KA48 System Board Specification

Order Number KA48-0-DBF

Revision/Update Information: X.01

John Kirk
MLO5-2/B6
223-4690
MAST::KIRK

5-SEP-1991 08:52:02.34

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may occur in this document.

This specification does not describe any program or product which is currently available from Digital Equipment Corporation, nor does Digital Equipment Corporation commit to implement this specification in any program or product. Digital Equipment Corporation makes no commitment that this document accurately describes any product it might ever make.

This document was prepared using VAX DOCUMENT, Version 1.2

Chapter 1

INTRODUCTION

1.1 Scope of Document

This document specifies the functional characteristics of the KA48-AA System Module.

1.2 General Description

The KA48-AA System Module forms the basis for one system:

- The VS4000 VLC single-user engineering workstation, which includes an LK401 keyboard, a VSXXX-AA mouse or VSXXX-AB tablet, SCSI and Ethernet controllers and four serial line controller. A daughter module plugs into the system module to provide an 8-plane video frame buffer and DAC that can support monochrome (single plane or gray scale) or color monitors.
- A variety of monitors support monochrome and color displays, see Chapter 13, Section 13.6.

Each VS4000 VLC is housed in a desktop enclosure which contains a KA48-AA system board, a ???? video/sound module and an H7109 power supply. The KA48-AA system board contains the following standard components:

- DC 222 (SOC) processor with integral FPA and cache
- Sockets for 8, 16 or 24 MBytes of RAM memory (2, 4 or 6 in alternative configuration)
- 256 Kbyte ROM memory (socketed)
- 32-byte network address ROM (socketed)
- time-of-year clock including 50 bytes of non-volatile RAM
- four asynchronous serial ports for
  keyboard
  mouse or tablet
  printer

communications (includes full modem control)

- controller for Transceiver cable connect Ethernet network
- SCSI controller
- graphics hardware assist integrated within the RAM controller
- 8-plane frame buffer, 1024 x 1024 capability (daughter module)

The ???? contains the following components:

- Video RAMs for 8-plane 1K x 1K display
- RAMDAC
- Sound I/O chip
- Keyboard, Mouse/Tablet/Sound I-O connectors
- HALT push-button
- Alternate Console Select Switch

1.3 Applicable Documents

The following documents describe the principal LSI chips which are used in the VS4000 VLC:

A-PS-2132064-0-0	DC 222 (SOC) processor chip
A-PS-21-34159-0-0	DC7201, memory/graphics/IO controller.
A-PS-21-35025-0-0	, logic compaction array #1.
A-PS-21-35024-0-0	DC7238, logic compaction array #2.
A-PS-2121672-0-0	LANCE Ethernet Controller chip
A-PS-2118795-0-0	MC146818 Watch chip
A-PS-2131889-0-0	53C94 SCSI Interface chip
A-PS-23365A1-0-0	Ethernet Network Address ROM
A-PS-2132756-0-0	79C30 (Sound Chip)
A-PS-21-??????-0-0	BT458, Video Digital to Analog Convertor.

The following documents specify the cables to external equipment to be used with the system:

A-PS-1700300-0-0	BCC16E cable for printer
A-PS-1700568-0-0	BC18P cable for monochrome video
A-PS-1701480-0-0	BC19S cable for color video

The following documents describe other components to be used with the system:

A-SP-H7109	Power supply and fan unit
A-SP-LK401-0-0	LK401 keyboard
A-SP-VRXXX-A-0	MONO 14" 14-inch monochrome monitor
A-SP-VRXXX-A-0	VR319 19-inch monochrome monitor
A-SP-VRXXX-A-0	VRC16 16-inch color monitor
A-PS- -0-0	VRT19 19-inch color monitor
A-PS- -0-0	VSXXX-AA mouse
A-PS- -0-0	VSXXX-AB tablet

The following documents describe the firmware in the system ROM:

VS4000 VLC System Firmware Specification
VS4000 VLC Power On Initialization
VS4000 VLC System Exerciser specification
VMB Specification for VS4000 VLC

The following Digital Standards documents contain material applicable to this system:

DEC STD 002	AC Power Wiring
DEC STD 032	VAX Architecture Standard
DEC STD 052	Serial Terminals and Serial System Interfaces
DEC STD 102	Environmental Standards
DEC STD 103	Electromagnetic Compatibility
DEC STD 104	Acoustic Noise Acceptability
DEC STD 119	Product Safety

DEC STD 122	AC Power Line Standard
DEC STD 134	DEC Ethernet Specification
DEC STD 200	DEC Remote Diagnosis Specification

The following documents are provided to guide customers in the installation and use of the system:

VS4000 VLC Hardware Installation Guide
VS4000 VLC Owner's Manual
VS4000 VLC Network Guide
VS4000 VLC Maintenance Guide
VS4000 VLC Technical Manual

1.4 Notation Conventions

Physical memory addresses are expressed as eight hexadecimal digits punctuated with a period between the fourth and fifth digits for readability.

Hexadecimal numbers are denoted by a trailing "h", as for example FEh whose decimal equivalent is 254. All other numbers are decimal.

Interrupt vectors are given as three hexadecimal digits which are a byte offset into an SCB.

Chapter 2

CENTRAL PROCESSOR

The system central processor is a single CMOS chip fabricated in Digital's CMOS-3 process. This chip is fully described in its specification document which is cited in Section 1.3. This section covers the manner in which the chip is used by the VS4000 VLC; it does not attempt to reproduce all the material in the chip specification.

2.1 Processor Summary

The processor chip is a 32-bit virtual memory microprocessor with integral floating point capability and an on-chip cache, derived directly from the existing CVAX CPU, CFPA floating point unit and CVAX Clock chips. The chip is implemented in Digital's CMOS-3 process. Its key features are:

1. Subset of VAX data types: byte, word, longword, quadword, character string, and variable-length bit fields, support for f_floating, d_floating, and g_floating point data types. Support for the remaining VAX data types can be provided by macrocode emulation.
2. Full base instruction group: integer and logical, address, variable-length bit field, control, procedure call, miscellaneous, queue, character string instructions CMPC3/CMPC5, LOCC, MOVC3/MOVC5, SCANC, SKPC and SPANC, and operating system support. F_ floating, d_floating and g_floating point instructions are supported by the CFPA chip (including the floating point instructions that are part of the emulate-only instruction group: POLYf, EMULf, and ACBf). The remaining VAX instructions can be implemented via macrocode emulation. The chip provides microcode assists for emulation of the character string instructions which are not included in the full base instruction group, and the decimal string, EDITPC, and CRC instructions.
3. Full VAX memory management: includes a demand paged memory management unit which is fully compatible with VAX memory management. System space addresses are virtually mapped through single-level page tables; process space addresses are virtually mapped through double-level page tables. Supports four gigabytes (2**32) of virtual memory and up to one gigabyte (2**30) of physical memory.

4. On-chip memory cache: the processor chip has 8 KBytes of on-chip cache to improve execution times by minimizing the performance required of the memory subsystem. The cache is an eight set direct mapped write-through cache. To improve manufacturing yield, one or more of the eight sets may be disabled during the chip test process, allowing for possible die defects without causing the entire chip to be rendered unuseable. It is likely that a minimum of six sets will be functional. See Section 2.11.2.2
5. Single package: the CPU is packaged in a single 132-pin surface mounted cerquad package.
6. External data bus is 32 bits wide. External bus I/O cycles run at one half the internal bus speed. They may be lengthened in increments of one internal bus period to accommodate the timing requirements of various devices connected to the bus.

2.2 Differences from Full VAX Architecture

The principal differences between this processor chip and the full VAX architecture are these:

1. These data types are omitted: decimal string, octaword, and h_floating. These data types can be supported by macrocode emulation.
2. These instruction classes are omitted: the three character string instructions MATCHC, MOVTC and MOVTUC, decimal string instructions, EDITPC, CRC, compatibility mode, octaword instructions, and h_floating instructions. The chip provides microcode assists for the macrocode emulation of the character string, decimal string, CRC, and EDITPC instuctions.
3. These internal processor registers are omitted: NICR, ICR, TODR, RXCS, RXDB, TXCS, TXDB, and PME.
4. The chip does not have a built-in console function. (The console is implemented in the system firmware.)

2.3 Processor Clock

The oscillator that determines the CPU operating frequency connects directly to the CPU where it is divided by four to produce two anti-phase clocks for use outside the CPU. For a CPU internal cycle of 40 ns, the oscillator has a frequency of 106 MHz. and results in external clocks with a period of 160ns.

2.4 CPU DMA Access

The CPU DMA feature is used in performing cache invalidate cycles, using the fast-invalidate feature - see Chapter 4, Section 4.3.

2.5 Internal Processor Registers

The internal processor register (IPR) space provides access to many types of processor control and status registers such as the memory management base registers, parts of the process status longword, and the multiple stack pointers. These registers are explicitly accessible only by the privileged Move to Processor Register (MTPR) and Move from Processor Register (MFPR) instructions.

The following table enumerates the processor registers and indicates how they are implemented in the VS4000 VLC by one of the following codes:

I – implemented as specified in the VAX Architecture Reference Manual (DEC STD 032).

M – implemented as specified in the Processor Chip Specification.

R – access not allowed (reserved operand fault).

X – not implemented; read as zero, NOPed on write.

Table 2–1: Processor Registers

Number	Name	Description	Note	Section
0	KSP	Kernel Stack Pointer	I	
1	ESP	Executive Stack Pointer	I	
2	SSP	Supervisor Stack Pointer	I	
3	USP	User Stack Pointer	I	
4	ISP	Interrupt Stack Pointer	I	
5		not implemented	X	
6		not implemented	X	
7		not implemented	X	
8	P0BR	P0 Base Register	I	
9	P0LR	P0 Length Register	I	
10	P1BR	P1 Base Register	I	
11	P1LR	P1 Length Register	I	
12	SBR	System Base Register	I	
13	SLR	System Length Register	I	
14		not implemented	X	
15		not implemented	X	
16	PCBB	Process Control Block Base	I	
17	SCBB	System Control Block Base	I	
18	IPL	Interrupt Priority Level	I	
19	ASTLVL	AST Level	I	
20	SIRR	Software Interrupt Request	I	
21	SISR	Software Interrupt Summary	I	
22		not implemented	X	
23		not implemented	X	

Table 2–1 (Cont.): Processor Registers

Number	Name	Description	Note	Section
24	ICCS	Interval Clock Control	M	2.5.3
25	NICR	Next Interval Count	X	
26	ICR	Interval Count	X	
27	TODR	Time of Year	X	
28		not implemented	X	
29		not implemented	X	
30		not implemented	X	
31		not implemented	X	
32		not implemented	X	
33		not implemented	X	
34		not implemented	X	
35		not implemented	X	
36		not implemented	X	
37	CCR	Cache Control	M	2–7
38		not implemented	X	
39	MSER	Memory System Error	M	2.11.2.3
40		not implemented	X	
41		not implemented	X	
42	SAVPC	Console Saved PC	M	2.10
43	SAVPSL	Console Saved PSL	M	2.10
44		not implemented	X	
45		not implemented	X	
46		not implemented	X	
47		not implemented	X	
48		not implemented	X	
49		not implemented	X	
50		not implemented	X	
51		not implemented	X	
52		not implemented	X	
53		not implemented	X	
54		not implemented	X	
55		not implemented	X	
56	MAPEN	Memory Management Enable	I	
57	TBIA	Translation Buffer Invalidate All	I	
58	TBIS	Translation Buffer Invalidate Single	I	
59		not implemented	X	
60		not implemented	X	
61		not implemented	X	
62	SID	System Identification	I	
63	TBCHK	Translation Buffer Check	I	
64:127		reserved	R	

2.5.1 System Identification Register (SID)

The SID register (internal processor register 62, read-only) has the following format:

Figure 2–1: System Identification Register (SID)

```
 3             2 2
 1             4 3                             8 7             0
+---------------+-------------------------------+---------------+
|     TYPE      |           reserved            | microcode rev |
+---------------+-------------------------------+---------------+
```

The TYPE field has the value 20 (decimal) which identifies the processor as a DC222 processor chip. The contents of the "reserved" field are UNPREDICTABLE. The "microcode rev" field identifies the revision level of the chip internal microcode.

2.5.2 System Type Register (SYS_TYPE)

The SYS_TYPE register is a read-only longword contained in the system's main board ROM at physical address 2004.0004h. It has the following format:

Figure 2–2: System Type Register (SYSTYPE)

```
 3             2 2             1 1
 1             4 3             6 5             8 7             0
+---------------+---------------+---------------+---------------+
|   SYS_TYPE    |   revision    |  SYS_DEPEND   |  ARCH_IDENT   |
+---------------+---------------+---------------+---------------+
```

The SYS_TYPE field has a value of 4 which indicates that this is a VS4000 VLC. The revision field must be a non-zero value. The SYS_DEPEND field is 0 or 1. The ARCH_IDENT field is a value of 1.

2.5.3 Interval Clock Control and Status (ICCS)

The ICCS register (internal processor register 24, read/write) controls interval timer interrupts. It is similar to the ICCS register in the full VAX architecture but contains only a single bit to enable or disable the interval timer interrupt.

Figure 2–3: Interval Clock Control and Status(ICCS)

```
 3
 1                                               7  6  5        0
+-------------------------------------------------+---+----------+
|                      ..0..                      |IEN|   ..0..  |
+-------------------------------------------------+---+----------+
```

<31:7> Unused. Read as zero; ignored on write.

IEN Interrupt Enable (bit 6). When this read/write bit is set, interval timer interrupts are enabled. When it is clear, interval timer interrupts are disabled. IEN is cleared upon power-on.

<5:0> Unused. Read as zero; ignored on write.

2.6 Reset and Power-On

The CPU chip input SYSRESET_L is asserted (held LOW) during Power-on until the power supply voltage has risen to its operating level, it is then de-asserted asynchronously. The CPU synchronises this signal and produces a reset signal for use by the rest of the system - RESET_L. This signal is de-asserted following a low to high transition of MCLKAO - one of the two clocks output by the CPU chip.

2.7 HALT

Logic external to the CPU asserts HALT_L to transfer control to console microcode. At the completion of the current macro-instruction, the CPU enters the restart process with a restart code of 2. See Section 2.10

2.8 System Control Block

The System Control Block (SCB) consists of two physically-contiguous pages (1024 bytes) containing the vectors for servicing interrupts and exceptions. The start of the first of its pages is pointed to by the System Control Block Base (SCBB) internal processor register. The following table summarizes the vectors used by this system:

Vectors	**Name**	**Type**	**Notes**
000	Passive Release	Interrupt	4
004	Machine Check	abort	1
008	Kernel Stack Invalid	abort	
00C	Power Fail	interrupt	2
010	Resv/Priv Instruction	fault	
014	Customer Resv Instr	fault	
018	Reserved Operand	fault/abort	
01C	Reserved Addressing Mode	fault	
020	Access Control Violation	fault	
024	Translation Not Valid	fault	
028	Trace Pending	fault	

Vectors	Name	Type	Notes
02C	Breakpoint Instruction	fault	
030	unused		
034	Arithmetic	trap/fault	
038 . . . 03C	unused		
040	CHMK	trap	
044	CHME	trap	
048	CHMS	trap	
04C	CHMU	trap	
050	unused		
054 . . . 05C	unused		
060	Memory Error	interrupt	2
064 . . . 0C	unused		
080	Interprocessor	Interrupt	
084 . . . 0BC	Software levels 1-15	interrupt	
0C0	Interval Timer	interrupt	3
0C4	unused		
0C8	Emulation Start	fault	
0CC	Emulation Continue	fault	
0D0 . . . 0FC	unused		
100 . . . 1FC	Adapter Vectors	interrupt	
200 . . . 3FC	Device Vectors	interrupt	4

Notes:

1. See Section 2.9.1.
2. See Section 2.9.
3. See Section 2.9.2.
4. See Chapter 3.

2.9 Interrupts and Exceptions

Both interrupts and exceptions divert program execution from its normal flow by pushing the processor status and program counter onto the stack and then beginning execution at the address found in one of the interrupt vectors in the system control block (SCB). Interrupts and exceptions can arise either from conditions detected within the processor chip or from external signals sent to the processor chip. The internally detected cases are fully described in the processor chip specification; they are not further discussed here.

External interrupt causes are signalled to the processor via one of nine signals:

Signal	Description	Vector	IPL	Notes
ERR	External machine check	004		not used
PWRFL	Power fail	00C	1E	not used

Signal	Description	Vector	IPL	Notes
MEMERR	Memory error	060	1D	not used
CRD	Corrected Read Data	054	1A	not used
IRQ3	Device interrupt		17	not used
INTTIM	Interval timer	0C0	16	see Section 2.9.2
IRQ2	Device Interrupt		16	not used
IRQ1	Device interrupt		15	see Chapter 3
IRQ0	Device interrupt		14	not used

The PWRFL, MEMERR, CRD, IRQ3 IRQ2 and IRQ0 signals are not used. Their pins are tied permanently to the inactive state so that the processor will never generate the corresponding interrupts.

The IRQ1 signal is generated by the interrupt controller which is described in chapter 3. This controller receives interrupt requests from all the I/O devices in a VS4000 VLC system and generates the appropriate vector for each source.

2.9.1 Machine Check Exceptions

A machine check exception results from either an internal processor of floating point unit error, or external response on the CPU ERR L line. In response to a machine check exception, the following parameters are pushed on the stack:

Figure 2–4: Machine Check Exception Stack Data

```
+-----------------------------------------+
|        byte count (0000.0010h)          |  :SP
+-----------------------------------------+
|        machine check code               |
+-----------------------------------------+
|        most recent virtual address      | * physical for codes 81/83h
+-----------------------------------------+
|        internal state data 1            | See CPU spec.
+-----------------------------------------+
|        internal state data 2            |  "   "   "
+-----------------------------------------+
|                PC                       | PC at start of instruction
+-----------------------------------------+
|                PSL                      | Current PSL
+-----------------------------------------+
```

Machine check code (HEX)	Reason
1	FPA protocol error
2	FPA reserved instruction
3	FPA unknown error
4	FPA unknown error
5	process PTE in P0 space (TB miss)
6	process PTE in P1 space (TB miss)
7	process PTE in P0 space (M = 0)
8	process PTE in P1 space (M = 0)
9	undefined interrupt IPL code
A	impossible microcode state (MOVCx)
80	read bus error, normal read
81	read bus error, SPTE, PCB, or SCB read
82	write bus error, normal write
83	write bus error, SPTE or PCB write

2.9.2 Interval Timer Interrupt

An interval timer interrupt request is generated every 10 milliseconds by a signal on the INTTIM pin which is derived from the serial line clock crystal. This interrupt is at IPL 16h and uses interrupt vector 0C0h. The ICCS processor register (see Section 2.5.3 contains a mask bit to enable or disable the interval timer interrupt.

2.9.3 Device Interrupts

Interrupt requests from the system's I/O controllers are collected by the interrupt controller (described in Chapter 3 of this specification) which ranks their priority and presents a single interrupt request to the processor on its IRQ1 pin. All such interrupt requests are presented at IPL 15h - see Figure 3–4. The number of the interrupt vector is determined by the interrupt controller according to the identity of the requesting I/O controller; the values are given in Section 3.7.

2.10 Processor Restarts

When the processor is signalled via the SYSRESET_L or HALT_L pins, or when it detects severe corruption of its operating environment, it performs a restart process which saves some context in internal processor registers SAVPC and SAVPSL, changes to unmapped mode, and begins execution in the system ROM at address 2004.0000h. Bits <13:8> of SAVPSL contain a *restart code* which indicates the cause of the restart. The VS4000 VLC console firmware determines which restarts are delivered to the operating system and which are displayed on the operator's console. The restart codes (in hex) are:

Table 2–2: CPU Restart Codes

Code	Reason
2	HALT asserted (see Section 2.10.2 below)
3	Power on
4	Interrupt stack not valid during exception
5	Machine check normal exception
6	HALT instruction executed in kernel mode
7	SCB vector bits <1:0> = 11
8	SCB vector bits <1:0> = 10
A	CHMx executed while on interrupt stack
10	ACV or TNV during machine check exception
11	ACV or TNV during kernel stack not valid exception
12	machine check during machine check exception
13	machine check during kernel stack not valid exception
19	PSL<26:24> = 101 during interrupt or exception
1A	PSL<26:24> = 110 during interrupt or exception
1B	PSL<26:24> = 111 during interrupt or exception
1D	PSL<26:24> = 101 during REI
1E	PSL<26:24> = 110 during REI
1F	PSL<26:24> = 111 during REI

The restart process sets the state of the chip as follows:

SAVPC	saved PC
SAVPSL	saved PSL<31:16, 7:0> in <31:16, 7:0> saved MAPEN<0> in <15> valid stack flag in <14> saved restart code in <13:8>
SP	interrupt stack pointer
PSL	041F0000 (hex)
PC	2004 0000 (hex)
MAPEN	0
SISR	0 (power-on only)
ASTLVL	4 (power-on only)
ICCS	0 (power-on only)
MSER	0 (power-on only)
CADR	0 (power-on only)
	all else undefined

2.10.1 Power-on Restart

When system power is applied, the processor is signalled via the SYSRESET_L pin. It performs a processor restart with a restart code of 3.

2.10.2 HALT Restarts

When the CPU is signalled via its HALT_L pin, it performs a processor restart with a restart code of 2. Two things can generate such a signal:

- – pressing the operator's CONSOLE MODE button
- – receipt of a BREAK signal from serial line 3 when the Operator's ALTERNATE CONSOLE switch has been set to select this line as the console device.

Upon receipt of a HALT signal, the system enters console mode. Operation in this mode is controlled by the firmware in the main board ROM, which is described in the KA48 System Firmware Specification.

2.11 Cache General Description

This section describes the operation of the CPU's cache. The VS4000 VLC differs from the KA44 System in that it has ONLY a primary cache.

2.11.1 Organization and Addressing

Each cache *set* consists of an array of *entries*. Each entry contains a *valid* flag, a *tag*, and a *data* block. There are a total of eight possible sets making up the CPU internal cache.

Figure 2–5: Cache Organization

```
                                               b
                V    TAG: t bits        DATA: 2  bytes
              +---+---------------+------------------------+
            / |   |               |                        |
            | |   |               |                        |
            | |   |               |                        |
 e          | |   |               |                        |
2  entries <  |   |               |                        |
            | |   |               |                        |
            | |___|_______________|________________________|
            | |___|_______________|________________________|<-+
            \ |   |               |                        |  |
              +---+---------------+------------------------+  |
                                                              |
                                            one cache entry---+
```

The data block holds 8 bytes (*b*=3) which are a copy of an aligned quadword in main memory. In each cache set, there is just one entry in which a copy of a given main memory quadword can be found. That entry is selected by an *e*-bit entry index which is extracted from the quadword's physical address. However, there are numerous quadwords which could occupy one cache entry, so they are distinguished by a *t*-bit tag which is also extracted from the quadword's physical address. The breakdown of the 30-bit VAX physical address is shown below:

Figure 2–6: General Cache Address Mapping

```
  2 2
  9 8                                       3 2      0
 +-+--------------------------+---------------+--------+
 |0|          t bits          |     e bits    | b bits |
 +-+--------------------------+---------------+--------+
```

Starting from bit 0, the value of *b* depends on the size of the data block, which in this system is 8 bytes, so *b* is 3 - (2^3 = 8). The values of *e* and *t* depend upon how many entries are in the set. For this CPU cache, there are 128 entries in each set, so *e*=7 and *t*=19.

Each entry's valid flag *V* indicates whether the entry contains a valid copy of data from main memory. The hardware automatically sets the valid flag whenever a processor read cycle stores data in the entry. All the valid flags in the caches must be cleared under program control before the caches are enabled after a period of processor operation with the caches disabled, since main memory data may have been changed during this period.

2.11.2 Control Registers

There are four registers associated with the memory system. The two associated with the cache control are the Cache Control Register which sets the mode of operation of the cache, is used in flushing the cache and provides an overall enable/disable function for the cache and the Bank-Enable Hit/Miss Register (BEHR) which enables/disables individual cache sets and provides information about the last cache cycle.

2.11.2.1 Cache Control (CCR)

Figure 2–7: Cache Control Register(CCR)- IPR 25h

```
  3                                               0 0 0 0 0
  1                                               4 3 2 1 0
 +-------------------------------------------------+-+-+-+-+
 |                                                 |C|E|F|D|
 |                       MBZ                       |W|N|L|I|
 |                                                 |P|A|U|A|
 +-------------------------------------------------+-+-+-+-+
```

<31:04> Not used. Read as zeros; ignored on write.

<03> Compare Wrong Parity. This bit is read/write and is for diagnostic use. When set, it inverts the sense of the data parity bits accessed from the cache. This will cause a parity error to be indicated and so may be used to check the parity generate/check logic associated with the cache. This bit does not affect Tag parity.

<02> Enable Cache. This bit is read/write. When set, normal operation of the cache is enabled and both I and D-Stream references will be stored in those banks of the cache selected by Enable_Bank<7:0> of the BEHR. This bit is cleared by SYSRESET_L.

<01> Flush_Cache. This bit is write-only and reads always as zero. Writing a one to this bit clears all valid bits in all the cache tag arrays.

<00> Diagnostic Mode. This bit is read-write. When it is set, the cache and BEHR register may be accessed via a region in I/O address space; when clear, references to that same region of I/O address space become normal external bus cycles.

Table 2–3: Diagnostic Mode Cache Address Ranges

BEHR	2015.0800 - 2015.0FFF
Cache Data	2015.0400 - 2015.07FF
Cache Tags	2015.0000 - 2015.03FF

When diagnostic mode is set, writes to the cache data addresses will write to the cache data longword indexed by address bits <09:02>. All banks that have the BEHR Enable_Bank bit set will be written. Reads will return data from the highest priority bank if more than one bank is enabled.

Writes to cache tag addresses write to the cache tag indexed by bits <09:03> of the address. All banks that have their Enable_Bank bit set in the BEHR will be written. The format of the data written is :-

Figure 2–8: Write Format, Tag Diagnostic Mode

```
 3 3 2 2                                        1 0           0
 1 0 9 8                                        0 9           0
+-+-+-+------------------------------------------+-------------+
|P|V|X|                 TAG                      |      X      |
+-+-+-+------------------------------------------+-------------+
 | |
 | +------ VALID Bit
 +-------- Tag Parity
```

Reads from cache tag addresses read the cache tag addresses plus the data parity from the data longword addressed by bits<09:02> of the address as shown. If more than one Enable_ Bank bit is set, the data returned will be from the highest priority bank. (bank 0 is highest priority; 7 the lowest)

Figure 2–9: Read format, Tag Diagnostic Mode

```
 3 3 2 2                                       1 0    0 0    0
 1 0 9 8                                       0 9    4 3    0
+-+-+-+-----------------------------------------+------+-------+
|P|V|X|                TAG                      |   X  | DP    |
+-+-+-+-----------------------------------------+--------------+
 | |
 | +------ VALID Bit
 +-------- Tag Parity
```

2.11.2.2 Bank Enable/Hit Miss Register

The BEHR register is accessible as a longword ONLY, single bytes may not be written. It may be accessed at any of the longword addresses in the range 2015.0800:2015.0FFF when the CCR Diagnostic bit is set.

Figure 2–10: Bank Enable/Hit Miss Register(BEHR)

```
   3                    1 1              0 0 0 0 0 0 0 0 0
   1                    6 5              8 7 6 5 4 3 2 1 0
  +-------- ...----------+----------------+-+-+-+-+-+-+-+-+
  |         MBZ          |       Bank     |     Bank      |
  |                      |     Hit/Miss   |    Enable     |
  +-------- ...----------+----------------+-+-+-+-+-+-+-+-+
```

<31:16> MBZ. These bits are ignored on write and read always as zero.

<15:08> BANK_HIT<7:0> These bits are read-only and are provided for cache testing purposes. They reflect the Hit/Miss status of each of the banks of the cache for the most recent D-stream read or write cycle. A "1" indicates that there was a HIT in the corresponding bank of the cache, a "0" a MISS. These bits are cleared when SYSRESET_L is asserted.

<07:00> ENABLE_BANK<7:0>. These bits are read/write and when set enable the respective banks of the cache. Note that the SOC cpu may have less than eight banks of cache functional and at chip test will have had those non-functional banks permanently disabled by lasar fusing. To determine how many and which banks of cache are functional on a particular chip, the ENABLE_BANK bits should all be set by writing FFh to the BEHR register. The BEHR register should then be read and those bits of the ENABLE_BANK field that return ones will indicate those banks of the cache that are actually enabled and useable. Whenever a bank is enabled or disabled, the cache should be flushed by writing a one to CCR<FLUSH>. ENABLE_BANK<7:0> are cleared when SYSRESET_L is asserted.

2.11.2.3 Memory System Error Register (MSER) - IPR 39h

This register is used to report information about the external CDAL bus and cache errors. Two types of errors are reported: those which do not cause an immediate problem in the operation of the CPU chip - these post an interrupt request; those which do cause an immediate problem and would disrupt the current instruction execution - these cause a machine check (trap via SCBB+4). The classification of errors into each type is discussed below.

Figure 2–11: Memory System Error Register(MSER)

```
 3                                       0 0 0 0 0 0 0 0
 1                                       7 6 5 4 3 2 1 0
+-----------------------------------------+-+-+-+-+-+-+-+
|                                         |B|C|D|T|T|T|I|
|         .... MBZ ...                    |E|P|P|P|P|P|N|
|                                         |R|E|E|E|2|1|T|
+-----------------------------------------+-+-+-+-+-+-+-+
```

<00> Read-Write, INTERRUPT. This bit is set by the occurance of one of the interrupt type errors. Its setting causes an interrupt at IPL 1A and disables the cache. (Note that the ENABLE_CACHE bit in the CCR register will remain set). MSER<6:3> will indicate which error has occurred. Once this bit is set, MSER<6:3> will only be overwritten by TRAP type errors, subsequent INTERRUPT type errors will be lost. This bit is cleared by writing a one to this bit position on an IPR write to the MSER and by RESET.

<01> Read-Write, TRAP1. This bit is set by the occurance of one of the trap type errors. Its setting causes a machine check abort and disables the cache. (Note that the ENABLE_CACHE bit in the CCR register will remain set). MSER<6:3> will indicate which error has occurred. If bit<0>, INTERRUPT, was already set, this error will overwrite MSER<6:3>. This bit is cleared by writing a one to this bit position on an IPR write to the MSER and by RESET.

<02> Read-Write, TRAP2. This bit is set by the occurance of one of the trap type errors, if TRAP1 has already been set and indicates a nested error. When this bit set, MSER<6:3> will indicate the error that had occurred when TRAP1 was set. The result of this bit being set is a machine check. This bit is cleared by writing a one to this bit position on an IPR write to the MSER and by RESET.

<03> Read-only, Tag Parity Error. This bit is set when incorrect atg parity is detected in an enabled bank of the cache during a cache read or write operation. For D-stream reads that encounter this error, the result is a machine check, for all other cache cycles where the error occurs, the rsult is an interrupt.

<04> Read-only, DAL Data Parity Error. This bit is set when incorrect DAL parity is detected on an external read cycle. For the first cycle of a D-stream read, it results in a machine check; for the second cycle of a D-stream read or an I-stream read, it results in an interrupt.

<05> Read-only, Cache Data Parity Error. This bit is set when a cache data parity error is detected. On a D-stream error, the result is a machine check, for an I-stream error the result is an interrupt.

<06> Read-only, Bus Error. This bit will be never be set for the VS4000 VLC system as the Error/Retry feature of the CPU is not used.

<31:07> Read-only, always return zero.

2.11.3 Diagnostic Operation

Refer to SOC chip spec 3.8.5

2.11.4 Cache Initialization

Before enabling the first-level cache after power-on or after operation with the cache disabled, all the valid flags in the cache must be cleared to ensure that no stale data will be read from the cache. This may be done by setting cache diagnostic mode (CCR<00> = 1), setting all the bank enable bits in the BEHR register and then writing all the tags in parallel with V=0.

Chapter 3

INTERRUPT CONTROLLER

The interrupt controller is a part of the DC7201 and performs two separate functions.

- It receives up to eight interrupt request signals from the system's I/O devices, synchronises and latches them. The latched requests are then masked by individual Enable bits and the results ORed to form a single interrupt, presented to CPU on IRQ1 L, Interrupt Priority Level 15h.
- It recognises CPU Interrupt Acknowledge Cycles for IPL 14h, 15h, 16h and 17h and generates a ROM cycle to retrieve an appropriate vector.

3.1 IPL15h Interrupt

The eight interrupt request sources that result in an interrupt at IPL15h have associated with them three registers :-

INT_REQ holds the latched interrupt requests received from I/O devices (read-only).

INT_MSK contains a mask which determines which interrupt requests will generate a processor interrupt (read/write).

INT_CLR enables a program to selectively reset interrupt request bits in the INT_REQ register (write-only).

The bits in each of these registers are uniformly associated with interrupt sources according to the numbering given in Section 3.3. For example, bit 0 in each register is associated with the SCSI controller interrupt.

The eight latches which comprise the INT_REQ register are edge-triggered: each latch records an interrupt request from an I/O device when the device's interrupt request signal changes from false to true. The contents of the INT_REQ register are ANDed with those of the INT_MSK register, the results ORed to form an interrupt request to the CPU. The resulting interrupt request is presented to the CPU as IRQ1 L.

When the CPU acknowledges an interrupt request, the interrupt controller sends to it the interrupt vector associated with the highest-numbered bit which is set in both the INT_REQ and INT_MSK registers and clears that bit in the INT_REQ register (the INT_MSK bit is not affected). The interrupt vector values are listed in Section 3.7.

Any set bit in the INT_REQ register can also be cleared by a writing a byte to the INT_CLR register which has a one in the corresponding bit position. (This is a transient operation; the data written is not stored and does not prevent the INT_REQ bit from being set in the future). This enables a program to handle a device in non-interrupt mode by clearing the device's INT_MSK bit, polling its requests by reading the INT_REQ register, and clearing its requests by writing to the INT_CLR register.

Since the bits of the INT_REQ register are edge sensitive rather than level sensitive, a program which responds to a request from a device (either in an interrupt service routine or by polling the INT_REQ register) and clears its bit in INT_REQ must do whatever is necessary to return that device's interrupt request signal to its false state in order that a subsequent request from the device will be able to set its bit in INT_REQ again.

Upon power-on, all bits in the INT_REQ and INT_MSK registers are cleared.

3.2 INTACK Cycles

For IPL 15h, the DC7201 generates a three bit number from the eight possible interrupt requests, using a fixed priority encoding, shifts this number 2 places to the left to form a number in the range <00:1C>h, then ORs this with the value 2004.0020h and presents the result - a number now in the range <2004.0020:2004.003C>h to the system ROM as a longword aligned address. The data which is retrieved from ROM is returned to the CPU to be added to the SCCB to form the interrupt vector specific to the highest priority interrupt currently active.
For IPL14h, 16h and 17h, a fixed address is presented to the system ROM from which a single interrupt vector for each level is retrieved.

IPL	ROM Address (HEX)
14	2004.0040
15	2004.0020:2004.003C
16	2004.0048
17	2004.004C

3.3 Interrupt Sources and Ranking

The eight interrupt sources that result in an IPL15h interrupt are listed in the following table. The interrupt numbers 7:0 indicate their bit positions in the INT_xxx registers and their relative priority when more than one request is pending; 7 is the highest priority.

Interrupts 0, 1, 3. 4, 5 and 6 are dedicated to devices on the system board. Interrupts 2 and 7 come from optional devices.

Table 3–1: Interrupt Signal Sources

Num	Name	Source
7	SR	unused
6	ST	79C30 controller request for service
5	AR	Async. line receiver done or silo full
4	AT	Async. line transmit done
3	G1	Graphics
2	G0	Graphics
1	NI	Network Controller
0	SC	Storage controller

3.4 Interrupt Request Register (INT_REQ)

The interrupt request register is an 8-bit read-only register at physical address 2008.000Fh each of whose bits reflects the state of the interrupt request latch for one interrupt source. Bits<7:0> correspond to interrupt numbers <7:0> as listed in Section 3.3.

Figure 3–1: Interrupt Request Register (INTREQ)

```
INT    7       6       5       4       3       2       1       0
   +-------+-------+-------+-------+-------+-------+-------+-------+
   |  NA   |  SO   |  AR   |  AT   |  G1   |  G0   |  NI   |  SC   |
   +-------+-------+-------+-------+-------+-------+-------+-------+
```

A bit in the INT_REQ register is set only by an active transition on the corresponding device's interrupt request line. The bit will be set by an active transition regardless of the state of the corresponding bit in the interrupt mask register, INT_MSK. However, an interrupt request is sent to the CPU only when the corresponding bits in both INT_REQ and INT_MSK are set.

A bit in the INT_REQ register is cleared either by a program which writes to the INT_CLR register with a one in the corresponding bit position, or by a CPU interrupt acknowledge cycle during which the bit is the highest-numbered bit in INT_REQ which is set and whose corresponding bit in the interrupt mask register INT_MSK is also set.

INT_REQ may be read at any time; reading it does not alter the state of the system in any way.

Upon power-on, the interrupt request register is cleared to zero.

3.5 Interrupt Clear Register (INT_CLR)

The interrupt clear register is an 8-bit write-only register at physical address 2008.000Fh which is used to selectively clear bits in the interrupt request register INT_REQ. Bits 7:0 correspond to interrupt numbers 7:0 as listed in Section 3.3.

Figure 3–2: Interrupt Clear Register (INT_CLR)

```
INT    7       6       5       4       3       2       1       0
   +-------+-------+-------+-------+-------+-------+-------+-------+
   |  NA   |  SO   |  AR   |  AT   |  G1   |  G0   |  NI   |  SC   |
   +-------+-------+-------+-------+-------+-------+-------+-------+
```

For each bit of INT_CLR which is a one, the corresponding bit of INT_REQ is cleared. For each bit of INT_CLR which is a zero, the corresponding bit of INT_REQ is not changed. The effect of writing to INT_CLR is transient; its contents are not stored and writing to it does not prevent any INTREQ bits from being set in the future.

WARNING

As INT_REQ and INT_CLR share the same address, use of a Read/Modify/Write instruction when accessing the INT_CLR register could cause loss of interrupts.

3.6 Interrupt Mask Register (INT_MSK)

The interrupt mask register is an 8-bit read/write register at physical address 2008.000Ch each of whose bits is a mask for one interrupt source. Bits 7:0 correspond to interrupt numbers 7:0 as listed in Section 3.3. Each mask bit is ANDed with the corresponding bit of the INT_REQ register before being input to a priority encoder, the output of which determines which bit in INT_REQ will be cleared (if more than one bit is set) when the CPU executes an Interrupt Acknowledge Cycle.

Figure 3–3: Interrupt Mask Register (INT_MSK)

```
MSK    7       6       5       4       3       2       1       0
   +-------+-------+-------+-------+-------+-------+-------+-------+
   |  NA   |  SO   |  AR   |  AT   |  G1   |  G0   |  NI   |  SC   |
   +-------+-------+-------+-------+-------+-------+-------+-------+
```

Note that a zero in a mask register bit does not prevent the corresponding device from setting its interrupt request register bit. If a request bit is set whose corresponding mask bit is zero, a CPU interrupt is not requested until the mask bit is subsequently set to one (assuming that the request bit has not meanwhile been cleared by writing to INT_CLR). A program which is changing from polled to interrupt servicing of a device should be sure to clear the device's bit in INT_REQ prior to setting its bit in INT_MSK in order to avoid a possible false interrupt signal to the CPU.

Upon power-on, the interrupt mask register is cleared to zero.

3.7 Interrupt Vector Generation

When the CPU acknowledges an interrupt from the interrupt controller, the interrupt controller causes a vector number to be placed on the CDAL bus which corresponds to the highest priority pending interrupt. It obtains this vector number from reserved locations in the System Board ROM, Chapter 5, Section 5.1.
The vector presented to the CPU in an INTACK cycle has the following format

Figure 3–4: Interrupt Vector Longword

```
    3                                   1 0                  0 0 0
    1                                   0 9                  2 1 0
   +-------------------------------------+-+------------------+-+-+
   |              ...0...                |1|     VNUM         |0|P|
   +-------------------------------------+-+------------------+-+-+
```

<31:10> Zero.

<09> Must be one.

<08:02> VNUM. Interrupt vector number which is multiplied by 4 to form an offset to a vector position in the current SCB. Since only vectors in the range 200h through 3FCh should be used for I/O devices, bits 15:10 are zero and bit 9 is a one.

<01> Must be zero.

<00> P. Priority level flag which selects the IPL to which the processor is raised when it acknowledges the interrupt. If this bit is zero, the IPL will be the interrupting IPL; if it is one, the IPL will be 17h. The normal setting is zero.

The conventional vector values established by the system ROM firmware for the various IPLs are as follows (the value in the Vect column represents bits 15:0 of the longword; the value in the P column is then placed in bit 0 of the longword):

Table 3–2: Interrupt Vectors

IPL	No	Name	Vect	P	Source
15	7				Unused
15	6	ST	02C4	0	79C30 request for service
15	5	AR	0250	0	Asynchronous serial line controller receiver done or silo full
15	4	AT	0254	0	Asynchronous serial line controller transmit done
15	3	G1	0244	0	Graphics interrupt 1
15	2	G0	0248	0	Graphics interrupt 0
15	1	NI	03F8	0	Network controller
15	0	SC	03FC	0	Storage controller
14			3F4	0	Unassigned
16			3F0	0	Unassigned
17			3EC	0	Unassigned

Note

The vectors listed above are subject to change; the System ROM Specification should be consulted for current values.

3.7.1 Passive Release

A special case occurs when the CPU executes an Interrupt Acknowledge Cycle and no interrupt is pending (INT_REQ<7:0> = 00000000). This case can only occur when for some reason a write has occurred to INT_CLR, with the interrupt system enabled, between the time that the interrupt controller has asserted IRQ1 L and the CPU initiates the Interrupt Acknowledge cycle. For this case, the interrupt controller does not access the System Board ROM, but internally generates a vector of all zeros.

Chapter 4

MAIN MEMORY

The memory controller allows for up to six memory SIMMs to be installed, in pairs, on the system module. The system module itself carries no memory. The SIMMs may be 256K x 36 or 1M x 36. All SIMM sockets must be populated with the same size SIMM. This gives possible memory capacities of 2, 4, 6, 8, 16 or 24 MBytes.

The six SIMM sockets have fixed addresses, depending upon the size of the SIMMs installed. For a small SIMM configuration, the first pair of SIMMs respond to addresses <0:1FFFFF>, the second pair to <200000:3FFFFF>, the third pair to <400000:5FFFFF>. For a large SIMM configuration, the first pair respond to addresses <0:7FFFFF>), the second pair to addresses <800000:FFFFFF> and the third pair to <1000000:17FFFFF>.

The DC7201 arbitrates between and services requests for main memory cycles from several sources; the CPU, the Ethernet Controller (NI), the Storage Controller (SCSI) and the Graphics Controller (GC) section of the memory controller. To optimize use of the available RAM bandwidth, data to/from these several requestors is buffered within the DC7201. To further minimize interaction between the requestors, the DC7201 has three data busses; the CDAL which connects it to the CPU; the EDAL which connects it to the NI, SCSI and Invalidate Filter and the MDAL which connects to the memory system, including the Video RAMS which make up the integral frame buffer.

The memory controller is capable of performing several types of RAM cycles; longword, quadword and octaword. Buffering between the several requestors and the memory allows these various cycles to be used in a way that makes best use of the available memory bandwidth. The sections that follow will discuss each requestor and the cycles generated.

The NI and SCSI controllers are DMA devices; the GC can generate addresses independently. Thus all three of these devices may attempt to write to memory locations that are currently cached. To maintain cache coherency it is necessary that for all writes requested by any of these devices, the CPU cache be checked and, if necessary, an entry invalidated. This could impose a significant load on the CPU to check the potential invalidates. For this reason the DC7201 controls a separate invalidate filter (backmap) which maintains a copy of the CPU's cache tags. In this way only those writes that do require a cache invalidate will disturb the CPU.

To allow greater flexibility, the two DMA devices access memory via a translation map contained in memory - see Section 4.4. The GC address generator performs virtual access to memory via the page tables - refer to the PVAX2 Graphics Controller Specification for further information.

4.1 Main Memory Requests

Main memory requests have a fixed priority, as follows (highest to lowest) :

- GC - shift register load
- GC - cursor buffer load
- Refresh
- NI Controller
- CPU
- SCSI Controller
- GC Address Generator

Refer to the PVAX2 Graphics Controller Specification for details of all GC functions.

4.1.1 NI Controller

The NI Controller is one of the requestors connected to the EDAL bus. It contends with the SCSI controller and the Invalidate Filter for access. The DC7201 maintains input and output buffers for the 16-bit wide data received from and to be sent to the NI Controller.

For NI controller data transfers to the DC7201, the 16-bit data words are accumulated within the DC7201 until a quadword address boundary is crossed or until the address of the next received data is seen to be no longer in sequence. This accumulated data is then written to memory in a single operation. For transfers that start at a quadword aligned address, the write to memory will be a page-mode quadword cycle. For transfers that start at a longword aligned address, the first write to memory will be a longword write of four bytes. For transfers that start at a non-longword aligned word address, the first write to memory will be a longword write to bytes <3:2> only. In either of these two non-quadword aligned cases, after this initial alignment write has occurred, all subsequent sequential transfers will cause page-mode quadword cycles. An interrupt from the NI Controller or a timeout - no data transfer from the NI Controller to the DC7201 within <<<tbd microseconds>>> will cause any accumulated data to be written to memory.

Requests from the NI Controller for data from memory that present a quadword aligned address will cause the DC7201 to retrieve a quadword of data from memory. The three additional 16-bit words read in addition to the word actually requested by the NI Controller remain buffered within the DC7201 to satisfy the (assumed) next three read requests from the NI controller. For non-quadword aligned address requests the DC7201 retrieves the longword of data that includes that actual word requested. Depending on the actual address alignment, the second word of this data may be used to fill the next NI Controller request or may be discarded.

This buffer/lookahead read mechanism is totally transparent to software, the operating system driver need have no knowledge that this is happening, but should be aware that progem loops that rely on exact timings may show inconsistencies.

All memory write requests that originate from the NI Controller are passed to the invalidate filter logic as the write occurs to memory - see Section 4.3.

4.1.2 SCSI Controller

The SCSI Controller is one of the requestors connected to the EDAL bus. It contends with the NI controller and the Invalidate Filter for access. The DC7201 maintains input and output buffers for the 16-bit wide data received from and to be sent to the SCSI Controller.

For SCSI controller data transfers to the DC7201, the 16-bit data words are accumulated until a quadword address boundary is crossed or until the address of the next received data is seen to be no longer in sequence. This accumulated data is then written to memory in a single operation. For transfers that start at a quadword aligned address, the write to memory will be a page-mode quadword cycle. For transfers that start at a longword aligned address, the first write to memory will be a longword write of all four bytes. For transfers that start at a non-longword aligned word address, the first write to memory will be a longword write to bytes <3:2> only. For transfers that start at a non-word aligned byte address the first write to memory will be a longword write to byte<3> only. For all of these non-quadword aligned cases, after this initial alignment write has occurred, all subsequent sequential transfers will cause page-mode quadword cycles. An interrupt from the SCSI Controller or a timeout - no data transfer from the SCSI Controller to the DC7201 within <<<tbd microseconds>>> will cause any accumulated data to be written to memory.

Requests from the SCSI Controller for data from memory that have a quadword aligned address will cause the DC7201 to retrieve a quadword of data from memory. The three additional words read in addition to the word actually requested by the SCSI Controller remain buffered within the DC7201 to satisfy the (assumed) next three read requests from the SCSI controller. For non-quadword aligned address requests the DC7201 retrieves the longword of data that includes that actual word requested. Depending on the actual address alignment, the remainder of this data may be used to fill the next SCSI Controller request(s) or may be discarded.

This buffer/lookahead read mechanism is totally transparent to software, the operating system driver need have no knowledge that this is happening.

All memory write requests that originate from the SCSI Controller are passed to the invalidate filter logic as the write occurs to memory - see Section 4.3.

4.1.3 CPU Memory Requests

The CPU can request longword or quadword read cycles and up to longword (specified by a four-bit byte mask) write cycles. As the CPU's internal cache is write-through, all CPU generated writes appear immediately as write requests to the DC7201. The DC7201 contains a two quadword write buffer for CPU write requests so that longword write requests to sequential longword addresses may be passed to the memory controller as quadword write requests to take advantage of the faster page-mode cycles possible when writes are performed to sequential addresses within the same page.

4.2 Write Buffer and Read Requests

Data may be retained within the DC7201 write buffer for some considerable time if writes are infrequent. For this reason the DC7201 allows read requests to be executed without flushing the write buffer. On any CPU generated read request when the write buffer is not empty, the DC7201 first compares the requested address with the address(es) of the data in the write buffer; if a match occurs, the data is returned to the CPU from the write buffer altough, for control simplicity, a memory read is still executed; the read data being ignored. If only a part of the requested data is found in the write buffer, the memory read supplies the missing bytes, the write buffer contents are not altered. If the requested data is not found in the write buffer, memory data is supplied to the CPU.

4.2.1 Write Buffer Flushing

Certain operations require that the data in the DC7201 write buffer be written back to memory before these operations are allowed to occur. The DC7201 makes this determination and ignores any CPU-generated *Clear Write Buffer* cycles.

The DC7201 will write back all data contained in its write buffer when it receives a request for a read or write to any address in I/O space (address bit <29> = 1) or when it recognises an INTACK Cycle. In both these cases, the data in the write buffer is first written to memory, then the I/O read or write occurs or the ROM read (INTACK) occurs. The result at the end of either operation is that the write buffer is empty.

4.3 Invalidate Filter

The DC7201 controls two 2K x 8 SRAMs where a copy of the CPU tag store is maintained. On CPU cache allocates, the tag value is written into this RAM. As the CPU cache is multiple set, it is necessary to maintain up to eight separate tag values for any given Index in this RAM. On any write generated by the NI, SCSI or GC, the eight entries are checked for a possible invalidate request to be passed back to the CPU. Only if a match is found will be CPU be disturbed and the entry in the invalidate filter SRAM be cleared. CPU invalidates are requested by assertion of the CPU DMA Request line and then executing a quadword or octaword fast invalidate cycle.

4.3.1 Invalidate Filter Cache Set Selection - CAC_LIM

The CPU may have from one to eight cache sets active. To minimize the overhead of the Invalidate Filter lookup operation when less than a full eight sets are active, a limit register is provided which may be loaded with the maximum set number. This register, CAC_LIM, is accessible at address 2008.0018h. Only bits <2:0> have significance and specifiy the maximum cache set number directly. The value to be loaded into this register is established by initially loading the CPU Bank Enable register, see Chapter 2, Section 2.11.2.2, then reading that same register. The bit pattern returned in bits<7:0> indicates which cache sets are active and enabled. This bit field should then be converted into a three bit number and loaded into CAC_LIM.

CAC_LIM is initialized on power-on to <111> to select all eight cache sets.

4.3.2 Invalidate Filter Initialization

Prior to enabling the CPU cache, the Invalidate Filter must be initialized so that it represents a copy of the initial CPU cache state, i.e. no entries cached. This is done by accessing the Invalidate Filter RAMs via a range of I/O addresses, 2020.0000:2020.0FFCh. Writes to sequential longword addresses in this range with data of zero will initialize the filter RAMs.

Following intialization, the Filter should be enabled by writing to bit <FILT_ENA> in the BWF0 Register - see Chapter 6, Section 6.6.

4.4 DMA Mapping

The NI and SCSI controllers access memory via a translation table which is contained in main memory. A Map Base Address Register - MAP_BASE - within the DC7201 points to the beginning of this reserved region of memory. The 32,768 longwords extending upwards from MAP_BASE provide translations for the page address supplied by either DMA device. Each DMA device has a two-entry cache of current translations (one for read, one for write) kept within the DC7201. It is the responsibility of the operating system to allocate entries for each DMA device in this common translation table.

Figure 4–1: Map Base Register MAP_BASE - Address 2008.0008

```
 3           2 2           1 1                           0
 1           5 4           7 6                           0
+-------------+-------------+-----------------------------+
|     IGN     |     BASE    |            IGN              |
+-------------+-------------+-----------------------------+
```

<1:25> IGN. Ignored on write, read as zero.

<24:17> BASE<7:0>. Specifies the start of a 32 K longword region in main memory. The longwords specified by this address and the 32,767 longwords that follow contain translation entries for DMA devices.

<15:00> IGN. Ignored on write, read as zero.

4.4.1 Translation Table Entry Format

Although the space allocated for the DMA translation table is 32,768 longwords, the actual space used is a function of the starting addresses supplied to each of the DMA devices and the maximum transfer length allowed.

Any of the 32,768 longwords allocated for DMA translation entries must be written in the following format prior to initiating any DMA transfer :

Figure 4–2: Translation Entry Format

```
 3 3                       1 1                                0
 1 0                       8 7                                0
+-+-------------------------+----------------------------------+
|V|           IGN           |          PAGE NUMBER             |
+-+-------------------------+----------------------------------+
```

<31> Valid Bit. Indicates to the DMA translation hardware within the DC7201 that this location contains a valid page number for use in translation. The operating system must set this bit for all translation table entries that it expects to be used.

<30:18> IGN - these bits are ignored.

<17:00> Page Number, the longword aligned start address of a 512 byte region of memory allocated as a part of a buffer for one of the DMA devices. The eighteen bits specified in this field allow for a maximum memory size of 128 MBytes.

4.4.2 Translation

Each DMA controller has its own 24-bit address counter for DMA transfers that has a page field - 15-bits and an address-within-page field - 9-bits. When a DMA controller presents an address to the DC7201 to perform a DMA cycle to/from main memory, the DC7201 translates the address supplied, using MAP_BASE and the translation table contained in main memory as follows. Bits <23:09> of the address supplied by the DMA controller - the page field - are compared to the address value held in either the read or write translation cache (as appropriate) for that DMA device. If the addresses match and if that entry is marked as Valid, the associated page address held in a field of that cache entry is concatenated with the address-within-page field of the supplied address to form the actual address to be used and the DMA cycle proceeds.

If the address match fails, indicating that this DMA transfer is to an address on a different page from the last DMA transfer that this device initiated, bits <23:09> of the DMA address supplied are concatenated with bits <24:17> of MAP_BASE to form a new 23-bit longword aligned map register address. This address is then used to retrieve data from the translation table in main memory. Bits <15:00> of the data returned from memory are then concatenated with the original address-within-page bits supplied by the controller to form a 25-bit address that is the actual address to be accessed. Bits<15:00> that were retrieved from the translation table are stored as a new value in the appropriate translation cache associated with that DMA device and the Valid bit set for that entry.

Figure 4–3: DMA Address Translation - Translation Cache Hit

```
                                                       Byte
                        2 <-- Unmapped page -->  0 0  within  0
                        3     Address            9 8   Page   0
                       +--------------------------+------------+ DMA
                       |        15 bits           |  9 bits    | Device
                       +--------------------------+------------+ Address
                       |                          |            |
    ___________________|                         /|            |
   /                ____________________________/ |            |
  |                /                              |            |
  |               |                               |            |
 3|3             1|1 <--- Mapped Page --> 0       |            |
 1|0             6|5      Address         0       |            |
+-+---------------+-----------------------+       |            | Cached
|V|               |       16 bits         |   <--------------  | Entry
+-+---------------+-----------------------+       |            | in DC7201
                  |                       |       |            |
                  |                        \_____ |            |
                  |                              \|            |
                  |                               |            |
                  +----------------------+--------+------------+
                                         |
                                         V
                                Address to memory
```

Figure 4–4: DMA Address Translation - Translation Cache Miss

Figure 4–4 Cont’d on next page

Figure 4–4 (Cont.): DMA Address Translation - Translation Cache Miss

```
            2       1 1                  0
            4       7 6                  0
+----------+---------+--------------------+
|          |  8 bits |                    |  MAP_BASE
+----------+---------+--------------------+
           |         |     2                     0 0          0
           |         |     3                     9 8          0
           |         |    +-----------------------+------------+ DMA
           |         |    |       15 bits         |  9 bits    | Device
           |         |    +-----------------------+------------+ Address
           |         |    |                       |            |
           |         | ___/             _________/|            |
           |         |/                /          |            |
           |2       1|1               |           |            |
           |4       7|6              2|1 0        |            |
+----------+---------+----------------+-+-+       |            | Access Main
|          |         |                |   |       |            | Memory at
+----------+---------+----------------+-+-+       |            | this address
           |                          |           |            |
           +--------------+-----------+           |            |
     Map Register Address |                       |            |
  +-----------------------+                       |            |
  |      3                  1                     |            |
  |      1                  5                    0|            |
  |     +------------------+----------------------+            |
  +---->|       ZERO       |       16 bits        |            |
        +------------------+----------------------+            |
                           | Save this value in   |            |
                           | cache, Set Valid     |            |
                           |                      |            |
                           +-----------------+----+------------+
                                             |
                                             V
                                    Address to memory
```

4.4.3 Example

As noted above, the actual memory space that need be allocated for translation entries can be up to 32,768 longwords. In many cases, less than this may be used. Consider, for example, that the two DMA devices of the system are restricted to maximum transfers of 64 KBytes, with only a single buffer available for each device. This requires that only 104 pages be allocated for each device.

The defined bit positions in MAP_BASE imply that the translation buffer must be aligned on a 104 KByte address boundary, and indeed, if the full 24-bit addressing capability of the DMA devices is to used, this is necessary. However, in the example being considered, we need only to define 104 contiguous entries for each device, the translation table can be offset from a 104 KByte address boundary by suitably modifying the start address supplied to each DMA controller; remember that it is the concatenation of the page address of the DMA controller address and MAP_BASE that forms the address from which the actual page address will be retrieved.

e.g. MAP_BASE = 40000h; SCSI Controller Start address = 1000h

Extracting the page address from the SCSI Controller start address yields 1000 binary. According to the translation algorithm above, this is then shifted two places higher in significance and merged with MAP_BASE. This yields an initial translation table address of 40020h. The 104 translation entries now should be placed at sequential longword addresses 40020:401F0h.

In this way, for transfer lengths of less than that allowed by the full 24-bit addressing of the DMA devices, the translation table can be placed more or less anywhere in memory, indeed the table can consist of two non-contiguous segments, one for the SCSI controller, the other for the NI Controller, simply by supplying the appropriate starting address to each controller.

4.5 Alternate Access to Map Registers

The translation map may be referenced without knowledge of the contents of MAP_BASE via a reserved range of I/O addresses. This address range is 2000.0000 - 2001.FFFFh, the 32,768 map entries are referenced as longwords within this address range. Note that although the translation map is being referenced by I/O addresses, which do not normally have parity appended, correct parity will be written into any entry written using this alternate access mode.

4.6 Parity

The CPU generates parity on all CDAL operations that reference memory (Address bit<29> = <0>) and additionally checks it if bit <PEN> is set to a ONE in the BWF0 Register - see Chapter 6, Section 6.6.

Chapter 5

ROM MEMORY

The system board ROM contains processor restart, diagnostic and console code and the primary bootstrap program. (The contents of this ROM is detailed in the KA48 System Firmware specification.) Another small ROM is uniquely programmed for each system with its network address.

5.1 System Board ROM

The system board contains two 40-pin sockets for 64K by 16 EPROM chips which collectively hold 256 Kbytes of data. ROM data appears at physical addresses 2004.0000 through 2007.FFFFh. The data path to this ROM is 32 bits wide. Certain physical addresses in the ROM have fixed uses. These are:

2004.0000h — Processor restart address. The processor begins execution at this address in non-mapped mode when a processor restart occurs. (See section 2.10)

2004.0004h — System type register SYS_TYPE. The contents of this longword supplement the internal processor SID register to identify the processor and system type. (See section 2.5.2)

2004.0020h — Interrupt vector numbers. Eleven consecutive longwords starting at this address are automatically referenced by the hardware to supply the interrupt vector numbers for the interrupt sources connected to the interrupt controller. (See section 3.7)

5.2 Network Address ROM

A 32-byte ROM on the system board contains a unique network address for each system. Data from this ROM is read in the low-order bytes of 32 consecutive longwords at physical addresses 2009.0000h through 2009.007Ch. The network address occupies the first six bytes (addresses 2009.0000h through 2009.0014h). The byte at 2009.0000h is the first byte to be transmitted or received in an address field of an Ethernet packet; its low-order bit (bit 0) is transmitted or received first in the serial bit stream. Digital purchase specification A-PS-23365A1-0-0 describes this ROM in detail and discusses the checksum bytes which follow the six address bytes.

This ROM is installed in a socket so it can be moved in the event that a system's system board is replaced.

Chapter 6

MISCELLANEOUS I/O REGISTERS

This chapter describes several miscellaneous I/O registers. Their names and physical addresses are listed in the table below.

Table 6–1: Miscellaneous I/O Registers

Name	Address (HEX)	Function
IORESET	2002.0000	I/O Reset register. Used to generate a reset signal to certain I/O controllers.
CFGTST	2002.0000	Configuration and Test register. Indicates which options are present in a system.
HLTCOD1	2008.0000	Halt Code register 1. Used as a temporary storage location by system firmware during a processor restart.
HLTCOD2	2008.0004	Halt Code register 2. Used as a temporary storage location by system firmware during a processor restart.
DIAGDISP	2008.0010	Diagnostic Display Register. Eight LEDs, turned ON or OFF by the setting of bits<7:0> of this address; a ONE lights the corresponding LED, a ZERO extinguishes it. Power-on turns ON all the LEDS.
BWF0	2008.0014	Miscellaneous Register 0. This contains many miscellaneous control bits that affect system operation.
DIAGTIMU	2008.001C	High resolution Diagnostic Timer register. A one microsecond time counter.
DIAGTIMM	2008.001E	Low resolution Diagnostic Timer register. A one millisecond time counter.

6.1 IO Reset Register (IORESET)

The IO reset register (IORESET) is a write-only byte register at physical address 2002.0000h. Any write access to this register (the data value is ignored) generates a reset signal to the following:

Storage Controller, Chapter 9
Network Controller, Chapter 8

The indicated chapters should be consulted for details of the effects of writing to IORESET. Note that the processor, FPA, interrupt controller, and serial line controller are *not* affected by IORESET.

The minimum duration of the reset signal is 700 nsec.

6.2 Configuration and Test Register (CFGTST)

The configuration and test register (CFGTST) is a 16-bit read-only register at physical address 2002.0000h.

Figure 6–1: Configuration and Test Register (CFGTST)

```
     15      14      13      12      11      10      9       8
  +------+-----------------------------------------------+-------+--
  | RES  |    0 .................................... 0   | ALCON |
  +------+-----------------------------------------------+-------+-

      7       6       5       4       3      2       1       0
 -+-------+--------+---------------+---------------------+--------+
  |MONTYP |    1   |  0 ..... 0    |       MSIZE         |  MTYP  |
--+-------+--------+---------------+---------------------+--------+
```

<15> Reserved. Reads as ONE.

<14:09> READ ONLY, read as ZERO, ignored on write.

<08> ALCON. Alternate Console. When Switch 2 on the VS4000 VLC System Board is set, allowing serial line 3, the printer port, to be used as the diagnostic console, this bit is reported as a one. When the switch is off, this bit is reported as a zero.

<07> READ ONLY, reads as ONE, ignored on write.

<06> MONTYP. Indicates for which monitor type the attached frame buffer module is set. For the two frame buffer modules currently defined, has the following meaning :
54-20772 : 1 = 72Hz; 0 = 60Hz.
54-20774 : 1 = 1024 x 768; 0 = VGA.
SeeSection 6.3

<05:04> READ ONLY, read as ZERO, ignored on write.

<03:01> MSIZE. Each bit indicates the presence or absence of one of the three possible pairs of SIMMs that make up system memory. The six SIMM sockets have fixed physical addresses assigned. The SIMMs **must** be installed in pairs, in adjacent positions starting with the socket whose populated bit is read as bit<01> of this register. SIMM sockets 1, 3 and 5 are the ones whose "SIMM present" bit is read in this field of this register. Each bit that reads as a one indicates the presence of a pair of SIMMs - i.e. if a SIMM is found presnt in SIMM socket 1, it is assumed that there is also a SIMM present in socket 2 etc. Note that it is the System ROM code's responsibility to verify that SIMMs have, indeed, been inserted in pairs.

<00> MTYP. This bit indicates whether 1MB or 4MB SIMMs are installed in the module. A ONE indicates 1MB SIMMs, a ZERO 4MB parts. Note that this bit reflects what is installed in SIMM socket 1 and the assumption is made that all sockets contain the same type of SIMM.

Table 6–2: Memory Sizes

MSIZE	MTYP	Memory Size
001	0	8 MB
011	0	16 MB
111	0	24 MB
001	1	2 MB
011	1	4 MB
111	1	6 MB

WARNING

The CFGTST register shares its physical address with the IORESET register (section 6.1, above). Programs must take care not to attempt to write to the CFGTST register, since this will generate an IORESET signal.

6.3 Video Configuration Register (VCR)

This read only register, accessible at address 200F.0000, indicates the type of video display module plugged into the system board. At present only one option is defined, more will follow.

Figure 6–2: Video Configuration Register - READ ONLY

```
 3
 1                                                       2  1   0
+-------------------------------------------------------+--+--+--+
|                 UNPREDICTABLE                         |  |  |  |
+-------------------------------------------------------+--+--+--+
                                                          |  |  |
                  1024x768/864, 72/60 Hz. -------------   1  1  1
                  1024x768/640x480 60 Hz. -------------   1  1  0
                                                        ( 1  0  1
                                     TBD     -----------(   . . .
                                                        ( 0  0  0
```

6.4 Halt Code Registers (HLTCOD1/2)

The halt code registers (HLTCOD1 and HLTCOD2) are a pair of read/write longword registers at physical addresses 2008.0000h and 2008.0004h. They are intended for use by the ROM-resident program when handling a processor restart.

Figure 6–3: Halt Code Registers (HLTCOD1/2)

```
 3
 1                                                              0
+----------------------------------------------------------------+
|                                                                |
+----------------------------------------------------------------+
```

6.5 Diagnostic Display Register (DIAGDISP)

This is a WRITE ONLY register used to control eight diagnostic LEDs visible from the rear of the system enclosure. bits<7:0> control the eight LEDs. All other bits are ignored. All eight LEDs are turned ON by power-on RESET. They are not affected by IO_RESET.

6.6 Miscellaneous Control Register 0 (BWF0)

This register is a read/write register at address 2008.0014h. It controls when the CPU is told to check for parity errors and what action is taken by the NI and SCSI controllers when a parity error is detected on a memory read. It also controls the operation of the Invalidate Filter and graphics address generator.

Figure 6–4: Miscellaneous Control Register 0 (BWF0)

```
 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0       0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7       4 3 2 1 0
+-+-+-+-----+-+-+-----+---+-+-+-+-----+---+-+-+-+--------+-----+-+
|R|S|W|     |G|F|     |   |M|P|P|     |   |M|P|P|        |     |P|
|E|O|B|0 0 0|M|E| RES |0 0|A|E|E| RES |0 0|A|E|E|  REV   |0 0 0|E|
|S|C|I|     |D|N|     |   |P| |N|     |   |P| |N|        |     |N|
+-+-+-+-----+-+-+-----+---+-+-+-+-----+---+-+-+-+--------+-----+-+
               FILTER      SCSI             NI                  CPU
```

<31> RESERVED. READ ONLY, value returned is UNPREDICTABLE.

<30> SOC. READ ONLY. Returns a ONE to indicate that the CPU type is a DC222.

<29> WBI. READ ONLY. Write Buffer Inhibit, controls the operation of the DC7201 write buffers.

<28:26> Reserved. READ ONLY, value returned is UNPREDICTABLE.

<25> GMD. Controls access by the Graphics Controller to main memory. This bit is normally set. When this bit is set, graphics drawing operations to main memory are restricted when the bus traffic is high to an extent that could cause the Network Controller to experience data over or underrun conditions. This bit is cleared by Power-on and is unaffected by IORESET.

<24> FILT_ENA. Control whether the Invalidate Filter is active or not. A one written to this bit position enables the invalidate filter. This bit is cleared by Power-on and is unaffected by IORESET.

<23:21> Reserved. Return UNPREDICTABLE data upon reading. Ignored on write.

<20:19> UNUSED, read as ZERO, ignored on write.

<18> SCSI_MAP, MAP Error. Read-only, this bit is set when the SCSI control logic accesses the translation map area of memory and finds an entry with the Valid bit clear. The setting of this bit causes a SCSI Interrupt, see Chapter 3. This bit is cleared by any write to this register - write data ignored - and by power-on.

<17> SCSI_PE, Parity Error. Read-only, this bit is set when the SCSI memory control section of the DC7201 detects a memory parity error on a read from main memory if bit<16> is also set. This condition causes an SCSI interrupt This bit is cleared by any write to this register - write data ignored - and by power-on.

<16> SCSI_PEN, SCSI Parity enable. Read/write, when this bit is one and bit<0> is also a one, the SCSI control section of the DC7201 checks the parity of data read from main memory during DMA cycles; when either this bit or bit<0> is zero, parity is not checked. Cleared to zero upon power-on. See bit<17> above.

<15:13> Reserved. Return UNPREDICTABLE data upon reading. Ignored on write.

<12:11> UNUSED, read as ZERO, ignored on write.

<10> NI_MAP, MAP Error. Read-only, this bit is set when the NI control logic accesses the translation map area of memory and finds an entry with the Valid bit clear. The setting of this bit causes an NI Interrupt, see chapter 3. This bit is cleared by any write to this register - write data ignored - and by power-on.

<09> NI_PE, Parity Error. Read-only, this bit is set when the NI memory control section of the DC7201 detects a memory parity error on a read from main memory if bit<08> is also set. This condition causes an NI interrupt by allowing the LANCE chip to time out on the read. The LANCE timeout will cause an NI Interrupt with the MERR bit set in the LANCE status register, see chapter 8, Section 8.3.3. This bit is cleared by any write to this register - write data ignored - and by power-on.

<08> NI_PEN, NI Parity enable. Read/write, when this bit is one and bit<0> is a one, the Ethernet control section of the DC7201 checks the parity of data read from main memory during DMA cycles; when either this bit or bit<0> is zero, parity is not checked. Cleared to zero upon power-on. See bit<09> above.

<07:04> REV. READ ONLY, indicates the DC7201 revision. Rev. A chips read back 0000; Rev. B chips read back 1011; Rev. C chips read back 1100. Ignored on Write.

<03:01> UNUSED, read as ZERO, ignored on write.

<00> PEN, CPU Parity Check Enable. Read/Write, If this bit is set, the CPU will check all its memory read accesses (ADR<29> = 0) for correct parity and will allow parity checking for the NI and/or SCSI - see above for NI and SCSI Parity Enables. If clear, no checks will be made. This bit is cleared by Power-on and is unaffected by IORESET.

6.7 Diagnostic Timer Registers (DIAGTIMU/M)

There are two diagnostic timer registers which may be read as fields of the same longword address - 2008.001Ch. Not all bits of the longword have meaning, hence the definition of two timers. The High resolution timer (DIAGTIMU) has a resolution of 1 microsecond (see note 1 below), the low resolution timer has a resolution of 1 millisecond (see note 2 below).

The high resolution timer may be read as bits<09:00> of address 2008.001Ch, i.e. it may be regarded as a 16-bit register at address 2008.001Ch whose maximum count is 3FFh - equivalent to a time of 1 millisecond (see note 1 below) - wrapping to 000 after this count. This timer's transition from 3FFh to 0 increments the low resolution timer.

The low resolution timer may be read as bits<31:16> of address 2008.001Ch. It has a maximum value of FFFFh, equivalent to a time of approx. 65 seconds (see note 2 below). It wraps to 000 after this highest count.

Reads of either register are synchronised to the timer count source so that a count cannot occur during a read and so data read will always be correct.

Writes to address 2008.001Ch affect DIAGTIMU/M according to the type of write. A write to either byte of the low word of this address will clear DIAGTIMU; a write to either of the high bytes will clear DIAGTIMM; a longword write will clear both timers.

Since the count source for these timers is asynchronous to the system clock, the time between clearing either timer with a write instruction and the first increment is uncertain.

DIAGTIMM/U are cleared to zero upon power-on. They are not affected by an I/O reset.

Figure 6–5: Diagnostic Timer Register (DIAGTIMU)

```
     3                                                           1
     1                                                           6
    +--------------------------------------------------------------+
    |                real time count (milliseconds)               |
    +--------------------------------------------------------------+ ...

     1            1 0
     5            0 9                                            0
... +--------------+-----------------------------------------------+
    | ..  ZERO ..  |    real time count (microseconds )           |
    +--------------+-----------------------------------------------+
```

Note 1 : the actual count source for the high resolution timer is 1.017 μs, thus the maximum count will overflow every 1.042 milliseconds.

Note 2 : the resolution of the low resolution timer is 1.042 milliseconds (see note 1 above), thus the maximum count will overflow every 68.26 seconds.

Chapter 7

GRAPHICS CONTROLLER

7.1 Introduction

The graphics controller for the VS4000 VLC is a part of the memory control section of the DC7201. It is one of the devices that compete for memory cycles - see chapter 4, Section 4.1 and following for details of the other devices. Graphics operations have the lowest priority of all requests for memory cycles. The graphics controller will be referred to from now on as the GC.

The GC supports both 8 plane and single plane 2D graphics and is optimized to execute the most commonly used primitives of X-Windows. Drawing is accomplished with linear addressing. The types of operations supported are: lines; one, two, and three operand rasterops; text. All rasterops can be solid colored, tiled, stippled, color expanded and plane compacted. All operations can be performed to both the frame buffer and non-displayable main system memory using virtual addressing with multiple clipping rectangles for overlapping window hardware support.

The KA48-AA has a plug-in 8-plane 1 Mpixel frame buffer that is used for either monochrome or color displays.

7.2 Screen Formats

The GC supports two screen formats, see below :

Table 7–1: Supported Screen Formats

Format	Color	Mono	Refresh Rate
1024 x 768	yes	yes	72 Hz
1024 x 864	yes	yes	60 Hz

A 2 plane 64x64 cursor is supported and is stored in off-screen memory - see Section 7.6.

The virtual drawing feature makes use of the operating system paging mechanism to allocate space in main memory as the GC address generator requires it. Hardware support is provided within the DC7201 to track CPU write references to the page table entries used by the GC.

7.3 Graphics Primitives

The following graphics primitives are supported:

- — Lines
- — 1, 2, or 3 Operand RasterOps with Tiling, Stipples
- — Text: Variable Sized Glyphs, Solid, Stenciled
- — Fill Spans with Tiling, Stipples for Implementing Polygons
- — Light Pixel for Implementing Complex Primitives
- — Multiple Clipping Rectangle Support for all Primitives
- — All Operands can be Virtual

This section of the VS4000 VLC specification will give only an overview of the GC, for complete details, the PVAX2 Graphics Controller Specification should be consulted.

7.4 CPU - GC Communication

Commands are passed from the CPU to the GC in the form of variable length packets. The CPU sends such a packet by writing to an address in I/O space. The write data is passed through the normal write buffers of the DC7201, see REFERENCE>(CPU_WRITE) in Chapter 4. The GC accepts the command packet data and if it is not currently busy and if the Clip List feature is not enabled, loads the command directly into its registers for execution. If the GC is busy, or if the Clip List feature is enabled, the command packet data is written by the GC into a circular buffer area in main memory - the LCG Command FIFO. The size and location of this FIFO are two of the setup parameters required by the GC.

Figure 7–1: General Format of Command Packets

```
 3       2|2      1|1              0
 1       4|3      6|5              0
+---------+--------+-----------------+
| OPCODE  | FLAGS  | OPCODE Specific |
+---------+--------+-----------------+
|
| ...  More Longwords, specified by <23:22> in FLAGS above
```

The first longword of any command packet has, within an eight bit flag field, two bits which specify how many additional longwords are contained in this packet.

The function that the GC performs is determined by the OPCODE field of the command packet. This is the most significant byte of the first longword of the packet.

7.5 Virtual Drawing

As the GC may be instructed to draw into main memory and because it is generating addresses which are unknown to the CPU when it sends the command packet, drawing to main meory is done to virtual addresses so that the operating system paging mechansim may be used to validate memory as the GC address generator requires. To control virtual memory operations, three translation buffer pointers (PTEs) are maintained within the GC, one for each of the three data structures the GC may read or write - Source, Mask and Destination. For every CPU write cycle, the address being referenced is compared to each of the PTEs, if a match occurs, the GC is halted and a CPU interrupt occurs - the assumption being the the CPU is accessing the PTE with the intention of clearing the Valid bit.

7.6 Cursor

The cursor data is stored in off-screen memory - main memory only. Cursor data is loaded into the GC from the appropriate memory area for each line of the active cursor region.

Chapter 8

NETWORK CONTROLLER

The network controller enables the connection of a VS4000 VLC system to an Ethernet network via a ThinWire connection using RG-58 coax cable or via a transceiver cable. Selection of which of these is active is made by a two position switch acessible from the rear of the machine. One of two LEDs will be lit, indicating the selected connector. The controller is a part of the system board and consists of a LANCE Ethernet controller chip, a Serial Interface Adapter and a 15 pin D-sub connector for the transceiver cable.

8.1 Ethernet Implementation

This option supports the Ethernet Data Link Layer which is specified in DEC Standard 134: *The Digital Ethernet Specification*.

8.1.1 Packet Format

Data is passed over the Ethernet at a serial data rate of 10 million bits per second in variable-length packets. Each packet has the following format:

Figure 8–1: Ethernet Packet Format

```
                +----------------------------+
       6 bytes  |    destination address     |
                +----------------------------+
       6 bytes  |      source address        |
                +----------------------------+
       2 bytes  |           type             |
                +----------------------------+
                |                            |
                |                            |
46..1500 bytes  |           data             |
                |                            |
                |                            |
                +----------------------------+
       4 bytes  |      CRC check code        |
                +----------------------------+
```

The minimum size of a packet is 64 bytes, which implies a minimum data length of 46 bytes. Packets shorter than this are called "runt packets" and are treated as erroneous when received by the network controller.

8.1.2 Network Addresses

Network addresses are 48 bits (6 bytes) long and are of two types:

Physical address: The unique address associated with a particular station on the Ethernet, which should be distinct from the physical address of any other station on any other Ethernet.

Multicast address: A multi-destination address associated with one or more stations on a given Ethernet (sometimes called a logical address). There are two kinds of multicast addresses:

Multicast-group address: An address associated by higher-level convention with a group of logically related stations.

Broadcast address: A predefined multicast address which denotes the set of *all* the stations on the Ethernet.

Bit 0 (the least significant bit of the first byte) of an address denotes the type: it is 0 for physical addresses and 1 for multicast addresses. In either case the remaining 47 bits form the address value. A value of 48 ones is always treated as the broadcast address.

The physical address of each VS4000 VLC system is determined at the time of manufacture and is stored in the Ethernet Address ROM on the system board (see section 5.2.

8.2 Lance Chip Overview

The Lance chip is a microprogrammed controller which can conduct extensive operations independently of the central processor. There are four control and status registers (CSRs) within the Lance chip which are programmed by the central processor (i.e. the CVAX CPU chip) to initialize the Lance chip and start its independent operation. Once started, the Lance uses its built-in DMA controller to directly access RAM memory to get additional operating parameters and to manage the buffers it uses to transfer packets to and from the Ethernet. The Lance uses three structures in memory:

1. Initialization Block—24 bytes of contiguous memory starting on a word boundary. The initialization block is set up by the central processor and is read by the Lance when the processor starts the Lance's initialization process. The initialization block contains the system's network address and pointers to the receive and transmit descriptor rings; it is described in section 8.6 below.

2. Descriptor Rings—two logically circular rings of buffer descriptors, one ring used by the chip receiver for incoming data and one ring used by the chip transmitter for outgoing data. Each buffer descriptor in a ring is 8 bytes long and starts on a quadword boundary. It points to a data buffer elsewhere in memory, contains the size of that buffer, and holds various status information about the buffer's contents. Buffer descriptors are described in section 8.7 below.

3. Data Buffers—contiguous portions of memory to buffer incoming or outgoing packets. Data buffers must be at least 64 bytes long (100 bytes for the first buffer of a packet to be transmitted) and may begin on any byte boundary. They are discussed in section 8.7 below.

When the system is ready to begin network operation, the central processor sets up the initialization block, the receive descriptor ring, the transmit descriptor ring, and their data buffers in memory, and then starts the Lance by writing to its CSR's. The Lance performs its initialization process and then enters its polling loop. In this loop, it listens to the network for packets whose destination addresses are of interest and it scans the transmit descriptor ring for descriptors which have been marked by the central processor to indicate that they contain outgoing data packets. When it detects a network packet of interest, it receives and stores that packet in one or more receive buffers and marks their descriptors accordingly. When it finds a packet to be transmitted, it transmits it to the network and marks its descriptor when transmission is complete. Whenever it completes a reception or transmission (or encounters an error condition), the Lance chip sets flags in its control and status register 0 to signal the central processor (usually by an interrupt) that it has done something of interest.

8.3 Program Control of the Lance

Program control of the Lance chip is via two 16-bit read/write ports, each of which appears as the low-order word of a longword address. These ports are:

Address (HEX)	Name	Description
200E.0000	NI_RDP	Register data port
200E.0004	NI_RAP	Register address port

These ports provide access to four 16-bit control and status registers which are named NI_CSR0 through NI_CSR3. A CSR is accessed by first writing its number into the register address port NI_RAP after which the contents of the CSR are read or written by accesses to the register data port NI_RDP. Note that registers other than NI_CSR0 may be accessed only while the STOP bit of NI_CSR0 is set.

8.3.1 Register Address Port (NI_RAP)

The register address port is a 16-bit read/write port at physical address 200E.0004h. It selects which of the four CSR's is accessed via the register data port.

Figure 8–2: Lance Register Address Port(NIRAP)

```
      15      14      13      12      11      10       9       8
   +-------+-------+-------+-------+-------+-------+-------+-------+--
   |                           reserved
   +-------+-------+-------+-------+-------+-------+-------+-------+-

       7       6       5       4       3       2       1       0
  -+-------+-------+-------+-------+-------+-------+-------+-------+
                   reserved                        |     CSRNO     |
 --+-------+-------+-------+-------+-------+-------+-------+-------+
```

<15:2> Reserved. Ignored on write; read as zeros.

CSRNO CSR select (bits 1:0). These read/write bits select which of the four CSR's is accessible via the register data port. They are cleared to zero upon power-on. Values are:

Bits 1:0	Register
0 0	NI_CSR0
0 1	NI_CSR1
1 0	NI_CSR2
1 1	NI_CSR3

8.3.2 Register Data Port (NI_RDP)

The register data port at physical address 200E.0000h is a 16-bit window through which the CPU can read and write the CSR designated by the register address port NI_RAP.

Note that registers NI_CSR1, NI_CSR2, and NI_CSR3 are accessible only while the STOP bit in NI_CSR0 is set. If that STOP bit is clear (i.e. the Lance chip is active), attempts to read from those CSR's will return UNDEFINED data and attempts to write to them will be ignored. Accesses to a CSR via NI_RDP do not alter the register address pointer NI_RAP. In normal operation only NI_CSR0 can be accessed, so NI_RAP should be set to point to NI_CSR0 and left that way.

8.3.3 Control and Status Register 0 (NI_CSR0)

This register is used by the controlling program to start and stop the operation of the Lance chip and to monitor its status. It is accessible to the processor via port NI_RDP when bits 1:0 of NI_RAP are set to 00. All of its bits can be read at any time and none of its bits is affected by reading the register. The effects of a write operation are described individually for each bit. When power is applied to the system, all the bits in this register are cleared except the STOP bit which is set.

Figure 8–3: Lance Control and Status Register 0 (NICSR0)

```
      15      14      13      12      11      10       9       8
   +-------+-------+-------+-------+-------+-------+-------+-------+--
   |  ERR  |  BABL |  CERR |  MISS |  MERR |  RINT |  TINT |  IDON |
   +-------+-------+-------+-------+-------+-------+-------+-------+-

       7       6       5       4       3       2       1       0
  -+-------+-------+-------+-------+-------+-------+-------+-------+
   |  INTR |  INEA |  RXON |  TXON |  TDMD |  STOP |  STRT |  INIT |
 --+-------+-------+-------+-------+-------+-------+-------+-------+
```

ERR — Error summary (bit 15). This read-only bit is one whenever any of the bits BABL, CERR, MISS, or MERR in this register are ones. Writing to this bit has no effect. It is cleared when all of the bits which set it are zero or when the STOP bit is set.

BABL — Transmitter timeout error (bit 14). This bit is set when the transmitter has been on the channel longer than the time required to send the maximum length packet. It will be set after 1519 data bytes have been transmitted (the chip will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted). This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

CERR — Collision error (bit 13). This bit is set when the collision input to the chip failed to activate within 2 microsec after a chip-initiated transmission is completed. This collision-after-transmission is a transceiver test feature. This function is also known as heartbeat or SQE (signal quality error) test. This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR bit is also one.

MISS — Missed packet (bit 12). This bit is set when the receiver loses a packet because it does not own a receive buffer. The MISS bit is not valid in internal loopback mode. This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

MERR — Memory error (bit 11). This bit is set when the chip attempts a DMA transfer and does not receive a ready response from the memory within 25.6 microsec after beginning the memory cycle. This condition occurs when a parity error occurred on an immediately preceding DMA bus read cycle. When MERR is set, the receiver and transmitter are turned off (bits RXON and TXON of this register are cleared to zero). This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

RINT — Receive interrupt (bit 10). This bit is set when the chip updates an entry in the receive descriptor ring for the last buffer received or when reception is stopped due to a failure. This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

TINT — Transmitter interrupt (bit 9). This bit is set when the chip updates an entry in the transmit descriptor ring for the last buffer sent or when transmission is stopped due to a failure. This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

IDON Initialization done (bit 8). This bit is set when the chip completes the initialization process which was started by setting the INIT bit in this register. When IDON is set, the chip has read the initialization block from memory and stored the new parameters. This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

INTR Interrupt request (bit 7). This read-only bit is one whenever any of the bits BABL, MISS, MERR, RINT, TINT, or IDON in this register are ones. Writing to this bit has no effect. It is cleared when all of the bits which set it are zero or when the STOP bit is set. When both the INTR and INEA bits in this register are set, an interrupt request is sent to the system interrupt controller (see section 8.4).

INEA Interrupt enable (bit 6). This read/write bit controls whether the setting of the INTR bit generates an interrupt request. When both the INTR and INEA bits in this register are set, an interrupt request is sent to the system interrupt controller (see section 8.4). This bit is set when a one is written to it. It is cleared when a zero is written to it or when the STOP bit is set.

RXON Receiver on (bit 5). This read-only bit indicates (when it is one) that the receiver is enabled. RXON is set when initialization is completed (i.e. when IDON is set, unless the DRX bit of the initialization block MODE register was one) and then the STRT bit in this register is set. Writing to this bit has no effect. RXON is cleared when either the MERR or STOP bits of this register are set.

TXON Transmitter on (bit 4). This read-only bit indicates (when it is one) that the transmitter is enabled. TXON is set when initialization is completed (i.e. when IDON is set, unless the DTX bit of the initialization block MODE register was one) and then the STRT bit in this register is set. Writing to this bit has no effect. TXON is cleared when either the MERR or STOP bits of this register are set or when any of bits UFLO, BUFF, or RTRY in a Transmit Buffer Descriptor (see section 8–14) are set.

TDMD Transmit demand (bit 3). Setting this bit signals the chip to access the transmit descriptor ring without waiting for the polltime interval to elapse. This bit need not be set to transmit a packet; setting it merely hastens the chip's response to the insertion of a transmit descriptor ring entry by the host program. This bit is set by writing a one to it (writing a zero has no effect) and is cleared by the chip when it recognizes the bit (the bit may read as one for a short time after it is set, depending upon the level of activity in the chip). TDMD is also cleared when the STOP bit is set.

STOP Stop external activity (bit 2). Setting this bit stops all external activity and clears the internal logic of the chip; this has the same effect as the electrical reset signalled upon power-on. The chip remains inactive and STOP remains set until the STRT or INIT bits in this register are set. This bit is set by writing a one to it (writing a zero has no effect) or upon power-on. It is cleared when either INIT or STRT is set. If the processor writes ones to STOP, INIT, and STRT at the same time, STOP takes precedence and neither STRT nor INIT is set. Setting STOP clears all the other bits in this register. After STOP has been set, the other three CSR's (NI_CSR1, NI_CSR2, and NI_CSR3) must be reloaded before setting INIT or STRT (note that those three registers may be accessed *only* while STOP is set).

STRT Start operation (bit 1). Setting this bit enables the chip to send and receive packets, perform DMA and do buffer management. The STOP bit must be set prior to setting the STRT bit (setting STRT then clears STOP). STRT is set by writing a one to it (writing a zero has no effect). It is cleared when the STOP bit is set.

INIT Initialize (bit 0). Setting this bit causes the chip to perform its initialization process, which reads the initialization block from the memory addressed by the contents of NI_CSR1 and NI_CSR2 using DMA accesses. The STOP bit must be set prior to setting the INIT bit (setting INIT then clears STOP). INIT is set by writing a one to it (writing a zero has no effect). It is cleared when the STOP bit is set.

NOTE

The INIT and STRT bits must not be set at the same time. The proper initialization procedure is as follows:

1. set STOP in NI_CSR0
2. set up the initialization block in memory
3. load NI_CSR1 and NI_CSR2 with the starting address of the initialization block
4. set INIT in NI_CSR0
5. wait for IDON in NI_CSR0 to become set
6. set STRT in NI_CSR0 to begin operation

8.3.4 Control and Status Register 1 (NI_CSR1)

This read/write register is used in conjunction with NI_CSR2 to supply the 24-bit physical memory address of the initialization block which the chip reads when it performs its initialization process. The register is accessible to the processor via NI_RDP when bits 1:0 of NI_RAP are 01 and the STOP bit of NI_CSR0 is set. Its contents upon power-on are UNPREDICTABLE.

Figure 8–4: Lance Control and Status Register 1 (NICSR1)

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |                           IADR 15:8
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
                              IADR 7:1                    |   0   |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

IADR Initialization block address (bits 15:0). These are the low-order sixteen bits of the (24-bit physical) byte address of the first byte of the initialization block. Note that since the block must be word-aligned, bit 0 must be zero.

8.3.5 Control and Status Register 2 (NI_CSR2)

This read/write register is used in conjunction with NI_CSR1 to supply the 24-bit physical memory address of the initialization block which the chip reads when it performs its initialization process. The register is accessible to the processor via NI_RDP when bits 1:0 of NI_RAP are 10 and the STOP bit of NI_CSR0 is set. Its contents upon power-on are UNPREDICTABLE.

Figure 8–5: Lance Control and Status Register 2(NICSR2)

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |                           reserved                            |
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
  |                           IADR 23:16                          |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

<15:8> Reserved. Write with zeros.

IADR Initialization block address (bits 7:0). These are the high-order eight bits of the (24-bit physical) byte address of the first byte of the initialization block.

8.3.6 Control and Status Register 3 (NI_CSR3)

This read/write register controls certain aspects of the electrical interface between the Lance chip and the system. It *must* be set as indicated for each bit. The register is accessible to the processor via NI_RDP when bits 1:0 of NI_RAP are 11 and the STOP bit of NI_CSR0 is set. Its contents upon power-on are entirely zeros.

Figure 8–6: Lance Control and Status Register 3(NICSR3)

```
      15      14      13      12      11      10       9       8
   +-------+-------+-------+-------+-------+-------+-------+-------+--
   |                           reserved
   +-------+-------+-------+-------+-------+-------+-------+-------+-

       7       6       5       4       3       2       1       0
  -+-------+-------+-------+-------+-------+-------+-------+-------+
                  reserved                 |  BSWP |  ACON |  BCON |
 --+-------+-------+-------+-------+-------+-------+-------+-------+
```

<15:3> Reserved. Ignored on write; read as zeros.

BSWP Byte swap (bit 2). When this bit is set, the chip will swap the high and low bytes for DMA data transfers between the silo and bus memory in order to accomodate processors which consider bus bits 15:08 to be the least significant byte of data. This bit is read/write; it is cleared when the STOP bit in NI_CSR0 is set. For this system, this bit *must be ZERO*.

ACON ALE control (bit 1). This bit controls the polarity of the signal emitted on the chip's ALE/AS pin during DMA operation. This bit is read/write; it is cleared when the STOP bit in NI_CSR0 is set. For this system, this bit *must be ZERO*.

BCON Byte control (bit 0). This bit controls the configuration of the byte mask and hold signals on the chip's pins during DMA operation. This bit is read/write; it is cleared when the STOP bit in NI_CSR0 is set. For this system, this bit *must be ZERO*.

8.4 Interrupts

The Lance chip asserts an interrupt request signal whenever the INTR and INEA bits bit in its control and status register 0 (NI_CSR0) are both ones. This signal is presented to the system interrupt controller as interrupt number 5, the "network controller primary" source, whose vector number is 250h. The change of the interrupt signal from false to true will set bit NP in the interrupt request register INT_REQ (which will generate a CPU interrupt when the corresponding bit in the interrupt mask register INT_MSK is also set). Note that since the input to INT_REG is transition sensitive rather than level sensitive, a program which services an interrupt request from the Lance must either service all the conditions which contributed to the setting of the INTR bit in NI_CSR0 so that INTR will become zero, or must generate another transition of the interrupt signal by setting the INEA bit of NI_CSR0 to zero and then back to one again. (The interrupt controller is described in chapter 3. Interrupt number 4, the "network controller secondary" source, is not used by this option.)

8.5 DMA Operation

The Lance chip contains a built-in DMA controller which can transfer data directly between the chip and main memory in the address range 0000.0000 through 07FF.FFFF. (In the VS4000 VLC only system board RAM and option board RAM appear in this address range.) The LANCE is actually connected to the DC7201 which contains data buffers to minimize the use of the RAM bus by the LANCE. The LANCE contains a 48-byte FIFO buffer to allow for DMA service latency and to minimize the number of request-grant arbitration cycles. When transferring large amounts of data in burst mode, the chip transfers 16 bytes per DMA request. Each longword transfer requires <TBS>, so a 16-byte burst will require either

<TBS> or <TBS> depending upon whether or not the data block is longword aligned. This DMA controller is used to read the initialization block, to read and write the descriptor rings, and to read and write data buffers. Note that all the memory addresses handled by the chip are physical addresses. Programs which operate with CPU memory management enabled must translate their addresses from virtual to physical form before presenting them to the Lance chip.

If the DPEN bit of the PAR_CTL register is set, then parity is checked during DMA read cycles. When a parity error is detected, the MERR bit of the NI_CSR0 register will be set (with the consequences described in section 8.3.3 but no CPU Machine Check will occur.

8.6 Initialization Block

When the Lance chip is initialized (by setting the INIT bit in NI_CSR0), it reads a 24-byte block of data called the initialization block from main memory using DMA accesses. The physical address of the initialization block (IADR) is taken from NI_CSR1 and NI_CSR2. Since the data must be word-aligned, the low-order bit of the address must be zero. The initialization block comprises 12 16-bit words arranged as follows:

Figure 8–7: Lance Initialization Block Layout

```
             +------------------------------+
IADR +  0    |             MODE             |
             +------------------------------+
IADR +  2    |         PADR <15:00>         |
             +------------------------------+
IADR +  4    |         PADR <31:16>         |
             +------------------------------+
IADR +  6    |         PADR <47:32>         |
             +------------------------------+
IADR +  8    |        LADRF <15:00>         |
             +------------------------------+
IADR + 10    |        LADRF <31:16>         |
             +------------------------------+
IADR + 12    |        LADRF <47:32>         |
             +------------------------------+
IADR + 14    |        LADRF <63:48>         |
             +------------------------------+
IADR + 16    |         RDRA <15:00>         |
             +------+-----------------------+
IADR + 18    | RLEN |  RDRA <23:16>         |
             +------+-----------------------+
IADR + 20    |         TDRA <15:00>         |
             +------+-----------------------+
IADR + 22    | TLEN |  TDRA <23:16>         |
             +------+-----------------------+
```

8.6.1 Initialization Block MODE Word (NIB_MODE)

The MODE word of the initialization block allows alteration of the Lance chip's normal operation for testing and special applications. For normal operation the MODE word is entirely zero.

Figure 8–8: Initialization Block MODE Word(NIBMODE)

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |  PROM |                       reserved
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
    resv  |  INTL |  DRTY |  COLL |  DTCR |  LOOP |  DTX  |  DRX  |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

PROM Promiscuous mode (bit 15). When this bit is set, all incoming packets are accepted regardless of their destination addresses.

<14:7> Reserved. Should be written with zeros.

INTL Internal loopback (bit 6). This bit is used in conjunction with the LOOP bit in this word to control loopback operation. See the description of the LOOP bit, below.

DRTY Disable retry (bit 5). When this bit is set, the chip will attempt only one transmission of a packet. If there is a collision on the first transmission attempt, a retry error (RTRY) will be reported in the transmit buffer descriptor (section 8–14).

COLL Force collision (bit 4). Setting this bit allows the collision logic to be tested. The chip must be in internal loopback mode for COLL to be used. When COLL is one a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in NI_TMD3.

DTCR Disable transmit CRC (bit 3). When DTCR is zero the transmitter will generate and append a 4-byte CRC to each transmitted packet (normal operation). When DTCR is one the CRC logic is allocated instead to the receiver and no CRC is sent with a transmitted packet. During loopback, setting DTCR to zero will cause a CRC to be generated and sent with the transmitted packet, but no CRC check can be done by the receiver since the CRC logic is shared and cannot both generate and check a CRC at the same time. The CRC transmitted with the packet will be received and written into memory following the data where it can be checked by software. If DTCR is set to one during loopback, the driving software must compute and append a CRC value to the data to be transmitted. The receiver will check this CRC upon reception and report any error.

LOOP Loopback control (bit 2). Loopback allows the Lance chip to operate in full duplex mode for test purposes. The maximum packet size is limited to 32 data bytes (in addition to which 4 CRC bytes may be appended). During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes). Setting LOOP to one allows simultaneous transmission and reception for a packet constrained to fit within the silo. The chip waits until the entire packet is in the silo before beginning serial transmission. The incoming data stream fills the silo from behind as it is being emptied. Moving the received packet out of the silo into memory does not begin until reception has ceased. In loopback mode, transmit data chaining is not possible. Receive data chaining is allowed regardless of the receive buffer length. (In normal operation, the receive buffers must be 64 bytes long, to allow time for buffer lookahead.) Valid loopback bit settings are:

LOOP	INTL	Operation
0	x	Normal on-line operation
1	0	External loopback
1	1	Internal loopback

Internal loopback allows the chip to receive its own transmitted packet without disturbing the network. The chip will not receive any packets from the network while it is in internal loopback mode. External loopback allows the chip to transmit a packet through the transceiver out to the network cable to check the operability of all circuits and connections between the Lance chip and the network cable. Multicast addressing in external loopback is valid only when DTCR is one (user needs to append the 4 CRC bytes). In external loopback, the chip also receives packets from other nodes.

DTX Disable transmitter (bit 1). If this bit is set, the chip will not set the TXON bit in NI_CSR0 at the completion of initialization. This will prevent the Lance chip from attempting to access the transmit descriptor ring, hence no transmissions will be attempted.

DRX Disable receiver (bit 0). If this bit is set, the chip will not set the RXON bit in NI_CSR0 at the completion of initialization. This will cause the chip to reject all incoming packets and to not attempt to access the receive descriptor ring.

8.6.2 Network Physical Address (NIB_PADR)

The 48-bit physical Ethernet network node address is contained in bytes 2:7 of the initialization block. (This is a network address; it has no relationship to any memory address.)

Figure 8–9: Network Physical Address(NIBPADR)

```
|<---IADR+6--->|<---IADR+4--->|<---IADR+2--->|

 47          32 31          16 15           0
+--------------+--------------+--------------+
|                                         |0|
+--------------+--------------+--------------+
```

The contents of NIB_PADR identify this station to the network and must be unique within the domain of the network. Its value is normally taken from the Network Address ROM (see section 5.2. The low-order bit (bit 0) of this address must be zero since it is a physical address.

8.6.3 Multicast Address Filter Mask (NIB_LADRF)

Bytes 8:15 of the initialization block contain the 64-bit multicast address filter mask. The multicast address filter is a partial filter which assists the network controller driver program to selectively receive packets which contain multicast network addresses.

Figure 8–10: Multicast Address Filter Mask(NIB LADRF)

```
|<---IADR+14--->|<---IADR+12--->|<---IADR+10--->|<---IADR+8---->|

 63           48 47           32 31           16 15            0
+---------------+---------------+---------------+---------------+
|                                                               |
+---------------+---------------+---------------+---------------+
```

Multicast Ethernet addresses are distinguished from physical network addresses by the presence of a one in bit 0 of the 48-bit address field. If an incoming packet contains a physical destination address (bit 0 is zero), then its entire 48 bits are compared with the contents of NIB_PADR and the packet is ignored if they are not equal. If the packet contains a multicast destination address which is all ones (the broadcast address), it is always accepted and stored regardless of the contents of the multicast address filter mask.

All other multicast addresses are processed through the multicast address filter to determine whether the incoming packet will be stored in a receive buffer. This filtering is performed by passing the multicast address field through the CRC generator. The high-order 6 bits of the resulting 32-bit CRC are used to select one of the 64 bits of NIB_LADRF. (These high-order six bits represent in binary the number of the bit in NIB_LADRF, according to the labelling in figure 8–10.) If the bit selected from NIB_LADRF is one, the packet is stored in a receive buffer; otherwise it is ignored. This mechanism effectively splits the entire domain of 2**47 multicast addresses into 64 parts, and multicast addresses falling into each part will be accepted or ignored according to the value of the corresponding bit in NIB_LADRF. The driver program must examine the addresses of the packets accepted by this partial filtering to complete the filtering task.

8.6.4 Receive Descriptor Ring Pointer (NIB_RDRP)

Bytes 16:19 of the initialization block describe the starting address and extent of the receive descriptor ring.

Figure 8–11: Receive Descriptor Ring Pointer(NIBRDRP)

```
|<-------IADR+18------->|<-------IADR+16------->|

 31  29 28  24 23     16 15                    0
+------+------+---------+----------------------+
| RLEN | resv |            RDRA             000|
+------+------+---------+----------------------+
```

RLEN Receive ring length (bits 31:29). This field gives the number of entries in the receive descriptor ring, expressed as a power of 2:

RLEN	*Entries*
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

<28:24> Reserved; should be zeros.

RDRA Receive descriptor ring address (bits 23:0). This is the physical address in system memory of the first element in the ring. Since each 8-byte element must be aligned on a quadword boundary, bits 2:0 of this address must be zero.

8.6.5 Transmit Descriptor Ring Pointer (NIB_TDRP)

Bytes 20:23 of the initialization block describe the starting address and extent of the Transmit descriptor ring.

Figure 8–12: Transmit Descriptor Ring Pointer(NIBTDRP)

```
|<-------IADR+22------->|<-------IADR+20------->|

 31  29 28  24 23     16 15                    0
+------+------+---------+----------------------+
| TLEN | resv |             TDRA            000|
+------+------+---------+----------------------+
```

TLEN Transmit ring length (bits 31:29). This field gives the number of entries in the transmit descriptor ring, expressed as a power of 2:

TLEN	*Entries*
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

<28:24> Reserved; should be zeros.

TDRA Transmit descriptor ring address (bits 23:0). This is the physical address in system memory of the first element in the ring. Since each 8-byte element must be aligned on a quadword boundary, bits 2:0 of this address must be zero.

8.7 Buffer Management

The Lance chip manages its data buffers by using two rings of buffer descriptors which are stored in memory: the receive descriptor ring and the transmit descriptor ring. Each buffer descriptor points to a data buffer elsewhere in memory, contains the size of that buffer, and contains status information about that buffer’s contents.

The starting location in memory of each ring and the number of descriptors in it are given to the Lance chip via the initialization block (see sections 8–11 and 8–12) during the chip initialization process. Each descriptor is 8 bytes long and must be aligned on a quadword boundary (the three low-order bits of its address must be zero). The descriptors in a ring are physically contiguous in memory and the number of descriptors must be a power of 2. The Lance keeps an internal index to its current position in each ring which it increments modulo the number of descriptors in the ring as it advances around each ring.

Once started, the Lance polls each ring to find descriptors for buffers in which to receive incoming packets and from which to transmit outgoing packets, and revises the status information in buffer descriptors as it processes their associated buffers. When polling, the Lance is limited to looking only one ahead of the descriptor with which it is currently working. The high speed of the data stream requires that each buffer be at least 64 bytes long to allow time to chain buffers for packets which are larger than one buffer. (The first buffer of a packet to be transmitted should be at least 100 bytes to avoid problems in case a late collision is detected.)

Each descriptor in a ring is "owned" either by the Lance chip or by the host processor; this status is indicated by the OWN bit in each descriptor. Mutual exclusion is accomplished by the rule that each device can only relinquish ownership of a descriptor to the other device, it can never take ownership; and that each device cannot change any field in a descriptor or its associated buffer after it has relinquished ownership. When the host processor sets up the rings of descriptors before starting the Lance, it sets the OWN bits such that the Lance will own all the descriptors in the receive descriptor ring (to be used by the Lance to receive packets from the network) and the host will own all the descriptors in the transmit descriptor ring (to be used by the host to set up packets to be transmitted to the network).

8.7.1 Receive Buffer Descriptor

A receive buffer descriptor comprises 4 words aligned in memory on a quadword address boundary.

Figure 8–13: Receive Buffer Descriptor

```
Memory
Offset
         1 1 1 1 1 1
         5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
        +---------------+---------------+
        |                               |
 +0:    |             LADR              |
        |                               |
        +---------------+---------------+
        |O E F O C B S E|               |
        |W R R F R U T N|               |
 +2:    |N R A L C F P P|     HADR      |
        |    M O   F    |               |
        +---------------+---------------+
        |       |                       |
 +4:    |1 1 1 1|         BCNT          |
        |       |                       |
        +---------------+---------------+
        |       |                       |
 +6:    |0 0 0 0|         MCNT          |
        |       |                       |
        +---------------+---------------+
```

LADR Low-order buffer address (offset 0, bits 15:0). These are the low-order 16 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the Lance.

HADR High-order buffer address (offset 2, bits 7:0). These are the high-order 8 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the Lance.

OWN Owned flag (offset 2, bit 15). This bit indicates whether the descriptor is owned by the host (OWN = 0) or by the Lance (OWN = 1). The Lance clears OWN after filling the buffer associated with the descriptor with an incoming packet. The host sets OWN after emptying the buffer. In each case, this must be the last bit changed by the current owner, since changing OWN passes ownership to the other party and the relinquishing party must not thereafter alter anything in the descriptor or its buffer.

ERR Error summary (offset 2, bit 14). This is the logical OR of the FRAM, OFLO, CRC and BUFF bits in this word. Set by the Lance and cleared by the host.

FRAM Framing error (offset 2, bit 13). This bit is set by the Lance to indicate that the incoming packet stored in the buffer had both a non-integral multiple of eight bits and a CRC error. It is cleared by the host.

OFLO Overflow error (offset 2, bit 12). This bit is set by the Lance to indicate that the receiver has lost part or all of an incoming packet because it could not store it in the buffer before the chip's silo overflowed. Cleared by the host.

CRC Checksum error (offset 2, bit 11). This bit is set by the Lance to indicate that the received packet has an invalid CRC checksum. Cleared by the host.

BUFF Buffer error (offset 2, bit 10). This bit is set by the Lance when it has used all its owned receive descriptors or when it could not get the next descriptor in time while attempting to chain to a new buffer in the midst of a packet. When a buffer error occurs, an overflow error (bit OFLO) also occurs because the Lance continues to attempt to get the next buffer until its silo overflows. BUFF is cleared by the host.

STP Start of packet (offset 2, bit 9). This bit is set by the Lance to indicate that this is the first buffer used for this packet. Cleared by the host.

ENP End of packet (offset 2, bit 8). This bit is set by the Lance to indicate that this is the last buffer used for this packet. When both STP and ENP are set in a descriptor, its buffer contains an entire packet; otherwise two or more buffers have been chained together to hold the packet. ENP is cleared by the host.

1111 Offset 4, bits 15:12 must be set by the host to ones. Unchanged by the Lance.

BCNT Buffer size (offset 4, bits 11:0). This is the number of bytes in the buffer (whose starting address is in HADR and LADR) in two's complement form. Note that the minimum buffer size is 64 bytes and that the maximum required for a legal packet is 1518 bytes. Written by the host; unchanged by the Lance.

0000 Offset 6, bits 15:12 are reserved; they should be set to zeros by the host when it constructs the descriptor.

MCNT Byte count (offset 6, bits 11:0). This is the length in bytes of the received packet for which this is the last or only descriptor. MCNT is valid only in a descriptor in which ENP is set (last buffer) and ERR is clear (no error). Set by the Lance and cleared by the host.

8.7.2 Transmit Buffer Descriptor

A transmit buffer descriptor comprises 4 words aligned in memory on a quadword address boundary.

Figure 8–14: Transmit Buffer Descriptor

```
Memory
Offset
         1 1 1 1 1 1
         5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
        +---------------+---------------+
        |                               |
 +0:    |             LADR              |
        |                               |
        +---------------+---------------+
        |O E r M O D S E|               |
        |W R e O N E T N|               |
 +2:    |N R s R E F P P|     HADR      |
        |    v E        |               |
        +---------------+---------------+
        |       |                       |
 +4:    |1 1 1 1|         BCNT          |
        |       |                       |
        +---------------+---------------+
        |B U r L L R|                   |
 +6:    |U F e C C T|                   |
        |F L s O A R|       TDR         |
        |F O v L R Y|                   |
        +---------------+---------------+
```

LADR — Low-order buffer address (offset 0, bits 15:0). These are the low-order 16 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the Lance.

HADR — High-order buffer address (offset 2, bits 7:0). These are the high-order 8 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the Lance.

OWN — Owned flag (offset 2, bit 15). This bit indicates whether the descriptor is owned by the host (OWN = 0) or by the Lance (OWN = 1). The host sets OWN after filling the buffer with a packet to be transmitted. The Lance clears OWN after transmitting the contents of the buffer. In each case, this must be the last bit changed by the current owner, since changing OWN passes ownership to the other party and the relinquishing party must not thereafter alter anything in the descriptor or its buffer.

ERR — Error summary (offset 2, bit 14). This is the logical OR of the LCOL, LCAR, UFLO and RTRY bits in this descriptor. Set by the Lance and cleared by the host.

resv — Offset 2, bit 13 is reserved. The Lance will write a zero in this bit.

MORE — More retries (offset 2, bit 12). The Lance sets this bit when more than one retry was required to transmit the packet. Cleared by the host.

ONE One retry (offset 2, bit 11). The Lance sets this bit when exactly one retry was required to transmit the packet. Cleared by the host.

DEF Deferred (offset 2, bit 10). The Lance sets this bit when it had to defer while trying to transmit the packet. This occurs when the network is busy when the Lance is ready to transmit. Cleared by the host.

STP Start of packet (offset 2, bit 9). This bit is set by the host to indicate that this is the first buffer used for this packet. STP is not changed by the Lance.

ENP End of packet (offset 2, bit 8). This bit is set by the host to indicate that this is the last buffer used for this packet. When both STP and ENP are set in a descriptor, its buffer contains an entire packet; otherwise two or more buffers have been chained together to hold the packet. ENP is not changed by the Lance.

1111 Offset 4, bits 15:12 must be set by the host to ones. Unchanged by the Lance.

BCNT Byte count (offset 4, bits 11:0). This is the number of bytes, in two's complement form, which the Lance will transmit from this buffer. Note that for any buffer which is not the last of a packet, at least 64 bytes (100 bytes if it is the start of the packet) must be transmitted to allow adequate time for the Lance to acquire the next buffer. Written by the host; unchanged by the Lance.

NOTE: The remaining fields of the descriptor (which make up its entire fourth word) are valid only when the ERR bit in the second word has been set by the Lance.

BUFF Buffer error (offset 6, bit 15). This bit is set by the Lance during transmission when it does not find the ENP bit set in the current descriptor and it does not own the next descriptor. When BUFF is set, the UFLO bit (below) is also set because the Lance continues to transmit until its silo becomes empty. BUFF is cleared by the host.

UFLO Underflow (offset 6, bit 14). This bit is set by the Lance when it truncates a packet being transmitted because it has drained its silo before it was able to obtain additional data from a buffer in memory. UFLO is cleared by the host.

RESV Offset 6, bit 13 is reserved. The Lance will write a zero in this bit.

LCOL Late collision (offset 6, bit 12). This bit is set by the Lance to indicate that a collision has occured after the slot time of the network channel has elapsed. The Lance does not retry after a late collision. LCOL is cleared by the host.

LCAR Loss of carrier (offset 6, bit 11). This bit is set by the Lance when the carrier-present input to the chip becomes false during a transmission initiated by the Lance. The Lance does not retry after such a failure. LCAR is cleared by the host.

RTRY Retries exhausted (offset 6, bit 10). This bit is set by the Lance after 16 attempts to transmit a packet have failed due to repeated collisions on the network. (If the DRTY bit of the initialization block MODE word (section 8–8) is set, RTRY will instead be set after only only one failed transmission attempt.) RTRY is cleared by the host.

TDR Time domain reflectometer (offset 6, bits 9:0). These bits are the value of an internal counter which is set by the Lance to count system clocks from the start of a transmission to the occurrence of a collision. This value is useful in determing the approximate distance to a cable fault; it is valid only when the RTRY bit in this word is set.

8.8 Lance Operation

The Lance chip operates independently of the host under control of its own internal microprogram. This section is a simplified description of the operation of the Lance in terms of its principal microcode routines (these should not be confused with device driver programming in the host, which is not a part of this specification). These microcode routines make use of numerous temporary storage cells within the Lance chip; most of these are not accessible from outside the chip but they are mentioned here when necessary to clarify the operation of the microcode.

Two such (conceptual) internal variables are of central importance: the pointers to the "current" entry in the receive descriptor ring and in the transmit descriptor ring, which are referred to below as TXP and RXP. Each of these designates the descriptor which the Lance will use for the next operation of that type. If the descriptor designated by one of these pointers is not owned by the Lance (the OWN bit is 0), then the Lance can neither perform activity of that type nor advance the pointer. For the transmit ring, the Lance will do nothing until the host sets up a packet in the buffer and sets the OWN bit in the descriptor designated by the Lance's TXP. (The host must keep track of the position of the TXP, since setting up a packet in some other descriptor will not be detected by the Lance). For the receive ring, if the Lance does not own the descriptor designated by RXP, it cannot receive a packet. In both rings, when the Lance finishes with a descriptor and relinquishes it to the host by clearing OWN, it then advances the ring pointer (modulo the number of entries in the ring).

When the Lance begins activity using the current descriptor (i.e. begins receiving or transmitting a packet), it may look ahead at the next descriptor and attempt to read its first three words in advance so it can chain to the next buffer in mid-packet without losing data. However, it does not actually advance its RXP or TXP until it has cleared the OWN bit in the current descriptor.

The Lance is a very complex chip and this system specification does not attempt to cover all the details of its operation. The chip purchase specification cited in section 1.3 and the chip vendor's literature should also be consulted.

8.8.1 Switch Routine

Upon power on, the STOP bit is set and the INIT and STRT bits are cleared in NI_CSR0. The Lance microprogram begins execution in the switch routine, which tests the INIT, STRT, and STOP bits. When the host sets either INIT or STRT, STOP is cleared. While STOP is set, if the host writes to NI_CSR1 and NI_CSR2, that data is stored for use by the initialization routine.

When the microprogram sees STOP cleared, it tests first the INIT bit and then the STRT bit. If INIT is set, it performs the initialization routine (section 8.8.2). Then if STRT is set, it begins active chip operation by jumping to the look-for-work routine (section 8.8.3). Control returns to the switch routine whenever the host again sets the STOP bit (which also clears the INIT and STRT bits). Note that the ring pointers RXP and TXP are not altered by the setting of either STOP or START; they are reset to the start of their rings only when INIT is set.

8.8.2 Initialization Routine

The initialization routine is called from the switch routine when the latter finds the INIT bit set. It reads the initialization block (section 8.6) from the memory addressed by NI_CSR1 and NI_CSR2 and stores its data within the Lance chip. This routine also sets the ring pointers RXP and TXP to the start of their rings (i.e. to point to the descriptor at the lowest memory address in the ring).

8.8.3 Look-for-work Routine

The look-for-work routine is executed while the Lance is active and looking for work. It is entered from the switch routine when the STRT bit is set, and is returned to from the receive and transmit routines after they have received or transmitted a packet.

This routine begins by testing whether the receiver is enabled (bit RXON of NI_CSR0 is set). If so, it tries to have a receive buffer available for immediate use when a packet addressed to this system arrives. It tests its internal registers to see whether it has already found a receive descriptor owned by the Lance and, if not, calls the receive poll routine (section 8.8.4) to attempt to get a receive buffer.

Next the routine tests whether the transmitter is enabled (bit TXON of NI_CSR0 is set). If so, it calls the transmit poll routine (section 8.8.7) to see whether there is a packet to be transmitted and to transmit it if so.

If there was no transmission and the TDMD bit of NI_CSR0 is not set, the microprogram delays 1.6 milliseconds and then goes to check the receive descriptor status again. If a packet was transmitted or the host has set TDMD, the delay is omitted so that multiple packets will be transmitted as quickly as possible.

If at any point in this routine the receiver detects an incoming packet whose destination address matches the station's physical address, is the broadcast address, or passes the multicast address filter (or if the PROM bit of NIB_MODE is set), the receive routine (section 8.8.5) is called.

8.8.4 Receive Poll Routine

The receive poll routine is called whenever the receiver is enabled and the Lance needs a free buffer from the receive descriptor ring. The routine reads the second word of the descriptor designated by RXP and, if the OWN bit in it is set, reads the first and third words also.

8.8.5 Receive Routine

The receive routine is called when the receiver is enabled and an incoming packet's destination address field matches one of the criteria described in 8.8.3 above. The routine has three sections: initialization, lookahead, and descriptor update.

In initialization, the routine checks whether a receive ring descriptor has already been acquired by the receive poll routine. If not, it makes one attempt to get the descriptor designated by RXP (if OWN is not set in it, MISS and ERR are set in NI_CSR0 and the packet is lost). The buffer thus acquired is used by the receive DMA routine to empty the silo.

In lookahead, the routine reads the second word of the next descriptor in the receive ring and, if the OWN bit is set, reads the rest of the descriptor and holds it in readiness for possible data chaining.

The descriptor update section is performed when either the current buffer is filled or the packet ends. If the packet ends but its total length is less than 64 bytes, it is an erroneous "runt packet" and is ignored: no status is posted in the descriptor, RXP is not moved, and the buffer will be reused for the next incoming packet (this is why a receive buffer must be at least 64 bytes long; otherwise the runt might be detected after advancing RXP).

If the packet ends (with or without error), the routine writes the packet length into MCNT, sets ENP and other appropriate status bits and clears OWN in the current descriptor, and sets RINT in NI_CSR0 to signal the host that a complete packet has been received. Then it advances RXP and returns to the look-for-work routine.

If the buffer is full and the packet has not ended, chaining is required. The routine releases the current buffer by writing status bits into its descriptor (clearing OWN and ENP, in particular), makes current the next descriptor data acquired in the lookahead section, advances RXP, and goes to the lookahead section to prepare for possible additional chaining. Note that RINT is not set in NI_CSR0, although the host would find OWN cleared if it looked at the descriptor, and it could begin work on that section of the packet, since the mutual exclusion rule prevents the Lance from going back and altering it.

8.8.6 Receive DMA Routine

The receive DMA routine is invoked asynchronously by the chip hardware during execution of the receive routine whenever the silo contains 16 or more bytes of incoming data or when the packet ends and the silo is not empty. It executes DMA cycles to drain data from the silo into the buffer designated by the current descriptor.

8.8.7 Transmit Poll Routine

The transmit poll routine is called by the look-for-work routine (section 8.8.3) to see whether a packet is ready for transmission. It reads the second word of the descriptor designated by TXP and tests the OWN bit. If OWN is zero, the Lance does not own the buffer and this routine returns to its caller. If OWN is set, the routine tests the STP bit, which should be set to indicate the start of a packet. If STP is clear, this is an invalid packet; the Lance sets its OWN bit to return it to the host, sets TINT in NI_CSR0 to notify the host, and advances TXP to the next transmit descriptor. If both OWN and STP are set, this is the beginning of a packet, so the transmit poll routine reads the rest of the descriptor and then calls the transmit routine (section 8.8.8) to transmit the packet. During this time the chip is still watching for incoming packets from the network and it will abort the transmit operation if one arrives.

8.8.8 Transmit Routine

The transmit routine is called from the transmit poll routine (section 8.8.7) when the latter finds the start of a packet to be transmitted. This routine has three sections: initialization, lookahead, and descriptor update. In initialization, the routine sets the chip's internal buffer address and byte count from the transmit descriptor, enables the transmit DMA engine, and starts transmission of the packet preamble. It then waits until the transmitter is actually sending the bit stream (including possible backoff-and-retry actions in case of collisions).

In lookahead, the transmit routine test the current descriptor to see whether it is the last in the packet (the ENP bit is set). If so, no additional buffer is required so the routine waits until all the bytes from the current packet have been transmitted. If not, the routine attempts to get the next descriptor and hold it in readiness for data chaining, and then waits until all the bytes from the current buffer have been transmitted.

Descriptor update is entered when all the bytes from a buffer have been transmitted or an error has occured. If there is no error and the buffer was not the last of the packet, the pre-fetched descriptor for the next buffer is made current for use by the transmit DMA routine. The routine writes the appropriate status bits and clears the OWN bits in the current descriptor and advances TXP. If this was the last buffer in the packet, the routine sets the TINT bit in NI_CSR0 to notify the host and returns to the look-for-work routine (section 8.8.3); otherwise it goes back to the lookahead section in this routine.

8.8.9 Transmit DMA Routine

The transmit DMA routine is invoked asynchronously by the chip hardware during execution of the transmit routine whenever the silo has 16 or more empty bytes. It executes DMA cycles to fill the silo with data from the buffer designated by the current descriptor.

8.8.10 Collision Detect Routine

This routine is invoked asynchronously by the chip hardware during execution of the transmit routine when a collision is detected on the network. It ensures that the "jam" sequence is transmitted, then backs up the chip's internal buffer address and byte count registers, waits for a pseudo-random backoff time, and then attempts the transmission again. If 15 retransmission attempts fail (a total of 16 attempts), it sends the microcode to the descriptor update routine to report an error in the current transmit descriptor (bits RTRY and ERR are set).

8.9 Lance Programming Notes

1. The interrupt signal is simply the OR of the interrupt-causing conditions. If another such condition occurs while the interrupt signal is already asserted, there will not be another active transition of the interrupt signal and the interrupt request bit in INT_REQ will not be set again. An interrupt service routine should use logic similar to the following to avoid losing interrupts:

   i Read NI_CSR0 and save the results in a register, say R0.

   ii Clear the interrupt enable bit INEA in the saved data in R0.

   iii Write NI_CSR0 with the saved data in R0. This will make the interrupt signal false because INEA is clear and will clear all the write-one-to-reset bits such as RINT, TINT and the error bits; it will not alter the STRT, INIT or STOP bits nor any interrupt-cause bits which came true after NI_CSR0 was read.

   iv Write NI_CSR0 with only INEA to enable interrupts again.

   v Service all the interrupt and error conditions indicated by the flags in the data in R0.

vi Exit from the interrupt service routine.

Note

> Be sure to access NI_CSR0 only with instructions which do a single access, such as MOVE. Instructions such as BIS which do a read-modify-write operation can have unintended side effects.

2. An interrupt is signalled to the host only when the *last* buffer of a multibuffer (chained) packet is received or transmitted. However, the OWN bit in each descriptor is cleared as soon as the Lance has finished with that portion of the packet, and the mutual exclusion rule makes it safe for the host to process such a descriptor and its buffer.

3. When a transmitter underflow occurs (UFLO is set in a transmit descriptor because the silo is not filled fast enough), the Lance will turn off its transmitter and the Lance must be restarted to turn the transmitter back on again. This can be done by setting STOP in NI_CSR0 and then setting STRT in NI_CSR0 (DTX will still be clear in the chip's internal copy of NIB_MODE). It is necessary to set INIT to reread the initialization block.

Note

> Note that setting STOP will immediately terminate any reception which is in progress. If the status of a receive descriptor has been updated and its OWN bit is now clear, then the contents of its buffer are valid. If the incoming packet was chained into more than one buffer, however, the packet is only valid if its last buffer has been completed (the one with the ENP bit set).

4. The network controller hardware requires up to five seconds after power on to become stable. Self-test routines must delay at least this time before attempting to use the controller for either internal or external testing.

5. The LCAR flag (loss of carrier) may be set in the transmit descriptor when a packet is sent in internal loopback mode. When the Lance is operating in internal loopback mode and a transmission is attempted with a non-matching address, the Lance will correctly reject that packet. If the next operation is an internal loopback transmission without first resetting the Lance, the packet will not be sent and LCAR will be set in the transmit descriptor for that packet. The receive descriptor will still be owned by the Lance. To avoid this problem, the Lance should be reinitialized after each internal loopback packet.

6. The ONE flag is occasionally set in a transmit descriptor after a late collision. The Lance does not attempt a retransmission even though ONE may be set. The host should disregard ONE if the LCOL flag is also set.

7. The chip's internal copy of NI_CSR1 may become invalid when the chip is stopped. The NI_CSR1 and NI_CSR2 registers should always be loaded prior to setting INIT to initialize the Lance chip.

8. Attempting an external loopback test on a busy network can cause a silo pointer misalignment if a transmit abort occurs while the chip was preparing to transmit the loopback packet. The resulting retransmission may cause the transmitter enable circuit to hang, and the resulting illegal length transmission must be terminated by the jabber timer in the transceiver. It is unlikely that there will be a corrupted receive buffer because the reception that caused the transmit abort will usually not pass address recognition.

Since external loopback is a controlled situation it is possible to implement a software procedure to detect a silo pointer misalignment problem and prevent continuous transmissions. Since the test is being done in loopback the exact length and contents of the receive packet are known; thus the software can determine whether the data in the receive buffer has been corrupted.

On transmission the diagnostic software should allow up to 32 retries before a hard error is flagged. This is not to say that 32 errors are allowed for each condition; the sum of all errors encountered in the test should not exceed 32. The diagnostic software should expect to get a transmit done interrupt with 1 millisecond of passing the transmit packet to the Lance. If this does not occur, it should reset the Lance and retry the test. This prevents a continuous transmission (babble) longer than the longest legal packet in case the Lance has become hung.

9. When the chip is in internal loopback mode and a CRC error is forced, a framing error will also be indicated along with the CRC error. In external loopback, when a CRC error is forced only that error is indicated; a framing error is indicated only if the Lance actually receives extra bits.

10. When transmit data chaining, a BUFF error will be set in the current transmit descriptor if a late collision or retry error occurred while the Lance was still transmitting data from the previous buffer. The BUFF error in this case is an invalid error indication and should be ignored. BUFF is valid only when UFLO is also set.

11. When the host program sets up a packet for transmission in chained buffers, it should set the OWN bits in all the transmit buffers *except the first one* (i.e. the one containing the STP bit), and then as its last act set the OWN bit in the first descriptor. Once that bit is set, the Lance will start packet transmission and may encounter an underflow error if the subsequent descriptors for the packet are not available.

12. Do not set INIT and STRT in NI_CSR0 at the same time. After stopping the chip, first set INIT and wait for IDON, then set STRT. If both are set at once, corrupt transmit or receive packets can be generated if RENA becomes true during the initialization process.

13. When a missed packet error occurs, the Lance chip must be halted using the stop bit and the re-loaded with it's initialization block. The stop bit must be set within 75 microseconds of the missed packet error or it may result in a silo pointer misalignment, which can cause transmission of the next packet to be bad.

Chapter 9

SCSI CONTROLLER

This chapter describes the SCSI controller portion of the VS4000 VLC system board.

The controller conforms to the ANSI Small Computer System Interface (SCSI) specification. It has a single port, connecting both to the single (optional) device within the VS4000 VLC system box and allowing for expansion externally.

9.1 SCSI Overview

The SCSI electrical and logical interface and operation is described in detail in the ANSI draft standard issued by ANSI task group X3T9.2, and the particular subset of that standard used by the devices attached to the VS4000 VLC system board are described in each device's specification. The programmer must use all of those documents in conjunction with this specification as a guide. This section reviews a few important features of the ANSI document to set the context for the following discussion of the VS4000 VLC implementation of the SCSI interface.

The SCSI interface is a single-ended bi-directional 8-bit-wide bus to which up to eight devices can be attached. The KA48-AA System Module itself is one of those devices, so up to seven additional devices can be attached. Devices may play one of two roles: initiator or target. An initiator originates an operation by sending a command to a specific target. A target performs an operation which was requested by an initiator. In this specification it is assumed that the VS4000 VLC is always an initiator and that all other SCSI devices attached to it are targets. (There is, however, no *hardware* feature of the VS4000 VLC SCSI interface which prevents its sharing the bus with a second, or third or . . . initiator or assuming the role of a target.)

Each device attached to the SCSI bus is identified by a unique device ID number in the range 0 through 7. During the arbitration, selection, and reselection bus phases in which an initiator and a target establish a connection, the device IDs of the initiator and target are both placed on the data bus by asserting the data bits corresponding to the device ID numbers. By convention, the ID number of the VS4000 VLC system is six (this is controlled by the programs which drive the SCSI interface; it is not fixed in VS4000 VLC hardware). The KA48 System Firmware gives details of the allocation of device IDs.

The electrical interface consists of 18 signal lines. Some of these lines are driven only by initiators, some only by targets, and some by either. These lines are summarized below, using the signal names by which they are called in this specification and in the ANSI specification.

In this specification and in all the registers of the SCSI interface chip, the "true" or "asserted" value of a signal appears as a "one", and the "false" or "negated" value of a signal appears as a "zero". The bus electrical signals are all low true and are driven by open-collector drivers.

The SCSI bus is always terminated at each end. The bus is permanently terminated at the controller (near end). When a disk drive is installed in the system box, a cable connects from the SCSI connector on the system module to the disk drive and on to a connector mounted in the rear panel of the system enclosure. Far end termination can take place in one of two locations:

At the expansion connector on the rear of the system enclosure.
At the second expansion connector on a storage expansion unit.

Note that none of the DEC supported SCSI storage devices provide SCSI bus termination.

9.2 SCSI Bus Signals

DB7..0 and DBP — comprise an 8-bit parallel data bus with an associated odd parity bit. The use of the parity bit is optional but strongly encouraged. These lines may be driven by either an initiator or a terminator, depending upon the direction of data transfer.

RST — signals all devices on the SCSI bus to reset to their initial power-on states. Thereafter, it should be asserted only as a last resort during error recovery since it indiscriminately affects all devices on the bus. An RST signal generated by some other device on the bus causes an internal reset of the 53C94 chip used in this controller and sets the interrupt request bit (INT in register SCS_STATUS).

BSY and SEL — are used by initators and targets during the arbitration, selection, and reselection bus phases to establish or resume a logical connection between an initiator and a target. Once the connection is established, the target asserts BSY and the SEL signal is not driven by anyone.

C/D, I/O and MSG — collectively indicate one of six possible information transfer phases, according to the following table. These signals are always driven by the target device.

MSG	C/D	I/O	Phase name	Transfer direction
0	0	0	Data out	to Target
0	0	1	Data in	to Initiator
0	1	0	Command	to Target
0	1	1	Status	to Initiator
1	0	0		(reserved)
1	0	1		(reserved)

MSG	C/D	I/O	Phase name	Transfer direction
1	1	0	Message out	to Target
1	1	1	Message in	to Initiator

ATN is used by an initiator to signal a target that it has a message ready. The target can receive the message by entering the "message out" phase. ATN is always driven by an initiator.

REQ and ACK are used to synchronize information transfers over the data bus during any of the six information transfer phases. REQ is always driven by the target, ACK is always driven by the initiator.

9.3 Controller Overview

At the heart of the VS4000 VLC storage controller is a single 53C94 SCSI controller chip through which a host program can examine and manipulate all the SCSI Bus signals. Associated with this chip is DMA logic which can transfer data between the SCSI bus and the DC7201. DMA data transfers are 16-bits wide, CPU initiated transfers are 8-bits wide using the 8 low order bits of the same 16-bit bus.

The 53C94 chip can be used by both initiator and target devices. In this specification, only its use as an initiator is described.

The 53C94 chip reduces CPU overhead by performing common SCSI commands as sequences in reponse to a single CPU command. It has an on-chip FIFO which may be accessed by the SCSI bus, the CPU directly or the DC7201 when performing DMA transfers to/from system memory. All Command, Data, Status and Message bytes pass through the FIFO going to/from the SCSI bus. As an example, the CPU can load a Command Descriptor Block and (optionally) one or three Message Bytes into the FIFO, issue one of several Selection Commands and then just wait for an interrupt. The 53C94 chip will wait for the SCSI bus to become free, Arbitrate for the bus until it acquires it, send the Message Bytes followed by the Command Descriptor Block and then interrupt the CPU.

The 53C94 SCSI Controller chip has thirteen addressable 8-bit internal registers that control its operation. Note that although data transfers to/from the 53C94 chip registers are eight bits wide; some registers actually are concatenated in use to form wider registers. Additionally some registers have different bit definitions for read operations than for write operations.

9.3.1 Register Definition

The thirteen internal registers of the 53C94 SCSI Contoller chip are accessed as bits<07:00> of thirteen sequential longwords. Writes to the remaining bits of each longword are ignored, reads of those bits return UNPREDICTABLE data. The address range assigned to the 53C94 chip register set is 200C.0000 to 200C.002Ch. In the following register description, only bits<5:2> of the address will be referenced.

Register 0

Write : Transfer Count - LOW. The value loaded into this register is copied to the low 8-bits of the Transfer Counter Register at the start of a subsequent DMA transfer. This then becomes the low 8-bits of the count of bytes to be transferred during that DMA transfer.

Note

Some confusion may be possible here, be careful to distinguish between the "Transfer Count" and the "Transfer Counter" registers.

Register 1

Write : Transfer Count - HIGH. The value loaded into this register is copied to the high 8-bits of the Transfer Counter Register at the start of a subsequent DMA transfer. This then becomes the high 8-bits of the count of bytes to be transferred during that DMA transfer.

Note: for sucessive DMA transfers having the same transfer length, Registers 0/1 need not be re-loaded between transfers. Loading Registers 0 and 1 with all zeroes specifies a maximum length transfer (65,536 bytes). Registers 0 and 1 are unaffected by any RESET, their contents are UNPREDICATABLE following power-on.

Register 0

Read : Transfer Counter - LOW. The value read is the low 8-bits of the Transfer Counter Register. This register may be read to help to determine the number of bytes remaining to be transferred if a transfer sequence terminates early.

Register 1

Read : Transfer Counter - HIGH. The value read is the high 8-bits of the Transfer Counter Register. This register may be read to help to determine the number of bytes remaining to be transferred if a transfer sequence terminates early.

Register 2

Read/Write : FIFO. The 53C94 chip FIFO is a 16-deep register through which all data to/from the SCSI bus flows. The SCSI bus may transfer 8 or 9 bit values into the FIFO depending on the setting of the parity check control bits in the Configuration 1 Register (see below). The last FIFO data element and associated flags are initialized to zero by any RESET, by a software RESET CHIP and by the beginning of a Bus initiated Selection or Re-selection. The other FIFO elements are not affected by any RESET, but when the flags are zero, successive FIFO reads will always return the contents of the bottom element.

Register 3

Read/Write : Command Register. This is actually a two deep FIFO in which up to two commands may be stacked. Once the first command has completed, the second, if present, will be executed immediately. The last command, or the currently executing command will be the value read from this register. If two commnads are stacked in the Command Register, subsequently two interrupts may result. If the first interrupt is not serviced before the second command completes, the second interrupt will be waiting. When the Interrupt Register is then read, servicing the first interrupt, the contents of the Status Register, Sequence Step Register and Interrupt Registers will then change to reflect the second (pending) interrupt status.

Bit 7 Enable DMA : When this bit is clear (0) and the 53C94 chip is sending data to the SCSI bus, data is transmitted to the bus until the FIFO is empty. Subsequent bytes, as needed, must be written to the FIFO register using the programmed I/O mode. The Transfer Counter Register is not modified when sending in this mode.

When this bit is clear (0) and the 53C94 chip is receiving data from the SCSI bus, a single byte of data will be transferred into the FIFO.

When this bit is set (1) and the 53C94 chip is sending data to the SCSI bus, data will be transferred from memory, via the DC7201 into the FIFO. The Transfer Counter Register will be decremented as data is loaded into the FIFO. Transmission will continue to the SCSI bus until the FIFO is empty and the Transfer Counter Register is zero.

When this bit is set (1) and the 53C94 chip is receiving data from the SCSI bus, the data received will be put into the FIFO and from there transferred via the DC7201 to memory. As data is put into the FIFO, the Transfer Counter Register is decremented. Data transfer from the SCSI bus into the FIFO continues until the Transfer Counter Register is zero. Data transfer out of the FIO to the DC7201 continues until the FIFO is empty and the Transfer Counter Register is zero.

Bits<6:0> Command Code : The 53C94 chip responds to 28 commands, specified by the command code.

Bit<6> = 1 - Disconnect Mode

Bit<5> = 1 - Target Mode

Bit<4> = 1 - Initiator Mode

Bits<6:4> all zero - Miscellaneous Mode

The Command Register is cleared by any of the following conditions :

—Hardware RESET
—Software RESET
—SCSI Bus RESET
—SCSI Bus Disconnect
—Bus initiated Selection or Re-selection
—Select Command
—Reconnect Command if ATN is set
—Select or Re-select Timeout
—Target Terminate Command
—Parity Error detected in Target Mode
—Assertion of ATN in Target Mode
—Any Phase change in Initiator Mode
—Illegal Command

Register 4 Write : Select/reselect Bus ID Register. This is a 3-bit write-only register that specifies the destination bus ID for a Select or Reselect command. Data is right justified in the byte. The other 5 bits are RESERVED. The destination ID is not affected by any RESET. These bits are UNPREDICTABLE following power-on.

Register 4 Read : Status Register, reading this register returns information on the status of the 53C94 chip and the reason for an interrupt having occurred.

Bit 7 Interrupt Bit. Any RESET or a read from the Interrupt Register will reset this bit.

Bit 6 Gross Error. This chip status bit is set when any one of the following has occurred.

– The top of the FIFO has been overwritten.

– The top of the Command Register has been overwritten.

– A DMA transfer has been attempted in the wrong direction.

– A phase change occurs in the middle of an initiator synchronous data transfer.

No interrupt is generated when a Gross Error occurs.

This bit is cleared by a hardware or software RESET, but not by a SCSI bus RESET, or by reading the Status Register if the Interrupt bit is set.

Bit 5 Parity Error. This bit is set when parity has been enabled in the Configuration 1 Register and the 53C94 chip subsequently detects a parity error on data received from the SCSI bus. This bit is cleared by a hardware or software RESET, but not by a SCSI bus RESET, or by reading the Status Register if the Interrupt bit is set.

Bit 4 Terminal Count. This bit sets when the Transfer Counter decrements to zero. It is reset when the Transfer Count is loaded. Reading the Interrupt Register *does not* clear this bit. This bit is cleared by a hardware or software RESET, but not by a SCSI bus RESET.

Bit 3 Valid Group Code. This bit is set during the target command sequence if the received command is one of the non-reserved groups (0, 1, 5, 6 or 7). It is reset when the Interrupt Register is read if the Interrupt bit is set. It is also cleared by a hardware or software RESET, but not by a SCSI bus RESET.

Bits<2:0> Phase Bits. These bits indicate the state of the SCSI MSG, C/D and I/O signals showing the current bus phase. These bits can be expected to be stable for any read following an interrupt.

Bits<2:0>	**SCSI Bus Phase**
000	Data OUT
001	Data IN
010	Command
011	Status
100	*** reserved ***
101	*** reserved ***
110	Message OUT
111	Messsage IN

Register 5

Read : Interrupt Register, this is an eight bit register used in conjunction with the Status Register and the Sequence Step Register to determine the cause of an interrupt. Reading this register when an interrupt has been posted by the 53C94 chip will clear all three of these registers.

Bit 7 - SCSI RESET Detected. This bit will be set if the SCSI Reset Reporting bit is set to 0 in the Configuration 1 Register AND the chip detects a Reset on the SCSI bus. If the interrupt is not serviced in <TBD>, the chip itself will assert Reset for <TBD>.

Bit 6 - Illegal Command. This bit will be set when the chip detects an unused code or an illegal command written to the Command Register.

Bit 5 - Disconnect. This bit will be set when the chip is operating in initiator mode and the current target disconnects from the SCSI bus or if a selct/reselect timeout occurs. In target mode, this bit is set when the Terminate Sequence or Command Complete Sequence command cuases the 53C94 chip to disconnect from the SCSI bus.

Bit 4 - Bus Service. When the chip is operating in target mode this bit is set when the Initiator requests service by asserting ATN. When the chip is operating in initiator mode this bit will be set whenever the target is requesting an Information Transfer Phase.

Bit 3 - Function Complete. When operating as a Target, this bit is set after completion of any command, When the chip is operating in initiator mode, the bit is set after a Target has been selected - by transferring any command phase bytes - or after a Transfer Information Command when the Target is requesting Message In Phase.

Bit 2 - Reselected. This bit is set to indicate that the chip has been reselected as an initiator.

Bit 1 - Selected with ATN. This bit is set during Selection Phase to indicate that the chip has been selected as a Target and that ATN was asserted on the SCSI bus.

Bit 0 - Selected. This bit is set during Selection Phase to indicate that the chip has been selected as a target and that the ATN line was false.

Reading the Interrupt Status Register with an interrupt pending will cause the Interrupt Request line to become de-asserted and the Interrupt Status and Sequence Step Registers to be cleared.

On receipt of an interrupt, the sequence of action should be always to read the Status Register and Sequence Step Registers before reading the Interrupt Status Register.

Register 5

Write : Select/reselect Timeout Register. This is an eight bit register that specifies the time to wait for a response during selection and reselection. More <TBS>.

Table 9–1 (Cont.): Initator Select with ATN and Stop

Register 6	Read : Sequence Step. The three least signicant bits of this register, <2:0> indicate for certain sequences which sub-steps of the current sequence had been executed when an interrupt occurs. Bit <3>, when read as a one, indicates when the synchronous offset limit has been reached. The upper four bits are reserved. Sequences that utilize this register are as follows :

Table 9–1: Initator Select with ATN and Stop

Sequence	Interrupt	
Step	**Register**	
210	**76543210**	**Interpretation**
000	00100000	Arbitration complete, Selection Time-out, disconnected
000	00011000	Arbitration and Selection Complete. Stopped because target did not assert Messsage Out Phase. ATN still asserted by the 53C94 chip.
001	00011000	Message Out Complete. Sent one message byte. ATN on.

Table 9–2: Initator Select without ATN

Sequence	Interrupt	
Step	**Register**	
210	**76543210**	**Interpretation**
000	00100000	Arbitration complete, Selection Time-out, disconnected
010	00011000	Arbitration and Selection Complete. Stopped because target did not assert Command Phase.
011	00011000	Stopped during Command Transfer because target prematurely changed phase.
100	00011000	Select sequence complete.

Table 9–3: Initator Select with ATN

Sequence	Interrupt	
Step	**Register**	
210	**76543210**	**Interpretation**
000	00100000	Arbitration complete, Selection Time-out, disconnected
000	00011000	Arbitration and Selection Complete. Stopped because target did not assert Message Out Phase. ATN still driven by 53C94 chip
010	00011000	Message Out complete. Sent one Message byte with ATN true, then released ATN. Stopped because Target did not assert Command Phase after Message byte was sent.

Table 9–3 (Cont.): Initator Select with ATN

Sequence Step 210	Interrupt Register 76543210	Interpretation
011	00011000	Stopped during Command transfer due to premature phase change. Some CDB bytes may not have been sent; check FIFO flags.
100	00011000	Selection with ATN sequence complete.

Table 9–4: Initator Select with ATN3

Sequence Step 210	Interrupt Register 76543210	Interpretation
000	00100000	Arbitration complete, Selection Time-out, disconnected
000	00011000	Arbitration and Selection Complete. Stopped because target did not assert Message Out Phase. ATN still driven by 53C94 chip
010	00011000	Sent 1, 2 or 3 Message bytes. Stopped because target prematurely changed from Message Out phase or did not assert Command phase after third Message byte. ATN released only if third Message byte *was* sent.
011	00011000	Stopped during Command transfer due to premature phase change. Some CDB bytes may not have been sent; check FIFO flags.
100	00011000	Selection with ATN3 sequence complete.

Table 9–5: Target Selected without ATN

Sequence Step 210	Interrupt Register 76543210	Interpretation
000	00000001	Selected, loaded Bus ID into FIFO, loaded null-byte message into FIFO
001	00000001	Stopped in Command Phase due to parity error. Some Command Descriptor Block bytes may not have been received. Check FIFO flags.
001	00010001	Same as above, Initator asserted ATN in Command phase.

Table 9–5 (Cont.): Target Selected without ATN

Sequence	Interrupt	
Step	Register	
210	76543210	Interpretation
010	00000001	Selected, received entire Command Descriptor Block. Check Valid Group Status Bits.
010	00010001	same as above, Initiator asserted ATN in Command phase.

Table 9–6: Target Selected with ATN, SCSI-2 bit not set

Sequence	Interrupt	
Step	Register	
210	76543210	Interpretation
000	00000010	Selected with ATN, stored Bus ID and one Message byte. Stopped due to either parity error or invalid ID Message.
000	00010010	Selected with ATN, stored Bus ID and one Message byte. Stopped because ATN remained true after first Message byte.
001	00000010	Stopped in Commmand phase due to a parity error. Some CDB bytes not received. Check Valid Group Code bit and FIFO flags.
001	00010010	Stopped in Command phase. Parity Error and ATN true.
010	00000010	Selection complete. Received one Message byte and the entire Command Descriptor Block.
010	00010010	Same as above, Initiator asserted ATN during Command phase.

Table 9–7: Target Receive Command

Sequence	Interrupt	
Step	Register	
210	76543210	Interpretation
001	00001000	Stopped during Commnad Transfer due to parity error. Check FIFO flags.
001	00011000	Stopped during Command Transfer due to parity error. ATN asserted by Initiator.
010	00001000	Received entire Command Descriptor Block.
010	00011000	Received entire CDB, Initiator asserted ATN.

Table 9–8: Target Disconnect Sequence

Sequence Step 210	Interrupt Register 76543210	Interpretation
000	00011000	Sent one Message byte. Stopped becasue Initiator set ATN.
001	00011000	Sent two Message bytes. Stopped because Initiator set ATN.
010	00101000	Disconnect Sequence complete. Disconnected, bus is free.

Table 9–9: Target Terminate Sequence

Sequence Step 210	Interrupt Register 76543210	Interpretation
000	00011000	Sent Status byte. Stopped because Initator set ATN.
001	00011000	Sent Status and Message bytes. Stopped because Initiator set ATN.
010	00101000	Terminate Sequence complete. Disconnected, bus is free.

Table 9–10: Target Command Complete Sequence

Sequence Step 210	Interrupt Register 76543210	Interpretation
000	00011000	Sent Status byte. Stopped because Initator set ATN.
001	00011000	Sent Status and Message bytes. Stopped because Initiator set ATN.
010	00101000	Command Complete Sequence complete.

Register 6

Write : Synchronous Transfer Period Register. The lower five bits of this register specify the minimum time between leading edges of successive REQ and ACK pulses when operating in synchronous mode. The high order 3 bits are reserved. The register is loaded after power on, or following a hardware RESET with a value of 00101. The time unit is the input clock period with the following relation between value loaded and transfer period :

	Register Value	**Offset (clock Periods)**
	00100	5
	00101	5
	00110	6
	00111	7
. . .		
	11111	31
	00000	32
	00001	33
	00010	34
	00011	35

Register 7

Write : Synchronous Offset Register. This four bit register specifies the maximum REQ/ACK offset allowed during synchronous operation. An offset of zero specifies asynchronous operation. When sending data out on the SCSI bus, the 53C94 chip will stop sending when th is offset is reached and from then on send one byte for each ACK it subsequently receives. When receiving data from the SCSI bus, the 53C94 chip will send an ACK every time a byte is removed from its FIFO.

Register 7

Read : FIFO Flags Register. Bits<4:0> of this register indicate the number of bytes remaining in the FIFO. Bits<7:5> are a copy of the Sequence Step Register bits when operating in normal mode.

Register 8

Read/write : Configuration-1 Register. This specifies different operating options for the 53C94 chip.

Bit 7 - slow cable mode. When set, this bit increases the minimum data setup time by one cycle when transmitting to the SCSI bus. This bit should be clear for the use of the 53C94 chip in this application.

Bit 6 - SCSI Reset Interrupt Disable. This bit controls the reporting of a SCSI Bus RESET condition. If a SCSI RESET occurs with this bit set, the 53C94 chip will disconnect from the SCSI bus and remain in the idle state without reporting an interrupt or timing out.

Bit 5 - Parity Test Mode. When set, this bit causes the parity line of the SCSI bus to be driven from bit 7 of the transmitted data byte rather than from the output of the parity generator of the 53C94 chip. It is intended for use in diagnostic mode ONLY. This bit must be cleared in normal operation.

Bit 4 - Parity Enable. When this bit is set, the 53C94 chip checks received data for correct parity on all bus transactions except during arbitration and when receiving pad bytes.

Bit 3 - Chip Test Mode. This bit must be cleared in normal operation. It enables test circuitry used only during chip test.

Bits <2:0> - My Bus ID. This three bit field specifies the SCSI bus ID to which the 53C94 chip responds.

Register 9

Write : Clock Conversion Register. This is a three bit register, using bits <2:0> that sets the chip timings relative the external clock frequency. Bits <7:3> are reserved. The value loaded into this register must never be one. For the clock frequency used in this application of the 53C94 chip, this register must be programmed with a value of 5.

Register A

Write : Test Register. This is a three bit register, using bits <2:0> that places the 53C94 chip in various test modes. For this register's settings to be enabled, the chip must first be placed in the Test Mode by setting bit <3> in Register 8. More tbs.

Register B

Read/write : Configuration 2 Register.

Bit 7 - Reserve FIFO Byte. This bit allows a 16-bit DMA transfer to begin on a non-word boundary for Initiator Synchronous Data In. When this bit is set, a single byte is reserved in the bottom of the FIFO when the phase changes to Synchronous Data In. The next byte written to register F will be placed in the bottom of the FIFO and will clear this bit. The first subsequent 16-bit DMA read will read this byte as the low byte of the first word; the first byte received across the SCSI bus will become the high byte of the first 16-bit word read in the DMA operation. The device driver should copy the low byte of the memory destination word into the FIFO in response to s Synchronous Data In interrupt before issuing the DMA Transfer Info. Command. This bit is cleared by a hardware or software reset but is unaffected by a SCSI bus reset.

Bit 6 - Enable Phase Latch. When this bit is cleared, the Pahse Bits, as reported in the Ststus Register, are live indicators of the state of the SCSI phase lines. When this bit is set, the phase is latched at the completion of any command. Reading the Interrupt Register resets the phase latch. This bit is cleared by a hardware or software reset but is unaffected by a SCSI bus reset.

Bit 5 - Enable Byte Control. This bit is unused for the operation mode of the 53C94 chip used here. This bit *MUST* be left clear. It is cleared by a hardware or software RESET.

Bit 4 - DREQ High Impedance. This bit *MUST* be clear. It is cleared by a hardware or software RESET.

Bit 3 - SCSI-2. When this bit is set, the 53C94 chip supports two new features that are adopted in SCSI-2; the three byte message exchange for Tagged-Queuing. and Group 2 Commands.

> Tagged-Queuing. When this bit is set and the 53C94 chip is selected with ATN, it will request either one or three Message Bytes depending on whether ATN remains true or goes false. If ATN is still truw after teh furst byte has been received, the 53C94 chip may request two more Message bytes before switching to Command Phase. If ATN goes false, it will switch to Command Phase after only a single byte. When this bit is clear, the 53C94 chip will abort the Selection if the target does not switch to Command phase after the transfer of a single Message Byte.

> Group 2 Commands. Group 2 commands are 10-byte sequences. Receiving a Group 2 Commnad with the SCSI-2 bit set will set the Valid Group Code bit in the Status Register. If the SCSI-2 bit is not set, Group 2 Commands will be treated as reserved commands; only six will be requested in Command Phase and the Valid Group Code bits will not be set.

Bit 2 - Target Bad Parity Abort. When this bit is set, the 53C94 chip will abort a Receive Command or Receive Data sequence if it deteects bad parity.

Bit 1 - Register Parity Enable. This bit must be left clear. It is cleared by a hardware or software RESET, but is unaffected by a SCSI RESET.

Bit 0 - DMA Parity Enable. This bit must be left clear. It is cleared by a hardware or software RESET, but is unaffected by a SCSI RESET.

Register C

Configuration 3 Register. Only the low three bit sof this register are used, the high order five bits are RESERVED.

Bit 2 - Save Residual Byte. If a DMA transfer is initiated that requires that an odd number of bytes be transferred and this bit is set, the last byte will be left in the FIFO, or will remain in the output buffer of the DC7201 and must be transferred by CPU intervention. If this bit is *not* set, the odd byte will be transferred as the low byte of the last 16-bit word and, on a transfer from the 53C94 chip to the DC7201, the high byte will be set to all ones. This bit is cleared by a hardware or software RESET, but is unaffected by a SCSI RESET.

Bit 1 - Alternate DMA Mode. This bit must be left clear. It is cleared by a hardware or software RESET, but is unaffected by a SCSI RESET.

Bit 0 - Threshold Eight. This bit must be left clear. It is cleared by a hardware or software RESET, but is unaffected by a SCSI RESET.

9.4 DMA Control Registers

Two registers control the DMA operations of the 53C94 chip. The first is a 24-bit register that specifies a byte address for the start of a SCSI DMA transfer. The high 15 bits of this address are used as an index into the reserved Map region of memory - see Chapter 4, Section 4.4. The other is a direction register, specifying the direction of DMA data transfer.

Figure 9–1: SCSI DMA Address Register - Address 200C.0000h

```
 3            2 2                                            0
 1            4 3                                            0
+--------------+----------------------------------------------+
|     IGN      |         DMA Start Address                    |
+--------------+----------------------------------------------+
```

Figure 9–2: SCSI DMA Direction Register - Address 200C.000Ch

```
 3                                                         0 0
 1                                                         1 0
+-----------------------------------------------------------+-+
|                        IGN                                |D|
+-----------------------------------------------------------+-+
```

D = 0, the DMA transfer is a READ - data from 53C94 chip to memory.

D = 1, the DMA transfer is a WRITE - data from memory to 53C94 chip.

Chapter 10

SERIAL LINE CONTROLLER

The VS4000 VLC system board serial line controller handles four asynchronous serial lines. The controller and a 48 entry silo shared by all four receive lines are parts of the DC7201.

Note: This controller is similar to a VAXstation 3100 serial controller. The operation of the receivers and transmitters is virtually identical, but the arrangement of modem signals, interrupt controls, and the register addresses is **NOT** the same.

10.1 Line Usage

The four serial lines are numbered 0 through 3, and each has a particular primary use, as follows:

Table 10–1: Serial Line Useage

Line	Device
0	Keyboard - connected to a 4-pin MJ connector mounted on the system board. Data leads only. Supports the LK401 keyboard.
1	Pointer - connected to an 8-pin DIN connector mounted on the System Board. Data leads only. Supports VSXXX-AA mouse or VSXXX-AB Tablet.
2	Communications - connected to a 25-pin D-sub connector mounted on the system board, RS423 compatible. Data leads plus Modem Control signals.
3	Printer - connected to a 6-pin MMJ connector mounted on the system board. DEC423 data leads only.

10.2 Diagnostic Terminal Connection

Line 3 is normally connected to a printer through a BC16E cable. If a switch located on the video module is set to enable, a received break condition on this line will assert the CPU halt signal which will cause a processor restart with a restart code of 2.

10.3 Interrupts

The controller generates two interrupt requests, each with a separate vector and enable bit in the INT_REQ and INT_MSK registers. These are receiver done or silo alarm, and transmitter done. Chapter 3, Section 3.7 lists the vector values. In order for these interrupts to be signalled to the CPU, the appropriate bits in the interrupt mask register INT_MSK must be set (see Chapter 3, Section 3.6. Note that, unlike a DZQ11 board, there are no interrupt enable bits in the control and status register SER_CSR.

10.4 Register Summary

The controller contains eight addressable registers as follows:

Table 10–2: Serial Line Controller Register Addresses

Address (HEX)	Name	Access	Description
200A.0000	SER_CSR	R/W	Control and status register
200A.0004	SER_RBUF	R	Receiver buffer - oldest data in silo
200A.0004	SER_LPR	W	Line parameter register
200A.0008	SER_TCR	R/W	Transmitter control register
200A.000C	SER_MSR	R	Modem status register
200A.000C	SER_TDR	W	Transmitter data register
200A.0010	DZ_LPR0	R	Line parameter register, line 0
200A.0014	DZ_LPR1	R	Line parameter register, line 1
200A.0018	DZ_LPR2	R	Line parameter register, line 2
200A.001C	DZ_LPR3	R	Line parameter register, line 3

10.4.1 Control and Status Register (SER_CSR)

The Control and Status register is a 16-bit register at address 200A.0000h. It must be read on a word basis but can be written on either a word or byte basis. All bits in SER_CSR are cleared to zero by power-on or by setting the master clear bit SER_CSR<CLR>.

Figure 10–1: Serial Line Control and Status Register (SER_CSR)

```
     15      14      13      12      11      10       9       8
   +-------+-------+-------+-------+-------+-------+-------+-------+--
   | TRDY  |   0   |  SA   |  SAE  |   0   |   0   |     TLINE     |
   +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
  -+-------+-------+-------+-------+-------+-------+-------+-------+
   | RDONE |   0   |  MSE  |  CLR  | MAINT |   0   |   0   |   0   |
 --+-------+-------+-------+-------+-------+-------+-------+-------+
```

TRDY — Transmitter Ready (bit 15). This read-only bit is set by the hardware when the transmitter scanner stops on a line whose transmitter buffer is ready to be loaded with another character and whose related transmitter control register bit SER_TCR<TXEN_x>is set. While the <TRDY> bit is one, and at no other time, the transmitter line number bits SER_CSR<TLINE> are valid. When <TRDY> changes from zero to one, the interrupt request register bit INT_REQ<ST> is set to one. If interrupt mask register bit INT_MSK<ST> is also one, a transmitter interrupt request to the CPU will be generated; otherwise SER_CSR<TRDY> can be polled by the host program. However, the interrupt request bit INT_REQ<ST> is not automatically cleared while interrupts are masked, so when changing from polled to interrupt operation there may be a spurious interrupt request to the CPU unless the host program clears INT_REQ<ST> by writing a one to INT_CLR<ST>. The <TRDY> bit is cleared when data is loaded into the transmitter for the line number indicated in SER_CSR<TLINE> by writing to register SER_TDR. If additional transmitter lines need service, <TRDY> is set again within 1.4 µs of the completion of the transmitter data load operation. The <TRDY> bit is also cleared when the master scan enable bit SER_CSR<MSE> is cleared, or when the related transmitter control register bit SER_TCR<TXEN_x> is cleared.

<14> — Not used, read as zero.

SA — Silo Alarm (bit 13). This read-only bit is set by the hardware when 16 characters have been entered into the FIFO silo buffer. While the silo alarm enable bit SER_CSR<SAE> is one, the transition of <SA> from zero to one sets interrupt request register bit INT_REQ<SR> to one. If interrupt mask register bit INT_MSK<SR> is also one, an interrupt is signalled to the CPU; otherwise the <SA> bit may be polled. However, the interrupt request register bit INT_REQ<SR> is not automatically cleared while that interrupt is masked, so when changing from polled to interrupt operation, there may be a spurious interrupt request to the CPU unless the host program clears INT_REQ<SR> by writing a one to INT_CLR<SR>.

Note

> The <SA> bit is cleared by reading the receiver buffer register SER_RBUF. When responding to a silo alarm, the host program need not empty the silo, <SA> will be re-asserted and a new interrupt will ocuur when sufficient new characters have entered the silo such that there are again 16 characters in the silo. This operation is different from that of the VAXstation 3100 serial line controller SILO.

The <SA> bit is always zero while the silo alarm enable bit SER_CSR<SAE> is zero.

SAE Silo Alarm Enable (bit 12). This read/write bit selects the source of the receive interrupt request signal which is sent to bit INT_REQ<SR> in the interrupt controller. If <SAE> is one, the silo alarm bit <SA> discussed above is used as the signal; if <SAE> is zero, the receiver done bit <RDONE> discussed below is used instead. Note that while <SAE> is zero, <SA> is also forced to be zero.

<11:10> Not used, read as zero.

TLINE Transmitter Line Number (bits 9:8). These read-only bits indicate the number of the line whose transmitter buffer needs servicing (bit 8 is the least significant bit). These bits are valid only while the transmitter ready bit SER_CSR<TRDY> is one. These bits are cleared when the master scan enable bit SER_CSR<MSE> is cleared.

RDONE Receiver Done (bit 7). This read-only bit is set by the hardware when an incoming character appears at the output of the silo buffer. While the silo alarm enable bit SER_CSR<SAE> is zero, the transition of <RDONE> from zero to one sets interrupt request register bit INT_REQ<SR> to one. If interrupt mask register bit INT_MSK<SR> is also one, an interrupt is signalled to the CPU; otherwise the <RDONE> bit may be polled. However, the interrupt request register bit INT_REQ<SR> is not automatically cleared while that interrupt is masked, so when changing from polled to interrupt operation, there may be a spurious interrupt request to the CPU unless the host program clears INT_REQ<SR> by writing a one to INT_CLR<SR>. <RDONE> is cleared when the receiver buffer register SER_RBUF is read. If another character is available in the silo, <RDONE> will be set again after a delay of between 0.1 and 1.0 microsecond. This bit is also cleared when the master scan enable bit SER_CSR<MSE> is cleared.

<6> Not used, read as zero.

MSE Master Scan Enable (bit 5). This read/write bit must be set to one to permit the receiver and transmitter control sections to scan the lines to service them. When this bit is zero, the transmitter ready bit SER_CSR<TRDY> is cleared and the receiver silo is cleared.

CLR Master Clear (bit 4). When this bit is set by a program, the hardware performs an internal initialization process. At the conclusion of this process the hardware clears this bit. Reading this bit will always return a zero. This initialization clears all registers, the silo, and all UARTs with the following exceptions:

—In the receiver buffer register only bit SER_RBUF<DVAL> is cleared; the remaining bits are not.
—Bits <15:8> of the transmitter control register SER_TCR (the modem control outputs) are not cleared.
—The modem status register SER_MSR is not cleared.

Note

PROGRAMMING NOTE: In previous implementations of this "DZ11-like" serial line controller, after setting the master clear bit SER_CSR<CLR>, a program needed to repeatedly read SER_CSR until it found SER_CSR<CLR> equal to zero before attempting any other operations with the serial line controller. This is no longer needed, any read of SER_CSR will *always* return a zero in bit position SER_CSR<CLR>.

Neither of the interrupt controller registers INT_REQ and INT_MSK are altered when <CLR> is set. The host program must clear bits <SR> and <ST> of INT_MSK to zero and must clear those bits in INT_REQ by writing ones to the corresponding bits of INT_CLR to complete the initialization process.

MAINT Maintenance (bit 3). This read/write bit, when set, loops the serial output connections of the transmitters to the corresponding serial input connections of the receivers. This feature is intended for hardware diagnostic use.

<2:0> Not used, read as zero.

10.4.2 Receiver Buffer Register (SER_RBUF)

The Receiver Buffer register is a 16-bit read-only register at address 200A.0004h which must be read as a word. It contains the received character at the bottom of the silo buffer (that is, the oldest character in the silo). Reading this register removes the character from the silo buffer, and all the other characters in the silo then shift down to the lowest unoccupied location. When this register is read (or when the master clear bit SER_CSR<CLR> is set or after a power-on reset), the data valid bit SER_RBUF<DVAL> is cleared and the remaining bits of the register (although not cleared) are invalid.

Figure 10–2: Serial Line Receiver Buffer Register (SER_RBUF)

```
      15     14     13     12     11     10      9      8
   +------+------+------+------+------+------+------+------+--
   | DVAL | OERR | FERR | PERR |  1   |  1   |    RLINE    |
   +------+------+------+------+------+------+------+------+--

       7      6      5      4      3      2      1      0
 --+------+------+------+------+------+------+------+------+
   |                          RCHAR                        |
---+------+------+------+------+------+------+------+------+
```

DVAL Data Valid (bit 15). This bit, when one, indicates that the data in bits <14:0> of the register is valid. This permits an interrupt handling program to read the receiver buffer register repeatedly and store each character until this bit is read as zero, which indicates that the silo is empty.

OERR Overrun Error (bit 14). This bit is one when a received character is overwritten in a UART buffer by a following character before the first character was transferred to the silo. This condition indicates that the program is not emptying the silo fast enough.

FERR Framing Error (bit 13). This bit is one if the received character did not have a stop bit present at the correct time. The combination of <FERR> set and <RCHAR> entirely zero is usually interpreted as indicating that a BREAK has been received. The receipt of a framing error on line 3 (the printer port) is a special case, the line controller hardware asserts a signal whose effect is described under Diagnostic Terminal Connection - Section 10.2, if this enabled.

PERR Parity Error (bit 12). This bit is one if the sense of the parity of the accompanying character does not agree with the parity which was defined for the line when its line parameter register SER_LPR was last loaded.

<11:10> Not used, read as ones.

RLINE Receiver Line Number (bits 9:8). These bits indicate the number of the line from which the character was received (bit 8 is the least significant bit).

RCHAR Received Character (bits 7:0). Characters with a width of fewer than 8 bits (as defined when the line's line parameter register was last loaded) are right justified with the unused bit positions cleared. The parity bit is not included in the received character.

10.4.3 Line Parameter Register (SER_LPR) and DZ_LPR<3:0>

The Line Parameter register, SER_LPR is a 16-bit register at address 200A.0004h which controls the operating parameters of each line.

This register is write-only and must be written as a 16-bit word. The parameters for each line must be reloaded after each power-on reset or setting of the master clear bit SER_CSR<CLR>. The operating parameters should not be modified for a line while data transmission or reception is in progress on that line.

The parameters for each line may be read individually by accessing the four read only registers DZ_LPR<3:0>.

Figure 10–3: Serial Line Parameter Register (SER_LPR) and DZ_LPR3:0

```
    15     14     13     12     11     10      9      8
 +------+------+------+------+------+------+------+------+---
 |   0  |   0  |   0  |RXENAB|         SPEED             |
 +------+------+------+------+------+------+------+------+--

       7      6      5      4      3      2      1      0
 --+------+------+-------------+------+------+------+------+
   |ODDPAR|PARENB|     IGN     |CHARW | IGN  |    PLINE    |
---+------+------+-------------+------+------+------+------+
```

<15:13> Not used.

<12> RXENAB Receiver. This bit must be set in order for the UART for this line to receive bits and assemble them into characters.

<11:08> SPEED Speed Code. These bits select the data bit rate for the receiver and transmitter for the line. The bits are encoded as follows:

11	10	9	8	Data rate (bits/second)	
0	0	0	0	300	
0	0	0	1	300	
0	0	1	0	300	
0	0	1	1	300	
0	1	0	0	300	
0	1	0	1	300	*
0	1	1	0	600	*
0	1	1	1	1200	*
1	0	0	0	2400	
1	0	0	1	2400	
1	0	1	0	2400	*
1	0	1	1	4800	
1	1	0	0	4800	*
1	1	0	1	9600	
1	1	1	0	9600	*
1	1	1	1	19200	*

Note 1 : only those speeds marked with an asterisk are supported, attempts to set other non-supported speeds, using the values established by the KA44 System System will default to the nearest supported speed as shown. The asterisks represent the KA44 System compatible settings for the supported speeds.

Note 2 : the highest speed supported is 19,200 baud - this is different from some older systems.

<07> ODDPAR Odd Parity. If this bit is set and the parity enable bit SER_LPR<PARENB> is also set, then characters with odd parity are transmitted to the line and characters received from the line are expected to have odd parity. If this bit is clear and the parity enable bit SER_LPR<PARENB> is set, then characters with even parity are transmitted to the line and characters received from the line are expected to have even parity. If the parity enable bit SER_LPR<PARENB> is clear, then the setting of this bit is immaterial.

<06> PARENB Parity Enable. If this bit is set, characters transmitted to the line have a parity bit appended and characters received from the line have their parity checked. The sense of the parity is according to the setting of the odd parity bit SER_LPR<ODDPAR>.

<05:04> IGN These two bits are ignored.

<03> CHARW Character width. Character widths of 7 and 8 bits only are supported. Bit<03> = 1 makes transmitted characters have 8 data elements and expects that received characters will have 8 data elements; bit<03> = 0 sets 7 data elements for transmit and receive.

<02> Ignored on write, reads as zero.

<01:00> PLINE Parameter Line Number. These bits specify the number of the line to which the parameters in the rest of the register apply. Bit 0 is the least significant bit.

10.4.4 Transmitter Control Register (SER_TCR)

The Transmitter Control register is a 16-bit register at address 200A.0008h which must be read on a word basis and can be written on either a word or byte basis.

Figure 10–4: Serial Line Transmitter Control Register (SER_TCR)

```
      15     14     13     12     11     10     9      8
   +------+------+------+------+------+------+------+------+---
   |         ... 0 ...         |LLBK_2|DTR_2 |DSRS_2|RTS_2 |
   +------+------+------+------+------+------+------+------+--


       7      6      5      4      3      2      1      0
 --+------+------+------+------+------+------+------+------+
   |         ... 0 ...         |TXEN_3|TXEN_2|TXEN_1|TXEN_0|
---+------+------+------+------+------+------+------+------+
```

<15:12> Not used, read as zero.

LLBK_2 Local Loopback (bit 11). This read/write bit controls the state of the Local Loopback Modem Control signal (CCITT Circuit 141) for line 2. Setting this bit asserts the ON state of the LLBK signal. This bit is cleared by power-on; it is *not* cleared when the master clear bit SER_CSR<CLR> is set - see Section 10.4.1.

DTR_2 Data Terminal Ready (bit 10). This read/write bit controls the state of the Data Terminal Ready modem control signal (CCITT circuit 108/2) for line 2. Setting the bit asserts the ON state of the DTR signal. This bit is cleared by a power-on reset; it is *not* cleared when the master clear bit SER_CSR<CLR> is set - see Section 10.4.1.

DSRS_2 Data Rate Signalling Rate Selector (bit 9). This read/write bit controls the state of the Data Signalling Rate Selector modem control signal (CCITT circuit 111) for line 2. Setting the bit asserts the ON state of the signal. This bit is cleared by power-on; it is *not* cleared when the master clear bit SER_CSR<CLR> is set - see Section 10.4.1.

RTS_2 Request to Send (bit 8). This read/write bit controls the state of the Request to Send modem control signal (CCITT circuit 105) for line 2. Setting this bit asserts the ON state of this signal. This bit is cleared by power-on; it is *not* cleared when the master clear bit SER_CSR<CLR> is set - see Section 10.4.1.

<7:4> Not used, read as zero.

TXEN_x Transmitter Line Enable (bits 3:0). These read/write bits enable the transmitter logic for lines 3, 2, 1, and 0 respectively. Setting each of these bits causes the transmitter scanner to stop and assert the transmitter ready bit SER_CSR<TRDY> if the UART for that line has a transmitter buffer empty condition. The transmitter scanner resumes scanning when either the transmitter data register for the line at which the scanner stopped is loaded with another character, or when that line's transmitter line enable bit is cleared. A transmitter line enable bit should only be cleared while the scanner is not running (i.e. when the transmitter ready bit SER_CSR<TRDY> is set or the master scan enable bit SER_CSR<MSE> is clear). The transmitter line enable bits are cleared by a power-on reset and whenever the master clear bit SER_CSR<CLR> is set.

10.4.5 Modem Status Register (SER_MSR)

The Modem Status register is a 16-bit read-only register at address 200A.000Ch which contains the status of modem input signals for line 2. The ON condition of a modem signal is presented as the set state of the corresponding bit.

Figure 10–5: Serial Line Modem Status Register (SER_MSR)

```
      15     14     13     12     11     10     9      8
   +------+------+------+------+------+------+------+------+---
   |                           |SPDI_2| CD_2 |DSR_2 |CTS_2 |
   +------+------+------+------+------+------+------+------+--


       7      6      5      4      3      2      1      0
 --+------+------+------+------+------+------+------+------+
   |                           |   1  | RI_2 |   1  |TMI_2 |
---+------+------+------+------+------+------+------+------+
```

<15:11> Not used; read values undefined.

SPDI_2 Speed Mode Indicate (bit 11). This bit reflects the state of the Speed Mode Indicate signal from a modem (CCITT circuit 112) connected to line 2.

CD_2 Carrier Detect (bit 10). This bit reflects the state of the Carrier Detect signal from a modem (CCITT circuit 109) connected to line 2. The set state corresponds to the ON state of this signal.

DSR_2 Data Set Ready (bit 9). This bit reflects the state of the Data Set Ready signal from an external modem (CCITT circuit 107) on line 2. The set state corresponds to the ON state of this signal.

CTS_2 Clear to Send (bit 8). This bit reflects the state of the Clear to Send signal from an external modem (CCITT circuit 106) on line 2. The set state corresponds to the ON state of this signal.

<7:4> Not used; read values undefined.

<3> Reserved, reads as 1.

RI_2 Ring Indicator (bit 2). This bit reflects the state of the Ring Indicator signal from an external modem (CCITT circuit 125) on line 2. The set state corresponds to the ON state of this signal.

<1> Reserved, reads as 1.

TMI_2 Test Mode Indicate (bit 0). This bit reflects the state of the Test Mode Indicate signal from an external modem (CCITT circuit 142) on line 2. The set state corresponds to the ON state of this signal.

10.4.6 Transmitter Data Register (SER_TDR)

The Transmitter Data register is a 16-bit write-only register at address 200A.000Ch. It can be written on either a word or byte basis.

Figure 10–6: Serial Line Transmitter Data Register(SER_TDR)

```
      15     14     13     12     11     10      9      8
   +------+------+------+------+------+------+------+------+---
   |                           |BRK_3 |BRK_2 |BRK_1 |BRK_0 |
   +------+------+------+------+------+------+------+------+--

       7      6      5      4      3      2      1      0
 --+------+------+------+------+------+------+------+------+
   |                        TXDATA                         |
---+------+------+------+------+------+------+------+------+
```

<15:12> Not used.

<11:08> BRK_x Break Control. These write-only bits control the assertion of a BREAK condition on lines 3, 2, 1, and 0, respectively. Setting a bit immediately forces the transmitter output for the corresponding line to the SPACE condition. This condition will persist until the break control bit is cleared. These bits are cleared by a power-on reset and when the master clear bit SER_CSR<CLR> is set.

<07:00> TXDATA Transmitter Buffer. Data to be transmitted by a line's UART is loaded into these 8 bits. If the character width is 7, the unused bit is the high-order (bit 7) of the byte. This register may be written to only while the transmitter ready bit SER_CSR<TRDY> is set. The line to which the character is sent is indicated by the transmitter line number bits SER_CSR<TLINE>.

Chapter 11

SOUND GENERATOR

This chapter decribes the sound output capability of the VS4000 VLC. Sound output uses the DTMF tone generation capability of the 79C30 chip. Two tone generators may be individually programmed for frequency and amplitude; their outputs appear at the headphone jack mouted on the video module. The resolution of the frequency generators is eight bits, giving a frequency range of 8 Hz. to approx. 2 kHz.

11.1 Sound Chip Register Access

The 79C30 has some registers that are directly accessible and some that are accessed indirectly. These indirectly accessible registers are accessed by selecting one of several groups of registers and a register within that group with a write to the Command Register - address 200D.0000h. This register is eight bits wide and takes data from bits<7:0> of the longword specified. Following selection of a register within a group, one or more bytes of data are then transferred to load the selected register(s) - a function of which Group and which register within a Group. The data is transferred by writing one or more times to the Data Register - address 200D.0004h.

The Command Register format for selecting a register is as shown below :

Figure 11–1: Command Register Format - address 200D.0000h

```
   7    6    5    4    3    2    1    0
+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+
 \            / \                      /
  +----+-----+   +---------+----------+
       |                   |
     Selects         Selects Register
 Register Group      within Group
```

For tone generation there are three Register Groups that need to be accessed, the INIT, MUX and MAP groups. Refer to the 79C30 chip vendor specification for full details of all register groups.

11.1.1 INIT Register Group

This group is used to enable/disable the tone generators. The Register Group Select bits are set to <001> and the Register Select bits set to <00001>, i.e. a value of 21h is loaded into the Command Register.

The INIT register then is loaded with <00000001> to Enable the tone generators and <00000000> to Disable them.

11.1.2 MUX Registers

This Group is selected to set up the internal connections within the 79C30. The Register Group Select bits are set to <010> and the four MUX Control Registers (MCRs) selected in sequence by setting the Register Select bits to <00001> - MCR1, <00010> - MCR2, <00011> - MCR3 and <00100> - MCR4. Each register requires a single byte of data, zero for MCR1, 2 and 4 and 33h for MCR3. Thus the load sequence for the MCR registers is eight writes :

Table 11–1: MUX Register Load Sequence

Write Address	Value
200D.0000h	41h
200D.0004h	0
200D.0000h	42h
200D.0004h	0
200D.0000h	43h
200D.0004h	33h
200D.0000h	44h
200D.0004h	0

11.1.3 MAP Registers

Of the ten registers in the MAP Register Group - Main Audio Processor, some need only be initialixed, some need not be programmed for this mode of operation of the chip and four control the frequency and amplitude of the tones generated.

11.1.3.1 MAP Registers - Initialization

Table 11–2: MAP Register Initialization

Write Address	Value	Bytes
200D.0000h	66h	2
200D.0004h	08h	
200D.0004h	08h	
200D.0000h	69h	1
200D.0004h	48h	
200D.0000h	6Ah	1
200D.0004h	06h	

11.1.4 MAP Registers - Frequency and Amplitude Selection

Four of the MAP group registers select frequency (FTGR1 and 2) and amplitude (ATGR1 and 2) of the generated tones. The tone duration is set by how often the FTGR registers are re-loaded. There are no interrupts associated with this mode of operation of the 79C30.

The 79C30 has two frequency generators. When the FTGR registers have been selected (a write to 200D.0000h with a data value of 67h), the frequency produced by each tone generator when the FTGR register is loaded with a value of "i" is 1000*i/128 Hz. See Table 11–3 for a table of values that produce a chromatic scale. The frequencies of two tone generators are loaded by two successive writes to the Data Register following selection of the FTGR registers.

The ATGR registers allows the amplitude of the tones generated to be varied by 18 dB in 2 dB steps. See Table 11–4 for a list of values v amplitude. The ATGR registers are selected by loading the Command Register with a value of 68h, then writing the amplitude values by two successive writes to the Data Register.

11.2 FTGR Values v Frequency

The following table shows the value that must be loaded (in decimal) to each/either tone generator to produce a given frequency output. To obtain a single tone output, both generators should be loaded with the same value.

Table 11–3: Approximate Chromatic Scale

	Frequency	Actual		
Note	Required	Frequency	FTGR Value	Note
C	262	265.6	34	Middle C
C#	277	273.4	35	
D	294	296.9	38	
D#	311	312.5	40	
E	330	328.1	42	
F	349	351.6	45	
F#	370	367.2	47	
G	392	390.6	50	
G#	415	414	53	

Table 11–3 (Cont.): Approximate Chromatic Scale

Note	Frequency Required	Actual Frequency	FTGR Value	Note
A	440	437.5	56	
A#	466	468.8	60	
B	494	492.2	63	
C	523	523.4	67	
C#	554	554.7	71	
D	587	585.9	75	
D#	622	625	80	
E	659	656.3	84	
F	698	695.3	89	
F#	740	742.2	95	
G	784	781.3	100	
G#	831	828.1	106	
A	880	882.8	113	
A#	932	929.7	119	
B	988	984.4	126	

11.3 Amplitude Selection

The following table shows the value that must be loaded (in decimal) to set a particular amplitude of a generated tone.

Table 11–4: Amplitude v ATGR value

Gain (dB)	ATGR Value
-18	55
-16	50
-14	49
-12	39
-10	34
-8	33
-6	32
-4	18
-2	17
0	16

Chapter 12

TIME OF YEAR CLOCK

The time of year clock consists of an MC146818BM CMOS watch chip which keeps the date and time of day and contains 50 bytes of general purpose RAM storage. This chip includes a time base oscillator and a lithium battery on-chip. The battery powers the logic and oscillator while system power is off.

It is expected that data from the watch chip will be used by an operating system during its startup to determine the date and time, which the operating system will thereafter maintain by using interrupts from the interval timer. Therefore the watch chip's alarm and periodic interrupt features are not used in this system and the watch chip cannot generate a processor interrupt.

12.1 Battery Backup

A lithium battery within the watch chip supplies power to the watch chip and its time base oscillator while system power is off. The battery will maintain the clock operation and the data stored in the 50 bytes of RAM for a minimum of 10 years before it becomes exhausted.

12.2 Watch Chip Registers

The watch chip contains 64 8-bit registers. Ten of these contain date and time data, 4 are control and status registers, and the remaining 50 provide general purpose RAM storage. The registers occupy 64 consecutive longwords of address space as shown in the table below.

Each register is accessed as bits<9:2> of a longword (bits<31:10> and <1:0> are ignored on writing and undefined on reading).

WARNING

Because each register spans two bytes on the system bus, only *word* or *longword* access instructions may be used to manipulate these registers. The effects of using byte access instructions are undefined. In particular, instructions for modifying bits such as BBSS, BBSC, BBCC and BBCS cannot be used–they generate byte-access read-modify-write cycles which will corrupt the portion of the register which is not in the byte being accessed.

Table 12–1: TOY Chip Register Addresses

Address (HEX)	Name	Description
200B.0000	WAT_SEC	Time seconds, 0 . . . 59
200B.0004	WAT_ALMS	Alarm seconds (not used)
200B.0008	WAT_MIN	Time minutes, 0 . . . 59
200B.000C	WAT_ALMM	Alarm minutes (not used)
200B.0010	WAT_HOUR	Time hours, 0 . . . 23
200B.0014	WAT_ALMH	Alarm hours (not used)
200B.0018	WAT_DOW	Day of week, 1 . . . 7
200B.001C	WAT_DAY	Day of month, 1 . . . 31
200B.0020	WAT_MON	Month of year, 1 . . . 12
200B.0024	WAT_YEAR	Year of century, 0 . . . 99
200B.0028	WAT_CSRA	Time base divisor
200B.002C	WAT_CSRB	Date mode and format
200B.0030	WAT_CSRC	Interrupt flags (not used)
200B.0034	WAT_CSRD	Valid RAM and time flag
200B.0038		First byte of RAM data
.		
.		
.		
200B.00FC		Last byte of RAM data

12.3 Control and Status Registers

Figure 12–1: Watch Time Base Divisor (WATCSRA)

```
    9       8       7       6       5       4       3       2
+-------+-------+-------+-------+-------+-------+-------+-------+
|  UIP  |          DVX          |             RSX               |
+-------+-------+-------+-------+-------+-------+-------+-------+
```

UIP Update in progress (bit 9). This read-only bit indicates when the date and time registers are being updated and are hence unstable. It is set to one 244 microseconds before the beginning of an update cycle and remains one until the cycle is complete.

DVX Time base divisor (bits 8:6). These read/write bits set the amount by which the time base oscillator input to the watch chip is divided. These bits must be set to "010" to accomodate the 32.768 KHz time base in this system.

RSX Rate select (bits 5:2). These read/write bits select the rate at which the watch chip generates periodic interrupts. Since this feature is not used, these bits must be set to "0000" to disable it.

Figure 12–2: Watch Date Mode and Format (WATCSRB)

```
     9       8       7       6       5       4       3       2
 +-------+-------+-------+-------+-------+-------+-------+-------+
 |  SET  |  PIE  |  AIE  |  UIE  | SQWE  |  DM   | 24/12 |  DSE  |
 +-------+-------+-------+-------+-------+-------+-------+-------+
```

SET Set Time (bit 9). When this read/write bit is zero, the time and date registers are updated once per second. When this bit is one, any update cycle in progress is aborted and updates are inhibited so that a program can set new date and time values.

PIE Periodic Interrupt Enable (bit 8). Not used; must be set to 0.

AIE Alarm Interrupt Enable (bit 7). Not used; must be set to 0.

UIE Update Interrupt Enable (bit 6). Not used; must be set to 0.

SQWE Square-wave Enable (bit 5). Not used; must be set to 0.

DM Data Mode (bit 4). This read/write bit selects the numeric representation in the time and date registers. If DM is one, the data format is binary; if DM is zero, the data format is two 4-bit decimal digits (BCD).

24/12 Hours Format (bit 3). This read/write bit selects the format of the WAT_HOUR AND WAT_ALMH registers. A value of one selects 24-hour mode; a value of zero selects 12-hour AM/PM mode. In the latter case, bit 7 of the hours registers is zero for AM and one for PM.

DSE Daylight Saving Enable (bit 2). This read/write bit is zero for normal operation. If set to one, two special time updates occur: on the last Sunday in April the time increments from 01:59:59 AM to 03:00:00 AM, and on the last Sunday in October when the time first reaches 01:59:59 AM it changes to 01:00:00 AM. (This feature is obsolete, since it does not conform to current start and end dates for Daylight Savings Time)

Figure 12–3: Watch Valid RAM and Time Flag (WATCSRD)

```
     9       8       7       6       5       4       3       2
 +-------+-------+-------+-------+-------+-------+-------+-------+
 |  VRT  |   0   |   0   |   0   |   0   |   0   |   0   |   0   |
 +-------+-------+-------+-------+-------+-------+-------+-------+
```

VRT Valid RAM and Time (bit 9). This bit indicates whether the contents of the time and RAM registers may have been corrupted by loss of power. This bit is set to zero whenever system power is off and the backup battery voltage drops below the value required for the watch chip to function properly. This bit is set to one after any read of this register (the register may not be written). Programming note: Since ANY read of this register to test the VRT bit will reset that bit to one, a program which finds VRT equal to zero must be prepared to either load valid data into the time and RAM registers or take other action to indicate that the contents of the watch chip are invalid.

<8:2> Not used. Always read as zeros.

12.4 Time of Year Data

The time of year is kept in six registers: WAT_SEC, WAT_MIN, WAT_HOUR, WAT_DAY, WAT_MON, and WAT_YEAR. A seventh register, WAT_DOW, indicates the day of the week (days are numbered from 1 (Sunday) through 7). The contents of each register may be in either binary form or BCD (two 4-bit decimal digits) as selected by register WAT_CSRB bit DM.

The time value is incremented once each second. Such an update requires 1948 microseconds, during which time the date and time register contents are unstable and should not be read by a program. Register WAT_CSRA bit UIP indicates when an update is in progress. This bit is one from 244 microseconds before the beginning of an update cycle until the cycle is complete. Therefore a program should read WAT_CSRA until it finds bit UIP zero, at which time it has at least 244 microseonds to read the date and time registers. The program should inhibit interrupts while reading the registers to ensure that an interrupt does not prolong its reading beyond the 244 microsecond window.

12.5 Non-volatile RAM Storage

The 50 bytes of RAM storage are used by system firmware. The use of these bytes is defined in the KA48 System Firmware specification.

12.6 Initialization

When a program finds the VRT bit equal to zero, it must assume that the contents of all other registers in the chip are invalid. To initialize the chip, a program should:

1. Load register WAT_CSRB with bit SET equal to one to inhibit time updates and bits PIE, AIE, UIE and SQWE equal to zero to disable unused features. Bits TM, 24/12 and DSE should be set for the desired date format.
2. Load the seven time registers with the current date and time.
3. Load register WAT_CSRA to set the proper time base divisor. The DVX bits should be set to "010" and the RSX bits to "0000".
4. Load register WAT_CSRB with the same value used in step 1 except that bit SET should now be zero to enable normal time updating.

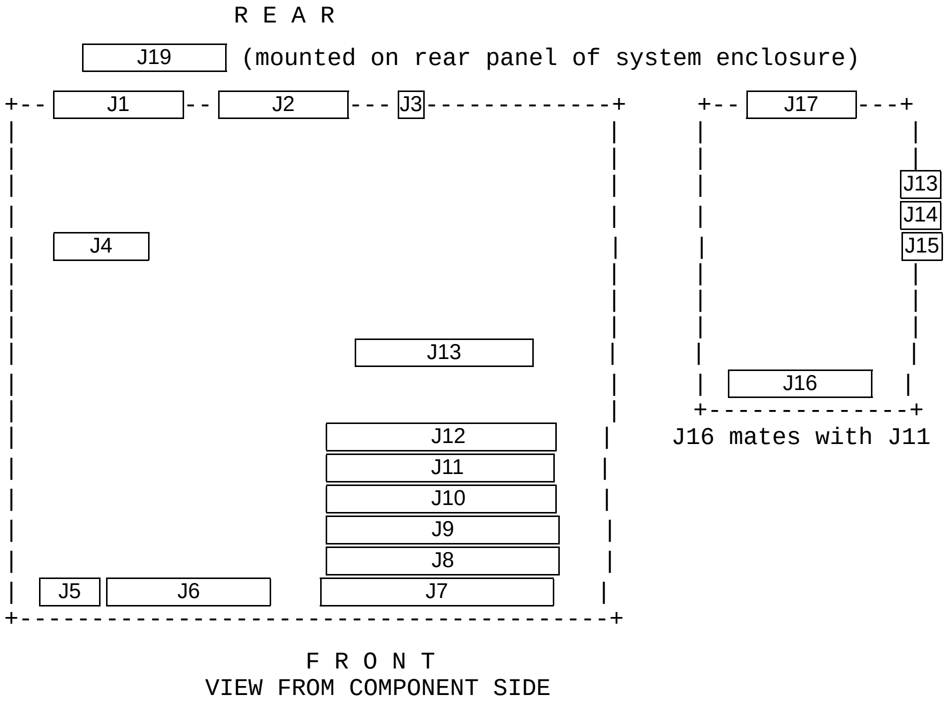
As long as the backup battery voltage is sufficient, the contents and operation of the watch chip are not affected by system power-on and power-off events.

Chapter 13

Connectors

The VS4000 VLC system board carries 13 connectors. Some of these are directly accessible at the rear of the machine, some are for connection of internal options only. The graphics daughter module carries a further five connectors.

Figure 13–1: System and Video Board connector Placement



NOTE: Connector numbers do not reflect actual component numbers.

13.1 Power - J4

The power supply connects to the system module by this 14-pin connector. Power for the internal SCSI devices runs via etch on the system module to J5 - see below.

Table 13–1: Power Pinout

Pin	Signal
1	-12V
2	GND
3	GND
4	GND
5	+5V
6	+5V
7	NC
8	+5V
9	+12V
10	NC
11	NC
12	GND
13	-9V
14	-9V RET

13.2 Memory - J<7:12>

There are six, 72-pin SIMM connectors to allow memory to be added to the system module. The rules for population of these connectors are given at the start of Chapter 4. As memory is always added as pairs of SIMMs, there are two pinouts; J7, J9 and J11 carry the high 32-bits of memory data; J8, J10 and J12 the low 32-bits.

Table 13–2: Memory SIMM Pinout - J7, 9, 11

Pin	Signal	Comments
1	GND	
2	DATA>32>	IO
3	DATA<48>	IO
4	DATA<33>	IO
5	DATA<49>	IO
6	DATA<34>	IO
7	DATA<50>	IO
8	DATA<35>	IO
9	DATA<51>	IO
10	+5V	
11	NC	
12	ADDR<0>	
13	ADDR<1>	
14	ADDR<2>	
15	ADDR<3>	
16	ADDR<4>	
17	ADDR<5>	
18	ADDR<6>	
19	NC	
20	DATA<36>	IO
21	DATA<52>	IO
22	DATA<37>	IO
23	DATA<53>	IO
24	DATA<38>	IO
25	DATA<54>	IO
26	DATA<39>	IO
27	DATA<55>	IO
28	ADDR<7>	
29	NC	
30	+5V	
31	ADDR<8>	
32	ADDR<9>	
33	RAS_L	
34	RAS_L	
35	PARITY>6>	
36	PARITY>4>	
37	PARITY>5>	
38	PARITY>7>	
39	GND	
40	CAS<6>	
41	CAS<5>	

Table 13–2 (Cont.): Memory SIMM Pinout - J7, 9, 11

Pin	Signal	Comments
42	CAS<7>	
43	CAS<4>	
44	RAS_L	
45	RAS_L	
46	NC	
47	WRITE_L	
48	NC	
49	DATA<40>	IO
50	DATA<56>	IO
51	DATA<41>	IO
52	DATA<57>	IO
53	DATA<42>	IO
54	DATA<58>	IO
55	DATA<43>	IO
56	DATA<59>	IO
57	DATA<44>	IO
58	DATA<60>	IO
59	+5V	
60	DATA<61>	IO
61	DATA<45>	IO
62	DATA<62>	IO
63	DATA<46>	IO
64	DATA<63>	IO
65	DATA<47>	IO
66	NC	
67	NC	
68	NC	
69	MEMSIZE	GND, Indicates SIMM present
70	MEMTYP	Indicates 1 or 4 MB SIMM type
71	NC	
72	GND	

Table 13–3: Memory SIMM Pinout - J8, 10, 12

Pin	Signal	Comments
1	GND	
2	DATA>00>	IO
3	DATA<16>	IO
4	DATA<01>	IO
5	DATA<17>	IO
6	DATA<02>	IO
7	DATA<18>	IO
8	DATA<03>	IO
9	DATA<19>	IO
10	+5V	
11	NC	
12	ADDR<0>	

Table 13–3 (Cont.): Memory SIMM Pinout - J8, 10, 12

Pin	Signal	Comments
13	ADDR<1>	
14	ADDR<2>	
15	ADDR<3>	
16	ADDR<4>	
17	ADDR<5>	
18	ADDR<6>	
19	NC	
20	DATA<04>	IO
21	DATA<20>	IO
22	DATA<05>	IO
23	DATA<21>	IO
24	DATA<06>	IO
25	DATA<22>	IO
26	DATA<07>	IO
27	DATA<23>	IO
28	ADDR<7>	
29	NC	
30	+5V	
31	ADDR<8>	
32	ADDR<9>	
33	RAS_L	
34	RAS_L	
35	PARITY>2>	
36	PARITY>0>	
37	PARITY>1>	
38	PARITY>3>	
39	GND	
40	CAS<6>	
41	CAS<5>	
42	CAS<7>	
43	CAS<4>	
44	RAS_L	
45	RAS_L	
46	NC	
47	WRITE_L	
48	NC	
49	DATA<08>	IO
50	DATA<24>	IO
51	DATA<09>	IO
52	DATA<25>	IO
53	DATA<10>	IO
54	DATA<26>	IO
55	DATA<11>	IO
56	DATA<27>	IO
57	DATA<12>	IO
58	DATA<28>	IO
59	+5V	
60	DATA<29>	IO

Table 13–3 (Cont.): Memory SIMM Pinout - J8, 10, 12

Pin	Signal	Comments
61	DATA<13>	IO
62	DATA<30>	IO
63	DATA<14>	IO
64	DATA<31>	IO
65	DATA<15>	IO
66	NC	
67	NC	
68	NC	
69	NC	
70	NC	
71	NC	
72	GND	

13.3 SCSI - J6

This 50 pin connector is used to cable to the (optional) internal SCSI disk drive and then on to the external SCSI connector (J19) mounted at the rear of the system enclosure. Termination for this end of the SCSI bus is provided on the system module close to this connector. The other end id terminated either at the extrenal connector or at the continuation connector at the rear of any expansion enclosure.

Table 13–4: SCSI Bus Pinout - J6

Pin	Signal	Comments
1	GND	
2	SCD<0>	See Chapter 9, Section 9.2
3	GND	
4	SCD<1>	
5	GND	
6	SCD<2>	
7	GND	
8	SCD<3>	
9	GND	
10	SCD<4>	
11	GND	
12	SCD<5>	
13	GND	
14	SCD<6>	
15	GND	
16	SCD<7>	
17	GND	
18	SCDIP	
19	GND	
20	GND	
21	GND	
22	GND	

Table 13–4 (Cont.): SCSI Bus Pinout - J6

Pin	Signal	Comments
23	GND	
24	GND	
25	N.C.	
26	TERMPWR	
27	GND	
28	GND	
29	GND	
30	GND	
31	GND	
32	SCATN	
33	GND	
34	GND	
35	GND	
36	SCBSY	
37	GND	
38	SCACK	
39	GND	
40	SCRST	
41	GND	
42	SCMSG	
43	GND	
44	SCSEL	
45	GND	
46	SCC/D	
47	GND	
48	SCREQ	
49	GND	
50	SCI/O	

13.4 External SCSI Connector - J19

J17 is a Champ^{tm} connector that allows external SCSI devices to be attached to the system box. Termination is provided at the far end of any option attached or by inserting a terminator into this connector if no external options are used. The pinout is defined by the ANSI SCSI Specification.

13.5 SCSI Power - J5

A single 4-pin power connector supplies SCSI device power from the system module.

Table 13–5: SCSI Power Pinout - J5

Pin	Signal	Comments
1	+12V	
2	GND	
3	GND	

Table 13–5 (Cont.): SCSI Power Pinout - J5

Pin	Signal	Comments
4	+5V	

13.6 Video Output - J18

J18 is a 9-pin (54-20772 module) or 15-pin (54-20774 modeule) D-subminiature connector that carries the three video signals to a monitor.

Table 13–6: Video Connector

Pin	Signal on 9-pin	Signal on 15-pin
1	GND	RED
2	BLUE	GREEN
3	GND	BLUE
4	GND	GND
5	GREEN	GND
6	GND	GND
7	GND	GND
8	RED	GND
9	GND	GND
10	–	GND
11	–	NC
12	–	GND
13	–	SYNC
14	–	GND
15	–	GND

13.7 Local Area Network - J1

J1 is a 15-pin D-subminiature connector, the Transceiver connect for the Ethernet LAN.

13.8 Communications - J3

J3 is a 25-pin D-subminiature connector allowing modem connect for serial line #2 of the four serial lines supported as a standard part of the system module. See Chapter 10 for signal definitions.

Table 13–7: Communications Connector

Pin	Signal	DIR	Comments
1	N.C.		May be connected to GND if W1 is installed
2	XMT	OUT	CCITT Circuit 103
3	RCV	IN	CCITT Circuit 104

Table 13–7 (Cont.): Communications Connector

Pin	Signal	DIR	Comments
4	RTS	IN	CCITT circuit 105
5	CTS	IN	CCITT Circuit 106
6	DSR	IN	CCITT Circuit 107
7	GND	NA	CCITT Circuit 102
8	DCD	IN	CCITT Cicruit 109
9	NC		
10	NC		
11	NC		
12	SPDI	IN	CCITT Cicruit 112
13	NC		
14	NC		
15	NC		
16	NC		
17	NC		
18	LLBK	OUT	CCITT Cicruit 141
19	NC		
20	DTR	OUT	CCITT Circuit 108/2
21	NC		
22	RI	IN	CCITT Circuit 125
23	DSRS	OUT	CCITT Circuit 111
24	NC		
25	TMI	IN	CCITT Cicruit 141

13.9 Printer - J2

J2 is a 6-pin MMJ jack supporting DEC423 data leads only connection. It is intended for use with a local printer and is connected to serial line #3 of the four serial lines supported as a standard part of the system module.

Table 13–8: DEC423 Connector

Pin	Signal	DIR	Comments
1	HIGH	OUT	150 Ω to +5V
2	XMT	OUT	
3	GND	NA	
4	RCV+	IN	
5	RCV-	IN	
6	TERM	IN	3K Ω to GND

13.10 Keyboard - J14

J14 is a 4-pin MJ jack intended for the connection of an LK401 keyboard. It is connected to serial line #0 of the four serial lines supported as a standard part of the system module.

Table 13–9: Keyboard Connector

Pin	Signal	DIR	Comments
1	RCV	IN	
2	+12V	OUT	Protected by self-resetting fuse
3	GND	NA	
4	XMT	OUT	

13.11 Mouse/Tablet - J15

J15 is an 8-pin DIN connector intended for connection of a VSXXX-AA or VSXXX-AB. It is connected to serial line #1 of the four serial lines supported as a standard part of the system module.

Table 13–10: Keyboard Connector

Pin	Signal	DIR	Comments
1	GND	NA	
2	XMT	OUT	
3	RCV	IN	
4	-12V	OUT	Protected by self-resetting fuse
5	+5V	OUT	Protected by self-resetting fuse
6	+12V	OUT	Protected by self-resetting fuse
7	NC		
8	NC		
9	NC		

13.12 Sound Connector - J16

J16 is a 4-pin MJ connector that carries audio IN and OUT.

Appendix A

PHYSICAL ADDRESS MAP

A.1 System Module

The following addresses are defined for those sections of the VS4000 VLC integral to the KA48-AA System Module.

Address(es) (HEX)	Name	R/W	Useage	Reference
0000.0000 - 00FF.FFFF		(RW)	Read/Write memory	Chapter 4
2000.0000 - 2001.FFFF		(RW)	DMA Translation Map Registers	Chapter 4 Section 4.4
2002.0000	CFGTST	(R)	Configuration/Test Register	Chapter 6 Section 6.2
2002.0000	IORESET	(W)	I/O Reset	Chapter 6, Section 6.1
2020;0000 - 2020.0FFF		(RW)	Invalidate Filter RAM	Chapter 4 Section 4.3
2004.0000 - 2007.FFFF		(R)	System ROM (256K Bytes)	Chapter 5
2008.0000	HLTCOD1	(RW)	Halt Code Register1	Chapter 6 Section 6.4
2008.0004	HLTCOD2	(RW)	Halt Code Register2	Chapter 6 Section 6.4
2008.0008	MAP_BASE	(RW)	DMA Translation Map Base	Chapter 4 Section 4.4
2008.000C	INTMSK	(RW)	Interrupt Mask Register	Chapter 3 Section 3.6
2008.000E			RESERVED	
2008.000F	INTREQ	(R)	Interrupt Request Pending	Chapter 3 Section 3.4
2008.000F	INTREQ	(W)	Interrupt Request Clear	Chapter 3 Section 3.5
2008.0010	DIAG_DISP	(W)	Diagnostic LEDs	Chapter 6, Section 6.5

Address(es) (HEX)	Name	R/W	Useage	Reference
2008.0014	BWF0	(RW)	Miscellaneous Control Register	Chapter 6 Section 6.6
2008.0018	CAC_LIM.	(RW)	Cache Active Set Limit Register	Chapter 4 Section 4.3.1
2008.001C	DIAGTIMU	(RW)	Diagnostic Timer, High Resolution	Chapter 6 Section 6.7
2008.001E	DIAGTIMM	(RW)	Diagnostic Timer, Low Resolution	Chapter 6 Section 6.7
2009.0000 - 2009.007F		(R)	Network Address ROM	Section 5.2
200A.0000	SER_CSR	(RW)	DZ CSR	Chapter 10 Figure 10–1
200A.0004	SER_RBUF	(R)	DZ Receive Buffer Register	Chapter 10 Section 10.4.2
200A.0004	SER_LPR	(W)	DZ Receive Line Parameter Register	Chapter 10 Section 10.4.3
200A.0008	SER_TCR		DZ Transmit Control Register	Chapter 10 Section 10.4.4
200A.000C	SER_MSR	(R)	DZ Modem Ststus Register	Chapter 10 Figure 10–5
200A.000C	SER_TDR		DZ transmit Data Register	Chapter 10 Section 10.4.6
200A.0010	SER_LPR0	(R)	DZ Line Parameter Register 0	Chapter 10 Section 10.4.3
200A.0014	SER_LPR1	(R)	DZ Line Parameter Register 1	Chapter 10 Section 10.4.3
200A.0018	SER_LPR2	(R)	DZ Line Parameter Register 2	Chapter 10 Section 10.4.3
200A.001C	SER_LPR3	(R)	DZ Line Parameter Register 3	Chapter 10 Section 10.4.3
200B.0000 - 200B.00FF		(RW)	TOY Clock/NVR	Chapter 12
200C.0000	SCSI_ADR	(RW)	SCSI controller DMA Address Register	Chapter 9 Figure 9–1
200C.000C	SCSI_DIR	(RW)	SCSI DMA Direction Register	Chapter 9 Figure 9–2
200C.0080 - 200D.00BF	SCSI_XXX	(RW)	SCSI Controller Chip Registers	Chapter 9 Section 9.3.1
200D.0000 - 200D.00FF		(RW)	Sound Chip Registers	Chapter 11 Section 11.1
200E.0000 - 200E.0007		(RW)	LANCE Chip Registers	Chapter 8 Section 8.3
200F.0000 - 200F.00FF	VCR	(R)	Video Configuration Register	Chapter 6 Section 6.3
2010.0000 - 2017.FFFF			GC Registers and ROM	
2018.0000 - 201F.FFFF			GC FIFO	
2100.0000 - 22FF.FFFF		(RW)	Video Frame Buffer	
2C00.0000 - 2CFF.FFFF			RESERVED	
3800.0000 - 38FF.FFFF			RESERVED	

Appendix B

REVISION HISTORY

Revision X00—Initial review draft

Revision X01—Video becomes daughter card, support for two video frequencies, different memory SIMMs. Maximum memory increased to 24 MB. Configuration register changed to indicate SIMM type.