XCRM - CRM Extended commands (Tape Version)

Mike Gleeson David Laird Robert Rossetto

Advanced Production Engineering Valid Logic Systems, Inc. DRAFT of May 29, 1986

1. Introduction

This document contains a description of the commands available on the extended version of CRM (ie. the Tape version). For a more complete description of the CPU diagnostics available see Valid 68010 and 68020 CPU board ROM based Diagnostics.

This document applies to XCRM tapes dated March 1986 or later.

2. System Requirements

The minimum system configuration to use XCRM is:

- CPU (68010 or 68020)
- ECM (must be ECM zero)
- Rimfire or Storager tape controller board
- Tape drive.

Note: Although XCRM will run with as little as 1/2 Mbytes of ECM, some commands are not operational without 2 Mbytes minimum (ECM0 - ECM3). For example, XBKUP and XRSTR use ECMs 2 and 3 as a data buffer. It is recommended, therefore, that your system contain at least 2 Mbytes of ECM.

3. How to Load the XCRM Tape

Power on the system, and wait for the monitor prompt. Put the tape in the tape drive and put it on-line (XREW rewinds the tape). If you are not using the default tape drive, use the STUN command to set the default to the required drive (*i.e. one of RTxy or ITxy where x is the* controller number and y is the unit number). To load the tape, type the following commands printed in *italics* (The system responses are printed in bold type):

If system firmware supports tape boot:

>XBOOT<cr>

Device (Primary, Alternate, Tape, None) = TStartup (Auto, Parms, Versabug, None) = P

The tape will load and rewind. XCRM will start up and display the following prompt when ready to receive commands:

XCRM>

If system firmware does not support tape boot:

>XTR.

Start bus addr = 14000Limit bus addr = 100000Buffer size (0 for default) = 0 < CRM >Tape status = (RFE RFC) (OEA) ()

When the tape is loaded, rewind the tape by typing:

XREW.

Start XCRM by typing:

>20000LJ<cr>

LCTRL7C

XCRM will start up and display the following prompt when ready to receive commands:

XCRM>

Note that the XCRM tape works for both CPU I and CPU II.

4. General

XCRM continuously listens to the console port for commands. It executes whatever it sees and understands. XCRM echos everything it receives from the console port except most control characters. The host port is currently not used by XCRM.

Upper-case and lower-case characters are equivalent in XCRM commands. XCRM has a rotating accumulator 4 bytes long. Any valid hex digit is pushed into the accumulator on the right, causing one hex digit to fall out on the left. For example, typing

123456789L

will load the value 0x23456789 into the accumulator. XCRM also has an address register, loaded from the accumulator by command. Various commands use the address register as a parameter. XCRM normally intercepts traps and prints the trap number and other critical information.

In this document 0x preceding a number signifies hexadecimal numbers, a "byte" is an 8 bit quantity, a "word" or a "short" is a 16 bit quantity, and a "long word" or a "long" is a 32 bit quantity.

5. Switch settings

The 8 switches on the CPU board have meaning to XCRM at reset time. A switch is considered on when it is positioned away from the board.

	Switch settings (on, off, ! => user's choice, sys => system dependent)						
Switch number		Meaning when switch is turned on					
8	(?)	Autoboot UNIX					
7	(off)	Enter Dynamic burnin-mode on power_up/reset					
6	(?)	Run extended diagnostics on power_up/reset					
5	(on)	CPU has 16Meg map (should always be on)					
4	(on)	Enable low Multibus memory access (off $=>$ Disable access)					
3	(sys)	Default DISK drive on Interphase controller (off => Rimfire controller)					
2	(sys)	Default TAPE drive on Interphase controller (off => Rimfire controller)					
1	(off)	UNUSED					

Note: XCRM will take the CPU board out of boot state at reset time only if low Multibus memory exists and switch 4 is set accordingly.

6. XCRM commands

The XCRM command parser accepts the following syntaxes:

- 1. {IMMEDIATE}
- 2. [count] {IMMED IA TE}
- 3. {NONIMMEDIATE}
- 4. [count] {NONIMMEDIATE}

All commands may start with X. The X is optional unless necessary to resolve a conflict with an IMMEDIATE command. For example, ALLT will be interpreted as AL L T (load the accumulator with 0xA, load it again with 0xA then display address 0xA) so the X is mandatory in that case.

Commands that could be interpreted as a count preceding a non-immediate command are resolved in favor of the non-count command. (e.g., BOOT is correctly interpreted as BOOT instead of (0xB)OOT)

Backspace deletes the last character typed, and "U deletes the entire line. Linefeed, Return, and '.' may all be used to terminate a command. "C and "X have the same function, namely to break out of the current command and return to the menu.

If XCRM does not understand the extended-command name, it will execute nothing and suggest that you Type ? for help !.

6.1. Baud Rate Selection

XCRM assumes a baud rate of 9600 baud at power up. The baud rate may be changed to match most terminals by hitting the *break* key repeatedly until the correct baud rate is achieved. (The *Esprit II* terminal requires *shift-break*.) Each time the break key is hit the next rate is selected from the table below:

Ι	Baud Rate Selection
	9600
	19200
	50
	75
	110
	134.5
	150
	300
	600
	1200
L	1800
	2000
L	2400
L	3600
L	4800
L	7200

6.2. Single-keystroke (IMMEDIATE) Commands

- # Restart CRM, (return to the boot rom) reinitializing any devices and ECMs.
- L Load the address register. The longword in the rotating accumulator is loaded into the address register, then the rotating accumulator is set to zero.

- 5 -

- Store a byte, short or long depending on current mode. The byte/short/long in the rotating accumulator is stored at the logical address which is currently in the address register.
- N Display contents of next address. The address register is incremented, and then the byte/short/long pointed to by the address register is read and displayed.
- **P** Display contents of previous address. The address register is decremented, and then the byte/short/long pointed to by the address register is read and displayed.
- T Display contents of this address. The byte/short/long pointed to by the address register is read and displayed.
- M Display a 64 byte block. A 64-byte block of memory is displayed starting from the address in the address register modulo 64.
- J Jump to the address pointed to by the address register.
- V Display XCRM version (number and date). Also shows the default disk and tape drives and the most recent exception along with corresponding fault PC and SP.
- 6.3. Basic (NONIMMEDIATE) Commands
- ? Print menu of basic extended commands.
- ! Echo input to console terminal.

XBOOT Boot unix. You will be asked for a boot device.

- **P** Primary stands for the primary boot block on your disk, and is the one used most often.
- A Alternate selects an alternate boot block. It is intended for booting if the primary area is corrupt.
- **T** Tape stands for the primary tape device to boot from tape.

The next prompt will ask for the startup mode.

- A Auto means it will boot all the way to multi-user (asking a few questions along the way.)
- **P** Parms means it will ask you which unix to boot and bring that unix up single user.
- **XMODE** Set mode (*ie. number of bytes per R/W access*) to one of byte/short/long. Commands affected are the immediate commands N, P, T and and the non-immediate command XLLA. Note that XMEMT does not use mode to determine unit of access. It always performs longword accesses.
- XLSPE Load scratch-page entry.

XCOREDUMP Make a 16M crash tape on the default tape drive.

- **?TST** Print diagnostic commands.
- **?ECM** Print ECM commands.
- **?GEN** Print generic disk/tape commands.
- **?DK** Print disk commands.
- **?TP** Print tape commands.

- **6.3.1.** XCRM only nonimmediate commands
- XINIT Assume default initialization.
- XIMAP Initialize the page and segment maps.

XIVEC Initialize the trap (exception) vectors.

XCMP Compare two chunks of memory.

XCOPY Copy a chunk of memory.

XDUMP Dump (display) exception registers and stack.

XBDL Begin download from host port.

XEDL End download from host port.

6.4. DIAGNOSTIC Commands

Whenever a diagnostic is executed, XCRM will attempt to light the red LED if it fails or the green LED if it passes. A pass/fail count of the form nnnnP nnnnF is displayed after each pass or fail. ALL diagnostic commands (except XLLA) can be preceded by a repetition count. If no repetion count is specified, the command will be executed indefinitely. All diagnostics (except XLR) can be prematurely terminated by typing °C or °X.

- XSR Status register test. All bits in the status register are checked. The test is nondestructive.
- XCR Context register test. All bits in the context register are checked. The test is nondestructive.
- XSM Segment map test. All locations in the segment map are tested. A big map (16k) is assumed. After testing, the segment map is reinitialized to the identity map.
- XPM Page map test. All locations in the page map are tested. A big map (16k) is assumed. After testing, the page map is reinitialized to the identity map.
- XCTC Counter timer chip test. An internal location in the timer chip is tested.
- XUARTS Console and host uart test. For each uart, an internal location is tested.
- XSW Switch test. The switches are not allowed to be all zeroes or all ones. This is checked.
- XALLT Execute all diagnostics. The following diagnostics are executed: (SR, CR, SM, PM, SW, CTC, UARTS) The sequence of tests is repeated a number of times taken from the rotating accumulator. The XALLT command is automatically executed at reset time only if the diagnostics switch is turned on.
- XLLA Loop on logical address (write or read). XLLA reads or writes byte/short/long data depending on the current MODE (set by XMODE). XLLA prompts for:

R/W? address? data? (only if Write) silent?

If silent is specified, XLLA loops forever on the address and the only way to abort is by doing a hardware reset. Before each read or write access, the CPU Red LED is turned on and off once (for scope trigger purposes). On data reads, if silent is not specified, the address and data read are displayed after each access. On data writes, if silent is not specified, the address and data read are displayed once. To abort, type °C or °X.

6.4.1. XCRM only diagnostic commands

XBDTST This command switches to the Board test diagnostic menu. Selecting the appropriate menu item will allow the operator to perform extended tests of Valid hardware. CCB, CGB, CPU2, PIB, VG and ZOID (Realdraw) are currently supported. For more complete descriptions of the board test diagnostics, see section 8. of this document.

6.5. ECM Commands

- **XDES** Display status of every Error Correcting Memory board (ECM).
- XIECM Initialize every ECM board.
- XMEMT Execute memory-test diagnostic. This command will prompt the user for start and end address. All memory accesses are long word accesses (ie. not affected by the XMODE command).

6.5.1. XCRM only ecm commands

XSMMT Test memory mapping between boards.

6.6. GENERIC Disk/Tape Commands

XCINIT Initialize the controllers.

- XBKUP Backup the current disk. XCRM will ask you for the start cylinder (default = 0), start head(0), start sector(1) and end cylinder(default = 0). To backup a 70 Mby drive on quarter inch tape go from 0 to 202 (hex) on one tape then repeat with 203 to 3F0 (hex). For half inch use 0 to 3f0. A 2400 ft. reel of tape will hold about 40 Mby. If you have more than one disk and you want to XBKUP a disk other than disk 0, you will have to change to default disk unit number with the XSDUN. (set disk unit number) XCRM command. Note also that there is a XDDUN. (display disk unit number) XCRM command.
- XRSTR Restore the current disk. This command prompts for start and end cylinders like the XBKUP command above. The same cylinder numbers as used in XBKUP should be used.

6.6.1. XCRM only generic Disk/Tape commands

XOCG Open the controller gates.

XIS Issue an interphase command.

XISRDCSR Read csr.

XISPBADDR Set parameter block address.

- XISSETPB Review/set parameter block values.
- XISGO Write csr.

XISGOBIO Write csr, poll and log csr into memory from 0x100000 to 0x200000.

6.7. Generic DISK Commands

- **XDDUN** Display the current unit number.
- XSDUN Set disk unit number. XCRM will prompt for the number you want as the new default disk unit number. XCRM initially defaults to RD00. The disk unit number is broken into four fields as follows:

(i)	Controller type.	(R => Rimfire, I => Interphase)
(ìi)	Device type.	(D => Disk)
(iii)	Controller number.	(System dependent)
(iv)	Device number.	(System dependent)

XDSTA Disk status.

XDR Disk read command.

XDW Disk write command.

XDRST Disk reset command.

8.7.1. XCRM only generic disk commands

XDDN Spin the current disk down. (This command applies only to Rimfire disk controllers).

XDDPB Print the disk parameter block.

XDKBN Burn in up to 8 disks serially using **DKTEST** (see below).

Ls and Gs terminate the program string since DKBN loops forever anyway.

XDKTEST Disk test program. The following tests can be conducted over a selected range of disk cylinders:

С	Clear	clear all flags
G	Grind	read or write one group of sectors forever
L	Loop	loop on previous commands forever
Μ	Map	set MAP ON ERROR flag
0	Oscillate	OSCILLATING seek test. Starting with
		head 0: Read the Nth track from the end
		then read the Nth track from the
		beginning then increment N and repeat.
		Increment head after all tracks in the
		selected range are read.
R	Random	RANDOM seek test. Read to 1000 random tracks.
S	Scan	SCAN test. Read consecutive tracks.
w	Write	set WRITE flag. Tests do read-write-read
		instead of just read.

A series of tests can be specified in the program string. For example MSSRWROL instructs *DKTEST* to set the MAP flag to automatically map out any bad tracks encountered during testing, run the *Scan* test twice (read only), run the *Random* test (read only), set the WRITE flag, run the *Random* test (read-write-read), run the *Oscillate* test (read-write-read), then loop back and continue.

DKTEST normally runs silently, only printing errors when they occur and pass/fail information at the end of each loop. All flags are cleared at the end of each loop. Current test status information can be turned on by typing Q and turned off by typing S while DKTEST is running.

- **XDLMT** List the mapped tracks.
- XDNOP Disk no-op command.
- XDUP Spin the current disk up. (This command applies only to Rimfire disk controllers).

XFMT Format the disk. (See XNEWDISK for more information on formatting with Interphase controllers.)

XMAPD Map out disk track. (See XNEWDISK for more information on mapping tracks with Interphase controllers.)

XNEWDISK Format and map tracks.

For *Rimfire* controllers: this command is equivilent to typing XFMT then XMAPD.

For Interphase controllers: If the controller is not configured for the current disk, a label must be created before formatting and bad track mapping can be done.

If the label exists but the disk will be reformatted, the label must be changed to set the **a_alloc** field to 0. (A_alloc tells XMAPD and UNIX how many alternate tracks to skip before searching for the next available track.) The controller will complain if it does not like the label parameters specified. If this happens check your work.

The label should be written onto the disk to save any changes made. This can be done before the format program is run because the label writing routine always formats track 0 before writing the label. The controller will complain if the disk can not be formatted according to the label specifications. If this happens check your work, then check the hardware.

The entire disk (including alternate area) can then be formatted. Whenever track 0 is formatted the label is automatically rewritten. Remember that *a_alloc* field should be set to 0 if the alternate area is reformatted (this is the normal case). Whenever formatting, we strongly recommend that you set FORMAT VERIFY mode. FORMAT VERIFY can be turned on by typing Q and turned off by typing S while FMT is running.

After formatting the disk, bad tracks can be mapped. You will be prompted for cylinder and head of each bad track. User area bad tracks are mapped to alternate area tracks. Alternate area tracks are misformatted to prevent future use. When the last bad track is entered, hit $\langle CR \rangle$ to return to XCRM.

6.8. Generic TAPE Commands

XSTUN Set the tape unit number. XCRM will prompt for the number you want as the new default tape unit number. XCRM initially defaults to RT00 (or IT00 if switch 2 is on). The tape unit number is broken into four fields as follows:

(i)	Controller type.	(R => Rimfire, I => Interphase)
(ii)	Device type.	(T => Tape)
(iii)	Controller number.	(System dependent)
(iv)	Device number.	(System dependent)

Note: if you make a syntax error the drive selection will not change.

- XDTUN Display tape unit number.
- XTSTA Tape status command.
- XTR Tape read command. XCRM will prompt for a "start bus address". Enter a memory address (hex). It will then prompt for an "limit bus address", enter another memory address (hex). "default buffer size" will appear. 0 defaults to 4K bytes, anything else and the buffer size is set to that number (hex).
- XTW Tape write command. XCRM will prompt for a "start bus address". Enter a memory address (hex). It will then prompt for an "limit bus address", enter another memory address (hex). "default buffer size" will appear. 0 defaults to 4K bytes, anything else and the buffer size is set to that number (hex).

XTRST Reset the tape drive.

XREW Rewind the tape.

XTRET Retension the tape.

XWFM Write a file mark.

6.8.1. XCRM only generic tape commands

- XTERM Terminate an open-ended command.
- XTNOP Tape no-op command.
- XDTPB Print the tape parameter block.

7. Alphabetical listing of XCRM commands:

:

Command	Description
t	echo
•	restart CRM
#	store value
- ?	print menu
! DK	disk commands
I ECM	ECM commands
! GEN	generic disk/tape commands
TP	tape commands
TST	diagnostic commands
J	restore and jump
J L	load address register from accumulator
л М	print a 64 byte block
N	print next address
P	print previous address
T	print this address
XALLT	execute all diagnostics
XBDL	begin download from host port
XBDTST	board test programs
XBKUP	backup the current disk
XBOOT	boot UNIX
XCINIT	initialize the controllers
XCMP	compare two chunks of memory
XCOPY	copy a chunk of memory
XCOREDUMP	create a core dump tape
XCR	context reg test
XCIC	timer test
XDDN	spin the current disk down print the disk parameter block
XDDPB	display disk unit number
XDDUN	display status of every ECM board
XDES XDKBN	burn in multiple disks
XDKTEST	flag unmapped bad tracks
-	· · · ·
XDNOP	disk no-op command disk read
XDR	disk reset
XDRST	
XDSTA	disk status
XDTPB	print the tape parameter block
XDTUN	display tape unit number
XDUMP	dump exception registers and stack
XDUP	spin the current disk up disk write
XDW	test every ECM board using signature analysis
XECMT	end download from host port
XEDL	format disk
XFMT	init every ECM board
XIECM	•
XIMAP	init the map assume default initialization
XINIT	
XIS	issue an interphase command
XISGO	write csr
XISGOBIO	write and log Interphase csr
XISPBADDR	set Interphase parameter block address
XISRDCSR	read csr

--

	- 12 -	
XISSETPB	review/set Interphase parameter block values	
XIVEC	init the trap vectors	
XLLA	loop on logical address	
XLR	local-RAM test (destructive)	
XLSPE	load scratch page entry	
XMAPD	map out disk track	
XMEMT	execute memory-test diagnostic	
XMODE	set mode (BYTE, SHORT, LONG)	
XNEWDISK	format and map tracks	
XOCG	open the controller gates	
XPM	page map test	
XREW	rewind the tape	
XRSTR	restore the current disk	
XSDUN	set disk unit number	
XSM	segment map test	
XSMMT	test memory mapping between boards	
XSR	status reg test	
XSTUN	set tape unit number	
XSW	switch test	
XTERM	terminate an open-ended command	
XTNOP	tape no-op command	
XIR	tape read command	
XTRET	retension the tape	
XTRST	reset the tape drive	
XTSTA	tape status	
XTW	tape write command	
XUARTS	console/host uart test	
XWFM	write a file mark	
V	print CRM version	

8. VALID Board Test Diagnostic Guide

To run the board test diagnostics, load the XCRM tape (see section 3. How to Load the XCRM Tape), and type the following commands printed in *italics*. The system responses are printed in **bold** type:

XCRM>XBDTST<cr>

BDTST> ?

XCCB	color controller
XCGB	color graphics
XCPU2	cpu II
XPIB	peripheral interface
XVGB	video graphics
XZOID	zoid
XCRM	return to XCRM
?	print this message

BDTST>

8.1. XCCB Color Controller Board Diagnostics

. .

8.2. XCGB Color Graphics Board (Peacock) Diagnostics

8.3. XCPU2 CPU II (68020 CPU) Diagnostics

8.3.1. INTRODUCTION

This section describes how to run the diagnostic program written for the CPU 20 board set.

8.3.2. DESCRIPTION

The 68020 CPU boards are very similar to the old 68010 CPU boards except for the addition of a cache. The cache is a fast memory that is between the 68020 and multibus. This memory is 16K bytes deep or 4K long words. We will refer to these 4K long words as lines.

Each line in the cache can map to many long words in memory. The way we keep track of which byte in memory is in the cache is by using the upper 10 bits of address and the valid bits as a marker. The address bits keep track of which long word of memory is mapped to a line in cache. The address bits are stored in a tag memory array. The valid bits keep track of which byte or bytes in a line are good. The valid bits are stored in a valid bit memory array. There four valid bits and a tag location for each line in cache.

When the 68020 writes to memory both the cache location and the multibus memory will be modified. This will modify both the tag and a number of valid bits depending on the length of the data. When the 68020 reads from memory it will first check the cache to see if the data is there. This is done by comparing the upper 10 bits of the read address with the tag of the cache line that maps to that address. Then the valid bits are compared with the byte enables from the 68020. If both the tag and the valid bits match then the data is in cache and the 68020 will continue(this is called a read hit). If no match occures(this is called a read miss) then the board will halt the 68020, empty the write queue, and start a multibus read. When the multibus read completes the data will be in the cache and the 68020 can complete the memory read.

The write queue is a 16 line first in first out buffer that is used to store a memory write from the 68020 until it can be written to multibus memory.

There is another piece of logic on the CPU20 board called the multibus listener. This logic is used to look at multibus addresses. If another multibus card writes to memory and that data is also in the cpu cache then the cache line must be made invalid. The listener logic does this by altering the valid bits.

In order to make the cpu board more easily tested the following features have been added:

- 1. Cache can be enable or disabled.
- 2. The cpu can be made to invalidate in the cache what it wrote(suicide mode).
- 3. Bus errors can be disabled.
- 4. Multibus access can be disabled.
- 5. The last data read or written from the valid and tag memories can be latched.

After the program is started, the diagnostic will inform the operator of the revision level of the code followed by the prompt. The following is a list of the possible commands.

XBMAP	Big seg map test
XCD	Cache data test
XCM	Cache misc test
XEREG	Test error register

XINTST	Level 6 and 3 interrupt test
XLI	Listener test
XALLT	Run all diagnostics
XRANDT	Random test
XRM	Rmcmp tag test
XIFP	68881 chip test
XIV	Tag valid ram test
XWQA	Write queue address test
XWQD	Write queue data test
XCU	This test uses the Cache
XBCE	CPU20 board cache enable
XBCD	CPU20 board cache disable
XHALT	Set halt mode flag
1	Print the menu
XICE	Internal 68020 cache enable
XICD	Internal 68020 cache disable
XMSG	Toggle print message flag
XQUIT	Quit test program

8.3.3. DESCRIPTION OF CPU2 DIAGNOSTIC COMMANDS

XBMAP

This tests the new big segment map. The new big segment map requires that 32 Mbyte addressing mode be set in the status register. The base address of the big segment map is 0x1800000. The address increment in 0x10000 steps. This program runs out of local memory. The map is tested by running two sections an address test and a data test. The address test fills all locations in the map with a unique value (a sequential count) then reads the data back and checks it. The data test writes different patterns to all the locations in the map then reads them back and checks them.

schematic locations:

CPU board segment map page U12H, U16H, and U20H

XCD

This tests the cache data rams. The program runs out of local memory. The start address is 0 and the addresses are incremented by four until 4K(dec) long words have been written. These rams are tested by writing the first pattern to all locations then reading each location back and writing the next pattern to that location. This is done until all patterns have been run. Then the start and stop addresses are reversed and the test is run again. The Cache Data test uses the same test patterns as the bigsegmap test(XBMAP).

schematic locations:

CPU board Cache data page U16D, U18D, U20D, U22D, U24D, U26D, U28D, U30D, U14B, U26H, U30M, U12C, U14C MEMWRT generation page U24F, U32E **XCM** The cache miscellaneous test check for the following:

- 1. That read hits don't invalidate data in the cache.
- 2. That read misses invalidate data in the cache.
- 3. That write hits don't invalidate in the cache.
- 4. That write misses invalidate data in the cache.
- 5. That the valid bits compare while doing all the above cases.

This program runs out of local memory.

schematic locations:

CPU board Cache tag page Tag rams U24C, U26C, U28C Valid rams U16C, U18C, U20C, U22C, U26A, U24A, U26H, U22K, U26H, U28A, U24A, U26E, U24E, U22B, U20B Cache bypass page U32H, U16K, U20L, U18F VALCS generation page U30C, U26B, U32C MEMWRT generation page U24F, U32E Read Miss comparator page U20A, U22A, U30B

XEREG This tests the error register. The program runs from multibus memory. The program assumes that no memory is above 0x800000 so that a bus error will be generated. After a bus error is generated the error register is read and compared with a expected value. The following are the bus address that are used and the expected pattern in the error register. NOTE: This test will generate error messages if a CCB board is in the system.

schematic locations:

Multibus board Multibus Commander U12H, U14M, U16M, U18M, U20M

CPU Board Cpu Local Commands U30L Cpu Errors and Timeout U26M

XINTST This tests level 3 and level 6 interrupts. The program runs from multibus memory. These two interrupts are generated by the cpu board. Level 3 is generated when a read is done to the uart chip. Level 6 is generated by putting the timer chip into mode 0 and giving the chip a count of 100.

schematic locations:

Multibus board Timer and Uarts page U8L, U8F, U8K Interrupts page U26C, U28B, U26B This tests the multibus listener logic and the multibus address lines. This program runs out of local memory. The strategy is to walk an one in a address field of zeros then reverse and walk a zero in a address field of ones. With suicide mode off and cache on write a block with various tags. Turn suicide mode on and cache off, write the same block with a tag of 0x200. Flush the write queue and with cache on verify that the block is still all there. With cache off and still in suicide mode, write the original block and flush the write queue. The above procedure is repeated with a tag of 0x1ff. There are six sections in the listener test:

section 1 and 2check for an inactive listenersection 3check for a miss on each 1section 4check for a hit on each 1section 5check for a miss on each 0section 6check for a hit on each 0

schematic locations:

Multibus board Multibus listener page U12K, U16L, U20K, U22K, U24K, U26E U18K, U20L, U22L, U16K, U14F

CPU board Cache tag page Tag rams U24C, U26C, U28C Valid rams U16C, U18C, U20C, U22C, U26A, U24A, U26H, U22K, U26H, U28A, U24A, U26E, U24E, U22B, U20B

XALLT Run all the tests. If no parameter is entered for count the test will run until a control C is entered or the white button is pressed. Count is a hex number representing the number of passes.

XRANDT This is a random memory test that runs out of multibus memory. The test does not have any safeguards to prevent it from overwriting the program so do not enter a start address between 0x20000 0x100000. The test generates both random addresses and random data for an 8k(dec) block of memory from the starting address.

XRM This tests the read miss comparator. This program runs out of local ram. The comparator is tested by walking a one thru an address field of zeros, then walking a zero thru an address field of ones and causing both hits and misses for each bit. The test is careful not to use tags 0x004-0x007 as these tags map to the local devices.

schematic locations:

CPU board Read Miss comparator page U20A, U22A, U30B Cache tag page Tag rams U24C, U26C, U28C Valid rams U16C, U18C, U20C, U22C, U26A, U24A, U26H, U22K, U26H, U28A, U24A, U26E, U24E, U22B, U20B

XLI

XTFP This tests the floating point coprocessor. The coprocessor is checked by having it do a floating point add, subtract, multiply, and divide.

schematic location:

CPU board FPU page U8M

XTV This tests the tag and valid rams. This program runs out of local memory. The testing is done by inhibiting read misses to make the valid-tag latch work. Then for each cache location it will verify that all valid bits can be set together, each valid bit can be set individually, and that patterns can be written and read for each tag.

schematic locations:

CPU board Cache tag page Tag rams U24C, U26C, U28C Valid rams U16C, U18C, U20C, U22C, U26A, U24A, U26H, U22K, U26H, U28A, U24A, U26E, U24E, U22B, U20B Cache bypass page U32H, U16K, U20L, U18F VALCS generation page U30C, U26B, U32C MEMWRT generation page U24F, U32E Read Miss comparator page U20A, U22A, U30B

XWQA This tests the write queue. The program runs out of local ram. To test the write queue it writes to 32 long words at address 0x50000, then reads back 16 long words. The pattern used is 1abcdef0 it is incremented by one for each new location.

schematic locations:

Multibus board

Write Queue page U14D1, U14D2, U14E, U16D1, U16D2, U16E U18D1, U18D2, U18E, U20D2, U20E, U22D2 U22E, U24D2, U24E U20D1, U22D1, U24F, U22C, U20C, U24D1, U26F, U24B, U24C U26D1, U28C, U28E, U28F, U28D, U22H, U24H, U26H, U28H, U22B, U18C

XWQD This tests the write queue. The program runs out of local ram. To test the write queue it writes to 32 long words at address 0x50000, then reads back 16 long words.

schematic locations:

Multibus board

Write Queue page U14D1, U14D2, U14E, U16D1, U16D2, U16E U18D1, U18D2, U18E, U20D2, U20E, U22D2, U22E, U24D2, U24E U20D1, U22D1, U24F, U22C, U20C, U24D1, U26F, U24B, U24C, U26D1 U28C, U28E, U28F, U28D, U22H, U24H, U26H, U28H, U22B, U18C

CU	This test uses the cache as any user would. It runs in local memory. The program enables cache then writes 8K bytes(using long words) of data(A's) starting at address 0x100000. It then reads and compares 4K bytes(using byte accesses). One of the failure modes of this test is the program will hang. The hang is most likely caused by the cache always getting a read miss.
	schematic locations:
	CPU board Read Miss Comparator page U20A, U22A
XBCE	Enable board cache.
XBCD	Disable board cache.
XHALT	Flip the halt flag on or off depending on its current state. The halt flag causes the next test run to stop execution on the first error.
t	Display a menu of the commands.
XICE	Enable the 68020 cache.
XICD	Disable the 68020 cache.
XMSG	Flip the Quiet flag on or off depending on its current state. The xmsg command will suppress the printing of all messages.
QUIT	Return to the resident monitor.

8.4. XPIB Peripheral Interface Board Diagnostics

--

· _

--

8.5. XVGB Video Graphics Board Diagnostics

--

· .

--

8.6. XZOID Zoid (Realdraw) Board Diagnostics

8.6.1. INTRODUCTION

This section describes how to run the diagnostic programs written for the Zoid board (Valid PN #710-00104).

8.6.2. DESCRIPTION

The Zoid board is basically a high performance auxiliary processor board. The Zoid board contains a 68020 microprocessor running at 12 MHz, 1 Megabyte of on-board dynamic ram, a 32-bit status register (accessible in 16-bit words), and associated multibus circuitry. All local devices (e.g. status register, counter timer chip) and local memory are accessible from the multibus as well as the 68020 microprocessor. Additionally, the local ram is "dual port" ram, and is accessible to multibus masters while the local 68020 is running.

The Zoid diagnostics are designed to test out the functioning of the Zoid board in the following manner:

- 1. Test multibus access to local devices and local memory.
- 2. Test local access to local devices and proper functioning of local memory, parity, and interrupts.
- 3. Test multibus access to Zoid local devices and memory while Zoid 68020 is accessing multibus memory (e.g. color display memory).

8.6.3. EQUIPMENT REQUIRED

The following equipment is required to run the Zoid board diagnostics:

- Test card cage with LH power supply and tape drive.
- Terminator board.
- Rimfire disk/tape controller.
- CPU board (68010 version or 68020 version).
- 2 1/2-Megabyte ECM boards or 1 2-Megabyte ECM board.
- 1 Color Controller Board/Color Ram Board set.
- 1 Color monitor and cable to Color Controller Board.

8.6.4. SETTING UP THE SYSTEM

- 1. Set the address of the color controller board to 0x390000.
- 2. Set the Zoid address to 0xE80000 by setting the switches to the following:

		MS	B	LSB				
x	X	x	Γ	X				ON
-	1		X		X	X	Χ	OFF
8	7	6	5	4	3	2	1	

NOTE Reading the Zoid switches can be confusing! Switch 8 is the most significant bit (MSB) and switch 1 is the least significant bit (LSB). For example, to set the Zoid address to 0x9C0000, set the switches to the following:

MSB					LSB			
Х		Γ	x	x	X			ON
	x	x				X	X	OFF
8	7	6	5	4	3	2	1	

2. Insert the Zoid into a master slot in the card cage.

8.6.5. RUNNING THE DIAGNOSTICS

After the program is started, the diagnostic will inform the operator of the default addresses of the Zoid board and the color controller board. (These addresses are the standard Scaldstar system addresses):

XBDTST>XZOID < CR>Default zoid address is 0xE80000Default ccb address is 0x390000Type ? for command menu

ZOID>

XALLT	Run all tests
XWBT	Word
XADDR	Zoid address line test
XDATA	Data pattern testing on soid memory
XBANK	Data pattern testing on 1 bank of zoid memory
XTIMER	Check out zoid 8254 timer chip
XFLASH	Flash zoid leds
XREFRESH	Set up CTC to start refreshing ram
XRESET	Reset the Zoid board
XSCOPE	Run scope loop
XSELECT	Enter zoid board address
QUIT	Return to BDTST prompt
XCRM	Return to XCRM prompt
1 STAT	Display zoid status register routines
? ZOID	Display zoid self-test routines
?	Print this menu

ZOID>

Type XALLT to run the entire battery of diagnostics. XALLT will execute the following diagnostics, in this order:

- 1. Status Register Test (same as XSTATIST)
- 2. Timer (8254 CTC) Test (same as XTIMER)
- 3. Long/word/byte Test (same as XWBT)
- 4. Zoid Ram Address Test (same as XADDR)
- 5. Zoid Ram Data Test (same as XDATA)
- 6. Zoid Self-tests (same as ZTEST)
- 7. Zoid Color Display (same as ZCOLOR)

The order of the tests listed above is generally the order that board debug should take. The status register and timer must work before any multibus memory operations can be performed. Multibus long/word/byte reads and writes must operate correctly before the address and data tests will pass. If Zoid self-tests do not pass, then try running ZFLASH first, to see if the 68020 on the Zoid will run code at all. After that, try running ZDEAD to verify that the board can access multibus memory while the CPU board is accessing zoid memory. If ZDEAD operates correctly, then ZTEST should begin to execute properly--the zoid should print pass/fail messages on the console terminal. After ZTEST passes the tests, ZCOLOR should run with no problems-- it is mainly a sanity check to verify that the zoid can write to color display memory.

8.6.6. OTHER MENUS

Besides the menu listed above, there are two other menus in the diagnostics. The commands to display both of the menus are listed in the main menu above:

7ZOID Typing this command will display a menu of commands to load code into the Zoid's local ram and start the Zoid's 68020 running:

ZFLASH	Load 68020 code to flash zoid leds
ZCOLOR	Load 68020 code to print out color display
ZDEAD	Load 68020 code to test deadlock conditions
ZTEST	Load and start soid self-diagnostics
XRESET	Reset the Zoid board
XCRM	Return to XCRM prompt
QUIT	Return to BDTST prompt
I STAT	Display zoid status register routines
? ZOID	Display zoid self-test routines
?	Display main menu

ZOID>

fSTAT Typing this command will display a menu of commands to display and alter the contents of the Zoid status register, and to test the status register:

XSTATIST	Zoid status register test
XHSTAT	Display zoid status register (high word)
XLSTAT	Display zoid status register (low word)
XHSETSTAT	Modify zoid status register (high word)
XLSETSTAT	Modify zoid status register (low word)
QUIT	Return to BDTST prompt
XCRM	Return to XCRM prompt
? STAT	Display zoid status register routines
?ZOID	Display zoid self-test routines
?	Display main menu

ZOID>

8.6.7. GENERAL COMMENTS

- 1. To exit a command before it is finished, type $\langle CTRL \rangle C$.
- 2. To suspend messages on the screen, type S. Type Q to resume.
- 3. All Zoid diagnostic commands are executable, regardless of the Zoid diagnostic menu currently displayed.

8.6.8. DESCRIPTION OF ZOID DIAGNOSTIC COMMANDS

The following is a description of all Zoid diagnostic commands. For a more detailed explanation of each diagnostic, see the document entitled ZOID DIAGNOSTIC ALGORITHMS.

XALLT This command will execute the following diagnostics, in this order:

- 1. Status Register Test (same as XSTATIST)
- 2. Timer (8254 CTC) Test (same as XTIMER)
- 3. Long/word/byte Test (same as XWBT)
- 4. Zoid Address Test (same as XADDR)

- 5. Zoid Data Test (same as XDATA)
- 6. Zoid Self-tests (same as ZTEST)
- 7. Zoid Color Display (same as ZCOLOR)
- XWBT This diagnostic tests whether the system CPU can write bytes and read words, and write words and read bytes in the Zoid local memory. Error message reports the address, expected data, and actual data.
- XADDR This diagnostic tests system CPU access to the dual port ram on the Zoid. From Multibus, Zoid memory is accessed through sixteen 64k banks (for Zoid with 4 Megabytes of local RAM, there are 64 64k banks). The bank is selected by setting the memory map bits of the low word of the status register. This diagnostic tests the addressing of Zoid local memory, one bank at a time. Error message reports the last two addresses that map to the same memory location in that bank.
- XDATA This diagnostic tests system CPU access to the dual port ram on the Zoid. From Multibus, Zoid memory is accessed through sixteen 64k banks (for Zoid with 4 Megabytes of local RAM, there are 64 64k banks). The bank is selected by setting the memory map bits of the low word of the status register. This diagnostic tests data reads and writes to Zoid local memory, one bank at a time. Any errors are reported with the address, the expected data, and the actual data.
- XBANK This diagnostic tests system CPU access to one bank of the dual port ram on the Zoid. This command prompts the operator for the bank to test. From Multibus, Zoid memory is accessed through sixteen 64k banks (for Zoid with 4 Megabytes of local RAM, there are 64 64k banks). The bank is selected by setting the memory map bits of the low word of the status register. This diagnostic tests data reads and writes to one bank of Zoid local memory. Any errors are reported with the address, the expected data, and the actual data.
- XTIMER This diagnostic tests whether the system CPU can write and read each counter register in the Zoid CTC. Error message prints the counter number, the expected data, and the actual data.
- XFLASH This command flashes the zoid red and green leds four times.
- XREFRESH This command initializes and starts the 8254 CTC, thus starting up the refresh cycle of the dynamic rams.
- XRESET This command halts the 68020 on the Zoid board.
- XSCOPE This command calls up the scope loop menu. The menu should be selfexplanatory.
- XSELECT This command prompts the operator for the addresses of the Zoid board and the color controller board. All subsequent diagnostics will reference the new Zoid and ccb addresses. Typing $\langle CR \rangle$ for either prompt will set the default address.
- *\$STAT* This command displays a menu of commands relating to the Zoid status register:
 - XSTATTST This diagnostic tests the Zoid status register high word and then the status register low word. Bits 10 and 11 of the high word of the status register are read-only, so they are not tested. Starting with bit 15, this diagnostic sequentially sets each bit, then tests to see if the bit was set. After all bits have been set, starting with bit 15 of the high word (or low word), the diagnostic sequentially resets each bit, then tests to see if the bit was reset. Error message gives contents of status register, and states which bit should have been set (or reset).
 - XHSTAT This command displays the contents of the high word of the Zoid status register.

- XHSETSTAT This command will modify the high word of the Zoid status register and display the contents in the same format as the *hstat* command.
- XLSETSTAT This command will modify the low word of the Zoid status register and display the contents in the same format as the *lstat* command.
- QUIT This command will exit out of the Zoid diagnostics and return to the BD TST prompt.
- XCRM This command will exit out of the Zoid diagnostics and return to the XCRM prompt.
- *fSTAT* This command displays the menu of commands relating to the Zoid status register.
- ?ZOID This command will display the Zoid self-test menu.
- ? This command will display the main Zoid diagnostic menu.
- *?ZOID* This command displays a menu of commands which load code into the Zoid board and start the 68020 running.
 - ZFLASH This command loads 68020 code into Zoid's local ram to flash the Zoid's leds.
 - ZCOLOR This command loads 68020 code into the Zoid to constantly display a pattern on the color screen. After the Zoid is loaded and started, the diagnostic will prompt the operator for a carriage return to start doing a memory test on the zoid board. This diagnostic requires a color monitor connected to a color controller board installed at 0x390000 and a color ram board.
 - ZDEAD This command loads 68020 code into the Zoid's local ram to flash leds on the color controller board. After the Zoid is loaded and started, the diagnostic will prompt the operator for a carriage return to start doing a memory test on the zoid board. This test verifies the Zoid's access to multibus memory while the cpu is accessing Zoid's memory, and that deadlock circuitry functions correctly.
 - ZTEST This command loads and starts the Zoid. If the Zoid is loaded correctly and executes code properly, a Zoid self-test menu will be displayed on the console screen:
 - X1 run all self-tests
 - X2 zoid 8254 timer chip self-test
 - X3 zoid status register self-test
 - X4 zoid interrupt self-test
 - X5 long/word/byte self-test
 - X6 bhen test
 - X7 multibus I/O test
 - X8 multibus data test
 - X9 zoid address line self-test
 - XA zoid data pattern self-test
 - XB test parity error circuitry
 - XC 68881 test
 - XD zoid on-board scope loop
 - Q Return to Zoid command prompt
 - ? Print this menu

Type your command:

For each of the tests above, the pass/fail information and error messages are printed on the CPU console terminal.

- X1 This command will execute the following on-board diagnostics, in this order:
 - 1. Timer test

This diagnostic tests counter 2 of the on-board 8254 CTC and related circuitry (same as X2).

2. Status register test

This diagnostic tests the Zoid status register high word and then the status register low word (same as X3).

3. Interrupt test

This diagnostic tests exception handling of bus errors, interrupt levels 3 through 5, and related circuitry (same as X4).

4. Long/word/byte test

This diagnostic tests long word, short word, and byte accesses to Zoid memory by the Zoid's 68020 (same as X5).

5. Multibus bhen test

This diagnostic tests the bhen circuitry on the zoid board by writing a word to multibus memory, and reading bytes, and by writing bytes and reading words (same as X6).

6. Multibus I/O space test

This diagnostic tests Zoid's access to multibus I/O space by flashing the green and red leds on ECM 0 (same as X7).

7. Multibus data test

This diagnostic tests Zoid's access to multibus memory by writing and reading patterns to multibus addresses 0x100000 to 0x140000 (same as X8).

8. Local ram address test

This diagnostic tests the address lines to the Zoid local memory (same as X9).

9. Data test

This diagnostic tests data reads and writes to Zoid local memory (same as XA). Any errors are reported with the address, the expected data, and the actual data.

- X2 This diagnostic tests counter 2 of the on-board 8254 CTC and related circuitry.
- X3 This diagnostic tests the Zoid status register high word and then the status register low word. Starting with bit 15 to bit 1, this diagnostic sequentially sets each bit, then tests to see if the bit was set. After all bits have been set, starting with bit 15 to bit 1, the diagnostic sequentially resets each bit, then tests to see if the bit was reset. The error message gives contents of status register, and states which bit could not be set (or reset).
- X4 This diagnostic tests exception handling of bus errors, interrupt levels 3 through 5, and related circuitry.

- X5 This diagnostic tests long word, short word, and byte accesses to Zoid memory by the Zoid's 68020.
- X6 This diagnostic tests the bhen circuitry on the zoid board by writing a word to multibus memory, and reading bytes, and by writing bytes and reading words.
- X7 This diagnostic tests Zoid's access to multibus I/O space by flashing the green and red leds on ECM 0.
- X8 This diagnostic tests Zoid's access to multibus memory by writing and reading patterns to multibus addresses 0x100000 to 0x140000.
- X9 This diagnostic tests the address lines to the Zoid local memory. Error message reports the last two addresses that map to the same memory location in that bank.
- XA This diagnostic tests data reads and writes to Zoid local memory. Any errors are reported with the address, the expected data, and the actual data.
- XB This diagnostic tests the parity error detection circuitry by forcing a parity error, and then verifying the correct parity error. Error messages report that a parity error was not received, and report the contents of the parity error register.
- XC This diagnostic tests rudimentary arithmetical operations using the optional on-board 68881 floating point co-processor. It tests floating point add, subtract, multiply, and divide operations. This diagnostic should be run only on Zoid boards with 68881 co-processor chip installed.
- XD This diagnostic invokes an on-board scope loop to aid in debugging the Zoid board. To exit the scope loop, the operator must reset the system.
- Q This command will return system control to the CPU in the ZOID menu.
- ? This command will display the Zoid self-test menu on the console screen.
- NOTE If the operator types $\langle CTRL \rangle C$ while running the Zoid self-tests, the self-test commands will not execute properly any more because the CPU is no longer communicating with the Zoid board. To run the Zoid self-tests again, type Q for quit. This will return operation to the XCRM prompt.
- XRESET This command halts the 68020 on the Zoid board.
- XCRM This command will exit out of the Zoid diagnostics and return to the XCRM prompt.
- QUIT This command will exit out of the Zoid diagnostics and return to the BD TST prompt.
- *STAT* This command displays the menu of commands relating to the Zoid status register.
- ?ZOID This command will display the Zoid self-test menu.
- ? This command will display the main Zoid diagnostic menu.
- This command will display the main Zoid diagnostic menu.

ş