Hardware

COLLABORATORS			
	TITLE : Hardware		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 28, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Hardware

1.1 Amiga® Hardware Reference Manual:A Register Summary Alphabetical Order

This appendix contains the definitive summary, in alphabetical order, of
the Amiga's custom chip register set and the usages of the individual bits.

The addresses shown here are used by the special custom chips (named
"Paula", "Agnus", and "Denise") for transferring data among themselves.
Also, the Copper uses these addresses for writing to the special chip
registers. To write to these registers with the 680x0, calculate the 680x0
address using this formula:

        680x0 address = (chip address) + $DFF000

For example, for the 680x0 to write to ADKCON (address = $09E), the
address would be $DFF09E. No other access address is valid. Do not
attempt to access any documented or unused registers.

All of the "pointer" type registers are organized as 32 bits on a long
word boundary. These registers may be written with one MOVE.L
instruction. The lowest bit of all pointers must be written as zero. The
custom chips can only access Chip memory; using a non-Chip address will
fail (See the AllocMem() documentation or your compiler manual for more
information on Chip memory). Disk data, sprite data, bitplane data, audio
data, copper lists and anything that will be blitted or accessed by custom
chip DMA must be located in chip memory.

When strobing any register which responds to either a read or a write,
(for example copjmp2) be sure to use a MOVE.W, not CLR.W. The CLR
instruction causes a read and a clear (two accesses) on a 68000, but only
a single access on 68020 processors. This will give different results on
different processors.

    Warning:

    Registers are either read-only or write-only. In the following
    descriptions, if a register is marked as a read-only register, only
    read its contents. Do not attempt to write to a read-only register,
    as this will cause unpredictable results. If a register is marked as
    a write-only register, do not attempt to read from it, as this may
    trash the register and crash the system.

If a bit is described as unused in a write-only register, be sure to keep
that bit clear when writing values to that register. Similarly, do not
rely on the values of unused bits when reading from a read only register.
Further, do not write to an address or register that is not documented or
defined in this appendix. Setting unused bits in a write-only register,
reading unused bits from a read only register and writing to undocumented
registers or addresses may cause serious future software incompatibility
if those bits or addresses are implemented in the future by Commodore
Amiga.

    About the ECS registers.

    Registers denoted with an "(E)" in the chip column means that those
    registers have been changed the Enhanced Chip Set(ECS). The ECS is
    found in the A3000, and is installable in the A500 and A2000.
    Certain ECS registers are completely new, others have been extended
    in their functionality. See the register map in Appendix C for
    information on which ECS registers are new and which have been
    modified.

```
ADKCON        BLTSIZH      COP1LCH      DSKDAT       JOY1DAT      STRHOR
ADKCONR       BLTxDAT      COP1LCL      DSKDATR      JOYTEST      STRLONG
AUDxDAT       BLTxMOD      COP2LCH      DSKLEN       POT0DAT      STRVBL
AUDxLCH       BLTxPTH      COP2LCL      DSKPTH       POT1DAT      VBSTOP
AUDxLCL       BLTxPTL      COPCON       DSKPTL       POTGO        VBSTRT
AUDxLEN       BPL1MOD      COPINS       DSKSYNC      POTGOR       VHPOSR
AUDxPER       BPL2MOD      COPJMP1      HBSTOP       REFPTR       VHPOSW
AUDxVOL       BPLCON0      COPJMP2      HBSTRT       SERDAT       VPOSR
BEAMCON0      BPLCON1      DDFSTOP      HCENTER      SERDATR      VPOSW
BLTAFWM       BPLCON2      DDFSTRT      HSSTOP       SERPER       VSSTOP
BLTALWM       BPLCON3      DENISEID     HSSTRT       SPRxCTL      VSSTRT
BLTCON0       BPLxDAT      DIWHIGH      HTOTAL       SPRxDATA     VTOTAL
BLTCON1       BPLxPTH      DIWSTOP      INTENA       SPRxDATB
BLTCON0L      BPLxPTL      DIWSTRT      INTENAR      SPRxPOS
BLTDDAT       CLXCON       DMACON       INTREQ       SPRxPTH
BLTSIZE       CLXDAT       DMACONR      INTREQR      SPRxPTL
BLTSIZV       COLORxx      DSKBYTR      JOY0DAT      STREQU
```

1.2   A Register Summary / ADKCON, ADKCONR

```
                          Agnus/
                 Read/    Denise/
Register Address Write    Paula           Function
-------- ------- -----    -------          --------
ADKCON    09E      W        P       Audio, disk, control write
ADKCONR   010      R        P       Audio, disk, 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 (3F)*2.
        09 MSBSYNC       Enables disk read synchronizing on the MSB
                         (most significant bit). Appl type GCR.
        08 FAST   Disk data clock rate control 1=fast(2us) 0=slow(4us).
                  (fast for MFM, slow for MFM or 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 word xxxxxxxx V6-V0 , Second word P15-P0 (etc)
```

1.3 A Register Summary / AUDxDAT

```
                         Agnus/
                 Read/   Denise/
Register Address Write   Paula           Function
-------- ------- -----   -------         --------
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 that
         are each 2's complement and are outputted
         sequentially (with digital-to-analog conversion)
         to the audio output pins.  (LSB = 3 MV)  The DMA
         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.
```

1.4 A Register Summary / AUDxLCH, AUDxLCL

```
                        Agnus/
                Read/   Denise/
Register Address Write  Paula        Function
-------- ------- -----  -------      --------
AUDxLCH   0A0     W      A( E )  Audio channel x location (high 3 bits,
                                        5 bits if ECS)
AUDxLCL   0A2     W      A       Audio channel x location (low 15 bits)
```

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

1.5   A Register Summary / AUDxLEN

```
                        Agnus/
                Read/   Denise/
Register Address Write  Paula        Function
-------- ------- -----  -------      --------
AUDxLEN   0A4     W      P       Audio channel x length
```

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

1.6   A Register Summary / AUDxPER

```
                        Agnus/
                Read/   Denise/
Register Address Write  Paula        Function
-------- ------- -----  -------      --------
AUDxPER   0A6     W      P( E )  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 decimal. This corresponds to
a maximum sample frequency of 28.86 khz.

1.7   A Register Summary / AUDxVOL

```
                        Agnus/
                Read/   Denise/
Register Address Write  Paula        Function
-------- ------- -----  -------      --------
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.

                    Bit#   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 1s, one 0)
```

1.8 A Register Summary / BEAMCON0

```
                              Agnus/
                       Read/  Denise/
Register Address Write  Paula          Function
-------- ------- -----  -------        --------
BEAMCON0  1DC      W     A( E ) Beam counter control register (SHRES,PAL)
```

1.9 A Register Summary / BLTAFWM, BLTALWM

```
                              Agnus/
                       Read/  Denise/
Register Address Write  Paula          Function
-------- ------- -----  -------        --------
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
                    overrides data from source A.  These registers
                    should be set to all 1s for fill mode or for
                    line-drawing mode.
```

1.10 A Register Summary / BLTCON0, BLTCON1

```
                              Agnus/
                       Read/  Denise/
Register Address Write  Paula          Function
-------- ------- -----  -------        --------
BLTCON0   040      W      A      Blitter control register 0
BLTCON1   042      W      A( E ) Blitter control register 1
```

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

                        AREA MODE ("normal")
                        ------------------------
```

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


          ASH3-0  Shift value of A source
          BSH3-0  Shift value of B source
          USEA    Mode control bit to use 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 (set to 0)


          LINE DRAW      LINE MODE (line draw)
          LINE DRAW      ----------------------------
          LINE DRAW      BIT# BLTCON0     BLTCON1
          LINE DRAW      ---- -------     -------
          LINE DRAW      15   START3      TEXTURE3
          LINE DRAW      14   START2      TEXTURE2
          LINE DRAW      13   START1      TEXTURE1
          LINE DRAW      12   START0      TEXTURE0
          LINE DRAW      11   1             0
          LINE DRAW      10   0             0
          LINE DRAW      09   1             0
          LINE DRAW      08   1             0
          LINE DRAW      07   LF7           0
          LINE DRAW      06   LF6           SIGN
          LINE DRAW      05   LF5            0 (Reserved)
          LINE DRAW      04   LF4           SUD
          LINE DRAW      03   LF3           SUL
          LINE DRAW      02   LF2           AUL
          LINE DRAW      01   LF1           SING
          LINE DRAW      00   LF0           LINE(=1)
          LINE DRAW
          LINE DRAW      START3-0  Starting point of line
          LINE DRAW                (0 thru 15 hex)


          LINE DRAW      LF7-0   Logic function minterm
```

```
              LINE DRAW     select lines should be preloaded
              LINE DRAW     with 4A to select the equation
              LINE DRAW     D=(AC+ABC).  Since A contains a
              LINE DRAW     single bit true (8000), most bits
              LINE DRAW     will pass the C field unchanged
              LINE DRAW     (not A and C), but one bit will
              LINE DRAW     invert the C field and combine it
              LINE DRAW     with texture (A and B and not C).
              LINE DRAW     The A bit is automatically moved
              LINE DRAW     across the word by the hardware.
              LINE DRAW
              LINE DRAW     LINE   Line mode control bit (set to 1)
              LINE DRAW     SIGN   Sign flag
              LINE DRAW     0      Reserved for new mode
              LINE DRAW     SING   Single bit per horizontal line for
              LINE DRAW            use with subsequent area fill
              LINE DRAW     SUD    Sometimes up or down (=AUD*)
              LINE DRAW     SUL    Sometimes up or left
              LINE DRAW     AUL    Always up or left

              LINE DRAW     The 3 bits above select the octant
              LINE DRAW     for line drawing:

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

              LINE DRAW     The "B" source is used for
              LINE DRAW     texturing the drawn lines.
```

1.11 A Register Summary / BLTCON0L

```
                            Agnus/
                    Read/   Denise/
Register Address    Write   Paula           Function
-------- -------    -----   -------         --------
BLTCON0L   05A        W     A( E ) Blitter control 0, lower 8 bits (minterms)
```

1.12 A Register Summary / BLTDDAT

```
                            Agnus/
                    Read/   Denise/
Register Address    Write   Paula           Function
-------- -------    -----   -------         --------
BLTDDAT   ---        -       ----   Blitter destination data register
```

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

1.13   A Register Summary / BLTSIZE, BLTSIZV, BLTSIZH

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- -------  -----  -------        --------
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 =horizontal pixels (6 bits=64
                        words=1024 pixels max)

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

```
BLTSIZV   05C      W     A( E )  Blitter V size (for 15 bit vertical size)
BLTSIZH   05E      W     A( E )  Blitter H size and start (for 11 bit
                                        H size)
```

1.14   A Register Summary / BLTxDAT

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- -------  -----  -------        --------
BLTxDAT   074      W       A       Blitter source x data register
```

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

```
                        preloaded by the microprocessor.

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

1.15   A Register Summary / BLTxMOD

```
                              Agnus/
                      Read/   Denise/
Register Address      Write   Paula          Function
-------- -------      -----   -------        --------
BLTxMOD    064          W        A     Blitter modulo x

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

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

1.16   A Register Summary / BLTxPTH, BLTxPTL

```
                              Agnus/
                      Read/   Denise/
Register Address      Write   Paula          Function
-------- -------      -----   -------        --------
BLTxPTH    050          W       A( E ) Blitter pointer to x (high 3 bits,
                                         5 bits if ECS)
BLTxPTL    052          W        A     Blitter pointer to x (low 15 bits)

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

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

1.17   A Register Summary / BPL1MOD, BPL2MOD

```
                            Agnus/
                     Read/  Denise/
Register Address     Write  Paula          Function
-------- -------     -----  -------        --------
BPL1MOD   108          W      A     Bitplane modulo (odd planes)
BPL2MOD   10A          W      A     Bitplane modulo (even planes)

            These registers contain the modulos for the odd
            and even bitplanes.  A modulo is a number that is
            automatically added to the address at the end of
            each line, so that the address then points to the
            start of the next line.
            Since they have separate modulos, the odd and even
            bitplanes may have sizes that are different from
            each other, as well as different from the display
            window size.
```

1.18   A Register Summary / BPLCON0, BPLCON1, BPLCON2, BPLCON3

```
                            Agnus/
                     Read/  Denise/
Register Address     Write  Paula          Function
-------- -------     -----  -------        --------
BPLCON0   100          W     A D( E ) Bitplane control register (misc.
                                        control bits)
BPLCON1   102          W       D       Bitplane control register
                                          (horizontal scroll control)
BPLCON2   104          W        D( E ) Bitplane control register
                                          (video priority control)

            These registers control the operation of the
            bitplanes and various aspects of the display.

            BIT#    BPLCON0   BPLCON1   BPLCON2
            ----    --------  --------  --------
            15      HIRES     X         X
            14      BPU2      X         X
            13      BPU1      X         X
            12      BPU0      X         X
            11      HOMOD     X         X
            10      DBLPF     X         X
```

```
                       09       COLOR       X              X
                       08       GAUD        X              X
                       07        X         PF2H3           X
                       06        X         PF2H2         PF2PRI
                       05        X         PF2H1         PF2P2
                       04        X         PF2H0         PF2P1
                       03       LPEN       PF1H3         PF2P0
                       02       LACE       PF1H2         PF1P2
                       01       ERSY       PF1H1         PF1P1
                       00        X         PF1H0         PF1P0
```

```
            HIRES=High-resolution (70 ns pixels)
            BPU  =Bitplane use code 000-110 (NONE through 6 inclusive)
            HOMOD=Hold-and-modify mode (1 = Hold-and-modify mode (HAM);
                    0 = Extra Half Brite (EHB) if HAM=0 and BPU=6
                    and DBLPF=0 then bitplane 6 controls an intensity
                    reduction in the other five bitplanes)
            DBLPF=Double playfield (PF1=odd PF2=even bitplanes)
            COLOR=Composite video COLOR enable
            GAUD=Genlock audio enable (muxed on BKGND pin
                    during vertical blanking
            LPEN =Light pen enable (reset on power up)
            LACE =Interlace enable (reset on power up)
            ERSY =External resync (HSYNC, VSYNC pads become
                    inputs) (reset on power up)
            PF2PRI=Playfield 2 (even planes) has priority over
                    (appears in front of) playfield 1
                    (odd planes).
            PF2P=Playfield 2 priority code (with respect
                    to sprites)
            PF1P=Playfield 1 priority code (with respect
                    to sprites)
            PF2H=Playfield 2 horizontal scroll code
            PF1H=Playfield 1 horizontal scroll code
```

```
BPLCON3    106      W      D( E ) Bitplane control (enhanced features)
```

1.19 A Register Summary / BPLxDAT

```
                             Agnus/
                    Read/    Denise/
Register Address    Write    Paula         Function
-------- -------    -----    -------       --------
BPLxDAT    110       W         D      Bitplane x data (parallel-to-serial
                                            convert)
```

```
            These registers receive the DMA data fetched from
            RAM by the bitplane address pointers described
            above.  They may also be written by either
            microprocessor.  They act as a six-word parallel-
            to-serial buffer for up to six memory bitplanes
            (x=1-6).  The parallel-to-serial conversion is
            triggered whenever bitplane #1 is written,
            indicating the completion of all bitplanes for
            that word (16 pixels).  The MSB is output first,
```

and is, therefore, always on the left.

1.20   A Register Summary / BPLxPTH, BPLxPTL

```
                            Agnus/
                    Read/   Denise/
Register Address    Write   Paula         Function
-------- -------    -----   -------       --------
BPLxPTH    0E0        W        A      Bitplane x pointer (high 3 bits)
BPLxPTL    0E2        W        A      Bitplane x pointer (low 15 bits)
```

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

1.21   A Register Summary / CLXCON

```
                            Agnus/
                    Read/   Denise/
Register Address    Write   Paula         Function
-------- -------    -----   -------       --------
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 OR-ing
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 bitplane 6 (match required
                    for collision)
10     ENBP5      Enable bitplane 5 (match required
                    for collision)
09     ENBP4      Enable bitplane 4 (match required
                    for collision)
08     ENBP3      Enable bitplane 3 (match required
                    for collision)
07     ENBP2      Enable bitplane 2 (match required
                    for collision)
06     ENBP1      Enable bitplane 1 (match required
                    for collision)
05     MVBP6      Match value for bitplane 6 collision
04     MVBP5      Match value for bitplane 5 collision
```

```
               03    MVBP4      Match value for bitplane 4 collision
               02    MVBP3      Match value for bitplane 3 collision
               01    MVBP2      Match value for bitplane 2 collision
               00    MVBP1      Match value for bitplane 1 collision

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

1.22 A Register Summary / CLXDAT

```
                         Agnus/
                 Read/   Denise/
Register Address Write   Paula         Function
-------- ------- -----   -------       --------
CLXDAT    00E     R        D      Collision data register (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
                     bitplanes.  Playfield 2 is all even-numbered
                     enabled bitplanes

               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
```

1.23 A Register Summary / COLORxx

```
                         Agnus/
                 Read/   Denise/
Register Address Write   Paula         Function
-------- ------- -----   -------       --------
COLORxx   180     W        D      Color table xx
```

There are 32 of these registers (xx=00-31) and they
are sometimes collectively called the "color
palette." They contain 12-bit codes representing
red, green, and blue colors for RGB systems.
One of these registers at a time is selected
(by the BPLxDAT serialized video code)
for presentation at the RGB video output pins.
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
     ----------- ----------- ----------- -----------
RGB  X  X  X  X  R3 R2 R1 R0 G3 G2 G1 G0 B3 B2 B1 B0

B=blue, G=green, R=red,
```

1.24   A Register Summary / COP1LCH, COP1LCL, COP2LCH, COP2LCL

```
                      Agnus/
             Read/    Denise/
Register Address Write Paula         Function
-------- ------- ----- -------       --------
COP1LCH    080     W     A( E ) Copper first location register
                                    (high 3 bits, high 5 bits if ECS)
COP1LCL    082     W      A     Copper first location register
                                    (low 15 bits)
COP2LCH    084     W     A( E ) Copper second location register
                                    (high 3 bits, high 5 bits if ECS)
COP2LCL    086     W      A     Copper second location register
                                    (low 15 bits)
```

These registers contain the jump addresses
described above.

1.25   A Register Summary / COPCON

```
                      Agnus/
             Read/    Denise/
Register Address Write Paula         Function
-------- ------- ----- -------       --------
COPCON    02E     W     A( E ) Copper control register
```

This is a 1-bit register that when set true, allows
the Copper to access the blitter hardware. This
bit is cleared by power-on reset, so that the Copper
cannot access the blitter hardware. See Appendix C
for ECS operation.

```
BIT#    NAME     FUNCTION
----    -------  ------------------
 01     CDANG    Copper danger mode. Allows Copper
                 access to blitter if true.
```

1.26   A Register Summary / COPINS

```
                        Agnus/
                 Read/  Denise/
Register Address Write  Paula          Function
-------- ------- -----  -------        --------
COPINS    08C      W       A     Copper instruction fetch identify
```

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

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

BIT#	MOVE IR1	IR2	WAIT UNTIL IR1	IR2	SKIP IF 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 Copper. When this
                        bit is zero, the Blitter Finished flag
                        must be true (in addition to the rest of
                        the bit comparisons) before the Copper
                        can exit from its wait state or skip
                        over an instruction. Note that the V7
                        comparison cannot be masked.

          The Copper is basically a two-cycle machine that
          requests the bus only during odd memory cycles
          (4 memory cycles per instruction). This prevents
          collisions with display, audio, disk, refresh, and
          sprites, all of which use only even cycles. It
          therefore needs (and has) priority over only the
          blitter and microprocessor.

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

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

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

1.27   A Register Summary / COPJMP1, COPJMP2

                          Agnus/
                   Read/ Denise/
Register Address Write   Paula   Function
-------- ------- -----   -------   --------
COPJMP1   088   S   A   Copper restart at first location
COPJMP2   08A   S   A   Copper restart at second location

          These addresses are strobe addresses. When written
          to, they cause the Copper to jump indirect using
          the address contained in the first or second
          location registers described below. The Copper
          itself can write to these addresses, causing its

```
                        own jump indirect.
```

1.28   A Register Summary / DDFSTOP, DDFSTRT

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- ------- -----  -------        --------
DDFSTOP    094     W       A     Display data fetch stop (horiz. position)
DDFSTRT    092     W       A     Display data fetch start (horiz. position)

          These registers control the horizontal timing of the
          beginning and end of the bitplane DMA display data
          fetch.  The vertical bitplane DMA timing is identical
          to the display windows described above.
          The bitplane modulos are dependent on the bitplane
          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  X  X

           (Always set X bits to 0 to maintain upward compatibility)

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

              DDFSTRT (left edge of display data fetch)
              -----------------------------------------
                    PURPOSE        H8,H7,H6,H5,H4
               ----------------- -------------
               Extra wide (max) * 0  0  1  0  1
               Wide               0  0  1  1  0
               Normal             0  0  1  1  1
               Narrow             0  1  0  0  0

              DDFSTOP (right edge of display data fetch)
              ------------------------------------------
                    PURPOSE        H8,H7,H6,H5,H4
               ------------------ -------------
               Narrow             1  1  0  0  1
               Normal             1  1  0  1  0
               Wide (max)         1  1  0  1  1
```

1.29   A Register Summary / DENISEID

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- ------- -----  -------        --------
```

```
DENISEID   07C        R        D( E ) Chip revision level for Denise
                                       (video out chip)
```

1.30   A Register Summary / DIWHIGH, DIWSTOP, DIWSTRT

```
                                Agnus/
                       Read/    Denise/
Register Address       Write    Paula           Function
-------- -------       -----    -------         --------
DIWHIGH   1E4           W        A,D( E ) Display window -  upper bits for
                                          start, stop

DIWSTOP   090           W          A       Display window stop (lower right
                                           vertical-horizontal position)
DIWSTRT   08E           W          A       Display window start (upper left
                                           vertical-horizontal position)
```

These registers control display window size and position
by locating the upper left and lower right corners.

```
BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
     ------------------------------------------------
USE  V7 V6 V5 V4 V3 V2 V1 V0 H7 H6 H5 H4 H3 H2 H1 H0
```

DIWSTRT is vertically restricted to the upper 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).

1.31   A Register Summary / DMACON, DMACONR

```
                                Agnus/
                       Read/    Denise/
Register Address       Write    Paula           Function
-------- -------       -----    -------         --------
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
                     08    BPLEN       Bitplane DMA enable
                     07    COPEN       Copper DMA enable
                     06    BLTEN       Blitter DMA enable
                     05    SPREN       Sprite DMA enable
                     04    DSKEN       Disk DMA enable
                     03    AUD3EN      Audio channel 3 DMA enable
                     02    AUD2EN      Audio channel 2 DMA enable
                     01    AUD1EN      Audio channel 1 DMA enable
                     00    AUD0EN      Audio channel 0 DMA enable
```

1.32   A Register Summary / DSKBYTR

```
                             Agnus/
                     Read/   Denise/
Register Address     Write   Paula          Function
-------- -------     -----   -------        --------
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      Mirror of bit 15 (DMAEN) in  DSKLEN ,
                           ANDed with Bit09 (DMAEN) in  DMACON
          13    DISKWRITE  Mirror of bit 14 (WRITE) in  DSKLEN
          12    WORDEQUAL  This bit true only while the
                            DSKSYNC  register equals the data
                           from disk.
          11-08 X          Not used
          07-00 DATA       Disk byte data
```

1.33   A Register Summary / DSKDAT, DSKDATR

```
                             Agnus/
                     Read/   Denise/
Register Address     Write   Paula          Function
-------- -------     -----   -------        --------
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 two bytes of data that are either sent
> (written) to or received (read) from 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.

1.34   A Register Summary / DSKLEN

```
                         Agnus/
                 Read/   Denise/
Register Address Write   Paula           Function
-------- ------- -----   -------         --------
DSKLEN    024      W        P     Disk length
```

> This register contains the length (number of words)
> of disk DMA data. It also contains two control
> bits, a DMA enable bit, and a DMA direction
> (read/write) bit.

```
BIT#   FUNCTION      DESCRIPTION
----   -----------   -------------------------------
15     DMAEN         Disk DMA enable
14     WRITE         Disk write (RAM to disk) if 1
13-0   LENGTH        Length (# of words) of DMA data.
```

1.35   A Register Summary / DSKPTH, DSKPTL

```
                         Agnus/
                 Read/   Denise/
Register Address Write   Paula           Function
-------- ------- -----   -------         --------
DSKPTH    020      W       A( E ) Disk pointer (high 3 bits,
                                    high 5 bits if ECS)
DSKPTL    022      W       A      Disk pointer (low 15 bits)
```

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

1.36   A Register Summary / DSKSYNC

```
                          Agnus/
                  Read/   Denise/
Register Address  Write   Paula           Function
-------- -------  -----   -------         --------
DSKSYNC   07E       W        P      Disk sync register
```

                holds the match code for disk read synchronization.
                See ADKCON bit 10.

1.37   A Register Summary / HBSTOP, HBSTRT, HCENTER, HSSTOP, HSSTRT, HTOTAL

```
                          Agnus/
                  Read/   Denise/
Register Address  Write   Paula           Function
-------- -------  -----   -------         --------
HBSTOP    1C6       W      A( E ) Horizontal line position for HBLANK stop
HBSTRT    1C4       W      A( E ) Horizontal line position for HBLANK start
HCENTER   1E2       W      A( E ) Horizontal position for Vsync on interlace
HSSTOP    1C2       W      A( E ) Horizontal line position for HSYNC stop
HSSTRT    1DE       W      A( E ) Horizontal sync start (VARHSY)
HTOTAL    1C0       W      A( E ) Highest number count, horiz. line
                                       (VARBEAMEN=1)
```

1.38   A Register Summary / INTENA, INTENAR

```
                          Agnus/
                  Read/   Denise/
Register Address  Write   Paula           Function
-------- -------  -----   -------         --------
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   Copper
         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
```

1.39   A Register Summary / INTREQ, INTREQR

```
                          Agnus/
                   Read/  Denise/
Register Address  Write   Paula         Function
-------- -------  -----   -------       --------
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;
          if enabled by the bits listed in the above register,
          they may cause processor interrupts.  Both a set and
          clear operation are required to load arbitrary data
          into this register.  These status bits are not
          automatically reset when the interrupt is serviced,
          and must be reset when desired by writing to this
          address.  The bit assignments are identical to the
           enable register .
```

1.40   A Register Summary / JOY0DAT, JOY1DAT

```
                          Agnus/
                   Read/  Denise/
Register Address  Write   Paula         Function
-------- -------  -----   -------       --------
JOY0DAT   00A       R        D    Joystick-mouse 0 data (left vertical,
                                     horizontal)
JOY1DAT   00C       R        D    Joystick-mouse 1 data (right vertical,
                                     horizontal)

          These addresses each read a pair of 8-bit mouse
          counters.  0=left controller pair, 1=right
          controller pair (four counters total).  The bit
          usage for both left and right addresses is shown
          below.  Each counter is clocked by signals from
          two controller pins.  Bits 1 and 0 of each counter
          may be read to determine the state of these two
          clock pins.  This allows these pins to double as
          joystick switch inputs.
```

```
                    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
                         ----------------------  ----------------------
                    0DAT Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0  X7 X6 X5 X4 X3 X2 X1 X0
                    1DAT Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0  X7 X6 X5 X4 X3 X2 X1 X0


                    The following table shows the mouse/joystick
                    connector pin usage.  The pins (and their functions)
                    are sampled (multiplexed) into the DENISE chip
                    during the clock times shown in the table.
                    This table is for reference only and should
                    not be needed by the programmer.  (Note that the
                    joystick functions are all "active low" at the
                    connector pins.)


                                                   Sampled by DENISE
                    Conn    Joystick  Mouse        -----------------
                    Pin     Function  Function  Pin  Name  Clock
                    ----    --------  --------  ---  ----  -----
                    L1      FORW*       Y        38   M0V  at CCK
                    L3      LEFT*       YQ       38   M0V  at CCK*
                    L2      BACK*       X         9   M0H  at CCK
                    L4      RIGH*       XQ        9   M0H  at CCK*
                    R1      FORW*       Y        39   M1V  at CCK
                    R3      LEFT*       YQ       39   M1V  at CCK*
                    R2      BACK*       X         8   M1H  at CCK
                    R4      RIGH*       XQ        8   M1H  at CCK*


                    After being sampled, these connector pin signals
                    are used in quadrature to clock the mouse counters.
                    The LEFT and RIGHT joystick functions (active high)
                    are directly available on the Y1 and X1 bits of
                    each counter.  In order to recreate the FORWARD
                    and BACK joystick functions, however, it is
                    necessary to logically combine (exclusive OR)
                    the lower two bits of each counter.
                    This is illustrated in the following table.


                    To detect              Read these counter bits
                    ---------              -----------------------
                    Forward                Y1 xor Y0 (BIT#09 xor BIT#08)
                    Left                   Y1
                    Back                   X1 xor X0 (BIT#01 xor BIT#00)
                    Right                  X1
```

1.41   A Register Summary / JOYTEST

```
                             Agnus/
                     Read/   Denise/
Register Address     Write   Paula           Function
-------- -------     -----   -------         --------
JOYTEST   036          W        D      Write to all four 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
                         ----------------------   ----------------------
                   0DAT  Y7 Y6 Y5 Y4 Y3 Y2 xx xx  X7 X6 X5 X4 X3 X2 xx xx
                   1DAT  Y7 Y6 Y5 Y4 Y3 Y2 xx xx  X7 X6 X5 X4 X3 X2 xx xx
```

1.42   A Register Summary / POT0DAT, POT1DAT

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- -------  -----  -------        --------
POT0DAT   012      R     P ( E ) Pot counter data left pair (vert,horiz.)
POT1DAT   014      R     P ( E ) Pot counter data right pair (vert,horiz.)
```

```
                   These addresses each read a pair of 8-bit pot counters.
                   (Four counters total.) The bit assignment for both
                   addresses is shown below. The counters are stopped by
                   signals from two controller connectors (left-right)
                   with two 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#   Pin Name
                         ---- ---  ---- ----     ----   --------
                         RIGHT  Y   RY    9       36    (POT1Y)
                         RIGHT  X   RX    5       35    (POT1X)
                         LEFT   Y   LY    9       33    (POT0Y)
                         LEFT   X   LX    5       32    (POT0X)
```

1.43   A Register Summary / POTGO, POTGOR

```
                         Agnus/
                  Read/  Denise/
Register Address  Write  Paula          Function
-------- -------  -----  -------        --------
POTGO     034      W      P      Pot port data write and start.

POTGOR    016      R      P      Pot port data read
                                        (formerly called POTINP).
```

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

```
                   BIT#  FUNCT   DESCRIPTION
                   ----  -------------------------------------
                   15    OUTRY   Output enable for Paula pin 36
                   14    DATRY   I/O data Paula pin 36
                   13    OUTRX   Output enable for Paula pin 35
                   12    DATRX   I/O data Paula pin 35
                   11    OUTLY   Output enable for Paula pin 33
                   10    DATLY   I/O data Paula pin 33
                   09    OUTLX   Output enable for Paula pin 32
                   08    DATLX   I/O data Paula pin 32
                   07-01  0      Reserved for chip ID code (presently 0)
                   00    START   Start pots (dump capacitors, start
                                     counters)
```

1.44   A Register Summary / REFPTR

```
                           Agnus/
                    Read/  Denise/
Register Address    Write  Paula          Function
-------- -------    -----  -------        --------
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 microprocessor.

1.45   A Register Summary / SERDAT, SERDATR

```
                           Agnus/
                    Read/  Denise/
Register Address    Write  Paula          Function
-------- -------    -----  -------        --------
SERDAT   030        W      P        Serial port data and stop bits write
                                        (transmit data buffer)
```

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
```

                                    (receive data buffer)

        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# SYM   FUNCTION
        ---- ----   ----------------------
        15   OVRUN   Serial port receiver overrun.
                           Reset by resetting bit 11 of
                            INTREQ .
        14   RBF   Serial port receive buffer full
                           (mirror).
        13   TBE   Serial port transmit buffer
                           empty (mirror).
        12   TSRE   Serial port transmit shift
                           register empty.
                           Reset by loading into buffer.
        11   RXD   RXD pin receives UART serial
                           data for direct bit test by
                           the microprocessor.
        10   0   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

1.46   A Register Summary / SERPER

                          Agnus/
                   Read/ Denise/
Register Address   Write Paula   Function
-------- -------   ----- -------   --------
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# SYM   FUNCTION
        ----- ----   --------------
        15   LONG   Defines serial receive as 9-bit word.
        14-00 RATE   Defines baud rate=1/((N+1)*.2794 microsec.)

1.47   A Register Summary / SPRxCTL, SPRxPOS

```
                              Agnus/
                      Read/   Denise/
Register Address      Write   Paula          Function
-------- -------      -----   -------        --------
SPRxCTL    142          W     A D( E )  Sprite x vert stop position and
                                          control data
SPRxPOS    140          W       A D     Sprite x vert-horiz start position data
```

These two registers work together as position, size and feature sprite-control registers. They are usually loaded by the sprite DMA channel during horizontal blank; however, they may be loaded by either processor at any time. SPRxPOS register:

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

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

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

1.48   A Register Summary / SPRxDATA, SPRxDATB

```
                              Agnus/
                      Read/   Denise/
Register Address      Write   Paula          Function
-------- -------      -----   -------        --------
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 comparison 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 outputted

```
                            whenever the beam counter equals the sprite horizontal
                            position value in the  SPRxPOS  register.
```

1.49   A Register Summary / SPRxPTH, SPRxPTL

```
                              Agnus/
                   Read/      Denise/
Register Address   Write      Paula          Function
-------- -------   -----      -------         --------
SPRxPTH   120        W          A      Sprite x pointer (high 3 bits)
SPRxPTL   122        W          A      Sprite x pointer (low 15 bits)

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

1.50   A Register Summary / STREQU, STRHOR, STRLONG, STRVBL

```
                              Agnus/
                   Read/      Denise/
Register Address   Write      Paula          Function
-------- -------   -----      -------         --------
STREQU    038        S          D      Strobe for horizontal sync with VB
                                             and EQU
STRHOR    03C        S          D P    Strobe for horizontal sync
STRLONG   03E        S          D( E ) Strobe for identification of long
                                             horizontal line

                    One of the first three strobe addresses above is
                    placed on the destination address bus during the
                    first refresh time slot.  The fourth strobe shown
                    above is used during the second refresh time slot of
                    every other line to identify lines with long counts
                    (228).  There are four refresh time slots, and any
                    not used for strobes will leave a null (FF) address
                    on the destination address bus.

STRVBL    03A        S          D      Strobe for horizontal sync with VB
                                             (vertical blank)
```

1.51   A Register Summary / VBSTOP, VBSTRT

```
                              Agnus/
                   Read/      Denise/
Register Address   Write      Paula          Function
-------- -------   -----      -------         --------
VBSTOP    1CE        W         A( E ) Vertical line for VBLANK stop
VBSTRT    1CC        W         A( E ) Vertical line for VBLANK start
```

1.52   A Register Summary / VHPOSR, VHPOSW

```
                            Agnus/
                   Read/    Denise/
Register  Address  Write    Paula          Function
--------  -------  -----    -------        --------
VHPOSR      006      R         A       Read vertical and horizontal position of
                                           beam or lightpen
VHPOSW      02C      W         A       Write vertical and horizontal position
                                           of beam or lightpen

          BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
               ------------------------------------------------
          USE  V7 V6 V5 V4 V3 V2 V1 V0,H8 H7 H6 H5 H4 H3 H2 H1

          RESOLUTION = 1/160 of screen width (280 ns)
```

1.53   A Register Summary / VPOSR, VPOSW

```
                            Agnus/
                   Read/    Denise/
Register  Address  Write    Paula          Function
--------  -------  -----    -------        --------
VPOSR       004      R       A( E ) Read vertical most significant bit
                                       (and frame flop)
VPOSW       02A      W         A     Write vertical most significant bit
                                       (and frame flop)

          BIT# 15,14,13,12,11,10,09,08,07,06,05,04,03,02,01,00
               ------------------------------------------------
          USE  LOF-- -- -- -- -- -- --,-- -- -- -- -- -- -- V8

          LOF=Long frame (auto toggle control bit in  BPLCON0 )
```

1.54   A Register Summary / VSSTOP, VSSTRT, VTOTAL

```
                            Agnus/
                   Read/    Denise/
Register  Address  Write    Paula          Function
--------  -------  -----    -------        --------
VSSTOP     1CA      W       A( E ) Vertical line position for VSYNC stop
VSSTRT     1E0      W       A( E ) Vertical sync start   (VARVSY)
VTOTAL     1C8      W       A( E ) Highest numbered vertical line
                                          (VARBEAMEN=1)
```