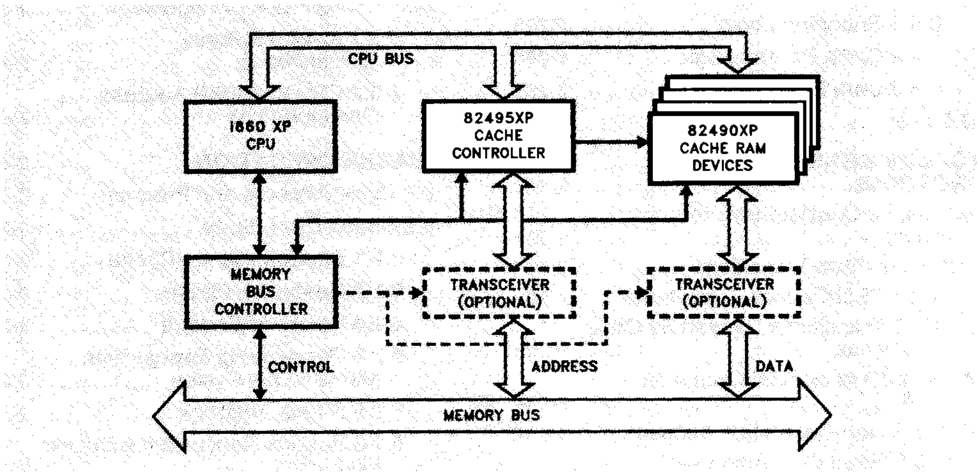
82495XP CACHE CONTROLLER/ 82490XP CACHE RAM

- **Two-Way, Set Associative, Secondary Cache for i860™ XP Microprocessor**
- **50 MHz "No Glue" Interface with CPU**
- **Configurable**
  - **—Cache Size 256 or 512 Kbytes**
  - **—Line Width 32, 64 or 128 Bytes**
  - **—Memory Bus Width 64 or 128 Bits**
- **Dual-Ported Structure Permits Simultaneous Operations on CPU and Memory Buses**
- **Efficient MRU Way Prediction**
  - **—Zero Wait States on MRU Hit**
  - **—One Wait State on MRU Miss**
- **Dynamically Selectable Update Policies**
  - **—Write-Through**
  - **—Write-Once**
  - **—Write-Back**
- **MESI Cache Consistency Protocol**
- **Hardware Cache Snooping**
- **Maintains Consistency with Primary Cache via Inclusion Principle**
- **Flexible User-Implemented Memory Interface Enables Wide Range of Product Differentiation**
  - **—Clocked or Strobed**
  - **—Synchronous or Asynchronous**
  - **—Pipelining**
  - **—Memory Bus Protocol**
- **82495XP Cache Controller Available in 208-Lead Ceramic Pin Grid Array Package**
- **82490XP Cache RAM Available in 84-Lead Plastic Quad Flatpack Package**
  (See *Packaging Handbook*, Order #240800)

The Intel 82495XP cache controller and 82490XP cache RAM, when coupled with a user-implemented memory bus controller, provide a second-level cache subsystem that eliminates the memory latency and bandwidth bottleneck for a wide range of multiprocessor systems based on the i860 XP microprocessor. The CPU interface is optimized to serve the i860 XP microprocessor with zero wait states at up to 50 MHz. A secondary cache built from the 82495XP and 82490XP isolates the CPU from the memory subsystem; the memory can run slower and follow a different protocol than the i860 XP microprocessor.



240956-60

Figure 0-1. Secondary Cache Configuration

Intel, intel, and i860 are trademarks of Intel Corporation.

December 1992
Order Number: 240956-002

82495XP Cache Controller/82490XP Cache RAM

CONTENTS

CONTENTS

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

FIGURES

CONTENTS

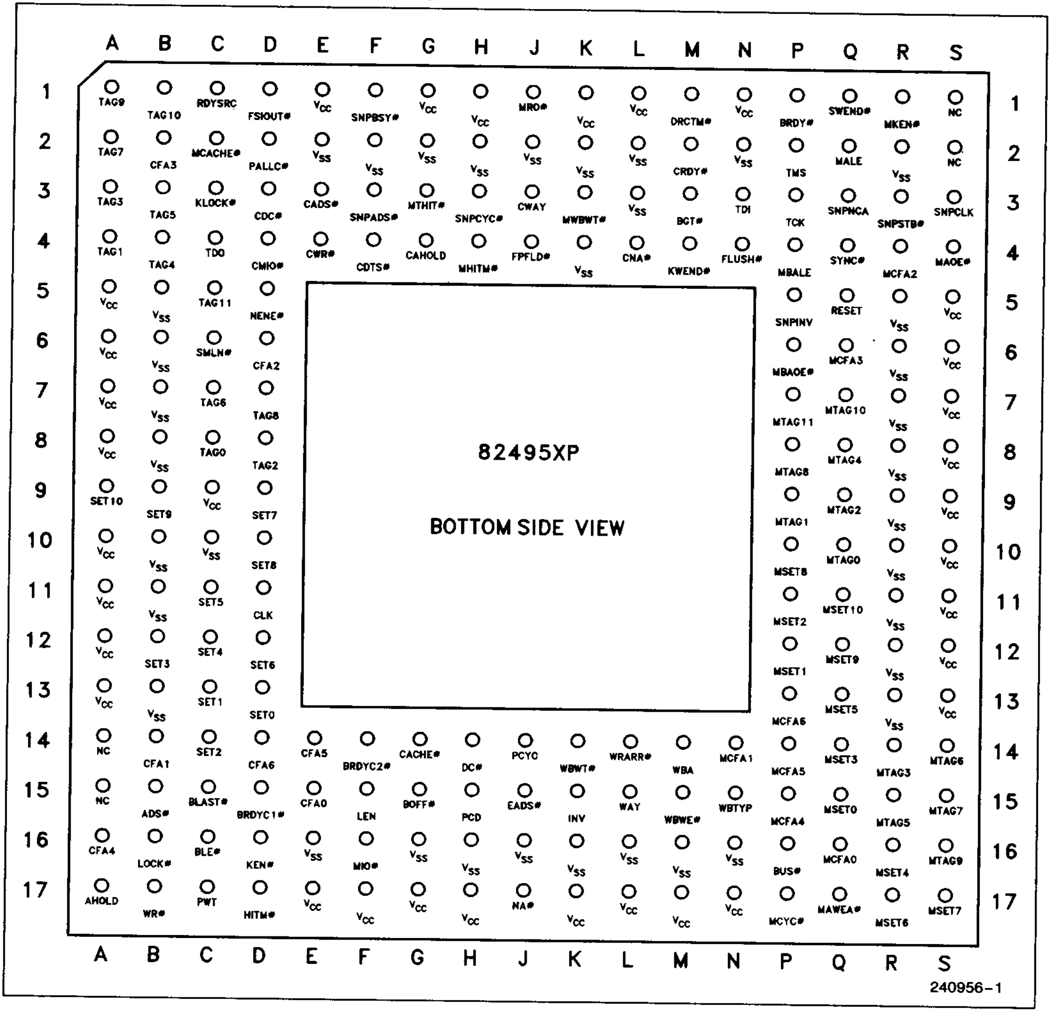
CONTENTS

CONTENTS PAGE

TABLES

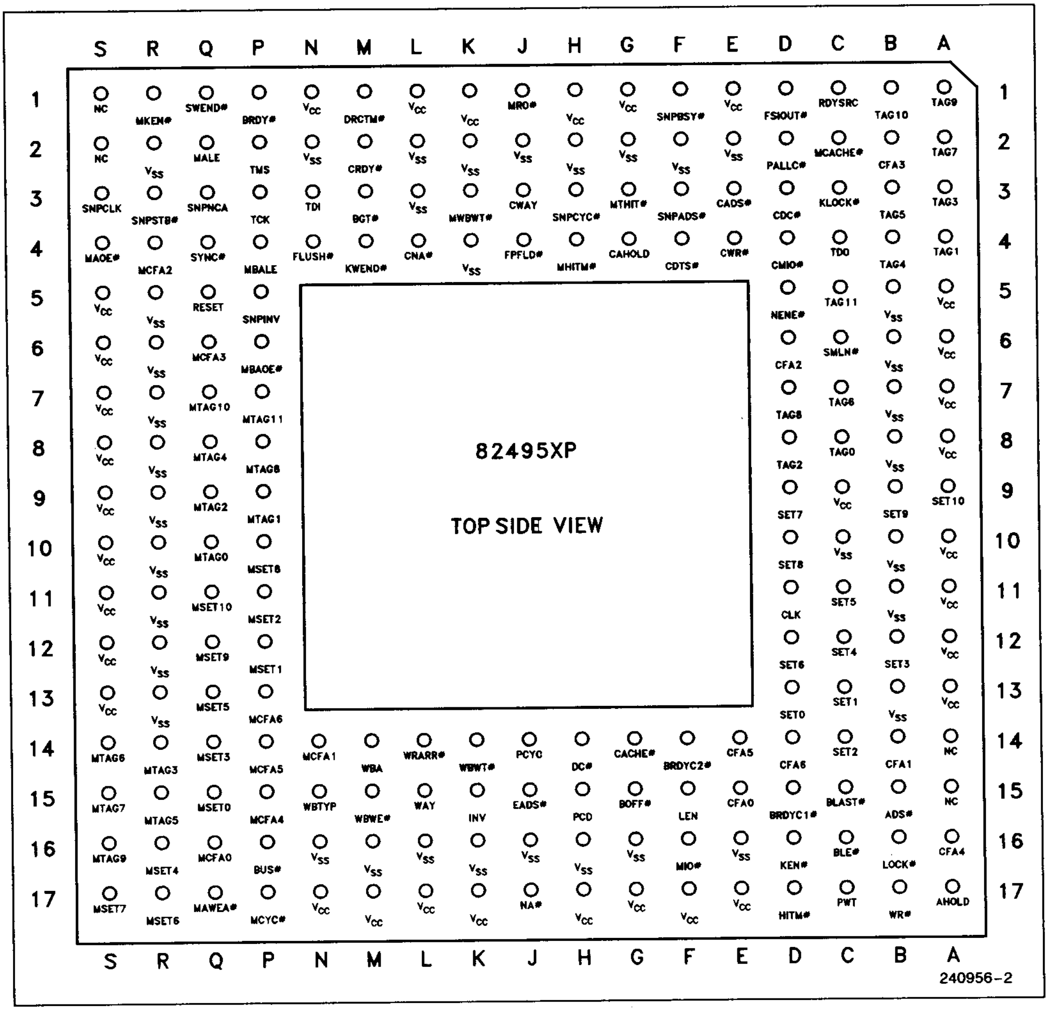
CONTENTS PAGE

1.0 82495XP/82490XP PINOUTS



	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S
1	TAG9	TAG10	RDYSRC	FSIOUT#	V_{CC}	SNPBSY#	V_{CC}	V_{CC}	MRO#	V_{CC}	V_{CC}	DRCTM#	V_{CC}	BRDY#	SWEND#	MKEN#	NC
2	TAG7	CFA3	MCACHE#	PALLC#	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	CRDY#	V_{SS}	TMS	MALE	V_{SS}	NC
3	TAG3	TAG5	KLOCK#	CDC#	CADS#	SNPADS#	MTHIT#	SNPCYC#	CWAY	MWBWT#	V_{SS}	BGT#	TDI	TCK	SNPNCA	SNPSTB#	SNPCLK
4	TAG1	TAG4	TDO	CMIO#	CWR#	CDTS#	CAHOLD	MHITM#	FPFLD#	V_{SS}	CNA#	KWEND#	FLUSH#	MBALE	SYNC#	MCFA2	MAOE#
5	V_{CC}	V_{SS}	TAG11	NENE#										SNPINV	RESET	V_{SS}	V_{CC}
6	V_{CC}	V_{SS}	SMLN#	CFA2										MBAOE#	MCFA3	V_{SS}	V_{CC}
7	V_{CC}	V_{SS}	TAG6	TAG8										MTAG11	MTAG10	V_{SS}	V_{CC}
8	V_{CC}	V_{SS}	TAG0	TAG2										MTAG8	MTAG4	V_{SS}	V_{CC}
9	SET10	SET9	V_{CC}	SET7										MTAG1	MTAG2	V_{SS}	V_{CC}
10	V_{CC}	V_{SS}	V_{SS}	SET8										MSET8	MTAG0	V_{SS}	V_{CC}
11	V_{CC}	V_{SS}	SET5	CLK										MSET2	MSET10	V_{SS}	V_{CC}
12	V_{CC}	SET3	SET4	SET6										MSET1	MSET9	V_{SS}	V_{CC}
13	V_{CC}	V_{SS}	SET1	SET0										MCFA6	MSET5	V_{SS}	V_{CC}
14	NC	CFA1	SET2	CFA6	CFA5	BRDYC2#	CACHE#	DC#	PCYC	WBWT#	WRARR#	WBA	MCFA1	MCFA5	MSET3	MTAG3	MTAG6
15	NC	ADS#	BLAST#	BRDYC1#	CFA0	LEN	BOFF#	PCD	EADS#	INV	WAY	WBWE#	WBTYP	MCFA4	MSET0	MTAG5	MTAG7
16	CFA4	LOCK#	BLE#	KEN#	V_{SS}	MIO#	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	BUS#	MCFA0	MSET4	MTAG9
17	AHOLD	WR#	PWT	HITM#	V_{CC}	V_{CC}	V_{CC}	V_{CC}	NA#	V_{CC}	V_{CC}	V_{CC}	V_{CC}	MCYC#	MAWEA#	MSET6	MSET7

Figure 1-1. 82495XP Pinout (Bottom View)



	S	R	Q	P	N	M	L	K	J	H	G	F	E	D	C	B	A
1	NC	MKEN#	SWEND#	BRDY#	V_{CC}	DRCTM#	V_{CC}	V_{CC}	MRO#	V_{CC}	V_{CC}	SNPBSY#	V_{CC}	FSIOUT#	RDYSRC	TAG10	TAG9
2	NC	V_{SS}	MALE	TMS	V_{SS}	CRDY#	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	PALLC#	MCACHE#	CFA3	TAG7
3	SNPCLK	SNPSTB#	SNPNCA	TCK	TDI	BGT#	V_{SS}	MWBWT#	CWAY	SNPCYC#	MTHIT#	SNPADS#	CADS#	CDC#	KLOCK#	TAG5	TAG3
4	MAOE#	MCFA2	SYNC#	MBALE	FLUSH#	KWEND#	CNA#	V_{SS}	FPFLD#	MHITM#	CAHOLD	CDTS#	CWR#	CMIO#	TDO	TAG4	TAG1
5	V_{CC}	V_{SS}	RESET	SNPINV										NENE#	TAG11	V_{SS}	V_{CC}
6	V_{CC}	V_{SS}	MCFA3	MBAOE#										CFA2	SMLN#	V_{SS}	V_{CC}
7	V_{CC}	V_{SS}	MTAG10	MTAG11										TAG8	TAG6	V_{SS}	V_{CC}
8	V_{CC}	V_{SS}	MTAG4	MTAG8										TAG2	TAG0	V_{SS}	V_{CC}
9	V_{CC}	V_{SS}	MTAG2	MTAG1										SET7	V_{CC}	SET9	SET10
10	V_{CC}	V_{SS}	MTAG0	MSET8										SET8	V_{SS}	V_{SS}	V_{CC}
11	V_{CC}	V_{SS}	MSET10	MSET2										CLK	SET5	V_{SS}	V_{CC}
12	V_{CC}	V_{SS}	MSET9	MSET1										SET6	SET4	SET3	V_{CC}
13	V_{CC}	V_{SS}	MSET5	MCFA6										SET0	SET1	V_{SS}	V_{CC}
14	MTAG6	MTAG3	MSET3	MCFA5	MCFA1	WBA	WRARR#	WBWT#	PCYC	DC#	CACHE#	BRDYC2#	CFA5	CFA6	SET2	CFA1	NC
15	MTAG7	MTAG5	MSET0	MCFA4	WBTYP	WBWE#	WAY	INV	EADS#	PCD	BOFF#	LEN	CFA0	BRDYC1#	BLAST#	ADS#	NC
16	MTAG9	MSET4	MCFA0	BUS#	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	MIO#	V_{SS}	KEN#	BLE#	LOCK#	CFA4
17	MSET7	MSET6	MAWEA#	MCYC#	V_{CC}	V_{CC}	V_{CC}	V_{CC}	NA#	V_{CC}	V_{CC}	V_{CC}	V_{CC}	HITM#	PWT	WR#	AHOLD

Figure 1-2. 82495XP Pinout (Top View)

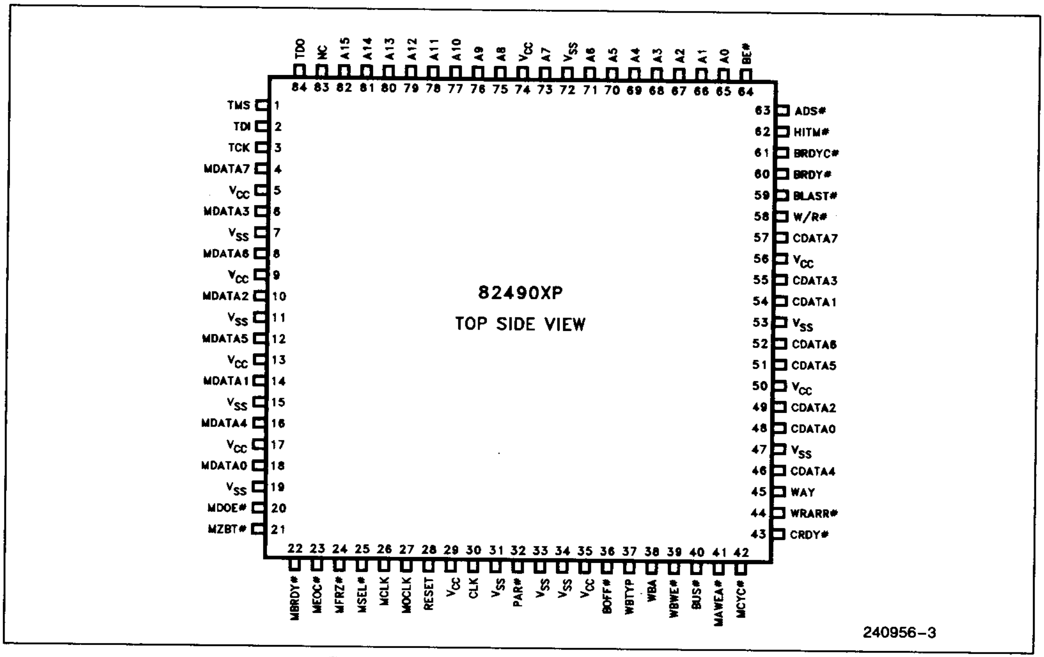


Figure 1-3. 82490XP Pinout (Top View)

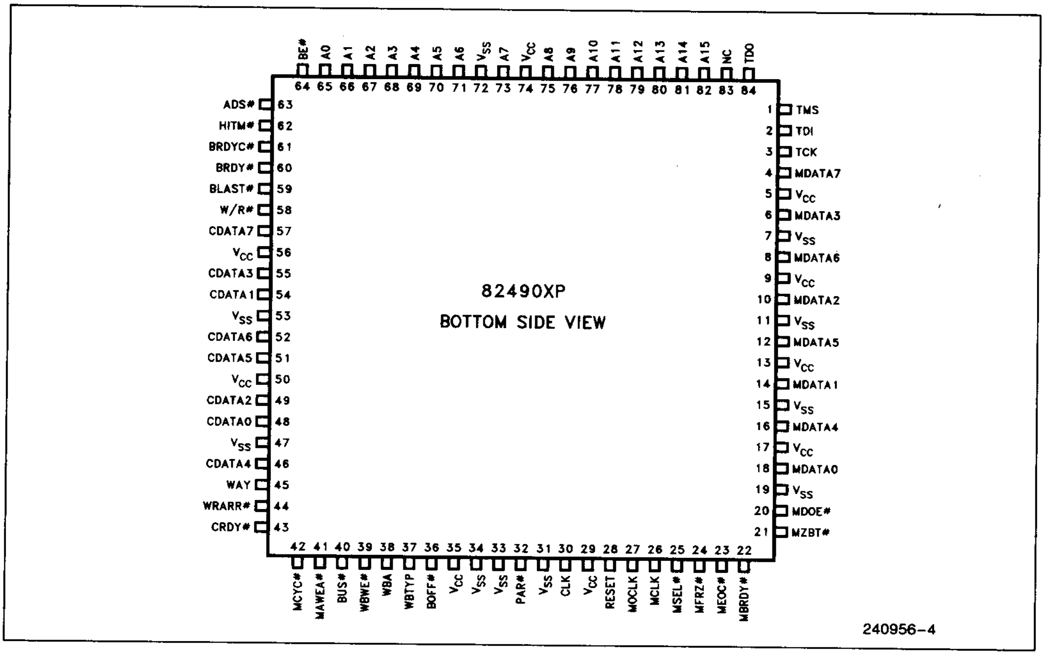


Figure 1-4. 82490XP Pinout (Bottom View)

1.1 Pin Cross Reference Tables

Table 1-1. 82495XP Pin Cross Reference by Name

Signal	Location	Signal	Location	Signal	Location
ADS#	B15	AHOLD	A17	BGT#	M03
BLAST#	C15	BLE#	C16	BOFF#[CLEN0]	G15
BRDY#	P01	BRDYC1#	D15	BRDYC2#	F14
BUS#	P16	CACHE#	G14	CADS#	E03
CAHOLD	G04	CDC#	D03	CDTS#	F04
CFA0	E15	CFA1	B14	CFA2	D06
CFA3	B02	CFA4	A16	CFA5	E14
CFA6	D14	CLK	D11	CMIO#	D04
CNA#[CFG0]	L04	CRDY#[SLFTST#]	M02	CWAY	J03
CWR#	E04	DC#	H14	DRCTM#	M01
EADS#	J15	EWBE#	S02	FLUSH#[NCPFLD#]	N04
FPFLD#[FPFLDEN]	J04	FSIOUT#	D01	HITM#	D17
INV[CLEN1]	K15	KEN#	D16	KLOCK#	C03
KWEND#[CFG2]	M04	LEN	F15	LOCK#	B16
MALE[WWOR#]	Q02	MAOE#	S04	MAWEA#	Q17
MBALE[HIGHZ#]	P04	MBAOE#	P06	MCACHE#	C02
MCFA0	Q16	MCFA1	N14	MCFA2	R04
MCFA3	Q06	MCFA4	P15	MCFA5	P14
MCFA6	P13	MCYC#	P17	MHITM#	H04
MIO#	F16	MKEN#	R01	MRO#	J01
MSET0	Q15	MSET1	P12	MSET10	Q11
MSET2	P11	MSET3	Q14	MSET4	R16
MSET5	Q13	MSET6	R17	MSET7	S17
MSET8	P10	MSET9	Q12	MTAG0	Q10
MTAG1	P09	MTAG10	Q07	MTAG11	P07
MTAG2	Q09	MTAG3	R14	MTAG4	Q08
MTAG5	R15	MTAG6	S14	MTAG7	S15
MTAG8	P08	MTAG9	S16	MTHIT#	G03
MWBWT#	K03	NA#	J17	NENE#	D05
PALLC#	D02	PCD	H15	PCYC	J14
PWT	C17	RDYSRC	C01	RESET	Q05
SET0	D13	SET1	C13	SET10	A09
SET2	C14	SET3	B12	SET4	C12
SET5	C11	SET6	D12	SET7	D09
SET8	D10	SET9	B09	SMLN#	C06
SNPADS#	F03	SNPBSY#	F01	SNPCLK[SNPMD]	S03
SNPCYC#	H03	SNPINV	P05	SNPNCA	Q03

Table 1-1. 82495XP Pin Cross Reference by Name (Continued)

Signal	Location	Signal	Location	Signal	Location
SNPSTB#	R03	SWEND# [CFG1]	Q01	SYNC# [MEMLDRV]	Q04
TAG0	C08	TAG1	A04	TAG10	B01
TAG11	C05	TAG2	D08	TAG3	A03
TAG4	B04	TAG5	B03	TAG6	C07
TAG7	A02	TAG8	D07	TAG9	A01
TCK	P03	TDI	N03	TDO	C04
TMS	P02	WAY	L15	WBA	M14
WBTYP	N15	WBWE#	M15	WBWT# [WRMRST]	K14
WR#	B17	WRARR#	L14		
NC A14, A15, S01, S02		Vcc A05–A08, A10–A13, E01, E17, H01, H17, K01, K17, L01, L17, C09, N17, F17, G01, G17, M17, N01, S05–S13		Vss B05–B08, B10–B11, B13, E02, E16, F02, H02, H16, J02, J16, K02, K04, K16, L02–L03, L16, C10, N16, G02, G16, R02, R05–R10, M16, N02, R11–R13	

Table 1-2. 82490XP Pin Cross Reference by Name

Signal	Location	Signal	Location	Signal	Location
A0	65	A1	66	A10	77
A11	78	A12	79	A13	80
A14	81	A15	82	A2	67
A3	68	A4	69	A5	70
A6	71	A7	73	A8	75
A9	76	ADS#	63	BE#	64
BLAST#	59	BOFF#	36	BRDY#	60
BRDYC#	61	BUS#	40	CDATA0	48
CDATA1	54	CDATA2	49	CDATA3	55
CDATA5	51	CDATA6	52	CDATA7	57
CDATA4	46	CLK	30	CRDY#	43
HITM#	62	MAWEA#	41	MBRDY# [MISTB]	22
MCLK[MSTBM]	26	MCYC#	42	MDATA0	18
MDATA1	14	MDATA2	10	MDATA3	6
MDATA4	16	MDATA5	12	MDATA6	8
MDATA7	4	MDOE#	20	MEOC#	23
MFRZ# [MEMLDRV]	24	MOCLK[MOSTB]	27	MSEL# [MTR4/MTR8#]	25
MZBT# [MX4/MX8#]	21	PAR#	32	RESET	28
TCK	3	TDI	2	TDO	84
TMS	1	WAY	45	WBA	38
WBTYP	37	WBWE#	39	WR#	58
WRARR#	44	Vcc 5, 9, 13, 17, 29, 35, 50, 56, 74		Vss 7, 11, 15, 19, 31, 33, 34, 47, 53, 72	
NC	83				

1.2 Quick Pin Reference

BGT# [C490LDRV]	I	Bus Guaranteed Transfer, [82490XP Low Drive] This signal is generated by the MBC to the 82495XP. It indicates to the 82495XP a commitment by the MBC to complete the cycle on the memory bus. Until BGT# activation the 82495XP owns the cycle and will abort it if intervening snoops happen. After BGT# the cycle is owned by the MBC until its completion. From BGT# until SWEND# snoops will be accepted, but none will be processed until SWEND# activation. During RESET's falling edge, this signal controls the driver's strength of the 82495XP to 82490XP interface signals. This strength is a function of the cache size, and therefore the number of 82490XP's. Refer to the layout specifications section for more details.
BLE#	O	BE Latch Enable The BLE# signal is used to control the enable line of an external '377-type latch. The latch captures the i860 XP CPU's BE (Byte Enable) signals and other CPU provided cycle attributes which do not go through the 82495XP.
BRDY#	I	82495XP Burst Ready This is the burst ready indication from the memory bus controller. The MBC should connect its burst ready indication to the CPU BRDY#, the 82495XP BRDY# and the 82490XP BRDY#. In the CPU, it provides the same function as that described in the CPU data sheet. The 82495XP will only use this indication for burst tracking purposes. In the 82490XP, it increments the CPU latch burst counter.
CADS#	O	Cache Address Strobe This signal is generated by the 82495XP and used by the memory bus controller. Its assertion requests execution of a memory bus cycle by the memory bus controller. This signal when active indicates that the cache cycle control and attribute signals are valid.
CAHOLD	O	82495XP AHOLD This signal is generated by the 82495XP to track the CPU AHOLD signal when used for warm-reset and LOCKed sequences. It also provides information about CPU and cache BIST.
CD/C#	O	Cache Data/Control This is a cycle definition signal driven by the 82495XP. It indicates the type of memory bus cycle requested. This signal is valid with CADS# and can be pipelined by the memory bus controller.
CDTS#	O	Cache Data Strobe This signal is driven by the 82495XP to the memory bus controller. CDTS# for read cycles indicates that in the next CLK the memory bus controller can generate the first BRDY# for the read cycle. For write cycles it indicates when data is available on the memory bus. Usage of this signal allows complete independency between address strobes (CADS#, SNPADS#) and data strobe.
CFG0-2	I	Cache Configuration bits 0–2 These signals are inputs to the 82495XP. CFG0-2 allow the 82495XP to be configured to 5 different modes. Different modes indicate 82495XP/CPU line ratio, tag size (4K/8K), lines per sector.

1.2 Quick Pin Reference (Continued)

CLK	I	Clock This signal provides the fundamental timing for the 82495XP, 82490XP and CPU. It must be provided to the 82495XP, 82490XPs, CPU and memory bus controller components with minimal skew.
CM/IO#	O	Cache Memory/IO This signal is driven by the 82495XP and is a cycle definition signal. It indicates the type of memory bus cycle requested. This signal is valid with CADS# and can be pipelined by the memory bus controller.
CNA# [CFG0]	I	82495XP Next Address Enable, [Configuration Pin 0] This signal is driven by the memory bus controller and supplied to the 82495XP. It is used by the memory bus controller to dynamically pipeline CADS# cycles. During RESET falling edge it functions as the 82495XP CFG0 input.
CRDY# [SLFTST#]	I	Cache Memory Bus Ready, [82495XP Self Test] This signal is generated by the memory bus controller and informs the 82495XP and 82490XP that a memory bus cycle has been completed. CRDY# activation ends the memory bus cycle. During RESET's falling edge, if this signal is sampled low(active) and MBALE is sampled high(active), 82495XP self test will be invoked.
CWAY	O	Cache Way CWAY is driven by the 82495XP and is a cycle definition signal that indicates to the memory bus controller the WAY to be used by the requested cycle. On line-fills it indicates the way the line will be loaded. For write-backs it indicates the WAY that was written-back. This signal is valid with CADS#.
CW/R#	O	Cache Write/Read This signal is driven by the 82495XP and is a 82495XP cycle definition signal. It indicates the type of memory bus cycle requested. This signal is valid with CADS# and can be pipelined by the memory bus controller.
DRCTM#	I	Memory Bus Direct to [M] State This signal is an input to the 82495XP. It is the mechanism by which the memory bus can dynamically inform the 82495XP of a request to skip the [E] state and move the line directly to the [M] state. This signal is sampled by the 82495XP when SWEND# is asserted.
FLUSH# [NCPFLD#]	I	Flush the 82495XP cache, [Enable Non-Cacheable PFLD] This signal is an input to the 82495XP. Flush when active will cause the 82495XP to write-back all of its modified lines into main memory then invalidate all tag locations. At the end of a flush operation the 82495XP tag array will be completely invalidated. During RESET activation, this pin functions as the NCPFLD# configuration signal which, with FPFLDEN, selects one of three modes for handling i860 XP CPU floating point load cycles.

1.2 Quick Pin Reference (Continued)

FPFLD# [FPFLDEN]	I/O	FIFO PFLD Enable [PFLD Mode Select] During RESET, FPFLDEN and NCPFLDEN# inputs select one of three modes to handle i860 XP CPU pipelined floating point load cycles. In the mode which supports an external FIFO, the FPFLD# output indicates a PFLD cycle to be loaded into the FIFO.
FSIOUT#	O	Flush/Sync/Initialization Output This signal is an output of the 82495XP and indicates the start and end of three operations: Flush, Sync, and Initialization. The output is activated when the operation internally begins and is de-activated when the operation ends.
KLOCK#	O	82495XP LOCK# This signal is driven by the 82495XP and indicates to the memory bus controller a request to execute atomic read-modify-write sequences. KLOCK# is active with the CADS# of the first LOCKed operation and remains active until at least the clock following CADS# of the last cycle of LOCKed operation.
KWEND# [CFG2]	I	Cacheability Window End, [Configuration Pin 2] This signal is generated by the MBC and indicates to the 82495XP that the Cacheability Window has expired. At this point the 82495XP will latch the memory cacheability signal (MKEN#) and make decisions based on the cacheability attribute. MRO# which indicates the Read-Only cycle attribute is also sampled at this point. During RESET's falling edge this line functions as the CFG2 configuration signal which is used to configure the 82495XP/82490XP with cache parameters.
MALE[WWOR#]	I	Memory Bus, Address Latch Enable[Weak Write Ordering] This signal is generated by the memory bus controller, and controls a 82495XP internal transparent address latch (373 like). CADS# will generate a new address at the input of the internal address latch. MALE activation(high) will allow the flowing of this address to the memory bus provided MAOE# is active. When MALE inactive(low), the address at the latch input is latched. WWOR# configures the 82495XP into strong or weak write-ordering mode.
MAOE#	I	Memory Bus Address Output Enable This signal is generated by the memory bus controller and controls the 82495XP's output buffer of the memory bus address latches. The 82495XP drives the memory bus address lines if MAOE# is active (low). Otherwise, it is tristated. MAOE# also serves as a qualifier for snooping cycles: when inactive snoops will be enabled.
MBALE[HIGHZ#]	I	Memory Bus, 82495XP sub-line-address Latch Enable[High Impedance Output] This signal has an exact function as MALE but controls only the 82495XP sub-line addresses. This signal is generated by the memory bus controller, and controls a 82495XP internal transparent address latch (373 like). CADS# will generate a new address at the input of the internal address latch. MBALE activation(high) will allow the flowing of the sub-line address to the memory bus provided MBAOE# is active. When MALE inactive(low), the sub-line address at the latch input is latched. HIGHZ#, if active along with SLFTST#, causes the 82495XP to float all of its outputs.

1.2 Quick Pin Reference (Continued)

MBAOE#	I	Memory Bus, 82495XP sub-line Address Output Enable This signal has a similar function than MAOE#, but controls only the 82495XP sub-line addresses. If MBAOE# is active(low), the 82495XP will drive the sub-line portion of the address onto the memory bus. Otherwise, it is tristated. MBAOE# is also sampled during snoop cycles. If MBAOE# is sampled inactive with SNPSTB#, the snoop write back cycle(if any) will begin at the sub-line address provided. If MBAOE# is active with SNPSTB#, the snoop write back will begin at sub-line address 0.
MBRDY#(MISTB)	I	Memory Bus Ready, (Memory Input Strobe) This pin is an input to the 82490XP. It is used in clocked bus mode to indicate the end of a transfer. When active(low) it indicates that the 82490XP should increment the burst counter and either output the next data or get ready to accept the next data. In strobed memory bus mode this pin is the input data strobe to the 82490XP. On each MISTB edge, the 82490XP latches the data and increments the burst counter.
MCACHE#	O	82495XP Internal Cacheability This signal is driven by the 82495XP. On read cycles, this signal indicates the cycle's internal cacheability attribute. In write cycles MCACHE# is only active for write-back cycles. MCACHE# is not activated for I/O, special cycles and Locked Cycles.
MCFA6-MCFA0 **MSET10-MSET0** **MTAG11-MTAG0**	I/O I/O I/O	Memory Bus Configurable address lines Memory bus SET number Memory bus TAG bits These are the memory bus address lines of the 82495XP and should be connected to the A31-A2 (A31-A3 for 64 bit bus) signals of the Memory Bus. These signals, along with the byte enables, define the physical area of memory or I/O accessed. The 82495XP drive these signals in normal memory bus cycles and have them as inputs during snooping.
MCLK[MSTBM]	I	Memory Bus Clock, [Memory Input Strobe] In clocked memory bus mode this pin provides the memory bus clock to the 82490XP. In clocked mode, memory bus signals and memory bus data are sampled on the rising edge of the **MCLK**. In a clocked memory bus write, data is driven off of **MCLK** or **MOCLK** depending upon the configuration. This pin is an input to the 82490XP. It is sampled during reset and determines the memory bus type. If this input is strapped high, the memory bus will be strobed. If a clock is detected at this input, this pin becomes the memory bus clock, and clocked memory bus mode is selected.
MDATA0-MDATA7	I/O	Memory Bus Data These pins are the 8 memory data pins of the 82490XP. All or part of these pins will be used depending on the cache configuration. In clocked memory bus mode, these pins are sampled with the rising edge of **MCLK**. New data is driven out on these pins with **MEOC#** or the rising edge of **MCLK** or **MOCLK** while **MBRDY#** or **MEOC#** is active. In strobed memory bus mode, **MDATA** is sampled on each **MISTB** edge. New data is driven out on these pins with each **MOSTB** edge or **MEOC#**-falling edge.

1.2 Quick Pin Reference (Continued)

MDOE#	I	Memory Data Output Enable This signal is an input to the 82490XP. The memory bus output enable is used to control the 82490XP's driving of data onto the memory bus. When this pin is inactive(high), the **MDATA[0:7]** pins are tristated. When this pin is active(low), the **MDATA[0:7]** pins are actively driving data. The function of this pin is the same for strobed or clocked memory bus operation as MDOE# has no relation to CLK or MCLK.
MEOC#	I	Memory End of Cycle This signal is an input to the 82490XP. Since it is synchronous to the memory bus, it may be used to end a cycle on the memory bus and begin a pending cycle without waiting for synchronization to the CPU CLK. MEOC# also causes the latching or driving of data and resetting of the memory burst counter.
MFRZ# [MEMLDRV]	I	Memory Freeze, [Memory Bus Low Drive] This signal is an input to the 82490XP. It is used for write cycles that could cause allocation cycles. When this pin is active(low), write data is latched in the 82490XP. The subsequent allocation will not overwrite data latched by the write. This prevents the actual write to memory from having to be performed on the memory bus. The allocated line will be placed in the [M] state in the cache since memory has not been updated. During RESET's falling edge, this signal is sampled to indicate the 82490XP's memory bus driving strength. The 82490XP provides normal and high drive capability buffers.
MHITM#	O	Memory Bus Hit to Modified Line This signal is driven by the 82495XP during snoop cycles and indicates whether the snooping address hit a Modified line in the 82495XP cache. The 82495XP automatically schedules the writing-back of modified lines when snoop hits occur. MHITM# is activated the CLK after SNPCYC# and will remain active until the next SNPSTB#.
MKEN#	I	Memory Bus Cacheability This signal is an input to the 82495XP. It is the memory bus cache enable pin. It is used to indicate to the 82495XP if the current memory bus cycle is cacheable or not. This pin is sampled by the 82495XP with KWEND# assertion.
MOCLK(MOSTB)	I	Memory Output Clock, (Memory Output Strobe) MOCLK controls a transparent latch at the 82490XP data outputs. By providing a clock input, skewed from MCLK, MDATA hold time may be increased. In strobed bus mode this pin is the data output strobe. On each MOSTB edge, new data will be output onto the memory bus.
MRO#	I	Memory Bus Read-Only This pin is an input to the 82495XP. It is the READ-ONLY attribute pin. It is used to indicate to the 82495XP that the accessed line should get a READ-ONLY attribute. READ-ONLY lines will be non-cacheable in the first level cache. READ-ONLY lines will be cached in the 82495XP if MKEN# is sampled active during KWEND# and will be cached in the [S] state. This pin is sampled by the 82495XP with KWEND# assertion.

1.2 Quick Pin Reference (Continued)

MSEL# [MTR4/TR8#]	I	Memory Select, [Memory Transfer] This signal is a chip select input to the 82490XP. MSEL# activation qualifies the MBRDY# input of the 82490XP. MSEL# going active causes the sampling of MZBT# for the next cycle. MSEL# going inactive resets the 82490XP's internal memory burst counter. This pin is used to determine the number of transfers necessary on the memory bus for each cache line. If high, there are 4 transfers on the memory bus for each cache line. If low, there are 8 transfers on the memory bus for each cache line.
MTHIT#	O	Memory Bus Tag Hit This signal is driven by the 82495XP during snoop cycles. It indicates whether the snooping address hit any line (exclusive, shared, or modified) in the 82495XP cache. MTHIT# is activated the CLK after SNPCYC# and will remain active until the next SNPSTB#.
MWB/WT#	I	Memory Bus Write Policy This signal is an input to the 82495XP. It is the mechanism by which the memory bus can dynamically inform the 82495XP of the cycle write policy (Write-Through/Write-Back). This signal is sampled by the 82495XP with SWEND# activation.
MZBT# [MX4/MX8#]	I	Memory Zero Based Transfer, [Memory I/O Bits] This signal is an input to the 82490XP. When this pin is sampled active (with MSEL# or MEOC#) it indicates that the memory bus cycle should start with burst location zero independent of the sub-line address requested by the CPU. This pin is used to determine the number of IO pins used for the memory bus. When HIGH it indicates that 4 IO pins are used per 82490XP. When LOW it indicates that 8 IO pins are used.
NENE#	O	Next Near This signal is generated by the 82495XP and indicates to the memory bus controller if the address of the requested memory cycle is "near" the address of the previously generated one (in the same 2K DRAM page). This information can be used by the memory bus controller to optimize access to paged or static column DRAMs. This signal is valid together with CADS#.
PALLC#	O	Potential Allocate This signal is generated by the 82495XP and indicates to the memory bus controller that the current write cycle can potentially allocate a cache line. Potential allocate cycles are cycles which are 82495XP misses with PCD, PWT inactive.
RDYSRC	O	Ready Source This signal is an output of the 82495XP. It indicates the source of the BRDY generation for the CPU. When high it indicates that the memory bus controller should generate BRDYs to the CPU, when low it indicates that the 82495XP will be the one providing BRDYs.

1.2 Quick Pin Reference (Continued)

RESET	I	Reset This signal forces the 82495XP and 82490XP to begin execution at a known state. It's falling edge will sample the state of the configuration pins. RESET is an asynchronous input to the 82495XP and 82490XP. **The following 82495XP pins are sampled during reset falling edge:** CNA# [CFG0]: CFG0 line of 82495XP configuration inputs. SWEND# [CFG1]: CFG1 line of 82495XP configuration inputs. KWEND# [CFG2]: CFG2 line of 82495XP configuration inputs. FLUSH# [NCPFLD#]: Enables decoding of the non-cacheable PFLD mode. Active if low. FPFLD# [FPFLDEN]: Enables the external FIFO pfld mode. Active high. BGT# [C490LDRV]: Indicates the driving strength of the 82495XP/82490XP interface. If high, the 82495XP can drive up to 10 82490XP's without derating. If low, the 82495XP can drive up to 18 82490XP's without derating. SYNC# [MEMLDRV]: Indicates the 82495XP's memory bus driving strength. SNPCLK[SNPMD]: Indicates the snoop mode, synchronous or asynchronous. CFG0–CFG2 signals are used to configure the 82495XP/82490XP with cache parameters. They define the lines/sector, line ratio, and number of tags. MALE[WWOR#]: Enforces strong or weak write-ordering consistency. MBALE[HIGHZ#]: If active along with SLFTST# will tristate all 82495XP outputs. **The following 82490XP pins are sampled during reset falling edge:** PAR#: If active(low), this pin configures the 82490XP as a parity storage device. The parity configuration stores the paritybits belonging to data stored in other 82490XP's. MZBT# [MX4/MX8#]: Determines the number of IO pins used for the memory bus interface. If high, four IO pins are chosen. If low, eight IO pins are chosen. MSEL# [MT4/MT8#]: Determines the number of transfers necessary on the memory bus for each cache line. If high, four memory bus transfers are needed to fill a cache line. If low, eight memory bus tranfers are needed to fill a cache line. MCLK[MSTBM#]: If active(low), this pin indicates a strobed memory bus configuration. If inactive(high), a clocked memory bus is chosen. MFRZ# [MEMLDRV]: Indicates the 82490XP's memory bus driving strength.
SMLN#	O	Same Cache Line This signal is an output of the 82495XP. It is used to indicate to the memory bus controller that the current cycle is to the same 82495XP line as the previous one. This indication can be used by the memory bus controller to selectively activate its SNPSTB# signal to other caches. For example, back to back snoop hits to the same line may be snooped only once. This signal is valid together with CADS#.

1.2 Quick Pin Reference (Continued)

SNPADS#	O	Cache Snoop Address Strobe This signal is an output of the 82495XP. It has an identical functionality as CADS#, but is generated only on snooping-write-back cycles. Considering that snoop write-back cycles are the only ones which are generated independent of CPU bus activity, this separate address strobe should ease implementation of the memory bus controller. Whenever active, the memory bus controller should abort all pending cycles (cycles for which BGT# was not issued yet. After BGT# the memory bus controller is responsible for the cycle completion). The 82495XP assumes that non-committed cycles are aborted upon SNPADS# and may re-issue them again after the completion of the snoop.
SNPBSY#	O	Snoop Busy This signal is driven by the 82495XP. When inactive(high), it indicates that the 82495XP is ready to accept another snoop cycle. SNPBSY# will be activated for one of two reasons: A snoop hit to a modified line, a back-invalidation is needed when there is one already in progress. In either of these cases, the 82495XP will not perform the look-up for a pending snoop until SNPBSY# is de-activated.
SNPCLK[SNPMD]	I	Snoop Clock [Snoop Mode] This pin provides the 82495XP with the snoop clock to be used in clocked memory interfaces. During clocked mode SNPSTB#, SNPINV, SNPNCA, MBAOE#, MAOE#, and the Address lines will be sampled by SNPCLK. During RESET activation, this pin functions as the SNPMD (snoop mode) signal. If high it indicates strobed snooping mode. If low it indicates synchronous snooping mode. For clocked snooping mode, SNPCLK is connected to the snoop clock source.
SNPCYC#	O	Snoop Cycle This signal is an output of the 82495XP. It indicates when the snooping look-up is actually taking place in the 82495XP tag RAM.
SNPINV	I	Snoop Invalidation This signal is an input to the 82495XP and indicates the resulting line state in case of a snoop hit cycle. If active, it forces the line to go to an invalid state. This signal is sampled with SNPSTB#.
SNPNCA	I	Snoop Non Caching Device Access This signal is an input to the 82495XP and provides the 82495XP information on whether the current memory bus master is a non caching device (DMA, etc). This indication allows the 82495XP to avoid changing line states from exclusive to shared unnecessarily.
SNPSTB#	I	Snoop Strobe This signal is an input to the 82495XP which is used to initiate a snoop. SNPSTB# causes the latching of the snoop address and parameters. The 82495XP supports three latching modes: Clocked, Strobed, Synchronous. In the clocked mode, address and attribute signals will be latched with the activation of SNPSTB#.SNPCLK. In the strobed mode, address and attributes will be latched by the SNPSTB# falling edge. In synchronous mode, address and attribute signals will be latched with the activation of SNPSTB#.CLK.

1.2 Quick Pin Reference (Continued)

SWEND# [CFG1]	I	Snoop Window End, [Configuration Pin 1] This signal is generated by the MBC and indicates to the 82495XP that the Snoop Window has expired. At this point the 82495XP will latch the memory bus attributes: write policy (MWB/WT#), and direct to [M] transfer (DRCTM#). At the end of the snooping window, all other devices have snooped the bus master's address and have generated address caching attributes on the bus. Once á cycle begins, the 82495XP prevents snooping until it has received SWEND#. The 82495XP will act based on those attributes and will update its tag RAM. During RESET's falling edge this line functions as the CFG1 configuration signal which is used to configure the 82495XP/82490XP with cache parameters.
SYNC# [MEMLDRV]	I	Synchronize 82495XP cache, [Memory Bus Low Drive] This signal is an input to the 82495XP. Activation of this line will cause the synchronization of the 82495XP tag array with main memory. All 82495XP modified lines will be written back to main memory. The difference between FLUSH and SYNC is that on SYNC the 82495XP and CPU tag array will NOT be invalidated. All the valid entries will be kept, with all modified lines (M state) becoming non-modified (E state). During RESET's falling edge, this signal is sampled to indicate the memory bus driving strength. If it is sampled low, the maximum capacitive load without derating is 100pf. If it is sampled high, the maximum capacitive load without derating is 50pf.
TCK	I	Testability Clock This signal is an input to both the 82495XP and 82490XP. This is the boundary scan clock. This signal has to be connected to a clock synchronous to CLK to insure initialization of the test logic.
TDI	I	Testability serial input This signal is an input to both the 82495XP and 82490XP.
TDO	O	Testability serial output This signal is an output of both the 82495XP and 82490XP.
TMS	I	Testability Control This signal is an input to both the 82495XP and 82490XP.

The following pins have internal pull-ups:

ADS#, NA#, FPFLD#, TDI, TMS, BGT#, KWEND#, SWEND#, CNA#, BRDY#, SYNC#, FLUSH#, SNPSTB#, MRO#, DRCTM#, TCK, SNPCLK, MFRZ#, MZBT#, MCLK, MOCLK, BOFF#, PAR#.

During tri-state output testing sequence, all pull-ups will be disabled.

The following signals are glitch free. These signals are always at a valid logic level following RESET:

CADS#, CDTS#, SNPADS#, SNPCYC#, KLOCK#.

1.3 Output Pins

Table 1-3 lists all output pins, from which part(s) they are driven, and their active levels.

Table 1-3. Output Pins

Name	Part	Active Level	Name	Part	Active Level
BLE#	82495XP	LOW	MTHIT#	82495XP	LOW
CADS#	82495XP	LOW	NENE#	82495XP	LOW
CAHOLD	82495XP	HIGH	PALLC#	82495XP	LOW
CDTS#	82495XP	LOW	RDYSRC	82495XP	HIGH
CWAY	82495XP	-	SMLN#	82495XP	LOW
CW/R#,CD/C#,CM/IO#	82495XP	-	SNPADS#	82495XP	LOW
FSIOUT#	82495XP	LOW	SNPBSY#	82495XP	LOW
KLOCK#	82495XP	LOW	SNPCYC#	82495XP	LOW
MCACHE#	82495XP	LOW	TDO	82495XP/82490XP	-
MHITM#	82495XP	LOW			

1.4 Input Pins

Table 1-4 lists all input pins, which part(s) they are input to, their active level, and whether they are synchronous or asynchronous inputs.

Table 1-4. Input Pins

Name	Part	Active Level	Synchronous/Asynchronous
BGT# [C490LDRV]	82495XP	LOW	Synchronous to CLK
BRDY#	82495XP/82490XP	LOW	Synchronous to CLK
CLK	82495XP/82490XP	-	-
CFG3	82495XP	-	Synchronous to CLK
CNA# (CFG0)	82495XP	LOW	Synchronous to CLK
CRDY# [SLFTST#]	82495XP/82490XP	LOW	Synchronous to CLK
DRCTM#	82495XP	LOW	Note 2
FLUSH# [NCPFLD#]	82495XP	LOW	Asynchronous
CPUTYP	82495XP	LOW	Synchronous to CLK
KWEND# (CFG2)	82495XP	LOW	Synchronous to CLK
MALE, MBALE	82495XP	HIGH	Asynchronous
MAOE#, MBAOE#	82495XP	LOW	Asynchronous
MCLK[MSTBM#]	82490XP	LOW	Synchronouos to MCLK
MBRDY# (MISTB)	82490XP		-
MDOE#	82490XP	LOW	Asynchronous
MEOC#	82490XP	LOW	Synchronous/Asynchronous, Note 1

Table 1-4. Input Pins (Continued)

Name	Part	Active Level	Synchronous/Asynchronous
MFRZ#	82490XP	Low	Synchronous/Asynchronous, Note 1
MOCLK(MOSTB)	82490XP		
MSEL[MTR4/TR8#]	82490XP	Low	Synchronous/Asynchronous, Note 1
MZBT#[MX4/MX8#]	82490XP	Low	Synchronous/Asynchronous, Note 1
MKEN#	82495XP	LOW	Note 2
MRO#	82495XP	LOW	Note 2
MWB/WT#	82495XP	-	Note 2
PAR#	82490XP	Low	Synchronous to CLK
RESET	82495XP/82490XP	HIGH	Asynchronous
SNPCLK[SNPMD]	82495XP	-	-
SNPINV	82495XP	HIGH	Note 3
SNPNCA	82495XP	HIGH	Note 3
SNPSTB#	82495XP	LOW	Note 3
SWEND# (CFG1)	82495XP	LOW	Synchronous to CLK
SYNC#[MEMLDRV]	82495XP	LOW	Asynchronous
TCK	82495XP/82490XP	-	
TDI	82495XP/82490XP	-	Synchronous to TCK
TMS	82495XP/82490XP	-	Synchronous to TCK

NOTES:
(1) In Clocked memory bus mode these pins are synchronous with **MCLK**. In Strobed memory bus mode these pins are asynchronous.
(2) MWB/WT#, DRCTM# must be synchronous to CLK during SWEND#. MKEN#, MRO# must be synchronous to CLK during KWEND#.
(3) In clocked memory bus mode these pins are synchronous with SNPCLK. In strobed memory mode these pins are asynchronous.

1.5 Input/Output Pins

Table 1-5 lists all input/output pins, which part they interface with, and when they are floated.

Table 1-5. Input/Output Pins

Name	Part	Synch/Asynch	When Floated
FPFLD#[FPFLDEN]	82495XP	Synchronous to CLK	-
MCFA0-MCFA6	82495XP	Note 1	MAOE# = High
MDATA0-MDATA7	82490XP	Note 2	MDOE# = Hight and during Reset
MSET0-MSET10	82495XP	Note 1	MAOE# = High
MTAG0-MTAG11	82495XP	Note 1	MAOE# = High

NOTES:
(1) With MALE high and MAOE# low, these pins are synchronous to CLK.
(2) In Clocked memory bus mode these pins are synchronous with **MCLK**. In Strobed memory bus mode these pins are asynchronous.

1.6 Pin State During Reset

Table 1-6. Pin State During Reset

Pin Name	Pin State during Reset
CADS#, CDTS#, SNPADS#	High
CW/R#, CD/C#, CM/IO#, MCACHE#	Undefined
RDYSRC, PALLC#, CWAY	Undefined
NENE#, SMLN#	Undefined
KLOCK#	High
FPFLD#	High
MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6	Note 1
CAHOLD	Note 2
MHITM#, MTHIT#	High
SNPCYC#, SNPBSY#	High
TDO	Note 3

NOTES:
(1)MSET, MTAG, and MCFA signals are high impedance during reset if MAOE# and MBAOE# are deasserted.
(2) The state of CAHOLD depends on whether self-test is selected (see testability chapter for details).
(3) The State of TDO is controlled by the boundary scan which is independent of other signals including RESET (see testability chapter for details).

1.7 Quick Pin Reference (Optimized Interface)

The table below lists and describes the pins that comprise the optimized interface. This is the interface between the i860 XP CPU, 82495XP Cache Controller, and the 82490XP SRAM that has been refined and optimized to allow zero wait state operation at 50 MHz.

The information provided in the table below is helpful in identifying the pins that make up the optimized interface to facilitate interface layout and debugging. Timing information appears in chapter ten.

Symbol	Type		Name and Function
A3–A31 A0–A15	O I	CPU 490	The i860 XP CPU **Address** Outputs A3–A31 are connected to the 82490XP address inputs A0–A15 and to the 82495XP address inputs of SET, TAG, and CFA.
ADS#	O I I	CPU 495 490	The i860 XP CPU **Address Strobe** output is connected to the 82495XP and 82490XP ADS# inputs and indicates the start of a CPU cycle. Please refer to the i860 XP CPU data sheet for details.
AHOLD	I O	CPU 495	**Address Hold** is driven by the 82495XP to the i860 XP CPU AHOLD input during back-invalidation cycles. Please refer to the i860 XP CPU data sheet for details.
BE0#–BE7# BE#	O I	CPU 490	The i860 XP CPU **Byte Enable** outputs are driven to an external latch controlled by the 82495XP. Each byte enable also goes to an 82490XP to control partial write cycles. Please refer to the i860 XP CPU data sheet for details.
BLAST#	O I	495 490	The 82495XP **Burst Last** output is driven to indicate the end of a cycle. It is connected to the 82490XP BLAST# input.
BOFF#	I O I	CPU 495 490	The **Backoff** input of the i860 XP CPU is driven by the 82495XP to resolve contention on the CPU bus. See the i860 XP CPU data sheet for details.

1.7 Quick Pin Reference (Optimized Interface) (Continued)

Symbol	Type		Name and Function
BRDYC#	I	490	**Burst Ready Cache** is an input to the 82490XP that is connected to the 82495XP BRDYC2# output and is used for tracking these hit cycles.
BRDYC1# RSRVD	O	495	**Burst Ready Cache 1** is output by the 82495XP to the i860 XP CPU RSRVD input during cache hit and posted cycles.
BRDYC2#	O	495	**Burst Ready Cache 2** is output by the 82495XP to the 82490XP BRDYC# input during cache hit and posted cycles.
BUS#	O I	495 490	The **Bus/Array Select** output of the 82495XP multiplexes either the memory bus path or array path to the CPU bus of the 82490XP.
CDATA0–7	I/O	490	**Cache Data** I/O pins are the 8 bits comprising the I/O data bus interface between each 82490XP and the i860 XP CPU data bus.
CFA0–CFA6	I/O	495	**Cache Function and Address** pins of the 82495XP are multiplexed to the i860 XP CPU address according to the 82495XP configuration.
D/C#	O I	CPU 495	The **Data/Code** output of the i860 XP CPU is used by the 82495XP to decode special cycles. Please refer to the i860 XP CPU data sheet for details.
D0–D63	I/O	CPU	Data Bus The i860 XP CPU pins are distributed to each 82490XP. Please refer to the i860 XP CPU data sheet for details.
DP0–DP7	I/O	CPU	The i860 XP CPU **Data Parity Bus** pins are distributed to each 82490XP. Please refer to the i860 XP CPU data sheet for details.
EADS#	I O	CPU 495	The i860 XP CPU **External Invalidation Address Strobe** is driven by the 82495XP during back-invalidation and inquire cycles to maintain inclusion. Please refer to the i860 XP CPU data sheet for details.
EWBE#	I O	CPU 495	The i860 XP CPU **External Writeback Buffer Empty** input is driven by the 82495XP for use in Strong Ordering mode. See the i860 XP CPU data sheet for details.
HITM#	0 I I	CPU 495 490	The i860 XP CPU **Hit Modified** output indicates a snoop hit to a modified line when the 82495XP performs a snoop to the CPU.
INV	I O	CPU 495	The **Invalidate** input to the i860 XP CPU is driven by the 82495XP on back invalidation cycles.
KEN#	I O	CPU 495	The i860 XP CPU **Cache Enable** input is driven by the 82495XP during cacheable cycles. Please refer to the i860 XP CPU data sheet for details.
LEN	O I	CPU 495	The i860 XP CPU drives the **Length** pin active to indicate a cycle of 2 transfers. With LEN inactive, the cycle length is 1 transfer. If a cycle is cacheable, LEN is undefined and the cycle length is 4 transfers.
LOCK#	O I	CPU 495	The **Cycle Lock** pin of the i860 XP CPU is driven to the 82495XP which in turn drives the memory bus KLOCK# output. Please refer to the i860 XP CPU data sheet for details.
M/IO#	O I	CPU 495	The i860 XP CPU **Memory/IO** pin is used by the 82495XP to decode memory and I/O cycles. Please refer to the i860 XP CPU data sheet for details.
MAWEA#	O I	495 490	The 82495XP asserts **Memory Bus Array Write Enable or Allocation** signal to the 82490XP to indicate that the data in the memory buffers should be written to the array, or that an allocation should occur.
MCYC#	O I	495 490	The 82495XP asserts **Memory Bus Cycle** to the 82490XP to indicate that the current cycle will use the memory buffers.
NA#	I O	CPU 495	The 82495XP drives **Next Address** to the i860 XP CPU when pipelineing CPU-to-cache cycles. See the i860 XP CPU data sheet for details.
PCD	O I	CPU 495	The i860 XP CPU **Page Cacheability Disable** attribute bit is driven on PCD to indicate cacheability. PCD active causes the 82495XP to make the current cycle non-cacheable. Please refer to the i860 XP CPU data sheet for details.

1.7 Quick Pin Reference (Optimized Interface) (Continued)

Symbol	Type		Name and Function
PCYC	O I	CPU 495	The i860 XP CPU drives the **Page Cycle** output during page-table special cycles. See the i860 XP CPU data sheet for details.
PWT	O I	CPU 495	The i860 XP CPU **Page Write-Through** attribute is driven on PWT to indicate write-through. PWT active causes the 82495XP to make the current cycle write-through. Please refer to the i860 XP CPU data sheet for details.
SET0–10	I/O	495	The 82495XP **Set Address** pins are connected to 11 bits of the CPU address.
TAG0–11	I/O	495	The 82495XP **Tag Address** pins are connected to 12 bits of the CPU address.
W/R#	O I I	CPU 495 490	The i860 XP CPU **Write/Read** pin is driven to the 82495XP and 82490XP to indicate a read or write cycle. Please refer to the i860 XP CPU data sheet for details.
WAY	O I	495 490	The 82495XP **Way** indication is used by the 82490XP to properly load and store buffers as well as update the MRU bit.
WB/WT#	I O	CPU 495	The **Write Back/Write Through** input to the i860 XP CPU is driven by the 82495XP to direct MESI protocol changes in the CPU's on-chip caches. See the i860 XP CPU data sheet for details.
WBA (SEC2#)	O I	495 490	The **Write Back Buffer Address (2 lines per sector)** pin is driven by the 82495XP to indicate which line is loaded by the 82490XP. The 82495XP drives SEC2# during RESET to the 82490XP to pass along lines/sector information. If active, SEC2# indicates 2 lines per sector.
WBTYP (LR0)	O I	495 490	The 82495XP **Write Back Cycle Type (Line Ratio 0)** pin is driven to the 82490XP to indicate a write-back or snoop write-back cycle. LR0 is driven by the 82495XP to the 82490XP at RESET to pass along line ratio information.
WBWE# (LR1)	O I	495 490	The 82495XP **Write-Back Buffer Write Enable (Line Ratio 1)** pin is used in conjunction with the WBA pin to load the write-back buffers. LR1 is driven by the 82495XP to the 82490XP at RESET to pass along line ratio information.
WRARR#	O I	495 490	The 82495XP **Write to 82490XP Array** signal controls the writing of data into the 82490XP array and updating of the MRU bit.

2.0 CHIPSET INTRODUCTION

The 82495XP/82490XP is a second-level cache controller chipset for the i860 XP CPU. The chipset provides a unified code and data cache which is software transparent. The 82495XP/82490XP has been designed to support a high-speed CPU/cache core interface, and a same or lower speed memory bus interface.

The 82495XP is the cache controller. It contains 8K tags and control logic to control up to a 512K size cache. The 82490XP is a custom cache data RAM designed to be used with the 82495XP. Between 8 and 18 82490XPs are required to create a 256K to 512K cache, respectively. The memory bus controller (MBC) is the set of logic required to interface the 82495XP and 82490XP to the memory bus. The MBC provides product differentiation, and its implementation ultimately determines system performance.

2.1 Main Features

The 82495XP/82490XP have the following main features:

— Tracks the speed of the i860 XP CPU

— Large Cache Size support:

4K or 8K Tags

1 or 2 lines per sector

4 or 8 transactions per line

64 memory bus or 128-bit wide memory bus

256K or 512K cache

— Write-Back cache with full multiprocessing consistency support:

supports the MESI protocol

watches memory bus to guarantee 1st level, 2nd level cache consistency

maintains inclusion

— Two-way set-associative with MRU hit prediction algorithm

- Zero wait state hit cycles on MRU hit. One wait state on MRU misses
- Concurrent CPU and Memory Bus transactions
- Supports synchronous, asynchronous, and strobed memory bus architectures

2.2 CPU/Cache Core Description

Figure 2-1 depicts a block diagram of the basic cache subsystem. The cache subsystem provides a gateway between the CPU and the memory bus. All CPU accesses which can be serviced locally by the cache subsystem will be filtered out from the memory bus traffic. Therefore local cycles (CPU cycles which hit the cache and do not require a memory bus cycle) will be completely invisible to the memory bus providing the reduction in memory bus bandwidth necessary for multiprocessing systems. Another very important function of the 82495XP cache subsystem is to provide speed decoupling between the CPU and memory busses. Processors are quickly achieving operating frequencies which can be very difficult for the memory subsystem to meet. The 82495XP cache subsystem is optimized to serve the CPU with zero wait-states up to very high frequencies (50 Mhz), at the same time providing the decoupling necessary to run slower memory bus cycles.

The Basic Functions of the cache subsystem elements are:

82495XP: Main control element, includes the tags and line states and provides hit or miss decisions. It handles the CPU bus requests completely and coordinates with the memory bus controller when an access needs the memory bus. It controls the 82490XP data paths for both hits/misses to provide the CPU with the correct data. It dynamically adds wait states based on the MRU prediction mechanism. The 82495XP is also responsible for performing memory bus snoop operations while other devices are using the memory bus. The 82495XP drives the cycle address and other attributes during a memory bus access. A block diagram of the 82495XP is shown in Figure 2-2.

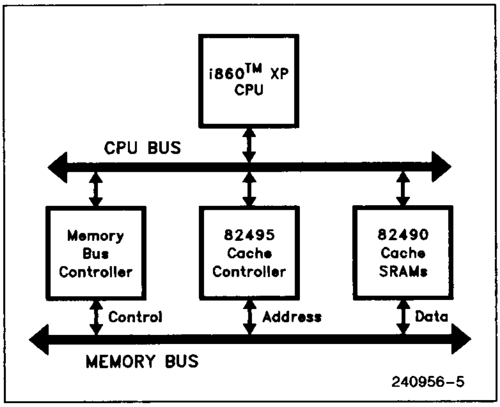


Figure 2-1. 82495XP Cache Subsystem

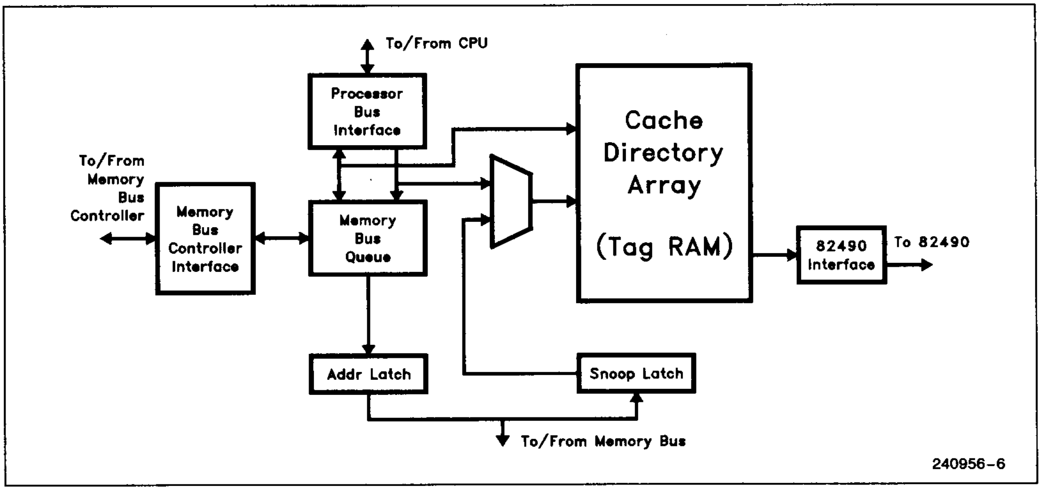


Figure 2-2. 82495XP Block Diagram

82490XP: Implements the cache SRAM storage and data path. It includes latches, muxes, logic which allow it to work in lock-step with the 82495XP to efficiently serve both hit and miss accesses. It takes full advantage of internal silicon flexibility to provide a degree of performance otherwise unachievable with discrete implementations. It supports zero wait state hit accesses, concurrent CPU and memory bus accesses, and includes a replication of the MRU bits for autonomous way prediction. During memory bus cycles it acts as a gateway between CPU and memory buses. A block diagram of the 82490XP is shown in Figure 2-3.

Memory Bus Controller: Server for memory bus cycles. It adapts the CPU/Cache core to a specific memory bus protocol. It coordinates with the 82495XP line fills, flushes, write-backs, etc. The memory bus controller's flexibility allows customers to easily adapt the 82495XP cache subsystem to their specific architectures, and to provide their own differentiation. Figure 2-4 shows an example memory bus controller. The MBC handles all cycle control, data transferring, snooping, and any synchronization.

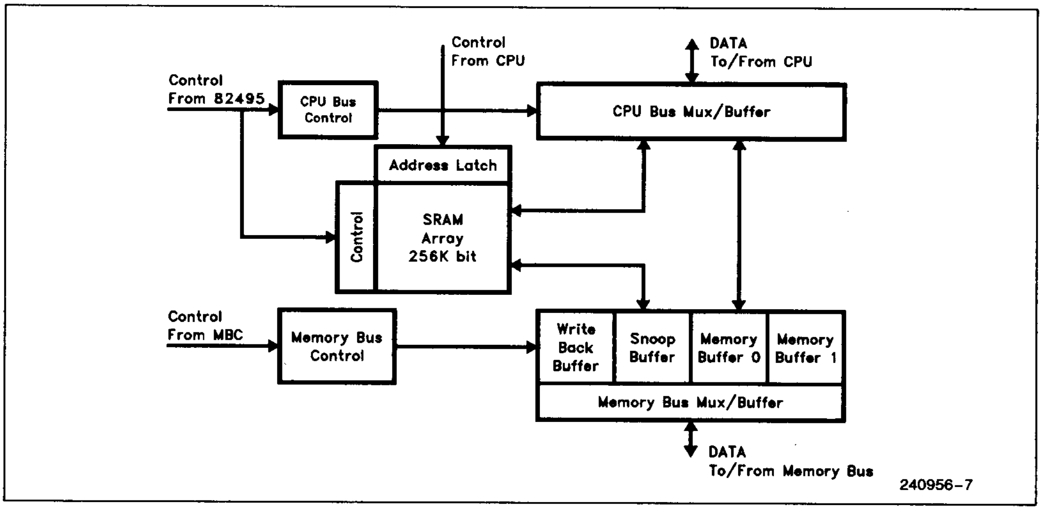


Figure 2-3. 82490XP Block Diagram

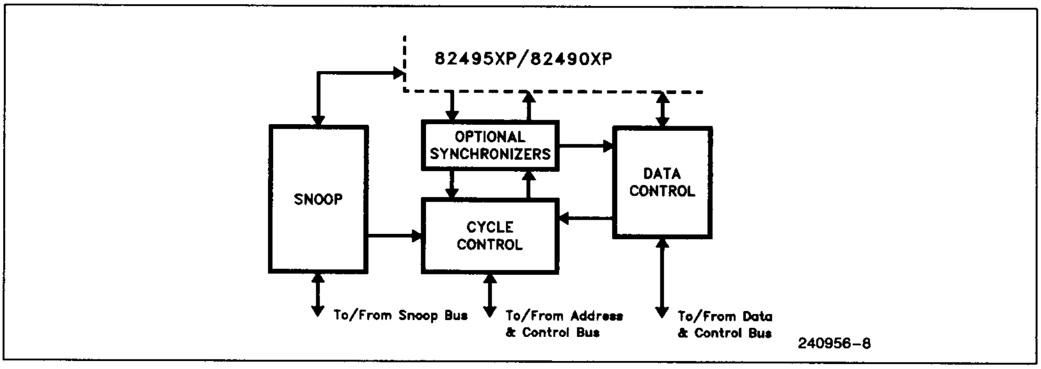


Figure 2-4. MBC Example Block Diagram

3.0 CACHE OVERVIEW

This chapter gives a brief description of 82495XP/82490XP configurations, interface, snooping mechanism, cycle control mechanism, and memory bus control mechanism. Each section of this overview is described in more detail in later chapters.

3.1 Configuration

The 82495XP/82490XP cache chipset offers a number of configuration options. The system designer can choose from a number of different operating characteristics, including memory bus modes, snooping modes, and internal physical attributes (line size, lines per sector, etc.). The flexibility of these configuration options allow the 82495XP/82490XP cache to be used in a wide range of applications.

Configurations are selected by altering the 82495XP/82490XP inputs during RESET. They are not dynamically changeable, and to conserve pins some configuration inputs become 82495XP or 82490XP inputs/outputs after RESET.

3.1.1 PHYSICAL CACHE

Physically, the 82495XP/82490XP can be configured to support many different cache configurations. By selecting one cache configuration, other configurations may be excluded. The 82495XP/82490XP can be configured to support:

- 256K or 512K cache
- 64 or 128 bit wide memory bus
- One or two lines per sector
- 1:1, 1:2, or 1:4 CPU to 82495XP line size ratio
- 4 or 8 memory bus transactions per line
- 4K or 8K tag size
- Strong or weak write ordering

Figure 3-1 summarizes the basic configurations available when using the 82495XP/82490XP.

3.1.2 SNOOP MODES

When another master snoops the 82495XP, the MBC must initiate the snoop request and pass on the response. The 82495XP allows the MBC to initiate this snoop request in one of three modes: synchronous, clocked, and strobed. The snoop response of the 82495XP is always synchronous.

When initiating the snoop in synchronous snoop mode, all snoop information is latched by the 82495XP synchronous to the CPU CLK. The snoop is then performed on the next CLK edge and the response given on the CLK edge after that. This is the fastest possible method of snooping.

In clocked snooping mode, information is latched by the 82495XP with respect to an external snoop clock (slower than CLK) source. The 82495XP must internally synchronize this information to CLK and provide a response.

In strobed snooping mode, information is latched into the 82495XP with respect to the falling edge of another signal. Thus, the snoop initiation is clock independent. The 82495XP again synchronizes this information with CLK.

	MEM BUS = 64 Bits		MEM BUS = 128 Bits		Number of 82490XP Devices
	4 Trans.	8 Trans.	4 Trans.	8 Trans.	
256K	1 LR = 1 Tags = 8k L/S = 1	2 LR = 2 Tags = 4k L/S = 1	▨	▨	8
512K	3 LR = 1 Tags = 8k L/S = 2	4 LR = 2 Tags = 8k L/S = 1	4 LR = 2 Tags = 8k L/S = 1	5 LR = 4 Tags = 4k L/S = 1	16

Cache Device
2, 4, 8 Bits Wide

▨ Not Supported   LR = 82495XP/CPU Line Ratio
L/S = 82495XP Lines/Sector

Figure 3-1. 82495XP/82490XP Configurations

3.1.3 MEMORY BUS MODES

The 82490XP may be configured to be in one of two memory bus modes. This mode determines how data will be passed on to and off of the data bus. The two modes are clocked mode and strobed mode. These modes need not have any relation to the snoop mode chosen.

In clocked mode, data is driven from an external memory clock source called MCLK, or read with respect to MCLK. MCLK is completely independent of the CPU CLK source. There are inherent performance advantages, however, in making this clock source synchronous or half-clock (divided) synchronous to the CPU CLK.

In strobed mode, data is driven from the rising edge of one signal, and read with the rising edge of another. Like the strobed snooping mode, this carries no clock skew problems, or memory bus speed limitations.

3.2 CPU Bus Interface

The CPU bus interface is the connection of the 82495XP and 82490XP to the i860 XP CPU. Because this interface is optimized to achieve the high speed performance, it is not a flexible interface. The majority of the signals in the CPU bus interface must be connected strictly between the 82495XP/82490XP cache and the i860 XP CPU. Chapter 10 addresses the use of such signals.

Some CPU signals are, however, accessible by the MBC. These are the following pins: RESET, CLK, BRDY2#, INT, BERR, PCHK#, PEN#, TCK, TDI, TMS, TRST#, and TDO. CPU pins KB0, KB1, HIT#, and BREQ are also available to the MBC, but are of limited use in an 82495XP/82490XP system.

Other CPU pins flow through a '377 type latch to the MBC. The latch enable is controlled by the 82495XP through the BLE# pin. The following CPU signals flow through this latch: PCD, PWT, BE0#-BE7#, CACHE#, LEN, PCYC, and CTYP.

3.3 82495XP/82490XP Interface

The 82495XP/82490XP interface is the connection between 82495XP and 82490XP. Like the CPU bus interface, this isolated interface is not flexible and may not be altered beyond what Intel has provided.

3.4 Memory Bus and Memory Bus Controller Interface

The memory bus controller (MBC) is the interface logic required to control the 82495XP/82490XP and connect it to the memory bus and rest of the system. The MBC may be simple enough to support a single-CPU write-through cache, or complex enough to support a multiprocessing cache with external tags. The 82495XP/82490XP is a very flexible chipset, and the MBC determines exactly how the 82495XP/82490XP will work in a system.

An MBC consists of a few basic blocks: a snoop logic block, a cycle control block (with synchronizers if necessary), and data path control block. The snoop block must be able to communicate with the other caches when snooping is necessary. At the same time, the cycle control block must interface to some arbitration logic for bus arbitration.

3.4.1 SNOOPING LOGIC

The MBC snooping logic is responsible for initiating a snoop in the 82495XP and providing the response to the rest of the system. Snoop logic must recognize what other caches are doing, and snoop if necessary. Snoop logic must also recognize when its 82495XP is not capable of snooping and delay its snoop initiation.

When a cycle begins on the bus, all other caches snoop. Once all the snoop results are returned to the master 82495XP, its snoop logic must recognize the result and alter the cycle appropriately. This could mean aborting the current cycle in memory, delaying the cycle until a write-back is performed, or changing the master's tag state according to the snoop information.

3.4.2 CYCLE CONTROL LOGIC

Cycle control logic is responsible for initiating a memory bus cycle, providing proper 82495XP cycle attributes during the cycle, and terminating the cycle. Cycle control logic determines the cacheability of the cycle, whether cycles are allocatable, pipelining, and all aspects of the progress of the current cycle.

Since cycle control logic interfaces memory bus signals to the 82495XP, and since the memory bus is not necessarily synchronous to the 82495XP CLK, it may also provide proper synchronization. Careful design of this synchronization logic can minimize or eliminate synchronization penalties.

3.4.3 DATA PATH CONTROL

Data path control logic controls how data is written from the 82490XP or read into the 82490XP and CPU. It handles the actual transferring of data to/from the memory data bus. Data path control logic also handles the CPU burst order, and the holding of data during allocation cycles. In systems with memory busses that are wider than the CPU bus, the data path control logic appropriately steers data to the correct 82490XP's.

3.5 Test

The 82495XP/82490XP provide two means of cache testing. These are a built-in self-test, and boundary scan test. The built-in self-test (BIST) is initiated during RESET. The boundary scan test uses separate and dedicated pins on the 82495XP. These are described in a later chapter.

4.0 CACHE CONSISTENCY PROTOCOL

One of the 82495XP objectives is to implement a high performance second level cache for multiprocessor systems. To fulfill this objective the 82495XP implements a "write-back" cache with full support for multiprocessing data consistency. Being a write-back cache means that the 82495XP may contain data which is not updated in the main memory. Therefore a mechanism is implemented to insure that data read by any system bus master, at any time, is correct.

A key feature for multiprocessing systems is reduction of the memory bus utilization. The memory bus quickly becomes a resource bottleneck with the addition of multiple processors. The 82495XP cache consistency mechanism insures minimal usage of memory bus bandwidth.

The 82495XP allows portions of memory to be defined as non-cacheable. For the cacheable areas, the 82495XP allows selected portions to be defined as write-through locations.

The 82495XP protocol is implemented by assigning state bits for each cached line. Those states are dependent on both 82495XP data transfer activities performed as the bus master, and snooping activities performed in response to snoop requests generated by other memory bus masters.

4.1 Cache Consistency Protocol Model

The 82495XP consistency protocol is the set of rules which allows the 82495XP to contain data that is not updated in main memory while ensuring that memory accesses by other devices do not receive stale data. This consistency is accomplished by assigning a special consistency state to every cached entry (line) in the 82495XP.

NOTE:
The following rules apply to memory read and write cycles. All I/O and special cycles bypass the cache.

The 82495XP protocol consists of 4 states. They define whether a line is valid (hit or miss), if it is available in other caches (shared or exclusive), and if it is modified (has been modified).

The 4 States are:

[I] - INVALID Indicates that the line is not available in the cache. A read to this line will be a miss and cause the 82495XP to execute a line fill (fetch the whole line and deposit it into the cache SRAM). A write to this line will cause the 82495XP to execute a write-through cycle to the memory bus and in some circumstances initiate an ALLOCATION.

[S] - SHARED This state indicates that this line is potentially shared with other caches (The same line may exist in more than one cache). A Shared line can be read out of the cache SRAM without a main memory access. Writing to a Shared line updates the 82495XP/82490XP cache, but also requires the 82495XP to generate a write-through cycle to the memory bus. In addition to updating main memory, the write-through cycle will invalidate this line in other caches. Since writing to a Shared line causes a write-through cycle, the system can enforce a "write-through policy" to selected addresses by forcing those addresses into the [S] state. This can be done by setting the PWT attribute in the CPU page table or asserting the MWB/WT# pin each time the address is referenced.

[E] - EXCLUSIVE This state indicates a line which is exclusively available in ONLY this cache, and that this line is NOT MODIFIED (main memory also has a valid copy). Writing to an Exlusive line causes it to change to the **M**odified state and can be done without informing other caches, so no memory bus activity is generated.

[M] - MODIFIED This state indicates a line which is exclusively available in ONLY this cache, and is MODIFIED (main memory's copy is stale). A **M**odified line can be updated locally in the cache without acquiring the memory bus. Because a **M**odified line is the only up-to-date copy of data, it is the 82495XP's responsibility to flush this data to memory on accesses to it. Flushing of this data to memory will be executed immediately after completion of the current CPU bus cycle.

4.2 Basic State Transitions

This section covers the most common, basic memory accesses. The special functions which force a cycle to be noncacheable, locked, read only, or direct-to-**M**odified are not in use. These might be used, for example, in read for ownership and cache to cache transfers, and are covered in section 4.3. This basic transitions section is divided into two parts: the first covers MESI state changes which occur in a CPU/cache core due to its own actions; the second describes MESI state transitions in a CPU/cache core caused by the actions of other, external devices. Figure 4-1 shows a partial state diagram of the MESI coherency protocol which includes these basic transitions.

The 82495XP accepts line attributes from the CPU and memory buses. The 82495XP assumes that all caches on the memory bus have the SAME number of bytes per line.

4.2.1 TRANSITIONS IN CACHE STATES CAUSED BY OWN CPU TRANSACTIONS

The MESI state of each 82495XP/82490XP cache line changes as the 82495XP/82490XP services the read and write requests generated by its CPU.

4.2.1.1 Read Hit

A read hit occurs when the CPU generates a read cycle on its bus, and the data is present in and returned by the 82495XP/82490XP. The state of the cache line (M, E, or S) remains unchanged by a read operation which hits the cache.

4.2.1.2 Read Miss

A read miss arises when the CPU generates a read, and the data is not present in the 82495XP/82490XP cache—either the tag lookup does not produce a match or a match occurs but the data is **I**nvalid. The 82495XP generates a memory access to fetch the data (which is assumed cacheable for this discussion) and the surrounding data needed to fill the cache line. This data is placed in the 82495XP/82490XP cache in an invalid line or (if both valid) replaces the least recently used line, which is written back to memory if **M**odified.

The new line is placed in the **E**xclusive state, unless either the CPU or memory indicates that it should be a write-through on its next write access using PWT or MWB/WT#, respectively. If either of these is asserted, the new line is placed in **S**hared state. A new line could also be read in and placed directly into **M**odified state: see section 4.3.4 for details and use.

4.2.1.3 Write Hit

When the CPU generates a write cycle, if the data is present in the 82495XP/82490XP cache, it is updated and may undergo a MESI state change.

If the hit line is originally in the **E**xclusive state, it changes to **M**odified state upon a write. If the hit line is originally in the **M**odified state, it remains in that state. Neither of these cases generates any bus activity.

A write to a line which is in the **S**hared state causes the 82495XP to write the data out to memory as well as update the 82495XP/82490XP cache. The write to main memory also serves to invalidate any copy of the data which resides in another cache. The cache line state changes according to activity on the PWT and MWB/WT# pins. If neither of these pins is asserted, the write hit line becomes Exclusive. If either of these pins is asserted, the line is forced to remain write-through, so the state remains **S**hared.

An existing line can also be written and forced directly into Modified state: see section 4.3.4 for details and use.

4.2.1.4 Write Miss

The CPU generates a write cycle, and the data is not present in the 82495XP/82490XP cache. In a simple write miss, the 82495XP/82490XP assists CPU in delivering data to memory, but the data is not placed in the cache. No cache lines are affected, so no state changes take place.

4.2.1.5 Write Miss with Allocate

This is a special case of a write miss where the memory location written by the CPU is not currently in the 82495XP/82490XP cache, but is brought into the cache and updated. Like a regular write miss, the 82495XP/82490XP assists the CPU in writing the data out to main memory. After the data is written to memory, the 82495XP/82490XP reads back the same data following the rules of a read miss, above.

The ability to perform an allocation depends on all of the following conditions:

the write is cacheable

PWT is not asserted, forcing write-through

the write is not LOCKed

the write is to memory (not to I/O)

4.2.2 TRANSITIONS CAUSED BY OTHER DEVICES ON BUS

MESI state transitions in the 82495XP/82490XP cache of one core (CPU/82495XP/82490XP) can be induced by actions initiated by other cores or devices on the shared memory bus. In the following, the 82495XP which is responding to actions of other devices does not currently own the bus, and may be referred to as a "slave" or, in the case of snooping, a "snooper". The device which currently owns the bus is the "master".

4.2.2.1 Snooping

The master which is accessing data from memory on the bus sends a request to all caching devices on the bus (snoopers) that they check or snoop their caches for a more recently updated version of the data being accessed. If one of the snoopers has a copy of the requested data, it is termed a "snoop hit".

If a snooper has a modified version of the data ("snoop hit to a **M**odified line"), it proceeds to generate an "inquire cycle" to the i860 XP CPU, asking the i860 XP CPU if it also has a **M**odified copy of the line (which would be more recently modified than the 82495XP/82490XP's version). The most up-to-date line is written out by the snooping 82495XP/82490XP to the bus (to main memory or directly to the requesting master) so that the requesting master can utilize it.

The changes in MESI protocol state in a snooping cache which has a snoop hit depend on attribute inputs SNPINV and SNPNCA, which are driven by the master.

The SNPINV input tells a snooping 82495XP/82490XP to invalidate the line being snooped if hit: the master requesting the snoop is about to write to its copy of this line and will therefore have the most up-to-date copy. When SNPINV is asserted on the snoop request, any snoop hit is placed in Invalid state, and a "back invalidation" is generated which instructs the CPU to check its cache and likewise invalidate a copy of the line. When the snooping 82495XP has a snoop hit to a **M**odified line and SNPINV was asserted by the bus master, the back invalidate is combined with the inquire cycle.

The SNPNCA input tells a snooping 82495XP/82490XP whether the requesting master is performing a **N**on-**C**aching **A**ccess. If the requesting master is not caching the data, a snoop hit to a **M**odified or **E**xclusive line can be placed in the **E**xclusive state: since the requester isn't caching the line, if the snooper has a future write hit to the line, an invalidation does not have to be broadcast. If the requesting master is caching the data, then a snoop hit to a **M**odified or **E**xclusive line must be placed in the **S**hared state, which insures that a future write hit causes an invalidation to other caches. Note that a snoop hit to a **S**hared line must remain in the **S**hared state regardless of SNPNCA. Also note that an asserted SNPINV always overrides SNPNCA.

4.2.2.2 Cache Synchronization

Cache synchronization is performed to bring the main memory up-to-date with respect to the 82495XP/82490XP. Two devices exist in the 82495XP/82490XP to accomplish this: FLUSH and SYNC.

A cache flush is initiated by asserting the 82495XP FLUSH# pin. Once initiated, the 82495XP writes all **M**odified lines out to main memory, performing back invalidations and inquire cycles on the CPU. When completed, all 82495XP/82490XP and CPU cache entries will be in the Invalid state.

Activation of the SYNC# pin also causes all of the 82495XP's **M**odified lines to be written to memory. Unlike the FLUSH# pin, the cache lines remain valid after the SYNCH# process has completed, with **M**odified lines changing to the **E**xclusive state.

4.3 The Effects of Special Cycles on MESI States

4.3.1 NON-CACHEABLE ACCESS

The 82495XP allows cacheability to be determined on both a per page and per line basis. The page cacheability function is determined by software, while cacheability on a line-by-line basis is driven by hardware.

The PCD (**P**age **C**aching **D**isabled) pin is a 82495XP input driven by the CPU's PCD output, which corresponds to a cacheability bit in the page table entry of a memory location's virtual address. If the PCD bit is asserted when the CPU presents a memory address, that location will not be cached in either the 82495XP or the CPU.

MKEN# is a 82495XP input which connects to the memory bus controller or the memory bus. MKEN# inactive prevents the caching of the memory location in both the 82495XP and the CPU, affecting only the current access.

If a read miss is indicated non-cacheable by either of these, the line is not placed in the 82495XP/82490XP or CPU cache, and no cache states are modified. On a write miss, a noncacheable indication from either input forces a write miss

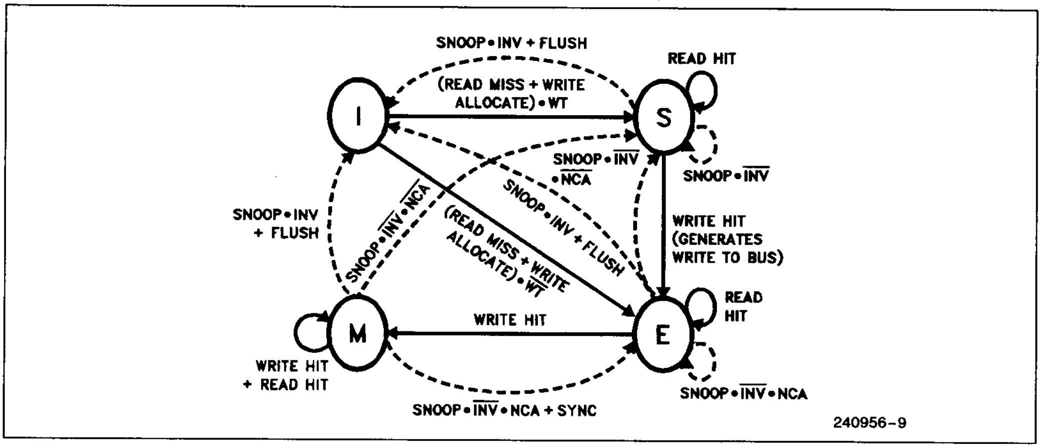

SNOOP•INV + FLUSH
(READ MISS + WRITE ALLOCATE)•WT
READ HIT
I
S
SNOOP•$\overline{INV}$ •$\overline{NCA}$
SNOOP•$\overline{INV}$
SNOOP•INV + FLUSH
SNOOP•$\overline{INV}$•NCA
(READ MISS + WRITE ALLOCATE)•$\overline{WT}$
SNOOP•INV+FLUSH
WRITE HIT (GENERATES WRITE TO BUS)
READ HIT
WRITE HIT
M
E
WRITE HIT + READ HIT
SNOOP•$\overline{INV}$•NCA + SYNC
SNOOP•$\overline{INV}$•NCA
240956-9

Figure 4-1. Major State Transitions

without allocation. Note that if the 82495XP/82490XP already has a valid copy of the line, the PCD attribute from the CPU is ignored.

4.3.2 READ ONLY ACCESSES: MRO#

The MRO# (Memory Read Only) input is driven by the memory bus to indicate that a memory location is read only.

When asserted during a read miss line fill, MRO# causes the line to be placed in the 82495XP/82490XP cache in the Shared state and also sets a read-only bit in the cache tag. MRO# accesses are not cached in the CPU. On subsequent write hits to a read-only line, the write is actually written through to memory without updating the 82495XP/82490XP line, which remains in the Shared state with the read-only bit set.

4.3.3 LOCKED ACCESSES: LOCK#

The LOCK# signal driven by the CPU indicates that the requested cycle should lock the memory location for an atomic memory access. Because locked cycles are used for interprocessor and intertask synchronization, all locked cycles will appear on the memory bus.

On a locked write, the 82495XP treats the access as a write-through cycle, sending the data to the memory bus—updating memory and invalidating other cached copies. If the data is also present in the 82495XP/82490XP cache, it is updated but its M, E, or S state remains unchanged.

For locked reads, the 82495XP assumes a cache miss and starts a memory read cycle. If the data resides in the 82495XP/82490XP, the M-E-S state of the data remains unchanged. If the requested data is in the 82495XP/82490XP and is in the **Modi**fied state when the memory bus returns data, the 82495XP will use the 82490XP data and ignore the memory bus data.

LOCKed read and write cycles which miss the 82495XP/82490XP cache are noncacheable in both the 82495XP/82490XP and CPU.

4.3.4 FORCING LINES DIRECT-TO-MODIFIED: DRCTM#

The DRCTM# (Direct To Modified) pin is an input which informs the 82495XP to skip the **E**xclusive state and place a line directly in the **M**odified state. The signal can be asserted during 82495XP/82490XP reads of the memory for special 82495XP/82490XP data accesses like read-for-ownership and cache-to-cache-transfer. The signal can also be asserted during writes, for purposes of cache tracking.

4.4 State Tables

Lines cached by the 82495XP can change states as a result of either the CPU bus activity (that sometimes require the 82495XP to become a memory bus master) or as a result of memory bus activity generated by other system masters (snooping).

State transitions are affected by the type of CPU/memory bus transactions (reads, writes) and by a set of external input signals and internally generated variables. In addition, the 82495XP will drive certain CPU/memory bus signals as a result of the consistency protocol.

4.4.1 CPU BUS

— *PWT* (Page Write Through, **PWT** Input pin) Indicates a CPU bus write-through request. Activated by the i860 XP CPU **PWT** pin. This signal affects line fills and will cause a line to be put in the [S] state if active. The 82495XP will NOT execute ALLOCATIONS (line fills triggered by a write) for write-through lines. If **PWT** is asserted, it overrides a write-back indication on the **MWB/WT#** pin.

— *PCD* (Page Cacheability Disable, **PCD** input pin): indicates that the accessed line is noncacheable. If **PCD** is asserted, it overrides a cacheable indication from an asserted **MKEN#**.

— *NWT* (i860 XP CPU Write-Through Indication, 82495XP's **WB/WT#** Output Pin): When low forces the i860 XP CPU to keep the accessed line into the SHARED state.

Write back mode (WB = 1) will be indicated by the *!NWT* notation. In those cases the i860 XP CPU is allowed to go into exclusive states [E], [M]. *NWT* is normally active unless explicitly stated.

— *KEN* (CPU caching enable, **KEN#** output pin): When active indicates that the requested line can be cached by the CPU 1st level cache. *KEN* is normally active unless explicitly stated.

4.4.2 MEMORY BUS

— *MWT* (Memory Bus Write-Through Indication, **MWB/WT#** Input Pin): When active forces the 82495XP to keep the accessed line into the SHARED state. Write back mode (MWB = 1) will be indicated by the */MWT* notation. In those cases the 82495XP is allowed to go into exclusive states [E], [M].

— *DRCTM* (Memory Bus Direct To [M] indication, DRCTM# Input Pin): When active forces skipping of the [E] state and direct transfer to [M].

— *MKEN* (Memory Bus Cacheability Enable, **MKEN#** Input pin): When Active Indicates that the memory bus cycle is cacheable.

— *MRO* (Memory Bus Read-Only Indication, **MRO#** Input Pin): When Active forces line to be READ-ONLY.

— *MTHIT* (Tag Hit, MTHIT# Output pin): Activated by the 82495XP during snoop cycles and indicates that the current snooped address hits the 82495XP cache.

— *MHITM* (Hit to a line in the [M] State, **MHITM#** Output pin): Activated by the 82495XP during snoop cycles and indicates that the current snooped address hits a modified line in the 82495XP cache.

— *SNPNCA* (Non Caching device access): When active indicates to the 82495XP that the current bus master is a non-caching device.

— *SNPINV* (Invalidation): When active indicates to the 82495XP that the current snoop cycle will invalidate that address.

4.4.3 TAG STATE

— *TRO* (Tag Read Only, 82495XP Tag bit): This bit when set indicates that the 1 or 2 lines associated with this tag are Read-Only lines.

As a function of State Changes the 82495XP may execute the following cycles:

— *BINV:* Execution of a CPU Back Invalidation Cycle (Snoop with **INV** active)

— *INQR:* Execution of a i860 XP CPU Inquire Cycle[1].

— *WBCK:* 82495XP Write-Back Cycle. This is a Memory Bus write cycle generated by the 82495XP when MODIFIED data cached in the 82495XP needs to be copied back into main memory. A write-back cycle affects a complete 82495XP line.

— *WTHR:* 82495XP Write Through Cycle. This is a system write cycle in response to a processor write. It may or may not affect the cache SRAM (update). In a write-through cycle, the 82495XP drives the Memory Bus with the same Address, Data and Control signals as the CPU does on the CPU Bus. Main Memory is updated, and other Caches invalidate their copies.

— *RTHR:* 82495XP Read Through cycle. This is a special cycle to support locked reads to lines that hit the 82495XP cache. The 82495XP will request a Memory Bus cycle for lock synchronization reasons, data will be supplied from the BUS except for [M] state which will have data supplied from the CACHE.

— *LFIL:* 82495XP Cache line fill. 82495XP will generate Memory Bus cycles to fetch a new line and deposit into the cache.

— *RNRM:* 82495XP Read Normal Cycle: This is a normal read cycle which will be executed by the 82495XP for non-cacheable accesses.

— *SRUP:* 82495XP SRAM UPDATE. Occurs any time new information is placed in the 82495XP cache. An SRAM update is implied in the *LFIL* cycle.

— *ALLOC:* 82495XP ALLOCATION. Write Miss cycle that has determined to be cacheable so the 82495XP issues a line read.

NOTE:

[1] An inquire cycle may be executed with INV active, performing a back-invalidation simultaneously.

STATE TABLES

Table 4-1. Master 82495XP Read Cycle

Pres. State	Condition: Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	!LOCK: M	—	!NWT	Normal Read Hit [M]
	LOCK: M	RTHR	!KEN	Read Through Cycle, Data From Array
E	!LOCK: E	—	NWT	Normal Read Hit [E]
	LOCK: E	RTHR	!KEN	Read Through Cycle, Data From Memory
S	!LOCK.!TRO: S	—	NWT	Normal Read Hit [S]
	!LOCK.TRO: S	—	!KEN	Normal Read to Read-Only sector. Stays in [S] state and deactivate KEN to prevent CPU from caching line
	LOCK: S	RTHR	!KEN	Read Through Cycle, Data from Memory
I	PCD + !MKEN + LOCK: I	RNRM	!KEN	Non-Cacheable Read, Locked cycles
	!PCD.MKEN.!LOCK.MRO: S	LFIL	!KEN	Cacheable read, Read-Only. Fill line to 82495XP. Do not allow CPU to cache line by deactivating KEN#. Set the 82495XP's TRO bit to indicate the sector read only attribute
	!PCD.MKEN.!LOCK.!MRO.(PWT + MWT):S	LFIL	NWT	Cacheable Reads, forced Write-Through
	!PCD.MKEN.!LOCK.!MRO.!PWT.!MWT.!DRCTM: E	LFIL	NWT	Line not shared, thus enabling the 82495XP to move into tan exclusive state
	!PCD.MKEN.!LOCK.!MRO.!PWT.!MWT.DRCTM: M	LFIL	NWT	As before with direct [M] state transfer. Keep i860 XP CPU in Write Through mode

Table 4-2. Master 82495XP Write Cycle

Pres. State	Condition: Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	!LOCK: M	-	SRUP, !NWT	Write hit. Write to cache. Allow i860 XP CPU to perform internal write cycles (Enter intor [E], [M] states).
	LOCK: M	WTHR	SRUP, !NWT	Locked Cycle. Write-Through updating cache SRAM. Most updated copy of the line is still owned by 82495XP. All Locked write cycles are posted.
E	!LOCK: M	-	SRUP, !NWT	Write hit. Update SRAM. Let i860 XP CPU execute internal write cycles.
	LOCK: E	WTHR	SRUP, NWT	Lock forces cycle to memory bus. Main memory remains updated.
S	TRO: S	WTHR	-	Read-Only. Write cycle with write through attribute from CPU or Memory Bus. Locked Cycles.
	!TRO.(PWT + MWT + LOCK): S	WTHR	SRUP, NWT	Not Read-Only. Write cycle with write through attribute from CPU or Memory Bus. Locked Cycles.
	!TRO.!PWT.!LOCK.!MWT.!DRCTM: E	WTHR	SRUP, NWT	Not Read-Only. No write-through cycle, no lock request allow going into exclusive state.
	!TRO.!PWT.!LOCK.!MWT.DRCTM: M	WTHR	SRUP, NWT	Not Read-Only. No write-through cycle, no lock request allow going into exclusive state. DRCTM forces final state to M.
I	PCD + !MKEN + PWT + LOCK + MRO: I	WTHR	-	Write Miss Non-Cacheable, Write-Through, locked cycle or Read-Only.
	!PCD.MKEN.!PWT.!LOCK.!MRO: I !PCD.MKEN.!PWT.!LOCK.MRO:S Allocation Final State MWT:S !MWT.!DRCTM:E !MWT.DRCTM:M	WTHR, LFIL ALLOC	-	Write Mis with allocation. After the write cycle, a line fill (allocation) is scheduled. If MKEN and MRO are asserted, an allocation to the [S] state will occur Allocation final state as a function of line fill attributes.

NOTE:
The **WB/WT#** pin will only be activated for 82495XP lines that are in the [M] state. In this state, the 82495XP always assumes that the line owner MAY be the i860 XP CPU. On all other states the i860 XP CPU will be forced to perform Write-Through cycles. This mechanism will make sure that any i860 XP CPU write cycle is seen at least once on the CPU Bus. Allocations, which are consequences of write-misses, will disregard the MKEN# and MRO# attributes during the line fill. In other words, once an allocation is scheduled, it cannot be cancelled.

Table 4-3. Snooping 82495XP without Invalidation Request

Pres. State	Condition: Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	!SNPNCA: S SNPNCA: E	MTHIT MHITM WBCK	INQR	Snoop hit to modified line. 82495XP indicates tag hit and modified hit. 82495XP schedules flushing of the modified line to memory. If non-cacheable device, stay in [E] state.
E	!SNPNCA: S SNPNCA: E	MTHIT	-	If snooping by cacheable device, indicate MTHIT and go to shared state. If no caching device only indicate MTHIT, stay exclusive.
S	S	MTHIT	-	
I	I	-	-	

NOTE:
Usage of DRCTM# to avoid [E] states may be in conflict with the SNPCNA cycle attribute. Note in the table that snoops with SNPNCA may cause an [E] state transition.

Table 4-4. Snooping 82495XP with Invalidation Request

Pres. State	Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	I	MTHIT MHITM WBCK	INQR, BINV	Snoop hit to modified line. 82495XP indicates tag hit and modified hit. 82495XP schedules flushing of the modified line to memory. Invalidate CPU.
E	I	MTHIT	BINV	Inidicate tag hit, infalidate 82495XP, CPU lines.
S	I	MTHIT	BINV	Same as before
I	I	-	-	

Table 4-5. SYNC Cycles

Pres. State	Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	E E	WBCK WBCK	INQR -	Get modified data from i860 XP CPU, flush to memory
E	E	-	-	Memory already synchronized
S	S	-	-	Memory already synchronized
I	I	-	-	

Table 4-6. FLUSH Cycles

Pres. State	Next State	Mem Bus Activity	CPU Bus Activity	Comments
M	I	WBCK	INQR, BINV	Flush and invalidate i860™ XP CPU
E	I	—	BINV	Invalidate i860 XP CPU
S	I	—	BINV	Invalidate i860 XP CPU
I	I	—	—	

NOTE:
Usage of DRCTM# to avoid [E] states may be in conflict with the SYNC cycle. Note in the table that SYNC cycles move an [M] state line to [E].

5.0 CONFIGURATIONS

The 82495XP/82490XP cache system was designed to fit a variety of applications. For the greatest performance, each application requires the 82495XP/82490XP to be configured differently. The 82495XP/82490XP therefore has many possible configurations that are set on RESET and affect the 82495XP/82490XP architecture, operation, and electrical characteristics.

5.1 Physical Cache

The physical configurations of the 82495XP/82490XP consist of parameters that alter the 82495XP/82490XP basic architecture. These are line ratio, tag size, lines per sector, bus width, and cache size. These parameters are sampled at the falling edge of RESET and are not dynamically changeable.

Because of physical cache constraints, choosing one parameter limits the flexibility of other parameters. The following table summarizes the possible i860 XP CPU basic cache configurations. CFG0–CFG2 are multiplexed to select one of 5 possible line ratio/tag size/lines per sector configurations. This information is automatically passed from the 82495XP to 82490XP during RESET. CFG0–CFG3 must be valid at least 10 clocks before RESET's falling edge.

	MEM BUS = 64 Bits		MEM BUS = 128 Bits		Number of 82490XP Devices
	4 Trans.	8 Trans.	4 Trans.	8 Trans.	
256KB	1 LR = 1 Tags = 8k L/S = 1	2 LR = 2 Tags = 4k L/S = 1	▨	▨	8
512KB	3 LR = 1 Tags = 8k L/S = 2	4 LR = 2 Tags = 8k L/S = 1	4 LR = 2 Tags = 8k L/S = 1	5 LR = 4 Tags = 4k L/S = 1	16

▨ Not Supported

LR = 82495XP/CPU Line Ratio
L/S = 82495XP Lines/Sector
Trans = Memory Bus Transactions per 82490XP Line Fill

Figure 5-1. 82495XP/82490XP Configurations

Table 5-1. CFG Configuration Inputs

Cfig No.	Line Ratio	Lines/ sec	No. of Tags	CFG2	CFG1	CFG0
1	1	1	8K	0	0	1
2	2	1	4K	1	1	1
3	1	2	8K	0	0	0
4	2	1	8K	0	1	1
5	4	1	4K	1	1	0

5.1.1 LINE RATIO (LR)

Line Ratio (LR) is the ratio of the 82495XP/82490XP cache line size to the CPU cache line size. For example, if LR=2 then the 82495XP/82490XP line size is 64 bytes. This information is also used to determine the number of back invalidations or inquire cycles to the i860 XP CPU.

5.1.2 TAG SIZE (TAGS)

The 82495XP/82490XP cache tag size may be 4K or 8K tag entries. By reducing tag size, the line ratio (LR) can be doubled without a change in cache size.

5.1.3 LINES PER SECTOR (L/S)

The 82495XP/82490XP may be non-sectored (L/S = 1) or contain two lines per sector (L/S = 2). If L/S = 2, then the 82495XP contains one tag for two consecutive cache lines and each cache line has its own set of MESI state bits. This allows just one line to be filled on replacements or written back on snoop hits. Both lines are written back during replacements, if both are modified.

5.1.4 BUS SIZE

The 82495XP/82490XP supports 64 and 128 bit memory bus widths for the i860 XP CPU.

5.1.5 CACHE SIZE

The 82495XP/82490XP may be configured to be 256K or 512K. Cache size is a direct result of the number of 82490XP devices used. It takes 8 82490XP's to make a 256K byte cache and 16 82490XP's for a 512K cache.

5.1.6 FUNCTION AND ADDRESS CONNECTIONS (CFA0–CFA6)

Table 5-2 lists which address lines should be connected to each of the CFA0–CFA6 lines for each cache configuration. CFA0–CFA6 provide the 82495XP with proper multiplexed addresses for each of the possible cache configurations. Depending on the mode selected, either CFA5 or CFA4 will operate as the 82495XP's CTYP input. This input is connected to the i860 XP CPU's CTYP output.

Table 5-2 also lists the connections between the 82495XP's TAG and SET lines and the remaining CPU address lines.

The 82495XP MCFA, MTAG, and MSET pins connect to the system memory address bus in the same order as the corresponding CFA, TAG, and SET pins are connected to the CPU. Note that the MCFA pin which corresponds to the CFA pin being used as CTYP should be left unconnected.

Table 5-2. CFA Address Connections

Cfig No.	Line Ratio	Lines/ sec	No. of Tags	CFA6	CFA5	CFA4	CFA3	CFA2	CFA1	CFA0	TAG[11:0]	SET[10:0]
1	1	1	8K	A5	CTYP	A31	A30	A29	A4	A3	A28–A17	A16–A6
2	2	1	4K	A5	CTYP	A31	A30	A29	A4	A3	A28–A17	A16–A6
3	1	2	8K	A6	A5	CTYP	A31	A30	A4	A3	A29–A18	A17–A7
4	2	1	8K	A6	A45	CTYP	A31	A30	A4	A3	A29–A18	A17–A7
5	4	1	4K	A6	A5	CTYP	A31	A30	A4	A3	A29–A18	A17–A7

5.2 Cache Modes

Cache modes are ways of configuring the 82495XP/82490XP to operate differently. These options are all sampled at RESET and are not dynamically changeable. If some of these configuration options share a pin, such as the 82495XP's SYNC# and MEMLDRV, the configuration option must meet a specific setup and hold time to RESET's falling edge. For the 82495XP, setup time is usually 4 clocks, and for the 82490XP, setup time is usually 1 clock. For both parts, the configuration option must be held until RESET is detected low.

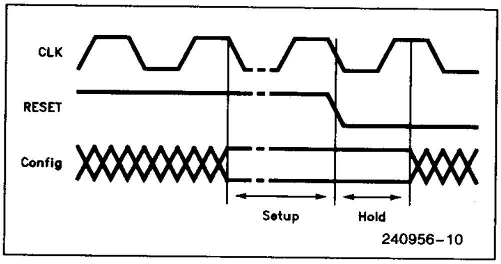


Figure 5-2. Configuration Input Sampling

5.2.1 MEMORY BUS MODES

The 82495XP/82490XP may be configured to have a clocked or strobed memory bus. Memory bus mode is selected by the 82490XP MSTBM pin (same as MCLK pin). If MSTBM is strapped high, the 82490XP's operate in strobed mode. If MSTBM is toggling, ie it is connected to the memory bus clock, the 82490XP operates in clocked mode. MCLK need not be synchronous to CLK.

5.2.2 SNOOPING MODES

The 82495XP/82490XP supports three snooping modes: synchronous, clocked, and strobed. Snooping mode is selected by the SNPMD (same as SNPCLK) pin. If SNPMD is low the 82495XP snoops synchronously. If SNPMD is high the 82495XP snoops in strobed mode. If SNPMD is toggling, clocked mode is selected and SNPMD becomes a snoop clock source, SNPCLK, which clocks in the snoop requests.

These three snooping modes only alter the way the memory bus controller may initiate a snoop request to the 82495XP. The 82495XP response is always synchronous to the CPU CLK.

5.2.3 BUS DRIVERS

The 82495XP/82490XP provide 2 types of memory bus drivers: High capacitance drivers and low capacitance drivers. The high capacitance drivers are selected by driving both the 82495XP and 82490XP MEMLDRV pins low at RESET. Similarly, the low capacitance drivers are selected with MEMLDRV high.

With C490LDRV the 82495XP also provides two types of drivers when driving the 82490XP's. Refer to the interface document to determine C490LDRV.

5.2.4 STRONG/WEAK WRITE ORDERING

If the 82495XP pin WWOR# is sampled low at RESET, the 82495XP enforces weak write-ordering. If sampled high, the 82495XP enforces strong write-ordering. Strong write-ordering prevents the 82495XP from completing a write cycle that would go to 'M' state if a posted write is pending (has not been granted the bus with BGT#). By doing this, strong ordering ensures that write cycles from the CPU are written to memory in the same order that they appear in the i860 XP CPU program.

5.2.5 i860™ XP CPU PFLD SUPPORT

The i860 XP microprocessor executes PFLD (Pipelined Floating-Point Load) instructions to implement special data handling, typically for vector operations. This instruction allows loading of data through a FIFO pipeline, to hide memory latency. The i860 XP CPU does not cache data returned by a PFLD cycle.

The 82495XP can be configured to decode the i860 XP microprocessor's PFLD cycles. The 82495XP supports 3 operational modes for PFLD cycle decoding:

Mode #1. PFLD cycles are cached in the 82495XP.

This mode is used in applications that can fit entirely in the 82495XP/82490XP cache. The 82495XP treats PFLD cycles as normal read cycles.

Mode #2. PFLD cycles are not cached in the 82495XP, without an external PFLD extension FIFO.

This mode is used when applications are too large to fit in the 82495XP/82490XP cache. The 82495XP treats PFLD cycles as noncacheable, using the same protocol as cycles with PCD=1 (if data is already cached, it will be supplied from the cache).

Mode #3. PFLD cycles not cached in the 82495XP, with an external PFLD extension FIFO.

This mode allows the PFLD FIFO to be extended beyond the three stages built into the i860 XP CPU by adding external FIFO hardware. The 82495XP, treats PFLD cycles in the same manner as its treatment of LOCKed cycles (all cycles go to the bus, even if data already present in cache). To support the external FIFO, the 82495XP identifies PFLD cycles by asserting its FPFLD output. For proper operation, data which can be accessed by PFLD must never be in the cache in the Modified state, and software must be aware of the length of the combined PFLD pipeline. Because this mode is not software transparent, it must be used with extreme care.

The choice of PFLD mode is largely application dependent. The PFLD mode of the 82495XP is selected by configuration pins FPFLDEN and NCPFLD#, which are sampled at RESET. FPFLDEN shares a pin with FPFLD, and NCPFLD# shares a pin with FLUSH#. Depending on the PFLD mode, data for reads will either be supplied to the CPU from the 82495XP, or from the memory bus. Table 5-3 summarizes, the 82495XP's support for i860 XP CPU PFLD cycles.

Table 5-3. 82495XP PFLD Modes

Mode #	FPFLDEN	NCPFLD #	Data Supplied From				Line Fill on [I]
			[I]	[S]	[E]	[M]	
1	0	1	MEMBUS	CACHE	CACHE	CACHE	Yes
2	0	0	MEMBUS	CACHE	CACHE	CACHE	No
3	1	1	MEMBUS	MEMBUS	MEMBUS	MEMBUS	No
X	1	0	Illegal Mode				

5.3 82490XP Bus Configuration

The 82490XP needs to be configured so it knows to drive 4 or 8 MDATA lines and whether it should do 4 or 8 memory transfers per line fill. This is done through the MX4/MX8# and the MTR4/MTR8# configuration inputs. For a given line ratio (memory bus line size / CPU line size), they should be sampled as follows:

Table 5-4. MX/MTR Configurations

Line Ratio	MX4/ MX8#	MTR4/ MTR8#	Membus I/O	CPUbus I/O
1	1	1	4	4
2	1	0	4	4
2	0	1	8	4
4	0	0	8	4
1	0	1	8	8
2	0	0	8	8

5.3.1 82490XP PARITY CONFIGURATION

A 82490XP may be designated as a parity device. This is done by strapping the PAR# pin low. In this configuration CDATA[0:3] are used to store 4 parity bits, and CDATA[4:7] are used as 4 bit enables. The four bit enables allow the writing of individual parity bits.

Every mode and configuration of a non-parity 82490XP may be used and selected on the parity 82490XP device. The 82490XP parity configurations are as follows:

Table 5-5. Parity Configurations

Cache Size	Memory Bus Width	Number of Parity Devices	82490XP I/O bits (CPU/Mem)
256K	64	2	4/4
516K	64	2	4/4
512K	128	2	4/8

5.3.2 CPU 82490XP ADDRESS CONFIGURATIONS

The 82490XP Address inputs (A) are multiplexed to the CPU address lines (CA) according to the cache size:

Table 5-6. 82490XP Address Connections

Size	82490XP Address Pins															
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
256K	CA 16	CA 15	CA 14	CA 13	CA 12	CA 11	CA 10	CA 9	CA 8	CA 7	CA 6	CA 5	CA 4	CA 3	V_{SS}	V_{SS}
512K	CA 17	CA 16	CA 15	CA 14	CA 13	CA 12	CA 11	CA 10	CA 9	CA 8	CA 7	CA 6	CA 5	CA 4	CA 3	V_{SS}

NC = No Connect.

6.0 CACHE OPERATION

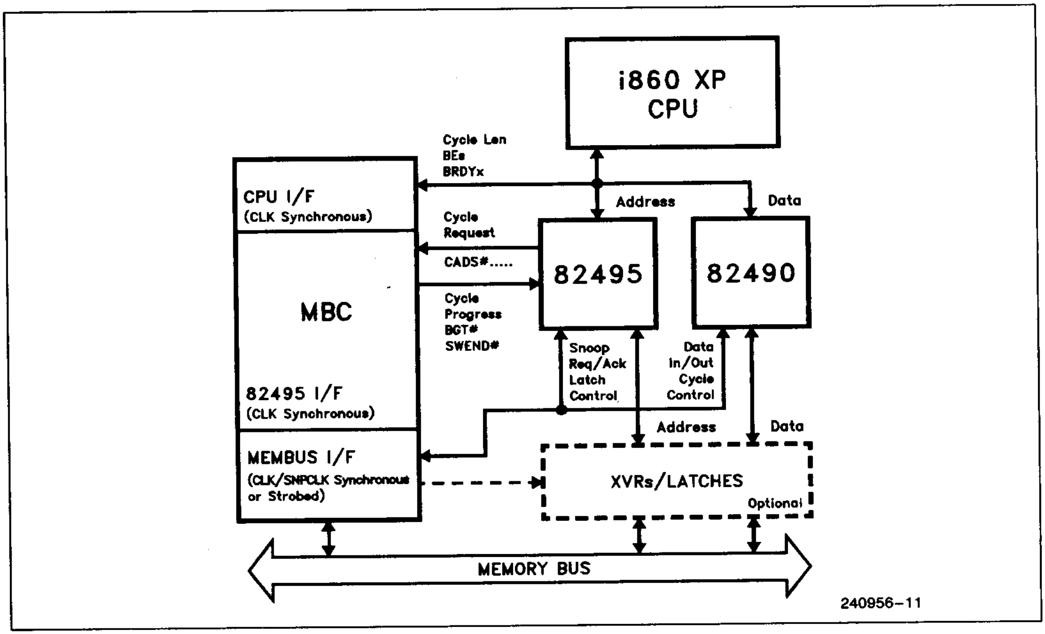


Figure 6-1. Memory Bus Controller Interface Model

Figure 6-1 shows the memory bus controller (MBC) interface model. The memory bus controller interfaces to the i860 XP CPU, 82495XP, 82490XP, and memory bus. The MBC interface was defined with a minimal set of assumptions as to the memory bus implementation. The chipset was designed to enable flexibility in the design of a memory bus and controller.

The 82495XP requests control of the memory bus by signalling the memory bus controller. The memory bus controller is responsible for arbitrating and granting the bus to the 82495XP. Once granted, the memory bus controller is responsible for executing the requested cycle, snooping the other caches, and ending the cycle. The 82495XP supports different modes of snooping, different modes of memory bus operation, and various special cycles. Memory Bus Controller design dictates which of these features are used, and exactly how they are used.

6.1 Cycle Attribute and Progress

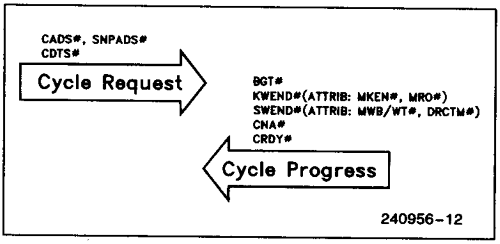


Figure 6-2. Cycle Attribute and Progress Signals

CADS# indicates the start of the cycle address phase. CDTS# tracks CADS# and indicates the start of the cycle data phase. For READ cycles it indicates that starting in the next CLK the CPU data bus is in read mode under the control of the MBC until the last BRDY#. In Read cycles, if the MBC already owns the CPU data bus, CDTS# will be activated with CADS#. For ALLOCATE cycles the MBC does not need the CPU data bus, therefore CDTS# is activated together with CADS#.

For Write cycles CDTS# indicates that the 1st piece of data is available on the memory bus. For write-back cycles CDTS# indicates that all data is available (write-back buffer or snoop buffer loaded with correct write-back data).

As a response to the cycle request, the memory bus controller responds with cycle progress signals. All cycle progress signals are sampled ONCE in specific windows and then ignored until CRDY# of the corresponding cycle. BGT# indicates a commitment by the memory bus controller to complete the cycle execution on the memory bus. Up until this point the 82495XP owns the cycle. This means that intervening snoop-write-backs will abort it and the 82495XP re-issues the cycle to the MBC. There is only one case where the 82495XP will issue a new, not a re-issued, cycle; if the original CADS# operation is a write-back cycle, and the interrupting snoop cycle hits that write-back buffer, then the subsequent CADS# will be for a completely new cycle (not a re-issuing of the interrupted CADS# operation).

After BGT# the memory bus controller owns the cycle. The 82495XP assumes the cycle will terminate and will not re-issue it on snoop-write-backs. Following BGT# comes KWEND# which indicates that the cacheability window is closed and that the 82495XP can sample MKEN#, MRO# attributes. Those indicate to the 82495XP cacheability and read-only respectively. These attributes can be determined by decoding the 82495XP address. Based on those attributes the 82495XP executes ALLOCATIONS, Line-fills, Replacements, etc.

Following KWEND#, SWEND# is activated. It indicates that the Snoop Window is closed. The 82495XP samples MWB/WT# and DRCTM# attributes. These attributes are determined by snooping the other caches in the system. At this point the 82495XP updates its TAGRAM state related to the line access in progress.

Lastly the MBC issues CRDY#, which indicates to the 82495XP the end of the transaction data phase.

The 82495XP allows memory bus pipelining by providing CNA# which allows the MBC to request a new address phase before the conclusion of the current data phase. The 82495XP supports a 1 level deep address pipeline on the Memory Bus.

6.2 Snoop Operations

The 82495XP provides the capability of snooping operations on the memory bus to ensure cache consistency. A snoop operation consists of two phases: 1) initiation phase and 2) response phase.

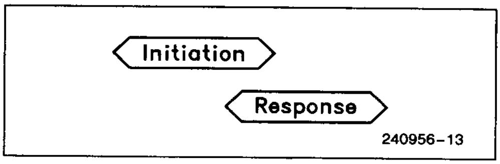


Figure 6-3. 82495XP Snooping Operations

During the initiation phase the MBC provides the 82495XP with the snoop address information. During the response phase the 82495XP provides the snoop status information.

6.2.1 SNOOP INITIATION PHASE

The 82495XP provides three modes for initiating snoops:

1. Strobed: the falling edge of SNPSTB# is used.
2. Clocked: SNPSTB# is sampled with SNPCLK.
3. Synchronous: SNPSTB# is sampled with CLK.

These three snooping modes are configured as follows:

1. Strobed: The SNPCLK[SNPMD] signal must be strapped high.
2. Clocked: The SNPCLK[SNPMD] signal must be connected to the snoop clock source.
3. Synchronous: The SNPCLK [SNPMD] signal must be strapped low.

NOTE:

The 82495XP samples the SNPCLK[SNPMD] signal at the falling edge of RESET to determine the snoop mode. If a rising edge occurs on the SNPCLK[SNPMD] after RESET has gone inactive, clocked mode will be selected. Systems using stobed or synchronous mode must ensure that no rising edge occur on SNPCLK[SNPMD] after RESET has gone inactive.

Figure 6-4 shows the strobed method of snoop initiation. The memory address, SNPNCA, SNPINV, and MBAOE# are latched with the falling edge of the SNPSTB#. If MAOE# is sampled active (low), the SNPSTB# will not cause a snoop. The snoop initiation is recognized by the 82495XP, is synchronized in the next clock, and causes a snoop in the following clock.

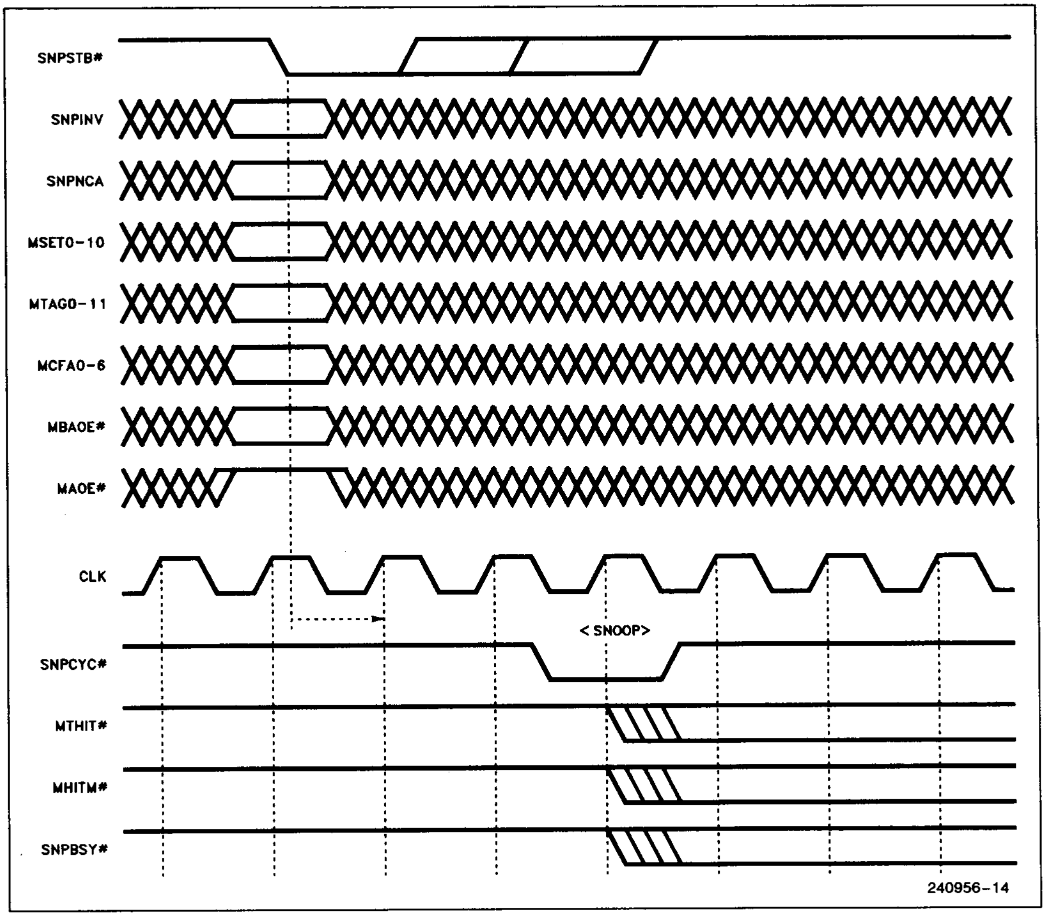


Figure 6-4. Strobed Snoop Mode

Figure 6-5 shows the clocked method of snoop initiation. The memory address, SNPNCA, SNPINV, and MBAOE# are latched with the rising edge of SNPCLK when SNPSTB# is first sampled low. SNPSTB# must be sampled high for at least one SNPCLK in order to rearm for another snoop. If MAOE# is sampled active (low), the SNPSTB# will not cause a snoop. The snoop initiation is recognized by the 82495XP, is synchronized in the next clock, and causes a snoop in the following clock.

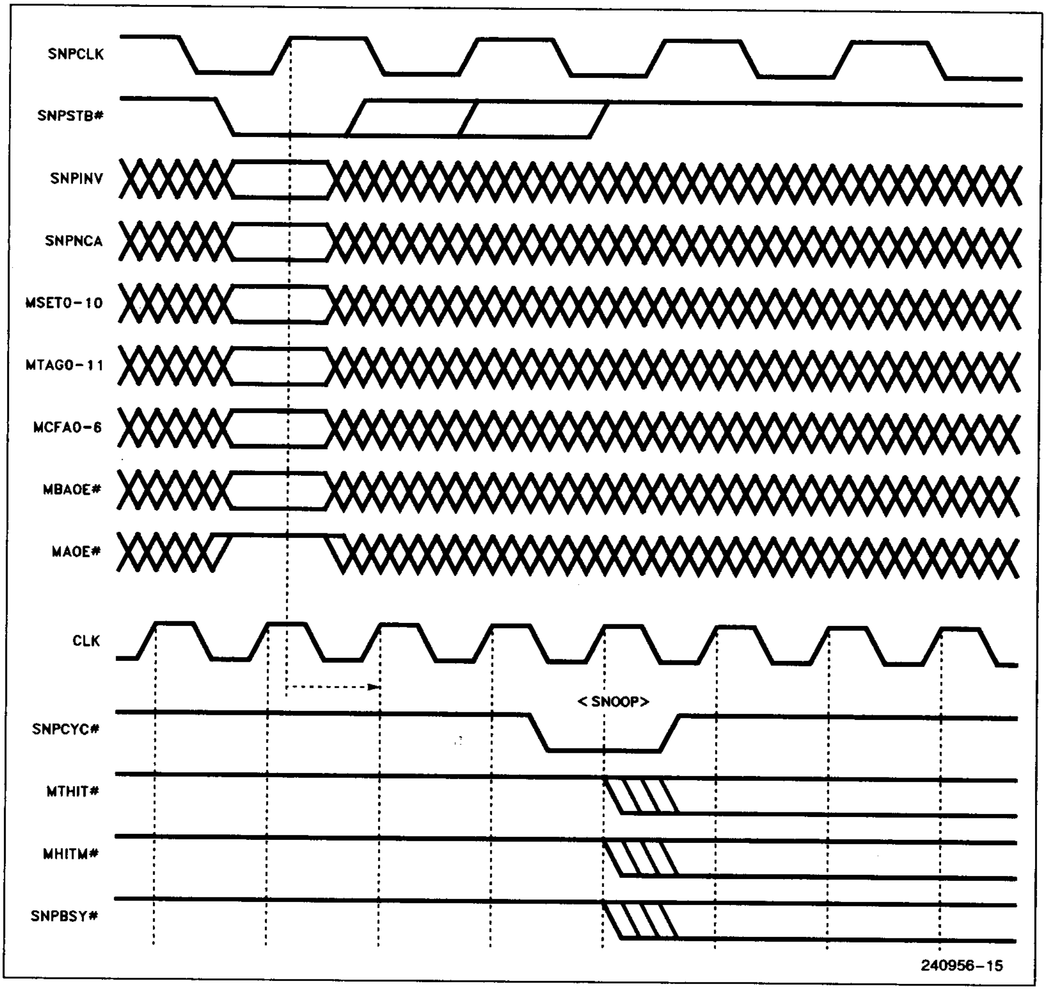


Figure 6-5. Clocked Snoop Mode

Figure 6-6 shows the synchronous snoop mode. The memory address, SNPNCA, SNPINV, and MBAOE# are latched with the rising edge of CLK when SNPSTB# is first sampled low. SNPSTB# must be sampled high for at least one CLK in order to rearm for another snoop. If MAOE# is sampled active (low), the SNPSTB# will not cause a snoop. The snoop initiation is recognized by the 82495XP, and causes a snoop in the next clock.

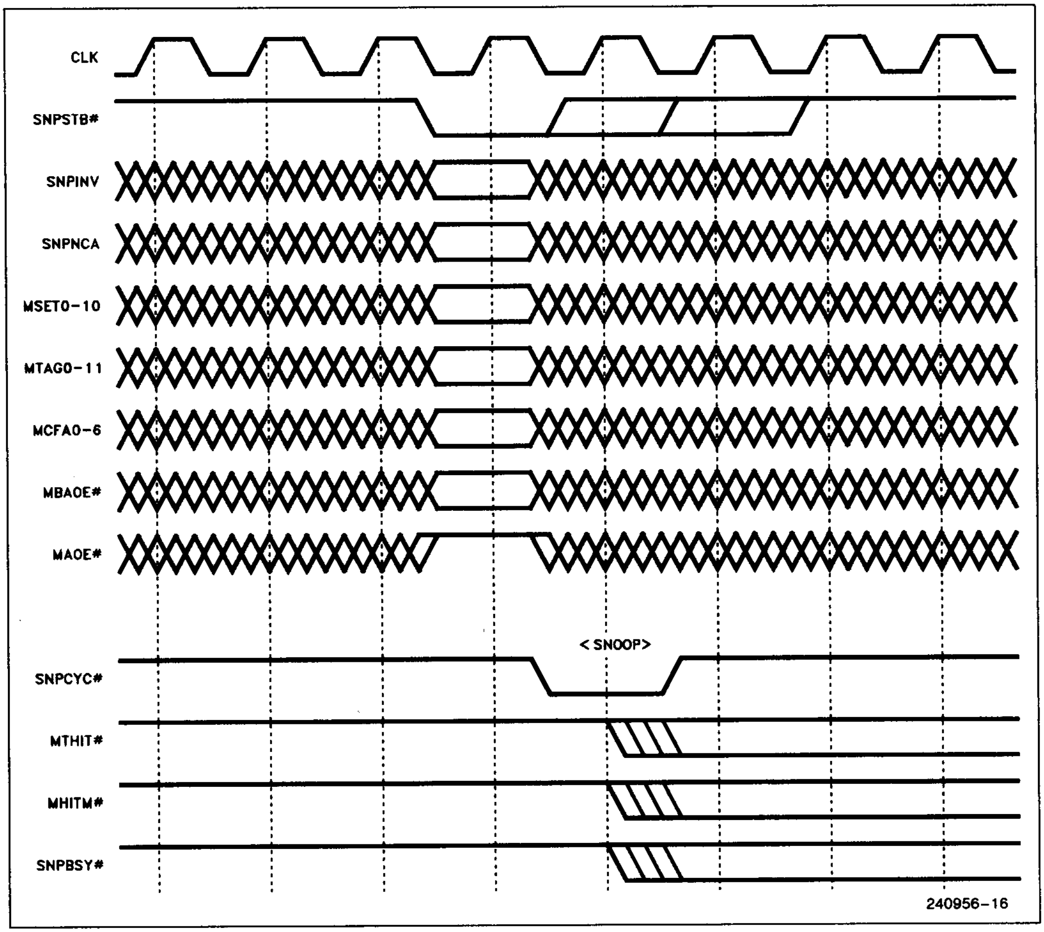


Figure 6-6. Synchronous Snoop Mode

6.2.2 RESPONSE PHASE

The snoop response phase consists of two parts: 1) 82495XP state indication 2) 82495XP snoop processing completion. The response phase is ALWAYS synchronous with the CPU CLK. The 82495XP state indication is presented on MHITM# and MTHIT# and remains stable until the next snoop. These signals indicate the state of the 82495XP line just prior to the snoop operation. The memory bus controller can predict the final state of the 82495XP line knowing the initial state and the SNPINV and SNPNCA inputs. The snoop completion information is determined by the SNPBSY# output. The SNPBSY# output inactive indicates that the 82495XP is ready to accept another snoop cycle.

Figure 6-7 shows the 82495XP response to snoops without invalidation. The first snoop is to a line which is not currently stored in the cache.

Figure 6-8 shows the 82495XP response to snoops with invalidation.

The SNPBSY# signal will be activated for one of two reasons: 1) a snoop hit to a modified line, SNPBSY# will remain active until the modified line has been written back. 2) a Back invalidation is needed and there is a back invalidation in process. The SNPBSY# minimum active time is two CLK periods. This allows an external logic to trap-hold active SNPBSY# using CLK. The external logic must first look for active SNPCYC# and then trap-hold SNPBSY#.

6.2.3 PIPELINED SNOOPS

The 82495XP allows the memory bus controller to pipeline snoop operations. The 82495XP allows the next snoop address to be supplied and the next snoop requested before the last snoop has completed.

There are a set of rules which govern the operation of pipelined snoops. These rules are as follows:

(1) For strobed mode snoops, the memory bus controller cannot cause a second falling edge of SNPSTB# until after the falling edge of SNPCYC#.

(2) For clocked mode snoops, the memory bus controller cannot cause a second falling edge of SNPSTB# to be sampled by SNPCLK, until after the falling edge of SNPCYC#.

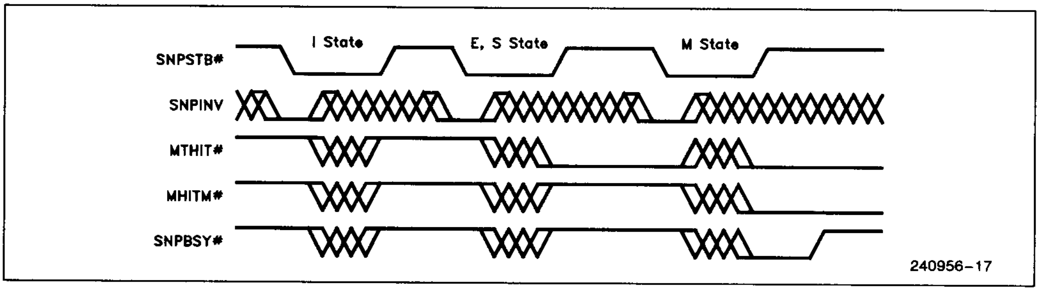


Figure 6-7. Snoops without Invalidation

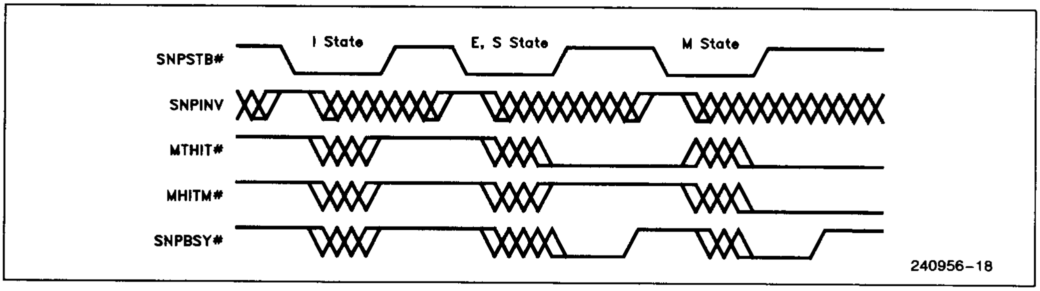


Figure 6-8. Snoops with Invalidation

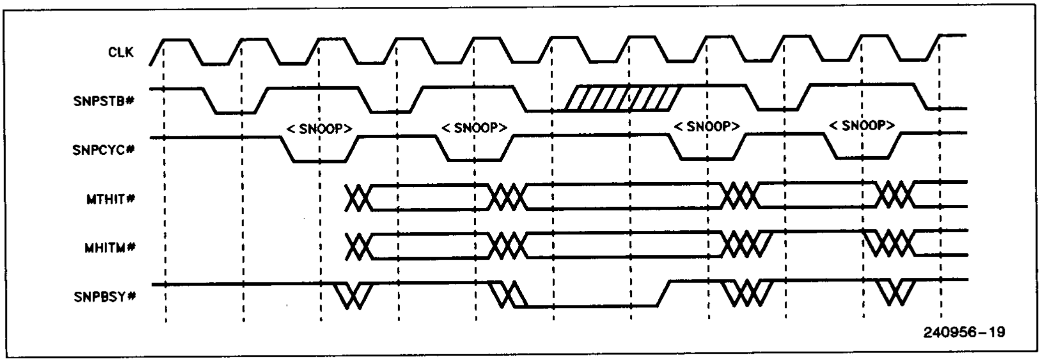


Figure 6-9. Fastest Possible Synchronous Snooping

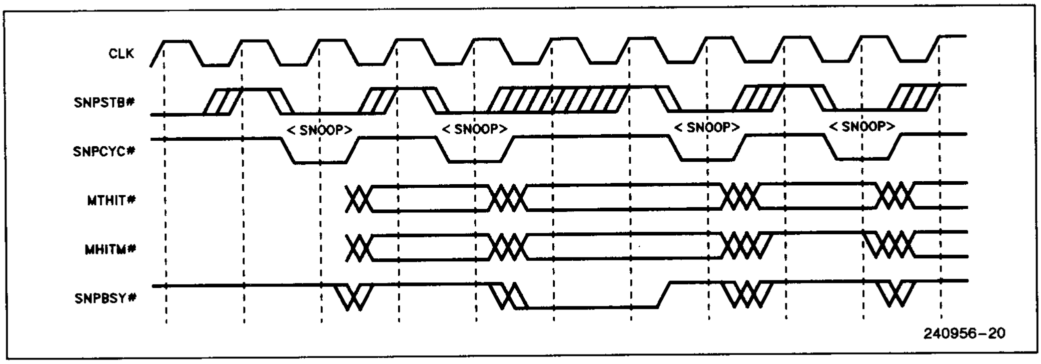
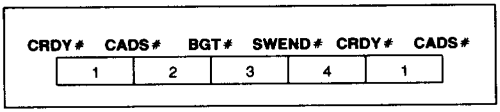


Figure 6-10. Fastest Possible Asynchronous Snooping

(3) For synchronous mode snoops, the memory bus controller cannot cause a second falling edge of SNPSTB# to be sampled by CLK, until the CLK after SNPCYC# is active.

6.2.4 OVERLAPPING SNOOPS WITH MEMORY BUS CYCLES

The 82495XP allows snoops to be overlapped with data transfers. The 82495XP divides the memory bus cycle into 4 main regions as shown below:



Region 1 is after a previous memory bus cycle (i.e. after CRDY#) and before the new memory bus cycle starts (before CADS#). A snoop in this region is looked up immediately and serviced immediately.

Region 2 is after a memory bus cycle has started (CADS#) but before the 82495XP has been granted the bus (BGT#). A snoop in this region is looked up immediately and serviced immediately. CADS# is re-issued for the aborted cycle once the snoop completes.

Region 3 is after the 82495XP has been granted the bus and before the SWEND# is completed. A snoop in this region has its lookup blocked until after the SWEND#. After SWEND#, the snoop response is given, but no write-back will be initiated until after CRDY#.

Region 4 is after SWEND# and before CRDY#. A snoop in this region is looked up immediately but serviced after CRDY#. This snoop is logically treated as if it occurred after CRDY# (snoop hits to modified data will schedule a write-back which will be executed after the conclusion of the current memory bus cycle). Note that the result of the snoop MHITM#, MTHIT# will be available immediately with the look-up.

6.2.5 SNOOP INTERLOCK

The 82495XP uses two interlock mechanisms to ensure that Snoops are identified within the proper region. The first interlock ensures that once a BGT# has been given snoops are blocked until after SWEND#. The second interlock ensures that once a snoop has been started BGT# cannot be given until after the snoop has been serviced.

Figure 6-11 shows how once the 82495XP sees a BGT# it blocks all snoops until after SWEND#. If a snoop has been initiated, and no SNPCYC# has been issued before BGT# assertion, the snoop has been blocked.

Figure 6-12 shows a snoop occurring before BGT#. Once the 82495XP has honored a snoop, the 82495XP, depending on the result of the snoop, may ignore BGT# until the snoop is serviced. The 82495XP will always ignore BGT# when SNPCYC# is active. If the snoop result is a hit to a modified line (MHITM# active), the 82495XP will ingore BGT# as long as both SNPBSY# and MHITM# remain active. In this case, it is the memory bus controller's responsibility to hold BGT# until SNPBSY# goes inactive or reassert it after SNPBSY# becomes inactive. If the snoop result is not a hit a modified line (MHITM# inactive), the 82495XP is capable of accepting BGT# even when SNPBSY# is active. This allows the memory bus controller to preceed with a memory bus cycle by asserting BGT# while the 82495XP is performing back-invalidations.

These two interlock mechanisms provide a flexible method of ensuring predictable handling of overlapped snoops.

NOTE:
Even when snoops are delayed, address latching is performed with SNPSTB# activation.

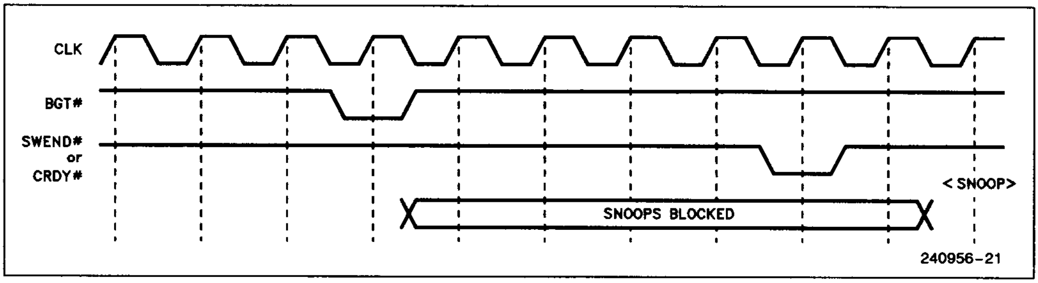


Figure 6-11. BGT# Blocking a Snoop

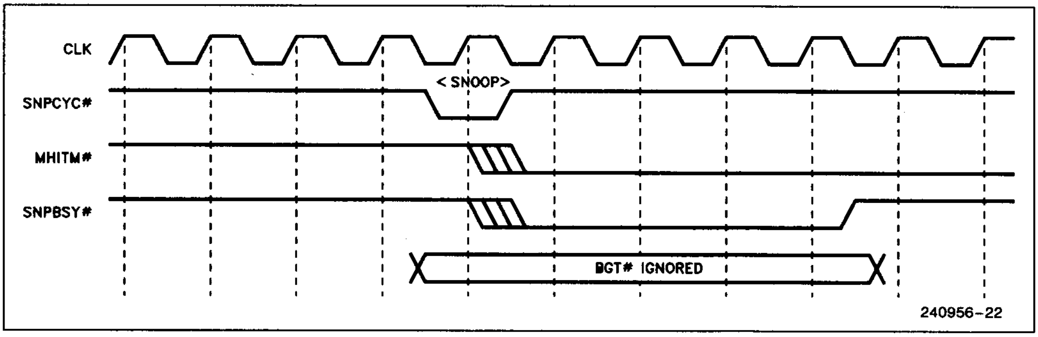


Figure 6-12. Snoop Occurring before BGT#

6.2.6 SNOOPS CONCURRENT WITH LINE FILL CYCLES

During snoops concurrent with line-fills/allocates, the following responsibility boundaries must be fullfilled in order to insure data consistency:

- If a snoop happens before BGT#, more precisely if SNPCYC# is active before BGT#, it is the system's responsibility not to return stale data within the line-fill/allocation.
- If a snoop happens after BGT#, more precisely if SNPCYC# is active after BGT#, then the 82495XP insures data consistency by providing interlocks with the CPU which avoid caching of stale data.

6.3 Memory Bus Controller Interface Rules

To begin a cache cycle, the 82495XP outputs the CADS# signal. The cache address and other cycle parameters are guaranteed to be stable with CADS# assertion. These parameters are guaranteed to be stable until CNA# or CRDY# of that cycle. After CNA# or CRDY# these parameters are undefined.

Either during, or after CADS# the CDTS# signal is asserted. Data is guaranteed to stable with CDTS# assertion, or the data path is available.

BGT# and CRDY# are required for all (non-snoop) cycles. KWEND# and SWEND# are only required for those cycles which sample them.

Once a signal has been sampled, it is a "don't care" until CRDY# of that cycle. Additionally, these signals plus the attributes MRO#, MKEN#, MWB/WT#, and DRCTM# need only follow setup and hold times when they are being sampled.

For pipelined cycles, the cycle attributes (BGT#, KWEND#, . . .) will only be sampled after CRDY# of the previous cycle.

Note that there are many other rules that govern when signals may be asserted in relation to one another. These may be found in the specific pin descriptions of each signal in chapter 7.

Snoop-Write-Back cycles are a subset of the normal cycles. Snoop-Write-Back cycles are requested as a consequence of snoop hits to Modified lines. Those are intervening cycles and are requested by activating SNPADS# instead of CADS#. For those cycles, the 82495XP only samples the CRDY# response. The 82495XP assumes that the memory bus controller owns the bus to perform the intervening write-back (restricted back-off protocol) and that no other agents will snoop this cycle. Also the 82495XP will ignore CNA# during Snoop-Write-Backs.

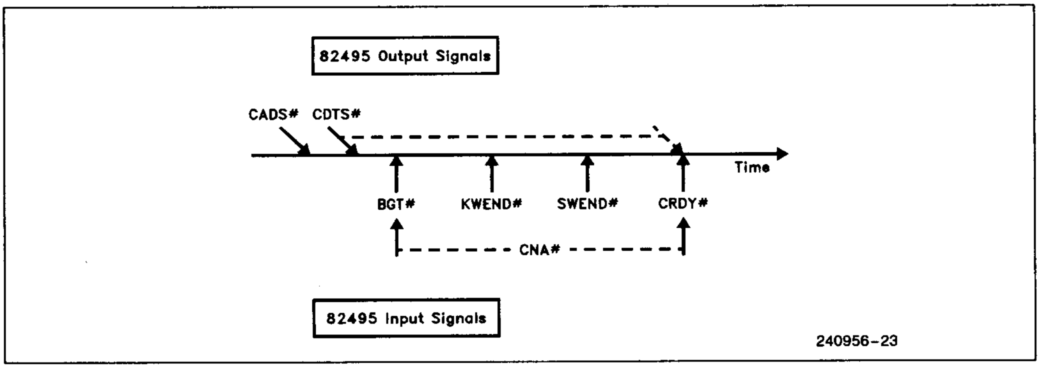


Figure 6-13. Cycle Progress

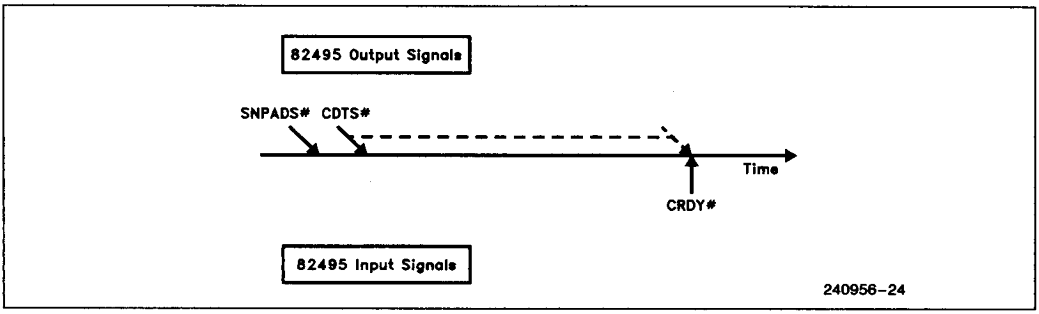


Figure 6-14. Cycle Progress for Snoop Cycles

6.4 LOCK# Protocol

The 82495XP provides a LOCK signal for the memory bus called KLOCK#. KLOCK# is generated by the 82495XP whenever the CPU generates the LOCK# signal. KLOCK#, like the other cycle attributes, is valid with CADS# assertion.

When the CPU generates a LOCK cycle, the 82495XP always generates a bus cycle. LOCK cycles are non-cacheable to both the 82495XP and CPU, so the information is passed through the 82490XPs to the CPU with BRDYs generated by the MBC. If the LOCKed read cycle is a hit in the 82495XP, the 82495XP ignores the data that it is receiving and supplies data from the 82490XP array (in accordance with the BRDYs supplied by the MBC). Locked writes are posted like any other write. LOCKed cycles, both reads and writes, never change the 82495XP tag state.

During a LOCKed cycle, the MBC must prevent other masters from snooping the 82495XP. Specifically, the MBC must prevent SNPSTB# between BGT# of the first LOCKed transfer, and SWEND# of the last LOCKed transfer.

6.4.1 SEMAPHORE CONSISTENCY

The 82495XP is optimized for high performance. In order to achieve this high performance the 82495XP overlaps back-invalidations of the primary cache with other activities. Normally this overlapping does not cause any problems. However, in systems where locked semaphores are used to insure mutual exclusion of processor access to shared data, special care is needed to insure that once the semaphore has been obtained, that all back-invalidation of the primary cache have been completed.

The 82495XP provides two write-ordering modes: weak write-ordering and strong write-ordering. In the weak write-ordering mode, the 82495XP does NOT guarantee that writes will be issued on the memory bus in the same order that they were issued by the processor. In the strong write-ordering mode the 82495XP, together with the CPU, guarantees that writes will be issued on the memory bus in exactly the same order as they were issued by the processor.

The weak write-ordering mode fits the weak-consistency model and relies on synchronization events visible to the hardware (locked cycles) to insure correct program execution. The strong write-ordering mode fits the Processor-Consistency model and insures data consistency for most applications.

The i860 XP CPU uses weak consistency during normal operations and strong consistency for LOCK# operations. Thus special care is needed to ensure that all data accessed within a CRITICAL REGION surrounded by a locked semaphore is strongly consistent. Therefore, the Memory Bus Controller (MBC) must obey an additional rule when handling locked read cycles. This additional rule is as follows:

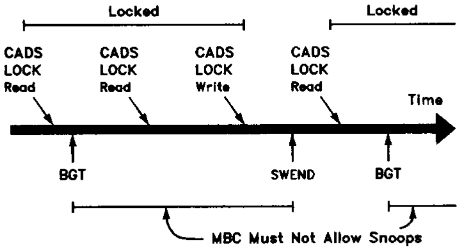
- The MBC must not provide the first BRDY# of a locked memory read until the C5 has finished all pending back-invalidations. The MBC must monitor CAHOLD to determine when all pending back-invalidations have completed.
- The first BRDY# of a locked memory read must be at least three CLKS after the last SNPCYC#, and CAHOLD must be inactive.

NOTE:
The 82495XP guarantees write-ordering to the memory bus, but write-ordering from the memory bus to the memory is a system responsibility.

6.5 Cycle Length

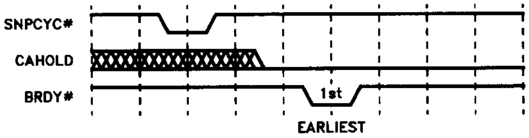
When CADS# is generated, the 82495XP outputs CW/R# and MCACHE#. These signals provide the MBC with enough information to determine the type of 82495XP cycle. Table 6-1 summarizes the cycle types for the 82495XP/82490XP. All line-fills and write-backs to the 82495XP/82490XP cache operate on the entire length of a cache line.

In addition to the length of the cycle from the 82495XP/82490XP, the memory bus controller may need to determine the length of the cycle to the CPU. Specifically, for those 82495XP cycles where RDYSRC=1, the MBC must decode the i860 XP CPU's W/R#, LEN, and CACHE# outputs to determine the number of BRDY#s which the MBC will provide to the CPU. These signals are captured for the current cycle by a user-provided BE latch (see Section 7.2 for details). Table 6-2 presents the CPU cycle length definitions; see the i860 XP microprocessor Data Sheet (Order #240874) for further details.



240956-25

Figure 6-15. Snooping During LOCKed Cycles



240956-61

Figure 6-16. BRDY# Timing for Semaphore Consistency Snooping During LOCKed Cycles

Table 6-1. 82495XP/82490XP Cycle Determination

Cycle Type	CW/R#	RDYSRC	MCACHE#	MKEN#
Posted Write	1	0	1	X
Write Backs	1	0	0	X
Non-Cacheable Read	0	1	1	X
Non-Cacheable Read	0	1	0	1
Cacheable Read	0	1	0	0
Allocation	0	0	0	X

Table 6-2. i860 XP CPU Cycle Determination

W/R#	LEN	CACHE#	MKEN#	Cycle Description	Burst Length
0	0	1	—	Non-Cacheable 64-Bit Read	1
0	0	—	1	Non-Cacheable 64-Bit Read	1
1	0	1	—	64-Bit Write	1
—	0	1	—	I/O and Special Cycles	1
0	1	1	—	Non-Cacheable 128-Bit Read	2
0	1	—	1	Non-Cacheable 128-Bit Read	2
1	1	1	—	128-Bit Write	2
0	—	0	0	Cache Line Fill	4
1	—	0	—	Cache Write-Back	4

NOTE:
If MRO# is asserted to the 82495XP, the effect on i860 XP CPU cycle determination is the same as when MKEN# = 1.

6.6 Consecutive Cycles

Because a 82495XP line can be longer than a CPU line, there are circumstances where a read miss will be to a line that is currently being filled. If this is the case, the 82495XP treats this like a read hit, but supplies data after CRDY# for the line fill. Data is supplied from the 82490XP array.

6.7 CPU/Memory Bus Concurrency

The 82495XP allows concurrency between the CPU and memory buses. CPU bus cycles will either be serviced locally by the 82495XP (hits) or require memory bus service. Whenever a CPU cycle requires memory bus service, it will be scheduled to run on the memory bus, and CPU bus activity will be allowed to continue.

Examples of concurrency are:

— Snoops and CPU bus operations

— Posted writes with CPU and memory bus operations

— CPU bus operation on the back of long line fills (82495XP line longer than the CPU line)

— Allocations and replacements with CPU and memory bus operations.

In certain cases, consistency of data and prevention of deadlocks preclude concurrency. Problems may occur when the current memory bus cycle changes the tag state and therefore affects the operation of the next CPU cycle request. In those cases the 82495XP will hold concurrency to ensure data consistency. Handling of those cases is completely transparent to the MBC.

The 82490XP supports two modes of memory bus operation: clocked mode and strobed mode. In clocked mode, memory bus signals are sampled by the 82490XP on rising edges of MCLK. Similarly, memory bus data and signals are output by the 82490XP with respect to MCLK (or MOCLK) rising edge transitions.

In strobed mode, memory bus signals are sampled or output with respect to rising and falling edges of other signals. Strobed mode has the advantage of not requiring setup and hold times to a CLK or MCLK edge.

6.8 Memory Bus Modes

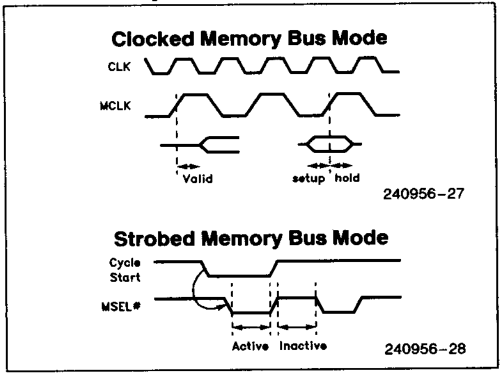


Figure 6-16a. Clocked and Strobed Mode Sampling

6.8.1 CLOCKED MODE

In clocked mode operation MCLK is used to reference the signals MDATA0-MDATA7, MSEL#, MFRZ#, MZBT#, MBRDY#, and MEOC#. Clocked mode will be selected if the 82490XP detects a clock at its MCLK input after RESET. MCLK need not have any relation to CLK. If this is the case, the memory bus is said to be operating in "clocked asynchronous" mode. If MCLK = CLK, the memory bus is operating in "clocked synchronous" mode. If MCLK × N = CLK (where N = 2, 3, 4 ...), the memory bus is operating in "clocked divided synchronous" mode. These three clocked modes, asynchronous, synchronous, and divided synchronous, are not differentiated by the 82490XP.

MOCLK controls a transparent latch at the 82490XP data output pins. If a clock is provided at this input, data is latched with MOCLK going low. This clock is available in clocked mode only. MOCLK allows the system to provide a greater MDATA hold time by skewing MOCLK from MCLK. If MOCLK is tied high, MDATA is driven from MCLK.

6.8.1.1 Synchronous Clocked Mode

In synchronous clocked mode MCLK = CLK. This means the CPU clock is used for 82495XP, 82490XP, and the memory bus. A synchronous memory bus allows memory to communicate with the 82495XP without synchronizers since the 82495XP runs with CLK. With a synchronous design, however, high clock frequencies must be routed to all parts of a system with minimal skew. As with any synchronous design, synchronous memory system and memory bus controller must be redesigned when future speed upgrades are required.

6.8.1.2 Asynchronous Clocked Mode

In asynchronous clocked mode, MCLK is not the same frequency as CLK. Some memory signals, since they reference MCLK, must be synchronized to CLK to communicate with the 82495XP. For example, when a cycle completes, the memory system asserts a signal, driven from MCLK, to the memory bus controller which will be synchronized to CLK to become CRDY#. This is because CRDY# is synchronous to CLK and not MCLK.

Asynchronous mode allows the rest of the system to run at a lower frequency than the CPU CLK. Not only does this simplify system design, but allows the designer to place hooks to allow the same design to scale easily to a higher frequency. If all the features of the 82495XP are used properly, an asynchronous memory design does not have to incur much synchronization penalty. For example, MEOC# is synchronous to the memory environment (MCLK). This allows the memory system to end the current cycle and start the next before CRDY# is synchronized in the CPU environment.

6.8.1.3 Divided Synchronous Clocked Mode

Divided synchronous clocked mode is a subset of synchronous clocked mode. It allows two things to happen: One, the memory system is capable of communicating with the 82495XP without synchronization. Two, a slower frequency clock may be routed around the system.

Divided synchronous mode still requires clock skew restrictions. It also carries the same scalability drawbacks that full synchronous mode does.

6.8.2 STROBED MODE

Strobed mode is configured on the 82490XP by strapping MCLK high. In strobed mode:

- — MDATA0-MDATA7 are sampled with respect to edges of MEOC#, MISTB, and MOSTB.
- — For write cycles, MFRZ# is sampled when MEOC# goes active.
- — MZBT# is sampled when MSEL# is inactive, and is latched when MSEL# goes active. MZBT# is also sampled for the next operation when MSEL# is active and MEOC# goes active.

6.9 Memory Bus Operation

All data is handled by the 82490XP cache RAMs. The 82495XP instructs the 82490XP whether to use the data array or buffers, and specifically which buffer to use. The MBC is responsible for bursting data in and out of the 82490XP's, in and out of the CPU during miss cycles, and indicating when the operation is finished. Communication between the 82490XPs and memory bus may be done in a clocked mode or strobed mode. See the Memory Bus Modes section for more details.

A 82490XP has 4 memory buffers. It has 2 memory cycle buffers, one write-back buffer, and one snoop buffer. Each buffer is capable of holding an entire 82495XP line of the longest configurable length.

The memory cycle buffers of the 82490XP are used for posting writes and holding data during 82495XP/82490XP line-fills. The write-back buffer is used for holding data from a cache replacement. This data is ready to be written out, and the write-back buffer is snoopable. The snoop buffer is used to hold modified data that has been hit by a snoop. Since snoop hits are the highest priority cycle, this buffer will be emptied before any other cycle or snoop request begins.

6.9.1 82490XP BUFFERS AND MUXES

The four 82490XP memory buffers are all multiplexed (muxed) to the memory bus. The mux is used to select which buffer is on the bus, and specifically which slice of that buffer is on the bus. MBRDY# assertion increments a counter for this mux which selects the next slice of that buffer.

The counter used to increment through the buffer slices is called the memory burst counter. The memory burst counter follows the CPU burst order depending on the subline address of the initial slice. Once the MBC is finished with a buffer, MEOC# is asserted to switch the mux to the next buffer to be used. MEOC# will also reset the counter and latch the last slice of data.

On the CPU side, the 82490XP contains a CPU buffer and mux. The CPU buffer captures data from the appropriate memory buffer or 82490XP array to feed it to the CPU. The mux selects which slice is muxed to the CPU bus. The counter for this mux is incremented with BRDY#.

The 82490XP array contains a mux that selects which way, based on the MRU algorithm, will be read during hit cycles. This mux is used during write cycles to write to the correct way.

6.9.2 MEMORY CYCLE BUFFERS

There are 2 memory cycle buffers in the 82490XP. They are used for line-fills, allocates, and memory writes. The buffers are 64-bits wide (per 82490XP) to support 8 transfers with 8 memory bus I/O pins (maximum configuration). The 82490XP alternates use of these buffers. When one buffer has a posted write or is being used for a memory read, the other one is available for the next cycle.

During allocation cycles, read for ownership may be implemented by using the MFRZ# signal. If MFRZ# is sampled active during the write cycle, the memory cycle buffer will freeze the write data in the buffer so the subsequent line-fill fills around it. This way the write cycle need not be written to memory. The line must be tagged as modified.

6.9.3 WRITE BACK BUFFER AND SNOOP BUFFER

The write back buffer and snoop buffer are both 64-bits to handle the maximum 82495XP line length. The write back buffer is used when replaced data must be written back to main memory (including FLUSH and SYNC cycles) and the snoop buffer is used when data must be written out from a snoop hit.

Before a line fill begins, the 82495XP checks to see if it must remove a modified line to make room for the new line. If so, the modified line is placed in the write back buffer and the new line is filled into a memory cycle buffer. Should the line-fill be selected as non-cacheable, both buffer contents are discarded and the 82490XP array value remains as it was before the line-fill.

There is no need to run the line-fill, replacement (write back), FLUSH, or SYNC cycles contiguously. If a snoop is requested between the two cycles, the write back buffer is snoopable, and data can be written directly out of it if need be.

6.9.4 MEMORY BUS CONTROL SIGNALS

The main memory bus control signals are MSEL#, MEOC#, MBRDY#, and CRDY#. These signals control the 82490XP data path, buffers, and muxes.

MSEL# selects which 82490XPs are being used in the current cycle by qualifying the MBRDY# signal. If MSEL# is inactive, MBRDY# is not recognized for that 82490XP. MSEL# is also used to reset the memory burst counter. If MSEL# goes inactive, the counter is initialized to its starting value. This is useful for aborted/restarted cycles. MSEL# may remain active for many or all cycles. MSEL# must, however, be inactive sometime after RESET to initialize the memory burst counter for the first time.

MEOC# is asserted by the MBC to end finish with the current buffer, and switch the memory bus to the next buffer to be used. MEOC# latches in the last piece of data and resets the memory burst counter before switching to the new buffer.

MBRDY# is used to increment the memory burst counter to select the next slice of data. This will strobe data out of the 82490XP (write cycles) or load data into the 82490XP (read cycles). MBRDY# is ignored by the 82490XP if MSEL# is inactive.

CRDY# finishes the current cycle. Once CRDY# is asserted, the 82490XP disposes of the information in the buffers used in that cycle, and loads information into the 82490XP array. CRDY# must be asserted on the clock or after MEOC# is asserted for a particular cycle.

6.9.4.1 Memory Burst Counter and Burst Latches

The burst order for replacements, allocates, and inquire cycles is always 0-1-2-3 for a 4 transfer bus and 0-1-2-3-4-5-6-7 for an 8 transfer bus. Write backs caused by snoop hits can be zero burst based (MZBT# sampled active), or start at the snoop subline address sent by the 82495XP. Memory reads, snoops and write throughs start at the address determined by the CPU and use the following burst orders:

4 Transfers	8 Transfers
0-1-2-3	0-1-2-3-4-5-6-7
1-0-3-2	1-0-3-2-5-4-7-6
2-3-0-1	2-3-0-1-6-7-4-5
3-2-1-0	3-2-1-0-7-6-5-4
	4-5-6-7-0-1-2-3
	5-4-7-6-1-0-3-2
	6-7-4-5-2-3-0-1
	7-6-5-4-3-2-1-0

6.9.5 82490XP DATA PATH

An example 82490XP read data path is shown in Figure 6-17. The path between the CPU and memory bus is a flow-thru' path, not a clocked path. Each entire 82495XP cache line of data in the CPU buffer is available at the memory buffer with some propagation delay. Likewise, each entire 82495XP cache line of data in the memory buffer is available in the CPU buffer with some propagation delay. Data is burst into and out of the memory buffer using MBRDY# or MISTB/MOSTB. Data is burst into and out of the CPU buffer using BRDY#. This means there is no synchronization required between memory and CPU data paths.

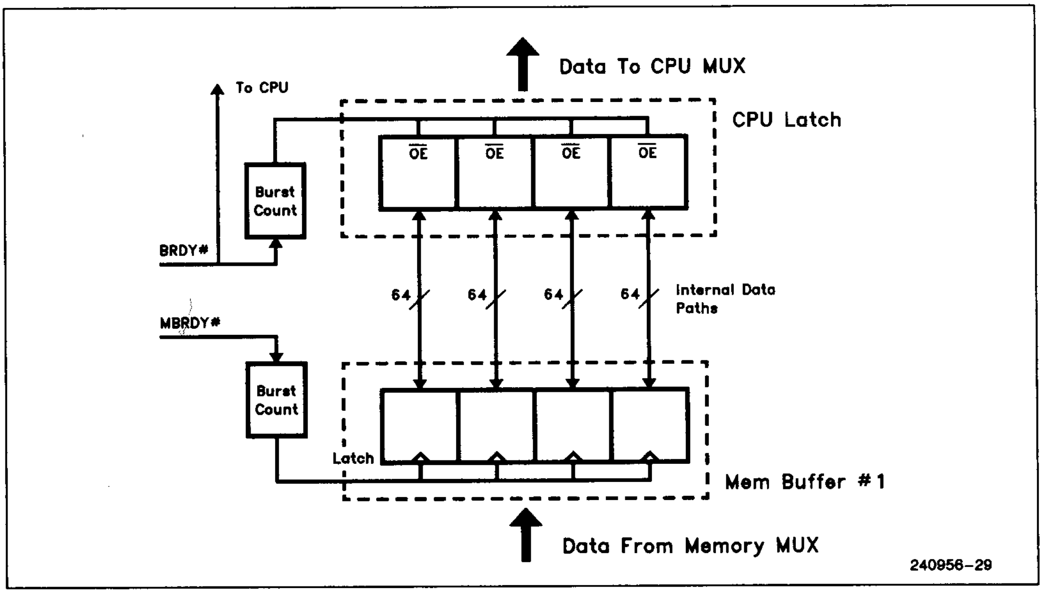


Figure 6-17. 82490XP Read Data Path

To give an example how the path works, during a CPU line fill, data may be returned to the CPU in two different fashions. One, each time the memory buffer fills a dword, BRDY# may be asserted a clock later to burst it back to the CPU. Two, the memory buffer can be filled and then BRDY# asserted on four consecutive clocks to burst data back to the CPU.

6.9.6 WRITE CYCLES

There are 3 basic types of write cycles: CPU generated write cycles, write back cycles caused by a cache replacement, and snoop write back cycles caused by a snoop hit. All write cycles, except the snoop write back, begin with CADS# assertion. The snoop write back cycle begins with SNPADS#.

6.9.6.1 CPU Generated Write Cycles

When the CPU begins a write cycle, four things can happen to it. One, the CPU write is a hit to a modified or exclusive line. In this case the write is terminated by the cache immediately and invisibly to the MBC.

Two, the write is to a shared location. This type of write is posted to the 82490XP memory cycle buffers and the cycle is terminated by the cache. If a memory cycle buffer is occupied with a write cycle, the CPU waits until the previous write completes. The write cycle must be written to the memory bus so that other copies of the write in other caches be invalidated.

Three, the write is a cache miss. This type of write is posted to a memory cycle buffer if the 82490XP is not waiting for another posted write to complete. If PALLC# is asserted, the write may be turned into an allocation.

Four, the write is a LOCKed write. LOCKed writes are posted regardless of the tag state. The write is then treated as if it were a miss except that there is no change in the tag state and no allocation allowed.

6.9.6.2 Cache Generated Write Cycles

The 82495XP/82490XP will generate a write cycle in three situations: a line fill or allocation causing a cache replacement, a snoop hit to a modified location, and write backs caused by FLUSH or SYNC. Write back caused by FLUSH or SYNC are indestinguishable from write-back cycles caused by replacement. Cache generated write cycles are the length of a cache line.

Cache replacements and FLUSH/SYNC cycles cause a line (or two lines if sectored) of cache data to be placed in the write-back buffer of the 82490XP. If no cycle is pending, CADS# is asserted and the data is written out. If a snoop hits the write-back buffer, the data is written out via SNPADS# like a normal snoop hit. The write back is then cancelled since the data was written through the snoop hit.

A snoop hit to a modified location causes a line of cache data to be written out to memory. Snoop hits are the highest priority cycle and must be serviced immediately. A snoop hit to a modified location causes the snooped line to be written to the snoop buffer of the 82490XP. SNPADS# is then asserted and the snoop is written out.

6.9.6.3 Memory Bus Controller Responsibility

The MBC recognizes a write cycle with CADS# and CW/R# (or SNPADS# for snoop cycles). If MCACHE# is active, the MBC knows the cycle is a write back cycle, otherwise it is a CPU-generated cycle.

CPU-generated write cycles are written to the main memory bus so that other caches can invalidate their copies of this information. The other caches do this by snooping with SNPINV active during snoop initiation if they detect a write cycle on the bus.

Once the MBC detects CDTS# active, the data will be available for writing in the next clock in the appropriate 82490XP buffer. The MBC should assert MSEL# so bursting is enabled, and burst through the write using MBRDY# (or MOSTB). MSEL# activation also caused MZBT# to be sampled. MZBT# must be inactive at this time if the data will be written according to CPU burst order.

Once the write cycle is complete, MEOC# must be asserted to end the write cycle and switch to the next pending cycle. If this write cycle is turned into an allocation, MFRZ# is sampled with MEOC# to freeze the write data in the 82490XP.

MEOC# simply switches buffers from the current one in use to the buffer of the next pending cycle. CRDY# needs to be asserted to actually end the cycle and allow the 82495XP and 82490XP to dispose of the information.

6.9.6.4 Write Allocation and Read for Ownership

The 82495XP/82490XP supports write allocation. An allocation cycle is a read of a cache line caused by a write miss to the same location. In its simplest form, a write miss is written to memory, then the 82495XP requests a line from that same location. Meanwhile, the CPU only sees the write cycle.

Write allocation may only be done if PALLC# is active during CADS# of the write cycle. For the allocation to occur, MKEN# must be returned active during KWEND# of the write cycle. The write cycle may be an actual write or a "dummy" write. Dummy writes are write cycles that are terminated in the 82495XP and 82490XP as if they were normal writes, but the data is not actually written to memory. This saves a data write to memory.

During write allocation, the write cycle will progress like a normal write cycle except MKEN# must be active during KWEND# activation. If the write cycle is a dummy write, MFRZ# must be used with MEOC# so that the line filled data is read around the write data into the 82490XP buffer. The line fill cycle is like any other line fill cycle except the CPU doesn't get any data. If a dummy write was performed, DRCTM# must be asserted during SWEND# activation to fill the line to the M state, and any cache supplying the data must invalidate its copy.

Using dummy write cycles and filling data to the M state from another cache or memory is called Read For Ownership. This is because ownership is being transferred. In read for ownership cycles, memory is avoided as much as possible. First, the dummy write cycle avoids memory. Second, a line fill is performed as a cache to cache transfer with DRCTM# asserted. All caches were snooped with invalidate to eliminate their copies.

For allocation cycles, SWEND# is not sampled for the write portion of the allocation.

6.9.7 READ CYCLES

The CPU initiates all read cycles. These are usually line fills to the CPU and line fills to the 82495XP/82490XP. The signal MCACHE# is output with CADS# to indicate whether this cycle may or may not be cacheable. If cacheable, MKEN# is returned by the MBC to ultimately determine cacheability.

Read hit cycles are serviced by the cache without MBC intervention. The only read cycles seen by the MBC (except I/O or special) are read misses and locked read cycles.

Read misses cause CADS# to be asserted at most two clocks after ADS# of the CPU read cycle. If cacheable, as determined from MCACHE#, the MBC will return 4 BRDYs back to the CPU and 4 or 8 MBRDYs to the 82495XP/82490XP. If the transfer is non-cacheable, the i860 XP CPU LEN and CACHE# outputs indicate the number of transfers to be given to the CPU. MBRDY# need not be used in the transfer if only a single piece of data is required by the CPU.

If the read cycle is cacheable, it may cause another cached line to be bumped out of the cache. This is called a replacement and, if modified, causes a write back cycle. While one of the 82490XP memory buffers is being filled for the line fill, the write back buffer is loaded. If the line fill turns out to be non-cacheable at the end of the transfer, the write-back buffer is discarded, and the line in the cache remains valid. Otherwise, CADS# will be generated after the read cycle so the write back can be performed. The write back need not happen immediately after the line fill since the write-back buffer is snoopable.

All locked reads go to the memory bus. If the read is a cache hit to M', the 82495XP/82490XP will ignore the data that the MBC returns, and provide it from its array. Locked reads are not cacheable by the CPU or the 82495XP/82490XP. Snoop write-backs that are a result of a LOCKed read/write request must update memory.

6.9.7.1 Memory Bus Controller Responsibility

Once the MBC sees a read cycle on the memory bus, it must determine whether the read is cacheable or non-cacheable using MCACHE# and its own address decoding. If non-cacheable, the CPU expects a number of transfers as determined by its LEN and CACHE# outputs. If cacheable, the CPU expects 4 transfers, and the cache expects 4 or 8 (configuration dependent).

MKEN# is sampled during KWEND# to determine cacheability. Before MKEN# is sampled, KEN# is active assuming cacheability for the CPU. MKEN# must be sampled 1 clock before the first BRDY# to make the cycle non-cacheable.

Once the read cycle is given to the memory system, all 82495XP/82490XP caches snoop to see if they contain the data in modified form. If so, the MBC must abort the cycle in memory and receive the data directly from the 82495XP/82490XP that has it, or wait until that cache writes it to memory. If the data transfer avoids memory, ie goes cache to cache, DRCTM# must be asserted with SWEND# to place the line in the M' state and the cache giving the data must invalidate its copy.

MSEL# is activated and MBRDY# (or MISTB) used to sample input data from the read cycle. Once CDTS# has been seen active, the CPU read data path is clear. BRDY# may be returned to the CPU sometime after each MBRDY# for each piece of input data (see MDATA setup to CLK). Once the transfer completes, MEOC# and CRDY# are asserted to complete the cycle in the 82495XP/82490XP.

6.9.8 I/O AND SPECIAL CYCLES

I/O and special cycles (flush, etc) are decoded by the 82495XP and not posted. These cycles wait until all buffers have been written, and all cycles have been completed, before they cause CADS# assertion. The CPU waits until the special cycle ends with the MBC's BRDY# assertion before it continues.

When the 82495XP/82490XP is performing a FLUSH or SYNC, many write back cycles are required. These cycles look like ordinary write back cycles, and should be handled as such. FSIOUT# is active during these write back cycles, so when FSIOUT# goes inactive the cycle is complete and the memory bus controller can supply BRDY# to the CPU.

6.10 Different Bus Widths

The 82490XP is capable of supporting either 64- or 128-bit memory bus widths. Depending on the configuration, the 82490XP's CPU and I/O busses may be multiplexed. The following diagram shows how an i860 XP CPU may be connected to a 128-bit memory bus:

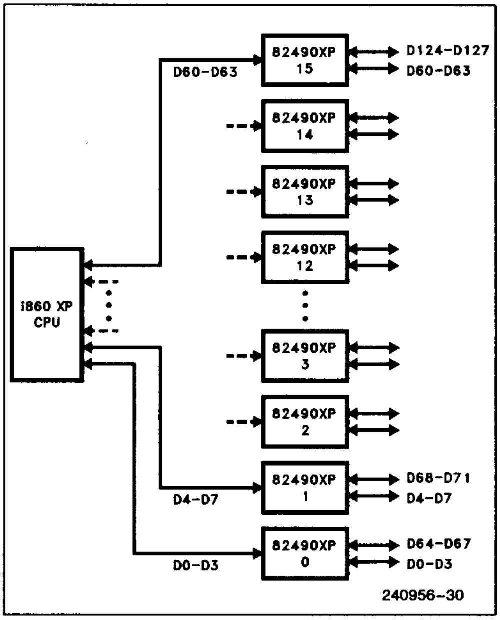


Figure 6-18. 82490XP On Wide Bus

In this example, the CPU port of the 82490XPs is in x4 mode and the memory bus port is in x8 mode. This allows all 128 bits of the memory bus to be multiplexed to the 64-bit CPU bus.

For read cycles, each MBRDY# loads 8 bits into each 82490XP. This is 128-bits of data. It will take 2 BRDY# assertions to load this into the CPU. The first BRDY# assertion loads the first 4 bits onto the CPU bus, and the next BRDY# assertion loads the remaining 4 bits.

For a 64-bit write cycle, the data is available at the on the appropriate data bits. On the i860 XP CPU with a 128-bit bus, this is determined by CPU address bit A3. The other data bits are undefined. For write-back cycles, all 128 bits are available at once. MBRDY# assertion will strobe the next 128 bits on the memory bus.

7.0 DETAILED PIN DESCRIPTIONS

The following chapter provides a detailed description of each pin of the 82495XP and 82490XP. The pins have been categorized by function. Each pin description has a heading which summarizes the most important aspects of the pin. The heading is organized as:

Pin Name

Name Meaning

Pin Function

I/O, 82495XP/82490XP/i860 XP CPU, (location) Signal Type

Synchronous/Asynchronous

for example,

CADS#

Cache Address Strobe

Indicates beginning of cache cycle

Output from 82495XP (pin E3) Cycle Control Signal

Synchronous to CLK

Following the heading are three sections. The first section, Signal Description, provides information of what the signal does, how to use it, and in what modes it operates. The second section, When Sampled or When Driven, indicates all the exact places where the part samples the signal, generates the signal, or neither. The third section, Relation to Other Signals, mentions the other signals that are affected by this signal, synchronization requirements, and shared pins.

All specific information about each pin is provided in this chapter.

7.0.1 CONFIGURATION SIGNALS

These signals are inputs to the 82495XP and 82490XP that are sampled at RESET and alter the configuration and operation of the cache.

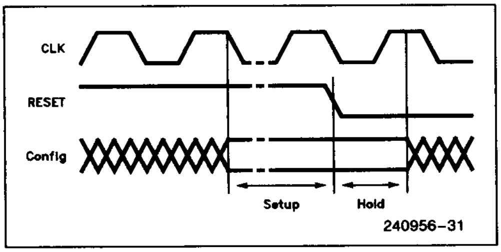


Figure 7-1. Configuration Input Setup and Hold

Each set of configuration inputs may have different setup times, but all signals have the same hold time: The signals may be released on the CPU clock edge that RESET is detected inactive. There are some configuration signals that are strapping options and cannot change their value during 82495XP operation.

7.0.2 CPU BUS INTERFACE SIGNALS

These pins comprise the interface between CPU and 82495XP/82490XP. The signals in this interface are not flexible; Chapter 10 addresses the use of these signals. The following are the CPU bus interface signals:

SET0–SET10	TAG0–TAG11	CFA0–CFA6
ADS#	W/R#	D/C#
M/IO#	HITM#	LOCK#
PWT	PCD	LEN
BRDYC1#	KEN#	AHOLD
EADS#	BE0–BE7#	INV
BOFF#		

The majority of these signals must be connected strictly between the i860 XP CPU and the 82495XP. However, a subset of these signals is needed by the MBC to decode the i860 XP CPU cycle in cases where the MBC provides BRDYs to the CPU. For these purposes the following signals must also be inputs to a latch controlled by the 82495XP's BLE# output:

BE0#-BE7#	CACHE#	CTYP
LEN	PCD	PCYC
PWT		

7.0.3 82495XP/82490XP INTERFACE SIGNALS

These pins comprise the interface between the 82495XP and 82490XP. The 82495XP uses these pins to control the 82490XP and its buffers. The signals in this interface are not flexible; Chapter 10 addresses the use of these signals. The following are the 82495XP/82490XP interface signals:

WRARR#	WAY	MAWEA#
BUS#	MCYC#	WBWE# [LR1]
WBA[SEC2]	WBTYP[LR0]	BRDYC2#
BLAST#	BOFF#	

SIGNAL DESCRIPTIONS

7.1 BGT#

Bus Guaranteed Transfer

Signals 82495XP of memory bus controller's commitment to complete the bus cycle.

Input to 82495XP (pin M4) Cycle Progress Signal

Synchronous

7.1.1 SIGNAL DESCRIPTION

The 82495XP owns all bus cycles (initiated by CADS#) until the memory bus controller accepts ownership. During this time cycles may be aborted due to a snoop. The memory bus controller signals its acceptance of ownership by driving BGT# active into the 82495XP. Once BGT# is driven active, the memory bus controller is responsible for completing the cycle on the memory bus. CRDY# signals completion of the cycle.

Once BGT# is asserted, other devices may not perform snoops into the 82495XP until the end of the snooping window, SWEND# activation. The snoop address is latched if SWEND is asserted between BGT# and SWEND#, but the snoop does not begin until after SWEND# is asserted. SNPCYC# will not be asserted until the snoop window ends with SWEND# asserted. The advantage of asserting BGT# early is that it allows the 82495XP to start inquiries to the CPU, load the write-back buffer, and progress forward in the CPU bus pipeline. The disadvantage is that snooping of this 82495XP is now blocked until SWEND# is asserted.

7.1.2 WHEN SAMPLED

After the 82495XP asserts CADS#, it begins sampling BGT# until it is sampled active.

BGT# is a "Don't Care" after it has been recognized for cycle N and prior to the assertion of

CADS# for cycle N+1. In addition, BGT# is a "Don't Care" once a cycle started by CADS# is aborted by a snoop, until the cycle is restored by the re-issueing of CADS#.

7.1.3 RELATION TO OTHER SIGNALS

When implementing BGT# in the MBC the following rules should be used:

1. BGT# must follow every assertion of CADS#, unless the cycle is aborted due to a snoop.
2. It must preceed CRDY# (for line fills and allocations BGT# must preceed CRDY# by at least 3 CLKS).
3. In addition BGT# must be asserted with or before the assertion of KWEND# and SWEND#.
4. BGT# must be asserted with or before the assertion of BRDY# by the MBC.
5. BGT# is not required following the assertion of SNPADS#.
6. BGT# must be asserted with or before MEOC# is asserted.

7.2 BLE#

BE Latch Enable

Controls latching of i860 XP CPU's byte enable and cycle attribute signals

Output of 82495XP (pin C16) Cycle Control Signal

Synchronous to CLK

7.2.1 SIGNAL DESCRIPTION

BLE# is used to control the enable line of an external latch (clock edge triggered '377 type). This latch is used to capture the i860 XP CPU's byte enables (BE0#-BE7#) and CPU cycle attribute signals which do not go through the 82495XP. The 82495XP manages the opening and closing of this latch: when BLE# is active, new values from the CPU enter the latch at each rising edge of CLK.

The 82495XP latches the byte enables after ADS# of a memory bus bound cycle. It relatches this information with CRDY# or CNA# of that cycle if another cycle is pending.

7.2.2 WHEN DRIVEN

The 82495XP latches the BE latch signals 1 clock after ADS# of a memory-bound cycle. Thus latched BE0#-BE7# are valid with CADS#. The 82495XP opens, then closes this latch if a cycle is pending and CNA# or CRDY# is asserted. Thus latched BE0#-BE7# are valid two clocks after CNA# or CRDY#, which is one clock AFTER CADS# for back-to-back cycles. The signals latched in the BE latch are only valid for CPU generated memory bus cycles (ie, not a 82495XP generated writeback or allocation).

7.2.3 RELATION TO OTHER SIGNALS

The following CPU signals must be latched in the BE latch:

BE0#-BE7#	CACHE#	CTYP
LEN	PCD	PCYC
PWT		

All other signals in the 82495XP to CPU interface (listed in sec. 7.0.2) must be connected only between the i860 XP CPU and the 82495XP.

7.3 BRDY#

Burst Ready

Memory Bus Controller Burst Ready input to 82495XP, 82490XP, and i860 XP CPU

Input to 82495XP and 82490XP (82495XP pin P1, 82490XP pin 60) Cycle Progress Signal

Input to i860 XP CPU (BRDY2#, pin U1)

Synchronous to CLK

7.3.1 SIGNAL DESCRIPTION

The BRDY# input to both the 82495XP and 82490XP must be connected to the BRDY# signal which the MBC is providing to the i860 XP CPU's BRDY2# pin. The signal is used by the 82495XP for burst tracking purposes. In the 82490XP, it increments the CPU latch burst counter.

During CPU read cycles, BRDY# allows the next 32 or 64-bit slice of read data to be available at the 82490XP's CDATA outputs (CPU bus) by advancing the CPU latch burst counter. At the same time, BRDY# is latching the previous slice of data into the i860 XP CPU. Refer to chapter 6 for more details.

During CPU write cycles, BRDY# is used to latch each slice of write data into the CPU latches and advance the latch counter.

During CPU special and I/O cycles (which are not posted) BRDY# is used to end the cycle.

BRDY# must not be asserted until the bus is granted (BGT# asserted) and until the data path is ready for transferring (CDTS# is asserted).

7.3.2 WHEN SAMPLED

BRDY# is sampled by the CPU, the 82495XP, and the 82490XP at every CLK edge. It must always meet proper setup and hold times to CLK. Even though the CPU latch may not be in use, BRDY# assertion will still advance the latch counter.

7.3.3 RELATION TO OTHER SIGNALS

BRDY# controls the CPU and 82490XP CPU latches. BRDY# has the following implication rules:

1. The last BRDY# for cycle N must be asserted 2 clocks before MEOC# for cycle N+1.
2. BRDY# ≥ BGT#
3. BRDY# > CDTS#

7.4 C490LDRV

82490XP Low Drive Buffer

Selects the 82495XP low capacitance driving buffers

Input to 82495XP (pin M3) Configuration Signal

Synchronous to CLK

7.4.1 SIGNAL DESCRIPTION

C490LDRV selects the driving strength of the 82495XP buffers that interface to the 82490XP. Refer to the layout specifications for information how C490LDRV should be connected.

7.4.2 WHEN SAMPLED

C490LDRV is a configuration input sampled like Figure 7-1. C490LDRV requires a setup time of 4 CPU clocks. After sampling, C490LDRV is a "don't care" until it is sampled as the BGT# pin after the first CADS# assertion.

7.4.3 RELATION OT OTHER SIGNALS

C490LDRV shares a pin with BGT#.

7.5 CADS#

Cache Address Strobe

Indicates beginning of cache cycle

Output from 82495XP (pin E3) Cycle Control Signal

Synchronous to CLK

7.5.1 SIGNAL DESCRIPTION

CADS# requests the execution of a memory bus cycle to the MBC, and indicates that the cycle attributes (ie. CD/C#, CM/IO#, CW/R#, PALLC#, etc.) are valid.

If the 82495XP receives a snoop hit to an [M] state line before BGT# is asserted by the MBC, the current CADS# is aborted and reissued after the snoop has completed. If the current line (issued by the stalled CADS#) is invalidated by the snoop, then that CADS# is cancelled (ie. will not be reissued after the snoop is completed).

CADS# is a glitch-free signal.

7.5.2 WHEN DRIVEN

CADS# is asserted by the 82495XP for exactly one CLK, and is always a valid logic level.

7.5.3 RELATION TO OTHER SIGNALS

CADS#, when asserted, indicates that the cache cycle control and attribute signals (ex. CD/C#, NENE#, CW/R#, etc.) are valid.

Since allocations do not require BRDY#s to the CPU, the CDTS# of an allocation cycle will always occur with CADS# of the allocation. In normal cycles the 82495XP will generate CADS# followed by CDTS#.

CADS# == CDTS# for all write-through cycles.

Once CADS# is active, PALLC#, CWAY, CDTS#, and BUS# are valid. Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS# active as well.

Every CADS# initiated cycle requires a BGT# and CRDY# input from the MBC.

CADS# and SNPADS# will never be asserted on the same CLK.

7.6 CAHOLD

82495XP AHOLD Output

Self-test result and AHOLD output status

Output of 82495XP (pin G4) Test Signal

Synchronous to CLK

7.6.1 SIGNAL DESCRIPTION

CAHOLD has two functions. One, it indicates the result of the built-in self-tests of the 82495XP. Two, it represents the 82495XP AHOLD into the i860 XP CPU.

The 82495XP drives CAHOLD after the 82495XP self-tests have completed. CAHOLD should be latched when FSIOUT# goes inactive after reset. If CAHOLD is high, the self-tests have passed, otherwise they have failed.

When the 82495XP drives AHOLD to the i860 XP CPU, it also drives CAHOLD, thus providing a means of tracking inquire cycles and back invalidations for performance monitoring.

7.6.2 WHEN DRIVEN

CAHOLD is always at a valid logic level. During self-test, CAHOLD is held until the clock edge that FSIOUT# is sampled inactive. After self-test, or reset, CAHOLD is asserted whenever the 82495XP asserts AHOLD.

7.6.3 RELATION TO OTHER SIGNALS

CAHOLD reflects the value of AHOLD except during self-test. During self-test, the value of CAHOLD should be latched with the falling edge of FSIOUT# to determine pass/fail.

7.7 CD/C#

Cache Data/Code

Indicates whether current cycle is Code or Data

Output from 82495XP (pin D3) Cycle Control Signal

Synchronous to CLK

7.7.1 SIGNAL DESCRIPTION

CD/C#, along with CW/R# and CM/IO#, is a 82495XP cycle definition signal. It indicates the type of bus cycle being requested of the MBC. CD/C# can be pipelined by the memory bus controller (by using the CNA# input to the 82495XP).

7.7.2 WHEN DRIVEN

CD/C# is valid in the same CLK as CADS# and remains valid until CRDY# or CNA#. C/DC# is always a valid logic level.

7.7.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.8 CDATA0–CDATA7

CPU Data Bus Connection

Data Bus Connection from 82490XP to CPU

Input/Output to 82490XP (pins 48, 54, 49, 55, 46, 51, 52, 57)

Isolated Interface

7.8.1 SIGNAL DESCRIPTION

CDATA0–7 is the 82490XP data bus connection to the CPU. All or part of these 8 pins will be used in connecting the 82490XP to the CPU depending on the cache configuration. See layout information for details.

7.9 CDTS#

Cache Data Strobe

Indicates availability of CPU data/data bus

Output from 82495XP (pin F4) Cycle Control Signal

Synchronous to CLK

7.9.1 SIGNAL DESCRIPTION

For read cycles, CDTS#, when asserted, indicates that in the next CPU clock the data bus path is available. This is the earliest time in which BRDY# may be supplied to the CPU. For CPU initiated write cycles, it indicates that the data is available on the memory bus. For i860 XP CPU inquire cycles, CDTS# informs the MBC that the last piece of inquire data is valid on the CPU bus.

Usage of this signal allows complete independence between address strobes (CADS# and SNPADS#) and data strobe. CDTS# allows the 82495XP to signal the MBC that a new cycle has begun as soon as addresses are available. This allows memory bus cycles to start before data is ready to be given/taken.

CDTS# is a glitch-free signal.

7.9.2 WHEN DRIVEN

CDTS# is asserted for one CLK, at the same time or later than CADS# for any given cycle.

7.9.3 RELATION TO OTHER SIGNALS

When the MBC samples CDTS# asserted, it can begin providing BRDY#s for the read cycle to the CPU in the next CLK. CDTS# must always be asserted before CRDY# and must be asserted prior to the first BRDY#.

The CDTS# of an allocation will always occur with CADS# of the allocation. In normal cycles the 82495XP will generate CDTS# following CADS#.

CDTS# will be asserted at least one CLK after SNPADS#.

7.10 CFG0–CFG2

Configuration Pins

Determine Cache Characteristics

Input to 82495XP (pins L4, Q1, M4,) Configuration Signals

Synchronous to CLK

7.10.1 SIGNAL DESCRIPTION

CFG0–CFG2 are the 3 cache configuration inputs that determine cache characteristics such as line ratio, tag size, and lines per sector. During RESET, this information is passed on to the 82490XPs. The following table maps CFG0–CFG2 to their respective configurations for the i860 XP CPU:

Config No.	Line Ratio	Lines/ Sector	No. of Tags	CFG2	CFG1	CFG0
1	1	1	8K	0	0	1
2	2	1	4K	1	1	1
3	1	2	8K	0	0	0
4	2	1	8K	0	1	1
5	4	1	4K	1	1	0

7.10.2 WHEN SAMPLED

CFG0–CFG2 are sampled like Figure 7-1 with a setup time of at least 10 CPU clocks. After sampling, CFG0, CFG1, and CFG2 become cycle progress input signals to the 82495XP and are sampled after CADS# of the first cycle.

7.10.3 RELATION TO OTHER SIGNALS

CFG0 shares a pin with CNA#, CFG1 shares a pin with SWEND#, and CFG2 shares a pin with KWEND#.

7.11 CLK

i860 XP CPU, 82495XP, 82490XP Clock

Input to the 82495XP (D11)

7.11.1 SIGNAL DESCRIPTION

The CLK input determines the execution rate and timing of the 82495XP, 82490XP, and CPU. Pin timings are specified relative to the rising edge of this signal. The i860 XP CPU, 82495XP, and 82490XP requires TTL levels on CLK for proper operation.

7.12 CM/IO#

Cache Memory/IO

Indicates whether current cycle is Memory or IO

Output from 82495XP (D4) Cycle Control Signal

Synchronous to CLK

7.12.1 SIGNAL DESCRIPTION

CM/IO#, along with CW/R# and CD/C#, is a 82495XP cycle definition signal. It indicates the type of bus cycle being requested of the MBC. CM/IO# can be pipelined by the memory bus controller (CNA# input to the 82495XP).

7.12.2 WHEN DRIVEN

CM/IO# is valid in the same CLK as CADS#, and remains active until CRDY# or CNA#.

7.12.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS# assertion.

7.13 CNA# [CFG0]

82495XP Next Address Enable

Dynamically pipelines CADS# cycles

Input to 82495XP (pin L4) Cycle Progress Signal

Synchronous to CLK

7.13.1 SIGNAL DESCRIPTION

CNA# is used by the MBC to dynamically pipeline CADS# cycles. When active it indicates to the 82495XP that the next MBC request can be started. Only one level of pipelining is allowed in the 82495XP.

CNA# is an optional input for all cycles initiated with CADS#.

7.13.2 WHEN SAMPLED

CNA# is sampled with the later of BGT# and CDTS#. If CDTS# was given for a cycle, CNA# will be sampled from the clock of BGT# until CRDY# is sampled active. If BGT# is given before CDTS#, CNA# will be sampled from the clock of CDTS# until CRDY# is sampled active. Once CNA# is sampled active, or once CRDY# has been given for a cycle, CNA# is ignored until the later of BGT# or CDTS# for the next cycle.

CNA# is ignored during snoop write-back cycles.

7.13.3 RELATION TO OTHER SIGNALS

Once the 82495XP samples this signal active, it issues the CADS# for the next memory bus cycle as soon as one begins.

7.14 CRDY#

Cache Ready

Ends a cycle in the 82495XP/82490XP

Input to 82495XP and 82490XP (pins M2, 43) Cycle Progress Signal

Synchronous to CLK

7.14.1 SIGNAL DESCRIPTION

CRDY# is used by the 82495XP and 82490XP to end a memory bus cycle. CRDY# indicates full completion of the cycle and allows the 82495XP/82490XP to free internal resources for the next cycle. In the 82490XP, this means that the current memory buffer in use is emptied (put in array, discarded, etc). In the 82495XP, CRDY# assertion allows 82495XP cycle progress signals (BGT#, KWEND#, SWEND#) to be sampled for the next cycle if pipelining is used.

CRDY# is required for all 82495XP/82490XP memory bus cycles, including snoop cycles. CRDY# must be asserted to the 82495XP and 82490XP at the same time.

7.14.2 WHEN SAMPLED

CRDY# for a given cycle is ignored until KWEND# is returned for that cycle. If KWEND# is not required for the cycle, CRDY# is ignored until BGT#. When CRDY# is ignored, it may violate setup and hold times.

7.14.3 RELATION TO OTHER SIGNALS

CRDY# must be sampled by the 82495XP and 82490XP at the same time. For the 82495XP, CRDY# has many cycle implication rules:

1. CRDY# > CDTS#
2. CRDY# > BGT#
3. CRDY# > BGT# + 2 clocks if cycle is a line-fill or allocation
4. CRDY# > KWEND# if cycle is a line-fill or write-through with potential allocation (PALLC# = 0)

For the 82490XP, CRDY# has three basic rules:

1. MEOC# for cycle N must be sampled with or before CRDY# for cycle N.
2. MEOC# for cycle N+1 must be sampled at least 2 CPU clocks after CRDY# for cycle N.
3. CRDY# for cycle N+1 must be after the last BRDY# for cycle N.

MBRDY# fills the current 82490XP memory buffer. CRDY# emties this buffer and makes it available for new cycles. CRDY# may be asserted on the same clock as MEOC# which may be asserted on the same clock as MBRDY#.

CRDY# shares a pin with SLFTST#.

7.15 CWAY

Cache Way

Indicates WAY used by the current cycle

Output from 82495XP (pin J3) Cycle Control Signal

Synchronous to CLK

7.15.1 SIGNAL DESCRIPTION

CWAY is a cycle definition signal which indicates to the MBC the WAY used by the requested cycle. On line-fills it indicates the way the line will be loaded. For write-hits (to [S] state or LOCKed) it indicates the way which was a hit. For write-backs it indicates the way that was written-back.

CWAY is utilized by external tracking machines in order for the 82495XP tags to be accurately duplicated.

7.15.2 WHEN DRIVEN

CWAY is valid together with CADS# and remains valid until CRDY# or CNA#.

7.15.3 RELATION TO OTHER SIGNALS

CWAY is valid with CADS#.

7.16 CW/R#

Cache Write/Read

Indicates whether current cycle is write or read

Output from 82495XP (pin E4) Cycle Control Signal

Synchronous to CLK

7.16.1 SIGNAL DESCRIPTION

CW/R#, along with CD/C# and CM/IO#, is a 82495XP cycle definition signal. It indicates the type of bus cycle being requested of the MBC. CW/R# can be pipelined by the memory bus controller (CNA# input to the 82495XP).

7.16.2 WHEN DRIVEN

CW/R# is valid in the same CLK as CADS# and is valid until CRDY# or CNA#.

7.16.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.17 DRCTM#

Memory Bus Direct to [M] State

Signals 82495XP to tag data direct to the [M] state, skipping the [E] and [S] states.

Input to the 82495XP (pin M1) Cycle Attribute Signal

Synchronous to CLK

7.17.1 SIGNAL DESCRIPTION

DRCTM# is an input to the 82495XP from the memory bus. When sampled active at the end of the snooping window (SWEND# activation), the 82495XP moves the line fill in progress directly to the [M] state.

There are three cases in which this is useful.

1. Simplifies External State Tracker

   External trackers can only track the [M], [S], and [I] states. The [E] state can not be tracked externally since cache write hits internally change [E] state lines to [M] state. DRCTM# can be used to eliminate the [E] state from the MESI protocol.

2. Read For Ownership

   During a write miss with allocation the write may go to the memory buffer and not be written to memory. A read from memory, in conjunction with the MFRZ# signal asserted, reads the data to fill around the bytes written by the CPU. The contents of the memory buffer are then entered into the cache. The cache would normally tag this data in the [E] state (The cache assumes the write went to main memory). The system has the option of never completing the write to memroy (increases performance by completing the allocation quicker). If the write is not performed to memory, the cache is the only owner of the new data and therefore the cache entry must be tagged to the [M] state.

3. Cache to Cache Transfer

   A cache to cache transfer may occur as a result of a snoop. For example, if CPU/Cache 1 performs a read from main memory and CPU/Cache 2 flags it as a snoop hit to an [M] state line. To expedite the transfer, the system may perform the writeback from CPU/Cache 2 directly to CPU/Cache 1, bypassing memory. CPU/Cache 1 assumes the write-back went to memory and would normally tag the line to the [S] state. Since the system did not perform the write to memory, the system should drive DRCTM# to force the line to the [M] state. In addition, the line should be invalidated in CPU/Cache 2 by driving SNPINV.

7.17.2 WHEN SAMPLED

DRCTM# is synchronous to CLK. It is only sampled when SWEND# is active (the end of the snooping window). When SWEND# is inactive DRCTM# is ignored and does not have to meet setup and hold times.

7.17.3 RELATION TO OTHER SIGNALS

DRCTM# (direct to [M]) and MWB/WT# (write policy) combine to define the memory bus attributes and are sampled on CLK at the end of the snooping window (SWEND# activation).

If MRO# is sampled active during KWEND#, DRCTM# is ignored.

7.18 FLUSH#

Flush

Causes a 82495XP Cache Flush

Input to 82495XP (N4) Cache Synchronization Signal

Asynchronous input

7.18.1 SIGNAL DESCRIPTION

This signal causes the 82495XP to flush all its modified lines to main memory. The flushing of modified lines require the 82495XP to perform back-invalidation and inquire cycles to the CPU. At the end of flush, the 82495XP tag array will be completely invalidated.

FLUSH# will invalidate the entire 82495XP tag array. It takes two clocks to look-up and invalidate a tag entry. The 82495XP will also invalidate tags in the CPU cache by running back-invalidation cycles. If the 82495XP tag state is modified, the 82495XP will run inquire cycles to the i860 XP CPU to see is the line is modified in its cache. If so, the i860 XP CPU will write back the line into the 82495XP write buffer. All modified 82495XP cache lines must be written to memory.

7.18.2 WHEN SAMPLED

FLUSH# can be asserted at any time. The 82495XP will complete all outstanding transactions on the CPU and memory bus before beginning the FLUSH# process. The memory bus controller does not have to prevent FLUSH# during locked cycles because the 82495XP will complete its locked transaction before the FLUSH# process will begin.

Once a FLUSH# operation has begun, the FLUSH# signal is ignored until the operation completes. If RESET is activated while the FLUSH# operation is in progress, the FLUSH# operation will be aborted and the RESET immediately executed.

FLUSH# is an asynchronous input. FLUSH# must have a pulse width of 2 CLK's in order to guarantee 82495XP recognition.

7.18.3 RELATION TO OTHER SIGNALS

To initiate a FLUSH#, the 82495XP will complete all pending cycles and prohibit the processor from issuing any further ADS#'s while the FLUSH# is in progress. The FSIOUT# output signal is used to indicate the start and end of the FLUSH# operation. It will become active when the FLUSH# signal is internally recognized (all outstanding cycles have completed) and will de-activate with the CRDY# of the last FLUSH# write-back.

The memory bus controller supplies BRDY# to the CPU once FSIOUT# has gone inactive and the FLUSH is complete. Once FLUSH# has begun, and FSIOUT# active, all CADS#'s and CRDY#'s correspond to write-backs caused by the FLUSH# operation.

The 82495XP can be snooped during FLUSH# cycles and the snooping protocols will be the same as that for any memory bus cycle.

7.19 FPFLD# [FPFLDEN]

External FIFO PFLD

Indicates PFLD cycle during external PFLD FIFO mode

Output of the 82495XP (J4) Cycle Control Signal

Sync to CLK

7.19.1 SIGNAL DESCRIPTION

During RESET, this pin functions as the FPFLDEN configuration signal. The 82495XP can be configured to decode the i860 XP microprocessor's PFLD cycles. The 82495XP supports 3 operational modes for PFLD cycle decoding, as defined by FPFLDEN and NCPFLD#:

Mode #1. PFLD cycles are cached in the 82495XP.

Mode #2. PFLD cycles are not cached in the 82495XP, without an external PFLD extension FIFO.

Mode #3. PFLD cycles not cached in the 82495XP, with an external PFLD extension FIFO.

Mode	FPFLDEN	NCPFLD#
1	0	1
2	0	0
3	1	1
Illegal Mode	1	0

If mode 3 has been selected, the 82495XP allows the PFLD pipeline to be extended with an external FIFO. After RESET, when this mode has been selected, the FPFLD output will indicate that the requested cycle is a PFLD cycle. See Section 5.2.5 for more details.

7.19.2 WHEN DRIVEN

FPFLDEN is sampled on RESET as in figure 7-1, with a setup time of 4 CPU clocks. In PFLD mode #3, the FPFLD# output is valid in the same CLK as CADS# and remains valid until CRDY# or CNA#.

7.19.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.20 FSIOUT#

Flush, Sync, Initialization Output

Indicates the start and end of the Flush,

Sync, and Initialization operations.

Output of the 82495XP (D1) Cache Synchronization Signal

Sync to CLK

7.20.1 SIGNAL DESCRIPTION

This signal indicates the start and the end of either a Flush, Sync, or Initialization (including self-test if requested) operation. These operations are mutually exclusive. This signal is activated when the 82495XP begins the operation and goes inactive upon completion of the operation.

7.20.2 WHEN DRIVEN

This signal will be asserted whenever a Flush, Sync, or Initialization operation is internally recognized by the 82495XP and is in progress.

7.20.3 RELATION TO OTHER SIGNALS

FSIOUT# active indicates that either Flush, Sync, or Initialization operation is in progress. Only one of these operations can be run within the 82495XP at a time.

The table below shows the priorities of these three operations:

Operation	Trigger	Priority
Initialization	RESET	Highest
Flush	FLUSH#	
Sync	SYNC#	Lowest

If a trigger of higher priority occurs while a lower priority operation is running, the lower priority operation is aborted and the higher priority one executed. If a trigger of lower priority occurs when a higher priority one is running, the lower priority trigger is ignored. Once a FLUSH# or SYNC# operation has begun, its trigger is ignored until the operation completes.

When a higher priority operation aborts a lower priority one, FSIOUT# remains active.

Since RESET, FLUSH# and SYNC# are all asynchronous, FSIOUT# will be activated when the 82495XP is actually internally executing the operation.

7.21 HIGHZ#

High Impedance Outputs

Causes 82495XP outputs to be tristated

Input to 82495XP (pin P4) Test Signal

Synchronous to CLK

7.21.1 SIGNAL DESCRIPTION

The 82495XP will enter self-test if both SLFTST# is active and HIGHZ# is inactive during reset. If SLFTST# is sampled active and HIGHZ# is sampled active during reset, the 82495XP floats all its outputs until the 82495XP is reset again. Activation of HIGHZ# without SLFTST# does nothing.

7.21.2 WHEN SAMPLED

HIGHZ# is sampled like figure 7-1 with a setup time of 10 CPU clocks. HIGHZ# is then a don't care until the 82495XP reset sequence is complete (with FSIOUT# going inactive) where it becomes the MBALE pin.

7.21.3 RELATION TO OTHER SIGNALS

HIGHZ# shares a pin with MBALE. 82495XP outputs are tristated if both HIGHZ# and SLFTST# are sampled active during reset.

7.22 KLOCK#

82495XP LOCK#

Request to MBC of LOCKed cycle

Output from 82495XP (pin C3) Cycle Control Signal

Synchronous to CLK

7.22.1 SIGNAL DESCRIPTION

KLOCK# indicates to the MBC that there is a request to execute a locked cycle. This signal follows the CPU lock request.

KLOCK# is simply a one-clock flow-through version of the CPU LOCK# signal. The 82495XP will activate KLOCK# with CADS# of the first cycle of a LOCKed operation and it will remain active until the CADS# of the last cycle of the LOCKed operation.

Note that if the memory bus is pipelined, there may be a situation in which KLOCK# deactivation is in the same CLK as its new activation (together with CADS#). In this case KLOCK# won't go inactive between back-to-back locked sequences. KLOCK# will never go inactive if the CPU LOCK# does not go inactive. The 82495XP will not open arbitration windows between back-to-back locked sequences; it is the memory bus controller's responsibility to implement this functionality by detecting a LOCKed write followed by a LOCKed read.

KLOCK# activation is not qualified by the tag array look-up (hit/miss indications); therefore, KLOCK# can be active before CADS# is asserted.

7.22.2 WHEN DRIVEN

KLOCK# assertion is a flow-through of 1 CLK from the CPU LOCK# after the 82495XP completes all pending cycles. KLOCK# deassertion is a flow-through of 1 CLK from the CPU LOCK# signal, and must be at least 1 CLK after the last CADS# of a LOCKed sequence. KLOCK# is always driven to a valid logic level.

7.22.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.23 KWEND#

Cacheability Window End

Closes 82495XP Cacheability Window

Input to 82495XP (pin M4) Cycle Progress Signal

Synchronous to CLK

7.23.1 SIGNAL DESCRIPTION

KWEND# is a cycle progress input to the 82495XP that, when active, closes the cacheability window and causes the cacheability attributes MKEN# and MRO# to be sampled.

KWEND# is sampled by the 82495XP after BGT# has been sampled active. KWEND# should be asserted by the MBC once the memory address has been decoded and cacheability (MKEN#) and read-only (MRO#) attributes have been determined.

The sampling of KWEND# active allows SWEND# to be sampled. Resolving KWEND# quickly allows the non-cacheable window between BGT# and SWEND# to be closed more quickly. KWEND# activation also allows the 82495XP to start allocations and begin replacements.

7.23.2 WHEN SAMPLED

KWEND# is sampled by the 82495XP on the clock, or after, BGT# has been sampled active. Once KWEND# is sampled active it is not sampled again until BGT# of the next cycle. KWEND# need not follow setup and hold times if it is not being sampled.

BGT#, KWEND# and SWEND# may be asserted on the same clock edge.

KWEND# need only be activated for those cycles which require the sampling of MKEN# and MRO#. These are line-fills and write cycles with potential allocation.

7.23.3 RELATION TO OTHER SIGNALS

KWEND# is sampled on or after BGT# and allows the sampling of SWEND#. KWEND# activation causes the sampling of MKEN# and MRO#.

According to cycle progress implication rules, CRDY# must be at least one clock after KWEND# for line fills and write-through cycles with potential allocate.

KWEND# shares a pin with CFG2.

7.24 MALE

Memory Address Latch Enable

Tristates/Enables Memory Address Outputs

Input to 82495XP (pin O2) Cycle Control Signal

Asynchronous

7.24.1 SIGNAL DESCRIPTION

The 82495XP contains an address latch which controls the last stage of the 82495XP address output. It is controlled by four signals: MAOE#, MBAOE#, MALE, and MBALE. The signals MALE and MBALE control the latching of the entire 82495XP address where MBALE controls the subline portion and MALE controls the rest.

MALE is provided so that the memory bus controller can control when the next pipelined address is driven. With MALE high, the 82495XP address latch is in 'flow-through' mode and the 82495XP address is available at the memory bus. Changes in the 82495XP address are seen immediately at the memory bus. When MALE is driven low the address at the latch input is latched. Any subsequent address driven by the 82495XP will not be seen at the memory bus outputs until MALE is driven high again.

MALE will latch 82495XP addresses regardless of the state of MAOE#. If MAOE# is inactive, MALE will still operate the latch properly, but the memory bus will be tristated.

7.24.2 WHEN SAMPLED

MALE is asynchronous and can be asserted and deasserted at any time. MALE should always be driven to a valid state since it directly controls the operation of the address latch.

7.24.3 RELATION TO OTHER SIGNALS

MALE together with MBALE control the latching of the entire 82495XP output address. The other latch control signals, MAOE# and MBAOE#, provide the memory bus controller complete command over the address outputs. MAOE# and MBAOE# do not affect the operation of MALE or MBALE.

MALE shares a pin with the WWOR# configuration pin.

7.25 MAOE#

Memory Address Output Enable

Tristates/Enables Memory Address Outputs

Input to 82495XP (pin S4) Cycle Control Signal

Asynchronous except during snoop cycles

7.25.1 SIGNAL DESCRIPTION

The 82495XP has an address latch which is controlled by a latch input, MALE, and an output enable input, MAOE#. MAOE# has two main functions. One, driving MAOE# active will enable the 82495XP to drive it's address lines MTAG0-11, MSET0-10, and MCFA0-6. Two, MAOE# is a qualifier for snoop cycles and must be inactive for the 82495XP to snoop.

In general, MAOE# should be active if its 82495XP is the current bus master. When that 82495XP gives up the bus, MAOE# should be inactive to float the address lines and allow another master to snoop.

MAOE# controls the output of the 82495XP address except the subline (burst) portion. This portion has a separate output control: MBAOE#.

7.25.2 WHEN SAMPLED

MAOE# is an asynchronous input (except during snoop cycles) and always has full control over the address output. For this reason, MAOE# must always be driven to a valid state.

The 82495XP does, however, sample MAOE# during snoop cycles. When sampled, MAOE# must meet proper setup and hold times. In synchronous snoop mode MAOE# is sampled on a CLK edge. In clocked mode MAOE# is sampled on a SNPCLK edge. In strobed mode MAOE# is sampled with the falling edge of SNPSTB#. If MAOE# is sampled active, the snoop will be ignored. This allows SNPSTB# to share a common line for multiple 82495XPs.

MAOE# need not meet any setup or hold time if it is not being sampled during a snoop cycle.

7.25.3 RELATION TO OTHER SIGNALS

MAOE# together with MBAOE# control the entire 82495XP address. Both signals are asynchronous and thus need never be synchronized to any clock. Both signals are, however, sampled during snoop cycles and require proper setup and hold times in these situations.

MALE and MAOE# together provide full control over the 82495XP address output latch.

7.26 MBALE

Memory Burst Address Latch Enable

Tristates/Enables Memory Burst Address Outputs

Input to 82495XP (pin P4) Cycle Control Signal

Asynchronous

7.26.1 SIGNAL DESCRIPTION

The 82495XP address latch is controlled by four signals: MAOE#, MBAOE#, MALE, and MBALE. The signals MALE and MBALE control the latching of the entire 82495XP address where MBALE controls the subline portion and MALE controls the rest.

MALE and MBALE are provided so that the memory bus controller has complete flexibility when the next address is driven. With MBALE high, the subline portion of the 82495XP address latch is in "flow-through" mode and the 82495XP subline address is available at the memory bus. Changes in the 82495XP subline address are seen immediately at the memory bus. When MBALE is driven low the subline address at the latch input is latched. Any subsequent subline address driven by the 82495XP will not be seen at the memory bus outputs until MBALE is driven high again.

MBALE will latch 82495XP addresses regardless of the state of MAOE# or MBAOE#. If MBAOE# is inactive, MBALE will still operate the latch properly, but the subline portion of the memory bus will be tristated.

Separate line and subline address latch controls are provided so that the latch outputs may be driven at different times. The table below indicates the subline address bits for each line size.

Line Size (Bytes)	Subline Address
32	A3, A4
64	A4, A5
128	A5, A6

7.26.2 WHEN SAMPLED

MBALE is asynchronous and can be asserted and deasserted at any time. MBALE should always be driven to a valid state since it directly controls the operation of the address latch.

7.26.3 RELATION TO OTHER SIGNALS

MALE together with MBALE control the latching of the entire 82495XP output address. The other latch control signals, MAOE# and MBAOE#, provide the memory bus controller complete command over the address outputs. MAOE# and MBAOE# do not affect the operation of MALE or MBALE.

MBALE shares a pin with the HIGHZ# configuration pin.

7.27 MBAOE#

Memory Burst Address Output Enable

Tristates/Enables Memory Subline Address Outputs

Input to 82495XP (pin P6) Cycle Control Signal

Asynchronous except during snoop cycles

7.27.1 SIGNAL DESCRIPTION

The 82495XP address latch is controlled by four signals: MAOE#, MBAOE#, MALE, and MBALE. MAOE# and MBAOE# are the output enables of this latch for the entire 82495XP address. Specifically, MBAOE# controls the subline address portion and MAOE# controls the rest.

MBAOE# has two functions. One, it can tristate the subline portion of the address separately from the rest of the address. Since the 82495XP does not sequence through burst addresses, the memory system may wish to provide the burst count. This requires that the 82495XP address burst portion be tristated after the first transfer. The Subline Address table appears in Section 7.26, MBALE.

Two, MBAOE# is sampled during snoop cycles. If MBAOE# is sampled inactive, the snoop write back cycle, if any, will begin at the subline address provided. If MBAOE# is sampled active, the snoop write back will begin at subline address 0. This allows snoop write backs to begin at the snooped subline address and progress through the normal burst order.

7.27.2 WHEN SAMPLED

Like MAOE#, MBAOE# is asynchronous except during snoop cycles and can be asserted or deasserted at any time. Since MBAOE# has direct control over the address latch, it must always be driven to a valid state.

MBAOE# is ,however, sampled during snoop cycles. In synchronous snooping mode, MBAOE# must meet proper setup and hold times to CLK's rising edge. In clocked mode, MBAOE# must meet setup and hold times to SNPCLK's rising edge. In strobed mode, MBAOE# must meet setup and hold times to SNPSTB#'s falling edge.

If MBAOE# is not being sampled for a snoop, ie. SNPSTB# is not asserted, MBAOE# need not meet any setup or hold time.

7.27.3 RELATION TO OTHER SIGNALS

MAOE# and MBAOE# control the entire 82495XP address output asynchronously. This address latch is completely controlled by MALE, MBALE, MAOE#, and MBAOE#.

MBAOE# is only sampled by the 82495XP during snoop cycles with SNPSTB#.

7.28 MBRDY#

Memory Burst Ready

Burst Ready input to 82490XP memory buffers

Input to 82490XP (pin 22) Cycle Progress Signal

Synchronous to MCLK

7.28.1 SIGNAL DESCRIPTION

When in clocked memory bus mode, MBRDY# (with MSEL# active) is used to advance the memory burst counter for the 82490XP buffer in use. This causes either new data to be latched from the memory bus (read cycle), or new data to be driven from the 82490XP buffer (write cycle). MBRDY# is sampled on all MCLK edges in which MSEL# is sampled active and has no relation to CLK. In strobed mode, MBRDY# must be tied high as MISTB/MOSTB strobes data in/out of the 82490XP.

For write cycles, the first piece of write data is available at the MDATA pins. MBRDY# assertion with MSEL# active causes the next 32, 64, or 128-bit slice of write data to be available. If only one slice is required, MSEL# and MBRDY# need never go active.

For read cycles, the first piece of read data flows through to the CPU. MBRDY# assertion with MSEL# active causes the next slice of memory data to be latched in the 82490XP buffer. BRDY# assertion will allow this data to be available on the CPU bus and latch it into the CPU. For cacheable cycles, MBRDY# needs to be asserted 4 or 8 times depending on the cache configuration.

7.28.2 WHEN SAMPLED

MBRDY# is sampled on all MCLK edges where MSEL# is sampled active. In this way MSEL# qualifies the MBRDY# input. If MSEL# is sampled inactive, MBRDY# need not follow setup and hold times to MCLK.

7.28.3 RELATION TO OTHER SIGNALS

MBRDY# is qualified by the MSEL# input. MBRDY# advances the memory burst counter for the 82490XP in use which either inputs or outputs data through MDATA.

MEOC# switches the 82490XP buffers to the next pending cycle, so the last MBRDY# must come before or on the clock of MEOC# assertion.

7.29 MCACHE#

82495XP Internal Cacheability

Indicates cycle cacheability attribute

Output from 82495XP (pin C2) Cycle Control Signal

Synchronous to CLK

7.29.1 SIGNAL DESCRIPTION

MCACHE# is driven by the 82495XP and indicates that the current cycle may be cached. Data cacheability is determined later in the cycle by MKEN# assertion. MCACHE# is asserted for allocation, replacement write-back cycles, and during cacheable read-miss cycles. (ie. read-miss cycles in which PCD is not asserted). It is not asserted for IO, special, or locked cycles.

Cycle Type	MCACHE#
Posted Writes	1
Write Backs	0
Read, PCD = 0	0
Read, PCD = 1	1
Allocation	0
I/O Cycles	1
Locked Cycles	1

7.29.2 WHEN DRIVEN

MCACHE# is valid in the same CLK as CADS# and remains valid until CRDY# or CNA#.

7.29.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0-MSET10, MTAG0-MTAG11, MCFA0-MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.30 MCFA0–MCFA6 MSET0–MSET10 MTAG0–MTAG11

MCFA0-MCFA6 Memory Configuration Address I/O

MSET0-MSET10 Memory Set Address I/O

MTAG0-MTAG11 Memory Tag Address I/O

82495XP Memory Address Inputs/Outputs

Input/Output of 82495XP (pins N14, P7-P15, O6-O16, R4, R14-R17, S14-S17) Cycle Control Signals

Input Synchronous to CLK, SNPCLK, or SNPSTB#.

Output from CLK, MAOE# active or MALE high.

7.30.1 SIGNAL DESCRIPTION

MSET0-10, MTAG0-11, and MCFA0-6 provide the complete 30 bit address input/output interface of the 82495XP to the memory bus. Together they span the entire CPU address range A2-A31. Depending on the cache configuration, each pin represents a different CPU address line (see configuration section for details).

MSET0-10, MTAG0-11, and MCFA0-6 pass through a 82495XP output latch. The latching of this latch is controlled by MALE/MBALE, and the output of this latch is controlled by MAOE#/MBAOE#.

With MAOE#/MBAOE# active, MSET/MTAG/MCFA are 82495XP outputs. They are valid at the start of a memory bus cycle at the input of the 82495XP address latch. If MALE/MBALE is high (flow-through) and MAOE#/MBAOE# is active (outputs enabled), they are driven to the memory bus with CADS#.

If a new cycle starts and MALE/MBALE is low, the previous address remains valid at the 82495XP MSET/MTAG/MCFA outputs. Once MALE/MBALE goes high, the new address flows through with the appropriate propagation delay (MSET/MTAG/MCFA address valid delay from MALE/MBALE going high). The new address will be driven to the 82495XP MSET/MTAG/MCFA outputs if MAOE#/MBAOE# is active.

If a new cycle starts, MALE/MBALE is high, and MAOE#/MBAOE# is inactive, the 82495XP MSET/MTAG/MCFA outputs will remain tristated. Once MAOE#/MBAOE# is asserted, the new address flows through with the appropriate propagation delay (MSET/MTAG/MCFA address valid from MAOE#/MBAOE# going active).

MSET0–10, MTAG0–11, and MCFA0–6 are used as inputs to the 82495XP during snoop cycles. Here, MAOE#/MBAOE# is inactive. MSET/MTAG/MCFA are sampled by the 82495XP during snoop initiation just like the other snoop attributes.

7.30.2 WHEN SAMPLED

If MALE/MBALE is high and MAOE#/MBAOE# is low, MSET0–10, MTAG0–11, and MCFA0–6 are valid with CADS# with a timing reference to CLK. Otherwise, they are asserted with a delay from MALE/MBALE high or MAOE#/MBAOE# active.

MSET0–10, MTAG0–11, and MCFA0–6 change once CNA# or CRDY# is sampled active. MSET0–10, MTAG0–11, and MCFA0–6 have a float delay from MAOE#/MBAOE# going inactive. These outputs are undefined after CRDY#/CNA# assertion and before the next CADS# assertion.

As inputs during snoop cycles (SNPSTB# asserted), they must be sampled like other snoop attributes with proper setup and hold times. In synchronous snoop mode this is with respect to CLK; in clocked mode, this is with respect to SNPCLK; and in strobed mode this is with respect to SNPSTB# falling edge.

If MAOE# is inactive and SNPSTB# is not asserted (no snoop), MSET0–10, MTAG0–11, and MCFA0–6 need not meet any setup or hold time.

7.30.3 RELATION TO OTHER SIGNALS

MSET0–10, MTAG0–11, and MCFA0–6 are asserted with CADS# so they are valid when CADS# is sampled active. This is true as long as MALE/MBALE is high and MAOE#/MBAOE# is active. If MSET0–10, MTAG0–11, and MCFA0–6 have been asserted but are blocked by MALE/MBALE or MAOE#/MBAOE#, they are asserted from MALE/MBALE going high or MAOE#/MBAOE# going active.

MSET0–10, MTAG0–11, and MCFA0–6 are deasserted or changed with CADS# or CNA# active. They may also be floated with MAOE# going inactive.

MSET0–10, MTAG0–11, and MCFA0–6 are used as inputs during snoop cycles. They are sampled with SNPSTB# like any other snoop attribute signal.

7.31 MCLK

Memory Bus Clock

Input to the 82490XP (Pin 26)

7.31.1 SIGNAL DESCRIPTION

In a clocked memory bus mode, this pin provides the memory bus clock. Memory bus signals and memory bus data are sampled on the rising edge of MCLK. Memory bus write data is driven off MCLK or MOCLK depending upon the configuration. MCLK has no relation to CLK.

7.31.3 RELATION TO OTHER SIGNALS

MCLK shares a pin with MISTB.

In clocked memory bus mode, the MDATA7–MDATA0, MSEL#, MFRZ#, MBRDY#, MZBT#, and MEOC# pins are sampled synchronously with the rising edge of MCLK. In a clocked memory bus write, MDATA7–MDATA0 are driven synchronous. with MCLK or MOCLK.

MOCLK is a delayed version of MCLK. If a clocked memory bus configuration is chosen, and the MOCLK rising edge is detected by the 82490XP after RESET, data will be driven off of MOCLK rather then MCLK. Only data is effected by MOCLK. MOCLK is used to allow the system designer to increase the minimum output time of MDATA relative to MCLK.

7.32 MDATA0–MDATA7

Memory Bus Data Pins

82490XP Connection to the Memory Bus

Input/Output of 82490XP (pins 18, 14, 10, 6, 16, 12, 8, 4)

Synchronous to CLK or MCLK or MOCLK or MISTB or MOSTB.

7.32.1 SIGNAL DESCRIPTION

MDATA0–7 is the 82490XP data bus connection to the memory bus. All or part of these pins will be used depending on the cache configuration. These pins are directly controlled by the MDOE# input. With MDOE# inactive, these pins are tristated and may be used as inputs.

For write cycles, the 82495XP asserts CDTS# to indicate that data will be available at the MDATA pins or in its buffer. Data is output with respect to CLK, MCLK, MOCLK, or MEOC# and is strobed with MBRDY#. In strobed memory bus mode, data is output using MOSTB.

For read cycles, CDTS# indicates that the CPU data path will be available for read data in the next clock. BRDY# reads data into the CPU from the 82490XP. Data is read into the 82490XPs through MDATA using MBRDY# or MISTB.

7.32.2 WHEN DRIVEN

When the CPU or 82495XP initiates a write cycle, the write data is written to the appropriate 82490XP buffer and CDTS# is asserted. If MDOE# is active, that first piece of write data will be available at the MDATA pins with some delay from the CPU CLK edge that CDTS# is asserted. Subsequent pieces of write data are output with some delay from MCLK or MOCLK (mode dependent) from the edge that MBRDY# is sampled active. In strobed mode, subsequent data is output with MOSTB assertion.

MDATA has no value before CDTS# assertion, after MEOC# with no pending cycle, or with MDOE# inactive.

For read cycles, the 82495XP asserts CDTS# the clock before the MDATA path is available for read data. MDOE# must be inactive for the 82490XP to read data. Read data is strobed into the 82490XP by asserting MBRDY# on MCLK edges. MEOC# will latch the last piece data as it switches buffers. In strobed mode, data is read by MISTB. Data that is read into MDATA must meet proper setup and hold times.

Data at the MDATA inputs need not follow setup and hold times to MCLK edges that sample MBRDY# inactive.

7.32.3 RELATION TO OTHER SIGNALS

CDTS# indicates that write data is in the 82490XP buffers. If MDOE# is active, write data is available at MDATA some time after CDTS# or MEOC# is sampled active. Subsequent write data is available at MDATA after MBRDY# assertion or MOSTB changing.

MDOE# must be inactive for MDATA to read data. CDTS# assertion by the 82495XP indicates that the read path is available in the next clock. Data must be read into MDATA with respect to MCLK or MISTB and must follow proper setup and hold times if MBRDY# is active or MISTB is changing.

The memory bus controller must account for the large setup time required to read data into the CPU. If properly done, data can be read into MDATA by asserting MBRDY# and in the next full CPU clock read into the CPU using BRDY#.

7.33 MDOE#

Memory Data Output Enable

Tristates/Enables Memory Data Outputs

Input to 82490XP (pin 20) Cycle Control Signal

Asynchronous

7.33.1 SIGNAL DESCRIPTION

MDOE# is an input to the 82490XP that, when asserted, causes the 82490XP to drive its MDATA0-MDATA7 outputs. When MDOE# is inactive, these lines are floated and may be used as inputs to the 82490XP. MDOE# is not sampled by any clock and is a direct connection to the 82490XP memory ouput driver.

7.33.2 WHEN SAMPLED

Since MDOE# is a direct connection to the 82490XP memory output drivers, MDOE# must always be driven to a valid level. With MDOE# inactive, data in the 82490XP's may be driven to MDATA outputs with some propagation delay from MDOE# going active. Similarly, there is some float delay from MDOE# going inactive.

MDOE# must be inactive for the 82490XP to read memory data.

7.33.3 RELATION TO OTHER SIGNALS

MDOE# has no relation to MCLK, MOCLK, or MOSTB. Since MDOE# controls the final stage of the MDATA output buffers, it has no effect on any other signal of the 82490XP.

7.34 MEMLDRV

Memory Low Capacitance Drivers

Selects the Low Capacitance Drivers for the 82495XP and the 82490XP

Inputs to 82495XP and 82490XP (pins Q4, 24) Configuration Signal

Synchronous to CLK

7.34.1 SIGNAL DESCRIPTION

MEMLDRV is a pin on both the 82495XP and 82490XP that, when high during reset, select normal driving memory output buffers. If this pin is driven low at reset, the high capacitance drivers are selected. Specifically, these are the 82495XP address outputs to the memory bus, and the 82490XP MDATA outputs. The normal output drivers are designed to drive up to 50 pF loads. The high capacitance drivers can drive up to 100 pF without derating.

7.34.2 WHEN SAMPLED

MEMLDRV is sampled like figure 7-1 with a setup time of 4 CPU clocks for the 82495XP and 1 CPU clock for the 82490XP. On the 82495XP, MEMLDRV becomes the SYNC# input once FSIOUT# goes inactive. On the 82490XP, MEMLDRV becomes the MFRZ# signal which is sampled after the first memory cycle begins.

7.34.3 RELATION TO OTHER SIGNALS

MEMLDRV shares a pin with SYNC# on the 82495XP and MFRZ# on the 82490XP.

7.35 MEOC#

Memory End of Cycle

Ends a cycle in 82490XP by switching buffers

Input to 82490XP (pin 23) Cycle Control Signal

Synchronous to MCLK or Asynchronous (strobed mode)

7.35.1 SIGNAL DESCRIPTIONS

MEOC# is an input to the 82490XP that ends the current cycle and switches memory buffers for new cycle. Switching to the next cycle does not cause information to be lost in the memory or CPU buffers in the 82490XP, but rather switches new buffers to the memory I/O bus of the 82490XP.

MEOC# is provided so that the memory system, which is synchronous to MCLK, can switch to a new cycle without synchronization. In clocked memory bus mode MEOC# is sampled with the rising edge of MCLK. In strobed memory bus mode the MEOC# function is performed with rising or falling edges of MEOC#.

For read or write cycles, MEOC# may be activated on or after the clock edge of the last MBRDY# of the current cycle. If a cycle is pending (pipelining is used), the next cycle will flow-through with a propagation delay from MEOC# assertion. MEOC# is required for all memory bus cycles.

In addition to switching memory buffers, MEOC# does three other things. One, MEOC# activation causes the memory burst counter to be reset to its start value and if MSEL# is active, MZBT# is sampled. This allows MSEL# to stay active between cycles. Two, MEOC# activation during a write cycle causes MFRZ# to be sampled for the a subsequent allocation (line-fill). Three, MEOC# latches in the last slice of data (like MBRDY#) before switching buffers.

7.35.2 WHEN SAMPLED

In clocked memory bus mode, MEOC# is sampled on every MCLK edge. It must always observe setup and hold times to MCLK. In strobed memory bus mode, MEOC# is always sampled and must meet proper active/inactive times.

7.35.3 RELATION TO OTHER SIGNALS

MEOC# is provided so that a cycle may end on the memory bus before CRDY# can be asserted. The implication rules surrounding MEOC# are:

1. MEOC# ≤ CRDY#
2. MEOC# for cycle N+1 ≥ 2 clocks after CRDY# of cycle N
3. MEOC# for cycle N+1 ≥ 2 clocks after last BRDY# of cycle N
4. MEOC# ≥ BGT#

MEOC# active with MSEL# active causes the sampling of MZBT# and MFRZ#.

7.36 MFRZ#

Memory Data Freeze

Freezes Memory Write Data in 82490XP Buffer

Input to 82490XP (pin 24) Cycle Control Signal

Synchronous to MCLK or Strobed

7.36.1 SIGNAL DESCRIPTION

MFRZ# is an input to the 82490XP that when active causes the 82490XP to "freeze" write data in the 82490XP memory buffer and allow a subsequent allocation to fill a cache line around it. MFRZ# is provided so that an actual write to memory need not be done to perform an allocation. Using MFRZ# to perform this dummy write cycle requires that the memory bus controller put the allocated line into the "M" state.

PALLC# must be active and MKEN# must be returned active for the write cycle to be turned into an allocation. MFRZ# is sampled when MEOC# goes active at the end of the write cycle. The subsequent line fill is then filled around the write data to complete the allocation.

7.36.2 WHEN SAMPLED

In clocked memory bus mode, MFRZ# is sampled with the MCLK rising edge that MEOC# is sampled active for all CPU write cycles. MFRZ# need only follow a proper setup and hold time in this situation.

In strobed mode, MFRZ# is sampled with the falling edge of MEOC# for write cycles. MFRZ# need only follow a proper setup and hold time in this situation.

7.36.3 RELATION TO OTHER SIGNALS

MFRZ# is sampled with the MEOC# going active or being active for write cycles. MFRZ# is used so that a dummy write cycle can be performed. If an allocation is done, DRCTM# must be asserted during the SWEND# window of the line fill to put the allocated line in the "M" state.

MFRZ# shares a pin with the MEMLDRV configuration input.

7.37 MHITM#

Memory Bus Hit [M]

Indicates snoop hit to modified line

Output from 82495XP (pin H4) Snooping Signal

Sync to CLK

7.37.1 SIGNAL DESCRIPTION

The MHITM# output is driven by the 82495XP during a snoop cycle to indicate that the snooping address has hit a Modified line. If the signal is logic high, the snoop has not hit a modified line; if the signal is logic low, the snoop has hit a modified line. When a snoop hits a modified line, the 82495XP automatically schedules a write-back of the hit modified line to the memory bus.

When the device which controls the memory bus (the master) performs a memory access, a snoop is requested of all other caching devices on the bus (snoopers). An asserted MHITM# pin from any of the snooper 82495XPs alerts the master that main memory's data is stale, and that the bus must be temporarily given to the snooper which has its MHITM# asserted so that the modified line can be written out to the memory bus.

7.37.2 WHEN DRIVEN

The snoop lookup is performed in the clock in which SNPCYC# is asserted. The MHITM# result for the snoop is driven on the CLK following SNPCYC#, and remains valid until the next assertion of SNPSTB#. The MHITM# signal is not valid from SNPSTB# until the CLK after SNPCYC#.

7.37.3 RELATION TO OTHER SIGNALS

MHITM# and MTHIT# outputs together indicate the results of a snoop lookup in the 82495XP.

A 82495XP can accept a snoop request while performing memory bus transfers of its own. If a snoop is requested of a 82495XP while it is performing a data transfer of its own, the results of the snoop may be delayed. If SNPSTB# is sampled at a 82495XP after it has received BGT# for its own cycle, the snoop lookup is performed (SNPCYC# active) after the SWEND# of its own cycle, and MHITM# is driven with valid results one CLK after SNPCYC# (see Sections 6.2.4 and 6.2.5).

7.38 MISTB

Memory Bus Input Strobe

Strobes data into the 82490XP

Input to 82490XP (pin 22) Cycle Control Signal

Asynchronous

7.38.1 SIGNAL DESCRIPTION

MISTB is an input to the 82490XP that, on rising or falling edges, causes the 82490XP to latch its MDATA inputs. MISTB is used in strobed memory bus mode. In clocked memory bus mode, MISTB is the MBRDY# input.

7.38.2 WHEN SAMPLED

MISTB is always sampled by the 82490XP. MISTB must meet proper strobed mode active and inactive times.

7.38.3 RELATION TO OTHER SIGNALS

MISTB causes the latching of the 82490XP MDATA inputs in strobed mode. MISTB shares a pin with MBRDY#.

7.39 MKEN#

Memory Cache Enable

Determines 82495XP and CPU cacheability

Input to 82495XP (pin R1) Cycle Attribute Signal

Synchronous to CLK

7.39.1 SIGNAL DESCRIPTION

MKEN# is an input to the 82495XP that is sampled at the closing of the cacheability window (KWEND# is sampled active). The 82495XP drives KEN# back to the CPU one clock after sampling the value of MKEN#. MKEN# thus determines whether the current cycle is cacheable in the 82495XP and in the CPU.

For read cycles, if MCACHE# is active (cacheable), KEN# is driven out of the 82495XP to the CPU to indicate cacheability. If MKEN# is sampled inactive during KWEND# activation, KEN# is brought inactive by the 82495XP, and the line will not be cacheable by the CPU or 82495XP. If MCACHE# is inactive, the line will be non-cacheable regardless of MKEN#. PCD active will cause MCACHE# to be inactive.

MKEN# is sampled during write-through cycles that are potentially allocatable (PALLC# is active during the write cycle). If MKEN# is sampled active during KWEND# activation of the write cycle, an allocation will occur, and a line-fill will follow the write cycle. MKEN# during the line-fill is ignored. The MBC indicates to the 82495XP that it intends to perform an allocation by asserting MKEN#.

MKEN# must be sampled 1 clock before the first BRDY# assertion to make a line-fill non-cacheable to the CPU.

7.39.2 WHEN SAMPLED

MKEN# is sampled on the clock edge that KWEND# is first sampled active. In all other places MKEN# may violate setup and hold times.

7.39.3 RELATION TO OTHER SIGNALS

MKEN# and MRO# are sampled with KWEND# active. MKEN# must be sampled at least 2 clocks before BRDY# assertion to make a line-fill non-cacheable.

7.40 MOCLK

Memory Data Output Clock

Separate Clock Reference for Memory Data Output

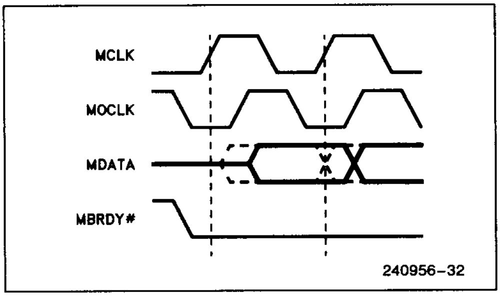
Input to 82490XP (pin 27)

Asynchronous

7.40.1 SIGNAL DESCRIPTION

MOCLK is the latch enable for the 82490XP memory data outputs (MDATA). MOCLK controls the latching of a transparent latch which, when high, causes MDATA to be driven from MCLK. When low, MDATA is latched. MOCLK may only be used in clocked memory bus mode and only affects output data. It is provided so that a greater MDATA output hold time can be generated.

To be used effectively, MOCLK must be a clock input that is skewed from MCLK. The following picture shows how MOCLK has increased the hold time of the output burst data:



7.40.2 WHEN SAMPLED

MOCLK is sampled during and after RESET to determine whether output data should be driven from MCLK or MOCLK. If toggling, MOCLK controls the MDATA outputs with MCLK. If high, data is driven from MCLK alone. Regardless, input data is never referenced to MOCLK.

In strobed memory bus mode the MOCLK signal becomes MOSTB. MOCLK is only used in clocked memory bus mode.

7.40.3 RELATION TO OTHER SIGNALS

To be used effectively, MOCLK must be the same frequency as MCLK but be skewed. This effectively increases MDATA hold time to main memory. Main memory must sample the data on MCLK edges.

MOCLK shares a pin with the MOSTB signal.

7.41 MOSTB

Memory Bus Output Strobe

Strobes data out of 82490XP

Input to 82490XP (pin 27) Cycle Control Signal

Asynchronous

7.41.1 SIGNAL DESCRIPTION

MOSTB is an input to the 82490XP that, on rising and falling edges, causes the 82490XP to output data through its MDATA outputs. MOSTB is only used in strobed memory bus mode. In clocked memory bus mode, MOSTB is the MOCLK input.

7.41.2 WHEN SAMPLED

MOSTB is always sampled by the 82490XP. MOSTB must meet strobed mode active and inactive times.

7.41.3 RELATION TO OTHER SIGNALS

MOSTB strobes data out of the 82490XP through MDATA. MOSTB shares a pin with MOCLK.

7.42 MRO#

Memory Read-Only

Designates current line as read-only

Input to 82495XP (pin J1) Cycle Attribute Signal

Synchronous to CLK

7.42.1 SIGNAL DESCRIPTION

MRO# is an input to the 82495XP that is sampled at the closing of the cacheability window (KWEND# activation). If sampled active, it causes the current line fill to the 82495XP to be put in the read-only state, and causes the line to be non-cacheable to the CPU. Writes to read-only lines in the 82495XP are treated as write-misses that are non-allocatable (PALLC# is inactive). MRO# is a bit in each 82495XP tag entry.

Once MRO# is sampled active during KWEND# activation, KEN# to the CPU is driven inactive regardless of the state of MKEN#. MKEN# does, however, determine whether the 82495XP will cache the read-only line. Once MRO# is returned active, the CPU will only require the number of transfers as indicated by LEN and CACHE#. If MKEN# is returned active, the 82495XP will require an entire cache line. 82495XP read-only cache lines are filled to the [S] state.

The line-fill portion of an allocation may be filled to the read-only state by returning MRO# active during KWEND# of the line-fill. MRO# is ignored during the write portion.

If MRO# is returned active during KWEND#, DRCTM# and MWB/WT# are ignored during SWEND#.

MRO# must be returned to the 82495XP at least 2 clocks before BRDY# is returned to the CPU so KEN# can be sampled properly.

There is one Read-Only bit per tag in the 82495XP.

7.42.2 WHEN SAMPLED

MRO# is sampled on the first clock that KWEND# is sampled active. In all other clocks, MRO# need not follow setup and hold times.

7.42.3 RELATION TO OTHER SIGNALS

MRO# and MKEN# are sampled with KWEND# activation. MRO# must be returned at least 2 clocks prior to the first BRDY#.

7.43 MSEL#

Memory Buffer Chip Select

Selects 82490XP, Causes Sampling of MZBT#

Input to 82490XP (pin 25) Cycle Control Signal

Synchronous to MCLK or Strobed

7.43.1 SIGNAL DESCRIPTION

MSEL# is an input to the 82490XP that has 3 main functions. One, MSEL# active qualifies the MBRDY# input to the 82490XP. If MSEL# is inactive for a particular 82490XP, MBRDY# will not be recognized by that 82490XP.

Two, MSEL# going active causes the sampling of MZBT# for the next transfer.

Three, MSEL# going inactive resets the 82490XP internal memory burst counter. The 82490XP contains a memory burst counter that counts through the CPU burst order with each MBRDY# assertion and increments a pointer to the 82490XP memory buffer being accessed.

MSEL# going inactive will reset this burst counter to its original burst value. By resetting this counter before MEOC# assertion, all information currently being read into the 82490XP is lost, but information that is being written out is maintained and may be rewritten.

In general, MSEL# may stay inactive for single transfer cycles such as posted 64-bit write cycles. Once active, MSEL# need not go inactive as the burst counter is reset with MEOC# activation. Since MZBT# may also be sampled with MEOC#, it is possible to leave MSEL# asserted throughout most basic transfers.

MSEL# or MEOC# must be used to reset the burst counter before any transfer begins. If transfers are interrupted (by a snoop hit before BGT# assertion for example), MSEL# must be brought inactive so the burst counter may be reset for the snoop write back.

MSEL# must be sampled inactive for at least 1 MCLK after reset. This resets the memory burst counter for the first transfer.

7.43.2 WHEN SAMPLED

In clocked memory bus mode, MSEL# is sampled with all rising edges of MCLK. In this mode, if MSEL# is sampled inactive, the memory burst counter is reset and MZBT# is sampled. If MSEL# is sampled active and MBRDY# is sampled active, the memory burst counter is incremented. Since it is constantly sampled with MCLK, MSEL# must always be driven to a known state and must always meet setup and hold times to every MCLK edge.

In strobed mode, MSEL# falling edge causes the sampling of MZBT#. While MSEL# is active, MISTB and MOSTB cause the memory burst counter to be incremented. The rising edge of MSEL# causes the memory burst counter to be reset.

MSEL# must be inactive sometime after RESET before the first transfer to initialize the burst counter.

7.43.3 RELATION TO OTHER SIGNALS

MSEL# causes the sampling of MZBT#, and qualifies the use of MBRDY#, MOSTB, and MISTB. Since MSEL# acts as a qualifier for these signals, MSEL# may be asserted at the same time as MBRDY#, MOSTB, or MISTB.

7.44 MTHIT#

Memory Bus Tag Hit

Indicates snoop hit

Output from 82495XP (pin G3) Snooping Signal

Sync to CLK

7.44.1 SIGNAL DESCRIPTION

The MTHIT# output is asserted by the 82495XP during snoop cycles to indicate that the snoop address has hit a line in the 82495XP cache. An asserted MTHIT# signal from any of the snooping 82495XP's alerts a bus master that the data being accessed resides in another cache. If SNPINV was not asserted on the snoop request, the copy of the data in a 82495XP asserting MTHIT# will remain valid and in the Shared state—so a caching master must also place his copy of the data in the Shared state.

7.44.2 WHEN DRIVEN

The snoop lookup is performed in the CLK in which SNPCYC# is asserted. The MTHIT# result for the snoop is driven on the next CLK and remains valid until the next assertion of SNPSTB#. The MTHIT# signal is not valid from SNPSTB# until the CLK after SNPCYC#.

7.44.3 RELATION TO OTHER SIGNALS

MTHIT# and MHITM# together indicate the results of a snoop lookup in the 82495XP.

An 82495XP can accept a snoop request while performing memory bus transfers of its own. If a snoop is requested while it is performing a transfer of its own, the results of the snoop may be delayed. If SNPSTB# is sampled at a 82495XP after it has received BGT# for its own cycle, the snoop lookup is performed (SNPCYC# active) after the SWEND# of its own cycle, and MTHIT# is driven with the valid result one CLK after SNPCYC# (see Sections 6.2.4 and 6.2.5).

Because an asserted MTHIT# from any snooping 82495XP requires the master to place the fetched line in the Shared state (unless it is an invalidating snoop), the memory bus controller should include the MTHIT# signals of other processors when generating the MWB/WT# signal to its own 82495XP.

7.45 MWB/WT#

Memory Write-back/Write-through

Forces lines to be filled to the [S] state

Input to 82495XP (pin K3) Cycle Attribute Signal

Synchronous to CLK

7.45.1 SIGNAL DESCRIPTION

MWB/WT# is an input to the 82495XP that is sampled at the closing of the snoop window (SWEND# activation). If sampled active, the current line-fill is filled to the [S] state in the 82495XP. The [S] state is a write-through state in the 82495XP.

MWB/WT# is used in many cases. If a cache to cache transfer updates memory and leaves the data valid in the other cache, the line must be filled to the [S] state instead of the [E] state default. A portion of memory may be designated as write-through by asserting MWB/WT# for appropriate addresses.

MWB/WT# has no effect on the 82495XP if DRCTM# is sampled active or MRO# has been sampled active during KWEND#. If PWT is active, MWB/WT# has no effect and the line is filled to the [S] state.

7.45.2 WHEN SAMPLED

MWB/WT# is sampled on the first clock edge that SWEND# is sampled active. If MWB/WT# is not being sampled, it need not follow setup and hold times.

7.45.3 RELATION TO OTHER SIGNALS

Both MWB/WT# and DRCTM# are sampled with SWEND#.

7.46 MX4/MX8# MTR4/MTR8#

Memory 4/8 I/O bits

Memory 4/8 Transfers

Selects MDATA Input/Output width and number of memory bus transfers

Inputs to 82490XP (pins 21, 25) Configuration Signals

Synchronous to CLK

7.46.1 SIGNAL DESCRIPTION

MX4/MX8# configures the 82490XP to use MDATA[0:3] or MDATA[0:7] memory bus I/O pins. MTR4/MTR8# selects whether the a cache line will take 4 or 8 transfers. These selections depend on the line ratio (82495XP line size / CPU line size) and must be configured according to the following table:

Line Ratio	MX4/ MX8#	MTR4/ MTR8 #	Membus I/O Pins	CPUbus I/O Pins
1	1	1	4	4
2	1	0	4	4
2	0	1	8	4
4	0	0	8	4
1	0	1	8	8
2	0	0	8	8

7.46.2 WHEN SAMPLED

These signals are sampled like Figure 7-1 with a setup time of 1 clock. Once the first CADS# is issued by the 82495XP these signals are sampled for the MZBT# and MSEL# functions.

7.46.3 RELATION TO OTHER SIGNALS

MX4/MX8# shares a pin with MZBT# and MTR4/MTR8# shares a pin with MSEL#.

7.47 MZBT#

Memory Zero Base Transfer

Forces cycles to begin at subline address 0

Input to 82490XP (pin 21) Cycle Control Signal

Synchronous to MCLK or Strobed

7.47.1 SIGNAL DESCRIPTION

MZBT# is an input to the 82490XP that forces a read or write cycle to begin with burst address 0 regardless of the CPU generated address.

MZBT# is sampled before the transfer begins. MZBT# is sampled with MSEL# and MEOC#. MZBT# is sampled with MSEL# going active for the current cycle. If MSEL# stays active between cycles, MZBT# is sampled with MEOC# going active for the previous cycle.

Once sampled, data input to the 82490XP's will start at burst address 0 and continue through 4, 8, C, etc. If the CPU is requesting a burst location other than 0, the memory bus controller must hold off any BRDY# until that bursted item is read from the memory bus.

7.47.2 WHEN SAMPLED

In clocked mode, MZBT# is sampled in two locations. First, MZBT# is sampled on all MCLK rising edges where MSEL# is sampled inactive. Once MSEL# is sampled active, the value of MZBT# that was sampled one MCLK before is used for the next transfer.

Second, MZBT# is sampled on MCLK rising edges where MEOC# is sampled active with MSEL# active. The MZBT# value sampled will be used for the next transfer. This allows MSEL# to stay asserted between transfers if so desired.

In strobed mode, MZBT# is sampled with the same two signals. First, it is sampled with the falling edge of MSEL#. Second, it is sampled with the falling edge of MEOC# if MSEL# is active.

In clocked memory bus mode MZBT# must follow setup and hold times to all MCLK edges where MSEL# is sampled inactive or MEOC# is sampled active with MSEL# active.

In strobed memory bus mode MZBT# must meet setup and hold times to MSEL# falling edge and MEOC# falling edge if MSEL# is active.

7.47.3 RELATION TO OTHER SIGNALS

MZBT# is sampled with MSEL# and MEOC# and has no effect otherwise. In systems that will never force a zero-based transfer, MZBT# may be driven high after RESET.

MZBT# shares a pin with the MX4/MX8# configuration input.

7.48 NCPFLD#

Non-Cacheable PFLD
Enables Non-Cacheable Floating Point Loads
Input to 82495XP (N4) Configuration Signal
Asychronous

7.48.1 SIGNAL DESCRIPTION

During RESET, this pin functions as the NCPLFD# configuration signal. The 82495XP can be configured to decode i860 XP CPU PFLD (Pipelined Floating Point Load) cycles. The 82495XP supports 3 operational modes for PFLD cycle decoding as defined by FPFLDEN and NCPFLD#:

Mode #1. PFLD cycles that are cached in the 82495XP.

Mode #2. PFLD cycles not cached in the 82495XP, without an external PFLD extension FIFO.

Mode #3. PFLD cycles not cached in the 82495XP, with an external PFLD extension FIFO.

Mode #	FPFLDEN	NCPFLD#
1	0	1
2	0	0
3	1	1
Illegal Mode	1	0

See Section 5.2.5 for details.

7.48.2 CASES IT IS ASSERTED AND DEASSERTED

NCPFLD# is sampled on the falling edge of RESET and is a don't care at any other time. NCPFLD# must be valid for at least 10 CLK's before RESET's falling edge.

7.48.3 RELATION TO OTHER SIGNALS

NCPFLD# shares a pin with FLUSH#. Both NCPFLD# and FPFLDEN describe the PFLD mode used.

7.49 NENE#

Next Near

Indicates current cycle address is near previous one.

Output from 82495XP (pin D5) Cycle Control Signal

Synchronous to CLK

7.49.1 SIGNAL DESCRIPTION

NENE# indicates to the MBC that the address of the requested memory cycle is "near" the address of the previously generated one (in the same 2K DRAM page). This information may be used by the MBC to optimize access to paged or static column DRAMs.

7.49.2 WHEN DRIVEN

NENE# is valid together with CADS# and will stay valid until CNA# or CRDY#.

7.49.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

NENE# may change state after CNA# or CRDY# are asserted to the 82495XP.

7.50 PALLC#

Potential Allocate

Indicates 82495XP intent to allocate current cycle

Output from 82495XP (pin D2) Cycle Control Signal

Synchronous to CLK

7.50.1 SIGNAL DESCRIPTION

PALLC# indicates to the MBC that the current write cycle may allocate (perform a line-fill on) a cache line. The MBC chooses to perform an allocation by asserting MKEN# during KWEND# of the write cycle. Potential allocate cycles are cycles which are 82495XP misses with PCD and PWT inactive.

The exact condition for assertion of PALLC# is:

Miss * !PCD * !PWT * LOCK# * W/R# * D/C# * M/IO#

PALLC# is inactive (HIGH) for any write-hit to a Read-Only line.

7.50.2 WHEN DRIVEN

PALLC# is valid in the same CLK as CADS# and is valid until CRDY# or CNA#.

7.50.3 RELATION TO OTHER SIGNALS

PALLC# is valid with CADS#.

7.51 PAR#

Parity Selection

Selects 82490XP as a Parity Device

Input to 82490XP (pin 32) Configuration Signal

Synchronous to CLK

7.51.1 SIGNAL DESCRIPTION

PAR# is a strapping option on the 82490XP that, when strapped low, configures that 82490XP device to be a dedicated parity device. A 82490XP parity device must be configured the same as all the other devices, however, the data lines are defined differently. CDATA[0:3] are 4 parity bit I/O lines and CDATA[4:7] are 4 bit select lines so each parity line may be written individually. Parity devices must be used as follows:

Cache Size	Memory Bus Width	Number of Parity Devices	82490XP I/O Bits (CPU:Mem)
256K	64	2	4:4
512K	128	2	4:8

7.51.2 WHEN SAMPLED

PAR# is a strapping option and must be tied either high or low.

7.51.3 RELATION TO OTHER SIGNALS

PAR# affects the definition of the CDATA and MDATA lines of the 82490XP.

7.52 RDYSRC

Ready Source

Cycle control signal to the MBC

Output from 82495XP (pin C1) Cycle Control Signal

Synchronous to CLK

7.52.1 SIGNAL DESCRIPTION

RDYSRC serves as a cycle control signal to the MBC. It indicates the source of the BRDY# generation (either 82495XP or MBC) for the CPU. When high it indicates that the MBC should generate the BRDY#s to the CPU, when low it indicates that the 82495XP will provide the BRDY#s.

RDYSRC is asserted for line-fill and not asserted for the write portion of allocation cycles. RDYSRC is also asserted (high) for all I/O cycles.

7.52.2 WHEN DRIVEN

RDYSRC is valid in the same CLK as CADS# and is valid until CRDY# or CNA#.

7.52.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, SMLN#, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.53 RESET

Reset

Forces the 82495XP to begin execution in a known state

Input to 82495XP (Q5)

Asynchronous

7.53.1 SIGNAL DESCRIPTION

The falling edge of this signal tells the 82495XP to sample all configuration inputs and initializes the 82495XP to a known state. See the specific configuration signals for setup and hold times relative to RESET's falling edge. RESET can be asserted at any time.

During initlialization, the 82495XP LRU bits are set to 1 indicating that the 82495XP LRU way is way 1. The 82490XP MRU bits are initlialized to 0 as are all tag array bits.

RESET takes about 4100 clocks in the 82495XP. RESET with self-test takes about 80,000 clocks.

7.53.2 WHEN SAMPLED

RESET is an asynchronous input. RESET must have a pulse width of at least 8 CLK's in order to guarantee 82495XP recognition.

7.53.3 RELATION TO OTHER SIGNALS

The following signals are sampled at RESET:

CNA# [CFG0]:	CFG0 line of 82495XP configuration inputs
SWEND# [CFG1]:	CFG1 line of 82495XP configuration inputs
KWEND# [CFG2]:	CFG2 line of 82495XP configuration inputs
FLUSH# [NCPFLD#]:	If low, enables decoding of i860XP non-cacheable PFLD mode.
FPFLD# [FPFLDEN]:	If high, enables the external FIFO for i860XP PFLD mode.
BGT# [C490LDRV]:	Indicates the driving strength of the 82495XP/82490XP interface.
SYNC# [MEMLDRV]:	Indicates the memory bus driving strength.
SNPCLK# [SNPMD]:	Indicates the snooping mode; synchronous or strobed.
CFG2-CFG0	Configure cache parameters such as lines/sector, line ratio, and number of tags.

7.54 SLFTST#

Self Test

Executes 82495XP self-test

Input to 82495XP (pin M2) Test Signal

Synchronous to CLK

7.54.1 SIGNAL DESCRIPTION

If SLFTST# is sampled low and HIGHZ# is sampled high, the 82495XP will perform a self-test after reset. The results of the self-tests are given by CAHOLD when FSIOUT# goes inactive.

7.54.2 WHEN SAMPLED

SLFTST# is sampled with reset like figure 7-1 with a setup time of 10 CPU clocks. SLFTST# is then a "don't care" until after the first CADS# activation when it becomes the CRDY# pin.

7.54.3 RELATION TO OTHER SIGNALS

SLFTST# shares a pin with CRDY#. The 82495XP enters self-test if both SLFTST# is sampled active and HIGHZ# is sampled inactive.

7.55 SMLN#

Same Line

Current cycle is same 82495XP line as previous one.

Output from 82495XP (pin C6) Cycle Control Signal

Synchronous to CLK

7.55.1 SIGNAL DESCRIPTION

SMLN# is used to indicate to the MBC that the current cycle is accessing the same 82495XP cache line as the previous cycle. This indication can be used by the MBC to selectively activate its SNPSTB# signal to other caches in the system. For example, back-to-back snoop hits to the same line may be snooped only once.

7.55.2 WHEN DRIVEN

SMLN# is asserted with CADS# and will stay valid until CNA# or CRDY#.

7.55.3 RELATION TO OTHER SIGNALS

Address and cycle specification signals (MSET0–MSET10, MTAG0–MTAG11, MCFA0–MCFA6, CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, NENE#, **SMLN#**, KLOCK#, and CPLOCK#) will be valid with CADS#.

7.56 SNPADS#

Cache Snoop Address Strobe

Initiates a snoop write back cycle

Output from 82495XP (pin F3) Snooping Signal

Sync to CLK

7.56.1 SIGNAL DESCRIPTION

The SNPADS# signal indicates valid cache control and attribute signals, functioning identically to CADS#, but is generated only on snoop write-backs. The separation of address status signals for normal and snoop write-back cycles eases memory bus controller implementation. When SNPADS# is activated, the memory bus controller should abort all pending cycles for which BGT# has not been issued. The 82495XP reissues these non-committed cycles after the snoop write-back has completed.

7.56.2 WHEN DRIVEN

SNPADS# is produced when a snoop hits a modified line. A modified line condition exists when a line in the cache has been updated, and copies of that memory location in other devices are no longer valid. A snoop is initiated by the master of a shared bus when accessing a memory location on the shared bus.

The response of the 82495XP to a snoop appears on the MTHIT# and MHITM# pins in the clock after SNPCYC# is active. If these pins are both driven low, the snoop resulted in a hit to a modified line, and a snoop write-back is initiated with the assertion of SNPADS#. SNPADS# is driven, at earliest, two clocks after SNPCYC#. Like CADS#, SNPADS# is active for one CLK, and is always valid.

7.56.3 RELATION TO OTHER SIGNALS

Cycles initiated by SNPADS# require only CRDY#; they do not require the other cycle progress signals (BGT#, KWEND#, SWEND#).

The SNPADS# signal is driven by the 82495XP to indicate the start of the write-back cycle; the 82495XP drives the following address and cycle specification signals valid with SNPADS#: CW/R#, CD/C#, CM/IO#, MCACHE#, RDYSRC, NENE#, SMLN#, and the address on MSET[0:10], MTAG[0:11], and MCFA[0:6]. Upon assertion of SNPADS#, the memory bus controller should cancel all pending cycles for which BGT# has not yet been asserted, because they will be reissued after the snoop write-back. The 82495XP will ignore BGT# while SNPBSY# and MHITM# are active (ie, during the write-back).

The 82495XP can accept a snoop request while performing memory bus transfers of its own. If a snoop is requested while it is performing a transfer of its own, the results of the snoop and any necessary snoop write-backs may be delayed. If SNPSTB# is sampled at a 82495XP after it has received BGT# for its own cycle, and the snoop hits a modified line, the snoop write-back will occur after CRDY# for the 82495XP's own cycle. See Sections 6.2.4 and 6.2.5 for details.

7.57 SNPBSY#

Snoop Busy

Indicates additional snoop processing in progress

Output from 82495XP (pin F1) Snooping Signal

Sync to CLK

7.57.1 SIGNAL DESCRIPTION

SNPBSY# and SNPCYC# indicate a snoop in progress. The SNPCYC# signal is asserted on the actual snoop look-up to the 82495XP tags. If the snoop look-up indicates a valid line is hit and the snoop is invalidating, the 82495XP must perform a back invalidation on the CPU. If a snoop hit occurs to a modified line, a snoop write-back must occur. SNPBSY# is asserted and remains active while either a back invalidation or a snoop write-back is in progress.

7.57.2 WHEN DRIVEN

SNPBSY# is activated for two conditions. First, SNPBSY# is activated whenever a back invalidation is necessary: the snoop returns MTHIT# active and SNPINV was asserted on the snoop initiation. Second, SNPBSY# is activated when a modified cache line is hit on a snoop, as indicated by MHITM#, until the modified line has been written back (CRDY# returned for the write-back).

SNPBSY# is valid in the CLK following SNPCYC#, and if active, remains active for a minimum of two CLKS.

7.57.3 RELATION TO OTHER SIGNALS

After SNPCYC# occurs for a snoop, a new snoop may be initiated. If SNPBSY# is asserted for the initial snoop, the SNPCYC# of the second snoop is delayed until the SNPBSY# signal is deasserted for the initial snoop, indicating that its snoop processing has completed.

7.58 SNPCLK [SNPMD]

Snoop Clock [Snooping Mode]

Selects 82495XP snooping mode

Input to 82495XP (pin S3) Snooping Signal

Synchronous to CLK

7.58.1 SIGNAL DESCRIPTION

SNPMD selects whether the 82495XP snoop initiation be in synchronous, clocked, or strobed mode. 82495XP snoop response is always synchronous to CLK.

Synchronous mode (to CLK) is selected by SNPMD sampled low during reset. Strobed mode is selcted by SNPMD sampled high during reset. Clocked mode is selected by connecting the snoop clock source to SNPMD, and thus SNPMD becomes the actual snoop clock (SNPCLK).

7.58.2 WHEN SAMPLED

SNPMD is sampled like figure 7-1 with a setup time of 4 CPU clocks. SNPMD is then not used unless clocked mode is being selected. If clocked mode is selected, SNPMD becomes SNPCLK to clock in snoop requests.

7.58.3 RELATION TO OTHER SIGNALS

SNPMD becomes SNPCLK if a clock signal is detected at reset. In this clocked mode, SNPCLK is then used to clock-in SNPSTB#, the snoop address, and all snoop attributes.

7.59 SNPCYC#

Snoop Cycle

Indicates snoop look-up occurring in 82495XP tags

Output from 82495XP (pin H3) Snooping Signal

Sync to CLK

7.59.1 SIGNAL DESCRIPTION

SNPCYC# is asserted by the 82495XP during the clock when the actual tag look-up for the snoop is performed. SNPCYC# may appear as early as the CLK following SNPSTB# assertion, or may be delayed several clocks while a snoop write-back or 82495XP memory bus cycle take place.

7.59.2 WHEN DRIVEN

SNPCYC# is always a valid 82495XP output. It is asserted once, for a single clock, for every snoop which is initiated in the 82495XP.

7.59.3 RELATION TO OTHER SIGNALS

A snoop is initiated by assertion of the SNPSTB# input if MAOE# is not asserted. The actual snoop, signalled by the assertion of SNPCYC#, can be delayed by a prior snoop's write-back in progress (SNPBSY# asserted) or by a 82495XP memory cycle in progress (SNPSTB# occurs after BGT#)—see SNPSTB# for details. If neither of these is occurring, strobed and clocked snooping modes can also delay snoop look-up for a clock while the snoop address and attributes are synchronized.

In the clock following SNPCYC#, MHITM# and MTHIT# report valid snoop results.

7.60 SNPINV

Snoop Invalidation

Forces invalidation of snoop hits

Input to 82495XP (pin P5) Snooping Signal

Sampled with SNPSTB# (see SNPSTB#)

7.60.1 SIGNAL DESCRIPTION

Assertion of the SNPINV signal during the initiation of a snoop request forces a snoop hit for that request into the Invalid state.

The SNPINV pin is sampled upon initiation of a snoop request with SNPSTB# activation, depending on snooping mode: rising edge of first CLK when SNPSTB is asserted (synchronous snooping mode), or rising edge of first SNPCLK when SNPSTB# is asserted (clocked mode), or falling edge of strobed SNPSTB# (strobed mode).

7.60.2 WHEN SAMPLED

When a bus master performs a bus access, the SNPSTB# of all other 82495XPs is asserted to initiate a snoop for that address. If the master's access is one which is modifying the data (a write to memory, etc.), the SNPINV pin of all snooping 82495XPs must be asserted during SNPSTB# so that the line is properly marked Invalid.

SNPINV is not asserted during SNPSTB# assertion if snoop hits are to remain valid: the master issuing the snoop does not require their invalidation (a read).

SNPINV assertion forces all snoop hits to be invalidated, overriding other inputs or attributes (ie SNPNCA). When SNPINV is not asserted, cache states change according to normal protocol.

SNPINV is only sampled with SNPSTB#, which may be qualified by CLK or SNPCLK depending on the snooping mode, and must meet setup and hold times for the edge of its sampling. When SNPSTB# is not being asserted, SNPINV is a don't care and need not follow setup and hold times.

7.60.3 RELATION TO OTHER SIGNALS

SNPINV is sampled according to SNPSTB#, which may be qualified by SNPCLK or CLK, depending on the snooping mode. SNPINV overrides the SNPNCA input, which may also be asserted with SNPSTB#. If MAOE# is active with SNPSTB# sampling, the snoop request is ignored.

7.61 SNPNCA

Snoop Non Caching device Access

Indicates to snooping 82495XP that the initiating master is a non- caching device

Input to 82495XP (pin Q3) Snooping Signal

Sampled with SNPSTB# (see SNPSTB#)

7.61.1 SIGNAL DESCRIPTION

SNPNCA indicates that the master which is initiating the snoop request will not cache the data. If the SNPNCA pin is not asserted and the snoop is noninvalidating (where noninvalidating = SNPINV not asserted), a snoop hit line must be placed in the Shared state, since the data will exist in another cache. If SNPNCA is asserted and the snoop is noninvalidating, a snoop hit line will not be entered into a new cache, so a hit Exclusive or Modified line will be placed in the Exclusive state by the 82495XP. A noninvalidating snoop hit to a Shared line must keep the hit line in the Shared state, regardless of SNPNCA.

SNPNCA is sampled upon initiation of a snoop request with SNPSTB# activation, depending on the snooping mode: rising edge of first CLK when SNPSTB# asserted (synchronous snooping mode), or the rising edge of SNPCLK when SNPSTB# is asserted (clocked snooping mode), or the falling edge of SNPSTB# (strobed snooping mode).

7.61.2 WHEN SAMPLED

To achieve maximum processor performance and minimum bus traffic, SNPNCA should be asserted when the noninvalidating snoop is caused by an access from a non-caching device like a DMA.

If the snoop is being caused by a device which will also be caching the data, SNPNCA must not be asserted, so that the 82495XP does not leave the hit line in an Exclusive state—subsequent writes to lines in this state do not appear on the bus, and stale data would result in the cache which incorrectly asserted SNPNCA.

If SNPNCA is asserted on a noninvalidating snoop request, the following outlines the behavior of the cache for a snoop hit in each of the MESI states:

Modified	The data is written to the bus, and the line is placed in the Exclusive state
Exclusive	The line remains in the Exclusive state
Shared	The line remains in the Shared state
Invalid	This is a cache miss. The line remains Invalid.

If SNPNCA is NOT asserted on a noninvalidating snoop request, an M, E, or S state hit line will be placed in the Shared state. Again, M state causes a write to the bus, Invalid lines remain Invalid.

SNPNCA is only sampled with SNPSTB#, which may be qualified by CLK or SNPCLK depending on the snooping mode, and must meet setup and hold times for the edge of this sampling. When SNPSTB# is not being sampled, SNPNCA is a don't care and need not follow set-up and hold times.

7.61.3 RELATION TO OTHER SIGNALS

SNPNCA is sampled with SNPSTB#, which may be qualified by SNPCLK or CLK, depending on snooping mode. The assertion of SNPINV overrides SNPNCA, and places all snoop hit lines into the Invalid state. If MAOE# is active on SNPSTB# sampling, the snoop request is ignored.

7.62 SNPSTB#

Snoop Strobe

Initiates 82495XP snoop and latches snoop address & attributes

Input to 82495XP (pin R3) Snooping Signal

Sync to CLK or SNPCLK, or strobed

7.62.1 SIGNAL DESCRIPTION

Snoop strobe initiates a 82495XP snoop request. It controls the latching of the snoop address and snoop attribute signals, in the manner specified by one of three snooping modes:

Snooping Modes

Mode	Snoop Address/ Attributes Sampled on:
Strobed	falling edge of SNPSTB#
Clocked	rising edge of SNPCLK when SNPSTB# sampled active
Synchronous	rising edge of CLK when SNPSTB# sampled

SNPSTB# must be asserted to initiate a snoop request. Snoops are initiated by a bus master for all memory accesses, to ensure that data residing in other caches is flushed if modified and invalidated if necessary.

SNPSTB# must be deasserted for at least one SNPCLK or CLK when clocked or synchronous snooping mode (respectively) is used, in order to rearm for the next snoop.

SNPSTB# can be asserted while a snoop is in progress, allowing one level of pipelining. However, the reassertion of SNPSTB# while snooping is in progress must not occur until after SNPCYC#—precisely, after the falling edge of SNPCYC# for strobed and clocked modes, or in the clock after SNPCYC# is active for synchronous mode. SNPSTB# must not be asserted between the first and last BGT# of a locked sequence. Similarly, SNPSTB# must not occur after the BGT# of the write through and before the BGT# of the allocation when a Read-for-Ownership transaction is occurring.

SNPSTB# itself does not affect the cache contents or states, but the snoop signals SNPINV and SNPNCA, latched upon SNPSTB#, force various changes in the cache on a snoop hit.

7.62.2 WHEN SAMPLED

SNPSTB# is sampled on every SNPCLK or CLK in clocked or synchronous modes, and is sampled constantly in strobed mode. While a snoop is in progress, a new SNPSTB# is recognized as a new, possibly pipelined, snoop request. After the assertion of a pipelined SNPSTB#, the SNPSTB# signal must not be reasserted until after the next SNPCYC#.

SNPSTB# should always meet proper set-up and hold times when operating in clocked or synchronous modes. When operating in strobed mode, it must meet minimum active/inactive times to be properly recognized in the next clock.

7.62.3 RELATION TO OTHER SIGNALS

SNPSTB# latches the following signals: SNPINV, SNPNCA, MBAOE#, and MAOE#, and the address on the MSET, MTAG, and MCFA pins. The address which appears on the MSET, MTAG, and MCFA address pins is to be snooped in the 82495XP. MAOE# acts as a qualifier for a snoop; if MAOE# is active when sampled on a SNPSTB# assertion, the snoop request is ignored. SNPINV and SNPNCA provide the 82495XP with snoop attributes which affect the state of a snoop hit cache entry.

If MBAOE# is active during SNPSTB# assertion, the 82495XP forces all bits in the subline address (those address bits which MBAOE# controls) to 0 on a snoop write back for that snoop.

Snoops and memory accesses are interlocked, such that after BGT# for a memory access has been issued, a SNPSTB# which is asserted will be latched, with its address and attributes, but will not cause a snoop until after SWEND# for that memory cycle. After BGT# has been issued for a cycle, snoop write-backs are delayed until after the CRDY# for that cycle. Likewise, once a snoop is underway (SNPCYC# active) BGT# is ignored until snoop completion.

SNPSTB# must not be deasserted and reasserted (specifically, cause a second falling edge) between its initial recognition and SNPCYC#—ie, SNPSTB# must not be asserted before the SNPCYC# of the previous SNPSTB#. In strobed and clocked modes, SNPSTB# can be reasserted after the falling edge of SNPCYC#; in synchronous mode, SNPSTB# can be reasserted in the CLK after SNPCYC# is active. This second assertion of SNPSTB#, after SNPCYC#, can occur while the first snoop is still progressing (SNPBSY# is active), allowing one level of snoop pipelining. In this case, a third assertion of SNPSTB# must not occur until after the SNPCYC# for the second, piped snoop request.

SNPSTB# must not be asserted while the 82495XP is executing a locked sequence (LOCK# active). Specifically, SNPSTB# must not be asserted after the BGT# for the first locked access and before the BGT# of the last locked access.

Systems which support Read-for-Ownership must not assert SNPSTB# between the BGT# of the write through and the BGT# of the allocation during a Read-for-Ownership operation.

7.63 SWEND#

Snoop Window End

Closes Snooping Window

Input to 82495XP (pin Q1) Cycle Progress Signal

Synchronous to CLK

7.63.1 SIGNAL DESCRIPTION

SWEND# is an input to the 82495XP that, when asserted, closes the snooping window and causes sampling of MWB/WT# and DRCTM#. Once snooping of all other 82495XP's is complete, DRCTM# and MWB/WT# can be determined.

Snoop response is blocked by the 82495XP between BGT# and SWEND# activation. Therefore, the faster SWEND# is closed, faster snoops can be determined.

All CPU-generated write cycles and cache read miss cycles must cause a snoop on the memory bus. SWEND# may be activated once snooping has completed for these cycles. SWEND# activation causes the 82495XP's internal tags to change state for the current cycle (if necessary). DRCTM# and MWB/WT# influence the state change decision.

SWEND# need only be activated for those cycles which require the sampling of DRCTM# and MWB/WT#.

If a cycle does not specifically require SWEND#, and SWEND# is not returned, snooping is blocked from BGT# to CRDY#. For this reason, it may be more efficient to always return SWEND#.

7.63.2 WHEN SAMPLED

SWEND# is sampled by the 82495XP on the clock or after KWEND# is sampled active for those cycles that sample KWEND#. For cycles that do not sample KWEND#, SWEND# is sampled with or after BGT#. Once SWEND# is sampled active, it is ignored until KWEND# of the next cycle. If SWEND# is not being sampled, it may violate setup and hold times.

Snoop response is blocked between BGT# and SWEND#. If a snoop is initiated between BGT# and SWEND#, the MTHIT# and MHITM# response is given after SWEND# activation. Any subsequent snoop write back would begin after CRDY#.

7.63.3 RELATION TO OTHER SIGNALS

SWEND# causes the sampling of MWB/WT# and DRCTM#. SWEND# is sampled once KWEND# is sampled active. BGT#, KWEND#, and SWEND# may be asserted in the same clock.

SWEND# shares a pin with CFG1.

7.64 SYNC#

Sync

Synchronizes 82495XP TAG array with Main Memory

Input to 82495XP (Q4) Cache Synchronization Signal

Asynchronous

7.64.1 SIGNAL DESCRIPTION

SYNC# activation will cause the synchronization of the 82495XP and i860 XP CPU tag arrays with main memory. The 82495XP will flush all modified entries to memory. All valid tag entries will be kept, with modified [M] state lines becoming non-modified [E] state lines.

7.64.2 WHEN SAMPLED

SYNC# can be asserted at any time. The 82495XP will complete all outstanding cycles on the CPU and memory bus before beginning the SYNC process. The memory bus controller does not have to prevent SYNC# during locked cycles because the 82495XP will complete its locked cycle before the SYNC process will begin.

Once a SYNC operation has begun, the SYNC# signal is ignored until the operation completes. If RESET or FLUSH# is asserted while the SYNC operation is in progress, the SYNC operation will be aborted and the RESET or FLUSH immediately executed.

SYNC# is an asynchronous input. SYNC# must have a pulse width of 2 CLK's in order to guarantee 82495XP recognition.

7.64.3 RELATION TO OTHER SIGNALS

To initiate a SYNC, the 82495XP will complete all pending cycles and prohibit further ADS#'s to occur while a SYNC is in progress. The FSIOUT# output signal is used to indicate the start and end of the SYNC operation. It will become active when the SYNC# signal is internally recognized (all outstanding cycles have completed) and will de-activate when the SYNC operation has completed.

The memory bus controller supplies BRDY# to the CPU once the SYNC has completed. Once SYNC has begun, and FSIOUT# active, all CADS#'s and CRDY#'s correspond to the write-backs caused by the SYNC operation.

The 82495XP can be snooped during SYNC cycles and the snooping protocols will be the same as that for any memory bus cycle.

7.65 TCK

Test Clock

Clock for the JTAG boundary scan tests

Input to the i860 XP CPU (pin Q1) Test Signal

Input to the 82495XP (pin P3)

Input to the 82490XP (pin 3)

Synchronous

7.65.1 SIGNAL DESCRIPTION

TCK is an input to the i860 XP CPU, 82495XP and 82490XP and provides the clocking function required by the JTAG boundary scan feature. TCK is used to clock state information and data into and out of the component. State select information and data are clocked into the component on the rising edge of TCK on TMS and TDI, respectively. Data is clocked out of the part on the falling edge of TCK on TDO.

In addition to using TCK as a free running clock, it may be stopped in a low, logic 0, state, indefinitely as described in IEEE 1149.1. While TCK is stopped in the low state, the boundary scan latches retain their state.

When boundary scan is not used, TCK should be tied low.

7.65.2 WHEN SAMPLED

TCK is a clock signal and is used as a reference for sampling other JTAG signals.

7.65.3 RELATION TO OTHER SIGNALS

On the rising edge of TCK, TMS and TDI are sampled. On the falling edge of TCK, RDO is driven.

7.66 TDI

Test Data Input

Receives serial test instructions and data

Input to the i860 XP CPU (pin S14) Test Signal

Input to the 82495XP (pin N3)

Input to the 82490XP (pin 2)

Synchronous to TCK

7.66.1 SIGNAL DESCRIPTION

TDI is the serial input used to shift JTAG instructions and data into the component. The shifting of instructions and data occurs during the SHIFT-IR and SHIFT- DR TAP controller states, respectively. These states are selected using the TMS signal as described in chapter 9.

An internal pull up resistor is provided on TDI to ensure a known logic state if an open circuit occurs on the TDI path. Note than when "1" is continuously shifted into the instruction register, the BYPASS instruction is selected.

7.66.2 WHEN SAMPLED

TDI is sampled on the rising edge of TCK, during the SHIFT-IR and the SHIFT-DR states. During all other TAP controller states, TDI is a "don't care".

7.66.3 RELATION TO OTHER SIGNALS

TDI is only sampled when TMS and TCK have been used to select the SHIFT-IR or SHIFT-DR states in the TAP controller.

For proper initialization of JTAG logic, TDI should be driven high, "1", for at least four TCK cycles following the rising edge of RESET.

7.67 TDO

Test Data Output

Outputs serial test instructions and data

Output from the i860 XP CPU (pin R10) Test Signal

Output from the 82495XP (pin C4)

Output from the 82490XP (pin 84)

Synchronous to TCK

7.67.1 SIGNAL DESCRIPTION

TDO is the serial output used to shift JTAG instructions and data out of the component. The shifting of instructions and data occurs during the SHIFT-IR and SHIFT- DR TAP controller states, respectively. These states are selected using the TMS signal as described in chapter 9.

When not in SHIFT-IR or SHIFT-DR state, TDO is driven to a high impedance state to allow connecting TDO of different devices in parallel.

7.67.2

TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT- DR TAP controller states. At all other times TDO is driven to the high impedance state.

7.67.3

TDO is only driven when TMS and TCK have been used to select the SHIFT- IR or SHIFT-DR states in the TAP controller.

7.68 TMS

Test Mode Select

Controls testing by selecting mode of operation

Input to the i860 XP CPU Test Signal

Input to the 82495XP (pin P2)

Input to the 82490XP (pin 1)

Synchronous to TCK

7.68.1 SIGNAL DESCRIPTION

TMS is decoded by the JTAG TAP (Tap Access Port) to select the operation of the test logic, as described in chapter 9.

To guarantee deterministic behavior of the TAP controller TMS is provided with an internal pull-up resistor. If boundary scan is not used, TMS may be tied high or left unconnected.

7.68.2 WHEN SAMPLED

TMS is sampled on every rising edge of TCK.

7.68.3 RELATION TO OTHER SIGNALS

TMS is used to select the internal TAP states required to load boundary scan instructions to data on TDI.

For proper initialization of the JTAG logic, TMS should be driven high, "1", for at least four TCK cycles following the rising edge of RESET.

7.69 Vcc and Vss

Power and Ground Pins

See Tables 1.1 and 1.2 for locations.

7.70 WWOR#

Weak Write Ordering Mode

Enforces strong/weak write-ordering policy

Input to 82495XP (pin Q2) Configuration Signal

Synchronous to CLK

7.70.1 SIGNAL DESCRIPTION

When asserted during reset, the 82495XP enforces a weak write ordering policy. If WWOR# is deasserted during reset, the 82495XP enforces a strong write-ordering policy.

In a strong write-ordering mode, writes to the memory bus are forced to occur in the order in which they were posted by the CPU. In a weak write-ordering mode it is possible for:

1. A CPU posted write (A) to be waiting in a 82495XP/82490XP memory buffer.
2. A subsequent CPU write (B) to complete in the 82495XP/82490XP because it was a hit to M or E state.
3. A snoop hit to B to cause a write back of B before A is written.

In this scenario, B is written to memory before A is, and thus CPU writes have been reordered.

7.70.2 WHEN SAMPLED

WWOR# is sampled during reset like figure 7-1 with a setup time of 4 CPU clocks. WWOR# becomes MALE once FSIOUT# indicates that the 82495XP reset sequence has completed.

7.70.3 RELATION TO OTHER SIGNALS

WWOR# shares a pin with MALE.

8.0 BUS FUNCTIONAL DESCRIPTION AND TIMING

The 82495XP/82490XP cache core supports a wide variety of bus transfers to meet the needs of high performance systems. Bus transfers can be single cycle or multiple cycle, cacheable or non-cacheable, 64- or 128-bit (memory bus), and locked. To support multiprocessing systems there are cache back-invalidation, inquire, snooping, read for ownership, cache to cache transfers, and locked cycles.

This section begins with read cycles, both cacheable and non-cacheable. It moves on to write cycles, cacheable and non-cacheable. Snooping cycles are discussed next with an example of each snooping mode. The remaining sections describe special cycles: read for ownership, I/O, and locked cycles.

The cycles shown in this chapter are examples of various types of 82495XP/82490XP cycles. The purpose of these examples is to show signal relationships, and are not necessarily best case scenarios.

8.1 Read Cycles

8.1.1 READ HITS

Read Hit cycles are executed completely within the CPU/Cache core, and will not be seen by the MBC.

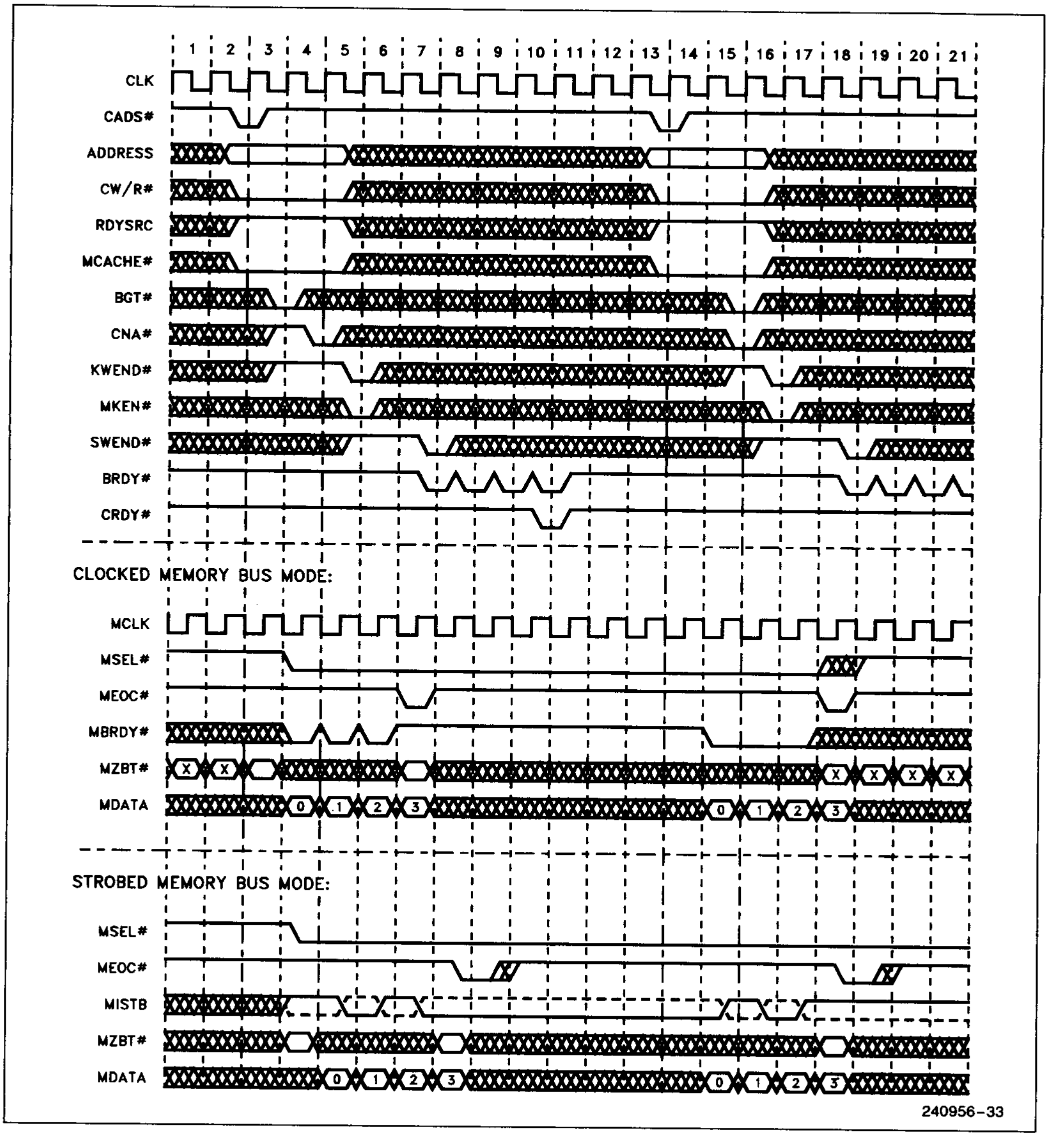


Figure 8-1. Cacheable Read Miss with Clean Replacement

8.1.2 CACHEABLE READ MISSES

8.1.2.1 Read Miss with Clean Replacement

Figure 8.1 illustrates CPU initiated Read cycles that miss the 82495XP/82490XP cache and replace a non-dirty (eg. clean or empty) line in the cache. In such cycles, the 82495XP will instruct the MBC to perform a cache line-fill cycle on the memory bus. A cache line-fill is a read of a complete 82495XP/82490XP line from main memory. The line is then written into the 82490XP's array, and data transferred to the CPU as requested. If the line fetched from main memory replaces a 82495XP/82490XP cache line which is in a valid unmodified state ([E] or [S]), then a back-invalidation cycle is performed on the CPU bus to guarantee that the replaced data is also removed from the CPU's first level cache, thus maintaining the inclusion property.

CACHE CONTROL SIGNALS:

The CPU initiates the read cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 2) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#) in order to schedule the cache line-fill operation. MCACHE# is active, indicating that the read miss is potentially cacheable by the 82495XP; RDYSRC is active, indicating that the MBC must supply BRDY#s to the CPU cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 2 and 13 for the two cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 5 and 16). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 3), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit in the cache.

CNA# is asserted by the MBC (clock 4) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# during KWEND# (clock 5) to determine that the cycle is indeed cacheable.

The MBC asserts SWEND# when the snoop window ends on the memory bus. The 82495XP samples MWB/WT# and DRCTM# during SWEND# (clock 7) and updates the cache tag state according to the consistency protocol. The closure of the snoop window also enables the MBC to start providing the CPU with data that has been stored in the 82490XP's memory cycle buffer. The MBC supplies BRDY#s to the CPU (clocks 7-10).

The first cycle ends when CRDY# is driven active by the MBC (clock 10). It is at this time that the data in the 82490XP's memory cycle buffers is loaded into the cache SRAM.

The 82495XP issues a new CADS# in clock 13, which also misses the 82495XP/82490XP cache. Note that once the cycle progress signals (BGT#, CNA#, KWEND#, SWEND#) of a cycle are sampled asserted, the 82495XP ignores them until the CRDY# of that cycle. The 82495XP does not pipeline the cycle progress signals (ie. it will not sample them again until after CRDY# of the current memory bus cycle).

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC. MDOE# must be inactive to allow the data pins to be used as inputs.

Some time after the address has been driven onto the memory bus, data will be supplied from the DRAM (main memory) to the 82490XP cache SRAM.

For Clocked Memory Bus Mode, MSEL# is driven active by the MBC (clock 4) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. MBRDY# is driven active by the MBC in clocks 4 to 6 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 7) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is latched at this time (when MEOC# is sampled asserted and MSEL# remains low) for the next transfer.

MBRDY# is driven active by the MBC in clocks 15 to 17 to read data into the 82490XP cache memory cycle buffers. The MBC asserts MEOC# (clock 18) to end the second read miss cycle on the memory bus and switch the memory cycle buffers for a new cycle.

For Strobed Memory Bus Mode, MSEL# is driven active by the MBC (clock 4) to allow MISTB operation and to latch MZBT# (on the falling edge of MSEL#) for the transfer. MISTB is toggled in clocks 5 to 7 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 8) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle, is sampled at this time on the falling edge of MEOC#.

MISTB is toggled by the MBC (clocks 15 to 17) to read data into the 82490XP memory cycle buffers. The MBC asserts MEOC# (clock 18) to end the second read miss cycle on the memory bus and switch the memory cycle buffers for a new cycle.

8.1.2.2 Read Miss with Replacement of Dirty Line

Figure 8.2 illustrates a CPU read cycle which misses the 82495XP cache, and requires the replacement of a modified line (eg. tag replacement, lines/sector=1 line ratio=1). In such cycles, the 82495XP will instruct the MBC to perform a cache line-fill on the memory bus, instruct the 82490XP to fill its write-back buffer with the contents of the array location corresponding to the line which must be replaced, and perform a back invalidation to the CPU to maintain the first and second level cache consistency. Once the cache line-fill has completed, the 82495XP/82490XP will write back the contents of the replaced line to main memory from the 82490XP write-back buffer.

CACHE CONTROL SIGNALS:

The CPU initiates the read cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#) in order to schedule the cache line-fill operation. MCACHE# is active, indicating that the read miss is potentially cacheable by the 82495XP; RDYSRC is active, indicating that the MBC must supply BRDY#s to the CPU cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 5 for the two cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 4 and 10). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the cycle is guaranteed to complete on the memory bus. At this point, the 82490XP's write-back buffer is prefilled with the line to be replaced. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

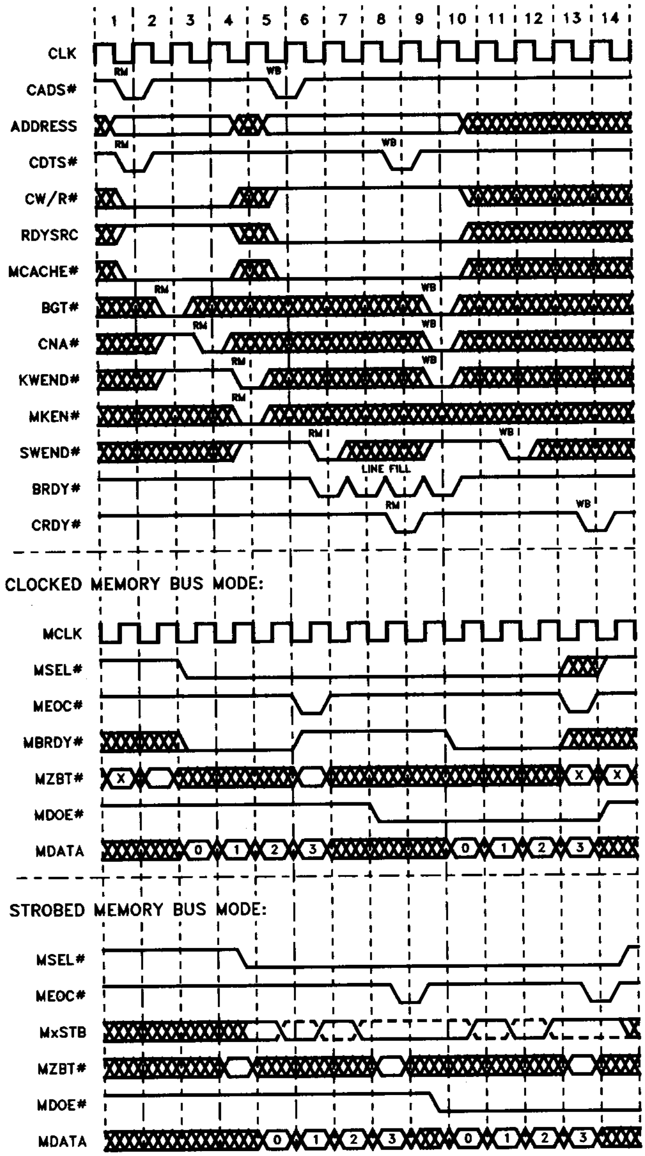
CNA# is asserted by the MBC (clock 3) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# during KWEND# (clock 4) to determine that the cycle is indeed cacheable.

The MBC asserts SWEND# (clock 6) when the snoop window ends on the memory bus. The closure of the snoop window enables the MBC to start providing the CPU with data that has been stored in the 82490XP's memory cycle buffer. The MBC supplies BRDY#s to the CPU (clocks 6-9) to serve the read cycle. Note that data may be supplied to the 82490XP's immediately after MSEL# activation, and need not wait for SWEND#.

On the memory bus, the 82495XP issues a write-back (WB) cycle. CNA# is sampled active in clock 3 causing the 82495XP to issue the CADS# (also CDTS#) of the write-back (clock 5). The MBC knows this is a write back cycle and not a CPU initiated write cycle by sampling MCACHE# asserted. This tells the MBC how many data transfers are necessary.

BGT#, CNA#, and KWEND# of the write-back are sampled asserted by the MBC (clock 9) after the CRDY# of the read miss cycle (clock 8). At this



240956-34

Figure 8-2. Cacheable Read Miss with Replacement of Dirty Line

point, the 82495XP may issue another CADS# for a new (unrelated) memory bus cycle. It is at this time that the data in the 82490XP's memory cycle buffers is loaded into the cache SRAM. The data to be written back to main memory is in the 82490XP's write back buffers.

The snoop window for the write back cycle is closed by the MBC in clock 11, and the cycle is ended by CRDY# sampled asserted in clock 13.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

Some time after the address has been driven onto the memory bus, data will be supplied from the DRAM (main memory) to the 82490XP cache SRAM.

For Clocked Memory Bus Mode, MSEL# is driven active by the MBC (clock 3) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. MBRDY# is driven active by the MBC in clocks 3 to 5 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 6) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is latched at this time (when MEOC# is sampled asserted) for the next transfer.

The MBC asserts the memory data output enable signal (MDOE#, clock 8) to drive the memory data outputs.

MBRDY# is driven active by the MBC in clocks 10 to 12 to write data from the 82490XP cache memory cycle buffers onto the memory bus. The MBC asserts MEOC# (clock13) to end the write back cycle on the memory bus and switch the memory cycle buffers for a new cycle.

For Strobed Memory Bus Mode, MSEL# is driven active by the MBC (clock 4) to allow MISTB operation and to latch MZBT# for the transfer (on MSEL# falling edge). MISTB is toggled in clocks 5 to 7 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cachememory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 8) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle, is latched at this time on the falling edge of MEOC#.

The MBC asserts MDOE# (clock 9) to drive the memory data outputs.

MOSTB is toggled by the MBC (clocks 10 to 12) to write data from the 82490XP memory cycle buffers onto the memory bus. The MBC asserts MEOC# (clock 13) to end the write back cycle on the memory bus and switch the memory cycle buffers for a new cycle.

8.1.3 NON-CACHEABLE READ MISSES

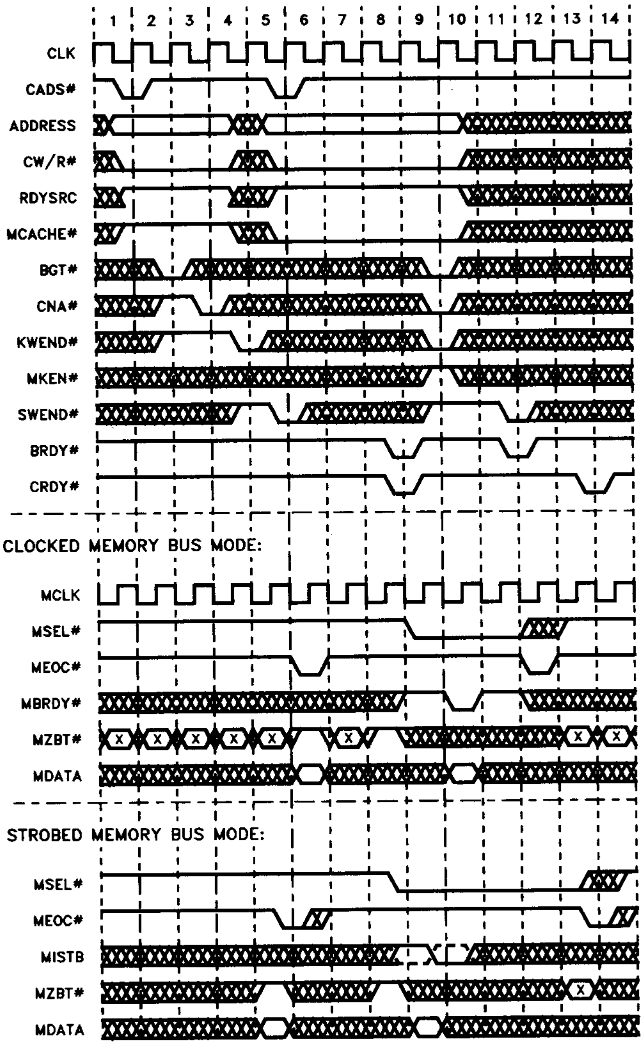
8.1.3.1 Read Misses not Cacheable by CPU/Cache Core and Cacheable by Core, but not by Memory Bus

Figure 8.3 illustrates two CPU read cycles which miss the 82495XP cache, and are non-cacheable. In the first cycle, the CPU/Cache core forces the read to be non-cacheable (as indicated by the MCACHE# output from the 82495XP). In the second cycle, non-cacheability of the data is forced by the memory bus (as indicated by the MKEN# input from the MBC). Since both cycles are not cacheable, there is no line-fill operation performed, the cycles are merely echoed to the memory bus.

CACHE CONTROL SIGNALS:

The CPU initiates the first read cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues a cycle request (CADS# in clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#) in order to schedule the read operation. RDYSRC is active, indicating that the MBC must provide BRDY# to the CPU; MCACHE# is not active, indicating that the read miss in not cacheable by the CPU/Cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 5 for the two cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 4 and 10). MALE and MBALE may be used to hold the address as necessary.



240956-35

Figure 8-3. Non-Cacheable Read Misses

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock3) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

This cycle has already been determined to be non-cacheable; therefore, The MBC does not need to assert SWEND#, KWEND#, or MKEN# to the 82495XP/82490XP cache. The MBC supplies BRDY# to the CPU to complete the cycle to the CPU. The MBC asserts CRDY (clock 8) to the 82495XP/82490XP to complete the read miss cycle on the memory bus.

The 82495XP issues a new (unrelated) cycle request (CADS# in clock 5) which also misses the 82495XP/82490XP cache. Since the 82495XP has already sampled CNA# asserted, it issues a new CADS# prior to receiving CRDY# of the current cycle (ie. this cycle is pipelined within the MBC). Note that once the cycle progress signals of a cycle are sampled asserted, the 82495XP ignores them until the CRDY# of that cycle. The 82495XP will not sample the cycle progress signals again until after the CRDY# of the current memory bus cycle. The current read cycle is completed on the bus in clock 8 with CRDY# assertion.

The cycle progress signals for the second read miss are also valid at this time (clock 5). RDYSRC is active, indicating that the MBC must provide BRDY#s to the CPU/Cache core; and MCACHE# is active, indicating that the read miss is potentially cacheable by the 82495XP/82490XP.

The MBC issues BGT# and CNA# to the 82495XP in clock 9 to indicate that the cycle is guaranteed to complete on the memory bus, and that it is ready to schedule a new memory bus cycle. KWEND# is asserted at this time to close the cacheability window. MKEN# is not active, indicating to the 82495XP that the read miss cycle is not cacheable by the memory bus. KWEND# and MKEN# must be returned to the 82495XP at least two clocks prior to BRDY# to inform the CPU that a line fill will not follow.

The MBC asserts SWEND# (clock 11) to close the snoop window, and CRDY# (clock 13) to complete the cycle to the 82495XP/82490XP. Note: SWEND# is not needed since the cycle was not cacheable.

NOTE:
Both examples show single cycle read requests.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC. The memory data output enable (MDOE#) must be inactive to allow the data pins to be used as inputs.

Some time after the address has been driven onto the memory bus, data will be supplied from the DRAM (main memory) to the 82490XP memory cycle buffers.

For Clocked Memory Bus Mode, MEOC# is asserted by the MBC (clock 6) to latch MZBT# for the next transfer, and end the current cycle on the memory bus (MBRDY# and MSEL# are not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with a non-zero burst address.

For the second non-cacheable read cycle, MSEL# is driven active by the MBC (clock 8) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MBRDY# is driven active by the MBC in clock 10 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 12) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle.

For Strobed Memory Bus Mode, MEOC# is driven active by the MBC (clock 5) to latch MZBT# for the transfer (on MEOC# falling edge), and end the current cycle on the memory bus (MISTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address.

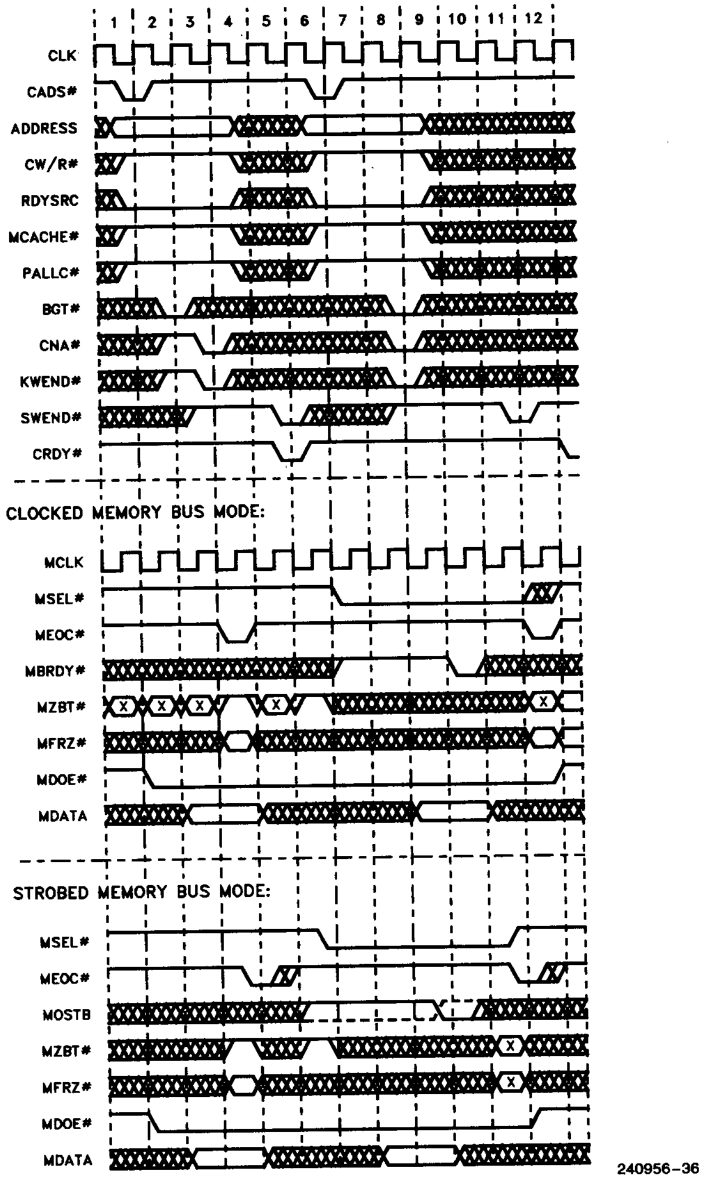


Figure 8-4. Write Hit to [S] State Line (Write-Through)

For the second non-cacheable read cycle, MSEL# is driven active by the MBC (clock 8) to allow MISTB operation and to latch MZBT# for the transfer (on MSEL# falling edge). Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MISTB is toggled in clock 9 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 13) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle (not shown), is sampled at this time on the falling edge of MEOC#.

8.2 Write Cycles

8.2.1 WRITE HITS

8.2.1.1 Write Hit to [E] or [M] States

CPU initiated write cycles which hit 82495XP entries tagged in the [E] or [M] states are executed completely within the CPU/Cache core, and will not be seen by the MBC.

8.2.1.2 Write Hit to [S] State

Figure 8.4 illustrates CPU initiated write cycles which hit lines in the 82495XP/82490XP cache array that are in the shared state. If the 82495XP/82490XP is used as a write through cache (not write back), the [S] state is the only state a cached line could be in. These cycles are posted as are all normal write cycles (as long as no other write miss is pending).

CACHE CONTROL SIGNALS:

The CPU initiates the write cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a hit to shared state, it posts the write and returns BRDY# to the CPU.

The 82495XP next issues a cycle request (CADS# in clock 1), and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, PALLC#) in order to schedule the write through operation. MCACHE# is not active since the write will be posted; RDYSRC is not active, indicating that the 82495XP will supply BRDY# to the CPU; PALLC# is not active, indicating that an allocation cycle will not be performed (regardless of MKEN# state) since the line is already available in the cache. The MBC must also latch PWT and PCD on BLE# falling edge in order to track hits and misses to the [S] state. This is how an external state tracker can track the [S] state.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 6 for the two cycles in this example) and remains valid until after CNA# is sampled active by the 82495XP (clocks 4 and 9). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 3) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid. KWEND# is also driven at this time since the cacheability of this cycle is already known and MKEN# is a don't care. It is not necessary that KWEND# be asserted at this time.

The 82495XP provides BRDY# to the CPU since the cycles are posted writes. The MBC completes the first write hit to [S] state in clock 5 when it asserts CRDY# to the 82495XP/82490XP cache. The data is latched in to the 82490XP array from the memory cycle buffer at this time.

In this example, the 82495XP issues a second write to [S] state in clock 6. For this cycle, the 82495XP issues the memory bus request (CADS#) as soon as it can after sampling CNA# asserted. The 82495XP will not wait for KWEND# (if it does not get asserted immediately as in this example) to issue CADS# since this is not a potential allocate cycle (ie. PALLC# active).

The MBC asserts BGT#, CNA#, and KWEND# together in clock 8 to indicate that the current cycle is guaranteed to complete and the 82495XP is free to schedule a new memory bus cycle.

Again, the 82495XP provides BRDY# to the CPU since the cycles are posted writes. The MBC completes the second write hit to [S] state in clock 12 when it asserts CRDY# to the 82495XP/82490XP cache. The data is latched in to the 82490XP array from the memory cycle buffer at this time.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, the memory data output enable signal (MDOE#) is asserted by the MBC in clock 2 to drive the memory data outputs.

MEOC# is asserted by the MBC (clock 4) to latch MZBT# for the transfer, and end the current cycle on the memory bus (MBRDY# is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the write cycle to begin with the correct burst address . MFRZ# is sampled here (it need not be active since the cycle is not potentially allocatable).

For the second write through cycle, MSEL# is driven active by the MBC (clock 7) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MBRDY# is driven active by the MBC in clock 10 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 12) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) is asserted by the MBC in clock 2 to drive the memory data outputs.

MEOC# is driven active by the MBC (clock 4) to latch MZBT# for the transfer (on MEOC# falling edge), and end the current cycle on the memory bus (MOSTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address.

For the second write through cycle, MSEL# is driven active by the MBC (clock 6) to allow MOSTB operation and to latch MZBT# for the transfer (on MSEL# falling edge). Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MOSTB is toggled in clock 9 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MOSTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 11) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle (not shown), is sampled at this time on the falling edge of MEOC#.

8.2.2 WRITE MISSES

8.2.2.1 Write Miss with no Allocation

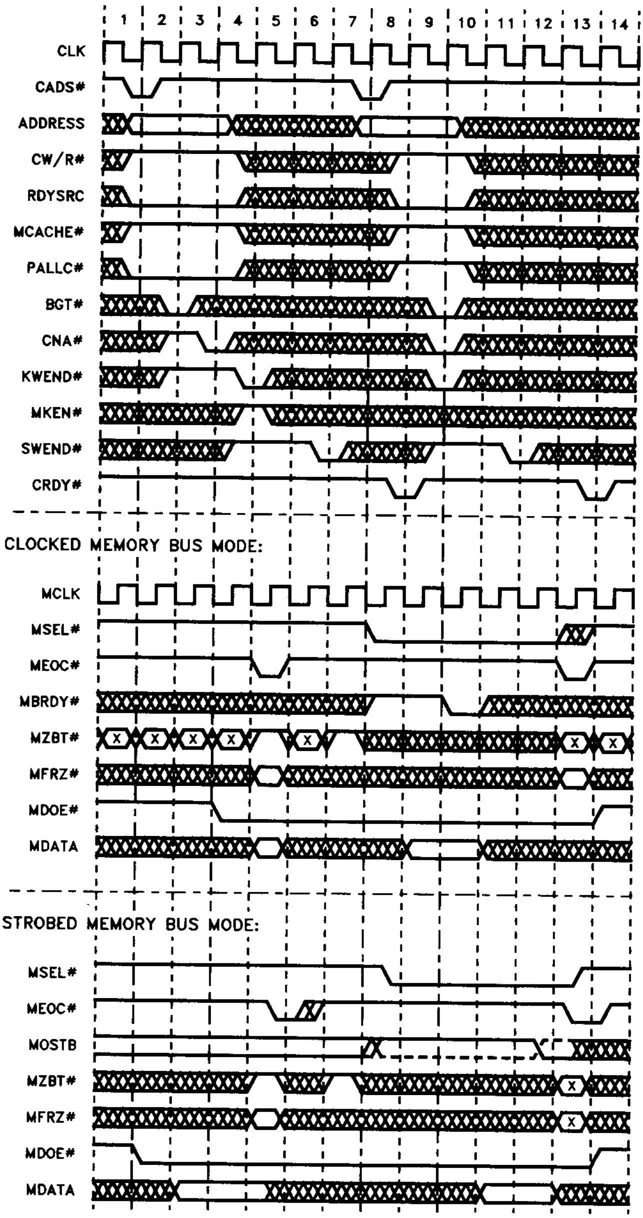
Figure 8.5 illustrates two CPU initiated write cycles which miss the 82495XP/82490XP cache and are not allocatable. The first write cycle begins as a potentially allocatable cycle, but MKEN# sampled inactive indicates that the cycle is not cacheable by the memory bus. The second write miss cycle is not cacheable by the CPU/82495XP/82490XP as indicated by the PALLC# output from the 82495XP.

CACHE CONTROL SIGNALS:

The CPU initiates the first write cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss. It issues a cycle request (CADS# in clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, PALLC#) in order to schedule the write miss operation. RDYSRC is not active, indicating that the 82495XP will supply BRDY# to the CPU; MCACHE# is not active; PALLC# is active, indicating that the cycle is potentially allocatable.

The write miss data is posted in the 82490XP's memory cycle buffer, and the cycle completes with no wait states to the CPU. The CPU is then free to issue another (non-related) cycle while the 82495XP completes the current write miss cycle and possible allocation. If this new cycle is a cache hit, it will be serviced by the 82495XP immediately; but if it is a cache miss, its service will wait until the CRDY# of the write cycle (and allocation cycle, if executed).

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 7 for the two cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 4 and 10). MALE and MBALE may be used to hold the address as necessary.



240956-37

Figure 8-5. Write Miss with No Allocation

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the write through cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 3) to indicate that it is ready to schedule a new memory bus cycle. Notice that the cycle control signals are not guaranteed to be valid after CNA# activation. NOTE that CNA# has no effect before KWEND#.

When the MBC has determined the cacheability attribute of the write through cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time (clock 4), indicating the end of the cacheability window. The 82495XP samples MKEN# inactive during KWEND#, indicating that the missed cycle is not cacheable and should not be allocated.

The MBC asserts SWEND# (clock 6) when the snoop window of the write through cycle ends on the memory bus. The MBC may return CRDY# to the 82495XP/82490XP cache any time after the closure of the snoop window. In this example, CRDY# is issued by the MBC in clock 8.

The 82495XP issues a cycle request for the second write miss cycle in clock 7. The cycle control signals are valid at this time. Note that PALLC# is inactive, indicating that the 82495XP/82490XP has determined the cycle to not be allocatable.

The MBC# asserts BGT#, CNA#, and KWEND# in clock 9. MKEN# is a don't care during the cacheability window since the cycle is not allocatable. The snoop window is closed in clock 11, and the cycle is completed on the memory bus in clock 13 with the assertion of CRDY# by the MBC.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, the memory data output enable (MDOE#) is asserted by the MBC in clock 4 to drive the memory data outputs.

MEOC# is asserted by the MBC (clock 5) to latch MZBT# for the transfer, and end the current cycle on the memory bus (MBRDY# is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address. MFRZ# is sampled here (it need not be active since the cycle is not potentially allocatable).

For the second non allocatable write cycle, MSEL# is driven active by the MBC (clock 8) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MBRDY# is driven active by the MBC in clock 10 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers.

The MBC drives MEOC# asserted (clock 13) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MFRZ# is sampled here (it need not be active since the cycle is not potentially allocatable). MZBT# is also sampled at this time.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) is asserted by the MBC in clock 2 to drive the memory data outputs.

MEOC# is driven active by the MBC (clock 5) to latch MZBT# for the transfer, and end the current cycle on the memory bus (MOSTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address.

For the second write through cycle, MSEL# is driven active by the MBC (clock 8) to allow MOSTB operation and to latch MZBT# for the transfer. Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address. MOSTB is toggled in clock 12 to cause the memory burst counter to be incremented, and data to be read from the 82490XP cache memory cycle buffers. Note: MOSTB latches the memory bus data on both the rising and falling edges.

The MBC drives MEOC# asserted (clock 13) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# and MFRZ# for the next cycle (not shown), is sampled at this time on the falling edge of MEOC#.

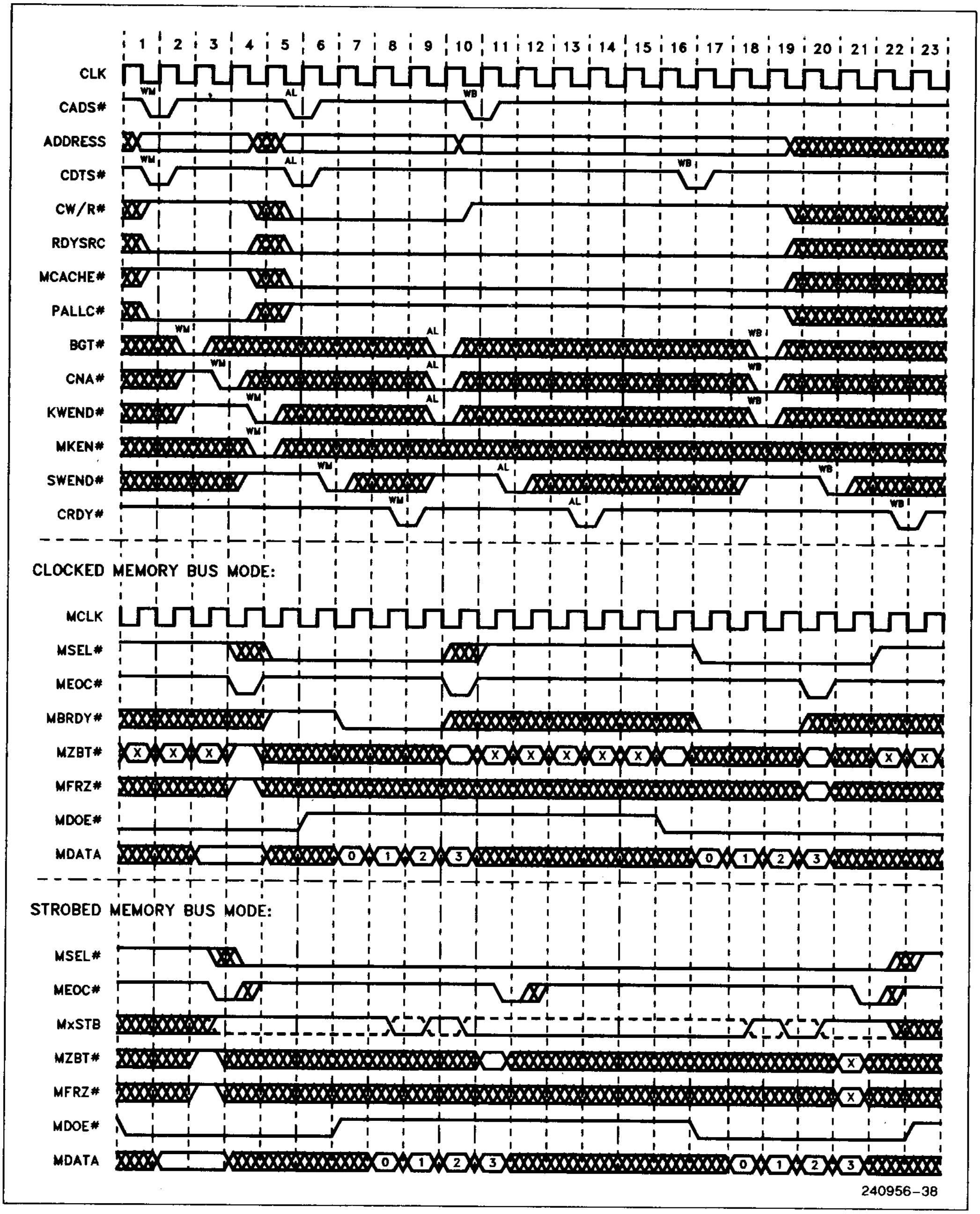


Figure 8-6. Write Miss with Allocation to [M] Line

8.2.2.2 Write Miss with Allocation

Figure 8.6 illustrates a CPU initiated write cycle which misses the 82495XP/82490XP cache and follows the write to main memory with an allocation cycle. An allocation is when the cache follows a write miss cycle with a line fill. This example assumes that allocating the new line requires the replacement of a modified line (ie. a write-back to main memory).

CACHE CONTROL SIGNALS:

The CPU initiates the write cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, PALLC#) in order to schedule the write operation. MCACHE# is not active; RDYSRC is not active, indicating that the 82495XP will supply BRDY#s to the CPU; PALLC# is asserted, indicating a potential allocate cycle after the write-through cycle.

The write miss data is posted in the 82490XP's memory cycle buffer, and the cycle completes with no wait states to the CPU. The CPU is free to issue another (non-related) cycle while waiting for the 82495XP to complete the allocation. If this new cycle is a cache hit, it will be serviced by the 82495XP immediately; but if it is a cache miss, its service will wait until the CRDY# of the allocation.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1, 5 and 10 for the three cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 4, 10 and 19). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the write through cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 3) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the write through cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# active during KWEND# (clock 4), indicating that the missed line should be allocated in the cache.

At the first available time (clock 5), the 82495XP asserts CADS# to request an allocation cycle. The cycle control signals are valid at this point: MCACHE# is active, indicating the cacheability of the line-fill cycle; RDYSRC is not active, indicating that the MBC need not supply BRDY#s to the CPU (no BRDY#s are necessary for an allocation cycle).

The MBC asserts SWEND# (clock 6) when the snoop window of the write through cycle ends on the memory bus.

The MBC may return CRDY# to the 82495XP/82490XP cache any time after the closure of the snoop window. In this example, CRDY# is issued by the MBC in clock 8. At this time, the cycle progress signals for the allocation cycle may be issued by the MBC to complete the line fill.

Once again, the MBC arbitrates for the memory bus and returns BGT# asserted (clock 9) for the allocation cycle. The MBC also asserts CNA# and KWEND# at this time. The 82495XP back-invalidates the CPU to maintain first and second level cache consistency.

In clock 10, the 82495XP asserts CADS# for the write back cycle (since the miss was to a dirty line). CDTS# is asserted by the 82495XP six clocks later (clock 16). Note that CDTS# of the write back cycle is not asserted with CADS# since the data is not yet available in the 82490XP's write-back buffer.

The MBC asserts SWEND# (clock 11) when the snoop window of the allocation cycle ends on the memory bus.

At this time, the MBC may assert CRDY# to the 82495XP/82490XP cache for the allocation cycle. CRDY# assertion will cause the data stored in the 82490XP's memory cycle buffers to be latched into the cache array.

On the memory bus, BGT#, CNA#, and KWEND# are sampled active in clock 18 for the write back cycle. The snoop window is closed two clocks later (clock 20) by the MBC with SWEND#, and the write back cycle is completed with CRDY# asserted in clock 22.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, the memory data output enable (MDOE#) has been asserted by the MBC to drive the memory data outputs.

MEOC# is asserted by the MBC (clock 4) to latch MZBT# for the transfer, and end the current cycle on the memory bus (MBRDY# is not necessary since this example shows a single transfer write miss cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address. MFRZ# is driven inactive by the MBC here, allowing the line to be placed into the exclusive ([E]) state and requiring the data to be written to main memory.

For the allocation (line fill) cycle, MSEL# is driven active by the MBC (clock 6) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. MDOE# is also deasserted in clock 6 to allow the data pins to be used as inputs for the allocation cycle.

MBRDY# is driven active by the MBC in clocks 7 to 9 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 10) to end the allocation cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is sampled and latched at this time for the next data transfer.

MDOE# is asserted by the MBC (clock 16) to drive the memory data outputs for the write back cycle.

The MBC again asserts MBRDY# (clocks 17 to 19) for the write back cycle to increment the memory burst counter and cause data to be read from the 82490XP memory cycle buffers. The write back cycle ends on the memory bus and switches memory cycle buffers with MEOC# assertion (clock 20). MZBT# and MFRZ# for the next transfer are sampled at this time. MFRZ# need not be active since the cycle is not potentially allocatable.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) has been asserted by the MBC to drive the memory data outputs for the write miss cycle.

MEOC# is driven active by the MBC (clock 4) to latch MZBT# for the transfer, and end the current cycle on the memory bus (MOSTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address. MFRZ# is driven deasserted by the MBC here, allowing the line to be placed into the exclusive ([E]) state.

For the allocation (line fill) cycle, MSEL# is driven active by the MBC (clock 6) to allow MISTB operation and to latch MZBT# for the transfer. MISTB is toggled in clocks 8 to 10 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. MDOE# is also deasserted in clock 6 to allow the data pins to be used as inputs for the allocation cycle.

The MBC drives MEOC# asserted (clock 11) to end the allocation cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle, is latched at this time on the falling edge of MEOC#.

MDOE# is asserted by the MBC (clock 18) to drive the memory data outputs for the write back cycle.

The MBC toggles MOSTB (clocks 19 to 21) for the write back cycle to increment the memory burst counter and cause data to be read from the 82490XP memory cycle buffers.

The write back cycle ends on the memory bus and switches memory cycle buffers with MEOC# assertion (clock 22). MZBT# and MFRZ# for the next transfer are sampled at this time. MFRZ# need not be active since the cycle is not potentially allocatable.

8.3 Snooping Cycles

8.3.1 SYNCHRONOUS SNOOPING MODE (HIT TO [M] LINE)

Figure 8.7 illustrates a snoop hit to a dirty line sequence occurring simultaneously with a CPU initiated read miss cycle. This example assumes synchronous snooping mode (ie. requests for snoops are done via SNPSTB# from the MBC, sampled on the 82495XP's CLK).

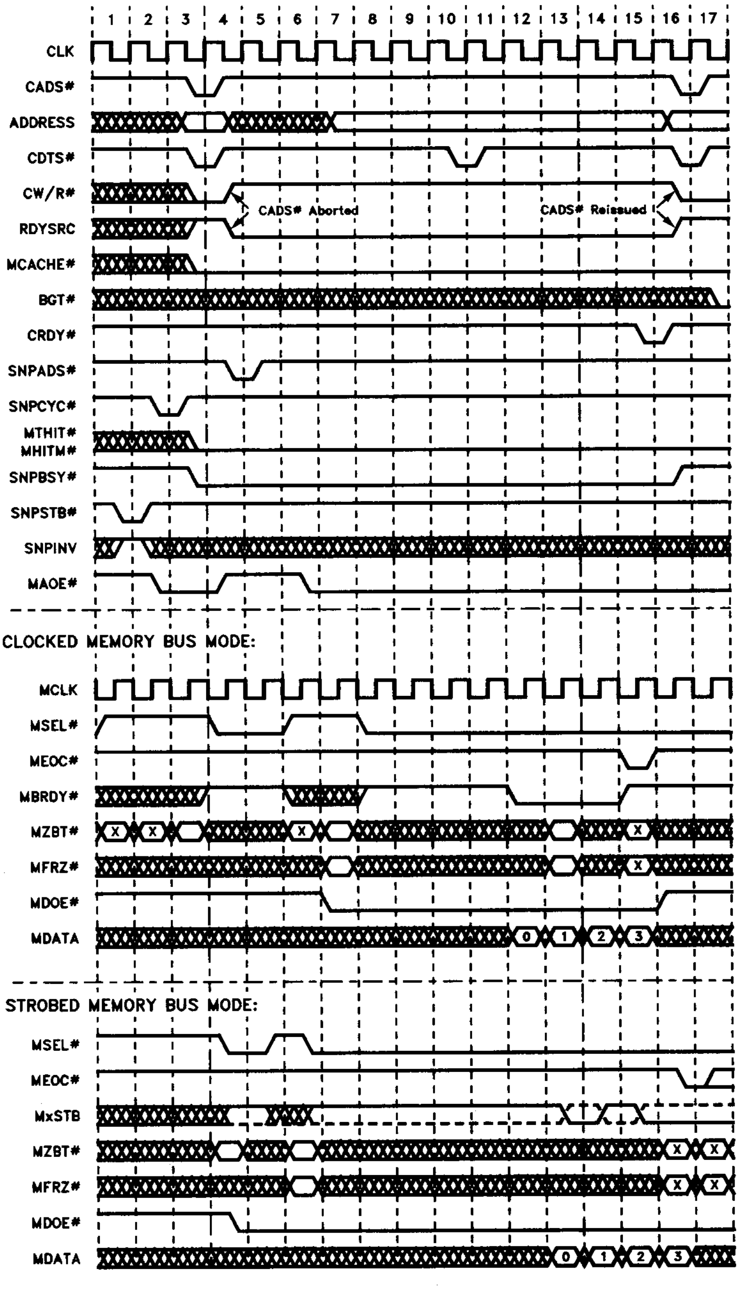


Figure 8-7. Synchronous Snooping Mode

CACHE CONTROL SIGNALS:

In clock 1 SNPSTB# is asserted by the MBC, indicating to the 82495XP a request for snooping. The 82495XP samples MAOE# (it must be inactive) in order to recognize the snoop request. It is latched together with the snoop address (MSET[0:10], MTAG[0:11], MCFA[0:6]), SNPINV, MBAOE#, and SNPNCA on the 82495XP's CLK during SNPSTB# assertion. The tag look-up is done immediately after SNPSTB# is sampled active since snoop operations have the highest priority in the cache tag state arbiter. The 82495XP issues SNPCYC# (clock 2), indicating that the snoop look-up is in progress. The results of the look-up are driven to the memory bus via MTHIT# and MHITM# in the next clock after SNPCYC#. Since the snoop hit a modified line, both signals are asserted (clock 3). SNPBSY# is also issued to indicate that the 82495XP is busy with CPU back-invalidations, the 82490XP's snoop buffer is full, or a write back is to follow. The 82495XP will accept snoops only when SNPBSY# is inactive.

Simultaneously with the memory bus activity due to the snoop request, the CPU initiates a read miss cycle. The 82495XP issues a memory bus request (CADS#), CDTS#, and cycle control signals to the MBC in clock 3. The MBC must wait for the pending snoop cycle to complete on the memory bus prior to servicing this read miss cycle.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is not valid until MAOE# goes active after CRDY# of the snoop write back cycle is sampled active by the 82495XP and the CADS# is reissued (clock 16).

In clock 4 the 82495XP issues SNPADS# and cycle control signals to the MBC, indicating a request to flush a modified line out of the cache. SNPADS# activation causes the MBC to abort the pending read miss cycle. It is the 82495XP responsibility to re-issue the aborted cycle after the completion of the write back, since BGT# was not asserted by the MBC.

Data is loaded into the 82490XP's snoop buffer. Since SNPINV was sampled asserted by the 82495XP (clock 1) during SNPSTB# assertion, it back-invalidated the CPUs first level cache.

The 82495XP asserts CDTS# (clock 10) indicating to the MBC that data is available in the snoop buffer. When the MBC complete the write back cycle on the memory bus, it activates CRDY# to the 82495XP/82490XP cache. At this time, the 82495XP deasserts SNPBSY# (clock 16) and re-issues the aborted read miss cycle (clock 16) by asserting CADS# and CDTS#.

MEMORY BUS SIGNALS:

For Clocked Memory Bus Mode, the memory data output enable (MDOE#) is not activated by the MBC to allow the memory data pins to be used as inputs.

MSEL# is driven active by the MBC (clock 4) to allow sampling of MBRDY# and to latch MZBT# for the read miss transfer. MZBT# is sampled on all MCLK rising edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer.

Since the read miss cycle is aborted due to the snoop hit to a modified line (requires a write back cycle), no MEOC# is given. MSEL# is deasserted by the MBC (clock 6) and reasserted (clock 8) to allow latching of MZBT# for the snoop write back cycle and sampling of MBRDY# for that cycle. MFRZ# is also sampled at this time.

The memory data output enable (MDOE#) signal is driven active by the MBC (clock 7) to drive the memory data outputs.

MBRDY# is driven active by the MBC in clocks 12 to 14 to cause the memory burst counter to be incremented and data to be written from the 82490XP cache snoop buffers. The MBC drives MEOC# asserted (clock 15) to end the write back cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# and MFRZ# are sampled and latched at this time for the next data transfer.

MDOE# is deasserted by the MBC (clock 16) to allow the memory data pins to be used as inputs for the reissued read cycle.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) has not been asserted by the MBC to allow the memory data pins to be used as inputs for the read miss cycle.

MSEL# is asserted by the MBC (clock 16) to allow sampling of MISTB and latch MZBT# (on the falling edge of MSEL#) for the read miss transfer.

Since the read miss cycle is aborted due to the snoop hit to a modified line (requires a write back cycle), no MEOC# is given. MSEL# is deasserted by the MBC (clock 5) and reasserted (clock 6) to allow latching of MZBT# for the snoop write back cycle and sampling of MOSTB for that cycle. MFRZ# is also sampled at this time.

MOSTB is toggled in clocks 13 to 15 to cause the memory burst counter to be incremented, and data to be read from the 82490XP cache memory cycle buffers. Note: MOSTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 16) to end the snoop write back cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# and MFRZ# for the next cycle, are latched at this time on the falling edge of MEOC#.

MDOE# is deasserted by the MBC (clock 16) to allow the memory data pins to be used as inputs for the reissued read miss cycle.

8.3.2 CLOCKED SNOOPING MODE

Figure 8.8 illustrates a CPU initiated Read cycle which misses the 82495XP/82490XP cache and the subsequent line fill replaces non dirty data (eg. clean or empty). Simultaneous with the read request to the MBC, that device initiates a snoop to the 82495XP which misses that line in the cache. The snoop is the result of a write cycle on the memory bus by some other cache core; therefore, asserting the snoop invalidation signal (SNPINV) to this 82495XP. This example assumes Clocked Snooping Mode (i.e. the requests for snoops are done via SNPSTB# from the MBC, sampled on the MBC's SNPCLK).

CACHE CONTROL SIGNALS:

The CPU initiates the read cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#) in order to schedule the cache line-fill operation. MCACHE# is active, indicating that the read miss in potentially cacheable by the 82495XP; RDYSRC is active, indicating that the MBC must supply BRDY#s to the CPU cache core.

In clock 3, SNPSTB# is asserted by the MBC at this time, indicating to the 82495XP a request for snooping. MAOE# is deasserted to allow the forthcoming snoop (the 82495XP will not recognize the snoop if MAOE# is active). It is latched together with the snoop address (MSET[0:10], MTAG[0:11], MCFA[0:6]), SNPINV, MBAOE#, and SNPNCA on the MBC's SNPCLK rising edge during SNPSTB# assertion. SNPINV is asserted from the MBC since the cache core which initiated the snoop issued a write cycle on the memory bus. If the response of the snoop to this 82495XP was a cache hit, the contents would no longer be valid due that write.

Following synchronization to the 82495XP CLK, it issues SNPCYC# (clock 5), indicating that the snoop look-up is in progress. The results of the look-up are driven to the memory bus via MTHIT# and MHITM# in the next clock after SNPCYC#. Since the snoop was a miss in the cache, both signals are inactive (clock 6). Note that SNPBSY# will not be asserted since the snoop was a miss to this cache. The snoop from another cache is complete at this point, and the read miss cycle will continue.

The MBC asserts MAOE# to allow this 82495XP to drive its address on the memory bus in order to complete the read miss cycle. The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid after MAOE# assertion # (clock 6 for the read cycle in this example) and remains valid until after CNA# is sampled active by the 82495XP (clock 8). MALE and MBALE may be used to hold the address as necessary.

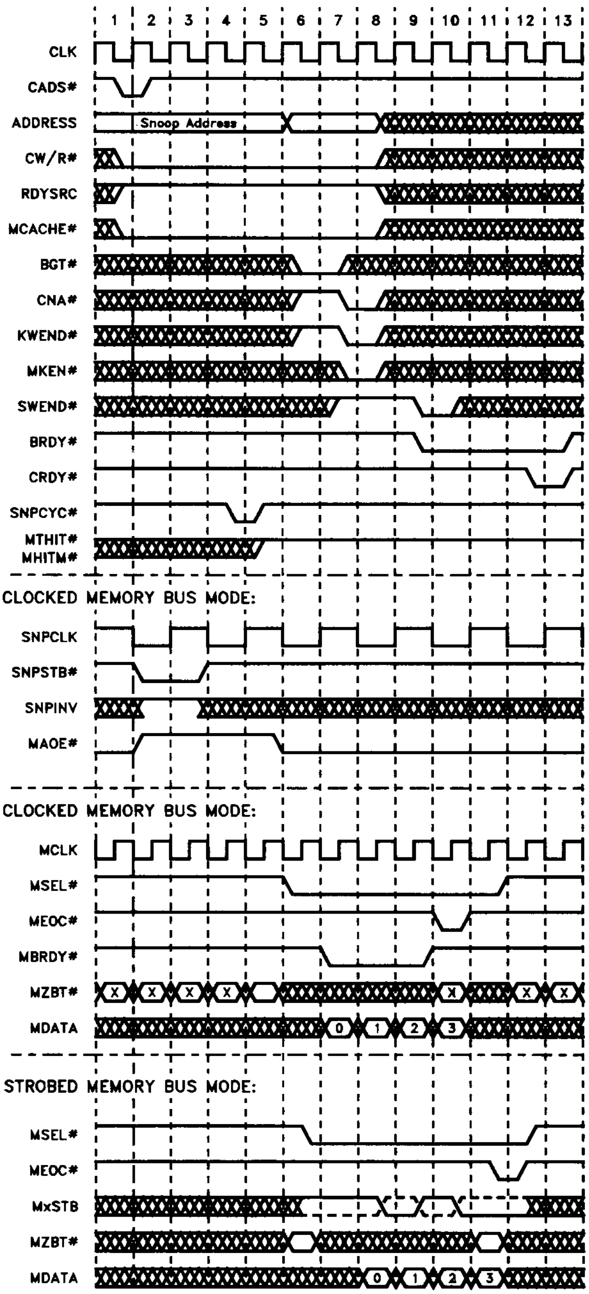
The MBC arbitrates for the memory bus and returns BGT# asserted (clock 6), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 7) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# during KWEND# (clock 7) to determine that the cycle is indeed cacheable.

The MBC asserts SWEND# when the snoop window ends on the memory bus. The 82495XP samples MWB/WT# during SWEND# (clock 9) and updates the cache tag state according to the consistency protocol. The closure of the snoop window also enables the MBC to start providing the CPU with data that has been stored in the 82490XP's memory cycle buffer. The MBC supplies BRDY#s to the CPU (clocks 9-12).

The read miss cycle ends when CRDY# is driven active by the MBC (clock 12). It is at this time that the data in the 82490XP's memory cycle buffers is loaded into the cache SRAM.



240956–40

Figure 8-8. Clocked Snooping Mode

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC. (Note the use of MAOE# for snooping at the beginning of the cache control signals section.) MDOE# must be inactive to allow the data pins to be used as inputs.

Some time after the address has been driven onto the memory bus, data will be supplied from the DRAM (main memory) to the 82490XP cache SRAM.

For Clocked Memory Bus Mode, MSEL# is driven active by the MBC (clock 6) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. MBRDY# is driven active by the MBC in clocks 7 to 9 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 10) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is sampled at this time (when MEOC# is sampled asserted and MSEL# remains low) for the next transfer.

For Strobed Memory Bus Mode, MSEL# is driven active by the MBC (clock 6) to allow MISTB operation and to latch MZBT# (on the falling edge of MSEL#) for the transfer. MISTB is toggled in clocks 8 to 10 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 11) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle, is sampled at this time on the falling edge of MEOC#.

8.3.3 STROBED SNOOPING MODE (HIT TO [M] LINE)

Figure 8.9 illustrates a snoop hit to a dirty line sequence occurring simultaneously with a CPU initiated read miss cycle. This example assumes strobed snooping mode (ie. requests for snoops are done from the falling edge of SNPSTB#).

CACHE CONTROL SIGNALS:

In clock 1 (totally asynchronous to any clock) SNPSTB# is asserted by the MBC, indicating to the 82495XP a request for snooping. The 82495XP samples MAOE# (it must be inactive) in order to recognize the snoop request. It is latched together with the snoop address (MSET[0:10], MTAG[0:11], MCFA[0:6]), SNPINV, MBAOE#, and SNPNCA on falling edge of SNPSTB#. The 82495XP issues SNPCYC# (clock 3), indicating that the snoop look-up is in progress. The results of the look-up are driven to the memory bus via MTHIT# and MHITM# in the next clock after SNPCYC#. Since the snoop hit a modified line, both signals are asserted (clock 4). SNPBSY# is also issued to indicate that the 82495XP is busy with CPU back-invalidations, the 82490XP's snoop buffer is full, or a write back is to follow. The 82495XP will accept snoops only when SNPBSY# is inactive.

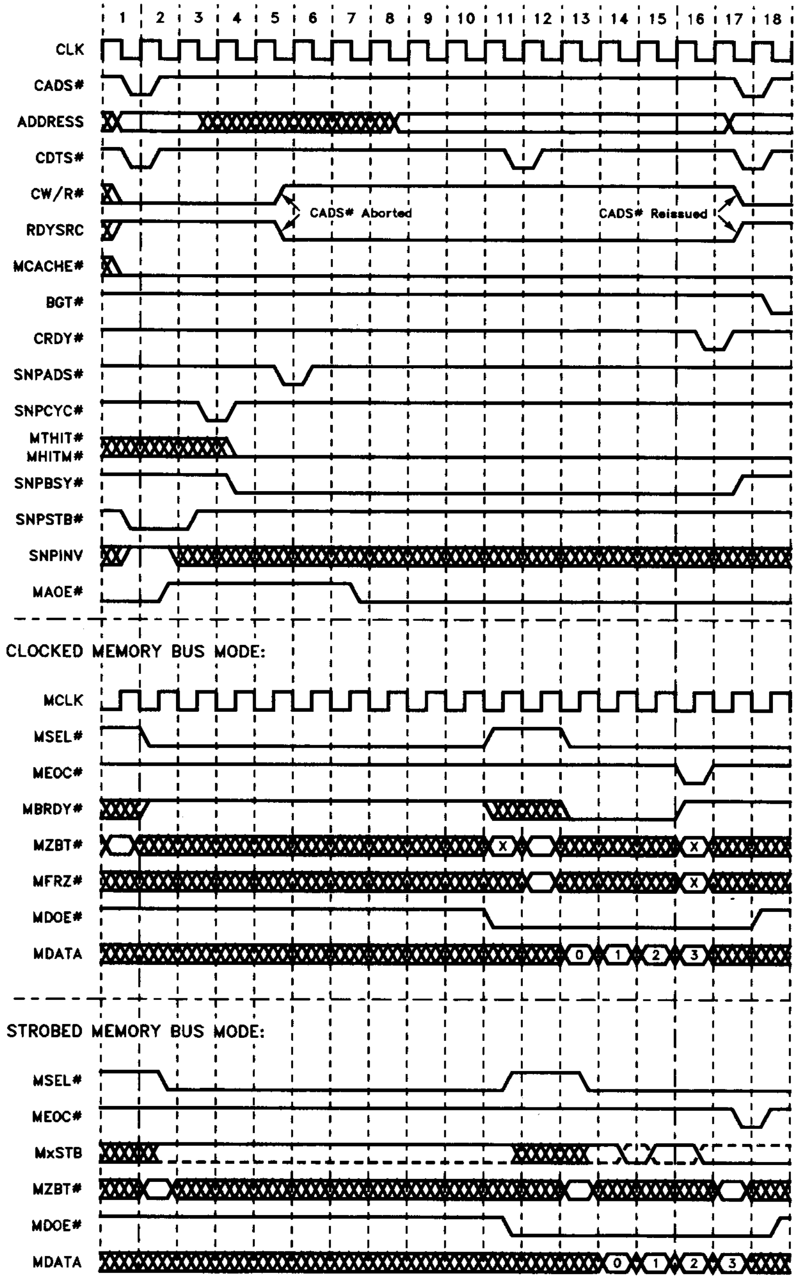
Simultaneously with the memory bus activity due to the snoop request, the CPU initiates a read miss cycle. The 82495XP issues a memory bus request (CADS#), CDTS#, and cycle control signals to the MBC in clock 1. The MBC must wait for the pending snoop cycle to complete on the memory bus prior to servicing this read miss cycle.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is not valid until MAOE# goes active after CRDY# of the snoop write back cycle is sampled active by the 82495XP and the CADS# is reissued (clock 17).

In clock 5 the 82495XP issues SNPADS# and cycle control signals to the MBC, indicating a request to flush a modified line out of the cache. SNPADS# activation causes the MBC to abort the pending read miss cycle. It is the 82495XP responsibility to re-issue the aborted cycle after the completion of the write back, since BGT# was not asserted by the MBC.

Data is loaded into the 82490XP's snoop buffer. Since SNPINV was sampled asserted by the 82495XP (clock 1) during SNPSTB# assertion, it back-invalidated the CPUs first level cache.

The 82495XP asserts CDTS# (clock 11) indicating to the MBC that data is available in the snoop buffer. When the MBC complete the write back cycle on the memory bus, it activates CRDY# to the 82495XP/82490XP cache. At this time, the 82495XP deasserts SNPBSY# (clock 17) and re-issues the aborted read miss cycle by asserting CADS# and CDTS#.



240956–41

Figure 8-9. Strobed Snooping Mode

MEMORY BUS SIGNALS:

For Clocked Memory Bus Mode, the memory data output enable (MDOE#) is not activated by the MBC to allow the memory data pins to be used as inputs.

MSEL# is driven active by the MBC (clock 2) to allow sampling of MBRDY# and to latch MZBT# for the read miss transfer. MZBT# is sampled on all MCLK rising edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer.

Since the read miss cycle is aborted due to the snoop hit to a modified line (requires a write back cycle), no MEOC# is given. MSEL# is deasserted by the MBC (clock 11) and reasserted (clock 13) to allow latching of MZBT# for the snoop write back cycle and sampling of MBRDY# for that cycle. MFRZ# is also sampled at this time.

The memory data output enable (MDOE#) signal is driven active by the MBC (clock 11) to drive the memory data outputs.

MBRDY# is driven active by the MBC in clocks 13 to 15 to cause the memory burst counter to be incremented and data to be written from the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 16) to end the write back cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# and MFRZ# are sampled and sampled at this time for the next data transfer.

MDOE# is deasserted by the MBC (clock 18) to allow the memory data pins to be used as inputs for the reissued read cycle.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) has not been asserted by the MBC to allow the memory data pins to be used as inputs for the read miss cycle.

MSEL# is asserted by the MBC (clock 2) to allow sampling of MISTB and latch MZBT# (on the falling edge of MSEL#) for the read miss transfer.

Since the read miss cycle is aborted due to the snoop hit to a modified line (requires a write back cycle), no MEOC# is given. MSEL# is deasserted by the MBC (clock 11) and reasserted (clock 13) to allow latching of MZBT# for the snoop write back cycle and sampling of MOSTB for that cycle. MFRZ# is also sampled at this time.

MOSTB is toggled in clocks 14 to 16 to cause the memory burst counter to be incremented, and data to be read from the 82490XP cache memory cycle buffers. Note: MOSTB latches the memory bus data on both the rising and falling edges.

The MBC drives MEOC# asserted (clock 17) to end the snoop write back cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# and MFRZ# for the next cycle, are sampled at this time on the falling edge of MEOC#.

MDOE# is deasserted by the MBC (clock 18) to allow the memory data pins to be used as inputs for the reissued read miss cycle.

8.3.4 CACHE TO CACHE TRANSFER

8.3.4.1 Read Cycles Causing a Snoop Hit to [M] Line

Figure 8.10 illustrates CPU initiated Read cycles that miss the 82495XP/82490XP cache and replace a non-dirty (eg. clean) line in the cache. During the snoop window, the memory bus attribute which indicates a direct to [M] state transfer is sampled active. In such cycles, the 82495XP will instruct the MBC to perform a cache line-fill cycle on the memory bus. The request for data will not go to main memory, but instead will go to the controller of the cache which contained the modified data. The line is then written into the 82490XP's array, and data transferred to the CPU as requested. If the line fetched from the second cache replaces a line which is in valid unmodified state ([E] or [S]), then a back-invalidation cycle is performed on the CPU bus to guarantee that the replaced data is also removed from the CPU's first level cache, thus maintaining the inclusion property.

CACHE CONTROL SIGNALS:

The CPU initiates the read cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 2) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#) in order to schedule the cache line-fill operation. MCACHE# is active, indicating that the read miss is potentially cacheable by the 82495XP; RDYSRC is active, indicating that the MBC must supply BRDY#s to the CPU cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 2 and 13 for the two read miss cycles in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 5 and 16). MALE and MBALE may be used to hold the address as necessary.

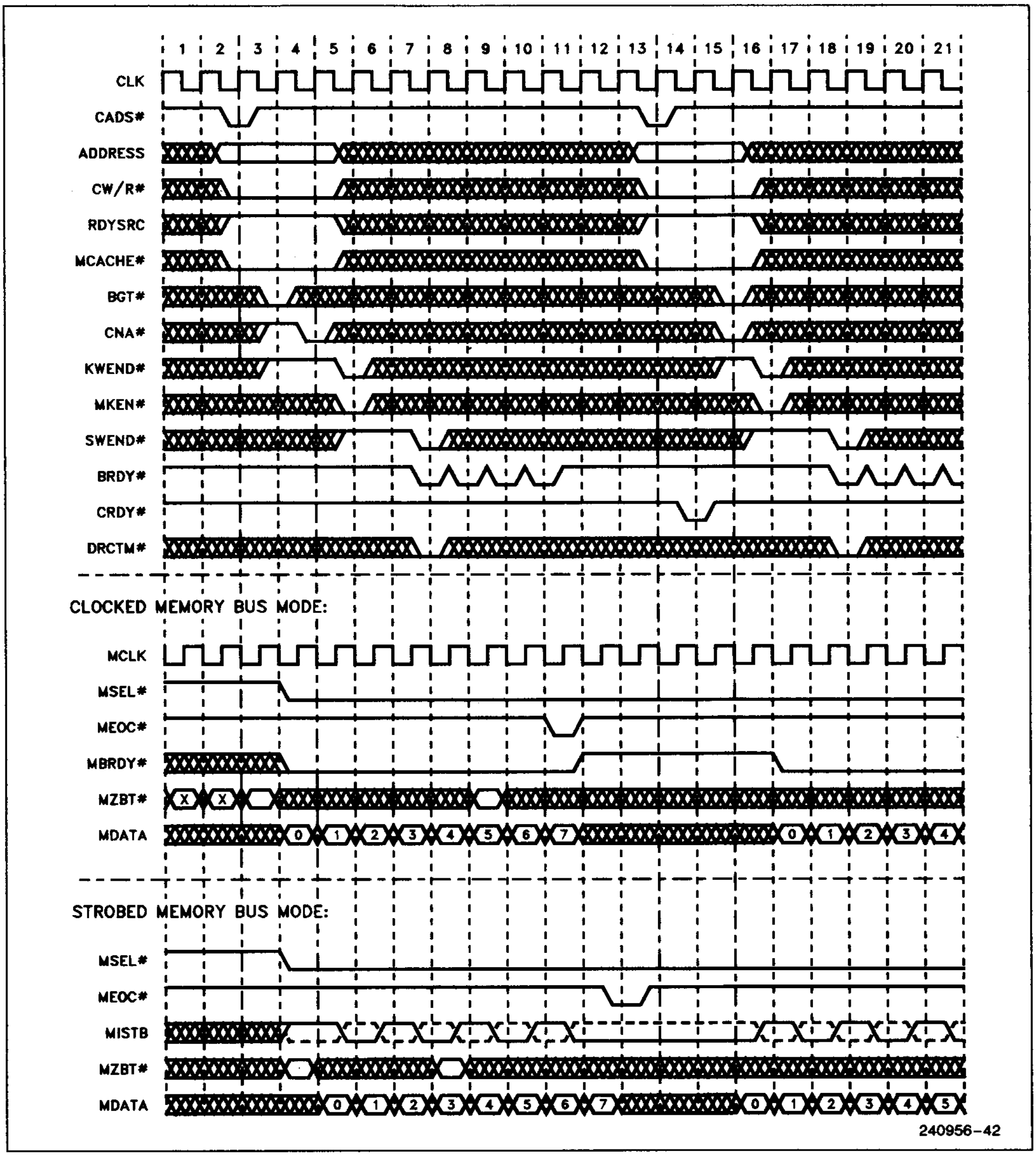


Figure 8-10. Cache to Cache Transfer: Cacheable Read Miss

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 3), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 4) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# and MRO# during KWEND# (clock 5) to determine that the cycle is indeed cacheable.

The MBC asserts SWEND# when the snoop window ends on the memory bus. The 82495XP samples MWB/WT# and DRCTM# during SWEND# (clock 7) and updates the cache tag state according to the consistency protocol. Since the result of the snoop was a hit to a modified line in another cache, the MBC asserts DRCTM# at this time (this is an option to save time by skipping the main memory access, not a requirement of the memory bus) so that the tag state will go immediately to the [M] state, skipping the [E] state. MWB/WT# must be in write back mode (high) to assure this transition. The closure of the snoop window also enables the MBC to start providing the CPU with data that has been stored in the 82490XP's memory cycle buffer. The MBC supplies BRDY#s to the CPU (clocks 7-10).

The 82495XP issues a new CADS# in clock 13, which also misses the 82495XP/82490XP cache. Since the 82495XP has already sampled CNA# asserted (clock 4), It issues a new CADS# prior to receiving CRDY# of the current cycle (ie. this cycle is pipelined within the MBC). Note that once the cycle progress signals (BGT#, CNA#, KWEND#, SWEND#) of a cycle are sampled asserted, the 82495XP ignores them until the CRDY# of that cycle. The 82495XP does not pipeline the cycle progress signals (ie. it will not sample them again until after CRDY# of the current memory bus cycle).

MEMORY BUS CYCLES:

At the start of this cycle, the master 82495XP does not know that the data will be coming from a slave 82495XP/82490XP and begins a read request to main memory to obtain the required data. Since the snoop resulted in a hit to a modified line in the second cache, the memory request must be backed off so that the snooped 82495XP may supply the data.

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC. The memory data output enable signal (MDOE#) must remain inactive to allow the data pins to be used as inputs.

For Clocked Memory Bus Mode, MSEL# is driven active by the MBC (clock 4) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer.

MBRDY# is driven active in clocks 4 to 10 to read data into the 82490XP cache memory cycle buffers. The MBC asserts MEOC# (clock 11) to end the read miss cycle on the memory bus and switch the memory cycle buffers for a new cycle. MZBT# is latched at this time for the next transfer. Note that there are 8 transfers needed to fill the 82495XP/82490XP cache line and only 4 needed for the CPU line fill.

MBRDY# is again driven active by the MBC in clocks 11 to 21 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers for the second read miss cycle.

For Strobed Memory Bus Mode, MSEL# is driven active by the MBC (clock 4) to allow MISTB operation and to latch MZBT# for the transfer (on the falling edge of MSEL#). MISTB is toggled in clocks 5 to 11 to cause the memory burst counter to be incremented, and data to be placed into the 82490XP cache memory cycle buffers. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 12) to end the current cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# for the next cycle is latched at this time on the falling edge of MEOC#.

The MBC toggles MISTB (clocks 16 to 21) for the second read miss cycle to increment the memory burst counter and cause data to be written into the 82490XP memory cycle buffers.

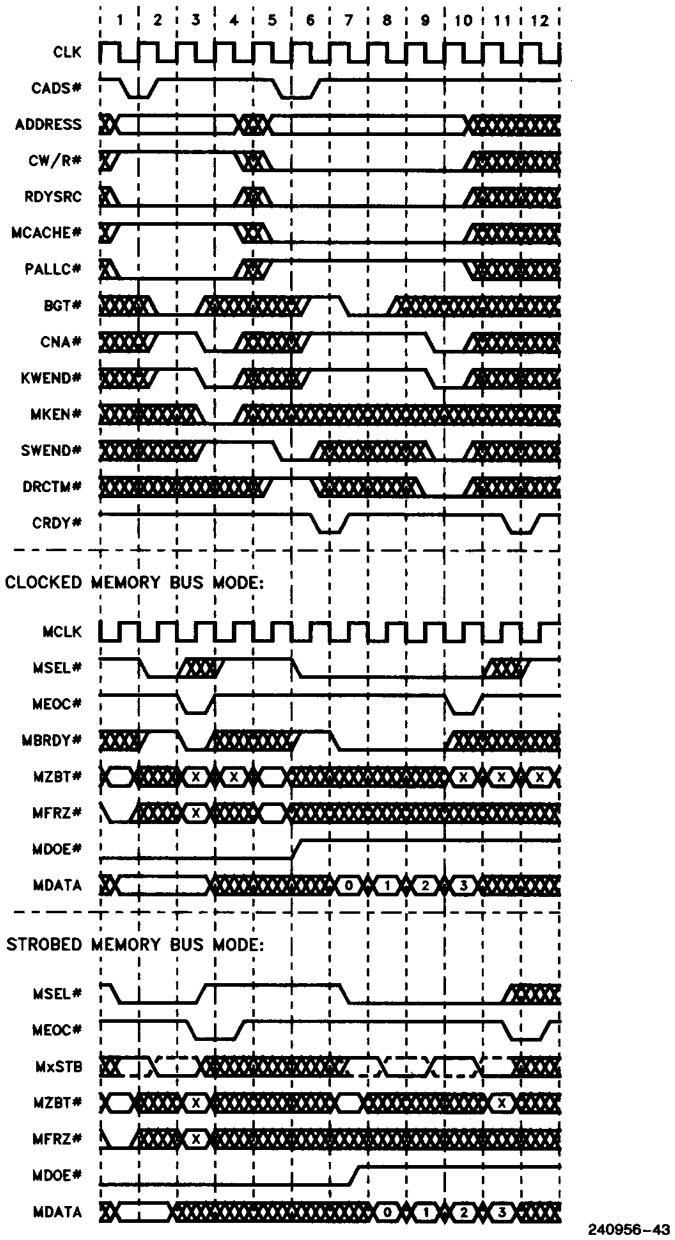


Figure 8-11. Read For Ownership

8.4 Read for Ownership

8.4.1 WRITE MISS WITH MFRZ# ASSERTED, FOLLOWED BY READ TO SAME LINE

Figure 8-11 illustrates a Read For Ownership cycle. First, a CPU initiates a write cycle which misses the 82495XP/82490XP cache. The MBC issues a "dummy" write to main memory (the write does not actually go out to main memory - to save valuable bus time). The 82490XP MFRZ# input is used by the MBC to indicate that the following line-fill (allocation) data (from either main memory or another cache) should be merged with the data of the write miss. The entire line is then placed into the internal tagram.

CACHE CONTROL SIGNALS:

The CPU initiates a write cycle to the 82495XP/82490XP cache where the cache tag state is looked up. Once the 82495XP determines the cycle to be a cache miss, it issues CADS# (clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#, PALLC#) in order to schedule the write operation. MCACHE# is not active; RDYSRC is not active, indicating that the 82495XP will supply BRDY#s to the CPU; PALLC# is active, indicating a potential allocate cycle after the write through cycle.

The write miss data is posted in the 82490XP's memory cycle buffer, and the cycle completes with no wait states to the CPU. The CPU is free to issue another (non-related) cycle while the 82495XP is processing the allocation. If this new cycle is a cache hit, it will be serviced by the 82495XP immediately; but if it is a cache miss, its service will wait until the CRDY# of the allocation.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 5 for the write miss and allocation cycle in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 4 and 10). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the write through cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# is asserted by the MBC (clock 3) to indicate that it is ready to schedule a new memory bus cycle. Note that after CNA# activation, cycle control signals are not guaranteed to be valid.

When the MBC has determined the cacheability attribute of the write through cycle, it drives the MKEN# signal accordingly. The MBC also drives the KWEND# signal at this time, indicating the end of the cacheability window. The 82495XP samples MKEN# active during KWEND# (clock 3), indicating that the missed line should be allocated in the cache.

The MBC asserts SWEND# (clock 5) when the snoop window of the write through cycle ends on the memory bus. Note that the direct to [M] state qualifier signal (DRCTM#) is sampled during SWEND# and is inactive for the write . The MBC also issued CRDY# to the 82495XP at this time so that the 82495XP thinks the write cycle completed on the memory bus when, in fact, it did not.

In this example, the 82495XP requests the allocation cycle by issuing CADS# in clock 5. The cycle control signals are valid at this point: MCACHE# is active, indicating the cacheability of the line-fill cycle; RDYSRC is not active, indicating that the MBC need not supply BRDY#s to the CPU (no BRDY#s are necessary for an allocation cycle).

Once again, the MBC arbitrates for the memory bus and returns BGT# asserted (clock 7) for the allocation cycle. The MBC asserts CNA#, KWEND#, and SWEND# (clock 9) to pipeline the memory bus and close the cacheability and snoop windows. Note that (for this example) DRCTM# is asserted during SWEND# to place the line in the modified state. Since this is done, all other caches must invalidate their copies.

CRDY# for the allocation (line-fill) cycle is issued by the MBC in clock 11 to complete the read cycle on the memory bus and place the data into the 82490XP cache array.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in the flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, the memory data output enable (MDOE#) has been asserted by the MBC to drive the memory data outputs.

The MBC asserts MSEL# (clock 2) to allow sampling of MBRDY# and to latch MZBT# and MFRZ# for the write. MBRDY# and MEOC# are asserted by the MBC (clock 3) to place the write data into the memory cycle buffers, sample MZBT# and MFRZ# for the next transfer, and end the current cycle on the memory bus. MFRZ# is driven active by the MBC here, indicating to the 82495XP that the data of the write through will be merged with the following allocation data.

For the allocation (line fill) cycle, MSEL# is driven active again by the MBC (clock 6) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. MDOE# is also deasserted in clock 6 to allow the data pins to be used as inputs for the allocation cycle.

MBRDY# is driven active by the MBC in clocks 7 to 9 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. During the line fill, the 82490XP will merge the data from the write through buffer with the incoming data from either main memory or another cache (if that line was a write hit to [M] in another cache).

The MBC drives MEOC# asserted (clock 10) to end the allocation cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is sampled at this time for the next data transfer.

For Strobed Memory Bus Mode, the memory data output enable (MDOE#) has been asserted by the MBC to drive the memory data outputs.

The MBC asserts MSEL# (clock 2) to allow toggling of MISTB and to latch MZBT# and MFRZ# for the write (on MSEL# falling edge). MISTB is toggled and MEOC# asserted by the MBC (clock 2) to place the write data into the memory cycle buffers, sample MZBT# and MFRZ# for the next transfer (on the falling edge of MEOC# while MSEL# is active), and end the current cycle on the memory bus. MFRZ# is driven active by the MBC here, indicating to the 82495XP that the data of the write through will be merged with the following allocation data.

For the allocation (line fill) cycle, MSEL# is driven active again by the MBC (clock7) to allow sampling of MOSTB and to latch MZBT# for the transfer. MDOE# is also deasserted in clock 7 to allow the data pins to be used as inputs for the allocation cycle.

MOSTB is toggled by the MBC in clocks 8 to 10 to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. During the line fill, the 82490XP will merge the data from the write through buffer with the incoming data from either main memory or another cache (if that line was a write hit to [M] in another cache).

The MBC drives MEOC# asserted (clock 11) to end the allocation cycle on the memory bus and switch memory cycle buffers for the new cycle. MZBT# is sampled at this time for the next data transfer.

8.5 I/O Cycles

Figure 8-12 illustrates CPU initiated I/O cycles, both read and write. I/O writes are the only write cycles not posted by the 82495XP/82490XP cache (ie. the cycle is not fully acknowledged to the CPU until it has completed on the memory bus).

CACHE CONTROL SIGNALS:

The CPU initiates an I/O write cycle to the 82495XP/82490XP. The 82495XP then issues CADS# and CDTS# (clock 1) and the associated cycle control signals to the MBC (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#). MCACHE# in not active, indicating that the cycle is not cacheable; RDYSRC is active, indicating that the MBC must supply BRDY#s to the CPU/Cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 10 for the two read s in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 6 and 17). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2) for the I/O write cycle, indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# for the write cycle is asserted by the MBC (clock 5) to indicate that it is ready to schedule a new memory bus cycle. Note that SWEND# and KWEND# are not needed for I/O cycles since they are not cacheable.

The MBC asserts BRDY# in clock 7 to complete the I/O write cycle from the CPU, and CRDY# in clock 8 to complete the cycle on the memory bus from the 82495XP/82490XP cache.

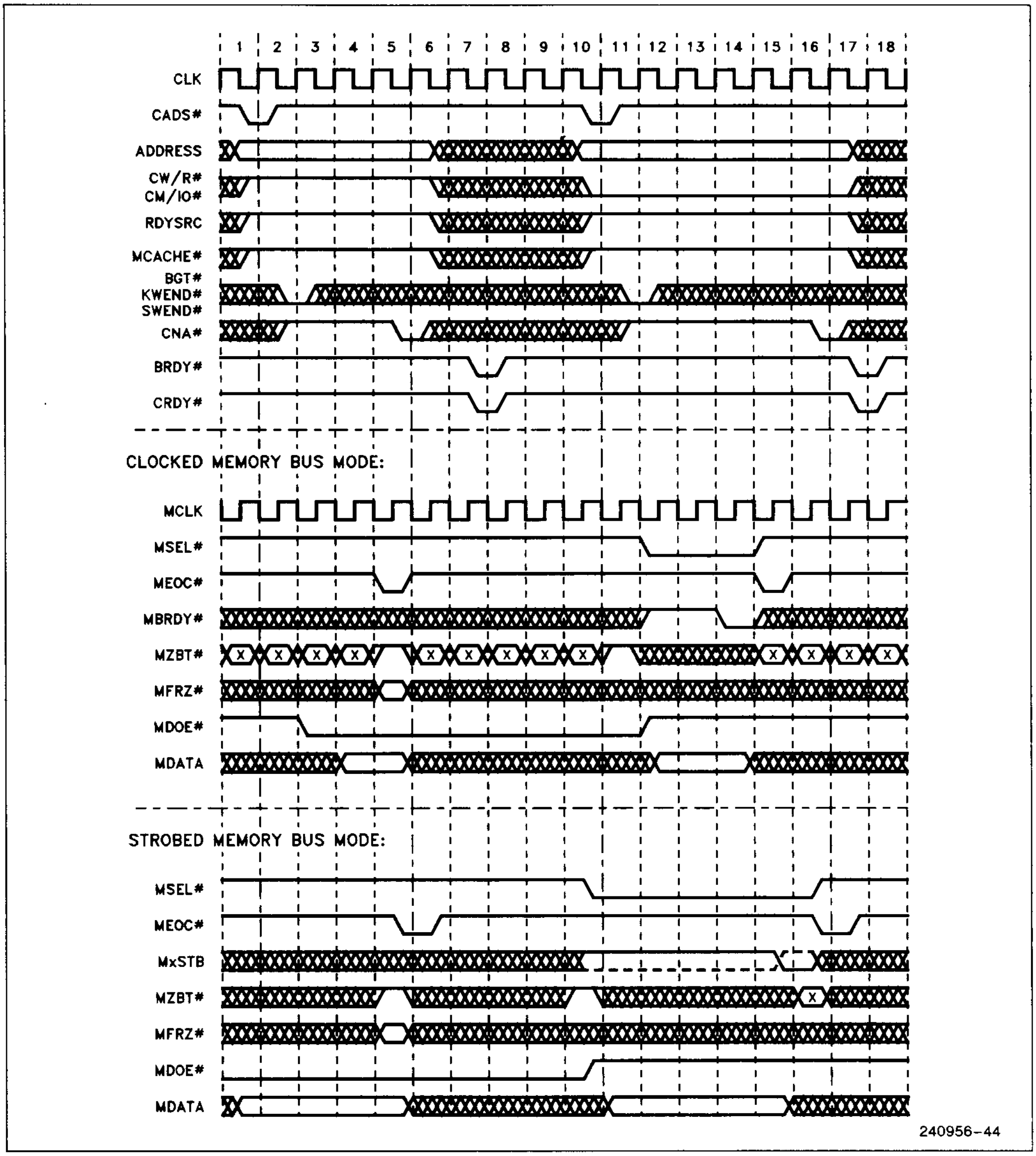


Figure 8-12. I/O Write and Read Cycles

A new CADS# is issued from the 82495XP in clock 10 for an I/O read cycle, along with the associated cycle control signals. MCACHE# is again not active, and RDYSRC is again active.

The MBC returns BGT# asserted right away (clock 11). The 82495XP can pipeline I/O cycles, but does not for the I/O read in this example.

Upon completing the access on the memory bus, the MBC activates BRDY# (clock 17) and CRDY# (clock 16). Note that BRDY# of a cycle may come before (as in the I/O write cycle of this example), with or after the CRDY# of the same cycle.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, The memory data output enable signal (MDOE#) is asserted by the MBC in clock 3 to drive the memory data outputs.

MEOC# is asserted by the MBC (clock 5) to latch MZBT# for the I/O write transfer, and end that cycle on the memory bus (MBRDY# is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the write cycle to begin with the correct burst address. MFRZ# is also sampled here (it need not be active since the cycle is not potentially allocatable).

For the I/O read cycle, MDOE# is deasserted (clock 12) by the MBC to allow the data pins to be used as inputs.

MSEL# is driven active by the MBC (clock 12) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer. Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address.

The MBC asserts MBRDY# (clock 14) to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# asserted (clock 15) to end the read cycle on the memory bus and switch memory cycle buffers for a new cycle. MZBT# for the next transfer is latched at this time.

For Strobed Memory Bus Mode, The memory data output enable signal (MDOE#) has been asserted by the MBC to drive the memory data outputs.

MEOC# is asserted by the MBC (clock 5) to latch MZBT# for the I/O write transfer (on MEOC# falling edge), and end that cycle on the memory bus (MOSTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the write cycle to begin with the correct burst address. MFRZ# is also sampled here (it need not be active since the cycle is not potentially allocatable).

For the I/O read cycle, MDOE# is deasserted (clock 10) by the MBC to allow the data pins to be used as inputs.

MSEL# is driven active by the MBC (clock 10) to allow operation of MISTB and to latch MZBT# for the transfer (on MSEL# falling edge). Again, MZBT# is driven high by the MBC to force the transfer to begin with the correct burst address.

The MBC toggles MISTB (clock 15) to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers for the I/O read cycle. Note: MISTB latches the memory bus data on both the rising and falling edges. The MBC drives MEOC# asserted (clock 16) to end the read cycle on the memory bus and switch memory cycle buffers for a new cycle. MZBT# for the next transfer is latched at this time (on the falling edge of MEOC#).

8.6 LOCKed Cycles

8.6.1 CPU READ MODIFY WRITE CYCLES

The 82495XP provides a facility to allow atomic accesses requested by the CPU (via CPU LOCK# activation) through the 82495XP KLOCK# signal. Figure 8-13 illustrates two back-to-back CPU initiated Locked read-modify-write cycles. KLOCK# activation indicates to the MBC that the memory bus should not be released between the KLOCKed cycles. KLOCK# will remain asserted from the beginning of the first cycle (with CADS#) until one clock after the CADS of the last cycle. The 82495XP does not distinguish between back-to-back locked operations and will not open an arbitration window (deassert KLOCK#) between them. It is the responsibility of the MBC to distinguish between the multiple RMW sequences, if it is so desired.

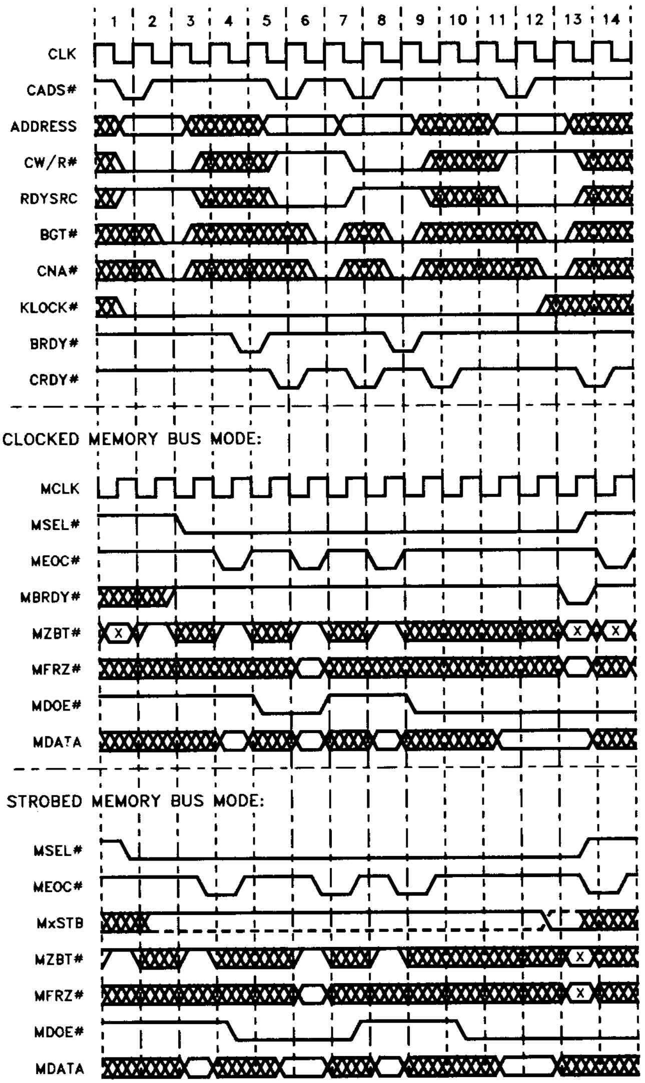


Figure 8-13. LOCKed Read-Modify-Write Cycles

The 82495XP issues a request for a memory bus access (CADS#) for every locked cycle (read or write) regardless if it hits the cache tag state or not. Locked read cycles are treated by the 82495XP as cache misses, and , if the line is in the [M] state, the 82495XP ignores the data on the memory bus and uses the data in the 82490XP array. Locked write cycles are treated as write through, and the tag state does not change even if the line is in the 82490XP array.

CACHE CONTROL SIGNALS:

The CPU initiates a Locked read cycle to the 82495XP/82490XP cache where, due to the assertion of CPU LOCK#, it assumes a cache miss and issues CADS# to the MBC (clock 1) along with the associated cycle control signals (eg. CW/R#, CM/IO#, CD/C#, RDYSRC, MCACHE#). MCACHE# is never asserted for LOCKed cycles; RDYSRC is active, indicating that the MBC must supply BRDY# to the CPU/Cache core.

The memory bus address (MSET[10:0], MTAG[11:0], MCFA[6:0]) is valid with CADS# (clocks 1 and 5, then 7 and 11 for the two locked RMW sequences in this example) and remain valid until after CNA# is sampled active by the 82495XP (clocks 3 and 7, then 9 and 13). MALE and MBALE may be used to hold the address as necessary.

The MBC arbitrates for the memory bus and returns BGT# asserted (clock 2), indicating that the cycle is guaranteed to complete on the memory bus. Once the 82495XP samples BGT# asserted, it must finish that cycle on the memory bus. Prior to this point, the cycle can be aborted by a snoop hit from another cache.

CNA# for the read cycle is also asserted by the MBC (clock 2) to indicate that it may schedule a new memory bus cycle. Note that the cycle control signals are not guaranteed to be valid after CNA# activation.

The MBC asserts BRDY# to the CPU/Cache core in clock 4. CRDY# for the locked read cycle is asserted to the 82495XP/82490XP from the MBC (clock 5) to load the data stored in the 82490XP's memory cycle buffers into the cache array. If the read was to a dirty line, the 82495XP is intelligent enough to ignore the data in the memory cycle buffers and use the data in the cache array.

Locked sequences always end in a write cycle, no new CPU initiated cycles may be inserted between the Locked read and Locked write cycles. Therefore, the 82495XP issues a new memory cycle request (CADS# in clock 5) for the Locked write as soon as it completes the Locked read cycle. The cycle control signals are also valid at this time. RDYSRC is not active, indicating that the 82495XP will supply BRDY# to the CPU.

The locked write cycle is posted like any other memory write cycle.

In this example, the CPU initiates a second read-modify-write cycle immediately. KLOCK# is not deasserted between the back-to-back locked sequences since the CPU LOCK# remains asserted. If snooping is required between these cycles, it is the MBC responsibility to predict this boundary and allow snooping. The 82495XP issues a memory bus request (CADS#) in clock 7 for the second locked read cycle, along with the new cycle control signals.

The second locked RMW sequence repeats the actions of the first. It's purpose in this example is to demonstrate that an arbitration window may not open between locked sequences if they follow one another with no idle or non-locked cycles between them.

MEMORY BUS SIGNALS:

The memory address latch enables (MALE and MBALE) may remain asserted by the MBC to place the address latches in flow through mode. If the 82495XP is the current bus master, the memory address output enables (MAOE# and MBAOE#) should be asserted by the MBC.

For Clocked Memory Bus Mode, MSEL# is driven active by the MBC (clock 3) to allow sampling of MBRDY# and to latch MZBT# for the transfer. MZBT# is sampled on all MCLK edges where MSEL# is inactive. Once MSEL# is sampled active by the 82495XP, the value of MZBT# sampled on the prior MCLK is used for the next transfer.

The memory data output enable signal (MDOE#) must be inactive to allow the data pins to be used as inputs for the first locked read cycle. The MBC asserts MEOC# (clock 4) to latch MZBT# for the next transfer, and end the current locked read cycle on the memory bus (MBRDY# is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address.

For the locked write cycle, MDOE# is asserted by the MBC (clock 5) to drive the memory data outputs.

MEOC# is again asserted (clock 6) to latch MZBT# for the next transfer, and end the current locked write cycle on the memory bus (MBRDY# is not necessary since this is a single transfer cycle). MZBT# is again driven high. MFRZ# is also sampled during write cycles when MEOC# is sampled active by the 82495XP.

MDOE# is deasserted by the MBC (clock 7) to allow the data pins to be used as inputs for the second locked read cycle. MEOC# is again asserted (clock 8) to latch MZBT# for the next transfer, and end the locked read cycle on the memory bus. MZBT# is again driven high.

MDOE# is asserted by the MBC (clock 9) to drive the memory data outputs for the second locked write cycle. MBRDY# is asserted (clock 13) to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# active and MSEL# inactive (clock 14) to end the locked write cycle on the memory bus and switch memory cycle buffers for a new cycle. MZBT# and MFRZ# for the next transfer are sampled at this time.

For Strobed Memory Bus Mode, MSEL# is driven active by the MBC (clock 1) to allow sampling of MxSTB and to latch MZBT# for the first locked read transfer (on the falling edge of MSEL#).

The memory data output enable signal (MDOE#) must be inactive to allow the data pins to be used as inputs for the first locked read cycle. The MBC asserts MEOC# (clock 3) to latch MZBT# for the next transfer (on MEOC# falling edge while MSEL# is active), and end the current locked read cycle on the memory bus (MISTB is not necessary since this example shows a single transfer cycle). MZBT# is driven high by the MBC in order to force the read cycle to begin with the correct burst address.

For the locked write cycle, MDOE# is asserted by the MBC (clock 4) to drive the memory data outputs. MEOC# is again asserted (clock 6) to latch MZBT# for the next transfer, and end the current locked write cycle on the memory bus (MOSTB is not necessary since this is a single transfer cycle). MZBT# is again driven high. MFRZ# is also sampled on the falling edge of MEOC#.

MDOE# is deasserted by the MBC (clock 7) to allow the data pins to be used as inputs for the second locked read cycle. MEOC# is again asserted (clock 8) to latch MZBT# for the next transfer, and end the locked read cycle on the memory bus. MZBT# is again driven high.

MDOE# is asserted by the MBC (clock 9) to drive the memory data outputs for the second locked write cycle. MOSTB is toggled (clock 12) to cause the memory burst counter to be incremented and data to be placed into the 82490XP cache memory cycle buffers. The MBC drives MEOC# active and MSEL# inactive (clock 13) to end the locked write cycle on the memory bus and switch memory cycle buffers for a new cycle. MZBT# and MFRZ# for the next transfer are sampled at this time.

9.0 TESTABILITY

Testing the 82495XP/82490XP chipset can be divided into three categories: Built-In Self Test (BIST), Boundary Scan, and external testing. BIST performs basic device testing on the 82495XP. Boundary Scan provides additional test hooks that conform to the IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std.1149.1). Additional testing can be performed by using software written to test the 82490XP cache SRAM.

9.1 Built-In Self Test (BIST)

BIST tests the internal functionality of the 82495XP. The 82495XP's BIST tests approximately 90% of the cache controller. It tests the tag RAM and comparators.

The 82495XP BIST is initiated by driving SLFTST#(CRDY#) low and HIGHZ#(MBALE) high at least 10 clocks before RESET goes inactive. The 82495XP Cache Controller reports the result of BIST on the CAHOLD signal. When the self test completes, the 82495XP drives FSIOUT# inactive and the BIST result on CAHOLD. If CAHOLD is driven active the BIST successfully passed. If CAHOLD is driven inactive, BIST detected a flaw in the cache controller. CAHOLD is valid for one clock after FSIOUT# deactivation and should be sampled on the rising edge of FSIOUT#.

On the 82495XP, BIST only informs the system that a failure did or did not occur. BIST is not able to indicate where a failure occurred. After completing BIST the cache controller perform reset and begin normal operation.

9.2 Boundary Scan

The 82495XP/82490XP chipset provides additional test ability features compatible with the IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std.1149.1). The test logic provided allows for testing to insure that components function correctly, that interconnections between various components are correct, and that various components interact correctly on the printed circuit board.

The boundary scan test logic consists of a boundary scan register and support logic that are accessed through a test access port (TAP). The TAP provides a simple serial interface that makes it possible to test all signal traces with only a few probes.

The TAP can be controlled via a bus master. The bus master can be either automatic test equipment or a component (PLD) that interfaces to the four-pin test bus.

9.2.1 BOUNDARY SCAN ARCHITECTURE

The boundary scan test logic contains the following elements:

- Test access port (TAP), consisting of input pins TMS, TCK, and TDI; and ouput pin TDO.
- TAP controller, which interprets the inputs on the test mode select (TMS) line and performs the corresponding operation. The operations performed by the TAP include controlling the instruction and data registers within the component.
- Instruction register (IR), which accepts instruction codes shifted into the test logic on the test data input (TDI) pin. The instruction codes are used to select the specific test operation to be performed or the test data register to be accessed.
- Test data registers: The 82495XP/82490XP chipset components each contain three test data registers: Bypass register (BPR), Device Identification register (DID), and Boundary Scan register (BSR).

The instruction and test data registers are separate shift-register paths connected in parallel and have a common serial data input and a common serial data output connected to the TAP signals, TDI and TDO, respectively.

9.2.2 DATA REGISTERS

The 82495XP and 82490XP both contain the two required test data registers; bypass register and boundary scan register. In addition, they also have a device identification register.

Each test data register is serially connected to TDI and TDO, with TDI connected to the most significant bit and TDO connected to the least significant bit of the test data register. Data is shifted one stage (bit position within the register) on each rising edge of the test clock (TCK).

9.2.2.1 Bypass Register

The Bypass Register is a one-bit shift register that provides the minimal length path between TDI and TDO. This path can be selected when no test operation is being performed by the component to allow rapid movement of test data to and from other components on the board. While the bypass register is selected data is transferred from TDI to TDO without inversion.

9.2.2.2 Boundary Scan Register

The Boundary Scan Register is a single shift register path containing the boundary scan cells that are connected to all input and output pins of the 82495XP/82490XP chipset. Figure 9.1 shows the logical structure of the boundary scan register. While output cells determine the value of the signal driven on the corresponding pin, input cells only capture data; they do not affect the normal operation of the device. Data is transferred without inversion from TDI to TDO through the boundary scan register during scanning. The boundary scan register can be operated by the EXTEST and SAMPLE instructions. The boundary scan register order is described in section 9.2.5.

9.2.2.3 Device Identification Register

The Device Identification Register contains the manufacturer's identification code, part number code, and version code in the format shown in Figure 9.2. Table 9.1 lists the codes corresponding to the 82495XP and 82490XP.

Table 9-1. Device ID Register Values

Component Code	Version Code	Part Number Code	Manufacturer Identity
82495XP (A0 or A1) 0Ah	0495h	0495h	09h
82495XP (B0)	0Bh	0495h	09h
82490XP (A0 or A1)	00h	49A0h	09h

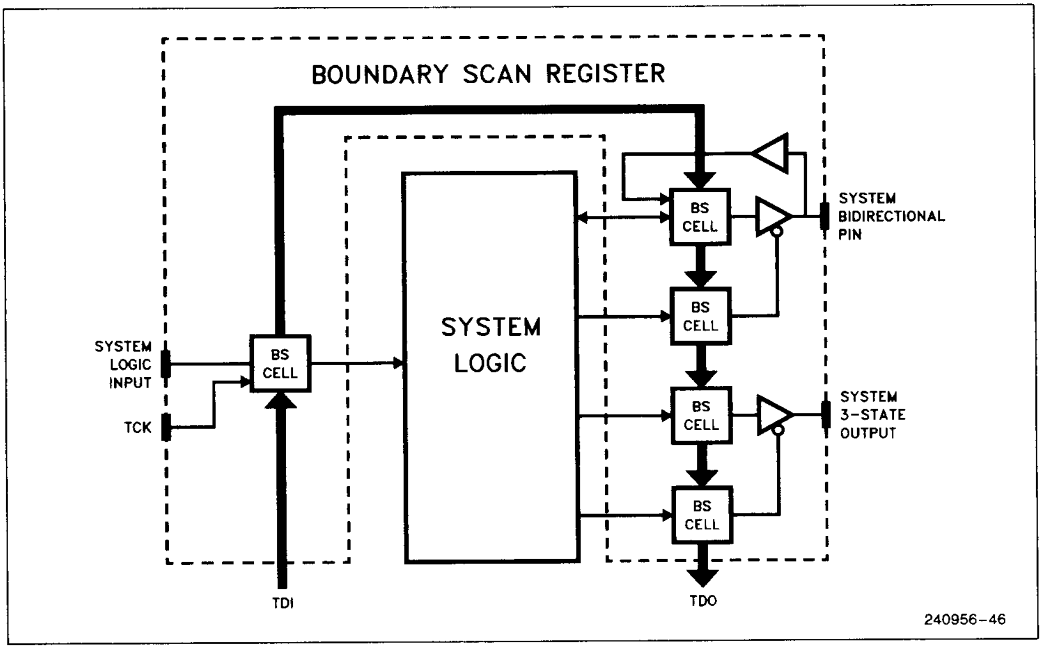


Figure 9-1. Boundary Scan Register Structure

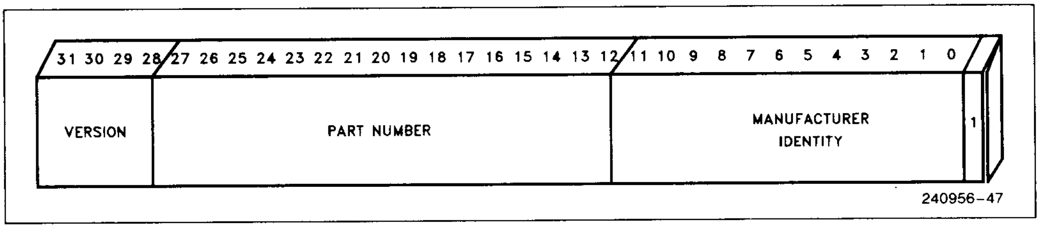


Figure 9-2. Device ID Register

9.2.2.4 Runbist Register

The Runbist Register is a one bit register used to report the results of the 82495XP BIST when it is initiated by the RUNBIST instruction. This register is loaded with a "1" prior to invoking the BIST and is loaded with "1" upon successfull completion. "0" indicates a failure occurred during BIST.

NOTE:
82495XP RUNBIST is not available in the A-stepping.

9.2.3 INSTRUCTION REGISTER

The Instruction Register (IR) allows instructions to be serially shifted into the device. The instruction selects the particular test to be performed, the test data register to be accessed, or both. The instruction register is four (4) bits wide. The most significant bit is connected to TDI and the least significant bit is connected to TDO. There are no parity bits associated with the Instruction register. Upon entering the Capture-IR TAP controller state, the Instruction register is loaded with the default instruction "0001", SAMPLE/PRELOAD. Instructions are shifted into the instruction register on the rising edge of TCK while the TAP controller is in the Shift-IR state.

9.2.3.1 82495XP Boundary Scan Instruction Set

The 82495XP cache controller supports all three mandatory boundary scan instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST) along with one optional instruction (IDCODE). On the B-Stepping of the 82495XP two additional optional instructions will be implemented (RUNBIST and TRISTATE). Table 9.3 lists the 82495XP boundary scan instruction codes. The instructions listed as PRIVATE cause TDO to become enabled in the Shift-DR state and cause "0" to be shifted out of TDO on the rising edge of TCK. Execution of the PRIVATE instructions will not cause hazardous operation of the 82495XP. Note that system tests should not execute instruction codes labeled "RESERVED". These instructions can put the component in an undeterminant state which can only be cleared by power on reset.

Table 9-2. 82495XP Boundary Scan Instruction Codes

Instruction Code	Instruction Name
0000	EXTEST
0001	SAMPLE
0010	IDCODE
0011	*RESERVED*
0100	*RESERVED*
0101	*RESERVED*
0110	*RESERVED*
0111	*RUNBIST
1000	*TRISTATE
1001	*RESERVED*
1010	PRIVATE
1011	PRIVATE
1100	PRIVATE
1101	PRIVATE
1110	PRIVATE
1111	BYPASS

* RUNBIST and TRISTATE are boundary scan instructions that will be implemented in the B-stepping of the 82495XP. They are not available on the A-stepping.

EXTEST The instruction code is "0000". The EXTEST instruction allows testing of circuitry external to the component package, typically board interconnects. It does so by driving the values loaded into the 82495XP boundary scan register out on the output pins corresponding to each boundary scan cell and capturing the values on 82495XP input pins to be loaded into their corresponding boundary scan register locations. I/O pins are selected as input or output, depending on the value loaded into their control setting locations in the boundary scan register. Values shifted into input latches in the boundary scan register are never used by the internal logic of the 82495XP. Note: after using the EXTEST instruction, the 82495XP must be reset before normal (non-boundary scan) use.

SAMPLE/ PRELOAD The instruction code is "0001". The SAMPLE/PRELOAD has two functions that it performs. When the TAP controller is in the Capture-DR state, the SAMPLE/PRELOAD instruction allows a "snap-shot" of the normal operation of the component without interfering with that normal operation. The instruction causes boundary scan register cells associated with outputs to sample the value being driven by the 82495XP. It causes the cells associated with inputs to sample the value being driven into the 82495XP. On both outputs and inputs the sampling occurs on the rising edge of TCK. When the TAP controller is in the Update-DR state, the SAMPLE/PRELOAD instruction preloads data to the device pins to be driven to the board by executing the EXTEST instruction. Data is preloaded to the pins from the boundary scan register on the falling edge of TCK.

IDCODE The instruction code is "0010". The IDCODE instruction selects the device identification register to be connected to TDI and TDO, allowing the devices identification code to be shifted out of the device on TDO. Note that the device identification register is not altered by data being shifted in on TDI.

BYPASS The instruction code is "1111". The BYPASS instruction selects the bypass register to be connected to TDI and TDO, effectively bypassing the test logic on the 82495XP by reducing the shift length of the device to one bit. Note that an open circuit fault in the board level test data path will cause the bypass register to be selected following an instruction scan cycle due to the pull-up resistor on the TDI input. This has been done to prevent any unwanted interference with the proper operation of the system logic.

RUNBIST The instruction code is "0111". The RUNBIST instruction selects the one (1) bit runbist register, loads a value of "0" into the runbist register, and connects it to TDO. It also initiates the built-in self test (BIST) feature of the 82495XP, which is able to detect approximately 90% of the stuck-at faults on the 82495XP. The 82495XP ac/dc specifications for VCC and CLK must be met and reset must have been asserted at least once prior to executing the RUNBIST boundary scan instruction. After loading the RUNBIST instruction code in the instruction register, the TAP controller must be placed in the Run-Test/Idle state. BIST begins on the first rising edge of TCK after entering the Run-Test/Idle state. The TAP controller must remain in the Run-Test/Idle state until BIST is completed. It requires 100K clock (CLK) cycles to complete BIST and report the result to the runbist register. After completing the 100K clock (CLK) cycles, the value in the runbist register should be shifted out on TDO during the Shift-DR state. A value of "1" being shifted out on TDO indicates BIST successfully completed. A value of "0" indicates a failure occurred. After executing the RUNBIST instruction, the 82495XP must be reset prior to normal operation. NOTE: This instruction is not available on the A-stepping of the 82495XP. It will be implemented in the B-stepping.

TRISTATE The instruction code is "1000". The TRISTATE instruction initiates the tristate output test mode. After loading the TRISTATE boundary scan instruction into the instruction register, the TAP controller must be placed in the Run-Test/Idle state. To terminate the tristate output test mode, the 82495XP must be reset. NOTE: This instruction is not available on the A-stepping of the 82495XP. It will be implemented in the B-stepping.

9.2.3.2 82490XP Boundary Scan Instruction Set

The 82490XP cache controller supports all three mandatory boundary scan instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST) along with one optional instruction (IDCODE). Table 9.4 lists the 82490XP boundary scan instruction codes. The instructions listed as PRIVATE cause TDO to become enabled in the Shift-DR state and cause "0" to be shifted out of TDO on the rising edge of TCK. Execution of the PRIVATE instructions will not cause hazardous operation of the 82490XP. Note that system tests should not execute instruction codes labeled "INTEL RESERVED". These instructions can put the component in an undeterminant state which can only be cleared by power on reset.

Table 9-3. 82490XP Boundary Scan Instruction Codes

Instruction Code	Instruction Name
0000	EXTEST
0001	SAMPLE
0010	IDCODE
0011	INTEL RESERVED
0100	INTEL RESERVED
0101	INTEL RESERVED
0110	INTEL RESERVED
0111	INTEL RESERVED
1000	INTEL RESERVED
1001	INTERL RESERVED
1010	PRIVATE
1011	PRIVATE
1100	PRIVATE
1101	PRIVATE
1110	PRIVATE
1111	BYPASS

EXTEST The instruction code is "0000". The EXTEST instruction allows testing of circuitry external to the component package, typically board interconnects. It does so by driving the values loaded into the 82490XP boundary scan register out on the output pins corresponding to each boundary scan cell and capturing the values on 82490XP input pins to be loaded into their corresponding boundary scan register locations. I/O pins are selected as input or output, depending on the value loaded into their control setting locations in the boundary scan register. Values shifted into input latches in the boundary scan register are never used by the internal logic of the 82490XP. Note: after using the EXTEST instruction, the 82490XP must be reset before normal (non-boundary scan) use.

SAMPLE/ PRELOAD The instruction code is "0001". The SAMPLE/PRELOAD has two functions that it performs. When the TAP controller is in the Capture-DR state, the SAMPLE/PRELOAD instruction allows a "snap-shot" of the normal operation of the component without interfering with that normal operation. The instruction causes boundary scan register cells associated with outputs to sample the value being driven by the 82490XP. It causes the cells associated with inputs to sample the value being driven into the 82490XP. On both outputs and inputs the sampling occurs on the rising edge of TCK. When the TAP controller is in the Update-DR state, the SAMPLE/ PRELOAD instruction preloads data to the device pins to be driven to the board by executing the EXTEST instruction. Data is preloaded to the pins from the boundary scan register on the falling edge of TCK.

IDCODE The instruction code is "0010". The IDCODE instruction selects the device identification register to be connected to TDI and TDO, allowing the devices identification code to be shifted out of the device on TDO. Note that the device identification register is not altered by data being shifted in on TDI.

BYPASS The instruction code is "1111". The BYPASS instruction selects the bypass register to be connected to TDI and TDO, effectively bypassing the test logic on the 82490XP by reducing the shift length of the device to one bit. Note that an open circuit fault in the board level test data path will cause the bypass register to be selected following an instruction scan cycle due to the pull-up resistor on the TDI input. This has been done to prevent any unwanted interference with the proper operation of the system logic.

9.2.4 TEST ACCESS PORT (TAP) CONTROLLER

The TAP controller is a synchronous, finite state machine. It controls the sequence of operations of the test logic. The TAP controller changes state only in response to the following events:

1. A rising edge of TCK
2. Power-up.

The value of the test mode state (TMS) input signal at a rising edge of TCK controls the sequence of the state changes. The state diagram for the TAP contoller is shown in figure 9.3. Test designers must consider the operation of the state machine in order to design the correct sequence of values to drive on TMS.

9.2.4.1 Test-Logic-Reset State

In this state, the test logic is disabled so that normal operation of the device can continue unhindered. This is achieved by initializing the instruction register such taht the IDCODE instruction is loaded. No matter what the original state of the controller, the controller enters Test-Logic-Reset state when the TMS input is held high (1) for at least five rising edges of TCK. The controller remains in this state while TMS is high. The TAP controller is also forced to enter this state at power-up.

9.2.4.2 Run-Test/Idle State

A controller state between scan operations. Once in this state, the controller remains in this state as long as TMS is held low. In devices supporting the RUNBIST instruction, the BIST is performed during this state and the result is reported in the runbist register. For instructions not causing functions to execute during this state, no activity occurs in the test logic. The instruction register and all test data registers retain their previous state. When TMS is high and a rising edge is applied to TCK, the controller moves to the Select-DR state.

9.2.4.3 Select-DR-Scan State

This is a temporary controller state. The test data register selected by the current instruction retains its previous state. If TMS is held low and a rising edge is applied to TCK when in this state, the controller moves into the Capture-DR state, and a scan sequence for the selected test data register is initiated. If TMS is held high and a rising edge is applied to TCK, the controller moves to the Select-IR-Scan state.

The instruction does not change in this state.

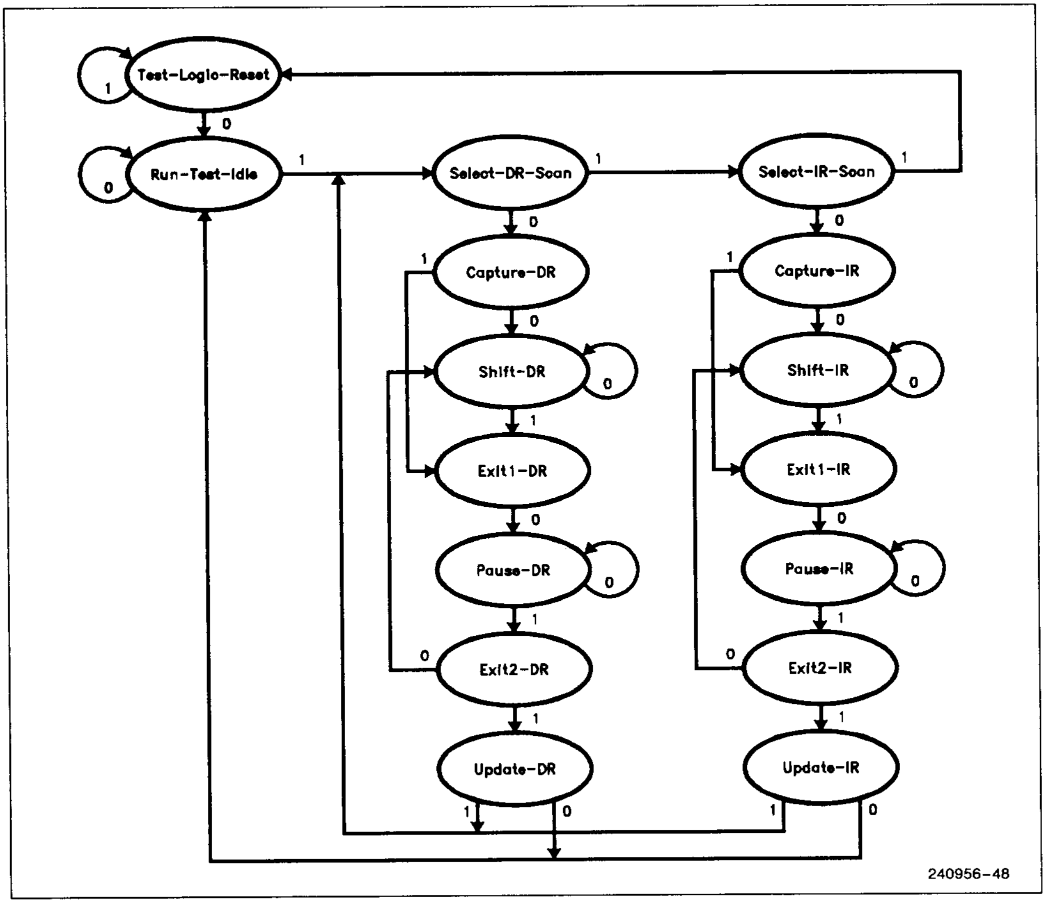


Figure 9-3. Tap Controller State Diagram

9.2.4.4 Capture-DR State

In this state, the boundary scan register captures input pin data if the current instruction is EXTEST or SAMPLE/PRELOAD. The other test data registers, which do not have parallel input, are not changed.

The instruction does not change in this state.

When the TAP controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-DR state if TMS is high or the Shift-DR state if TMS is low.

9.2.4.5 Shift-DR State

In this controller state, the test data register connected between TDI and TDO as a result of the current instruction, shifts data one stage toward its serial output on each rising edge of TCK.

The instruction does not change in this state.

When the TAP controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-DR state if TMS is high or remains in the Shift-DR state if TMS is low.

9.2.4.6 Exit1-DR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-DR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Pause-DR state.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

9.2.4.7 Pause-DR State

The pause state allows the test controller to temporarily halt the shifting of data through the test data register in the serial path between TDI and TDO. An example of using this state could be to allow a tester to reload its pin memory from disk during application of a long test sequence.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

The controller remains in this state as long as TMS is low. Whne TMS goes high and a rising edge is applied to TCK, the controller moves to the Exit2-DR state.

9.2.4.8 Exit2-DR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-DR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Shift-DR state.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

9.2.4.9 Update-DR State

The boundary scan register is provided with a latched parallel output to prevent changes at the parallel output while data is shifted in response to the EXTEST and SAMPLE/PRELOAD instructions. When the TAP controller is in this state and the boundary scan register is selected, data is latched onto the parallel output of this register from the shift-register path on the falling edge of TCK. The data held at the latched parallel output does not change other than in this state.

All shift-register stages in test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

9.2.4.10 Select-IR-Scan State

This is a temporary controller state. The test data register selected by the current instruction retains its previous state. If TMS is held low and a rising edge is applied to TCK when in this state, the controller moves into the Capture-IR state, and a scan sequence for the instruction register is initiated. If TMS is held high and a rising edge is applied to TCK, the controller moves to the Test-Logic-Reset state.

The instruction does not change in this state.

9.2.4.11 Capture-IR State

In this controller state the shift register contained in the instruction register loads the fixed value "0001" on the rising edge of TCK.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

When the controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-IR state if TMS is held high, or the Shift-IR state if TMS is held low.

9.2.4.12 Shift-IR State

In this state the shift register contained in the instruction register is connected between TDI and TDO and shifts data one stage towards its serial output on each rising edge of TCK.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

When the controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-IR state if TMS is held high, or remains in the Shift-IR state if TMS is held low.

9.2.4.13 Exit1-IR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-IR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Pause-IR state.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

9.2.4.14 Pause-IR State

The pause state allows the test controller to temporarily halt the shifting of data through the instruction register.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

The controller remains in this state as long as TMS is low. When TMS goes high and a rising edge is applied to TCK, the controller moves to the Exit2-IR state.

9.2.4.15 Exit2-IR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-IR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Shift-IR state.

The test data register selected by the current instruciton retains its previous value during this state. The instruction does not change in this state.

9.2.4.16 Update-IR State

The instruction shifted into the instruction register is latched onto the parallel output from the shift-register path on the falling edge of TCK. Once the new instruction has been latched, it becomes the current instruction.

Test data registers selected by the current instruction retain the previous value.

9.2.5 BOUNDARY SCAN REGISTER CELL

The boundary scan register for each component contains a cell for each pin, as well as cells for control of I/O and tristate pins.

9.2.5.1 82495XP Boundary Scan Register Cell

The following is the bit order of the 82495XP boundary scan register: (from left to right and top to bottom)

TDI → MKEN# KWEND# SWEND# BGT# CNA# BRDY# RESERVED CRDY# MWBWT# DRCTM# MRO# CWAY# FPFLD# SNPCYC# SNPBSY# MHITM# MTHIT# CAHOLD FSIOUT# PALLC# SNPADS# CADS# CDTS# CWR# CDC# CMIO# RDYSRC MCACHE# KLOCK# SMLN# NENE# CFA3 CFA2 TAG11 TAG10 TAG9 TAG8 TAG7 TAG6 TAG5 TAG4 TAG3 TAG2 TAG1 TAG0 SET10 SET9 SET8 SET7 CLK SET6 SET5 SET4 SET3 SET2 SET1 SET0 CFA6 CFA5 CFA4 CFA1 CFA0 ADS# LEN BLAST# BRDYC1# BRDYC2# CACHE# LOCK# BLE# BOFF# KEN# AHOLD WR# MIO# DC# PWT PCD HITM# PCYC EADS# NA# INV WBWT# WAY WRARR# MCYC# BUS# MAWEA# WBWE# WBA WBTYP MCFA0 MCFA1 MCFA4 MCFA5 MCFA6 MSET0 MSET1 MSET2 MSET3 MSET4 MSET5 MSET6 MSET7 MSET8 MSET9 MSET10 MTAG0 MTAG1 MTAG2 MTAG3 MTAG4 MTAG5 MTAG6 MTAG7 MTAG8 MTAG9 MTAG10 MTAG11 MCFA2 MCFA3 RESET MAOE# MBAOE# SNPCLK SNPSTB# EWBE# MPIC# SNPINV FLUSH# SNYC# SNPNCA MBALE MALE MACTL OCTL CFA4CTL CFA5CTL CACTL FPFLDCTL WBWTCTL NACTL → TDO

"RESERVED" signals correspond to no connect "NC" signals on the 82495XP.

EWBE# and MPIC# will be implemented in the 82495XP B-stepping, omit from boundary scan register for A-stepping 82495XPs.

All the *CTL cells are control cells that are used to select the direction of bidirectional pins or tristate output pins. If "1" is loaded into the control cell(*CTL), the associated pin(s) are tristated or selected as input. The following lists the control cells and their corresponding pins.

1. MACTL controls the MSET0–10, MTAG0–11, and MCFA0–6 pins.
2. OCTL controls the WAY, WRARR#, MCYC#, MAWEA#, BUS#, WBWE#, WBA, WBTYP, INV, EADS#, AHOLD, KEN#, BOFF#, BLE#, BRDYC2#, BRDYC1#, BLAST#, NENE#, SMLN#, KLOCK#, MCACHE#, RDYSRC, CMIO#, CDC#, CWR#, CDTS#, CADS#, SNPADS#, PALLC#, FSIOUT#, CAHOLD, MTHIT#, MHITM#, SNPBSY#, SNPCYC#, CWAY, EWBE#, and MPIC# output pins.
3. CFA4CTL controls the CFA4 pin.
4. CFA5CTL controls the CFA5 pin.
5. CACTL controls the SET0–10, TAG0–11, CFA0–3, and CFA6 pins.
6. FPFLDCTL controls the FPFLD# pin.
7. WBWTCTL controls the WB/WT# pin.
8. NACTL controls the NA# pin.

9.2.5.2 82490XP Boundary Scan Register Cell

The following is the bit order of the 82490XP boundary scan register: (from left to right and top to bottom)

TDI → CDCTL WR# BLAST# BRDYC# BRDY# HITM# ADS# BE# A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 MDATA7 MDATA3 MDATA6 MDATA2 MDATA5 MDATA1 MDATA4 MDATA0 MDCTL MDOE# MZBT# MBRDY# MOEC# MFRZ# MSEL# MCLK MOCLK RESET PAR# RESERVED BOFF# WBTYP WBA WBWE# BUS# MAWEA# MCYC# CRDY# WRARR# WAY CDATA4 CDATA0 CDATA2 CDATA5 CDATA6 CDATA1 CDATA3 CDATA7 → TDO

"RESERVED" signals correspond to no connect "NC" signals on the 82490XP.

All the *CTL cells are control cells that are used to select the direction of bidirectional pins or tristate output pins. If "1" is loaded into the control cell(*CTL), the associated pin(s) are tristated or selected as input. The following lists the control cells and their corresponding pins.

1. CDCTL controls the CDATA0–7 pins.
2. MDCTL controls the MDATA0–7 pins.

9.2.6 TAP CONTROLLER INITIALIZATION

The TAP controller is automatically intialized when a device is powered up. In addition, the TAP controller can be initialized by applying a high signal level on the TMS input for five TCK periods.

9.2.7 BOUNDARY SCAN SIGNAL DESCRIPTION AND TIMINGS

The functionality of TDI, TMS, TDO, and TCK are described in Chapter 7. The A.C. timing specifications for the boundary scan signals are located in Chapter 10.

9.3 Tri-State Output Test Mode

The 82495XP has the ability to tri-state all of its outputs and bidirectional pins and to disable all pull-ups and pull-downs. During tri-state output test mode all pins floated during bus hold as well as those which are never floated during normal operation are tri-stated. When the 82495XP is in tri-state output test mode, external testing can be used to test board interconnections.

On the 82495XP, tri-state output test mode is invoked by driving HIGHZ#(MBALE) and SLFTST#-(CRDY#) active to the 82495XP at least 10 clocks prior to the deassertion of RESET. Note that HIGHZ# has priority over SLFTST#. When both HIGHZ# and SLFTST# are driven active the 82495XP will invoke the tri-state output mode and not invoke BIST.

Once tri-state output test mode is invoked, the 82495XP remains in it until the next RESET.

9.4 82490XP Cache SRAM Testing

The 82490XP cache SRAM can be tested using standard cache memory testing techniques. Code must be written to:

1. Flush and reset the 82495XP/82490XP/CPU cache
2. Write 1's to every bit of a block of memory equal to the cache size
3. Read the block of memory to fill the cache, tagging the data as read-only using the MRO# signal
4. Write 0's to every bit in the block of memory
5. Read the block, the cache hits should be all 1's
6. Repeat the process, exchanging 0 for 1 and 1 for 0

In this example, the code to test the cache must be non-cacheable to the 82495XP. Also, the CPU cache must be on so that the 82495XP will perform line-fills.

10.0 AC/DC SPECIFICATIONS

10.1 Background

The 82495XP has four main interfaces: CPU Bus, memory bus controller, memory bus, and 82490XP. The memory bus controller is typically implemented with PLD devices. The MBC interface signal timings are, therefore, generated based on available, off-the-shelf PLD specs. The memory bus interface was specified to suit a generic memory interface which works up to CPU frequency.

10.2 D.C. Specifications

Table 10-1. D.C. Specifications

Vcc = 5V ±5%, Tcase = 0 to +85°C

Symbol	Parameter	Min	Max	Unit	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	TTL Level
V_{IH}	Input High Voltage 2.0	2.0	Vcc+0.3	V	TTL Level
V_{OL}	Output Low Voltage		0.45	V	TTL Level (1)
V_{OH}	Output High Voltage	2.4		V	TTL Level (2)
I_{CC}	Power Supply Current		600 300	mA	82495XP @ 50 MHz, (3) 82490XP @ 50 MHz, (3)
Power	Power Dissipation		3.00 1.50	W	82495XP @ 50 MHz, (3) 82490XP @ 50 MHz, (3)
I_{LI}	Input Leakage Current		±15	uA	$0 < V_{IN} < V_{CC}$
I_{LO}	Output Leakage Current		±15	uA	$0 \leq V_{OUT} \leq V_{CC}$ Tristate
I_{IL}	Input Leakage Current		200	uA	$V_{IN} = 0.45V$, (4)

C_{IN}	Input Capacitance		14
			5
				pF	for 82495XP
					for 82490XP

C_O	Output Capacitance		18
			15
				pF	for 82495XP
					for 82490XP

$C_{I/O}$	I/O Capacitance		18
			15
				pF	for 82495XP
					for 82490XP

C_{CLK}	CLK Input Capacitance		11
			5
				pF	for 82495XP
					for 82490XP

C_{TIN}	Test Input Capacitance		15
			10
				pF	for 82495XP
					for 82490XP

C_{TOUT}	Test Output Capacitance		15
			10
				pF	for 82495XP
					for 82490XP

C_{TCK}	Test Clock Capacitance		15
			10
				pF	for 82495XP
					for 82490XP

NOTES:

(1) Parameter measured at 4mA Iload.
For MCFA6-FCFA0, MSET10-MSET0, and MTAG11-MTAG0, this parameter is measured at 12 mA Iload.

(2) Parameter measured at 1mA Iload.
For MCFA6-MCFA0, MSET10-MSET0, and MTAG11-MTAG0, this parameter is measured at 2 mA Iload.

(3) Represents maximum power consumption given a typical cache cycle mix.

(4) This parameter is for input with pullup.

10.3 A.C. Specifications

NOTE:
Please contact your local Intel Sales Office for the latest timing information.

All TTL timing specs are measured at 1.5V for both "0" and "1" logic level.

Table 10-2. Clock, Reset, and Configuration

Vcc = 5V ± 5%, Tcase = 0 to +85 °C
C_L = 0 pF unless otherwise specified.
All Inputs and Outputs are TTL Level.

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t0	CLK, MCLK, MOCLK Frequency	25	50	MHz		1x clock
t1	CLK, MCLK, MOCLK Stability		0.1	%		
t2	CLK, MCLK, MOCLK Period	20	40	ns	10-1	
t3	CLK, MCLK, MOCLK High Time	7		ns	10-1	(1)
t4	CLK, MCLK, MOCLK Low Time	7		ns	10-1	(1)
t5	CLK, MCLK, MOCLK Rise Time		2	ns		(1)
t6	CLK, MCLK, MOCLK Fall Time		2	ns		(1)
t7	RESET Setup Time	6		ns	10-4	
t8	RESET Hold Time	2		ns	10-4	

t9	RESET Duration	8xt2
		15xt2
				ns	10-4	for 82495XP, (2)
						for 82490XP

t10	All Configurations CFG3-CFG0, SNPMD, MEMLDRV, C490LDRV, HIGHZ#, SLFTST# Setup Time	10xt2		ns	10-4	(3), (4)
t11	All Configurations CFG3-CFG0, SNPMD, MEMLDRV, C490LDRV, HIGHZ#, SLFTST# Hold Time	0		ns	10-4	(3), (5)
t12	FLUSH#, SYNC# Setup Time	8		ns	10-3	for 82495XP, (6)
t13	FLUSH#, SYNC# Hold Time	1		ns	10-3	for 82495XP, (7)
t14	FLUSH#, SYNC# Duration	2xt2		ns		(8)
t15	MOCLK falling edge to MCLK rising edge	2		ns		

NOTE:
(1) Rise/Fall, High/Low times measured between 0.8V and 2.0V.
(2) Power up reset duration should be 1 ms after Vcc and CLK are stable. If configuration inputs with pullups are left floated, 10 us RESET duration is required.
(3) Timing is referenced to reset falling edge.
(4) 8ns setup time is required to guarantee recognition on next clock.
(5) 1ns hold time is required to guarantee recognition on next clock.
(6) To guarantee recognition on next clock.
(7) Synchronous mode only.
(8) Asynchronous mode only. To guarantee recognition.

Table 10-3. Memory Bus Controller 82495XP/82490XP Interface

Vcc = 5V ± 5%, Tcase = 0 to +85 °C C_L = 0 pF unless otherwise specified. All Inputs and Outputs are TTL Level.						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
t30	BRDY#, CRDY#, KWEND#, SWEND#, BGT#, CNA#, Setup Time	8		ns	10-3	82495XP Only
t30a	BRDY#, CRDY# Setup Time	7		ns	10-3	82490XP Only
t31	BRDY#, CRDY#, KWEND#, SWEND#, BGT#, CNA#, Hold Time	1		ns	10-3	82495XP Only
t32	CW/R#, CD/C#, CMI/O#, RDYSRC, MCACHE#, KLOCK#, BLE#, PALLC#, CAHOLD, CWAY, FSIOUT#, CADS#, CDTS#, SNPADS# FPFLD# Valid Delay	2	10	ns	10-2	
t33	NENE#, SMLN# Valid Delay	2	14	ns	10-2	
t34	MDATA Setup to CLK (clock before BRDY# active)	6		ns	10-3	(1)
t35	MDATA Valid Delay from CLK (CLK from CDTS# valid, MDOE# active)	2	13	ns	10-2	
t36	MDATA Valid Delay from MDOE# active		8	ns	10-2	
t37	MDATA Float Delay from MDOE# inactive	0	14	ns		

NOTE:
(1) Even if MBRDY# or MISTB is not used for this cycle, the data must still be held according to t43.

Table 10-4. 82495XP Memory Interface

Vcc = 5V ± 5%, Tcase = 0 to +85 °C C_L = 0 pF unless otherwise specified. All Inputs and Outputs are TTL Level.						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
t50	SNPCLK Frequency	8	50	MHz		1x clock (10)
t51	SNPCLK Period	20	125	ns	10-1	(11)
t52	SNPCLK High Time	7		ns	10-1	
t53	SNPCLK Low Time	7		ns	10-1	
t54	SNPCLK Rise Time		2	ns		(1)
t55	SNPCLK Fall Time		2	ns		(1)
t56	MCFA6-MCFA0, MSET10-MSET0, MTAG11-MTAG0 Valid Delay	2	11	ns	10-5	(2), (3)
t57	MCFA6-MCFA0, MSET10-MSET0, MTAG11-MTAG0 Float Delay	2	15	ns	10-5	(4)
t58	MCFA6-MCFA0, MSET10-MSET0, MTAG11-MTAG0 Valid Delay	2	13	ns	10-5	(5)

2

Table 10-4. 82495XP Memory Interface (Continued)

Vcc = 5V ± 5%, Tcase = 0 to +85°C
C_L = 0 pF unless otherwise specified.
All Inputs and Outputs are TTL Level.

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t60	MCFA6–MCFA0, MSET10–MSET0, MTAG11–MTAG0 Valid Delay	2	13	ns	10-2	(6)
t62a	MCFA6–MCFA0, MSET10–MSET0, MTAG11–MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE#, SNPSTB# Setup Time	8		ns	10-3	(7a)
t62b	MCFA6–MCFA0, MSET10–MSET0, MTAG11-MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE# Setup Time	1		ns	10-3	(7b)
t62c	MCFA6–MCFA0, MSET10–MSET0, MTAG11-MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE#, SNPSTB# Setup Time	8		ns	10-3	(7c)
t63a	MCFA6–MCFA0, MSET10–MSET0, MTAG11–MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE#, SNPSTB# Hold Time	1		ns	10-3	(7a)
t63b	MCFA6–MCFA0, MSET10–MSET0, MTAG11–MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE# Hold Time	8		ns	10-3	(7b)
t63c	MCFA6–MCFA0, MSET10-MSET0, MTAG11–MTAG0, SNPINV, SNPNCA, MAOE#, MBAOE#, SNPSTB# Hold Time	1		ns	10-3	(7c)
t64	SNPSTB# Setup Time	8		ns	10-3	(8)
t65	SNPSTB# Hold Time	1		ns	10-3	(8)
t66	SNPSTB# Active/Inactive Time	8		ns	10-3	(9)
t67	MRO#, MKEN#, DRCTM#, MWB/WT# Setup Time	8		ns	10-3	
t68	MRO#, MKEN#, DRCTM#, MWB/WT# Hold Time	1		ns	10-3	
t69	MTHIT#, MHITM#, SNPBSY#, Valid Delay	2	12	ns	10-2	
t69a	SNPCYC# Valid Delay	2	10	ns	10-2	

NOTES:
(1) Rise/fall times measured between 0.45V and 2.4V
(2) See capacitive derating curves for additional loading delay.
(3) Valid delay from MAOE#, MBAOE# going active (low)
(4) Float delay from MAOE#, MBAOE# going inactive (high)
(5) Valid delay from MALE or MBALE if both MAOE#, MBAOE# are active
(6) Valid delay from CLK only if MALE or MBALE, MAOE# and MBAOE# are active
(7) a. In clocked mode referenced to SNPCLK rising edge
b. In strobed mode referenced to SNPSTB# falling edge
c. In synchronous mode, refer to CLK
(8) Asynchronous clocked mode only. Timings referenced to SNPCLK
(9) Asynchronous signal. Time to guarantee recognition on next clock
(10) SNPCLK is only used for the clocked memory bus mode. SNPCLK Min frequency not tested.
(11) t51 > t2

Table 10-5. 82490XP Clocked Mode

Vcc = 5V ± 5%, Tcase = 0 to +85°C
C_L = 0 pF unless otherwise specified.
All Inputs and Outputs are TTL Level.

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t38	MBRDY#, MSEL#, MEOC# Setup to MCLK	5		ns	10-3	
t39	MBRDY#, MSEL#, MEOC# Hold from MCLK	2		ns	10-3	
t40	MZBT#, MFRZ# Setup to MCLK	5		ns	10-3	
t41	MZBT#, MFRZ# Hold from MCLK	2		ns	10-3	
t42	MDATA Setup to MCLK	5		ns	10-3	
t43	MDATA Hold from MCLK	2		ns	10-3	
t44	MDATA Valid Delay from MCLK*MBRDY#	2	14	ns	10-2	
t45	MDATA Valid Delay from MCLK*MEOC#, MCLK*MSEL#	2	20	ns	10-2	
t46	MDATA Valid Delay from MOCLK	2	10	ns	10-2	

Table 10-6. 82490XP Strobed Mode

Vcc = 5V ± 5%, Tcase = 0 to +85°C
C_L = 0 pF unless otherwise specified.
All Inputs and Outputs are TTL Level.

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t85	MISTB, MOSTB High Time	12		ns	10-6	
t86	MISTB, MOSTB Low time	12		ns	10-6	
t87	MEOC# High time	8		ns	10-6	
t88	MEOC# Low time	8		ns	10-6	
t89	MxSTB, MEOC# Rise time		2	ns		(1)
t90	MxSTB, MEOC# Fall time		2	ns		(1)
t91	MSEL# High time for restart	8		ns	10-6	
t92	MSEL# Setup before transition on MxSTB	5		ns	10-8	
t93	MSEL# Hold after transition on MxSTB	10		ns	10-8	
t92	MSEL# Hold after transition on MEOC#	2		ns	10-8	
t95	MxSTB transition to/from MEOC# falling transition	10		ns		
t96	MZBT# Setup to MSEL# or MEOC# falling edge	5		ns	10-7	
t97	MZBT# Hold from MSEL# or MEOC# falling edge	2		ns	10-7	
t98	MFRZ# Setup to MEOC# falling edge	5		ns	10-7	
t99	MFRZ# Hold from MEOC# falling edge	2		ns	10-7	
t100	MDATA Setup to MxSTB or MEOC# falling transition	5		ns	10-7	
t101	MDATA Hold from MxSTB or MEOC# falling transition	2		ns	10-7	
t102	MDATA Valid Delay from MxSTB transition	2	14	ns	10-9	
t103	MDATA Valid Delay from MEOC# falling transition or MSEL# deactivation	2	20	ns	10-9	

NOTE:
(1) Rise/Fall times are measured between 0.8V and 2.0V

Table 10-7. Test Mode

Vcc = 5V ± 5%, Tcase = 0 to +85 °C C_L = 0 pF unless otherwise specified. All Inputs and Outputs are TTL Level.						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
t120	TCK Frequency		25	MHz		1x clock
t121	TCK Period	40		ns		(2)
t122	TCK High Time	10		ns		@ 2.0V
t123	TCK Low Time	10		ns		@ 0.8V
t124	TCK Rise Time		4	ns		(1)
t125	TCK Fall Time		4	ns		(1)
t126	TDI, TMS Setup Time	8		ns	10-10	
t127	TDI, TMS Hold Time	7		ns	10-10	
t128	TDO Valid Delay	3	25	ns	10-10	
t129	TDO Float Delay					
t130	All Outputs Valid Delay	3	25	ns	10-10	(3)
t131	All Outputs Float Delay		36	ns	10-10	(3)

NOTES:

(1) Rise/Fall times are measured between 0.8V and 2.0V Rise/Fall times can be relaxed by 1ns per 10ns increase in TCK period

(2) TCK period ≥ CLK period

(3) Parameter measured from TCK

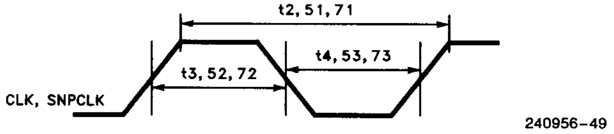
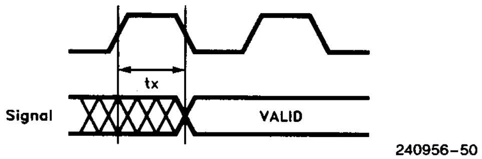


Figure 10-1. Clock Waveform



tx = t16, 32, 33, 35, 36, 44, 45, 60, 69

Figure 10-2. Valid Delay Timings

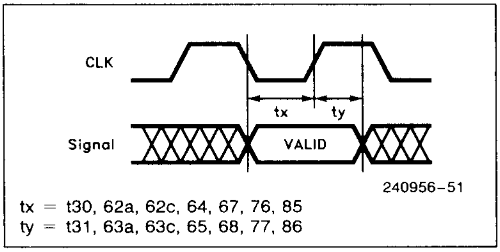

CLK
Signal
tx
ty
VALID
240956-51
tx = t30, 62a, 62c, 64, 67, 76, 85
ty = t31, 63a, 63c, 65, 68, 77, 86

Figure 10-3. Setup and Hold Timings

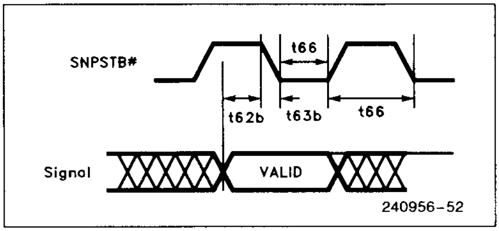


Figure 10-3a. Setup and Hold Timings in Strobed Snooping Mode

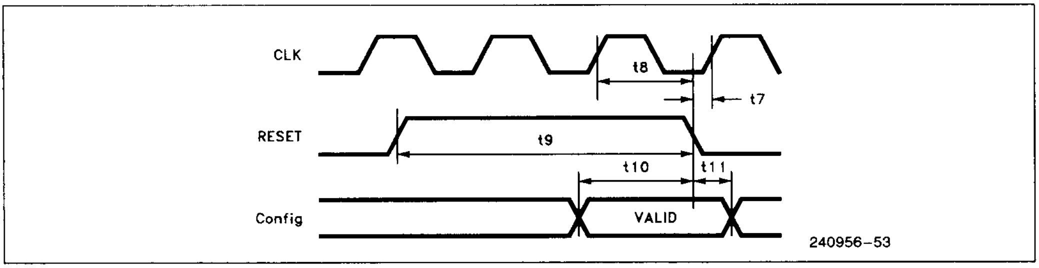


Figure 10-4. Reset and Configuration Timings

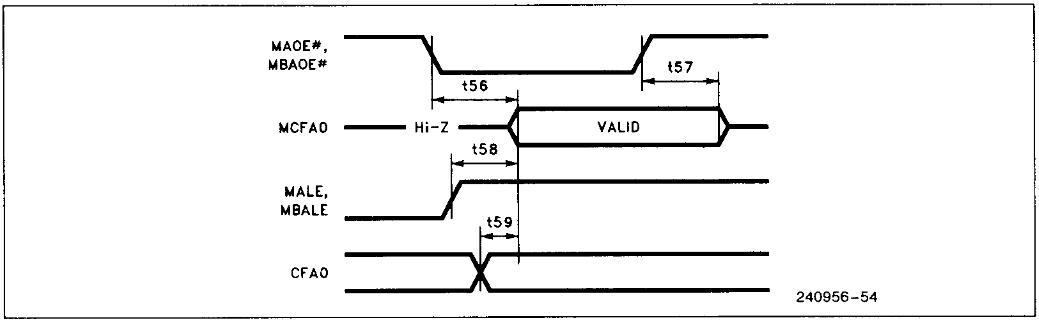


Figure 10-5. Memory Interface Signals

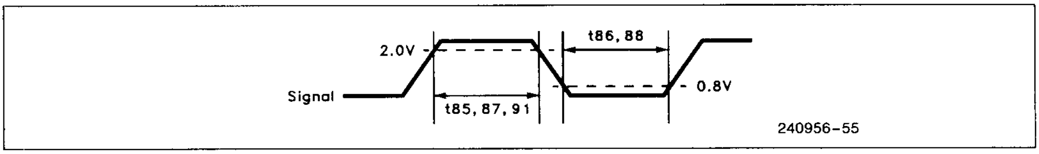


Figure 10-6. Active/Inactive Timing

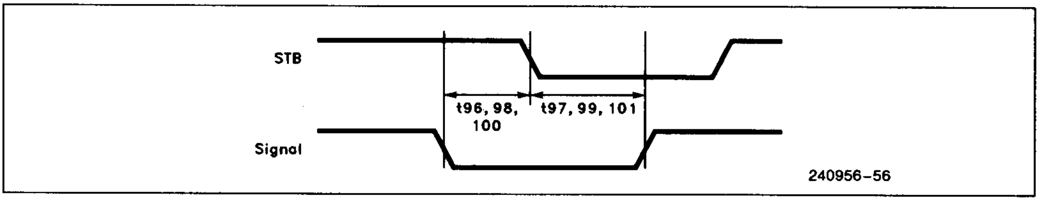


Figure 10-7. Setup and Hold Timing

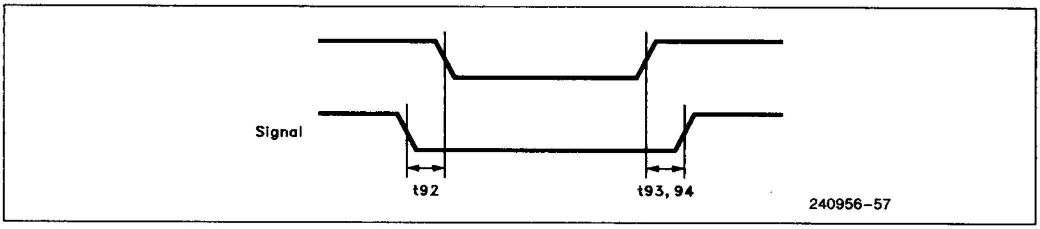


Figure 10-8. Setup and Hold Timing

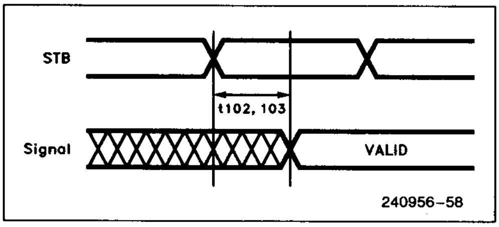


Figure 10-9. Valid Delay Timing

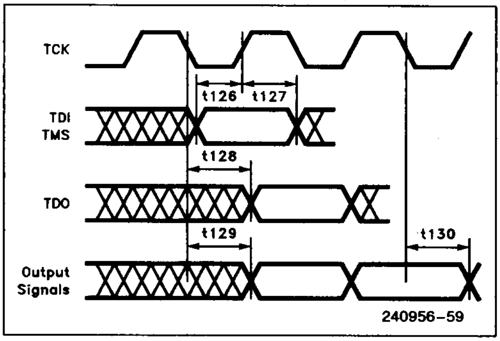


Figure 10-10. Test Timings

10.4 Optimized Interface Specifications

The optimized interface is the high performance interconnect between the i860 XP microprocessor, 82495XP cache controller, and 82490XP cache RAM. This interface is tuned for the known configuration options of the chip set and includes non-standard input and output buffers optimized for the defined electrical environment of each signal path. The specification of this interface is also non-standard. The output delay of a signal driver and the input setup time of the corresponding receiver are summed into a single value. The time remaining in a clock period must account for clock skew and signal trace delay. In designing the layout between these components, the designer must ensure that each signal in the optimized interface is carefully simulated using buffer models, and that signal timing achieves the specified limits.

Tables 10-8 through 10-11 define the clock skew and combined setup + valid delay spec for each path in the optimized interface.

NOTE:
Please contact your local Intel Sales office for the latest timing information.

Table 10-8. Signal Group: i860™ XP CPU to Cache **256 Kbyte Version**

Driver	Receiver	Combined Setup + Valid Delay	Maximum CLK Skew
CPU A3	495 CFA0	16.7	0.5
CPU A4–A5	495 CFA1, CFA6	16.7	0.5
CPU A6–A16	495 SET0–SET10	16.7	0.5
CPU A17-A21	495 TAG0–TAG4	17.9	0.5
CPU A22–A28	495 TAG5–TAG11	17.9	0.5
CPU A29–A31	495 CFA2–CFA4	17.9	0.5
495 CFA0*	CPU A3	31.9	0.5
495 CFA1, CFA6*	CPU A4, A5	31.9	0.5
495 CFA2–CFA4*	CPU A29–A31	37.4	0.5
495 SET0–SET10*	CPU A16–A16	31.9	0.5
495 TAG0–TAG4*	CPU A17–A21	37.4	0.5
495 TAG5–TAG11*	CPU A22–A28	37.4	0.5
CPU A3–A16	490 A2–A15	16.2	1.0
CPU ADS#	495 ADS#	16.8	0.5
CPU ADS#	490 ADS#	16.3	1.0
CPU HITM#	495 HITM#	16.8	0.5
CPU HITM#	490 HITM#	16.3	1.0
CPU W/R#	495 W/R#	16.8	0.5
CPU W/R#	490 W/R#	16.3	1.0

*Total time for these signals is 40 ns: they are driven one clock before EADS#.

Table 10-9. Signal Group: i860™ XP CPU to 82495XP 256 Kbyte Version

Driver	Receiver	Combined Setup + Valid Delay	Maximum CLK Skew
CPU CACHE#	495 CACHE#	17.9	0.5
CPU CTYP	495 CFA5	16.8	0.5
CPU D/C#	495 D/C#	17.9	0.5
CPU LEN	495 LEN	18.0	0.5
CPU LOCK#	495 LOCK#	18.0	0.5
CPU M/IO#	495 M/IO#	17.9	0.5
CPU PCD	495 PCD	17.9	0.5
CPU PCYC	495 PCYC	17.9	0.5
CPU PWT	495 PWT	17.9	0.5
495 AHOLD	CPU AHOLD	17.6	0.5
495 BOFF#	CPU BOFF#	17.6	0.5
495 BRDYC1#	CPU BRDY1# (RSRVD)	18.1	0.5
495 EADS#	CPU EADS	17.6	0.5
495 EWBE#	CPU EWBE	17.6	0.5
495 INV	CPU INV	17.6	0.5
495 KEN#	CPU KEN#	17.6	0.5
495 NA#	CPU NA#	17.6	0.5
495 WB/WT#	CPU WB/WT#	17.6	0.5

Table 10-10. Signal Group: i860™ XP CPU to 82490XP 256 Kbyte Version

Driver	Receiver	Combined Setup + Valid Delay	Maximum CLK Skew
CPU BE7#-BE0#	490 BE#	16.3	1.0
CPU BE7#-BE0#	490 CDATA7-4	16.3	1.0
CPU D63-D0	490 CDATA7-0	16.9	1.0
490 CDATA7-0	CPU D63-D0	16.8	1.0
CPU DP7-DP0	490 CDATA3-0	16.9	1.0
490 CDATA3-0	CPU DP7-DP0	16.8	1.0

Table 10-11. Signal Group: 82495XP to 82490XP — 256 Kbyte Version

Driver	Receiver	Combined Setup + Valid Delay	Maximum CLK Skew
495 BLAST#	490 BLAST#	17.2	1.0
495 BOFF#	490 BOFF#	17.5	1.0
495 BRDYC2#	490 BRDYC#	17.3	1.0
495 BUS#	490 BUS#	16.9	1.0
495 MAWEA#	490 MAWEA#	16.9	1.0
495 MCYC#	490 MCYC#	16.9	1.0
495 WAY	490 WAY	17.0	1.0
495 WBA	490 WBA	16.8	1.0
495 WBTYP	490 WBTYP	16.9	1.0
495 WBWE#	490 WBWE#	16.9	1.0
495 WRARR#	490 WRARR#	17.1	1.0

10.5 The First Order Electrical Buffer Model

The first order electrical buffer model provides an accurate and simple representation of the buffers used in the inputs and outputs of the 82495XP/82490XP. The model output consists of four components:

1. Linear voltage waveform (dV/dt)
2. Intrinsic buffer delay due to C_L (t_0)
3. Buffer output impedance (R_O)
3. Buffer output capacitance (C_O)

as shown in Figure 10-11.

A fitting algorithm has been used to arrive at values for dV/dt, C_O, and R_O such that R_O matches the actual buffer impedance and C_O, the intrinsic buffer output capacitance whether the output is on or off, remains constant across the operating range while minimizing the difference between the full buffer circuit and its simplified electrical model for a set of different loads (lumped capacitance, and short and long transmission lines). dV/dt is the slope of the voltage ramp, while t_0 is the intrinsic buffer delay associated with a given C_L. t_0 accounts for the intrinsic delay by offsetting the excitation of the model by the amount of the delay.

NOTE:
t_0 is zero for $C_L = 0$ and when the load is represented by a transmission line.

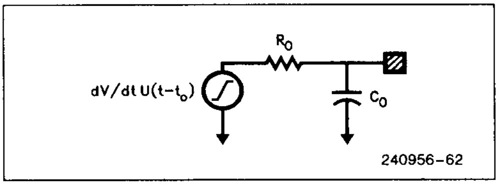


Figure 10-11. Output Model

The input model consists of one component, buffer capacitance (C_{IN}), as shown in Figure 10-12.

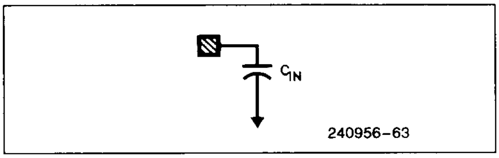


Figure 10-12. Input Model

10.5.1 FIRST ORDER ELECTRICAL MODEL PARAMETER VALUES

The parameters that make up the first order electrical model vary with the buffer design. In addition, these parameters also vary with the operating condition (i.e. temperature and V_{CC}) of the buffer and process. The typical process corner is being modeled. Three sizes of buffer are used on these components, labelled here as small, large, and extra large. The parameter values found in tables 10-12 through 10-14 list dV/dt, t_O, R_O, and C_O. These parameters are provided for both low-to-high and high-to-low transitions at the typical process corner for three operating conditions (V_{CC} = 5.5V and T_J = −10°C, V_{CC} = 5.0V and T_J = 80°C, and V_{CC} = 4.5V and T_J = 125°C.

10.5.2 PACKAGE PARAMETERS

In addition to buffer characteristics, package characteristics are also included to complete the model. Package inductance, capacitance and resistance values vary with design geometry and material properties of the package. Figure 10-13 shows a model of the package including these parameters and should be placed between the first order electrical buffer model as shown in Figure 10-14 and the board interconnects. Notice the package model only includes the package inductance (L_P) and capacitance (C_P). This is sufficient since the package resistance is so small it is negligible.

Tables 10-15 and 10-16 list the buffer model parameters for each pin of the 82495XP and 82490XP cache components. The tables give the package model parameters for each pin, followed by the input capacitance (input and I/O pins) and/or output buffer size (outputs and I/O). In those cases where the buffer used by a pin is an option selected at reset, the output buffer column lists the sizes available.

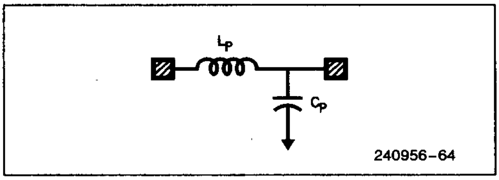


Figure 10-13. Package Model

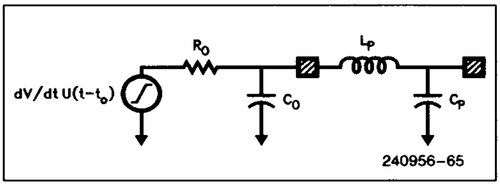


Figure 10-14a. Output Buffer and Package Model

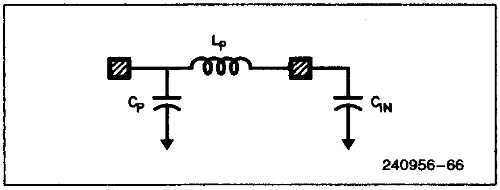


Figure 10-14b. Input Buffer and Package Model

10.5.3 BOARD INTERCONNECTS

The board interconnect can be considered as a lumped parameter (capacitive load) or as a transmission line. As a rule of thumb, an unterminated board interconnect may be considered as a capacitive load if the round trip time (time for signal to travel from one end of the interconnect to the other and back) is short compared to the transition time of the signal. At frequencies of 50 MHz and above most interconnects behave as transmission lines (Figure 10-15). For accurate results at high frequencies, these transmission line effects must be taken into account and modeled.

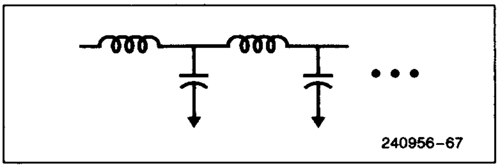


Figure 10-15. Transmission Line Model

Table 10-12. Small Buffer First Order Electrical Model Parameter Values

Transition	V_{CC} (V)	T_J (C)	R_O (Ω)	C_O (pF)	dV/dt	t_O (ns) C_L (pF) =					
						0	5	25	50	100	150
low-to-high	5.5	−10	28.0	4.3	5.5/1.2	0	0.0	0.1	0.3	0.7	1.1
low-to-high	5.5	80	36.4	4.3	5.5/1.4	0	0.0	0.1	0.8	0.8	1.2
low-to-high	5.5	125	40.4	4.3	5.5/1.5	0	0.0	0.1	0.4	0.8	1.2
low-to-high	5.0	−10	30.2	4.3	5.0/1.2	0	0.0	0.1	0.4	0.8	1.2
low-to-high	5.0	80	39.2	4.3	5.0/1.4	0	0.0	0.2	0.4	0.9	1.3
low-to-high	5.0	125	43.5	4.3	5.0/1.6	0	0.0	0.2	0.4	0.9	1.3
low-to-high	4.5	−10	33.0	4.3	4.5/1.2	0	0.0	0.2	0.5	1.0	1.4
low-to-high	4.5	80	42.8	4.3	4.5/1.6	0	0.0	0.2	0.5	1.0	1.5
low-to-high	4.5	125	47.4	4.3	4.5/1.6	0	0.0	0.3	0.6	1.1	1.6
high-to-low	5.5	−10	23.2	4.3	5.5/1.0	0	0.0	0.4	0.7	1.2	1.6
high-to-low	5.5	80	31.4	4.3	5.5/1.4	0	0.0	0.4	0.9	1.3	1.8
high-to-low	5.5	125	36.1	4.3	5.5/1.6	0	0.0	0.5	0.8	1.3	1.8
high-to-low	5.0	−10	24.0	4.3	5.0/1.1	0	0.0	0.5	0.9	1.2	1.7
high-to-low	5.0	80	32.8	4.3	5.0/1.4	0	0.0	0.5	0.9	1.5	1.9
high-to-low	5.0	125	37.8	4.3	5.0/1.7	0	0.0	0.5	0.9	1.4	1.8
high-to-low	4.5	−10	25.1	4.3	4.5/1.2	0	0.0	0.4	0.7	1.2	1.7
high-to-low	4.5	80	34.5	4.3	4.5/1.6	0	0.0	0.4	0.8	1.3	1.8
high-to-low	4.5	125	39.9	4.3	4.5/1.8	0	0.0	0.5	0.9	1.4	1.9

Table 10-13. Large Buffer First Order Electrical Model Parameter Values

Transition	V_{CC} (V)	T_J (C)	R_O (Ω)	C_O (pF)	dV/dt	t_o (ns) C_L (pF) =								
						0	5	25	50	100	150	200	250	300
low-to-high	5.5	−10	12.1	4.3	5.5/0.7	0	0.0	0.1	0.3	0.6	0.8	1.0	1.3	1.5
low-to-high	5.5	80	15.5	4.3	5.5/0.9	0	0.0	0.2	0.3	0.6	0.9	1.1	1.4	1.7
low-to-high	5.5	125	17.2	4.3	5.5/1.1	0	0.0	0.2	0.4	0.7	1.0	1.2	1.4	1.7
low-to-high	5.0	−10	13.0	4.3	5.0/0.9	0	0.0	0.1	0.3	0.6	0.9	1.1	1.4	1.7
low-to-high	5.0	80	16.7	4.3	5.0/1.0	0	0.0	0.2	0.4	0.8	1.1	1.4	1.7	2.0
low-to-high	5.0	125	18.5	4.3	5.0/1.2	0	0.0	0.2	0.4	0.8	1.1	1.4	1.7	2.0
low-to-high	4.5	−10	14.1	4.3	4.5/0.9	0	0.0	0.2	0.4	0.7	1.1	1.4	1.7	2.0
low-to-high	4.5	80	18.0	4.3	4.5/1.2	0	0.0	0.2	0.4	0.9	1.2	1.5	1.9	2.2
low-to-high	4.5	125	19.9	4.3	4.5/1.3	0	0.0	0.2	0.5	0.8	1.2	1.5	1.9	2.2
high-to-low	5.5	−10	10.6	4.3	5.5/0.7	0	0.0	0.3	0.6	0.9	1.2	1.5	1.8	2.0
high-to-low	5.5	80	13.9	4.3	5.5/1.0	0	0.0	0.4	0.7	1.2	1.5	1.9	2.2	2.5
high-to-low	5.5	125	15.8	4.3	5.5/1.1	0	0.0	0.4	0.8	1.3	1.7	2.0	2.4	2.8
high-to-low	5.0	−10	11.0	4.3	5.0/0.8	0	0.0	0.4	0.7	1.0	1.3	1.6	1.9	2.1
high-to-low	5.0	80	14.5	4.3	5.0/1.0	0	0.0	0.4	0.8	1.2	1.6	2.0	2.3	2.6
high-to-low	5.0	125	16.5	4.3	5.0/1.2	0	0.0	0.4	0.8	1.3	1.7	2.1	2.5	2.8
high-to-low	4.5	−10	11.3	4.3	4.5/0.9	0	0.0	0.4	0.7	1.1	1.4	1.7	2.0	2.4
high-to-low	4.5	80	15.2	4.3	4.5/1.2	0	0.0	0.4	0.8	1.3	1.6	2.0	2.3	2.7
high-to-low	4.5	125	17.4	4.3	4.5/1.3	0	0.0	0.4	0.8	1.3	1.7	2.1	2.5	2.8

Table 10-14. Extra Large Buffer First Order Electrical Model Parameter Values

Transition	V_{CC} (V)	T_J (C)	R_O (Ω)	C_O (pF)	dV/dt	t_o (ns) C_L (pF) =								
						0	5	25	50	100	150	200	250	300
low-to-high	5.5	−10	7.7	8.9	5.5/0.8	0	0.0	0.1	0.2	0.4	0.6	0.8	0.9	1.1
low-to-high	5.5	80	9.8	8.9	5.5/1.1	0	0.0	0.2	0.3	0.5	0.8	1.0	1.2	1.4
low-to-high	5.5	125	10.8	8.9	5.5/1.2	0	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
low-to-high	5.0	−10	8.2	8.9	5.0/0.9	0	0.0	0.1	0.2	0.5	0.7	0.9	1.1	1.3
low-to-high	5.0	80	10.5	8.9	5.0/1.1	0	0.0	0.2	0.3	0.7	0.9	1.2	1.4	1.6
low-to-high	5.0	125	11.5	8.9	5.0/1.2	0	0.0	0.2	0.4	0.8	1.0	1.3	1.5	1.7
low-to-high	4.5	−10	8.9	8.9	4.5/1.0	0	0.0	0.2	0.3	0.6	0.8	1.0	1.2	1.5
low-to-high	4.5	80	11.3	8.9	4.5/1.2	0	0.0	0.1	0.3	0.7	0.9	1.2	1.4	1.6
low-to-high	4.5	125	12.4	8.9	4.5/1.2	0	0.0	0.2	0.4	0.8	1.1	1.4	1.6	1.9
high-to-low	5.5	−10	8.5	8.9	5.5/0.8	0	0.0	0.2	0.4	0.7	0.9	1.0	1.2	1.4
high-to-low	5.5	80	10.5	8.9	5.5/1.1	0	0.0	0.3	0.5	0.9	1.2	1.4	1.6	1.8
high-to-low	5.5	125	11.7	8.9	5.5/1.2	0	0.0	0.3	0.6	1.0	1.3	1.5	1.8	2.0
high-to-low	5.0	−10	8.7	8.9	5.0/0.9	0	0.0	0.2	0.4	0.7	0.9	1.1	1.2	1.4
high-to-low	5.0	80	10.9	8.9	5.0/1.1	0	0.0	0.3	0.6	0.9	1.2	1.5	1.7	1.8
high-to-low	5.0	125	12.1	8.9	5.0/1.3	0	0.0	0.4	0.6	1.0	1.3	1.6	1.8	2.1
high-to-low	4.5	−10	9.0	8.9	4.5/1.0	0	0.0	0.2	0.5	0.8	1.0	1.2	1.3	1.5
high-to-low	4.5	80	11.3	8.9	4.5/1.2	0	0.0	0.3	0.6	1.0	1.3	1.5	1.7	1.9
high-to-low	4.5	125	12.6	8.9	4.5/1.3	0	0.0	0.4	0.7	1.1	1.4	1.7	2.0	2.2

Table 10-15. 82495XP Cache Controller Buffer Models

Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
ADS#	B15	7.7	12.0	3.8	
AHOLD	A17	9.5	16.0		S
BGT#	M03	5.9	10.1	1.9	
BLAST#	C15	7.7	11.2		L/X
BLE#	C16	8.0	14.1		S
BOFF#	G15	7.7	9.4		L/X
BRDY#	P01	7.3	12.9	1.7	
BRDYC1#	D15	8.0	10.5		S
BRDYC2#	F14	8.1	8.7		X
BUS#	P16	5.9	8.4		X
CACHE#	G14	8.1	8.4	2.3	
CADS#	E03	6.4	10.4		S
CAHOLD	G04	6.2	8.6		S
CD/C#	D03	6.7	10.2		S
CDTS#	F04	6.5	8.8		L
CFA0	E15	8.3	10.2	4.3	
CFA1	B14	7.2	11.3	4.3	
CFA2	D06	7.9	8.8	4.3	
CFA3	B02	8.8	13.5	4.3	
CFA4	A16	7.5	14.5	4.3	
CFA5	E14	7.2	9.3	4.3	
CFA6	D14	7.1	10.2	5.5	
CLK	D11	6.5	8.8	3.1	
CM/IO#	D04	7.3	10.2		S
CNA#	L04	5.8	8.7	1.7	
CRDY#	M02	6.5	10.8	1.8	
CWAY	J03	5.9	8.4	0.0	S
CW/R#	E04	7.1	9.4		S
D/C#	H14	5.8	8.4	2.3	
DRCTM#	M01	6.4	11.8	3.9	
EADS#	J15	5.9	8.4		S
EWBE#	S02	7.4	12.9		S
FLUSH#	N04	8.1	8.2	1.6	
FPFLD#	J04	5.3	8.1		

Table 10-15. 82495XP Cache Controller Buffer Models (Continued)

Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
FSIOUT#	D01	6.9	12.3	S	S
HITM#	D17	8.7	15.4	2.5	
INV	K15	5.6	9.2		S
KEN#	D16	7.9	13.5		S
KLOCK#	C03	8.1	11.5		S
KWEND#	M04	6.3	9.1	2.0	
LEN	F15	8.1	9.7	2.3	
LOCK#	B16	8.5	13.8	2.3	
MALE	Q02	7.5	11.8	1.6	
MAOE#	S04	6.8	11.7	1.6	
MAWEA#	Q17	7.2	12.6		X
MBALE	P04	7.2	10.5	1.7	
MBAOE#	P06	7.1	8.6	1.6	
MCACHE#	C02	7.6	12.1		S
MCFA0	Q16	7.6	11.5	6.6	S/L
MCFA1	N14	6.9	9.7	6.6	S/L
MCFA2	R04	7.0	11.9	6.6	S/L
MCFA3	Q06	6.6	8.9	6.6	S/L
MCFA4	P15	7.2	11.2	6.6	S/L
MCFA5	P14	7.3	10.4	6.6	S/L
MCFA6	P13	7.5	9.6	6.6	S/L
MCYC#	P17	6.9	12.0		X
MHITM#	H04	6.1	8.2		L
M/IO#	F16	8.5	11.0	2.3	
MKEN#	R01	7.7	13.8	1.7	
MRO#	J01	5.7	10.6	4.0	
MSET0	Q15	8.0	11.3	6.6	S/L
MSET1	P12	7.7	8.9	6.6	S/L
MSET10	Q11	9.7	9.5	6.6	S/L
MSET2	P11	8.8	8.2	6.6	S/L
MSET3	Q14	8.4	10.9	6.6	S/L
MSET4	R16	8.7	14.4	6.6	S/L
MSET5	Q13	8.8	10.2	6.6	S/L
MSET6	R17	8.8	16.2	6.6	S/L
MSET7	S17	9.7	16.8	6.6	S/L

Table 10-15. 82495XP Cache Controller Buffer Models (Continued)

Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
MSET8	P10	5.9	8.1	6.6	S/L
MSET9	Q12	9.3	10.1	6.6	S/L
MTAG0	Q10	6.1	9.4	6.6	S/L
MTAG1	P09	5.8	7.9	6.6	S/L
MTAG10	Q07	6.0	9.3	6.6	S/L
MTAG11	P07	6.0	8.2	6.6	S/L
MTAG2	Q09	5.9	9.1	6.6	S/L
MTAG3	R14	10.9	14.9	6.6	S/L
MTAG4	Q08	5.9	9.1	6.6	S/L
MTAG5	R15	11.0	16.3	6.6	S/L
MTAG6	S14	9.4	16.5	6.6	S/L
MTAG7	S15	10.4	18.1	6.6	S/L
MTAG8	P08	5.9	8.0	6.6	S/L
MTAG9	S16	10.3	19.6	6.6	S/L
MTHIT#	G03	6.0	9.5		L
MWB/WT#	K03	5.4	9.2	4.0	
NA#	J17	5.9	10.6		S
NENE#	D05	7.7	9.2		L
PALLC#	D02	7.1	11.3		S
PCD	H15	5.8	9.4	2.3	
PCYC	J14	5.6	8.0	2.3	
PWT	C17	9.1	15.9	2.3	
RDYSRC	C01	7.2	12.7		S
RESET	Q05	7.1	10.3	1.6	
SET0	D13	6.8	8.4	4.9	
SET1	C13	6.7	9.3	4.9	
SET10	A09	6.1	10.8	4.9	
SET2	C14	7.1	11.0	4.9	
SET3	B12	6.6	10.0	4.9	
SET4	C12	6.4	8.7	4.9	
SET5	C11	6.2	8.5	4.9	
SET6	D12	6.6	9.2	4.9	
SET7	D09	6.2	7.3	4.9	
SET8	D10	6.0	8.2	4.9	

Table 10-15. 82495XP Cache Controller Buffer Models (Continued)

Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
SET9	B09	6.1	9.6	4.9	
SMLN#	C06	8.5	9.8		L
SNPADS#	F03	6.1	9.8		S
SNPBSY#	F01	5.5	9.2		L
SNPCLK	S03	6.8	12.3	1.7	
SNPCYC#	H03	5.8	11.3		S
SNPINV	P05	7.9	9.8	1.7	
SNPNCA	Q03	7.6	11.2	1.6	
SNPSTB#	R03	7.1	12.3	1.7	
SWEND#	Q01	7.4	13.1	1.7	
SYNC#	Q04	7.7	11.3	1.7	
TAG0	C08	6.0	9.1	4.2	
TAG1	A04	8.3	14.2	4.2	
TAG10	B01	8.1	14.3	4.2	
TAG11	C05	8.2	10.4	4.2	
TAG2	D08	5.8	8.1	4.2	
TAG3	A03	8.7	15.0	4.2	
TAG4	B04	8.3	14.2	4.2	
TAG5	B03	8.5	14.7	4.2	
TAG6	C07	8.5	9.4	4.2	
TAG7	A02	9.0	15.7	4.2	
TAG8	D07	8.5	8.4	4.2	
TAG9	A01	9.4	16.1	4.2	
TCK	P03	7.0	11.2	1.7	
TDI	N03	8.1	9.6	1.6	
TDO	C04	7.7	11.0		S
TMS	P02	6.9	12.1	1.7	
WAY	L15	5.7	9.4		X
WBA	M14	6.0	8.7		X
WBTYP	N15	6.3	10.1		X
WBWE#	M15	5.9	9.6		X
WB/WT#	K14	5.6	8.1		S
W/R#	B17	8.6	15.2	2.3	
WRARR#	L14	5.8	8.3		X

Table 10-16. 82490XP Cache RAM Buffer Models

Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
A0	65	1.0	8.9	1.7	
A1	66	1.0	8.8	1.7	
A2	67	1.0	8.6	1.7	
A3	68	1.0	8.4	1.7	
A4	69	1.0	8.3	1.7	
A5	70	1.0	8.2	1.7	
A6	71	1.0	8.2	1.7	
A7	73	1.0	8.1	1.7	
A8	75	1.0	8.1	1.7	
A9	76	1.0	8.1	1.7	
A10	77	1.0	8.2	1.7	
A11	78	1.0	8.2	1.7	
A12	79	1.0	8.3	1.7	
A13	80	1.0	8.4	1.7	
A14	81	1.0	8.6	1.7	
A15	82	1.0	8.7	1.7	
ADS#	63	1.0	9.6	2.0	
BE#	64	1.0	9.1	1.7	
BLAST#	59	1.0	8.9	1.7	
BOFF#	36	1.0	9.6	2.0	
BRDY#	60	1.0	9.1	2.0	
BRDYC#	61	1.0	9.2	2.0	
BUS#	40	1.0	8.7	1.7	
CDATA0	48	1.0	8.8	3.9	S
CDATA1	54	1.0	8.6	3.9	S
CDATA2	49	1.0	8.7	3.9	S
CDATA3	55	1.0	8.6	3.9	S
CDATA4	46	1.0	9.1	3.9	S
CDATA5	51	1.0	8.6	3.9	S
CDATA6	52	1.0	8.6	3.9	S
CDATA7	57	1.0	8.7	3.9	S
CLK	30	1.0	8.1	3.4	
CRDY#	43	1.0	9.6	1.7	
HITM#	62	1.0	9.6	2.0	

Table 10-16. 82490XP Cache RAM Buffer Models (Continued)

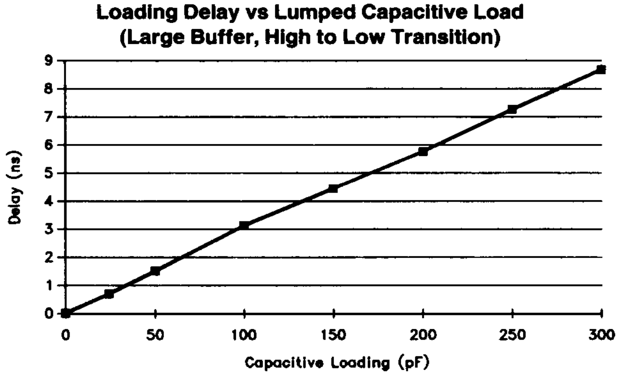
Pin Name	Location	Cp (pF) Typ	Lp (nH) Typ	Input Buffer Cin (pF) Typ	Output Buffer Size
MAWEA#	41	1.0	8.9	1.7	
MBRDY#	22	1.0	9.1	1.7	
MCLK	26	1.0	8.4	2.0	
MCYC#	42	1.0	9.1	1.7	
MDATA0	18	1.0	9.1	6.6	S/L
MDATA1	14	1.0	8.6	6.6	S/L
MDATA2	10	1.0	8.6	6.6	S/L
MDATA3	6	1.0	8.8	6.6	S/L
MDATA4	16	1.0	8.8	6.6	S/L
MDATA5	12	1.0	8.6	6.6	S/L
MDATA6	8	1.0	8.6	6.6	S/L
MDATA7	4	1.0	9.1	6.6	S/L
MDOE#	20	1.0	9.4	1.7	
MEOC#	23	1.0	8.9	1.7	
MFRZ#	24	1.0	8.7	1.7	
MOCLK	27	1.0	8.3	2.0	
MSEL#	25	1.0	8.6	2.0	
MZBT#	21	1.0	9.6	1.7	
PAR#	32	STRAPPING OPTION			
RESET	28	1.0	8.2	1.7	
TCK	3	1.0	9.2	1.7	
TDI	2	1.0	9.4	1.7	
TDO	84	1.0	9.1		S
TMS	1	1.0	9.6	1.7	
WAY	45	1.0	9.2	1.7	
WBA	38	1.0	8.4	1.7	
WBTYP	37	1.0	8.3	1.7	
WBWE#	39	1.0	8.6	1.7	
W/R#	58	1.0	9.2	1.7	
WRARR#	44	1.0	9.4	1.7	

10.6 Capacitive Derating

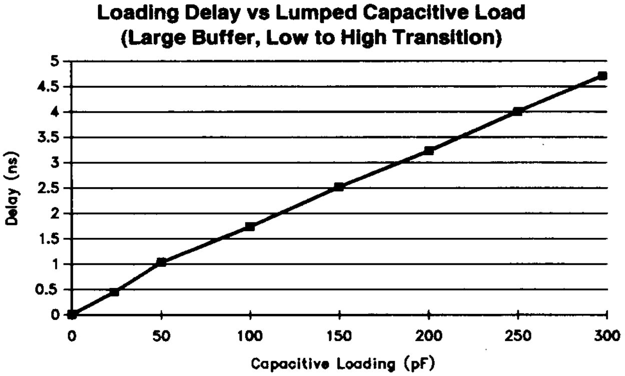
The 82495XP/82490XP Cache chip set AC Timing Specifications are all given at a 0 pF capacitive load. For this reason, each output must be derated according to the load being driven. Capacitive derating is not a precise method of determining a signal's delay, and must only be applied to signals that interface to the Memory Bus (t_{32} to t_{130}). A more accurate determination of delay and signal quality may be made by modeling the buffer using the first order buffer models.

The following graphs represent the lumped-load capacitive derating curves for the 82495XP and 82490XP buffers used to drive the memory bus. The large buffers are used for the signals CDTS#, NENE#, SMLN#, MTHIT#, MHITM#, and SNPBSY#, and when MEMLDRV is configured low for the signals MCFA, MSET, MTAG and MDATA. The small (normal) buffers are used with all other buffers and with MCFA, MSET, MTAG, and MDATA when MEMLDRV is configured high. Note that for both buffers, the capacitive derating for a low-to-high transition is different from a high-to-low transition.

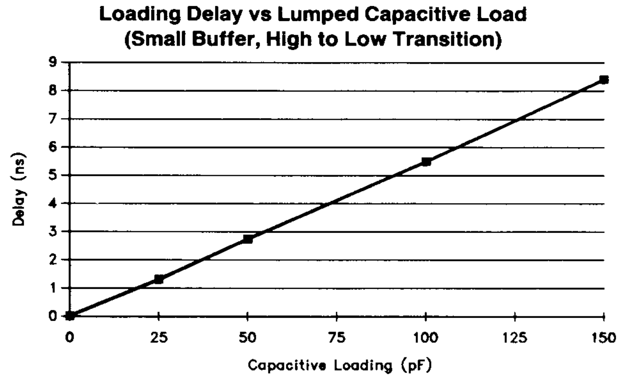
Small buffers provide the best signal quality for capacitive loads of about 50 pF or less. Large buffers provide the best quality for loads between 100 pF and 150 pF. The 82495XP/82490XP chip set component buffer model provides detailed information about buffer performance in specific environments.



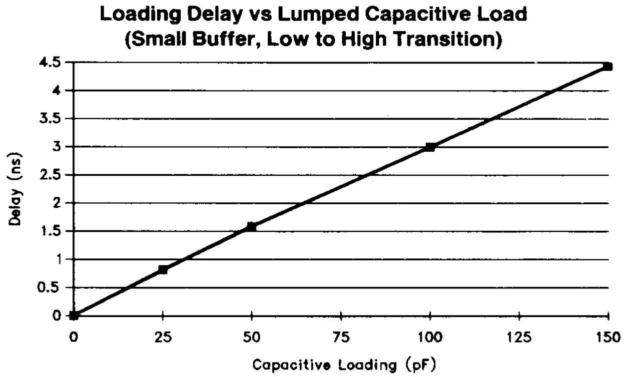
240956–68



240956–69



240956-70



240956-71

11.0 THERMAL DATA

The 82495XP and 82490XP are specified for operation when T_C (case temperature) are within the range of 0°C–85°C. T_C may be measured in any environment to determine whether the components are within the specified operating range. The case temperature should be measured at the center of the top surface, opposite the pins.

The ambient temperature (T_A) is guaranteed as long as T_C is not violated. The ambient temperature can be calculated from θ_{JC} and θ_{JA} using the following equations:

$$T_J = T_C + P \bullet \theta_{JC}$$

$$T_A = T_J + P \bullet \theta_{JA}$$

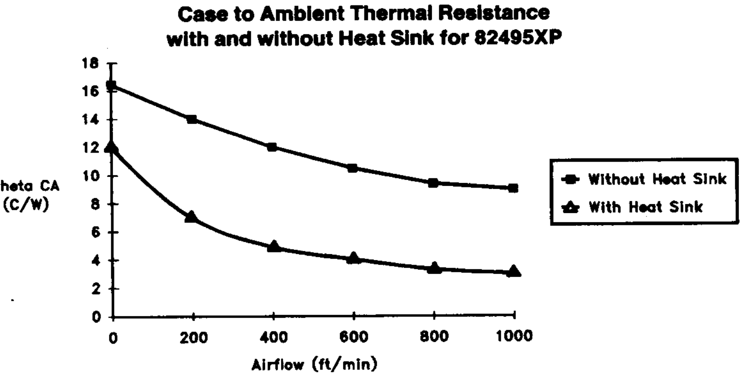
$$T_C = T_A + P \bullet (\theta_{JC} - \theta_{JC})$$

This is true where T_J, T_A and T_C = junction, ambient and case temperature, respectively, and where θ_{JC} and θ_{JA} = junction-to-case and junction-to-ambient thermal resistance, respectively. P = maximum power consumption.

The heat sink referenced for all parts is a unidirectional heat sink, 0.350″ high, 40 MIL fin width, and 155 MIL center-to-center fin spacing.

82495XP

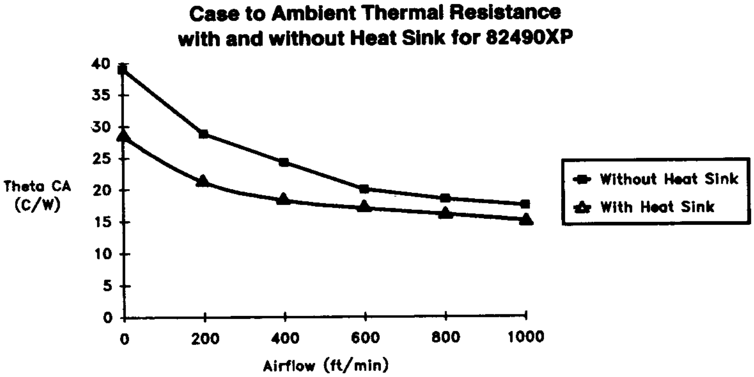
θ_{JC} (C/W) = 1.5 without heat sink
2.0 with heat sink



240956-72

82490XP

θ_{JC} (C/W) = 7.5 without heat sink
8.0 with heat sink



240956-73

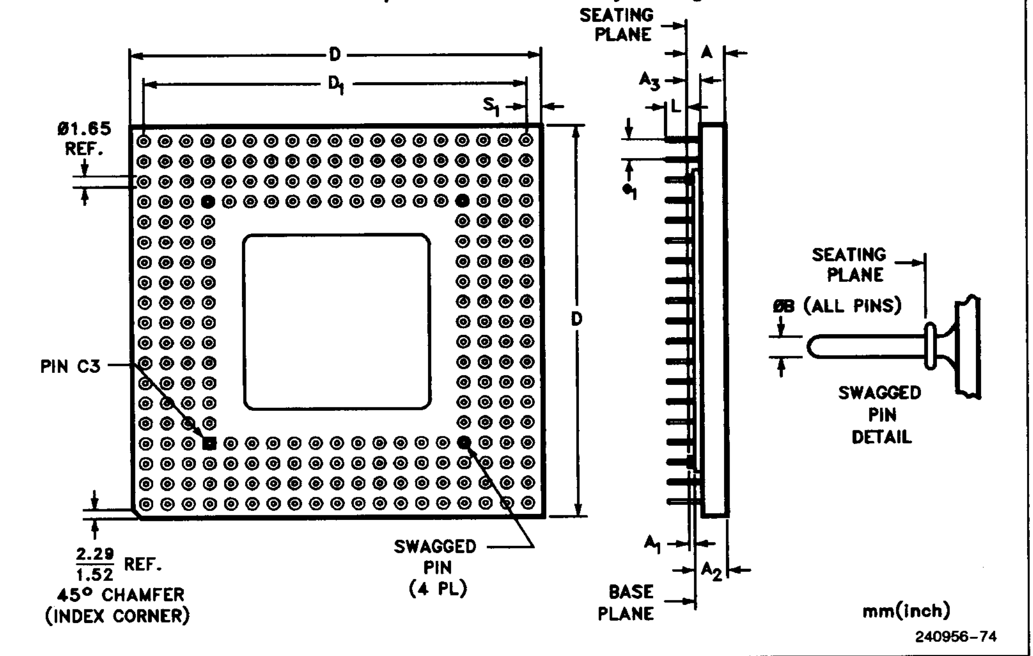
12.0 MECHANICAL DATA

Letter or Symbol	Description of Dimensions
A	Distance from seating plane to highest point of body
A_1	Distance between seating plane and base plane (lid)
A_2	Distance from base plane to highest point of body
A_3	Distance from seating plane to bottom of body
B	Diameter of terminal lead pin
D	Largest overall package dimension of length
D_1	A body length dimension, outer lead center to outer lead center
e_1	Linear spacing between true lead position centerlines
L	Distance from seating plane to end of lead
S_1	Other body dimension, outer lead center to edge of body

NOTES:
1. Controlling dimension: millimeter.
2. Dimension "e_1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by P.C. board hole size: 0.0415 inch–0.0430 inch
4. Dimensions "B", "B_1" and "C" are nominal.
5. Details of Pin 1 identifier are optional.

82495XP

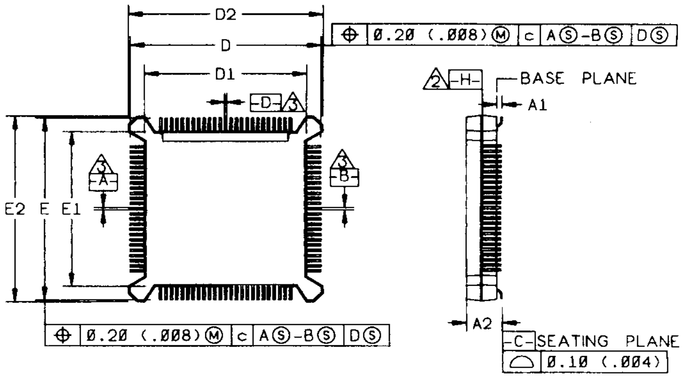
1.95" Sq Ceramic Pin Grid Array Package



Family: Ceramic Pin Grid Array Package						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
A	3.56	4.57		0.140	0.180	
A_1	0.64	1.14	Solid Lid	0.025	0.045	Solid Lid
A_2	0.23	0.30	Solid Lid	0.110	0.140	Solid Lid
A_3	1.14	1.40		0.045	0.055	
B	0.43	0.51		0.017	0.020	
D	49.53	50.17		1.950	1.975	
D_1	45.59	45.85		1.795	1.805	
e_1	2.29	2.79		0.090	0.110	
L	2.54	3.30		0.100	0.130	
N	240	280		240	280	
S_1	1.52	2.54		0.060	0.100	
ISSUE	IWS 9/90					

Figure 12-1. 82495XP Mechanical Specifications

82490XP



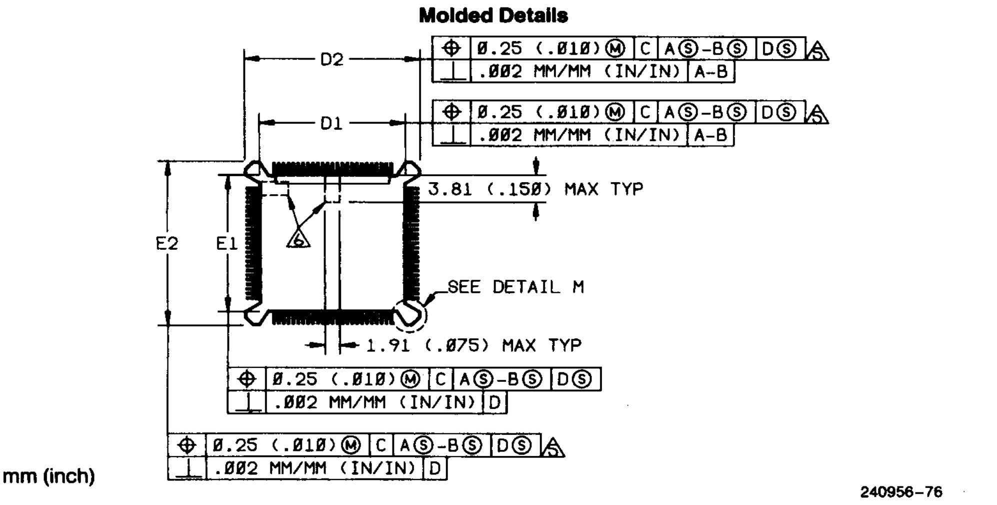
mm (inch)

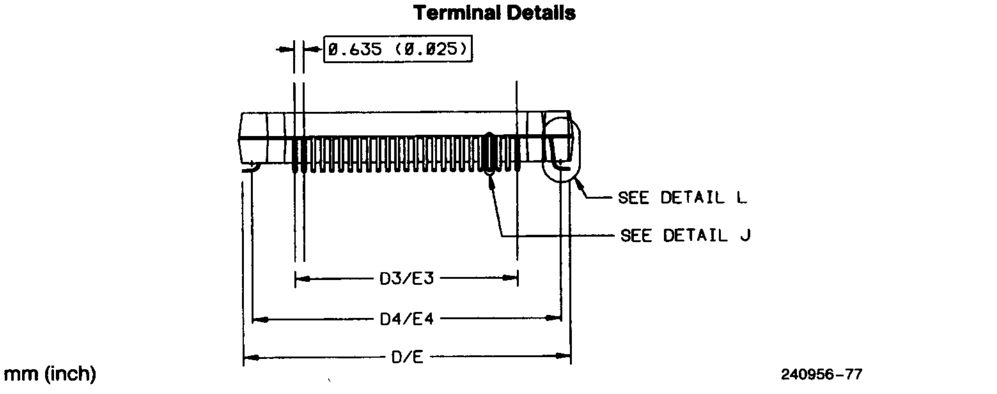
240956-75

Plastic Quad Flatpack (PQFP) 0.025 in. (0.635 mm) Pitch

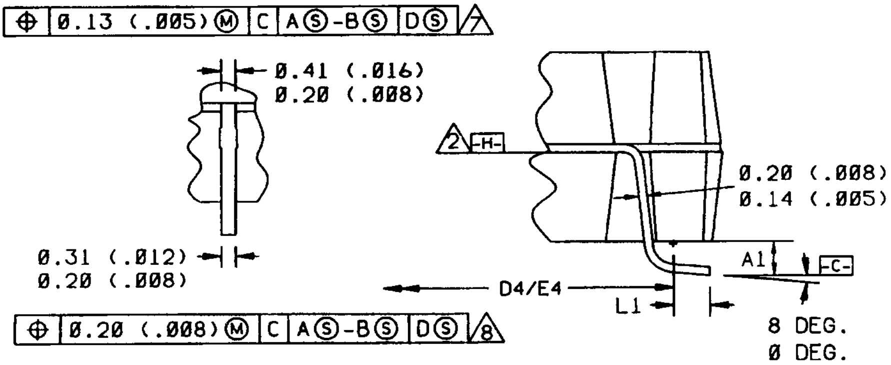
Symbol	Description	Min (mm)	Max (mm)	Min (in.)	Max (in.)
N	Leadcount	84		84	
A	Package Height	4.06	4.57	0.160	0.180
A1	Standoff	0.51	1.02	0.020	0.040
D, E	Terminal Dimension	19.56	20.07	0.770	0.790
D1, E1	Package Body	16.43	16.59	0.647	0.653
D2, E2	Bumper Distance	20.24	20.39	0.797	0.803
D3, E3	Lead Dimension	12.70 REF		0.500 REF	
D4, E4	Foot Radius Location	18.36	18.71	0.723	0.737
L1	Foot Length	0.51	0.76	0.020	0.030
Issue					

Figure 12-2. 82490XP Mechanical Specifications


Molded Details
D2
D1
⌖ Ø.25 (.Ø1Ø) Ⓜ C A Ⓢ-B Ⓢ D Ⓢ 5
⊥ .ØØ2 MM/MM (IN/IN) A-B
3.81 (.15Ø) MAX TYP
E2
E1
6
SEE DETAIL M
1.91 (.Ø75) MAX TYP
⌖ Ø.25 (.Ø1Ø) Ⓜ C A Ⓢ-B Ⓢ D Ⓢ
⊥ .ØØ2 MM/MM (IN/IN) D
mm (inch)
240956–76


Terminal Details
Ø.635 (Ø.Ø25)
SEE DETAIL L
SEE DETAIL J
D3/E3
D4/E4
D/E
mm (inch)
240956–77

Typical Lead



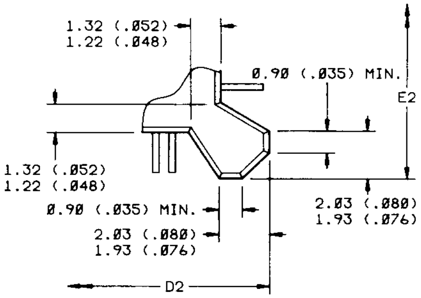
mm (inch)

240956-78

Detail J

Detail L

Detail M



240956-79

mm (inch)

13.0 REVISION HISTORY

The following list represents the major differences between version -002 and version -001 of the i860 XP Microprocessor Data Sheet.

Section 2.2.4	**AI** bit has been changed to **TAI** in Figure 2.5.
	The explanation for **PI** bit has been expanded.
Section 4.2.33	**PCHK#** signal description has been expanded.
Section 4.2.35	Output buffer configuration has been added in **PEN#** signal description.
Section 5.1.3	Table 5.2 has been corrected.
Section 5.2.2.4–5	The explanation of late back-off mode has been expanded.
Section 5.2.4	Figure 5.27 has been corrected.
Section 9.2	D.C. Characteristics are corrected.
Section 9.3	A.C. Characteristics are replaced with nominal timings based on $C_L = 0$ pF.
	Figure 9.3 and Figure 9.4 have been replaced with nominal A.C. Timings based on $C_L = 0$ pF.
	Figure 9.5 has been corrected for normal and high-current output buffers.
Section 9.4	Component mbuffer model has been added.
Section 10.4	Programming restrictions on **flush** instruction has been added.