ISA Bus Timing Diagrams

P/N 5001321 Revision A

COMPUTERS, INCORPORATED

4757 Hellyer Avenue, San Jose, CA 95138
Phone: 408 360-0200, FAX: 408 360-0222, Web: www.ampro.com

REVISION HISTORY

Revision	Reason for Change	Date
A	Initial Release	6/98

ISA Bus Timing Diagrams

Ampro's ISA bus timing diagrams are derived from diagrams in the IEEE P996 draft specification which were, in turn, derived from the timing of the original IBM AT computer.

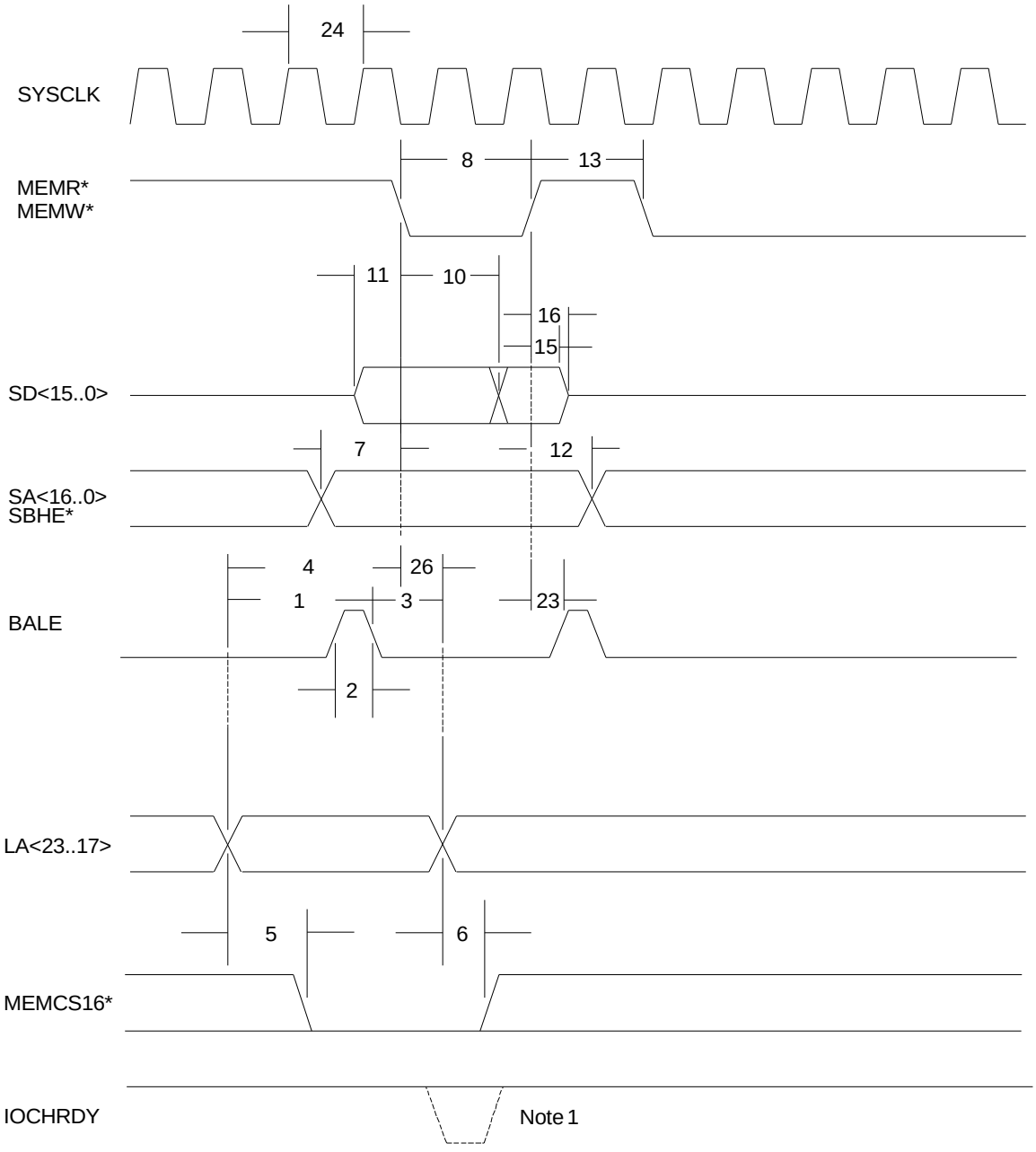
Please note that the IEEE P996 draft specification was never completed by the IEEE and is not an IEEE approved spec. Also, the "latest" IEEE draft is known to contain errors. In the absence of an approved IEEE specification, manufacturers of PC chip sets attempt to meet a "consensus" ISA bus standard. This has resulted in minor variations in signal interpretation and timing among the various PC chipset vendors. For this reason, Ampro recommends that designers of interfaces to the ISA bus use the minimum number of bus signals needed to perform a required function (e.g. chip selection or signal synchronization). For example, at least one popular chipset does not drive AEN high during REFRESH. In certain instances, Ampro has added logic to improve bus timing and/or signal relationships on CPU and peripheral boards.

Ampro's ISA bus timing diagrams include several corrections relative to the IEEE P996 draft specification. However, since these diagrams are derived from an uncompleted and unapproved IEEE specification, they may contain other errors.

For comprehensive technical details on the ISA architecture and bus, Ampro recommends the following book: *ISA & EISA Theory and Operation*, by Edward Solari; published by Annabooks (www.annabooks.com). This book contains a detailed technical exposition of the ISA and EISA buses and is written by the principal author of the IEEE P996 draft specification.

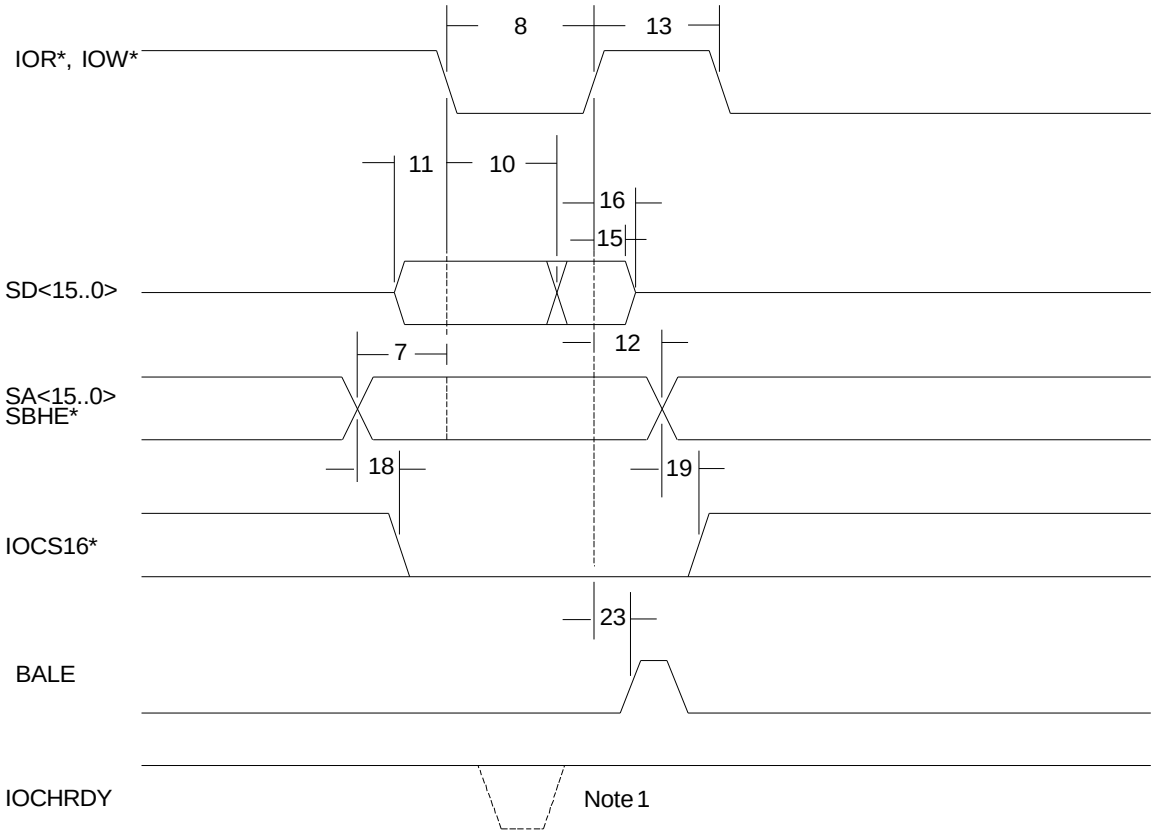
REF	TYPE	SIZE	DESCRIPTION	DRIVER		RECEIVER	
				MIN	MAX	MIN	MAX
1	M,IO	8/16	LA setup to BALE deasserted	111		100	
2	M,IO	8/16	BALE pulse width	61		50	
3	M,IO	8/16	LA hold from BALE deasserted	26		15	
4a	M	16	LA setup to MEMx* asserted	120		109	
4b	M	8	LA setup to MEMx* asserted	183		172	
5	M	8/16	MEMCS16* valid from LA		66		102
6	M	8/16	MEMCS16* hold from LA	0		0	
7a	M	16	SA, SBHE* setup to MEMx*	39		28	
7b	IO	16	SA, SBHE* setup to IOx*	102		91	
7c	M,IO	8	SA, SBHE* setup to IOx* or MEMx*	102		91	
8a	M	16	Command width	240		219	
8b	IO	16	Command width	165		154	
8c	M	16	Command width with ENDXFR* asserted	103		92	
8d	M,IO	8	Command width	541		530	
10a	M	16	Read data access		173		195
10b	IO	16	Read data access		110		132
10c	M,IO	16	Read data access with ENDXFR* asserted		48		70
10d	M,IO	8	Read data access		482		504
11a	M	16	Write data setup	-34		-45	
11b	IO	16	Write data setup	33		22	
11c	M,IO	8	Write data setup (even)	7		-4	
11d	M,IO	8	Write data setup (odd)	-45		-56	
12	M,IO	8/16	SA, SBHE* hold	53		42	
13a	M	16	Command deasserted	108		97	
13b	M	8	Command deasserted	170		159	
13c	IO	8/16	Command deasserted	170		159	
15a	M,IO	8/16	Read data hold	0		0	
15b	M,IO	8/16	Write data hold	25		25	
16	M,IO	8/16	Read command to SD disabled		30		30
17	M	16	ENDXFR* asserted from command		10		32
18	IO	8/16	IOCS16* asserted from SA		74		122
19	IO	8/16	IOCS16* hold from SA	0		0	
20a	M,IO	8/16	IOCHRDY valid from command asserted		70		159
20b	M,IO	8	IOCHRDY valid from command asserted		373		462
21	M,IO	8/16	IOCHRDY deasserted pulse width	125	15600	125	15611
22	M,IO	8/16	Command hold from IOCHRDY	125			
23	M,IO	8/16	BALE asserted from command deasserted	46		35	
24	M,IO	8/16	Clock period (Tclk)	120	167	120	167
25a	M,IO	8/16	Data setup to IOCHRDY deasserted (8-bit even)		85		74
25b	M,IO	8	Data setup to IOCHRDY deasserted (8-bit odd)		75		64
26a	M	16	LA hold to MEMx* active	41		30	
26b	M	8	LA hold to MEMx* active	-21		-32	
28	M	16	ENDXFR* setup to SYSCLK falling edge	22			
29	M	16	ENDXFR* hold from SYSCLK falling edge	22			
36	M	16	LA setup to ENDXFR* asserted		180		158
37	M	16	SA setup to ENDXFR* asserted		83		61

Table 1. Memory and I/O Timing



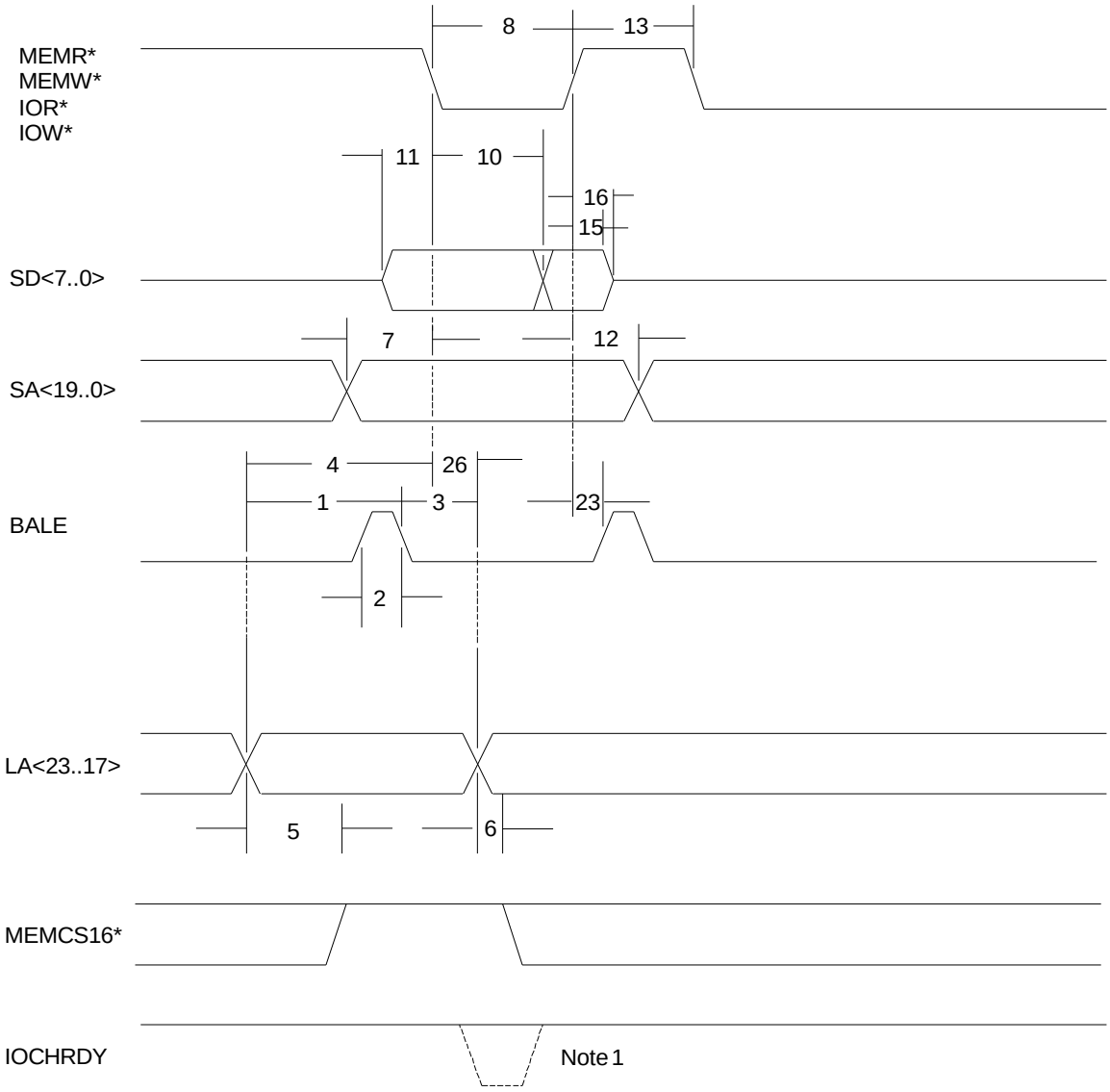
Note 1: IOCHRDY timings apply if deasserted. See Figure 4.

Figure 1. 16-bit Memory Timing



Note 1: IOCHRDY timings apply if deasserted. See Figure 4.

Figure 2. 16-bit I/O Timing



Note 1: IOCHRDY timings apply if deasserted. See Figure 4.

Figure 3. 8-bit Memory and I/O Timing

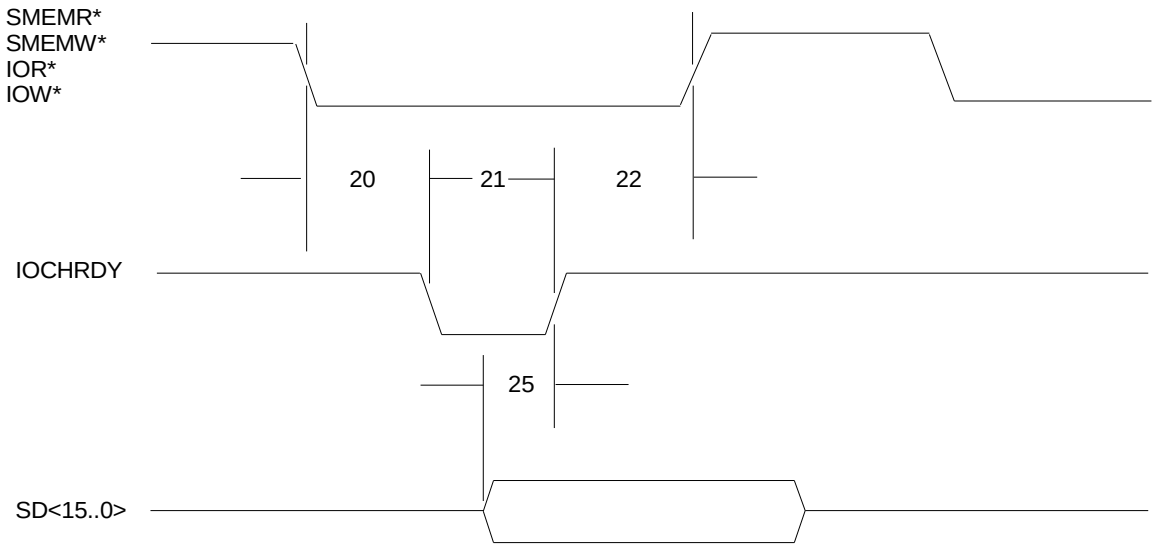
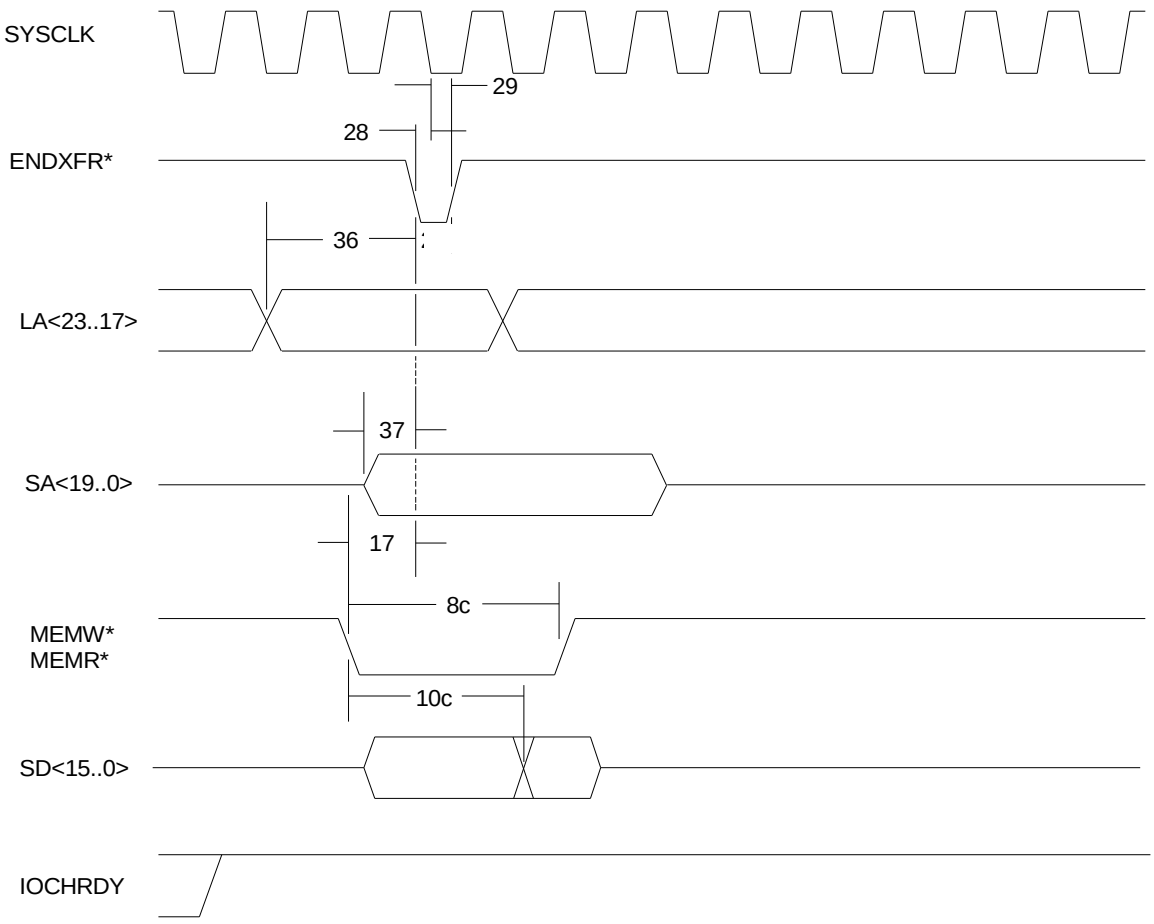


Figure 4. IOCHRDY Timing

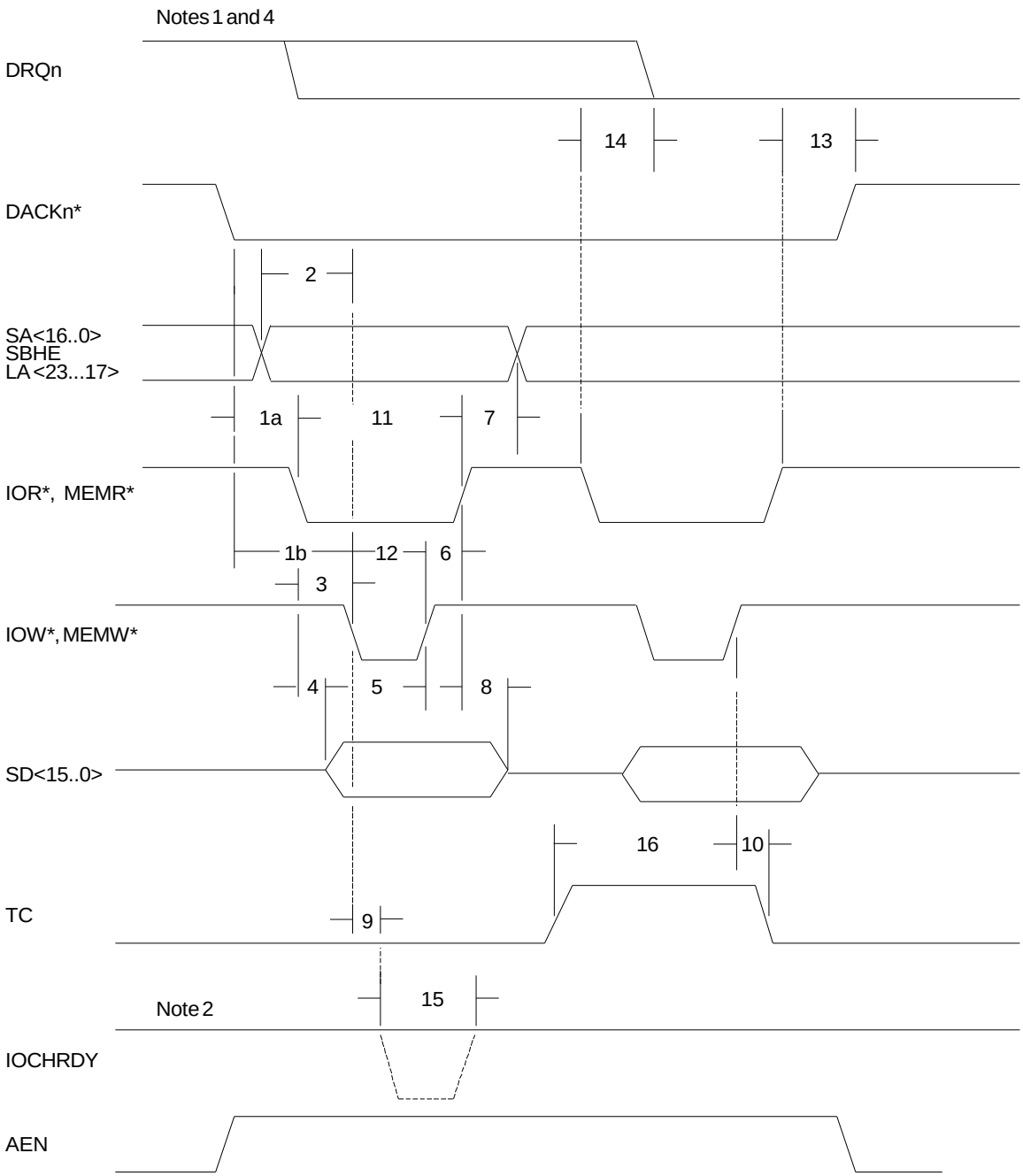


Note 1: Assertion of ENDXFR* within the maximum time from command is only required for a 16-bit cycle with zero wait states. Otherwise, ENDXFR* may be asserted at any time during the cycle while command is asserted.

Figure 5. ENDXFR Timing*

REF	DESCRIPTION	DRIVER		RECEIVER	
		MIN	MAX	MIN	MAX
1a 1b	DACKn*, AEN setup to IOR* DACKn*, AEN setup to IORW*	76 321		65 310	
2	Address setup to MEMW*, IOW*	102		91	
3a 3b	IOR* setup to MEMW* MEMR* setup to IOW*	246 0		234 0	
4a 4b 4c	Data access from IOR* 8/16bit Data access from MEMR* 16bit Data access from MEMR* 8bit		220 173 332		242 195 337
5	Data setup to IOW* unasserted	164		142	
6	Read command hold from write command	50		39	
7	SBHE*, address hold	53		42	
8	Data hold from read command	11		0	
9a 9b	IOCHRDY deasserted from 16bit memory command IOCHRDY deasserted from 8bit memory command		81 384		103 406
10	TC hold from command unasserted	60		49	
11a 11b	IOR* pulse width MEMR* pulse width	797 547		786 536	
12	IOW*, MEMW* width	500		489	
13a 13b 13c	DACKn* hold from IOW* DACKn* hold from IOW* AEN hold from command	114 173 41		103 162 30	
14	DREQ inactive from IOx*		119		141
15	IOCHRDY low width	Tclk	15600	Tclk	15611
16	TC setup to command unasserted	511		500	

Table 2. DMA Timing



Note 1: DRQn may be deasserted any time after DACKn* during a block mode DMA transfer.

Note 2: IOCHRDY may be deasserted to insert additional wait states. Additional bus wait states are added in units of two bus clocks.

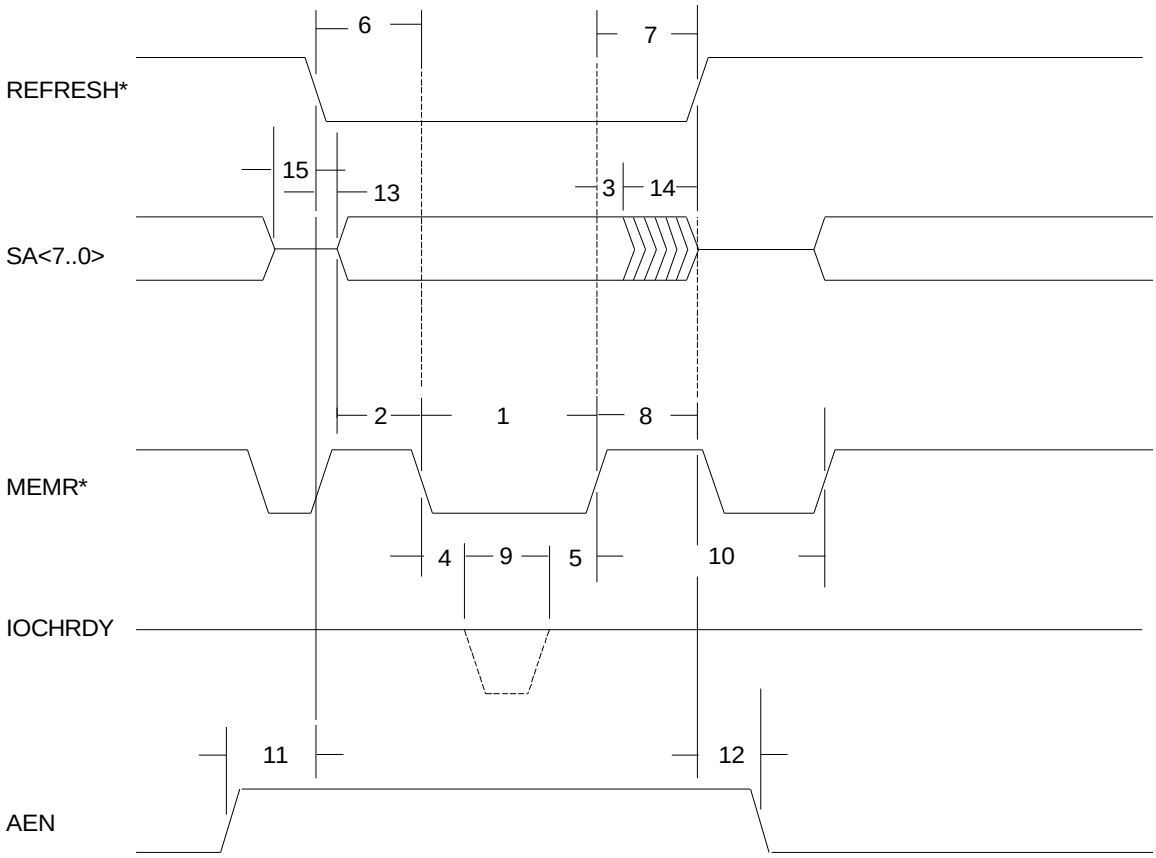
Note 3: The DMA controller activates TC during the last cycle of a DMA request.

Note 4: DMA transfers may be broken up into multiple back-to-back cycles where the DMA controller removes DACKn* and optionally releases the bus to allow higher priority cycles to occur. In this case, DACKn* will be temporarily deasserted even though DRQn is still asserted.

Figure 6. DMA Timing

REF	DESCRIPTION	DRIVER		RECEIVER	
		MIN	MAX	MIN	MAX
1	MEMR* pulse width	214		203	
2	SA<0...7> setup to MEMR*	81		70	
3	SA<0...7> hold from MEMR*	36		25	
4	IOCHRDY deasserted from MEMR*		81		159
5	MEMR* deasserted from IOCHRDY	125		125	
6	REFRESH* setup to MEMR*	125		114	
7	REFRESH* hold from MEMR* (Note 1)	31	250	20	239
8	SA<11...0> tri-state from MEMR* high		Tclk		
9	IOCHRDY width	Tclk		Tclk	
10	AM ownership delay (Note 2)	2*Tclk		2*Tclk	
11	AEN asserted to REFRESH* active	11		0	
12	AEN hold to REFRESH* inactive	11		0	
13	REFRESH* asserted to SA<0...7> valid	11		0	
14	REFRESH* hold from SA<0...7> valid	11		0	
15	Address and Control disabled to REFRESH* asserted	0		0	

Table 3. Refresh Timing



Note 1: The temporary master may exceed the maximum REFRESH* hold time in order to conduct another refresh operation.

Note 2: The temporary master, if the current master, must tri-state the address and command signals prior to driving REFRESH* high (1).

Figure 7. REFRESH Timing