Q209

DUAL PROCESSOR

fairlight

FIELD CHANGE NOTICE

DATE 22/ 10/ 92
NUMBER 110

ORIGINATOR Chris Alfred

PRODUCT: CMI / MFX

ASSEMBLY No. Q209 **DESCRIPTION** Dual Processor Card

This FCN applies to REV No: Rev 8 **The New REV No is:** Rev 8.1

REASON FOR CHANGE:

New pcb layout Logic error on pcb design

DETAILS OF CHANGE:

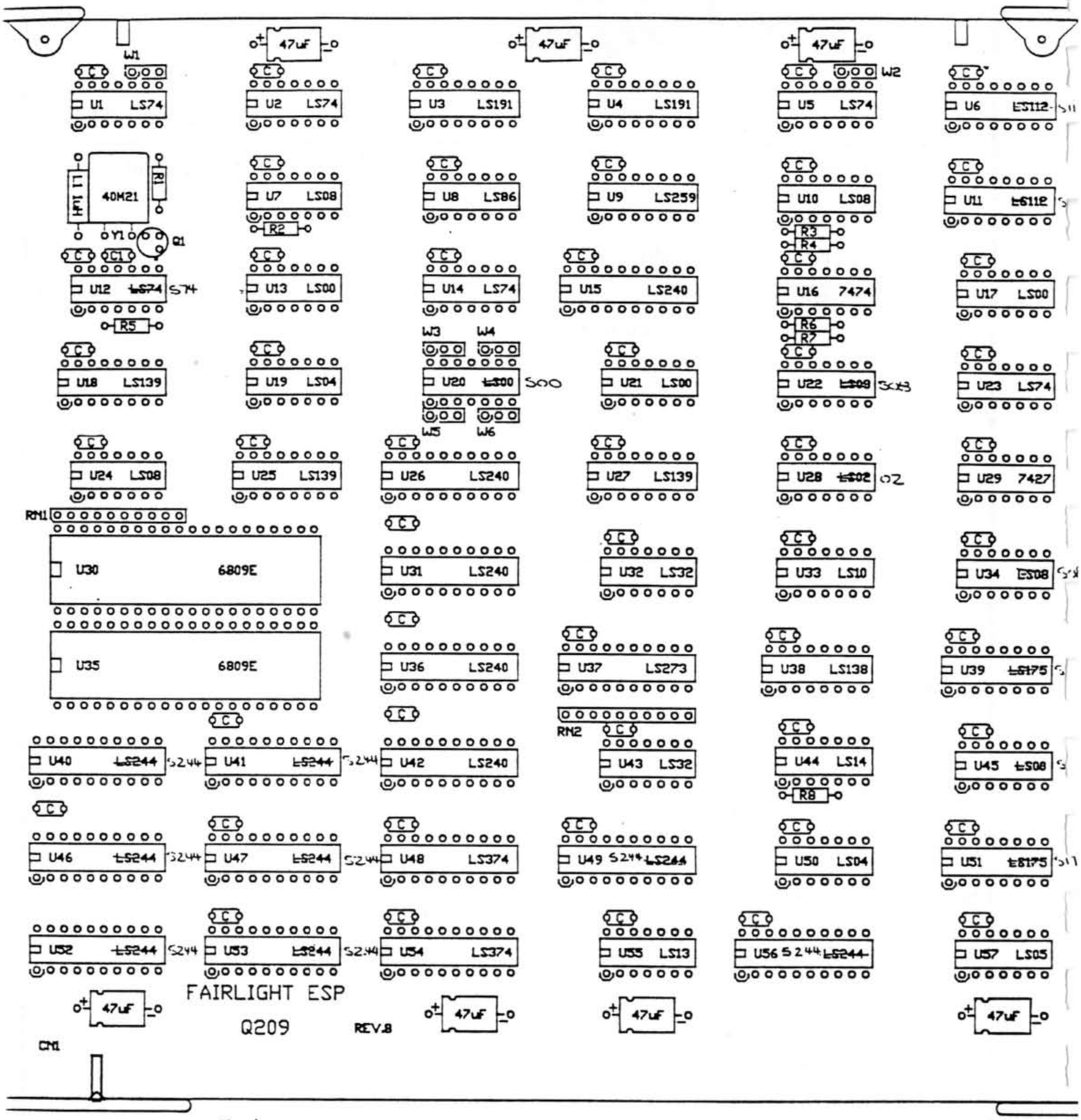
1. Cut component side track to U15 pin 1 (74S240)
2. Connect U22 pin 8 (74S08) to U19 pin 5 (74LS04)
3. Connect U19 pin 6 (74LS04) to U15 pin 1 (74S240)
4. Mark board as Rev 8.1

See track layout attached

Must be done in production of new cards

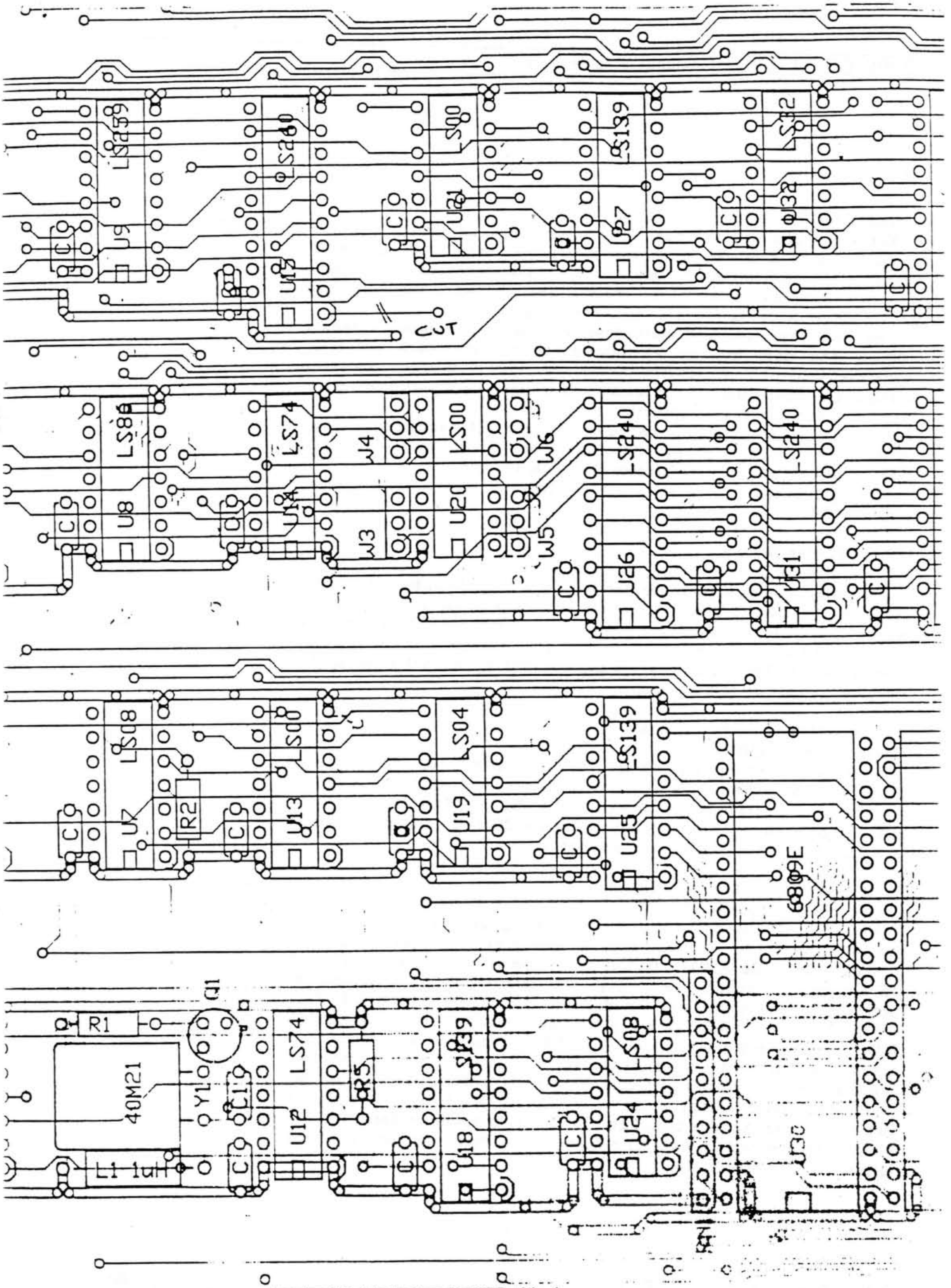
ORIGINATOR: C F Alfred	**DATE:** 25-10-1992	**PROD:** Mario Pedus	**DATE:** 26/10/92 25-10-1992
SERVICE:	**DATE:**	**KIT LIST CHANGE:**	**YES** (**NO**)

Q209PC | SHEET 6 OF 6
artwork PCB DESIGN BUREAU PH. (617) 07-203 0634



* Note: U12, U20, U22, U28, U34, U39, U40, U41, U45, U46, U47, U49, U51, U52, U53, U54, U56 ARE "S" Series NOT "LS" Series

FAIRLIGHT ESP		ISSUE 1
		DATE 27.08.1992
- OVERLAY	TITLE Q209	REV 8
Q209PC	NUMBER -	SHEET 1 OF 6
artwork	PCB DESIGN BUREAU	PH. (617) 07-203 0634



component side view

introduction

The Q209 contains the dual 6809 processors , on board processor communication hardware entailing indivisible instructions, processor readable identification / map state , interprocessor interrupts, automatic map switching FUSE register and hardware trace logic to enable single stepping for software debugging.

The Dual Processor card multiplexes each processor onto a common address and data buss in an interleaved manner, each processor therefore may simultaneously access the same memory location without any contention, if the memory is mapped onto both processors. (See Q256 functional description)

The memory addresses are issued to the buss 225 nanoseconds prior to the access cycle, allowing addresses to be mapped by the memory card, to allow for accessing greater than 64K of RAM.

Many global timing signals are issued from the processor for general buss control .

Master Timing Signals

(refer schematic Q209-00)

All system timing signals are derived from crystal-controlled 40MHz oscillator Q1. Flip-flop 10F derives two opposite phase 20 MHz square waves. Quad D-type latch D2, together with the NAND gate in 7F, forms a 10 state Johnson, or twisted-tail ring counter. Each state is of 50ns duration. The system signals are decoded by NAND gates in 8E from the output of this counter.

Dynamic Memory Timing Signals

(refer schematic Q209-00)

Four non-inverting buffers of 10A are driven by latch E1 to provide CAS(Column Address Strobe), RAS (Row Address Strobe), CA (Column Address , active low) and RA (Row Address, active low). RAS is delayed relative to CAS by about 20ns by the propagation delay of 11E. RA and CA are complementary.

Data and Address Buss Multiplexing

(refer schematic Q209-03)

Flip-flop 6F , along with associated gating, generates the 6809's E signals. The system address buss is multiplexed by the ADDRESS signals ADD1 and ADD2 , (active low). One-of-four decoders 3E and 4E are used to enable the appropriate address and data buffers, to perform the multiplexing. The data buffer enable signals WRITE1, WRITE2, READ1, READ2 are generated by logical combinations of R/W , VMA , processor phase 2 and DMA lines. The address buss is actually multiplexed 4 ways, as the vectored interupt system may also acquire the buss' least significant bits of the address buss for either processor's vector fetch cycle. The address buffer enables are a function of the Address signal and the Interupt acknowledge.

Phase 2 reference and Address references for each Processor are feed to the buss via buss drivers.

Interrupt Strobe Generation
(refer schematic Q209-02)
Dual D-type flip-flop 9D and 3-input AND gate 8D feed Interrupt Strobe pulses to the buss. These are used by the Priority Interrupt Control Units (PICUs) used to provide vectored interrupts, and also to strobe the vector address latches 8A and 9A. The PICUs are located on the Q133 card. These signals strobe the priority latches continuously, until an interrupt is acknowledged. In this way the Interrupt Priority is maintained at its latest level regardless of delay between an interrupt request being received by the PICU and the associated vector-fetch cycle being executed.

Direct Memory Access
(refer to drawing Q209-00)
DMA requests for each processor are clocked into flip-flop 11D on the falling edge of the phase 2 signal of the respective processor. DMA acknowledge is sent to the buss via buffers and drive signals to the processors are suspended in the phase 1 state for the duration of the DMA cycle. The maximum permissible DMA duration is 5 microseconds. Worst-case DMA latency is 1 microsecond. Latency is the time required to service the request.

Vector-Fetch Decoders
(refer to drawing Q209-01)
The vector state of the processors are decoded by the one-of-four decoders, 2D and NOR gates in 1D. These correspond to addresses in the range FFF0 to FFFF. They correspond to the processor fetching vectors FIRQ, NMI, SWI1, SWI2, SWI3, IRQ and RESTART. The Restart vectors come from ROM so when this is sensed the ROM is enabled and the ram disabled. This is achieved by the ROMEN signal on buss pin 44. On detection of an Interrupt Request vector address from the processor, decoder 2D causes the normal address buss drivers for bits 1 to 4 to be disabled and the Interrupt Address buffers to be enabled in lieu.

Processor System Control
These general functions are controlled through ports at the following locations ...

$FC5E	Indivisible instructions	read
$FC5E	Various CPU functions	write
$FC5F	Map status and CPU ID	read
$FC5F	Automatic map switching FUSE	write

The ID can be read to determine the status of the memory map switching hardware. The CPU ID bit can be read by the CPU to find out which CPU is running the program.
The bits are defined as

D0	CPU ID 0=P1 1=P2
D1	P1 map status 0=map B, 1=map A
D2	P2 map status
D3	zero
D4	n/c (indeterminate)
D5	n/c
D6	n/c
D7	n/c

The "Various CPU functions" is an 8 bit register in which each bit may be independently written to. This register is at location 6D, and it is decoded by devices at 9B and 7A. When written to, the bit address is selected by the 3 least significant bits of the data byte. The state of data bit 3 determines whether the bit is set or cleared.

The four functions provided per processor from this register are:-

0+P interprocessor interrupt
2+P hardware trace
4+P map switch select
6+P fast interrupt request

where P is "0" for processor 1 and "1" for processor 2

Fast interrupts may be generated either by an external signal or from the bus. The on-card FIRQ must be reset by the processor concerned, by writing a reset bit to the register.

Automatic Map Switching

(refer schematic Q209-02)

The memory cards support hardware selectable memory maps. The processors can control the A/$\overline{B}$ select lines, allowing automatic switching between "user(B)" and "system(A)" maps.

Whenever an interrupt or processor restart occurs , the A map will be automatically selected. During an interrupt, the switching will occur after the registers are stacked and before the interrupt vector is fetched.

A FUSE location is provided which causes the map to be switched after a specified number of CPU clock cycles have elapsed. These counters are at 8C for CPU1 and 9C for CPU2. The data written to these counters is buffered by the buffer at 7C. The map changed to is determined by the value of the map switch select bit. The map switch will occur after the Nth CPU cycle after the FUSE register write. The delay is required so that a known number of instructions can be executed for house keeping before the CPU's memory is swapped.

Hardware also selects the A map whenever DMA occurs.

Hardware Trace

An NMI may be generated after each instruction execution for software debugging. This is done by flip-flops 5D and half of AND gate 4F. This function is enabled under software, by access to $FC5E.

Enabling this function inverts the NMI signal from the front panel, so that if the trace hardware is left in the triggered state, front panel NMI requests will still be recognized, but on the opposite edge (since NMI is an edge triggered input).

Indivisible Instructions

(refer to schematic Q209-00)

For the test-and-set and double byte load/store instructions to be effective, the processor not executing the instruction must not be able to alter the flag memory location in question. To this end, the execution of these read/modify/write instructions effectively hangs the other processor for its duration, thus preventing race conditions .

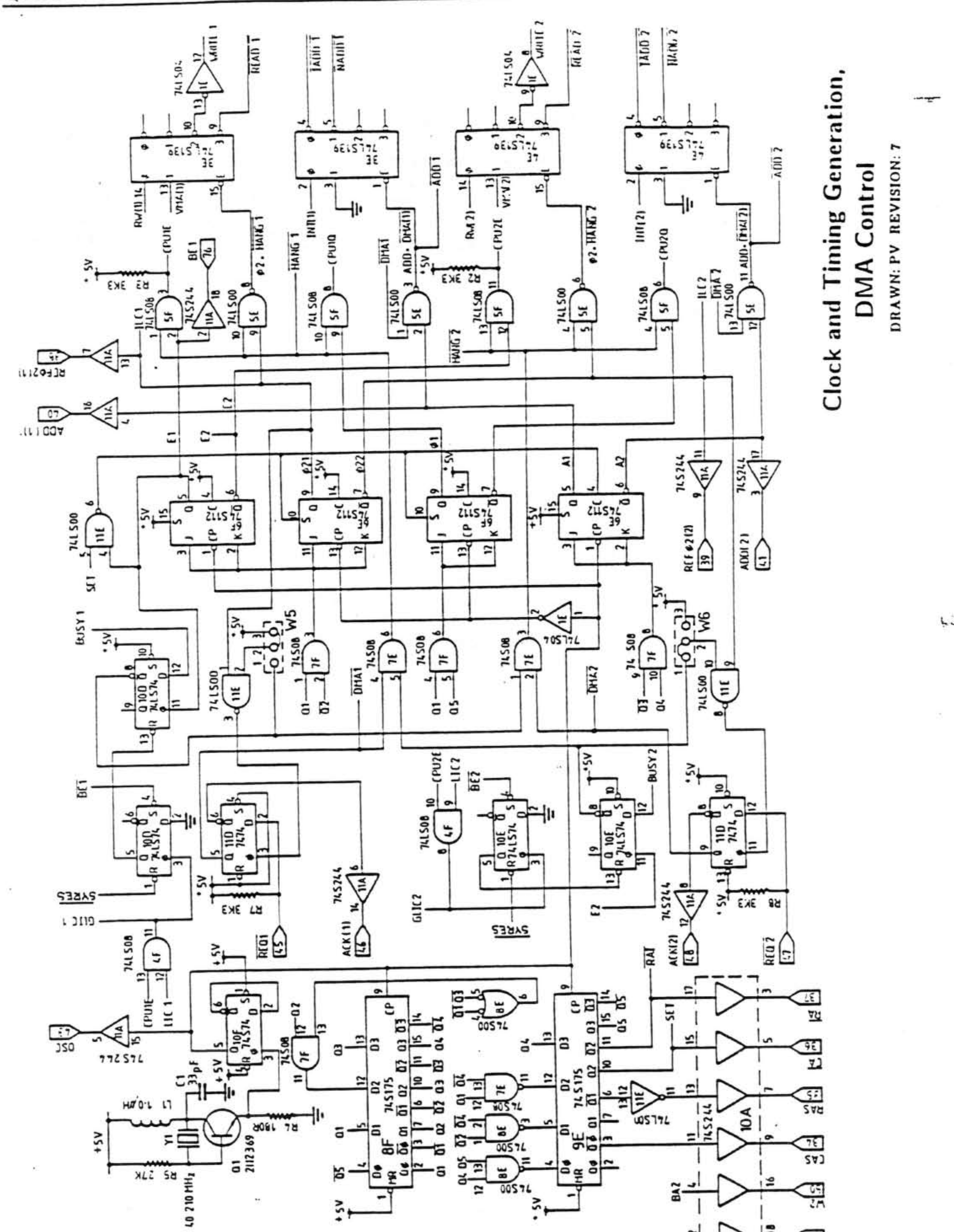
This is achieved by flip-flops 10D, 10E, associated gating and the BUSY outputs from the processors. The BUSY outputs are active when the test and set instruction is executed , and the other processors clock is stopped for its cycle, in the same manner as for DMA transfers. The flip-flops asociated with this are reset at power on to enable the clocks to the 6809's to allow them to be internally reset, at power on reset. To enable this function the instruction to be made indivisible must be immediately preceded by a read from hardware location $FC5E. No interrupt must be allowed to occur between the read and the instruction. This function is automatically disabled at the end of the instruction following the read.

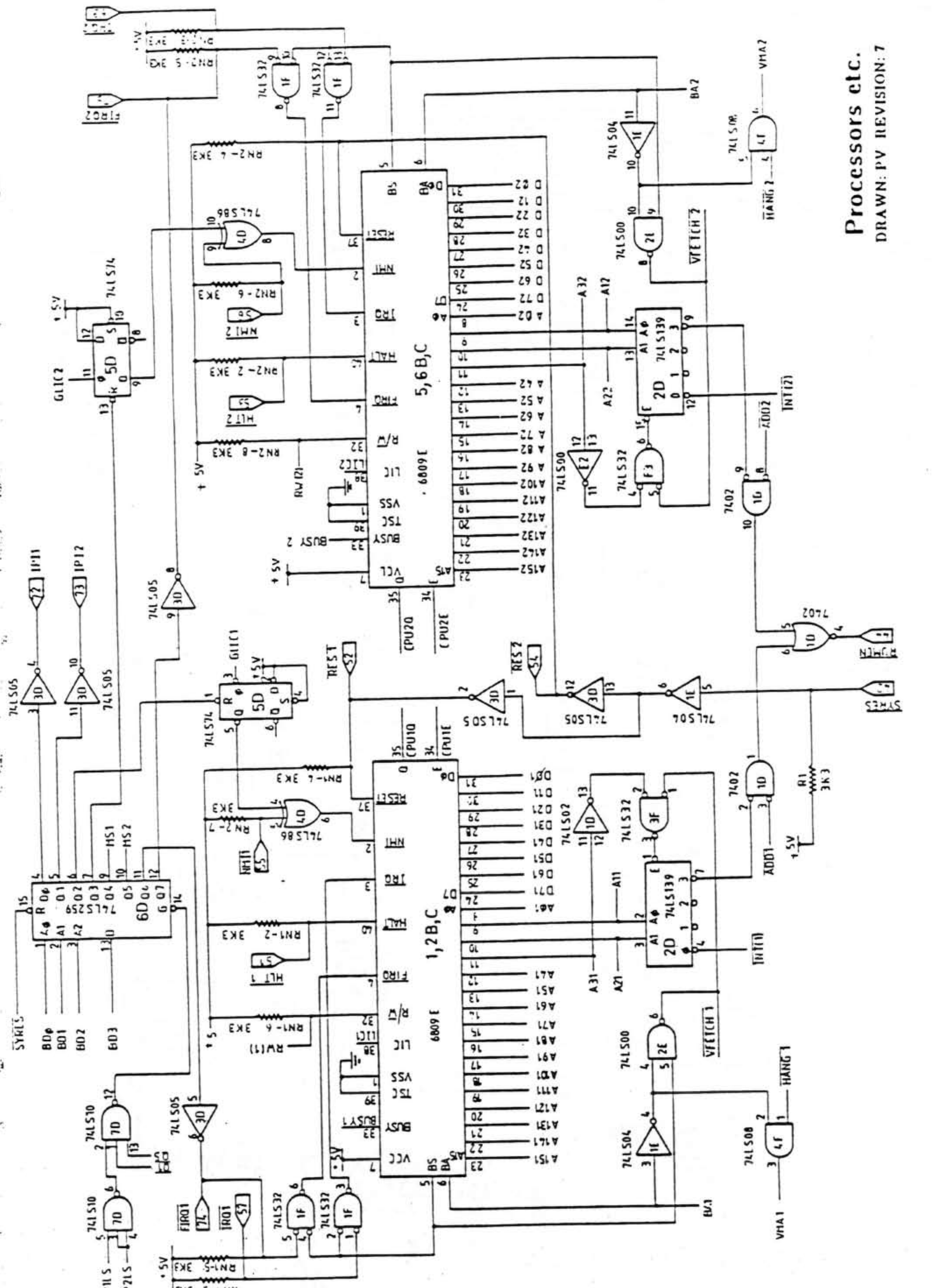
Link Options

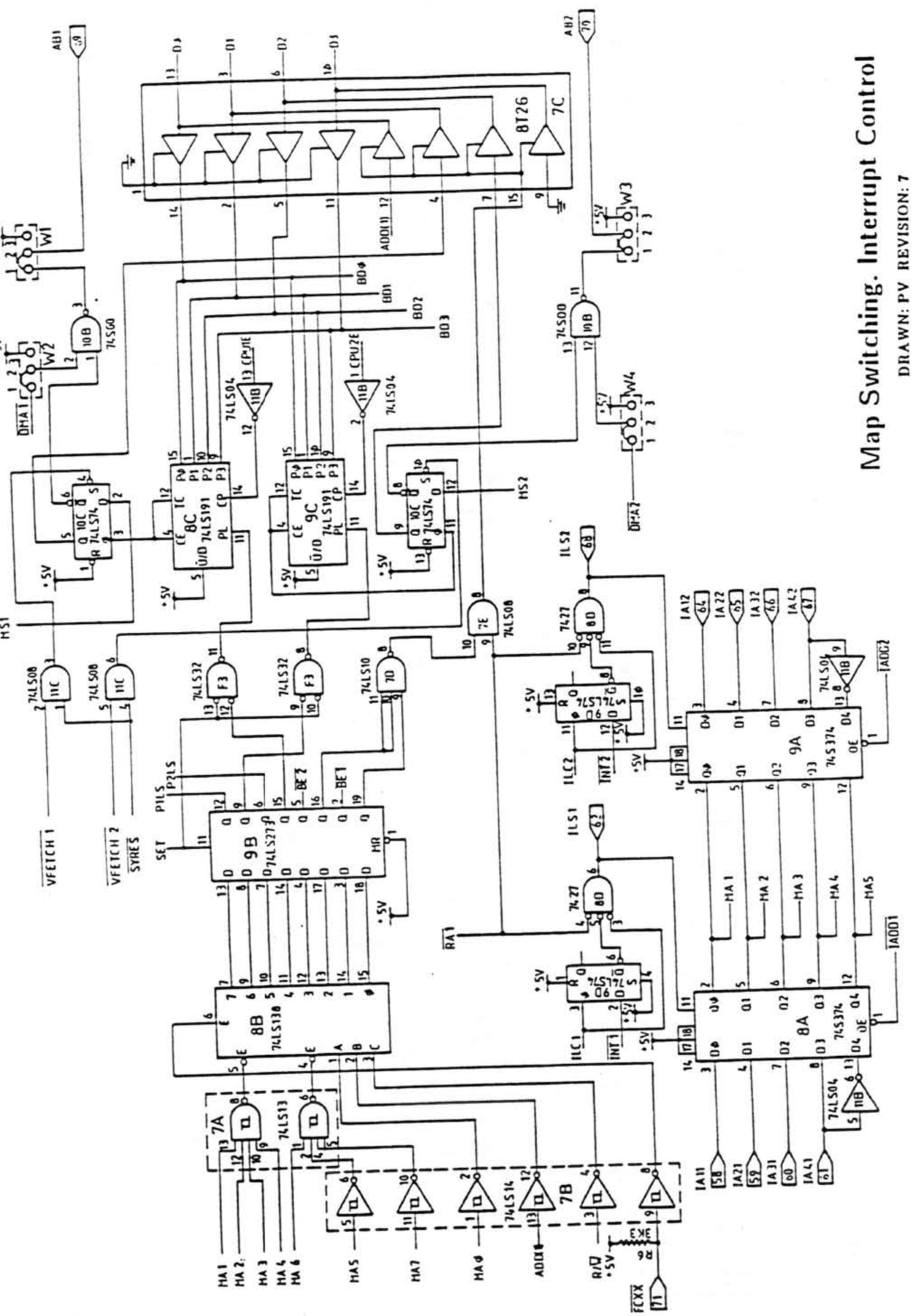
The links have the following functions:

Option	CPU	LINK	Function
W1	P1	1-2 *	enable map select output
		2-3	disable
W2	P1	1-2 *	enable DMA to select A map
		2-3	disable
W3	P2	1-2 *	enable map select output
		2-3	disable
W4	P2	1-2 *	enable DMA to select A map
		2-3	disable
W5	P1	1-2	disable P1 DMA during indivisible P2 cycles
		2-3 *	enable
W6	P2	1-2 *	disable P2 DMA during indivisible P1 cycles
		2-3	enable

These links are set by PCB traces in the positions marked by * . Links

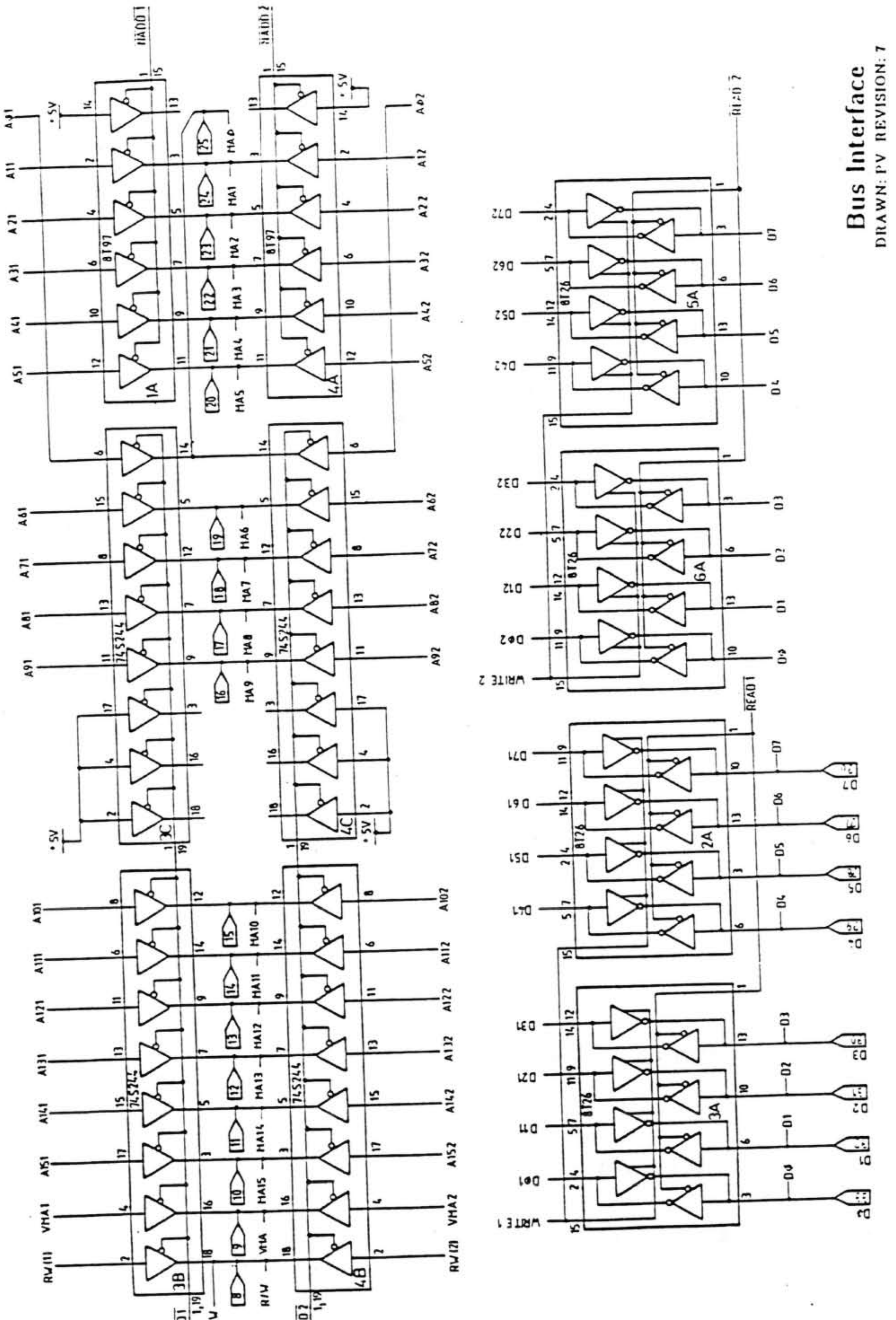

Clock and Timing Generation,
DMA Control
DRAWN: PV REVISION: 7


Processors etc.
DRAWN: PV REVISION: 7
5,6B,C
1,2B,C



Map Switching. Interrupt Control

DRAWN: PV REVISION: 7


Bus Interface
DRAWN: PV REVISION: 7

{xx{xx}{x

Q209 TESTING

Document Revision: 1.0 6.1.1993 CEA

Fairlight ESP Pty. Ltd.
30 Bay Street,
Broadway,
Sydney Australia 2007.

Q209 Testing
Fairlight ESP

1. Introduction

To read memory locations accessable by P1, a monochrome monitor must be connected to a Q050 debug TVT card. P1 NMI should be enabled on the front panel and P2 NMI disabled. The NMI button is then pressed to enter the monitor. The monitor prompt (!) will appear on the monochrome screen.

To read memory accessable by P2, enable P2 NMI and disable P1 NMI and press the NMI button. The monitor prompt (:) will appear on the main graphics screen.

The locations can also be observed by connecting the In-Circuit Emulator to the appropriate processor socket.

All IC references are for revision 8 and 9 cards.

2. TESTING

2.1 Test and Set hardware

```
NMI P1
read $FC5E
     asserts *BE1 for 500nS (U5/4 74LS74)
     holds CPU2E and CPU2Q low while BUSY1 asserted (U5/3 74LS74)
     by asserting *HANG2 (U10/13 74LS08)
     BUSY signal is asserted during read/modify/write
     type instructions.

NMIP2
read $FC5E
     asserts *BE2 for 500nS (U1/4 74LS74)
     holds CPU1E and CPU1Q low while BUSY2 asserted (U1/3 74LS74)
     by asserting *HANG1 (U10/13 74LS08)
```

2.2 Control Latch (U9 74LS259)

```
-  NMI P1 or P2
-  write to $FC5E
      Bits 0..2     select control bit
      Bit  3        bit state
      This may cause the system to do strange things, but
      the control bits should change.
```

2.3 Status (U15A 74LS240)

```
-  NMI P1
   read $FC5F     (should get $X6)
        The bit 0 of the status is the processor ID (0 = P1).

-  NMI P2
   read $FC5F     (should get $X7)
```

2.4 Map selection and FUSE

- NMI P1
  Any interrupt causes the map to change to A.
  read $FC5F (should be $X6)
  write $0F to $FC5F
  Writes $0F to FUSE latch. After $0F instructions the FUSE latch counts to 0 and resets map to value of MSx control bit (U9 pins 9,10).
  read $FC5F (should be $X4)
  Status bit 1 is P1 map status (0 = map B).

- NMI P2
  read $FC5F (should be $X7)
  write $0F to $FC5F
  read $FC5F (should be $X3)

2.5 Interrupts

Tested via DBTST subtest INT.

2.6 DMA

P1 DMA is used every 16uS by refresh.

P2 DMA can be exercised by causing QFC9 to upload its ROM
write $0A to $FCE1 (see QFC9 documentation).

3. Notes

- The system cannot boot from floppy without P2 DMA working.
- Once OS9 is booted,all the Q209 functions MUST be working.
- Printing of the initial signon message 'BOOT ROM ...' does not require any of the above features to be operational. However, the system may soon crash if they are faulty.

BOOT SEQUENCE FOR P1/P2

Document Revision:	1.0	4.1.1993	CEA	
	1.1	6.1.1992	CEA	corrections

Fairlight ESP Pty. Ltd.
30 Bay Street,
Broadway,
Sydney Australia 2007.

1. Introduction

This document describes to sequence of events upon reset of the CMI. The document is valid for Q133 debug card Revision 17 ROMs.

To read memory locations accessable by P1, a monochrome monitor must be connected to a Q050 debug TVT card. P1 NMI should be enabled on the front panel and P2 NMI disabled. The NMI button is then pressed to enter the monitor. The monitor prompt (!) will appear on the monochrome screen.

To read memory accessable by P2, enable P2 NMI and disable P1 NMI and press the NMI button. The monitor prompt (:) will appear on the main graphics screen.

The locations can also be observed by connecting the In-Circuit Emulator to the appropriate processor socket.

All IC references are for the newest revision of cards:

Q209 Rev 8,9
Q133 Rev 8,9

2. P1 RESET SEQUENCE

2.1 Reset vectoring

- Power on reset or front panel reset asserts *SYRES (Q133 U21 555) for approx 500mS to reset system.
  This asserts *RES1 to processor P1 (Q209 U38 6809E).

- P1 does vector fetch from $FFFE-$FFFF to get reset address.
  Vector fetch asserts *VFETCH1 which causes *ROMEN assertion to map F8LMRK17 rom to $FF00-$FFFF.
  P1 jumps to address at $FFFE-$FFFF ($F819).
  reset vector for processor can be seen at address $FBFE-$FBFF in monitor.

2.2 P1 Initialisation

- disable interrupts
  interrupts remain disabled throughout boot process

  load stack pointer to $FF40
  clear TRIG ($FE00)
  TRIG is set to non-zero by P2 when P2 wants P1 to execute special functions.

  clear IOFLAG ($FE17)
  IOFLAG is set non-zero if IOPACK is running

  clear P1 interrupt controller (8214 U48 Q133 at $FCFC)
  clear P1 control bits in CPU control latch (74LS259 U9 Q209 at $FE5E - bits 6,4,2,0).

- Do dummy sofware interrupt to set P1 to A memory map via the assertion of *VFETCH1 which sets A/B map select latch (74LS74 U14A Q133).
  **The B map is used during normal program operation.
  The A map is used during interrupts. Once the A map is selected it is only reset to B when the FUSE latch counter reaches 0 (74LS191 U3,U4 Q133).**
  (as the FUSE latch is not set, P1 remains in map A).

- initialise the Q356 MSELRAM memory mapping for P1 and P2
  P1 maps ($FC40-$FC47) set to $21 (map 30)
  P2 maps ($FC48-$FC4F) set to $20 (map 31)

  Set all maps for sequential blocks $0000-$FFFF
  (P1 and P2 share the one 64K memory range)

- clear Q356 parity errors by reading parity latchs (74LS374 7A Q356) for 8 cards ($FC40-$FC47).
  There can be up to 8 Q256 cards installed in the system. Which is equivalent to 2 Q356 cards.

  check PWRFLAG ($FE3A). If not $AA then clear all of the Q356 memory and then set PWRFLAG=$AA.
  PWRFLG is used to detect whether this is a power-up reset (PWRFLAG not $AA).

- set P2 end of memory ENDRAM ($FE65) to $E000
  set GRFLAG ($FE68) to $52 ('G') to default to Q219 graphics when booting from QDOS.

2.3 Reset Peripherals

- Reset PIAs (6821)
  - Real time clock PIA (U8 Q133)
    - PA0..3 outputs = $F
    - PA4..7 inputs
    - PB0..7 outputs = $82
  - Debug PIA (U2 Q133)
    - PA0..7 inputs
    - PB0..7 inputs
    - **These are later changed by the WS to initialise the router.**
  - Monochrome Graphics PIA (DE4,5 Q219)
    - PA0..7 outputs = $00
    - PB0..7 mixed = $1A
    - **These are changed later by P1 if CG2 or CG3 installed.**

- Reset timers (6840)
  - Monochrome Graphics (F4,5 Q219).
  - System timer (U20 Q133).

- Reset Colour Graphics card

  The Monochrome Graphics timer ($FCC8) is checked by writing the complement of the data read at that location. If correct value returned, the card is assumed to be Q219.

  Memory locations which do not have devices installed will return $FF as the inverted data buss is pulled down by the termination.

  Enable the colour graphics card by asserting *CGEN (U8/39 CG2 or CG3)

  Check whether low resolution colour graphics (CG1)

control latch	($FCC6) = $1A	(74LS374 U39 CG2,3)
colour latch	($FCCA) = $00	(74LS374 U40 CG2,3)
erase latch	($FCCB) = $00	(74LS374 U85 CG2,3)

  write $AA to screen using byte access
  if read-back value is non-zero then card is lo-res as CG1 card does not have erase latch to clear the value read.

  Set colour latch = $00
  Set erase latch = $FF

  Set control latch
  - CG1: $1A
  - CG2,3: $1B (hi-res enabled)
  - and copy image of value to VRIMAGE ($FE90)

  Set graphics card type flag GFX.TYPE ($FE96)
  - MONO: $4E (M)

```
            CG1:      $43  (C)
            CG2,3:    $44  (D)

        Clear background colour image ($FE97)
        Clear foreground colour image ($FE05)
```

- Initialise system timer to generate 50Hz from timer 3.

2.4 Reset I/O system

- initialise P1 TVT control flags
- clear horizontal scroll register (6821 port B U8 CG2,3)
- initialise INTFLG ($FE1E) = 1 so P1 does all keyboard input.
- reset ACIAs (6551)

keyboard	(U19 Q133)	9600,1 stop bit ,no parity
printer0	(U15 Q133)	9600,2 stop bits,no parity
mfx	(U18 Q133)	19200,2 stop bits,no parity
comms	(U16 Q133)	9600,1 stop bit ,no parity

  The comms port connects to P1 on the Q137 front panel board.

- reset keyboard input queue
  KFERPT ($FE19) = 0
  KFERPT is set to the key received by P1 for P2 to read the value. P2 clears KFERPT when it has read the key.
- initilise monitor variables
  clear the breakpoint table ($FFA0-$FFA9)
- initialise P1 service routine to default service loop
  SERVAD ($FE09) = $F8FE

2.5 P2 command request loop

- check TRIG ($FE00)
  if TRIG is zero execute the default service loop
  otherwise do function requested by P2
  For P2 to request P1 to execute special commands, P2 writes the address of the command to JUMPAD ($FE01) and sets TRIG ($FE00) to non-zero.
  Upon P1 completion of the command, P1 clears TRIG and loops waiting for TRIG to be set and executes the default service loop while waiting.

2.6 Default Service Loop

- Maintain system timers TIMER0..3 ($FE34..7)
  decremented every 50Hz from system timer if non-zero
- Output characters for P2
  For P2 to display character to the screen, P2 writes the character to VFERPT ($FE1D). If P1 finds this location

non-zero, P1 displays the character to screen. Once the character has been accepted, P1 clears VFERPT.

- Input characters from keyboard
  If INTFLG ($FE1E) is zero, then P2 is handling keyboard input (e.g. when OS9 is running) and P1 ignores the keyboard.
  If INTFLG is non-zero (as it is on startup) P1 places any keys received at KFERPT ($FE19) ready for P2 to accept. P2 reads the key from KFERPT and then clears KFERPT.

3. P2 RESET SEQUENCE

3.1 Reset Vectoring

- Power-on or front panel reset generates 500mS *SYRES reset (555 U21 Q133) and resets P2 (6809E U35 Q209) via *RES2.

- P2 does vector fetch from $FFFE-$FFFF to get reset address ($F96A) from the F8LMRK17 Q133 ROM mapped to $FF00-$FFFF by the assertion of *VFETCH2 causing *ROMEN to be asserted.
  reset vector can be seen in monitor at $FBFE-$FBFF

3.2 P2 initialisation

- disable interrupts
  interrupts remain disabled throughout boot process

  set stack pointer to $FD40

- clear interrupt controller PIC2 (8214 U37 Q133 at $FCFD)
  clear P2 control bits 7,5,3,1 in CPU control latch (74LS259 U9 Q209).

3.3 Monitor Initialisation

- clear ABOF ($FFCA) abort flag
  clear FLAG ($FF90)
  clear breakpoint table ($FFA0-$FFA9)
  Note that debug ram $FE00-$FEFF is common to P1,P2 and debug ram $FF00-$FFFD is processor unique.

3.4 Synchronisation to P1

- wait for TRIG=0 ($FE00)
  P1 clears TRIG during its initialisation sequence
  write P1SYNC command ($FE15) to JUMPAD ($FE01)
  wait for P1 to execute command and clear TRIG indicating P1 has reached the P2 request loop in its initialisation.
  (see the P2 request loop description in P1 RESET SEQUENCE)

3.5 Display signon message

- P2 directly loads the default palette (for GC2,3) and clears the screen and prints the initial 'BOOT ROM ...' message.

  pallete loading
  - set vertical scroll register to 0
    (6821 port A U8 CG2,3)
    copy this value to image at SCRRAM ($FE0C)
  - if not colour card then complete
  - set control latch to $19 to enable palette ram
    (74LS374 U39 CG2,3)
    The palette ram maps to the low 8K of video memory when enabled.

- initialise all of OSK palette ram
  During boot and at the shell, the OSK palette is used for display.
- restore control latch ($1B)
- zero all video memory to clear screen

printing boot message
- uses bytewise Y auto increments to display characters.
  All characters during boot (before IOPACK) and in QDOS are displayed this way.

3.6 Load disk drivers

- clear DRLIST ($FE69)
  The disk drivers are held in a linked list terminated by $0000 starting at DRLIST. Each entry in the linked list is an 8 byte packet starting with the link to the next packet.
  The disk drivers are loaded at the top of P2 memory.

clear HDBTDR ($FDDC)	hard disk boot driver
clear FDBTFR ($FDD8)	floppy disk boot driver

- initialise end of memory ENDMEM ($FDFE) to $E000 as setup by P1.

3.6.1 Load Waveform Supervisor Driver

- write 'LDRIV' to WSPACK ($FE80)
  Commands are issued to the WS by setting WSPACK to a 4 byte code string. Upon completion, the WS will clear WSPACK.

  assert *RESET and *HALT to WS to hold WS in reset
  WS control latch (74LS259 D3 CMI41)= $0E
  WS control latch = $0A
  set all WS control latch bits high to start WS
  wait for WS to clear WSPACK indicating WS started and driver loaded into P2 memory and linked into DRLIST.
  The WS checks the DRLIST and writes its driver at the highest address below any already loaded drivers.

3.6.2 Load Floppy disk driver

- load DMA address (74LS193 C1,2,3,4 QFC9) to $C000 (location where QFC9 driver is to reside)
  clear control latch (74LS273 B7 QFC9)
  load DMA counter (74LS161 C5,6,7,8 QFC9) to $07FF (size of QFC9 driver rom - 1).
  access *DEVROM latch (74LS74 C11 QFC9) to begin DMA of QFC9 rom into P2 memory.
  This asserts *DTRQST.
  set control latch drive number to 3 (i.e. invalid drive)

- wait for *DTRQST=1 signifying end of DMA of ROM.

- relocate the driver from $C000 to the highest address below any already loaded drivers in P2 memory.

3.7 Print System Information

- clear WSBTFLG ($FE0D)

  **The top bit of WSBTFLG is used to indicate that the system is to boot from the TSCSI port.
  This bit effectively swaps the assignment of the WS SCSI port and the TSCSI port.**

- issue SCMD command to WS via WSPACK to get WS do display system infomation

  **There are several commands sent to the WS with 'CMD' ending. The first letter selects which command to do.
  The WS loads the COM68 program into P2 at $0020.
  COM68 is a program that allows provides an interface between the WS and P2.**

- wait for approx 1s for any user key presses to invoke debugging commands.
  if a key is pressed, the key and 'CMD' is issued as a command to the WS via WSPACK.

3.8 Boot system from disk

- attempt to boot from floppy
  attempt to boot from WS SCSI
  attempt to boot from TSCSI (top bit of WSBTFLG set)

- if waiting for boot, display rotating 'windmill'.

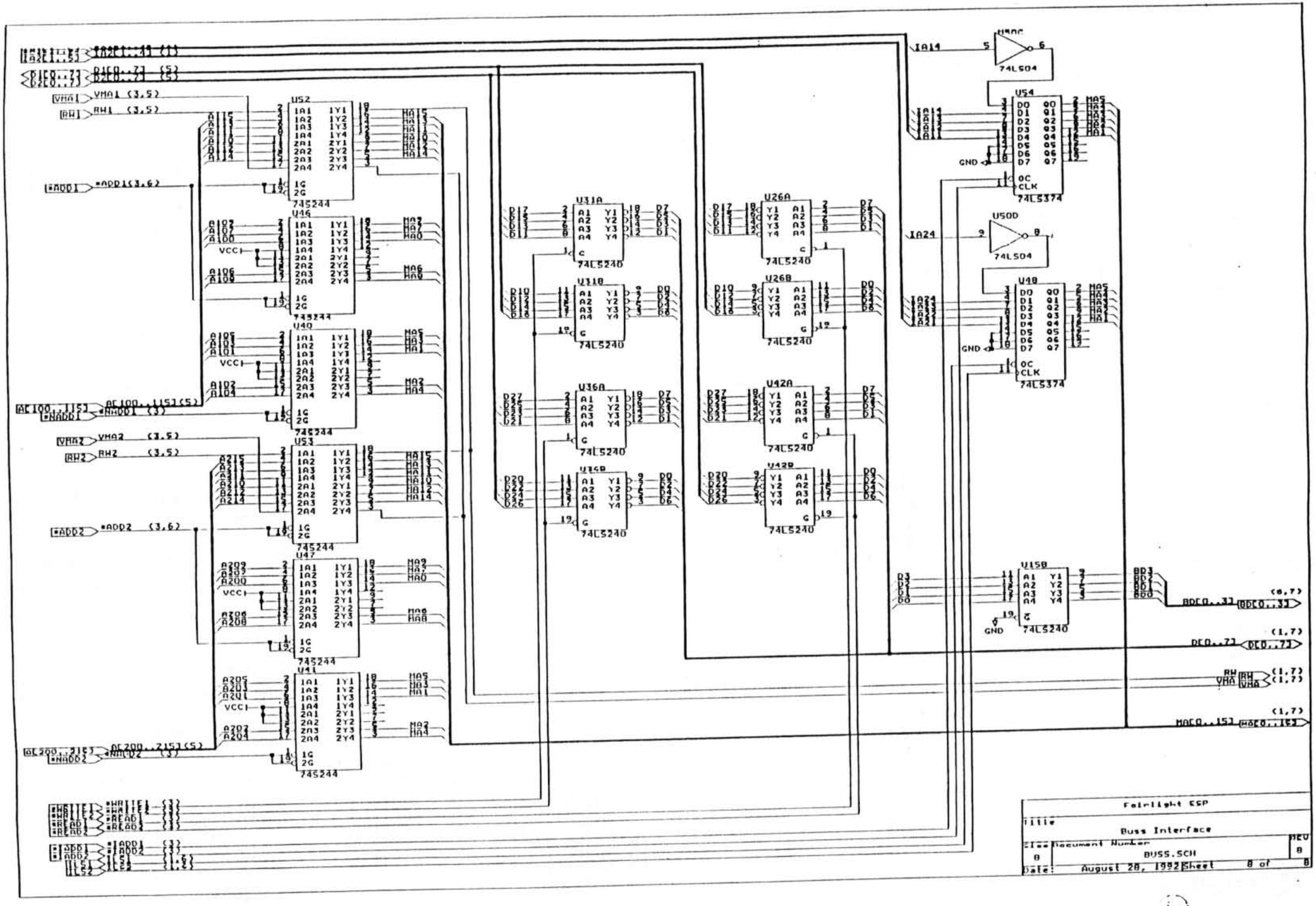
  The boot program is held in sector zero of drive 0 (scsi device 0) and loaded into P2 memory at $0020.

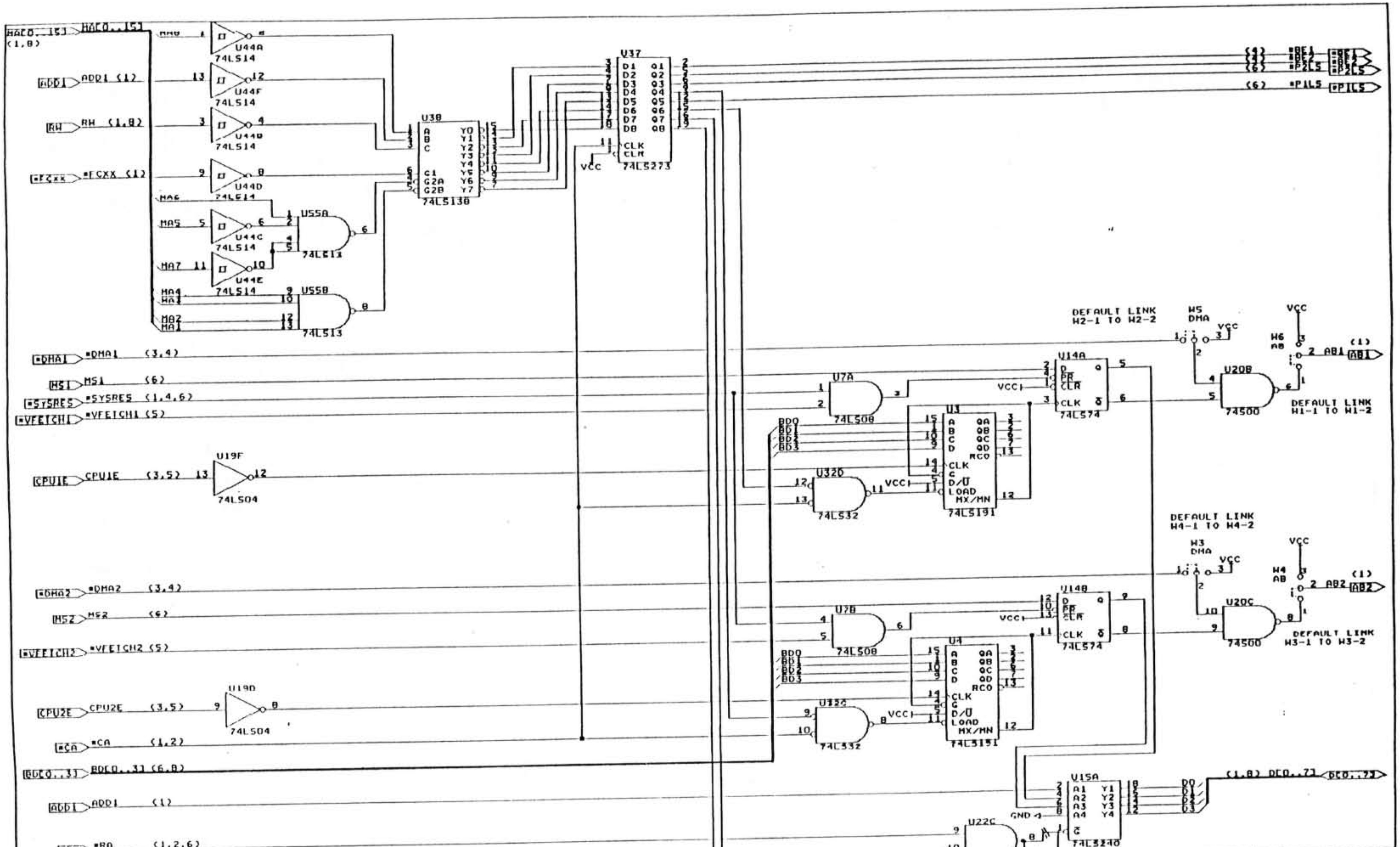
4. BOOTING

- try to read sector 0 at 256 bytes per sector (OS9 BOOT)
  if unsuccessful, then try 128 bytes per sector (QDOS BOOT)
  if unsuccessful, then keep trying to read sector 0.

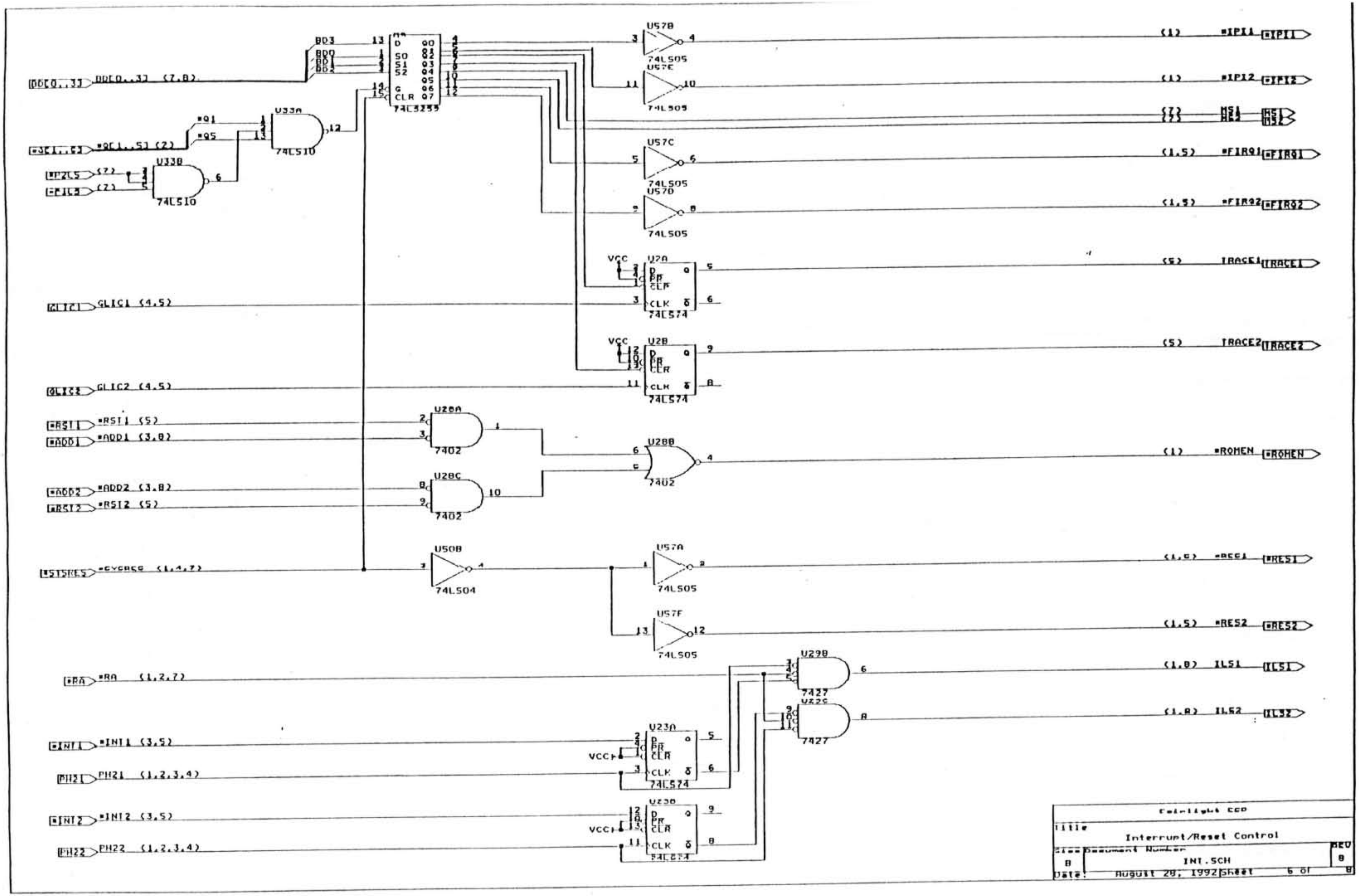
- if QDOS boot
  the first two bytes (at $0020) should be $2112 indicating a valid bootblock - otherwise display 'BAD BOOTBLOCK'

- if OS9 boot
  check for 'GO' as module name of program otherwise display 'BAD BOOTBLOCK'
  The OS9 module name 16 bit offset from the head of the file is held 4 bytes from the start of the file ($0024).

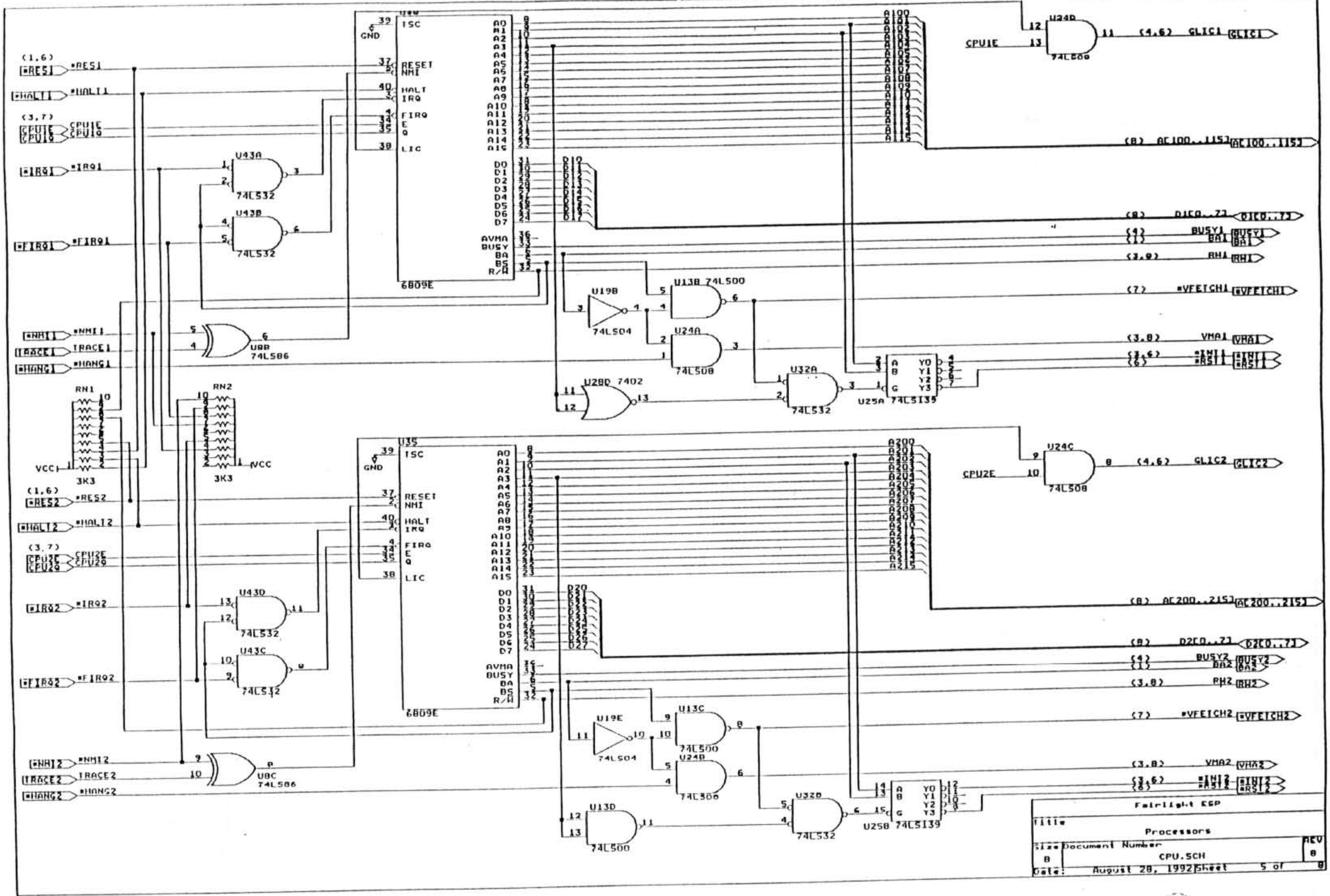

U52
U46
U40
U53
U47
U41
74S244
U31A
U31B
U36A
U36B
U26A
U26B
U42A
U42B
U15B
74LS240
U54
U48
74LS374
U50C
U50D
74LS04
Fairlight ESP
Title
Buss Interface
Size Document Number
B
BUSS.SCH
REV
B
Date: August 28, 1992 Sheet 8 of 8



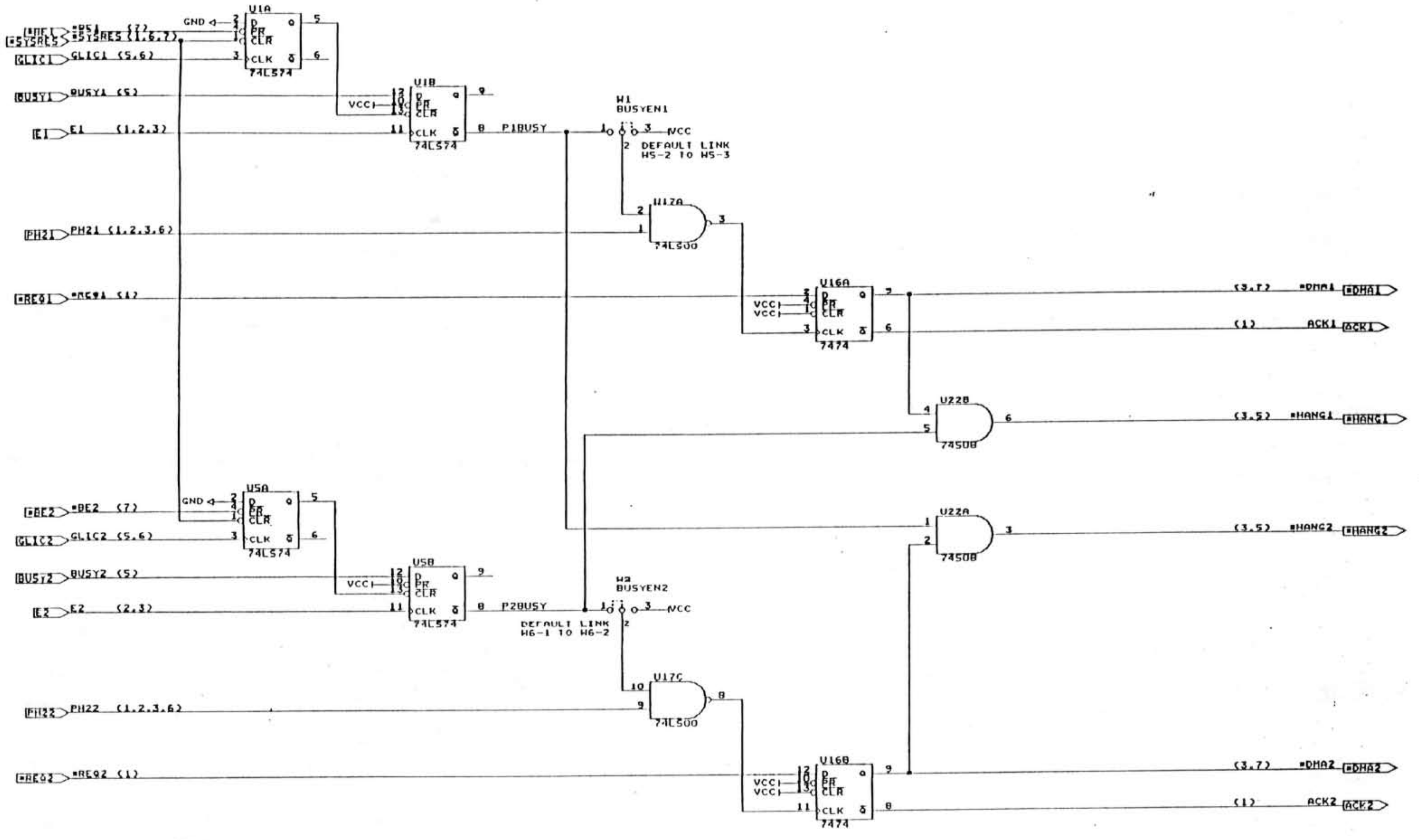
	Fairlight ESP
Title	Map Switching
Size B	Document Number MAP.SCH
REV	B
Date:	August 28, 1992 Sheet 7 of 8



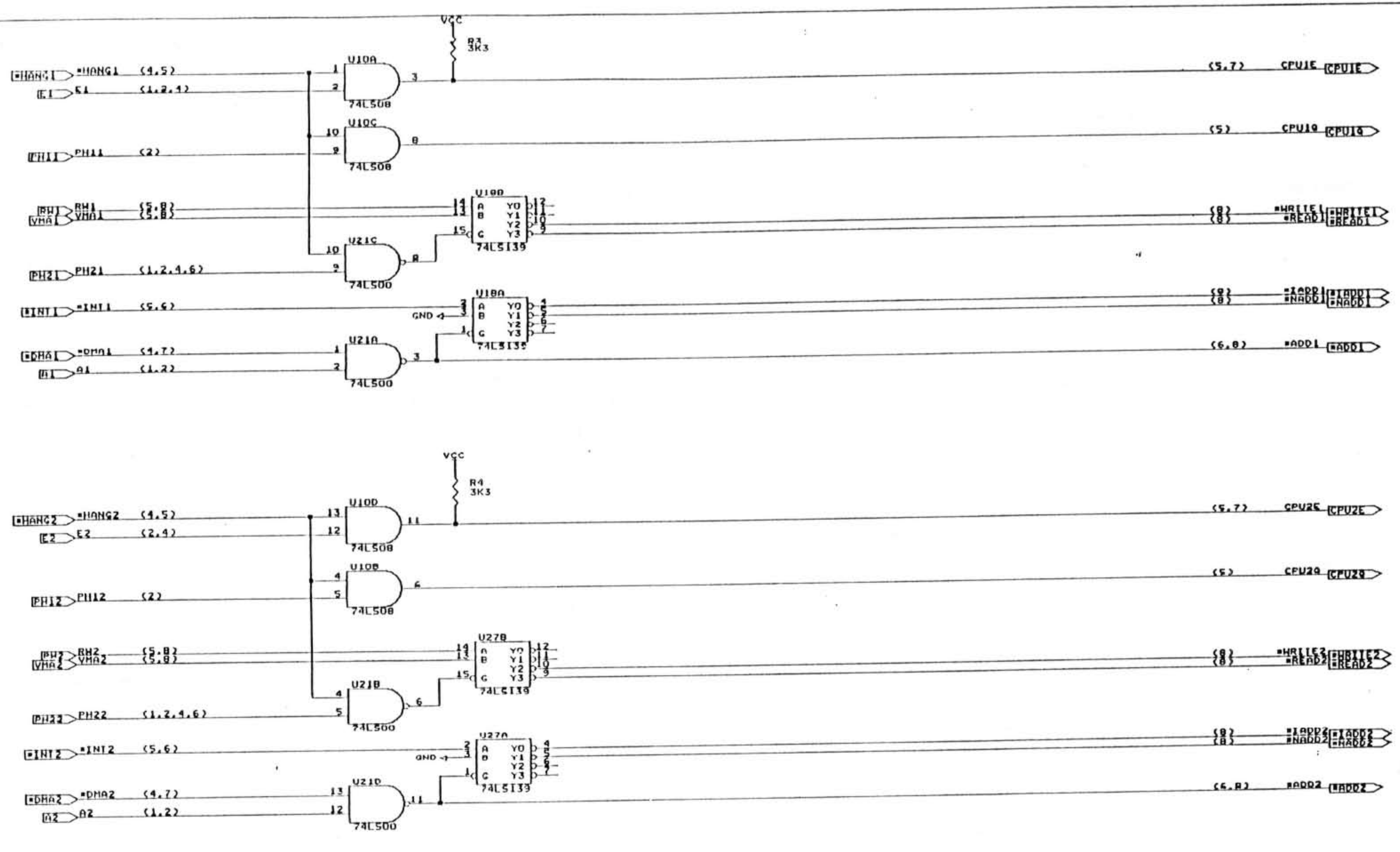
Title	Interrupt/Reset Control
Size: B	Document Number: INT.SCH
REV	B
Date:	August 20, 1992
Sheet	6 of 8



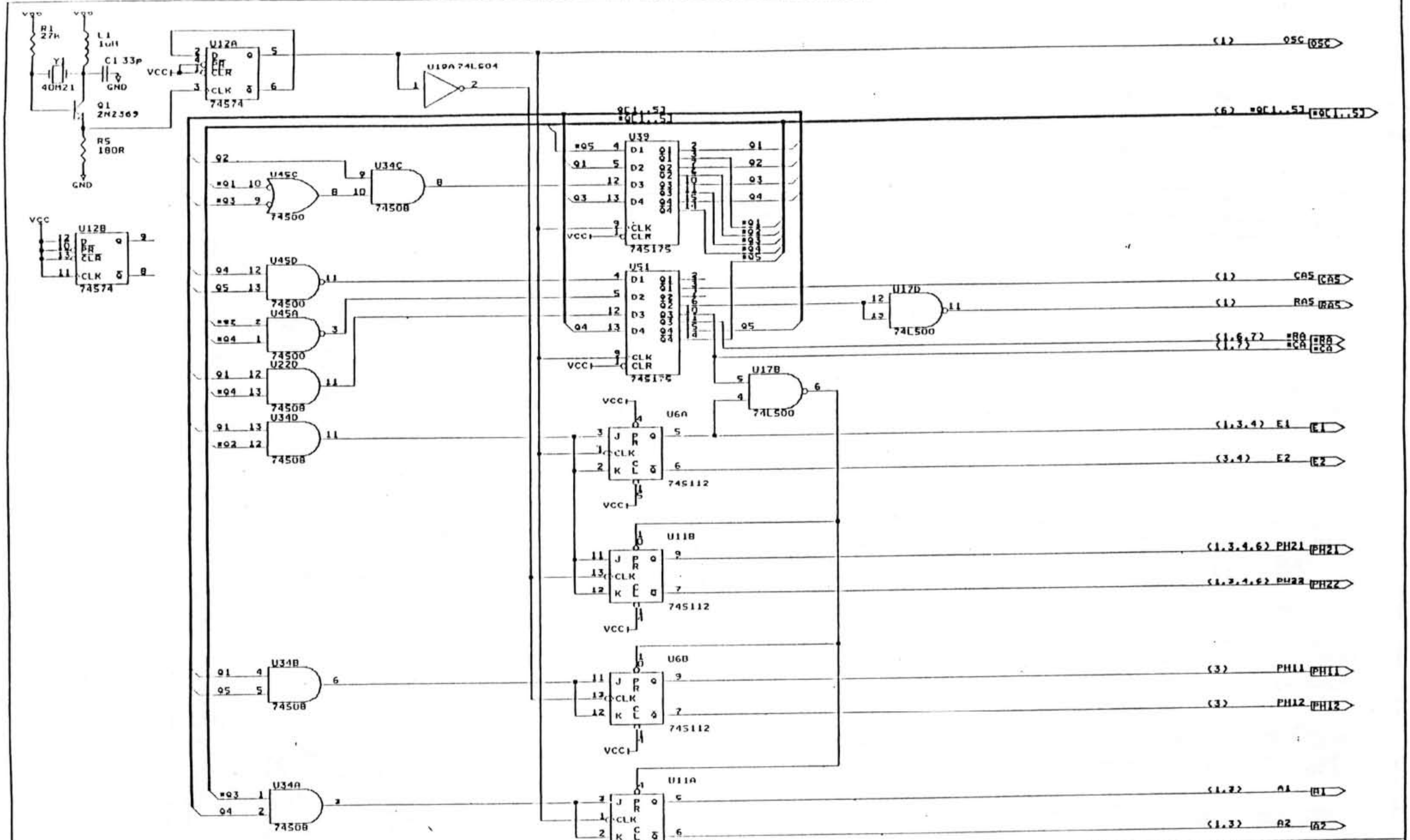
Fairlight ESP		
Title	Processors	
Size B	Document Number CPU.SCH	REV B
Date: August 28, 1992	Sheet 5 of 8	



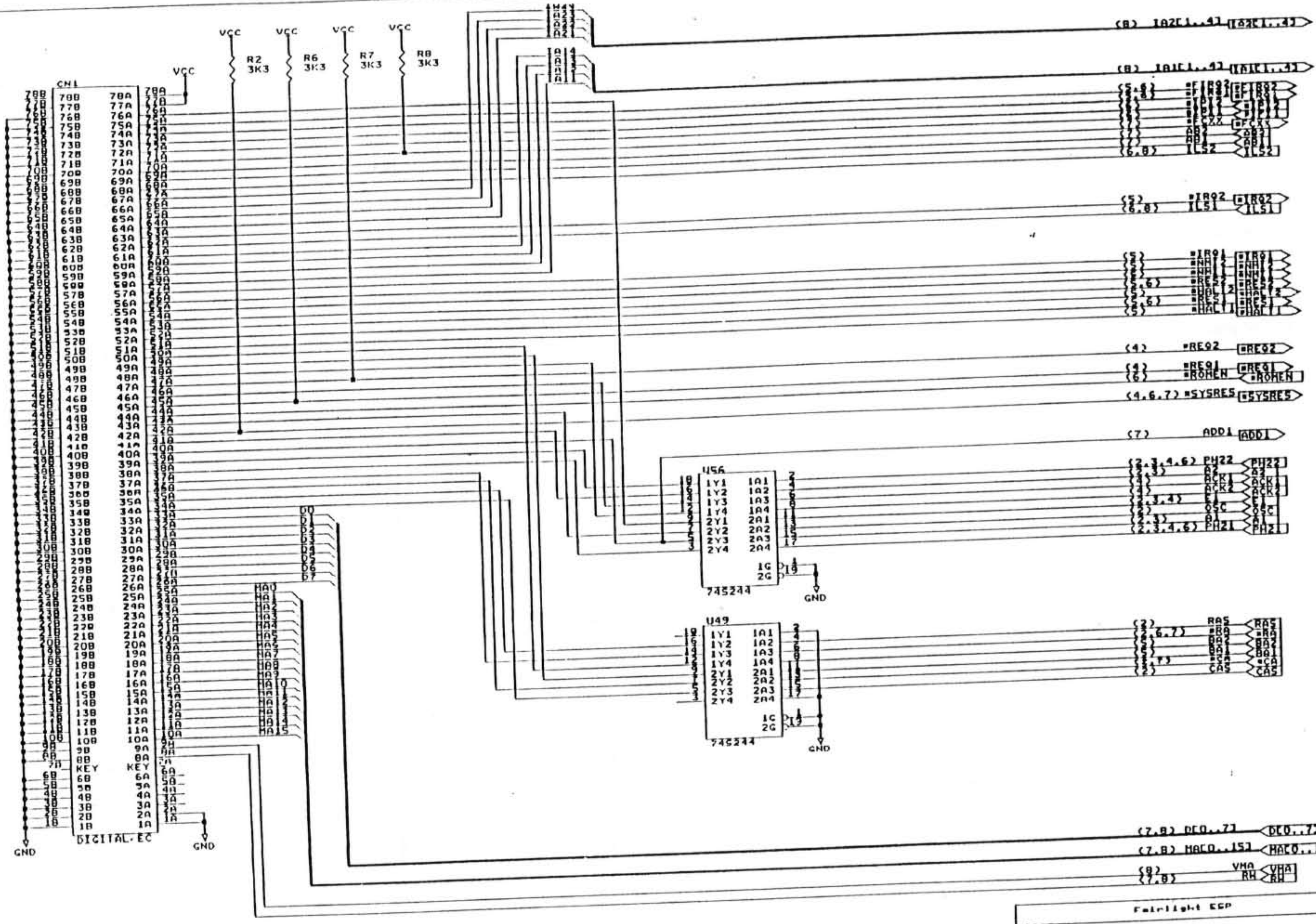
Fairlight EPP			
Title	DMA Control		
Size	Document Number		REV
B	DMA.SCH		B
Date:	August 28, 1992	Sheet 4 of 8	



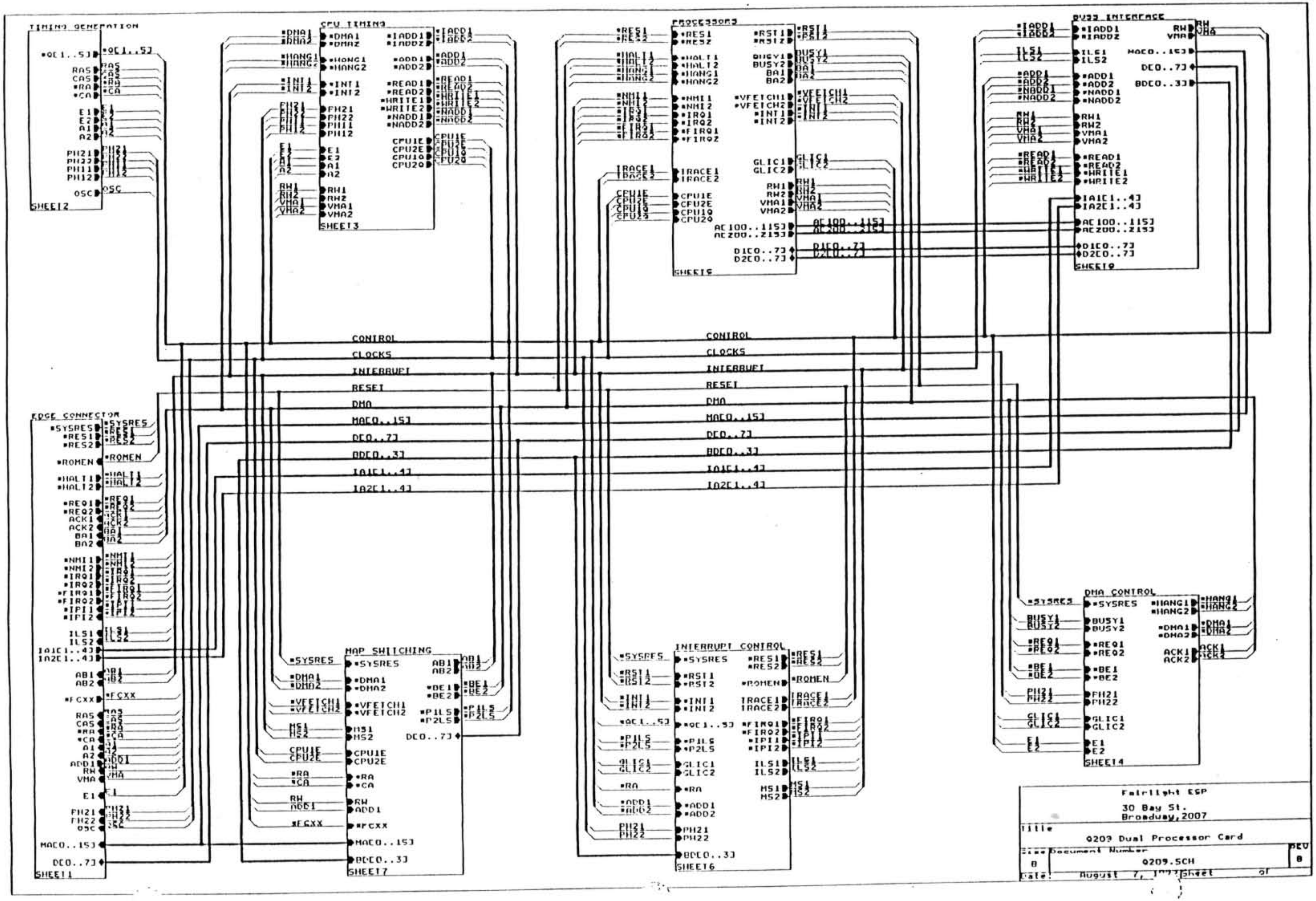
Fairlight EGP		
Title: CPU Timing Generation		
Size: B	Document Number: CPUTIM.SCH	REV: B
Date: August 28, 1992	Sheet 3 of 8	



Fairlight ESP		
Title: Clock and Timing Generation		
Size: B	Document Number: CLKGEN.SCH	REV: B
Date: August 28, 1992	Sheet 2 of 8	



	Fairlight EGP	
Title	Edge Connector	
Size B	Document Number EDGE.SCH	REV B
Date:	August 26, 1992 Sheet 1 of 8	


TIMING GENERATION
SHEET2
CPU TIMING
SHEET3
PROCESSORS
SHEET5
BUSS INTERFACE
SHEET9
CONTROL
CLOCKS
INTERRUPT
RESET
DMA
MAC0..15]
DC0..7]
BDC0..3]
IA1C1..4]
IA2C1..4]
EDGE CONNECTOR
SHEET1
MAP SWITCHING
SHEET7
INTERRUPT CONTROL
SHEET6
DMA CONTROL
SHEET4
Fairlight ESP
30 Bay St.
Broadway,2007
Title
Q209 Dual Processor Card
Size Document Number
B
Q209.SCH
REV
B
Date: August 7,
Sheet
of