DECstation 5000/200 KN02 System Module Functional Specification

Revision 1.3

Workstation Systems Engineering
Digital Equipment Corporation
100 Hamilton Avenue
Palo Alto, CA 94301

August 27, 1990

August 1990

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 responsibilty for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Digital Equipment Coporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital or its affiliated companies.

Blank Page

Table of Contents

1. DS5000/200 KN02 System Module Functional Specification

2. External Interface

3. Power Requirements

4. Subsystem Summary

Revision History

Date	Version	Content/Changes
14 Aug 89	1.0	First release
24 Sep 89	1.1	Removed instruction streaming reference
		Added additional ROM address mapping to 0x1fc00000
		Modified CSR TXDIS, LEDIAG, CORRECT, ECCMD to be r/w
		Added CSR IOINTEN<7:0>, REFEVEN, PSWARN
		Removed CSR ROMWP
		Added DMA overrun errors
		Modified definition of ERRADR ECCERR bit
		Modified CPU writes to be lower priority than DMA
		Added note to indicate I/O space is not cacheable
26 Feb 90	1.2	Changed document to Company Confidential
		Finished all TBD references.
		Replaced references to write buffer with memory buffer.
		Replaced references to MAXbus with TURBOchannel.
		Changed UNSCUR bit to PRSVNVR bit in status register.
23 Aug 90	1.3	Changed document to not confidential
		Changed names to DS5000/200 and appropriate module names.

Blank Page

1. DS5000/200 KN02 System Module Functional Specification

This document is the functional specification of the DS5000/200 KN02 system module. The KN02 is the core module used to construct DS5000/200 workstations and servers. The KN02 processor has ~20 times the processor performance of a VAX 11/780, 8 to 480 Mbytes of memory, an integral SCSI interface, an integral Ethernet interface, an integral quad-serial interface, 256 KBytes of ROM, miscellaneous system support functions, and three TURBOchannel I/O option slots. The TURBOchannel slots allow a range of system configurations such as 2D/3D workstations and servers. This specification describes the functionality of the KN02 and the programmer's interface.

2. External Interface

The KN02 rear bulkhead has the following external interfaces:

- AC power switch
- AC line receptacle
- AC convenience receptacle
- Mouse/keyboard DB15 male connector with EIA-232-D TX/RX signals, +12 Volts at 600 mAmps for the keyboard/tablet, +5.3 Volts at 150 mAmps for the mouse, and -12 Volts at 20 mAmps for the mouse
- Communications DB25 male connector with DTR, RTS, TX, RX, CTS, DSR, CD, and RI EIA-232-D signals at up to 38.4 KBaud
- Printer DB25 male connector with DTR, RTS, TX, RX, CTS, DSR, CD, and RI EIA-232-D signals at up to 38.4 KBaud
- Momentary push-button reset switch
- 8 red diagnostic LEDs
- ThinWire Ethernet BNC female connector to internal ThinWire transceiver at 10 MBits/second
- SCSI-2 high-density female connector for synchronous-mode, 8-bit data bus at up to 5 MBytes/second
- 3 TURBOchannel option bulkhead slots

3. Power Requirements

DS5000/200 electronics has a theoretical maximum power dissipation of 225 Watts. Table 3-1 lists the theoretical worst case power supply requirements for a 480-MByte DS5000/200. The worst case supply requirements include power provided to external LK201 keyboard, VSXXX-AA mouse, external SCSI terminator, and three TURBOchannel options. The typical supply currents include power for a keyboard and mouse. All currents are for steady state conditions.

Table 3-1: Supply Requirements

Supply (Volts)	Maximum Current (Amps)	Typical Current (Amps)
+5.1 +/-4%	20.0	16.3
+12.1 +/-4%	2.70	0.08
-12.1 +/-4%	0.22	0.04

Table 3-2 lists the maximum power that TURBOchannel options may draw through their system connector(s).

Table 3-2: Maximum TURBOchannel Option Power

Supply (Volts)	Single Option (Amps)	Double Option (Amps)	Triple Option (Amps)
+5 +/-5%	4.0	8.0	12.0
+12 +/-5%	0.5	1.0	1.5

4. Subsystem Summary

The KN02 implements the following subsystems:

- R3000/R3010/R3220 25 MHz CPU/FPU/MB
- 64 KByte instruction cache
- 64 KByte write-through data cache
- 15 memory array module connectors
- 256 KByte power-up self test and bootstrap ROM
- System control and status register and diagnostic LEDs
- RTC-based system clock and 50-byte battery-backed-up (BBU) RAM
- DZ-based serial lines for keyboard, mouse, communications, and printer
- Error address status register
- ECC error check/syndrome status register
- LANCE-based network interface for ThinWire Ethernet
- Disk/tape interface for SCSI peripherals
- 3 TURBOchannel I/O option connectors
- Reset switch

4.1. Address Space

The R3000 CPU has a 4 GByte virtual address space consisting of four regions. Table 4-1 lists these regions. The KSEG0, KSEG1, and KSEG2 regions are only accessible while the R3000 is in kernel mode. Refer to the *MIPS R2000 RISC Architecture* for a detailed discussion of its virtual address space.

The R3000 CPU is configured for little-endian byte order. All address space descriptions in this document are correspondingly little-endian.

Table 4-1: R3000 Virtual Address Space

Address Range	Size (GBytes)	Region	Properties
0x00000000..0x7FFFFFFF	2.0	KUSEG	Mapped and cached
0x80000000..0x9FFFFFFF	0.5	KSEG0	Unmapped and cached
0xA0000000..0xBFFFFFFF	0.5	KSEG1	Unmapped and uncached
0xC0000000..0xFFFFFFFF	1.0	KSEG2	Mapped and cached

The KN02 has a 512 MByte physical address space. Physical addresses beyond this range are reserved and must not be referenced. Table 4-2 summarizes the physical address space decoding of the KN02.

Table 4-2: KN02 Physical Address Space

Address Range	Size (MBytes)	Region	Subsystem
0x00000000..0x1DFFFFFF	480	Memory	Array slots 0 to 14
0x1E000000..0x1E3FFFFF	4	I/O Slot 0	TURBOchannel slot 0
0x1E400000..0x1E7FFFFF	4	I/O Slot 1	TURBOchannel slot 1
0x1E800000..0x1EBFFFFF	4	I/O Slot 2	TURBOchannel slot 2
0x1EC00000..0x1EFFFFFF	4	I/O Slot 3	Reserved
0x1F000000..0x1F3FFFFF	4	I/O Slot 4	Reserved
0x1F400000..0x1F7FFFFF	4	I/O Slot 5	SCSI interface
0x1F800000..0x1FBFFFFF	4	I/O Slot 6	Ethernet interface
0x1FC00000..0x1FFFFFFF	4	I/O Slot 7	System interface

The 15 memory banks are double mapped with both 8-Mbyte and 32-Mbyte strides in the first 480 Mbytes of physical address space. The bank stride is set by the CSR<BANK32M> bit. At powerup/reset, the CSR is cleared resulting in an 8 MByte stride. The programmable stride allows systems with only 8-Mbyte modules, or only 32-Mbyte modules, to have physically contiguous memory. The hardware supports mixtures of 8 and 32 Mbyte modules. 8 Mbyte modules are aliased four times when referenced with 32-Mbyte bank stride.

The 8 I/O slots each have a 4 MByte address region. I/O slots 0 through 2 are optionally implemented via TURBOchannel option modules. I/O slots 3 through 7 are integral to the KN02 module. I/O space may not be accessed through cached address space.

The KN02 powerup ROM software may test each memory slot to automatically determine the memory configuration. Each TURBOchannel option, and integral options, specify via their ROM, module characteristics to support automatic configuration of the I/O subsystems.

4.2. Interrupts

Due to the increased number of I/O devices supported in DS5000/200 systems, all I/O slot interrupts are merged into a single R3000 interrupt. However, the individual I/O slot interrupts are visible through CSR<7:0>. The FPU, memory subsystem, and system clock interrupts are direct R3000 interrupt pins as shown in Table 4-3 An interrupt signal only generates a R3000 interrupt if it is enabled in the STATUS register interrupt mask field, and interrupts are enabled by the STATUS<0> register bit. The interrupt signals are visible in the CAUSE register regardless of the state of the STATUS interrupt mask. That is, the operating system interrupt dispatcher must explicitly check that a given interrupt level, which is asserted in the CAUSE register, is enabled before activating that interrupt level’s handler.

Table 4-3: R3000 Interrupts

Level	CAUSE/STATUS bit	Source
5	15	FPU
4	14	Reserved
3	13	Memory controller
2	12	Reserved
1	11	RTC
0	10	I/O slots

The R3000 CAUSE register continually reflects the state of the interrupt signals. The CSR<7:0> interrupt field continually reflects the state of the I/O slot interrupt signals. The interrupt handling software may service or disable the interrupting device to clear the interrupt signals. Alternatively, each I/O slot interrupt may be individually masked from R3000 interrupt level 0, by clearing the corresponding IOINTEN<7:0> bit in the CSR.

4.3. Processor Subsystem

The processor subsystem implements a complete MIPS processor via a R3000 central processing unit (CPU), a R3010 floating point unit (FPU), a R3220 six-stage write/memory buffer (MB), a 64 KByte instruction cache, a 64 KByte write-through data cache, and associated clock and configuration logic.

The R3000 and R3010 implement the MIPS R-Series MACH-1 instruction set. The R3000 is configured for partial write fixups. The R3000 multiprocessor features are not supported. Consequently, the caches must be flushed under software control after I/O device DMA into memory. Refer to the *MIPS Architecture LR3000 Processor and LR3010 Coprocessor Interface* and the *MIPS R-Series Processor Architecture* for further details.

The instruction and data caches are configured with a four-word line size with loads and stores nominally completing in one cycle. Instruction and data cache fills take advantage of page mode memory cycles to complete a four-word fill in 11 access latency cycles, 4 data transfer cycles, plus miss and memory latency overhead. This results in a peak memory read bandwidth of 21 MBytes/second with a 25 MHz system clock.

The tag and data stores of each cache are byte-parity protected, with cache parity errors transparently generating cache misses to reload the cache from memory. Operating system software should poll the CPU STATUS<PE> bit in its clock-interrupt handler to detect excessive rates of cache parity errors.

The R3220 buffers CPU/FPU stores for up to six pending writes. There is no gathering or reordering of writes in the R3220. The data consistency conflicts between reads of write-buffered locations are automatically resolved by the memory controller. The R3220 takes advantage of the memory system’s page mode cycles to retire writes within the same page each cycle. This results in a peak write bandwidth of 100 MBytes/second with a 25 MHz system clock.

For I/O register writes that have side effects at other I/O register addresses, and hence must complete before I/O reads can be issued, software can determine whether or not all writes have completed by branching on the coprocessor zero condition (BC0F branches if MB is non-empty). Due to pipelining in the R3220 and the memory controller, there is a three cycle latency between issuing writes and a valid coprocessor condition indication. This latency will normally be buried in the procedure call overhead of a *wbFlush* routine.

4.4. Memory Subsystem

The memory subsystem is ECC protected on each 32-bit memory word for single-bit-error correction and double-bit-error detection (7 check bits). System interface status registers preserve the address, check bits, and syndrome when errors occur, as well as generating an interrupt/exception to the processor.

The memory system implements two-way, low-order interleaving to supply or accept one word each cycle during page mode cache fills, CPU writes, and TURBOchannel DMA. The memory system performs non-page-mode references to support uncached memory reads, and writes to different pages. The memory system also supports word read-modify-write cycles to support R3000 byte, half-word, and tri-byte store instructions. However, partial writes are very inefficient and should be eliminated from software wherever possible.

The memory system supports TURBOchannel page-mode DMA transfers of 1 through 128 words. TURBOchannel DMA arbitration is fixed priority with option slot 2 having highest priority and slot 0 having lowest priority.

Memory bank refresh transactions occupy the memory system for 5 cycles, and are scheduled every 195 cycles.

4.4.1. Memory Access Priority

Table 4-4 summarizes the service priority of the memory controller, with priority 1 being highest.

Table 4-4: Memory Access Priority

Priority	Source
1	DRAM refresh
2	CPU reads
3	Slot 2 DMA
4	Slot 1 DMA
5	Slot 0 DMA
6	MB writes

4.4.2. Memory System Performance

Table 4-5 summarizes the performance of the memory system itself. There is a latency of two cycles for CPU writes to the MB, before the memory controller receives the write request, that is not included in the calculations. The R3000 fixup/load cycles are not included in the calculations. There are also 4 cycles required to drain the memory write pipeline after CPU page-mode writes and DMA writes until it can service a new transaction; this overhead is not included in the calculations.

Table 4-5: Memory System Performance

Transaction	Cycles	Bandwidth MB/S (25 MHz Clock)
Uncached CPU read	10	10.0
CPU cache fill	13	30.8
Partial writes	11	9.1
Non-page writes	5	20.0
Page writes	1	100.0
DMA 1-word read	11	9.1
DMA 128-word read	138	92.8
DMA 1-word write	7	14.3
DMA 128-word write	134	95.5
Refresh	5	N/A

4.4.3. Memory System Interrupts

The memory system detects three classes of errors: I/O timeouts, memory read ECC errors, and DMA overrun errors. References to unpopulated TURBOchannel slots, reserved I/O slots, and holes in I/O subsystems will be aborted by a TURBOchannel timer after 10 microseconds. Whenever memory is read, the data and check bits are processed by ECC logic that determines if no errors are present, a single-bit is in error which is transparently corrected, or two or more bits are in error. Also, DMA cycles that attempt to read or write more than 128 words will be aborted by a TURBOchannel counter.

The memory system generates an interrupt if TURBOchannel I/O read/write timeouts, CPU memory read ECC errors, CPU partial write ECC errors, DMA read ECC errors, or TURBOchannel DMA overrun errors occur. In addition, provided error status is not already preserved, the ERRADR register updates to reflect: whether it was a CPU or DMA transaction; whether it was a read or write transaction; whether it was a TURBOchannel timeout/overrun or memory ECC error; and the address. In the case of ECC errors, the CHKSYN register is frozen with the value of memory bank check and syndrome bits for the read cycle. The memory interrupt may be cleared by writing the ERRADR register.

If a timeout error occurs on a CPU I/O read, or a multiple-bit ECC error occurs on a CPU memory read transaction, the memory controller also generates a bus error exception (which suppresses any cache fills of erroneous data). If a multiple-bit ECC error occurs on a DMA read, the memory controller asserts the ~err signal while it drives those words on the TURBOchannel. TURBOchannel DMA transactions that overrun the maximum transfer length are terminated with the ~err signal.

Refer to the ERRADR and CHKSYN register descriptions for further details.

4.4.4. Memory Array Module (MS02)

Each memory array module contains two interleaved banks of ECC memory. With 1-Mbit dynamic RAMs, the memory array module implements 8 Mbytes of memory. This allows system configurations from 8 Mbytes to 120 Mbytes. With 4-Mbit dynamic RAMs, the memory array module implements 32 Mbytes of memory. This allows system configurations from 32 Mbytes to 480 Mbytes. The system module supports mixtures of 8-Mbyte and 32-Mbyte memory array modules.

Refer to the *DS5000/200 Memory Array Card (MS02) Specification* for further details.

4.4.5. ECC Logic

The ECC logic uses seven check bits to correct single-bit errors, detect all double-bit errors, and detect some multi-bit errors on each 32-bit data word. Table 4-6 lists the modified Hamming Code used to generate the check bits during memory writes.

Table 4-6: Check Bit Encoding Of Data Bits

Check bit	Parity	Participating Data Bits
CX	even	0 4 6 7 8 9 11 14 17 18 19 21 26 28 29 31
C0	even	0 1 2 4 6 8 10 12 16 17 18 20 22 24 26 28
C1	odd	0 3 4 7 9 10 13 15 16 19 20 23 25 26 29 31
C2	odd	0 1 5 6 7 11 12 13 16 17 21 22 23 27 28 29
C4	even	2 3 4 5 6 7 14 15 18 19 20 21 22 23 30 31
C8	even	8 9 10 11 12 14 15 24 25 26 27 28 29 30 31
C16	even	0 1 2 3 4 5 6 7 24 25 26 27 28 29 30 31

During memory reads, the ECC logic calculates check bits for the memory data and XORs them with the memory check bits to form the syndrome bits. These syndrome bits indicate the number of errors present in the memory data/check-bit word, and in the case of single-bit errors, the position of the error. Table 4-7 lists the decoding of the syndrome <S16,S8,S4,S2,S1,S0,SX> bits.

Table 4-7: Syndrome Bit Decoding

Syndrome	Error	Syndrome	Error	Syndrome	Error	Syndrome	Error
00	none	20	C8	40	C16	60	double
01	CX	21	double	41	double	61	multi
02	C0	22	double	42	double	62	D24
03	double	23	D8	43	multi	63	double
04	C1	24	double	44	double	64	D25
05	double	25	D9	45	multi	65	double
06	double	26	D10	46	multi	66	double
07	multi	27	double	47	double	67	D26
08	C2	28	double	48	double	68	D27
09	double	29	D11	49	multi	69	double
0A	double	2A	D12	4A	D1	6A	double
0B	D17	2B	double	4B	double	6B	D28
0C	double	2C	D13	4C	multi	6C	double
0D	multi	2D	double	4D	double	6D	D29
0E	D16	2E	double	4E	double	6E	multi
0F	double	2F	multi	4F	D0	6F	double
10	C4	30	double	50	double	70	D30
11	double	31	D14	51	multi	71	double
12	double	32	multi	52	D2	72	double
13	D18	33	double	53	double	73	multi
14	double	34	D15	54	D3	74	double
15	D19	35	double	55	double	75	D31
16	D20	36	double	56	double	76	multi
17	double	37	multi	57	D4	77	double
18	double	38	multi	58	D5	78	double
19	D21	39	double	59	double	79	multi
1A	D22	3A	double	5A	double	7A	multi
1B	double	3B	multi	5B	D6	7B	double
1C	D23	3C	double	5C	double	7C	multi
1D	double	3D	multi	5D	D7	7D	double
1E	double	3E	multi	5E	multi	7E	double
1F	multi	3F	double	5F	double	7F	multi

The ECC logic supports several modes to facilitate diagnostic verification of the data paths in conjunction with a diagnostic latch embedded in the ECC logic. The diagnostic latch can supply an alternate set of check bits during write or read cycles. Refer to the CSR<LEDIAG> bit for details on loading the ECC diagnostic latch. Table 4-8 lists the modes available via the CSR<ECCMD,CORRECT> bits.

Table 4-8: ECC Logic Modes

ECCMD	CORRECT	Cycle Type	Check Bit Source	Function
0	N/A	Write	Generated from data	Normal generation
0	0	Read	Memory check bits	Normal detection
0	1	Read	Memory check bits	Normal correction
1	N/A	Write	Diagnostic latch	Diagnostic generation
1	0	Read	Memory check bits	Normal detection
1	1	Read	Memory check bits	Normal correction
2	N/A	Write	Generated from data	Normal generation
2	0	Read	Diagnostic latch	Diagnostic detection
2	1	Read	Diagnostic latch	Diagnostic correction
3	N/A	Write	Generated from zero	Initialize
3	0	Read	Memory check bits	Pass through

During the *pass through* diagnostic mode, the ECC error detection logic is disabled. Hence, no memory interrupts or bus exceptions will be generated when erroneous data is read. This mode may be used to retrieve erroneous data for error logging.

4.5. System Interface

The KN02 serial interface, real time clock, system control and status registers, and ROM are implemented in a system subsystem in the integral I/O option slot 7. Table 4-9 lists the address ranges of each subsystem in the system interface.

Table 4-9: System Interface Address Space

Address Range	Subsystem
0x1FC00000..0x1FC7FFFF	ROM device
0x1FC80000..0x1FCFFFFF	Reserved
0x1FD00000..0x1FD7FFFF	CHKSYN register
0x1FD80000..0x1FDFFFFF	ERRADR register
0x1FE00000..0x1FE7FFFF	DZ device
0x1FE80000..0x1FEFFFFF	RTC device
0x1FF00000..0x1FF7FFFF	CSR register
0x1FF80000..0x1FFFFFFF	ROM device

4.5.1. System ROM

A 256 KByte, socketted system ROM contains the powerup test, system initialization, and console software. ROM reads nominally complete in 13 cycles.

Since the ROM does not contain parity, ROM software should contain a checksum that is verified during system restart.

4.5.2. System Control and Status Register (CSR)

The CSR supports various operational and diagnostics modes of the KN02, the diagnostic LEDs, and allows observation of the I/O slot interrupt signals. The CSR control bits are automatically cleared during system powerup/reset. CSR reads nominally complete in 9 cycles. CSR writes nominally complete in 7 cycles. Figure 4-1 shows the format of CSR during reads (STATUS).

Name: STATUS Address: 1FF00000 Access: R

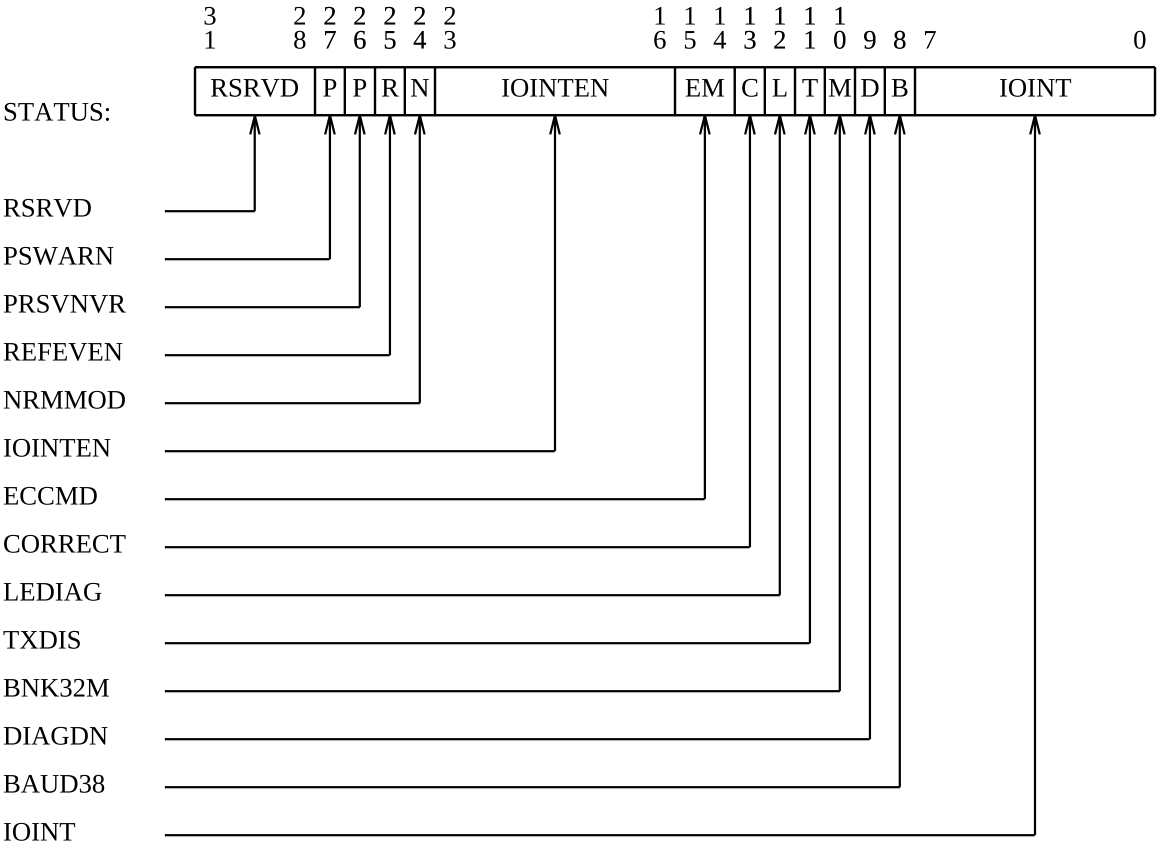

Figure 4-1: Status Register

IOINT
These bits reflect the state of the I/O slot interrupt signals. When an IOINT bit is one, the corresponding interrupt is active. When an IOINT bit is zero, the corresponding interrupt is inactive. The IOINT bits continually reflect the state of the I/O interrupt signals, regardless of the state of the IOINTEN bits or the interrupt enable and masks in the R3000. Operating system software should use these bits to resolve which slot(s) are generating an interrupt, to service an R3000 level 0 interrupt. Bit IOINT<n> corresponds to slot n, for example, IOINT<0> corresponds to slot 0, and IOINT<7> corresponds to slot 7.

BAUD38
This bit reflects the state of the BAUD38 bit in the control portion of CSR.

DIAGDN
This bit reflects the state of the DIAGDN bit in the control portion of CSR.

BNK32M
This bit reflects the state of the BNK32M bit in the control portion of CSR.

TXDIS
This bit reflects the state of the TXDIS bit in the control portion of CSR.

LEDIAG
This bit reflects the state of the LEDIAG bit in the control portion of CSR.

CORRECT
This bit reflects the state of the CORRECT bit in the control portion of CSR.

ECCMD
These bits reflect the state of the ECCMD bits in the control portion of CSR.

IOINTEN
These bits reflect the state of the IOINTEN bits in the control portion of CSR.

NRMMOD
This bit reflects the state of the manufacturing jumper on the module. When the jumper is absent, NRMMOD is one and the console should perform its normal powerup/reset tests and boot. When the jumper is installed, NRMMOD is zero and the console should execute manufacturing tests.

REFEVEN
This bit reflects which of the two possible memory banks will be refreshed during the next DRAM refresh cycle. When refresh is running, this bit will toggle every 195 cycles, or 7.8 microseconds. Diagnostic software should verify that this bit toggles.

PRSVNVR
This bit reflects the state of the security jumper on the module. When the jumper is absent, PRSVNVR (preserve NVR) is one and the console should not modify NVR. When the jumper is installed, PRSVNVR is zero and the console should re-initialize NVR and clear any NVR console passwords.

PSWARN
If the PSWARN bit is one, the power supply has detected an overtemperature condition. The operating system should monitor PSWARN during each clock interrupt and provide appropriate error logging and user notification if PSWARN changes.

RSRVD
These bits are reserved. They must be written with zeros for compatibility with future implementations.

Figure 4-2 shows the format of CSR during writes (CONTROL).

Name: CONTROL Address: 1FF00000 Access: W

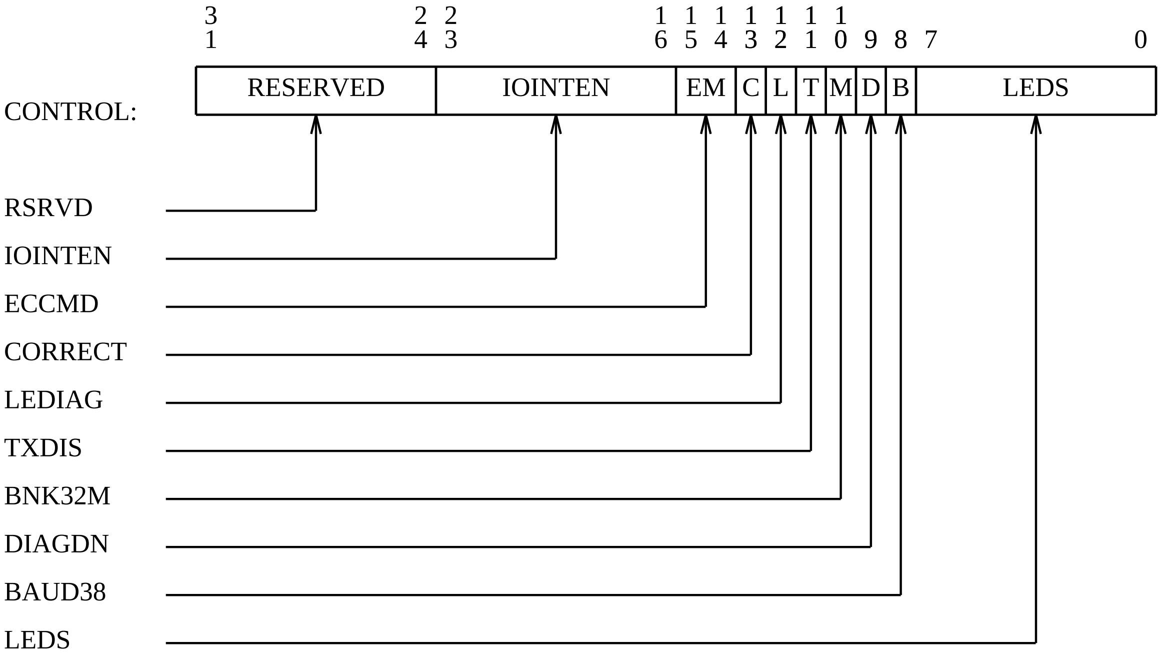

Figure 4-2: Control Register

LEDS
These bits determine the illumination of the diagnostic LEDs. When a bit is zero, the corresponding LED is illuminated. When a bit is one, the corresponding LED is off. Since the LEDS bits are automatically cleared at powerup/reset, all the LEDs are initially illuminated. Bit LEDS<n> corresponds to LED n, as labelled on the enclosure rear bulkhead, with LEDS<7> as the most significant bit and LEDS<0> as the least significant bit.

BAUD38
This bit controls the baud rate for the DZ baud select 1111. If BAUD38 is zero, the rate is 19.2 KBaud. If BAUD38 is one, the rate is 38.4 KBaud. Since the BAUD38 bit is automatically cleared at powerup/reset, the DZ baud select 1111 defaults to 19.2 KBaud. The BAUD38 affects all four serial channels of the DZ, hence, simultaneous 19.2 KBaud and 38.4 KBaud on different serial channels is not possible.

DIAGDN
This bit is connected to the manufacturing jumper. Manufacturing diagnostics should set DIAGDN to one to indicate successful completion of their tests.

BNK32M
This bit controls the memory bank stride. If BNK32M is zero, the stride is 8 MBytes. If BNK32M is one, the stride is 32 MBytes. Powerup/reset software should set the BNK32M bit, test each module to determine whether it is an 8 or 32 MByte array card, and if no 32 MByte modules are found, then clear the BNK32M bit. Since the BNK32M bit is automatically cleared at powerup/reset, the memory bank stride defaults to 8 MBytes.

TXDIS
This bit allows diagnostics to disable the EIA drivers on the serial lines during loopback testing. When TXDIS is zero, the EIA drivers are active. When TXDIS is one, the EIA drivers are disabled. Since the TXDIS bit is automatically cleared at powerup/reset, the EIA drivers are enabled by default.

LEDIAG
Pulsing this bit to one and then immediately writing it back to zero will latch the last memory write data value into the diagnostic latch in the memory system ECC logic. The ECC logic for the high bank (memory locations with odd word addresses) receives the last write data to the high bank. The ECC logic for the low bank (memory locations with even word addresses) receives the last write data to the low bank. The ECC diagnostic latch contains an alternate set of check bits for use during ECC diagnostic generate and detect modes. Table 4-10 lists the correspondence between write data bits and diagnostic check bits.

Table 4-10: ECC Diagnostic Latch Data/Check Correspondence

Data bit	Check bit
31..7	Write with zero
6	C16
5	C8
4	C4
3	C2
2	C1
1	C0
0	CX

CORRECT
This bit controls whether or not the ECC logic corrects single bit errors in memory read data. When CORRECT is zero, read data is passed through the ECC logic unmodified from memory. When CORRECT is one, read data with a single-bit error has the erroneous bit complemented as specified by the ECC syndrome. When CORRECT is one, and the ECC logic detects a multi-bit error, the output of the ECC logic is undefined. The state of the CORRECT bit does not affect memory interrupts, error logging, or bus errors; it only controls modification of erroneous data. Since the CORRECT bit is automatically cleared at powerup/reset, ECC correction is disabled by default.

ECCMD
These bits specify the operational mode of the ECC logic. Refer to the memory subsystem description for a description of the ECC logic modes. Since the ECCMD bits are automatically cleared at powerup/reset, the ECC logic defaults to normal generate/detect mode.

IOINTEN
These bits specify which I/O slot interrupt signals will be merged into the single R3000 level 0 interrupt. If IOINTEN<0> is a one, the slot 0 interrupt will be merged with the other slot interrupts. If IOINTEN<0> is a zero, the slot 0 interrupt will not be merged with the other slot interrupts. Bit IOINTEN<n> corresponds to slot n, where n represents a valid slot number. Operating system software should use the IOINTEN bits to individually mask the slot-specific interrupt signals. On the KN02 system module, IOINTEN<n> for slots 3 and 4 have no effect.

RSRVD
These bits are reserved and must be written with zero for compatibility with future implementations.

4.5.3. Real Time Clock (RTC)

The DS1287 real time clock chip provides a system clock, battery-backed-up time of year clock, and battery-backed-up 50 bytes of RAM for use as system startup, configuration parameters. The time-of-year clock continues operation, and the 50 bytes of RAM preserve their contents, via a battery, when the KN02 is un-powered. The system clock interrupt rate is programmable from 122 microseconds to 500 milliseconds.

A battery supplies power to the RTC and its time base oscillator while system power is off. When starting from a fully charged condition, the battery will maintain valid time and RAM data in the RTC for 10 years. The battery is automatically recharged while system power is on.

The RTC interrupts the CPU at level 1, which is visible in the R3000 CAUSE<11> register bit.

The RTC only supports byte reads and writes. The RTC registers are word-aligned. Reads of the RTC stall the CPU for 21 cycles. Writes to the RTC complete in 18 cycles. Table 4-11 lists the RTC register addresses.

Table 4-11: RTC Register Addresses

Address	Name	Register	Range
0x1FE80000	RTC_SEC	Seconds	0..59
0x1FE80004	RTC_ALMS	Seconds alarm	0..59
0x1FE80008	RTC_MIN	Minutes	0.59
0x1FE8000C	RTC_ALMM	Minutes alarm	0..59
0x1FE80010	RTC_HOUR	Hours	0..23
0x1FE80014	RTC_ALMH	Hours alarm	0..23
0x1FE80018	RTC_DOW	Day of week	1..7
0x1FE8001C	RTC_DAY	Date of month	1..31
0x1FE80020	RTC_MON	Month	1..12
0x1FE80024	RTC_YEAR	Year	0..99
0x1FE80028	RTC_REGA	Register A	
0x1FE8002C	RTC_REGB	Register B	
0x1FE80030	RTC_REGC	Register C	
0x1FE80034	RTC_REGD	Register D	
0x1FE80038	RTC_RAM	Base of BBU RAM	

4.5.4. RTC Registers

The RTC contains 64 8-bit registers. Ten of these contain the date and time data, four are control and status registers, and the remaining 50 provide general purpose RAM storage. The alarm functions of the RTC are not supported and should not be enabled by software.

4.5.4.1. Control Register A

```
    7     6     5     4     3     2     1     0
 +-----+-----+-----+-----+-----+-----+-----+-----+
 | UIP |       DVX       |          RSX          |
 +-----+-----+-----+-----+-----+-----+-----+-----+
```

RTC_REGA<7> Update In Progress (UIP)

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.

RTC_REGA<6:4> Timebase Divisor (DVX)

These read/write bits set the amount by which the time base oscillator input to the RTC is divided. These bits must be set to 010 to accommodate the 32.768 KHz time base in this system.

RTC_REGA<3:0> Rate Select (RSX)

These read/write bits select the rate at which the RTC generates periodic interrupts as shown in the following table. Software must also assert RTC_REGB<PIE> to enable the periodic interrupts.

RSX	Rate
0x0	none
0x1	3.90625 ms
0x2	7.8125 ms
0x3	122.070 us
0x4	244.141 us
0x5	488.281 us
0x6	976.562 us
0x7	1.953125 ms
0x8	3.90625 ms
0x9	7.8125 ms
0xA	15.625 ms
0xB	31.25 ms
0xC	62.5 ms
0xD	125 ms
0xE	250 ms
0xF	500 ms

4.5.4.2. Control Register B

```
   7     6     5     4     3     2     1     0
+-----+-----+-----+-----+-----+-----+-----+-----+
| SET | PIE | AIE | UIE | SQWE| DM  |24/12| DSE |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

RTC_REGB<7> Set Time (SET)

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 software can set new date and time values.

RTC_REGB<6> Periodic Interrupt Enable (PIE)

When this bit is asserted, periodic interrupts occur as the rate specified by RTC_REGA<RSX>.

RTC_REGB<5> Alarm Interrupt Enable (AIE)

This bit is not used and must be set to zero.

RTC_REGB<4> Update Interrupt Enable (UIE)

When this bit is asserted, an interrupt occurs when the RTC_REGA<UIP> bit is asserted.

RTC_REGB<3>   Square-wave Enable (SQWE)

This bit is not used and must be set to zero.

RTC_REGB<2>   Date Mode (DM)

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).

RTC_REGB<1>   Hours Format (24/12)

This read/write bit selects the format of the RTC_HOUR and RTC_ALMH registers. A value of one selects 24-hour mode; a value of zero selects 12-hour mode. In the latter case, bit 7 of the hours register is zero for AM and one for PM.

RTC_REGB<0>   Daylight Savings Enable (DSE)

This read/write bit is obsolete and must be set to zero. Software must explicitly compensate the time to account for local daylight savings conventions.

4.5.4.3. Control Register C

```
   7     6     5     4     3     2     1     0
+-----+-----+-----+-----+-----+-----+-----+-----+
|IRQF |  PF |  AF |  UF |           0           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

The RTC_REGC register clears itself, and any pending interrupts, when read. If software enables more than one RTC interrupt source, it must save a copy of this register and dispatch to handlers from the saved copy.

RTC_REGC<7>   Interrupt Request (IRQF)

When this read-only bit is set, it indicates that a RTC interrupt is pending.

RTC_REGC<6>   Periodic Interrupt Flag (PF)

This read-only bit indicates that a RTC periodic interrupt is pending.

RTC_REGC<5>   Alarm Interrupt Flag (AF)

This read-only bit indicates that a RTC alarm interrupt is pending.

RTC_REGC<4>   Update Interrupt Flag (UF)

This read-only bit indicates that a RTC update interrupt is pending.

RTC_REGC<3:0>   RAZ

Not used; read as zero.

4.5.4.4. Control Register D

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

RTC_REGD<7> Valid RAM/Time (VRT)

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 the system power is off and the backup battery voltage drops below the value required for the RTC to function properly. The bit is set to one after any read of this register (the register is read-only).

RTC_REGD<6:0> RAZ

Not used; read as zero.

4.5.4.5. Time of Year Registers

The time of year is kept in six registers: RTC_SEC, RTC_MIN, RTC_HOUR, RTC_DAY, RTC_MON, and RTC_YEAR. A seventh register, RTC_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 as selected by the RTC_REGB<DM> bit.

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. The RTC_REGA<UIP> bit 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 RTC_REGA until it finds bit UIP zero, at which time is has at least 244 microseconds 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.

4.5.4.6. Non-volatile RAM Storage

The 50 bytes of RAM storage are accessible at all times. They retain their value during power down provided the RTC_REGD<VRT> bit is asserted after power-up.

It is recommended that software implement a checksum on the non-volatile RAM contents as an additional safeguard against data corruption during battery-backed-up operation.

4.5.4.7. Initialization

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

1. Load register RTC_REGB with bit SET equal to one to inhibit time updates and bits PIE, UIE, DM, and 24/12 set as desired.

2. Load the seven time registers with the current date and time.

3. Load register RTC_REGA to set the proper time base divisor and the desired periodic interrupt rate.

4. Load register RTC_REGB 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 RTC are not affected by system power-on and power-off events.

4.5.5. Serial Interface (DZ)

The DC7085-based serial interface supports the keyboard and mouse in workstation configurations along with communications and printer ports for optional serial peripherals. In server configurations, the printer port connects to the console terminal, with the communications port available for an optional serial peripheral. The serial transmitters are double buffered, while the receivers share a 64-entry FIFO. The baud rate of each serial line is independently programmable to 50 through 9600 bits per second. In addition, baud rate select 1111 for each channel selects the external baud rate generator that may be set for 19200 or 38400 baud via the CSR<BAUD38> bit. Since the external baud rate generator is common to all serial channels, 19200 and 38400 baud cannot be in use simultaneously on different channels.

The keyboard and mouse ports are data-leads-only serial lines on a DB15 connector. The communications and printer serial ports implement TX, RX, DTR, DSR, RTS, CTS, CD, and RI control signals on DB25 connectors. All serial signals are EIA-232-D electrical levels. Table 4-12 lists the binding of devices to serial lines.

Table 4-12: Serial Device DZ Line Numbers

DZ Line	Serial Device
0	Keyboard
1	Mouse
2	Modem
3	Printer

The DZ interrupts the CPU at I/O slot 7, R3000 level 0.

The DC7085 supports half-word and byte reads and writes. The DC7085 registers are 8-byte-aligned in the processor address space. Reads to the DZ nominally stall the CPU for 15 cycles. Writes to the DZ nominally complete in 14 cycles. Table 4-13 lists the addresses of the DZ registers.

Table 4-13: DZ Register Addresses

Address	Access	Name	Function
0x1FE00000	R/W	DZ_CSR	Control and status
0x1FE00008	R	DZ_RBUF	Receiver buffer
0x1FE00008	W	DZ_LPR	Line parameters
0x1FE00010	R/W	DZ_TCR	Transmitter control
0x1FE00018	R	DZ_MSR	Modem status
0x1FE00018	W	DZ_TDR	Transmit data

4.5.6. Control And Status Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| TRDY | TIE  | ---- | ---- | ---- | ---- |TLINEB|TLINEA|
+------+------+------+------+------+------+------+------+
   RO     RW                                 RO     RO

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
|RDONE | RIE  | MSE  | CLR  |MAINT | ---- | ---- | ---- |
+------+------+------+------+------+------+------+------+
   RO     RW     RW     RW     RW
```

All bits in the DZ_CSR register are cleared by device reset or by asserting *master clear* (DZ_CSR<CLR>).

DZ_CSR<15> Transmitter Ready (TRDY)

This read-only bit is set when the transmitter scanner stops on a line whose transmit buffer may be loaded with another character and whose related DZ_TCR<LNENBX> bit is set. If the DZ_CSR<TIE> bit is also set, an interrupt request will be generated. When DZ_CSR<TRDY> is set, and at no other time, the *transmitter line number* (bits DZ_CSR<TLINEB:TLINEA>) is valid.

This bit is cleared when data is loaded into the transmitter for the line number indicated in DZ_CSR<TLINEB:TLINEA>. If additional transmit lines need service, DZ_CSR<TRDY> appears again within 1.4 microseconds of the completion of the transmitter data load operation. Since DZ_CSR<TRDY> requires 1.4 microseconds to update, software must not examine this bit during that period of time or it may erroneously interpret slow deassertion of the bit as a transmitter ready condition.

This bit is also cleared when *master scan enable* (DZ_CSR<MSE>) is cleared, or when the related DZ_TCR<LNENBX> bit is cleared.

DZ_CSR<14> Transmitter Interrupt Enable (TIE)

When this read/write bit is set, the setting of DZ_CSR<TRDY> will generate an interrupt request.

DZ_CSR<13:10> MBZ

DZ_CSR<09:08> Transmitter Line Number (TLINEB, TLINEA)

These read-only bits indicate the line number whose transmit buffer needs servicing. These bits are valid only when *transmitter ready* (DZ_CSR<TRDY)>) is set, and are cleared when *master scan enable* (DZ_CSR<MSE>) is cleared. Bit DZ_CSR<08> is the least significant bit.

DZ_CSR<07> Receiver Done (RDONE)

This is a read-only bit that is set when a character appears at the output of the silo. If the *receiver interrupt enable* (DZ_CSR<RIE>) is set, an interrupt request will be generated. If DZ_CSR<RIE> is clear, no interrupt request is generated, and the program may poll this bit to detect available characters.

This bit is cleared when the receiver buffer register (DZ_RBUF) is read. If another character is available in the silo, this bit will be cleared for a period between 100 nanoseconds and 1 microsecond, and will then be re-asserted.

This bit is also cleared when *master scan enable* (DZ_CSR<MSE>) is cleared.

DZ_CSR<06> Receiver Interrupt Enable (RIE)

This read/write bit permits the generation of an interrupt request when DZ_CSR<RDONE> is set.

DZ_CSR<05> Master Scan Enable (MSE)

This read/write bit must be set to permit the receiver and transmitter control sections to start the flag scanning process. When this bit is clear, *transmitter ready* (DZ_CSR<TRDY>) is inhibited from setting and the receiver silo is cleared.

DZ_CSR<04> Master Clear (CLR)

When written to a 1, this bit generates *initialize* within the chip. A read-back of this register with this bit set indicates that initialization is still in progress. This bit is self-clearing. All registers, silos, and UART functions are cleared with the following exceptions:

1. Only DZ_RBUF<DVAL> is cleared; the other bits are not
2. The modem control output bits are not cleared
3. The modem status register is not cleared

DZ_CSR<03> Maintenance (MAINT)

This is a read/write bit which, when set, loops the serial output connections of the transmitters to the corresponding serial input connections of the receivers. This feature is used only for maintenance.

While in maintenance mode, the transmitter outputs are still active. To avoid sending internal loopback data to external serial devices, the EIA drivers may be disabled by asserting the system control and status register TXDIS bit during the loopback test.

DZ_CSR<02:00> MBZ

4.5.7. Receiver Buffer Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| DVAL | OERR | FERR | PERR | ---- | ---- |RLINEB|RLINEA|
+------+------+------+------+------+------+------+------+
   RO     RO     RO     RO                   RO     RO

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
|RBUF07|RBUF06|RBUF05|RBUF04|RBUF03|RBUF02|RBUF01|RBUF00|
+------+------+------+------+------+------+------+------+
   RO     RO     RO     RO     RO     RO     RO     RO
```

The receiver buffer register (DZ_RBUF) is a 16-bit read-only register that contains the received character as the output of the 64-location silo buffer. A read of the register causes the character entry to be removed from the buffer, and all other entries shift down to the lowest location that is not occupied. Only the *data valid bit* (DZ_RBUF<15>) is cleared by *master clear* (DZ_CSR<CLR>) or device reset. The other bits have **unpredictable** values.

DZ_RBUF<15> Data Valid (DVAL)

This bit, when set, indicates that the data in bits DZ_RBUF<14:00> is valid. This permits an interrupt handling program to read the silo repeatedly and test each entry (after it has been moved out of the silo) until the program finds an entry for which this bit is zero, indicating that the silo is now empty.

DZ_RBUF<14> Overrun Error (OERR)

This bit becomes set when a received character is overwritten in the UART buffer by a following character before it has been transferred to the silo by the scanner. This condition indicates that the program is not removing characters from the silo sufficiently quickly, resulting in silo full conditions.

DZ_RBUF<13> Framing Error (FERR)

This bit is set if the received character did not have a stop bit present at the correct time. This bit is usually interpreted as indicating that a BREAK has been received.

DZ_RBUF<12> Parity Error (PERR)

This bit is set if the sense of the parity of the received character does not agree with the parity defined for that line.

DZ_RBUF<11:10> RAZ

DZ_RBUF<09:08> Received Line Number (RLINEB, RLINEA)

These bits contain the line number upon which the received character arrived. Bit DZ_RBUF<08> is the least significant.

DZ_RBUF<07:00> Received Character (DZ_RBUF7 - DZ_RBUF0)

These bits contain the received character. Characters of less than eight bits in length are right justified with unused bit positions shown as zeroes. The least significant bit is bit DZ_RBUF<00>. The parity bit is not shown.

4.5.8. Line Parameter Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| ---- | ---- | ---- |RXENAB| SC D | SC C | SC B | SC A |
+------+------+------+------+------+------+------+------+
                        WO     WO     WO     WO     WO

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
|ODDPAR|PARENB| STOP |CHAR B|CHAR A| ---- |LINE B|LINE A|
+------+------+------+------+------+------+------+------+
   WO     WO     WO     WO     WO            WO     WO
```

The line parameter register (DZ_LPR) controls the operating parameters related to each line in the chip. The DZ_LPR must be addressed with a word address and is a write-only register. The line parameters for each line must be loaded again after the setting of *master clear* (DZ_CSR<CLR>) or the assertion of the device reset pin. This register should not be modified while data transmission or reception is in progress on the associated line.

DZ_LPR<15:13> MBZ

DZ_LPR<12> Receiver Enable (RXENAB)

This bit must be set before the UART receiver logic for this line can assemble characters from the serial input line.

DZ_LPR<11:08> Speed Code (SC D, SC C, SC B, SC A)

The state of these bits determines the operating speed for the transmitter and receiver of the selected line. The system control and status register BAUD38 bit selects between 19200 and 38400 baud in the external baud rate generator common to all channels.

11	10	09	08	Speed
SC D	SC C	SC B	SC A	
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19200/38400

DZ_LPR<07> Odd Parity (ODDPAR)

If this bit is set (and DZ_LPR<PARENB> is set), characters of odd parity are generated on the line and incoming characters are expected to have odd parity. If this bit is not set (and DZ_LPR<PARENB> is set), characters of even parity are generated and incoming characters are expected to have even parity. If DZ_LPR<PARENB> is not set, the state of this bit is immaterial.

DZ_LPR<06> Parity Enable (PARENB)

If this bit is set, characters transmitted on the line have an appropriate parity bit added, and characters received on the line have their parity checked.

DZ_LPR<05> Stop Code (STOP)

This bit sets the stop code length; 0 = 1 unit stop, 1 = 2 unit stop (or 1.5 unit stop if 5-bit character length is selected).

DZ_LPR<04:03> Character Length (CHAR B, CHAR A)

The bits control the length of the characters generated by the transmitter and expected by the receiver according to the table below:

04	03	Character Length
CHAR B	CHAR A	
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

DZ_LPR<02> MBZ

DZ_LPR<01:00> Parameter Line Number (LINE B, LINE A)

These bits specify the line number for which the parameter information bits <12:03> is to apply. Bit DZ_LPR<00> is the least significant bit.

4.5.9. Transmit Control Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| ---- | ---- | ---- | ---- |RTS 2 |DTR 2 |RTS 3 |DTR 3 |
+------+------+------+------+------+------+------+------+
                                      RW

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
| ---- | ---- | ---- | ---- |LNENB3|LNENB2|LNENB1|LNENB0|
+------+------+------+------+------+------+------+------+
                               RW     RW     RW     RW
```

DZ_TCR<15:12> MBZ

DZ_TCR<11> Communications Request To Send (RTS2)

This read/write bit controls the assertion of the RTS signal for line 2. This bit is **not** cleared by the setting of *master clear* (DZ_CSR<CLR>). It is cleared by a device reset.

DZ_TCR<10> Communications Data Terminal Ready (DTR2)

This read/write bit controls the assertion of the DTR signal for line 2. This bit is **not** cleared by the setting of *master clear* (DZ_CSR<CLR>). It is cleared by a device reset.

DZ_TCR<09> Printer Request To Send (RTS3)

This read/write bit controls the assertion of the RTS signal for line 3. This bit is **not** cleared by the setting of *master clear* (DZ_CSR<CLR>). It is cleared by a device reset.

DZ_TCR<08> Printer Data Terminal Ready (DTR3)

This read/write bit controls the assertion of the DTR signal for line 3. This bit is **not** cleared by the setting of *master clear* (DZ_CSR<CLR>). It is cleared by a device reset.

DZ_TCR<07:04> MBZ

DZ_TCR<03:00> Transmitter Line Enable (LNENB3, LNENB2, LNENB1, LNENB0)

These read/write bits enable the transmitter logic for lines 3, 2, 1, and 0 respectively. Setting one of these bits causes the transmitter scanner to stop if the UART for that line has a transmitter buffer empty condition. An interrupt is then generated if transmitter interrupts are enabled. The scanner restarts when either the transmit data register (DZ_TDR) is loaded with a character or when the DZ_TCR<LNENBX> bit is cleared for the line upon which the scanner stopped. DZ_TCR<LNENBX> bits should only be cleared while the scanner is not running, i.e. when *transmitter ready* (DZ_CSR<TRDY>) is set, or *master scan enable* (DZ_CSR<MSE>) is clear.

These bits are cleared by setting *master clear* (DZ_CSR<CLR>) or by asserting the device reset pin.

4.5.10. Modem Status Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| ---- | ---- | ---- | ---- | RI 2 | CD 2 |DSR 2 |CTS 2 |
+------+------+------+------+------+------+------+------+
                                              RO

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
| ---- | ---- | ---- | ---- | RI 3 | CD 3 |DSR 3 |CTS 3 |
+------+------+------+------+------+------+------+------+
```

The modem status register (DZ_MSR) is a 16-bit read-only register. A read of this register gives the status of the communications and printer port control signals. The ON condition of a control signal is interpreted as a logical one.

DZ_MSR<15:12> RAZ

DZ_MSR<11> Communications Ring Indicate (RI 2)

This bit reflects the state of the *ring indicate* signal from line 2.

DZ_MSR<10> Communications Carrier Detect (CD 2)

This bit reflects the state of the *carrier detect* signal from line 2.

DZ_MSR<09> Communications Data Set Ready (DSR 2)

This bit reflects the state of the *data set ready* signal from line 2.

DZ_MSR<08> Communications Clear To Send (CTS 2)

This bit reflects the state of the *clear to send* signal from line 2.

DZ_MSR<07:04> RAZ

DZ_MSR<03> Printer Ring Indicate (RI 3)

This bit reflects the state of the *ring indicate* signal from line 3.

DZ_MSR<10> Printer Carrier Detect (CD 3)

This bit reflects the state of the *carrier detect* signal from line 3.

DZ_MSR<09> Printer Data Set Ready (DSR 3)

This bit reflects the state of the *data set ready* signal from line 3.

DZ_MSR<08> Printer Clear To Send (CTS 3)

This bit reflects the state of the *clear to send* signal from line 3.

4.5.11. Transmit Data Register

```
+------+------+------+------+------+------+------+------+
|  15  |  14  |  13  |  12  |  11  |  10  |  09  |  08  |
| ---- | ---- | ---- | ---- |BRK 3 |BRK 2 |BRK 1 |BRK 0 |
+------+------+------+------+------+------+------+------+
                               WO     WO     WO     WO

+------+------+------+------+------+------+------+------+
|  07  |  06  |  05  |  04  |  03  |  02  |  01  |  00  |
|TBUF07|TBUF06|TBUF05|TBUF04|TBUF03|TBUF02|TBUF01|TBUF00|
+------+------+------+------+------+------+------+------+
   WO     WO     WO     WO     WO     WO     WO     WO
```

DZ_TDR<15:12> MBZ

DZ_TDR<11:08> Break Control (BRK 3, BRK 2, BRK 1, BRK 0)

These bits control the assertion of BREAK on lines 3, 2, 1, and 0 respectively. Setting a break bit IMMEDIATELY forces the output of that line to space.

DZ_TDR<07:00> Transmitter Buffer

Characters for transmission are loaded into these bits. DZ_TDR<00> is the least significant bit. Loading of a character should occur only when *transmitter ready* (DZ_CSR<15>) is set. The character that is loaded into this register is routed to the line defined in DZ_CSR<TLINEB:TLINEA>.

This register is cleared by setting *master clear* (DZ_CSR<CLR>) or by asserting the device reset pin. This register can be used regardless of the state of the *maintenance* bit (DZ_CSR<MAINT>).

4.5.12. Error Address Status Register (ERRADR)

The ERRADR status register indicates the address, CPU or DMA bus master, read or write transaction, timeout/DMA overrun or ECC error, and a valid bit, from the transaction that generated the memory interrupt. ERRADR preserves the first error status in the event of multiple errors. Issuing a write to ERRADR clears the register, and the memory interrupt. ERRADR is also automatically cleared during system powerup/reset. ERRADR reads or writes nominally complete in 9 cycles. Figure 4-3 shows the format of ERRADR during reads. All bits are undefined unless the VALID bit is one.

Name: ERRADR Address: 1FD80000 Access: W

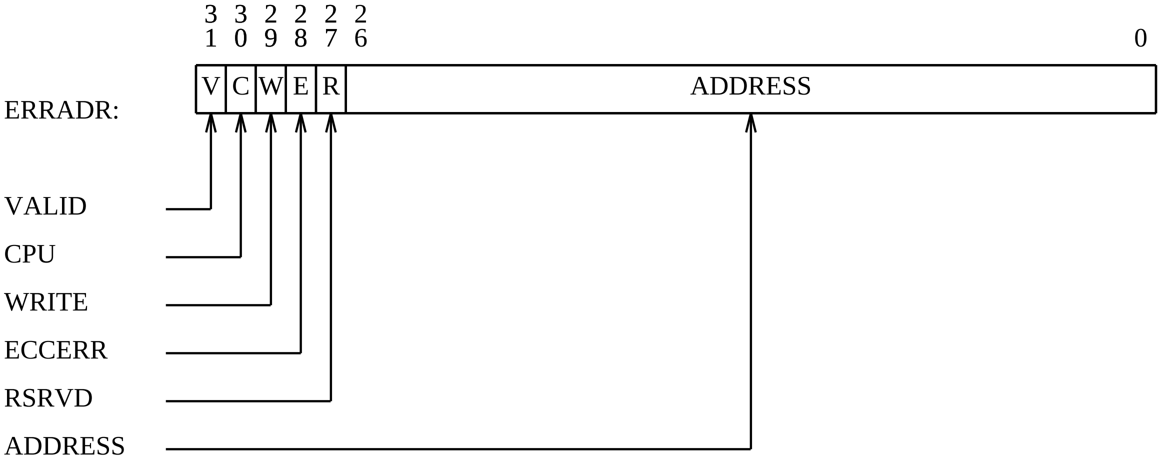

Figure 4-3: Error Address Register

ADDRESS The ADDRESS field records the value of the pipelined address in effect at the time the error occurred. For I/O transactions and partial memory writes, this is the word address issued by the CPU. For DMA overrun errors, this is the word address of the last valid word transferred (127). For CPU and DMA memory reads, this is the word address in the memory controller. However, due to pipelining of the memory controller, the column field of the word address has advanced five stages before the ECC error status is available. Software must extract ADDRESS<11:0>, perform a signed subtract of five, and then reinsert this value into ADDRESS<11:0> to recover the address of the word which contained the ECC error.

RSRVD This bit is reserved.

ECCERR If the ECCERR bit is one, an ECC error occurred. If the ECCERR bit is zero, an I/O timeout or DMA overrun occurred.

WRITE If the WRITE bit is one, the error occurred on an I/O write or memory write transaction. If the WRITE bit is zero, the error occurred on an I/O read or memory read transaction.

CPU If the CPU bit is one, the error occurred during a CPU transaction. If the CPU bit is zero, the error occurred during a TURBOchannel DMA transaction.

VALID This bit is set to one when error information is clocked into the register. When VALID is already set, no further status information is preserved. Table 4-14 shows the value of the ECCERR, WRITE, and CPU ERRADR bits for the different types of system errors. During read conflicts, the memory controller may service the same read request several times (while stalling the CPU) until conflicting write data in the write buffer has been flushed. Note that it is possible for ECC read errors to occur during CPU read conflicts when the CPU is stalled. However, after the write buffer is flushed, the error is overwritten with new data, so the processor will not receive a bus error on termination of the read. Also note that if the CPU is waiting for a memory-space partial write to complete, and a multi-bit ECC error occurs during the read/modify/write of the partial data, invalid data and valid ECC check bits will be loaded into memory. In this case, the ensuing read will complete without causing an exception even though the read data is invalid. If the address is a cached location, invalid data will be loaded into the cache and the cache entry will be incorrectly marked valid. Regardless of the type of masked error, a memory interrupt will be generated, and the offending ECC read error or CPU partial write error will be correctly logged in the ERRADR and ERRSYN registers.

Table 4-14: System Error Type Encoding

CPU	WRITE	ECCERR	Error Type Description
0	0	0	DMA read overrun
0	0	1	DMA memory read ECC
0	1	0	DMA write overrun
0	1	1	invalid combination
1	0	0	CPU I/O read timeout
1	0	1	CPU memory read ECC
1	1	0	CPU I/O write timeout
1	1	1	CPU partial memory write ECC

4.5.13. ECC Check/Syndrome Status Register (CHKSYN)

The CHKSYN status registers records the check bits from last memory read in the abcense of memory errors. When a memory error occurs, the CHKSYN check bit fields are frozen, and it records the syndrome from the memory bank on which the error occurred. The CHKSYN register is frozen until the ERRADR register is written. Issuing a write to CHKSYN clears the register. CHKSYN is also automatically cleared during system powerup/reset. CHKSYN reads or writes nominally complete in 9 cycles. Figure 4-4 shows the format of CHKSYN during reads. The syndrome bytes are only valid if the ERRADR<VALID> bit is set. The check bytes are only valid if their VALID bit is set.

Name: CHKSYN Address: 1FD00000 Access: W

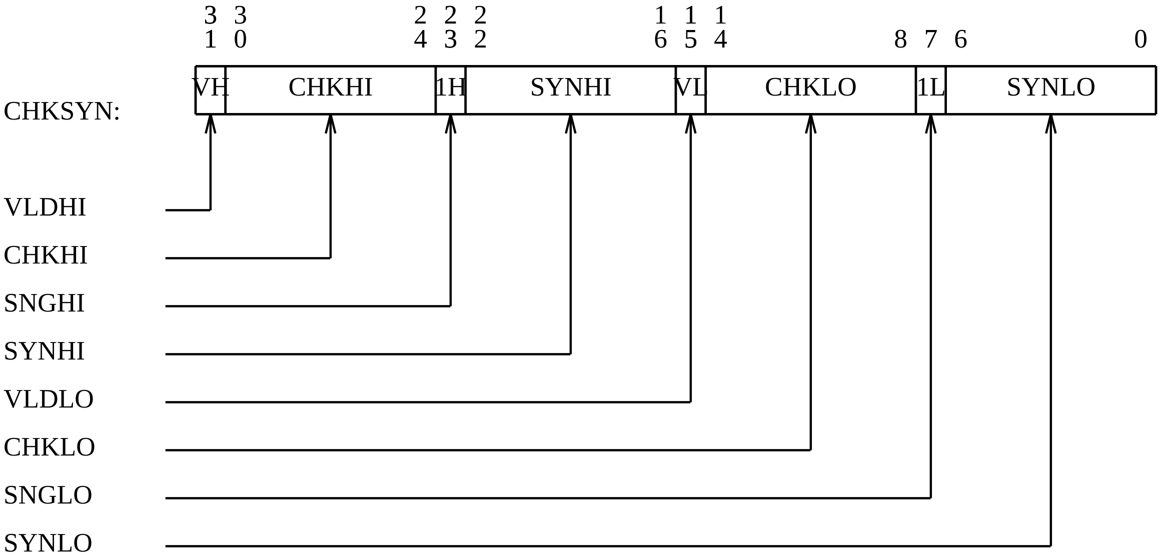

Figure 4-4: Check-bit/Syndrome Register

SYNLO This field records the syndrome bits calculated by the ECC logic at the time that an error was detected by the low bank of memory (even words). The ERRADR<ADDRESS> field must be used to determine whether the error pertains to a low or high word of memory. This field is undefined for high bank errors. The syndrome may be used to determine which bit was in error. Refer to the memory subsystem for a description of the syndrome logic.

SNGLO This bit records the single- versus multi-bit error output of the ECC logic at the time that an error was detected by the low bank of memory. If it is one, a single-bit error occurred. If it is zero, a multi-bit error occurred. This bit is valid when the SYNLO field is valid.

CHKLO In the abcense of errors, this field records the last check bits read from the low bank of memory (even word). Once an error occurs, and the ERRADR<VALID> bit is set, the CHKLO field is frozen.

VLDLO This bit is set to one when ever the CHKLO field is updated.

SYNHI This field records the syndrome bits calculated by the ECC logic at the time that an error was detected by the high bank of memory (odd words). The ERRADR<ADDRESS> field must be used to determine whether the error pertains to a low or high word of memory. This field is undefined for low bank errors. The syndrome may be used to determine which bit was in error. Refer to the memory subsystem for a description of the syndrome logic.

SNGHI This bit records the single- versus multi-bit error output of the ECC logic at the time that an error was detected by the high bank of memory. If it is one, a single-bit error occurred. If it is zero, a multi-bit error occurred. This bit is valid when the SYNHI field is valid.

CHKHI In the abcense of errors, this field records the last check bits read from the high bank of memory (odd word). Once an error occurs, and the ERRADR<VALID> bit is set, the CHKHI field is frozen.

VLDHI This bit is set to one when ever the CHKHI field is updated.

4.6. Ethernet Interface

The Ethernet interface is an integral implementation of the TURBOchannel Ethernet option, along with an integral ThinWire Ethernet transceiver, in I/O option slot 6. Refer to the *PMAD-AA TURBOchannel Ethernet Module Specification* for further details.

4.7. SCSI Interface

The SCSI interface is an integral implementation of the TURBOchannel SCSI option in I/O option slot 5. Refer to the *PMAZ-AA SCSI Module Specification* for further details.

4.8. Reserved I/O Slots

I/O option slots 4 and 3 are reserved for future implementations. I/O read and write transactions to these slots will generate I/O bus timeout errors.

4.9. TURBOchannel Option Slots

I/O option slots 2, 1, and 0 are available for TURBOchannel option modules to configure the DS5000/200 system for a range of workloads and environments. I/O read and write transactions to unpopulated slots will generate I/O bus timeout errors.

4.10. Reset Switch

A momentary push button switch activates the KN02 reset logic as per a powerup condition. The reset condition exists until the push button is released. All eight diagnostic LEDS are illuminated during reset.