sun microsystems

SPARCstation 1 S4 Chip Set

This documentation package includes all of the specifications and electrical characterization reports for the SPARCstation 1 S4 Chip Set.

The "Campus 1" Programmer's Model is also included.

960–1043–03

B. PIN DESCRIPTION:
==========

SYMBOL	TYPE	DESCRIPTION
IU INTERFACE	34	
CLK	DRVC16	Clock
/IU D(31:0)	BD4TRU	Integer Unit Data bus
/IU MEXC	IBUFNU	Integer Unit Memory Exception
SCHOOLBUS INTERFACE	39	
/SB A(1:0)	IBUFU	SBus Address Bus
/SB SIZ(1:0)	IBUF	SBus Size
/SB /AS	IBUF	SBus Address Stroke
/SB RD	IBUFU	SBus Read
/SB D(31:0)	BD4TRU	SBus Data bus
/SB /ACK32	IBUFNU	SBus 32-bit Acknowledge
/SB /ACK8	IBUFNU	SBus 8-bit Acknowledge
/SB /ERR	BT8	SBus Error
/SB /RESET	IBUFN	SBus Reset
MISCELLANEOUS	20	
PAR(3:0)	BD4TRU	Parity data bus
/PAR /EN	IBUFN	Parity Enable
/PAR /CS	IBUFN	Parity Chip Select
/PIO /SEL	IBUFN	Parallel I/O Port Select
PIO(6:0)	BD4TOD	Parallel I/O Port
IOD(7:0)	BD4TRU	Input Output Data bus
/IOD /EN	IBUFN	Input Output Data bus Enable
/WB /OE	IBUFN	Write Buffer Output Enable
/WB /CE	IBUF	Write Buffer Clock Enable
/CD /OE	IBUFN	Catche Data Output Enable
/S4B /OD	IBUFNU	S4-Buffer chip Output Disable
/S4b /TEST	IBUFNU	Low for chip test mode
PARA	BT1	Parametric Test Output

INPUT/OUTPUT BUFFER DEFINITIONS

DRVC# : Input Clock Buffer, CMOS, non-inverting. # indicates drive capability.
IBUF : Input buffer, CMOS, non-inverting
IBUFU : Input buffer, CMOS, non-inverting, internal pullup
IBUFN : Input buffer, CMOS, inverting
IBUFNU : Input buffer, CMOS, inverting, internal pullup
TLCHT : Input buffer, TTL, non-inverting
TLCHTN : Input buffer, TTL, inverting
BD#TRU : Bidirectional buffer, TTL input levels, # indicates

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
tTCT	TCLK Period	40		ns
tTCH	TCLK High Time	17		ns
tTCL	TCLK Low Time	17		ns
	TCLK to Output Valid		20	ns
	TCLK to Output Invalid	3		ns
	Input Setup to Clock	15		ns
	Input Hold Time to Clock	0		ns

Output drive, internal pullup.

BT# : Tri-statable output buffer, CMOS, # indicates output drive current.

BD4TOD : Open drain buffer, TTL, non-inverting

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	Typ	Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

7. AC OPERATING REQUIREMENTS*
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t	From XCLK High to XCLK High	40		ns
t	From CLK to /IU D	10.5	23.5	ns
t	From CLK to /SB D	18.0	27.0	ns
t	From CLK to PAR	23.0	34.5	ns
t	From CLK to PIO	12.0	21.5	ns
t	From CLK to IOD	8.0	31.5	ns
t	From CLK to /SB MERR	16.0	23.5	ns
t	From /IU MEXC to /IU D	6.5	12.0	ns
t	From /WB /OE to /SB D	5.5	21.0	ns

* Setup time for all signals is 15ns
Hold time for all signals is 3ns.

8. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CI/O	Input/Output Capacitance per Slot		20	pF
CLS	Maximum Capacitance Load per System		100	pF

9. TRUTH TABLE: N/A

10. FUNCTIONAL TABLE: N/A

11. OUTLINE DRAWING: N/A

12. WAVEFORMS: N/A

DATE: 1/04/89

AUTHOR: PL

~~~~~~
~~~~~~

Contents

960-1043-03

SPARCstation-1 Programmer's Model

Stephen A. Chessin

DRAFT 8
Version 8.4, 90/02/11

1. Introduction

This paper describes the programmer's view of SPARCstation-1: address spaces, caching and memory management, and interrupt levels. It is a synthesis of information contained in the hardware specifications, but organized to be useful to a programmer.

Where appropriate, comparisons with the "standard" Sun4 architecture are made.

WARNING: This document is a DRAFT and may contain errors. Please report all mistakes to the author for correction.

Major Changes Since Draft 1 (Version 1.7)

(1) The page size has changed from 8K to 4K.

(2) The size of a physical address has changed from 29 bits to 28 bits.

(3) The Sbus has moved from Type 0 space to Type 1 space, and there has been a major reorganization of the Type 1 addresses to accommodate this.

Changes Since Draft 2 (Version 2.4)

(4) Minor typographical and editorial changes.

(5) Better explanations.

Changes Since Draft 3 (Version 3.7)

(6) The Interrupt Register is used to clear level 15 interrupts.

(7) All Sbus devices are now described using relative offsets.

(8) More bits are used in the Auxiliary Input/Output Register. (Which used to be the Auxiliary Output Register.)

Changes Since Draft 4 (Version 4.7)

(9) The interrupt levels have been changed slightly. All Sbus devices, including the builtin ones, interrupt on Sbus IRQ levels only.

(10) The Auxiliary Input/Output Register has changed slightly.

(11) The definition of the DMA Write bit was backwards.

(12) The video subsystem is off the board, again.

Changes Since Draft 5 (Version 5.6)

Better explanations and addition of more examples.

Changes Since Draft 6 (Version 6.1)

(1) Added warnings that this is still a DRAFT document and may not be completely accurate.

(2) Described the bugs in various levels of hardware:

Synchronous parity errors cause asynchronous traps (fixed in P1.7)
SER records asynchronous errors (won't be fixed)
ASER and ASEVAR latch on synchronous memory errors (won't be fixed)
On cache fill errors, SEVAR may not have exact address of problem (won't be fixed)

ASER sometimes isn't set on asynchronous errors (won't be fixed)
ASEVAR isn't properly sign-extended on DVMA errors (won't be fixed)

(3) Audio/ISDN replaces Audio DAC.

(4) Level 8 interrupts can be masked.

(5) Video goes into slot 3.

(6) Sbus IRQ6 and IRQ7 now map to SPARC level 8 and 9, instead of 9 and 13, respectively.

(7) Miscellaneous corrections.

Changes Since Draft 7 (Version 7.15)

(8) Changed name to SPARCstation 1

(9) Added section describing how to handle simultaneous errors.

2. Address Spaces

The SPARC Architecture defines the existence of at least 4 address spaces. A given implementation may define more than 4 address spaces. Selection of a particular address space is done via the Address Space Indicator (ASI) field of the load and store alternate address space instructions. Ordinary load and store instructions automatically go to User or Supervisor Data space, depending upon the mode of the CPU. Instruction fetches by the CPU automatically go to User or Supervisor Instruction space, again depending upon the mode of the CPU.

The following table describes the address spaces defined by the Sun4 Architecture and the SPARCstation-1 implementation.

ASI	Sun4 Use	SPARCstation-1 Use	Comments
0x0	Reserved	Reserved	
0x1	Reserved	Reserved	
0x2	System Space	Same	Note 1
0x3	Segment Map	Same	
0x4	Page Map	Same	
0x5	Block Copy	Reserved	Note 2
0x6	Region Map	Reserved	Note 2
0x7	Flush Cache (Region)	Reserved	Note 2
0x8	User Instruction	Same	
0x9	Supervisor Instruction	Same	
0xA	User Data	Same	
0xB	Supervisor Data	Same	
0xC	Flush Cache (Segment)	Same	
0xD	Flush Cache (Page)	Same	
0xE	Flush Cache (Context)	Same	
0xF	Flush Cache (User)	Reserved	Note 3
0x10	Flush I-Cache (Segment)	Reserved	Note 2
0x11	Flush I-Cache (Page)	Reserved	Note 2
0x12	Flush I-Cache (Context)	Reserved	Note 2
0x13	Flush I-Cache (User)	Reserved	Note 2
0x14	Flush D-Cache (Segment)	Reserved	Note 2
0x15	Flush D-Cache (Page)	Reserved	Note 2
0x16	Flush D-Cache (Context)	Reserved	Note 2
0x17	Flush D-Cache (User)	Reserved	Note 2
0x1B	Flush I-Cache (Region)	Reserved	Note 2
0x1F	Flush D-Cache (Region)	Reserved	Note 2

Note 1. See System Space table (next section)
Note 2. SPARCstation-1 has no corresponding function.
Note 3. This is a change in the specification between Sunrise and Sunray.

User and Supervisor Instruction and Data spaces are collectively known as "Device Space". All accesses to Device Space go through the Memory Mangement Unit (MMU). All the other address spaces are collectively known as "Control Space". The non-System Space portions of Control Space all deal with Cache and MMU management, and are discussed in the section on "Contexts, Caching, and the MMU". System Space is discussed in the next section.

3. System Space (ASI = 2)

System Space is a portion of control space that is used to access various devices, as the following table indicates:

A31:28	Sun4 Use	SPARCstation-1 Use	Comments
0x0	ID Prom	Reserved	Note 1
0x1	Reserved	Reserved	
0x2	Reserved	Reserved	
0x3	Context Register	Same	
0x4	System Enable Register	Same	
0x5	Reserved	Reserved	
0x6	Bus Error Register	Bus Error Registers	Note 5
0x7	Diagnostic Register	Unused	Note 2
0x8	(D-)Cache Tags	Cache Tags	
0x9	(D-)Cache Data	Same	Note 3
0xA	I-Cache Tags	Reserved	Note 4
0xB	I-Cache Data	Reserved	Note 4
0xC	Reserved	Reserved	
0xD	Reserved	Reserved	
0xE	VME Interrupt Vector	Reserved	Note 4
0xF	Serial Port	Same	MMU bypass

Note 1. SPARCstation-1 does not have an ID Prom and a timeout will occur.
Note 2. SPARCstation-1 has no diagnostic register but a write to this address will just be ignored and not cause a timeout.
Note 3. This is a change in the specification between Sunrise and Sunray.
Note 4. SPARCstation-1 has no corresponding function.
Note 5. SPARCstation-1 has four Bus Error Registers, compared to Sun4's one.

The Context Register, Cache Tags, and Cache Data are described in the section on "Contexts, Caching, and the MMU". The rest of the registers in System Space are described below.

3.1. System Enable Register

The System Enable Register is referenced via byte loads and stores at location (ASI=0x2, A31:28=0x4). It has the following format:

```
 7 6 5 4 3 2 1 0
v---------------v
|N|0|S|C|0|R|0|D|
^---------------^
```

N	ENA_NOTBOOT	0 = all supervisor references go to EPROM 1 = normal MMU operation
	-	Reserved (Enables I/O Cache in Sun4)
S	ENA_SDVMA	1 = all DVMA is enabled
C	ENA_CACHE	1 = Cache enabled
	-	Reserved (Enables video display in Sun4)
R	ENA_RESET	1 = Reset the System (asserts SBRESET)
	-	Reserved (Resets VMEbus in Sun4)
D	ENA_DIAG	Always 0 (Diagnostic/Monitor in Sun4)

All bits are initialized to zero by a reset. Setting ENA_RESET to one will cause a reset, and control will not be returned to the program that does so; rather, a reboot will occur. Software (or the boot PROM) should set ENA_NOTBOOT to one after initializing the MMU.

3.2. Bus Error Registers

There are four registers, divided into two sets of two, used to indicate the type and location of bus errors. One set is for synchronous errors, and the other for asynchronous errors. Synchronous errors are those that occur due to the execution of the current instruction and are reported to the CPU by a trap at the end of that instruction's execution. All errors that cannot be associated with the execution of the current instruction, but are related to such things as DVMA activity, buffered writes, or cache write-back[1], are considered asynchronous and are reported via an interrupt on level 15. After servicing the level 15 interrupt, it is cleared by toggling bit 0 of the Interrupt Register.

There is an exception to the above rule. On machines prior to the P1.7 level, parity errors that occur (or any condition that causes SE_MEMERR, described below, to be set) during CPU memory accesses cause the reporting of both a synchronous and asynchronous error. For parity errors that occur during data fetches, the data-access trap occurs first and the level 15 interrupt remains pending. Software may clear the level 15 interrupt while processing the data-access trap. For parity errors that occur during instruction fetches, the level 15 interrupt occurs first and the text-access trap never occurs. Software can distinguish true asynchronous errors from instruction fetch errors by maintaining an invalid value in the SEVAR and comparing the SEVAR to the ASEVAR on asynchronous errors. If they compare equal, then this is an instruction-fetch error, otherwise it is a true asynchronous error. Software must remember to reload the SEVAR with the invalid value after processing all synchronous (including instruction-fetch) errors.

On P1.7 and later boards, memory errors during CPU memory accesses only cause the reporting of a synchronous error; a level 15 interrupt does not occur. (The asynchronous registers still latch on synchronous memory errors, however, and must be cleared; see the descriptions of the ASER and ASEVAR, below.)

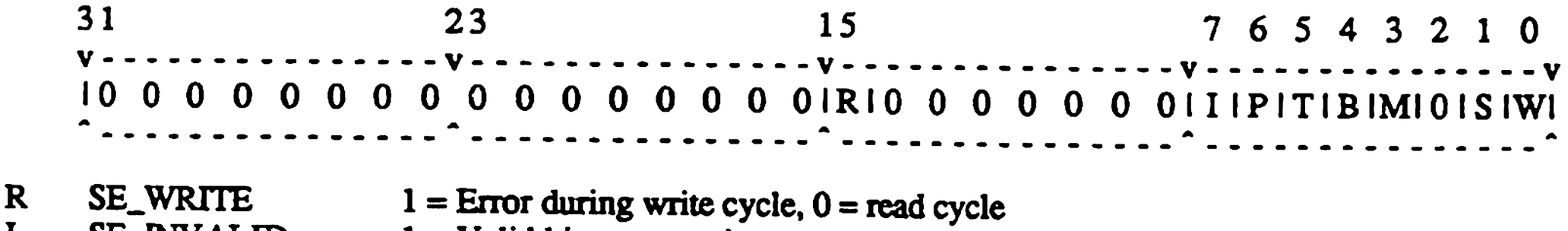
The Bus Error Registers are all fullword in size, although they can be accessed via byte, halfword, or fullword loads and stores, just as memory is. They reside at the following addresses in ASI=2 space:

Address	Description
0x60000000	Synchronous Error Register
0x60000004	Synchronous Error Virtual Address Register
0x60000008	Asynchronous Error Register
0x6000000C	Asynchronous Error Virtual Address Register

Although in normal use the registers can be treated as read-only, they can be written for diagnostic purposes.

3.2.1. Synchronous Error Register

The Synchronous Error Register (SER) occupies four bytes at locations (ASI=0x2, A31:28=0x6, A3:0=0x0 to 0x3). Reading any portion of the register also clears that portion. It has the following format:



R	SE_WRITE	1 = Error during write cycle, 0 = read cycle
I	SE_INVALID	1 = Valid bit was zero in a page map entry
P	SE_PROTERR	1 = Protection error (see below)
T	SE_TIMEOUT	1 = Non-existent device was addressed
B	SE_SBERR	1 = bus error during Sbus master access

[1] SPARCstation-1 does not have a write-back cache, but if it did it could cause asynchronous errors.

M	SE_MEMERR	1 = Memory (parity or ECC) error
S	SE_SIZERR	1 = Incorrect size transfer attempted
W	SE_WATCHDOG	1 = Restart due to IU error

The SER records all errors since it was last cleared. **This includes asynchronous errors as well; the SER must be read to clear it as part of asynchronous error processing.** The SE_WRITE bit records the type of access (read or write) of the last error.

A protection error can be caused by an attempted write to a read-only page, or by a user-mode access to a supervisor-only page.

A timeout is reported on access to a non-existent device, except for accesses to non-existent physical memory. See the section "Type 0 Space," below.

The Memory Error Register must be inspected when a memory error occurs, to further isolate the cause of the error. Note that synchronous memory errors also cause the Asynchronous Error Register and Asynchronous Error Virtual Address Register to be latched; see the description of these registers below for more information.

Not all bus errors cause immediate traps. Due to pipelining, the CPU fetches instructions four cycles before they will be executed, so it is possible that the CPU will attempt to fetch an instruction that will not, in fact, be executed. To prevent spurious traps, the CPU does not trap on memory exceptions until it actually needs to execute the instruction that it was unable to fetch.

For example, suppose we have the following instruction sequence in virtual memory, where a, b, c, etc. represent miscellaneous instructions:

```
a
b
bz,a                    label
d
---                     page boundary
e  <--this page is marked invalid
f
g
----                    page boundary
label:                  <--this page is valid
x
y
z
```

These instructions will advance through the pipeline as follows:

Time	1	2	3	4	5
Fetch	d	-	x	y	z
Decode	bz	d	-	x	y
Execute	b	bz	d	-	x
Write	a	b	bz	d	-

At time (2), the CPU wants to fetch e but the page is marked invalid, so the invalid bit is set in the SER and the instruction address is set in the SEVAR. However, the branch (if taken) means that e is never needed, so that it would be incorrect for the CPU to trap on a page fault due to the attempt to fetch e.

Now let's examine the following sequence:

```
a
b
st something to a read-only page
d
----    page boundary
e       <--this page is marked invalid
f
g
```

The pipeline now looks as follows:

Time	1	2	3	4	5
Fetch	d	-	-	x	y
Decode	st	d	-	-	x
Execute	b	st	d	-	-
Write	a	b	st	-	-

The attempt to fetch e from an invalid page at time (2) will turn on the SE_INVALID bit in the SER, but the CPU will not take an instruction access exception until it actually needs to execute e, at time (5). The store to a read-only page at time (3), however, does result in an immediate data access exception, and the CPU will find both the SE_INVALID bit and the SE_PROTERR bit on in the SER. (The exception results in a flush of the pipe, and instruction d never does get to the Write stage in step (4)).

A similar scenario, where the store is replaced by a branch (in user mode) to a supervisor-only page, can result in multiple bits being on for instruction access exceptions.

It is up to the software to determine the true cause of the exception when multiple bits are on in the SER. Here is one algorithm:

```
SEVAR = getsevar();
SER = SERsave = getser();
SER &= ~(SE_WRITE | SE_WATCHDOG);
if (data access exception)
        error_addr = SEVAR;
else if (instruction access exception)
        error_addr = old PC;
else
        /* CAN'T HAPPEN */;

if (SER & (SER - 1)) {
        /* multiple bits on; must manually probe the PME */
        pme = getpme(error_addr);
        if (pme valid) {
                if ((SER & SE_PROTERR) && (pme denies access)) {
                        SER = SE_PROTERR;
                } else
                        SER &= ~(SE_PROTERRISE_INVALID);
        } else
                SER = SE_INVALID;
}

/*
 * Note: we could still have other multiple bits on (TIMEOUT,
 * MEMERR, SIZERR, SBERR), but we probably won't recover from
 * this condition anyway, so it really doesn't matter.
 *
 * But if you really wanted to, know you'd do something like
 * this:
 */

/* more than one of TIMEOUT, SBERR, MEMERR, or SIZERR */
(void) getser();  /* make sure it's clear */
if (on_fault())
        newSER = getser();
else {
        register int x;

        newSER = 0;
        x = *error_addr; /* probe the address to see what happens */
```

```
}
    no_fault();
    /* use newSER to figure out what the problem was, if any */
```

3.2.2. Synchronous Error Virtual Address Register

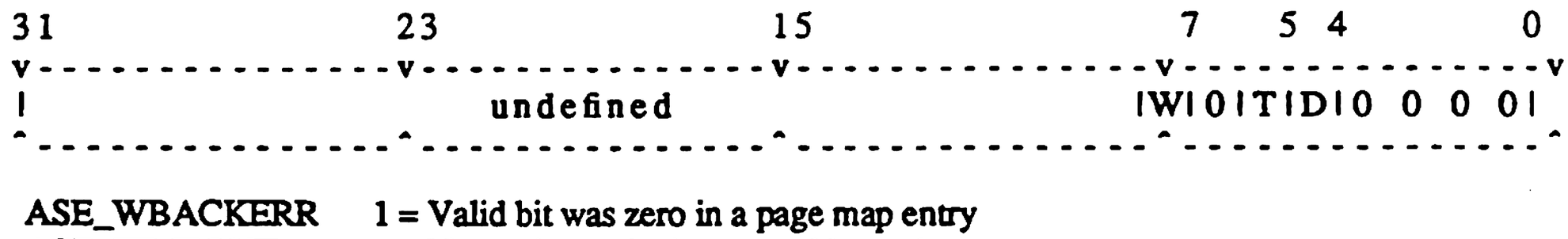
The Synchronous Error Virtual Address Register (SEVAR) occupies four bytes at locations (ASI=0x2, A31:28=0x6, A3:0=0x4 to 0x7). It contains the virtual address associated with the last synchronous bus error. It is not latched.

Note that on errors resulting from cache-fill operations, the SEVAR will contain the address that the CPU presented to the cache chip that triggered the cache-fill operation. This may or may not be the address of the word that actually caused the error.

The SEVAR has the following format:

3.2.3. Asynchronous Error Register

The Asynchronous Error Register (ASER) occupies four bytes at locations (ASI=0x2, A31:28=0x6, A3:0=0x8 to 0xB). Reading any portion of the register also clears that portion. It has the following format:



W	ASE_WBACKERR	1 = Valid bit was zero in a page map entry
T	ASE_TIMEOUT	1 = Non-existent device was addressed
D	ASE_DVMAERR	1 = bus error during DVMA access

The ASER latches (freezes) with the cause of an asynchronous error, ignoring subsequent asynchronous errors, until read and cleared. **It is also latched when a synchronous memory error (SE_MEMERR) occurs, and should be read to unlatch it as part of SE_MEMERR processing. Note that bits in the SER are set when bits in the ASER are set; thus the SER should be read to clear it as part of asynchronous error processing.**

A write-back error can occur on systems with write-back caches, and/or on systems that do buffered writes, when either the hardware malfunctions or the MMU mapping is changed without properly flushing the cache. In addition, certain devices (for example, frame buffers) will generate write-back errors under device-specific conditions when a store is attempted to them.

A timeout is reported on access to a non-existent device, except that accesses to non-existent physical memory may produce detectable behavior other than timeouts. (See the section "Type 0 Space," below.) For SPARCstation-1, this can only happen if the MMU is set up to map a non-existent device or if the hardware malfunctions.

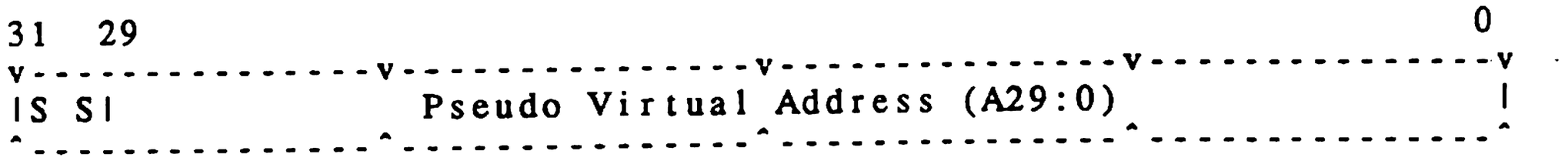
The specific cause of a DMVA bus error must be determined by polling the possible sources to see which indicated the error. All possible sources of DVMA errors of this type must be recognizable in some way. For SPARCstation-1, the only possible source of DVMA bus errors is memory parity errors. These can be determined by examining the Memory Error Register, described below.

Due to a bug in the cache chip, the ASER is not always set when an asynchronous error occurs. In this event, the ASER can be reconstructed from the bits in the SER. SE_MEMERR should be on in the SER. In addition, SE_TIMEOUT indicates that ASE_TIMEOUT should have been reported, and SE_SBERR indicates that ASE_WBACKERR should have been reported. The address in the ASEVER is correct even when the ASER is not set. This bug is in all versions of the hardware, including P1.7's, and is not expected to be fixed.

3.2.4. Asynchronous Error Virtual Address Register

The Asynchronous Error Virtual Address Register (ASEVAR) occupies four bytes at locations (ASI=0x2, A31:28=0x6, A3:0=0xC to 0xF). It contains the (pseudo) virtual address associated with the asynchronous bus error described in the ASER. It is latched under the same conditions that the ASER is latched. It is unlatched when it is read, not when the ASER is read. Thus, the ASEVAR should be read to unlatch it as part of SE_MEMERR processing.

The ASEVAR has the following format:



S Bits 31:30 are copies of bit 29.

The address is called a "pseudo-virtual" address because the hardware only carries the low-order 30 bits of the virtual address onto the bus, and assumes that bits 31:29 are all the same. The ASEVAR reverses this process by copying bit 29 into bits 31 and 30 on asynchronous errors reported by the IU. **Due to a bug in the cache chip, bits 31 and 30 are zero on DBMA asynchronous errors (ASE_DVMAERR is on). Software must do the sign extension itself.**

Determining the context register value associated with an asynchronous error is usually straightforward; there is only one tricky case.

Since DVMA is always done using context 0, the address associated with a DVMA error will always be context 0.

Non-DVMA asynchronous errors are due to buffer chip activity. The buffer chip allows only one outstanding store; a subsequent store will stall the CPU in the middle of execution of the second store until the outsanding store completes. If it completes with an asynchronous error, the error will be reported to the CPU immediately after execution of the second store instruction finishes. (This is not necessarily completion of the second store itself, as it may itself be buffered. This is just completion of the store instruction from the CPU's point of view.) Unless the second store is a write to the context register, the address of the asynchronous error will be associated with the value in the context register (the current context).

If the second store does modify the context register, then the address of the asynchronous error is associated with the previous context, which must be determined by software. (If, for example, the first store was to a supervisor-only page, then the actual context is irrelevant as supervisor-only pages are mapped into all contexts.)

One can construct pathological cases where it would be impossible to determine that an asynchronous error is associated with the previous context (for example, a store to a user page, followed by a branch, with the store to the context register in the delay slot of the branch). It is up to software to avoid these pathologies.

3.2.5. Simultaneous errors

It is possible for both a synchronous and an asynchronous error to be reported simultaneously. Consider the following case:

```
st      %g0, [%l0]! this address causes an asynchronous timeout
st      %g0, [%l1]! this address causes a page fault
```

Depending upon the alignment of the instructions in the cache, it is possible for the IU to take the page fault trap (a synchronous error) first, and while it is disabled for traps but before the SER has been read, the asynchronous fault can be reported. This will turn on the MEMERR bit in the SER, which can lead software to believe that this is a synchronous memory error. Since the MEMERR is really asynchronous, there will be a level 15 interrupt pending. If software treats this error as synchronous, and diligently reads the SER, SEVAR, ASER, and ASEVAR to clear and/or unlatch them, then when traps are eventually enabled and the level 15 interrupt occurs software will discover that there is no information in either the ASER or the SER pertaining to the asynchronous interrupt.

Software can avoid this difficulty by comparing the ASEVAR to the SEVAR whenever MEMERR is set on a synchronous trap. If they are identical, then this is a true synchronous MEMERR. If they are different, then the MEMERR is associated with the asynchronous trap. Software should clear the pending level 15 interrupt and process the asynchronous error, using the ASER and ASEVAR values, and being cognizant of the bugs in asynchronous error reporting described previously. The synchronous error can be ignored, for it will recur if and when execution of the program is resumed.

3.2.6. Serial Port

The serial port is referenced by byte loads and stores at locations beginning at (ASI=0x2, A=0xF0000000). This access is provided so that the serial port may be used before the MMU has been initialized, for example by the PROM monitor. Software normally accesses the serial port via I/O space through the MMU.

See the section "Serial Ports" under "Type 1 Space", below, for more information on the serial port registers.

4. Physical Space

The MMU maps virtual addresses in Device Space to physical addresses in Physical Space. Physical space is further subdivided into four types, as indicated in the following table.

Type	Sun4 Use	SPARCstation-1 Use	Comments
0	Main Memory	Same	
1	I/O Space	Same	
2	VMEbus, 16-bit data	Unused	Note 1
3	VMEbus, 32-bit data	Unused	Note 1

Note 1. In SPARCstation-1, references to type 2 or 3 space cause a timeout.

The size of a physical address is 28 bits.

4.1. Type 0 Space

Type 0 space contains the main memory (RAM) in SPARCstation-1. Since PA27:0 are used for RAM device decoding, the Sbus can support a theoretical maximum of 256 Mbytes of RAM. However, the SPARCstation-1 implementation only supports a maximum of 64 Mbytes. In addition, individual SPARCstation-1 machines can be configured with as little as 4 Mbytes of memory. To explain what happens when non-existent RAM is addressed, the implementation must be explained and some terms defined.

The SPARCstation-1 memory subsystem contains two RAM controllers. Each RAM controller controls a "bank" of 32 Mbytes of address space. Each bank is made up of two "sets" spanning 16 Mbytes each. Each set contains four SIMMs (Single Inline Memory Modules) each. Each SIMM consists of 9 chips. Each chip is either a 1 Mbit or a 4 Mbit DRAM. All the chips in a SIMM are of the same type, and all the SIMMs in a set must be of the same type. A set of 1 Mbit DRAMs contains 4 Mbytes of memory, and a set of 4 Mbit DRAMs contains 16 Mbytes of memory. The SIMMs in one set can be of a different type than the SIMMs in another set, even in the same bank.

The RAM controllers require PA27 to be zero. If PA27=1, then no controller responds and a bus timeout occurs.

PA26:25 selects the appropriate RAM controller. One controller responds to 0x0, the other responds to 0x1. If PA26:25=2 or 3, then no controller responds and a bus timeout occurs.

PA24 selects one of the two sets of SIMMs controlled by a controller. If the selected set is not installed (a hole), then on writes the data is thrown away and on reads the bus lines remain high (subject to noise) and a characteristic bit pattern (normally all ones) is returned. Software can detect a hole by doing a store to followed by a load from a byte on 16 Mbyte boundary. If the data read does not agree with the data written, then a hole exists. If they agree, the same test with a different bit pattern should be used before concluding that real memory exists. (Note that parity checking should be disabled when doing these checks, as parity errors will be reported if the noise pattern contains bad parity and parity checking is enabled.)

If the selected set consists of 4 Mbit DRAMs, then all 16 Mbytes of address space spanned by that set are valid and correspond to unique memory locations. If the selected set consists of 1 Mbit DRAMs, then only 4 Mbytes of unique memory exist, but it appears four times in the 16 Mbytes of address space spanned by the set, repeating at every 4 Mbyte boundary. This "mirror" behavior can be detected by software by doing a store of one bit pattern to offset 0 of a set, followed by a store of another bit pattern to offset 4 Meg (0x00400000) of the set, followed by a load from offset 0. If the data at offset 0 was changed by the store to offset 4 Meg, then only 4 Mbytes of memory is present and the rest is filled with mirrors.

The following decision table summarizes this behavior.

PA27	PA26	SIMM Set	PA23:22	Action
1	-	-	-	Timeout
0	1	-	-	Timeout
0	0	none	-	Hole
0	0	4 Mbit	-	Memory (16 Mbytes worth)
0	0	1 Mbit	00	Memory (4 Mbytes worth)
0	0	1 Mbit	01	Mirror
			10	
			11	

4.2. Type 1 Space

Type 1 space contains all of the I/O devices, including those that are associated with the Sbus. Bit PA27 is used to indicate an onboard device (PA27=0) or an Sbus device (PA27=1). For onboard devices, PA26:24 (and in some cases PA26:20) determine the particular device. For Sbus devices, PA26:25 select one of four Sbus slots. The (physical or logical) board plugged into the Sbus slot then has an address space of 25 bits, or 32 Mbytes, to divide up as it sees fit. Sbus addressing is further described in the Section "Sbus Devices", below. For compatibility with the Sun4 architecture conventions, the non-existent bits (PA31:28) are assumed to be all ones. The following table describes the layout of Type 1 space:

Address	SPARCstation-1 Use	Comments
0xF0000000	Keyboard/Mouse	Note 1
0xF1000000	Serial Ports	Note 1
0xF2000000	TOD Clock and NVRAM	Note 2
0xF3000000	Counter-Timer Registers	Note 3
0xF4000000	Memory Error Registers	Note 1
0xF5000000	Interrupt Register	Note 1
0xF6000000	EPROM	Note 3
0xF7000000	EPD "Private":	Note 4
0xF7100000	ECC registers	(HPD only)
0xF7200000	Floppy Controller	
0xF7201000	Audio/ISDN	
0xF7400003	Auxiliary Input/Output Register	
0xF7F00000	VME Control Register	(SunFed only)
0xF8000000	Sbus Slot 0 (25 bits)	Note 4
0xF9000000	"	"
0xFA000000	Sbus Slot 1 (25 bits)	Note 4
0xFB000000	"	"
0xFC000000	Sbus Slot 2 (25 bits)	Note 4
0xFD000000	"	"
0xFE000000	Sbus Slot 3 (25 bits)	Note 4
0xFF000000	"	"

Note 1. Same as Sun4 use.
Note 2. Sun4 has a different kind of TOD at this address. It also has an EEPROM at a different address.
Note 3. Sun4 has same function, but at a different address.
Note 4. Sun4 has no corresponding function.

Reference to a Type 1 address to which no device responds results in a timeout.

4.2.1. Onboard Devices

4.2.1.1. Keyboard/Mouse

The keyboard/mouse UART is a Z8530 chip (Zilog or AMD equivalent) accessed via byte loads and stores at the following addresses:

Address	Description
0xF0000000	Mouse Control Port
0xF0000002	Mouse Transmit (W)/Receive (R) Data Port
0xF0000004	Keyboard Control Port
0xF0000006	Keyboard Transmit (W)/Receive (R) Data Port

The Z8530 contains an array of read registers and write registers, accessed through the control port. Access to a register is done by writing the register index to the control port, and then reading or writing the register data to the control port. In addition, the UART transmit and receive data registers may be directly accessed by writing and reading, respectively, from the Transmit/Receive Data Port.

See the Z8530 data sheet for more information.

4.2.1.2. Serial Ports

The serial ports UART is also a Z8530 chip, identical to the one used for the keyboard/mouse. It is addressed as follows:

Address	Description
0xF1000000	Serial Port B Control Port
0xF1000002	Serial Port B Transmit (W)/Receive (R) Data Port
0xF1000004	Serial Port A Control Port
0xF1000006	Serial Port A Transmit (W)/Receive (R) Data Port

4.2.1.3. TOD Clock and NVRAM (EEPROM)

The Time of Day Clock is a Mostek MK48T12-15 Zeropower/Timekeeper RAM which includes 2K of RAM, the topmost 8 bytes of which are the clock. The Timekeeper contains its own battery backup, which has a worst-case storage life (oscillator off or power on) of 11 years at 70°C and a worst case consumption life (oscillator on and power off) of 2.8 years at 0°C. Unlike EEPROMs, there is no limitation on the number of times the CMOS RAM can be written, nor are special write timings required.

The Clock/NVRAM is accessed via byte, halfword, or fullword loads and stores at the following addresses:

Address	Description
0xF2000000 to 0xF20007d7	NVRAM
0xF20007d8 to 0xF20007f7	"IDPROM"
0xF20007f8	TOD Control
0xF20007f9	Seconds (00-59)
0xF20007fa	Minutes (00-59)
0xF20007fb	Hour (00-23)
0xF20007fc	Day (01-07)
0xF20007fd	Date (01-31)
0xF20007fe	Month (01-12)
0xF20007ff	Year (00-99)

Thirty-two bytes of NVRAM acts as the ID prom" of SPARCstation-1. The id_machine byte contains 0x51; 0x50 is the architecture code for Sun4C, and 0x51 indicates the SPARCstation-1 machine.

The TOD Control register should only be written with byte stores to prevent modifying the data to be read.

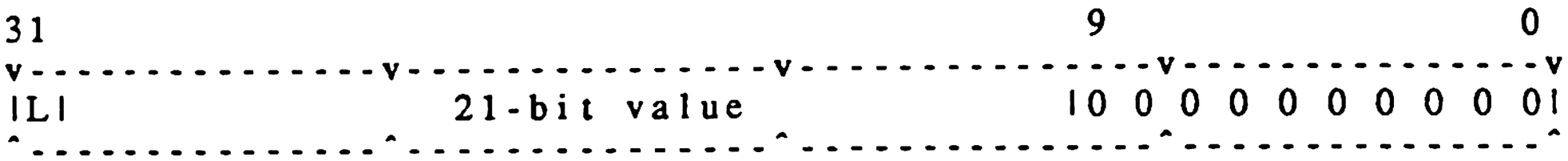
The time and date information is stored in 24 hour BCD format. For more information, including the protocol to be used to read, write, start, and stop the clock, see the MK48T12-15 data sheet.

4.2.1.4. Counter-Timer Registers

The Counter-Timer Registers are accessed via fullword loads and stores at the following addresses:

Address	Description
0xF3000000	Counter 0
0xF3000004	Limit 0
0xF3000008	Counter 1
0xF300000C	Limit 1

All registers have the following format:



L Limit Reached

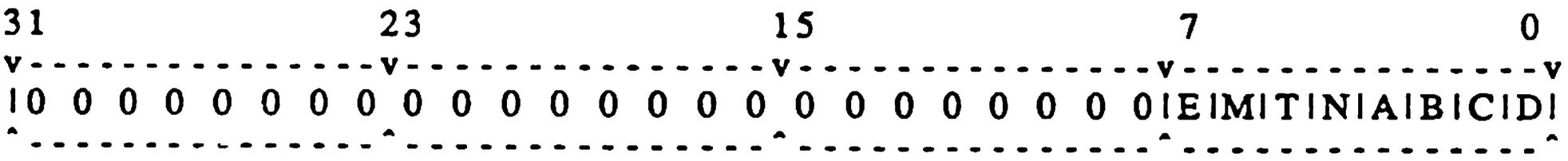
Each counter is incremented by one in bit position 10 at one microsecond intervals. When a counter reaches the value in its corresponding limit register, it is reset to "one microsecond," the limit-reached bit in both the counter and limit registers is set, and an interrupt is generated (if enabled) at level 10 for Counter 0 and level 14 for Counter 1.

The interrupt is cleared and the limit bits reset by reading the appropriate limit register. Reading the counter register does not change the state of the limit bits. Writing the limit register resets the counter register to a value equivalent to one microsecond. Except for testing purposes, the counter registers should not be written.

Setting a limit register to zero causes the corresponding counter to freerun. Interrupts will occur when the counter overflows back to zero, approximately every 2 seconds.

4.2.1.5. Memory Error Registers

SPARCstation-1 uses a single Parity Control Register. This is a fullword read/write register at location 0xF4000000 in Type 1 physical space. The format of this register is as follows:



E Parity Error. Set on any parity error.
M Multiple Errors. Set when a parity error occurs and E=1.
T Parity Test. When set, inverse parity is generated.
N Parity Check. Enables parity checking.
A Parity Error 24. Records parity error on data bits 31:24.
B Parity Error 16. Records parity error on data bits 23:16.
C Parity Error 08. Records parity error on data bits 15:8.
D Parity Error 00. Records parity error on data bits 7:0.

The bits that indicate errors (E, M, and A-D) are cleared when the register is read. All bits are cleared on reset.

Note that when a parity error occurs, the cache will have loaded itself with the data from memory anyway. This means that software must flush the cache after parity errors if it is to continue operation. On a single parity error (M=0), only the affected cache line (as determined from the old PC, the SEVAR, or

the ASEVAR, as appropriate) need be flushed. On multiple parity errors (M=1), the entire cache must be flushed.

Also note that the address in the SEVAR or ASEVAR, as appropriate, may not be the address of the word with the parity error, if the error occurred during a cache-fill operation.

4.2.1.6. Interrupt Register

The Interrupt Register is a one-byte read/write register at location 0xF5000000 in Type 1 physical space. The format of this register is as followed:

A Enable Level 14 Interrupts
C Enable Level 10 Interrupts
D Enable Level 8 Interrupts
E Software Interrupt Level 6
F Software Interrupt Level 4
G Software Interrupt Level 1
H Enable all Interrupts

Writing a zero to an Enable Level *N* Interrupt bit only masks out that interrupt, it does not clear the source. Writing a one to a software interrupt bit requests an interrupt on that level; the bit must be cleared to clear the request.

Writing a zero to the Enable All Interrupts bit will clear the Asynchronous Memory (level 15) Interrupt, as well as masking all interrupts. Of course, interrupts should be immediately re-enabled by writing a one.

On reset, all bits are cleared and all interrupts are reset.

4.2.1.7. EPROM

SPARCstation-1 has 128K bytes of EPROM containing the boot monitor beginning at location 0xF6000000 in Type 1 physical space. The EPROM is also referenced by all Supervisor Virtual addresses when the ENA_NOTBOOT bit in the System Enable Register is zero, for example at boot time. The boot code must initialize the MMU to at least map itself before setting the ENA_NOTBOOT bit to one.

Note that the EPROM does not obey the normal memory mapping rules. PA[16:0] into the EPROM always come from VA[16:0]. Although VA[29:12] are processed by the MMU to select a physical address, when bits PA[27:24] of that physical address select the EPROM then bits PA[23:12] from the MMU are ignored. This means that, for proper operation of the EPROM, it must be mapped one-for-one to contiguous virtual pages beginning on a 128K boundary.

4.2.1.8. Floppy Controller

The Floppy Disk Controller is an Intel 82072. It is accessed using byte loads and stores at the following addresses:

Address	Description
0xF7200000	Main Status (R)/Data Rate Select Register (W)
0xF7200001	FIFO Data Port (R/W)

For more information see the Intel 82072 data sheet. Note that the floppy must be selected as drive 1 (or 3, but 1 is preferred) in the command sequence sent to the controller. See also the Terminal Count and Floppy Eject bits in the "Auxiliary Input/Output Register" described below.

4.2.1.9. Audio/ISDN

The audio interface of the SPARCstation-1 is provided through the Main Audio Processor (MAP) of the AMD 79C30A Digital Subscriber Controller. The 79C30A is a highly integrated circuit which provides an ISDN 4-wire subscriber level interface, an audio processing circuit, a parallel microprocessor interface, and a serial interface. For SPARCstation-1 Audio use the microprocessor interface and the audio processing circuits are the only portions of the circuit which are used.

The interrupt from the 79C30 is attached to IRQ<13> of the MMU (which is interrupt level 13). The data bus is connected to the IO data bus. The circuit includes an oscillator circuit which uses an externally provided 12.288 MHz crystal with a tolerance of + or - 80 ppm. The oscillator is a parallel resonant circuit.

The 79C30 registers are located at a base address of 0xF7201000. The 79C30 is accessed using byte loads and stores at the following addresses:

Address	WR*	RD*	Register description
0xF7201000	0	1	Command Register (CR), write only
	1	0	Interrupt Register (IR), read only
0xF7201001	0	1	Data Register (DR), write
	1	0	Data Register (DR), read
0xF7201002	1	0	D-channel Status Register 1 (DSR1), read only
0xF7201003	1	0	D-channel Error Register (DER), read only
0xF7201004	0	1	D-channel Transmit Buffer (DCTB), write only (8-byte FIFO)
0xF7201004	1	0	D-channel Receive Buffer (DCRB), read only (8-byte FIFO)
0xF7201005	0	1	Bb channel Transmit Buffer (BBTB), write only
0xF7201005	1	0	Bb channel Receive Buffer (BBRB), read only
0xF7201006	0	1	Bc channel Transmit Buffer (BBTB), write only
0xF7201006	1	0	Bc channel Receive Buffer (BBRB), read only
0xF7201007	1	0	D-channel Status Register 2 (DSR2), read only

Note that the other registers in the 79C30, of which there are many, are indirectly accessed through the command register. Pages 2-71 through 2-77 of the 79C30A Data Sheet describe this indirect addressing.

Please refer to the 79C30A Data Sheet for full details on operation of this circuit.

4.2.1.10. Auxiliary Input/Output Register

The Auxiliary Input/Output Register is a one-byte, read-write register at location 0xF7400003 in Type 1 physical space. It has the following format:

```
 7 6 5 4 3 2 1 0
v---------------v
|1 1|D|C|S|T|E|L|
^---------------^
```

D	In	Density
C	In	Floppy Diskette Change (must be written as one)
S	Out	Floppy Drive Select
T	Out	TC (Floppy controller Terminal Count input)
E	Out	Floppy Eject
L	Out	LED (1=on, 0=off)

All bits are set to one on reset.

Bit 5 (Density) is a signal from the drive indicating the density of the diskette inserted. A 1 indicates high density, a 0 indicates low density. This signal is meaningful only if the floppy drive is capable of sensing the "density" hole in the diskette. The Sony drives do not generate this signal; for them, software must through trial and error determine the density of the inserted diskette. This can be done by initializing the controller with parameters for a given density and attempting to read the diskette; if the wrong parameters were chosen read errors will occur. Note that the density of an unformatted floppy cannot be determined through this method; the floppy format software must have a user option to set the density to be

used. (If the user selects the wrong density, the floppy will be unusable, but the user will quickly discover this mistake.)

Bit 4 (Floppy Diskette Change) is an input bit that signifies when a diskette has been removed from the drive. This bit must always be written as one in order for it to work. It reads as one when the drive is selected and there is no diskette in the drive. It reads as zero if the drive is not selected or if a diskette is present in the drive. The Sony drives reset the bit when they receive a step pulse from the controller; i.e., when the software issues a "Seek" command. Other vendor drives require a separate Diskette Change Reset signal; a bit will need to be provided for this function in the Auxiliary Input/Output Register if a non-Sony drive is used on a SPARCstation.

Bit 3 (Floppy Drive Select) is connected to the floppy drive select pin. It is used in conjunction with all floppy operations, whether through the Floppy Disk Controller registers or the bits in the Auxiliary I/O Register. A one selects the floppy drive; a zero de-selects it.

Bit 2 (TC) is connected to the Terminal Count input pin of the floppy controller. It is used to signal the floppy controller (which is designed to be connected to a DMA controller, even though in SPARCstation-1 it is not) that all the data for a given operation has been transferred. This is done by writing a 1 to this bit, delaying for a specific amount of time, and then writing a 0 to it. (The specific amount of time depends upon the data rate and can be found in the Intel 82072 data sheet.)

Bit 1 (Floppy Eject) is connected to the floppy drive eject mechanism. To eject a floppy, set bit 3 (Floppy Drive Select), wait 2.0 microseconds, set bit 1, hold it set for at least 2.0 microseconds, then reset both it and bit 3 to zero.

Bit 0 (LED) controls the LED on the front panel.

Unused bit positions should be written with ones when writing to the register. This will allow them to be used for input signals if this becomes necessary.

4.2.2. Sbus Devices

Unlike previous busses, the Sbus is geographically addressed. PA26:25 select which of four Sbus "slots" is being referenced. A board plugged into an Sbus slot has PA24:0, or 25 bits or 32 Mbytes of address space addressability to divide up among the devices contained on that board. A Forth program beginning at offset 0 of the slot describes the devices on that board to the system. The details of the Forth specification are described in Sun Forth User's Guide.

Slot 0 is not a physical slot. Rather, it refers to the onboard DMA, SCSI, and Ethernet controllers which, for convenience, are viewed as being plugged into Slot 0.

Slots 1, 2, and 3 are physical slots into which the user may plug boards containing devices. Slots 1 and 2 have DVMA-master capability; slot 3 is a slave-only slot and does not support boards that operate as DVMA masters. The board containing the video subsystem (video control registers, RAMDAC, and frame buffer) is usually, but need not be, plugged into Slot 3.

If no device responds to a particular Sbus address, a bus timeout will occur.

The following table summarizes the devices:

PA26:25	Device
0 0	Onboard DMA, SCSI, and Ethernet controllers
0 1	Sbus Slot 1
1 0	Sbus Slot 2
1 1	Sbus Slot 3 (usually video subsystem)

4.2.2.1. DMA, SCSI, and Ethernet Devices

The following table describes the offsets to the onboard DMA, SCSI, and Ethernet devices, relative to the beginning of Sbus "Slot 0" (base physical address 0xF8000000 in Type 1 space).

Offset	Description
0x000000	ID (4 bytes, 0xFE810101)

Offset	Description
0x400000	DMA Registers
0x800000	SCSI Registers
0xC00000	Ethernet Registers

4.2.2.1.1. DMA Registers

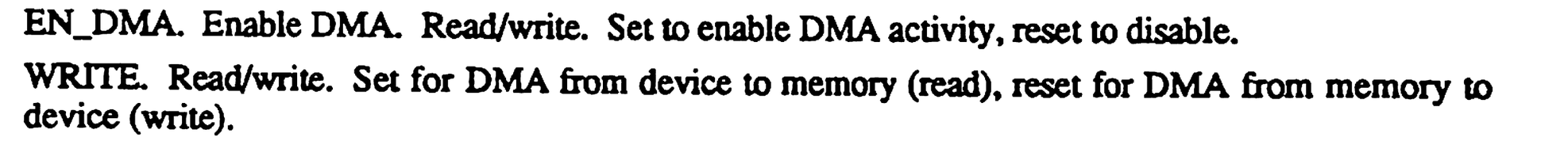
The DMA registers are accessed via fullword loads and stores to the following offsets (the addresses in this table do not include the slot base address, which must be added to the device offset):

Address	Description
0x400000	DMA Control/Status Register
0x400004	DMA Address Register
0x400008	DMA Byte Count
0x40000C	Diagnostic Register

The DMA registers are used when programming SCSI operations. Other than the ILACC bit in the DMA Control/Status Register, they are not used when programming Ethernet operations.

4.2.2.1.1.1. DMA Control/Status Register

The DMA Control/Status Register has the following format:



DEV_ID
DEV_ID. Device ID. Read-only. (0b1000 in this implementation.)

L L. ILACC. When 0, the Ethernet/DMA interface is configured to use the Lance Ethernet controller. When 1, the interface is configured to use ILACC, "the new Ethernet chip from AMDpq (Cliff Buckley).

T TC. Terminal Count. Read-only. Byte counter has expired. This bit is cleared by setting the Flush bit (bit 5).

C EN_CNT. Enable Count. Read/write. Enables the DMA Byte Count Register. (Not used in normal SPARCstation-1 operation.)

ADR BYTE_ADDR. Read-only. Next byte number to be accessed.

P REQ_PEND. Request pending. Read-only. Set when the DMA interface is active. RESET and FLUSH must not be asserted if REQ_PEND is one.

N EN_DMA. Enable DMA. Read/write. Set to enable DMA activity, reset to disable.

W WRITE. Read/write. Set for DMA from device to memory (read), reset for DMA from memory to device (write).

R RESET. Read/write. When set, acts as a hardware reset. ERR_PEND, PACK_CNT, INT_EN, FLUSH, DRAIN, WRITE, EN_DMA, REQ_PEND, EN_CNT, and TC are all set to zero. RESET remains at 1, and must be set back to 0 by software to resume operation.

D DRAIN. Read/write. Set to force remaining pack register bytes to be drained to memory. Clears itself.

F FLUSH. Write-only. Set to force PACK_CNT and ERR_PEND to zero. Also clears TC and the interrupt TC=1 causes. Always reads as zero.

I INT_EN. Interrupt enable. Read/write. Set to enable interrupts.

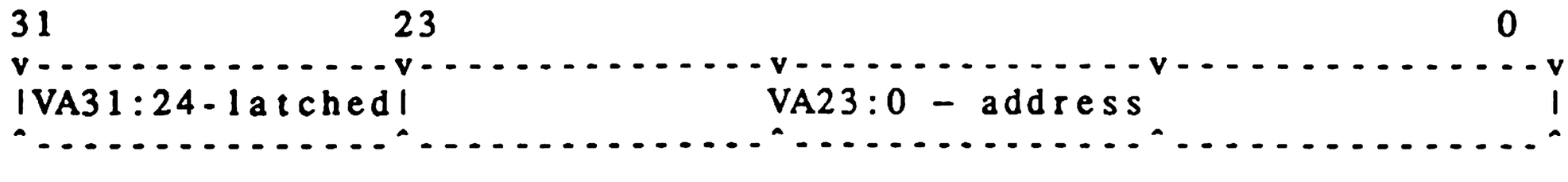
PCK PACK_CNT. Pack Count. Read-only. Number of bytes in Pack Register.

E ERR_PEND. Error Pending. Read-only. Set when a memory exception occurs. Reset by setting FLUSH. DMA activity stops until reset.

J INT_PEND. Interrupt Pending. Read-only. Set when TC=1 or when external device raises an interrupt. Cleared when read (if TC=1 is the cause) or by servicing the external device (if that is the cause).

4.2.2.1.1.2. DMA Address Register

The DMA Address Register has the following format:



The high byte is latched by the hardware and indicates which 16 Mbyte region of Virtual Memory is accessed. (The MMU recognizes a DMA virtual address and forces Context 0 to be selected.) The low-order 3 bytes contain the address of the byte to be transferred. Rollover is only through the low-order 24 bits.

4.2.2.1.1.3. DMA Byte Count

The DMA Byte Count Register has the following format:

This register is only used when EN_CNT is on in the DMA Control/Status Register, and so is not used in normal SPARCstation-1 operation. The high byte is unused and will always read back as zero. The low order bytes contain the number of bytes to be transferred, and counts down to zero. When zero is reached, TC, and thus INT_PEND, are set to one. Further DMA transfers cannot take place until a new value is loaded into the Byte Count Regiser.

4.2.2.1.1.4. Diagnostic Register

The format of the Diagnostic Register is not available.

4.2.2.1.2. SCSI Registers

The SCSI registers are accessed via byte loads and stores to the following offsets (the addresses in this table do not include the slot base address, which must be added to the device offset):

Address	Description
0x800000	Transfer Count Low
0x800004	Transfer Count High
0x800008	FIFO Data
0x80000C	Command
0x800010	Status/Bus ID
0x800014	Interrupt/Status Timeout
0x800018	Sequential step/Synchronization transfer period
0x80001C	FIFO flags/Synchronization offset
0x800020	Configuration
0x800024	Clock Conversion Factor (write only)
0x800028	ESP TEST (chip test use only)
0x80002C	ESP II Configuration-2

Note that byte accesses must be performed even though the addresses are all fullword-aligned.

Since the SCSI controller uses the DMA controller to perform the actual transfer of data to and from memory, the two devices must be programmed together. One possible algorithm is as follows:

scsi_start()

```
{
        /* start an operation on the SCSI */
        lock data pages into contiguous virtual memory;
        DMA_address_register = starting virtual address;
        setup SCSI registers (except for "go");
        DMA_control_status_register = (EN_DMA | INT_EN | (other bits));
        start SCSI;
        /* The SCSI will interrupt us when it is done. */
}

scsi_interrupt()
{
        /* must drain DMA on a read from disk/write to memory */
        if (last operation == READ) {
                DMA_control_status_register = (DRAIN);
        }
}
```

For a detailed description of the SCSI registers, see the NCR 53C90 Data Sheet.

4.2.2.1.3. Ethernet Registers

The Ethernet registers are accessed via halfword loads and stores to the following offsets (the addresses in this table do not include the slot base address, which must be added to the device offset):

Address	Description
0xC00000	Register Data Port (RDP)
0xC00002	Register Address Port (RAP)

For a detailed description of the Ethernet registers, see the AMD Am7990 Data Sheet.

4.2.2.2. Video Subsystem

The following table describes the offsets to the devices located on the Video Subsystem Board. This board is usually plugged into Sbus "Slot 3" (base physical address 0xFE000000 in Type 1 space).

Offset	Description
0x000000	ID (4 bytes, 0xFE010101)
0x400000	Video and DAC Registers
0x800000	Frame Buffer

4.2.2.2.1. Video and DAC Registers

The Video and DAC registers are accessed via byte loads and stores to the following offsets (the addresses in this table do not include the slot base address, which must be added to the device offset):

Address	Description
0x400000	Video Control Register
0x400001	Video Status Register
0x400002	HBS (Horizontal Blank Set)
0x400003	HBC (Horizontal Blank Clear)
0x400004	HSS (Horizontal Sync Set)
0x400005	HSC0 (Horizontal Sync Clear, !VS)
0x400006	HSC1 (Horizontal Sync Clear, VS)
0x400007	VBSH (Vertical Blank Set High Byte)
0x400008	VBSL (Vertical Blank Set Low Byte)
0x400009	VBC (Vertical Blank Clear)
0x40000A	VSS (Vertical Sync Start)

Address	Description
0x40000B	VSC (Vertical Sync Clear)
0x400010	DAC Address Register
0x400014	DAC Color Palette Register Port
0x400018	DAC Control Register Port
0x40001C	DAC Overlay Palette Register Port

See the S-4 Video data sheet for a detailed description of the Video Registers, and the Brooktree Bt458/451 data sheet for a detailed description of the DAC Registers. Note that setting incorrect values into the registers can damage the attached monitor.

Note that the DAC registers are 8-bits wide even though they are aligned on fullword boundaries. Fullword accesses can be used to quickly read or write one or more palette entries, by storing the index of the first palette to be accessed in the address register and then doing fullword accesses to the appropriate palette port. The data must be packed into bytes in the order "RGBRGBRGBRGB"; in other words, 3 fullwords will hold 4 palette entries. Palette entries are only stored when the Blue value is written; partial update of a palette is not possible.

4.2.2.2.2. Frame Buffer

The frame buffer is a megabyte of RAM occupying offsets from 0x800000 to 0x8FFFFF. Each byte corresponds to one pixel. Accesses may be by bytes, by halfwords, or by fullwords.

If the frame buffer is only half-populated, then only the lower four bits of each byte will be significant. As the upper four bits will be (weakly) pulled up with resistors, only the upper 16 color map entries (entries 240 through 255) in the DAC will be usable. Software can detect this case by writing, then reading, the frame buffer. If the upper four bits always read back as ones, independent of the data written, then the frame buffer is half-populated. **(This is grody — Ed.)**

5. Interrupt Levels

The following table describes the interrupt levels defined by the Sun4 Architecture and the SPARCstation-1 implementation.

Level	Sun4 Use	SPARCstation-1 Use
15	Memory Error	Asynchronous Memory Error
14	Clock	Counter 1
13	VMEbus level 7	Audio
12	Keyboard, Mouse, Serial Ports	Same
11	VMEbus level 6	Floppy
10	Clock	Counter 0
9	VMEbus level 5	Sbus IRQ7
8	Video	Sbus IRQ6
7	VMEbus level 4	Video, Sbus IRQ5
6	Ethernet, Software request 6	Software request 6
5	VMEbus level 3	Ethernet, Sbus IRQ4
4	SCSI, Software request 4	Software request 4
3	VMEbus level 2	SCSI, DMA, Sbus IRQ3
2	VMEbus level 1	Sbus IRQ2
1	Software request 1	Same, plus Sbus IRQ1

6. Resets

Although there is only one type of reset in SPARCstation-1 (a reset of the entire machine that causes system registers to be restored to a known state), there are three ways to effect a reset:

(1) Power-on. A power-on reset (POR) occurs when power is initially applied to SPARCstation-1.

(2) Watchdog. A watchdog reset occurs when the IU signals an error condition. This can occur, for example, if the IU attempts to take a trap when traps are disabled.

(3) Software. Software can initiate a reset by writing a one to the ENA_RESET bit of the System Enable Register.

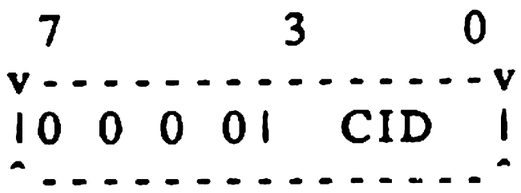
The SE_WATCHDOG bit in the Synchronous Error Register is set to one on watchdog-initiated resets, and set to zero for all other resets.

7. Contexts, Caching, and the MMU

This section describes the interaction of the context register, the cache, and the MMU from the programmers perspective.

7.1. Context Register (ASI=2, A=0x30000000, byte access only)

The Context Register has the following format:



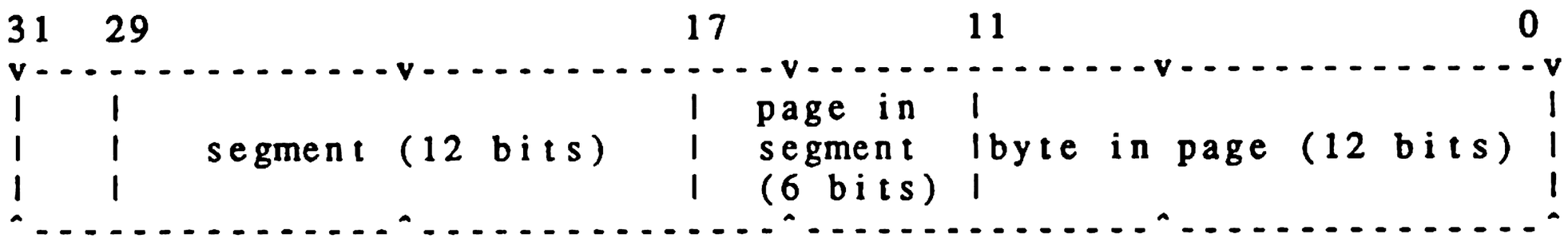
Note that although the CID is four bits wide, only the low-order 3 bits (CID2:0) are actually used. CID3 is ignored.

The context register selects one of 8 contexts for translating User Mode addresses. It exists in both the Cache and the MMU.

Programming note: A byte store (STBA) into (ASI=2, A31:28=0x3) writes both the MMU and Cache Context Registers. A byte load (LDUBA, LDSBA) from (ASI=2, A31:28=0x3, A0=0) reads the MMU's Context Register, and a byte load from (ASI=2, A31:28=0x3, A0=1) reads the Cache's Context Register. The ability to read each register separately is provided for diagnostic purposes; they should always contain the same value and standard software will usually just read the MMU's Context Register.

7.2. MMU decoding of Virtual Addresses

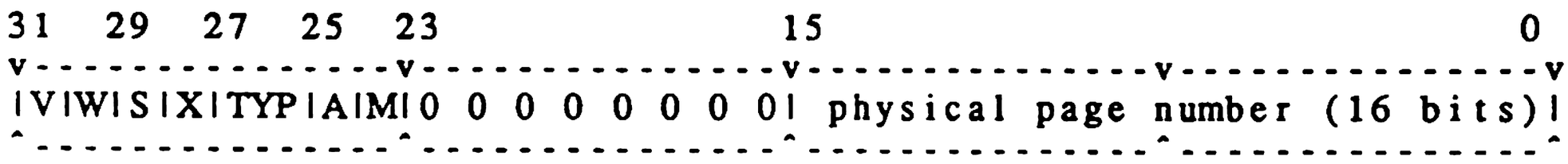
From the MMU's standpoint, a virtual address has the following format:



Note: VA31:29 must all be the same (all 0 or all 1). An SE_INVALID error results otherwise.

CID2:0 is concatenated with VA29:18 to select one of 32K segment map entries. (One can view the segment map as consisting of 8 contexts, each context containing 4K segments.) The segment map entry is 8 bits wide, although only the lower 7 bits are used, and points to a Page Map Entry Group (PMEG):

PMEG6:0 is concatenated with VA17:12 to select one of 8K Page Map Entries (PME). (One can view the page map as consisting of 128 PMEGs, each PMEG containing 64 pages.) The PME is 32 bits wide, organized as follows:



V 1=entry is valid

W 1=write access allowed
S 1=Supervisor mode access only
X 1=don't cache this page
TYP 0=Main Memory; 1=Sbus and I/O space; 2,3=reserved for VMEbus
A 1=page has been accessed
M 1=page had been modified

PME15:0 is concatenated with VA11:0 to form a 28-bit physical address whose interpretation depends upon the type field.

Programming Notes:

(1) A page is 4K bytes. A segment is 64 pages or 256K bytes. A context contains 4K segments or 1G byte. This last is divided into two address ranges of 512M bytes each, from 0x00000000-0x1fffffff and from 0xe0000000-0xffffffff.

(2) Unlike architectures used by other vendors, in this architecture there is no way to explicitly mark a segment as invalid. However, the operating system can reserve one PMEG and mark all of its PMEs invalid, and then point invalid segments at this PMEG. SunOS has traditionally used the last PMEG for this purpose, but this may be subject to change.

(3) Because the cache ignores the context register when resolving accesses to supervisor-mode-only pages, the kernel segments should be identical in each context. This can be accomplished by repeating the same PMEG in the appropriate segment map entries.

(4) A context is selected by performing a byte store into the Context Register (ASI=2, A31:28=0x3).

A segment map is initialized by selecting a context, and then performing byte stores into (ASI=3, A29:18=0x0 to 0xfff). (Half and fullword stores will work but are not recommended.)

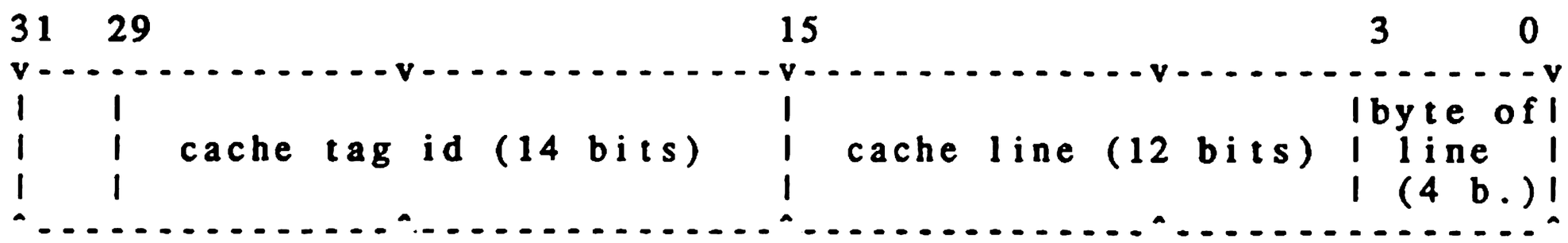
A PMEG is initialized by selecting a context, and then performing fullword stores into (ASI=4, A29:18=desired segment, A17:12=0x0 to 0x3f).

(5) The hardware does not insure consistency between the cache and the MMU. The operating system software must flush the cache appropriately before updating the MMU. Before changing the mapping of a context, a Flush Cache (Context) operation must be performed. Before changing the mapping of segment, a Flush Cache (Segment) operation must be performed. Before changing the mapping of a page, a Flush Cache (Page) operation must be performed. These operations are described in the Cache section, below. Also note that these are not the only circumstances when flushing the cache is necessary.

7.3. Cache decoding of Virtual Addresses

To improve performance, SPARCstation-1 contains a 64K byte virtual address cache, consisting of 4K lines of 16 bytes each. The cache is one-way set associative, with each virtual address mapping to one and only one possible cache line. There is a 4 byte cache tag associated with each data line.

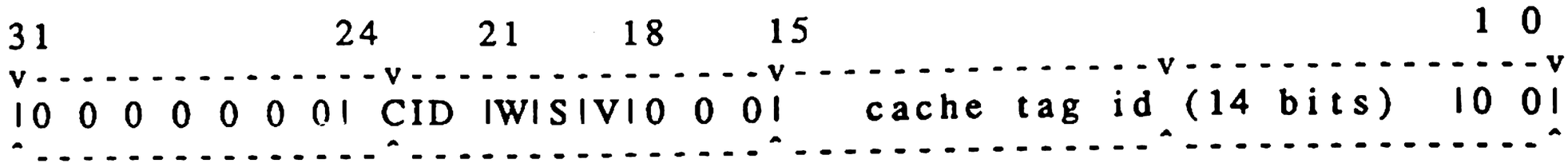
From the Cache's standpoint, a virtual address has the following format:



Note: VA31:29 must all be the same (all 0 or all 1). An SE_INVALID error occurs otherwise.

VA15:4 selects one of 4K cache lines. If the cache tag id matches (and, for non-supervisor-mode-only pages, the context ID), then a cache hit occurs. VA3:2 selects the desired word from the cache line.

A cache tag has the following format:



CID Cache Tag Context (copied from Cache Context Register when cache line is filled.) Note that only CID2:0 are present.

W 1=write access allowed (copied from MMU when cache line is filled.)

S 1=Supervisor mode access only (copied from MMU when cache line is filled.)

V 1=entry is valid

Programming Notes:

(1) The cache tags must be initialized by software before the cache is enabled, by clearing the valid bit in the cache tag of each cache line. It is sufficient to do fullword stores of zero into (ASI=2, A31:28=0x8, A15:4=0x0 to 0xfff).

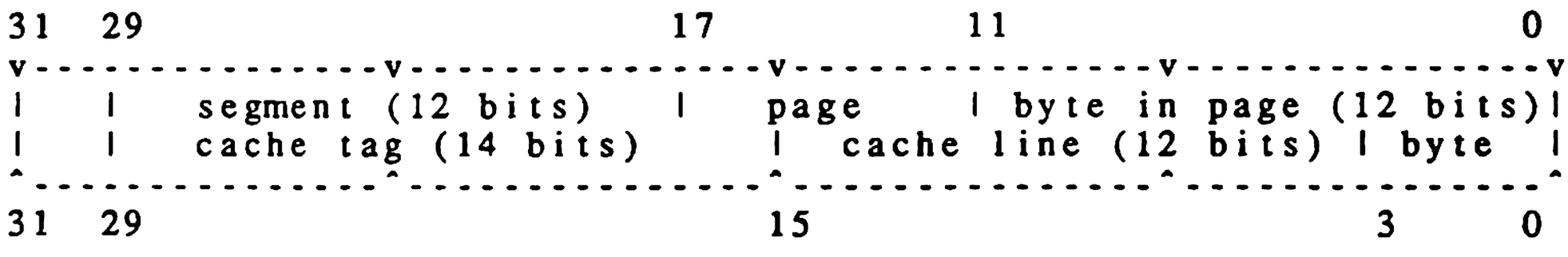
(2) To flush all references to a context from the cache, a Flush Cache (Context) operation must be performed by selecting the appropriate context (by performing a byte store into the Context Register, (ASI=2, A31:28=0x3)) and doing fullword stores of zero into (ASI=0xe, A15:4=0x0 to 0xfff).

(3) To flush all references to a segment from the cache, a Flush Cache (Segment) operation must be performed by selecting the appropriate context and doing fullword stores of zero into (ASI=0xc, A29:18=desired segment, A15:A4=0x0 to 0xfff). A17:16 are ignored for this operation.

(4) To flush all references to a page from the cache, a Flush Cache (Page) operation must be performed by selecting the appropriate context and doing fullword stores of zero into (ASI=0xd, A29:12=desired page, A11:4=0x0 to 0xff).

7.4. Aliasing

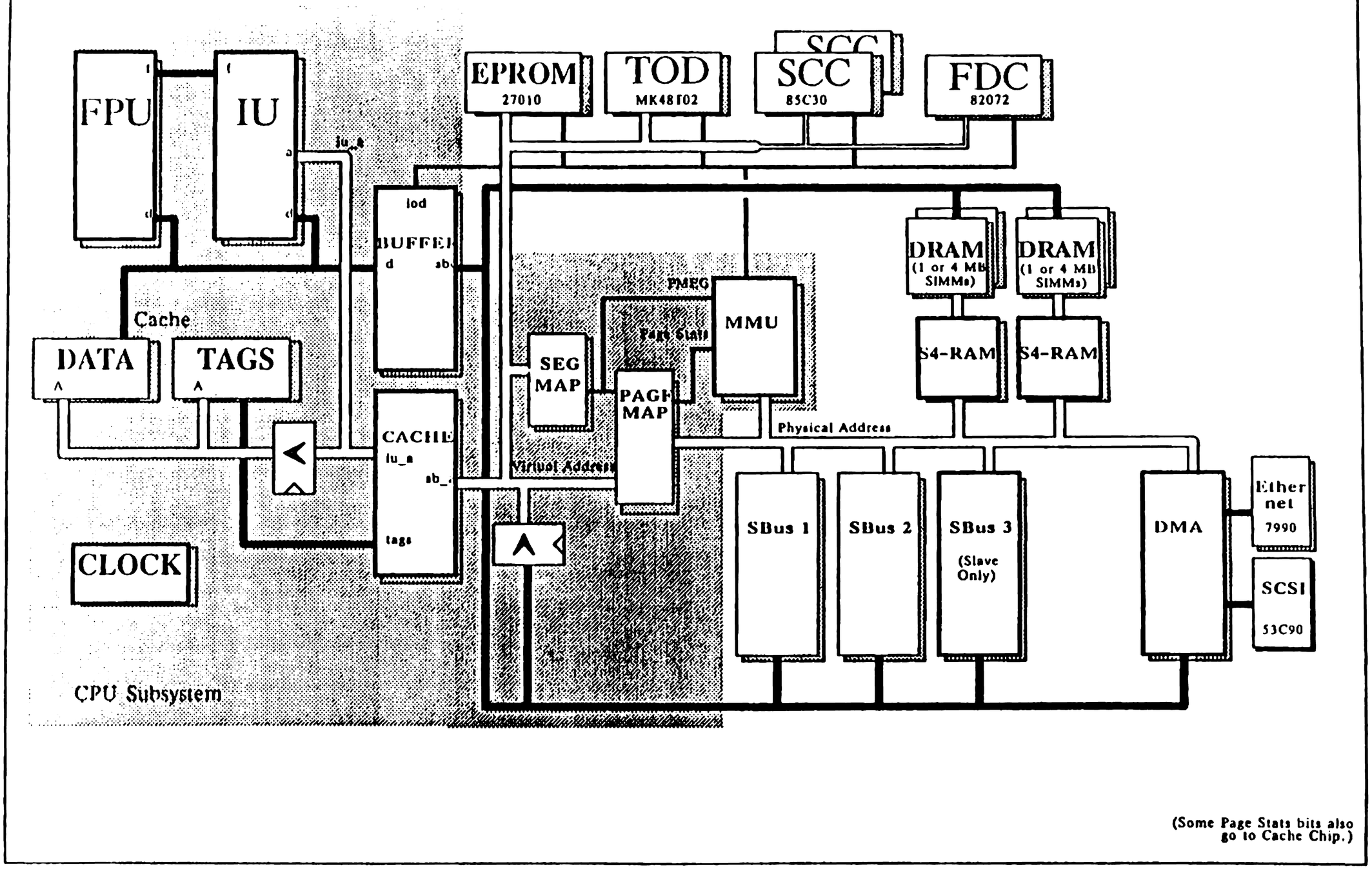
Because the cache is bigger than a page, a physical page that is mapped by two (or more) distinct virtual addresses could result in data from the same physical address appearing in two (or more) cache lines:

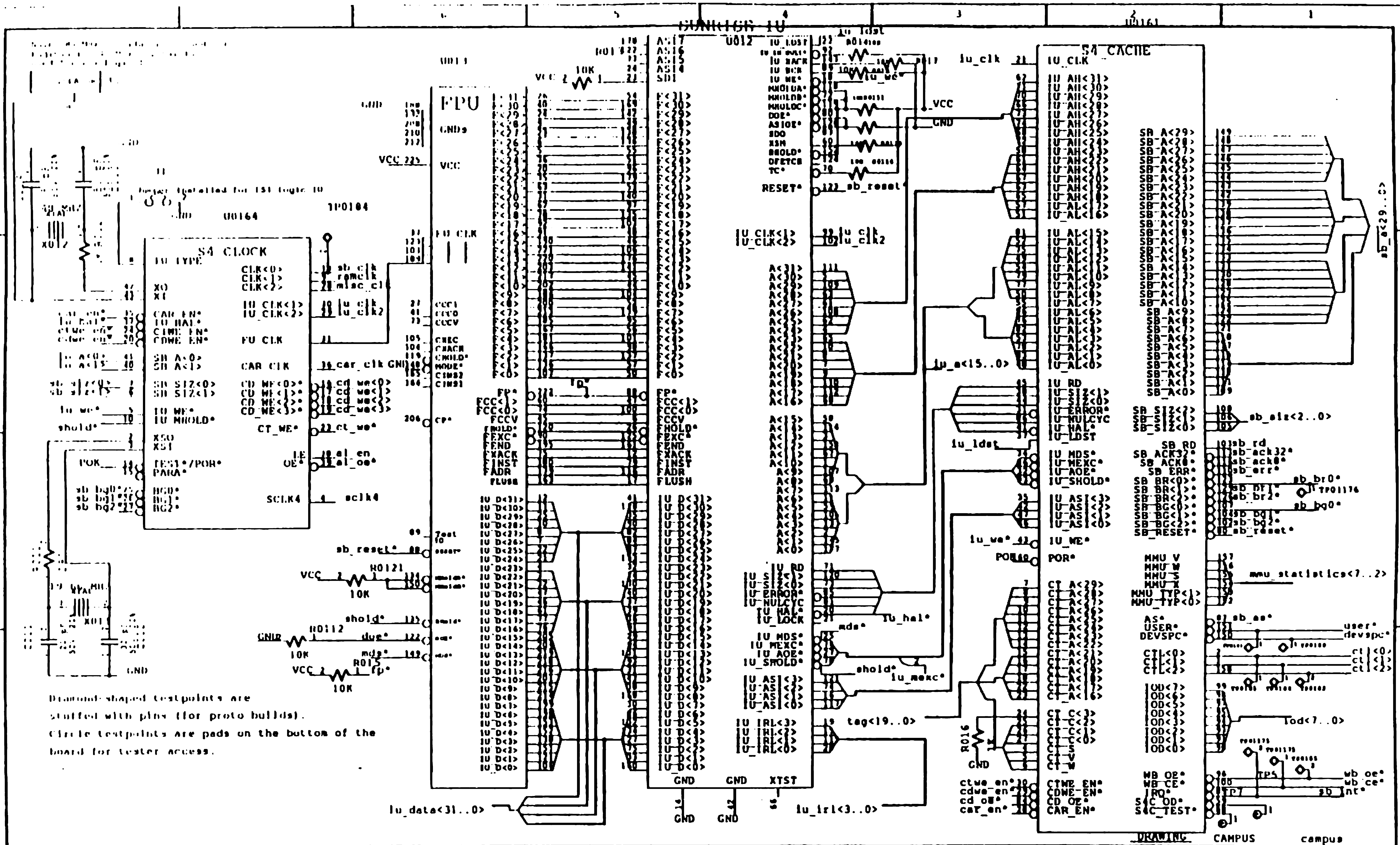


This situation cannot be detected by the hardware and must be avoided by the software. There are two methods that may be used:

(1) All the virtual addresses for an aliased page must be identical in bits A15:12. That is, the virtual addresses must be congruent modulo 64K (the cache size). This will result in the same cache line being used for the different virtual addresses that map to the same physical address. This is the preferred method. (Note that the hardware doesn't know that the different virtual addresses map to the same physical address, and alternate use of the different virtual addresses will result in invalidating and then refilling the cache line from the same physical address. Also, the hardware automatically invalidates a cache line when a cache miss occurs on a write operation. This insures the consistency of the cache with memory when aliasing via this method occurs.)

(2) Each PME that points to the aliased physical page must have the "Don't Cache" bit (PME28) set. This method must be used if the previous method cannot.


FPU
IU
EPROM
27010
TOD
MK48T02
SCC
85C30
FDC
82072
DRAM
(1 or 4 MB
SIMMs)
DRAM
(1 or 4 MB
SIMMs)
S4-RAM
S4-RAM
MMU
SEG
MAP
PAGE
MAP
Physical Address
Virtual Address
Cache
DATA
TAGS
CACHE
tags
CLOCK
CPU Subsystem
SBus 1
SBus 2
SBus 3
(Slave
Only)
DMA
Ether
net
7990
SCSI
53C90
(Some Page Stats bits also
go to Cache Chip.)



Diamond-shaped testpoints are stuffed with pins (for proto builds).
Circle testpoints are pads on the bottom of the board for tester access.

sun microsystems

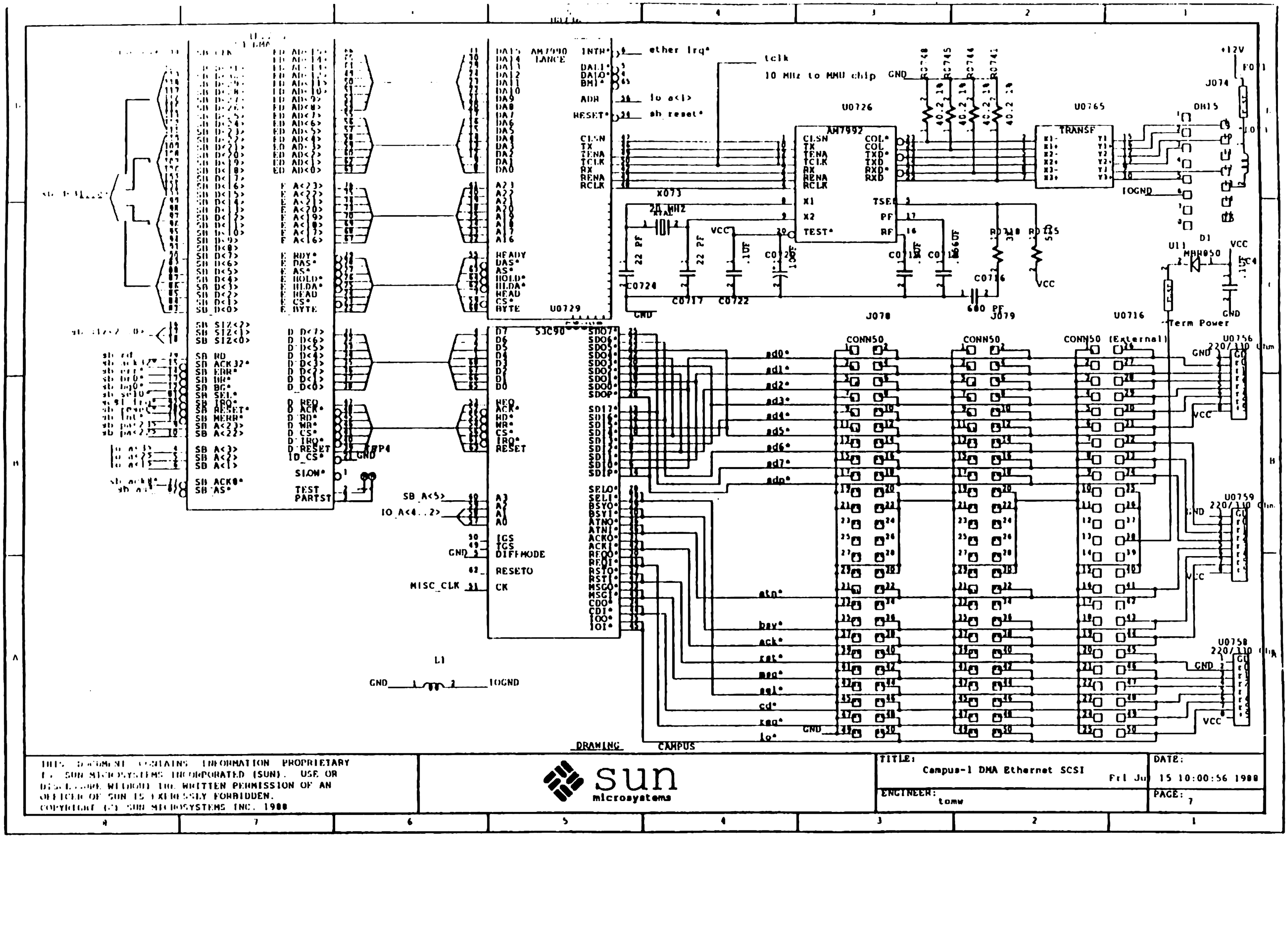
DRAWING: CAMPUS campus

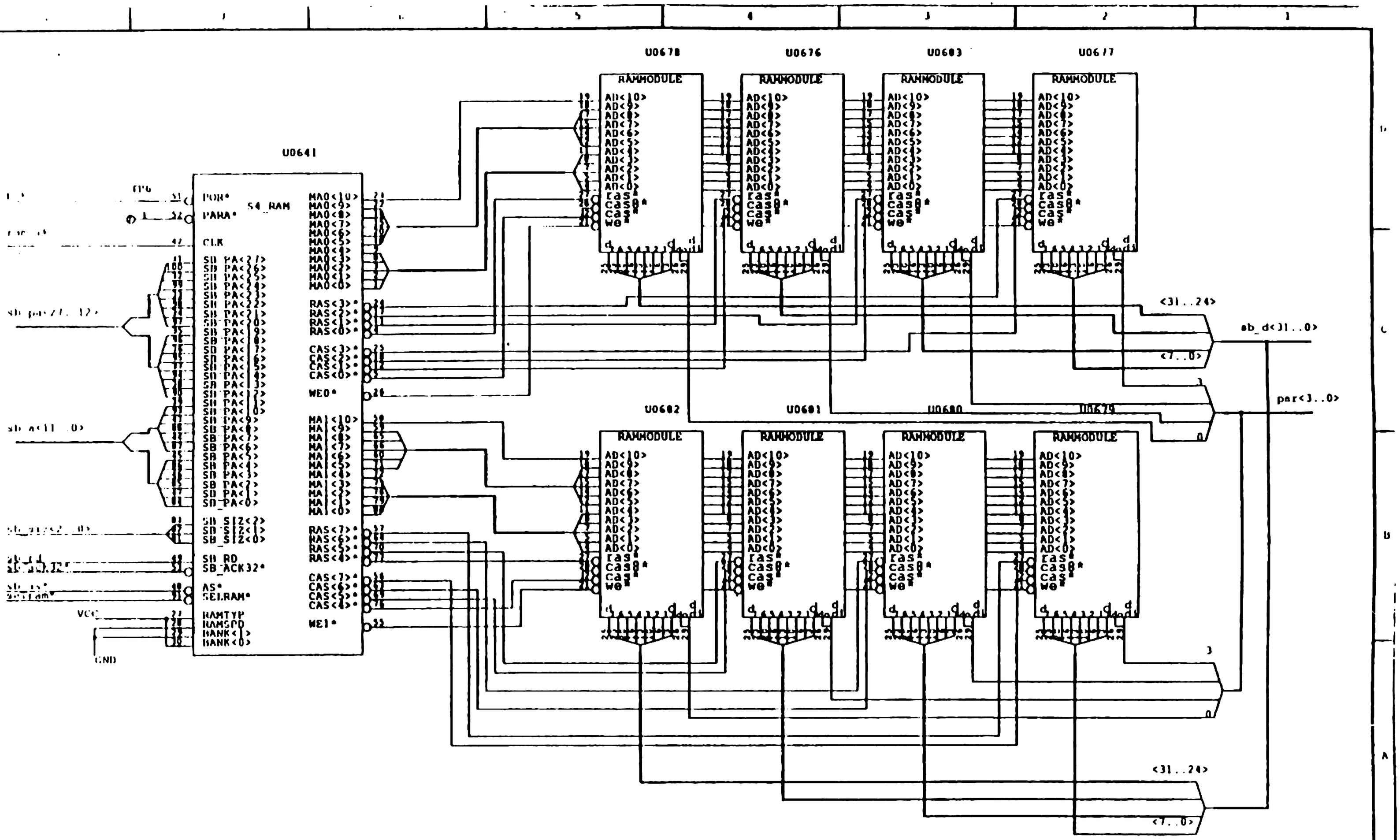
TITLE: Campus-1 schematic IU, FPU, S4-Cache, io_a buffers

DATE: Fri Jul 15 09:57:11 1988

ENGINEER: tomw

PAGE: 1


AM7990 LANCE
U0729
53C90
U0726
AM7992
U0765
TRANSF
U0716
MMH050
CONN50
CONN50
CONN50 (External)
Term Power
U0756
U0759
U0758
20 MHz
10 MHz to MMU chip
+12V
VCC
GND
TOGND
DIFFMODE
TEST
PARTST
TGS
CAMPUS
DRAWING
Campus-1 DMA Ethernet SCSI
Fri Jul 15 10:00:56 1988
tomw
7



DRAWING CAMPUS

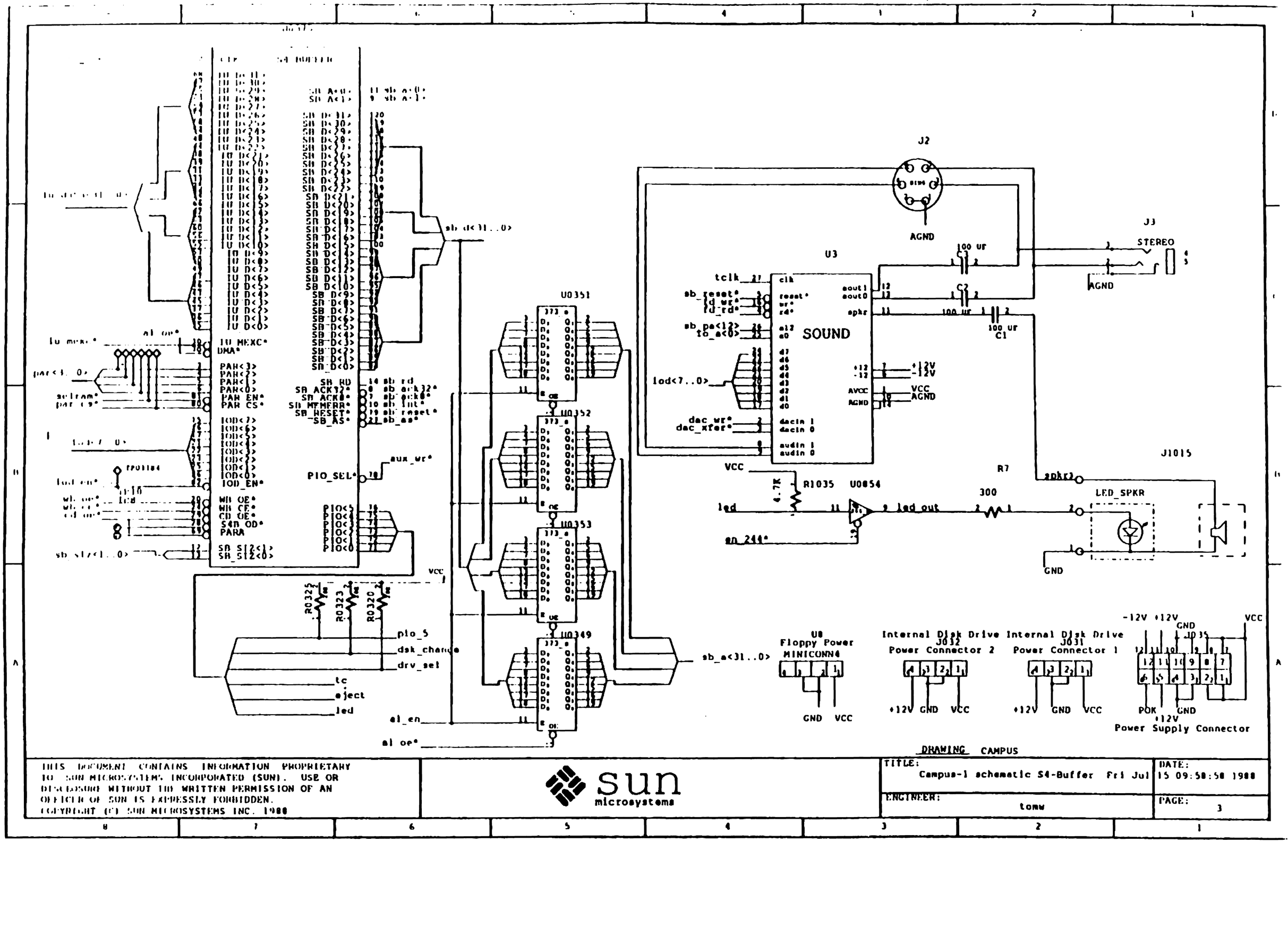
sun microsystems

TITLE: Campus-1 schematic, S4-RAM and SIMMs

ENGINEER: tomw

DATE: Fri Jul 15 10:00:19 1988

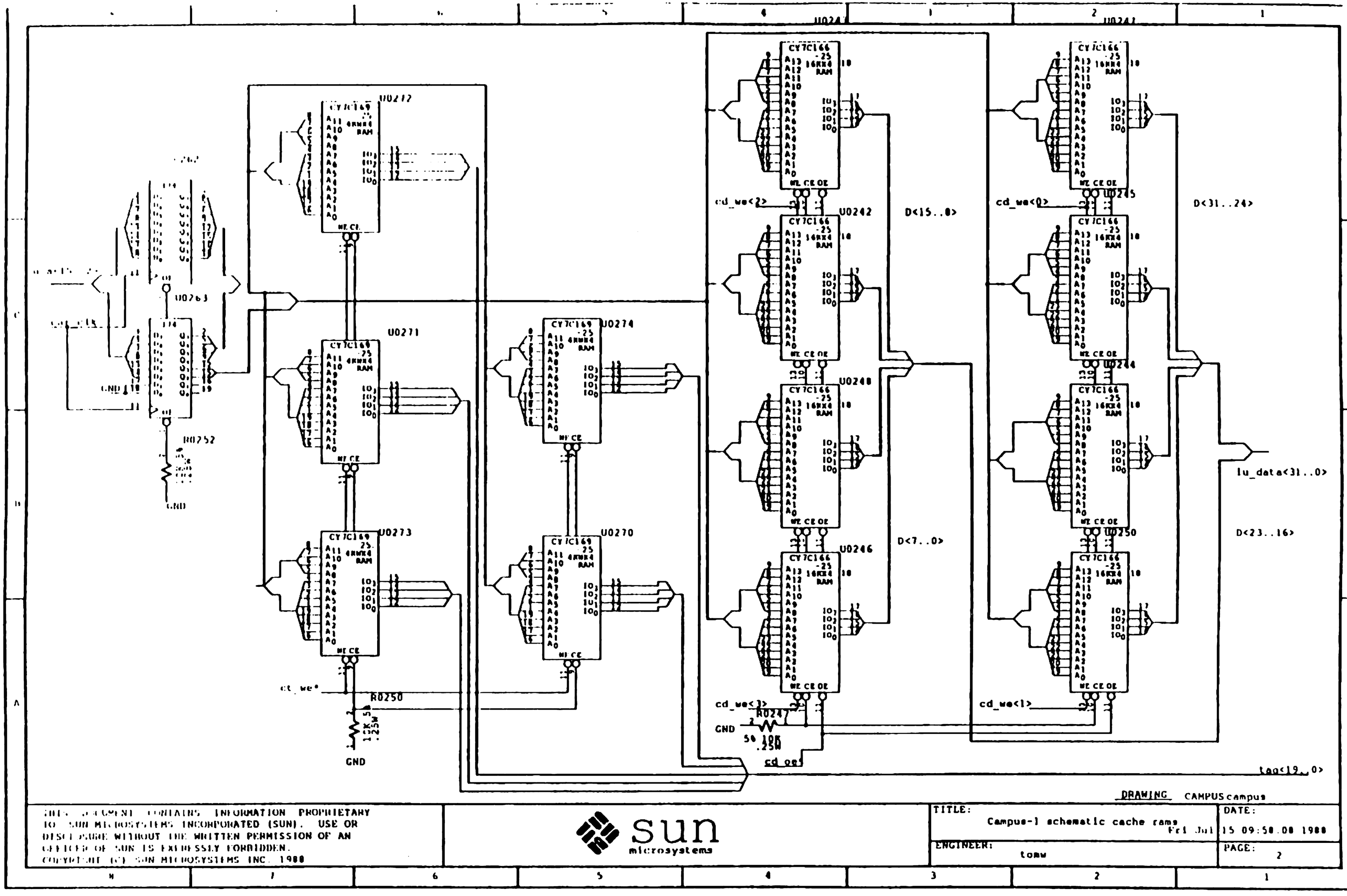
PAGE: 6



sun microsystems

DRAWING CAMPUS

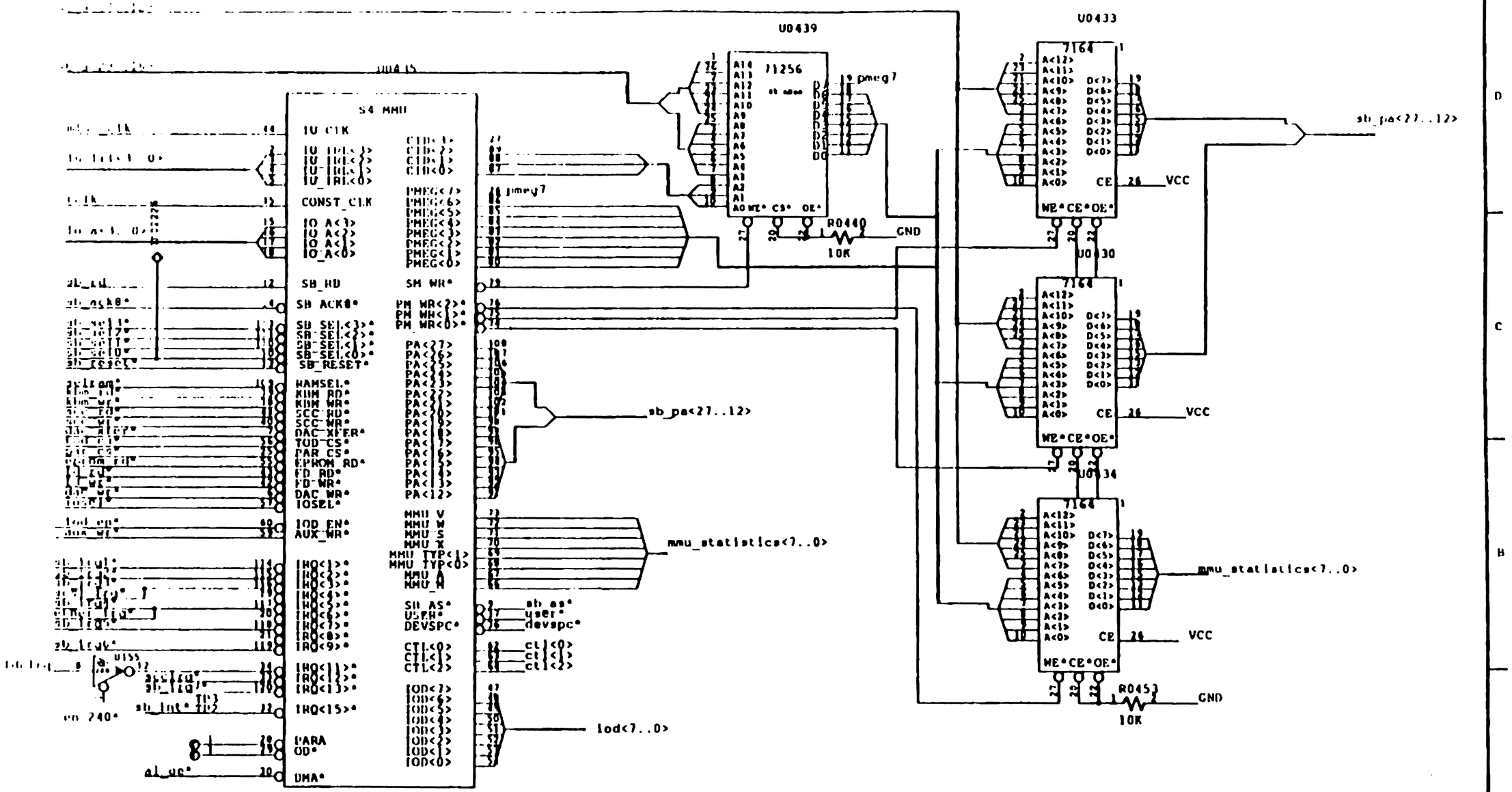
TITLE: Campus-1 schematic S4-Buffer Fri Jul	DATE: 15 09:58:58 1988
ENGINEER: tomw	PAGE: 3



DRAWING CAMPUS campus

sun microsystems

TITLE: Campus-1 schematic cache rams	DATE: Fri Jul 15 09:58:08 1988
ENGINEER: tomw	PAGE: 2



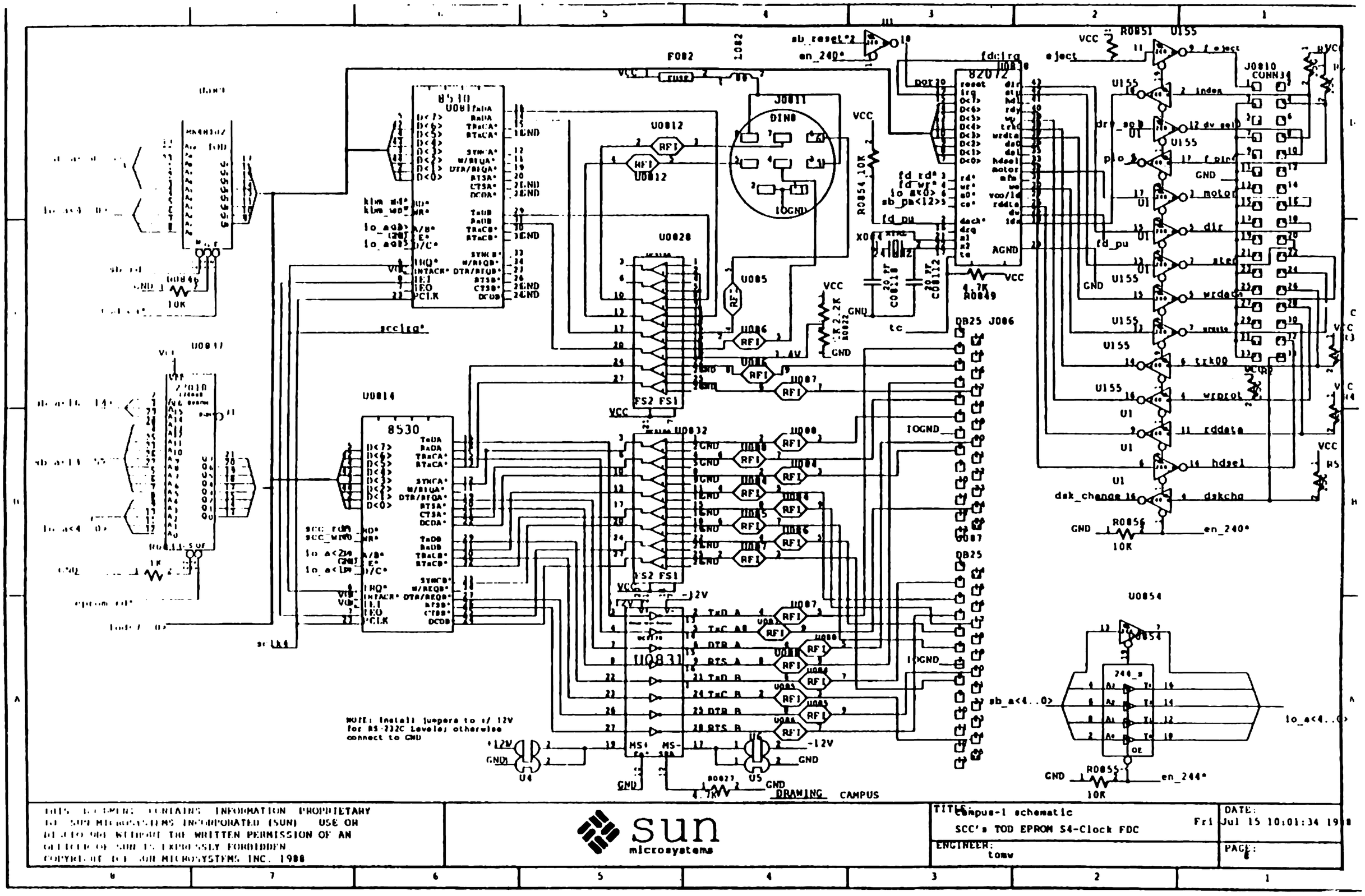
DRAWING CAMPUS

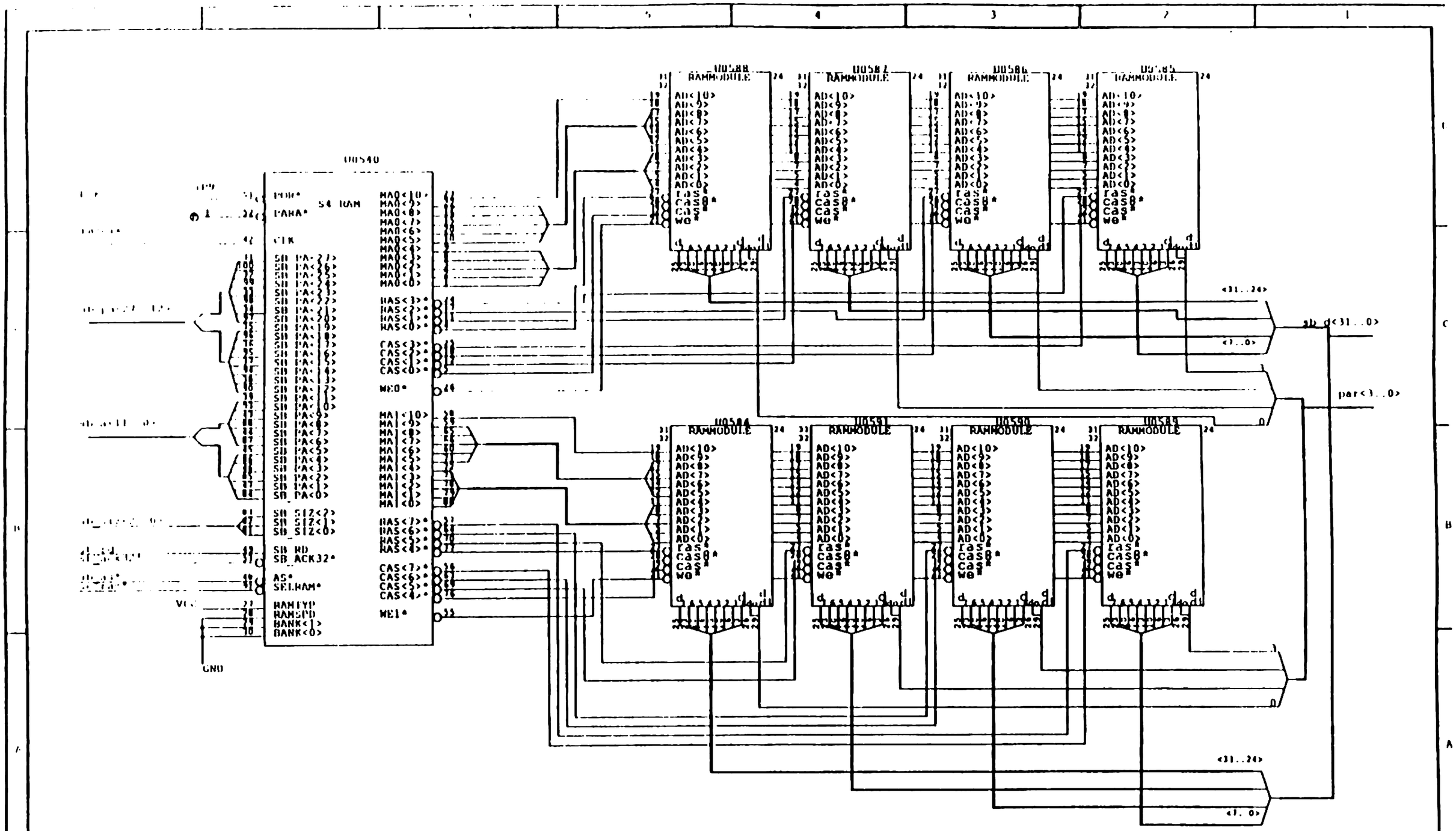
TITLE: Campus-1 schematic S4-MMU and RAMs Fri Jul

DATE: 15 09:59:35 1988

ENGINEER: tomw

PAGE: 4


8530
8510
8272
RF1
FS2 FS1
DB25 J006
DB25
DIN8
sb_reset*2
en_240*
eject
index
dv sel0
f_eject
f_dir
motor
dir
step
wrdata
wrgate
trk00
wrprot
rddata
hdsel
dskcha
dsk_change
en_244*
sb_a<4..0>
io_a<4..0>
fd_rd*
fd_wr*
fd_pu
tc
NOTE: Install jumpers to +/-12V for RS-232C Levels; otherwise connect to GND
DRAWING CAMPUS
sun microsystems
Campus-1 schematic
SCC's TOD EPROM S4-Clock FDC
Fri Jul 15 10:01:34 1988
tomw



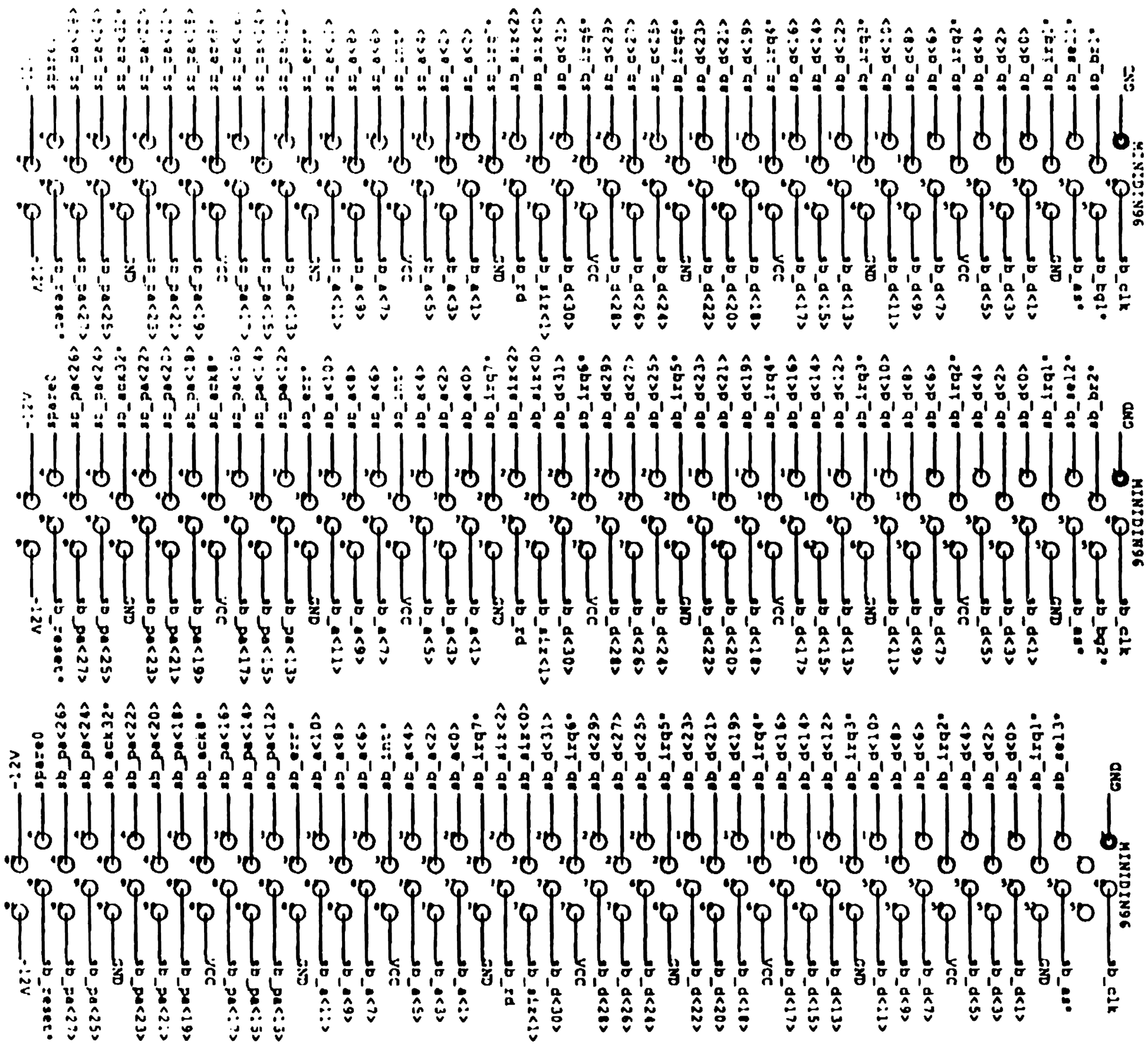
sun microsystems

TITLE: Campus-1 schematic, S4-RAM and SIMMs

DATE: Fri Jul 15 09:59:58 1988

ENGINEER: tomw

PAGE: 5



CAMPUS DRAWING

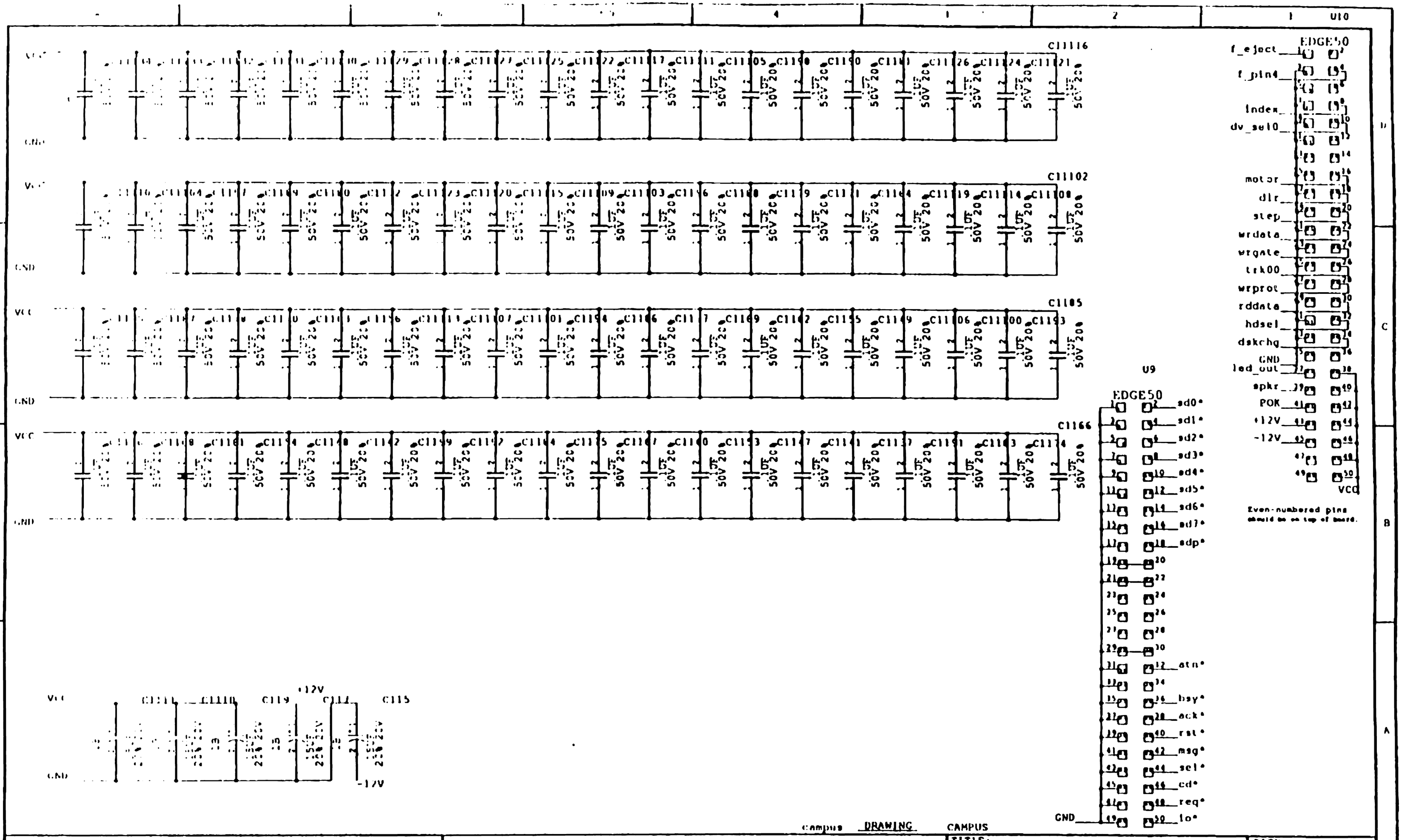
sun microsystems

TITLE: Campus-1 SBus Slots

DATE: Fri Jul 5 10:02:02 1988

ENGINEER: tomw

PAGE: 9



Campus DRAWING CAMPUS

sun microsystems

TITLE: Campus-1 Schematic Bypass Caps

ENGINEER: tomw

DATE: Fri Jul 15 09:55:25 1988

PAGE: 11

Sun Microsystems S4 Chip Set

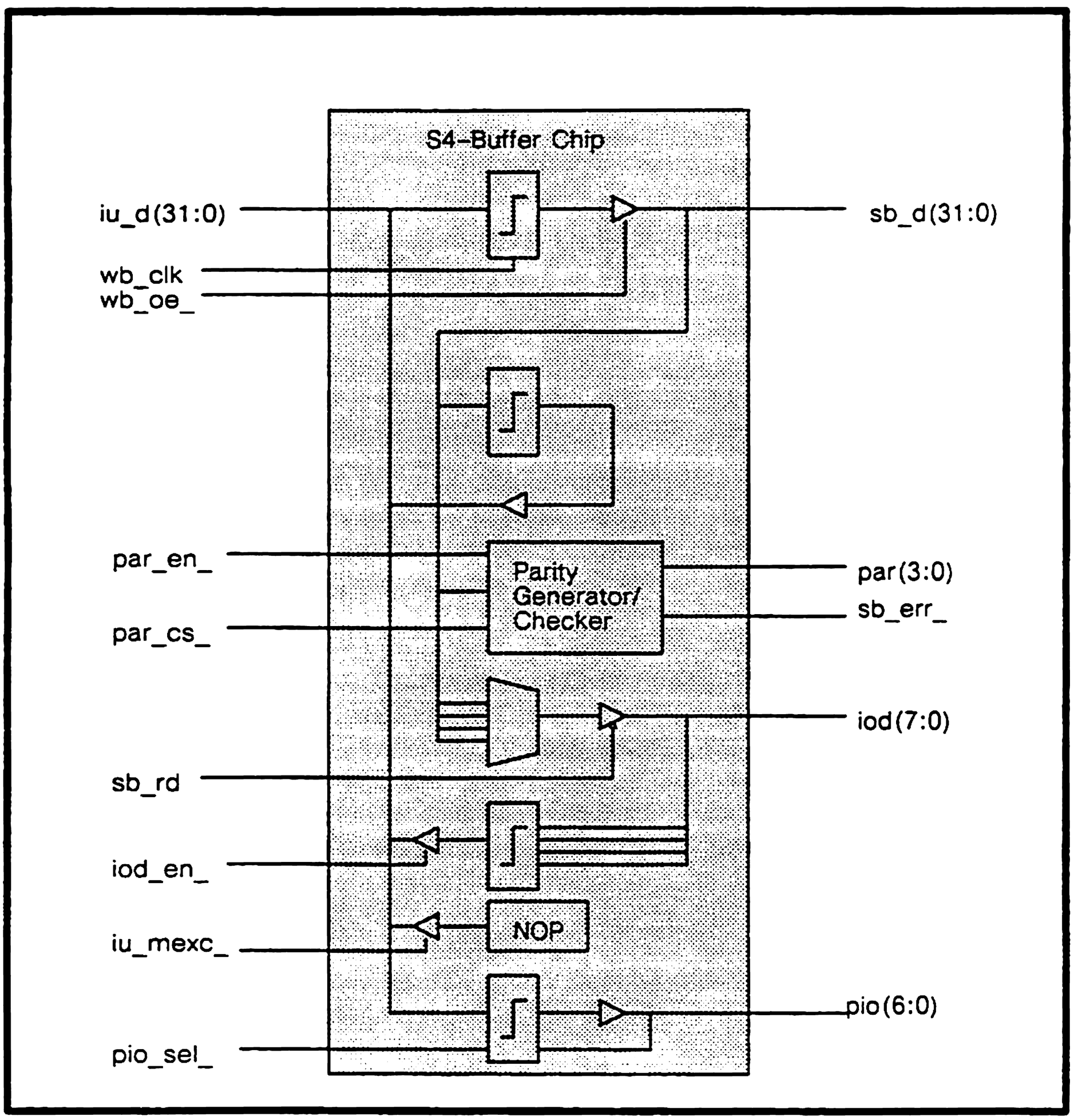
Sun SPARC(TM) Support System

S4-Buffer

Preliminary

Features

* Generates and checks parity on main memory accesses
* Performs buffered write cycles in conjunction with the S4-Cache chip
* Multiplexes 32-bit IU data bus down to 8-bit IO data bus on write cycles
* Demultiplexes and latches 8-bit IO data bus up to 32-bit IU data bus on read cycles
* Contains byte-packing registers for dynamically sized reads from SchoolBus data bus
* Contains Sun-4 Parity Control Register
* Contains 7-bit open-drain general purpose I/O register (PIO)
* Forces No Op on memory exceptions



Pin Description

Symbol	Type	Description
IU Interface	**34**	
clk	DRVC16	Clock
iu_d(31:0)	BD4TRU	Integer Unit Data bus.
iu_mexc_	IBUFNU	Integer Unit Memory Exception.
SchoolBus Interface	**39**	
sb_a(1:0)	IBUFU	SBus Address Bus.
sb_siz(1:0)	IBUF	SBus Size.
sb_as_	IBUF	SBus Address Strobe.
sb_rd	IBUFU	SBus Read.
sb_d(31:0)	BD4TRU	SBus Data bus.
sb_ack32_	IBUFNU	SBus 32-bit Acknowledge.
sb_ack8_	IBUFNU	SBus 8-bit Acknowledge.
sb_err_	BT8	SBus Error.
sb_reset_	IBUFN	SBus Reset.
Miscellaneous	**20**	
par(3:0)	BD4TRU	Parity data bus.
par_en_	IBUFN	Parity Enable. Enables parity check on current cycle.
par_cs-	IBUFN	Parity Chip Select. Address decode for Parity Ctl Reg.
pio_sel_	IBUFN	Parallel I/O Port Select. Address decode for PIO Reg.
pio(6:0)	BD4TOD	Parallel I/O Port.
iod(7:0)	BD4TRU	Input Output Data bus.
iod_en_	IBUFN	Input Output Data Bus Enable.
wb_oe_	IBUFN	Write Buffer Output Enable.
wb_ce_	IBUF	Write Buffer Clock Enable.
cd_oe_	IBUFN	Cache Data Output Enable
s4b_od_	IBUFNU	S4-Buffer chip Output Disable
s4b_test_	IBUFNU	Low for chip test mode
para	BT1	Parametric Test Output.
Signals	105	
Device Type:	LMA9141	(IO:110 VDD:2 VSS:6)
Package Type:	PFP120	(PADS:120 VDD:6 VSS:10)

Input/Output Buffer Definitions

DRVC#	Input Clock Buffer, CMOS, non-inverting. # indicates drive capability.
IBUF	Input buffer, CMOS, non-inverting
IBUFU	Input buffer, CMOS, non-inverting, internal pullup

IBUFN	Input buffer, CMOS, inverting
IBUFNU	Input buffer, CMOS, inverting, internal pullup
TLCHT	Input buffer, TTL, non-inverting
TLCHTN	Input buffer, TTL, inverting
BD#TRU	Bidirectional buffer, TTL input levels, # indicates output drive, internal pullup
BT#	Tri-statable output buffer, CMOS, # indicates output drive current.
BD4TOD	Open drain buffer, TTL, non-inverting.

Functional Description

Read Cycles

On read cycles the S4-Buffer chip will latch data on the rising edge of the clock after an acknowledge is sampled true. Data will be taken from the IOD(7:0) bus if IOD_EN_ is asserted, and from the SB_D(31:0) bus if IOD_EN_ is negated. On all read cycles the entire IU_D(31:0) bus is driven with the current contents of one set of the internal read latches. On read cycles from the Parallel I/O Register, the wire-or of pio(6:0) with the contents of the register will be read. If pio(6:0) is set to 7F, the contents of the register will be read.

Byte Packing
It is desirable to be able to execute code contained in 8-bit devices on either the SBus or the IOD bus. This necessitates packing the bytes up to fit the word length of the SPARC chip, as instruction fetches assume this data width. An IU word-length read will be converted into the appropriate number of shorter read cycles if the accessed device indicates its port width is less than 32 bits. When reading from the IOD bus (IOD_EN_ is asserted), the data will be presented to the IU as shown in the following table:

sb_a(1)	sb_a(0)	Activated Read Latch
0	0	iu_d(31:24)
0	1	iu_d(23:16)
1	0	iu_d(15:8)
1	1	iu_d(7:0)

When reading from the SB_D bus (IOD_EN_ not asserted), data will be latched from the SB_D bus depending on the port size acknowledge as shown in the following tables:

Acknowledge & Port Size Decoding

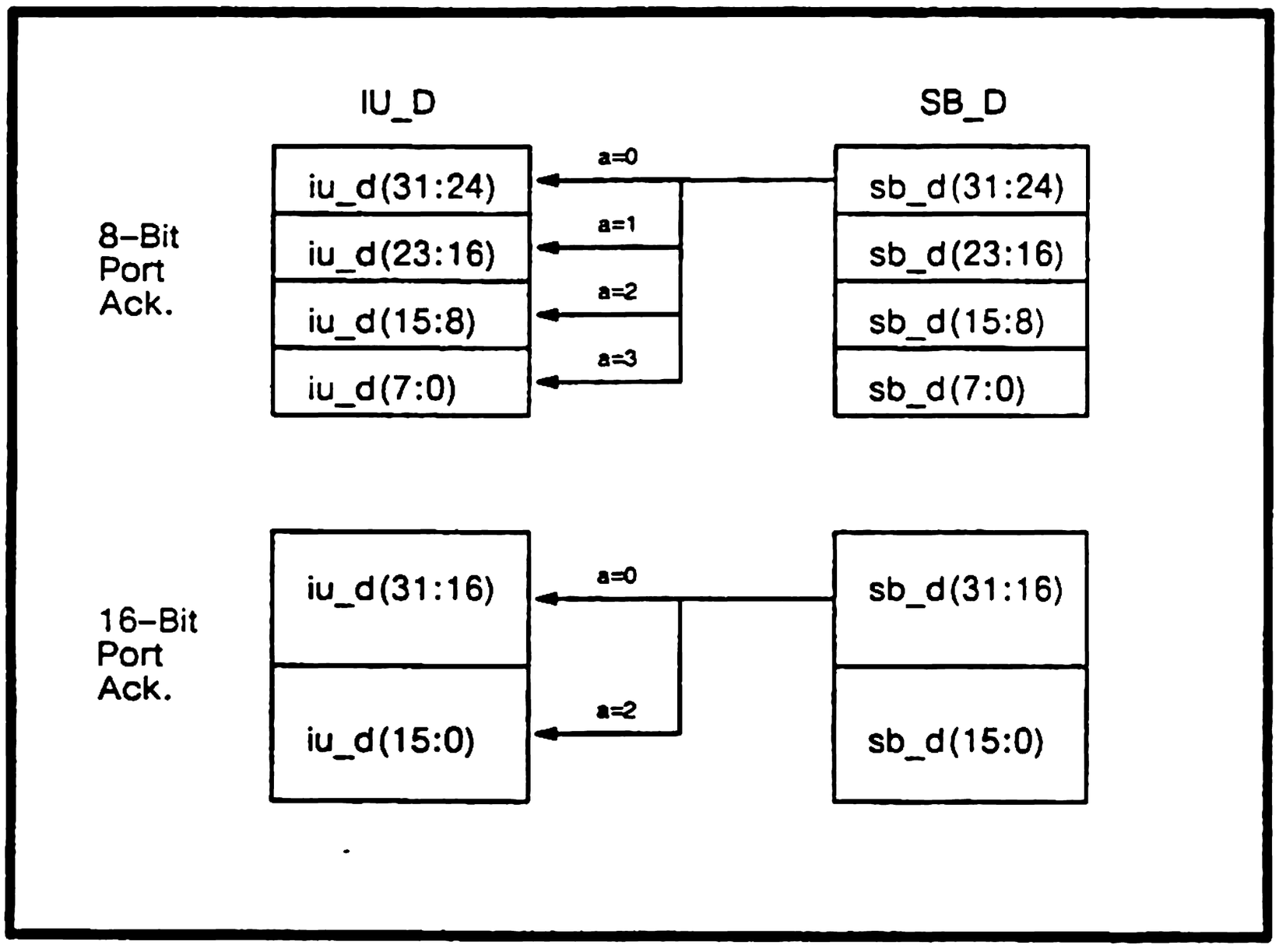
sb_ack32_	sb_ack8_	sb_err_	Definition
1	1	1	Insert Wait States
1	1	0	Error
1	0	1	8-bit port ack.
1	0	0	Rerun
0	1	1	32-bit port ack.
0	1	0	Error
0	0	1	16-bit port ack.
0	0	0	Reserved

Port Location

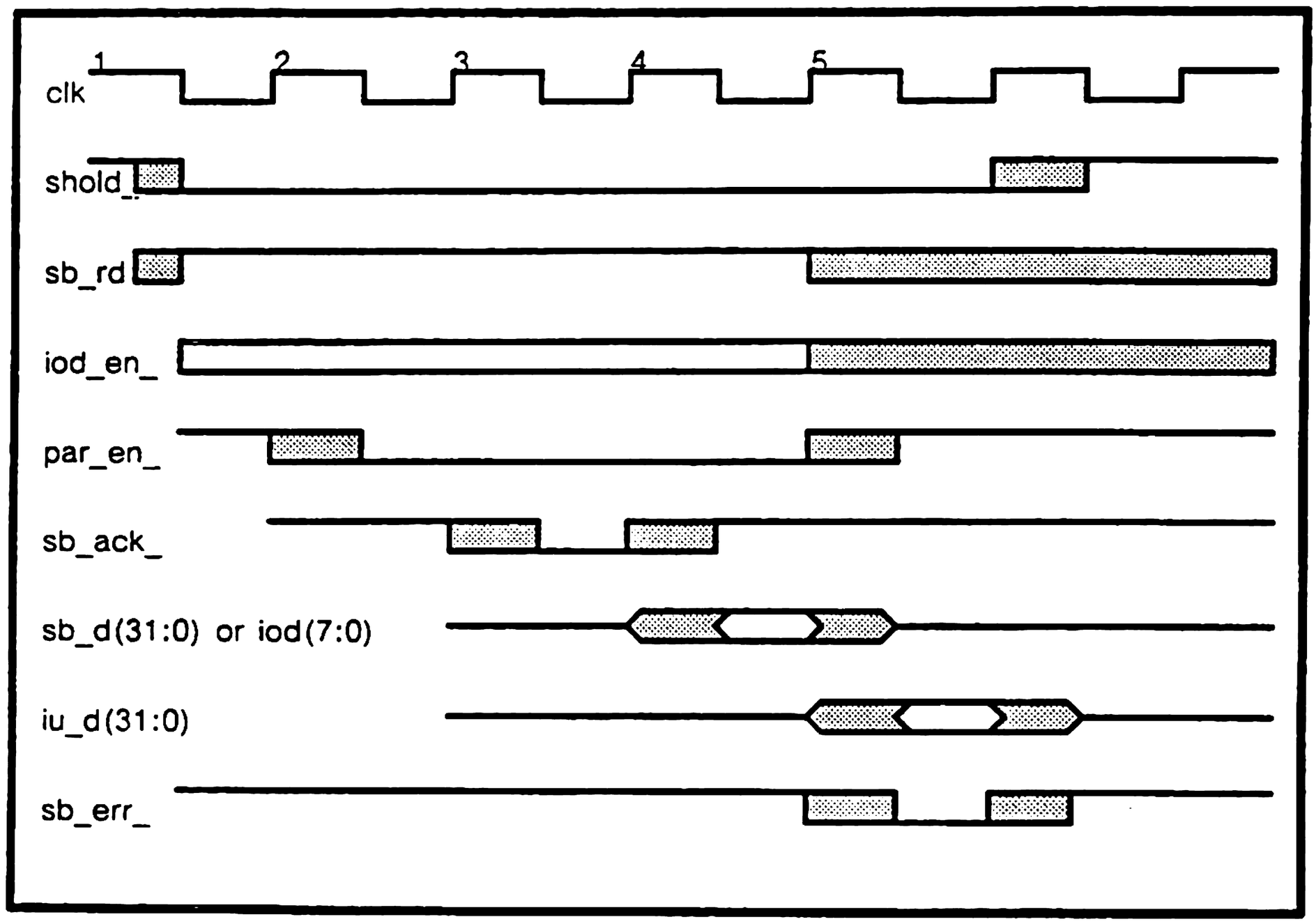
The location of 8, 16 and 32-bit ports on the 32-bit SchoolBus data bus is defined as follows:

sb_d(31:24)	sb_d(23:16)	sb_d(15:8)	sb_d(7:0)
8-bit port			
16-bit port			
32-bit port			

SB_D Read Data Latching

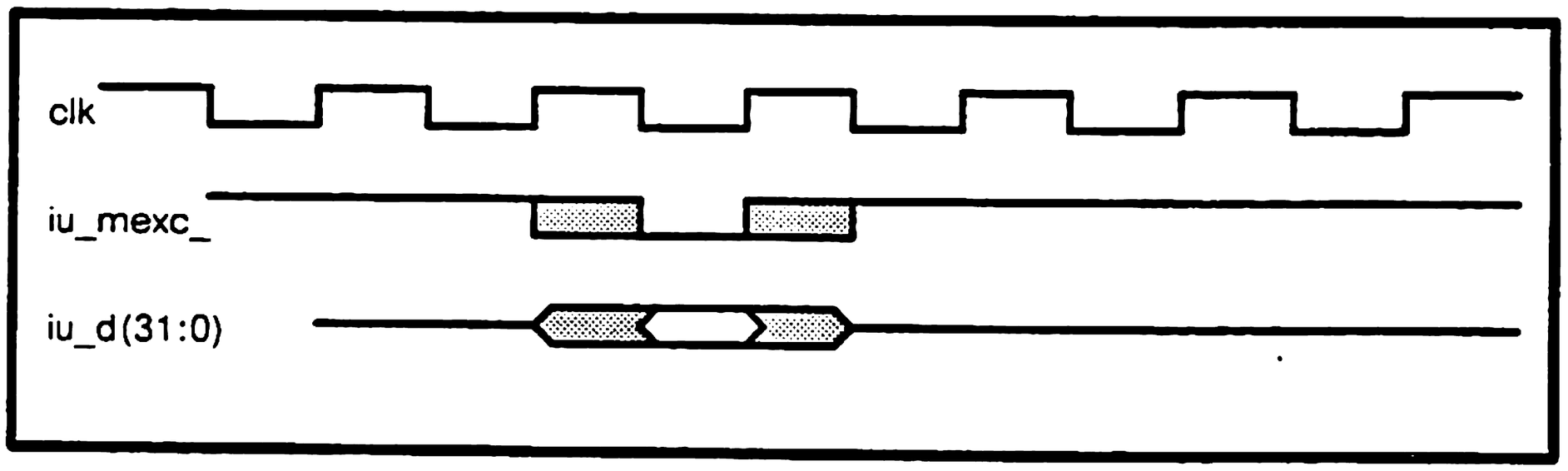


Read Data Timing Diagram



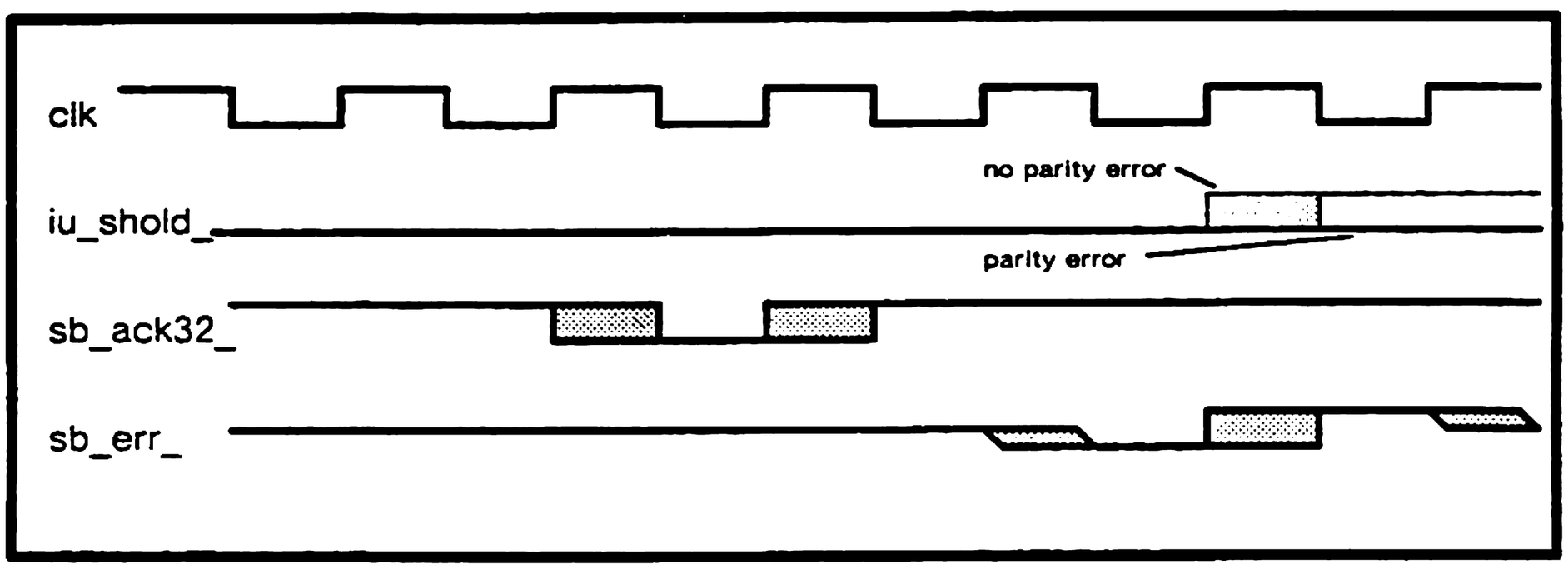
Memory Exceptions on Read Cycles

When IU_MEXC_ is asserted ona read cycle, the S4-Buffer chip will drive the IU_D bus with a SPARC No Op binary code for the duration of the IU_MEXC_ signal as shown in the following diagram.

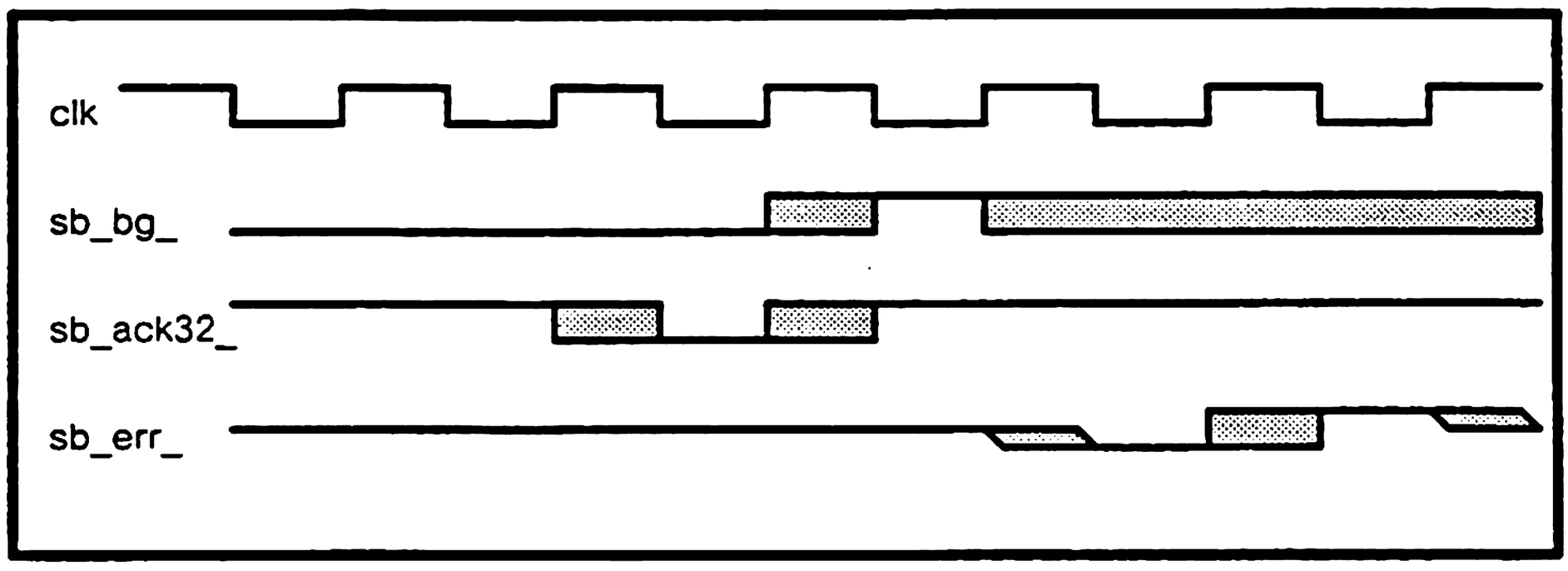


Parity Checking
Parity is checked on read cycles during which PAR_EN_ is active and the Parity Check bit is set in the Parity Control Register (See below for a description of the Parity Control Register). Parity errors are reported by asserting SB_ERR_ for one clock period, and setting the bits in the parity control register corresponding to the bytes in which parity errors were detected. SB_ERR_ will cause the S4-Cache chip to assert IU_MEXC_, causing the IU to take a memory exception trap. Parity checking is even, meaning a byte of ones requires a zero parity bit, so that a data and parity bus floating high will cause a parity error.

Parity errors are reported on IU cycles by a one-clock low pulse on the SB_ERR_ signal, two clocks after SB_ACK32_, as shown in the following diagram:



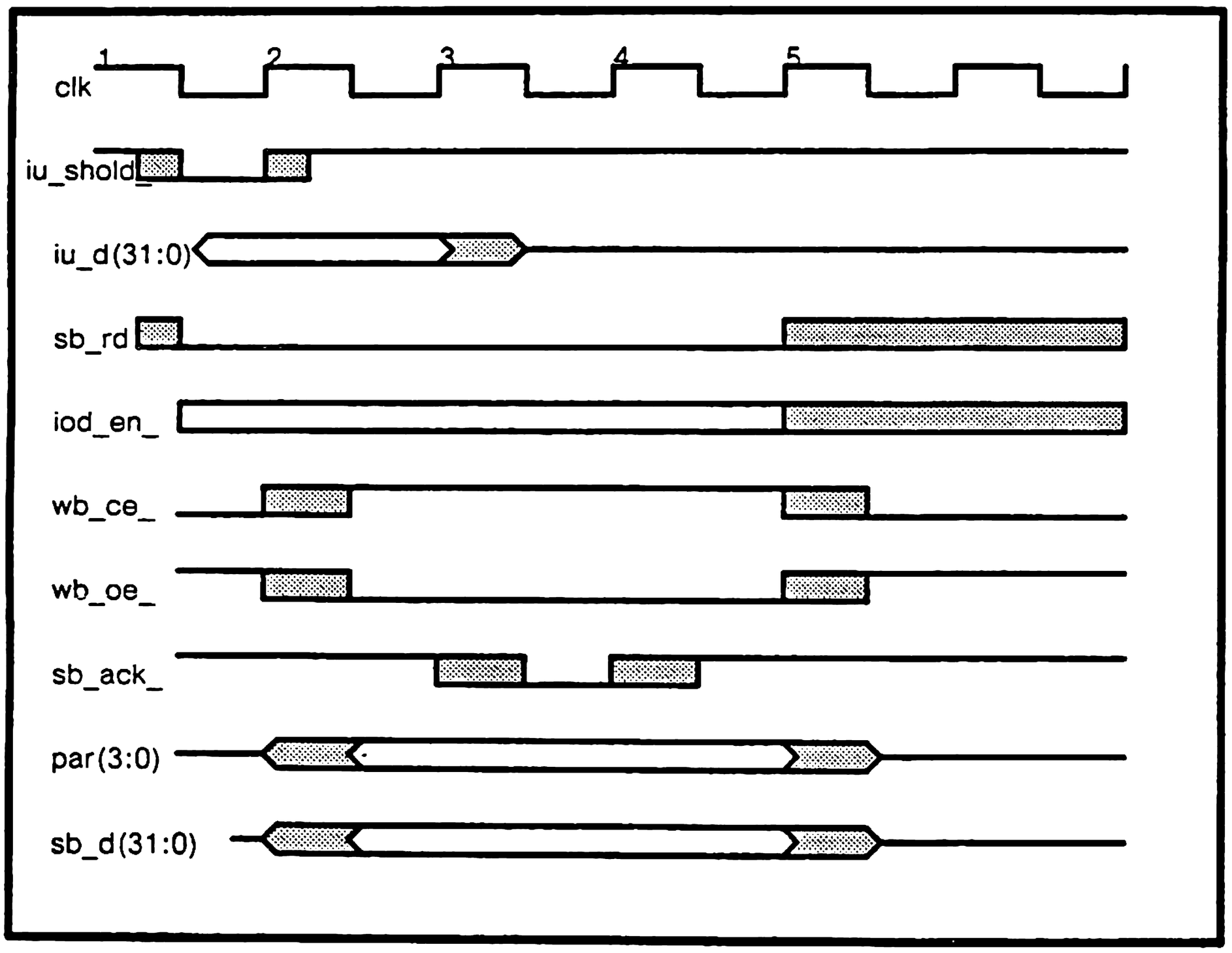
Parity errors are reported on DVMA cycles by a one-clock low pulse on the SB_ERR_ signal, one clock after SB_ACK32_, as shown in the following diagram. Note that on DVMA cycles, this SB_ERR_ signal could occur after SB_BG_ has been asserted to another device, so that device must take care not to react.



Write Cycles

The S4-Buffer chip contains a 32-bit write buffer, a holding register for IU write data that allows the IU to continue executing from the cache while the write cycle is performed on the SchoolBus. Data is clocked into the register when WB_CE_ is asserted, and the output is enabled by WB_OE_. These two signals are generated by the S4_Cache chip. WB_OE_ will always be negated during DMA so that DMA devices can drive the SB_D(31:0) bus. Any byte of the write buffer can be multiplexed onto the IOD(7:0) bus, which will be output enabled on write cycles when IOD_EN_ is asserted. Parity is generated on all write cycles.

Write Cycle Timing Diagram



The system bus controller implements dynamic bus sizing for CPU cycles. This function is performed through the joint efforts of the S4-Cache and the S4-Buffer.Taking the desired transfer width and the port size into account, the bus controller packs data from narrower ports up to the desired width by performing several bus cycles. This byte packing is performed only for CPU cycles, not for DMA cycles. The cycles appear as separate cycles indistinguishable from cycles that don't involve byte packing.

Transfer Size	Port Size	Controller Response
1-Byte	Any	Single BYTE cycle
2-Byte	8-bit	Two BYTE cycles
"	16-bit	One HALF cycle
"	32-bit	One HALF cycle
4-Byte	8-bit	Four BYTE cycles
"	16-bit	Two HALF cycles
"	32-bit	One WORD cycle

Revisions

Date	Description	By
8/17/88	Corrected table on page 10.	DED
3/31/89	Corrected parity register on page 11.	DED

The steering of the data from the IU data bus to the S-Bus data bus is a function of the size of the IU data transfer, the address and the port size acknowledge.

For IU data BYTE 0 = w, BYTE 1 = x, BYTE 2 = y and BYTE 3 = z, the following table describes the byte steering that takes place for the various combinations of transfer size attempted by the IU and port sizes which acknowledge.

IU Data

BYTE 0	BYTE 1	BYTE 2	BYTE 3
w	x	y	z

SBus Data

8 bit transfer	w	w	w	w	a = 0
	x	x	x	x	a = 1
	y	y	y	y	a = 2
	z	z	z	z	a = 3
16 bit transfer	w	x	w	x	a = 0
	x	y	x	y	a = 1
	y	z	y	z	a = 2
	z	w	z	w	a = 3
32 bit transfer	w	x	y	z	a = 0
	x	y	z	w	a = 1
	y	z	w	x	a = 2
	z	w	x	y	a = 3

Notes: 1. a = SB_A(1:0)
2. S4-Cache controls SB_A(1:0) and SB_ACK8_, SB_ACK32_ to generate proper number of cycles.

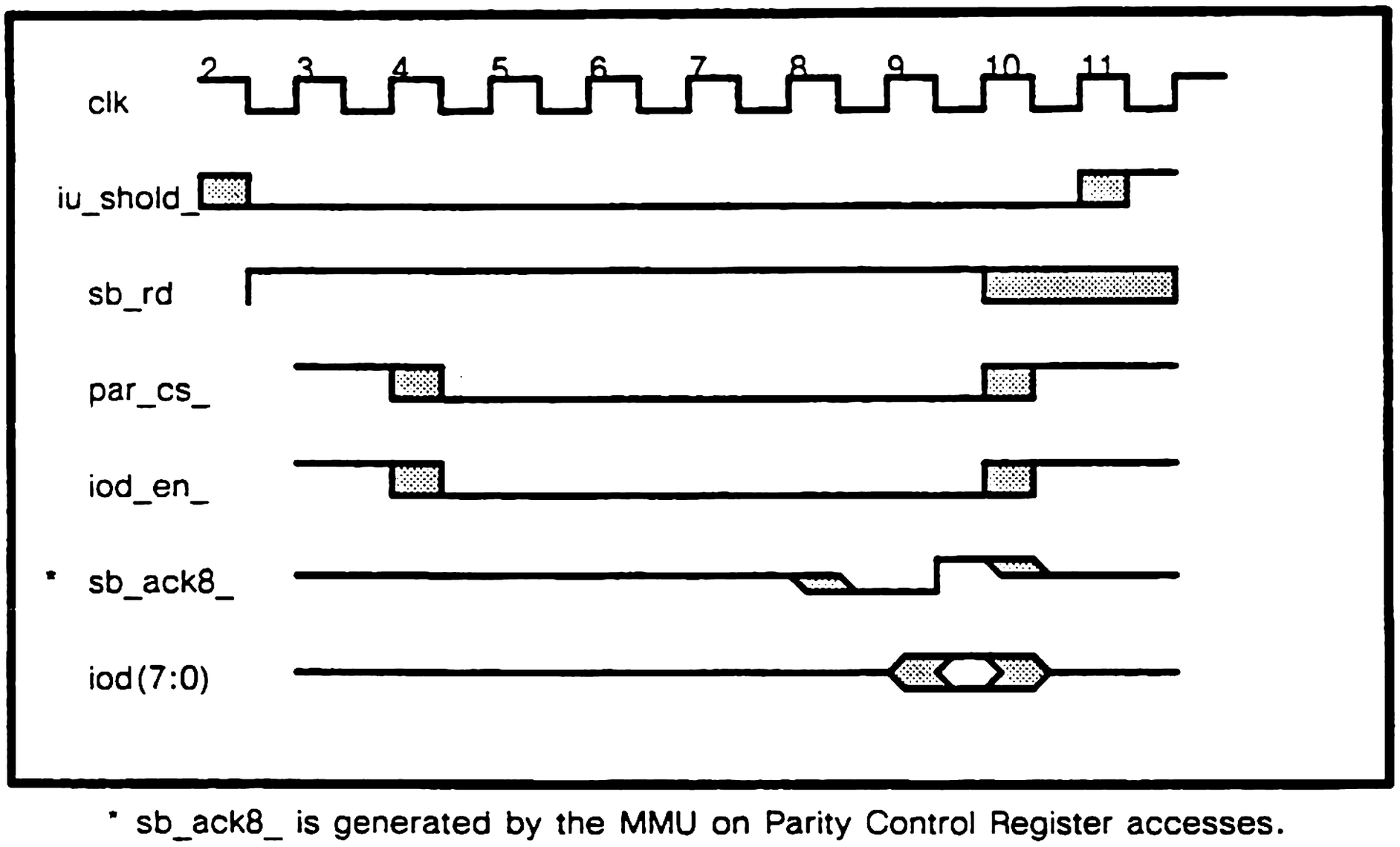
Parity Control Register

The Parity Control Register provides facilities for enabling and reporting parity errors and for testing the parity generation and checking logic. It is a 32-bit read/write register, cleared on SB_RESET_, accessible 8 bits at a time over the IOD bus. It has the following fields:

D(31:8)	Reserved	Read as zero
D(7)	Parity Error	Set on any parity error
D(6)	Second Error	Set if D(7) is set and new error occurs
D(5)	Parity Test	Set to write parity with the inverse polarity to test the operation of the parity error circuitry. With Parity Test off, correct parity is generated on all memory write cycles.
D(4)	Parity Check	Enables parity checking
D(3)	Parity Error 24	Records parity error on data bits 31:24
D(2)	Parity Error 16	Records parity error on data bits 23:16
D(1)	Parity Error 08	Records parity error on data bits 15:8
D(0)	Parity Error 00	Records parity error on data bits 7:0

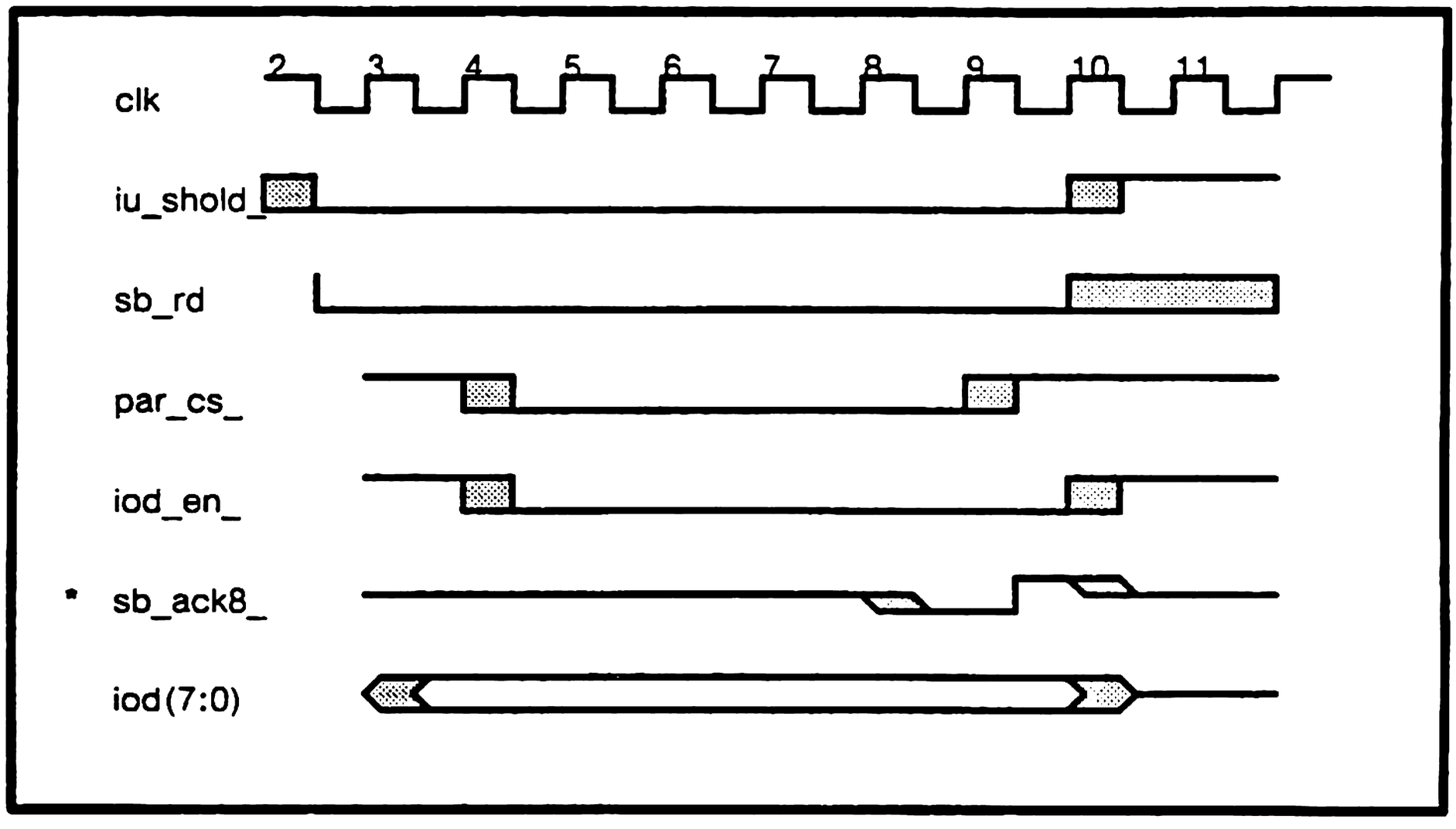
Note that the Error Bits D(7, 6, 3:0) are not writable. They are set by errors and reset automatically when read back.

Parity Control Register Read



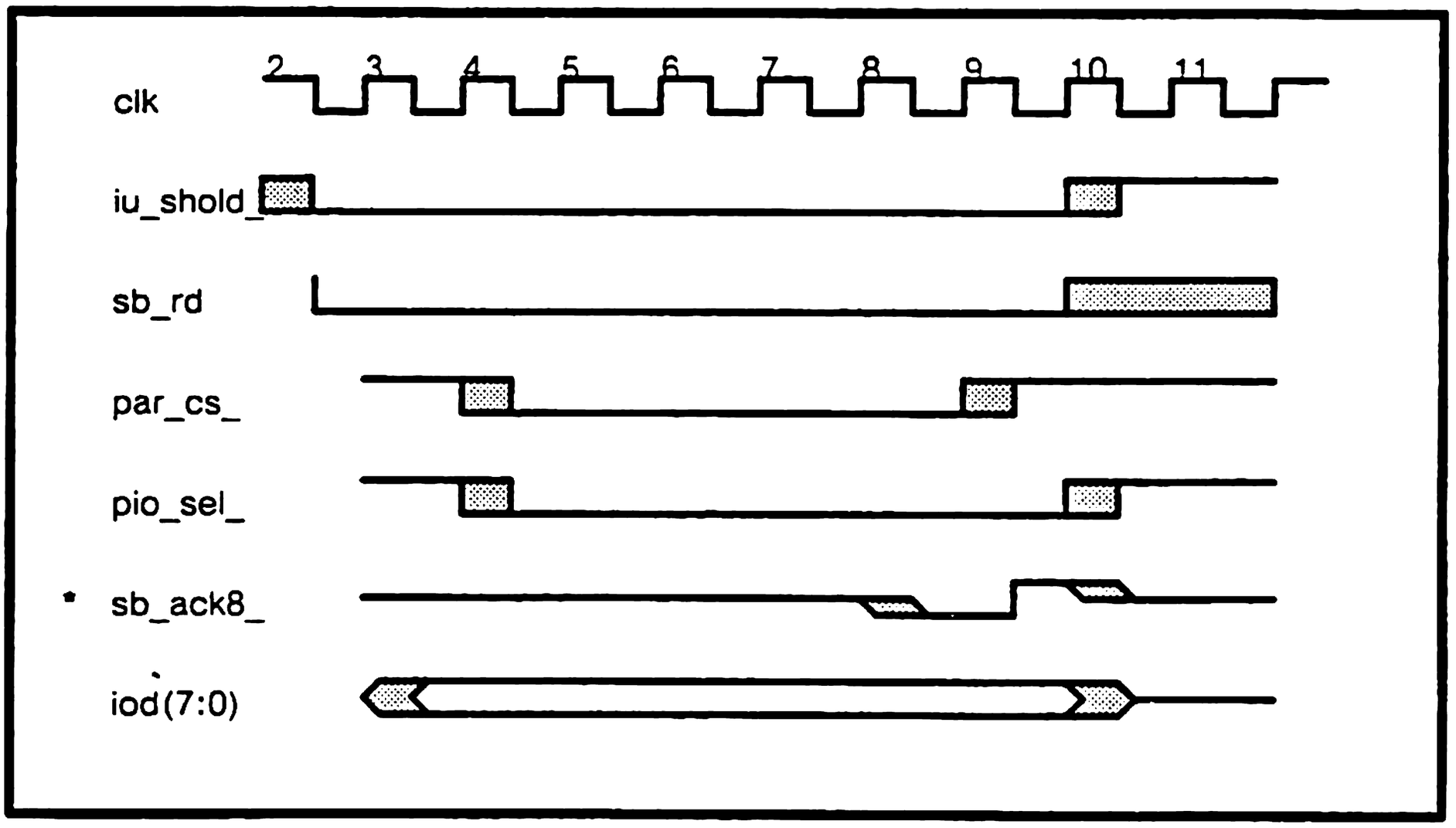
* sb_ack8_ is generated by the MMU on Parity Control Register accesses.

Parity Control Register Write



* sb_ack8_ is generated by the MMU on Parity Control Register accesses.

Parallel I/O Register Write



* sb_ack8_ is generated by the MMU on Parallel I/O Register accesses.

The Parallel I/O Register is selected by the pio_sel_ signal. The output of this register is an open drain data bus, pio(6:0). The timing for this register is identical to that of the Parity Control Register with pio_sel_ replacing par_cs- in the timing diagrams. The register is a 32-bit device with bits 31 through 7 reserved. The reserved bits read back as zeros.

Timing Specifications

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C, Output Load=100 pF

Symbol	From	To	min	max	unit
t	xclk high	xclk high	40	---	ns
t	clk	iu_d	10.5	23.5	ns
t	clk	sb_d	18	27	ns
t	clk	par	23	34.5	ns
t	clk	pio	12	21.5	ns
t	clk	iod	8	31.5	ns
t	clk	sb_merr_	16	23.5	ns
t	iu_mexc	iu_d	6.5	12	ns
t	wb_oe_	sb_d	5.5	21	ns

Setup time for all signals is 15 ns. Hold time for all signals is 3 ns.

Sun Microsystems S4 Chip Set

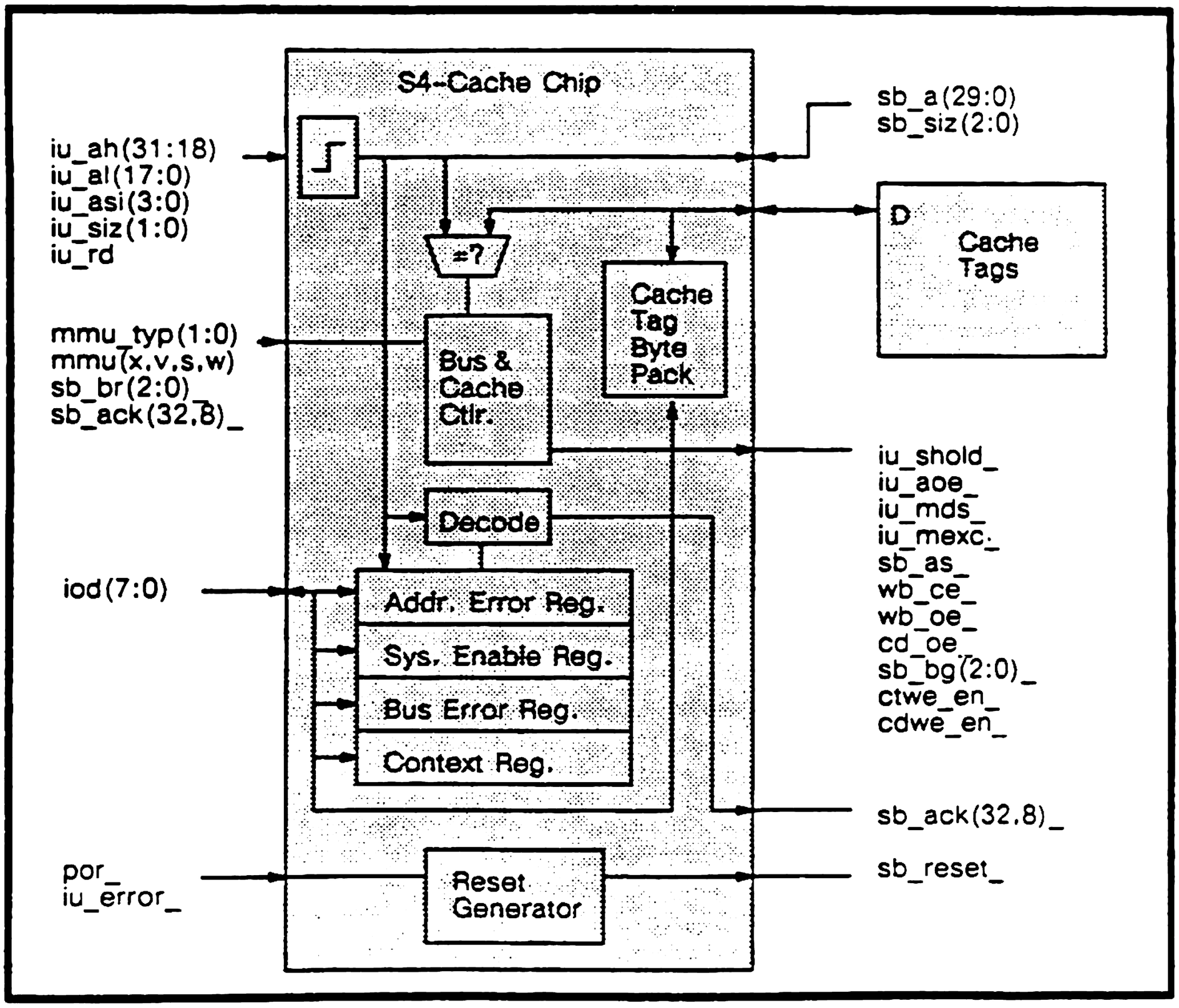
Sun SPARC(TM) Support System

S4-Cache

Preliminary

Features

- Implements 64-256 KByte write-through Instruction/Data cache with 16-byte line size
- Performs cache tag comparison
- Controls SBus reads and writes
- Automatically fills cache on cache misses
- Controls mastership of SBus for DMA
- Performs buffered writes with external write buffer
- Replaces cache tag read/write buffers
- Performs cache flush comparisons
- Controls system-wide byte packing
- Contains Sun-4 Virtual Address Error Latches
- Maintains copy of 4-bit Sun-4 context register
- Contains Sun-4 System Enable Register
- Contains Sun-4 Bus Error Registers
- Monitors bus for unacknowledged transfers
- Generates system reset



Pin Description

Symbol	Type	Description
IU Interface	**50**	
iu_clk	DRVT8	Integer Unit clock. The main system clock.
iu_rd	TLCHTU	Integer Unit Read. High for read cycles, low for writes.
iu_wr_	TLCHTU	Integer Unit Write. Low in second clock of write cycle.
iu_siz(1:0)	TLCHTU	Integer Unit Size. Indicates number of bytes in transfer
iu_error_	TLCHTNU	Integer Unit Error. Low when IU halts due to errors.
iu_nulcyc	TLCHTD	Integer Unit Null Cycle. Address bus is not valid.
iu_hal_	TLCHTU	Integer Unit Hold Address Low. Don't clock Cache Address Register (Sunrise designs only).
iu_ldst	TLCHTN	Integer Unit Lock. Don't steal the bus from the IU.
iu_asi(3:0)	TLCHTU	Integer Unit Address Space Identifiers.
iu_ah(31:18)	TLCHTU	Integer Unit High Addresses.
iu_al(17:0)	BD4TU	Integer Unit Low Addresses.
iu_shold_	BT8	Integer Unit Hold. Stops internal IU clock.
iu_mds_	BT4	Integer Unit Memory Data Strobe.
iu_mexc_	BT4	Integer Unit Memory Exception.
iu_aoe_	BT4	Integer Unit Address Output Enable.
por_	TLCHTNU	Power-On Reset. Low until power supply is stable.
SBus Interface	**46**	
sb_as_	BT4	Address Strobe. Physical addresses avail. next clock.
sb_br(2:0)_	TLCHTNU	Schoolbus Bus Request. Request to perform DMA.
sb_bg(2:0)_	BT4	Schoolbus Bus Grant. Permission to perform DMA.
sb_rd	BD4TU	Schoolbus Read. High for read cycles, low for writes.
sb_siz(2:0)	BT4TU	Schoolbus Size. Indicates number of bytes transferred
sb_a(29:0)	BD4TU	Schoolbus Address Bus. Virtual addresses.
sb_ack8_	BD8TU	SBus 8-bit Acknowledge.
sb_ack32_	BD8TU	SBus 32-bit Acknowledge.
sb_err_	BD4TU	Schoolbus Error. Failed data transfer.
sb_merr_	BD8TU	Schoolbus Memory Error. Parity or ECC error detected
sb_reset_	BT8	SBus reset. Initializes all registers and logic.
MMU Interface	**19**	
mmu_s	TLCHT	MMU Supervisor Only protection bit.
mmu_wa	TLCHT	MMU Write Allowed protection bit.
mmu_x	TLCHT	MMU Don't Cache bit.
mmu_v	TLCHT	MMU Valid bit.
mmu_typ(1:0)	TLCHT	MMU Type bits. Indicate Memory, I/O, VMEbus, etc.
user_	BT4	User-mode Address Space Identifiers from the IU.
devspc_	BT4	Device/Control Space, low for device space.
ctl(2:0)	BT4	Encoded Control space device IDs.
iod(7:0)	BD4TRU	Input/Output Data Bus. Connects to 8-bit I/O Devices.

Cache Interface	**25**	
ct_a(29:16)	BD4TU	Cache Tag Address bits
ct_c(3:0)	BD4TU	Cache Tag Context bits
ct_s	BD4TU	Cache Tag Supervisor
ct_v	BD4TU	Cache Taq Valid
ct_wa	BD4TU	Cache Tag Write Allowed
ctwe_en_	BT4	Cache Tag Write Enable Enable. Goes to S4-Clock.
cdwe_en_	BT4	Cache Data Write Enable Enable. Goes to S4-Clock.
cd_oe_	BT4	Cache Data Output Enable.
car_en_	BT4	Cache Address Register Clock Enable.
Miscellaneous	**4**	
wb_oe_	BT4	Write Buffer Output Enable
wb_ce_	BT4	Write Buffer Clock Enable.
s4c_oe_	TLCHTNU	S4-Cache chip output enable.
s4c_test_	IBUFNU	S4-Cache chip Test mode.

Signals:	144	
Device Type:	LMA9284	(IO:158 VDD:4 VSS:6)
Package Type:	PFP160	(PADS:160 VDD:7 VSS:9)

Input/Output Buffer Definitions

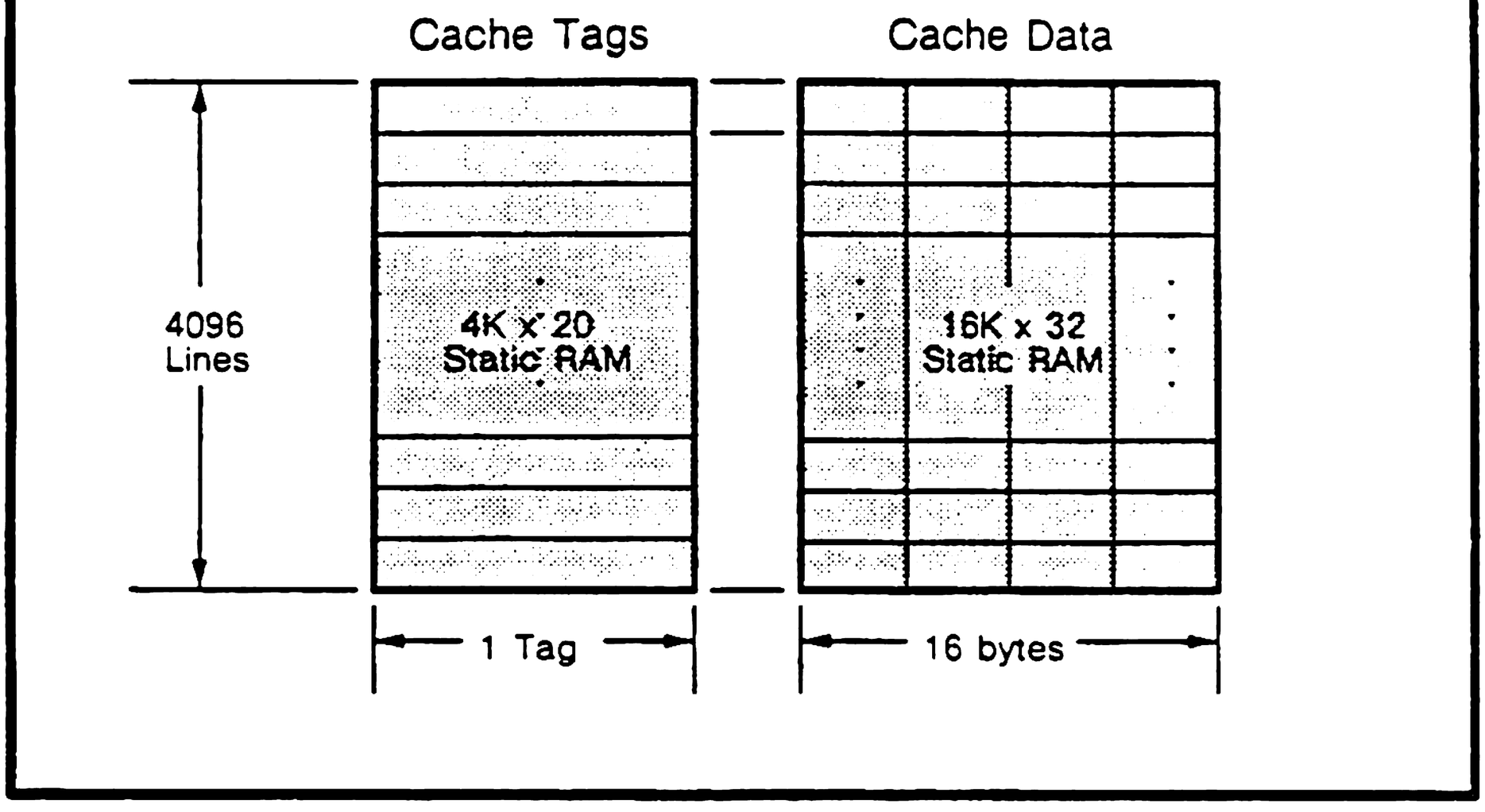
DRVC8	Input clock driver
IBUFNU	Input buffer, CMOS level, inverting, internal pullup
TLCHT	Input buffer, TTL level, non-inverting
TLCHTU	Input buffer, TTL level, non-inverting, internal pullup
TLCHTNU	Input buffer, TTL level, inverting
BD#TU	Bidirectional buffer, TTL input levels, internal pullup, # indicates output drive
BD#TRU	Bidirectional buffer, TTL input levels, internal pullup, slew-rate controlled output, # indicates output drive
BT#	Tri-statable Output buffer, CMOS, # indicates output drive current.

Table of Contents

Functional Description

Cache Overview

The cache implemented with the aid of the S4-Cache chip is a write-through mixed instruction/data cache with a 16-byte line size. A typical implementation is shown in the following diagram:

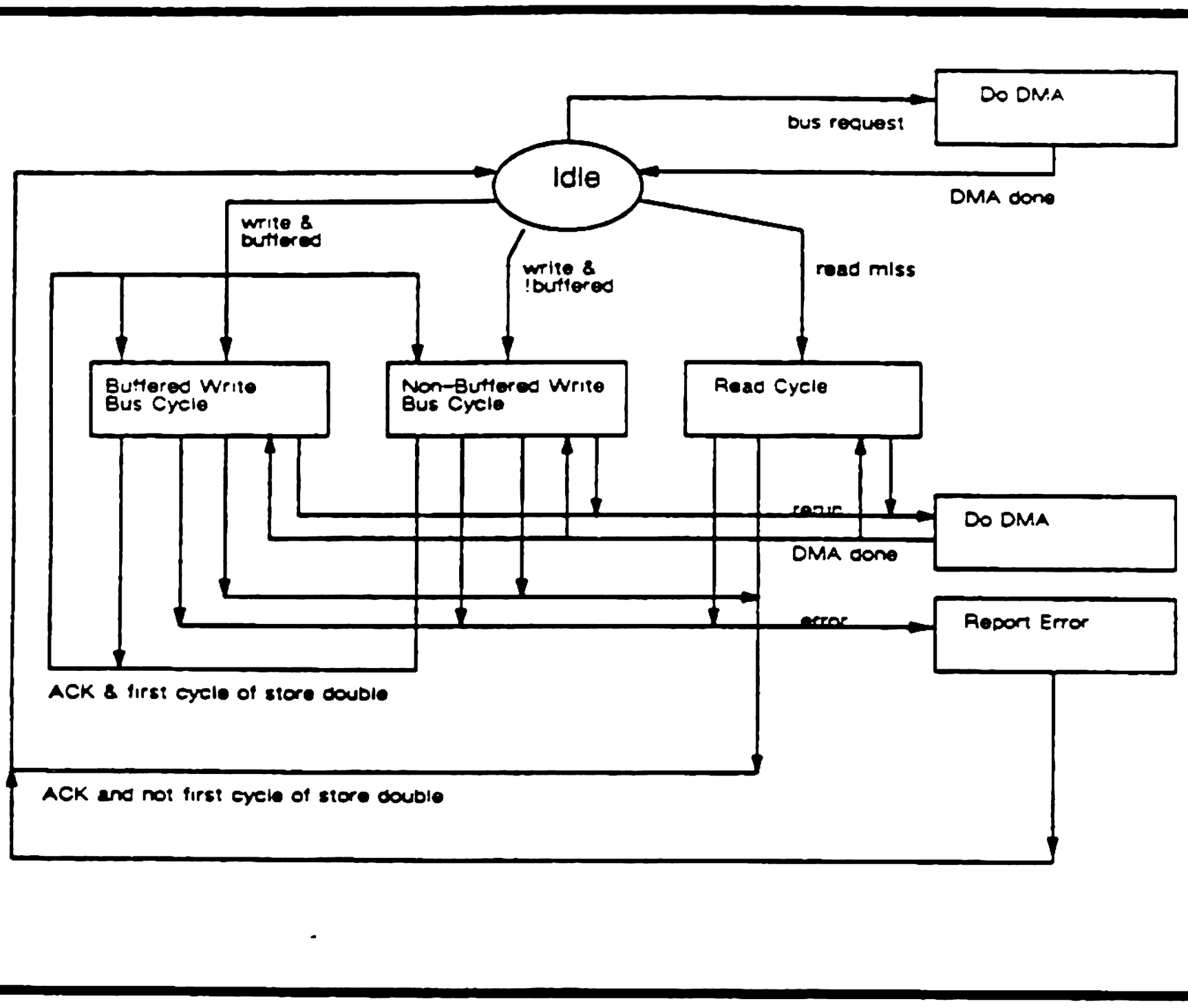


The cache tag and cache data memories are built using external generic static RAM chips. Although the programmer's model of the cache data RAM is 4096 lines of 16 bytes, it is currently implemented with eight 16K x 4 static RAMs.

The size of the cache may vary from 4096 lines deep to 16,384 lines deep. Larger implementations of the cache will connect the unused cache tag pins to the appropriate address bits latched in the cache address register.

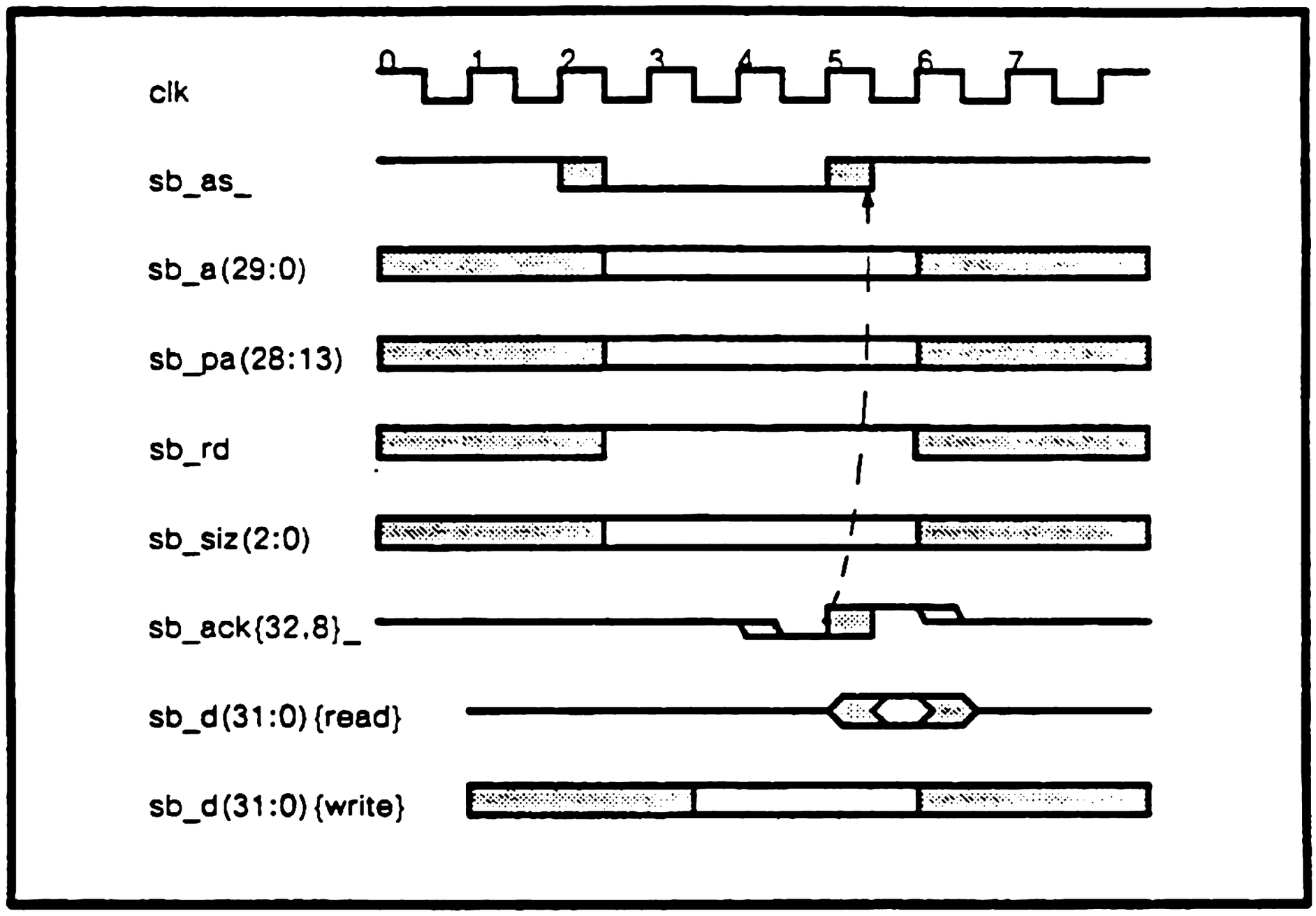
SBus Controller

The S4 chip set communicates over the SBus, a synchronous bus defined by the SBus Specification and controlled by the SBus controller, one state machine that controls SBus reads, SBus (buffered) writes, cache filling, DMA arbitration, and DMA cycles. A simplified state diagram for the bus controller is shown below.



SBus Overview

The SBus fundamental operation is shown in the diagram below. The SB_AS_ signal indicates the validity of SB_PA(28:00). SB_RD, SB_SIZ(2:0) and the signals derived combinatorially from these signals. On the rising clock edge at which AS_ is sampled true, these signals will also be valid with the setup specified. The cycle will continue until an acknowledge is received from the accessed device. Wait states will be inserted on the SBus until the acknowledge is received.

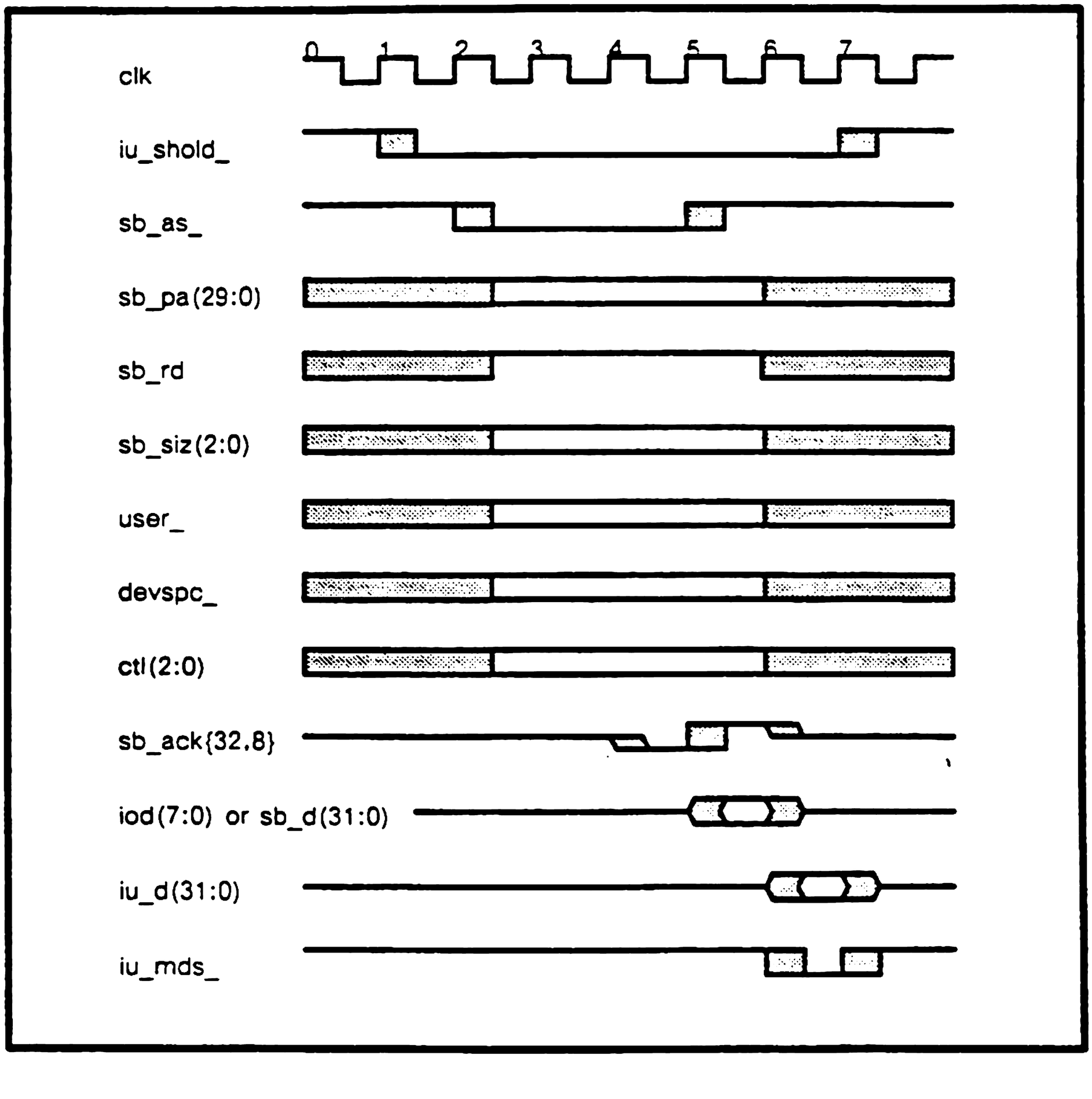


The addresses, read, and size signals will be held valid until the clock edge after the one on which the acknowledge is sampled true. See the tables below for acknowledge and size encoding.

Shared control signals SB_ACK32_, SB_ACK8_, SB_ERR_, and SB_MERR_ must follow a special protocol, which requires that the signal is taken out of tri-state mode, driven low for the desired number of clocks, then driven high for one clock before being tri-stated again. See the SBus specification for further details.

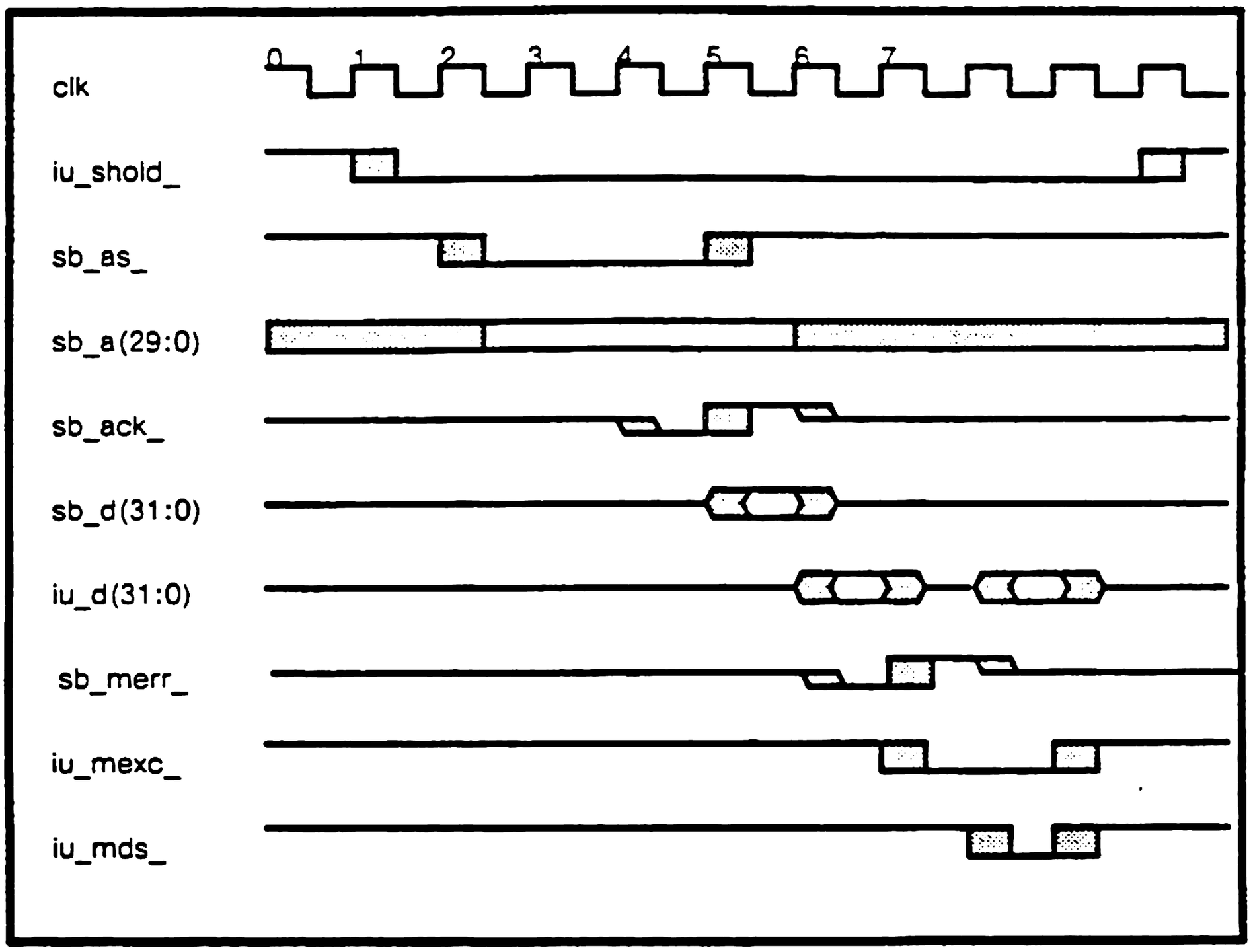
SBus Reads

An identical protocol is observed for all non-cacheable reads on the SBus, whether the accessed device is in control or device space (except for cache flushes), on the main PC board or on expansion boards, and using the 32-bit SB_D bus or the 8-bit IOD bus. As soon as the miss is detected, the IU is stopped by asserting IU_SHOLD_. SB_AS_ is asserted to indicate that on the next rising clock edge physical addresses, SB_RD, SB_SIZ(2:0), DEVSPC_, and CTL(2:0) will be valid. The bus controller then waits for an acknowledge encoded on SB_ACK32_, SB_ACK8_, and SB_ERR_. See "Byte Packing" below for the encoding of these signals. On the rising edge after the acknowledge is sampled as asserted, the data will be latched from the appropriate data bus into the S4-Buffer chip, and on the rising edge after that it will be strobed into the IU by the IU_MDS_ signal. NOTE THIS PIPELINING OF THE ACKNOWLEDGE.



Parity Errors

Parity errors are reported by the S4-Buffer chip to the S4-Cache chip via SB_MERR_. The S4-Cache chip reports parity errors to the IU on IU cycles by asserting IU_MEXC_ as shown in the following diagram.

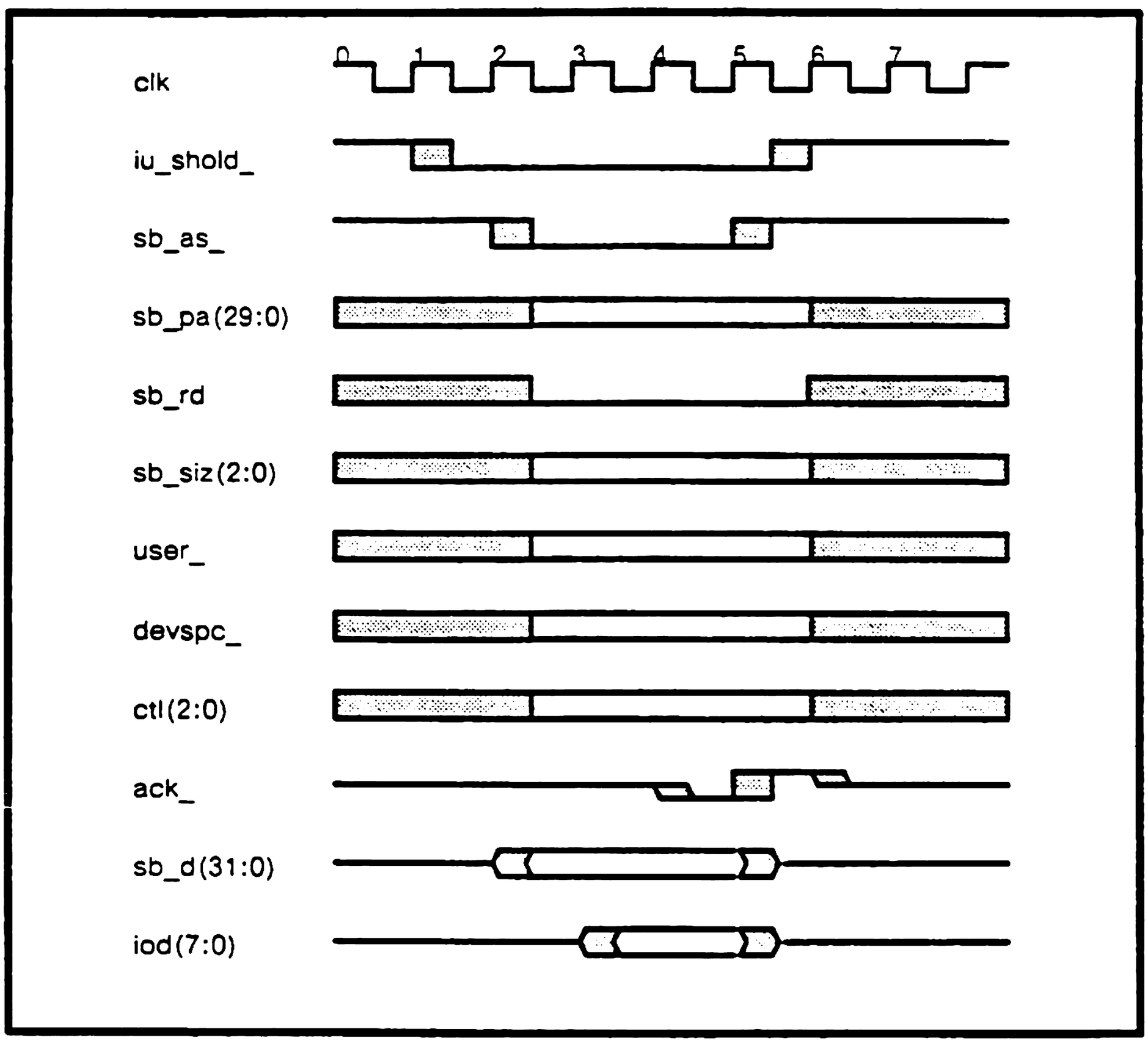


SBus Writes

The S4-Cache chip implements both standard write cycles and buffered write cycles. During a standard write cycle, the IU is held throughout the SBus cycle. During a buffered write cycle, the IU is started again after the MMU has been checked, so that the SBus write cycle can proceed concurrently with IU instruction and data fetches from the cache.

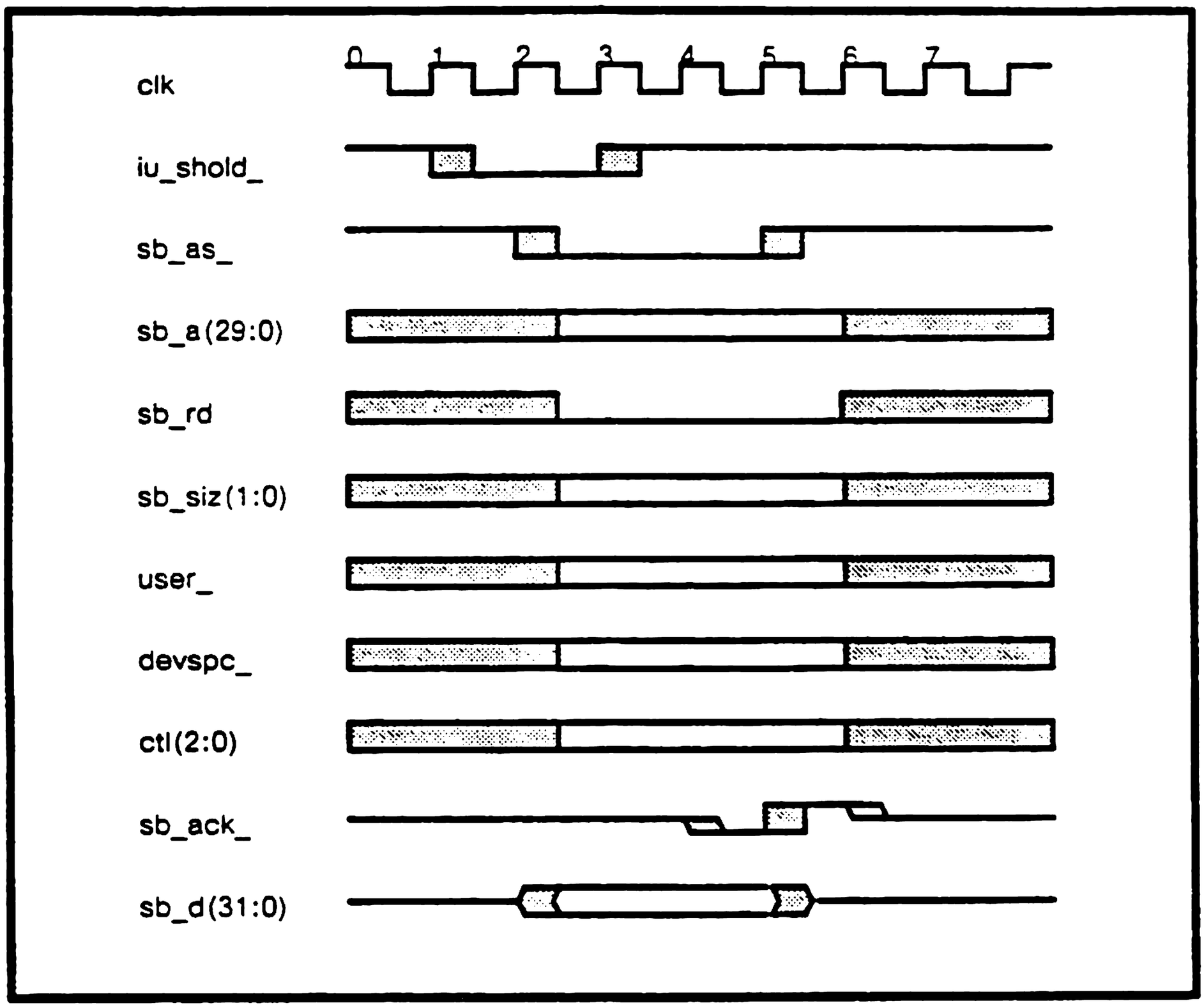
SBus Standard Writes

The IU is held starting when the miss is detected and ending when the acknowledge is received. All writes are implemented as standard writes except those to Device Spaces Type 0 and Type 1 as defined by the type bits from the MMU. Write data is available on the SBus on the rising edge at which AS_ is sampled true, and one clock later on the IOD bus.



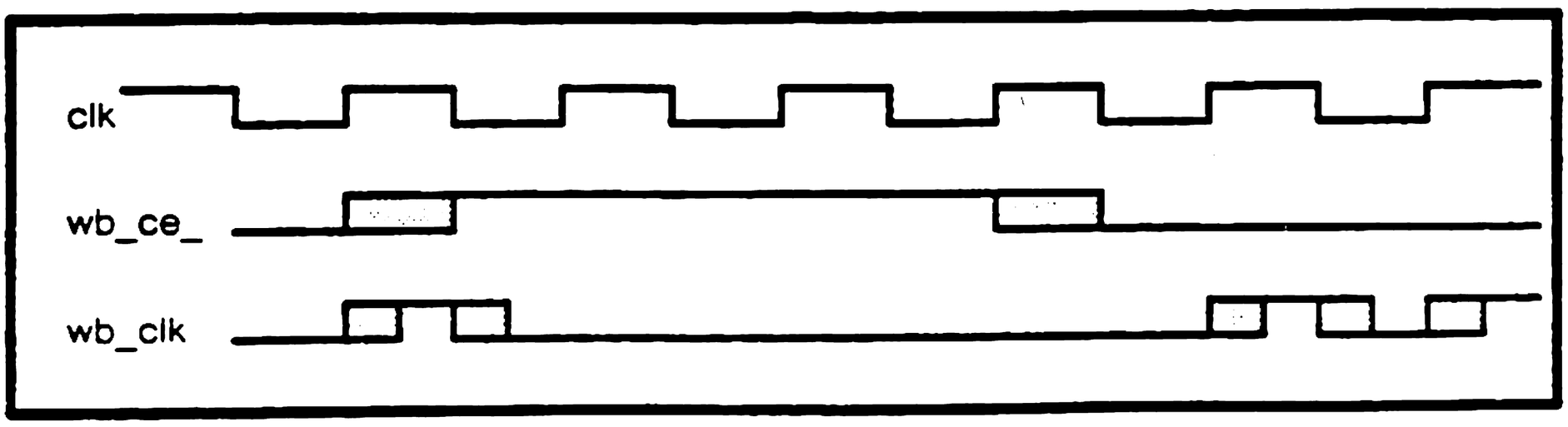
SBus Buffered Writes

The S4-Cache chip performs buffered writes to Type 0 and Type 1 Spaces using the write buffer in the S4-Buffer chip. The IU is held starting when the miss is detected and ending when the MMU has been checked. This occurs invisibly to the SBus, where the buffered write is indistinguishable from a standard write. Write data is available on the SBus on the rising edge at which AS_ is sampled true, and on the IOD bus one clock later.



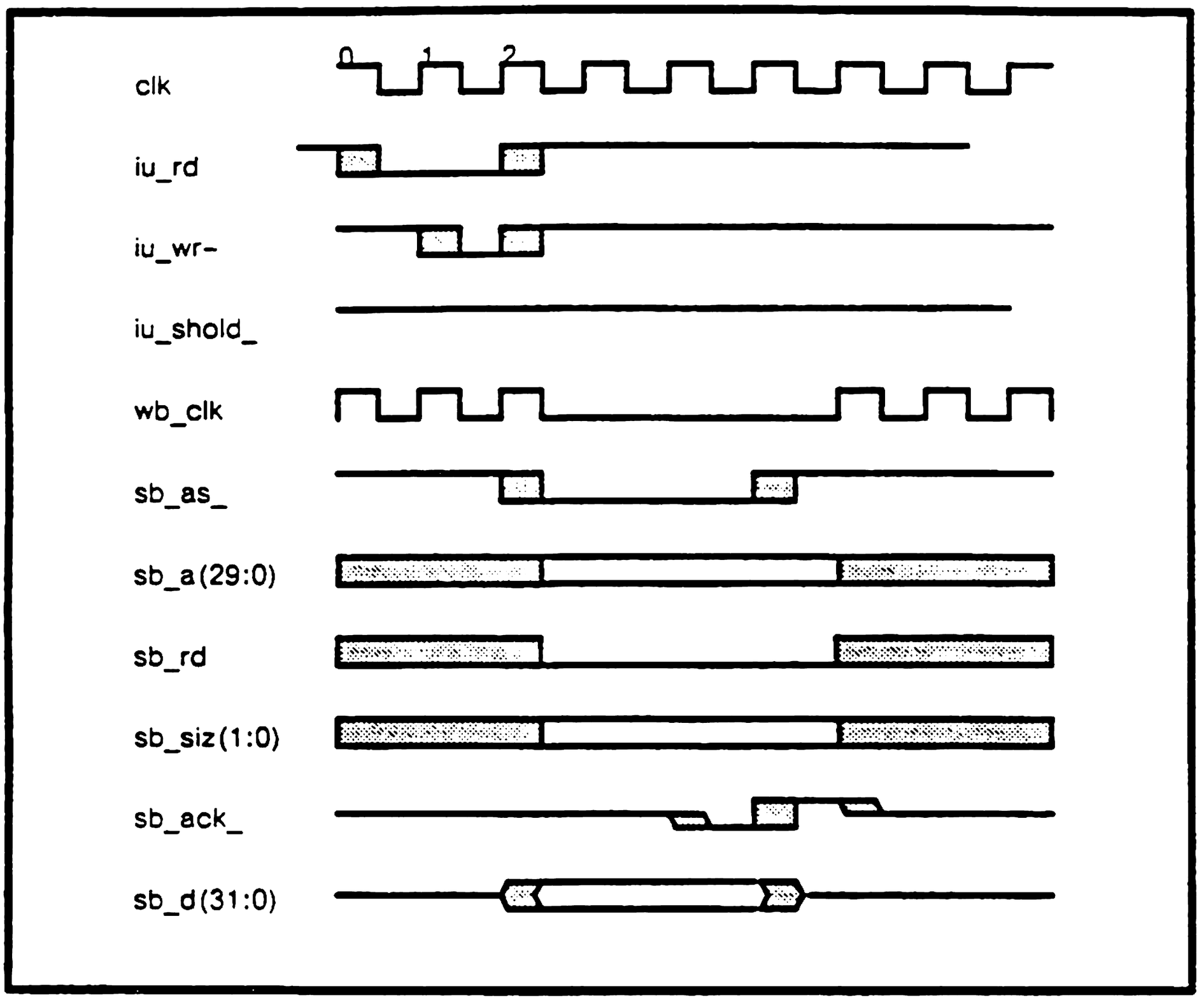
WB_CE_ Function

The WB_CE_ signal goes to the S4-Buffer chip, where it is used to generate the clock to the write buffer as shown in the following diagram:



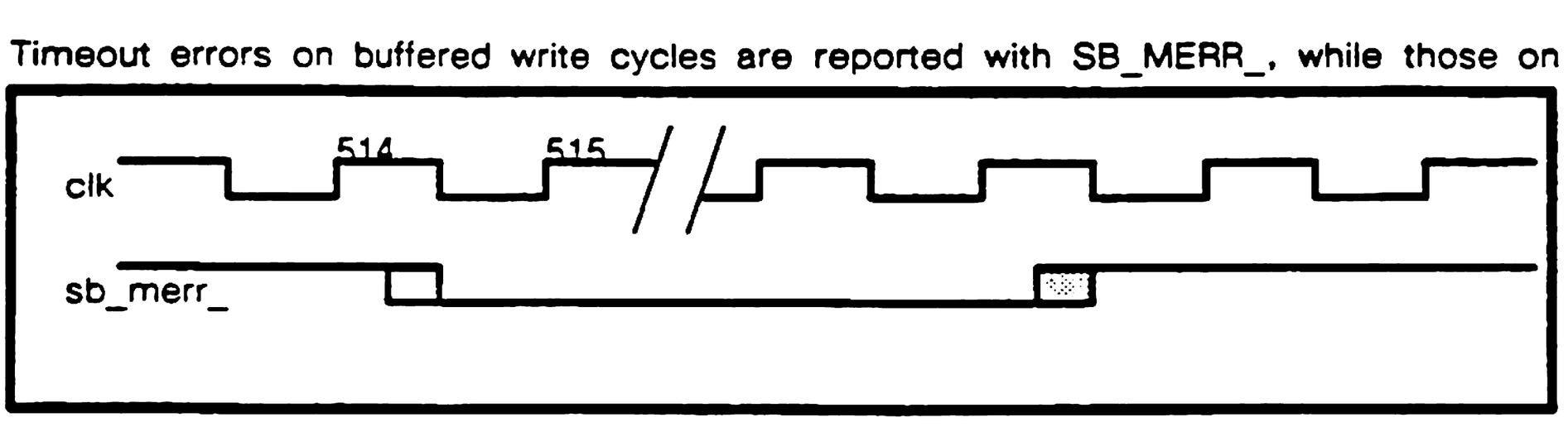
Cache Write Hits

On cache write hits the cache data is updated and an SBus write cycle is performed. The S4-Clock chip receives the CDWE_EN_ signal and generates four byte writes (CD_WE(3:0)) to the cache data RAMs. These four signals depend on the low-order address bits and the size bits so that only the appropriate bytes in the cache are updated.



Timeouts on Writes

Timeout errors on buffered write cycles are reported with SB_MERR_, while those on



standard write cycles are reported with IU_MEXC_.

Dynamic Bus Sizing

Byte Packing

To execute code contained in 8-bit devices on either the SBus or the IOD bus, the S4_Cache chip must pack the bytes up to fit the word length of the SPARC chip, as instruction fetches assume this data width. The S4-Cache chip transforms the SPARC data bus into a dynamically-sized bus somewhat like that of the Motorola 68020. The number of bytes involved in the first cycle is encoded on the three SB_SIZ signals. The current slave device responds with its port width encoded on the two SB_ACK signals. An IU word-length access will be converted into the appropriate number of shorter accesses if the accessed device indicates its port width is less than 32 bits.

Transfer Size Encoding

sb_siz2	sb_siz1	sb_siz0	Transfer Size
0	0	0	4 Bytes
0	0	1	1 Byte
0	1	0	2 Bytes
0	1	1	Not Used
1	0	0	16-Byte Burst
1	0	1	Not Used
1	1	0	Not Used
1	1	1	Not Used

Although the SBus specification allows 3-byte operations, none will be generated by the S4-Cache chip because all SPARC transfers are aligned.

Acknowledge Decoding

sb_ack32_	sb_ack8_	sb_err_	Definition
N	N	N	Insert Wait States
N	N	A	Error
N	A	N	8-bit port ack.
N	A	A	Rerun
A	N	N	32-bit port ack.
A	N	A	Reserved
A	A	N	16-bit port ack.
A	A	A	Reserved

Note: A: Asserted
N: Negated

Controller Response to Port Size

If the port acknowledge is narrower than the width of the current transfer on the SBus. the bus controller inside the S4-Cache chip will automatically perform the proper number of shorter cycles required to equal the longer cycle. The following table lists the possible combinations.

Transfer Size	Port Size	Controller Response
1-Byte	Any	Single BYTE cycle
2-Byte	8-bit	Two BYTE cycles
"	16-bit	One HALF cycle
"	32-bit	One HALF cycle
4-Byte	8-bit	Four BYTE cycles
"	16-bit	Two HALF cycles
"	32-bit	One WORD cycle

DMA Cycles

Bus Arbitration

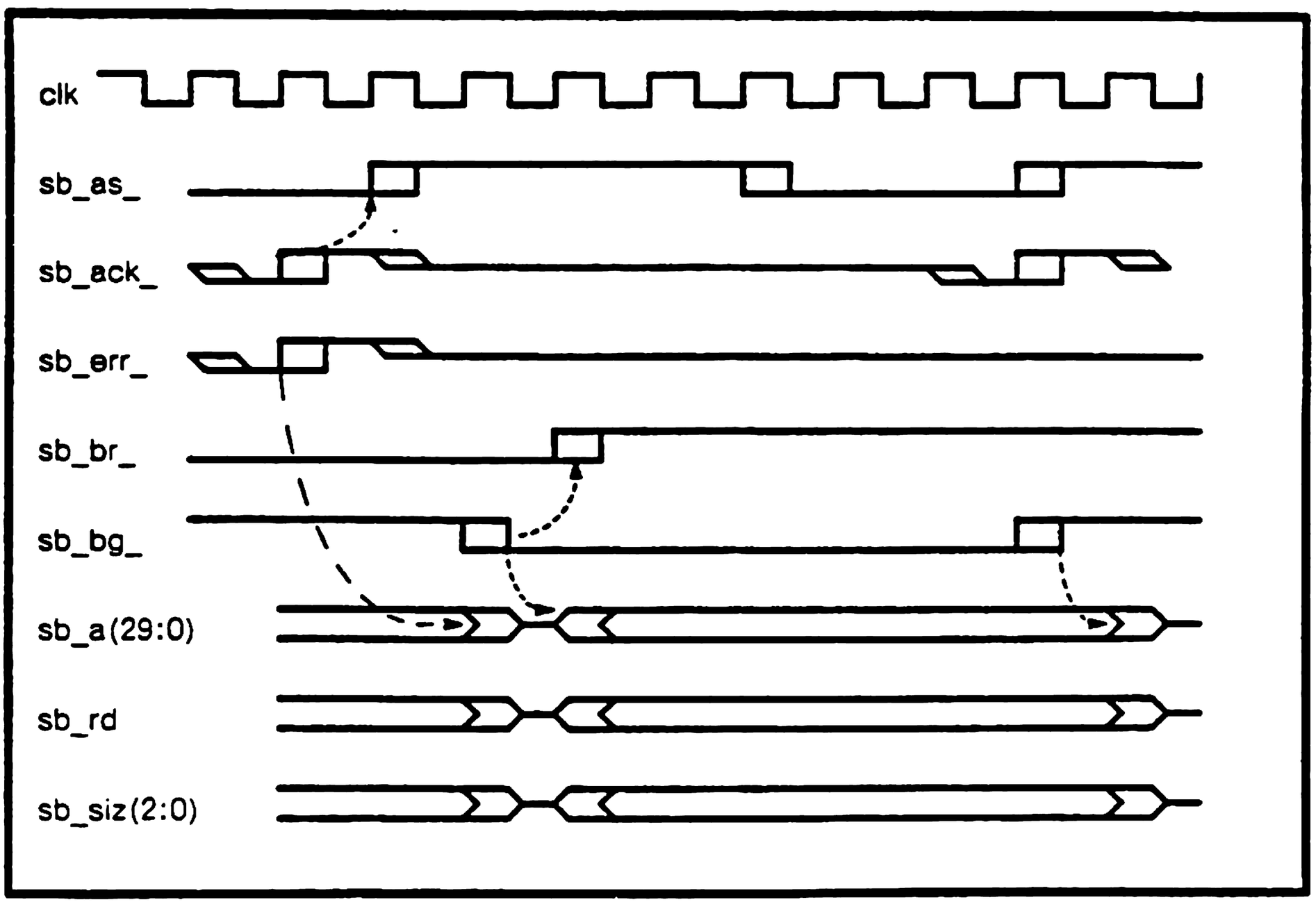
The S4-Cache chip receives three levels of DMA bus request {SB_BR(2:0)-} and generates three corresponding levels of bus grants {SB_BG(2:0)-}. In case more than one bus request is received simultaneously, the bus request priorities are as follows:

IU Write Hits	Highest Priority
SB_BR0_	
SB_BR1_	
SB_BR2_	
IU Misses	Lowest Priority

If a bus request is pending at the end of a DMA cycle, the bus arbiter will use a round-robin bus grant scheme so that all DMA masters can share equal bus bandwidth.

Rerun Cycles

The S4-Cache chip implements a rerun protocol that causes the current SBus cycle to be aborted and restarted later. This allows resolution of deadlocks between the IU and DMA, and allows SBus slaves to have long read latency without locking out DMA.



Deadlocks can occur when a single functional module is capable of being both a SBus slave and a DMA master. Such a module typically selects either its master or slave mode.

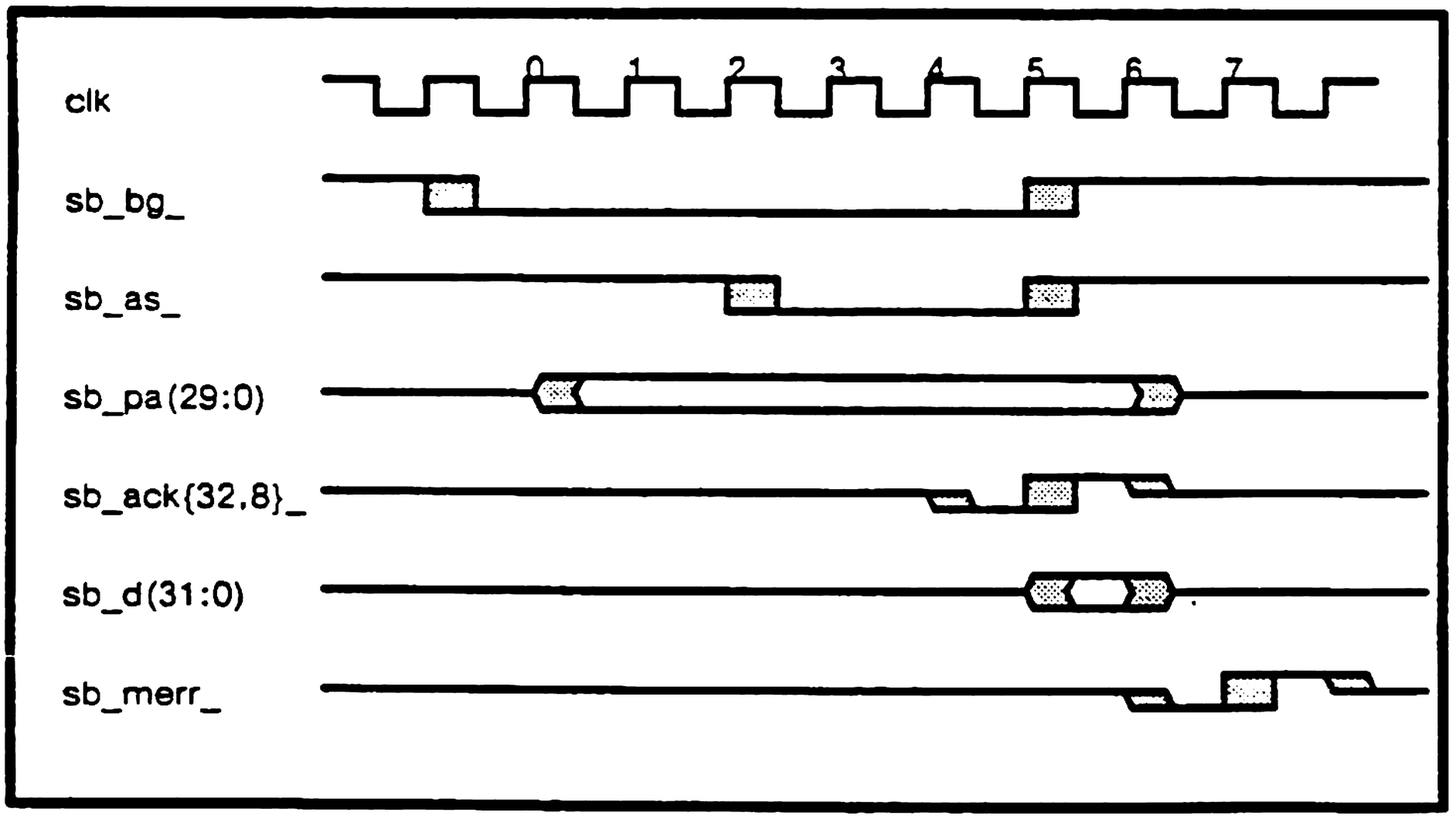
and a deadlock happens when the IU attempts to access the module after the module has switched to its master mode. The module must detect this condition and assert a Rerun Acknowledge (SB_ERR_ and SB_ACK32_ or SB_ACK8_, as shown in table above) along with a Bus Request. The S4-Cache chip will respond by negating the SBus address strobe, tri-stating the SBus, and issuing a bus grant. Standard SBus protocols will be observed after this point.

A device with a latency greater than the SBus timeout period must respond before the timeout with a Rerun Acknowledge. The current cycle will be aborted, any pending DMA requests will be serviced, and the cycle will be restarted. It is the responsibility of slave devices with long latency to latch the addresses and data, as during the rerun period these will no longer be valid.

A maximum of 64 rerun acknowledges are allowed for a single attempted access. After 64 reruns, SB_ERR_ will be asserted by the S4-Cache chip at State 3 to end the cycle.

DMA Parity Errors

Parity errors during DMA cycles are reported by the S4-Buffer chip to the SBus via SB_MERR_. The S4-Cache chip reports these errors to the IU via a bit in the Asynchronous Error Register, while the S4-MMU chip converts SB_MERR_ into a level on the IU Interrupt Request pins. Note that the SB_MERR_ signal can come up to two clocks after SB_BG_ is negated at the end of the cycle.



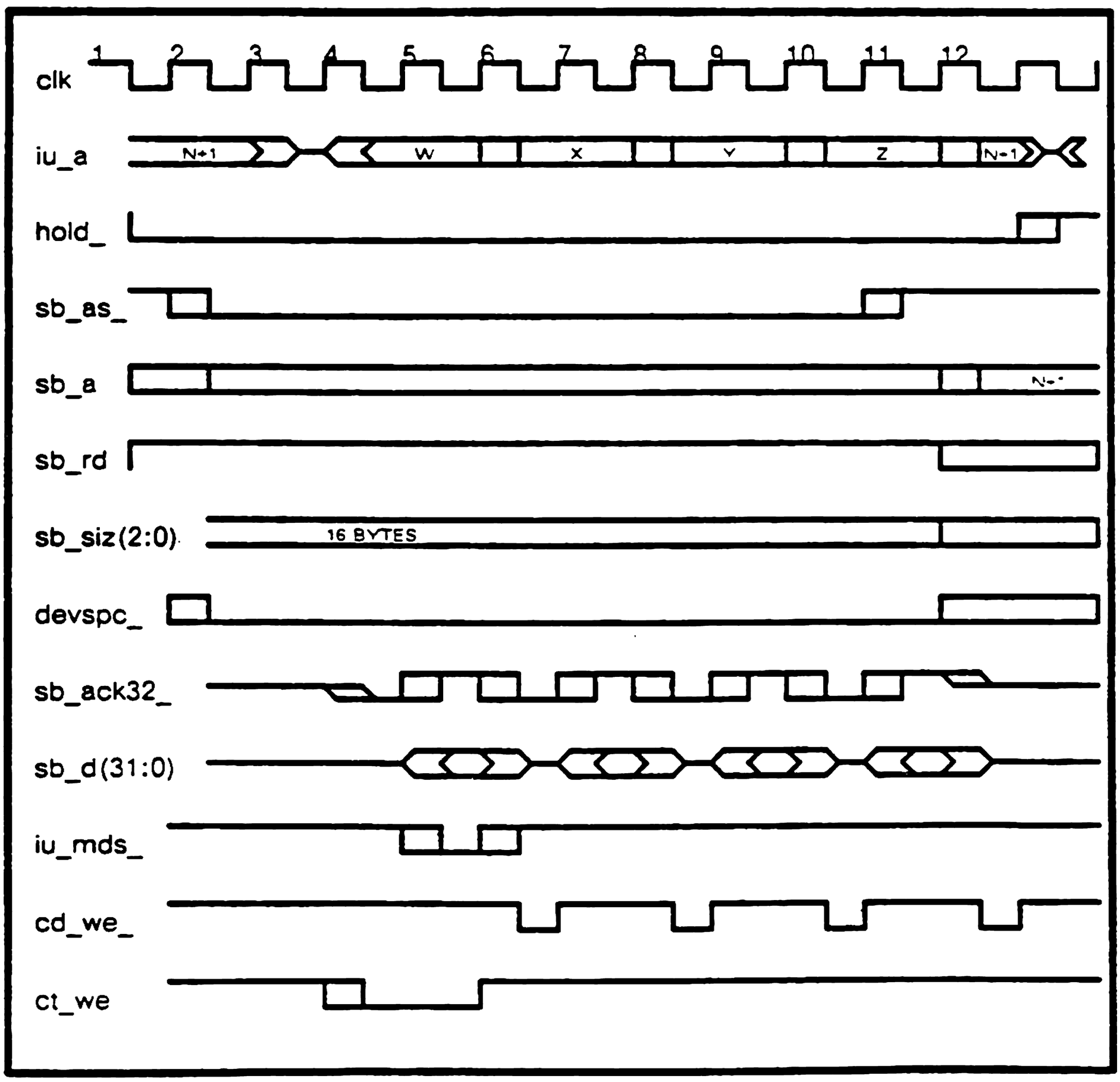
Cache Fills

The cache is filled under the following conditions:

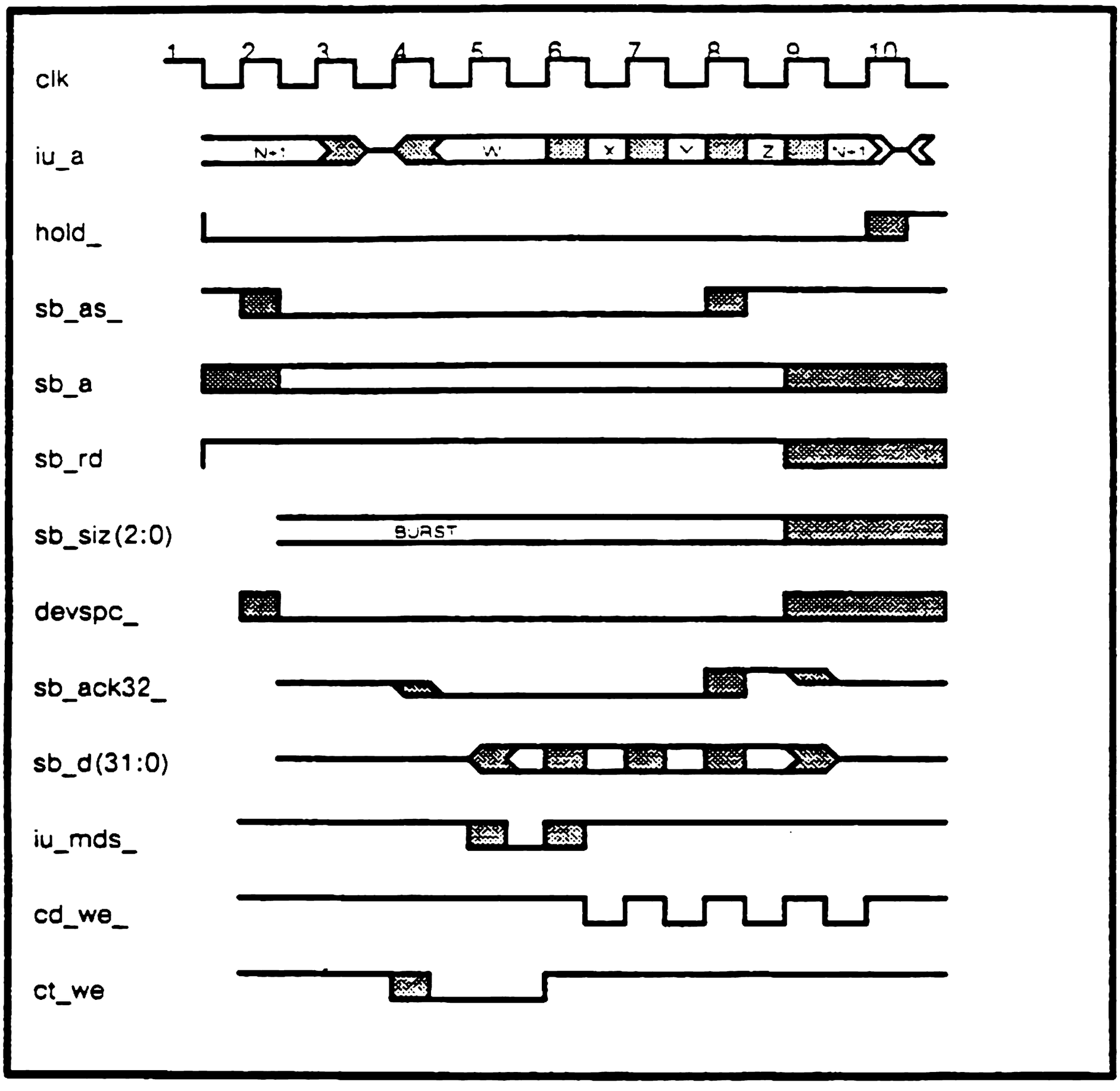
Read cycle &
Device space &
Page is marked cacheable (!MMU_X) &
EN_CACHE bit in System Enable Register is set &
No protection error is detected.

A cache fill cycle consists of four 32-bit reads of main memory. As the cache controller is capable of accepting an acknowledge on every clock, the four reads will typically be done using a high-speed burst mode access of the main RAMs. After the first acknowledge the bus controller will strobe the data into the IU, making the assumption that the memory provides the requested word first rather than providing the first word in the line.

Cache Fill with Non-Continuous ACKs



Cache Fill with Continuous ACKs

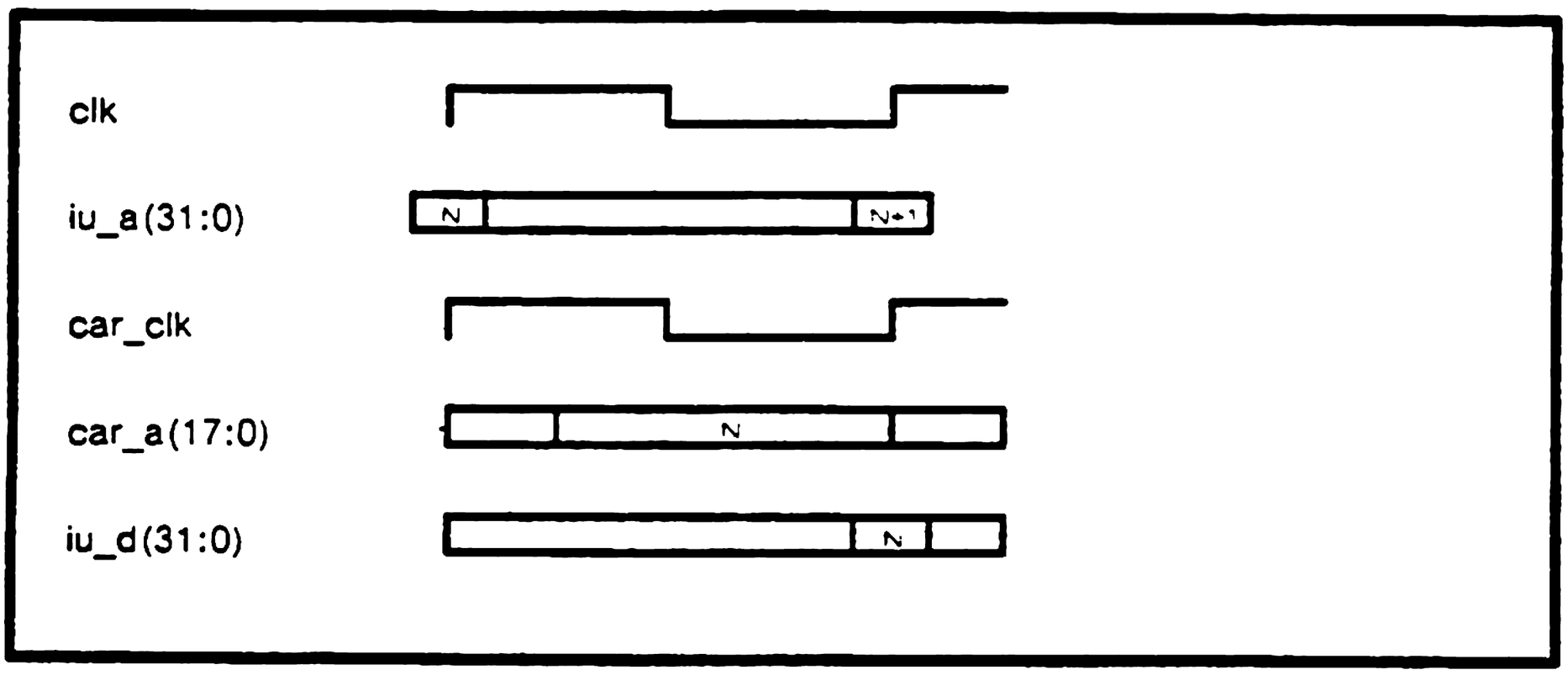


Cache Hits

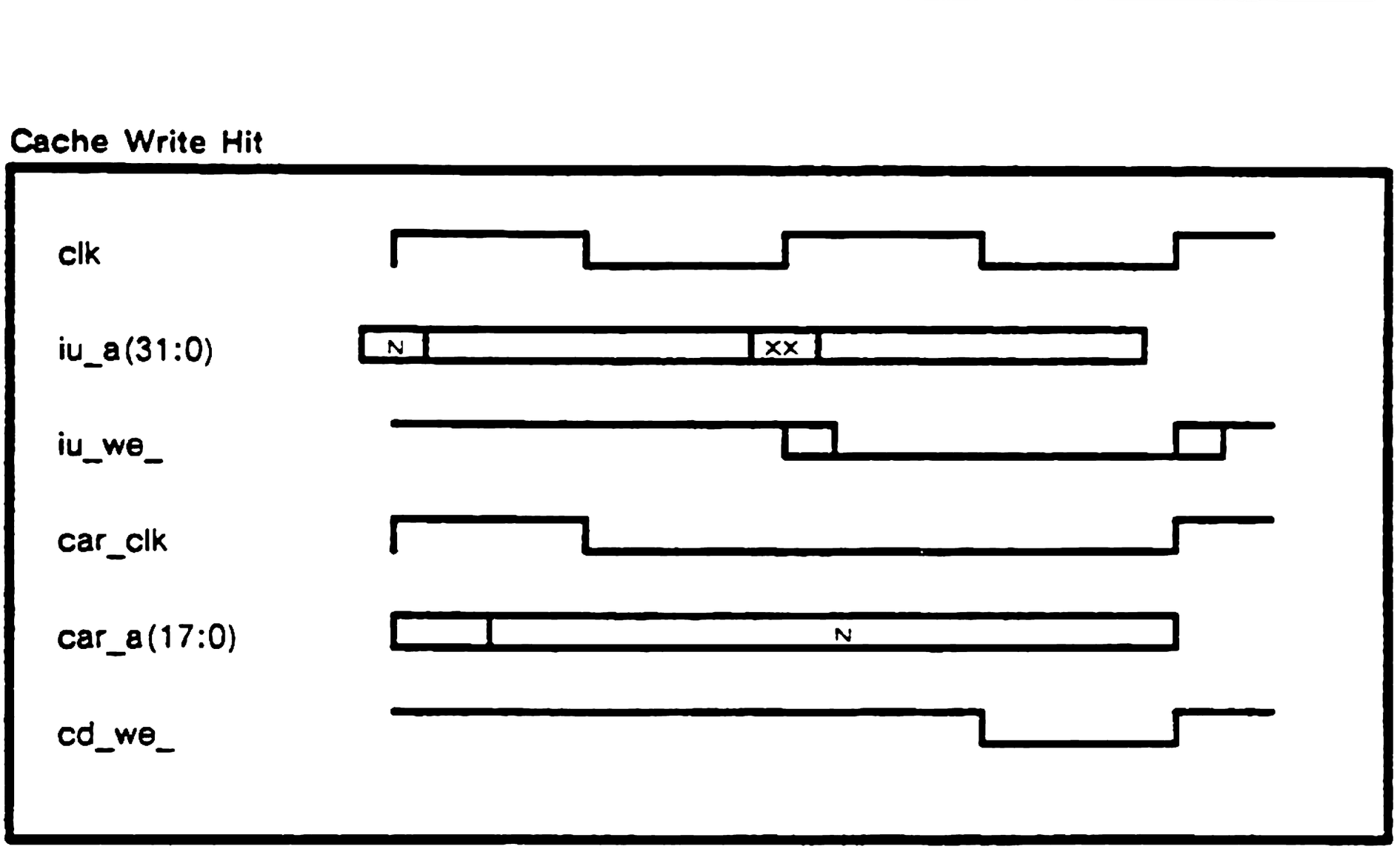
A cache hit occurs under the following conditions:

Device space &
CT_V high (cache tag is valid) &
CT_A(29:16) == latched IU_A(29:16) &
IU_A(31) == IU_A(30) == IU_A(29) &
{CT_S & Supervisor cycle} OR {!CT_S & CT_C(3:0) == CID(3:0)} &
{IU_RD OR (CT_WA & !Store double & SBus Idle)}

Cache Read Hit



Cache Write Hit



Cache Flushing

Three cache flush operations are defined, depending on the match criteria: the context match flush, the page match flush, and the segment match flush. A write operation tc the addresses specified in the address map below will flush one cache line if the tag satisfies the specified match criteria.

Context Match Flush Criteria:

```
CT_C(3:0) == CID(3:0)  &
!CT_S
```

where CID(3:0) is the current contents of the on-chip context register.

A context flush is used to ensure cache addressing consistency whenever a new active context replaces an old context in the MMU.

Page Match Flush Criteria:

```
CT_A(29:16) == SB_A(29:16) &
CT_S OR {!CT_S & CT_C(3:0) == CID(3:0)}
```

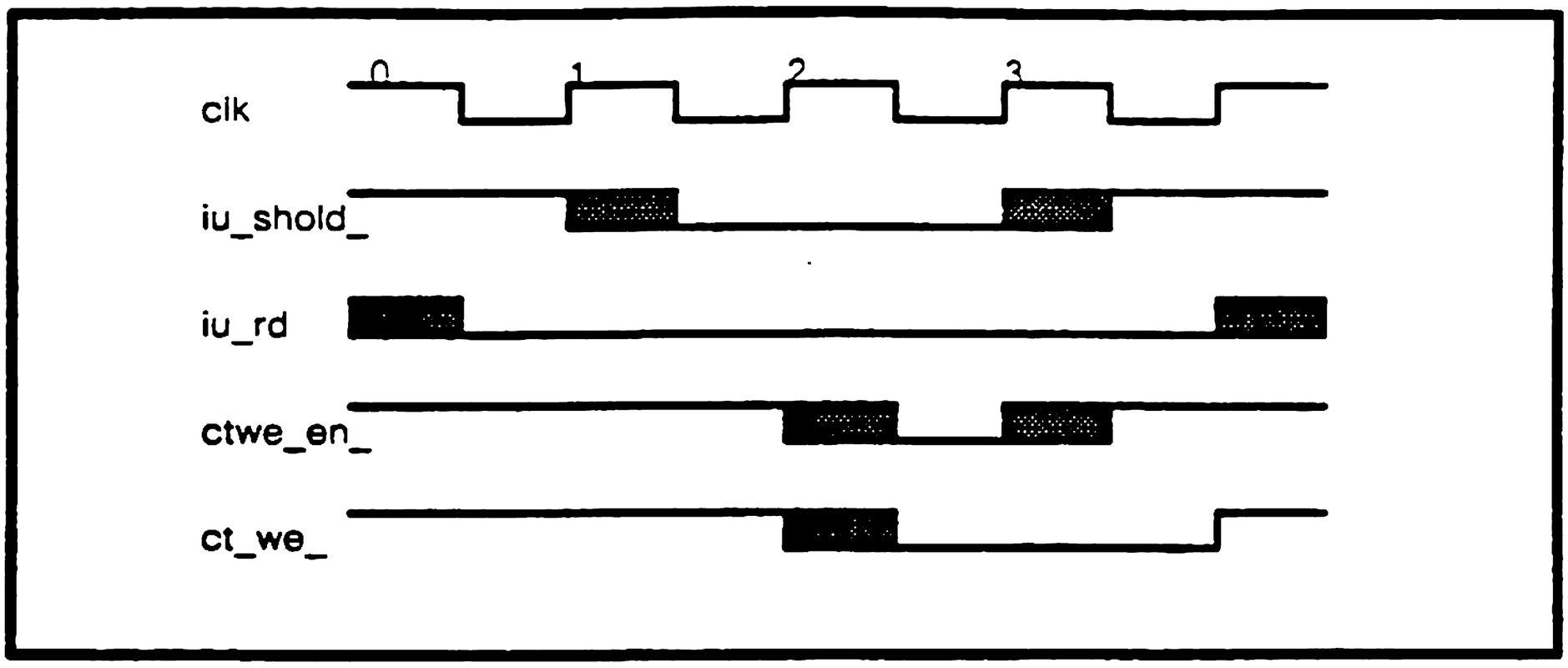
The page flush is used during page management to purge all references to a virtual page from the cache. Flushing an entire page requires 512 page flush commands to cover all combinations of A12..A4 that correspond to the desired page.

Segment Match Flush Criteria:

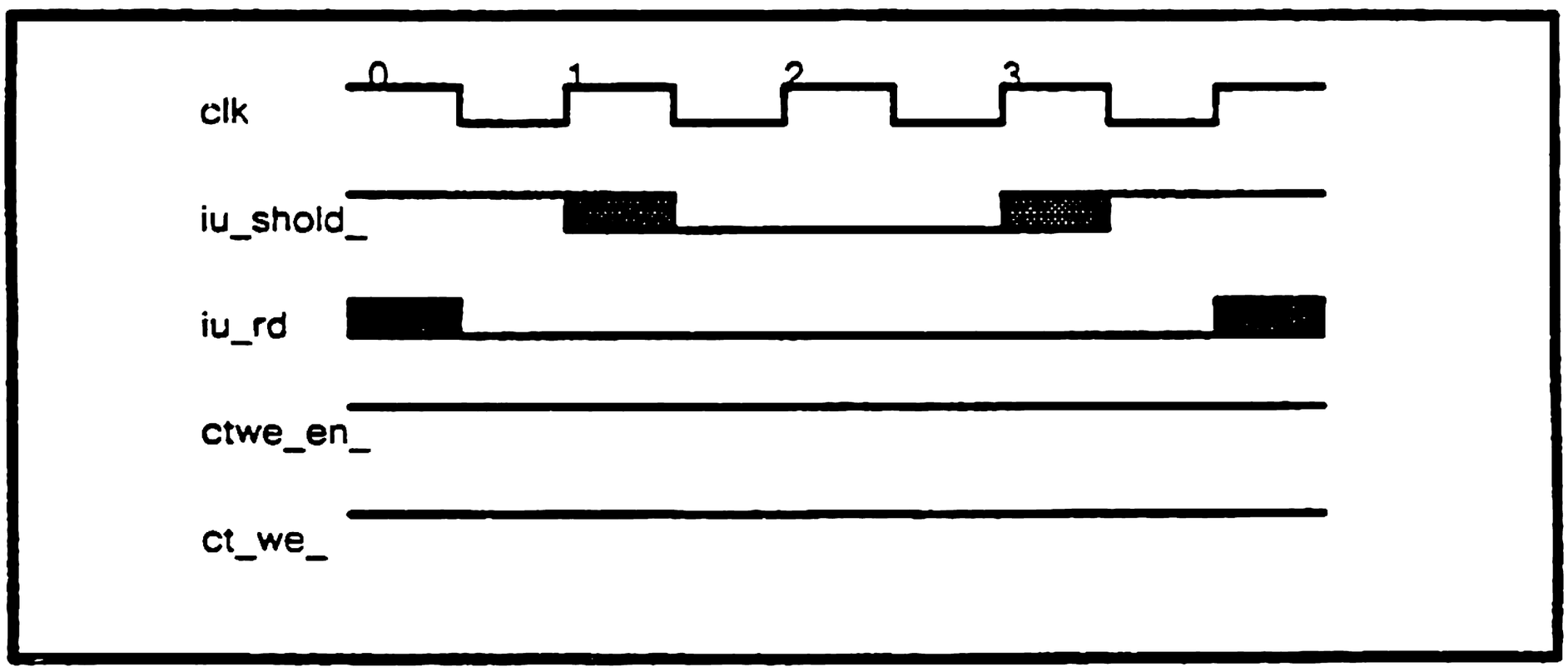
```
CT_A(29:18) == SB_A(29:18) &
CT_S OR {!CT_S & CT_C(3:0) == CID(3:0)}
```

The segment flush is used during page management to purge all references to a virtual segment from the cache. It is required whenever an active Page Map Entry Group must be replaced.

Cache Flush Satisfying Match Criteria



Cache Flush Not Satisfying Match Criteria



Miscellaneous Functions

Bus Timeouts

A module inside the S4-Cache chip monitors the duration of all address strobes. After 256 clocks it generates an SB_ERR_ signal on IU Read Cycles, IU Standard Write Cycles, and DMA Cycles, or an SB_MERR_ signal on IU Buffered Writes and IU Cache Write Hits. If an SBus slave needs longer latency than this allows, it must use the Rerun Acknowledge mechanism. A timeout on any DMA cycle will cause an SB_ERR_ signal.

A separate module monitors the number of Rerun Acknowledges that occur for a single attempted access. After 64 have occurred, it will generate an SB_ERR_ signal on IU read cycles or an IRQ_ signal on IU writes.

Reset Generation

Conditions causing SB_RESET_:

- POR_ (synchronized internally)
- IU asserts IU_ERROR_ due to multiple memory exceptions
- Low-to-high transition on RESET bit of System Enable Register.

All SB_RESET_ pulses are at least 512 clocks long. SB_RESET_ will be asserted for 512 clocks after POR_ is negated.

Address Map

Device Space and Control Space

The SPARC address space identifiers are divided into two "spaces" according to the following table: The signal DEVSPC_ chooses between device space and control space address maps. Device space devices are accessed with physical addresses provided by the MMU, while control space devices are accessed with virtual addresses on the SBus.

ASI	Function	Space
0-1	Reserved	Control
2	IU Extensions	Control
3	Segment Map	Control
4	Page Map	Control
5-7	Reserved	Control
8	User Instruction	Device
9	Supervisor Instr.	Device
A	User Data	Device
B	Supervisor Data	Device
C	Segment Flush	Control
D	Page Flush	Control
E	Context Flush	Control
F	Reserved	Control

IU Extension Address Map

The IU Extensions are accessed with ASI(2:0) == 2 and IU_A(31:28) as shown in the following table.

iu_a(31:28)	Device
0-2	Reserved
3	Context Register
4	System Enable Register
5	Reserved
6	Bus Error Registers
7	Reserved
8	Cache Tags
9	Cache Data
A-C	Unused
D	Reserved
E	VME Interrupt Acknowledge
F	Serial Controller Chip (MMU Bypass)

CTL(2:0) Encoding

Addresses for Control Space devices external to the S4-Cache chip are uniquely encoded on CTL(2:0) as shown in the following table. This minimizes the number of wires required to communicate with these devices.

ctl(2:0)	Device
0	Device on S4-Cache Chip
1	VME Interrupt Acknowledge
2	Context Register
3	Diagnostic Register
4	Serial Controller Chip (MMU Bypass)
5	Segment Map
6	Page Map
7	EPROM (Boot Cycle, Supv. Instr. Fetch)

Registers

Shadow Context Register

The Shadow Context Register maintains a copy of the Context Register that is found in the S4-MMU chip. It is used internally to the S4-Cache chip in the cache hit comparator, the cache flush comparator, and the cache tag write data. It is cleared on SB_RESET_ and written simulteneously with the Context Register in the S4_MMU chip. It can be read only with 8-bit operations on an odd-byte location. The bits are assigned as follows:

Write:			
	D(31:28)	Unused	Read back as zeroes
	D(27:24)	CID(3:0)	Write Only
Read	D(23:20)	Unused	Read back as zeroes
	D(19:16)	CID(3:0)	Read Only

System Enable Register

The System Enable Register enables various system functions and allows booting. This register can be read and written under software control, but can only be accessed with 8-bit operations. All bits are initialized to zero by SB_RESET_. Bits are assigned as follows:

D(31)	EN_BOOT_	Enable Boot State
D(30)	Unused	
D(29)	EN.DVMA	Enable Direct Virtual Memory Access
D(28)	EN_CACHE	Enable Cache Fills & Hits
D(27)	Reserved	
D(26)	SWRESET	Software Reset.
D(25)	Reserved	
D(24)	Reserved	Reads back as zero. Write has no effect.

EN_BOOT_. Boot state (active low) forces all supervisor program fetches to the EPROM device independent of the setting of the memory management. All other types of references are unaffected and will be mapped as during normal operation of the processor.
EN_DVMA. This bit enables all DVMA, including on-board and off-board.
EN_CACHE. When this bit is cleared, no cache fills will be performed and all IU reads will miss.
SWRESET. A low-to-high transition on this bit will generate a SB_RESET_.

BUS ERROR REGISTERS

Four bus error registers are contained in the S4-Cache chip, located at the following addresses:

0x6000 0000	Synchronous Error Register
0x6000 0004	Synhronous Error Virtual Address Register
0x6000 0008	Asynchronous Error Register
0x6000 000C	Asynchronous Error Virtual Address Register

Synchronous Error Register

When a memory exception occurs during an IU Sbus cycle, the cause of the error is captured in the Synchronous Error Register. The address is simultaneously captured in the Synchronous Error Virtual Address Register. No other bus cycles alter the Synchronous Error Register, including DVMA cycles. Parity errors are reported during the current cycle and so cause a memory exception instead of a high-level interrupt. The Bus Error Register contains the sum of all the errors that have occurred since it was last cleared. It is cleared upon being read, and on SB_RESET_.

The bits are assigned as follows:

D(31:16)	Zeroes	
D(15)	SE_WRITE	1=error during write cycle.
D(14:8)	Zeroes	
D(7)	SE_INVALID	The MMU Valid bit was zero on a device space cycle
D(6)	SE_PROTERR	Protection Error
D(5)	SE_TIMEOUT	The bus timeout timer went off
D(4)	SE_SBERR	SBus Error
D(3)	SE_MEMERR	Memory Error (parity or ECC)
D(2)	Zero	
D(1)	SE_SIZERR	Size Error
D(0)	SE_WATCHDOG	The IU asserted IU_ERROR_ and was restarted

Synchronous Error Virtual Address Register

The Synchronous Error Virtual Address Register stores the 32-bit virtual address of a bus cycle that causes a memory exception. It will latch the first error that occurs and will stay latched until it is read. It is a R/W register, cleared on SB_RESET_ and when read.

Asynchronous Error Register

The Asynchronous Error Register records the cause of the first level 15 interrupt received since last time the AER was read. It is unlatched upon being read, and cleared on SB_RESET_.

The bits are assigned as follows:

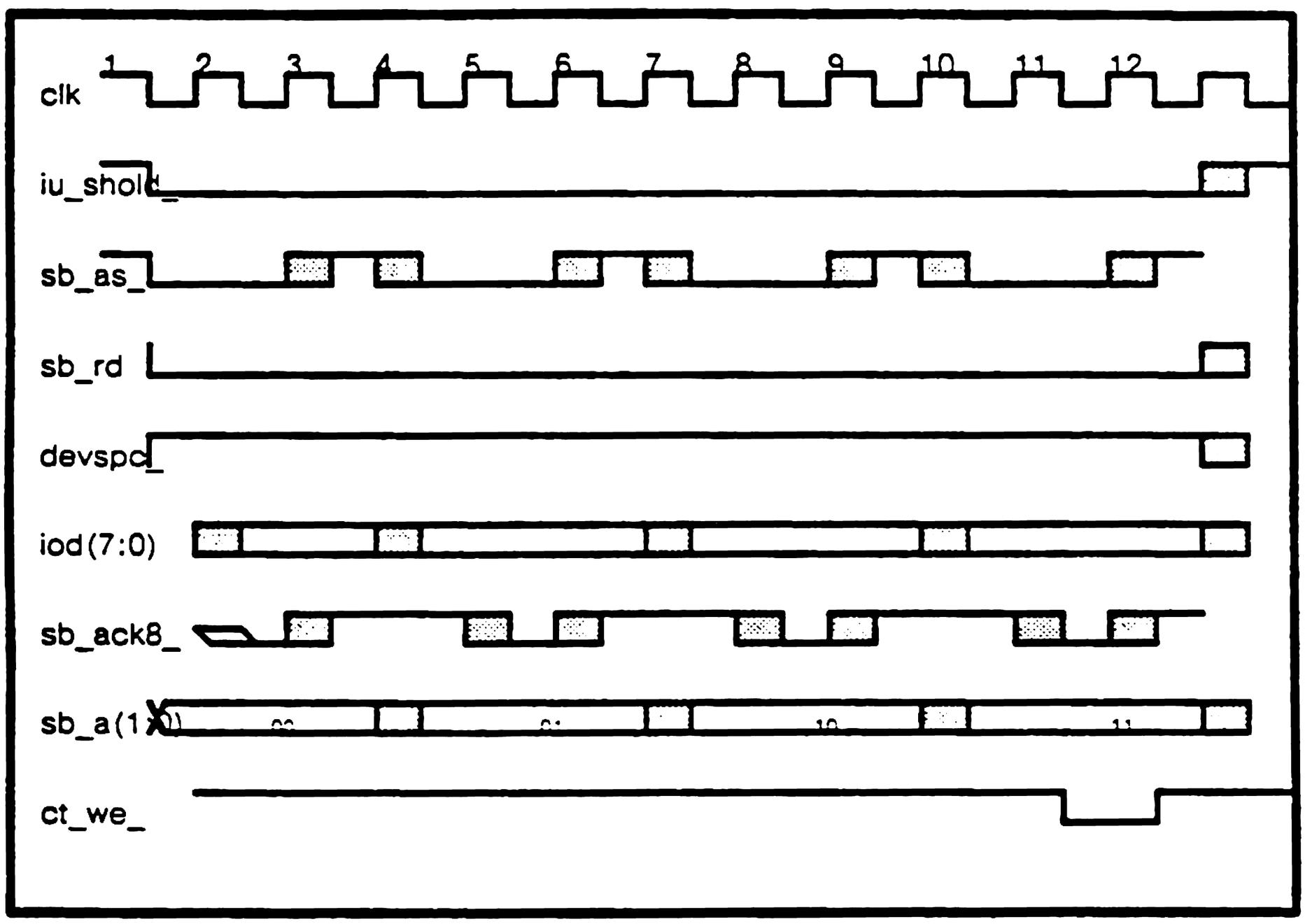
D(31:8)	Undefined	
D(7)	ASE_WBACKERR	1=Valid bit was zero in page map.
D(6)	Zero	
D(5)	ASE_TIMEOUT	1=Non-existent device was accessed.
D(4)	ASE_DVMAERR	1=SBus error during DVMA access.
D(3:0)	Zeroes	

Asynchronous Error Virtual Address Register

The Asynchronous Error Virtual Address Register latches the 32-bit address of the first asynchronous error that occurs since the last time the AEVAR was read. It is unlatched upon being read, and cleared on SB_RESET_. For errors that occur during DVMA, bits 31:30 are copies of bit 29.

Cache Tags

The cache tags are directly readable and writable in control space. Write cycles must be performed with 32-bit accesses only. Other widths during writes will cause a Size Error Memory Exception because the S4-Cache chip includes a byte packing register that demultiplexes the 8-bit IOD bus up to the 32-bit cache tag bus, and it can only operate four bytes at at time. The cache tags are not initialized in hardware, and so zeroes must be written to all CT_V bits before the cache is enabled. Cache tag direct reads make use of the standard byte-packing feature of the S4 chip set described earlier. The following diagram shows the operation of the cache tag byte packing register in the S4-Cache chip on a cache tag direct write:

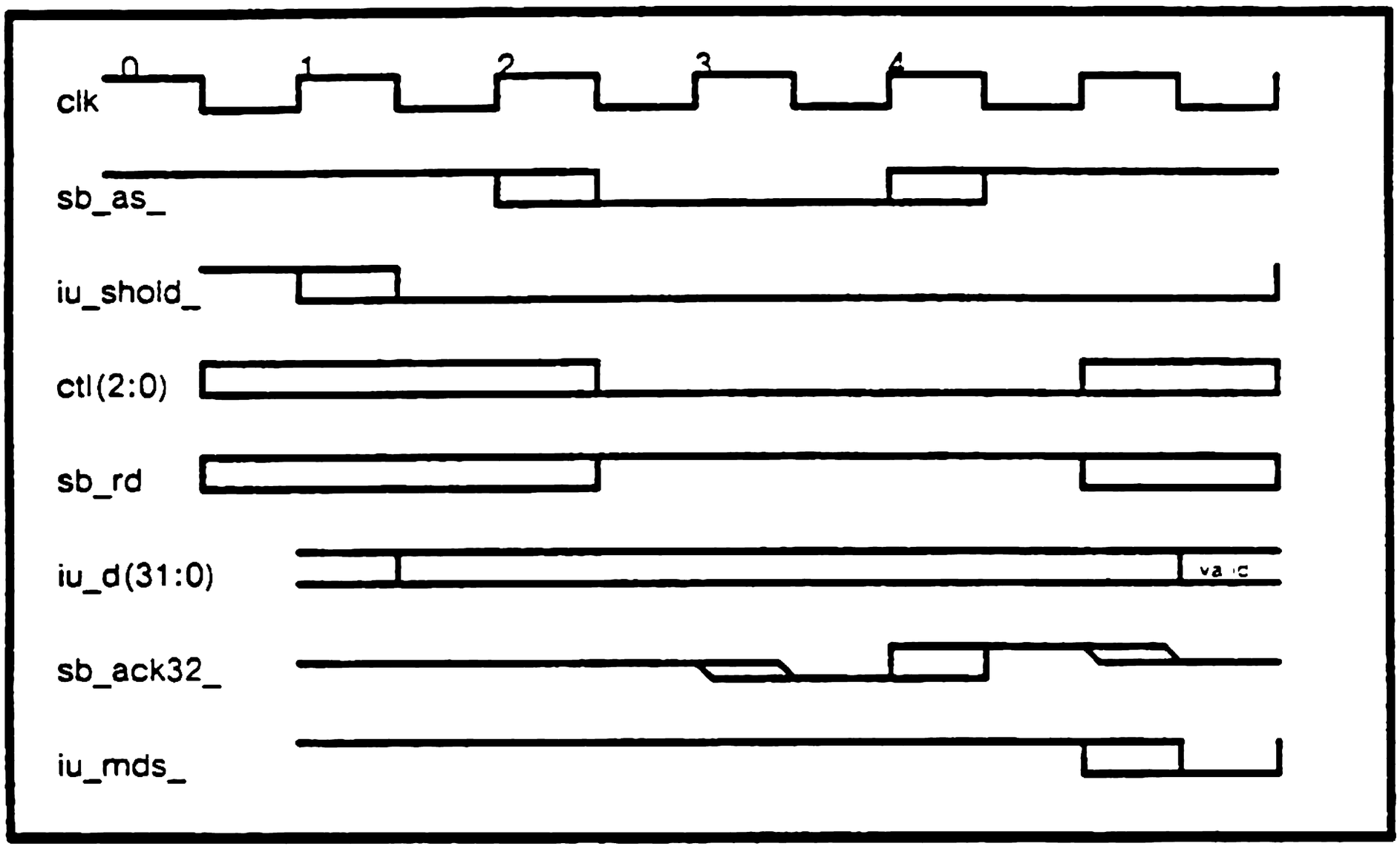


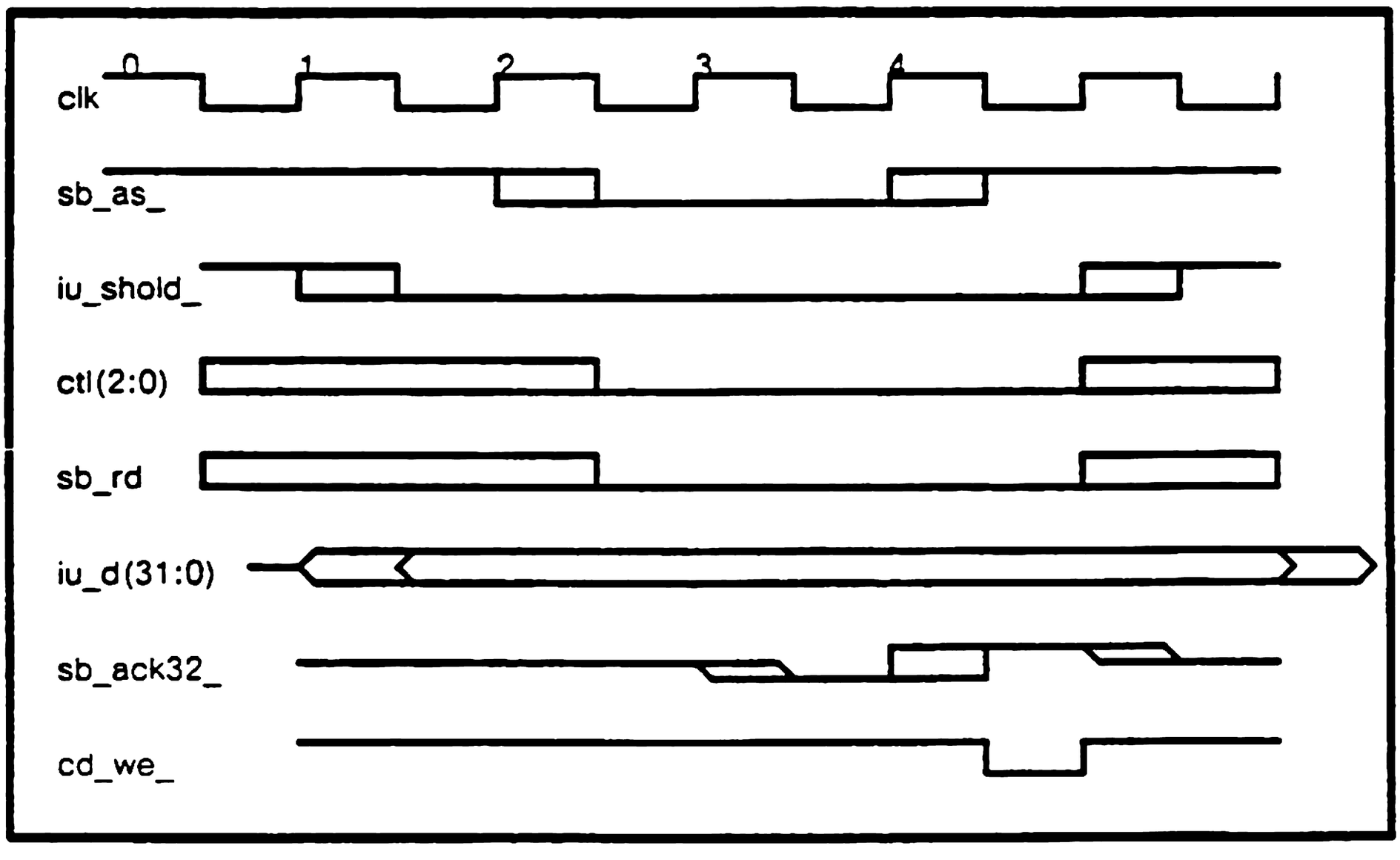
The format of the cache tags is as follows:

D(31:26)		Unused
D(25:22)	CT_C(3:0)	Cache Tag Context bits
D(21)	CT_WA	Write Allowed
D(20)	CT_S	Supervisor-only access protection bit
D(19)	CT_V	Cache Tag Valid
D(18:16)		Unused
D(15:2)	CT_A(29:16)	Virtual address bits A(29:16)

Cache Data

The cache data is also directly readable and writable in control space. Because the cache data RAMs directly drive the IU data bus, the data does not pass through the S4-Buffer chip as it does for all other types of accesses. **A Cache Data Control Space Read must not be immediately followed by a write cycle of any type.** The following diagrams show a direct read and a direct write of the cache data RAMs.





Timing Specifications

Output Delays

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C, Output Load=15 pF

Symbol	From	To	min	max	unit
t1	clk high	clk high	50	---	ns
t2	clk	iu_al		34.4	ns
t3	clk	iu_aoe_		17.8	ns
t4	clk	iu_mds_		17.1	ns
t5	clk	iu_mexc_		17.4	ns
t6	clk	iu_mhold_		16.3	ns
t7	clk	sb_a (untransl.)		41.8	ns
t8	clk	sb_a (seg. map)		30.3	ns
t9	clk	sb_ack32_		24.4	ns
t10	clk	sb_ack8_		24.4	ns
t11	clk	sb_as_		27.7	ns
t12	clk	sb_bg_		19.8	ns
t13	clk	sb_err_		26.1	ns
t14	clk	sb_merr_		24.7	ns
t15	clk	sb_rd		29.1	ns
t16	clk	sb_reset_		23.0	ns
t17	clk	sb_siz		36.9	ns
t18	clk	car_en_		16.7	ns
t19	clk	cd_oe_		22.0	ns
t20	sb_rd_	cd_oe_		16.1	ns
t21	iu_rd_	cd_oe_		11.8	ns
t22	clk	cdwe_en_		15.6	ns
t23	clk	ct_a		33.7	ns
t24	clk	ct_c		24.9	ns
t25	clk	ct_s		24.7	ns
t26	clk	ct_v		24.8	ns
t27	sb_rd	ct_v		12.5	ns
t28	clk	ct_wa		24.7	ns
t29	clk	ctl		16.0	ns
t30	clk	ctwe_en_		18.2	ns
t31	clk	devspc_		16.8	ns
t32	clk	io_d		74.5	ns
t33	sb_a	io_d		41.0	ns
t34	clk	user_		17.1	ns
t35	clk	wb_ce_		16.0	ns
t36	clk	wb_oe_		16.4	ns

Input Setup Requirements

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C, Output Load=15 pF

Symbol	From	To	min	max	unit
t37	iu_ah	clk	-1.1		ns
t38	iu_al	clk	-1.2		ns
t39	iu_asi	clk	10.0		ns
t40	iu_error_	clk	5.3		ns
t41	iu_hal_	clk	5.3		ns
t41	iu_ldst	clk	9.5		ns
t43	iu_nulcyc	clk	0.6		ns
t44	iu_rd	clk	2.5		ns
t45	iu_siz	clk	10.0		ns
t46	iu_we	clk	11.2		ns
t47	sb_a	clk	10.5		ns
t48	sb_ack_	clk	7.3		ns
t49	sb_err_	clk	10.8		ns
t50	sb_merr-	clk	4.5		ns
t51	sb_rd	clk	18.4		ns
t52	ct_a	clk	9.7		ns
t53	ct_c	clk	9.7		ns
t54	ct_s	clk	3.3		ns
t55	ct_v	clk	2.6		ns
t56	ct_wa	clk	2.9		ns
t57	io_d	clk	10.6		ns
t58	mmu_{s,typ,v,w,x}	clk	5.5		ns

Change History

2/1/88

Sunray support--	Removed.
Hardware Cache Consistency--	Removed.
SB_ACK @ State 4--	Removed restriction of no ACKs before state 5.
Cache filling--	Removed restriction to Type 0 Space. Added requirement of IMMU_X.
Cache Hit definition--	Added term for write hits.
Context Flush criteria--	Fixed bug in CT_S polarity.
Table of Contents--	Added.
Timing Specifications--	Added a few.

7/18/88

Cleaned up errors everywhere.....

Timing--	Added many new timing specs. Used post-route timings.
Reruns--	SB_AS_ is negated one clock later than prev. spec.
Cache Hits--	Changed definition of cache hit on page 19.
Cache Flushing--	Removed notes about flushes before changing MMU. Modified timing diagrams; IU_SHOLD_ for 2 clocks.
Bus Error Registers--	Added SER, SEVAR, AER, AEVAR definitions.
Cache Data--	Added restriction: no write after control space read.

Errata

7/18/88

DMA Timeouts:

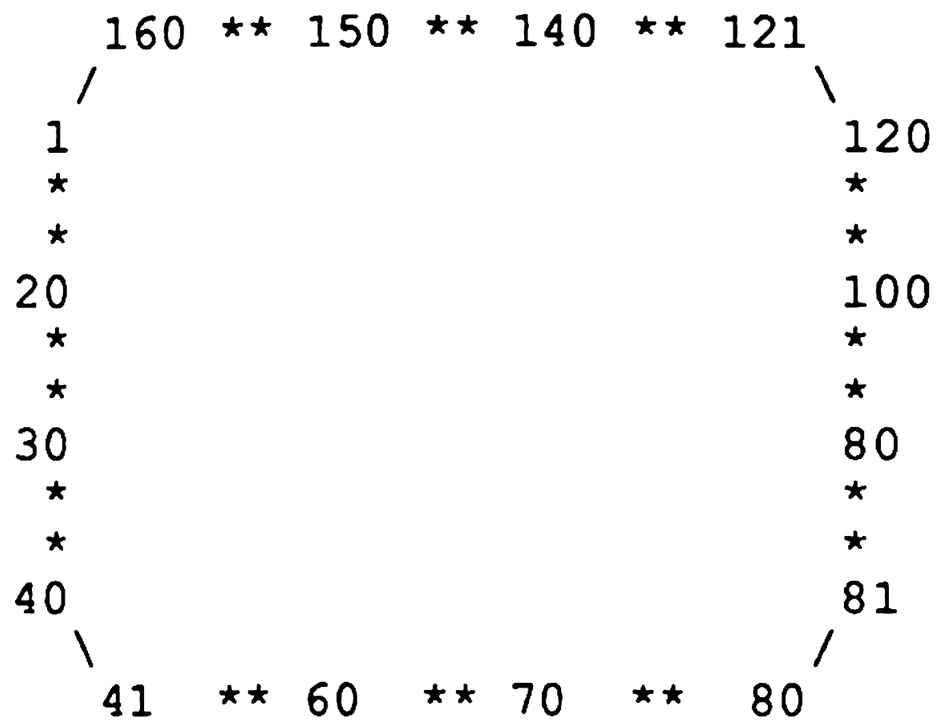
Timeouts that terminate DMA cycles will cause the TO_ERR bit in the Synchronous Error Register will be set incorrectly.

1. P/N: 100-1810-06 2. REV: 01

3. DESCRIPTION: IC,GA,S4-CATCH

4. PHYSICAL MODEL: (160 Pin PFP)

A. Pinout: *Top View



1 - CTL.1	41 - /IU /HAL	81 - /IU /AL.15	121 - /SB A.7
2 - CTL.0	42 - /IU /ASI.2	82 - /IU /AL.4	122 - /SB A.8
3 - /POR	43 - /IU /WE	83 - /IU /AL.1	123 - /SB /RESET
4 - /CT S	44 - /IU /SIZ.0	84 - /CE /OE	124 - /SB A.9
5 - /CT V	45 - /IU /RD	85 - /SB /MERR	125 - /SB A.10
6 - /CT WA	46 - /IU /ASI.0	86 - /IU /NULCYC	126 - /SB /BR.2
7 - /CT A.29	47 - /IU /ASI.1	87 - /SB /AS	127 - /SB A.11
8 - /CT A.28	48 - /IU /AL.0	88 - /S4C /TEST	128 - /SB A.12
9 - /CT A.27	49 - /IU /AL.12	89 - /S4C OE	129 - /SB /BR.1
10 - /CT A.26	50 - /IU /AL.8	90 - IOD.0	130 - /SB A.13
11 - /CT A.25	51 - /IU /AL.16	91 - IOD.1	131 - /SB A.14
12 - /CT A.24	52 - /IU /AL.14	92 - /WB /CE	132 - VSS
13 - /CT A.23	53 - /IU /AL.3	93 - IOD.2	133 - /SB A.15
14 - /CT A.22	54 - /IU /AL.11	94 - IOD.3	134 - /SB A.16
15 - /CT A.21	55 - /IU /AL.7	95 - IOD.4	135 - /SB A.17
16 - /CT A.20	56 - /IU /AL.6	96 - /WB /OE	136 - /SB A.18
17 - /CT A.19	57 - /IU /AL.17	97 - IOD.5	137 - /SB A.19
18 - /CT A.18	58 - /IU /A.23	98 - IOD.6	138 - /SB A.20
19 - VDD	59 - VDD	99 - IOD.7	139 - /SB A.21
20 - VSS	60 - VSS	100 - VDD	140 - /SB /BR.0
21 - CLKIN	61 - /IU /A.22	101 - VSS	141 - VSS2
22 - /CT A.17	62 - /IU /A.31	102 - /SB /BG.2	142 - /SB A.22
23 - /CT A.16	63 - /IU /A.19	103 - /SB RD	143 - /SB A.23
24 - /CT C.3	64 - /IU /A.21	104 - /SB /BG.1	144 - /SB A.24
25 - /CT C.2	65 - /IU /A.18	105 - /SB /SIZ.0	145 - /SB A.25
26 - /CT C.1	66 - /IU /A.28	106 - /SB /SIZ.1	146 - /SB A.26
27 - /CT C.0	67 - /IU /A.20	107 - /SB /BG.0	147 - /SB A.27
28 - /CAR /EN	68 - /IU /A.25	108 - /SB /SIZ.2	148 - /SB A.28
29 - /CDWE /EN	69 - VSS	109 - /SB A.0	149 - /SB A.29
30 - /CTWE /EN	70 - /IU /A.29	110 - /SB /ERR	150 - /DEVSPC
31 - /IU /ERROR	71 - /IU /A.30	111 - /SB A.1	151 - USER

32 - /IU /AOE	72 - /IU /A.24	112 - /SB A.2	152 - /MMU TY
33 - /IU /MHOLD	73 - /IU /A.27	113 - /SB /ACK32	153 - /MMU TY
34 - /IU /MDS	74 - /IU /A.26	114 - /SB A.3	154 - /MMU X
35 - /IU /ASI.3	75 - /IU /AL.13	115 - /SB A.4	155 - /MMU S
36 - /IU /MEXC	76 - /IU /AL.5	116 - /SB /ACK8	156 - /MMU W
37 - /IU /LDST	77 - /IU /AL.9	117 - /SB A.5	157 - /MMU V
38 - /IU /SIZ.1	78 - /IU /AL.2	118 - /SB A.6	158 - CTL.2
39 - VDD	79 - /IU /AL.10	119 - VDD	159 - VDD
40 - VSS	80 - VDD	120 - VSS	160 - VSS

B. PIN DESCRIPTION:
==========

SYMBOL	TYPE	DESCRIPTION
IU INTERFACE	50	
/IU CLK	DRVT8	Integer Unit Clock
/IU RD	TLCHTU	Integer Unit Read
/IU /WR	TLCHTU	Integer Unit Write
/IU SIZ(1:0)	TLCHTU	Integer Unit Size
/IU /ERROR	TLCHTNU	Integer Unit Error
/IU NULCYC	TLCHTD	Integer Unit Null Cycle
IU /HAL	TLCHTU	Integer Unit Hold Address Low
/IU IDST	TLCHTN	Integer Unit Lock
/IU ASI(3:0)	TLCHTU	Integer Unit Address Space Identifiers
/IU AH(31:18)	TLCHTU	Integer Unit High Addresses
/IU AL(17:0)	BD4TU	Integer Unit Low Addresses
/IU /SHOLD	BT8	Integer Unit Hold
/IU /MDS	BT4	Integer Unit Memory Data Strobe
/IU /MEXC	BT4	Integer Unit Memory Exeption
/IU /AOE	BT4	Integer Unit Address Output Enable
/POR	TLCHTNU	Power-On Reset
SBUS INTERFACE	46	
/SB /AS	BT4	Address Strobe
/SB /BR(2:0)	TLCHTNU	Schoolbus Bus Request
/SB /BG(2:0)	BT4	Schoolbus Bus Grant
/SB RD	BD4TU	Schoolbus Read
/SB SIZ(2:0)	BT4TU	Schoolbus Size
/SB A(29:0)	BD4TU	Schoolbus Address Bus
/SB /ACK8	BD8TU	SBus 8-bit Acknowledge
/SB /ACK32	BD8TU	SBus 32-bit Acknowledge
/SB /ERR	BD4TU	Schoolbus Error
/SB /MERR	BD8TU	Schoolbus Memory Error
/SB /RESET	BT8	SBus Reset
MMU INTERFACE	19	
/MMU S	TLCHT	MMU Supervisor Only protection bit
/MMU WA	TLCHT	MMU Write Allowed protection bit
/MMU X	TLCHT	MMU Don't Catche bit
/MMU V	TLCHT	MMU Valid bit
/MMU TYP(1:0)	TLCHT	MMU Type bits
/USER	BT4	User-Mode Address Space Identifiers
/DEVSPC	BT4	Device/Control Space
CTL(2:0)	BT4	Encoded Control space device IDs
IOD(7:0)	BD4TRU	Input/Output Data Bus

CATCHE INTERFACE	25	
/CT A(29:0)	BD4TU	Cache Tag Address bits
/CT C(3:0)	BD4TU	Catche Tag Context bits
/CT S	BD4TU	Cache Tag Supervisor
/CT V	BD4TU	Cache Tag Valid
/CT WA	BD4TU	Cache Tag Write Allowed
/CTWE /EN	BT4	Cache Tag Write Enable Enable
/CDWE /EN	BT4	Cache Data Write Enable Enable
/CD /OE	BT4	Cache Data Output Enable
/CAR /EN	BT4	Cache Address Register Clock Enable

MISCELLANEOUS	4	
/WB /OE	BT4	Write Buffer Output Enable
/WB /CE	BT4	Write Buffer Clock Enable
/S4C /OE	TLCHTNU	S4-Cache Chip Output Enable
/S4C /TEST	IBUFNU	S4-Cache Chip Test mode

INPUT/OUTPUT BUFFER DEFINITIONS:

DRVC8	Input Clock Driver
IBUFNU	Input Buffer, CMOS level, inverting, internal pullup.
TLCHT	Input buffer, TTL level, non-inverting
TLCHTU	Input buffer, TTL level, non-inverting, internal pullup.
TLCHTNU	Input buffer, TTL level, inverting
BD#TU	Bidirectional buffer, TTL input levels, internal pullup, # indicates output drive.
BD#TRU	Bidirectional buffer, TTL input levels, internal pullup, slew-rate controlled output, # indicates output drive.
BT#	Tri-statable output buffer, CMOS, # indicates drive current .

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	Typ	Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 15 pF)

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
t1	From CLK High to CLK High	50		ns
t2	From CLK to /IU AL		34.4	ns
t3	From CLK to /IU /AOE		17.8	ns
t4	From CLK to /IU /MDS		17.1	ns
t5	From CLK to /IU /MEXC		17.4	ns
t6	From CLK to /IU /MHOLD		16.3	ns
t7	From CLK to /SB A (untransl.)		41.8	ns
t8	From CLK to /SB A (seg. map)		30.3	ns
t9	From CLK to /SB /ACK32		24.4	ns
t10	From CLK to /SB /ACK8		24.4	ns
t11	From CLK to /SB /AS		27.7	ns
t12	From CLK to /SB /BG		19.8	ns
t13	From CLK to /SB /ERR		26.1	ns
t14	From CLK to /SB /MERR		24.7	ns
t15	From CLK to /SB RD		29.1	ns
t16	From CLK to /SB /RESET		23.0	ns
t17	From CLK to /SB SIZ		36.9	ns
t18	From CLK to /CAR /EN		16.7	ns
t19	From CLK to /CD /OE		22.0	ns
t20	From /SB /RD to /CD /OE		16.1	ns
t21	From /IU /RD to /CD /OE		11.8	ns
t22	From CLK to /CDWE /EN		15.6	ns
t23	From CLK to /CT A		33.7	ns
t24	From CLK to /CT C		24.9	ns
t25	From From CLK to /CT S		24.7	ns

8. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CIN	Input Capacitance			pF
COUT	Output Capacitance			pF

9. TRUTH TABLE: N/A

10. FUNCTIONAL TABLE: N/A

11. OUTLINE DRAWING: N/A

12. WAVEFORMS: N/A

DATE: 1/10/89

AUTHOR: PL

t26	From CLK to /CT V		24.8	ns
t27	From /SB RD to /CT V		12.5	ns
t28	From CLK to /CT WA		24.7	ns
t29	From CLK to CTL		16.0	ns
t30	From CLK to /CTWE /EN		18.2	ns
t31	From CLK to /DEVSPC		16.8	ns
t32	From CLK to /IO D		74.5	ns
t33	From /SB A to /IO D		41.0	ns
t34	From CLK to /USER		17.1	ns
t35	From CLK to /WB /CE		16.0	ns
t36	From CLK to /WB /OE		16.4	ns

7. AC OPERATING REQUIREMENTS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 15 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t37	From /IU AH to CLK	-1.1		ns
t38	From /IU AL to CLK	-1.2		ns
t39	From /IU ASI to CLK	10.0		ns
t40	From /IU /ERROR to CLK	5.3		ns
t41	From /IU /HAL to CLK	5.3		ns
t42	From /IU LDST to CLK	9.5		ns
t43	From /IU NULCYC to CLK	0.6		ns
t44	From /IU RD to CLK	2.5		ns
t45	From /IU SIZ to CLK	10.0		ns
t46	From /IU WE to CLK	11.2		ns
t47	From /SB A to CLK	10.5		ns
t48	From /SB /ACK to CLK	7.3		ns
t49	From /SB /ERR to CLK	10.8		ns
t50	From /SB /MERR to CLK	4.5		ns
t51	From /SB /RD to CLK	18.4		ns
t52	From /CT A to CLK	9.7		ns
t53	From /CT C to CLK	9.7		ns
t54	From /CT S to CLK	3.3		ns
t55	From /CT V to CLK	2.6		ns
t56	From /CT WA to CLK	2.9		ns
t57	From /IO D to CLK	10.6		ns
t58	From /MMU (s,typ,v,w,x) to CLK	5.5		ns

Sun Microsystems S4 Chip Set

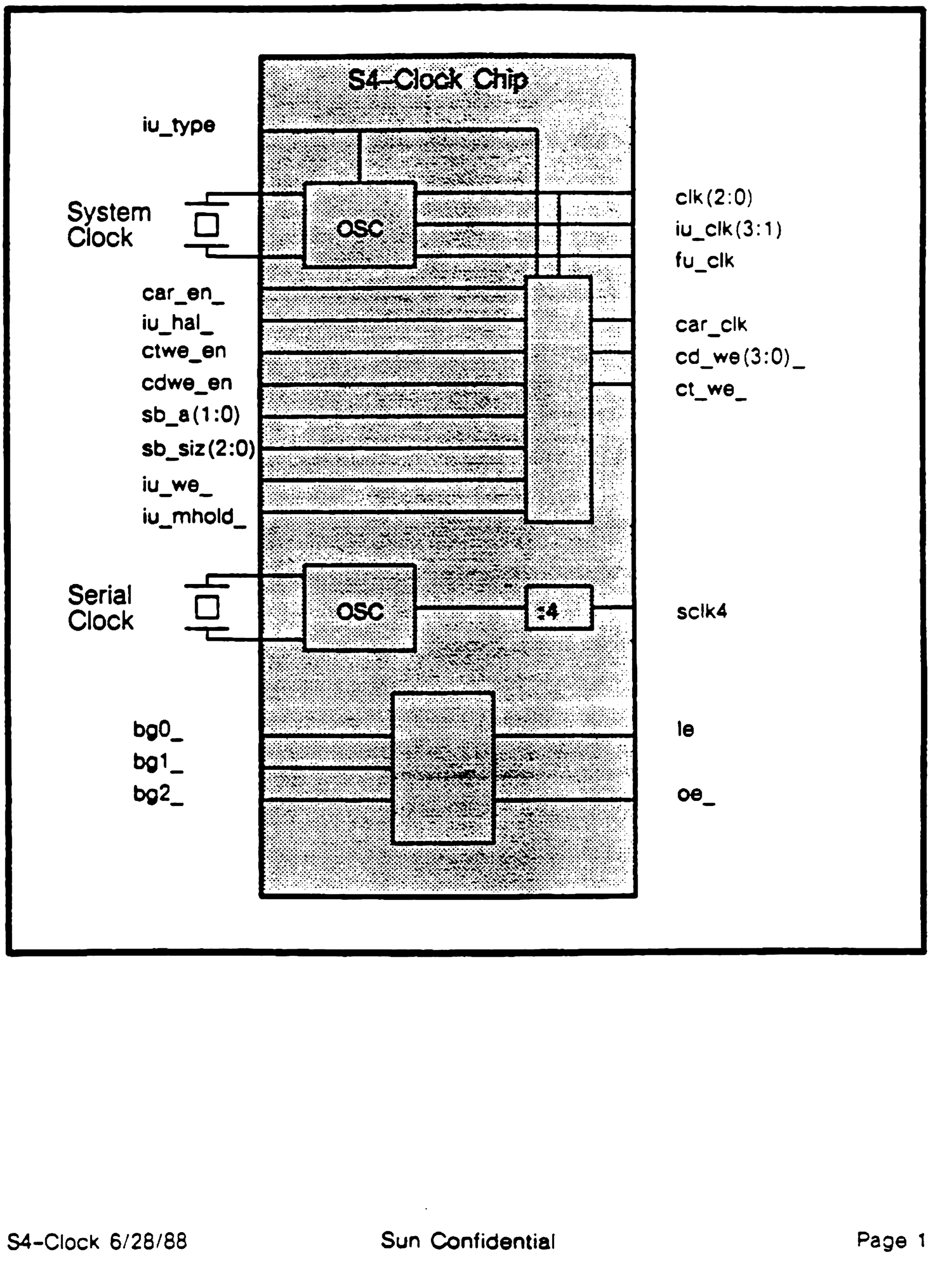
Sun SPARC(TM) Support System

S4-Clock

Preliminary

Features

- Generates all system clocks
- Generates critical write strobes
- Generates serial controller clock
- Generates refresh clock and periodic interrupt clock



Pin Description

Symbol	Type	Description
System Clocks	**25**	
xi,xo	OSCIM	Main crystal oscillator
clk(2:0)	BT8	System clock (3 outputs)
iu_clk1	BT8	Integer Unit Clock1 for LSI and Fujitsu
iu_clk2	BT8	Integer Unit Clock2 for Fujitsu only
fu_clk	BT8	Floating Point Unit Clock, positive
car_en_	TLCHT	Enable Car Clk
sb_a(1:0)	TLCHTU	S Bus Address Lines
sb_siz(1:0)	TLCHTU	S Bus Size Bits
ctwe_en	TLCHT	Cache Tag Write Enable Enable
ct_we_	BT8	Cache Tag Write Enable
cdwe_en	TLCHT	Cache Data Write Enable Enable
cd_we(3:0)_	BT4	Cache Data Write Enable
iu_hal_	TLCHN	Integer Unit Hold Address Low
— iu_we_	TLCHN	Integer Unit Write Enable
— iu_mhold_	TLCHT	Integer Unit MHOLDA
car_clk	BT8	Cache Address Register Clock
iu_type	IBUF	Selects LSI (low) or Fujitsu (high) timing
Serial Clock	**3**	
xso,xsi	OSCIM	Crystal Serial Clock Oscillator (nominal 19.6608 MHz)
sclk4	BT2	Serial Contoller Clock Out (to SCC chips)
DMA Control Logic	**5**	
bg(2:0)_	TLCHN	Bus grant signals from cache
le	BT8	Latch enable
oe_	BT4	Output Enable
Test Pins	**2**	
por_	IBUFNU	Low for chip test mode
para_	BD1CNU	Parametric Test Output/Output Disable

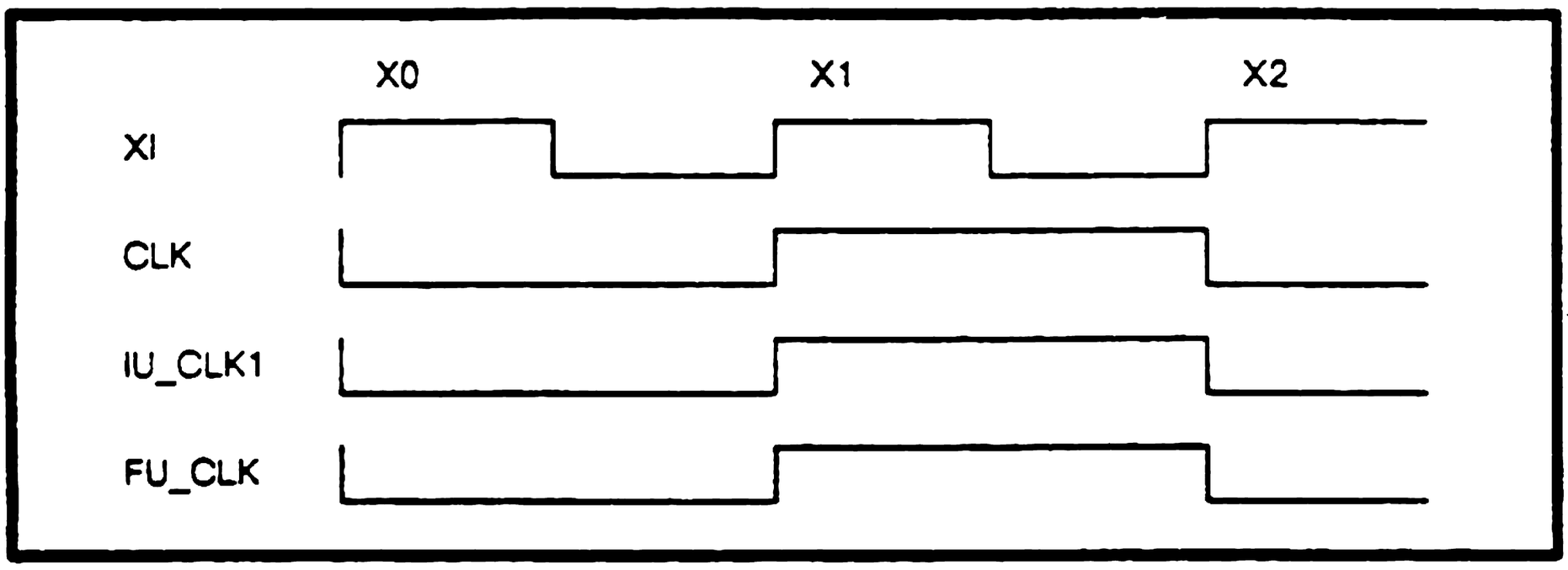
Signals	35	Package	PLCC44
VDD	4	Chip	LMA9020
GND	5		
Total Pins	44		

Functional Description

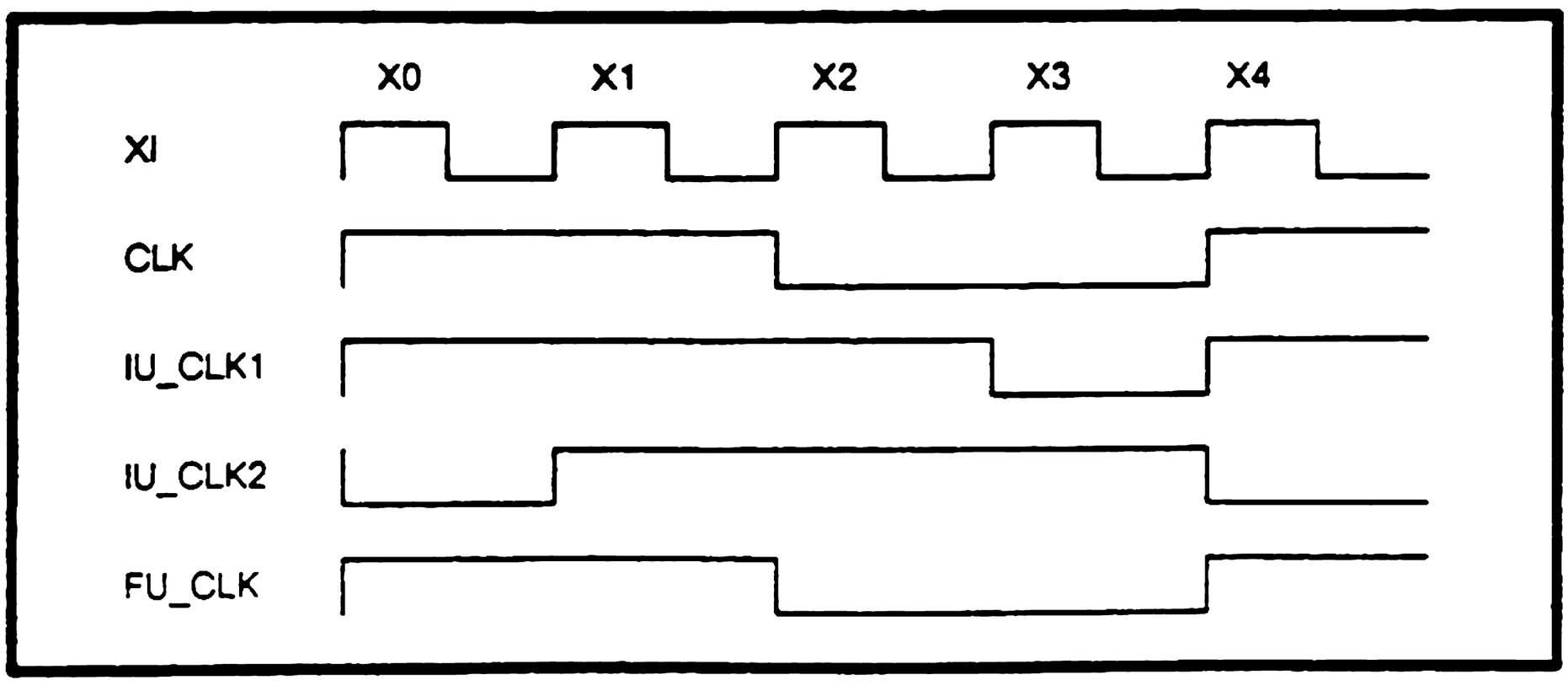
System Clocks

The S4-Clock chip generates all system clocks required by either an LSI Logic or a Fujitsu implementation of the SUNRISE varient of the SUN SPARC IU. The LSI Logic version of the IU requires a single phase 50% duty cycle clock. The Fujitsu version requires 2 phase 72-25% clocks. The clock chip requires a 2X cyrstal frequency for LSI applications and a 4X crystal for Fujitsu applications.

IU SYSTEM CLOCKS (LSI Logic Version)



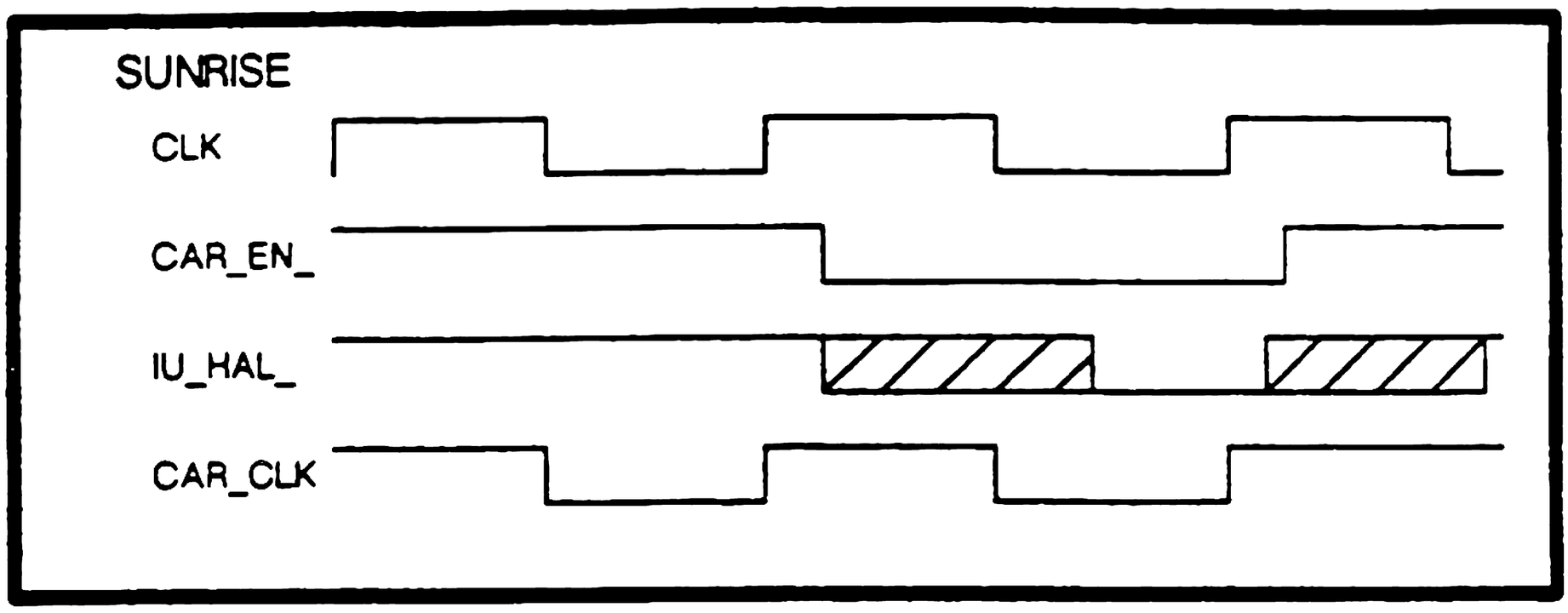
IU SYSTEM CLOCKS (Fujitsu Version)



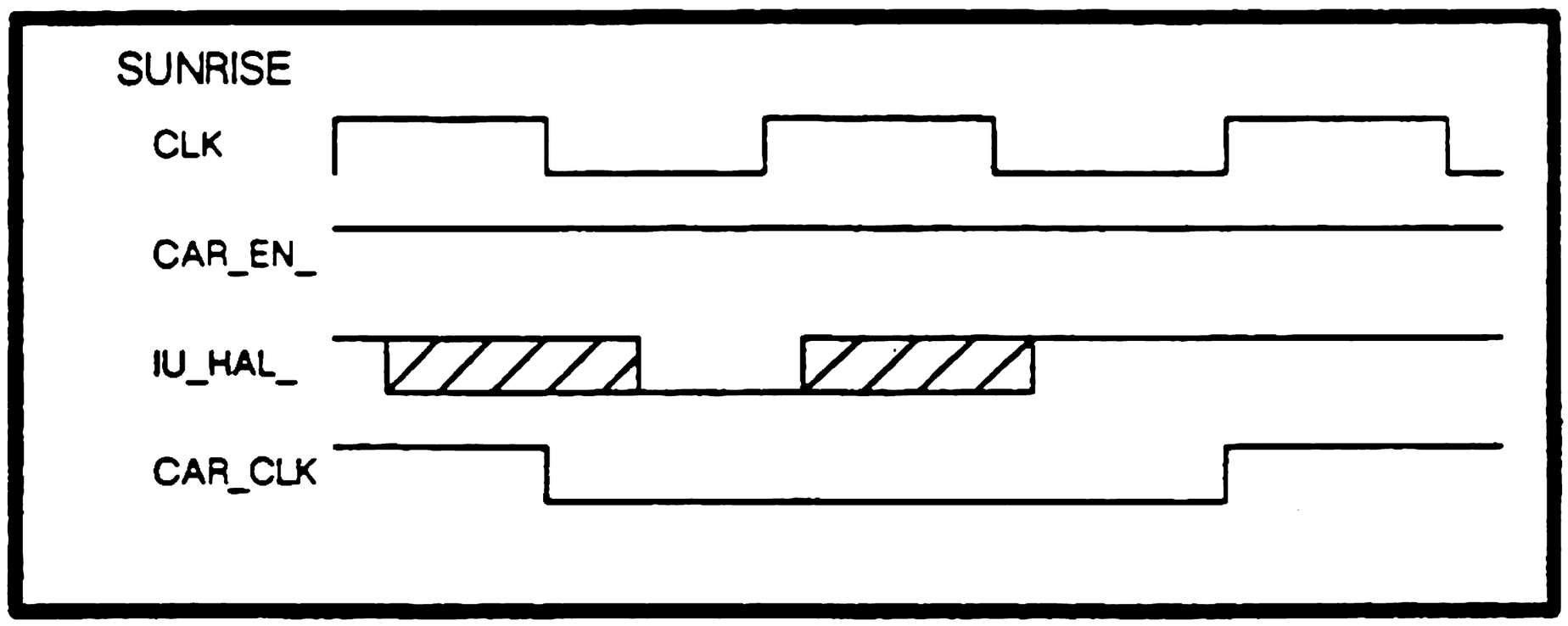
CAR_CLK

The signal CAR_CLK is used to clock the Cash Address Register. The rising edge of CAR_CLK must be inhibited if IU_HAL becomes active. However if IU_HAL becomes active and CAR_EN is also active, then the CAR_CLK is not inhibited.

CAR_CLK (CAR_EN_ Asserted)



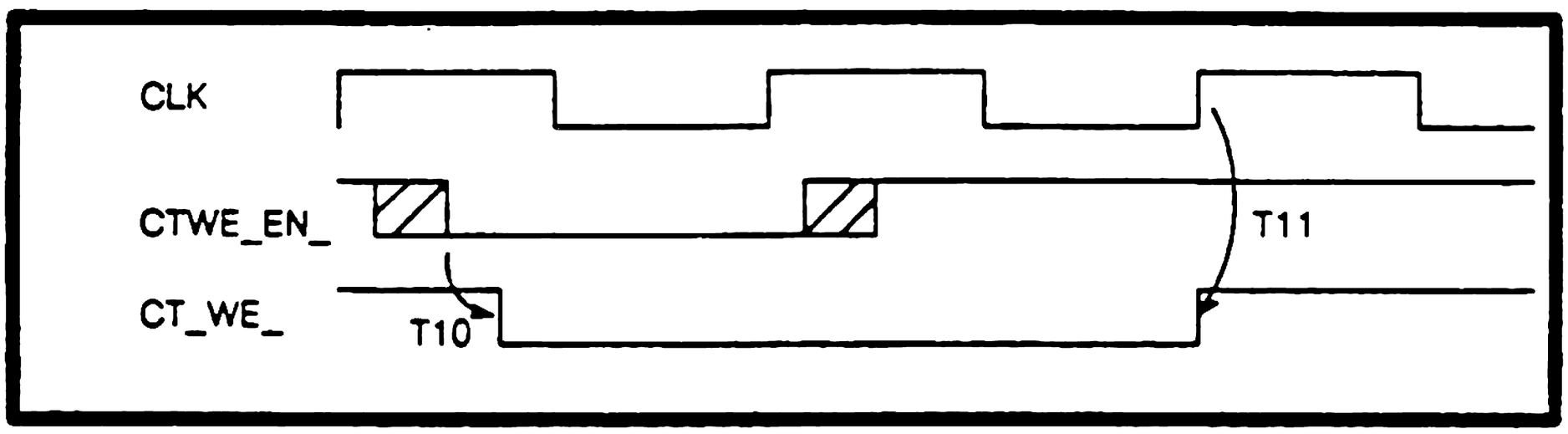
CAR_CLK (CAR_EN_ De-asserted)



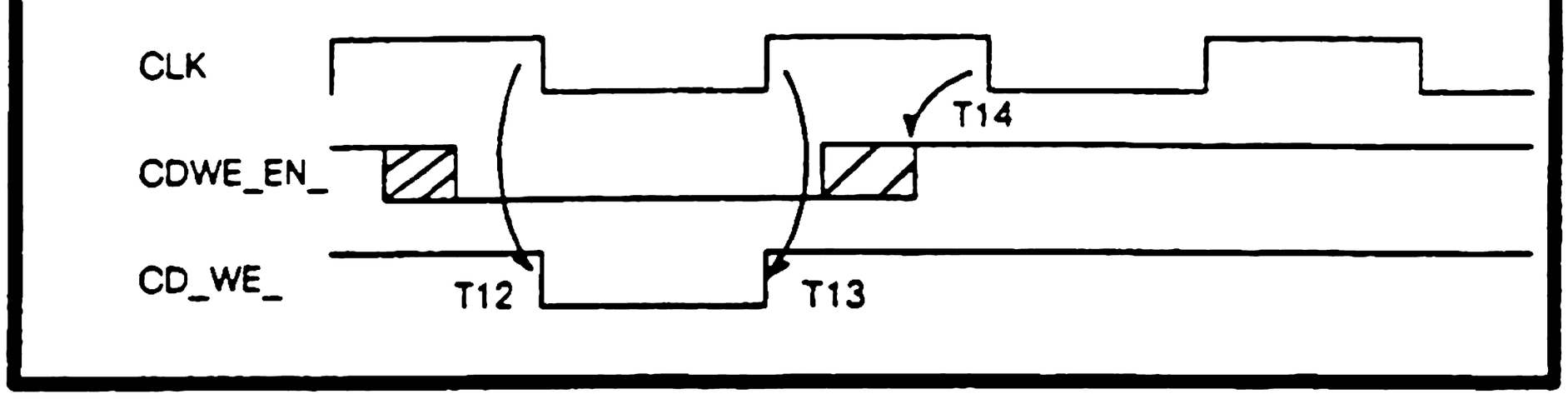
Cache Data and Cache Tag Write enables

The write enables for the Cache Data and Cache Tag rams are generated by the S4_Clock chip. The wave forms are dependant on the final CLK wavefrom and are thus independent of the type of IU chip, Sunray or Sunrise. There should be no more than 3ns skew between the rising edge of CLK and the rising edge of CT_WE_. In the case of CD_WE_, there should be no more than 3ns of skew between the CLK and CD_WE_ on either edge. CT_WE_ goes active immediately following CTWE_EN_ going active, and remains active until the second rising clock (CLK) edge following. CD_WE_ is controlled by CDWE_EN_. During single cycle writes, CD_WE_ goes active synchronus with the CLK signal going low, and returns inactive with the next positive transition on CLK. Multiple cycle writes are achieved by keeping the CDWE_EN_ signal asserted through the end of a clock cycle. In this case the CD_WE_ signal will go inactive for one half of the clock period and return active at the beginning of the next cycle. This pattern repeats until the CDEN_EN_ signal is de-asserted.

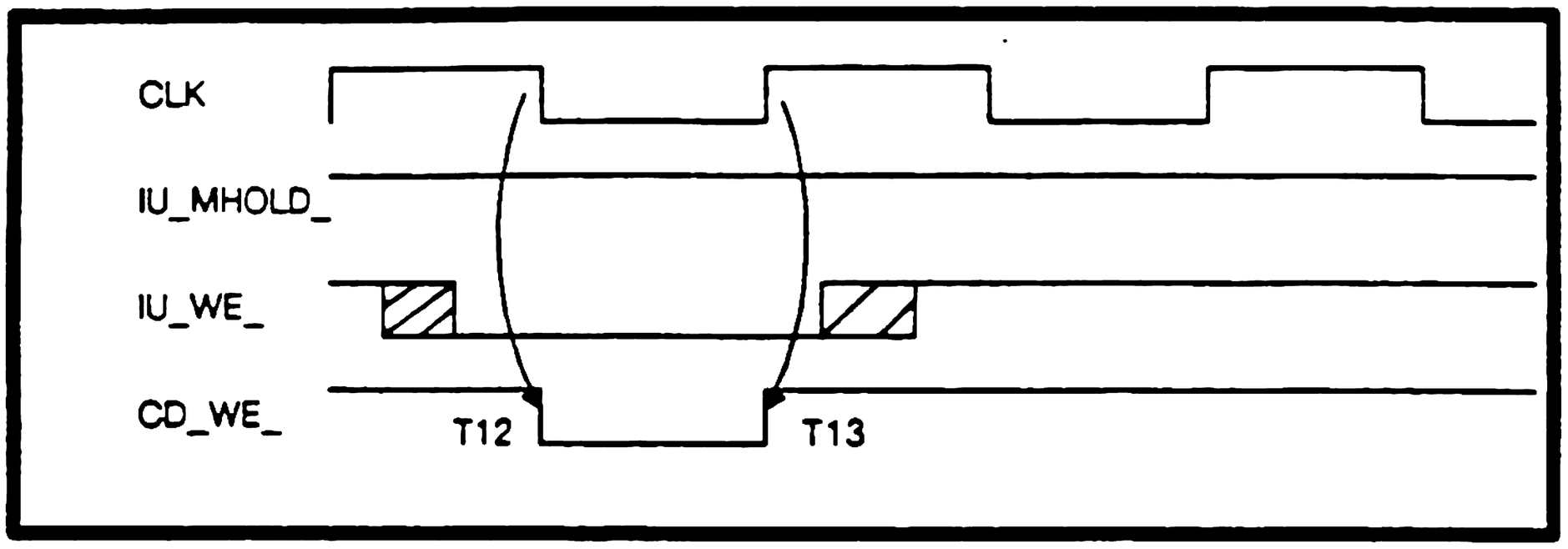
CACHE TAG WRITE ENABLE



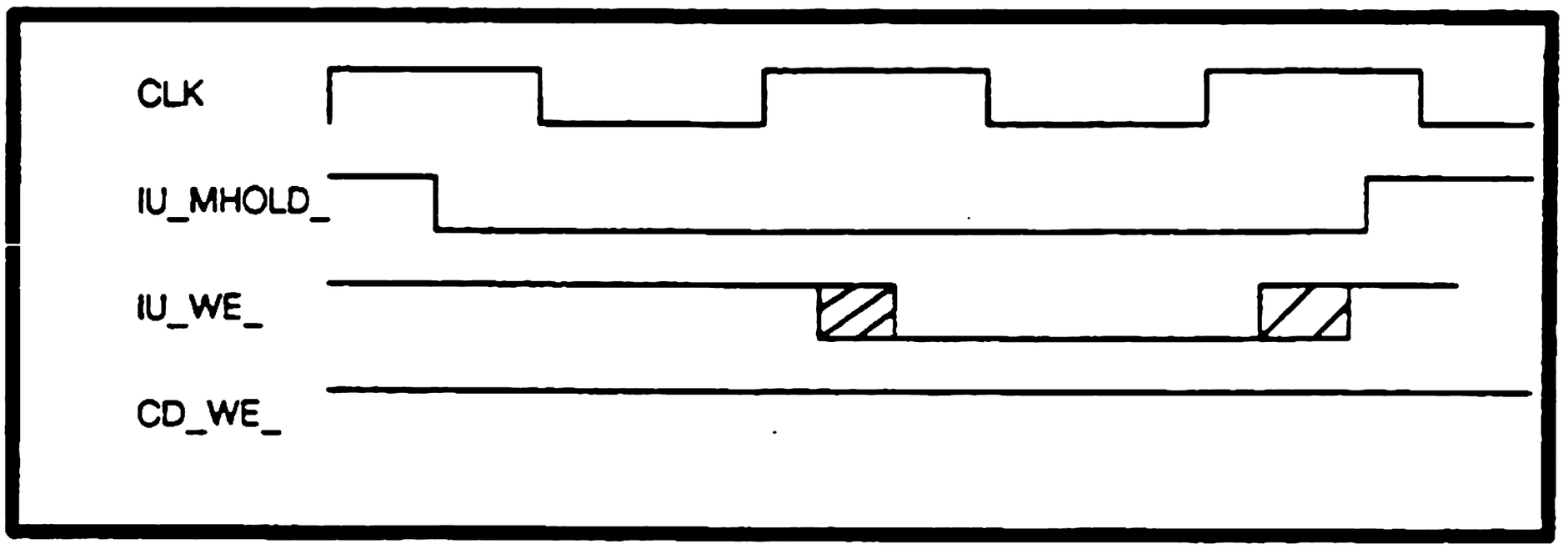
CACHE DATA WRITE ENABLE (Single Cycle, cd_we controlled)



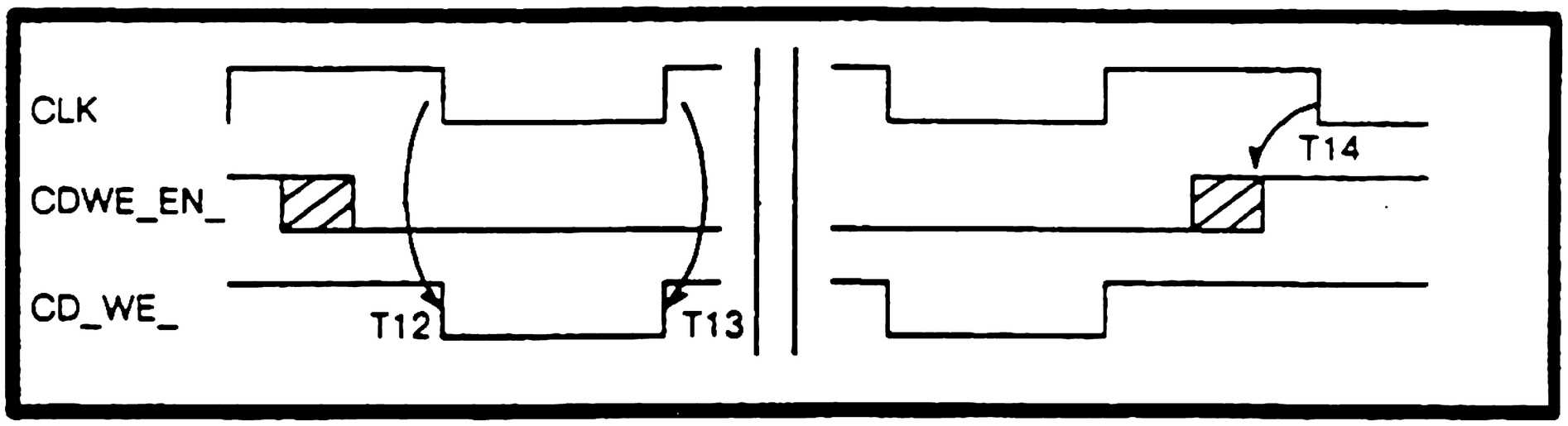
CACHE DATA WRITE ENABLE (Single Cycle, iu_we_ controlled)



CACHE DATA WRITE ENABLE (Single Cycle, iu_mhold_ cancel)

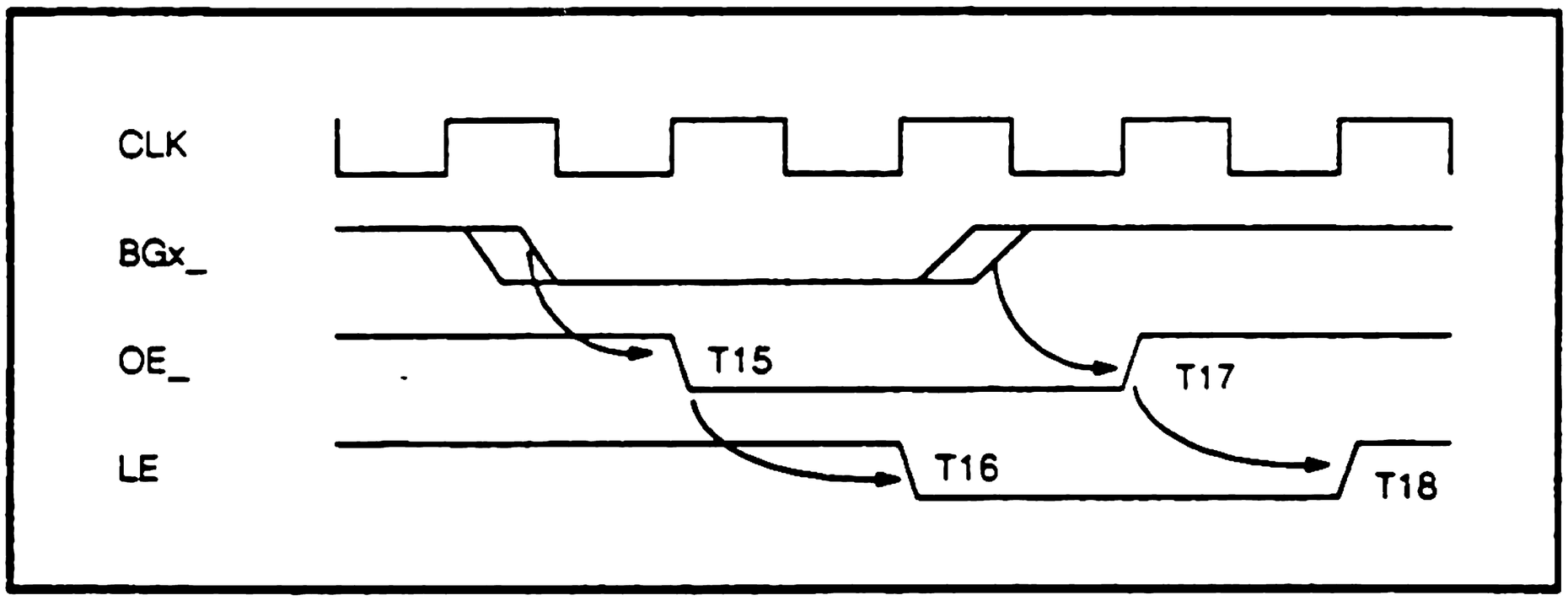


CACHE DATA WRITE ENABLE (Multiple Cycle, cd_we_ controlled)



DMA Control Logic

The S4-Clock chip generates the appropriate control signals to control a data latch for DMA devices. Data bus signals from DMA devices are fed into a 32 bit wide latch. The clock chip monitors the bus grant signals (bg(2:0)_) and generates a latch enable (le) and output enable (oe_) signal accordingly.



Byte Writes to Cache Data

There are actually 4 CD_WE_ output pins. This is done to allow writing individual bytes to the cache. Which of the four CD_WE_ signals go active on a particular write cycle is determined by the SB_SIZ(2:0) and SB_A(1:0) pins. The decoding of the signals is as follows:

CACHE BYTE WRITE DECODE

SB_SIZ 1	SB_SIZ 0	SB_A 1	SB_A 0	CD_WE_ 0	CD_WE_ 1	CD_WE_ 2	CD_WE_ 3
0	0	0	0	X	X	X	X
0	0	0	1		X	X	X
0	0	1	0			X	X
0	0	1	1				X
0	1	0	0	X			
0	1	0	1		X		
0	1	1	0			X	
0	1	1	1				X
1	0	0	0	X	X		
1	0	0	1		X	X	
1	0	1	0			X	X
1	0	1	1				X
1	1	0	0	X	X	X	
1	1	0	1		X	X	X
1	1	1	0			X	X
1	1	1	1				X

An X indicates that the signal is active during a write cycle.

Serial Clock

The S4_Clock chip also generates the serial controller clock. The nominal serial clock crystal is 19.6608 MHz and the SCLK4 output drives the SCC at 4.9152 MHz.

Test Pins

There are two dedicated test pins which are used in combination with other pins to enhance testing of the design. These two pins are TEST_ and PARA_. When the TEST_

pin is inactive, the pin PARA_ acts as an input to force all output pins (except the crystal pins) into a tri-state mode.

When the TEST_ pin is active, several other pins are used in conjunction with TEST_ pin to aid in the testing of the part. Bringing the SUNRAY pin active while the TEST_ pin is active will cause all of the oscillator logic associated with the system clocks, tclks and sclks to reset. The SB_SIZ(1:0) pins are used in conjunction with the TEST_ pin to force the divide by 100K divider into special count modes which allow the counter to be tested more easily.

Additionally when the TEST_ pin is active, the pin PARA_ acts as an output which reflects the state of the parametric nand tree imbeded into all inputs (with the exception of the crystal input pins) of the chip.

Timing Specifications

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C, Output Load=15 pF except:

cd_we(3:0)	20pf
ct_we	40pf
le	50pf
oe_	50pf

Symbol	From	To	min	max	unit
t1	xclk high	xclk high+1	10	---	ns
t2	xclk high	xclk low	3	---	ns
t3	xclk low	xclk high	3	---	ns
t4	sclk high	sclk high+1	50	---	ns
t5	sclk high	sclk low	20	---	ns
t6	sclk low	sclk high	20	---	ns
t7	tclk high	tclk high+1	50	---	ns
t8	tclk high	tclk low	20	---	ns
t9	tclk low	tclk high	20	---	ns
t10	ctwe_en_ low	ct_we_ low	---	10	ns
t11	clk high	ct_we_ high	-3	3	ns
t12	clk low	cd_we_ low	-3	3	ns
t13	clk high	cd_we_ high	-3	3	ns
t14	ctwe_en_ valid	clk high	tbd	tbd	ns
t15	clk high	oe_ low	-3	3	ns
t16	clk high	le low	-3	3	ns
t17	clk high	oe_ high	-3	3	ns
t18	clk high	le high	-3	3	ns

Change History

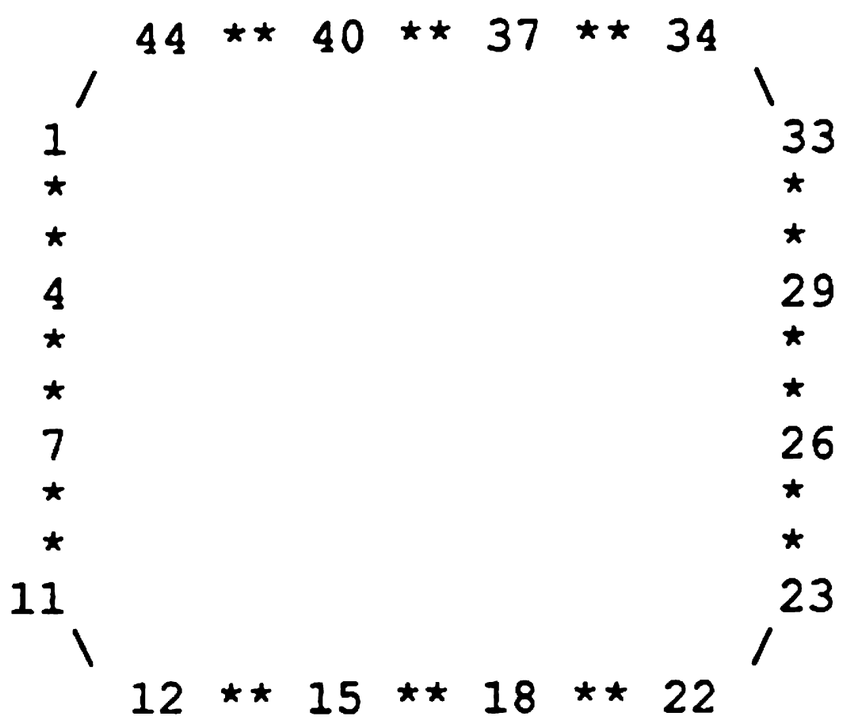
Date	Change	By
12/22/87	First release	AVB
2/8/88	Corrected pin descriptions	MWI
2/29/88	Removed timer clock, added DMA control logic, went to 44 pins	MWI
3/7/88	Corrected Fujitsu clock timing per Fujitsu March 88 spec.	MWI
5/13/88	Added sb_size2, changed le_ out from BT4 to BT8, rmv'd one GND	MWI
6/28/88	Added iu_we_ and iu_mhold_	MWI

1. P/N: 100-1809-04 2. REV: 01

3. DESCRIPTION: IC,GA,S4-CLOCK

4. PHYSICAL MODEL: (44 Pin PLCC)

A. Pinout: *Top View



1 - VSS	12 - VSS	23 - /CT /WE	34 - VSS
2 -	13 - CLK.0	24 - /CTWE /EN	35 - /CAR /EN
3 - XSI	14 - /POR	25 - /BG.0	36 - /CAR /CLI
4 - SCLK4	15 - /PARA	26 - /BG.1	37 - /IU /HAL
5 - /IU /WE	16 - /CD /WE.0	27 - /BG.2	38 - LE
6 - /SB SIZ.1	17 - /CD /WE.1	28 - CLK.2	39 - /OE
7 - /SB SIZ.0	18 - /CD /WE.2	29 - /IU CLK.2	40 - /SB A.1
8 - /IU TYPE	19 - /CD /WE.3	30 - /IU CLK.1	41 - /SB A.0
9 - CLK.1	20 - /CDWE /EN	31 - /FU CLK	42 -
10 - /IU /MHOLD	21 - VSS	32 - VSS	43 - XI
11 - VDD	22 - VDD	33 - VDD	44 - VDD

B. PIN DESCRIPTION:
==========

SYMBOL	TYPE	DESCRIPTION
SYSTEM CLOCKS	25	
XI,X0	OSCIM	Main crystal osillator
CLK(2:0)	BT8	System Clock (3:0 outputs)
/IU CLK1	BT8	Integer Unit Clock 1 for LSI and Fujitsu.
/IU CLK2	BT8	Integer Unit Clock 2 for Fujitsu only
/FU CLK	BT8	Floating Point Unit Clock, positive
/CAR /EN	TLCHT	Enable CAR CLK
/SB A(1:0)	TLCHTU	S Bus Address Lines
/SB SIZ(1:0)	TLCHTU	S Bus Size Bits
/CTWE EN	TLCHT	Catche Tag Write Enable Enable
/CT /WE	BT8	Catche Tag Write Enable
/CDWE EN	TLCHT	Catche Data Write Enable Enable
/CD /WE(3:0)	BT4	Catche Data Write Enable
/IU /HAL	TLCHN	Integer Unit Hold Address Low
/IU /WE	TLCHN	Integer Unit Write Enable
/IU /MHOLD	TLCHT	Integer Unit MHOLDA
/CAR /CLK	BT8	Catche Address Register Clock
/IU TYPE	IBUF	Selects LSI (Low) or Fujitsu (High) timing.
SERIAL CLOCK	3	
XSO,XSI	OSCIM	Crystal Serial Clock Oscillator (nominal 19.6608 MHz)
SCLK4	BT2	Serial Controller Clock Out (to SCC chips)
DMA CONTROL LOGIC	5	
/BG(2:0)	TLCHN	Bus Grant signals from catche
LE	BT8	Latch Enable
/OE	BT4	Output Enable
TEST PINS	2	
/POR	IBUFNU	Low for chip test mode
/PARA	BD1CNU	Parametric Test Output/Output Disabl

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
tTCT	TCLK Period	40		ns
tTCH	TCLK High Time	17		ns
tTCL	TCLK Low Time	17		ns
	TCLK to Output Valid		20	ns
	TCLK to Output Invalid	3		ns
	Input Setup to Clock	15		ns
	Input Hold Time to Clock	0		ns

7. AC OPERATING REQUIREMENTS*
(VCC = 4.75 to 5.25V, TA = 0 to +70oC)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t1	From XCLK High to XCLK High+1	10		ns
t2	From XCLK High to XCLK Low	3.0		ns
t3	From XCLK Low to XCLK High	3.0		ns
t4	From SCLK High to SCLK High+1	50		ns
t5	From SCLK High to SCLK High	20		ns
t6	From SCLK Low to SCLK High	20		ns
t7	From TCLK High to TCLK High+1	50		ns
t8	From TCLK High to TCLK Low	20		ns
t9	From TCLK Low to TCLK High	20		ns
t10	From /CTWE /EN Low to /CT /WE Low		10	ns ns
t11	From CLK High to /CT /WE High	-3.0	3.0	ns ns
t12	From CLK Low to /CD /WE Low	-3.0	3.0	ns
t13	From CLK High to /CD /WE High	-3.0	3.0	ns
t14	From /CTWE /EN Valid to CLK High	TBD	TBD	ns
t15	From CLK High to /OE Low	-3.0	3.0	ns
t16	From CLK High to LE Low	-3.0	3.0	ns
t17	From CLK High to /OE High	-3.0	3.0	ns
t18	From CLK High to LE High	-3.0	3.0	ns

* Output Load = 15 pF except:

/CD WE(3:0)	20 pF
/CT WE	40 pF
LE	50 pF
/OE	50 pF

8. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CI/O	Input/Output Capacitance per Slot		20	pF
CLS	Maximum Capacitance Load per System		100	pF

9. TRUTH TABLE: N/A

10. FUNCTIONAL TABLE: N/A

11. OUTLINE DRAWING: N/A

12. WAVEFORMS: N/A

DATE: 1/05/89

AUTHOR: PL

~~~~~~
~~~~~~

Sun Microsystems S4 Chip Set

Sun SPARC(TM) Support System

S4-DMA

Features

* Single chip interface between Ethernet (LANCE), SCSI (ESP) and Sbus
* Handles 32 bit packing and unpacking
* Generic support for 8 bit peripherals
* Supports externally programmable Sbus ID
* Low cost 120PFP package

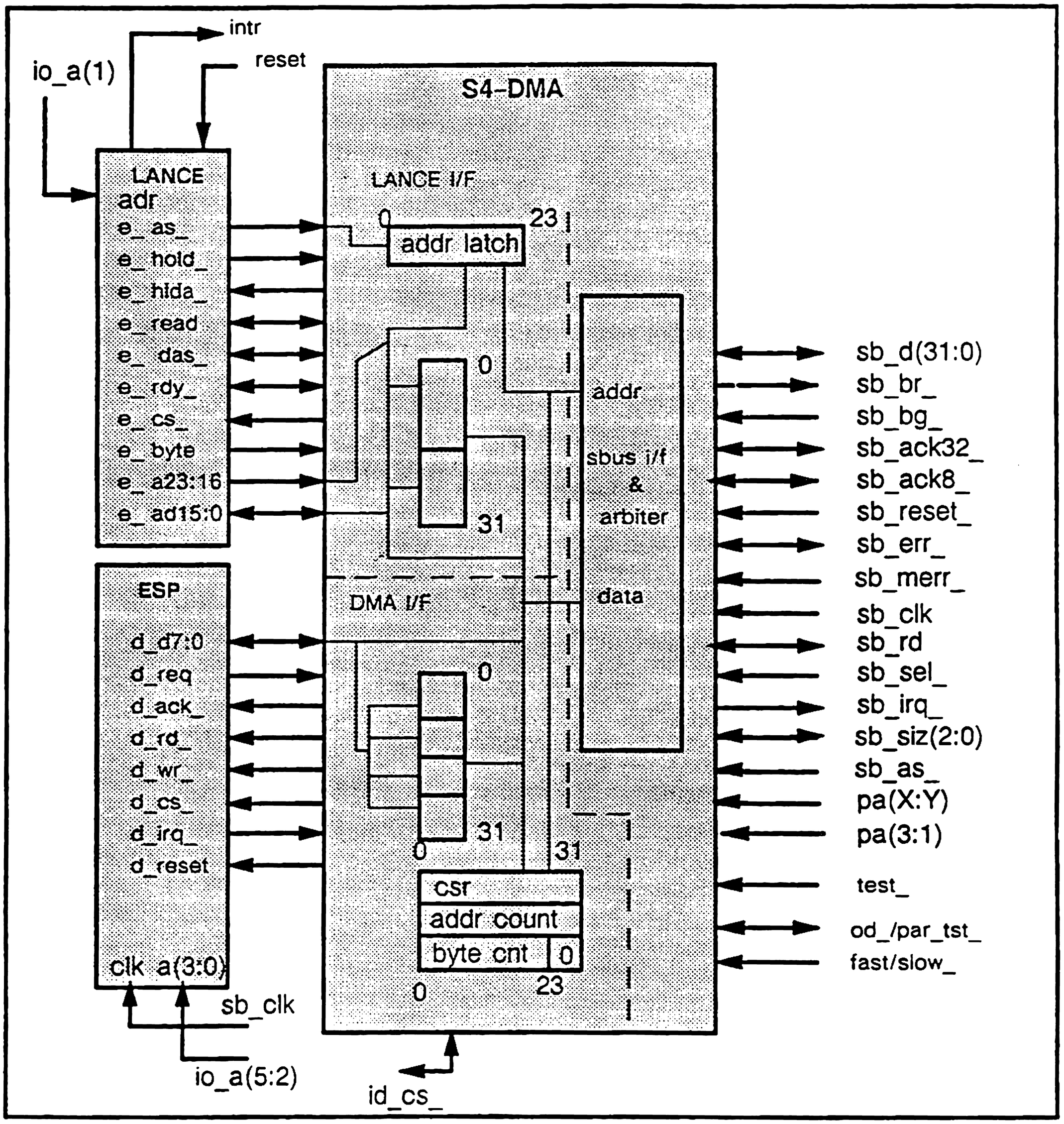


Table of Contents

1.0 Pin Description

Name	Type	Description
Bus Interface	**51**	
sb_d(31:0)	BD4TU	Sbus Data Bus
sb_br_	BT4	Sbus Bus Request
sb_bg_	TLCHTU	Sbus Bus Grant
sb_ack32_	BD4TNU	Sbus 32bit Acknowledge
sb_ack8_	BD4TNU	Sbus 8bit Acknowledge
sb_reset_	TLCHTU	Sbus Reset
sb_err_	BD4TNU	Sbus Error
sb_merr_	TLCHTU	Sbus Memory Error (INT15)
sb_clk	DRVC16	Sbus Clock input
sb_rd	BD4TU	Sbus Read/Write_
sb_sel_	TLCHTU	Sbus Select
sb_irq_	BD4TOD	Interrupt Request (open-drain)
sb_siz(2:0)	BD4TU	Sbus transfer Size
sb_as_	TLCHTU	Address strobe (addr is valid)
pa(X:Y)	TLCHTU	Physical Address lines (for slave decodes)
pa(3:1)	TLCHTU	Physical Address bits
Ethernet Interface	**32**	
e_as_	TLCHTD	Ethernet Address Strobe
e_hold_	TLCHTU	Ethernet Hold
e_hlda_	BT4	Ethernet Hold Acknowledge
e_read	BD4TU	Ethernet Read
e_das_	BD4TU	Ethernet Data Strobe
e_rdy_	BD4TU	Ethernet Ready
e_cs_	BT4	Ethernet Chip Select
e_byte	TLCHTU	Ethernet Byte marker
e_a23:16	TLCHTD	Ethernet High Order Address
e_ad15:0	BD4TU	Ethernet Address / Data Bus
DMA Interface	**16**	
d_d7:0	BD4TD	DMA Data Bus
d_req	TLCHT	DMA Request
d_ack_	BT4	DMA Acknowledge
d_rd_	BT4	DMA Read Strobe. (reg read or dma to memory).
d_wr_	BT4	DMA Write Strobe. (reg write or dma from memory).
d_cs_	BT4	DMA Chip Select for slave register access.
d_irq_	TLCHTU	DMA Interrupt Request
d_reset	BT4	DMA Reset

id_cs_	BD4T	Secondary Device Select (boot prom) output Strap high to specify existence of external prom

Miscellaneous **4**

Name	**Type**	**Description**
fast/slow_	TLCHTU	fast or slow DMA acknowledge cycles (For use with ESP SCSI controller fast mode should be used – internal pullup means this requires no external connection for fast mode).
test_	IBUFU	Test control. When low enables parametric test output and disables all tri-state drivers. When high disables parametric test output.
od_tst	BT4	parametric test output

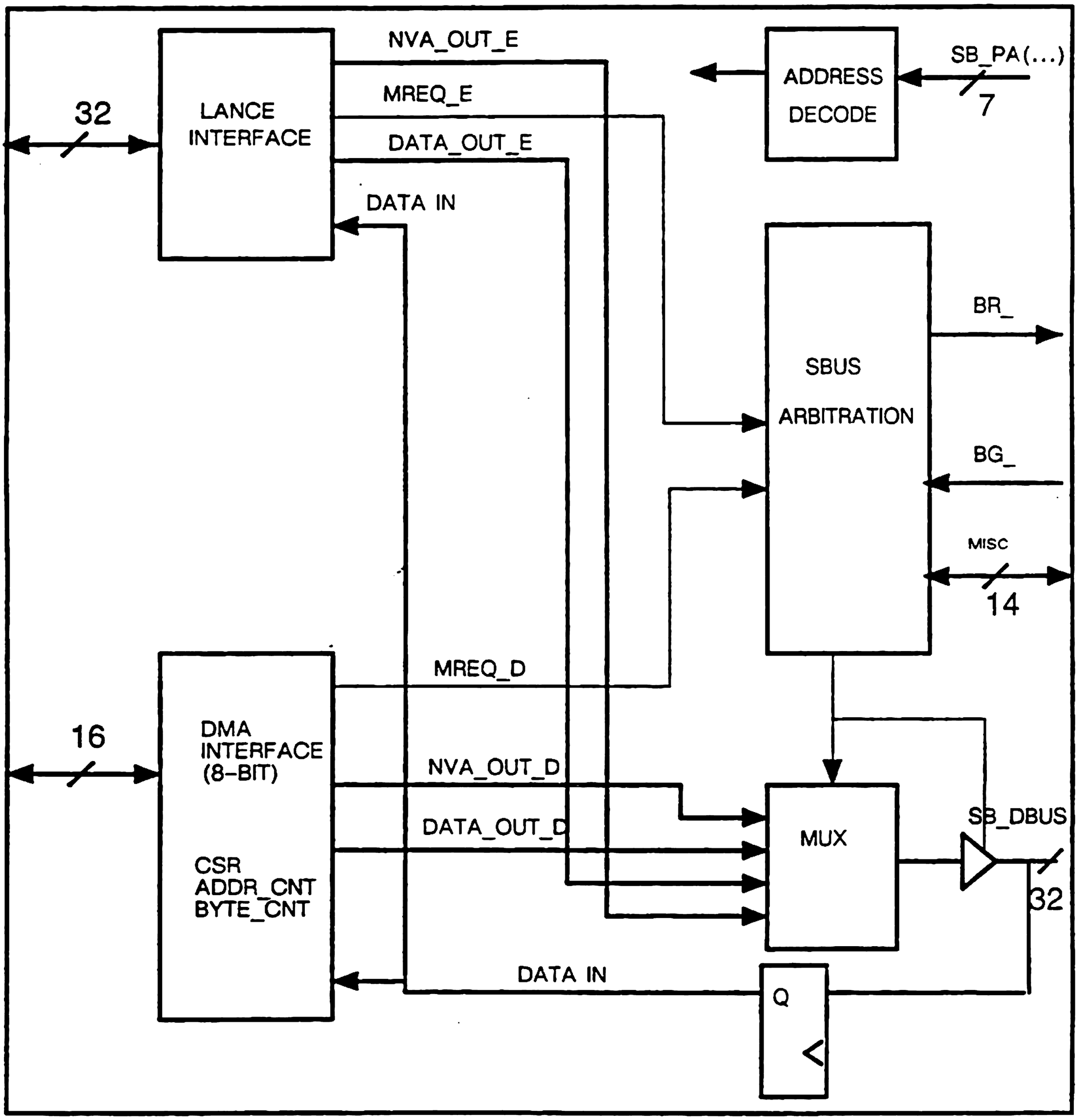
Signals: 103
VDD: 6
VSS: 11
TOTAL: 120

Part: LMA9191
Package: PFP120

1.1 BLOCK DIAGRAM

The S4-DMA gatearray provides three independent functions;

1. Sbus Identification
2. Ethernet Interface to the Sbus
3. Sbus DMA Channel



2.0 Sbus Interface

The S4-DMA gatearray provides both Slave cycle and Master cycle accesses over the Sbus. Devices which can be Masters on the Sbus include the Ethernet Controller and a device such as the ESP SCSI interface Controller. Master cycles can only access data in Type 0 Device Space. Address and Control registers in the S4-DMA gatearray, along with similar registers in both the Ethernet Controller and the attached DMA device, can be accessed by the CPU via Slave cycles.

2.1 Master Cycles

To perform a Master Cycle the S4-DMA requests the use of the Sbus by asserting sb_br_, the bus request signal. On reception of sb_bg_, bus grant, the S4-DMA takes control of the Sbus to transfer data to or from memory, using the signals sb_d(31:0), sb_read, sb_siz(2:0), sb_err, sb_ack8_ and sb_ack32_ . The address is multiplexed onto the sb_d bus for all accesses on the Sbus.

All DMA reads from memory will be full word reads. DMA writes to memory can be either byte, half-word, or word sized.

The SBus interface is capable of supporting reruns on DMA cycles. The requested DMA transfer will repeat until either it completes or an error indicator (sb_err_) is received, whereupon it will be aborted.

2.2 Slave Cycles

The CPU accesses registers in the S4-DMA chip and attached devices by the use of a geographical select signal on the Sbus, .. sb_sel_ , in conjunction with sb_as_. Within the geographical select the S4-DMA will decode the Physical address, (pa(X:Y)) in the following way;

pa(X:Y)	Addressed device	transfer size
00	Internal ID field (or external if id_cs_ is pulled high)	word
01	On chip DMA registers	word
10	DMA device registers (off-chip)	byte
11	Ethernet registers (off-chip)	2 byte

NOTE: Slave accesses to these registers must use the correct transfer size corresponding to the device being accessed. The S4-DMA gatearray will always respond with it's own port size i.e. with sb_ack32_, except when an offboard ID field is read whereupon the response will be via sb_ack8_.

During Slave Cycles the S4-DMA takes control of the sb_err, sb_ack8_ and sb_ack32_, signals. The combination of responses are as follows;

sb_ack8_	sb_ack32_	sb_err_	Definition	
1	1	1	insert wait states	**
1	1	0	Error	
1	0	1	32-bit port ack	**
1	0	0	Error	
0	1	0	Rerun	**
0	0	1	16-bit port ack	
0	1	1	8-bit port ack	**
0	0	0	Reserved	

This table represents all possible SBus responses. The S4-DMA gate-array can, however, only generate those responses marked with a **.

3.0 Sbus Identification

This is a mechanism which allows software to uniquely identify each Sbus device, since each device can have a unique ID.

Unique ID's will be provided by Sun. The onboard id is hardwired to the 32-bit value fe810101. This value will be returned when the ID field is accessed by the IU (and the -id_cs_ pin is tied low). If the id_cs_ pin is pulled high then access to the ID field will cause an external access using the id_cs_ pin as a external chip select. Refer to Campus-1 Programmers Model for further details.

4.0 Ethernet Interface

The Ethernet Controller is the AMD AM7990 LANCE chip and the AM7992 Serial Interface Adapter. The LANCE connects directly to the S4-DMA, over unique Ethernet interface signals. The S4-DMA will provide all necessary buffering and arbitration functions to allow the Ethernet chip to access main memory, over the Sbus, (master cycle), and the CPU to access the Ethernet chip, (slave cycle). The S4-DMA Ethernet interface contains a 1 word (32bit) pack/unpack register with consistency control logic. Consistency control ensures that all data written by the Ethernet chip gets to main memory in a deterministic manner.

The LANCE uses multiplexed address and data, so the S4-DMA demultiplexes them internally. The address supported by the LANCE is 24 bits; as per the SUN-4 Architectural Specification the upper 8 bits of the 32 bit virtual address are driven to 0xFF. The LANCE has a 16 bit data path which can accomodate 8 or 16 bit accesses, by the use of byte masking capabilities.

Data packing and unpacking is used to reduce the bus bandwidth impact of LANCE transfers on the Schoolbus. The LANCE performs 2 distinct types of memory access; data transfer from or to the network, and descriptor access for buffer management control. A write to a descriptor will always be a single half-word memory access, reads of descriptors will always be single word memory accesses. Network data will be transferred to/from memory as single words, wherever possible, there are a few exceptions to this case, (i.e. leading or trailing word fragments). Network data is packed or unpacked in the S4-DMA gatearray as a consequence of LANCE burst mode accesses. Burst mode is recognized when the LANCE keeps the e_hold_ signal asserted for more than 1 access cycle. It is guaranteed by AMD that burst mode accesses are always to contiguous memory locations. The LANCE will always do half-word data transfers except in the case of word fragments trailing a packet or by byte mis-aligned memory transfers.

When the LANCE is transferring data to memory it will be packed whenever possible, but in cases where the access is either not word aligned and there is no data buffered, or it is a single byte, then the S4-DMA will transfer the data to memory immediately. Data which is word aligned will be held pending the next sequential write, (assuming it is part of the same burst transfer) , whereupon the entire word will be transferred to memory. When a burst cycle ends any data remaining in the buffer will automatically be drained to memory. When there is data in the buffer, pending a sequential transfer, and that transfer turns out to be a byte write, the data in the buffer will be written to memory before the byte is written to the buffer and subsequently drained to memory. These rules ensure that no data is left in the pack buffer.

When the LANCE requests a transfer from memory the S4-DMA will read the word which contains the half-word requested, regardless of it's alignment. The S4-DMA gatearray will steer the appropriate half-word to the LANCE. Whenever a read burst ends the data, if any, left in the read buffer is discarded. Every time a word boundary is crossed a new word is read from memory.

Slave accesses to the LANCE registers will be completed with the sb_ack32_ signal, as defined in the Sbus specification. (see timing diagrams for Ethernet reads, writes, and slave accesses).

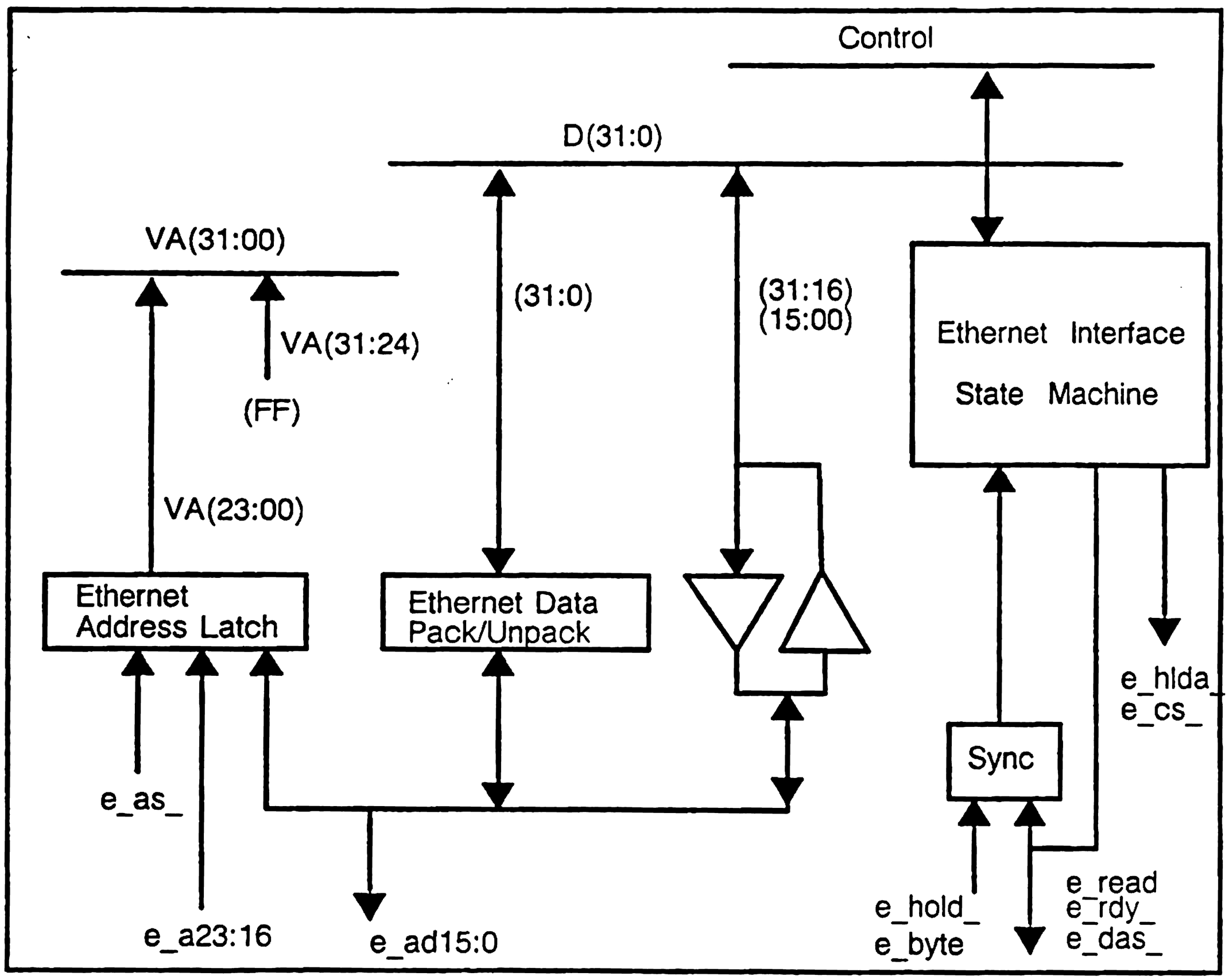
Memory errors occurring during a LANCE access will be reported over the sb_err_ signal which will consequently disable e_rdy_ generation causing the LANCE to timeout. The LANCE will then post a memory error and interrupt the CPU.

Access to registers on the LANCE is acheived, through the S4-DMA, using Slave Cycles. An address map is given below.

sb_sel_ & sb_as_	pa(X:Y)	io_a(1)	Register accessed	Size	Type
0	11	0	Register Data Port (RDP)	16-bit	R/W
0	11	1	Register Address Port (RAP)	16-bit	R/W

Once the S4-DMA has granted access of it's local bus to the LANCE, the CPU cannot access the LANCE until the pending cycles are completed. In order to remove the potential deadlock condition which results, the S4-DMA will cause a rerun according to the table on page 7.

4.1 Ethernet Interface Block Diagram



5.0 DMA Channel Architecture

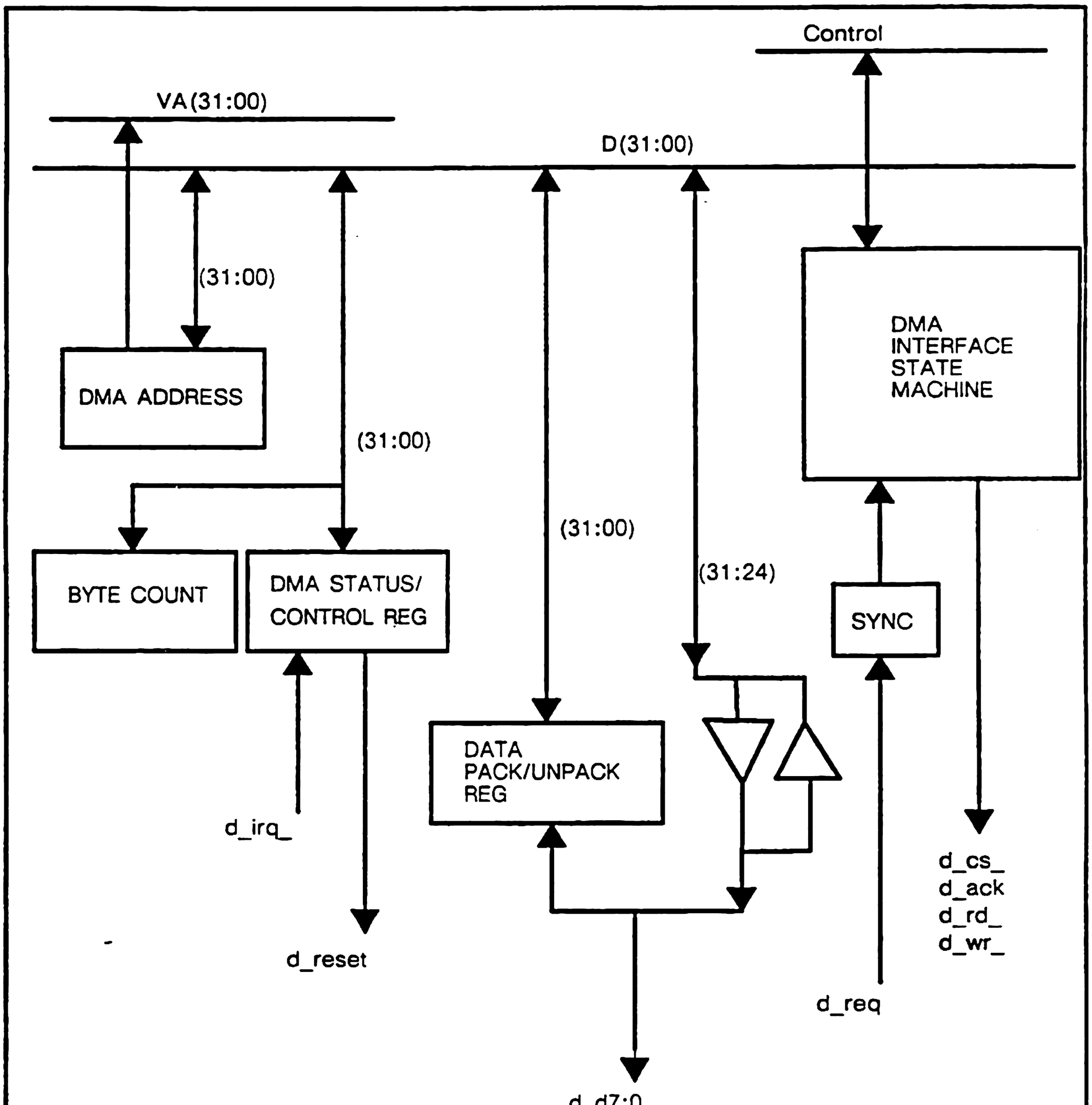
The DMA channel is designed to interface specifically to the Emulex ESP SCSI controller chip. The S4–DMA gatearray provides paths for the SCSI chip to transfer data to and from memory and for the CPU to access control and status registers on the SCSI chip. In order to support the DMA activity of the SCSI chip the gatearray contains a 24 bit address counter, and a 32 bit control/status register, aswell as a 24 bit byte counter which may be used for more generic control of future devices. The SCSI chip has a 16 bit transfer count, however the address counter supports a 24 bit count for future enhancements.

The SCSI chip has an 8 bit data bus, so in order to reduce the impact on the Sbus bandwidth, all SCSI DMA transfers go through a 32 bit pack/unpack register. All Sbus transfers are 32 bit whenever possible. On SCSI to memory transfers data is packed into a 32 bit word such that one Sbus memory cycle is required for every four SCSI transfers. Leading fragments, bytes not word aligned, are transferred to memory as bytes until they become word aligned. Trailing fragments, bytes not filling a word must be drained to memory on command. The status/control register in the S4–DMA gatearray contains commands for FLUSH, (set pack count to zero without a memory access), and DRAIN, (write remaining bytes to memory then zero pack count). The pack count is available in the status register.

Transfers from memory to the SCSI chip are completed as 32 bit reads from memory into the unpack register from where 4 consecutive bytes are unpacked and written to the SCSI chip. Every time the address crosses a word boundary a new word is read from memory, the address is increased by 4, and the pack count set to 3, (first byte consumed immediately). The next 3 byte transfers to the SCSI chip will use the data from the unpack register. Each transfer will cause the pack count to be decremented. Writing to the FLUSH bit in the control/status register, or writing a new address in the address counter will cause any data left in the unpack register to be marked invalid and a new memory access will be forced.

All external DMA signals will be two–register synchronized to avoid metastability problems.

5.1 DMA Interface Block Diagram



Access to control and status registers on the SCSI chip is acheived using the address map below.

io_a(5:0) *	Register	Size	Type
0	Transfer Count Low	8-bit	R/W
4	Transfer Count High	8-bit	R/W
8	FIFO Data	8-bit	R/W
C	Command	8-bit	R/W
10	Status/Bus ID	8-bit	R/W
14	Interrupt/Status Timeout	8-bit	R/W
18	Seq. step/Synch transfer period	8-bit	R/W
1C	FIFO flags/Synch offset	8-bit	R/W
20	Configuration	8-bit	R/W
24	Clock Conversion Factor	8-bit	W only
28	ESP TEST (chip test use only)	8-bit	R/W
2C	Configuration 2 (ESP-2 only)	8-bit	R/W
30-3C	reserved	--	--

* These addresses are external to the S4-DMA gate-array, however the gate-array does decode the appropriate higher order address bits and generate the chip select and read/write lines to the external device.

5.2 Internal Registers

pa(3:2)	Register	Size	Type
00	DMA Control/Status Register	32-bit	R/W
01	DMA Address Register	32-bit	R/W
10	DMA Byte Count * *	24-bit	R/W
11	Reserved for testing purposes	32-bit	

** The DMA Byte Count is only used when enabled by bit13 in the DMA Control/Status Register. It is intended for use when the external peripheral does not contain it's own byte count for DMA operations.

5.3 DMA Control/Status Register Assignments *(DMA_CSR)*

Bit	Mnemonic	Description	Type
0	INT_PEND	Set when d_irq_ or TC asserted. Reset when not	R
1	ERR_PEND	Set when mem. exc occurred DMA stopped Reset on FLUSH command	R
3:2	PACK_CNT	Number of bytes in Pack Register	R
4	INT_EN	When set enables d_irq_ state onto sb_irq_	R/W
5	FLUSH	When set causes PACK_CNT, ERR_PEND and TC to be reset. Reads as 0	W
6	DRAIN	When set causes remaining pack register bits to be drained to memory. PACK_CNT = 00 Clears itself	R/W
7	RESET *	When set acts as a hardware reset.	R/W
8	WRITE	DMA direction; 1= to memory 0 = from memory	R/W
9	EN_DMA	When set allows the device to respond to DMA device requests	R/W
10	REQ_PEND	When set the DMA i/f is active. DO NOT assert RESET or FLUSH	R
12:11	BYTE_ADDR	Next byte number to be accessed.	R
13	EN_CNT	When set enables the internal byte counter. (not used with the ESP SCSI chip)	R/W
14	TC * *	Terminal Count. Byte counter has expired	R
15	ILACC * * *	When set this bit instructs the ethernet interface to act slightly differently -- see note below	R/W
27:14	------	Reserved (all unused bits to read as 0)	R
31:28	DEV_ID	Device ID (for this implementation = 1000)	R

* RESET

POWER_ON RESET or RESET from bit 7 will leave the device in the following state; ERR_PEND = PACK_CNT = INT_EN = FLUSH = DRAIN = WRITE = EN_DMA = REQ_PEND = EN_CNT = TC = 0, RESET = 1, and BYTE_ADDR = 00. All interface state-machines will revert to their idle states.

* * Terminal Count bit 14 will only be cleared by either sb_reset or a FLUSH command. It can only be set when a byte count makes a transition from 000001 to 000000 and the EN_CNT bit (13) is set in the *DMA_CSR*.

* * * ILACC – This bit will normally be cleared to permit operation with the AM7990, but when set it will allow the S4–DMA gate–array to interface to the AM79C900. This is an unconfirmed option since the AM79C900 is not yet in production and hence its specification may change.

5.4 DMA Address Counter *(DMA_ADDR)*

Bit	Mnemonic	Description	Type
D(31:00)	VA(31:00)	Virtual Address for DMA access	R/W

The value in this register after a RESET is indeterminate.

The Address Counter is a 24–bit counter and an 8–bit register. The lower 3 bytes constitute the address count and the upper byte is the register. It is assumed that transfers will not cross a 16MB boundary.

5.5 DMA Byte Counter *(DMA_BCNT)*

Bit	Mnemonic	Description	Type
D(23:0)	BCNT(23:0)	Byte Count counts down to 0 (generates interrupt when 0 reached)	R/W

The value in this register after a RESET is indeterminate.

The most significant 8 bits read all 0.

5.6 Programming Notes

The address counter always points at the next memory location to be accessed. When the direction of transfer is to memory the counter is incremented by the size of the write (1 or 4) upon completion of the transfer. When the direction of transfer is from memory the address is always incremented by 4, but the lower 2 bits are driven low such that all reads are word sized and word aligned. Byte alignment is done inside the gate-array.

There is a 2-bit byte counter BYTE_ADDR that always points to the next byte location that the DMA device will access. This counter is incremented by 1 each time a byte is transferred between the external device and the gate-array. Note the byte counter is controlled by the DMA interface whereas the address counter is controlled by the memory interface, hence the two may disagree. This byte counter is loaded at the same time the address is loaded and receives the two least significant bits of the address.

Another 2-bit counter PACK_CNT keeps track of how many bytes are stored in the internal PACK register. Note this pack count is only valid for transfers to memory. Whenever the PACK_CNT= 3 and another byte is accepted, a word write is scheduled with the memory interface. If a DMA transfer completes leaving a non-word fragment in the PACK register, then this counter is used by the hardware to determine how many bytes to write to memory when the DRAIN command is received. Both PACK_CNT and BYTE_ADDR can be read in the Control and Status Register (*DMA_CSR*).

If the driver desires to terminate a transfer, two control bits in the *DMA_CSR* can be used. The EN_DMA bit can be used to ignore new transfer requests from the DMA device when it is cleared. Memory accesses by the memory interface are unaffected by this bit. Th EN_DMA bit can be set or cleared at any time without affecting the state of a transfer currently in progress. The FLUSH bit is provided to clear the PACK_CNT if the driver wishes to clean up the state of a transfer, without draining the packed data to memory. It is also used to clear the ERR_PEND indicator, allowing an error condition, which subsequently halts the DMA interface state machine, to be cleared cleanly.

Th DRAIN bit will cause all packed data to be sent to memory. This is intended for use when a transfer completes and the data for transfer to memory does not fill the 32 bit word. It can also be used to leave a transfer in a clean state if a transfer is stopped via the EN_DMA bit, which may be restarted later. A DRAIN sequence will leave the address counter pointing to the byte address beyond the last byte or word written. *Hence the address counter must be reloaded before the next transfer to properly set the BYTE_ADDR.*

The *DMA_CSR* also contains a RESET bit which will generate an external reset signal and reset all DMA interface logic (state machines). It is vital the RESET and/or FLUSH bits are not set if any memory activity is still pending: a REQ_PEND bit is provided in the *DMA_CSR* to show the driver if the memory interface is active. If REQ_PEND is asserted the driver should poll it until it is deasserted. Similarly writing to the Address Counter, changing the WRITE bit in the *DMA_CSR*, or writing the Byte Counter

when the REQ_PEND bit is set will cause unknown problems.

Interrupts from the DMA device are visible as INT_PEND in the *DMA_CSR*. The INT_EN bit is provided to enable or disable the generation of an interrupt to the IU. If an error condition exists during a memory access, (this could be parity errors, protection errors, timeouts etc.), the ERR_PEND bit will be set. This will cause an interrupt (if enabled) to the IU. Similarly eexpiration of the Byte Counter when enabled will cause the INT_PEND bit to be set an an interrupt generated. The ERR_PEND bit can only be cleared by a FLUSH command.

If it is desired to restart a transfer which was terminated in the middle, the address counter will contain the correct address, if the direction of transfer was to memory and a DRAIN was done. The address counter must be reloaded with the address to set the correct internal state for mis-aligned restarts. If the transfer was from memory, the address will be correct.

Use of Internal Byte Counter (this function is not required on the Campus I Mother board)

When using the internal Byte Counter and the TC flag in the *DMA_CSR* it is necessary to perform the following procedure for correct operation, (Note: all the above mentioned functions still apply).

Load DMA address into *DMA_ADDR*

Load Byte Count into *DMA_BCNT*

Load Peripheral Device with relevant command(s)

Load *DMA_CSR* with enables and direction bits (EN_DMA EN_CNT and WRITE)

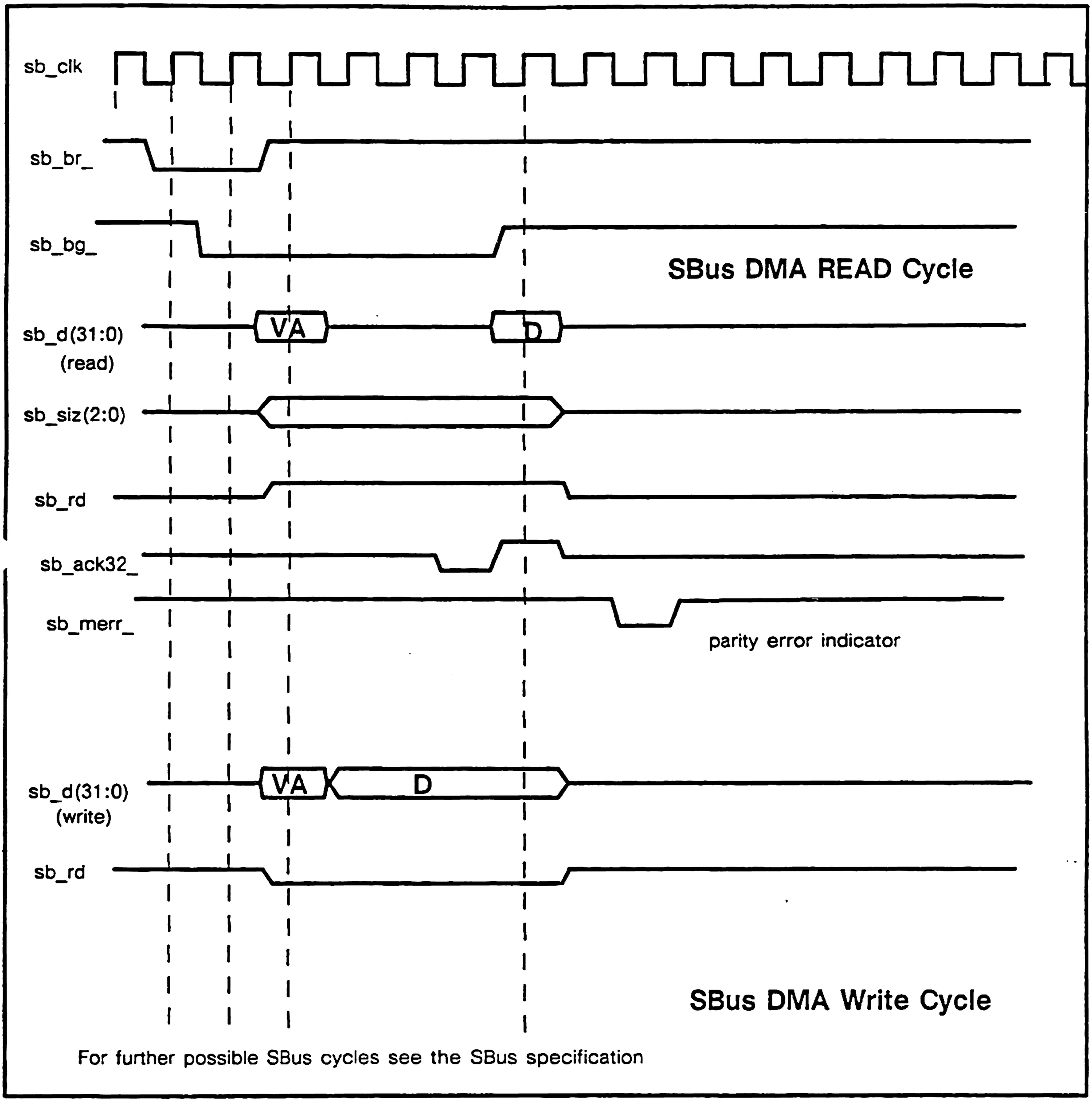
Data will be transferred as directed until the Byte Count expires, at which point the TC flag will be set in the *DMA_CSR* and an interrupt will be generated, if enabled. The Byte Counter will also be disabled at this time. Once the CPU has serviced the interrupt and wishes to start another DMA operation then it is necessary to proceed as follows;

Issue FLUSH command and then repeat the above procedure.

This time the action of writing to the *DMA_BCNT* will effectively re-enable it.

(NOTE: the above procedure is not shown to suggest that this is the only possible way to use the device, merely as an example to show that the Byte Counter will remain locked until written to again).

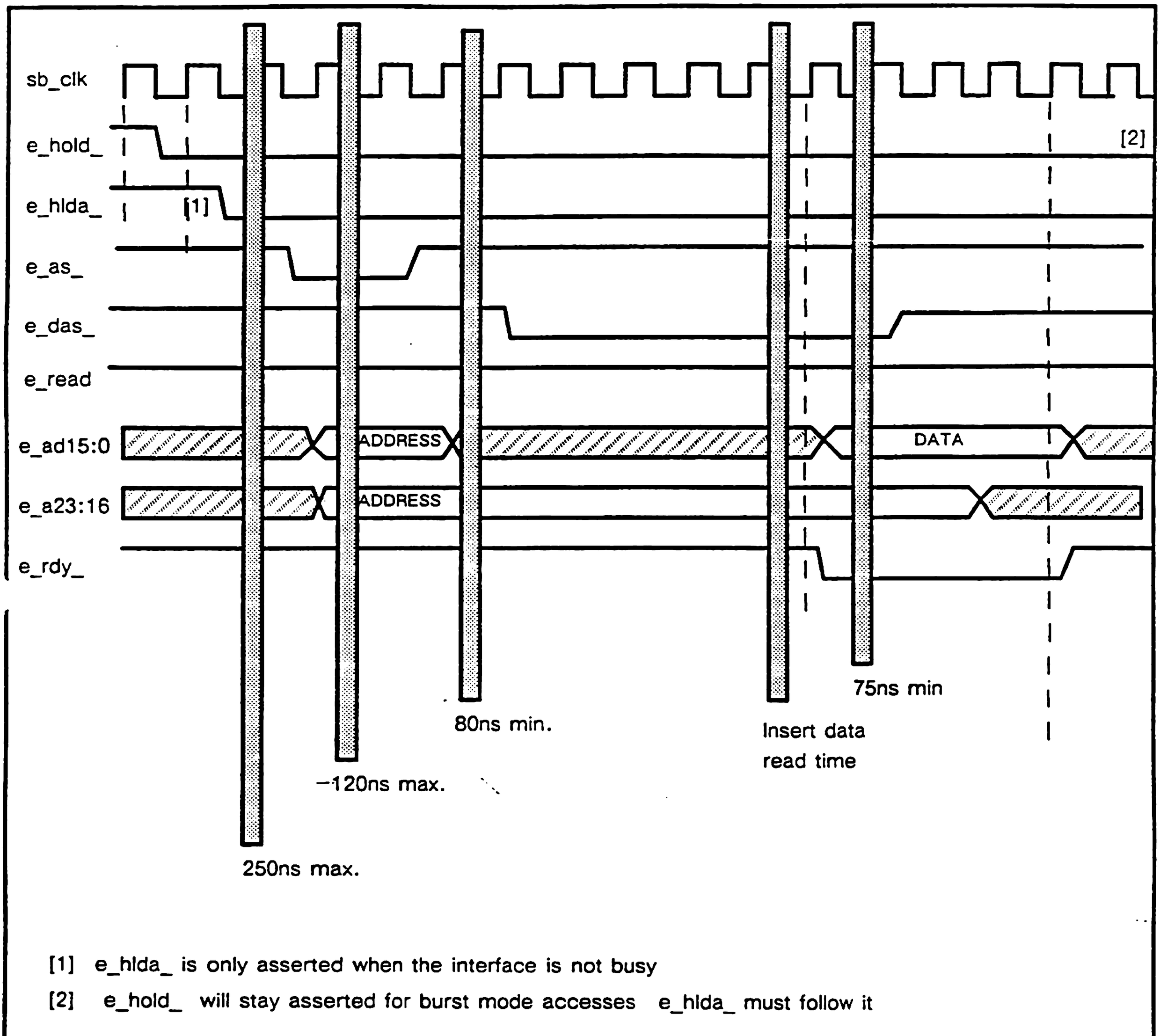
6.0 Timing Diagrams



SBus DMA READ Cycle

SBus DMA Write Cycle

For further possible SBus cycles see the SBus specification

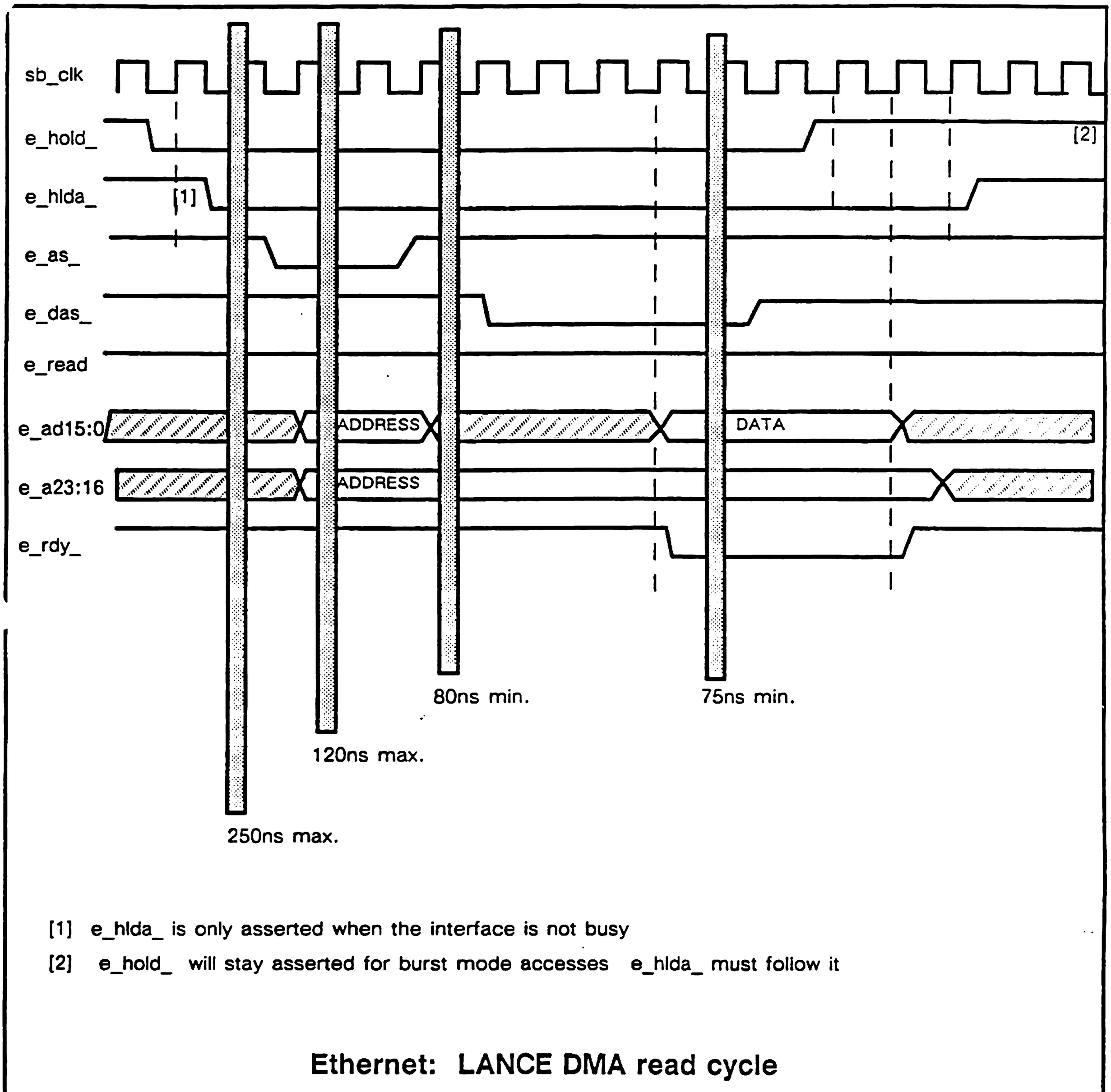


[1] e_hlda_ is only asserted when the interface is not busy

[2] e_hold_ will stay asserted for burst mode accesses e_hlda_ must follow it

Ethernet: LANCE DMA read cycle

(DATA not available in PACK Reg)

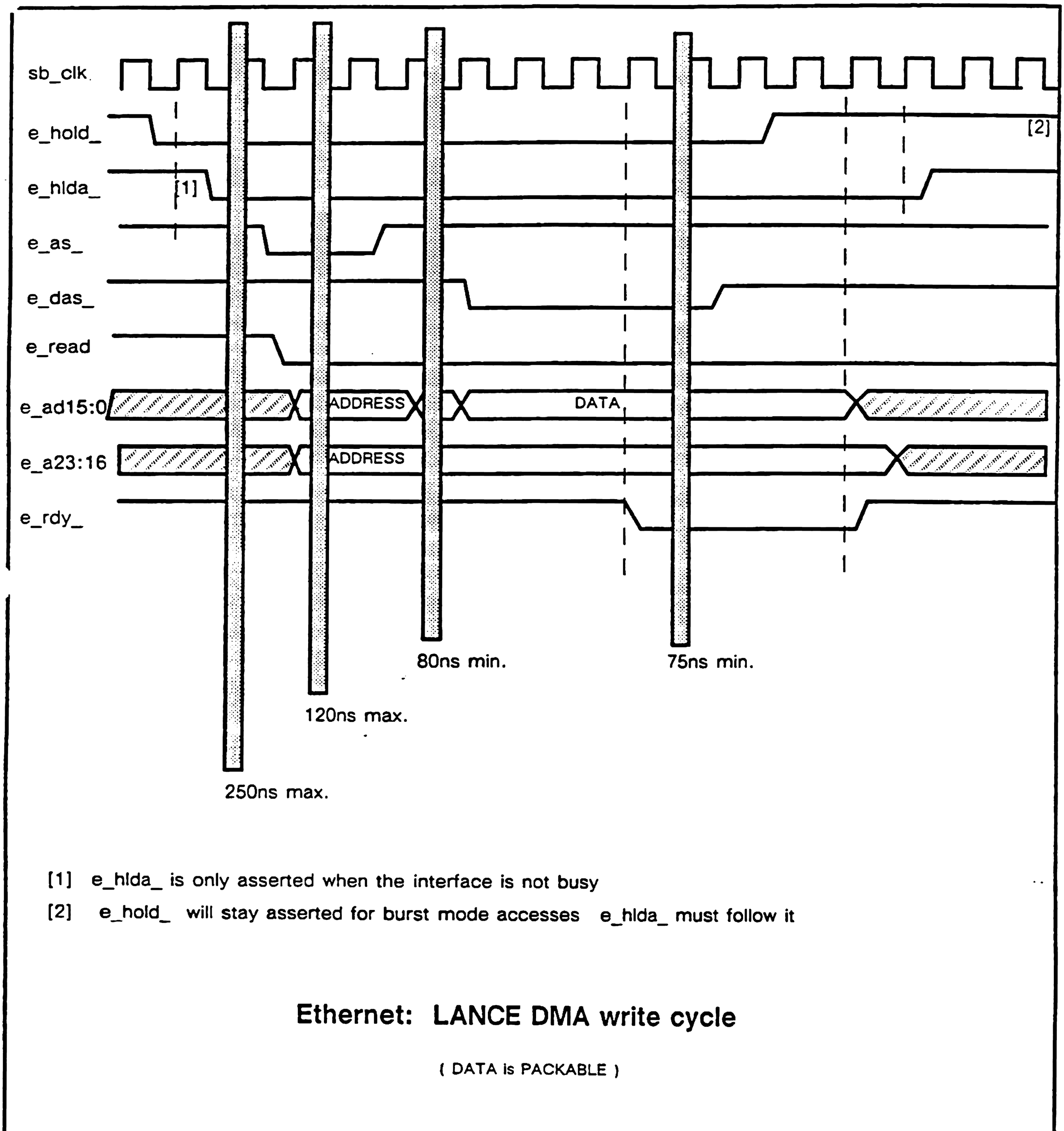


[1] e_hlda_ is only asserted when the interface is not busy

[2] e_hold_ will stay asserted for burst mode accesses e_hlda_ must follow it

Ethernet: LANCE DMA read cycle

(DATA available in PACK Reg)

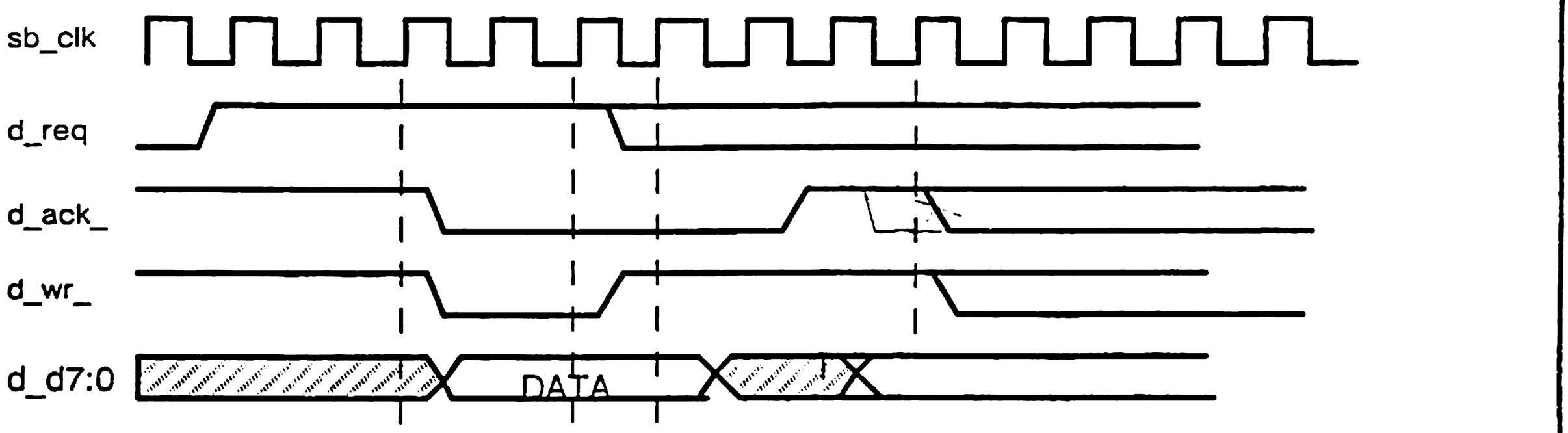


[1] e_hlda_ is only asserted when the interface is not busy

[2] e_hold_ will stay asserted for burst mode accesses e_hlda_ must follow it

Ethernet: LANCE DMA write cycle

(DATA is PACKABLE)

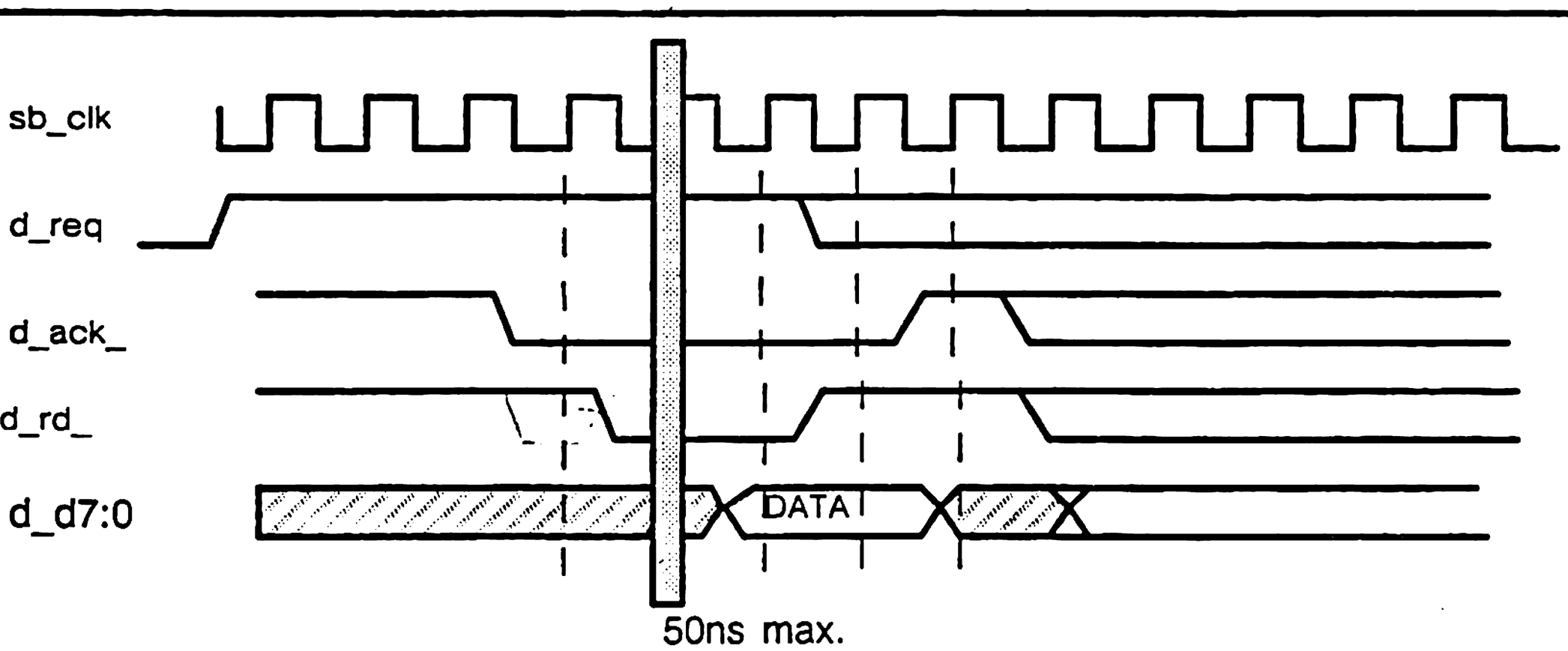


DMA Read Cycle [fast cycle]

(DATA available in UNPACK Register)

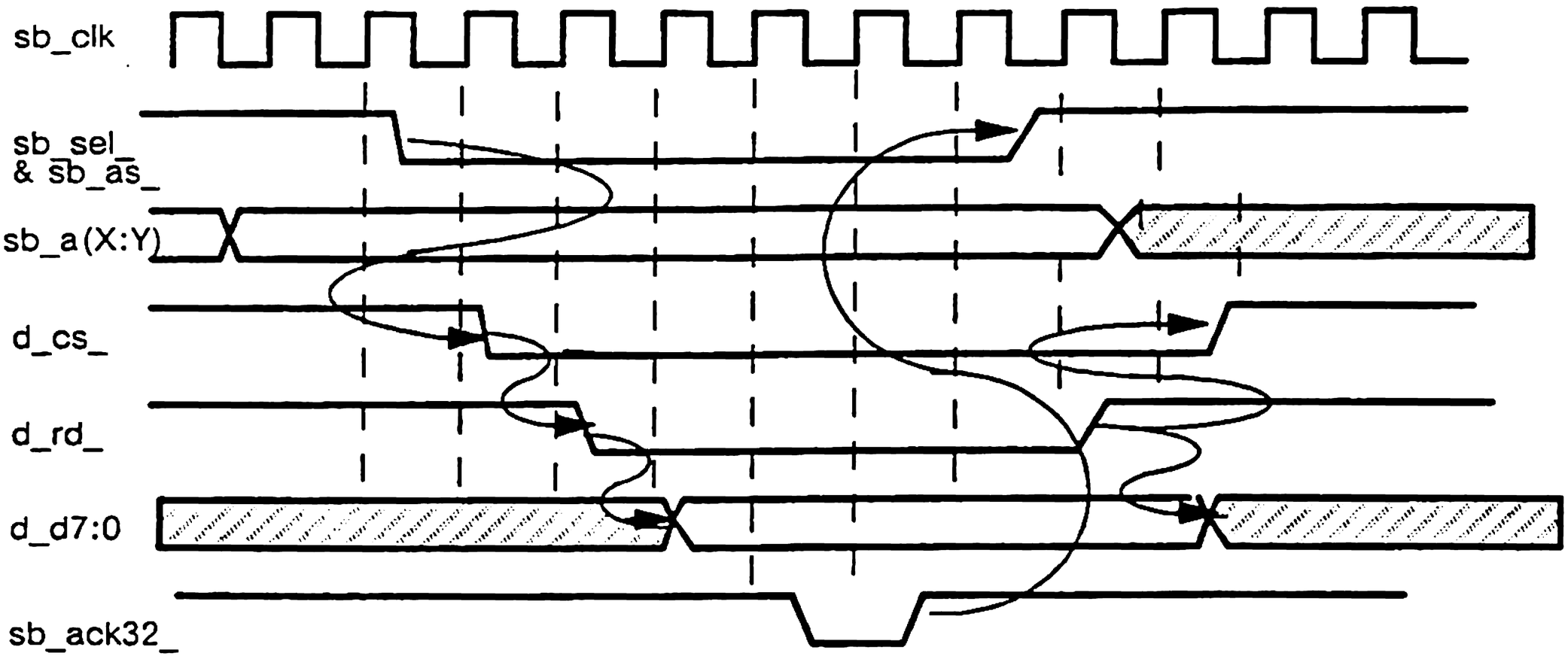
'READ' indicates transfer from memory to DMA device

When UNPACK reg is empty a memory read must occur which will subsequently lengthen this operation (see SBus READ Cycle)



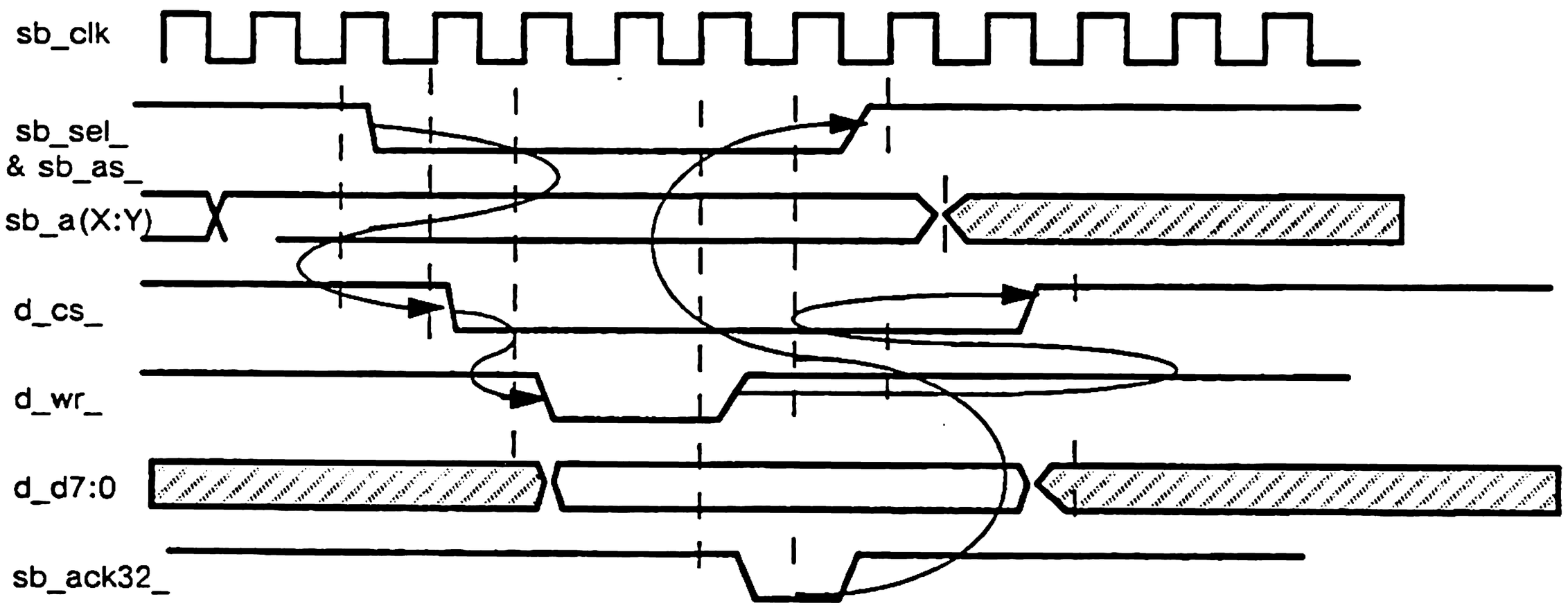
DMA Write Cycle [fast cycle]

'Write ' indicates transfer is from DMA device to memory



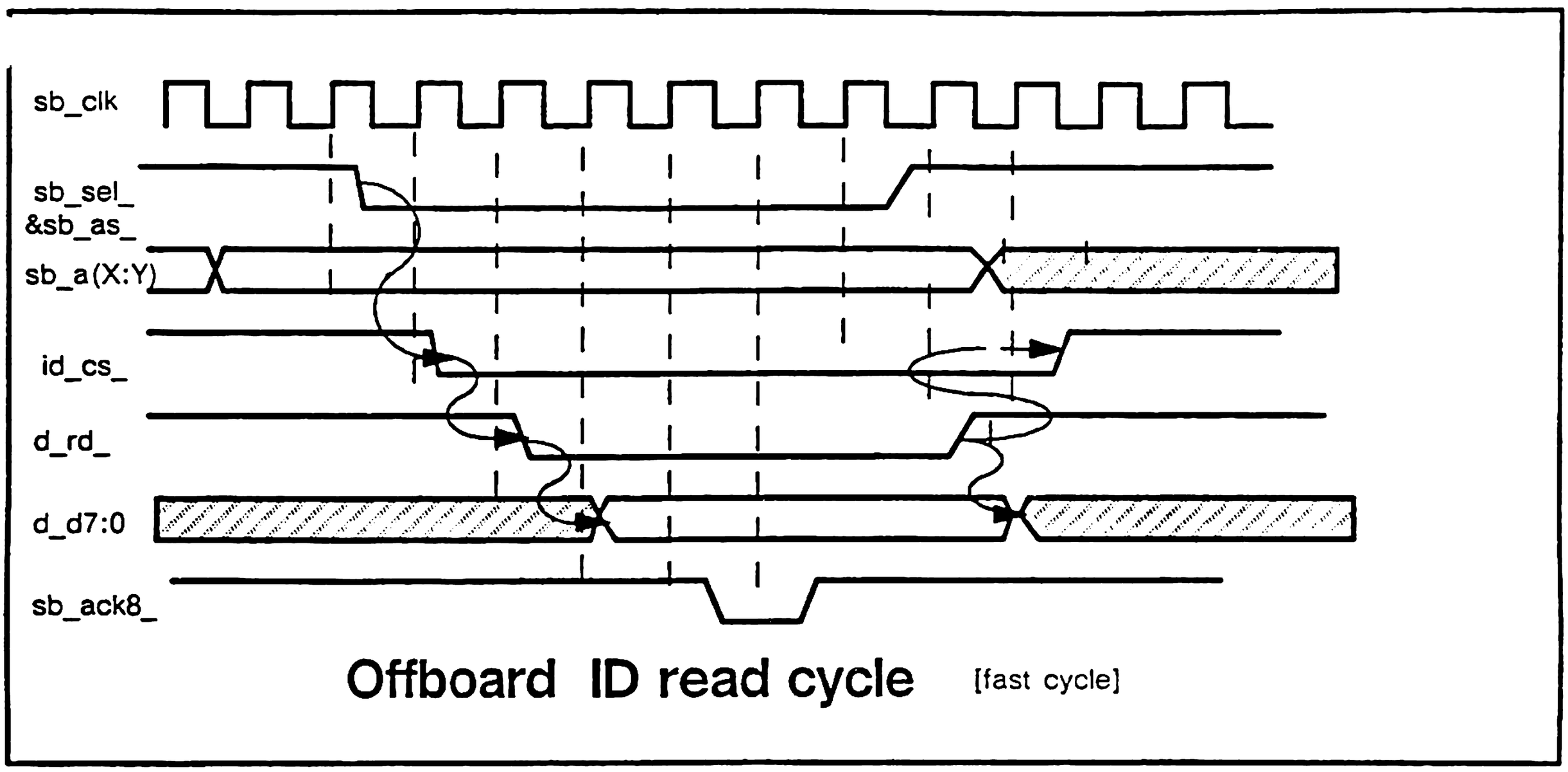
DMA device register read cycle

[fast cycle]

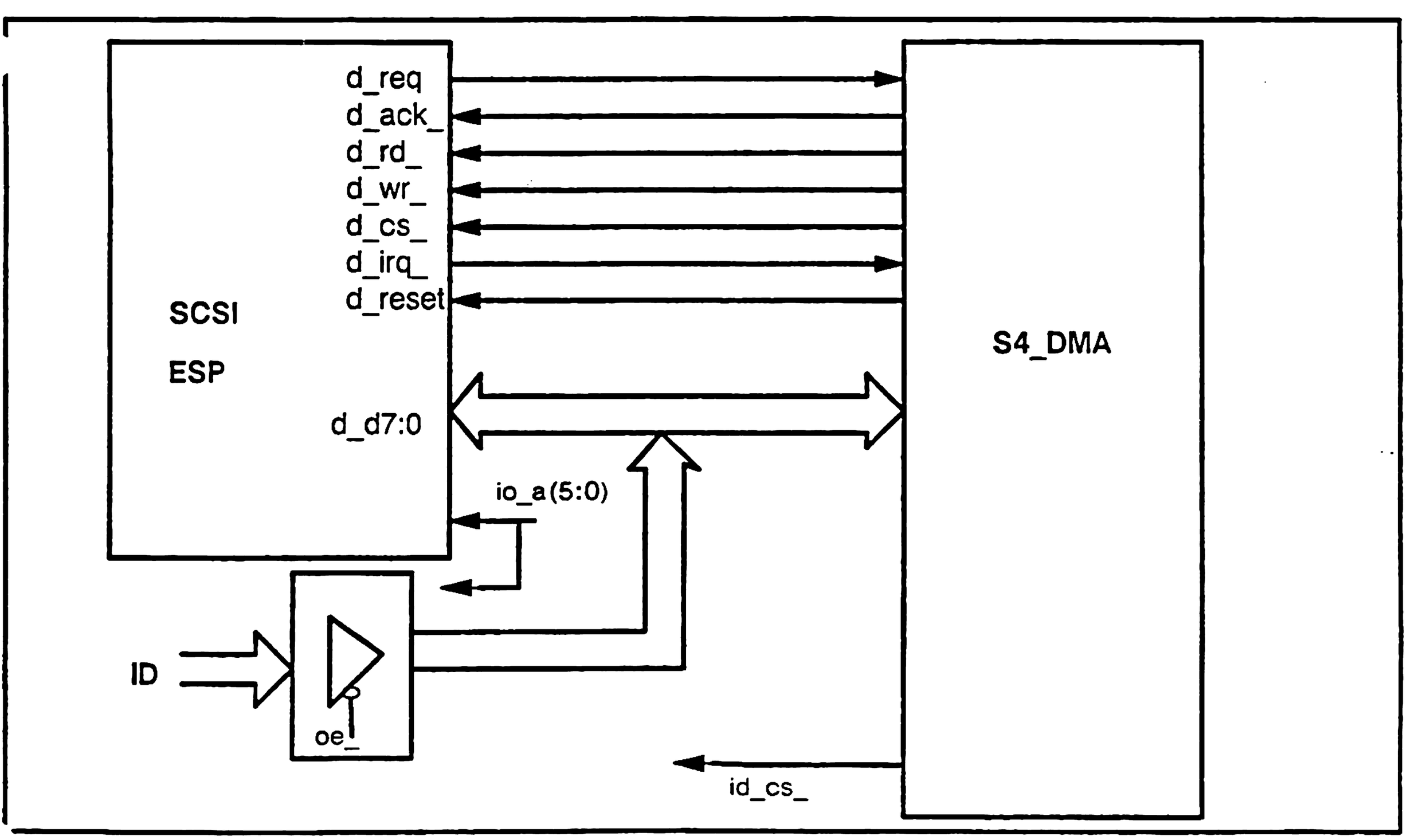


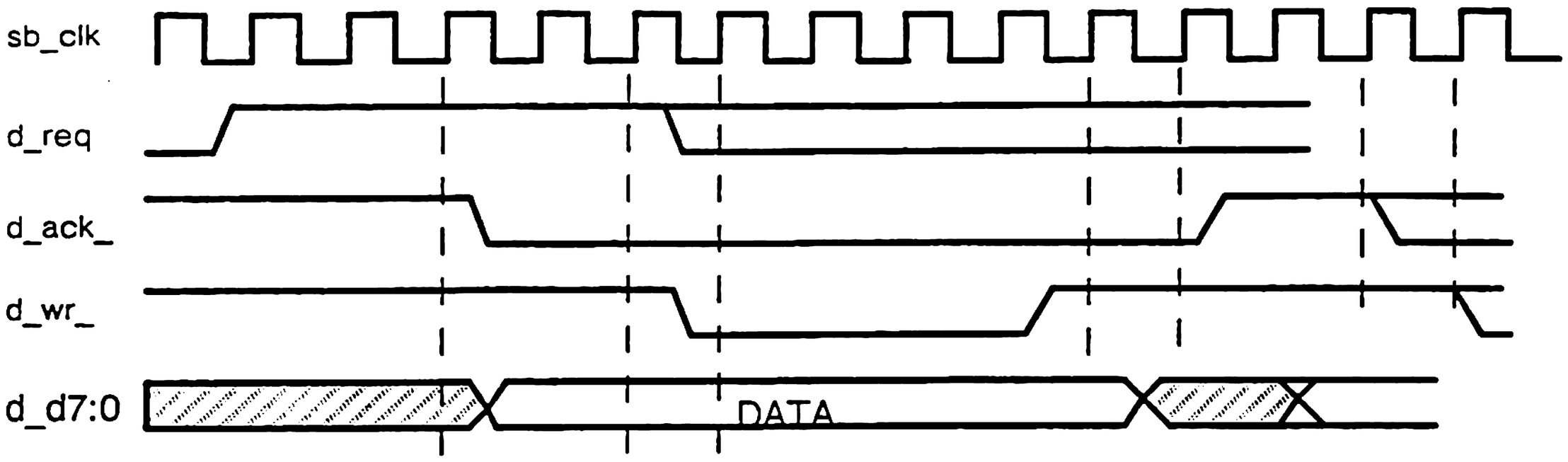
DMA device register write cycle

[fast cycle]



Offboard ID read cycle [fast cycle]

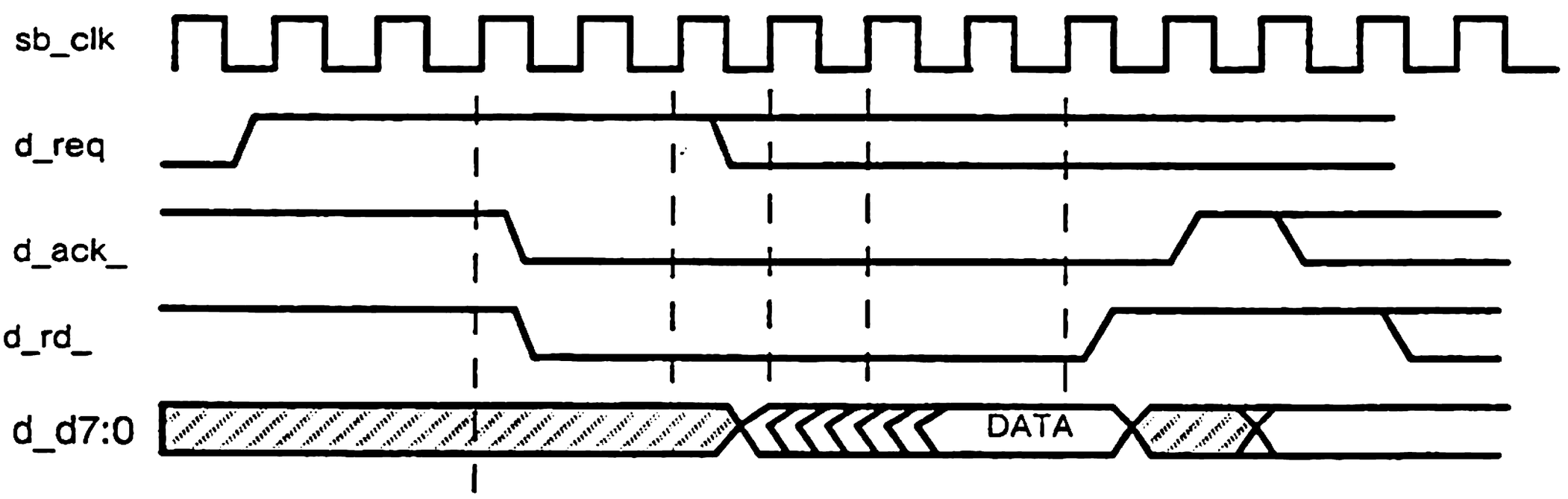




Extended DMA Read Cycle [slow cycle]

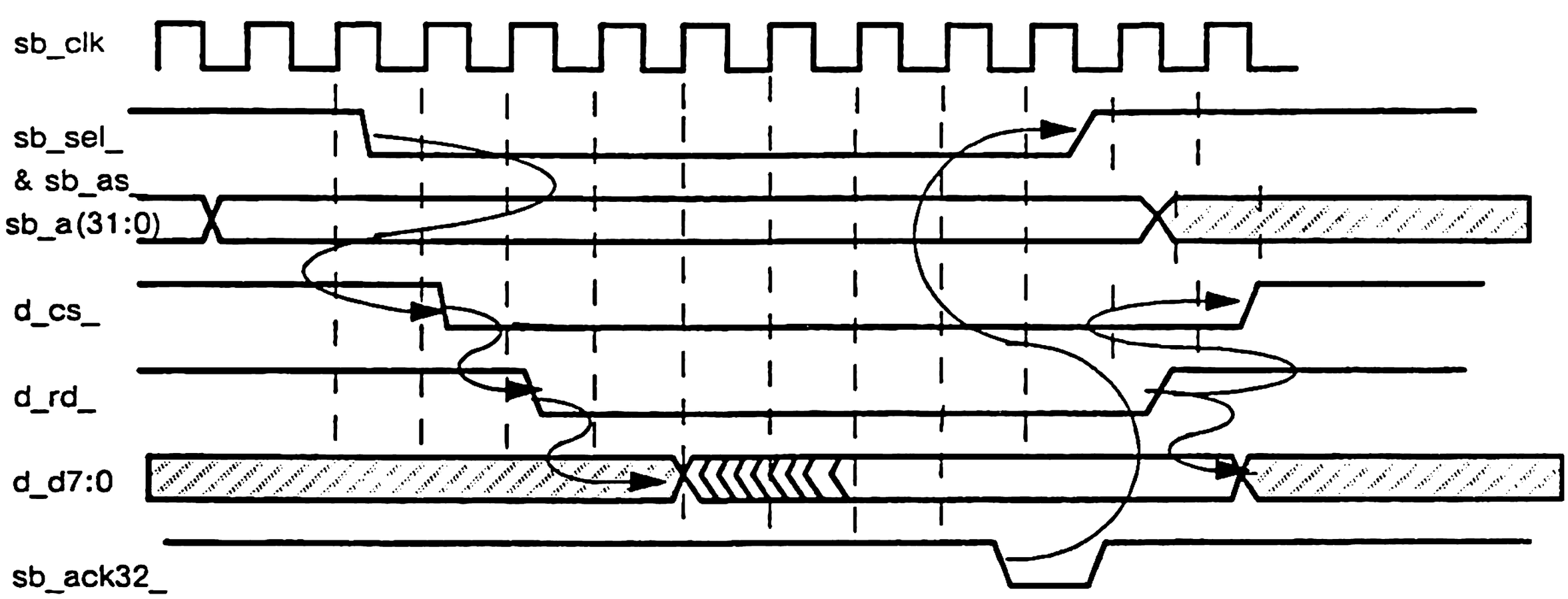
(DATA available in UNPACK Register)

'READ' indicates transfer from memory to DMA device

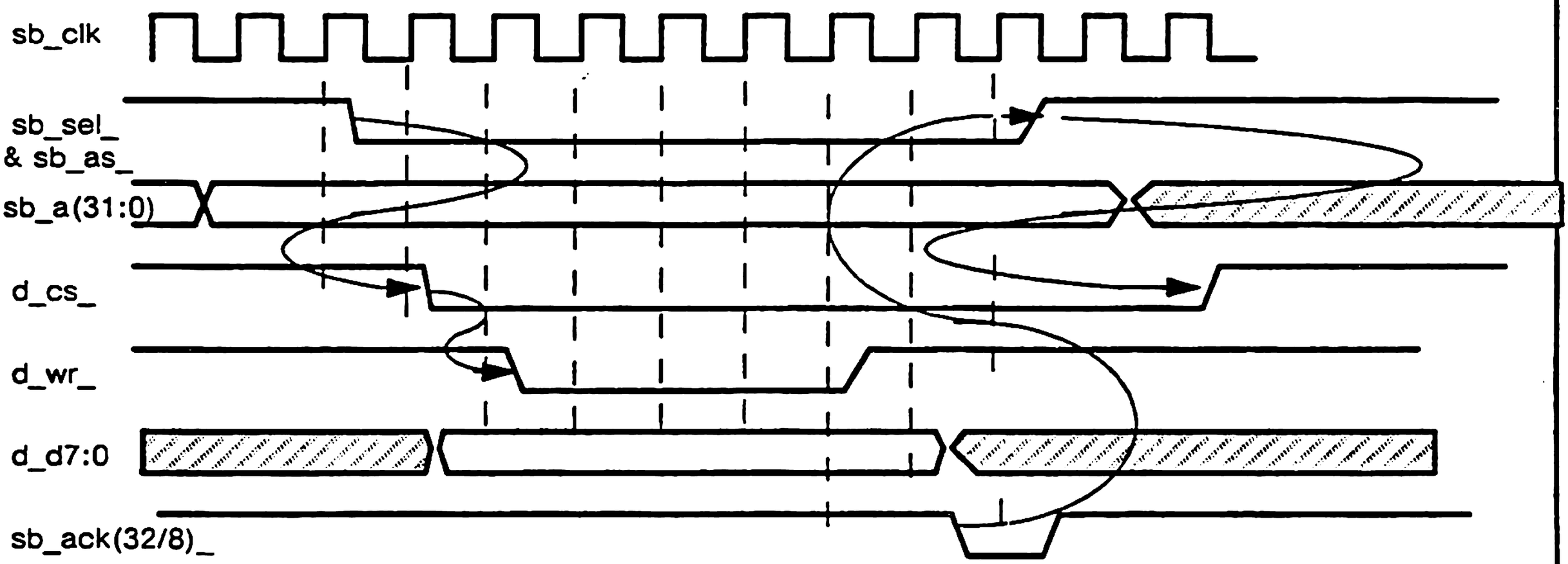


Extended DMA Write Cycle [slow cycle]

'Write ' indicates transfer is from DMA device to memory



Extended DMA device register read cycle [slow cycle]



Extended DMA device register write cycle [slow cycle]

7.0 Electrical Specification

Operating Conditions

Temperature(Ta)	0 to +70C
Supply Voltage (Vcc)	+4.75 to +5.25 V

Capacitance

Symbol	Description	min	max	unit
Cin	input capacitance	2.5	5	pf
Cout	output capacitance			pf

Switching Characteristics

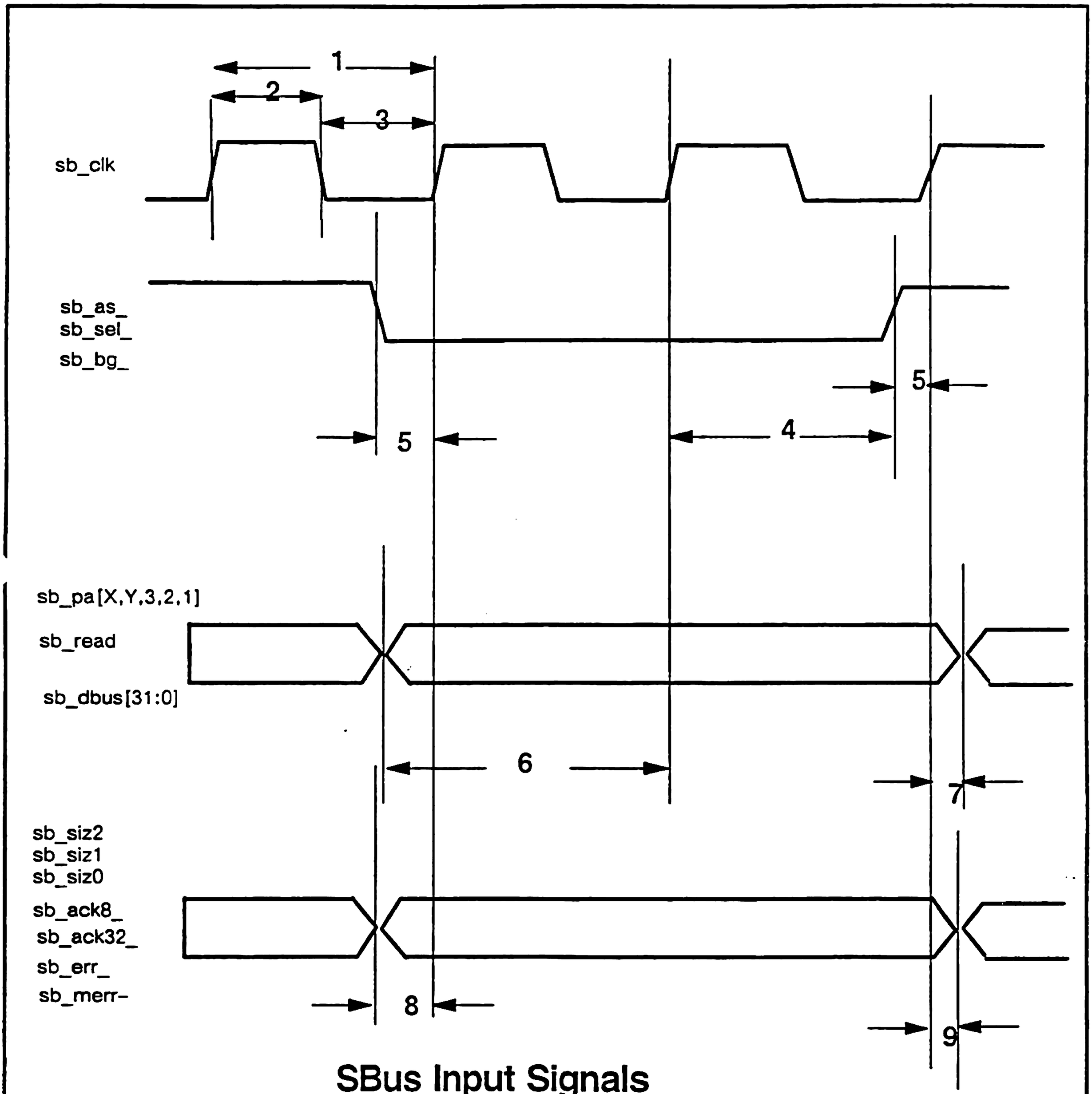
No.	SIGNAL	DESCRIPTION	CONDITIONS	min	max	units
1	CLK	clock period		30		ns
2		clock high				ns
3		clock low				ns
4	Note 1	hold wrt clk ^		0		ns
5	Note 1	setup to clk ^		14.0		ns
6	Note 1	setup to clk ^		23.0		ns
7	Note 1	hold wrt clk ^		5.0		ns
8	Note 1	setup to clk ^		13.5		ns
9	Note 1	hold wrt clk ^		0		ns
10	Note 1	clk ^ to output valid	Load = 100pf		30.4	ns
11	Note 1	clk ^ to output invalid	Load = 100pf		22.0	ns
12	Note 1	clk ^ to output valid	Load = 130pf		31.4	ns
13	Note 1	clk ^ to output invalid	Load = 130pf		19.7	ns
14	Note 1	clk ^ to output low	Load = 100pf		24	ns
15	Note 1	clk ^ to output high	Load = 100pf		18.5	ns
						ns
						ns

No.	SIGNALS	DESCRIPTION	CONDITIONS	min	max	units
16	d_req	settup to clk ^		0		ns
17	d_req	hold wrt clk ^		3.7		ns
18	d_dbus[7:0]	settup to clk ^		2.0		ns
19	d_dbus[7:0]	hold wrt clk ^		3.5		ns
20	d_irq	settup to clk ^		0		ns
21	d_irq	hold wrt clk ^		4.0		ns
22	d_rd_	clk ^ to output low	80pf		25.5	ns
23	d_rd_	clk ^ to output high	80pf		20	ns
24	d_wr_	clk ^ to output low	80pf		22.5	ns
25	d_wr_	clk ^ to output high	80pf		18.0	ns
26	d_dbus[7:0]	clk ^ to output valid	80pf		29.0	ns
27	d_dbus[7:0]	clk ^ to output invalid	80pf		21.0	ns
28	Note2	clk ^ to output low	80pf		24	ns
29	Note2	clk ^ to output high	80pf		19.0	ns
30	Note3	d_rd_ to d_dbus valid			50	ns
31	Note3	d_rd_ to d_dbus invalid		0		ns
32	d_reset	clk ^ to output low	80pf		20.0	ns
33	d_reset	clk ^ to output high	80pf		18.0	ns
34	d_ack_	clk ^ to output low	80pf		23.5	ns
35	d_ack_	clk ^ to output high	80pf		17.0	ns

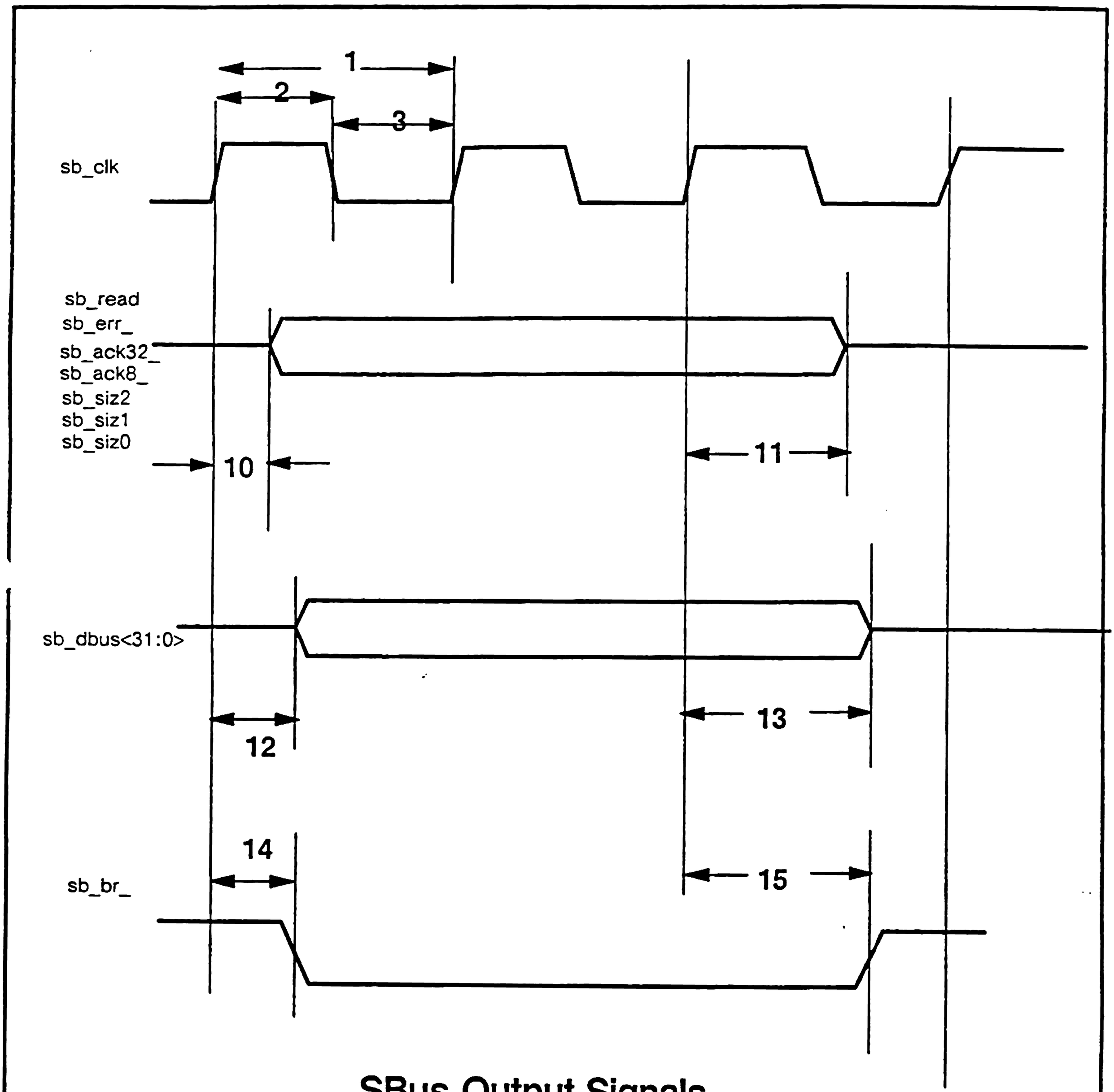
No.	SIGNAL	DESCRIPTION	CONDITIONS	min	max	units
36	e_ad[15:0]	settup to clk ^ Note4		1.0		ns
37	e_ad[15:0]	hold wrt to clk ^ Note4		4.0		ns
38	e_ad[15:0]	clk ^ to output valid	80pf		36.0	ns
39	e_ad[15:0]	clk ^ to output invalid	80pf		25	ns
40	e_hlda_	clk ^ to output high	80pf		18.0	ns
41	e_hlda_	clk ^ to output low	80pf		21.5	ns
42	e_read	clk ^ to output valid	80pf		15.5	ns
43	e_read	clk ^ to output invalid	80pf		12	ns
44	e_das_	clk ^ to output valid	80pf		23.0	ns
45	e_das_	clk ^ to output invalid	80pf		18.5	ns
46	e_rdy_	clk ^ to output valid	80pf		23.0	ns
47	e_rdy_	clk ^ to output invalid	80pf		17.5	ns
48	e_cs_	clk ^ to output high	80pf		15.5	ns
49	e_cs_	clk ^ to output low	80pf		20.0	ns
50	e_rdy_	settup to clk ^			0	ns
51	e_rdy_	hold wrt to clk ^			2.8	ns
52	e_ad[15:0]	ADDR settup to e_as_ low			15.0	ns
53	e_ad[15:0]	ADDR hold wrt e_as_ high			0	ns
54	e_hold_	settup to clk ^			0	ns
55	e_hold_	hold wrt to clk ^			4.0	ns

No.	SIGNAL	DESCRIPTION	CONDITIONS	min	max	units
56	Note1	settup to clk ^		0		ns
57	Note1	hold wrt to clk ^		3		ns
58		clk^ to output valid			15.5	ns
59		clk^ to output invalid			12.0	ns

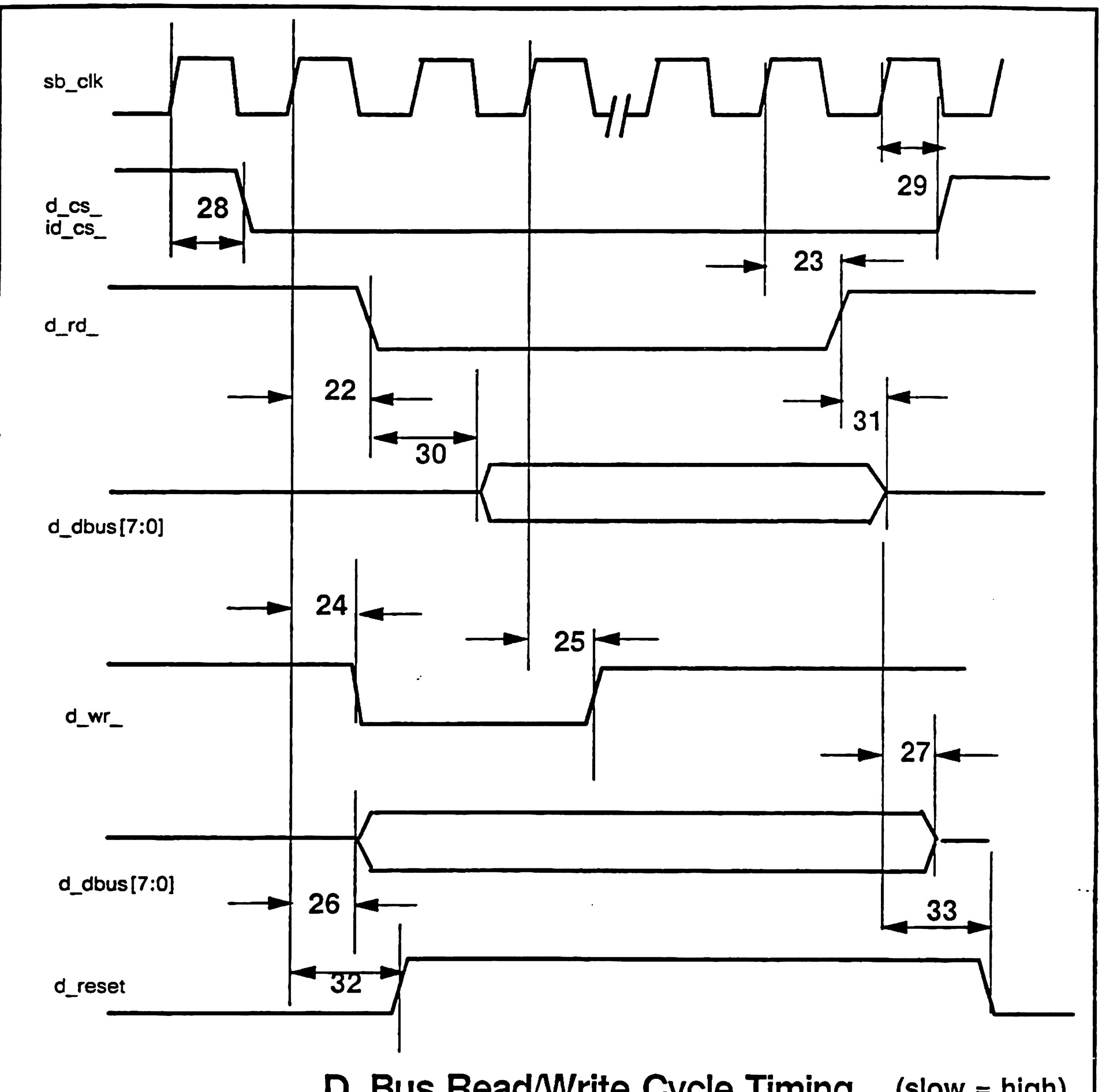
Timing Diagrams



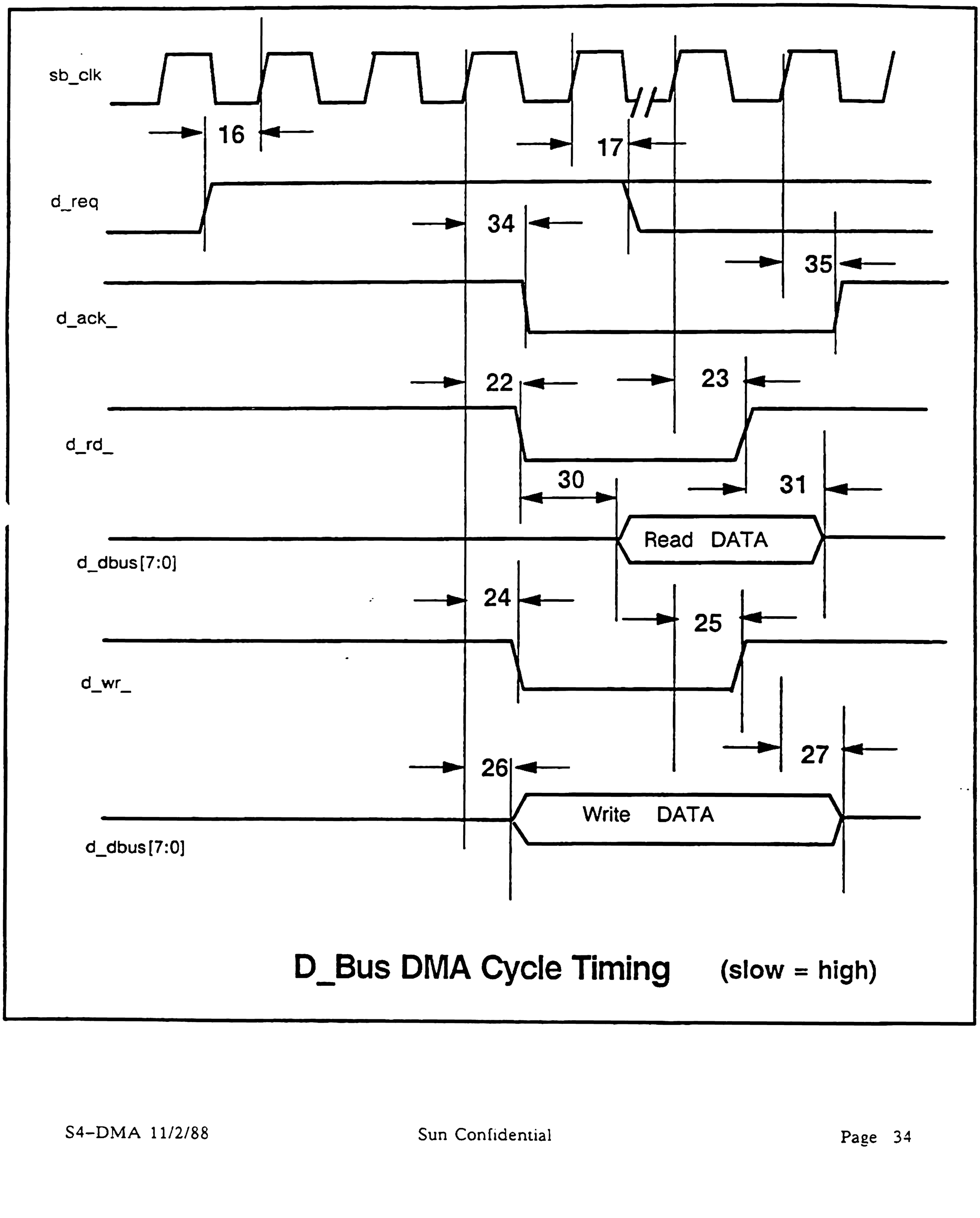
SBus Input Signals



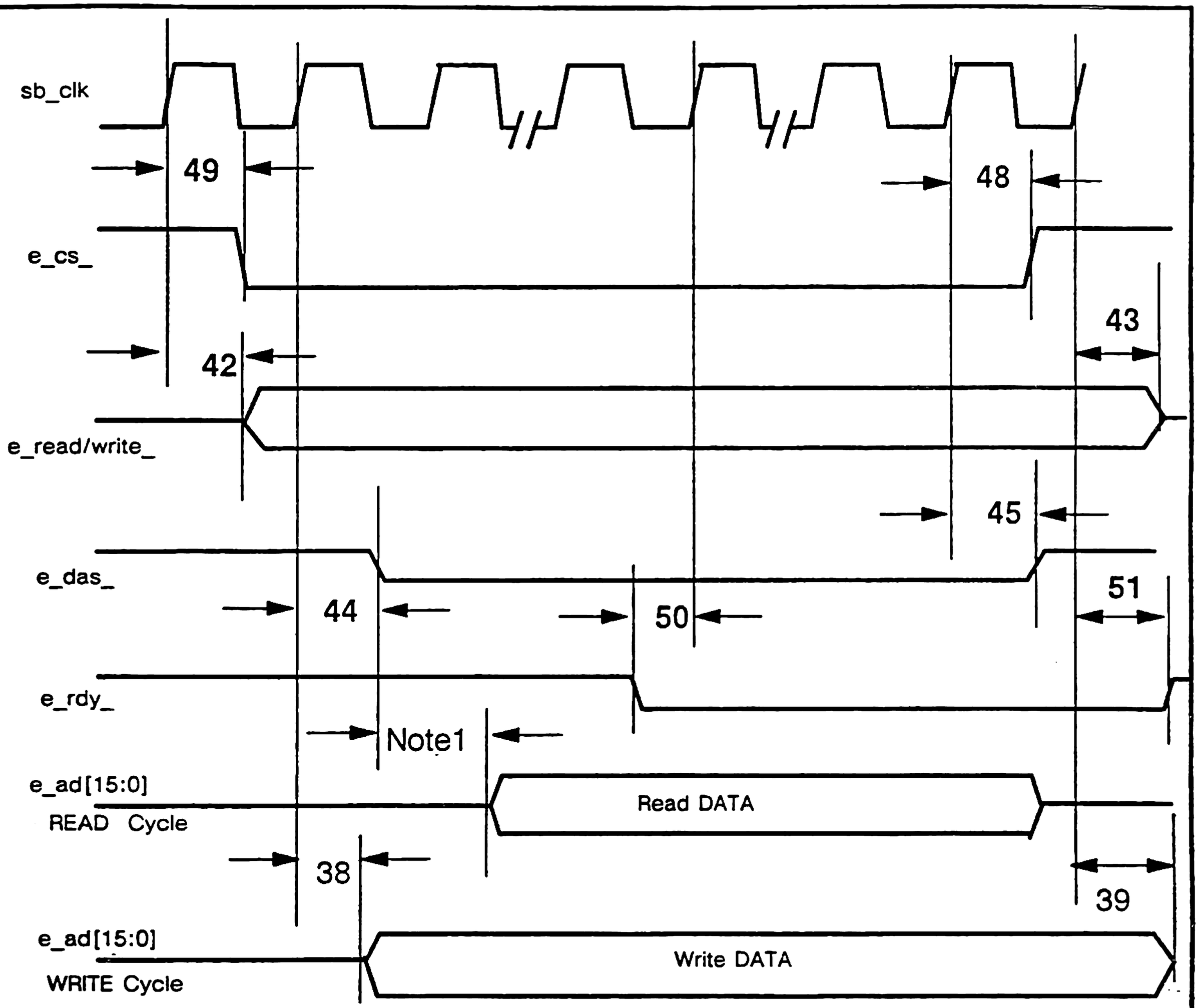
SBus Output Signals



D_Bus Read/Write Cycle Timing (slow = high)

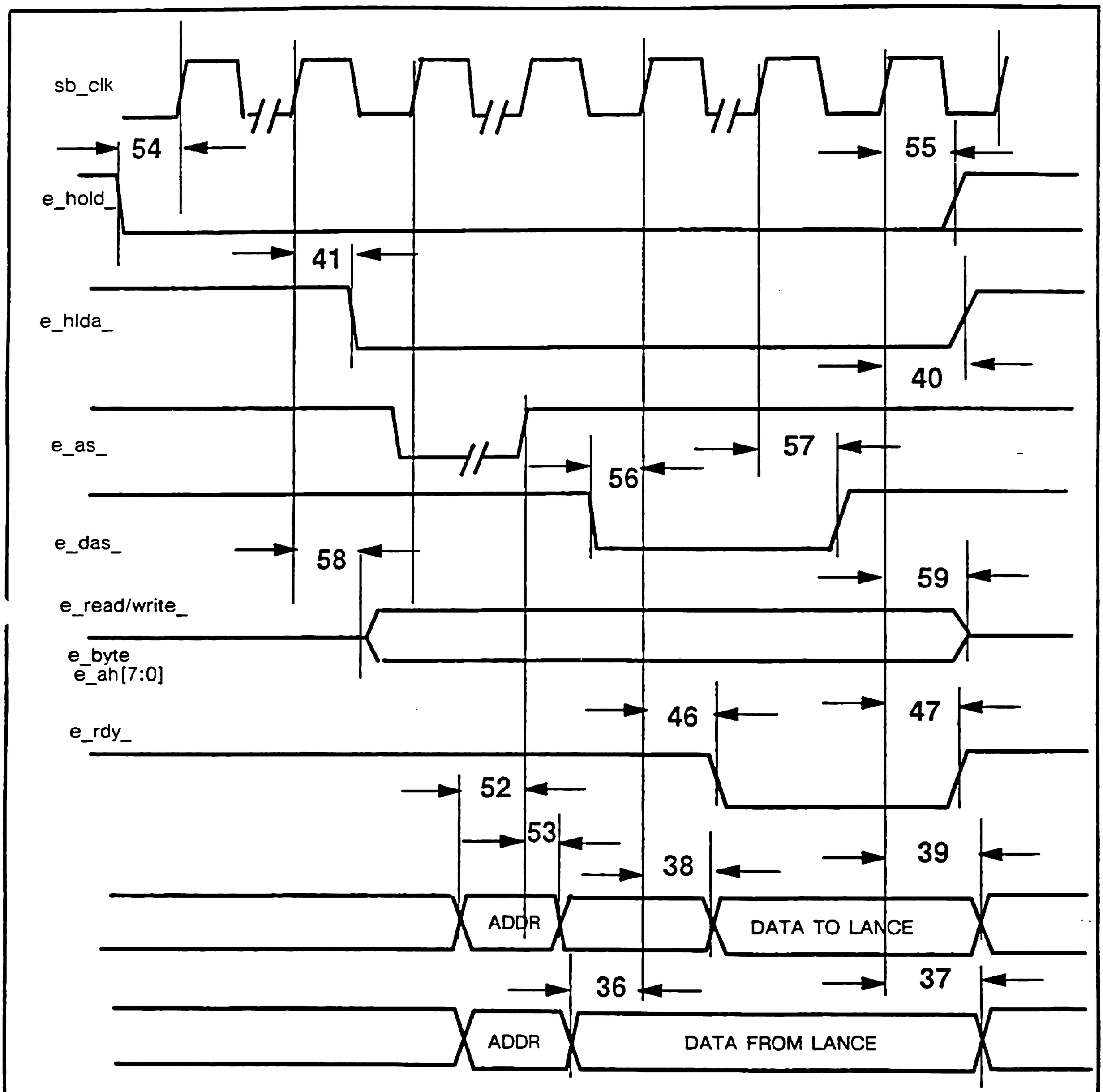


D_Bus DMA Cycle Timing (slow = high)



Note1 Refer to LANCE timing specs

LANCE READ and WRITE Cycle Timing



LANCE DMA READ and WRITE Cycle Timing

Note1: These values represent the timing characteristics of groups of signals. By referring to the Timing Diagrams it can be seen that one mnemonic value can represent many different signal paths.

Note2: These timing parameters are true for both the signals d_cs_ and id_cs_.

Note3: The documented values represent the timing of an external device (in this case the ESP SCSI chip), to which this gate-array is matched by design.

Note4: The settup and hold times refer to the timing diagram on which they are shown, and in particular to the clock edges shown. The e_ad bus is designed to be that of the LANCE Ethernet controller. Internal to this chip the e_ad bus is not latched for at least 2 clock cycles to alleviate any potential timing problems. Hence the 0ns timing requirements shown are true only if the cycle by cycle handshaking specified by the LANCE is maintained.

8.0 Revision History

12/22/87	First Release.
2/11/88	Remove sb_address bus and multiplex addr/data on sb_d bus. Add SBus Identification information.
3/28/88	Revised pinout. Corrected errors in register addressing. Added more info on programming. Updated timing diagrams. Revised block diagrams. Added register in MSbyte of ADDR_CNT. Revised operation of Terminal Count bit.
6/21/88	Added timing specs.
7/26/88	Included post_route timings
11/2/88	Changed offboard ID read response to sb_ack8_

Sun Microsystems S4 Chip Set

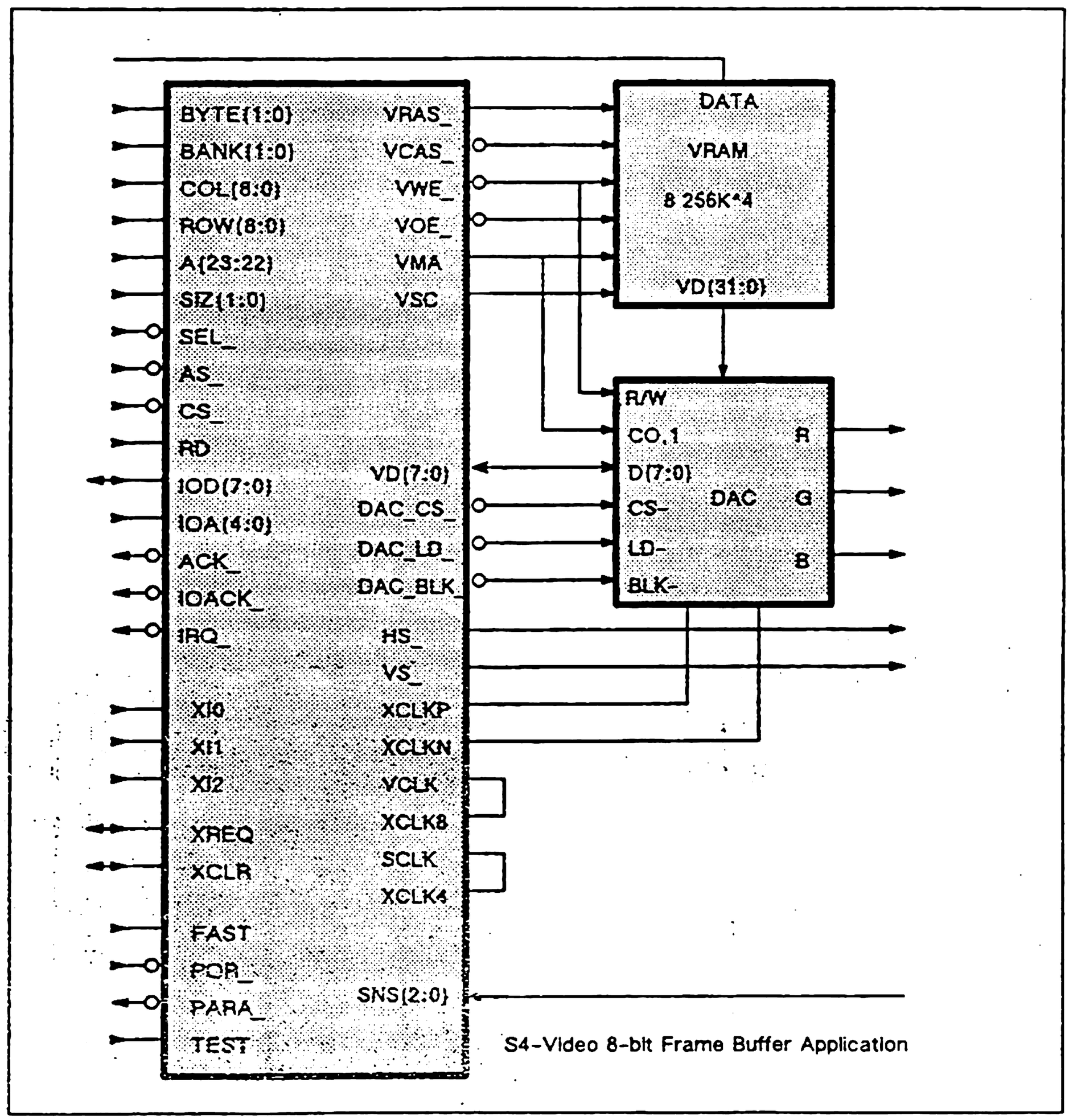
Sun SPARC(TM) Support System

S4-Video

Preliminary

Features

- Single-chip video subsystem
- Directly interfaces to Sbus Interface
- Supports 256*4, 128K*8, and 64K*4 Video RAM
- Supports 1-bit, 8-bit, and 24-bit per pixel frame buffers
- Fully programmable video timing and resolution
- Supports up to 4 video clocks (software selectable)
- Supports Sun Video Monitor sense lines (for auto configuration)
- Directly interfaces to VRAM and RAMDAC (no external components required)
- Built-in Video Shifter for 1-bit frame buffers (maximum pixel clock 100 MHz)
- Low-cost 120PFP package



S4-Video 8-bit Frame Buffer Application

Pin Description

Name	Type	Description
SBus Interface	*47*	
clk	TLCHT	system clock
siz(1:0)	TLCHTU	size of writes. 0:4 bytes, 1:1 byte, 2:2 bytes, 3:3 bytes.
byte(1:0)	TLCHT	Byte Address
bank(1:0)	TLCHT	Bank Address
col(8:0)	TLCHT	Column Address
row(8:0)	TLCHT	Row Address
a(23:22)	TLCHT	Function Select
ioa(4:0)	TLCHTU	5-bit I/O Address
iod(7:0)	BD4TU	8-bit I/O Data Bus
as_	TLCHN	address strobe
cs_	TLCHN	chip select
rd	TLCHTU	read cycle
ack_	BT4	Video RAM acknowledge
ioack_	BT4	Input/Output acknowledge
irq_	BT4OD	Interrupt Request (active low open drain)
b_err_	BT4	Buss Error
Clocks	*10*	
xi(2:0)	TLCHT	Pixel Clock In (3:0)
xclkp	B4	Pixel Clock Out positive (to DAC)
xclkn	B4	Pixel Clock Out negative (to DAC)
xclk4	B4	Pixel Clock Out X4
xclk8	B4	Pixel Clock Out X8
xclk16	B4	Pixel Clock Out X16
xvideo	B4	Pixel Data Out
Video Controller/DAC	*19*	
sclk	TLCHT	Video RAM Shift Clock Input (X4, X8, or X16)
vclk	TLCHT	Video Controller Clock Input (X8)
hs_	BT4	Horizontal Sync Output
vs_	BT4	Vertical Sync Output
x_cs_	BD1TU	External Chip Select
sns(2:0)	TLCHTU	Monitor Sense Inputs (2:0)
vd(7:0)	BD1TU	Video/DAC Data Bus
dac_cs_	BD1TU	DAC Chip Select Output
dac_ld_	BT1	DAC Load Output
dac_blk_	BT1	DAC Blank Output
xreq	BD1TD	External Transfer Request
xclr	BD1TD	External Transfer Counter Clear

VRAM Interface	*20*	
vma(8:0)	BT4	Video Multiplexed Address
vcas(3:0)_	BT4	Video Cas Enable (Byte select)
vras(3:0)_	BT4	Video Ras Enable (Bank select)
voe_	BT4	Video Output Enable
vwe_	BT4	Video Write Enable
vsc	BT4	Video Shift Clock
Misc Pins	*8*	
mode(3:0)	TLCHTD	memory mode and configuration
type	TLCHTD	VRAM type: 0:256K, 1:1Mbit
por_	TLCHNU	Power On Reset. Clears control register
para_	BD1TU	Parametric Test Output and Output Disable

Device Number:	LMA9141	(PAD: 118, VDD:6 VSS:8, IO:104)
Package Type:	PFP120	

Overview

The S4-Video ASIC has the following functional blocks:

- Sbus Interface
- Video Clock
- Video Shift Register
- Video Controller/DAC Interface
- Video RAM Interface

Each of these blocks is described below in detail.

Sbus Interface

The Sbus interface is those set of signals that interface to the Sbus. The wiring of these signals to the Sbus depends on the specific frame buffer configuration. This is further explained in the section "Frame Buffer Configuration" below. Note that dependent on configuration some Sbus interface signals will not be used or share one Sbus signal. Unused inputs should be grounded.

Pixel Clock

The Video Clock section handles the generation of the DAC Clock, the Video RAM Shift Clock, and the Video Controller Clock. The Input to this section is up to 4 crystal clock oscillators which are selectable and sub-dividable under software control to from a particular pixel clock. The Pixel Clock is output differentially for the DAC and in X4, X8, and X16 form for the Video RAM Shift Clock and the Video Controller Clock. For ease of testing and configuration, the Shift Clock and the Video Controller Clock are externally connected to the SCLK and VCLK inputs, respectively. The maximum video clock input is 100 MHz.

Video Shift Register

The Video Shift Register generates a serial video data stream for 1-bit frame buffer applications. This function is enabled when the DAC_EN input is false (active high). This turns the VD(7:0) data bus into an 8-bit parallel bus that is loaded into the video shift register synchronously with the X16 rising clock edge. The load function is suppressed if the BLANK output from the video controller is active and the shift register will continue to shift out "0" instead.

Video Controller and DAC Interface

The video controller section provides a set of registers that provide Sbus identification, monitor status, video scanning status, and the video timing parameters. The video controller is programmed by software during initialization to provide the actual video timing that is desired. The DAC Interface provides all signals and appropriate timing for the external DAC.

VRAM Interface

The Video RAM interface generates the correct timing and the proper signals for operating the VRAM. The on-chip drivers are suitable to directly drive VRAM, no external components are required.

Address Space Decoding

This section explains the S4-Video registers and decoding. On Reset, the master control register is initialized to 0. All other registers are not initialized.

Master Decoding (A23:22)

When the S4-Video chip is selected, address A(23:22) define the master mode as follows:

0x000000-0x3FFFFF	Sbus ID
0x400000-0x7FFFFF	Video Registers
0x800000-0xBFFFFF	Video RAM
0xC00000-0xFFFFFF	Reserved

Sbus ID

The Sbus ID is either internal in the S4-Video chip or provided externally, as determined by the status of X_CS_ at the end of POR_. If X_CS_ is grounded externally, the ID will be provided internally and read as 0xFE01010y, where (y) is MODE(3:0). If X_CS_ is not grounded externally then the Sbus ID will be provided by an external PROM that is selected by X_CS_. The PROM can have a size up to 4 MBytes.

Video Registers

The video registers start at the four megabyte (0x400000) boundary and extend up to the 8 megabyte boundary. There are a total of 16 registers, including the external DAC, which are decoded with IOA(4:0).

Video RAM

The frame buffer is decoded at the eight megabyte boundary (0x800000) up to the twelve megabyte boundary (0xC00000). It is up to software to map in only memory that is physically present on the frame buffer.

Reserved

Accessing this area will return an ACK but cause no actions on the chip. This area can be used to provide "dummy pages" for software.

Interrupt

When enabled, Interrupt is asserted at the beginning of vertical blank. The interrupt is cleared by writing to the (read-only) status register.

Video Register Decoding

This section includes all the video registers and provides decoding for the external DAC.

0x400000 DAC Select (A2=C0, A3=C1)

0x400010 Master Control Register (Read-Write, Cleared on power-on-reset)

D7: Interrupt Enable
D6: Video Enable
D5: Timing Enable/Slave Mode
D4: Cursor Enable Register
D(2:3) Oscillator Select (2:3 = 1, 2, 3 selects xi(1), xi(2), xi(3).
2:3 = 4 stops video clocks)
D(0:1) Divider (1:1,2,3,4)

0x400011 Status Register (Read-Only, Writing clears interrupt)

D7: Interrupt Pending
D6: Monitor Sense (2)
D5: Monitor Sense (1)
D4: Monitor Sense (0)
D3: Memory Type (3)
D2: Memory Type (2)
D1: Memory Type (1)
D0: Memory Type (0)

0x400012 Cursor Start Address
0x400013 Cursor End Address

0x400014	HBS	Horizontal Blank Set
0x400015	HBC	Horizontal Blank Clear
0x400016	HSS	Horizontal Sync Set
0x400017	HSC	Horizontal Sync Clear
0x400018	CSC	Composite Sync Clear
0x400019	VBSH	Vertical Blank Set High Byte
0x40001A	VBSL	Vertical Blank Set Low Byte
0x40001B	VBC	Vertical Blank Clear
0x40001C	VSS	Vertical Sync Set
0x40001D	VSC	Vertical Sync Clear
0x40001E	XCS	Transfer Cycle hold off Set
0x40001F	XCC	Transfer Cycle hold off Clear

Video Register Description

DAC Select

Accesses to these addresses are passed through to the external DAC for reading and writing of the DAC registers.

Master Control Register

Bit	Function
7	Enables interupts. When enabled, the S4-Video chip will generate an interupt when the end of the end of the frame is reached (start of VBLANK). The interupt is cleared by reading the status register.
6	Video Enable. When set to 0, the blank output is constantly asserted independent of the internal counters. When set to 1, the VBLANK output follows what is programmed into the timing registers.
5	Timing Enable/Slave Mode. When set to 0, the internal video timers are disabled and the internal state-machine that controls the transfer cycles is triggered from the external inputs XREQ and XCLR. When set to a 1, the video chip generates timing based on the values programmed and drives the XREQ and XCLR pins as outputs.
4	Cursor Enable Register. When set to a 1, accesses to the frame buffer will cause a buss error if the address is within the range of the two address values programmed into the Cursor Start Address and the Cursor End Address Registers located at 0x400012 and 0x400013.
2:3	Oscillator Select. Selects one of the three XI inputs as the source for the video timing. Selecting input 4 (2:3 = 0x11) causes the video logic to stop.
0:1	Divider. Selects a divide by 1, 2, 3 or 4 of the selected XI input.

Status Register

Bit	Function
7	Interupt Pending. An interupt was generated by the chip.
4:6	Monitor Sense. These three bits come directly from the three SNS inputs to the S4-Video chip. Usefull for determining the type of monitor connected then frame buffer.
0:3	Memory Mode. These four pins come directly from the MODE inputs the S4-Video chip. Usefull for determining what type of memory the frame buffer uses.

Cursor Start Address

Defines the lower address limit that will generate a buss error when control register bit 4 is set. The eight high order bits of the row address are compared against this value. If the row address is equal to or greater than the value in the cursor start address register (but not greater than the value in the cursor end address register) then accesses to that address will generate a bus error.

Cursor End Address

Defines the upper address limit that will generate a buss error when control register bit 4 is set. The eight high order bits of the row address are compared against this value. If the row address is equal to or less than the value in the cursor start address register (but not less than the value in the cursor start address register) then accesses to that address will generate a bus error.

HBS

The HBS register contains the horizontal blank start value. It is programmed in multiples of 8 pixels. When a horizontal sweep begins, the internal horizontal timing counter starts a zero, counts up to HBS, then starts over. When the HBS value is reached, the composite blank output (DAC_BLK_) goes active.

HBC

The HBC register contains the horizontal blank end value. It is programmed in multiples of 8 pixels. When the horizontal counter reaches this value, the composite blank (DAC_BLK_) goes inactive. The value for HBC must be less than the value for HBS.

HSS

The HSS register contains the horizontal sync start value. It is programmed in multiples of 8 pixels. When the horizontal counter reaches HSS, the horizontal sync output (HS_) goes active. HSS should be programmed to go active within the blanking envelope defined by HSS and HSC.

HSC

The HSC register contains the horizontal sync end value. It is programmed in multiples of 8 pixels. When the horizontal counter reaches HSC, the horizontal sync output (HS_) goes inactive. HSC must be programmed to a value greater than HSS, and should be less than HBC. A basic horizontal sweep with respect to the horizontal counter should look something like:

0....HSS....HSC....HBC.....HBS

CSC

The CSC register contains the composite sync clear value. it is programmed in multiples of 8 pixels. During vertical blanking, this register value is used instead of HSC to define the end of the horizontal sync.

VBSH

The VBSH register contains the high order bits of the line number to start vertical blanking on. The vertical counter is a 12 bit counter which requires two registers to program. The four least significant bits of VBSH are used with VBSL to form the 12 bit line count. The four high order bits of VBSH are don't cares, and are read as zero.

VBSL

The VBSL register contains the low order bits of the line number to start vertical blanking on. The vertical counter is a 12 bit counter which requires two registers to program. The four least significant bits of VBSH are used with VBSL to form the 12 bit line count. The VBS registers are programmed in multiples of lines.

VBC

The VBC register contains the vertical blank end value. It is programmed in multiples of lines. When the vertical counter reaches this value, the composite blank (DAC_BLK_) goes active. The value for VBC must be programmed to be less than VBSH + VBSL.

VSS

The VSS register contains the vertical sync start value. It is programmed in multiples of lines. When the vertical counter reaches VSS, the vertical sync output (VS_) goes active. VSS must be programmed to be less than VBSH + VBSL, and should be less than VBC.

VSC

The VSC register contains the vertical sync end value. it is programmed in multiples of lines. When the vertical counter reaches VSC, the vertical sync output (VS_) goes inactive. VSC must be programmed to be less VBSH + VBSL. It must also be programmed to be greater than VSS and should be less than VBC. A basic vertical sweep with respect to the vertical counter should look something like:

0....VSS....VSC....VBC.....VBSH + VBSL

XCS

The XCS register contains the transfer hold off start value. It is programmed in multiples of 8 pixels. The S4-Video chip generates transfer cycles as necessary by counting shift clocks. The shift clock is inactive during horizontal blanking however. If an access to the frame buffer or any of the internal registers were attempted during a horizontal blank which occurred during a transfer cycle, The S4-VIDEO chip would not be able to respond until after the blanking and transfer were completed which in computer time could be a very long time degrading performance. The XCS and XCC registers allow for a window to be programmed around relative to the horizontal blank window (defined by HBS and HBC) which will prevent a transfer cycle from starting until late in the horizontal blank period, thus allowing other accesses to the video chip in the mean time.

The values for XCS anc XCC will be less than HBS and HBC respectively and will depend greatly on the relationship between the system clock and the video clock. The most important timing

relationship however is the relationship between XCC and HBC. XCC must occur sufficiently before HBC to allow a transfer cycle to complete before the horizontal counter reaches HBC. A transfer cycle requires at minimum 7 system clocks to complete

XCC

The XCC register contains the the transfer cycle hold off end value. It is programmed in multiples of 8 pixels. See XCS for explanation.

Video Controller Function

Horizontal and Vertical State Machines

The horizontal statemachine is normally programmed in units of 8 pixels. Since all quantities are 8-bit registers, this limits the maximum horziontal period to 8*256 or 2048 pixels.

The vertical state machine is programmed in units of lines. All units are 8-bit except for the vertical blank start which is a 12-bit register accessed through its low and high byte.

In general, the numbers programmed in these registers is the actual number desired minus 1.

Shift and Load Clock

The video RAM shift clock and the load-dac clock are generated from the SCLK input. The load DAC clock is continously active and is synchronous to SCLK. The Video RAM shift clock is normally the same as load DAC clock but is held high during horizontal blanking. The shift clock will not be held high during vertical blanking or video disable in order to meet video RAM specifications.

Reload Counter

The reload counter counts the number of shift clocks and asserts transfer request XREQ to the Memory Controller. The total count is 256 for 256K VRAM and 512 for 1MBit VRAM. However, XREQ must be asserted sufficiently early to allow a CPU cycle and a Video cycle to be completed before the total count is achieved. This time is fixed in the chip to be 8 shift clocks for 64K*4 VRAM and 16 shift clocks for 256K*4 VRAM before the total count value. The reload counter is reset with condition XCLR (see below).

Transfer Counter

The transfer counter maintains the address of the next row to be transferred into the VRAM shift register. The transfer counter is incremented at the end of each video reload cycle to point at the next row to be transferred. The transfer counter is reset with condition XCLR (see below). To prevent the S4-Video chip from locking up the IU if a transfer request occurs during a horizontal blanking period, two registers (XCS and XCC) are used to hold off transfer requests. The XCS and XCC register values are prgrammed to match the HBS and HBC registers except that they are offset in time to occur earlier in time than the HBS and HBC counts. If a transfer request

occurs before XCS (and HBS) then the request is processed. If the request occurs after XCS but before HBS, then the S4–video chip suspends the transfer cycle until XCC. This will allow the IU to continue with other accesses during the blanking period.

Memory Controller Interface

The video controller interfaces to the memory controller via two signals: XREQ and XCLR.

XREQ (Transfer Cycle Request) forces a video RAM reload cycle using the address of the transfer counter. Asserting XREQ causes the memory controller to begin a video reload cycle as soon as current cycles are complete. The memory controller will wait until XREQ drops, asynchronously deassert DT/OE and then complete the video reload cycle. This allows on–the–fly video reload cycles.

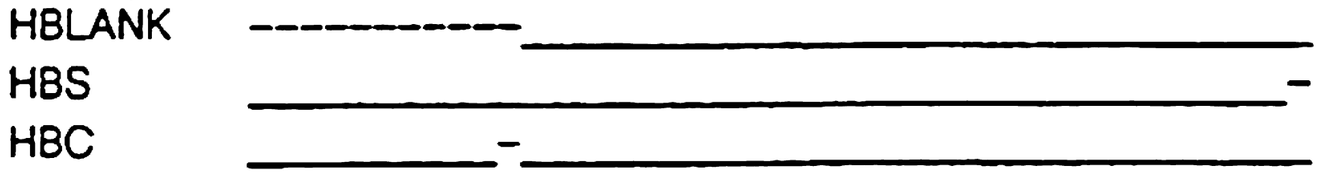
XCLR (Transfer Clear) clears the reload counter and the transfer counter and forces a minimum length, reload cycle. XCLR is asserted in the state following (VBC & HBS).

When the S4–video chip is in the master mode (control register bit 5 = 1) the XREQ and XCLR signal pins are driven as outputs mirroring the internally generated transfer requset and transfer clear signals. These two signal pins can then be connected to a parallel S4–Video chip which is configured in the slave mode (control register bit 5 = 0) to synchronize the two chips. When in the slave mode, the XREQ and XCLR signal pins are treated as inputs to the internal state–machines for synchronization purposes.

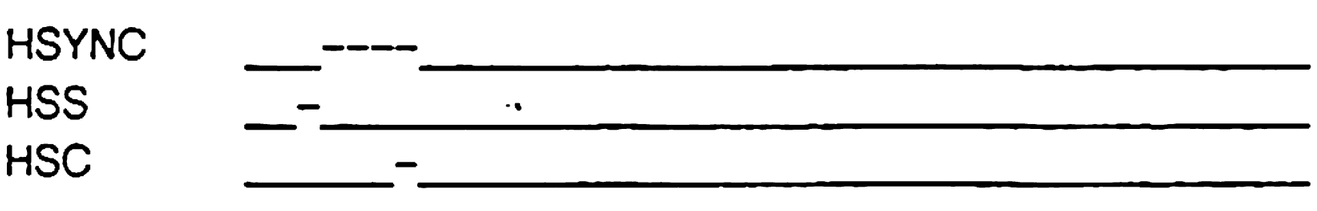
Video Controller Timing Diagrams

The following timing diagrams illustrate the function of the video controller. The composite sync output is identical to Horizontal Sync when no vertical sync is active and identical to the diagram shown if vertical sync is active.

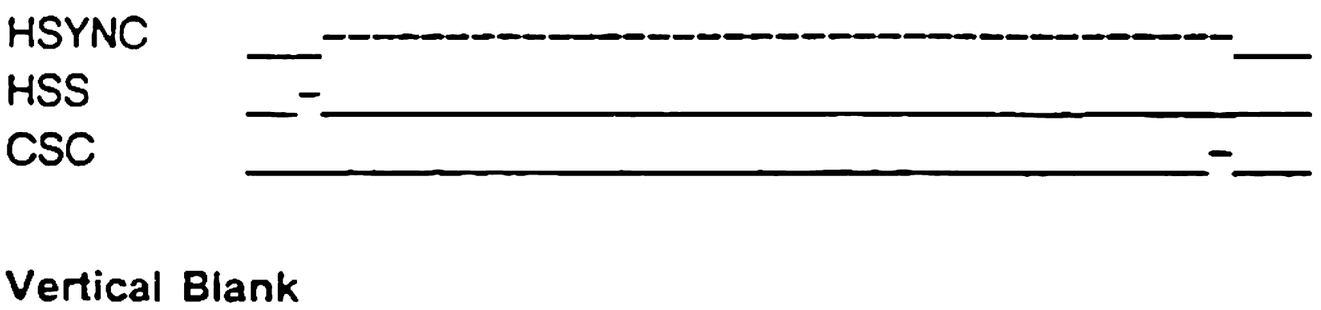
Horizontal Blank



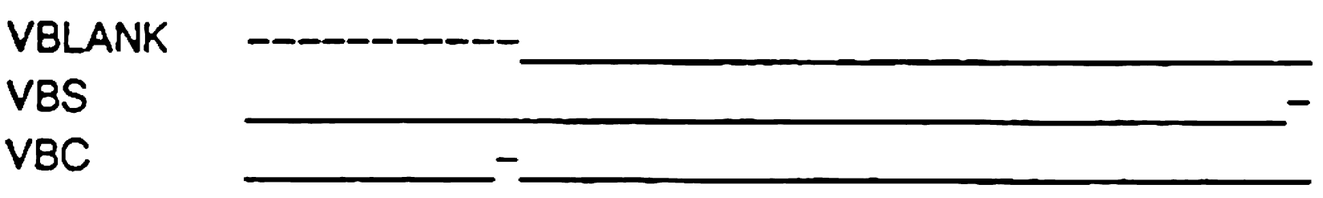
Horizontal Sync



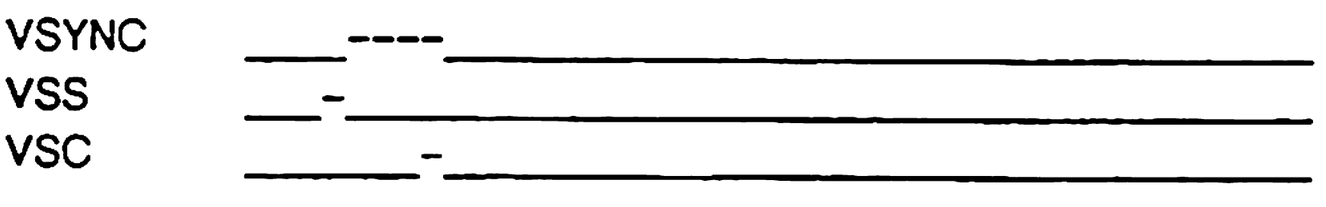
Composite Sync (during Vsync)



Vertical Blank



Vertical Sync



Video RAM Interface

The S4–Video controller generates its video memory outputs as follows:

```
VRAS(x)    = RAS * (VIDEO + CPU * (BANK(1:0) == x))

VCAS(x)    = CAS * (VIDEO + CPU * (SIZ==1) * (BYTE == x)
           + (SIZ==2) * ((BYTE ==x)+(BYTE==x+1)
           + (SIZ==3) * ((BYTE==x)+(BYTE==x+1)+(BYTE==x+2)
           +(SIZ==4)* ((BYTE==x)+(BYTE==x+1)+(BYTE==x+2)+(BYTE==x+3)))

VMA(8:0)   = MUX * (VIDEO * X(8:0) + CPU * ROW(8:0)
           + IMUX * (VIDEO * 0 + CPU * COL(8:0)
```

Cycle	CPU		Video/Refresh	
	row	col	row	col
vma0	row0	col0	x0	0
vma1	row1	col1	x1	0
vma2	row2	col2	x2	0
vma3	row3	col3	x3	0
vma4	row4	col4	x4	0
vma5	row5	col5	x5	0
vma6	row6	col6	x6	0
vma7	row7	col7	x7	0
vma8	row8	col8	x8	0

Size sb_siz(1:0)	Address sb_pa(1:0)	Byte(0) CAS(0)	Byte(1) CAS(1)	Byte(2) CAS(2)	Byte(3) CAS(3)
0,0	0,0	X	X	X	X
	0,1		X	X	X
	1,0			X	X
	1,1				X
0,1	0,0	X			
	0,1		X		
	1,0			X	
	1,1				X
1,0	0,0	X	X		
	0,1		X	X	
	1,0			X	X
	1,1				X
1,1	0,0	X	X	X	
	0,1		X	X	X
	1,0			X	X
	1,1				X

Frame Buffer Configurations

The S4-Video chip supports a number of frame buffer configurations, some of which are illustrated in the table below. The basic rules are as follows:

1. *Databus Width:* For 8-bit databus, wire VACK to SB_ACK8 and SIZ(1:0) to GND.
   For 32-bit databus, wire VACK to SB_ACK32 and SIZ(1:0) to SB_SIZ(1:0).

2. *Pixel/Load:* Wire the SCLK input to the corresponding XCLK output.

3. *Byte/Bank/Col/Row:* Wire to corresponding Sbus address bits. Ground unused inputs.

VRAM	256K*4	256K*4	256K*4	256K*4	128K*8	128K*8	128K*8	64K*4	[Size]
VRAMs	2	8	16	24/32	1	4	8	8	[#]
DataBus	8	32	32	24/32	8	32	32	32	[bit]
Banks	1/4	1	2	4	1/4	1	2	1	[#]
Bit/Pixel	1	8	8	24/32	1	8	8	1	[bit]
Pixel/Ld	8	4	8	4	8	4	8	32	[#]
Siz(2:0)	0	size	size	size	0	size	size	size	
Byte0	–	a0	a0	a0	–	a0	a0	a0	
Byte1	–	a1	a1	a1	–	a1	a1	a1	
Bank0	–	–	a2	a2	–	–	a2	–	
Bank1	–	–	–	a3	–	–	–	–	
col0	a0	a2	a3	a4	a0	a2	a3	a2	
col1	a1	a3	a4	a5	a1	a3	a4	a3	
col2	a2	a4	a5	a6	a2	a4	a5	a4	
col3	a3	a5	a6	a7	a3	a5	a6	a5	
col4	a4	a6	a7	a8	a4	a6	a7	a6	
col5	a5	a7	a8	a9	a5	a7	a8	a7	
col6	a6	a8	a9	a10	a6	a8	a9	a8	
col7	a7	a9	a10	a11	a7	a9	a10	a9	
col8	a8	a10	a11	a12	–	–	–	–	
row0	a9	a11	a12	a13	a8	a10	a11	a10	
row1	a10	a12	a13	a14	a9	a11	a12	a11	
row2	a11	a13	a14	a15	a10	a12	a13	a12	
row3	a12	a14	a15	a16	a11	a13	a14	a13	
row4	a13	a15	a16	a17	a12	a14	a15	a14	
row5	a14	a16	a17	a18	a13	a15	a16	a15	
row6	a15	a17	a18	a19	a14	a16	a17	a16	
row7	a16	a18	a19	a20	a15	a17	a18	a17	
row8	a17	a19	a20	a21	a16	a18	a19	–	

Timing Diagrams

The S4-Video controller supports four basic types of cycles: Refresh cycle, Video Cycle, CPU cycle, and Burst Cycle. For both the refresh cycle and the CPU cycle, the S4 RAM controller supports two VRAM speeds via the speed input: *fast* and *slow*. The *fast* mode supports a minimum cycle of 4 states for CPU cycles and 5 states for Refresh. In *slow* mode, RAS is extended by one additional state. This allows to use slower RAMs at the cost of an increased cycle time. There is no separate *fast* and *slow* mode for burst cycles.

Cycle Overview. The S4-RAM controller stays in the idle state (S0) until either activated by a refresh request, causing a refresh cycle, a video request, causing a video cycle, or by a CPU select, causing a CPU cycle. In case of simultaneous video, refresh, and CPU request the video request is the highest priority and the refresh request second.

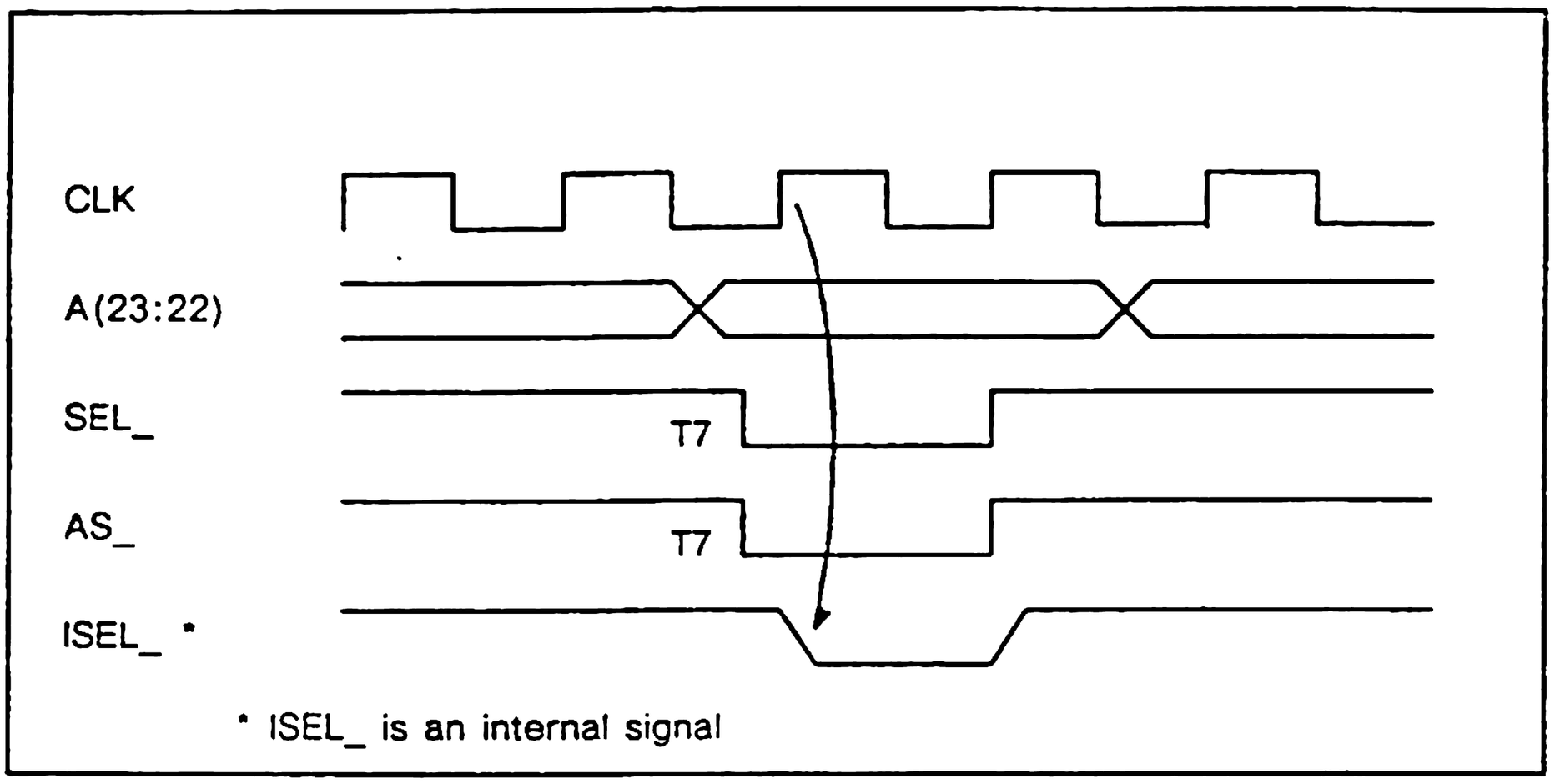
CPU cycles are initiated when a select RAM signal is received in conjunction with a matching set of addresses (see address decoding table). In response to the CPU request, the RAM controller activates RAS for the bank of memory decoded by the addresses.

Refresh cycles are generated internally by a refresh request which occurs every 320 system clocks. For a 20 MHz system clock, this is one refresh cycle every 16 usec.

Video cycles are initiated by a transition on input XREQ, which is generated by the video controller whenever a video transfer is necessary.

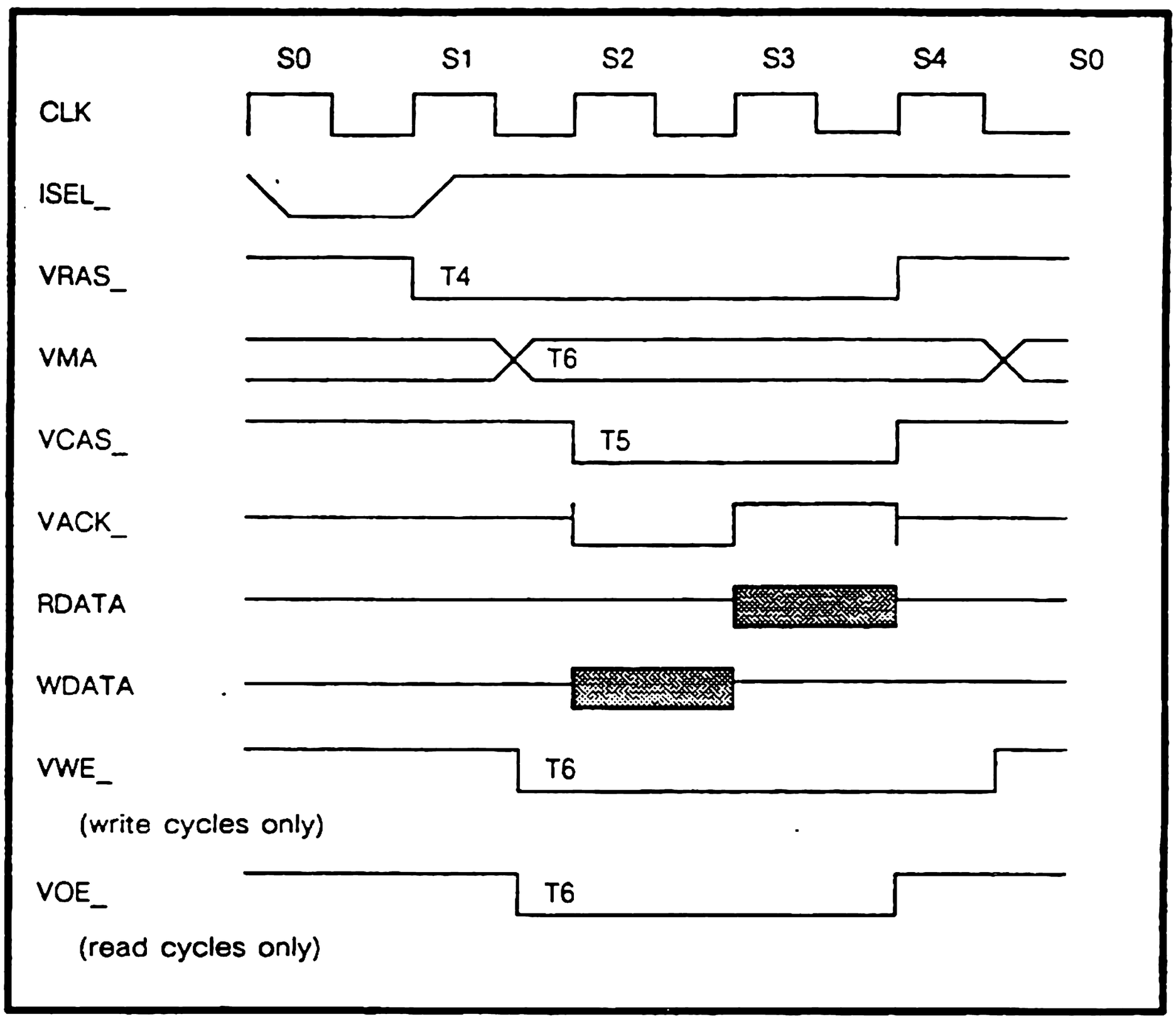
Selecting the S-4 Video Chip

The S-4 Video chip is accessed by presenting an appropriate address to address pins, by bringing CS_ active, and finally by bringing AS_ active. When addressing the frame buffer, A(23) must be set high. To access the conrol and status registers, A(23) is set low. Once an access cycle has started, it cannot be aborted. the S-4 Video chip will run through the complete cycle once it determines it has been selected.



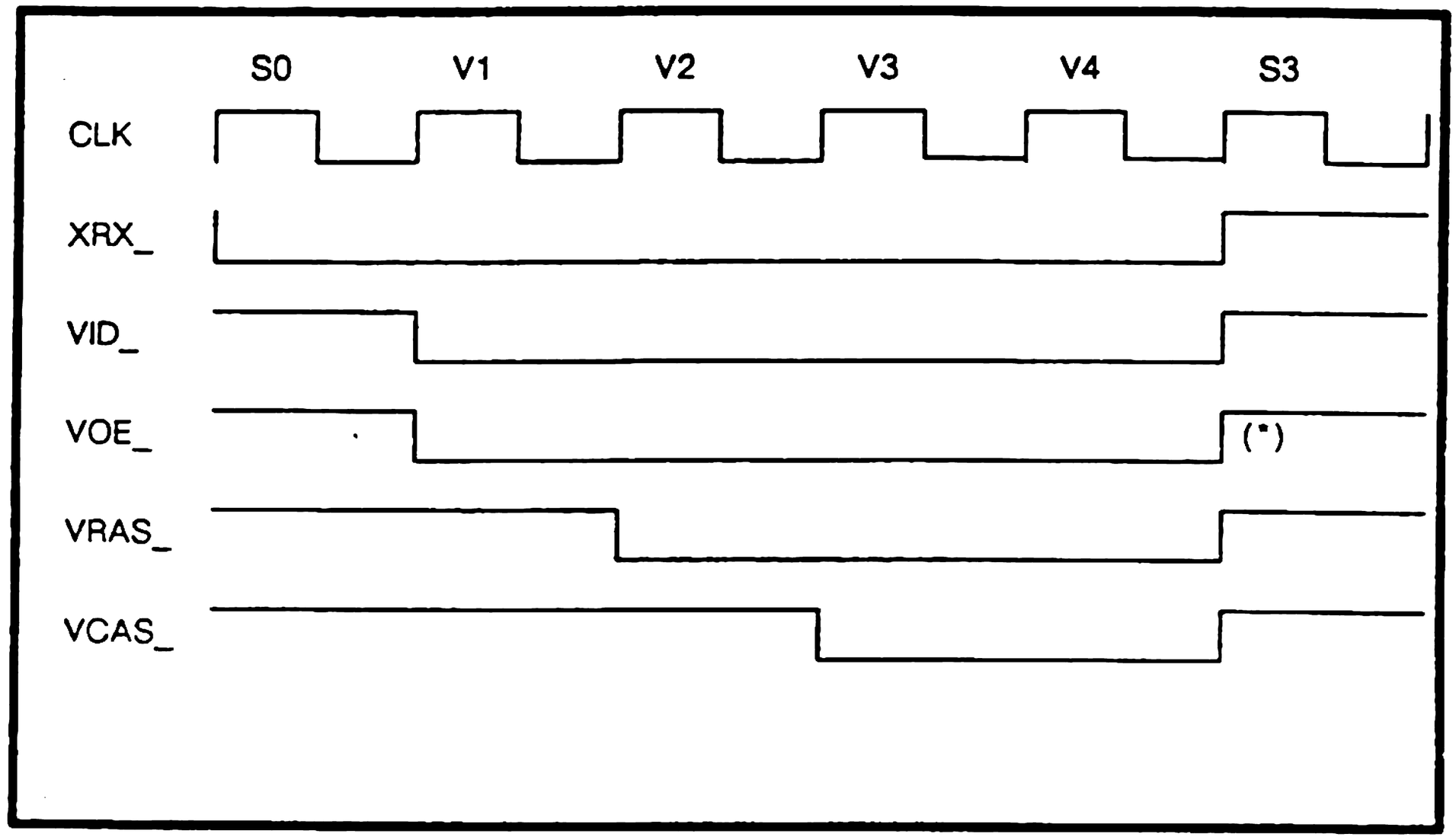
CPU Cycle

In response to a select, the RAM controller enters state S1 and asserts RAS for the bank of memory decoded by the addresses. The row/column addresses are multiplexed on the half-state following RAS. In state S2, the RAM controller asserts CAS and acknowledge signal VACK. Following S2, in fast mode the RAM controller finishes up with S10 which deasserts RAS and VACK while keeping CAS asserted. In slow mode, the RAM controller extends RAS in state S9, and then deasserts all control signals in state S10. In both cases, write data (WDATA) must be valid at beginning of CAS and read data (RDATA) is valid at the end of CAS.



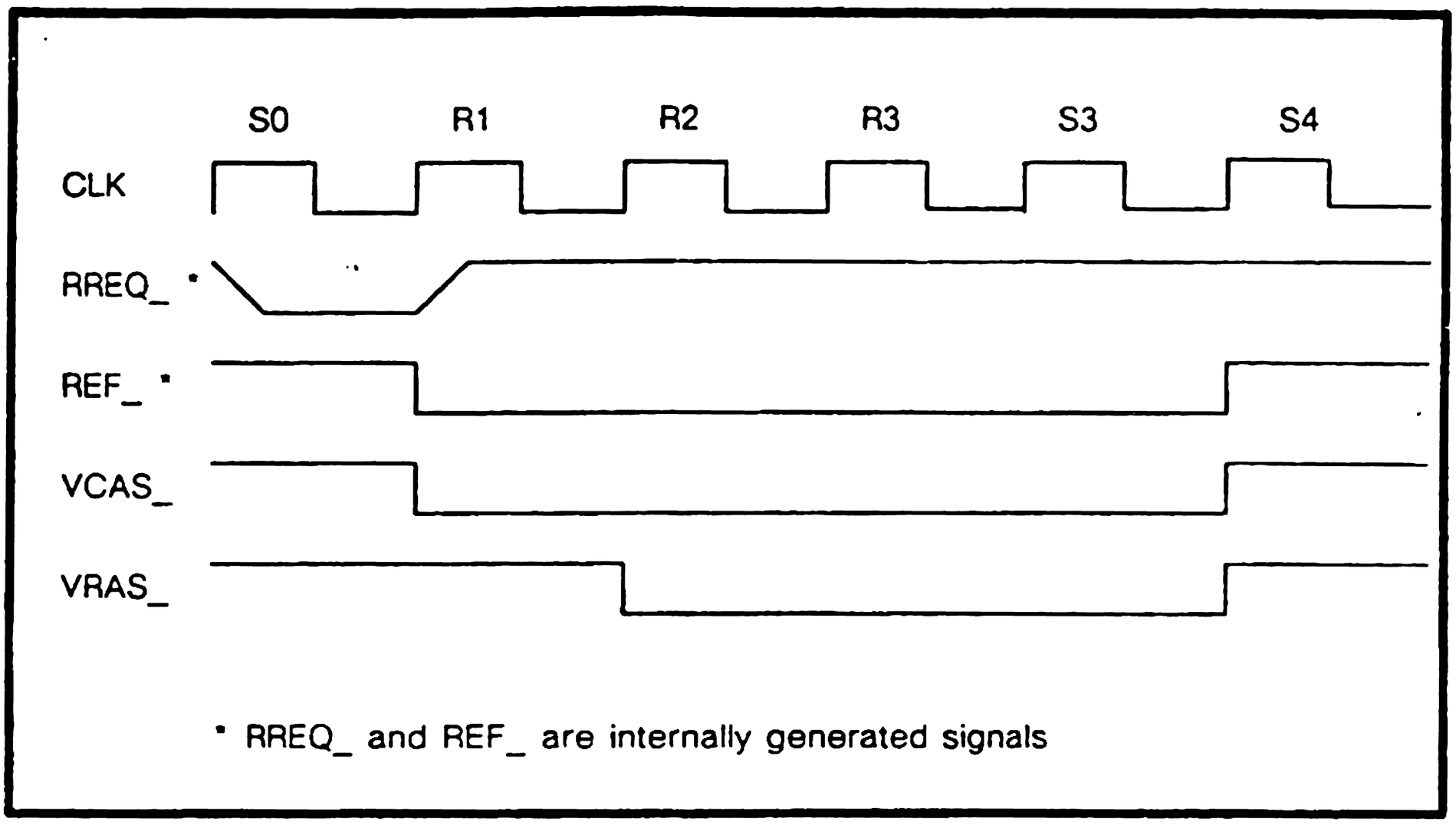
Video Transfer Cycle

The Video Cycle loads the video shift register from a row of the video RAM into the Video RAM shift register. It starts with signal XRX (transfer request synchronized) and it terminates with XRX inactive. Signal OE is asynchronously cleared with (asynchronous) transfer request inactive. Following state V3 the statemachine continues with states S12 and S13 before returning to S0.



Refresh Cycle

Refresh is implemented with a "CAS-before-RAS" cycle. Once a refresh request is recognized, all CAS outputs are asserted during state R0 followed by all RAS outputs asserted at state R1. REF, RAS, and CAS stay asserted during R2 and are deasserted in state S10. In "slow" mode, a state S9 is inserted that extends all control signals for one extra state. Refresh request takes priority over CPU cycles that arrive at the same time. Pending CPU cycles have to wait until they are recognized in S0.



Register Cycle

The Register Cycle provides access to the onboard registers or the DAC registers. Registers are selected by the appropriate addressing. If the DAC is accessed, then DAC_CS_ will be asserted during state X1, X2 and X3. On a read cycle, data from the DAC will be latched at the end of X3 and output on the system data bus during S13. SB_ACK8_ is asserted in state X3.

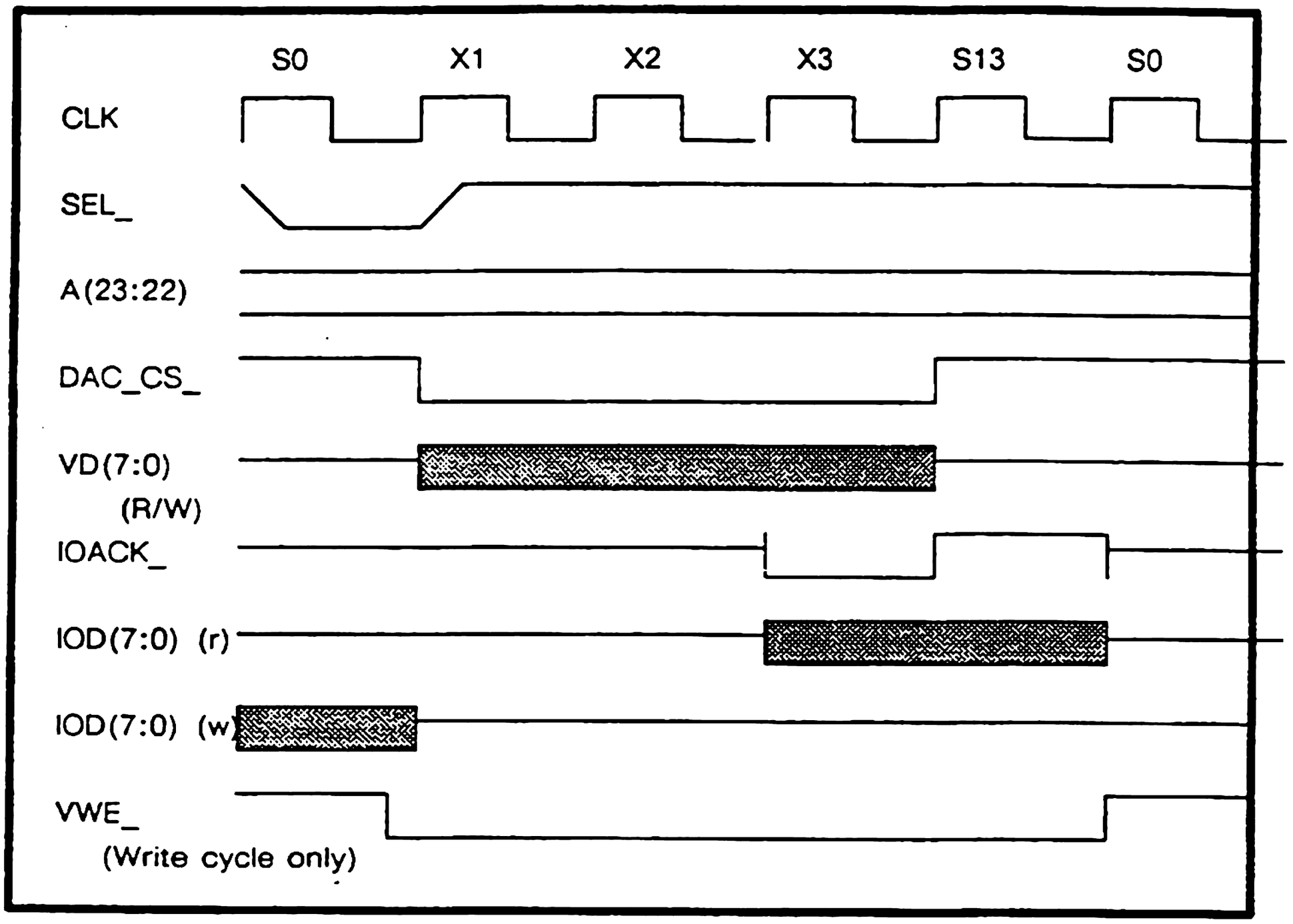


Table of Monitor Timings

Type	5 Sun 1600.1280	4 Sun 1280.1024	3 Sun 1152.900	2 Sun 1024.768	1 Apple 1152.870	0 Apple 640.480	Unit
HRes	1600	1280	1152	1024	1152	640	Pixel
VRes	1280	1024	900	768	870	480	Pixel
PClock	200.00	135.00	92.9405	70.400	100.00	30.000	MHz
HClock	89.00		61.80	53.66	68.700	35.000	kHz
HTime			16.182	18.64	14.56	28.5714	usec
VClock	67.00		65.96	66	75	66.666	Hz
VTime			15.163	15.15	13.32	16.000	msec

Register	Value	Value	Value	Value	Value	Unit	Conversion
HBS			1504	1312	1456	(856)	(X/8)-1
HBC			352	288	304	(216)	(X/8)-1
HSS			16	1312	32	(32)	(X/8)-1
HSC			144	160	160	(48)	(X/8)-1
VBS			937	813	915	525	(X)-1
VBC			37	45	45	45	(X)-1
VSS			2	(2)	3	(2)	(X)-1
VSC			6	6	6	(8)	(X)-1

Note1: HSC0 = HSC, HSC1 = HBS - HSC
Note2: VBSH = (X DIV 256), VBSL = (X MOD 256)-1
Note3: Values in parenthesis are estimates at this time.

Timing Specifications

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C

Cout	capacitive load all outputs except those below	15 pF
Cma	capacitive load MA,WE,RAS,SC	100 pF(16*6pF)
Cctl	capacitive load CAS	24pF(4*6pF)

Symbol	From	To	min	max	unit
t1	clk high	clk cycle + clk high	40	---	ns
t2	clk high	clk low	13	---	ns
t3	clk low	clk high	13	---	ns
t4	clk high	ras valid	0	15	ns
t5	clk low	cas valid	5	15	ns
t6	clk high	address, we valid	5	25	ns
t7	as valid	clk high	10	---	ns
t8	proterr valid	clk high	10	---	ns
t9	clk high	as invalid	10	---	ns
t10	xclk high	xclk cycle + xclk high	10	---	ns
t11	xclk high	xclk low	3	---	ns
t12	xclk low	xclk high	3	---	ns
t13	xclkp valid	xclkn valid	0	3	ns
t14	xi(*) high	xo(*) low	0	5	ns
t15	clk4 high	dac_ld high	0	20	ns
t16	clk4 high	sc high	2	30	ns
t17	clk4 high	blank,sync valid	2	30	ns
t18	ld high	sc high	2	10	ns
t19	ld high	blank valid	2	20	ns
t20	ld high	sync valid	2	20	ns
t21	cs active	databus valid	0	30	ns
t22	cs inactive	databus tristate	0	30	ns
t23	cs active	dac_cs acvtive	0	20	ns
t24	cs inactive	dac_cs inacvtive	0	20	ns
t25	rd,a valid	cs active	0	---	ns
t26	cs inactive	rd,a invalid	0	---	ns

Rivision History

Date	Change	By
7/18/88	First Release.	AVB
7/20/88	Defined XCS and XCC registers for use in delaying transfer cycles.	MWI
7/22/88	Added cursor registers, corrected timing diagram labels, corrected pin counts, removed diagnostic register, fixed address mappings for 64K x 4 VRAMs.	MWI
10/4/88	Added XREQ and XCLR, removed xi(3) and FAST, removed all timing diagrams related to FAST mode, added words about synchrnous operation of two chips, added words about cursor start and end address registers.	MWI
10/6/88	Removed confusing wording about cursor address registers in in description of control register	MWI

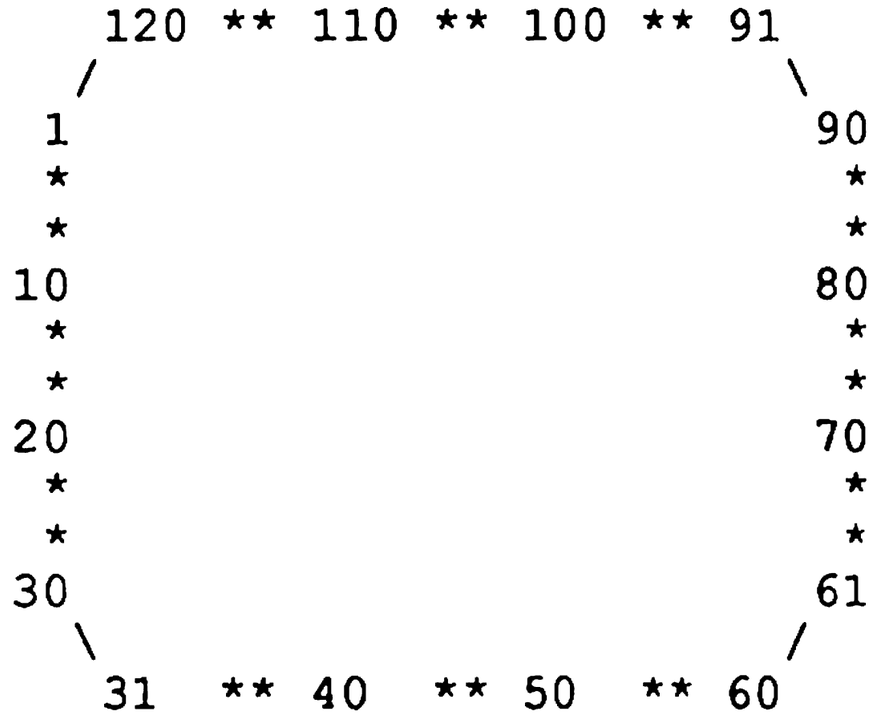
Oct 13 16:30 1989

1. P/N: 100-1804-04 2. REV: 01

3. DESCRIPTION: IC,GA,S4-VIDEO

4. PHYSICAL MODEL: (120 Pin PFP)

A. Pinout: *Top View



1 - /POR	31 -	61 - /VRAS.1	91 -
2 - VSS	32 - XCLKN	62 - /VRAS.2	92 - BYTE.0
3 - VDD	33 - XCLKP	63 - /VRAS.3	93 - BYTE.1
4 - TYPE	34 - /DAC /CS	64 - /VWE	94 - BANK.0
5 - MODE.3	35 - VD.7	65 - /VOE	95 - BANK.1
6 - MODE.2	36 - VD.6	66 - VSC	96 - COL.0
7 - MODE.1	37 - VD.5	67 - VSS	97 - COL.1
8 - MODE.0	38 - VD.4	68 - CLK	98 - COL.2
9 - /PARA	39 - VD.3	69 - /AS	99 - COL.3
10 - /X /CS	40 - VD.2	70 - /IRQ	100 - COL.4
11 - XREQ	41 - VD.1	71 - IOD.0	101 - COL.5
12 - XI.0	42 - VD.0	72 - IOD.1	102 - COL.6
13 - XI.1	43 - /DAC /BLK	73 - IOD.2	103 - COL.7
14 - XI.2	44 - /DAC /LD	74 - IOD.3	104 - COL.8
15 - VDD	45 - VSS	75 - VSS	105 - VDD
16 - VSS	46 - VDD	76 - VSS	106 - VSS
17 - VSS	47 - /VCAS.0	77 - VDD	107 - ROW.0
18 - SEL.2	48 - /VCAS.1	78 - IOD.4	108 - /B /ERR
19 - SEL.1	49 - /VCAS.2	79 - IOD.5	109 - ROW.1
20 - SEL.0	50 - /VCAS.3	80 - IOD.6	110 - ROW.2
21 - XCLR	51 - VMA.0	81 - IOD.7	111 - ROW.3
22 - VCLK	52 - VMA.1	82 - SIZ.1	112 - ROW.4
23 - XCLK8	53 - VMA.2	83 - SIZ.0	113 - ROW.5
24 - SCLK	54 - VMA.3	84 - RD	114 - ROW.6
25 - XCLK16	55 - VMA.7	85 - IOA.0	115 - /IOACK
26 - XCLK4	56 - VMA.4	86 - IOA.1	116 - ROW.7
27 - /VS	57 - VMA.5	87 - IOA.2	117 - ROW.8
28 - /HS	58 - VMA.6	88 - IOA.3	118 - A.22
29 - XVIDEO	59 - VMA.8	89 - IOA.4	119 - A.23

Oct 13 16:30 1989

30 - VDD

60 - /VRAS.0

90 - /CS

120 - /ACK

B. PIN DESCRIPTION:

SYMBOL	TYPE	DESCRIPTION
SBUS INTERFACE	**47**	
CLK	TLCHT	System Clock
SIZ(1:0)	TLCHTU	Size of writes. 0:4 bytes, 1:1 byte, 2:2 bytes, 3:3 bytes.
BYTE(1:0)	TLCHT	Byte Address
BANK(1:0)	TLCHT	Bank Address
COL(8:0)	TLCHT	Column Address
ROW(8:0)	TLCHT	Row Address
A(23:22)	TLCHT	Function Select
IOA(4:0)	TLCHTU	5-bit I/O Address
IOD(7:0)	BD4TU	8-bit I/O Data Bus
/AS	TLCHN	Address Strobe
/CS	TLCHN	Chip Select
RD	TLCHTU	Read Cycle
/ACK	BT4	Video RAM Acknowledge
/IOACK	BT4	Input/Output Acknowledge
/IRQ	BT4OD	Interrup Request (active low open drain)
/B /ERR	BT4	Buss Error
CLOCKS	**10**	
XI(2:0)	TLCHT	Pixel Clock In (3:0)
XCLKP	BT4	Pixel Clock Out Positive (to DAC)
XCLKN	BT4	Pixel Clock Out Negative (to DAC)
XCLK4	BT4	Pixel Clock Out X4
XCLK8	BT4	Pixel Clock Out X8
XCLK16	BT4	Pixel Clock Out X16
XVIDEO	BT4	Pixel Data Out
VIDEO CONTROLLER/DAC	**19**	
SCLK	TLCHT	Video RAM Shift Clock Input (X4,X8, or X16)
VCLK	TLCHT	Video Controller Clock Input (X8)
/HS	BT4	Horizontal Sync Output
/VS	BT4	Vertical Sync Output
/X /CS	BD1TU	External Chip Select
SNS(2:0)	TLCHTU	Monitor Sense Inputs (2:0)
VD(7:0)	BD1TU	Video/DAC Data Bus
/DAC /CS	BD1TU	DAC Chip Select Output
/DAC /ID	BT1	DAC Load Output
/DAC /BLK	BT1	DAC Blank Output
XREQ	BD1TD	External Transfer Request
XCLR	BD1TD	External Transfer Counter Clear

VRAM INTERFACE	20	
VMA(8:0)	BT4	Video Multiplexed Address
VCAS /(3:0)	BT4	Video CAS Enable (Byte select)
VRAS /(3:0)	BT4	Video RAS Enable (Bank select)
/VOE	BT4	Video Output Enable
/VWE	BT4	Video Write Enable
VSC	BT4	Video Shift Clock
MISC PINS	8	
MODE(3:0)	TLCHTD	Memory mode and configuration
TYPE	TLCHTD	VRAM type: 0:256K, 1:1Mbit
/POR	TLCHNU	Power On Reset. Clears control register
/PARA	BD1TU	Parametric Test Output and Output Disable

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0oC to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15

	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
tTCT	TCLK Period	40		ns
tTCH	TCLK High Time	17		ns
tTCL	TCLK Low Time	17		ns
	TCLK to Output Valid		20	ns
	TCLK to Output Invalid	3		ns
	Input Setup to Clock	15		ns
	Input Hold Time to Clock	0		ns

t23	From CS active to /DAC CS Active	0	20.0	ns	
t24	From CS Inactive to /DAC CS Inactive	0	20.0	ns	
t25	From RD,A Valid to CS Active	0		ns	
t26	From CS Inactive RD,A Invalid	0		ns	

* COUT Capacitive load all outputs except those below 15 pF
CMA Capacitive load MA,WE,RAS,SC 100 pF(16*6pF)
CCTL Capacitive load CAS 24 pF(4*6pF)

8. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CI/O	Input/Output Capacitance per Slot		20	pF
CLS	Maximum Capacitance Load per System		100	pF

9. TRUTH TABLE: N/A

10. FUNCTIONAL TABLE: N/A

11. OUTLINE DRAWING: N/A

12. WAVEFORMS: N/A

DATE: 1/04/89

AUTHOR: PL

7. AC OPERATING REQUIREMENTS*
(VCC = 4.75 to 5.25V, TA = 0 to +70oC)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t1	From CLK High to CLK Cycle + CLK High	40		ns ns
t2	From CLK High to CLK Low	13.0		ns
t3	From CLK Low to CLK High	13.0		ns
t4	From CLK High to RAS Valid	0	15.0	ns
t5	From CLK Low to CAS Valid	5.0	15.0	ns
t6	From CLK High to Address, WE Valid	5.0	25.0	ns ns
t7	From AS Valid to CLK High	10.0		ns
t8	From Proterr Valid to CLK High	10.0		ns
t9	From CLK High to AS Invalid	10.0		ns
t10	From XCLK High to XCLK Cycle + XCLK High	10.0		ns
t11	From XCLK High to XCLK Low	3.0		ns
t12	From XCLK Low to XCLK High	3.0		ns
t13	From XCLKP Valid to XCLKN Valid	0	3.0	ns
t14	From XI(*) High to XO(*) Low	0	5.0	ns
t15	From CLK4 High to /DAC ID High	0	20.0	ns
t16	From CLK4 High to SC High	2.0	30.0	ns
t17	From CLK4 High to BLANK, SYNC Valid	2.0	30.0	ns
t18	From ID High to SC High	2.0	10.0	ns
t19	From ID High to BLANK Valid	2.0	20.0	ns
t20	From ID High to SYNC Valid	2.0	20.0	ns
t21	From CS Active DATABUS Valid	0	30.0	ns
t22	From CS Inactive to DATABUS Tristate	0	30.0	ns

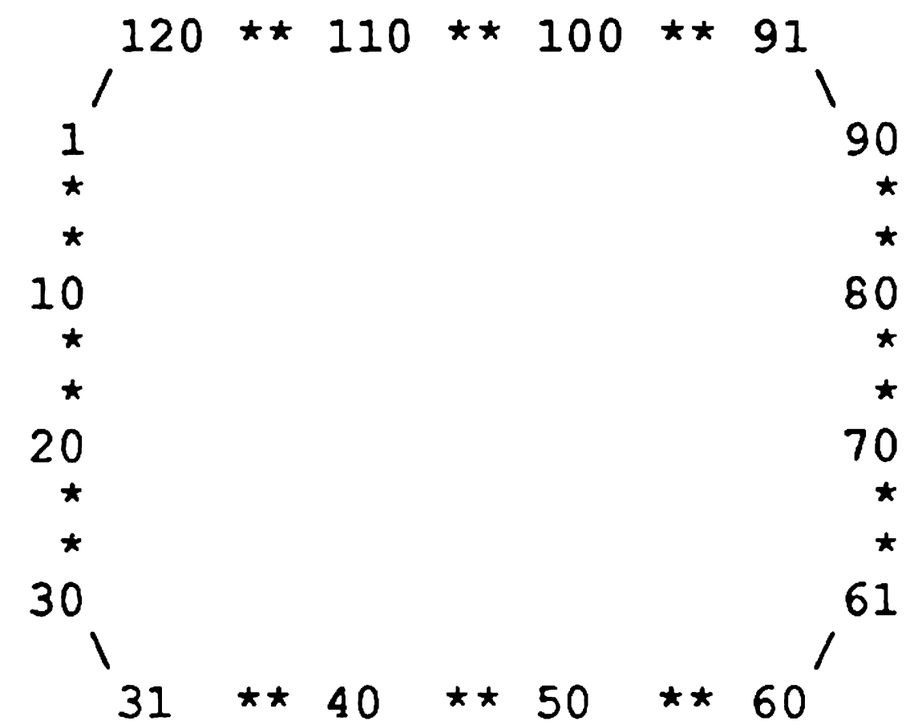
Oct 13 16:30 1989

1. P/N: 100-1803-02 2. REV: 50

3. DESCRIPTION: IC,GA,S4-DMA

4. PHYSICAL MODEL: (120 Pin PFP)

A. Pinout: *Top View



1 - /SLOW	31 - /D DBUS.7	61 - VDD	91 - VSS
2 - TEST	32 - /D DBUS.6	62 - /E AD.1	92 - /SB /IRQ
3 - /OD TST	33 - /D DBUS.5	63 - /E AD.0	93 - /SB DBUS.8
4 - /SB PA3	34 - /D DBUS.4	64 - /E AD.13	94 - /SB DBUS.9
5 - /SB PA2	35 - /D DBUS.3	65 - /E AD.14	95 - /SB DBUS.10
6 - /SB PA1	36 - /D DBUS.2	66 - /E AD.15	96 - /SB DBUS.11
7 - VDD	37 - /D DBUS.1	67 - /E AH.0	97 - /SB DBUS.12
8 - VSS	38 - /D DBUS.0	68 - /E AH.1	98 - /SB DBUS.13
9 - /SB PAX	39 - /D RESET	69 - /E AH.2	99 - /SB DBUS.14
10 - /SB PAY	40 - /D /IRQ	70 - /E AH.3	100 - /SB DBUS.15
11 - /SB /MERR	41 - /D /CS	71 - VSS	101 - /SB DBUS.16
12 - /SB /BG	42 - VSS	72 - VDD	102 - /SB DBUS.17
13 - /SB /BR	43 - VDD	73 - /E AH.4	103 -
14 - /SB /ERR	44 - /SB CLK	74 - /E AH.5	104 - VDD
15 - /SB /ACK32	45 - /D /RD	75 - /E AH.6	105 - VSS
16 - /SB SIZ2	46 - /D /WR	76 - /E AH.7	106 - /SB DBUS.18
17 - /SB SIZ1	47 - /D REQ	77 - /SB /ACK8	107 - /SB DBUS.19
18 - /SB SIZ0	48 - /D /ACK	78 - /SB /RESET	108 - /SB DBUS.20
19 - VDD	49 - /E AD.12	79 - /SB READ	109 - /SB DBUS.21
20 -	50 - /E AD.11	80 - VSS	110 - /SB DBUS.22
21 - /ID /CS	51 - /E AD.10	81 - /SB /SEL	111 - /SB DBUS.23
22 - /E BYTE	52 - VSS	82 - /SB /AS	112 - /SB DBUS.24
23 - /E /CS	53 - /E AD.9	83 - /SB /DBUS.0	113 - /SB DBUS.25
24 - /E READ	54 - /E AD.8	84 - /SB /DBUS.1	114 - VSS
25 - /E /HLDA	55 - /E AD.7	85 - /SB /DBUS.2	115 - /SB DBUS.26
26 - /E /HOLD	56 - /E AD.6	86 - /SB /DBUS.3	116 - /SB DBUS.27
27 - /E /AS	57 - /E AD.5	87 - /SB /DBUS.4	117 - /SB DBUS.28
28 - /E /DAS	58 - /E AD.4	88 - /SB /DBUS.5	118 - /SB DBUS.29
29 - /E /RDY	59 - /E AD.3	89 - /SB /DBUS.6	119 - /SB DBUS.30
30 - VSS	60 - /E AD.2	90 - /SB /DBUS.7	120 - /SB DBUS.31

B. PIN DESCRIPTION:

SYMBOL	TYPE	DESCRIPTION
/SB D(31:0)	BD4TU	Sbus Data Bus
/SB /BR	BT4	Sbus Bus Request
/SB /BG	TLCHTU	Sbus Bus Grant
/SB /ACK32	BD4TNU	Sbus 32bit Acknowledge
/SB /ACK8	BD4TNU	Sbus 8bit Acknowledge
/SB /RESET	TLCHTU	Sbus Reset
/SB /ERR	BD4TNU	Sbus Error
/SB /MERR	TLCHTU	Sbus Memory Error (INT15)
/SB CLK	DRVC16	Sbus Clock Input
/SB RD	BD4TU	Sbus Read/Write
/SB /SEL	TLCHTU	Sbus Select
/SB /IRQ	BD4TOD	Interrupt Request (open-drain)
/SB SIZ(2:0)	BD4TU	Sbus Transfer Size
/SB /AS	TLCHTU	Address Strobe (address is valid)
PA(X:Y)	TLCHTU	Physical Address lines (for slave decodes)
PA(3:1)	TLCHTU	Physical Address bits
ETHERNET INTERFACE	32	
/E /AS	TLCHTD	Ethernet Address Strobe
/E /HOLD	TLCHTU	Ethernet Hold
/E /HLDA	BT4	Ethernet Hold Acknowledge
/E /READ	BD4TU	Ethernet Read
/E /DAS	BD4TU	Ethernet Data Strobe
/E /RDY	BD4TU	Ethernet Ready
/E /CS	BT4	Ethernet Chip Select
/E BYTE	TLCHTU	Ethernet Byte marker
/E A23:16	TLCHTD	Ethernet High Order Address
/E AD15:0	BD4TU	Ethernet Address / Data Bus
DMA INTERFACE	16	
/D D7:0	BD4TD	DMA Data Bus
/D REQ	TLCHT	DMA Request
/D /ACK	BT4	DMA Acknowledge
/D /RD	BT4	DMA Read Strobe. (reg read or DMA to memory).
/D /WR	BT4	DMA Write Strobe. (reg write or DMA from memory).
/D /CS	BT4	DMA Chip Select for slave register acces
/D /IRQ	TLCHTU	DMA Interrupt Request
/D /RESET	BT4	DMA Reset
/ID /CS	BD4T	Secondary Device Select (boot prom) output Strap High to specify existence of external prom.

MISCELLANEOUS 4

FAST//SLOW	TLCHTU	Fast or Slow DMA Acknowledge cycles (For use with ESP SCSI controller fast mode should be used - internal pullup means this requires no external connection for fast mode).
/TEST	IBUFU	Test control. When Low enable parametric test output and disables all tri-state drivers. When High disables parametric test output.
/OD TST	BT4	Parametric test output

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	Typ	Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16

IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

Oct 13 16:30 1989

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC)

NO	SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
1	CLK	Clock period	30		ns
2	CLK	Clock High			ns
3	CLK	Clock Low			ns
4	Note 22	Hold WRT CLK^	0		ns
5	Note 22	Setup to CLK^	14.0		ns
6	Note 22	Setup to CLK^	23.0		
7	Note 22	Hold WRT CLK^	5.0		ns
8	Note 22	Setup to CLK^	13.5		ns
9	Note 22	Hold WRT CLK^	0		ns
10	Note 22, 26	CLK^ to Output Valid		30.4	ns
11	Note 22, 26	CLK^ to Output Invalid		22.0	ns
12	Note 22, 27	CLK^ to Output Valid		31.4	ns
13	Note 22, 27	CLK^ to Output Invalid		19.7	ns
14	Note 22, 26	CLK^ to Output Low		24.0	ns
15	Note 22, 26	CLK^ to Output High		18.5	ns
16	/D REQ	Setup to CLK^	0		ns
17	/D REQ	Hold WRT to CLK^	3.7		ns
18	/D DBUS (7:0)	Setup to CLK^	2.0		ns
19	/D DBUS (7:0)	Hold WRT to CLK^	3.5		ns
20	/D IRQ	Setup to CLK^	0		ns
21	/D IRQ	Hold WRT to CLK^	4.0		ns

22	/D /RD	CLK^ to Output Low (Note 28)		25.5	ns
23	/D /RD	CLK^ to Output High (Note 28)		20.0	ns
24	/D /WR	CLK^ to Output Low (Note 28)		22.5	ns
25	/D /WR	CLK^ to Output High (Note 28)		18.0	ns
26	/D DBUS (7:0)	CLK^ to Output Valid (Note 28)		29.0	ns
27	/D DBUS (7:0)	CLK^ to Output Invalid (Note 28)		21.0	ns
28	Note 2,7	CLK^ to Output Low		24.0	ns
29	Note 2,7	CLK^ to Output High		19.0	ns
30	Note 3	/D /RD to /D DBUS Valid		50	ns
31	Note 3	/D /RD to /D DBUS Invalid	0		ns
32	/D RESET	CLK^ to Output Low (Note 28)		20.0	ns
33	/D RESET	CLK^ to Output High (Note 28)		18.0	ns
34	/D /ACK	CLK^ to Output Low (Note 28)		28.5	ns
35	/D /ACK	CLK^ to Output High (Note 28)		17.0	ns
36	/E AD (15:0)	Setup to CLK^ (Note 28)	1.0		ns
37	/E AD (15:0)	Hold WRT to CLK^ (Note 25)	4.0		ns
38	/E AD (15:0)	CLK^ to Output Valid (Note 28)		36.0	ns
39	/E AD (15:0)	CLK^ to Output Invalid (Note 28)		25.0	ns
40	/E /HLDA	CLK^ to Output High (Note 28)		18.0	ns
41	/E /HLDA	CLK^ to Output Low (Note 28)		21.5	ns

42	/E READ	CLK^ to Output Valid (Note 28)		15.5	ns
43	/E READ	CLK^ to Output Invalid (Note 28)		12.0	ns
44	/E /DAS	CLK^ to Output Valid (Note 28)		23.0	ns
45	/E /DAS	CLK^ to Output Invalid (Note 28)		18.5	ns
46	/E /RDY	CLK^ to Output Valid (Note 28)		23.0	ns
47	/E /RDY	CLK^ to Output Invalid (Note 28)		17.5	ns
48	/E /CS	CLK^ to Output High (Note 28)		15.5	ns
49	/E /CS	CLK^ to Output Low (Note 28)		20.0	ns
50	/E /RDY	Setup to CLK^		0	ns
51	/E /RDY	Hold WRT to CLK^		2.8	ns
52	/E AD (15:0)	ADDR Setup to /E /AS Low		15.0	ns
53	/E AD (15:0)	ADDR Hold WRT /E /AS High		0	ns
54	/E /HOLD	Setup to CLK^		0	ns
55	/E /HOLD	Hold WRT to CLK^		4.0	ns
56	Note 22	Setup to CLK^	0		ns
57	Note 22	Hold WRT to CLK^	3.0		ns
58		CLK^ to Output Valid		15.5	ns
59		CLK^ to Output Invalid		12.0	ns

NOTES:

22. These values represent the timing characteristics of groups of signals. By referring to the Timing Diagrams it can be seen that one mnemonic value can represent many different signal paths.
23. These timing parameters are true for both the singals /D /CS and /ID /CS.
24. The documented values represent the timing of an external device (in this case the ESP SCSI chip), to which this gate-array is matched by design.
25. The Setup and Hold times refer to the timing diagram on which they are shown, and in particular to the clock edges shown. the /E AD bus is designed to be that of the LANCE Ethernet controller. Internal to this chip the /E AD bus is not latched for at least 2 clock cycles to alleviate any potential timing problems. Hence the Ons timing requirements shown are true only if the cycle by cycle handshaking specified by the LANCE is maintained.
26. Load = 100 pF
27. Load = 130 pF
28. Load = 80 pF

7. TRUTH TABLE: N/A

8. FUNCTIONAL TABLE: N/A

9. OUTLINE DRAWING: N/A

10. WAVEFORMS: N/A

DATE: 1/03/89

AUTHOR: PL

Features

- **Pin compatibility and software compatibility to the S4-DMA**
- **Support for concurrent 5 MByte/sec SCSI and 1.25 MByte/sec Ethernet transfers with a 20 or 25 MHz SBus Clock**
- **Support for 16-byte SBus bursts**
- **"NEXT" address/byte count feature for data block chaining**

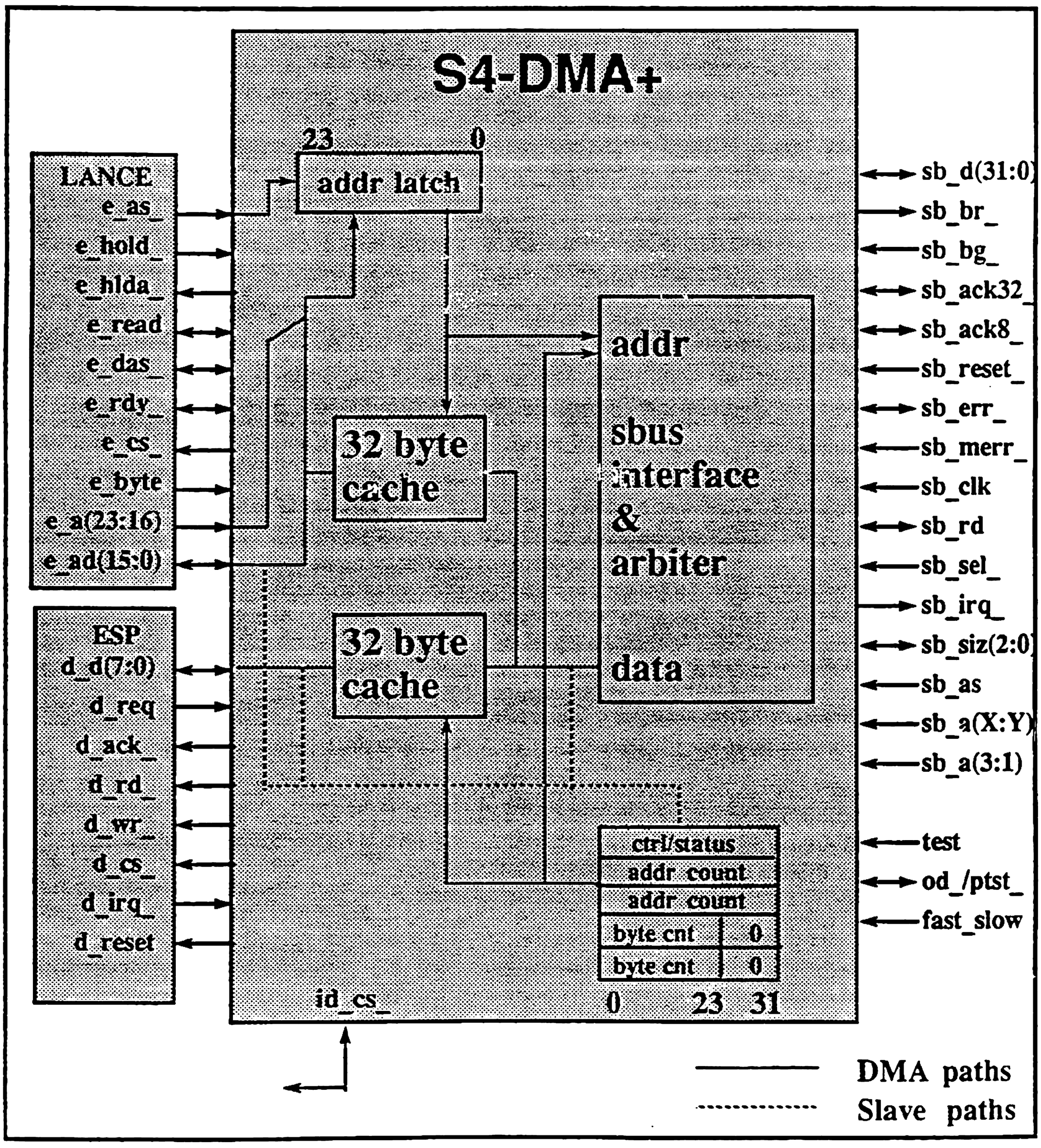


Table of Contents

1.0 Pin Description

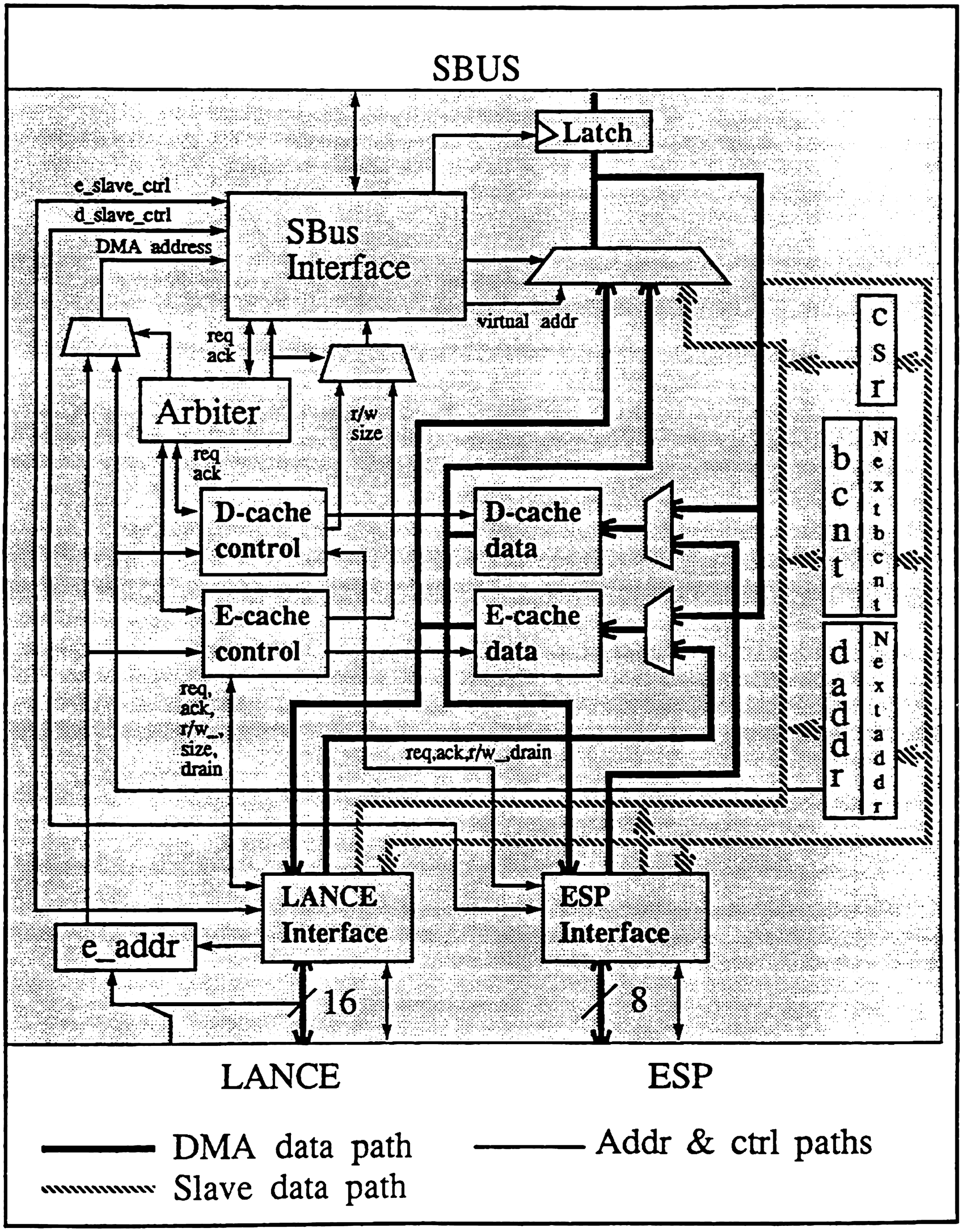
Name	Pin	Type	Description
SBus Interface		**51**	
sb_d(31:0)	120-115, 113-106, 102-93, 90-83	BD4TU	SBus Data Bus
sb_br_	13	BT4	SBus Bus Request
sb_bg_	12	TLCHTU	SBus Bus Grant
sb_ack32_	15	BD4TNU	SBus 32bit Acknowledge
sb_ack8_	77	BD4TNU	SBus 8bit Acknowledge
sb_reset_	78	TLCHTU	SBus Reset
sb_err_	14	BD4TNU	SBus Error
sb_merr_	11	TLCHTU	SBus Memory Error (INT15)
sb_clk	44	DRVC16	SBus Clock input
sb_rd	79	BD4TU	SBus Read/Write_
sb_sel_	81	TLCHTU	SBus Select
sb_irq_	92	BD4TOD	Interrupt Request (open-drain)
sb_siz(2:0)	16-18	BD4TU	SBus transfer Size
sb_as_	82	TLCHTU	Address strobe (addr is valid)
sb_a(X:Y)	9-10	TLCHTU	Physical Address lines (for slave decodes)
sb_a(3:2)	4-5	TLCHTU	Physical Address bits
LANCE Interface	**32**		
e_as_	27	TLCHTD	Ethernet Address Strobe
e_hold_	26	TLCHTU	Ethernet Hold
e_hlda_	25	BT4	Ethernet Hold Acknowledge
e_read	24	BD4TU	Ethernet Read
e_das_	28	BD4TU	Ethernet Data Strobe
e_rdy_	29	BD4TU	Ethernet Ready
e_cs_	23	BT4	Ethernet Chip Select
e_byte	22	TLCHTU	Ethernet Byte marker
e_a23:16	76-73, 70-67	TLCHTD	Ethernet High Order Address
e_ad15:0	66-64, 49-51, 53-60, 62-63	BD4TU	Ethernet Address / Data Bus
ESP Interface	**16**		
d_d7:0	31-38	BD4TD	DMA Data Bus
d_req	47	TLCHT	DMA Request
d_ack_	48	BT4	DMA Acknowledge
d_rd_	45	BT4	DMA Read Strobe. (reg read or dma data to memory).
d_wr_	46	BT4	DMA Write Strobe. (reg write or dma from memory).
d_cs_	41	BT4	DMA Chip Select for slave register access.
d_irq_	40	TLCHTU	DMA Interrupt Request
d_reset	39	BT4	DMA Reset
id_cs_	21	BD4T	Secondary Device Select (boot prom) output Pull high to specify existence of external prom

Miscellaneous 4

Name	Pin	Type	Description
fast/slow_	1	TLCHTU	fast or slow DMA acknowledge cycles (For use with ESP SCSI controller fast mode should be used - internal pullup means this requires no external connection for fast mode).
test_	2	IBUFU	Test control. When low enables parametric test output and disables all tri-state drivers. When high disables parametric test output.
od_tst	3	BT4	parametric test output

Name	Pin	Number	Description
Signals:		103	
VDD:	7, 19, 43, 61, 72, 104	6	+5Volt power connection
VSS:	8, 30, 42, 52, 71, 80, 91, 105, 114	9	Ground connection
NC	6, 20, 103	3	No connection
TOTAL:		120	
Part:			
Package:		PFP120	

1.1 Block Diagram



2.0 SBus Interface

The S4-DMA+ provides for the communication between the SBus and two I/O devices, each of which are connected to separate ports of the S4-DMA+. One I/O device port is tailored to interface with the AMD AM7990 LANCE Ethernet Controller, and the other port is tailored to interface with the NCR 53C90 ESP SCSI controller. However, it is possible to interface with other I/O devices with some external logic. In this document, the two ports will be referred to as the ESP interface and the LANCE interface, and it will be assumed that the ESP and LANCE are connected to their respective interfaces. Also, the two caches associated with these interfaces will be referred to as the D-cache and the E-cache, respectively. (For historical reasons, things associated with the ESP have names beginning with "d" and things associated with the LANCE have names beginning with "e").

The S4-DMA+ provides both Slave cycle and Master cycle accesses over the SBus. Address and Control registers in the S4-DMA+ gate array, along with similar registers in both the LANCE and the ESP (or other devices in their places), can be accessed by the CPU via Slave cycles.

2.1 Master Cycles

To perform a Master Cycle the S4-DMA+ requests the use of the SBus by asserting sb_br_, the bus request signal. On reception of sb_bg_, bus grant, the S4-DMA+ takes control of the SBus to transfer data to or from memory, using the signals sb_d(31:0), sb_read, sb_siz(2:0), sb_err, sb_ack8_ and sb_ack32_ . The address is multiplexed onto the sb_d bus for all accesses on the SBus.

All DMA reads from memory will be 4-word bursts. DMA writes to memory can be any size (i.e. byte, half-word, word or 4-words.) The S4-DMA+ SBus interface will always use the largest size possible when writing to memory. The largest possible size is determined by which bytes have been written to the S4DMA+ by the I/O device.

The SBus interface of the S4-DMA+ is capable of supporting rerun acknowledgments from slaves. The requested DMA transfer will repeat until either it completes or an error indicator (sb_err_) is received, whereupon it will be aborted. When the S4-DMA+ receives a rerun acknowledge, it will unassert sb_br_ for one clock to give the CPU and other DMA masters a chance to win SBus arbitration. Then it will re-assert sb_br_ to retry the cycle.

Once the S4-DMA+ has asserted sb_br_, it will not unassert it until it has received sb_bg_.

2.2 Slave Cycles

The CPU accesses registers in the S4-DMA+ chip and attached devices by the use of a geographical select signal on the SBus, sb_sel_ , in conjunction with sb_as_. Within the geographical select the S4-DMA+ will decode the Physical address, (sb_a(X:Y)) in the following way;

sb_a(X:Y)	Addressed device	transfer size
00	Internal ID field (or external if id_cs_ is pulled high)	word *
01	On chip registers	word *
10	ESP registers (off-chip)	byte **
11	LANCE registers (off-chip)	2 byte ***

Note: sb_a(X:Y) = sb_a(23:22) on Campus and Calvin.

* Byte and half-word sizes are also allowed on reads of these fields.

** Word and half-word accesses to this field are also allowed if the 8-bit ESP registers are at contiguous addresses (they are not at contiguous addresses on Campus or Calvin).

*** Byte accesses are also allowed on reads of this field. Word accesses to this field are also allowed, but they don't necessarily make sense for the LANCE.

During Slave Cycles the S4-DMA+ takes control of the sb_err, sb_ack8_ and sb_ack32_, signals. The combination of responses are as follows:

sb_ack8_	sb_ack32_	sb_err_	Definition
1	1	1	insert wait states **
1	1	0	error **
1	0	1	32-bit port ack **
1	0	0	alternate error
0	1	0	rerun **
0	0	1	16-bit port ack **
0	1	1	8-bit port ack **
0	0	0	reserved

This table represents all possible SBus responses. The S4-DMA+ gate-array can, however, only generate those responses marked with a **.

The S4-DMA+ will give a rerun acknowledge if the slave access is to an I/O device (ESP or LANCE) that is currently active with a DMA transfer to/from the S4-DMA+. It will not give a rerun if the slave access is to an internal register of the S4-DMA+; these accesses can take place concurrently with DMA activity between the I/O devices and the S4-DMA+. Forcing the SBus master (usually the CPU) to rerun its slave accesses in this manner will insure that a deadlock does not occur between the ESP or LANCE and the SBus master.

If the slave accesses to the S4-DMA+ does not cause a rerun acknowledge, then the S4-DMA+ will respond in the following way:

Acknowledge	Field accessed
32-bit port ack	Internal ID or internal register
16-port ack	LANCE register
8-bit port ack	ESP register or external ID
sb_err_ ack	A byte or half-word write to an On-chip register, or a byte write to the LANCE registers or a slave access of some size other than word, half-word, or byte.

3.0 SBus Identification

This is a mechanism which allows software to uniquely identify each SBus device, since each device can have a unique ID.

The onboard ID of the S4-DMA+ is hardwired to the 32-bit value fe810101, which the same as that of the S4-DMA. This value will be returned when the ID field is read by the CPU if the id_cs_ pin is tied low. If the id_cs_ pin is pulled high through a large (~4.7k) resistor, then access to the ID field will cause an external access using the id_cs_ pin as a external chip select. For offboard ID reads, ID data should be driven onto the d_dbus(7:0) when id_cs_ and d_rd_ are asserted. sb_a bits can be used to address the external ID PROM. Refer to Campus-1 Programmers Model for further details.

Note that for offboard ID reads, (external to the S4-DMA+,) the id_cs_ is simply an address decode from sb_a(X:Y); therefore, it can be active for a slave read cycle or a slave write cycle.

4.0 LANCE Interface

The S4-DMA+ will provide all necessary buffering and arbitration functions to allow the LANCE to access main memory, over the SBus, (master cycle), and the CPU to access the LANCE, (slave cycle). Between the LANCE interface and the SBus interface is a 2 line, 4-word/line cache (E-cache) with consistency control logic. Each byte in the E-cache has its own valid/dirty bit, and each line has a bit that determines the meaning of the valid/dirty bits for that line (i.e. valid or dirty). Bytes being transferred to memory are marked dirty as they are loaded into the E-cache. Bytes being transferred from memory are marked as valid as they are loaded into the E-cache. Consistency control ensures that all data written by the LANCE gets to main memory in a deterministic manner, and is handled completely in hardware.

The LANCE uses multiplexed address and data, so the S4-DMA+ demultiplexes them internally. The address supported by the LANCE is 24 bits; as per the SUN-4 Architectural Specification the upper 8 bits of the 32 bit virtual address are driven to 0xFF. The LANCE has a 16 bit data path which can accommodate 8 or 16 bit accesses, by the use of byte masking capabilities.

Data packing and unpacking is used to reduce the bus bandwidth impact of LANCE transfers on the SBus. The LANCE performs 2 distinct types of memory access; data transfer from or to the network, and descriptor access for buffer management control. The LANCE performs data transfers in bursts of up to 8 half-word accesses (16 bytes total), and message descriptor transfers as single half-word accesses. The end of a LANCE transfer is detected by the unassertion of the e_hold_ signal.

Note: Any I/O device connected to the LANCE interface of the S4-DMA+ is required to unassert e_hold_ between write cycles and read cycles, and it is not allowed to unassert e_hold_ during the e_as_ or e_das_ pulses, or between these pulses. Also note that an I/O device connected to the LANCE interface can easily drain all dirty bytes in the E-cache simply by asserting and then unasserting e_hold_.

4.1 Transfers to memory

Data from the LANCE is packed into the E-cache and marked as "dirty" until the last byte (byte 15) of an E-cache line is written to, or the end of the LANCE transfer (e_hold_ unasserted) is detected. If the last byte of a cache line is written to while the LANCE is still transferring, the dirty bytes of that cache line will be queued for draining to memory and data will immediately start to be packed into the other cache line. Any time the end of a LANCE transfer is detected, all dirty bytes in the LANCE cache are queued for draining to memory. This rule ensures that no data is left in the pack buffer when the LANCE interrupts the CPU.

Note that if some I/O device connected to the LANCE interface writes to the S4-DMA+ at a variety of addresses without unasserting e_hold_, it is possible arrive at a situation where both E-cache lines contain dirty

bytes when the I/O device attempts to write to an address whose tag does not match one of the E-cache line tags. In this case, both E-cache lines will automatically be queued for draining in order to make room for the new data to be written.

Note that if an I/O device connected to the LANCE interface makes more than one write to the same address without unasserting e_hold_, the data previously stored at that address in the E-cache will be written over each time new data is written to that address.

4.2 Transfers from memory

If the data requested by the LANCE is not in E-cache or marked as not valid, then the Least Recently Used (LRU) E-cache line will be filled with the line from memory that contains the data requested. As soon as the data requested by the LANCE is written to the E-cache, it is transferred on to the LANCE (even if the entire E-cache line has not yet been filled). Also, the word containing the requested data will be the first one to be read from memory.

If the data requested by the LANCE is in the E-cache and it is valid, then that LANCE gets a cache hit, and the requested data is transferred to the LANCE and marked as no longer valid in the E-cache. If part or all of the data on a cache hit comes from the byte in the E-cache line that is described by bits 16:19 of the CSR (see sec. 5.7), and the next cache line is on the same 4k page as the current line, and data is not being filled into the E-cache from memory, then a read-ahead operation will be performed. When a read-ahead occurs, the E-cache line that does not contain the data requested by the LANCE will be filled with the next line from memory. The read-ahead happens concurrently with the transfer of the requested data to the LANCE.

When a line of data is prefetched, that line is marked as Most Recently Used (MRU). Otherwise, the line that was most recently accessed by the LANCE is marked as MRU. The line that is not marked as MRU is marked as LRU.

While interfacing with the LANCE* , RD_MARK = CSR(16:19) = 14 (dec). Positioning the read-ahead line here, at the second to the last byte of the line, causes data from large blocks to be automatically pre-fetched, but does not cause superfluous pre-fetches on descriptor reads. It also insures that pre-fetched lines will not be clobbered by descriptor reads because the line whose read-ahead boundary has been crossed can be marked as least recently used.

* In this case, "LANCE" really means LANCE instead of "An I/O device that connects to the LANCE interface of the S4-DMA+."

A CPU slave write to any of the LANCE's internal registers will cause all bytes in the E-cache to be marked not valid. This ensures that no prefetched data from one LANCE read can be used as valid data in a LANCE read that takes place after the LANCE has been reprogrammed. It also provides for an easy way of clearing the E-cache valid bits, if it is necessary.

4.3 Memory errors

A transfer from the LANCE to memory is composed of two separate transfers: one from the LANCE to the S4-DMA+, and another from the S4-DMA+ to memory. Similarly, a transfer from memory to the LANCE is composed of a transfer from memory to the S4-DMA+, and another from the S4-DMA+ to the LANCE. A memory error can only occur in a transfer between the S4-DMA+ and memory. Note that the S4-DMA+ reports memory errors on transfers to and from the LANCE by forcing the LANCE to time out. The S4-DMA+ forces the LANCE to time out by withholding the e_rdy_ signal after it has given an e_hlda_ signal. When the LANCE times out, it generates an interrupt to the CPU.

To restart LANCE DMA activity after an error, it will be necessary to write to the LANCE's internal registers. This slave write to the LANCE will reset the E-cache, causing all bytes to be marked as not valid.

4.3.1 For a memory time-out or protection error on a transfer from the LANCE to memory, the S4-DMA+ behaves in the following way:

Each transfer from the LANCE to the S4-DMA+ is completed before the data from that transfer is sent on to memory. Therefore, if an error occurs in the transfer to memory, the S4-DMA+ is not able force the LANCE to time out until the LANCE tries to transfer more data. The LANCE is guaranteed to request another transfer within 1.6 milliseconds of its last transfer (assuming it is enabled). Therefore, 1.6 milliseconds is the maximum possible latency from the time the error occurs, until the LANCE is forced to time out. After the transfer that caused the error, no further transfers will take place from the S4-DMA+ to memory; however, up to seven additional words could be transferred from the LANCE to the S4-DMA+ before the LANCE is forced to time out.

4.3.2 For a parity, memory time-out or protection error on a transfer from memory to the LANCE, the S4-DMA+ behaves in the following way:

Since the S4-DMA+ reads data that the LANCE has not yet requested (without crossing page boundaries), it is possible that an error could be detected on a data read that LANCE has never requested. If this happens, the LANCE will be forced to time out the next time it requests data. If an error occurs on a memory read that LANCE did request, the data from the transfer that caused the error will never reach the LANCE. Instead, the S4-DMA+ will immediately force the LANCE to time out.

4.4 Slave accesses

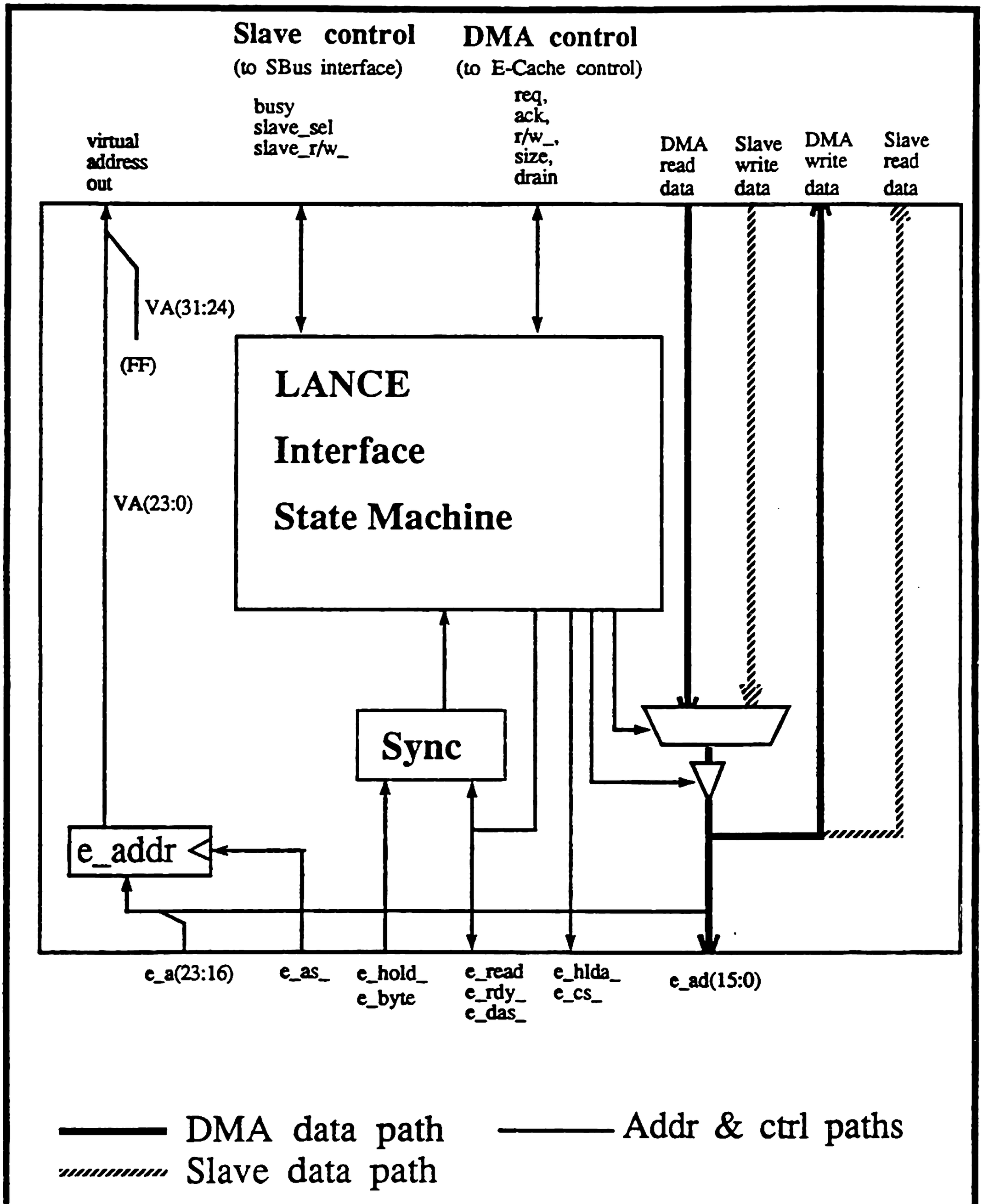
Slave accesses to the LANCE registers will be completed with a 16-bit port acknowledge signal (sb_ack8 & sb_ack32), as defined in the SBus specification. (See timing AM7990 LANCE documentation for timing diagrams of slave accesses to the LANCE).

Access to registers on the LANCE is achieved, through the S4-DMA, using Slave Cycles. An address map is given below. (The sb_a column of this table is specific to the way LANCE is connected on Campus and Calvin).

sb_sel_ & sb_as_	sb_a(X:Y)	sb_a(1)	Register accessed	Size	Type
0	11	0	Register Data Port (RDP)	16-bit	R/W
0	11	1	Register Address Port (RAP)	16-bit	R/W

Note that a word access to the LANCE registers would cause an access to the RDP and then the RAP.

4.5 LANCE Interface Block Diagram



5.0 ESP Interface

The S4-DMA+ gate array provides paths for the ESP to transfer data to and from memory and for the CPU to access control and status registers on the ESP. In order to support the DMA activity of the ESP the S4-DMA+ contains two 24 bit address counters, and two 32 bit control/status register, as well as two 24 bit byte counters which may be used for more generic control of future devices. The ESP has a 16 bit transfer count, however the address counter supports a 24 bit count for future enhancements.

Between the ESP interface and the SBus interface is a 2 line, 4-word/line cache (D-cache) with consistency control logic. (This cache is bypassed for slave accesses to the ESP.) Each byte in the D-cache has its own valid/dirty bit, and each line has a bit that determines the meaning of the valid/dirty bits for that line (i.e. valid or dirty). Bytes being transferred to memory are marked dirty as they are loaded into the D-cache. Bytes being transferred from memory are marked as valid as they are loaded into the D-cache. Consistency control ensures that all data written by the ESP gets to main memory in a deterministic manner, and is handled completely in hardware.

The ESP has an 8 bit data bus, so in order to reduce the impact on the SBus bandwidth, all ESP DMA transfers are packed into or unpacked from the D-cache. For error recovery, the status/control register in the S4-DMA+ gate array contains a FLUSH command (see definition in section 5.7)

5.1 Transfers to memory

As each byte of data from the ESP is packed into the D-cache it is marked as "dirty" until the last byte (byte 15) in a D-cache line is written to. Then, the dirty bytes of that cache line will be queued for draining to memory and data will immediately start to be packed into the other cache line. Any time an ESP interrupt, or byte count expiration, or CPU slave access (read or write) to an ESP-related register* is detected, all dirty bytes in the ESP cache are queued for draining to memory, unless an memory error has occurred. Bytes that are queued for draining to memory are drained automatically by hardware. A mechanism for determining when the dirty bytes have been drained is provided in the CSR register, which is defined in section 5.7.

* ESP related registers include NEXT_ADDR, NEXT_BCNT, D_ADDR, D_BCNT, D_ CSR, or any register in the ESP itself. These registers, and byte count expiration are defined in sections 5.5 through 5.9.

5.2 Transfers from memory

If the data requested by the ESP is not in the D-cache or marked as invalid, then the Least Recently Used (LRU) D-cache line will be filled with the line from memory that contains the data requested. As soon as the data requested by the ESP is written to the D-cache, it is transferred on to the ESP (even if the entire D-cache line has not yet been filled). Also, the word containing the requested data will be the first one to be read from memory.

If the data requested by the ESP is in the D-cache and it is valid, then it is transferred to the ESP and marked as no longer valid. If this data is the 9th byte in a D-cache line (i.e. the last four bits of its address are 1000) and the next line is on the same 4K page as the current line, then a read-ahead operation will be performed. When a read-ahead occurs, the D-cache line that does not contain the data requested by the ESP will be filled with the next line from memory. The read-ahead happens concurrently with the transfer of the requested data to the ESP.

When a line of data is prefetched, that line is marked as Most Recently Used (MRU). Otherwise, the line that was most recently accessed by the ESP is marked as MRU. The line that is not marked as MRU is marked as LRU.

5.3 Memory errors

A transfer from the ESP to memory is composed of two separate transfers: one from the ESP to the S4-DMA+, and another from the S4-DMA+ to memory. Similarly, a transfer from memory to the ESP is composed of a transfer from memory to the S4-DMA+, and another from the S4-DMA+ to the ESP. A memory error can only occur in a transfer between the S4-DMA+ and memory. Note that the S4-DMA+ reports memory errors on transfers to and from the ESP by setting the ERR_PEND bit in the control/status register, which generates an interrupt request to the CPU when enabled. The S4-DMA+ can also generate an interrupt request to the CPU when the ESP requests an interrupt (asserts d_irq_).

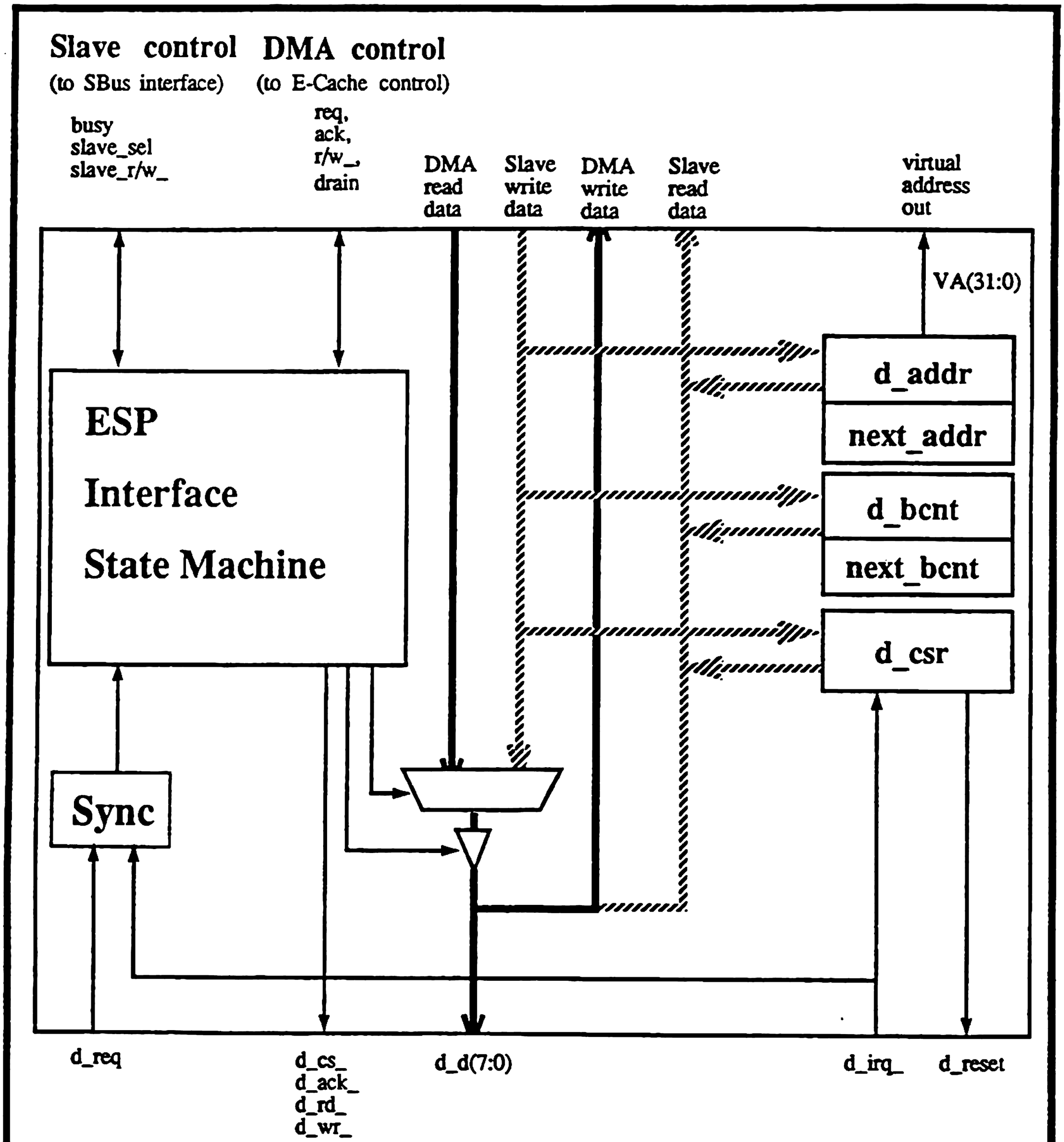
5.3.1 For a memory time-out or protection error on a transfer from the ESP to memory, the S4-DMA+ behaves in the following way:

The ERR_PEND bit in the S4-DMA+ CSR is set as soon as the error is detected, causing it to generate an interrupt. After the transfer that caused the error, no further transfers will take place from the S4-DMA+ to memory; however, up to seven additional words could be transferred from the ESP to the S4-DMA+ before the interrupt is generated. The interrupt is active until the ERR_PEND bit has been cleared by setting the S4-DMA+'s flush bit or reset bit.

5.3.2 For a parity, memory time-out or protection error on a transfer from memory to the ESP, the S4-DMA+ behaves in the following way:

Since the S4-DMA+ reads data that the ESP has not yet requested (without crossing page boundaries), it is possible that an error could be detected on a data read that the ESP has never requested. Whether the ESP requested the memory read that caused the error or not, the data from that read will not be written to the ESP. Instead the ERR_PEND bit is set, which causes the S4-DMA+ to generate an interrupt. No transfers to the ESP take place until that bit is cleared.

5.4 ESP Interface Block Diagram



5.5 Registers inside the ESP

Access to control and status registers in the ESP on a Campus or a Calvin is achieved using the address map below. These addresses are external to the S4-DMA+ gate-array, however the gate-array does decode the appropriate higher order address bits, sb_a(X:Y) = 10, and generate the chip select and read/write lines to the external device.

sb_a(5:0)	Register accessed	Size	Type
0	Transfer Count Low	8-bit	R/W
4	Transfer Count High	8-bit	R/W
8	FIFO Data	8-bit	R/W
C	Command	8-bit	R/W
10	Status/Bus ID	8-bit	R/W
14	Interrupt/Status Time-out	8-bit	R/W
18	Seq. Step/Synch transfer period	8-bit	R/W
1C	FIFO flags/Synch offset	8-bit	R/W
20	Configuration	8-bit	R/W
24	Clock Conversion Factor	8-bit	R/W
28	ESP TEST (chip test use only)	8-bit	R/W
2C	Configuration 2 (ESP-2 only)	8-bit	R/W
30-3C	reserved	8-bit	R/W

The S4-DMA+ responds to slave accesses of these registers with sb_ack8.
Note that on Campus or Calvin, a word access to the ESP register at address 4, for example, would cause accesses to the registers at addresses 4, 5, 6, and 7. Since there are no registers defined for addresses 5, 6 and 7, the word access would produce undefined results.

5.6 ESP support registers inside the S4-DMA+

The registers which provide status and control for the Enhanced SCSI Processor (ESP) are selected by the host when sb_sel_ and sb_as_ are zero and sb_a(X:Y) = 01. The Next feature registers are selected when EN_NEXT= 1.

sb_a (3 : 2)	EN_NEXT	Register Accessed	Size	Type
0 0	X	Control/Status Register (CSR)	32-bit	R/W
0 1	0	Address Register (D_ADDR)	32-bit	W
0 1	1	Next Address Register (NEXT_ADDR)	32-bit	W
0 1	X	Address Register (D_ADDR)	32-bit	R
1 0	0	Byte Count Resister* (D_BCNT)	24-bit	W
1 0	1	Next Byte Count Register* (NEXT_BCNT)	24-bit	W
1 0	X	Byte Count Resister* (D_BCNT)	24-bit	R
1 1	X	Reserved for testing purposes	32-bit	R

* The Byte Count Registers are only used when enabled by bit 13, EN_CNT, in the DMA Control/Status Register.

5.7 Control/Status Register (CSR)

5.7.1 CSR definition

Bit	Mnemonic	Description	Type
31:0	D_CSR	Control/Status Register	R/W

The DMA Control/Status Register provides control and status information for the ESP DMA interface. Bit assignments are as follows:

Control/Status Register Assignments (CSR)

Bit	Mnemonic	Description	Type
0	INT_PEND	Set when d_irq active, or when TC set and TCI_DIS not set.	R
1	ERR_PEND	Set when memory time-out, protection error or parity error detected on ESP DMA transfer. DMA to/from ESP is stopped while this bit is set. Reset on FLUSH command or RESET.	R
3:2	DRAINING	Bits read as 1 if D_CACHE draining dirty data. Do NOT assert RESET or FLUSH or write to D_ADDR reg. (Was PACK_CNT in S4-DMA.)	R
4	INT_EN	When set, enables sb_irq_ to become active when INT_PEND or ERR_PEND is set.	R/W
5	FLUSH	When set, marks all bytes in D-cache as not valid. Causes ERR_PEND and TC to be reset. Also causes A_LOADED and NA_LOADED to be reset if EN_NEXT= 1. Resets itself. Reads as 0.	W
6	SLAVE_ERR	Set on slave size error. Reset on CSR read. (Was DRAIN in S4-DMA.)	R
7	RESET	When set, acts as a hardware reset. (See section 5.7.2)	R/W
8	WRITE	DMA direction for ESP transfers; 1 = to memory	R/W
9	EN_DMA	When set, allows the S4-DMA+ to respond to DMA requests from the ESP if DMA not stopped on other conditions. (See DMA_ON bit.)	R/W
12:10	----------	Unused. Read as 0. (Were REQ_PEND & BYTE_ADDR in S4-DMA.)	R
13	EN_CNT	When set, enables the internal byte counter, D_BCNT, to be decremented on byte transfers between the ESP and S4-DMA+.	R/W
14	TC	Terminal Count; D_BCNT has expired. (See section 5.7.2)	R
15	ILACC	When set, modifies LANCE DMA read cycle. (See section 5.7.2)	R/W
19:16	RD_MARK	Read-ahead mark for E-cache; read-ahead occurs when LANCE reads the RD_MARKth byte of the E-cache. Defaults to 14 (decimal), which is the correct setting for the LANCE.	R/W
20	ALE/AS_	Defines pin 27 as ALE or AS_ (active high or active low); 1 = ALE, 0 = AS_. Defaults to 0.	R/W
21	LANCE_ERR	Set when a memory error occurs on a transfer to/from LANCE. Reset on a slave write to the LANCE. (See section 5.7.2)	R
22	----------	Reserved. Read as 0.	R/W
23	TCI_DIS	Set to disable TC interrupt. Defaults to 0.	R/W
24	EN_NEXT	When set, enables next address auto-load mechanism (see sections 5.7.2, 5.8 and 5.9). Defaults to 0.	R/W
25	DMA_ON	Reads as 1 when (A_LOADED or NA_LOADED) & EN_DMA & NOT ERR_PEND; otherwise, reads as 0. When set, indicates that the S4-DMA+ is able to respond to DMA requests from the ESP.	R
26	A_LOADED	Address Loaded. Set when D_ADDR register written (with EN_NEXT=0) or when NEXT_ADDR copied to D_ADDR. Reset by RESET or D_BCNT expiration. Also reset by FLUSH when EN_NEXT=1. (See section 5.7.2)	R
27	NA_LOADED	Next Address Loaded. Set when NEXT_ADDR written (with EN_NEXT=1). Reset by RESET,, EN_CNT = 0 or EN_NEXT = 0. Also reset by FLUSH when EN_NEXT=1. Also reset when NEXT_ADDR copied to D_ADDR. (See section 5.7.2)	R
31:28	DEV_ID	Device ID (For this implementation = 1001)	R

Note: During a transfer, a write to the DMA CSR should only change the TCI_DIS or EN_DMA bits. The following bits have changed meaning from the S4_DMA CHIP: PACK_CNT to DRAINING, DRAIN to SLAVE_ERR, REQ_PEND read as 0, DEV_ID changed to 1001. RD_MARK, ALE/AS_, LANCE_ERR, EN_NEXT, DMA_ON, A_LOADED, NA_LOADED and TCI_DIS are new.

5.7.2 Control/Status Register bit definitions

RESET - Bit 7.
RESET will remain active once it is written as a one until written as a zero. RESET from bit 7 or sb_reset_ will leave the device in the following state:
ERR_PEND = DRAINING = INT_EN = FLUSH = SLAVE_ERR = WRITE = EN_DMA = REQ_PEND = EN_CNT = TC = ILACC = ALE/AS_ = LANCE_ERR = EN_NEXT = DMA_ON = 0, RD_MARK = 14 (Dec), RESET = 1, and all valid/dirty bits in the D-cache set to not valid. All ESP interface state-machines will revert to their idle states, and the d_reset pin will be asserted for as long as the RESET bit or the sb_reset_ signal are active.

EN_CNT & EN_NEXT - bits 13 and 24
The EN_CNT and EN_NEXT control the modes of operation of the DMA Address register, Next Address register, Byte count and Next Byte count registers as follows:

EN_CNT	EN_NEXT	Mode
0	X	Backward compatible with S4_DMA
1	0	Backward compatible with S4_DMA using byte count Flush clears TC flag
1	1	Next Address and Next Count values loaded into Address and Count registers at Count expiration. Halt DMA on Count expiration if NEXT_ADDR not loaded

TC - Terminal Count - Bit 14
The TC bit in the CSR will be set when D_BCNT makes a transition from 000001 to 000000. When it is set, an interrupt will be generated on the d_irq_ pin (if enabled by INT_EN and not disabled by TCI_DIS).
When EN_NEXT= 0, TC is cleared only by FLUSH, RESET or sb_reset_.
When EN_NEXT= 1, TC can also be cleared by a CSR read.

ILACC - Integrated Local Area Controller Chip - Bit 15.
This bit 15, was originally intended to allow the S4-DMA to interface with the ILACC chip. Now, it merely modifies the LANCE DMA read cycle as follows:
When LANCE (or something in its place) is master, and it performs a read (from memory to the LANCE), the S4-DMA+ will stop driving data to the LANCE when the LANCE unasserts e_das_.

LANCE_ERR - Local Area Network Controller for Ethernet - Bit 21.
LANCE_ERR being set does not generate an interrupt. (For information on LANCE interface error handling, see section 5.3.)

A_Loaded & NA_Loaded - bits 26 and 27.
These bits define the validity of the values stored in the D_ADDR and NEXT_ADDR registers. When the state is reached where a valid NEXT_ADDR has been loaded, and the current D_ADDR has been marked as invalid (NA_LOADED = 1 & A_LOADED = 0), then the contents of the NEXT_ADDR register are copied to the D_ADDR register. The copy takes place on the same clock edge where the stated condition is sampled as true. So, if address chaining has been set up to take place when the byte count expires, the actual sequence of events will be the following: First, A_LOADED will be cleared (on expiration of byte count). Second, the NEXT_ADDR will be copied to the D_ADDR register. These two events will probably be seen as one by software.

5.8 Address Register (D_ADDR) & DMA NEXT Address Register (NEXT_ADDR)

Bit	Mnemonic	Description	Type
31:0	D_ADDR	Virtual Address Counter for ESP access VA(31:0)	R/W
31:0		NEXT_ADDR NEXT Address Register	

The value in this register after a RESET is indeterminate.

The Address Counter is a 24-bit counter and an 8-bit register. The lower 3 bytes constitute the address count and the upper byte is the register. It is assumed that transfers will not cross a 16MB boundary.

The Address counter always points to the next byte that will be accessed by the ESP, independent of which bytes in memory have been accessed by the S4-DMA+.

If the EN_NEXT (enable next address) bit in the CSR is set, then a write to the D_ADDR register will write to the NEXT_ADDR register instead. If EN_NEXT is set when the byte counter (D_BCNT) expires, and the NEXT_ADDR register has been written to since the last time the byte counter expired, then the contents NEXT_ADDR register are copied into the D_ADDR register. If EN_NEXT is set when the byte counter (D_BCNT) expires, but the NEXT_ADDR register has not been written to since the last time the byte counter expired, then DMA activity is stopped and DMA requests from the ESP will be ignored until NEXT_ADDR is written to, or EN_NEXT is cleared. (Also, the DMA_ON bit will read as 0 while DMA is stopped because of this). When DMA is re-enabled by writing to the NEXT_ADDR register, the contents of the NEXT_ADDR register are copied to the D_ADDR register before DMA activity actually begins.

Note: A write to the D_ADDR register will reset the D-cache and mark all bytes in the D-cache as not valid. A write to the NEXT_ADDR register does not have this effect.

5.9 Byte Counter (D_BCNT) & DMA NEXT Byte Counter (NEXT_BCNT)

Bit	Mnemonic	Description	Type
23:0	D_BCNT	Byte Count; counts down to 0, then sets the TC bit in the CSR	R/W
23:0	NEXT_BCNT	NEXT Byte Count Register	

The value in this register after a RESET is indeterminate.

When reading this register as a word, bits 31:24 will read as 0s.

When enabled, the Byte counter is decremented every time a byte is transferred between the S4-DMA+ and the ESP in an ESP DMA cycle. It is decremented immediately after the byte has been transferred. It is not decremented on slave accesses to the ESP or on transfers between the S4-DMA+ and memory.

If the EN_NEXT bit in the CSR is set, then a write to the D_BCNT register will write to the NEXT_BCNT register instead. If the NEXT_ADDR register is being copied into the D_ADDR register, and EN_NEXT is set, then the NEXT_BCNT register will be copied into the D_BCNT register at the same time.

If EN_NEXT is not set when D_BCNT expires (changes from 00001 to 00000), then DMA activity between the ESP and the S4-DMA+ will be stopped and the DMA_ON bit will read as 0 until D_ADDR is written to

(and EN_DMA is set). If EN_NEXT is set, then DMA will be stopped on D_BCNT expiration as described in section 5.8.

Note: Loading D_BCNT with 0 will allow 2**24 bytes to be transferred before it expires. (See sections 5.7 - 5.9 for descriptions of the various things that happen when the byte counter expires).

5.10 Programming Notes

5.10.1 Stopping DMA to/from the ESP

If the driver desires to stop a DMA transfer in progress, the following commands should be issued to the CSR. First, the EN_DMA bit should be cleared to ignore new transfer requests from the ESP. Memory accesses by the SBus interface of the S4-DMA+ are unaffected by this bit. The EN_DMA bit can be set or cleared at any time without affecting the state of a transfer currently in progress. Once the EN_DMA bit has been cleared, the REQ_PEND bit should read 0 -indicating that it is safe to set the FLUSH or RESET bits. The act of clearing the EN_DMA bit will cause all dirty bytes in the D-cache to immediately be drained to memory (and marked as no longer dirty in the cache). Bytes that are marked as valid in the D-cache remain marked as valid when the this automatic draining function is performed. To mark them as not valid, the driver should issue a FLUSH command to the CSR.

DMA will automatically be stopped when a memory error occurs (see section on memory errors). If the driver wishes to clear the ERR_PEND bit and mark all D-cache bytes as not valid (as it would in an error condition) then it should simply issue the FLUSH command. Note that the ERR_PEND bit must be cleared before another DMA transfer can be initiated.

Note that DMA activity does not stop when the ESP asserts d_irq_.

5.10.2 Starting DMA to/from the ESP

If DMA was stopped by the driver as described above, then it can be restarted simply by setting the EN_DMA bit. If EN_NEXT = 0, the address in d_addr will be correct because it always points to the next byte to be accessed by the ESP. If the driver wishes to clear all prefetched data from the D-cache before restarting DMA, it can issue a FLUSH command or write to the d_addr register. When starting a new transfer or restarting a transfer after a memory error has occurred, a new value should be written to d_addr.

5.10.3 Use of Internal Byte Counter with Next Address feature disabled

When using the internal Byte Counter and the TC flag in the CSR with EN_NEXT = 0, it is necessary to perform the following procedure for correct operation.

Load Byte Count into D_BCNT

Load DMA address into D_ADDR

Load Peripheral Device with relevant command(s)

Load CSR with enables and direction bits (EN_DMA EN_CNT and WRITE)

Data will be transferred as directed until the Byte Count expires, at which point the TC flag will be set in the CSR and an interrupt will be generated, if enabled. DMA will also be stopped at this time (DMA_ON will be cleared). DMA will remain stopped, independent of the value of EN_DMA, until D_ADDR is loaded with a new value. (Note: this implies that the interrupt service routine should clear EN_DMA before writing a new address to D_ADDR.) Once the CPU has serviced the interrupt and wishes to start another DMA operation it is necessary to proceed as follows;

Issue a FLUSH command and then repeat the above procedure. (Clears TC bit of CSR)

5.10.4 Use of Internal Byte Counter with Next Address feature enabled

When using the internal Byte Counter, the TC flag in the CSR and the NEXT_ADDRESS feature, it is necessary to perform the following procedure for correct operation.

NOTE: This is a suggested procedure since several methods of programming the chip are possible.

INITIALIZATION:
Initialization if the state of the chip is not defined such as after an error.
Write Control register:
TCI_DIS, EN_DMA, EN_NEXT, RESET = 0
FLUSH, EN_CNT, INT_EN = 1
WRITE, ALE/AS, ILACC, RD_MARK = value when read

MULTIPLE BLOCK TRANSFERS:
To do a multiple block transfer with an interrupt after each block:
Write Control register:
INT_EN, EN_DMA, EN_CNT, EN_NEXT = 1
TCI_DIS, FLUSH, RESET = 0
WRITE, ALE/AS, ILACC, RD_MARK = value as read or desired
Write D_BCNT with byte count of the first block.
Write D_ADDR with the starting address of the first block. Set-up and start the SCSI chip to do its transfer.
Write NEXT_BCNT with the byte count of the 'next' block.
-Loading of the NEXT_BCNT is optional because the initial loading of the D_BCNT also loads the next count register.
Write NEXT_ADDR with the address of the 'next' block. The transfer of the first block is enabled and the transfer of the next block will happen automatically when Terminal Count is reached, i.e.: the next address and byte counts will be used. (This assumes the loading of the NEXT count and address occur before the first block transfer is complete.)

INTERRUPT:
After each interrupt:
Read Status register:
TC and DMA_ON should both = 1.
- If DMA_ON = 0 then, the NEXT_ADDR register did not get updated or there is an error pending, ERR_PEND. (It could also mean the DMA is not enabled, EN_DMA, or the next address feature is not enabled, EN_NEXT, but they were set = 1 so this should not be the case.)
Write NEXT_BCNT with the byte count of the 'next' block.
-Loading of the NEXT_BCNT register is optional, but if loaded, must be loaded before the NEXT_ADDR register.
Write NEXT_ADDR with the address of the 'next' block.

LAST BLOCK:
If no Terminal Count interrupt is desired after the last block is transferred: (because we expect an interrupt from the SCSI chip at the end of the transfer.)
On interrupt of the next to last block:
Read Status register:

TC and DMA_ON should both = 1.
Write Control register:
TCI_DIS, INT_EN, EN_DMA, EN_CNT, EN_NEXT = 1
FLUSH, RESET = 0
WRITE, ALE/AS, ILACC, RD_MARK = value as read or desired
NOTE: The NEXT_ADDR register is not loaded so the transfer will stop.

NEXT TRANSFER:
The initialization for the next multi-block transfer is:
Write Control register:
INT_EN, EN_DMA, EN_CNT, EN_NEXT = 1
TCI_DIS, FLUSH, RESET = 0
WRITE, ALE/AS, ILACC, RD_MARK = value as read or desired
NOTE: This assumes the chip is in a known state because, if not, the write to the D_BCNT and D_ADDR registers may go to the NEXT registers (See INITIALIZATION).
Write D_BCNT with byte count of the first block.
Write D_ADDR with the starting address of the first block.
Set-up and start the SCSI chip to do its transfer.
Write NEXT_BCNT with the byte count of the 'next' block.
- Loading of the NEXT_BCNT is optional because the initial loading of the D_BCNT also loads the next count register.
Write NEXT_ADDR with the address of the 'next' block.

TAKING ADVANTAGE OF 'NEXT' FEATURE:
If an interrupt occurs and the 'next' address is not available, there is no room in the buffer or data is not available, the processor can take advantage of the fail-safe capability by not loading the NEXT_ADDR and the DMA will stop when the count goes to 0.
After the 'next' address becomes known the processor must determine if the transfer of the current block has completed. If it has not, the processor should load the 'next' count and address to continue the transfer. If the transfer has completed, the processor should load the current count and address and the next count and next address to restart the transfer

NOTES:
- There will be an interrupt generated when the count goes to zero.
- The firmware must understand that if the transfer has stopped because the NEXT_ADDR was not loaded before the terminal count was reached, it is effectively loading the D_BCNT and D_ADDR registers, not the NEXT registers. This will enable the DMA to restart the transfer.
- It is necessary to load the count before the address because the loading of the address register causes the restart of the transfer.

5.10.5 Additional notes

A REQ_PEND bit is provided in the CSR to show the driver if a transfer between the ESP and S4-DMA+ is in progress. Writing to the Address Counter (D_ADDR), changing the WRITE bit in the CSR, or writing the Byte Counter when the REQ_PEND bit is set will cause unknown problems.

Interrupts from the ESP are visible as INT_PEND in the DMA_CSR. The INT_EN bit is provided to enable or disable the generation of an interrupt to the IU. If an error condition exists during a memory access, (this could be parity errors, protection errors, time-outs etc.), the ERR_PEND bit will be set. This will cause an interrupt (if enabled) to the IU. Similarly, expiration of the Byte Counter will cause the INT_PEND bit to be set and an interrupt (if enabled). The ERR_PEND bit can only be cleared by a FLUSH or RESET command, or SB_RESET.

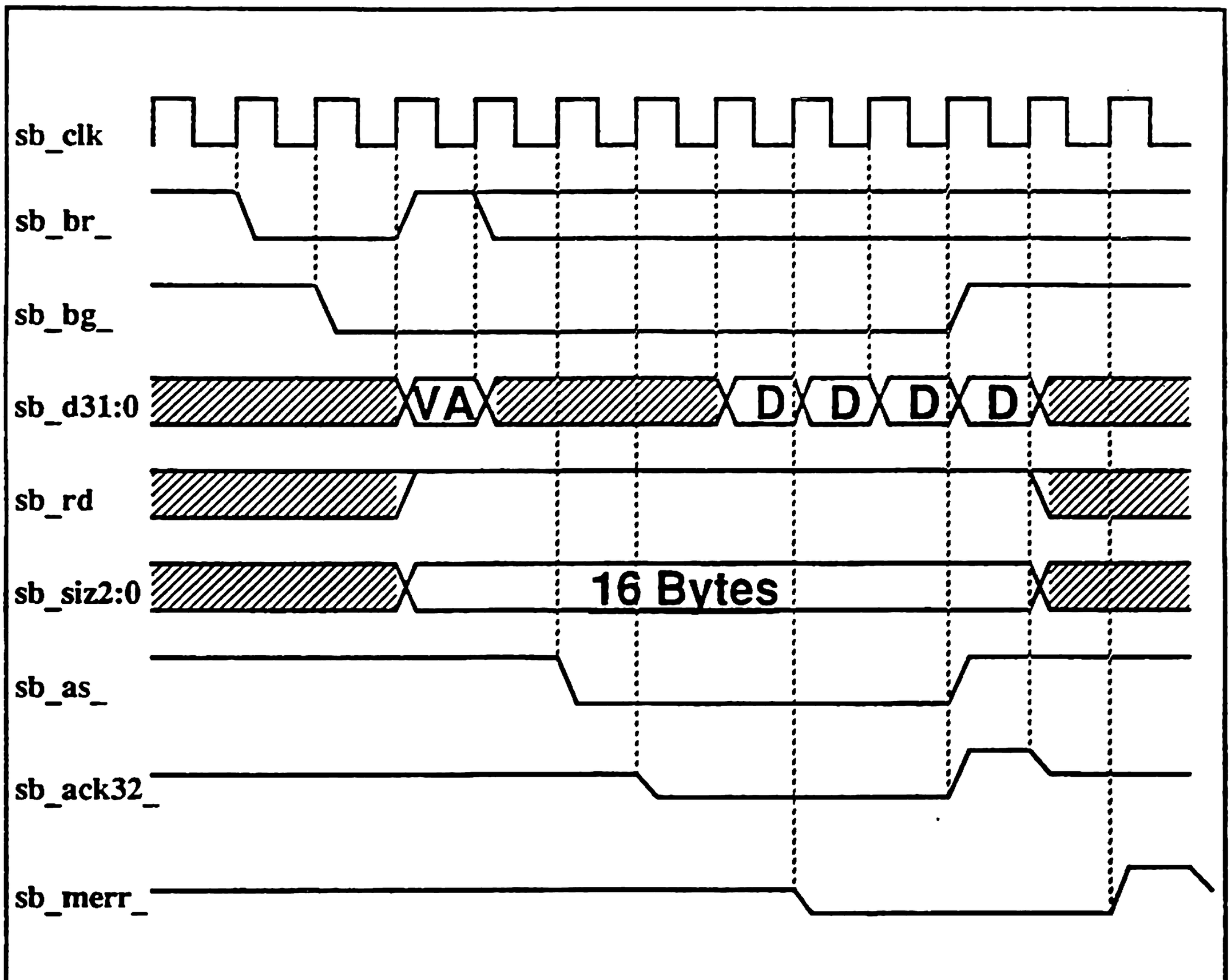
5.10.6 DIAGNOSTIC CAPABILITY

5.10.6.1 TESTING the 'NEXT' REGISTERS

When the processor writes to the D_BCNT and D_ADDR registers the data goes through the 'NEXT' registers. This 'feature' can be used to test the 'NEXT' registers since they can not be read directly.

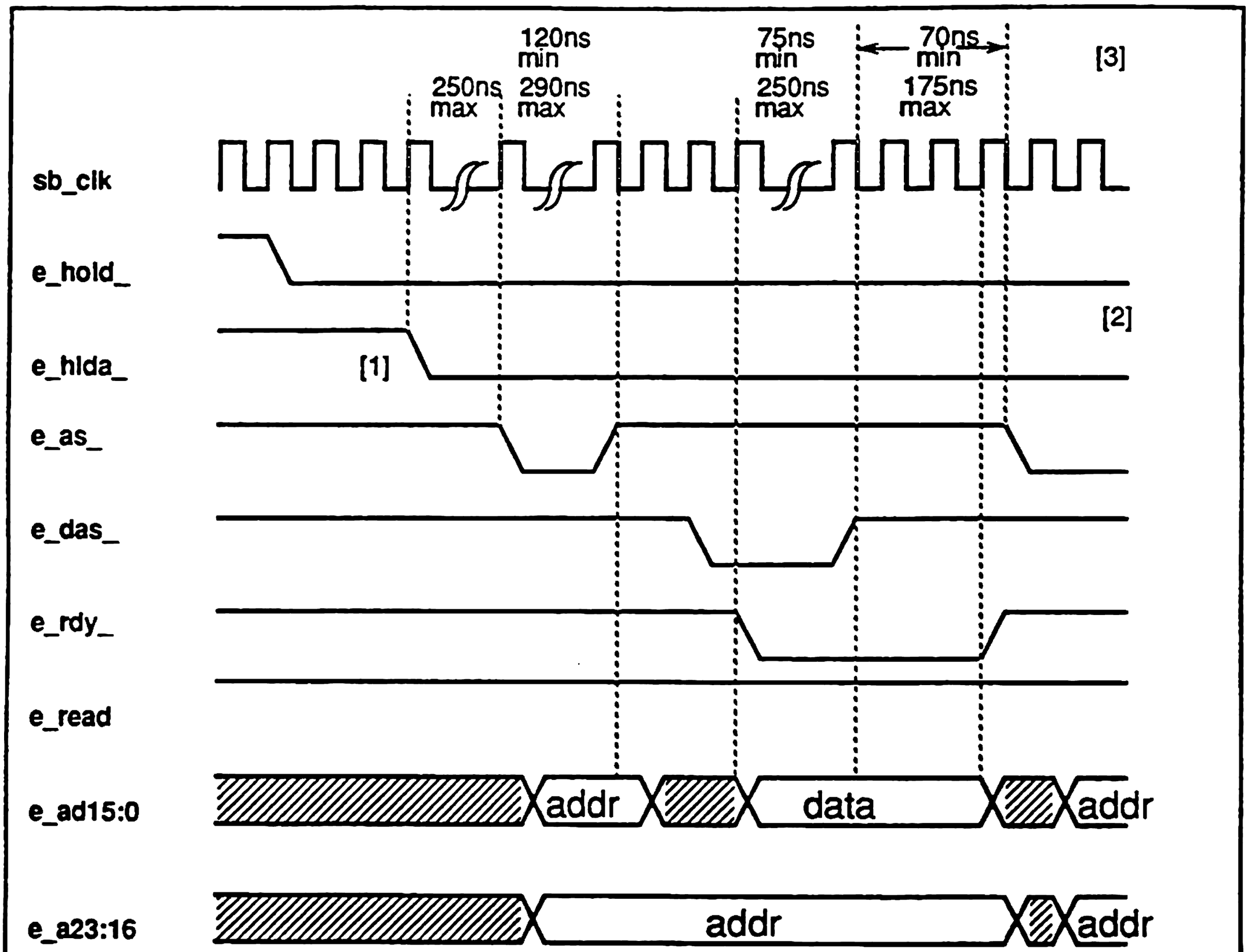
6.0 Timing Diagrams

Note that this section is missing diagrams for extended (slow) cycles for the ESP interface. For reference, they will be the same as for the S4-DMA. (Basically, slow cycles just extend the d_rd_ or d_wr_ pulse by 2 clocks, except for the slow DMA write cycle which extends the d_rd_ pulse by 4 clocks).



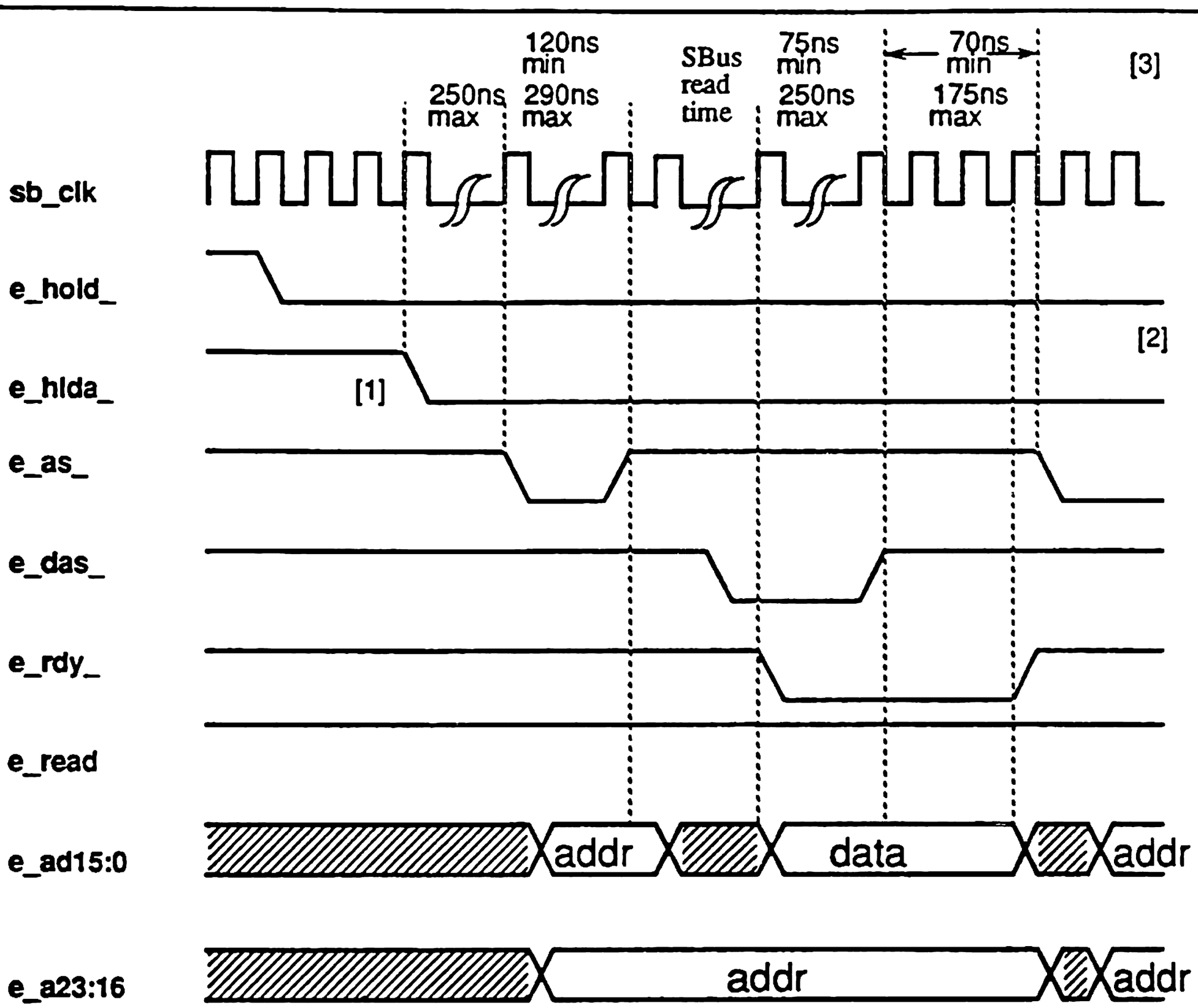
SBus DMA burst read (1 word/clk)
(Data flows from memory to the S4-DMA+)

For other possible SBus cycles, see the SBus specification; this is just one example.



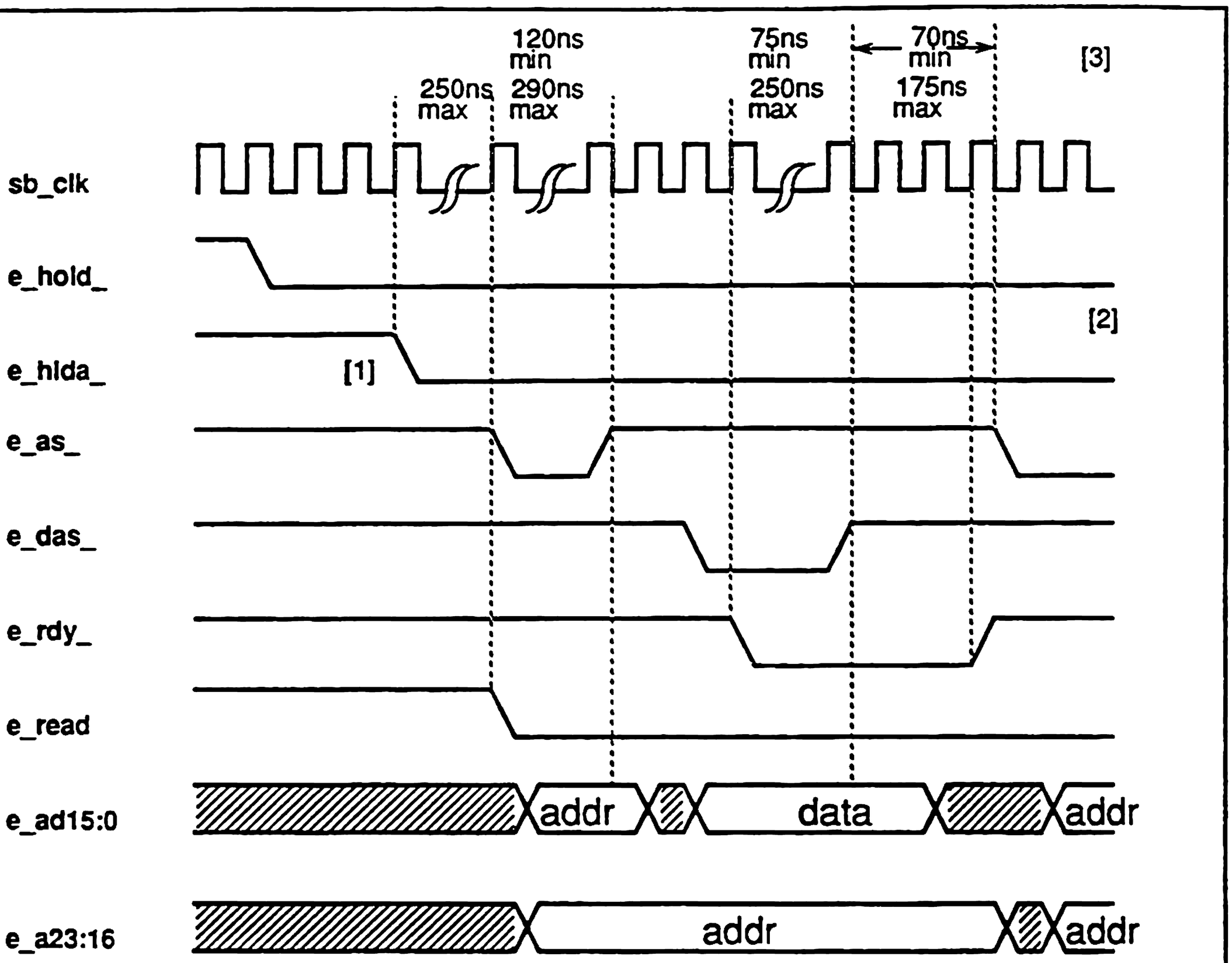
[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.
[3] min and max times given are those of the AMD 7990 LANCE

LANCE DMA burst read (from memory) with valid data in E-cache



[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.
[3] min and max times given are those of the AMD 7990 LANCE

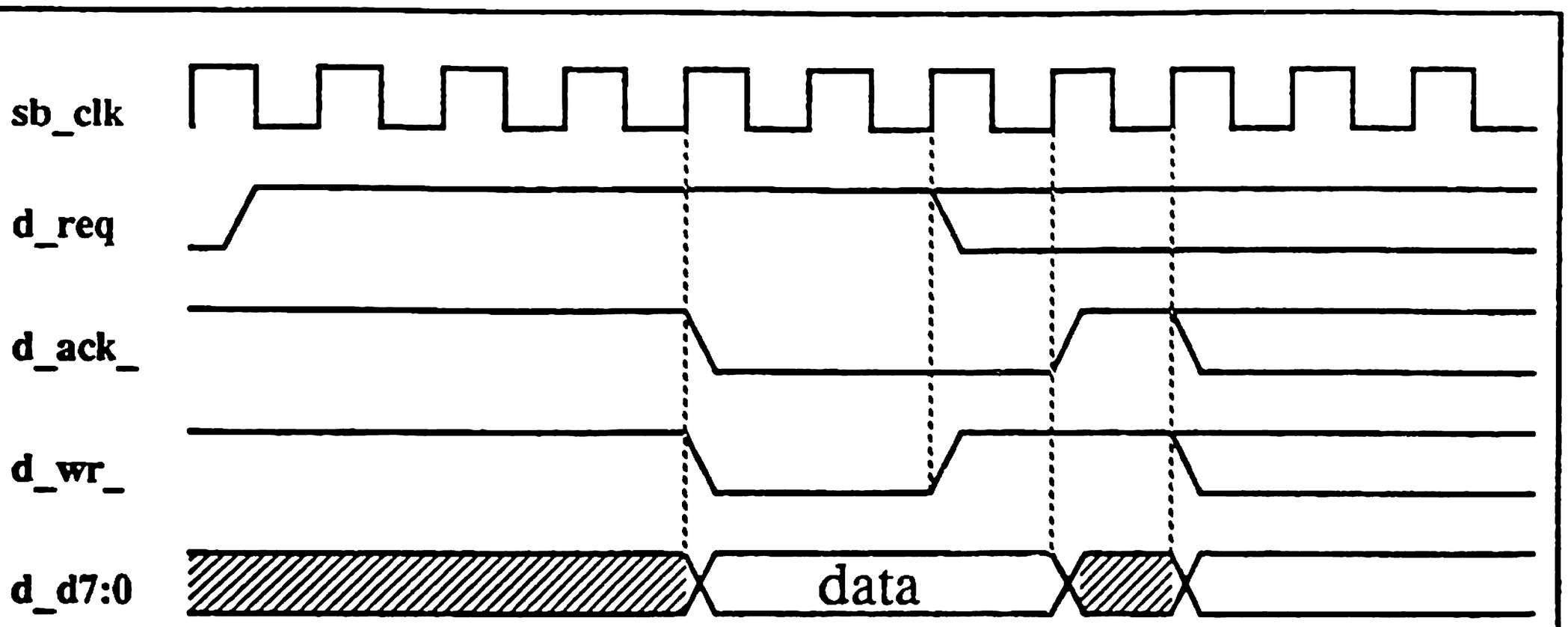
LANCE DMA burst read (from memory) with no data in E-cache



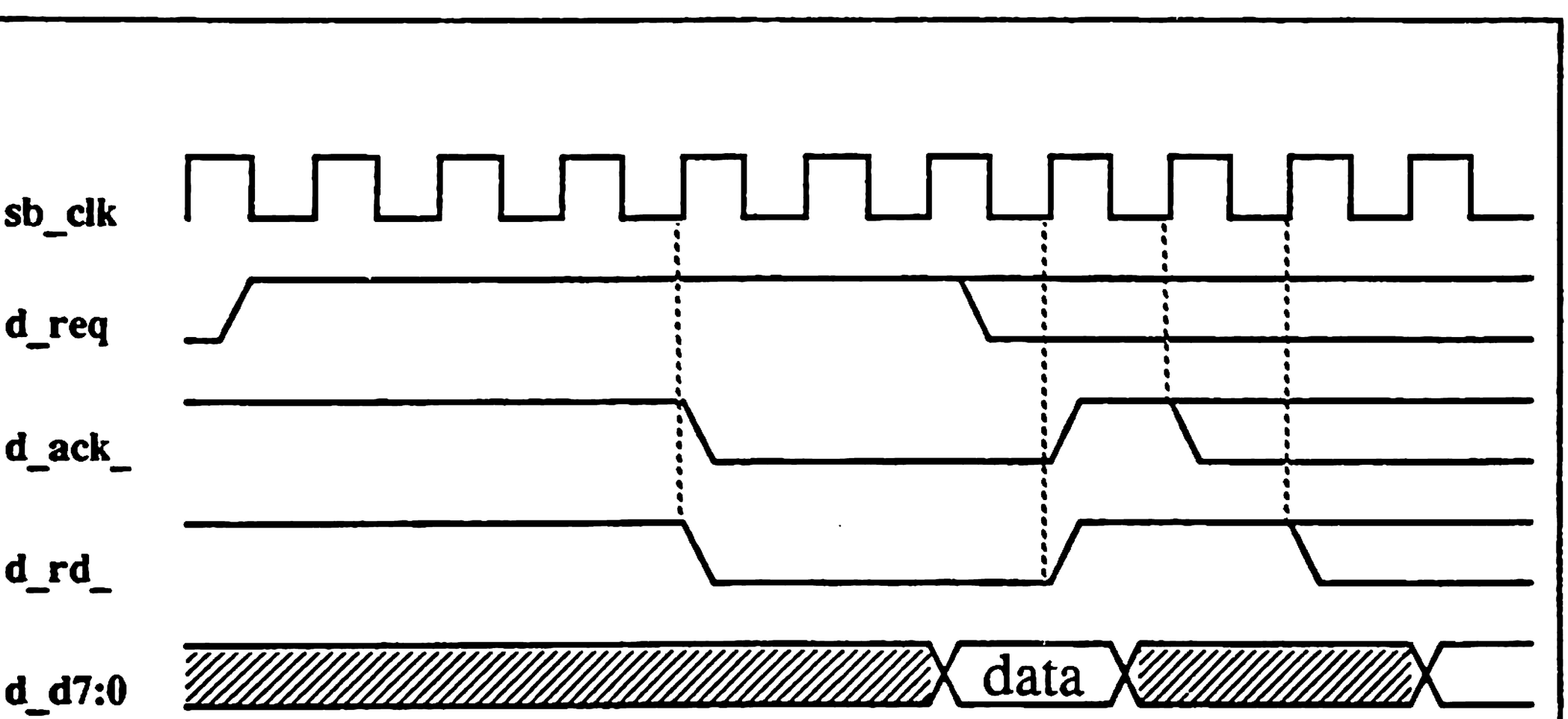
[1] e_hlda_ is only asserted when the interface is not busy.
[2] e_hold_ will stay asserted for burst mode accesses. e_hlda_ must follow it.
[3] min and max times given are those of the AMD 7990 LANCE

LANCE DMA burst write (to memory)

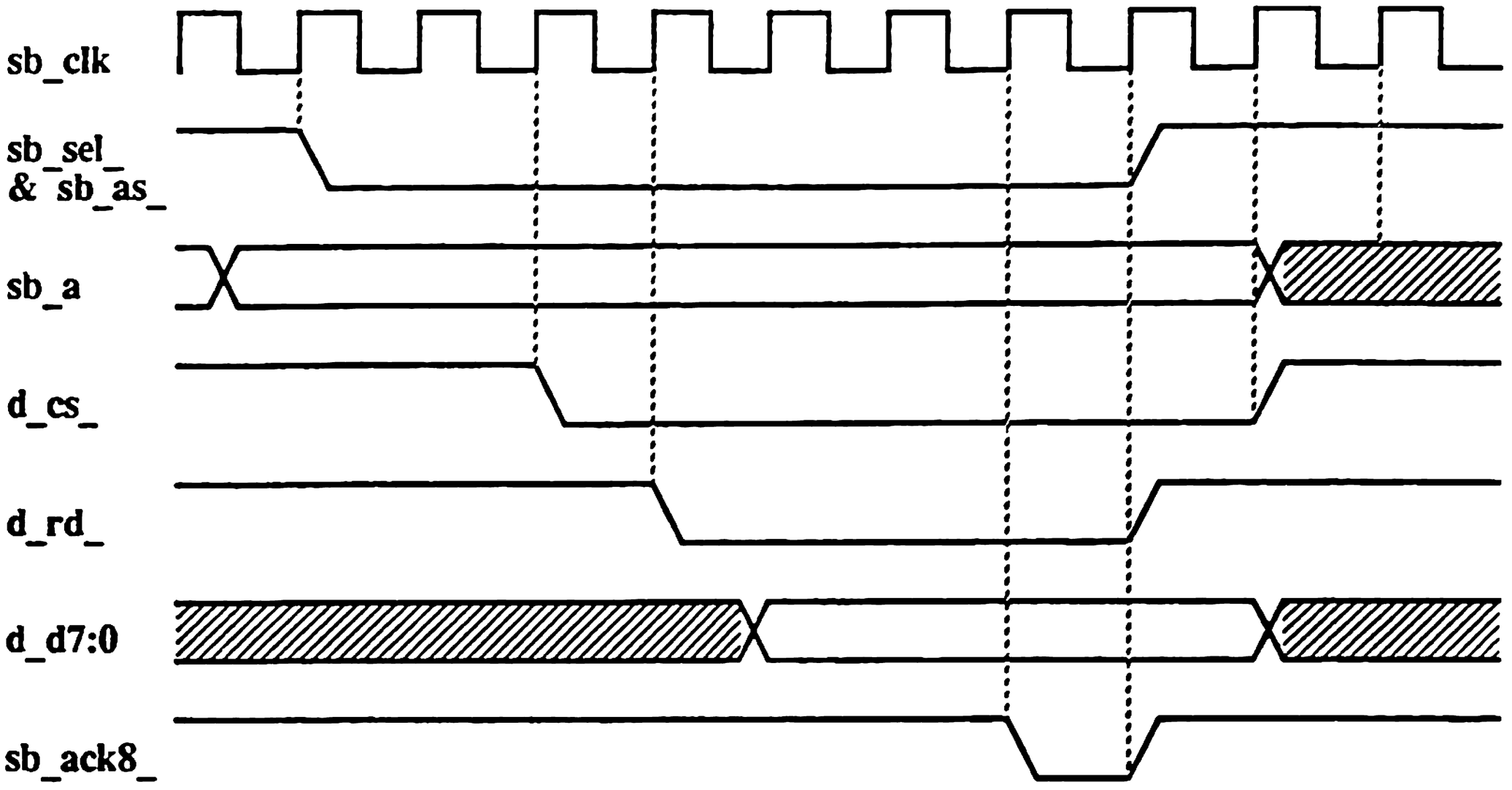
Note: assumes a cache line is available for the LANCE to write to



ESP DMA read (from memory) with valid data in D-cache [fast cycle]

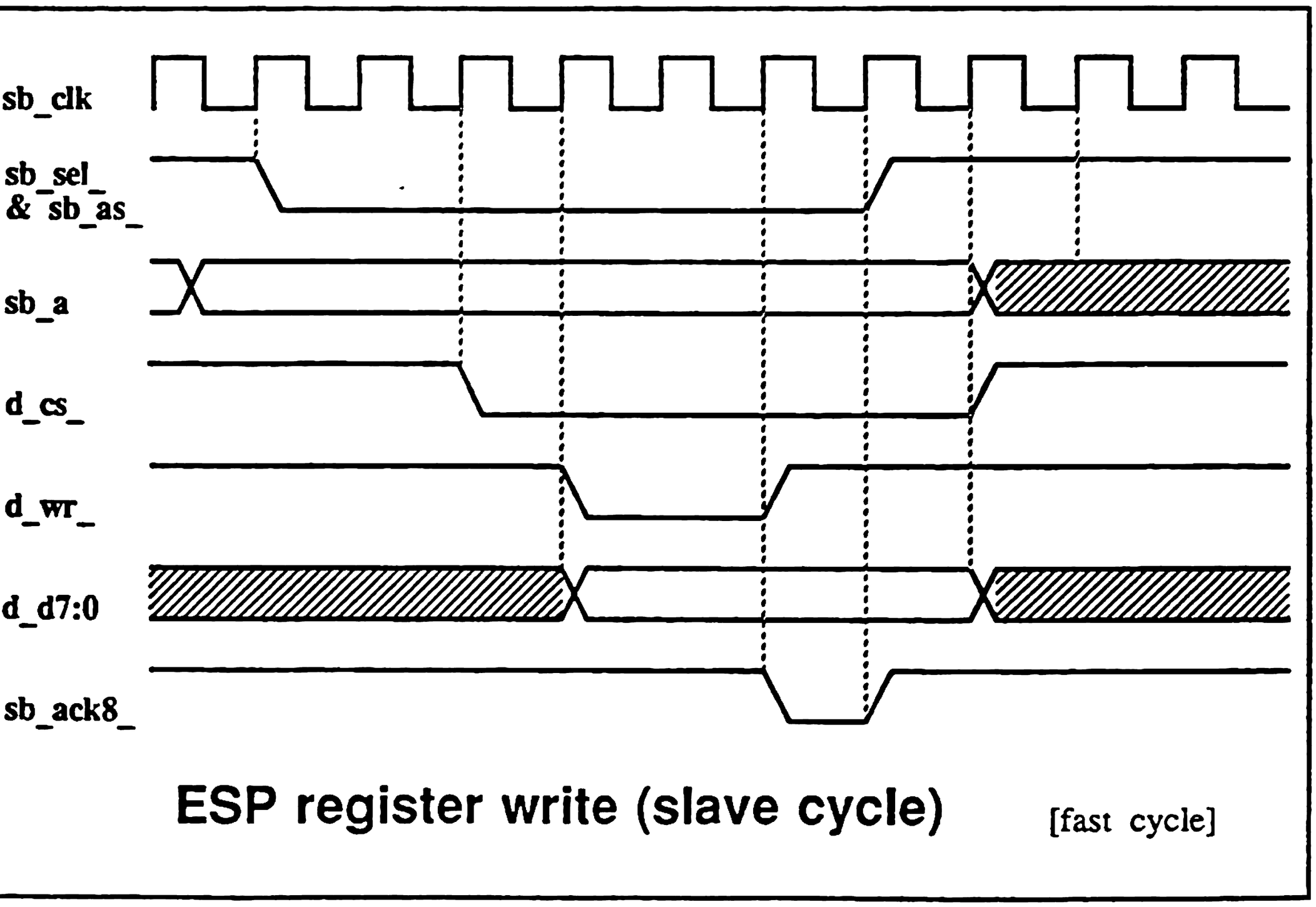


ESP DMA write (to memory) [fast cycle]



ESP register read (slave cycle) [fast cycle]

(Same cycle applies to offboard ID read cycle except id_cs_ is asserted instead of d_cs_)



ESP register write (slave cycle) [fast cycle]

7.0 Electrical Specifications

Operating Conditions

Temperature (Ta)	0 to +70C
Supply Voltage (Vcc)	

7.1 Capacitance

Symbol	Description	min	max	unit
Cin	input capacitance			pf
Cout	output capacitance			pf

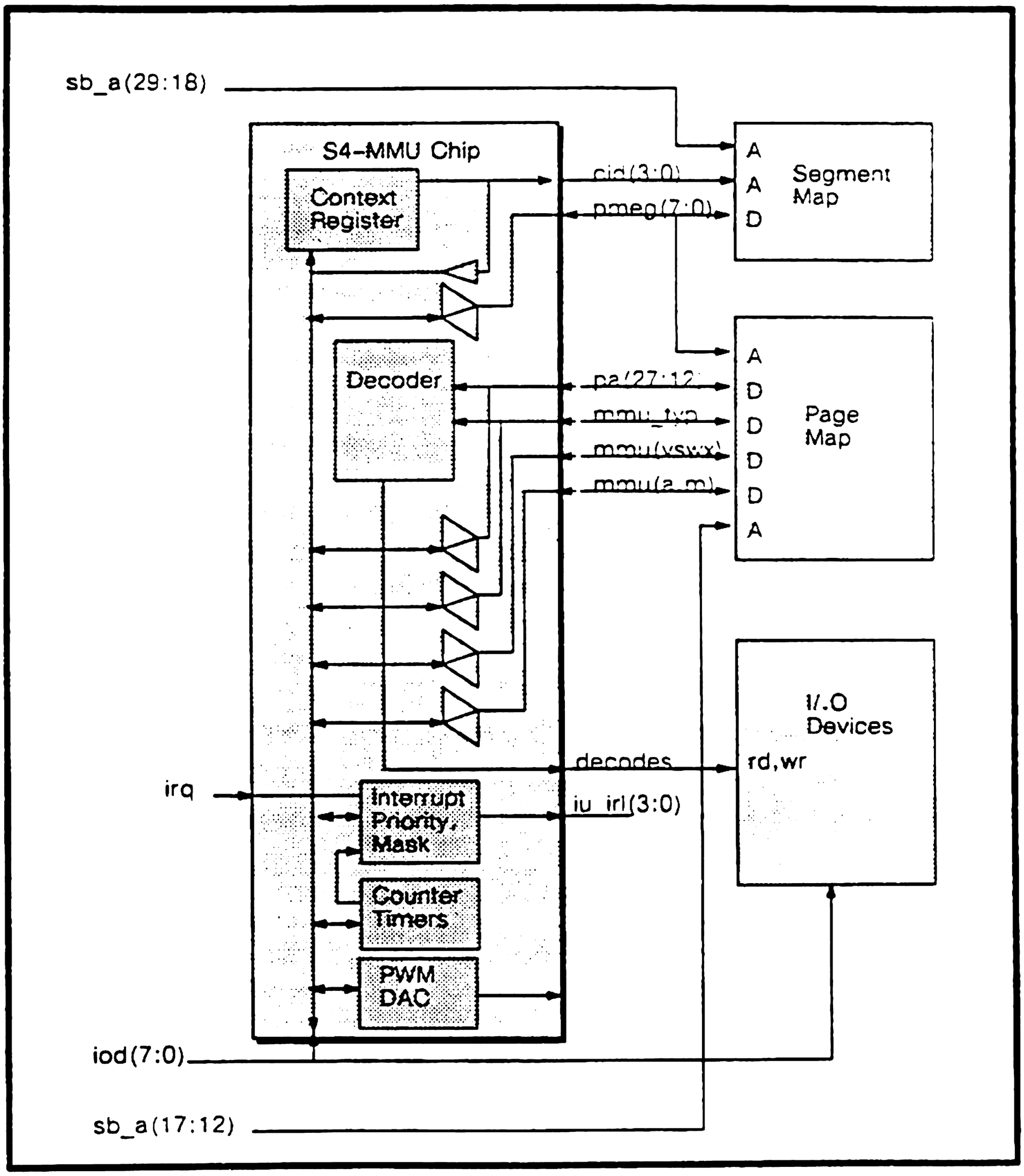
8.0 Revision history

3/29/89	cm	Created.
4/7/89	cm	Made auto-draining cache consistency function a mandatory function of the hardware; not an option. Also clarified sections on slave accesses and ID accesses.
6/3/89	cm	Clarified restrictions on when I/O devices can assert/unassert certain control signals. Updated block diagrams to reflect actual implementation. Updated section 2.1 to reflect decisions made on how the S4-DMA+ responds to rerun acknowledges when it is a master. Updated section 2.2 to reflect decisions made on how the S4-DMA+ acknowledges different types of slave accesses. Added Next Address and Next Byte Count functions. Changed CSR definition (section 5.7) to include these functions. Also changed CSR definition such that no bits that can be modified by hardware can be written to by the CPU (thus preventing destructive Read-Modify-Write accesses to the CSR).
6/19/89	curtr	Added pin numbers to pin list. Added some explanations and diagrams to describe the NEXT feature (formerly called the Next Address and Next Byte Count features). Changed addressing decode definition in section 5.6. Added notes about differences between S4-DMA and S4-DMA+.
6/19/89	dex	Added section 5.10.4: programming notes for NEXT feature.
6/29/89	cm	Made minor changes and clarifications to reflect actual hardware implementations. These changes were in the following areas: When an I/O device connected to the LANCE interface can/must assert/unassert the e_hold_ signal, when a read-ahead is performed and when it is not (for E-cache and D-cache), when the E-cache and D-cache are reset. Also changed the meaning of the REQ_PEND bit and the DRAINING bits in the CSR from the 6/3/89 revision of the Spec. Made slight modifications to ESP interface timing diagrams to reflect implementation (both master and slave cycles).
7/12/89	cm	Made clarifications of NEXT feature. Fixed ESP DMA timing diagrams. Removed REQ_PEND bit from CSR.

Sun Microsystems S4 Chip Set Sun SPARC(TM) Support System	**S4-MMU** Preliminary

Features

- Provides decodes and timing strobes for all Sun-4 Type 1 devices
- Replaces all MMU read/write buffers
- Automatically updates MMU statistics bits during bus cycles
- Prioritizes 15 levels of Interrupts
- Sun-4 Interrupt register provides software interrupts, interrupt enable
- 4-bit context register provides switchable MMU contexts
- Two counters generate high-resolution periodic interrupts



Pin Description

Symbol	Type	Description
iu_clk	DRVT4	Integer Unit clock. Main system clock.
const_clk	TLCHT	100 nsec clock divided by ten internally for counters.
io_a(3:0)	TLCHT	Buffered address bits from SBus.
ctl(2:0)	TLCHT	Encoded control space selects.
devspc_	TLCHT	Lo for device space accesses, high for control space.
sb_rd	TLCHTU	SBus Read. High during IU reads, low for writes.
as_	TLCHT	Address Strobe from S4-Cache. Indicates valid PAdr.
sb_reset_	TLCHN	SBus Reset. Lo for power-on & S/W resets.
iod(7:0)	BD4TU	Input/Output Data bus.
sb_ack8_	BT4	SBus 8-bit Acknowledge.
user_	TLCHN	Lo for User Space access, High for Supervisor space.
dma_	TLCHT	Lo during DVMA accesses; forces CID output to zero.
cid(3:0)	BT2	Context ID. Hi-order addresses into the Segment Map
pmeg(7:0)	BD4TD	Page Map Entry Group. Data bits from Segment Map.
pa(27:12)	BD4T	Physical Address bits. Output from Page Map.
mmu(v,w,s,x)	BD4T	MMU Valid, Write-allowed, Supv.-only, Don't Cache
mmu_typ(1:0)	BD4T	MMU Type bits.
mmu(a,m)	BD4T	MMU Accessed, Modified. Usage statistics.
sm_wr_	BT4	Segment Map Write.
pm_wr(2:0)_	BT4	Page Map Write bits.
ramsel_	BT2	DRAM select.
kbm_rd_	BT2	Keyboard/Mouse Read.
kbm_wr_	BT2	Keyboard/Mouse Write.
scc_rd_	BT2	Serial Controller Chip Read
scc_wr_	BT2	Serial Controller Chip Write
tod_cs_	BT2	Time-Of-Day Access
par_cs_	BT2	Parity Chip Select
eprom_rd_	BT2	EPROM Read
fd_rd_	BT2	Floppy Disk Read
fd_wr_	BT2	Floppy Disk Write
dac_wr_	BT2	Audio DAC write strobe.
dac_xfer_	BT2	Audio DAC transfer strobe.
iosel_	BT2	Select for Type 1 device, PA28 high.
sb_sel(3:0)_	BT2	SBus Selects. (includes Video)
iod_en_	BT4	Input/Output Data bus Enable.
aux_wr_	BT4	Auxiliary Output Write Strobe.
irq(15,13:11,9:1)_	SCHMITCN	Interrupt Requests (ext. pullups assumed).
iu_irl(3:0)	BT2	Integer Unit Encoded Interrupt Requests
od_.	TLCHTU	Output Disable input.
para	BT1	Parametric output.

Functional Description

Device Space and Control Space

The SPARC address space identifiers are divided into two "spaces" according to the following table:

ASI	Function	Space
0-1	Reserved	
2	IU Extensions	Control
3	Segment Map	Control
4	Page Map	Control
5-7	Reserved	
8	User Instruction	Device
9	Supervisor Instr.	Device
A	User Data	Device
B	Supervisor Data	Device
C-F	Reserved	

The signal DEVSPC_ chooses between device space and control space address maps. Device space devices are accessed with physical addresses provided by the MMU, while control space devices are accessed with virtual addresses provided by the SPARC processor.

Control Space

CTL(2:0) Encoding (Control Space Address Map)

ctl(2:0)	Device
0	Device on S4-Cache Chip
1	Reserved for VME IACK
2	Context Register *
3	Diagnostic Register (unused)
4	Serial Controller Chip (MMU Bypass)
5	Segment Map
6	Page Map
7	EPROM (Boot Cycle, Supv. Instr. Fetch)

* - Context reg access requires A0 low.

In Device Space (DEVSPC_ low) the ctl(0) input is used as an invalidation input for any cycle from the cache chip. It is used when the cache chip determines an illegal virtual address (a(31:28) not all ones or all zeroes) which the MMU cannot detect, to inhibit

generation of any device selects. In addition, in Device Space, the ctl(1) input is used to indicate an atomic LDST instruction is being executed. This requires that the selected page be writeable in order to pass protection checks (preventing the instruction from executing the load and then aborting the store, when it could have been predicted

Context Register

The context register drives the upper four address bits of the segment map. It allows address translations for sixteen separate contexts to be simultaneously stored by the MMU, and allows switching from the address translations belonging to one context to those of another context by performing a single write operation.

A shadow copy of the context register is also kept in the S4-Cache chip, where it is used as part of the cache tag comparator logic. On writes, both copies of the context register are modified. On reads, the S4-MMU context registers requires A0 low (even byte address) in order to respond. The S4-Cache copy responds to reads on odd byte locations.

The CID[3:0] output pins reflect the state of the context register and are fed to the Segment Map address inputs. When the DMA_ input is asserted, the CID outputs are forced to zero.

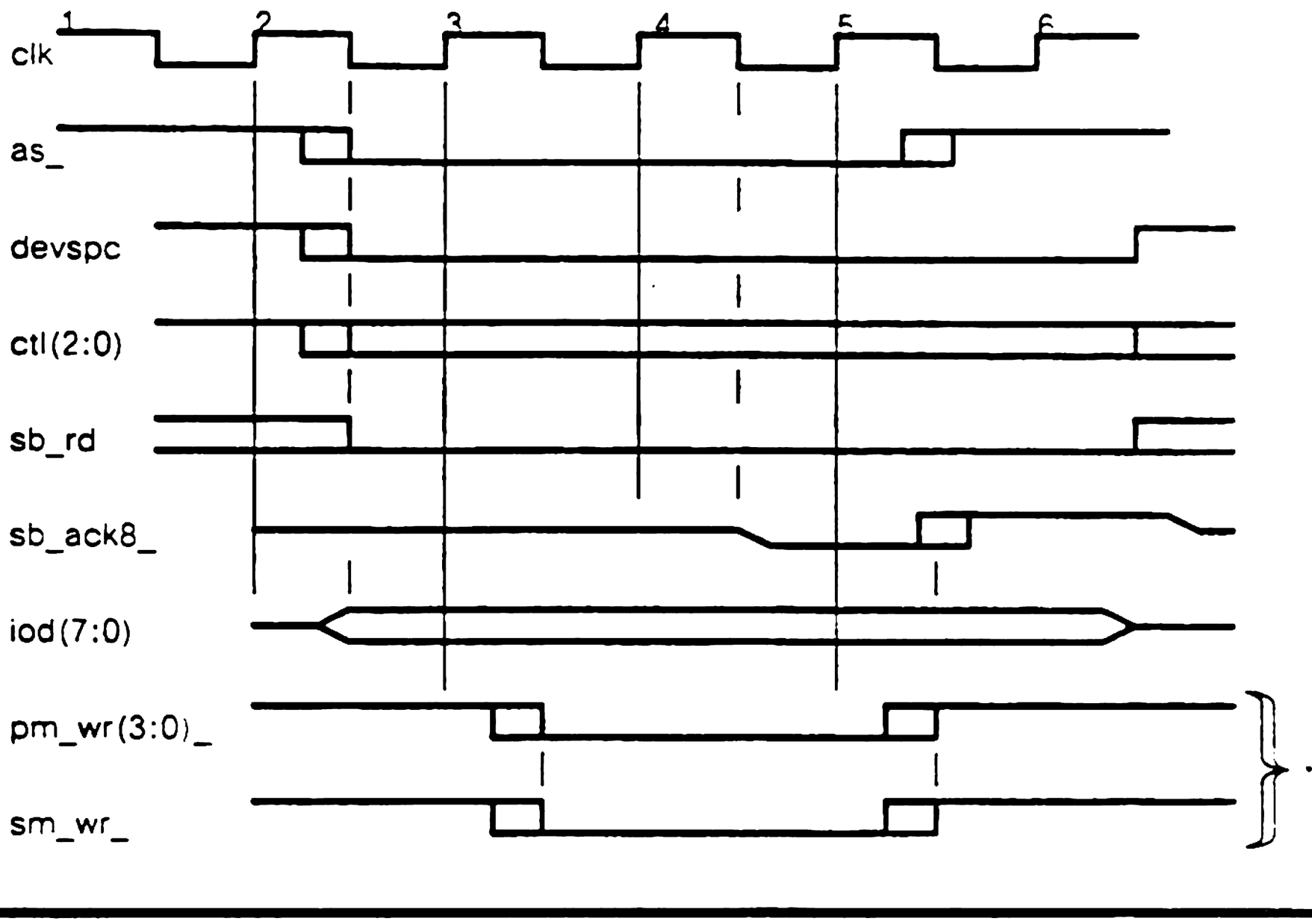
Serial Controller Chip

The serial controller chip (SCC) is accessible both in device space and control space. This allows diagnostic code to communicate with an asynchronous computer terminal even before initializing the MMU and even if the MMU is not operating correctly.

Segment Map

The segment map is the first level of the two-level MMU. It is indexed by the four bits of the context register concatenated with virtual address bits SB_A(29:18). The segment map entries (32k or 64k) are 7 or 8 bits wide and point to a Page Map Entry Group, or PMEG, in the page map. The segment map is loaded and read by the SPARC processor

though the PMEG(7:0) bus. The following diagram shows a Segment Map write cycle



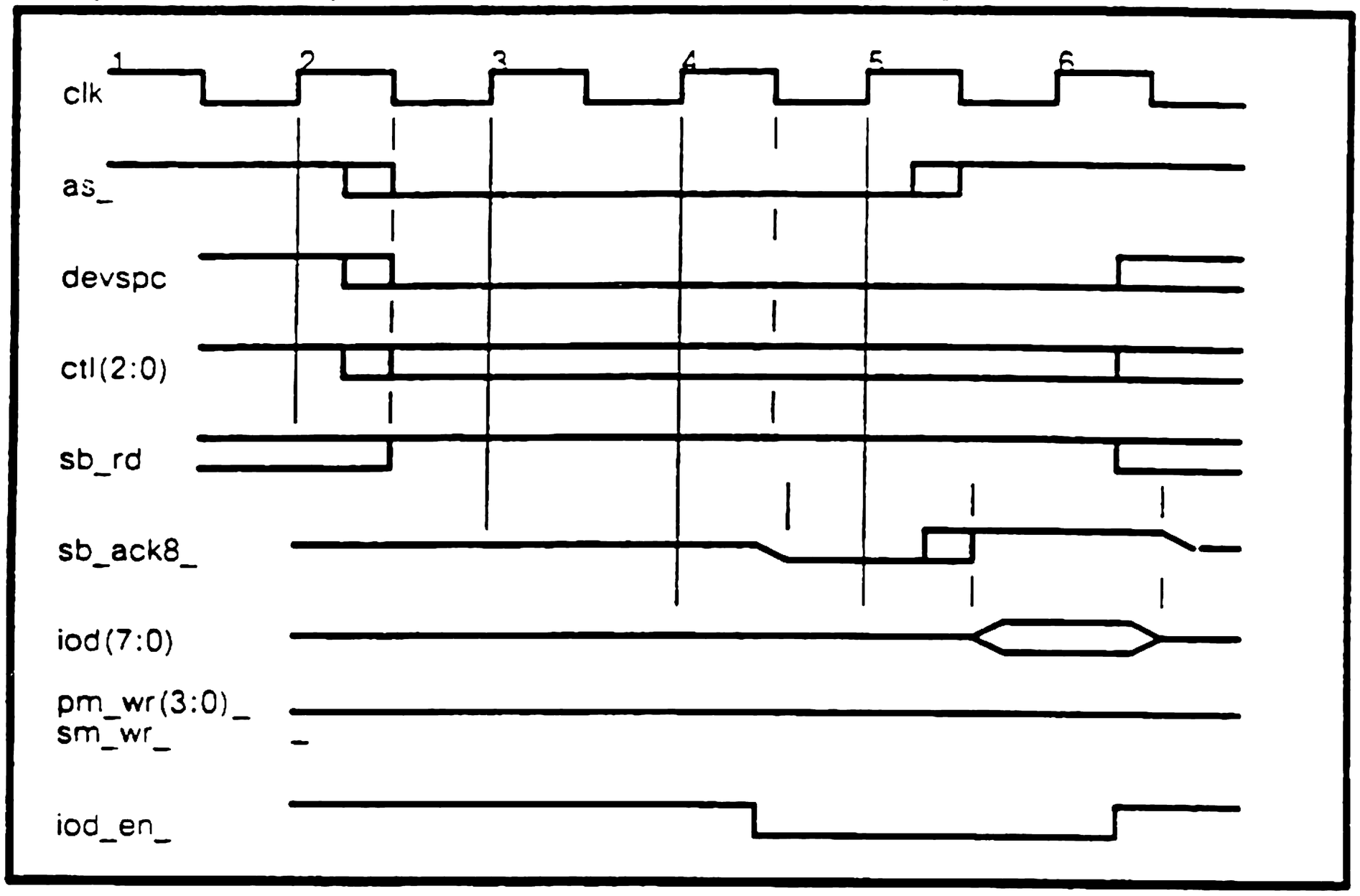
* Note that only one of these signals is asserted at a time.

Page Map

The page map is the second level of the two-level MMU, and contains 8k or 16k page map entries each mapping an 4 Kbyte page. It is indexed by the 7/8-bit PMEG provided by the segment map concatenated with virtual address bits SB_A(17:12). The page map bit definition is as follows:

Bit	Type	Description
31	V	valid bit, implies read access
30	W	write allowed protection bit
29	S	supervisor only protection bit
28	X	don't cache bit
27:26	MMU_TYP(1:0)	0 => main memory
		1 => input/output space
		2,3 => reserved for VMEbus
25	A	accessed (statistic bit)
24	M	modified (statistic bit)
23:16	none	reserved
15:0	page	physical page number

A Page Map read cycle is shown in the following timing diagram:



EPROM

The EPROM is addressable both in device space and control space. When implementing a Sun-4 architecture, this feature allows all supervisor instruction fetches in boot state to be forced to the EPROM. The system powers up in boot state and remains in that state until the appropriate bit in the System Enable Register is set.

Diagnostic Register

Any access to the Diagnostic Register will be ignored (with an ACK) by the S4-Cacne chip.

Device Space Address Map

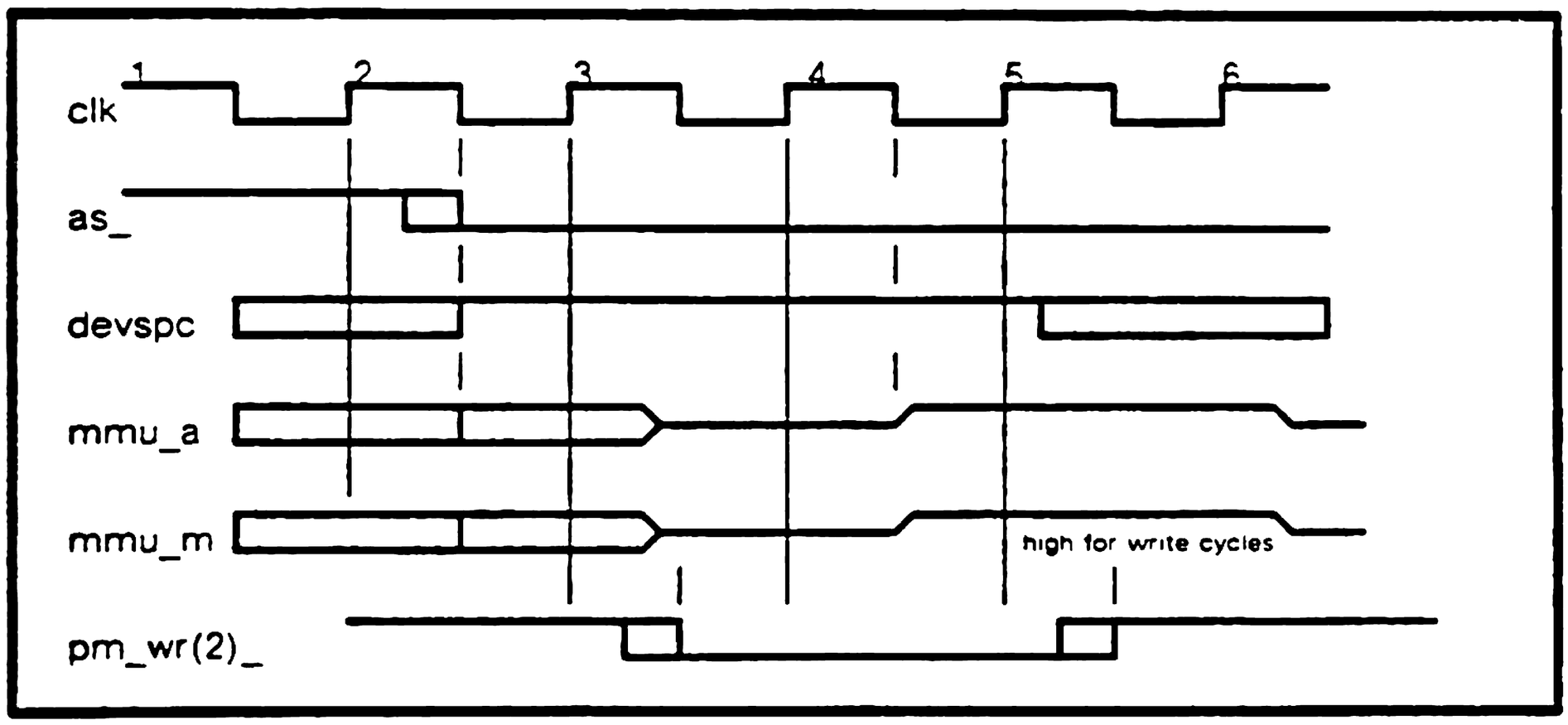
mmu_typ(1:0)	pa	device
0	[28 26] 0 X X	RAMSEL (main RAM)
1	[31:20] * F 0 X F 1 X F 2 X F 3 X F 4 X F 5 X F 6 X F 7 2 F 7 3 F 7 4 F 8 X F 9 X F A X F B X F C X F D X F E X F F X	Keyboard/Mouse Serial Controller Chip TOD Clk, NVRAM Counter Registers Parity Ctrl/Aux Interrupt Register EPROM Floppy Controller Audio DAC Aux Out Register SchoolBus Onboard Video Onboard SchoolBus Slot 1 SchoolBus Slot 2
2	all	Unused
3	all	Unused

Video Onboard

* PA[31:28] are not actually decoded, but assumed to be 1's on Type 1 accesses and 0's on Type 0 accesses

Main RAM-- Statistics Update Cycles

The operating system requires certain information about the read/write history of each page mapped into main memory. The S4-MMU chip maintains this information in the MMU_A and MMU_M bits, automatically updating them on any reads or writes of main memory. A statistics update cycle is shown below:



Because the PM_WR_ signals will be asserted in Cycle 3 and negated in Cycle 5, addresses *must* remain stable to the MMU RAMs throughout Cycle 5: the earliest they may change is Cycle 6. Statistics bits are tri-stated in Cycle 6 No data collision occurs because the addresses do not change; we are reading the data we wrote.

Main RAM

The RAMSEL_ signal is a combinatorial output for selecting valid Type0 space accesses. It must be qualified with AS_ at the RAM chip. It may be used as a Parity Enable signal on the S4-Buffer chip.

Protection Errors

An access violation may occur on any transaction. This is detected by comparing the USER_ signal with the MMU_S bit (user mode may not access Supervisor-only memory) and SB_RD with MMU_W (writes must be allowed). When AS_ is recognized in Cycle 2 along with any access violation, no Update cycle takes place (and, of course, no select is generated for any peripheral chip). It is assumed that the S4-Cache chip will also detect this violation, and terminate the access itself; no acknowledge is required. Ctl(0) must be zero on any device space transaction, and ctl(1) high requires that the page be writeable, even if the current bus cycle is a read operation (it implies a LDST instruction, so a write *will* be coming).

SBus Onboard

The first Ethernet and SCSI controllers are standard on the CPU board, but are viewed as SBus devices in "Slot 0". The Video Control Registers, RAMDAC, and Frame Buffer are viewed as "Slot 1." *All SBus select signals are strictly combinatorial; the destination chips must qualify them with AS_.*

SBus Slots

These slots are for system expansion boards. The SB_SEL(3:2)_ signals used to read and write these boards in slave mode.

Keyboard/Mouse & Serial Controller Chips

The read and write strobes for these devices implement timing for the Z8530 Serial Controller Chip from AMD and Zilog. On SB_RESET_, both the read and write strobes are asserted to initialize the SCCs.

Note: The SCC chips require a minimum quiescent time between accesses to allow the on-chip synchronization circuitry to recover. This recovery time is guaranteed by the S4-MMU chip; *any accesses to an SCC within 34 clocks of a previous SCC access will be held off until the recovery period expires*. The recovery timer does not differentiate between the two SCC's. (34 clocks is 1.36 uSec at 25 MHz, 1.7 uSec at 20 MHz.)

Time-of-Day Clock

Timing is implemented for a Mostek MK48T12-15 Zeropower/Timekeeper RAM which also includes 2 KBytes of CMOS RAM. Please see the Mostek data sheet for a description of internal registers. The TOD_CS_ signal is asynchronously held high while the SB_RESET_ input is held asserted.

Parity Control Register

The Parity Control Register is located in the S4-Buffer chip.

Interrupt Register

The Interrupt Register provides for software generation of interrupts and allows the CPU to disable all interrupts or only certain ones. It is cleared on SB_RESET_, and has the following fields:

31	30	29	28	27	26	25	24
Enable Level 14 Interrupts	Reserved	Enable Level 10 Interrupts	Enable Level 8 Interrupts	Software Interrupt Level 6	Software Interrupt Level 4	Software Interrupt Level 1	Enable Interrupts Clears Level 15 when 0

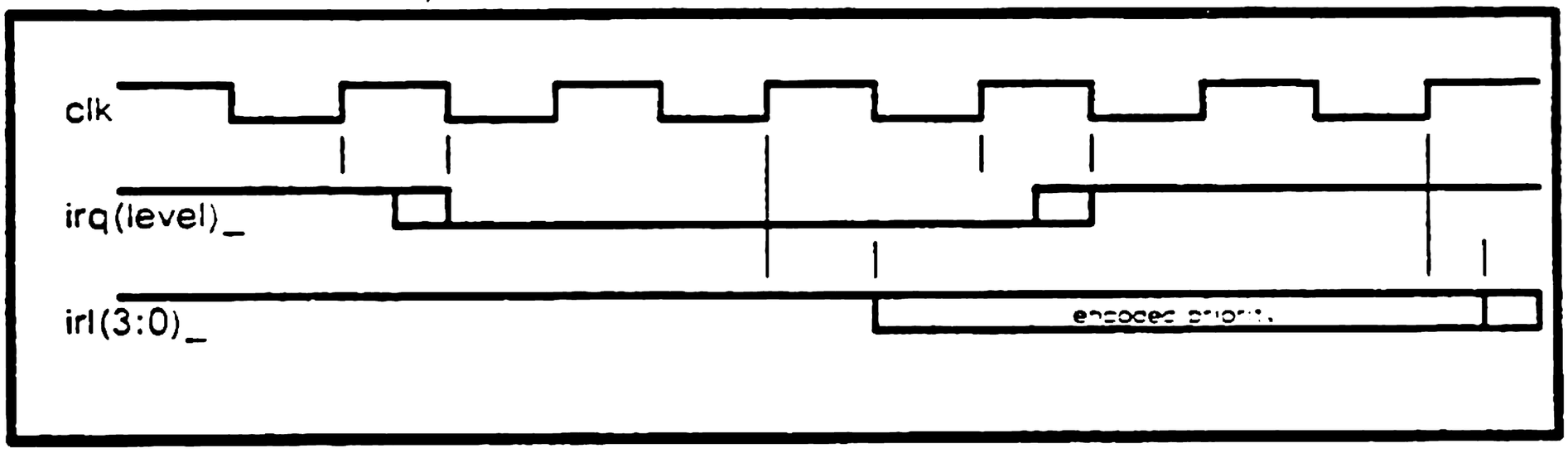
All IRQ(13:1)_ signals may be asynchronous to the system clock.

Software interrupts may be generated on levels 6, 4, and 1 by writing a 1 into bits 27, 26, or 25 when interrupts are enabled (bit 24 high).

Level 15 Interrupt requests are captured on a clock edge and held asserted to the CPU until bit 0 of the Interrupt Register is cleared.

Note that writing a zero to the Enable bits in the Interrupt Register only masks out that level's interrupt *it does not clear the source* (with the exception of Level 15 requests). This is different from the Sun-4 Architecture, in that the periodic interrupts at Levels 10 and 14 must be cleared by accessing their respective Limit registers.

Level-Sensitive Interrupts:



Interrupting Devices (assumed system configuration):

Int Level	Device
15	Buffered Write Timeout Error
14	Clock Interrupt 14 from Counter 1
13	Bus IRQ13
12	Keyboard/Mouse Serial Ports
11	Bus IRQ11 Floppy
10	Clock Interrupt 10 from Counter 0
9	Bus IRQ9
8	Video
7	Bus IRQ7
6	SWIRQ6 Ethernet
5	Bus IRQ5
4	SWIRQ SCSI DMA
3	Bus IRQ3
2	Unused
1	SWIRQ1 Bus IRQ1

EPROM

A decode is included for an arbitrarily deep bytewide EPROM for storing power-up diagnostics and standard boot routines. The EPROM select is generated from physical addresses, but the EPROM itself is connected to the virtual address bus so it is accessible without setting up the MMU after power-up. This necessitates special mapping in the MMU once out of Boot State (See System Enable Register in S4-Cache chip).

Floppy Disk Controller
The FDC is an Intel 82072.

IOSEL

The IOSEL_ signal is simply an address decode that is intended to go to an external PAL. It is asserted on any Device Space cycle which passes the protection checks. *External logic should decode AS_, MMU_TYPE and addresses.* IOSEL_ decodes should not overlap those decodes for which the MMU generates automatic acknowledges or delays. **IOSEL_ is an asynchronous signal.**

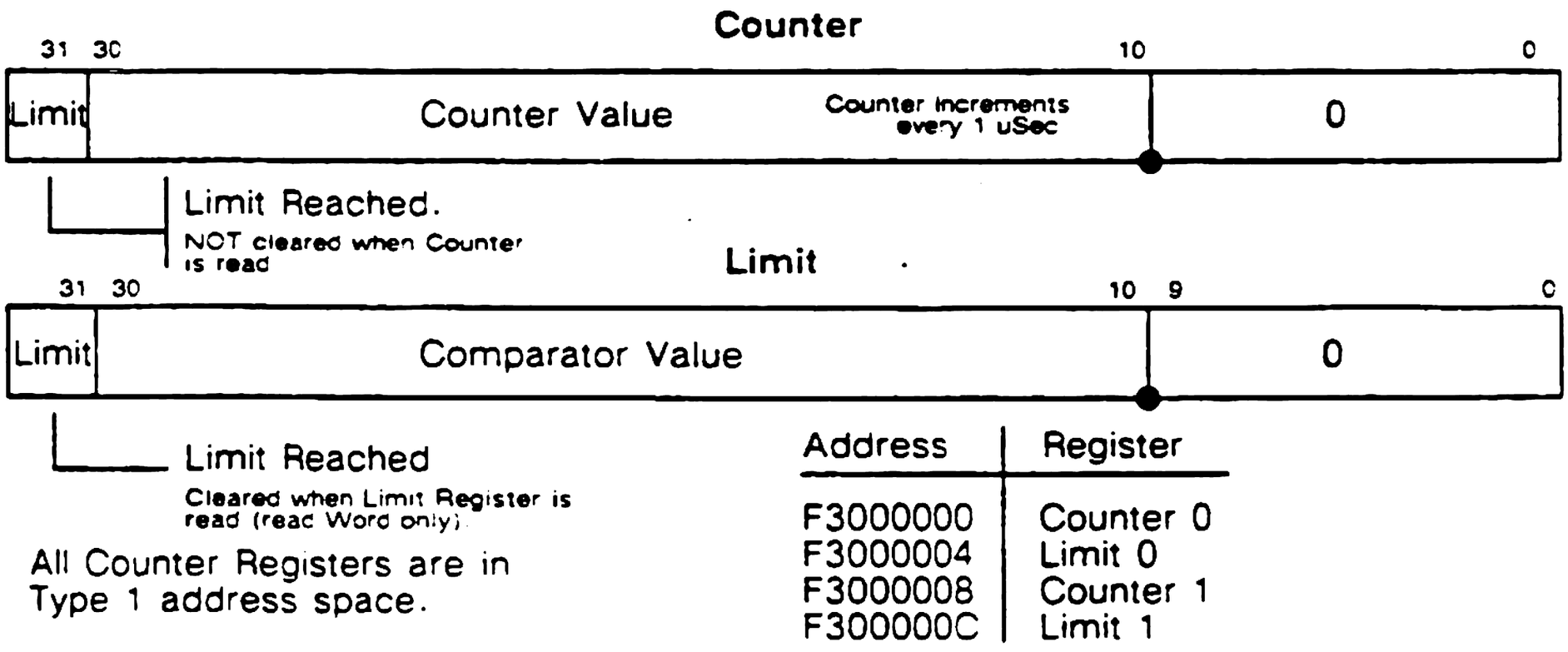
IODEN_

The IODEN_ signal is asserted for all 8-bit accesses decoded by the MMU, and for all Control Space accesses (to include the Cache chip).

Counter/Timer Registers

A 21-bit counter incrementing at 1 uSec is accessible in Type 1 Device space. It is set to "1.0 uSec" on SB_RESET_ and free-runs. The least significant 10 bits of this counter always read as zeroes to allow for future expansion.

A 21-bit Limit register is compared against the counter value each increment. If the Counter and Comparator bits match, the counter is set back to "1.0" on the next increment. As the counter is set an interrupt occurs at level 10 or 14 for Counter 0 or 1 if the corresponding IE bit of the Interrupt Enable register is set. Whenever a Counter is reset by exceeding the Limit register, the Limit bits (31) of the corresponding register pair are set. This happens regardless of the state of the Interrupt Enable bit. These bits are cleared when the Limit register is read, also clearing the interrupt. The Comparator Value in the Limit register is set to all 0's on Reset, allowing the Counter to freerun.



All Counter Registers are in Type 1 address space.

Address	Register
F3000000	Counter 0
F3000004	Limit 0
F3000008	Counter 1
F300000C	Limit 1

Writing to the Limit register will set its corresponding Counter back to 1.0 uSec. Both Counters may be read at any time. (Counter increments are internally held off until the read completes, but no increments will be missed.)

Both Counters are separately writeable for testing purposes. They should not be written in normal operation. Because of the 8-bit interface unpredictable carrys could occur.

Auxiliary Output Registers

An additional read/write strobe has been added for a set of Auxiliary Output Registers located in Type 1 Device space beginning at F7400000.

DAC Write and Transfer strobes

The DAC_WR_ and DAC_XFER_ signals are somewhat overloaded. In the power-up mode, they are used to access an external double-buffered DAC. The DAC_WR_ signal is asserted when the cpu attempts to write to the audio DAC address range. It is a slow device, inserting 7 waitstates, like the SCC's. The DAC_XFER_ signal is asserted when counter 1 hits its limit register value, transferring the holding register data into the DAC internal register. It is asserted for 6 clocks or until the interrupt source (Limit 1) is removed, whichever comes first.

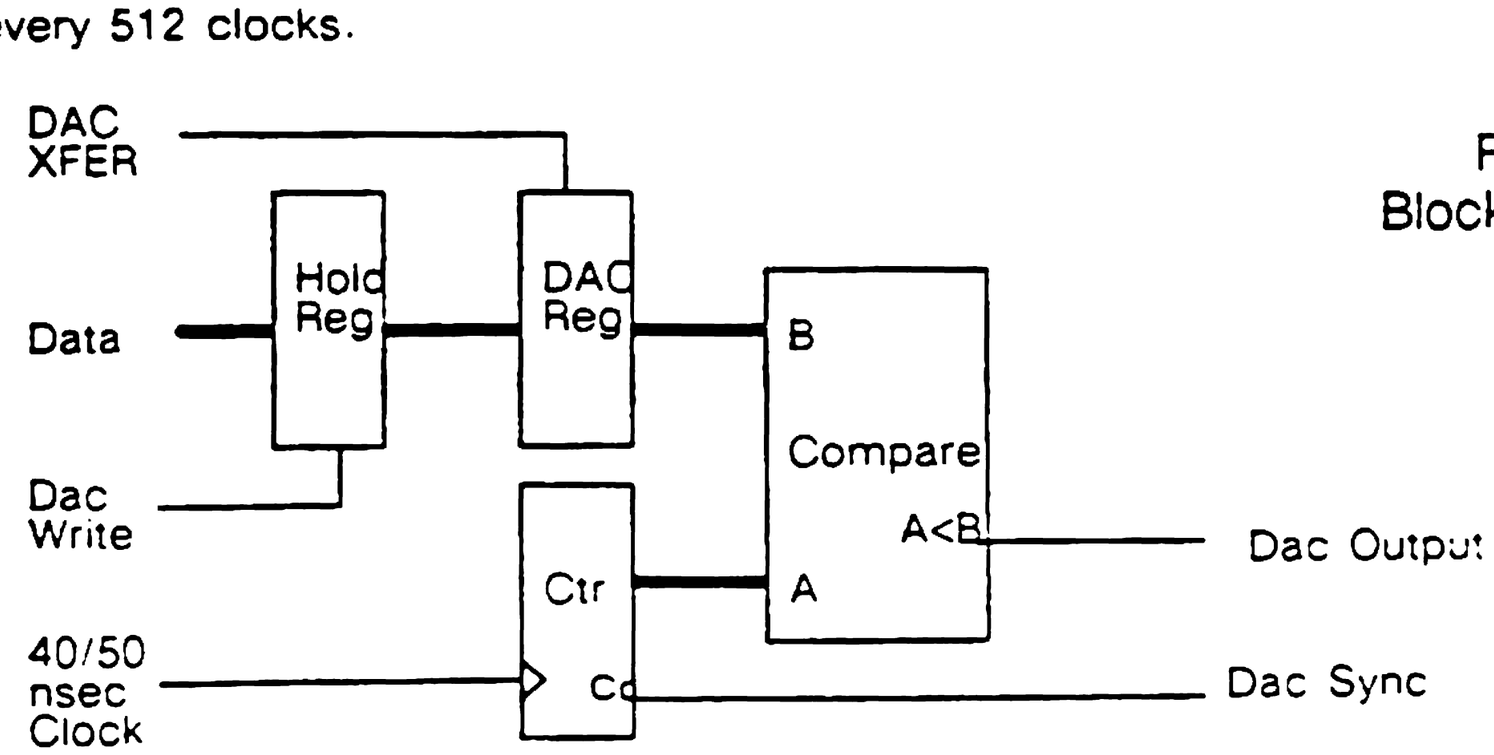
When the Internal DAC is enabled (see below) the DAC_WR_ pin becomes the DAC2 output. The DAC_XFER_ pin becomes the PWM output, varying in duty-cycle between 0-511 CLKs out of 512.

In addition, the DAC_WR_ signal is asserted for both reads and writes at location 0xF7FXXXXX. This is used as an S4-VME chip select signal.

Internal PWM DAC

Two 8-bit Pulse-Width Modulation DACs are implemented, operating off of the 40/50 nSec CLK input. When enabled, this DAC outputs replace the DAC_WR_ and DAC_XFER_ output pins. It responds to the same address space as the external DAC, only faster: Type 1 Device Space, $F7300000.

The output of the PWM DAC is a square wave with a duty cycle between 0 and just under 100%. When the DAC data register is programmed with 0's, the output is never high. When it is programmed with $0080 (least-significant bit of 9-bit DAC set), the output is high for one clock every 512. When it is programmed with $FF80 the output is high 511 out of every 512 clocks.



PWM DAC
Block Diagram

To use the internal DAC, the Internal DAC Enable bit at F7300002, bit 0 must be set. The Counter Test Mode, AEB, AGB, ALB, and Counter Bits are meant to be read for chip testing only. The Counter Test Mode bit should never be set to one; it speeds the two Counter/Timers up to the IU Clock frequency and is intended for chip testing only.

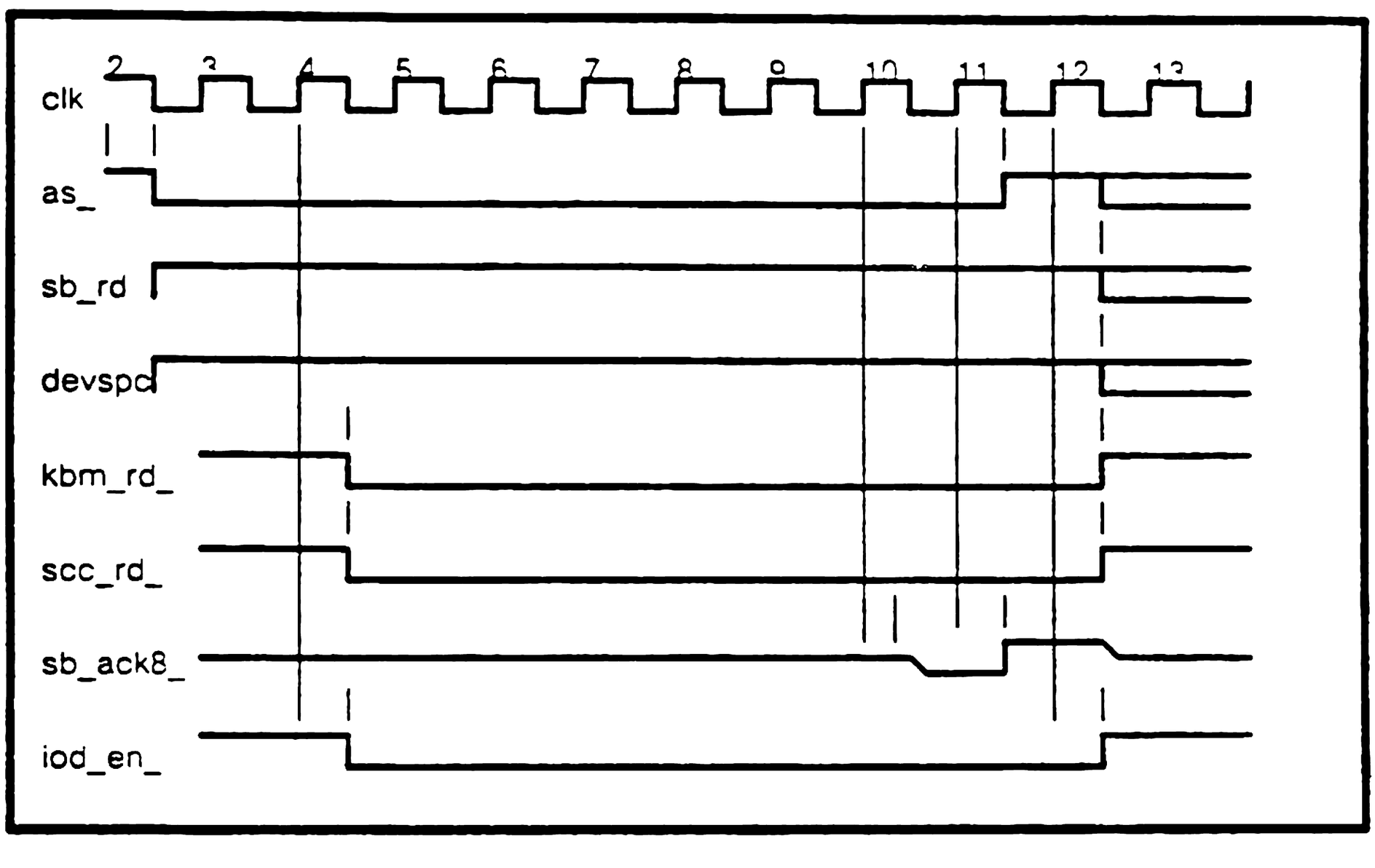
Values read back from locations F7300000 and F7300001 are *not* the same as the values just written. You write to the Hold Register and read from the DAC Register. The Data in the Hold Register is transferred to the DAC Register when the internal DAC_XFER pulse is asserted (when Counter 1 reaches its limit).

All accesses to the Internal DAC registers are fast after the Internal DAC Enable bit is set. No reads from internal DAC registers may be made until this bit is set.
32-bit values should **not** be written to the DAC location as the upper two bytes are used for control bits, as shown below:

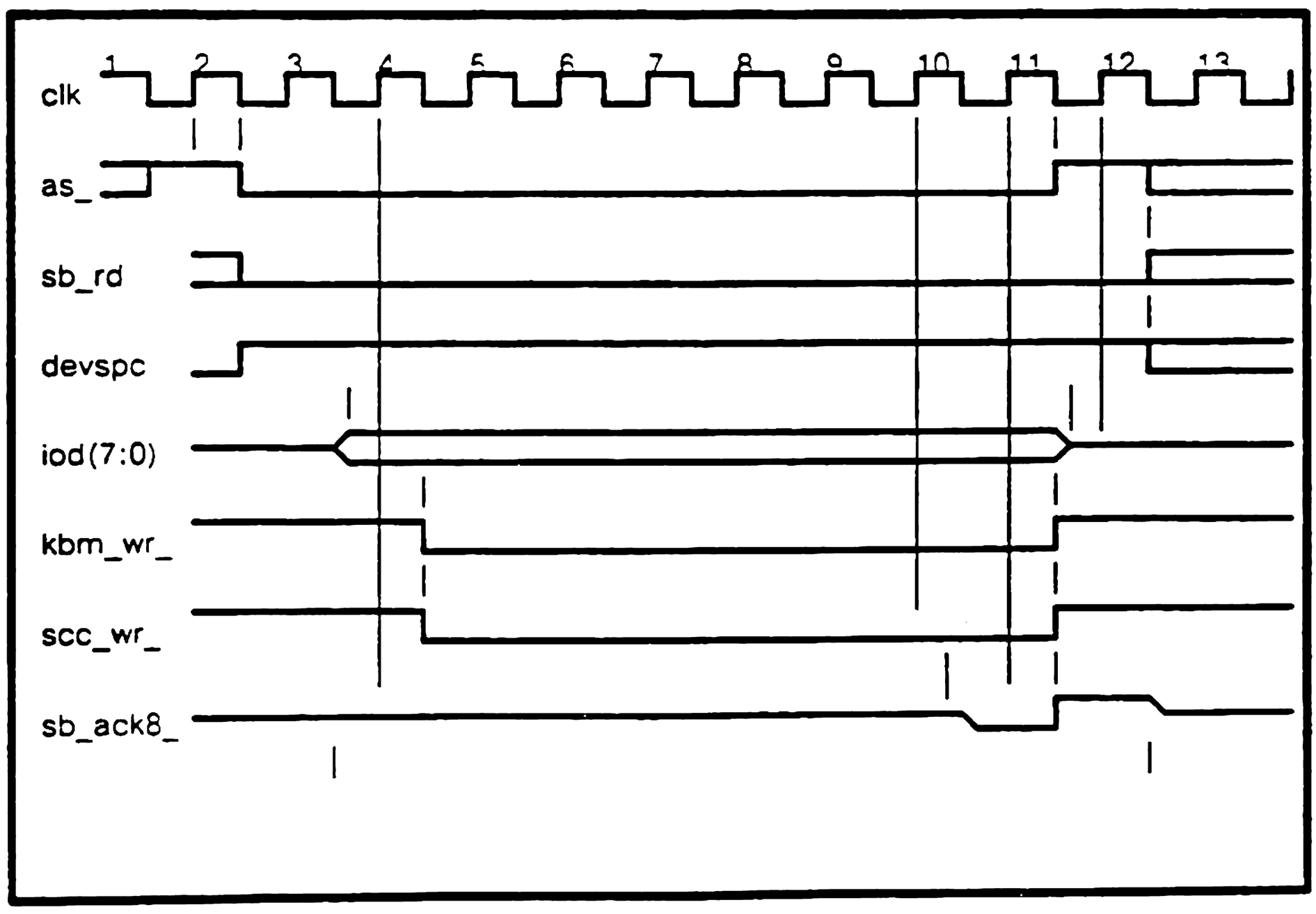
A[1:0]		7							0
0	W	Hold Register Bits [7:0]							
	R	DAC Register Bits [7:0]							
1	W	Hold Register Bits [7:0] (DAC2)							
	R	DAC Register Bits [7:0] (DAC2)							
2	W	0	0	0	0	0	0	Counter Test Mode	Internal DAC Enable
	R	0	0	0	0	0	0	Counter Test Mode	Internal DAC Enable
3	R	Counter Bits [7:0] (test only)							

Functional Timing Diagrams

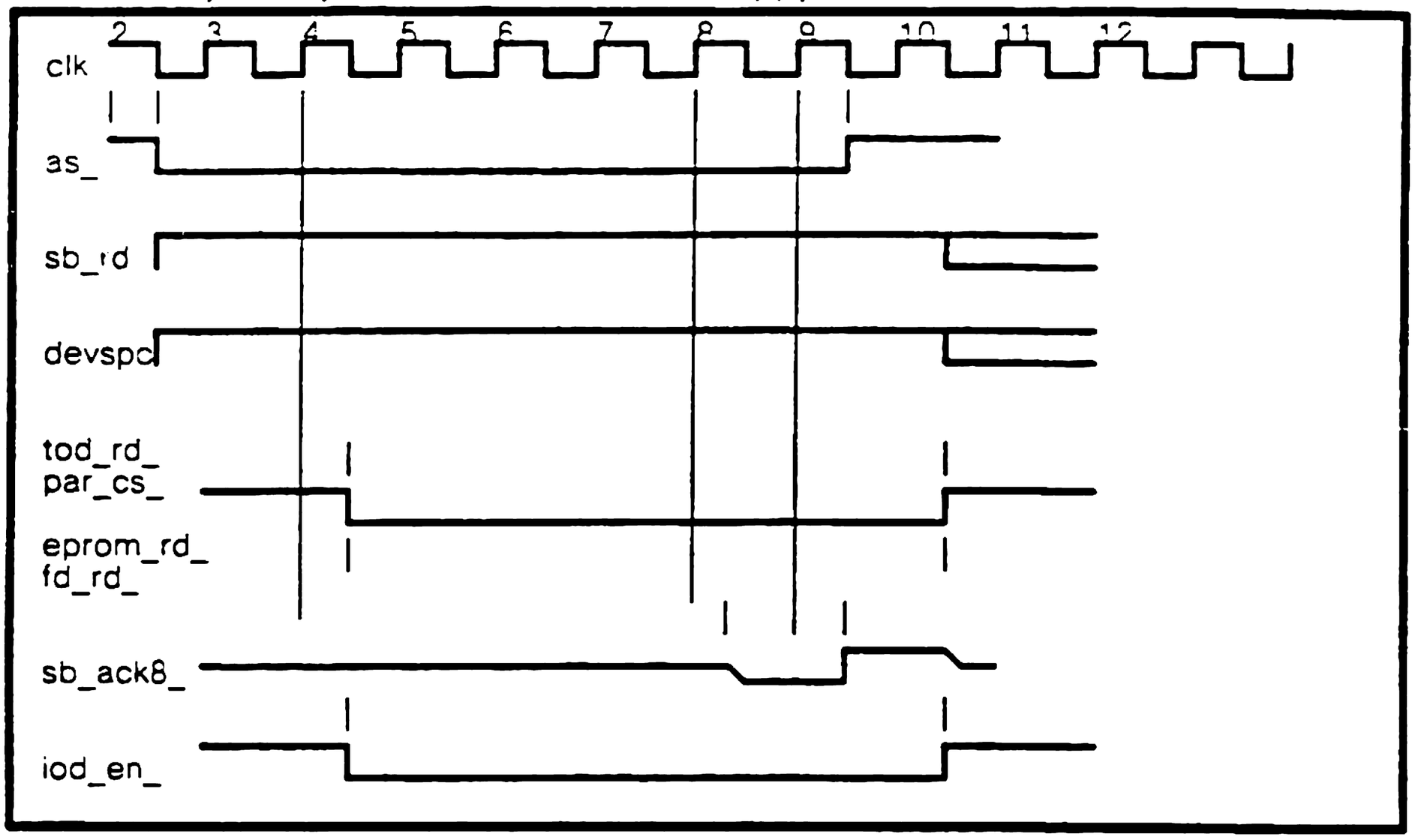
Keyboard/Mouse or SCC Read



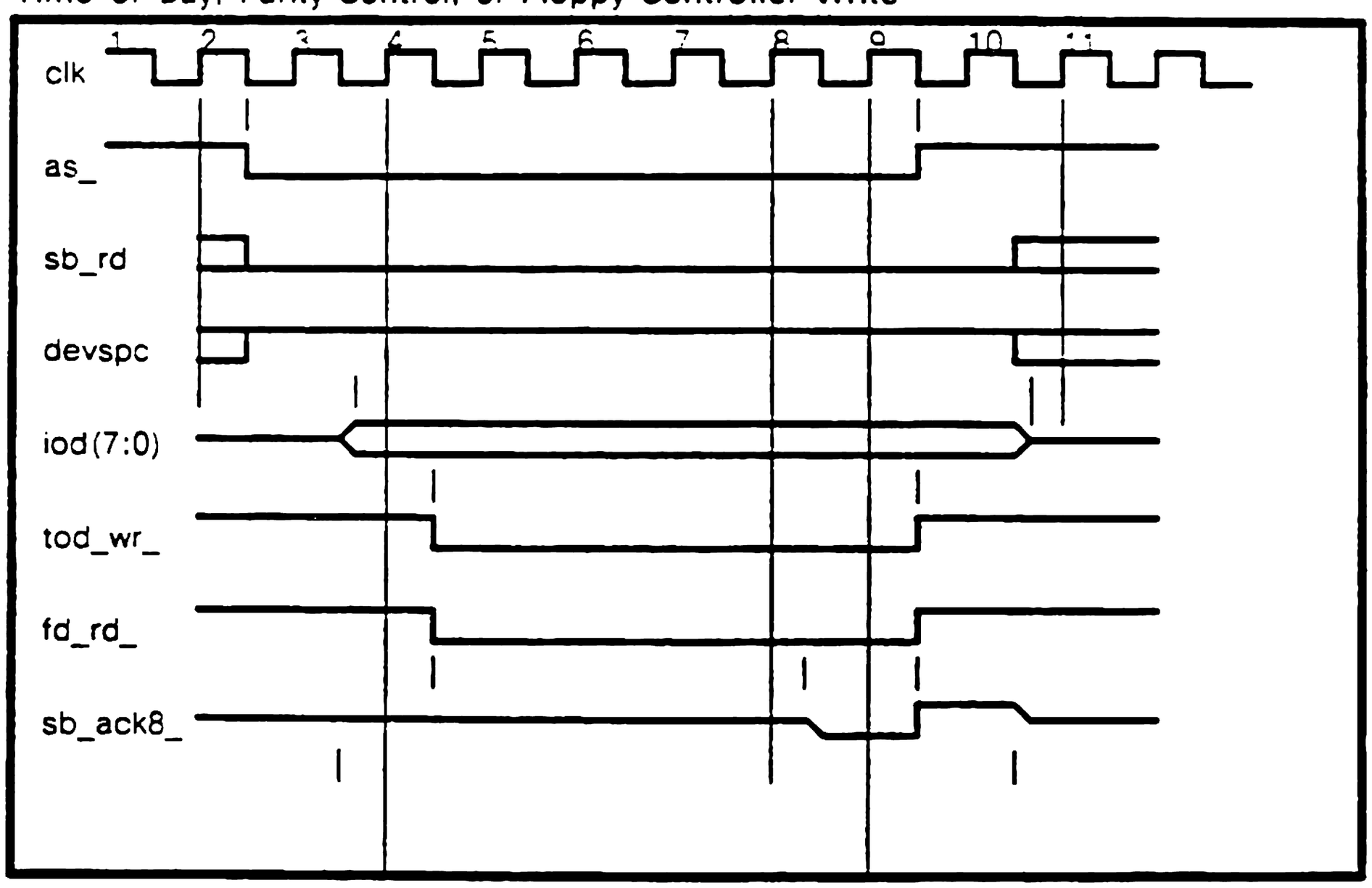
Keyboard/Mouse, or SCC Write



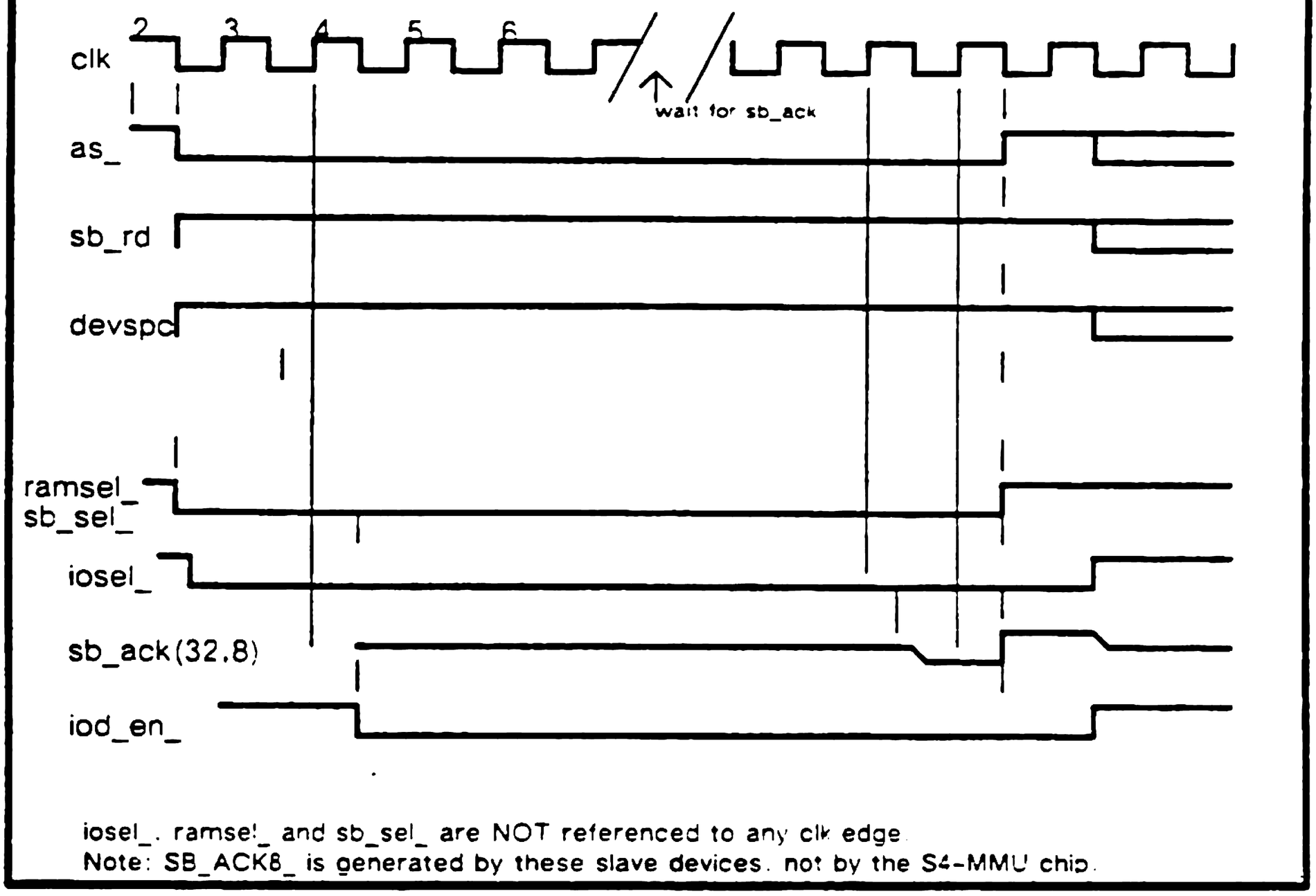
Time-of-Day, Parity Control, EPROM, or Floppy Controller Read



Time-of-Day, Parity Control, or Floppy Controller Write

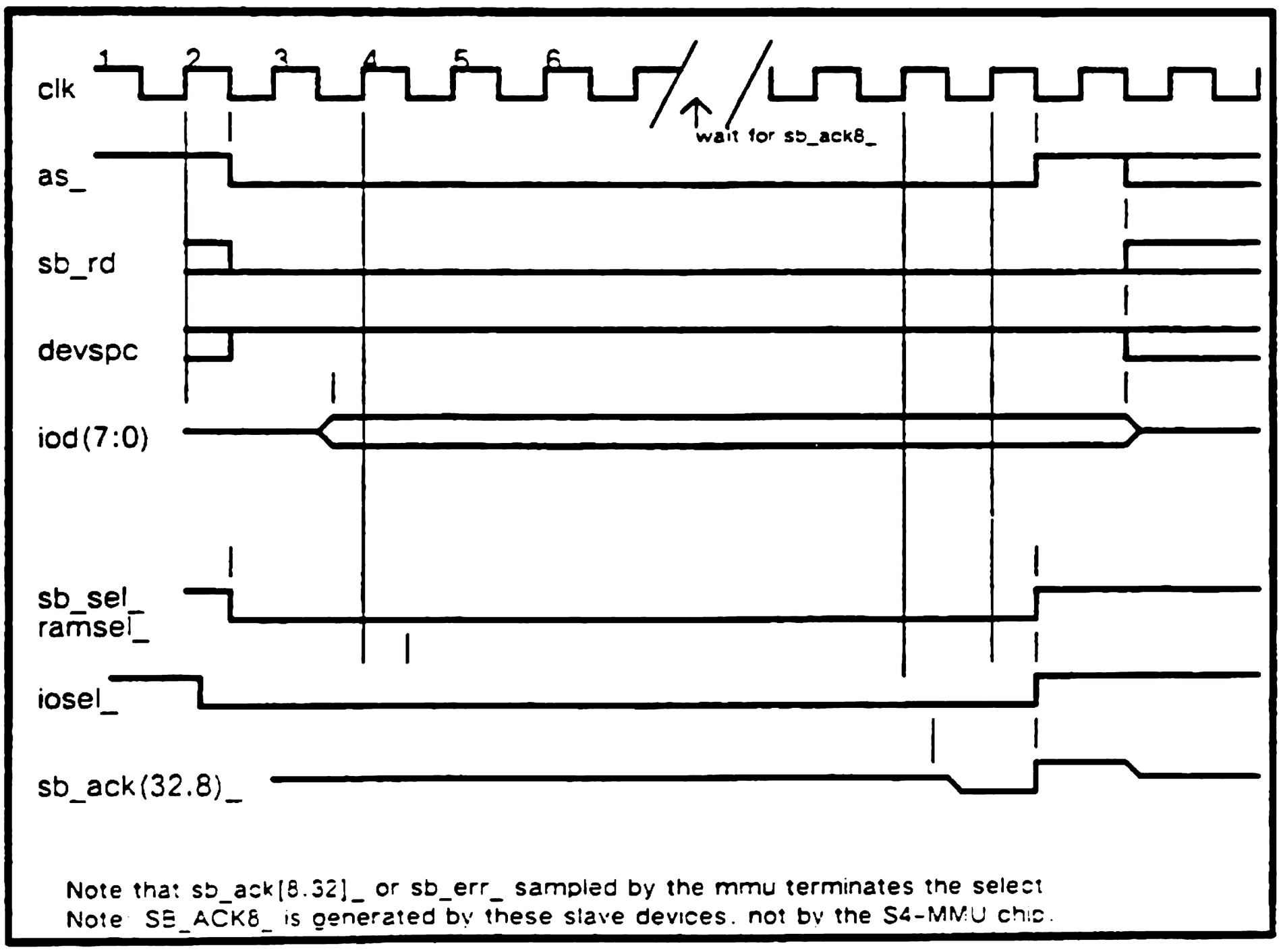


SBus, RAM, or IOSEL Read



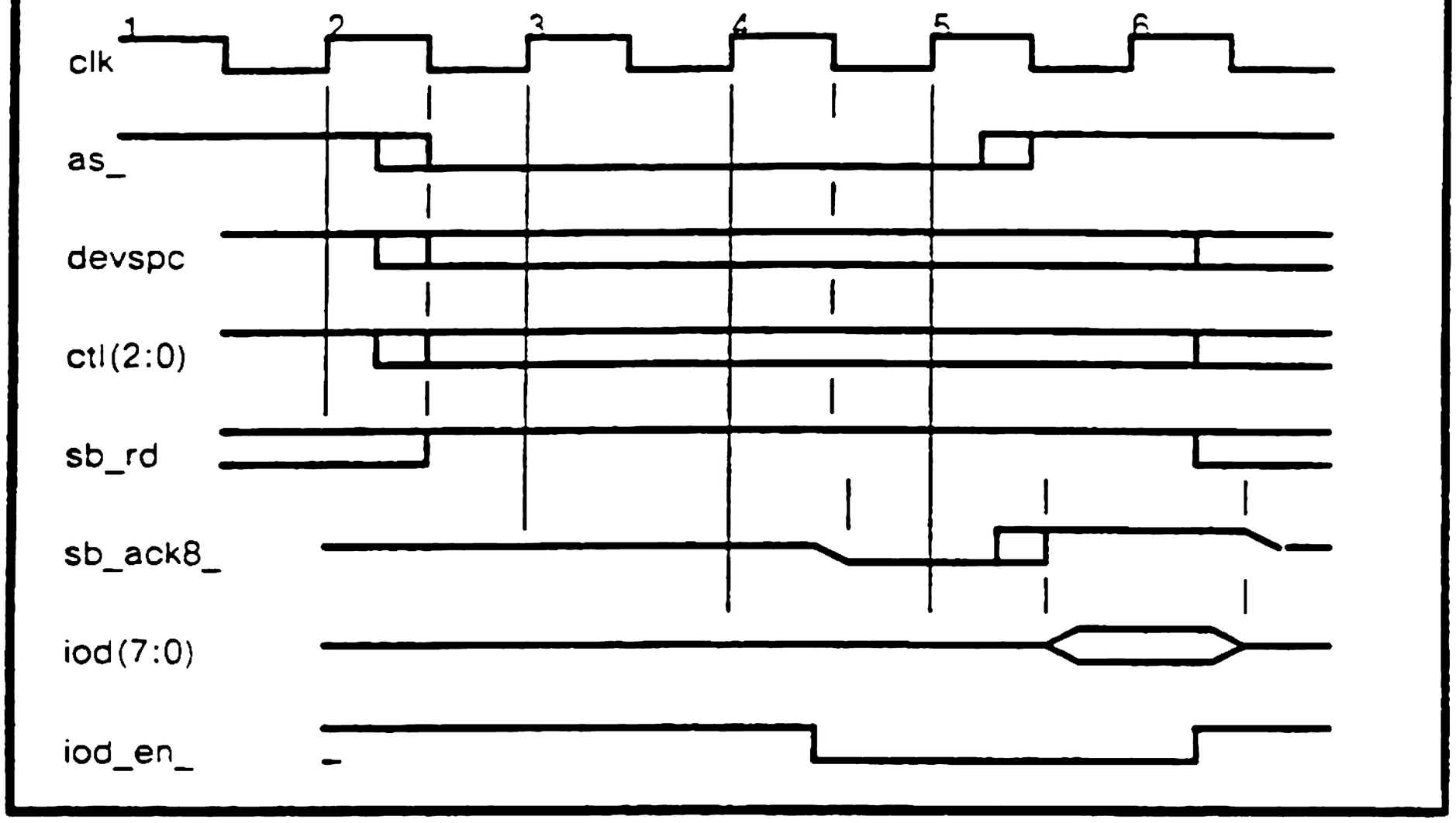
iosel_, ramsel_ and sb_sel_ are NOT referenced to any clk edge.
Note: SB_ACK8_ is generated by these slave devices, not by the S4-MMU chip.

SBus, RAM, or IOSEL Write

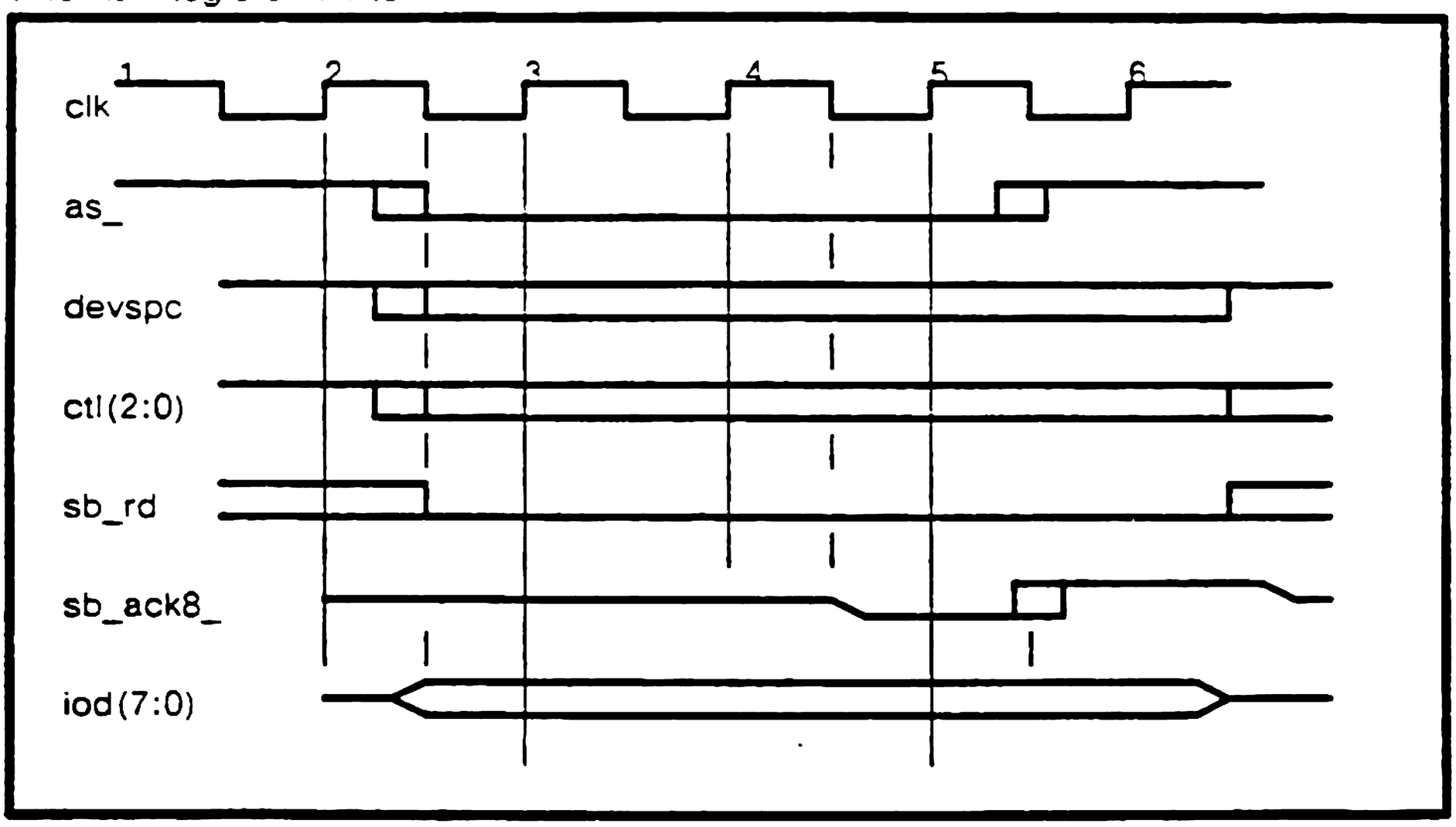


Note that sb_ack[8.32]_ or sb_err_ sampled by the mmu terminates the select
Note: SB_ACK8_ is generated by these slave devices, not by the S4-MMU chip.

Internal Register Read

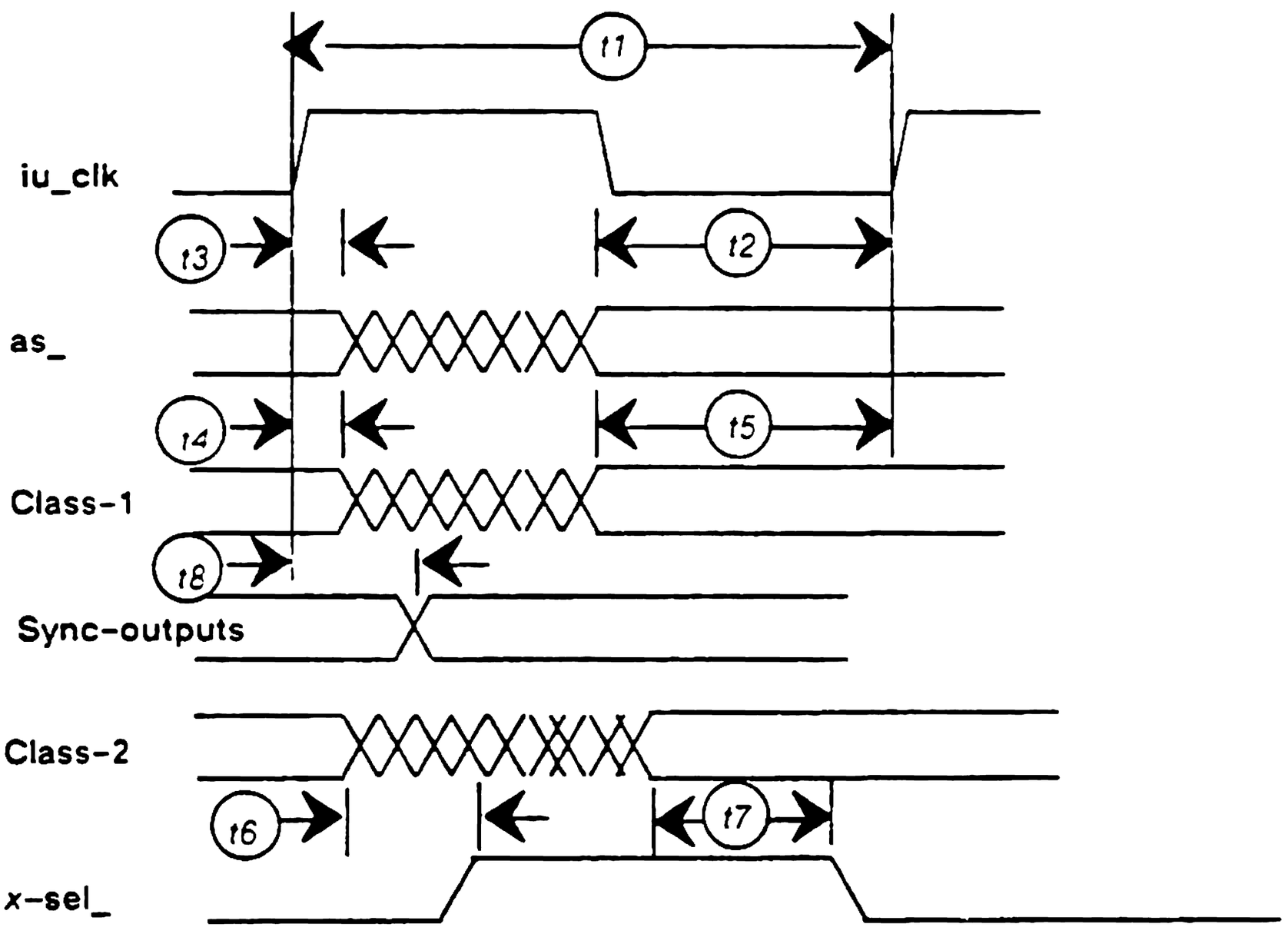


Internal Register Write



Timing Specifications and Diagrams

Tn	Description	min	max
t1	iu_clk cycle time	40	
t2	Setup time, as_ signals before clk	3	
t3	Hold time, as_ signals after clk	15	
t4	Hold time, Class-1 signals after clk	0	
t5	Setup time, Class-1 signals before clk	15	
t6	Delay Class-2 to *x*-sel_ negated		22
t7	Delay Class-2 to *x*-sel_ asserted		23
t8	Synchronous output delay		22



Class-1 signals are: io_a[3:0], ctl[2:0], devspc_, sb_rd, user_, pmeg[7:0] (in), pa[27:12] (in), mmu_[vwsxam] (in), mmu_typ[1:0] (in). *These signals are used synchronously in this case.*

Class-2 signals are: pa[27:12] (in), mmu_[vwsxam] (in), mmu_typ[1:0] (in), ctl[1:0], devspc_, sb_rd, user_. *These signals are used asynchronously in this case, affecting outputs sb_sel[3:0]_ and ramsel_.*

Capacitive Loading

Pin Name	Driver	Load Assumed (pf)
iu_clk	DRVT4	*
const_clk	TLCHT	*
io_a(3:0)	TLCHT	*
ctl(2:0)	TLCHT	*
devspc_	TLCHT	*
sb_rd	TLCHTU	*
as_	TLCHT	*
sb_reset_	TLCHN	*
iod(7:0)	BD4TU	100
sb_ack8_	BT4	100
user_	TLCHN	*
dma_	TLCHT	*
cid(3:0)	BT2	15
pmeg(7:0)	BD4TD	15
pa(27:12)	BD4T	41
mmu(v,w,s,x)	BD4T	15
mmu_typ(1:0)	BD4T	15
mmu(a,m)	BD4T	15
sm_wr_	BT4	15
pm_wr(2:0)_	BT4	15
ramsel_	BT2	10
kbm_rd_	BT2	15
kbm_wr_	BT2	15
scc_rd_	BT2	15
scc_wr_	BT2	15
tod_cs_	BT2	15
par_cs_	BT2	10
eprom_rd_	BT2	15
fd_rd_	BT2	15
fd_wr_	BT2	15
dac_wr_	BT2	15
dac_xfer_	BT2	15
iosel_	BT2	15
sb_sel(3:0)_	BT2	15
iod_en_	BT4	10
aux_wr_	BT4	15
irq(15,13:11,9:1)_	SCHMITCN	*
iu_irl(3:0)	BT2	10
od_,	TLCHTU	*
para	BT1	10

* input only.

Conditions: VCC=4.75 to 5.25V, TA=0 to +70C, Output Load=15 pF

Symbol	From	To	min	max	unit	note

t1	clk cycle	40	---	ns	
t2	as_ setup to clk	15			
t3	as_ hold from clk	2			1

Notes:

1. This timinig specification does not meet the ideal requirements for 25 MHz system operation.

- NOTE: IO_DEN_ is asserted only on READS. It is assumed that all write cycles drive the iod bus.

Change History

12/15 tw Config register is gone.
Counter/Timer is 30 bits. Added Interrupt_Occurred bit.
Diag register and bit added.

12/17 tw Added sb_ack32_, made sb_ack8 and sb_err BD4's.
Statistics updates tristate in Cycle 6.

12/18 tw Modified Counter/Timer to freerun on reset.
Moved DACWR, Ctr, Limit, Floppy to E01-4.

12/18 tw Two Counter/Limit register sets, dedicated at Int levels 10 and 14.
Deleted IRQ inputs 10 and 14.
Deleted PARA output, multiplexed with od_ input in test mode.
Diag is now a BT8.
Added one more SB_SEL_ signal, deleted vctl_cs and vramsel.
Deleted DMA Starvation timeout, deleted SB_BG pins.
Added A2 and 3, gathered Counters and Limit registers in one page.

12/21 tw DAC_WR Gone. It's now in the Video chip.
Counter starts at 1.

12/22 tw ramsel, vramsel (sbsel1) are now combinatorial.
All inputs are ttl levels.

12/29 tw PAR_EN_ signal removed. S4-Buffer will use RAMSEL_ instead.

1/5/88 tw RAMSEL is now *all* of Type1 Device Space.

1/14/88 tw DIAG changed to AUX_WR_. IOSEL changed slightly.

1/21 tw Added Limit bit to Counter. Moved Counter to EF. Moved SB Slots to Type 0 Space.

1/27 tw TOD is now just a CS_. Added DAC_XFER to allow for double-buffered DAC.

1/29 tw SB_SELn_ are now all asynchronous.

2/8 tw Removed SB_ACK32_ and SB_ERR_.

2/23 tw Added DMA_ pin and description.

2/29 tw Fixed mmu ram write pulse in Pg 4 diagram.

3/8 tw Added internal pwm dac. IODEN_ documented.

3/15 tw 4k pages. Changed memory map.

4/7 tw iosel_ asynchronous. VME select address removed due to lack of use.

4/18 tw Level 15 interrupts captured and held. Cleared by turning off all interrupts.

4/19 tw Changed Device Address Map to remove reference to onboard video.

4/26 tw Video is back. Ignore previous change.

4/27 tw Documented cclk divide by ten and added ctl0 function to invalidate device-space accesses.

6/21 tw Documented ctl1 function (implied LDST).

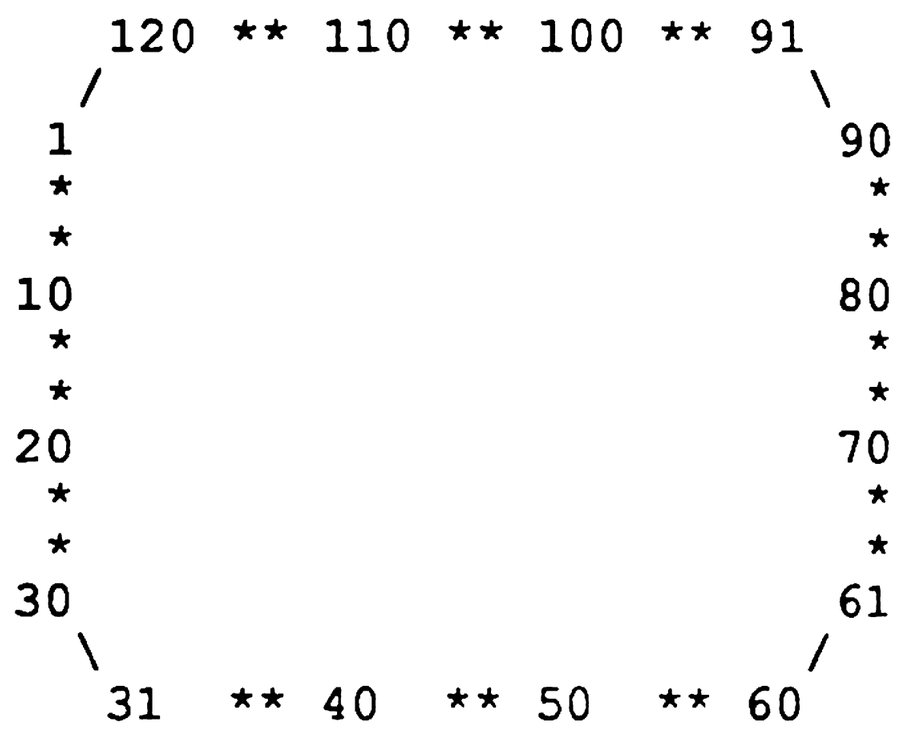
7/19 tw Added timings.

1. P/N: 100-1802-02 2. REV: 50

3. DESCRIPTION: IC,GA,S4-MMU

4. PHYSICAL MODEL: (120 Pin PFP)

A. Pinout: *Top View



1 - GND	31 - GND	61 - VCC	91 - PA.12
2 - /IU IRL.3	32 - IRQ15	62 - CTL.0	92 - PA.13
3 - /IU IRL.2	33 - IRQ12	63 - CTL.1	93 - PA.14
4 - /IU IRL.1	34 - IRQ11	64 - CTL.2	94 - PA.15
5 - /IU IRL.0	35 - CCLK	65 - NC59	95 - PA.16
6 - /DAC WR	36 - DEVSPC	66 - /MMU M	96 - PA.17
7 - XFER	37 - USER	67 - /MMU A	97 - PA.18
8 - VCC	38 - /KBM WR	68 - /MMU TYP.0	98 - PA.19
9 - AS	39 - /KBM RD	69 - /MMU TYP.1	99 - GND
10 - /SB SEL.0	40 - /SCC WR	70 - /MMU X	100 - VCC
11 - GND	41 - /SCC WR	71 - /MMU S	101 - PA.20
12 - /SB RD	42 - /FD WR	72 - /MMU W	102 - PA.21
13 - RESET	43 - /FD RD	73 - /MMU V	103 - PA.22
14 - /SB CAK8	44 - /IU CLK	74 - /PM WR.0	104 - PA.23
15 - /IO A.3	45 - GND	75 - /PM WR.1	105 - PA.24
16 - /IO A.2	46 - VCC	76 - /PM WR.2	106 - PA.25
17 - /IO A.1	47 - IOD.7	77 - VCC	107 - PA.26
18 - /IO A.0	48 - IOD.6	78 - GND	108 - PA.27
19 - IRQ4	49 - IOD.5	79 - /SM WR	109 - RAMSEL
20 - IRQ6	50 - IOD.4	80 - PMEG.0	110 - /SB SEL.1
21 - IRQ8	51 - IOD.3	81 - PMEG.1	111 - /SB SEL.2
22 - GND	52 - IOD.2	82 - PMEG.2	112 - GND
23 - VCC	53 - IOD.1	83 - PMEG.3	113 - /SB SEL.3
24 - NC58	54 - IOD.0	84 - PMEG.4	114 - IRQ1
25 - /PAR CS	55 - PROMRD	85 - PMEG.5	115 - IRQ2
26 - PMEG.7	56 - /TOD WR	86 - PMEG.6	116 - IRQ3
27 - CID.3	57 - IOSEL	87 - CID.0	117 - IRQ5
28 - PARA	58 - GND	88 - CID.1	118 - IRQ7
29 - OD	59 - /AUX WR	89 - CID.2	119 - IRQ9
30 - DMA	60 - /IOD EN	90 - GND	120 - IRQ13

B. PIN DESCRIPTION:

SYMBOL	TYPE	DESCRIPTION
/IU CLK	DRVT4	Integer Unit Clock
/CONST CLK	TLCHT	100 nsec clock divided by ten internally for counters.
/IO A(3:0)	TLCHT	Buffered Address bits from SBus
CTL(2:0)	TLCHT	Encoded control space selects
/DEVSPC	TLCHT	Lo for device space accesses
/SB RD	TLCHTU	SBus Read. High during IU reads.
/AS	TLCHT	Address strobe from S4-Cache
/SB /RESET	TLCHN	SBus Reset. Lo for power-on & S/W resets.
IOD(7:0)	BD4TU	Input/Output Data bus
/SB /ACK8	BT4	SBus 8-bit Acknowledge
/USER	TLCHN	Lo for User Space accesses
/DMA	TLCHT	Lo during DVMA accesses
CID(3:0)	BT2	Context ID. Hi-order addresses into the Segment Map.
PMEG(7:0)	BD4TD	Page Map Entry Groups
PA(27:12)	BD4T	Physical Address bits
MMU(v,w,s,x)	BD4T	MMU Valid, Write-allowed, Supv.-only, Don't Cache
/MMU TYP(1:0)	BD4T	MMU Type bits
MMU(a,m)	BD4T	MMU Accessed, Modified.
/SM /WR	BT4	Segment Map Write
/PM /WR(2:0)	BT4	Page Map Write bits
/RAMSEL	BT2	DRAM Select
/KBM /RD	BT2	Keyboard/Mouse Read
/KBM /WR	BT2	Keyboard/Mouse Write
/SCC /RD	BT2	Serial Controller Chip Read
/SCC /WR	BT2	Serial Controller Chip Write
/TOD /CS	BT2	Time-Of-Day Access
/PAR /CS	BT2	Parity Chip Select
/EPROM /RD	BT2	Eprom Read
/FD /RD	BT2	Floppy Disk Read
/FD /WR	BT2	Floppy Disk Write
/DAC /WR	BT2	Audio DAC write strobe
/DAC /XFER	BT2	Audio DAC transfer strobe
/IOSEL	BT2	Select for type 1 device
/SB /SEL(3:0)	BT2	SBus Selects. (includes Video)
/IOD /EN	BT4	Input/Output Data bus Enable
/AUX /WR	BT4	Auxiliary Output Write Strobe
/IRQ(15,13:11,9:1)	SCHMITCN	Interrupt Request
/IU IRL(3:0)	BT2	Integer Unit Encoded Interrupt Requests.
/OD	TLCHTU	Output Disable input
PARA	BT1	Parametric output

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT	NOTES
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t1	/IU CLK cycle time	40		ns
t2	Setup time, /AS signals before CLK	3.0		ns
t3	Hold time, /AS signals after CLK	15.0		ns
t4	Hold time, Class-1 signals after CLK	0		ns
t5	Setup time, Class-1 signals before CLK	15.0		ns
t6	Delay Class-2 to X-/SEL negated		22.0	ns
t7	Delay Class-2 to X-/SEL asserted		23.0	ns
t8	Synchronous output delay		22.0	ns

Oct 13 16:30 1989

CAPACITIVE LOADING

Signal Name	Type Cell	Unit (pF)
/IU CLK	DRVT4	*
/CONST CLK	TLCHT	*
/IO A(3:0)	TLCHT	*
CTL(2:0)	TLCHT	*
/DEVSPC	TLCHT	*
/SB RD	TLCHTU	*
/AS	TLCHT	*
/SB /RESET	TLCHN	*
IOD(7:0)	BD4TU	100
/SB /ACK8	BT4	100
/USER	TLCHN	*
/DMA	TLCHT	*
CID(3:0)	BT2	15
PMEG(7:0)	BD4TD	15
PA(27:12)	BD4T	41
MMU(v,w,s,x)	BD4T	15
/MMU TYP(1:0)	BD4T	15
MMU(a,m)	BD4T	15
/SM /WR	BT4	15
/PM /WR(2:0)	BT4	15
/RAMSEL	BT2	10
/KBM /RD	BT2	15
/KBM /WR	BT2	15
/SCC /RD	BT2	15
/SCC /WR	BT2	15
/TOD /CS	BT2	15
/PAR /CS	BT2	10
/EPROM /RD	BT2	15
/FD /RD	BT2	15
/FD /WR	BT2	15
/DAC /WR	BT2	15
/DAC /XFER	BT2	15
/IOSEL	BT2	15
/SB /SEL(3:0)	BT2	10
/IOD /EN	BT4	15
/AUX /WR	BT4	15
/IRQ(15,13:11,9:1)	SCHMITCN	*
/IU IRL(3:0)	BT2	10
/OD	TLCHTU	*
PARA	BT1	*

* Input only

7. AC OPERATING REQUIREMENTS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 15 pF)

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
t1	CLK Cycle	40		ns
t2	/AS Setup to CLK	15.0		ns
t3	/AS Hold from CLK	2.0		ns

8. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CIN	Input Capacitance		20	pF
COUT	Output Capacitance		100	pF

9. TRUTH TABLE: N/A

10. FUNCTIONAL TABLE: N/A

11. OUTLINE DRAWING: N/A

12. WAVEFORMS: N/A

DATE: 1/12/89

AUTHOR: PL

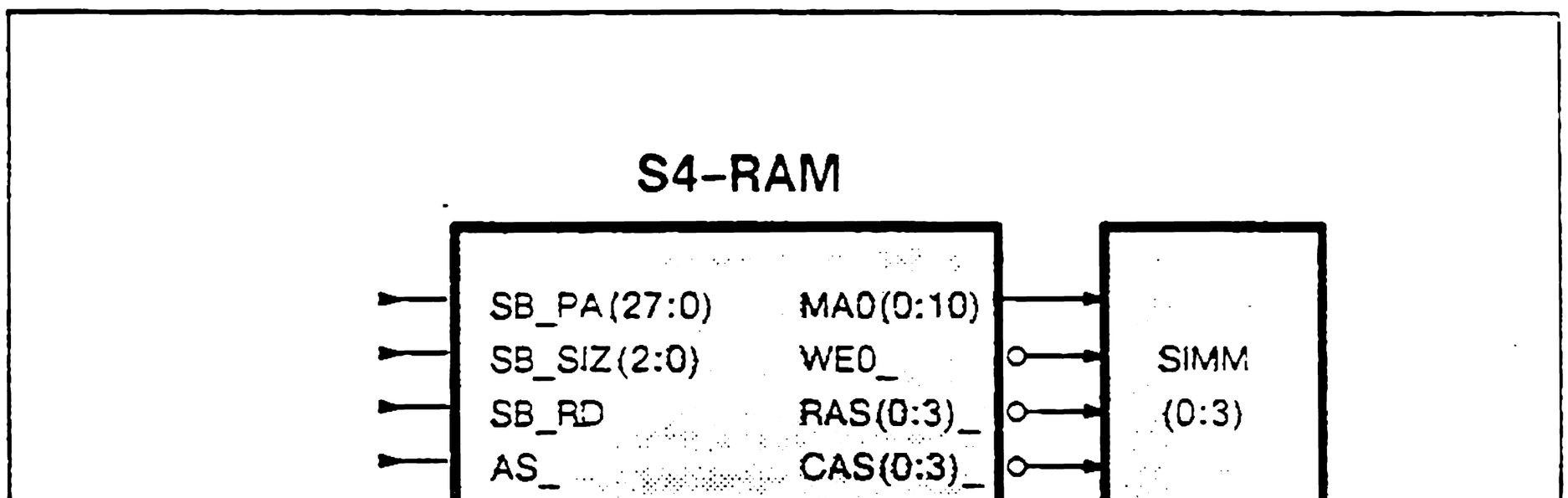
~~~~~~
~~~~~~

Sun Microsystems S4 Chip Set Sun SPARC(TM) Support System	S4-RAM Preliminary

Features

- Supports 1M and 4M DRAM SIMMs
- Directly drives 8 SIMMs (8/32 MBytes)
- Decodes configurations up to 32 SIMMs (32/128 MBytes)
- Supports high-speed burst mode with fast-page RAMs
- Contains refresh logic and timer
- Low-cost 100PFP package

The S4-RAM controller provides all address handling, RAS and CAS decoding, and control for a memory system of four or eight SIMM memory modules. The controller includes two sets of output buffers, each one driving a group of 4 SIMMs. The S4-RAM controller supports 1Mbit and 4MBit SIMMs. Up to four S4-RAM Controllers can be bank-selected for a maximum memory configurations of 32 SIMMs.



Pin Description

Name	Type	Description
Bus Interface	35	
clk	TLCHT	system clock
sb_pa(0:27)	TLCHT	physical address bus
sb_siz(0:2)	TLCHT	size of write cycles. 0:4 bytes. 1:1 byte. 2:2 bytes. 3:3 bytes. 4:4 byte bursts. 5.6.7:Not allowed
sb_rd	TLCHT	read cycle
sb_ack32_	BT4	word acknowledge
as_	TLCHN	address strobe
selram_	TLCHN	select ram
Other Inputs	6	
ramtype	IBUFD	ram type: 0:1MbitDRAM. 1:4MbitDRAM
ramspd	IBUFD	ram speed: 0:slow RAS. 1:fast RAS
bank(0:1)	IBUFD	ram bank: selects base address
por_	IBUFNU	power-on-reset and test
para_	BD1CNU	parametric test output/output disable
RAM Drivers	40	
ma0(0:10)	BT8	multiplexed address set0. drives 4 SIMMs
ma1(0:10)	BT8	multiplexed address set1. drives 4 SIMMs
we0_	BT8	write enable set 0. drives 4 SIMMs
we1_	BT8	write enable set 1. drives 4 SIMMs
ras(0:7)_	BT8	ras. drives 1 SIMM. selects bank (32-bits)
cas(0:7)_	BT8	cas. drives 1 SIMM. selects byte within bank
TOTAL:	82	

Device Number:	LMA9095	(IO:92 VDD:2 VSS:4)
Package Type:	PFP100	(PADS:82 VDD:6 VSS:10)

Functional Description

The S4-RAM controller uses addresses as follows:

Cycle	row	column
ma0	a12	a2
ma1	a13	a3
ma2	a14	a4
ma3	a15	a5
ma4	a16	a6
ma5	a17	a7
ma6	a18	a8
ma7	a19	a9
ma8	a20	a10
ma9	a21	a11
ma10	a22	a23

(ma10 is used for 4 Mbit RAMS only)

Address	4 MB RAM	1 MB RAM
a0	byte	byte
a1	byte	byte
a2	col0	col0
a3	col1	col1
a4	col2	col2
a5	col3	col3
a6	col4	col4
a7	col5	col5
a8	col6	col6
a9	col7	col7
a10	col8	col8
a11	col9	col9
a12	row0	row0
a13	row1	row1
a14	row2	row2
a15	row3	row3
a16	row4	row4
a17	row5	row5
a18	row6	row6
a19	row7	row7
a20	row8	row8
a21	row9	row9
a22	row10	set
a23	col10	bank0
a24	set	bank1
a25	bank0	0
a26	bank1	0
a27	0	0

Decoding Tables

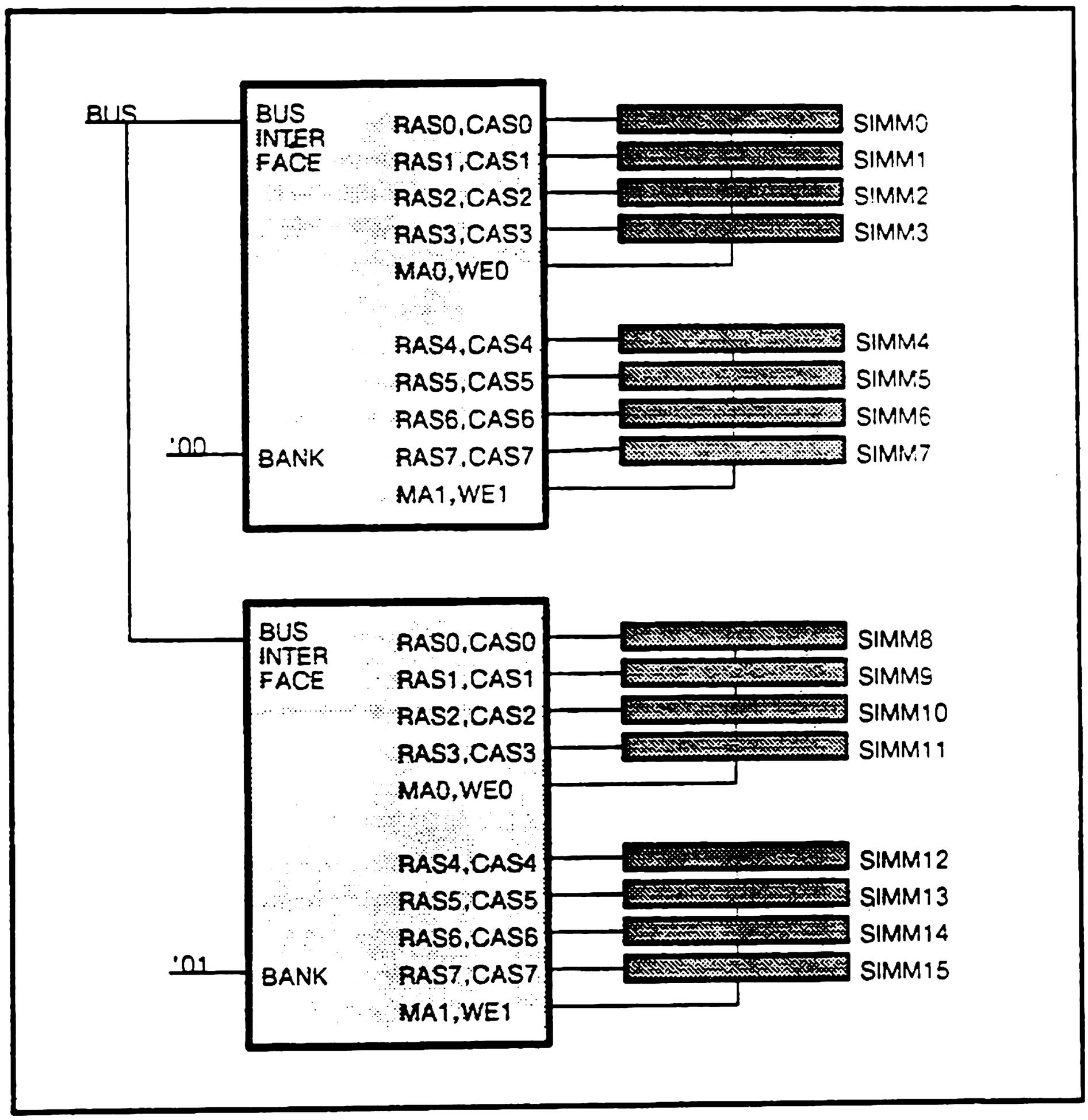
The S4-RAM controller decodes RAS and CAS as follows:

Size sb_siz1.sb_siz0	Address sb_pa1.sb_pa0	Byte(0) CAS(0,4)	Byte(1) CAS(1,5)	Byte(2) CAS(2,6)	Byte(3) CAS(3,7)
0,0	0,0	X	X	X	X
	0,1		X	X	X
	1,0			X	X
	1,1				X
0,1	0,0	X			
	0,1		X		
	1,0			X	
	1,1				X
1,0	0,0	X	X		
	0,1		X	X	
	1,0			X	X
	1,1				X
1,1	0,0	X	X	X	
	0,1		X	X	X
	1,0			X	X
	1,1				X

Mode	Bank Select	Address	RAS(3:0)	RAS(7:4)
1 Mbit	Bank(1:0)=SB_PA(24:23)	SB_PA22=0	X	
		SB_PA22=1		X
4 MBit	Bank(1:0)=SB_PA(26:25)	SB_PA24=0	X	
		SB_PA24=1		X

Multiple Chip Configurations

Up to four S4-RAM controllers can be configured into one memory subsystem by means of the bank decoding inputs. Since each S4-RAM chip can drive 8 SIMMs the maximum configuration is 32 SIMMs (32/128 Megabytes). The figure below illustrates a 16 SIMM memory configuration. Mixed 1 Mbit and 4 Mbit configurations are supported in the 4 Mbit ramtype mode which causes all banks of memory to start on 16 MByte boundaries.



Timing Diagrams

The S4-RAM controller supports three basic types of cycles: Refresh cycle, CPU cycle, and Burst Cycle. For both the refresh cycle and the CPU cycle, the S4 RAM controller supports two RAM speeds via the RAMSPD input: *fast* and *slow*. The *fast* mode supports a minimum cycle of 4 states for CPU cycles and 5 states for Refresh. In *slow* mode, RAS is extended by one additional state. This allows to use slower RAMs at the cost of an increased cycle time. There is no separate *fast* and *slow* mode for burst cycles.

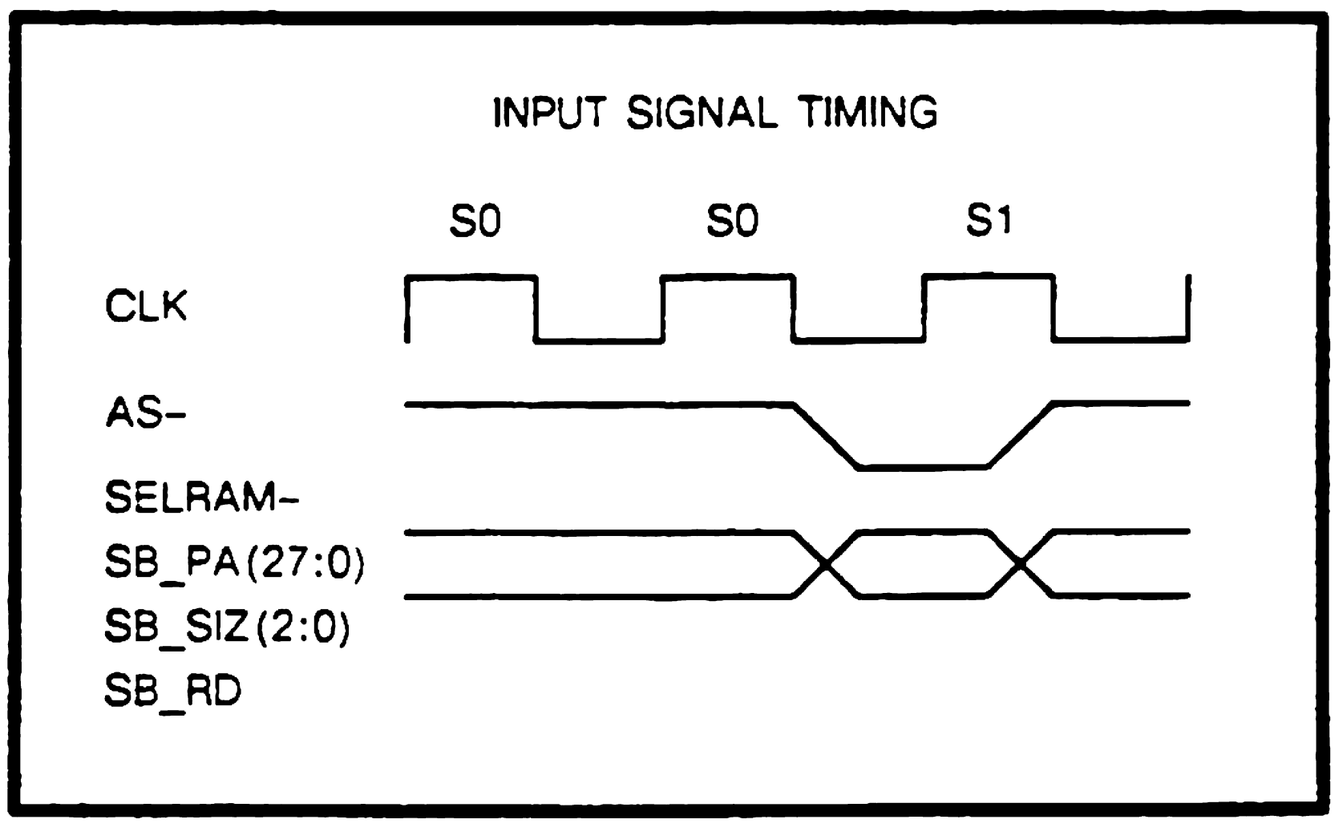
Cycle Overview. The S4-RAM controller stays in the idle state (S0) until either activated by a refresh request, causing a refresh cycle, or by a CPU select, causing a CPU or burst cycle. In case of simultaneous refresh request and CPU select the refresh request receives priority.

CPU cycles are initiated when the S4-RAM controller simultaneously receives address strobe (AS), select RAM (SELRAM), and a matching set of physical addresses (see address decoding table). In response to the CPU request, the RAM controller activates RAS for the bank of memory decoded by the addresses.

Burst cycles are a special type of CPU cycle indicated by SB_SIZ(2)=1. Burst mode causes a transfer of a 4 word block aligned on a 4 word boundary. The address presented at the beginning of the cycle is the address of the first word to be transferred; consecutive addresses are generated within the S4-RAM controller modulo 4 word addresses.

Refresh cycles are generated internally by a refresh request which occurs every 320 system clocks. This will cause one refresh request RREQ every 16 usec for a 20 MHz system clock. In order to distribute the refresh current among multiple RAM controllers, the refresh counter is preloaded on power-on-reset to a value of (bank * 2^16).

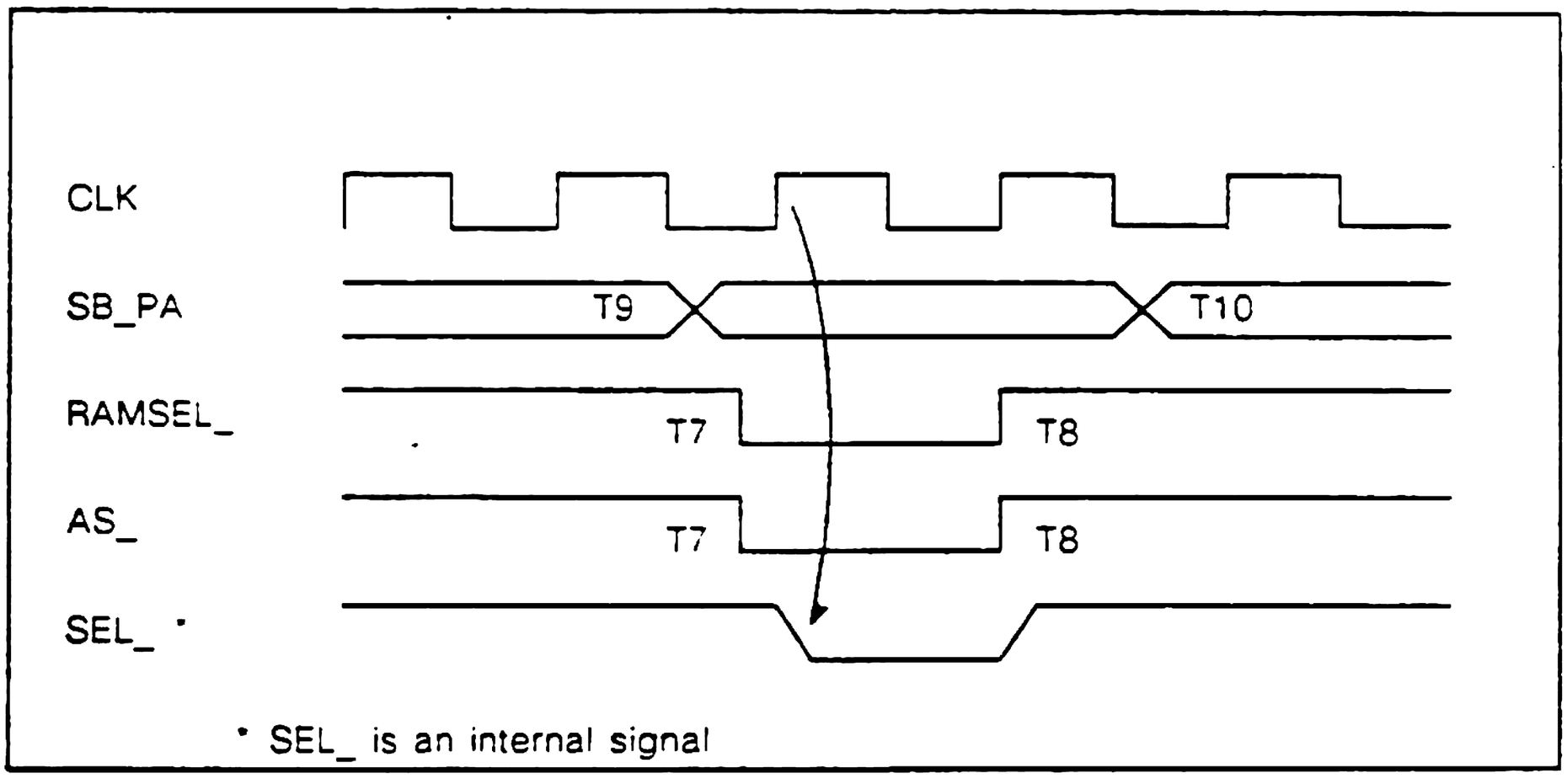
The timing diagrams below illustrate the waveforms for the different cycles. Signal transitions are illustrated simultaneous to the clock edge that causes them.



Selecting The S-4 RAM Chip

The S4-RAM chip is selected by presenting an appropriate set of addresses, by bringing RAMSEL_ active, and finally by bringing AS_ active. The address lines used to determine if the chip is to be selected depend on two other factors, the state of the BANK and RAMTYPE pins. These pins in various combinations alter the selection of the chip based on addresses. If one megabit rams are used, SB_PA(23) and SB_PA(24) must be the same state as BANK(0) and BANK(1), and SB_PA(25), SB_PA(26), SB_PA(27) must all be logic zero for the chip to be selected. The selection process is completed by bringing RAMSEL_ active (comes from S4-MMU chip) and AS_ active (comes from S4-CACHE chip). In the case of four megabit rams, SB_PA(25) and SB_PA(26) must be the same as the BANK(0) and BANK(1) pins, and SB_PA(27) must at logic zero.

The input signal combinations need only be present through the rising edge of one clock cycle. Once this happens, the S4-RAM chip will start the appropriate ram cycle and continue until completed. Once the S4-RAM chip starts a memory cycle, there is no way to abort it.

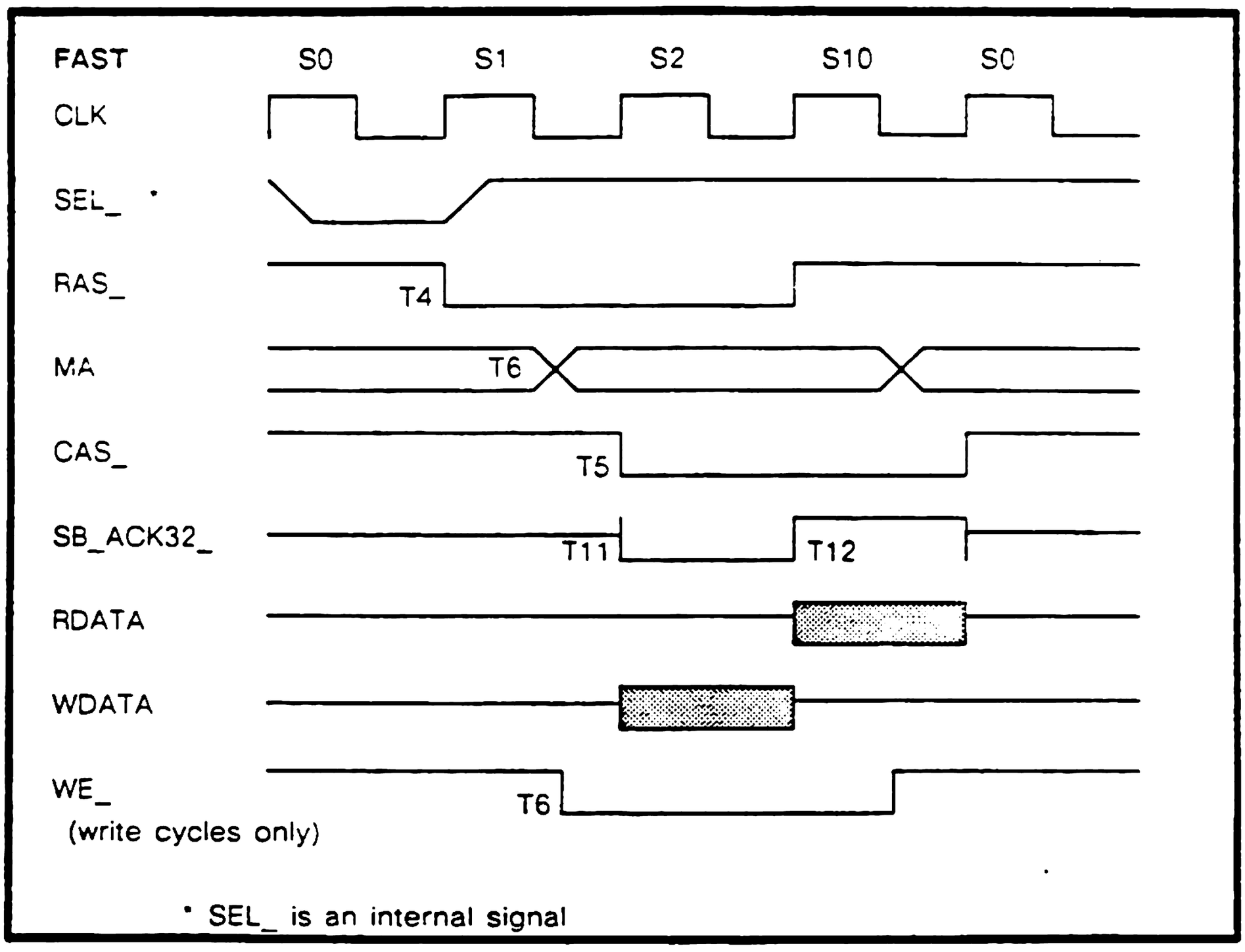


* SEL_ is an internal signal

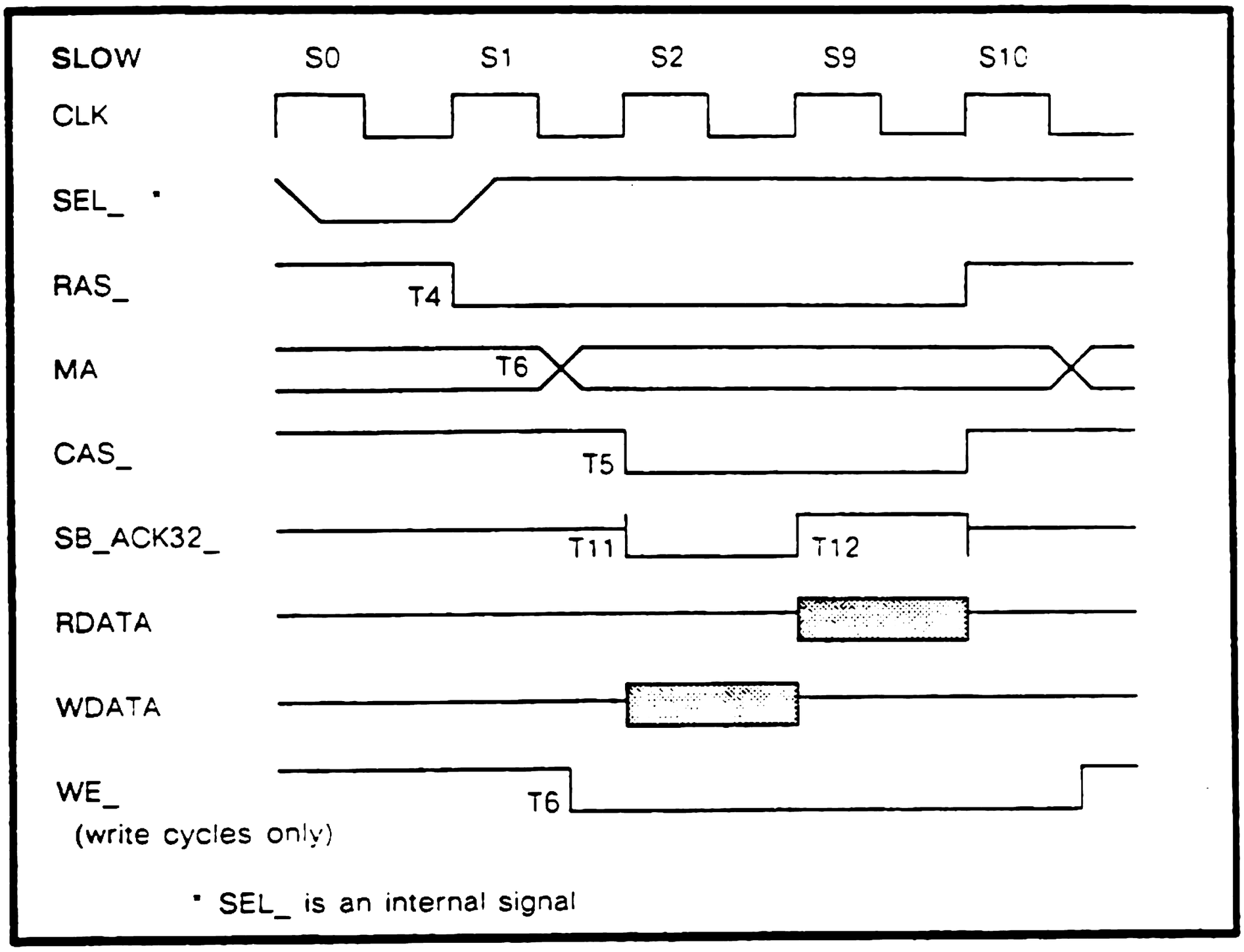
CPU Cycle

In response to a select, the RAM controller enters state S1 and asserts RAS for the bank of memory decoded by the addresses. The row/column addresses are multiplexed on the half-state following RAS. In state S2, the RAM controller asserts CAS and acknowledge signal ACK32. Following S2, in fast mode the RAM controller finishes up with S10 which deasserts RAS and ACK32 while keeping CAS asserted. In slow mode, the RAM controller extends RAS in state S9, and then deasserts all control signals in state S10. In both cases, write data (WDATA) must be valid at beginning of CAS and read data (RDATA) is valid at the end of CAS.

The following diagram illustrates timing for fast rams.

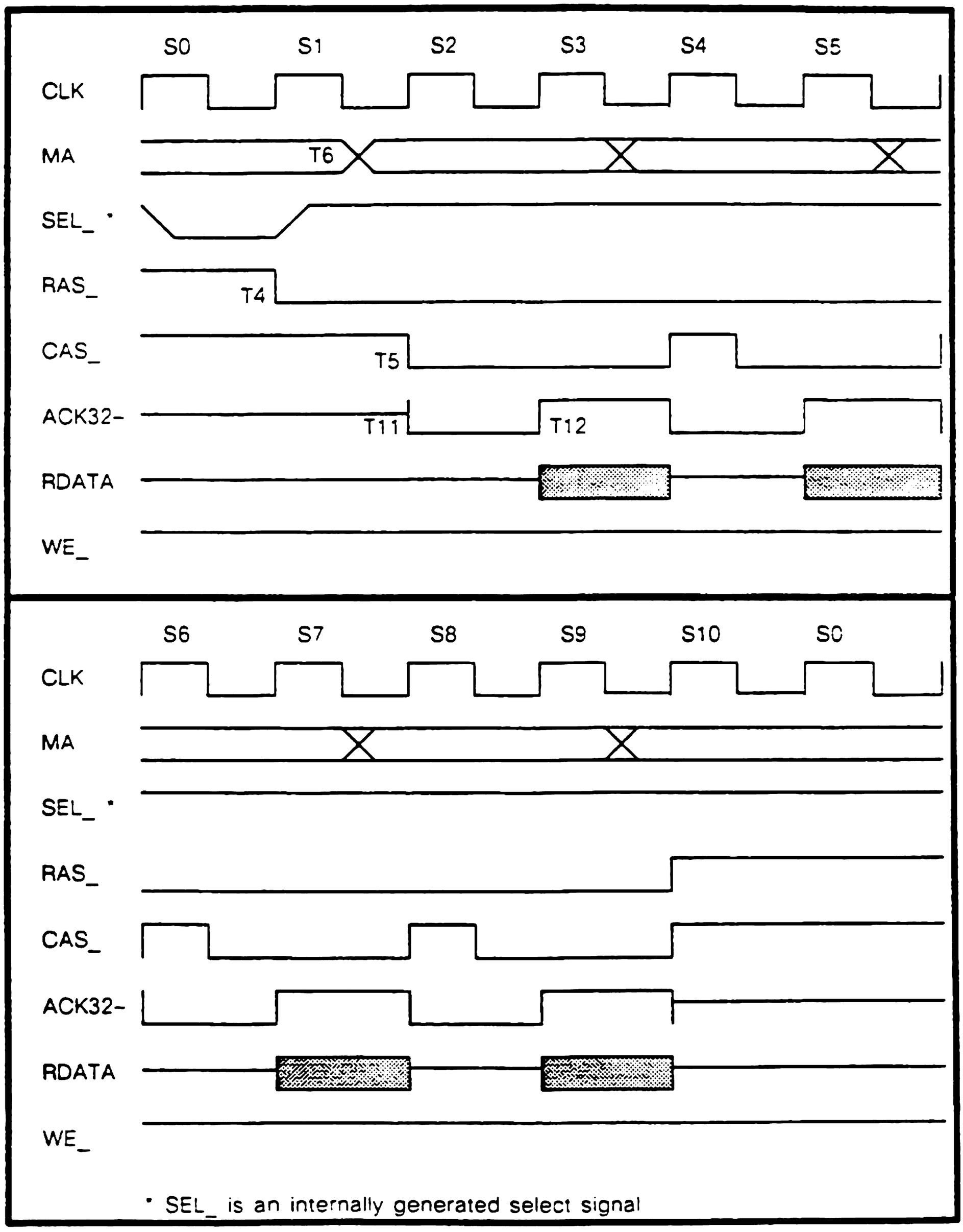


The following timing diagram illustrate timing for slow rams.



Read Burst Cycle

The S4-RAM controller supports a burst mode cycle of 4 words, using fast-page mode RAMs. In burst mode, the RAM controller will cycle through states S3 through S9, while incrementing the word-address modulo 4. There is no separate FAST and SLOW mode for burst cycles.

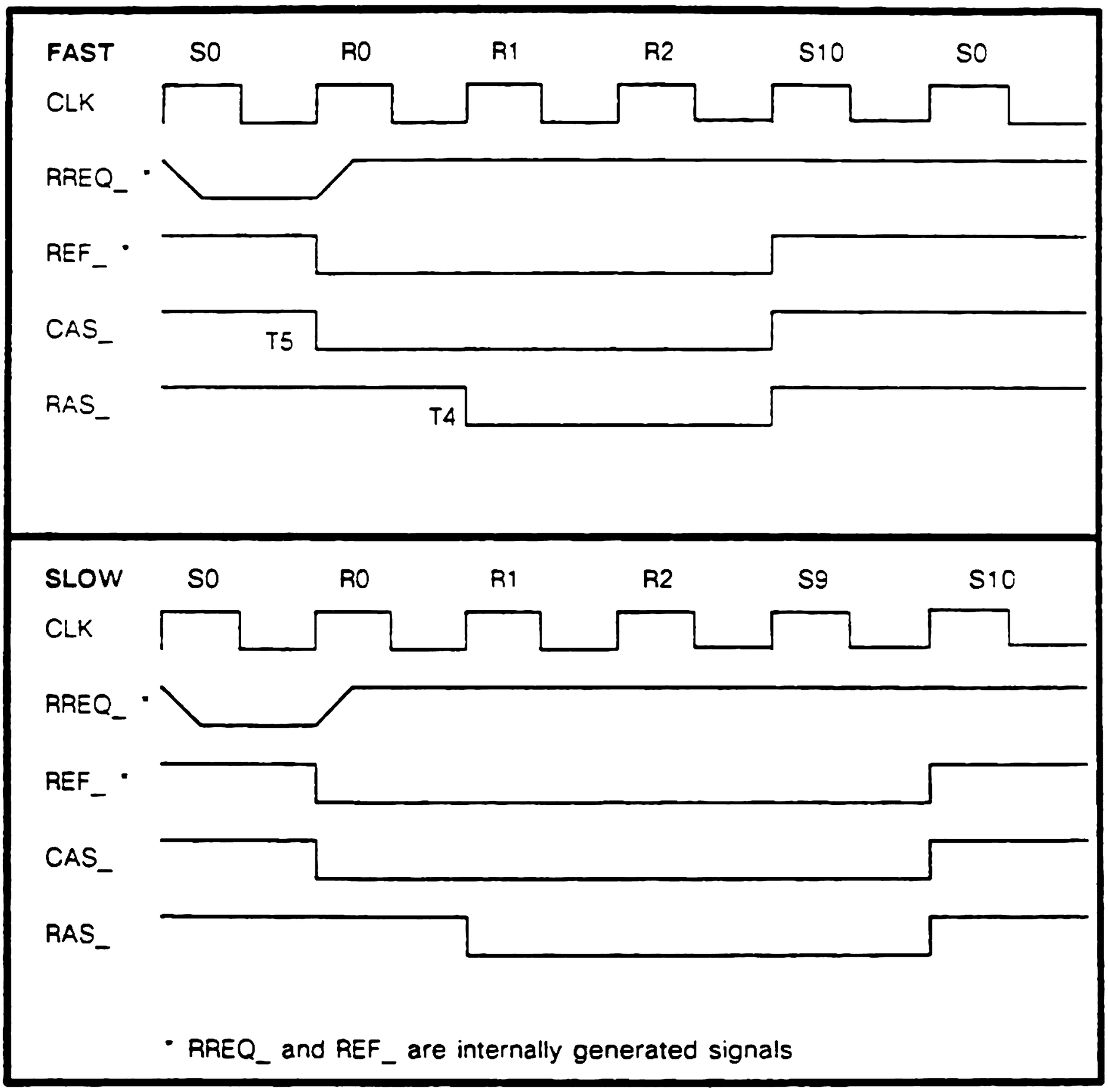


Write Burst Cycle

The write burst cycle is similar to the read burst cycle except for the CAS timing which reduces write hold time. There is no separate FAST and SLOW mode for burst cycles

Refresh Cycle

Refresh is implemented with a "CAS-before-RAS" cycle. Once a refresh request is recognized, all CAS outputs are asserted during state R0 followed by all RAS outputs asserted at state R1. REF, RAS, and CAS stay asserted during R2 and are deasserted in state S10. In "slow" mode, a state S9 is inserted that extends all control signals for one extra state. Refresh request takes priority over CPU cycles that arrive at the same time. Pending CPU cycles have to wait until they are recognized in S0.



Timing Specifications

Conditions:

VCC	4.75 to 5.25V	
TA	0 to +70C	
Cma	capacitve load MA,WE	180 pF(36*5pF)
Cctl	capacitve load RAS,CAS	85 pF (9*9pF)
Cack	capacitive load ACK32	100 pF

Symbol	From	To	min	max	unit
t1	clk high	clk cycle + clk high	40	---	ns
t2	clk high	clk low	15	---	ns
t3	clk low	clk high	15	---	ns
t4	clk high	ras valid	5	16.5	ns
t5	clk low	cas valid	5	16.5	ns
t6	clk high	ma, we valid	5	25	ns
t7	input setup	clk high	15	---	ns
t8	clk high	input hold	---	0	ns
t9	PA setup	clk high	18	---	ns
t10	PA hold	clk high	---	0	ns
t11	clk high	SB_ACK32_ valid	6.5	23	ns
t12	clk high	SB_ACK32_ invalid	4.5	16.5	ns

Input Setup Times (measured relative to clock)

Pin	max	unit
SB_PA(27:0)	18	ns
RAMSEL_	15.5	ns
all other inputs (except clock)	15.0	ns

Change History

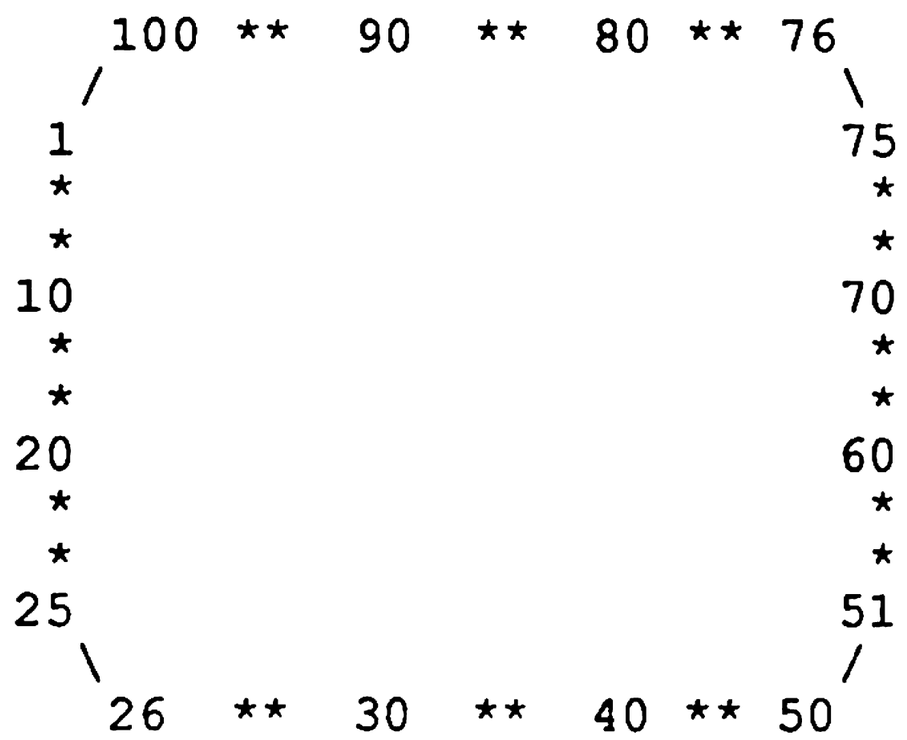
Date	Change	By
12/14/87	First release	AVB
1/18/88	Added Select information, additional timing waveforms and delay numbers.	MW
1/27/88	Corrected typos	MWi
1/29/88	Corrected block picture pin names	MWi
3/3/88	Corrected pin lables on page one	MWI
7/18/88	Updated Timing Numbers	MWI

1. P/N: 100-1805-02 2. REV: 50

3. DESCRIPTION: IC,GA,S4-RAM

4. PHYSICAL MODEL: (100 Pin PFP)

A. Pinout: *Top View



1 - MA0.0	26 - /WE0	51 - /POR	76 - /CAS.4
2 - MA0.1	27 - RAMTYPE	52 - /PARA	77 - /RAS.4
3 - MA0.2	28 - RAMSPD	53 - /SB /ACK32	78 - MA1.2
4 - /RAS.0	29 - BANK.1	54 - VDD	79 - MA1.1
5 - /CAS.0	30 - BANK.0	55 - /WE1	80 - MA1.0
6 - VDD	31 - /SB PA.27	56 - /CAS.7	81 - /SB SIZ.0
7 - VSS	32 - /SB PA.25	57 - /RAS.7	82 - /SB SIZ.1
8 - MA0.3	33 - /SB PA.23	58 - MA1.10	83 - /SB SIZ.2
9 - MA0.4	34 - /SB PA.21	59 - MA1.9	84 - /SB PA.0
10 - MA0.5	35 - /SB PA.19	60 - MA1.6	85 - /SB PA.2
11 - /RAS.1	36 - /SB PA.17	61 - VSS	86 - /SB PA.4
12 - /CAS.1	37 - /SB PA.15	62 - VDD	87 - /SB PA.6
13 - VSS	38 - /SB PA.13	63 - /CAS.6	88 - /SB PA.8
14 - VDD	39 - /SB PA.11	64 - /RAS.6	89 - VDD
15 - MA0.7	40 - VSS	65 - MA1.8	90 - /SB PA.12
16 - MA0.8	41 - VDD	66 - MA1.7	91 - /SELRAM
17 - /RAS.2	42 - CLK	67 - VDD	92 - VSS
18 - /CAS.2	43 - /SB PA.9	68 - VSS	93 - /SB PA.10
19 - VDD	44 - /SB PA.7	69 - /CAS.5	94 - /SB PA.14
20 - MA0.6	45 - /SB PA.5	70 - /RAS.5	95 - /SB PA.16
21 - VSS	46 - /SB PA.3	71 - MA1.5	96 - /SB PA.18
22 - MA0.9	47 - /SB PA.1	72 - MA1.4	97 - /SB PA.20
23 - MA0.10	48 - /AS	73 - MA1.3	98 - /SB PA.22
24 - /RAS.3	49 - /SB RD	74 - VSS	99 - /SB PA.24
25 - /CAS.3	50 - VSS	75 - VDD	100 - /SB PA.26

B. PIN DESCRIPTION:
==========

SYMBOL	TYPE	DESCRIPTION
BUS INTERFACE	36	
CLK	TLCHT	System Clock
/SB PA(0:27)	TLCHT	Physical Address Bus
/SB SIZ(0:2)	TLCHT	Size of write cycle. 0:4 bytes, 1:1 byte, 2:2 bytes, 3:3 bytes, 4:4 byte bursts. 5,6,7:Not allowed
/SB RD	TLCHT	Read Cycle
/SB /ACK32	BT4	Word Acknowlege
/AS	TLCHN	Address Strobe
SELRAM	TLCHN	Select Ram
OTHER INPUTS	6	
RAMTYPE	IBUFD	Ram Type: 0:1MbitDRAM, 1:4MbitDRAM
RAMSPD	IBUFD	Ram Speed: 0:slow RAS, 1:fast RAS
BANK(0:1)	IBUFD	Ram bank: selects base address
/POR	IBUFNU	Power-On-Reset and Test
/PARA	BD1CNU	Parametric test output/output disable
RAM DRIVERS	40	
MA0(0:10)	BT8	Multiplexed address set0, drives 4 SIMMs
MA1(0:10)	BT8	Multiplexed address set1, drives 4 SIMMs
/WE0	BT8	Write enable set 0, drives 4 SIMMs
/WE1	BT8	Write Enable set 1, drives 4 SIMMs
/RAS(0:7)	BT8	RAS, drives 1 SIMM, selects bank (32-bits)
/CAS(0:7)	BT8	CAS, drives 1 SIMM, selects byte within bank
TOTAL:	82	

5. DC CHARACTERISTICS:
(VCC = 4.75 to 5.25V, TA = 0 to +70oC, Output Load = 100 pF)

SYMBOL	PARAMETER	LIMITS			UNIT	NOTES
		Min	Typ	Max		
VIH	Input High Voltage					
	TTL Inputs	2.0			V	
	Temperature Range					
	CMOS Levels	3.5			V	
VIL	Input Low Voltage					
	TTL Inputs			0.8	V	
	CMOS Levels			1.5	V	
VT+	Schmitt-Trigger, Positive-going Threshold		3.0	4.0	V	
VT-	Schmitt-Trigger, Negative-going Threshold	1.0	1.5		V	
VOH	Output High Voltage					
	Type B1					1
	Type B2	2.4	4.5		V	2
	Type B4					3
	Type B8					4,6
	Type B12					5,7
VOL	Output Low Voltage					
	Type B1				V	8
	Type B2		0.2	0.4	V	9
	Type B4				V	10
	Type B8				V	11,6
	Type B12				V	12,7
	Hysteresis, Schiimitt Trigger	1.0	1.5		V	13
II	Input Current, CMOS, TTL Inputs	-10	+/-1	10	uA	14
	Inputs with Pulldown Resistors	10	35	120	uA	15
	Inputs with Pullup Resistors	-100	-30	-8	uA	16
IOS	Output Short Circuit Current	15	50	130	mA	17,21
		-5	-25	-100	mA	18,21
IOZ	3-State Output Leakage Current	-10	+/-1	10	uA	19
IDD	Quiescent Supply Current	User-Design Dependent				20

NOTES:

1. IOH = -1.0 mA
2. IOH = -2.0 mA
3. IOH = -4.0 mA
4. IOH = -8.0 mA
5. IOH = -12 mA
6. Requires one output pad
7. Requires two output pads
8. IOL = 1.0 mA
9. IOL = 2.0 mA
10. IOL = 4.0 mA
11. IOL = 8.0 mA
12. IOL = 12 mA
13. VIL to VIH
    VIH to VIL
14. VIN = VDD or VSS
15. VIN = VDD
16. VIN = VSS
17. VDD = Max, VO = VDD
18. VDD = Max, VO = 0V
19. VOH = VSS or VDD
20. VIN = VDD or VSS
21. Type B4 output. Output short circuit current for other outputs will scale. Not more than one output may be shorted at a time for a maximum duration of one second.

6. AC CHARACTERISTICS:

(VCC = 4.75 to 5.25V, TA = 0 to +70oC)

CMA Capacitive load MA,WE 180 pF(36*5pF)

CCTL Capacitive load RAS,CAS 85 pF(9*9pF)

CACK Capacitive load ACK32 100 pF

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
t1	From CLK High to CLK Cycle + CLK High	40		ns
t2	From CLK High to CLK Low	15.0		ns
t3	From CLK Low to CLK High	15.0		ns
t4	From CLK High to RAS Valid	5.0	16.5	ns
t5	From CLK Low to CAS Valid	5.0	16.5	ns
t6	From CLK High to MA, WE Valid	5.0	25.0	ns
t7	From Input Setup to CLK High	16.0		ns
t8	From CLK High to Input Hold		0	ns
t9	From PA Setup to CLK High	18.0		ns
t10	From PA Hold to CLK High		0	ns
t11	From CLK High to /SB /ACK32 valid	6.5	23.0	ns
t12	From CLK High to /SB /ACK32 Invalid	4.5	16.5	ns

INPUT SETUP TIMES (measured relative to clock)

PIN	LIMITS Min	LIMITS Max	UNIT
/SB PA(27:0)		18.0	ns
/RAMSEL		15.5	ns
All other inputs (except clock)		15.0	ns

7. CAPACITANCE:

SYMBOL	PARAMETER	LIMITS Min	LIMITS Max	UNIT
CI/O	Input/Output Capacitance per Slot		20	pF
CLS	Maximum Capacitive Load per System		100	pF

8. TRUTH TABLE: N/A

9. FUNCTIONAL TABLE: N/A

10. OUTLINE DRAWING: N/A

11. WAVEFORMS: N/A

DATE: 1/04/89

AUTHOR: PL

~~~~~~
~~~~~~