Fat Agnus Chip

DESCRIPTION

This specification describes the Fat Agnus chip, an N-channel HMOS DMA Controller. This IC device is able to produce, in a 68000 microprocessor environment, DMA addresses by using a RAM Address Generator and a Register Address Encoder. This device contains 25 DMA channel controllers, including the Blitter, Bitplanes, Copper, Audio, Sprites, Disk and Memory refresh.

The IC accepts a 28.63636 MHz crystal clock for the purpose of generating 7.16 MHz and 3.58 MHz system clocks, dynamic RAM interface for addressing up to 1 megabyte of memory and NTSC video synchronization pulses.

Refer to Figure 6.1 for pin configuration, Figure 6.2 for IC block diagram and Table 6-1 for pin description.

This IC device is equivalent to an 8370.

Warning

Improved versions of the Amiga custom chips are under development. These chips are intended to be software compatible with the existing chips. Writing incorrect values to reserved bits, accessing undefined register addresses, reading write-only registers or excessive cleverness may lead to compatibility problems.

CONFIGURATION

This IC device is configured in a standard 84 pin plastic chip carrier package.

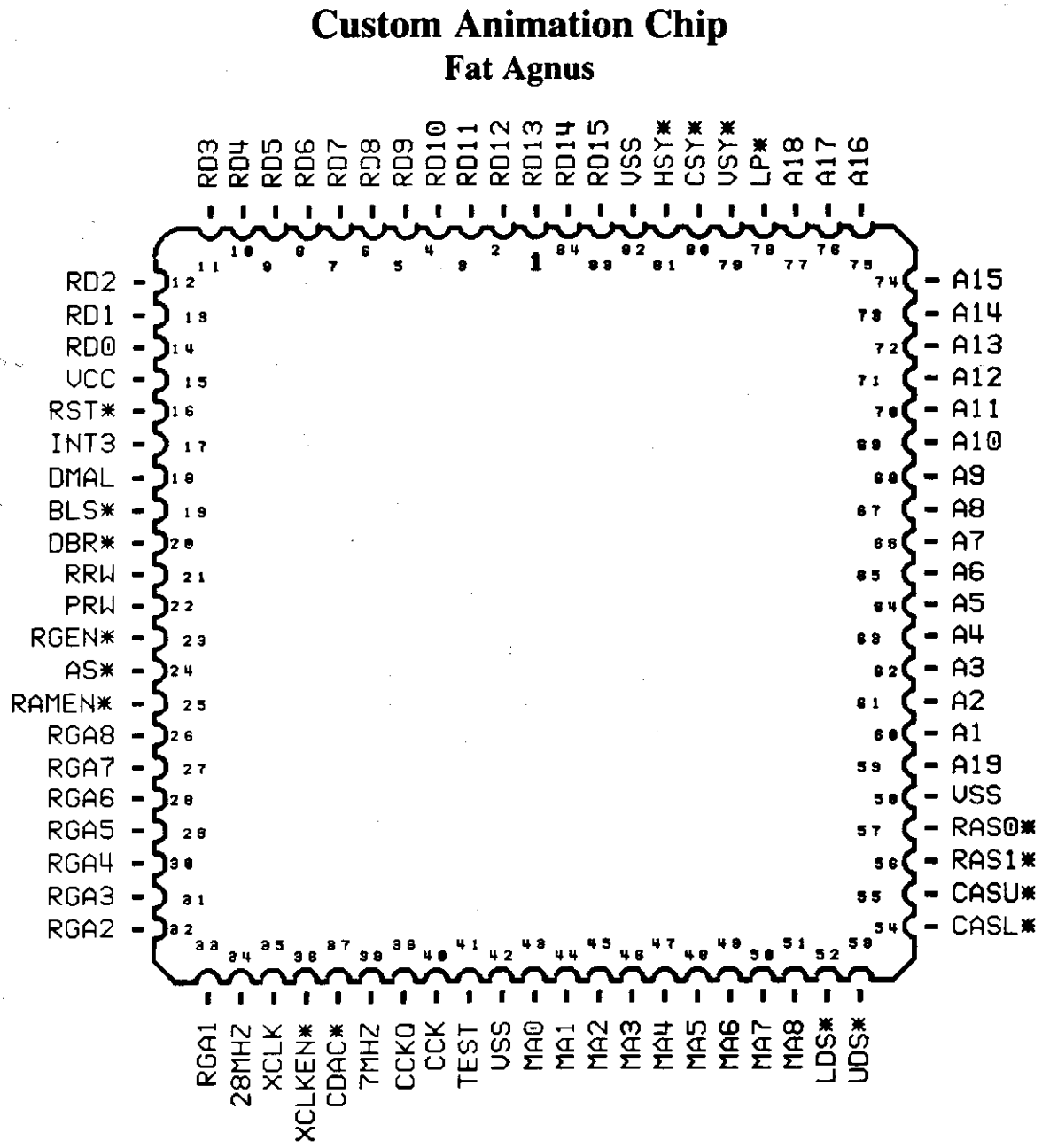


Figure 6.1 Configuration

Fat Agnus Block Diagram

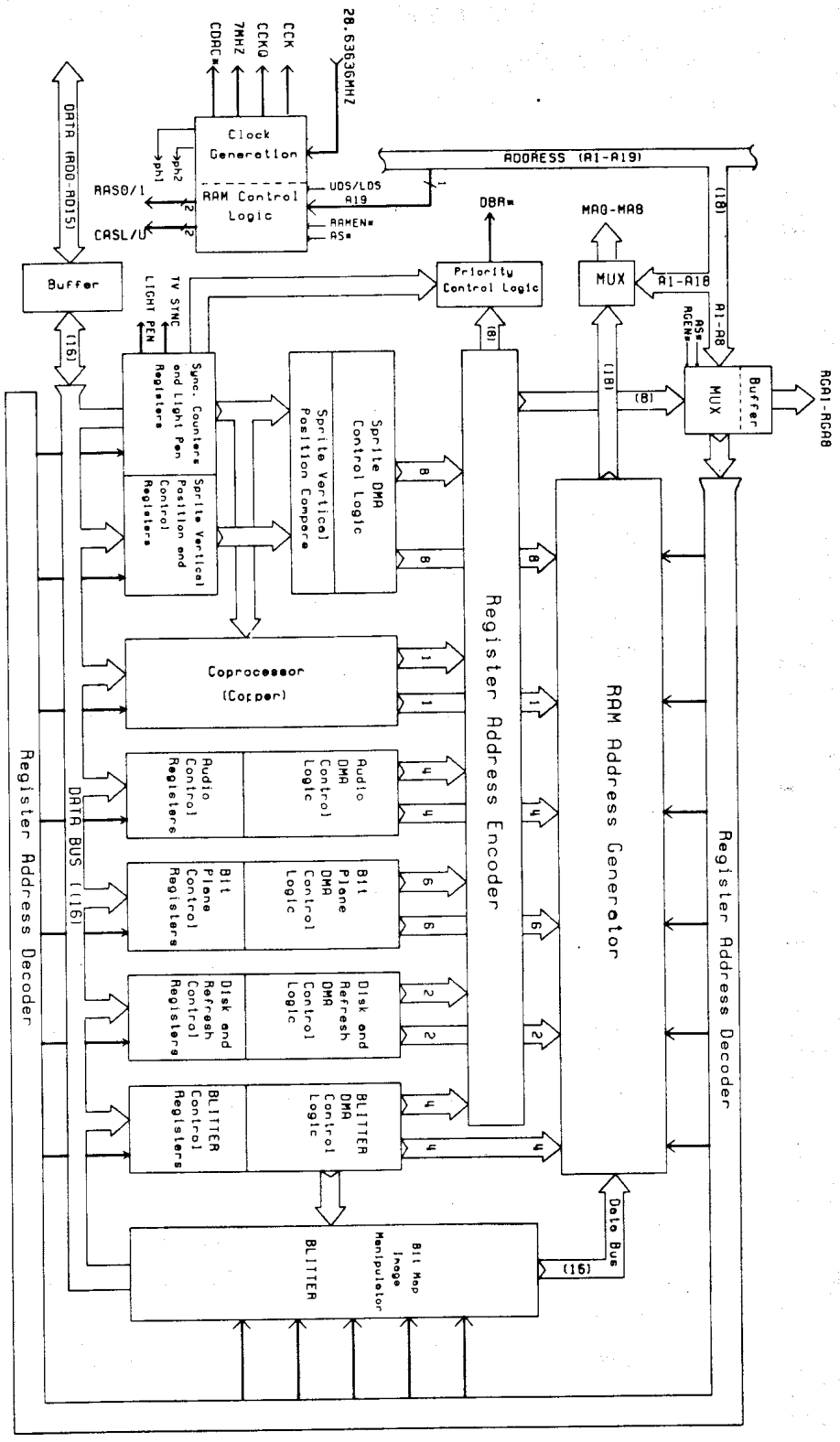


Figure 6.2 Block Diagram

Table 6-1 Pin Description

PIN NAME	PIN NUMBER	SIGNAL DIRECTION	DESCRIPTION
A19-A1	59 thru 77	IN	Address bus—A1 to A8 are used by the processor to select the internal registers and put an address code on the RGA lines to select registers outside the device. The processor uses A1 to A18 to generate multiplexed DRAM addresses on the MA outputs. The A19 line is used to indicate which RAS line is activated. If A19 is high, RAS1* is asserted; if low, RAS0* is asserted.
RD15-RD0	1 thru 14 and 83 & 84	I/O	This data bus is buffered and is used by the processor to access the device registers. The data bus is also accessed during DMA operations.
AS*	24	IN	Active low. This input is the processor address strobe signal. When asserted, it indicates that the address lines (A1 to A19) are valid.
RGEN*	23	IN	Active low. When this signal is asserted along with AS*, the processor uses A1 to A8 to access one of the device registers or put a value on the RGA outputs to select registers outside the device.
RAMEN*	25	IN	Active low. When this signal is asserted together with AS*, the processor is doing a DRAM access. The processor supplies an address on the A1 to A18 inputs and the device multiplexes this address onto the MA outputs; during the same cycle, the processor controls the A19 line to select one of the RAS lines.
PRW	22	IN	This signal defines the data bus transfer as a read or write cycle to memory. The signal is only enabled when the processor is undergoing a DRAM access. A low on this signal signifies a processor write cycle to memory; a high indicates a processor read cycle from memory.
RRW	21	OUT	The device controls this signal to indicate either a DMA or processor DRAM read/write access. In both cases, a low on this line indicates a write operation and a high indicates a read operation.

PIN NAME	PIN NUMBER	SIGNAL DIRECTION	DESCRIPTION
MA0-MA8	43 thru 51	OUT	Output bus. This 9 bit output bus provides multiplexed addresses to DRAMs. This bus operates in two cycles. The first cycle provides the DRAMs with the row address, the second cycle with the column address. It includes full 512K addressing for use with 256KX1 DRAMS. The IC only activates this bus when the processor is doing a DRAM access (RAMEN* is low) or when the device itself is performing a DMA data transfer (DBR* is low).
LDS*	52	IN	Active low. This input is the processor lower data strobe. It is enabled only during a processor DRAM access and forces the IC to assert CASL*.
UDS*	53	IN	Active low. This input is the processor upper data strobe. It is enabled only during a processor DRAM access and forces the IC to assert CASU*.
CASL*	54	OUT	Active low. This output strobes the column address into the DRAMS and corresponds to the low byte of the data word.
CASU*	55	OUT	Active low. This output strobes the column address into the DRAMS and corresponds to the high byte of the data word.
RAS0*	57	OUT	Active low. This output is used to strobe the row address into the DRAMs. This signal is asserted only if the processor is doing a DRAM access and A19 is low or if the IC is performing a DMA cycle (DBR* is low). RAS0* corresponds to the lower 512K bytes of memory.
RAS1*	56	OUT	Active low. This output is used to strobe the row address into the DRAMs. This signal is asserted only if the processor is doing a DRAM access and A19 is high. The signal is not asserted when the device is doing a DMA cycle. RAS1* corresponds to the upper 512K bytes of memory.
DBR*	20	OUT	Active low. The device asserts this signal to indicate that a DMA cycle is underway. The device performs only DMAs on the lower 512K bytes of memory when DBR* is low and RAS0* is asserted. The only exception is when the device is performing a DRAM refresh, in which case RAS0*, RAS1* and DBR* are all asserted. The device also asserts both CASL* and CASU* during DMAS except on a DRAM refresh cycle.

PIN NAME	PIN NUMBER	SIGNAL DIRECTION	DESCRIPTION
RGA8-RGA1	26 thru 33	OUT	Output bus. The 8 bit output bus allows the device and the processor to access registers located outside the device.
HSY*	81		This line is bidirectional and buffered. This signal is the horizontal synchronization pulse and is NTSC compatible. When set as an input, an external video source drives this signal to synchronize the horizontal beam counter.
VSY*	79		This line is bidirectional and buffered. This signal is the vertical synchronization pulse and is NTSC compatible. When set as an input, an external video source drives this signal to synchronize the vertical beam counter.
CSY*	80	OUT	This signal is the composite video synchronization pulse and is NTSC compatible.
LP*	78	OUT	Active low. This input is used to indicate when the light pen is coincident with the monitor beam.
RST*	18	IN	Active low. This input initializes the device to a known state.
INT3*	17	OUT	Active low. The device asserts this line to indicate that the blitter has completed the requested data transfer and that the blitter is then ready to accept another task.
DMAL	18	IN	Active high. When this signal is enabled, it indicates that an external device is requesting audio and/or disk DMA cycles to be executed by the device.
BLS*	19	IN	Active low. When this line is asserted, the device suspends its blitter operation and allows the processer to have control of the cycle.
28MHZ	34	IN	This is a 28.63636MHz input clock that provides the master time base for the device. This clock is enabled only when XCLKEN* is high.
XCLK	35	IN	This input is an alternate master clock to the device. It is enabled when XCLKEN* is low. This input is used to synchronize the device with an external video source.

PIN NAME	PIN NUMBER	SIGNAL DIRECTION	DESCRIPTION
XCLKEN*	36	IN	This input is used to select the master clock to the device. If it is high, the 28MHz input is enabled; if low, the XCLK is enabled.
CCK	40	OUT	This signal is a clock, which is obtained after dividing the 28.63 MHZ clock by eight. It is also known as the color clock frequency for NTSC applications.
CCKQ	39	OUT	This clock is the CCK clock shifted by 90 degrees.
7MHZ	38	OUT	This clock is obtained after dividing the 28MHZ clock by four.
CDAC*	37	OUT	This clock is obtained after inverting the 7MHZ clock and shifting it by 90 degrees.
TEST	41	IN	Active high. When this signal is asserted, it disables the processor cycle and the 8370 internal registers can be accessed on every CCK clock cycle.

MODES OF OPERATION

General Information

This device is an address generator type IC. Its main function is as a RAM address generator and register address encoder that produces all DMA addresses from 25 channels.

The block diagram (Figure 6.3) for this device shows the DMA control and address bus logic. The output of each controller indicates the number of DMA channels driving the Register Address Encoder and RAM Address Generator.

The Register Address Encoder is a simple PLA type of structure that produces a predetermined address on the RGA bus whenever one of the DMA channels is active.

The RAM Address Generator contains an 18-bit pointer register for each of the 25 DMA channels. It also contains pointer restart (backup) registers and jump registers for six (6) of the channels. A full 18-bit adder carries out the pointer increments and adds for jumps.

The priority control logic looks at the pipe-lined DMA requests from each controller and stages the DMA cycles based upon their programmed priority and sync counter time slot. Then it signals the processor to get off the bus by asserting the DBR line.

The following is a brief description of the device's major operational modes.

Blitter

The procedure for moving and combining bit-mapped images in memory received the name Bit Blit from a computer instruction that did block transfers of data on bit boundaries. These routines became known as Bit Blitters or Blitters. The Blitter DMA Controller is preloaded with the address and size of three source images (A, B, and C) and one destination (D) in the dynamic RAM (refer to Figure 6.3). These images can be as small as a single character or as large as twice the screen size. They can be full images or smaller windows of a larger image. After one work of each source image is sequentially loaded into the source buffer (A, B, C) they are shifted and then combined together in the logical unit to perform image movement overlay, masking, and replacements. The result is captured in the destination buffer (D) and sent back to the RAM memory destination address.

This operation is repeated until the complete image has been processed. The unit has extensive pipelining to allow for shifter and logic unit propagation time, while the next set of source words is being fetched.

A control register determines which of 256 possible logic operations is to be performed as the source images are combined and how far they are to be moved (barrel shifted). In addition to the image combining and movement powers, the Blitter can be programmed to do line drawing or area fill between lines.

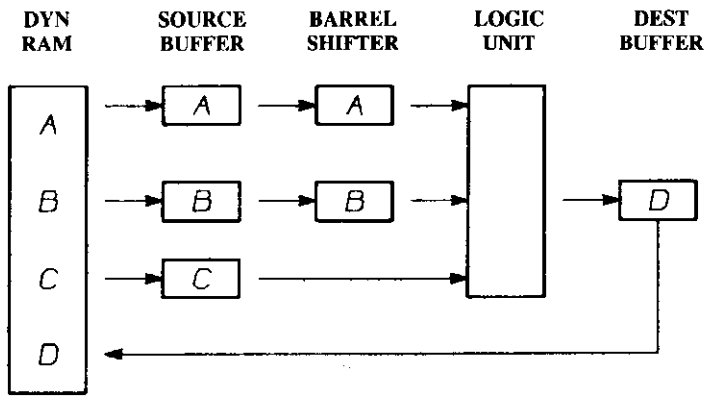


FIGURE 6.3. Blitter Block Diagram

Bitplane Addressing

Some computer bitmap displays are organized so that the bitplanes for each pixel are all located within the same address. This is called *pixel addressing.* If the entire data word of one address is used for a single pixel with 8 bit planes, the data word will look like this. (numbers are bitplanes):

12345678--------

The data compression can be improved by packing more than one pixel into a single address like this:

1234567812345678

or like this, if there are only 4 bitplanes:

1234123412341234

The IC device uses a bitmap technique called *Bitplane Addressing.* This separates the bitplanes in memory. To create a 4 plan (16 color) image, the bitplane display DMA channels fetch from 4 separate areas of memory like this:

1111111111111111
2222222222222222
3333333333333333
4444444444444444

These are held in buffer registers and are used together as pixels, one bit at a time, by the display (left to right).

This technique allows reduced odd numbers of bitplanes (such as 3 or 5) while maintaining packing efficiency and speed. It also allows grouping bitplanes into two separate images, each with independent hardware high speed image manipulation, line draw, and area fill.

DMA Channel Functions

Each channel has an 18 bit RAM address pointer that is placed on the MA memory address bus and is used to select the location of the DMA data transfer from anywhere in 256K words (512K bytes) of RAM.

An eight bit destination address is simultaneously placed on the register address bus (RGA), sending the data to the corresponding register.

Figure 6.4 shows a typical DMA channel; almost all channels have RAM as source and chip registers as destination.

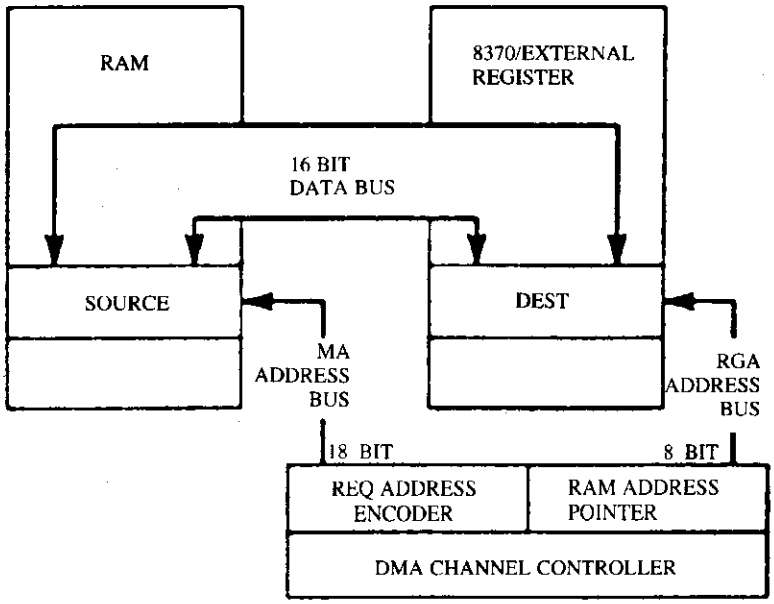


FIGURE 6.4. DMA CHANNEL (TYPICAL)

The pointer must be preloaded and is automatically incremented each time a data transfer occurs.

Each controller utilizes one or more of these DMA channels for its own purposes. The following is a brief summary of these controllers and the DMA channels they use.

A-BLITTER (4 CHANNELS)

The Blitter uses four DMA channels, three sources and one destination as previously described.

Once the Blitter has been started, the four DMA channels are synchronized and pipelined to automatically handle the data transfers without further processor intervention. The images are manipulated in memory, independent of the display (bitplane DMA).

B-BITPLANE (SIX (6) CHANNELS)

The bitplane controller continuously (during display) transfers display data from memory to display buffer registers. There are six DMA channels to handle the data from six independent bit planes. The buffers convert this bitplane data into pixel data for the display.

Each bitplane can be a full image or a window into an image that is up to four times the screen size. They can be grouped into two separate images, each with its own color registers.

C-COPPER (ONE (1) CHANNEL)

The Copper is a coprocessor that uses one of the DMA channels to fetch its instructions. The DMA pointer is the instruction counter and must be preloaded with the starting address of Copper's instructions.

The Copper can move (write) data into chip registers. It can skip, jump, and wait (halt). These simple instructions give great power and flexibility because of the following features.

When the Copper is halted, it is off the data bus, using no bus cycles until the wait is over. The programmed wait value is compared to a counter that keeps track of the TV beam position (Beam Counter) and when they are equal, the Copper will resume fetching instructions.

It can cause interrupts, reload the color registers, start the Blitter or service the audio. It can modify almost any register inside or outside the IC device, based on the TV screen coordinates given by the Beam Counter and the actual address encoded on the RGA bus.

D-AUDIO (FOUR (4) CHANNELS)

There are four audio channels, all of which are located outside of the DMA Controller IC. Each controller is independent and uses one DMA channel from the DMA Controller IC and fetches its data during a dedicated timing slot within horizontal blanking. This is accomplished by a controller asserting the DMAL input on the DMA Controller.

E-SPRITES (EIGHT (8) CHANNELS)

There are eight independent Sprite controllers, each with its own DMA channel and its own dedicated time slot for DMA data transfer. Sprites are line buffered objects that can move very fast because of their position are controlled by hardware registers and compacitors.

Each Sprite has two 16 bit data registers that define a 16 pixel wide Sprite with four colors. Each has a horizontal position register, a vertical start position register and a vertical stop position register. This allows variable vertical size sprites.

The Sprite DMA controller fetches image and position data automatically from anywhere in 512K of memory.

Sprites can be run automatically in DMA mode or they can be loaded and controlled by the microprocessor.

Each Sprite can be reused vertically as often as desired. Horizontal reusing is also possible with microprocessor control.

F-DISK (ONE (1) CHANNEL)

The disk controller, which is located outside of the DMA controller, uses a single DMA channel from the device. The controller uses this DMA time slot for data transfer and can read or write a block of data up to 128K anywhere in 512K of memory.

G-MEMORY REFRESH (ONE (1) CHANNEL)

The refresh controller uses a single DMA channel with its own time slots. It places RAS addresses on the memory address bus (MAS) during these slots, in order to refresh the dynamic RAM. Memory is refreshed on every roster line.

During the DMA no data transfer actually takes place. The register address bus (RGA) is used to supply video synchronizing codes. At this time, RAS0* and RAS1* are low and CASU* and CASL* are inactive.

RAM and Register Addressing

The device generates RAM addresses from two sources, the processor or from the device performing DMA cycles selected by a multiplexer. This multiplexer allows the processor to access RAM when AS* and RAMEN* are both low. At this time, the device also multiplexes the processor address (A1-A18) onto the MA bus. The device places A1 to A8 & A17 on the MA0 to MA9 outputs, respectively, during the row address time and places A9 to A16 & A18 on the MA0 to MA9, respectively, during the column address time. The A19 line is used by the IC to determine which RAS line is to be asserted. If A19 is low, RAS0* is enabled, and if high, RAS1* is enabled. The device also senses the LDS* and UDS* inputs to determine which CAS to drop. If LDS* is low, the IC will drop CASL*; if UDS* is low, CASU* is dropped.

When the device needs to do a DMA cycle, the multiplexer disables the processor from accessing RAM by asserting the Data Bus Request line (DBR*). At this time, the device multiplexes its generated RAM address onto the MA lines and will only make RAS0* 90 low, unless it is a refresh cycle where RAS1* will also go low. During a DMA cycle, the IC device also asserts both CASU* and CASL*, unless it is a refresh cycle where they both remain inactive.

The device also generates RGA addresses from either the processor or device DMAs, each of which is selected by another internal multiplexer. This multiplexer allows the processor to perform a register read/write access when AS* and RGEN* are both low. The device then takes the low order byte of the processor address A1 to A8 and reflects its value on the RGA output bus RGA1 to RGA8. The device will reflect the status of PRW input on the RRW output line, to indicate a memory read or write operation.

During a device DMA cycle, the multiplexer prevents the processor from doing a register access by asserting the DBR* line. The device then places the contents of its register address encoder onto the RGA bus.

REGISTER DESCRIPTION

This DMA controller device contains 97 registers that can be accessed after the following conditions have been met: the state of AS* and RGEN* must be an active low level and the least 8 significant address bits (A1 thru A8) must contain the valid address of the register to be accessed.

The following is a detailed description of the register set.

REGISTER	FUNCTION
AUD x LCH	*Audio channel x location (high 3 bits)*
AUD x LCL	*Audio channel x location (low 15 bits)*

This pair of registers contains the 18 bit starting address (location) of Audio channel x (x = 0,1,2,3) DMA data. This is not a pointer register and therefore only needs to be reloaded if a different memory location is to be outputted.

BLT x PTH *Blitter pointer to x (high 3 bits)*
BLT x PTL *Blitter pointer to x (low 15 bits)*

This pair of registers contains the 18 bit address of Blitter source (x = A,B,C) or dest. (x = D) DMA data. This pointer must be preloaded with the starting address of the data to be processed by the blitter. After the Blitter is finished it will contain the last data address (plus increment and modulo).

LINE DRAW: BLTAPTL is used as an accumulator register and must be preloaded with the starting value of (2Y-X) where Y/X is the line slope. BLTCPT and BLTDPT (both H and L) must be preloaded with the starting address of the line.

BLT x MOD *Blitter Modulox*

This register contains the Modulo for Blitter source (x = A,B,C) or Dest (x = D). A Modulo is a number that is automatically added to the address then points to the start of the next line. Each source or destination has its own Modulo, allowing each to be a different size, while an identical area of each is used in the Blitter operation.

LINE DRAW: BLTAMOD and BLTBMOD are used as slope storage registers and must be preloaded with the values (4Y-4X) and (4Y) respectively. Y/X = line slope BLTCMOD and BLTDMOD must both be preloaded with the width (in bytes) of the image into which the line is being drawn (normally 2 times the screen width in words).

BLTAFWM *Blitter first word mask for Source A*
BLTALWM *Blitter last word mask for Source A*

The patterns in these two registers are "anded" with the first and last words of each line of data from Source A into the Blitter. A zero in any bit overrides data from Source A. These registers should be set to all "ones" for fill mode or for line drawing mode.

BLT x DAT *Blitter source x data register*

This register holds Source x (x = A,B,C) data for use by the Blitter. It is normally loaded by the Blitter DMA channel, however, it may also be preloaded by the microprocessor.

LINE DRAW: BLTADAT is used as an index register and must be preloaded with 8000. BLTBDAT is used for texture. It must be preloaded with FF if no texture (solid line) is desired.

BLTDDAT *Blitter destination data register*

This register holds the data resulting from each word of Blitter operation until it is sent to a RAM destination. This is a dummy address and cannot be read by the micro. The transfer is automatic during Blitter operation.

BLTCON0 *Blitter control register 0*
BLTCON1 *Blitter control register 1*

These two control registers are used together to control Blitter operations. There are 2 basic modes, area and line, which are selected by bit 0 of BLTCON1, as shown below.

AREA MODE ("normal")

BIT#	BLTCON0	BLTCON1
15	ASH3	BSH3
14	ASH2	BSH2
13	ASH1	BSH1
12	ASA0	BSH0
11	USEA	X
10	USEB	X
09	USEC	X
08	USED	X
07	LF7	X
06	LF6	X
05	LF5	X
04	LF4	EFE
03	LF3	IFE
02	LF2	FCI
01	LF1	DESC
00	LF0	LINE(=0)

ASH3-0	Shift value of A source
BSH3-0	Shift value of B source
USEA	Mode control bit to use Surce A
USEB	Mode control bit to use Source B
USEC	Mode control bit to use Source C
USED	Mode control bit to use Destination D
LF7-0	Logic function minterm select lines
EFE	Exclusive fill enable
IFE	Inclusive fill enable
FCI	Fill carry input
DESC	Descending (decreasing address) control bit
LINE	Line mode control bit (set to 0)

LINE DRAW:

LINE MODE (line draw)

BIT#	BLTCON0	BLTCON1
15	START3	0
14	START2	0
13	START1	0
12	START0	0
11	1	0
10	0	0
09	1	0
08	1	0
07	LF7	0
06	LF6	SIGN
05	LF5	OVF
04	LF4	SUD
03	LF3	SUL
02	LF2	AUL
01	LF1	SING
00	LF0	LINE(=1)

START3-0 Starting point of line (0 thru 15 hex)

LF7-0 Logic function minterm select lines should be preloaded with 4A in order to select the equation $D = (\bar{A}C + AB\bar{C})$. Since A contains a single bit true (8000), most bits will pass the C field unchanged (not A and C), but one bit will invert the C Field and combine it with texture (A and B and not C). The A bit is automatically moved across the word by the hardware.

LINE Line mode control bit (set to 1)

SIGN Sign flag

OVF Word overflow flag

SING Single bit per horiz. line for use with subsequent Area Fill

SUD Sometimes Up or Down (=AUD*)

SUL Sometimes Up or Left

AUL Always Up or Left

The 3 bits above select the Octant for line draw:

OCT	SUD	SUL	AUL
0	1	1	0
1	0	0	1
2	0	1	1
3	1	1	1
4	1	0	1
5	0	1	0
6	0	0	0
7	0	0	0

Blitter start and size (Window, width height)

This register contains the width and height of the blitter operation (in line mode width must = 2, height = line length). Writing to this register starts the Blitter, and should be done last, after all pointers and control registers have been initialized.

BIT# 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00
h9 h8 h7 h6 h5 h4 h3 h2 h1 h0, w5 w4 w3 w2 w1 w0

h = Height = Vertical lines (10 bits = 1024 lines max)
w = Width = Horiz. pixels (6 bits = 64 words = 1024 pixels max)

LINE DRAW: BLTSIZE controls the line length and starts the line draw when written to. The h field controls the line length (10 bits gives lines up to 1024 dots long). The w field must be set to 02 for all line drawing.

BPL x PTH	*Bit plane x pointer (high 3 bits)*
BPL x PTL	*Bit plane x pointer (low 15 bits)*

This pair of registers contains the 18 bit pointer to the address of Bit plane x (x = 1,2,3,4,5,6) DMA data. This pointer must be reinitialized by the processor or Copper to point to the beginning of Bit Plane data every vertical blank time.

BPL1MOD	*Bit plane modulo (odd planes)*
BPL2MOD	*Bit plane modulo (even planes)*

These registers contain the Modulos for the odd and even bit planes. A Modulo is a number that is automatically added to the address at the end of each line, in order that the address then points to the start of the next line. Since they have separate modulos, the odd and even bit planes may have sizes that are different from each other, as well as different from the Display Window size.

BPLCONO	*Bit plan control register (miscellaneous control bits)*

This register controls the operation of the Bit Planes and various aspects of the display.

BIT#	BPLCON0
15	HIRES
14	BPU2
13	BPU1
12	BPU0
11	HOMOD
10	DBLPF
09	COLOR
08	GAUD
07	X
06	X
05	X
04	X
03	LPEN
02	LACE
01	ERSY
00	X

HIRES = High resolution (640) mode
BPU = Bit plane use code 000-110 (NONE through 6 inclusive)
HOMOD = Hold and Modify mode
DBLPF = Double playfield (PF1 = odd, PF2 = even bit planes)
COLOR = Composite video COLOR enable
GAUD = Genlock audio enable (mixed on BKGND pin during vertical blanking)
LPEN = Light pen enable (reset on power up)
LACE = Interlace enable (reset on power up)
ERSY = External Resync (HSYNC, VSYNC pads become inputs; reset on power up)

COPCON *Copper control register*

This is a 1-bit register that when set true, allows the Copper to access the Blitter hardware. This bit is cleared by power on reset, so that the Copper cannot access the Blitter hardware.

BIT#	NAME	FUNCTION
01	CDANG	Copper danger mode. Allows Copper access to Blitter if true.

COPJMP1	Copper restart at first location
COPJMP2	Copper restart at second location

These addresses are strobe addresses; when written to, they cause the Copper to jump indirect using the address contained in the First or Second Location registers described below. The Copper itself can write to these addresses, causing its own jump indirect.

COP1LCH	Copper first location register (high 3 bits)
COP1LCL	Copper first location register (low 15 bits)
COP2LCH	Copper second location register (high 3 bits)
COP2LCL	Copper second location register (low 15 bits)
COP1NS	*Copper instruction fetch identify*

This is a dummy address that is generated by the Copper whenever it is loading instructions into its own instruction register. This actually occurs every Copper cycle except for the second (IR2) cycle of the MOVE instruction. The three types of instructions are shown below:

MOVE	*Move immediate* to dest.
WAIT	*Wait until* beam counter is equal to, or greater than (keeps Copper off of bus until beam position has been reached).
SKIP	*Skip if* beam counter is equal to, or greater than (skips following MOVE inst. unless beam position has been reached).

	MOVE		WAIT UNTIL			SKIP IF		
BIT#	IR1	IR2	IR1	IR2		IR1	IR2	
15	X	RD15	VP7	BFD	*	VP7	BFD	*
14	X	RD14	VP6	VE6		VP6	VE6	
13	X	RD13	VP5	VE5		VP5	VE5	
12	X	RD12	VP4	VE4		VP4	VE4	
11	X	RD11	VP3	VE3		VP3	VE3	
10	X	RD10	VP2	VE2		VP2	VE2	
09	X	RD09	VP1	VE1		VP1	VE1	
08	DA8	RD08	VP0	VE0		VP0	VE0	
07	DA7	RD07	HP8	HE6		HP8	HE6	
06	DA6	RD06	HP7	HE7		HP7	HE7	
05	DA5	RD05	HP6	HE6		HP6	HE6	
04	DA4	RD04	HP5	HE5		HP5	HE5	
03	DA3	RD03	HP4	HE4		HP4	HE4	
02	DA2	RD02	HP3	HE3		HP3	HE3	
01	DA1	RD01	HP2	HE2		HP2	HE2	
00	0	RD00	1	1		1	1	

IR1 = First instruction register
IR2 = Second instruction register
DA = Destination Address for MOVE instruction. Fetched during IR1 time, used during IR2 time on RGA bus.
RD = RAM data moved by MOVE instruction at IR2 time directly from RAM to the address given by the DA field.
VP = Vertical Beam Position comparison bit
HP = Horizontal Beam Position comparison bit
VE = Enable comparison (mask bit)
HE = Enable comparison (mask bit)

*NOTE BFD = Blitter finished disable. When this bit is true, the Blitter Finished flag will have no effect on the Copper. When this bit is zero, the Blitter Finished flag must be true (in addition to the rest of the bit comparisons) before the Copper can exit from its wait state, or skip over an instruction. Note that the V7 comparison cannot be masked.

The Copper is basically a 2-cycle machine that requests the bus only during odd memory cycles (4 memory cycles per in). This prevents collisions with Display, Audio, Disk, Refresh, and Sprites, all of which use only even cycles. It therefore needs (and has) priority over only the Blitter and Micro.

There are only three types of instructions: MOVE immediate, WAIT until, and SKIP if. All instructions (except for WAIT) require 2 bus cycles (and two instruction words). Since only the odd bus cycles are requested, 4 memory cycle times are required per instruction (memory cycles are 280 ns).

There are two indirect jump registers, COP1LC and COP2LC. These are 18-bit pointer registers whose contents are used to modify the program counter for initialization or jumps. They are transferred to the program counter whenever strobe addresses COPJMP1 or COPJMP2 are written. In addition, COP1LC is automatically used at the beginning of each vertical blank time.

It is important that one of the jump registers be initialized and its jump strobe address hit, after power up but before Copper DMA is initialized. This insures a determined startup address and state.

DIWSTRT	*Display window start (upper left vertical-horizontal position)*
DIWSTOP	*Display window stop (lower right vertical-horizontal position)*

These registers control the Display Window size and position, by locating the upper left and lower right corners.

BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
USE v7 v6 v5 v4 v3 v2 v1 v0 h7 h6 h5 h4 h3 h2 h1 h0

DIWSTRT is vertically restricted to the upper ⅔ of the display (v8 = 0), and horizontally restricted to the left ¾ of the display (h8 = 0).
DIWSTOP is vertically restricted to the lower ½ of the display (v8 = / = v7), and horizontally restricted to the right ¼ of the display (h8 = 1).

DDFSTRT	*Display data fetch start (horiz.position)*
DDFSTOP	*Display data fetch stop (horiz.position)*

These registers control the horizontal timing of the beginning and end of the Bit Plane DMA display data fetch. The vertical Bit Plane DMA timing is identical to the Display windows described above. The Bit Plane Modulos are dependent on the Bit Plane horizontal size, and on this data fetch window size.

Register bit assignment

BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
USE X X X X X X X X HB H7 H6 H5 H4 H3 X X

(X bits should always be driven with 0 to maintain upward compatibility)

The tables below show the start and stop timing for different register contents.

DDFSTRT (Left edge of display data fetch)

PURPOSE	H8,	H7,	H6,	H5,	H4
Extra wide (max) *	0	0	1	0	1
wide	0	0	1	1	0
normal	0	0	1	1	1
narrow	0	1	0	0	0

DDFSTOP (Right edge of display data fetch)

PURPOSE	H8,	H7,	H6,	H5,	H4,
narrow	1	1	0	0	1
normal	1	1	0	1	0
wide (max)	1	1	0	1	1

DMACON *DMA control write (clear or set)*
DMACONR *DMA control (and Blitter status) read*

This register controls all of the DMA channels, and contains Blitter DMA status bits.

BIT#	FUNCTION	DESCRIPTION
15	SET/CLR	Set/Clear control bit. Determines if bits written with a 1 get set or cleared.
14	BBUSY	Blitter busy status bit (read only)
13	BZERO	Blitter logic zero status bit (read only)
12	X	
11	X	
10	BLTPRI	Blitter DMA priority (over CPU micro) —also called "Blitter Nasty" —disables /BLS pin, preventing micro from stealing any bus cycles while blitter DMA is running.

BIT#	FUNCTION	DESCRIPTION
09	DMAEN	Enable all DMA below.
08	DPLEN	Bit Plane DMA enable.
07	COPEN	Copper DMA enable.
06	BLTEN	Blitter DMA enable.
05	SPREN	Sprite DMA enable.
04	DSKEN	Disk DMA enable.
03	AUD3EN	Audio channel 3 DMA enable.
02	AUD2EN	Audio channel 2 DMA enable.
01	AUD1EN	Audio channel 1 DMA enable.
00	AUDOEN	Audio channel 0 DMA enable.

DSKPTH *Disk pointer (high 3 bits)*
DSKPTL *Disk pointer (low 15 bits)*

This pair of registers contains the 18-bit address of Disk DMA data. These address registers must be initialized by the processor or Copper before disk DMA is enabled.

REFPTR *Refresh pointer*

This register is used as a Dynamic RAM refresh address generator. It is writeable for test purposes only, and should never be written by the microprocessor.

SPRxPTH *Sprite x pointer (high 3 bits)*
SPRxPTL *Sprite x pointer (low 15 bits)*

This pair of registers contains the 18-bit address of Sprite x (x = 0,1,2,3,4,5,6,7) DMA data. These address registers must be initialized by the processor or Copper every vertical blank time.

SPRxPOS *Sprite x vertical-horizontal position data*
SPRxCTL *Sprite x vertical-horizontal*

These 2 registers work together as position, size and feature Sprite control registers. They are usually loaded by the Sprite DMA channel, during horizontal blank; however, they may be loaded by either processor any time.

SPRxPOS register:

BIT#	SYM	FUNCTION
15-08	SV7-SV0	Start vertical value. High bit (SV8) is in SPRxCTL reg. below.
07-00	SH8-SH1	Start horizontal value. Low bit (SH0) is in SPRxCTL reg. below.

SPRxCTL register (writing this address disables sprite horizontal comparator circuit):

BIT#	SYM	FUNCTION
15-08	EV7-EV0	End (stop) vert.value.low 8 bits
07	ATT	Sprite attach control bit (odd sprites)
06-04	X	Not used
02	SV8	Start vert. value high bit
01	EV8	End (stop) vert. value high bit
00	SH0	Start horiz. value low bit

VPOSR *Read vertical most significant bit (and frame flop)*

VPOSW *Write vertical most significant bit (and frame flop)*

BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
USE LOF_____V8
LOF = Long frame (auto toggle control bit in BPLCONO)

VHPOSR *Read vertical and horizontal position of beam or lightpen*

VHPOSW *Write vertical and horizontal position of beam or lightpen*

BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
USE V7 V6 V5 V4 V3 V1 V0,H8 H7 H6 H5 H4 H3 H1
RESOLUTION = 1/160 OF SCREEN WIDTH (280 NS)

AGNUS NOTES

1) The Agnus pointer registers are updated via a pipelining scheme that requires that a register not be accessed on two contiguous cycles.

   This precludes the use of "single operand" blitter functions that might seem possible based on the register descriptions.

   Caution is also required to prevent processor access to registers that may be subject to concurrent DMA access.

DMA Time Slot Allocation/Horizontal Line

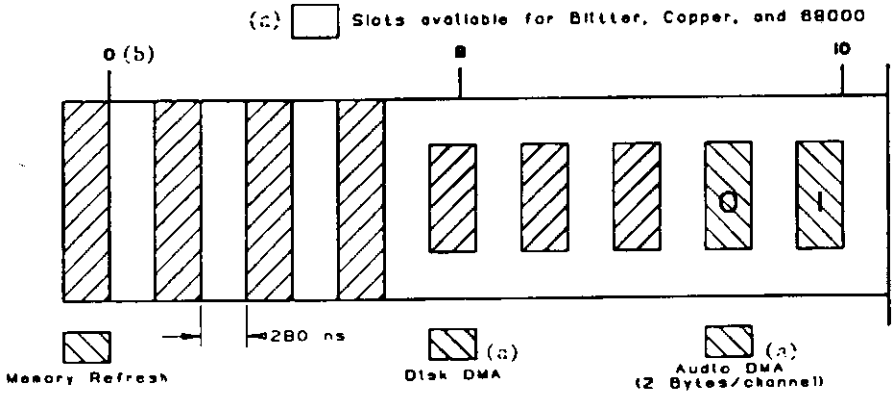
NOTES

1) These operations only take slots if the associated operation is being performed

   Note: Copper Data Move instructions require 4 slots.
   Copper Wait instructions require 6 slots.

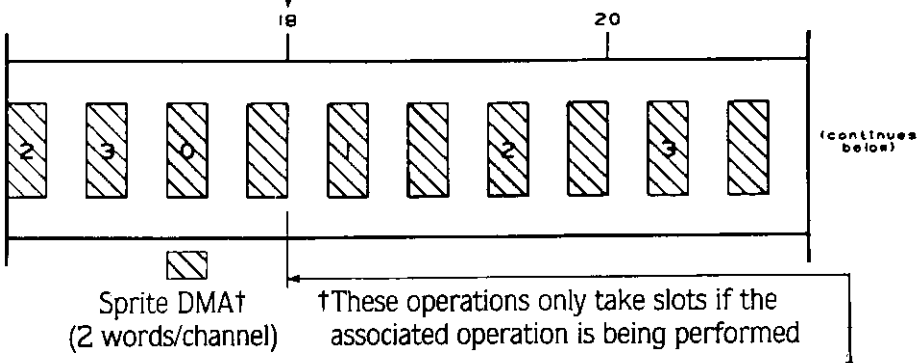
2) This cycle 0 appears to exclude one of the memory refresh cycles. This is not the case.

   Actual system hardware demands certain specific values for data fetch start and display start. Therefore this timing chart has been "adjusted" to match those requirements.

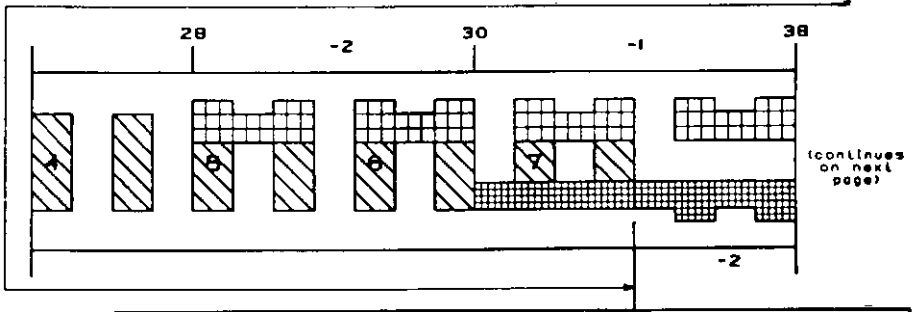


DMA Time Slot Allocation / Horizontal Line (Cont'd)

Hardware stop installed here. Data fetch cannot begin any sooner than cycle 18. This allows the user to wipe out most of the sprites if desired (by defining an extra-wide display) but leaves the audio and disk DMA untouched.



†These operations only take slots if the associated operation is being performed

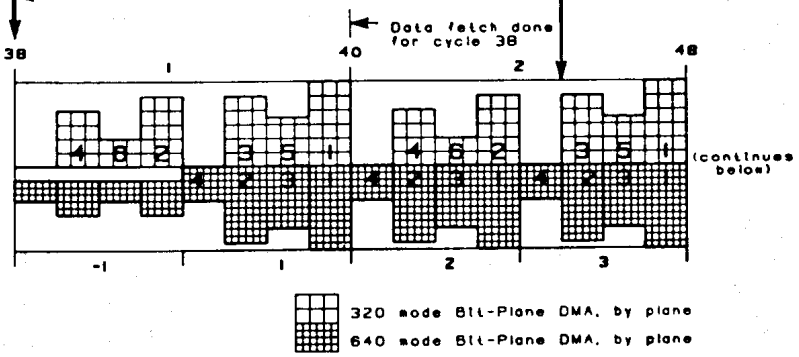


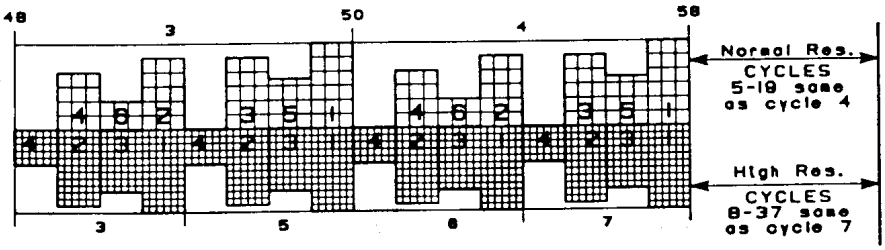
Some sprites are unusable if the display starts early due to an extra word(s) associated with a wide display and/or horizontal scrolling. In this case, the bit-plane DMA steals the cycles normally allocated to the sprites, as illustrated above.

DMA Time Slot Allocation/Horizontal Line (Cont'd)

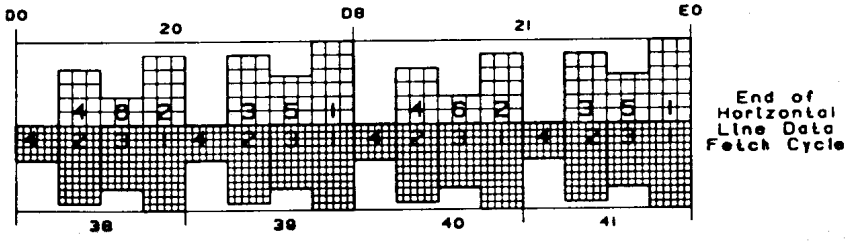
Data fetch start can only be specified at even multiples of 8 clocks. This is the clock position which should be specified for the normal width display. (20 word fetch for 320 pixel, 40 word fetch for 640 pixel width).

Five clocks must occur before the data which was fetched for a particular position can appear onscreen. For example, if data fetch start is specified as 38, it will not be available for display until clock number 45.





A hardware data-fetch stop has been installed at count D8 so as to prevent the bit-plane data-fetch from overrunning the time allotted for the memory refresh or disk DMA.



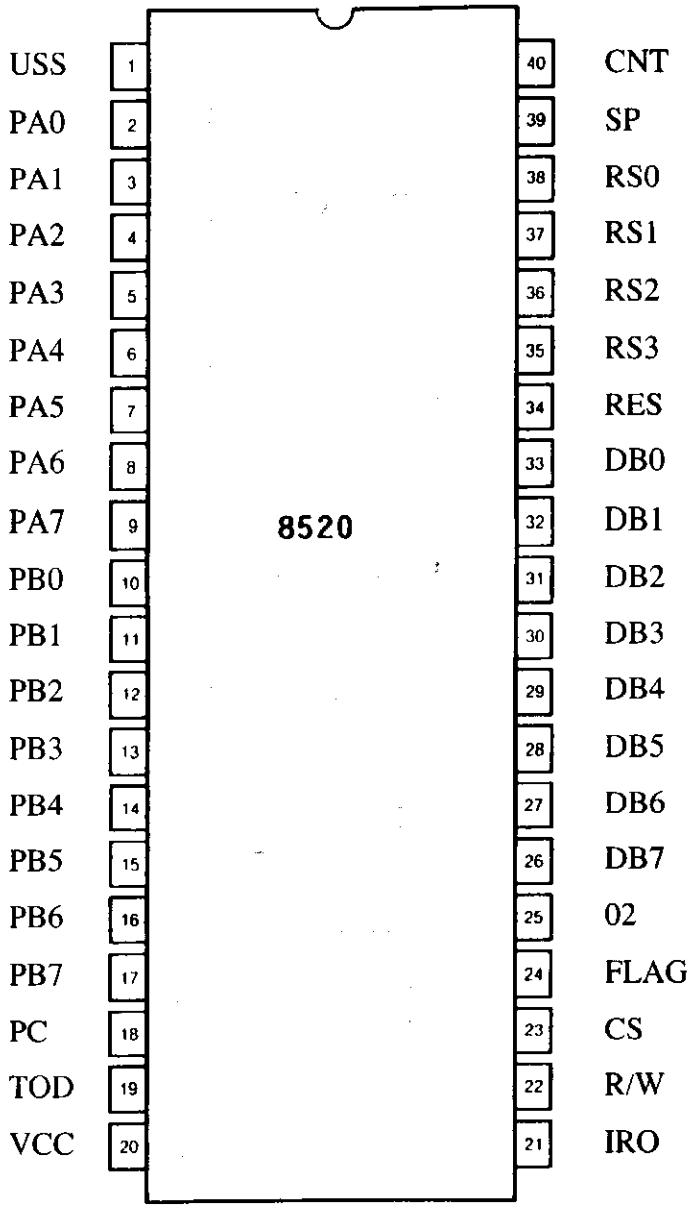


Figure 6.5. 8520 Pin Configuration

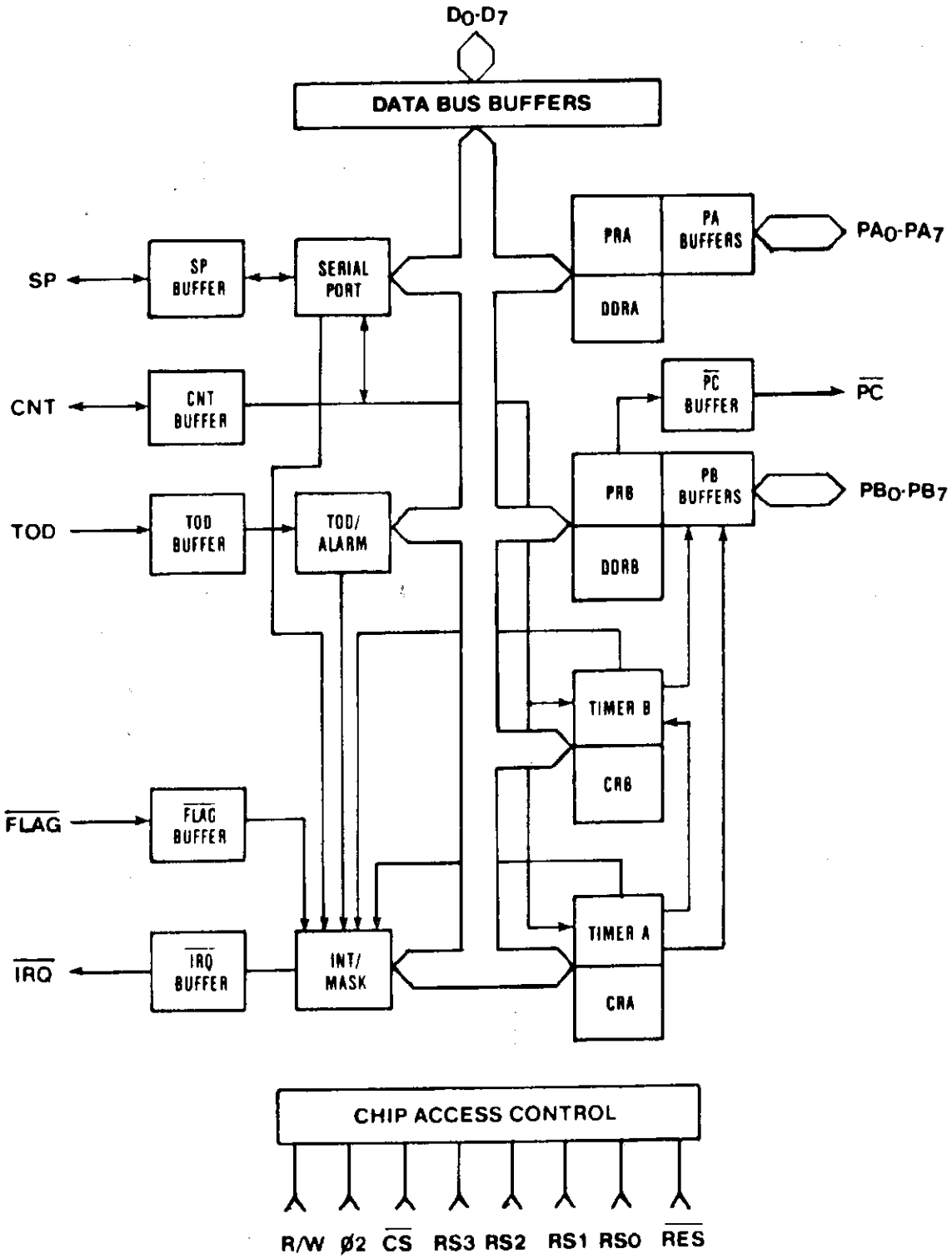
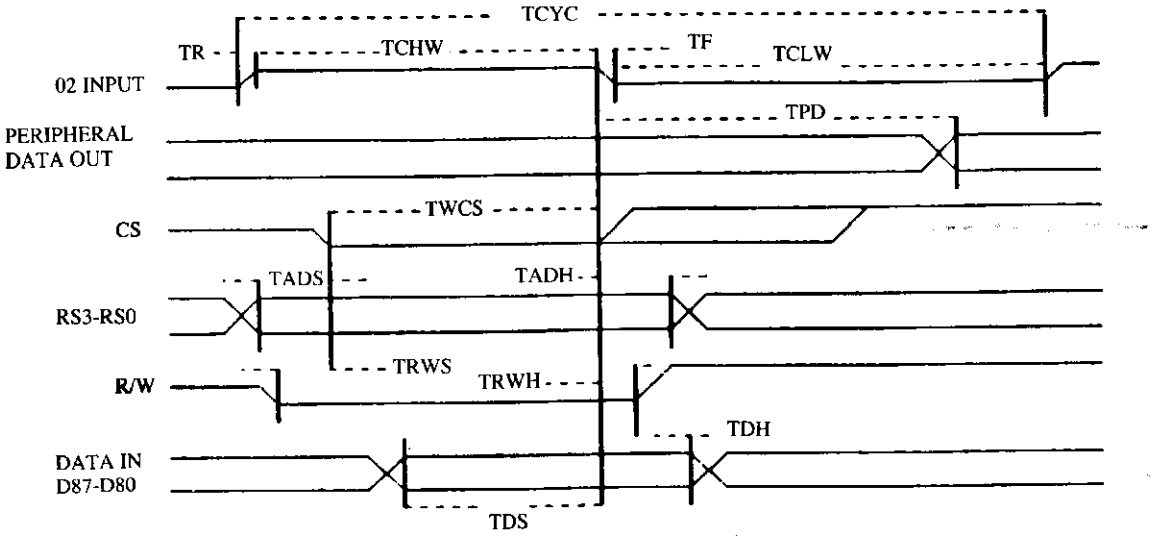


Figure 6.6. 8520 Block Diagram

WRITE TIMING DIAGRAM



READ TIMING DIAGRAM

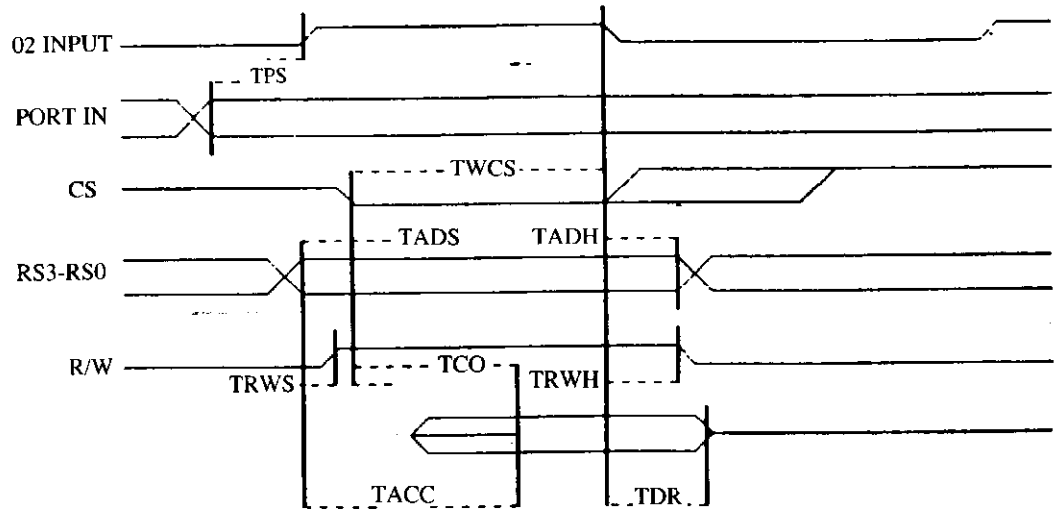


Figure 6.7. 8520 Timing Diagrams

INTERFACE SIGNALS

02-Clock Input

The 02 clock is a TTL compatible input used for internal device operation and as a timing reference for communicating with the system data bus.

CS-Chip Select Input

The CS input controls the activity of the 8520. A low level on CS while 02 is high causes the device to respond to signals on the R/W and address (RS) lines. A high on CS prevents these lines from controlling the 8520. The CS line is normally activated (low) at 02 by the appropriate address combination.

R/W-Read/Write Input

The R/W signal is normally supplied by the microprocessor and controls the direction of data transfers of the 8520. A high on R/W indicates a read (data transfer out of the 8520), while a low indicates a write (data transfer into the 8520).

RS3-RS0 — Address Inputs

The address inputs select the internal registers as described by the Register Map.

DB7-DB0 — Data Bus Inputs/Outputs

The eight bit data bus transfers information between the 8520 and the system data bus. These pins are high impedance inputs unless CS is low and R/W and 02 are high, to read the device. During this read, the data bus output buffers are enabled, driving the data from the selected register onto the system data bus.

IRQ-Interrupt Request Output

IRQ is an open drain output normally connected to the processor interrupt input. An external pullup resistor holds the signal high, allowing multiple IRQ-outputs to be connected together. The IRQ output is normally off (high impedance) and is activated low as indicated in the functional description.

RES-Reset Input

A low on the RES pin resets all internal registers. The port pins are set as inputs and port registers to zero (although a read of the ports will return all highs because of passive pullups). The timer control registers are set to zero and the timer latches to all ones. All other registers are reset to zero.

REGISTER MAP

RS3	RS2	RS1	RS0	REG		
0	0	0	0	0	PRA	Peripheral Data Reg. A
0	0	0	1	1	PRB	Peripheral Data Reg. B
0	0	1	0	2	DDRA	Data Direction Reg. A
0	0	1	1	3	DDRB	Data Direction Reg. B
0	1	0	0	4	TA LO	Timer A Low Register
0	1	0	1	5	TA HI	Timer A High Register
0	1	1	0	6	TB LO	Timer B Low Register
0	1	1	1	7	TB HI	Timer B High Register
1	0	0	0	8		Event LSB
1	0	0	1	9		Event 8-15
1	0	1	0	A		Event MSB
1	0	1	1	B		No Connect
1	1	0	0	C	SDR	Serial Data Register
1	1	0	1	D	ICR	Interrupt Control Register
1	1	1	0	E	CRA	Control Register A
1	1	1	1	F	CRB	Control Register B

FUNCTIONAL DESCRIPTION

I/O Ports (PRA, PRB, DDRA, DDRB)

Ports A and B each consist of an 8-bit Peripheral Data Register (PR) and an 8-bit Data Direction Register (DDR). If a bit in the DDR is set to the corresponding bit in the PR is an output if a DDR bit is set to zero, the corresponding PR bit is defined as an input. On a READ, the PR reflects the information present on the actual port pins (PA0-PA7, PB0-PB7) for both input and output bits. Port A has both passive and active pullup devices, providing both CMOS and TTL compatibility. It can drive 2 TTL loads. Port B has only passive pullup devices and has a much higher current-sinking capability.

Handshaking

Handshaking on data transfers can be accomplished using the PC output pin and the FLAG input pin. PC will go low on the 3rd cycle after a PORT B access. This signal can be used to indicate "data ready" at PORT B or "data accepted" from PORT B. Handshaking on a 16-bit data transfers (using both PORT A and PORT B) is possible by always reading or writing PORT A first. FLAG is a negative edge sensitive input which can be used for receiving the PC output from another 8520 or as a general purpose interrupt input. Any negative transition on FLAG will set the FLAG interrupt bit.

Reg	Name	D7	D6	D5	D4	D3	D2	D1	D0
0	PRA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
1	PPB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
2	DDRA	DPA7	DPA6	DPA5	DPA4	DPA3	DPA2	DPA1	DPA0
3	DDRB	DPB7	DPB6	DPB5	DPB4	DPB3	DPB2	DPB1	DPB0

Interval Timers (Timer A, Timer B)

Each interval timer consists of a 16-bit read-only Timer Counter and a 16-bit write-only Timer Latch. Data written to the timer are latched in the Timer Latch, while data read from the timer are the present contents of the Timer Counter. The timers can be used independently or linked for extended operations. The various timer modes allow generation of long time delays, variable width pulses, pulse trains and variable frequency waveforms. Utilizing the CNT input, the timers can count external pulses or measure frequency, pulse width and delay times of external signals. Each timer has an associated control register, providing independent control of the following functions:

Start/Stop

A control bit allows the timer to be started or stopped by the microprocessor at any time.

PB On/Off

A control bit allows the timer output to appear on a PORT B output line (PB6 for TIMER A and PB7 for TIMER B). This function overrides the DDRB control bit and forces the appropriate PB line to an output.

Toggle/Pulse

A control bit selects the output applied to PORT B. On every timer underflow the output can either toggle or generate a single positive pulse of one cycle duration. The toggle output is set high whenever the timer is started and is set low by RES.

One-Shot/Continuous

A control bit selects either timer mode. In one-shot mode, the timer will count down from the latched value to zero, generate an interrupt, reload the latched value, then stop. In continuous mode, the timer will count from the latched value to zero, generate an interrupt, reload the latched value and repeat the procedure continuously. In one-shot mode; a write to Timer High (registers 5 for TIMER A, 7 for TIMER B) will transfer the timer latch to the counter and initiate counting regardless of the start bit.

Force Load

A strobe bit allows the timer latch to be loaded into the timer counter at any time, whether the timer is running or not.

Input Mode

Control bits allow selection of the clock used to decrement the timer. TIMER A can count 02 pulses or external pulses applied to the CNT pin. TIMER B can count 02 pulses, external CNT pulses, TIMER A underflow pulses or TIMER A underflow pulses while the CNT pin is held high.

The timer latch is loaded into the timer on any timer underflow, on a force load or following a write to the high byte of the prescaler while the timer is stopped. If the timer is running, a write to the high byte will load the timer latch, but not reload the counter.

READ (TIMER)

REG	Name								
4	TA LO	TAL7	TAL6	TAL5	TAL4	TAL3	TAL2	TAL1	TAL0
5	TA HI	TAH7	TAH6	TAH5	TAH4	TAH3	TAH2	TAH1	TAH0
6	TB LO	TBL7	TBL6	TBL5	TBL4	TBL3	TBL2	TBL1	TBL0
7	TB HI	TBH7	TBH6	TBH5	TBH4	TBH3	TBH2	TBH1	TBH0

WRITE (PRESCALER)

REG	Name								
4	TA LO	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
5	TA HI	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0
6	TB LO	PBL7	PBL6	PBL5	PBL4	PBL3	PBL2	PBL1	PBL0
7	TB HI	PBH7	PBH6	PBH5	PBH4	PBH3	PBH2	PBH1	PBH0

TOD

TOD consists of a 24 bit binary counter. Positive edge transitions on this pin cause the binary to increment. The TOD pin has a passive pull-up on it. A programmable ALARM is provided for generating an interrupt at a desired time. The ALARM registers are located at the same addresses as the corresponding TOD register. Access to the ALARM is governed by a Control Register bit. The ALARM is write-only; any read of a TOD address will read time regardless of the state of the ALARM access bit.

A specific sequence of events must be followed for proper setting and reading of TOD. TOD is automatically stopped whenever a write to the register occurs. The clock will not start again until after a write to the LSB Event Register. This assures TOD will always start at the desired time. Since a carry from one stage to the next can occur at any time with respect to a read operation, a latching function is included to keep all Time of Day information constant during a

read sequence. All TOD registers latch on a read of MSB event and remain latched until after a read of LSB Event. The TOD clock continues to count when the output registers are latched. If only one register is to be read, there is no carry problem and the register can be read "on the fly", provided that any read of MSB Event is followed by a read of LSB Event to disable the latching.

READ

REG	NAME								
8	LSB EVENT	E7	E6	E5	E4	E3	E2	E1	E0
9	EVENT 8-15	E15	E14	E13	E12	E11	E10	E9	E8
A	MSB EVENT	E23	E22	E21	E20	E19	E18	E17	E16

WRITE

CRB7 = 0
CRB7 = 1 ALARM
(SAME FORMAT AS READ)

Serial Port (SDR)

The serial port is a buffered, 8-bit synchronous shift register system. A control bit selects input or output mode. In input mode, data on the SP pin is shifted into the shift register on the rising edge of the signal applied to the CNT pin. After 8 CNT pulses, the data in the shift register is dumped into the Serial Data Register and an interrupt is generated. In the output mode, TIMER A is used for the baud rate generator. Data is shifted out on the SP pin at ½ the underflow rate of TIMER A. The maximum baud rate possible is 02 divided by 6, but the maximum useable baud rate will be determined by line loading and the speed at which the receiver responds to input data. Transmission will start following a write to the Serial Data Register (provided TIMER A is running and in continuous mode). The clock signal derived from TIMER A appears as an output on the CNT pin. The data in the Serial Data Register will be loaded into the shift register then shift out to the SP pin when a CNT pulse occurs. Data shifted out becomes valid on the falling edge of CNT and remains valid until the next falling edge. After 8 CNT pulses, an interrupt is generated to indicate more data can be sent. If the Serial Data Register was loaded with new information prior to this interrupt, the new data will automatically be loaded into the shift register and transmission will be continuous. If no further data is to be transmitted, after the 8th CNT pulse, CNT will return high and SP will remain at the level of the last data bit transmitted. SDR data is shifted out MSB first and serial input data should also appear in this format.

The bidirectional capability of the Serial Port and CNT clock allows several devices to be connected to a common serial communication bus on which one acts as a master, sourcing data and shift clock, while all other chips act as slaves. Both CNT and SP outputs are open

drain, with passive pullups, to allow such a common bus. Protocol for slave/master selection can be transmitted over the serial bus, or via dedicated handshaking lines.

REG	NAME								
C	SDR	S7	S6	S5	S4	S3	S2	S1	S0

Interrupt Control (ICR)

There are five sources of interrupts on the 8520: underflow from TIMER A, underflow from TIMER B, TOD ALARM, Serial Port full/empty and FLAG. A single register provides masking and interrupt information. The Interrupt Control Register consists of a write-only MASK register and a read-only DATA register. Any interrupt which is enabled by the MASK register will set the IR bit (MSB) of the DATA register and bring the IRQ pin low. In a multi-chip system, the IR bit can be polled to detect which chip has generated an interrupt request.

The interrupt DATA register is cleared and the IRQ line returns high following a read of the DATA register. Since each interrupt sets an interrupt bit regardless of the MASK, and each interrupt bit can be selectively masked to prevent the generation of a processor interrupt, it is possible to intermix polled interrupts with true interrupts. However, polling the IR bit will cause the DATA register to clear, therefore, it is up to the user to preserve the information contained in the DATA register if any polled interrupts were present.

The MASK register provides convenient control of individual mask bits. When writing to the MASK register, if bit 7 (SET/CLEAR) of the data written is a ZERO, any mask bit written with a one will be cleared, while those mask bits written with a zero will be unaffected. If bit 7 of the data written is a ONE, any mask bit written with a one will be set, while those mask bits written with a zero will be unaffected. In order for an interrupt flag to set IR and generate an Interrupt Request, corresponding MASK bit must be set.

READ (INT DATA)

REG	NAME								
D	IRA	IR	0	0	FLG	SP	ALRM	TB	TA

WRITE (INT MASK)

REG	NAME								
D	IRC	S/C	X	X	FLG	SP	ALRM	TB	TA

Control Registers

There are two control registers in the 8520: CRA and CRB. CRA is associated with TIMER A and CRB is associated with TIMER B.

The register format is as follows:

CRA:

BIT	NAME	FUNCTION
0	START	1 = START TIMER A. 0 = STOP TIMER A. This bit is automatically reset when underflow occurs during one-shot mode.
1	PBON	1 = TIMER A output appears on PB6, 0 = PB6 normal operation
2	OUTMODE	1 = TOGGLE, 0 = PULSE
3	RUNMODE	1 = ONE-SHOT, 0 = CONTINUOUS
4	LOAD	1 = FORCE LOAD (this is a STROBE input, there is no data storage, bit 4 will always read back a zero and writing a zero has no effect.
5	INMODE	1 = TIMER A counts positive CNT transitions, 0 = TIMER A counts 02 pulses.
6	SPMODE	1 = SERIAL PORT output (CNT sources shift clock). 0 = SERIAL PORT input (external shift clock required).
7	TODIN	1 = 50 Hz clock required on TOD pin for accurate time. 0 = 60 Hz clock required on TOD pin for accurate time.

CRB:

BIT	NAME	FUNCTION
		(Bits CRB0-CRB4 are identical to CRA0-CRA4 for TIMER B with the exception that bit 1 controls the output of TIMER B on PB7).
5,6	INMODE	Bits CRB5 and CRB6 select one of four input modes for TIMER B as:

CRB6	CRB5	
0	0	TIMER B counts 02 pulses
0	1	TIMER B counts positive CNT transitions
1	0	TIMER B counts TIMER A underflow pulses
1	1	TIMER B counts TIMER A underflow pulses while CNT is high

7	ALARM	1 = writing to TOD registers set ALARM. 0 = writing to TOD registers sets TOD clock.