REGBITS.IAA (25-MAY-90) Stuff added for Pandora Hi Res chip set (rjr)

P= new register in Pandora Chip Set
p= stuff added or changedin hires chips
H= new register in hires chips
h= stuff added or changed in hires chips

ADKCON 09E W P Audio, Disk, Uart, Control write
ADKCONR 010 R P Audio, Disk, Uart, Control read

BIT#	USE	
15	SET/CLR	Set/Clear control bit. Determines if bits written with a 1 get set or cleared. Bits written with a zero are always unchanged.
14-13	PRECOMP 1-0	CODE PRECOMP VALUE 00 none 01 140 ns 10 280 ns 11 560 ns
12	MFMPREC	(1=MFM precomp 0=GCR precomp)
11	UARTBRK	Forces a UART break (clears TXD) if true
10	WORDSYNC	Enables disk read synchronizing on a word equal to DISK SYNC CODE, located in address DSKSYNC (7E).
09	MSBSYNC	Enables disk read synchronizing on the MSB (most signif bit). Appl type GCR
08	FAST	Disk data clock rate control 1=fast(2us) 0=slow(4us) (fast for MFM or 2us GCR, slow for 4us GCR)
07	USE3PN	Use audio channel 3 to modulate nothing
06	USE2P3	Use audio channel 2 to modulate period of channel 3
05	USE1P2	Use audio channel 1 to modulate period of channel 2
04	USE0P1	Use audio channel 0 to modulate period of channel 1
03	USE3VN	Use audio channel 3 to modulate nothing
02	USE2V3	Use audio channel 2 to modulate volume of channel 3
01	USE1V2	Use audio channel 1 to modulate volume of channel 2
00	USE0V1	Use audio channel 0 to modulate volume of channel 1

NOTE If both period and volume are modulated on the same channel, the period and volume wil be alternated.
First AUDxDAT word is used for V6-V0 of AUDxVOL
Second AUDxDAT word is used for P15-P0 of AUDxPER
this alternating sequence is repeated

AUDxLCH h 0A0 W A Audio channel x location (High 5 bits) (old-3 bits)
AUDxLCL 0A2 W A Audio channel x location (Low 15 bits)

This pair of registers contains the 20 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.

AUDxLEN 0A4 W P Audio Channel x length

This register contains the length (number of words) of Audio Channel x DMA data.

AUDxPER h 0A6 W P Audio channel x Period

This register contains the Period (rate) of Audio channel x DMA data transfer.

The minimum period is 124 color clocks. This means that the smallest number that should be placed in this register is 124. This corresponds to a maximum sample frequency of 28.86 khz. (124 decimal = 7C hex). For running with a horiz. frequency that is different (VARBEAMEN=1) the sampling frequency can be higher. Minimum period is (HTOTAL+21)/2 color clocks. See BEAMCON0 for details of varbeamen.

AUDxVOL 0A8 W P Audio Channel x Volume

This register contains the Volume setting for Audio Channel x. Bits 6,5,4,3,2,1,0 specify 65 linear volume levels as shown below.

BITS	USE
15-07	Not used
06	Forces volume to max (64 ones, no zeros)
05-00	Sets one of 64 levels (000000=no output (111111=63 0nes, one zero)

AUDxDAT 0AA W P Audio channel x Data

This register is the Audio channel x (x=0,1,2,3) DMA data buffer. It contains 2 bytes of data (each byte is a twos complement signed integer) that are outputed sequentially (with digital to analog conversion) to the audio output pins. With maximum volume, each byte can drive the audio outputs with 0.8 volts(peak to peak, typ) The audio DMA channel controller automatically transfers data to this register from RAM. The Processor can also write directly to this register. When the DMA data is finished (words outputted=Length) and the data in this register has been used, an audio channel interrupt request is set.

BEAMCON0 H 1DC W A Beam Counter control bits

Bit	Function
15	(unused)
14	HARDDIS
13	LPENDIS
12	VARVBEN
11	LOLDIS
10	CSCBEN
9	VARVSYEN
8	VARHSYEN
7	VARBEAMEN
6	DUAL
5	PAL
4	VARCSYEN
3	(unused, formerly BLANKEN)
2	CSYTRUE
1	VSYTRUE
0	HSYTRUE

HARDDIS= This bit is used to disable the hardwire vertical and horizontal window limits. It is cleared upon reset.
LPENDIS= When this bit is a low and LPE (BPLCON0,BIT 3) is enabled, the light-pen latched value (beam hit position) will be read by VHPOSR, VPOSR and HHPOSR. When the bit is a high the light-pen latched value is ignored and the actual

beam counter position is read by VHPOSR, VPOSR and HHPOSR.
VARVBEN= Use the comparator generated Vertical Blank (from VBSTRT,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 vertical display window.
LOLDIS= Disable long line/short line toggle. This is useful for DUAL mode where even multiples are wanted, or in any single display where this toggling is not desired.
CSCBEN= The variable composite sync comes out on the HSY* pin, and the variable composite blank comes out on the VSY* pin. The idea is to allow all the information to come out of the chip for a DUAL mode display. The normal monitor uses the normal composite sync, and the variable composite sync & blank come out the HSY* & VSY* pins. The bits VARVSYEN & VARHSYEN (below) have priority over this control bit.
VARVSYEN= comparator VSY -> VSY* pin. The variable VSY is set vertically on VSSTRT, reset vertically on VSSTOP, with the horizontal position for set & reset HSSTRT on short fields (all fields are short if LACE=0) and HCENTER on long fields (every other field if LACE=1)
VARHSYEN= 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 horiz counter. It also disables hard display stops on both horizontal & 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 pointers 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 2 displays synced, the horizontal line lengths should be multiples of each other. If you are amazingly clever, you might not need to do this.
PAL= Set appropriate decodes (in normal mode) for PAL. In variable beam counter mode this bit disables the long line/short line toggle- ends up short line.
VARCSYEN= enable CSY* from the variable decoders to come out the CSY* (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 & HBSTOP matches during VSY. A reasonable composite can be generated by setting HCENTER half a horiz line from HSSTRT, and HBSTOP at (HSSTOP-HSSTRT) before HCENTER, with HBSTRT at (HSSTOP-HSSTRT) before HSSTRT.
HSYTRUE,VSYTRUE,CSYTRUE= These change the polarity of the HSY*, VSY*, & CSY* pins to HSY, VSY,& CSY respectively for input & output.

BLTxPTH	h 050	W	A	Blitter Pointer to x (High 5 bits)
BLTxPTL	052	W	A	Blitter Pointer to x (Low 15 bits)

This pair of registers contains the 20 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).

BLTxMOD 064 W A Blitter Modulo x

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 at the end of each line, in order that the address then points to the start of the next line. Each source or destination has it's own Modulo, allowing each to be a different size, while an identical area of each is used in the Blitter operation.

BLTAFWM 044 W A Blitter first word mask for source A
BLTALWM 046 W A 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 overides data from Source A. These registers should be set to all "ones" for fill mode or for line drawing mode.

BLTxDAT 074 W A Blitter source x data reg

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.

BLTDDAT 000 W A 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 040 W A Blitter control register 0
BLTCON0L H 05A W A Blitter control register 0 (write lower 8 bits only)

This is to speed up software-- the upper bits are often the same.

BLTCON1 h 042 W A 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")		LINE MODE (line draw)		
BIT#	BLTCON0	BLTCON1	BIT#	BLTCON0	BLTCON1
15	ASH3	BSH3	15	ASH3	BSH3
14	ASH2	BSH2	14	ASH2	BSH2
13	ASH1	BSH1	13	ASH1	BSH1
12	ASA0	BSH0	12	ASH0	BSH0
11	USEA	0	11	1	0
10	USEB	0	10	0	0
09	USEC	0	09	1	0
08	USED	0	08	1	0
07	LF7	DOFF	07	LF7	DOFF
06	LF6	0	06	LF6	SIGN
05	LF5	0	05	LF5	OVF
04	LF4	EFE	04	LF4	SUD
03	LF3	IFE	03	LF3	SUL
02	LF2	FCI	02	LF2	AUL
01	LF1	DESC	01	LF1	SING
00	LF0	LINE(=0)	00	LF0	LINE(=1)

ASH3-0 Shift value of A source

```
                BSH3-0  Shift value of B source and line texture
                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   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
                SIGN    Line draw sign flag
                OVF     Line draw r/l word overflow flag
                SUD     Line draw, Sometimes Up or Down (=AUD*)
                SUL     Line draw, Sometimes Up or Left
                AUL     Line draw, Always Up or Left
                SING    Line draw, Single bit per horiz. line
                DOFF    Disables the D output- for external ALUs
                        The cycle occurs normally, but the data bus is
                        tristate.(hires chips only)

BLTSIZE   058  W   A       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 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 =Horiz pixels (6 bits=64 words=1024 pixels max)

BLTSIZH H 05E  W   A       Blitter H size & 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  12  12  11  10  09 08 07 06 05 04 03 02 01 00
                     x h14 h13 h12 h11 h10 h9 h8 h7 h6 h5 h4 h3 h2 h1 h0
              These are the blitter size registers for blits larger than
              the earlier chips could accept. The original commands are
              retained for compatibility. 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. max size of blit 32k pixels * 32k lines
              x s should be written to 0 for upward compatibility.

BPLHDAT H 07A  W ext logic UHRES bit plane pointer identifier

BPLHMOD H 1E6  W   A       UHRES bit plane modulo
              This is the number (sign extended) that is added to the
              UHRES bit plane pointer (BPLHPTL,H) every line, and then
              another 2 is added, just like the other modulos.


BPLHPTH H 1EC  W   A       UHRES (VRAM) bit plane pntr (High 5 bits)
BPLHPTL H 1EE  W   A       UHRES (VRAM) bit plane pntr (Low 15 bits)
              When UHRES is enabled, this pointer comes out on the 2nd
              'free' cycle after the start of each horiz. line. Its modulo
              is added every time it comes out. 'Free' means priority
              above the copper and below the fixed stuff (audio,sprites...)
```

BPLHDAT comes out as an identifier on the RGA lines when the pointer address is valid so that external detectors can use this to do the special cycle for the VRAMs. The SHRHDAT gets the first and third free cycles.

BPLHSTOP p 1D6 W A UHRES bit plane vertical stop
This controls the line when the data fetch stops for the BPLHPTH,L pointers.

Bit#	Name
15	BPLHWRM
14-11	unused
10-0	V10-V0

BPLHWRM= Swaps the polarity of ARW* when the BPLHDAT comes out so that external devices can detect the RGA and put things into memory (ECS and later versions).

BPLHSTRT H 1D4 W A UHRES bit plane vertical start
This controls the line when the data fetch starts for the BPLHPTH,L pointers. V10-V0 on DB10-0.

BPLxPTH 0E0 W A Bit plane x pointer (High 5 bits)
0E4 x=1,2,3,45,6,7,8
0E8
0EC
0F0
0F4
P 0F8
P 0FC

BPLxPTL 0E2 W A Bit plane x pointer (Low 15 bits)
0E6 Address of Bit plane x (x=1,2,3,4,5,6,7,8) DMA data. This
0EA pointer must be reinitalized by the processor or coprocessor
0EE to point to the beginning of Bit Plane data every vertical
0F2 blank time.
0F6
P 0FA
P 0FE

BPLxDAT 110 W D Bit plane x data (Parallel to serial convert)
112 These registers receive the DMA data fetched from RAM
114 by the Bit Plane address pointers described above.
116 They may also be written by either micro.
118 They act as a 8 word parallel to serial buffer
11A for up to 8 memory "Bit Planes". (x=1-8 The parallel to
P 11C serial conversion is triggered whenever bit plane #1
P 11E is written, indicing the completion of all bit planes
for that word (16/32/64 pixels). The MSB is output first, and
is therefore always on the left.

BPL1MOD 108 W A Bit plane modulo (odd planes)
BPL2MOD 10A W A 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.

```
BPLCON0 p 100  W   A D     Bit plane control reg.(misc control bits)
BPLCON1 p 102  W     D     Bit plane control reg.(horiz. scroll control)
BPLCON3 p 104  W     D     Bit plane control reg.(new control bits)
BPLCON3 p 106  W     D     Bit Plane control reg.(enhanced features)
BPLCON4 P 10C  W     D     BIt Plane control reg.(display masks)
```

These registers control the operation of the Bit Planes and various aspects of the display.

BIT#	BPLCON0	BPLCON1	BPLCON2	BPLCON3	BPLCON4
15	HIRES	PF2H7=0	x	BANK2=0	BPLAM7=0
14	BPU2	PF2H6=0	ZDBPSEL2	BANK1=0	BPLAM6=0
13	BPU1	PF2H1=0	ZDBPSEL1	BANK0=0	BPLAM5=0
12	BPU0	PF2H0=0	ZDBPSEL0	PF2OF2=0	BPLAM4=0
11	HAM	PF1H7=0	ZDBPEN	PF2OF1=1	BPLAM3=0
10	DPF	PF1H6=0	ZDCTEN	PF2OF0=1	BPLAM2=0
09	COLOR	PF1H1=0	KILLEHB	LOCT=0	BPLAM1=0
08	GAUD	PF1H0=0	RDRAM=0	x	BPLAM0=0
07	UHRES	PF2H5	SOGEN=0	SPRES1=0	ESPRM7=0
06	SHRES	PF2H4	PF2PRI	SPRES0=0	ESPRM6=0
05	BYPASS=0	PF2H3	PF2P2	BRDRBLNK=0	ESPRM5=0
04	BPU3=0	PF2H2	PF2P1	BRDNTRAN=0	ESPRM4=0
03	LPEN	PF1H5	PF2P0	x	OSPRM7=0
02	LACE	PF1H4	PF1P2	ZDCLKEN=0	OSPRM6=0
01	ERSY	PF1H3	PF1P1	BRDSPRT=0	OSPRM5=0
00	ECSENA=0	PF1H2	PF1P0	EXTBLKEN=0	OSPRM4=0

x= don’t care; but drive to 0 for upward compatibility !
=0/=1 bit values initialized by RST_ pin going low.

HIRES=High resolution(640*200/640*400interlace) mode
BPUx =Bit plane use code 0000-1000 (NONE thru 8 inclusive)
HAM=Hold and Modify mode,now using either 6 or 8 bitplanes.
New HAM mode is invoked when this bit is set and
BPU =1000. Now available in all resolutions.
DPF=Double playfield (PF1=odd PF2=even bit planes)
now available in all resolutions
(If BPU=6 and HAM=0 and DPF=0 a special mode is defined that allows bitplane 6 to cause an intensity reduction of the other 5 bitplanes. The color register output selected by 5 bitplanes is shifted to half intensity by the 6th bitplane. This is called EXTRA-HALFBRITE Mode.
COLOR= enables color burst output signal.
GAUD=Genlock audio enable. This level appears on the ZD pin on Denise during all blanking periods, unless ZDCLK bit is set.
UHRES= ultrahi res enables the UHRES pointers (for 1k*1k) (also needs bits in DMACON) (hires chips only) Disables hard stops for vert., horiz. display windows
SHRES= super-hi-res mode (35nS pixel width)
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)

ECSENA= When set enables all bits not present in the original amiga chip set, when reset Denise returns to normal operation.
PF2Hx= Playfield 2 horizontal scroll code,x=0-7
PF1Hx= Playfield 1 horizontal scroll code,x=0-7
where PFyH0=LSB=35nS=1 SHRES pixel(bits have been renamed,old PFyH0 now PFyH2,etc.). Note that the scroll range has been quadrupled to allow for wider (32 or 64 bit) bitplanes.
ZDBPSELx= 3 bit field which selects which Bit plane is to be used for ZD when ZDBBPEN is set;000 selects BP1 and 111 selects BP8.
ZDBPEN= causes ZD pin to mirror bitplane selected by ZDBPSELx bits. This does not disable the ZD mode defined by ZDCTEN, but rather is "ored" with it.
ZDCTEN= causes ZD pin to mirror bit #15 of the active entry in the high color table. When ZDCTEN is reset ZD reverts to mirroring color(0).
KILLEHB= disables Extra Half Brite mode.
RDRAM= causes color table addresses to read the color table instead of writing to it.
SOGEN= when set causes SOG output pin to go high
PF2PRI= gives Playfield 2 priority over Playfield 1.
PF2Px= Playfield 2 priority code (with resp. to sprites)
PF1Px= Playfield 1 priority code (with resp. to sprites)
BANKx= selects one of eight color table banks, x=0-2.
PF2OFx= determine bitplane color table offset when playfield 2 has priority in dual playfield mode:

PF2OF			AFFECTED BITPLANE								OFFSET
2	1	0	8	7	6	5	4	3	2	1	(decimal)
0	0	0	-	-	-	-	-	-	-	-	none
0	0	1	-	-	-	-	-	-	1	-	2
0	1	0	-	-	-	-	-	1	-	-	4
0	1	1	-	-	-	-	1	-	-	-	8 (default)
1	0	0	-	-	-	1	-	-	-	-	16
1	0	1	-	-	1	-	-	-	-	-	32
1	1	0	-	1	-	-	-	-	-	-	64
1	1	1	1	-	-	-	-	-	-	-	128

LOCT= dictates that subsequent color palette values will be written to a second 12-bit color palette, constituting the RGB low-order bits. Writes to the normal high-order color palette are automatically copied to the low-order for backwards compatibility.
SPRESx= determine resolution of all 8 sprites (x=0,1):

SPRES1	SPRES0	SPRITE RESOLUTION
0	0	ECS defaults(LORES,HIRES=140nS,SHRES=70nS
0	1	LORES(140nS)
1	0	HIRES(70nS)
1	1	SHRES(35nS)

BRDRBLNK= "border area" is blanked instead of color(0).
BRDNTRAN= "border area" is non-transparent(ZD pin is low when border is displayed).
ZDCLKEN= ZD pin outputs a 14MHZ clock whose falling edge coincides with high-res(7MHZ) video data. This bit when

set disables all other ZD functions.
BRDRSPRT= enables sprites outside the display window.
EXTBLKEN= causes BLANK output to be programmable instead of reflecting internal fixed decodes.
BPLAMx= this 8 bit field is XOR'ed with the 8 bitplane color address, thereby altering the color address sent to the color table(x=1-8).
ESPRMx= 4 bit field provides the 4 high order color table address bits for even sprites:SPR0,SPR2,SPR4,SPR6. Default value is 0001 binary.(x=7-4)
OSPRMx= 4 bit field provides the 4 high order color table address bits for odd sprites:SPR1,SPR3,SPR5,SPR7. Default value is 0001 binary.(x=7-4)

CLXCON 098 W D Collision control

This register controls which Bitplanes are included (enabled) in collision detection, and their required state if included. It also controls the individual inclusion of odd numbered sprites in the collision detection, by logically ORing them with their corresponding even numbered sprite.

BIT#	FUNCTION	DESCRIPTION
15	ENSP7	Enable Sprite 7 (ORed with Sprite 6)
14	ENSP5	Enable Sprite 5 (ORed with Sprite 4)
13	ENSP3	Enable Sprite 3 (ORed with Sprite 2)
12	ENSP1	Enable Sprite 1 (ORed with Sprite 0)
11	ENBP6	Enable Bit Plane 6 (Match reqd. for collision)
10	ENBP5	Enable Bit Plane 5 (Match reqd. for collision)
09	ENBP4	Enable Bit Plane 4 (Match reqd. for collision)
08	ENBP3	Enable Bit Plane 3 (Match reqd. for collision)
07	ENBP2	Enable Bit Plane 2 (Match reqd. for collision)
06	ENBP1	Enable Bit Plane 1 (Match reqd. for collision)
05	MVBP6	Match value for Bit Plane 6 collision
04	MVBP5	Match value for Bit Plane 5 collision
03	MVBP4	Match value for Bit Plane 4 collision
02	MVBP3	Match value for Bit Plane 3 collision
01	MVBP2	Match value for Bit Plane 2 collision
00	MVBP1	Match value for Bit Plane 1 collision

CLXCON2 P 10C W D Extended Collision Control

This register controls when bitplanes 7 & 8 are included in collision detection, and their required state if included.
***** BITS INITIALIZED BY RESET *****

BIT#	FUNCTION	DESCRIPTION
15-08	-	unused
07	ENBP8	Enable Bit Plane 8 (Match reqd. for collision)
06	ENBP7	Enable Bit Plane 7 (Match reqd. for collision)
05-02	-	unused
01	MVBP8	Match value for Bit Plane 8 collision
00	MVBP7	Match value for Bit Plane 7 collision

Note: Disabled Bit Planes cannot prevent collisions. Therefore if all bitplanes are disabled, collisions will be continuous, regardless of the match values.

CLXDAT 00E R D Collision data reg.(Read and clear)

This address reads (and clears) the collision detection register. The bit assignments are below.

NOTE: Playfield 1 is all odd numbered enabled bit planes.
Playfield 2 is all even numbered enabled bit planes

BIT#	COLLISIONS REGISTERED
15	not used
14	Sprite 4 (or 5) to Sprite 6 (or 7)
13	Sprite 2 (or 3) to Sprite 6 (or 7)
12	Sprite 2 (or 3) to Sprite 4 (or 5)
11	Sprite 0 (or 1) to Sprite 6 (or 7)
10	Sprite 0 (or 1) to Sprite 4 (or 5)
09	Sprite 0 (or 1) to Sprite 2 (or 3)
08	Playfield 2 to Sprite 6 (or 7)
07	Playfield 2 to Sprite 4 (or 5)
06	Playfield 2 to Sprite 2 (or 3)
05	Playfield 2 to Sprite 0 (or 1)
04	Playfield 1 to Sprite 6 (or 7)
03	Playfield 1 to Sprite 4 (or 5)
02	Playfield 1 to Sprite 2 (or 3)
01	Playfield 1 to Sprite 0 (or 1)
00	Playfield 1 to Playfield 2

COLORxx 180-1BE W COLOR table xx

There are thirty-two (32) of these registers (xx=00-31) and together with the banking bits they address the 256 locations in the color palette. There are actually two sets of color registers, selection of which is controlled by the LOCT register bit. When LOCT=0 the 4 MSB of RED,GREEN, and BLUE video data are selected along with the T bit for genlocks. The low order set of registers is also selected simultaneously, so that the 4 bit values are automatically extended to 8 bits. This provides compatibility with old software. If the full range of palette values are desired, then LOCT can be set high and independant values for the 4 LSB of RED,GREEN,and BLUE can be written. The low order color registers do not contain a transparency(T) bit. The Table below shows the color register bit usage.

BIT #	15,14,13,12	11,10,09,08	07,06,05,04	03,02,01,00
LOCT=0	T X X X	R7 R6 R5 R4	G7 G6 G5 G4	B7 B6 B5 B4
LOCT=1	X X X X	R3 R2 R1 R0	G3 G2 G1 G0	B3 B2 B1 B0

T = TRANSPARENCY R = RED G = GREEN B = BLUE X = UNUSED

T bit of COLOR00 thru COLOR31 sets ZD_ pin HI when that color is selected in all video modes.

COPCON h 02E W A Coprocessor control register

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

BIT#	NAME	FUNCTION
01	CDANG	Coprocessor danger mode. Allows coprocessor access to all RGA registers if true. (if 0, access to RGA>7E) (on old chips- access to only RGA>3E if CDANG=1) (see VPOSR)

COPJMP1 088 S A Coprocessor restart at first location

COPJMP2 08A S A Coprocessor restart at second location

These addresses are strobe addresses, that when written to cause the Coprocessor to jump indirect using the address contained in the First or Second Location registers described below. The Coprocessor itself can write to these addresses, causing it's own jump indirect.

COP1LCH h 080 W A Coprocessor first location reg (High 5 bits) (old-3 bits)
COP1LCL 082 W A Coprocessor first location reg. (Low 15 bits)
COP2LCH h 084 W A Coprocessor second location reg. (High 5 bits)(old-3 bits)
COP2LCL 086 W A Coprocessor second location reg(Low 15 bits)

These registers contain the jump addresses described above.

COPINS 08C W A Coprocessor inst. fetch identify

This is a dummy address that is generated by the Coprocessor whenever it is loading instructions into it's own instruction register. This actually occurs every Coprocessor 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 Coprocessor 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	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	0	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 Coprocessor. When this bit is zero

the Blitter Finished flag must be true
(in addition to the rest of the bit comparisons)
before the Coprocessor can exit from its wait
state, or skip over an instruction. Note that the V7
comparison cannot be masked.

The Coprocessor is basically a 2 cycle machine that requests the bus only during odd memory cycles. (4 memory cycles per in) It has priority over only the Blitter and Micro.

There are only three types of instructions, MOVE immediate, WAIT until ,and SKIP if. All instructions 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 20 bit pointer registers whose contents are used to modify the program counter for initalization or jumps. They are transfered to the program counter whenever strobe addreses 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 initalized and it's jump strobe address hit, after power up but before Coprocessor DMA is initalized. This insures a determined startup address, and state.

DDFSTRT 092 W A Display data fetch start(Horiz.Position)
DDFSTOP 094 W A 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	H8	H7	H6	H5	H4	H3	H2	X

(X bits should always 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 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
                  Note that these numbers will vary with variable beam counter
                   mode set. (The maxes & mins, that is)

DIWSTRT   08E  W   A D     Display Window Start (upper left vert-hor pos)
DIWSTOP   090  W   A D     Display Window Stop (lower right vert-hor pos)
                  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 H9 H8 H7 H6 H5 H4 H3 H2

                  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).*
                  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).*
                  * Poof.. ( See DIWHIGH for exceptions)

DIWHIGH p 1E4  W   A D     Display Window upper bits for  start,stop
                  This is an added register for the Hires chips, and allows
                  larger start & 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 window, it sets
                  direct start & stop positions anywhere on the screen.
                  It doesn't affect the UHRES pointers.

         Bit#   15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
        ------------------------------------------------------------------------
                x   x   H10 H1  H0  V10 V9  V8  x   x   H10 H1  H0  V10 V9  V8
                                  (stop)        |         (start)

                  Don't care (x) bits should always be written to 0 to
                  maintain upwards compatibility. H1 and H0 values define
                 70nS and 35nS increments respectively, and are new LISA
                 bits.

                 NOTE:  In all 3 display window registers, horizontal bit
                       positions have been renamed to reflect HIRES pixel
                       increments, e.g. what used be called H0 is now
                       referred to as H2.

DMACON    096  W   A D P   DMA control write(clear or set)
DMACONR   002  R   A   P   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.
                                    Bits written with a zero are unchanged.
                  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)
```

```
                09    DMAEN       Enable all DMA below (also UHRES dma)
                08    BPLEN       Bit Plane DMA enable
                07    COPEN       Coprocessor 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

DSKPTH  h 020  W   A       Disk pointer (High 5 bits) (old-3 bits)
DSKPTL    022  W   A       Disk pointer (Low 15 bits)
              This pair of registers contains the 20 bit address
              of Disk  DMA data. These address
              registers must be initalized by the processor or coprocessor
              before disk DMA is enabled.

DSKLEN    024  W       P   Disk length
              This register contains the length (number of words) of
              Disk DMA data. It also contains 2 control bits. These
              are a DMA enable bit, and a DMA direction(read/write) bit.
              BIT#
              ----  -----------   --------------------------------
              15    DMAEN         Disk DMA Enable
              14    WRITE         Disk Write(RAM to Disk ) if 1
              13-0  LENGTH        Length (# of words) of DMA data.

DSKDAT    026  W       P   Disk DMA Data write
DSKDATR   008  ER      P   Disk DMA Data read (early read dummy address)
              This register is the Disk-DMA data buffer. It contains 2
              bytes of data that are either sent to (write) or received
              from (read) the disk. The write mode is enabled by bit
              14 of the LENGTH register. The DMA controller automatically
              transfers data to or from this register and RAM, and when
              the DMA data is finished (Length=0) it causes a Disk
              Block Interrupt. See interrupts below.

DSKBYTR   01A  R       P   Disk Data byte and status read
              This register is the Disk-Microprocessor data buffer.
              Data from the disk (in read mode) is loaded into this
              register one byte at a time, and bit 15 (DSKBYT) is set true.
              BIT#
              15    DSKBYT    Disk byte ready (reset on read)
              14    DMAON     DMAEN(DSKLEN) & DMAEN(DMACON) & DSKEN(DMACON)
              13    DISKWRITE Mirror of bit 14 (WRITE) in DSKLEN
              12    WORDEQUAL This bit true only while DSKSYNC register
                              equals the data from disk
              11-08  0          Not used
              07-00 DATA        Disk byte data

DSKSYNC   07E  W       P   Disk sync register, holds the match code for disk
                             read synchronization. See ADKCON bit 10

FMODE   P 076  W           Memory Fetch Mode
            This register controls the fetch mechanism for different
            types of Chip Ram accesses:

       Bit#    Function     Description
       ---------------------------------------------------------------
       15-04     -          unused
```

```
        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 (as before)      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

        SPAGEM SPR32  Sprite Fetch Increment       Memory Cycle  Bus Width
        ------------------------------------------------------------------
          0     0     by 2 bytes (as before)      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

HBSTOP  H 1C6  W   A       Horiz line position for HBLANK stop
HBSTRT  H 1C4  W   A       Horiz line position for HBLANK strt
                 These are the start,stop positions (in 280ns increments)
                 for the HBLANK that comes out on the CSY* pin when
                 BLANKEN bit in BEAMCON0 is 1. It also affects VARCSY--
                 see BEAMCON0.
HCENTER H 1E2  W   A       Horizontal position (CCKs) of VSYNC on long field
                 This is necessary for interlace mode with variable beam
                 counters. See BEAMCON0 for when it affects chip outputs.
                 See HTOTAL for bits.
HHPOSR  H 1DA  R   A       DUAL mode hires H beam counter read
HHPOSW  H 1D8  W   A       DUAL mode hires H beam counter write
                 This is the secondary beam counter for
                 the faster mode, triggering the UHRES pointers & doing the
                 comparisons for HBSTRT,STOP,HTOTAL,HSSTRT,HSSTOP
                 (See HTOTAL for bits)

HSSTOP  H 1C2  W   A       Horiz line position for HSYNC stop
                 Sets # of color clocks for sync stop (HTOTAL for bits)
HSSTRT  H 1DE  W   A       Horiz line position for HSYNC stop
                 Sets # of color clocks for sync start (HTOTAL for bits)
                 See BEAMCON0 for details of when these 2 are active.
HTOTAL  H 1C0  W   A       Highest color clock count in horiz line
                 Bit# 15 14 13 12 11 10 09 08 07 06 05 04 03 02 10 00
                       x  x  x  x  x  x  x  x h8 h7 h6 h5 h4 h3 h2 h1
                 (x's should be driven to 0 for upward compatibility)
                 Horiz line has this many + 1  280ns increments. If the
                 PAL bit & LOLDIS are not high, long line/short line toggle will
                 occur, and there will be this many + 2 every other line.
                 Active if VARBEAMEN=1 or DUAL=1.

INTREQ    09C  W       P   Interrupt Request bits (clear or set)
INTREQR   01E  R       P   Interrupt request bits (read)
                 This register contains interrupt request bits(or flags).
                 These bits may be polled by the processor, and if enabled
                 by the bits listed in the next register, they may cause
                 processor interrupts. Both a set and clear operation
                 are required to load arbitary data into this register.
                 The bit assignments are identical to the Enable register below.

INTENA    09A  W       P   Interrupt Enable bits (clear or set  bits)
INTENAR   01C  R       P   Interrupt Enable bits Read
```

This register contains interrupt enable bits. The bit assignment for both the request, and enable registers is given below.

BIT#	FUNCT	LEVEL	DESCRIPTION
15	SET/CLR		Set/Clear control bit. Determines if bits written with a 1 get set or cleared. Bits written with a zero are always unchanged.
14	INTEN		Master interrupt (enable only , no request)
13	EXTER	6	External interrupt
12	DSKSYN	5	Disk Sync register(DSKSYNC)matches Disk data
11	RBF	5	Serial port Receive Buffer Full
10	AUD3	4	Audio channel 3 block finished
09	AUD2	4	Audio channel 2 block finished
08	AUD1	4	Audio channel 1 block finished
07	AUD0	4	Audio channel 0 block finished
06	BLIT	3	Blitter finished
05	VERTB	3	Start of Vertical blank
04	COPER	3	Coprocessor
03	PORTS	2	I/O Ports and timers
02	SOFT	1	Reserved for software initiated interrupt.
01	DSKBLK	1	Disk Block finished
00	TBE	1	Serial port Transmit Buffer Empty

JOY0DAT	00A	R	D	Joystick-mouse 0 data (left vert,horiz)
JOY1DAT	00C	R	D	Joystick-mouse 1 data (right vert,horiz)

These addresses each read a 16 bit register. These in turn are loaded from the MDAT serial stream and are clocked in on the rising edge of SCLK. MLD* output is used to parallel load the external parallel-to-serial converter. This in turn is loaded with the 4 quadrature inputs from each of two game controller ports(8 total) plus 8 miscellaneous control bits which are new for LISA and can be read in upper 8 bits of LISAID. Register bits are as follows:

Mouse counter usage(pins 1,3 =Yclock, pins 2,4 =Xclock)

BIT#	15,14,13,12,11,10,09,08	07,06,05,04,03,02,01,00
JOY0DAT	Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0	X7 X6 X5 X4 X3 X2 X1 X0
JOY1DAT	Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0	X7 X6 X5 X4 X3 X2 X1 X0

0=left controller pair, 1=right controller pair.
(4 counters total). The bit usage for both left and right addresses is shown below. Each 6 bit counter(Y7-Y2,X7-X2) is clocked by 2 of the signals input from the mouse serial stream. Starting with first bit received:

Serial bit	Name	Description
0	M0H	JOY0DAT Horizontal Clock
1	M0HQ	JOY0DAT Horizontal Clock(quadrature)
2	M0V	JOY0DAT Vertical Clock
3	M0VQ	JOY0DAT Vertical Clock(quadrature)
4	M1H	JOY1DAT Horizontal Clock
5	M1HQ	JOY1DAT Horizontal Clock(quadrature)
6	M1V	JOY1DAT Vertical Clock
7	M1VQ	JOY1DAT Vertical Clock(quadrature)

Bits 1 and 0 of each counter (Y1-Y0,X1-X0) may be read to determine the state of the related input signalpair. This allows these pins to double as joystick switch inputs. Joystick switch closures can be deciphered as follows:

Directions	Pin#	counter bits
Forward	1	Y1 xor Y0 (BIT#09 xor BIT#08)
Left	3	Y1
Back	2	X1 xor X0 (BIT#01 xor BIT#00)
Right	4	X1

JOYTEST 036 W D Write to all 4 Joystick-mouse counters at once.
Mouse counter write test data:

BIT#	15,14,13,12,11,10,09,08	07,06,05,04,03,02,01,00
JOY0DAT	Y7 Y6 Y5 Y4 Y3 Y2 xx xx	X7 X6 X5 X4 X3 X2 xx xx
JOY1DAT	Y7 Y6 Y5 Y4 Y3 Y2 xx xx	X7 X6 X5 X4 X3 X2 xx xx

LISAID H 07C R D Denise/Lisa (video out chip) revision level
The original Denise (8362) does not have this register, so whatever value is left over on the bus from the last cycle will be there. ECS DENISE(8373) returns hex(FC) in the lower 8 bits. LISA returns hex (F8). The upper 8 bits of this register are loaded from the serial mouse bus, and are reserved for future hardware implentation.

POT0DAT h 012 R P Pot counter data left pair (vert,horiz)
POT1DAT h 014 R P Pot counter data right pair (vert,horiz)
These addresses each read a pair of 8 bit pot counters. (4 counters total). The bit assignment for both addresses is shown below. The counters are stopped by signals from 2 controller connectors(left-right) with 2 pins each.

BIT#	15,14,13,12,11,10,09,08	07,06,05,04,03,02,01,00
RIGHT	Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0	X7 X6 X5 X4 X3 X2 X1 X0
LEFT	Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0	X7 X6 X5 X4 X3 X2 X1 X0

	CONNECTORS			PAULA
loc.	dir.	sym	pin	pin
RIGHT	Y	RY	9	33
RIGHT	X	RX	5	32
LEFT	Y	LY	9	36
LEFT	X	LX	5	35

With normal (NTSC or PAL) horiz. line rate, the pots will give a full scale (FF) reading with about 500kohms in one frame time. With proportionally faster horiz line times, the counters will count proportionally faster.
This should be noted when doing variable beam displays.

POTGO 034 W P Pot Port (4 bit) Direction and Data, and Pot Counter start.

POTINP 016 R P Pot pin data read
This register controls a 4 bit bi-directional I/O port that shares the same 4 pins as the 4 pot counters above.

BIT#	FUNCT	DESCRIPTION
15	OUTRY	Output enable for Paula pin 33
14	DATRY	I/O data Paula pin 33
13	OUTRX	Output enable for Paula pin 32
12	DATRX	I/O data Paula pin 32
11	OUTLY	Output enable for Paula pin 36
10	DATLY	I/O data Paula pin 36

```
                09    OUTLX    Output enable for Paula pin 35
                08    DATLX    I/O data Paula pin 35
                07-01   X      not used
                00    START     Start pots (dump capacitors,start counters)

REFPTR    028  W   A       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 microprocesor.

SERDAT    030  W       P   Serial Port Data and stop bits write
                This address writes data to a Transmit data buffer.
                Data from this buffer is moved into a serial shift register
                for output transmission, whenever it is empty. This sets the
                Interrupt Request TBE (transmit buffer empty). A stop bit
                must be provided as part of the data word. The length
                of the data word is set by the position of the stop bit.
                BIT#  15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
                USE    0  0  0  0  0  0  S D8 D7 D6 D5 D4 D3 D2 D1 D0
                note: S= stop bit =1 ,  D= data bits

SERDATR   018  R       P   Serial Port Data and Status read
                This address reads data from a Receive data buffer.
                Data in this buffer is loaded from a receiving  shift register
                whenever it is full. Several interrupt request bits are also
                read at this address, along with the data,as shown below.
                BIT#
                15    OVRUN        Serial port receiver overun
                14    RBF          Serial port Receive Buffer Full(mirror)
                13    TBE          Serial port Transmit Buffer Empty(mirror)
                12    TSRE         Serial port Transmit shift reg. empty
                11    RXD          RXD pin receives UART serial data for direct
                                   bit test by the micro
                10     x           Not used
                09    STP          Stop bit
                08  STP-DB8        Stop bit if LONG, Data bit if not.
                07    DB7          Data bit
                06    DB6          Data bit
                05    DB5          Data bit
                04    DB4          Data bit
                03    DB3          Data bit
                02    DB2          Data bit
                01    DB1          Data bit
                00    DB0          Data bit

SERPER    032  W       P   Serial Port Period and control
                This register contains the control bit LONG reffered to
                above, and a 15 bit number defining the serial port
                Baud Rate. If this number is N, then the Baud Rate is
                1 bit every (N+1)*.2794  Microseconds.
                BIT#
                15     LONG       Defines Serial Receive as 9 bit word.
                14-00  RATE       Defines Baud Rate=1/((N+1)*.2794 microsec)

SPRHDAT H 078  W ext logic UHRES sprite identifier & data
                This identifies the cycle when this pointer address is on the
                bus accessing the memory.

SPRHPTH H 1E8  W   A       UHRES sprite pointer (High 5 bits)
SPRHPTL H 1EA  W   A       UHRES sprite pointer (Low 15 bits)
                This pointer is activated in the 1st & 3rd 'free' cycles
```

(see BPLHPTH,L) after horiz line start. It increments
for the next line.

SPRHSTOP H 1D2 W A UHRES sprite vertical display stop

```
Bit#     15   14 13 12 11  10 09 08 07 06 05 04 03 02 01 00
      SPRHWRM  x  x  x  x v10 v9 v8 v7 v6 v5 v4 v3 v2 v1 v0
```

SPRHWRM= Swaps the polarity of ARW* when the SPRHDAT comes
out so that external devices can detect the RGA and put
things into memory. (ECS and later chips only)

SPRHSTRT H 1D0 W A UHRES sprite vertical display start

```
Bit# 15 14 13 12 11  10 09 08 07 06 05 04 03 02 01 00
      x  x  x  x  x v10 v9 v8 v7 v6 v5 v4 v3 v2 v1 v0
```

SPRxPTH 120 W A Sprite x pointer (High 5 bits)
SPRxPTL 122 W A Sprite x pointer (Low 15 bits)
This pair of registers contains the 20 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 coprocessor
every vertical blank time.

SPRxPOS 140 W A D Sprite x Vert-Horiz start position data

BIT#	SYM	FUNCTION
15-08	SV7-SV0	Start vertical value. High bit(SV8) is in SPRxCTL reg below.
07-00	SH10-SH3	Start horizontal value. Low order bits are in SPRxCTL reg. below.

SPRxCTL p 142 W A D Sprite x position and control data

BIT#	SYM	FUNCTION
15-08	EV7-EV0	End (stop) vert. value. low 8 bits
07	ATT	Sprite attach control bit(odd sprites only)
06	SV9	start vert. value 10th bit
05	EV9	end (stop) vert. value 10th bit
04	SH1=0	start horiz. value, 70nS increment
03	SH0=0	start horiz. value, 35nS increment
02	SV8	Start vert. value 9th bit
01	EV8	End (stop) vert. value 9th bit
00	SH2	start horiz. value, 140nS increment

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.
Writing to SPRxCTL disables the corresponding sprite.

SPRxDATA 144 W D Sprite x image data register A
SPRxDATB 146 W D Sprite x image data register B

These registers buffer the Sprite image data. They are
usually loaded by the Sprite DMA channel but may be
loaded by either processor at any time. When a horizontal
coincidence occurs the buffers are dumped into shift registers
and serially outputed to the display, MSB first on the left.
Note: Writing to the A buffer enables (arms) the sprite.

```
                Writing to the SPRxCTL register disables the sprite.
                If enabled, data in the A and B buffers will be output
                whenever the beam counter equals the sprite horizontal
                position value in the SPRxPOS register. In lowres mode,
                1 sprite pixel is 1 bitplane pixel wide. In HRES and
                SHRES mode, 1 sprite pixel is 2 bitplane pixels.
                The DATB bits are the 2SBs (worth 2) for the color registers,
                and MSB for SHRES. DATA bits are LSBs of the pixels.

STREQU    038  S     D     Strobe for horiz sync with VB(vert. blank) and EQU
STRVBL    03A  S     D     Strobe for horiz sync with VB
STRHOR    03C  S     D P   Strobe for horiz sync
STRLONG h 03E  S     D     Strobe for identification of long horz. line(228 CC)
                One of the first 3 strobe addresses above, it is placed on the
                RGA bus during the first refresh time slot.
                The 4th strobe shown above is used during the second
                refresh time slot of every other line, to identify
                lines with long counts(228- NTSC, HTOTAL+2- VARBEAMEN=1
                hires chips only).  There are 4 refresh time slots
                and any not used for strobes will leave a null (1FE)
                address on the RGA bus.

VBSTOP  H 1CE  W   A       Vertical line for VBLANK stop
VBSTRT  H 1CC  W   A       Vertical line for VBLANK strt
                (V10-0 <- D10-0) Affects CSY* pin if BLANKEN=1 &
                VSY* Pin if CSCBEN=1 (see BEAMCON0)

VPOSR   p 004  R   A       Read Vert most sig. bits (and frame flop)
VPOSW     02A  W   A       Write Vert most sig. bits (and frame flop)
                BIT#  15,14,13,12,11,10,09,08, 07,06,05,04,03,02,01,00
                USE  LOF I6 I5 I4 I3 I2 I1 I0,LOL -- -- -- --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-pal)  = 00
                  8372(fat-hr) (agnushr),thru rev. 4      = 20 PAL,30 NTSC
                8372(fat-hr) (agnushr),rev. 5           = 21 PAL,31 NTSC
                8374(alice)                         = 22 PAL,32 NTSC
                LOL=Long line bit. When low, it indicates short raster line.
                v9,v10-- hires chips only (20,30 identifiers)

VHPOSR    006  R   A       Read Vert and Horiz Position of beam, or lightpen
VHPOSW    02C  W   A       Write Vert and Horiz 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 V2 V1 V0,H8 H7 H6 H5 H4 H3 H2 H1
                RESOLUTION=1/160 OF SCREEN WIDTH (280 NS)
VSSTOP  H 1CA  W   A       Vert. position for VSYNC stop.
                See BEAMCON0 for more details. (V10-0 <- D10-0)
VSSTRT  H 1E0  W   A       Vert. position for VSYNC start
VTOTAL  H 1C8  W   A       Highest numbered vertical line (VARBEAMEN=1)
                It's the line number to reset the counter,
                so there's this many + 1 in a field. The exception is
                if the LACE bit is set (BPLCON0), in which case every
                other field is this many +2 & the short field is this
                many + 1.
```