BIT-SLICE

DAVID STEINGART and RODNAY ZAKS

SYBEX

EUROPE:
313 RUE LECOURBE, 75015 - PARIS, FRANCE
TEL: (1) 828-2502 TELEX 200858

U.S.A.:
510 GRIZZLY PEAK BLVD., BERKELEY, CALIFORNIA 94708
TEL: (415) 526-2748 TELEX 336311

2nd edition

SYBEX

CONTENTS

SYBEX

0. INTRODUCTION

INTRODUCTION

THE PURPOSES OF THIS COURSE :

1. TO EXPLAIN :

   WHAT BIT - SLICE DEVICES DO

   WHY BIT - SLICE DEVICES EXIST

2. TO DEMONSTRATE IN DETAIL THE PROCEDURE FOR DESIGNING WITH BIT - SLICE DEVICES

3. TO SURVEY THE BIT - SLICE DEVICES ON THE MARKET

4. TO SURVEY THE RANGE OF APPLICATIONS OF BIT - SLICE DEVICES.

BASIC CONCEPTS

THIS COURSE EXPLORES TOPICS IN THESE AREAS :

1. CENTRAL PROCESSOR ARCHITECTURES
2. PROGRAMMING AND MICROPROGRAMMING
3. LARGE SCALE INTEGRATION (LSI)
4. BIT - SLICE ARCHITECTURE
5. PROGRAMMED LOGIC ARRAYS (PLA'S)
6. COMPUTER ARITHMETIC
7. EMULATION
8. SPECIAL FUNCTION PROCESSOR ARCHITECTURES

DEFINITIONS

CENTRAL PROCESSOR : THE PART OF A COMPUTER THAT TRANSFORMS DATA, MAKES DECISIONS.

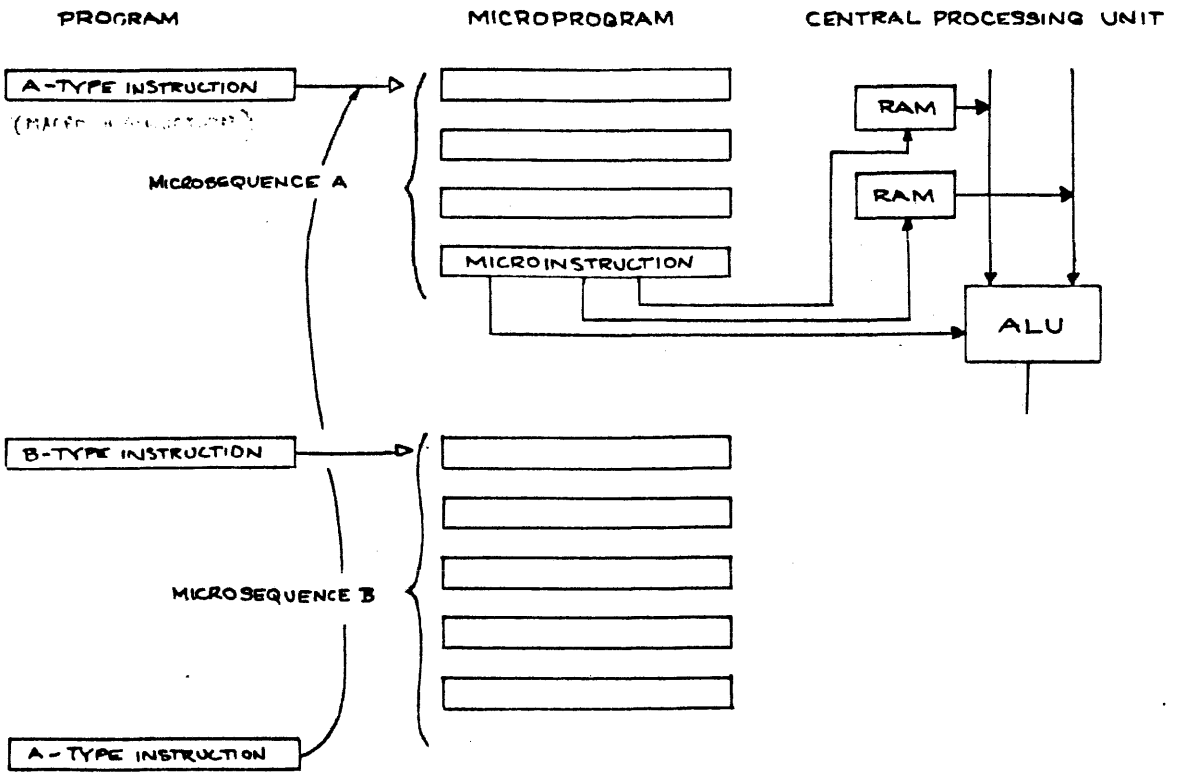
- USUALLY COMPRISES REGISTERS, ARITHMETIC - LOGICAL UNIT, AND CONTROL LOGIC

PROGRAM : THE DATA WHICH ULTIMATELY CONTROLS THE ACTIONS OF THE CENTRAL PROCESSOR

- USUALLY COMPRISES SEQUENCE OF INSTRUCTION WORDS IN MEMORY

MICROPROGRAM : A FIXED SEQUENCE OF INSTRUCTION WORDS WHICH CONTROL THE GATES OF A PROCESSOR. A MICROPROGRAM SEQUENCE IS INVOKED BY A SINGLE INSTRUCTION OF THE (MACRO)PROGRAM.

RELATIONSHIP OF PROGRAM, MICROPROGRAM, AND CPU



DEFINITIONS (CONTINUED)

LARGE SCALE INTEGRATION	:	THE TECHNOLOGY OF ASSEMBLING CIRCUITS OF ONE THOUSAND OR MORE GATES ON A SINGLE CHIP (E.G., MICROCOMPUTERS, ROMS & RAMS, BIT - SLICE DEVICES)
BIT - SLICE ARCHITECTURE	:	THE INTERCONNECTIONS OF LOGICAL ELEMENTS WITHIN THE BIT - SLICE CHIP
PROGRAMMED LOGIC ARRAY	:	AN LSI DEVICE FOR REPLACING DISCRETE LOGIC NETWORKS BY PERFORMING THEIR LOGICAL EQUIVALENTS THROUGH MATRIX MAPPING
COMPUTER ARITHMETIC	:	THE ALGORITHMS (PROGRAM STEPS) BY WHICH COMPUTERS PERFORM NUMERICAL CALCULATIONS

DEFINITIONS (CONTINUED)

EMULATION : THE PROGRAMMING OF A COMPUTER TO MAKE IT INTERPRET THE INSTRUCTION SET OF ANOTHER COMPUTER

- MOST OFTEN : MICROPROGRAMMED EMULATION - A MAJOR APPLICATION OF BIT-SLICE DEVICES

SPECIAL FUNCTION PROCESSES : COMPUTERS DESIGNED EXCLUSIVELY FOR SPECIFIC APPLICATIONS (E.G., I / O DEVICE CONTROLLERS, TEXT HANDLERS)

1. BRIEF HISTORY OF CPU DESIGN

SYBEX

CPU DESIGN EVOLUTION

B I T - S L I C E T E C H N O L O G Y

A STEP IN CPU DESIGN EVOLUTION ?

OR

A SEMICONDUCTOR INDUSTRY BY - PRODUCT ?

EVOLUTION

D E F I N I T I O N S -

H O R I Z O N T A L S T R U C T U R E S A R E C I R C U I T S

F O R M I N G

C O N T R O L L I N G THE DATA WORD

- REGISTERS
- ADDERS
- COMPARATORS
- MEMORY REGISTERS

V E R T I C A L S T R U C T U R E S A R E C I R C U I T S

F O R M I N G

C O N T R O L L I N G THE DATA PATH

- BUSSES
- ADDRESS SELECTION
- MULTIPLEXING

EVOLUTION

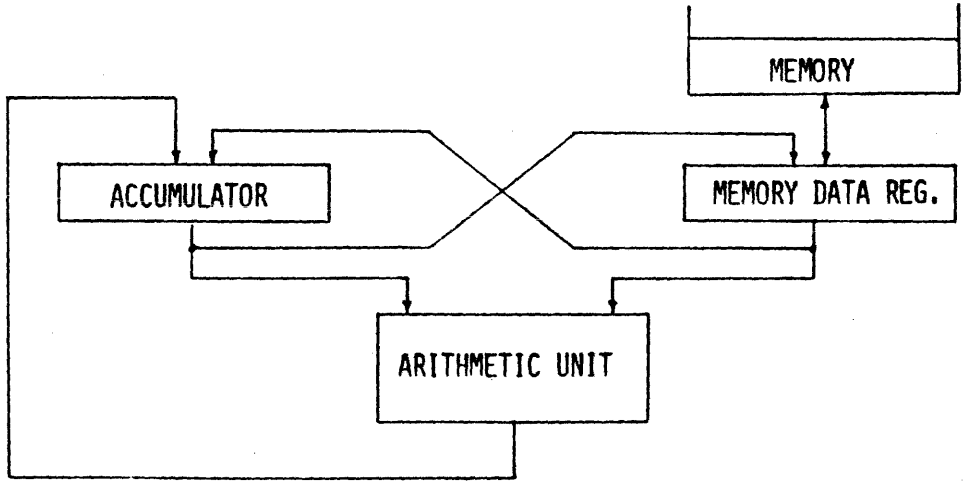
CPU DESIGN HAS EVOLVED FROM

HORIZONTAL DEVELOPMENT

TO

VERTICAL DEVELOPMENT

HORIZONTAL MACHINE ORGANIZATION



EVOLUTION

CPU DESIGN FOR MATHEMATICAL APPLICATIONS
EMPHASIZED HORIZONTAL ENHANCEMENTS

1. WIDER DATA WORDS
2. POWERFUL ARITHMETIC FUNCTIONS
3. FASTER ALU PROPAGATE TIMES
   TO COMPENSATE FOR WIDER DATA

VERTICAL ORGANIZATION IS THE SAME IN MOST MACHINES
FROM PDP - 1 TO CDC 7600

EVOLUTION

NEW APPLICATIONS BROUGHT NEW CPU DESIGN PHILOSOPHY

DATA STRUCTURE APPLICATIONS

- FILE SYSTEMS
- ASSEMBLERS
- COMPILERS
- TEXT EDITORS
- COMMUNICATION SYSTEMS
- NON-ARITHMETIC LANGUAGES: LISP

EVOLUTION

DATA STRUCTURE PROBLEMS ARE

WORD ORDER TRANSFORMATIONS. . .

WORD ORDER TRANSFORMATIONS ARE ACCOMPLISHED BY

DATA PATH MANIPULATIONS. . .

THEY ARE A FUNCTION OF THE VERTICAL STRUCTURE OF THE CPU

EVOLUTION

EXAMPLES OF DATA STRUCTURING PROBLEMS :

PARSING
FILE MOVEMENT
FILE EDITING
LINK - LIST BUILDING
STACK MANIPULATION
STRING EDITING

EVOLUTION

FUNCTIONAL DEMANDS CHANGED →

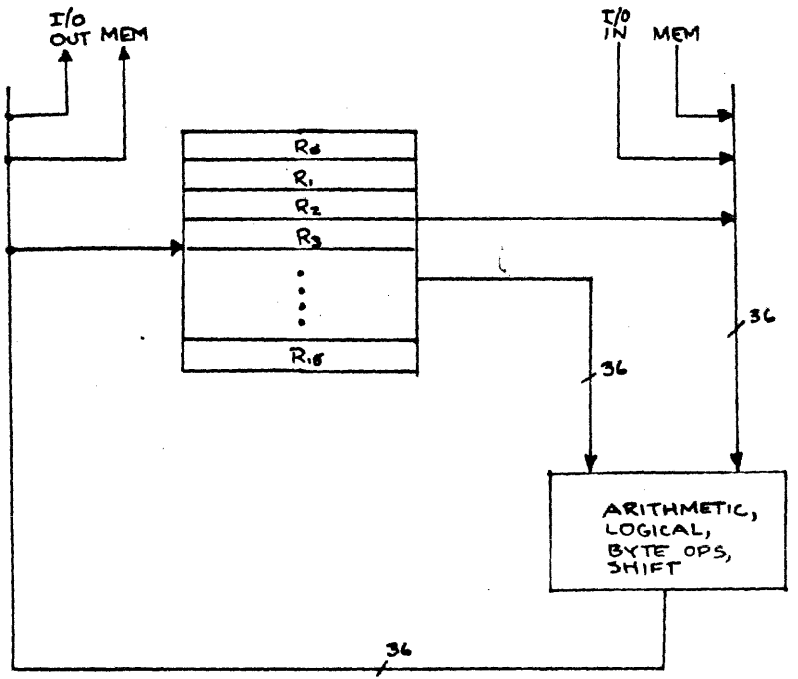
CPU STRUCTURAL DESIGN EVOLVED

HORIZONTAL → VERTICAL

EXAMPLE: FORTRAN → LISP

IBM 7090 → DEC PDP - 6

DEC PDP - 6 ROUGHLY



EVOLUTION - PDP - 6

PDP - 6 FEATURES

DOUBLE ADDRESSING
(REGISTER TO REGISTER OPERATIONS)

20_8 ACCUMULATORS

AUTOMATED STRING MOVES

(VARIABLE LENGTH BYTE OPERATIONS)

PDP - 6 ARCHITECTURE IS PRECURSOR OF BIT SLICE ARCHITECTURE

EVOLUTION - MACHINE POPULATION

SATURATION OF "NUMBER CRUNCHER" MARKET

GROWTH OF MINI-COMPUTER MARKET

RAPID ENHANCEMENT OF MINI-COMPUTERS

1. AUGMENTED REGISTER ADDRESSING (REG-TO-REG OPS)
2. AUGMENTED BYTE AND VECTOR HANDLING (BLTR)
3. CONTEXT SWITCHING
4. FASTER CYCLE TIMES FOR NON ARTHMETIC TRANSFER

BUT LITTLE OR NO ARITHMETIC AUGMENTATION

STANDARDIZED LENGTH 12 OR 16 BITS

EVOLUTION: SUMMARY

THE COMPUTER MARKET NOW DEMANDS
VERTICAL POWER AND FLEXIBILITY
AT THE EXPENSE OF
HORIZONTAL POWER

VERTICAL	HORIZONTAL
REGISTER SPACE	WORD WIDTH
COMPREHENSIVE BUSSING	ARITHMETIC FUNCTIONS
ITERATIVE DATA TRANSFERS	FAST PROPAGATE TIMES
TRANSFER CYCLE TIME	

2. BIT-SLICE PRINCIPLES

SYBEX

BIT SLICE PRINCIPLES

LARGE SCALE INTEGRATION BROUGHT TWO DEVELOPMENTS

1. TOTALLY INTEGRATED MICRO COMPUTERS
2. INTEGRATED BLOCKS OF ARCHITECTURE FOR LARGER COMPUTERS

LIMITATIONS OF CURRENT LSI

1. LOWER DENSITY FOR HIGH SPEED LOGIC
2. HIGHER DISSIPATION FOR HIGH SPEED LOGIC
3. PIN COUNT LIMITS

WHY THE BIT SLICE?

LSI DESIGNERS HAD TWO CHOICES

HORIZONTAL PARTITIONING

WIDE REGISTERS

WIDE MULTIPLEXERS

WIDE ALUs

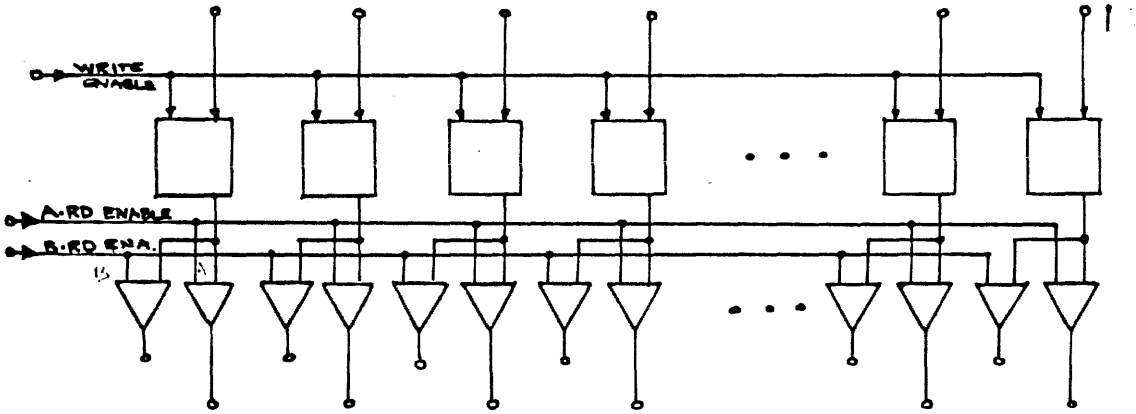
VERTICAL PARTITIONING

BIT WIDE SLICES OF THE ENTIRE DATA PATH OF CPU

WHY THE BIT SLICE?

HORIZONTAL LSI COMPONENTS (HYPOTHETICAL)

1. 16 BIT DUAL PORT REGISTER



~100 GATES

48 DATA PINS
3 CONTROL PINS
2 POWER PINS

2 GATES PER PIN

53 PINS

WHY THE BIT SLICE?

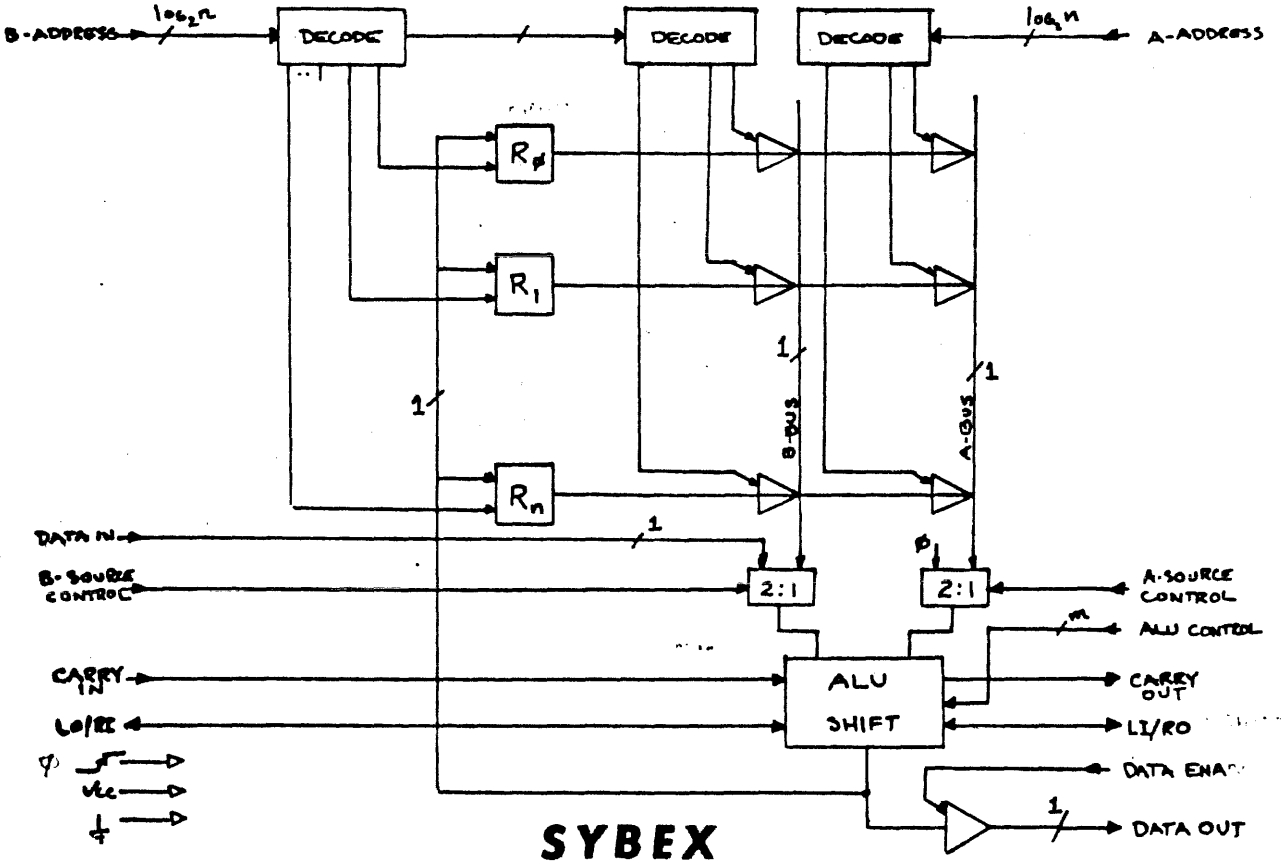
OTHER HYPOTHETICAL HORIZONTAL LSI COMPONENTS

2. 16 BIT FAST ALU ~400 GATES / 53 PINS: ~8 GATES / PIN
3. 16 BIT 3:1 MULTIPLEXER <100 GATES / 68 PINS: 1½ GATES / PINS

THESE ARE NOT TECHNICALLY LSI

HORIZONTAL STRUCTURES ARE PIN LIMITED TO MSI

HYPOTHETICAL ONE-BIT-WIDE SLICE



WHY THE BIT SLICE?

ASSUME 16 REGISTERS: 300 GATES / 23 PINS

OR

13 GATES PER PIN

MULTIPLYING SLICES ADDS 2 PINS PER SLICE:

EXTERNAL INPUT, EXTERNAL OUTPUT

MULTIPLYING SLICES ADDS 200 GATES PER SLICE

EXAMPLE 8 BIT SLICE 1700 GATES

39 PINS

43 GATES / PIN

OTHER FACTORS FAVORING VERTICAL INTEGRATION

SPEED:

EXTERNAL CONNECTIONS INCREASE
CAPACITANCE → DECREASE SPEED

DISSIPATION:

T^2L COMPATIBLE DRIVE REQUIRED EXTERNAL
TO CHIP -- 10mW (5V 2mA)
INTERNAL DISSIPATION PER GATE TYPICALLY
1.2 - 1.4mW: LOW POWER SHOTTKY
100 MICROWATTS: I^2L (TI SBP0400)

BIT SLICE STRATEGY:

MAXIMIZE INTERNAL VERTICAL INTERCONNECTIONS
AT THE EXPENSE OF EXTERNAL HORIZONTAL INTERCONNECTIONS

BIT-SLICE DEVICES

1974: THE COMMERCIAL BIRTH YEAR OF BIT-SLICE

NATIONAL SEMICONDUCTOR	IMP SERIES
INTEL	3000 SERIES

SINCE THEN

MONOLITHIC MEMORIES	5700/6700 SERIES
ADVANCED MICRO DEVICES	2900 SERIES
MOTOROLA	10800
TEXAS	SBP0400

AND SECOND SOURCES IN U.S. AND EUROPE

BIT SLICE APPLICATIONS

BIT SLICE DEVICES ARE TO DESIGNERS IN 1976
WHAT MSI WAS TO DESIGNERS IN 1973

BIT SLICE DEVICES ARE FOUND IN PRODUCTION
MODELS OF PROTOTYPES OF

POPULAR MINI COMPUTERS

FLOATING POINT ADD-ONS

HIGH SPEED I/O ADAPTERS: DISK & COMMUNICATIONS

SPECIAL FUNCTION BOXES: FFT, NAVIGATION

SPECIAL ARCHITECTURES: DISTRIBUTED PROCESSOR DESIGNS

A BIT SLICE DEVICE IS NOT A MICROPROCESSOR

	BIT-SLICE	LSI MICRO	BIT-SLICE ADD-ONS
ALU	X	X	
SHIFT	X	X	
REGISTERS	X	X	
REGISTER SELECTION	X	X	
INTPUT TO BUS	X	X	
OUTPUT FROM BUS	X	X	
MICRO INSTRUCTION SEQUENCING		X	X
MEMORY ADDRESSING		X	X
ALU + SHIFT CONTROL		X	X
I/O CONTROL		X	X

BIT SLICE PRINCIPLES: SUMMARY

BIT-SLICE DEVICES

ARE VERTICAL STRUCTURES:
2 OR 4/N^{THS} OF THE DATA PATH.
CIRCUITRY OF AN N-BIT PROCESSOR

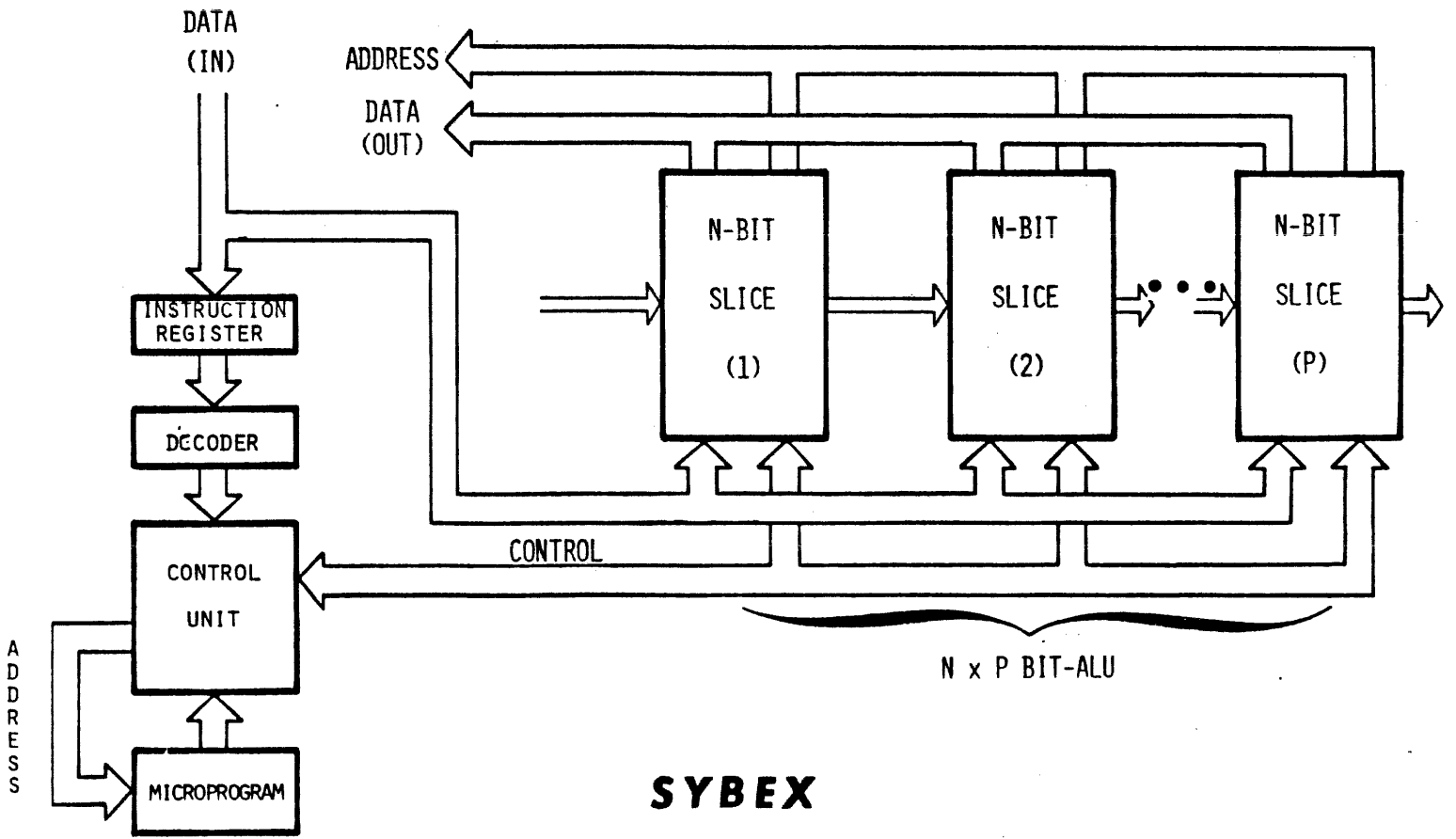
HAVE NO CONTROL LOGIC-
CONTROL SIGNALS ARE SUPPLIED FROM EXTERNAL CONTROLLER CHIPS

ARE INTENDED FOR FAST APPLICATIONS
CYCLE SPEEDS BETTER THAN SHOTTKY MSI EQUIVALENTS

ARE NOT INTENDED FOR MINIMUM COST APPLICATIONS

MUCH EXTERNAL LOGIC REQUIRED

BIT-SLICE SYSTEM

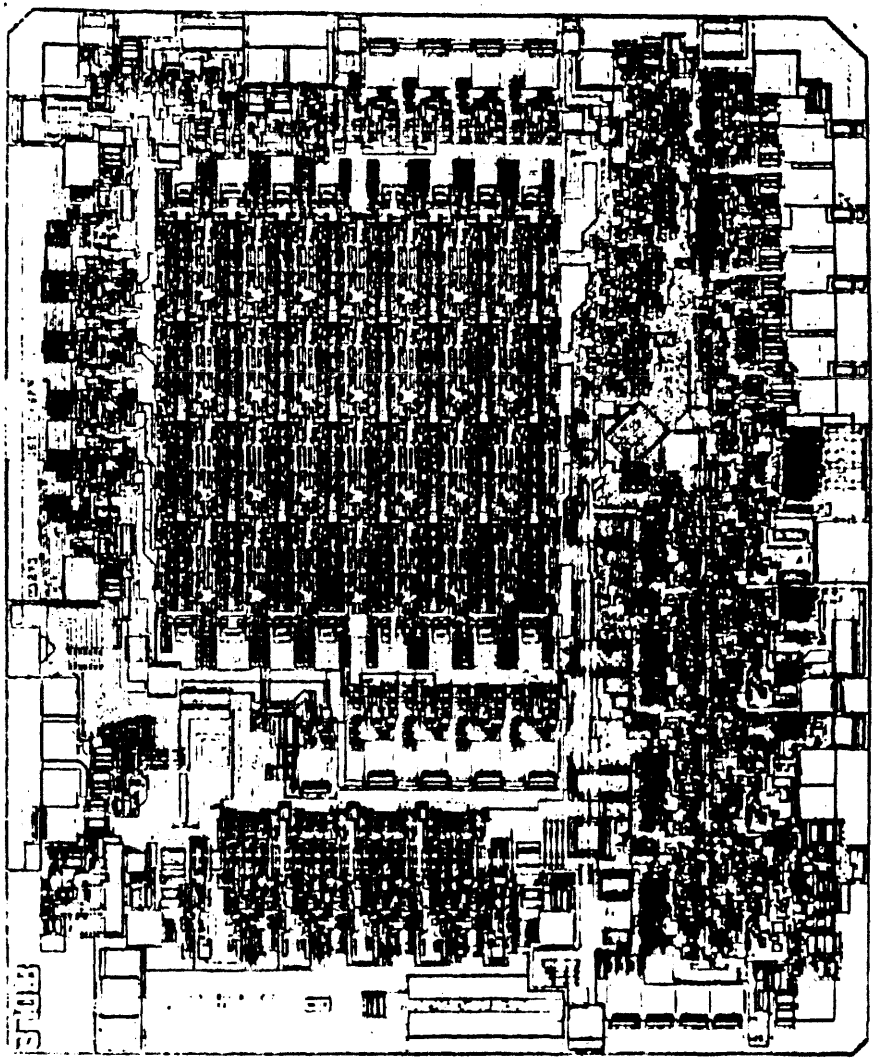


3. BIT-SLICE IN DETAIL:

BUILDING WITH BIT-SLICE DEVICES

SYBEX

AM 2901 BIT-SLICE PHOTOMICROGRAPH



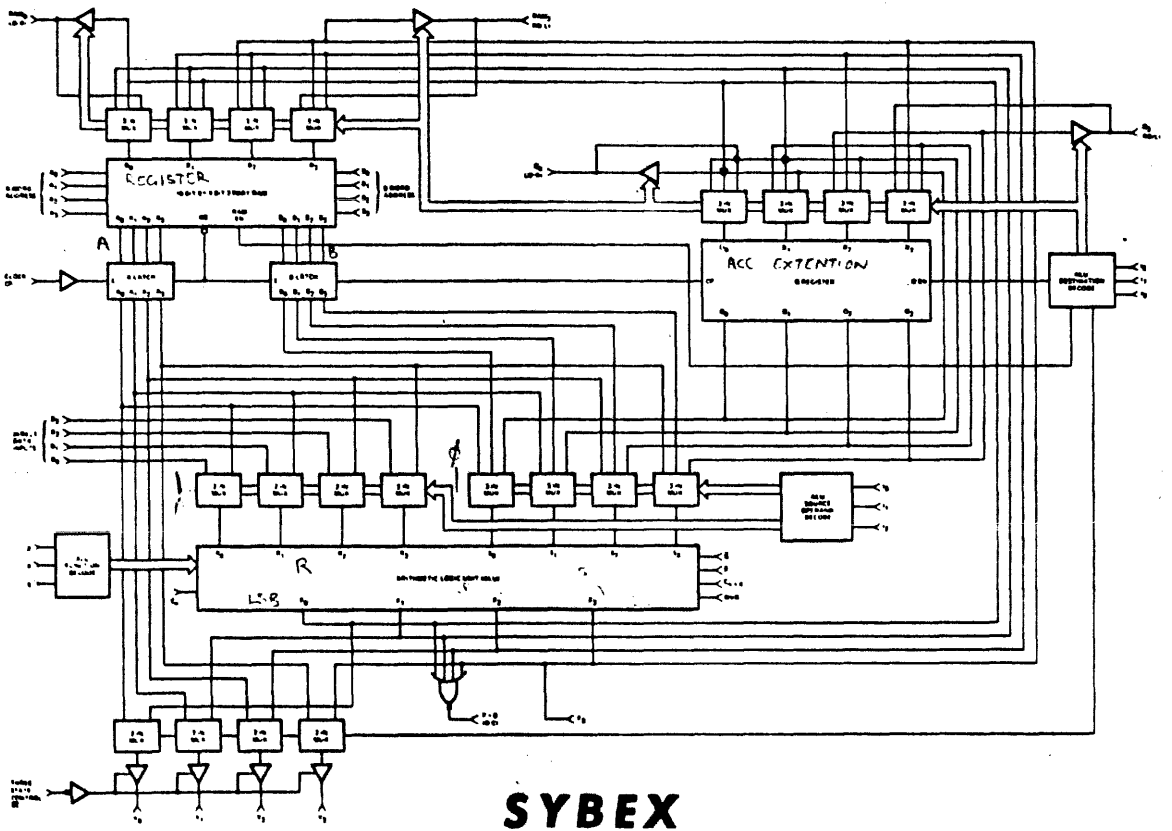
A BIT SLICE DEVICE IN DETAIL

A D V A N C E D M I C R O D E V I C E S 2901

F E A T U R E S

- 4 BIT SLICE
- 16 WORD TWO PORT RAM
- 8 FUNCTION ALU
- INDEPENDENT SHIFT
- FOUR STATUS FLAGS
- 9-BIT CONTROL MICRO-INSTRUCTION
- BIPOLAR: LOW POWER SHOTTKY IMPLEMENTATION
- MIN 105 NS CLOCK FOR COMMERCIAL VERSION

AMD 2901 BLOCK DIAGRAM



AMD 2901 - BUS STRUCTURE

ALU INPUT SOURCES

	R BUS	ABR		S BUS	ABR
1.	RAM A-LATCH (1 OF 16 REGISTERS)	A	1.	RAM A-LATCH (1 OF 16 REGISTERS)	A
2.	DIRECT DATA INPUTS	D	2.	RAM B-LATCH (1 OF 16 REGISTERS)	B
	AND INHIBIT		3.	Q - REGISTER	Q
	(FOR ON BUS)	Ø		AND INHIBIT (FOR ON BUS)	Ø

AMD 2901 - BUS STRUCTURE

ALU OUTPUT-F BUS DESTINATIONS

1. CHIP OUTPUT (Y BUS) MULTIPLEXER
2. Q REGISTER MULTIPLEXER
3. SHIFTER/RAM MULTIPLEXER

OTHER PATHS

1. A-LATCH TO Y-MULTIPLEXER
2. Q-REGISTER OUTPUT TO Q REGISTER MULTIPLEXER

INPUTS { SHIFT LEFT
SHIFT RIGHT

AMD 2901 - FLAGS AND STATUS OUTPUT

OTHER DATA OUTPUT

1. CARRY GENERATE $\overline{G}$
2. CARRY PROPAGATE $\overline{P}$
3. CARRY OUT C
4. OVERFLOW ($C_{n+3} \neq C_{n+4}$) OVR
5. F-BUS = $\emptyset$ F = 0
6. F-BUS MSB (SIGNBIT) F_3
7. RAM SHIFT LEFT OUT (SHARED)
8. RAM SHIFT RIGHT OUT (SHARED)
9. Q-REG SHIFT LEFT OUT (SHARED)
10. Q-REG SHIFT RIGHT OUT (SHARED)

AMD 2901 - FLAGS AND STATUS INPUT

O T H E R D A T A I N P U T

1. CARRY IN C_N

2. RAM SHIFT LEFT IN (SHARES WITH RAM SHIFT RIGHT OUT)

3. RAM SHIFT RIGHT IN (SHARES WITH RAM SHIFT LEFT OUT)

4. Q REG SHIFT LEFT IN (SHARES Q SHIFT RIGHT OUT)

5. Q REG SHIFT RIGHT IN (SHARES Q SHIFT LEFT OUT)

AMD 2901 - CHIP CONTROL

AREAS OF CONTROL

1. ALU INPUTS R AND S
2. ALU FUNCTIONS
3. Y BUS MUX, Y BUS OUTPUT ENABLE
4. RAM SHIFT MULTIPLEXER
5. RAM A ADDRESS
6. RAM B ADDRESS
7. Q-REGISTER MULTIPLEXER

PLUS CLOCK

AMD 2901 - CONTROL WORD

CONTROL WORD FORMAT 9 BIT

I

8	7	6	5	4	3	2	1	0
DESTIN.			F(A L U)			SOURCE		

1. ALU SOURCE CONTROL

I_2 I_1 I_0	R	S
0	A	Q
1	A	B
2	Ø	Q
3	Ø	B
4	Ø	A
5	D	A
6	D	Q
7	D	Ø

AMD 2901 - CONTROL WORD

2. ALU FUNCTION CONTROL

I_5 I_4 I_3	ALU FUNCTION	SYMBOL
0	R PLUS S	R + S
1	S MINUS R	S - R
2	R MINUS S	R - S
3	R OR S	R ∨ S
4	R AND S	R ∧ S
5	$\overline{R}$ AND S	$\overline{R}$ ∧ S
6	R X OR S	R ⊻ S
7	R X NOR S	$\overline{(R \veebar S)}$

AMD 2901 - CONTROL WORD

3. DESTINATION CONTROL

$I_{8,7,6}$	RAM SHIFT	RAM LOAD	Q SHIFT	Q LOAD	Y OUTPUT
0			NO	$F_{\phi-3}$	F
1					F
2	NO	$F_{\phi-3}$			A
3	NO	$F_{\phi-3}$			F
4	LEFT ($F_\phi \rightarrow$ RLO)	F_{1-3}, RLI	LEFT ($Q_\phi \rightarrow$ QLO)	Q_{1-3}, QLI	F
5	LEFT ($F_\phi \rightarrow$ RLO)	F_{1-3}, RLI			F
6	RIGHT ($F_3 \rightarrow$ RRO)	$F_{\phi-2}$, RRI	RIGHT ($Q_3 \rightarrow$ QRO)	$Q_{\phi-2}$, QRI	F
7	RIGHT ($F_3 \rightarrow$ RRO)	$F_{\phi-2}$, RRI			F

AMD 2901 - ALU FUNCTIONS

BY CONTROL OF I_{ϕ} - I_5 AND C_N THE AM2901 PERFORMS THE FOLLOWING ARITHMETIC FUNCTIONS IN ONE STEP

I_{5-3}	I_{2-0}		$C_N = 0$		$C_N = 1$
	0	ADD	A + Q	ADD PLUS ONE	A + Q + 1
	1		A + B		A + B + 1
Ø	5		D + A		D + A + 1
	6		D + Q		D + Q + 1
	2	PASS	Q	INCREMENT	Q + 1
	3		B		B = 1
Ø	4		A		A = 1
	7		D		D = 1
1	2	DECREMENT	Q - 1	PASS	Q
1	3		B - 1		B
1	4		A - 1		A
2	7		D - 1		D

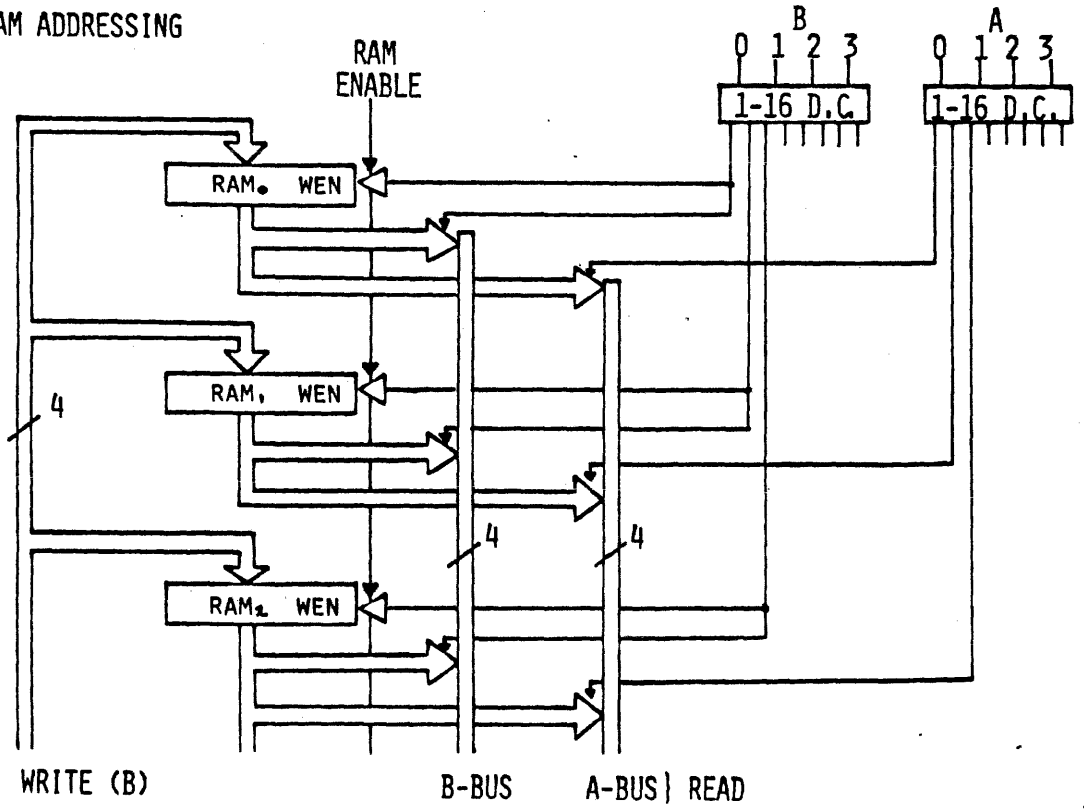
AMD 2901 - ALU FUNCTIONS CONTINUED

I_{5-3},	I_{2-0}	C_N		$C_N = 1$	
2	2	ONES COMPL.	- Q - 1	TWOS COMP.	-Q
2	3		- B - 1	(NEGATE)	-B
2	4		- A - 1		-A
1	7		- D - 1		-D
1	0	SUBTRACT	Q - A - 1		Q - A
1	1	ONES COMPL.	B - A - 1		B - A
1	5		A - D - 1		A - D
1	6		Q - D - 1		Q - D
2	0		A - Q - 1		A - Q
2	1		A - B - 1		A - B
2	5		D - A - 1		D - A
2	6		D - Q - 1		D - Q

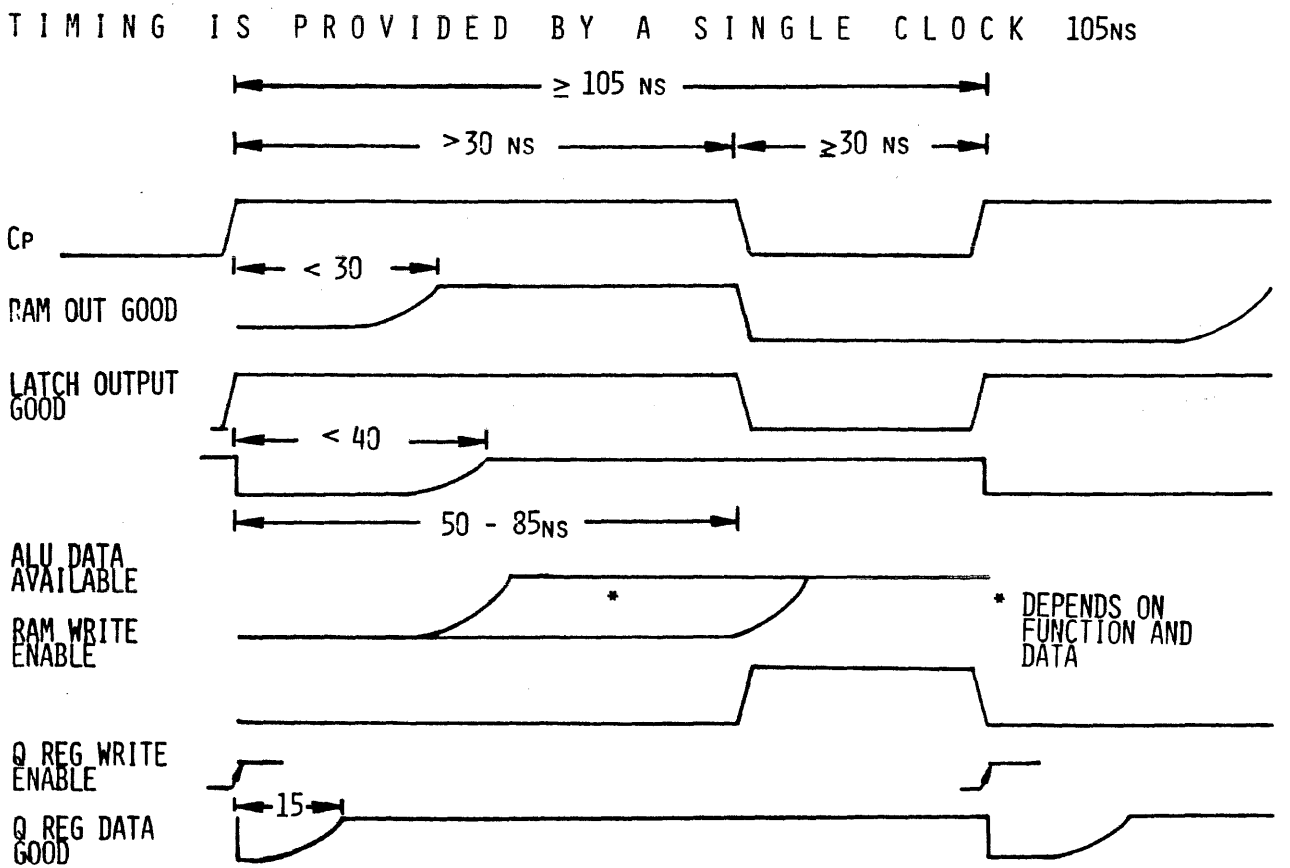
AMD 2901 CONTROL

9 BIT CONTROL WORD PROVIDES ALL CHIP CONTROL
EXCEPT RAM ADDRESSING

RAM ADDRESSING

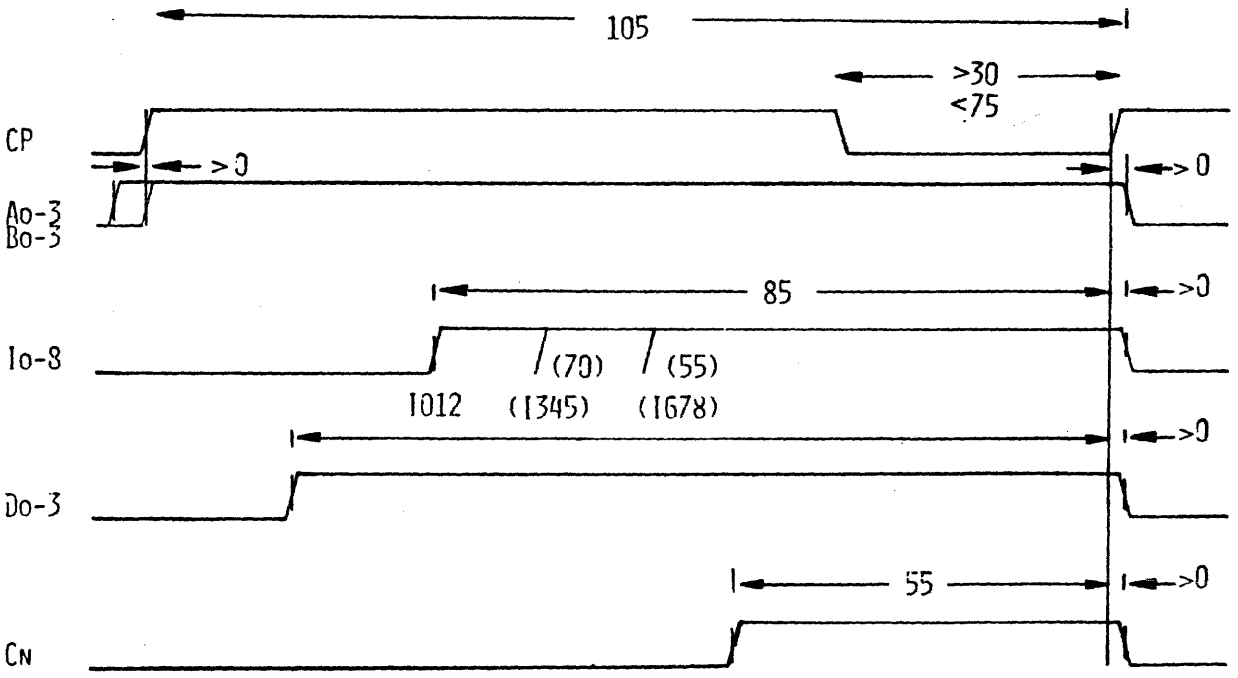


AMD 2901 TIMING



AMD 2901: TIMING CONTINUED

EXTERNAL SIGNAL TIMING REQUIREMENTS



- RAM ADDRESSES ARE THE CRITICAL TIMING SIGNAL

AMD 2901: TIMING TABLES

CYCLE TIME AND CLOCK CHARACTERISTICS

TIME	Am2901DC	Am2901DM
Minimum Read-Modify-Write Cycle (time from selection of A, B registers to end of cycle)	105 ns	120 ns
Maximum Clock Frequency to Shift Q Register (50% duty cycle)	9.5 MHz	8.3 MHz
Minimum Clock LOW Time	30 ns	30 ns
Minimum Clock HIGH Time	30 ns	30 ns
Minimum Clock Period	105 ns	120 ns

GUARANTEED OPERATING CONDITIONS

Tables I, II, and III below define the timing requirements of the Am2901 in a system. The Am2901 is guaranteed to function correctly over the operating range when used within the delay and set up time constraints of these tables for the appropriate device type. The tables are divided into three types of parameters: clock characteristics, combinational delays from inputs to outputs, and set-up and hold time requirements. The latter table defines the time prior to the end of the cycle (i.e., clock LOW-to-HIGH transition) that each input must be stable to guarantee that the correct data is written into one of the internal registers.

The performance of the Am2901 within the limits of these tables is guaranteed by the testing defined as "Group A, Subgroup 9" Electrical Testing. For a copy of the tests and limits used for subgroup 9, contact Advanced Micro Devices' Product Marketing.

TABLE II

MAXIMUM COMBINATIONAL PROPAGATION DELAYS (all in ns, $C_L \leq 15pF$)

From Input \ To Output	Am2901DC Y	F_3	C_{n+4}	$\overline{G}, \overline{P}$	F=0 R_L=470	OVR	RO, LO RAM	RO, LO Q	Am2901DM Y	F_3	C_{n+4}	$\overline{G}, \overline{P}$	F=0 R_L=470	OVR	RO, LO RAM	RO, LO Q
Clock ↓	115	85	100	100	110	95	105	60	125	95	110	110	120	105	115	65
A, B	110	85	80	80	110	75	110	–	120	95	90	90	120	85	120	–
D	100	70	70	70	100	60	60	–	110	80	75	75	110	65	65	–
C_n	55	35	30	–	50	40	55	–	60	40	30	–	55	45	60	–
I_{012}	85	65	65	65	80	65	80	–	90	70	70	70	85	70	85	–
I_{345}	70	55	60	60	70	60	65	–	75	60	65	65	75	65	70	–
I_{678}	55	–	–	–	–	–	45	45	60	–	–	–	–	–	50	50
$\overline{OE}$ Enable/Disable	40/25	–	–	–	–	–	–	–	40/25	–	–	–	–	–	–	–
A bypassing ALU (I = 2xx)	60	–	–	–	–	–	–	–	65	–	–	–	–	–	–	–

SET-UP AND HOLD TIMES (minimum cycles from each input)

Set-up and hold times are defined relative to the clock LOW-to-HIGH edge. Inputs must be steady at all times from the set-up time prior to the clock until the hold time after the clock. The set-up times allow sufficient time to perform the correct operation on the correct data so that the correct ALU data can be written into one of the registers.

TABLE III

Set-Up and Hold Times (all in ns) (Note 1)

From Input	Notes	Am2901DC Set-Up Time	Am2901DC Hold Time	Am2901DM Set-Up Time	Am2901DM Hold Time
A, B Source	2, 3, 4	105 $t_{pw}L + 30$	0	120 $t_{pw}L + 30$	0
B Dest.	2, 4	$t_{pw}L + 15$	0	$t_{pw}L + 15$	0
D		100	0	110	0
C_n		55	0	60	0
I_{012}		85	0	90	0
I_{345}		70	0	75	0
I_{678}	4	$t_{pw}L + 15$	0	$t_{pw}L + 15$	0
RI, LI (RAM or Q)		30	0	30	0

Notes: 1. See Figure 11 and 12.
2. If the B address is used as a source operand, allow for the "A, B source" set-up time. If it is used only for the destination address, use the "B dest." set-up time.
3. Where two numbers are shown, both must be met.
4. "$t_{pw}L$" is the clock LOW time.

AMD 2901: OTHER DATA

E L E C T R I C A L

1. 5.0 V O L T S U P P L Y

2. 185 M A C U R R E N T T Y P I C A L - 280 M A X (1.4 W A T T S)

T E M P E R A T U R E

C O M M E R C I A L 0°C TO +70°C / 4.75V TO 5.25V / 105ns

M I L I T A R Y -55°C TO + 125°C / 4.50 TO 5.50V / 120ns

P H Y S I C A L

40 P I N D U A L I N - L I N E P A C K A G E (A L L P I N S U S E D)

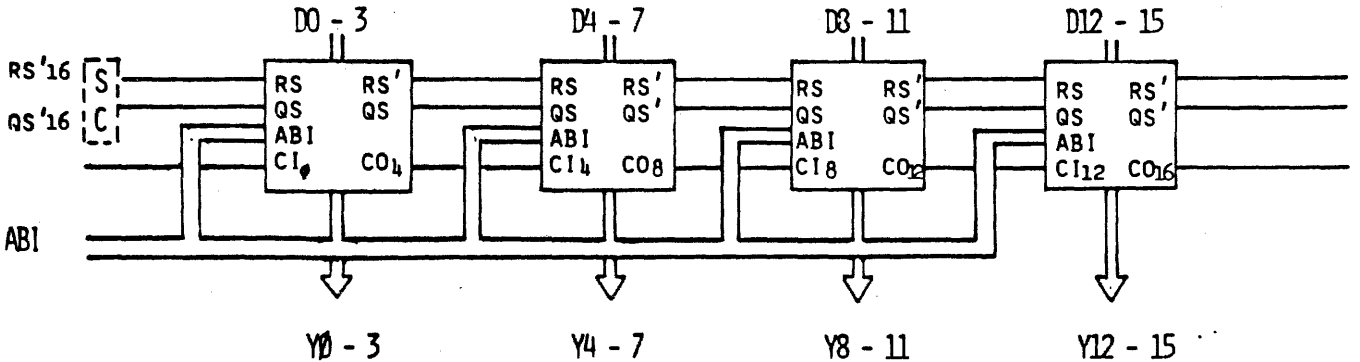
.2.000 I N C H E S M Y .580 I N C H E S

BIT SLICE APPLICATION: CASCADING 2901s

S I M P L E C A S C A D I N G

RIPPLE CARRY
SHIFT OFF ENDS OR END AROUND SHIFT

E X A M P L E: 16 BIT CPU



RS = RAM LO/RI RS' = RAM RO/LI
QS = QREG LO/RI QS' = QREG RO/LI
ABI = A_{0-3} B_{0-3} I_{0-8}

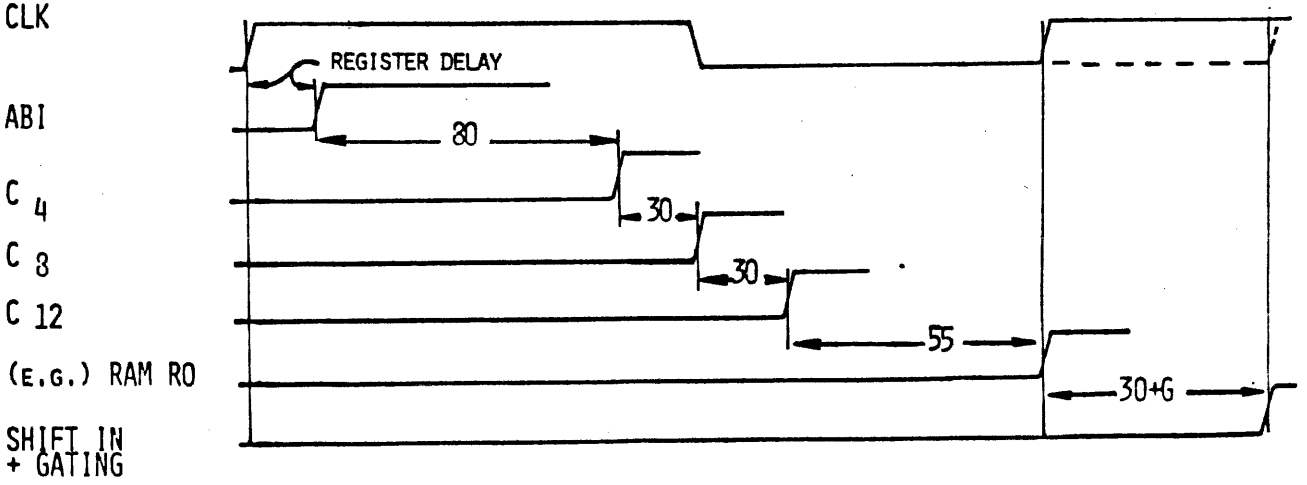
SIMPLE CASCADE — TIMING IMPLICATIONS

CLOCK RATE MUST ACCOMODATE WORST CASE TIMING

FULL 16 BIT RIPPLE CARRY

CRITICAL TIMES

CLOCK	TO	$C_N + 4$ = 100NS
A, B	TO	$C_N + 4$ = 80NS
C_N	TO	$C_N + 4$ = 30NS
C_N	TO	RAM RO/LO (LAST OUTPUT) = 55NS



AMD 2901 SIMPLE CASCADE TIMING

SHIFT OF END - CLOCK RATE

REGISTER SET UP + CARRY DELAYS + LAST OUTPUT DELAY

20 + 140 + 55 = 215NS

END AROUND SHIFT

ADD SHIFT IN + GATING

215 + 30 + 15 = 260NS

MINIMUM CLOCK FOR A PRACTICAL 16-BIT RIPPLE CPU

CP = 260NS

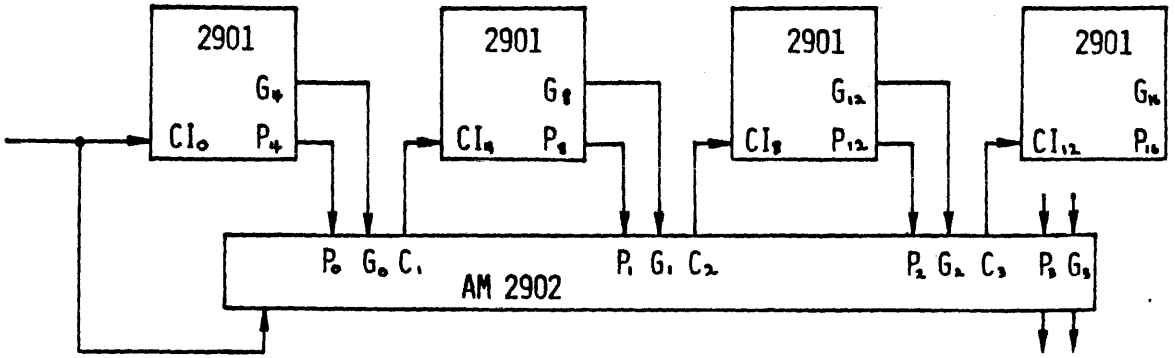
AMD 2901: LOOK AHEAD CASCADE

GENERATE AND PROPAGATE OUTPUTS

DESIGNED FOR STANDARD LOOK AHEAD
AM 2902/ 74S182

SIMULTANEOUS WITH C_{N+4} OUTPUT

CONFIGURATION FOR 16 BIT CPU

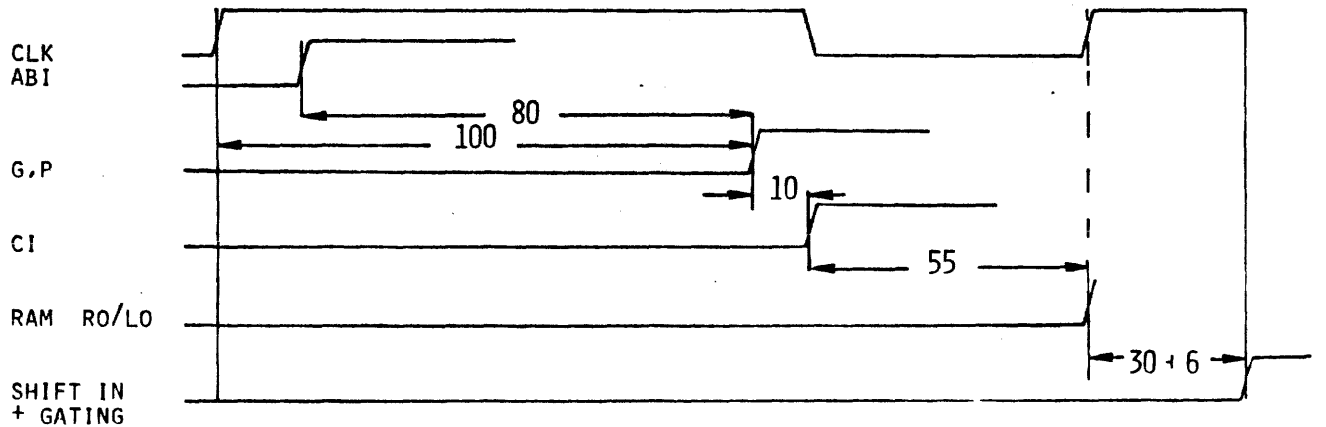


SYBEX

AMD 2901: LOOK AHEAD CASCADE TIMING

C R I T I C A L T I M E S

CI TO AM2902 C_1 C_2 C_3 = 10NS
ABI TO G,P = 100NS
CLK TO G,P = 80NS
CI TO RAM RO/LO = 55NS
SHIFT IN SETUP = 30NS



AMD 2901 LOOK AHEAD CASCADE TIMING

MINIMUM CLOCK WITH LOOK - AHEAD/ WITHOUT SHIFT AROUND

CP = 165NS

MINIMUM CLOCK WITH LOOK - AHEAD AND SHIFT AROUND

CP = 210NS

LOOK AHEAD CIRCUITRY GAINS 50NS PER CYCLE FOR 16 BIT CPU

GAINS INCREASE AS WORD WIDTH INCREASES

EXAMPLE: 48 BIT RIPPLE CP = 500NS

48 BIT LOOK - AHEAD CP = 240NS.

AMD 2901 RIPPLE VERSUS LOOK-AHEAD

	RIPPLE	LOOK - AHEAD
16 - BIT SIMPLE SHIFT	215	165
16 - BIT END AROUND	260	210
48 - BIT SIMPLE SHIFT	455	195
48 - BIT END AROUND	500	240

FORMULAE FOR N BIT SIMPLE SHIFT

RIPPLE: $CP = 155 + 30 ((N/4) - 2)$ NS

LOOK AHEAD: $CP = 155 + 10 L$ NS

WHERE L = LENGTH OF 74S182'S CASCADE

ADD 30 PLUS GATING FOR END AROUND SHIFT

BLANK

USING THE BIT-SLICE DEVICE

AN ARRAY OF BIT SLICE DEVICES FORMS A PROCESSING UNIT.

AN ALGORITHM IS A SEQUENCE OF STEPS PERFORMED BY THE PROCESSING UNIT.

ALGORITHMS MAY ORIGINATE FROM

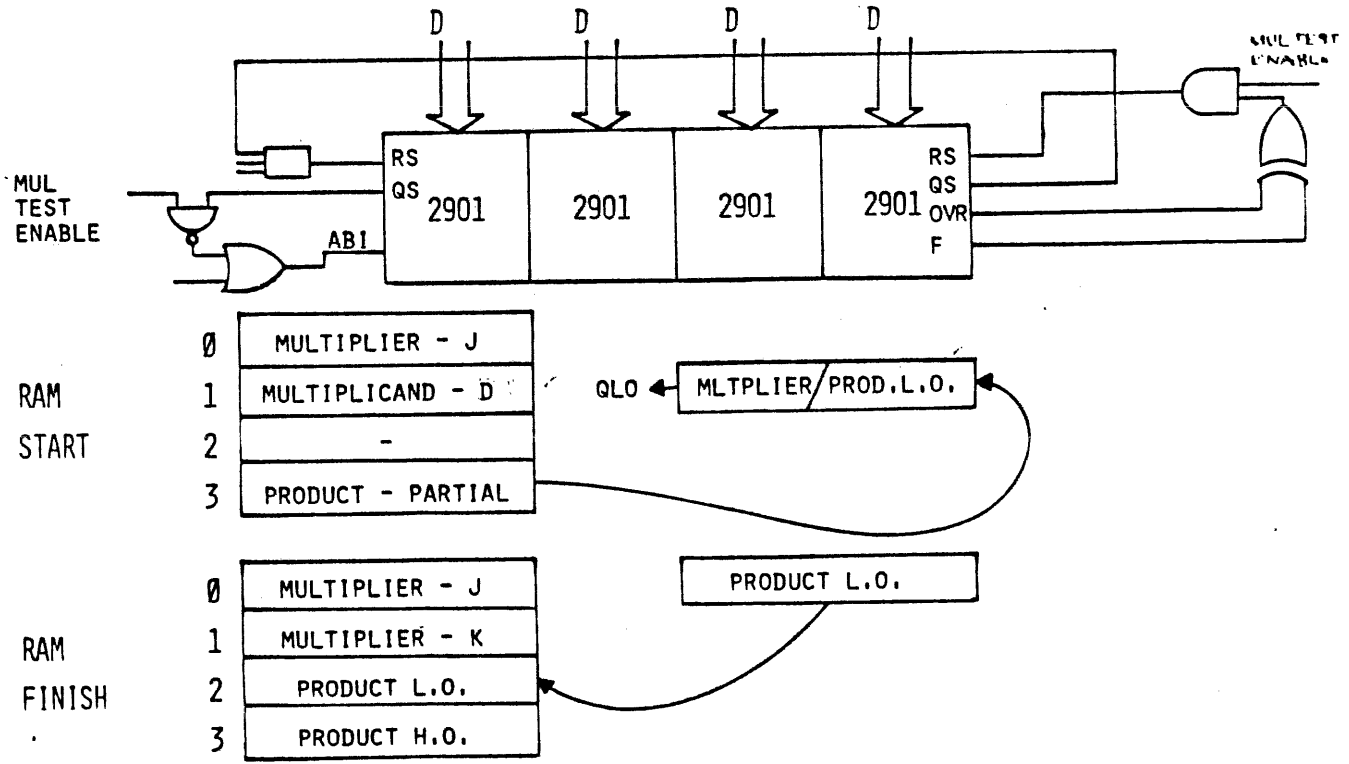
- MACRO-PROGRAMS
- MICRO-PROGRAMS
- LOGIC

THE INSTRUCTION AND REGISTER ADDRESS LOGIC OF THE BIT SLICE ARRAY

DETERMINE THE FINAL FORMAT OF CONTROL FOR A GIVEN ALGORITHM.

A MULTIPLY ALGORITHM FOR A 16-BIT 2901 CPU

CONFIGURATION - 16 BITS (WITH OR WITHOUT LOOKAHEAD)



MULTIPLY ALGORITHM

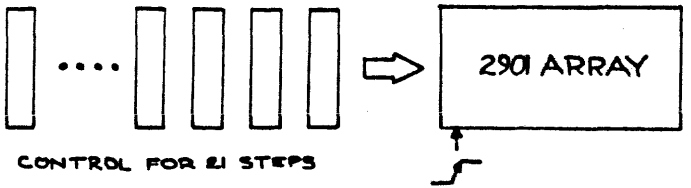
EXT	SOURCE	FUNCTION	DESTINATION	A	B	
J → D	D, Ø	D V Ø	B		Ø	LOAD MULTIPLIER R Ø
K → D	D, Ø	D V Ø	B		1	LOAD MULTIPLICAND R 1
	Ø, A	Ø V A	Q	Ø		MOVE J TO Q-REG
	Ø, B	Ø ∧ B	B		3	CLEAR R 3
MULTEST ENA ← 1	Ø, B OR* A, B	Ø/A + B, SHIFT LEFT	B	1	3	ADD MULTIPLICAND TO R 3 IF LSB OF Q IS ONE. THEN SHIFT LEFT COMBINED PRODUCT R 3 (NOTE TIMING QLO)
	REPEAT 15 TIMES					
MULTEST ENA ← 1	Ø, B OR* A, B	B - Ø/A, SHIFT LEFT	B	1	3	TRIAL SUBTRACT AND GATING OF SIGN INTO MSB OF R 3
MULTEST ENA ← Ø	Ø, Q	Ø V Q	B		2	STORE PRODUCT L.O FROM Q TO R 2

* $I_{2,1,\emptyset} = \emptyset$, $I_1 \vee (\text{MULTESTENA} \wedge \text{QLO})$, $1 = \emptyset\emptyset 1 / \emptyset 11$

BIT-SLICE ARRAY: SEQUENCING

MULTIPLY ALGORITHM: 21 INSTRUCTION STEPS

(16 BIT RIPPLE 260 NS 16 BIT LOOK AHEAD 210 NS)



SEQUENCE DATA IS CALLED

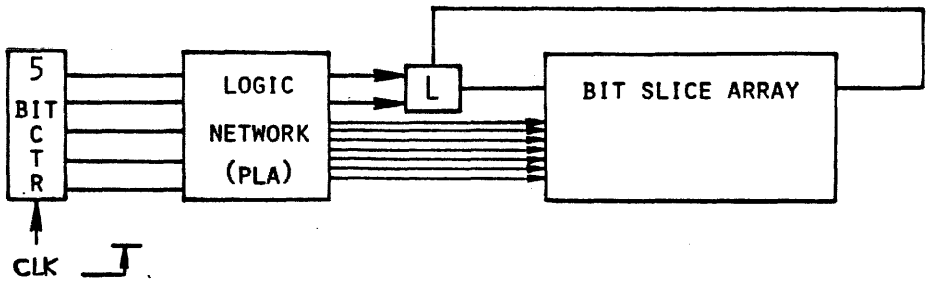
MICRO SEQUENCE
MICRO PROGRAM
MACRO PROGRAM

FOR THE MULTIPLY CONFIGURATION 21 x 18 BITS

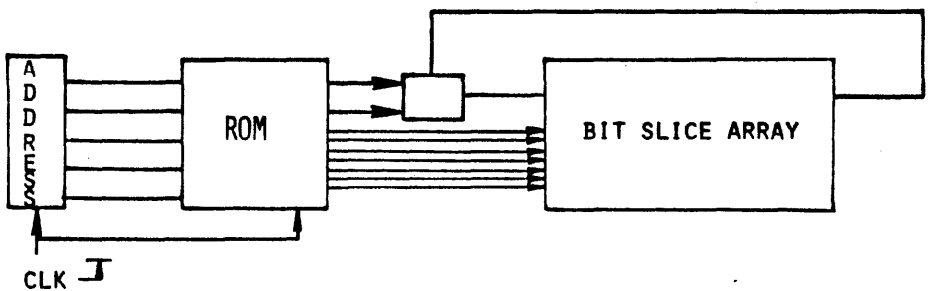
BIT SLICE ARRAY SEQUENCING TECHNIQUES

VARIOUS METHODS OF APPLYING SEQUENCE DATA TO DEVICE INPUTS

LOGIC

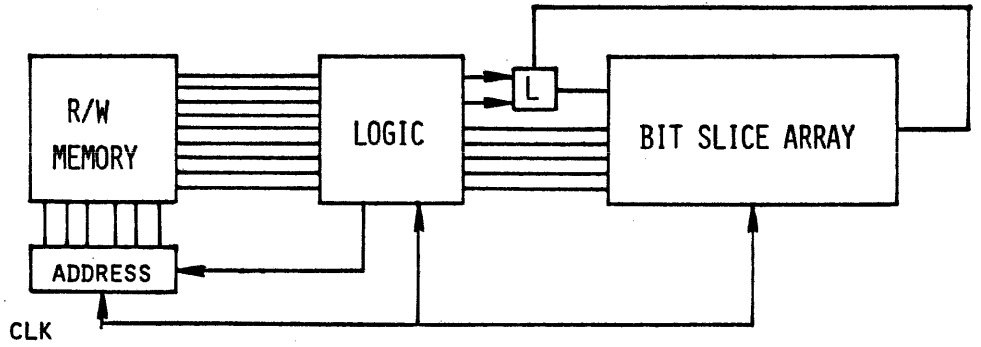


ROM

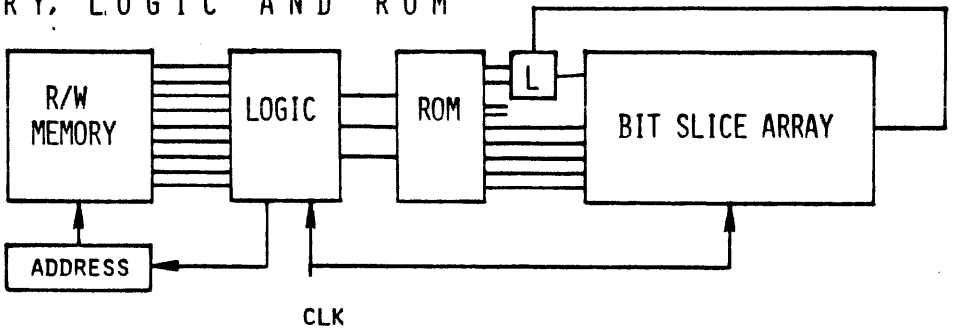


BIT SLICE ARRAY SEQUENCING TECHNIQUES

MEMORY AND LOGIC



MEMORY, LOGIC AND ROM



BIT SLICE ARRAY SEQUENCING TECHNIQUES

LOGIC ONLY (HARDWARE PROCESSOR)

FAST, EXPENSIVE, INFLEXIBLE, FROZEN

ROM ONLY (MICROSEQUENCED PROCESSOR)

FAST, INEXPENSIVE, FLEXIBLE, FROZEN

MEMORY AND LOGIC (PROGRAMMED PROCESSOR)

SLOW, EXPENSIVE, FLEXIBLE, PROGRAMABLE

MEMORY, LOGIC, AND ROM (MICROPROGRAMMED PROCESSOR)

FAST, INEXPENSIVE (?), FLEXIBLE, PROGRAMABLE

--THE LAST CATEGORY COVERS ALMOST EVERY MACHINE MARKETED SINCE ROMS AND MSI BECAME COMMERCIAL

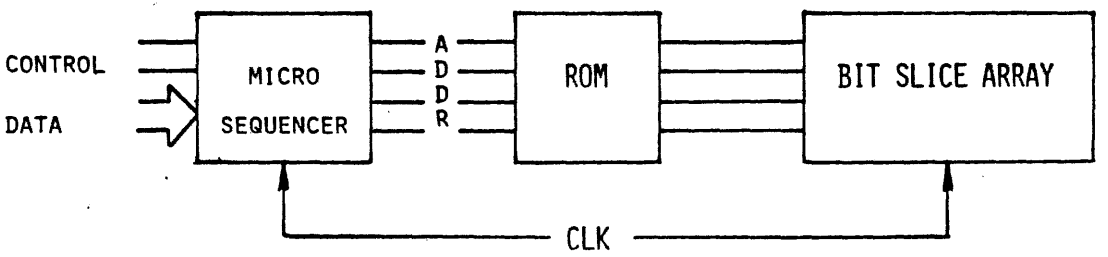
BIT SLICE MICROPROGRAM SEQUENCERS

CONTROL ROM INSTRUCTION SOURCE

MAY CONTROL BIT SLICE STATUS AND SHIFT LINES

MAY CONTROL ANCILLARY LOGIC

SIMPLEST FORM



LOADS ADDRESS REGISTER FROM EXTERNAL DATA

INCREMENTS ADDRESS REGISTER

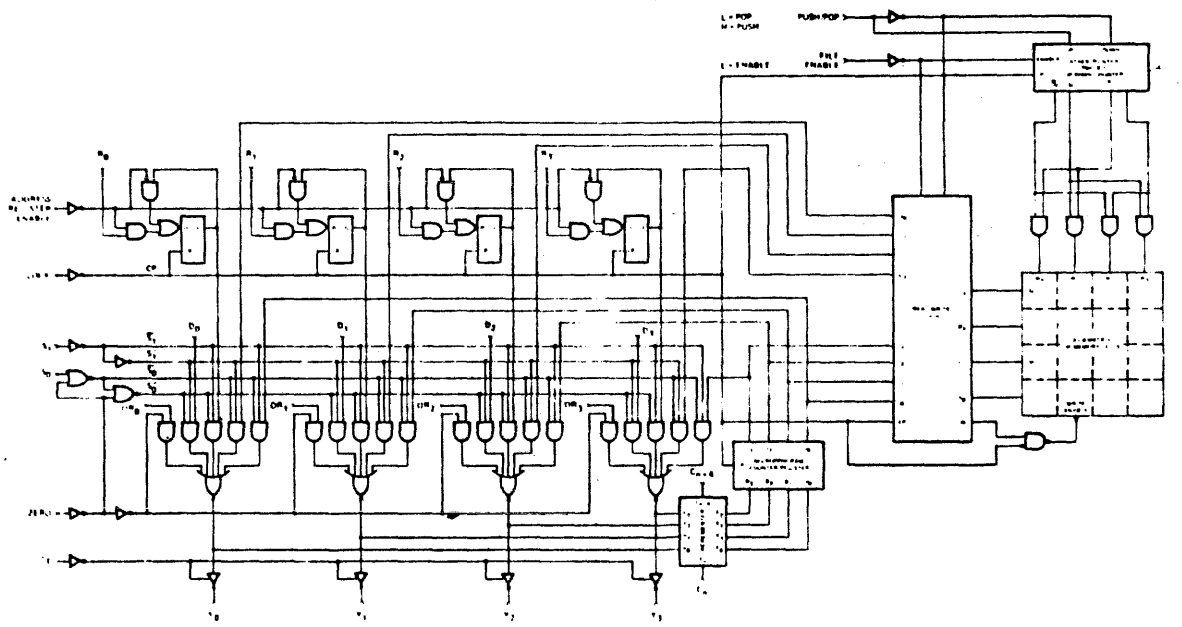
4 BIT SLICE CASCADABLE TO ANY WIDTH

ADDRESS REGISTER
FOUR DEEP PUSH DOWN STACK WITH PUSH/POP CONTROL
MICRO PROGRAM COUNTER REGISTER
ADDRESS INCREMENTER

DIRECT ADDRESS REGISTER INPUT
MULTIPLEXER DATA INPUT
OR'ED DATA INPUT
28 PIN PACK

AMD 2911 SAME EXCEPT NO OR'ED INPUTS. DATA AND ADDRESS INPUTS SHARED
20 PIN PACK

AMD 2909 MICROPROGRAM SEQUENCER



AMD 2909 MICROPROGRAM SEQUENCER CONTROL

THE SEQUENCER ALSO CONTROLS THE MICROPROGRAM TO ITSELF.

MICROSEQUENCER CONTROL INPUTS

S_1 S_0	ADDRESS SOURCE SELECTION
0	μPC
1	AR
2	STKØ
3	D_i

$\overline{FE}$, PUP	STACK CONTROL
1 –	NO CHANGE
0 1	INCREMENT STKPTR, PUSH μPC → STKØ
0 0	DECREMENT STKPTR (POP)

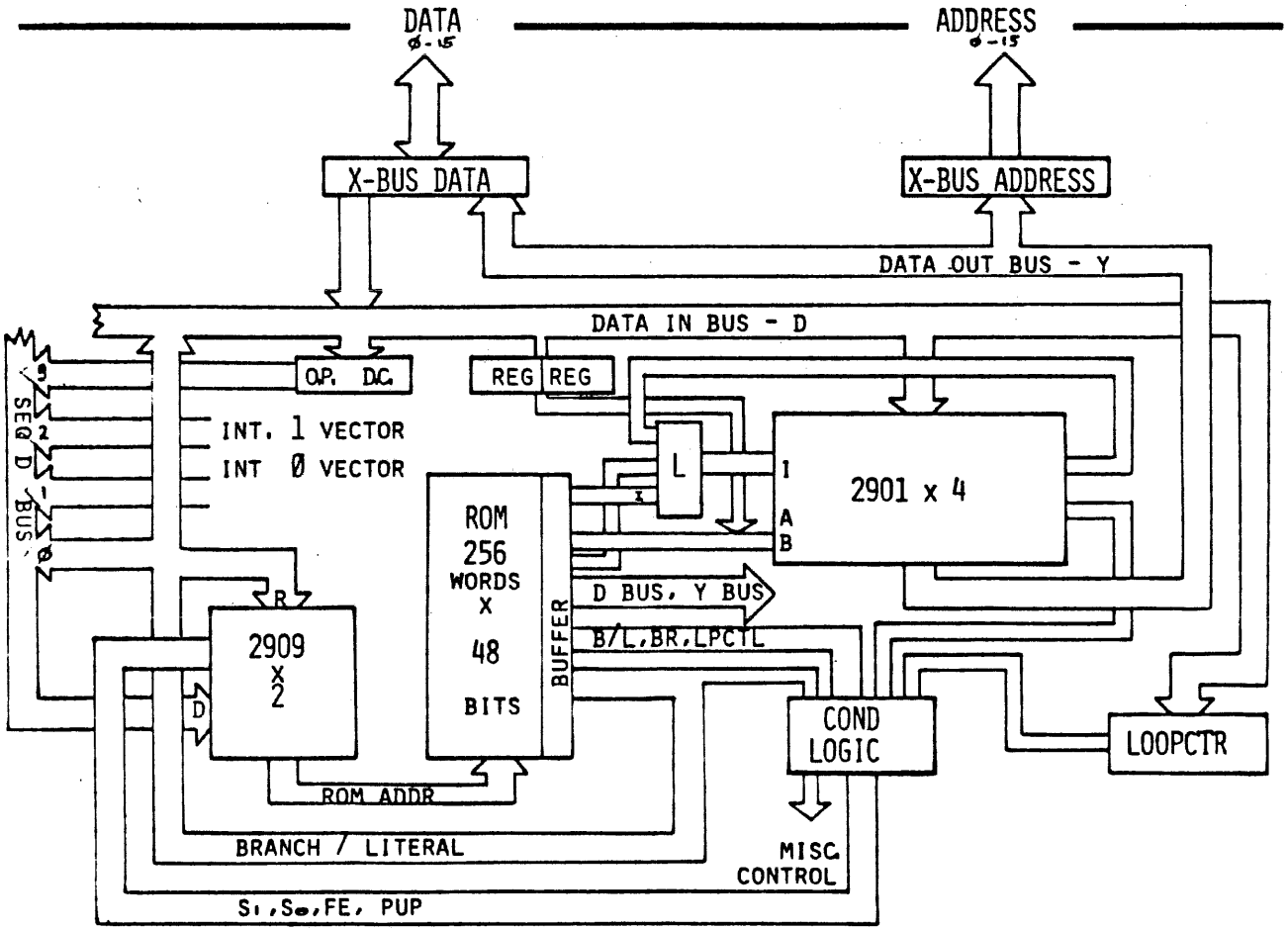
$\overline{RE}$ ENABLE ADDRESS REGISTER

AMD 2901, 2909 MICROPROGRAM INSTRUCTION

EVERY MICROINSTRUCTION MUST CONTAIN:

1. BIT SLICE ARRAY CONTROL
2. ARRAY ANCILLARY LOGIC CONTROL
   - SHIFT END AROUND
   - MULTIPLY TEST ENABLE
3. MICRO SEQUENCER CONTROL
4. MICROPROGRAM BRANCH ADDRESSING
5. SEQUENCER ANCILLARY LOGIC CONTROL
   - CONDITIONAL TESTS
   - MICRO LOOP COUNTER
6. PROCESSOR BUS CONTROL
   - MEMORY
   - I/O
   - INSTRUCTION DECODE

MICROPROGRAMMED 16-BIT PROCESSOR WITH AMD 2901/2909



BLANK

SELECTING MICROPROGRAM STORAGE - ROM

TWO CRITERIA

1. SIZE
2. SPEED

OTHER CRITERIA

1. PROM AVAILABLE
2. SECOND SOURCE

SIZE: EXAMPLE

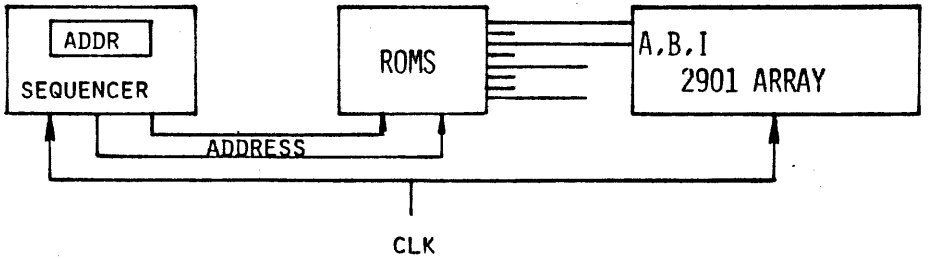
IF TOTAL ROM SPACE REQUIRED 256 WORDS x 48 BITS
CHOICE 256 x 4 TWELVE REQUIRED
256 x 8 SIX REQUIRED (HALF BOARD SPACE)
BUT SLIGHT TIME PENALTY

ROM SELECTION AND USE

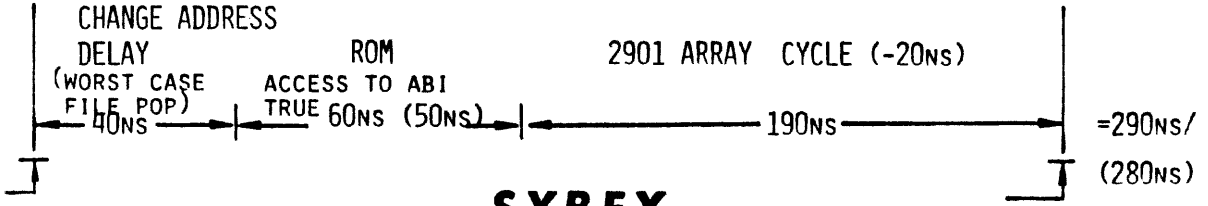
SPEED OF CHIP

ROMS/	SIGNETICS 825229 256x4	TA = 50NS
	SIGNETICS 825214 256x8	TA = 60NS
PROMS/	SIGNETICS 825129 256x4	TA = 50NS
	SIGNETICS 825114 256x8	TA = 60NS

SPEED IN SYSTEM (SERIAL)



± FROM COMPLETION OF CYCLE



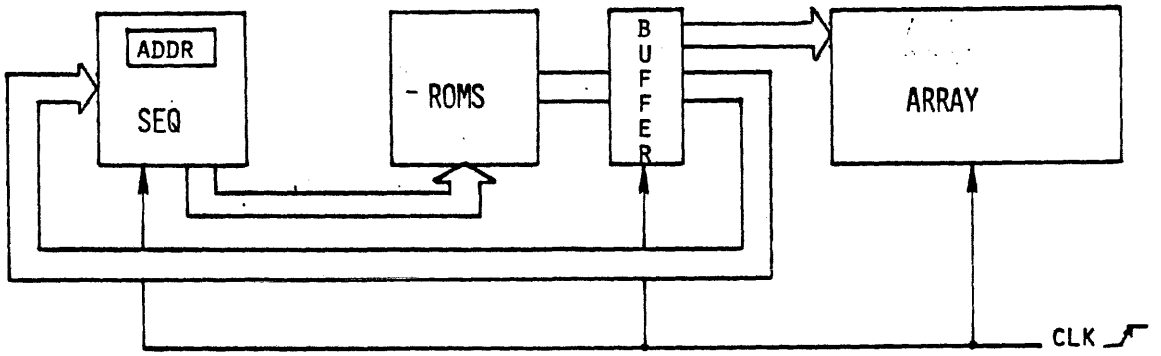
BUFFERED OPERATION (PIPELINING)

INSTEAD OF SERIAL OPERATION OF THE SEQ→ROM→ARRAY

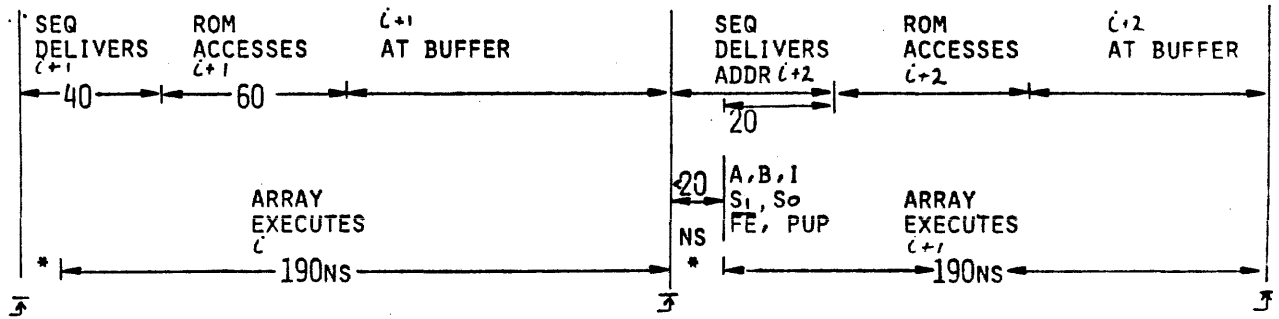
BUFFERED OPERATION

ARRAY AND SEQ/ROM CONTROL IN PARALLEL

ARRAY LAGS ONE STEP BEHIND SEQ/ROM



BUFFERED SYSTEM TIMING



* THROUGH BUFFER DELAY ~15NS (AM 2918) / ALLOW 20NS

ADVANTAGES

- 210NS VERSUS 290NS
- NON-CRITICAL SEQUENCER AND ROM TIMINGS
  - ALLOWS MORE LOGIC IN CONTROL CIRCUITS
  - ALLOWS SLOWER ROMS

DISADVANTAGES

- REQUIRES BUFFER
- "INITIATE/EXECUTE OFFSET"

INITIATE/EXECUTE OFFSET PROBLEMS

THE BUFFERED ARCHITECTURE REQUIRES

SEQUENCER/ROM TO INITIATE $i+1$ WHILE ARRAY IS EXECUTING

(NOTE: THE MICRO-INSTRUCTION i IN THE BUFFER IS CONDITIONING THE SEQUENCER DURING THE INITIATION OF $i+1$.)

CONSEQUENCES:

1. IF THE OUTCOME OF INSTRUCTION i MUST CONDITION $i+1$ ADDRESS
   THEN $i+1$ MUST BE DELAYED ONE CYCLE
2. IF A BRANCH IS TO BE EXECUTED IN CYCLE AFTER INSTRUCTION i
   THEN THE NEW ADDRESS MUST BE AVAILABLE TO SEQUENCER AT BEGINNING OF i
3. IF INSTRUCTION i IS LAST IN A LOOP
   THEN LOOP COUNTER MUST BE DECREMENTED DURING $i-1$

CONDITION TESTS - FOR 2901/2909 CPU

CONDITIONS TO BE TESTED		TIME AVAILABLE
FROM ARRAY:	CARRY	END CYCLE i
	OVR	END CYCLE i
	SIGN	END CYCLE i
	F=0	END CYCLE i
FROM LOOP CTR:	CTR=Ø	BEGIN. CYCLE i
OTHERS:	BUS ACCESS	ASYNCHRONOUS
	INTERRUPT	ASYNCHRONOUS

CONDITION MICROINSTRUCTION SEQUENCING

AN "ABORT" FACILITY IS REQUIRED

CYCLE	1	2	3	1′	2′
EXECUTE	i	ABORTED	i(BR)	i	$i+1$
FETCH	$i+1$	i(BR)	i(BR)+I	$i+1$	$i+2$
CONDITION	Ti (→BR)	-	-	OR Fi ¬(→BR)	-
ABORT NEXT CYCLE EXECUTE?	YES	(NO)	(NO)	(NO)	(NO)

NOTE:

CONDITIONS AVAILABLE END CYCLE 1/1′

ABORT DECISION MADE EARLY IN CYCLE 2/2′

EXECUTION OF $i+1$ IN PROGRESS DURING CYCLE 2

EXECUTION ABORT

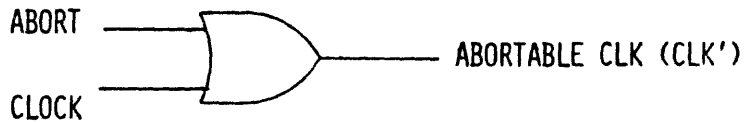
DEFINITION

AN INSTRUCTION IS ABORTED IF PREVENTED
FROM ALTERING REGISTER OR CONDITION STATES.

NOTE: ALL 2901/2909 REGISTERS AND FLAGS ENABLES
ON CLOCK ⌐
(RAM LATCHES MAY BE DISREGARDED)

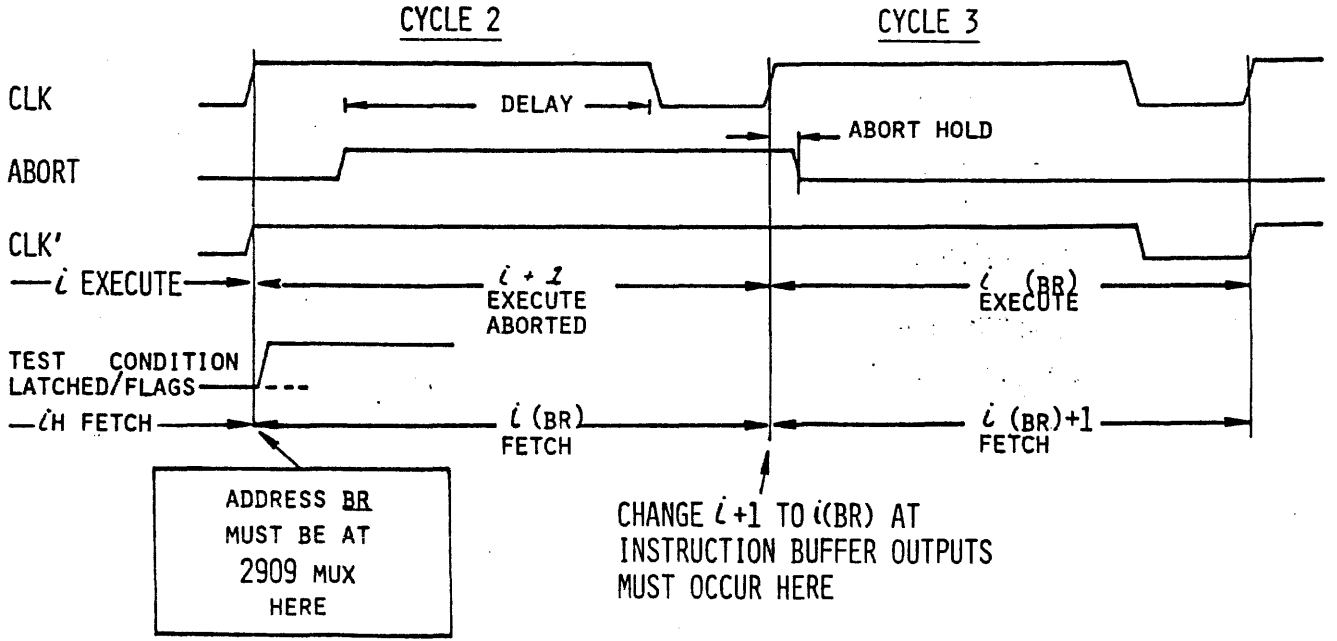
ABORT IS ACCOMPLISHED BY PREVENTING ⌐
FOR GIVEN INSTRUCTION

IMPLEMENTATION



EXECUTION ABORT TIMING

ABORT SIGNAL MUST STABILIZE BEFORE CLK$_{HL}$

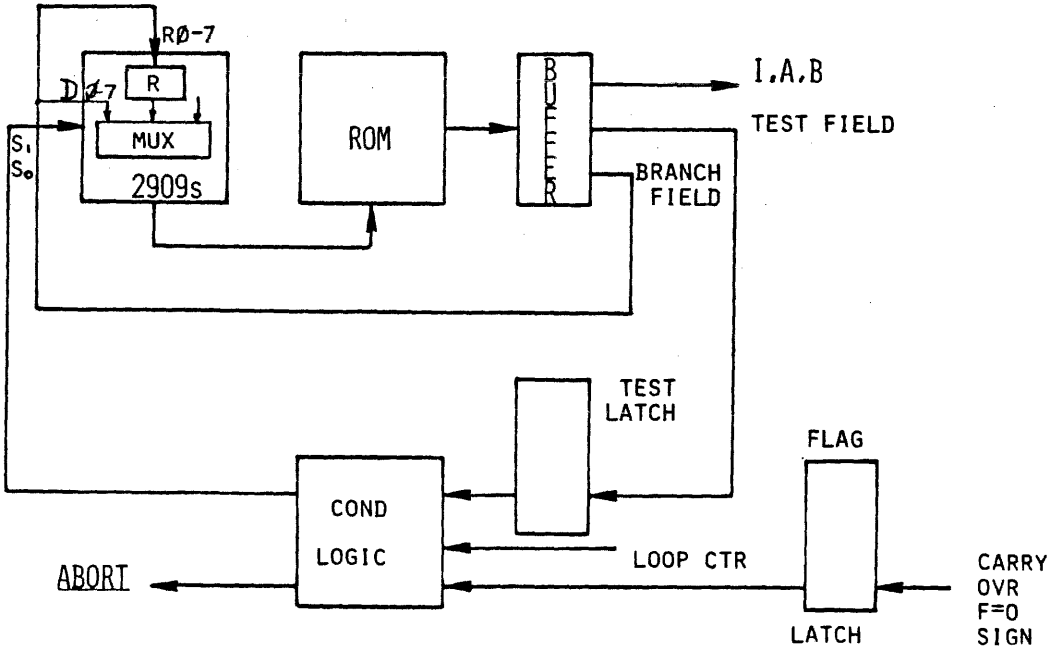


CONDITIONAL TEST IMPLEMENTATION USING 2909

TEST AND BRANCH IN INSTRUCTION i (EXAMPLE P.63)

BRANCH ADDRESS AND TEST MUST BE LATCHED

DURING i + 1. (USE ADR REG IN 2909s FOR BR)



CONDITIONAL TEST IMPLEMENTATIONS USING 2909

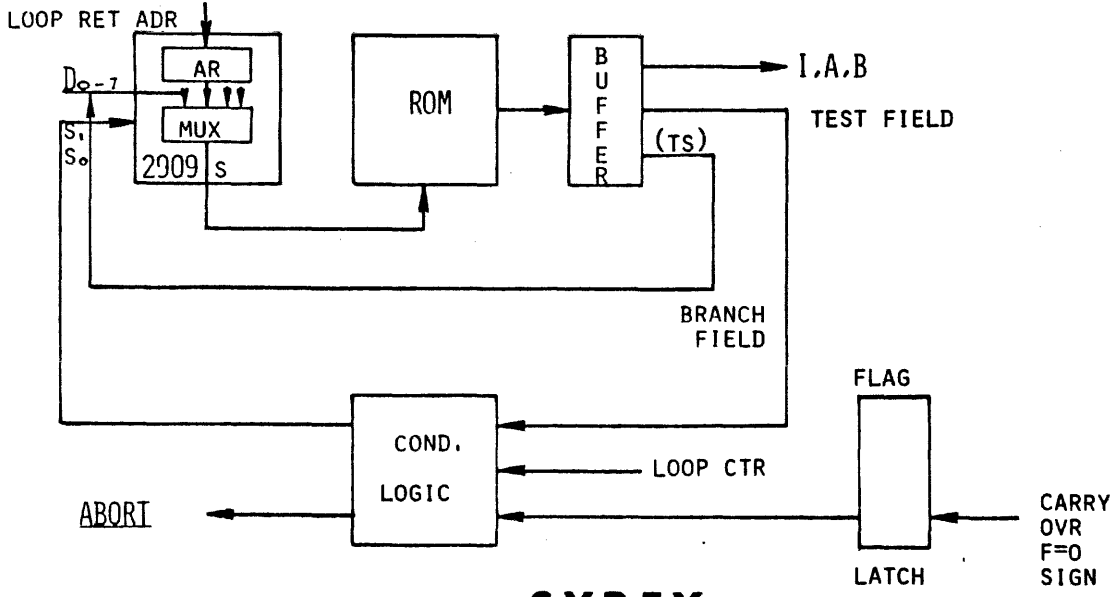
A L T E R N A T I V E

T E S T A N D B R A N C H I N I N S T R U C T I O N i + 1

FLAGS OF INSTRUCTION i ARE LATCHED

TEST FIELD AND BRANCH ADDRESS GATED FROM

BUFFER DURING i + 1 (USE 2909 D_{0-7} INPUTS)



COMPARISON OF BRANCH/TEST IMPLEMENTATIONS

FAR-REACHING IMPLICATIONS:

BRANCH/TEST IN i

ADVANTAGES

- CLEANER, MORE COMPACT PROGRAM
- GREATER PARALLELISM (OVERLAP OF TEST AND EXECUTION OF $i+1$ IF TEST FAILS)
- CLEANER TIMING (BUFFERED ADDRESS LOOP

DISADVANTAGES

- WIDER ROM WORD
- EXTRA CHIP FOR TEST LATCH

BRANCH/TEST IN $i+1$

ADVANTAGES

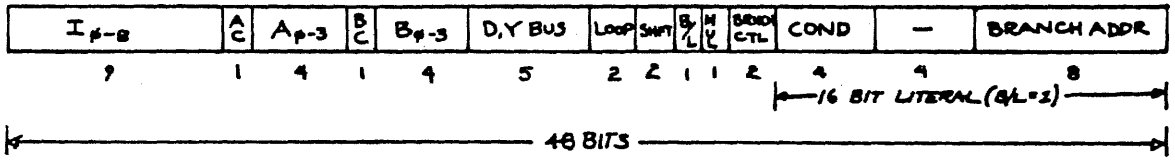
- ENCODED MICROINSTRUCTION SAVES ROM WIDTH
- LEAVES 2909 ADR. REGISTER FREE FOR LOOP IMPLEMENTATION
- LESS HARDWARE (NO TEST LATCHES - ONE CHIP)

DISADVANTAGES

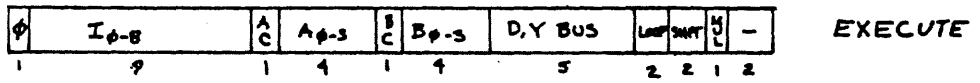
- LESS COMPACT MICROPROGRAM
- LESS PARALLELISM
  1. NO OVERLAP OF TEST
  2. EXTRA INSTRUCTION FOR BRANCHES AND RETURNS
  3. EXTRA STEP FOR INITIALIZING LOOP RETURNS

COMPARISON OF MICRO INSTRUCTION FORMATS

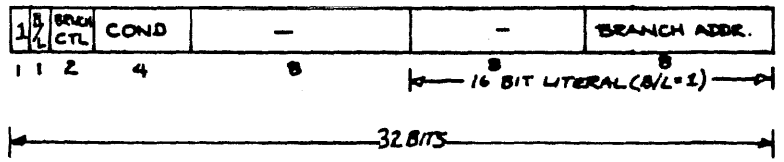
BRANCH IN i



BRANCH / TEST IN i+1



EXECUTE



BRANCH

COMPARISON: SUMMARY

1. BRANCH/TEST IN i

   FASTER BUT MORE EXPENSIVE

   FEWER ROM WORDS/WIDER FIELD

2. BRANCH/TEST IN i+1

   CHEAPER BUT SLOWER

   MORE WORDS/NARROWER FIELD

HOW TO DECIDE

IF SPEED IS CRUCIAL -- CHOOSE 1

IF COST IS CRUCIAL: PROBABLY BIT SLICES AREN'T THE RIGHT TECHNOLOGY

IF SPACE IS CRUCIAL THEN CHOOSE 2

IE THE LACK OF MICROPROGRAM COMPACTNESS DOESN'T FORCE THE ADDITION OF ANOTHER BANK OF ROMS

THE APPLICATION MUST DECIDE

CONDITIONAL LOGIC TIMING REQUIREMENTS

T I M I N G: ASSUME 210 NS CLOCK WITH T_L = 50
ASSUME 20 NS CLK·ABORT → CLK'

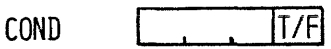
C O N D I T I O N A L L O G I C M U S T

1. GENERATE ABORT SIGNAL BEFORE CLK ↓
   210 NS - T_L - DELAY = 140 NS
2. GENERATE S S_0 TO SELECT BRANCH ADDRESS
   IN TIME FOR: ROM OUT → BUFFER AT ↑
   210 NS - (SEQUENCER DELAY + ROM DELAY)
   (S_1 S_0 → Y_i)
   210 NS - (20 NS + 60 NS) = 130 NS

ASSUME 20 NS LATCH/BUFFER DELAY AT START OF CYCLE
MAX CONDITIONAL LOGIC DELAY = 110 NS

CONDITIONAL TESTS

MICROINSTRUCTION FIELD



COND	
Ø	TRUE (UNCONDITIONAL)
1	CARRY = 1
2	OVR = 1
3	F_{15} = 1
4	$F_{Ø-15}$ = Ø
5	$F_{Ø-7}$ = Ø (LOWER BYTE)
6	F_{8-15} = Ø (UPPER BYTE)
7	BUS

T/F	
Ø :	BRANCH IF CONDITION TRUE
1 :	BRANCH IF CONDITION FALSE

BRANCHING OPERATIONS DEFINITION

ASSUME BRANCH/TEST IN INSTRUCTION (48 BIT IMPLEMENTATION)

I - FIELD B/L = Ø (BRANCH)

BR: [,]	SYMBOL	2909 ACTION	2909 PINS S_1	S_0	FE	PUP	ABORT
0	-	μPC TO ROM ADR	0	0	1	x	NO
1	BR	AR TO ROM ADR	0	1	1	x	YES
2	PUSH BR	AR TO ROM ADR μPC TO STK	0	1	0	1	YES
3	POPRET	STKØ TO ROM ADR	1	0	0	0	YES
PLUS LOOP RETURN (SPECIFIED IN LP FIELD)							
-	LPRET	BRANCH FIELD/D_{0-7} TO ROM ADR	1	1	1	x	NO!
AND INTERRUPT (AUTOMATED)							
-	INTERRUPT	INT.ADR/D_{0-7} TO ROM ADR μPC TO STK	1	1	0	1	NO

PRIORITY : LOOP RETURN > LITERAL/BROPS > INTERRUPTS

LOOP RETURN MICRO COMMANDS

LR [LD|TD] LD: LOAD LOOP COUNTER 0 = NOP/1 = LOAD

TD: TEST AND DECREMENT 0 = NOP/1 = TEST LPCTR≠Ø ?
BRANCH IF TRUE
DECREMENT IF TRUE
NOP IF FALSE

TIMING:

UNLIKE ALU TESTS, LOOP CTR ≠ Ø IS AVAILABLE AT START OF CYCLE

BRANCH ADDRESS IN INSTRUCTION BUFFER DURING

1. GATED INTO 2909 D INPUT IF COUNTER ≠ Ø
2. IGNORED OTHERWISE

NO ABORT IS REQUIRED

A CYCLE IS SAVED IN EACH ITERATION

LOOP RETURN TIMING

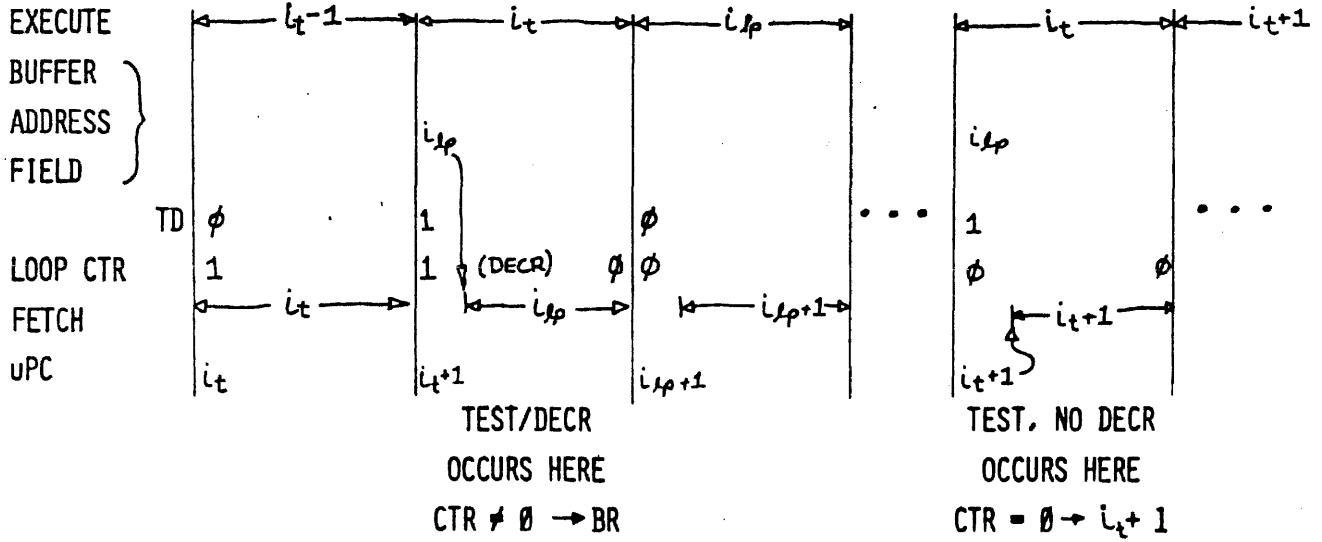
ASSUME A MICROPROGRAM LOOP

i_{lp} /FIRST INSTRUCTION

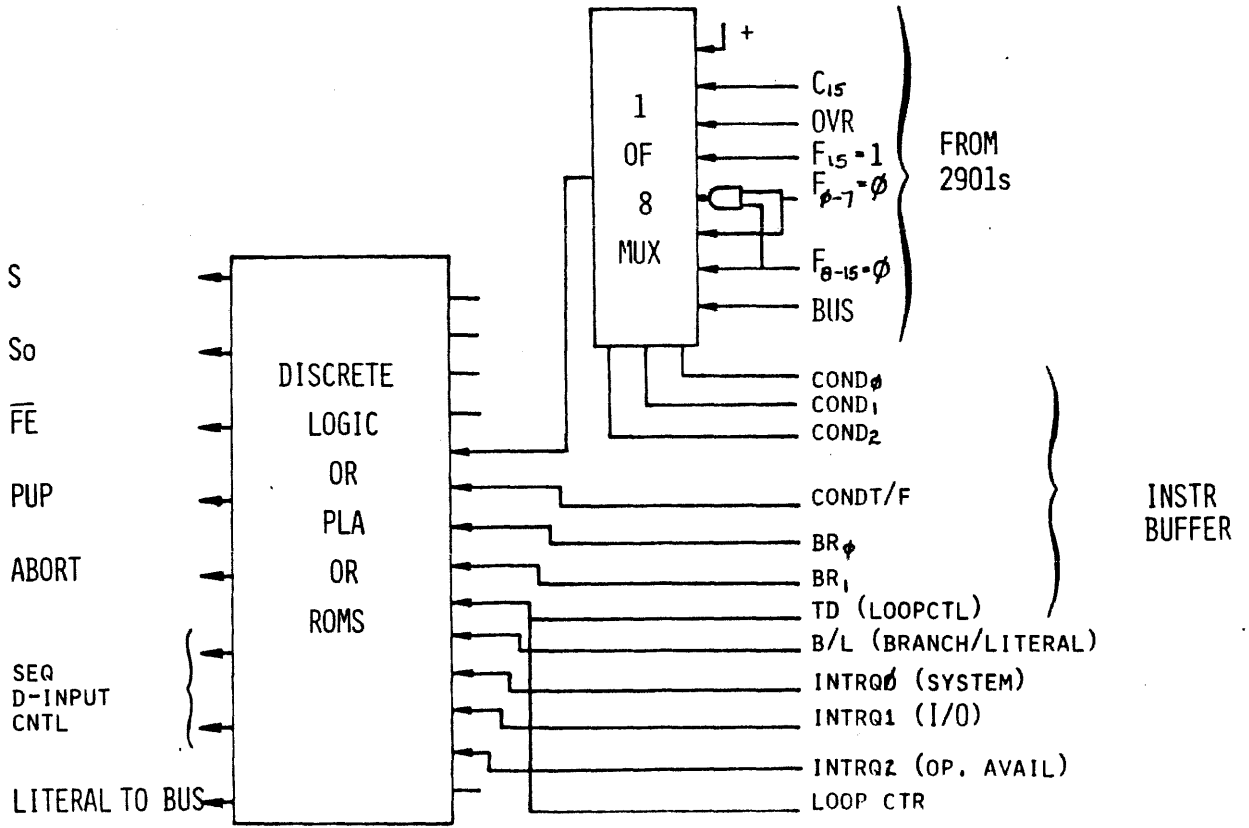
⋮

i_t /... TD = 1, BR. ADDR= i_{lp} /END OF LOOP

i_{t+1} /NEXT INSTRUCTION



BRANCH/TEST CONTROL LOGIC



BRANCH LOGIC IMPLEMENTATION

THE LOGIC IN THE BOX CAN BE PERFORMED BY

8 CHIPS DISCRETE LOGIC
OR 1 ROM 512 x 8
OR 1 PLA 14(16) x 48 x 8

COMPARISON

METHOD	SPEED	PINS*	DISSIPATION	COST	SPACE	INPUT EXPANSION
8 CHIPS L.S.	40 NS	8 x 16 = 128	8 x 50 MW = 400 MW	$8	4 IN^2	NO
1 ROM 512 x 8	60 NS	24	700 MW	$15	1 IN^2	NO
1 PLA (48)	50 NS	24/28	600 MW	PROM ($33) $20?**	1 IN^2	YES (5 SIGNALS)

*RELIABILITY AND BOARD COMPLEXITY

**HIGHLY UNPREDICTABLE

P.L.A. REVIEW

P.L.A. IS A THREE STAGE ARRAY

FUNCTION STAGE 1: PRODUCTS OF INPUT TERMS

$$P_n = K_{n,\phi} \cdot i_\phi \wedge K_{n,1} \cdot i_1 \wedge K_{n,2} \cdot i_2 \wedge \ldots$$

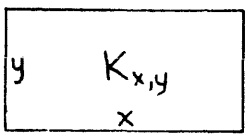
WHERE $K_{n,m} = 1, \phi, -1$ SUCH THAT $1 \cdot i = i$

$\phi \cdot i =$ UNDEFINED (I.E. NO TERM)

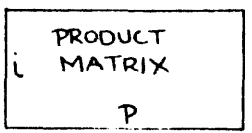
$-1 \cdot i = \overline{i}$

$n = 0, 1, 2, \ldots$

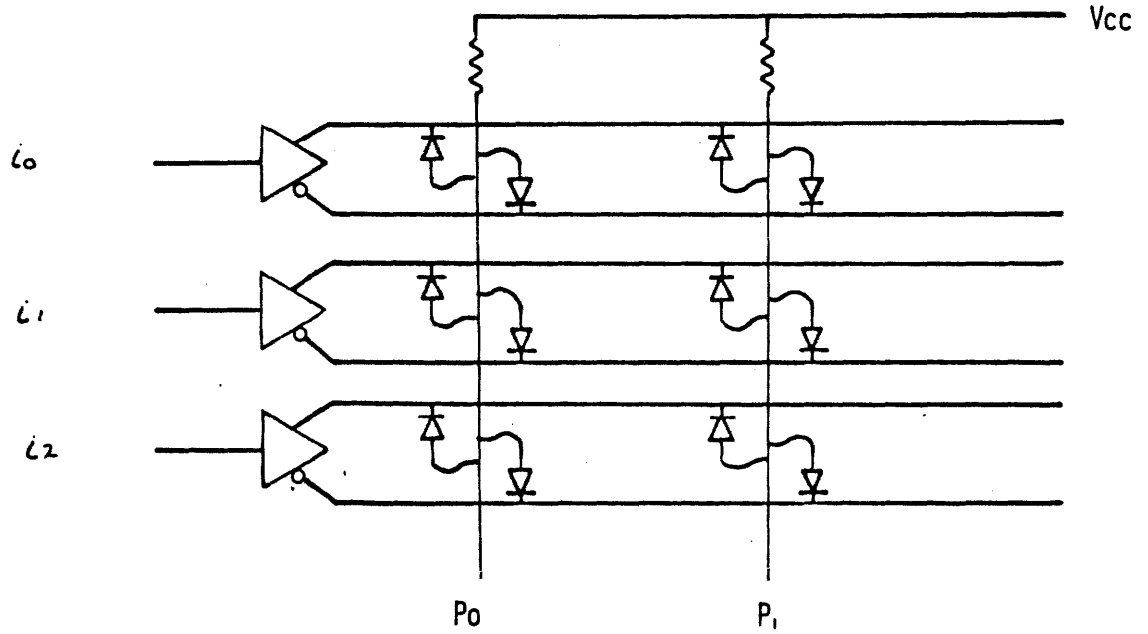
K IS A MATRIX



WHICH MAPS ONTO



CIRCUIT OF PRODUCT (STAGE 1) MATRIX



AT EACH INTERSECTION AT LEAST ONE FUSE MUST BE BLOWN

STANDARD INPUTS i_y :14 OR16

PRODUCTS Px:48 OR 96

PLA STAGE 2: SUMMATION

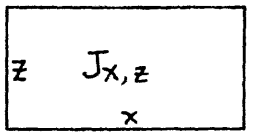
FUNCTION STAGE 2 : SUMMATION OF PRODUCT TERMS

$$S'_r = J_{r,\phi} \cdot P_\phi \vee J_{r,1} \cdot P_1 \vee J_{r,2} \cdot P_2 \vee \ldots\ldots$$

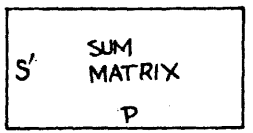
WHERE $J_{r,s} = 1, \emptyset$ SUCH THAT $1 \cdot P = P$

$\emptyset \cdot P$ = UNDEFINED (I.E. NO TERM)

J IS A MATRIX



WHICH MAPS ONTO

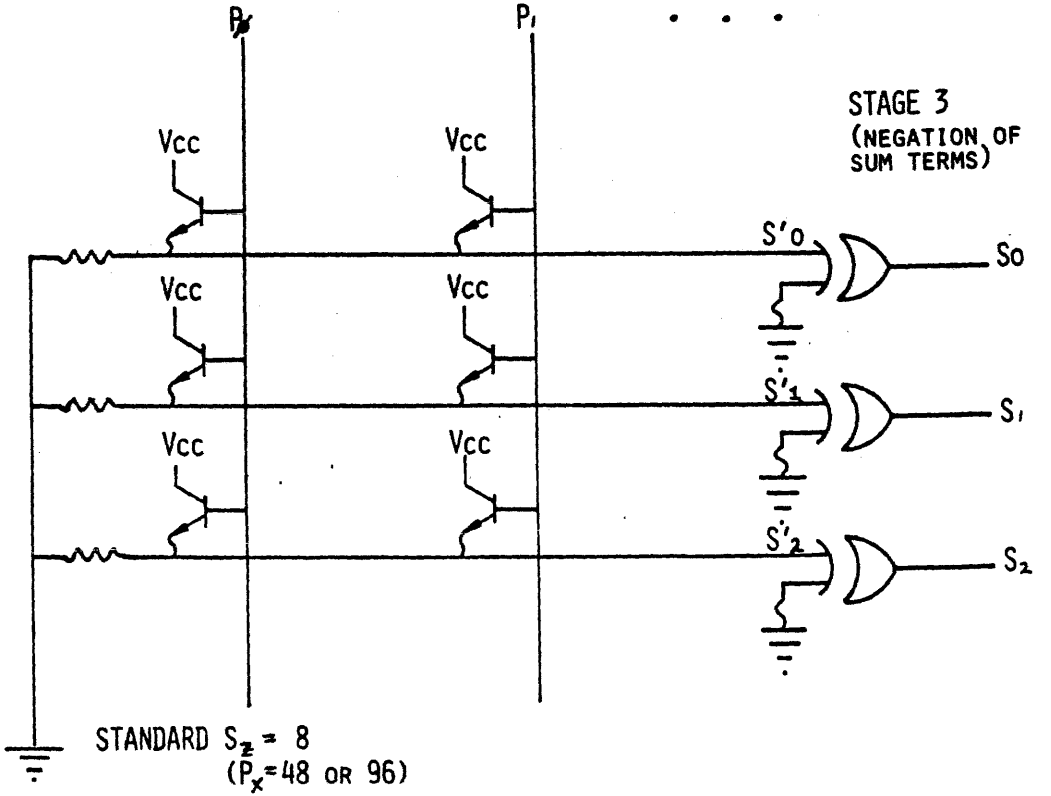


NOTE: STAGE 3 FUNCTION -- NEGATION OF SUM TERMS

$S_r = L_r \; S'_r$ WHERE $L_r = 1, -1$ SUCH THAT $1 \; S' = S'$

$-1 \; S' = \overline{S'}$

CIRCUIT OF SUMMATION (STAGE 2) MATRIX



PLA CHARACTERISTICS

THREE STANDARD CONFIGURATION

14 INPUTS 48 PRODUCTS 8 OUTPUTS (PLA, FPLA)
16 INPUTS 48 PRODUCTS 8 OUTPUTS (PLA, FPLA)
14 INPUTS 96 PRODUCTS 8 OUTPUTS (PLA ONLY)

AVAILABLE OC OR TRI-STATE OUTPUT, OUT ENABLE

RANGE OF SPEED 50 NS - 100 NS

DISSIPATION: EXAMPLE (SIGNETICS 82S101 16 x 48 x 8 FPLA 50 NS)
600 mW

24 OR 28 PIN PACKS

PROGRAMMING: SPECIFY J,K,L MATRICES BY PAPERTAPE OR CARDS
PLA: FACTORY MASKED
FPLA: FIELD PROGRAMMABLE WITH FUSE BLOWER

PLA MASK CODING: CARDS OR TAPE

PROGRAM TABLE OF SAMPLE DEVICE

CUSTOMER NAME ADVANCED LOGIC DE :TED BY SIGNETICS
PURCHASE ORDER # NONE (FREE SAN EX-OR, MUX, ADD
SIGNETICS DEVICE # 82S100I101
TOTAL NUMBER OF PARTS 1 IEN YOU LIKE
PROGRAM TABLE # REV OSE LOGIC APPLICATIONS

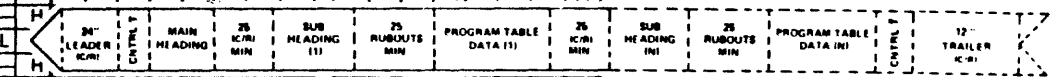
PROGRAM TABLE ENTRIES

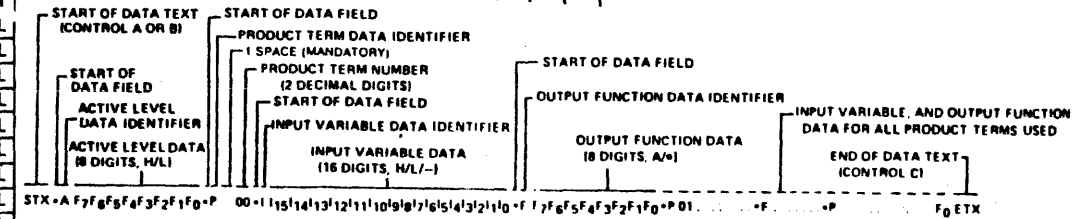
INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
I_m	$\overline{I}_m$	DON'T CARE	PROD. TERM PRESENT IN F_θ	PROD. TERM *NOT* PRESENT IN F_θ	ACTIVE HIGH	ACTIVE LOW
H	L	– (dash)	A	• (period)	H	L
NOTE Enter (–) for *unused* inputs of *used* P-terms			NOTES 1) Entries independent of output polarity 2) Enter (A) for *unused* outputs of *used* P-terms		NOTES 1) Polarity programmed once only 2) Enter (H) for all *unused* outputs	

PRODUCT TERM* NO.	INPUT VARIABLE 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ACTIVE LEVEL L	L	L	L	H	H	H	H	OUTPUT FUNCTION* 7	6	5	4	3	2	1	0
0	—	—	—	—	—	L	—	H	—	—	—	—	—	—	—	H									•	•	•	A	•	•	•	A
1	—	—	—	—	—	L	L	H	—	—	—	H	—	—	—	—									•	•	•	A	•	•	•	A
2	—	—	—	—	—	L	—	H																								
3	—	—	—	—	—	L	L																									
4	—	—	—	—	—	L	—	H																								
5	—	—	—	—	—	L	L	H	—	H	—	—	—	—	—	—									•	A	•	•	•	A	•	•
←	—	—	—	—	—	L	—	H	—	—	—	—	H	—	—	—									A	•	•	•	A	•	•	•
30	—	—	—	—	—	L	L	H	H	—	—	—	—	—	—	—									A	•	•	•	A	•	•	•
31	—	L	—	—	—	L	L	L	—	—	—	H	—	—	—	H									•	•	•	A	•	•	•	A
32	—	L	—	—	—	·	·																									
33	—	L	—	—	—	H	L																									
34	—	—	—	—	H	H	L																									
35	—	—	—	—	H	H	L																									
36	—	—	—	L	—	H	L																									
37	—	—	—	L	—	H	L																									
38	—	—	L	—	—	H	L																									
39	—	—	L	—	—	H	L																									
40	—	—	—	—	—	H	L																									
41	—	—	—	—	—	H	L																									
42	—	—	—	—	—	H	L																									
43	—	—	—	—	—	H	L																									
44																																
45																																
46																																
47																																

*Input and Output fields of *unused* P-terms can be left

TWX TAPE CODING FORMAT

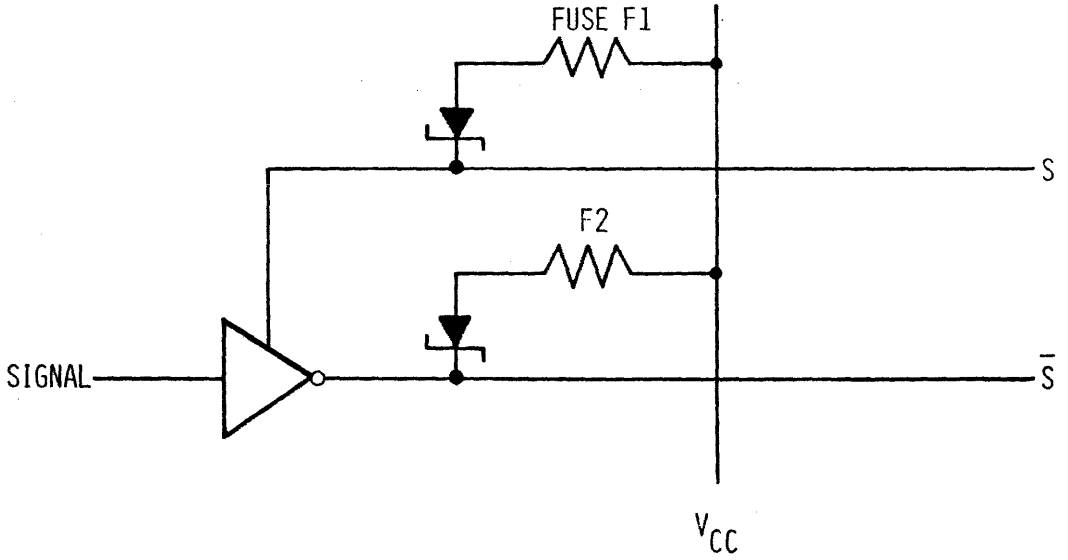




Entries for the 3 Data Fields are determined in accordance with the following Table:

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
I_m	$\overline{I}_m$	DON'T CARE	PROD. TERM PRESENT IN F_θ	PROD. TERM *NOT* PRESENT IN F_θ	ACTIVE HIGH	ACTIVE LOW
H	L	– (dash)	A	• (period)	H	L
NOTE Enter (–) for *unused* inputs of *used* P-terms			NOTES 1) Entries independent of output polarity 2) Enter (A) for *unused* outputs of *used* P-terms		NOTES 1) Polarity programmed once only 2) Enter (H) for all *unused* outputs	

FPLA: FUSIBLE LINK



EITHER FUSE F1 OR F2 IS BLOWN

A TYPICAL FPLA TRUTH TABLE


```
STX                                    *A  LLLLLLLL

*P  00  *I  LL---L-------H  *F  -----AA-
*P  01  *I  LLLLLH--------  *F  A----AAA
*P  02  *I  HLLLLH--------  *F  AA---AA-
*P  03  *I  HHLLLH--------  *F  -A---A--
*P  04  *I  LHLLLH------H-  *F  --A--AA-
*P  05  *I  LHLLLH-----HL-  *F  ---A-AA-
*P  06  *I  LHLLLH----HLL-  *F  --AA-AA-
*P  07  *I  LHLLLH---HLLL-  *F  ----AAA-
*P  08  *I  LHLLLH--HLLLL-  *F  --A-AAA-
*P  09  *I  LHLLLH-HLLLLL-  *F  ---AAAA-
*P  10  *I  LHLLLHHLLLLLL-  *F  --AAAAA-
*P  11  *I  LHLLLHLLLLLLL-  *F  --------
*P  12  *I  LL---L------HL  *F  --A--AA-
*P  13  *I  LLHLLH--------  *F  A-A--AAA
*P  14  *I  HLHLLH--------  *F  AAA--AA-
*P  15  *I  HHHLLH--------  *F  -AA--A--
*P  16  *I  LHHLLH-----H--  *F  ---A-AA-
*P  17  *I  LHHLLH-----HL--  *F  --AA-AA-
*P  18  *I  LHHLLH---HLL--  *F  ----AAA-
*P  19  *I  LHHLLH--HLLL--  *F  --A-AAA-
*P  20  *I  LHHLLH-HLLLL--  *F  ---AAAA-
*P  21  *I  LHHLLHHLLLLL--  *F  --AAAAA-
*P  22  *I  LHHLLHLLLLLL-H  *F  -----AA-
*P  23  *I  LHHLLHLLLLLL-L  *F  --------
*P  24  *I  LL---L-----HLL  *F  ---A-AA-
*P  25  *I  LLLHLH--------  *F  A--A--AAA
*P  26  *I  HLLHLH--------  *F  AA-A-AA-
*P  27  *I  HHLHLH--------  *F  -A-A-A--
*P  28  *I  LHLHLH----H---  *F  -AA-AA-
*P  29  *I  LHLHLH--HL----  *F  ---AAA-
*P  30  *I  LHLHLH--HLL---  *F  --A-AAA-
*P  31  *I  LHLHLH-HLLL---  *F  ---AAAA-
*P  32  *I  LHLHLHHLLLL---  *F  --AAAAA-
*P  33  *I  LHLHLHLLLLL--H  *F  -----AA-
*P  34  *I  LHLHLHLLLLL-HL  *F  --A--AA-
*P  35  *I  LHLHLHLLLLL-LL  *F  --------
*P  36  *I  LL---L----HLLL  *F  --AA-AA-
*P  37  *I  LLHHLH--------  *F  A-AA-AAA
*P  38  *I  HLHHLH--------  *F  AAAA-AA-
*P  39  *I  HHHHLH--------  *F  -AAA-A--
*P  40  *I  LHHHLH---H----  *F  ----AAA-
*P  41  *I  LHHHLH--HL----  *F  --A-AAA-
*P  42  *I  LHHHLH-HLL----  *F  ---AAAA-
*P  43  *I  LHHHLHHLLL----  *F  --AAAAA-
*P  44  *I  LHHHLHLLLL---H  *F  -----AA-
*P  45  *I  LHHHLHLLLL--HL  *F  --A--AA-
*P  46  *I  LHHHLHLLLL--LL  *F  ---A-AA-
*P  47  *I  LHHHLHLLLL-LLL  *F  --------

                                       ETX
```

FPLA'S

	INPUTS	PRODUCT TERMS	OUTPUTS	PINS
A M D	16	48	8	28
INTERSIL	14	48	8	24
MMI	14	48	8	28
SIGNETICS	16	48	8	24

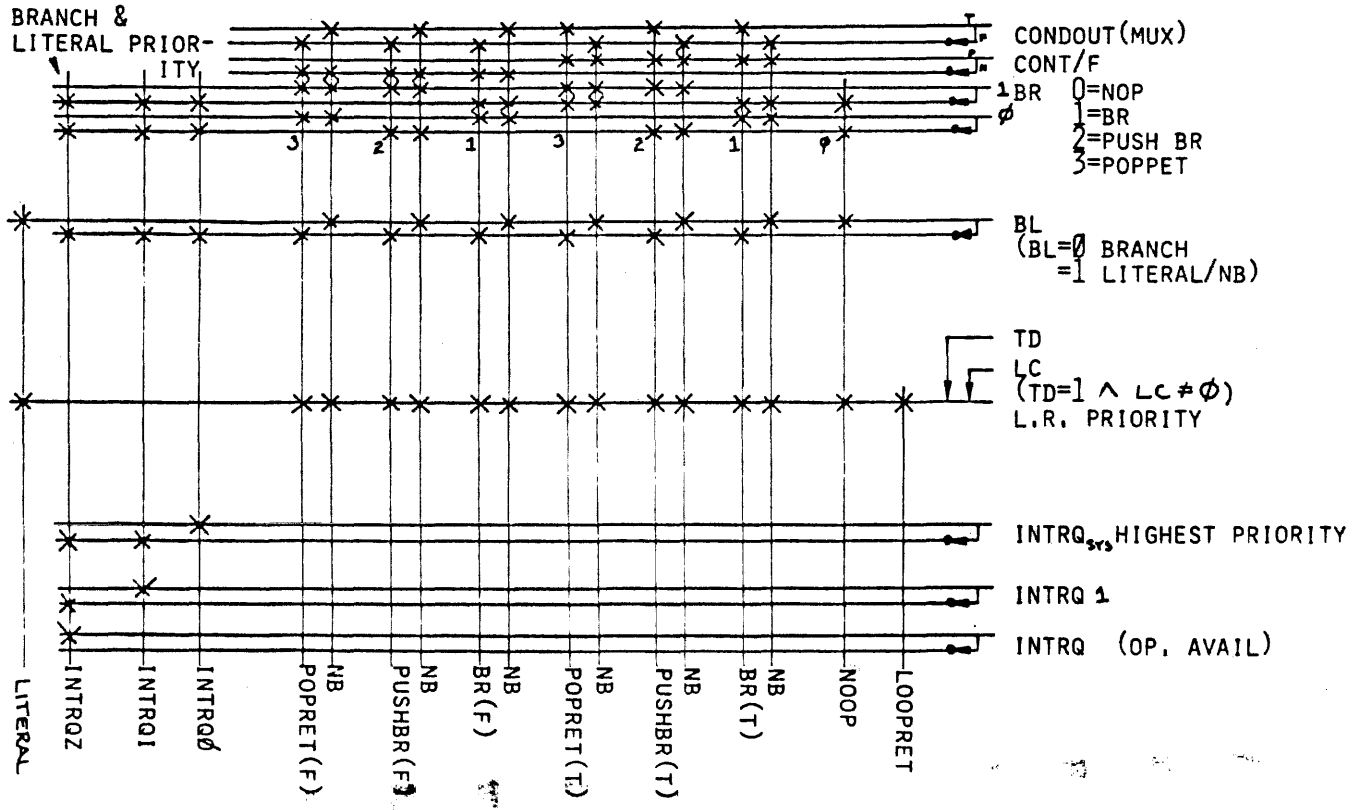
CURRENT PLA'S

- HUGHES
- INTERSIL
- MMI
- NATIONAL SEMICONDUCTOR
- SIGNETICS

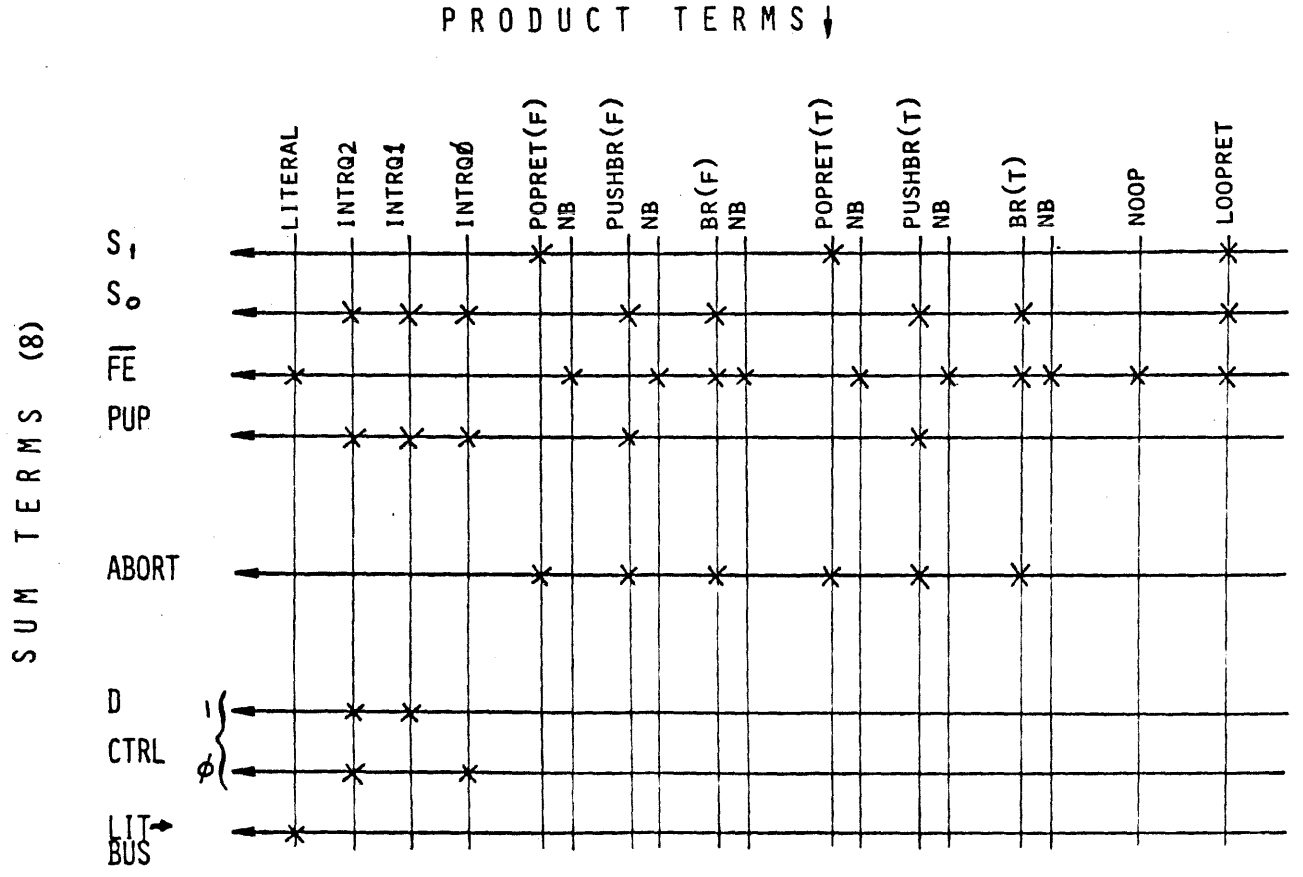
PLA APPLICATIONS

- SEQUENCING
- ALU CONTROL = MICROINSTRUCTION DECODING
- CODE CONVERSION
- ASC II STANDARD INTERFACE BUS (IEEE 488-1975)
- FAST DECODING (50-100 NS)
- ROM PATCHES

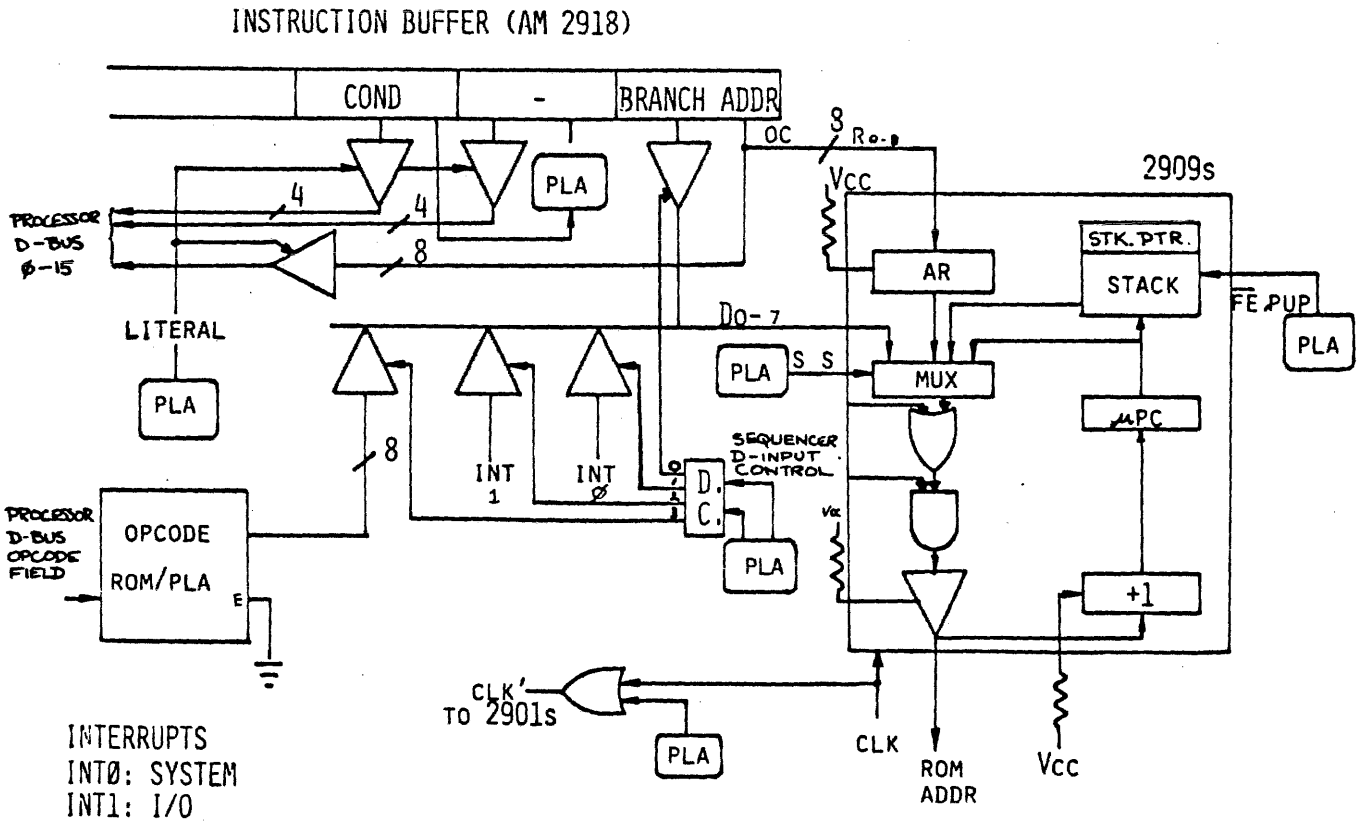
BRANCH CONTROL PLA PRODUCT TERMS



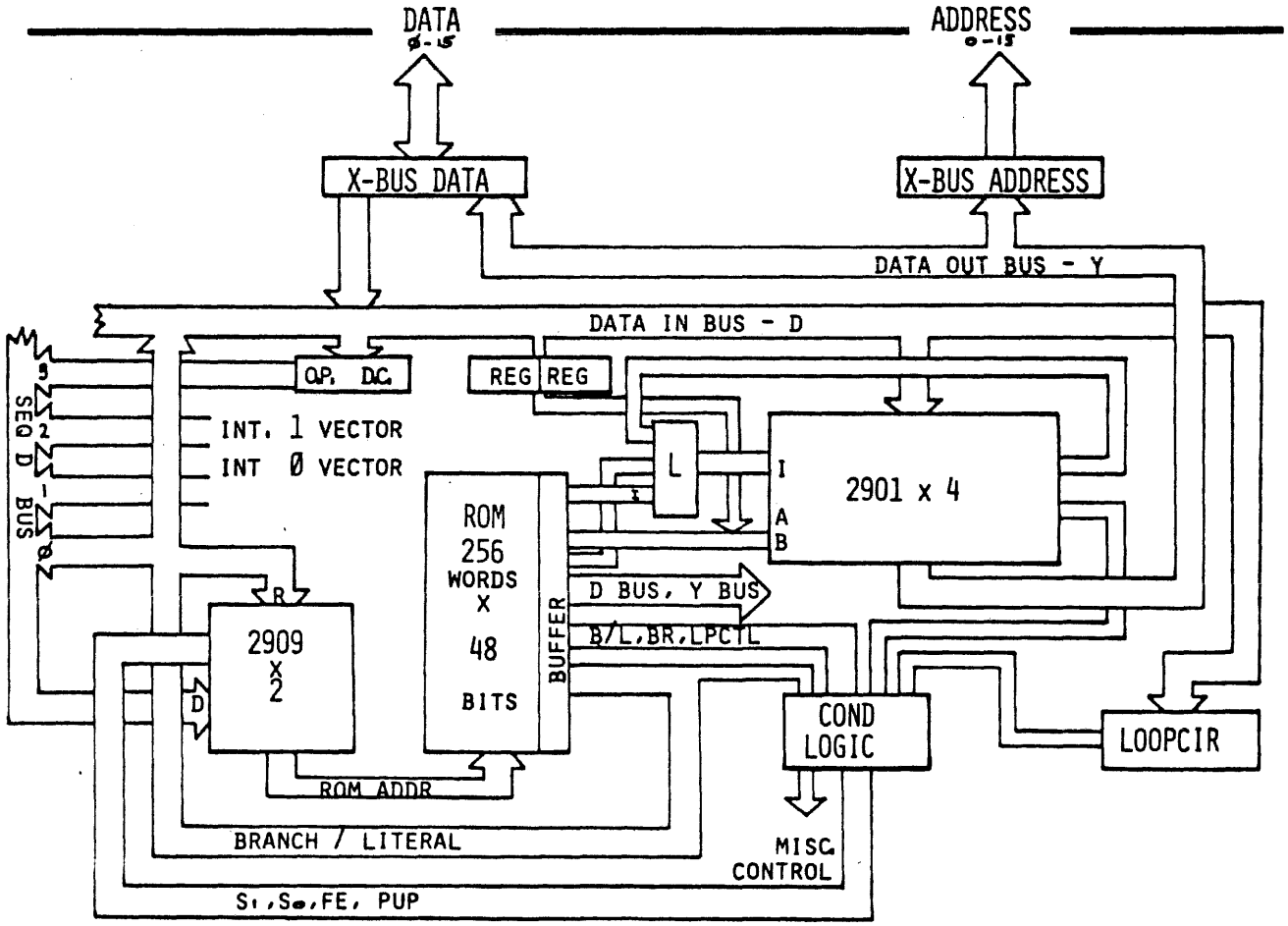
BRANCH CONTROL: PLA SUM TERMS



CONTROL AND DATA TO 2909 SEQUENCER



MICROPROGRAMMED 16 BIT PROCESSOR WITH AMD 2901/2909



BUS CONTROL

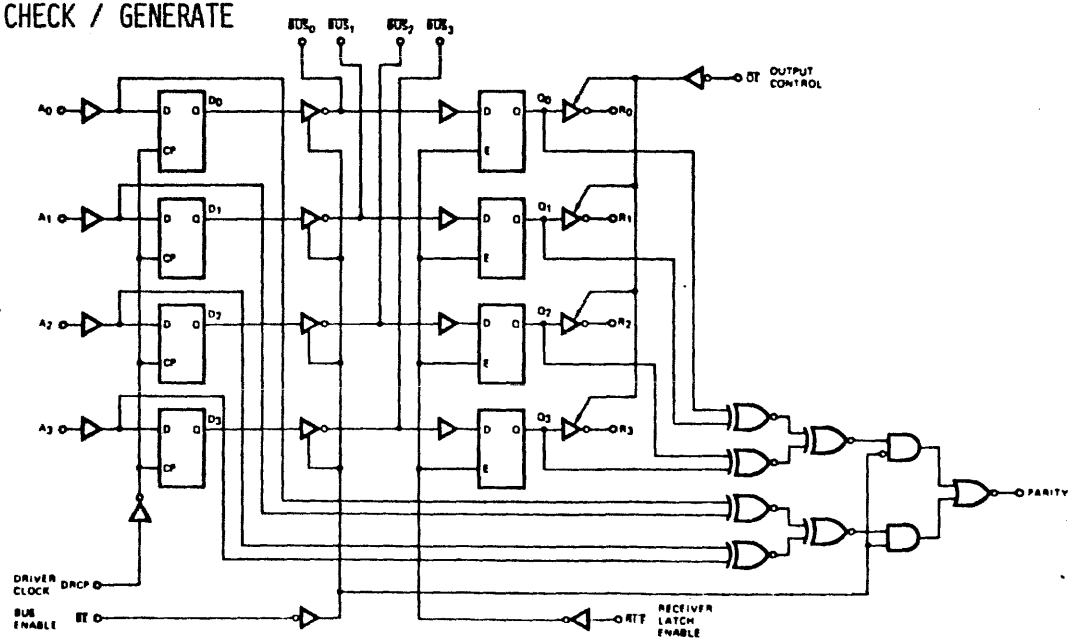
D BUS SOURCES	CONTROLLED BY
LITERAL → D	B/L = 1
X BUS DATA → D	D BUS$_{\phi}$ = 1
D BUS DESTINATIONS	
D → OP DECODE	ALWAYS ENABLED
D → A/B REG. REGS	D BUS$_{1}$ = 1
D → 2901s	$I_{0,1,2}$
D → LOOP CTR	LC_{LD} = 1
Y BUS DESTINATIONS	
Y → X BUS DATA	Y BUS$_{\phi}$ = 1
Y → X BUS ADDR/RD	Y BUS$_{1,2}$ = 01
Y → X BUS ADDR/WR	Y BUS$_{1,2}$ = 10
Y → X BUS ADDR/OPR *	R BUS$_{1,2}$ = 11

*NOTE: CAUSES PAUSE → INT3 WITH OP

D TO X - BUS LOGIC: DATA TRANSCEIVER

AM 2907 QUAD BUS TRANSCEIVER
TRI STATE RECEIVED OUTPUTS
LATCHED OPERATION
PARITY CHECK / GENERATE

LS 400 MW / 35 NS
20 PIN PACK



SYBEX

INT 1: I/O INTERRUPT GENERATION

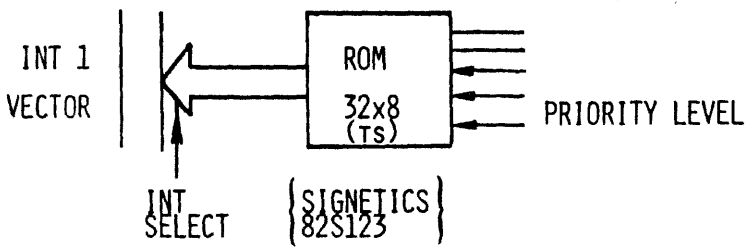
INT1 IS GENERATED BY I/O REQUESTS

WHEN INT1 IS ACKNOWLEDGED THE INTØ VECTOR IS GATED TO SEQUENCER D INPUTS, μPC IS STACKED

THE INTERRUPT VECTOR IS THE ADDRESS OF THE MICROPROGRAM SEQUENCE TO SERVICE THE INTERRUPT

SINGLE LEVEL INTERRUPT SIMPLY FORCES FIXED ADDRESS

PRIORITIZED INTERRUPTS MUST BE SORTED
VECTORS MUST BE GENERATED



VECTORED PRIORITY INTERRUPT ENCODER AM 2914

8 LEVELS

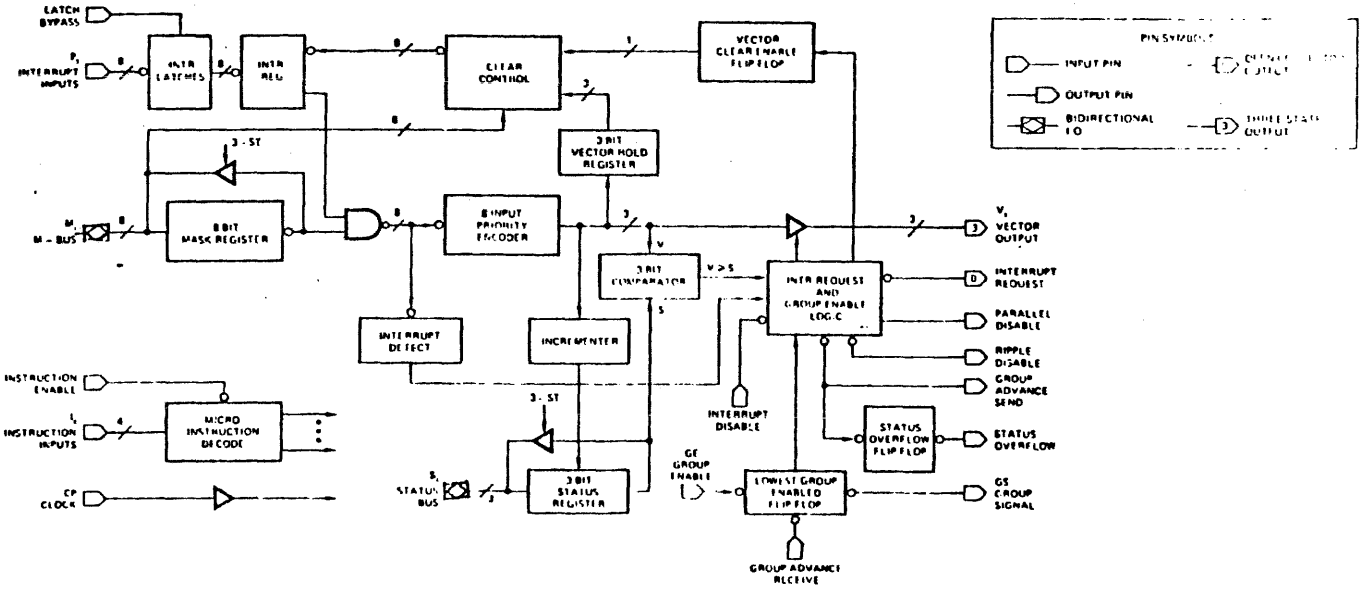
MASKING REGISTER

STATUS FOR LOWEST PERMISSABLE LEVEL

VECTOR OUTPUT

CASCADABLE FOR 8 LEVELS

70 NS DELAY



SYBEX

SUMMARY

16 BIT PROCESSOR EXPLOITS MAXIMUM CAPABILITIES OF THE AMD 2900 BIT SLICE SERIES

- MINIMUM CYCLE TIME
- POWERFUL MICROINSTRUCTION
- HIGH DEGREE OF PARALLELISM
  - LOOK AHEAD
  - FETCH PIPELINE
  - BRANCH PREPARATION (SINGLE INSTRUCTION INTERRUPT SERVICE)
  - CONDITION TESTING
  - LOOP COUNTING (SINGLE INSTRUCTION LOOPS)
- POWERFUL DATA PATHS
- SIGNIFICANT AUTOMATISM
  - OP CODE VECTORING
  - LOOP CONTROL
  - INTERRUPTS { SYSTEM / I/O

~50 CHIPS

NEW AMD CHIPS

- AM 2901A SCHEDULED (76Q4)

  IMPROVED SWITCHING SPEED

- NEW CONTROL CIRCUITS: (76Q4)

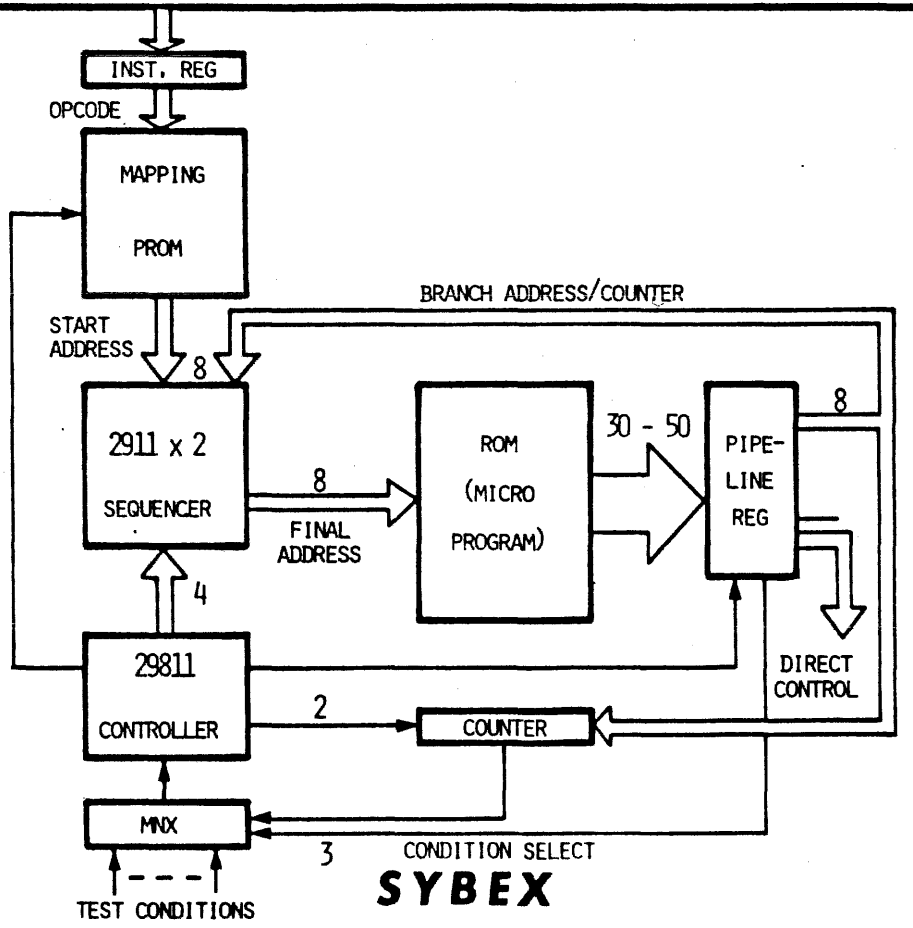
  - 29811 FOR SELECT LINES AND STACK CONTROL OF 2911
    CONTAINS LOOP COUNTER

  - 29803 CONTROLS OR INPUTS ON 2909 = MULTI-WAY BRANCHING
    (UP TO 4 TEST LINES)

- NEW 2-PORT 16 x 4 RAMS 29704, 29705 (76Q4)

- 2910 MICROPROGRAM CONTROLLER

USING A CONTROLLER



AM 29811 SEQUENCE CONTROLLER

JUMPS

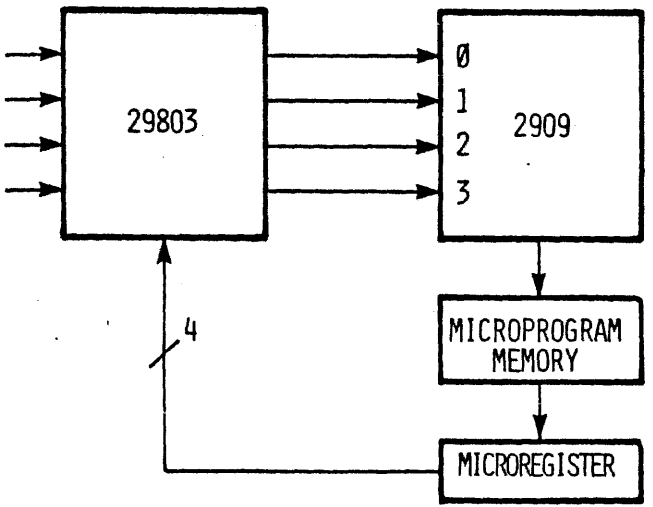
- TO ZERO
- TO ADDRESS
- REPEAT IF COUNTER ≠ 0
- TO SUBROUTINE (1 OF 2)
- TO EXTERNAL ADDRESS
- TO MAP ADDRESS

COUNTER OPERATIONS

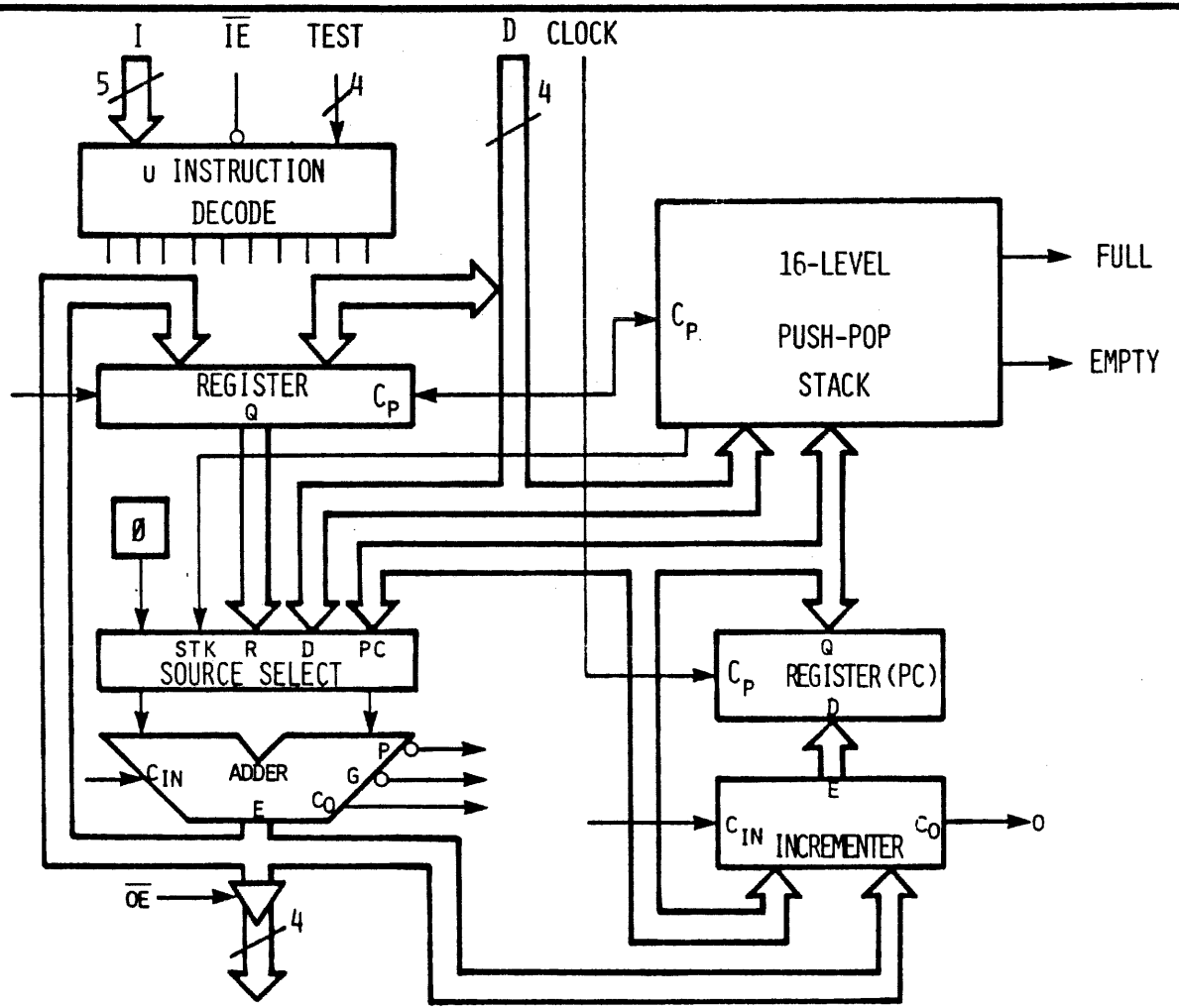
- LOAD
- TEST AND JUMP

STACK OPERATIONS

- PUSH PC
- RETURN
- JUMP AND POP


29803
2909
0
1
2
3
4
MICROPROGRAM MEMORY
MICROREGISTER

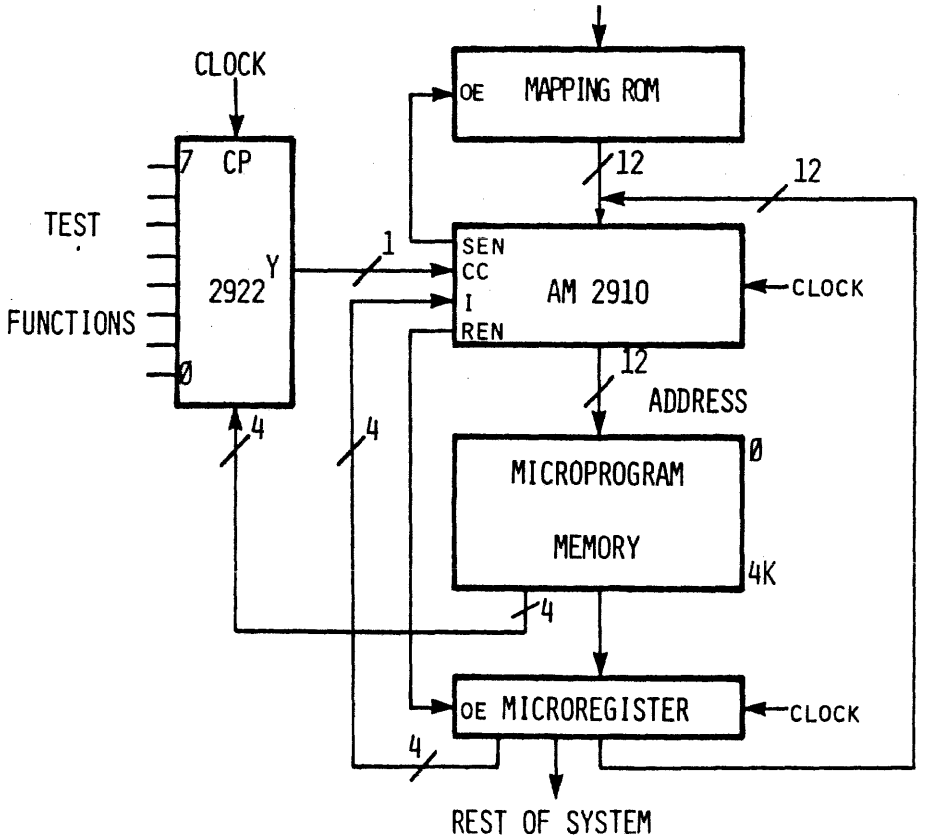
2930 PROGRAM CONTROL UNIT



2910 CHARACTERISTICS

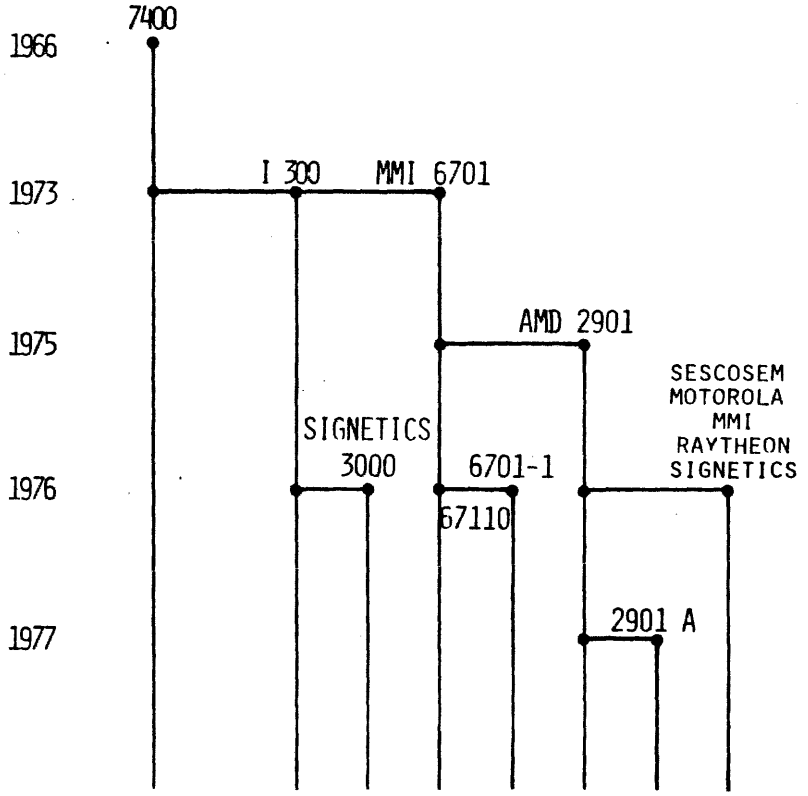
- 12-BIT MICROPROGRAM ADDRESS GENERATOR
- 4 LEVEL STACK
- 12-BIT LOOP COUNTER (TEST/DECREMENT)
- 16 PAIRS OF INSTRUCTIONS:
    - 4 BIT PAIR SELECT
    - CONDITION CODE CC + LOOP COUNTER SELECTS ONE OF PAIR
- 40-PIN DIP

AMD 2910 MICROPROGRAM CONTROL UNIT



4. OTHER BIT-SLICE DEVICES

THE BIT-SLICE FAMILY TREE

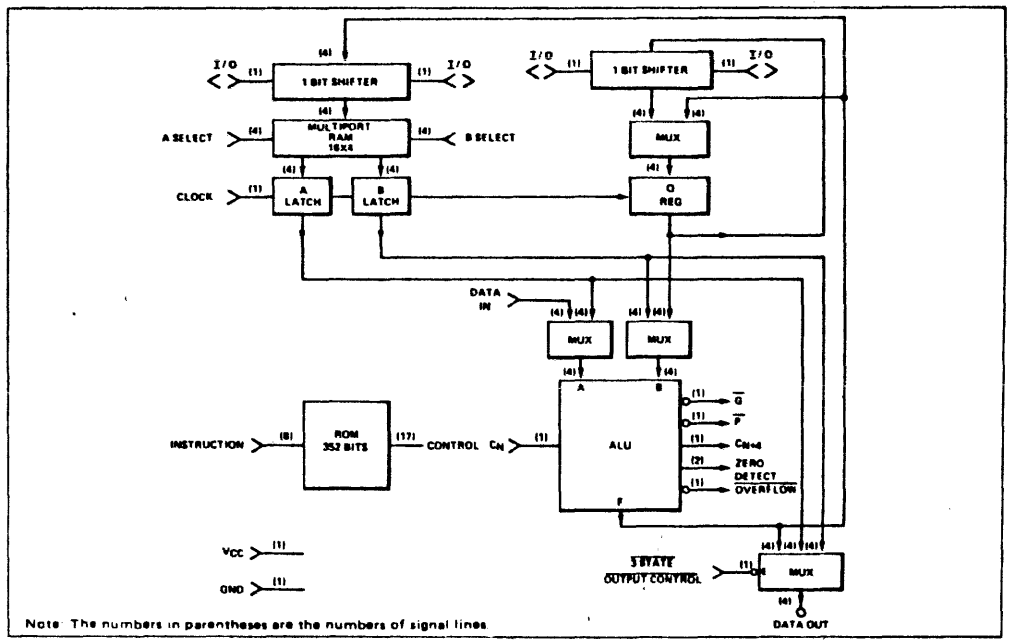


SYBEX

OTHER BIT-SLICE SERIES - MM 5700/6700 SERIES

MONOLITHIC MEMORIES 5701/6701

IMMEDIATE PREDECESSOR OF AMD 2901
ALMOST IDENTICAL ARCHITECTURE

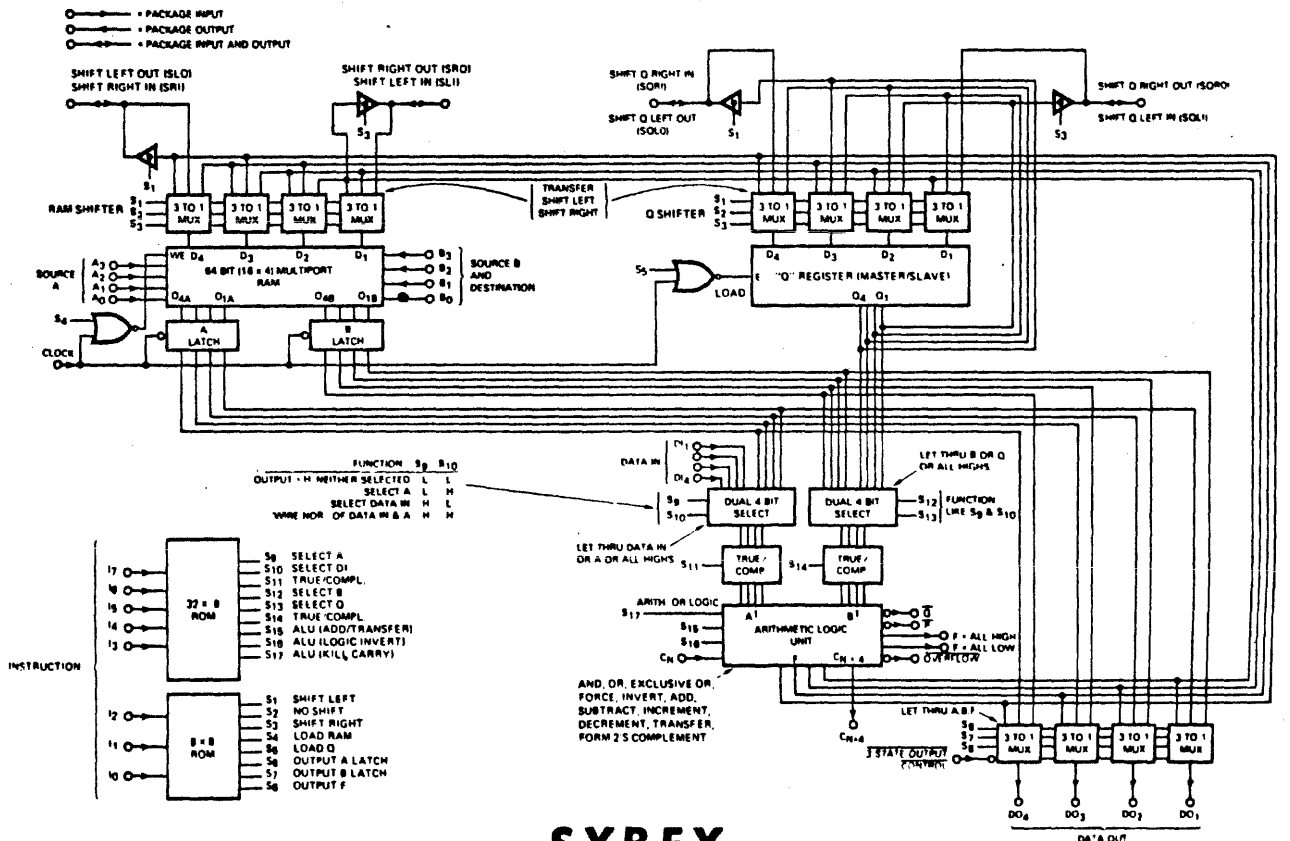


MM 6701 VERSUS AMD 2901

	6701	2901	
CYCLE	185 NS (SPEC 175 NS) (6701 - 1: 90 NS)	105 NS	A, B CLK
ADDRESS HOLD	-10 NS/CLK ↓	Ø NS/CLK ↓	
16 BIT COMPARISON (W/O SHIFT AROUND)	185 NS (175)	145 NS	
FLAGS OUT	C, OVR, F = Ø, F = 1111 (6701 - 1 HAS N)	C, OVR, N, F = Ø	
BUSSING	A OR B OR F TO DATA OUT B ⊙ D→B ALWAYS LOAD B AND/OR Q	A OR F TO DATA OUT A ⊙ D→B LOAD B AND/OR Q OR NO LOAD	

(MISCELLANEOUS ALU DIFFERENCES)

MM 6701 DETAILED DIAGRAM



SYBEX

MM 6701 TIMING

A.C CHARACTERISTICS*

PARAMETER	SYMBOL/CONDITIONS	TIME IN NANOSECONDS 5 V ± 10% -55° TO 125°C 6701 MIN	TYP†	MAX	5 V ± 5% 0° TO 75°C 6701 MIN	TYP†	MAX
Delay from PAN Address (A_{PA}) or B_{PA} to	Figure 5A, Clock High						
C_i			85	130		85	100
P			85	130		85	100
C_{MP4}			95	140		95	110
Zero Detect			120	170		120	140
Overflow			100	155		100	120
RAM Shifter Outputs			110	165		110	130
Data Outputs from the ALU	T_{CI}	20	85	130	20	85	100
Data Outputs from the ALU	T_{AA}	40	110	170	40	110	140
Delay from Instruction Input (I_{PA}) to	Figure 5A						
C_i			95	95		85	75
P			95	95		95	75
C_{MP4}			80	100		80	80
Zero Detect			80	120		80	105
Overflow			75	100		75	95
RAM Shifter Outputs			75	120		75	95
Q Shifter Outputs			25	80		25	40
Data Outputs	T_{ID}	30	85	120	10	85	100
Delay from Data Inputs (ID_{PA}) to	Figure 5A						
C_i			25	80		25	40
P			25	80		25	40
C_{MP4}			25	85		25	40
Zero Detect			20	95		20	75
Overflow			40	75		40	85
RAM Shifter Outputs			80	80		80	70
Data Outputs	T_{DO}	20	30	85	10	30	15
Delay from Carry In (IC_{PA}) to	Figure 5A						
C_{MP4}			15	30		15	25
Zero Detect			30	45		30	40
Overflow			25	40		25	35
RAM Shifter Outputs			25	40		25	35
Data Outputs		5	30	45	5	30	40
Delay from Clock Rise or Fall to	A_{PA}, B_{PA}, I_{PA}, D_{IN} constant						
C_i				145		85	115
P				145		85	115
C_{MP4}				115		100	125
Zero Detect				170		120	140
Overflow				165		115	135
RAM Shifter Outputs				165		110	130
Q Shifter Outputs				80		30	40
Data Outputs from the ALU	T_{CI}	20	80	145	20	85	115
Data Outputs from the ALU				170	30	115	140
Delay Time 3 State Control to	All Inputs constant						
Low Data Output	T_{EA}	10	20	30	10	20	25
High Impedance Output	T_{ER}	5	14	25	5	15	20

* The maximum of all possible input and output combinations is specified with outputs sinking maximum current through a resistor to V_{CC} and maximum total load (CL pF on each output).

† Typical values are measured at 5.0 V and 25°C

A.C CHARACTERISTICS* – Cont'd

PARAMETER	SYMBOL/CONDITIONS	TIME IN NANOSECONDS 5 V ± 10% -55° TO 125°C 6701 MIN	TYP†	MAX	5 V ± 5% 0° TO 75°C 6701 MIN	TYP†	MAX
Setup Time Before the High to Low Transition of Clock	Figure 5B						
A_{PA}	T_{AS}	115	80		85	60	
B_{PA}	T_{BS1}	120	80		90	60	
I_0, I_1, I_2	T_{IS1}	30	5		15	5	
Setup Time Before the Low to High Transition of Clock	Figure 5B						
A_{PA}, B_{PA}	T_{BS2}	240	130		185	130	
C_{IN}	T_{CINS}	85	45		70	45	
I_3, I_4, I_5, I_6, I_7	T_{IS2}	190	90		120	90	
Data Inputs (ID_{IN})	T_{DH}	95	80		70	50	
RAM Shifter Inputs	T_{RS}	60	30		45	30	
Q Shifter Inputs	T_{QS}	45	15		30	15	
Hold Time After the Low to High Transition of Clock	Figure 5B						
A_{PA}	T_{AH}	$-T_{CW}-10$	$-T_{CW}-25$		$-T_{CW}-15$	$-T_{CW}-10$	
B_{PA}	T_{BH}	-10	-30		-10	20	
I_0, I_1, I_2	T_{IH1}	-20	-40		-20	-40	
I_3, I_4, I_5, I_6, I_7	T_{IH2}	-30	-60		-30	-60	
C_{IN}	T_{CINH}	-20	-40		-20	-40	
Data Inputs (D_{IN})	T_{DH}	-75	-50		-25	-50	
RAM Shifter Inputs	T_{RH}	-25	-50		-25	-50	
Q Shifter Inputs	T_{QH}	-15	-30		-15	-30	
System Parameters	Figure 5B						
Clock Pulse Width (Low Time)	T_{CW}	80	75		50	25	
Minimum Microinstruction Cycle Time	$T_{CYCLE} = T_{BS2} + T_{BH}$	230	110	∞	175	110	∞

* The maximum of all possible input and output combinations is specified with outputs sinking maximum current through a resistor to V_{CC} and driving a 30 pF load (15 pF on Shift I/O to ground).

† Typical values are measured at V_{CC} = 5.0 V and 25°C

SYBEX

MM 6701 MICRO INSTRUCTIONS

INSTRUCTIONS IN THE 32 x 8 ROM - POSITIVE LOGIC (1 = H, 0 = L) INTERPRETATION

ROM (ROM 3)						ALU Instruction (See Pg. 10 for Symbology)	ALU OUTPUT		TYPICAL USES
I_2	I_1	I_0	I_3	Decimal	Octal		No Carry In (C_N = L)	With Carry In (C_N = H)	
L	L	L	L	0	00				
L	L	L	H	1	01				AND
L	L	H	L	2	02				
L	L	H	H	3	03				OR
L	H	L	L	4	04				
L	H	L	H	5	05				Exclusive OR
L	H	H	L	6	06				
L	H	H	H	7	07				Invert
H	L	L	L	8	10				
H	L	L	H	9	11				
H	L	H	L	10	12				
H	L	H	H	11	13				
H	H	L	L	12	14				
H	H	L	H	13	15				
H	H	H	L	14	16				
H	H	H	H	15	17				
H	L	L	L	16	20				
H	L	L	H	17	21				
H	L	H	L	18	22				
H	L	H	H	19	23				
H	H	L	L	20	24				
H	H	L	H	21	25				
H	H	H	L	22	26				
H	H	H	H	23	27				
H	L	L	L	24	30				
H	L	L	H	25	31				
H	L	H	L	26	32				
H	L	H	H	27	33				
H	H	L	L	28	34				
H	H	L	H	29	35				
H	H	H	L	30	36				
H	H	H	H	31	37				

INSTRUCTION MODIFIERS IN THE 8 x 8 ROM -
POSITIVE LOGIC (1 = H, 0 = L) INTERPRETATION

Rom Word			Rom Word	Load Control		Shift Control		Destination Control			
I_2	I_1	I_0	Decimal	Load Ram B_3	Load Q	Shift Left	Shift Right	Don't Shift	A Latch	B Latch	ALU Output F
L	L	L	0	X				X			X
L	L	H	1	X				X	X		
L	H	L	2	X				X		X	
L	H	H	3	X		X					X
H	L	L	4	X			X				X
H	L	H	5	X	X	X					X
H	H	L	6	X			X				X
H	H	H	7	X	X		X				X

PIN CONFIGURATION

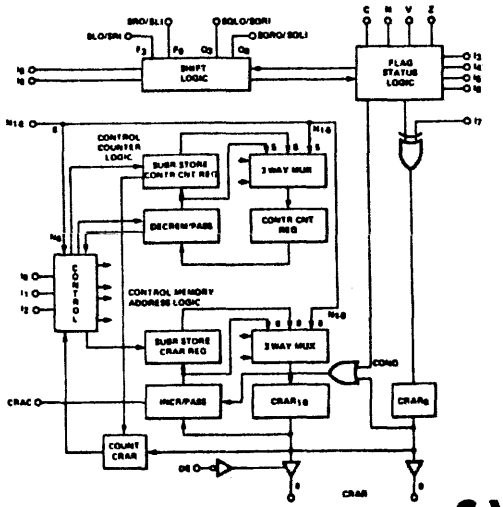
SYBEX

MM 5710/6710 MICROPROGRAM CONTROL UNIT

COMPLETE CONTROL IN ONE CHIP

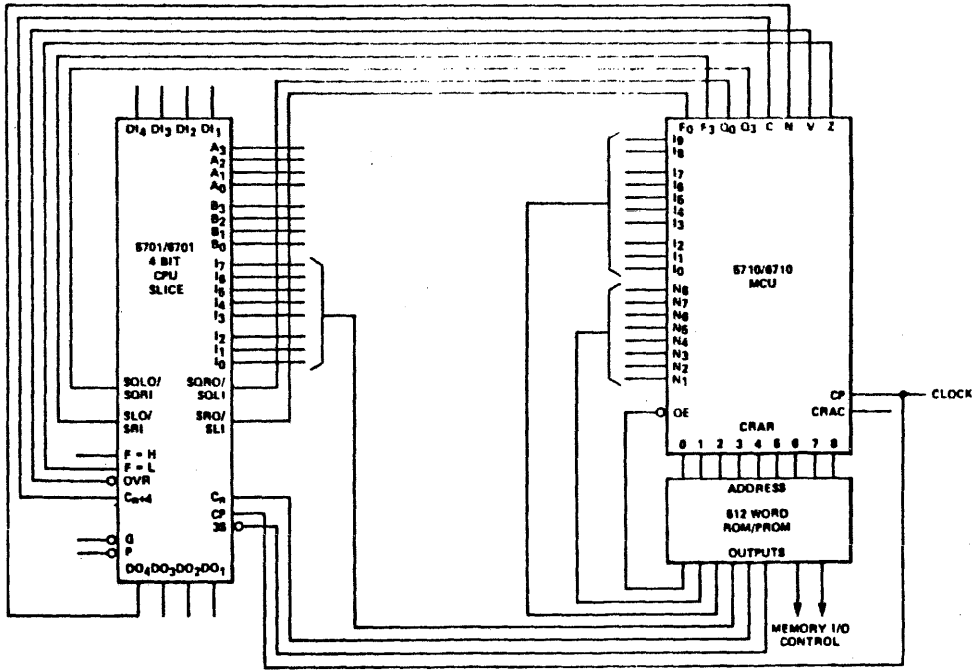
CONDITIONAL BRANCHING
SHIFT CONTROL
1-LEVEL SUBROUTINE REGISTER
LOOP COUNTER
ADDRESS 512 WORDS MICRO CONTROL

40 PIN PACK
100 NS CLOCK MIN
(30-LOW, 70-HIGH)



SYBEX

6701/6710 INTERCONNECTIONS



SYBEX

6710 MICRO CONTROL SET

Table I – MCU Control Options

Control Code			Address Field Destination	Control Action
I_2	I_1	I_0		
0	0	0	None	Continue to next μinstruction
0	0	1	Control Counter	Continue to next μinstruction
0	1	0	None/CRAR (Cond. Jump)	Jump to next μinstruction if Control Counter ≠ 0, Decrement Control Counter
0	1	1	None/CRAR (Cond. Subr. Jump)	Subroutine Jump to next μinstruction if Control Counter ≠ 0, Decrement Control Counter
1	0	0 *	None	Return from Subroutine
1	0	1 **	CRAR (Jump Subroutine)	Jump to next μinstruction Return from Subroutine when Control Counter Subroutine Latch ≡ $CRAR_{0-4}$
1	1	0	CRAR (Jump)	Jump to next μinstruction
1	1	1	CRAR (Jump Subroutine)	Subroutine Jump to next μinstruction

Table III – Shift Control

Control Code		Shifting Operation	Bidirectional Shift Lines Acting as Outputs			
			F_3 (SLO/SRI)	F_0 (SRO/SLI)	Q_3 (SQLO/SQRI)	Q_0 (SQRO/SQLI)
I_9	I_8					
0	0	Arithmetic Shift Left	–	Q_3	–	Flag (SC)
0	1	Arithmetic Shift Right	N⊕V	–	F_0	–
1	0	Rotate Shift Left		F_3	–	Flag (SC)
1	1	Rotate Shift Right	F_0	–	–	–

High Impedance State
SC Contents of Carry Flip Flop

6710 MICRO CONTROL CONTINUED

Table II – Flag Status Control Options

Control Code				Action		
I_6	I_5	I_4	I_3			
0	0	0	0	None	I_7 to $CRAR_0$	Unconditional Branch
0	0	0	1	Store C	I_7 to $CRAR_0$	Unconditional Branch
0	0	1	0	Store N, V, Z	I_7 to $CRAR_0$	Unconditional Branch
0	0	1	1	Store C, N, V, Z	I_7 to $CRAR_0$	Unconditional Branch
0	1	0	0	Shift Flag Register into Q_0	I_7 to $CRAR_0$	Unconditional Branch
0	1	0	1	Shift Flag Register out of Q_0	I_7 to $CRAR_0$	Unconditional Branch
0	1	1	0	Instantaneous value of Q_3 to $CRAR_0$		Conditional Branch
0	1	1	1	Instantaneous value of Q_0 to $CRAR_0$		Conditional Branch
1	0	0	0	Stored value of C to $CRAR_0$		Conditional Branch
1	0	0	1	Stored value of N to $CRAR_0$		Conditional Branch
1	0	1	0	Stored value of V to $CRAR_0$		Conditional Branch
1	0	1	1	Stored value of Z to $CRAR_0$		Conditional Branch
1	1	0	0	Instantaneous value of C to $CRAR_0$		Conditional Branch
1	1	0	1	Instantaneous value of N to $CRAR_0$		Conditional Branch
1	1	1	0	Instantaneous value of V to $CRAR_0$		Conditional Branch
1	1	1	1	Instantaneous value of Z to $CRAR_0$		Conditional Branch

Code bit I_7 inverts the status of output line so that the condition is dependent upon $\overline{C}$, etc. For the first six entries in the table if $I_7 = 0$ there is an unconditional branch to X,0. If $I_7 = 1$ an unconditional branch to X,1.

COMPARISON 6710 VERSUS 2909

	6710	2909
FUNCTIONS	ADDRESS GATING LOOP COUNTING FLAG TESTING SHIFT CONTROL	ADDRESS GATING STACK CONTROL
ARCHITECTURE	COMPLETE CONTROL UNIT FOR SMALL SYSTEM	SLICE OF CONTROL UNIT FOR LARGE SYSTEM
SPEED: CYCLE	100 NS MIN	60 NS MIN
SPEED: CLOCK TO ADDROUT	25 NS	40 NS (FILE POP MODE)
SIZE	40 PIN	28/20 PIN

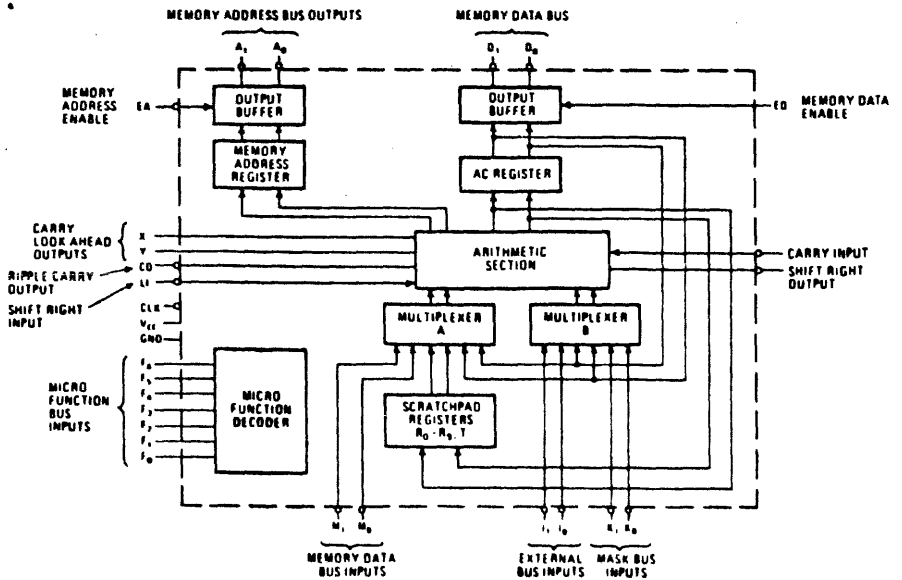
OTHER MM 5700/6700 SERIES DEVICES

5702/6702	4 BIT CPU SLICE WITH ADDITIONAL BIDIRECTIONAL I/O PORT (NO Q REGISTER)
5716/6716	PRIORITY INTERRUPT CONTROLLER
5717/6717	DMA CONTROLLER
5718/6718	I/O INTERFACE CONTROLLER

INTEL 3000 SERIES

3002 CENTRAL PROCESSING ELEMENT

- 2 BIT WIDE SLICE
- 11 GENERAL REGISTERS
- ACCUMULATOR
- MEMORY ADDRESS REGISTER
- 3 INPUT CHANNELS
- 2 OUTPUT CHANNELS
- 100 NS CYCLE
- 28 PIN PACK



INTEL 3002 CPE

16 BIT CYCLE TIME WITH LOOK AHEAD (NO SHIFT AROUND): 155 NS
(INTEL 3003)

ADVANTAGES:

- SPEED
- MULTIPLE BUS INPUTS AND OUTPUTS
- BUFFERED OUTPUTS

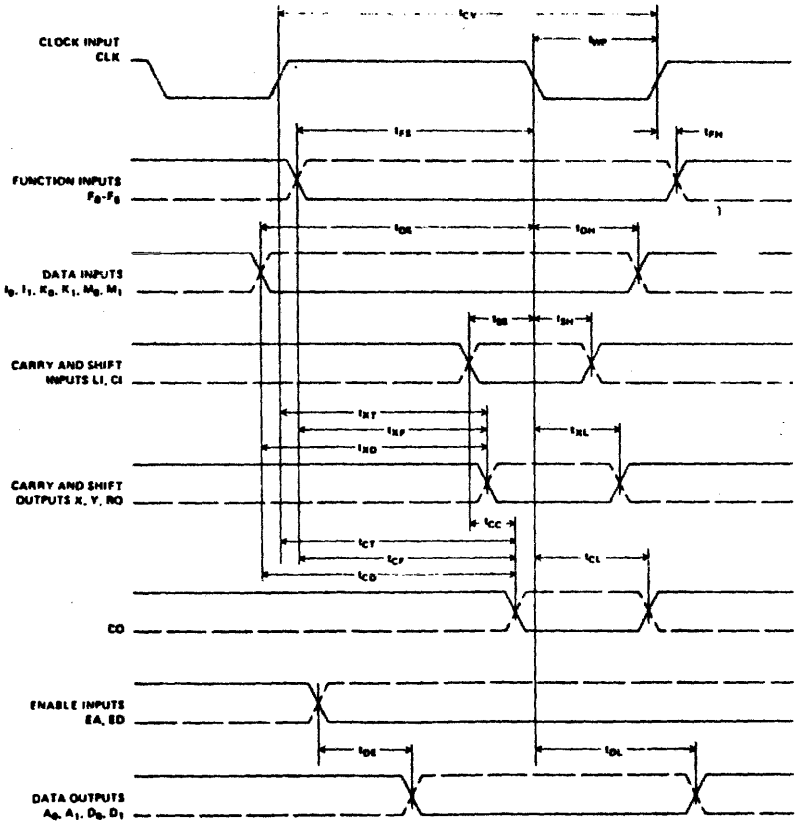
DISADVANTAGES:

- 2 BITS PER CHIP
- LIMITED GENERAL REGISTER ACCESS - ALU A - MUX ONLY
- COMPLICATED MICROINSTRUCTION SET
- LIMITED FLAG OUT (CARRY AND LOOK AHEAD)

INTEL 3002 CPE TIMING

T_A = 0°C to 70°C, Vcc = 5V ±5%

SYMBOL	PARAMETER	MIN	TYP[1]	MAX	UNIT
t_{CY}	Clock Cycle Time[2]	100	70		ns
t_{WP}	Clock Pulse Width	33	20		ns
t_{FS}	Function Input Set-Up Time (F_0 through F_8)	60	40		ns
t_{DS}	Data Set-Up Time I_0, I_1, M_0, M_1, K_0, K_1	50	30		ns
t_{SS}	LI, CI	27	13		ns
	Data and Function Hold Time:				
t_{FH}	F_0 through F_8	5	-2		ns
t_{DH}	I_0, I_1, M_0, M_1, K_0, K_1	5	-4		ns
t_{SH}	LI, CI	15	2		ns
	Propagation Delay to X, Y, RO from:				
t_{XF}	Any Function Input		37	52	ns
t_{XD}	Any Data Input		29	42	ns
t_{XT}	Trailing Edge of CLK		40	80	ns
t_{XL}	Leading Edge of CLK	20			ns
	Propagation Delay to CO from:				
t_{CL}	Leading Edge of CLK	20			ns
t_{CT}	Trailing Edge of CLK		48	70	ns
t_{CF}	Any Function Input		43	65	ns
t_{CD}	Any+Data Input		30	55	ns
t_{CC}	CI (Ripple Carry)		14	25	ns
	Propagation Delay to A_0, A_1, D_0, D_1 from:				
t_{DL}	Leading Edge of CLK	5	32	50	ns
t_{DE}	Enable Input ED, EA		12	25	ns



SYBEX

INTEL 3002 CPE MICRO FUNCTIONS

APPENDIX A MICRO-FUNCTION SUMMARY

F GROUP	R-GROUP	MICRO-FUNCTION	
0	I	$R_n + (AC \wedge K) + CI \rightarrow R_n, AC$	
	II	$M + (AC \wedge K) + CI \rightarrow AT$	
	III	$AT_L \wedge (\overline{I_L \wedge K_L}) \rightarrow RO$ $[AT_L \wedge (I_L \wedge K_L)] \vee [AT_H \vee (I_H \wedge K_H)] \rightarrow AT_L$	$LI \vee [(I_H \wedge K_H) \wedge AT_H] \rightarrow AT_H$
1	I	$K \vee R_n \rightarrow MAR$	$R_n + K + CI \rightarrow R_n$
	II	$K \vee M \rightarrow MAR$	$M + K + CI \rightarrow AT$
	III	$(\overline{AT} \vee K) + (AT \wedge K) + CI \rightarrow AT$	
2	I	$(AC \wedge K) - 1 + CI \rightarrow R_n$	(see Note 1)
	II	$(AC \wedge K) - 1 + CI \rightarrow AT$	
	III	$(I \wedge K) - 1 + CI \rightarrow AT$	
3	I	$R_n + (AC \wedge K) + CI \rightarrow R_n$	
	II	$M + (AC \wedge K) + CI \rightarrow AT$	
	III	$AT + (I \wedge K) + CI \rightarrow AT$	
4	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$	$R_n \wedge (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$	$M \wedge (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$	$AT \wedge (I \wedge K) \rightarrow AT$
5	I	$CI \vee (R_n \wedge K) \rightarrow CO$	$K \wedge R_n \rightarrow R_n$
	II	$CI \vee (M \wedge K) \rightarrow CO$	$K \wedge M \rightarrow AT$
	III	$CI \vee (AT \wedge K) \rightarrow CO$	$K \wedge AT \rightarrow AT$
6	I	$CI \vee (AC \wedge K) \rightarrow CO$	$R_n \vee (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (AC \wedge K) \rightarrow CO$	$M \vee (AC \wedge K) \rightarrow AT$
	III	$CI \vee (I \wedge K) \rightarrow CO$	$AT \vee (I \wedge K) \rightarrow AT$
7	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$	$R_n \overline{\vee} (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$	$M \overline{\vee} (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$	$AT \overline{\vee} (I \wedge K) \rightarrow AT$

NOTES

1. 2's complement arithmetic adds 111 11 to perform subtraction of 000 01
2. R_n includes T and AC as source and destination registers in R group I micro functions
3. Standard arithmetic carry output values are generated in F-group 0, 1, 2 and 3 instructions

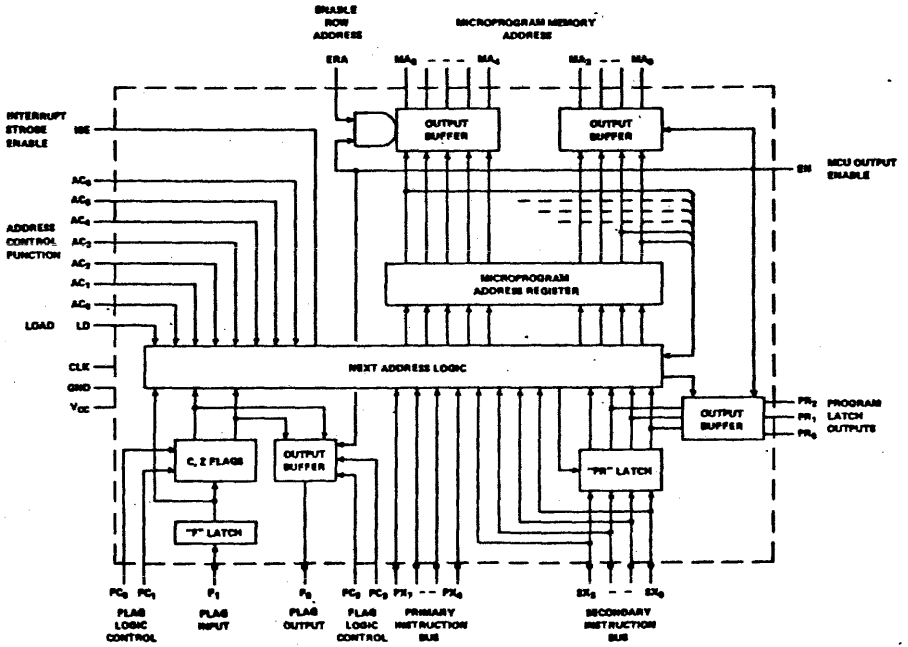
SYMBOL	MEANING
I, K, M	Data on the I, K, and M busses, respectively
CI, LI	Data on the carry input and left input, respectively
CO, RO	Data on the carry output and right output, respectively
R_n	Contents of register n including T and AC (R Group II
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
MAR	Contents of the memory address register
L, H	As subscripts, designate low and high order bit, respectively
+	2's complement addition
-	2's complement subtraction
$\wedge$	Logical AND
$\vee$	Logical OR
$\overline{\vee}$	Exclusive NOR
$\rightarrow$	Transfer into

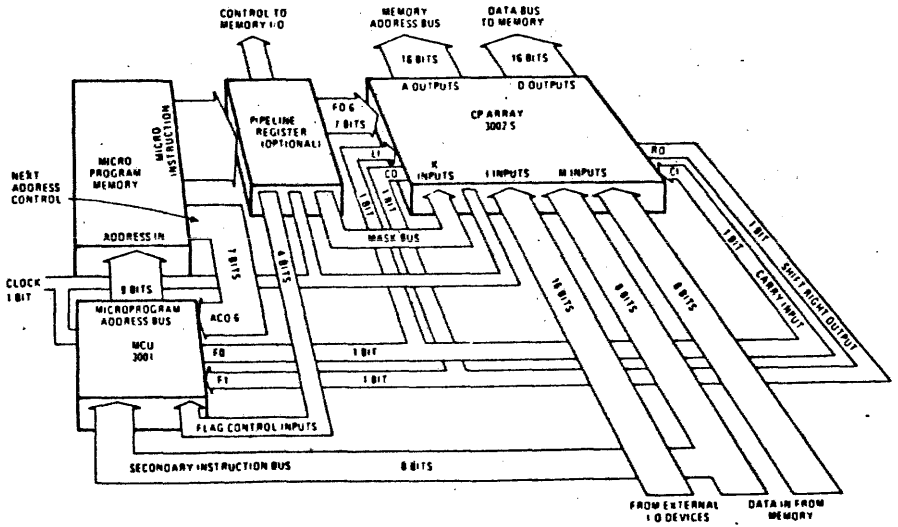
APPENDIX B ALL-ZERO AND ALL-ONE K-BUS MICRO-FUNCTIONS

F-GROUP	R-GROUP	K-BUS = 00 MICRO-FUNCTION	MNEMONIC	K-BUS = 11 MICRO-FUNCTION	MNEMONIC
0	I	$R_n + CI \rightarrow R_n$, AC	ILR	AC + R_n + CI → R_n, AC	ALR
	II	M + CI → AT	ACM	M + AC + CI → AT	AMA
	III	$AT_L \rightarrow RO$ $AT_H \rightarrow AT_L$ $LI \rightarrow AT_H$	SRA	(See Appendix A)	–
1	I	$R_n \rightarrow MAR$ $R_n + CI \rightarrow R_n$	LMI	11 → MAR $R_n - 1 + CI \rightarrow R_n$	DSM
	II	M → MAR M + CI → AT	LMM	11 → MAR M − 1 + CI → AT	LDM
	III	$\overline{AT}$ + CI → AT	CIA	AT − 1 + CI → AT	DCA
2	I	CI − 1 → R_n See Note 1	CSR	AC − 1 + CI → R_n See Note 1	SDR
	II	CI − 1 → AT See Notes 1,4	CSA	AC − 1 + CI → AT See Notes 1,4	SDA
	III	(See CSA above)	–	I − 1 + CI → AT	LDI
3	I	R_n + CI → R_n	INR	AC + R_n + CI → R_n	ADR
	II	(See ACM above)	–	(See AMA above)	–
	III	AT + CI → AT	INA	I + AT + CI → AT	AIA
4	I	CI → CO 0 → R_n	CLR	CI ∨ (R_n ∧ AC) → CO R_n ∧ AC → R_n	ANR
	II	CI → CO 0 → AT	CLA	CI ∨ (M ∧ AC) → CO M ∧ AC → AT	ANM
	III	(See CLA above)	–	CI ∨ (AT ∧ I) → CO AT ∧ I → AT	ANI
5	I	(See CLR above)	–	CI ∨ R_n → CO R_n → R_n	TZR
	II	(See CLA above)	–	CI ∨ M → CO M → AT	LTM
	III	(See CLA above)	–	CI ∨ AT → CO AT → AT	TZA
6	I	CI → CO $R_n \rightarrow R_n$	NOP	CI ∨ AC → CO R_n ∨ AC → R_n	ORR
	II	CI → CO M → AT	LMF	CI ∨ AC → CO M ∨ AC → AT	ORM
	III	(See NOP above)	–	CI ∨ I → CO I ∨ AT → AT	ORI
7	I	CI → CO $\overline{R_n} \rightarrow R_n$	CMR	CI ∨ (R_n ∓ AC) → CO R_n ∓ AC → R_n	XNR
	II	CI → CO $\overline{M}$ → AT	LCM	CI ∨ (M ∓ AC) → CO M ∓ AC → AT	XNM
	III	CI → CO $\overline{AT}$ → AT	CMA	CI ∨ (AT ∓ I) → CO I ∓ AT → AT	XNI

4 The more general operations, CSR and SDR, should be used in place of the CSA and SDA operations, respectively.

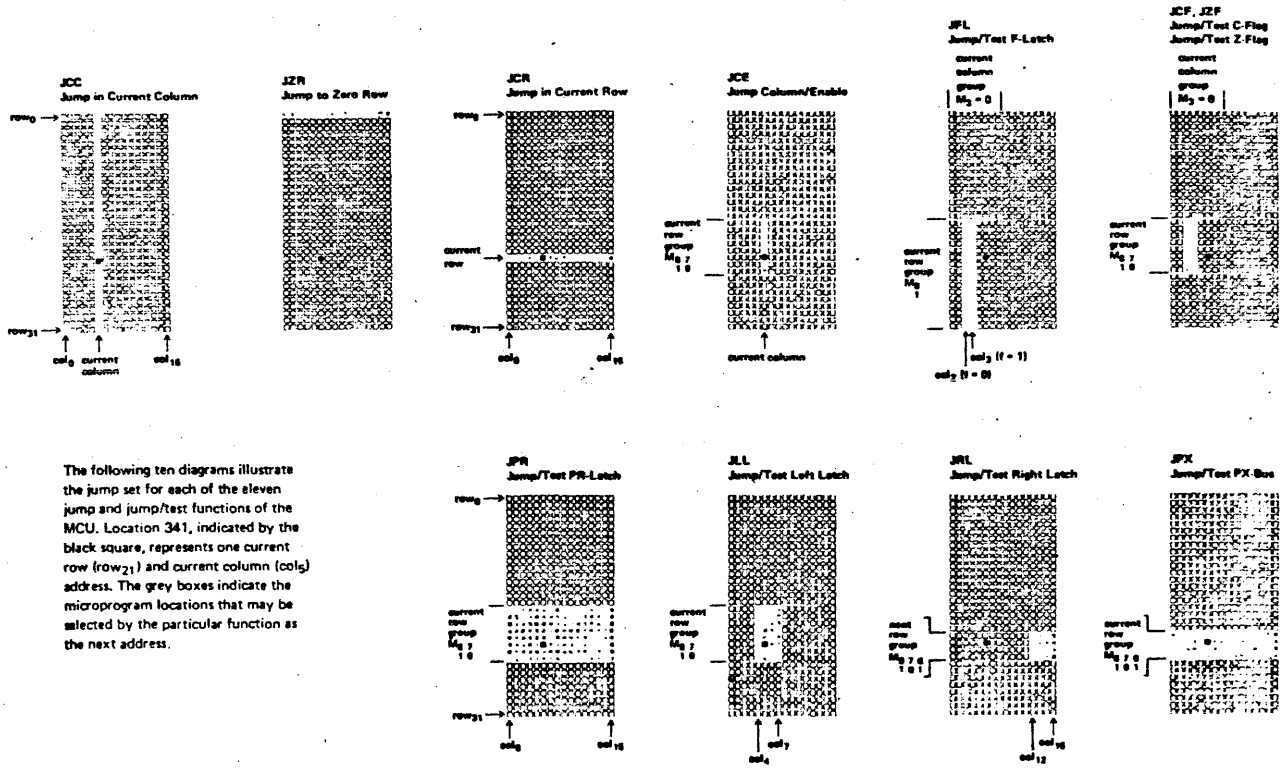
512 INSTRUCTION ADDRESSING
CONDITIONAL TESTS
BRANCH CONTROL




CONTROL TO MEMORY I/O
MEMORY ADDRESS BUS
DATA BUS TO MEMORY
16 BITS
16 BITS
A OUTPUTS
D OUTPUTS
PIPELINE REGISTER (OPTIONAL)
FOG
7 BITS
CP ARRAY 3002 S
MICRO PROGRAM MEMORY
MICRO INSTRUCTION
NEXT ADDRESS CONTROL
ADDRESS IN
CLOCK 1 BIT
9 BITS
MICROPROGRAM ADDRESS BUS
MCU 3001
ACOG
F0
F1
1 BIT
1 BIT
4 BITS
1 BITS
MASK BUS
R INPUTS
S INPUTS
M INPUTS
16 BITS
8 BITS
8 BITS
1 BIT
1 BIT
SHIFT RIGHT OUTPUT
CARRY INPUT
FLAG CONTROL INPUTS
SECONDARY INSTRUCTION BUS
8 BITS
FROM EXTERNAL I/O DEVICES
DATA IN FROM MEMORY

MNEMONIC	DESCRIPTION	FUNCTION							NEXT ROW					NEXT COL			
		AC_6	5	4	3	2	1	0	MA_8	7	6	5	4	MA_3	2	1	0
JCC	Jump in current column	0	0	d_4	d_3	d_2	d_1	d_0	d_4	d_3	d_2	d_1	d_0	m_3	m_2	m_1	m_0
JZR	Jump to zero row	0	1	0	d_3	d_2	d_1	d_0	0	0	0	0	0	d_3	d_2	d_1	d_0
JCR	Jump in current row	0	1	1	d_3	d_2	d_1	d_0	m_8	m_7	m_6	m_5	m_4	d_3	d_2	d_1	d_0
JCE	Jump in column/enable	1	1	1	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	m_2	m_1	m_0
JFL	Jump/test F-latch	1	0	0	d_3	d_2	d_1	d_0	m_8	d_3	d_2	d_1	d_0	m_3	0	1	f
JCF	Jump/test C-flag	1	0	1	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	0	1	c
JZF	Jump/test Z flag	1	0	1	1	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	0	1	z
JPR	Jump/test PR-latches	1	1	0	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	p_3	p_2	p_1	p_0
JLL	Jump/test left PR bits	1	1	0	1	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	0	1	p_3	p_2
JRL	Jump/test right PR bits	1	1	1	1	1	d_1	d_0	m_8	m_7	1	d_1	d_0	1	1	p_1	p_0
JPX	Jump/test PX-bus	1	1	1	1	0	d_1	d_0	m_8	m_7	m_6	d_1	d_0	x_7	x_6	x_5	x_4

SYMBOL	MEANING
d_n	Data on address control line n
m_n	Data in microprogram address register bit n
p_n	Data in PR-latch bit n
x_n	Data on PX-bus line n (active LOW)
f, c, z	Contents of F-latch, C-flag, or Z-flag, respectively



The following ten diagrams illustrate the jump set for each of the eleven jump and jump/test functions of the MCU. Location 341, indicated by the black square, represents one current row (row_{21}) and current column (col_5) address. The grey boxes indicate the microprogram locations that may be selected by the particular function as the next address.

3212 MULTIMODE 8 BIT LATCH (2 REQUIRED FOR BI-DIRECTIONAL)

3214 INTERRUPT CONTROL UNIT

3216 4 BIT BI-DIRECTION BUS DRIVERS/RECEIVERS

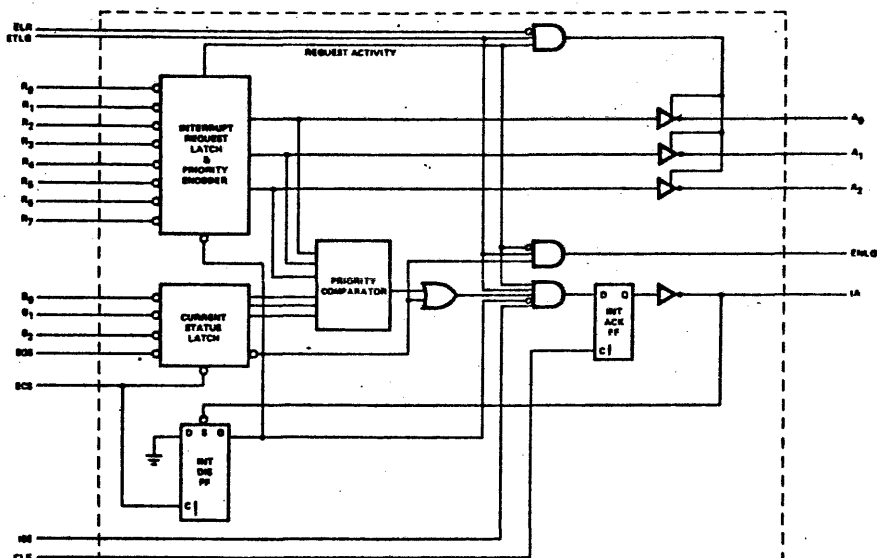


Figure 1. 3214 Block Diagram.

4 BIT CPU SLICE IN I^2L

10 REGISTERS

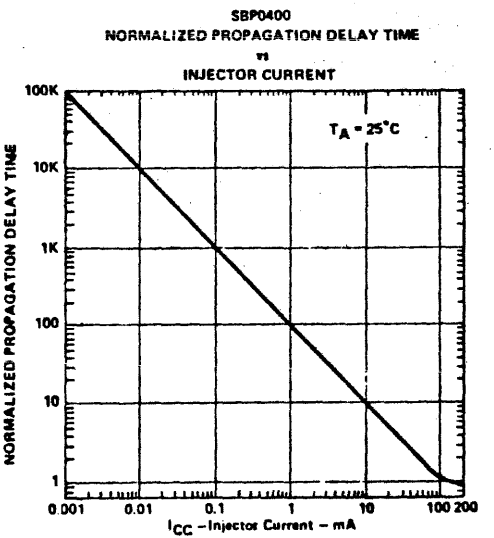
FACTORY PROGRAMMABLE PLA ON CHIP

1000 NS CLOCK: TYPICAL ALU PROPAGATE=500 NS

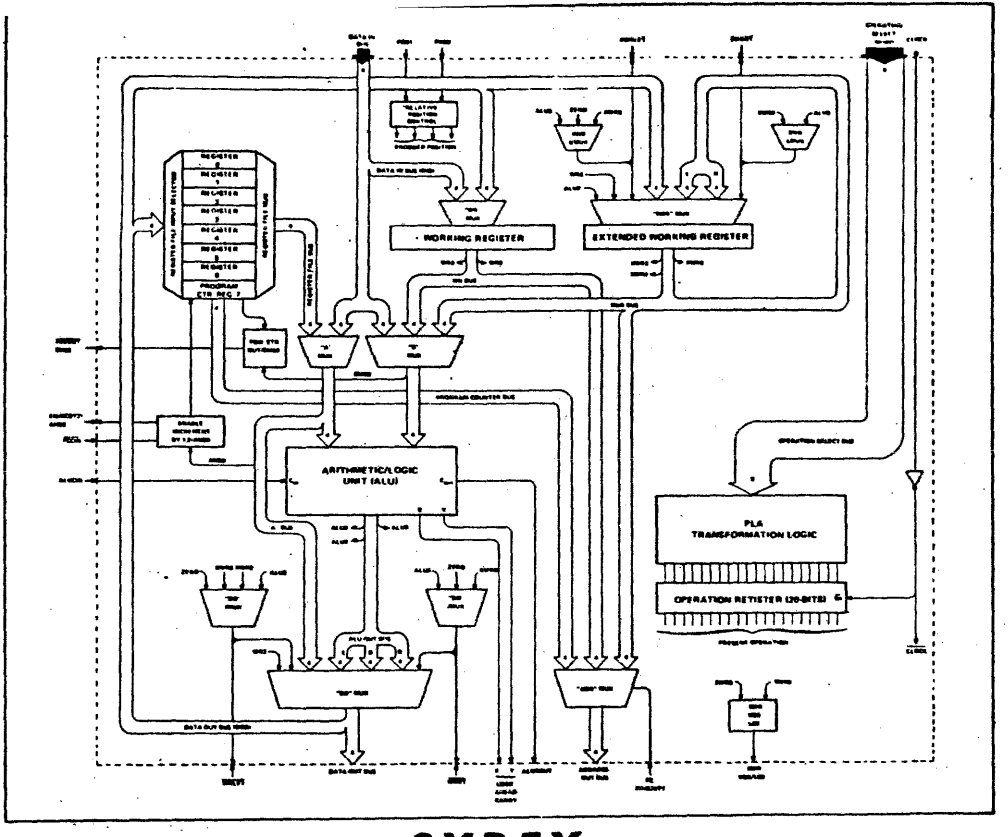
FUNCTIONS ON .85 V @ 150 MA 128 MW MAX

FULL MILITARY TEMPERATURE RANGE AVAILABLE

CAN OPERATE OVER SEVERAL ORDERS OF MAGNITUDE OF CURRENT/PROPAGATE



SYBEX


WORKING REGISTER
EXTENDED WORKING REGISTER
ARITHMETIC/LOGIC UNIT (ALU)
PLA TRANSFORMATION LOGIC

SYBEX

REGISTER-ALU SLICE/MOS-P CHANNEL

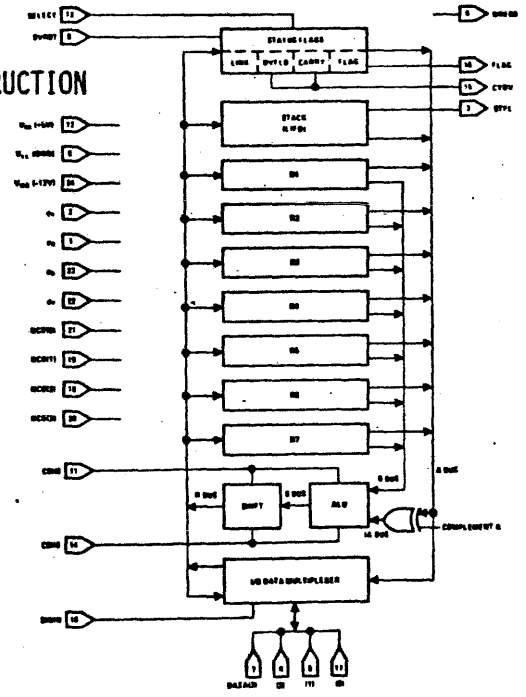
MULTIPLEXED COMMAND BUS - FOUR PHASE EXECUTION

7 GENERAL REGISTER

16 WORD STACK

BIPOLAR COMPATIBLE OUTPUTS

330 NS MIN PER PHASE / 1.3 uS PER INSTRUCTION



SYBEX

1.A Command Inputs

COMMAND BITS:[1]

TIME INTERVAL	NCB(3)	NCB(2)	NCB(1)	NCB(0)
T1	STACK	← A BUS →		
T3	COMP	← B BUS →		
T5	← CTL →		← ALU →	
T7	I/O	← R BUS →		

1.B Command Codes[3]

ALU FUNCTIONS

NCB(1), (0) @ T5	FUNCTION
11	AND
10	XOR
01	OR
00	ADD

CTL FUNCTIONS

NCB(3), (2) @ T5	FUNCTION
11	NONE
10	R BUS CONTROL
01	SHIFT LEFT
00	SHIFT RIGHT

A, B & R BUS ADDRESSES

NCB(2), (1), (0)	ADDRESS
111	ZEROES, FLAGS, STACK[2]
110	R1
101	R2
100	R3
011	R4
010	R5
001	R6
000	R7

R BUS CONTROL

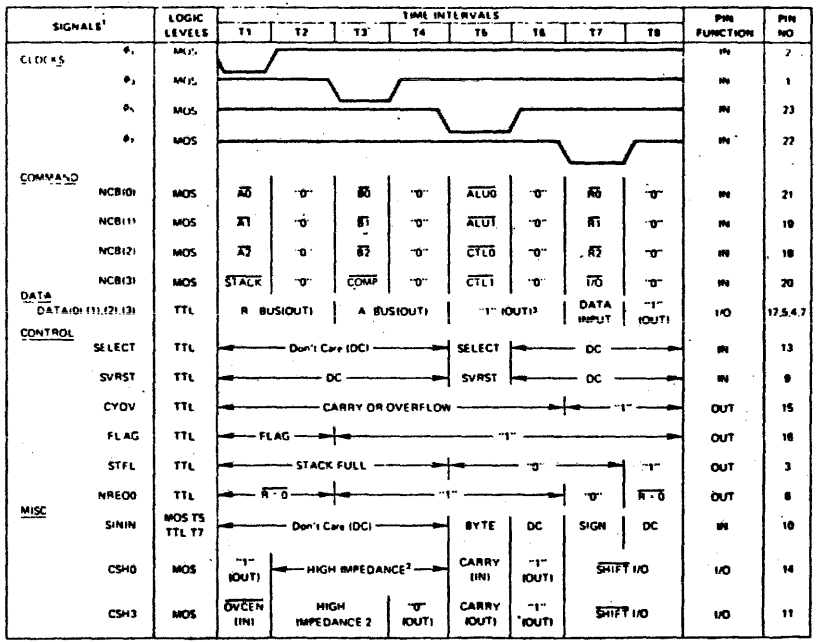
I/O (NCB(3) @ T7)	BYTE (SININ @ T5)	R BUS VALUE
1	0	OUTPUT OF SHIFTER
1	1	OUTPUT OF SHIFTER
0	0	OUTPUT OF I/O MUX
0	1	VALUE OF SIGN INPUT ON SININ @ T7

Note 1: Commands are complemented signals

Note 2: See text and Tables II and III for addressing flags and stack. B bus only addresses zeroes

Note 3. Logic values shown are values which must be applied to NCB inputs to get indicated results

NS IMP RALU TIMING DIAGRAM



Note 1: A positive true logic convention is used for all signals (i.e. "1" = more positive voltage, "0" = more negative voltage). Signal names beginning with R are complemented signals.

Note 2: CSH0 and CSH3 high impedance state for intervals T_2 through T_4 is the TRI STATE mode for output drivers.

Note 3: "1" (OUT) means RALU is driving this node to the "1" logic level during the defined interval. For bidirectional I/O lines the logic state is defined as "in" or "out".

CONTROL UNIT AND ROM

PROVIDES MULTIPLEXED INSTRUCTIONS FOR RALU

BRANCH CONTROL

SHIFT AND CARRY CONTROL

SUBROUTINING

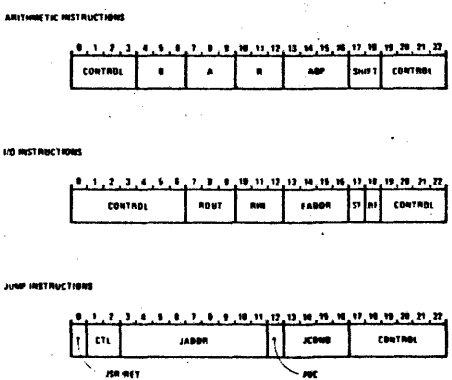
MICROPROGRAMMMABLE : 100 INSTRUCTIONS 23 BITS

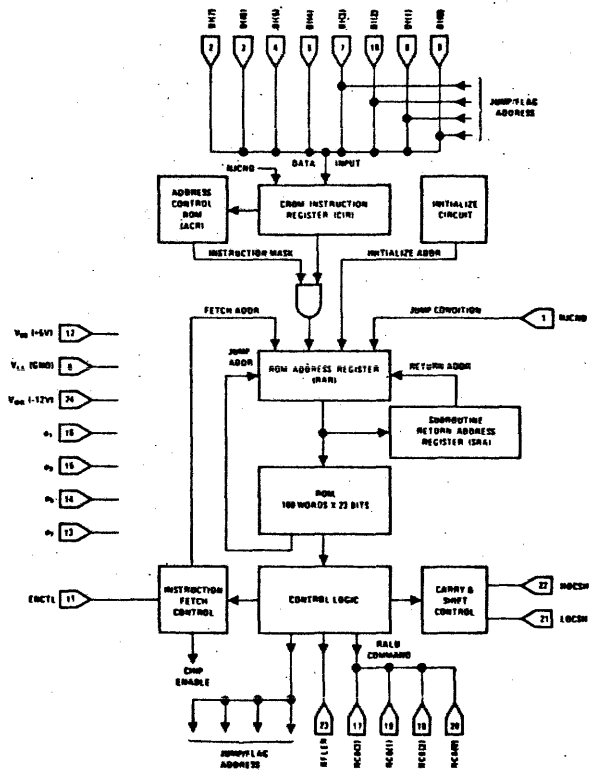
STANDARD 8 AND 16 BIT EMULATIONS AVAILABLE

DRIVES UP TO 8 RALUS

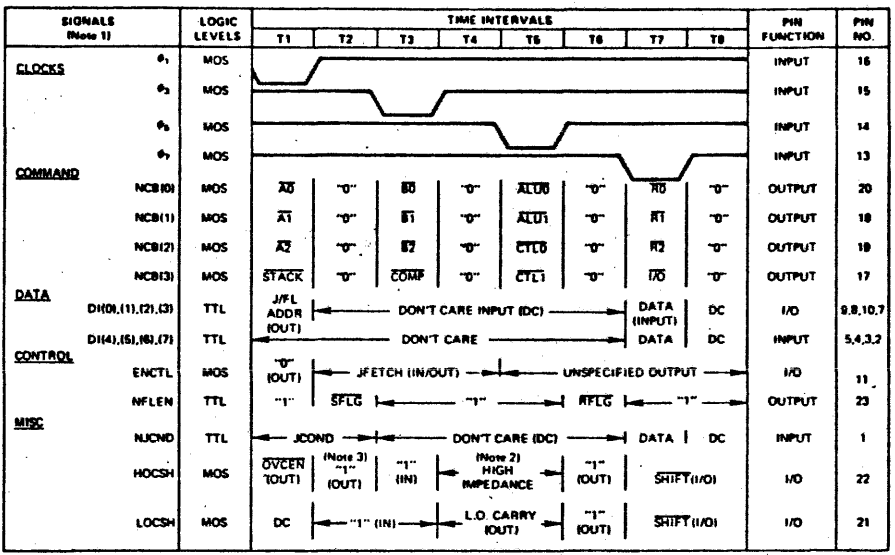
CONTROLLERS ARE ARRAYABLE

TABLE I. IMP Microinstruction Word Formats




DATA
INPUT
JUMP/FLAG ADDRESS
ADDRESS CONTROL ROM (ACR)
CROM INSTRUCTION REGISTER (CIR)
INITIALIZE CIRCUIT
INSTRUCTION MASK
INITIALIZE ADDR
FETCH ADDR
JUMP CONDITION
JUMP ADDR
ROM ADDRESS REGISTER (RAR)
RETURN ADDR
SUBROUTINE RETURN ADDRESS REGISTER (SRA)
ROM 100 WORDS X 23 BITS
INSTRUCTION FETCH CONTROL
CONTROL LOGIC
CARRY & SHIFT CONTROL
CMP ENABLE
RALU COMMAND
JUMP/FLAG ADDRESS

SYBEX



Note 1: A positive true logic convention is used for all signals except clocks. Signal names beginning with N are complemented signals.

Note 2: HOCSH at T4 and T5 is in the TRI-STATE high impedance output mode of CROM load drivers.

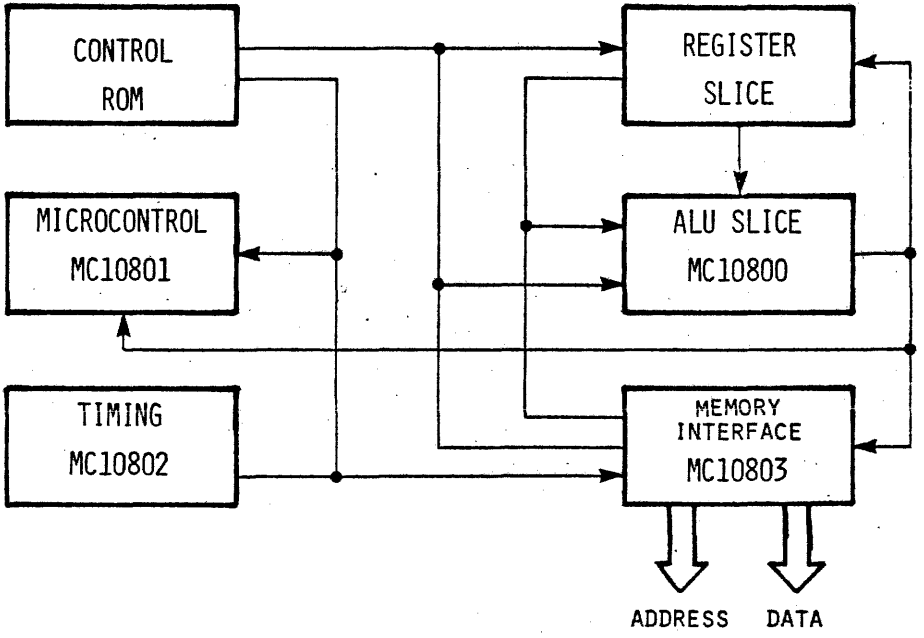
Note 3: "1" (OUT) means CROM is driving this node to the logic "1" level during the defined interval. For I/O lines the logic state is defined as "in" or "out." Input or output nodes are defined only as "1" or "0."

MOTOROLA 10800 RALU 4 BIT (MSI)

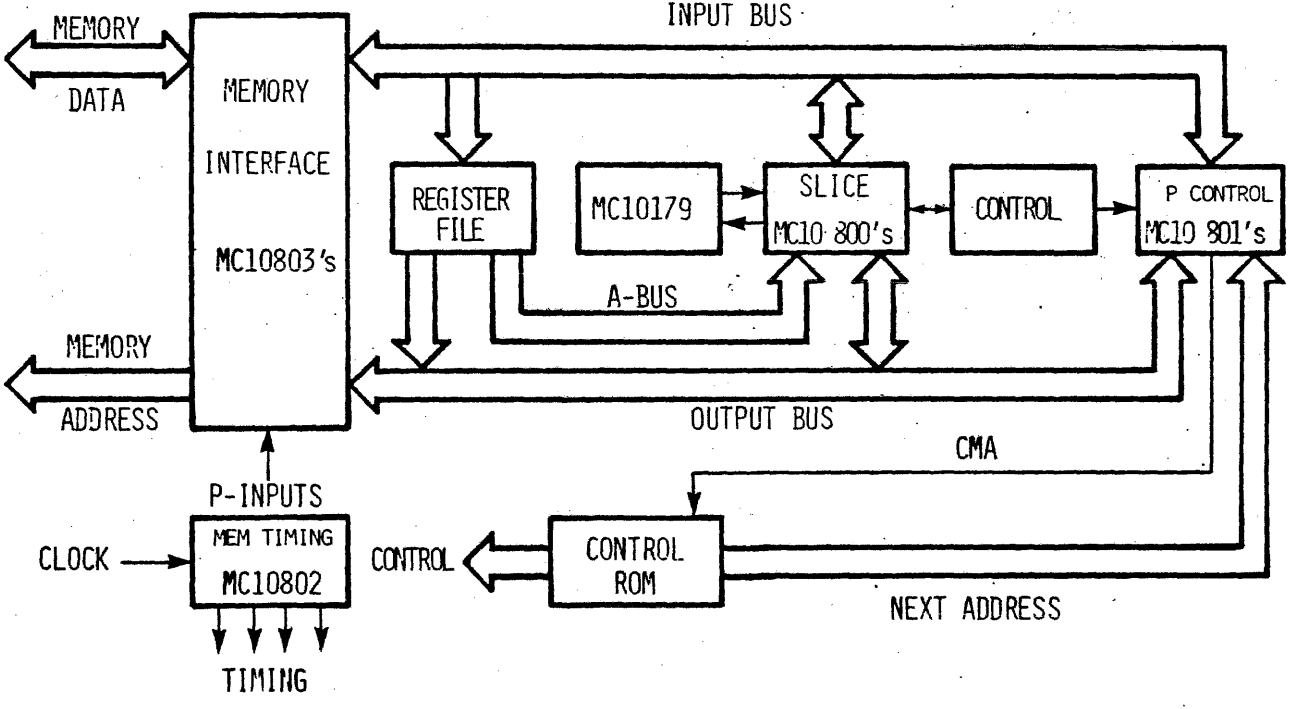
50 NS ECL IMPLEMENTATION
1 ACCUMULATOR, 1 INPUT LATCH
BCD FUNCTIONS

FAIRCHILD MACRO LOGIC SERIES

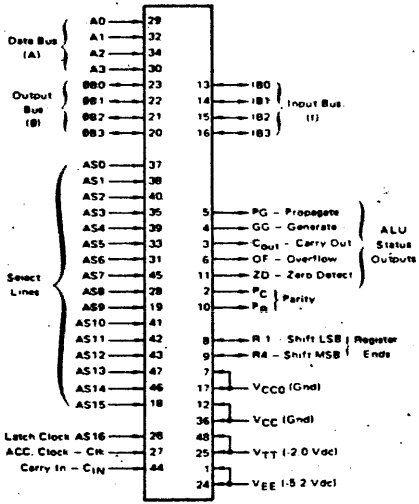
9405 RALU 4 BIT (MSI)
8 REGISTERS
ONE ALU INPUT "BARE"
9404 DATA PATH 4 BIT
SHIFTING, MULTIPLEXING, MASKING
9407 DATA ACCESS 4 BIT
ADDITION, MULTIPLEXING
HOLDING REGISTERS


CONTROL
ROM
REGISTER
SLICE
MICROCONTROL
MC10801
ALU SLICE
MC10800
TIMING
MC10802
MEMORY
INTERFACE
MC10803
ADDRESS
DATA

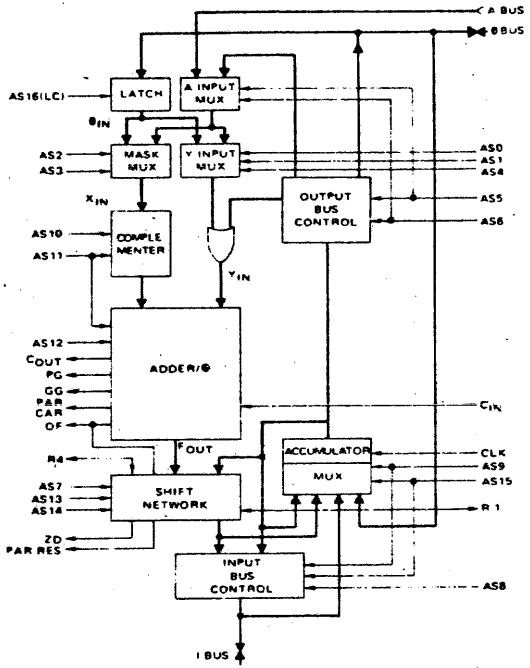
MOTOROLA BIT SLICE ORGANIZATION



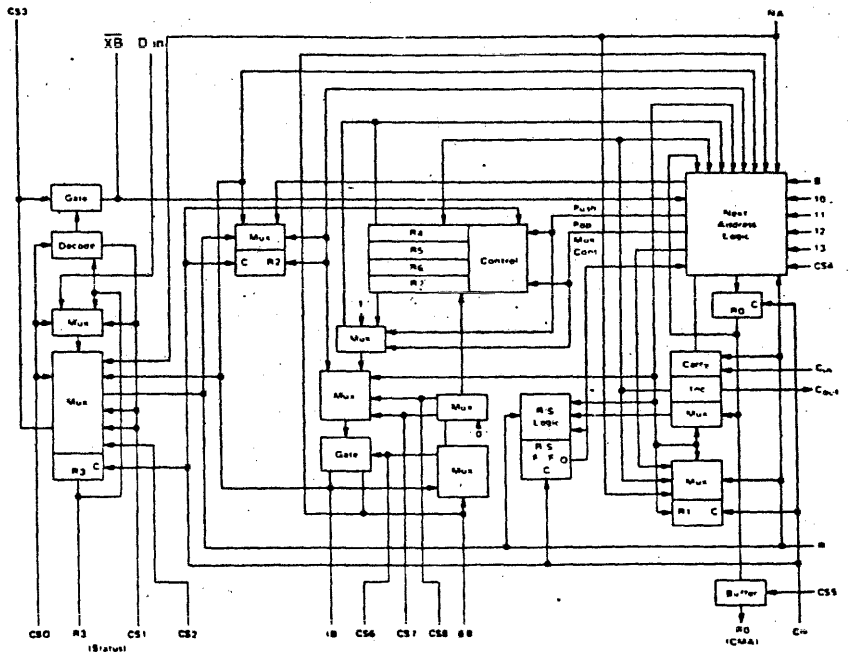
INPUT/OUTPUT DIAGRAM – MC10800



4-BIT SLICE BLOCK DIAGRAM – MC10800



SYBEX


CS3
XB
D in
NA
Gate
Decode
Mux
R3
C
R2
R4
R5
R6
R7
Control
Push
Pop
Mux Cont
Next Address Logic
8
10
11
12
13
CS4
R0
Carry
Inc
R1
RS Logic
RS FFO
Buffer
CS5
CS0
CS1
CS2
CS6
CS7
CS8

- ASSOCIATED TO SMS 360 IV BYTE

- AVAILABLE IN CARD AND MODULE FORM FROM SMS.
  MODIFIED CHIPS AVAILABLE FROM SIGNETICS

- BIPOLAR DESIGN, 48-PIN DIP
  300 TO 600 NS/INSTRUCTION

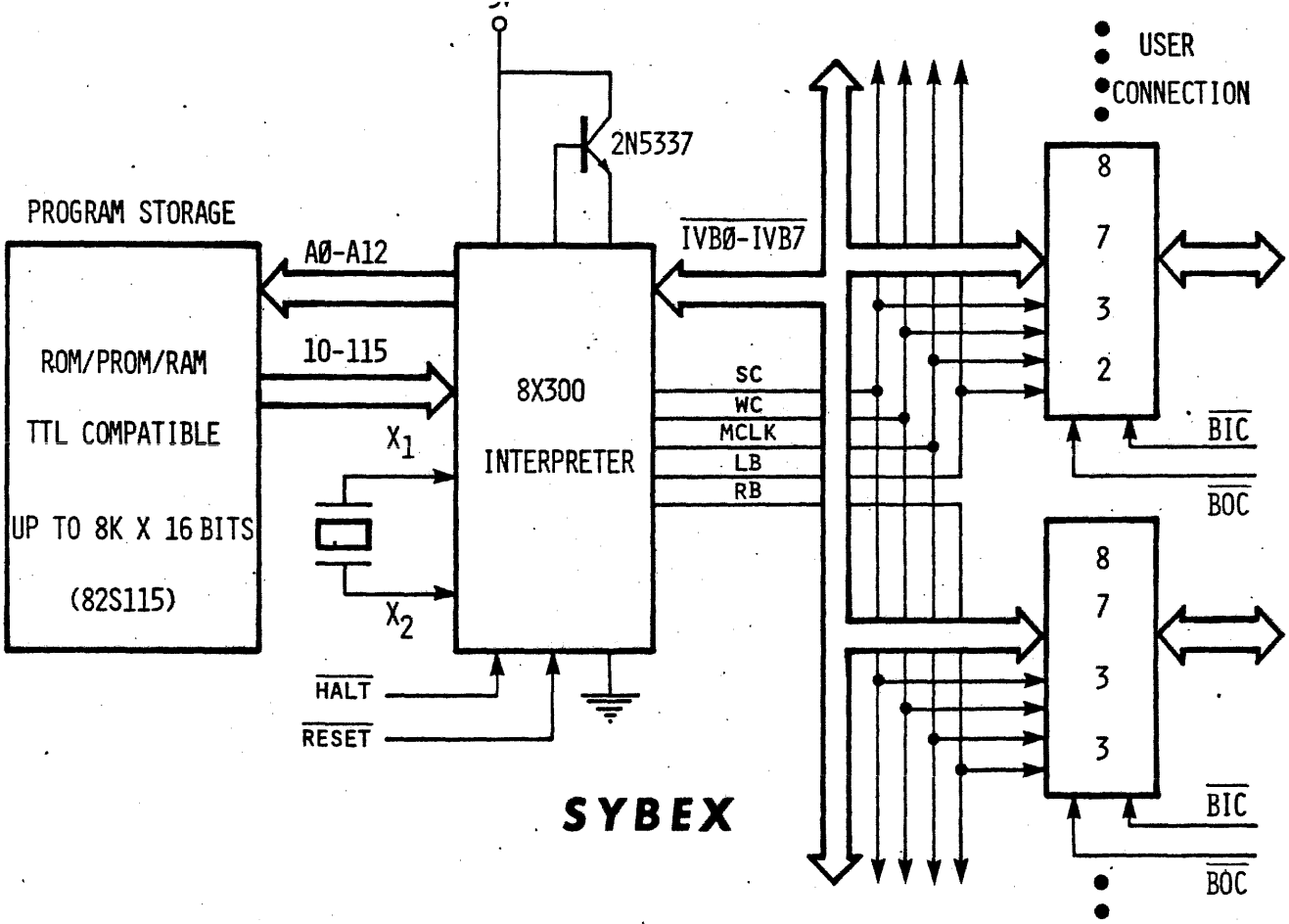
- 8 REGISTERS (ACCUMULATORS). NO STACK POINTER.

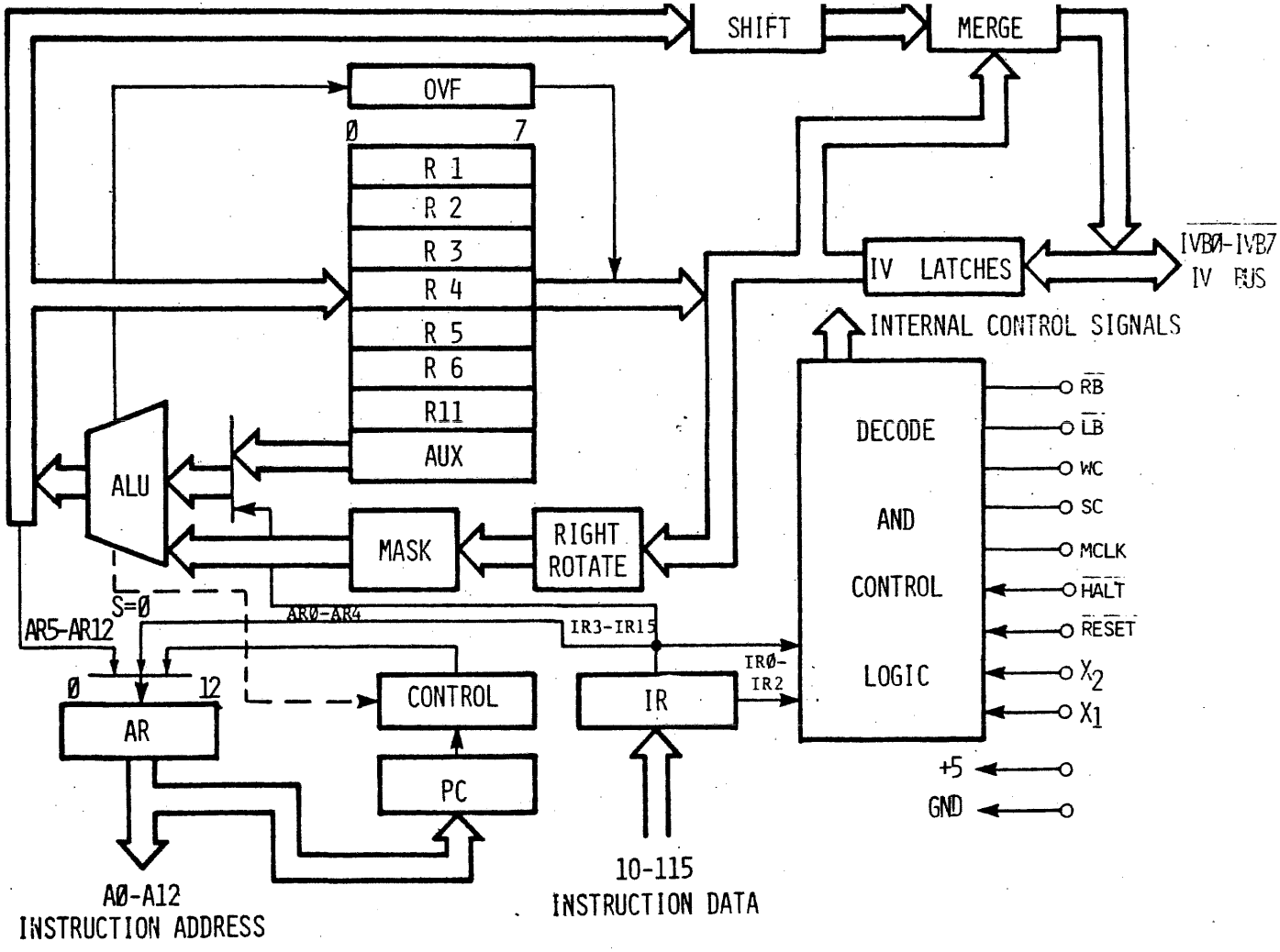
- 13-BIT PROGRAM COUNTER = 8 K ADDRESSING

- ORIENTED TOWARDS FAST SIGNAL PROCESSING WITH SMALL RAM

- NON-STANDARD INSTRUCTION SET

- 300 NS INSTRUCTION CYCLE
- 16-BIT MICROINSTRUCTIONS
- 13-BIT ADDRESS
- 8-BIT DATA
- 8 REGISTERS
- ON-CHIP OSCILLATOR
- BIPOLAR SCHOTTKY


PROGRAM STORAGE
ROM/PROM/RAM
TTL COMPATIBLE
UP TO 8K X 16 BITS
(82S115)
A0-A12
10-115
X1
X2
8X300
INTERPRETER
2N5337
IVB0-IVB7
SC
WC
MCLK
LB
RB
HALT
RESET
USER
CONNECTION
8
7
3
2
BIC
BOC
8
7
3
3
BIC
BOC
SYBEX


SHIFT
MERGE
OVF
0
7
R 1
R 2
R 3
R 4
R 5
R 6
R11
AUX
IV LATCHES
IVB0-IVB7
IV BUS
INTERNAL CONTROL SIGNALS
ALU
MASK
RIGHT ROTATE
DECODE
AND
CONTROL
LOGIC
RB
LB
WC
SC
MCLK
HALT
RESET
X2
X1
+5
GND
S=0
AR5-AR12
AR0-AR4
IR3-IR15
IR0-IR2
0
12
AR
CONTROL
IR
PC
A0-A12
INSTRUCTION ADDRESS
10-115
INSTRUCTION DATA

- LOW-POWER SCHOTTKY

- 50 NS CYCLE TIME

- 10-BIT ADDRESS = 1K MICROINSTRUCTION ADDRESSABILITY

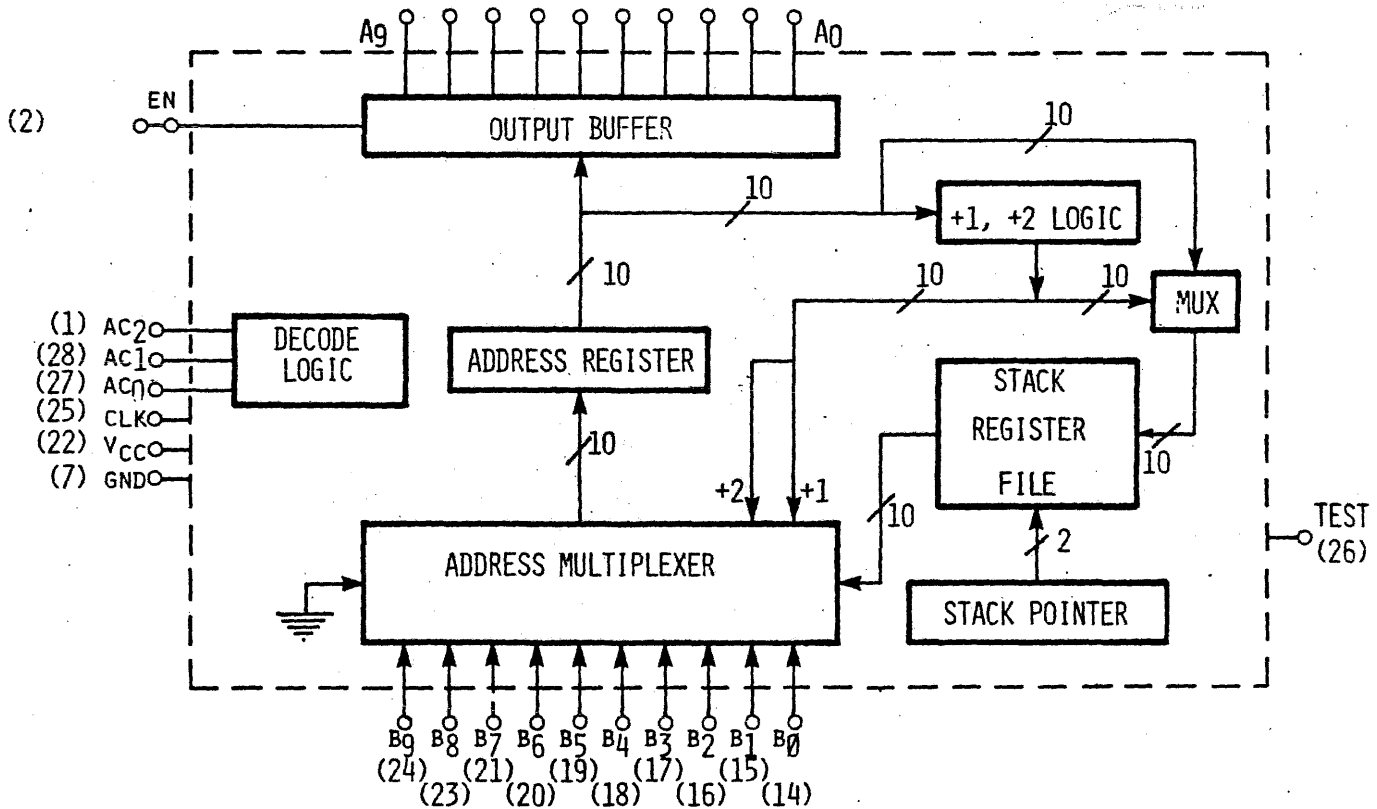
- N-WAY BRANCH

- 4-LEVEL STACK

- TEST AND SKIP ON INPUT LINE

- 3-BIT COMMAND CODE

- AUTO-RESET TO ADDRESS Ø DURING POWER-UP


A9
A0
EN
(2)
OUTPUT BUFFER
10
10
+1, +2 LOGIC
10
10
10
MUX
(1) AC2
(28) AC1
(27) AC0
(25) CLK
(22) VCC
(7) GND
DECODE LOGIC
ADDRESS REGISTER
10
STACK REGISTER FILE
10
+2
+1
10
TEST (26)
ADDRESS MULTIPLEXER
2
STACK POINTER
B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
(24) (23) (21) (20) (19) (18) (17) (16) (15) (14)

5. BIT-SLICE APPLICATIONS

SYBEX

IMP - 16

MM 300/600 (NOVA EMULATION)

INTEL DESIGN PROJECT 16 BIT CPU

INTEL DISK CONTROLLER

SPECIAL ARCHITECTURES

- FLOATING POINT PROCESSOR
- LARGE SCALE COMPUTERS (CHIP REDUCTION/COST REDUCTION)
- HIGH LEVEL LANGUAGE INTERPRETERS
- CHARACTER STRING PROCESSORS

MICRO APPLICATIONS

- MULTI CHANNEL SERIAL I/O ADAPTOR
- SERIAL NIBBLE PROCESSOR
- LSI MICROCOMPUTER EMULATIONS

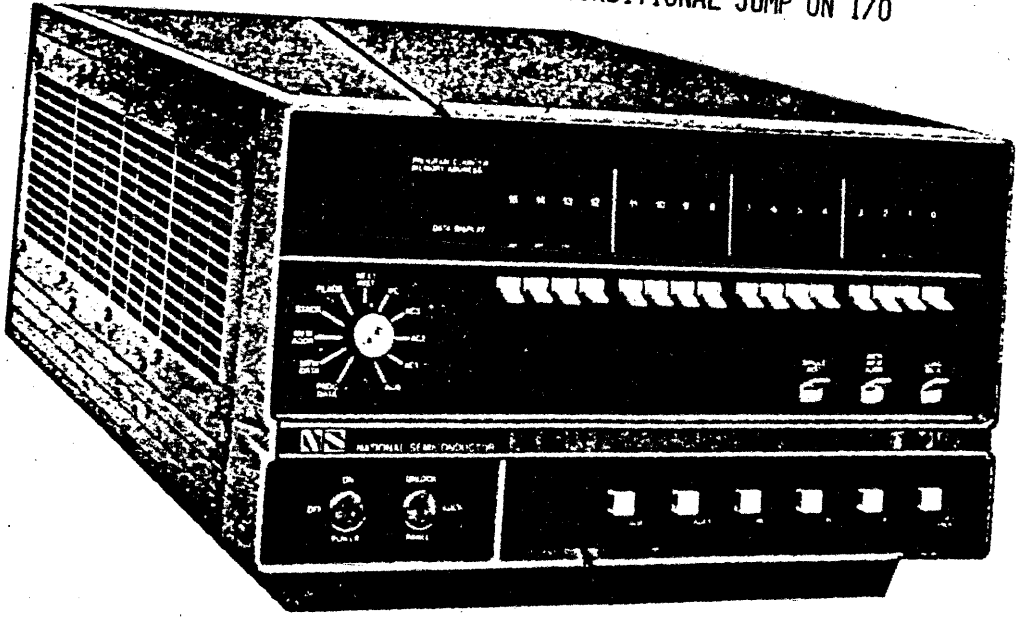
45 INSTRUCTIONS STANDARD (MORE IN OPTIONAL ROM)

MEMORY - UP TO 65K 16 BIT WORDS

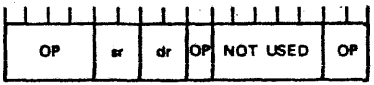
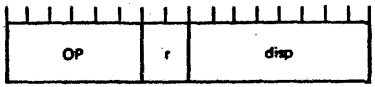
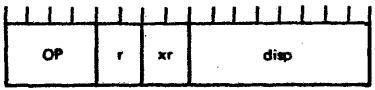
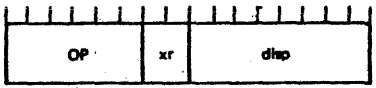
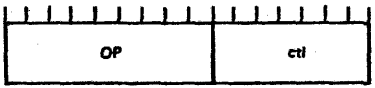
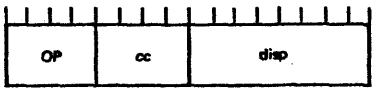
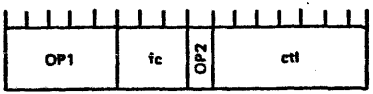
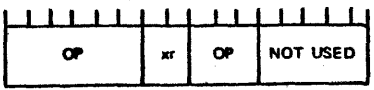
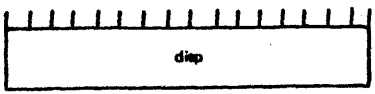
PAGE ADDRESSING (256 WORDS) / INDEXING OUT OF PAGE

1.4 uS MICROCYCLE

COMPREHENSIVE I/O BUSSING / CONDITIONAL JUMP ON I/O



IMP 16P INSTRUCTION FORMATS

Instruction Type	Machine Format
Register to Register	
Register to Memory	
Memory Reference (Class 1)	
Memory Reference (Class 2)	
I/O and Miscellaneous	
Branch	
Control Flags	
Memory Reference (Double Word)	 

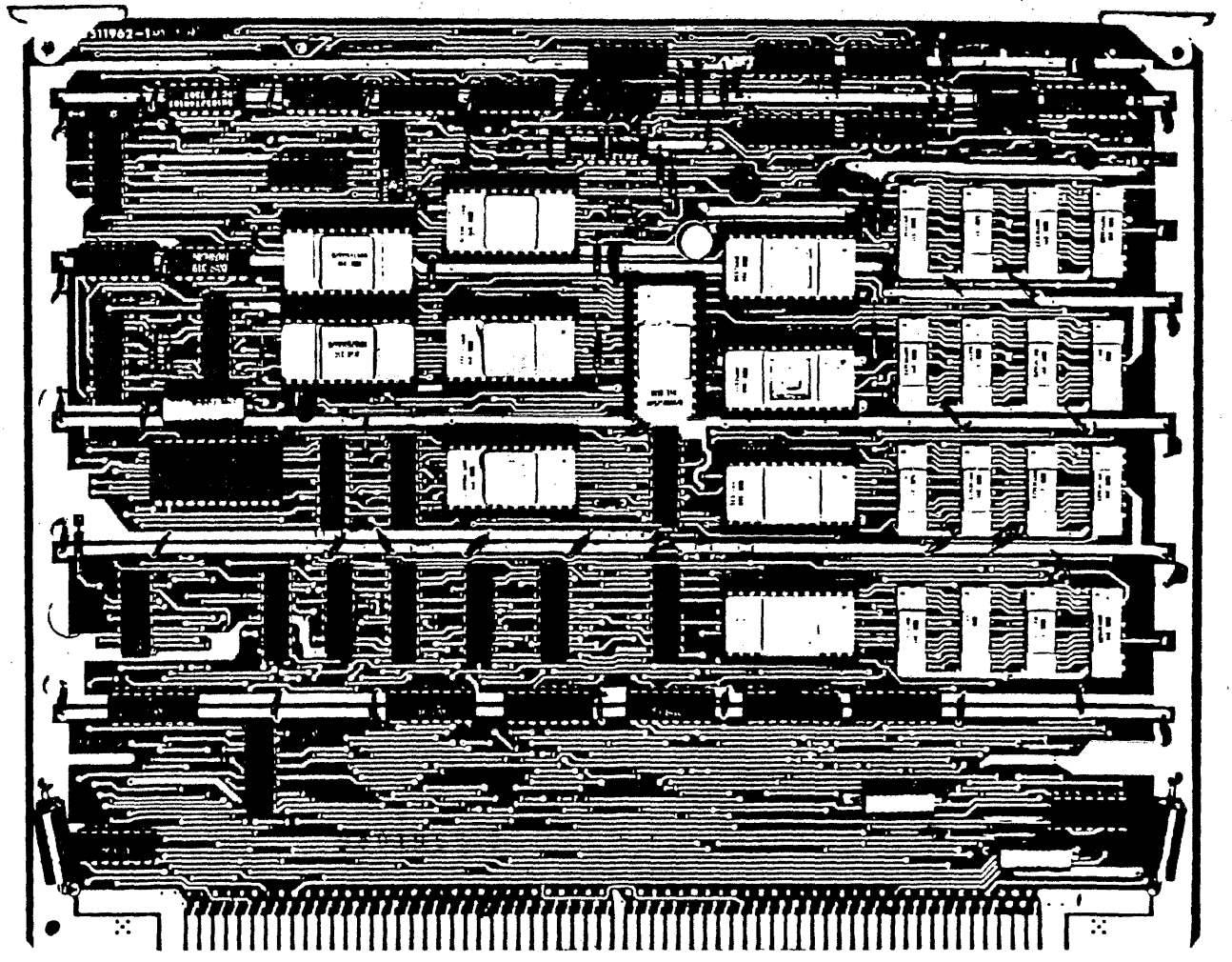
Explanation of Symbols

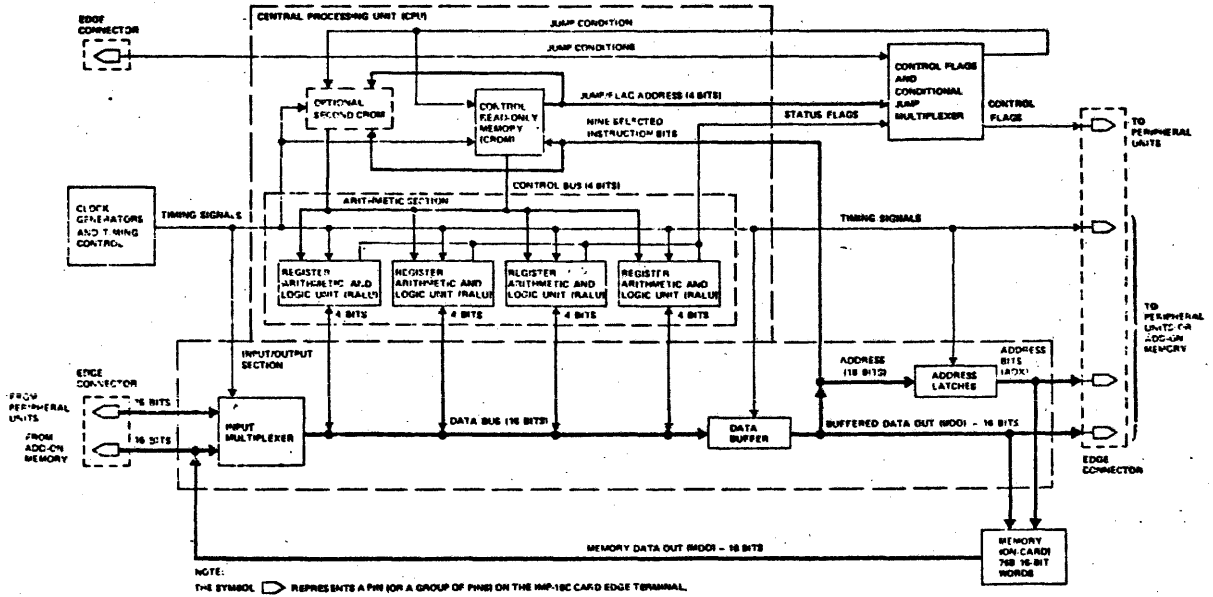
- Op – Instruction Mnemonic
- OP – Operation Code
- sr – Source Register Value
- dr – Destination Register Value
- xr – Index Register Value (2 or 3)
- disp – Displacement Value
- cc – Condition Code Value
- r – Register Value
- ctl – Control Bits Value

Instruction	Mnemonic	Execution Cycles	Memory Read Cycles	Memory Write Cycles
erence Instructions				
	LD	5	2	–
irect[1]	LD	5	3	–
	ST	6	1	1
irect[1]	ST	8	2	1
	ADD	5	2	–
	SUB	5	2	–
	JMP	3	1	–
irect[1]	JMP	5	2	–
Subroutine	JSR	4	1	–
Subroutine Indirect[1]	JSR	6	2	–
t and Skip if Zero	ISZ	7,8 if SKIP	2	1
t and Skip if Zero	DSZ	8,9 if SKIP	2	1
ND is Zero	SKAZ	6,7 if SKIP	2	
eater	SKG	Like Signs: 8,9 if SKIP	2	
erence Instructions				
o Stack Register	PUSH	3	1	
Stack	PULL	3	1	
ediate, Skip if Zero	AISZ	4,5 if SKIP	1	
nediate	LI	3	1	
ent and Add Immediate	CAI	3	1	
Copy	RCPY	6	1	
Register and Top of Stack	XCHRS	5	1	
Registers	RXCH	8	1	
And	RAND	6	1	
Exclusive Or	RXOR	6	1	
Add	RADD	3	1	
	SHL	4 + 3K	1	
ht	SHR	4 + 3K	1	
eft	ROL	4 + 3K	1	
ight	ROR	4 + 3K	1	

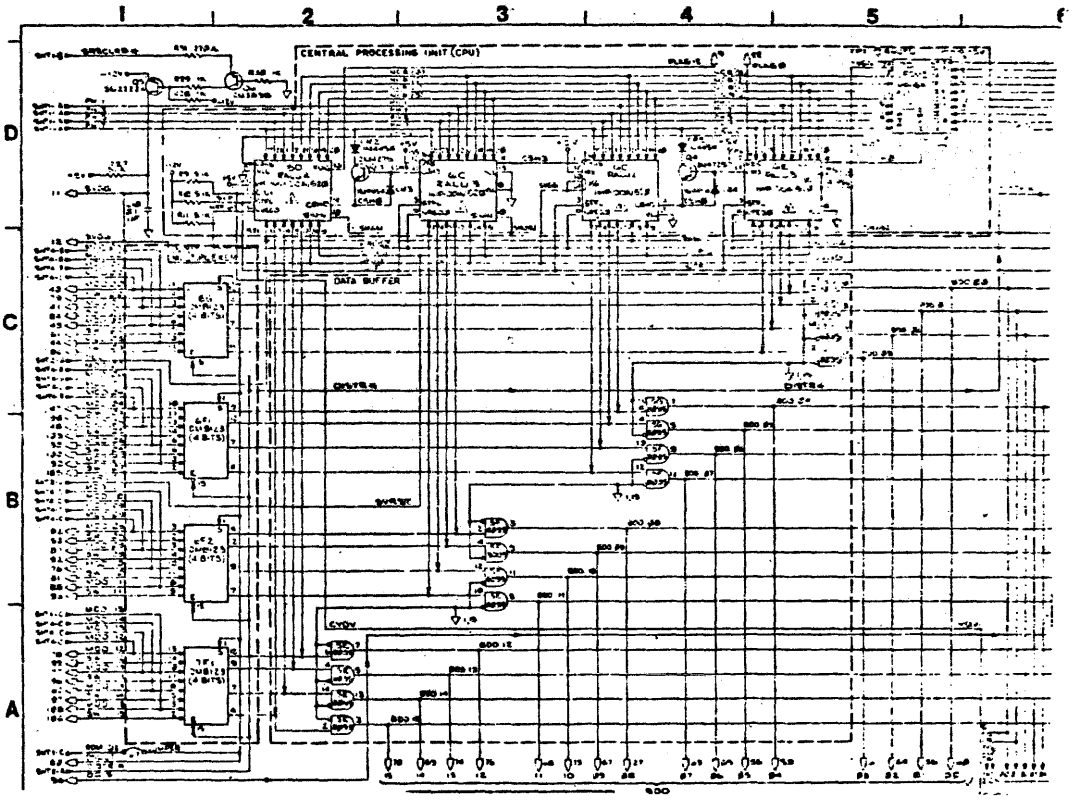
Table A–2. INP 16C Extended Instruction Set (Executed by CROM II)

Instruction	Mnemonic	Execution Cycles	Memory Read Cycles	Memory Write Cycles
Multiply	MPY	108 to 122	3	–
Divide	DIV	125 to 159	3	–
Double Precision Add	DADD	12	4	–
Double Precision Subtract	DSUB	12	4	–
Load Byte	LDB	20 (left) 12 (right)	4	–
Store Byte	STB	24 (left) 17 (right)	4	1
Set Status Flag	SETST	17 to 36	1	–
Clear Status Flag	CLRST	17 to 36	1	–
Skip If Status Flag True	SKSTF	19 to 39	1	–
Set Bit	SETBIT	15 to 34	1	–
Clear Bit	CLRBIT	15 to 34	1	–
Complement Bit	CMPBIT	15 to 34	1	–
Skip If Bit True	SKBIT	19 to 39	1	–
Interrupt Scan	ISCAN	9 to 80	1	–
Jump Indirect to Level Zero Interrupt	JINT	7	2	–
Jump Through Pointer	JMPP	7	3	–
Jump to Subroutine Through Pointer	JSRP	8	3	–




EDGE CONNECTOR
CENTRAL PROCESSING UNIT (CPU)
JUMP CONDITION
JUMP CONDITIONS
OPTIONAL SECOND CROM
CONTROL READ-ONLY MEMORY (CROM)
JUMP/FLAC ADDRESS (4 BITS)
NINE SELECTED INSTRUCTION BITS
STATUS FLAGS
CONTROL FLAGS AND CONDITIONAL JUMP MULTIPLEXER
CONTROL FLAGS
TO PERIPHERAL UNITS
CLOCK GENERATORS AND TIMING CONTROL
TIMING SIGNALS
CONTROL BUS (4 BITS)
ARITHMETIC SECTION
REGISTER ARITHMETIC AND LOGIC UNIT (RALU)
REGISTER ARITHMETIC AND LOGIC UNIT (RALU)
REGISTER ARITHMETIC AND LOGIC UNIT (RALU)
REGISTER ARITHMETIC AND LOGIC UNIT (RALU)
4 BITS
4 BITS
4 BITS
4 BITS
TIMING SIGNALS
TO PERIPHERAL UNITS OR ADD-ON MEMORY
INPUT/OUTPUT SECTION
EDGE CONNECTOR
FROM PERIPHERAL UNITS
FROM ADD-ON MEMORY
16 BITS
16 BITS
INPUT MULTIPLEXER
DATA BUS (16 BITS)
DATA BUFFER
ADDRESS (16 BITS)
ADDRESS LATCHES
ADDRESS BITS (MDX)
BUFFERED DATA OUT (MDO) – 16 BITS
EDGE CONNECTOR
MEMORY DATA OUT (MDO) – 16 BITS
MEMORY (OR CARD) FOR 16 BIT WORDS
NOTE:
THE SYMBOL REPRESENTS A PIN (OR A GROUP OF PINS) ON THE IMP-16C CARD EDGE TERMINAL.

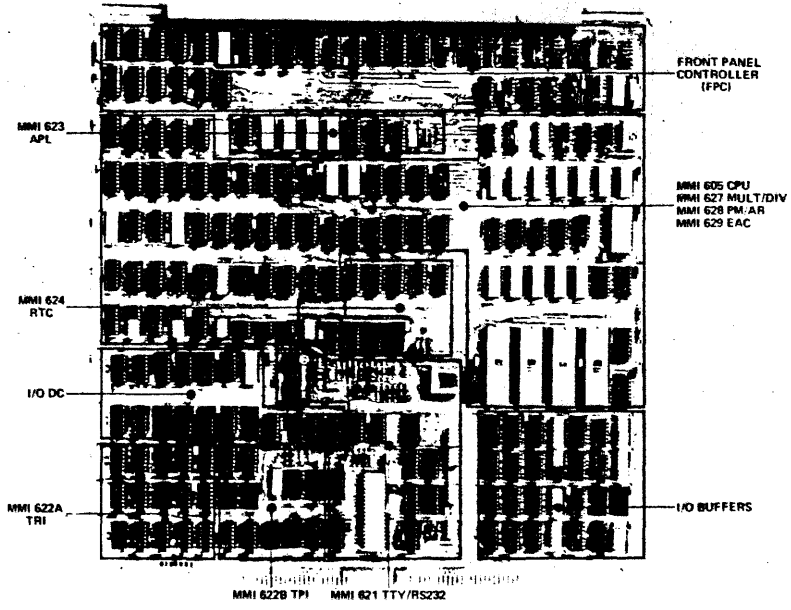
IMP 16P DETAIL OF PROCESSOR CIRCUIT: RALUS AND CROM



NS MICROCYCLE

NS CYCLE MOS MEMORY

MM 600 PROCESSOR CARD

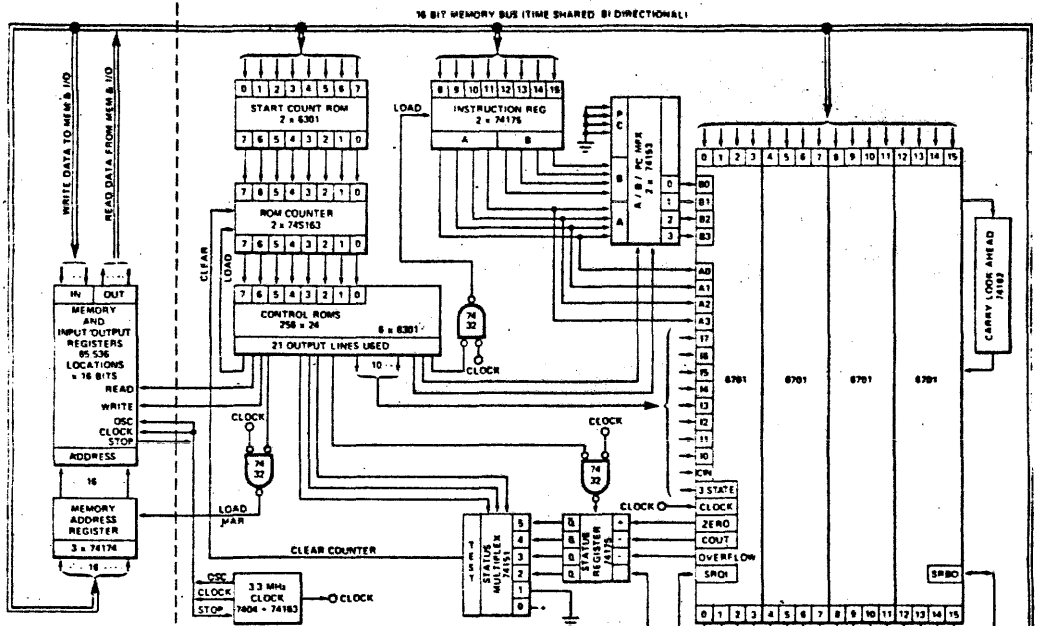


SYBEX

MM 600: MACRO INSTRUCTION TIMING

Instruction Mnemonic	Execution Time (μs)	Instruction Mnemonic	Execution Time (μs)
ISZ	3.0	JMP@	2.3
DSZ	3.0	NIOSCPU (INTEN)	3.5
JMP	1.0	NIOCCPU (INTDS)	3.5
JSR	1.9	DIACPU (READS)	3.5
LDA●	2.0	DOBCPU (MASKO)	3.5
STA	2.0	DIBCPU (INTA)	3.5
NIO	3.5	DICCPU (IORST)	3.5
SKPBN	3.2*	DOCCPU (HALT)	3.5
SKPBZ	3.2*	SKPBNCPU	3.2*
SKPDN	3.2*	DKPBZCPU	3.2*
SKPDZ	3.2*	SKPDNCPU	3.2*
DIA,B,C	3.5	SKPDZCPU	3.2*
DOA,B,C	3.5	DMA IN	2.4
COM	1.3	DMA OUT	2.1
NEG	1.3	DMA INC	3.6
MOV	1.3	DMA ADD	4.7
INC	1.3	MULT	9.2
ADC	1.3	DIV	10.1
SUB	1.3	LDA@	3.0
ADD	1.3	STA@	3.0
AND	1.3	JMP@@	3.3

* Add 0.3 μs if skip occurs.



MM 600 CPU IS SIMILAR

INTEL DESIGN PROJECT CPU USING 3000 SERIES

16 BIT PROCESSOR

PIPELINED MICRO CONTROL

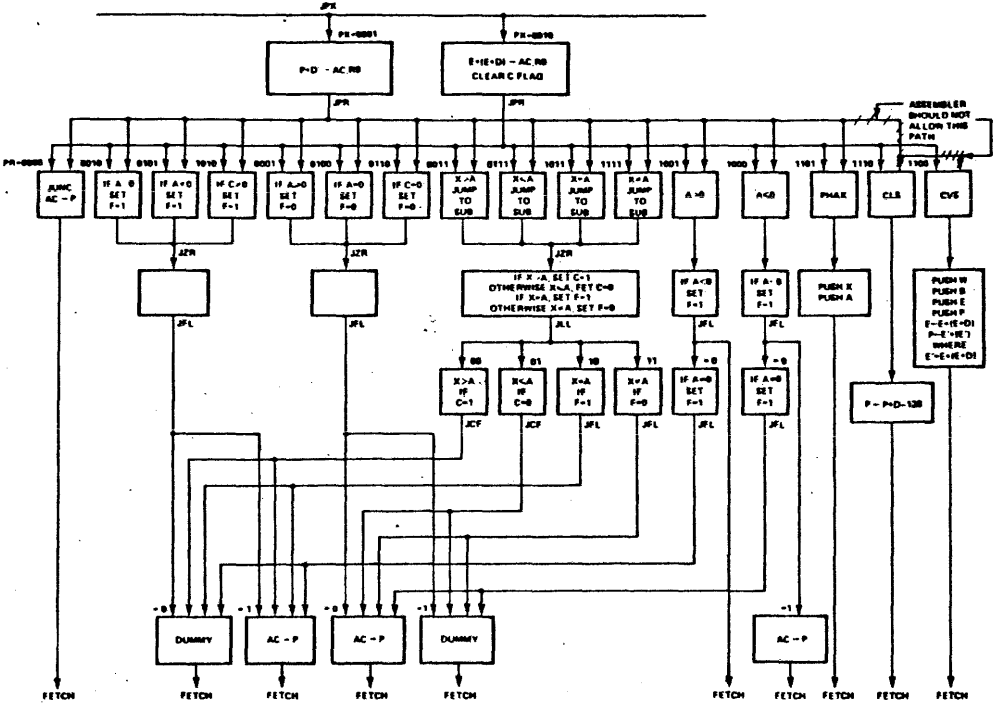
LOOK AHEAD CARRY

~200 NS CYCLE

~30 CHIPS

COMPREHENSIVE MACRO INSTRUCTION SET

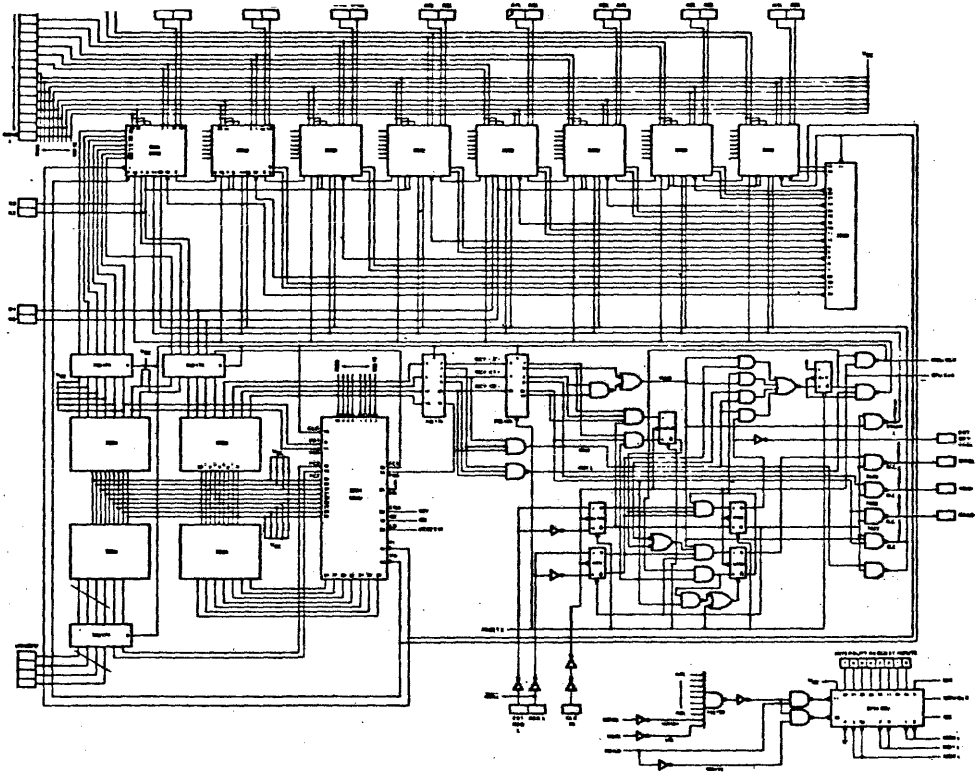
IMPLEMENTED IN 256 MICRO INSTRUCTIONS



Jump Group Flowchart

EXAMPLE OF MICROPROGRAM MEMORY

	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
	JCC	JFL	JZR	JZR	JFL	JFL	JZR	JFL	JCC	JRL	JCF	JCC	JCC	JCC	JCC	JCC
	0090H	0072H	000FH	000FH	0062H	0002H	000FH	0042H	0028H	007CH	006AH	0048H	00FCH	00B0H	001EH	001FH
000A																
	67	296	316	317	319	316	506	302	580	190	183	335	92	394	663	100
	0	1	4	4	3	3	2	1	2	1	1	4	1	2	1	38
	JLL	JCC	JCR	JLL	JCF	JCR	JCR	JCR	JCR	JCF	JCR	JCR	JCH	JZR	JCC	JCC
	0024H	0041H	0010H	0024H	0022H	0014H	0018H	0016H	0019H	001AH	001CH	001DH	001UH	000FH	0049EH	009FH
001H																
	361	74	360	366	505	504	511	510	512	513	517	514	516	515	464	104
	1	1	1	1	2	1	2	1	1	1	1	1	1	2	1	1
	JAH	JCC	JCH	JCR	JCF	JCF	JFL	JFL	JZR	JZR	JCR	JCR		JCR	JLL	JCC
	000FH	0011H	0020H	0020H	0062H	0002H	0062H	0002H	000FH	000FH	0028H	0029H		002FH	0014H	004FH
002H																
	506	73	506	507	370	371	372	373	581	477	472	476		459	502	460
	1	1	1	1	3	3	3	3	1	1	1	1		1	1	1
	JZH	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JCC	JCC	JCC	JZH
	000FH	000DH	0004H	000AH	0005H	0004H	0005H	000BH	0007H	0001H	0004H	000BH	005CH	004DH	00AEH	000BH
003H																
	266	290	291	328	292	293	309	329	301	295	310	330	380	411	496	331
	3	2	2	2	2	2	2	2	2	2	2	2	3	2	2	2
	JCC	JCH	JFL	JZH	JCC	JCC	JCC	JCR	JZR	JCR	JCR	JCC	JCC	JCH	JCR	JZH
	007UH	0044H	0002H	000FH	00F4H	0085H	0056H	0045H	000CH	004AH	0049H	005BH	003CH	004DH	004CH	000EH
004H																
	413	75	304	305	79	208	195	207	87	86	85	336	564	412	563	462
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	JCC	JFL	JFL	JLL	JRL	JZR	JCH	JCC	JCR	JCC	JCR	JCR	JCC	JCC	JCH	JCH
	0060H	0052H	0012H	0024H	007CH	000AH	0055H	0047H	0059H	0069H	0051H	005AH	005DH	00ADH	0056H	0058H
005H																
	280	346	356	351	177	182	201	206	597	598	341	337	381	382	217	593
	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
	JPH	JPR	JZH	JZR	JZR	JZR	JCC	JCR	JCC	JCR	JZR	JRL	JCR	JCR	JZR	JCC
	0036H	0030H	000FH	000FH	000FH	000FH	00F6H	0066H	00F8H	006CH	0009H	007CH	0086H	006FH	000DH	00AFH
006H																
	281	272	321	322	236	237	238	242	243	599	189	184	603	607	407	608
	1	1	4	4	1	1	1	1	1	1	1	1	1	1	1	1
	JCC	JCC	JFL	JZH	JZR	JZR	JCR	JCR	JCC	JCC	JCR	JCC	JCC	JCC	JLL	JLL
	00F0H	0061H	0062H	000FH	000FH	000FH	0078H	0079H	00A8H	00F9H	007BH	00E8H	009CH	009DH	0064H	0074H
007H																
	415	271	298	299	250	251	255	262	256	263	386	388	224	229	234	248
	1	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3



SYBEX

INTEL DISK CONTROLLER USING 3000 SERIES

CONTROLS UP TO FOUR DISKS (2310/5440)

DRIVE RATE 2.5 MHz

PERFORMS ALL DATA CHANNEL FUNCTIONS

STATUS CHECKING

COMMAND EXECUTION

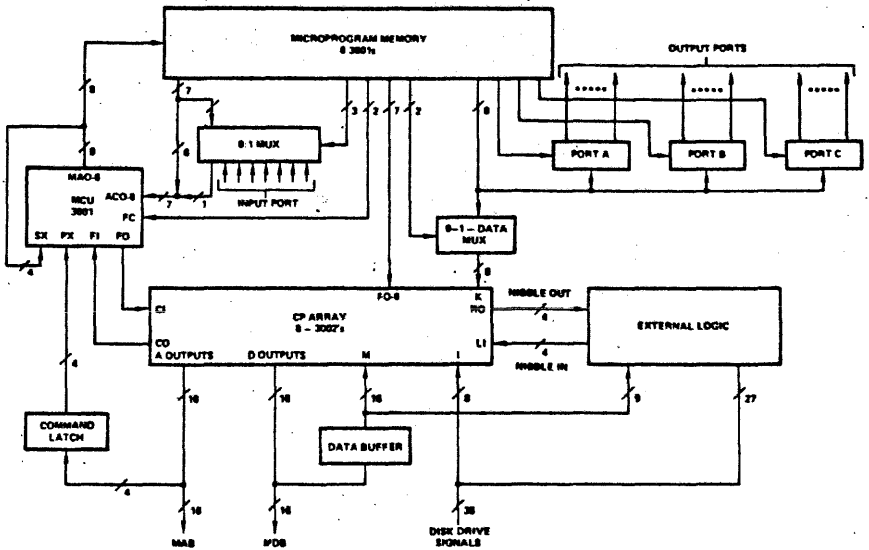
16 BIT CPE ARRAY

400 NS CLOCK (NO PIPELINE, NO LOOK AHEAD)

SPECIAL NIBBLE TO WORD TO NIBBLE ARCHITECTURE

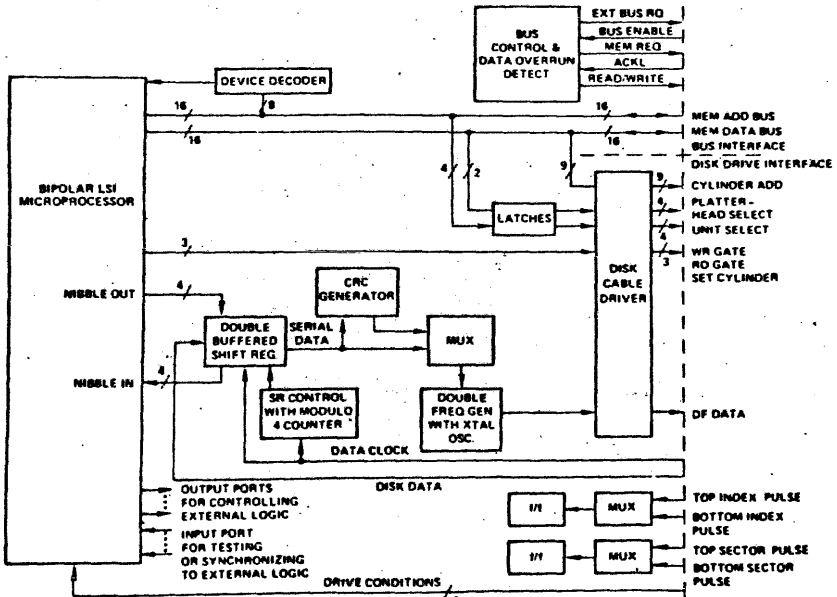
230 MICROINSTRUCTION IMPLEMENTATION (32 BIT)

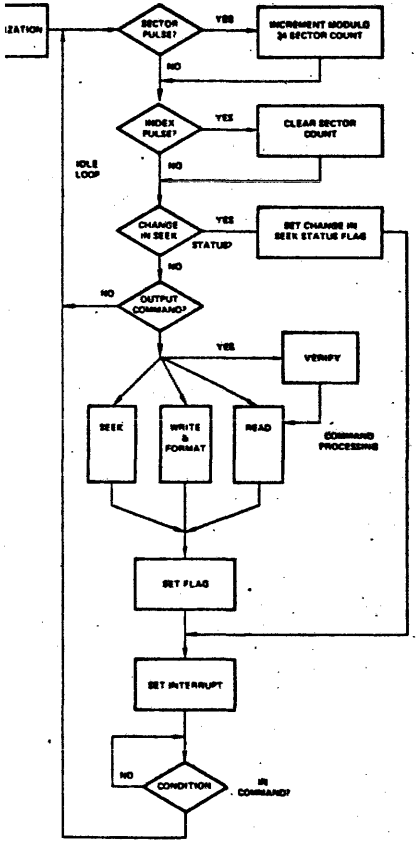
67 CHIP ON 8 IN. × 15 IN. CARD

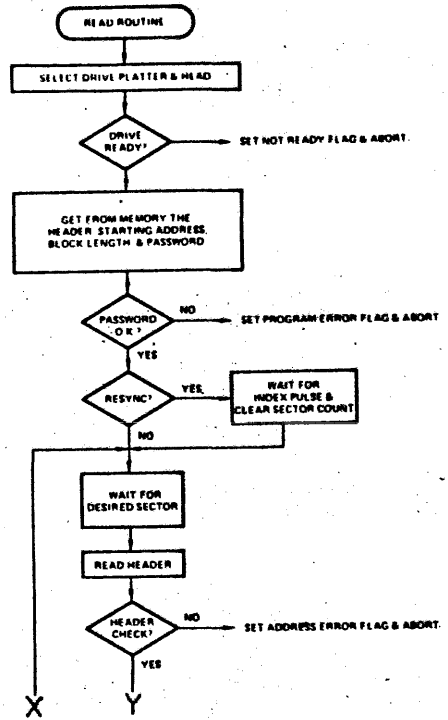

MICROPROGRAM MEMORY
8 3001s
OUTPUT PORTS
8:1 MUX
PORT A
PORT B
PORT C
MAO-8
MCU
3001
ACO-8
FC
SX
PX
FI
PO
INPUT PORT
9-1 – DATA
MUX
FO-8
CI
CO
CP ARRAY
8 – 3002's
E
RO
LI
NIBBLE OUT
NIBBLE IN
EXTERNAL LOGIC
A OUTPUTS
B OUTPUTS
M
I
COMMAND
LATCH
DATA BUFFER
MAB
MDB
DISK DRIVE
SIGNALS

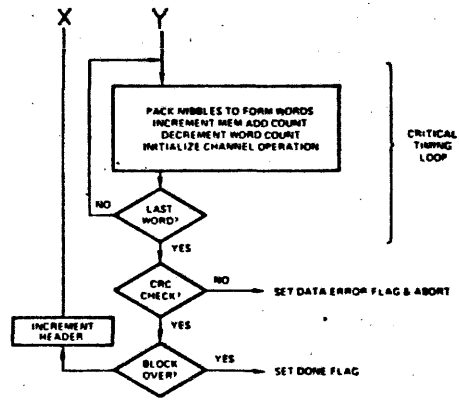
SYBEX

DISK CONTROLLER - SPECIAL LOGIC DIAGRAM




IZATION
SECTOR PULSE?
YES
INCREMENT MODULO 24 SECTOR COUNT
NO
INDEX PULSE?
YES
CLEAR SECTOR COUNT
NO
IDLE LOOP
CHANGE IN SEEK STATUS?
YES
SET CHANGE IN SEEK STATUS FLAG
NO
NO
OUTPUT COMMAND?
YES
VERIFY
SEEK
WRITE & FORMAT
READ
COMMAND PROCESSING
SET FLAG
SET INTERRUPT
NO
CONDITION
IN COMMAND?


READ ROUTINE
SELECT DRIVE PLATTER & HEAD
DRIVE READY?
SET NOT READY FLAG & ABORT
GET FROM MEMORY THE HEADER STARTING ADDRESS, BLOCK LENGTH & PASSWORD
PASSWORD OK?
NO
SET PROGRAM ERROR FLAG & ABORT
YES
RESYNC?
YES
WAIT FOR INDEX PULSE & CLEAR SECTOR COUNT
NO
WAIT FOR DESIRED SECTOR
READ HEADER
HEADER CHECK?
NO
SET ADDRESS ERROR FLAG & ABORT
YES
X
Y


X
Y
PACK NIBBLES TO FORM WORDS
INCREMENT MEM ADD COUNT
DECREMENT WORD COUNT
INITIALIZE CHANNEL OPERATION
CRITICAL TIMING LOOP
NO
LAST WORD?
YES
CRC CHECK?
NO
SET DATA ERROR FLAG & ABORT
YES
INCREMENT HEADER
BLOCK OVER?
YES
SET DONE FLAG

SYBEX

BLANK

APPLICATION OF BIT-SLICE DEVICES TO "NUMBER CRUNCHING"

ARITHMETIC OPERATIONS ON FLOATING POINT DATA

HIGH LEVEL FUNCTIONS

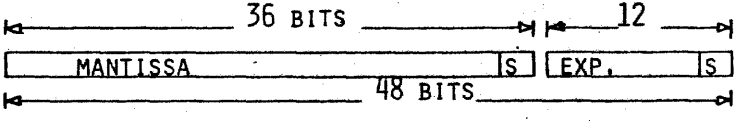
LOGARITHMIC	VECTOR
TRIGONOMETRIC	MATRIX

HIGH LEVEL LANGUAGES: PL - 1, ALGOL, APL, FORTRAN

FLOATING POINT DATA (BINARY SCIENTIFIC NOTATION)

$$\text{MANTISSA} \times 2^{\text{EXPONENT}}$$

EXAMPLE



>10 DECIMAL DIGITS ACCURACY

RANGE 2^{-2048} TO 2^{+2048} APPROX 10^{-600} TO 10^{+600}

FLOATING POINT OPERATIONS

ADDITION

REQUIRES SCALING

$M \times 2^{E} + M' \times 2^{E'} =$ NOTE: MANTISSA IS CONSIDERED FRACTIONAL (I.E. M 1)

IF $E = E'$ THEN

$M + M' \times 2^{E}$

ELSE EQUALIZE THE EXPONENTS BY RAISING THE SMALLER/SCALING ITS FRACTION

LET $E = E' + N$

THEN $M' \times 2^{E'} = (M' / 2^{N}) \times 2^{E}$

NOTE: $M' / 2^{N}$ IS M' SHIFTED DOWN N BITS

IN GENERAL, FLOATING POINT ADDITION:

LET E E' SUCH THAT $E = E' + N$

$$M \times 2^{E} + M' + 2^{E'} = M + (M' / 2^{N}) \times 2^{E}$$

THE SCALING OPERATION IS THE CRUX OF FLOATING ADD AND SUBTRACT

SUBTRACTION (AS IN ADDITION)

MULTIPLICATION

DOES NOT REQUIRE SCALING

ONLY HIGH ORDER PRODUCT IS USED

$$(M \times 2^{E}) \times (M' \times 2^{E'}) = (M \times M')_{H.O.} \times 2^{E+E'}$$

DIVISION

$$(M \times 2^{E}) \; / \; (M' \times 2^{E'}) = (M \; / \; M') \times 2^{E-E'}$$

FLOATING POINT NORMALIZATION

ALL FLOATING POINT OPERATIONS USE NORMALIZED DATA
AND RETURN NORMALIZED DATA

NORMALIZED = HIGH ORDER JUSTIFIED
I.E. THE BIT TO THE LEFT OF THE SIGN ≠ SIGN

THE LAST STEP OF ANY FLOATING POINT OPERATION
IS NORMALIZATION
I.E. SHIFTING MANTISSA UP UNTIL MSB ≠ SIGN
ADJUSTING EXPONENT DOWN TO COMPENSATE

MULTIPLY AND DIVIDE REQUIRE AT MOST 1 SHIFT TO NORMALIZE

ADD AND SUBTRACT MAY REQUIRE UP TO 34 SHIFTS
(AND MUST DETECT ZERO SUM OR DIFFERENCE)

FOR A 36 BIT MANTISSA

MULTIPLY AND DIVIDE TAKE 36 INTERATIONS:
ADD AND SUBTRACT MAY REQUIRE UP TO 34 BIT SHIFTS

EXAMPLE MULTIPLY

USING OPTIMAL 2901 CONFIGURATION FOR 36 BITS (W/O DBL LENGTH)
uCYCLE = 220 NS (175 NS)
36 ITERATIONS FOR MULTIPLY = 7.92 uSEC (6.30 uSEC)
(MANTISSA ADDITION CONCURRENT)
PLUS SET UP AND RESTORE 5 CYCLES = 1.10 uSEC (.88 uSEC)

FLOATING MUL: 9.02 uSEC (OR 7.18 FOR FLOATING ONLY CPU)

REDUCTION FOR FLOATING MULTIPLY

FOUR STAGE MULTIPLICATION OF MANTISSAS

$$M = M_H \times 2^{18} + M_L$$
$$M' = M'_H \times 2^{18} + M'_L$$
$$M \times M' = M_H \cdot M_H \cdot 2^{36} + (M_H \cdot M'_L + M'_H \cdot M_L) \cdot 2^{18} + M_L \cdot M'_L$$

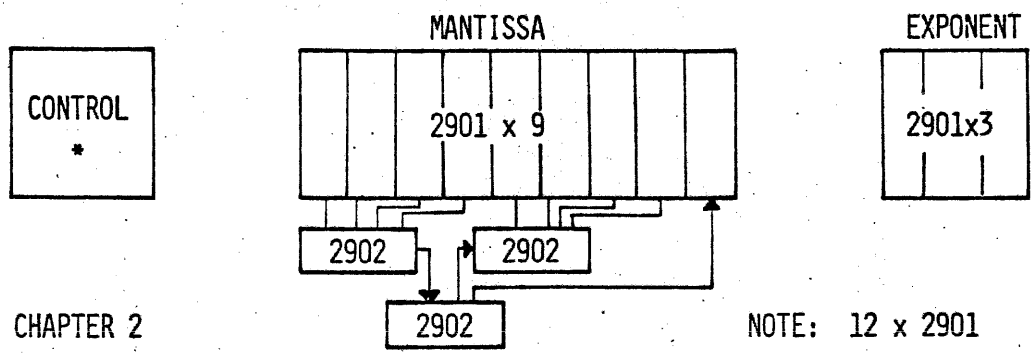
(NOTE: FOR CLOSE APPROXIMATION ONE CAN IGNORE LOWER 18 BITS OF 2^{18} PRODUCT
ALL 36 BITS OF 2^{0} PRODUCT)

TOTAL TIME REQUIRED

4 (OR 3) PARALLEL 18 BIT MULTIPLIES = 18 x 220 NS = 3.96 uSEC
(18 x 175 NS = 3.15 uSEC)
PLUS THREE 36 BIT ADDITIONS = 660 NS (525 NS)

FOUR STAGE FLOATING MUL: 4.62 uSEC (3.68 uSEC)

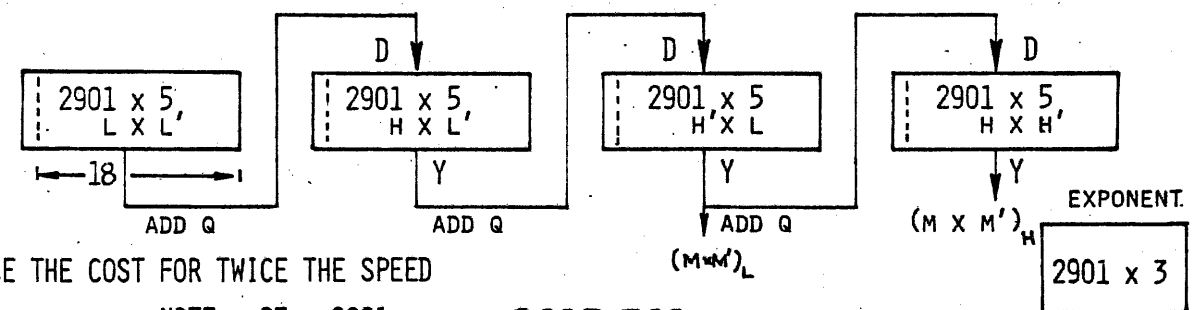
TRAIGHT FLOATING MULTIPLY



* AS IN CHAPTER 2

NOTE: 12 x 2901

OUR STAGE FLOATING MULTIPLY



TWICE THE COST FOR TWICE THE SPEED

NOTE: 23 x 2901

COST OF SCALING IN FLOATING ADD

ADDITION

THE TIME OF ADDITION IN FLOATING ADD IS NEGLIGIBLE

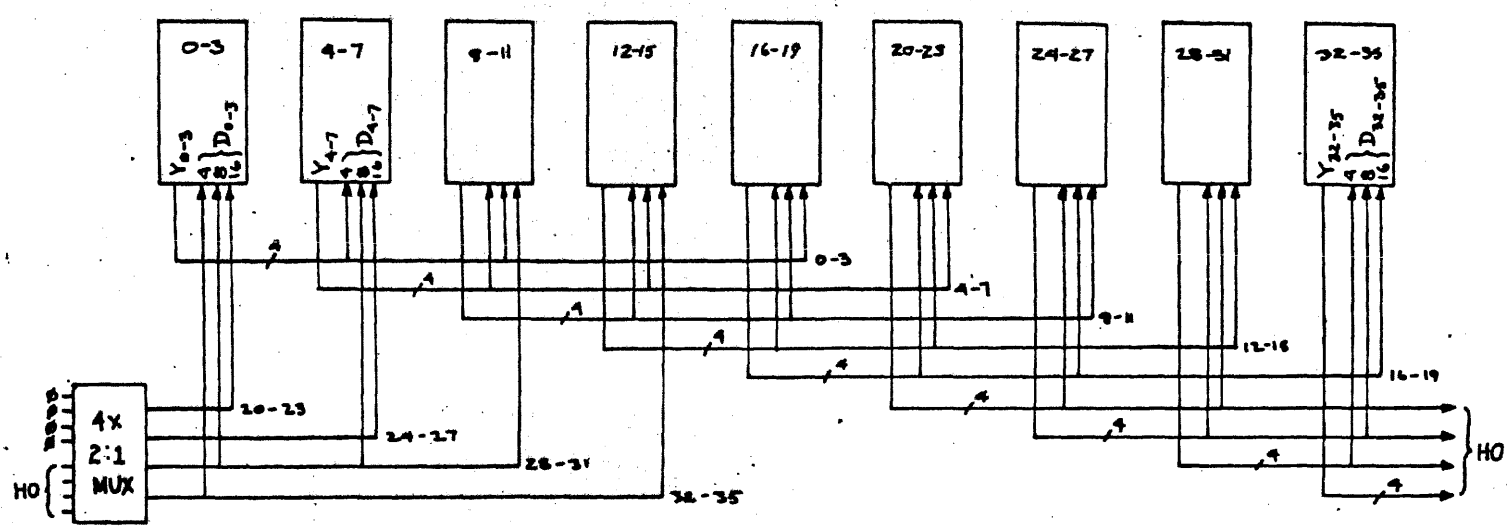
1 CYCLE - 220 NS

WITHOUT SPECIAL DATA PATHS SCALING MAY TAKE

34 x 220 NS = 7.48 uSEC

EXPONENTIAL SHIFTS CAN REDUCE TO

6 MAXIMUM x 220 = 1.32 uSEC (31 POSITIONS RIGHT, 5 POSITIONS LEFT)



SINGLE CYCLE SHIFT RIGHT 1, 4, 8, 16

SINGLE CYCLE SHIFT LEFT BY ROTATING RIGHT AND DISABLING MUX ABOVE SIGNIFICANT NIBBLE

PLA DECODES EXPONENT DIFFERENCES FOR SCALING SEQUENCE

PLA DECODES LEADING ZERO NIBBLES FOR NORMALIZE SEQUENCE

HARDWARE REQUIRED

18 DUAL 4 LINE TO 1 LINE MUX (74S253)

1 QUAD 2 LINE TO 1 LINE MUX (74S257)

1 PLA

PLA INPUTS DRIVEN FROM EXPONENT ELEMENTS

TOTAL FLOATING ADD <2 uSEC

FLOATING POINT SUMMARY — BIT SLICE APPLICATION

STRAIGHT FORWARD BIT SLICE F.P. IMPLEMENTATION FOR 36/12 DATA

MUL/DIV 7.2 TO 9 uSEC
ADD/SUB 6.4 TO 8 uSEC MAX

SPECIAL IMPLEMENTATION WITH BIT SLICES

MUL 3.7 TO 4.6 uSEC
ADD/SUB 2 uSEC MAX

THESE FIGURES AVAILABLE WITH CHIP COUNT UNDER 50 FOR DATA PATH ELEMENTS!

CONCLUSION: BIT SLICE DEVICES WILL CERTAINLY FIND THEIR WAY INTO LARGE SCALE COMPUTERS

CHIP REDUCTION

A 48 BIT FLOATING POINT CPU
WITH FAST FUNCTION ARCHITECTURES
WITH MAXIMUM SPEED, MAXIMUM PARALLEL CONTROLLER
(AS IN SECTION 2) WITH 1024 MICROINSTRUCTION ROM
WITH 48 BIT MEMORY/IO BUS

APPROXIMATELY 120 CHIPS
(WITHOUT BIT SLICE LSI: APPROXIMATELY 720 CHIPS)

SPEED

INHERENT CHIP SPEED
ARCHITECTURE ENHANCEMENTS
DISSIPATION REDUCTION

COST

NUMBER CRUNCHERS REDUCED TO MINI COMPUTER PRICES

PRESENTLY THE POPULATION OF COMPUTERS WITH FLOATING POINTS

100,000 WORLD WIDE

IS THERE ANY USE FOR A MINI PRICED F.P. MACHINE?

- NAVIGATION REDUCTION FOR SHIPS AND SMALL PLANES
- TRAINING SIMULATORS
- COMPLEX FUNCTION MONITORS - INDUSTRIAL CONTROL, AIR CONTROL
- PATTERN RECOGNITION APPLICATIONS
  - VOICE RECOGNITION
  - SCENE ANALYSIS
- DISPLAY SYSTEMS INVOLVING
  - ROTATION
  - PERSPECTIVE
  - DISAPPEARANCE
- MILITARY

BIT SLICE MACHINE CHARACTERISTICS

- FAST CYCLE
- POWERFUL MICROINSTRUCTION
- LARGE REGISTER COMPLEMENT

MAKE POSSIBLE MICRO LEVEL INTERPRETERS FOR

- APL
- ALGOL
- PL-1
- FORTRAN

EXAMPLE: 2000 CONTROL WORD VERSION OF CHAPTER 3 MACHINE COULD IMPLEMENT COMPLETE APL - SPEAKING MACHINE.

"POWER TO BURN" IMPLIES REDUCTION OF SOFTWARE DEVELOPMENT

- TIME
- COST

CRUCIAL IN LARGE SYSTEMS WHERE SOFTWARE/HARDWARE COST IS VERY HIGH

SPECIAL ARCHITECTURES - CHARACTER PROCESSORS

CHARACTER STRING PROCESSING

NARROW DATA: 8 BITS

SIMPLE FUNCTIONS

- STRING SEARCH
- STRING DELETE
- STRING MODIFY
- STRING INSERT

SPECIAL FUNCTIONS

- CYCLIC REDUNDANCY CHECK
- PARITY

OPTIMAL HARDWARE ALLOCATION

CHEAP PROCESSOR AND EXPENSIVE MEMORY

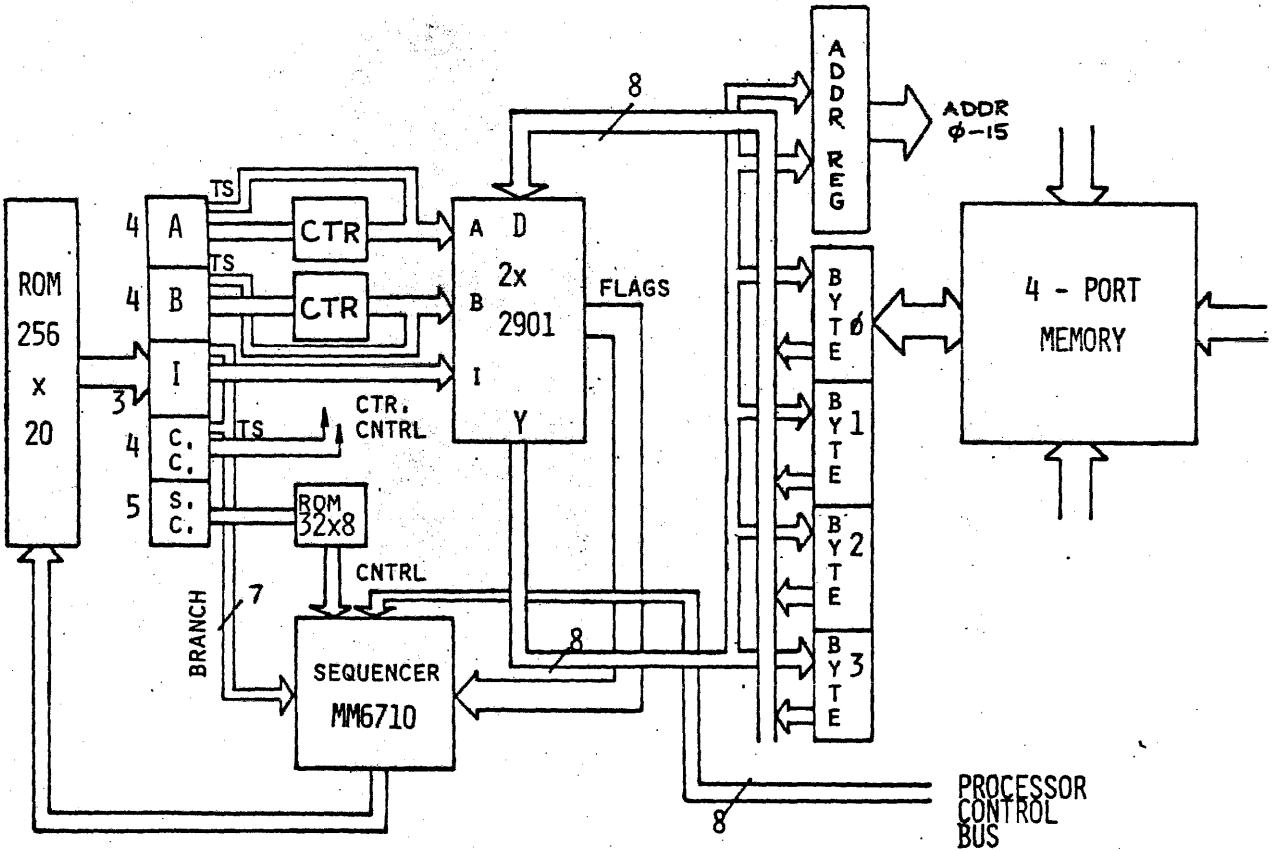
DESIRABLE CHARACTERISTICS

ASYNCHRONOUS MEMORY ACCESS
MANY REGISTERS, ITERATIVE ACCESS
AUTOMATED LOOPS - LOW OVERHEAD
MASK FUNCTION

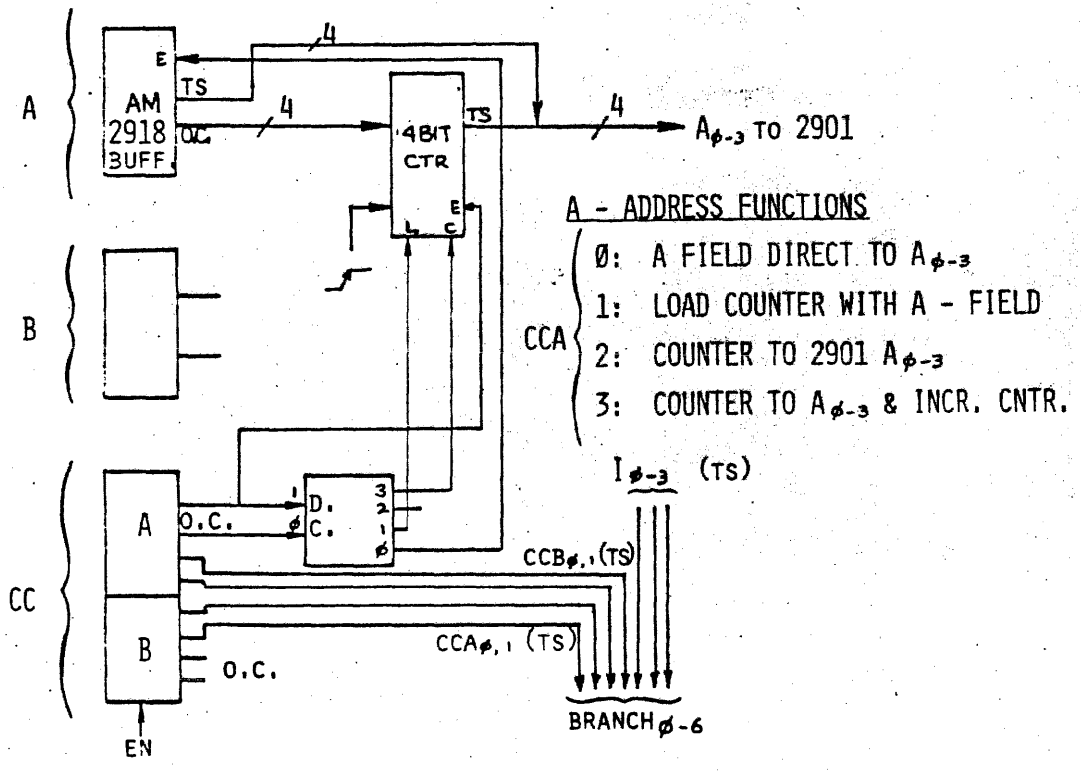
MULTI - BYTE PARALLEL MEMORY ACCESS
SUCH THAT (BYTES/FETCH) x (FETCH/SECOND) = μCYCLE/SECOND

MINIMUM μCYCLE (EFFECTIVE)

ALSO
POSSIBILITY OF MULTIPROCESSORS



PROCESSOR INCLUDING BUFFERS APPROX 26 CHIPS


A
B
CC
AM
2918
BUFF
E
TS
O.C.
4
4BIT
CTR
TS
E
L
C
$A_{\phi-3}$ TO 2901
A - ADDRESS FUNCTIONS
CCA
Ø: A FIELD DIRECT TO $A_{\phi-3}$
1: LOAD COUNTER WITH A - FIELD
2: COUNTER TO 2901 $A_{\phi-3}$
3: COUNTER TO $A_{\phi-3}$ & INCR. CNTR.
$I_{\phi-3}$ (TS)
D.
C.
3
2
1
Ø
A
O.C.
B
O.C.
EN
$CCB_{\phi,1}$ (TS)
$CCA_{\phi,1}$ (TS)
BRANCH Ø-6

PROCESSOR CYCLE = 150 NS

(NO END AROUND SHIFTS) :

ITERATIVE ONE INSTRUCTION LOOPS UP TO 16 AT 150 NS/INSTR.

E.G. COMPARE CONTENTS OF 15 REGS TO ONE BYTE

SET LOOP COUNTER = 15

LOAD A COUNTER = Ø

LOOP:

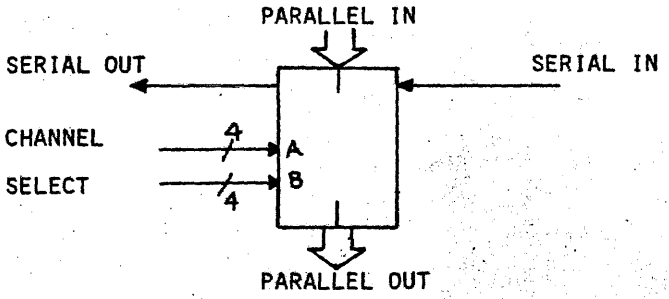
CTR A(+1)	ADR B = 15	A XOR B	BRANCH IF F=Ø

MAX 2.25 uSEC + .3 uSEC = 2.55 uSEC

SEARCH 8 CHARACTER SEGMENT FOR 1 OF 8 CHARACTERS

.3 + 8(.3 + 8(150)) = 12.3 uSEC

MULTIPLE SERIAL I/O ADAPTOR (UP TO 16 CHANNELS)



ONE REGISTER PER CHANNEL

120 NS CYCLE (NO ARITHMETIC)

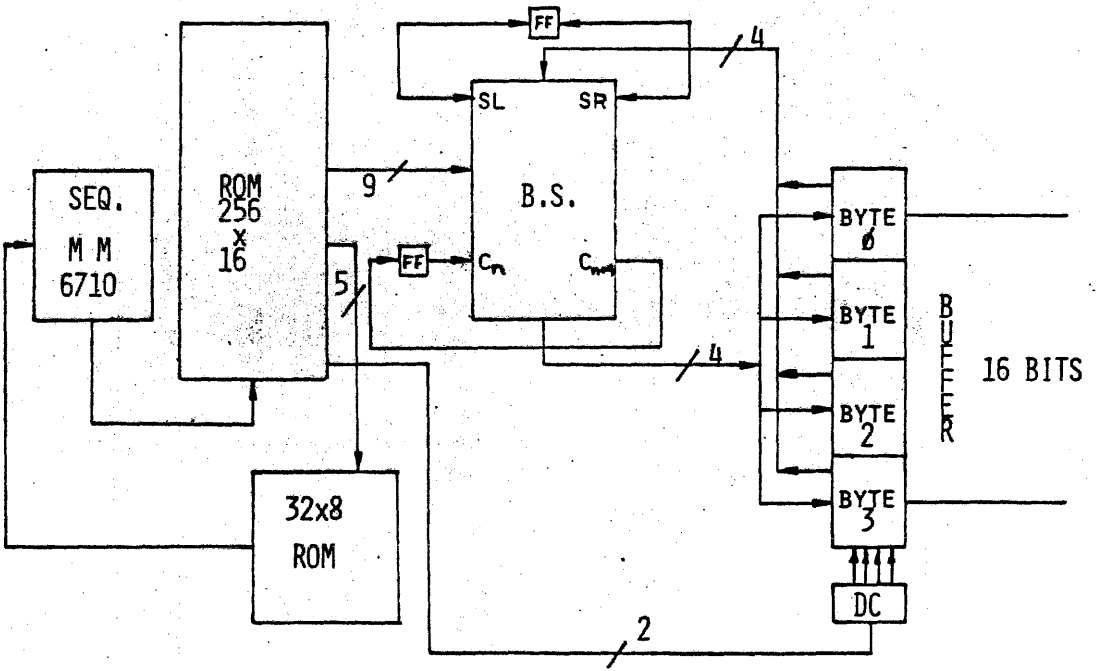
ASSUME: 1 SERIAL IN PHASE
1 PARALLEL OUT PHASE
1 PARALLEL IN PHASE
1 SERIAL OUT PHASE

PER OPERATION CYCLE = 480 NS

RATE = 2 MHz/NO. OF CHANNELS

SERIAL PROCESSOR 1 CHIP

EXAMPLE: 16 REGISTER NIBBLE (4 BIT) MACHINE



BIT SLICE EMULATIONS OF POPULAR MOS MICROPROCESSORS

BIPOLAR SPEED

PROGRAM COMPATIBILITY

I/O COMPATIBILITY

EXAMPLE

SIGNETICS 80E EMULATION OF 8080A

USES SIGNETCS (INTEL) 3000 SERIES

TWO TO TWELVE TIMES FASTER DEPENDING ON APPLICATION

BLANK

6. DEVELOPMENT AIDS

SYBEX

1. ROM SIMULATOR

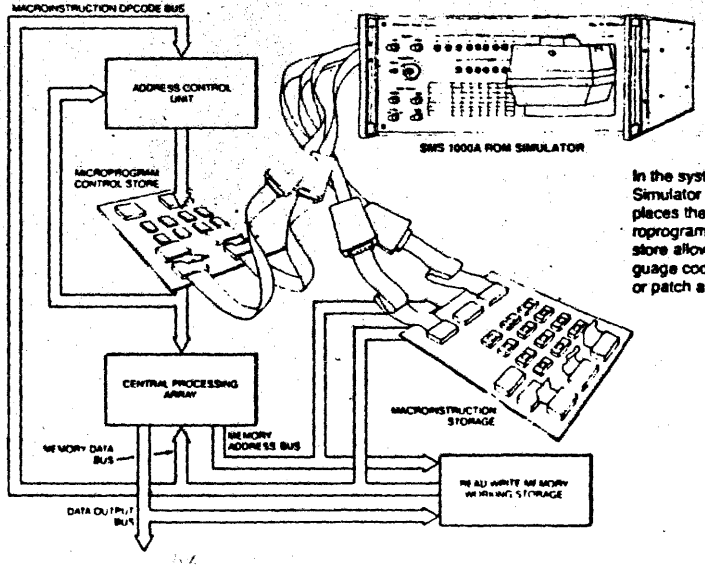
VARIABLE CONFIGURATION

PLUG IN SIMULATION OF ROMS

100 NS PERFORMANCE MAY REQUIRE SLOW CLOCK

PAPER TAPE LOADING -- TTY DUMPING

FRONT PANEL INSTRUCTION MODIFICATION AID VERIFICATON



In the system shown, the ROM Simulator simultaneously replaces the ROMS in both the microprogram and macroprogram store allowing the machine language code to be changed to find or patch a problem.

2. PROMS, EROMS, FPLAS

3. PORTABLE PROM AND PLA PROGRAMMERS

4. USER DEFINED ASSEMBLERS (E.G. RAPID)

SPEEDS DEFINITION OF MICROPROGRAM

CONVERTS SYMBOLIC PROGRAM TO ROM BIT PATTERNS

ACCEPTS USER DEFINITION OF

WORD LENGTH

FIELDS IN WORD, MULTIPLE FORMATS IF NECESSARY

VALID SYMBOLS

BINARY EQUIVALENT OF SYMBOLS

GENERATES ROM (TAPE OR CARD) IMAGE

RUNS ON IBM OR OTHER HOST SYSTEMS

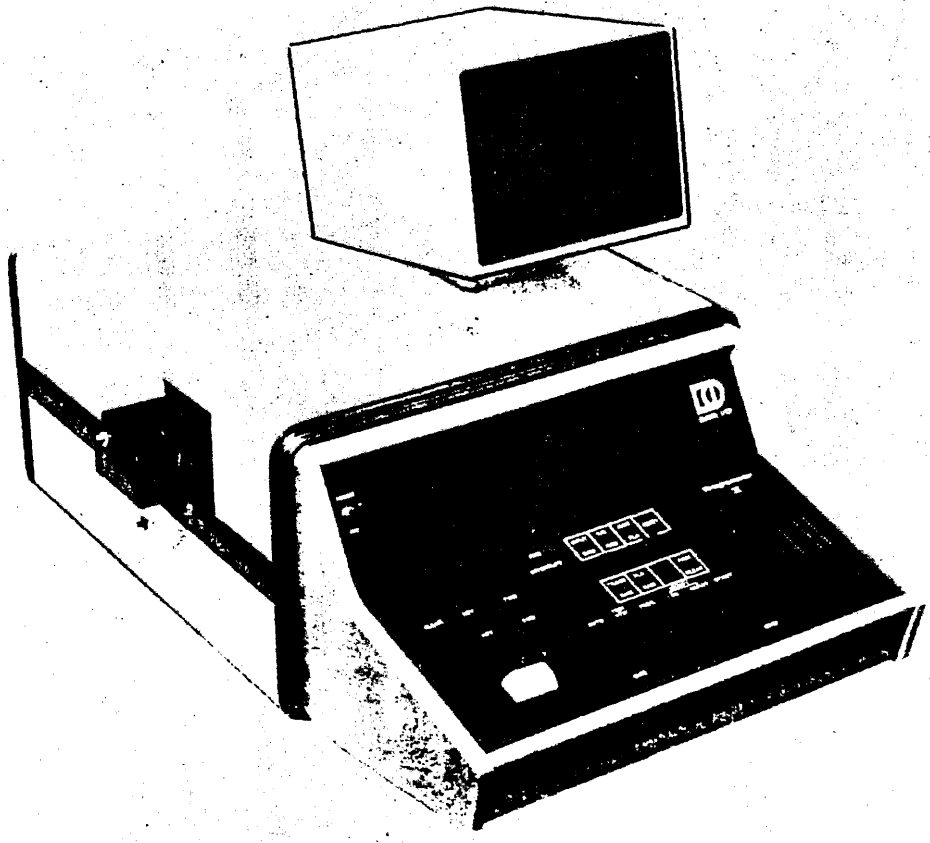
VERY USEFUL IN CONJUNCTION WITH ROM SIMULATOR

5. DEVICE SIMULATORS (E.G. ICE - 30)

FOR DEBUGGING LOGICAL DESIGN OF SYSTEM
EMULATES DEVICE (E.G. 3001 INTEL) IN CIRCUIT
ALLOWS CONTROL AND MONITORING OF SIGNALS

- FROM DATA I/O
- PROGRAMS INTERSIL AND SIGNETICS
- INPUT:
    - KEYBOARD
    - MARK SENSE CARDS
    - ASC II TAPE

DATA I/O MODEL X ALLOWS UP TO 63 PRODUCT TERMS



7. CONCLUSION

SYBEX

BIT SLICES ARE A POWERFUL BUT FLEXIBLE UNIT OF INTEGRATION.

- WIDE RANGE OF APPLICATION COMPLEXITY
- WIDE RANGE OF PERFORMANCE
- SIGNIFICANT REDUCTIONS IN COST, CHIP COUNT AND DISSIPATION MADE POSSIBLE FOR GIVEN ARCHITECTURE
- SIGNIFICANT ENHANCEMENT IN COMPUTATIONAL POWER, SPEED, RELIABILITY FOR GIVEN COST / SPACE / COOLING

BIT - SLICE DEVICES WILL BECOME A STANDARD DESIGN ELEMENT.

1. ARCHITECTURES OF DEVICES ARE CONVERGING
2. SECOND SOURCES ARE PROLIFERATING
3. USERS ARE PROLIFERATING
4. CUSTOMER DEMANDS FOR > MICROCOMPUTER HARDWARE ARE PROLIFERATING
5. BIT - SLICE DEVICES MERGE VERY WELL WITH OTHER LSI (PLA'S, ROM'S, RAM'S, I / O DRIVERS, INTERRUPT HANDLERS, ETC.)

SUMMARY

THIS COURSE HAS EXAMINED THE BIT - SLICE TECHNOLOGY IN RELATION TO :

- THE EVOLUTION OF CPU DESIGN.
- LSI CRITERIA.
- THE DESIGN PROCESS OF A MODERN CPU.
- OTHER APPLICATIONS FROM THE MICRO LEVEL TO THE NUMBER CRUNCHER LEVEL.
- ITS INTEGRATION WITH OTHER LSI DEVICES

AND IT HAS PRESENTED A SURVEY OF THE EXISTING BIT - SLICE DEVICES AND SOME OF THE PRODUCTS IN WHICH THEY ARE USED.

8. APPENDICES

SYBEX

FAMILY CHARACTERISTICS	SSI GATES Propagation Delay	FLIP-FLOPS Toggle Rate	MSI ALU 4-Bit Add Time
STANDARD TTL 54/7400 Series SSI and MSI—112 types 8200 Series MSI—60 types 8T Series Interface—36 types Standard "gold doped" TTL is the industry's longest selling digital logic family still in high volume production. New system designs generally favor the Low Power Schottky TTL equivalent functions.	10ns at 10mW	25 MHz	28 ns
LOW POWER SCHOTTKY TTL 54/74LS00 Series SSI and MSI—79 types 3000 Series Microprocessor Set Low power Schottky provides the same speed as standard TTL at 1/5 the power. The power savings and LSI potential are encouraging the use of 74LS in most new system designs.	10ns at 2mW	30 MHz	21 ns
HIGH SPEED TTL 54/74H00 Series SSI—30 types 8200 Series MSI—18 types Higher speed versions of standard TTL SSI devices. Generally being replaced by Schottky TTL in new designs.	6ns at 22mW	45 MHz	no MSI
SCHOTTKY TTL 54/74S00 Series SSI and MSI—55 types Schottky TTL uses a diode clamp design to insure the highest speed possible at TTL logic levels.	3ns at 30mW	90 MHz	11ns
ECL 10,00 Series SSI and MSI—69 types ECL devices use a narrow logic swing to provide the highest speed standard logic family for use in large mainframe computers and test equipment.			

TTL EQUIVALENT

The 5701/6701 is similar in function to the 25 TTL MSI packages listed below. It saves 375 I/O pins, 5.6 watts and 30 square inches of board area.

TABLE 1

Function	TTL#	#14 Pin or #16 Pin Pkgs.	#24 Pin Pkgs.	Advertised Gate Complexity (Each Pkg.)	Gate Complexity Total	Typical Power Each (Watts)	Total Power (Watts)
32 x 9 & 8 x 8 ROMs	7488	3		70	210	.50	1.50
16 x 4 Multiport RAM	74172		4	110*	440	.56	2.24
Arithmetic Logic Unit	74181		1	75	75	.55	.55
Storage Latches	7475	2		28	56	.16	.32
J-K Flip Flop (Q Reg)	74107	2		22	44	.10	.20
4 to 1 MUX	74153	6		16	96	.20	1.20
O/I True Complement	74H87	2		18	36	.27	.54
Dual 4 Bit Select	74157	2		15	30	.15	.30
Quad 2 to 1 MUX with 3 State Outputs	74S257	2		15	30	.30	.60
3 State Buffer	DM8094	1		5	5	.18	.18
Totals		20	5		1022		6.63

*The 74172 is advertised at 201 gate complexity but we are using only 2 of the 3 address capability, hence we have counted it as 110 gates.

APPENDIX C: CARRY LOOK-AHEAD LOGIC

ALU FUNCTIONS (AM 2901)

LOGIC FUNCTIONS FOR G, P, C_{n+4}, AND OVR

The four signals G, P, C_{n+4}, and OVR are designed to indicate carry and overflow conditions when the Am2901 is in the add or subtract mode. The table below indicates the logic equations for these four signals for each of the eight ALU functions. The R and S inputs are the two inputs selected according to Figure 2.

Definitions (+ = OR)

$P_0 = R_0 + S_0$ $\quad G_0 = R_0 S_0$

$P_1 = R_1 + S_1$ $\quad G_1 = R_1 S_1$

$P_2 = R_2 + S_2$ $\quad G_2 = R_2 S_2$

$P_3 = R_3 + S_3$ $\quad G_3 = R_3 S_3$

$$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_n$$

$$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_n$$

I_{543}	Function	$\overline{P}$	$\overline{G}$	C_{n+4}	OVR
0	R + S	$\overline{P_3P_2P_1P_0}$	$\overline{G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0}$	C_4	$C_3 \forall C_4$
1	S − R	← Same as R + S equations, but substitute $\overline{R_i}$ for R_i in definitions →			
2	R − S	← Same as R + S equations, but substitute $\overline{S_i}$ for S_i in definitions →			
3	R ∨ S	LOW	$P_3P_2P_1P_0$	$\overline{P_3P_2P_1P_0} + C_n$	$\overline{P_3P_2P_1P_0} + C_n$
4	R ∧ S	LOW	$\overline{G_3 + G_2 + G_1 + G_0}$	$G_3 + G_2 + G_1 + G_0 + C_n$	$\overline{G_3 + G_2 + G_1 + G_0} + \overline{C_n}$
5	$\overline{R}$ ∧ S	LOW	← Same as R ∧ S equations, but substitute $\overline{R_i}$ for R_i in definitions →		
6	R ∀ S	← Same as $\overline{R \forall S}$, but substitute $\overline{R_i}$ for R_i in definitions →			
7	$\overline{R \forall S}$	$G_3 + G_2 + G_1 + G_0$	$P_3G_3 + P_3P_2G_2 + P_3P_2P_1G_1 + P_3P_2P_1P_0$	$P_3G_3 + P_3P_2G_2 + P_3P_2P_1G_1 + P_3P_2P_1P_0 (G_0 + \overline{C_n})$	Complement of C_{n+4} at left

+ = OR

DEVICE	DESCRIPTION				
CAMS					
8220	8-bit CAM	4 x 2	OC	40	C
10155	16-bit CAM	8 x 2	OE	13	M,C
SAMS					
82S12	32-bit SAM	8 x 4	OC	35	C
82S112	32-bit SAM	8 x 4	TS	35	C
RAMS					
82S21	64-bit RAM	32 x 2	OC	50	C
82S25	64-bit RAM	16 x 4	OC	50	M,C
54/74S89	64-bit RAM	16 x 4	OC	50	M,C
54/74S189	64-bit RAM	16 x 4	TS	35	M,C
3101A	64-bit RAM	16 x 4	OC	35	M,C
82S16	256-bit RAM	256 x 1	TS	50	M,C
82S17	256-bit RAM	256 x 1	OC	50	M,C
82S116	256-bit RAM	256 x 1	TS	40	C
82S117	256-bit RAM	256 x 1	OC	40	C
54/74S200	256-bit RAM	256 x 1	TS	50	M,C
54/74S201	256-bit RAM	256 x 1	TS	50	M,C
54/74S301	256-bit RAM	256 x 1	OC	50	M,C
10144	256-bit RAM	256 x 1	OE	30	M,C
82S10	1024-bit RAM	1024 x 1	OC	45	M,C
82S11	1024-bit RAM	1024 x 1	TS	45	M,C
93415A	1024-bit RAM	1024 x 1	OC	45	M,C
93425A	1024-bit RAM	1024 x 1	TS	45	M,C
ROMS					
82S226	1024-bit ROM	256 x 4	OC	50	M,C
82S229	1024-bit ROM	256 x 4	TS	50	M,C
82S230	2048-bit ROM	512 x 4	OC	50	M,C
82S231	2048-bit ROM	512 x 4	TS	50	M,C
82S214	2048-bit ROM	256 x 8	TS	60	M,C
8228	4096-bit ROM	1024 x 4	TTL	75	C
82S215	4096-bit ROM	512 x 8	TS	60	M,C
PROMS					
82S23	256-bit PROM	32 x 8	OC	50	M,C
82S123	256-bit PROM	32 x 8	TS	50	M,C
10139	256-bit PROM	32 x 8	OE	20	M,C
82S27	1024-bit PROM	256 x 4	OC	40	C
82S126	1024-bit PROM	256 x 4	OC	50	M,C
82S129	1024-bit PROM	256 x 4	TS	50	M,C
10149	1024-bit PROM	256 x 4	OE	17	M,C
82S114	2048-bit PROM	256 x 8	TS	60	M,C
82S130	2048-bit PROM	512 x 4	OC	50	M,C
82S131	2048-bit PROM	512 x 4	TS	50	M,C
82S115	4096-bit PROM	512 x 8	TS	60	M,C
825136	4096-bit PROM	1024 x 4	OC	60	M,C
82S137	4096-bit PROM	1024 x 4	TS	60	M,C
FPLA					
82S100	FPLA	16 x 48 x 8	TS	50	M,C
82S101	FPLA	16 x 48 x 8	OC	50	M,C

* TEMPERATURE RANGE
  C = Commercial (0°C to +75°C)
  M = Military (−55°C to +125°C)
  All ECL 10,000 (−30°C to +85°C)

** OUTPUT
  TS = Tri-State
  OC = Open Collector
  OE = Open Emitter

Signetics offers a broad line of MOS memories and memory-related products with many different access modes, architectures and speeds. Sizes range from 100 bits to 8192 bits on a single chip.

The memory products listed on the following pages can be categorized by both technology and product type. Basically, two technologies are used in the manufacture of Signetics MOS memory products. These are:

P-Channel, Silicon Gate
N-Channel, Ion-Implanted Silicon Gate

The P-Channel process is used in Signetics' 2500 series and the N-Channel is used in both the 2100 and 2600 series.

The 2500 series consists of shift registers—both static and dynamic, character generators and custom programmable ROMS, 256-bit static RAMs and 1103 type dynamic RAM.

The 2100 series is a family of 1024-bit static RAMs. This family consists of standard devices (2102), low power devices (21L02), high speed types (21F02), and military products (M2102). This series give designer the choice of access times from 250r 1.0us and power supply currents from 40m 70MA.

The 2600 series consists of a 256 x 4 static F the 4096 x 1 dynamic RAMs, and a 1024 x 8 m programmable ROM. All N-Channel devices ope with a single 5V power supply and are compatible.

Most MOS memory products are available ur Signetics Upgraded Product Reliability (SU Program and both products and processes are c tinually monitored under Signetics' SURE Reliab Program. Both programs are described elsewh in this catalog.

A variety of packaging options are available most MOS memory products. They are:

A = 14-pin Silicon DIP
B = 16-pin Silicon DIP
F = 16-pin CERDIP
I = Ceramic DIP
K = 10-pin TO-100
N = 24-pin Silicon D
TA = 8-pin TO-99
V = 8-pin Silicon DIF
XA = 18-pin Silicon DI

		ORGANIZATION	SIGNETICS PART NUMBER	ACCESS TIME (ns)	POWER SUPPLY	PACKAGE
MEMORIES	STATIC RAM	256 x 1	1101/2501	1000	+5, −9, −9	B, I
		256 x 1	25L01	1000	+5, −12	B, I
		256 x 4	2606	750	+5, 0	B, F
		256 x 4	2602-1	500	+5, 0	B, F, I
		1024 x 1	2102/2602	1000	+5, 0	B, F, I
		1024 x 1	2102-1/2602-1	500	+5, 0	B, F, I
		1024 x 1	2102-2/2602-2	650	+5, 0	B, F, I
		1024 x 1	21F02	350	+5, 0	B, F, I
		1024 x 1	21F02-4	450	+5, 0	B, F, I
		1024 x 1	21L02	1000	+5, 0	B, F, I
		1024 x 1	21L02-1	500	+5, 0	B, F, I
		1024 x 1	21L02-2	650	+5, 0	B, F, I
		1024 x 1	21L02-3	400	+5, 0	B, F, I
		1024 x 1	M2102-4	450	+5, 0	F, I
		1024 x 1	M2102-6	650	+5, 0	F, I
	DYNAMIC RAM	1024 x 1	1103	310	+16, 0, +19	XA, I
		1024 x 1	1103-1	180	+19, 0, +21	XA, I
		4096 x 1	2680	200-300	+12, +5, 0, −5	I, F
		4096 x 1	2660	200-350	+12, +5, 0, −5	I, F
	STATIC ROM	512 x 8	2530	700	+5, 0, −12	N, I
		1024 x 8	2608	650	+5, 0	I
		2048 x 4	2580	950	+5, 0, −12	N, I
CHARACTER GENERATORS		64 x 8 x 5	2513	600	+5, 0, −12	N, I
		64 x 6 x 8	2516	600	+5, −12	N, I
		64 x 9 x 9	2526	700	+5, 0, −12	N, I

MOS MEMORIES

	ORGANIZATION/ REGISTER LENGTH	SIGNETICS PART NUMBER	SPEED (MHz)	POWER SUPPLY	PACKAGE
STATIC	32 x 6	2518	2.0	)⅜, [⅛¼	B, I
	40 x 6	2519	2.0	+5, −12	B, I
	50 x 2	2509	1.5	+5, , −5, −12	A, K
	80 x 4	2532	1.5	+5, 0, −12	B, I
	100 x 2	2510	1.5	+5, −5, −12	A, K
	128 x 2	2521	1.5	+5, −12	V
	132 x 2	2522	1.5	+5, −12	V
	200 x 2	2511	1.5	+5, −5, −12	A, K
	240 x 2	2529	1.5	+5, −12	V
	250 x 2	2528	1.5	+5, −12	V
	256 x 2	2527	1.5	+5, −12	V
	1024 x 1	2533	1.5	+5, 0, −12	V
DYNAMIC	100 x 2	2506/7/17	3.0	+5, −5	T, V
	256 x 4	2502/1402	8.0	+5, −5	B, I
	512 x 1	2505	2.5	+5, −5	K
	512 x 1	2524	3.0	+5, −5	V
	512 x 2	2503/1403	8.0	+5, −5	TA, V
	1024 x 1	2504/1404	8.0	+5, −5	TA, V
	1024 x 1	2525	3.0	+5, −5	V
	1024 x 1	2512	2.5	+5, −5	K

MOS MEMORIES CROSS REFERENCE

AMD	SIGNETICS
AM1402A/2802	1402B
AM1402APC	2502B
AM1403A/2803	1403TA
AM1403A	2503TA
AM1404A/2804	1404V
AM1404A	2504TA
AM1405A/2805	1405K
AM1506T	1506V
AM1507T	2506T/2517T
AM2505K	2505K
AM2806HC	2512K
AM2807PC	2524V
AM2808PC	2525V
AM2809PC	2521V
AM2833PC	2533V

FAIRCHILD	SIGNETICS
3343	2521V
3344	2522V
3347	2532B
3349	2518B
3533	2533V

G.I.	SIGNETICS
2509	2509K
2510	2510A
2511	2511A
2533	2533V

INTERSIL	SIGNETICS
IM7712C	2512K
IM7722C	2525V
IM7780C	3347

INTEL	SIGNETICS
C1402A	1402B/2502B
C1403A	1403TA/2503TA
M1404A	2504TA
M1405A	1405K/2505K

MOSTEK	SIGNETICS
MK1007P	2532B

M.I.I.	SIGNETICS
MF1402A	2502B
MF1403A	2503TA
MF1405A	2504TA
MF1406	2506T
MF1407	2517T

NATIONAL	SIGNETICS
MM506H	1506V/2506T
MM507H	1507T/2517T
MM1402A	1402B/2502B
MM1403A	1403TA/2503TA
MM1404A	1404V/2504TA
MM2521	2521V
MM2522	2522V
MM5058	2533V

T.I.	SIGNETICS
TMS3112NC	2518B
TMS3120NC	2532B
TMS3128NC	2521V
TMS3129NC	2522V
TMS3133NC	2533V

MOS MEMORIES CROSS REFERENCE

MOS N-CHANNEL RAMS

MD	SIGNETICS
2102	2102F
2102-1	2102-1F
2102-2	2102-2F
2102	2102B/2602B
2102-1	2102-1B/2602-1B
2102-2	2102-2B/2602-2B
M9102DC	21F02-4F
M9102PC	21F02-48
M91L02ADC	21L02-1F
M91L02ADM	M2102-41
M91L02APC	21L02-1B
M91L02BDC	21L02-3F
M91L02BPC	21L02-3B
M91L02DC	21L02-2F
M92L02DM	M2102-61
M91L02PC	21L02-2B

IRCHILD	SIGNETICS
2FDC	21F02F
2FPC	21F02B
2DC	2102-2F
2PC	2102-2B
2-1DC	21F02-4F
2-1PC	21F02-48

ERSIL	SIGNETICS
52C	2102B/2602B
52-1C	2102-1B/2602-1B
52-2C	2102-2B/2602-2B

MEMORIES

INTEL	SIGNETICS
C2012	2102F
C2102-1	2102-1F
C2102-2	2102-2F
C2102A	21F02F
C2102A-2	21F02-2F
C2102A-4	21F02-4F
C2107A	26041
MC2102A-4	M2102-41
MC2102A-6	M2102-61
P2102	2102B/2602B
P2102-1	2102-1B/2602-1B
P2102-2	2102-2B/2602-2B
P2102A	21F02B
P2102A-2	21F02-2B
P2102A-4	21F02-48

MOSTEK	SIGNETICS
MK4102	2102B/2602B
MK4102-1	2102-1B/2602-1B

M.I.L.	SIGNETICS
MF2102	2102B/2602B
MF2102-1	2102-1B/2602-1B
MF2102-2	2102-2B/2602-2B

NATIONAL	SIGNETICS
MM2102D	2102B/2602B
MM2102-1	2102-1B/2602-1B
MM2102-2	2102-2B/2602-2B

T.I.	SIGNETICS
TMS4033	2602-2B
TMS4033JL	21F02-4F
TMS4033NL	21F02-48
TMS4034	2602-1B
TMS4034JL	2102-2F
TMS4034NL	2102-2B
TMS4035	2602B
TMS4035JL	2102F
TMS4035NL	2102B

MOS P-CHANNEL RAMS

AMD	SIGNETICS
AM1101A1	1101B
P1101A	2501B

INTERSIL	SIGNETICS
IM7501/11/12C	1101B/2501B

INTEL	SIGNETICS
P1101A	2501B
P1101A1	1101B
P1103	1103XA

MOSTEK	SIGNETICS
MK4007	2501B/25L01B
MK4007P	1101B

M. I. L.	SIGNETICS
MF1101A	1101B/2501B
MF1103	1103XA

NATIONAL	SIGNETICS
MM1101A	2501B
MM1101A1	1101B

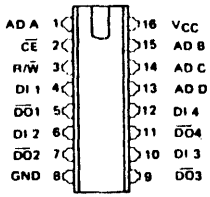
MOS RAMS

G.I.	SIGNETICS
2580	2580N

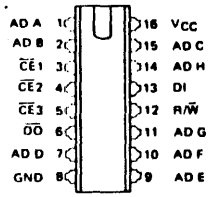
SERIES 54S/74S
RANDOM-ACCESS READ/WRITE MEMORIES

BULLETIN NO. DL-S 7512257, MAY 1975

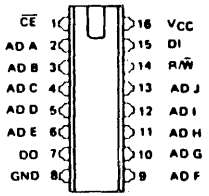
64 BITS (16 WORDS BY 4 BITS)
'S189, 'S289



256 BITS (256 WORDS BY 1 BIT)
'S201, 'S301



1024 BITS (1024 WORDS BY 1 BIT)
SN74S209, SN74S309



Pin assignments for all of these memories are the same for all packages.

- Schottky-Clamped for High Performance
- Full On-Chip Decoding and Fast Chip-Enable Simplify System Decoding
- P-N-P Inputs Reduce Loading on System Buffers/Drivers
- Choice of 3-State or Open-Collector Outputs

TYPE NUMBER (PACKAGES)		TYPE OF	BIT SIZE	TYPICAL ACCESS TIMES		WRITE CYCLE TIME	
−55°C to 125°C	0°C to 70°C	OUTPUT(S)	(ORGANIZATION)	CHIP SELECT	ADDRESS	SN54S'	SN74S'
SN54S189(J, W)	SN74S189(J, N)	3-State	64 Bits (16 W x 4 B)	12 ns	25 ns	25 ns	25 ns
SN54S289(J, W)	SN74S289(J, N)	Open-Collector					
SN54S201(J, W)	SN74S201(J, N)	3-State	256 Bits (256 W x 1 B)	13 ns	42 ns	100 ns	65 ns
SN54S301(J, W)	SN74S301(J, N)	Open-Collector					
	SN74S209(J, N)	3-State	1024 Bits (1024 W x 1 B)	20 ns	70 ns		150 ns
	SN74S309(J, N)	Open-Collector					

description

These monolithic TTL memories feature Schottky clamping for high performance, a fast chip select access time to enhance decoding at the system level, and the 'S201 and 'S209 RAMs utilize inverted-cell memory elements to achieve high densities. The memories feature p-n-p input transistors that reduce the low level input current requirement to a maximum of −0.25 milliamperes, only one-eighth that of a Series 54S/74S standard load factor.

A three-state-output version and an open-collector-output version are offered for each of the three organizations. A three-state output offers the convenience of an open-collector output with the speed of a totem-pole output; it can be bus-connected to other similar outputs, yet it retains the fast rise time characteristic of the TTL totem-pole output. An open collector output offers the capability of direct interface with a data line having a passive pull-up.

write cycle

Information to be stored in the memory is written into the selected address (AD) location when the chip enable ($\overline{CE}$) and the read/write (R/$\overline{W}$) inputs are low. While the read/write input is low, the memory output(s) is(are) off (three-state = Hi-Z, open-collector = high). When a number of outputs are bus-connected, this off state neither loads nor drives the data bus; however, it permits the bus line to be driven by other active outputs or a passive pull-up.

read cycle

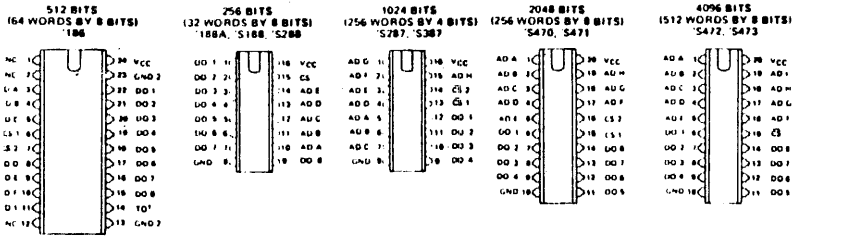
Information stored in the memory (see function table for input/output phase relationship) is available at the output(s) when the read/write input is high and the chip-enable input(s) is(are) low. When one(or more) chip-enable input is(are) high, the output(s) will be off.

SERIES 54/74, 54S/74S
PROGRAMMABLE READ-ONLY MEMORIES

BULLETIN NO. DL-S 7512258, MAY 1975

- Titanium-Tungsten (Ti-W) Fuse Links for Fast, Low-Voltage, Reliable Programming
- All Schottky-Clamped PROM's Offer:
  - Fast Chip Select to Simplify System Decode
  - Choice of Three-State or Open-Collector Outputs
  - P-N-P Inputs for Reduced Loading on System Buffers/Drivers
- Full Decoding and Chip Select Simplify System Design
- Applications Include:
  - Microprogramming/Firmware Loaders
  - Code Converters/Character Generators
  - Translators/Emulators
  - Address Mapping/Look-Up Tables

TYPE NUMBER (PACKAGES)		BIT SIZE	OUTPUT	TYPICAL ACCESS TIME (ns)	
−55°C to 125°C	0°C to 70°C	(ORGANIZATION)	CONFIGURATION	FROM ADDRESS	FROM CHIP SELECT
SN54186(J, W)	SN74186(J, N)	512 bits (64 W × 8 B)	open-collector	60	55
SN54188A(J, W)	SN74188A(J, N)	256 bits (32 W × 8 B)	open-collector	30	34
SN54S188(J, W)	SN74S188(J, N)		open-collector	25	12
SN54S288(J, W)	SN74S288(J, N)		three-state	25	12
SN54S287(J, W)	SN74S287(J, N)	1024 bits (256 W × 4 B)	three-state	42	15
SN54S387(J, W)	SN74S387(J, N)		open-collector	42	15
SN54S470(J)	SN74S470(J, N)	2048 bits (256 W × 8 B)	open-collector	50	20
SN54S471(J)	SN74S471(J, N)		three-state	50	20
SN54S472(J)	SN74S472(J, N)	4096 bits (512 W × 8 B)	three-state	55	20
SN54S473(J)	SN74S473(J, N)		open-collector	55	20



Pin assignments for all of these memories are the same for all packages.

Description

These monolithic TTL programmable read-only memories (PROM's) feature titanium-tungsten (Ti-W) fuse links with each link designed to program in one millisecond or less. The Schottky-clamped versions of these PROM's offer considerable flexibility for upgrading existing designs or improving new designs as they feature full Schottky clamping for improved performance, low-current MOS-compatible p-n-p inputs, choice of bus-driving three-state or open-collector outputs, and improved chip-select access times.

The high-complexity 2048- and 4096-bit PROM's can be used to significantly improve system density for fixed memories as all are offered in the 20-pin dual-in-line package having pin-row spacings of 0.300 inch.

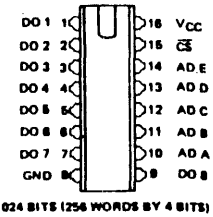
SERIES 54/74, 54S/74S
READ-ONLY MEMORIES

BULLETIN NO. DL-S 7612769, MAY 1976

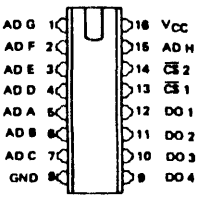
- Mask-Programmed Memories That Can Replace PROMs
- Full On-Chip Decoding and Fast Chip Select(s) Simplify System Decoding
- All Schottky-Clamped ROMs Offer
  - —Choice of 3-State or Open-Collector Outputs
  - —P-N-P Inputs for Reduced Loading on System Buffers/Drivers
- Applications Include:
  - —Microprogramming Firmware/Firmware Loaders
  - —Code Converters/Character Generators
  - —Translators/Emulators
  - —Address Mapping/Look-Up Tables

TYPE NUMBER (PACKAGES)		TYPE OF OUTPUT(S)	BIT SIZE (ORGANIZATION)	TYPICAL ACCESS TIMES	
−55°C to 125°C	0°C to 70°C			CHIP SELECT	ADDRESS
SN5488A(J, W)	SN7488A(J, N)	Open-Collector	256 Bits (32 W × 8 B)	22 ns	26 ns
SN54187(J, W)	SN74187(J, N)	Open-Collector	1024 Bits (256 W × 4 B)	20 ns	40 ns
SN54S270(J)	SN74S270(J, N)	Open-Collector	2048 Bits (512 W × 4 B)	15 ns	45 ns
SN54S370(J)	SN74S370(J, N)	3-State			
SN54S271(J)	SN74S271(J, N)	Open-Collector	2048 Bits (256 W × 8 B)	15 ns	45 ns
SN54S371(J)	SN74S371(J, N)	3-State			

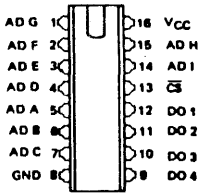
256 BITS (32 WORDS BY 8 BITS)
'88A



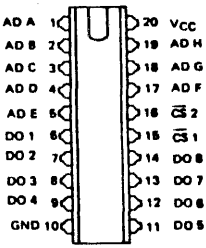
1024 BITS (256 WORDS BY 4 BITS)
'187



2048 BITS (512 WORDS BY 4 BITS)
'S270, 'S370



2048 BITS (256 WORDS BY 8 BITS)
'S271, 'S371



Pin assignments for all of these memories are the same for all packages

description

These monolithic TTL custom-programmed read-only memories (ROMs) are particularly attractive for applications requiring medium to large quantities of the same bit pattern. Plug-in replacements can be obtained for most of the popular TTL PROMs.

The high-complexity 2048-bit ROMs can be used to significantly improve system bit density for fixed memory as all are offered in compact 16- or 20-pin dual-in-line packages having pin-row spacings of 0.300-inch.

The Schottky-clamped versions offer considerable flexibility for upgrading existing designs or improving new designs as they feature improved performance; plus, they offer low-current MOS-compatible p-n-p inputs, choice of bus-driving three-state or open-collector outputs, and improved chip-select access times.

Data from a sequenced deck of data cards punched according to the specified format are permanently programmed by the factory into the monolithic structure for all bit locations. Upon receipt of the order, Texas Instruments will assign a special identifying number for each pattern programmed according to the order. The completed devices will be marked with the appropriate TI special device number. It is important that the customer specify not only the output levels desired at all bit locations, but also the other information requested under ordering instructions.

The three-state outputs offer the convenience of an open-collector output with the speed of a totem-pole output: they can be bus-connected to other similar outputs yet they retain the fast rise time characteristic of the TTL totem-pole output. The open-collector outputs offer the capability of direct interface with a data line having a passive pull-up.

Word-addressing is accomplished in straight positive-logic binary and the memory may be read when all $\overline{CS}$ inputs are low. A high at any $\overline{CS}$ input causes the outputs to be off.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

APPENDIX E: INTEGRATED INJECTION LOGIC

I^2L BIPOLAR LOGIC

DESCRIPTION OF GATE

IMPLEMENTATION OF LOGIC

FABRICATION

INTERFACING

FROM TEXAS INSTRUMENTS

INTEGRATED INJECTION LOGIC, I^2L

I^2L is a highly efficient new bipolar technology which reduces a basic gate function to a single current injected transistor switch. The logical simplicity of a single geometry gate, requiring no isolation, no load resistors and no ground metallization, achieves gate component densities 10 times those of conventional TTL or CMOS. I^2L gates can be operated along a virtually constant speed/power product value over better than 5 magnitudes of injector current-from picoamps to microamps - at speeds ranging from hundreds of microseconds to tens of nanoseconds (Figure 21). They can be powered up for maximum speed then powered down without loss of function or data.

In addition, I^2L gates are static, requiring no multiphase clocks, and are capable of stable operation in severe temperature environments.

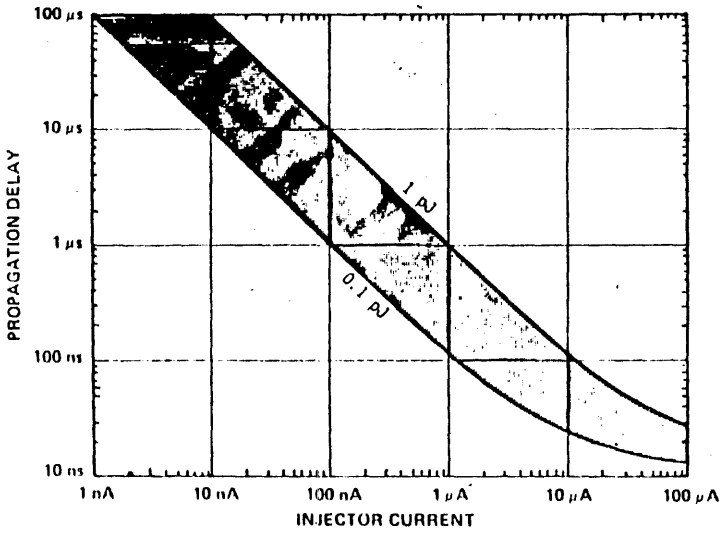


FIGURE 21. I^2L GATE PERFORMANCE RANGE

I^2L GATE

basic I^2L gate is a NPN grounded-emitter transistor switch as shown in Figure 22. itive NAND logic is implemented, and logical isolation is accomplished by the use multiple collector outputs (C1 and C2). When the base (B) is open or high and an iector (Ep) current source (I) is applied, the I^2L NPN transistor is normally ised "on" (low-level output) by the current (I) supplied by a PNP current-injector insistor. Switching action is accomplished by the steering of this injector rent. As shown on the inverter/buffer schematic, low input voltage to the base of less than one V_{BE} (750mV) starts pulling injector current out of the input ough the "on" (low) output of the driving gate. When the driving gate reaches a sat, the input is robbed of its base drive, and the driven I^2L transistor/gate l turn off causing its open-collector output to assume a high logic level. As h any open-collector logic, this output voltage level is determined by the load cuit or "pull-up" utilized. Internally, for a typical I^2L circuit design, this simply the clamp level at the input of the next stage, one V_{BE} (750mV above und). A high input logic level is achieved by default whenever a low-impedance h of less than one V_{BE} potential is absent from the input. Deprived of a ground h of less than one V_{BE} potential, the injector current will forward bias the I^2L nsistor/gate "on" and produce an output low logic level one V_{CEsat} above ground, ically 50mV. A typical I^2L internal logic swing of 700mV, from a V_{CEsat} of mV to a V_{BE} of +750mV, is thereby achieved by current steering of a NPN switch.

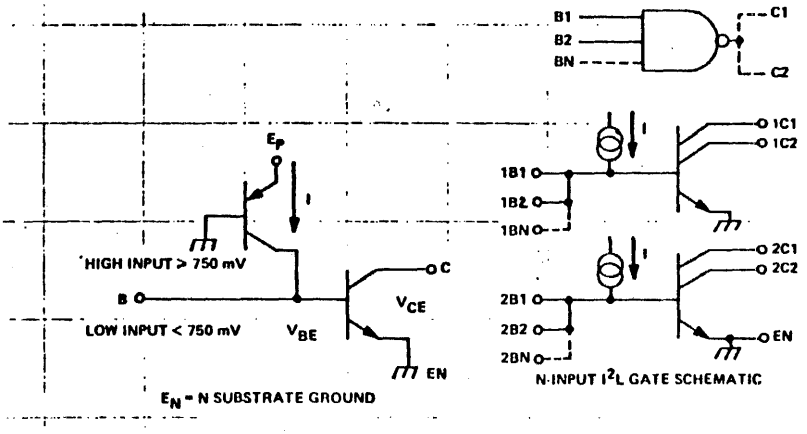


FIGURE 22. BASIC I^2L GATE

e that the base region of the NPN transistor serves as an N-wide input. number of steering inputs can be connected to the base. As all of these uts are common to the base, and each is driven by a separate output source, ical ambiguity of the drivers can be avoided only if each driver exists as individual source. Hence, the mechanism used is individual collectors for h load. Of the two collectors illustrated, C1 may be connected as the base ut of another gate. Simultaneously, this same base may be driven by the lector of another NPN gate.

IMPLEMENTING I^2L LOGIC

Figure 23 displays the manner in which I^2L transistors/gates are interconnected to perform logic. The NAND gate logic diagrammed is that of a common D-type flip flop. The schematic directly below it is the same D-type flip flop in I^2L logic at a component count of one transistor/gate.

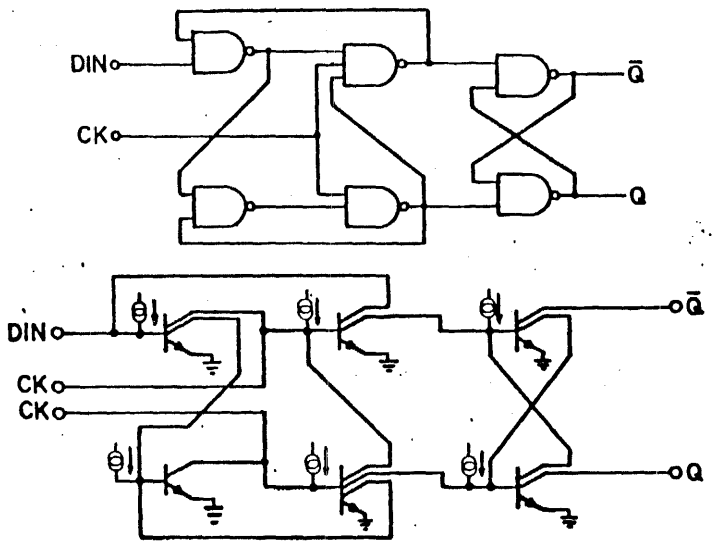


FIGURE 23. I^2L D-TYPE FLIP FLOP LOGIC

The logic diagram indicates that each gate input is a discrete entity; whereas, the I^2L schematic shows that each base input has more than one driving source. The requirement for multiple isolated I^2L collectors becomes quite evident as they identify the active and inactive driving sources; or, in other words, they isolate each discrete logical decision.

The common clock input, to the two logic gates, drives two isolated inputs. Separate clock inputs, shown in the I^2L schematic, would be driven from isolated (individual) collectors of the same npn transistor.

ATION

s-section of two I^2L gates is shown in Figure 24. The N+ circuit sub- serves as both the mechanical base for fabrication and a common ground or interconnection of all the grounded-emitter transistor/gates in a thic I^2L structure. One reason for I^2L high densities is apparent here; face metallization is required for ground interconnections as the entire t substrate serves this purpose. N epitaxial, grown on top of substrate, provides both the grounded-emitter region of the vertical itch and the grounded-base region of the lateral PNP injector. The of two diffusions serves as both the P base region of the vertical NPN collector of the lateral PNP injector. The second diffusion then com- the I^2L component geometries by providing the multiple-collector N+ s of the vertical NPN. Metallization is then deposited and etched to e interconnection between I^2L transistors/gates. Note that the lateral integrated into the vertical NPN and therefore does not exist as a te component. A symmetrical lateral PNP transistor can be utilized current injector for multiple NPN/gates. Non-isolated I^2L density is enhanced by the fact that the single transistor gate requires no ent isolation.

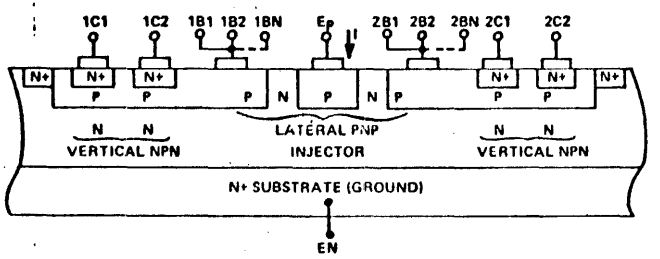


FIGURE 24. MONOLITHIC I^2L STRUCTURE

SBP0400 ELECTRICAL CHARACTERISTICS

NG

00 input/output characteristics were selected with one objective in mind - full TTL lity. The schematics and characteristics of the SBP0400 are shown in Figure 25. circuit chosen is actually an RTL configuration modified for TTL compatibility. threshold of nominally +1.5 volts is achieved by use of two 10K ohm resistors ng as a voltage divider to boost the one V_{BE} threshold of the input transistor to e input electrical characteristics are plotted as input current versus input volt- e both the 10K and 20K ohm load lines and the threshold knee at +1.5 volts. The dance, high-threshold characteristics were chosen to reduce input loading and

increase the input noise margin over a standard TTL input yet retain full capability with virtually all 5 volt logic families. The I^2L input also utilizes an input-clamping diode to limit negative excursions, "ringing", on the receiving end of a transmission line.

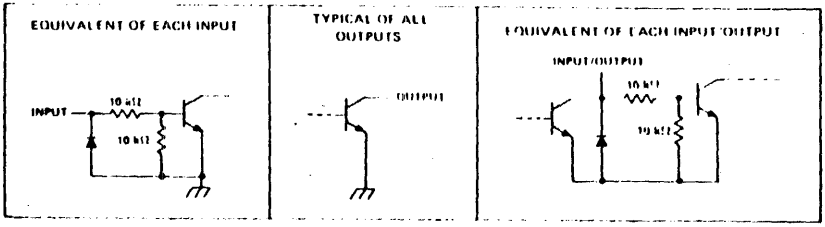


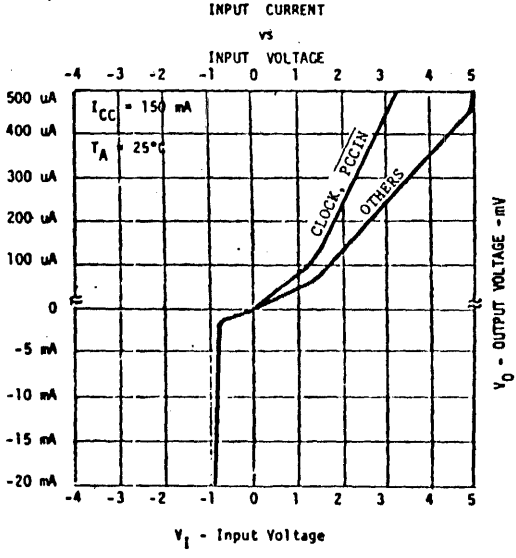
FIGURE 25. SCHEMATICS OF EQUIVALENT INPUTS, OUTPUTS, INPUTS/OUTPUTS

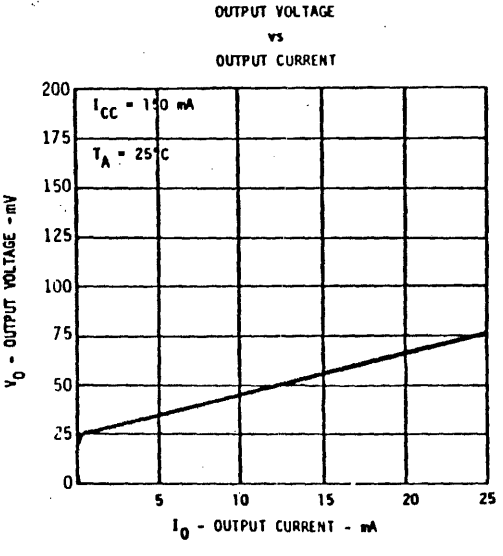
The output schematic is identical with that of an open-collector TTL circuit. The output characteristics are similar to that of the TTL output, but the I^2L output demonstrates a considerably improved low-level output voltage, typically 0.06v, at rated load currents. Typical V_{OL} versus I_{OL} is shown in Figure 26. The output high logic level, as well as output rise times, and next stage input noise immunity are a function of the load circuit used. The load can be:

a. The input of the next stage if no source current is required.

b. A discrete pull-up resistor for greater noise immunity and improved rise times.

Common input/output configurations are also utilized for improved functional performance and increased packing densities. The schematic is recognizable as a "joining" of the separate input/output schematics and electrical characteristics already described.





APPENDIX F: FIELD PROGRAMMABLE LOGIC ARRAYS

SIGNETICS 82S100/82S101 DEVICE DESCRIPTION

FPLA MANUAL FUSER CIRCUIT

;CRIPTION

82S100 (Tri-State Outputs) and the 82S101 en Collector Outputs) are Bipolar Programmable ic Arrays, containing 48 Product terms (AND ns), and 8 Sum terms (OR terms). Each OR term trols an output function which can be program- d either true active-High (Fp), or true active-Low). The true state of each output function is acti- ed by any logical combination of 16 input vari- es, or their complements, up to 48 terms. Both /ices are field-programmable, which means that stom patterns are immediately available by fol- ving the fusing procedure outlined in this data eet.

e 82S100 and 82S101 are fully TTL compatible, d include chip-enable control for expansion of ɔut variables, and output inhibit. They feature her Open Collector or Tri-State outputs for ease expansion of product terms and application in s-organized systems.

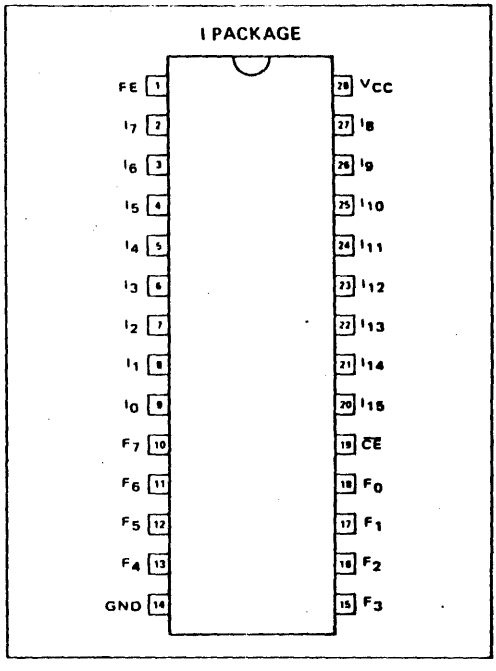
:ATURES

- FIELD PROGRAMMABLE (Ni-Cr LINK)
- INPUT VARIABLES—16
- OUTPUT FUNCTIONS—8
- PRODUCT TERMS—48
- ADDRESS ACCESS TIME—50 ns, MAXIMUM
- POWER DISSIPATION—600mW, TYPICAL
- INPUT LOADING—(−100μA), MAXIMUM
- OUTPUT OPTION:
  - TRI-STATE OUTPUTS—82S100
  - OPEN COLLECTOR OUTPUTS—82S101
- OUTPUT DISABLE FUNCTION:
  - TRI-STATE—Hi-Z
  - OPEN COLLECTOR—Hi
- CERAMIC DIP

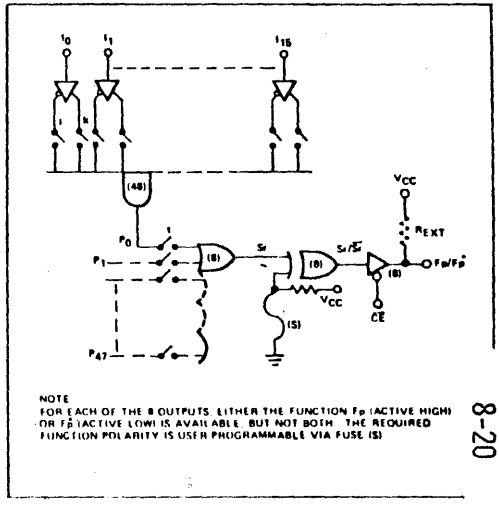
APPLICATIONS

- LARGE READ ONLY MEMORY
- RANDOM LOGIC
- CODE CONVERSION
- PERIPHERAL CONTROLLERS
- LOOK-UP AND DECISION TABLES
- MICROPROGRAMMING
- ADDRESS MAPPING
- CHARACTER GENERATORS
- SEQUENTIAL CONTROLLERS

PIN CONFIGURATION

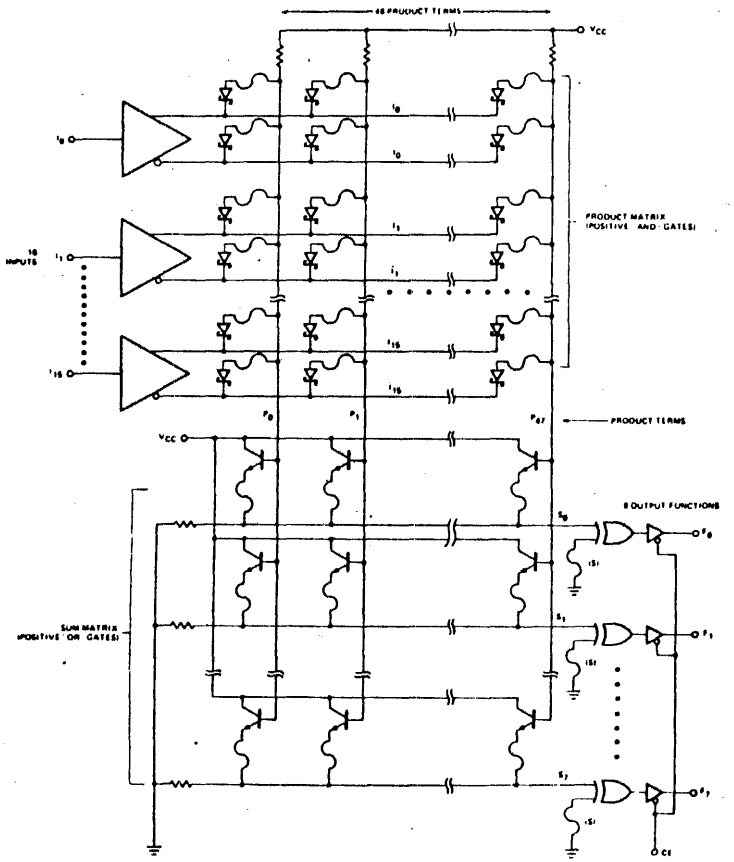


FPLA EQUIVALENT LOGIC PATH



NOTE
FOR EACH OF THE 8 OUTPUTS, EITHER THE FUNCTION Fp (ACTIVE HIGH) OR F̄p (ACTIVE LOW) IS AVAILABLE, BUT NOT BOTH. THE REQUIRED FUNCTION POLARITY IS USER PROGRAMMABLE VIA FUSE (S).

IAGRAM



BLE

$\sum_{0}^{15} (k_m l_m t_j m \overline{l_m})$; k = 0, 1, X (Don't Care)

n = 0, 1, 2 ... 47

MODE	P_n	$\overline{CE}$	$S_r \stackrel{?}{=} f(P_n)$	F_p	F_p^*
Disabled				1	1

ABSOLUTE MAXIMUM RATINGS

PARAMETER[1]		RATING	UNIT
V_{CC}	Power Supply Voltage	+7	Vdc
V_{in}	Input Voltage	+5.5	Vdc
V_{OH}	High Level Output Voltage (82S101)	+5.5	Vdc
V_O	Off-State Output Voltage (82S100)	+5.5	Vdc
T_A	Operating Temperature Range	0° to +75°	°C
T_{stg}	Storage Temperature Range	−65° to +150°	°C

ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq 75°C$; $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER[1]		TEST CONDITIONS		LIMITS MIN	LIMITS TYP[2]	LIMITS MAX	UNIT	NOTES
V_{IH}	High-Level Input Voltage	V_{CC} = 5.25V			2		V	1
V_{IL}	Low-Level Input Voltage	V_{CC} = 4.75V				0.8	V	
V_{IC}	Input Clamp Voltage	V_{CC} = 4.75, I_{IN} = −18mA			0.8	1.2	V	1, 7
V_{OH}	High-Level Output Voltage (82S100)	V_{CC} = 4.75V, I_{OH} = −2mA		2.4			V	1, 5
V_{OL}	Low-Level Output Voltage	V_{CC} = 4.75V, I_{OL} = 9.6mA			0.35	0.45	V	1, 8
I_{OLK}	Output Leakage Current (82S101)	V_{CC} = 5.25V	V_{OUT} = 5.25V		1	40	μA	6
$I_{O(OFF)}$	Hi-Z State Output Current (82S100)	V_{CC} = 5.25V	V_{OUT} = 5.25V		1	40	μA	6
			V_{OUT} = 0.45V		−1	−40	μA	
I_{IH}	High-Level Input Current	V_{IN} = 5.5V			<1	25	μA	
I_{IL}	Low-Level Input Current	V_{IN} = 0.45V			−10	−100	μA	
I_{OS}	Short-Circuit Output Current (82S100)	V_{CC} = 5.25V, V_{OUT} = 0V		−20		−70	mA	3, 7
I_{CC}	V_{CC} Supply Current (82S100, 82S101)	V_{CC} = 5.25V			120	170	mA	4
C_{IN}	Input Capacitance	V_{CC} = 5.0V	V_{IN} = 2.0V		5		pF	
C_O	Output Capacitance		V_{OUT} = 2.0V		8		pF	6

SWITCHING CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

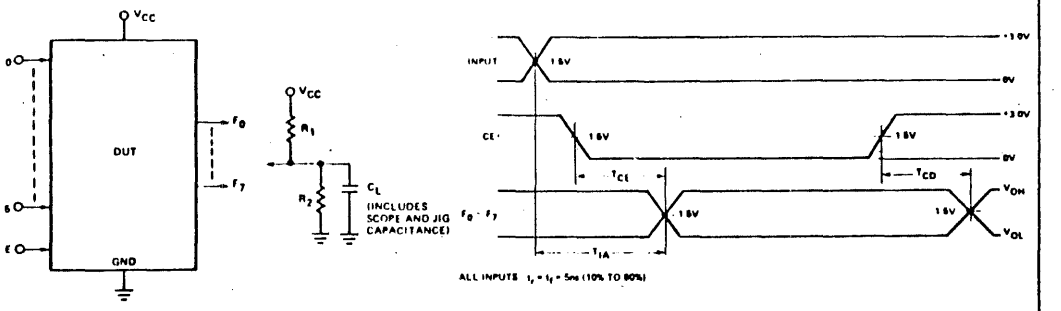
PARAMETER		TEST CONDITIONS	LIMITS MIN	LIMITS TYP[2]	LIMITS MAX	UNIT
Propagation Delay						
T_{IA}	Input to Output	C_L = 30pF		35	50	ns
T_{CD}	Chip Disable to Output	R_1 = 270		15	30	ns
T_{CE}	Chip Enable to Output	R_2 = 600		15	30	ns

NOTES:
1. All voltage values are with respect to network ground terminal.
2. All typical values are at V_{CC} = 5V, T_A = 25°C.
3. Duration of short circuit should not exceed one second.
4. I_{CC} is measured with the chip enable input grounded, all other inputs at 4.5V and the outputs open.
5. Measured with V_{IL} applied to $\overline{CE}$ and a logic "1" stored.
6. Measured with V_{IH} applied to $\overline{CE}$.
7. Test each output one at the time.
8. Measured with a programmed logic condition for which the output under test is at a "0" logic level. Output sink current is supplied thru a resistor to V_{CC}.

AMMING SPECIFICATIONS[4] (Testing of these limits may cause programming of device.) T_A = +25°C

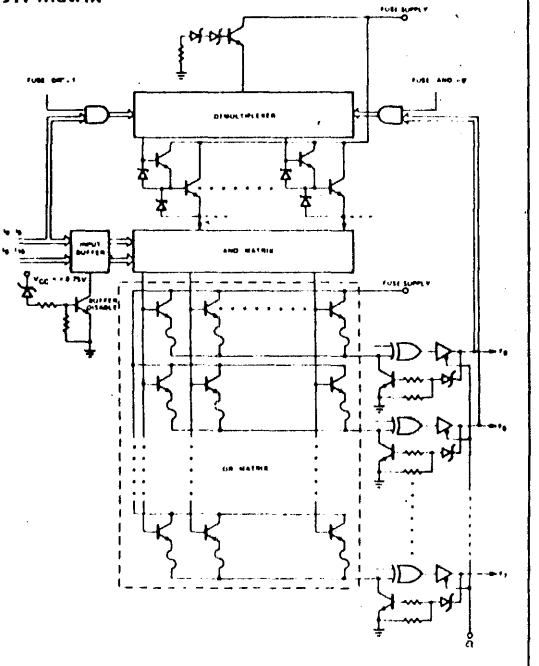
PARAMETER	TEST CONDITIONS	LIMITS			UNIT
		MIN	TYP	MAX	
V_{CC} Supply (Program "OR")	I_{CCS} = 550 mA, min. (Transient or steady state)	8.5	8.75	9.0	V
V_{CC} Supply (Program Output Polarity)		0	0.4	0.8	V
I_{CC} Limit (Program "OR")	V_{CCS} = +8.75 ±.25V	550		1,000	mA
Output Voltage (Program Output Polarity)	I_{OPH} = 300 ±25mA	16.0	17.0	18.0	V
Output Voltage (Idle)		0	0.4	0.8	V
Output Current Limit (Program Output Polarity)	V_{OPH} = +17 ±1V	275	300	325	mA
Input Voltage (Logic "1")		2.4		5.5	V
Input Voltage (Logic "0")		0	0.4	0.8	V
Input Current (Logic "1")	V_{IH} = +5.5V			50	μA
Input Current (Logic "0")	V_{IL} = 0V			−500	μA
Forced Output (Logic "1")		2.4		5.5	V
Forced Output (Logic "0")		0	0.4	0.8	V
Output Current (Logic "1")	V_{OHF} = +5.5V			100	μA
Output Current (Logic "0")	V_{OLF} = 0V			−1	mA
$\overline{CE}$ Program Enable Level		9.5	10	10.5	V
Input Variables Current	V_{IX} = +10V			2.5	mA
$\overline{CE}$ Input Current	V_{IX} = +10V			5.0	mA
FE Supply (Program)	I_{FEH} = 300 ±25mA (Transient or steady state)	16.0	17.0	18.0	V
FE Supply (Idle)		0	0.4	0.8	V
FE Supply Current Limit	V_{FEH} = +17 ±1V	275	300	325	mA
V_{CC} Supply (Program "AND")	I_{CCP} = 550 mA, min. (Transient or steady state)	4.75	5.0	5.25	V
I_{CC} Limit (Program "AND")	V_{CCP} = +5.0 ±.25V	550		1,000	mA
Forced Output (Program)		9.5	10	10.5	V
Output Current (Program)				10	mA
Output Pulse Rise Time		10		50	μs
$\overline{CE}$ Programming Pulse Width		1		1.5	ms
Pulse Sequence Delay		10			μs
Programming Time			2		ms
Programming Duty Cycle				50	%
Fusing Attempts per Link				3	cycle
Verify Threshold		0.9	1.0	1.1	V

: TEST FIGURE AND WAVEFORMS

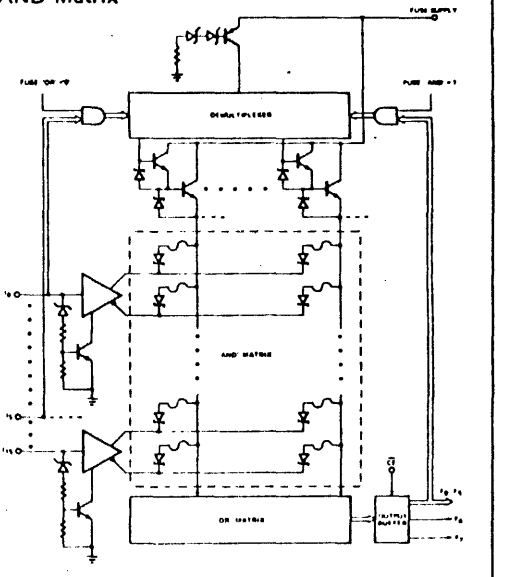


YPICAL FUSING PATHS

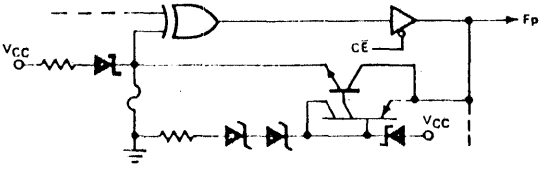
)R Matrix



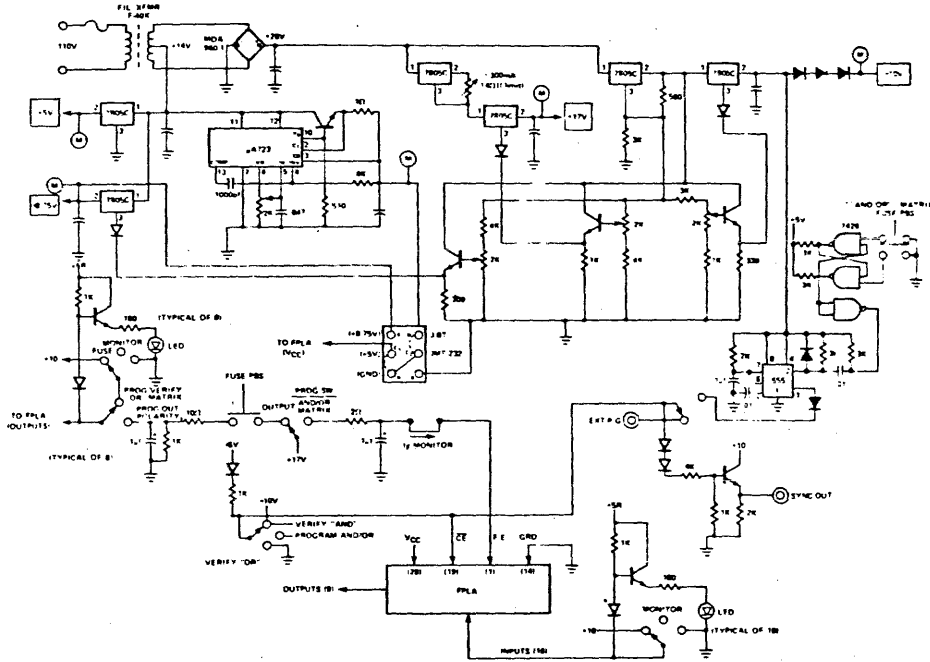
AND Matrix



OUTPUT POLARITY



FPLA MANUAL FUSER CIRCUIT



MICROPROGRAM ASSEMBLER: SYNTAX DESCRIPTION

Definitions

The RAPID assembly language consists of six kinds of statements. In this section we will begin the definition of these statements by introducing the character set out of which the statements are formed, the notation in which we will define elements of statements, and the language primitives common to all the statements.

The character set RAPID recognizes may be viewed as having three parts; alphabetic, numeric, and special symbols. The alphabetic part contains the twenty-six letters, A, B, C, . . . , Z. The ten digits 0, 1, 2, . . ., 9 comprise the numeric part. The special symbols are eleven characters used for punctuation and operator names; ampersand (&), apostrophe ('), asterisk (*), comma (,), dollar sign ($), equals (=), minus (-), parentheses (()), plus (+), and slash (/). The uses of the characters are fully described in the definitions.

Statement definitions will be given in the form of syntax equations. That is, statements will be divided (parsed, really) into syntactic elements and each of the elements defined. These elements will often be further divided into more primitive syntactic elements and those elements defined. This process is repeated until the definitions finally resolve to the character set described above.

The definitions accomplish three things:

1. They name the syntactic elements of each statement.

2. They specify the order in which those elements occur in the statements.

3. They indicate what characters are used to form each element.

The syntax equations possess their own notation. These symbols are not part of the character set used to express statements, but merely give form to the definitions. This notation is as follows:

<>	The left and right acute brackets enclose the names of syntactic elements.
:=	This symbol separates the name of an element from its definition; it means "is defined as".
|	The vertical bar indicates an alternative definition; that is, the elements on either side of the bar are equally valid.

$\{\ \}_n$ The syntactic elements enclosed in braces may occur at this place in the statement zero or more times, up to n times maximum; if no subscript is given, either there is no limit to the repetitions, or it depends on other variables.

$[\ ]_n$ The syntactic element enclosed in brackets begins in card column n.

n example, we can define the character set for statements:

⟨alphabetic⟩ : = | A | B | C | . . . | X | Y | Z

⟨numeric⟩ : = | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

⟨special symbol⟩ : | = | | ' | * | , | $ | = | - | (|) | + | /

se equations give a name to each of the 47 characters. Since we are in-
ested only in what set of characters applies to each name, the elements
given as alternatives; 2 is as valid a numeric as 9 or $ is as valid a
cial symbol as *. Whenever a character or sequence of characters occurs
side the acute brackets, it is a literal (it represents itself) and should
written in a statement exactly as shown.

name of a syntactic element may be used in a definition as well as literals.
s is in fact how statement definitions are built from more primitive defini-
ns. For instance:

⟨alphanumeric⟩ : = ⟨alphabetic⟩ | ⟨numeric⟩

ines an alphanumeric character as any one of thirty-six alphabetics and
erics.

many definitions, we will be interested in the order of elements. To specify
articular order, we merely write one element after the other without any
ervening vertical bar. Thus:

⟨symbol⟩ : = ⟨alphabetic⟩ $\{$ ⟨alphanumeric⟩ $\}_4$

cifies that the element known as a symbol always begins with an alphabetic
aracter and may be followed by as many as four alphanumerics. Examples of
bols are A, PQ, B101, and Z0Z1. 5E and ABCDEF would not be valid. RAPID
l detect any violation of these syntax definitions and signal an error on
assembly listing.

he rest of the primitives deal with the definition of constants:

⟨constant⟩ : = ⟨integer⟩ |
'A { ⟨alphabetic⟩ } ~ 5 |
'B { ⟨binary⟩ } ~ 128 |
'H { ⟨hexadecimal⟩ } ~ 32 |
'O { ⟨octal⟩ } ~ 43 |

⟨integer⟩ : = ⟨numeric⟩ { ⟨numeric⟩ } ~ 8 |

⟨binary⟩ : = 0 | 1

⟨hexadecimal⟩ : = ⟨numeric⟩ | A | B | C | D | E | F

⟨octal⟩ : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

constant, as will be seen later, has several very specific uses. Quite ten a constant is a decimal number, the integer defined above, of no more an nine digits. Alphabetic, binary, hexadecimal, and octal constants are ecified by prefixing an apostrophe and A, B, H, or O to the constant. Conants must fit in whatever field they are designed for; the maximum lengths ecified here will rarely, if ever, be required. Examples of constants are: 177, 'HFF, 899, 'B101, and 'AXY. Note that 'B101 is the three-bit constant ual to 5, and B101 without the apostrophe is a symbol. The primitives of e RAPID assembly language, then, are symbols and constants.

this discussion of statements proceeds in the following chapters, each oup of syntax equations is followed by an explanation and several examples clarify the definitions. This manner of defining statements, a modificaon of the Backus Normal Form, is quite precise. Once a reader becomes quainted with it, he should find it easy and unambiguous to use as a reence.

mary of Statements

s section briefly introduces the six statement types. The definitions for h type will be given completely at the beginning of their respective pters.

⟨RAPID input⟩ : = ⟨option statement⟩⟨parity statement⟩
⟨format statements⟩ ⟨opcodes statement⟩
⟨program section⟩ ⟨end statement⟩

e statements must appear in the specified order.

⟨option statement⟩ : = [$ROM] $_1$ ⟨option specifications⟩

ıe option specifications describe the bit storage configurations of ROM, its ıpping into words, and call for various assembly outputs.

⟨parity statement⟩ : = [$PARITY] $_1$ ⟨parity specifications⟩

ırity specifications describe how parity bits, if any are desired, will be ɜnerated in each output word. For example, this permits the designer to in- icate he wants bit 5 in every word to reflect odd parity.

⟨format statements⟩ : = [$FORMATS] $_1$ ⟨format specifications⟩

hese statements name and describe the various formats that output words can ake and set up the correspondence between input fields and output fields. or example, one statement might specify that the symbol in the third input ield (in a program statement naming a given format) be translated into a code hich will appear in bits 12 through 15 in the output word.

⟨opcodes statement⟩ : = [$OPCODES] $_1$ ⟨opcode specifications⟩

he opcodes specifications list the operation codes in symbolic form and the ıachine language representations into which they should be translated. For :xample, it could instruct RAPID to translate the opcode ADD into the binary :ode 1010.

⟨program section⟩ : = [$PROGRAM] $_1$ ⟨program statements⟩

The program section contains the symbolic program for ROM as a series of state- nents. Using the previous four kinds of statements, RAPID interprets program statements and translates them to machine language.

⟨end statement⟩ : = [$END] $_1$

The end card terminates the input to RAPID. It contains no other information.

Each of these six statements begins with $ in column one. The option, parity, and end statements must each be given on one card. The others may use as many cards as desired. At least one blank must follow the words ROM, PARITY, FORMATS, and OPCODES, and the names of formats in format and program statements.

OPTION STATEMENT

ption statement provides RAPID with information about the configuration e ROM modules, the width of the program word, and the types of output ed from the assembler.

⟨option specifications⟩ : = ⟨option⟩ $\left\{, \langle option \rangle\right\}_5$

⟨option⟩ : = CHIP = ⟨depth⟩ * ⟨width⟩ |
WORD = ⟨word size⟩ |
LIST |
MASK |
MEMORY = ⟨table type⟩ |
NULL = ⟨binary⟩

⟨depth⟩ : = ⟨integer⟩

⟨width⟩ : = ⟨integer⟩

⟨word size⟩ : = ⟨integer⟩

⟨table type⟩ : = COMPACT |
EXTENDED

·e 8 shows a number of legitimate option statements. This statement may ıave continuation cards. One option is separated from the next by a ı; between the word ROM and the first option is at least one blank.

:HIP option describes to RAPID the configuration of the memory element to implement the control store. For example, if a 256-bit semiconductor (Random Access Memory) is used and it is organized 128 by 2 bits, the ification would be CHIP = 128*2. In the case of a 4096 bit ROM (Read Memory) it might be CHIP = 4096*1, CHIP = 1024*4, CHIP = 512*8, etc. vidth may be no greater than 64 bits.

vord option specifies the number of bits of a word in ROM. The maximum size is 128 bits. RAPID will assume as many chips side-by-side as ssary to provide the specified word size.

ɔutput options, LIST and MASK control listing and punched card output, ectively. LIST causes all of the input to be listed. Any cards in h RAPID detects errors will be listed with a diagnostic message, whether is selected or not.

PARITY STATEMENT

Output words from a control store may contain one or more parity bits to validate the contents of the words. The parity statement describes to RAPID what parity bits, if any, to generate for each word.

⟨parity specifications⟩ : = ⟨parity declaration⟩ { , ⟨parity declaration⟩ } 4 | NONE

⟨parity declaration⟩ : = ⟨parity type⟩ (⟨bit position⟩) = ⟨field⟩ { + ⟨field⟩ }

⟨parity type⟩ : = ODD | EVEN

⟨bit position⟩ : = ⟨integer⟩

⟨field⟩ : = (⟨left bit⟩ / ⟨length⟩)

⟨left bit⟩ : = ⟨integer⟩

⟨length⟩ : = ⟨integer⟩

The parity specifications must occur on one card; hence, no more than five may be given. Each declaration is separated from the next by a comma, and the first from the word PARITY by at least one blank. If no parity bits are to be generated, the word NONE is written.

For each specified parity bit, its type, odd or even, is indicated as well as the fields over which parity is computed. A field is defined by giving the number of the left-most bit position of the field and the total number of bits in the field. For instance, the three fields shown in the word in Figure 10 would be specified as (6/6), (13/11), and (12/1). Fields may overlap one another, but not the parity bit positions themselves.

Figure 11 shows several examples of parity statements for a 64-bit word. The last example defines six fields. Bits 1-16 contribute to even parity in bit 0; bits 18-21, 26-35 and 55-63 contribute to odd parity in bit 54; and bits 22-25 and 36-53 contribute to odd parity in bit 17. Notice that parity declarations may be given in any order in the statement.

FORMAT STATEMENTS

t statements establish the correspondence between input fields and elds. By careful use of these statements and the opcodes statement ion on Opcodes Statements), the designer can define a convenient for expressing his control program.

mat specifications> : = $\left[\langle\text{format id}\rangle\right]_1$ ⟨format definition⟩ $\left\{,\langle\text{format definition}\rangle\right\}$

mat id> : = ⟨alphabetic⟩ $\left\{\langle\text{alphabetic}\rangle\right\}_4$

mat definition> : = ⟨field⟩ = (⟨field type⟩)

ld type> : = A | O | T

r of formats may be specified. Each format is a separate statement ins in column one with the name of the format, up to five characters. nt may have continuation cards, though a format definition may not across cards. That is, a right parenthesis and comma must complete definition before continuing the statement on the next card. The ction of format statements is headed by the $FORMATS statement card. 2 is an example of a format section.

grammed systems often use several control word or microinstruction Each format is independent of the others, though, of course, they y to words of the same length as defined in the option statement. e 12, the three formats named CMD, TEST, and JUMP clearly identify truction type.

definition describes one field. A field is specified as shown in ious section; field type must be Address, Opcode, or Truncated address. statement may contain any number of definitions in any order. How- elds must not overlap one another or the parity bits, and they must in the ROM words.

point, it is important to understand that each program statement er to a format. It is that named format which will guide the trans- f the program statement. Information in the program statement must ed with the fields of its format. This matching is accomplished in . An opcode is matched against a list in the opcodes statement (see on Opcodes Statement); that list points to one of the fields of the nd the value associated with that opcode is placed in that field. ively, an address expression in the program statement itself points f the fields of the format; the result of the expression fills that The Program Statements Section describes fully this matching process kinds of information in program statements.

pcodes are the mnemonic symbols the programmer may use in program statements o direct that specific fields in the output word be given certain values. he opcodes statement lists all the valid symbols for each opcode field of ach format and their associated values.

⟨opcode specifications⟩ : = ⟨opcode field⟩ { ⟨opcode field⟩ }

⟨opcode field⟩ : = ⟨format id⟩ ⟨field number⟩ = (⟨opcode⟩ { , ⟨opcode⟩ })

⟨field number⟩ : = ⟨integer⟩

⟨opcode⟩ : = ⟨symbol⟩ / ⟨constant⟩

he opcodes specification is one statement. As many continuation cards as lesired may be used. Cards may break the statement at three places: immediately ollowing a right parenthesis completing an opcode field specification; after comma between opcodes for one field; or after a hyphen within a numeric onstant. In the latter case, the numeric constant is hyphenated on one card nd the rest of the constant continues (somewhere beyond column one) on the ollowing card. This is a useful facility mainly for long binary constants.

he programmer should define one opcode field for each opcode type in the ormat statements. He may list as many opcodes for one field as he wishes. igure 13 illustrates an opcode statement for the CMD, TEST and JUMP formats of Figure 12.

ormat id and field number together point to a specific field in a format statement. Format id identifies the format statement. Fields in that statement are numbered, starting with 1, in the order they are defined. It is this order of definition that is important, not the left-to-right order in the word. Thus, field JUMP1 of Figure 12 is the twelve-bit address field beginning in bit six.

There are three kinds of errors in the opcodes statement. First, any field pointed to must be an opcode type. Second, the value associated with a symbol must fit into the designated field. And third, for one format, no opcode symbol may be repeated.

The reason for this last restriction will become clear in the next section. Briefly, for a given format, a particular opcode field is singled out in a program statement by the appearance of one of the opcode symbols defined for it. Thus, all of the opcode fields of that format must share the same set of symbols; if one symbol is repeated, it is ambiguous in identifying a field. Of course, from format to format symbols may be repeated.

PROGRAM STATEMENTS

tatements represent the encoding of the customer's control logic. logy, formats, and mapping he has designed previously are all used ate this symbolic program to a binary representation for ROM. The lready established, is now used:

rogram statements> : = <custom statement> | <predefined statement>

ustom statement> : = [<format id>]$_1$ {<label symbol>}$_1$ <separator> <program field> {<separator> <program field>}

label symbol> : = /<symbol>

separator> : = , | <blank>

program field> : = <opcode symbol> | <address expression>

pcode symbol> : = <symbol>

address expression> : = <field number> (<expression>)

expression> : = {<operator>}$_1$ <term> {<operator> <term>}

operator> : = + | -

term> : = <label symbol> |
<constant> |
*

predefined statement> : = [*]$_1$ <predefined type>

predefined type> : = <align statement> |
<constant statement> |
<equate statement>

⟨align statement⟩ : = $\left\{\langle\text{label symbol}\rangle\right\}_1$ ALIGN ⟨constant⟩

⟨constant statement⟩ : = $\left\{\langle\text{label symbol}\rangle\right\}_1$ CONSTANT ⟨constant⟩

⟨equate statement⟩ : = ⟨label symbol⟩ EQUATE ⟨expression⟩

The program section begins with the $PROGRAM statement and ends with the $END statement. Each statement between these two is a program statement. Theoretically, there is no limit to the number of statements to a program. Program statements fall into two major classes: those whose symbology the customer has defined, the predefined statements. We will consider these two classes separately.

MICROPROCESSOR BOOKS

THE FOLLOWING BOOKS MAY BE OBTAINED FROM SYBEX :

A 1 - MICROCOMPUTERS :
FROM CHIPS TO SYSTEMS

A 2 - MICROPROCESSOR PROGRAMMING

B 3 - SEVERE - ENVIRONMENT MICROPROCESSOR SYSTEMS :
MILITARY, AVIONICS, AEROSPACE, MARINE, INDUSTRIAL

B 5 - BIT - SLICE :
TECHNOLOGY AND APPLICATIONS

CONTACT US FOR SEMINARS AND SPECIAL IN - HOUSE COURSES.

USA:
SYBEX INCORPORATED
510 GRIZZLY PEAK BLVD.
BERKELEY, CALIFORNIA 94708
TEL : (415) 526 - 2748

EUROPE:
SYBEX - EUROPE
313 RUE LECOURBE
75015 - PARIS, FRANCE
TEL : (1) 828 - 2502

SYBEX

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS ARE GRATEFULLY EXTENDED TO THE FOLLOWING COMPANIES OR ORGANIZATIONS W
HAVE PROVIDED TECHNICAL INFORMATION, PERSONAL INTERVIEWS, OR AUTHORIZED THE REPRODUCTI
OF COPYRIGHTED MATERIAL.

ACTRON
ADVANCED MICRO DEVICES
AUTONETICS
AMERICAN MICROSYSTEMS
CMC
DATA GENERAL
DCA LABS
DEC
E & L
ELECTRONIC ARRAYS
EMM
FAIRCHILD SEMICONDUCTOR
FLUKE
GENERAL AUTOMATION
GENERAL INSTRUMENTS
HARRIS SEMICONDUCTOR
HEWLETT - PACKARD
HUGHES
IBM
IMSAI
INTEL CORPORATION
INTERSIL
LOCKHEED
MARTIN RESEARCH
MICROCOMPUTER ASSOCIATES
MONOLITHIC MEMORIES
MOS TECHNOLOGY
MOSTEK CORPORATION
MOTOROLA SEMICONDUCTOR
MYCRO - TEK
NATIONAL SEMICONDUCTOR
NEC
NELC
OSBORNE AND ASSOCIATES
PHILIPS
PLESSEY
PROLOG
R 2 E
RAYTHEON
RCA SOLID STATE
ROCKWELL INTERNATIONAL
SCIENTIFIC MICROSYSTEMS
SESCOSEM
SIEMENS
SIGNETICS
TEKTRONIX
TEXAS INSTRUMENTS
TRANSITRON ELECTRONICS
TRENDAR
WESTERN DIGITAL CORPORATIC
ZILOG

AMD (ADVANCED MICRODEVICES)
901 THOMPSON PLACE, SUNNYVALE, CALIFORNIA 94806
(408) 732 - 2400 TELEX : 346306

AMI (AMERICAN MICROSYSTEMS)
3800 HOMESTEAD ROAD, SANTA CLARA, CALIFORNIA 95051
(408) 246 - 0330

DATA GENERAL
SOUTHBORO, MASSACHUSETTS 01772
(617) 485 - 9100 TELEX : 48460

FAIRCHILD SEMICONDUCTOR
1725 TECHNOLOGY DRIVE, SAN JOSE, CALIFORNIA 95110
(408) 998 - 0123

MICROPROCESSOR MANUFACTURERS (II)

GI (GENERAL INSTRUMENTS)

600 WEST JOHN STREET, HICKSVILLE, NEW YORK 16002

(516) 733 - 3107 TWX : (510) 221 - 1666

INTEL

3065 BOWERS AVENUE, SANTA CLARA, CALIFORNIA 95051

(408) 246 - 7501 TELEX : 346372

INTERSIL

10090 NORTH TANTAU AVENUE, CUPERTINO, CALIFORNIA 95014

(408) 996 - 5000 TWX : (916) 338 - 0228

MMI (MONOLITHIC MEMORIES)

1165 EAST ARQUES AVENUE, SUNNYVALE, CALIFORNIA 94086

(408) 739 - 3535

MOS TECHNOLOGY

MICROPROCESSOR MANUFACTURERS (III)

MOSTEK

1215 WEST CROSBY ROAD, CAROLLTON, TEXAS 75006

(214) 242 - 0444 TELEX : 30423

MOTOROLA SEMICONDUCTOR

BOX 20912, PHOENIX, ARIZONA 85036

(602) 244 - 6900 TELEX : 67325

NS (NATIONAL SEMICONDUCTOR)

2900 SEMICONDUCTOR DRIVE, SANTA CLARA, CALIFORNIA 95051

(408) 732 - 5000 TWX : (910) 339 - 9240

RCA SOLID STATE

BOX 3200, SOMERVILLE, NEW JERSEY 08876

(201) 722 - 3200 TWX : (718) 480 - 9333

ROCKWELL INTERNATIONAL

BOX 3669, ANAHEIM, CALIFORNIA 92803

(714) 632 - 3698

SIGNETICS

811 EAST ARQUES AVENUE, SUNNYVALE, CALIFORNIA 94086

(408) 739 - 7700

TI (TEXAS INSTRUMENTS) - DIGITAL SYSTEMS DIVISION

P.O. BOX 1444, HOUSTON, TEXAS 77001

(713) 494 - 5115

WESTERN DIGITAL

3128 REDHILL AVENUE, NEWPORT BEACH, CALIFORNIA 92663

(714) 557 - 3550 TWX : (910) 595 - 1139

ZILOG

170 STATE STREET, LOS ALTOS, CALIFORNIA 94022

(415) 526 - 2748 TWX : (910) 370 - 7955

MICROPROCESSOR BOOKS

B 1 - MICROPROCESSORS: FROM CHIPS TO SYSTEMS
B 2 - MICROCOMPUTER PROGRAMMING
B 3 - MILITARY MICROPROCESSOR SYSTEMS
B 5 - BIT SLICE
B 6 - INDUSTRIAL MICROPROCESSOR SYSTEMS
B 7 - MICROPROCESSOR INTERFACING TECHNIQUES
C 4 - LES MICROPROCESSEURS (IN FRENCH)
C 10 - INTRODUCTION AUX MICROPROCESSEURS (IN FRENCH)
E 5 - MICROPROCESSOR ENCYCLOPEDIA: BIT SLICES
E 8 - MICROPROCESSOR ENCYCLOPEDIA: 8-BITS
F 1 - THE 30 MICROPROCESSOR FAMILIES - THE CHIPS
F 2 - THE 30 MICROPROCESSOR FAMILIES - INSTRUCTION SETS
M 7 - INTERNATIONAL MICROPROCESSOR DICTIONARY
M 11 - AN INTRODUCTION TO MICROCOMPUTERS: BASIC CONCEPT
M 12 - AN INTRODUCTION TO MICROCOMPUTERS: SOME REAL PROD
M 13 - 8080 PROGRAMMING FOR LOGIC DESIGN
M 14 - 6800 PROGRAMMING FOR LOGIC DESIGN
ACR - ACRONYMS
L 1 - MICROPROCESSOR DESIGN LABORATORY

SYBEX

IN HOUSE COURSES

SYBEX seminars are presented throughout the world in major cities at regular intervals. They can also be taught at your facility for a minimum number of participants (10 to 15) at substantial savings.

Partial list of courses:

A 1 - Microprocessors
A 2 - Programming Microprocessors
B 3 - Military Systems
B 4 - Microprocessors for Microwave Engineers
B 5 - Bit-Slice
B 6 - Industrial Applications
B 7 - Interfacing
M 12 - Comparative Evaluation
L 1 - Laboratory Course
MOS 1 - MOS fabrication
MOS 2 - MOS circuit design
MOS 3 - N-channel process control
P 20 - Microprogrammed APL

Contact the nearest SYBEX office for full details.

SYBEX

ITC sa 113, rue de Javel Paris – dl 1096 - 2ème tr. 77 - ISBN 2-902414-04-8 – PRINTED IN FRANCE