

**AGA**

<b>COLLABORATORS</b>
----------------------

	TITLE : AGA		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AGA</b>	<b>1</b>
1.1	Pandora Chipset Documentation	1
1.2	summary	1
1.3	explanation	2
1.4	registersindex	6
1.5	Some Notes to Start Width	11
1.6	ADKCON	12
1.7	AUDxLCH	13
1.8	AUDxLCL	13
1.9	AUDxLEN	13
1.10	AUDxPER	13
1.11	AUDxVOL	14
1.12	AUDxDAT	14
1.13	BEAMCON0	14
1.14	BLTxPTH	16
1.15	BLTxPTL	16
1.16	BLTxMOD	16
1.17	BLTAFWM	17
1.18	BLTALWM	17
1.19	BLTxDAT	17
1.20	BLTDDAT	17
1.21	BLTSIZE	18
1.22	BLTCON0	18
1.23	BLTSIZH	19
1.24	BLTSIZV	19
1.25	BPLHDAT	20
1.26	BPLHMOD	20
1.27	BPLHPTH	20
1.28	BPLHPTL	20
1.29	bplhstop	21

---

1.30 BPLHSTRT . . . . .	21
1.31 BPLxPTH . . . . .	21
1.32 BPLxPTL . . . . .	21
1.33 BPLxDAT . . . . .	22
1.34 BPLxMOD . . . . .	22
1.35 BPLCON0 . . . . .	22
1.36 BPLCON1 . . . . .	23
1.37 BPLCON2 . . . . .	24
1.38 BPLTCON3 . . . . .	24
1.39 BPLCON4 . . . . .	26
1.40 CLXCON . . . . .	26
1.41 CLXCON2 . . . . .	27
1.42 CLXDAT . . . . .	27
1.43 COLORx . . . . .	28
1.44 COPCON . . . . .	29
1.45 COPJMP1 . . . . .	29
1.46 COPJMP2 . . . . .	29
1.47 COP1LCH . . . . .	29
1.48 COPINS . . . . .	30
1.49 DDFSTRT . . . . .	31
1.50 DIWSTRT . . . . .	32
1.51 DIWHIGH . . . . .	33
1.52 DMACON . . . . .	33
1.53 dskpth . . . . .	34
1.54 DSKLEN . . . . .	34
1.55 DSKDAT . . . . .	34
1.56 DSKDATR . . . . .	35
1.57 DSKBYTR . . . . .	35
1.58 DSKSYNC . . . . .	35
1.59 FMODE . . . . .	35
1.60 HBSTOP . . . . .	36
1.61 HCENTER . . . . .	37
1.62 HHPOSR . . . . .	37
1.63 HSSTOP . . . . .	37
1.64 HSSTRT . . . . .	37
1.65 HTOTAL . . . . .	38
1.66 INTREQ . . . . .	38
1.67 INTENA . . . . .	38
1.68 JOYxDAT . . . . .	39

---

---

1.69 JOYTEST . . . . .	40
1.70 LISPID . . . . .	40
1.71 POTxDAT . . . . .	41
1.72 POTGO . . . . .	41
1.73 POTINP . . . . .	41
1.74 REFPTR . . . . .	42
1.75 SERDAT . . . . .	42
1.76 SERDATR . . . . .	42
1.77 SERPER . . . . .	43
1.78 SPRHDAT . . . . .	43
1.79 SPRHPTH . . . . .	44
1.80 SPRHSTOP . . . . .	44
1.81 SPRHSTRT . . . . .	44
1.82 SPRxPTH . . . . .	44
1.83 sprxpos . . . . .	45
1.84 sprxctl . . . . .	45
1.85 SPRxDAT . . . . .	45
1.86 STREQU . . . . .	46
1.87 vbstop . . . . .	46
1.88 VPOSR . . . . .	46
1.89 VHPOSR . . . . .	47
1.90 VSSTOP . . . . .	47
1.91 lisamodes . . . . .	47

---

# Chapter 1

## AGA

### 1.1 Pandora Chipset Documentation

(C) 1993 by TFA

-- - --+- -- - --+\*>> ArTiSt'S WiTh An AtTiTuDe <<\*+- - -- --+-- -

Specification for the  
Advanced Amiga (AA) Chip Set

TYPED BY FIREFLASH/C18, SPONGE/C18  
AMIGAGUIDE VERSION & PAGE 19 BY SCHWARZENEGGER/TFA

Please select any of the topics listed below and follow up on the links as they appear.

1. Summary of new features for AA
2. Explanation of new features
3. List of Registers ordered by address
4. List of Registers ordered alphabetically
5. New LISA Display & Sprite Modes

### 1.2 summary

1. Summary of new features for AA
- 

32 bit wide data bus supports input of 32-bit wide bitplane data and allows doubling of memory bandwidth. Additional doubling of bandwidth can be achieved by using FAST page mode Ram. The same bandwidth enhancements are available for sprites. Also the maximum number of bitplanes useable in all modes was increased to eight (8).

The color Palette has been expanded to 256 colors deep and 25 bits wide (8 RED, 8 GREEN, 8 BLUE, 1 GENLOCK). This permits display of 256 simultaneous colors in all resolutions. A palette of 16,777,216 colors are available in all resolutions.

---

28Mhz clock input allows for cleaner definition of HIRES and SHRES pixels  
 ALICE'S clock generator is synchronized by means of LISA's 14MHz and SCLK  
 outputs, Genlock XCLK and XCLKEN pins have been eliminated (external MUX is  
 now required).

A new register bit allows sprites to appear in the screen border regions  
 (BRDRSPRT - See BPLCON3).

A bitplane mask field of 8 bits allows an address offset into the color  
 palette.

Two 4-bit mask fields do the same for odd and even sprites.

In Dual Playfield modes, 2 4-bitplane playfields are now possible in all  
 resolutions.

Two Extra high-order playfield scrollbits allow seamless scrolling of up to  
 64 bit wide bitplanes in all resolutions. Resolution of bitplane scroll,  
 display window, and horizontal sprite position has been improved to 35ns in  
 all resolutions.

A new 8-bitplane HAM mode has been created, 6 for colors and 2 for control  
 bits. All HAM modes are available in all resolutions (not just LORES as  
 before).

A RST\_input pin has been added, which resets all the bits contained in  
 registers that were new for ECS or LISA:  
 BPLCON3, BPLCON4, CLXCON2, DIWHIGH, FMODE.

Sprite resolution can be set to LORES, HIRES, SHRES, independant of bitplane  
 resolution.

Attached Sprites are now available in all resolutions.

Hardware Scan Doubling support has been added for bitplanes and sprites.  
 This is intended to allow 15KHz screens to be intelligently displayed on a  
 31KHz monitor and share the display with 31KHz screens.

## 1.3 explanation

### 2. Explanation of new features

#### Bitplanes

There are now 8 bitplanes instead of 6. In single playfield modes they can  
 address 256 colors instead of 64. As long as the memory architecture can  
 support the bandwidth, all 8 bitplanes are available in all 3 resolutions  
 In the same vein, 4+4 bitplane dualplayfield is available in all 3  
 resolutions, unless bitplane scan-doubling is enabled, in which case  
 both playfields share the same bitplane modulus register. Bits 15 thru 8 of  
 BPLCON4 comprise an 8 bit mask for the 8 bitplane address, XOR'ing the  
 individual bits. This allows the copper to exchange color maps with a  
 single instruction.

BPLCON1 now contains an 8 bit scroll value for each of the playfields. Granularity of scroll now extends down to 35nSec.(1 SHRES pixel), and scroll can delay playfield thru 32 bus cycles. Bits BPAGEM and BPL32 in new register FMODE control size of bitplane data in BPL1DAT thru BPL8DAT.

The old 6 bitplane HAM mode, unlike before, works in HIRES and SHRES resolutions.

As before bitplanes 5 and 6 control it's function as follows:

BP6	BP5	RED	GREEN	BLUE	
0	0	select new base register (1 of 16)			
0	1	hold	hold	modify	
1	0	modify	hold	hold	
1	1	hold	modify	hold	

There is a new 8 bitplane HAM (Hold and Modify) mode. This mode is invoked when BPU field in BPLCON0 is set to 8 , and HAMEN is set. Bitplanes 1 and 2 are used as control bits analagous to the function of bitplanes 5 and 6 in 6 bitplane HAM mode:

BP2	BP1	RED	GREEN	BLUE	
0	1	select new base register (1 of 64)			
0	1	hold	hold	modify	
1	0	modify	hold	hold	
1	1	hold	modify	hold	

Since only 6 bitplanes are available for modify data, the data is placed in 6 MSB. The 2 LSB are left unmodified, which allows creation of all 16,777,216 colors simultaneously, assuming one had a large enough screen and picked one's base registers judiciously. This HAM mode also works in HIRES and SHRES modes.

For compatibility reasons EHB mode remains intact. Its existence is rather moot in that we have more than enough colors in the color table to replace its functionality. As before, EHB is invoked whenever SHRES = HIRES = HAMEN= DPF = 0 and BPU = 6. Please note that starting with ECS DENISE there is a bit in BPLCON2 which disables this mode (KILLEHB).

Bits PF2OF2,1,0 in BPLCON3 determine second playfield's offset into the color table. This is now necessary since playfields in DPF mode can have up to 4 bitplanes. Offset value are as defined in register map.

BSCAN2 bit in FMODE enables bitplane scan-doubling. When V0 bit of DIWSTRT matches V0 of vertical beam counter, BPL1MOD contains the modulus for the

display line, else BPL2MOD is used. When scan-doubled both odd and even bitplanes use the same modulus on a given line, whereas in normal mode odd bitplanes used BPL1MOD and even bitplanes used BPL2MOD. As a result Dual Playfields screens will probably not display correctly when scan-doubled.

### Sprites

-----

Bits SPAGEM and SPR32 in FMODE whether size of sprite load data in SPR0DATA(B) thru SPR7DATA(B) is 16,32, or 64 bits, analagous to bitplanes. BPLCON3 contains several bits relating to sprite behavior. SPRES1 and SPRES0 control sprite resolution, whether they conform to the ECS standard or override to LORES,HIRES,or SHRES. BRDRSPRT, when high,allows sprites to be visible in border areas. ESPRM7 thru ESPRM4 allow relocation of the even sprite color map. OSPRM7 thru OSPRM4 allow relocation of the odd sprite color map. In the case of attached sprites OSPRM bits are used.

SSCAN2 bit in FMODE enables sprite scan-doubling. When enabled, individual SH10 bits in SPRxPOS registers control whether or not a given sprite is to be scan-doubled. When V0 bit of SPRxPOS register matches V0 bit of vertical beam counter, the given sprite's DMA is allowed to proceed as before. If the don't match, then sprite DMA is disabled and LISA reuses the sprite data from the previous line. When sprites are scan-doubled, only the position and control registers need be modified by the programmer; the data registers need no modification.

NOTE: Sprite vertical start and stop positions must be of the same parity, i.e. both odd or even.

### Compatibility

-----

RST\_pin resets all bits in all registers new to AA. These registers include: BPLCON3, BPLCON4, CLXCON2, DIWHIGH, FMODE.

ECSENA bit (formerly ENBPLCN3) is used to disable those register bits in BPLCON3 that are never accessed by old copper lists, and in addition are required by old style copper lists to be in their default settings. Specifically ECSENA forces the following bits to their default low settings: BRDRBLNK, BRDNTRAN, ZDCLKEN, EXTBLKEN, and BRDRSPRT.

CLXCON2 is reset by a write to CLXCON, so that old game programs will be able to correctly detect collisions.

DIWHIGH is reset by writes to DIWSTRT or DIWSTOP. This is interlock is inherited from ECS Denise.

### Genlock

-----

Lots of new genlock features were added to ECS DENISE and are carried over to LISA. ZDBPEN in BPLCON2 allows any bitplane, selected by ZDBPSEL2,1,0, to be used as a transparency mask (ZD pin mirrors contents of selected bitplane). ZDCTEN disables the old COLOR00 is transparent mode, and allows the bit31 position of each color in the color table to control transparency. ZDCLKEN generates a 14MHz clock synchronized with the video

data that can be used by video post-processors. Finally, BRDNTRAN in BPLCON3 generates an opaque border region which can be used to frame live video.

#### Color Lookup Table

The color table has grown from 32 13-bit registers to 256 25-bit registers. Several new register bits have been added to BPLCON3 to facilitate loading the table with only 32 register addresses. LOCT, selects either the 16 MSB or LSB for loading. Loading the MSB always loads the LSB as well for compatibility, so when 24 bit colors are desired load LSB after MSB. BANK2,1,0 of 8 32 address banks for loading as follows:

BANK2	BANK1	BANK0	COLOR ADDRESS RANGE
0	0	0	COLOR00 - COLOR1F
0	0	1	COLOR20 - COLOR3F
0	1	0	COLOR40 - COLOR5F
0	1	1	COLOR60 - COLOR7F
1	0	0	COLOR80 - COLOR9F
1	0	1	COLORA0 - COLORBF
1	1	0	COLORC0 - COLORDF
1	1	1	COLORE0 - COLORFF

RDRAM bit in BPLCON2 causes LISA to interpret all color table accesses as reads.

Note: There is no longer any need to "scramble" SHRES color table entries. This artifice is no longer required and people who bypass ECS graphics library calls to do their own 28MHz graphics are to be pointed at and publicly humiliated.

#### Collision

A new register CLXCON2 contains 4 new bits. ENBP7 and ENBP6 are the enable bits for bitplanes 7 and 8, respectively. Similarly, MVBP7 and MPBP8 are their match value bits. CLXDAT is unchanged.

#### Horizontal Comparators

All programmable comparators with the exception of VHPOSW have 35nSec resolution.: DIWHIGH, HBSTOP, SPRCTL, BPLCON1. BPLCON1 has additional high-order bits as well. Note that horizontal bit position representing 140nSec resolution has been changed to 3rd least significant bit, where before it used to be a field's LSB, For example, bit 00 in BPLCON1 used to be named PF1H0 and now it's called PF1H2.

#### Coercion of 15KHz to 31KHz:

We have added new hardware features to LISA to aid in properly displaying 15KHz and 31KHz viewports together on the same 31KHz display. LISA can globally set sprite resolution to LORES, HIRES, or SHRES.

LISA will ignore SH10 compare bits in SPRxPOS when scan-doubling, thereby allowing ALICE to use these bits individually set scan-doubling.

## 1.4 registersindex

### 3. List of registers ordered by address

-----  
 Symbols Used:

& = Register used by DMA channel only.

% = Register used by DMA channel usually, processors sometimes.

+ = Address register pair. Low word uses DB1-DB15, High word DB0-DB4.

~ = Address not writable by the coprocessor unless COPCON bit 1 is set true

h = new for HiRes chip set.

p = new for IAA chip set.

A = Agnus/Alice chip set.

D = Denise/Lisa chip set.

P = Paula chip.

W = Write.

R = Read.

ER= Early read. This is a DMA transfer to RAM, from either the disk or from the blitter. Ram timing requires data to be on the bus earlier than microprocessor read cycles. These transfers are therefore initiated by Agnus timing, rather than a read address on the register address bus (RGA).

S = Strobe (Write address with no register bits).

PTL,PTH = 20 bit pointer that addresses DMA data. Must be reloaded by a processor before use (Vertical blank for bit plane and sprite pointers. and prior to starting the blitter for blitter pointers). (old chips - 18 bits).

LCL,LCH = 20 bit location (starting address) of DMAdata. Used to automatically restart pointers. such as the Coprocessor program counter (during vertical blank), and the audio sample counter (whenever the audio length count is finished), (Old chips - 18 bits).

MOD = 15 bit Modulo. A number that is automatically added to the memory address at the end of each line to generate the address for the beginning of the next line. This allows the blitter (or the display window) to operate on (or display) a window of data that is smaller than the actual picture in memory. (memory map) Uses 15 bits, plus sign extended.

NAME	ADDR	R/W	CHIP(s)	FUNCTION
BLTDDAT	& ~000	ER	A	Blitter dest. early read (dummy address)
DMACONR	~002	R	A P	Dma control (and blitter status) read
VPOSR	~004	R	A	Read vert most sig. bits (and frame flop
VHPOSR	~006	R	A	Read vert and horiz position of beam
DSKDATR	& ~008	ER	P	Disk data early read (dummy address)
JOY0DAT	~00A	R	D	Joystick-mouse 0 data (vert,horiz)
JOT1DAT	~00C	R	D	Joystick-mouse 1 data (vert,horiz)
CLXDAT	~00E	R	D	Collision data reg. (read and clear)
ADKCONR	~010	R	P	Audio,disk control register read
POT0DAT	~012	R	P	Pot counter pair 0 data (vert,horiz)

POT1DAT	~014	R		P	Pot counter pair 1 data (vert,horiz)
POTINP	~016	R		P	Pot pin data read
SERDATR	~018	R		P	Serial port data and status read
DSKBYTR	~01A	R		P	Disk data byte and status read
INTENAR	~01C	R		P	Interrupt enable bits read
INTREQR	~01E	R		P	Interrupt request bits read
DSKPTH	+ ~020	W	A		Disk pointer (high 5 bits)
DSKPTL	+ ~022	W	A		Disk pointer (low 15 bits)
DSKLEN	~024	W		P	Disk lentgh
DSKDAT	& ~026	W		P	Disk DMA data write
REFPTR	& ~028	W	A		Refresh pointer
VPOSW	~02A	W	A		Write vert most sig. bits(and frame flop)
VHPOSW	~02C	W	A	D	Write vert and horiz pos of beam
COPCON	~-2E	W	A		Coprocessor control reg (CDANG)
SERDAT	~030	W		P	Serial port data and stop bits write
SERPER	~032	W		P	Serial port period and control
POTGO	~034	W		P	Pot count start,pot pin drive enable data
JOYTEST	~036	W		D	Write to all 4 joystick-mouse counters at once
STREQU	& ~038	S		D	Strobe for horiz sync with VB and EQU
STRVBL	& ~03A	S		D	Strobe for horiz sync with VB (vert blank)
STRHOR	& ~03C	S		D P	Strobe for horiz sync
STRLONG	& ~03E	S		D	Strobe for identification of long horiz line
BLTCON0	~040	W	A		Blitter control reg 0
BLTCON1	~042	W	A		Blitter control reg 1
BLTAFWM	~044	W	A		Blitter first word mask for source A
BLTALWM	~046	W	A		Blitter last word mask for source A
BLTCPTH	+ ~048	W	A		Blitter pointer to source C (high 5 bits)
BLTCPTL	+ ~04A	W	A		Blitter pointer to source C (low 15 bits)
BLTBPTH	+ ~04C	W	A		Blitter pointer to source B (high 5 bits)
BLTBPTL	+ ~04E	W	A		Blitter pointer to source B (low 15 bits)
BLTAPTH	+ ~050	W	A		Blitter pointer to source A (high 5 bits)
BLTAPTL	+ ~052	W	A		Blitter pointer to source A (low 15 bits)
BPTDPTH	+ ~054	W	A		Blitter pointer to destn D (high 5 bits)
BLTDPTL	+ ~056	W	A		Blitter pointer to destn D (low 15 bits)
BLTSIZE	~058	W	A		Blitter start and size (win/width,height)
BLTCON0L	h ~05A	W	A		Blitter control 0 lower 8 bits (minterms)
BLTSIZV	h ~05C	W	A		Blitter V size (for 15 bit vert size)
BLTSIZH	h ~05E	W	A		Blitter H size & start (for 11 bit H size)
BLTCMOD	~060	W	A		Blitter modulo for source C
BLTBMOD	~062	W	A		Blitter modulo for source B
BLTAMOD	~064	W	A		Blitter modulo for source A
BLTDMOD	~066	W	A		Blitter modulo for destn D
	~068				
	~06a				
	~06c				
	~06e				
BLTCDAT	& ~070	W	A		Blitter source C data reg
BLTBDAT	& ~072	W	A		Blitter source B data reg
BLTADAT	& ~074	W	A		Blitter source A data reg
	~076				
SPRHDAT	&h 078	W	A		Ext logic UHRES sprite pointer and data identifier
(BPLHDAT)	~07A	????			?????
LISAID	h ~07C	R		D	Chip revision level for Denise/Lisa
DSKSYNC	~07E	W		P	Disk sync pattern reg for disk read

COP1LCH	+	080	W	A		Coprocessor first location reg (high 5 bits)
COP1LCL	+	082	W	A		Coprocessor first location reg (low 15 bits)
COP2LCH	+	084	W	A		Coprocessor second reg (high 5 bits)
COP2LCL	+	086	W	A		Coprocessor second reg (low 15 bits)
COPJMP1		088	S	A		Coprocessor restart at first location
COPJMP2		08A	S	A		Coprocessor restart at second location
COPINS		08C	W	A		Coprocessor inst fetch identify
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)
DDFSTRT		092	W	A		Display bit plane data fetch start.hor pos
DDFSTOP		094	W	A		Display bit plane data fetch stop.hor pos
DMACON		096	W	A	P	DMA control write (clear or set)
CLXCON		098	W		D	Collision control
INTENA		09A	W		P	Interrupt enable bits (clear or set bits)
INTREQ		09C	W		P	Interrupt request bits (clear or set bits)
ADKCON		09E	W		P	Audio,disk,UART,control
AUD0LCH	+	0A0	W	A		Audio channel 0 location (high 5 bits)
AUD0LCL	+	0A2	W	A		Audio channel 0 location (low 15 bits)
AUD0LEN		0A4	W		P	Audio channel 0 lentgh
AUD0PER		0A6	W		P	Audio channel 0 period
AUD0VOL		0A8	W		P	Audio channel 0 volume
AUD0DAT	&	0AA	W		P	Audio channel 0 data
		0AC				
		0AE				
AUD1LCH	+	0B0	W	A		Audio channel 1 location (high 5 bits)
AUD1LCL	+	0B2	W	A		Audio channel 1 location (low 15 bits)
AUD1LEN		0B4	W		P	Audio channel 1 lentgh
AUD1PER		0B6	W		P	Audio channel 1 period
AUD1VOL		0B8	W		P	Audio channel 1 volume
AUD1DAT	&	0BA	W		P	Audio channel 1 data
		0BC				
		0BE				
AUD2LCH	+	0C0	W	A		Audio channel 2 location (high 5 bits)
AUD2LCL	+	0C2	W	A		Audio channel 2 location (low 15 bits)
AUD2LEN		0C4	W		P	Audio channel 2 lentgh
AUD2PER		0C6	W		P	Audio channel 2 period
AUD2VOL		0C8	W		P	Audio channel 2 volume
AUD2DAT	&	0CA	W		P	Audio channel 2 data
		0CC				
		0CE				
AUD3LCH	+	0D0	W	A		Audio channel 3 location (high 5 bits)
AUD3LCL	+	0D2	W	A		Audio channel 3 location (low 15 bits)
AUD3LEN		0D4	W		P	Audio channel 3 lentgh
AUD3PER		0D6	W		P	Audio channel 3 period
AUD3VOL		0D8	W		P	Audio channel 3 volume
AUD3DAT	&	0DA	W		P	Audio channel 3 data
		0DC				
		0DE				
BPL1PTH	+	0E0	W	A		Bit plane pointer 1 (high 5 bits)
BPL1PTL	+	0E2	W	A		Bit plane pointer 1 (low 15 bits)
BPL2PTH	+	0E4	W	A		Bit plane pointer 2 (high 5 bits)

BPL2PTL	+	0E6	W	A	Bit plane pointer 2 (low 15 bits)
BPL3PTH	+	0E8	W	A	Bit plane pointer 3 (high 5 bits)
BPL3PTL	+	0EA	W	A	Bit plane pointer 3 (low 15 bits)
BPL4PTH	+	0EC	W	A	Bit plane pointer 4 (high 5 bits)
BPL4PTL	+	0EE	W	A	Bit plane pointer 4 (low 15 bits)
BPL5PTH	+	0F0	W	A	Bit plane pointer 5 (high 5 bits)
BPL5PTL	+	0F2	W	A	Bit plane pointer 5 (low 15 bits)
BPL6PTH	+	0F4	W	A	Bit plane pointer 6 (high 5 bits)
BPL6PTL	+	0F6	W	A	Bit plane pointer 6 (low 15 bits)
BPL7PTH	+	0F8	W	A	Bit plane pointer 7 (high 5 bits)
BPL7PTL	+	0FA	W	A	Bit plane pointer 7 (low 15 bits)
BPL8PTH	+	0FC	W	A	Bit plane pointer 8 (high 5 bits)
BPL8PTL	+	0FE	W	A	Bit plane pointer 8 (low 15 bits)
BPLCON0		100	W	A D	Bit plane control reg (misc control bits)
BPLCON1		102	W	D	Bit plane control reg (scroll val PF1,PF2)
BPLCON2		104	W	D	Bit plane control reg (priority control)
BPLCON3		106	W	D	Bit plane control reg (enhanced features)
BPL1MOD		108	W	A	Bit plane modulo (odd planes, or active- fetch lines if bitplane scan-doubling is enabled)
BPL2MOD		10A	W	A	Bit plane modulo (even planes or inactive- fetch lines if bitplane scan-doubling is enabled)
BPLCON4	p	10C	W	D	Bit plane control reg (bitplane and sprite masks)
CLXCON2	p	10e	W	D	Extended collision control reg
BPL1DAT	&	110	W	D	Bit plane 1 data (parallel to serial con- vert)
BPL2DAT	&	112	W	D	Bit plane 2 data (parallel to serial con- vert)
BPL3DAT	&	114	W	D	Bit plane 3 data (parallel to serial con- vert)
BPL4DAT	&	116	W	D	Bit plane 4 data (parallel to serial con- vert)
BPL5DAT	&	118	W	D	Bit plane 5 data (parallel to serial con- vert)
BPL6DAT	&	11a	W	D	Bit plane 6 data (parallel to serial con- vert)
BPL7DAT	&p	11c	W	D	Bit plane 7 data (parallel to serial con- vert)
BPL8DAT	&p	11e	W	D	Bit plane 8 data (parallel to serial con- vert)
SPR0PTH	+	120	W	A	Sprite 0 pointer (high 5 bits)
SPR0PTL	+	122	W	A	Sprite 0 pointer (low 15 bits)
SPR1PTH	+	124	W	A	Sprite 1 pointer (high 5 bits)
SPR1PTL	+	126	W	A	Sprite 1 pointer (low 15 bits)
SPR2PTH	+	128	W	A	Sprite 2 pointer (high 5 bits)
SPR2PTL	+	12A	W	A	Sprite 2 pointer (low 15 bits)
SPR3PTH	+	12C	W	A	Sprite 3 pointer (high 5 bits)
SPR3PTL	+	12E	W	A	Sprite 3 pointer (low 15 bits)
SPR4PTH	+	130	W	A	Sprite 4 pointer (high 5 bits)
SPR4PTL	+	132	W	A	Sprite 4 pointer (low 15 bits)
SPR5PTH	+	134	W	A	Sprite 5 pointer (high 5 bits)
SPR5PTL	+	136	W	A	Sprite 5 pointer (low 15 bits)
SPR6PTH	+	138	W	A	Sprite 6 pointer (high 5 bits)
SPR6PTL	+	13A	W	A	Sprite 6 pointer (low 15 bits)
SPR7PTH	+	13C	W	A	Sprite 7 pointer (high 5 bits)

SPR7PTL	+	13E	W	A		Sprite 7 pointer (low 15 bits)
SPR0POS	%	140	W	A	D	Sprite 0 vert-horiz start pos data
SPR0CTL	%	142	W	A	D	Sprite 0 position and control data
SPR0DATA	%	144	W		D	Sprite 0 image data register A
SPR0DATB	%	146	W		D	Sprite 0 image data register B
SPR1POS	%	148	W	A	D	Sprite 1 vert-horiz start pos data
SPR1CTL	%	14A	W	A	D	Sprite 1 position and control data
SPR1DATA	%	14C	W		D	Sprite 1 image data register A
SPR1DATB	%	14E	W		D	Sprite 1 image data register B
SPR2POS	%	150	W	A	D	Sprite 2 vert-horiz start pos data
SPR2CTL	%	152	W	A	D	Sprite 2 position and control data
SPR2DATA	%	154	W		D	Sprite 2 image data register A
SPR2DATB	%	156	W		D	Sprite 2 image data register B
SPR3POS	%	158	W	A	D	Sprite 3 vert-horiz start pos data
SPR3CTL	%	15A	W	A	D	Sprite 3 position and control data
SPR3DATA	%	15C	W		D	Sprite 3 image data register A
SPR3DATB	%	15E	W		D	Sprite 3 image data register B
SPR4POS	%	160	W	A	D	Sprite 4 vert-horiz start pos data
SPR4CTL	%	162	W	A	D	Sprite 4 position and control data
SPR4DATA	%	164	W		D	Sprite 4 image data register A
SPR4DATB	%	166	W		D	Sprite 4 image data register B
SPR5POS	%	168	W	A	D	Sprite 5 vert-horiz start pos data
SPR5CTL	%	16A	W	A	D	Sprite 5 position and control data
SPR5DATA	%	16C	W		D	Sprite 5 image data register A
SPR5DATB	%	16E	W		D	Sprite 5 image data register B
SPR6POS	%	170	W	A	D	Sprite 6 vert-horiz start pos data
SPR6CTL	%	172	W	A	D	Sprite 6 position and control data
SPR6DATA	%	174	W		D	Sprite 6 image data register A
SPR6DATB	%	176	W		D	Sprite 6 image data register B
SPR7POS	%	178	W	A	D	Sprite 7 vert-horiz start pos data
SPR7CTL	%	17A	W	A	D	Sprite 7 position and control data
SPR7DATA	%	17C	W		D	Sprite 7 image data register A
SPR7DATB	%	17E	W		D	Sprite 7 image data register B
COLOR00		180	W		D	Color table 00
COLOR01		182	W		D	Color table 01
COLOR02		184	W		D	Color table 02
COLOR03		186	W		D	Color table 03
COLOR04		188	W		D	Color table 04
COLOR05		18A	W		D	Color table 05
COLOR06		18C	W		D	Color table 06
COLOR07		18E	W		D	Color table 07
COLOR08		190	W		D	Color table 08
COLOR09		192	W		D	Color table 09
COLOR10		194	W		D	Color table 10
COLOR11		196	W		D	Color table 11
COLOR12		198	W		D	Color table 12
COLOR13		19A	W		D	Color table 13
COLOR14		19C	W		D	Color table 14
COLOR15		19E	W		D	Color table 15
COLOR16		1A0	W		D	Color table 16
COLOR17		1A2	W		D	Color table 17
COLOR18		1A4	W		D	Color table 18
COLOR19		1A6	W		D	Color table 19
COLOR20		1A8	W		D	Color table 20
COLOR21		1AA	W		D	Color table 21
COLOR22		1AC	W		D	Color table 22
COLOR23		1AE	W		D	Color table 23

COLOR24		1B0	W	D	Color table 24	
COLOR25		1B2	W	D	Color table 25	
COLOR26		1B4	W	D	Color table 26	
COLOR27		1B6	W	D	Color table 27	
COLOR28		1B8	W	D	Color table 28	
COLOR29		1BA	W	D	Color table 29	
COLOR30		1BC	W	D	Color table 30	
COLOR31		1BE	W	D	Color table 31	
HTOTAL	h	1C0	W	A	Highest number count in horiz line (VARBEAMEN = 1)	
HSSTOP	h	1C2	W	A	Horiz line pos for HSYNC stop	
HBSTRT	h	1C4	W	A	D	Horiz line pos for HBLANK start
HBSTOP	h	1C6	W	A	D	Horiz line pos for HBLANK stop
VTOTAL	h	1C8	W	A	Highest numbered vertical line (VARBEAMEN = 1)	
VSSTOP	h	1CA	W	A	Vert line for VBLANK start	
VBSTRT	h	1CC	W	A	Vert line for VBLANK start	
VBSTOP	h	1CE	W	A	Vert line for VBLANK stop	
SPRHSTRT	h	1D0	W	A	UHRES sprite vertical start	
SPRHSTOP	h	1D2	W	A	UHRES sprite vertical stop	
BPLHSTRT	h	1D4	W	A	UHRES bit plane vertical stop	
BPLHSTOP	h	1D6	W	A	UHRES bit plane vertical stop	
HHPOSW	h	1D8	W	A	DUAL mode hires H beam counter write	
HHPOSR	h	1DA	R	A	DUAL mode hires H beam counter read	
BEAMCON0	h	1DC	W	A	Beam counter control register (SHRES,UHRES,PAL)	
HSSTRT	h	1DE	W	A	Horizontal sync start (VARHSY)	
VSSTRT	h	1E0	W	A	Vertical sync start (VARVSY)	
HCENTER	h	1E2	W	A	Horizontal pos for vsync on interlace	
DIWHIGH	h	1E4	W	A	D	Display window upper bits for start/stop
BPLHMOD	h	1E6	W	A	UHRES bit plane modulo	
SPRHPTH	+h	1E8	W	A	UHRES sprite pointer (high 5 bits)	
SPRHPTL	+h	1EA	W	A	UHRES sprite pointer (low 15 bits)	
BPLHPTH	+h	1EC	W	A	VRam (UHRES) bitplane pointer (hi 5 bits)	
BPLHPTL	+h	1EE	W	A	VRam (UHRES) bitplane pointer (lo 15 bits)	
RESERVED		1F0 - 1FA				
FMODE	p	1FC	W	A	D	Fetch mode register
NO-OP (NULL)		1FE				Can also indicate last 2 or 3 refresh cycles or the restart of the COPPER after lockup.

## 1.5 Some Notes to Start Width

### 4. List of Registers Ordered Alphabetically

-----

P = New register in Pandora chip set  
p = Stuff added or changed in hires chips  
H = New register in hires chips  
h = stuff added or changed in hires chips

A = Agnus/Alice chip  
D = Denise/Lisa chip  
P = Paula chip

W = Write

R = Read

ER = Early read. This is a DMA data transfer to RAM, from either the disk or from the blitter, Ram timing requires data to be on the bus earlier than microprocessor read cycles. These transfers are therefore initiated by Agnus timing, rather than a read address on the register address bus (RGA).

## 1.6 ADKCON

NAME rev ADDR type chip Description

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

BITS	USE	DESCRIPTION										
15	SET/CLEAR	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	<table border="1"> <thead> <tr> <th>CODE</th> <th>PRECOMP VALUE</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>none</td> </tr> <tr> <td>01</td> <td>140 ns</td> </tr> <tr> <td>10</td> <td>280 ns</td> </tr> <tr> <td>11</td> <td>560 ns</td> </tr> </tbody> </table>	CODE	PRECOMP VALUE	00	none	01	140 ns	10	280 ns	11	560 ns
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,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 will 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.

## 1.7 AUDxLCH

NAME	rev	ADDR	type	chip	Description
AUDxLCH	h	0A0	W	A	Audio channel x location (high 5 bits) (old-3 bits)

## 1.8 AUDxLCL

NAME	rev	ADDR	type	chip	Description
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 reg and therefore only needs to be reloaded if a different memory location is to be outputted.

## 1.9 AUDxLEN

NAME	rev	ADDR	type	chip	Description
AUDxLEN		0A4	W	P	Audio channel x length

This reg contains the length (number of words) of audio channel x DMA data.

## 1.10 AUDxPER

NAME	rev	ADDR	type	chip	Description
AUDxPER	h	0A6	W	P	Audio channel x period

This reg contains the period (rate) of audio channel x DMA data transfer.

The minimum period is 124 clocks. This means that the smallest number that should be placed in this reg is 124.

## 1.11 AUDxVOL

NAME rev ADDR type chip Description

AUDxVOL 0A8 W P Audio channel x volume

This reg 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 the 64 levels (000000 = no output, 111111 = 63 ones, one zero)

## 1.12 AUDxDAT

NAME rev ADDR type chip Description

AUDxDAT 0AA W P Audio channel x data

This reg 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 outputted 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 reg from RAM. The processor can also write directly to this reg. When the DMA data is finished (words outputted=length) and the data in this reg has been used, an audio channel interrupt request is set.

## 1.13 BEAMCON0

NAME rev ADDR type chip Description

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 hardware vertical 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 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 conosite 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 VARVSTEN & 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 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 diffrent beam counter total values) on the main horiz counter. It also disables hard display stops on 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

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 two 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**= Enables 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.

## 1.14 BLTxPTH

NAME	rev	ADDR	type	chip	Description
BLTxPTH	h	050	W	A	Blitter Point to x (High 5 bits)

See also: BLTxPTL

## 1.15 BLTxPTL

NAME	rev	ADDR	type	chip	Description
BLTxPTL	h	052	W	A	Blitter Pointer to x (Low 15 bits)

This pair of registers (see also: BLTxPTH) 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).

## 1.16 BLTxMOD

NAME	rev	ADDR	type	chip	Description
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.

## 1.17 BLTAFWM

NAME	rev	ADDR	type	chip	Description
BLTAFWM		044	W	A	Blitter first word mask for source A

See also: BLTALWM

## 1.18 BLTALWM

NAME	rev	ADDR	type	chip	Description
BLTALWM		046	W	A	Blitter last word mask for source A

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

## 1.19 BLTxDAT

NAME	rev	ADDR	type	chip	Description
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.

## 1.20 BLTDDAT

NAME	rev	ADDR	type	chip	Description
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.

## 1.21 BLTSIZE

NAME rev ADDR type chip Description

-----  
 BLTSIZE 058 W A Blitter start and size (win/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)

## 1.22 BLTCON0

NAME rev ADDR type chip Description

-----  
 BLTCON0 040 W A Blitter control register 0  
 BLTCON0L H 05A W A Blitter control register 0 (lower 8 bits)  
 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, are and line, which are selected by bit 0 of BLTCON1, as show 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	DPF7
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 (dec 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)

## 1.23 BLTSIZH

NAME rev ADDR type chip Description

-----  
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

```

See also: BLTSIZV

## 1.24 BLTSIZV

NAME rev ADDR type chip Description

-----  
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 h9 h8 h7 h6 h5 h4 h3 h2 h1 h0

```

These are the blitter size regs 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 bits 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.

## 1.25 BPLHDAT

NAME rev ADDR type chip Description

-----  
 BPLHDAT h 07A W Ext logic UHRES bit plane identifier

This is the number (sign extended) that is added to the UHRES bitplane pointer (BPLHPTL,H) every line, and then another 2 is added, just like the other modulus.

## 1.26 BPLHMOD

NAME rev ADDR type chip Description

-----  
 BPLHMOD h 1E6 W A Uhres bit plane modulo

This is the number (sign extended) that is added to the UHRES bitplane pointer (BPLHPTL,H) every line, and then another 2 is added, just like the other modulus.

## 1.27 BPLHPTH

NAME rev ADDR type chip Description

-----  
 BPLHPTH h 1EC W A UHRES (VRAM) bit plane pntr (high 5 bits)

When UHRES is enabled, this pointer comes out on the 2nd 'free' cycle after the start of each horizontal line. It's 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.

## 1.28 BPLHPTL

NAME rev ADDR type chip Description

-----  
 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 horizontal line. It's 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.

## 1.29 bplhstop

NAME rev ADDR type chip Description

-----  
 BPLHSTOP p 1D6 W A UHRES bit plane vertical stop

```

+-----+-----+
| 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).

## 1.30 BPLHSTRT

NAME rev ADDR type chip Description

-----  
 BPLHSTRT h 1D4 W A UHRES bit plane vertical stop

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

## 1.31 BPLxPTH

NAME rev ADDR type chip Description

-----  
 BPLxPTH 0E0 W A Bit plane x pointer (high 5 bits)

```

0E8          x=1,2,3,4,5,6,7,8
0EC
0F0
0F4
p 0F8
p 0FC

```

## 1.32 BPLxPTL

NAME rev ADDR type chip Description

-----  
 BPLxPTL 0E2 W A Bit plane pointer (low 15 bits)

```

0EA          Address of bit plane x (x=1,2,3,4,5,6,7,8) DMA data.
0EE          This pointer must be reinitialized by the processor or
0F2          coprocessor to point to the beginning of bit plane data

```

0F6            every vertical blank time.  
 p 0FA  
 p 0FE

### 1.33 BPLxDAT

NAME	rev	ADDR	type	chip	Description
BPLxDAT	110	W	A		Bit plane x data (parallel to serial convert)
	112				These regs recieve the DMA data fetched from RAM by the
	114				bit plane address pointers described above.
	116				They may also be rewritten by either micro.
	118				they act as a 8 word parallel to serial buffer for up
	11A				to 8 memory 'bit planes'. x=1-8 the parallel to serial
p	11C				conversion id triggered whenever bit plane #1 is
p	11E				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.

### 1.34 BPLxMOD

NAME	rev	ADDR	type	chip	Description
BPL1MOD	108	W	A		Bit plane modulo (odd planes)
BPL2MOD	10A	W	A		Bit plane modulo (even planes)

These registers contain the modulus 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 modulus, the odd and even bit planes may have sizes that are different from each other, as well as different from the display window size.

If scan-doubling is enabled, BPL1MOD serves as the primary bitplane modulus and BPL2MOD serves as the alternate. Lines whose LSBs of beam counter and DIWSTRT match are designated primary, whereas lines whose LSBs don't match are designated alternate.

### 1.35 BPLCON0

NAME	rev	ADDR	type	chip	Description
BPLCON0	p 100	W	D		Bit plane control reg. (misc, control bits)

BIT#	BPLCON0	DESCRIPTION

15	HIRES	HIRES = High resolution (640*200/640*400 interlace mode)
14	BPU2	Bit plane use code 0000-1000 (NODE thru 8 inclusive)
13	BPU1	
12	BPU0	
11	HAM	Hold and modify mode, now using either 6 or 8 bit planes.
10	DPF	Double playfield (PFI=odd FP2= 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 bit plane. This is called EXTRA-HALFBRITE Mode.
09	COLOR	Enables color burst output signal
08	GAUD	Genlock audio enable. This level appears on the ZD pin on denise during all blanking periods, unless ZDCLK bit is set.
07	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.
06	SHRES	Super hi-res mode (35ns pixel width)
05	BYPASS=0	Bitplanes are scrolled and prioritized normally, but bypass color table and 8 bit wide data appear on R(7:0).
04	BPU3=0	See above (BPU0/1/2)
03	LPEN	Light pen enable (reset on power up)
02	LACE	Interlace enable (reset on power up)
01	ERSY	External resync (HSYNC, VSYNC pads become inputs) (reset on power up)
00	ECSENA=0	When low (default), the following bits in BPLCON3 are disabled: BRDRBLNK, BRDNTRAN, ZDCLKEN, BRDSPRT, and EXTBLKEN. These 5 bits can always be set by writing to BPLCON3, however there effects are inhibited until ECSENA goes high. This allows rapid context switching between pre-ECS viewports and new ones.

## 1.36 BPLCON1

NAME rev ADDR type chip Description

BPLCON1 p 102 W D Bit plane control reg. (horiz, scroll counter)

BIT#	BPLCON1	DESCRIPTION
15	PF2H7=0	(PF2Hx =) Playfield 2 horizontal scroll code, x=0-7
14	PF2H6=0	
13	PF2H1=0	
12	PF2H0=0	
11	PF1H7=0	(PF1Hx =) Playfield 1 horizontal scroll code, x=0-7
10	PF1H6=0	where PFyH0=LSB=35ns SHRES pixel (bits have been
09	PF1H1=0	renamed, old PFyH0 now PFyH2, ect). Now that the scroll
08	PF1H0=0	range has been quadrupled to allow for wider (32 or
		64 bits) bitplanes.

07	PF2H5		
06	PF2H4		
05	PF2H3		
04	PF2H2		
03	PF1H5		
02	PF1H4		
01	PF1H3		
00	PF1H2		
+-----+			

### 1.37 BPLCON2

NAME rev ADDR type chip Description

BPLCON2 p 104 W D Bit plane control reg. (new control bits)

BIT#	BPLCON2	DESCRIPTION
15	X	don't care- but drive to 0 for upward compatibility!
14	ZDBPSEL2	3 bit field which selects which bitplane is to be used
		for ZD when ZDBBPEN is set- 000 selects BB1 and 111
		selects BP8.
13	ZDBPSEL1	
12	ZDBPSEL0	
11	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.
10	ZDCTEN	Causes ZD pin to mirror bit #15 of the active entry in
		high color table. When ZDCTEN is reset ZD reverts to
		mirroring color (0).
09	KILLEHB	Disables extra half brite mode.
08	RDRAM=0	Causes color table address to read the color table
		instead of writing to it.
07	SOGEN=0	When set causes SOG output pin to go high
06	PF2PRI	Gives playfield 2 priority over playfield 1.
05	PF2P2	Playfield 2 priority code (with resp. to sprites).
04	PF2P1	
03	PF2P0	
02	PF1P2	Playfield 1 priority code (with resp. to sprites).
01	PF1P1	
00	PF1P0	

### 1.38 BPLTCON3

NAME rev ADDR type chip Description

BPLCON3 p 106 W D Bit plane control reg. (enhanced features)

BIT#	BPLCON3	DESCRIPTION
------	---------	-------------

15	BANK2=0	BANKx = Selects one of eight color banks, x=0-2.																																																																																																																																		
14	BANK1=0																																																																																																																																			
13	BANK0=0																																																																																																																																			
12	PF2OF2=0	Determine bit plane color table offset when playfield 2 has priority in dual playfield mode:																																																																																																																																		
		<table border="1"> <thead> <tr> <th>PF20F</th> <th colspan="11">AFFECTED BITPLANE</th> <th>OFFSET</th> </tr> <tr> <th></th> <th>2</th> <th>1</th> <th>0</th> <th>8</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>(decimal)</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>none</td> </tr> <tr> <td></td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>-</td> <td>2</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>-</td> <td>-</td> <td>4</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>-</td> <td>-</td> <td>8 (default)</td> </tr> <tr> <td></td> <td>1</td> <td>0</td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>16</td> </tr> <tr> <td></td> <td>1</td> <td>0</td> <td>1</td> <td>-</td> <td>-</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>32</td> </tr> <tr> <td></td> <td>1</td> <td>1</td> <td>0</td> <td>-</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>64</td> </tr> <tr> <td></td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>128</td> </tr> </tbody> </table>	PF20F	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
PF20F	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																																																																																																																								
11	PF2OF1=1																																																																																																																																			
10	PF2OF0=1																																																																																																																																			
09	LOCT=0	Dictates that subsequent color palette values will be written to a second 12- bit color palette, constituting the RGB low minus order bits. Writes to the normal hi minus order color palette automatically copied to the low order for backwards compatibility.																																																																																																																																		
08	X	don't care- but drive to 0 for upward compatibility!																																																																																																																																		
07	SPRES1=0	Determine resolution of all 8 sprites (x=0,1):																																																																																																																																		
		<table border="1"> <thead> <tr> <th>SPRES1</th> <th>SPRES0</th> <th>SPRITE RESOLUTION</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>ECS defaults (LORES,HIRES=140ns,SHRES=70ns)</td> </tr> <tr> <td>0</td> <td>1</td> <td>LORES (140ns)</td> </tr> <tr> <td>1</td> <td>0</td> <td>HIRES (70ns)</td> </tr> <tr> <td>1</td> <td>1</td> <td>SHRES (35ns)</td> </tr> </tbody> </table>	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)																																																																																																																			
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)																																																																																																																																		
06	SPRES0=0																																																																																																																																			
05	BRDRBLNK=0	"Border area" is blanked instead of color (0). Disabled when ECSENA low.																																																																																																																																		
04	BRDNTRAN=0	"Border area" is non minus transparant (ZD pin is low when border is displayed). Disabled when ECSENA low.																																																																																																																																		
03	X	don't care- but drive to 0 for upward compatibility!																																																																																																																																		
02	ZDCLKEN=0	ZD pin outputs a 14MHz clock whose falling edge coincides with hires (7MHz) video data. this bit when set disables all other ZD functions. Disabled when ESCENA low.																																																																																																																																		
01	BRDSPRT=0	Enables sprites outside the display window. disabled when ESCENA low.																																																																																																																																		
00	EXTBLKEN=0	Causes BLANK output to be programmable instead of reflecting internal fixed decodes. Disabled when ESCENA low.																																																																																																																																		

### 1.39 BPLCON4

NAME rev ADDR type chip Description

BPLCON4 p 10c W D Bit plane control reg. (display masks)

BIT#	BPLCON4	DESCRIPTION
15	BPLAM7=0	This 8 bit field is XOR'ed with the 8 bit plane color address, thereby altering the color address sent to the color table (x=1-8)
14	BPLAM6=0	
13	BPLAM5=0	
12	BPLAM4=0	
11	BPLAM3=0	
10	BPLAM2=0	
09	BPLAM1=0	
08	BPLAM0=0	
07	ESPRM7=0	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)
06	ESPRM6=0	
05	ESPRM5=0	
04	ESPRM4=1	
03	OSPRM7=0	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)
02	OSPRM6=0	
01	OSPRM5=0	
00	OSPRM4=1	

### 1.40 CLXCON

NAME rev ADDR type chip Description

CLXCON 098 W A 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. Writing to this register resets the bits in CLXCON2.

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 Sprite2)	
12	ENSP1	Enable Sprite 1 (ORed with Sprite 0)	
11	ENSP6	Enable bit plane 6 (match reqd. for collision	
10	ENSP5	Enable bit plane 5 (match reqd. for collision	
09	ENSP4	Enable bit plane 4 (match reqd. for collision	
08	ENSP3	Enable bit plane 3 (match reqd. for collision	
07	ENSP2	Enable bit plane 2 (match reqd. for collision	
06	ENSP1	Enable bit plane 1 (match reqd. for collision	
05	ENSP6	Match value for bit plane 6 collision	
04	ENSP5	Match value for bit plane 5 collision	
03	ENSP4	Match value for bit plane 4 collision	
02	ENSP3	Match value for bit plane 3 collision	
01	ENSP2	Match value for bit plane 2 collision	
00	ENSP1	Match value for bit plane 1 collision	

## 1.41 CLXCON2

NAME rev ADDR type chip Description

-----  
 CLXCON2 P 10C W D Extended collision control

This reg controls when bit planes 7 and 8 are included in collision detection, and there required state if included. Contents of this register are reset by a write to CLXCON.

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: Disable bit planes cannot prevent collisions. Therefore if all bitplanes are disabled, collision will be continuous, regardless of the match values.

## 1.42 CLXDAT

NAME rev ADDR type chip Description

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

This address reads (and clears) the collision detection reg. 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 2 to Playfield 2

## 1.43 COLORx

NAME	rev	ADDR	type	chip	Description
COLORxx	180-1BE	W			COLOR table xx

There 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 regs, selection of which is controlled by the LOCT reg 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 as well, so that the 4 bits-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.

## 1.44 COPCON

NAME rev ADDR type chip Description

-----  
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 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)

## 1.45 COPJMP1

NAME rev ADDR type chip Description

-----  
COPJMP1 088 S A Coprocessor restart at first location

See: COPJMP2

## 1.46 COPJMP2

NAME rev ADDR type chip Description

-----  
COPJMP2 08A S A Coprocessor restart at second location

These address are strobe address, that when written to cause the coprocessor to jump indirect using the address contained in the first or second location regs described below. The coprocessor itself can write to these address, causing it's own jump indirect.

## 1.47 COP1LCH

NAME	rev	ADDR	type	chip	Description
COP1LCH	h	080	W	A	A Coprocessor first location reg (high 5 bits) (old-3 bits)
COP1LCL		082	W	A	A Coprocessor first location reg (low 15 bits)
COP2LCH	h	084	W	A	A Coprocessor second location reg (high 5 bits) (old-3 bits)
COP2LCL		086	W	A	A Coprocessor second location reg (low 15 bits)

These regs contain the jump addresses described in COPINS

### 1.48 COPINS

NAME	rev	ADDR	type	chip	Description
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 its 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 it's 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 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 program counter for initalization or jumps.

They are transfered to the program counter whenever strobe address 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.T his insures a determined startup address, and state.

## 1.49 DDFSTRT

NAME    rev ADDR type chip Description

```

-----
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 timing 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  XX   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 and mins, that is)

### 1.50 DIWSTRT

NAME rev ADDR type chip Description

```

-----
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).  
 \* Poof.. (see DIWHIGH for exceptions)

### 1.51 DIWHIGH

NAME rev ADDR type chip Description

-----  
 DIWHIGH p 1E4 W A D Display window upper bits for start, stop  
 This is an added register for Hires chips, and allows  
 larger start & stop ranges. If it is not written, the above  
 (DIWSTRT,STOP) description holds. If this register is written,  
 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)
    
```

Take care (X) bits should always be written to 0 to  
 maintain upwards compatibility. H1 and H0 values define  
 70ns amd 35ns increments respectively, and 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 to be called H0 is now referred to as H2.

### 1.52 DMACON

NAME rev ADDR type chip Description

-----  
 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 prioiry (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)



## 1.56 DSKDATR

NAME rev ADDR type chip Description

-----  
 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 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.

## 1.57 DSKBYTR

NAME rev ADDR type chip Description

-----  
 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#	FUNCTION	DESCRIPTION
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

## 1.58 DSKSYNC

NAME rev ADDR type chip Description

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

## 1.59 FMODE

NAME rev ADDR type chip Description

-----  
 FMODE P 1FC W Memory Fetch Mode

This register controls the fetch mechanism for different types of Chip RAM accesses:

BIT#	FUNCTION	DESCRIPTION
15	SSCAN2	Global enable for sprite scan-doubling.
14	BSCAN2	Enables the use of 2nd P/F modulus on an alternate line basis to support bitplane scan-doubling.
13-04	Unused	
03	SPAGEM	Sprite page mode (double CAS)
02	SPR32	Sprite 32 bit wide mode
01	BPAGEM	Bitplane Page Mode (double CAS)
00	BLP32	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

## 1.60 HBSTOP

NAME rev ADDR type chip Description

HBSTOP 1C6 W D Horizontal STOP position  
 HBSTRT 1C4 W D Horizontal START position

Bits 7-0 contain the stop and start positions, respectively, for programmed horizontal blanking in 280nS increments. Bits 10-8 provide a fine position control in 35nS increments.

BIT#	FUNCTION	DESCRIPTION
15-11	x	(unused)
10	H1	140nS
09	H1	70nS
08	H0	35nS
07	H10	35840nS
06	H9	17920nS

05	H8	8960nS	
04	H7	4480nS	
03	H6	2240nS	
02	H5	1120nS	
01	H4	560nS	
00	H3	280nS	
+-----+-----+-----+-----+			

## 1.61 HCENTER

NAME rev ADDR type chip Description

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.

## 1.62 HHPOSR

NAME rev ADDR type chip Description

HHPOSR H 1DA R A DUAL mode hires Hbeam counter read  
HHPOSW H 1D8 W A DUAL mode hires Hbeam counter write

This the secondary beam counter for the faster mode, triggering the UHRES pointers & doing the comparisons for HBSTRT, STOP, HTOTAL, HSSRT, HSSTOP (See HTOTAL for bits)

## 1.63 HSSTOP

NAME rev ADDR type chip Description

HSSTOP H 1C2 W A Horiz line position for SYNC stop

Sets # of colour clocks for sync stop (HTOTAL for bits)

## 1.64 HSSTRT

NAME rev ADDR type chip Description

HSSTRT H 1DE W A Horiz line position for HSYNC stop

Sets # of colour clocks for sync start (HTOTAL for bits)  
See BEAMCON0 for details of when these 2 are active.

## 1.65 HTOTAL

NAME rev ADDR type chip Description

-----  
 HTOTAL H 1C0 W A Highest colour clock count in horiz line

```

BIT# 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 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 theis many + 1 280nS increments. If the  
 pal bit & LOLDIS are not high, long line/skort line toggle  
 will occur, and there will be this many +2 every other line.  
 Active if VARBEAMEN=1 or DUAL+1.

## 1.66 INTREQ

NAME rev ADDR type chip Description

-----  
 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 arbitrary data into this register.  
 The bit assignments are identical to the enable register  
 below.

## 1.67 INTENA

NAME rev ADDR type chip Description

-----  
 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#	FUNCTION	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
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 has 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	

### 1.68 JOYxDAT

NAME rev ADDR type chip Description

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 LISAIID.

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	M1V	JOY1DAT Horizontal Clock
5	M1VQ	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 signal pair. 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

## 1.69 JOYTEST

NAME rev ADDR type chip Description

-----  
 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

## 1.70 LISAIID

NAME rev ADDR type chip Description

-----  
 LISAIID 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 implementation.

The 8 low-order bits are encoded as follows:

BIT#	Description
7-4	Lisa/Denise/ECS Denise Revision level (decrement to bump revision level, hex F represents 0th rev. level).
3	Maintain as a 1 for future generation
2	When low indicates AA feature set (LISA)
1	When low indicates ECS feature set (LISA or ECS DENISE)
0	Maintain as a 1 for future generation

## 1.71 POTxDAT

NAME rev ADDR type chip Description

```
-----
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	RX	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.

## 1.72 POTGO

NAME rev ADDR type chip Description

```
-----
POTGO 034 W      P Pot port (4 bit) bi-direction and data,
and pot counter start.
```

## 1.73 POTINP

NAME rev ADDR type chip Description

```
-----
POTINP 016 R      P Pot pin data read
```

This register controls a 4 bit bi-direction I/O port that shares the same 4 pins as the 4 pot counters above.

BIT#	FUNCTION	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	Out put 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)	
+-----+			

## 1.74 REFPTR

NAME rev ADDR type chip Description

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.

## 1.75 SERDAT

NAME rev ADDR type chip Description

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

## 1.76 SERDATR

NAME rev ADDR type chip Description

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#	FUNCTION	DESCRIPTION
15	OVRUN	Serial port receiver overrun
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.

## 1.77 SERPER

NAME rev ADDR type chip Description

SERPER 032 W P Serial port period and control.

This register contains the control bit LONG referred 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#	FUNCTION	DESCRIPTION
15	LONG	Defines serial receive as 9 bit word.
14-00	RATE	Defines baud rate = $1 / ((N+1) * .2794 \text{ microseconds})$

## 1.78 SPRHDAT

NAME rev ADDR type chip Description

SPRHDAT H 078 W exe logic UHRES sprite identifier and data

This identifies the cycle when this pointer address is on the bus accessing the memory.

## 1.79 SPRHPTH

NAME rev ADDR type chip Description

```
-----
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 and 3rd 'free' cycles (see BPLHPTH,L) after horiz line start. It increments for the next line.

## 1.80 SPRHSTOP

NAME rev ADDR type chip Description

```
-----
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   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)

## 1.81 SPRHSTRT

NAME rev ADDR type chip Description

```
-----
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
```

## 1.82 SPRxPTH

NAME rev ADDR type chip Description

```
-----
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.

## 1.83 sprxpos

NAME rev ADDR type chip Description

-----  
 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 register below.
07-00	SH10-SH3	Sprite horizontal start value. Low order 3 bits are in SPRxCTL register below. If SSCAN2 bit in FMODE is set, then disable SH10 horizontal coincidence detect.This bit is then free to be used by ALICE as an individual scan double enable.

## 1.84 sprxctl

NAME rev ADDR type chip Description

-----  
 SPRxCTL p 142 W A D Sprite 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.

## 1.85 SPRxDAT

NAME rev ADDR type chip Description

-----  
 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 outputted to the display, MSB first on the left.

NOTE: Writing to the A buffer enables (arms) the sprite. Writing to the SPRxCTL registers 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.

## 1.86 STREQU

NAME	rev	ADDR	type	chip	Description
STREQU	038	S	D		Strobe for horiz sync with VB (vert blank) and EQU
STRVBL	038	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 horiz line (228CC)

One of the first 3 strobe addresses above, it is placed on the RGA bus during the first 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.

## 1.87 vbstop

NAME	rev	ADDR	type	chip	Description
VBSTOP	H 1CE	W	A		Vertical line for VBLANK stop
VBSRTR	H 1CC	W	A		Vertical line for VBLANK start (V10-0 <- D10-0) Affects CSY pin if BLAKEN=1 and VSY pin if CSCBEN=1 (see BEAMCON0)

## 1.88 VPOSR

NAME	rev	ADDR	type	chip	Description
VPOSR	p 004	R	A		Read vert most sig. bits (and frame flop)
VPOSW	02A	W	A		Write 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 rev4      = 20 Pal, 30 NTSC
8372 (Fat-hr) (agnushr), rev 5          = 22 Pal, 31 NTSC
8374 (Alice) thru rev 2                 = 22 Pal, 32 NTSC
8374 (Alice) rev 3 thru rev 4          = 23 Pal, 33 NTSC

```

LOL = Long line bit. When low, it indicates short raster line.  
v9,10 -- hires chips only (20,30 identifiers)

## 1.89 VHPOSR

```

NAME    rev ADDR type chip Description
-----

```

```

VHPOSR   006 R  A    Read vert and horiz position of beam, or lightpen
VHPOSW   02C W  A    Write vert 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 WITH ( 280 nS)

## 1.90 VSSTOP

```

NAME    rev ADDR type chip Description
-----

```

```

VSSTOP H  1CA 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 and the short field is this  
many + 1.

## 1.91 lisamodes

### 5. New LISA Display Modes

We now have a palette of 2~24 colours.

```

LORES  (320x200)
-----

```

6 Bitplane (non HAM, non EHB)	64 colours	!
7 Bitplane	128 colours	!
8 Bitplane	256 colours	!
8 Bitplane HAM	Any $2^{24}$ colours	!

Dual playfield, Max 4 bitplane per playfield. 16 colours per playfield. The bank of 16 colours in the 256 colour palette is selectable per playfield. !

#### HIRES (640x200)

-----

5 Bitplanes	32 colours	@
6 Bitplanes	64 colours	@
7 Bitplanes	128 colours	@
8 Bitplanes	256 colours	@
6 Bitplanes EHB	32 * 2 colours	@
6 Bitplanes HAM	4096 colours	@
8 Bitplanes HAM	any of $2^{24}$ colours	@

Dual playfield, max 4 bitplane per playfield 16 colours per playfield. The bank of 16 colours in the 256 colour palette is selectable per playfield. ! or @

#### SUPERHIRES (1280X200)

-----

1 or 2 bitplanes, as ECS, but no colour fudging	!
3 Bitplanes 8 colours	@
4 Bitplanes 16 colours	@
5 Bitplanes 32 colours	\$
6 Bitplanes 64 colours	\$
7 Bitplanes 128 colours	\$
8 Bitplanes 256 colours	\$
6 Bitplanes EHB 32 * 2 colours	\$
6 Bitplanes HAM 4096 colours	\$
8 Bitplanes HAM any of 2~24 colours	\$

Dual Playfield, max 4 bitplanes per playfield @ or \$  
16 colours per playfield. The bank of 16 of colours in the 256 colours palette is selectable per playfield.

#### VGA (640X480 non-interlaced)

-----

1 or 2 bitplanes, as ECS, but no colour fudging	!
3 Bitplanes 8 colours	@
4 Bitplanes 16 colours	@
5 Bitplanes 32 colours	\$
6 Bitplanes 64 colours	\$
7 Bitplanes 128 colours	\$
8 Bitplanes 256 colours	\$
6 Bitplanes EHB 32 * 2 colours	\$
6 Bitplanes HAM 4096 colours	\$
8 Bitplanes HAM any of 2~24 colours	\$

Dual playfield, Max 4 bitplanes per playfield @ or \$  
 16 colours per playfield . The bank of 16 colours  
 in the 256 colour palette is selectable per playfield

Super 72 (848x614 interlaced, 70 Hz frame rate)  
 -----

1 or 2 bitplanes, as ECS, but no colour fudging		1X
3 Bitplanes	8 colours	2X
4 Bitplanes	16 colours	2X
5 Bitplanes	32 colours	4X
6 Bitplanes	64 colours	4X
7 Bitplanes	128 colours	4X
8 Bitplanes	256 colours	4X
6 Bitplanes EHB	32 * 2 colours	4X
6 Bitplanes HAM	4096 colours	4X
8 Bitplanes HAM	any of 2~24 colours	4X

Dual playfield, Max 4 bitplanes per playfield 2X or 4X  
 16 colours per playfield . The bank of 16 colours  
 in the 256 colour palette is selectable per playfield

All playfield scrolling is now in 35ns increments.  
 Pre AA scrolling was in 140ns increments.

Scroll Range as Programmed in BPLCON1  
 -----

1X Modes	LORES Pixels	SHRES Pixels
LORES	0-15	0-63
HIRES	0-7	0-31
SHRES	0-3	0-15

2X Modes	LORES Pixels	SHRES Pixels
LORES	0-31	0-127
HIRES	0-15	0-63
SHRES	0-7	0-31

4X Modes	LORES Pixels	SHRES Pixels
LORES	0-63	0-255
HIRES	0-31	0-127
SHRES	0-15	0-63

Sprites

---

-----

All sprites can now be displayed in either:

- 1) ECS default mode
- 2) 140 ns (this is not ECS mode!)
- 3) 70 ns
- 4) 35 ns

on display resolution. eg 35 ns sprites on a lores screen, or 140 ns sprites on a superhires screen.

Sprites are either 16, 32, or 64 bits wide.

Sprites can be attached in any mode (formerly could not attach sprites in the ECS SHRES 35ns resolution mode).

Can use any bank of 16 colours from the 256 colour palette for the sprite colours.

Key:

- ! needs 1x Bandwidth (old modes)
  - @ needs 2x Bandwidth (normal CAS 32bit bus with or double CAS 16 bit bus width)
  - \$ needs 4x Bandwidth (double CAS 32bit bus width)
-