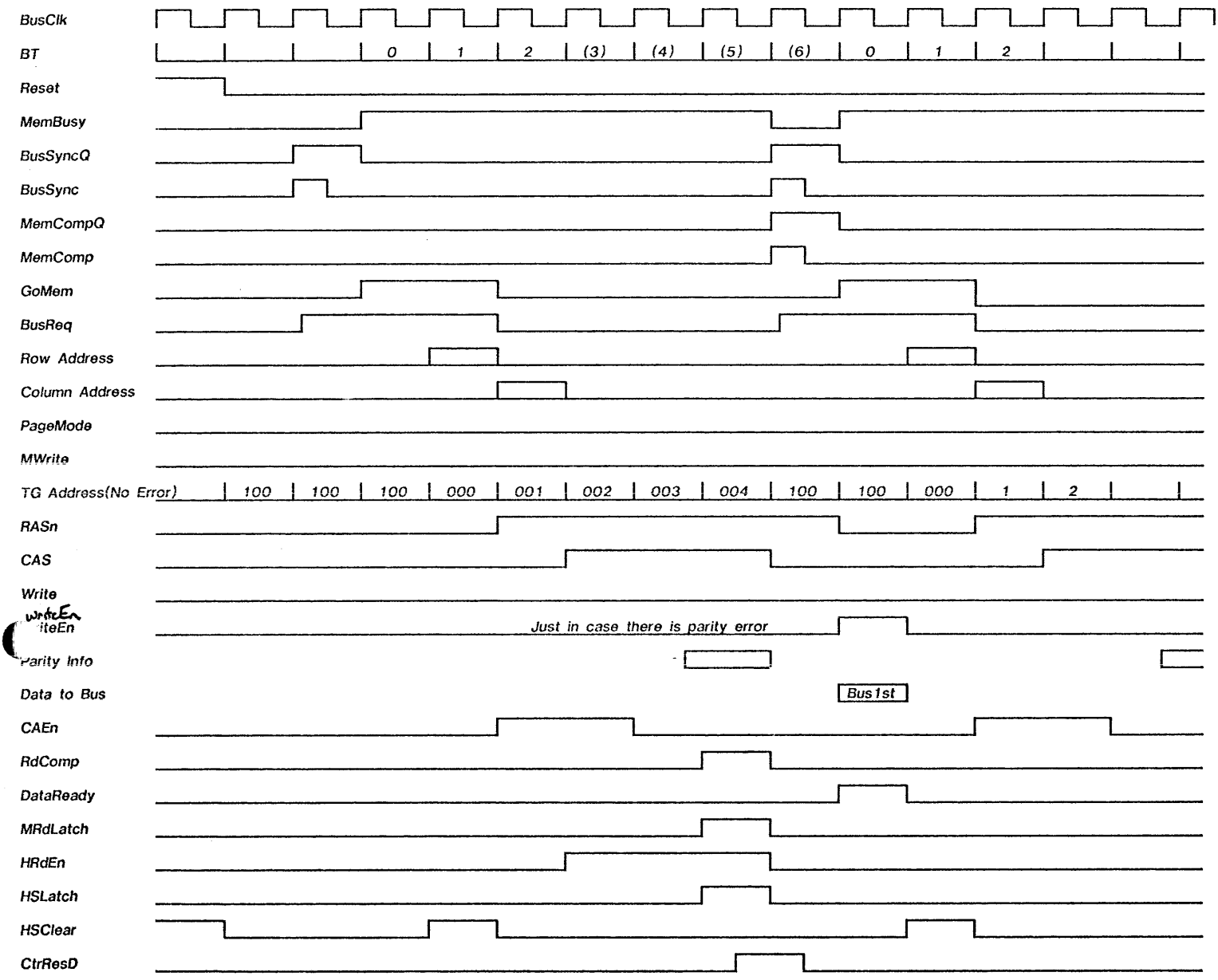
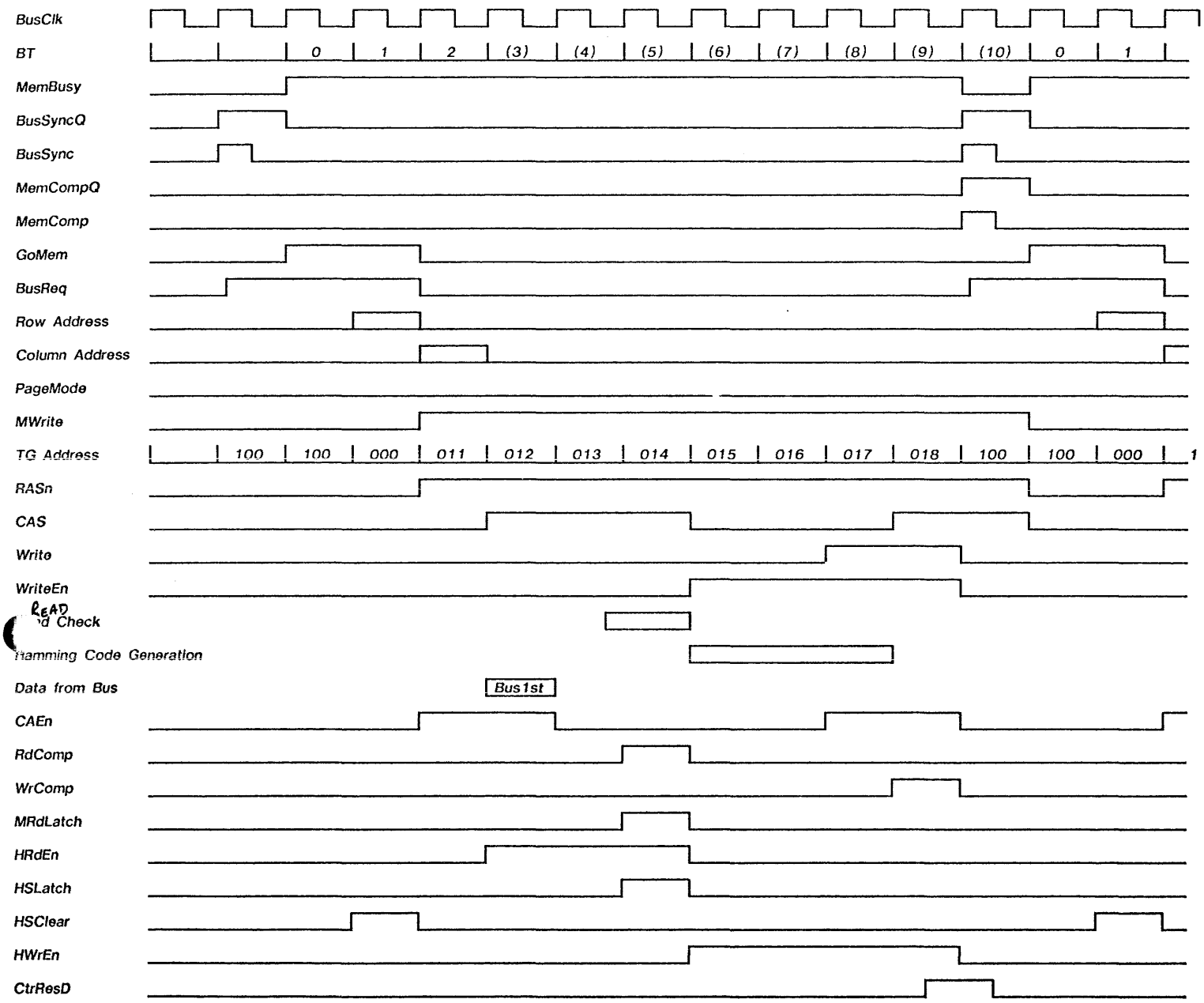
Memory Read Timing



XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt01.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 1 of 11

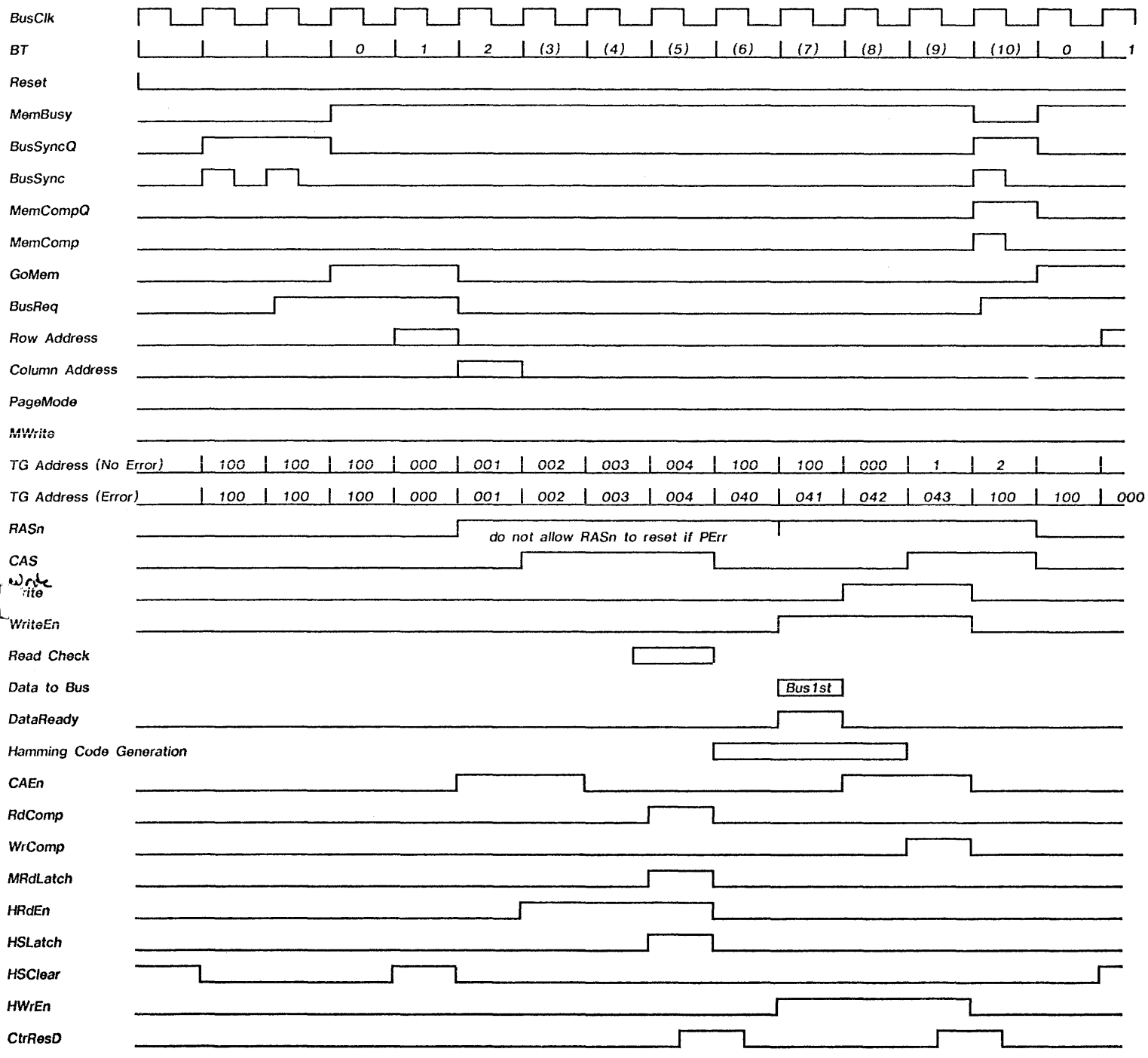
Memory Write Timing



[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetaker	Title: Memory Address Timing	File: ntmt02.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 2 of 11

Memory Read Timing (With Error)

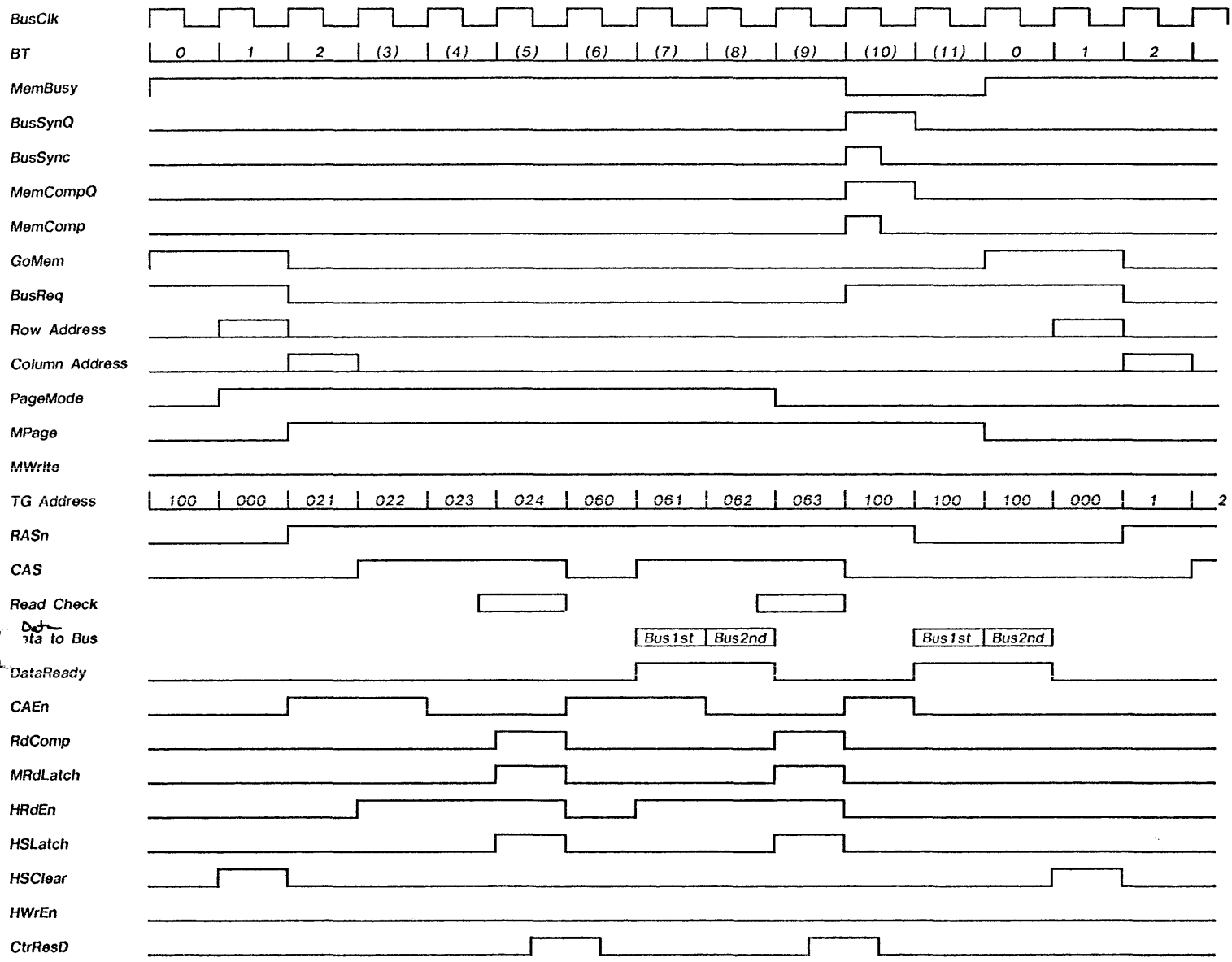


[ifs-2] < note > ntmt.dm

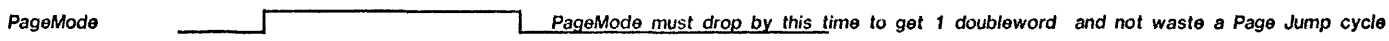
XEROX ASD	Project	Title	File	Designer	Rev	Date	Page
	Notetaker	Memory Address Timing	ntmt03.sil	Sato	A	10-27-78	3 of 11

d

Memory Page Read Timing



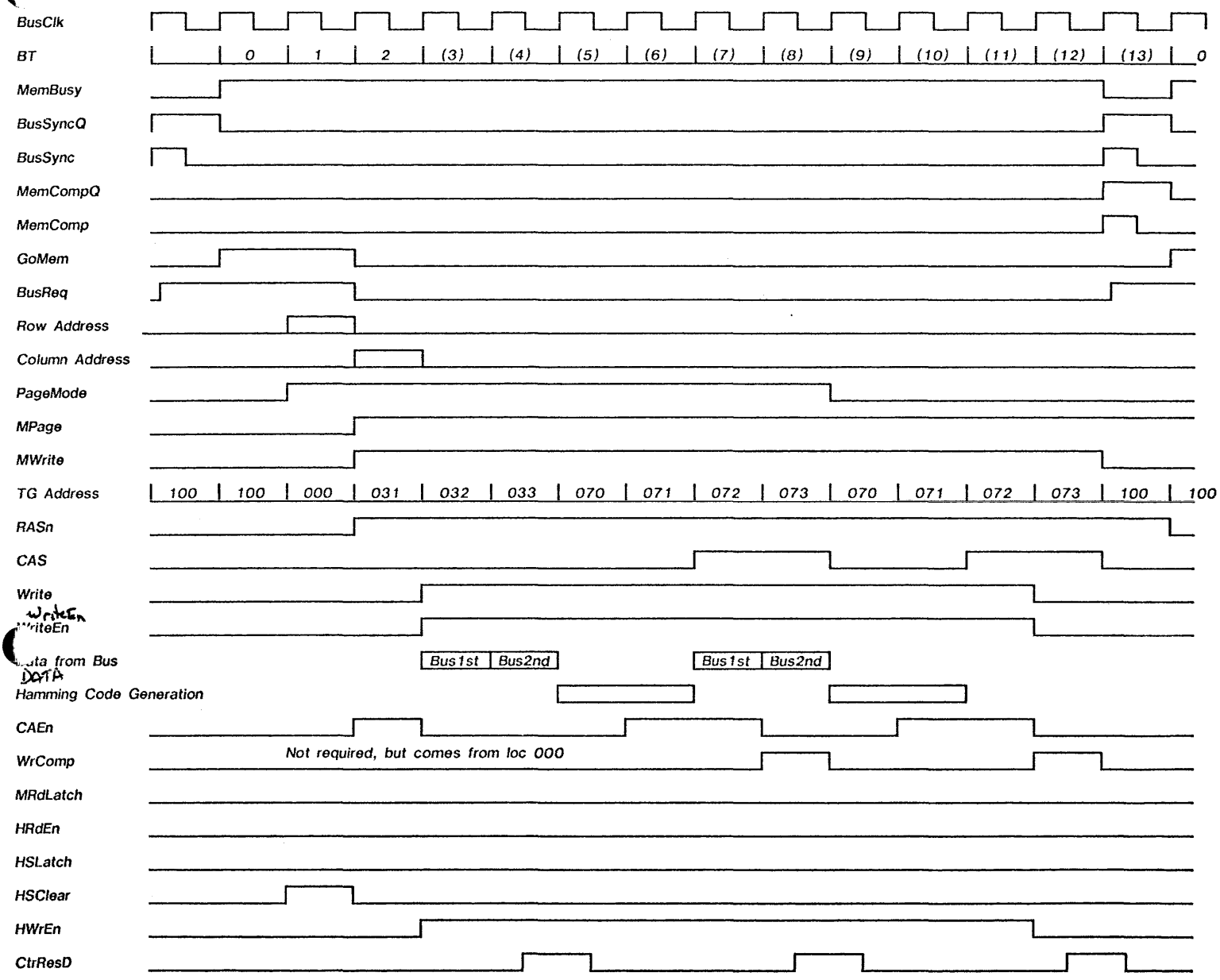
Note: If PageMode is dropped one clock time after Column Address for 1 doubleword, or after the previous DataReady for 2 or more doublewords, the next user can get the bus before the timing generator starts a new Page Jump cycle.



[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt04.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 4 of 11

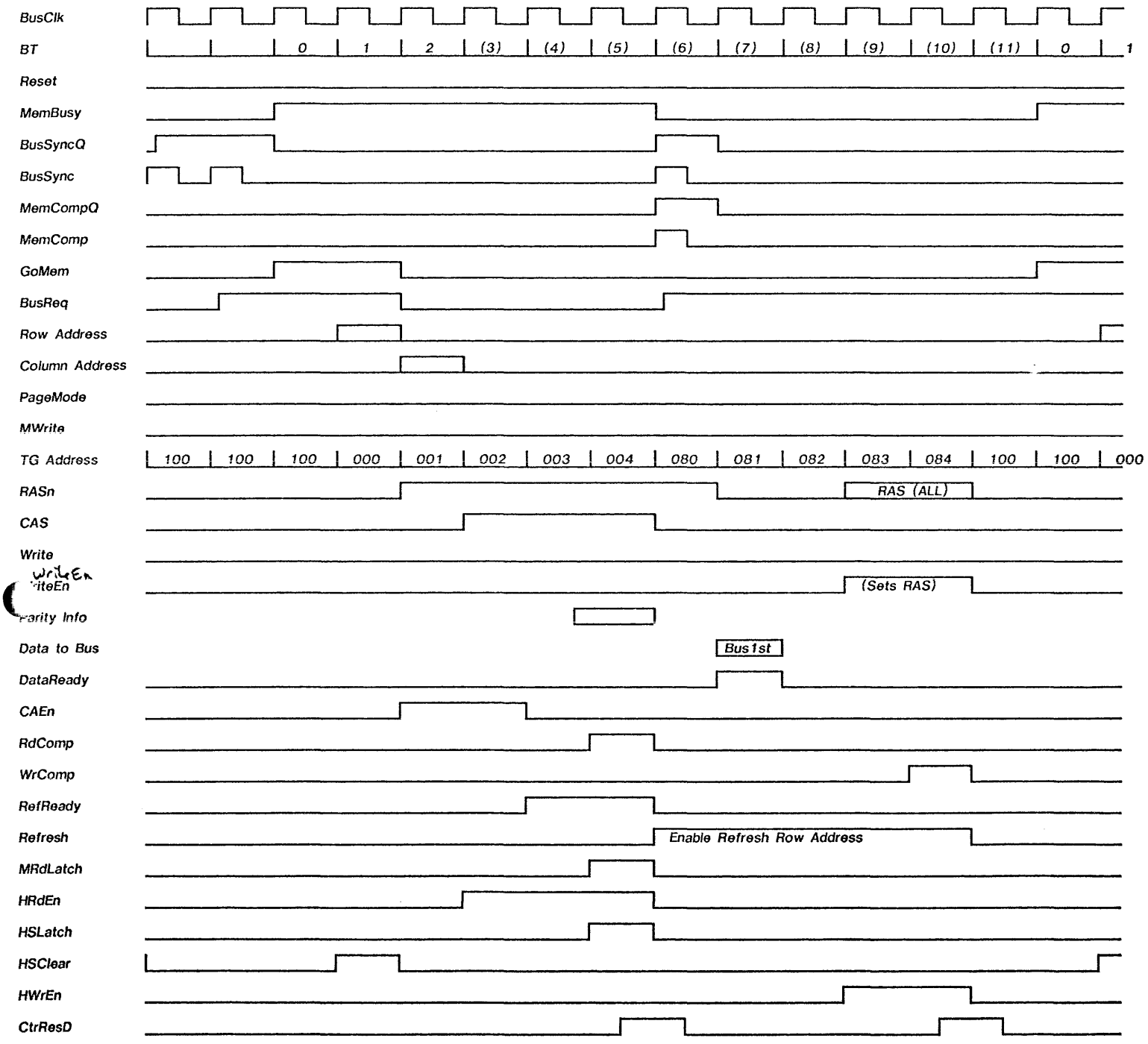
Memory Page Write Timing



[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt05.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 5 of 11

Memory Refresh Timing Following Read



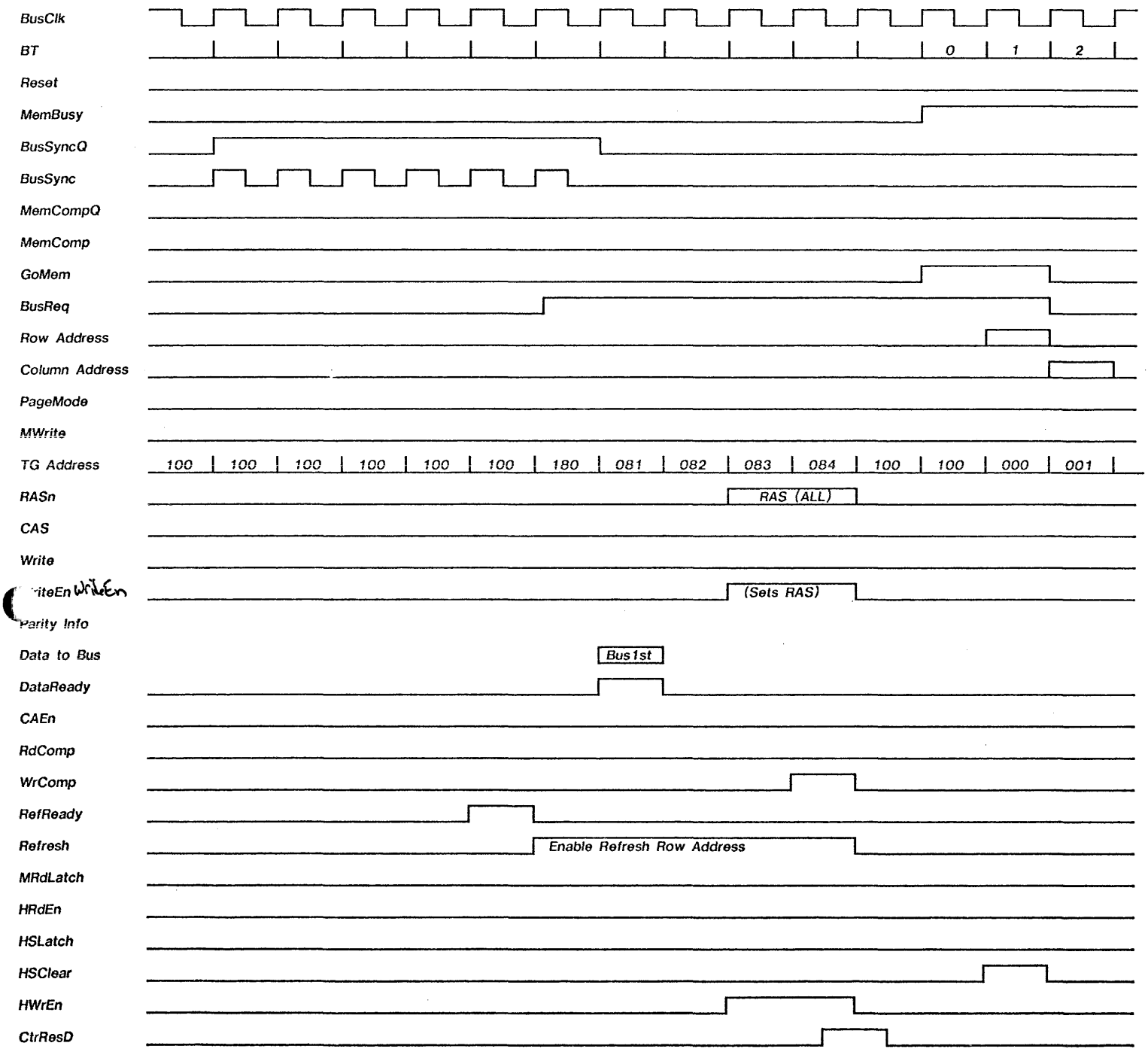
Notes: 1. Refresh is like lowest priority BusReq.

2. Refresh requires 5 clocks if following Write, so basic cycle is 5 clocks.

[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt06.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 6 of 11

Memory Refresh Timing when MemBusy'



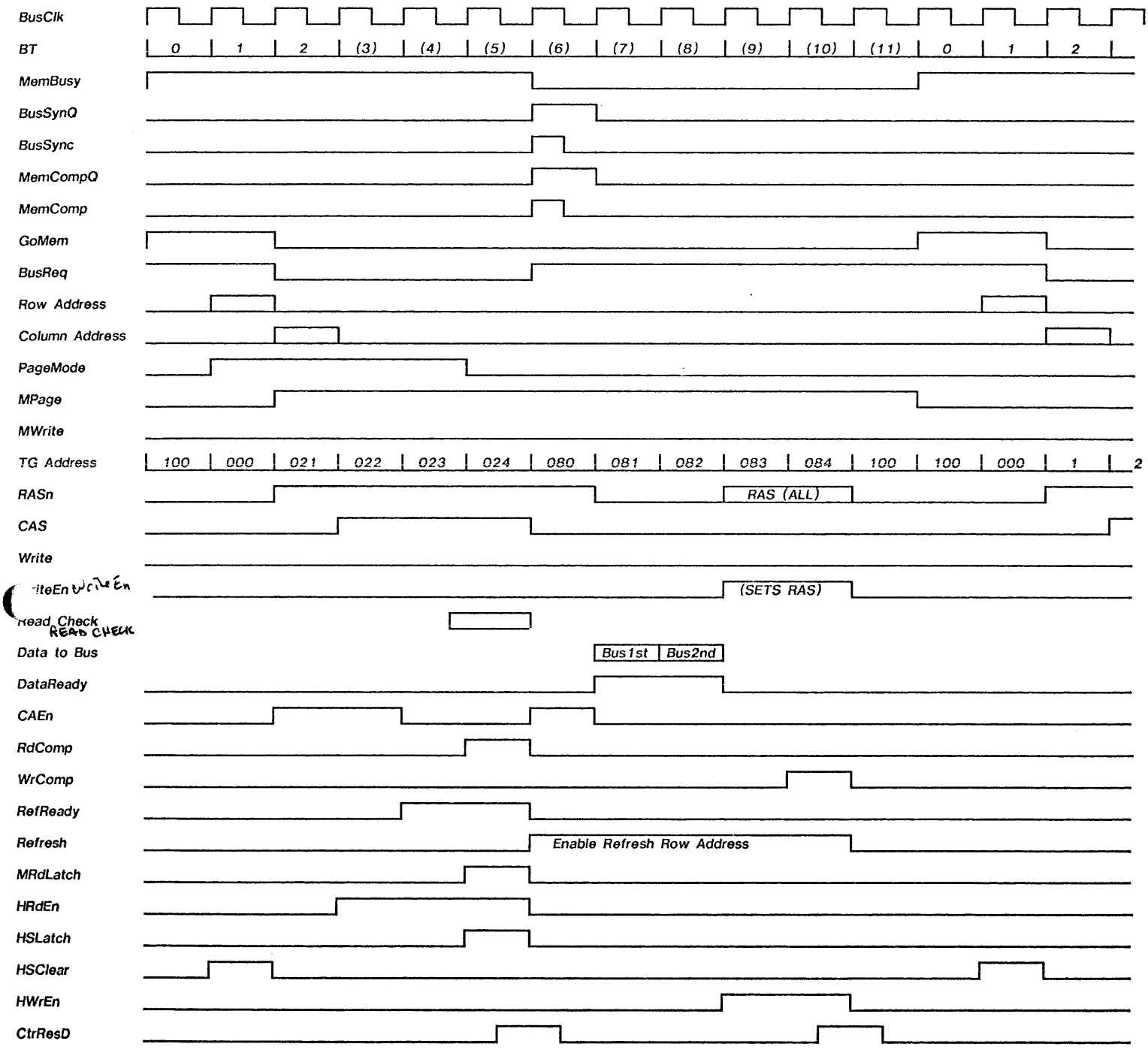
Note: If Refready precedes BusReq, Refresh occurs first. If RefReady sets at same time or after BusReq, Refresh occurs after BusReq's are serviced.

[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt07.sil	Designer: Sato	Rev: A	Date: 10-27-78	Page: 7 of 11

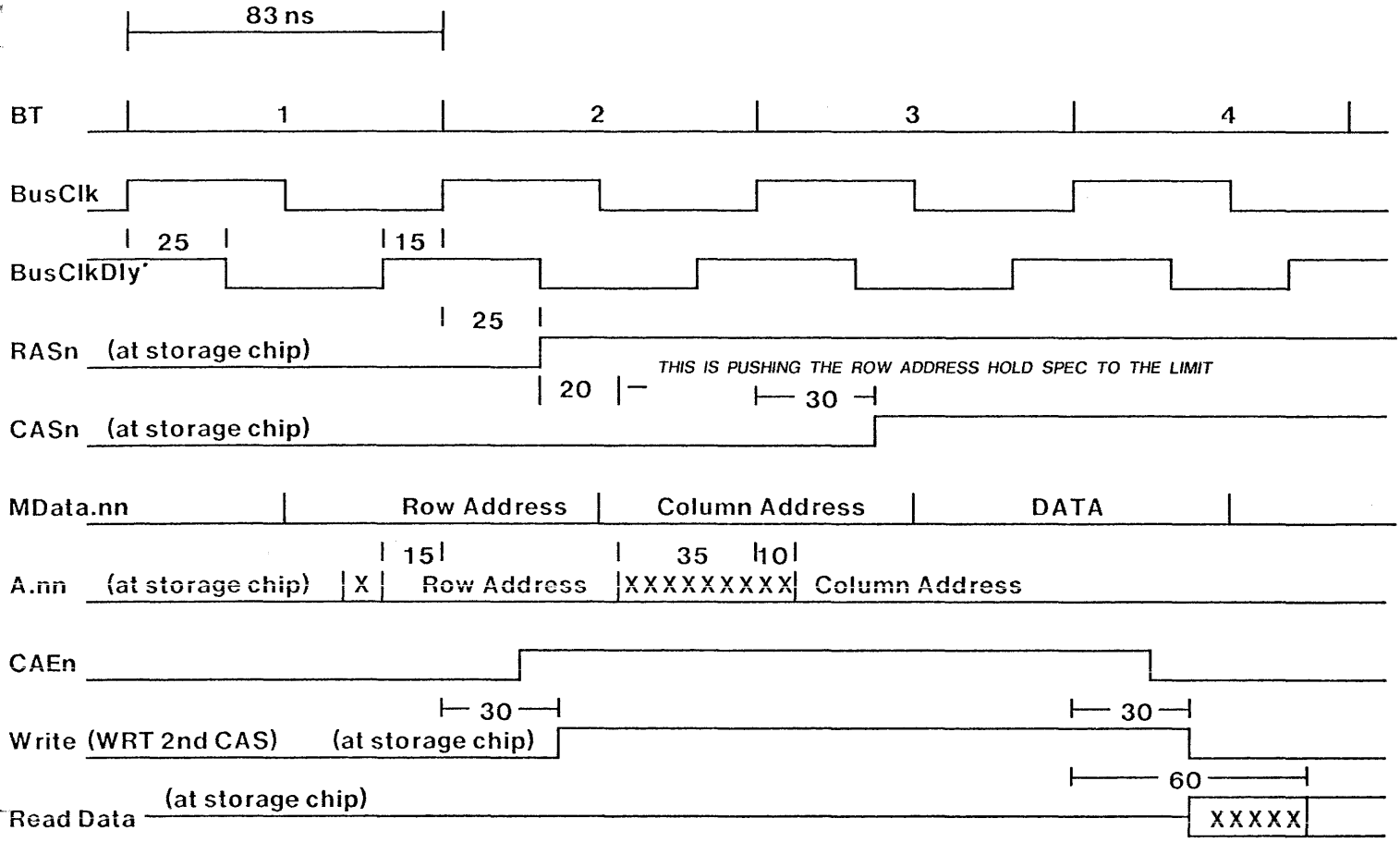
d

Memory Refresh Timing following Page Read



[ifs-2] < note > ntmt.dm

XEROX ASD	Project: Notetake	Title: Memory Address Timing	File: ntmt08.sil	Designer: Sato	Rev: A	Date: 10-10-78	Page: 8 of 11



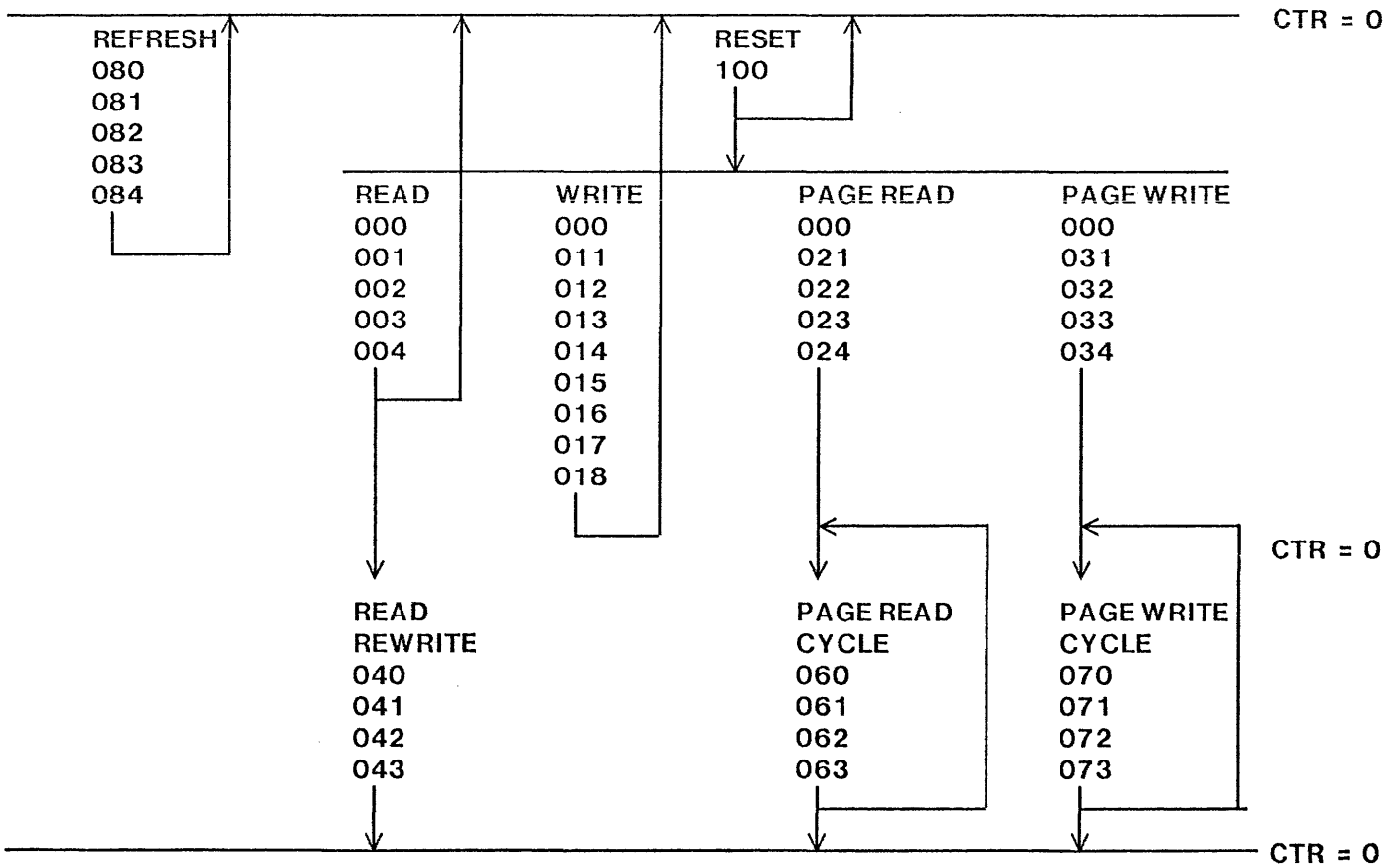
NOTE: BT TIMING IS IDEALIZED. REFERENCE TIMING POINT IS RISING EDGE OF BUSCLK.

Timing Generator Notes

1. The timing generator contains 2 512 x 8 PROMs which generate most of the memory control signals required.
2. A four bit (16 state) counter provides sequencing for each of the major timing cycles.
3. Since each timing cycle may contain up to 16 steps, there are nominally 32 sectors available for timing cycle implementation. The sectors actually used are as follows:

Reset	Refresh	Jump	MPage	MWrite	Sequence
0	0	0	0	0	Read Word or Byte or Status
0	0	0	0	1	Write Word or Byte or Interrupt/Boot
0	0	0	1	0	Page Read (first cycle)
0	0	0	1	1	Page Write (first cycle)
0	0	1	0	0	Rewrite Error Correction if PErr
0	0	1	0	1	N/A
0	0	1	1	0	Page Read (repeat cycles)
0	0	1	1	1	Page Write (repeat cycles)
0	1	X	X	X	Refresh
1	X	X	X	X	Reset

4. TG address consists of 9 bits; MS bit is Clear, 2nd LS byte are conditions as in 3 above, and LS byte is a counter. The counter starts at zero for each sequence. TG address for Clear is 100 (hex). All cycles depending on bus information start at 000. TG addresses in Hex are as follows:



5. TGJumps are used as a convenient way to repeat cycles or jump to another sector. In PageMode, TGJumps are unconditional; while in Read mode, they are conditional on PErr.

Note: Bits are from left to right;

1 CASD
2 WrD
3 WrEnD
4 CtrResD
5 CAEnD
6 RdCompD
7 WrCompD
8 Bus1stD

MSB ADDRESS LSB	00000 READ	00001 WRITE	00010 PAGE READ (1ST)	00011 PAGE WRITE (1ST)
0000	00001000	00001000	00001000	00001000
0001	10001000	10001001	10001000	01100001
0010	10000000	10000000	10000000	01100000
0011	10000100	10000100	10000100	01110000
0100	00010001	00100000	00011001	00000000
0101	00000000	00100000	00000000	
0110		01101000		
0111		11101010		
1000		10010000		
1001		00000000		
1010				
1011				
1100				
1101				
1110				
1111	00000000	00000000	00000000	00000000

	00100 READ REWRITE	00101 N/A	00110 PAGE READ (2+)	00111 PAGE WRITE (2+)
0000	00100000	00000000	10001000	01101000
0001	01101000		10000000	11101001
0010	11101010		10000100	11100010
0011	10010000		00011001	01110000
0100	00000000		00000000	00000000
0101				
0110				
0111				
1000				
1001				
1010				
1011				
1100				
1101				
1110				
1111	00000000	00000000	00000000	00000000

	01XXX REFRESH			1XXXX RESET
0000	00000000	00000000	00000000	00000000
0001	00000000			
0010	00100000			
0011	00100010			
0100	00010000			
0101	00000000			
0110				
0111				
1000				
1001				
1010				
1011				
1100				
1101				
1110				
1111	00000000	00000000	00000000	00000000