```
     ****
     ****
     ****
    ******
   ** ** **
  **  **  **
**    **    **
```

Atari Corporation

WORKING DRAFT

Atari FALCON Product Specification

3 December 1991

Table of Contents

Section 1 INTRODUCTION

FALCON is the base design of a series of Atari computers extending the TOS compatible product line which began with the ST. FALCON provides enhanced video, graphics, and sound as well as greater bus bandwidth and operating speeds.

The FALCON architecture accepts 32-bit Motorola MC68030 or MC68040 family processors at clock speeds up to 33 MHz. Both processor families feature on-chip data and instruction caches which can be filled in bursts of 32 bit data fetches. The MC68040 also includes an internal floating point coprocessor. MC68030 based designs may include an external MC68882 floating point coprocessor.

The architecture also includes the industry standard VMEbus to facilitate expansion. The system supports the latest revision (C.1) of the VMEbus specification.

In order to function in a multisystem environment, FALCON contains an on-board moderate speed LAN port and an IO expansion port with DMA capability. Additionally, each FALCON has and internal modem port and two RS-232C serial ports.

Some highlights of the FALCON architecture:

- Motorola MC68030 or MC68040 up to 33 MHz
- Motorola MC68882 Floating Point Coprocessor (68030 only) at the CPU speed
- Two banks of dual-purpose (video/system) RAM, each bank consisting of 0.5, 2, or 8 megabytes, allowing up to 15 Mb (this memory appears 64-bits wide to the video logic and 32-bits wide to the system bus)
- Up to 32 Mb of fast expansion RAM
- 512Kb of 32 bit wide ROM
- Video modes that are a superset of those in the Atari ST and TT series-- Color: 320x200x16, 320x480x256, 640x200x4, 640x480x16 DuoChrome: 640x400x2. Monochrome: 1280x960x1
- New programmable video modes XxYxN where N can be 1,2,4,8 or 24 bits/pixel (ST and TT compatible modes are planar bit mapped while new modes are packed pixel)
- Programmable video timing to allow complete software control over sync rates and screen widths

- True broadcast (NTSC/PAL/SECAM) timing, true VGA, and higher resolution capability

- Powerful graphics coprocessor

- Parallel I/O port

- Internal audio amp and speaker

- Two asynchronous serial I/O ports (up to 19.2 Kbaud)

- High-speed SDLC serial modem and LAN ports (interface through a proprietary DMA channel)

- Real time clock (RTC) with 50 bytes of non-volatile RAM

- ST/MEGA compatible intelligent keyboard, with mouse and joystick ports

- Floppy disk controller and interface on a proprietary DMA channel (double, high, and quad density)

- Musical Instrument Digital Interface (MIDI)

- Atari ST compatible cartridge port (128 Kb storage)

- SCSI interface on a proprietary DMA channel

- Expansion IO port with DMA

- Multi-slot VME bus (full multi-master VME with the address space divided into A32/D32, A24/D16, and A16/D16 areas)

Section 2 MAIN SYSTEM

The FALCON architecture is designed to be a high performance computing platform. By including the VMEbus and facilities for multi-processing the system can be expanded for future needs.

2.1 Processor and MMU

FALCON accepts the Motorola MC68030 or MC68040 32-bit microprocessor. Each processor contains a paged memory management unit, and independent instruction and data caches. The 68030 and 68040 are complex instruction set computers (CISCs) that extend the 68000 instruction set and enhance the addressing modes. The processor can operate at clock speeds up to 33 MHz.

Both processors contain internal memory management units. Refer to the respective user manual for a complete description.

The on-chip instruction and data caches maximize processor throughput while reducing the bus bandwidth necessary to fuel the processor.

2.2 Floating Point Coprocessor

FALCON designs using MC68030 processors may include an external MC68882 floating point coprocessor. The MC68040 contains an internal FPU and does not support the coprocessor interface..

The floating point operations are performed in accordance with IEEE Standard 754, with both 32-bit (single) and 64-bit (double) precision external access.

The external floating point coprocessor in FALCON designs using the MC68030 is run at the same clock speed as the main processor. It appears as the "standard" floating point coprocessor ID of 01h in the 68030 CPU address space.

2.3 System Read only Memory (ROM)

The system includes 32 bit wide ROM providing up to 512Kb of ROM space. Jumpers are provided to allow the use of 27256, 27512, 27010/27C1001, and 57101/27C1000 EPROMs, in addition to 53100 ROMs. The default jumper position allows the use of 27512 EPROMs (for a total of 256 Kb of ROM) as well as 571001/27C1000 EPROMs or 531000 ROMs (for a total of 512 Kb of ROM). ROM cycle time is software selectable allowing use of ROMs with access times of 300ns to 100ns. The particular jumper configurations may vary with PCB design. Refer to specific system documentation for jumper settings.

An image of the first eight bytes of ROM appears at 0x00000000-0x00000007 accessible only in supervisor mode for system reset. Attempts to read from this area in user mode or any write results in a bus error. A VMEbus master would have to do privileged accesses to read the ROM at these locations. The full ROM begins at memory location 0xFFE00000.

Among the tasks this ROM must perform are system initialization, power-on diagnostics, and operating system boot.

2.4 System Random Access Memory (RAM)

The basic system must include one bank (0.5, 2, or 8 Mb) of dual-purpose RAM used for both video and system memory. Dual-purpose or video memory uses fast page mode DRAMs. Fast page mode is used to support both video refresh and system burst accesses. When video refresh is in progress, none of the dual-purpose RAM is available for the system so system performance can vary with the video mode selected. The video modes have different refresh bandwidth requirements. In general, the greater the video resolution and the more colors available, the greater the required refresh bandwidth. The effect of video on system performance will depend on the specific application. Code which avoids dual-purpose RAM or maintains a high cache hit rate will show less effect.

The memory control unit (MCU) supports two banks of dual-purpose RAM. Each bank can be 64Kbit, 256Kbit, or 1Mbit deep depending on the type DRAMs used. Each bank is 64 bits wide and all 64 bits must be installed. Either of the banks can support any of the three sizes of DRAMs. The following combinations are therefore possible:

Memory Depth		Total RAM
Bank 0	Bank 1	
64Kb	none	512Kb
none	64Kb	512Kb
64Kb	64Kb	512Kb+512Kb
256Kb	none	2Mb
none	256Kb	2Mb
256Kb	64Kb	2Mb+512Kb
64Kb	256Kb	512Kb+2Mb
256Kb	256Kb	2Mb+2Mb
1Mb	none	8Mb
none	1Mb	8Mb
1Mb	64Kb	8Mb+512Kb
64Kb	1Mb	512Kb+8Mb
1Mb	256Kb	8Mb+2Mb
256Kb	1Mb	2Mb+8Mb
1Mb	1Mb	8Mb+7Mb [1]

Table 2.1

[1] Note that IO space occupies the upper 1Mb of the 16Mb address space RAM would occupy so that 15Mb is the maximum amount of dual-purpose RAM possible. The upper 1Mb of 16Mb RAM cannot be used. Also note that in the high ST image (0xFF000000-0xFFFFFFFF), ROM will occupy an additional 1Mb limiting RAM in that image to 14Mb (see memory map).

Optional RAM modules allow additional single purpose expansion RAM. By eliminating the video timing constraints on this RAM, the memory appears faster, on average, to the processor. A maximum of 64 Mb total expansion RAM has been defined but signal loading limits this RAM to two banks or a pratical limit of 32 MB (using 4Mb deep parts). The single-purpose memory system will support FBUS line transfers but not FBUS wide mode.

Additional memory can be installed in the system by plugging in VME memory cards. If A32/D32 cards are used, the VME RAM can be contiguous with single purpose expansion RAM. The VME RAM cards will appear slower than the expansion RAM as all VME accesses incur extra wait states per bus cycle and do not support line transfers.

There is no provision for parity or ECC protection on the RAM. The reliability of current DRAM technology makes this unnecessary. However, such features could be included in VME cards.

RAM on the system board is accessible from the VMEbus as bytes, words, or double words.

The first 0x800 bytes (2Kb) of RAM (0x00000008-0x000007FF, and 0xFF000000-0xFF0007FF) are accessible only in supervisor mode. Attempts to read or write to this area in user mode results in a bus error. VMEbus masters must do privileged accesses to use this RAM.

2.4.1 Memory Control and Configuration Registers

2.4.1.1 Main Configuration Register

Address xxFF8001:

D7 - ROM cycle time

0 = slow
1 = fast

ROM Access Time	System Clock Frequency			
	25Mhz	33Mhz	40Mhz	50Mhz
300ns	slow	slow	na	na
250ns	slow	slow	na	na
200ns	slow	slow	slow	na
150ns	fast	slow	slow	slow
120ns	fast	fast	slow	slow
100ns	fast	fast	slow	slow

D6 - Video Memory DRAM Access Speed

0 = slow
1 = fast

DRAM Access Time	System Clock Frequency			
	25Mhz	33Mhz	40Mhz	50Mhz
100ns	fast	slow	slow	na
80ns	fast	fast	slow	slow
70ns	fast	fast	fast	slow
60ns	fast	fast	fast	fast

D5 - Fast Memory DRAM Access Speed

0 = slow
1 = fast

DRAM Access Time	System Clock Frequency			
	25Mhz	33Mhz	40Mhz	50Mhz
100ns	fast	slow	slow	na
80ns	fast	fast	slow	slow
70ns	fast	fast	fast	slow
60ns	fast	fast	fast	fast

D4 - Not used (always reads 0)
D3 - D0 Not used (reserved read/write bit)

2.4.1.2 Refresh Control Registers

The MCU defaults to a 15.5 us maximum refresh interval after a reset. This corresponds to the most common refresh rate for currently available DRAMs, e.g. 512 row/8 ms for 256k deep parts, and 1024 row/16 ms for 1M deep parts. Refresh cycles can then be customized under software control. When the counter is enabled with a time constant of zero, refresh is turned off.

The fixed clock to the refresh control counter runs at 2 MHz. The default refresh interval corresponds to a value of 001D loaded into the counter. The minimum value of 1 in the counter provides a refresh interval of 1500ns. The maximum value of 7FFF provides a refresh interval of 16.384 ms.

Address = xxFF8003:

```
D7 - Refresh Interval Control
          0 = Default Interval
          1 = Counter
D6 - D0   Refresh Time Constant Bits 14-8
```

Address = xxFF8005:

```
D7 - D0   Refresh Time Constant Bits 7-0
```

2.4.1.3 External Cache Control Register

Address = xxFF8007:

```
D7 -           Reset Cache Tag SRAM
          1 = Cache enabled
          0 = Reset cache
D6 - D2   Not used (always reads 0)
D1 -           Capture Data Cache Push Access
          1 = yes
          0 = no
D0 -      Clear Cache on Change of Bus Master
          1 = no
          0 = yes
```

2.4.1.4 Video Memory Configuration Register

Address = xxFF8009:

D7 - D4   SIMM Speed Select bits (read only)

SIMM Speed Select	DRAM access time
0000	
0001	80ns
0010	
0011	
0100	
0101	70ns
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	100ns
1110	
1111	

D3 - D2   Bank 1 Size Select

D1 - D0   Bank 0 Size Select

Size Select Bits	Bank Size (DRAM Depth)
00	not installed
01	512Kb (64K)
10	2Mb (256K)
11	8Mb (1M)

2.4.1.5 Expansion (fast) Memory Configuration Register

Address = xxFF800B:

D7 - D4   SIMM Speed Select bits (read only)

SIMM Speed Select	DRAM access time
0000	
0001	80ns
0010	
0011	
0100	
0101	70ns
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	100ns
1110	
1111	

D3 - D2   Bank 1 Size Select

D1 - D0   Bank 0 Size Select

Size Select Bits	Bank Size (DRAM Depth)
00	not installed
01	1Mb (256K)
10	4Mb (1M)
11	16Mb (4M)

2.5 Interrupt Control

The IO Control Unit (IOCU) provides an additional level of interrupt control for the system as well as the interface for an internal IO bus and decoding for the internal peripheral circuits. It contains registers that allow the software generation of interrupts. All of the IOCU registers are reset at power-on, but not by the reset push button or a processor reset instruction.

2.5.1 Interrupt Mask and Current Status

The IOCU contains two mask registers that permit independent control over which interrupt levels will be seen by the processor. One register masks interrupts generated on the system board and the other masks VMEbus interrupts. These registers are cleared at power-up, disabling all interrupts. The state of these registers is not affected by the reset button.

There are also system and VME interrupt request registers that show the current state of the seven interrupt request levels from each. These registers show the physical state of the interrupt lines before they are AND'd with the IOCU's mask registers.

The system board sources for IRQ5 and IRQ6 can be serviced by either the CPU or a VMEbus master. IRQ5 and IRQ6 look to the CPU like VME interrupts, and can not be masked independently of VME level 5 and 6 interrupts by the IOCU system board interrupt mask register.

2.5.2 System Control Registers

The IOCU also contains two read/write registers that can be used for system configuration information. Since these registers are only reset at power-on, their contents can be used across system resets.

2.5.3 Interrupt Generator

The system can write to an I/O address to generate a low priority (level 1) interrupt to the CPU. This I/O address contains a read/write status/control port, only the least significant bit is defined. When set to 1, it generates an autovectored level 1 interrupt. When cleared, the interrupt request is taken away.

The IOCU is configured so that:

- only system interrupts 5 and 6 and VME interrupts are capable of generating vectored interrupts to the CPU

- IOCU generated level 1 and 3 interrupts are always autovectored

- The IOCU generated level 1 interrupt can be detected only by the CPU, not by a VMEbus master

- VMEbus SYSFAIL generates a system level 7 interrupt, but does not generate a VME level 7 interrupt.

2.6 Bus Timer

The MCU implements a system bus timer. Bus cycles not terminated within about 256us will cause a bus error. The MCU will compensate for delays associated with video memory.

2.7 Real Time Clock

The FALCON system includes a Motorola MC146818A like real time clock function. This provides time of day (down to one second resolution), date, and a programmable periodic interrupt. The RTC is provided with a 32.768 kHz oscillator that is independent of all other system clocks.

The interrupt output of the real time clock chip connects to one of the MFP parallel inputs.

The circuit also includes 50 bytes of battery backed up (non-volatile) RAM that is used for storing diagnostic and configuration data.

The control registers are accessed through two byte ports. The first byte is write-only and used to set the register address desired. The other byte is the read/write data port. When doing a write to a register, it is possible to do a long word write; the long word would contain both the address and the data. The IOCU will break the write into two transfers in the correct order for the RTC circuit.

Section 3 IO Channels

The FALCON architecture supports the following IO channels:

- SCSI (as defined by the ANSI X3T9.2 committee)

- floppy disk interface with DMA channel

- a modem port and LAN port through the SCC chip

- a slot for a network card

- two asynchronous serial ports and an interrupt control through two MFP controllers (MC68901)

- a parallel printer port

- a ST/MEGA compatible intelligent keyboard, mouse, and joystick interface

- a ST compatible cartridge port supporting application and diagnostic cartridges

3.1 DMA Controllers

The FALCON design includes four independent DMA channels:

1) the AUX port (includes the SCC and network)

2) the SCSI port

3) the ST floppy disk port

4) digital sound playback and record.

Additionally, the VMEbus interface permits a VMEbus master to perform DMA into system memory. The following is the DMA bus mastership priorities:

priority	function
highest	SCSI DMA Channel
	Aux DMA Channel
	Floppy disk DMA channel
	Digital sound DMA channel
	VMEbus Masters
lowest	CPU

3.1.1 AUX/SCC and SCSI DMA Channels

The AUX/SCC and SCSI DMA controllers assemble the bytes from the peripheral into longwords for writing to the system bus. DMA can be done to any byte boundary, either on the main system board or on the VMEbus. DMA is done in physical address space.

The programmer’s model of each of these DMA channel consists of:

- a byte wide read/write status/control register that contains direction, enable and bus error bits
- four bytes forming a 32-bit DMA pointer
- data residue register that must be merged with RAM contents under CPU control if the DMA input is done to a point in RAM that is not on a longword boundary or if DMA is not done in multiples of four bytes
- four bytes forming a 32-bit wide DMA byte count

The software that sets up the DMAC for DMA transfers must account for the DMA registers being a byte-wide and appearing at odd byte addresses. This requires the CPU either to use the MOVEP instruction or to do rotates and four separate byte output operations to put out a 32-bit address or byte count.

DMA Controller Registers

offset	width	function
Ox00	OB	DMA Pointer Upper
Ox02	OB	DMA Pointer Upper-Middle
Ox04	OB	DMA Pointer Lower-Middle
Ox06	OB	DMA Pointer Lower
Ox08	OB	Byte Count Upper
Ox0A	OB	Byte Count Upper-Middle
Ox0C	OB	Byte Count Lower-Middle
Ox0E	OB	Byte Count Lower
Ox10	W	Data Residue Register High
Ox12	W	Data Residue Register Low
Ox14	OB	Control Register

The control register bit-map:

bit	function
0	DMA Direction Out (1 = out to port)
1	Enable (0 = off, 1 = on)
2	SCC channel (0=A, 1=B) Aux/SCC channel only
3	Aux/SCC select (1=aux, 0=SCC) Aux/SCC channel only
4	<reserved>
5	data under/overrun
6	Byte Count Zero (1 = terminal count)
7	Bus Error (1 = Bus Error occurred during DMA by this channel)

To perform DMA:

1) set the DMA controller direction
2) set the base address
3) set up the peripheral for DMA
4) then set the enable bit

The direction and enable bits should not be set in the same operation. If DMA input is done to anything but a longword aligned destination, or if the length is not a multiple of four, the final byte(s) of the transfer will not be written to the system RAM. It is then the programmer’s responsibility to read the Data Residue Register and merge the input with the contents of the appropriate longword in RAM. (The least significant two bits of the DMA pointer are correctly incremented, which can be used to determine how much of the Residue Register is valid.) For best system performance, software should try to maintain DMA operations on longword boundaries and keep byte counts in multiples of four.

If an attempted DMA operation generates a bus error, DMA operation is immediately disabled and the bus error bit set in the Control/Status register. The bus error status bit generates an interrupt. The interrupt output of both of the SCSI and SCC DMA controllers are OR’d together and connected to one of the MFP input bits where they can be read or optionally used to generate a processor interrupt. The bus error status for a channel is automatically cleared by reading the channel’s control register.

The DMA byte count register generates an interrupt when the byte count reaches 0. The DMA is automatically disabled by reaching the terminal count.

The 5380 SCSI Interface Chip must not be used in its BLOCK MODE DMA. The SCC should be in programmed to use the WAIT/*REQ pin in *REQ mode when doing DMA.

The AUX channel controls the SCC and network slot. DMA can transfer data to the SCC A port, SCC B port, or network slot. Only one of the ports can be accessed via DMA at a time.

3.1.2 SCSI Output

FALCON implements the complete single-ended (non-differential) SCSI bus using a 5380 SCSI Controller. The 5380 is used in its 8-bit asynchronous data transfer mode up to 4.0 Mb/second, adequate for current disk drives.

The external SCSI connector provides for connection of SCSI compatible devices through a 50 pin SCSI II connector.

External SCSI Connector					
Pin	Signal	Pin	Signal	Pin	Signal
1	GND	2	GND	3	GND
4	GND	5	GND	6	GND
7	GND	8	GND	9	GND
10	GND	11	GND	12	reserved
13	nc	14	reserved	15	GND
16	GND	17	GND	18	GND
19	GND	20	GND	21	GND
22	GND	23	GND	24	GND
25	GND	26	DB0*	27	DB1*
28	DB2*	29	DB3*	30	DB4*
31	DB5*	32	DB6*	33	DB7*
34	DBP*	35	GND	36	GND
37	reserved	38	termpwr	39	reseved
40	GND	41	ATN*	42	GND
43	BSY*	44	ACK*	45	RST*
46	MSG*	47	SEL*	48	C/D
49	REQ*			50	I/O

Table 3.1

Devices connected to the external SCSI connector should provide standard SCSI bus termination in the last physical device.

In a typical configuration, the SCSI bus will be used to provide the main mass storage elements of the system. The SCSI bus can also be used for removable media devices such as the Syquest cartridge drives and magnetic tape controllers.

The SCSI bus can support up to seven major devices.

3.2 Floppy Interface

The floppy disk DMA channel is fully ST compatible. It provides a port to the 1772 like floppy disk controller (FDC). The DMA channel operates identically with the ACSI/Floppy DMA channel of previous ST architectures, except there is no ACSI port and therefore no external devices accessible. For a further description of this DMA channel, see the Atari ACSI/DMA Integration Guide.

A register is provided to control the floppy density similar to the TT. FALCON enhances the function of this register to enable sensing and control of extended (quad) density floppy drives.

The floppy disk density select register (IO+860Fh) provides control of disk density. Bits 4 and 0 are used to select the frequency of the clock sent to the floppy controller circuit. The remaining bits control two outputs and provide two inputs which may be used (TBD) in the density selection process. The disk change signal has also been added.

The FALCON floppy disk subsystem is designed around a WD1772 like Floppy Disk Controller supporting up to two daisy-chained floppy disk drives. The interface can support double, high, and quad density drives.

The internal drive cabling supports the disk change signal from the floppy drive(s). The signal is asserted when power is applied or a diskette is removed from the drive. The signal is cleared by issuing a step command to the drive with a disk inserted.

3.3 Serial and Modem Ports

The Zilog 85C30 SCC, a dual channel, multi-protocol data communications peripheral, is included in FALCON to provide the modem and LAN ports.

The input/output of SCC channel A is routed through RS-423 level converters to the LAN connector, an 8-pin mini-DIN connector.

The SCC handles both asynchronous formats and synchronous byte-oriented protocols such as HDLC and IBM’s SDLC.

The SCC port B is connected to an internal modem port. Modem control signals are derived directly from the 85C30 port B control lines. This port can operate with split transmit and receive baud rates.

The PCLK input to the SCC is 8 MHz. The RTxCA input is provided with a 3.6864 MHz clock. The input to TRxCA comes from the low speed LAN connector. RTxCB is run at 2.4576 MHz. TRxCB is generated by the Timer C output of the second MFP.

3.3.1 Internal Modem Port Pinout

SCC Modem Port Pinout	
Pin	Signal
1	Carrier Detect (CD, input)
3	Receive Data (RD, input)
5	Transmit Data (TD,output)
7	Data Terminal Ready (DTR,output)
9	Data Set Ready (DSR,input)
11	Request to Send (RTS,output)
13	Clear to Send (CTS,input)
15	Ring Indicator (RI,input) [1]
14	Microwire Enable (MWEN*)
16	Microwire Data (MWD)
18	Microwire Clock (MWC)
17	Audio (from phoneline)
19	Audio (to phoneline)
20	Audio ground
2,4,6,8	logic ground
10,12	VCC (+5V)

Table 3.2

The modem connector is a 20 pin card edge. It is provided to accept an internal card for adding modem and related functions. All signal levels are TTL. Cards should provide no more than 1 TTL load per pin or 50pf. Output drive is sufficient to drive 2 TTL loads.

[1] The modem port ring indicator (RI) signal is connected to bit 3 of the MFP-2 General Purpose I/O Port (GPIP).

3.3.2 LAN Connector Pinout

The LAN connector is an 8 pin female mini-DIN.

SCC Port A LAN Pinout	
Pin	Signal
1	Output Handshake (DTR, RS-423)
2	Input Handshake/External Clock
3	Transmit Data -
4	Ground
5	Receive Data -
6	Transmit Data +
7	<reserved>
8	Receive Data +

Table 3.3

3.4 Expansion IO Port with DMA

The auxiliary DMA channel provides an interface for a expansion IO port to accept high speed data transfer modules (such as ethernet). The interface allows the module to transfer data via the DMA channel to or from the Falcon bus and provides memory mapped access from the Falcon bus to the module.

Data transfer takes place via an eight bit bi-directional data bus. There are two types of transfer cycles, IO and DMA. IO cycles are controlled by separate read and write strobes (IOR* and IOW*) and chip select (CS*). Four address lines (A0-A3) are provided for selection of registers. DMA cycles assert an acknowledge signal (DACK*) with the read or write strobe and do not assert the chip select. The address lines are not used during DMA cycles. IO cycles are always initiated by the Falcon bus master. DMA cycles are initiated by the device via the DMA request signal (DRQ). An interrupt signal (IRQ*) may be included to allow the device to signal the processor. The interrupt signal, if used, must only be driven low as it may be part of a wire-or structure. The device should drive the data lines ONLY when IOR* AND either (CS* or DACK*) are true.

3.4.1 Expansion IO Port Pinout

Expansion IO Port			
pin	signal	pin	signal
1	A0	2	GND
3	A1	4	GND
5	A2	6	VCC
7	A3	8	VCC
9	CS*	10	DRQ
11	IOR*	12	DACK*
13	IOW*	14	IOACK*
15	D0	16	D1
17	D2	18	D3
19	D4	20	D5
21	D6	22	D7
23	IRQ*	24	RESET*
25	SGND	26	12v

Table 3.4

The connector will be a 26 pin card edge.

3.4.2 Signal Description

A0-A3 Outputs from Falcon select one of sixteen registers during IO cycles.

CS* Output from Falcon is low during IO cycles. Devices should ignore IOR* and IOW* when CS* and DACK* are high.

IOR* Output from Falcon is low during IO and DMA cycles when data is transferred from the device to Falcon. The data lines are input by Falcon when IOR* is low.

IOW* Output from Falcon is low during IO and DMA cycles when data is transferred to the device from Falcon. The data lines are output by Falcon when IOW* is low.

D0-D7 Bi-directional data lines are used to transfer data between Falcon and the device. The direction is indicated by IOR* and IOW*.

DRQ Input to Falcon requests a DMA transfer when high.

DACK* Output from Falcon is low during DMA cycles. Devices should ignore IOR* and IOW* when CS* and DACK* are high.

RESET* Output from Falcon is low during system reset. (minimum width of reset pulse is 10us)

IRQ* (TBD) Input to Falcon can generate an interrupt to the processor when driven low. Should only be driven low. There will be a 2.2K pull up in Falcon.

IOACK* Input to Falcon for handshake of IO cycles. IO cycles are extended indefinately while IOACK* is high. IO cycles terminate when IOACK* is low.

VCC +5 volts +/- 5% 1a

GND Logic ground

SGND (TBD) Shield ground

12V +12 volts +/- 5% 100ma

3.5 MFP

Two 68901 Multi-Function Peripheral (MFP) controllers are used to provide system timers, RS-232C serial ports, and interrupt controllers. One MFP, designated MFP-ST, is used in a way that is compatible with the ST. It provides both a serial port and interrupt control. A second MFP provides another serial port and more I/O and interrupt pins.

The baud rate clock for the MFPs serial transmitter and receiver is derived from the timer D output of each MFP. Given the MFPs' 2.4576 MHz clock, baud rates up to 19.2 Kbaud can be supported on these serial ports.

3.5.1 MFP Serial Port Pinouts

Both MFP serial ports are pinned out on DB-9P connectors in a way that is compatible with most PCs. One of the MFP serial ports has a complete complement of modem control lines compatible with the ST, but pinned out in a 9 pin D connector. The other MFP serial port will have modem control via the GP IO register. Note that a DSR signal has been added to the ST compatible serial port.

MFP Serial Port Pinouts		
Pin	MFP-ST	MFP-2
1	Carrier Detect (CD,input)	Carrier Detect (CD,input)
2	Receive Data (RD,input)	Receive Data (RD,input)
3	Transmit Data (TD,output)	Transmit Data (TD,output)
4	Data Terminal Ready (DTR,output)	Data Terminal Ready (DTR,output)
5	ground	ground
6	Data Set Ready (DSR, input)	Data Set Ready (DSR, input)
7	Request to Send (RTS,output)	Request to Send (RTS,output)
8	Clear to Send (CTS,input)	Clear to Send (CTS,input)
9	Ring Indicator (RI,input) [1]	Ring Indicator (RI,input)

Table 3.5

[1] The Ring Indicator (RI) signal is connected to bit 6 of the MFP-ST General Purpose I/O Port (GPIP).

The GP IO register serial port bits are defined as follows:

bit 8	ro	CD	state of MFP-2 serial port pin 1
bit 9	ro	DSR	state of MFP-ST serial port pin 6
bit 10	ro	DSR	state of MFP-2 serial port pin 6
bit 11	ro	CTS	state of MFP-2 serial port pin 8
bit 12	ro	RI	state of MFP-2 serial port pin 9
bit 13	wo	DTR	sets MFP-2 serial port pin 4
bit 14	wo	RTS	sets MFP-2 serial port pin 7

3.5.2 Uncommitted I/O Pins

The least significant two bits of the General Purpose I/O Port on MFP-2 are not currently used and are routed to stakes for convenience. These are simple unbuffered TTL level signals that can be used for either input or output.

3.6 Parallel Printer Port

The FALCON includes a bi-directional 8-bit parallel printer port similar to most PCs. The data interface is through the programmable sound generator (PSG) chip IO port B. It is pinned out to a DB-25 connector. The Centronics STROBE signal is generated from the PSG IO port A. The BUSY signal from the printer is routed to one of the parallel input lines of the MFP to permit interrupt driven printing. The GP IO register at IO+8804h provides the remaining signals.

Parallel Port Pinout	
1	Strobe (STB-)
2	Data0
3	Data1
4	Data2
5	Data3
6	Data4
7	Data5
8	Data6
9	Data7
10	Acknowledge (ACK-)
11	Busy (BUSY-)
12	Paper out (PE)
13	Select (SLCT)
14	Autofeed (AFD-)
15	Error (ERROR-)
16	Init (INIT-)
17	Select In (SLCTIN-)
18-25	Ground

Table 3.6

The GP IO register parallel port bits are defined as follows:

bit 0	ro	ERROR-	state of pin 15
bit 1	ro	SLCT	state of pin 13
bit 2	ro	PE	state of pin 12
bit 3	ro	ACK-	state of pin 10
bit 4	wo	AFD-	sets the output state of pin 14
bit 5	wo	INIT-	sets the output state of pin 16
bit 6	wo	SLCTIN-	sets the output state of pin 17

3.7 Keyboard Interface

The FALCON keyboard interface is completely compatible with the ST/MEGA computers. The keyboard is equipped with a combination mouse/joystick port and a joystick only port. The keyboard transmits encoded make/break key scan codes (with two key rollover), mouse/trackball data, joystick data, and time-of-day. The keyboard receives commands and sends data via bidirectional communication implemented with a MC6850 Asynchronous Communications Interface Adapter (ACIA). The data transfer rate is 7812.5 bits/second. (See the Atari, Intelligent Keyboard (ikbd) Protocol, February 26, 1985.)

Additional circuitry has been included to support flow control of keyboard data. The keyboard may monitor the IKBD uart receive interrupt to inhibit sending data when it is active. The keyboard may also inhibit the IKBD uart transmit clock via a control signal to pause the data flow from FALCON.

3.8 ROM Cartridge

The FALCON cartridge port is fully compatible with ST cartridges. The cartridge is physically connected through a 40 pin card edge connector ROM cartridge slot. Cartridge ROMs are mapped to a 128Kb memory region starting at 0x00FA0000, extending to 0x00FBFFFF (with an image at 0xFFFA0000 to 0xFFFBFFFF).

Section 4 Video Subsystem

The FALCON video subsystem is designed to extend the existing ST and TT modes. Additional modes are available on the FALCON that allow more colors and larger screen sizes. This subsystem is one of the basic components required to support the industry standard X Windows windowing system allowing the FALCON to exist as a fully-compatible X Windows workstation.

4.1 Video Configuration

The various modes available on the FALCON are:

mode register	resolution	planes	palette	colors/DAC
xxFF8260	ST modes			
00	320x200	4	16	512/3bit
01	640x200	2	4	512/3bit
10	640x400	1	2	512/3bit
xxFF8262	TT modes			
000	320x200	4	16	4096/4bit
001	640x200	2	4	4096/4bit
010	640x400	1	2	4096/4bit
100	640x480	4	16	4096/4bit
110	1280x960	1	monochrome	4096/4bit
111	320x480	8	256	4096/4bit

Table 4.1

mode register	resolution	bits/ pixel	palette	colors/DAC
xxFF8268				
000	XxY	1	monochrome	16M/8bit
001	XxY	2	2	16M/8bit
010	XxY	4	16	16M/8bit
011	XxY	8	256	16M/8bit
101	XxY	24	--	16M/8bit
110	XxY	8/24	--	16M/8bit
100	XxY	4	16	4096/4bit
111	XxY	1	monochrome	4096/4bit

Table 4.2

The modes are set through the respective (ST, TT, or FALCON) video mode register. In the ST mode, 16 word-wide registers comprise the ST color palette (also known as the Color LookUp Table - CLUT). Contained in each entry are nine-bits of color: 3-bits each for red, green, and blue. Therefore, a total of 512 possible color combinations (8 x 8 x 8) are selectable for each entry. Through bank select bits in the TT mode register, 16 banks of 16 ST CLUT registers can be mapped into the ST CLUT address space.

Mode 00 (320x200x4) can index all sixteen ST palette colors, while mode 01 (640x200x2) can index just the first four (Reg0 - Reg3) palette colors. The duochrome mode (10 - 640x400x1) uses two entries of the TT color palette (foreground, Reg255 and background, Reg254) and is provided with an invertor for inverse video controlled by bit 0 of the ST palette Reg 0 or bit 1 of the TT palette register 0. Color palette 0 is also used to assign a border color while in multi-plane mode.

Additional resolution modes are available by programming the video through the TT shift mode register. In these modes, there are a maximum of 256 TT color palette registers each containing 12-bits of color: 4-bits each for red, green, and blue. Therefore, a total of 4096 possible color combinations (16 x 16 x 16) are selectable. Through the ST palette bank (lowest 4 bits of the TT Shift Mode Register) one of 16 banks may be selected from the TT color palette for use in ST modes. This allows modes 000, 001, 010, and 100 to seemingly select from up to 256 registers by simply setting the palette bank. Only mode 111 (320x480x8) can index all 256 registers.

All accesses to either the ST or TT shift mode registers will program the RAMDAC for the appropriate video mode Likewise, all accesses to either ST or TT color palettes will update the RAMDAC color look-up tables appropriately.

It should be noted that even though the ST, TT, and FALCON color palettes are referenced as if they are separate entities, they are actually implemented as different access paths to the same physical storage. The color palette memory physically exists in the RAMDAC as a 256x24 static RAM. When one of the 16 ST palette registers is accessed, one out of the 256 physical registers selected and data steering is enabled to map each of the three 3-bit color definitions into the 24-bit register in such a way as to produce the same color as would have been produced in a ST. Similarly a TT palette register access will map to one of the 256 registers with the 4-bit color definitions mapped into the 24 bits. FALCON palette accesses map directly. ST modes can access all 256 register 16 at a time via the ST palette bank register as in the TT. Writing to the color palette via any of the three paths change the same physical memory so a screen displayed in a FALCON video mode would be affected by writes to the ST palette.

Falcon modes 000/111, 001, 010/100, and 011, support X x Y display modes of 2, 4, 16, and 256 colors, respectively. These modes require storage of video data as <u>packed</u> pixels, instead of the planar pixel format supported in the ST and TT modes (see section 4.2).

> Note: Modes 000 qnd 111 as well as 010 and 100 are identical with respect to palette access, i.e., the same addresses are used for foreground and background. However, modes 111 and 100, which are high res modes, only allow 4096 (16x16x16) different possible values for background and foreground colors while the low res modes 000 and 010 allow 16M (256x256x256). High res modes must program the upper and lower nybbles of each CLUT entry with identical values, hence 4096 colors instead of 16M.

Falcon mode 000 and 111 access only two palette entries: entries 254 and 255. This not affected by the bank select. Video inversion can be turned on and off by setting or clearing <u>either</u> ST palette entry 0 bit 0 or TT palette entry 0 bit 1. The table below defines how inversion affects access to the palette.

<u>Invert</u>	<u>Background</u>	<u>Foreground</u>	<u>Border</u>
0	254	255	254
1	255	254	254

This is identical to ST and TT duochrome modes.

Border color selection from the palette is identical to that in the corresponding TT modes. Modes which allow bank selection use the first entry in the selected bank. Eight bit and true color modes use palette entry 0 for borders. Duochrome modes use palette entry 254 for border color regardless of whether or not inverse video is enabled.

Three new video modes are supported in FALCON: true color mode of 16 million colors and a separate true color mode which provides 256 color overlays are possible in resolutions up to and including 640 x 480 VGA. Higher resolutions are precluded due to memory bandwidth. A 16 color hires mode is provided for resolutions up to 1280 x 960.

The true color mode, mode 101, requires 24 bits of data for each pixel displayed; one byte for each color. This data provides an 8 bit address for each of the three CLUTs in the RAMDAC. The CLUTs can be programmed to provide gamma correction for a specific monitor.

The "pseudo/true color" mode, Falcon mode 110, requires 32 bits of data per pixel. Eight bits per color provide data directly to the DACs for 16 million true color generation. The extra byte of data provides for one of 256 colors of overlay. This byte is compared to a mask value (stored in the RAMDAC control register) on a pixel per pixel basis. When a non-zero mask value is present in this byte, the data is used as an address to all three CLUTs. The data in this palette entry then replaces the 24 bit value as input to the DACs. This mode can be used to quickly move a 256 color window around the screen without the overhead of altering the true color data which comprises the background. It also provides a way to turn overlays off and on with a single write to alter the mask value.

4.1.1 Compatible Mode Support

Video compatibility support for ST and TT software is provided by special handling of data in the color palette RAM. In the RAMDAC, data steering directs color data bits to their respective CLUTs in the following manner:

> In the ST Color Palette, only three bits per color are defined. Therefore, each three bit color pattern is written to bit ranges 7-5, 4-2, and 1-0 of the appropriate CLUT entry within the selected bank as determined by control register bits 3 through 0. For example, a bit pattern of 101 will be written as 10110110. This mapping ensures true black and white levels on the DACs. Thus, if the processor initializes an entry as full scale, which is 7H in this mode, a value of FF will be entered.
>
> For accesses to the TT Color Palette, four bits/entry are defined and must be written to both nybbles of each byte-wide entry. This configuration will support true black and white levels for TT modes. Since the TT pallette is 256 colors, bank select values in the control register do not affect programming of CLUTs within this address space.

Initializing the CLUTs for operation in the high resolution FALCON modes requires that each color value written to the FALCON palette have identical upper and lower nybbles. This can be more easily done through the TT palette address range. Because of the nybble duplication used for TT palette initialization in the RAMDAC , the two 4 bit DACs in each color pair can be initialized with the same value, as required in high resolution modes. Thus, one word write will initialize all three colors within a specific entry when using the TT palette, whereas two words (and two bus cycles) would be required when writing to the FALCON pallete.

Accesses to Falcon color pallete are direct one-to-one mapped.

Switching color banks or changing palette values "on the fly" during active video will probably cause the video to "sparkle" as the CLUT rams are single ported. To avoid this, palette changes should be made during blanking intervals.

4.2 Video RAM

In ST and TT video modes, display memory is configured as logical planes (1, 2, 4, or 8) of interwoven contiguous words forming a 32,000 byte (for ST modes) or 153,600 byte (for TT modes) physical plane. The starting address can be set to any 8 byte boundary (in dual-purpose RAM only). The starting address(es) of display memory are loaded into the Video Base Register(s) (the most significant byte of the thirty two bit addresses is always zero, i.e. within the ST image). One of these registers is loaded into the Video Address Counter at the beginning of each frame. The address counter is incremented as the BitMap is read. Note that there are two Video Base Registers for even and odd fields. Only the even register set need be used for non-interlace modes.

BitMap planes are transferred from RAM to data steering ahead of the video FIFO where the planes are translated into packed pixels for video processing. The translated data proceeds through the video FIFO to the RAMDAC where one bit from each plane is collectively used as the index (plane 0 appears first in RAM and provides the least significant bit of each pixel) to a specific ST or TT palette register (depending on the Shift Mode).

FALCON video modes require video display memory to be organized as packed pixels instead of planes. Each pixel is defined by 1, 2, 4, 8, 24, or 32 consecutive bits in memory depending on the mode. In the 2, 4, and 8 bit/pixel modes the bits form an address to select one of 4, 16, or 256 palette registers respectively. In pseudo-true color mode, the most significant eight bits of each longword can contain a separate image. In this mode, there is one longword for each pixel. The lower 24 bits contain data fed directly to the DACs, eight bits each of red, green, and blue. The upper eight bits, if non-zero, are used to select one of 255 entries from the CLUT. In standard true color mode, the upper eight bits of each longword are ignored.

4.2.1 Video Data Word Formats

Video data stored in RAM in planar fashion must comply with ST/TT format. Data stored in packed form must comply to the formats defined below.

```
     8 bits/pxl    |7......0|7......0|7......0|7......0|

     4 bits/pxl    |32103210|32103210|32103210|32103210|

     2 bits/pxl    |10101010|10101010|10101010|10101010|

     1 bit/pxl          |00000000|00000000|00000000|00000000|

     Pixels always go left to right
```

True Color Mode

```
   ---------------------------------------------
  | XXXX XXXX | R7....R0 | G7....G0 | B7....B0 |
   ---------------------------------------------
    ^                                       ^
    |                                       |

    Data31                                  Data0
```

Psuedo/True Color Mode

```
   ---------------------------------------------
  | PC7...PC0 | R7....R0 | G7....G0 | B7....B0 |
   ---------------------------------------------
```

4.3 External Video Interface

Because of the wide range of video resolutions supported by Falcon, some display devices will have to be driven by a daughter card connected to the motherboard at the video expansion connector.

The motherboard will support VGA and super VGA monitors. The RAMDAC is capable of generating RGB signals for a variety of video modes which are not displayable on a VGA or super VGA monitor. Separate horizontal and vertical outputs are supplied to the Expansion connector by the resident timing generator (VTG). For NTSC and PAL cards, the VTG will generate broadcast standard sync signals. VGA and super VGA compatible dot clocks are provided by oscillators, through a mux, to the RAMDAC. Dot clock selection is controlled by bit 10 of the VTG master control register. An external dot clock input from the expansion card is muxed through to the RAMDAC by grounding the MUXSEL pin on the connector.

The external video port also provides a bidirectional parallel data interface to the Falcon video subsystem for external devices such as video digitizers and shifters. As an input port, this port supports direct display of incoming video data through the RAMDAC and/or storage of video frame data in main memory. As an output port, it can be used to supply a data stream from the video buffer to a shifter for generation of extremely high resolution monochrome displays. The 32 bit data bus on the interface can also be used for transfer of display lists from the video buffer for such devices as polygon rendering engines.

4.3.1 Frame Grabbing

The Falcon architecture allows a frame storage operation to occur as the frame is displayed through the RAMDAC. In order to perform frame storage operations, Falcon and the external digitizer must be genlocked. This can be done in one of two ways. The external card may genlock the video system by supplying horizontal and possibly vertical syncs and a free running, phase locked pixel clock, or it must synchronize to the internal timing generater. In either case, all of the timing signals except the two syncs and the pixel clock are always generated by the Falcon VTG. When genlocking to the external card, software must program the VTG Video Master Control Register for external sync mode.

A typical frame grab should proceed in the following manner. At the beginning of the vertical blank interval preceeding the frame to be stored, the external card must assert EVSEL1* and/or EVSEL2*. Assertion of both EVSELs does two things: it gates off VBREQ* from the VTG which requests video data bursts from the Memory Control Unit (MCU) and it switches the Data Funnel (FNL) external video port onto it’s internal video buffer inputs. During a frame store, all video data is supplied by the external card, and the FNL video buffer is used as temporary storage for the incoming data until the MCU can write the data into DRAM. No other actions are required during the VS interval. At or before the end of VBLNK, the external device must drive the first 32 bit unit of data onto the data port. This data must be valid on the internal video bus before the rising edge of DEN. Once active video commences, i.e., while DEN is driven high, the device must supply new pixel data upon demand to the RAMDAC within 15ns of the rising edge of NXT. Additionally, a write strobe (EVSTRB) must be supplied with each unit of data to store it in the video buffer. As soon as data is stobed into the video buffer, it is set up at the 64 bit memory port of the FNL for transfer to DRAM by the MCU at a later time. The controller internal to the FNL will wait until the buffer is half full before asserting VWR* to inform the MCU that external video data is present. The MCU gives this interrupt the same priority as VBREQ* and will begin servicing as soon as the current operation is completed; the maximum latency is such that data reads from the buffer are guaranteed to begin before the buffer can overflow. The MCU then asserts VDEN* to turn on the FNL memory port buffers and strobes the quad word of video data into DRAM on the falling edge of VACK*. This continues until the buffer is empty and EVSEL*s are inactive. The external device must negate these inputs at the end of the frame once the last pixel word has been strobed into the buffer. Note that certain conditions may exist in which the VBLNK interval following a frame store does not provide sufficient time for the MCU to finish storing the frame and begin loading the buffer with data for the next frame. In such cases, the display will be allowed to flicker, and the system will recover by requesting a video burst from main memory during the following VBLNK interval.

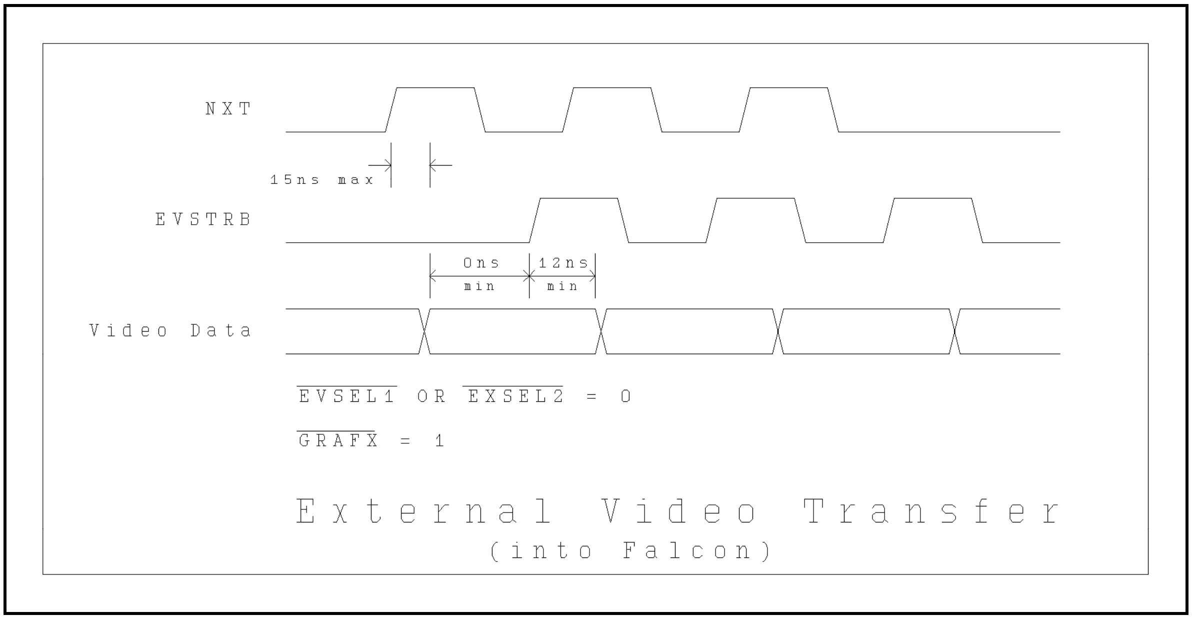

Figure 4.1

4.3.2 Graphic Overlays on External Video

Falcon also supports overlay of video in screen memory onto external video in Falcon mode 110 (see section 4.1). This operation requires that the external video source and Falcon be genlocked. Two modes of operation are supported. For external 24 bit true color data mixed with internal 8 bit overlay data, EVSEL1* must be asserted before the end of the VBLNK interval preceding the first frame of incoming video which is to have an overlay. EVSEL2* must not be driven or may be driven high allowing the FNL chip to drive the top byte of the video bus to the RAMDAC. For an external overlay on internal true color data, EVSEL2* must be asserted and EVSEL1* negated.

Assertion of EVSEL1* configures the video port to input the lower 24 data lines from the connector and disables the FNL output buffers that normally drive this section of the video bus to the RAMDAC. Assertion of EVSEL2* likewise configures the upper 8 bits of the port. Assertion of both disables all display of internal video data and configures the port for external data input as noted in the previous section.

4.3.3 Alternate Display Support

The red(R), green(G), and blue(B) outputs from the RAMDAC as well as horizontal and vertical sync signals are available on the external video connector. Cards for this port should contain a clock driver to supply a pixel clock if the available 28 Mhz or 80Mhz clocks resident on the motherboard are not appropriate. The MUXSEL connector pin controls the pixel clock multiplexer located on the motherboard. For expansion cards supplying an alternate clock, this pin should be tied to ground.

Use of this port for driving an external shifter is only necessary when extremely high resolution displays are desired. A shifter card attached to this port must only supply two signals to the Falcon motherboard. The GRAFX* input must be driven low to tell the video buffer to shift data out on demand from the external shifter (not from the RAMDAC). The other signal is the read pulse to the video buffer, EVSTRB. The external device may read the buffer at any rate up to about 32 MHz, the memory refresh bandwidth limit.

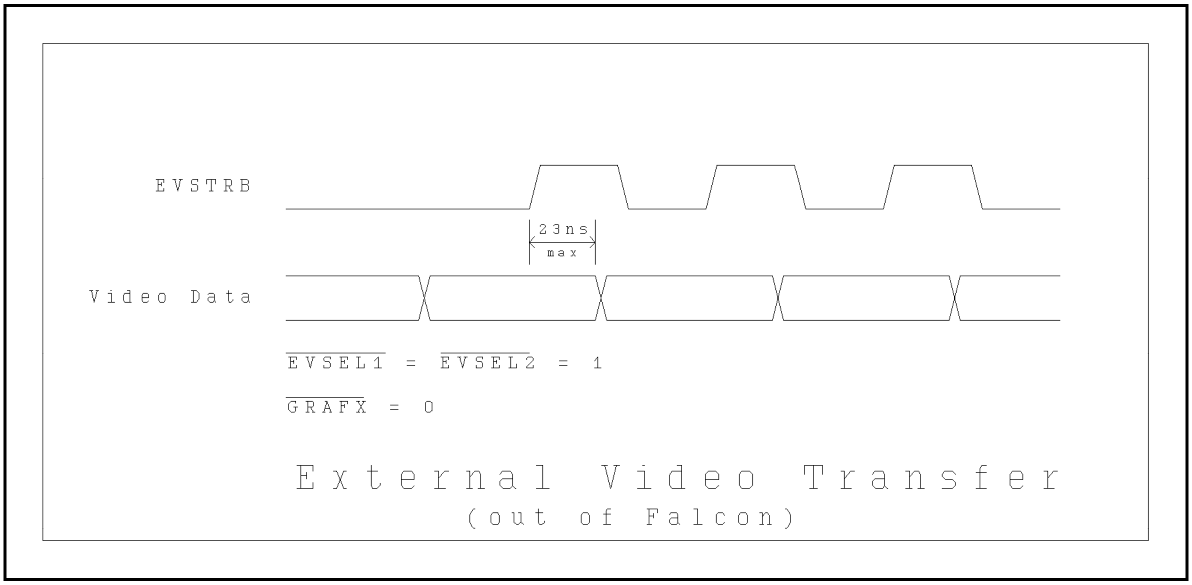

Figure 4.2

Note: R, G, and B signal traces on the daughter board **must not be terminated.** Keep traces to the monitor connector as short as possible.

4.3.4 External Video Interface Description

The expansion card interface is composed of the following:

- a 32 bit bidirectional video data bus
- bidirectional video timing signals
- an 8 bit I/O bus for accessing control registers
- various control signals.
- analog R,G, and B outputs
- a dotclk input to the system
- +/- 5 and 12 volt supplies

The 32 bit data bus is a bidirectional bus for transfer of video data onto and off of the expansion card. The expansion card should use transceivers to drive this bus. Directional controls must be pinned out to the connector for controlling corresponding tristate buffers on the motherboard.

The I/O port consists of an 8 bit bidirectional data bus (IODATA7 through IODATA0) located on the low byte of the sytem I/O data bus, 4 address lines (IOADDR5 through IOADDR1), a read strobe, and a write strobe. Data buffers on the expansion card are enabled by an active low level on the read strobe, and data can be latched off the IO bus on the rising edge of the write strobe.

Signal Name	Dir	Function
EVSEL1*	I	External video select 1. Active low. Controls the source of video vdata[23:0] on the motherboard. Source is expansion card, when active.
EVSEL2*	I	External video select 2. Active low. Controls the source of video vdata[31:24] on the motherboard. Source is expansion card, when active.
EVSTRB	I	External Video Strobe. Active high. Read or write strobe to the video data buffer (see the truth table to follow.)
NXT	O	RAMDAC output read strobe to the active source of video data. Monitored by the external card when supplying any or all video data to the RAMDAC.
GRAFX*	I	When active, the video data buffer supplies instruction list to external device on the rsing edge of EVSTRB. All analog video signals must be supplied by external device to a monitor connector located on the external card.

Table 4.3

The following truth table describes all combinations of control signals and the corresponding function implemented by them.

EVSEL1*	EVSEL2*	GRAFX*	FUNCTION
0	0	0	Invalid
0	0	1	EV card driving all video data lines for frame store and/or display through RAMDAC. EVSTRB writes data into buffer.
0	1	0	Invalid
0	1	1	EV card supplies the lower 24 video data bits for true color background. The video data buffer supplies upper 8 bits of overlay data. <u>Falcon mode 6 only</u>
1	0	0	Invalid
1	0	1	EV card drives upper byte of video data for overlays onto true color background as supplied by the video data buffer. <u>Falcon mode 6 only</u>
1	1	0	EV card reading 32 bit instruction list words from video data buffer. EVSTRB reads data from buffer.
1	1	1	All video data supplied by video data buffer. All EV card video data buffer outputs are high impedance.

Table 4.4

Video Expansion Connector							
pin	signal	pin	signal	pin	signal	pin	signal
1	EVDATA0	25	EVDATA24	49	GND	73	IOADDR1
2	EVDATA1	26	EVDATA25	50	GND	74	IOADDR2
3	EVDATA2	27	EVDATA26	51	GND	75	IOADDR3
4	EVDATA3	28	EVDATA27	52	GND	76	IOADDR4
5	EVDATA4	29	EVDATA28	53	VCC	77	IORD*
6	EVDATA5	30	EVDATA29	54	VCC	78	IOWR*
7	EVDATA6	31	EVDATA30	55	VCC	79	DEN
8	EVDATA7	32	EVDATA31	56	VCC	80	EVSEL1*
9	EVDATA8	33	RESET*	57	GND	81	EVSEL2*
10	EVDATA9	34	GND	58	GND	82	EVSTRB
11	EVDATA10	35	GND	59	GND	83	NXT
12	EVDATA11	36	GND	60	HSYNC	84	GRAFX*
13	EVDATA12	37	GND	61	VSYNC	85	+12V
14	EVDATA13	38	GND	62	RRETURN	86	+12V
15	EVDATA14	39	GND	63	GRETURN	87	-12V
16	EVDATA15	40	GND	64	BRETURN	88	-5V
17	EVDATA16	41	GND	65	IODATA0	89	GND
18	EVDATA17	42	GND	66	IODATA1	90	XDOTCLK
19	EVDATA18	43	GND	67	IODATA2	91	GND
20	EVDATA19	44	GND	68	IODATA3	92	HBLANK
21	EVDATA20	45	GND	69	IODATA4	93	VBLANK
22	EVDATA21	46	GND	70	IODATA5	94	RED
23	EVDATA22	47	MUXSEL	71	IODATA6	95	GREEN
24	EVDATA23	48	GND	72	IODATA7	96	BLUE

Table 4.5

The connector is a 96 pin Eurocard type.

4.3.5 Monitor Connector

Standard video output is provided on a 3 row 15 pin VGA compatible connector.

VGA Connector Pinout	
Pin	Signal
1	Red
2	Green
3	Blue
4	Monitor ID 2
5	Ground
6	Red return
7	Green return
8	Blue return
9	key
10	Ground
11	Monitor ID 0
12	Monitor ID 1
13	Horizontal sync
14	Vertical sync
15	

Table 4.6

4.4 Video Timing Control

All video timing control signals are generated by the Video Timing Generator chip (VTG) except when the video is genlocked to an external device on the external video port. The VTG also generates all video related signals to the Memory Control Unit (MCU) and Data Funnel (FNL) required to ensure accurate transfer of data from screen memory to the RAMDAC in all video modes.

4.4.1 Control Functions

Two registers, the video master control register (VMC) and video timing control register (VTC), are principally involved in control of outputs to other chips and to the display. These registers are defined as follows:

VMC Video Master Control (xxFF82C0)

abcd efgh ijkl mnop

p	Hsync source	0=Internal 1=External
o	Hsync level	0=Active low 1=Active high
n	Hsync enable	0=Disable 1=Enable
m	H-counter on	0=Reset to 0 1=Count
l	Vsync source	0=Internal 1=External
k	Vsync level	0=Active low 1=Active high
j	Vsync enable	0=Disable 1=Enable
i	V-counter on	0=Reset to 0 1=Count
h	Csync level	0=Active low 1=Active high
g	Csync enable	0=Disable 1=Enable
f	Dotclk select	0=VGA 1=Super VGA
e	Reserved	
d	Alternate fields	0=disabled 1=enabled
c	Equalization	0=Enabled 1=Disabled
b	Wide Equ’n	0=Disable 1=Enable
a	PAL/NTSC	0=PAL (5 pulses) 1=NTSC (6 pulses)

```
VCO  Video control register (xxFF82C2)
     0000 0000 0smm mvnr
                   r    Repeat lines   0=Disabled
                                       1=Enabled
                   (Doesn’t work correctly in interlaced mode)
                   n    Prescale dotclk     0=No prescale
                                       1=Divide by 2
                   v    Register select     0=VDB0/VDE0
                                       1=VDB1/VDE1
                   mmm  Video mode*
                        000 1  bpp Duochrome
                        001 2  bpp
                        010 4  bpp
                        011 8  bpp
                        100 4  bpp hi-res
                        101 24 bpp True colour
                        110 24 bpp True colour + overlay
                        111 1  bpp hi-res monochrome
                   s    Line skip          0=Disabled
                                       1=Skip alt’ lines
```

*Note: The Video Mode lines are read only in the Video Control Register. They are set by writes to the ST, TT, or Falcon Video Mode Registers.

The HSYNC and VSYNC pins on the VTG are bidirectional. The selection of input/output function is controlled by bits in the VMC that must be set by software to select the appropriate sync direction for genlock of the video system to an external device. These bits are cleared (select internal sync) on initialization.

The VCO register controls all timing parameters that must be changed to display the various ST and TT compatible modes. A set of VCO parameters for each compatible mode are stored in the VC# registers. These parameters must be loaded at boot for the type of monitor connected. Values are automatically written from the appropriate VC# register to the VCO register when accesses to ST or TT shift mode registers indicate that a compatible mode is being selected.

The VC# registers are defined as follows:

xxFF82E0 VC1 definitions for 320 X 200, 16 color mode
(old ST mode 00/ TT mode 000)

xxFF82E2 VC2 definitions for 640 X 200, 4 color mode
(old ST mode 01/ TT mode 001)

xxFF82E4 VC3 definitions for 640 X 400, 2 color mode
(old ST mode 10/ TT mode 010)

xxFF82E6 VC4 definitions for 640 X 480, 16 color mode
(old TT mode 100)

xxFF82E8 VC5 definitions for 1280 X 960, hi-res monochrome
(old TT mode 110*)

xxFF82EA VC6 definitions for 320 X 480, 256 color mode
(old TT mode 111)

*Note: true compatibility support is not possible for TT mode 110 mode because timing parameter register values must be changed from the default VGA or TV values. Also this mode exceeds the bandwidth for VGA and TV monitors.

Table 1 indicates the settings required in each VC# register for support of either TV or VGA monitors. Software must configure these registers for the connected monitor or video card at boot.

CSn	Resolution	TV	VGA
		s mmm v n r	s mmm v n r
1	320x200x4	1 010 0 1 0	0 010 0 1 1
2	640x200x2	1 001 0 0 0	0 001 0 0 1
3	640x400x1	0 000 0 0 0	0 000 0 0 0
4	640x480x4	0 010 1 0 0	0 010 1 0 0
5	1280x960x1	0 111 0 0 0	0 111 0 0 0
6	320x480x8	0 011 1 1 0	0 011 1 1 0

Table 4.7

The VTG monitors accesses to the RAMDAC control register; writes to this register will update the mode bits, set the PACKED* output to a low state (active), turn off DIV2, Rline, and sline, and clear the v bit.

ST/TT compatibility is fully accomplished using the VC# registers without the need to change any timing parameter register values. The VTG will respond to any access to ST or TT shift mode registers by driving DTACK and, on reads, data. Any write to these registers will set the PACKED* output high (inactive). The VTG also responds to writes of the TT Shift Mode register bit 12 by setting it's HMONO output accordingly. This output goes to the RAMDAC which steers CLUT data to the DACs appropriately.

4.4.2 Timing Generater

The majority of the VTG circuitry forms a programmable function generator for producing the syncs, blanking, and display enable signals. The following registers control the circuits configuration:

8280 - HORIZONTAL COUNTER

Counts number of TGCLK's per line.

8282 - HORIZONTAL HALF LINE TOTAL

Number of half-lines per frame

8284 - HORIZONTAL BLANK BEGIN

TGCLK number on a line where horizontal blanking begins.

8286 - HORIZONTAL BLANK END

TGCLK number on a line where horizontal blanking ends.

8288 - HORIZONTAL DISPLAY BEGIN

TGCLK number on a line where horizontal display enable begins. The 'Line Half bit' in bit 12 indicates in which line half the display enable will start.

828A - HORIZONTAL DISPLAY END

TGCLK number on a line where horizontal display enable ends. The 'Line Half bit' in bit 12 indicates in which line half the display enable will end.

828C - HORIZONTAL SYNC START

TGCLK number on a line where horizontal sync begins. This is the same point that both field sync and equalization pulses begin if enabled.

828E - HORIZONTAL FIELD SYNC

TGCLK number on a line where horizontal field sync pulses end if equalization is enabled.

8290 - HORIZONTAL EQUALIZATION END

TGCLK number on a line where equalization pulses end if enabled.

8292 - VIDEO BURST TIME

Indicates which TGCLK after the start of horizontal blanking that the next line's data may be requested from the MCU.

8294 - HORIZONTAL WORD COUNT

This counter indicates the number of VACK's from the MCU that have occurred on the current line. This number is used to determine whether or not the line's data requirement has been satisfied.

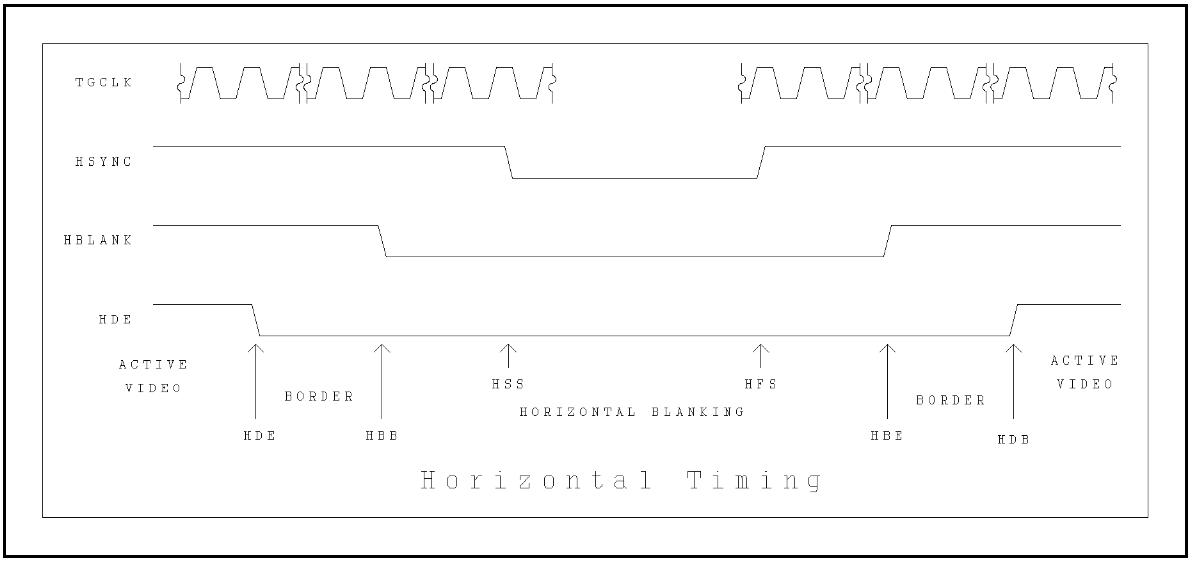

Figure 4.3

82A0 - VERTICAL COUNTER

Counts number of half lines per frame.

82A2 - VERTICAL FIELD TOTAL

Counts number of half lines per field. This determines when a new field should be started.

82A4 - VERTICAL BLANK BEGIN

Half line where vertical blank begins.

82A6 - VERTICAL BLANK END

Half line where vertical blank ends.

82A8 - VERTICAL DISPLAY BEGIN (REGISTERS 0,1)

Determines when to begin vertical display enable.

82AA - VERTICAL DISPLAY END (REGISTERS 0,1)

Determines when to end vertical display enable.

Note: The above two registers are organized in pairs. Which pair is used is determined by the 'V' bit in the VCO register at 82C2.

82AC - VERTICAL SYNC START

Half line upon which vertical sync begins.

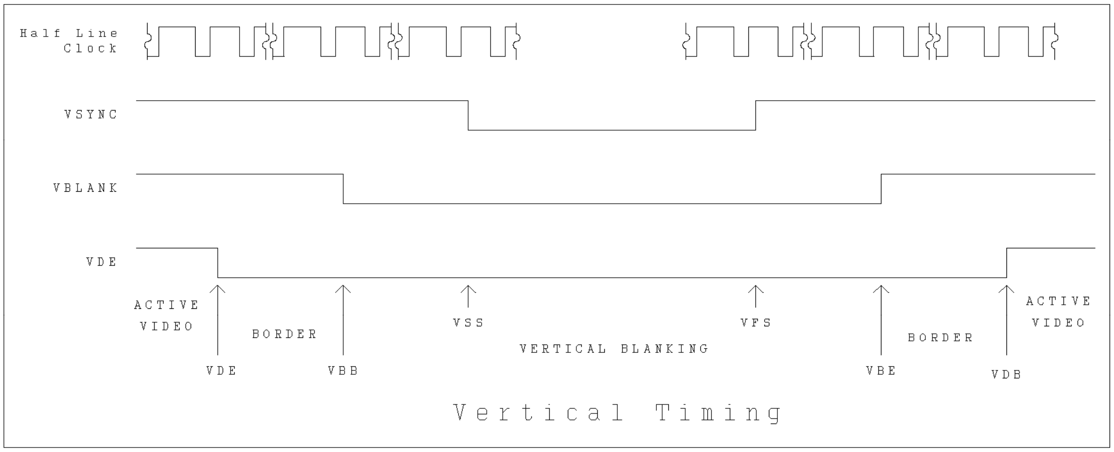

Figure 4.4

Section 5   Graphics Coprocessor

*** Information to be supplied by Martin Brennan ***

Section 6 Sound Subsystem

The FALCON architecture extends the music subsystem presently available on the ST/MEGA/TT computers. In FALCON, the output of the ST PSG chip,external analog line input, modem audio, and microphone input are mixed and fed to a proprietary 8 bit A-to-D converter. The digitized audio can then be stored in memory via the sound DMA channel or mixed with digital sound data from memory and played via a proprietary 16 bit D-to-A converter (DAC) through an internal speaker, line output, or stereo headphones. Additionally, external digital sound data can be stored in memory or read from memory making it possible to implement a digital recorder.

FALCON extends the STE and TT digital sound modes to include 16-bit stereo and allows the sample clock to come from external circuitry making it possible to connect to wide variety of external digital audio sources.

FALCON is also equipped with a Musical Instrument Digital Interface (MIDI) which provides high speed serial communication of musical data to and from more sophisticated synthesizer devices.

6.1 Programmable Sound Generator

FALCON contains a PSG circuit compatible with the ST sound system. The PSG produces music synthesis, sound effects, and audio feedback. With an applied clock input of 2 MHz, the PSG is capable of providing a frequency response range between 30 Hz (audible) and 124 KHz (post-audible). The generator places minimal amount of processing burden on the main system (which acts as the sequencer) and has the ability to perform using three independent voice channels. The three sound channel outputs are combined and added to a stereo analog audio input which is then fed to the A-to-D channel. The resulting 16-bit digital data stream is available to be mixed with the digital sound output or to be recorded via a DMA channel into memory.

(Reference Engineering Hardware Specification of the Atari ST Computer System, page 10.)

6.2 DMA Sound

FALCON also includes a DMA-driven digital sound system that allows the playback or synthesis of complex waveforms and recording at a variety of sampling rates.

6.2.1 Overview

Sound in the form of digitized samples can be stored in or retrieved from system memory. Playback samples are fetched from memory via DMA and provided to a digital-to-analog converter (DAC) at a constant sample frequency specified by the user. The sound processing circuit adjusts volume, tone, and balance along with the mixing with PSG and external audio. The analog audio signal is then available at a external headphone jack, an internal speaker, the internal modem, and line output. The digital data stream is also available for use by external hardware. Record samples taken from the A-to-D or external source are stored in memory via DMA.

Two channels of D-to-A are provided. They are intended to be used as the left and right channels of a stereo system. Of course, they are mixed together when fed to the internal speaker. A mono mode is provided which will feed the same data to both channels simultaneously (STE/TT compatible 8 bit modes only). The only restriction placed on mono mode is that there must be an even number of samples (see data format section for details).

6.2.2 Data Format

In the 8-bit modes each sample is stored as an eight bit quantity. The most significant bit is the sign and the other seven bits are magnitude. In the stereo 8-bit modes there is one word per sample, the upper byte contains the left channel sample and the lower byte contains the right channel sample. In the 8-bit mono mode bytes are accessed sequentially. However, they are still fetched a word at a time. Therefore, there must be an even number of samples.

In the 16-bit stereo mode each sample is stored as a word in memory. The most significant bit is the sign and the other fifteen bits are magnitude. The left channel word is first with the remaining words alternating right-left-right etc. Therefore, there must be an even number of samples. The DMA channel into memory (record) can only store samples in the 16 bit form.

A group of samples is called a frame. A frame may be played once or can automatically be repeated forever. Frames occupy a contiguous block of memory and are specified by their starting and ending addresses. The ending address is the address of the last sample + 2 (the address of the word following the last sample). A clock pulse is generated at each frame boundary and fed to timer A of the MFP-1. Using the MFP counter mode, frames can thus be counted. This pulse can also be used to generate an interrupt on bit 7 of the MFP-1 general purpose I/O port. Frames may be linked together by defining a new frame while the current frame is being played. The new frame will begin at the end of the current frame.

As an example, suppose you have three frames (A, B, and C) and we want to play frame A once, then play frame B five times, and finally play frame C twice. To accomplish this you can do the following:

1. Setup frame A.

2. Write 03h to the sound DMA control register to start playing with repeat.

3. Setup timer A to use an external clock, initialize its count to 05h, and have it interrupt when count = 0.

4. Setup frame B.

5. Go do something else until interrupted.

6. Setup frame C.

7. Setup timer A count to 03h.

8. Go do something else until interrupted.

9. Write 1 to the sound DMA control register to cause playing to stop at the end of the frame.

In this example no mention is made of setting the sample rate, volume or tone controls. It’s assumed that all of these have been set up ahead of time. It should be obvious how this example can be extended to allow volume or tone to be modified at specific points during playback.

Note If we had loaded the sound DMA control register with a 1 in step 2, frame A would have been played once and sound would have been disabled. A zero can be written to the sound DMA control register at any time to stop playback immediately.

The DMA channel does not determine how the samples are defined, only their location in memory and the order in which they are handled. The data need not be digitized sound at all. If the data is to be monitored by the internal DAC, or if the track selection is to function correctly, then the samples must be stored in a certain order. See the sample format section for a description of the various formats.

6.2.3 MICROWIRE Interface

A general purpose MICROWIRE interface is provided to access certain sound control registers and allow the future addition of other MICROWIRE devices. For this reason, the following description of its use will make no assumptions about the device being addressed.

The MICROWIRE bus is a three wire serial connection and protocol designed to allow multiple devices to be individually addressed by the controller. The length of the serial data stream depends on the destination device. In general, the stream consists of N bits of address, followed by zero or more don’t care bits, followed by M bits of data. The hardware interface which has been provided consists of two 16 bit read/write registers. One data register which contains the actual bit stream to be shifted out and one mask register which indicates which bits are valid.

Let’s consider a mythical device which requires two address bits and one data bit. For this device the total bit stream is three bits (minimum). Any contiguous three bits of the register pair may be used. However, since the most significant bit is shifted first, the command will be received by the device soonest if the three most significant bits are used. Let’s assume: 01 is the device’s address, D is the data to be written, and X’s are don’t cares. Then all of the following register combinations will provide the same information to the device.

```
1110 0000 0000 0000  Mask
01DX XXXX XXXX XXXX  Data

0000 0000 0000 0111  Mask
XXXX XXXX XXXX X01D  Data

0000 0001 1100 0000  Mask
XXXX XXX0 1DXX XXXX  Data

0000 1111 1111 0000  Mask
XXXX 01XX XXXD 0000  Data

1111 1111 1111 1111  Mask
01XX XXXX XXXX XXXD  Data
```

The mask register needs to be written before the data register. Sending commences when the data register is written and takes approximately 16uS. Subsequent writes to the data and mask registers are blocked until sending is complete. Reading the registers while sending is in progress will return a snapshot of the shift register shifting the data and mask out. This means that it is safe to send the next command when the mask register returns to its original state or the data register has shifted out all 16 bits. Note that the mask register does not need to be rewritten if it is already correct. That is, when sending a series of commands the mask register only needs to be written once.

6.2.4 Volume and Tone Control

Circuitry exists to provide volume, tone, and mixing control. It is controlled via the MICROWIRE interface. The circuit has a two bit address field, address = %10, and a nine bit data field. There is no way of reading the current settings.

Data Field

```
     011 DDD DDD  Set Master Volume
         ||| |||
         000 000  -80 dB
         010 100  -40 dB
         101 XXX    0 dB

     101 XDD DDD  Set Left Channel Volume
          || |||
          00 000  -40 dB
          01 010  -20 dB
          10 1XX    0 dB

     100 XDD DDD  Set Right Channel Volume
          || |||
          00 000  -40 dB
          01 010  -20 dB
          10 1XX    0 dB

     010 XXD DDD  Set Treble
           | |||
           0 000  -12 dB
           0 110    0 dB (Flat)
           1 100  +12 dB

     001 XXD DDD  Set Bass
           | |||
           0 000  -12 dB
           0 110    0 dB (Flat)
           1 100  +12 dB

     000 000 0xs  Input Select
               |
               0  PSG/external disabled
                   (reset state)
               1  PSG/external enabled

     111 XDD DDD control
          || |||
          00 |||  select external clock
          01 |||  select 8Mhz clock
          10 |||  select 8Mhz clock (reset state)
          11 |||  select CD clock
             0||  mix ADC output with primary data input
             1||  mix aux input with primary data input
              00  data output = ADC output
              01  data output = ADC output
              10  data output = primary input data
              11  data output = aux input data
```

Note: The volume controls attenuate in 2 dB steps. The tone controls attenuate in 2 dB steps at 50 Hz and 15 kHz.

The input selector is used to control mixing of the PSG/external sound with the DMA sound. After reset, the input is grounded, and should be switched to either states 01 or 02 during initialization to avoid level mismatches during later switching.

The control register is used to select the source of the sound system master clock (the sample rate is derived from the master clock) and to configure the sound data path. The external clock comes from the external digital sound connector. The internal clock has a fixed frequency of 8 Mhz. The input to the D-to-A converter is the digital data stream from the DMA channel added to either the A-to-D converter output or the aux data input as controlled by bit 2. The data output (which is fed to the external connector and can be recorded) comes from either the A-to-D converter output, the data from the playback DMA channel, or the aux data input as controlled by bits 0 and 1.

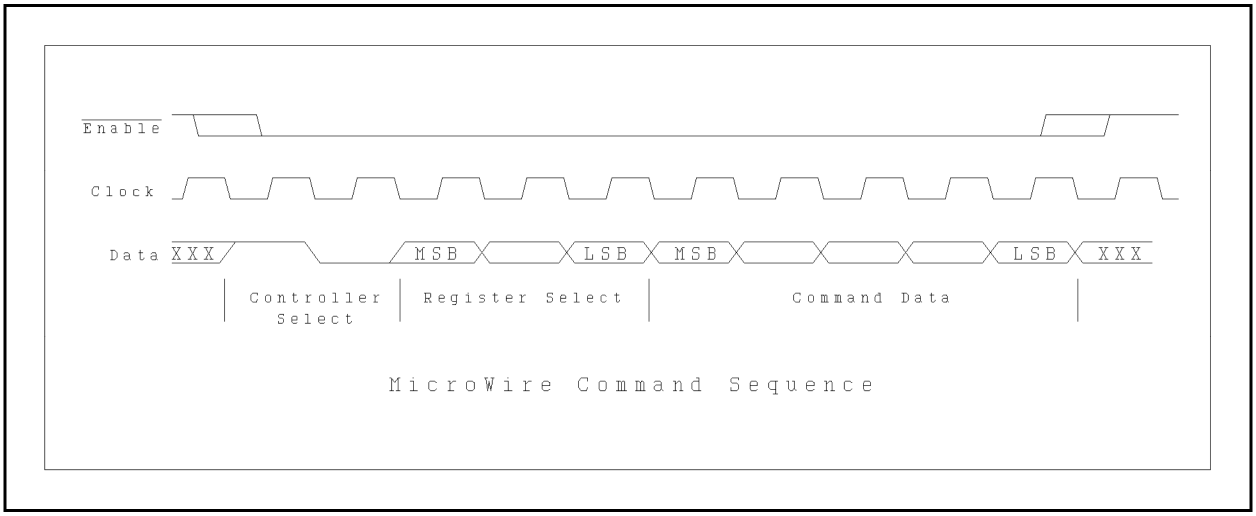

Figure 6.1

6.2.5 Sample Rate Selection

The Sound Mode Control register (IO+8920h) is used to select the sample rate clock prescale. Bits 0 and 1 select the prescale value. With the internal clock (8 Mhz) selected and the prescale value set to 160, the sample rate would be 50 Khz.

6.2.6 Sample Format

Bits 8, 9, and 10 of the Sound Mode Control register (IO+8920h) select the number of playback tracks. A track is a series of related samples which when converted to analog produce a single audio signal. There must be one sample per sample period per track. Two track mode is the default (track1-left and track2-right). The internal DAC is comprised of two audio channels so can process two tracks. If more than two tracks are selected, the two which are fed to the DAC are selected by bits 12, 13, and 14 of the Sound Mode Control register. Samples are always stored in memory as words with the left channel sample first followed by alternating right-left-right channel samples. For example, suppose six tracks are selected in 16 stereo mode. The first eight words of the frame would contain $track1_1$, $track2_1$, $track3_1$, $track4_1$, $track5_1$, $track6_1$, $track1_2$, $track2_2$ ... If the tracks are defined such that track1 is the left signal of a stereo channel and track2 is the right, then setting bits 12, 13, and 14 of the Sound Mode Control Register to 0, 0, and 0 respectively will allow that stereo channel to be monitored via the DAC.

The DMA hardware always fetches the samples sequentially from memory. The meaning of the samples is up to external hardware. For example, if four tracks are selected, four samples per sample period are output. The four samples are fetched from ascending word memory locations. The four samples may represent four tracks (independent audio channels) as is the normal case, or four samples of the same audio track (4x the selected sample rate). The first case could use the internal DAC to monitor two channels. The latter case could not use the internal DAC since it would not see all the samples and the output would be garbled. A maximum of ten tracks (five stereo channels) can be selected. Note that selecting more tracks for a given sample rate increases both the memory required and the memory bandwidth used by DMA.

The internal A-to-D always produces two samples (left and right channels) per sample period. The sample rate is the same as is currently set up for playback. Although the resolution of the A-to-D is 8 bits, the samples are recorded as 16 bit quatities.

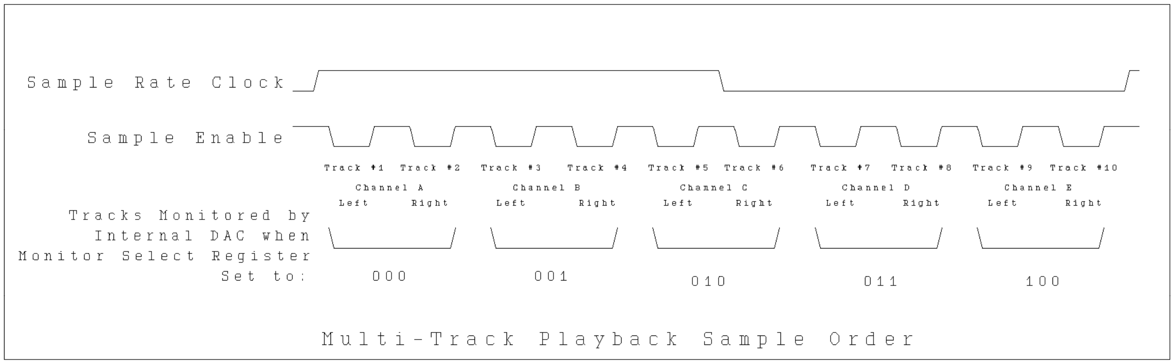

Figure 6.2

6.2.7 DMA Sound Record

Digital sound data can be recorded into memory. Any data present in the area of memory defined by the frame will be replaced with incoming samples. The frame is defined for record exactly as it is for playback. Bit 7 in the DMA Sound Control register (IO+8900h) selects whether the playback or record register set is addressed. The two independent register sets are identical and occupy the same IO locations. When recording the output of the A-to-D converter, the sample rate is the same as that set for playback and the samples are stored as 16 bit samples even though the A-to-D has a resolution of 8 bits.

The DMA channel itself is not sensitive to the sample rate when recording external data. DMA stores the first left channel sample received after the DMA has been enabled then alternating right and left samples (same format as 16-bit stereo playback). Multiple frames can be combined just as during playback. However, if the frame is allowed to repeat, that is store data into the same memory range, the original data will be overwritten. Software doing record will probably use the frame repeat to alternate between two buffers so that one buffer can be written to disk while the other is filling. Note that the Sound Mode control register does not affect the record DMA channel. Incoming samples are stored sequentially in memory irrespective of track format, sample rate, etc.

6.2.8 External Digital Sound Data

FALCON will have a rear panel connector for input and output of the digital sound data. The data is transferred via a four wire serial interface. The channel select (CHAN) determines which channel (0 = left or 1 = right) is selected for the current sample. The enable signal (EN) is active low to enable the data transfer. The other three signals are ignored when enable is high (the enable signal should be brought high for a minimum of six clock periods between samples). The enable and data signals are sampled by rising edges of the clock. The enable signal should be low for 16 rising edges of the clock to allow data to be sampled 16 times thus transfering a 16 bit quantity (MSB first). A master clock may be generated by the external device to produce specific sample rates or to synchronize with an external device. The sample rate is determined by dividing the master clock frequency by the sample rate prescale. The internal master clock is always 8 Mhz. The external device may elect to use the sample rate clock (FCLK) for data synchronization. Output data is always synchronous with the sample rate clock. Input data may be asynchronous unless it is intended to be monitored via the internal DAC. Then it must be synchronized to the sample rate clock. (Only one each left and right channel sample per sample rate clock cycle can be monitored.)

Digital Sound Data Connector			
pin	signal	pin	signal
1	MicroWire Ebable*	11	GND
2	MicroWire Clock	12	MicroWire Data
3	Input Data Clock	13	GND
4	Input Channel Select	14	Input Data Enable*
5	Input Data	15	GND
6	Output data	16	Output Data Enable*
7	Output Channel Select	17	GND
8	Output Data Clock	18	GND
	External Master Clock	19	GND
10	Sample Rate Clock		

Table 6.1

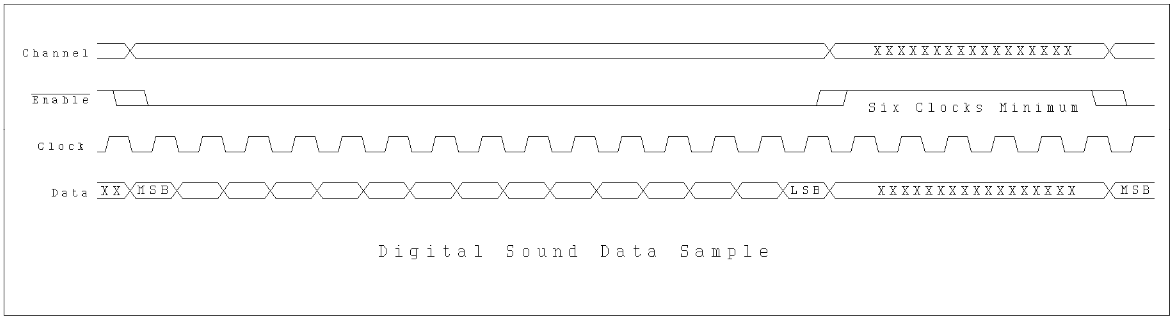

Figure 6.3

6.3 Musical Instrument Digital Interface (MIDI)

The MIDI allows the control of music synthesizers, sequencers, drum boxes, and other devices possessing MIDI interfaces. High speed (31.25 Kbaud) serial communication of keyboard and program information is provided by two ports, MIDI OUT and MIDI IN (the MIDI OUT also includes MIDI through data).

The MIDI communicates through the MC6850 Asynchronous Communications Interface Adapter (ACIA) to the system. The data transfer rate is a constant 31.25 Kbaud of 8-bit asynchronous data. (Reference Engineering Hardware Specification of the Atari ST Computer System, pages 11 and 17 for more information on the MIDI and ACIA.)

Section 7 VME Bus

FALCON systems provide for expansion by implementing the industry standard VMEbus, revision C.1. A FALCON VME interface can also accommodate alternate bus masters such as DMA cards.

7.1 System Controller

The main system board serves as the VMEbus system controller (a slot 1 "card") and implements the following functions:

- single-level (level-three) VMEbus arbiter
- IACK* daisy-chain driver
- global SYSCLK (16 MHz, independent of processor speed)
- global VMEbus time-out that drives BERR*

The level-three arbiter is designed to meet the VMEbus specification requirements.

The IACK* daisy-chain driver is designed to meet the VMEbus specification requirements.

The SYSRESET* line is driven low when a power-up occurs or when the CPU asserts its RESET* signal.

7.2 Address Partitioning

The starting address of the VME address space as seen by the CPU can be configured to be contiguous with the top of the single-purpose expansion RAM (by straps). Part of the 32-bit wide physical address space is partitioned to provide VME A24 and A16 address spaces.

The A24/D16 VMEbus interface is fixed at: 0xFE000000-0xFEFFFFFF.

7.3 Read-Modify-Write Cycles

The bus can not be arbitrated away from the CPU if it is in the midst of a read-modify-write (locked) cycle.

7.4 VME Interrupter

The system can write to an I/O address (0x00FF8E06) to generate a level 3 interrupt on the VMEbus. It can monitor a status register that indicates when that interrupt has been acknowledged and serviced. An I/O address contains a read/write status/control port, only the least significant bit of the least significant byte is defined. When set to 1, it generates a VMEbus level 3 interrupt. When cleared, the interrupt request is taken away.

Note that the level 3 interrupt must be masked off (either by setting the processor’s IPL or by masking the interrupt in the system controller) or the CPU will be immediately interrupted.

The system board responds to a VMEbus interrupt acknowledge cycle with the status ID of 0xFF for interrupts that would be auto vectored if serviced by the CPU. For levels 5 and 6, the status ID will be the 8-bit vector produced by the MFPs or SCC.

Section 8 System Bus

The following description of the FALCON internal bus (FBUS) assumes an intimate knowledge of both the 68030 and 68040 bus definitions. A thorough study of the Motorola user's manuals for both processors is highly recommended (see reference section).

8.1 FBUS Summary

The internal bus for FALCON, called the FBUS, is a hybrid of the 68030 and 68040 busses. The objective is an architecture which will accommodate both processors with a minimum of external glue logic. The FBUS is thus a subset of either bus.
2

The FBUS defines two modes of operation. One which is 030 like and the other 040 like. Most FBUS signals are defined such that they can be connected directly to the corresponding signal of either processor. Timing and function of these signals will therefore depend the current bus mode. Logic in each slave port must adjust how these signals are interpreted accordingly. The FBUS signal BMODE shall identify the current bus mode. FBUS masters can use either mode and masters using different modes can coexist on the same bus.

In order to keep the glue logic reasonable, many compromises were necessary. The FBUS does not support the dynamic bus sizing of the 030 or bus snooping by the 040. CPU control involving halts or retries is not supported in either bus mode.

The FBUS does offer unique features. The FBUS wide mode allows devices to connect directly to the 64 bit video memory data bus permitting double longword transfers. The FBUS defines a burst write in 030 bus mode, which, although never used by a 68030, is available to a 030 mode bus master.

Slave ports must monitor the BMODE signal and support either bus mode on a cycle by cycle basis. Bus masters must drive BMODE according to the mode they use. The particular timing on the bus should conform to or be compatible with the 33Mhz timing specification for the MC68030 or MC68040 for the corresponding bus mode.

The video memory is the only 64 bit wide port currently defined in FALCON. The MCU will monitor WID0-WID1 and drive WEN. Masters connected to the memory data bus may use wide mode for data transfers. Wide mode is explained more fully later. Masters connecting to the 32 bit FBUS data section may not use wide mode. Slaves other than the MCU must ignore wide mode cycles.

All ports in FALCON are 32 bit ports to the FBUS (except for video memory as described above). Slaves ports must drive or receive data on the byte lanes appropriate to the address. All size/address combinations for 32 bit ports and burst reads defined by the 68030 are defined for the FBUS in 030 mode. All size/address combinations and line reads and writes defined by the 68040 are defined for the FBUS.

8.2 SIGNAL DEFINITION

8.2.1 Signal Mapping

FBUS to Processor Signal Correspondence		
FBUS signal	68030 signal	68040
A0-A31	A0-A31	A0-A31
D0-D31	D0-D31	D0-D31
TT0	CBREQ	TT0
TT1	’1’	TT1
TM0-TM2	FC0-FC2	TM0-TM2
SIZ0-SIZ1	SIZ0-SIZ1	SIZ0-SIZ1
LOCK	RMC	LOCK
AV	AVEC	AVEC
TA0	STERM	TA
TA1	CBACK	TBI
TE	BERR	TEA
R/W	R/W	R/W
CO	CIOUT	CIOUT
CI	CIIN	TCI
ST0	AS	TS
ST1	DS	TIP
I0-I2	IPL0-IPL2	IPL0-IPL2
IP	IPEND	IPEND
CLK	CLK	BCLK
RES	RESET	RSTI,RSTO [1]
BR	BR	BR [2]
BG	BG	BG [2]
BA	BGACK	BB [2]
BMODE	--	--
WID0-WID1	--	--
WEN	--	--
SIZ2		

Table 8.1

[1] The RSTI and RSTO signals of the 68040 will be combined by external circuitry into the FBUS signal RES. The ideal situation would be for the 68040 RSTO signal to drive the FBUS and the FBUS to drive the 68040 RSTI input without colliding (so the 040 reset instruction will not cause a reset of the processor), but this circuit is not yet designed and the final implementation may be different.

[2] An external bus arbiter circuit will be necessary when using a 68040. The processor BR, BG, and BB will connect to the arbiter circuit as will the FBUS signals BR, BG, and BA. The 68030 BR, BG, and BGACK will connect directly to the FBUS BR, BG, and BA respectively. Attempts will be made to maintain consistent timing for bus arbitration with either processor.

68030 signals not supported:
HALT
DBEN
CDIS
MMUDIS
REFILL
STATUS
DSACK0
DSACK1

68040 signals not supported:
TLN0-TLN1
UPA0-UPA1
LOCKE [1]
DLE
SC0-SC1
MI
CDIS
MDIS
PST0-PST3
TCK,TMS
TDI,TDO,TRST

Unsupported outputs are not connected. Unsupported inputs are tied to an inactive level.

[1] LOCKE may be used by the external arbiter circuit but is not defined for the FBUS.

8.2.2 FBUS signal definition

FBUS Signal Definition		
Signal	Direction (from master)	Function
A0-A31	O	Address bus
D0-D31	IO	Data bus
TT0-TT1	O	Transfer type [1]
TM0-TM2	O	Transfer modifier [1]
SIZ0-SIZ1	O	Transfer size [1]
LOCK	O	Bus cycle lock [1]
AV	I	Auto vector request [1,3]
TA0-TA1	I	Transfer acknowledge [1,3]
TE	I	Transfer error [1,3]
R/W	O	Read / write
CO	O	External cache inhibit [2]
CI	I	Cache inhibit [2,3]
ST0-ST1	O	Transfer strobes [1]
IO-I2	--	Interrupt level [2]
IP	I	Interrupt pending [2]
CLK	I	Bus clock
RES	IO	System reset [3]
BR	O	Bus request [3]
BG	I	Bus grant [4]
BA	O	Bus acknowledge [3]
BMODE	O	Bus mode select [5]
WID0-WID1	O	Wide mode select [3,6]
WEN	I	Wide mode enable [6]
SIZ2	O	Transfer size (wide mode only)

Table 8.2

[1] Signal or signal groups function or timing is different in 030 bus mode and 040 bus mode.

[2] These signals have meaning to the CPU. They will probably be ignored by most masters. CO and IP are always driven by the CPU.

[3] Indicates wire-or'd signals. Any master driving these signals must only drive them low, except AV, TA0-TA1, TE, CI, BA, and WID0-WID1 should be actively negated before release. Normally RES is an input to devices on the bus. Any master may drive RES low to reset the system, but note that the reset state returns bus ownership to the processor so an alternate master which drives RES low must release the bus and arbitrate after RES returns high.

[4] The BG signal is an output from the arbiter, the 68030 BG output or the external arbiter with a 68040. It is an input to the first master in the BG daisy chain.

[5] The BMODE signal should only change when bus ownership is transferred. The timing for BMODE is the same as BA and like BA, BMODE is a wire-or'd signal and should only be driven low. 030 mode bus masters need not drive BMODE since a pull-up will insure it is high. Slave ports must monitor BMODE to determine how they respond.

[6] Wide mode is provided to allow devices to connect to the 64 bit video memory data bus instead of the FBUS 32 bit data bus. These two busses are connected via the funnel logic which is controlled by the MCU. A separate section of this specification provides a complete description of wide mode operation.

8.3 FBUS 040 Mode

The BMODE signal is LOW for 040 mode.

8.3.1 General Description

Ports should be designed to work properly with 32mhz 68040 timing to support FBUS mode 0. Refer to the 68040 users manual for AC timing specifications. Designers of ports supporting line bursts should note that ports must ignore A0 and A1 and internally increment A2 and A3 during line burst cycles. Ports not supporting bursts must drive FBUS TA1 (040 TBI). Most ports may ignore CI and CO. Hardware on the mother board will drive CI for all accesses to IO space. System hardware will also drive the TE signal after about 16us if a cycle is not acknowledged. Only ports which respond to interrupt acknowledge cycles and do not supply a vector should drive AV. Ports should NOT respond to 040 alternate function code cycles. Only ports connecting to the 64 bit memory data bus and capable of 64 bit data transfers may respond when WID1 is low.

Masters should meet the AC timing specification for the 32mhz 68040. Masters which do not have internal caches should ignore CI and not drive CO or drive it high. Masters may not use the 040 alternate function code cycle. Masters which never do interrupt acknowledge cycles may ignore AV. Masters should not drive IP but may monitor it to see an interrupt is pending. Masters may monitor BR to see if another master needs the bus. Masters which keep the bus for long times should give up the bus as quickly as possible when another needs it. In NO case may a master keep the bus for more than 256 contiguous clock cycles.

8.3.2 040 Mode Signal Definitions

040 Mode Transfer Modifiers			
TM2	TM1	TM0	Normal and MOVE16 accesses only [1]
0	0	0	Data cache push access
0	0	1	User data access
0	1	0	User code access
0	1	1	MMU table search data access
1	0	0	MMU table search code access
1	0	1	Supervisor data access
1	1	0	Supervisor code access
1	1	1	Reserved

Table 8.3

[1] TM0-TM2 are connected directly to the 68040 TM0-TM2 lines. These lines carry the interrupt level during acknowledge cycles and the alternate access function code during alternate logical function cycles. See the 68040 user's manual for details of the TMx line usage.

040 Mode Transfer Type		
TT1	TT0	
0	0	Normal access
0	1	MOVE16 access
1	0	Alternate logical function code access
1	1	Acknowledge access

Table 8.4

TT0-TT1 connect directly to the 040 TT0-TT1.

040 Mode Size Encoding		
SIZ1	SIZ0	
0	0	Longword
0	1	Byte
1	0	Word
1	1	Line (16 bytes)

Table 8.5

SIZ0-SIZ1 connect directly to the 040 SIZ0-SIZ1.

040 Mode Data Bus Active Sections								
Xfer size	SIZ 1	SIZ 0	A1	A0	D31: D24	D23: D16	D15: D08	D07: D00
Byte	0	1	0	0	A			
Byte	0	1	0	1		A		
Byte	0	1	1	0			A	
Byte	0	1	1	1				A
Word	1	0	0	0	A	A		
Word	1	0	1	0			A	A
Line	1	1	X	X	A	A	A	A
Long	0	0	X	X	A	A	A	A

Table 8.6

8.4 FBUS 030 mode

The BMODE signal is HIGH for 030 mode.

8.4.1 030 Mode General Description

Ports should be designed to work properly with 32mhz 68030 timing to support FBUS mode 1. Refer to the 68030 users manual for AC timing specifications. Designers of ports supporting line bursts should note that ports must ignore A0 and A1 and internally increment A2 and A3 during line burst cycles. Ports supporting bursts must drive FBUS TA1 (030 CBACK) low. Most ports may ignore CI and CO. Hardware on the mother board will drive CI for all accesses to IO space. System hardware will also drive the TE (030 BERR) signal low after 16us if a cycle is not acknowledged. Only ports which respond to interrupt acknowledge cycles and do not supply a vector should drive AV. Only ports connecting to the 64 bit memory data bus and capable of 64 bit data transfers may respond when WID1 is low.

Masters should meet the AC timing specification for the 32mhz 68030. Masters which do not have internal caches should ignore CI and not drive CO or drive it high. Masters which never do interrupt acknowledge cycles may ignore AV. Masters should not drive IP but may monitor it to see an interrupt is pending. Masters may monitor BR to see if another master needs the bus. Masters which keep the bus for long times should give up the bus as quickly as possible when another needs it. In NO case may a master keep the bus for more than 256 contiguous clock cycles.

8.4.2 030 Mode Signal Definition

030 Mode Transfer Modifiers			
TM2	TM1	TM0	
0	0	0	Reserved
0	0	1	User data access
0	1	0	User code access
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Supervisor data access
1	1	0	Supervisor code access
1	1	1	CPU space access [1]

Table 8.7

[1] TM0-TM2 are connected directly to the 68030 FC0-FC2 lines. Note that interrupt acknowledge is a CPU space cycle and the slave must also decode A1-A3 and A16-A19. See the 68030 user’s manual for details of the FCx line usage.

030 Mode Transfer Type		
TT1	TT0	
1	0	Burst requested cycle
1	1	Normal cycle

Table 8.8

TT0 connects to the 030 CBREQ and TT1 is driven high.

030 Mode Size Encoding		
SIZ1	SIZ0	
0	0	Longword
0	1	Byte
1	0	Word
1	1	Three byte

Table 8.9

SIZ0-SIZ1 connect directly to the 030 SIZ0-SIZ1.

030 Mode Data Bus Active Sections								
Xfer size	SIZ 1	SIZ 0	A1	A0	D31: D24	D23: D16	D15: D08	D07: D00
Byte	0	1	0	0	A			
Byte	0	1	0	1		A		
Byte	0	1	1	0			A	
Byte	0	1	1	1				A
Word	1	0	0	0	A	A		
Word	1	0	0	1		A	A	
Word	1	0	1	0			A	A
Word	1	0	1	1				A
3-byte	1	1	0	0	A	A	A	
3-byte	1	1	0	1		A	A	A
3-byte	1	1	1	0			A	A
3-byte	1	1	1	1				A
Long	0	0	0	0	A	A	A	A
Long	0	0	0	1		A	A	A
Long	0	0	1	0			A	A
Long	0	0	1	1				A

Table 8.10

8.5 BUS ARBITRATION

The 68030 internal bus arbiter will perform arbitration for the FBUS when it is the system CPU. When a 68040 is used, an external arbiter circuit which conforms to similar timing will perform the bus arbitration. FBUS master designs should arbitrate correctly with a 32mhz 68030. Note that BMODE may only change states at the time BA changes. 040 masters may drive BMODE low using the same enable that is used to drive the BA signal low. 030 masters need not drive BMODE at all.

All masters should have a BG input and output so that a daisy chain may be implemented. A master which receives a low on its BG input should drive its BG output low as quickly as possible if it is not requesting the bus. A master requesting the bus should not drive its BG output low until it releases the bus (provided its BG input is still low). Also once a master has passed the BG along, should it subsequently decide to request the bus, may NOT drive BR low until its BG input has returned high. This will insure that the BG daisy chain sequence is preserved. The master connected directly to the FBUS BG signal is the first in the chain and so has the highest priority. Subsequent master's BG inputs are connected to the previous master's BG output in order of decreasing priority. The lowest priority master's BG output will not be connected.

Notes on bus arbitration:

To request the bus, a master drives BR low. A master may drive BR low anytime except when it is driving its BG output low. If a master is driving its BG output low and it decides to request the bus, it must wait until it drives its BG output high (as a result of its BG input going high) before it drives BR low.

If a masters BG input goes low and it is not driving BR low, then the master should drive its BG output low with a minimum of delay. If a masters BG input goes low and it is driving BR low, then it must maintain its BG output high.

A master may drive BA low only when; it is driving BR low AND its BG input is low AND its BG output is high AND ST0 and ST1 are high AND TA0 is high AND BA is high AND a rising edge occurs on CLK.

Once a master has driven BA low, it should release BR.

Once a master has driven BA low, it may drive appropriate bus lines to perform defined bus cycles.

Upon completion of the last bus cycle a master wishes to perform, the master should release the bus by tri-stating its bus drivers and releasing BA. Note that bus control signals ST0-ST1 and TA0-TA1 should not be tri-stated until they have been actively negated. BA should also be actively negated.

Once a master has released BA and if its BG input is low, it should drive its BG output low.

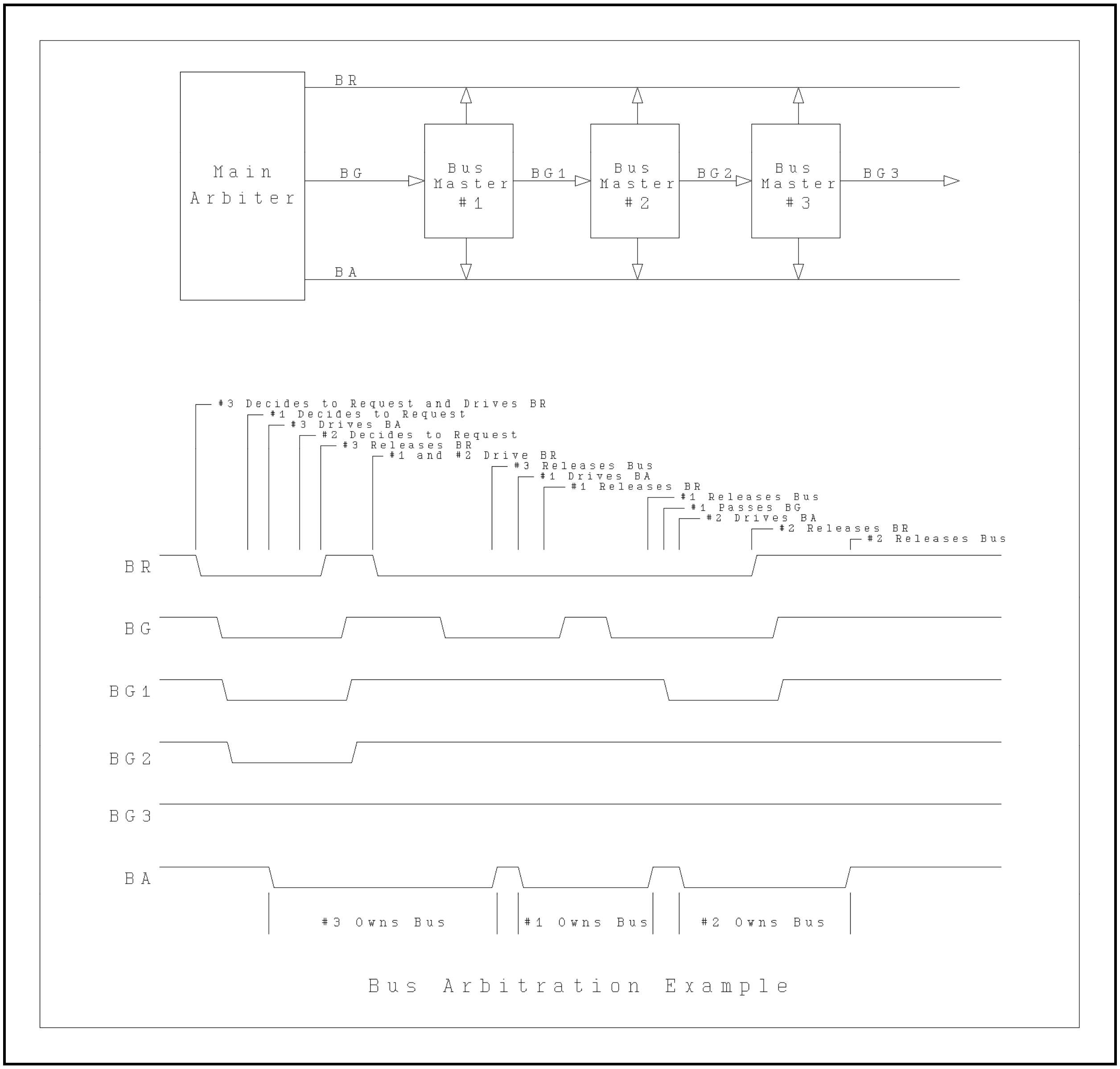

Figure 8.1

8.6 WIDE MODE

Four control signals (WID0, WID1, WEN, and SIZ2) have been added to the FBUS to allow devices to connect directly to the 64 bit video memory data bus. Thus connected, masters which exchange large amounts of data with wide slave ports (video memory is the only wide slave port currently defined or envisioned for FALCON) have twice the available bandwidth. Provision is also made for wide masters to access the 32 bit data bus via the funnel logic.

WID0 and WID1 are control signals from the master which indicate the type of transfer requested.

Wide Mode Signal Definition		
WID1	WID0	Type Transfer
1	1	Normal transfer
0	1	Direct access of video memory data bus
1	0	Access of FBUS via funnel logic
0	0	Access of wide device via funnel logic

Table 8.11

WID0-WID1 are normally kept high by pullups so standard FBUS masters need not drive these signals. A wide master can access the video memory using 64 bit (double longword) transfers by driving WID1 low. When WID1 is driven low the SIZ2 signal is used in conjunction with the SIZ0, SIZ1, A0, A1, and A2 signals to select the byte(s) of the 64 bit data.

Wide Mode Data Bus Active Sections (030 mode)

SIZ 210	A 210	D63: D56	D55: D48	D47: D40	D39 :32	D31: D24	D23: D16	D15: D08	D07: D00
000	000	A	A	A	A	A	A	A	A
001	000	A							
001	001		A						
001	010			A					
001	011				A				
001	100					A			
001	101						A		
001	110							A	
001	111								A
010	000	A	A						
010	001		A	A					
010	010			A	A				
010	011				A	A			
010	100					A	A		
010	101						A	A	
010	110							A	A
011	000	A	A	A					
011	001		A	A	A				
011	010			A	A	A			
011	011				A	A	A		
011	100					A	A	A	
011	101						A	A	A
100	000	A	A	A	A				
100	001		A	A	A	A			
100	010			A	A	A	A		
100	011				A	A	A	A	
100	100					A	A	A	A
101	000	A	A	A	A	A			
101	001		A	A	A	A	A		

101	010			A	A	A	A	A	
101	011				A	A	A	A	A
110	000	A	A	A	A	A	A		
110	001		A	A	A	A	A	A	
110	010			A	A	A	A	A	A
111	000	A	A	A	A	A	A	A	
111	001		A	A	A	A	A	A	A

Table 8.12

Wide Mode Data Bus Active Sections (040 mode)									
SIZ 210	A 210	D63: D56	D55: D48	D47: D40	D39 :32	D31: D24	D23: D16	D15: D08	D07: D00
000	XXX	A	A	A	A	A	A	A	A
001	000	A							
001	001		A						
001	010			A					
001	011				A				
001	100					A			
001	101						A		
001	110							A	
001	111								A
010	000	A	A						
010	010			A	A				
010	100					A	A		
010	110							A	A
100	000	A	A	A	A				
100	100					A	A	A	A
X11	XXX	A	A	A	A	A	A	A	A

Table 8.13

A wide master which does not connect to the 32 bit data bus may access the FBUS via the funnel logic by driving WID0 low. With WID0 driven low, all FBUS signals operate normally and devices on the FBUS see no difference to any other cycle. The MCU recognizes WID0 low and controls the funnel logic to couple the data busses for the transfer. A word of caution for wide masters operating in this way, the funnel logic will introduce delay in the data path. The master must allow for this delay so as not to violate FBUS timing and to insure proper operation.

The WID0 and WID1 low case is provided for wide masters containing registers which must be accessible to other normal FBUS masters. When such a wide master does not own the bus but detects an access to one of its internal registers, it may drive WID0 and WID1 low to indicate that the MCU should enable the funnel logic to couple the data busses. Again the funnel logic delay will be present and the wide master responding as a slave must compensate.

Discussion of wide mode so far has assumed that the video memory data bus is free when the data transfer begins. Standard and wide masters doing transfers which use the video memory data bus, such as CPU to memory or wide master to FBUS, arbitrate for the FBUS but not for the video memory data bus. The MCU however, also exchanges data between the video memory data bus and the video circuit without arbitrating the FBUS. The WEN signal is provided to prevent conflicts on the video memory data bus. The WEN is driven by the MCU and must be monitored by wide masters. When a transfer of a wide master via the video memory data bus is initiated, the MCU will drive WEN low if the video memory data bus is free. If WEN is not driven low, then the wide master may not drive the video memory data bus until it is. Also the MCU will not enable the funnel logic to drive the data bus either thus preventing conflicts. Note that in order to give the MCU time to drive the WEN signal and the master time to recognize it, there is an extra clock cycle inserted between the assertion of address and the assertion of the strobes ST0 and ST1 for all cycles where WID0 or WID1 is driven low.

Of course a wide master may connect to both the 64 bit video memory data bus and the 32 bit FBUS data bus. In that case the master may access or be accessed via standard FBUS cycles using the 32 bit data bus and only use the 64 bit bus for transfers to or from video memory. Such a master need only drive WID1 and must only monitor WEN when using the video memory data bus.

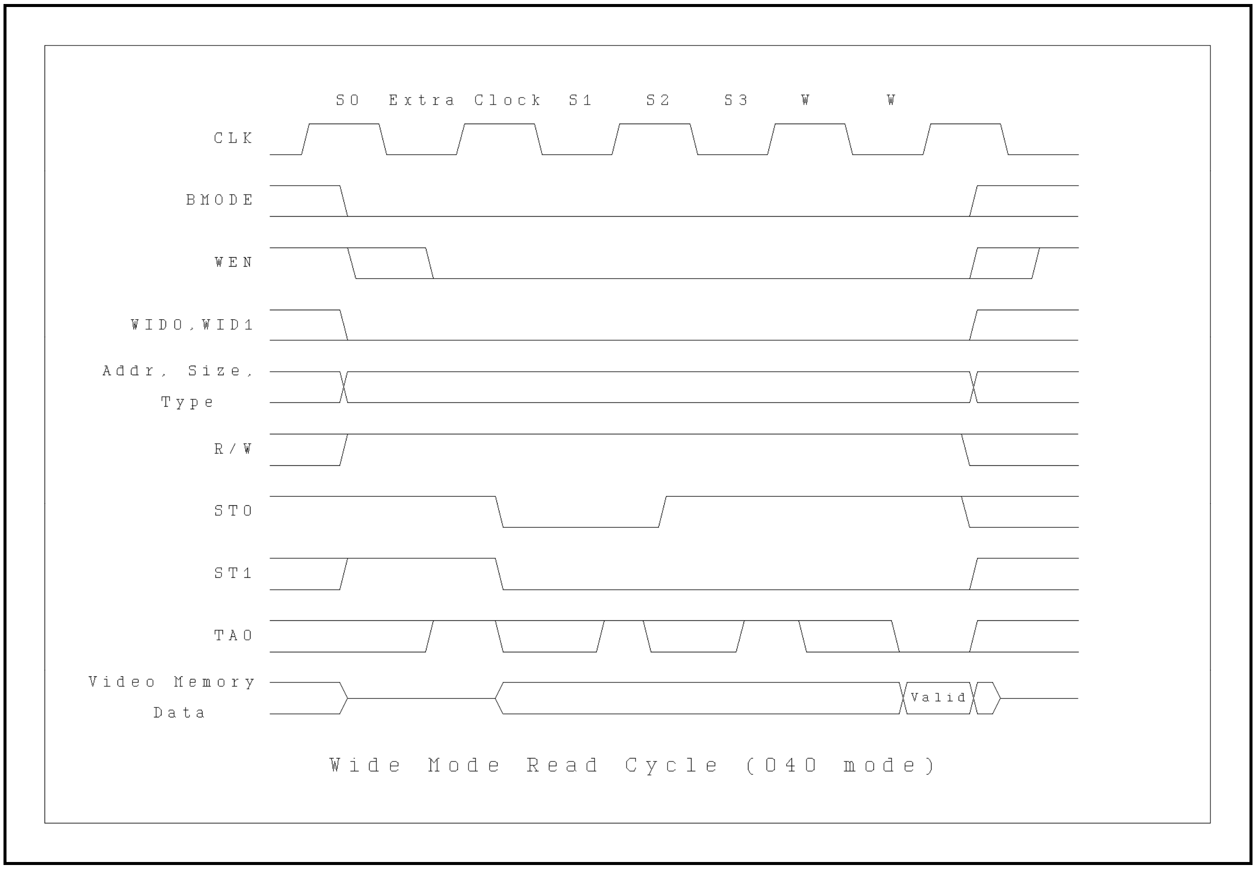
S0
Extra Clock
S1
S2
S3
W
W
CLK
BMODE
WEN
WID0,WID1
Addr, Size,
Type
R/W
ST0
ST1
TA0
Video Memory
Data
Valid
Wide Mode Read Cycle (040 mode)

Figure 8.2

Section 9 SYSTEM

9.1 Boot Sequence

The FALCON ROM will contain power-on diagnostics to verify that the processor, memory, and I/O subsystems are functional. The boot sequence begins after these diagnostic tests are successfully completed.

The boot process has three general stages:

1) The ROM boot procedure searches peripherals for boot code.

2) If found, this boot code, or device boot, is loaded by the ROM boot. The boot loaded from the hard disk is referred to as the "hard disk boot", or simply, "disk boot". The device boot consists of 512 bytes of boot code from the "boot sector" of the device and is loaded at a known point in dual-purpose RAM (see Section 2.3). Some devices, such as hard disks, load a second sector of boot code that calls code in the first sector.

3) The UNIX boot is loaded by the device boot. It is typically a moderately sized program (32 to 64 Kb) that actually loads the UNIX operating system. It need not be position independent if its location has been agreed upon with the device boot.

The ROM boot procedure's main purpose is to detect boot devices and load and run the device boot code on these devices. It checks the following devices in order:

1) Cartridge
2) Floppy drive 0
3) SCSI hard disk drives
4) Network (if present)]
5) ROM

For detailed information on the FALCON boot procedures, wait for the specification to be written.

9.2 Operating System

The FALCON is intended for use with Atari's TOS operating system or the UniPlus+ V.4.

9.3 Device Drivers

UniPlus+ is supplied with configuration tools which allow device drivers to be fully configurable with the UNIX kernel.

9.4 Networking Support

To permit its use in a wide variety of environments, FALCON may have software support for the Ethernet network, the Internet networking protocols (TCP/IP), Sun Microsystem’s Network File System (NFS), and/or LocalTalk network protocol.

9.5 Windowing User Interface

The UNIX operating system may include a windowing user interface built on the X-Windows (Version 11.4) package.

Section 10 Mechanical Considerations

Questions and comments on mechanical aspects of FALCON systems should be directed to Steven Chan and Ira Valenski.

10.1 Power Supply

See Atari specification C302695-001.

Approx 130 watts, 5V@22A, +12V@1.2A, -12V@500ma

The power supply MUST generate a "power good" signal (active high) that is asserted after the supply voltages are stable, and is removed before the supply voltages are removed.

Section 11 Memory, I/O, and Interrupt Map

11.1 MEMORY MAP as seen by the CPU

permissible access s - supervisor mode
u - user mode
r - read
w - write
b - burst
c - cachable

address	access	description
00000000-00000007	src	First 8 bytes of ROM bank 0 (Initial SP & PC)
00000008-000007FF	srwbc	Video RAM (protected)
00000800-0007FFFF	surwbc	Video RAM 512 kB
000FFFFF		512 kB + 512 kB
001FFFFF		2 MB
0027FFFF		2 MB + 512 kB
003FFFFF		2 MB + 2 MB
007FFFFF		8 MB
0087FFFF		8 MB + 512 kB
009FFFFF		8 MB + 2 MB
00EFFFFF		8 MB + 8 MB (15 MB usable)
00F00000-00F9FFFF	--	reserved
00FA0000-00FAFFFF	sur	cartridge port A
00FB0000-00FBFFFF	sur	cartridge port B
00FC0000-00FDFFFF	--	reserved
00FE0000-00FEFFFF	srw	Graphics Coprocessor
00FF0000-00FF7FFF	--	reserved
00FF8000-00FFFFFF	sr/srw	IO

```
01000000-010FFFFF  surwbc   Fast RAM (optional) 1 MB
         011FFFFF                          1 MB + 1 MB
         013FFFFF                          4 MB
         014FFFFF                          4 MB + 1 MB
         017FFFFF                          4 MB + 4 MB
         01FFFFFF                          16 MB
         020FFFFF                          16 MB + 1 MB
         023FFFFF                          16 MB + 4 MB
         02FFFFFF                          16 MB + 16 MB

01000000-7FFFFFFF  surwc    VME A32/D32
01400000-
01800000-
01C00000-
02000000-
02400000-
02800000-
02C00000-
03000000-
03400000-
03800000-
04000000-
04400000-
05000000-

80000000-BFFFFFFF  surw          VME A32/D32

C0000000-FCFFFFFF  surw          VME A32/D16

FD000000-FDFFFFFF  surw          VME A24/D32

FE000000-FEFEFFFF  surw          VME A24/D16

FEFF0000-FEFFFFFF  surw          VME A16/D16

============================================================
FF000000-FFFFFFFF  --       image of 00000000-00FFFFFF
*** except! ***
FFE00000-FFE7FFFF  surc          TOS rom bank0
FFE80000-FFEFFFFF  surc          TOS rom bank1

FFFFFF80  ro   xxxx xxxx         IOC ID byte
FFFFFF81  ro   xxxx xxxx         MCU ID byte
FFFFFF82  ro   xxxx xxxx         DMA-A ID byte
FFFFFF83  ro   xxxx xxxx         DMA-B ID byte
FFFFFF84  ro   xxxx xxxx         VMEC ID byte
FFFFFF85  ro   xxxx xxxx         GPU ID byte
```

11.1.1 Detail of IO Section

Addr 00FF0000 + N

```
N    acc  byte          use
---------------------------------------------------------
8001      srw  abc0 xxxx     Memory Configuration Register
                    a - ROM cycle speed select
                         0=slow
                         1=fast
                    b - Video DRAM access speed
                         0=slow
                         1=fast
                    c - Expansion DRAM access speed
                         0=slow
                         1=fast

8003      srw  axxx xxxx     Refresh Time Constant high byte
                    a - Refresh interval control
                         0=default
                         1=counter

8005      srw  xxxx xxxx     Refresh Time Constant low byte
8007      srw  a000 00bc     External Cache Control Register
                    a - Reset tag
                         0=reset
                         1=enable
                    b - Capture data cache push
                         0=no
                         1=yes
                    c - Clear on bus arbitration
                         0=yes
                         1=no

8009      srw  ssss bbBB     Video Memory Configuration Register
                    ssss - SIMM speed select
                    bb   - Bank 1 size select
                    BB   - Bank 2 size select

800B      srw  ssss bbBB     Fast Memory Configuration Register
                    ssss - SIMM speed select
                    bb   - Bank 1 size select
                    BB   - Bank 2 size select

8201      srw  xxxx xxxx     Video Base Address Even high byte
8203      srw  xxxx xxxx     Video Base Address Even mid byte
8205      srw  xxxx xxxx     Video Address Counter Even high byte
8207      srw  xxxx xxxx     Video Address Counter Even mid byte
8209      srw  xxxx x000     Video Address Counter Even low byte
820D      srw  xxxx x000     Video Base Address Even low byte

8213      srw  xxxx xxxx     Video Base Address Odd high byte
8215      srw  xxxx xxxx     Video Base Address Odd mid byte
8217      srw  xxxx x000     Video Base Address Odd low byte
```

```
821B      srw  xxxx xxxx    Video Address Counter Odd high byte
821D      srw  xxxx xxxx    Video Address Counter Odd mid byte
821F      srw  xxxx x000    Video Address Counter Odd low
8221      srw  a000 0bcd    Video Mode Control
                    a - disable video refresh
                        0=no
                        1=yes
                    b - Overwrite byte 0
                        0=yes
                        1=no
                    c - skip line enable
                        0=no
                        1=yes
                    d - skip phrase
                        0=no
                        1=yes

8240      rw   ---- -rrr    ST Color Palette Reg0 (RAMDAC)
8241           -ggg -bbb
8242      rw   ---- -rrr    ST Color Palette Reg1 (RAMDAC)
8243           -ggg -bbb
 ||
825E      rw   ---- -rrr    ST Color Palette Reg15 (RAMDAC)
825F           -ggg -bbb

8260      rw   ---- --ss    ST Video Mode (VTG)
                    ss - mode select
                        00 320x200, 4 plane
                        01 640x200, 2 plane
                        10 640x400, 1 plane
                        11 <reserved>

8262      rw   s--h -mmm    TT Video Mode (VTG)
8263           ---- bbbb    s - sample and hold mode
                        0 - off, 1 - on
                    h - hyper mono mode
                        0 - off, 1 - on
                    mmm - mode select
                        000 320x200x4
                        001 640x200x2
                        010 640x400x1
                        100 640x480x4
                        110 1280x960x1
                        111 320x480x8
                    bbbb - ST palette bank
```

```
8268      rw   mmmm mmmm     Psuedo Color Mask (RAMDAC)

8269      rw   smmm bbbb     FALCON shift mode (RAMDAC/VTG)
                    s - Sync on green enabled
                         0=yes
                         1=no

                    mmm - Mode select
                    000 =  1 bit per pixel (low res duochrome)
                    001 =  2 bit  "     "
                    010 =  4 bit  "     "  (low res)
                    011 =  8 bit  "     "
                    100 =  4 bit  "     "  (high res)
                    101 =  True color
                    110 =  Psuedo/True color
                    111 =  1 bit per pixel (high res duochrome)

            bbbb    Bank select


8280      rw   0000 xxxx xxxx xxxx HC  Horizontal counter
8282      rw   0000 xxxx xxxx xxxx     HHT  Horizontal half line
total
8284      rw   0000 xxxx xxxx xxxx     HBB  Horizontal blank begin
8286      rw   0000 xxxx xxxx xxxx     HBE  Horizontal Blank end
8288      rw   000h xxxx xxxx xxxx     HDB  Horizontal display
begin
                                  h - Line half.
                                       0=First half line
                                       1=Second half
828A      rw   000h xxxx xxxx xxxx     HDE  Horizontal display end
                                  h - Line half.
                                       0=First half line
                                       1=Second half
828C      rw   0000 xxxx xxxx xxxx     HSS  Horizontal sync start
828E      rw   0000 xxxx xxxx xxxx     HFS  Horizontal field sync
8290      rw   0000 xxxx xxxx xxxx     HEE  Horizontal
Equalisation End
8292      rw   0000 xxxx xxxx xxxx     VBT  Video Burst time
8294      rw   000x xxxx xxxx xxxx     HWC  Horizontal word count

82A0      rw   0000 xxxx xxxx xxxx     VC   Vertical counter
82A2      rw   0000 xxxx xxxx xxxi     VFT  Vertical Field Total
                                  i - Interlace
                                       0=Interlaced
                                       1=Non interlaced
82A4      rw   0000 xxxx xxxx xxxx     VBB  Vertical Blank Begin
82A6      rw   0000 xxxx xxxx xxxx     VBE  Vertical Blank End
82A8      rw   0000 xxxx xxxx xxxx     VDB0,VDB1     Vertical
Display Begin
82AA      rw   0000 xxxx xxxx xxxx     VDE0,VDB1     Vertical
Display End
82AC      rw   0000 xxxx xxxx xxxx     VSS  Vertical Sync Start
```

```
82C0      rw   abxd xfgh ijkl mnop     VMC  Video Master Control
                                  p - Hsync source
                                       0=Internal
                                       1=External
                                  o - Hsync level
                                       0=Active low
                                       1=Active high
                                  n - Hsync enable
                                       0=Disable
                                       1=Enable
                                  m - H-counter on
                                       0=Reset to 0
                                       1=Count
                                  l - Vsync source
                                       0=Internal
                                       1=External
                                  k - Vsync level
                                       0=Active low
                                       1=Active high
                                  j - Vsync enable
                                       0=Disable
                                       1=Enable
                                  i - V-counter on
                                       0=Reset to 0
                                       1=Count
                                  h - Csync level
                                       0=Active low
                                       1=Active high
                                  g - Csync enable
                                       0=Disable
                                       1=Enable
                                  f - Dotclk select
                                       0=VGA
                                       1=Super VGA
                                  e - Reserved
                                  d - Alternate fields
                                       0=disabled
                                       1=enabled
                                  c - Equalization pulses
                                       0=off
                                       1=on
                                  b - Wide Equ'n
                                       0=Disable
                                       1=Enable
                                  a - PAL/NTSC
                                       0=PAL (5 pulses)
                                       1=NTSC (6 pulses)
```

```
82C2      rw   0000 0000 0smm mvnr     VCO  Video control register
                                  r - Repeat lines
                                       0=Disabled
                                       1=Enabled
                                  Doesn't work correctly in
                                  interlaced mode
                                  n - Prescale dotclk
                                       0=No prescale
                                       1=Divide by 2
                                  v - Register select
                                       0=VDB0/VDE0
1=VDB1/VDE1
                                  mmm - Video mode*
                                  000 1bpp Duochrome
                                  001 2  bpp
                                  010 4  bpp
                                  011 8  bpp
                                  100 4  bpp hi-res
                                  101 24 bpp
                                  110 8/24 bpp
                                  111 1  bpp hi-res monochrome
                                  s - Line skip
                                       0=Disabled
                                       1=Skip alt' lines
```

*Note: The Video Mode lines are read only in the Video Control Register. They are set by writes to the ST, TT, or Falcon Video Mode Registers.

```
82E0      rw   0000 0000 0smm mvnr     VC1  Video control for TT
mode 0
82E2      rw   0000 0000 0smm mvnr     VC2  Video control for TT
mode 1
82E4      rw   0000 0000 0smm mvnr     VC3  Video control for TT
mode 2
82E6      rw   0000 0000 0smm mvnr     VC4  Video control for TT
mode 4
82E8      rw   0000 0000 0smm mvnr     VC5  Video control for TT
mode 6
82EA      rw   0000 0000 0smm mvnr     VC6  Video control for TT
mode 7

82EC      r    ---- ---- ---- -mmm     MID  Monitor Identification
bits
                             mmm  Currently undefined

8400      rw   ---- rrrr     TT Palette Reg0 (RAMDAC)
8401           gggg bbbb
8402      rw   ---- rrrr     TT Palette Reg1 (RAMDAC)
8403           gggg bbbb
 ||
85FE      rw   ---- rrrr     TT Palette Reg255 (RAMDAC)
85FF           gggg bbbb
```

```
8601     rw   xxxx xxxx     ACSI base upper upper byte (DMAC)

8604     rw   ---- ---- xxxx xxxx     DMA Data -wdc- (DMAC)

8606     w    ---- ---a bcde fghi     DMA Mode -wdl- (DMAC)

                   a - ACSI direction bit (DMAC)
                       0 - port into memory
                       1 - memory out to port
                   b - DMA request source
                       0 - not used
                       1 - FDC
                   cd - reserved
                   e - Block count register select
                   f - CS out select
                       0 - FDC
                       1 - not used
                   ghi - peripheral address (i not used)

8606     r    ---- ---- ---- -xxx     DMA Status (DMAC)

8609     rw   xxxx xxxx     ACSI base upper middle byte (DMAC) [1]
860B     rw   xxxx xxxx     ACSI base lower middle byte (DMAC) [1]
860D     rw   xxxx xxx0     ACSI base lower lower byte (DMAC) [1]
860F     rw   abcd sefg     Floppy density select

                   a - Disk change (input) pin (read only)
                   b - Media detect 2 (input) pin (read only)
                   c - Mode select 2 (output) pin
                       0 = low (reset)
                       1 = high
                   s - ACSI DMA in progress
                       0 = no
                       1 = yes
                   e - Media detect 1 (input) pin (read only)
                   f - Mode select 1 (output) pin
                       0 = low (reset)
                       1 = high
                   dg - FCCLK Frequency
                       00 = 8MHz (reset)
                       01 = 16MHz
                       10 = 32Mhz
                       11 = FCCLK Off

8701     rw   xxxx xxxx     SCSI DMA pointer upper
8703     rw   xxxx xxxx     SCSI DMA pointer upper-middle
8705     rw   xxxx xxxx     SCSI DMA pointer lower-middle
8707     rw   xxxx xxxx     SCSI DMA pointer lower

8709     rw   xxxx xxxx     SCSI DMA byte count upper
870B     rw   xxxx xxxx     SCSI DMA byte count upper-middle
870D     rw   xxxx xxxx     SCSI DMA byte count lower-middle
870F     rw   xxxx xxxx     SCSI DMA byte count lower
```

8710	r	xxxx xxxx	SCSI Data Residue Register upper-upper byte
8711	r	xxxx xxxx	SCSI Data Residue Register upper-middle byte
8712	r	xxxx xxxx	SCSI Data Residue Register lower-middle byte
8713	r	xxxx xxxx	SCSI Data Residue Register lower-lower byte
8715	rw	bzu0 00ed	SCSI DMA Control Register

b - bus error during DMA
(read only, cleared by read)
z - byte count zero
(read only, cleared by read)
u - data underrun
(read only, cleared by read)
e - DMA enable 0=off, 1=on
d - DMA direction:
0=in from port
1=out to port

8781	rw	xxxx xxxx	5380 Data Register
8783	rw	xxxx xxxx	5380 Initiator Command Register
8785	rw	xxxx xxxx	5380 Mode Register
8787	rw	xxxx xxxx	5380 Target Command Register
8789	rw	xxxx xxxx	5380 ID Select/SCSI Control Register
878B	rw	xxxx xxxx	5380 DMA Start/DMA Status Register
878D	rw	xxxx xxxx	5380 DMA Target Receive/Input Data
878F	rw	xxxx xxxx	5380 DMA Initiator Receive/Reset
8800	r	xxxx xxxx	PSG Read Data
8800	w	0000 xxxx	PSG Register Select
8802	w	xxxx xxxx	PSG Write Data

PSG Port A Bit Assignments

7	Printer Port ACK*
6	SPKON (internal speaker on when low)
5	Printer Port Strobe
4	*DTR (MFP-ST serial port)
3	*RTS (MFP-ST serial port)
2	*Floppy 1 Select
1	*Floppy 0 Select
0	*Floppy Side 0 Select

PSG Port B Bit Assignments

7-0	Printer Port bits 7-0

```
8804      r/rw     zabc defg xhij klmn     GPIO port
                            a - Serial port 2 RTS
                            b -    "     "  " DTR
                            c -    "     "  " RI
                            d -    "     "  " CTS
                            e -    "     "  " DSR
                            f - Serial port 1 DSR
                            g - Serial port 2 CD

                            h - Paralle port SLCTIN*
                            i -   "       "  INIT*
                            j -   "       "  AFD*
                            k -   "       "  ACK*
                            l -   "       "  PE
                            m -   "       "  SLCT
                            n -   "       "  ERROR*


8900      rw   0000 abcd     Sound DMA Control

                   ab - SINT control
                        00 - high
                        01 - playback channel
                        10 - record channel
                        11 - playback OR record
                   cd - SCNT control
                        00 - high
                        01 - playback channel
                        10 - record channel
                        11 - playback OR record

8901      rw   a0bc 00de     Sound DMA Control
                   a - Register Set Select
                        0 = playback registers
                        1 = record registers
                   b - Record Frame Repeat Select
                        0 = Single Frame
                        1 = Repeat
                   c - Record Enable
                        0 = Off (reset state)
                        1 = On
                   d - Playback Frame Repeat Select
                        0 = Single Frame
                        1 = Repeat
                   e - Playback Enable
                        0 = Off (reset state)
                        1 = On

8903      rw   xxxx xxxx     Frame Base Address upper-middle byte
8905      rw   xxxx xxxx     Frame Base Address lower-middle byte
8907      rw   xxxx xxxx     Frame Base Address lower-lower byte
```

```
8909      r    xxxx xxxx     Frame Address Counter upper-middle
byte
890B      r    xxxx xxxx     Frame Address Counter lower-middle
byte
890D      r    xxxx xxxx     Frame Address Counter lower-lower
byte

890F      rw   xxxx xxxx     Frame End Address upper-middle byte
8911      rw   xxxx xxxx     Frame End Address lower-middle byte
8913      rw   xxxx xxxx     Frame End Address lower-lower byte

8915      rw   xxxx xxxx     Frame Base Address upper-upper byte
8917      r    xxxx xxxx     Frame Address Counter upper-upper
byte
8919      rw   xxxx xxxx     Frame End Address upper-upper byte

8920      rw   0abc 0def gh00 00ij Playback Mode Control
                    abc Monitor Select
                         000 - tracks 1 & 2
                         001 - tracks 3 & 4
                         010 - tracks 5 & 6
                         011 - tracks 7 & 8
                         100 - tracks 9 & 10
                         101 - reserved
                         110 - reserved
                         111 - monitor off
                    def Active Tracks Select
                         000 - 2 tracks
                         001 - 4 tracks
                         010 - 6 tracks
                         011 - 8 tracks
                         100 - 10 tracks
                         101 - reserved
                         110 - reserved
                         111 - none
                    g - Mode (8 bit only)
                         0 = Stereo (reset state)
                         1 = Mono
                    h - Mode
                         0 = 8 bit samples (reset state)
                         1 = 16 bit samples
                    ij - Sample Rate Prescale
                         00 =  1280
                         01 =  640
                         10 =  320
                         11 =  160

8922      rw   xxxx xxxx     xxxx xxxx MICROWIRE Data register
8924      rw   xxxx xxxx     xxxx xxxx MICROWIRE Mask register

8961      w    xxxx xxxx     Real Time Clock Address Register
8963      rw   xxxx xxxx     Real Time Clock Data Register

8C01      rw   xxxx xxxx     SCC DMA pointer upper
```

8C03	rw	xxxx xxxx	SCC DMA pointer upper-middle
8C05	rw	xxxx xxxx	SCC DMA pointer lower-middle
8C07	rw	xxxx xxxx	SCC DMA pointer lower
8C09	rw	xxxx xxxx	SCC DMA byte count upper
8C0B	rw	xxxx xxxx	SCC DMA byte count upper-middle
8C0D	rw	xxxx xxxx	SCC DMA byte count lower-middle
8C0F	rw	xxxx xxxx	SCC DMA byte count lower
8C10	r	xxxx xxxx	SCC Data Residue Register upper-upper byte
8C11	r	xxxx xxxx	SCC Data Residue Register upper-middle byte
8C12	r	xxxx xxxx	SCC Data Residue Register lower-middle byte
8C13	r	xxxx xxxx	SCC Data Residue Register lower-lower byte
8C15	rw	bzu0 rsed	SCC DMA Control Register

b - bus error during DMA
(read only, cleared by read)
z - byte count zero
(read only, cleared by read)
u - data underrun
(read only, cleared by read)
r - Aux/SCC select
0=SCC
1=AUX
s - SCC channel
0=A
1=B
e - DMA enable 0=off, 1=on
d - DMA direction:
0=in from port
1=out to port

8C81	rw	xxxx xxxx	SCC A control register
8C83	rw	xxxx xxxx	SCC A data register
8C85	rw	xxxx xxxx	SCC B control register
8C87	rw	xxxx xxxx	SCC B data register
8CA0-8CBF		(odd bytes)	Expansion IO port
8E01	rw	xxxx xxx0	System Interrupt Mask (B7 - B1; B0 unused)
8E03	r	xxxx xxx0	System Interrupt State (before mask register)
8E05	rw	xxxx xxxa	System Interrupter (a=1 generate interrupt 1)
8E07	rw	xxxx xxxb	VME Interrupter (b=1 generate VME IRQ3)
8E09	rw	xxxx xxxx	General Purpose Reg 1
8E0B	rw	xxxx xxxx	General Purpose Reg 2

8E0D	rw	xxxx xxx0	VME Interrupt Mask (B7 - B1; B0 unused)
8E0F	r	xxxx xxx0	VME Interrupt State (before mask register)
9200	r	xxxx xxxx	System Configuration Straps
9800	--	XXXX XXXX	FALCON palette register 0
9801	rw	rrrr rrrr	
9802	rw	gggg gggg	
9803	rw	bbbb bbbb	
9804	--	XXXX XXXX	FALCON palette register 1
9805	rw	rrrr rrrr	
9806	rw	gggg gggg	
9807	rw	bbbb bbbb	
||			
9BFC	--	XXXX XXXX	FALCON palette register 255
9BFD	rw	rrrr rrrr	
9BFE	rw	gggg gggg	
9BFF	rw	bbbb bbbb	
A000-A1FF			IO expansion area 1 (IOCS1) [2]
A200-A3FF			IO expansion area 2 (IOCS2) [2]
FA01	rw	xxxx xxxx	MFP-1 GPIP
FA03	rw	xxxx xxxx	MFP-1 AER
FA05	rw	xxxx xxxx	MFP-1 DDR
FA07	rw	xxxx xxxx	MFP-1 IERA
FA09	rw	xxxx xxxx	MFP-1 IERB
FA0B	rw	xxxx xxxx	MFP-1 IPRA
FA0D	rw	xxxx xxxx	MFP-1 IPRB
FA0F	rw	xxxx xxxx	MFP-1 ISRA
FA11	rw	xxxx xxxx	MFP-1 ISRB
FA13	rw	xxxx xxxx	MFP-1 IMRA
FA15	rw	xxxx xxxx	MFP-1 IMRB
FA17	rw	xxxx xxxx	MFP-1 VR
FA19	rw	xxxx xxxx	MFP-1 TACR
FA1B	rw	xxxx xxxx	MFP-1 TBCR
FA1D	rw	xxxx xxxx	MFP-1 TCDCR
FA1F	rw	xxxx xxxx	MFP-1 TADR
FA21	rw	xxxx xxxx	MFP-1 TBDR
FA23	rw	xxxx xxxx	MFP-1 TCDR
FA25	rw	xxxx xxxx	MFP-1 TDDR
FA27	rw	xxxx xxxx	MFP-1 SCR
FA29	rw	xxxx xxxx	MFP-1 UCR
FA2B	rw	xxxx xxxx	MFP-1 RSR
FA2D	rw	xxxx xxxx	MFP-1 TSR
FA2F	rw	xxxx xxxx	MFP-1 UDR
FA81	rw	xxxx xxxx	MFP-2 GPIP
FA83	rw	xxxx xxxx	MFP-2 AER
FA85	rw	xxxx xxxx	MFP-2 DDR
FA87	rw	xxxx xxxx	MFP-2 IERA
FA89	rw	xxxx xxxx	MFP-2 IERB

FA8B	rw	xxxx xxxx	MFP-2	IPRA
FA8D	rw	xxxx xxxx	MFP-2	IPRB
FA8F	rw	xxxx xxxx	MFP-2	ISRA
FA91	rw	xxxx xxxx	MFP-2	ISRB
FA93	rw	xxxx xxxx	MFP-2	IMRA
FA95	rw	xxxx xxxx	MFP-2	IMRB
FA97	rw	xxxx xxxx	MFP-2	VR
FA99	rw	xxxx xxxx	MFP-2	TACR
FA9B	rw	xxxx xxxx	MFP-2	TBCR
FA9D	rw	xxxx xxxx	MFP-2	TCDCR
FA9F	rw	xxxx xxxx	MFP-2	TADR
FAA1	rw	xxxx xxxx	MFP-2	TBDR
FAA3	rw	xxxx xxxx	MFP-2	TCDR
FAA5	rw	xxxx xxxx	MFP-2	TDDR
FAA7	rw	xxxx xxxx	MFP-2	SCR
FAA9	rw	xxxx xxxx	MFP-2	UCR
FAAB	rw	xxxx xxxx	MFP-2	RSR
FAAD	rw	xxxx xxxx	MFP-2	TSR
FAAF	rw	xxxx xxxx	MFP-2	UDR
FC00	rw	xxxx xxxx	Keyboard ACIA Control	
FC02	rw	xxxx xxxx	Keyboard ACIA Data	
FC04	rw	xxxx xxxx	MIDI ACIA Control	
FC06	rw	xxxx xxxx	MIDI ACIA Data	

[1] A write to the ACSI DMA base upper middle, lower middle, or lower lower byte will clear the upper upper byte (8601).

[2] Two general purpose IO select signals, IOCS1 and IOCS2, are generated for IO addresses 00FFA000-00FFA1FF and 00FFA200-00FFA3FF, respectively. These pins minimize decoding when adding peripherals to the main board sometime in the future.

Any IO address not expressly listed in this section should be considered reserved. Any additions or changes to the FALCON memory map must be approved by the FALCON design team at Atari Microsystems in Dallas.

11.2 VME ADDRESS SPACE

11.2.1 ADDRESS SPACE SEEN BY A32 VME BUS MASTER

(logically equal to that seen by CPU, but without the address modifiers and size constraints)

address	size	use
00000000-00000007	D32	first 8 bytes of rom bank0 (restart vector)
00000008-000007FF	D32	Video RAM (protected)
00000800-0007FFFF 000FFFFF	D32	Video RAM 512 kB 512 kB + 512 kB

```
       001FFFFF         2 MB
       0027FFFF         2 MB + 512 kB
       003FFFFF         2 MB + 2 MB
       007FFFFF         8 MB
       0087FFFF         8 MB + 512 kB
       009FFFFF         8 MB + 2 MB
       00EFFFFF         8 MB + 8 MB (15 MB usable)

00FA0000-00FAFFFF  D16  cartridge port A
00FB0000-00FBFFFF  D16  cartridge port B

00F00000-00FFFFFF  D16  IO

01000000-010FFFFF  D32  Fast RAM (optional) 1 MB
         011FFFFF       1 MB + 1 MB
         013FFFFF       4 MB
         014FFFFF       4 MB + 1 MB
         017FFFFF       4 MB + 4 MB
         01FFFFFF       16 MB
         020FFFFF       16 MB + 1 MB
         023FFFFF       16 MB + 4 MB
         02FFFFFF       16 MB + 16 MB

01000000-|
01400000-|
01800000-|
01C00000-|
02000000-|
02400000-|\
02800000-|-FEFFFFFF     D32,D16  VME bus
02C00000-|/
03000000-|
03400000-|
03800000-|
04000000-|
04400000-|
05000000-|

FF000000-FFDFFFFF  D32  Image of video ram
FFE00000-FFEFFFFF  D32  TOS ROM
FFF00000-FFFFFFFF  D16  Image of IO
```

11.2.2 ADDRESS SPACE SEEN BY A24 VME BUS MASTER

<u>(sees only the ST Image)</u>

address	size	use
000000-000007	D32	first 8 bytes of rom bank0 (restart vector)
000008-0007FF	D32	video ram (protected)

000800-07FFFF	D32	Video RAM 512 kB
0FFFFF		512 kB + 512 kB
1FFFFF		2 MB
27FFFF		2 MB + 512 kB
3FFFFF		2 MB + 2 MB
7FFFFF		8 MB
87FFFF		8 MB + 512 kB
9FFFFF		8 MB + 2 MB
EFFFFF		8 MB + 8 MB (15 MB usable)
E00000-EFFFFF	D32	TOS ROMs
FA0000-FAFFFF	D16	cartridge port A
FB0000-FBFFFF	D16	cartridge port B
F00000-FFFFFF	D16	IO

11.2.3 VME CONTROLLER STARTING ADDRESSES

(Four pins strapped on the VME controller giving 14 starting addresses for VME space)

VME Starting Address Select				
SA3	SA2	SA1	SA0	Starting address
0	0	0	0	01000000
0	0	0	1	01400000
0	0	1	0	01800000
0	0	1	1	01C00000
0	1	0	0	02000000
0	1	0	1	02400000
0	1	1	0	02800000
0	1	1	1	02C00000
1	0	0	0	03000000
1	0	0	1	03400000
1	0	1	0	03800000
1	0	1	1	04000000
1	1	0	0	04400000
1	1	0	1	05000000
1	1	1	0	Reserved
1	1	1	1	Reserved

Table 11.1

11.3 INTERRUPT ASSIGNMENTS

Interrupt Assignments			
Interrupt level	System source [1]	Interrupt acknowledge response [2]	VME source
7	VME SYSFAIL	Auto vector	IRQ7
6	none	Vector	MFPs and IRQ6
5	none	Vector	SCC and IRQ5
4	VSYNC	Auto vector	IRQ4
3	none [3]	Auto vector	VME interrupter + IRQ3
2	HSYNC	Auto vector	IRQ2
1	System interrupter	Auto vector	IRQ1

Table 11.2

[1] Within each level, the system interrupt has higher priority than the VME interrupt. And, within the shared Level5 and Level6 interrupts, the part on the motherboard has higher priority than the VME interrupt.

[2] The VME interrupts use their interrupt status ID byte as their interrupt vector.

[3] The level 3 system interrupt mask must be enabled for the level 3 VME interrupt to actually be generated.

11.3.1 MFP Interrupt Assignments

MFP-ST (ST Compatible)

int	function
GPIP7	DMA Sound IRQ
GPIP6	Ring Indicator
TimerA	
RxRDY	
RxERR	
TxEMPTY	
TxERR	
TimerB	
GPIP5	FDC Interrupt
GPIP4	MIDI / Keyboard Interface
TimerC	
TimerD	
GPIP3	<reserved>
GPIP2	CTS
GPIP1	DCD
GPIP0	Parallel port

MFP 2

int	function
GPIP7	SCSI Controller IRQ (active high)
GPIP6	RTC IRQ (active low, cleared by reading RTC register 0x0C)
TimerA	
RxRDY	
RxERR	
TxEMPTY	
TxERR	
TimerB	
GPIP5	SCSI DMAC Interrupt (active low)
GPIP4	DiskChangeLine
TimerC	
TimerD	
GPIP3	Ring Indicator (SCC B)
GPIP2	SCC DMAC Interrupt (active low)
GPIP1	general purpose I/O pin
GPIP0	general purpose I/O pin

11.4 DMA/BUS MASTERSHIP PRIORITIES

priority	function
highest	SCSI DMA Channel
	AUX DMA Channel
	Floppy DMA channel
	Digital sound DMA channel
	VMEbus Masters
lowest	CPU

Section 12 Revisions

FBUS Version 1.0, October 22, 1990, original
December 7, 1990, clarifications to bus arbitration, redefinition of wide mode.

January 4, 1991, First draft of this document
April 8, 1991, Second draft
April 19, 1991 Third Draft
September 7, 1991 Forth Draft
December 3, 1991 Fifth Draft

Section 13 References

The VMEbus is defined by:

VMEbus International Trade Association (VITA), VMEbus Specification Manual, Revision C.1, October, 1985.

The Small Computer Systems Interface (SCSI) is defined by:

Adaptive Data Systems, Inc., SCSI Guidebook, Issue Number 2, June, 1985.

Specific SCSI device implementation details for typical devices are available in:

Adaptec Inc., Description of SCSI Command Set for Communications Devices, Revision 0.91, 1988.

Archive Corporation, VIPER Product Manual; SCSI Models 2060S and 2150S, Part No. 21391-001, June, 1988.

Maxtor Corporation, XT-4000S OEM Manual & Product Specification, 1014995, 1987.

Quantum Corporation, Q200 Series Programmer's Manual, 81-45416, Rev. B, 1987.

Details of the major commercially available chips used in the FALCON architecture are contained in:

General Instrument Corp., AY-3-8910/8912 Programmable Sound Generator Data Manual, February, 1979.

Logic Devices Inc., L5380/L53C80 CMOS SCSI Bus Controllers, September 1988

Motorola, Inc., MC68030 Enhanced 32-Bit Microprocessor User's Manual, 3rd. Edition, 1990.

Motorola, Inc., MC68881/MC68882 Floating-Point Coprocessor User's Manual, First Edition, 1987.

Motorola, Inc., MC68901 Multi-Function Peripheral, January, 1984.

Motorola, Inc., MC68040 32-Bit Microprocessor User's Manual, 1st. Edition, 1989.

Motorola, Inc., MC146818A Real-Time Clock Plus RAM (RTC), 1984.

Motorola, Inc., MC6850 Asynchronous Communications Interface Adapter, 19??.

Western Digital Corp., WD1772-02 Floppy Disk Formatter/Controller

Zilog, Inc., Z80C30 CMOS Z-BUS SCC / Z85C30 CMOS SCC Serial Communications Controller - Preliminary Product Specification, October, 1987.

Zilog, Inc., Z80C30 CMOS Z-BUS SCC / Z85C30 CMOS SCC Serial Communications Controller - Technical Manual, September, 1986.

The ST compatible hardware interfaces are also described in:

Atari Corporation, Engineering Hardware Specification of the Atari ST Computer System, January 7, 1986.

Atari Corporation, ST DMA Sound Technical Reference, July 18, 1988.

Atari Corporation, Intelligent Keyboard (ikbd) Protocol, February 26, 1985.

Atari Corporation, Atari ACSI/DMA Integration Guide, June 28, 1991.

All questions or comments about FALCON or this specification should be directed to:

John D. Horton Jr.
FALCON Design Team
Atari Microsystems
4115 Keller Springs Rd. #200
Dallas, TX 75244
(214) 713-9111   Fax (214) 713-9040