MODIFICATION SUMMARY
8374R0 versus 8374R1

The modifications listed below have been made to the 8374R0 database to create the 8374R1 database. Modifications 1,2,3 affect hardware only.

1. MULTIPLEXED ADDRESS TIMING PROBLEMS(ADDGEN, SHEET 16)

   a) Reworked/resized multiplexer select logic to increase setup time of MA lines versus CAS falling edge.

   b) Added two inverters in CAS logic path to match double CAS timing with single CAS timing.

2. STATIC COLUMN DRAM COMPATIBILITY (SHEET 16)

   a) Modified DRFF signal (forces MA lines high at end of RAS/CAS cycle) to allow MA lines to remain stable for an additional 70ns. This should allow ALICE to be compatible with both STATIC COLUMN DRAM's and PAGE-MODE DRAMS.

↳ BW INCREASE W/SC DRAM's

3. SCLK SYNCHRONIZATION---Added full dynamic flip-flop to SCLK path to allow resynchronization of LISA's SCLK signal. LISA has been modifed to move SCLK one 14MHz clock earlier. (HRCLOCKGEN)

4. FMODE moved from RGA address 076 to 1FC (RGADEC)

5. BITPLANE FETCH ENGINE---HIRES(2X,4X), SHRES(2X,4X) MODES REPAIRED (SHEET 8)

   a) Modified 8374R0 logic slightly to eliminated GAPS in bitplane fetch cycle. These gaps were in HIRES and SHRES displays in 2X and 4X modes.

   b) The BPRUN signal will be modifed to eliminate a problem with the 4X LORES mode where the BPRUN signal runs into the next line and kills SPRITE0.

6. SPRITE SCAN DOUBLING (SHEET 7---CREATED NEW CELLS-STCMSD,SDBL)

   a) Modified sprite logic to implement scan doubling. Added bit in FMODE logic to enable scan doubling

      Modified sprite compare logic to start scan doubling as soon as sprite starts and double odd or even lines depending upon sprite start position

      Added scan double cell which allows individual scan double control of each sprite

7. BITPLANE SCAN DOUBLING (SHEET 7, ADDGEN)

   a) Added bit in FMODE to enable bitplane scan doubling

   b) Modified modulu logic to use negative(or alternate) modulo during scan doubled lines.

PRELIMINARY

Integrated Circuit Specification

for the

ALICE (8374)

AGNUS REPLACEMENT CHIP FOR AA

Commodore P/N XXXXXXXX

Date: March 13, 1991

Engineer: Bill J. Thomas

Chingtao Shen

1.0 DESCRIPTION

This specification describes the requirements for an N-channel HMOS DMA Controller. The IC device described herein shall produce, in a 680x0 microprocessor environment DMA addresses using a RAM Address Generator and a Register Address Encoder. This device shall contain 27 DMA channel controllers that include the Blitter, Bitplane, Copper, Audio, Sprites, Disk and Memory Refresh.

The IC shall generate 7 MHz and 3.5 MHz system clocks, dynamic RAMinterface to address up two megabytes of memory, and NTSC/PAL video synchronization pulses. These pulses are also completely programmable to interface to higher resolution monitors. The timings are based on a clock input of 14.31818 MHz for NTSC mode and 14.188 MHz for PAL mode.

1.1 CONFIGURATION

This IC device shall be configured in a standard 84-pin plastic chip carrier package. *(See Fig. 1)*

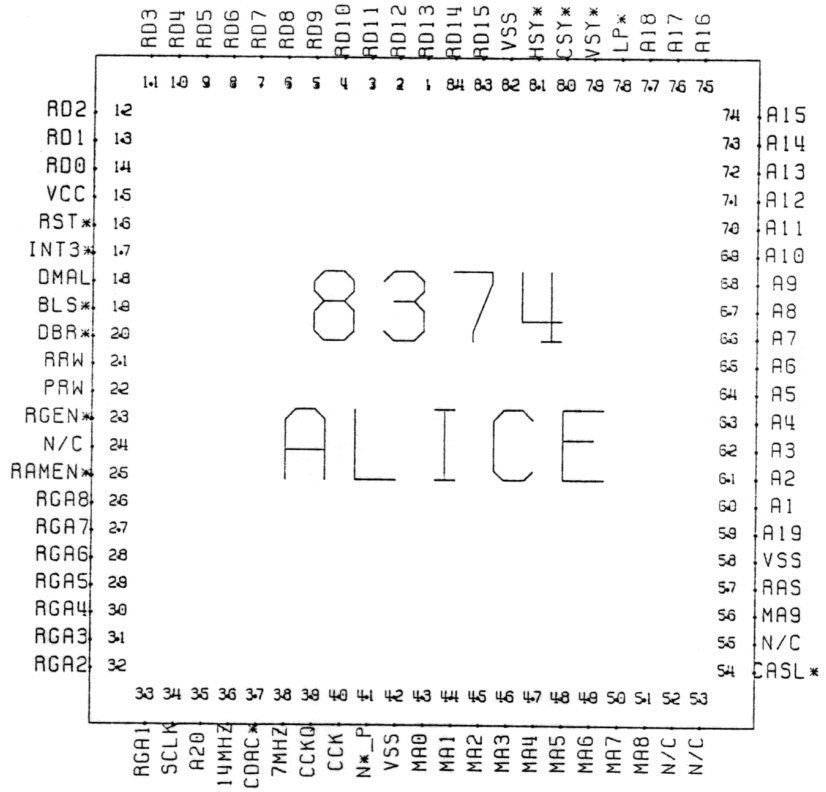

Fig. 1 Pin-out Diagram

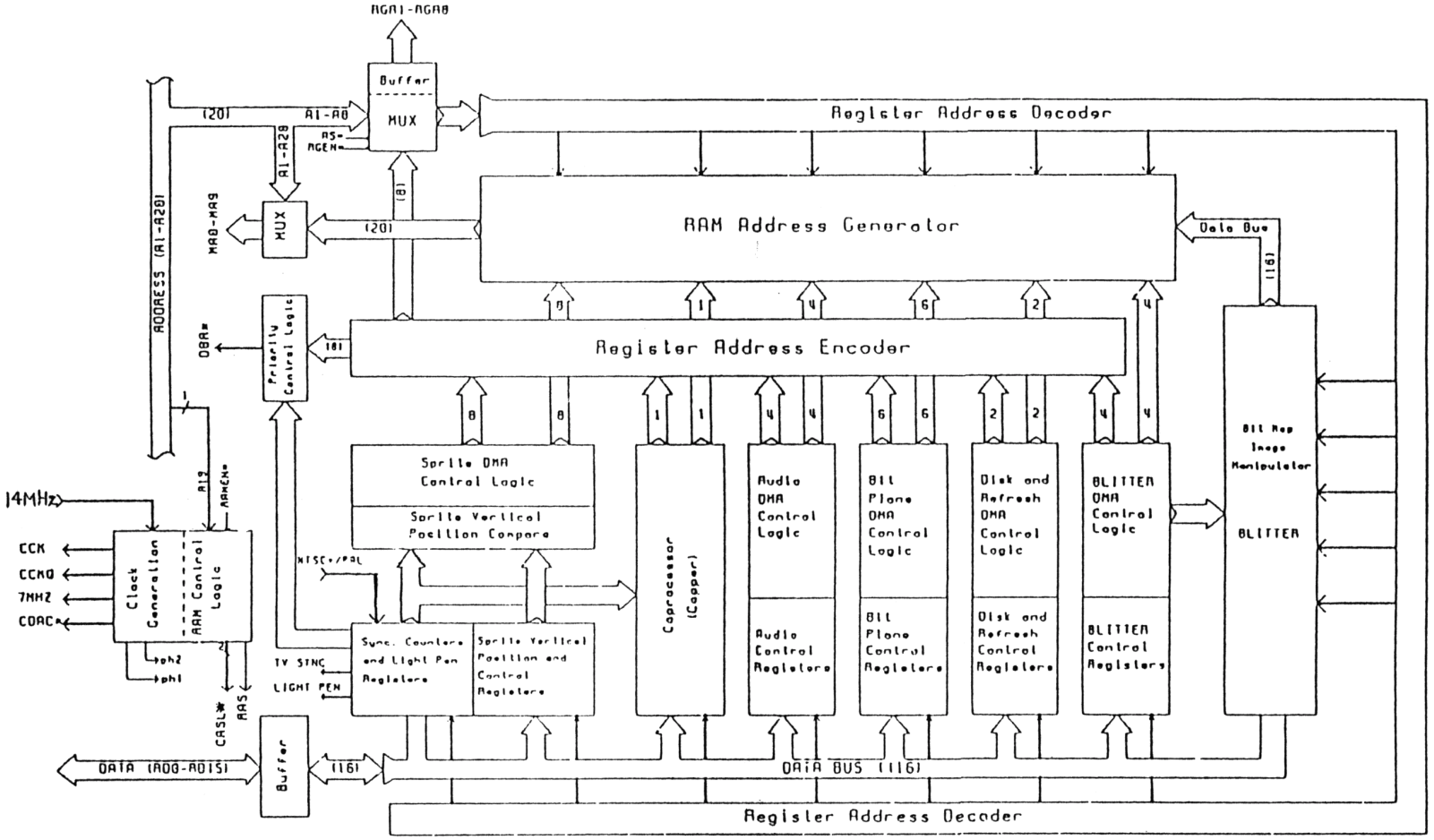

Fig. 2 BLOCK DIAGRAM

1.2 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
A20	35	IN	to select the internal registers and put an address on the RGA lines to select registers outside the device. The processor uses A1 to A20 to generate multiplexed DRAM addresses on the MA outputs.
RD15-RD0	1 thru 14, 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
RGEN*	23	IN	Active low. 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. The processor is doing a DRAM access. The processor supplies an address on the A1 to A20 inputs and the device multiplexes this address onto the MA outputs. The RAS line is always asserted on a DRAM access.
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.
MA0-MA8	43 thru 51	OUT	Output bus. This 10 bit output bus provides
MA9	56	OUT	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 can address up to 2 Megabytes of RAM. 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).

CASL*	54	OUT	Active low. This output strobes the column address into the DRAMS
RAS	57	OUT	Active high. This output is used to strobe the row address into the DRAMs. This signal will be asserted only if the processor is doing a DRAM access and A19 is high, or if the IC is performing a DMA cycle (DRB* is low).
DBR*	20	OUT	Active low. The device asserts this signal when a DMA cycle is under way. RAS is always asserted during a DMA cycle On a memeory refresh cycle DBR* and RAS are asserted; CASL* is deasserted. During a DMA cycle the device will assert CASL*.
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	I/O	This line is bidirectional and buffered. This signal is the horizontal synchronization pulse and is NTSC/PAL compatible. When set as an input, an external video source drives this signal to synchronize the horizontal beam counter. See register description for programming modes.
VSY*	79	I/O	This line is bidirectional and buffered. This signal is the vertical synchronization pulse and is NTSC/PAL compatible. When set as an input, an external video source drives this signal to synchronize the vertical beam counter. See register descriptions for programming modes.
CSY*	80	OUT	This signal is the composite video synchronization pulse and is NTSC/PAL compatible. See register descriptions for programming modes.
LP*	78	OUT	Active low. This input is used to indicate when the light pen is coincident with the monitor beam.
RST*	16	IN	Active low. This input will initialize 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 will suspend its blitter operation and allows the processor to have control of the cycle.
SCLK	34	IN	This is the synchronization clock from LISA along with the 14 MHz clock.
14MHz	36	IN	This is the input clock from LISA that provides the master time base for the device.
CCK	40	OUT	This signal is a clock, which is obtained after dividing the 14 MHz line by four. It is also known as the color clock frequency for NTSC/PAL applications.
CCKQ	39	OUT	This clock is the CCK clock shifted by 90 degrees.
7MHZ	38	OUT	This clock is obtained after dividing the 14 MHZ Clock line by two.
CDAC*	37	OUT	This clock is obtained after inverting the 7 MHz clock and shifting it by 90 degrees.
N*_P	41	IN	This signal is used to determine which video standard is to be activated at power-up. During reset, if N*_P is low, the device powers-up in NTSC mode; if N*_P is high, it powers-up in PAL mode. This line is connected to a passive pull up. Therefore, if this pin is disconnected, the device will power-up in PAL mode. PAL or NTSC modes may be reselected after power-up by writing the desired value in the BEAMCON0 register, bit 5.
VCC	15	P	Power
VSS	42,58,82	P	Ground
N/C	24,52,53,55		Not connected

2.0 ELECTRICAL PARAMETERS

2.1 ABSOLUTE MAXIMUM RATINGS

Stress above those listed may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied.

CHARACTERISTIC	*MIN*	*MAX*	*UNITS*
2.1.1 Ambient Temperature	-25	+125	deg C
2.1.2 Storage Temperature	-65	+150	deg C
2.1.3 Applied Supply Voltage	-0.5	+7.0	V
2.1.4 Applied Output Voltage	-0.5	+5.5	V
2.1.5 Applied Input Voltage	-2.0	+7.0	V
2.1.6 Power Dissipation	1.5		W

2.2 OPERATING CONDITIONS

All electrical characteristics are specified over the entire range of conditions, unless specifically noted. All voltages are operating conditions, unless specifically noted. All voltages are referenced to Vss = 0.0v.

CONDITION	*MIN*	*MAX*	*UNITS*
2.2.1 Supply Voltage (Vcc)	4.75	5.25	V
2.2.2 Case temperature	0	60	deg C

2.3 D.C. CHARACTERISTICS

CHARACTERISTIC	*SYMBOL*	*MIN*	*MAX*	*UNITS*	*CONDITIONS*
2.3.1 Input High Level	Vih	2.5	Vcc+1	V	
2.3.2 Input Low Level	Vil	-0.5	0.6	V	
2.3.3 Output High Level	Voh	+2.4	-	V	Ioh=300uA
2.3.4 Output Low Level	Vol	-	0.4	V	IOL=4.8mA
2.3.5 Input Leakage	Iin	-10	+10	uA	0.0V^0
2.3.6 Output Leakage	Ilk	-10	20	uA	.4V << Vout <<2.4v
2.3.7 Supply Current	Icc	-	250	mA	Outputs open
2.3.8 Capacitance	Cpin	-	10	pF	

2.4 A.C. CHARACTERISTICS

Refer to Figure 6 through 9 for waveform diagrams.

CLOCK RELATIONS (Refer to Figure 6)

	SYMBOL	MIN	MAX	UNIT	
2.4.1 14MHz clock cycle	t14MC	69.14	70.54	ns	
2.4.2 14MHz clock high	t14MHi	24.0	45.8	ns	
2.4.3 14MHz clock low	t14MLo	24.0	45.8	ns	
2.4.4 CCK clock cycle	tCYC	260	290	ns	
2.4.5 CCK clock high	tCH	130	150	ns	
2.4.6 CCK Clock low	tCL	130	150	ns	
2.4.7 CCK-CCKQ clock separation	tCQ	65	75	ns	
2.4.8 7MHz clock cycle	t7MC	130	150	ns	
2.4.9 7MHz Clock high	t7MHi	65	75	ns	
2.4.10 7MHz clock low	t7MLo	65	75	ns	
2.4.11 7MHz-CDACQ clock separation	t7MQ	30	40	ns	
2.4.12 CCK to 7MHz delay	tC7M	0	15	ns	
2.4.13 CCKQ to 7MHz delay	tQ7M	0	15	ns	
2.4.14 Clock rise time	tR	0	10	ns	
2.4.15 Clock fall time	tF	0	10	ns	
2.4.16 SCLK clock cycle	tSCLK	276.56	282.16	ns	(1)
2.4.17 SCLK clock high	tSCLKHi	48.0	91.6	ns	(2)
2.4.18 SCLK clock low	tSCLKLo	144.0	274.8	ns	(3)
2.4.19 14MHz to SCLK delay	tSYNC	5	25	ns	

PROCESSOR ACCESS (2.4.23.2 Data input setup time-Refer to Figure 7)

	SYMBOL	MIN	MAX	UNIT
2.4.20 Address input setup time	tAddinS	45	-	ns
2.4.21 Address input hold time	tAddinH	30	260	ns
2.4.22 Processor access control setup time	tAccS	10	-	ns
2.4.23 Processor access control hold time	tAccH	0	220	ns
2.4.24 Processor CAS access setup time	tPCS	10	-	ns
2.4.25 Processor CAS access hold time	tPCH	10	270	ns
2.4.26 Data input setup time	tDinS	50	-	ns
2.4.27 Data input hold time	tDinH	0	-	ns
2.4.28 Reset input setup time	tResS	50	-	ns
2.4.29 Reset input hold time	tResH	50	-	ns
2.4.30 Write Pulse Width	tWP	45	-	ns

DEVICE ACCESS (Refer to Figure 8)

		SYMBOL	MIN	MAX	UNIT	
2.4.31	CCK low to DBR valid prop time	tCLDBR	70	120	ns	
2.4.32	RAS pulse width	tRP	72	82	ns	
2.4.33	CCK low to CAS low prop time	tCKCL	-	15	ns	
2.4.34	CCK high to CAS high prop time	tCKCH	-	15	ns	
2.4.35	RAS address setup time	tRASS	0	-	ns	
2.4.36	RAS address hold time	tRASH	15	-	ns	
2.4.37	CAS pulse width	tCP	20	-	ns	
2.4.38	CAS address setup time	tCASS	0	-	ns	
2.4.39	CAS address hold time	tCASH	25	-	ns	
2.4.40	CCK low to RGA valid prop time	tCKRGO	-	90	ns	
2.4.41	CCK low to RGA invalid prop time	tCLRGAOH	10	-	ns	
2.4.42	CCKQ high to Data valid prop time	tQHDO	0	150	ns	
2.4.43	CCK high to Data invalid prop tim	tCHDOH	0	85	ns	
2.4.44	CCK high to Early read Data prop	tCHEDO	0	125	ns	
2.4.45	Write command setup time	tWCS	-	20	ns	
2.4.46	Write command hold time	tWCH	45	-	ns	
2.4.47	CCKQ low to RGA valid prop time	tQLRGAO	-	110	ns	(4)

MISCELLANEOUS (Refer to Figure 9)

		SYMBOL	MIN	MAX	UNIT
2.4.48	LP*, DMAL input setup time	tIASCH	50	-	ns
2.4.49	LP* input hold time	tCHIAH	50	-	ns
2.4.50	DMUiL input hold time	tCHDMALH	40	-	ns
2.4.51	BLS* input setup time	tIASCL	50	-	ns
2.4.52	BLS* input hold time	tCLIAH	50	-	ns
2.4.53	VSY*, INT3* output prop time	tCHOB	10	110	ns
2.4.54	CSY*, HSY* output prop time	tCOB	0	110	ns
2.4.55	VSY*, HSY* input setup time	tIBSCH	40	-	ns
2.4.56	VSY*, HSY* input hold time	tCHIBH	30	-	ns

(1) Should be 4 X t14MC

(2) Should be 2 X t14MHi

(3) Should be 6 X t14MLo

(2) Only valid when RGEN* is low

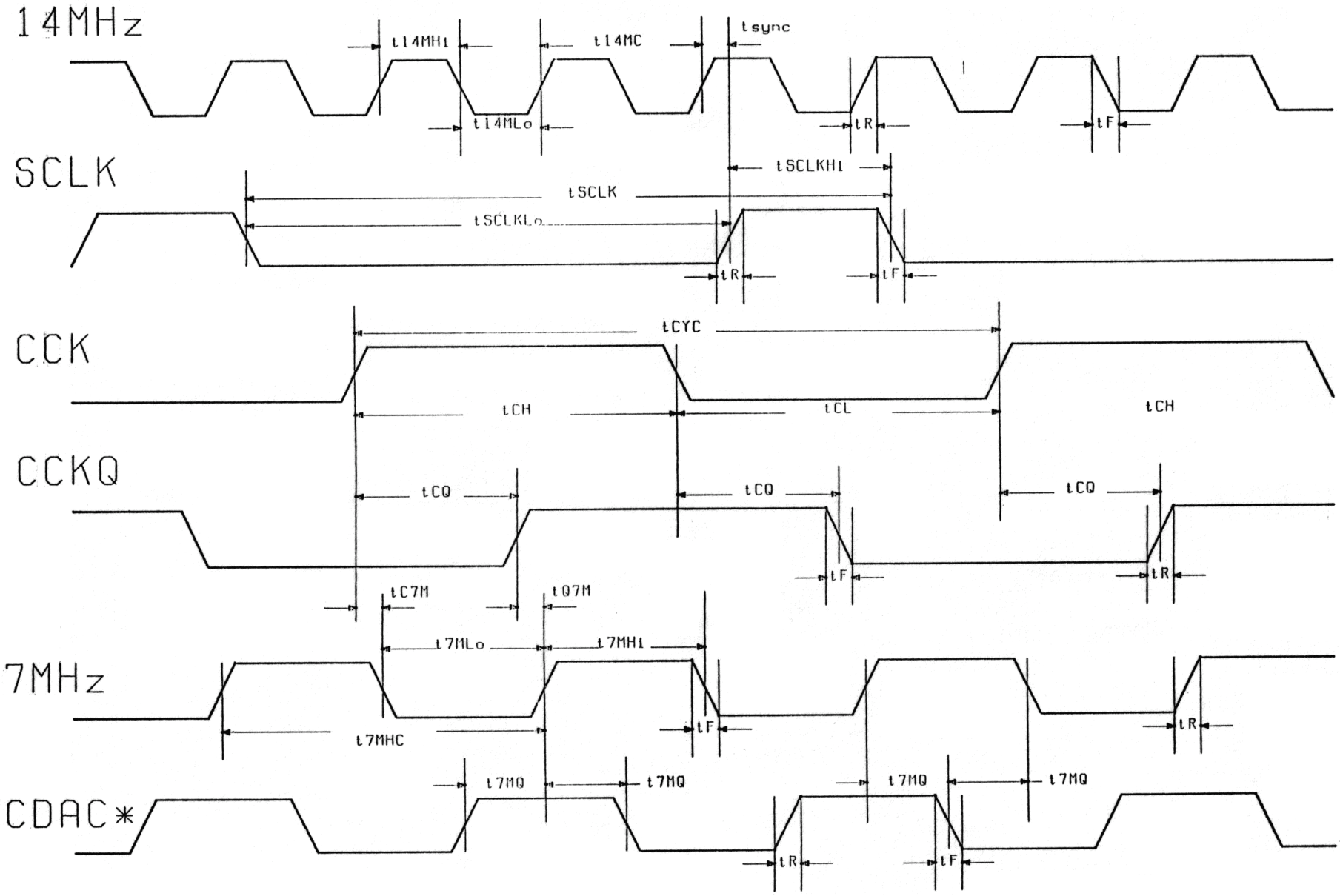

Fig. 3 CLOCK RELATIONS

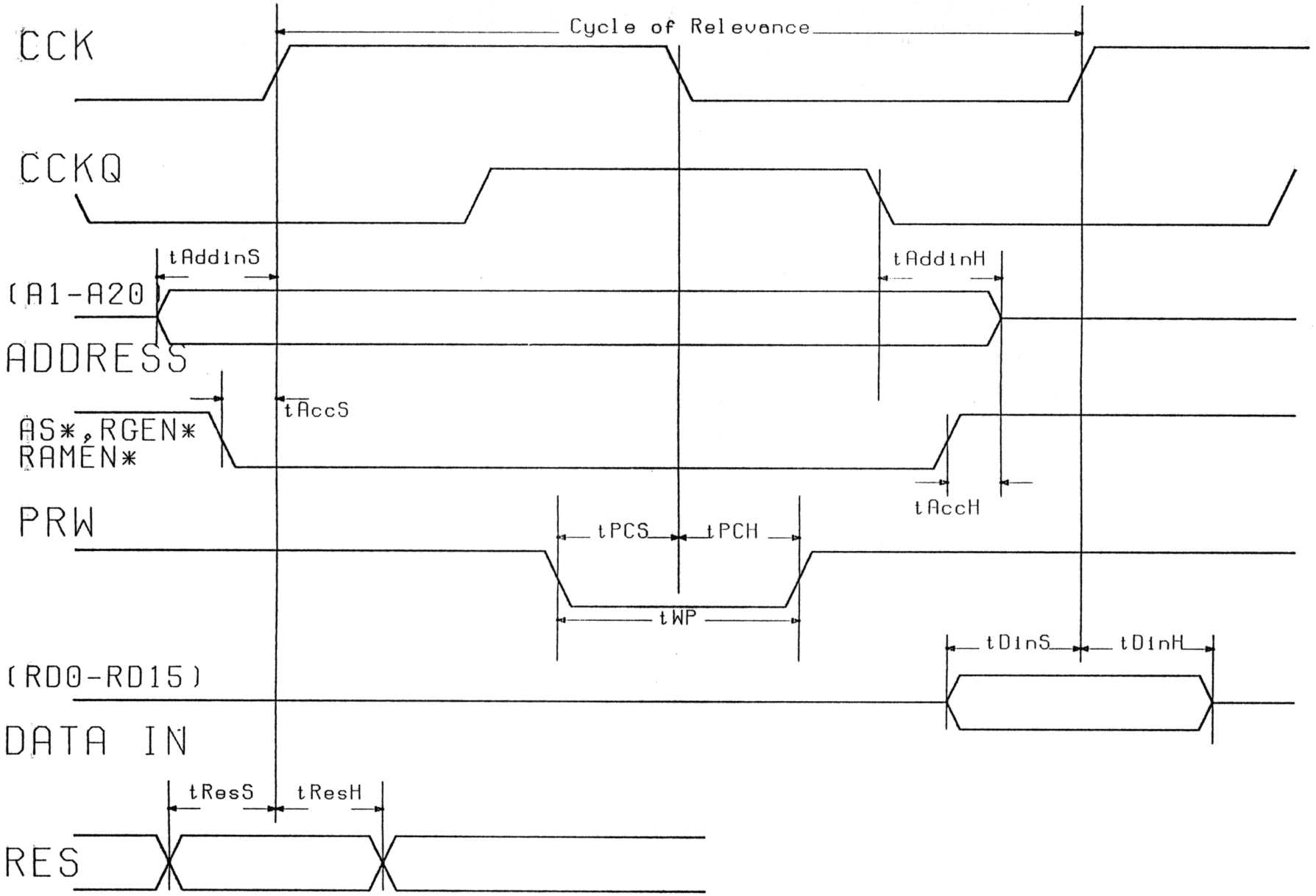

Fig. 4 PROCESSOR ACCESS

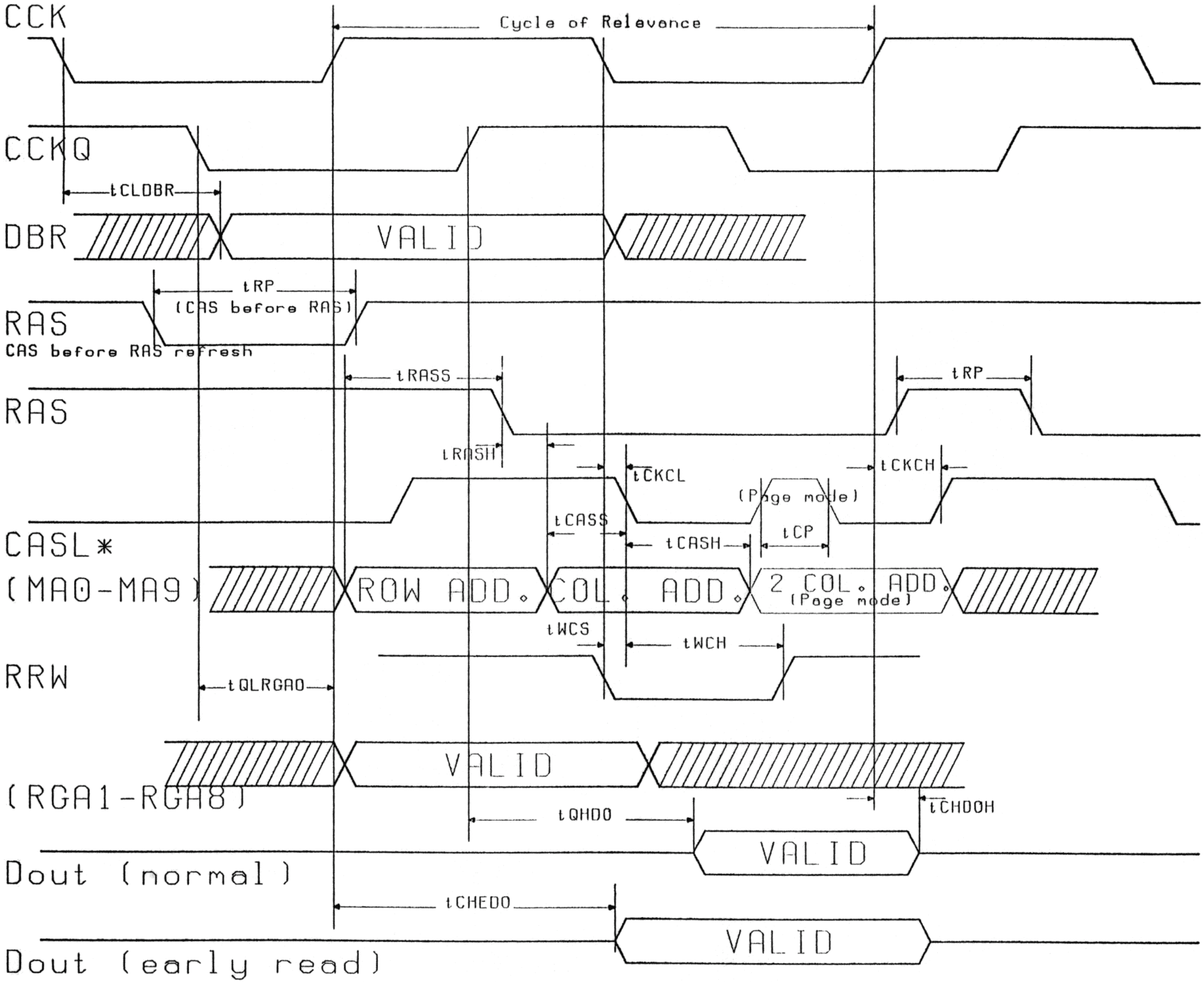

Fig. 5 DEVICE ACCESS

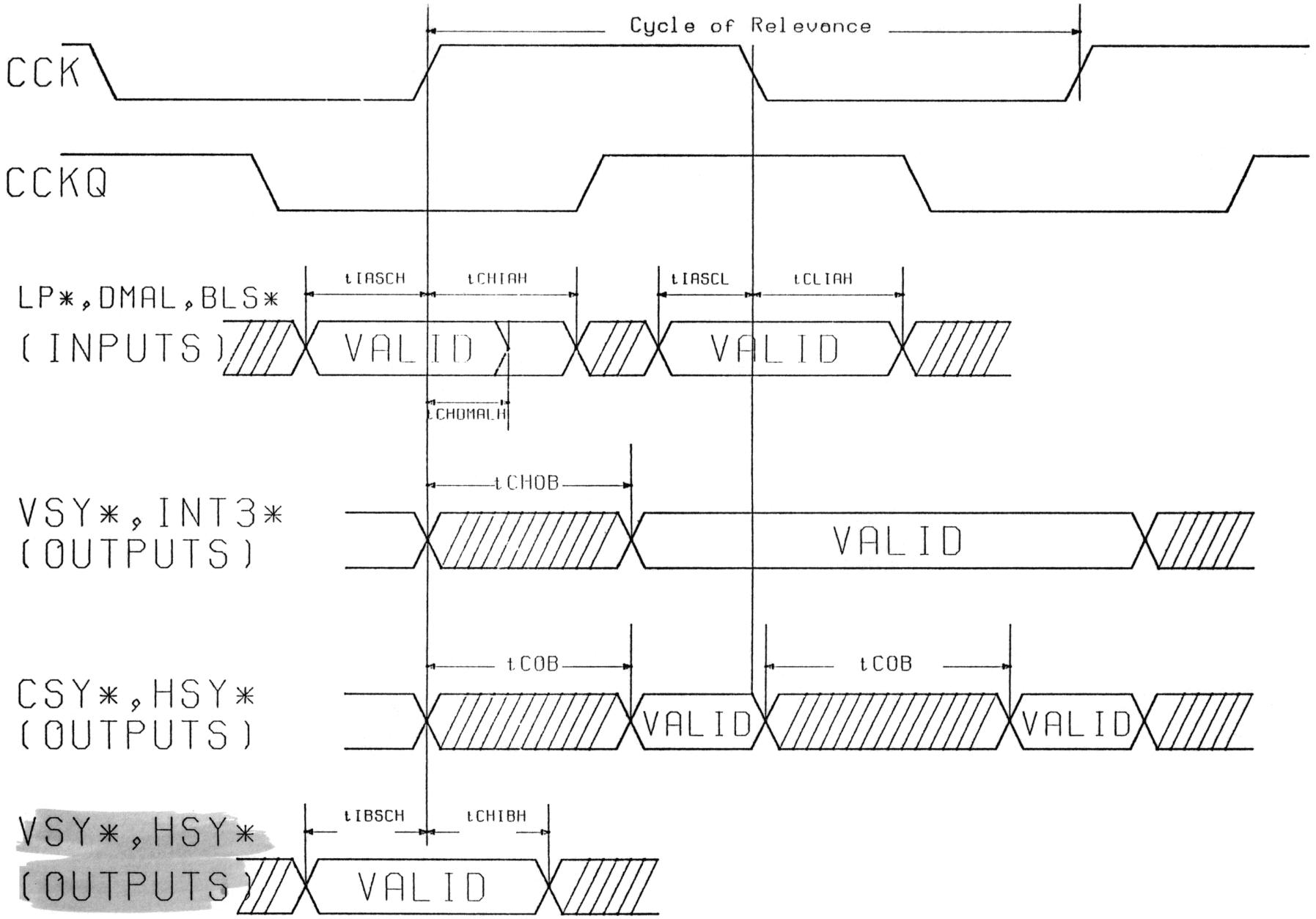

Fig. 6 MISCELLANEOUS

3.0 MODES OF OPERATION

3.1 GENERAL

This device is an address generator type IC. Its main function is as a RAM address generator and register address encoder that shall produce all DMA addresses for 27 channels.

The block diagram (figure 5) 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 RAM Address Generator contains an 20 bit pointer register for each of the 25 DMA channels and also it contains pointer restart (backup) registers and jump registers for six (6) of the channels. A full 20 bit adder carries out the pointer increments and adds for jumps.

The priority control logic looks at the pipe-lined DMA request 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.

The procedure for moving and combining bit mapped images in memory received the name Bit Blit from the computer instruction that did block transfer of data on bit boundries. The routines became known as Bit Blitters or Blitters. The Blitter DMA controller is preloaded with the address and size of 3 source images (A,B, and C) and one (1) destination (D) in the dynamic RAM (Refer to Figure 4). These images can be as small as a single character or a large as twice the screen size. They can be full images or smaller windows of a larger image. After one word of each source image is sequentially loaded into the source buffer (A,B,C) they are shifted and then combined together in the logic 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 deterines 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.

3.2 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 (3) source images (A, B, and C) and one (1) destination (D) in the dynamic RAM (refer to figure 11). 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. The actual pixel resolution is controlled by the BLTSIZE (BLTSIZH and BLTSIZV) registers which contain up to 15 bits for the image height (15 bits = 32K dots max.) and up to 11 bits for the image width (11 bits = 2k words = 32K pixels max.). After one word of each source image is sequentially loaded into the source buffer (A, B, C) they are shifted and then combined together in the logic 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 propogation time, while the next set of source words is being fetched.

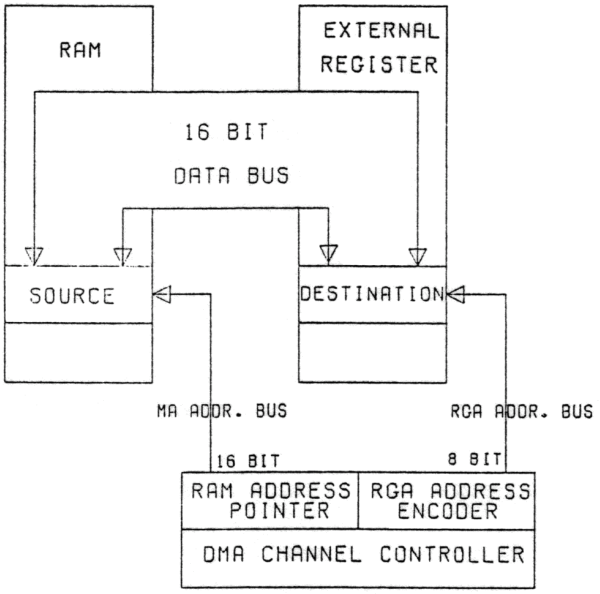

Fig. 7 DMA Channel (Typical)

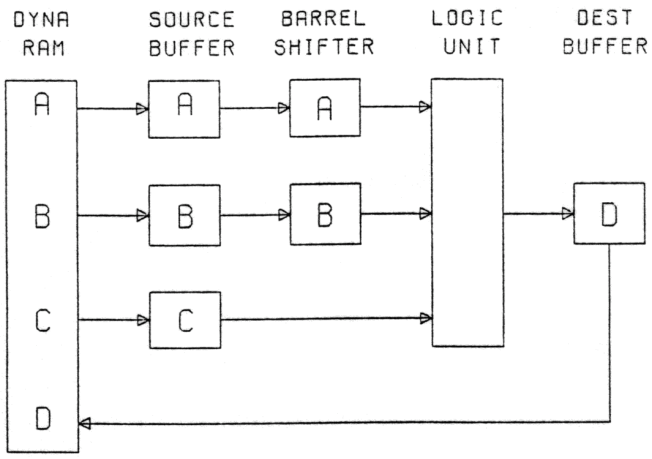

Fig. 8 Blitter Block Diagram

3.3 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 plane (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 2 separate images, each with independent hardware high speed image manipulation, line draw, and area fill.

3.4 DMA CHANNEL FUNCTIONS

Each channel has an 20 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 1M words (2M bytes) of RAM.

An eight (8) bit destination address is simultaneously placed on the register address bus (RGA), sending the data to the corresponding register.

Figure 10 shows a typical DMA channel and almost all channels have DRAM as source and chip registers as destination.

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.

Refer to Appendix A of this specification for raster line time allocation for each of these DMA channels.

A-Blitter (four (4) Channels)

The Blitter uses four (4) DMA channels, Three source and one (1) destination as previously described.

Once the Blitter has been started, the four (4) 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 the display (bitplane DMA).

B-Bitplane (eight (8) channels)

The bitplane controller continuously (during display) transfers display data from memory to display buffer registers. There are eight (8) DMA channels to handle the data from eight (8) independent bit planes. The buffers convert this bitplane dat into pixel data for the display.

C-Copper (one (1) Channel)

The Copper is a co-processor 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 the 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 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 (4) 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 (8) 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 their positions are controlled hardware registers and comparators.

Each sprite has two (2) sixteen bit data registers that define a 16 pixel wide Sprite with 4 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 2 Megabytes of memory depending on device pin configuration.

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

Each Sprite can be re-used vertically as often as desired. Horizontal re-using is also allowed with micoprocessor control.

F-Disk (one (1) channel)

The disk controller, which is located outside of the DMA, uses a single DMA channel from the device. The controller uses the DMA time slot for data transfer and can read or write a block of data up to 128K anywhere in 2 Megabytes of memory depending on device pin configuration.

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 (MA) during these slots, in order to refresh the dynamic RAM. Memory is refreshed on every raster 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 RAS1* and RAS are low. CASU* and CASL* are inactive during this cycle.

RAM AND REGISTER ADDRESSING

The device generates RAM addresses from two sources, the processor or the device performing DMA cycles. The processor accesses RAM whenever AS* and RAMEN* are both low. At this time, the device also multiplexes the processor address (A1-A20) onto the MA bus. During row address time A9- A17 and A19 are placed onto MA0-MA8, MA9, respectively; during column address time A1-A8, A18

and A20 are placed onto MA0-MA7, MA8 and MA9, respectively . In the 1 meg configuration, A19 is still used to determine the RAS line to be asserted. If A19 is low RAS0* is active and if high RAS1* is active. In the 2 meg option RAS will always be active on a RAM access. The IC will assert CASL* if LDS* is low or CASU* if UDS* is low.

When the device needs to do a DMA cycle, the device 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 activate RAS and the proper RAS0* or RAS1* line unless it is a refresh cycle where all RAS lines are active. During a DMA cycle, the IC device will also assert 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 an 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 device prevents the processor from doing a register access by asserting the DBR* line. The device will then place the contents of its register address encoder onto the RGA bus.

4.0 REGISTER DESCRIPTION

This DMA controller device contains 125 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. Refer to Table 2 for complete list of register addresses and type.

The following is a detailed description of the register set.

Register names followed by "H" signify new hi-res registers. Those followed by "h" signify hi-res enhancements.

REGISTER	*FUNCTION*
AUD x LCH (H)	AUDIO CHANNEL X LOCATION (HIGH 5 BITS)
AUD x LCL	AUDIO CHANNEL X LOCATION (LOW 15 BITS)

This pair of registers contains the 20 bit starting address (location) of Audio channel x (x= 1,2,3,4) DMA data. This is not a pointer register and therefore only needs to be reloaded if a different memory location is to be outputted.

REGISTER ***FUNCTION***

BEAMCON0

BIT #	*FUNCTION*
15	RESERVED
14	HARDDIS
13	LPENDIS
12	VARVBEN
11	LOLDIS
10	CSCBEN
9	VARVSYEN
8	VARHSYEN
7	VARBEAMEN
6	DUAL
5	PAL
4	VARCSYN
3	BLANKEN
2	CSYTRUE
1	VSYTRUE
0	HSYTRUE

HARDDIS This bit is used to disable the hardware vertical and horizontal window limits. This bit is cleared upon RESET.

LPENDIS When this bit is a logical 0 and lpe is enabled, the light pen latched value (hit location) will be read by VHPOSR, VPOSR and HHPOSR. When the bit is a logical 1 the light pen latched value is bypassed and the actual beam counter positon is read by VHPOSR, VPOSR and HHPOSR.

VARVBEN Use the comparator generated Vertical Blank (from VBSTART, VBSTOP) to run the internal chip stuff sending RGA signals to DENISE, starting Sprites, resetting light pen. It also disables the hard stop on the dispaly window.

LOLDIS Disable long line/short line toggle. This is useful for dual mode where even mukltiples are wanted, or in any single display where toggling is not desired.

CSCBEN The variable composit sync comes out on the HSY* pin, and the variable composit blank comes out on the VSY* pin. The idea is to allow all the information to come out of the chip the chip for dual mode dispaly. The normal monitor uses the normal composit sync, and the variable composit sync and blank come out of the HSY* and VSY* pins. The bits VARVSYEN and VARHSYEN have priority over this control bit.

VARVSYN Comparator for VSY ->> VSY* pin. The variable VSY is set vertically on VSSTRT, reset vertically on VSSTOP, with the horizontal position fro set and reset HSTRT on short fields (all field are short if LACE =1) and HCENTER on long fields (every other field if LACE = 1)

VARHSYN Comparator HSY ->> HSY* pin. Set on HSSTRT value, reset on HSSTOP value

VARBEAMEN Enables the variable beam counter comparators to operate (allowing different beam counter total values) on the main horizontal counter. It also disables hard dispaly stops both horizontal and vertical.

DUAL Run the horizontal comparators with the alternate horizontal beam counter, and starts the UHRES pointer chain with the reset of this counter rather than the normal one. This allows the UHRES pointer to come out more than once in a horizontal line, assuming there is some memory bandwidth left (it doesn't work in 640 * 400 * 4 interlace mode). Also, to keep the two displays synced, the horizontal line lenghts should be multiples of each other. If you are clever, you might not need to do this.

PAL Set the appropriate decodes (in normal mode) for PAL. In variable beam counter mode this bit disables the long/short line toggle (sets all lines short) When PAL is reset to a low, the device will enable the NTSC mode decades and the long/short line toggle is re-enabled.

VARCSYEN Enable CSY* from the variable decoders to come out the CSY* pin (VARCSY is set on HSSTRT match always and also on HCENTER match when in vertical sync). It is reset on HSSTOP match when VSY* and on both HBSTRT and HBSTOP matches during VSY. A reasonable composit can be generated by setting HCENTER half a horizontal line line from HSSTRT, and HSSTOP at (HSTOP-HSTRT) before HSSTRT.

BLANKEN Enable CB* (composit blank) to come out the CSY* pin. HB or VB that are generated from comparators. If neither BLANKEN or VARCSYEN are high, the normal CSY* from the regual decode comes out. This may be rather strange in variable beam modes, as some of the fixed decodes may not happen.

HSYTRUE, VSYTRUE, CSYTRUE These change the polarity of the HYS*, VSY*, and CSY* pin respectivly for input and output.

REGISTER	*FUNCTION*
BLT x PTH (h)	Blitter pointer to x (high 5 bits)
BLT x PTL	Blitter pointer to x (low 15 bits)

This pair of registers contains the 20 bit address of the Blitter source (x =a,b,c) or destination (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. BLTDPT and BLTCPT (both high and low) must be preloaded with the starting address of the line.

REGISTER	*FUNCTION*
BLT x MOD	Blitter modulo

This register contains the modulo for the blitter source (X= A,B,C) or the destination ((X = D). A modulo is a number that is automatically added to the address, which 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).

REGISTER	*FUNCTION*
BLTAFWM	Blitter firstword 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 set to all "ones" for fill mode or for line drawing mode.

REGISTER	*FUNCTION*
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. ...is used for texture. It must be preloaded with FF if no texture (solid line) is desired.

REGISTER	*FUNCTION*
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.

REGISTER	*FUNCTION*
BLTCON0	Blitter control register 0
BLTCON0L	Blitter control register 0 (write lower 8 bits only) This is to speed up software - the upper bits are often the same.
BLTCON1	Blitter control register 1

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

AREA MODE

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

LINE MODE

BIT #	*BLTCON0*	*BLTCON1*
15	ASH3	BSH3
14	ASH2	BSH2
13	ASH1	BSH1
12	ASA0	BSH0
11	1	0
10	0	0
09	1	0
08	1	0
07	LF7	DOFF
06	LF6	SIGN
05	LF5	OVF
04	LF4	SUD
03	LF3	SUL
02	LF2	AUL
01	LF1	SING
00	LF0	LINE(=1

ASH3-0	shift value of A source
BSH3-0	shift value of B source
USEA	mode control bit to use source 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	Logical minterm select lines
EFE	Exclusive fill enable
IFE	inclusive fill enable
FCI	fill carry input
DESC	decending (decreasing address) control bit
LINE	line mode control bit
SIGN	line draw sign flag
OVF	line draw r/l word overflow flag
SUD	line draw, Somtimes Up or Down (= AUD*)
SUL	Line draw, Sometimes Up or Left
AUL	Line draw, Always Up or Left

SING Line draw, Single bit per horizontal line
DOFF Disable D output - for external ALUs. The cycle occurs normally, but the data bus is tristated (hi res cips only)

The line draw octants are decoded as follows:

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	1	0	0

REGISTER	*FUNCTION*
BLTSIZE	Blitter start 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 will start 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 = Horizontal 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.

```
BLTSIZH H 05E W A       Blitter H size and start (11 bit width)

bit #   15 14 13 12 11 10  09  08  07  06  05  04  03  02  01  00
        X  X  X  X  X  W10 W9  W8  W7  W6  W5  W4  W3  W2  W1  W0

BLTSIZV H 05C W A       Blitter V size (15 bit height)

bit #   15 14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
        X  h14 h13 h12 h11 h10 h09 h08 h07 h06 h05 h04 h03 h02 h01 h0
```

REGISTER	*FUNCTION*
BPLxPTH	Bit plane x pointer (high 3 bits)
BPLxPTL	Bit plane x pointer (low 15 bits)

These are the blitter size registers for blits larger than the earlier chips could accept. The original commands are retained for compatabily.

BLTSIZV should be written first, followed by BLTSIZH, which starts the blitter. BLTSIZV need not be rewritten for subsequent blits if the vertical size is the same. The maximum size of a blit is 32K pixels * 32K lines. All don't cares (x's) should be written to 0's for upward compatability.

BPLHDAT H 07A W A	ext logic UHRES bit plane pointer identifier
BPLHMOD H 1E6 W A	UHRES bit plane modulo. This is the number (sign extended) that is addedd to the UHRES bit plane pointer (BPLHPTL,H) for every line, and then another 2 is added, just like the other modulos.
BPLHPTH H 1EC W A	UHRES (VRAM) bit plane pointer (high 5 bits)
BPLHPTL H 1EE W A	UHRES (VRAM) bit plane pointer (low 15 bits)

When UHRES is enabled, this pointer comes out on the second 'free' cycle after the start of each horizontal line. Its modulo is added every time it comes out. 'Free' means priority above COPPER and below the fixed stuff (audio, sprites...). BPLHDAT comes out as an identifier on the RGA lines when ithe pointer address is valid so that external detectors can use this to do special cycles for the VRAMs. The SHRHDAT get the first and third free cycles.

BPLHSTOP H 1D6 W A UHRES bit plane vertial stop

This controls the line when the data fetch stops for the BPLHPTH,L pointers. V10-V0 on DB10-0

BPLHSTRT H 1D4 W A UHRES bit plane pointer vertical stop.

This controls the line when the data fetch starts for the BPLHPTH,L pointers. V10-V0 on DB10-DB0.

REGISTER	*FUNCTION*
BPL x PTH	Bit plane x pointer (high 3 bits)
BPL x PTL	Bit plane 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 blanking time.

REGISTER	*FUNCTION*
BPL1MOD	Bit plane modulo (odd plane)
BPL2MOD	Bit plane modulo (even plane)

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 seperate modulos, the odd and even bit planes may have sizes that are different from each other, as well as different from the display widow size.

REGISTER	*FUNCTION*
BPLCON0 h	Bit plane control register (miscellaneous bits)

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

BIT#	*FUNCTION*
15	HIRES
14	BPU2
13	BPU1
12	BPU0
11	HAM
10	DPF
09	COLOR
08	GAUD
07	UHRES (1)
06	SHRES (1)
05	BYPASS
04	BPU3
03	LPEN
02	LACE
01	ERSY
00	X

(1) hi res chips only

HIRES = High resolution (640) mode
BPU = Bit plane use code 000-110 (NONE through 6 inclusive)
HAM = Hold and Modify mode
DPF = Double playfield (PF1=odd PF2=even bit planes)
COLOR = Composite video
COLOR = enable
GAUD = Genlock audio enable (mixed on BKGND pin during vertical blanking
UHRES = ultrahi res enables the UHRES pointers (for 1K x 1K) (also needs bits in DMACON)
SHRES = superhires (640 by 400 noninterlaced) sets the bit plane control for this mode - doubles the of the output of a given bit plane over HRES. (hi res chips only). Two bits planes maximum. If priority is less than 4, the 1 available sprite has priority. IF >>= 4, the sprite and bit plane are XOR'ed. Disables hard stops in vertical and horizontal display widows
BYPASS= Bitplanes are scrolled and prioritized normally, but bypass color table and 8 bit wide data appear on R(7:0)
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)

REGISTER	*FUNCTION*
COPCON h	Copper control register

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

BIT #	*NAME*	*FUNCTION*
01	CDANG	Copper danger mode. Allows Copper access to all RGA registers if true. Otherwise, it will access RGA addresses greater than 7E (hex).

REGISTER	*FUNCTION*
COPJMP1	Copper restart at first location
COPJMP2	Copper restart at second location

These addresses are strobe addresses, that when written to cause 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.

REGISTER	*FUNCTION*
COP1LCH h	Copper first location register (high 5 bits)
COP1LCL	Copper first location register (low 15 bits
COP2LCH h	Copper second location register (high 5 bits)
COP2LCL	Copper second location register (low 15 bits)
COPINS	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 destination
WAIT	Wait until the beam counter is equal to, or greater than. (keeps Copper off the bus until beam position has been reached)
SKIP	Skip if beam counter is equal to or greater than. (skips the following Move instruction unless beam position has been reached)

	MOVE		WAIT		SKIP	
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	HE8	HP8	HE8
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 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), from its wait state, or skip 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 WAIT) require 2 bus cycles (and two instruction words). Since only 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 20 bit

pointer registers whose contents are used to modify the program counter for initialization or jumps. They are transfered 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.

REGISTER	*FUNCTION*
DIWSTRT	Display window start (upper left vertical-horizontal positon)
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

DIWSTOP is vertically restricted to the lower 1/2 of the display (V8=/=V7), and horizontally restricted to the right 1/4 of the display (H8=1).

DIWSTRT is vertically restricted to the upper 2/3 of the display (V8=0), and horizontally restricted to the left 3/4 of the display (H8=0).

REGISTER	*FUNCTION*
DIWHIGH H	Display window upper bits for start, stop.

This is an added register for the hires chips, and allows larger start and stop ranges. If it is not written, the above (DIWSTRT, STOP description holds. If this register is written last in a sequence of setting the display widow, it sets direct start and stop positions anywhere on the screen. It does not affect the UHRES pointers.

BIT #	15,	14,	13,	12,	11,	10,	9,	8,	7,	6,	5,	4,	3,	2,	1,	0,
	X,	X,	H8,	X,	X,	V10,	V9,	V8,	X,	X,	H8,	X,	X,	V10,	V9,	V8

(X) don't care. Don't cares should always be written to 0 to maintain upward compatability.

REGISTER	*FUNCTION*
)DFSTRT	Display data fetch start (horiz.position)
)DFSTOP	Display data fetch stop (horiz.position)

These registers control the horizontal timing of the beginning and each end of the Bit Plane DMA display data fetch. The vertical Bit Plan 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 ASSIGNMENTS

BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USE	X	X	X	X	X	X	X	X	H8	H7	H6	H5	H4	H3	H2	X

X bits should alway be driven with 0 to maintain upward compatability

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)

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

* Note that these numbers will vary with variable beam counter modes (The maxes and mins that is)

REGISTER	*FUNCTION*
DMACON	DMA control write (clear or set)
DMACONR	DMA control read (and blitter status)

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 set with a 1 get set or cleared
14	BBUSY	Blitter busy status (read only)
13	BZERO	Blitter logic zero status bit (read only)
12	X	
11	X	
10	BLTPRI	Blitter DMA priority. (over CPU) Also called "Blitter Nasty", disable /BLS pin preventing micro from stealing

		Nasty", disable /BLS pin preventing micro from stealing any bus cycles while blitter DMA is running
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	AUD0EN	Audio channel 0 DMA enable

REGISTER	*FUNCTION*
DSKPTH h	Disk pointer (high 5 bits)
DSKPTL	disk pointer (low 15 bits)

This pair of registers contains the 20 bit address of the Disk DMA data. These address registers must be initialized by the processor or Copper before Disk DMA is enabled.

REGISTER	***FUNCTION***
FMODE	Fetch mode

This newly added register provides fetch mode for the bitplane and sprite fetches and scan doubling mode.

BIT #	*FUNCTION*	*DESCRIPTION*
15-08	-	unused
07	SODD	Sprite Odd line Scan doubling
06	SSCAN2	Sprite Scan doubling Enable
05	BODD	Bitplane Odd line Scan doubling
04	BSCAN2	Bitplane Scan doubling Enable
03	SPAGEM	Sprite Page Mode (double CAS)
02	SPR32	Sprite 32 Bit Wide Mode
01	BPAGEM	Bitplane Page Mode (double CAS)
00	BPL32	Bitplane 32 Bit Wide Mode

BPAGEM	*BPL32*	*Bitplane Fetch Increment*	*Memory Cycle*	*Bus Width*
0	0	by 2 bytes	normal CAS	16
0	1	by 4 bytes	normal CAS	32
1	0	by 4 bytes	double CAS	16
1	1	by 8 bytes	double CAS	32

REGISTER	*FUNCTION*
HBSTOP H	Horizontal line position for HBLANK stop
HBSTRT	Horizontal line position for HBLANK start. These are the start and stop positions (in 280 nS increments) for the HBLANK that comes out on the CSY* pin when BLANKEN bit in BEAMCONO is set to 1. It also affects VARSSY (see BEAMSONO).
HCENTER H	Horizontal line position (CCKs) of VSYNC on long field. This is nessesary for interlace mode with variable beam counters. See BEAMCONO for when it affects chip outputs. See HTOTAL for bits.
HHPOSR H	Dual mode hires H beam counter read.
HHPOSW H	Dual mode hires beam counter write. This is the secondary beam counter for the faster mode, triggering on UHRES pointers and doing comparisons for HBSTRT, STOP, HTOTAL, HSSTRT, HSSTOP (see HTOTAL for bits).
HSSTOP H	orizontal line position for HSYNC stop. Sets the number of color clocks for the sync stop (see HTOTAL for bits)
HSSTRT H	Horizontal line position for HSYNC start. Sets the number of color clocks for sync start (see HTOTAL for bits). See BEAMCONO for details when these two are active.
HTOTAL H	Highest color clock count in a horizontal line.

Bit # 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
X, X, X, X, X, X, X, X, h8,h7,h6,h5,h4,h3,h2,h1,

X's should be drive to 0 for upward compatability.
Horizontal line has this many +1 280 nS increments. If the PAL bit and LOLDIS are not both high, long/short line toggle will occur, and there will be this many +2 every other line. Active if VARBEAMEN = 1 or DUAL = 1.

REGISTER	*FUNCTION*
REFPTR	Refresh pointer

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

REGISTER	*FUNCTION*
SPR x PTH	Sprite x pointer (high 3 bits)
SPR x PTL	Sprite x pointer (low 15 bits)

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

REGISTER	*FUNCTION*
SPR x POS	Sprite x vertical - horizontal position data
SPR x CTL	Sprite x vertical - horizontal

This pair of 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 at any time.

SPRxPOS Register

BIT#	*SYMBOL*	*FUNCTION*
15-8	SV7-SV0	Start vertical value. High bit (SV8) is in SPRxCTL reg. below
07-00	SH8-SH0	Start vertical value. Low bit (SH0) is in SPRxCTL reg. below

SPRxCTL Register (writing this address disables sprite horizontal comparator circuit)

BIT#	*SYMBOL*	*FUNCTION*
15-8	EV7-EV0	End (stop) vertical value low 8 bits
07	ATT	Sprite attch control bit (odd sprites)

REGISTER	*FUNCTION*
VBSTOP H	Vertical line for VBLANK stop
VBSTRT H	Vertical line for VBLANK start (v10-0 <<- D10-0 affects CSY* pin if BLANKEN = 1 and VSY* pin if CSCBEN = 1 (see BEAMCON0).
VPOSR h	Read vertical most significant bits (and frame flop).
VPOSW	Write vertical most significant bits (and frame flop). BIT # 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, USE LOF I6, I5, I4, I3, I2,I1,I0,LOL, X, X, X, X,V10,V9,V8, LOF long frame (auto toggle control bit in BPLCON0) I0-I6 chip identification

8361 regular	or	8370 fat	agnus-ntsc	= 10
8367 pal	or	8371 fat-pal	agnus-ntsc	= 00
8368 hr	or	8372 fat hr	agnushr pal	= 20
8368 hr	or	8372 fat hr	agnushr ntsc	= 30

V9,V10 hires chips only 20,30 identifiers
LOL = long line bit. If logical 0 it indicates short line.