

# Q133

CPU Control Card

# 2.3

<b>Introduction.....</b>	<b>2.3.2</b>
<b>Address map.....</b>	<b>2.3.2</b>
<b>Restart and interrupt vectors.....</b>	<b>2.3.2</b>
<b>Debug monitor ROM.....</b>	<b>2.3.2</b>
<b>System boot/disk ROM.....</b>	<b>2.3.4</b>
<b>Address decoding.....</b>	<b>2.3.4</b>
<b>RAM refresh control.....</b>	<b>2.3.4</b>
<b>Static RAM.....</b>	<b>2.3.5</b>
<b>EPROM.....</b>	<b>2.3.5</b>
<b>ACIA.....</b>	<b>2.3.5</b>
<b>PIA.....</b>	<b>2.3.5</b>
<b>Manual controls.....</b>	<b>2.3.6</b>
<b>Power-on reset.....</b>	<b>2.3.6</b>
<b>Interrupt priority logic.....</b>	<b>2.3.6</b>
<b>Schematic diagrams.....</b>	<b>2.3.7</b>

---

## Q133 CPU Control Card

---

### Introduction

The CPU Control Card provides several support functions required by the CPU card. These include startup and bootstrap ROM, 4 serial communication ports, interrupt prioritisation, dynamic RAM refresh, day/date/time of day clock, P1 DMA daisy chain, and a parallel port.

### Address Map

The Debug Card occupies the last 4K bytes of the 65K byte memory addressing space and is set up as follows:-

ADDRESS (HEX)	FUNCTION
F000-F7FF	Rom0 common rom
F800-FBFF	Rom1 processor unique
FC00-FCEF	Available for peripherals
FC80-FC8F	ACIA registers
FC90-FC97	Timer (6840)
FCF0-FCFF	PIA registers, user and clock
FCFC	CPU#1 interrupt prioritiser
FCFD	CPU#2 interrupt prioritiser
FD00-FEFF	Shared 512 byte RAM
FF00-FFFF	Unique 256 byte RAM for each processor

### Restart and Interrupt Vectors

RAM space allocated uniquely to each processor provides independent restart and interrupt vectoring. The vector locations are as follows:

ADDRESS (HEX)	VECTOR
FFFE/F	Restart
FFFC/D	NMI
FFFA/B	SWI1
FFF8/9	Unused
FFF6/7	FIRQ
FFF4/5	SWI2
FFF2/3	SWI3
FFF0/1	Unused
FFEE/F (lowest)	IRQ level 7
FFEC/D	IRQ level 6
FFEA/B	IRQ level 5
FFE8/9	IRQ level 4
FFE6/7	IRQ level 3
FFE4/5	IRQ level 2
FFE2/3	IRQ level 1
FFE0/1 (highest)	IRQ level 0 (highest)

### Debug Monitor

The Q133 contains two 2K ROMs that contain all the basic driver and initialization routines, such as loading the Disc drivers and Q256's maprams.

The monitor ROM occupies 1K bytes from F000 to F3FF and may be accessed by either processor. Processor-unique workspace RAM is used by the monitor so both processors can be executing the monitor independently.

Commands	
/	Reopen last open address as a 1-byte unit
AAAA	Open 2-byte unit at address AAAA
	Reopen last open address as a 2-byte unit
\$A	Open CPU accumulator A
\$B	Open CPU accumulator B
\$X	Open CPU index register X
\$P	Open CPU program counter
\$H	Open user SWI handler address
\$C	Open CPU Condition Code register
\$D	Open CPU D register (A,B concatenated)
\$Y	Open CPU index register Y
\$U	Open CPU User Stack pointer U
\$S	Open CPU Stack pointer S
\$R	Open program segment Relocation Register
\$G	Open CPU direct page register
\$F	Open monitor flag byte
<return>	Close the open location
<linefeed>	Close current, open next location
v	Close current, open previous location
>	Close current, take branch offset and open
@	Open location pointed by current location
AAAA;B	Insert a breakpoint at address AAAA
;L	List all active breakpoints
AAAA;D	Delete breakpoint at address AAAA
;C	Clear all breakpoints
AAAA;T	Insert tracepoint at location AAAA (non-stopping breakpoint)
AAAA;K	Kill tracepoint or breakpoint at AAAA
AAAA;G	Start a user program at address AAAA
;P	Proceed from breakpoint, abort, or call
AAAA;O	Calculate branch offset from open location to address AAAA
HH;F	Fill memory from beginning address to end address
BEG ADDR	User prompt for beginning address
END ADDR	User prompt for end address
<CTRL X>	Abort current command line, take no action
<	Close current location, return to sequence start and open
AAAA,R	Relocate address AAAA by register R. R may be any of the CPU registers, the user relocation register, the monitor flag byte or the currently open location
AAAA.	Relocate address AAAA by Relocation Register \$R
:	Same as linefeed (CTRL J) except that no new line is taken, and neither the address nor contents of the next location is displayed
AAAA#LL	Memory dump of LL lines (16 bytes/line) starting from, address AAAA
'<ASCII chr>	Input ASCII character value instead of hex value for any of the above commands

The 6809 monitor will also accept input of signed hex numbers.

---

## Q133 CPU Control Card

---

### System Boot/Disk ROM

This ROM is used by CPU#2 for disk booting operations and occupies locations F800 to FBFF in the unique ROM space for CPU#2.

The following functions calls are provided:-

- \* Boot load QDOS operating system from disk
- \* Initialise disk controller
- \* Read full last sector
- \* Read partial last sector
- \* Read verify (CRC check only)
- \* Write and verify CRC
- \* Restore head (seek track 0)
- \* Seek to specified track
- \* Write test
- \* Write D.D. mark to sector
- \* Write sectors and verify CRC
- \* Write sectors and don't verify CRC
- \* Check and abort if non-recoverable error

This ROM contains the code to load the actual disk drivers into system RAM. The driver routines themselves are stored in RAM after being loaded from the ROM on the QFC9 floppy controller card, the Q077/Q087 Hard Disk card if present and the Q777 SCSI Controller card.

### Address Decoding

(refer schematic Q133-00)

The System Address Buss is buffered by non-inverting buffers A1 and A4. NAND gate B1 generates an output (asserted LOW) when an address in the range FXXX is detected. This is fed out to the buss on edge connector pin 60B. Further decoding by combinational logic at B4, D4, B3, C2 and C1 generate select signals for the two EPROMS, ROM0, ROM1.

Selection of the on-card static RAM and peripheral devices in the FCFX range are also decoded.

These six select signals are latched by hex flip-flop B6. Hex flip-flops C7 and D7 latch the 11 low-order address bits, as well as the READ/WRITE signal. When any of the on-board devices are read from, inverting data buss transceiver A5 drives the buss. (See drawing Q133-03). At other times, A5 buffers the data into the card.

### RAM Refresh Control

Rate multiplier C10 is configured to produce a 1 microsecond pulse every 16 microseconds. This output generates a DMA request for Processor 1 (RDMA), via DMA hardware at C8, B10, B5 and C4. The refresh has the highest priority in the P1 DMA daisy chain.

The ENL signal, (Enable Next Level), indicates to the next device along the daisy chain when it may make DMA requests. It normally goes low every second P1 cycle, but if a refresh request is pending, the low pulse is inhibited.

When this request is acknowledged by the ACK1 buss signal from the Q209 CPU (asserted HIGH) flip-flop B10 generates a /REF (Refresh, asserted LOW) signal on the buss, which signals a refresh cycle to the dynamic RAMs in the system. At the same time, the output of the refresh address counter A2 is driven onto the buss by tri-state buffers A3. At the completion of the refresh (DMA) cycle, the refresh address counter is incremented ready for the next cycle.

**Static RAM**

*(refer schematic Q133-01)*

A small amount of static RAM is provided for use as scratchpad during disk calls and monitor firmware execution. It is organised as follows:

CPU #1 FF00-FFFD  
CPU #2 FF00-FFFD  
Both FD00-FEFFF

The addressing function for this purpose is generated by multiplexer C9 which is driven by an OR function of address bits 8 and 9. The RAM itself is in the form of two 1K X 4 devices at D8 and D9.

**EPROM**

Four kilobytes of U.V. erasable ROM are used. These are 2716s/2516 single 5 volt supply type.

Their functions are:

Location	Address Range	CPU #	Function
D2	F800-FBFF	1	Startup
D4	F800-FBFF	2	Disk boot
D5	F400-F7FF	Both	I/O functions
D6	F000-F3FF	Both	Debug monitor

**ACIAS (Asynchronous Communications Interface Adapter)**

*(refer schematic Q133-02)*

6551 ACIAs at E4, E5, E1, E2, E3 are used to receive and transmit serial data. The BAUD rate is determined internally via internal dividers, from the baud-rate generator master 1.8432MHz oscillator at D1.

Interrupts generated by the ACIAs go to the system buss via pin 68B of the edge connector.

Data input and output level conversion for the RS232 standard is provided by circuitry on Sheet 3.

The 6551 used for keyboard data is at location E2, 3. The ACIAs at E4 and E5 have optional RS422 transceivers at F7 and F8 as well as RS232, at F6 and F5. The 555 timer at D10 is used in the RS422 buss timeout control.

**PIA (Peripheral Interface Adapters)**

*(refer schematic Q133-01)*

PIA (F10,11,12) is used to provide two general purpose parallel ports. Peripheral connections are made through a 26-way ribbon cable connector on the front of the card.

Interrupts from the PIA are presented to the buss via pins 66B and 67B.

PIA (E9,10,11) is used to interface the clock/calender chip at E12. This clock has a 3.7 volt Lithium cell to maintain the time when the computer is turned off. The battery is not rechargeable and must be replaced when flat. Battery life is approximately 3 years.

---

## Q133 CPU Control Card

---

Diodes CR5 and CR4 isolate the battery from the 5 volt supply, so that the battery is only connected to the clock when the 5 volt supply drops. Transistors Q2, Q1 on drawing Q133-03, and associated components interface the PIA's signal levels to the clock and control the power-down function of the clock so that no false writes occur at power-on and off. An optically isolated power down signal is available at connector pins 61B and 62B, from the opto-isolator at A11.

### Manual Controls

Restart, halt and interrupt controls are provided on the front-panel card Q137. The sole use is for system debugging. In normal use all signals from the Q137 are inactive.

Activating either HALT switch on the front panel sends /HLT1 or /HLT2 to the corresponding processor on the Q209 CPU. When halted, the Buss Available signals from the CPU card W1 and W2 drive open-collector buffers B12 to turn on the WAIT LEDs on the card.

The system can run without a front panel being connected.

### Power-on Reset

555 Timer A12 is used to generate a system-reset signal on power-up or manual restart from the front panel console, if restart is enabled on both processors. This is a low-going pulse of about 500 milliseconds on buss pin 42.

### Interrupt Priority Logic

8214 Priority Interrupt Control Units (PICU) are used to latch interrupt requests and generate a priority level which is used by the CPU card to create an interrupt vector address. Each processor has its own PICU.

The priority level for each PICU is established by writing the complement of the desired priority level into the status register. The address for CPU 1 is FCFD, for CPU 2 it is FCFC. Decoding for this purpose is performed by one-of-eight selector B8.

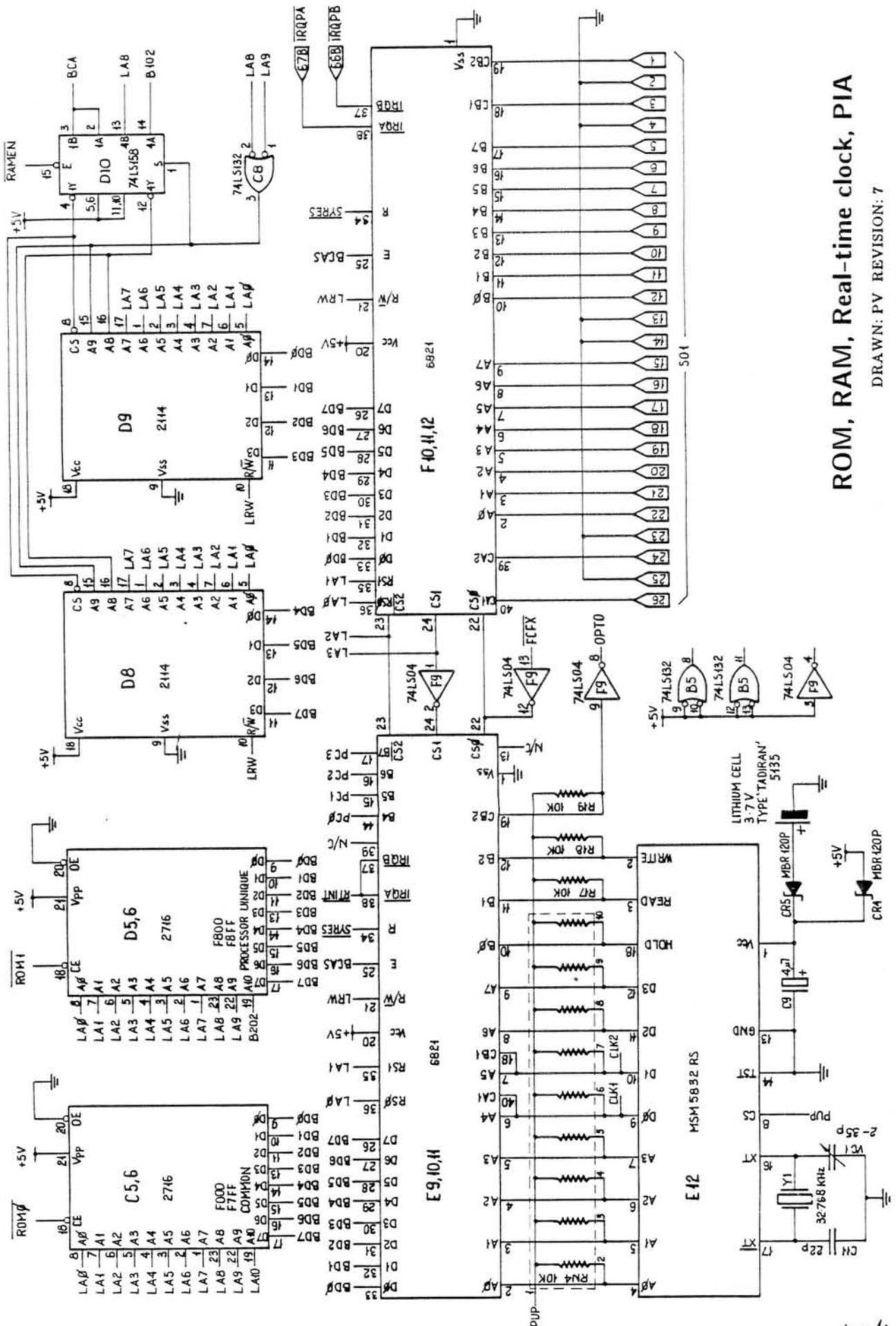
Interrupt requests generated by the PICU are latched by flip-flops B9, which are reset when the PICUs are written to to establish the new priority level mask.

The PICUs are clocked by Interrupt Latch Strobe signals from the bus (ILS1 and ILS2).

Each PICU supports up to eight levels of interrupt.



# Q133-01 CPU Control Card

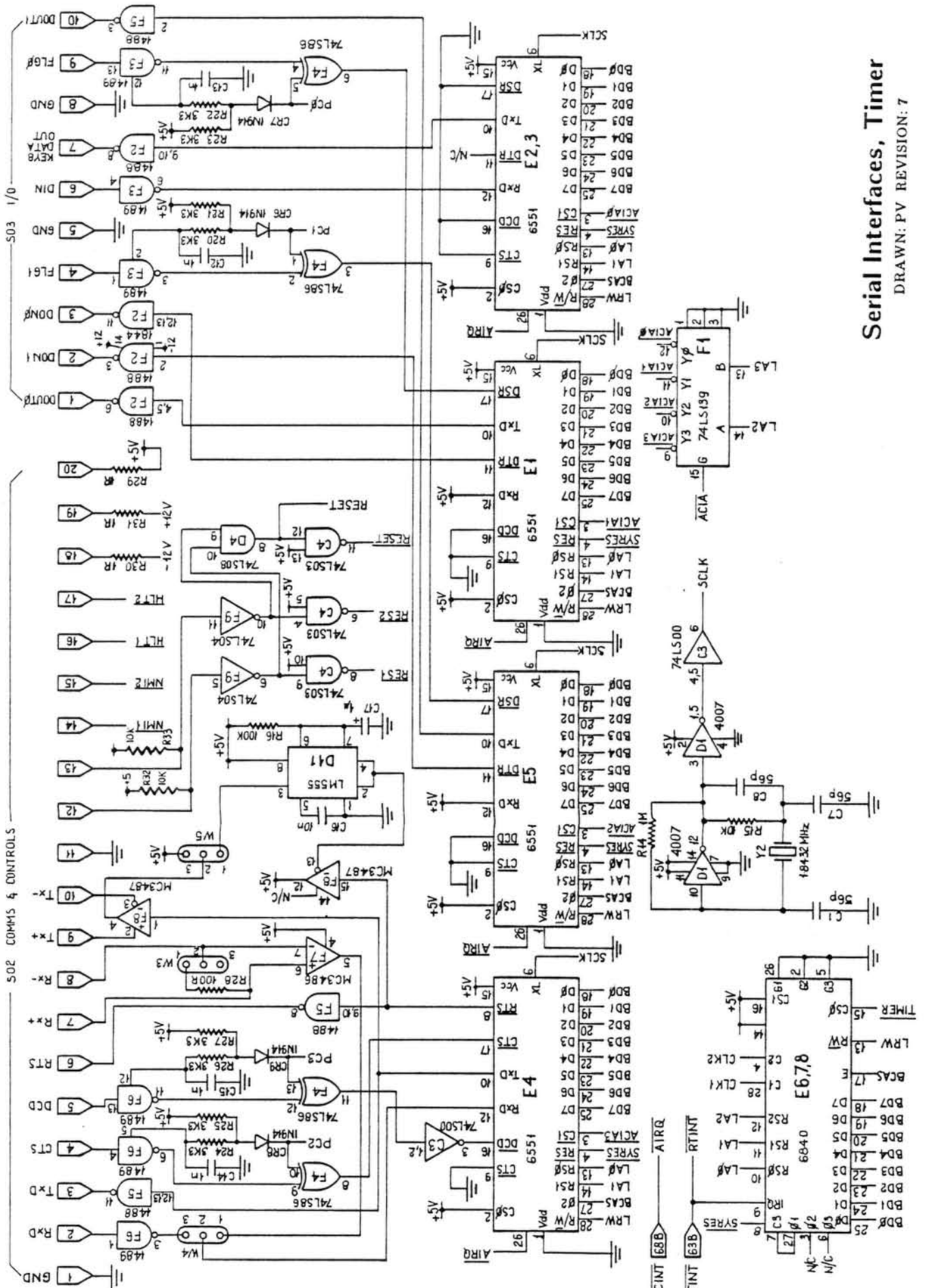


ROM, RAM, Real-time clock, PIA

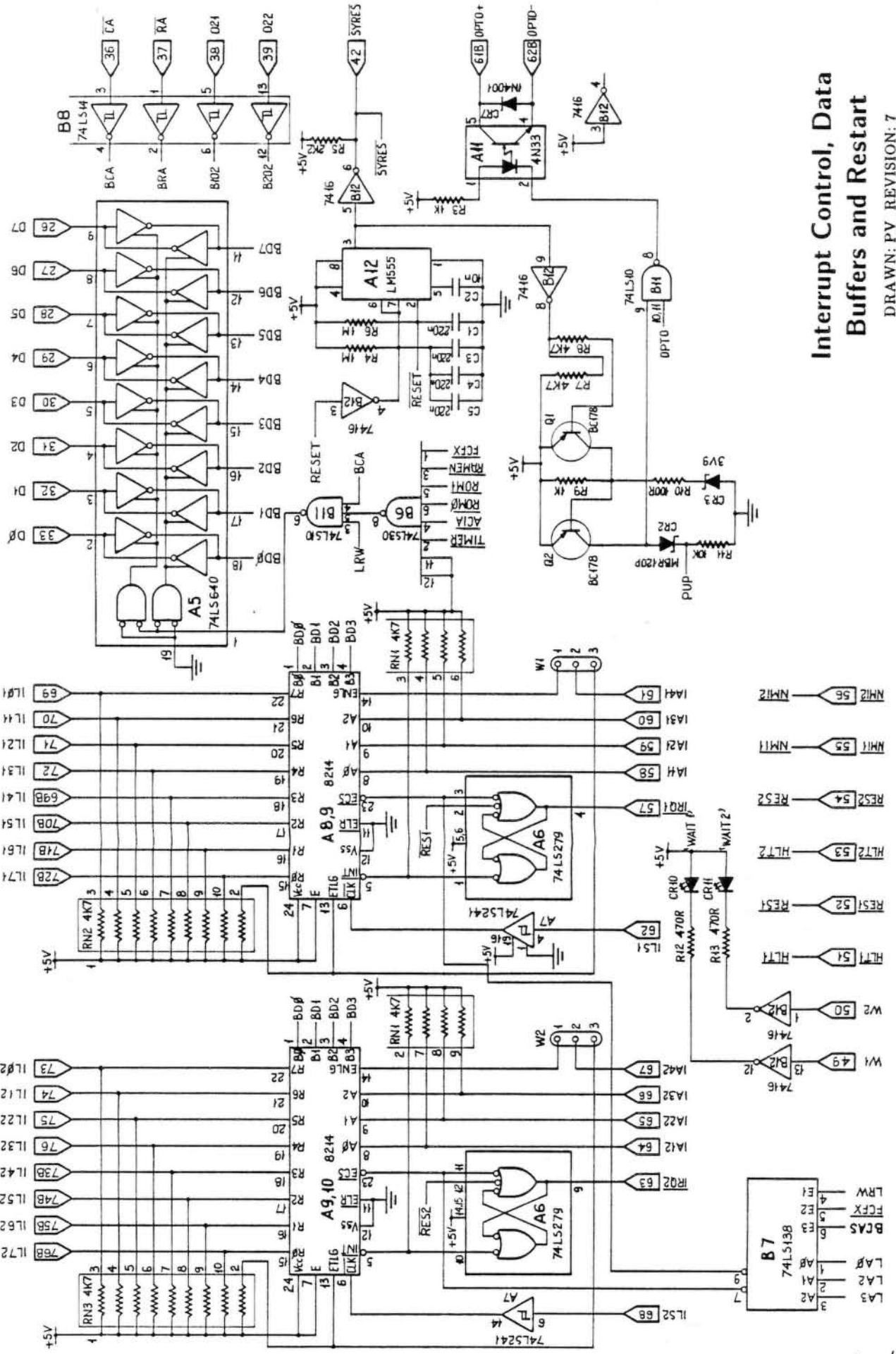
DRAWN: PV REVISION: 7



Serial Interfaces, Timer  
DRAWN: PV REVISION: 7



*fairlight*

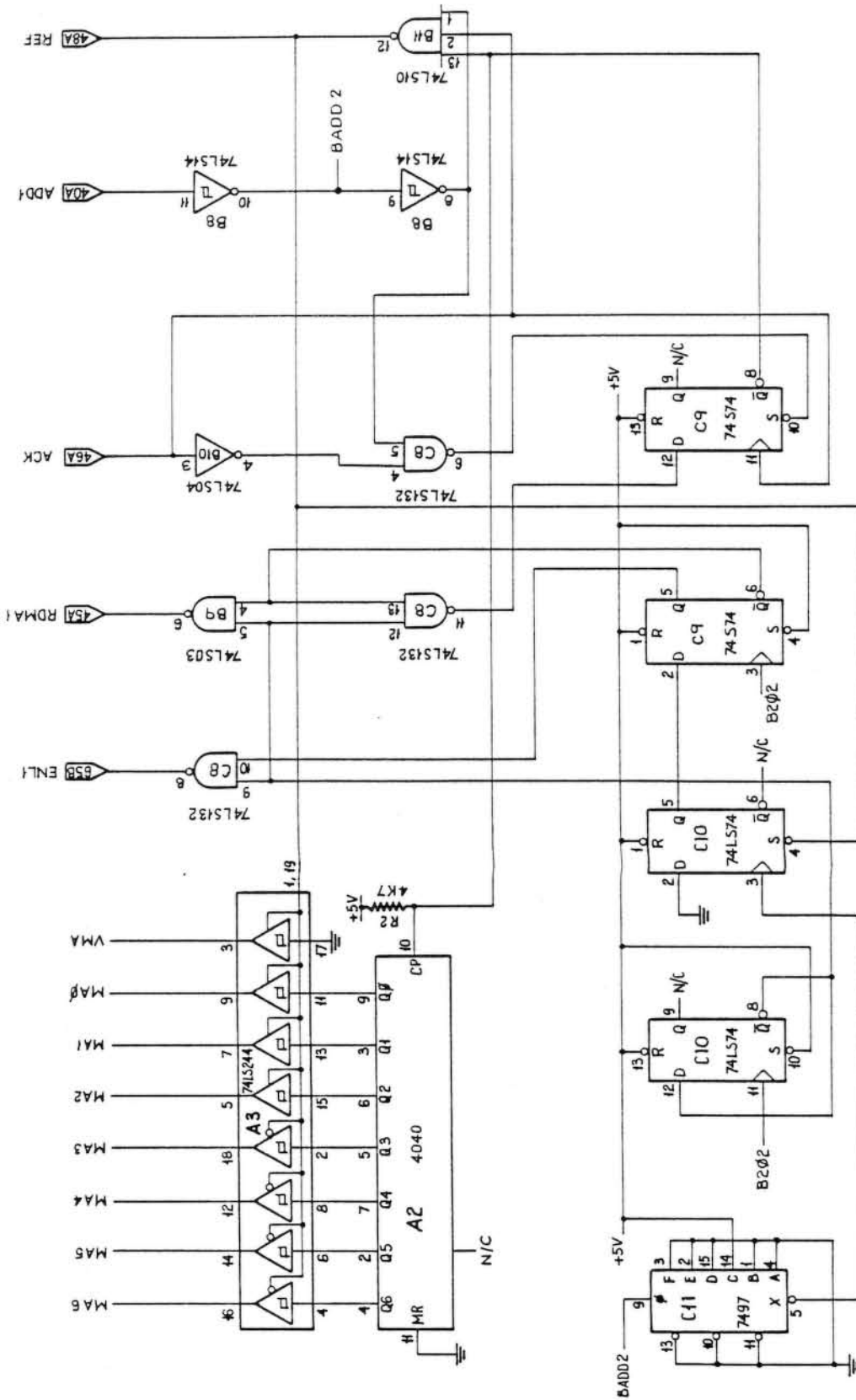


Interrupt Control, Data Buffers and Restart

DRAWN: PV REVISION: 7



**Memory Refresh**  
DRAWN: PV REVISION: 7



*fairlight*



# Q256

256K Ram Card

# 2.4

Introduction.....	2.4.2
Options.....	2.4.2
Map selection logic.....	2.4.3
Address translation.....	2.4.4
Memory block decoding.....	2.4.6
Bus interface.....	2.4.6
Parity system.....	2.4.6
Memory Array.....	2.4.7
Schematic diagrams.....	2.4.8

---

## Q256 RAM Card

---

**Note** - In this document active low signals are indicated by a slash (/) in front of the signal name. This document is updated for the Rev. 2 Q256.

### Introduction

The Q256 is a 256K x 9 bit dynamic RAM, organised as four blocks of 64K and mapped in 2 or 4K chunks. 32 different mappings from "processor space" to physical memory space may be set up. The mapping selected for any given cycle is automatically switched according to the current machine state. The machine state comprises which processor is on the buss, the user state/system state output of the processor, and which DMA channel is active, if any.

The ninth bit in the memory is a parity bit. Parity generation takes place automatically upon writing to the RAM and parity error detection is automatic when reading. If a parity error is detected, an interrupt is generated. A status register records that a parity error occurred, the physical memory block in which it occurred, and the upper five bits of the processor address which was active at the time of the error. The error status bit is automatically cleared after the register has been read.

The map selection logic also generates three control signals: 1) PENB is a universal buss signal which enables or inhibits access to all peripherals on the buss. This signal is fed via the motherboard back into the Q256 since the map selection logic and the mapram are themselves peripherals. The output signal is forced active after power up to ensure configuration of the mapping system is possible, and released by the first read of the parity status register.

- 2) VENB is a video RAM enable bit which allows accesses in the range \$8000 to \$BFFF to read or write to the graphics RAM or user RAM which may be mapped into this area instead.
- 3) PERGEN is a bit used to artificially generate parity errors to for testing purposes.

### Options

Option blocks W1 and W2 allow selection of mapping on 2K or 4K byte boundaries as marked on the component overlay. If 4K mapping is selected, every second double-byte mapram location is not used. W1 and W2 must be configured identically. Default links on the PCB are for 2K mapping.

Option block W3 is for debugging purposes and need only be installed if a faulty card is crashing the data buss of the test machine. The default PCB link to +5V must be cut. There are two non-default options. Option 3 (GND) permanently inhibits the data buss driver buffer. The memory can still be written to but read data will only get as far as the buss output buffer. This facility allows the operating system to boot and test programs to run on a healthy board while the faulty board runs in parallel without driving the data buss e.g. for signature analysis.

Option 1 enables the data buss permanently and is not normally used. The default link to DBEN prevents the buss driver from outputting data only until the first read from the status register. This allows the restart ROM program to initialize the memory mapping before enabling memory mapping.

A four-way DIP switch is provided to allow multiple Q256 boards to be installed. Only switches 1-3 are used, so up to 8 boards can be installed. Close a switch for each zero in the board number, i.e. card 0 has all switches closed. SW1 is the LSB.

### Map Selection Logic

(refer schematic Q256-00)

The function of the map selection logic is to encode the current system state and generate a five-bit map selection number. It also generates the peripheral enable output signal (PENBOUT), the video ram enable signal (VENB) and a parity error generate signal (PERGEN) which forces an artificial parity error for testing purposes.

There are six possible states for each processor:

A or System state, no DMA

B or User state, no DMA

DMA on any one of four DMA channels. (Processor automatically switches to A state for DMA cycles).

The A/B/DMA state is encoded as three bits and the processor phase signal is added as a fourth bit and presented as an address via the multiplexor IC 4B to the map selection RAM (mapsel) ICs 5D and 6D. Thus each state corresponds to a location in the mapsel and its output data is the map selection number.

DMA claim signals (DMACpn, where p=processor and n=DMA channel) occur when a DMA peripheral has received a DMA acknowledge from the processor and arrive during the data phase preceding the actual DMA cycle. IC 13A is an 8 to 3 line encoder but since there are four channels for P1 and four for P2 the most significant output line only duplicates the processor phase signal and is not used. The GS output indicates that some DMA channel is active. Each processor has its own A/B line (AB1 and AB2). The nand multiplexor 9A selects whichever is relevant for the next cycle. This signal, plus the combinatorial logic of 10B, 11A and 11B and the buffered phase signal ( $\overline{B\phi 22}$ ) provide the four-bit state number to the multiplexor 4B.

The mapsel RAM occupies locations FC40-FC4F. Since there are six possible states per processor only twelve locations are actually used, as follows:

FC40	P2 B state
FC43	P2 A state, no DMA
FC44	P2 A state, DMA channel 1
FC45	P2 A state, DMA channel 2
FC46	P2 A state, DMA channel 3
FC47	P2 A state, DMA channel 4
FC48	P1 B state
FC4B	P1 A state, no DMA
FC4C	P1 A state, DMA channel 1
FC4D	P1 A state, DMA channel 2
FC4E	P1 A state, DMA channel 3
FC4F	P1 A state, DMA channel 4

Access to these locations for initialization of the mapsel is decoded by IC 1B, along with the peripheral enable input PENBIN. The output of IC 1B is latched on rising BRA by IC 4A (drawing Q256-03) to produce LFC4X.

Writing to the mapsel RAM is the most time-critical of all operations on the Q256. Normally the entire current data phase is available to generate the map selection number for the next cycle but when writing, the current data phase must be used to write to the mapsel ram as well. This is achieved by making mapsel write cycles very short. To choose locations within \$FC4X the mapsel address multiplexor IC 4B is switched over to the lower four latched address bits on the falling edge of ADD2ø2 (data and address busses are in phase). The data is written into the mapsel ram on the rising of BRA (IC 10B). The write pulse is removed and the multiplexor switched back to the current state number only 50nS later by the falling of BCAS (IC 10A).

The lower five bits of the mapsel RAM are the map select number (MAPSEL0-4). This plus the controls PENBOUT, VENB and PERGEN are latched on the rising of ADD2ø2 which begins the next address cycle.

The peripheral enable signal (PENBOUT) and the graphics enable (VENB) come from bits 7 and 5 of the mapsel ram respectively. After power up or system reset they are both forced active (high) by the flip-flop IC 14B so that access to the mapsel ram is ensured for initialization. This flip-flop also inhibits the data buss driver IC 9B (drwg Q256-03) if W3 has been linked to option 2 for debugging. The flip-flop is triggered as soon as either processor reads the parity status register and will remain set until the next SYRES.

Format of data written to the mapsel RAM is as follows:

- Bit 0-4 Map select number MAPSEL0-4. (Inverted)
- 5 VENB (Write 0 to enable graphics ram)
- 6 PERGEN (Normally 0, 1 to force a parity error)
- 7 PENBOUT (Write 0 to enable peripherals)

### Address Translation

*(refer to Drawing Q256-01)*

This section performs the mapping from the 64K processor address space onto the 256K physical address space. The outputs LMAP0-4 and MAP5,6 plus standard address lines MA0-10 constitute the 18 bit address required to uniquely access any location in 256K.

All mapping data is stored in the two 2148 static RAMs (mapram), each containing 1024 x 4 bits. If 2K mapping is selected, the lower five address lines of the mapram come from the upper five processor address bits via multiplexor ICs 2B, 3B and 3D. This divides the 64K processor space into 32 blocks of 2K each. In 4K mode only the top 4 processor lines are used and the LSB of the mapram address is tied low. The upper five mapram address lines come from the latched map select number, thus any one of 32 different complete mappings may be selected. The data outputs from the mapram include a card select bit (CSEL), a two-bit 64K rank select (MAP5,6), and a five bit page select which is latched on rising BRA (LMAP0-4). The page select bits become the upper physical address bits.

Although the mapram is 1K bytes in size it in fact occupies 2K of address space, from \$F000-F7FF. Even locations in this range are dummy locations which serve only to set up the CSEL flip-flop, IC 1E. Writes to odd locations then store this single bit along with 7 bits of mapping data in the mapram. So a single 16-bit write to the mapram maps one 2K or 4K page of processor space to any 2K or 4K physical page in the 256K available. The processor space being mapped is determined by the address within \$F000-F7FF written to and the physical page selected is given by the data written.

Valid addresses in the mapram range are decoded with the buffered peripheral enable signal (BPENBIN) by IC 2A and the write map (WMAP) signal is latched by IC 4A (drwg Q256-03). The mapram address lines are switched over to LA1-LA10 on the rising of  $\overline{ADD2\phi 2}$  by IC 2D. The 2148s have common data-in and data-out lines multiplexed by the write ( $\overline{W}$ ) input. If an odd address is being written, the buffered data buss is driven into the 2148s from IC 4C and the  $\overline{W}$  input strobed by the upper LS20, IC 1D on the rising of BRA. The write pulse is terminated by falling  $\overline{ADD2\phi 2}$ , and data hold time to the 2148s is provided by the disable delay of IC 4C.

Writing to an even address triggers the CSEL flip flop IC 1E on rising BRA. The data to this flip flop is the result of the three-bit comparison of BD0-2 with the card select no. set up on the DIP switch, qualified by BD7 which may be used as an overall page enable bit. Thus mapram data format is as follows:

Even locations

- Bit 7 Page enable
- Bits 5-3 Unused
- Bits 2-0 Card select

Odd locations

- Bit 7 Unused
  - Bits 6,5 Physical 64K block select
  - Bits 4-0 Physical 2K block select
- (Bit 0 not used in 4K mapping configuration)

Note that when mapping multiple pages of memory in or out the CSEL flip-flop need only be set up once, after which only writes to odd locations in mapram are required.

---

## Q256 RAM Card

---

### Memory Block Decoding

(refer schematic Q256-02)

This section generates RAS and CAS controls for each of the four memory ranks according to mapping outputs MAP5,6 and the card select signal CSEL.

MAP5,6 are decoded to a 1-of-4 block select signal by IC 1E provided a valid address is on the buss (VMA), the Debug's Ram inhibit (RAMINH) is inactive, and neither SFC4X nor the mapram are being accessed. IC 5E latches this select along with the refresh signal REF, read/write (mapsel, mapram, or status), read/write memory, and access memory signals on rising BRA. IC 2E generates the READ signal to drive the data buss while BRA is high (see drwg Q256-03).

Refresh cycles cause the system RAS signal to be distributed to all four memory ranks simultaneously through ICs 6F and 6E. No rank select and thus no CAS is generated during refresh cycles. During a valid memory access one rank is selected and CAS is routed to that block through ICs 5F and 6E. The R/W signal drives all ranks through ICs 5F and 3E.

### BUSS Interface

(refer schematic Q256-03)

The buss address lines are buffered by ICs 6B and 5A and latched on rising BRA by ICs 4A and 5B for mapram and mapsel ram writing and status register reading, along with address decode signals WMAP and FC4X.

Unmapped address lines MA0-10 (and MA11 in the 4K mapping configuration) plus mapping outputs LMAP0-LMAP4 (excluding LMAP0 for 4K mapping) are multiplexed onto the RAM address lines by ICs 5C and 6C.

IC 9B latches the RAM outputs on the falling edge of CAS and drives the buss when READ is high unless the debug option W3 prevents this. Data is buffered off the buss to mapsel, mapram, main RAM and parity generator by IC 7B.

### Parity System

(refer schematic Q256-04)

The parity bit is automatically generated by IC 7D from the data on the buss when writing to RAM. This bit is written into the appropriate 6665 when the corresponding CAS signal is generated (see drawing Q256-02).

A RAM read cycle causes the parity bit from the selected block to be read out of the parity RAM while IC 7D simultaneously regenerates the same parity bit from the data actually leaving the main RAM and coming back in again through data buffer IC 7B. Thus parity errors may be caused by buss errors as well as RAM faults. The parity memory bit and the regenerated parity bit are compared by exclusive or gate IC 4F. The result of this comparison is only valid for about 150nS and is clocked by rising BRAS into flip flop 14B to generate the parity error interrupt (PERRINT) and the parity error status bit (PERR). Further clocking to the flip flop is disabled by the Q output going low.

On each RAM cycle, the rank select lines MAP5, 6 and 5 upper processor address lines are latched by IC 7A. As soon as a parity error occurs this latching is disabled by the /Q output of the parity flip-flop so that the physical block and the processor 2K address space in which the error occurred is recorded.

The parity system status register occupies the same space as the mapsel RAM: FC40 to FC4F. (The mapsel is write only, the status register is read only). The lower 3 bits of the latched address are compared with the module select lines MS0-2 from the DIP switch by IC 2F so that in a multi-card system parity registers on cards 0-7 are at FC40-7 respectively. A read from the status register enables the latch 7A and buffer 6A onto the data buss on the rising of /BRA and the parity error is cleared by a very short pulse (approx 20nS) on the rising of BRA at the termination of the read cycle. The flip flop is also cleared automatically by /SYRES.

The parity error generate signal PERGEN is used to artificially create a parity error for testing purposes. The I input of IC 7D is simply a ninth data input so that if it is 0 when a RAM location is written it must also be 0 when that location is read, otherwise an "error" will be generated. When testing the parity system, a location is written with the I input 0 and the read back just as the I input is switched to 1. PERGEN is the control for this and comes from the mapsel logic. Since the mapsel outputs are of a lookahead nature, the PERGEN signal is delayed by half a cycle by first latching on rising ADD2φ2 at IC 4D then clocking on rising /ADD2φ2 at flip flop IC1E.

In the diagnostic tests the A state is initialised with PERGEN reset and the B state is initialised with PERGEN set. After writing a location in the A state, the processor switches to the B state at the instant of reading back the location.

#### Memory Array

(refer schematic Q256-05)

The dynamic memory array consists of four ranks of nine RAM ICs. Each IC is a 64K x 1 bit device, so each rank forms one byte plus parity bit. Which rank is accessed depends on which /RAS and /CAS lines are driven low by the decoding circuitry.

The RAM chips each have 8 address lines. Since 64k space requires 16 address bits the full address is multiplexed onto the 8 lines. The lower 8 bits ("Row" address) is latched into the RAM chips when /RAS goes low. After a period of address hold time the address multiplexor switches over and drives the upper 8 bits ("Column" address) into the RAMs. This is latched into the RAMs when /CAS goes low.

During a write cycle, data is latched into the RAMs on the falling edge of /CAS. In the case of a read cycle, data out becomes valid within 75nS of the falling edge of /CAS. The data lines are driven by the selected rank of RAMs while /CAS is low, and go tri-state at other times.

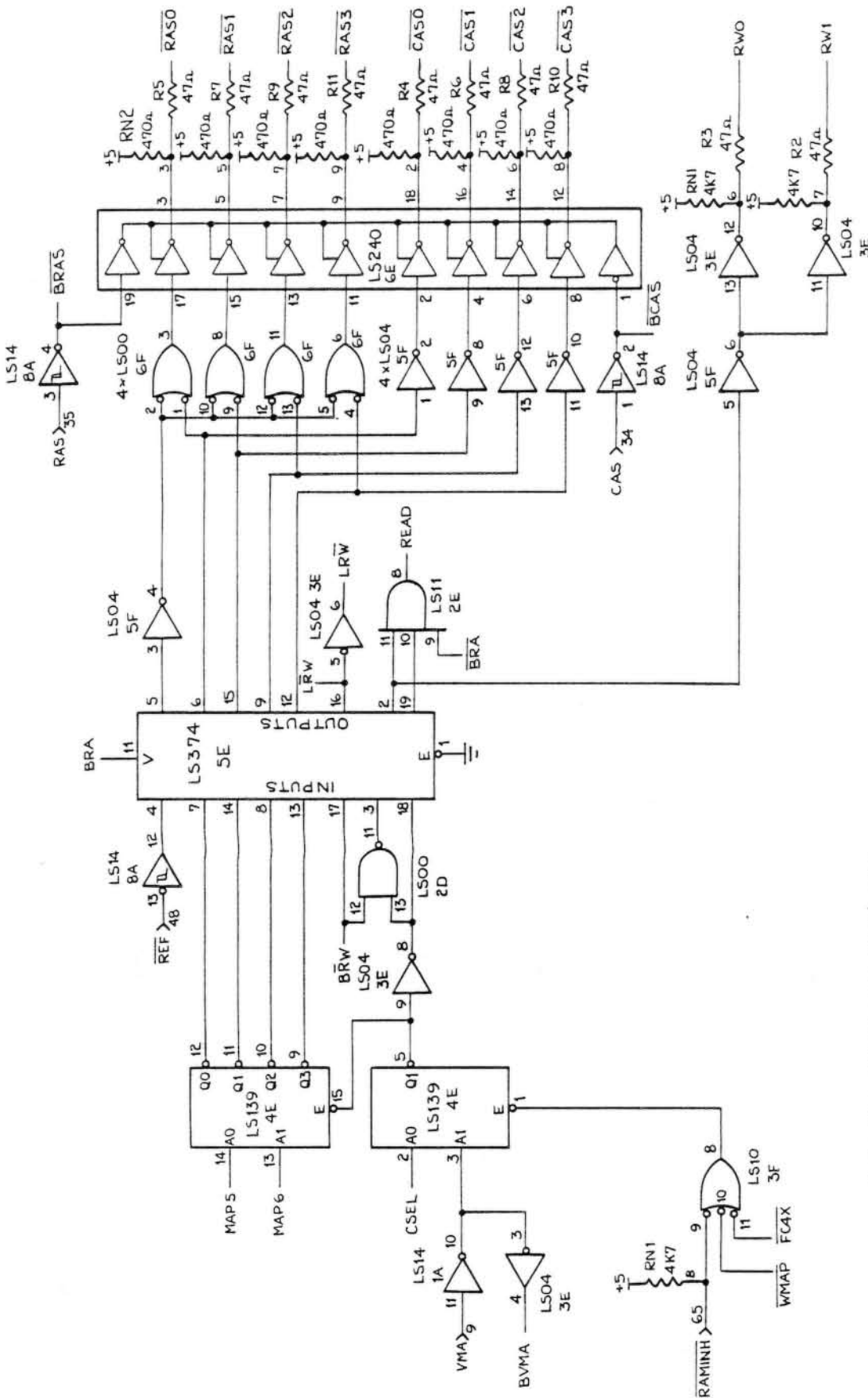
The state of the /W signal while /CAS is low determines whether the cycle is read or write.

Due to the capacitive input impedance of the MOS RAM ICs, the address and data input lines are driven through series resistors to limit the voltage undershoot.





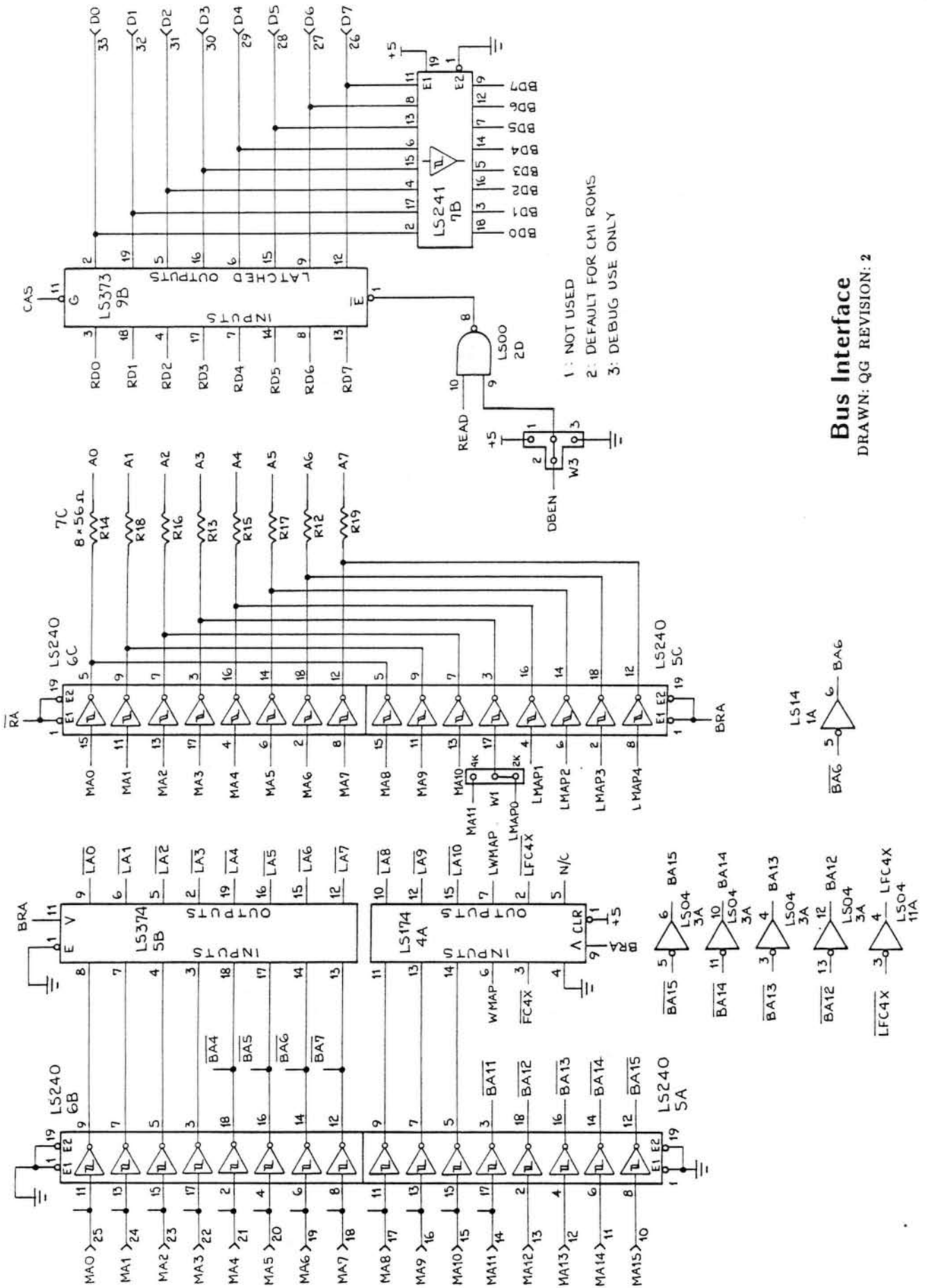
# Q256-02 RAM Card



NOTE: 70ns ACCESS TIME 214B'S MAY BE USED AT 1C AND 2C PROVIDED AN 5139 IS USED AT 4E

## Memory Block Decoding

DRAWN: QG REVISION: 2

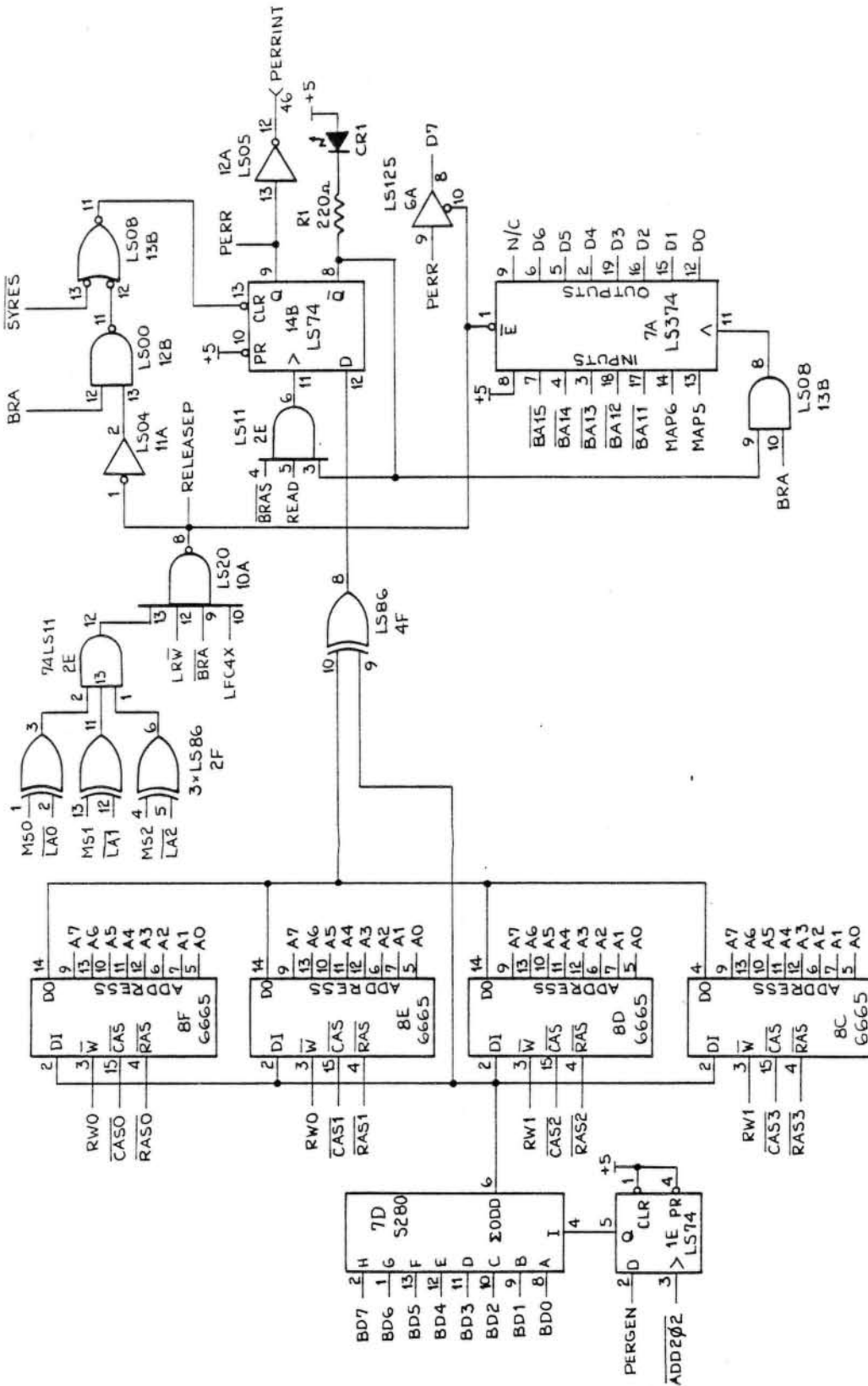


1: NOT USED  
 2: DEFAULT FOR CMI ROMS  
 3: DEBUG USE ONLY

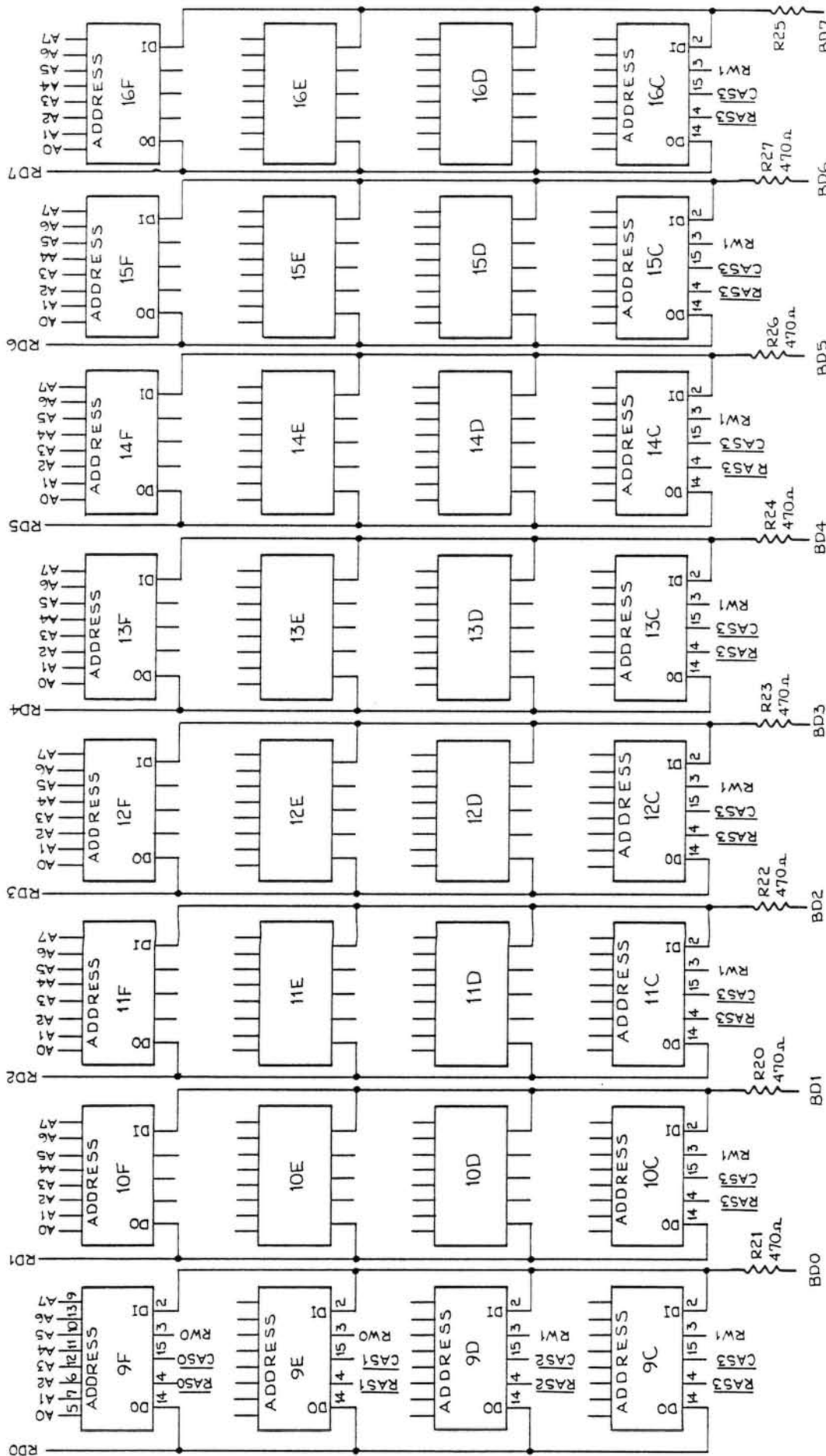
**Bus Interface**  
 DRAWN: QG REVISION: 2



# Q256-04 RAM Card



**Parity System**  
DRAWN: QG REVISION: 2



**Memory Array**  
DRAWN: QG REVISION: 2

ALL IC'S 6665  
N.B. Vcc +5V - PIN 8  
Vss GND - PIN 16  
TIE PIN 1 TO +5V



# QFC9

Floppy Disc Controller

# 2.5

Introduction.....	2.5.2
Address map.....	2.5.3
Commands.....	2.5.4
DMA address counter.....	2.5.4
DMA BYTE transfer counters.....	2.5.4
Data buffers.....	2.5.4
Address decoding.....	2.5.5
Controller LSI.....	2.5.5
DMA logic.....	2.5.5
Control register.....	2.5.5
Master oscillator.....	2.5.5
Write recompensation.....	2.5.6
Data Separator.....	2.5.6
Device driver ROM.....	2.5.6
Schematic diagrams.....	2.5.7

### Introduction

The floppy interface card interfaces the bit parallel/serial buss of floppy disk drives to the CMI's interleaved parallel buss. The interface is a combination of device driver software, controlling disk data format and initialization of data transfer parameters, and the hardware which carries out the transfers without processor intervention.

Data is stored on the floppy disk itself on its magnetic coating, in concentric rings. In a standard, 8 inch floppy there are 77 such rings on each side called tracks. Tracks are divided into data blocks called sectors. Sectors in Fairlight disk formats are either 128 or 256 bytes per sector, depending on operating system being used. The smallest amount of data that can be read to or from a disk is one sector. Sectors on a disk may be randomly accessed.

Track 0 is outermost. The controller automatically restores to this track on power on, and on reaching track 0, signals the controller by a mechanical switch generated signal. All head movement is relative to this reset state.

Floppy drives transfer data serially. They also have parallel control lines to control drive number selection, head stepping direction, head load, disk write and disk write enable. The head of the drive must be lowered to the disk surface before a transfer may take place, this is operation referred to as "head load". Index pulses are generated by the drive so that the controller knows the location of the rotating disk. This pulse occurs once per revolution, so the start of tracks can be determined by the controller. Also, the controller generates pulses that are used to step the head in and out to position it over the required track.

The Floppy Disk Controller/Formatter uses the WD1791 LSI controller. It is software selectable to double density, double sided in addition to single density, single sided. It is designed to work with CPU #2's, transferring data to and from memory by DMA on Processor 2. The processor is not involved with transferring data to and from the disk. Once a data transfer is set up the processor may continue processing other tasks until the interrupt for "command complete" is issued by the controller.

**Address Map**

(refer schematic QFC9-01)

The controller is accessed through two locations, in a memory map which enables access to peripherals. An address register is set up to point to the required controller register. All data is read or written through a single data register.

ADDRESS (HEX)	READ	WRITE
FCE0	data	data
FCE1	status register	address register

The 7 controller registers are:

00	control register
02	DMA address (low byte)
04	DMA address (high byte)
06	byte count to read/write (low byte, inverted)
08	byte count to read/write (high byte, inverted)
0A	command location to load device driver ROM into RAM WD1791 L.S.I
0C	cmd (write) status (read)
0D	track
0E	sector
0F	data

The definitions of the control register bits are:

0	DS0 drive select address bit 0
1	DS1 drive select address bit 1
2	enable interrupt (active high)
3	enable DMA address incrementing (active low)
4	DMA transfer direction (1= to disk)
5	side select
6	retrig head load timer
7	DENS density selection

The definitions of the control status bits are:

0	0
1	n/c
2	n/c
3	ready
4	two sided
5	disk change
6	interrupt
7	device driver loading (active low)

---

## QFC9 Floppy Disc Controller

---

### Commands

The extensive instruction set of the 1791 LSI can be obtained from the manufacturers data sheets for the 1791. This device handles all data conversions between the disk drive and the CMI buss.

### DMA address Counters

*(refer schematic QFC9-02)*

Sixteen bit counter chain C1 to C4 is used to provide the address for DMA transfers. The starting address for each disk transfer is established by writing the appropriate byte address to the address register then writing the address byte to the data register and then repeating for the other address byte. This causes the address to be preset into the DMA address counters by means of parallel-load strobe pulses STAL (low byte) and STAH (high byte). The incrementing of the DMA counters may be inhibited under software control, so that disk data may be dumped directly into the data portholes on channel cards.

### DMA Byte Transfer Counters

*(refer schematic QFC9-04)*

Sixteen bit counter chain C5 to C8 is used to transfer the required number of bytes to or from disk. It must be initialized with the inverse of the number of bytes to be transferred. Any number may be specified up to a maximum of 65,535 bytes. Only those bytes specified will be transferred to memory on a disk read. This allows less than a sector to be read from disk, and saves the software overhead required to handle partial sector reads. The read takes place but the buss VMA signal goes inactive after the required number of bytes have been transferred, so disabling memory writes. The VMA disable signal is generated from the ripple carry out on this counter chain, by buffering /FINPS, (Finished Partial Sector).

When a transfer occurs, the DMAC ( Direct Memory Access Claim ) line is generated so that the memory card swaps maps, allowing data to be dumped into memory currently not mapped into the processor's address space. This signal is generated by the components around flip-flop A11.

### Data Buffers

*(refer schematic QFC9-02)*

Data is propagated from the system data bus via latch B6 which hold the data across the processor 1 phase. This latched data also becomes the DATA FROM BUS via buffer B5, to the floppy-controller LSI.

Data written to the system control register at 00 is latched by B7. This controls such functions as drive select and DMA direction.

### Address Decoding

Address range \$FCE0-\$FCE1 is decoded by gates B1, E1, B2, E1 and latched by D2.

Address \$FCE0 is used to enable the internal data buss to read and write to controller functions.

Address \$FCE1 data is latched by B8 and with the access to FCE0 generates the internal chip selects and read/write strobes through C9.

Inverting buffer E5 and open collector drivers E6, E7 are used to interface the 1791 LSI controller to the disk drive cable. Incoming disk status signals are pulled up by 150 ohm terminating resistors.

### Controller LSI

The Interrupt Request from the LSI is gated with the Interrupt Enable to provide an open-collector interrupt signal for the system I.R.Q. on buss pin 63A.

### DMA Logic

*(refer schematic QFC9-03)*

Data requests from the 1791 or Device Driver rom loading are synchronised with Processor 2 Phase 2 using flip-flops C1 and A10. This sets up a DMA request to the processor (RDMA).

DMA cycles are granted by ACK acknowledge signal.

Flip-flop A11 only allows a DMA cycle to occur every second Processor cycle (the floppy drive can not transfer at that rate but this is a system constraint on other DMA devices in the DMA daisy chain).

The DMA daisy chain is controlled by /ENL and /EDL. Respectively these stand for, Enable Next Level and Enable Dma Level. When /EDL is active, a DMA request may be requested by the highest priority device. The /ENL signal informs the next device in the daisy chain that it may make a request if higher priority devices have not.

Depending on which function has been requested (Reset, Read, write ) the required DTB (Data to Bus ), and ATB ( Address to Bus) signals are issued.

### Control Register

The control register contains the drive number select bits, density selection, interrupt enable, increment DMA address enable, data transfer direction, side select and retrigger head load delay.

This register is the latch at B7.

The "retrig head" signal is used to reactivate the head load delay when the drive number has been changed, to allow for head bounce.

### Master Oscillator

*(refer schematic QFC9-05)*

The LSIs used on the card require a master 16MHz clock. This is generated with the components around the 74S04 at F5. The FDC9229 generates the 2 MHz clock for the 1791, by dividing this internally.

---

## QFC9 Floppy Disc Controller

---

### Write Precompensation

In double density operation there may be a time shift applied to the data when it is being written to the inner disk tracks (>45). The amount of shift is determined by the LSI and the floppy disk support device 9229. They produce a programmable delay of 1 to 3 clock cycles.

The amount, if any, is specified by the drive manufacturer. Links W4, W5, W6, W7 select the amount. The amount on inner and outer tracks can be independently set.

W5	W6	W7	Precompensation Value (in,nS)
0	0	0	0
0	0	1	62.5
0	1	0	125
0	1	1	187.5
1	0	0	250
1	0	1	250
1	1	0	312.5
1	1	1	312.5

The precompensation value is normally set to 0 on inner and outer tracks. It is more important on inner tracks as the bit density on the disk is greater.

W4 selects minifloppy drives and also requires the board to be made with a 34 pin connector (or a special cable) and an 8MHz crystal.

### Data Separator

The serial data stream that comes from the drive is in a synchronous form. It has embedded the required data as well as clock pulses and synchronization "marker bytes".

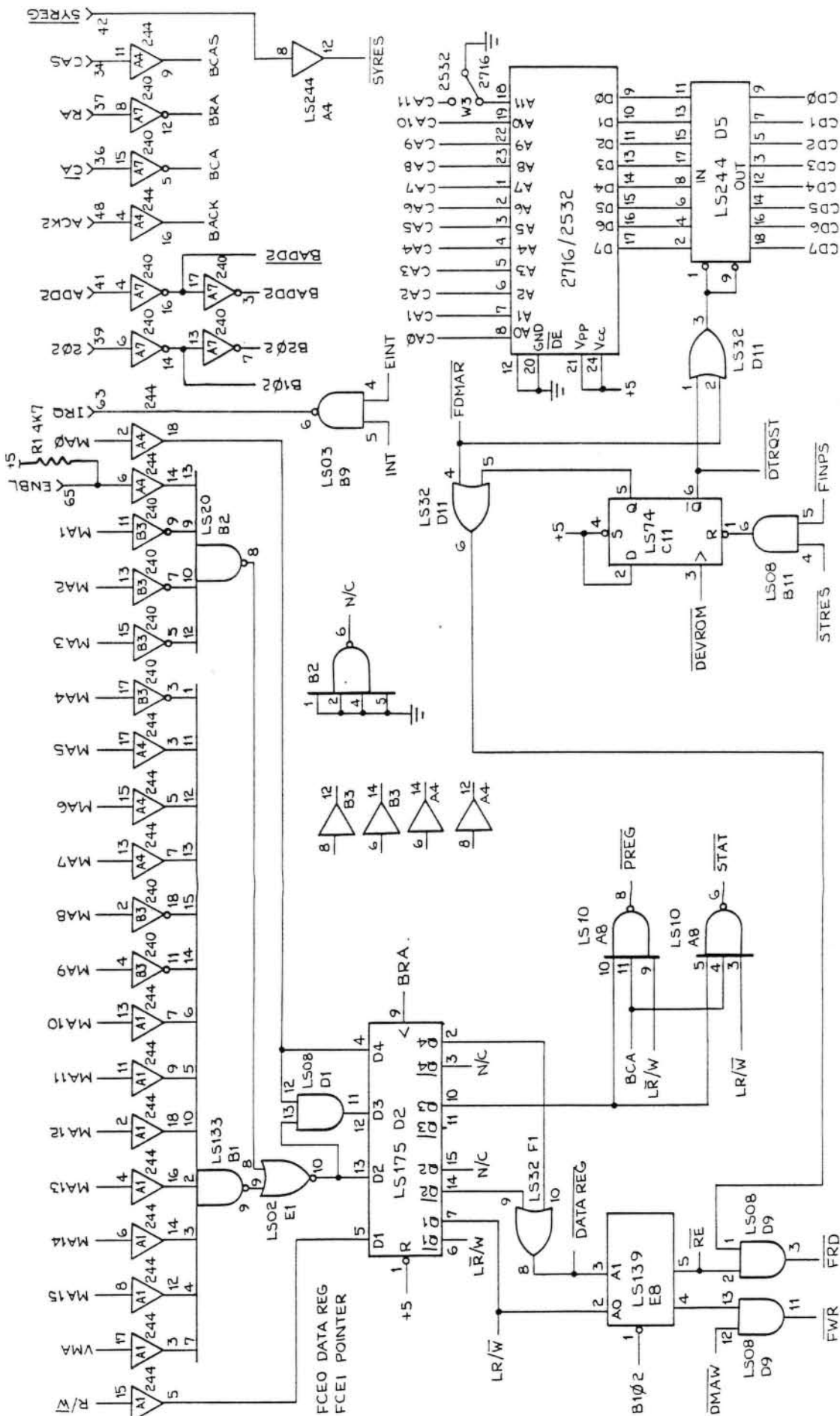
The data separator is used to generate the data window from the FM (single density) or MFM (double density) encoded READ DATA supplied by the disk drive. The separator tracks, so that incoming data is always in the center of the data window. This data window informs the controller chip which bits in the data stream it is receiving are data and which are clock bits just used in the data encoding scheme.

The separator is a digital phase locked loop in the FDC9229 chip at E10. This chip does all the work of data separation.

### Device Driver ROM

The disk controller software may be placed in a 2K or 4K EPROM on the controller card. This EPROM is not in the processor's directly addressable memory. It is executed by reading the software into RAM. This is done by DMA. The EPROM is copied into RAM as if reading a disk, except much faster.

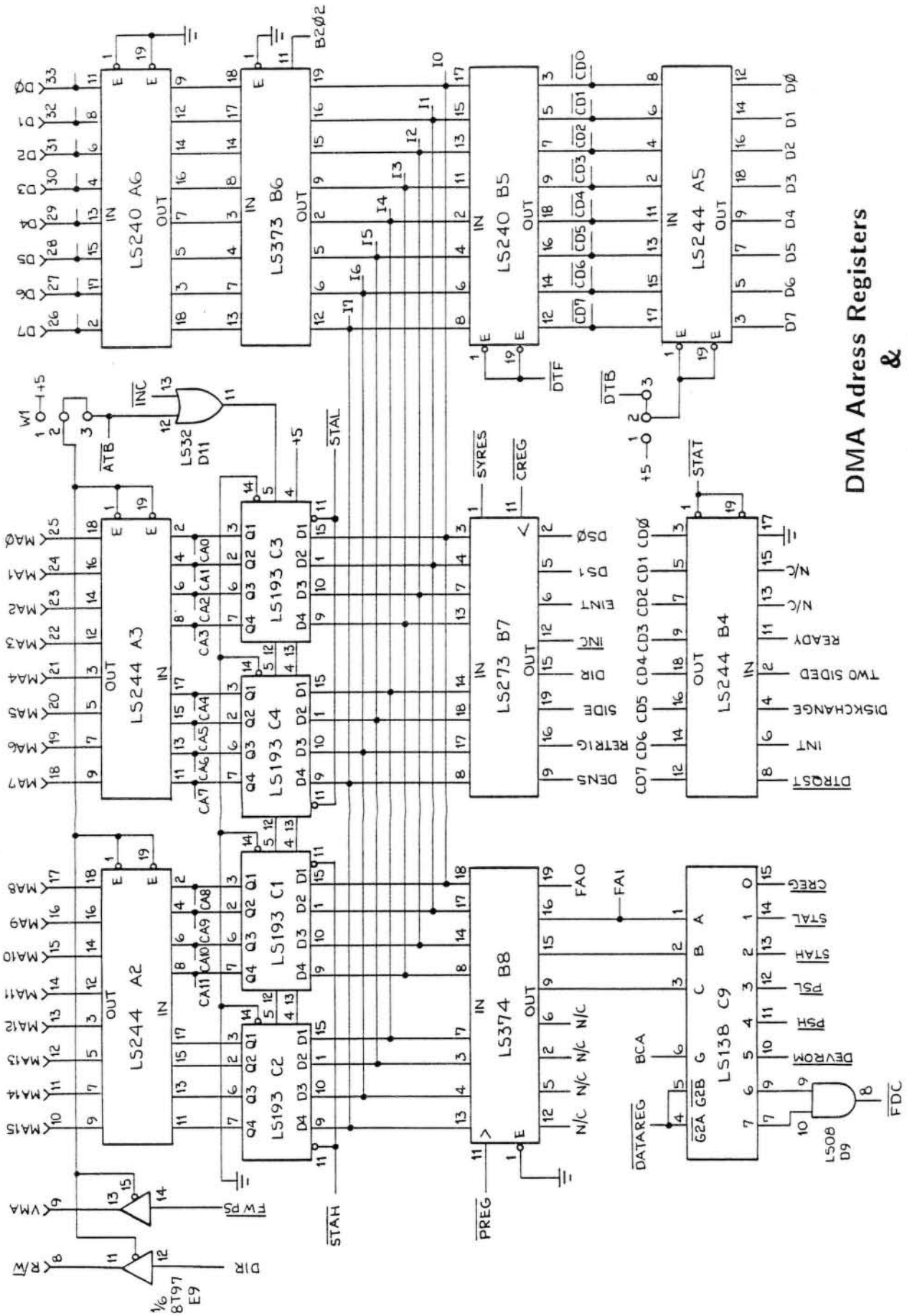
The least significant DMA counter lines are used as addresses on the EPROM, so the EPROM can only be loaded into memory on 2k or 4k boundaries. The flip-flop C11 and gates in D11 and B11 produce a DMA write to memory request that is terminated after the byte counter times out.



**Address/Control Buffers  
Devicer Driver ROM**  
DRAWN: A.B REVISION: 6

ALL UNMARKED NON INVERTING BUFFERS LS244  
" " INVERTING " LS240  
WITH PINS 1 AND 19 GROUNDED.

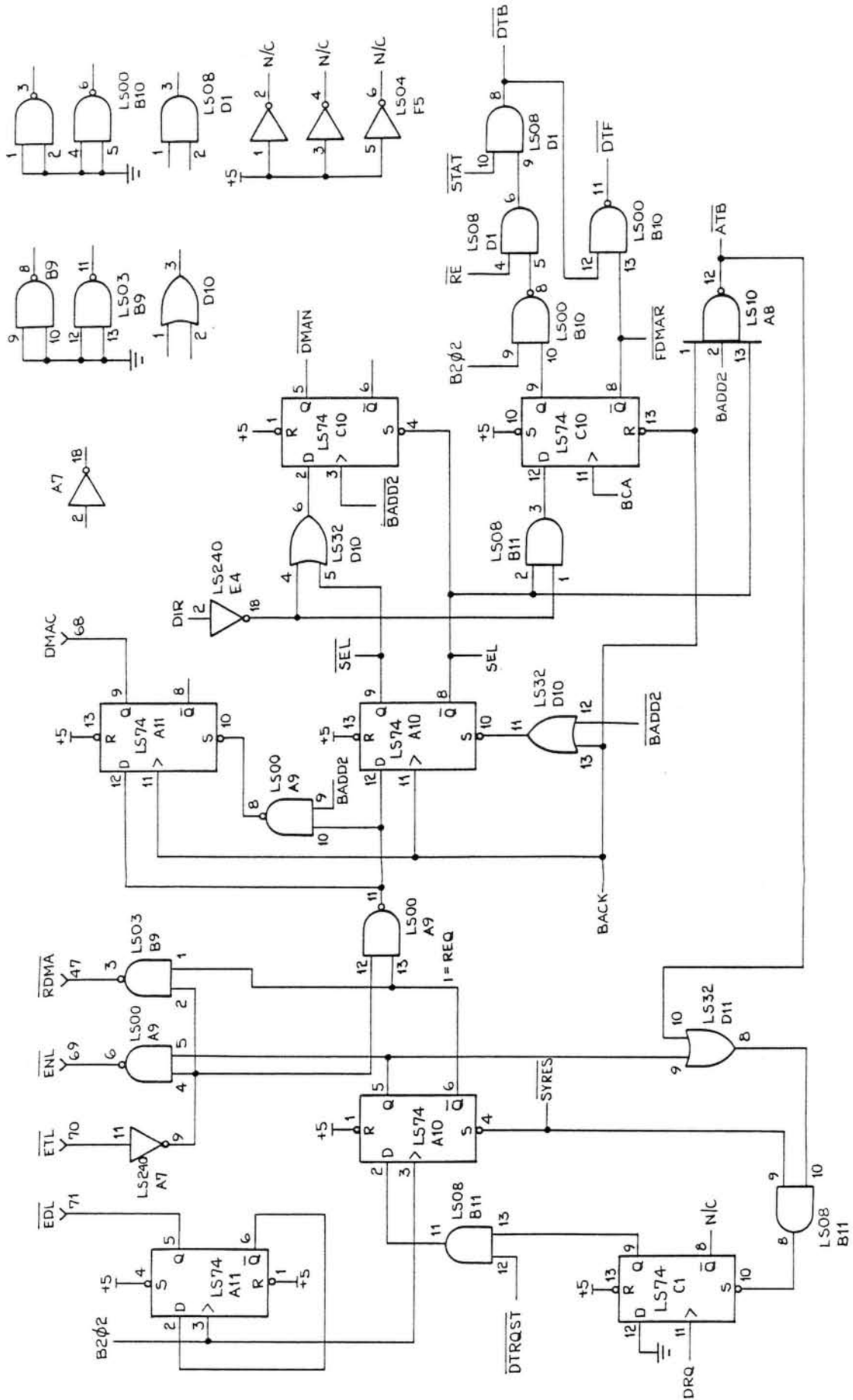
# QFC9-02 Floppy Disc Controller



## DMA Address Registers & Control Buffers

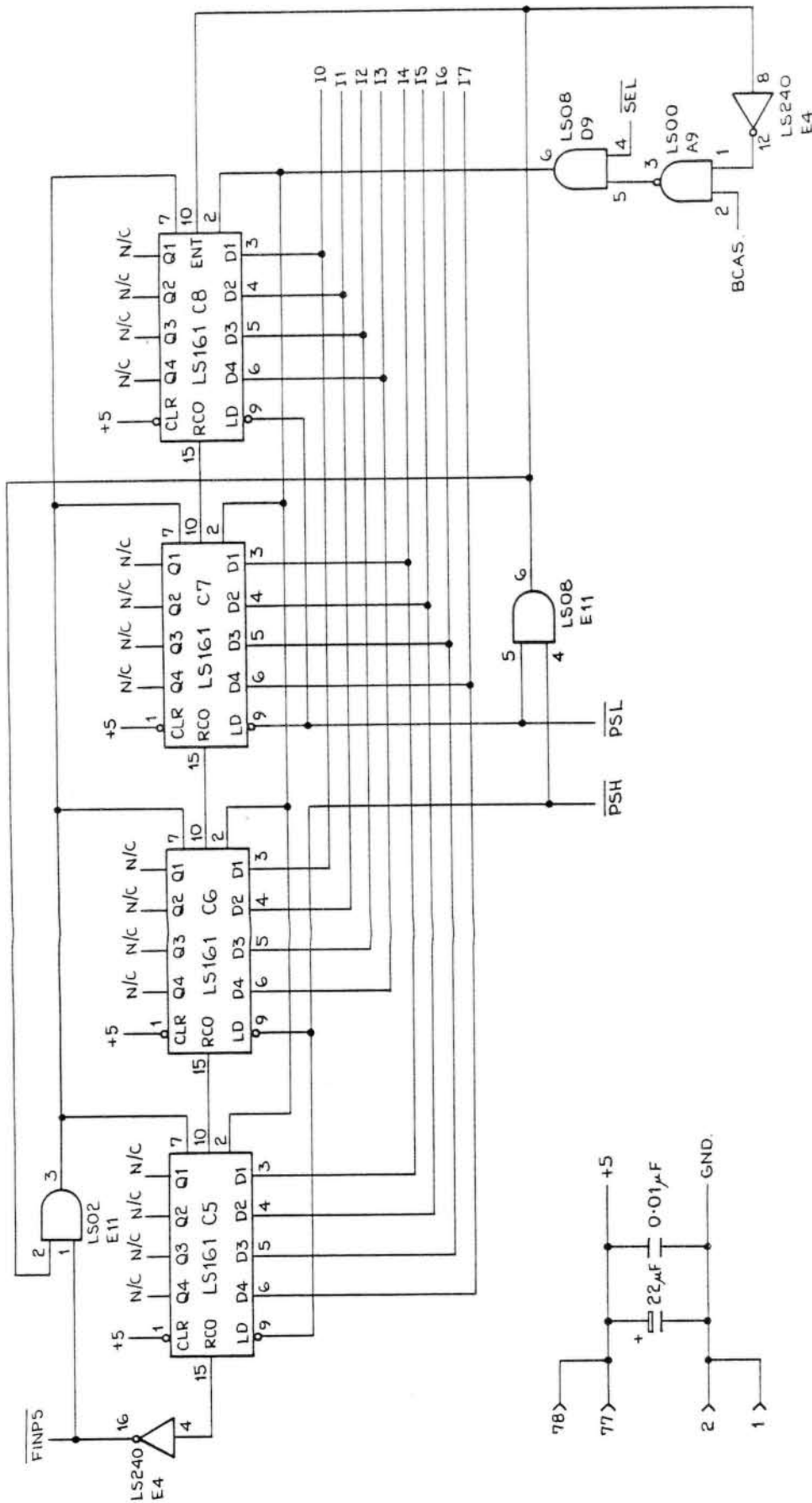
DRAWN: A.B REVISION: 6





DMA Logic Daisy Chain

DRAWN: A.B REVISION: 6



Byte Transfer Counter

DRAWN: A.B REVISION: 6



# QFC9-06 Floppy Disc Controller

1		2	GND
3		4	HDL3
5	READ DATA	6	HDL2
7	WRITE PROTECT	8	
9	TRACK Ø	10	
11	WRITE GATE	12	
13	WRITE DATA	14	
15	STEP	16	
17	DIRECTION	18	
19	DS4	20	
21	DS3	22	
23	DS2	24	
25	DS1	26	
27		28	HDL1
29	READY	30	
31	INDEX	32	
33	HEAD LOAD	34	HDLØ
35	ALT	36	IN USE
37	SIDE SELECT	38	
39		40	DISK CHANGE
41		42	TWO SIDE
43		44	
45		46	
47		48	
49	LOW CURRENT	50	

Cable Connector Signals

DRAWN: A.B REVISION: 6