system bus: 32 data bits
Physical Addressability	4 gigabytes (Address pins A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	ISA- and PI-bus-compatible transfers Four-word burst-mode system transfers Dynamic bus resizing for 8-bit transfers
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	3.0-V–3.6-V core 3.0-V–5.5-V I/O interface
Frequency Options	25 MHz or 33 MHz core operation
Clocking Regime	Core operating frequency = 1 × CLK input
Active Power Dissipation	1.16 W @ 3.3 V and 25 MHz (core only; worst case)
Power-Control Features	3.3 V operation; static core design; Intel SMM
Process Technology	0.8 $\mu$ three-layer-metal CMOS
Transistor Count	1.4M transistors
Die Size	488 × 532 mils (167 mm <sup>2</sup> )
Package Options	196-pin PQFP, 208-pin slim QFP, or 227-land LGA
Other Features	Configurable I/O drive voltage and current
Notes	A part ahead of its time

Table 6-26. Intel i486SL feature summary.

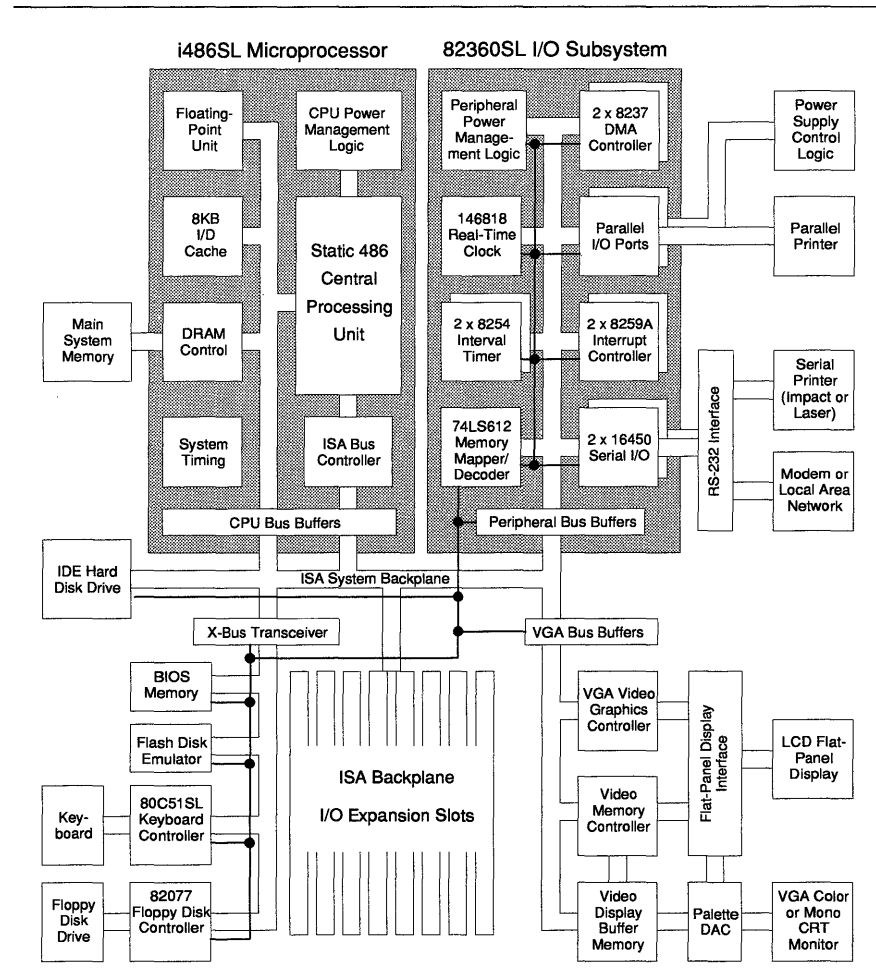


Figure 6-18. Intel i486SL system partitioning.

## Background

The i486SL was developed in an attempt to move the portable processor market to the 486 architecture. It is based on a 486 core that has been modified to provide fully static operation and add system-management mode (SMM). In addition to its standard 486 CPU core, cache, and FPU, the i486SL includes a DRAM controller and an ISA bus interface.

Figure 6-18 shows the functional partitioning of an i486SL-based system. The i486SL is designed to work with the same 82360SL I/O chip designed for use with the 386SL, which provides peripheral power management, timers, a real-time clock, interrupt and DMA control, two serial ports, and a parallel port. Intel says these functions were left off the CPU because of pin limitations, not die-size barriers.

The i486SL is designed for mixed-voltage systems. The processor core chip logic always operates at 3.3 V. A separate power pin controls the voltage for the bus and DRAM interfaces, allowing them to run at either 3.3 V or 5 V. While 3.3-V versions of the 82360SL I/O chip are now available to complement these 3.3-V i486SL processors, i486SL-based systems are likely to continue using mixed-voltage operation until 3.3-V DRAM and peripherals become more widely available.

The i486SL is available in the same 196-pin PQFP as the 386SL, but the pinout is different because the external cache RAM interface is eliminated and the data bus width is increased to 32 bits. The i486SL is also offered in a smaller 208-pin SQFP (slim quad flat pack) that has a finer lead pitch. Even with the standard PQFP package, the total board space required is reduced because there is no need for external cache RAMs or an FPU (or FPU socket). For designers who prefer to socket their CPUs, the i486SL is also available in a 227-lead LGA (land grid array).

Like the 386SL, the i486SL provides a high-speed peripheral interface (PI) bus that uses the 16-bit data path of the ISA bus interface, but with a separate set of control signals. This frees speed-critical peripherals, such as display controllers, from the antiquated timing constraints of the ISA bus. The PI bus may also be useful for flash-memory disk replacements or PCMCIA interfaces.

**“The Best Laid Plans...”**

When Intel introduced the i486SL, a wide range of i486SL follow-on products were promised. These included versions with and without the floating-point unit, and with clock rates as low as 12 MHz to reach low price points. In the end, Intel offered only the version with the on-board FPU, and only in 25- and 33-MHz flavors.

The i486SL captured a number of design wins, including systems introduced as late as this year. Nevertheless, it appears the integration path represented by the i486SL—adding a bus controller, DRAM controller, and system logic to an already crowded processor chip—proved to be a misguided effort.

The problem was largely simple economics: the i486SL chips are expensive to build. The peripheral functions included on the i486SL swell its die size to twice that of the i486DX, making it a much more expensive chip to produce. These functions can be replicated in an external chip set at very low cost, so it is more economical to use a standard i486DX and a third-party chip set.

The crux of the problem is that Intel's profit margins on its CPUs are far higher than the profits available for system-logic functions. Integrating these functions onto the same die as a processor forced the company to accept a lower margins—not a popular choice at Intel, particularly when the company has no excess factory capacity for lower-profit chips. The new strategy forces the system-logic functions back onto an external chip set, permitting Intel to retain its high margins.

Also, the i486SL stifled creativity. System designers could choose from a veritable banquet of design options supported by the part, but it was hard to add unique new capabilities to an i486SL system. OEMs found this made it difficult to distinguish their products from those of their competitors.

The i486SL—with its limited performance, limited configuration options, and high manufacturing cost—was also a poor match for the “Green PC” products. These new PCs implement power-management capabilities in desktop systems and require access to the full spectrum of Intel processor offerings. The SL-enhanced 486 chips allow system vendors to build power-wise PCs without any additional CPU cost. In addition, low-end and midrange systems can cut CPU power in half without a performance penalty, by using 3.3-V processors.

In short, the i486SL was over-integrated for today's economics. In the long run, i486SL-style integration may make sense, but not without denser chip geometries and more carefully compacted designs. Adding several blocks of random logic and high-power buffers to a tightly packed CPU core dramatically decreases the chip's overall transistor density, and the added transistors have relatively low value because of the extremely low margin pricing prevalent in the chip set market.

### **Vital Statistics**

The i486SL uses the 0.8-micron, three-layer-metal process also used for the 50-MHz i486DX and the i486DX2 products. Due to the more advanced process, the 1.4-million-transistor i486SL die is about the same size as the 850,000-transistor 386SL. (The 386SL is 13.1 x 12.9 mm on the 1.0-micron process, or about 169 mm<sup>2</sup> compared with 167 mm<sup>2</sup> for the i486SL.)

A more revealing comparison, however, is with the 0.8-micron i486DX, which is a mere 82 mm<sup>2</sup>—half the size of the i486SL. This means that the production cost of the i486SL will be dramatically higher than that of the i486DX, yet an i486DX-25 is priced higher than the i486SL.



## 6.13 The Intel “RapidCAD” 386 Microprocessor

And now for something completely different: an Intel chip that’s of interest to practically nobody, save perhaps connoisseurs of high-tech trivia. The “RapidCAD” 386 microprocessor is Intel’s “stealth” entry in the 386 sweepstakes: a two-chip set designed to improve the floating-point performance of existing PCs based on the i386DX CPU and i387DX FPU. Table 6-27 summarizes the general features and specifications of the RapidCAD 386 microprocessor.

Product Name	Intel “RapidCAD” 386
Introduction Date	N.A.
Prognosis	Lost in the shuffle
Device Integration Level	Pipelined 32-bit IEU and PMMU Microcoded 80-bit floating-point unit
CPU Architecture Level	Standard 386 IU and 387 FPU instruction set
Core Technology	Modified Intel 486 core
Pinout	Same as standard 386DX and 387DX
Data Bus Width	Same as standard 386DX
Physical Addressability	Same as standard 386DX
Data-Transfer Modes	Same as standard 386DX
Cache Support	On-chip 486 cache disabled; optional external 82385DX cache controller or 82395DX integrated cache peripheral
Floating-Point Support	On-chip 80-bit microcoded FPU
Operating Voltage	4.75 V to 5.25 V
Frequency Options	33-MHz core operation
Clocking Regime	Core operating frequency = CLK2 freq + 2
Active Power Dissipation	N.A.
Power-Control Features	None
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Transistor Count	N.A.
Die Size	414 $\times$ 619 mils (165 mm <sup>2</sup> ) (IU replacement device)
Package Options	132-pin PGA (IU replacement device) 68-pin PGA (FPU replacement device)
Notes	Intel’s “stealth” entry in the 386 sweepstakes

Table 6-27. Intel “RapidCAD” 386 feature summary.

### Features

This report groups the RapidCAD with Intel’s 486-family products because, despite their part numbers, these devices derive from 486 core technology. The “i386DX” half of the two-chip set

is, in fact, an i486DX core with various features to assure i386DX socket compatibility. The i486DX I/D cache has been stripped from the die, leaving large blank rectangles where it used to reside, and certain microcode routines have been rewritten to slow them down to near 386-class speeds (I'm not making this up), but integer code still runs 10% to 25% faster than with the original 386 part due to the improved pipeline.

Where the RapidCAD product shines, not surprisingly, is in its floating-point throughput. The 486 FPU is still enabled and can effectively double the performance of the i387DX device it supersedes.

There's just one hitch: in a conventional 386/387 system design, floating-point errors are signaled by the FPU coprocessor, whereas in a conventional 486 design the FPU resides within the main processor package. In order to preserve compatibility with existing motherboards, the "387" piece of the RapidCAD chip set is nothing more than an addressable latch, which can be induced to set and clear error-reporting pins in response to special bus cycles put out by the CPU. From there, the error signals are passed to an external interrupt controller, to be processed according the original design.

### **Vital Statistics**

The RapidCAD 386 integer unit is housed in a 132-pin ceramic pin-grid array package with a pinout identical to the i386DX PGA package defined in **Chapter 5: Intel 386 Microprocessors**. The FPU unit is packaged in a 68-pin PGA package with the same pinout as the i387DX PGA package. Both devices are offered only in a version that runs with up to a 33-MHz processor clock.

## 6.14 Futures

**The “P24T”** Since mid-1992 Intel has been promising system vendors an OverDrive processor based on the Pentium processor core for i486DX2 systems. This device, code named “P24T,” will use a 240-pin PGA package derived by adding a fourth row of pins around the outside of a standard 169-pin OverDrive socket. See **Chapter 12: The Intel Pentium Family** for details.

## 6.15 Commentary

Intel now offers more than two dozen 486-class processors, each with a different combination of features, optimization modes, pinouts, packages, and distribution channels. Table 6-28 summarizes the distinctions between these chips.

The breadth of these offerings has led to considerable confusion among system designers, as has the multiplicity of pinouts offered for products that might otherwise be fully pin compatible. Table 6-29 summarizes how the functions on certain pins of the i486DX PGA package seem to drift aimlessly over time.

Even more confusing, Intel now uses the term OverDrive to describe chips with not only three different PGA pinout patterns, but three different PGA pin *counts*. OverDrive chips that replace the original CPU in i486DX-based systems use one pinout, OverDrive chips that are socket-compatible with the original “generic” i487SX use another, and future P24T Pentium-based OverDrive chips use yet a third. This has surely lead to a higher confusion quotient in the market.

If this be madness, might there be some method to it?

It's possible, of course, that Intel believes it may profit all the more from enhanced customer confusion. Once a customer throws up his hands in desperation and calls in an expert for help, the company with the best brand-name recognition and the largest base of sales “experts” will be better able to protect its market share.

More likely, though, Intel is merely pursuing to an extreme a strategy it has long followed, that of offering system vendors a range of design options.

Product Name	SL-Enhanced?	Vcc (Core/IO)	Clock Factor	Cache Size	FPU?	Max Freq (Core/Bus)	Package Type	Pinout Class	Distribution Channel (see notes)
i486SX	None	5.0 V	x1	8KB	No	33 MHz	PGA or PQFP	i486SX	OEM
i486SX-LP	None	5.0 V	+2	8KB	No	33 MHz	PQFP	i486SX	EOL
i486SX	Yes	3.3 V	x1	8KB	No	33 MHz	SQFP	i486SX	OEM
i486SX-LP	Yes	3.3 V	+2	8KB	No	33 MHz	SQFP	i486SX	EOL
i486SX	Yes	5 V	x1	8KB	No	33 MHz	PGA or PQFP	i486SX	OEM
i486SX-LP	Yes	5 V	+2	8KB	No	33 MHz	PQFP	i486SX	EOL
i486SX2	Yes	5 V	x2	8KB	No	50 MHz/ 25 MHz	PGA	i486SX	OEM
i486DX	No	5.0 V	x1	8KB	Yes	33 MHz	PGA or PQFP	i486DX	OEM
i486DX-LP	No	5.0 V	+2	8KB	Yes	33 MHz	PQFP	i486DX	OEM
i486DX	Yes	3.3 V	x1	8KB	Yes	33 MHz	SQFP	i486DX	OEM
i486DX-LP	Yes	3.3 V	+2	8KB	Yes	33 MHz	SQFP	i486DX	EOL
i486DX	Yes	5 V	x1	8KB	Yes	33 MHz	PGA or PQFP	i486DX	OEM
i486DX-LP	Yes	5 V	+2	8KB	Yes	33 MHz	PQFP	i486DX	EOL
i486DX-50	No	5 V	x1	8KB	Yes	50 MHz	PGA	i486DX	OEM
i486DX2	No	5 V	x2	8KB	Yes	66 MHz/ 33 MHz	PGA	i486DX	OEM
i486DX2	Yes	3.3 V	x2	8KB	Yes	50 MHz/ 25 MHz	SQFP	i486DX	OEM
i486DX2	Yes	5 V	x2	8KB	Yes	66 MHz/ 33 MHz	PGA	i486DX	OEM
i487SX	No	5 V	x1	8KB	Yes	33 MHz	PGA	i487SX	Retail
OverDrive 486	No	5 V	x2	8KB	Yes	66 MHz/ 33 MHz	PGA	i486DX/ i487SX	Retail
i486SL	Yes	3.3V/ 5.0V	x1	8KB	Yes	33 MHz	SQFP	Custom	EOL
IntelDX4	Yes	3.3V	x2	16KB	Yes	100 MHz/ 50 MHz	PGA or SQFP	i486DX	OEM
RapidCAD 386	No	5 V	+ 2	None	Yes	33 MHz	PGA	i386DX/ i387DX	EOL

Table 6-28. Intel 486 product feature comparison.

Notes: OEM = Direct order from Intel. EOL = Custom order only. Retail = Sold directly to end users through retail outlets.

Initially, 486 microprocessors were offered with operating frequencies of 25 MHz, 33 MHz, etc. With the introduction of the IntelDX4, Intel has attained a position from which it can now offer a spectrum of 486 devices with core frequencies anywhere from 20 to 100 MHz, with a smoothly increasing performance increment of 20–25% between successive models. Because these options can all be supported with a single 20- to 33-MHz motherboard, system vendors have little reason not to supply all of them. Intel's strategy avoids leaving any gaps for its competitors to fill.

Pin	i486SX	i486SX &E	i486DX	i486DX &E	i486DX-50	i486DX-50 &E	i487SX/ Upgrade
A3	N.C.	TCK	N.C.	TCK	TCK	TCK	N.C.
A13	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	FERR#
A14	N.C.	TDI	N.C.	TDI	TDI	TDI	N.C.
A15	NMI	NMI	IGNNE#	IGNNE#	IGNNE#	IGNNE#	IGNNE#
B10	N.C.	SMI	N.C.	SMI	N.C.	SMI	N.C.
B14	N.C.	TMS	N.C.	TMS	TMS	TMS	UP# (Gnd)
B15	N.C.	N.C.	NMI	NMI	NMI	NMI	NMI
B16	N.C.	TDO	N.C.	TDO	TDO	TDO	N.C.
C10	N.C.	SRESET	N.C.	SRESET	N.C.	SRESET	N.C.
C11	N.C.	UP#	N.C.	UP#	N.C.	UP#	N.C.
C12	N.C.	SMIACT#	N.C.	SMIACT#	N.C.	SMIACT#	N.C.
C14	N.C.	N.C.	FERR#	FERR#	FERR#	FERR#	N.C.
D4	N/A	N/A	N/A	N/A	N/A	N/A	Key (N.C.)
G15	N.C.	STPCLK#	N.C.	STPCLK#	N.C.	STPCLK#	N.C.

Table 6-29. Intel 486 product PGA pinout comparison.

Intel can now slash the price of its i486DX2 processors, forcing other x86 vendors to match the new prices and accept lower margins. With the unique IntelDX4 parts, however, Intel can maintain its traditional high margins. For end users, these changes will translate into higher performance at all system price points. Low-end i486DX-based designs will likely jump to the i486DX2 as Intel drops the price of that part. Midrange i486DX2 boxes will move to the IntelDX4, while high-end systems will move from today's 60-MHz Pentium to faster 90-MHz and 100-MHz parts. Thus, PC users at every price point will see a 50% performance increase for the same system price.

As far as the upgrade aftermarket is concerned, Intel must be thrilled at the prospect of selling more than one processor per system. Users should be happy as well, since OverDrive processors give them a low-cost upgrade path. The only downside is for system makers, which may be unhappy to find that they no longer get to sell CPU upgrades, and that users may hold on to their computers longer.

From the system maker's perspective, it's hard to make much profit from the pass-through resale of a chip-level product. Thus, some may prefer to sell upgrade CPU cards, offering larger caches or other features in addition to the faster processor. Some system makers, more focused on profits than on benefits to their customers, might ensure that their BIOS contains speed-sensitive code as a way of making users come to them for

an authorized upgrade, so they can charge more than the street price of an i486DX2 chip for the new processor and a new BIOS ROM.

## 6.16 For More Information...

Additional technical information on the Intel 486 product line may be found in the following publications:

- |                                       |   |
|---------------------------------------|---|
| <b>Vendor Publications</b>            | <ol style="list-style-type: none"> <li>1: <i>Intel OverDrive Processor Performance Report</i>. Intel Corporation, 8/94, order #297130-007.</li> <li>2: <i>Intel486 DX Microprocessor Data Book</i>. Intel Corporation, 6/91, order #240440-004.</li> <li>3: <i>Intel486 DX2 Microprocessor Data Book</i>. Intel Corporation, 7/92, order #241245-002.</li> <li>4: <i>Intel486 DX2 Microprocessor Performance Brief</i>. Intel Corporation, 3/92, order #241254-001. <i>(Text and graphs on i486DX2 performance using standard benchmarks and applications.)</i></li> <li>5: <i>Intel486 Microprocessor Family Product Briefs</i>. Intel Corporation, 1992, order #240459-005. <i>(Brochure full of 486 family product briefs.)</i></li> <li>6: <i>Intel486 SL Microprocessor SuperSet Data Book</i>. Intel Corporation, 11/92, order #241325-001.</li> <li>7: <i>Intel486 SL Microprocessor SuperSet Programmer's Reference Manual</i>. Intel Corporation, 11/92, order #241327-001.</li> <li>8: <i>Intel486 SL Microprocessor SuperSet System Design Guide</i>. Intel Corporation, 11/92, order #241326-001.</li> <li>9: <i>Intel486 SX Microprocessor Data Book</i>. Intel Corporation, 8/92, order #240950-003.</li> <li>10: <i>Microprocessors Data Book Volume II: Intel486 Microprocessors</i>. Intel Corporation, 1994, order #241731-001.</li> <li>11: <i>Write-Back Enhanced IntelDX2 Processor Performance Brief Release 1.0</i>. Intel Corporation, 10/94, order #242308-001.</li> </ol> |
| <b>Microprocessor Report Articles</b> | <ol style="list-style-type: none"> <li>12: <i>Intel 80486 Rumored to Use Downloadable Microcode</i>. John Wharton, MPR vol. 2 no. 10, 10/88, pg. 6. <i>(Unsubstantiated and erroneous speculation on possible new state-of-the-art implementation techniques.)</i></li> </ol>   |

- 13: *Intel 486 to be Announced at Comdex*. MPR vol. 3 no. 3, 3/89, pg. 2. (*Most Significant Bits item.*)
- 14: *The Intel 80486 Strikes Back\**. John Wharton, MPR vol. 3 no. 4, 4/89, pg. 1. (*Cover story.*)
- 15: *Revenge of the CISCs*. MPR vol. 3 no. 4, 4/89, pg. 13. (*Feature article.*)
- 16: *Intel's 486 Bus Optimized for Cache Support\**. Michael Slater, MPR vol. 3 no. 5, 10/89, pg. 8. (*Feature article.*)
- 17: *Parallel 486 Pipelines Produce Peak Processor Performance\**. John Wharton, MPR vol. 3 no. 6, 6/89, pg. 13. (*Feature article.*)
- 18: *Intel Restructures 486 Control Flags*. MPR vol. 3 no. 8, 8/89, pg. 2. (*Most Significant Bits item.*)
- 19: *Guidelines for 486 Software Design\**. John Wharton, MPR vol. 3 no. 8, 8/89, pg. 10. (*Feature article.*)
- 20: *Intel Can't Quite Make 486s Yet*. MPR vol. 3 no. 9, 9/89, pg. 4. (*Most Significant Bits item.*)
- 21: *Intel Says 486 is in Production*. MPR vol. 3 no. 10, 10/89, pg. 2. (*Most Significant Bits item.*)
- 22: *Intel Says Corrected 486 Chips are Shipping*. MPR vol. 3 no. 12, 12/89, pg. 2. (*Most Significant Bits item.*)
- 23: *33-MHz 486 Released*. MPR vol. 4 no. 9, 5/18/90, pg. 4. (*Most Significant Bits item.*)
- 24: *Intel Samples "TurboCache486" Module*. MPR vol. 4 no. 9, 5/18/90, pg. 5. (*Most Significant Bits item.*)
- 25: *Intel to Skip 40-MHz, Ship 50-MHz 486 in '91*. MPR vol. 4 no. 18, 10/17/90, pg. 5. (*Most Significant Bits item.*)
- 26: *Intel's P23 is Low-Cost 486*. MPR vol. 4 no. 22, 11/28/90, pg. 4. (*Most Significant Bits item.*)
- 27: *Intel Previews High-Speed 486 Processor\**. Michael Slater, MPR vol. 5 no. 4, 3/6/91, pg. 1. (*Cover story.*)
- 28: *What Comes After the 486?\**. John Wharton, MPR vol. 5 no. 5, 3/20/91, pg. 11. (*Oblique Perspective column.*)
- 29: *Intel's 486SX Aims to Displace 386DX\**. Michael Slater, MPR vol. 5 no. 8, 5/1/91, pg. 1. (*Cover story.*)
- 30: *Have the Marketing Gurus Gone Too Far?\**. John Wharton, MPR vol. 5 no. 9, 5/15/91, pg. 16. (*Oblique Perspective column.*)

- 31: *Intel Announces 50-MHz 486\**. Michael Slater, MPR vol. 5 no. 12, 6/26/91, pg. 1. (Cover story.)
- 32: *Intel Hits Snag with 50-MHz 486*. MPR vol. 5 no. 16, 9/4/91, pg. 5. (Most Significant Bits item.)
- 33: *Intel offers New 386SL, 486SX Versions*. MPR vol. 5 no. 18, 10/2/91, pg. 4. (Most Significant Bits item.)
- 34: *Intel Previews Upgrade Processors*. MPR vol. 5 no. 19, 10/16/91, pg. 4. (Most Significant Bits item.)
- 35: *IBM and Intel To Jointly Develop x86 Chips\**. Michael Slater, MPR vol. 5 no. 22, 12/4/91, pg. 18. (Most Significant Bits item.)
- 36: *Intel Clock-Doubler 486 Debuts as 486DX2\**. Michael Slater, MPR vol. 6 no. 3, 3/4/92, pg. 19. (Feature article.)
- 37: *Intel Slashes Prices on 486SX*. MPR vol. 6 no. 7, 5/27/92, pg. 4. (Most Significant Bits item.)
- 38: *Cyrix Challenges 486DX with C486DLC\**. Michael Slater, MPR vol. 6 no. 8, 6/17/92, pg. 1. (Cover story.)
- 39: *Intel Ships OverDrive Processors\**. Michael Slater, MPR vol. 6 no. 8, 6/17/92, pg. 7. (Feature article.)
- 40: *Intel Announces 66-MHz 486DX2*. MPR vol. 6 no. 11, 8/19/92, pg. 4. (Most Significant Bits item.)
- 41: *Write Buffers Enhance 486 Performance\**. Mark Thorson, MPR vol. 6 no. 11, 8/19/92, pg. 10. (Feature article.)
- 42: *Intel Announces DX OverDrive Processors*. MPR vol. 6 no. 12, 9/16/92, pg. 4. (Most Significant Bits item.)
- 43: *Intel Moves 486SX Up a Notch*. MPR vol. 6 no. 13, 10/7/92, pg. 4. (Most Significant Bits item.)
- 44: *Intel's 486SL Follows in 386SL's Footsteps\**. Michael Slater, MPR vol. 6 no. 15, 11/18/92, pg. 1. (Cover story.)
- 45: *Intel Launches "OverDrive Ready" Campaign*. MPR vol. 6 no. 16, 12/9/92, pg. 5. (Most Significant Bits item.)
- 46: *Intel Adds Low-Power Features to Every i486*. Linley Gwennap, MPR vol. 7 no. 8, 6/21/93, pg. 1. (Cover story.)
- 47: *Continuing to Push the Limits of Integration*. Linley Gwennap, MPR vol. 7 no. 8, 6/21/93, pg. 3. (Editorial.)
- 48: *VLSI Integrates 486SL Power Management*. Linley Gwennap, MPR vol. 7 no. 9, 7/12/93, pg. 16. (Feature article.)



- 49: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (Cover story.)
- 50: *Intel Extends 486, Pentium Families*. Linley Gwennap, MPR vol. 8 no. 3, 3/7/94, pg. 1. (Cover story.)
- 51: *Intel Reveals Low-Power 486 Plans*. MPR vol. 8 no. 4, 3/28/94, pg. 4.
- 52: *Intel Matches AMD's 486SX2....* MPR vol. 8 no. 4, 3/28/94, pg. 5.
- 53: *Intel Offers \$250 Upgrade Chip*. MPR vol. 8 no. 5, 4/18/94, pg. 4.
- 54: *Intel Cuts Back on DX4, Pushes Pentium-60*. MPR vol. 8 no. 6, 5/9/94, pg. 5.
- 55: *Compaq Aero Uses 486SXJ*. MPR vol. 8 no. 6, 5/9/94, pg. 15.
- 56: *Intel Slashes Prices on Pentium, 486 DX2*. Linley Gwennap, MPR vol. 8 no. 9, 7/11/94, pg. 13. (Feature article.)
- 57: *Intel Slashes Pentium, 486 Prices*. MPR vol. 8 no. 14, 10/24/94, pg. 4. (Most Significant Bits item.)
- 58: *New Pentiums for Notebooks, 486 Upgrades*. Michael Slater, MPR vol. 8 no. 15, 11/14/94, pg. 14. (Feature article.)
- Other Technical References**
- 59: *Computer Architecture: A Quantitative Approach*. John Hennessy and David Patterson, Morgan Kaufmann Publishers, 1990, ISBN 1-55860-069-8. (The definitive textbook on modern computer architecture design methodologies.)
- 60: *Marketing High Technology*. William Davidow, Free Press, 1986. (Case histories of Intel marketing strategies.)
- Other Periodicals**
- 61: *Will the Pentium Kill the 486?*. Gina Smith, PC Computing, 5/93, pg. 116. (Cover story about Pentium features, design, performance, and system vendors.)
- 62: *Pentium Poised to Oust 486*. Neal Boudette, PCWeek, vol. 11 no. 26, 7/4/94, pg. 1.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



# 7

## AMD 386 and 486 Microprocessors

---

In April of 1991 Advanced Micro Devices (AMD) became the first non-Intel vendor to begin selling 386-class microprocessors. Since then, AMD has become remarkably successful as a full-service supplier, second-sourcing each of Intel's mainstream 386 and 486 processors with designs that provide better timing or electrical specifications than the original Intel devices and that introduce some all-new capabilities.

The AMD product line is significant from several perspectives. For OEM system designers, AMD's products broke the Intel monopoly and freed manufacturers from their dependence on a sole-source vendor. The enhanced characteristics of AMD 386 and 486 devices over earlier parts have also made possible a level of system performance not attainable with Intel-based designs.

For programmers, the extensions AMD made to the architecture of certain 386 products are a factor that must be considered in developing new BIOS and system software. For the financial community (and Intel!), the presence of AMD as a credible second source has hastened the introduction of new Intel products and caused prices to fall at a faster rate.

This chapter begins with an overview of AMD's company background, design methodology, and compatibility issues, and then describes each of the AMD 386 and 486 microprocessors, roughly in order of increasing processor capability.

### **Background**

In order to fully understand the AMD 386 and 486 product line and strategy, it helps to understand the company's history and

the long, convoluted story of its business relationship with Intel. Since its inception, AMD has had a reputation as a small, aggressive, and innovative vendor of high-quality microprocessors, peripherals, and programmable logic devices—both designing and selling its own proprietary products and acting as an alternate source of other vendors' products.

In a 1993 survey of subscribers to *Microprocessor Report*, AMD was rated the top microprocessor vendor in the industry, and was the only one of 21 vendors evaluated that ranked in the top quartile on all eight evaluation categories. (Intel, in contrast, was rated twelfth, and ranked in the top quartile only once, in the category “credibility of performance claims.”)

In 1976 Intel and AMD entered into an agreement that made AMD an approved alternate source of Intel microprocessors. The agreement granted AMD rights to Intel's microprocessor patents as well as to microcode used in Intel microcomputer and peripheral products.

AMD served as a licensed second source of the 8086, 80186, and 80286 processors; that is, Intel provided AMD with the design technology and databases (if not the complete photomasks) needed to build fully functional, fully compatible devices. This gave Intel products the benefit of increased visibility in the marketplace, and gave customers increased confidence that Intel's products would always be available in adequate supply and at competitive prices. In this pre-IBM PC era, success of the 8086 was far from assured, and by arranging for alternate sources Intel hoped to boost the part's prospects for success.

In 1982 Intel and AMD entered into a further agreement to cross-license microprocessor and peripheral designs for a period of 10 years. In principle, AMD would receive “credits” for developing peripherals and other support components, and “trade” said credits for the rights to build future Intel x86 processors. But by the time the 386 was introduced in 1985, the x86 architecture had become so well established that Intel no longer needed a second source, and refused to supply AMD with the 386 design information AMD had expected. Intel claimed the peripherals AMD had designed were not numerous or sophisticated enough for AMD to “deserve” the right to share in 386 production.

AMD fought Intel for several years over rights to the 386 design before concluding that whether or not the courts ultimately did award AMD the right to build Intel designs, it would be too late

to matter. Finally, in frustration, AMD decided to develop its own implementations of the Intel 386 and 486 families. (See **Chapter 16: Legal Issues** for further details on the ongoing legal feud between these two plaintiffs.)

## 7.1 Core Design

At the 386 and 486 levels, then, AMD has been forced to reverse-engineer Intel's microprocessor logic and extract the microcode in its design labs. While AMD's designs do not physically duplicate the layout of Intel's die, they are derived directly from the corresponding devices and have a functionally equivalent logic design.

### Design Methodology

AMD's design process begins by physically dissecting the silicon of an Intel chip. Each layer of the die is photographed to determine its layout geometry. A chemical or abrasive process then strips off the top layer of metal, silicon, or diffusion material to expose the next layer down, and the process is repeated until the engineers strike substrate.

By analyzing the patterns on successive layers, AMD can determine how the transistors in the original design are interconnected. Next, groups of closely coupled transistors are combined into gates, and a gate-level logic design is derived from the transistor-level schematic.

AMD performs the gate-level schematic extraction process twice to check for errors, and the resulting design is then extensively simulated. Computer analysis then locates the dynamic registers and other nodes within the Intel design that preclude the original parts from low-frequency operation; these dynamic gates and nodes are converted to static operation.

In the case of the 386-family products, converting the core to static operation added about 4,000 transistors to the Intel design, but made it possible for the parts to operate at arbitrarily low frequencies. The clock on certain 386- and 486-family products may be stopped entirely in order greatly to reduce power consumption when the system is in a standby or suspended state.

Finally, the design database is converted into an original device layout, following design rules for the targeted fabrication process. Thus, while device geometries may differ radically and

there may be no physical resemblance between parts, the derived versions are functionally equivalent to Intel's with respect to registers, ALUs, buses, control signals, etc.

The process of extracting the Intel microcode is more straightforward. Even though the courts have held that the microcode inside a microprocessor is protected by copyright, AMD maintains that the 1976 cross-licensing agreement gave it rights to Intel's microcomputer-related microcode; it therefore feels justified in photographically extracting the bit pattern from the Intel microROMs and dropping it intact into its own reconstructed designs.

The Intel devices from which AMD extracted its 486 designs were produced before Intel added the SL enhancements to the 486 product line. Thus, while some of the AMD parts allow static operation and support a "System Management Mode" analogous to that of Intel's SL-enhanced devices, the mechanisms by which they do so are different and incompatible. Moreover, the new instructions, new control signals, and other new functions included in the recent Intel devices are not currently supported by the AMD derivatives.

### **Compatibility and Performance**

Since the AMD chips are functionally equivalent to the Intel's designs and use the same microcode, the risk of functional compatibility problems or timing variations that might otherwise result from an independent reimplementations of the microcode was held to a minimum. AMD says its early silicon performed flawlessly on extensive tests, including software running under the DOS, Windows, and OS/2 operating systems, as well as DESQview, Xenix, and Phar Lap's DOS extender. In five months of testing by AMD's own engineers, an independent research lab, and more than 20 customers, no problems were found.

The redesigned devices thus appear to be compatible with Intel's with respect to instruction semantics, execution timing, pinouts, and electrical characteristics. AMD says they are "bug-for-bug compatible" with Intel's: there are no errata except for those on the errata list for the Intel parts from which the core logic design was extracted. Later steppings of AMD's chips were made merely to improve yield, AMD says, not to fix bugs.

Because of the similarity in logic design and the use of identical microcode, AMD devices should deliver exactly the same performance on user-mode software as the Intel devices. Device-level and system-level testing tend to confirm this, to within the range of testing error.

**Availability**

The latter half of this chapter describes a number of AMD 486-family devices, some with built-in floating-point units, some not, some with clock-doubling circuitry, some without. As is described in this chapter and in **Chapter 15: Manufacturing Costs**, many of these devices contain essentially the same die, and thus have about the same manufacturing cost.

As 1994 drew to a close, AMD found itself in the happy position of being unable to satisfy the growing demand for its 486-based products. The company's response was quite naturally to push sales of the higher-priced (and therefore higher-margin) variations, and actively de-emphasize sales of the stripped-down, low-margin products--to the point, even of refusing to quote prices for the lower-priced parts.

At this writing, AMD was in effect only willing to accept orders for the Am486DX2 and Am486DX4 versions of its products. Other, less capable devices such as the Am486SX have effectively been put on hold--not available, exactly, but not officially discontinued, either. As more fabrication capacity comes on line in 1995, perhaps this situation will change, and AMD will again find itself competing in the 486SX and 486DX arena.

## 7.2 The AMD Am386SX and Am386SXL Microprocessors

The Am386SX is AMD's pin-compatible second-source version of the Intel i386SX. The Am386SXL is a related device with improved specifications for low-power and battery-operated applications. Table 7-1 summarizes the general features and specifications of the Am386SX and Am386SXL microprocessors.

Product Names	AMD Am386SX and Am386SXL
Introduction Date	July 1991
Prognosis	Dormant
Device Integration Level	Same as i386SX
CPU Architecture Level	Same as i386SX
Core Technology	Design extracted from that of the i386SX, modified for static operation
Pinout	Same as i386SX
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as i386SX
Cache Support	None
Floating-Point Support	Optional external 387SX-class FPU
Operating Voltage	4.5 V to 5.5 V 40-MHz operation requires 4.75 V to 5.25 V
Frequency Options	25-, 33-, or 40-MHz core operation
Clocking Regime	Core operating frequency = CLK2 freq ÷ 2
Active Power Dissipation	1.475 W @ 5.0 V and 40 MHz (worst case)
Power-Control Features	Am386SX: None Am386SXL: Allows low-freq and stopped-clock operation (Iccsb < 150 µA @ 0.0 MHz)
Process Technology	0.8µ two-layer-metal CMOS
Die Size	74,000 mils <sup>2</sup> (47.7 mm <sup>2</sup> )
Transistor Count	161,000 actual transistors (approximately 279,000 total transistor sites)
Package Options	100-pin PQFP
Other Features	Am386SXL allows stopped-clock operation
Notes	Both devices contain the same silicon and have the same price; only the minimum frequency specs and testing procedure differ

Table 7-1. AMD Am386SX/Am386SXL feature summary.

**Features** Both devices are fully compatible with the Intel part with respect to functions, software, system interface, timing, and



electrical specs, but are available at frequencies up to 40 MHz. AMD says yields at 40 MHz are very good.

The Am386SXL has lower power-consumption specifications and has no minimum clock-frequency specification. Both devices contain the same silicon die and generally carry the same price tag; while it is possible that AMD may have slightly higher yields on the non-SXL versions and may save some test time by not having to verify low clock-frequency operation, the distinction between the two devices is primarily one of marketing.

### System Interface

The Am386SX and Am386SXL provide the same system interface as the i386SX with the same pin functions, signal names, package types, and pinouts as the original Intel parts; indeed, one of the “Distinctive Characteristics” listed for the parts on the first page of the AMD data sheet is that they are a “pin-for-pin” replacement for the Intel i386SX. (See **Chapter 5: Intel 386 Microprocessors** for details.) Timing and electrical specifications are likewise identical to those of the i386SX, so naïve system manufacturers and purchasing agents can specify it as an exact functional replacement for the Intel parts.

Even for the 40-MHz parts, the only specifications that differ from Intel’s 33-MHz figures are the minimum CLK2 period (12.5 ns) and CLK2 high and low times (5 ns at a 2-V threshold or 3.25 ns at 0.8 V/3.7 V). Setup and hold times are unchanged, making bus timing at the higher frequency very tight. While board design gets tricky at higher clock rates, the task of pushing a design to 40 MHz has been considerably simplified by the introduction of 40-MHz chip sets.

### Vital Statistics

The Am386SX and Am386SXL are fabricated using AMD’s “CS21S” process, a linear shrink of the “CS21” process with a resulting minimum drawn feature size of 0.8 microns and an average feature size of 0.9 microns. The devices each contain about 160,000 transistors (by AMD’s count) on a 74,000 mil<sup>2</sup> die. Intel’s dynamic 386 design uses about 4,000 fewer transistors and occupies 66,000 mil<sup>2</sup> in the 1.0-micron CHMOS-IV version. (Curiously, Intel says the 386 core contains 275,000 transistors, not 156,000. Intel counts all possible transistor sites, including every bit position within microcode ROMs and every signal crosspoint within instruction decoders and control PLAs.)

Because it uses smaller process geometry, the Am386SXL uses less power than the i386SX at the same frequency. At its 33-

MHz maximum clock rate, the i386SX consumes 550 mA (worst case), whereas the Am386SXL is spec'd at 395 mA. With a minimum-speed 4-MHz internal clock, the i386SX consumes 133 mA. In contrast, standby current for the Am386SXL with its clock stopped is guaranteed to be under 150  $\mu$ A.

The parts are supplied in the same 100-lead PQFP package as the i386SX. Both are offered at frequencies of 25 or 33 MHz, the same as the Intel parts, as well as 40 MHz, filling a niche that Intel chose to forgo.

## 7.3 The AMD Am386SXLV Microprocessor

The AMD Am386SXLV is a low-voltage, lower-power version of the Am386SXL device. Table 7-2 summarizes the general features and specifications of the Am386SXLV microprocessor.

Product Name	AMD Am386SXLV
Introduction Date	October 1991
Prognosis	Terminated
Device Integration Level	Same as i386SX with AMD SMM circuitry
CPU Architecture Level	Same as i386SX with AMD SMM extensions
Core Technology	Standard AMD 386 core
Pinout	Enhanced i386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as i386SX
Cache Support	None
Floating-Point Support	Optional external 387SX-class FPU
Operating Voltage	3.0 V to 5.5 V
Frequency Options	25-MHz core operation
Clocking Regime	Core operating frequency = CLK2 freq ÷ 2
Active Power Dissipation	412 mW @ 3.3 V and 25 MHz (worst case)
Power-Control Features	Allows low-freq and stopped-clock operation; includes AMD SMM extensions
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	74,000 mils <sup>2</sup> (47.7 mm <sup>2</sup> )
Transistor Count	161,000 actual transistors (approximately 279,000 total transistor sites)
Package Options	100-pin PQFP
Notes	Iccsb @ 3.3 V, 0 MHz = 10 $\mu$ A (typical), Iccsb @ 3.3 V, 0 MHz < 150 $\mu$ A (worst case)

Table 7-2. AMD Am386SXLV feature summary.

### Architecture Extensions

The Am386SXLV extends the original 386 architecture to include a new operating mode known as “system management mode” (SMM). While SMM is active, a special memory space is enabled that is separate from the conventional system memory and cannot be accessed by conventional system- or user-mode software. Support for SMM software includes the three new instructions listed in Table 7-3.

Instruction	Mode	Operation	Opcode
SMI	System	Invoke SMM interrupt routine	F1H
UMOV dest,src	SMM only	Move source to destination with SMM memory space active	0F10H– 0F13H
RES4	SMM only	Resume normal execution	0F07H

Table 7-3. AMD Am386SXLV new instructions.

The SMI instruction allows software directly to invoke the system management interrupt service routine. All of the user-visible registers are automatically stored into a special region of the protected system management memory space, and CPU operation enters system management mode.

The UMOV instruction allows an 8-, 16-, or 32-bit operand to be loaded from or stored to the protected system management memory space, rather than conventional system memory.

The RES4 instruction is executed at the end of the SMI service routine. The CPU state variables saved upon entering the SMI routine are retrieved from system management memory and execution resumes in the normal operating mode.

## System Interface

The Am386SXLV system interface is based on that of the Am386SX, with the addition of four signals that control the system management mode protocol. Figure 7-1 shows the Am386SXLV system interface.

The names and functions of Am386SXLV signals that differ from those of the standard i386SX system interface are summarized in Table 7-4.

Symbol	Direction	Signal Name/Function	PQFP Pin	i386SX Signal
SMI#	I/O	System management interrupt request	43	N.C.
SMIADS#	Out	System management interrupt address status	31	N.C.
SMIRDY#	In	System management interrupt transfer ready	30	N.C.
IIBEN#	In	I/O instruction break enable	29	N.C.

Table 7-4. AMD Am386SXLV interface signals.

External logic asserts the SMI# signal to invoke a system management interrupt. Once system management mode has been entered, the Am386SXLV continues to drive SMI# low until normal execution is resumed.

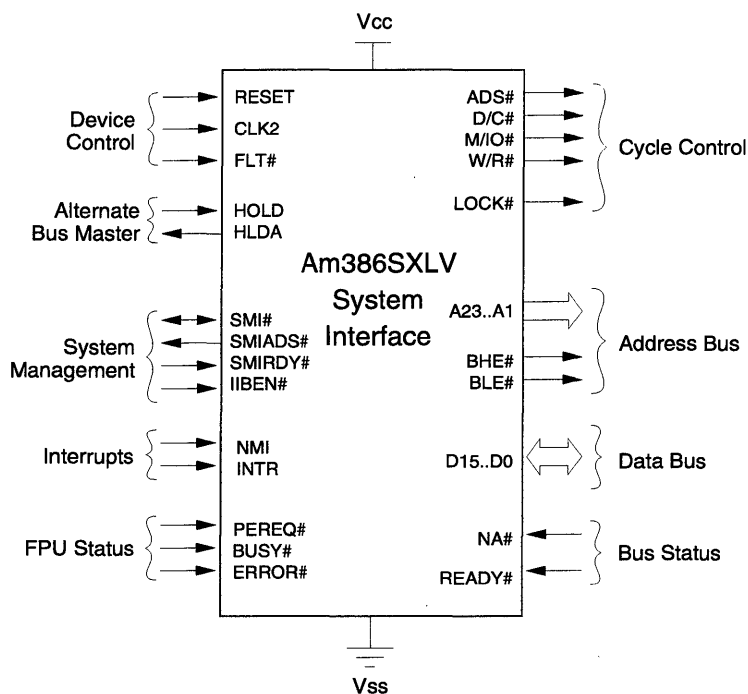


Figure 7-1. AMD Am386SXLV system interface.

The Am386SXLV asserts the SMIADS# output signal to indicate the start of a memory cycle that should access protected SMM memory rather than conventional system memory.

External logic then asserts the SMIRDY# input signal to indicate the completion of each system management memory cycle.

External logic may assert the IIBEN# input signal to indicate to the Am386SXLV that I/O cycles are interruptible. If IIBEN# is active and SMI# is asserted during an I/O read or write instruction, the I/O operation will be aborted and the SMM service routine will be invoked. Upon completion of the SMI service routine, the I/O instruction that had been interrupted will be restarted.

### Vital Statistics

The Am386SXLV is fabricated using the same 0.8-micron process as the Am386SX/Am386SXLV. The device also contains about 161,000 transistors by AMD's count on a 74,000 mil<sup>2</sup> die. The parts are supplied in the same 100-lead PQFP package as the i386SX. It operates at frequencies up to 25 MHz and can operate on power supply voltages between 3.0 V and 5.5 V.

## 7.4 The AMD Am386DX and Am386DXL Microprocessors

The Am386DX is AMD's second-source version of the i386DX. The Am386DXL is an enhanced version of the same device. Table 7-5 summarizes the general features and specifications of the Am386DX and Am386DXL microprocessors.

Product Name	AMD Am386DX and Am386DXL
Introduction Date	March 1991
Prognosis	Dormant
Device Integration Level	Same as i386DX
CPU Architecture Level	Same as i386DX
Core Technology	Standard AMD 386 core
Pinout	Same as i386DX
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4 GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i386DX
Cache Support	None
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	4.5 V to 5.5 V (4.75 V to 5.25 V for 40-MHz operation)
Frequency Options	33- or 40-MHz core operation
Clocking Regime	Core operating frequency = CLK2 freq ÷ 2
Active Power Dissipation	2.0 W @ 5.0 V and 40 MHz (worst case)
Power-Control Features	Am386DX: None Am386DXL: Allows low-freq and stopped-clock operation; includes AMD SMM features (Iccsb < 150 $\mu$ A @ 5.0 V and 0.0 MHz)
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	74,000 mils <sup>2</sup> (47.7 mm <sup>2</sup> )
Transistor Count	161,000 actual transistors (approximately 279,000 total transistor sites)
Package Options	132-pin PGA or 132-lead PQFP package
Other Features	Am386DXL allows stopped-clock operation
Notes	The Am386DX and Am386DXL both contain the same silicon; only the minimum frequency specs and testing procedure differ

Table 7-5. AMD Am386DX and Am386DXL feature summary.

### System Interface

The Am386DX and Am386DXL are fully compatible with the Intel i386DX with respect to pin functions, timing, and electrical characteristics, and likewise provide the same system inter-

face, pin functions, signal names, package types, and pinouts as the original i386DX.

The one deviation is the FLT# input signal, which disables all output pins to simplify post-assembly board-level testing. AMD supports FLT# on both the PGA and PQFP packages. On the Intel i386DX PGA package, the corresponding pin is a no-connect. An internal pull-up resistor in the AMD parts pulls this signal to its inactive state in order to assure i386DX PGA socket interchangeability.

### **Vital Statistics**

The AMD Am386DX and Am386DXL devices each contain about 160,000 transistors (by AMD's count) on a 74,000 mil<sup>2</sup> die. AMD offers the Am386DX at the same 33-MHz clock rate as the Intel i386DX, as well as at 40 MHz, filling a niche that Intel chose to forgo. The Am386DXL is offered at these same frequencies. Power consumption for the "DX" devices matches the AMD "SX" parts described above.

Each of the devices is offered in a 132-pin ceramic pin-grid-array (PGA) package as well as in a 132-pin plastic quad flat-pack (PQFP). See **Chapter 5: Intel 386 Microprocessors** for details on pin functions and pinout assignments.

## 7.5 The AMD Am386DXLV Microprocessor

The AMD Am386DXLV microprocessor is a variation of the Am386DXL intended for low-power, low-voltage applications. Table 7-6 summarizes the general features and specifications of the Am386DXLV microprocessor.

Product Name	AMD Am386DXLV
Introduction Date	October 1991
Prognosis	Terminated
Device Integration Level	Same as i386DX
CPU Architecture Level	Same as i386DX
Core Technology	Standard AMD 386 core
Pinout	Extended i386DX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as i386DX
Cache Support	None
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	3.0 V to 5.5 V
Frequency Options	25-MHz core frequency @ 3.0 V–3.6 V; 25- or 33-MHz core frequency @ 4.5 V–5.5 V
Clocking Regime	Core operating frequency = CLK2 freq ÷ 2
Active Power Dissipation	1.65 W @ 5.0 V and 33 MHz (worst case) 445 mW @ 3.3 V and 25 MHz (worst case)
Power-Control Features	Allows low-freq and stopped-clock operation; includes AMD SMM extensions
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	74,000 mils <sup>2</sup> (47.7 mm <sup>2</sup> )
Transistor Count	161,000 actual transistors (279,000 total transistor sites)
Package Options	100-pin PQFP
Other Features	Includes AMD SMM H/W and S/W features Provides IEEE/JTAG boundary scan test port
Notes	Iccsb @ 3.3 V, 0 MHz = 10 $\mu$ A (typical), Iccsb @ 3.3 V, 0 MHz < 150 $\mu$ A (worst case)

Table 7-6. AMD Am386DXLV feature summary.

**Features** The AMD Am386DXLV microprocessor is to the Am386DX and Am386DXL essentially what the Am386SXLV is to the Am386SX and Am386SXL. It implements the same SMM architecture extensions and executes the same new instructions as the Am386SXLV. Its system interface matches that of the



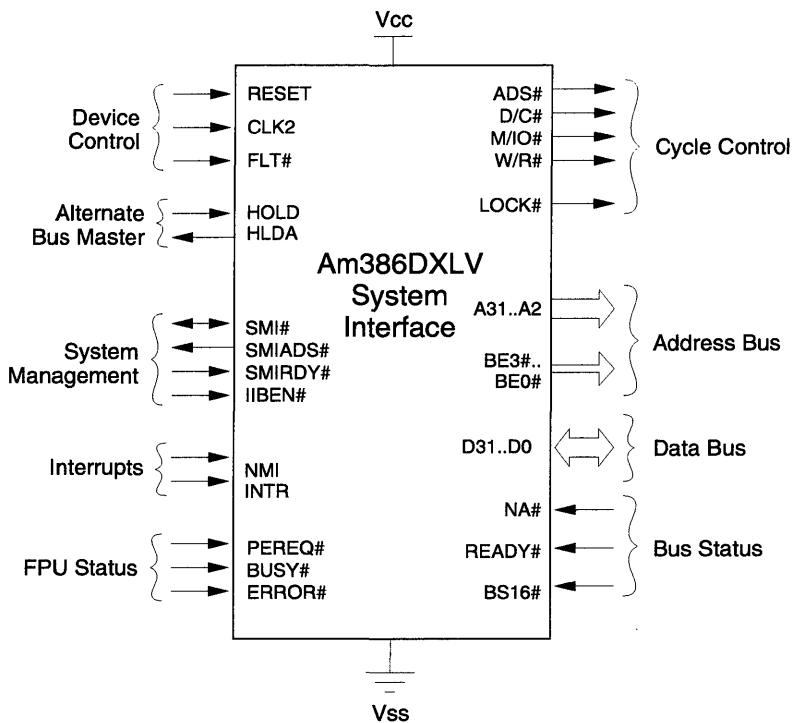


Figure 7-2. AMD Am386DXLV system interface.

Am386DX, with the addition of the same new interface signals as the Am386SXLV, as shown in Figure 7-2. See the description of the Am386SXLV earlier in this chapter for details on the architecture extensions and power-management interface signals.

### Vital Statistics

The Am386DXLV is fabricated using the same 0.8-micron process as the rest of the AMD 386 product line. The device contains about 161,000 transistors (by AMD's count) on a 74,000 mil<sup>2</sup> die. It is offered at frequencies of 25 or 33 MHz. The 25-MHz version allows Vcc to range from 3.0 V to 5.5 V. The 33-MHz version requires Vcc to be between 4.5 V and 5.5 V. The device is offered in a 132-pin PGA or 132-lead PQFP package with the same pinout as the i386DX.

## 7.6 The AMD Am386SC300 “Elan” Microprocessor

The Am386SC300 is AMD's entry in the high-integration CPU chip-set sweepstakes, and a spiritual successor to the Intel i386SL and i486SL. This device combines a 386 CPU with the system logic and I/O devices needed for a standard DOS or Windows run-time environment, all on a single die. Table 7-7 summarizes the general features and specifications of the Am386SC microprocessor.

Product Name	AMD Am386SC300 “Elan”
Introduction Date	October 1993
Prognosis	Sampling
Device Integration Level	Am386SXLV core plus ISA system-integration logic
CPU Architecture Level	Same as Am386SXLV
Core Technology	Adapted from design derived from Intel 386 core
Pinout	Custom
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16 MB (24-bit address multiplexed on 11 pins)
Data-Transfer Modes	Similar to i386SX plus direct DRAM interface
Cache Support	None
Floating-Point Unit	None
Operating Voltage	3.3 V $\pm$ 5%
Frequency Options	25-MHz core operation
Clocking Regime	Configured via power-management logic
Active Power Dissipation	0.48 W @ 3.3 V and 25 MHz (worst case)
Power-Control Features	Static
Process Technology	0.9 $\mu$ two-layer-metal CMOS
Die Size	(122 mm <sup>2</sup> )
Transistor Count	473,700 actual transistors
Package Options	208-lead PQFP
Other Features	Also contains ISA bus-control logic, DRAM memory controller, DMA controller, interrupt controller, timers, serial and parallel ports, LCD graphics interface, PCMCIA interface, real-time clock, and power-management circuitry
Notes	Same die as Am486DX

Table 7-7. AMD Am386SC300 “Elan” feature summary.

The Am386SC300 (hereafter called the Am386SC) is the first member of a family of products collectively known as “Elan.” The part is designed for hand-held computers, the vast majority of which are currently sold into vertical markets. Such systems

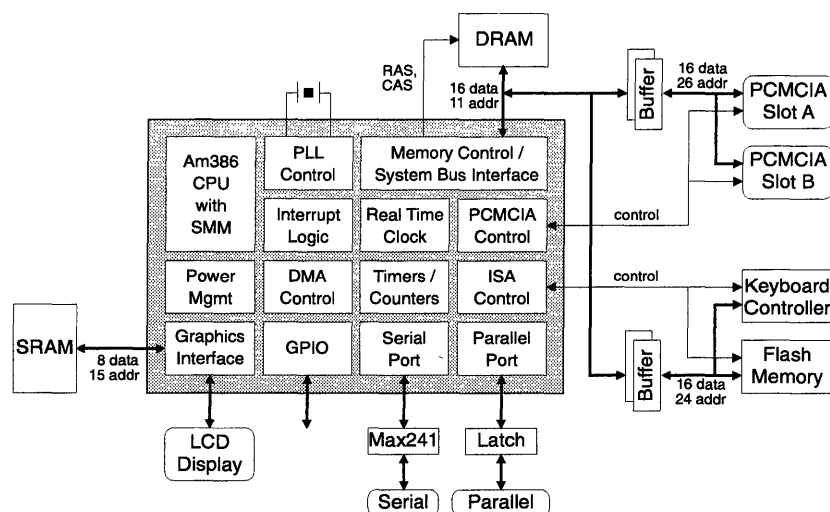


Figure 7-3. AMD Am386SC300 "Elan" Block diagram and system interface.

typically are DOS- or Windows-based; the Am386SC can also run the GeoWorks OS. If Microsoft's WinPad had emerged on schedule in 1994, the Am386SC would have been a platform for that operating system. With WinPad's redefinition and delay until 1996, however, this OS will require a 486-based successor to the Am386SC. The peripheral integration lessons AMD has learned in designing the Am386SC should be applicable to future designs as well.

**Features** The Am386SC core is derived from the Am386SXLV CPU. The CPU core operates at 3.3 V, and its I/O circuitry can connect to either 3.3-V or 5-V peripherals. The static processor core can operate at any clock speed from D.C. to 25 MHz and retains its state when the clocks are stopped. (Curiously, the Am386SXLV from which the part is derived is specified for operation up to 33 MHz.)

As shown in Figure 7-3, the Am386SC includes a memory controller, real-time clock, and the equivalent of an 82C206 for DOS compatibility. Some of this logic was licensed from a third party, rumored to be Taiwanese chip-set vendor Tidalwave. The chip does not support an on-chip or an external cache, but 70-ns DRAMs provide zero wait states at 33 MHz, reducing the need for a cache.

The memory controller supports a 16-bit data path to DRAM or SRAM with no external buffers required. While this memory

width will offer performance comparable to a typical 386SX, the lack of a 32-bit memory path will leave the Am386SC unable to match Am386DX performance levels. The new chip can handle up to 16M of memory in two banks; a low-end design can use a single 1M × 16 DRAM for a 2MB memory system.

A built-in power-management unit (PMU) implements a variety of features to reduce power consumption and extend battery life. The PMU monitors system activity and automatically switches operation between various power-reduction modes. If a specified duration (set by configuration software) elapses without any system activity, the PMU reduces the CPU clock speed to 2–18 MHz to save power.

After a longer interval, the PMU stops the CPU clock entirely and slows the clocks to the peripherals. Later, the processor can completely shut down after peripheral state is saved in memory. If the PMU detects new system activity, it can return the processor to a full-speed mode. The PMU can also be configured to trigger a system-management interrupt when the processor shifts to a new power mode.

### **System Interface**

The Am386SC provides a set of ISA control signals for adding functions using standard ISA peripheral chips. As with PCMCIA devices, ISA devices are connected through buffers. On-chip logic decodes addresses and generates device-select signals for a keyboard controller and a non-volatile memory device such as Flash EPROMs.

Two standard I/O interfaces are included: a bidirectional parallel port and a serial port, both compatible with DOS standards. The parallel port requires only a single external component between the Am386SC and the connector. Similarly, a simple buffer chip interfaces the processor to the serial connector. Alternatively, the serial port can connect to a digitizer or a modem.

The Am386SC also supports two PCMCIA 2.0 slots. Most handheld devices today implement one or two PCMCIA slots for add-in memory and/or peripheral cards. By including control for these devices on-chip, the Am386SC eliminates the need for external PCMCIA interfaces. These slots are driven from the memory address and memory data buses through logic buffers. Additional voltage buffers are needed for hot insertion of add-in cards, but many portable devices rely on a physical interlock instead.

And finally, the device includes an LCD controller that is 6845-compatible and supports panels up to  $640 \times 400$  pixels. It provides CGA emulation for DOS compatibility. An external SRAM contains the graphics frame buffer. This memory is connected internally to the processor local bus, permitting fast data transfers. For systems that require higher graphics performance, the frame buffer can be replaced by an external graphics-accelerator chip; in this configuration, the Am386SC provides a 16-bit local bus interface.

## 7.7 The AMD Am486SX and Am486SX2 Microprocessors

The Am486SX and Am486SX2 are AMD's second-source implementations of the Intel i486SX and i486SX2. Table 7-8 summarizes the general features and specifications of the Am486SX and Am486SX2 microprocessors.

Product Names	AMD Am486SX and Am486SX2
Introduction Dates	Am486SX: July 1993 Am486SX2: February 1994
Prognoses	Am486SX: Deceased Am486SX2: Healthy
Device Integration Levels	Same as i486SX and i486SX2
CPU Architecture Level	Same as i486SX and i486SX2
Core Technology	Same as the Am486DX; extracted from the i486DX
Pinout	Enhanced standard i486SX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486SX
Cache Support	Same as i486SX
Floating-Point Support	None; requires external upgrade processor
Operating Voltage	4.75 V to 5.25 V
Frequency Options	Am486SX: 33- or 40-MHz core operation Am486SX2: 50- or 66-MHz core operation
Clocking Regime	Am486SX2: Core operating freq = CLK input $\times$ 1
Active Power Dissipation (worst case)	Am486SX: 4.25 W @ 5.0 V and 40 MHz Am486SX2: 4.5 W @ 5.0 V and 66 MHz core freq
Power-Control Features	None
Process Technology	0.7 $\mu$ three-layer-metal CMOS
Die Size	360 x 384 mils (89 mm <sup>2</sup> )
Transistor Count	Approximately 930,000 actual transistors
Package Options	168-pin PGA
Other Features	Provides IEEE/JTAG boundary-scan test port
Notes	Contains same die as Am486DX (described below) Both Intel and AMD $\mu$ code versions produced

Table 7-8. AMD Am486SX and Am486SX2 feature summary.

The Am486SX2 includes a clock-doubler circuit like that described for the i486SX2 and i486DX2 in the previous chapter. As the prices and profit margins of low-end 486 devices fell during 1994, the former device was discontinued and supplanted by the latter, clock-doubled version.

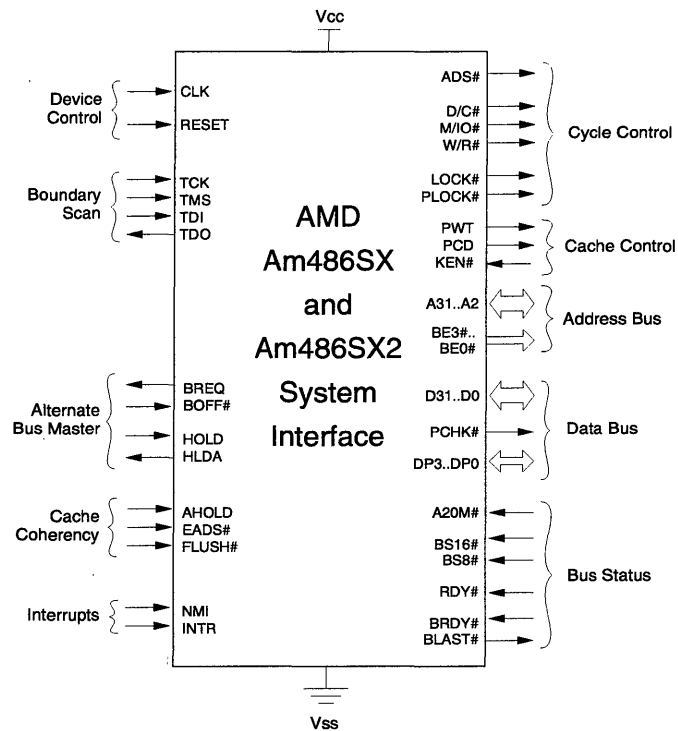


Figure 7-4. AMD Am486SX and Am486SX2 system interface.

**System Interface**

The Am486SX2 is fully compatible with the i486SX2 with respect to hardware capabilities, system interface functions, timing, and electrical specs. In addition, the Am486SX2 PGA device includes the UP# (Upgrade Processor present) and JTAG interface signals, which, in the case of the Intel PGA-packaged devices, are supported only by SL-enhanced variations.

The Am486SX2 system interface is shown in Figure 7-4. Table 7-9 lists the names and functions of Am486SX2 signals not supported by the original i486SX device. See **Chapter 6: Intel 486 Microprocessors** for details.

Symbol	Direction	Signal Name/Function	PGA Pin	i486SX Signal
TCK	In	JTAG boundary scan clock	A3	N.C.
TMS	In	JTAG boundary scan mode select	B14	N.C.
TDI	In	JTAG boundary scan data in	A14	N.C.
TDO	Out	JTAG boundary scan data out	B16	N.C.

Table 7-9. AMD Am486SX interface signals.

**Vital Statistics** Since the AMD chips are derived from Intel's logic design, the performance of the respective parts should be equivalent at a given clock rate. In the case of parts that contain Intel microcode, the AMD devices should match Intel's on a state-for-state basis. Versions that contain AMD-developed microcode may differ slightly in certain instruction sequences, but any such discrepancies should be minor.

The Am486SX2 is fabricated in a 0.7-micron, three-level-metal process. Chip size is 138K mil<sup>2</sup> (89 mm<sup>2</sup>)—considerably larger than Intel's implementation, which is 72 mm<sup>2</sup>, in part because Intel redesigned its chip to actually omit the FPU, and AMD has not.

Power consumption for the 33-MHz Am486SX at 5.0 V is 600 mA typical (700 mA maximum), about 100 mA less than the specification for the i486SX-33 because the Intel chip parameters were characterized from a 1.0-micron process. Power consumption at 40 MHz and 5.0 V is 700 mA typical, 850 mA maximum. The Am486SX is supplied in the same 168-pin PGA package as the i486SX.



## 7.8 The AMD Am486SXLV Microprocessor

The Am486SXLV is an enhanced version of the Am486SX that includes hardware and software features to facilitate low-power operation in battery-powered applications. Table 7-10 summarizes the general features and specifications of the Am486SXLV microprocessor.

Product Name	AMD Am486SXLV
Introduction Date	July 1993
Prognosis	Terminated
Device Integration Level	Same as i486SX
CPU Architecture Level	Same as i486SX
Core Technology	Standard AMD 486 core
Pinout	Enhanced standard i486SX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486SX
Cache Support	Same as i486SX
Floating-Point Support	None; requires external upgrade processor
Operating Voltage	3.0 V to 3.6 V
Frequency Options	33-MHz core operation
Clocking Regime	Core operating frequency = CLK input $\times$ 1
Active Power Dissipation	1.4 W @3.3 V and 33 MHz (worst case)
Power-Control Features	Allows low-freq and stopped-clock operation
Process Technology	0.7 $\mu$ three-layer-metal CMOS
Die Size	360 x 384 mils (89 mm <sup>2</sup> )
Transistor Count	Approximately 930,000 actual transistors
Package Options	196-lead PQFP
Other Features	Includes AMD SMM H/W and S/W features Provides IEEE/JTAG boundary-scan test port
Notes	Contains the same die as the Am486SX

Table 7-10. AMD Am486SXLV feature summary.

The Am486SXLV requires a 3.3 V power supply, allows static operation, and supports AMD's system management mode for power management. The part is very similar in concept to Intel's SL-enhanced i486SX. The primary difference is that AMD's SMM differs from Intel's in the details of its operation.

### Architecture Extensions

The Am486SXLV supports the same SMM extensions to the 486 architecture as the Am386SXLV and Am386DXLV. Table 7-11 shows two new SMM instructions.

These instructions perform the same functions as the Am386SXLV, described earlier in this chapter. Note that the Am486SXLV does not support the SMI software interrupt instruction defined by the AMD 386 family.

Instruction	Mode	Operation	Opcode
UMOV dest,src	SMM only	Move source to destination with SMM memory space active	0F10H–0F13H
RES4	SMM only	Resume normal execution	0F07H

Table 7-11. AMD Am486SXLV new instructions.

## System Interface

The Am486SXLV system interface is shown in Figure 7-5. The system interface is derived from that of the i486SX, with the addition of the JTAG boundary-scan test port and system management control signals.

The names and functions of Am486SXLV signals that differ from those of the standard i486SX pinout are summarized in Table 7-12. Note that the Am486SXLV does not implement the IBEN# signal defined by the AMD 386 family.

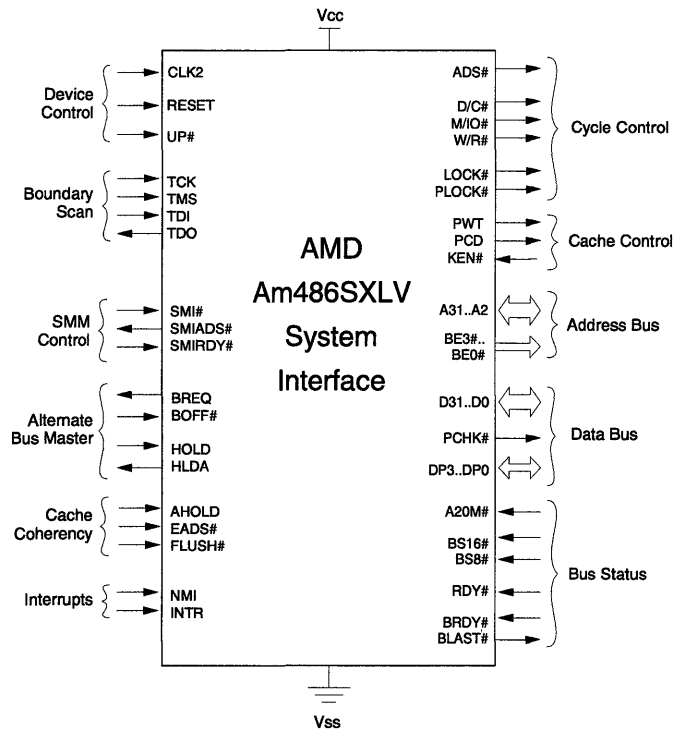


Figure 7-5. AMD Am486SXLV system interface.

Symbol	Direction	Signal Name/Function	PQFP Pin	i486SX Signal
CLK2	In	Internal clock frequency $\times 2$	123	CLK
TCK	In	JTAG boundary-scan clock	128	TCK
TMS	In	JTAG boundary scan mode select	187	TMS
TDI	In	JTAG boundary scan data in	185	TDI
TDO	Out	JTAG boundary scan data out	80	TDO
SMI#	I/O	System management interrupt request	82	N.C.
SMIADS#	Out	SMI address strobe	140	N.C.
SMIRDY#	In	SMI transfer ready	134	N.C.

Table 7-12. AMD Am486SXLV interface signals.

**Vital Statistics**

The Am486SXLV is fabricated using the same 0.7-micron process as the Am486SX. The device is supplied in the same 196-lead PQFP package as the i486SX. Operation is specified to a maximum frequency of 33 MHz at 3.3 V.

## 7.9 The AMD Am486DX Microprocessor

The Am486DX is AMD's designation for a second-source implementation of the i486DX. Table 7-13 summarizes the general features and specifications of the Am486DX microprocessor.

Product Name	AMD Am486DX
Introduction Date	May 1993
Prognosis	Discontinued
Device Integration Level	Same as i486DX
CPU Architecture Level	Same as i486DX
Core Technology	Standard AMD 486 core
Pinout	Enhanced standard i486DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	4.5 V to 5.5 V
Frequency Options	40-MHz core operation
Clocking Regime	Core operating frequency = CLK input × 1
Active Power Dissipation	4.25 W @ 5.0 V and 40 MHz (worst case)
Power-Control Features	None
Process Technology	0.7 $\mu$ three-layer-metal CMOS
Die Size	360 x 384 mils (89 mm <sup>2</sup> )
Transistor Count	Approximately 930,000 actual transistors
Package Options	168-pin PGA
Other Features	Provides IEEE/JTAG boundary-scan test port
Notes	Same die as Am486SX with FPU enabled

Table 7-13. AMD Am486DX feature summary.

The Am486DX provides the same system-interface signals, package types, and pinout as the original i486DX. It is fully compatible with Intel's specifications with respect to software compatibility and electrical characteristics.

### Vital Statistics

The Am486DX is supplied in a 168-pin PGA package at frequencies of 33 or 40 MHz.

## 7.10 The AMD Am486DX2 Microprocessor

The Am486DX2 is AMD's second-source implementation of the i486DX2. Table 7-7 summarizes the general features and specifications of the Am486DX2 microprocessor.

Product Name	AMD Am486DX2
Introduction Date	May 1993
Prognosis	Thriving
Device Integration Level	Same as i486DX
CPU Architecture Level	Same as i486DX
Core Technology	Standard AMD 486 core
Pinout	Enhanced standard i486DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DXr
Operating Voltage	4.75 V to 5.25 V
Frequency Options	50-, 66-, or 80-MHz core operation
Clocking Regime	Core operating frequency = CLK input $\times$ 2 Bus interface frequency = CLK input $\times$ 1
Active Power Dissipation	7.5 W @ 5.0 V and 80 MHz (worst case)
Power-Control Features	None
Process Technology	0.7 $\mu$ three-layer-metal CMOS
Die Size	360 x 384 mils (89 mm <sup>2</sup> )
Transistor Count	Approximately 930,000 actual transistors
Package Options	168-pin PGA
Other Features	Provides IEEE/JTAG boundary-scan test port
Notes	Same die as Am486DX

Table 7-14. AMD Am486DX2 feature summary.

**Features** The Am486DX2 provides the same clock-doubler circuit, system-interface signals, package types, and pinout as the original i486DX2, with the addition of the JTAG boundary-scan test port described for the Am486SX device above. It is fully compatible with the i486DX2 in its hardware capabilities, system-interface functions, and electrical characteristics.

**Vital Statistics** The Am486DX2 is supplied in a 168-pin PGA package with internal core frequencies of 50 or 66 MHz.

## 7.11 The AMD Am486DXL and Am486DXLV Microprocessors

The AMD Am486DXL is a variation of the Am486DX with improved specifications for power consumption and electrical characteristics for battery-powered systems. Its pinout is upwardly compatible with that of a standard i486DX. The AMD Am486DXLV is a lower-voltage and even lower power variation of the Am486DXL. Table 7-15 summarizes the general features and specifications of the Am486DXLV microprocessor.

Product Name	AMD Am486DXL and Am486DXLV
Introduction Date	May 1993
Prognosis	Am486DXL: Terminated Am486DXLV: Terminated
Device Integration Level	Same as i486DX
CPU Architecture Level	Same as i486DX with AMD SMM extensions
Core Technology	Standard AMD 486 core
Pinout	Enhanced standard i486DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX
Cache Support	Same as i486DX
Floating-Point Support	Same as i486DX
Operating Voltage	Am486DXL: 4.5 V to 5.5 V Am486DXLV: 3.0 V to 3.6 V
Frequency Options	Am486DXL: 33- or 40-MHz core frequency Am486DXLV: 33-MHz core frequency
Clocking Regime	Core operating frequency = CLK input $\times$ 1
Active Power Dissipation	Am486DXL: N.A. Am486DXLV: 1.4 W @ 3.3 V and 33 MHz (w.c.)
Power-Control Features	Allows low-frequency and stopped-clock operation
Process Technology	0.7 $\mu$ three-layer-metal CMOS
Die Size	360 x 384 mils (89 mm <sup>2</sup> )
Transistor Count	Approximately 930,000 actual transistors
Package Options	196-pin PQFP
Other Features	Provides IEEE/JTAG boundary-scan test port
Notes	Same die as Am486SXLV with FPU enabled

Table 7-15. AMD Am486DXL and Am486DXLV feature summary.

The Am486DXL requires a 5.0 V power supply, allows static operation, and supports AMD's SMM for power management. The Am486DXLV has the same capabilities but requires only a

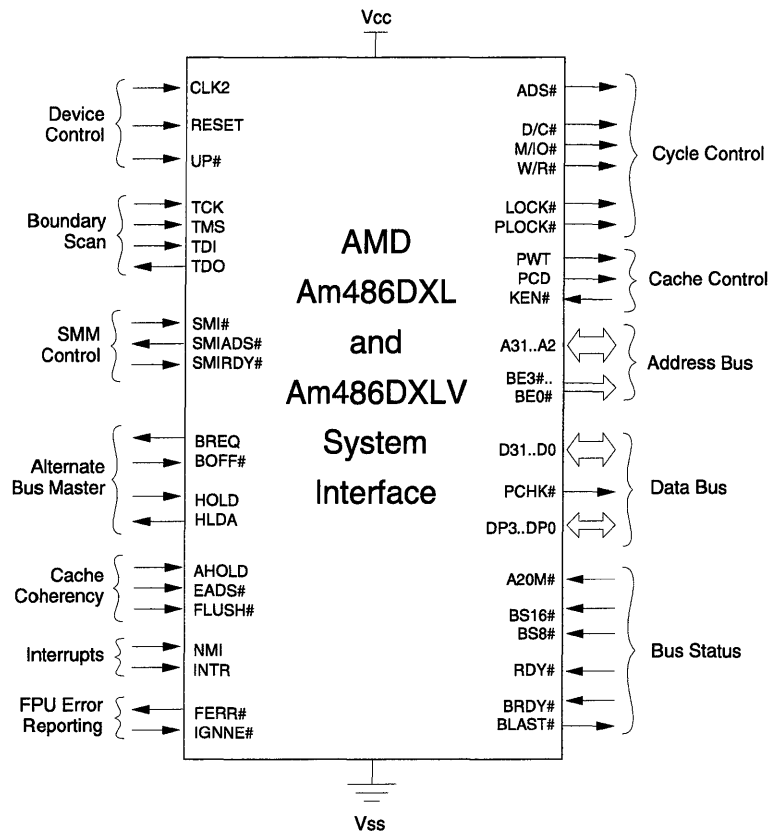


Figure 7-6. AMD Am486DXL/Am486DXLV system interface.

3.3 V power supply. The part is very similar in concept to Intel's SL-enhanced i486DX. The primary difference between AMD's SMM and Intel's lies in the details of its operation.

### System Interface

The system interface of the Am486DXL and Am486DXLV is based on that of the Am486DX, with the addition of the system management control signals defined for the Am486SXLV. Figure 7-6 shows the Am486DXL/Am486DXLV system interface.

### Vital Statistics

The Am486DXL and Am486DXLV are fabricated using the same 0.7-micron process as the other members of the AMD 486 product line. The devices are supplied in a 196-lead PQFP package. The Am486DXL requires a 5.0 V power supply and has a maximum frequency of 40 MHz. The Am486DXLV allows 3.3-V operation but has a top frequency of 33 MHz.

## 7.12 The AMD Am486DX4 Microprocessor

The Am486DX4 is AMD's answer to the IntelDX4. Table 7-7 summarizes the general features and specifications of the Am486DX4 microprocessor.

Product Name	AMD Am486DX4
Introduction Date	October 1994
Prognosis	Promising
Device Integration Level	Same as Am486DX2
CPU Architecture Level	Same as Am486DX2
Core Technology	Derived from Intel 486 core
Pinout	Enhanced standard i486DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX
Cache Support	Same as Am486DX2 (8KB on-chip cache)
Floating-Point Support	Same as Am486DX2
Operating Voltage	3.3 V $\pm$ 5%
Frequency Options	Up to 100-MHz core operation
Clocking Regime	Core operating frequency = CLK input $\times$ 3 Bus interface frequency = CLK input $\times$ 1
Active Power Dissipation	3.3 W @ 3.3 V and 100 MHz (worst case)
Power-Control Features	Static operation with stop-clock input
Process Technology	0.5 $\mu$ three-layer-metal CMOS
Die Size	56 mm <sup>2</sup>
Transistor Count	Approximately 938,000 actual transistors
Package Options	168-pin PGA
Other Features	Provides IEEE/JTAG boundary-scan test port
Notes	Derived from same die as Am486DX2

Table 7-16. AMD Am486DX4 feature summary.

The Am486DX4 resembles the IntelDX4 in its name, clock-tripler capability, system-interface, and pinout. But whereas Intel doubled the size of the cache on its DX4, AMD kept its cache size the same. The reason is that the Am486DX2 and Am486DX4 contain the same die, with the clock-multiplier circuit configured at assembly time.

The Am486DX4 requires a 3.3-V power supply and supports core frequencies up to 100 MHz. The part is supplied in a standard 168-pin PGA package.



## 7.13 Futures

While the Am486SX and Am486DX were constrained to be as close to Intel's designs as legally allowed, AMD plans its own proliferation of 486 variants to further differentiate its parts from Intel's.

The cleanest and clearest opportunities for 486 derivative products lie in the area of cache designs. Increasing cache size is one opportunity. This would be especially useful for clock-doubled and -tripled chips, since the higher internal clock rate doubles the cache miss penalty. Chips with larger on-chip caches could be fully pin-compatible with standard 486 chips, and the larger cache would enable systems using the chip to lead any performance comparisons. Now that Intel has introduced its IntelDX4, a clock-tripled 486 with twice as much cache as a conventional 486, AMD is under considerable pressure to double the cache on its chips as well.

Changing the caches to support copy-back as well as write-through operation is another possibility. (The IntelDX4 supports write-through operation only.) While the performance boost resulting from such a change might be minor, as would be the case with cache enlargements, even a few percent improvement would be noticeable—certainly enough to raise systems using such chips to the top of the charts in magazine roundups—and would incur minimal additional production cost.

Copy-back cache designs cry out for burst writes for dirty cache line write-backs, which the 486 bus does not define, so AMD might enhance the bus in this way. Bus extensions would also be needed to support cache coherency; a write-back cache must be snooped on read cycles from other bus masters, while a write-through cache needs to be snooped only on write cycles. (Chip set makers are already revising their designs to support write-back caches for Intel's Pentium and for the future 32-bit-bus version, the P24T. These chips are described in **Chapter 12: Pentium Microprocessors.**)

The additional signals could be added on "no-connect" pins of the standard 486, and the chip could default to write-through mode, providing full compatibility with existing designs. (Intel and Cyrix have both done so.) To fully exploit a write-back cache, however, motherboard designs would need to be revised, so it is natural for AMD to delay introduction of such a part until it has established its presence in the 486 marketplace and

tapped into the easiest business—simply filling unmet demand for standard 486 chips.

There are numerous other possibilities for products based on the 486 core, including chips that plug into Intel's OverDrive socket. AMD clearly has opportunities to broaden its product line and could, in theory, produce whatever of these products make marketing and financial sense.

But in the long run, AMD executives say that they now recognize the need to break free from merely duplicating Intel's designs, regardless of the legal situation; the 486 is the last Intel design that AMD plans to duplicate gate for gate. The AMD "K86" family of next-generation, superscalar processors is being developed independently of Intel's products. (The "K" in these code names is said to stand for "Kryptonite"—that glowing green metal in the comics that has the power to bring Superman to his knees.)

Information on the first of these products (code-named "K5") was revealed at the Microprocessor Forum held in October of 1994. While AMD hasn't yet established a track record for designing independent implementations of the 386 architecture, it does have experienced design teams that have been working on variations of Intel's designs and on the 29000 embedded RISC processor family. Some of the senior 29000 staff are now working on the K86 project. For further information see **Chapter 18: Future Directions**.

## 7.14 Commentary

In just three years of participation in the 386 and 486 market, AMD has become remarkably well established. Its customers include not only dozens of third-tier companies but most of the second tier and some of the first. Even IBM selected AMD processors in a low-end machine sold in Europe, and major companies such as Digital Equipment Corp. and AST and, more recently, Compaq have now begun featuring AMD-based products.

In the case of the 386 family, AMD was able to capitalize on its process technology. The AMD 386 product line was the first to go into volume production using the "CS21S" process, a linear shrink of AMD's "CS21" process with a minimum feature size of 0.8 micron. Even though AMD's products are derived from

Intel's logic design and microcode, AMD was thus able to distinguish its 386 product line by offering higher clock rates, lower power consumption, and low-voltage versions, plus the benefits of static operation and a system management mode in an otherwise-compatible pinout.

In the case of the 486 family, AMD was able to capitalize on a well-timed entry into the market. Throughout 1993, Intel was unable to meet the soaring demand for 486-family products, so there was a ready market for additional supply that AMD was able to provide. While the AMD 486s do not have the clock-rate advantage over Intel's chips that its 386s enjoy, its 40-MHz part offers makers of 486DX-33 systems an upgrade alternative to the 486DX2-66. AMD does not charge a premium for the 40-MHz part, while DX2 chips are significantly more expensive, so this will be a less costly enhancement.

AMD has also established dozens of customer relationships with PC makers and a track record for providing compatible products that made its 486 sales easier than its early 386 sales. AMD ramped 486 production steadily throughout 1993 and 1994, reaching a run rate of over one million units per quarter.

As 1994 drew to a close, AMD still found itself essentially production limited at its 486 fabrication plants. In order to maximize revenues, AMD is currently emphasizing only its high-margin Am486DX2 and Am486DX4 products, and is refusing even to quote prices on non-FPU and non-clock-doubled products. (Since the die sizes are the same, it costs AMD just as much to build an Am486SX as an Am486DX2.) This situation may change in 1995 as new fabrication capacity begins to come on-line.

### **Technical Comparisons**

Table 7-17 summarizes the technical differences among the various AMD 386 and 486 family products.

The worst-case power dissipation of AMD's devices is about one-third less than similar devices from Intel. Typical power dissipation is 44% lower. AMD's 386 also has the advantage of having no minimum clock frequency, allowing power consumption to be reduced further if speed can be sacrificed. With the clock stopped, the Am386DXL is specified to consume 1 mA maximum and 80  $\mu$ A typical.

### **Legal Entanglements**

Since even before its first 386 products were announced, AMD has been embroiled in a series of lawsuits with Intel over AMD's right to develop its 386 and 486 products as it did. Intel insists

Product Name	Static Operation?	AMD SMM?	Cache Size	FPU?	Clock Multiplier	V <sub>cc</sub>	F <sub>max</sub> (Core/ Bus)	F <sub>min</sub> (Core)	Pinout Class	Package Type
Am386SX	No	No	N.A.	387SX	+2	5.0 V	40 MHz	2.0 MHz	i386SX	PQFP
Am386SXL	Yes	No	N.A.	387SX	+2	5.0 V	40 MHz	0.0 MHz	i386SX	PQFP
Am386SXLV	Yes	Yes	N.A.	387SX	+2	3.3–5.0 V	33 MHz	0.0 MHz	i386SX	PQFP
Am386DX	No	No	N.A.	387DX	+2	5.0 V	40 MHz	2.0 MHz	i386DX	PGA or PQFP
Am386DXL	Yes	No	N.A.	387DX	+2	5.0 V	40 MHz	0.0 MHz	i386DX	PGA or PQFP
Am386DXLV	Yes	Yes	N.A.	387DX	+2	3.3 V	33 MHz	0.0 MHz	i386DX	PQFP
Am486SX2	Yes	No	8KB	None	×2	5.0 V	66 MHz/ 33 MHz	16.0 MHz	i486SX	PQFP
Am486DX	No	No	8KB	On-chip	×1	5.0 V	40 MHz	8.0 MHz	i486DX	PGA
Am486DX2	Yes	No	8KB	On-chip	×2	5.0 V	66 MHz/ 33 MHz	16.0 MHz	i486DX	PQFP
Am486DXL	Yes	Yes	8KB	On-chip	×1	5.0 V	40 MHz	0.0 MHz	i486DX	PQFP

Table 7-17. AMD 386 and 486 product feature comparison.

that AMD has infringed Intel microcode and other software copyrights, the 1976 agreement notwithstanding

At various times Intel has claimed (publicly, if not always in court) that the copyright agreement gave AMD the right to *copy* its microcode but not to *distribute* devices containing said copies; that AMD may copy microcode in *microcomputer system-level* products but not in *microprocessor component-level* products; and that *AMD* may have been licensed to copy Intel microcode but could not arrange to have the code copied by *outside foundries*, and so forth.

Moreover, Intel has claimed the 1976 agreement that covered *microcode* did *not* cover copyrighted software *other than microcode*, such as the “software” bit patterns in control PLAs, and that both the “overall control program” and the “floating-point control program” (whatever *they* are) both fall into this category. Intel has also claimed that AMD has used circuitry and microcode designed for support of Intel’s in-circuit emulators to implement its system management mode. This is an issue because the Intel/AMD agreement explicitly prohibits AMD from producing “bond-out” versions of the parts that provide access to this circuitry and microcode.

And finally, Intel asserts that AMD's patent license expires at the end of 1995. Not surprisingly, AMD disagrees with this interpretation, asserting that the license agreement may not cover patents applied for after 1995, but that its rights to existing patents last forever.

Verdicts swing like a pendulum between the two companies: Intel wins an arbitration, but the monetary damage award is insignificant; AMD wins one lawsuit, Intel wins another; one jury verdict gets set aside by a judge, another gets overturned on appeal. Intel has vowed to appeal its cases all the way to the Supreme Court if need be. In the meantime, AMD has continued to sell whatever devices it can find markets for, in ever-increasing volumes, the cloud of litigation be damned. (For full details and the latest information on the legal issues at stake, trials, verdicts, appeals, and so forth, see **Chapter 16: Legal Issues**.)

Legally speaking, AMD's not out of the water yet. If Intel were ultimately to prevail on any of the microcode licensing issues, AMD would be forced to switch to clean-room microcode by 1996. AMD has spent the last several years developing a "clean-room" version of the Intel microcode—several, in fact. If the tides turn against AMD, it could phase its production over to such a version. Customers would then need to requalify the clean-room devices (see **Chapter 17: Compatibility Issues**), but might continue using Intel-microcode chips in the meantime.

Should there be further delays or compatibility problems with the clean-room microcode, however, AMD's inability to use the Intel microcode would become significant. It is certainly easier to convince prospective customers of the compatibility of the part with Intel microcode, which is a key reason why AMD pursued this path in the first place. AMD may find it tough to get customers to switch to the AMD-microcode part if they have the option to stay with the Intel-microcode version, which would increase AMD's exposure to a future legal loss.

### **Business Strategy**

In order to recoup its design costs (to say nothing of its legal expenses), AMD hopes to receive the same fat margins Intel enjoyed for so long. AMD thus has little motivation to undercut Intel's price structure. Instead, AMD's strategy is to match Intel's prices one clock-step down.

As long as AMD is production-limited, it will seek to keep the price umbrella up. AMD also wants to avoid being perceived as

a bargain supplier and would prefer to emphasize its added features, such as faster clock rates and lower power, at comparable prices. While 486 prices are likely to drop significantly in the long run as a result of AMD's introduction, the big drops will probably not occur until mid-to-late 1995, when 486 supply begins to significantly exceed demand.

### **Production Limitations**

Aside from the technical, business, and legal challenges facing AMD, production capacity may be an area worthy of concern. The only facility at which AMD can currently build its 486 chips is at its sub-micron development center (SDC) in Sunnyvale, Calif. The SDC has been used for flash memory production, but AMD is moving flash production to its Fab 14 in Austin, Texas. The SDC is also used for research and development, and for some production of the 29000 embedded-processor family.

In late 1992, AMD began a \$160 million campaign to outfit the SDC as a production fab for 486 processors. AMD has stated that once the conversion is completed, it expects to ship \$250 million worth of 486 chips in the first 12 months of production and to achieve a run rate of \$100 million per quarter from the SDC alone. Based on our estimate of at least 60 good die per wafer and a low estimate of \$150 average selling price, AMD would need fewer than 1,000 wafers per week to reach its \$100 million quarterly goal. When fully outfitted, the SDC will be able to start nearly 3,000 six-inch wafers per week.

AMD's ability to ramp its 486 capacity beyond this level—and to build chips using its 0.5- and 0.35-micron processes, which are currently in development—is dependent on an as-yet-unfinished plant called Fab 25. This facility, adjacent to AMD's current buildings in Austin, initially will include 60,000 square feet of clean-room space. It will be capable of producing 5,000 eight-inch wafers per week when fully built out to its 80,000-square-foot capacity. The first test wafers from Fab 25 were due out by the end of 1994, with full production in mid-to-late 1995.

In contrast, Intel has fab sites in Santa Clara, Albuquerque, and Ireland that can each handle up to 5,000 eight-inch 0.6-micron wafer starts per week.

In February of 1994, AMD announced that Digital Equipment Corp. would act as a foundry for 486 products, beginning in 4Q94. DEC's production volume is expected to grow to 500,000 die per quarter by 2Q95, and will do much to bridge the gap until AMD's Fab 25 plant comes on line in late 1995.

TSMC is expected to take over most 486 production, and Fab 25 will be devoted primarily to manufacturing the K5.

## 7.15 For More Information...

Additional technical information on the AMD 386- and 486-family product lines may be found in the following publications:

### Vendor Publications

- 1: *3-Volt System Logic for Personal Computers Data Book*. Advanced Micro Devices, Inc., 1994, order #17028C.
- 2: *Am386 and Am486 Microprocessors Motherboard / System Manufacturers*. Advanced Micro Devices, 5/93, order #17672C. (Itemized list of 63 motherboard manufacturers and 50 system vendors.)
- 3: *Am386 Microprocessors for Personal Computers Data Book*. Advanced Micro Devices, 1992, order #11339C.
- 4: *AM486 DX2-80 High Performance, Clock-Doubled, 32-Bit Microprocessor*. Advanced Micro Devices, 7/94, order #19177.
- 5: *Am486 Microprocessor Low-Voltage Design Manual*. Advanced Micro Devices, 1993, order #17571A.
- 6: *Am486DX Data Sheet*. Advanced Micro Devices, 5/93, order #17852A.
- 7: *Am486DX2 Data Sheet*. Advanced Micro Devices, 5/93, order #17914A.
- 8: *Am486DXLV Data Sheet*. Advanced Micro Devices, 5/93, order #17381A.
- 9: *Am486SX Data Sheet*. Advanced Micro Devices, 6/93, order #18009A.
- 10: *Am486SX2-50 MHz Data Sheet*. Advanced Micro Devices, 2/94, order #17815B.
- 11: *Am486SXLV Data Sheet*. Advanced Micro Devices, 6/93, order #17878A.
- 12: *AMD K86 Microprocessor Family Architecture Press Kit*. Advanced Micro Devices, 10/18/94.
- 13: *AMD's Impact on Personal Computers*. Advanced Micro Devices, 9/94, order #18457B.

**Microprocessor  
Report Articles**

- 14: *Personal Computer Microprocessors Data Book*. Advanced Micro Devices, 1991, order #11339B.
- 15: *System Management Mode Application Note*. Advanced Micro Devices, 6/93, order #17927A.
- 16: *Intel Sues AMD Over 386 Trademark*. MPR vol. 4 no. 18, 10/17/90, pg. 4. (*Most Significant Bits item.*)
- 17: *Phelps Rules Intel Breached Contract with AMD\**. Michael Slater, MPR Report vol. 4 no. 19, 10/31/90, pg. 10. (*Feature article.*)
- 18: *AMD to Show 386-Compatible*. MPR vol. 4 no. 21, 11/14/90, pg. 4. (*Most Significant Bits item.*)
- 19: *AMD Ends 386 Monopoly\**. Michael Slater, MPR vol. 4 no. 22, 11/28/90, pg. 1. (*Cover story.*)
- 20: *AMD Formally Announces Am386DX\**. Michael Slater, MPR vol. 5 no. 6, 4/3/91, pg. 6. (*Feature article.*)
- 21: *Intel and AMD Settle Trademark-Related Issues*. MPR vol. 5 no. 7, 4/17/91, pg. 4. (*Most Significant Bits item.*)
- 22: *AMD Samples 386SX at 25 MHz*. MPR vol. 5 no. 8, 5/1/91, pg. 4. (*Most Significant Bits item.*)
- 23: *AMD Formally Announces 386SX*. MPR vol. 5 no. 13, 7/24/91, pg. 4. (*Most Significant Bits item.*)
- 24: *AMD Ships 386DX in Plastic*. MPR vol. 5 no. 17, 9/18/91, pg. 5. (*Most Significant Bits item.*)
- 25: *AMD Sues Intel, Alleging Anti-Trust Violations\**. Michael Slater, MPR vol. 5 no. 17, 9/18/91, pg. 10. (*Feature article.*)
- 26: *A History of Intel and AMD's Relationship--According to AMD\**. MPR vol. 5 no. 17, 9/18/91, pg. 12. (*Feature article.*)
- 27: *AMD Leads 3.3-V Charge, Adds SL-Like SMM*. MPR vol. 5 no. 20, 10/30/91, pg. 4. (*Most Significant Bits item.*)
- 28: *AMD Announces 486 Plans*. MPR vol. 6 no. 2, 2/12/92, pg. 4. (*Most Significant Bits item.*)
- 29: *386 Battle Advances*. MPR vol. 6 no. 2, 2/12/92, pg. 7.
- 30: *AMD Awarded 386 Rights, \$15 Million Damages\**. Michael Slater, MPR vol. 6 no. 4, 3/25/92, pg. 7. (*Feature article.*)
- 31: *AMD Loses 287 Microcode Case\**. Michael Slater, MPR vol. 6 no. 9, 7/8/92, pg. 1. (*Cover story.*)



- 32: *AMD Adds 40-MHz 386SX*. MPR vol. 6 no. 13, 10/7/92, pg. 4. (*Most Significant Bits item.*)
- 33: *AMD Plans Intel-Microcode 486*. MPR vol. 6 no. 14, 10/28/92, pg. 4. (*Most Significant Bits item.*)
- 34: *AMD Puts Intel on Notice for 486 Microcode*. MPR vol. 6 no. 15, 11/18/92, pg. 4. (*Most Significant Bits item.*)
- 35: *Judge Ingram Blocks AMD Use of Intel Microcode*. MPR vol. 6 no. 16, 12/9/92, pg. 4. (*Most Significant Bits item.*)
- 36: *AMD Expanding Fab in Anticipation of 486...* MPR vol. 6 no. 17, 12/30/92, pg. 4. (*Most Significant Bits item.*)
- 37: *AMD and HP Cooperate on New IC Process*. MPR vol. 7 no. 2, 2/15/93, pg. 5. (*Most Significant Bits item.*)
- 38: *Readers Pick AMD as Top Processor Vendor*. Linley Gwennap, MPR vol. 7 no. 2, 2/15/93, pg. 15. (*Feature article.*)
- 39: *AMD Jumps Into 486 Market*. Michael Slater, MPR vol. 7 no. 6, 5/10/93, pg. 1. (*Cover story.*)
- 40: *Intel /AMD Arbitration Ruling Gutted*. MPR vol. 7 no. 8, 6/21/93, pg. 4. (*Most Significant Bits item.*)
- 41: *AMD Cleans 486 Chips, Adds 486SX*. MPR vol. 7 no. 9, 7/12/93, pg. 4. (*Most Significant Bits item.*)
- 42: *AMD Pursues Palmtops With Elan*. MPR vol. 7 no. 9, 7/12/93, pg. 5. (*Most Significant Bits item.*)
- 43: *AMD Rumored to Have 50-MHz 486SX2*. MPR vol. 7 no. 11, 8/23/93, pg. 4.
- 44: *AMD Loses OmniBook Socket to TI*. MPR vol. 7 no. 12, 9/13/93, pg. 5. (*Most Significant Bits item.*)
- 45: *AMD Used Dirty "Clean Room"*. MPR vol. 7 no. 13, 10/4/93, pg. 4.
- 46: *Court Overturns Reversal of AMD Ruling*. MPR vol. 7 no. 13, 10/4/93, pg. 4.
- 47: *AMD's Elan Puts 386 PC in Pocket*. Linley Gwennap, MPR vol. 7 no. 14, 10/25/93, pg. 20. (*Feature article.*)
- 48: *AMD Extends 486 Line*. MPR vol. 7 no. 15, 11/15/93, pg. 4.
- 49: *AMD Describes Enhanced 486*. Michael Slater, MPR vol. 7 no. 15, 11/15/93, pg. 17. (*Feature article.*)
- 50: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (*Cover story.*)

- 51: *PDAs Begin Shipping in 1993*. Linley Gwennap, MPR vol. 8 no. 1, 1/24/94, pg. 18. (Feature article.)
- 52: *Compaq to Buy AMD 486 Chips*. MPR vol. 8 no. 2, 2/14/94, pg. 5.
- 53: *AMD Introduces Embedded 386 Chips*. MPR vol. 8 no. 3, 3/7/94, pg. 4.
- 54: *AMD Revs Up with 486SX2*. MPR vol. 8 no. 3, 3/7/94, pg. 4.
- 55: *Intel Matches AMD's 486SX2...* MPR vol. 8 no. 4, 3/28/94, pg. 5.
- 56: *AMD Wins Key Microcode Court Case*. Michael Slater, MPR vol. 8 no. 4, 3/28/94, pg. 10. (Feature article.)
- 57: *AMD Samples Half-Micron 486*. MPR vol. 8 no. 13, 10/3/94, pg. 4. (Most Significant Bits item.)
- 58: *AMD's K5 Designed to Outrun Pentium*. Michael Slater, MPR vol. 8 no. 14, 10/24/94, pg. 1. (Cover story.)
- 59: *Court Allows AMD to Continue 486 Shipments*. MPR vol. 8 no. 15, 11/14/94, pg. 4. (Most Significant Bits item.)
- Other Technical References** 60: *Superscalar Microprocessor Design*. W. M. Johnson, Prentice-Hall, 1991, ISBN 0-13-875634-1. (authored by Mike Johnson.)
- Other Periodicals** 61: *Compaq Rocks Corporate World with AMD Chip*. Brooke Crothers and Bob Francis, Info World, vol. 16 no. 38, 9/19/94, pg. 1.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)

# 8

## C&T 386 Microprocessors

---

After more than three years of development, Chips and Technologies made a long-awaited plunge into the microprocessor arena in 1991 by announcing a barrage of at least six planned products. The "Super386" product line was intended to include 386SX- and 386DX-class processors that were pin-compatible with Intel's chips (but offered somewhat higher performance at the same frequency), processors with faster clocks, and non-pin-compatible chips with a small on-chip instruction cache.

C&T also announced plans to support a set of new architecture extensions for improved power management called "SuperState," analogous to Intel's System Management Mode, and to produce "bond-out" versions of the chips, which would provide in-circuit emulator makers access to internal status signals.

These products were to be the culmination of a \$50 million R&D project that would propel the company beyond being merely a builder of support chips into becoming a one-stop-shopping PC component supplier. When the new product line was combined with its existing line of chip sets, peripheral controllers, and LAN controllers, C&T was able to provide all the components for a personal computer except the system memory and glue logic. In addition to improving the company's margins in the cut-throat chip-set business, this strategy was intended to provide C&T with the building blocks for next-generation, highly integrated "system chips."

**Overview** Whereas the Intel 386 and 486 were based on completely original designs, and the various AMD 386 and 486 designs were extracted from Intel's, C&T designed its version of the 386 from scratch. This produced both good news and bad news.

On the plus side, with the design completely under its control, C&T was able to design an entirely new pipeline, enhance the pinout, extend the underlying architecture, and develop all-new microcode. This gave the C&T parts the potential to perform considerably better than comparable offerings from Intel and AMD.

On the minus side, with the ability to design an entirely new pipeline, enhance the pinout, extend the underlying architecture, and develop all-new microcode, C&T chose to do exactly that. The already daunting design task was further delayed by a midcourse correction affecting the aggressiveness of the pipeline enhancements, and the design required several debugging and revision cycles before it was ready for prime time.

Design engineering, alas, proved to be only the first of C&T's challenges. While the resulting parts were indeed better in some ways than Intel's, they were nevertheless different. System designers were concerned about the possibility of software compatibility problems, and were reluctant to commit their companies' futures to a sole-sourced product line from a company with no track record in microprocessors.

Intel fought back with saturation brand-name promotion campaigns, the "Intel Inside" advertising rebate program, a none-too-subtle threat of litigation, and other techniques to "persuade" system vendors and buyers to remain faithful. Only two of the planned Super386 designs ever reached fruition, and they sold poorly—if at all. Further design work was soon discontinued.

While C&T is no longer selling its 386 products, it is still instructive to examine the products it *did* produce, in order to understand the alternative implementation techniques incorporated into these designs. And who knows? The C&T core logic may someday rise again, in the form of a highly integrated palm-top CPU or 32-bit embedded controller.

**Core Design** The Super386 design was purportedly entirely original. C&T did not examine or reverse-engineer the Intel chip-level design; instead it developed a target specification by starting with Intel's public 386 documentation and writing software test

routines to determine how Intel's chips behaved under unspecified conditions. C&T studied Intel's patents and went to considerable lengths to work around them. The C&T microcode was developed under "clean-room" conditions in order not to infringe Intel's copyright.

Like AMD's design, the C&T Super386 core was fully static. Its three-stage pipeline was a clear improvement over the Intel and AMD chips. Register-to-register operations could complete in a single clock cycle vs two for the Intel core. Instructions that fetched values from memory required a minimum of two clock cycles instead of four. Instructions that operated on memory-based operands require one or two fewer cycles to complete than do Intel processors.

Much of the improvement of the Super386, though, came from optimized branch-processing logic. The Intel 386 core requires a minimum of eight clock cycles to perform a branch. The Super386 cut this to six. Moreover, a dedicated adder and a special one-entry instruction TLB in the Super386 precomputed branch-offset addresses and reduced the time required for most conditional branches with an eight-bit relative offset to just two cycles, four times faster than Intel. Assuming jumps and branches account for about 12% of all x86 instructions executed, this single optimization boosted performance by nearly 10%.

## 8.1 The C&T 38600DX Microprocessor

The Chips and Technologies 38600DX was designed to be a slightly enhanced but pin-comparable variation on the original i386DX. Table 8-1 summarizes the general features and specifications of the 38600DX microprocessor.

Product Name	C&T 38600DX
Introduction Date	September 1991
Production Status	Deceased
Device Integration Level	Same as i386DX
CPU Architecture Level	Standard 386 integer instruction set
Core Technology	C&T-designed static 386 core
Pinout	Upwardly compatible with i386DX
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i386DX
Cache Support	Optional external cache controller
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	25- and 33-MHz core operation sampled; 40-MHz version planned
Clocking Regime	Core operation frequency externally configurable to be $CLKIN \times 1$ or $CLKIN \div 2$
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Die Size	Not released; assumed to be large
Transistor Count	Not released; assumed to be large
Package Options	132-pin PGA or 132-lead plastic QFP

Table 8-1. C&T 38600DX feature summary.

### System Interface

The 38600DX system interface is essentially identical to that of the i386DX. The only differences involve one new signal and the modified operation of another. The names and functions of these signals are summarized in Table 8-2.

Signal	Direction	Function	i386DX Signal
USE2X	In	Selects between standard 2 $\times$ clock input and optional 1 $\times$ clock mode	Vcc
CLKIN	In	Configurable-mode clock input signal	CLK2

Table 8-2. C&T 38600DX special interface signals.

These signals give designers the option to use the part with a 1× external system clock instead of the 2× clock required by Intel's 386. The need for a double-frequency oscillator makes system design and FCC approval more difficult, especially at frequencies above 33 MHz. Since C&T had originally planned to build 40-MHz parts someday, the 38600DX incorporated configurable clock-generation logic.

If the USE2X input pin is high, CLKIN required a standard 2× external system clock. If USE2X is low, a 1× clock may be used. The pin to which C&T assigned the USE2X pin serves as a Vcc supply pin on i386DX chips, so if a 38600DX were to be plugged into a standard 386 motherboard, it would default to “normal” operation. A minor board change was needed to connect this pin to ground, thereby letting system designers switch to a 1× external oscillator.

### **Package and Frequency Options**

The C&T 38600DX was offered in the same 132-pin PGA and 132-lead PQFP packages as the i386DX. It was initially sampled in 25-MHz and 33-MHz versions. A 40-MHz version was planned and announced but never produced.

### **Relative Performance**

Since the internal logic of the 38600DX differed from that of the Intel and AMD devices, it exhibited different instruction timing. All told, C&T claimed the 38600DX pipeline improvement mentioned above made it run about 10% faster than the i386DX at any given clock frequency.

## 8.2 The C&T 38605DX Microprocessor

The C&T 38605DX was a 38600DX with an on-chip instruction cache and an enhanced (but physically incompatible) pinout. Table 8-3 summarizes the general features and specifications of the 38605DX microprocessor.

Product Name	C&T 38605DX
Introduction Date	September 1991
Prognosis	Deceased
Device Integration Level	Microcoded 32-bit IEU + PMMU 512-byte instruction cache
CPU Architecture Level	Standard 386 integer instruction set
Core Technology	Static C&T-designed 386 core
Pinout	Expanded 386DX functions in custom PGA package
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i386DX
Cache Support	On-chip direct-mapped 512-byte instruction cache
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	25- and 33-MHz core operation sampled; 40-MHz version planned
Clocking Regime	Core operation frequency externally configurable to be $CLKIN \times 1$ or $CLKIN \div 2$
Process Technology	1.0 $\mu$ two-layer-metal CMOS
Die Size	Not released; assumed to be large
Transistor Count	Not released; assumed to be large
Package Options	144-pin PGA or 132-lead PQFP
Notes	Bond-out option using same die as 38600DX

Table 8-3. C&T 38605DX feature summary.

In fact, the 38605DX contained the same silicon die as the 38600DX, but with the on-chip cache logic enabled and additional signals bonded out.

### Cache Design

More efficient pipelines require greater bus bandwidth. Since a standard i386DX consumes approximately 75% of its available bandwidth for instruction and data transfers, there's a limit to the extent an improved pipeline might actually affect overall system performance. No matter how fast the core, the bus would saturate if performance were to improve by even 33%.



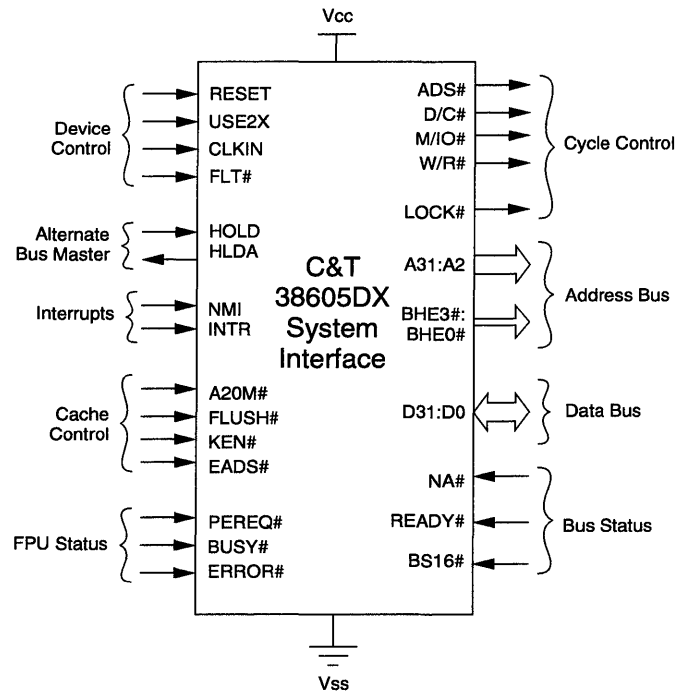


Figure 8-1. C&T 38605DX system interface.

The 38605DX thus contained a small, 512-byte instruction cache. The cache was direct mapped, with 32 lines of 16 bytes each. While even a small cache was a tremendous improvement over the original 386, the 38605DX cache was severely compromised. Compared to an Intel i486DX cache, for example, the C&T design had just one-sixteenth the capacity, could buffer instructions only (vs integrated instructions and data), did not support multiple set associativity, and did not support burst-mode transfers. Thus, the extent to which the on-chip cache might have improved device performance fell far short of its potential.

### System Interface

In order to support the new cache, the 38605DX system interface required several enhancements over that of the i386DX. These are shown in Figure 8-1.

Six new signals were added to the standard i386DX pinout. The names and functions of these signals are summarized in Table 8-4.

Signal	Direction	Function
USE2X	In	Selects between standard 2× clock input and optional 1× clock mode
CLKIN	In	Configurable-mode clock input signal
A20M#	In	Address-bit 20 mask. Forces address-bit 20 low internally so cache addresses that overflow 1MB wrap around as required for PC compatibility
KEN#	In	Cache enable. Allows data read from system memory to be stored in the cache
FLUSH#	In	Cache flush. Marks the entire cache as invalid
EADS#	In	External address strobe. Indicates a cache-snoop address is present on the address bus

Table 8-4. C&T 38605DX special interface signals.

The first two signals operate as on the 38600DX. The others perform the same functions as similarly named signals in Intel 486-class devices (see **Chapter 6**).

Curiously, the 38605DX did not adopt the 38600DX approach of mapping newly defined signals onto existing power or no-connect pins. Instead, C&T put the 38605DX in a new, larger 144-pin PGA. The rationale appeared to be that motherboards would need to be redesigned anyway to make best use of the new cache-snooping capability, so C&T saw little benefit in maintaining strict i386DX socket compatibility.

In a concession to raising the system makers' comfort levels, however, the pin assignments of the 38605DX package were chosen such that motherboards could accept either a standard 386DX from Intel, AMD, or C&T or the enhanced C&T device. This "universal socket" design was based on a single 176-pin PGA footprint with the inner and outer rows of pins connected together. "Standard" 386DX-pinout parts would plug into the inner rows of the socket, while the larger 38605DX would plug into the outer rows. This was supposed to allow system vendors to produce both standard 386 systems and enhanced Super386 systems from a common system-board design.

### Relative Performance

C&T claimed the 38605DX could typically deliver a 20% to 40% performance edge over the i386DX at the same core frequency and with the same memory system. Most of the benchmarks C&T used in its comparisons, though, were fairly small and contained short loops that likely exaggerated the value of the CPU's tiny on-chip cache.

## 8.3 Commentary

When the Super386 family was announced, C&T set its pricing to be comparable to the Intel products with which it competed. Starting a price war clearly would not have been in C&T's best interest, since its products were larger and more expensive to build and had many fewer years of experience riding the cost-reduction learning curve. Instead, the technical advantages of a faster pipeline, the ability to use a 1× system clock, and an optional on-chip cache were supposed to be a bigger lure for system makers.

A second factor in the Super386 family's favor was supposed to have been C&T's established relationships with PC clone vendors. While new to CPUs, C&T had long-established relationships with many prospective buyers. Bundling processors with its chip sets might have given C&T a further edge over its chip-set-only and CPU-only competitors.

To head off compatibility concerns, C&T had performed extensive software testing, both in-house and at a third-party lab. The company claimed there were no compatibility problems. User concerns remained, nevertheless. C&T had to fight the brand-name image Intel had built, which made system vendors—and, in time, end-users—wary of using “off-brand” products.

### **Too Little, Too Different, Too Late?**

In the end, the problems that killed the Super386 family seem to reflect the fact that it had been technology driven rather than market driven. Each of its features and improvements was more gimmicky than profound.

The noncached 38600DX was aimed at the upgrade market for mainstream applications but offered little benefit over a standard device. Its 10% performance boost, while enough to register in benchmark comparison reports, was not enough of an improvement to make a user-perceptible difference. Moreover, the device was inherently more expensive to build, since it contained the same cache logic (albeit disabled) as the 38605DX on the same oversize die.

Any real performance improvements required system vendors to adopt the non-pin-compatible 38605DX, whose additional signals and larger package required system boards to be redesigned to accommodate them. While the revised design might well have enabled significantly better performance, getting sys-

tem makers to create a new motherboard specifically for C&T's chips proved much harder than getting them to try out a pin-compatible product, and committed them to a unique device built only by a company with unstable finances and no micro-processor track record.

In this regard, the 176-pin "universal PGA socket" pinout gimmick may also have been too clever by half, and may have backfired. The message it presented to system designers was essentially this: "Here's how you can design your boards so you can buy our chips for now and yet not burn any bridges to the future. If C&T drops the ball trying to build these things, hey, no problem! You can always go back to using parts with a conventional pinout later!" This built-in fall-back contingency plan turned into a self-fulfilling prophesy.

Moreover, even the ability of the C&T devices to use a 1× system was, in practice, of only limited usefulness. Since 386 chip sets generally require a 2× system clock anyway, eliminating the need for a double-speed clock didn't really matter.

Then there was the inevitable issue of Intel litigation. C&T may have been convinced that its efforts to make the Super386 "litigation proof" would help the company prevail in court, but such efforts could not, of course, eliminate the *threat* of Intel's suing C&T in the first place. Intel did indeed sue, claiming that *any* device that was compatible with the x86 architecture and PMMU structure must inherently violate *some* Intel patent. (See **Chapter 16: Legal Issues** for a discussion of Intel's dreaded "338" patent.) C&T settled out of court; the terms of settlement are not known.

But the biggest factor in C&T's undoing may have been that Cyrix began shipping its parts at about the same time, parts with superior performance, full pin compatibility, and a product-numbering scheme that was much easier to promote. System vendors seemed to understand all of these concerns, and the resulting fear, uncertainty, and doubt caused them to flock away from the C&T parts in droves.

## 8.4 For More Information...

Additional information on the C&T products may be found in the following publications:

- Vendor Publications** 1: *Super386 DX Performance Test Report*. Chips and Technologies, 1991, order #080030-001.
- Microprocessor Report Articles** 2: *Chips and Technologies Launches "Super386," 387 Coprocessors, and a Single-Chip PC\**. Michael Slater and Brian Case, MPR vol. 5 no. 18, 10/2/91, pg. 1. (Cover story.)
- 3: *Intel Sues C&T for Patent Infringement\**. Michael Slater, MPR vol. 6 no. 4, 3/25/92, pg. 11. (Feature article.)
- 4: *C&T Files Counterclaims Against Intel*. MPR vol. 6 no. 8, 6/17/92, pg. 4. (Most Significant Bits item.)
- 5: *C&T Cancels 386SX, 486 Programs*. MPR Report vol. 6 no. 11, 8/19/92, pg. 4. (Most Significant Bits item.)
- 6: *IBM Picks Up C&T's x86 Code*. MPR vol. 8 no. 5, 4/18/94, pg. 5. (Most Significant Bits item.)

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



# 9

## Cyrix 486 Microprocessors

---

Cyrix is perhaps Intel's most aggressive competitor, at least from the perspective of design innovation. In a relatively short time this fairly young company managed to field an impressive array of 486-class microprocessors of varying pinouts and capabilities. Cyrix's stated objective is to locate gaps left in the price/performance continuum by Intel and AMD, and to fill those gaps with unique designs. This chapter describes the 486-family devices in the Cyrix 486 CPU arsenal.

Cyrix entered the 386/486 microprocessor market in 2Q92 with a pair of chips that combined a 486-like integer core and a 1-Kilobyte cache with 386SX- and 386DX-class bus interfaces and pinouts. While the initial devices did not provide an on-chip FPU, either could be used with a standard 387-type coprocessor from Cyrix, Intel, or other vendors. Later 486-family products have been based on the same core as the early introductions, but have been augmented by adding caches with larger capacity and a new copy-back mode, clock-doubler circuits, higher-bandwidth system interfaces, and on-chip floating-point units.

### 9.1 Core Design

Cyrix developed its 486-family processors from scratch, creating the logic design and writing the necessary microcode based on publicly available specifications and the observable behavior of Intel 486-family devices.

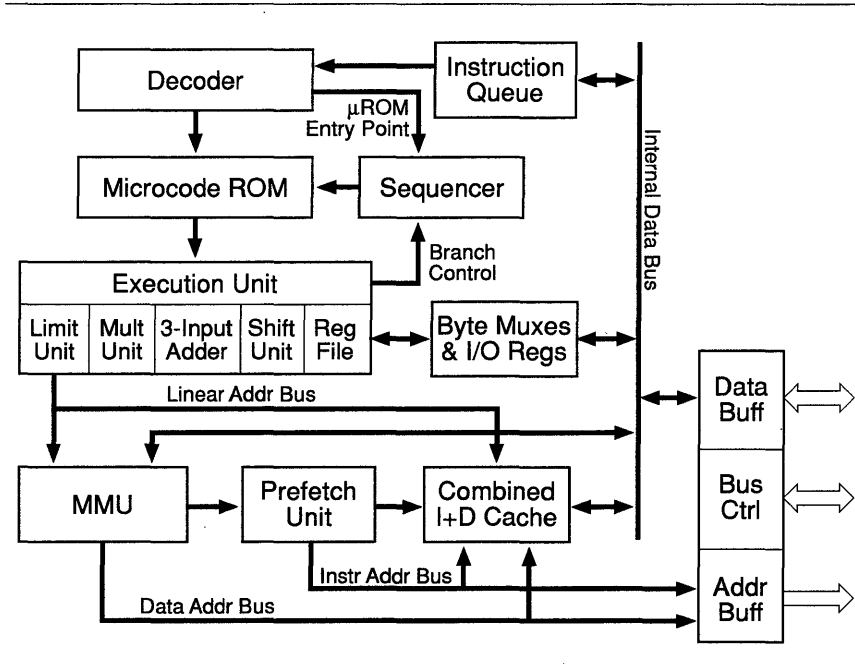


Figure 9-1. Cyrix 486 core microarchitecture.

As shown in the block diagram in Figure 9-1, the major functional logic blocks of the Cyrix core are the microcode ROM, the instruction prefetch queue, the five-stage execution pipeline, the TLB, a two-entry write buffer, and a combined instruction/data cache. Internal data paths between units are generally 32 bits wide. The decode logic processes four bytes from the instruction stream during each cycle, regardless of instruction boundaries.

The five-stage execution pipeline is very similar to that of the 486 core. The five stages are fetch, decode, micro-ROM access, execute, and register write-back. Intel's 486 has two decode stages, but the micro-ROM access of the Cyrix 486 core is essentially a decode function, so these pipelines appear to be nearly identical. In particular, the same branch penalty considerations should apply to each.

The most unusual execution resource in the Cyrix 486 core is a hardware multiplier, which produces a 32-bit result from two 16-bit operands in just three clock cycles, as compared to 12 to 25 cycles for an Intel 386 core and 13 cycles for an Intel 486. Devoting additional silicon area to a fast integer multiplier is uncommon in general-purpose microprocessors, but Cyrix claims it boosts the performance of display drivers and is also valuable for handwriting recognition in pen-based systems. In



addition, the fast multiply could enable the chip to be used for some DSP functions.

The Cyrix core can execute simple instructions in one cycle, but, as with Intel's 486, additional cycles are required for operand specifiers and instruction-prefix bytes. One difference in the resources included in each processor is that the Cyrix core uses the adder logic within the ALU to compute memory addresses, adding an extra clock cycle to most instructions that must compute a memory address.

Instruction	Intel/AMD 386	Cyrix 486	Intel/AMD 486	Comments
ADD, SUB, AND, OR, XOR				
reg-to-reg	2	1	1	
mem-to-reg	6	3	2	
reg-to-mem	10	3	3	
CMP				
reg-to-reg	2	1	1	
mem-to-reg	5	3	2	
reg-to-mem	5	3	2	
MUL (acc with reg)				
multiply byte	12-17	3	13-18	min-max
multiply word	12-25	3	13-26	min-max
multiply dblwd	12-41	7	13-42	min-max
SHL/SHR (shift left/right)				
reg by 1	3	1	3	
reg by CL	3	2	3	
String Instructions				
REPNE CMPS †	5+9c	5+8c	7+7c	(find match), count>0
REP MOVS †	7+4c	5+4c	7+5c	(move string), count>1
REPNE SCAS †	5+8c	4+5c	7+5c	(scan string), count>0
STC, CLC	2	1	2	

Table 9-1. ALU instruction core cycle count comparison.

Note: shaded cells indicate lowest cycle count  
 † c=count (number of iterations of string operation)

## Pipeline Performance

Tables 9-1 through 9-3 compare the clock counts for selected instructions in the Intel/AMD 386 core, the Cyrix 486 core, and the Intel 486 integer core. The Cyrix core matches the performance of the Intel/AMD 486 core on most simple instructions,

Instruction	Intel 386	Cyrix 486	Intel 486	Comments
MOV				
reg-to-reg	2	1	1	
mem-to-reg	4	2	1	
reg-to-mem	5	2	1	
POP				
register short form	6	3	1	
memory				
POPA	40	18	9	(pop all), 16-bit/32-bit operands
POPF	5	4	6	(pop flags)
PUSH				
register short form	4	2	1	
memory				
PUSHA	34	17	11	(push all)
PUSHF	4	2	3	(push flags)

Table 9-2. Data-transfer instruction core cycle counts.

Note: shaded cells indicate lowest cycle count

Instruction	Intel 386†	Cyrix 486	Intel 486	Comments
Jump conditional	7+m	4/1	3/1	taken/not taken
JMP (within segment)				
8-bit	7+m	4	3	
register indirect	9+m	6	5	
CALL				
direct within segment	9+m	7	3	
indirect within segment	9+m	8	5	
direct intersegment	42+m	12	20	to same level
indirect intersegment	46+m	14	20	to same level
RET				
within segment	12+m	10	5	
intersegment	36+m	13	18	to same level
LOOP	11+m	9/3	7/6	loop/no loop

Table 9-3. Protected-mode control instruction core cycle counts.

Note: shaded cells indicate lowest cycle count  
 † m=number of fields in target instruction

which are generally the most frequent instruction formats used. In a few instructions—most notably the multiply instructions—the Cyrix core is actually faster than Intel's. The Cyrix core is not, however, as fast as Intel's on many other instruc-

tions. In particular, the lack of a dedicated address adder in the Cyrix core slows down jumps and calls, and most memory-reference instructions that involve multiple address components.

### **Architecture Enhancements**

The Cyrix 486 core implements the complete standard 486 integer instruction set, i.e., the entire 386 instruction set plus the six new instructions defined by Intel for the original 486 devices; see **Chapter 6: Intel 486 Microprocessors** for an description of these instructions. All Cyrix products currently in production implement additional instructions for system management, as discussed later in this chapter.

The Cyrix 486 core also supports each of the control, test, and debug registers implemented within Intel's original (non-SL-enhanced) 486 design. In addition, Cyrix has added new "configuration control registers" to enable cache and other device-specific extensions not defined by the original Intel architecture. The bit-fields and functions performed by these registers are defined within the product descriptions in this chapter.

### **System Management and Standby Modes**

Each of the current Cyrix products intended for OEM system applications implements Cyrix's own private flavor of system management mode (SMM) operation. From a system-interface perspective, the Cyrix SMM functions resemble AMD's approach more than Intel's. Two new pins are associated with SMM: a system management interrupt request and a system management address strobe. Alternatively, SMM may be entered by setting an SMM access bit in a control register, or by executing a new SMINT instruction opcode.

Like the AMD parts, the Cyrix 486 core uses a static circuit design, such that the processor clock may be stopped at any point in its execution cycle to reduce power consumption. In addition, though, the Cyrix core supports a special "suspend mode" that may be invoked before stopping the clock for even greater savings.

The CPU enters suspend mode in response to the assertion of the SUSP# input pin (see Figure 9-2) or the execution of a HALT instruction. In either case, the processor completes any pending instructions and data transfer operations before asserting SUSPA#. External circuitry can then stop the processor's clock.

Entering suspend Mode reduces current drain by about three orders of magnitude. Put another way: for every hour of active CPU life a battery can supply, suspending the CPU will extend its life by about six weeks. Stopping the CLK2 input reduces cur-

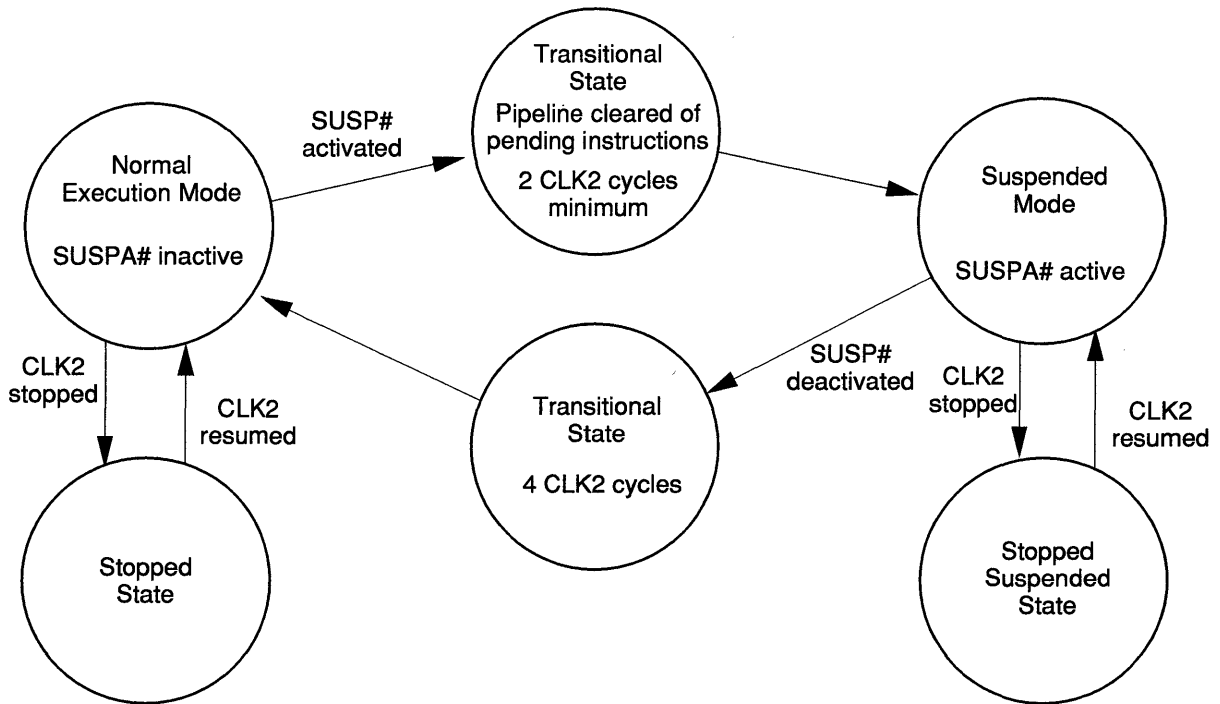


Figure 9-2. Cyrix core Suspend Mode state transition diagram.

rent by another factor of 10—extending the one-hour CPU battery's life by an extra year or so.

## 9.2 The Cyrix Cx486SLC and Cx486SLC/e Microprocessors

The Cx486SLC and Cx486SLC/e microprocessors are enhanced high-performance implementations of a 486SX-class device in a 386SX-class pinout. Table 9-4 summarizes the general features and specifications of these two products.

Product Names	Cyrix Cx486SLC and Cx486SLC/e
Introduction Date	Cx486SLC: April 1992 Cx486SLC/e: November 1992
Prognosis	Cx486SLC: Deceased Cx486SLC/e: Stable
Device Integration Level	Pipelined 32-bit IEU and PMMU 1K-byte unified instruction/data cache Hardware 16-bit x 16-bit multiplier
CPU Architecture Level	Standard 486 integer instruction set "e" adds Cyrix SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible i386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Two-cycles minimum per 16-bit transfer Optional one-half cycle address pipelining
Cache Support	1K-byte unified I- and D-cache Direct or two-way set associative Write-through operation only
Floating-Point Support	Optional external Cx87SLC, Cx3S87 or i387SX FPU
Operating Voltage	4.5 V to 5.5 V (core frequencies up to 25 MHz) 4.75 V to 5.25 V (at 33 MHz)
Frequency Options	25- or 33-MHz core operation
Clocking Regime	Core operating frequency = 1/2 x Clkin
Active Power Dissipation	3.75 W @ 5.0 V and 33 MHz (worst case)
Power-Control Features	Cyrix SMM extensions Stopped-clock and suspend-mode operation
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils x 410 mils (10.5 mm x 10.5 mm)
Transistor Count	600,000 transistors
Package Options	100-pin PQFP

Table 9-4. Cyrix Cx486SLC and Cx486SLC/e feature summary.

The first device introduced was designated simply the Cx486SLC. A later redesign, designated the Cx486SLC/e, provided a set of hardware and software enhancements for a Cyrix-defined System Management Mode (SMM). In time, the origi-

nal, nonenhanced device was discontinued, and the term “Cx486SLC” is often applied to either part. Unless otherwise stated, within the text of this chapter, the simpler, non-“/e” designation is used to describe features and capabilities that apply to both devices.

The Cx486SLC devices are aimed primarily at notebook computers, but may be suitable for entry-level desktop systems. The device can also be used to upgrade existing 386SX designs. On reset, the part is initialized to a state in which it operates like a standard i386SX or Am386SX device. The cache, pinout extensions, and other features that might lead to software incompatibilities are all automatically disabled.

**Features** On the surface, Cyrix’s Cx486SLC is similar in concept to C&T’s ill-fated Super386 series; each design combined a pipelined CPU core and a small cache within a 386-inspired pinout. What set Cyrix’s approach apart is that its core was faster than C&T’s, its 1K cache was twice as large and stored data as well as instructions, and, most important, the Cyrix device did not require circuit board redesign to take advantage of its new features.

While the on-chip cache requires minor hardware modifications in order to deliver maximum performance, the chip’s software configuration options make it possible to install the device in existing systems with no hardware modifications. The processor can be configured by instructions in the BIOS boot ROMs or by a small program executed in DOS’s autoexec.bat to initialize the on-chip hardware into a fail-safe mode. On-chip registers can be programmed to enable the cache and to limit the conditions under which external data is cached on-chip.

Software can optionally enable the new hardware features in various ways, depending on the capabilities of the host system design, in order to deliver a level of performance considerably greater than a conventional 386SX. Its smaller cache, lack of dedicated address-generation circuitry, and narrower data bus, however, limit the part’s performance to somewhat below that of a “true” 486SX.

The remainder of this chapter describes operation of the Cx486SLC/e devices, noting any major differences between them and the original, unenhanced device.

**Cache Configuration** The Cx486SLC has a rich set of cache and cache-related features. The 1K-byte combined instruction/data, write-through

cache can be configured under software control for either a two-way set-associative or direct-mapped organization. As with the 486 cache, a write miss does not cause a cache line to be allocated.

The fact that the Cx486SLC's 386SX-style 16-bit system interface provides significantly less bandwidth than that of a "true" (Intel-pinout) 486 and the part's lack of support for burst-mode data transfers have several ramifications on the Cyrix caching strategy. Whereas a device with an Intel-style 486 pinout can fetch a 16-byte burst in 5 cycles total, it would require at least 16 cycles for a Cx486SLC to perform the same feat. Thus it makes sense for the part to fetch just the memory locations needed. It takes a different cache organization and extra cache logic to keep these partial-line transfers organized. The Cyrix cache has a 4-byte line size with one valid bit per byte vs a 16-byte line size with a single valid bit for Intel 486s. Adding three extra tag fields and 15 extra valid bits per 16-byte block costs die size—the Cyrix tag and valid-bit arrays consume as much die area as the data array itself—but improves performance by eliminating the need to fetch unnecessary values when a cache miss occurs for a single 16-bit access.

In addition to the A20M# pin, the KEN# pin and the noncacheable bits that are part of the 486 page-table structure, the Cx486SLC provides two other software-determined cacheability controls. Software can set the starting address and size of up to four noncacheable address regions by writing to control registers. Noncacheable regions can range in size from 4K bytes to 4G bytes.

The other cacheability control makes uncacheable the first 64K bytes of every 1M byte region. This facility provides a software alternative to the A20M# pin for solving the problems created by the 8086 artifact of address wrap-around at the 1M-byte boundary, and it allows the Cx486SLC to be used in unmodified i386SX systems that don't provide the A20M# signal.

One feature of Intel 486 cache designs that is missing in the Cx486SLC is bus snooping. This feature allows external bus activity controlled by an external master during periods of bus hold to cause individual cache-line invalidations in the 486 internal cache. On an Intel 486, this function is enabled by driving an external address onto the device address bus and asserting the EADS# pin. This feature is not supported by 386SX chip sets. Cyrix believed the prospect of retrofitting existing motherboard designs to support bus snooping capability would have

required additional and unnecessary system complexity, so the feature was omitted.

To compensate for the absence of this capability, a bit in a Cx486SLC control register can be set so that the internal cache is completely flushed whenever bus hold is entered. Invalidating individual cache lines with EADS# is important for the 486 because its 8K-byte cache is relatively large. Since the Cx486SLC cache is small, simply flushing it does not cause as significant a penalty.

In a typical notebook or low-end desktop system, the only bus-master device other than the processor is the DMA controller, and DMA is typically used only for the floppy disk and, if it is present, a LAN interface.

Memory coherency can be ensured with the Cx486SLC either by using the automatic cache flush during bus hold, as described above, or by marking as non-cacheable the memory areas used for DMA data buffers. For systems that have other bus masters, including a display controller, bus snooping is more important and present high-end chips from Cyrix already implement it.

### **Instruction Set Additions**

The Cx486SLC implements the standard 486 integer instruction set, i.e., each of the instructions defined by the 386, plus the six new instructions introduced by the original Intel 486 devices. In addition, the "/e" version of the device defines seven new instructions used within system management mode. Table 9-5 lists the functions performed by each of these instructions.

BSWAP, XADD, CMPXCHG, INVD, WBINVD, and INVLPG perform the same operations as the Intel 486 devices; refer to **Chapter 6: Intel 486 Microprocessors** for details.

At the beginning of a system management interrupt routine the SVDC, SVLDT, and SVTS instructions may be used to save (respectively) the contents of the segment registers, the Local Descriptor Table Register, and the Task State Register to SMM memory, along with their associated segment descriptors. The RSDC, RSLDT, and RSTS instructions restore the same registers before returning from the SMI service routine.

The RSM instruction restores the CPU state registers and resumes normal CPU operation in its prior execution mode, following completion of an SMM interrupt service routine.



Instruction	Mode	Operation	Opcode
BSWAP	User/ System	Byte swap. Reverse byte order within register	0FC8H
XADD	User/ System	Atomic (indivisible) exchange and add	0FC0H
CMPXCHG	User/ System	Atomic (indivisible) compare and exchange	0FB0H
INVD	System	Invalidate data cache	0F08H
WBINVD	System	Perform write-back cycle and invalidate cache	0F09H
INVLPG	System	Invalidate TLB page entry	0F01H
SVDC †	SMM	Save segment register (DS, ES, FS, GS, or SS) and associated descriptor	0F78H
SVLDT †	SMM	Save LDTR and descriptor	0F7AH
SVTS †	SMM	Save TSR and descriptor	0F7CH
RSDC †	SMM	Restore segment register and descriptor	0F79H
RSLDT †	SMM	Restore LDTR and descriptor	0F7BH
RSTS †	SMM	Restore TSR and descriptor	0F7DH
RSM †	SMM	Resume normal execution mode	0FAAH

Table 9-5. Cyrix Cx486SLC/e instruction set additions.

† = instructions supported only by enhanced ("e") devices

## Configuration Registers

In addition to the standard Control Registers, Test Registers, Breakpoint Registers, and so forth defined by the Intel 486 architecture, the Cx486SLC/e defines several new device configuration registers, as shown in Figure 9-3.

Most of the new hardware functions of the Cx486SLC may be optionally enabled by setting control bits in two new Configuration Control Registers, designated CCR0 and CCR1. The bit fields within these registers and the functions performed by each are defined in Figures 9-4 and 9-5.

In many PCs there exist regions within the system-memory address space that should not be cached within the CPU. For example, if a particular block of the system address space is known to contain memory-mapped I/O ports or a DMA-transfer buffer for a high-speed network adapter, CPU accesses to this region should *not* be cached, to ensure that the CPU will retrieve updated data (rather than reread a copy of previously fetched data) each time the data is loaded.

With a "true" 486-style bus interface, it is the responsibility of external logic to deassert the KEN pin when noncacheable memory regions are addressed. Alternatively, a 386 protected-mode operating system may set attribute bits within page tables to ensure that dynamic memory regions are not cached. Often, in

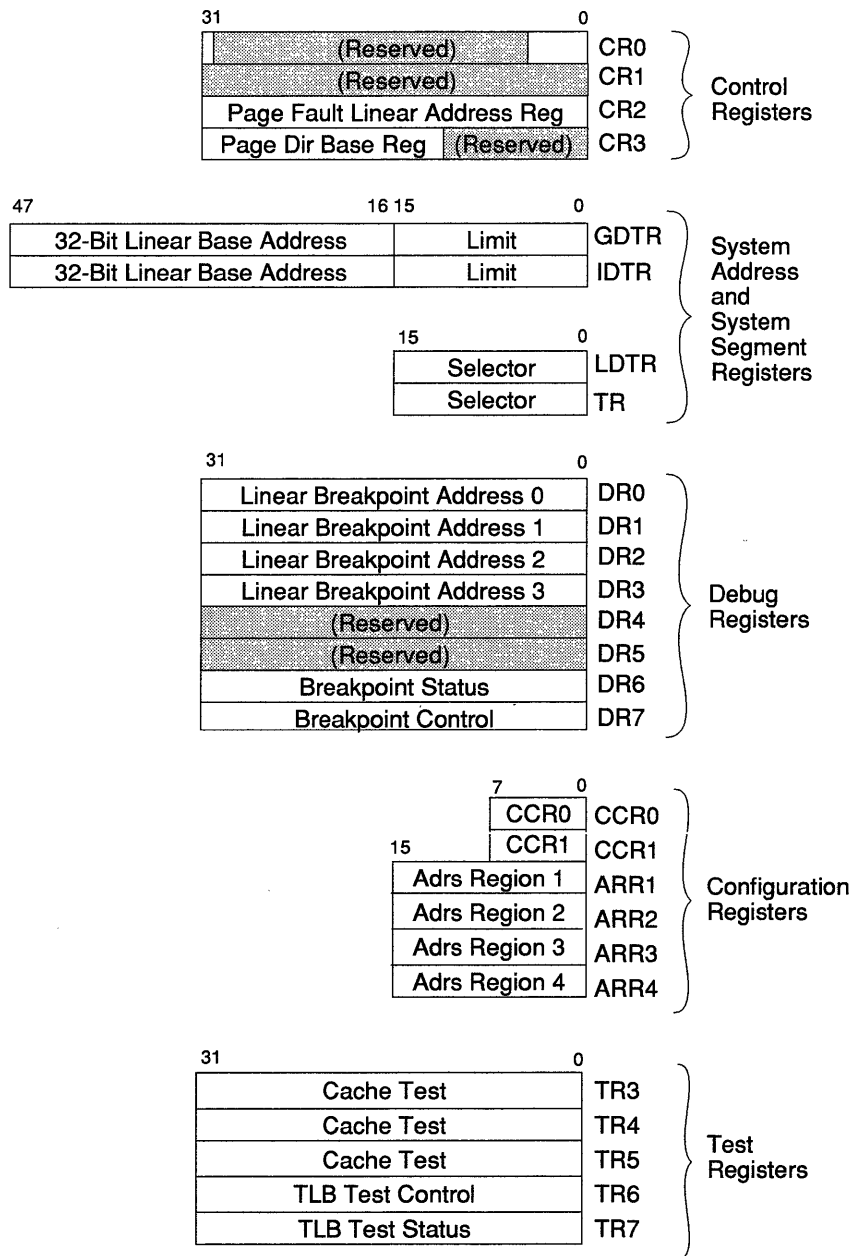


Figure 9-3. Cyrix Cx486SLC system registers.

the course of upgrading an existing 386-based PC design, however, neither of these approaches may be practical.

Instead, the (nonenhanced) Cx486SLC contains four address-region control registers (ARR1 through ARR4) that may be configured to define four arbitrary regions of system memory that

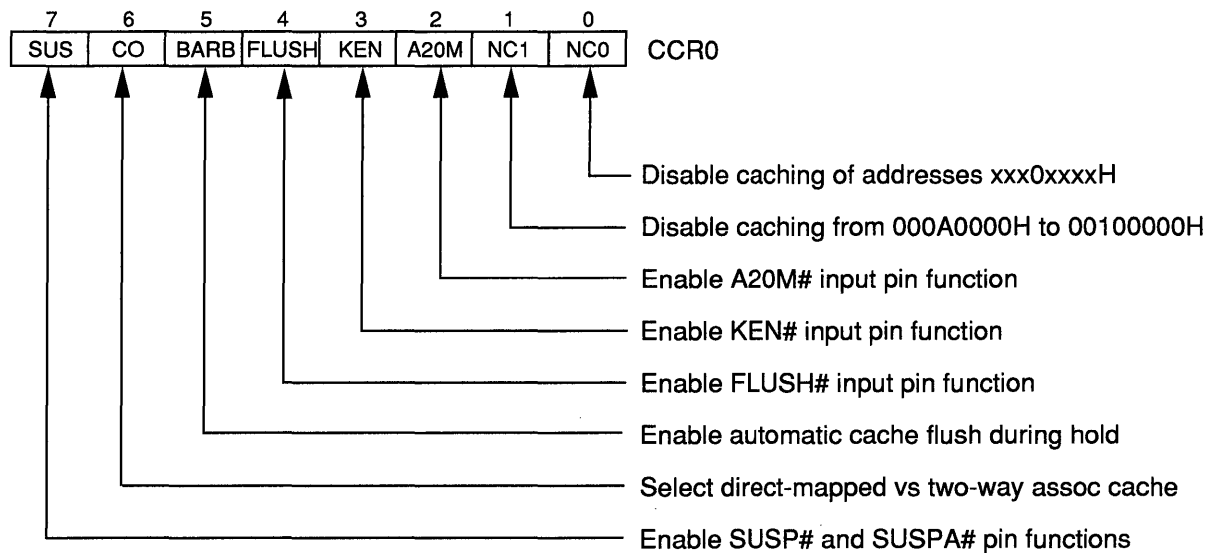


Figure 9-4. Cyrix Cx486SLC/e configuration control register 0.

the CPU should consider to be noncacheable. Each register contains a 16-bit value. The low-order four bits define the size of the noncacheable region to be any power of two from 4K bytes to 32M bytes. The high-order 12 bits of each register determine the starting address of the associated region, defined by multiplying the selected block size by any integer value from 0 to 4095.

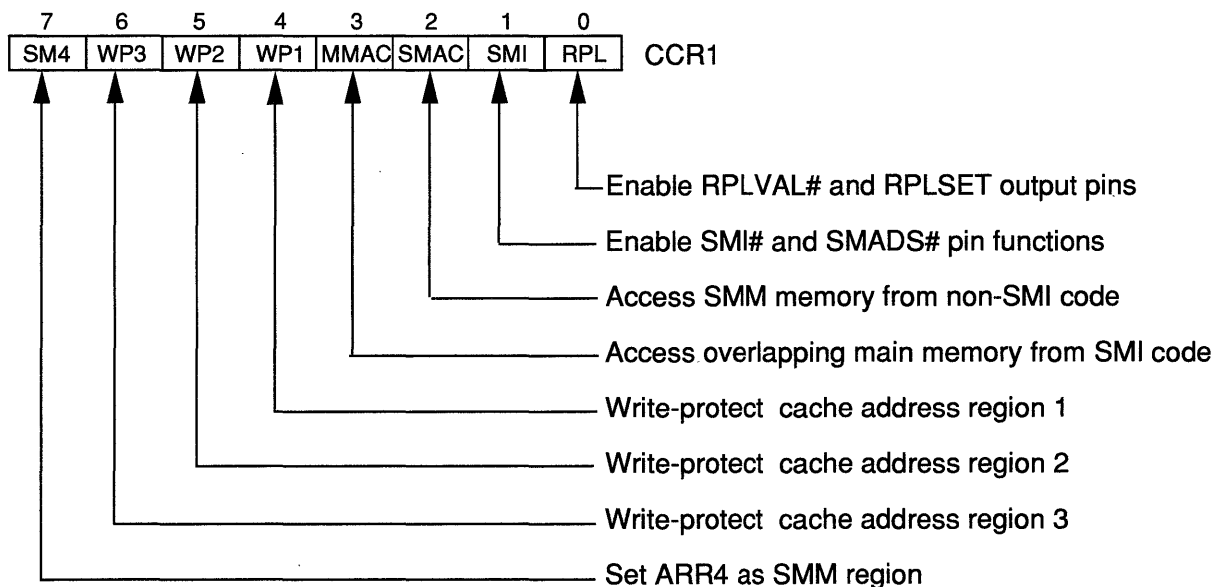


Figure 9-5. Cyrix Cx486SLC/e configuration control register 1.

(In the case of the Cx486SLC/e, registers ARR1 through ARR3 define noncacheable memory regions, and ARR4 may be used either to define a fourth noncacheable region or to define the starting address and block size of the SMM memory region.)

## System Interface

The Cx486SLC/e system interface is a curious hybrid of three existing standards plus a few unique signals. The Cx486SLC/e supports each of the basic bus-interface signals originally defined by the i386SX, but the presence of an on-chip cache requires several additional signals similar to those of the i486SX. The Cyrix SMM hardware interface takes its cue from the AMD Am386SXLV and Am486DXL designs, while still more signals control unique aspects of the Cyrix cache interface and clock. Figure 9-6 illustrates the system interface defined by the Cx486SLC/e.

Table 9-6 summarizes the names and functions of each of the Cx486SLC/e signals not provided by the i386SX. Each of the new signals defined for the Cx486SLC/e replaces a pin originally defined as a no-connect by the i386SX pinout.

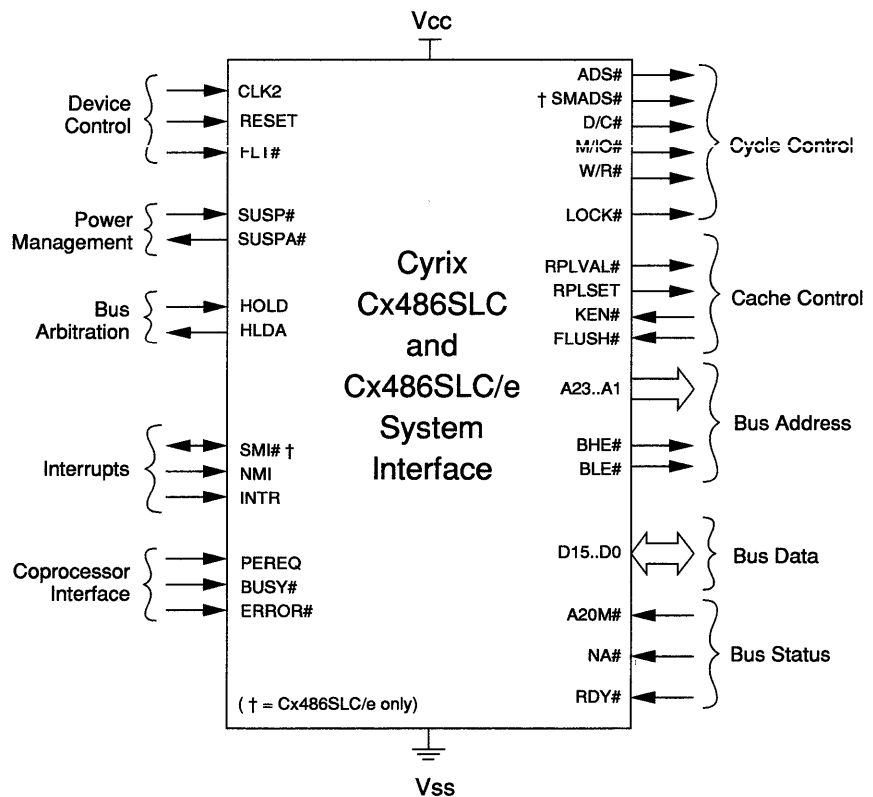


Figure 9-6. Cyrix Cx486SLC and Cx486SLC/e system interface.

Symbol	Direction	Signal Name/Function	PQFP Pin #	Replaces i386SX Signal
A20M#	In	Address-bit 20 mask	31	N.C.
KEN#	In	Cacheability enabled for requested data	29	N.C.
FLUSH#	In	Flush cache data	30	N.C.
RPLVAL#	Out	Cache line replacement valid	46	N.C.
RPLSET	Out	Cache replacement set selected	45	N.C.
SMI#	I/O	SMM interrupt request/active	47	N.C.
SMADS#	Out	SMM memory address strobe	20	N.C.
SUSP#	In	Suspend normal execution	43	N.C.
SUSPA#	Out	Suspend mode acknowledge	44	N.C.

Table 9-6. Cyrix Cx486SLC/e special interface signals.

As in the 486, A20M# allows external circuitry to force the processor to mask address bit 20 for internal cache look-up and external bus writes. KEN# and FLUSH# are also 486-compatible signals; KEN# allows external circuitry to control whether or not data being read by the processor is cacheable, and FLUSH# causes the entire contents of the on-chip cache to be invalidated. The functions of A20M#, FLUSH#, and KEN# are optional, and are enabled by setting bits in control register CCR0.

RPLVAL# and RPLSET, which are not present on the 486, allow external circuitry to deduce where data is being stored in the internal cache. RPLVAL# indicates that a cache line is being replaced and that signal RPLSET is valid, while RPLSET indicates in turn which of the two sets is overwritten during a cache-line replacement. These signals make it possible for systems with second-level caches to keep track of the contents of the on-chip first-level cache.

SUSP# and SUSPA# form a suspend request/acknowledge handshake pair; these signals are further discussed below.

There are two signals on the Cx486SLC/e associated with SMM that perform the same functions as equivalent signals in the Am386SXLV. Asserting SMI# activates SMM. SMI# is bidirectional; the processor continues to hold the pin asserted while operating in SMM.

The processor then asserts SMADS# at the start of a bus cycle to indicate that it is accessing the SMM address space. The SMM address space is configured by on-chip configuration registers, and it can range from 4K-bytes to 4G-bytes. While operating in

SMM, any processor access to the SMM address space causes SMADS# to be asserted; accesses outside this range cause the normal ADS# to be asserted. SMM accesses are not cached. While an SMM interrupt routine is executing a control-register bit may be set in order to let the CPU access regions of main memory that overlap the SMM address space.

The SM# pin also allows trapping of I/O accesses, which is useful to detect accesses to peripherals that power-management software has turned off. If SM# is asserted at least three CLK2 edges before READY# is asserted, then the processor enters SMM and jumps to the system management interrupt handler. The address of the I/O instruction that caused the trap is pushed on the stack, allowing power-management software to re-execute the instruction after power-management software has re-enabled the powered-down peripheral.

(Two of the cache-related signals defined for the Intel 486 are *not* supported by the Cx486SLC. These are the PCD and PWT pins. In an Intel or AMD 486 device, these signals inform an external second-level cache of the state of the cache-disable and write-through attribute bits corresponding to the memory page being addressed. These signals would have little if any value in systems derived from earlier 386-era designs.)

### **Clocking Regimes**

Since the Cx486SLC/e is designed to operate in existing 386SX sockets, the on-chip clock circuit is designed to be compatible with that of existing 386SX motherboard designs. An external clock input must be supplied to the CLK2 input; the frequency of this input is divided by two to determine the frequency at which the core and the bus interface operate.

### **Relative Performance**

According to Cyrix, the Cx486SLC delivers between 2.2 and 3.2 times the performance of a 386SX device when executing common PC benchmarks such as Landmark V2.00, Norton SI V6.0, and the Ziff-Davis processor test. On these same benchmarks, Cyrix says the part is between 79% and 99% as fast as a “true” 486SX, all normalized for core clock rate.

The synthetic benchmarks cited by Cyrix are quite small, however, and thus have high hit rates in the Cx486SLC’s relatively small cache. While these factors may indicate the peak performance of the Cyrix core, they do not represent the performance most users will see. On application-level benchmarks, Cyrix says Cx486SLC performance is just 1.4 to 1.6 times that of a 386SX, and just 60% to 90% of a 486SX. The 486SX design, of

course, has the advantage of a slightly faster core, an eight-times-larger cache, and a 32-bit burst-mode bus interface.

**Package and Pinout**

The Cx486SLC/e is packaged in a standard 100-lead PQFP package. Figure 9-7 illustrates the device pinout.

**Vital Statistics**

The Cx486SLC/e is implemented in a 0.8-micron CMOS technology and integrates about 600,000 transistors on a 410 × 410 mil die (about 168,000 mil<sup>2</sup>). While smaller than Intel's original 1.0-micron 486 design, the Cyrix die is 30% larger than Intel's current 0.8-micron, three-level-metal i486DX implementation—despite its lack of an FPU and its much smaller on-chip

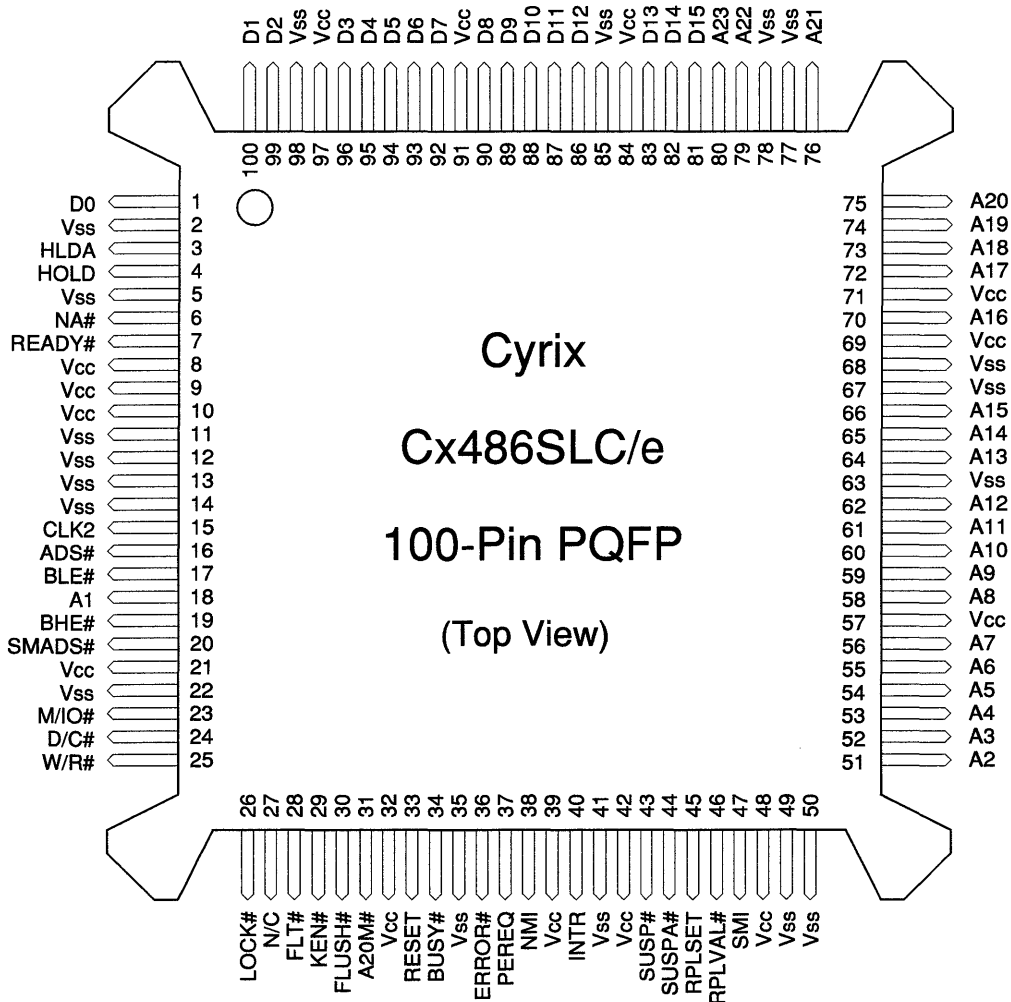


Figure 9-7. Cyrix Cx486SLC/e PQFP pinout.

cache. The larger die size of the Cyrix chip is due to its use of a two-level-metal process (one less than Intel's) and a less rigorously compacted design.

Cyrix offers the Cx486SLC/e in 25- and 33-MHz versions. Production of the original, nonenhanced device has been discontinued.



## 9.3 The Cyrix Cx486SLC/e-V Microprocessor

The Cx486SLC/e-V is a low-voltage version of the Cx486SLC/e. Table 9-7 summarizes the general specifications of the part.

Product Names	Cyrix Cx486SLC/e-V
Introduction Date	November 1992
Prognosis	Stable
Device Integration Level	Same as Cx486SLC/e
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix SMM extensions
Core Technology	Same as Cx486SLC/e
Pinout	Same as Cx486SLC/e
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as Cx486SLC/e
Cache Support	Same as Cx486SLC/e
Floating-Point Support	Optional external Cx87SLC or i387SX FPU
Operating Voltage	3.0 V to 3.6 V
Frequency Options	20- or 25-MHz core operation
Clocking Regime	Core operating frequency = 1/2 × Clkin
Active Power Dissipation	0.94 W (worst case) @ 3.3 V and 25 MHz
Power-Control Features	Cyrix SMM extensions Stopped-clock and suspend-mode operation
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils × 410 mils (10.5 mm × 10.5 mm)
Transistor Count	600,000 transistors
Package Options	100-pin PQFP
Notes	Low-voltage binning of Cx486SLC/e die

Table 9-7. Cyrix Cx486SLC-V feature summary.

Aside from its lower supply- and I/O-pin voltages, the system interface of the Cx486SLC/e-V matches that of the original, non-“V” version. Because of its lower supply voltage, active current drain is reduced to just 285 mA (worst case) at 25 MHz. In suspend-mode with the CLK2 input stopped, the device typically draws just 300  $\mu$ A.

### Vital Statistics

The Cx486SLC/e-V uses the same chip design, with the same die size, complexity, and manufacturing process, as the Cx486SLC/e. Cyrix offers the part in an Intel-compatible 100-pin PQFP package in both 20-MHz and 25-MHz versions.

## 9.4 The Cyrix Cx486SLC2 Microprocessors

The Cx486SLC2 microprocessor is a clock-doubled implementation of the Cx486SLC/e. Table 9-4 summarizes the general features and specifications of this device.

Product Names	Cyrix Cx486SLC2
Introduction Date	October 1993
Prognosis	Encouraging
Device Integration Level	Same as Cx486SLC/e plus clock-doubling circuitry
CPU Architecture Level	Standard 486 integer instruction set
Core Technology	Same as Cx486SLC/e
Pinout	Augmented compatible i386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as Cx486SLC/e
Cache Support	1K-byte unified I- and D-cache Direct or two-way set associative Write-through operation only
Floating-Point Support	Optional external Cx87SLC, Cx3S87 or i387SX FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	50-MHz core operation
Clocking Regime	Core operating frequency = Clkin $\times$ 1
Active Power Dissipation	3.6 W @ 5.0 V and 50 MHz core freq (worst case)
Power-Control Features	Cyrix SMM extensions Stopped-clock and suspend-mode operation
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils $\times$ 410 mils (10.5 mm $\times$ 10.5 mm)
Transistor Count	600,000 transistors
Package Options	100-pin Metal QFP

Table 9-8. Cyrix Cx486SLC2 feature summary.

The Cx486SLC2 combines the best of both worlds for notebook PCs: the small package and standard 25-MHz bus interface simplify system design and minimize board area, while the 50-MHz core frequency delivers high performance while running from cache.

### Vital Statistics

The Cx486SLC2 is supplied only in a 100-lead metal QFP package compatible with PQFP dimensions, and is rated for operation at core frequencies up to 50 MHz. With the clock stopped in suspend mode, current requirements typically drop to 0.1 mA.

## 9.5 The Cyrix Cx486DLC Microprocessor

The Cx486DLC is device that combines the core logic and cache capabilities of the Cx486SLC with a 386DX-class pinout and package. By default, the device is socket-interchangeable with the i386DX, but software can optionally enable its 1K-byte on-chip cache and other new hardware features. Table 9-9 summarizes the general features and specifications of the Cx486DLC microprocessor.

Product Name	Cyrix Cx486DLC
Introduction Date	June 1992
Prognosis	Deceased; reincarnated as the TI486DLC
Device Integration Level	Same as Cx486SLC (non-"/e" version)
CPU Architecture Level	Same as Cx486SLC
Core Technology	Same as Cx486SLC
Pinout	Augmented compatible i386DX PGA pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i386DX
Cache Support	Same as Cx486SLC
Floating-Point Support	Optional external Cx87DLC or i387DX FPU
Operating Voltage	4.75 V to 5.25 V
Frequency Options	33- or 40-MHz core operation
Clocking Regime	Core operating frequency = 1/2 × Clkin
Active Power Dissipation	3.5 W (worst case) @ 5.0 V and 40 MHz
Power-Control Features	Stopped-clock and suspend-mode operation
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils × 410 mils (10.5 mm × 10.5 mm)
Transistor Count	600,000 transistors
Package Options	132-pin PGA
Notes	Uses same die as Cx486SLC

Table 9-9. Cyrix Cx486DLC feature summary.

### System Interface

The Cx486DLC system interface resembles very closely that of a standard 386DX, with the addition of the signals present on a (nonenhanced) Cx486SLC. Figure 9-8 illustrates the system interface used by the part.

Table 9-10 summarizes the names and functions of Cx486DLC signals not defined for the standard 386DX pinout.

Symbol	Direction	Signal Name/Function	PGA Pin #	Replaces i386DX Signal
A20M#	In	Address-bit 20 mask	F13	N.C.
KEN#	In	Cacheability enabled for requested data	B12	N.C.
FLUSH#	In	Flush cache data	E13	N.C.
RPLVAL#	Out	Cache line replacement valid	C7	N.C.
RPLSET	Out	Cache replacement set selected	C6	N.C.
SUSP#	In	Suspend normal execution	A4	N.C.
SUSPA#	Out	Suspend mode acknowledge	B4	N.C.

Table 9-10. Cyrix Cx486DLC special interface signals.

Each of these signals performs the same function as on a Cx486SLC device. Consult the preceding sections for details.

**Relative Performance**

On integer benchmarks, the Cx486DLC is significantly faster than a (noncached) 386DX device at the same clock rate, but

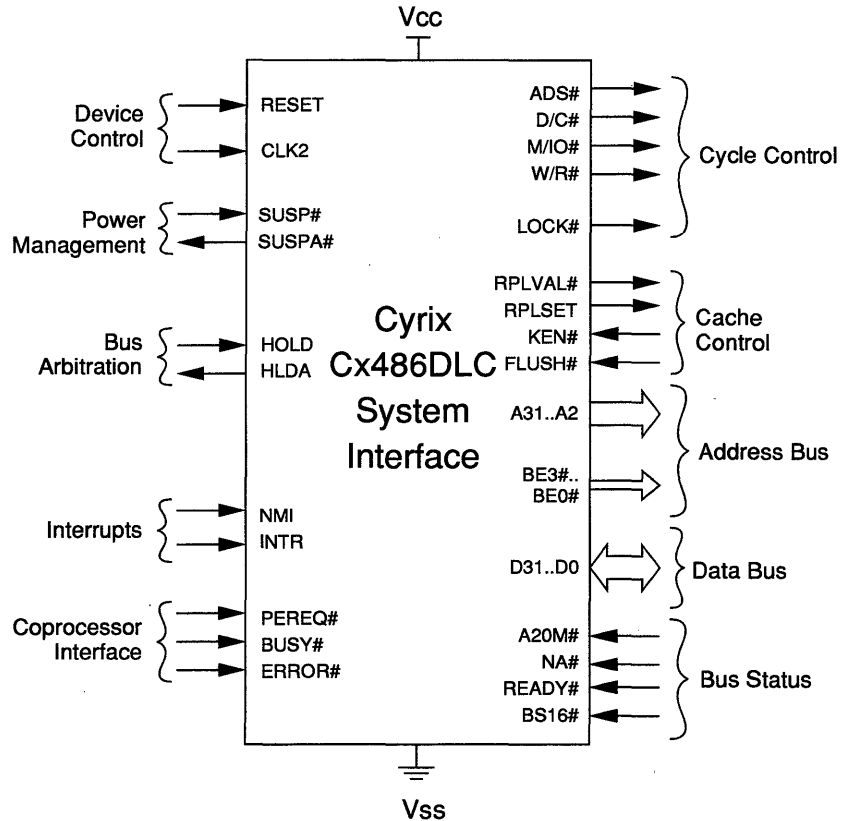


Figure 9-8. Cyrix Cx486DLC system interface.

somewhat slower than a 486SX. Cyrix initially attempted to counter this imbalance by setting the price of its parts such that OEMs would pay the same amount for an Intel 486 or for a Cyrix product of the next higher frequency, which would seem to make the price/performance issue come out a wash.

While running the Cyrix part at a higher speed may on the surface seem to ameliorate any performance differences due to the less efficient Cyrix implementation, designers should realize that running a CPU at higher clock rate to achieve a desired performance level will likely have ramifications on the cost and complexity of the system logic, motherboard design, and DRAM cost.

### **Mortal Statistics**

Cyrix discontinued shipments of the Cx486DLC in late 1993. Until that fateful day, the Cx486DLC contained the same die as the Cx486SLC, with the same manufacturing process, die size, and transistor count. Prior to its discontinuation, Cyrix offered the part in an Intel-compatible 132-pin PGA package, in both 33- and 40-MHz core-frequency versions.

## 9.6 The Cyrix Cx486SRx<sup>2</sup> Microprocessor

The Cx486SRx<sup>2</sup> is an aftermarket processor module designed to upgrade existing 386SX-based PCs to deliver performance levels closer to those of a “true” 486. Table 9-11 summarizes the general features and specifications of the Cx486SRx<sup>2</sup> microprocessor.

Product Name	Cyrix Cx486SRx <sup>2</sup>
Introduction Date	December 1993
Prognosis	Good
Device Integration Level	Pipelined 32-bit IEU and PMMU 1-Kbyte unified instruction/data cache Proprietary cache consistency logic Clock stabilization and frequency-doubler circuitry
CPU Architecture Level	Standard 486 integer instruction set
Core Technology	Same as Cx486SLC
Pinout	Module attaches to standard 386SX package
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#..BLE#)
Data-Transfer Modes	Same as standard i386SX
Cache Support	1K bytes unified I- and D-cache Two-way set associative Write-through operation only
Floating-Point Support	Optional external Cx83S87 or i387SX FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	32-, 40-, or 50-MHz core operation
Clocking Regime	Core operating frequency = 1 × C <sub>kin</sub> (2× bus interface clock)
Active Power Dissipation	N.A.
Power-Control Features	None
Process Technology	0.8μ two-layer-metal CMOS
Die Size	410 mils × 410 mils (10.5 mm × 10.5 mm)
Transistor Count	600,000 transistors
Package Options	Custom module clips onto 100-lead PQFP
Other Features	On-chip buffers accelerate I/O operations Cache logic preconfigured to be compatible with existing 386SX system hardware
Notes	Designed for field upgrades of 386SX PCs

Table 9-11. Cyrix Cx486SRx<sup>2</sup> feature summary.

By the time Cyrix entered the x86-compatible microprocessor market, tens of millions of 386-based PCs were already in use worldwide. These systems—or more specifically, the CPU sock-

ets within them—created a natural and potentially lucrative market opportunity for Cyrix's 386-pinout products. The company pursued this market by combining one of its standard 486 CPUs with a discrete clock-doubler circuit and cache control logic on a small daughterboard module. Cyrix initially offered this module to a fairly well defined market: corporate computing sites that had hundreds of 386 systems from IBM and Compaq that were in desperate need of a cost-effective upgrade.

### Product Overview

The Cx486SRx<sup>2</sup> is essentially a single-chip implementation of the earlier multiple-chip daughterboard. It is based on the same core, cache, and bus-interface circuitry as the Cx486SLC. Since the device is intended for in-the-field upgrades of existing systems, though, it must operate within a rather formidable set of design constraints.

Key among these is the need for maintaining cache coherency in systems not designed for processors with an on-chip cache. In a system environment, an attached processor (typically a DMA controller used for floppy-disk transfers) may modify shared system memory. If the modified region includes locations currently within the CPU cache, the corresponding cached lines must be flushed (i.e., marked as invalid).

Unfortunately, existing PC motherboards provide no mechanism by which the CPU can be informed when system memory is altered—nor should existing systems have any reason to do so. Cyrix thus developed a proprietary scheme for recognizing when the cache should be flushed. While Cyrix has not divulged the exact circuit design, the logic likely detects certain patterns of events, such as extended Hold or Wait periods, or I/O operations performed to port addresses that correspond to DMA controllers in a standard PC environment.

A second constraint is imposed by the need for the Cx486SRx<sup>2</sup> to be driven by an input clock signal of indeterminate characteristics. Cyrix claims to have included clock stabilization circuitry that “cleans up” irregular clock duty cycles sufficiently to drive an on-chip clock-frequency doubler.

A third constraint is imposed by the bottleneck between the CPU and system I/O ports. During the approximately one-microsecond period required for an ISA-bus system to complete an input or output instruction, a 50-MHz 486 core could potentially execute up to 50 new instructions. Cyrix literature makes some fuzzy allusions to additional proprietary circuitry that lets I/O operations complete in parallel with the execution of ensu-

ing instructions. Presumably this circuitry includes write buffers for data written to an output port, and possibly an accumulator scoreboarding and bypass mechanism for delayed posting of data read from a port.

A final constraint is imposed by the fact that most 386SX processors are surface-mount-soldered directly to a PC motherboard, and cannot be readily removed or replaced. For this, Cyrix supplies the Cx486SRx<sup>2</sup> in a very cleverly designed custom module that clips over a standard PQFP package and makes contact with the pins of the original CPU. The clip drives the 386SX FLT# pin permanently active, disabling the pin drivers of the original device.

One problem with this approach is that the clip-on module requires a one-inch clearance above the chip for a heat sink and airflow. This mechanically precludes use of the device in some desktop systems and most portables. Moreover, some of the early 16-MHz i386SX devices did not recognize FLT#. Short of having these chips unsoldered from the motherboard, systems based on these parts cannot be upgraded. Ironically, neither can systems in which the original 386SX PQFP device is itself held in a socket, rather than being soldered to the motherboard, although such systems are rare.

### **Frequency Options**

Cyrix offers the Cx486SRx<sup>2</sup> in a single, 50-MHz core-frequency version for use in 16-, 20-, or 25-MHz 386-based PC designs.



## 9.7 The Cyrix Cx486DRx<sup>2</sup> Microprocessor

The Cx486DRx<sup>2</sup> is, as one might suppose, an aftermarket upgrade processor for 386DX-based PCs, and is designed to be a direct pin-for-pin replacement for the i386DX. Table 9-12 summarizes the general features and specifications of the Cx486DRx<sup>2</sup> microprocessors.

Product Name	Cyrix Cx486DRx <sup>2</sup>
Introduction Date	August 1993
Prognosis	Good
Device Integration Level	Pipelined 32-bit IEU and PMMU 1K-byte unified instruction/data cache Proprietary cache consistency logic Clock stabilization and frequency-doubler circuitry
CPU Architecture Level	Standard 486 integer instruction set
Core Technology	Cyrix-designed static 486 core
Pinout	Standard 386DX PGA pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Two cycles minimum per 32-bit transfer One-half cycle address pipelining optional Dynamic bus resizing for 16-bit transfers
Cache Support	1K bytes unified I- and D-cache two-way set associative Write-through operation only
Floating-Point Support	Optional external Cx87DLC or i387DX FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	32-, 40-, 50-, or 66-MHz core operation
Clocking Regime	Core operating frequency = 1 × Clkin (2× bus interface clock)
Active Power Dissipation	N.A.
Power-Control Features	None
Process Technology	0.8μ two-layer-metal CMOS
Die Size	410 mils × 410 mils (10.5 mm × 10.5 mm)
Transistor Count	600,000 transistors
Package Options	Standard 132-pin PGA
Other Features	On-chip buffers accelerate I/O operations Cache logic preconfigured to be compatible with existing 386DX system hardware
Notes	Designed for field upgrades of 386DX PCs

Table 9-12. Cyrix Cx486DRx<sup>2</sup> feature summary.

The Cx486DRx<sup>2</sup> is derived from the Cx486DLC processor core. Aside from providing a 32-bit data-bus interface and a full 32 bits of address, the part operates in a manner analogous to the Cx486SRx<sup>2</sup> described. One difference is that most 386DX-based systems contain PGA-packaged parts which can be removed from a socket on the motherboard and replaced by the Cx486DRx<sup>2</sup>. Whereas the Cx486SRx<sup>2</sup> can be used only with PQFP parts, the Cx486DRx<sup>2</sup> device cannot be used with those rare systems that have a PQFP-packaged 386DX device soldered to the motherboard. Go figure.

**Frequency Options**

Cyril offers versions of the Cx486DRx<sup>2</sup> that support core frequencies of 32, 40, 50, or 66 MHz, respectively, for use in upgrading 16-, 20-, 25, or 33-MHz 386-based PC designs.

## 9.8 The Cyrix Cx486S and Cx486S2 Microprocessors

The Cx486S family is Cyrix's entry-level pin-compatible replacement for 486SX-class devices. The Cx486S2 is a Cx486S with an on-chip clock-frequency doubler. The Cx486S-V is a 3.3 V implementation of the Cx486S. Table 9-13 summarizes the general features and specifications of each of these products.

Product Name	Cyrix Cx486S and Cx486S2
Introduction Date	May 1993
Prognosis	Deceased
Device Integration Level	Pipelined 32-bit IEU and PMMU Microcoded 80-bit floating-point unit 2K-byte unified instruction/data cache
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix SMM extensions
Core Technology	Standard Cyrix 486 core
Pinout	Augmented compatible 486SX pinout
Data Bus Width	32 bits with parity (D31..D0 plus DP3..DP0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486SX, plus optional burst-mode data write capability
Cache Support	2K bytes unified I- and D-cache Four-way set associative Write-through or copy-back operation
Floating-Point Support	Optional off-chip Cx487S FPU
Operating Voltage	Cx486S, Cx486S2: 4.75 V to 5.25 V Cx486S-V: 3.0 V to 3.6 V
Frequency Options	Cx486S: 33-, 40-, or 50-MHz core operation Cx486S2: 50-MHz core operation Cx486S-V: 25- or 33-MHz core operation
Clocking Regime	Cx486S, Cx486S-V: Core operating freq = 1 × Clkin Cx486S2: Core operating freq = 2 × Clkin
Active Power Dissipation	Cx486S, Cx486S2: 4.45 W (worst case) @ 5.0 V and 50 MHz (core freq) Cx486S-V: 1.27 W (worst case) @ 3.3 V and 33 MHz
Power-Control Features	Stopped-clock and suspend-mode operation
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	112 mm <sup>2</sup>
Transistor Count	700,000 transistors
Package Options	168-pin PGA or 196-pin Metal QFP

Table 9-13. Cyrix Cx486S and Cx486S2 feature summary.

The Cx486S-family processors are based on the same core CPU as the Cx486SLC and Cx486DLC, but with double the amount of on-chip cache, a higher-bandwidth system interface, and optional clock-doubling capability.

### **Cache Characteristics**

Although the 2K-byte on-chip cache is just twice as large as that of the Cx486SLC, two factors enhance its efficiency considerably. First, the Cx486S-family cache uses a four-way set-associative organization vs two-way for earlier Cyrix parts. A standard cache-design rule of thumb states that doubling the set associativity of a cache with a given size should improve its hit rate about as much as doubling its raw capacity with the same set associativity. Applying this rule, the 2K-byte, four-way cache used by the Cx486S should have as high a hit rate as a (hypothetical) 4K-byte, two-way set-associative design.

Second, the Cx486S-family cache supports an optional copy-back protocol that is potentially more efficient than the write-through design used by earlier Cyrix and Intel designs.

The cache line size is 16 bytes; there is one valid bit per line and one dirty bit per 4 bytes. Line replacement uses a pseudo-least-recently-used (LRU) algorithm. When a dirty cache line must be reallocated, the entire line need not be copied back to memory; only the modified 32-bit words must be written. Cache lines are not allocated on writes.

The Cx486S-series cache also includes a “no-lock” feature, not supported by Intel, that caches the contents of protected-mode segment registers, speeding later accesses to cache descriptors. Cyrix claims this one feature can improve protected-mode performance by up to 5%. A minor change must be made to the system initialization code, either in the BIOS ROMs or via an application-level configuration program, in order to enable the no-lock feature.

Still, the Cx486S-series cache is just one-fourth the size of that in the i486SX, so inevitably it has a lower hit rate. Given that the Cx486SLC CPU core is also slightly slower than Intel’s, these parts deliver somewhat lower performance than Intel’s, and must run at a higher core frequency to achieve the same overall throughput. For example, the 40-MHz Cx486S scores about as well on most PC benchmarks as a 33-MHz 486SX.

### **Floating-Point Strategy**

Intel 486SX-type products provide no direct hooks for an external FPU. Instead, the main CPU must be entirely disabled and replaced by a fully featured 486DX-class upgrade processor.

In contrast, the Cx486S product line follows the more conventional approach of augmenting its integer CPU with a separate Cx487S floating-point coprocessor chip. Like the Cx486S, the Cx487S has a static design, supports a suspend mode, and is also available in a 3.3-V version. The device also reduces power consumption automatically when not in use.

It's therefore quite cost-effective to upgrade a Cx486S-based design to include floating-point support, since the Cx487S device is (by today's standards) cheap to fabricate and uses a small, inexpensive 80-pin PQFP package. On the other hand, the chip-to-chip coprocessor interface adds a certain amount of communication overhead to every FPU operation or transfer, restricting FPU performance somewhat. This added overhead is unfortunately most significant for the simplest, quickest, and most prevalent floating-point operations.

### System Interface

Figure 9-9 illustrates the system interface supported by Cx486S-family devices.

Table 9-14 summarizes the names and functions of Cx486S signals not defined by Intel's original (non-SL-enhanced) i486SX pinout.

Symbol	Direction	Signal Name/Function	Cyrix PGA Pin #	Replaces i486SX Signal
WM_RST	In	Warm reset	B13	N.C.
CLKMODE	In	Clock multiplier mode (Cx486S2 only)	B14	N.C.
SMI#	I/O	SMM interrupt request/active	A12	N.C.
SMADS#	Out	SMM memory address strobe	C10	N.C.
SUSP#	In	Suspend normal execution	G15	N.C.
SUSPA#	Out	Suspend mode acknowledge	A10	N.C.
RPLVAL#	Out	Cache line replacement valid	C13	N.C.
RPLSET1	Out	Cache replacement set selected	A13	N.C.
RPLSET0	Out		B15	N.C.
HITM#	Out	Snooping hit on modified value	R17	N.C.
INVAL	In	Invalidate cache value	S4	N.C.
PEREQ	In	Processor extension (FPU) service request	B10	N.C.
BUSY#	In	Busy (FPU coprocessor status)	C12	N.C.
ERROR#	In	Floating-point error detected	C14	N.C.

Table 9-14. Cyrix Cx486S and Cx486S2 special interface signals.

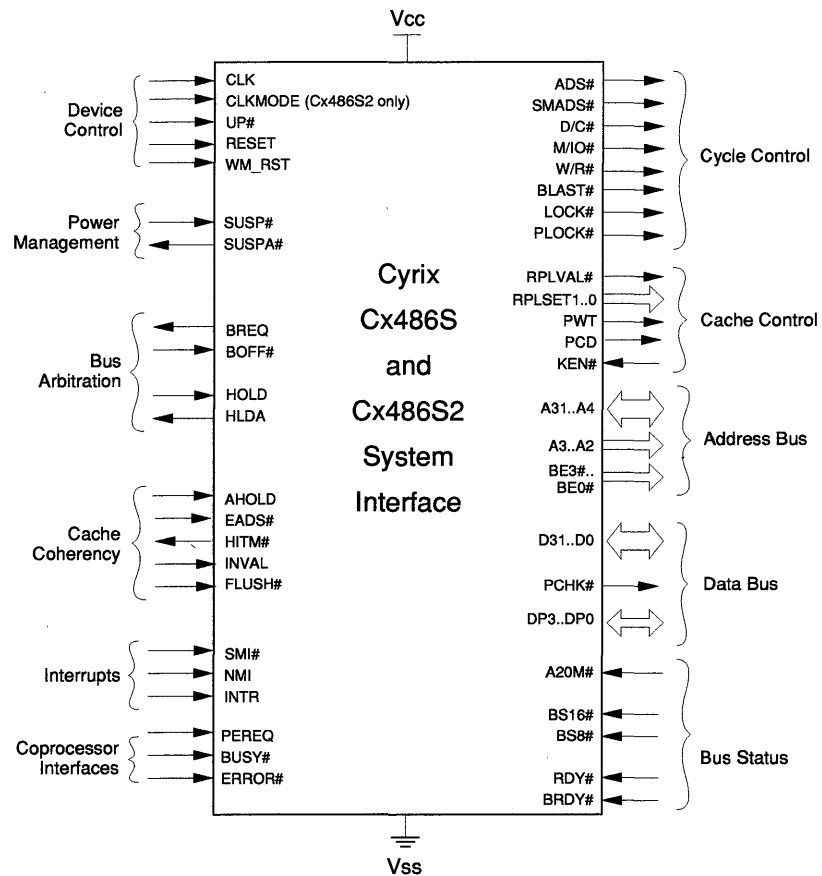


Figure 9-9. Cyrix Cx486S and Cx486S2 system interface.

The Warm Reset (WM\_RESET) signal resets the processor without modifying the device configuration registers, the cache tag and data arrays, or the cache dirty and valid bits. This feature is provided for compatibility with older software that resets the processor to switch from protected to real mode.

The Clock Mode (CLKMODE) signal (present only on Cx486S2 devices) must be strapped to Vcc in order to enable the clock-doubling feature of the Cx486S2. If this pin is grounded or left floating, the clock-doubling circuitry is disabled, so the part operates as a nondoubled Cx486S device, i.e., the core logic and the system bus interface both run at the same frequency as the CLK input signal. A Cx486S2 chip rated for 50-MHz core frequency, for example, can thus be used either as a CPU with a 25-MHz bus clock and a 50-MHz core frequency, or as a conventional 50-MHz part with a 50-MHz bus.

SMI# and SMADS# support the same SMM interrupt request/acknowledge and address-strobe functions on the Cx486S as they do on Cx486SLC-class products.

SUSP# and SUSPA# support the same power-reduction features for the Cx486S as for Cx486SLC-class products. Even the clock-doubled Cx486S2 chips let the clock be stopped, reducing typical power consumption to 2 mW at 5.0 V. The 3.3-V versions reduce power drain even further, typically to just 660  $\mu$ W.

As with the Cx486SLC devices, the RPLVAL# signal informs an external second-level cache when an on-chip cache line is being replaced. But whereas devices with simpler caches could use a single RPLSET pin to identify which line in two different sets was being replaced, the Cx486S needs two such pins (RPLSET1 and RPLSET0) to distinguish among four cache lines. Note that Intel 486 devices lack these signals, and as a result, it is impossible for a second-level cache in an Intel-based design to precisely track the contents of the on-chip cache.

Because the write-back cache can contain data that is more recent than the data in main memory, the Cx486S cache must be consulted whenever an external bus master (such as a DMA controller or a second processor) attempts to read a shared memory region. This is implemented via cache inquiry (“snooping”) cycles, which let an external bus master poll the CPU’s on-chip cache. When a read snoop hit occurs for on-chip dirty data, the Cx486DX must write the data from the cache back to system memory before the auxiliary cache can proceed.

Two new signals implement the basic control for the write-back cache. Hit-Modified (HITM#) is asserted by the processor when a dirty cache line hit occurs during an inquiry cycle, i.e., when a cache-inquiry cycle is in progress and the cache holds modified data for that address.

The Cx486DX implements an “abort and retry” protocol by asserting HITM# and writing the dirty cache line (or the dirty words within the line) to memory. The other bus master then reads the data from memory. This protocol is very similar to that implemented by Pentium. If the Invalidate (INVAL) input signal is active during a cache-inquiry cycle when a hit occurs, the Cx486S will also clear the on-chip Valid bit.

External logic can also force dirty data in the cache to be written to memory by asserting the FLUSH# input. This would be

used prior to stopping the processor clock, for example, since no snooping can occur between clock cycles.

To allow the Cx486S to be used in systems that lack hardware snooping signals, the device may be configured such that dirty cache data will be written to memory whenever the HOLD input is asserted, for example when a DMA controller appropriates use of the bus. The processor will then flush all dirty locations to system memory before asserting HLDA. In order to enable this mode, control bit BARB in configuration register CCR2 must be set by initialization software.

The Cx486S may perform burst-mode writes whenever all four 32-bit words of a cache line are dirty and the line must be flushed or replaced. Burst writes use the same BRDY# control signal to pace the transfer as Intel-standard burst reads. Assuming a three-clock initial transfer and single-cycle transfers within the burst (i.e., 3-1-1-1), burst mode cuts the line write time in half. Burst-mode writes are enabled by setting control bit BWRT in configuration register CCR2.

(Note that since Intel 486 devices have a write-through cache, there is never any dirty data on-chip. Cache inquiry cycles thus need not be performed when an attached processor attempts to read a shared memory space. Cache inquiry cycles need only be performed when an external bus master writes to shared memory, and then the only action required of the cache is that its line-valid bit must be cleared to mark the data as stale.)

The PEREQ, BUSY, and ERROR# signals perform the same external FPU interface functions as on the Cx486SLC device. Refer to the earlier product description for details.

### **Clocking Regimes**

The Cx486S device's static design allows its clock to be slowed or stopped to reduce power. A novel feature of the Cyrix clock-doubler circuit is that it does not employ a phase-locked loop (PLL).

Instead, Cyrix's clock-doubler circuit uses a digital delay circuit that generates a series of pulses after each clock edge. In response to this edge, the Cyrix clock-doubler circuit generates a series of four pulses, with the time between pulses set by an on-chip delay line. Each pulse toggles a flip-flop, which creates the frequency-doubled output. The delay time between pulses is set so that even at the maximum clock frequency, the fourth pulse arrives sufficiently early before the next rising edge on the clock input.



As the clock input is slowed, the spacing of the four pulses remains constant, so only the last half-cycle of every alternate clock cycle is stretched. This stretching does not bother the logic, however, and this circuit allows the clock frequency to be changed dynamically without restriction.

(In a chip that does have an analog PLL clock generator, such as the i486SX2, chip frequency cannot be changed rapidly because the PLL cannot remain locked to a rapidly slewing frequency. This limits the degree to which power-management circuitry can dynamically slow the clock to save power.)

### Instruction Set Additions

The Cx486S family implements the complete standard 486 integer instruction set plus the eight SMM instructions listed in Table 9-15.

Instruction	Mode	Operation	Opcode
SVDC	SMM	Save segment register (DS, ES, FS, GS, or SS) and associated descriptor	0F78H
SVLDT	SMM	Save LDTR and descriptor	0F7AH
SVTS	SMM	Save TSR and descriptor	0F7CH
RSDC	SMM	Restore segment register and descriptor	0F79H
RSLDT	SMM	Restore LDTR and descriptor	0F7BH
RSTS	SMM	Restore TSR and descriptor	0F7DH
RSM	SMM	Resume normal execution mode	0FAAH
SMINT	System	Software-invoked system-management interrupt service routine	0F7EH

Table 9-15. Cyrix Cx486S and Cx486S2 instruction set additions.

The first seven of these instructions are also implemented by the Cx486SLC/e; refer to the instruction set descriptions earlier in this chapter for details.

The SMINT instruction allows operating system software to enter system management mode and invoke the SMI interrupt handler. This instruction can only be executed from within the highest privilege level, i.e., when the current privilege level is 0, and only after initialization software has enabled the SMI interrupt logic.

### Configuration Registers

Like earlier Cyrix 486 products, the Cx486S-family devices implement each of the system control, system segment, debug, and test registers defined for Intel 486 devices. However, the configuration and memory region control registers contained in Cx486SLC-class devices have been supplanted by a slightly dif-

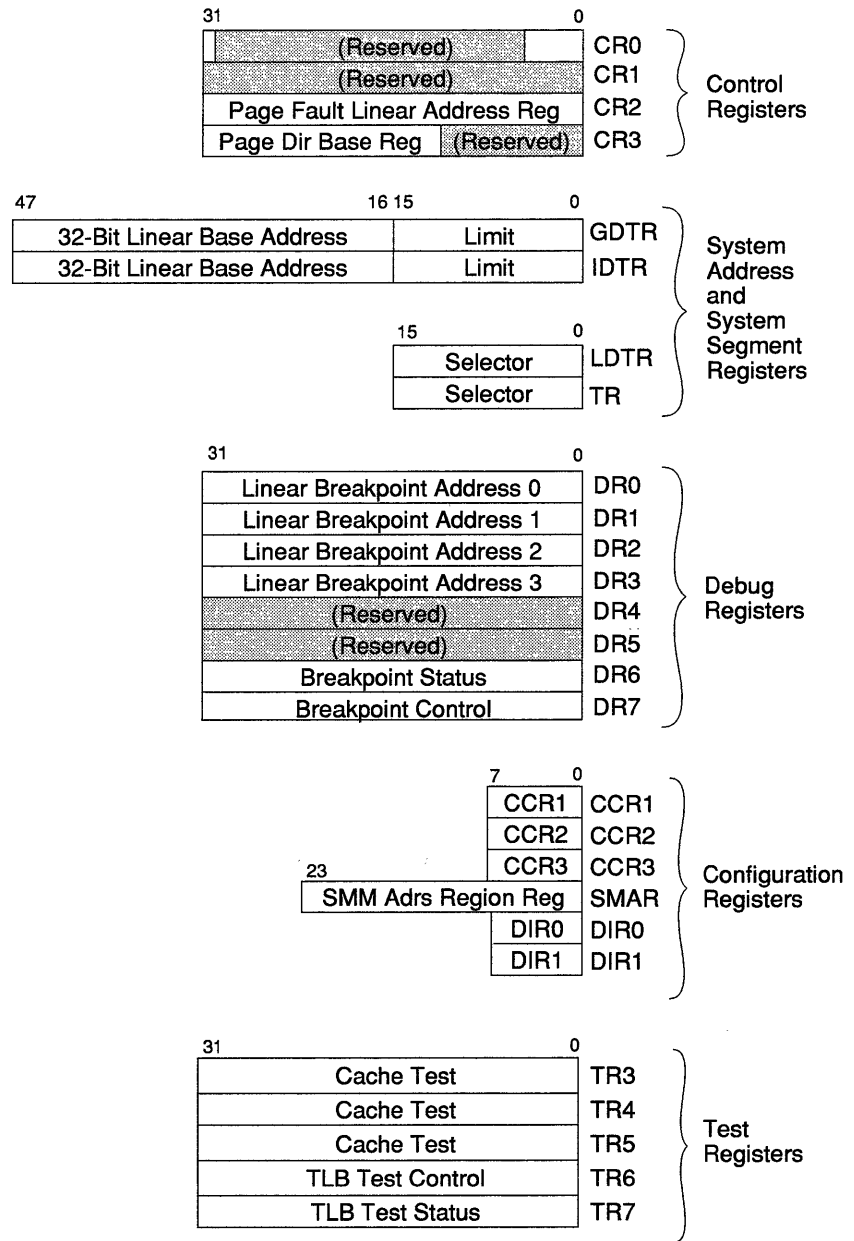


Figure 9-10. Cyrix Cx486S and Cx486S2 system registers.

ferent set of resources. Figure 9-10 shows the complete Cx486S system register set.

The hardware functions of the Cx486S may be enabled by setting control bits in three new Configuration Control Registers, designated CCR1, CCR2, and CCR3. The bit fields within these registers operate as shown in Figures 9-11 through 9-13.

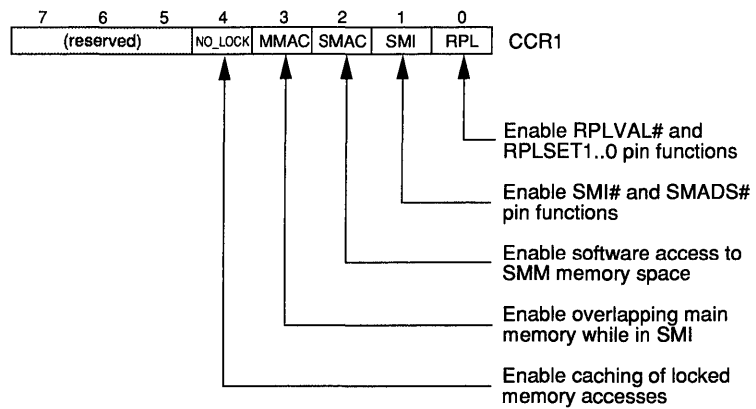


Figure 9-11. Cyrix Cx486S and Cx486S2 configuration control register 1.

The SMM Address Register (SMAR) is a 32-bit register that replaces the function of register ARR4 on the Cx486SLC/e. The SMM memory space may be defined to have a size equal to any power of two from 4KB to 32MB, and may be initialized starting at any block-aligned address anywhere in the 4GB memory space.

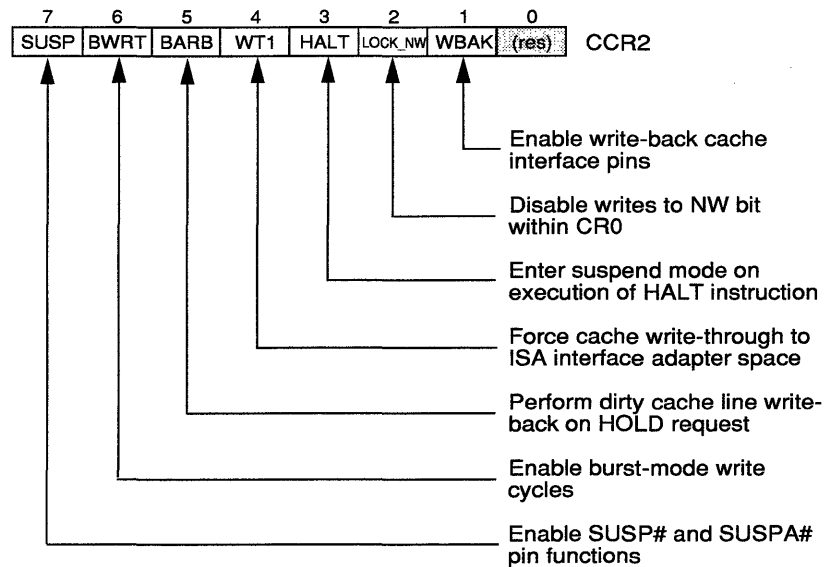


Figure 9-12. Cyrix Cx486S and Cx486S2 configuration control register 2.

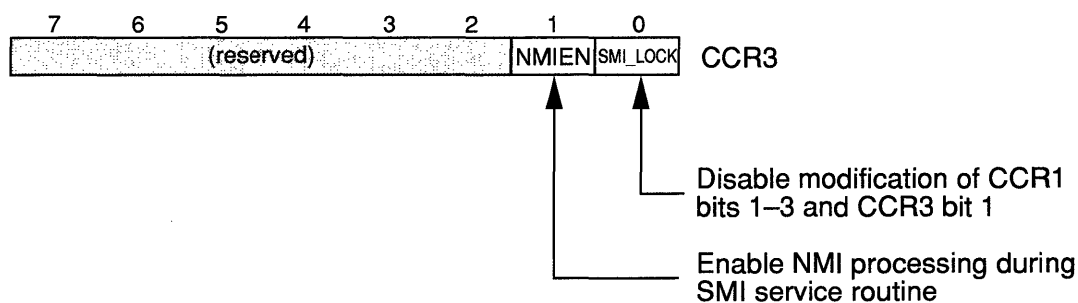


Figure 9-13. Cyrix Cx486S and Cx486S2 configuration control register 3.

Device Identification Registers DIR0 and DIR1 provide two eight-bit values that include fields for device identification, revision number, and stepping. Each register is read-only.

### Vital Statistics

The Cx486S and Cx486S2 are implemented in a 0.8-micron CMOS technology and integrate about 700,000 transistors on a 117 mm<sup>2</sup> die.

Cyrix offers the Cx486S in a standard 168-pin PGA package with operating frequencies of 33, 40, or 50 MHz. In addition, the 33- and 40-MHz versions are available in a 196-lead metal QFP package (MQFP).

The Cx486S2 is housed only in a 168-pin PGA package, and is offered only with a 50-MHz speed rating. Each of the Cx486S and Cx486S2 devices requires a supply voltage between 4.75 V and 5.25 V.

The Cx486S-V device operates on supply voltages between 3.0 V and 3.6 V. It is offered only in the 196-lead MQFP housing, at frequencies of 25 and 33 MHz.

## 9.9 The Cyrix Cx486DX and Cx486DX2 Microprocessors

The Cx486DX is a more-or-less direct replacement for the Intel i486DX. The Cx486DX2 also includes an on-chip clock-frequency doubler. Each is available in both 5-V and 3.3V versions, the latter designated by a “-V” suffix. Table 9-16 summarizes the general features and specifications of the Cx486DX and Cx486DX2 product family.

Product Name	Cyrix Cx486DX/Cx486DX2/Cx486DX-V/Cx486DX2-V
Introduction Date	Fall 1993
Prognosis	Thriving
Device Integration Level	Pipelined 32-bit IEU and PMMU Microcoded 80-bit floating-point unit 8K-byte unified instruction/data cache
CPU Architecture Level	Standard 486 integer and FPU instruction sets, augmented with Cyrix SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 486DX pinout
Data Bus Width	32 bits with parity (D31..D0 plus DP3..DP0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX, plus optional burst-mode data write-back capability
Cache Support	8K bytes unified I- and D-cache Four-way set associative Write-through or copy-back operation
Floating-Point Support	On-chip high-performance microcoded FPU
Operating Voltage	Cx486DX, Cx486DX2: 4.75 V to 5.25 V Cx486DX-V, Cx486DX2-V: 3.0 V to 3.6 V
Frequency Options	Cx486DX: 33-, 40-, or 50-MHz operation Cx486DX2: 50- and 66-MHz core operation Cx486DX-V: 33- or 40-MHz operation Cx486DX2-V: 50-, 66-, or 80-MHz core operation
Clocking Regime	Cx486DX, Cx486DX-V: Core operating freq = 1 × Clkin Cx486DX2, Cx486DX2-V: Core freq = 2 × Clkin
Active Power Dissipation (worst case)	Cx486DX: 5.85 W @ 5.0 V and 50 MHz Cx486DX2: 6.62 W @ 5.0 V and 66 MHz (core freq) Cx486DX-V: 2.24 W @ 3.3 V and 40 MHz Cx486DX2-V: 3.14 W @ 3.3 V and 80 MHz (core freq)
Power-Control Features	Cyrix system-management mode extensions
Process Technology	0.8μ two-layer-metal CMOS
Die Size	476 × 480 mils
Transistor Count	900,000 transistors
Package Options	168-pin PGA or 208-lead plastic QFP

Table 9-16. Cyrix Cx486DX and Cx486DX2 feature summary.

The Cx486DX and Cx486DX2 are based on the same core technology as the Cx486S family. The devices expand the write-back cache to 8K, integrate a math coprocessor on chip, and are compatible with the standard i486DX PGA pinout. The products were developed under the code name "M7."

The devices implement the full 486 integer and floating-point instruction sets, although the underlying microarchitecture still has no dedicated address adder. As a result, although Cyrix's core matches Intel's performance on most register operations, it is one cycle slower on instructions that involve an address computation, including all instructions with memory-based operands, all jumps, and all calls.

In the Cx486DX, the slower speed of the core is partially offset by the write-back cache, which Intel chips lack. To provide fast cache line flushes, the Cx486DX extends Intel's 486 bus protocol with burst-mode writes. The overall performance of the Cx486DX also benefits from Cyrix's faster FPU design.

While the Cx486DX includes an optional clock-doubler, Cyrix has promoted the 50-MHz version for use primarily in non-clock-doubled systems using a VESA local bus. This configuration brings it closest in performance to a 486DX2-66, which has a faster CPU speed but a slower (33-MHz) local bus. The Cx486DX 50-MHz local bus could improve graphics performance somewhat, assuming a display controller that can keep up with that rate, but it remains to be seen how significant this is.

### **Cache Design**

Aside from its larger size, the cache in the Cx486DX and Cx486DX2 has the same characteristics as Cx486S-family devices. See the preceding section for details.

### **Floating-Point Unit**

Cyrix has been in the math coprocessor business for some time, so adding an FPU to the chip was presumably not a fundamentally difficult task for the company. In order to compete with Intel in the 387 market, Cyrix was forced to develop more sophisticated and faster floating-point hardware. The Cx486DX family has been able to capitalize on this technology and experience, and is in fact up to 10% faster at floating-point-intensive applications than the equivalent Intel products. The downside is that the Cyrix FPU core is much larger than Intel's, with a resulting impact on die size and cost (see the *Vital Statistics* section below).

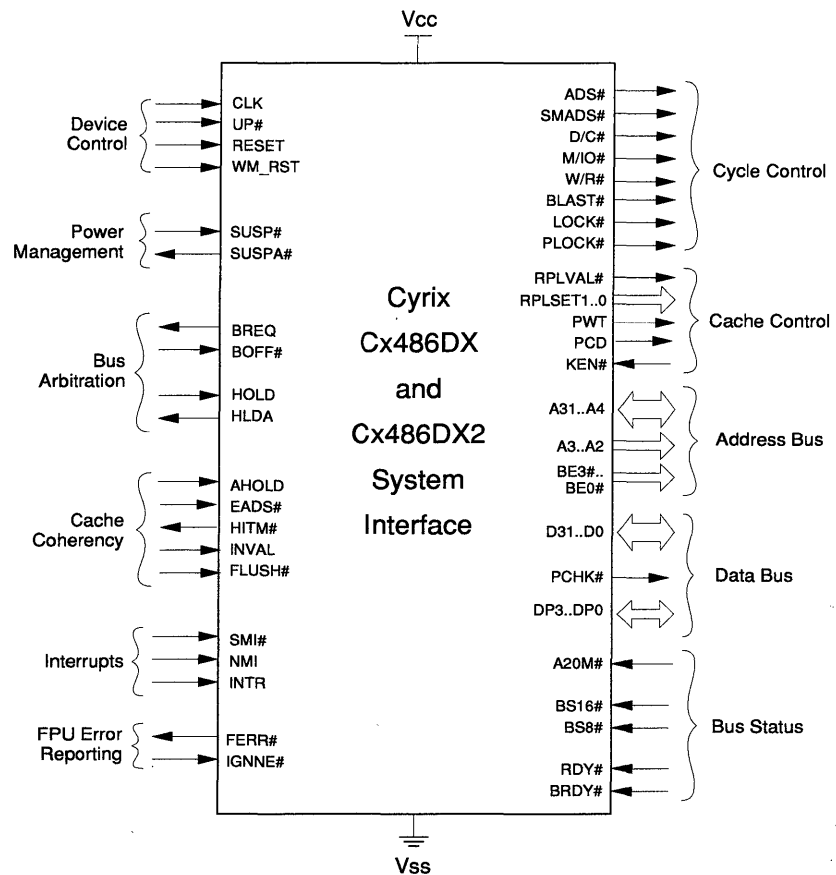


Figure 9-14. Cyrix Cx486DX and Cx486DX2 system interface.

### System Interface

The system interface of the Cx486DX and Cx486DX2 is a superset of that of the i486DX, as shown in Figure 9-14. Table 9-17 lists names and functions of Cx486DX and Cx486DX2 signals that are not present on the original i486DX pinout. Each of these signals performs the same function as for the Cx486S-series products described in the preceding section of this chapter.

### Power Management

Typical current drain is 860 mA at 66 MHz with a 5-V supply, or 1.325 A worst-case. The 3.3-V version typically draws 630 mA at 80 MHz or 950 mA worst case. The wide spread between typical and maximum values is due, in part, to the fact that the FPU is powered down when no FP instructions are being executed. The maximum rating is measured while the FPU is repeatedly executing the FCOS instruction; the typical value is measured while running Whetstone.

Symbol	Direction	Signal Name/Function	Cyrix PGA Pin #	Replaces i486DX Signal
WM_RST	In	Warm reset	B13	N.C.
SMI#	I/O	SMM interrupt request/active	A12	N.C.
SMADS#	Out	SMM memory address strobe	C10	N.C.
SUSP#	In	Suspend normal execution	G15	N.C.
SUSPA#	Out	Suspend mode acknowledge	A10	N.C.
RPLVAL#	Out	Cache line replacement valid	C13	N.C.
RPLSET1	Out	Cache replacement set selected	A13	N.C.
RPLSET0	Out		B15	N.C.
HITM#	Out	Snooping hit on modified value	R17	N.C.
INVAL	In	Invalidate cache value	S4	N.C.

Table 9-17. Cyrix Cx486DX/Cx486DX2 special interface signals.

### Relative Performance

Cyrix claims performance of the Cx486DX is about 9% slower than Intel's 486DX on integer code and 10% faster on floating-point software at a given clock rate, as measured by the PowerMeter MIPS and Whetstone benchmarks. Because these benchmarks fit in the on-chip cache, they yield essentially the same results for a clock-doubled system or for one with a full-frequency system bus. These performance figures are also independent of the second-level cache and memory system.

According to Cyrix, the BAPCo benchmark, which better reflects application-level performance, shows that the Cx486DX provides performance equal to Intel's i486DX2-50 in a cacheless system design, where the on-chip write-back cache is especially valuable. In a system with a 256K second-level cache, the Cx486DX's BAPCo performance falls short of Intel's by 4% in clock-doubled mode and by 7% for full 50-MHz operation.

In a cacheless system, the write-back cache is particularly valuable. Cyrix's measurements show that while Intel's 486 is faster in a system with a second-level cache, the Cx486DX matches the 486 in a cacheless system with fast (3-2-2-2 access pattern) DRAM and outperforms the Intel chip by 5-10% with slower DRAM.

### Vital Statistics

The Cx486DX, Cx486DX2, Cx486DX-V, and Cx486DX2-V devices all use essentially the same die, which contains 900,000 transistors and measures a rather portly  $476 \times 480$  mils ( $228\text{K mils}^2$ ) on a 0.8-micron process—78% larger than Intel's 0.8-micron i486DX! The addition of the FPU is one factor in this die inflation; the Cyrix design is faster than Intel's, and thus more complex and larger. A bigger culprit, though, is the fact



that Cyrix designed the part for a 0.8-micron CMOS process with only two layers of metal compared to Intel's three, and that Cyrix's layout is less dense.

The parts are available in a tremendously wide variety of voltage, frequency, and packaging options. The 5-V Cx486DX is currently offered in a standard 168-pin PGA package at frequencies of 33, 40, or 50 MHz. The 5-V Cx486DX2 is offered in the same package with core frequencies of 50 or 66 MHz.

Cyrix provides a broader range of options in the 3.3-V domain, though. The Cx486DX-V is offered in either a PGA package or an Intel-compatible 208-lead PQFP at either 33 or 40 MHz; the Cx486DX2-V is offered in the same two package types, in 50- and 66-MHz variations. An 80-MHz version is offered in a PGA package only.

Curiously, the lower-voltage devices have higher maximum frequencies than the 5-V devices. Presumably the top speed of these parts is limited by heat-dissipation issues, not internal gate-propagation delays.

## 9.10 Commentary

In many ways, Cyrix is the most promising of the Intel-compatible microprocessor vendors. Its products provide more differentiation from the Intel mold than those of AMD or any other contender. Through late-1994, Cyrix was the only company delivering 486 microprocessor designs with write-back caches. IBM's second-sourcing of Cyrix products (See **Chapter 10: IBM 386 and 486 Microprocessors**) firmly established these designs as a leading alternative to Intel's.

### What's in a Name?

Cyrix touched off a minor religious war in the x86 community when it jumped into the market by choosing to apply the digits "486" to the Cx486SLC and Cx486DLC—devices that were most definitely *not* like any 486 the industry had previously seen. "*Good Heavens!*" their detractors screamed, "The caches on these parts are *tiny*, and less sophisticated than *other* 486's, the bandwidth allowed by their 386-style (!) system interfaces falls *woefully* short of the Intel burst-mode 'standard,' and they don't contain floating-point units *at all!*"

Moreover, as more information emerged, it was discovered that the CPU itself was lacking in some of the social graces, most noticeably the extra adder for address calculations. "*So what if a hardware multiplier was included instead?*" they asked. "If hardware multipliers were any good, wouldn't *Intel* have thought of that, too?" Nor did it seem to make sense to upgrade the parts with Intel's clock-doubled "OverDrive" processors: if the memory system were able to support 32-bit buses and burst-mode transfers, why squander these resources on a crippled 386-style bus?

Intel was quick to attack the parts as not being "true" 486s. At best, Intel said, such a part should have been called a "turbo-386," or some such, following the precedent of Chips & Technologies' "Super386" family. And, from a strictly hardware perspective, Cyrix's detractors had a point.

In Cyrix's defense, however, cache effectiveness is an extremely nonlinear function of size. A 1K-byte cache is a whole lot better than none; further increasing cache capacity by even eight times does not provide an eight fold increase in throughput—nor even two fold. By putting write buffers and even a small amount of cache on chip, Cyrix was able to greatly decouple CPU throughput from bus bandwidth for many applications. Each of the Cyrix parts, including the Cx486S and Cx486S2,

could be upgraded with simple 387-style FPUs, which provided a smaller and much less expensive alternative to the i487SX, albeit with reduced performance. And from a software perspective, implementing the six new instructions defined by the 486 made the entire Cyrix product line 486 material.

Clearly, the marketing value of calling even its early product a “Cx486SLC” was great—the name prompted users to think of the part as an entry-level competitor to the 486—and the resulting sales volume has shown that Cyrix made the right choice. The sad, sad fate of C&T’s (more appropriately named) Super386 family only reinforces this conclusion.

### **Business Issues**

From the start, Cyrix focused on selling to leading U.S. PC makers. Several major U.S. vendors announced plans to use the part within days of the chip’s announcement. The same customers that adopted early AMD products also formed a ready-made customer base for Cyrix. AMD customers are, by definition, willing to consider sources other than Intel, and the Cyrix parts made it possible to produce significantly faster machines with only a minor redesign.

One reason for Cyrix’s initial success is that throughout 1992 and 1993 Intel was unable to keep up with 486 demand, leaving many smaller system vendors with inadequate supplies. Cyrix’s chips gave these vendors an available, economical alternative that provides a good fraction of 486 performance and lets these vendors market the systems as 486-based.

Intel’s response has been to bombard the market with a broad range of processor choices, forcing Cyrix to price its chips more aggressively or look for new niches. The notebook and low-power PC markets may, for now, provide enough room for both companies, but Cyrix’s less compact die and its dependence on outside foundries would seem to put it at a disadvantage.

In response, Cyrix is focusing its more recent designs on niches in which it can provide some benefit that Intel’s chips do not offer. Since Cyrix is a fabless chip vendor with no direct control over its own production capacity, it decided not to pitch the Cx486DX as yet another alternate source for the commodity i486DX or Am486DX market, but focused instead on the 50-MHz and faster variations. This strategy will continue with the introduction of the Cyrix “M1” processor in 1995; see **Chapter 18: Futures** for a description of the M1 design.

**Legal Issues** Even before Cyrix had formally announced any parts, Intel filed a lawsuit claiming the Cyrix design infringed four of its patents. Intel conceded that it filed the suit without having examined a Cyrix chip, basing its claim on the assertion that any x86-compatible device must *surely* infringe *some* patent. Intel's quick action in filing this suit was indicative of the degree to which Intel feels threatened by Cyrix.

Cyrix claimed both that its part does not infringe Intel's patents and that, in any case, it has been manufactured only by foundries that have patent cross-license agreements with Intel. Cyrix used SGS-Thomson and Texas Instruments as its initial foundries, each of which has patent cross-licensing agreements with Intel in place, and has now turned to IBM for the bulk of its future production.

**Compatibility** While AMD designed its 386 by matching Intel's design very closely and making parametric improvements (such as a higher clock rate), Cyrix designed a completely new processor core. While this is what made it possible for the Cyrix devices to achieve higher performance levels than the corresponding Intel and AMD 386 families, it also made the burden of proof with respect to compatibility somewhat greater.

AMD's approach may have been best for the first non-Intel 386 chip, when customers were just getting accustomed to the idea of a supplier other than Intel, and skepticism about compatibility was high. As the market got used to the idea of multiple implementations of x86 CPU cores, however, AMD was put in an increasingly difficult position.

It appears that Cyrix did its homework well. Cyrix's first silicon (called the A-0 version) was successfully tested using DOS, Windows, and UNIX environments—an impressive accomplishment for so complex a device. Cyrix says only three minor compatibility problems were detected, and these were corrected in the "A-1" version. The B-0 version added support for system management mode.

With the incorporation of an FPU into the Cx486DX family, however, new compatibility issues have arisen. The Cyrix IU may work fine, the FPU may work fine, but the interface between them can still raise the potential for new quirks to appear. See **Chapter 18: Compatibility** for details of one such quirk that necessitated a slight modification to the Cyrix chip.

## 9.11 For More Information...

Additional technical information on Cyrix products may be found in the following publications:

### Vendor Publications

- 1: *Cx486DLC Performance Report*. Cyrix Corporation, 6/92, order #94076-02.
- 2: *Cx486DX/DX2 3 and 5 Volt Microprocessors*. Cyrix Corporation, 1994, order #94113-01.
- 3: *Cx486DX2-V Microprocessors*. Cyrix Corporation, 9/94. (*data sheet for Cx486DX2-V.*)
- 4: *Cx486SLC and Cx486DLC Compatibility Report*. Cyrix Corporation, 7/92, order #94074-00.
- 5: *Cx486SLC & Cx486DLC Application Notebook*. Cyrix Corporation, 5/92, order #94060-15.
- 6: *Cx486SLC2 Microprocessor Press Kit*. Cyrix Corporation, 11/93.
- 7: *Cyrix Cx486DLC Microprocessor Data Sheet*. Cyrix Corporation, 1992, order #94076-01.
- 8: *Cyrix Cx486S and Cx486S2 Processors Data Book*. Cyrix Corporation, 1993, order #94102-00.
- 9: *Cyrix Cx486SLC Microprocessor Data Sheet*. Cyrix Corporation, 1992, order #94085.
- 10: *Cyrix Cx486SLC2 Microprocessor Data Sheet*. Cyrix Corporation, 10/93, order #94123-00.

### Microprocessor Report Articles

- 11: *Cyrix Introduces SX Version*. MPR vol. 4 no. 6, 4/4/90, pg. 4. (*Most Significant Bits item.*)
- 12: *Intel and Cyrix Exchange Lawsuits\**. Michael Slater, MPR vol. 5 no. 1, 1/23/91, pg. 16. (*Feature article.*)
- 13: *Intel Loses Bid for Injunction Against Cyrix*. MPR vol. 5 no. 22, 12/4/91, pg. 4.
- 14: *Cyrix Joins x86 Fray with 386/486 Hybrid\**. Brian Case and Michael Slater, MPR vol. 6 no. 5, 4/15/92, pg. 1. (*Cover story.*)
- 15: *Cyrix Challenges 486DX with C486DLC\**. Michael Slater, MPR vol. 6 no. 8, 6/17/92, pg. 1. (*Cover story.*)

- 16: *Judge Rules SGS-Thomson License Protects Cyrix\**. Michael Slater, MPR vol. 6 no. 11, 8/19/92, pg. 1. (Cover story.)
- 17: *Cyrix Beta-Testing Clock-Doubler for 386 Systems*. MPR vol. 6 no. 12, 9/16/92, pg. 4. (Most Significant Bits item.)
- 18: *Cyrix Adds Extended 486SLC*. MPR vol. 6 no. 15, 11/18/92, pg. 4. (Most Significant Bits item.)
- 19: *Cyrix Reveals Its First 486-Pinout Processor*. MPR vol. 6 no. 15, 11/18/92, pg. 4. (Most Significant Bits item.)
- 20: *Cyrix Delivers Revamped M6 Processors*. Linley Gwennap, MPR vol. 7 no. 7, 5/31/93, pg. 14. (Feature article.)
- 21: *Cyrix IPO Reveals Fab Issues*. MPR vol. 7 no. 9, 7/12/93, pg. 19. (Most Significant Bits item.)
- 22: *Cyrix Chip Upgrades 386 System to 486*. MPR vol. 7 no. 10, 8/2/93, pg. 4. (Most Significant Bits item.)
- 23: *Cyrix Readies 486DX-Compatible CPU*. Michael Slater, MPR vol. 7 no. 11, 8/23/93, pg. 1. (Cover story.)
- 24: *AMD Loses OmniBook Socket to TI*. MPR vol. 7 no. 12, 9/13/93, pg. 5. (Most Significant Bits item.)
- 25: *Intel, Cyrix Drop Court Cases*. MPR vol. 7 no. 12, 9/13/93, pg. 5.
- 26: *Cyrix Describes Pentium Competitor*. Linley Gwennap, MPR vol. 7 no. 14, 10/25/93, pg. 1. (Cover story.)
- 27: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (Cover story.)
- 28: *Cyrix Gets Aggressive with 486DX*. MPR vol. 8 no. 5, 4/18/94, pg. 5.
- 29: *IBM and Cyrix Ink Five-Year Pact*. Michael Slater, MPR vol. 8 no. 6, 5/9/94, pg. 10. (Feature article.)
- 30: *Cyrix, IBM Deliver First Fruit of Partnership*. MPR vol. 8 no. 8, 6/20/94, pg. 5. (Most Significant Bits item.)

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)

# 10

## IBM 386 and 486 Microprocessors

If there's any company in the world that can challenge Intel—at least in terms of sheer leading-edge fabrication capacity—it's IBM. Intel and IBM have been engaged in a rather strange dance since 1980, when IBM selected Intel to supply CPUs for the first IBM PC. The IBM PC Company division has since become Intel's best customer, and was for most of those years the world's largest manufacturer of IBM-compatible (natch!) PCs. More recently, other IBM divisions have begun trying to muscle into the Intel market, hoping to grab whatever slices of the x86 pie they can finagle.

### **Dysfunctional Corporate Relations**

IBM and Intel were once chronically codependent. In 1983, as business conditions turned down and Intel feared it might become the target of a hostile takeover, IBM stepped in as a white-knight-in-waiting, acquiring 12% of Intel's outstanding stock and securing a position on its board of directors. Once the danger had passed, IBM divested its Intel holdings, in 1987—and realized a considerable profit for its troubles.

When Intel introduced its 386 microprocessor line in 1985, and its 486 processor in 1989, IBM was the only company with which Intel established cross-licensing agreements. IBM was granted access to Intel's design database, production test vectors, and so forth, and was granted the rights build a limited number of compatible processors for internal use. IBM was also allowed to develop customized processors derived from the Intel designs to meet its own needs. While the details of these agreements have never been made public, the industry consensus is that in return for access to Intel's design database, IBM was prohibited from selling chips directly on the open market.

In addition, Intel set limits on the fraction of overall unit demand IBM was allowed to satisfy from its in-house production, may have collected royalties on whatever chips IBM did produce, and apparently restricted IBM's customized designs to using more primitive pinouts. Moreover, Intel restricted the conditions under which IBM could distribute products derived from the Intel designs. While IBM could incorporate these parts into motherboards and CPU daughterboards for a PC or a PC upgrade, it was not allowed to sell the chips individually.

More recently, as mainframe revenues have fallen, the IBM Microelectronics Division has been trying to become a stronger force in the microprocessor components arena. The company has designed and aggressively promoted the RISC-based PowerPC family as a direct competitor to Intel's high-end desktop processors, thereby also bolstering the financial viability of Intel's strongest competitor, Motorola.

Meanwhile, other arms of the IBM behemoth have apparently been looking for creative ways to stretch the intent of the Intel cross-licensing deals, forming strategic alliances with several of Intel's competitors, and attempting to find other ways to compete with Intel head on.

At press time, IBM was in production with and attempting to sell indirectly three 386- and 486-class microprocessors designed under the 1985 Intel technology exchange agreement, and is a licensed second source openly selling two of Cyrix's designs. Each of the x86 processors currently in the IBM stable is described in the sections below.

### **Creatively Licensed**

In late 1992, in order to circumvent Intel's licensing restrictions, IBM reportedly began broadening the definition of a "system or multiple-chip module" to include systems approaching trivial complexity. A microprocessor chip and gate array that enhanced its bus interface might be combined onto a small, pin-compatible module that plugged directly into an Intel CPU socket on an existing motherboard might be classified (from IBM's perspective) as a multiple-chip computer system.

Moreover, while IBM was not allowed to sell its microprocessor chips to OEM system vendors, as such, it decided that it *would be* legal to hire an OEM system vendor (Compaq, say, for the sake of discussion) to assemble motherboards, provide that vendor with the otherwise-unmarketable chips to install on the boards, and then sell those same motherboards back to the same OEM system vendor.



Technically, the chips themselves would not be sold per se, but it's a safe bet that any contractors/customers would pay IBM more for the board with the IBM CPU installed than the amount IBM had paid them for the service of doing the installation. It's not known if IBM ever managed to pull off this ploy, but PC vendors were alerted that such arrangements might be possible.

### **Competitive Thrusts**

In September 1993 IBM leaked word that a future PowerPC processor designated the 615 would offer built-in x86 emulation hardware, allowing x86 code to be run without an Intel CPU. Later in the year IBM became a foundry for Cyrix components, plugging the gap left when Cyrix and TI parted ways.

In February of 1994 IBM let it be known that it would *not* be acquiring rights to the Intel Pentium, or staking its future on Pentium's success. In April, IBM acquired from Chips and Technologies the entire x86 design database that C&T had developed in the course of pursuing its ill-fated Super386 strategy (see **Chapter 8**).

In May, IBM signed a five-year pact with Cyrix to serve as Cyrix's foundry, share its leading-edge process technology, and act as a second source for selling Cyrix parts. And in June, IBM struck a deal to become a foundry and second source for Nex-Gen as well, showing just how single-minded IBM is in attempting to divert the Intel juggernaut.

Meanwhile, back in Armonk, the IBM PC Company continues to introduce new systems employing Pentia and other processors purchased from Intel, but there have been unconfirmed press reports that even if the Microelectronics Division succeeds in building the x86-accelerated PowerPC 615, the PC Company does not plan to use the part in its systems.

## 10.1 The IBM 386SLC Microprocessor

The 386SLC is an enhanced 386-class integer core with an 8K-byte I/D cache with enhanced 386SX-compatible pinout. Table 10-1 summarizes the general features and specifications of the 386SLC microprocessor.

Product Name	IBM 386SLC
Introduction Date	October 1991
Prognosis	Fading
Device Integration Level	Pipelined 32-bit IEU and PMMU 8K-byte unified instruction/data cache
CPU Architecture Level	Extended 486 integer instruction set
Core Technology	IBM-designed 386-like static integer core
Pinout	Augmented compatible 386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Two-cycles minimum per 16-bit transfer One-half cycle address pipelining optional
Cache Support	8K bytes unified I- and D-cache with parity Two-way set associative Write-through operation only
Floating-Point Support	Optional external 387SX-class FPU
Operating Voltage	4.5 V to 5.5 V
Frequency Options	16-, 20-, or 25-MHz core operation
Clocking Regime	Core operating frequency = $1/2 \times \text{Clkin}$
Active Power Dissipation	3.25 W @ 5.5 V and 25 MHz (worst case)
Power-Control Features	IBM "Low-Power" halt mode plus In-Circuit Emulation/IBM Power-Management Modes
Process Technology	0.9 $\mu$ two-layer-metal CMOS
Die Size	500 mils $\times$ 500 mils (250,000 mil <sup>2</sup> ) 12.7 mm $\times$ 12.7 mm (161 mm <sup>2</sup> )
Transistor Count	875,000 transistors
Package Options	100-pin metal quad flat pack
Other Features	Cache coherency logic supports bus snooping and invalidation on a per-line basis.

Table 10-1. IBM 386SLC feature summary.

The IBM 386SLC story reads much like the Cyrix Cx486SLC's: soup-up a 386-class CPU, add a combined instruction/data cache, leave off the FPU, and box it all up in a package compatible with the 386SX-type pinout. IBM strayed from Cyrix's strategy, however, by starting with a microcoded core implementation that requires at least two clock cycles even for simple instructions, by adding a considerably larger and more

sophisticated cache, and by allowing cache coherency to be maintained with much finer granularity.

### **Cache Configuration**

The IBM 386SLC cache has many of the same characteristics as a conventional (Intel-style) 486 core. In both designs the cache has 8K bytes of capacity, a 16-byte line size, and write-through operation only.

In some ways, the 386SLC cache has been embellished. Consistent with IBM's well-established compulsion to include parity in its memory systems, a parity bit has been added for each byte in the cache. Any time a parity error is detected on data read from the cache, the data is discarded, the corresponding word in cache is flushed (marked invalid), and a new external fetch cycle is initiated. Curiously, no other 386- or 486-class chip vendor found on-chip data parity to be an issue, apparently assuming that once data arrived "clean" at a processor's bus pins, one could feel pretty confident the chip itself would work the way it should. In IBM's defense, however, enforcing on-chip cache parity could serve as an (albeit crude) form of production fault tolerance, increasing effective die yield by allowing devices with single-bit cache defects to continue to work properly—more or less. And at least IBM chose to forgo CRC!

The IBM cache design also allocates a new line on write operations that miss the cache called allocation write, whereas Intel's and Cyrix's designs do not. Write cycles that miss the cache will write through to external memory, and then immediately initiate a cache-reload sequence to refill a cache line with the 16-byte block of data to which the write occurred.

Offsetting these sophisticated enhancements is IBM's seemingly inexplicable decision to make its cache just two-way set associative vs Intel's four-way configuration. Even though the overall capacities of the two parts are the same, simple rules of thumb for cache-design suggest that by cutting the number of ways in half, the IBM cache achieves about the same hit rate as an Intel-design cache with only half the capacity.

### **System Interface**

The IBM 386SLC system interface is derived from that of a conventional 386SX. By default, the device may be inserted into an existing 386SX socket, and should execute existing software safely, albeit with the on-chip cache and other hardware features disabled.

Since the 386SX lacks the signals needed to support cache functions, IBM (like Cyrix) appropriated certain existing, underuti-

lized pins in such a way as to assure default compatibility. As with the Cyrix designs, configuration software may optionally set bits within device-specific registers to enable various enhanced functions according to the features the system hardware can support.

Figure 10-1 shows the IBM 386SLC system interface. Note that the address bus and bus control signals have been made (optionally) bidirectional in order to facilitate cache coherency. Each of the Cx486SLC's cache-flushing options is supported by IBM, but in addition, IBM allows full, per-line cache snooping.

IBM 386SLC interface signals not defined by a conventional 386SX pinout are described in Table 10-2.

The first six entries in Table 10-2 indicate pins that operate (by default) the same as the corresponding pins on a 386SX-class device. Each, however, has ancillary functions that may be

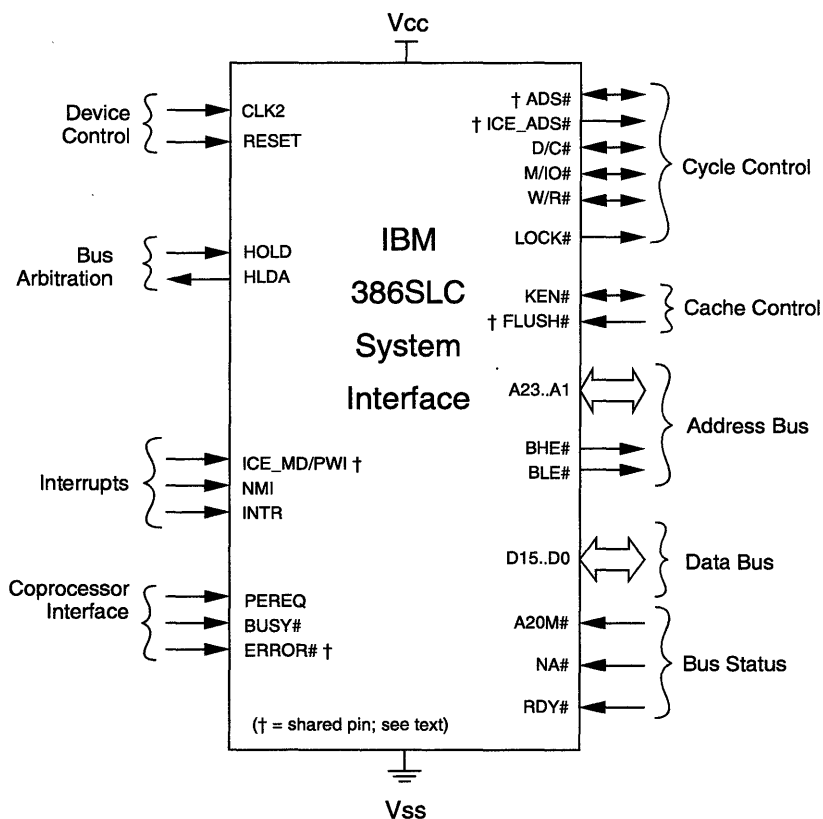


Figure 10-1. IBM 386SLC system interface.

Symbol	Dir	Signal Name/Function	386SLC PQFP Pin #	Replaces 386SX PQFP Signal
HOLD	In	Default: Hold outputs for other master Optional: Hold outputs and flush cache	4	(note 2)
ADS#/ FLUSH	I/O	Out: Address strobe; start bus cycle In: Start cache snoop cycle or flush entire cache (note 1)	16	(note 2)
A23..A1	I/O	Out: Address for system bus cycles In: system address for cache snooping	80..51, with gaps	(note 2)
M/I0#	I/O	Out: Defines memory vs I/O bus cycles In: cycle type for cache snooping	23	(note 2)
W/R#	I/O	Out: Defines write vs read bus cycles In: cycle type for cache snooping	25	(note 2)
BHE# BLE#	I/O	Out: Byte high enable, byte low enable In: production test inputs	19 17	(note 2)
A20M#	In	Address-bit 20 mask	31	N.C.
KEN#	I/O	In: Cacheability enabled for system data Out: Cacheability allowed, cache- reload in progress, or write-buffer output request pending (note 1)	29	N.C.
ICE_MD/ PWI	In	In-circuit emulation/Power interrupt request (note 1)	27	N.C.
ERROR#/ ICE_ADS	I/O	SMM memory Address Strobe (note 1)	36	N.C.
N.C.	—	No connect	28	FLOAT#

Table 10-2. IBM 386SLC special interface signals.

note 1: pin direction/function determined by software configuration register

note 2: standard 386SX pinout defines signal on same pin as output only

enabled by software if the host-system hardware supports it. These functions are as follows:

- **HOLD:** In addition to its conventional functions as a bus-arbitration request, the HOLD pin can be programmed to flush the cache (invalidate all cache lines) whenever the CPU releases control of the bus. This is a crude but effective way to ensure that no other bus master can modify a memory location already held in on-chip cache, thereby rendering the cached copy stale.
- **ADS#/FLUSH:** For 386SLC-initiated bus cycles, the ADS# pin operates normally. When the CPU has released control of the system bus following the normal HOLD/HLDA handshake protocol, and assuming the above "HOLD flushes all" func-

tion is disabled, ADS# may be configured to provide two additional levels of cache-coherency elegance. In the first, externally asserting ADS# while the CPU is in the Hold state will flush the entire cache. In the second, asserting ADS# while the CPU is being held will initiate an Intel-486-like bus-snooping cycle, using the address value currently being driven onto the address bus (see below).

- A23..A01: By default, pins A23..A01 act as a conventional 386SX address bus. When software enables the ADS#-initiated per-address bus-snooping function described above, these pins act as inputs, allowing the CPU to monitor the addresses involved in system-memory write cycles.
- M/IO# and W/R#: By default, M/IO# and W/R# perform the same cycle-type definition functions as on a conventional 386SX. When per-address bus snooping is enabled, the CPU monitors these pins as inputs to qualify whether to snoop a system-memory cycle.
- BHE# and BLE#: BHE# and BLE# also perform the same byte-enabling functions as on a conventional 386SX. According to the IBM 386SLC data sheet, they also serve as input pins for (undefined) testing functions.
- A20M#: This is a newly added signal for 386SX-pinout parts, and may be configured to perform the same internal address-line masking function described previously in **Chapter 6** for Intel devices or **Chapter 9** for the Cyrix family.
- KEN#: The KEN# signal, too, is new to the 386SX pinout. It may also be configured to perform any of four separate functions. The most obvious is the same cache-enabling function described in **Chapters 6 and 9**.

As with the Cyrix devices, though, the IBM 386SLC may be configured to define the cacheability and noncacheability of memory regions according to the settings of internal configuration registers. When this option is enabled, KEN# may be reconfigured as an output, informing the outside system (and any second-level caches therein) that value currently being fetched has been designated as not to be cached.

Moreover, KEN# may be configured to serve as one of two status flags, indicating to the outside system either that an internal cache-line update is occurring, or that the internal write buffers have unwritten data pending.

- ICE\_MD/PWI: The Power Interrupt pin, if enabled, performs the same function as a System Management Interrupt on numerous other products. (The ICE\_MD prefix is a vestigial reference to a special mode for in-circuit emulation and debugging support.)
- ERROR#/ICE\_ADS: The ERROR# input pin is normally part of the floating-point coprocessor error-reporting interface. In systems that support IBM's power-management mode, the pin may alternatively be configured to initiate accesses to the power-management memory space.

One curious distinction between the IBM 386SLC and other 486-family products is the order in which it fills its internal cache lines. The Intel-designed 486 CPU core fills its 16-byte cache lines with a burst of four successive read cycles in the order described in **Chapter 6**. Cyrix-designed CPU cores maintain a separate valid bit for each 16-bit word, and thus need not fill an entire cache line at a time. The IBM 386SLC, in contrast, transforms every cacheable memory request into a series of eight successive memory read cycles, following the order shown in Table 10-3.

Target Address	1st Word	2nd Word	3rd Word	4th Word	5th Word	6th Word	7th Word	8th Word
xxxx0H	xxx0	xxx2	xxx4	xxx6	xxx8	xxxA	xxxC	xxxE
xxxx2H	xxx2	xxx0	xxx4	xxx6	xxx8	xxxA	xxxC	xxxE
xxxx4H	xxx4	xxx6	xxx8	xxxA	xxxC	xxxE	xxx0	xxx2
xxxx6H	xxx6	xxx4	xxx8	xxxA	xxxC	xxxE	xxx0	xxx2
xxxx8H	xxx8	xxxA	xxxC	xxxE	xxx0	xxx2	xxx4	xxx6
xxxxAH	xxxA	xxx8	xxxC	xxxE	xxx0	xxx2	xxx4	xxx6
xxxxCH	xxxC	xxxE	xxx0	xxx2	xxx4	xxx6	xxx8	xxxA
xxxxEH	xxxE	xxxC	xxx0	xxx2	xxx4	xxx6	xxx8	xxxA

Table 10-3. IBM 386SLC cache-line fill order.

Note that, whatever the originally requested memory location, the sequence shown in Table 10-3 ensures that the 16-bit word requested is retrieved first, so that it can be passed directly to whichever unit (the instruction decoder or execution pipeline) requested it. Next, the chip fetches the other half of the 32-bit aligned value involved. The bus interface then retrieves whatever values remain in the 16-bit cache line affected, in ascending sequence, and then wraps around to pick up any cache-line values with addresses lower than the original target.

**Programming Model Extensions**

The IBM 386SLC user-mode and system-mode programming models match those of the conventional (i.e., Intel-designed) 486 integer core. The one exception is the addition of two new model-specific configuration registers (MSRs) used to enable the hardware configuration options described above, and to make certain error and event flags visible to system software. Figure 10-2 shows the complete system register set.

Each of the two MSRs is defined to be 64 bits wide, though most of the bits are unused. MSR 1000H contains 15 assorted hardware-configuration and status-reporting flags, as described in Figure 10-3. This register is cleared by a hardware reset, with the effect that all optional features are disabled, and the device operates as a simple, standard, compatible, unembellished “safe” 386SX.

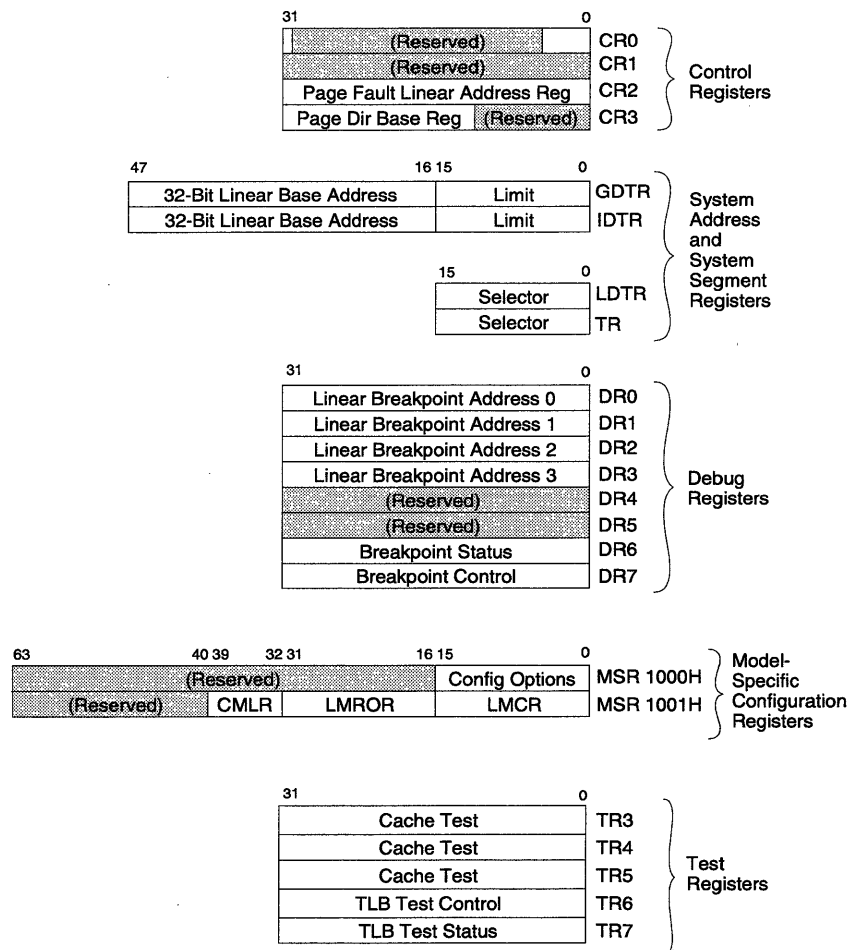


Figure 10-2. IBM 386SLC system register model.



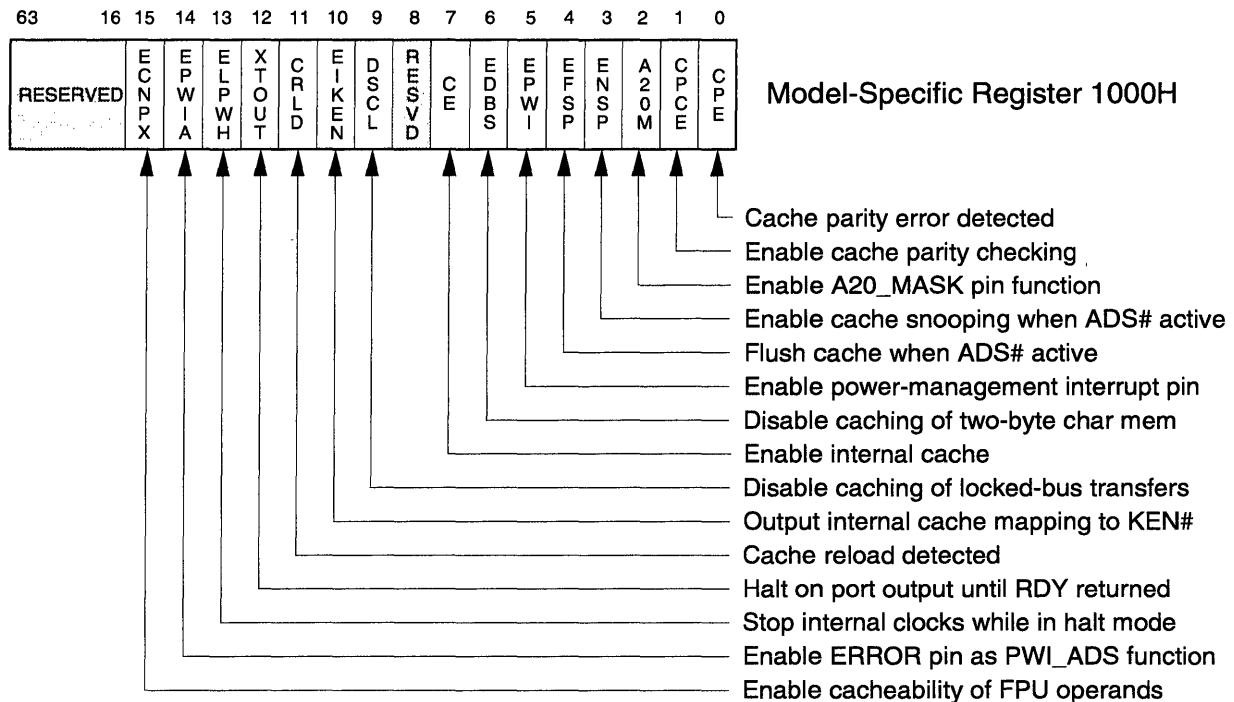


Figure 10-3. IBM 386SLC model-specific register 1000H.

MSR 1001H lets software define the cacheability characteristics of memory regions throughout the 16M-byte physical memory space. Bits 15..0 of MSR 1001H (the low-order memory cacheability register) define the cacheability of the low-order 1M bytes, with 64K-byte granularity. Setting bit 0 means system memory region 000000H to 00FFFFH may be cached; setting bit 9 enables caching for memory with addresses 090000H through 09FFFFH, and so forth. Bits 31..16 of MSR 1001H (the low-order memory read-only register) allow the same set of memory regions to be marked as read-only, i.e., setting bit 31 means memory with addresses 0FxxxxH may not be written.

The one exception to the aforescribed scheme is that if bit EDBS (bit 6 of MSR 1000H) is set, then memory addresses 0E0000H through 0E0FFFH will not be cacheable, regardless of the state of MSR 1001H bit 14. (I'm not making this up; apparently, in IBM PCs, this particular 4K block is reserved for dynamically redefined character sets for languages that require two bytes to define each character, such as Kanji.) (*Oh, those clever IBM designers!*)

Bits 39..32 of MSR 1001H (the cacheable-memory limit register) enable the cacheability of memory regions above one megabyte. Depending on the value stored in this field, the same number of contiguous 64K-byte blocks will be cacheable. (The IBM data sheet uses the word “segment” here, and then immediately inserts a paragraph to explain that the word “segment” is misleading in this context, since it has nothing to do with the x86-style memory segmentation and is actually meant to imply a block of contiguous addresses.) Storing the value 28, for example, enables caching for memory locations 1M through  $(1M + (28 \times 64k) - 1)$ .

MSR 1001H is also cleared by a hardware reset, again causing the device to default to simple, “safe” 386SX operation. Until further modification, the entire 16M-byte physical address space is treated as uncacheable and read-write-able.

### Instruction Set Extensions

The IBM 386SLC supports the six new instructions defined by the original i486DX. In addition, the device implements six new instructions, as listed in Table 10-4.

Instruction	Mode	Operation	Opcode
BSWAP	User/ System	Byte swap. Reverse byte order within 32-bit register	0F C8
XADD	User/ System	Atomic (indivisible) exchange and add	0F C0
CMPXCHG	User/ System	Atomic (indivisible) compare and exchange	0F B0
INVD	System	Invalidate data cache	0F 08
WBINVD	System	Perform write-back cycle and invalidate cache	0F 09
INVLPG	System	Invalidate TLB page entry	0F 01
WRMSR	System	Write model-specific register	0F 30
RDMSR	System	Read model-specific register	0F 32
ICEBP	System	ICE/PWI Breakpoint	F1
ICERET	SMM	Resume normal execution mode	0F 07
UMOV	SMM	User-space move (load)	0F 12
UMOV	SMM	User-space move (store)	0F 10

Table 10-4. IBM 386SLC instruction set additions.

- WRMSR and RDMSR write and read the two model-specific registers defined above. Actually, the instructions are severely underutilized; before invoking either instruction, register ECX must hold the full 32-bit identification code of the MSR register to be referenced. Future designs could

thus easily expand this scheme to include nearly 4G additional 64-bit registers. (*Oh, those thorough IBM designers!*) Register pair EDX:EAX serves as the 64-bit value to be written or retrieved.

- ICEBP provides a software mechanism by which the IBM power-management mode may be invoked.
- ICERET is a special return instruction that restores normal operation following a power-management service routine.
- The UMOV instructions provide a mechanism for reading and writing the conventional (i.e., non-power-management) system memory space when power-management mode is active IBM.

### **Vital Statistics**

The IBM 386SLC is fabricated using a 0.9-micron CMOS process with three layers of metal. The design employs about 875,000 transistors on a die that measures 500 mils  $\times$  500 mils, or 250,000 mils<sup>2</sup> (161 mm<sup>2</sup>).

Since the device is not technically a commercial product, it would be inappropriate to say it is “offered” or “available” in any particular configurations. However, the 386SLC data sheet states that the part is housed in a 100-lead MQFP (metal quad flat package) with the same physical dimensions and pinout as the Intel or AMD PQFP. Operation is specified for supply voltages between 4.5 V and 5.5 V, and the data sheet gives timing specifications for device operation with core frequencies of 16, 20, or 25 MHz.

## 10.2 The IBM BL486SLC2 Microprocessor

The BL486SLC2 is an enhanced implementation of the IBM 386SLC, with twice the amount of on-chip cache, a software-controlled on-chip clock doubler, and lower-voltage operation. Alas, the part is still constrained by its 386SX-class pinout. Table 10-5 summarizes the features and specifications of the BL486SLC2 microprocessor.

Product Name	IBM BL486SLC2
Introduction Date	August 1992
Prognosis	Constrained
Device Integration Level	Pipelined 32-bit IEU and PMMU 16K-byte unified instruction/data cache Core-logic frequency-doubler circuitry
CPU Architecture Level	Standard 486 integer instruction set
Core Technology	IBM 486 core
Pinout	Augmented compatible 386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Two cycles minimum per 16-bit transfer One-half cycle address pipelining optional
Cache Support	16K bytes unified I- and D-cache with parity Four-way set associative Write-through operation only
Floating-Point Support	Optional external i387SX FPU
Operating Voltage	2.97 V to 3.78 V
Frequency Options	40-, 50-, or 66-MHz core operation
Clocking Regime	Core operating frequency = 1 × Clkin
Active Power Dissipation	3.0 W @ 3.6 V and 66 MHz (worst case)
Power-Control Features	IBM system-management mode extensions
Process Technology	0.8μ four-layer-metal CMOS
Die Size	303 mils × 354 mils (107,000 mm <sup>2</sup> ) 7.7 mm × 9.0 mm (70.3 mm <sup>2</sup> )
Transistor Count	1,349,000 transistors
Package Options	100-pin metal quad flat pack

Table 10-5. IBM BL486SLC2 feature summary.

### Cache Configuration

The biggest addition to the BL486SLC2 over its 386SLC predecessor—in raw die area if nothing else—is its newly enlarged and refined cache design. Total capacity has been doubled to 16K bytes, and now accounts for more than two-thirds of the total device transistor count. Just as important to cache-design

fans is the fact that the cache is now four-way set associative, like the caches in the 486SX and 486DX products from Intel, AMD, and Cyrix. Academicians would contend that doubling the set associativity should have as much effect on improving hit rates as doubling raw capacity.

The cache still supports parity, still has a 16-byte line size, is still write-through only, and still allocates and reloads new cache lines on writes. If IBM's data sheet is to be taken literally, the cache institutes a full least-recently-used (LRU) line-replacement algorithm. Intel's follows a "pseudo-LRU" approach that requires just four "valid" bits and three usage-state bits per four-way set. The IBM approach, if truly uncompromised, would require at least a couple of extra bits, and considerably messier way-selection logic. (*Oh, those aesthetic IBM designers!*)

### Clocking Regimes

The other major change to the BL486SLC2 is its clock-generation circuit. The device contains an on-chip clock doubler such that the core may execute instructions at twice the frequency of the bus interface. Note, though, that since the bus divides the frequency of the CLK2 input signal by two, the net effect is merely to restore operation of a 1× clock.

In order to enable the clock-doubling capability, software must initialize a configuration register. Once doubling has been enabled, though, it may not be disabled short of reinitiating a full hardware reset. In order to later reduce the clock frequency, the clock-doubler logic must first be brought back in phase with the external clock input. The BL486SLC2 defines both hardware and software protocols for doing so.

### System Interface

The IBM BL486SLC2 system interface is essentially identical to that of the 386SLC, and is shown in Figure 10-4. In addition to the system interface signals defined for the 386SLC, two new pin functions have been added to the part, as shown in Table 10-6.

When operating in clock-doubled mode, the on-chip doubling circuitry is not able to tolerate dynamic changes to the input clock frequency. Asserting DFS\_REQ# instructs the CPU to drop back out of PLL operation and resynchronize itself with the externally supplied clock.

Once that has been accomplished, the processor will assert the DFS\_RDY# signal, informing the external system that it may now safely alter its input clock, for example to reduce power during

Symbol	Dir	Signal Name/Function	BL486-SLC2 PQFP Pin #	Replaces 386SX PQFP Signal
DFS_REQ#	In	Dynamic frequency change request	20	N.C.
KEN#/ DFS_RDY#	I/O	In: Cacheability enabled for system data Out: Cacheability allowed, cache-reload in progress, write-buffer output request pending, or dynamic frequency change request ready (note 1)	29	N.C.

Table 10-6. IBM BL486SLC2 special interface signals.

note 1: pin direction/function determined by software-configuration register

idle periods. When DFS\_RDY# is deasserted, internal clock operation will resume whatever mode was set when the clock-configuration register was initialized. (Curiously, IBM selected pin 29—KEN#—to perform the DFS\_RDY# handshake function. This was the one pin on the package already being forced to juggle one other input and three other output functions.)

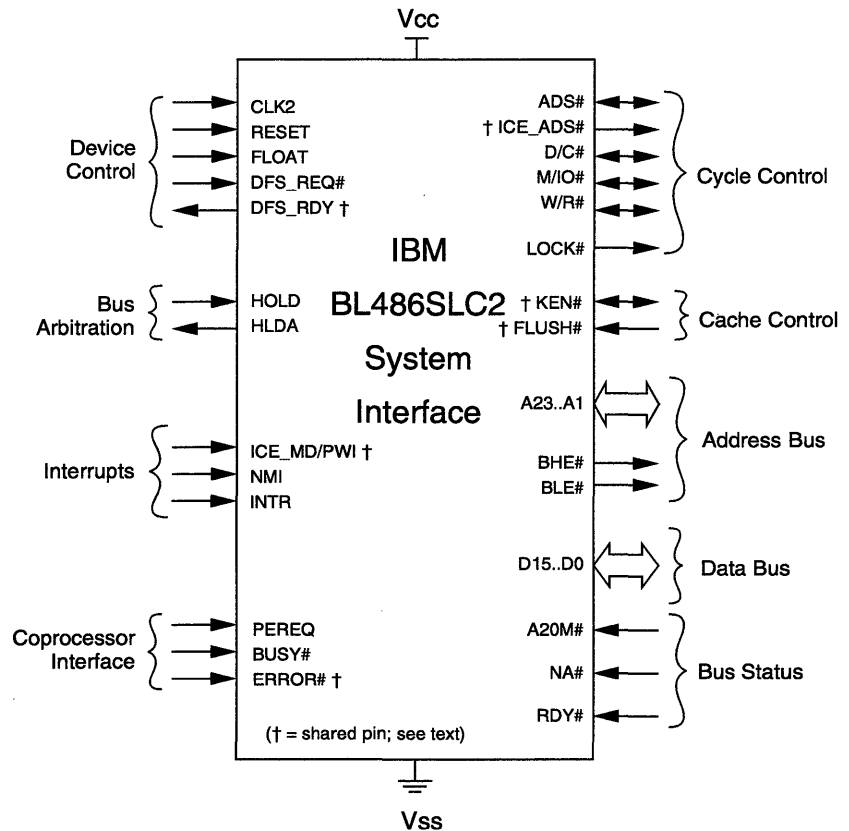


Figure 10-4. IBM BL486SLC2 system interface.

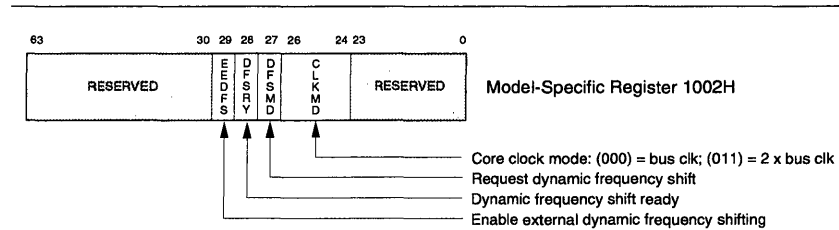


Figure 10-5. IBM BL486SLC2 model specific register 1002H

## Architecture Extensions

The instruction set of the BL486SLC2 is identical to that of the IBM 386SLC. The programming models of the two parts are the same, with a few minor enhancements. MSR 1000H and MSR 1001H contain all the same bits and perform all the same functions as on the 386SLC. In addition, MSR 1000H of the BL486SLC2 implements three additional bit functions:

- CPGE (bit 16), when set, will force the on-chip cache parity logic to intentionally store the incorrect parity value for testing purposes.
- BUSRD (bit 17), when set, forces all memory read cycles to be made from the external bus, even if the on-chip cache is enabled and detects a hit. Values read from memory will be copied into the cache, and memory system coherency is maintained.
- LWPLA (bit 18), when set, disables power to dynamic on-chip PLAs when operating in Halt mode. Additional cycles will be needed to re-enable PLAs in response to external events.

Finally, a new model-specific register has been added to the 486DLC2. MSR 1002H contains just six bits, which configure the on-chip clock circuitry as shown in Figure 10-5. Bits 29 through 27 enable a software-controlled protocol for software dynamically changing the external clock input, following a handshake sequence analogous to the hardware handshake described above.

## Vital Statistics

The BL486SLC2 is fabricated using a 0.8-micron CMOS process with four metal layers. Its die measures 7.7 mm × 9.0 mm, and (according to the data sheet) contains 1,349,000 transistors—10% more than Intel's full 486DX implementation—more than two-thirds of which are contained in the cache! Compared with the IBM 386SLC, the newer part packs more than half again as many transistors onto a die just 43% as large, for nearly 3.5× the device density.

Subject to the productization caveats of the previous section, the BL486SLC2 data sheet states that the part is housed in the same 100-lead MQFP as the 386SLC. The data sheet gives timing specifications for device operation with bus frequencies of 20, 25, or 33 MHz, with core operation up to 66 MHz. Supply voltages must be between 2.97 V and 3.78 V (*Oh, those fussy IBM designers!*) for core operation up to 25 MHz, or between 3.42 V and 3.78 V for core operation from 40 to 66 MHz.



## 10.3 The IBM BL486SX2/SX3 “Blue Lightning” Microprocessor

The official designation for IBM’s highest-end proprietary 386SX-pinout microprocessor is the BL486SX2/SX3, but the device was introduced and has been widely promoted as “Blue Lightning,” the code-name under which it was developed. The device is similar to the IBM BL486SLC2, but it uses a 386DX-class pinout in which address and data buses are a full 32 bits, and adds it aconfigurable clock-doubling or -trebling capability in order to allow core operation up to 100 MHz. Table 10-7 summarizes the features and specifications of the part.

Product Name	IBM BL486SX2/SX3 (“Blue Lightning”)
Introduction Date	August 1993
Prognosis	Fading fast
Device Integration Level	Pipelined 32-bit IEU and PMMU 16K-byte unified instruction/data cache Core-logic frequency-tripler circuitry
CPU Architecture Level	Standard 486 integer instruction set
Core Technology	IBM 486 core
Pinout	Augmented compatible 386DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Two cycles minimum per 32-bit transfer One-half cycle address pipelining optional Dynamic bus resizing for 16-bit transfers
Cache Support	16K bytes unified I- and D-cache with parity Four-way set associative Write-through operation only
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	3.0 V to 3.6 V
Frequency Options	25- or 33-MHz bus clock 50-, 66-, 75-, or 100-MHz core operation
Clocking Regime	Core operating frequency = 2 × or 3 × C <sub>kin</sub>
Active Power Dissipation	4.0 W @ 3.3V and 100-MHz (worst-case)
Power-Control Features	IBM system management mode extensions
Process Technology	0.8μ four-layer metal CMOS
Die Size	(82 mm <sup>2</sup> )
Transistor Count	1.43M transistors
Package Options	132-pin metal quad flat pack

Table 10-7. IBM BL486SX2/SX3 “Blue Lightning” feature summary.

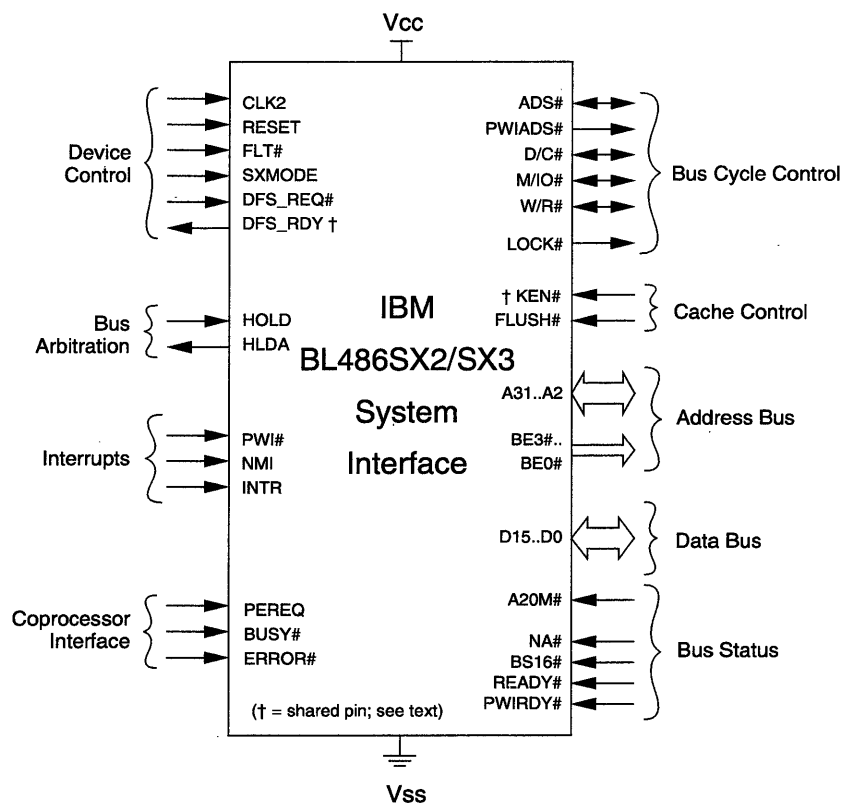


Figure 10-6. IBM BL486SX2/SX3 system interface.

### System Interface

The BL486SX2/SX3 system interface resembles that of a conventional 386DX device, with the addition of the IBM enhancement signals defined for the BL486SLC2. Figure 10-6 shows the BL486SX2/SX3 system interface schematically. Table 10-8 lists BL486SX2/SX3 signals not included in the standard 386DX pinout. Each of these signals functions as described above for other IBM processors.

Because of its wider bus interface, the BL486SX2/SX3 can refill a cache line in just half as many transfers. Again, though, the transfer order departs from the standard defined by 486DX-class processors. This order is shown in Table 10-9.

### Vital Statistics

The BL486SX2/SX3 die weighs in at 82 mm<sup>2</sup> and is fabricated using a 0.8-micron CMOS process with four metal layers. The part is housed in a 386DX-compatible 132-pin metal QFP package, and is specified for operation at bus frequencies of 25 or 33 MHz. Depending on whether the clock is doubled or tripled, the core may operate at frequencies of 50, 66, 75, or 100 MHz.

Symbol	Direction	Signal Name/Function	BL486SX2/SX3 PQFP Pin #	Replaces 386SX PQFP Signal
HOLD	In	Default: Hold outputs for other master Optional: Hold outputs and flush cache	28	(note 1)
ADS#/ FLUSH	I/O	Out: Address strobe; start bus cycle Optional In: Start cache snoop cycle or flush entire cache (note 2)	39	(note 1)
A31..A2	I/O	Out: Address for system bus cycles Opt. In: system address for cache snooping	104..67 (with gaps)	(note 1)
M/IO#	I/O	Out: Defines memory vs I/O bus cycles Opt In: cycle type for cache snooping	40	(note 1)
W/R#	I/O	Out: Defines write vs. read bus cycles Opt In: cycle type for cache snooping	43	(note 1)
BE3#.. BE0#	I/O	Out: Byte high enable, byte low enable In: production test inputs	38, 33, 32, 31	(note 1)
A20M#	In	Address-bit 20 mask	39	N.C.
SXMODE	In	386SX bus interface (16-bit bus) mode	62	N.C.
PW!	In	Power Interrupt request	59	N.C.
PWIADS#	Out	Power-management memory address strobe	37	N.C.
PWIRDY#	In	Power-management memory transfer ready	36	N.C.
DFS_REQ#	In	Dynamic frequency shift request	60	N.C.
KEN#/ DFS_RDY#	I/O	In: Cacheability Enabled for system data Optional Out: Cacheability allowed, cache-reload in progress, write-buffer output request pending, or dynamic frequency shift ready (note 2)	61	N.C.

Table 10-8. IBM BL486SX2/SX3 special interface signals.

note 1: standard 386SX pinout defines signal on same pin as output only  
note 2: pin direction/function determined by software-configuration register

Target Address	1st Word	2nd Word	3rd Word	4th Word
xxxxxxx0H	xxxxxxx0H	xxxxxxx4H	xxxxxxx8H	xxxxxxxCH
xxxxxxx4H	xxxxxxx4H	xxxxxxx8H	xxxxxxxCH	xxxxxxx0H
xxxxxxx8H	xxxxxxx8H	xxxxxxxCH	xxxxxxx0H	xxxxxxx4H
xxxxxxxCH	xxxxxxxCH	xxxxxxx0H	xxxxxxx4H	xxxxxxx8H

Table 10-9. IBM BL486SX2/SX3 cache-line fill order.

## 10.4 The IBM BL486DX and BL486DX2 Microprocessors

The BL486DX and BL486DX2 are licensed second-source versions of the Cyrix Cx486DX and Cx486DX2. Table 10-10 summarizes the general features and specifications of the BL486DX and BL486DX2 products.

Product Names	IBM BL486DX and BL486DX2
Introduction Date	June 1994
Prognosis	Encouraging
Device Integration Level	Pipelined 32-bit IEU and PMMU Microcoded 80-bit floating-point unit 8K-byte unified instruction/data cache
CPU Architecture Level	Standard 486 integer and FPU instruction sets, augmented with Cyrix SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 486DX pinout
Data Bus Width	32 bits with parity (D31..D0 plus DP3..DP0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as i486DX, plus optional burst-mode data write-back capability
Cache Support	8K bytes unified I- and D-cache Two-way set associative Write-through or copy-back operation
Floating-Point Support	Built-in high-performance microcoded FPU
Operating Voltage	BL486DX, BL486DX2: 4.75 V to 5.25 V BL486DX-V, BL486DX2-V: 3.0 V to 3.6 V
Frequency Options	BL486DX: 33-, 40-, or 50-MHz operation BL486DX2: 50- and 66-MHz core frequency BL486DX-V: 33- or 40-MHz operation BL486DX2-V: 50-, 66-, or 80-MHz core freq
Clocking Regime	BL486DX, BL486DX-V: Core freq = 1 × Clkin BL486DX2: Core operating freq = 2 × Clkin
Active Power Dissipation (worst-case)	BL486DX: 6.14 W @ 5.25 V and 50 MHz BL486DX2: 6.96 W @ 5.25 V and 66 MHz (core) BL486DX-V: 2.45 W @ 3.6 V and 40 MHz BL486DX2-V: 3.42 W @ 3.6 V and 80 MHz (core)
Power-Control Features	Stopped-clock and suspend-mode operation plus Cyrix-style SMM extensions
Process Technology	0.8μ two-layer-metal CMOS
Die Size	476 mils × 480 mils (228,000 mils <sup>2</sup> )
Transistor Count	900,000 transistors
Package Options	168-pin PGA or 196-pin Plastic QFP

Table 10-10. IBM BL486DX and BL486DX2 feature summary.

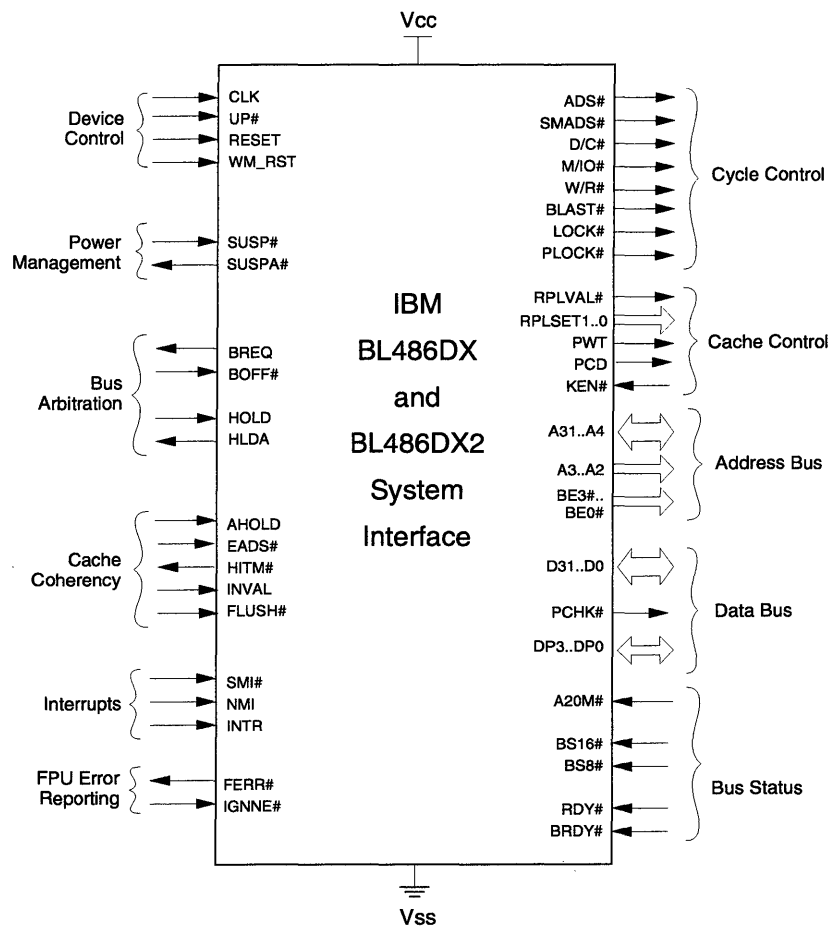


Figure 10-7. IBM BL486DX and BL486DX2 system interface.

The BL486DX and BL486DX2 contain the same core technology as the Cyrix 486 product family. The devices integrate a math coprocessor on chip, and include 8K bytes of copy-back cache. The devices implement the full 486 integer and floating-point instruction sets, plus the Cyrix-defined configuration control registers and system management instruction-set extensions. See the description of the Cx486DX and Cx486DX2 in **Chapter 9** for details.

### System Interface

The BL486DX and BL486DX2 are upwardly compatible with the standard i486DX PGA pinout, and provide the same system interface as the Cx486DX and Cx486DX2, as shown in Figure 10-7.

**Vital Statistics** The BL486DX and BL486DX2 die contains 900,000 transistors and measures  $476 \times 480$  mil ( $228,000$  mils<sup>2</sup>) on a 0.8-micron process.

If the data sheet is to be believed, the BL486DX family available in a variety of voltage, frequency, and packaging options. In practice, the only versions IBM seems to be building or promoting are the top-of-the-line BL486DX2 parts. The 5-V BL486DX2 is currently offered only in a standard 168-pin PGA package at frequencies of 50 or 66 MHz. In the 3.3-V domain, the BL486DX2-V is offered in either the PGA package or an Intel-compatible 208-lead PQFP, in variations that allow 50- or 66-MHz maximum core operation. An 80-MHz version requires a 4.0-V supply.

## 10.5 Futures

If ever there was a company that knew how to keep its future plans under wraps, IBM is it. This is due in part to IBM's long experience with the value of intellectual property and company secrets, and to a corporate culture deeply set against tipping its hand. At least as big a factor, though, is the fact that IBM is so large, has had such an erratic history, and is in such apparent internal disarray, that there may well *not be* anyone within IBM who *knows* what strategic direction the company is likely to take. Future market forces are not always knowable, and IBM has the financial wherewithal to cover its bases on any number of fronts, to redeploy its resources, and to decide after the fact whether to introduce development projects or kill them as market opportunities arise or disappear.

In the short term, though, the agreement with Cyrix should let IBM Microelectronics market the full Cyrix product line. IBM and Cyrix are redesigning the parts for a three-layer version of IBM's 0.7-micron process. This should result in a considerable die size reduction and speed increase. Such chips would be capable competitors to the IntelDX4 line—especially if the cache size were increased.

IBM also has rights to use the Cyrix CPU cores in ASICs, which could be valuable in building highly integrated chips for sub-notebook and hand-held computers for OEM customers or for the IBM PC Company. IBM will undoubtedly also second-source the Cyrix "M1" processor, which is targeted at a 0.65-micron, four-layer-metal CMOS process similar to that used by IBM for the PowerPC 603 and 604.

IBM has said it will continue to enhance the Blue Lightning product line, but more likely its focus for the Pentium-class market will be on high-end CPUs from Cyrix and NexGen. Many of the engineers who worked on Blue Lightning have reportedly been transferred to PowerPC, and it now appears that IBM's internal efforts are focused on retrofitting x86 support into the PowerPC/x86 hybrid chips—using, no doubt, technology acquired from C&T, Cyrix, and NexGen.

## 10.6 Commentary

To its credit, IBM is one of very few semiconductor makers in the world that offers foundry customers its leading-edge process technology. In addition to providing the needed capacity to Cyrix and NexGen, IBM processes should enable high clock rates and produce competitively sized die. The company also possesses the all-important Intel patent license that may provide protection from Intel's legal assaults.

IBM's agreement to manufacture microprocessors for Cyrix and market them under the IBM Microelectronics name puts IBM into direct competition with Intel. The deal is with IBM Microelectronics, not the IBM PC Company, but the PC Company will presumably be more interested in the Cyrix designs now that they will be made by IBM—at least the stability and capacity of the manufacturer shouldn't be in question. As the one-time largest maker of PCs (now fallen to #2 or #3), IBM would be a valuable design win for Cyrix's processors.

The agreement enables IBM to compete unfettered in the merchant market for x86 processors for the first time. IBM may consume internally or sell on the open market only as many chips as it supplies to Cyrix, which ensures that IBM will gain no more than a 50% market share of the Cyrix-designed chips.

Still, the IBM/Cyrix combination could easily overtake AMD for the number two spot in the x86 market. IBM will not quantify its production capacity, but claims it won't likely become production limited any time soon. Sources indicate that IBM could allocate just 10% of its fab capacity to M1-class processors and still fabricate millions of units per year.

### **Strategic Direction**

IBM's foundry agreements shed new light on its decision not to endorse Pentium. IBM would like to reduce its dependence on Intel and is perhaps also motivated by a desire to blunt Intel's power. Also, the BiCMOS Pentium would have required IBM to make significant investments to provide a compatible process. The M1 is a CMOS device and is being designed for IBM's process technology. Besides, Intel would have surely refused to let IBM sell Pentia on the merchant market.

That IBM would depend on an outside supplier for its x86 processor designs is indicative of its strategic focus on PowerPC. The x86 processors represent an opportunity to produce consid-



erable near-term revenue at high profit margins, while the PowerPC family will take longer to reach comparable volumes.

IBM's foundry deals with Cyrix and NexGen might inadvertently have a negative impact on the PowerPC. IBM's participation in the market will undoubtedly force the price of high-end x86 performance down, thereby making x86 chips stronger competitors to the PowerPC and reducing any price/performance advantages of the RISC line.

Yet, given the immense size and high profits of the x86 market, IBM may feel it has no choice but to grab on. With its large production capacity, advanced process technology, and established brand name now combined with Cyrix's designs, Intel's one-time benefactor and white knight may soon transmogrify itself into Intel's worst nightmare.

### Terminology Footnote

It's often difficult for those outside the IBM fold to digest the company's documentation. Table 10-11 is presented here as a guide to the uninitiated.

When IBM says:	What IBM really means is:
RWM	= RAM
ROS	= ROM
Module	= Integrated Circuit
Planar	= Motherboard
Hard file	= Hard-disk drive
Cache macro	= Cache
Control Store	= Microcode
Cycle Time	= 1/Clock Freq
Pin 006	= Pin 6
2.97V	= 3V
X'00001000'	= 1000H
ICE/PWI Mode	= SMM
RCVR	= Input signal
BIDI	= Bidirectional I/O signal
TSCOD	= Tri-state output signal

Table 10-11. Neophyte's IBM-to-English phrase book.

## 10.7 For More Information...

Additional technical information on the IBM 386 and 486 product lines may be found in the following publications:

- Vendor Publications**
- 1: *386SLC Microprocessor Data Sheet*. International Business Machines, 1992. (Primary 386SLC product technical reference.)
  - 2: *486SLC2 Microprocessor Data Sheet*. International Business Machines, 1993. (Primary 486SLC2 product technical reference.)
  - 3: *Blue Lightning Microprocessor Data Sheet*. International Business Machines, 2/7/94, order #MPIBLS-DBU. (Primary technical reference for the BL486SX2/SX3.)
  - 4: *Databook, 3 and 5 Volt Microprocessors*. International Business Machines Corporation, 1994, order #MPIDX2DSU-01. (Primary BL486DX and BL486DX2 product technical reference; actually a repackaged copy of the Cyrix Cx486DX databook, including the Cyrix copyright statement; quite possibly the first book in history to place its even numbered pages on the right!.)
- Microprocessor Report Articles**
- 5: *IBM to Make 386SX Variant with Cache*. MPR vol. 5 no. 17, 9/18/91, pg. 5. (Most Significant Bits item.)
  - 6: *IBM Announces Upgrade with Enhanced 386SX*. MPR vol. 5 no. 19, 10/16/91, pg. 5. (Most Significant Bits item.)
  - 7: *IBM and Intel To Jointly Develop x86 Chips\**. Michael Slater, MPR vol. 5 no. 22, 12/4/91, pg. 18. (Most Significant Bits item.)
  - 8: *IBM Previews 386SLC Follow-On*. MPR vol. 6 no. 4, 3/25/92, pg. 5. (Most Significant Bits item.)
  - 9: *IBM Selling 386SLC Processor Modules*. MPR vol. 6 no. 11, 8/19/92, pg. 5. (Most Significant Bits item.)
  - 10: *IBM Demonstrates 100-MHz "Blue Lightning"*. MPR vol. 6 no. 16, 12/9/92, pg. 5. (Most Significant Bits item.)
  - 11: *IBM Makes Its 486SLC2 Available via OEMs*. MPR vol. 7 no. 5, 4/19/93, pg. 5. (Most Significant Bits item.)
  - 12: *IBM Announces Clock-Tripled 486*. MPR vol. 7 no. 10, 8/2/93, pg. 4. (Most Significant Bits item.)
  - 13: *PowerPC May Emulate x86 in Hardware*. MPR vol. 7 no. 12, 9/13/93, pg. 3. (Most Significant Bits item.)
  - 14: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (Cover story.)
  - 15: *IBM, Intel Revise x86 Pact*. MPR vol. 8 no. 2, 2/14/94, pg. 5. (Most Significant Bits item.)

- 16: *IBM Picks Up C&T's x86 Code*. MPR vol. 8 no. 5, 4/18/94, pg. 5. (*Most Significant Bits item.*)
  - 17: *IBM and Cyrix Ink Five-Year Pact*. Michael Slater, MPR vol. 8 no. 6, 5/9/94, pg. 10. (*Feature article.*)
  - 18: *Cyrix, IBM Deliver First Fruit of Partnership*. MPR vol. 8 no. 8, 6/20/94, pg. 5. (*Most Significant Bits item.*)
  - 19: *NexGen, IBM Finally Come to Terms*. MPR vol. 8 no. 8, 6/20/94, pg. 5. (*Most Significant Bits item.*)
- Other Periodicals**
- 20: *Rethinking IBM*. Judith Dobrzynski, Business Week, 10/4/93, pg. 86. (*Business viewpoint of Lou Gerstner's first six months.*)

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



# 11

## Texas Instruments 486 Microprocessors

---

Texas Instruments entered the x86 market as a foundry and licensed second source for the full Cyrix 486 microprocessor family in 1992. As one of the oldest and largest semiconductor companies in the country, TI lent a level of manufacturing credibility and a cachet of legal protection to upstart Cyrix. Cyrix's demonstrated design skills, in turn, lent a level of technical credibility to TI's production lines.

In 1993, a rift developed between the two companies, and Cyrix shifted production to SGS-Thomson and, more recently, to IBM. TI was left with the right to continue building and selling the first-generation products, and to adapt the existing Cyrix core for use in its own proprietary designs.

In 4Q93 and 1Q94, TI introduced three families of derivative products. This chapter reviews each of the products currently in the Texas Instruments stable, both the parts second-sourced from Cyrix and its own proprietary products.

## 11.1 The TI486SLC/E and TI486SLC/E-V Microprocessors

The TI486SLC/E is Texas Instruments' designation for its second-sourced version of the (more or less) equivalent Cx486SLC/e device. The TI486SLC/E-V is Texas Instruments' version of the Cx486SLC/e-V. Table 11-1 summarizes the general features and specifications of these parts.

Product Names	Texas Instruments TI486SLC/E and TI486SLC/E-V
Introduction Date	TI486SLC/E: October 1992 TI486SLC/E-V: January 1993
Prognosis	Embedded-ridden
Device Integration Level	Pipelined 32-bit IEU and PMMU 1K-byte unified instruction/data cache
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix-style SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as 386SX
Cache Support	1K bytes unified I- and D-cache Direct mapped or two-way set associative Write-through operation only
Floating-Point Support	Optional external 387SX-class FPU
Operating Voltage	TI486SLC/E: 4.75 V to 5.25 V TI486SLC/E-V: 3.0 V to 3.6 V
Frequency Options	TI486SLC/E: 25-, 33-, or 40-MHz core operation TI486SLC/E-V: 25-MHz core operation
Clocking Regime	Core operating frequency = $1/2 \times \text{Clkin}$
Active Power Dissipation (worst case)	TI486SLC/E: 3.0 W @ 5.0 V and 33 MHz TI486SLC/E-V: 0.95W @ 3.3 V and 25 MHz
Power-Control Features	Stopped-clock and suspend-mode operation plus Cyrix-style SMM extensions
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils $\times$ 410 mils (110 mm <sup>2</sup> )
Transistor Count	600,000 transistors
Package Options	100-pin PQFP
Notes	Contains the same die as the Cx486SLC/e

Table 11-1. TI486SLC/E and TI486SLC/E-V feature summary.

These parts are fabricated under license from Cyrix, using the Cyrix database. They provide essentially the same on-chip resources, cache configurations, instruction-set extensions,

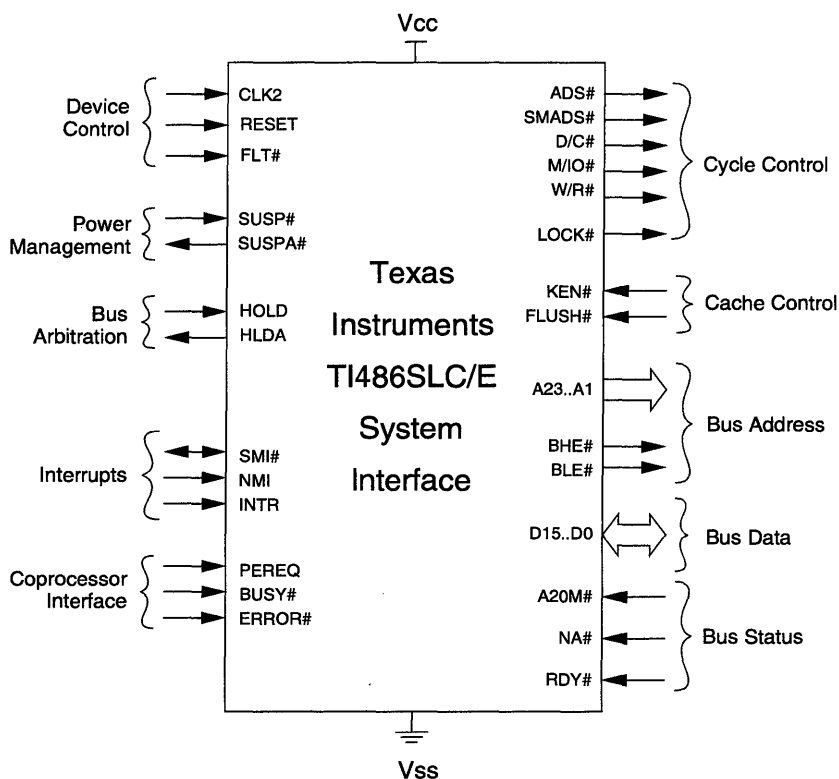


Figure 11-1. TI486SLC/E and TI486SLC/E-V system interface.

device-configuration registers, system interfaces, package types, and pinouts as the originals. Neither device provides any on-chip support for floating-point operations, although each can be used in conjunction with a standard 386SX-class floating-point coprocessor.

### System Interface

Figure 11-1 shows that the TI486SLC/E and TI486SLC/E-V provide a system interface derived from a standard 386SX. Table 11-2 lists each of the TI486SLC/E signals not defined for the standard 386SX pinout.

Each of these signals performs the same function as the corresponding signal on a Cx486SLC/e device; refer to **Chapter 9: Cyrix 486 Microprocessors** for technical details on the Cyrix Cx486SLC/e-family pinout.

### Compatibility

The Texas Instruments devices' system interface does differ, however, from their Cyrix forebears' in two ways, both minor.

Symbol	Direction	Signal Name/Function	TI486SLC/E PQFP Pin #	Replaces 386SX PQFP Signal
A20M#	In	Address-bit 20 mask	31	N.C.
KEN#	In	Cacheability enabled for requested data	29	N.C.
FLUSH#	In	Flush cache data	30	N.C.
SMI#	I/O	SMM interrupt request/active	47	N.C.
SMADS#	Out	SMM memory address strobe	20	N.C.
SUSP#	In	Suspend normal execution	43	N.C.
SUSPA#	Out	Suspend mode acknowledge	44	N.C.

Table 11-2. TI486SLC/E special interface signals.

First, for some inexplicable reason, Texas Instruments chose not to bond out the RPLVAL# or RPLSET signals defined by the Cyrix devices. In the Cyrix design, RPLVAL# and RPLSET make it possible for system designers to build a set-associative second-level cache that maintains an inclusion relationship with the on-chip cache. Doing so would increase processor efficiency, since modifications to shared external memory would not need to flush the on-chip cache unless the second-level cache circuitry detects a hit.

TI may have considered these two signals to be of minimal value, since most PC chip sets designed for 486-class bus interfaces do not make use of these pins. Or perhaps the Texas Instruments licensing agreement with Cyrix demanded that minor differences be introduced in device functionality. Or possibly TI needed additional pins for internal testing purposes, and chose for some reason to appropriate these two. Whatever the rationale, as a result of this differentiation, the TI devices may not be directly interchangeable with certain systems designed according to the Cyrix specifications.

A second distinction between the TI and Cyrix designs concerns the interpretation of bit 0 of device-configuration register CCR1. In the Cyrix parts this bit had served to optionally enable RPLVAL# and RPLSET. In the TI family the bit is, of course, undefined, and reserved for future use. In principle, this may preclude the use of TI parts with certain BIOS ROMs or configuration utilities intended for Cyrix devices, though systems based on the TI devices would presumably not attempt to enable this function.



**Vital Statistics** The die used by the TI486SLC/E and TI486SLC/E-V are the same as the Cyrix equivalents and are fabricated using the same 0.8-micron, two-layer-metal CMOS process technology. The die contains approximately 600,000 transistors, and measures  $410 \times 410$  mils. Each is housed in a standard 100-pin PQFP package. The (5-V) TI486SLC/E device is available in 25-, 33-, and 40-MHz variations. The (3.3-V) TI486SLC/E-V is only available at 25 MHz.

## 11.2 The TI486DLC/E and TI486DLC/E-V Microprocessors

The TI486DLC/E and TI486DLC/E-V are Texas Instruments' designations for enhanced versions of the Cyrix Cx486DLC device. Table 11-3 summarizes the general features and specifications of these parts.

Product Names	Texas Instruments TI486DLC/E and TI486DLC/E-V
Introduction Date	TI486DLC/E: October 1992 TI486DLC/E-V: March 1993
Prognosis	Terminal
Device Integration Level	Pipelined 32-bit IEU and PMMU 1K-byte unified instruction/data cache
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix-style SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 386DX pinout
Data Bus Width	32 bits (D31..D0)
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same as 386DX
Cache Support	Same as TI486SLC/E
Floating-Point Support	Optional external 387DX-class FPU
Operating Voltage	TI486DLC/E: 4.75 V to 5.25 V TI486DLC/E-V: 3.0 V to 3.6 V
Frequency Options	TI486DLC/E: 33- or 40-MHz core operation TI486DLC/E-V: 25- or 33-MHz core operation
Clocking Regime	Core operating frequency = 1/2 x C <sub>lkin</sub>
Active Power Dissipation	TI486DLC/E: 3.5 W @ 5.0 V and 40 MHz TI486DLC/E-V: 1.25 W @ 3.3 V and 33 MHz
Power-Control Features	Stopped-clock and suspend-mode operation plus Cyrix-style SMM extensions
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	410 mils $\times$ 410 mils (110 mm <sup>2</sup> )
Transistor Count	600,000 transistors
Package Options	132-pin PGA
Notes	Contains same die as TI486SLC/E

Table 11-3. TI486DLC/E and TI486DLC/E-V feature summary.

The TI486DLC/E and TI486DLC/E-V are also fabricated from a design database and mask set provided by Cyrix, but they include features not present on the original (now discontinued) Cyrix products. Specifically, the instruction set and pinout enhancements incorporated into the "e" versions of the

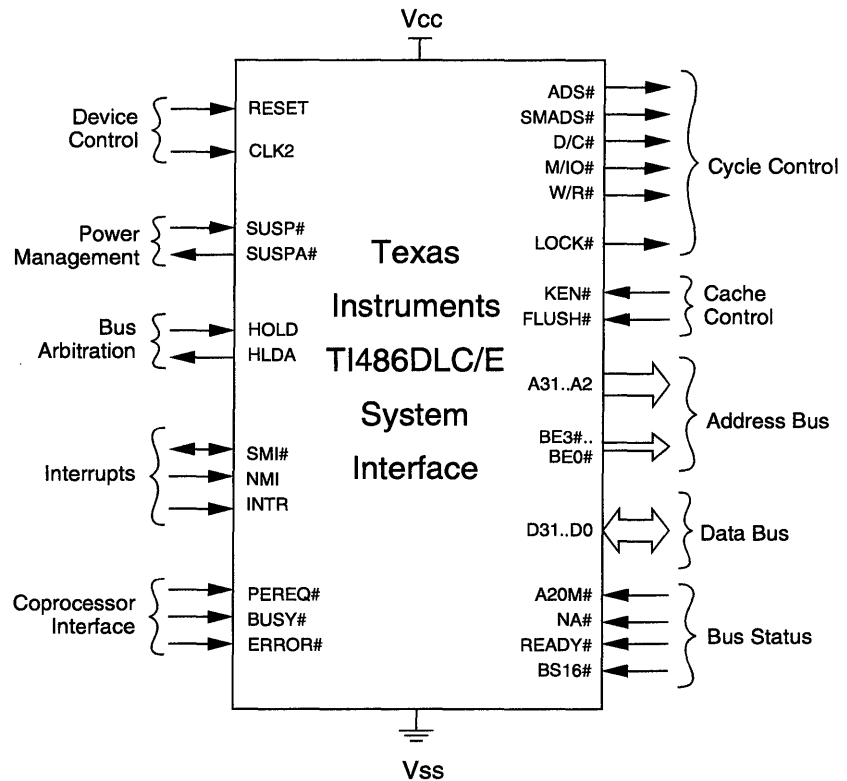


Figure 11-2. TI486DLC/E and TI486DLC/E-V system interface.

Cx486SLC-family devices are enabled in the TI486DLC/E family as well.

### System Interface

The TI486DLC/E and TI486DLC/E-V system interface closely resembles that of a standard 386DX, as shown in Figure 11-2.

As the “E” suffix might imply, the system interface for these parts supports the same enhancement signals defined for the TI486SLC/E. Table 11-4 summarizes the names and functions of the TI486DLC/E signals not provided by the standard 386DX pinout, and the pins to which each is assigned.

Each of these signals performs the same function as on a Cx486SLC/e or TI486SLC/E device. Consult the related signal descriptions in **Chapter 9** for details. Once again, though, the TI chips do not bond out the RPLVAL# and RPLSET signals defined by the original Cyrix design.

### Vital Statistics

The TI486DLC/E and TI486DLC/E-V contain the same die, with the same design characteristics, as the TI486SLC/E and

Symbol	Direction	Signal Name/Function	TI486DLC/E PGA Pin #	Replaces 386DX PGA Signal
A20M#	In	Address-bit 20 mask	F13	N.C.
KEN#	In	Cacheability enabled for requested data	B12	N.C.
FLUSH#	In	Flush cache data	E13	N.C.
SMI#	I/O	SMM interrupt request/active	C7	N.C.
SMADS#	Out	SMM memory address strobe	C6	N.C.
SUSP#	In	Suspend normal execution	A4	N.C.
SUSPA#	Out	Suspend mode acknowledge	B4	N.C.

Table 11-4. TI486DLC/E special interface signals.

TI486SLC/E-V. Each is housed in a standard 132-pin PGA package. The former, 5-V device is available in 33- and 40-MHz versions. The latter, 3.3-V, variation comes in 25- and 33-MHz flavors.

## 11.3 The TI486SXLC and TI486SXLC2 Microprocessors

The TI486SXLC and TI486SXLC2 are TI's first internally designed derivatives of the Cyrix CPU core, and the first parts to include original features. The devices expand the on-chip cache to 8K bytes and add clock-doubling capability within a 486SLC (extended 386SX) pinout. Table 11-5 summarizes the general features and specifications of these parts.

Product Names	Texas Instruments TI486SXLC and TI486SXLC2
Introduction Date	November 1993
Prognosis	Production
Device Integration Level	Pipelined 32-bit IEU and PMMU 8K-byte unified instruction/data cache Optional clock-doubler circuitry
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix-style SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 386SX pinout
Data Bus Width	16 bits (D15..D0)
Physical Addressability	16MB (Address A23..A1 plus BHE#, BLE#)
Data-Transfer Modes	Same as 386SX
Cache Support	8K bytes unified I- and D-cache Two-way set associative Write-through operation only
Floating-Point Support	Optional external 487SX-class FPU
Operating Voltage	TI486SXLC/SXLC2: 4.75 V to 5.25 V TI486SXLC/SXLC2-V: 3.0 V to 3.6 V
Frequency Options	TI486SXLC: 33-MHz core operation TI486SXLC2: 50-MHz core operation TI486SXLC-V: 33-MHz core operation TI486SXLC2-V: 40-MHz core operation
Clocking Regime	TI486SXLC: Core operating freq = $1/2 \times \text{Clkin}$ TI486SXLC2: Core freq = $1/2 \times$ or $1 \times \text{Clkin}$
Active Power Dissipation	TI486SXLC2: 2.35 W @ 5.0 V and 50 MHz (w.c.) TI486SXLC2-V: 1.2 W @ 3.3 V and 40-MHz (w.c.)
Power-Control Features	Stopped-clock and suspend-mode operation plus Cyrix-style SMM extensions
Process Technology	0.8 $\mu$ two-layer-metal CMOS
Die Size	130 mm <sup>2</sup>
Transistor Count	900,000 transistors
Package Options	100-lead PQFP

Table 11-5. TI486SXLC and TI486SXLC2 feature summary.

With their larger caches and clock-doubled cores, these devices take, at least for now, the performance lead among merchant-market processors in the 16-bit 386SX package.

While the core processor logic can run at speeds up to 50 MHz, the bus interfaces are not spec'd for operation faster than 33 MHz. In order to obtain maximum performance, then, system designers must choose between running the core and bus at the same medium-high frequency, or reducing the bus frequency somewhat and doubling the internal clock.

### **Clock Circuitry**

The TI chips are fully static. In contrast to the Cyrix designs, TI's clock-doubling circuitry incorporates an analog phase-locked loop (PLL), similar to Intel's original i486DX2. As a result, the external clock input cannot change frequency as rapidly as the Cyrix parts without wreaking havoc on PLL synchronization. The TI chips allow the clock-doubling function to be software configured, however, so software can switch the chip out of clock-doubled mode to reduce power, and then redouble the clock as needed for maximum performance.

The clock can also be stopped at the output of the PLL to put the chip into a low-power standby mode without actually stopping the oscillator input. The clock-multiplier circuit itself thus continues to run (and consume power), but operation can resume nearly instantly, without incurring the oscillator start-up and stabilization delays that would be required if the PLL were itself to be stopped.

### **Vital Statistics**

The TI486SXLC-family products contain about 900,000 transistors, with a die size of approximately 130 mm<sup>2</sup> (200,000 mils<sup>2</sup>) in a 0.8-micron, two-level-metal CMOS process. This is nearly twice the area of the 0.8-micron i486SX, which benefits from tighter circuit packing and a third metal signal-routing layer.

Each part includes both 3.3-V and 5-V versions. At 5 V, the TI486SXLC operates up to 33 MHz, or 50 MHz (core frequency) for the TI486SXLC2. At 3.3 V, the TI486SXLC-V device runs at up to 40 MHz, or 50 MHz (internal) with clock doubling.

Typical power consumption at 40 MHz with a 5-V supply is 2.5 W or less; at 3.3 V and 33 MHz, typical dissipation is under 1 W. By comparison, Intel's i486SX has a typical power consumption of 990 mW at 3.3 V and 33 MHz, or just under 3 W at 5 V and 33 MHz. Thus, the TI and Intel chips consume similar power to deliver comparable performance. With the clock stopped, typical current drain drops below 20  $\mu$ A.

## 11.4 The TI486SXL and TI486SXL2 Microprocessors

The TI486SXL and TI486SXL2 are TI's answers to the Intel/AMD 486SX and Cyrix Cx486S families. Each provides clock doubling and a reasonable complement of on-chip cache in a 486SX-compliant pinout. Table 11-6 summarizes the general features and specifications of these parts.

Product Names	Texas Instruments TI486SXL and TI486SXL2
Introduction Date	November 1993
Prognosis	Production
Device Integration Level	Pipelined 32-bit IEU and PMMU 8K-byte unified instruction/data cache Optional clock-doubler circuitry
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix-style SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Augmented compatible 486SX pinout
Data Bus Width	32 bits (D31..D0) + per-byte parity
Physical Addressability	4GB (Address A31..A2 plus BE3#..BE0#)
Data-Transfer Modes	Same transfer modes as the 386DX, although packaged with a 486SX-class pinout
Cache Support	8K bytes unified I- and D-cache Four-way set associative Write-through operation only
Floating-Point Support	None; requires 487-style replacement CPU
Operating Voltage	TI486SXL/SXL2: 4.75 V to 5.25 V TI486SXL-V/SXL2-V: 3.0 V to 3.6 V
Frequency Options	TI486SXL: 33-MHz core operation TI486SXL2: 50-MHz core operation TI486SXL-V: 33-MHz core operation TI486SXL2-V: 40-MHz core operation
Clocking Regime	TI486SXL: Core operating freq = 1 × Clkin TI486SXL2: Core freq = 1 × or 2 × Clkin
Active Power Dissipation (worst case)	TI486SXL2: 3.3 W @ 5.0 V and 50-MHz core (w.c.) TI486SXL2-V: 0.9 W @ 3.3 V and 40-MHz (w.c.)
Power-Control Features	Stopped-clock and suspend-mode operation plus Cyrix-style SMM extensions
Process Technology	0.8μ two-layer-metal CMOS
Die Size	130 mm <sup>2</sup>
Transistor Count	900,000 transistors
Package Options	100-lead PQFP
Notes	Contains the same die as TI486SXL family

Table 11-6. TI486SXL and TI486SXL2 feature summary.

The TI486SXL and TI486SXL2 contain the same die as the TI486SXLC and TI486SXLC2. Despite the use of a 486SX-compatible pinout and a full 8K bytes of cache, the parts fall short of “true” 486SX implementations in two respects. First, the CPU core, which is the same as that in the original Cx486SLC and Cx486DLC, omits the dedicated address adder provided by the Intel and AMD designs and is thus somewhat slower at the same core frequency.

More important, the TI bus interface does not support burst-mode transfers. Essentially, these chips implement a 386DX-like bus interface in a 486SX pinout. Using burst mode lets Intel and AMD 486SX and 486SX2 devices sustain nearly twice the bus bandwidth in the same system motherboard.

**Vital Statistics**

The TI486SXL and TI486SXL2 contain the same die as the TI486SXLC and TI486SXLC2, repackaged in a 168-pin PGA housing. At 5 V, the TI486SXL allows operation up to 33 MHz, or 50 MHz (core frequency) for the TI486SXL2. At 3.3 V, the TI486SXL-V supports clock rates up to 40 MHz, or 50 MHz (internal) with clock-doubling enabled.



## 11.5 The TI “Rio Grande” Processor Chip Set

“Rio Grande” was the code name for a highly integrated TI processor chip set, including a 486-class CPU and two peripheral devices, designed for notebook systems. Texas Instruments completed product development, formally announced the family, built and distributed samples, and then waited in vain for customers to appear. None did. After several months with no industry interest TI quietly pulled the plug and let the product line die. Table 11-7 summarizes the general features and specifications of the Rio Grande CPU.

Product Name	Texas Instruments TI “Rio Grande”
Introduction Date	February 1994
Prognosis	Stillborn
Device Integration Level	Pipelined 32-bit IEU and PMMU 8K-byte unified instruction/data cache On-chip DRAM controller and buffers On-chip PCI interface control logic and drivers
CPU Architecture Level	Standard 486 integer instruction set plus Cyrix-style SMM extensions
Core Technology	Cyrix-designed static 486 core
Pinout	Custom
Data Bus Width	32-bit PCI system bus Separate 32-bit local DRAM bus
Physical Addressability	4GB (PCI protocol)
Data-Transfer Modes	Custom
Cache Support	8KB on-chip combined I- and D-cache Four-way set associative Write-through operation only
Floating-Point Support	Optional 386DX-class coprocessor
Operating Voltage	4.75 V to 5.25 V or 3.0 V to 3.6 V
Frequency Options	66-MHz core operation
Clocking Regime	Core operating freq. = 1 × Clkin PCI bus interface = 1/2 × Clkin
Active Power Dissipation	N.A.
Power-Control Features	Static operation plus Cyrix-style SMM
Process Technology	0.65 $\mu$ three-layer-metal CMOS
Die Size	115 mm <sup>2</sup>
Transistor Count	N.A.
Package Options	208-lead PQFP

Table 11-7. TI “Rio Grande” CPU feature summary.

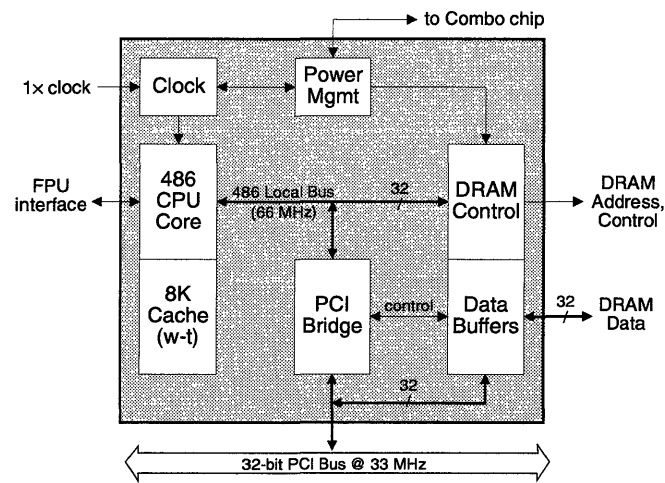


Figure 11-3. TI "Rio Grande" CPU block diagram.

The Rio Grande CPU was based on the same Cyrix 486 core as TI's other processors. Its cache had the same specifications as that on a conventional 486-class CPU: 8 Kilobytes of capacity, a 16-byte line size, four-way set associativity, write-through operation, and an LRU replacement policy. In addition, the integrated processor chip contained DRAM memory-control logic, on-chip power management circuitry, and a direct-drive PCI bus interface (see Figure 11-3).

Rio Grande did not contain an FPU, although a standard 387DX-class math coprocessor could be added externally. Nor did the part provide any direct support for an external second-level cache; it was thought that small, low-end notebook systems wouldn't need such accouterments.

Note from Figure 11-3 that the DRAM controller and PCI bridge connect to the CPU through an internal, conventional 486-style bus, as though the core modules were implemented with discrete components. Since the Cyrix CPU core could not support burst-mode transfers, refilling a cache line took the equivalent of at least 24 core clock cycles, vs 10 for a 486DX2. The "local" bus was, however, clocked at the full CPU speed vs the half-speed system bus.

Depending on the mix of read and write transactions, the 66-MHz Rio Grande processor bus should have had about the same performance as an Intel 33-MHz 486 bus. While this does not seem impressive, it means that the faster clock speed, made possible by the fact that the entire CPU local bus is contained in

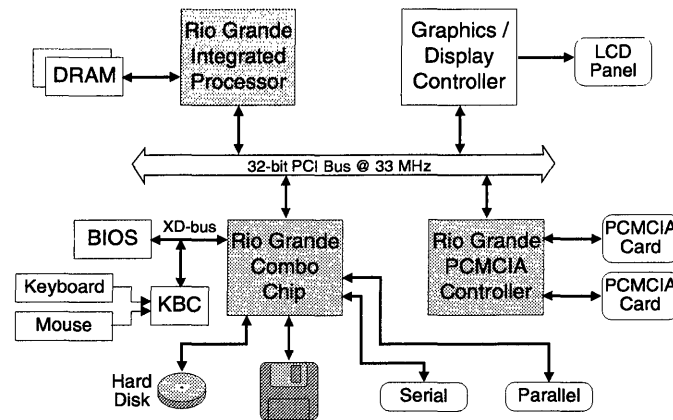


Figure 11-4. TI "Rio Grande" system interface.

the Rio Grande processor chip, offset the performance loss caused by the lack of burst-mode transactions. A 66-MHz Rio Grande should have been similar to a 50-MHz DX2 in performance on system-level benchmarks.

The Rio Grande processor required a full-speed clock input—that is, a 50- or 66-MHz oscillator for 50- or 66-MHz core operation. An on-chip PLL further doubled the clock frequency to obtain internal timing signals. By combining all high-frequency components (i.e., those that ran faster than 33 MHz) on the processor chip, Rio Grande reduced the need for fast signal routing on the motherboard. Still, the high-speed clock input was a cause for concern in system designs that required FCC emissions certification.

### Support Logic

The Rio Grande CPU operates as part of a three-chip set. An "I/O combo" chip was to provide system logic and handle the low-speed I/O interfaces, including serial, parallel, and IDE ports. The third chip in the family supported two PCMCIA slots. All three were directly interconnected via a 33-MHz PCI bus, as shown in Figure 11-4.

The combo chip contained most of the system logic and standard peripherals needed for a simple PC system, including:

- A PCI bus arbiter
- PC/AT system logic (DMA, interrupts, etc.)
- One serial port, compatible with the National 16550
- One Centronics-compatible enhanced parallel port

- A fast IDE (hard-disk) interface
- An 82077SL-compatible floppy-disk controller
- A real-time clock
- 128 bytes of battery-backed SRAM
- An XD bus to support external peripheral expansion

The combo chip also contained a power-management unit with six power states, controlled by activity timers and software intervention. The chip could monitor each of the integrated peripherals, the VGA frame buffer, the PCI bus, two off-chip peripherals, and four interrupt requests. The combo chip had a pulse-width-modulated output that controls the brightness of the LCD backlight. The system can resume processing after a power-down due to a variety of interrupts and alarms.

The PCMCIA controller complied with PCMCIA version 2.0 and ExCA version 4.1. It was register-compatible with Intel's 82365SL DF. The PCI bus interface could assemble 8- or 16-bit data from the cards and transmits it as 32-bit words. Unlike most earlier controllers, the PCMCIA chip provided separate, electrically isolated buffers to allow "hot" card insertion and removal. For additional expansion, up to four controller chips can be combined in a single system.

For system management, the processor implemented the Cyrix SMM protocol. All three chips were fully static and together drew less than 100  $\mu$ A with the clock stopped.

### **Vital Statistics**

The Rio Grande processor was designed for a 0.65-micron, three-layer-metal CMOS process, representing a 10% shrink from the process used for the TI486SXL. The die size was approximately 115 mm<sup>2</sup>. The processor used a modular design strategy, with the memory and PCI controllers implemented as gate arrays surrounding the custom CPU core; a fully custom design might have been more compact but would have taken longer to design.

Each of the chips in the Rio Grande family operated at 3.3 V or 5 V, as did the PCI bus that connects them. Even at the lower voltage, the CPU ran at 50 or 66 MHz, and the PCI bus could be clocked at up to 33 MHz (one-half of the CPU speed). Either 3.3-V or 5-V DRAMs could be used, and the PCMCIA controller supports cards at either voltage. Each of the three chips in the set was packaged in a 208-lead PQFP.

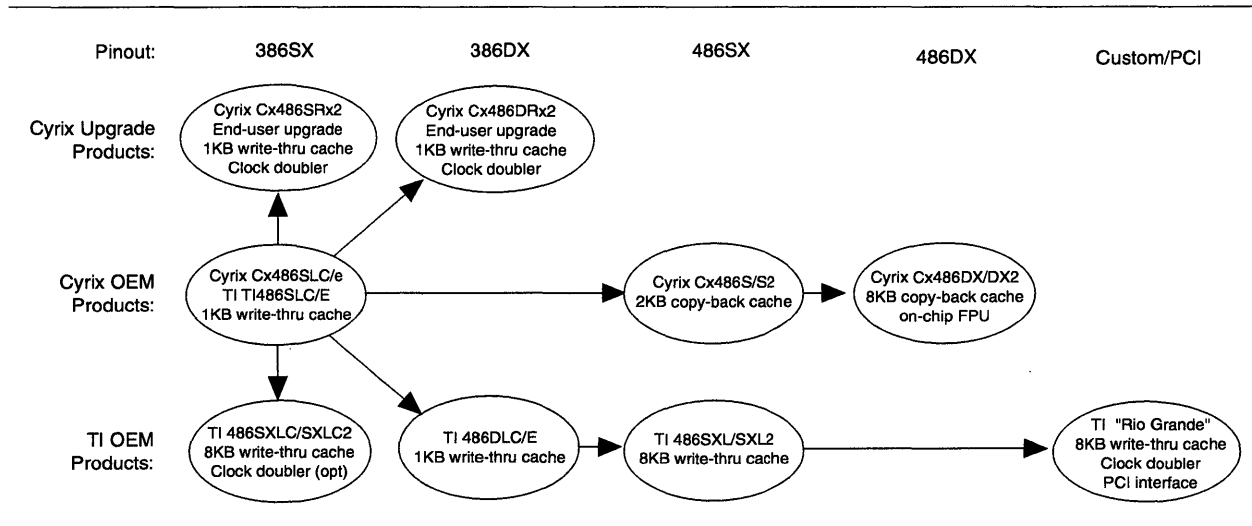


Figure 11-5. Divergence between TI and Cyrix product strategies.

## 11.6 Commentary

TI seems to be stuck in a reactionary mode. When Cyrix first began dropping hints about its upcoming M1, TI responded immediately by saying the company has its own design team working on a next-generation CPU. Little has been heard from the TI project since. TI sources say, however, that a large project remains underway to develop a next-generation x86 core.

With their respective new product introductions, it's clear that Cyrix and TI march to the beat of different strategic drummers. At each new generation, Cyrix has continued to enhance the core logic of its 486 product line to include more sophisticated cache features, bus protocols, and floating-point capabilities. The original Cx486SLC had a 1K-byte, write-through cache and a 386SX-compatible pinout. The Cx486DLC, provided a 32-bit bus in a 386DX-compatible pinout. Cyrix added a 2K-byte copy-back cache for its Cx486S-series, and an 8K-byte copy-back cache and FPU for its 486DX-series parts.

TI's enhancements, in contrast, have been limited to cosmetic changes to the pinout, brute-force expansions of the cache, and increased system-level integration. Figure 11-5 shows the respective road maps of the Cyrix and Texas Instruments product expansions. As a rule, the marketplace has seemed to reward more aggressive designs, and remain underwhelmed by brute-force engineering feats.

TI lacks in-house expertise in x86-family floating-point technology, and can thus neither incorporate this function within its processor cores nor bundle coprocessors with their integer CPUs at a discounted price. As Intel attempts to increase demand for floating-point capability by emphasizing the performance of its i486DX2 and IntelDX4 chips, and as Cyrix promotes the fact that its FPU is even faster than Intel's, it seems TI can only counter with larger caches and more grandiose chip sets. As Intel discovered with the i386SL and i486SL, however, system integration features will not support a significant price premium.

Ironically, TI's best chance of success may lie with the TI486SXLC2—a device seemingly severely handicapped by its primitive, 16-bit bus interface. Because of its larger cache and double-speed clock, the TI486SXLC2 will generally perform better than competing chips with the same pinout. Vendors that favor the small package size and lower cost of the 386SX pinout for subnotebook PCs, or that wish to extend the life of existing 386SX-based hardware designs, will find TI able to deliver performance superior to that possible from Intel, AMD, or Cyrix CPUs.

### **The i486SL Redux?**

Integration of system logic and other functions with the CPU a la Rio Grande has not proven to be terribly lucrative, as Intel, AMD, Chips and Technologies, and VLSI Technology keep discovering. Intel abandoned its otherwise appealing i386SL and i486SL integration strategy after discovering that combining space-consuming, low-value-added system logic and drivers to an already aggressive chip layout was inherently cost-ineffective.

Nevertheless, there were many who hoped TI would make a go of the CPU/chip-set business. TI has done a better job than Intel of integrating memory and bus interfaces efficiently onto the processor die. Finally, by choosing to integrate PCI instead of ISA, TI let system designers add higher-speed peripherals. Rio Grande might thus have succeeded where the i486SL failed.

And while high-integration processor chip sets may not bring in as many dollars per silicon nanoacre as a sexy leading-edge CPU, they may still be very attractive in comparison to NAND gates, calculator chips, and other commodity semiconductors. TI is used to operating on considerably lower profit margins than Intel or AMD, and has little presence at the higher ends of the microprocessor market. The company might therefore be

more willing to eat the higher costs associated with putting system logic on chip.

TI's biggest problems have come from its legal struggle with Cyrix. TI's derivative x86 products rely on core designs licensed from Cyrix. With TI losing rights to key future products, it may have no future in the business unless its internal core development comes to a successful fruition.

## 11.7 For More Information...

Additional technical information on TI processors may be found in the following publications:

- |                                       |  |
|---------------------------------------|--|
| <b>Vendor Publications</b>            | 1: <i>TI486 Microprocessor Reference Guide</i> . Texas Instruments, 1993, order #SRZU005A. ( <i>Primary technical reference.</i> ) |
| <b>Microprocessor Report Articles</b> | 2: <i>Texas Instruments Announces 486SLC Plans</i> . MPR vol. 6 no. 7, 5/27/92, pg. 4. ( <i>Most Significant Bits item.</i> )      |
|                                       | 3: <i>TI Announces Production of 486SLC/DLC</i> . MPR vol. 6 no. 14, 10/28/92, pg. 4. ( <i>Most Significant Bits item.</i> )       |
|                                       | 4: <i>TI Announces 3.3V, 33-MHz 486DLC</i> . MPR vol. 7 no. 4, 3/29/93, pg. 4. ( <i>Most Significant Bits item.</i> )              |
|                                       | 5: <i>AMD Loses OmniBook Socket to TI</i> . MPR vol. 7 no. 12, 9/13/93, pg. 5. ( <i>Most Significant Bits item.</i> )              |
|                                       | 6: <i>Texas Instruments Extends 486 Line</i> . Michael Slater, MPR vol. 7 no. 15, 11/15/93, pg. 14. ( <i>Feature article.</i> )    |
|                                       | 7: <i>PC Market Centers on Growing 486 Family</i> . Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. ( <i>Cover story.</i> )      |
|                                       | 8: <i>TI Shows Integrated x86 CPU for Notebooks</i> . Linley Gwennap, MPR vol. 8 no. 2, 2/14/94, pg. 1. ( <i>Cover story.</i> )    |
|                                       | 9: <i>Number Two Doesn't Always Try Harder</i> . Linley Gwennap, MPR vol. 8 no. 3, 3/7/94, pg. 3. ( <i>Editorial.</i> )            |





# Part IV: Pentium-Class Processors

---

Intel formally announced its long-delayed and eagerly-awaited Pentium microprocessor in March of 1993. While the introduction itself was relatively low-key, the technical trade press and OEM system vendors responded to the announcement with more hype, hoopla, and fanfare than they had for any microprocessor in history. Though initial system shipments didn't begin until 2Q93, by the year's end Pentium-based PCs were thought to be shipping faster than any RISC-based workstation made. By the end of 1994, the installed base of Pentium PCs surpassed the combined total shipments of all RISC workstations to date.

The Pentium design uses a number of novel techniques to deliver more than twice the performance of competing 386- and 486-class devices. While AMD, Cyrix, and other competing vendors have announced plans to introduce future products with Pentium-class performance, the first such product sampled was the NexGen Nx586 microprocessor.

**Part IV** of this report details the Pentium and NexGen devices, including their implementations, system interfaces, architectural extensions, and performance. It has two chapters:

**Chapter 12: The Intel Pentium Family**

**Chapter 13: The NexGen Nx586 Microprocessor**



# 12

## The Intel Pentium Family

The Pentium microprocessor family is Intel's highest-performance implementation of the x86 architecture. On integer programs it delivers roughly twice the performance of an i486DX2 at the same internal clock frequency, and is up to five times faster on optimized floating-point code.

**Overview** From a hardware perspective, Pentium's key features include a superscalar execution pipeline that can execute up to two integer instructions during every clock cycle, an 8K-byte instruction cache, a separate 8K-byte write-back data cache, and a high-performance pipelined floating-point unit.

A newly added branch target buffer (BTB—also called a branch history table) caches the destination address for previously encountered branches, along with bits that record the history of past branching patterns. The BTB can significantly reduce the latency of all branches, jumps, and CALL instructions, such that correctly predicted branches execute in a single cycle with no pipeline delays.

The system interface is also enhanced. A 64-bit external data bus with pipelined burst-mode transfers more than doubles the bus bandwidth of a 486 at a given frequency. To improve system integrity and allow the design of fault-tolerant systems, automatic parity checking is performed for the address and data buses, all internal cache data and TLB RAM arrays, and the internal microcode ROM.

From a software perspective, Pentium implements essentially the same user-mode architecture, programming model, and

instruction set as the 486, which is, in turn, essentially the same as the 386 user-mode architecture. At the system level, though, a number of functions have been enhanced. A handful of new instructions have been added to support new hardware functions and new operating modes. Several new system registers have been defined, and a number of bits that had been reserved in earlier x86 processors now perform new functions.

The Pentium PMMU now supports larger page sizes, and emulation of virtual-mode 8086 programs has been improved. Pentium was also the first high-performance desktop microprocessor to support the System Management Mode (SMM) functions first introduced on power-miserly processors for notebooks and other battery-based applications, although similar functions have since migrated to high-end 486 devices from Intel, AMD, and Cyrix.

Figure 12-1 shows a block diagram of the Pentium core. Intel estimates about 30% of the Pentium transistor budget was devoted to compatibility with the x86 architecture. Much of this overhead is probably in the microcode ROM, the instruction decode and control unit, and the adders in the two address generators, but there are other effects of the complex instruction set. For example, the more frequent occurrence of memory references in x86 programs compared to RISC code mandated the implementation of a novel dual-access data cache described below.

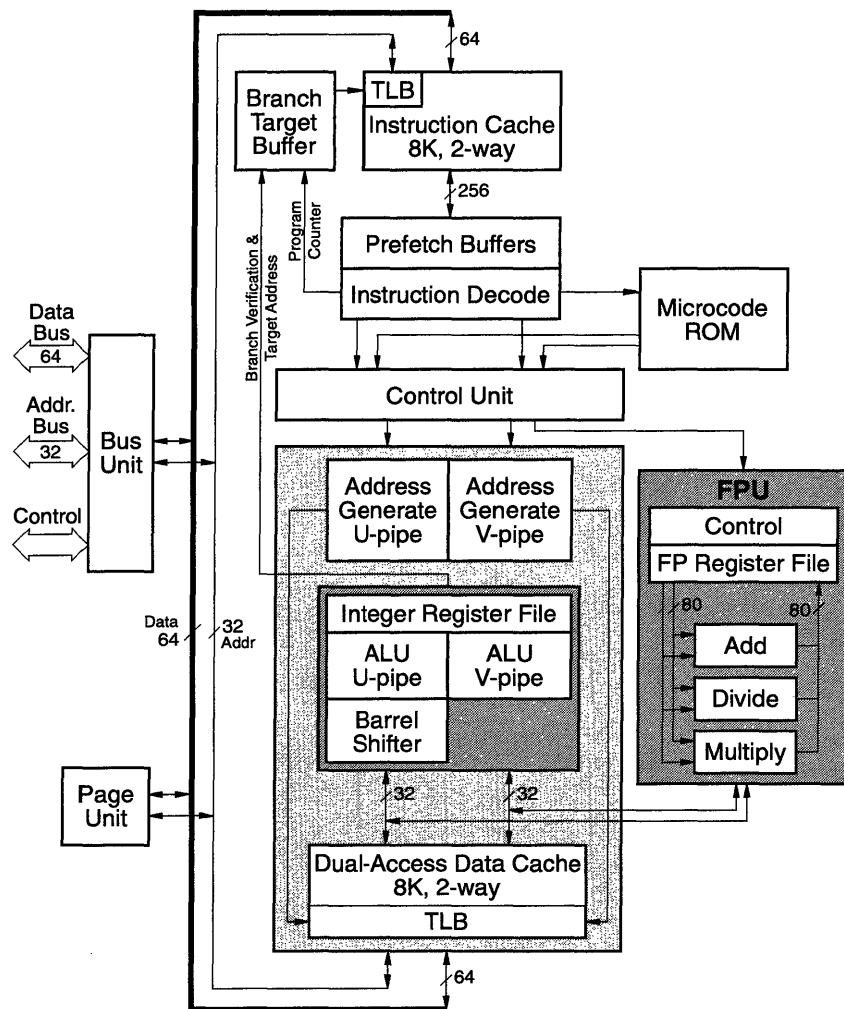


Figure 12-1. Intel Pentium microprocessor block diagram.

### Pipeline Operation

The pipeline design, shown in Figure 12-2 consists of five stages: Fetch, Decode 1, Decode 2, Execute, and Write-Back. The first two stages simultaneously process a pair of instructions. The last three stages are duplicated, forming two separate pipelines, which Intel designates the U-pipe and the V-pipe. Each pipeline contains a full ALU, and each can execute integer, branch, and control operation. When certain conditions (detailed below) are met, two integer instructions can be executed during every clock cycle.

Figure 12-3 is a detailed representation of the Pentium data-path pipelines. Even though Pentium has two integer pipelines, the basic five-stage pipeline structure is the same as the 486.

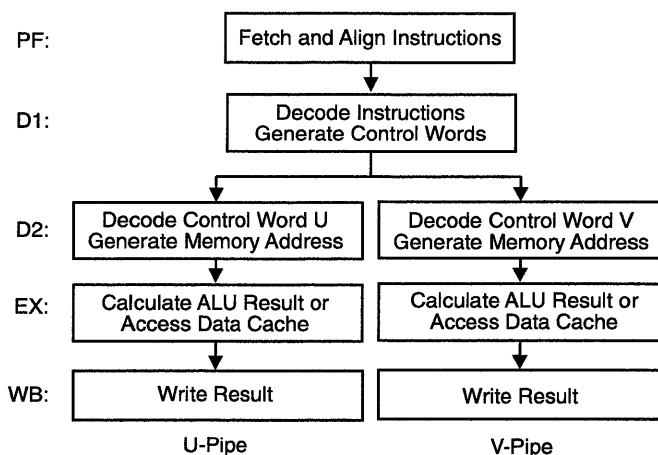


Figure 12-2. Intel Pentium integer unit pipeline operation.

In Figure 12-3, circles containing an equal sign represent logic that detects resource conflicts. Situations such as register dependencies that require serial execution are detected by these blocks. When a conflict is detected, the instruction dispatched to the U pipeline has priority. The U-pipe can execute a slightly wider range of instructions than the V-pipe, and consequently acts as the primary pipeline whenever two instructions cannot be issued simultaneously.

The pipelines are in many ways similar to the 486: instructions are first prefetched from cache into an instruction buffer, then decoded in two pipeline stages in order to accommodate the semantically rich (i.e., complex) x86 instruction set. The final two stages are the traditional execution and write-back pipeline phases.

Even though Pentium has the same high-level structure as the 486 pipeline, there are many subtle implementation differences. For example, total prefetch capacity has been increased by a factor of four, and the address adders in the D2 stage have four inputs instead of three to permit even the most complex addressing modes to complete in a single clock cycle.

**Prefetch Stage.** The Prefetch (PF) stage retrieves instructions from a dedicated 8K-byte instruction cache. (The 486, in contrast, provides a single 8K-byte cache for both instructions and data.) Separating the instruction and data caches improves

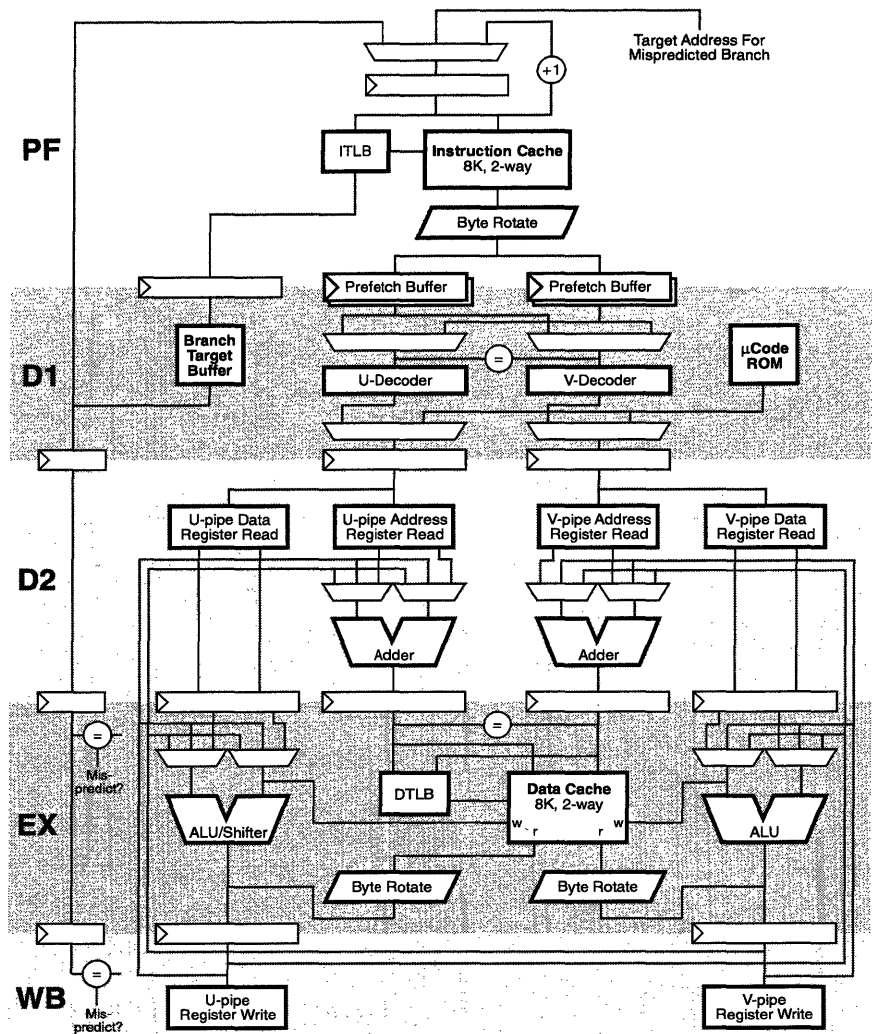


Figure 12-3. Pentium integer unit data-path pipeline stages.

instruction fetch efficiency because instruction and data accesses need not compete for a single cache resource.

Instructions fetched from the I-cache are stored in four prefetch buffers, each of which is the length of one cache line (32 bytes). The prefetch buffers are organized as two pairs, with each pair acting as a 64-byte circular queue. During sequential program execution an entire line of instructions is retrieved from the I-cache and written in parallel to a buffer in one of the circular queues. Instruction bytes can then be extracted from the buffer and passed to the instruction decode logic as needed.

Meanwhile, the prefetch unit reads the next sequential I-cache line into the buffer that serves as the other half of the “active” circular queue. By the time decode logic finishes processing the first 32 bytes read from the I-cache, the next 32 bytes will more than likely be waiting, and the decoder can begin extracting instruction bytes from that buffer as needed, while the first buffer is refilled with yet another sequential cache line. A single prefetch queue can thus generally stay far enough ahead of the instruction pointer that the decoder will rarely stall for lack of instructions, even if the processor must go off-chip to satisfy a given instruction request.

When a conditional branch is detected, the prefetch logic begins filling the alternate circular queue, starting with the instruction specified by the branch destination field. If control logic decides the branch should indeed be taken, the second queue will already have prefetched the destination instruction stream, and the process described above will repeat. If the branch is not taken, the original circular queue will still hold the instruction sequence following the branch, and execution may resume directly. Pentium thus needs seldom wait for instructions, except after cache misses and mispredicted branches.

**Decode 1 Stage.** The Decode 1 (D1) stage performs preliminary instruction alignment and decoding. Pentium uses hard-wired logic rather than microcode to decode many of the most common instructions and formats. Even seemingly complex memory-to-register and register-to-memory arithmetic operations do not require microcode assistance for their processing. Instead, a single internal microword is generated by the D1 decoding logic that triggers a hardware state machine in the EX stage. Thus, while memory/register operations do not require microcode, they do still require sequencing and multiple cycles.

For instructions that are complex enough to require a microcode routine, the first microword is generated by the D1 decoding logic. This microword proceeds to the D2 stage, where the microcode engine takes over the Pentium execution resources.

As shown in Figure 12-3, microwords from the microcode ROM control both integer pipelines; consequently, the pipelines operate independently only for pairs of instructions that use hard-wired control.

Microcode routines use the resources of both integer pipelines wherever possible. This reduces the number of cycles needed for many of the complex x86 instructions. For example, repeated



string-move instructions execute at three clock cycles per iteration on the 486. The Pentium microcode actually contains an unrolled loop that writes the element of the destination string in the U pipeline in parallel with the reading of the next source string element in the V pipeline, allowing string moves to execute at one cycle per iteration.

The Pentium microcode ROM contains about 4K microwords, each 92 bits long. Since microcoded routines take over all execution resources, it is not possible for Pentium to pair microinstructions with regular, x86 instructions. Thus, instruction fetching and dispatch typically stall during the execution of a complex, microcoded instruction.

Branch prediction, another major function of the D1 stage, is discussed in detail below.

**Decode 2 Stage.** The primary function of the Decode 2 (D2) stage is to read operands from the register file for use by the ALUs during simple register-to-register operations. The D2 stage also includes a dedicated Address Generation Unit (AGU) to perform the multiple component address computations commonly encountered in x86 programs.

The AGU within each integer pipeline contains a dedicated four-input address adder. Four inputs are needed because x86 operand addresses may include four components: a segment descriptor base, a base address from a general register, an index (possibly scaled) from a general register, and a displacement constant from the instruction. Address adders in the 486 have only three inputs, so instructions that require two D2 cycles on the 486 can complete in a single cycle on Pentium. (In Figure 12-3, the address adders are portrayed with only two inputs to reduce drawing complexity.)

Not shown in Figure 12-3 is the segment limit-check logic. Architecturally, x86 addressing requires that all segment accesses be checked against the limit stored in the segment descriptor. This check requires a separate four-component addition, so Pentium contains yet two more four-input, 32-bit adders to perform this check in parallel. The (single) 486 limit-check adder has only three inputs. While the need for this hardware probably does not affect the cycle time of the Pentium implementation, it certainly adds to die area and power. This is one way Pentium pays for the complexity of the x86 architecture.

**Execute Stage.** The Execute (EX) stage contains the integer ALUs and the data cache. The U-pipe has a full ALU and a barrel shifter, while the V-pipe has only an ALU. Thus, all shift instructions must be processed in the U-pipe, and the logic in the D1 stage that detects resource requirements takes care of enforcing this rule. Note that if the U-pipe contains any kind of branch, the V-pipe will be idle.

**Write-Back Stage.** During the Write-Back (WB) stage the data resulting from computations and load operations is written into the register file. This is shown conceptually in Figure 12-3 with separate boxes labeled “Register Write” in the WB stage. In actuality, the write-back stages of both pipelines update the same register file logic.

One level of sophistication not described in Intel’s technical documentation is the fact that Pentium does indeed implement two complete, separate register files. Each of these files contains an identical copy of each of the working register values. One of the register files feeds register-based variables to both Integer Execution Units. The second file feeds register data used in computing memory addresses directly to both Address Generation Units. When a register value is changed—for example, when an arithmetic instruction modifies a general register, or when a PUSH or POP instruction modifies the stack pointer—the new value is written simultaneously into each file.

Partitioning the register files in this way serves two purposes. Given that the IEUs and AGUs together need to read up to eight register values during a given clock cycle, it’s more efficient to design one file with four read ports and a second file with four more than to design a single file with all eight ports. Second, duplicating the register files allows each set of registers to be located physically closer to the logic it drives, with its read timing optimized as appropriate for the function it performs.

### **Instruction Issue Rules**

In order for Pentium to issue two successive integer instructions in a single clock cycle, they must satisfy certain constraints:

- Both instructions must be “simple,” or the first must be simple and the second be a jump or branch.
- Neither instruction may contain both a constant displacement field and an immediate data value.

- If the first instruction modifies a register, the second instruction may not read or modify the same register.

For the purposes of these rules, simple instructions are defined as any combination of the operations and operands shown in Table 12-1. Most of the “simple” instructions are hardwired and execute in a single clock cycle. Exceptions are noted in the right-most column of the table.

Operation	Destination	Source	Cycles
MOV	register	register	1
		memory	2
		immediate	1
MOV	memory	register	1
		immediate	1
ALU-Op (ADD, SUB, AND, OR, XOR, etc.)	register	register	1
		memory	2
		immediate	1
ALU-Op (ADD, SUB, AND, OR, XOR, etc.)	memory	register	3
		immediate	3
INC	register	—	1
	memory	—	1
DEC	register	—	1
	memory	—	1
LEA	register	memory	1
PUSH	—	register	1
		memory	2
POP	register	—	1
JUMP	near offset	—	1
CALL Jcc			1
NOP	—	—	1

Table 12-1. “Simple” Pentium instruction formats and operands.

In general, the U- and V-pipes can execute separate instructions simultaneously only if the instructions they contain are independent. Special-case exceptions are supported to allow the simultaneous dispatch any combination of stack PUSH and POP operations, a branch-offset-size override prefix followed by a branch or jump instruction, or a compare instruction followed immediately by a conditional-branch.

Register dependencies can prevent dual-instruction issue. If an ALU operation that modifies a particular working register is followed by an instruction that reads the modified value, the two may not be dispatched together. Two successive instructions

that modify the same register would likewise be dispatched serially, though the utility of such a sequence is highly questionable.

Note that, from this perspective, the condition code register often acts as an implicit shared resource: an ALU instruction that sets the carry flag, for example, cannot be paired with an ALU instruction that reads the same flag. ADDC, SUBB, and shift instructions can be executed only in the U pipeline, so they must be the first instruction in a pair.

When the first of two register/memory instructions modifies data memory, and the second instruction might read or modify the same physical location, a hazard exists: to be safe, the second instruction should not read or alter the memory word until the first has completed its modifications.

Cycle	EX-Stage Activity W + W		EX-Stage Activity W + R/M/W		EX-Stage Activity R/M/W+ W		EX-Stage Activity R/M/W+R/M/W	
	U Pipe	V Pipe	U Pipe	V Pipe	U Pipe	V Pipe	U Pipe	V Pipe
n	store	-idle-	store	load	load	-idle-	load	-idle-
n + 1	-idle-	store	-idle-	ALU	ALU	-idle-	ALU	-idle-
n + 2	-idle-	-idle-	-idle-	store	store	-idle-	store	load
n + 3	-idle-	-idle-	-idle-	-idle-	-idle-	store	-idle-	ALU
n + 4	-idle-	-idle-	-idle-	-idle-	-idle-	-idle-	-idle-	store

Table 12-2. Serialization of accesses to D-cache.

For example, consider the case of two successive store instructions, shown on the left in Table 12-2. The two instructions are issued simultaneously into the U and V pipelines, and proceed concurrently to the EX stage. Once there, however, Pentium forces serialized execution: V-pipe execution stalls until the U pipe is done.

The worst-case situation of two successive instructions that increment the same memory-based variable is shown on the right of in Table 12-2. V-pipe execution stalls at the Execute stage until the last cycle of U-pipe instruction. Note that a single cycle of overlap is okay; the V-pipe can read a new value properly during the same cycle it's written to the D-cache. In Table 12-2 the overlapping of the V-pipe load with the U-pipe store at cycle  $n+2$  saves one clock.

There are, however, three important exceptions which allow otherwise dependent instructions to be dispatched and execute

together. The first exception allows a compare instruction followed by a conditional-branch to be dispatched together because branch prediction will likely provide the branch target anyway. If branch prediction is correct, a cycle is saved by pairing the compare and the conditional-branch. Since most compare/conditional-branch pairs that occur during program execution will be in loops, and since most loops execute many times, branch prediction should perform very well for this situation.

The second exception allows two PUSH or POP instructions to be paired, despite the fact that the stack-pointer value used by the second instruction would seem to be dependent on the SP update performed by the first.

The third exception allows successive arithmetic instructions that both modify the condition code register but otherwise have no dependencies to be paired. Condition-code logic “magically” (in the words of a Pentium design manager) determines what the net result of any such instruction combination should be, and updates the flag register with the effective net result of the two instructions executed.

All things considered, Intel claims that between about 30% and 40% of all instructions execute in the second, V pipeline, as measured for recompiled software. All remaining instructions (i.e., between 68% and 60%, respectively) execute in the U-pipe. This implies that up to two-thirds of all instruction-dispatch cycles involve simultaneous issuing of two instructions. Figure 12-4 shows the dual-dispatch efficiency for a variety of SPEC integer benchmark programs.

(In Figure 12-4 and several similar graphs that follow, each of these programs has been recompiled to optimize its operation for the Pentium microarchitecture.)

## **Instruction Cache and TLB**

The Pentium instruction cache contains 8K bytes. The I-cache has a 32-byte line size and is two-way set associative. Two-way set associativity was selected for Pentium (versus the four-way design of the 486 cache) as a compromise between performance and implementation constraints. The Pentium I-cache implements an LRU (least-recently-used) replacement policy. According to Intel, the measured instruction-cache hit rate for programs in the SPECint89 applications suite is typically between 93% and 97%, as shown in Figure 12-5.

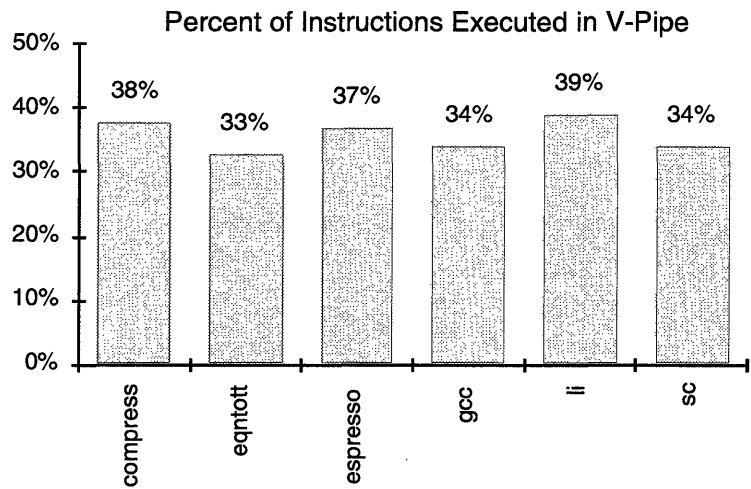


Figure 12-4. Pentium dual-instruction issue efficiency. (Source: Intel test results)

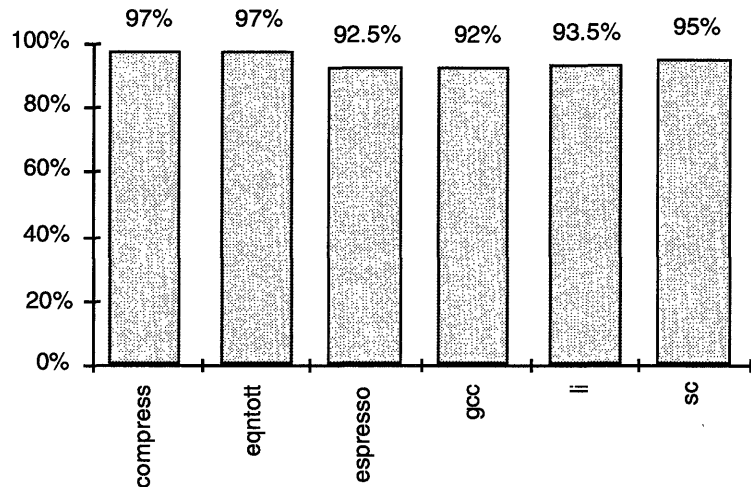


Figure 12-5. Pentium instruction cache hit rates. (Source: Intel test results)

Pentium further improves instruction fetch efficiency by implementing a “split fetch” capability not present in the 486, which ensures that Pentium can fetch at least 17 contiguous instruction bytes every cycle, even if the bytes are split across two instruction cache lines. I-cache operation is explained in detail below.

Full coherency is maintained between the I-cache and external memory via hardware snooping. The I-cache tags are fully

triple-ported, with one port associated with each half of a split I-cache line and a third port dedicated to I-cache snooping operations. Snooping can thus be performed without interfering or contending with instruction prefetch cycles. The cache arrays implement internal parity checking, with one parity bit per eight bytes of data and an additional bit for each tag.

As shown by the worst-case alignment scenario in Figure 12-6, split fetching allows a minimum of 17 bytes to be fetched from the cache because a fetch can straddle the boundary between two consecutive half-lines. According to Intel's measurements, the split-fetch capability improves Pentium performance by a few percent.

Split instruction fetching is a design technique often used in superscalar microprocessors in order to simultaneously issue and execute the maximum allowable number of instructions as often as possible. Indeed, the first microprocessor to implement split instruction fetching was Intel's i960CA superscalar embedded controller, introduced in 1989. Other superscalar processors also implement some form of split fetching—although different names are used—to make sure instruction-fetch bottlenecks do not limit performance.

The i960 and all other superscalar microprocessors introduced to date have RISC architectures. The word-alignment of RISC instructions results in less complex logic to eliminate alignment restrictions. The split-fetching logic, which must take care of byte-aligned x86 instructions, is one place where Pentium pays a price for the complex x86 architecture.

The Pentium instruction TLB has 32 entries, is four-way set-associative, and uses a pseudo-LRU replacement algorithm;

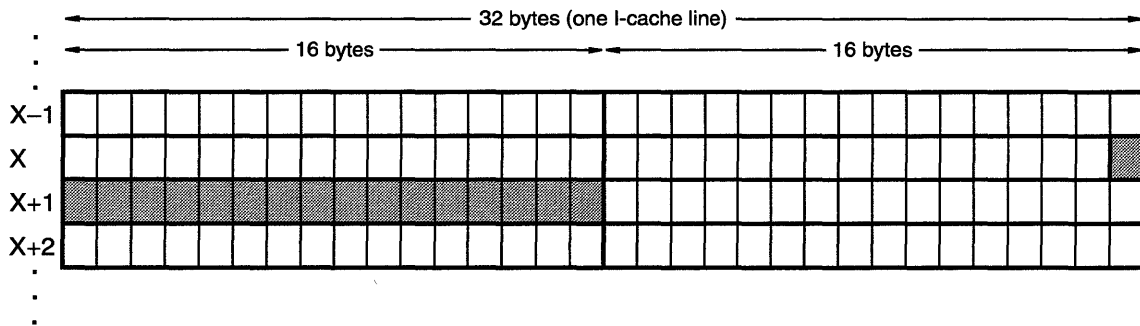


Figure 12-6. Pentium split-fetch instruction cache operation.

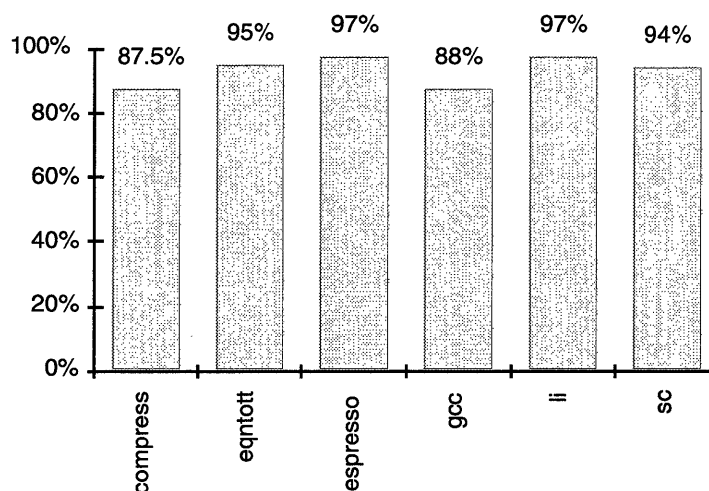


Figure 12-7. Pentium data cache hit rates. (Source: Intel test results)

ITLB misses are handled in hardware. The dedicated ITLB allows the I-cache to be physically tagged, which reduces the frequency of I-cache flushes. (The 486 indexes its cache with physical addresses as well.)

### Data Cache and TLBs

The data cache is one of Pentium's more innovative features. Like the instruction cache, it is a two-way set-associative, 8K-byte cache with a 32-byte line size. Intel says tests of the SPEC integer benchmark suite show the data-cache hit rate to range from about 88% to 97% for recompiled code, as shown in Figure 12-7.

Because the x86 architecture has a relatively small register set, as well as instructions that combine memory references with computations, the number of data memory references per instruction is considerably higher than for RISCs. Intel estimates that optimized, 32-bit x86 code has an average of 0.6 data references per instruction, while standard RISCs average about 0.3 data references per instruction. Because data memory accesses occur so frequently, D-cache efficiency is critical.

The Pentium D-cache was designed to allow two data references to occur simultaneously. The data array itself is single-ported, but each 32-byte cache line is divided into eight four-byte groups. Each group, or bank, has its own address decoders and data buffers. As a result, any two D-cache accesses that involve separate banks (i.e., that differ in address bits A4..A2) can be



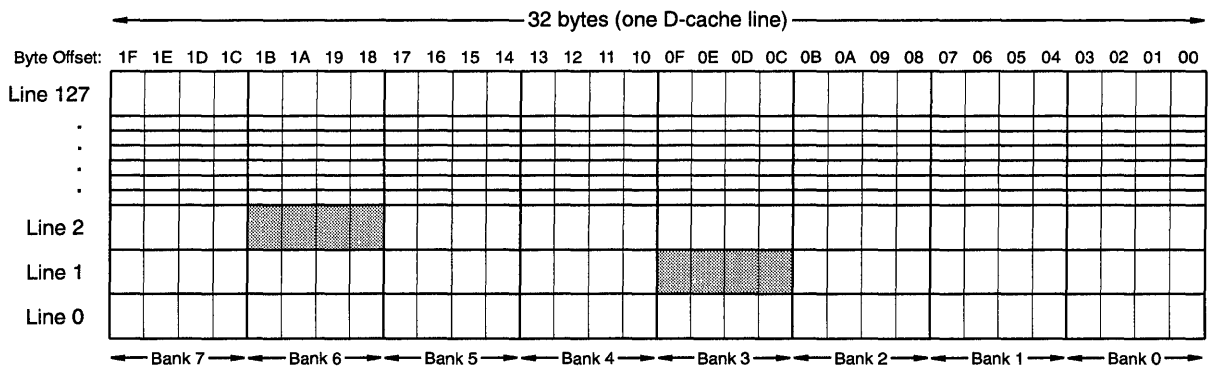


Figure 12-8. Pentium data cache interleaved bank partitioning.

performed during the same clock cycle without conflict. Figure 12-8 illustrates the bank partitioning scheme.

The dual-access capability, which lets both pipelines access the data cache simultaneously, is implemented by interleaving the data array into eight banks (four-byte granularity within a 32-byte cache line). As long as the data accesses from each pipe are to separate banks, both accesses can be processed simultaneously by the cache in a single cycle. Memory values stored in (shaded) cache locations 002CH and 0058H in Figure 12-8 may be read simultaneously, for example, since the first resides in Bank 3 and the second in Bank 6. Pentium is the first microprocessor of any architectural philosophy to provide this capability.

Figure 12-9 shows the conflict-detection circuitry that makes dual cache accesses possible. If a bank conflict is detected, the U-pipe access is allowed to proceed first, and the V-pipe access is stalled for one cycle.

The data-cache TLB (translation lookaside buffer) is fully dual-ported to allow simultaneous translation of memory accesses performed by the U and V pipelines. The data-cache tags are fully triple-ported in order to allow snoop cycles to occur without stalling cache accesses from either the U- or V-pipe.

The data arrays are not fully dual-ported because doing so would have nearly doubled their physical area. The single-ported, interleaved cache structure is considerably denser, which allowed the cache capacity to be increased. Intel's designers believed the higher hit rate resulting from the higher-capacity, single-ported cache would more than compensate for the loss in efficiency due to stalls resulting from bank conflicts.

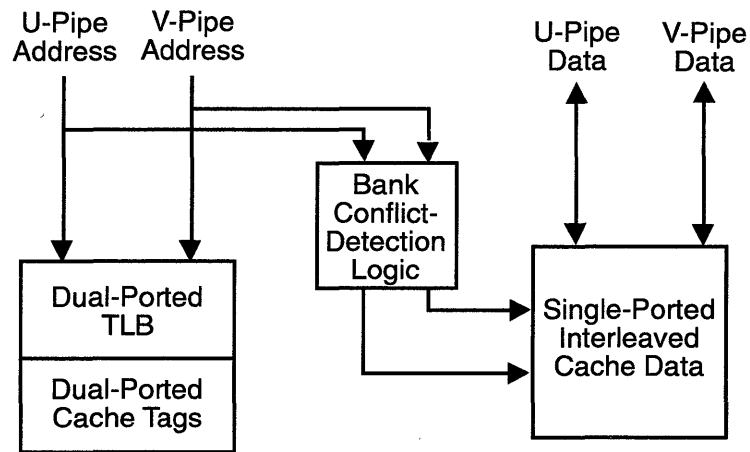


Figure 12-9. Pentium interleaved data cache operation.

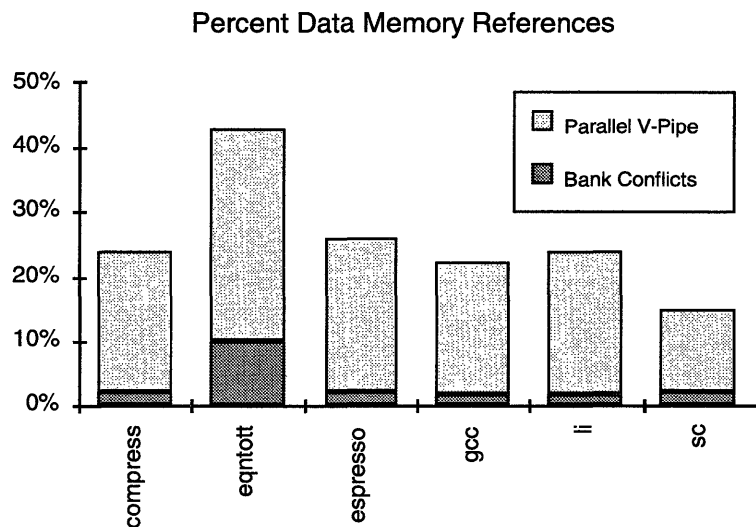


Figure 12-10. Pentium dual-access D-cache efficiency. (Source: Intel test results)

As shown in Figure 12-10, up to 44% of all data references involve simultaneous accesses by the U and V pipelines. Conflicts for the same cache interleave block typically occur during between 2% and 10% of all memory-access cycles.

### D-Cache Snooping

To maintain cache coherency in both single- and multiple-processor systems, Pentium implements a four-state MESI (Modified/Exclusive/Shared/Invalid) cache consistency protocol with both internal and external cache snooping. Internal snooping occurs under three conditions.

First, an internal snoop is conducted if a miss is detected in the instruction cache. If the snoop hits in the data cache and the accessed line is in either the Shared or Exclusive state, the line is simply invalidated. If the accessed data cache line is in the Modified state, the line is first written back to external RAM or cache and then invalidated in the data cache. In all cases, the original instruction-cache miss is satisfied by a cache line fill from external RAM or second-level cache.

Second, an internal snoop to the instruction cache occurs for internal data cache misses. If the snoop hits in the instruction cache, the line in the instruction cache is invalidated. These first two cases handle self-modifying code.

Third, an internal snoop to both caches occurs if there is a write to the “accessed” and/or “dirty” bits in the TLB entries. If the snoop hits in either or both caches, the accessed lines are invalidated. If the accessed line in the data cache is in the M state, it is written back first. This is done because the in-cache copies are stale after the change is made by the MMU to both the TLB entries and the page-table entries in memory.

Since the cache stores physical tags, the data TLB must be able to perform two address translations simultaneously. This capability is provided by a dual-ported, 64-entry, four-way set-associative DTLB.

The DTLB stores translations for the standard 4K pages of the 386 architecture. There is a separate eight-entry, four-way set-associative DTLB, also dual-ported, for 4M pages. Large-page mapping has become commonplace on high-end processors and is useful because mapping operating system segments and graphics frame buffers can be done with only one 4M translation entry instead of many 4K entries. This keeps OS and frame-buffer references from “polluting” the main TLB.

### **Branch Prediction Logic**

Pentium uses a BTB (branch target buffer) to perform branch prediction. In principle, whenever a branch is taken, the address of the branch instruction itself and the address of its destination are copied into the buffer. If the instruction initiating the branch is executed again later, the BTB logic recognizes its address and immediately begins prefetching a new instruction stream, beginning with the target address to which the branch was last taken. Prefetch logic thus gets a head start on execution, without having to wait for the branch to wind its way through the pipeline.

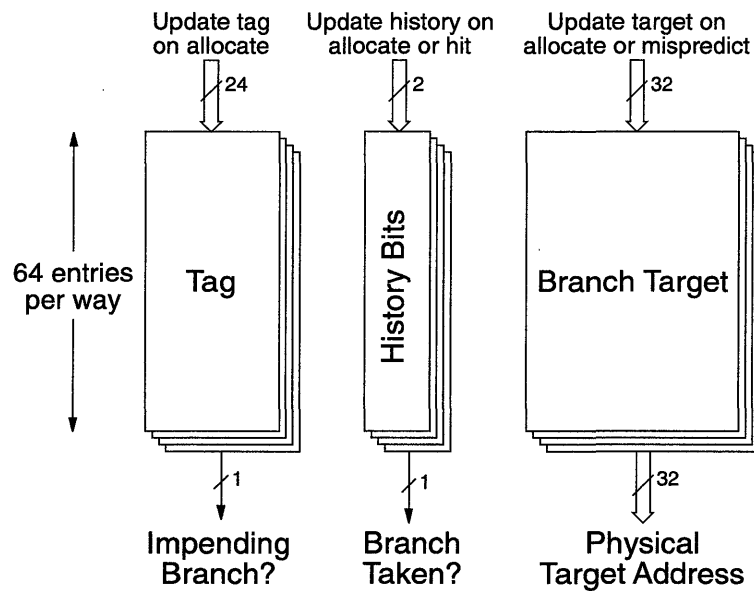


Figure 12-11. Pentium branch target buffer organization.

The preceding overview matches the BTB description contained in Intel documentation, but the low-level pipeline timing prevents this scheme from working as Intel says. In actuality, the address stored and recognized by the BTB logic is not of the instruction that contains the branch, but of the instruction executed immediately beforehand.

As shown in Figure 12-3, the BTB is accessed in stage D1 with the 32-bit linear address (or virtual address, if memory management is enabled) of the instruction executed before the branch. As the branch instruction enters the D1 stage, the BTB logic returns the branch target address. As the branch enters the D2 stage, the destination instruction returned by the prefetch unit enters the D1 stage. Correctly predicted branches can thus complete in effectively one clock cycle.

The BTB stores a single predicted target for a branch. As Figure 12-11 illustrates, the BTB cache stores 256 branch predictions with a four-way set-associative organization. Note that this is different from the branch target cache in the AMD 29000 embedded RISC processor, which stores the first few instructions themselves at the branch destination. Pentium's BTB stores target addresses only, not the contents of the instructions so addressed.

Intel simulated several branch-prediction algorithms during the Pentium design process, finally settling on a method described by J. Lee and A.J. Smith in a paper from the UC Berkeley (see reference 51 at the end of this chapter). This algorithm uses two bits to hold the prediction state, with transitions between the four states occurring as necessary when a branch is encountered.

Figure 12-12 shows the state-transition diagram. The four states are ST (strongly taken), WT (weakly taken), WNT (weakly not taken), and SNT (strongly not taken). Each time there is a hit in the BTB (though not necessarily a correct prediction), the state bits are updated. When the state bits are either ST or WT, the next prediction for the given branch will be “taken.” WNT and SNT mean the next prediction will be “not taken.”

The two middle states provide a degree of misprediction hysteresis to avoid thrashing in certain cases. The hysteresis is provided by the fact that it takes two consecutive incorrect predictions to change the prediction polarity. For example, a branch that has been taken many times in a row will continue to be predicted as taken, even if on rare occasions the branch is indeed not taken.

The BTB allocation policy is that an unbuffered branch allocates an entry in the buffer only if it is a taken branch (i.e., no allocate on miss). As a result, the state bits are always initialized to ST for a newly allocated branch. Branches that cause a miss in the BTB are initially assumed (predicted) to be not-taken.

As an example of the prediction state transition operation, if this newly allocated branch is not taken the next time it is encountered, its state bits will make a transition to WT. The next prediction will thus be “taken,” but if this is also a mispre-

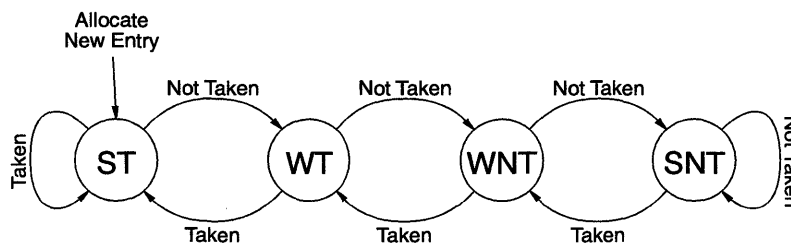


Figure 12-12. Pentium branch history bit state transitions.

diction, the prediction state will make the transition to WNT. The next prediction will be “not taken,” and so on.

(Note: Figure 12-12 portrays the branch-history state transition diagram as it has appeared in Intel presentations and documentation. In fact, there is a subtle nuance of the control logic not reflected by the figure: the two bits that encode the prediction state variable are also used to distinguish valid from invalid entries in the BTB cache. The state designated SNT does double duty as the “invalid” state; when it becomes necessary to allocate a new entry in the BTB cache, existing entries in the SNT state are considered as candidates for reassignment.)

Down the left side of Figure 12-3 is a very simplified representation of the pipeline used to verify branch prediction. The predicted destination of the branch is carried along with the branch instruction as it moves through the pipeline. As soon as possible, the prediction and the actual direction taken are compared. For unconditional branches in the V pipeline and all branches in the U pipeline, a comparator in the EX stage (represented by the circle containing an equal sign) does the check. For conditionals in V, the check is made by the comparator in WB to allow resolution of a possible paired “compare” in the U-pipe.

When an incorrect prediction is discovered or when the predicted target is wrong, the pipelines are flushed and the correct target fetched. Thus, based on the stage in which the misprediction is discovered, mispredicted unconditionals and U pipeline conditionals incur a three-clock delay, while V pipeline conditional branches incur a four-clock delay.

According to Intel’s measurements of Pentium branch behavior on the SPEC89 integer application suite, the percentage of dynamic branches correctly predicted is about 70% and 85%, including not-taken branches that miss (see Figure 12-14). The branch distribution between pipelines appears to be balanced at about 50% for each pipeline on code produced by both 486-optimized and Pentium-optimized compilers.

### **Floating-Point Unit**

In the past, the floating-point performance of x86 microprocessors has been poor. Even with the 486, the SPECfp92 rating is less than half the SPECint92 rating. This is not primarily a result of the x86 architecture, but rather of Intel’s priorities: making floating-point go fast takes lots of transistors, and in traditional PC markets it isn’t that important. Thus, Intel did

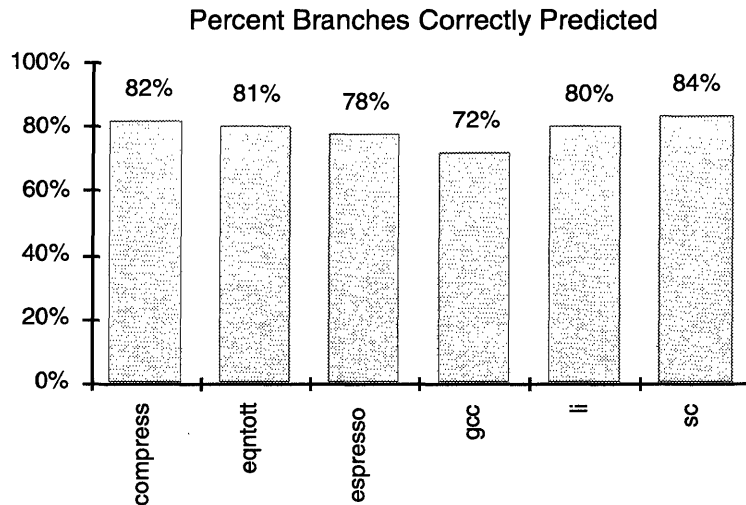


Figure 12-13. Branch-prediction logic accuracy. (Source: Intel test results)

not devote much design effort or transistor budget to the floating-point unit in the 486.

With Pentium, however, the equation has changed. While the floating-point needs of the typical PC user haven't increased much, it has become strategically important for Intel to match the performance of RISC microprocessors, whose biggest performance lead is in floating point. PC applications are becoming more floating-point-intensive with increased use of 3-D graphics, and Intel also hopes to push Pentium into technical workstation markets where fast floating point is essential.

Pentium's floating-point unit is fully compatible with that of the 486, but its performance has been greatly enhanced. The eight-stage floating-point pipeline is integrated with the integer pipelines, and the first four stages are the same. Both the U-pipe and the V-pipe are used to fetch operands, allowing both data-cache access paths to be used in parallel to load a 64-bit floating-point value in a single clock cycle. Floating-point execution is performed in the U-pipe.

**FPU Pipeline Design.** Pentium's floating-point performance is vastly improved over the 486 because the simple, serial floating-point unit of the 486 is replaced with fully pipelined, parallel execution units. The FPU pipeline is eight stages, where the first four are shared with the integer pipeline:

- PF (prefetch)

- D1 (instruction decode)
- D2 (address generation)
- EX (memory and register read, memory write if FP store instruction)
- X1 (FP execute first stage, write operand to FP register file if FP load)
- X2 (FP execute second stage)
- WF (rounding and write result to FP register file)
- ER (error reporting, update status word)

This pipeline structure is similar to that of other high-performance processors. The integer execute stage is used to fetch operands, and it is followed by three floating-point execution stages. The final stage of the floating-point pipeline is used for error reporting; results of calculations are available at the start of this stage, so it does not affect latency.

Like most high-end RISCs, Pentium's FPU is fully pipelined for add/subtract and multiply operations; it can start a new operation on every clock cycle for double-precision, memory-to-register operations (assuming a cache hit, of course) and also for extended-precision (80-bit), register-to-register operations.

The floating-point adder and multiplier provide single-cycle throughput and three-cycle latency for all precisions (single, double, and extended). The divider processes two bits of quotient in each cycle. For a double-precision value with a 52-bit fraction, this implies a divide time of 26 cycles plus setup and normalization time.

Pentium's floating-point unit is the first high-performance design to implement transcendental functions. These functions aren't included in RISC instruction sets; Motorola decided to trap these operations in the 68040 and implement them using trap handlers. Pentium abandons the CORDIC algorithms used by the 486's FPU and earlier x87 coprocessors, and instead uses table-driven algorithms with polynomial approximation.

As with most other high-performance processors, Pentium allows concurrency between the floating-point and integer units. Thus, the issue and execution of integer instructions can



proceed in parallel with the execution of a long-latency floating-point operation.

**FPU Performance.** As shown in Table 12-3, Pentium has floating-point operation latency and throughput comparable to other processors for basic arithmetic operations.

Processor	FP Add	FP Subtract	FP Multiply	FP Divide
Pentium	3/1	3/1	3/2	39/39
486	8-20/8-20	8-20/8-20	16/16	73/73
R4000	4/3	4/3	8/4	36/36
Alpha	4/1	4/1	4/1	61/61
PowerPC 601	4/1	4/1	4/2	31/29

Table 12-3. Pentium FPU instruction latency and throughput.

From Table 12-3, it is tempting to conclude that Pentium could approximately match the floating-point performance of many RISC processors. Pentium is hampered, however, by its stack-oriented floating-point register file architecture and by the need to transfer floating-point condition codes to the integer unit before a conditional branch can be executed.

For floating-point operands, Pentium maintains backward compatibility with previous x86 FPUs: there is a file of eight, 80-bit operand registers that are conceptually a stack and only marginally directly addressable. Since most floating-point instructions implicitly use the top of this register stack as one operand, there is a “top-of-stack bottleneck.” To circumvent this, programs use the FXCH (floating-point register exchange) instruction to swap the top of stack with an operand deeper in the register file.

In general, floating-point instructions cannot be issued simultaneously with each other or in conjunction with integer instructions. There is one exception, however. The FXCH instruction, can be paired with a “simple” floating-point instruction. Simple floating-point instructions in this context include:

- FLD single/double
- FLD ST(i)
- All forms of FADD, FSUB, FMUL, FDIV
- All forms of FCOM, FUCOM, FTST, FABS, and FCHS

The FXCH must be the second instruction in the pair. If an integer instruction immediately follows the FXCH, it will stall for one or four clocks depending on the operands to the pair of floating-point instructions.

This optimization is important because the top-of-stack serves as the floating-point accumulator, creating a bottleneck not found on register-file-oriented floating-point processors. The parallel execution of the exchange instruction partially ameliorates this bottleneck. The exchange is effectively performed after the computation completes, so it has the effect of directing the result to any register in the stack. At the same time, it brings a value up from that register into the top-of-stack, where it can be used by the next instruction.

Even with the rapid execution of an FP-operation/FXCH pair, Pentium will be hampered by the small, eight-register file. In addition, an FP-operation/FXCH pair followed immediately by an integer instruction will incur a one-cycle penalty.

Another performance problem for Pentium is presented by branching on floating-point conditions. Most microprocessor architectures allow the results of a floating-point comparison to be tested directly, but the x86 architecture requires that the floating-point condition codes be transferred to the integer condition-code register, where a normal integer conditional branch can test them.

To effect a floating-point conditional branch requires four instructions:

1. An FP operation that sets the condition codes
2. FSTSW AX (move FP status word to AX register)
3. SAHF (transfer to upper half of EFLAGS)
4. Jcc (integer jump conditional)

This sequence takes nine clock cycles to execute on Pentium because the floating-point condition codes are updated late in the floating-point pipeline. Four of these clocks can be recovered by inserting integer instructions between the first and second floating-point instructions listed here.

Although many floating-point loops iterate based on an integer condition, such as a loop count equal to the number of elements

in an array, the need to transfer condition codes from the FPU to the integer unit creates a significant penalty for the case of loops with a floating-point termination condition, and for if-then statements with floating-point conditions.

**FPU Exception Model.** Some architectures, such as DEC's Alpha, sacrifice precise exceptions to improve floating-point performance. This means that one or more instructions beyond the instruction causing the exception may be executed before the exception is recognized. Intel did not have this option if full compatibility with existing programs was to be maintained, but having to wait until a floating-point instruction was complete before launching the next instruction would have caused a significant loss in performance.

Pentium tackles this problem by adding hardware that examines the input operands for each floating-point operation that could generate an exception to determine if the calculation is "safe," that is, if it can be guaranteed not to generate an exception. For example, the addition or subtraction of any two double-precision (64-bit) values is guaranteed never to cause an overflow because all data is stored in the Pentium register stack in the 80-bit extended-precision format, providing additional bits in the exponent.

If an operation can be determined in advance to be "safe," the exception-processing pipeline stages are short circuited, and ensuing instructions may begin immediately. Only if an operation has the potential to generate an exception is the next instruction delayed until the first operation completes. Unsafe operand combinations are very rare; according to Intel, none were detected in the entire SPECfp89 suite.

### **Architecture Extensions**

While Pentium incorporates several architectural changes from the 486, only a few are significant. It makes little sense for Intel to change the instruction set of the most successful general-purpose microprocessor architecture in existence.

The last three of these instructions, as well as a number of other extensions to the Pentium architecture, are partially or wholly described in Appendix H of the *Pentium Processor User's Manual: Volume 3*; this volume, by itself, is over 1,000 pages long. Unfortunately, Appendix H contains only a three-sentence explanation that the information is considered Intel confidential and proprietary and is provided in the *Supplement to the Pentium Processor User's Manual*, available only under appropriate nondisclosure.

Intel says it is willing to provide the supplement to operating-system vendors, compiler writers, BIOS developers, ISVs, major customers, and others with (in Intel's eyes?) "a need to know." This policy lets Intel keep Pentium-specific details secret from its competitors. It remains to be seen whether future OSs will come in two versions—one for the 486, one for Pentium—or whether a single version that checks processor type will be delivered.

The Pentium programming architecture has been extended to include a handful of new control registers, new control and status bits in existing registers, and eight new instructions. Three instructions have been added to the user-mode instruction set; five more may be used by system-mode software only. Table 12-4 describes the operation of these instructions.

Mnemonic	Mode	Description
CMPXCH8B	User/ System	Compare and exchange eight bytes
CPUID	User/ System	Load CPU identification code
RDTSC	User/ System	Read TSC register (Details contained in Intel "Appendix H")
RSM	System	Return from SMM interrupt
MOV CR4,r32	System	Write to Control Register 4
MOV r32,CR4	System	Read from Control Register 4
RDMSR	System	Read model-specific register (Details contained in Intel "Appendix H")
WRMSR	System	Write model-specific register (Details contained in Intel "Appendix H")

Table 12-4. Pentium-specific x86 instruction set extensions.

CMPXCH8B is an eight-byte version of the compare-and-exchange instruction that was introduced on the 486. When used with the LOCK prefix, this instruction acts as a mutual-exclusion primitive in multiprocessor algorithms.

CPUID is a new instruction that allows a program to directly learn certain key manufacturing parameters about a particular chip. (This instruction has also been retrofitted to Intel's "SL-enhanced" 486 devices.) This instruction returns different values, depending on the value contained in the 32-bit EAX register.

If EAX is initially set to zero, the instruction returns the string "GenuineIntel" as three, four-character ASCII strings in EBX,

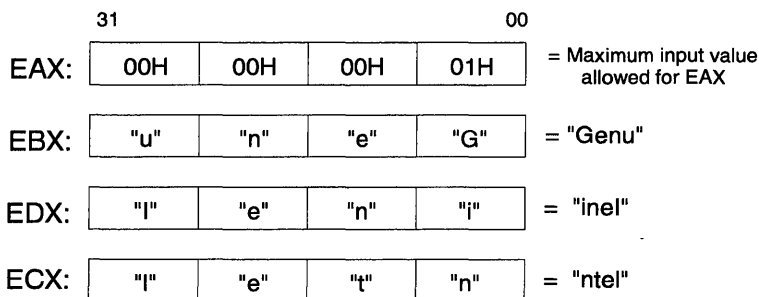


Figure 12-14. CPU registers after invoking CPUID with EAX = 0.

EDX, and ECX, as shown in Figure 12-14. (Note the apparent influence of Intel’s marketing and legal departments in architectural design!) The EAX register holds the value 00000001H upon completion of the instruction, which indicates the maximum initial EAX value allowed when CPUID is invoked.

If EAX is set to 00000001H before invoking CPUID, the instruction returns code values in EAX and EDX identifying the vendor, family, model, stepping, and feature flags of the microprocessor on which it is executing. Three of the feature flags tell whether there is an on-chip FPU, whether the machine-check exception is implemented, and whether the CMPXCHG8B instruction is implemented. Six additional bits are described only in the mysterious Appendix H. This situation is shown in Figure 12-15.

Operation of the CPUID instruction is not defined for initial EAX index values other than 0 and 1, but further Intel chips may define behavior for higher values.

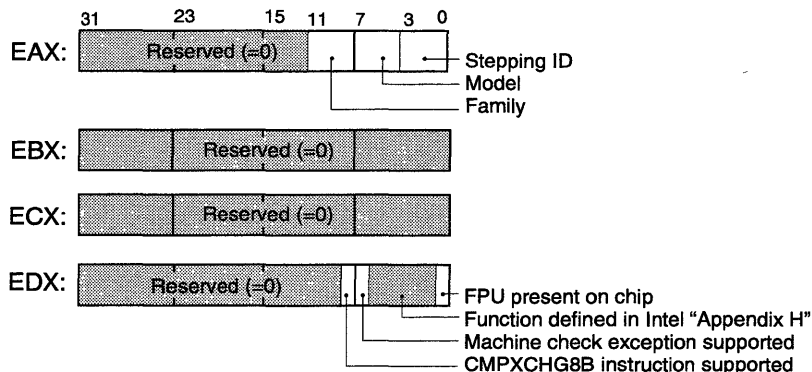


Figure 12-15. CPU registers after invoking CPUID with EAX = 1.

The third new user-mode instruction is RDTSC, and provides support for Pentium's new performance-monitoring timers. Unfortunately, full details are contained only in Appendix H.

The five new system-mode instructions listed in Table 12-4 support new Pentium features and may be executed only in privileged execution mode. The RSM instruction is used to return from system management mode (discussed below) to the interrupted processor operating mode.

The two new forms of the MOV instruction copy data into or out of Pentium's control register number 4, which is not implemented in the 486. This control register implements six bits: MCE (enable machine-check exceptions), PSE (documented in Appendix H), DE (enable debugging extensions), TSD (documented in Appendix H), PVI (documented in Appendix H), and VME (documented in Appendix H). The machine-check exception is used to report parity errors, so trapping on parity errors can be turned off by disabling this exception. (Parity checking on the bus is always enabled; this is covered in greater detail below.)

The RDMSR and WRMSR instructions read and write various model-specific registers, respectively. The forms of the MOV instruction that were used in the 486 to access the test registers have been removed in Pentium. A new set of test registers has been defined for the caches, TLBs, and the BTB, and these "model-specific" registers—documented, naturally, only in Appendix H—are accessed with RDMSR and WRMSR.

The 32-bit EFLAGS register has three new Pentium-specific bits. The ID bit allows a program to determine if the processor on which it is running supports the CPUID instruction. This bit did not exist on earlier devices, and its state was undefined and unchangeable. If, however, the ID bit is implemented, and can be set and cleared under program control, then the CPUID instruction is supported. The VIP (virtual interrupt pending) and VIF (virtual interrupt flag) bits support changes to the way virtual-86 mode is implemented on Pentium. Unfortunately, full details are contained only in Appendix H.

Pentium implements three new extensions to the exception model. Exception #13, the general protection fault, is triggered by trying to write a 1 into any reserved bit position in a special control register. Exception #14, the page-fault exception, is triggered on Pentium in the case of a page fault or when a 1 is detected in any reserved bit position in a page table entry, a

page directory entry, or the page directory pointer during address translation. Exception #18, the machine check exception, is used to report internal parity errors and other hardware faults.

Pentium extends the virtual address translation model of the 386 and 486. In earlier devices, the virtual address translation mechanism only supported memory pages that were 4K bytes in size. The Pentium translation hardware supports 4M-byte pages as well. Documentation for 4M-byte-page table entries is contained in Appendix H, but most likely it allows page-directory entries, which normally indicate tables of 1024 4K-byte page table entries, to be used alone to describe a single 4M-byte page directly.

Pentium implements some additional extensions to the virtual-86 processor mode, which allows programs written for the 8086 to run in a virtual machine environment as a separate, protected task. The extensions, such as the VIP and VIF bits in the EFLAGS register, are documented only in Appendix H. These extensions are rumored to dramatically speed interrupt handling in virtual-86 mode.

## Software Optimization

As with most superscalar processors, extracting the full performance of which the hardware is capable requires a compiler that properly optimizes for the processor's pipeline structure. The usual techniques of instruction scheduling, register allocation, and loop unrolling all apply to Pentium. Good register allocation is especially important, since the register set is relatively small. In addition, there are some considerations that differ from those of RISC processors. For example, the compiler should select "simple" opcodes whenever possible, since only these instructions can be dual-issued. For floating-point code, different code-generation strategies are required to take advantage of the ability to parallel-issue the exchange instruction with computation instructions.

In the PC world, where there is a massive installed base of existing applications, the ability to perform well on old binaries is important. It remains to be seen how much of Pentium's potential performance boost will be realized on old binaries. The most performance-critical programs, however, are likely to be the first to be compiled, and Pentium-specific optimizations should not hurt performance on earlier processors. Intel's own compiler group worked with outside compiler vendors to assure that Pentium-specific optimizations would be supported by compilers announced simultaneously with processor introduction.

## 12.1 The Intel 0.8 $\mu$ Pentium “P5”

The original Pentium design (developed under the code name “P5,” and hereafter designated the “0.8 $\mu$  Pentium” for clarity) is fabricated using a 5-V 0.8-micron three-layer-metal BiCMOS process that combines bipolar and CMOS technologies for improved speed. Table 12-5 summarizes the key features of this design.

Product Name	Intel 0.8 $\mu$ Pentium “P5”
Introduction Date	March 1993
Device Integration Level	Superscalar 32-bit integer execution unit PMMU with optional expanded page size 8K-bytes each instruction and data cache High-speed floating-point unit Branch-prediction cache
CPU Architecture Level	Extended 486 IU and FPU instruction set
Core Technology	Superscalar dual 486-like pipelines
Pinout Standard	De facto “standard” Pentium pinout
Data Bus Width	64 data bits plus eight parity bits
Physical Addressability	4 gigabytes (Address pins A31..A3 plus BE7#..BE0#)
Data-Transfer Modes	Four $\times$ eight-byte burst-mode transfers
Cache Support	I-cache: 8K-byte split-line 2-way associative D-cache: 8K-byte 2-way associative with 8-way interleaved access and write-through or copy-back operation
Floating-Point Support	On-chip pipelined high-speed FPU
Operating Voltage	4.75 V to 5.25 V
Frequency Options	60- or 66-MHz core frequency
Clocking Regime	Core frequency = 1 $\times$ Clkin
Maximum Power Dissipation	16 W @ 5.0 V and 66 MHz (worst case)
Power-Control Features	Intel System Management Mode support
Process Technology	0.8 $\mu$ BiCMOS, three-layer-metal
Transistor Count	3.1 million transistors
Die Size	16.7 $\times$ 17.6 mm
Package Options	273-pin ceramic PGA package
Other Features	Functional redundancy support JTAG boundary-scan logic On-chip parity and integrity-checking logic

Table 12-5. Intel 0.8-micron Pentium “P5” feature summary.

The 0.8 $\mu$  Pentium design uses about 3.1 million transistors on a huge 294 mm<sup>2</sup> (456k mils<sup>2</sup>) die. At 17 mm on a side, this is one of the largest microprocessors ever fabricated, and probably pushes Intel’s production equipment to its limits.



**System Interface** Pentium implements a sophisticated, high-speed bus that builds on the protocols of the 486. With pipelined back-to-back cache-line fills, a 66-MHz Pentium can achieve a 528 Mbytes/s burst transfer rate—more than three times the 160 Mbytes/s of the 50-MHz 486 and five times the rate of the 66-MHz i486DX2.

While the Pentium bus is conceptually like a 64-bit version of the 486 bus, there are a few major changes and many subtle ones. As with some of the major changes to the x86 architecture, some of the enhancements to the Pentium bus structures are documented in yet another unavailable appendix (Appendix A to the *Pentium Processor User's Manual: Volume 1*).

The standard Pentium package has a total of 273 pins, with 173 signal pins and 100 power and ground pins. Tables 12-6 through 12-10 list each signal pin and its direction, and provide a brief description.

Signal	Direction	Function
A31..A3	I/O	Address bus
BE7#..BE0#	Out	Byte enable controls
D63..D0	I/O	Data bus
DP7..DP0	I/O	Data bus byte parity bits (even)
PEN#	In	Data bus parity check enable
PCHK#	Out	Data bus parity error detected
A20M#	In	Address-bit 20 Mask
AP	I/O	Address bus parity bit (even)
APCHK#	Out	Address bus parity error detected

Table 12-6. Pentium address and data bus signals.

**Data and Address Buses.** The most obvious change from the 486 is that Pentium's data bus is 64 bits wide. This allows the larger, 32-byte cache lines (vs 16 for the 486) to be filled using the same number (four) of transfer cycles. There are also eight parity bits (vs four) that are active for both input and output of data. Pentium uses even parity. Each byte has a separate byte-enable pin, and parity is checked or driven only for the bytes that are enabled.

Data parity checking is always enabled on input, and Pentium always generates parity for enabled bytes on output. The PCHK# output is asserted if a parity error is detected on input, which allows hardware to log parity errors or signal an interrupt. Pentium can also be configured to automatically cause an internal exception on parity errors. This exception can be blocked

either by disabling the machine-check exception (via the MCE bit in CR4) or by deasserting PEN on a cycle-by-cycle basis. Thus, it is possible for Pentium to automatically take action on parity errors, to have external hardware decide when to interrupt Pentium, or both.

Signal	Direction	Function
ADS#	Out	Address strobe (start of new bus cycle)
M/IO#	Out	Memory vs I/O bus cycle
D/C#	Out	Data vs Code bus cycle
W/R#	Out	Write vs Read bus cycle
CACHE#	Out	Read cycles: data returned may be cached Write cycles: burst-mode cache-line write-back
LOCK#	Out	Locked (indivisible) bus cycle
SCYC#	Out	Split cycles for locked-transfer transaction
NA#	In	Next address (allows external address pipelining)
BRDY#	In	Burst-mode transfer ready
BUSCHK#	In	Bus check (bus cycle completed unsuccessfully)
BOFF#	In	Back off (abort all outstanding bus cycles)
HOLD	In	Bus hold request (external master request)
HLDA	Out	Bus hold acknowledge (bus available)
BREQ	Out	Bus request (internal bus cycle pending)
SMI#	In	System management mode interrupt request
SMIACK#	Out	System management mode active

Table 12-7. System bus cycle control and status signals.

Signal	Direction	Function
PCD	Out	Page cache disable bit for requested data
PWT	Out	Page write-through bit for requested data
KEN#	In	Cacheability enabled for requested data
AHOLD	In	Address hold (float address bus next cycle)
EADS#	In	External snoop address driven to bus
INV	In	Invalidate cache line if inquire cycle hits in cache
HIT#	Out	Hit detected (result of inquire cycle)
HITM#	Out	Hit detected in modified line (result of inquire)
FLUSH#	In	Write-back cache data and flush cache
EWBE#	In	External write-buffer empty

Table 12-8. Cache control and status signals.

The address bus consists of 29 address lines and the eight byte-enables just mentioned. Parity is checked on the address lines, but only A31 through A5 participate; A4 and A3 are not checked. This is apparently due to the fact that only A31 through A5 are

used for cache-snooping operations (described later). Address-bus parity errors are signaled by the APCHK# signal. Since it is not possible to cause an internal exception as a result of an address parity error, external hardware must be used to either deal with the problem or cause an interrupt.

Signal	Direction	Function
CLK	In	Processor clock input (CPU freq = 1 x CLK)
RESET	In	Processor reset
INIT	In	Initialize (distinguishes cold from warm-start reset)
NMI	In	Non-maskable interrupt
INTR	In	Maskable interrupt request
FERR#	Out	Floating-point error detected
IGNNE#	In	Ignore floating-point errors
FRCMC#	In	Functional redundancy check master/checker
IERR#	Out	Internal parity or FRC error detected
TRST#	In	JTAG boundary-scan logic reset
TCK	In	JTAG boundary-scan logic Clock
TMS	In	JTAG boundary-scan mode select
TDI	In	JTAG data in
TDO	Out	JTAG data out
R/S#	In	Intel debug port Run/Stop control
PRDY	Out	Intel debug port Stop acknowledge

Table 12-9. Miscellaneous device control and status signals.

Signal	Direction	Function
BT3..BT0	Out	Branch trace (three LSBs of target; special cycle)
IU, IV	Out, Out	U-pipe, V-pipe instruction completed
IBT	Out	Instruction branch taken
BP3, BP2	Out, Out	Breakpoint 3, 2 condition detected
PM1/BP1	Out	Breakpoint/performance Monitoring pin 1
PM0/BP0	Out	Breakpoint/performance Monitoring pin 0

Table 12-10. Performance monitoring and tracing signals.

**Bus Cycle Types.** A Pentium bus cycle begins by asserting ADS# while driving valid address and transfer control signals onto the corresponding buses. Each bus cycle may consist of one or four transfers. A cycle ends when the last BRDY# is returned.

On the 486, the difference between a simple bus cycle and a burst cycle is determined by the acknowledgment: RDY# for simple or BRDY# for burst, with RDY# taking precedence. On Pentium, the difference between simple and burst cycles is

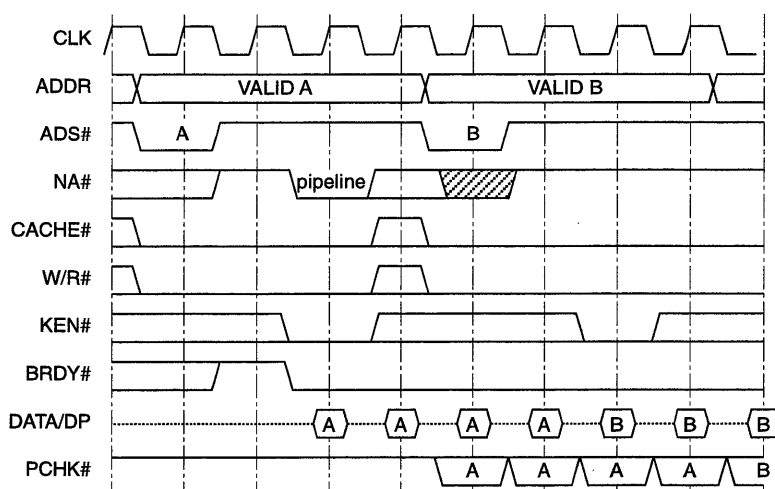


Figure 12-16. Back-to-back Pentium cache-fill timing.

determined by cacheability. A cacheable transaction is a burst of four 64-bit data transfers; all others are single, simple data transfers of 64 bits or less. Consequently, burst support is required in Pentium systems, while 486 systems can choose to implement burst transactions to improve performance or leave it out to simplify the system design. This requirement will likely have little effect on most Pentium system designers, since chip sets will provide the burst support.

Bus pipelining, supported by Intel 386-family processors but not by the 486, allows Pentium to begin a new external access while a previous access is still uncompleted. Pentium supports up to two pending bus cycles with the NA# signal.

Figure 12-16 shows a bus timing diagram for two back-to-back pipelined cache line fills. Each four-transfer cache-fill cycle is begun by simultaneously driving an address and asserting ADS#. Since CACHE# is asserted and KEN# is returned with the first BRDY#, the data is cacheable and the cycle will be a four-transfer line fill. NA# is asserted to pipeline the next line fill. Two cycles after NA#, the next address and ADS# are asserted. KEN# is asserted along with the first BRDY# of the second line fill. The result is two cache-line fills that can proceed at the full bus speed of eight bytes every cycle.

Table 12-11 lists the bus cycles that can be initiated by Pentium and how the bus signals encode them. Note that cycles consist of four transfers if and only if data is cacheable (CACHE# and KEN# asserted). Another type of bus cycle, the inquire cycle (described

below), can be generated by the external system by asserting EADS#.

M/O#	D/C#	W/R#	CACHE#	KEN#	Cycle Description	# of Transfers
0	0	0	1	X	Interrupt acknowledge (2 locked cycles)	1 each cycle
0	0	1	1	X	Special cycle (see text)	1
0	1	0	1	X	I/O read, 32 bits or less, noncacheable	1
0	1	1	1	X	I/O write, 32 bits or less, noncacheable	1
1	0	0	1	X	Code read, 64 bits; CPU deasserts CACHE# to indicate that value will not be cached	1
1	0	0	0	1	Code read, 64 bits; system deasserts KEN# to indicate value should not be cached	1
1	0	0	0	0	Code read, 256-bit burst line fill	4
1	0	1	X	X	Intel reserved (will not be driven by Pentium)	n/a
1	1	0	1	X	Memory read, 64 bits or less; CPU deasserts CACHE# to indicate that value will not be cached	1
1	1	0	0	1	Memory read, 64 bits or less; system deasserts KEN# to indicate value should not be cached	1
1	1	0	0	0	Memory read, 256-bit burst line fill	4
1	1	1	1	X	Memory write, 64 bits or less	1
1	1	1	0	X	256-bit burst write-back	4

Table 12-11. Pentium bus transfer cycle-type definitions.

**Burst Transfer Order.** In burst cycles—either cache-line fills or write-backs—Pentium supplies only the first address. For a cache-line fill, Pentium supplies the address of the data requested by the program; for a write-back, the first address identifies the lowest-order 64-bit word in the line. The other three addresses for the burst line fill or write-back must be generated by external hardware according to Table 12-12, which shows the hex value of the five low-order address bits.

Target Data Address	1st Address Accessed	2nd Address Accessed	3rd Address Accessed	4th Address Accessed
xxxxxx00H	xxxxxx00H	xxxxxx08H	xxxxxx10H	xxxxxx18H
xxxxxx08H	xxxxxx08H	xxxxxx00H	xxxxxx18H	xxxxxx10H
xxxxxx10H	xxxxxx10H	xxxxxx18H	xxxxxx00H	xxxxxx08H
xxxxxx18H	xxxxxx18H	xxxxxx10H	xxxxxx08H	xxxxxx00H

Table 12-12. Pentium burst-mode transfer order.

For example, if a program requests a data word with the low five address bits equal to 08H and the data cache misses,

Pentium supplies the address (xxxxxx08H) of the first 64-bit word, but external hardware must return the next three words from addresses xxxxxx00H, xxxxxx18H, and xxxxxx10H, respectively. The patterns shown in Table 12-12 are analogous to the access sequence followed in 486-based systems, adjusted for Pentium's wider data bus and longer cache lines.

**Special Cycle Types.** The special bus cycles listed in Table 12-13 are provided to indicate that certain instructions have been executed or that certain conditions have occurred internally. As shown in Table 12-11, special bus cycles are subencoded as variations on the impossible case of an attempted write to "code" in the I/O space (M/I0# and D/C# = 0, W/R# = 1). During special cycles, the data bus is undefined and address lines A31 through A3 are driven to zero (unless the address pins are being used for branch tracing). Special bus cycles are acknowledged with BRDY#.

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	Special Bus Cycle
1	1	1	1	1	1	1	0	Shutdown
1	1	1	1	1	1	0	1	Flush (INVD, WBINVD instruction)
1	1	1	1	1	0	1	1	Halt
1	1	1	1	0	1	1	1	Write-back (WBINVD instruction)
1	1	1	0	1	1	1	1	Flush acknowledge (FLUSH#)
1	1	0	1	1	1	1	1	Branch trace message

Table 12-13. Special Pentium bus cycle encodings.

The *shutdown* special cycle can be generated if Pentium gets an exception while it is invoking the double-fault handler or if an internal parity error is detected. The *halt* special cycle is driven after a HLT instruction is executed. The halt state is like shutdown except that halt can be exited by maskable or non-maskable interrupts.

The *flush* special cycle is driven after the INVD (invalidate cache) or WBINVD (write-back and invalidate cache) instructions are executed. The *flush-acknowledge* special cycle indicates the completion of the cache flush operation in response to the assertion of the FLUSH# pin. This operation is implemented as an interrupt to a microcode routine.

The *write-back* special cycle is driven after the WBINVD instruction is executed to indicate that lines marked "modified" in the Pentium data cache were written back to memory or a second-

level cache and that lines marked “modified” in any external caches should then be written back as well.

The *branch trace message* special cycle is driven every time a branch is taken if the execution-tracing enable bit in TR12 (test register 12) is set to one. (BT is asserted on taken branches, regardless.) This special cycle is the only one that does not drive zeros on the address bus; instead, the address bus and BT2..BT0 contain the branch target linear address.

**External Cache Snooping.** External cache snooping occurs when the system asserts EADS# to request a cache consistency check called an “inquire” cycle. Inquire cycles could be used to keep caches and memory consistent during DMA transfers or during cache miss processing in multiprocessor systems. Since the external system must supply Pentium with a snoop address via the address bus, Pentium must first be told via AHOLD to float its address bus. AHOLD must be asserted a minimum of two cycles before EADS# is driven active.

An inquire cycle can have one of two goals: to simply discover if Pentium has an on-chip copy of data, or to cause Pentium to invalidate any on-chip copy. Asserting the INV pin will cause Pentium to invalidate on-chip copies if the snoop hits.

Driving the snoop address and asserting EADS# and INV are done simultaneously (two cycles after AHOLD) to start an inquire cycle. Since an entire cache line is affected by an inquire cycle, only address lines A31 through A5 are significant, but for electrical reasons the other pins must be driven to a valid logic level. The AHOLD/EADS# sequence can be performed even while Pentium is processing a data transfer (the data transfer in progress is not interrupted).

The external system is informed of a snoop hit in the on-chip caches through the HIT# and HITM# signals. These signals are valid two cycles after the assertion of EADS#. HIT# is always asserted if a hit occurred, while HITM# is asserted only if the snoop hits a data-cache line in the M state.

If an inquire cycle hits an M-state line in the data cache, the modified data in the accessed line will be written back immediately so that the line can be invalidated. Figure 12-17 shows a timing diagram for this case in which INV is asserted at the start of the snoop.

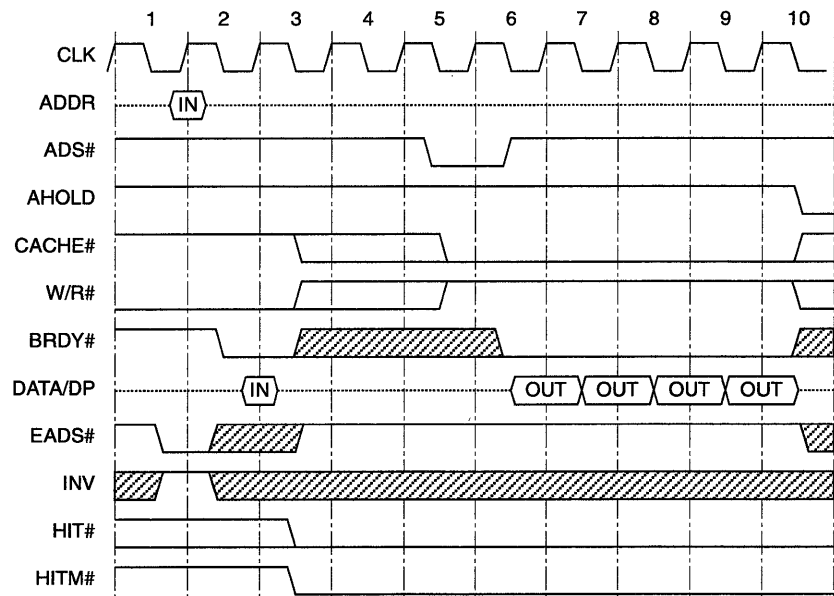


Figure 12-17. Pentium cache-line invalidation sequence timing.

At the end of cycle 1, EADS# and INV are asserted to request an inquire cycle with invalidation. At the end of cycle 2, a previous data transaction is completed. At the end of cycle 3, HIT# and HITM# are asserted to indicate a hit, indicating that Pentium will start a write-back cycle with the next assertion of ADS#. (The only reason ADS# can be asserted during AHOLD is for an inquire-induced write-back.) At the end of cycle 5, ADS#, CACHE#, and W/R# are all driven to signal the start of the write-back. The four write transfers follow. HITM# stays asserted until two cycles after the last BRDY# of the write-back.

Since AHOLD is asserted during the entirety of this write-back transaction, Pentium is unable to drive addresses to the external system. Thus, in this case, the external system is required to drive and sequence all address bits for the write-back data transfers.

If desired, however, the external system can deassert AHOLD before Pentium begins the write-back cycle (before cycle 5 in Figure 12-17) to cause Pentium to drive the write-back address on the address bus. This can be done to simplify external hardware a little or to account for the possibility of an address parity error on an inquire cycle (see below). Even in this case, the



external system is still responsible for sequencing addresses (as in all burst transactions).

Inquire cycles always snoop the internal instruction and data caches, but if the snoop is requested during a cache-line fill, Pentium also snoops the line currently being filled (in a read buffer). If more than one cacheable cycle is outstanding because of address pipelining, Pentium snoops both transactions.

Similarly, if an M-state line is in a write buffer in the process of being written back, Pentium will snoop the write buffer on behalf of an inquire cycle. In this case, Pentium asserts HIT# and HITM# as usual, but there will not be a separate write-back of the M-state line, since it was already in progress.

Address parity is checked for inquire cycles, but Pentium can do nothing about parity errors; if an address parity error occurs, the snoop cycle is not inhibited. If an inquire hits an M-state line and AHOLD remains asserted, it is not possible for Pentium to drive the address bus to tell the system what address was actually used for the snoop.

Thus, it is possible that Pentium will start a write-back of incorrect data, and if the external system uses the address it supplied to Pentium for the inquire cycle, memory could be corrupted. In light of how much Intel is making of Pentium's error-checking capabilities, it seems odd that address-parity errors are not handled more gracefully.

**External Program Monitoring.** As on all highly integrated processors, it is difficult to monitor program behavior on Pentium in detail because so much activity is occurring only between on-chip components. To address this problem, Pentium has many pins that expose internal operations and allow external program monitoring. These pins include BP[3..0] (breakpoint) and PM[1..0] (performance monitoring); BT[3..0] (branch trace); IBT (instruction branch taken); and IU and IV (instruction completed in pipelines).

The four breakpoint pins correspond to internal debug registers, and the pins are asserted when a match is detected in the corresponding debug register. While BP3 and BP2 have dedicated pins, BP1 and BP0 are multiplexed with PM1 and PM0. Unfortunately, the PM pin functions are covered in Intel's secret Appendix A.

IBT is asserted each time Pentium takes a branch. If enabled, each assertion of IBT is accompanied by a special bus cycle, the branch trace message special cycle (see Table 12-13). On each of these cycles, pins BT[3..0] are also valid. They provide the low three bits of the branch target address (unavailable on the address bus) and tell whether the branch was a 16-bit or 32-bit instruction.

IU and IV simply indicate each instruction completion in the respective instruction pipeline. Note that IBT will be accompanied by either IU or IV and that IU and IV can be (and, it is hoped, often are) asserted simultaneously.

The INIT pin is a new “warm restart” pin that causes a reset-like action but does not cause the values in caches and FP registers to be lost. INIT can be used to switch via hardware from protected mode to real mode. Also, holding INIT high during reset invokes an automatic built-in self-test mode.

## System Management Functions

Pentium was the first high-performance microprocessor to implement a system management mode (SMM). Ordinarily, SMM capability is designed into processors intended for portable applications in order to facilitate power-saving functions such as powering down idle peripherals and restarting them only when they are accessed. Intel had initially implemented SMM functions in its i386SL and i486SL, and has since added SMM to all 486 processors.

SMM is an operating mode that takes precedence over all other modes and interrupts. Just as interrupts and traps allow an operating system to transparently add functions to application software, SMM allows software functions to be added to a system without making changes to the operating system.

Pentium support for SMM consists of the SMI# interrupt input pin, the SMIACK# status output pin, and the RSM instruction. Triggering SMI# is the only way to enter SMM. When SMI# is detected, the SMIACK# pin is asserted in order to enable a special SMM memory area (SMRAM). Pentium then saves its register state in SMRAM and disables further interrupts. Interrupts may be re-enabled in SMM after taking care to set up correct interrupt vectors.

By default, SMM begins execution at address 00008000H in the code segment. The SMRAM memory space is essentially a flat, four-gigabyte, real-address-mode linear address space. The default operand and address sizes are set to 16 bits, but oper-

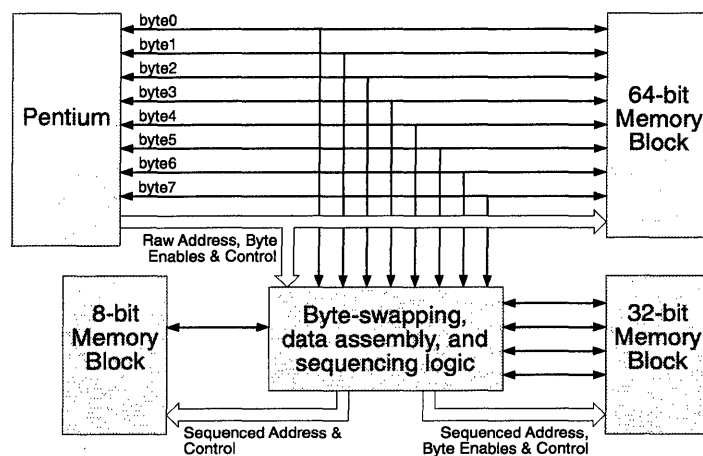


Figure 12-18. Packing/unpacking logic for partial-word transfers.

and-size and address-size override prefixes can be used to access data and code anywhere in the four-gigabyte SMRAM space. When the SMM routine completes, the previous machine state is restored and SMM is exited with the special RSM instruction.

SMM can be used to implement power savings, security options, and other features. While SMM will likely not be used in many Pentium desktop systems, Intel has decided to provide SMM functions on all mainstream x86 processors. Systems based on later Pentium derivatives will undoubtedly make better use of SMM than earlier Pentium systems.

### Bus Sizing

Pentium has a 64-bit bus, but some system implementations will probably prefer a 32-bit memory system. Unfortunately, Pentium requires that all of the enabled bytes for a given cycle be returned from memory to the processor simultaneously. Thus, for narrower memories, such as 32-bit RAM and byte-wide bootstrap PROMs, external logic is required to sequence addresses, swap bytes, and buffer data, as shown in Figure 12-18.

Efforts to simplify Pentium's bus controller led the designers to eliminate the BS8# and BS16# inputs (8-bit and 16-bit bus-size indicators) that allowed the 486 to work easily with narrow devices. For most systems, this is probably not much of an issue, since the required logic can, and therefore will, be incorporated into 32-bit Pentium chip sets. Also, Intel may develop reduced-width bus versions of Pentium for specific markets.

## Functional Redundancy Checking

One hardware function that is unique to Pentium (unique, that is among x86 and mainstream RISC workstation CPUs) is a feature called Functional Redundancy Checking (FRC). Functional redundancy provides a mechanism to enhance system integrity by using a second Pentium microprocessor to verify the correctness of all operations performed by the first.

Figure 12-19 shows the interconnections needed to support FRC operation. The FRCMC# (Functional Redundancy Check Master/Checker) input on the first device is driven by a high-level logic signal, causing the device to operate as a system “master.” FRCMC# is driven low for the second processor, causing it to operate as a system “checker.” All other signal pins on the checker processor—inputs, outputs, and bidirectional signals—except the IERR# (internal error) pin are connected directly to the corresponding pin on the master.

As its name suggests, the master processor controls system operation, driving its outputs and reading its input signals as for “normal” operation; indeed, this is the mode in which CPU operates for single-processor, non-FRC systems. On the checker processor, all of the I/O pins that operate as inputs continue to behave normally, but the output drivers for pins that would otherwise be output are disabled. Instead, the logic level driven onto each of these pads is sampled during every clock cycle.

Both CPUs begin operating during the same clock cycle following reset. Since both CPUs are otherwise identical, since Pentium systems are fully synchronous, and since operation is fully deterministic, each address generated by the master will simultaneously be generated by the checker, each instruction retrieved and executed by the master processor will be read and

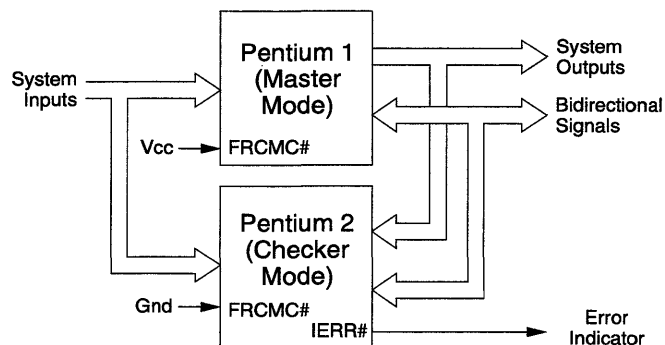


Figure 12-19. Functional redundancy checking interconnections.

executed simultaneously by the checker, and so forth. Execution should therefore proceed in lockstep indefinitely thereafter.

During every clock cycle, the checker-mode processor reads the logic level driven onto its output pins by the corresponding output pin of the master processor. Comparator circuitry checks whether that level matches the value that it (the checker) would have emitted, had it been configured as a master. As long as each device operates properly, all such signals should continue to be equal.

If a mismatch is ever detected, then, at least one of the devices must have malfunctioned at some previous time. The checker-mode CPU asserts its  $\overline{\text{IERR\#}}$  output pin and halts. External logic may be designed to freeze system operation when  $\overline{\text{IERR\#}}$  is asserted, thus preventing the completion of any memory or I/O cycle that might otherwise corrupt system data.

FRC operation makes possible a slightly more elegant scheme that supports not just fault detection but fault tolerance. In this scheme, known as Quad Redundancy Checking, two more identical Pentium CPUs, also configured as an FRC pair, act as a “hot backup” to the first. As long as neither pair detects an error, the first pair drives its outputs onto the system bus. If the first pair detects a hardware failure, system logic disables its system-level bus drivers and immediately enables those of the second pair. Operation can thus proceed smoothly, despite a hardware failure.

On the other hand, if the second CPU pair detects an internal hardware failure, system hardware would presumably deactivate the hot backup and notify the system user to perform a graceful shutdown and contact a maintenance engineer.

Note that activation of the FRC capability is completely optional. Motherboards can even be designed that include one CPU configured as a master but provide just an empty socket for the checker. Such a board would, by default, be fully functional. Merely inserting a second, identical, CPU into the empty socket would immediately enable FRC verification.

Note, though, that for FRC operation to work, the two CPUs must be absolutely identical in every respect. Even the most subtle difference in microcode could cause the CPUs to break out of lockstep execution. Thus, it is necessary for OEMs to be able to verify the exact product, microcode version, and mask set used to generate each device; this is Intel’s justification (at

least, its *public* justification) of providing the CPUID instruction that returns family, product, and stepping information.

(It's interesting to note that FRC operation was not actually invented for Pentium. This capability first appeared on the iAPX432 micromainframe—the super-sophisticated CPU whose schedule slippages prompted the creation of the original 8086. More recently, FRC capabilities have appeared on several members of the Intel i960 embedded-processor family.)

### **Vital Statistics**

The 0.8 $\mu$  Pentium design is offered in 60- and 66-MHz versions. Intel claims the device yields well at 66 MHz. On the other hand, the performance difference between the two speed grades is (at best) only 10%, hardly enough to incite customers to move from one to the other. The existence of an only-slightly-slower option suggests that a significant number of chips don't quite work at the full target frequency. Even at 60 MHz, though, Pentium nearly doubles the performance of Intel's top-of-the-line i486DX2 processor on recompiled code.

The 5-V 0.8 $\mu$  Pentium is a hot, hot chip indeed. Typical power is quoted at 13 watts at 66 MHz, with a maximum power dissipation of 16 watts. This is a major jump from the first 486, which used less than 4 watts. Even the i486DX2, which Intel ships with its own 0.35-inch heat sink, peaks at 6 watts.

The chips are shipped without a heat sink, giving system vendors a choice. With a heat sink similar to the i486DX2's, the system must provide a gale-force airflow of 650 ft/min to cool the CPU in ambient temperatures up to 40° C. With a 0.65-inch heat sink, the airflow can be reduced to a merely stormy 300 ft/min.

Designers of large systems are used to such power levels, and Pentium dissipates considerably less heat than the PA7100 or Alpha chips, both of which exceed 20 watts. Since PC fans typically provide only 50–100 ft<sup>3</sup>/min of airflow, however, PC designers must rethink the thermal engineering of their system designs to allow for such a high-powered chip.

## 12.2 The Intel 0.6 $\mu$ Pentium “P54C”

In March of 1994 Intel introduced a second member of the Pentium family code-named the “P54C.” This product is based on the Pentium core design but built using a 0.6-micron, 3.3-V BiCMOS process. (Intel also calls this part simply “Pentium,” but it designated the “0.6 $\mu$  Pentium” hereafter in this report.) The part can operate at core frequencies up to 100 MHz, 50% faster than the 0.8 $\mu$  design. The new process also significantly decreases the power dissipation and manufacturing cost relative to the 0.8 $\mu$  design. Table 12-14 summarizes the features of the 0.6 $\mu$  device.

Product Name	Intel 0.6 $\mu$ Pentium “P54C”
Introduction Date	March 1994
Device Integration Level	Same as 0.8 $\mu$ Pentium plus advanced priority interrupt control logic
CPU Architecture Level	Standard Pentium IU and FPU instruction set
Core Technology	Superscalar dual 486-like pipelines
Pinout Standard	Extended Pentium pinout
Data Bus Width	64 data bits plus 8 parity bits
Physical Addressability	4 gigabytes (Address pins A31..A3 plus BE7#..BE0#)
Data-Transfer Modes	Four $\times$ eight-byte burst-mode transfers
Cache Support	I-cache: 8K-byte split-line 2-way associative D-cache: 8K-byte 2-way associative with 8-way interleaved access and write-through or copy-back operation
Floating-Point Support	On-chip pipelined high-speed FPU
Operating Voltage	3.15 V to 3.45 V
Frequency Options	75-, 90-, or 100-MHz core frequency
Clocking Regime	Core frequency = 1.5 $\times$ or 2 $\times$ Clkin
Maximum Power Dissipation	10.1 W @ 3.3 V and 100 MHz (worst case)
Power-Control Features	Intel System Management Mode support Clock disabled to unused logic dynamically Stop-Clock and Auto-Halt modes I/O instructions may be trapped and restarted
Process Technology	0.6 $\mu$ BiCMOS, four-layer-metal
Transistor Count	3.3 million transistors
Die Size	13.3 $\times$ 112.3 mm (163 mm <sup>2</sup> )
Package Options	296-pin ceramic PGA package
Other Features	Functional redundancy system support JTAG boundary-scan logic, On-chip parity and integrity-checking logic On-chip APIC controller for glueless dual processing

Table 12-14. Intel 0.6-micron Pentium “P54C” feature summary.

**Overview** When Intel first began leaking information about the Pentium processor, it looked almost too good to be true. Indeed it was. Upon introduction, system designers discovered the 0.8 $\mu$  Pentium devices were expensive to build, difficult to design with, used nearly three times as much power as a 486, and yet had a clock speed no faster than a 486DX2. Any of these problems could have prevented Pentium from ever becoming a volume desktop CPU.

Fortunately, Intel has found that a new IC process resolves these issues. The 3.3-V, 0.6 $\mu$  Pentium "P54C" design is everything the 0.8 $\mu$  Pentium was supposed to be: a 100-MHz processor with reasonable power dissipation and moderate manufacturing cost. It includes a few minor enhancements: a variable clock multiplier circuit, power management, and an interrupt controller intended to facilitate multiprocessor system designs.

The new chip is not socket-compatible with the 0.8 $\mu$  Pentium, both because of its 3.3-V-only operation and various pinout enhancements. With over 100-SPECint92 performance, the 0.6 $\mu$  Pentium is a potent weapon against PowerPC and other RISC processors. The cost reductions also will allow Intel to cut the price of Pentium and eventually move it into the PC mainstream.

This chip is one of the first products to use Intel's new 0.6-micron fabrication process, now on line at Fab D2 in Santa Clara, Calif., and at Fab 10 in Leixlip, Ireland. Intel had also planned to begin 0.6-micron production at Fab 11 in Albuquerque, New Mexico by the end of 1994. These fabs, which use 200-mm wafers instead of 150-mm ones, will greatly increase Intel's manufacturing capacity and remedy its current inability to meet demand for its processor chips.

The benefits of moving Pentium to the new process are clear. The new process shrinks the drawn transistor size from 0.8 micron to 0.6 micron, reducing circuit area by more than 30%. A fourth layer of metal, which is used to route power and clocks while the other three metal layers carry signals, reduces area by another 10%. The BiCMOS process incorporates both CMOS and bipolar transistors, which can be combined to form BiNMOS drivers that reduce the transmission delays of heavily loaded signals.

Although these features reduce die area and improve performance, they also increase wafer cost by about 30% over a three-



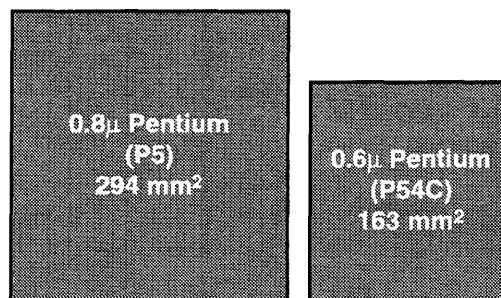


Figure 12-20. 0.8μ and 0.6μ Pentium die size comparison.

metal CMOS process with a similar transistor size. Taking into account the smaller geometries and larger wafer size, total wafer cost for the new process is more than twice that of Intel's 0.8-micron BiCMOS process. Given Intel's high production volume, the company is willing to trade reductions in die area against increases in wafer cost.

Intel is continuing to invest heavily in more advanced fabrication methods and plans to accelerate its IC process development cycle from three years to two. The company is building still another new factory in Albuquerque that will use a 0.4-micron process that is reportedly CMOS only (not BiCMOS). Intel expects that this new process will be in production by the end of 1995 and will be used for both the Pentium and "P6" microprocessor families.

The 0.8μ Pentium has a die size of 294 mm<sup>2</sup>. (See **Chapter 15** for estimates of device manufacturing costs.) Figure 12-20 shows that the 0.6μ Pentium die measures just 163 mm<sup>2</sup>, a reduction of 45%. More important, the estimated cost of building Pentium is reduced by more than half. This cost will drop by another 25% or so as the new process matures.

The new design will also help Intel greatly increase the number of Pentium processors it can produce. Whereas the 0.8μ Pentium design was thought to yield four to six good die per wafer, according to MicroDesign Resources estimates, the 0.6μ Pentium should yield more than 30 die per wafer. (Unconfirmed rumors have suggested that the very earliest test wafers from Intel's new Ireland plant produced as many as 40 die per wafer at 100 MHz, and that later wafers yielded up to 60 working die!)

Three factors create this sixfold improvement. Proportionately more of a smaller die will fit in a given wafer area, and the 0.6-micron factories use 200-mm wafers with 80% more area than the 150-mm wafers used in older fabs. Moreover, smaller die are less likely to contain defects, so a higher percentage of the total die manufactured are generally good. The increases in yield are somewhat offset by higher wafer processing costs, but the net effect is still much improved.

The 0.6 $\mu$  Pentium takes several steps to address the power consumption issue. Cutting the supply voltage to 3.3 V reduces the power dissipation by 50% at a given core frequency. Maximum power dissipation for the 0.6 $\mu$  Pentium is just 10 W (worst case) at 100 MHz compared to 16W for the 5-V Pentium at 66 MHz.

To further reduce power, the new design automatically stops the clock, on a cycle-by-cycle basis, to the caches or to the floating-point unit when those circuits are not being used, reducing power with no effect on performance. Intel claims these features reduce the average power dissipation to less than 4 W in a typical application.

The 0.6 $\mu$  Pentium also includes the full SL Enhanced feature set used in Intel's current 486 processors. These features include system management mode, the ability to stop and quickly restart the processor clock, and an automatic power-shutdown mode. While the 0.8 $\mu$  Pentium also supported SMM, the "stop clock" and "auto halt" features are new to the 0.6 $\mu$  Pentium.

These power-reduction features make the 0.6 $\mu$  Pentium processor far more suitable than the 0.8 $\mu$  Pentium for use in notebook systems, since battery life depends on typical power consumption. System designers must still allow for worst-case current and cooling capacities, but in a notebook system an active power-management system might be able to monitor the temperature of the CPU and slow the clock if the chip is getting too hot. Pentium notebook systems should begin rolling out by Fall '94 Comdex and become widespread in 1995.

### **Bus Interface**

The 0.6 $\mu$  Pentium uses a 296-pin PGA, 23 pins larger than the 0.8 $\mu$  Pentium package. Three of the new pins are used for the APIC, and most of the rest are defined as no-connects to allow for "future functional enhancements." Because of the new package and the 3.3-V I/O, it is impossible to simply drop the P54 processor into an existing Pentium motherboard; in fact, upgrading to 0.6 $\mu$  Pentium will require a significant redesign.

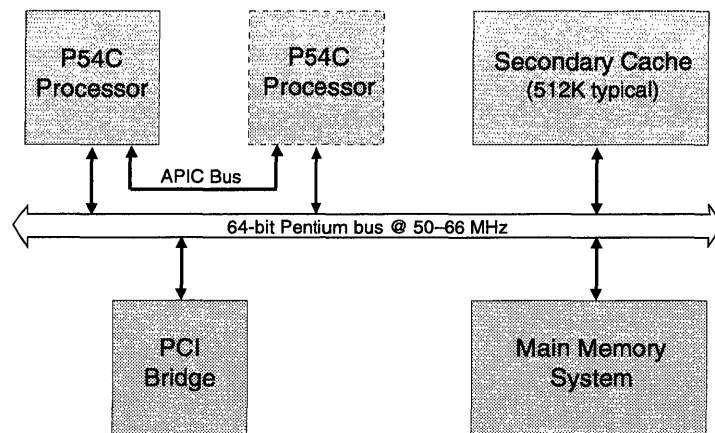


Figure 12-21. 0.6µ Pentium multiprocessing system architecture.

The 0.6µ Pentium supports only 3.3-V I/O signals, forcing system designers to use low-voltage cache memory and chip sets. Intel has released a 3.3-V version of its 82430 chip set with 5-V level translators for new designs, and expects that similar chip sets will be available from Opti, VLSI, and others by the end of 1994. By forcing a move to 3.3 V, Intel is preparing system vendors to support future Pentium chips and Pentium upgrades, all of which will use 3.3-V I/O.

### Multiprocessor Support

While the 0.6µ Pentium core is functionally nearly identical to the 0.8µ Pentium design, Intel has taken the opportunity to make a few functional improvements. In addition to the new SL Enhanced features, the design now incorporates Intel's advanced priority interrupt controller, or APIC, making the new chip suitable for a glueless dual-processor configuration in which two CPUs share the same cache (see Figure 12-21).

Since this dual-processor configuration shares a single bus and L2 cache, it would not deliver the same performance boost as a traditional MP design with separate L2 caches for each CPU. Intel estimates the gain to range from 30% to 70%, depending on the application. This design would be much less expensive than a traditional multiprocessor configuration, however, since the only expense of adding the second processor to a system would be the cost of the CPU chip itself.

Intel could, of course, have left off the APIC and assumed that it would be in the system logic. Chip-set vendors balked at the added cost of the APIC, however, and any chip sets that include the logic to support dual processors will be more expensive than

standard uniprocessor chip sets. Intel believes that this cost differential, in the highly competitive PC market, would have led to a dearth of MP-capable systems. By including the APIC, which uses less than 5% of the die area, on the CPU, system vendors can offer dual-processor capability for the cost of a second CPU socket, seeding the market with lots of these systems.

Of course, to take advantage of the second processor at all, a multiprocessor operating system is required. And unless an application is multithreaded, the second CPU is active only when two or more tasks are running. Given that neither DOS nor Windows (including the future Chicago version) can handle multiple processors, Intel expects that the dual-processor Pentium will be used primarily for high-end desktops or servers running UNIX or Windows NT.

In these high-end markets, the dual-processor mode can be used as an upgrade strategy for the 0.6 $\mu$  Pentium. For the majority of users, however, Intel will provide a traditional upgrade chip that usurps control of the system from the original CPU. The company will not discuss any specifics about this upgrade part but expects it to be available in 1996. Thus, it is possible that the upgrade will take advantage of the P6 processor core.

### **Interrupt Control Logic**

The advanced priority interrupt controller (APIC) architecture included in the 0.6 $\mu$  Pentium first appeared in late 1992 in the form of the 82489DX. The APIC architecture replaces the old 8259A interrupt controller that was originally designed for the 8080, modified for the 8085, and inherited by all PCs since then. With a more flexible priority scheme and faster response time, the APIC has some benefit for uniprocessor systems, but its major advantage is in supporting multiprocessor systems, which is not possible with the simple 8259.

The APIC is physically divided between the processors and the system logic. The "I/O APIC," typically part of the system logic, accepts system interrupts much like the 8259. Unlike the older part, however, the I/O APIC can transfer pending interrupt requests to other processors, each of which must have its own local APIC module. The various APIC modules are connected via a private interrupt bus, allowing interrupts to be communicated without disrupting the normal system bus.

The 82489DX has not been widely used. Most vendors with multiprocessor x86 systems had already defined their own MP interrupt protocol and saw no reason to change, although a few have adopted the APIC. Desktop systems have not incorporated

the 82489DX due to its cost, and system-logic vendors have seen no reason to incorporate a complete APIC (both I/O and local modules) into their chip sets.

The 0.6 $\mu$  Pentium integrates the local APIC module and communicates to the I/O APIC and other 0.6 $\mu$  Pentium processors using a three-wire bus. Although the 0.6 $\mu$  Pentium implementation is register-compatible with the 82489DX, the three-wire bus is not compatible with the 82489DX's five-wire protocol. Intel now includes the I/O APIC logic in its support chip sets and is licensing the design to other system-logic vendors. The I/O APIC is relatively small, and Intel expects most vendors to include it in their basic chip sets.

For compatibility with software that does not include APIC code, the 0.6 $\mu$  Pentium can disable its on-chip APIC and use an external 8259-type controller. Today, few software vendors support the APIC, although a special HAL for Windows NT is available. By increasing the installed base of APIC hardware, Intel hopes to spur other MP operating systems to support it.

### **Clock-Generator Circuitry**

The other new feature of the 0.6 $\mu$  Pentium design is a phase-locked loop (PLL) that lets the CPU run at 1.5 $\times$  or 2 $\times$  the system bus frequency. This keeps the system bus between 50 and 66 MHz while the CPU runs as fast as 100 MHz. Although the 2 $\times$  ratio allows for a 100/50-MHz system, this configuration would perform comparably to a 90/60-MHz design, so vendors may try for a 100/66-MHz arrangement to maximize performance. The 0.6 $\mu$  Pentium clock multiplier is pin-selectable; there is a single 100-MHz version of the chip that supports either bus frequency.

### **Vital Statistics**

The 0.6 $\mu$  Pentium design is offered in 90- and 100-MHz versions. Power dissipation at 3.3 V and 100 MHz is 4 W typical, or 10.1 W worst-case.

## 12.3 Futures

At the ISSCC conference held in February of 1994, Intel displayed a 150-MHz Pentium processor; this same system was later demonstrated continuously for several hours in a suite upstairs. This particular chip was no doubt hand-picked from the production line and operated with a special power supply and cooling unit, but Intel hinted that the 150-MHz Pentium may eventually become a product. On the other hand, Intel also presented a 100-MHz 486 at the 1991 ISSCC but three years and a new turn of the manufacturing technology were needed before it was able to ship such a part. Once Intel has progressed to a 0.4-micron process, faster Pentia will be possible—and therefore inevitable.

### The Intel “P24T”

One of the most eagerly awaited but not yet announced products in the history of the microprocessor industry is a product code-named the “P24T.” This device is supposed to serve as an end-user upgrade for i486DX2-based systems. While the product itself has not been announced or completed, its pinout has been defined since mid-1992, and many system vendors have been cranking out and selling PCs with empty upgrade sockets in anticipation of the device since 4Q92.

In the mean time, P24T plans have undergone some major changes; only the pinout standard has remained unchanged. Current expectations are that the device will be derived from the 3.3-V 0.6 $\mu$  Pentium core, with a core frequency 2.5 $\times$  the bus clock, twice the on-chip cache, a 32-bit data bus interface, and a pinout that’s a superset of that of standard 486 OverDrive processor.

Unfortunately, the motherboards that await these chips were designed for a 5-V part—so the P24T is now expected also to incorporate an on-module heat-sink, voltage regulator, and fan.

## 12.4 Commentary

Pentium was a long time coming. The first public mention of the device promised a 1H92 introduction, with “volume system shipments by the end of 1992.” The introduction schedule later slipped to the end of 1992, then to spring of 1993. A month before the expected May 1993 unveiling, Intel revealed that certain key issues, including the price and performance of

Pentium-based systems, would not be discussed until May. Shortly thereafter, an e-mail message began circulating through the Internet, purporting to identify (in the best David Letterman fashion) the “Top Ten Reasons Why Pentium Was Late.” This list appears in Table 12-15.

#	Reason
10:	Quality control complained about the rattling noises the chip makes whenever it's reset.
9:	Intel hoped to outfox AMD developers this time by waiting for them to release their “Pentium” first.
8:	Intel's still trying to figure out how to mount a three-foot high cooling tower on a two-inch square package.
7:	Marketing's prediction that all of IBM's top executives would be killed by space aliens, followed by IBM engineering's insistence on a return to an Intel strategy, did not appear to pan out.
6:	The sales force needs to be retrained to sell a processor that doesn't end in “86”.
5:	As a result of poor documentation practices, nobody can remember what the function of the WOOF* pin is.
4:	Military insisted at the last minute on 8080-compatibility mode.
3:	Employees complained about being harassed by engineers who offered to demonstrate “Probe Mode”.
2:	All those millions of dollars in processor research and development were cutting into the CEO's Christmas bonus.
1:	Intel needed to hire more lawyers.

Table 12-15. “Top Ten Reasons” why Intel delayed announcing Pentium.

Despite the vast manufacturing cost reduction of the 0.6 $\mu$  Pentium, Intel says that it will continue to build the 0.8 $\mu$  5-V Pentium device for the foreseeable future. Because of the extensive redesign required for the new version, many vendors will continue to ship systems using 60- and 66-MHz Pentium chips for some time. The lower frequencies also provide additional performance points for Pentium systems, although the 100-MHz DX4 overlaps the 60-MHz Pentium on some applications.

Continuing the 5-V line also maximizes the number of Pentium chips that Intel can produce. By the end of 1994 Intel expected to have three factories building the 0.6 $\mu$  Pentium design along with the two currently making the 0.8 $\mu$  version. Given the amount of fab capacity coming on line, the company may find

itself obligated to build a large number of chips simply to defray the costs of building the new fabs.

Thus, Intel is promoting Pentium chips aggressively, cutting prices on the 0.8-micron versions even though they are more expensive to build than the higher-speed chips. These price cuts make room for the higher-speed parts.

These price cuts will reduce the margins on the 0.8 $\mu$  Pentium parts below the high margins of Intel's other processors, but the company will still make a significant profit on them. Furthermore, by increasing the penetration of Pentium in the PC market, Intel devalues the product offerings of its 486-based competitors.

To build further momentum for Pentium, Intel must convince PC makers to move beyond the 5-V, 33-MHz system bus that they are comfortable with. Because of the difficulty of designing with the 60-MHz system bus, many companies buy complete Pentium motherboards directly from Intel. By some accounts, Intel is the largest vendor of Pentium motherboards in the world. Unless Intel plans to continue growing its motherboard business, which upsets those vendors that actually have the resources to design their own products, it must provide simple design kits and chip sets that can handle the faster Pentium processors.

At first glance, Pentium may appear to be not as aggressive in its issue strategy as some of the latest RISCs. It is limited to two instructions per clock, while SuperSPARC and IBM's RS/6000 can issue three instructions under optimal circumstances. Upcoming versions of the Power2 family will be able to issue up to six instructions per clock. Pentium cannot issue integer and floating-point operations in the same cycle, as can all superscalar RISCs.

Still, the Pentium family appears to have found ways to overcome many of the x86 architectural handicaps. The small register set, for example, means that there are more memory references, but the dual-access data cache helps minimize the performance impact. The stack-oriented floating-point register file creates an accumulator bottleneck, but the parallel execution of the exchange instruction reduces its effect. In this sense, Pentium appears to support the contention that the x86's architectural handicaps can be overcome with some implementation creativity.



But in doing so Pentium shows the extent to which its CISC architecture increases its design complexity. For example, RISC processors have not had to go to the complexity of dual-access data caches to reach comparable performance levels. This also illustrates how the x86's architectural limitations affect many aspects of the design; it is not as simple as designing a nice, clean RISC processor with a small "compatibility unit" on the side, as some proponents have described Pentium. While Pentium achieved the same performance as the R4000, it did so a year later, and required three times as many transistors and with a more complex process technology.

The richer semantic content of x86 instructions, however, means that two instructions often do the same work that would require three or more instructions in a RISC architecture. For example, the memory-to-register instructions in the x86 architecture eliminate the need for separate load and store instructions. This also makes it less important to issue floating-point and integer instructions together, since many of the integer instructions in floating-point programs are loads and stores. Address-calculation instructions are also sometimes eliminated by the x86's richer addressing modes.

While any x86 program will benefit from Pentium's performance features, the full performance potential will be realized only for programs that are structured to take maximum advantage of Pentium's capabilities. Instruction sequences must be carefully selected to use the instructions that can be dual-issued and scheduled to fill all available execution slots.

With respect to its FPU, Pentium will bring a new level of performance to the PC market. It will not, however, outperform its Windows NT competitors because of the weaknesses of its floating-point architecture and because the R4000 and Alpha processors will be operating at much higher raw clock speeds.

Unfortunately, this same high-speed FPU has cost Intel dearly. In November of 1994 it was discovered that the high-speed algorithm used to cut FPU divide times in half was not quite always 100.000% accurate. See **Appendix E** or details on the Pentium FPU bug.

In some ways—the elimination of RDY#, simpler burst determination, and no bus sizing—Pentium has a simpler bus than the 486. The addition of new features, however, such as pipelining, cache snooping, and forced burst support, means that system hardware will need considerably more sophisticated bus con-

trollers, especially if second-level caches are used in multiprocessor systems. As with earlier x86 generations, chip-set vendors will likely spare system makers the headache of designing bus-control state machines and interface logic.

Other than NexGen, no vendor has delivered, or even formally announced, a Pentium-class product. AMD's K5 and Cyrix's M1 aren't likely to begin volume shipments until mid- to late-1995. The first real Pentium competition may emerge from NexGen, which has been sampling its 586 processor since early 1994. Because of its small size and fabless status, it will take time for NexGen to build enough market presence to make much of a dent in the Pentium market.

Thus, Intel can still wield the 0.6 $\mu$  Pentium largely unopposed in the high-end market, leaving its competitors to fight over the low-margin bottom end. Intel will not abandon the low end; profits from its flagship products will subsidize heavy discounting of i486s and other low-end chips.

The market is moving much faster than in the past, however, and Intel's monopoly will not last as long as its four years of 486 dominance; Cyrix plans to bring its "M1" product to market by mid 1995 and has IBM's fab on its side. Ultimately, Intel will have to rely on aggressive pricing to maintain its market share, but it has the R&D and manufacturing skills to succeed in this competition.

### **Competition with RISC**

Pentium will allow Intel to protect its enormous and highly profitable market share from competing RISC and x86-compatible vendors. Intel should be able to manage the price of its chips so that Pentium systems remain competitive against low-end workstations. Initially, Intel need only maintain parity in price/performance, as the overwhelming x86 software base will work to its advantage, but Windows NT may begin to level the playing field if it gains in popularity.

For commercial applications, such as database and file servers, the performance advantage of the RISC chips over Pentium is smaller, since these applications rely mainly on integer performance. These large servers often use multiple processors, which can further negate the absolute performance advantage of RISC. For these designs, the price/performance of the processor is most important.

On the desktop, simply matching Pentium will not make up for the overwhelming software advantage of the x86, leaving the

RISC chips with an uphill battle. DEC and MIPS will try to offer superior performance at the same price; MIPS also has the advantage of two very low cost chips (the R4200 and Orion) available. HP may use the 7100LC's multimedia acceleration as a differentiator. IBM/Motorola's PowerPC 601 has a much smaller die size and a correspondingly lower cost than Pentium.

Intel has shown that it will not relinquish its performance leadership willingly. Given the ability of PowerPC to deliver similar performance at a lower CPU price, Apple and other system vendors are attempting to translate this advantage into a system-level price/performance advantage, but the jury is still out on whether their customers will care. Until PowerPC can improve its position, it will not be a serious threat to Intel's sales. And in the meantime, the new x86 chips should allow Intel to continue its dominance of the x86 market while generating its traditional enormous profits.

Recently, workstations have been replacing PCs on the desktop of some professionals, such as engineering managers and financial analysts, who use a mixture of commercial and technical applications. This market segment, potentially much larger than the pure technical segment, may prove to be fruitful ground for Pentium "workstations" that offer near-RISC performance plus full compatibility with over 50,000 x86 applications.

Pentium is a significant microprocessor milestone. It implements sophisticated caching, multiprocessor support, and branch prediction. It is also the first superscalar CISC microprocessor and the first high-end microprocessor to implement a system management mode. The Pentium core will be around for years as Intel attempts to exploit its lead by offering variations with varied cache sizes, bus widths, and bus speeds. As for Pentium's technological position in the marketplace, some RISCs will be faster or cheaper or both, but with x86 compatibility, multiprocessor support, and significant performance gains over the 486, Pentium will satisfy most users' needs.

## 12.5 For More Information...

Additional information on Pentium may be found in the following publications:

- |                                       |   |
|---------------------------------------|---|
| <b>Vendor Publications</b>            | <ol style="list-style-type: none"> <li>1: <i>Intel Corporation Advances Pentium Processor Technology to Notebook Computers Press Kit</i>. Intel Corporation, 10/10/94.</li> <li>2: <i>Intel Pentium Processor (610/75) Performance Brief for Mobile Applications Release 1.0</i>. Intel Corporation, 9/94.</li> <li>3: <i>Microprocessors Data Book Volume III: Pentium Processors</i>. Intel Corporation, 1994, order #241732-001.</li> <li>4: <i>Pentium Processor Performance Brief</i>. Intel Corporation, 1993, order #241557-001. (<i>Brochure listing Intel performance results for assorted applications.</i>)</li> <li>5: <i>Pentium Processor User's Manual Volume 1: Pentium Processor Data Book</i>. Intel Corporation, 1994, order #241428-002.</li> <li>6: <i>Pentium Processor User's Manual Volume 2: 82496 Cache Controller and 82491 Cache SRAM Data Book</i>. Intel Corporation, 1994, order #241429-002.</li> <li>7: <i>Pentium Processor User's Manual Volume 3: Architecture and Programming Manual</i>. Intel Corporation, 1993, order #241430-001.</li> </ol> |
| <b>Microprocessor Report Articles</b> | <ol style="list-style-type: none"> <li>8: <i>Intel Announces MESI Second-Level Cache*</i>. Mark Thorson, MPR vol. 5 no. 12, 6/26/91, pg. 8. (<i>Feature article.</i>)</li> <li>9: <i>P5 Details Surfacing</i>. MPR vol. 5 no. 18, 10/2/91, pg. 4. (<i>Most Significant Bits item.</i>)</li> <li>10: <i>P5 Rumor Update</i>. MPR vol. 5 no. 22, 12/4/91, pg. 5. (<i>Most Significant Bits item.</i>)</li> <li>11: <i>First Silicon on Intel's P5</i>. MPR vol. 6 no. 7, 5/27/92, pg. 4. (<i>Most Significant Bits item.</i>)</li> <li>12: <i>P5 Not to be Called the 586?</i>. MPR vol. 6 no. 9, 7/8/92, pg. 5. (<i>Most Significant Bits item.</i>)</li> <li>13: <i>Intel Demos P5, Sets 1Q93 Intro Date</i>. MPR vol. 6 no. 10, 7/29/92, pg. 4. (<i>Most Significant Bits item.</i>)</li> <li>14: <i>Intel Begins Gradual P5 Unveiling*</i>. Michael Slater, MPR vol. 6 no. 12, 9/16/92, pg. 1. (<i>Cover story.</i>)</li> </ol>   |

- 15: *P5 Christened "Pentium"*. MPR vol. 6 no. 14, 10/28/92, pg. 4. (*Most Significant Bits item.*)
- 16: *Intel Describes P5 Internal Architecture\**. Linley Gwennap, MPR vol. 6 no. 14, 10/28/92, pg. 25. (*Feature article.*)
- 17: *Pentium Falls Short of P5's Promises\**. Michael Slater, MPR vol. 6 no. 15, 11/18/92, pg. 3. (*Editorial.*)
- 18: *Intel Announces New Interrupt Controller*. MPR vol. 6 no. 15, 11/18/92, pg. 5. (*Most Significant Bits item.*)
- 19: *Erratum—P5 Not to Provide 36-Bit Addressing*. MPR vol. 6 no. 16, 12/9/92, pg. 4. (*Most Significant Bits item.*)
- 20: *Intel Demonstrates Pentium Systems*. MPR vol. 6 no. 16, 12/9/92, pg. 4. (*Most Significant Bits item.*)
- 21: *Pentium Has Pins—And Now We Know How Many*. MPR vol. 7 no. 1, 1/25/93, pg. 4. (*Most Significant Bits item.*)
- 22: *Intel Delays Pentium "Announcement"*. MPR vol. 7 no. 2, 2/15/93, pg. 4. (*Most Significant Bits item.*)
- 23: *Pentium Approaches RISC Performance\**. Linley Gwennap, MPR vol. 7 no. 4, 3/29/93, pg. 1. (*Cover story.*)
- 24: *Pentium vs. the RISCs\**. Michael Slater, MPR vol. 7 no. 4, 3/29/93, pg. 3. (*Editorial.*)
- 25: *Intel Reveals Pentium Implementation Details\**. Brian Case, MPR Report vol. 7 no. 4, 3/29/93, pg. 9. (*Feature article.*)
- 26: *Intel provides PCI Chip Set for Pentium\**. Linley Gwennap, MPR vol. 7 no. 4, 3/29/93, pg. 18. (*Feature article.*)
- 27: *Strategic Product Rescheduling\**. John Wharton, MPR vol. 7 no. 4, 3/29/93, pg. 28. (*Oblique Perspective column.*)
- 28: *Pentium Extends 486 Bus to 64 Bits\**. Brian Case, MPR vol. 7 no. 5, 4/19/93, pg. 10. (*Feature article.*)
- 29: *Pentium Chip Sets Poised For Launch\**. Mark Thorson and Michael Slater, MPR vol. 7 no. 5, 4/19/93, pg. 15. (*Feature article.*)
- 30: *Pentium Debuts at \$965—Systems Available*. MPR vol. 7 no. 7, 5/31/93, pg. 4. (*Most Significant Bits item.*)
- 31: *P24T to Ship with "Active" Heat Sink*. MPR vol. 7 no. 8, 6/21/93, pg. 5. (*Most Significant Bits item.*)

- 32: *PC Makers Serve Pentium Before Its Time*. Mike Feibus and Dean McCarron, MPR vol. 7 no. 8, 6/21/93, pg. 18. (Feature article.)
- 33: *Understanding Pentium and PCI*. MPR vol. 7 no. 9, 7/12/93, pg. 28.
- 34: *Cyrix Describes Pentium Competitor*. Linley Gwennap, MPR vol. 7 no. 14, 10/25/93, pg. 1. (Cover story.)
- 35: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (Cover story.)
- 36: *IBM, Intel Revise x86 Pact*. MPR vol. 8 no. 2, 2/14/94, pg. 5. (Most Significant Bits item.)
- 37: *Intel Extends 486, Pentium Families*. Linley Gwennap, MPR vol. 8 no. 3, 3/7/94, pg. 1. (Cover story.)
- 38: *Workstation Market May Implode*. Linley Gwennap, MPR vol. 8 no. 4, 3/28/94, pg. 3. (Feature article.)
- 39: *PPC 604 Powers Past Pentium*. Linley Gwennap, MPR vol. 8 no. 5, 4/18/94, pg. 1. (Cover story.)
- 40: *Intel Offers \$250 Upgrade Chip*. MPR vol. 8 no. 5, 4/18/94, pg. 4.
- 41: *Intel Cuts Back on DX4, Pushes Pentium-60*. MPR vol. 8 no. 6, 5/9/94, pg. 5.
- 42: *Competitors Seek to Counter Pentium Push*. Michael Slater, MPR vol. 8 no. 7, 5/30/94, pg. 10. (Feature article.)
- 43: *Intel Slashes Prices on Pentium, 486 DX2*. Linley Gwennap, MPR vol. 8 no. 9, 7/11/94, pg. 13. (Feature article.)
- 44: *Intel Slashes Pentium, 486 Prices*. MPR vol. 8 no. 14, 10/24/94, pg. 4. (Most Significant Bits item.)
- 45: *Intel to Offer DSP Software for Pentium*. MPR vol. 8 no. 15, 11/14/94, pg. 5. (Most Significant Bits item.)
- 46: *New Pentiums for Notebooks, 486 Upgrades*. Michael Slater, MPR vol. 8 no. 15, 11/14/94, pg. 14. (Feature article.)
- 47: *Intel Fixes Pentium FPU Bug*. MPR vol. 8 no. 16, 12/5/94, pg. 5. (Most Significant Bits item.)
- 48: *Aspects of Cache Memory and Instruction Buffer Performance*. M. D. Hill, U.C. Berkeley, 1987. (Ph.D. dissertation.)

**Other Technical  
References**

- 49: *Optimizing Systems Performance Based on Pentium Processors*. John Novitsky, Mani Azimi, Raheel Ghaznavi, IEEE Press/CompCon 93, 2/93, pg. 63. (Detailed analysis of how cache and main memory system design affect Pentium system performance.)
- 50: *Pentium System Architecture*. MindShare Press.

**Other Periodicals**

- 51: *Branch Prediction Strategies and Branch Target Buffer Design*. J. Lee and A. J. Smith, IEEE Computer, 1/84, pg. 6. (Ph.D. dissertation.)
- 52: *Intel Fixes a Pentium FPU Glitch*. Electronic Engineering Times, no. 822, 11/7/94, pg. 1.
- 53: *Inside Pentium*. Nick Stam, PC Magazine, vol. 12 no. 8, 4/27/93, pg. 123. (Intel chip design, Cyrix sidebar, AMD sidebar, Pipelines, IBM sidebar, RISC competition sidebar.)
- 54: *Pentium Power*. Michael Feibus and Michael Slater, PC Magazine, vol. 12 no. 8, 4/27/93, pg. 108. (Pentium design, competitor sidebars.)
- 55: *Pentium: The 586 You've Been Waiting For?*. Michael Feibus and Michael Slater, PC Magazine, vol. 12 no. 8, 4/27/93, pg. 108. (Cover story about Pentium design and history.)
- 56: *Pentium at Home in PC Market*. Rebecca Smith, San Jose Mercury News, 5/9/94, pg. 1.
- 57: *Will the Pentium Kill the 486?*. Gina Smith, PC Computing, 5/93, pg. 116. (Cover story about Pentium features, design, performance, and system vendors.)
- 58: *Pentium Arrives*. William Gee, Windows Magazine, 6/93, pg. 115. (Cover Story: Performance, Competitors, Software.)
- 59: *80x86 Wars*. Tom Halfhill, Byte, vol. 19 no. 6, 6/94, pg. 74. (Cover Story about Intel and its strongest x86 and RISC competition.)
- 60: *Pentium Poised to Oust 486*. Neal Boudette, PCWeek, vol. 11 no. 26, 7/4/94, pg. 1.

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)





# 13

## NexGen Microprocessors

---

The first non-Intel Pentium-class microprocessor emerged from a source that some thought was unlikely: perennial startup NexGen, which labored for eight years to create its first product, the Nx586 microprocessor.

### 13.1 The NexGen Nx586 Microprocessor

The NexGen Nx586 is a high-performance fifth-generation processor that uses a number of new implementation techniques to deliver Pentium-class performance. The device was announced in 1Q94 and began shipping in mid-year. Table 13-1 summarizes the features and capabilities of the Nx586 device.

The Nx586 is partitioned differently than a conventional 486 or Pentium device. Whereas Pentium folds a complete x86 integer processor, floating-point unit, and 16 kilobytes of combined instruction/data cache onto one chip, NexGen chose instead to relegate the FPU to a separate chip, thereby freeing sufficient die area to double the cache size and include on-chip control logic for a second-level cache.

The optional external FPU for the Nx586 was officially announced in 1Q94 with the designation Nx587, in keeping with the business model and nomenclature of the 386/387. In 3Q94 the Nx587 was “de-announced.” Omitting the FPU from the CPU proper lowers the cost of entry-level systems. Since most programs make little (if any) use of floating-point math,

Product Name	NexGen Nx586
Introduction Date	March 1994
Prognosis	Emerging from a protracted gestation
Device Integration Level	Superpipelined 32-bit IEU, PMMU, separate 16KB instruction and data caches, on-chip control logic for an external L2 cache (Pin-compatible module with onboard FPU planned)
CPU Architecture Level	486 integer instruction set with NexGen SMM (Future versions will support 486 FPU instructions)
Core Technology	NexGen-designed high-performance core
Pinout	Custom
Data Bus Width	64 bits (D63..D0)
Physical Addressability	4GB (Address A31..A3)
Data-Transfer Modes	Information not available
Cache Support	16K bytes each I- and D-cache 4-way set associative Write-through operation only L2 cache controller supports 256KB or 1MB of SRAM for unified I/D cache with copy-back protocol
Floating-Point Support	Currently none; replacement part that includes a high-speed FPU coprocessor planned for 1H95
Operating Voltage	4 V (tolerance not available)
Frequency Options	70-, 75-, 84-, or 93-MHz core operation
Clocking Regime	Information not available
Active Power Dissipation	15 W @ 4.0 V and 93 MHz
Power-Control Features	Static core design, NexGen SMM extensions
Process Technology	0.5-micron 5-layer-metal CMOS
Die Size	14.1 × 14.1 mm (200 mm <sup>2</sup> )
Transistor Count	Integer processor: 3.5 million (Future FPU: approx. 700,000 transistors)
Package Options	463-pin interstitial PGA or multichip module

Table 13-1. NexGen Nx586 CPU feature summary.

these FPU-less systems will be adequate for most users. It also no doubt simplified the design process somewhat; or at least deferred for an extra year the tedious task of trying to get all the FPU algorithms to be perfect.

### Core Design

The Nx586 contains an integer processor, a PMMU, separate 8-Kilobyte instruction and data caches, and a second-level (L2) cache controller. The integer processor in turn contains four largely independent execution units: two integer units, an address unit, and, in time, the optional floating-point unit (see Figure 13-1).

On every clock cycle decode logic can crack a single x86 instruction and translates it into one or more internal instructions, which are later executed by the various execution units. These

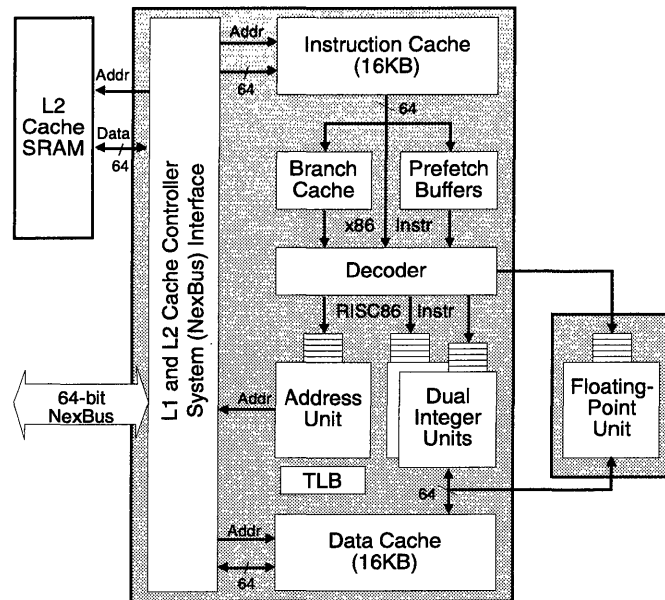


Figure 13-1. NexGen Nx586 microarchitecture.

instructions can be executed speculatively and out of order to increase the potential for parallelism.

Instructions sent to the execution units are predecoded to simplify processing within each execution unit. These instructions are represented internally by a word approximately 100 bits wide. (These wide-word instructions are never stored in memory or transferred to other chips, so their large size does not cause significant problems.)

NexGen uses the label “RISC86” to describe the internal format of these internal instruction word, since (according to NexGen) they correspond to the types of operations performed by a conventional RISC processor. Conventional RISC architectures, however, must typically pack an operation code, three operand fields, and possibly a few bits of constant into a single 32-bit memory word. It would be just as appropriate (but perhaps somewhat less trendy) to refer to these words as “microcode.”

Many x86 instructions convert directly to a single RISC86 instruction. All register-to-register ALU operations, for example, have RISC86 counterparts. Because RISC86 uses a load-store model, however, many x86 instructions that access memory require two or three RISC86 instructions. For example, the x86 instruction:

```
ADD     mem, CX
```

translates into three RISC86 instructions:

```
LOAD    R2, R1
ADD     R2, R3
STORE  R1, R2
```

where the physical register numbers are assigned by the decoder to map the x86 registers appropriately.

Iterative x86 string instructions translate into an indefinitely long sequence of internal instructions. The decoder issues RISC86 instructions as fast as possible; when the core detects the termination condition, remaining iterations are invalidated.

The Nx586 goes to great lengths to maintain this rate of one instruction per cycle, regardless of data dependencies, cache misses, branches, resource conflicts, and other events that cause glitches in most other processors.

To further exploit the parallelism of the RISC86 instructions, a 14-entry queue precedes each of the function units. If RISC86 instructions cannot immediately execute due to dependencies or resource conflicts, they simply wait in the queues; other instructions in the queues continue to execute in other function units.

The queues prevent the instruction decoder from stalling when a function unit is busy, as it can simply issue RISC86 instructions into the queues. If the instruction at the front of a queue cannot be executed for any reason, however, that function unit stalls. This problem will completely tie up one unit while the others continue. The dual integer units provide some redundancy; if one stalls, the other can continue processing (non dependent) integer operations.

The Nx586 allows up to 14 RISC86 instructions to be pending at any one time; NexGen says that there are often 8 or more instructions in process, and that it is not unusual to reach the limit of 14. One effect of the queues is that instructions can be executed out of order, though they are always issued and retired in order. The CPU tags each instruction with a sequence number. The tags help ensure that instructions with dependencies are executed in the proper order. The use of register renaming reduces the number of dependencies, increasing parallelism in the instruction stream.

Like most microprocessors with multiple pipelines, the Nx586 execution units are not symmetrical. The primary unit can handle all RISC86 integer operations, including multiply and divide, while the second integer unit performs only simple (single-cycle) operations. The decoder has a load-balancing algorithm to allocate instructions that could be sent to either integer unit.

RISC86 load and store instructions are routed to the address unit, which calculates the target address and performs translation and validation according to the x86 standard. Since there is a single address unit, only one RISC86 load or store instruction can be executed on each cycle. The chip contains a 32-entry unified TLB for virtual address translation.

The Nx586 uses an instruction prefetch buffer to solve the problems of variable length and alignment inherent in x86 code. Unlike Pentium's cache, which contains special logic to fetch up to 31 consecutive unaligned bytes, the Nx586 instruction cache delivers instructions in groups of 8 aligned bytes. The prefetch buffer holds up to three groups of 24 bytes each, prefetching along the sequential path and two predicted paths. On each cycle, the Nx586 decoder can fetch up to 8 unaligned bytes from the prefetch buffer. It can also fetch directly from the cache but is restricted to aligned accesses of 8 bytes.

One of the bottlenecks of the x86 architecture is that it defines only eight general-purpose registers. The Nx586 uses register renaming to ameliorate this bottleneck, and implements 22 physical registers. At any given time eight of these are mapped onto the eight logical registers of the x86 architecture. This technique, called register renaming, can circumvent register conflicts common in x86 programs. Register renaming also used in the Cyrix "M1," AMD "K5," and Intel "P6."

Proper handling of exceptions can be complex in an out-of-order machine. The Nx586 always retires instructions in order, even if their results were generated out of order. Exceptions are handled when the excepting instruction is retired; the results of all successive (unretired) instructions are nullified. Register renaming simplifies this process. Values in the physical registers are not overwritten until the instruction that generated them is retired; intermediate values are kept in other physical registers. Nullifying instructions is simply a matter of updating the register mapping.

Although out-of-order execution does not require additional overhead for the general-purpose registers, the Nx586 must keep multiple copies of special registers, such as the flags and segment registers, to correctly handle exceptions.

Memory writes are queued in an eight-entry write-reservation station and are not executed until the write instruction is retired, ensuring that the cache/memory system always sees in-order, nonspeculative writes. Reads can take data directly from the reservation station, bypassing the first- and second-level caches.

### **Is it Superscalar Yet?**

The unusual design of the Nx586 core makes the part difficult to categorize. Because one x86 instruction can turn into two or more RISC86 instructions, the decoder can issue multiple RISC86 instructions per cycle, one to each function unit. The function units are designed to work in parallel, at times executing multiple RISC86 instructions in a single cycle.

NexGen describes the Nx586 design as “superscalar,” pointing to the fact that all four execution units may sometimes be executing different RISC86/microcode operations. A more conventional definition of the word “superscalar,” however, relates to a processor’s ability to fetch, decode, issue, execute, and retire more than one instruction during every clock cycle. At the x86 macroinstruction level, the Nx586 can actually decode and issue, at most, one x86 instruction per cycle, so the part falls short of this definition.

From the standpoint of x86 instructions, then, the Nx586 may best be described as a scalar processor with a very deep pipeline (superpipelined, if you will), and FIFO “slip-joints” between the stages. In NexGen’s defense, the Nx586 design does appear to be able to sustain execution rates very close to one new x86 macroinstruction nearly every cycle (one IPC)—approximately the same as a dual-pipeline superscalar Pentium device.

### **Cache Logic**

The Nx586 contains 16 Kilobytes each of instruction and data cache, twice the collective size of Pentium’s combined cache. Each of NexGen’s caches are four-way set-associative, further increasing the hit rate compared with Pentium’s two-way caches. Each is physically indexed and tagged.

Two cache accesses can occur during each cycle. Because the Nx586 has only one address unit, the second cache access is used for snooping or for moving data to and from the L2 cache or the system bus. Many processors block the cache when these

events occur, stalling CPU accesses, but the Nx586 can handle them without slowing instruction execution.

The built-in L2 cache controller connects to an external L2 cache constructed of standard asynchronous SRAMs. Only two configurations are supported: 256Kbytes or 1Mbyte, both using eight  $\times 8$  SRAMs. The L2 cache is unified (instructions and data) and, like the L1 caches, is four-way set-associative. The controller allows two cycles to access the external cache, requiring 15-ns SRAMs at 70 or 75 MHz, and 12-ns SRAMs at 84 and 93 MHz.

The tags are stored in the same chips as the cache data, reducing the amount of memory available for data by 6%. A cache access requires two cycles to read the tags, then two cycles to read each quad word of data (4-2-2-2 access timing pattern). Reading the tags in series with the data simplifies the implementation of a set-associative cache, since the correct set is determined before the data is read; otherwise, the chip would have to support a 256-bit SRAM interface to read all four sets at once.

NexGen's set-associative design should have a higher hit rate than a direct-mapped cache of the same size for a Pentium chip. Another advantage of the NexGen design is that it can maintain the same access pattern at higher frequencies because the cache bus is clocked at a different speed than the system bus.

The on-chip caches use a write-through protocol, taking advantage of the direct path to the L2 cache. The external cache uses a write-back design to reduce traffic on the system bus. Writes are sent to both the data and instruction caches to support self-modifying code.

According to NexGen a 93-MHz Nx586 should perform about as well as a 100-MHz Pentium with a 66-MHz system bus. The Pentium device would likely transfer a new word of data only every other bus cycle, equivalent to every third CPU cycle, whereas the Nx586 can perform cache transfers every to the CPU clock. Even so, the NexGen part in this example would require 12-nsec SRAMs, whereas the Pentium could get by with 15-nsec parts.

### Execution Timing

It's difficult to pin down exactly how long it takes the Nx586 to execute an instruction. The decoder can issue nearly all instructions in a single cycle, but execution may be delayed, depending

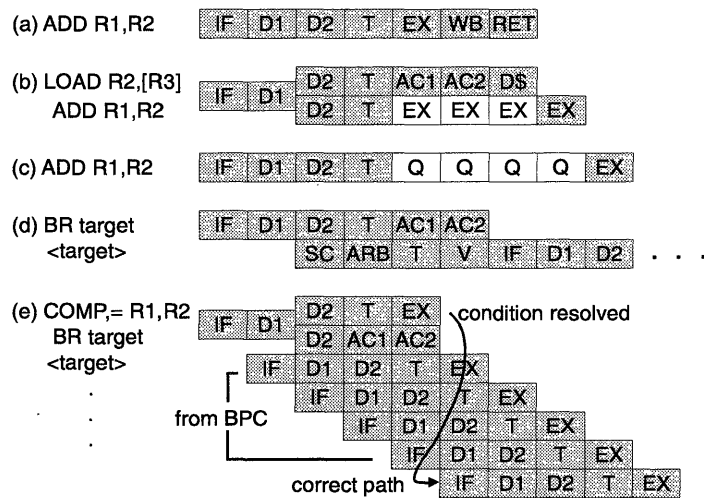


Figure 13-2. NexGen Nx586 execution pipeline timing.

on interactions in the RISC86 core. Even basic pipeline concepts are difficult to apply to this design.

Figure 13-2 shows the execution timing of several types of instructions. The first few pipeline stages are the same for most instructions. During the first stage, an instruction is fetched from either the instruction prefetch buffer or the instruction cache. This stage will stall for two cycles if the prefetch buffer is empty and the requested instruction stretches across an eight-byte boundary, but this situation occurs infrequently.

Once the x86 instruction is fetched, it takes two cycles to decode and translate it into a RISC86 instruction sequence. A third cycle allows these instructions to transit to the function units. This is mainly a vestige of NexGen's original eight-chip design.

Here things start to get sticky. The simplest case is a register-to-register integer calculation (see line a of Figure 13-2). Assuming that the queues are empty, it can be executed in a single cycle and retired in two cycles. The scoreboard and the register map are updated on the final cycle.

Memory-to-register calculations are translated into two RISC86 instructions (see Figure 13-2 line b):

```
LOAD    R2, R3
ADD     R1, R2
```



The LOAD is sent to the address unit, while the ADD goes to one of the integer units. Again assuming that the queues are empty, the LOAD begins processing immediately, but the ADD stalls until the LOAD completes. This stall ties up one integer unit, but the other integer unit (and the FPU) could process subsequent instructions during that period. The LOAD itself takes three cycles: two to generate, verify, and translate the address, and one to access the data cache.

In practice, however, several RISC86 instructions usually are queued at any given time. In this situation, one or more delay cycles may be inserted into the execution of a particular RISC86 instruction (see Figure 13-2 line c). Because the core can execute multiple instructions per cycle, these delays usually are not reflected in the apparent execution of the x86 instruction stream.

### **Branch Prediction**

When the Nx586 encounters a branch, it predicts the outcome and begins to execute subsequent instructions. This is called speculative execution, since these instructions may be incorrect if the branch condition is mispredicted. The Nx586 can speculatively execute beyond two predicted branches; in most cases, the first branch condition will be resolved by the time a third branch is encountered.

To reduce taken-branch penalties, NexGen has implemented a 96-entry branch prediction cache (BPC). The company has not released details on the structure of the Nx586 BPC. However, NexGen has received a patent (number 5,230,068) that describes a BPC that contains the first 24 instruction bytes at each target address, along with the target address itself. This design is similar to the branch target cache in AMD's 29000 but is different from Pentium's, which contains only target addresses.

The BPC described in the patent is indexed by the address of the branch instruction, so it could be checked during the IF stage. Twenty-four instruction bytes would be enough to bridge the gap until the instruction cache begins responding, even for most misses to the L2 cache.

Line d of Figure 13-2 shows a branch predicted to be taken. By the end of the D1 phase, the target address has been calculated from the instruction. This address is then used to start an instruction fetch by assuming that the target is on the same virtual page as the previous address. The virtual target address is translated by the address unit in parallel with the fetch, and

the fetch is restarted if the translation indicates that the target is on a different page.

In the meantime, it takes four cycles to transmit the address to the instruction cache and begin receiving data. This seems absurdly long for an on-chip access, but most of these cycles are a legacy from the old multichip design; an extra cycle is also required to update the scoreboard. In total, there are five cycles during which sequential x86 instructions could have been decoded and issued and many more RISC86 instructions could have been executed; these instructions must all be invalidated.

Line e of Figure 13-2 shows the pipeline timing of a conditional branch instruction, divided into a compare instruction followed by a branch. As the branch itself is handled by the BPC as described above, the compare is dispatched to one of the integer units for evaluation. If the queues are empty, as in the figure, five cycles are lost if the result of the compare indicates that the branch was mispredicted. If the queues are not empty, 19 or more cycles can be lost before the misprediction is detected. These penalties give the Nx586 the appearance of a very deep pipeline.

The Nx586 uses the same two-bit Smith and Lee algorithm used by Pentium to predict branches. According to the patent, each BPC entry contains two prediction bits. If a branch misses the BPC, an additional 2,048-entry, two-bit-wide branch history table is checked, increasing the prediction accuracy over Pentium's 256-entry branch target buffer.

Although these two structures will correctly predict most conditional branches, they are less effective for RET instructions. Returns are hard to handle because the target address can change on each iteration. The Nx586 includes a return address stack, NexGen claims, that handles up to eight subroutine calls. The combination of these three structures should push the prediction success rate above 90% on most code, compensating for the significant penalties that can occur when the Nx586 mispredicts a branch.

### **Floating-Point Unit**

An upcoming version of the Nx586 will include an FPU chip to handle all floating-point operations. The FPU will receive instructions from the decoder at the same time and in much the same way as the other three function units. The FPU should execute double-precision adds and multiplies in just two cycles, one fewer than Pentium; Table 13-2 shows the latencies for various math operations.

		Intel Pentium	NexGen Nx586 with FPU
FP Add (DP)	latency	3	2
	throughput	1	2
FP Multiply (DP)	latency	3	2
	throughput	2	2
FP Divide (DP)	latency	39	40
	throughput	39	40

Table 13-2. NexGen FPU instruction execution times.

Current plans are for the existing integer processor to be combined with an FPU die within a pin-compatible multichip module to be introduced in 1H95. End users will be able to upgrade existing systems for support floating-point operations by removing the original integer processor from its socket and inserting a two-chip module. One disadvantage to this approach is that expanding an Nx586 system to include an FPU requires removing the CPU from its socket and discarding it (or perhaps returning it to a vendor for credit).

Even though NexGen's plans now call for combining the two die onto a single multichip module, the module still includes a hundred or so "no connect" pins that previously been reserved for the FPU interface. As a result, the integer-only device and the upcoming multichip module each require a 463-pin PGA that likely costs twice as much as a Pentium 296-pin PGA.

NexGen did not implement Pentium's parallel FXCH feature and thus chose not to pipeline the FPU, since most code requires an FXCH between each math operation. Pentium's ability to issue an FXCH along with a math operation may balance out the performance advantage of the faster adds and multiplies.

### System Interface

NexGen chose an unusual partitioning of the system-bus design as well. Instead of a standard 486 or Pentium bus, the Nx586 connects to the system via its own proprietary 64-bit NexBus. NexGen currently offers a single product which connects to the main-memory, ISA, and VL-buses. An external 82C206 is required for standard system logic such as interrupts and timers. The company is developing a second device that provides a PCI interface and integrates the 82C206 functions.

A second dedicated 64-bit bus connects to an external SRAM cache. At the chip level, a third dedicated 64-bit bus connects the device to the external FPU, as shown in Figure 13-3.

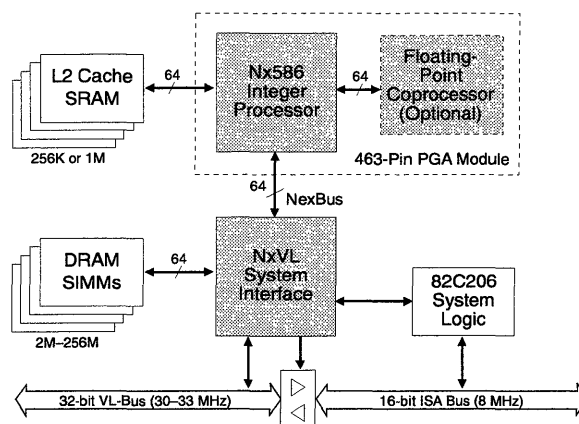


Figure 13-3. NexGen Nx586 system partitioning.

The dedicated cache bus eliminates bus conflicts with memory and I/O traffic and, in future versions, will allow the cache bus to run at the CPU frequency while the system bus stays at a more reasonable speed.

All caches and the write buffer maintain coherency with other data in the system, using a MESI cache-coherency protocol. (Chapter 12 contains an introduction to the MESI standard.) This protocol allows other caches (typically other processors) to coexist in the system. The Nx586 snoops all transactions on the NexBus; if a read snoop hits in any of its caches, it aborts the bus transaction and writes the dirty data back to main memory. Because of the double-speed L1 caches, most snoop transactions are transparent to the processor.

**Performance** NexGen says the Nx586 can deliver about 7% more performance than Pentium on integer code at the same clock frequency, so a 93-MHz Nx586 is purportedly comparable in throughput to a 100-MHz Pentium. At this writing, NexGen has not released SPEC benchmark ratings for its parts.

The NexGen processor should have somewhat better cache performance than Pentium due to its larger on-chip caches and set-associative L2 cache, compensating for the extra penalty cycle on L2 accesses. The NexBus interface adds overhead to memory accesses, however, so Pentium may hold the edge on programs with inherently high cache-miss rates. In the absence of independent benchmark data, it appears that the two chips could well offer similar performance on many applications.

NexGen hopes to further extend the Nx586 family with a future “686” design intended to deliver two to four times the performance of the Nx586. Although the company has not discussed details of this future product, the Nx586 architecture could be extended with a second decode unit and additional function units.

### Vital Statistics

The Nx586 is currently manufactured for NexGen by IBM in its five-layer-metal 0.5-micron CMOS process. The device measures  $14.1 \times 14.1$  mm (200 mm<sup>2</sup>), halfway between Intel’s 0.8-micron and 0.6-micron Pentia. Yet another redesign will (NexGen hopes) shrink the layout to an area smaller than the 0.6-micron Pentium, but even if the two products had similar die areas, the Nx586’s 0.5-micron process technology would likely make the Nx586 more expensive to build. The Nx586 also uses a more costly package.

## 13.2 Commentary

NexGen originally planned to build high-performance multiprocessor systems but abandoned this effort to focus on completing and marketing its processor chip set. Without shipping a product, the company has raised \$90 million from a long list of backers including ASCII Corp., Compaq, Olivetti, and noted venture capital firm Kleiner, Perkins, Caufield, and Byers.

NexGen and its investors should be congratulated for persevering on the long road to shipment of its first product and for delivering that product at a competitive price/performance point (see Table 13-3). The company’s initial goal, however, had been to deliver x86 performance two or more times greater than Intel parts. These goals no doubt faded with time; even by NexGen’s evaluation, its fastest 93-MHz parts are currently no faster than the 100-MHz Pentia that Intel has been shipping in volume for many months.

(A NexGen spokesman claims that some of the press reports cited in Table 13-3 were erroneous to begin with. “So what if ‘press reports’ said the company planned something that didn’t happen,” he countered. “Press reports’ *also* claim *aliens abduct humans!*”)

To establish itself in the market NexGen must meet certain criteria. Like all new x86 vendors, the company must first demonstrate uncompromising compatibility with x86 software. One

Date	Event or Announcement
1986	NexGen founded.
1988	Company begins development of superscalar multichip x86 processors for use in systems intended to reach the market in 1989. Nick Tredennick, company cofounder and contributor to this Report, resigns.
4/89	NexGen reveals plans to build 386-compatible processor for internal use only, with no plans to sell chips to other system vendors. NexGen design to use seven custom chips plus standard SRAM cache. Costs likely to exceed 486-based systems; company to depend on higher performance than Intel-based systems. Design delays postpone shipments to 1990.
3/90	Compaq invests in NexGen. NexGen promises to deliver double the performance of a 486. NexGen says it has always planned to work with system partners. Early investor Olivetti, which helped with system-level design and had planned to market a chip-set based system, adopts 486 due to NexGen's scheduling delays.
10/90	Thampy Thomas, company president, reveals a few details of its 386/486 architecture multichip processor at Microprocessor Forum. He claims the last chip is already in fab, so company soon expects to know how well its years of effort—and tens of millions of dollars—have paid off.
1/91	Atiq Raza appointed CEO.
10/91	8-chip design in 1.2 $\mu$ CMOS validates RISC86 microarchitecture.
3/92	Design work begins on a three-chip implementation to use 0.5 $\mu$ technology.
4/92	Three-chip design repartitioned to require just two chips.
5/92	NexGen exhausts initial capital and receives \$1.75 million bridge loan from Kleiner Perkins, ASCII Corp., and Olivetti. Seeks to raise additional \$15–\$30 million in private-placement offering. System shipments planned for second half of 1992, to be priced from \$7,000. Prototype successfully tested for compatibility by an independent testing lab, VeriTest. Runs various DOS and protected-mode Windows 3.0 applications. While the company will initially focus on selling complete systems based on the 8-chip set, the 3-chip set will be sold to other system vendors. NexGen expects the initial 8-chip set to run at 33 MHz and deliver 25 SPECmarks, twice that of a 486 at the same clock rate. Three-chip version expected to deliver 60 SPECmarks at 66 MHz.
8/92	Press reports say NexGen planning to market two-chip design in Japan through a joint venture with investor ASCII Corp. The company closes a new round of financing from private investors. Total investment in the company is rumored to approach \$50 million.
10/92	Eight-chip system implementation efforts discontinued.
10/93	Taiwan sources report samples of NexGen's 586 microprocessor have been delivered to computer makers there. NexGen declines comment. NexGen has, to date, spent five years and \$60 million developing its x86-compatible processor. Design is said to be bug-free.
3/94	NexGen delivers samples of Nx586 chips built by IBM in five-layer-metal 0.5 $\mu$ CMOS process. The Nx586 is predicted to ship in 2Q94 with the Nx587 FPU version scheduled to ship by 2Q95.
6/94	IBM contracts to build Nx586 and FPU chips on 0.5 $\mu$ fab line.
8/94	System repartitioned to combine IEU and FPU chips into a single multichip module. Nx587 discontinued as a separate product. New product-numbering scheme introduced in which parts are given a numeric suffix 7% higher than actual maximum rated frequency, to indicate expected system performance levels compared to Pentium. Chip redesign initiated to further reduce die size.

Table 13-3. NexGen announcement chronology.

advantage of the product's long development cycle is that the company has spent literally years testing its design with a wide variety of applications; it claims that the current version is fully compatible, but only time will tell. Cyrix and IBM have shown that independently designed new chips can be compatible with Intel's, priming the market for other competitors.

NexGen must also demonstrate an ability to deliver an adequate supply of parts. As Intel floods the market with tens of millions of Pentium chips per year, system vendors' unit demands will increase, raising the bar for NexGen.

Finally, the startup must weather the inevitable legal challenges from Intel. Since NexGen uses IBM as a fab, it can deploy the same patent-laundering defense that Cyrix has used. But even if NexGen were unable to hide behind IBM's patent portfolio, the company says it could still sell the Nx586 because its design and microcode were developed independently and do not (NexGen says) violate any Intel patents. Maybe so; the company's reluctance to discuss the Nx586's address-translation mechanism indicates some nervousness over the notorious '338 patent and other memory-management issues.

At this writing, the issues of compatibility and the availability of the PCI interface must still be resolved. Even then, NexGen must maintain a price/performance advantage over Intel. The lack of an on-chip FPU and the higher manufacturing cost of NexGen's chip will put it at a disadvantage if Intel continues to cut Pentium prices aggressively. If Intel falters, however, NexGen will be first in line to fill the gap.

## 13.3 For More Information...

Additional technical information on NexGen product plans may be found in the following publications:

- |                                       |  |
|---------------------------------------|--|
| <b>Vendor Publications</b>            | 1: <i>NexGen Nx586 Family Enters Volume Production With P100, P90, P80, and P75 Processors Press Kit.</i> NexGen, 9/19/94. |
| <b>Microprocessor Report Articles</b> | 2: <i>NexGen Prepares to Launch Systems.</i> MPR vol. 2 no. 7, 7/88, pg. 2. ( <i>Most Significant Bits item.</i> )         |
|                                       | 3: <i>NexGen Aims to Beat 486 Performance*.</i> MPR vol. 3 no. 4, 4/89, pg. 6. ( <i>Feature article.</i> )                 |

- 4: *Compaq Investment in NexGen Revealed*. MPR vol. 4 no. 5, 3/21/90, pg. 5. (*Most Significant Bits item.*)
  - 5: *It Doesn't Have to be RISC to be Good*. Thampy Thomas, MPR vol. 4 no. 11, 6/20/90, pg. 3. (*Viewpoint.*)
  - 6: *NexGen Presents Superscalar 386 Approach\**. MPR vol. 4 no. 20, 11/7/90, pg. 6. (*Feature article.*)
  - 7: *NexGen Seeking New Funding*. MPR vol. 6 no. 7, 5/27/92, pg. 4. (*Most Significant Bits item.*)
  - 8: *NexGen Developing Two-Chip P5 Competitor*. MPR vol. 6 no. 11, 8/19/92, pg. 5. (*Most Significant Bits item.*)
  - 9: *NexGen Quietly Samples 586, At Last*. MPR vol. 7 no. 14, 10/25/93, pg. 4.
  - 10: *PC Market Centers on Growing 486 Family*. Michael Slater, MPR vol. 8 no. 1, 1/24/94, pg. 1. (*Cover story.*)
  - 11: *NexGen Enters Market with 66-MHz Nx586*. Linley Gwennap, MPR vol. 8 no. 4, 3/28/94, pg. 12. (*Feature article.*)
  - 12: *NexGen, IBM Finally Come to Terms*. MPR vol. 8 no. 8, 6/20/94, pg. 5. (*Most Significant Bits item.*)
  - 13: *NexGen Pushes 586 to 93 MHz*. MPR vol. 8 no. 13, 10/3/94, pg. 4. (*Most Significant Bits item.*)
- Other Periodicals**
- 14: *80x86 Wars*. Tom Halfhill, Byte, vol. 19 no. 6, 6/94, pg. 74. (*Cover Story about Intel and its strongest x86 and RISC competition.*)

(\*Note: Items marked with an asterisk are available in *Understanding x86 Microprocessors*, a collection of article reprints from *Microprocessor Report*.)



## REQUEST FOR MORE INFORMATION

# MicroDesign Resources Technical Library

Just check the appropriate boxes below, fill in the bottom of the form, and fax to 707.823.0504. You'll receive detailed information on the following Technical Library reports:

**THE COMPLETE x86—The Definitive Guide to 386, 486, and Pentium-Class Microprocessors**

The resource for the most up-to-date, detailed information and analysis on every 386, 486, and Pentium-class chip made today. Edited by John Wharton, this 750+ page report outlines the history of x86, profiles the manufacturers, provides chip-by-chip reviews and comparisons, predicts future plans, and puts it all in perspective. PRICE \$2,695.00\*

**RISC ON THE DESKTOP—A Comprehensive Analysis of RISC Microprocessors for PCs, Workstations, and Servers**

This report addresses the technological issues for each processor in the context of the competition. You'll learn about the market strategy behind each architecture, and the manufacturers' prospects. You'll get a complete review of each chip's microarchitecture, cache design, system interface, and performance. Edited by Linley Gwennap of *Microprocessor Report*. PRICE \$2,695.00\*

**NEW DRAM TECHNOLOGIES—A Comprehensive Analysis of the New Architectures**

This one-of-a-kind resource is designed to give you a giant head start in your analysis when choosing between wide DRAMs, extended data-out DRAMs, enhanced DRAMs, cached DRAMs, and Rambus DRAMs. Written by Steven Przybylski, Ph.D., the report delivers an expert analysis of system-level implications, side-by-side architecture comparisons and a historical perspective. PRICE \$2,695.00\*

**HIGH-PERFORMANCE EMBEDDED MICROPROCESSORS**

Embedded microprocessors are the most pervasive, highest-volume CPUs in the world today, and the fastest-growing part of the microprocessor market. This report covers all the high-performance, 32-bit embedded microprocessors from the top down, and from every angle. Edited by James L. Turley of *Microprocessor Report*, it also includes an analysis of the features that mean success or defeat in the competitive embedded arena. PRICE TBD

**BUYERS GUIDE TO DSP PROCESSORS**

This guide provides key insights into each processor's strengths and weaknesses, as well as complete sets of tables to directly compare sets of processors for particular features or performance metrics. Written by DSP consultants, Berkeley Design Technology, this report evaluates processor performance based on BDT's own benchmarks. PRICE \$2,450.00\*\*

\*Discounts available to *Microprocessor Report* subscribers; substantial discounts for additional copies to same site.

\*\*Additional copy discount available.

Please send the brochures on the reports I've checked to:

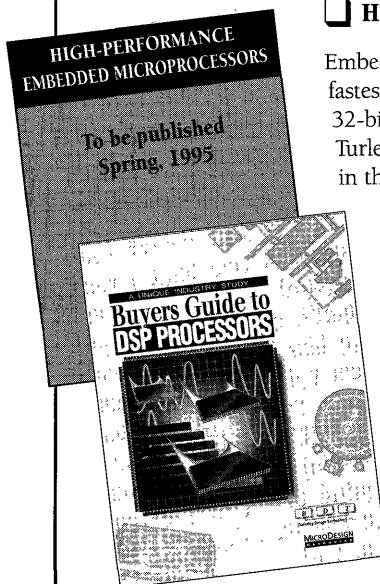
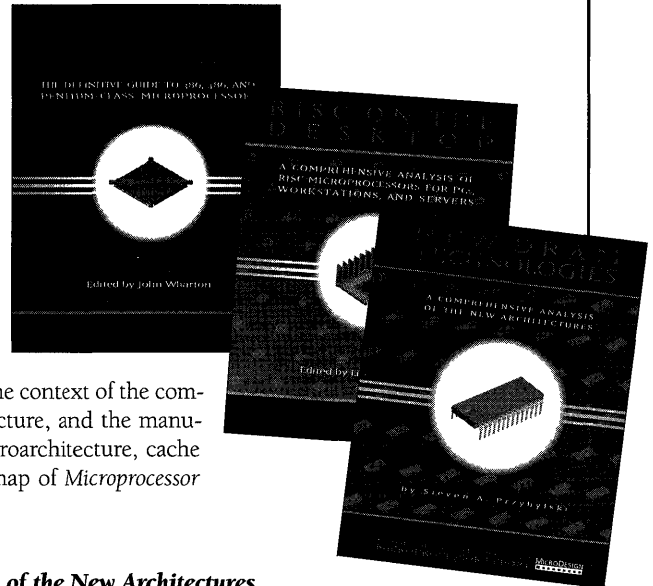
Name \_\_\_\_\_ Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone ( ) \_\_\_\_\_ Fax ( ) \_\_\_\_\_ Email \_\_\_\_\_



**MICRODESIGN  
RESOURCES**

874 Gravenstein Hwy. So.  
Sebastopol, CA 95472  
707.824.4001

FAX THIS FORM TO: 707.823.0504



## REQUEST FOR MORE INFORMATION

# MicroDesign Resources Technical Library

Just check the appropriate boxes below, fill in the bottom of the form, and fax to 707.823.0504. You'll receive detailed information on the following Technical Library reports:

**THE COMPLETE x86—The Definitive Guide to 386, 486, and Pentium-Class Microprocessors**

The resource for the most up-to-date, detailed information and analysis on every 386, 486, and Pentium-class chip made today. Edited by John Wharton, this 750+ page report outlines the history of x86, profiles the manufacturers, provides chip-by-chip reviews and comparisons, predicts future plans, and puts it all in perspective. PRICE \$2,695.00\*

**RISC ON THE DESKTOP—A Comprehensive Analysis of RISC Microprocessors for PCs, Workstations, and Servers**

This report addresses the technological issues for each processor in the context of the competition. You'll learn about the market strategy behind each architecture, and the manufacturers' prospects. You'll get a complete review of each chip's microarchitecture, cache design, system interface, and performance. Edited by Linley Gwennap of *Microprocessor Report*. PRICE \$2,695.00\*

**NEW DRAM TECHNOLOGIES—A Comprehensive Analysis of the New Architectures**

This one-of-a-kind resource is designed to give you a giant head start in your analysis when choosing between wide DRAMs, extended data-out DRAMs, enhanced DRAMs, cached DRAMs, and Rambus DRAMs. Written by Steven Przybylski, Ph.D., the report delivers an expert analysis of system-level implications, side-by-side architecture comparisons and a historical perspective. PRICE \$2,695.00\*

**HIGH-PERFORMANCE EMBEDDED MICROPROCESSORS**

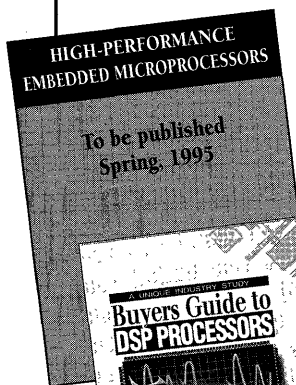
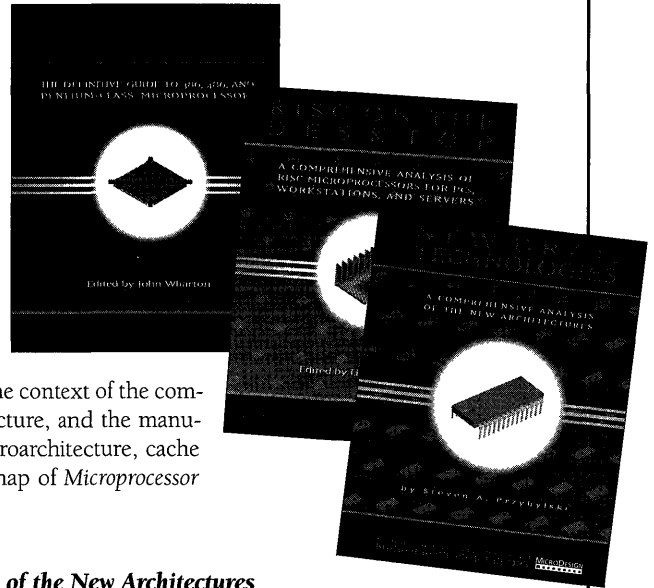
Embedded microprocessors are the most pervasive, highest-volume CPUs in the world today, and the fastest-growing part of the microprocessor market. This report covers all the high-performance, 32-bit embedded microprocessors from the top down, and from every angle. Edited by James L. Turley of *Microprocessor Report*, it also includes an analysis of the features that mean success or defeat in the competitive embedded arena. PRICE TBD

**BUYERS GUIDE TO DSP PROCESSORS**

This guide provides key insights into each processor's strengths and weaknesses, as well as complete sets of tables to directly compare sets of processors for particular features or performance metrics. Written by DSP consultants, Berkeley Design Technology, this report evaluates processor performance based on BDT's own benchmarks. PRICE \$2,450.00\*\*

\*Discounts available to *Microprocessor Report* subscribers; substantial discounts for additional copies to same site.

\*\*Additional copy discount available.



**MICRODESIGN**  
RESOURCES

874 Gravenstein Hwy. So.  
Sebastopol, CA 95472  
707.824.4001

Please send the brochures on the reports I've checked to:

Name \_\_\_\_\_ Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone ( ) \_\_\_\_\_ Fax ( ) \_\_\_\_\_ Email \_\_\_\_\_

FAX THIS FORM TO: 707.823.0504



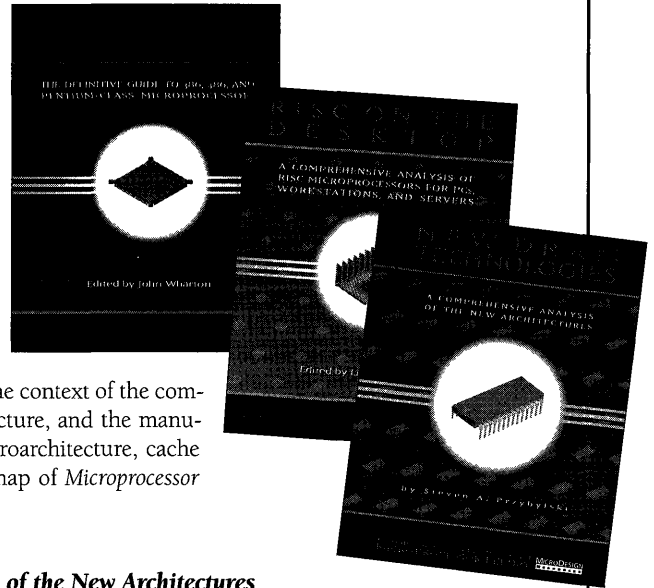
## REQUEST FOR MORE INFORMATION

# MicroDesign Resources Technical Library

Just check the appropriate boxes below, fill in the bottom of the form, and fax to 707.823.0504. You'll receive detailed information on the following Technical Library reports:

**THE COMPLETE x86—The Definitive Guide to 386, 486, and Pentium-Class Microprocessors**

The resource for the most up-to-date, detailed information and analysis on every 386, 486, and Pentium-class chip made today. Edited by John Wharton, this 750+ page report outlines the history of x86, profiles the manufacturers, provides chip-by-chip reviews and comparisons, predicts future plans, and puts it all in perspective. PRICE \$2,695.00\*



**RISC ON THE DESKTOP—A Comprehensive Analysis of RISC Microprocessors for PCs, Workstations, and Servers**

This report addresses the technological issues for each processor in the context of the competition. You'll learn about the market strategy behind each architecture, and the manufacturers' prospects. You'll get a complete review of each chip's microarchitecture, cache design, system interface, and performance. Edited by Linley Gwennap of *Microprocessor Report*. PRICE \$2,695.00\*

**NEW DRAM TECHNOLOGIES—A Comprehensive Analysis of the New Architectures**

This one-of-a-kind resource is designed to give you a giant head start in your analysis when choosing between wide DRAMs, extended data-out DRAMs, enhanced DRAMs, cached DRAMs, and Rambus DRAMs. Written by Steven Przybylski, Ph.D., the report delivers an expert analysis of system-level implications, side-by-side architecture comparisons and a historical perspective. PRICE \$2,695.00\*

**HIGH-PERFORMANCE EMBEDDED MICROPROCESSORS**

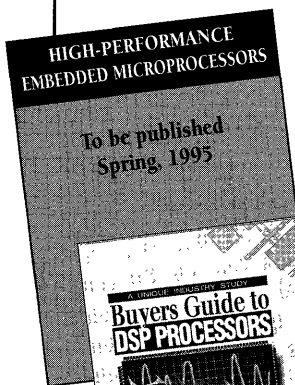
Embedded microprocessors are the most pervasive, highest-volume CPUs in the world today, and the fastest-growing part of the microprocessor market. This report covers all the high-performance, 32-bit embedded microprocessors from the top down, and from every angle. Edited by James L. Turley of *Microprocessor Report*, it also includes an analysis of the features that mean success or defeat in the competitive embedded arena. PRICE TBD

**BUYERS GUIDE TO DSP PROCESSORS**

This guide provides key insights into each processor's strengths and weaknesses, as well as complete sets of tables to directly compare sets of processors for particular features or performance metrics. Written by DSP consultants, Berkeley Design Technology, this report evaluates processor performance based on BDT's own benchmarks. PRICE \$2,450.00\*\*

\*Discounts available to *Microprocessor Report* subscribers; substantial discounts for additional copies to same site.

\*\*Additional copy discount available.



**MICRODESIGN  
RESOURCES**

874 Gravenstein Hwy. So.  
Sebastopol, CA 95472  
707.824.4001

Please send the brochures on the reports I've checked to:

Name \_\_\_\_\_ Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone ( ) \_\_\_\_\_ Fax ( ) \_\_\_\_\_ Email \_\_\_\_\_

FAX THIS FORM TO: 707.823.0504

