CMI-41

WAVEFORM SUPERVISOR

fairlight

FIELD CHANGE NOTICE

DATE 13/ 08/ 93

NUMBER 132

ORIGINATOR Chris Alfred

PRODUCT: CMI / MFX

ASSEMBLY No. CMI 41 DESCRIPTION WAVEFORM SUPERVISOR

This FCN applies to REV No: 1.12 New REV No is: 1.13

REASON FOR CHANGE:

DMA OVERRUN OR " DMA ERROR 02 " CAN OCCOUR DURING STEREO RECORDING OR DUAL MONO RECORDING, DUE TO TIMING DIFFERENCES BETWEEN THE P1/P2 BUS AND THE WAVE SUPER BUS. THIS FCN MUST BE DONE IN CONJUNCTION WITH FCN 131.

DETAILS OF CHANGE:

(1) REPLACE C2GAL AT LOCATION C5 WITH C2GAL2

(2) MARK PCB WITH NEW REVISION NUMBER

ORIGINATOR: C ALFRED DATE: 11/8/93

SERVICE: [signature] DATE: 13/8/93

TEST: [signature] DATE: 13/8/93

PROD: M. Peebles DATE: 13/8/93

KIT LIST CHANGE: YES NO

fairlight

FIELD CHANGE NOTICE

DATE 09/ 08/ 93

NUMBER 131

ORIGINATOR Chris Alfred

PRODUCT: CMI / MFX

ASSEMBLY No. CMI 41 DESCRIPTION WAVEFORM SUPERVISOR

This FCN applies to REV No: 1.11 New REV No is: 1.12

REASON FOR CHANGE:

Sampler overrun errors when using the Waveform Supervisor SCSI port in Rev 11 and later software. Fast Hard Drives tend to overload the DMA Controller resulting in Samples not being processed in time and an overrun occours, this mod restricts the duration that the SCSI port can hold the DMA Controller.

DETAILS OF CHANGE:

1a\ Replace R7 which is currently 220K ohm with a 150K ohm resistor

alternatively

1b\ Connect 680K ohm resistor in parallel with existing 220K ohm resistor

2\ Mark PCB with revision new revision.

ORIGINATOR:	DATE:	TEST:	DATE:
		PROD:	DATE:
SERVICE:	DATE:	KIT LIST CHANGE:	YES NO

fairlight

FIELD CHANGE NOTICE

DATE 8/ 2 / 93
NUMBER 119

ORIGINATOR Steve Rance **PRODUCT: CMI / MFX**

ASSEMBLY No. CMI 41 **DESCRIPTION** WAVEFORM SUPER

This FCN applies to REV No: REV 1.10 **The New REV No is:** REV 1.11

REASON FOR CHANGE:
Diagnostic bug fix in Rom 8.01 concerning DMA hang.

On start up screen displays hard ware protection level, A/B/C/D.

DETAILS OF CHANGE:

Replace ROM with KMON 20-512 Rev 8.05 and at LK 5 link pins 1&2 to Change.

Must be done with FCN 108

Relabel to indicate new revision 1.11

ORIGINATOR:	**DATE:**	**PROD:**	**DATE:**
SERVICE:	**DATE:**	**KIT LIST CHANGE:**	**YES NO**

fairlight

FIELD CHANGE NOTICE

DATE 22/ 10/92
NUMBER 109A

ORIGINATOR Steve Rance **PRODUCT: CMI / MFX**

ASSEMBLY No. CMI-41 **DESCRIPTION** Waveform Supervisor

This FCN applies to REV No: Rev 1.9 **The New REV No is:** Rev 1.10

REASON FOR CHANGE:
THIS FCN REPLACES FCN 109

Unified ROMS for all revisions of CMI
Q219
CG-1
CG-2
CG-3

DETAILS OF CHANGE:

Replace ROM with KMON 20-512 Rev 8.01 and at LK 5 link pins 1&2

Must be done with FCN 108 see attached

ORIGINATOR:	**DATE:**	**PROD:**	**DATE:**
SERVICE:	**DATE:**	**KIT LIST CHANGE:**	**YES NO**

ESP

FIELD CHANGE NOTICE

Field Change Notice No 91

PRODUCT CMI Series III

ORIGINATOR: Chris Alfred **DATE:** 3 / 6 / 92

ASSEMBLY No: CMI 41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.8 **The New rev No is:** 1.9

REASONS FOR CHANGE:

Terminate sampler interface to remove glitches.
This stops clicks during sampling.

DETAILS OF CHANGE:

1. Connect a 330R resistor between J2/7 (10 way IDC sampler interface connector) and J2/8.
2. Connect a 330R resistor between J2/9 and J2/10.
3. Replace R19 with a 470R resistor.
4. Replace R20 with a 470R resistor.
5. Mark PCB as Rev 1.9

NOTE: to be done in conjunction with FCN 90
some CMI 41 PCBs may already have 470R for R19 and R20.

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager			
Customer Service			

ESP

FIELD CHANGE NOTICE

Field Change Notice No 86

PRODUCT CMI Series III

ORIGINATOR: Chris Alfred **DATE:** 24 / 3 / 92

ASSEMBLY No: CMI 41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.7 **The New rev No is:** 1.8

REASONS FOR CHANGE:

a) Latch request to waveform buss 100nsec before start of waveform buss cycle ensuring adequate detection time by Turbo SCSI card.

b) Missing Track may cause DMA over run during Monitoring or Recording.

DETAILS OF CHANGE:

1) Replace ASPAL/ASGAL with ASGAL3 aligning Pin 1 of the socket with pin 2 of the ASGAL3. (The original Ic was a 20 pin 16v8 which is now replaced by a 24 pin GAL22v10)
2) On ASGAL3 connect pin11 to pin 12 , on the Ic.
3) On ASGAL3 connect pin23 to pin 24, on the Ic.
4) Connect Ic A16/16 (74hct240) to Ic A16/15
5) Connect Ic A16/5 to ASGAL3/1
6) Connect Ic D2/12 (74ls112) to Ic D2/8 (relates to b)

7) Mark the board Rev 1.8

NOTE: ASGAL3 must be a Lattice GAL 22V10 15nsec

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager	C E Alfred	6/4/92	
Customer Service			

ESP

FIELD CHANGE NOTICE

Field Change Notice No 80A

PRODUCT CMI [X] CVI [ ]

ORIGINATOR: Chris Alfred **DATE:** 11 / 9 / 91

ASSEMBLY No: CMI 41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.6 **The New rev No is:** 1.7

REASONS FOR CHANGE:

This FCN is required when Rev 10 software is installed.
It is NOT necessary for Rev 9 and below.

1M x 4bit DRAMs (4M DRAMs) require the *WE (write enable) to remain high during CAS-before-RAS refresh cycles. The 256K x 4bit DRAMs (1M DRAMs) considered the *WE signal as DONT CARE during refresh.

DETAILS OF CHANGE:

Replace DRPAL with DRGAL2
Cut component side track from ICB2/18 (74HCT244) to the plate-thru between ICC1/10 (74HCT240) and the SPARE socket pin 1.
Connect IcB2/18 (74HCT244) to DRGAL2/8.
Connect RN3/8 (33R at D5) to DRGAL2/14.
Replace current Ram with 1M x 4bit DRAMs
LK3 located at D6 - Remove the wire linking pins 1&2. Add a wire link between 2&3.

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager			
Customer Service			

ESP

FIELD CHANGE NOTICE

Field Change Notice No 77

PRODUCT CMI [X] CVI [ ]

ORIGINATOR: Chris Alfred **DATE:** 9 / 7 / 91

ASSEMBLY No: CMI-41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.5 **The New rev No is:** 1.6

REASONS FOR CHANGE:

The reverse bias doides remove glitches generated by the opto isolators during current switching.

** This FCN is to be done in conjunction with FCN 75 & 76 **

DETAILS OF CHANGE:

Orient the card such that, viewing the wiring side, the gold fingers are pointing down.
Locate the10 way connector J2 and the unused right hand mounting hole. Just below this hole there are a group of 6 tracks running horizontally (going to Ic F3/1 (i.e. HP2631 pin 1) & Ic F3/2 & Ic G3/1 (HP2631) & Ic G3/2 & Ic G3/3 & Ic G3/4.

1. Cut down through all 6 tracks.
2. Connect 1N4148 diode: Cathode to IcF3/1 Anode to IcF3/2
3. Connect 1N4148 diode: Cathode to IcG3/1 Anode to IcG3/2
4. Connect 1N4148 diode: Cathode to IcG3/4 Anode to IcG3/3
5. Connect J2/1 (10 way IDC connector) to Ic G3/2 (HP2631)
6. Connect J2/2 to Ic G3/1
7. Connect J2/3 to Ic G3/3
8. Connect J2/4 to Ic G3/4
9. Connect J2/5 to Ic F3/2 (HP2631)
10. Connect J2/6 to Ic F3/1

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager			
Customer Service			

ESP

FIELD CHANGE NOTICE

Field Change Notice No 72

PRODUCT CMI [X] CVI [ ]

ORIGINATOR: Chris Alfred **DATE:** 29 / 5 / 90

ASSEMBLY No: CMI-41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.4 **The New rev No is:** 1.5

REASONS FOR CHANGE:
Inaccurate Detection of Channel Card slices

The *TSTAKEN signal is a wire-or signal terminated at the Waveform Supervisor. Currently, *TSTAKEN is teminated at 2.5V. This has been found to provide inadequate noise immunity. To fix the problem, *TSTAKEN is now terminated at 4V.

DETAILS OF CHANGE:

Replace R4 (820R) with 120R.

** TO BE DONE IN CONJUNCTION WITH FCN 73 **
FCN 73 on CMI31 Channel card, changing Ic 74ls03 to 7438 supports the higher current.

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager	C E Alfred	22/8/91	
Customer Service	M. Paolino	22/8/91	

ESP

FIELD CHANGE NOTICE

Field Change Notice No 71

PRODUCT CMI [X] CVI [ ]

ORIGINATOR: Chris Alfred **DATE:** 29 / 5 / 91

ASSEMBLY No: CMI-41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.3 **The New rev No is:** 1.4

REASONS FOR CHANGE:
Corruption during Waveform Memory Refresh

Waveform memory corrupted during refresh cycles as a result of glitches on *WREF signal. The solution is to buffer the *WREF signal with a 74HCT244 buffer; rather than drive the signal directly from WBPAL.

DETAILS OF CHANGE:

Cut track to edge connector pin 71A at the edge connector on wiring side.
Connect ICB16 (WBPAL) pin 16 to ICB2 (74HCT244) pin 11.
Connect ICB2 (74HCT244) pin 9 to edge connector pin 71A.

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager	C E Alfred	22/8/91	
Customer Service	M. Paolini	22/8/91	

ESP

FIELD CHANGE NOTICE

Field Change Notice No 60

PRODUCT CMI [X] CVI [ ]

ORIGINATOR: Chris Alfred **DATE:** 7 / 8 / 90

ASSEMBLY No: CMI 41

DESCRIPTION: Waveform Supervisor

This FCN applies to rev No: 1.2 **The New rev No is:** 1.3

REASONS FOR CHANGE:

Removes noise generated by ESP-RT1 router on power-up.
Removes distortion of channels due to uninitialised router latches.

DETAILS OF CHANGE:

Replace EPROM KMON with Rom KMON 3.19

DEPT	SIGNATURE	DATE	COMMENTS
Project Manager			
Customer Service	Mario Paolini	7/8/90	

1) Overview

The CMI-41 Waveform Supervisor (WS) has the primary functions of

1) managing waveform memory.
2) interfacing to the sample card.
3) providing the interface between the CMI and the external SCSI bus.

The Waveform Super is also used for waveform manipulation and calculations, and for any other functions requiring a large computational load. It is anticipated that more system functions will be taken over by the Waveform Super as the CMI evolves.

1.1) Description

A brief description of each component of the waveform supervisor follows:

CPU section - comprising CPU, FPU, DMA controller, EPROM, dynamic RAM and clock. Facilities are provided for interrupt handing and internal control functions. Clocks for the CPU and for the waveform bus interface are generated by the channel support card.

CMI bus interface - this allows the waveform supervisor direct access to two full 64KB maps in the 6809 bus address space. All communication between between the waveform super and the 6809 software must take place using this mechanism, since the 6809 has no access to the CMI-41 internal bus or peripherals, except for the external control latch, which provides control of the RESET and HALT lines, and generation of interprocessor interrupts.

Accesses to the CMI bus are made using DMA cycles, and may steal cycles from either P1 or P2. Separate map select registers (MAPSELRAM) are provided for waveform super P1 and P2 cycles. The waveform super can access 6809 bus peripherals (including the MAPSELRAM) provided that the selected DMA map number has peripherals enabled (PERENB=0).

SCSI Interface - NCR5380 SCSI controller and associated interface circuitry for CPU and DMA transfers. This allows high speed (up to 1.1MByte/s) transfers between disk and waveform memory. The actual transfer rate achieved is dependent on the the type of drive and controller. 68020 software must be provided to allow access to the disk by 6809 programs. SCSI bus terminating resistor packs may optionally be fitted.

Waveform Bus Interface - provides access to the 3.3MHz synchronous waveform bus. Also generates control signals during channel card accesses to waveform memory. Up to 64MB of waveform memory can be managed by the CMI-41 (requires 8MB waveform RAM cards), of which the first 32MB is accessible by the channel cards. The remaining 32MB may be used for general workspace, disk cache, RAM disk, preloading of voice data etc.

Sample Card Interface - provides access to the CMI-337 or CMI-347 sample card. Communication is via a 17 bit synchronous serial link running at up to 10Mbit/s. In normal operation, samples are received from the sample card and control data is transmitted to the sample card at the same time. Provision is made for samples to be transmitted as well although current sample card hardware does not support this.

2) Detailed Circuit Description

The following subsections correspond directly with the pages of the schematic diagram.

1 CPU/FPU etc
2 EPROM
3 DRAM
4 DMAC
5 SCSI INTERFACE
6 WFM INTERFACE
7 SAMPLE CARD INTERFACE
8 CMIBUS INTERFACE
9 EXTERNAL CONTROL LATCH
10 INTERNAL CLOCK

2.1) CPU/FPU etc

The CPU is a 32 bit Motorola MC68020 running at 10MHz.
Hardware features of this processor include:

1) dynamic bus sizing, which allows devices connected to the internal bus of the CMI-41 to be either 8, 16 or 32 bits wide. The bus size in use is encoded by the peripheral device on the control lines DSACK0/DSACK1.
2) bus cycle length of 300nS (with no wait states).
3) 7 levels of vectored interrupts, of which 3 are used by internal CMI-41 hardware. The remaining 4 levels can be generated by writing to the external control latch.
4) 4GB address space, of which the CMI-41 used 128MB (27 bits).

A brief description of a CPU bus cycle is as follows:

State 0 begins on a rising edge of PCLK. During this state the CPU places the address on the bus, including function codes FC0-2, data size codes SIZ0-3 and R/W.

State 1 begins at the next falling edge of PCLK. During this state the CPU asserts address strobe (AS) which indicates to external devices that the address is valid and a bus cycle has begun. In the case of a read cycle, data strobe (DS) is also asserted.

State 2 begins at the next rising edge of PCLK. During this state, in the case of a write cycle, the CPU places the data to be written on the data bus.

State 3 begins at the next falling edge of PCLK. If DSACK0/1 is asserted and synchronised by this falling edge then the CPU will continue into state 4 on the next clock edge, otherwise pairs of wait states will be inserted until DSACK0/1 is asserted. In addition, in the case of a write cycle, data strobe (DS) is asserted during this state.

State 4 begins at the next rising edge of PCLK. No action takes place during this state.

State 5 begins at the next falling edge of PCLK. In the case of a read cycle data is latched at the beginning of this state. In all cases, address strobe (AS) and data strobe (DS) are negated during this state.

The next rising edge of PCLK may be the beginning of the next bus cycle, in which case state 0 will begin immediately, otherwise the bus will become idle. Note that the address lines will generally remain at the address of the last bus cycle when the bus is idle.

For more details of CPU operation see the Motorola 68020 User's Manual.

A 16MHz MC68881 floating point coprocessor (FPU) may optionally be fitted. If the FPU is not present then any attempt to access it will generate a bus error signal to be generated, which will cause the CPU to take the F-line

emulator trap.

Address decoding is performed by ASPAL and IOPAL. Only the top few address lines are used for decoding, so most peripheral systems are found at several alias addresses. See figure 2.

Address Map:

00000000-007FFFFF:	EPROM or DRAM depending on RAMEN
00800000-009FFFFF:	CMI Bus access (P1=820000,P2=860000) *
00A00000-00FFFFFF:	not used
01000000-011FFFFF:	I/O section (see below)
01200000-013FFFFF:	NCR5380 SCSI controller
01400000-015FFFFF:	68450 DMA controller
01600000-017FFFFF:	EPROM
01800000-01FFFFFF:	DRAM
02000000-05FFFFFF:	Waveform Bus Access
06000000-077FFFFF:	not used
07800000-07FFFFFF:	DRAM

I/O section:

01000000-0100FFFF:	Internal Control Latch *
01010000-0101FFFF:	Sample Card interface
01020000-0103FFFF:	FPU (peripheral mode)

NB: Peripherals marked * have active high select lines, all others are active low.

Note that address range 00000000-007FFFFF is either EPROM or DRAM. Immediately after RESET signal RAMEN is negated, causing this address range to select EPROM. The first write by the CPU to the internal control latch causes RAMEN to be asserted, after which DRAM will appear in this address range. This allows restart vectors to be fetched from EPROM, to be replaced by RAM vectors when the system is up and running. The VBR (vector base register) could be used for this purpose, but that would cause compatibility problems with existing 68000 software. Both EPROM and DRAM are accessible at all times using other alias address ranges.

IOPAL also decodes CPU space accesses by the CPU. These are used for coprocessor (FPU) access and for interrupt acknowledge cycles. All interrupts except level 4 cause AVEC to be asserted (auto vector). Level 4 interrupt acknowledge is passed onto the DMA controller which supplies the appropriate vector number.

A bus watchdog timer is provided which will generate a bus error if AS is asserted for more than approx 30µS. This may be disabled by changing jumper LK7 to connect 2-3. In normal operation connect 1-2.

The internal control latch is used to configure the sample card interface and the SCSI interface. This is a write only register, so a copy must be kept in RAM to allow partial updates to be made. Bit definitions are as follows:

bit number	value	function
0	$01	SCSI interface DMA direction: 0=READ 1=WRITE
1	$02	SCSI interface DMA enable
2	$04	sample card interface - control bit 16
3	$08	(not used)
4	$10	sample card interface mode: 0=MONO 1=STEREO
5	$20	sample card interface DMA enable
6	$40	SCSI interface interrupt enable
7	$80	sample card interface interrupt enable

2.2) EPROM

A single 8 bit EPROM provides either 32KB or 64KB for system startup and debugging tools.

This makes use of the dynamic bus sizing capabilities of the CPU, which requires that 8 bit devices reside on the high 8 bits of the 32 bit data bus (D24-31). A 250nS access time device is used, requiring one wait cycle (100nS). To achieve this DSACK0 is asserted 100nS after AS, using signal DAS (address strobe delayed by 100nS).

Provision is made for either 256K or 512K devices via a jumper LK5. Link 1-2 selects 512K, link 2-3 selects 256K.

2.3) DRAM

1 MB of on board dynamic RAM (referred to as "private RAM") is provided, running with zero wait states. This is configured as 256K x 32 bits using 256K x 4 parts. Provision is made for fitting 1M x 4 parts when these become available via jumper LK3. Link 1-2 selects 256K x 4, link 2-3 selects 1M x 4.

Control signals are generated by DRPAL (see figure 3) and byte selection performed by BYPAL, as follows:

Initially, CA is negated, so the row address is present at the output of the address multiplexer as soon as the CPU sets up A0-31.

The CPU asserts AS on a falling edge of PCLK. Assuming that no refresh request is pending, RAS will be asserted immediately. This causes the DRAM chips to latch the row address. CA will be asserted on the next rising edge of PCLK, presenting column addresses at the output of the multiplexer. CA is internally gated with the actual RAM RAS signal (MRAS) to ensure sufficient row address hold time even if AS is not asserted until near the end of state 1.

CAS is asserted on the next falling edge of PCLK, latching the column address into the RAM chips. In the case of a write cycle, DS must also be asserted before CAS is asserted, since data will also be latched at this point.

The CAS output of DRPAL is fed into BYPAL, which generates a separate RAM CAS signal for each byte. Byte decoding is performed by BYPAL using the CPU signals A0-1, SIZ0-1 (see the MC68020 users's manual for details of byte decoding). In the case of a DMA cycle the DMAC signals A1, LDS and UDS are used to perform byte selection.

RAS, CA and CAS now remain asserted until AS is negated by the bus master, signalling the end of the cycle. In the case of CPU bus cycles, this will occur 100nS later on the next falling edge of PCLK. Since CAS also functions as the output enable for the RAM chips on read cycles, data will remain on the bus until CAS is negated.

Refresh is performed with CAS before RAS cycles, using the internal refresh counter of the DRAM chips. Refresh cycles take priority over CPU/DMAC cycles.

Arbitration between CPU/DMAC cycles and refresh requests is performed by sampling the refresh request pending (RPEND) signal gated with not-AS on the falling edge of PCLK. Both AS and RPEND are asserted shortly after this edge, so they are both guaranteed to be stable at this time.

CAS is asserted immediately RCYC is asserted (indicating a refresh cycle has begun) with BYPAL asserting all four RAM CAS lines. RAS is asserted one half clock later.

Refresh requests are generated by the CMI system refresh pulse (RFSH), which is a 500nS pulse every 16uS. This signal is synchronised and edge detected by DRPAL to yield LRFSH (latched refresh request). LRFSH is delayed by 100nS to give RPEND. LRFSH is negated as soon as RCYC is asserted, and the combination of LRFSH, RPEND and RCYC results in a 300nS pulse on RCYC.

2.4) DMAC

The DMA controller is a Motorola 68450. This is by no means the ideal par for use with the 68020, since it is a 68000 peripheral and has 24 bit address and 16 bit data busses, but there is nothing much else available.

To achieve a 128MB addressing range the three function code bits are used as extra address lines. The only disadvantage of this is that if a transfer crosses a 16MB boundary then A24-26 will not be incremented. This must be allowed for in software by splitting such a transfer into two separate blocks.

The 16 bit data bus is not a problem functionally, since the DMA devices ir the system are only 8 or 16 bits wide anyway. However provision must be made for the DMAC to access the 32 bit wide DRAM. The DMAC data bus is connected to D16-31, as required by the 68020 for a 16 bit peripheral. When the DMAC accesses an odd word address (A1 = 1), BYPAL asserts the signal LOWORD, which enables 74ACT245 buffers (F8, F9) to connect the upper and lower words of the data bus, so that the DMAC can access D0-15.

The DMAC also requires a buffer (F11) to connect the upper and lower bytes of its data bus so that 8 bit DMA devices can access both halves of the 16 bit bus. The various combinations possible for these three buffers are shown schematically in Figure 4.

Note that no provision is made for the DMAC to access the 8 bit EPROM, and any attempt to do so will generate a bus timeout error.

The DMAC is connected to the 8 bit SCSI Interface and the 16 bit Sample Card Interface. In both cases single address mode (implicit addressing with ACK) is used, although dual address mode is possible. Refer to the 68450 manual for descriptions of the timing of bus cycles. Note that the 68450 asserts address strobe (AS) on the rising edge of its clock. To make this consistent with the 68020, the DMAC clock input is connected to the inverse of the CPU clock (PCLK). This is necessary to ensure correct functioning of the dynamic RAM and SCSI interface peripherals.

The DMAC generates two data strobes, UDS and LDS, which also perform byte selection. These are combined (ORed) by BYPAL to generate DS when the DMAC is bus master. When the CPU is bus master BYPAL generates UDS and LDS from DS and A0 for use by other peripherals.

The DMAC requests use of the bus by asserting BR (bus request). The CPU responds when the bus is available by asserting BG (bus grant). The DMAC responds by asserting BGACK (bus grant acknowledge) and OWN, and negating BR. The CPU then negates BG. When the DMAC relinquishes the bus it negates OWN and BGACK.

2.5) SCSI INTERFACE

The SCSI Interface uses the NCR5380 SCSI controller chip. Timing control signals are generated by SMPAL and SCPAL (see figure 5).

Due to the tightly synchronised nature of the DMA control signals from the 5380, a 74ACT646 buffer (F14) must be provided to increase the DMA throughput. Without the buffer, the 68450 would relinquish the bus after every byte transferred because the next request would not be generated by the 5380 (DRQ) until after the completion of the current transfer. The '646 buffer allows the 5380 to get one byte ahead so that DRQ can be asserted during the curren cycle, and the 68450 retain the bus.

When the 5380 is accessed by the CPU the '646 is switched to unbuffered mode and the latched data is not affected.

A timer (A13) disables SCSI DMA requests if the DMAC has not relinquished the bus for approx 30uS. This is to ensure that the CPU is not completely locked out by a high bandwidth disk transfer.

CPU Access:

When SCSI and AS are asserted CS will be asserted on the next falling edge of PCLK. Either IOR or IOW will also be asserted depending on the state of R/W. Assertion of CS causes the '646 to switch to direct (unbuffered) mode, and DBEN is asserted to connect the 5380 data bus to D16-23.

DSACK1 is also asserted at this time. This results in one CPU wait cycle (2 wait states) since DSACK is asserted just after the CPU samples it for the first time.

Timing reference signal STRB is asserted on the next falling edge of PCLK. In the case of a write cycle, this causes IOW to be negated and the 5380 to latch incoming data. On the next falling edge of PCLK the cycle ends with the negation of all control signals (except STRB, but this has no effect).

CPU accesses take priority over DMA transfers between the 5380 and the '646 buffer.

Note that this peripheral is configured as an 8 bit device residing on the low byte of a 16 bit bus, so it appears at consecutive odd addresses to the CPU. This is necessary to interface correctly to the DMA controller.

DMAC Access:

DMA transfers occur as two distinct transactions, between the 5380 and the '646, and the '646 and the DMAC/data bus. The first transaction is initiated by the assertion of DRQ by the 5380, the second by the assertion of SACK by the DMAC. It is easiest to consider read and write as two separate cases.

Read from Disk - Write to Memory (figure 6):

After receiving a byte from the SCSI bus. the 5380 asserts DRQ. This causes the DMA request line SREQ to be asserted. On the next falling edge of PCLK, DACK is asserted along with IOR to perform a DMA read cycle from the 5380. At the same time, the '646 clock line LDAT goes low. 200nS later LDAT goes high, latching the data from the 5380 into the '646. DACK and IOR are negated at this time. The 200nS pulse on DACK is generated using STRB as an intermediate timing signal (as for CPU access).

The assertion of DACK also causes signal REQED to be asserted, which indicates that the data buffer is full and DRQ will be ignored until it has been emptied.

The DMAC will recognise the assertion of SREQ and begin a DMA cycle some time later by asserting SACK. This causes the '646 output enable line DBEN to be asserted, placing the latched data on D16-23. REQED will be negated when SACK is asserted, but SREQ will only be negated if there is no DRQ waiting. If DRQ is asserted, then it will be recognised and DACK asserted as soon as AS is negated by the DMAC, to read the next byte into the latch.

Read from Memory - Write to Disk (figure 7):

To begin a write to disk transfer, the CPU must assert the internal control signal SCDMAW. The rising edge of SCDMAW causes SREQ to be asserted, causing the DMAC to begin transferring the first byte to the buffer. REQED is also asserted, indicating that the buffer is empty and DRQ will be ignored until it is full.

When the DMAC asserts SACK, the '646 clock line LDAT is driven low, and goes high again when the DMAC asserts DTC (data transfer complete), latching the data from D16-23 into the buffer.

REQED will be negated when SACK is asserted, allowing DRQ to be recognised as soon as the DMAC negates AS.

When DRQ is recognised, DACK, DBEN and IOW are asserted on the next falling edge of PCLK, and IOW is negated again 100nS later, causing the data from the buffer to be latched by the 5380. The write cycle is terminated after 200nS using STRB as a timing reference.

Note that a bus error results if the DMAC attempts to read from the SCSI interface when SCDMAW is asserted, or vice versa. SCPAL is also used to generate the DMAC control signals BEC0 and BEC1 from RESET, DHALT and BERR.

2.6) WAVEFORM BUS INTERFACE

The waveform bus is a 3.3MHz synchronous bus with 16 bit data and 24 bit addresses. This bus provides access to waveform RAM by up to eight voice channels as well as the waveform super. Each channel is allocated a time slice every 2.4uS, and the waveform super may use any time slice that is not required by a channel. Refresh requests take priority over waveform super access, but not channel accesses, since refresh will only be completely pre-empted if all channels are running at maximum speed, which is extremely unlikely.

All timing is derived from SCLK, which is a 300nS signal generated by the channel support card. The channel support card also generates a pulse every 2.4uS which is daisy chained through the channels and defines each channel's time slot. If a channel requires its time slot then it asserts TSTAKEN (time slice taken) which is an open collector wired-OR signal, and places its waveform address on the bus.

A waveform memory access is initiated by asserting WAS on the falling edge of SCLK. WUDS and/or WLDS must also be asserted to select the byte(s) to be accessed. A refresh cycle is initiated by asserting WREF only.

Waveform bus control signals WAS, WUDS, WLDS and WREF are generated by WBPAL as follows:

WBPAL is clocked on the falling edge of SCLK. If TSTAKEN is asserted, indicating a channel access, then WAS, WUDS and WLDS are asserted for 300nS. Otherwise, if RFSH is asserted then WREF is asserted for 300nS. Finally, if WFM is asserted by the CPU or DMAC, then WAS is asserted to begin a waveform supervisor access cycle.

In this case, WATB is also asserted to place the address on the bus, and WUDS and/or WLDS, corresponding to UDS and LDS. These are generated by BYPAL during CPU cycles (from A0, SIZ0 and SIZ1), or directly by the DMAC. WR/W is driven low for 100nS (300nS in rev 1.1 and below cards) if the access is a write cycle, otherwise WR/W is pulled high by a 270R pullup.

100nS later, WDTB is asserted to enable the data buffers to/from the bus WDTB is also 300nS long, so the overall length of a waveform bus cycle is 400nS, with 100nS overlap. DSACK1 is asserted 100nS later, indicating to the CPU that the bus width is 16 bits.

This gives a minimum of 2 wait cycles (4 wait states) for a CPU access (overall cycle length of 500nS).

Since there are 24 address lines (x 16 bit) the bus has an effective address range of 32MB. When the 8/32MB waveform RAM card (CMI-43) is in use the address range is extended to 64MB as follows. If A25 is asserted during a Waveform Super access cycle, then WAS is not asserted, but WUDS and/or WLDS still are. Since one of these must always be asserted, the assertion of WAS is actually redundant, and it is interpreted by the 8/32MB waveform RAM as an extra address line.

Note that channel cards can still only access the first 32MB, and also that 2MB and 4MB waveform RAM cards will ignore cycles in which WAS is not asserted, so there is no compatibility problem.

Also note that unmodified 2MB cards interpret WA24 as 8 bit mode select so care must be taken when they are installed in a waveform super system.

The decoding of the WFM select signal includes data strobe (DS) when the cycle is initiated by the DMA controller. This delays the start of DMAC transfers to waveform memory, which improves performance of SCSI transfers by allowing the next DMA request to be generated before the previous cycle ends. Otherwise, the DMAC will relinquish the bus and then re-arbitrate between every byte transferred.

2.7) SAMPLE CARD INTERFACE

The sample card (CMI-337 or CMI-347) is connected to the waveform super via a bi-directional high speed serial link. Sample data is transmitted from the sample card to the waveform super, while control data or sample data is sent from the waveform super back to the sample card. Separate clock and data lines

are optically isolated at the receiving ends, the cable carrying current mode data. Outgoing lines are driven by a constant current source (approx 7-8 mA) consisting of a CMOS gate driving a 470R resistor into a LM394 current mirror. This setup allows the HP2631 opto-isolators to operate at up to 10Mbits/s.

The clock is generated by the sample card and used for transmission in both directions. A start conversion pulse is generated by channel card 0 and transmitted to the sample card by the waveform super (ADSTART); this signal is not used by the waveform super, and in digital or crystal locked mode is ignored by the CMI-347.

Both data channels are 17 bits wide. For incoming sample data, the 17th bit is used as a left/right channel select (0=left, 1=right) for stereo sampling. For control data, the 17th bit is intended to be used as a control/data select (0=control, 1=data), although data out is not currently supported by the sample card. See figure 8 for data transmission format.

Data transmission begins when EOC (end of conversion) is negated. At this time, the control data register (F1, F5) is latched into the data out shift register. Incoming sample data is clocked into the 17 bit shift register consisting of F2, F4 and F6 on each rising clock edge. At the same time, control data is clocked out of the 16 bit shift register consisting of F1 and F5. After 16 clocks, the 17th control data bit (set by the internal control latch bit ADB16) is present on the serial out line. The assertion edge of EOC is not used, due to timing requirements of the old (CMI-337) sample card. Thus, the incoming sample data is not latched until the *following* negation of EOC (ie, the beginning of the next sample transmission). At this time, sample data is latched into the holding register and the DATA READY bit (ADRDY) is set in the status register (ADSTAT).

If DMA mode is enabled, then a DMA request is also generated at this time on the appropriate channel as selected by the 17th bit of the incoming data. The value of this bit is also accessible to the CPU in the AD status register.

If the ADC interrupt enable internal control bit is set then a level 6 interrupt will be presented to the CPU whenever the READY status bit is set.

The READY status bit (D2 pin 6) is cleared at the end of a CPU READ from the AD DATA register, unless DMA mode is enabled. If DMA mode is enabled then the READY bit is cleared when the DMAC asserts DTC at the end of a READ OR WRITE cycle. Thus the CPU may read the AD DATA register during DMA transfers without clearing the READY bit (and thus DMA REQ).

If the READY bit has not been cleared by the next active EOC edge then the overrun status bit (D2 pin 9) is set. This occurs in either CPU or DMA mode. The overrun bit remains set until cleared by a CPU write to the status register address.

DMA requests remain asserted until the beginning of the next DMA cycle (LACK or RACK asserted). Thus the DMA controller may be configured in either BURST mode or CYCLE STEAL mode (BURST mode is suggested).

Control and status signals are generated by ADPAL (see figures 9 and 10).

Note that bit 15 of incoming sample data is inverted by the hardware to convert to twos-complement notation.

2.8) CMI BUS INTERFACE

Access to the CMI bus is provided via separate 64k windows for P1 and P2 DMA cycles. Word accesses by the CPU or DMAC are automatically split into two byte cycles, with the lowest address (A0=0) transferred first. In the case of write cycles, data and address are latched by the hardware and the cycle proceeds concurrently with further internal bus cycles.

Since the 6809 bus runs asynchronously with the waveform super clock requests must be synchronised by latching the CMI bus request line (CMI) on the rising edge of RA. The latched signal LCMI is sampled by C2PAL 50nS later when RAS is asserted, causing CYCLE to be asserted. The rising edge of CYCLE latches the address, R/W, UDS and LDS and A18, which selects whether a P1 or P2 DMA cycle is to be generated. LDATL and LDATH are also asserted to latch the

data (although this is redundant in READ cycles), and in the case of a WRITE cycle, DSACK1 is asserted, terminating the internal bus cycle. When the internal cycle terminates, LOCK is asserted to prevent any further CMI bus accesses being accepted until the WRITE is complete.

Multiplexor B3 switches the processor dependent 6809 bus control signals (ADDx, PH2x, ACKx, ETLx) to select the appropriate processor according to latched A18 (LP2).

When CYCLE is asserted the waveform super will begin arbitration for the CMI bus. The daisy chain consists of signals ETL (enable this level) and ENL (enable next level). Assertion of ETL indicates that no device higher on the daisy chain is requesting DMA. The first device on the daisy chain receives ETL for 1uS out of every 2uS (this is to ensure that the 6809 CPU is not locked out completely). If the WS does not require DMA then ETL is passed onto the next device as ENL. If a request is pending, then RDMA is asserted and ENL is not. The Q209 CPU card samples RDMA on the falling edge of PH2, and asserts ACK to grant a DMA cycle.

The rising edge of ACK clocks ETL, causing LETL to be asserted when a DMA cycle is granted to the waveform super. This signals the beginning of the active part of the DMA cycle.

DMAC (DMA claim) is asserted for 350nS to select the appropriate memory map on the Q256/Q356 system RAM cards. When ADDx is asserted, ATB is asserted, causing the 6809 address to be placed on the address bus. The appropriate value for A0 is generated by C2PAL taking into account the values of UDS and LDS, and whether this is the first or second cycle of a word transfer.

Data buffer enable (DBENH/DBENL) is asserted when PH2 is asserted. This places the WRITE data onto the CMI bus (redundant in READ cycles). In the case of READ cycles, the appropriate LDAT goes low at the same time, and goes high to latch the CMI bus data on the falling edge of RAS.

Byte selection is performed by UDS and LDS, which are generated by the DMAC, or by BYPAL during CPU cycles. If both LDS and UDS are asserted then two DMA cycles are generated. LA0 is initially zero, and is set to one as soon as ATB is negated after the first byte transfer.

Completion of the bus access is indicated by the assertion of LSTBYT, which occurs during the data phase of the last byte transfer. When LSTBYT is asserted, CYCLE will be negated on the next falling edge of PH2. In the case of a READ cycle, DSACK1 will be asserted on the falling edge of RAS, and both DBENs will be asserted and remain asserted (placing the latched data from the CMI bus onto the internal bus) until the end of the internal bus cycle is indicated by the negation of CMI. See figure 11 for timing diagram.

Address decoding is arranged in such a way that 4 aliases of each 64k window are contiguous. The nominal base address of the CMI bus interface is the centre of this 256k region. This allows addressing modes with 16 bit offsets or indices, which will be sign extended to 32 bits, to be used. In addition, block transfers may wrap around the end of the 64k space without causing problems.

Tming:

The time required for an access to the CMI bus will depend on

1) the phase of the CMI bus at the time the bus cycle starts
2) the presence of higher priority DMA requests
3) whether the cycle is a READ or WRITE cycle
4) the number of bytes being transferred
5) whether a previous WRITE cycle is still in progress.

The best case timing occurs when CMI is asserted just before the rising edge of RA, at the same time as ETL is being asserted for the appropriate 6809 processor.

Best case timing for a byte READ cycle is 1.6uS, worst case 3.7uS (resulting from a CMI bus synchronisation delay of 2uS and a DSACK synchronisation delay of 100nS). For a word transfer add 2uS. For each byte transferred by a higher priority device add 2uS.

For WRITE cycles, best case is 400nS, worst case 1uS. The maximum synchronisation delay to the 68020 is only 600nS, since CMI is sampled every 500nS; however, the CMI bus cycle may not start for another 1500nS. The actual transfer will then take the same amount of time as for an equivalent READ cycle.

In either case, if a previous WRITE cycle is still in progress, the new access will be delayed until the WRITE is complete, and will then proceed with a synchronisation delay of 500nS.

2.9) INTERNAL CONTROL LATCH

An addressable latch is provided for control of the CMI-41 from the CMI bus. This latch appears in the peripheral region of the CMI bus at address $FC5C (same address as CMI-33). Eight control bits may be individually set or cleared. The bit number is selected by D1-3, and the data written is in D0. Since all of the control lines are active low D0 must be zero to assert the signal.

bit number	function	value to ASSERT	value to NEGATE
0	interrupt level 1	$00	$01
1	interrupt level 3	$02	$03
2	interrupt level 5	$04	$05
3	interrupt level 7 (NMI)	$06	$07
4	cache disable	$08	$09
5	DMAC halt	$0A	$0B
6	CPU halt	$0C	$0D
7	RESET	$0E	$0F

Control signals are generated by CLPAL for the internal control latch, and also for the CMI bus data buffer.

The address lines, including FCXX and R/W, are sampled for 50nS after RA goes high, until RAS goes low. The result is latched, and the LS138 addressable latch is enabled during CAS. Data must be stable during this time, since the LS138 has a transparent enable.

The direction of the LS640 data bus buffer is normally to read data from the CMI bus. When ATB (from C1PAL) is asserted and R/W indicates a WRITE cycle, the buffer is reversed to allow the WS to place its data onto the CMI bus.

2.10) INTERNAL CLOCK

The system clock for all subsystems except the FPU is a 10MHz clock, normally received from the channel support card. A separate clock is provided for the FPU so that a higher speed part can be used (normally 16MHz). This is generated by an on board 32MHz crystal oscillator.

In addition, link options are provided to allow the CMI-41 to be used without a channel support card. In this case, the on board clock is used for all subsystems, and must be limited to 10MHz.

It is possible in this case to use a 20MHz FPU by fitting a 40MHz oscillator and dividing it twice for the CPU etc. However, due to an error in the rev1.0 artwork, this would require a couple of signals to be moved.

Figure 1 - CMI-41 Internal Bus Devices

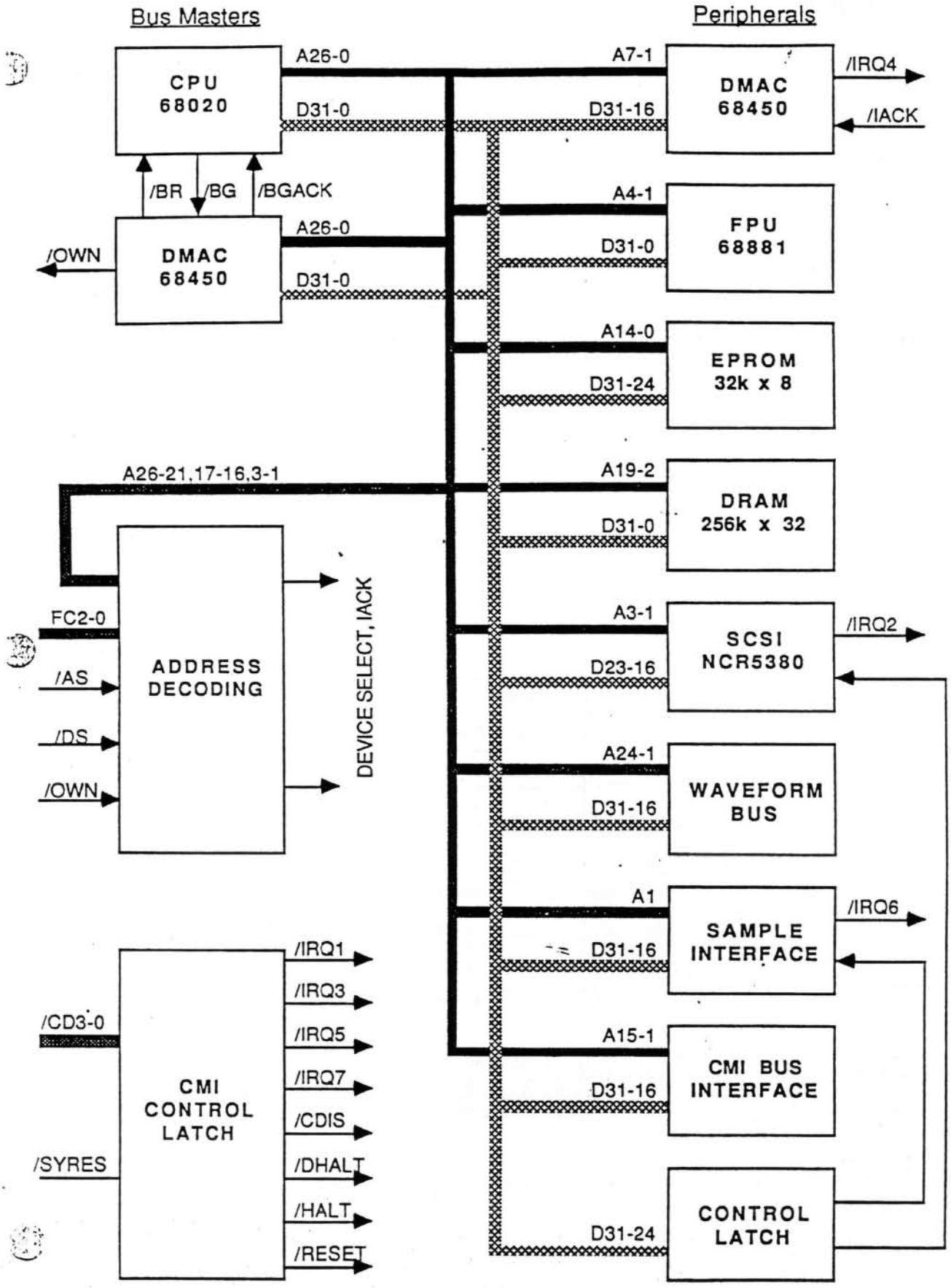

Figure 2 - CMI-41 Waveform Supervisor - Memory Map