```
Nov 11 10:43 1985 pat8000 Page 1


PAL16L8
PAT8000          **-5176-**                    mk 85-05-31
Strobe decoder for X35 video adapter

dir bas prm a23 a22 a21 a19 a11 a09 GND
a08 iorq rd0 wr0 wr1 wr2 crt bds vsp VCC

;equations:

if(VCC) /vsp     = /bas*/prm*/a23* a22*/a21

if(VCC) /crt     = /bas*/prm*/a23* a22*/a21* a19*/a11*/a09* a08

if(VCC) /wr2     = /bas*/prm*/a23* a22*/a21* a19* a11* a09*/a08*/dir*/bds

if(VCC) /wr1     = /bas*/prm*/a23* a22*/a21* a19* a11*/a09* a08*/dir*/bds

if(VCC) /wr0     = /bas*/prm*/a23* a22*/a21* a19* a11*/a09*/a08*/dir*/bds

if(VCC) /rd0     = /bas*/prm*/a23* a22*/a21* a19* a11*/a09*/a08* dir*/bds

if(VCC) /iorq    = /bas*/prm*/a23* a22*/a21* a19*/a11*/a09* a08*/bds


FUNCTION TABLE

bas prm a23 a22 a21 a19 dir a11 a09 a08 bds
vsp crt iorq wr2 wr1 wr0 rd0

;bp aaaa d aaa b  v ci wwwr
;ar 2221 i 100 d  s ro rrrd
;sm 3219 r 198 s  p tr 2100
;                  q
--------------------------
 HX XXXX X XXX X  H HH HHHH
 LH XXXX X XXX X  H HH HHHH
 LL HXXX X XXX X  H HH HHHH
 LL LLXX X XXX X  H HH HHHH
```

```
LL LHHX X XXX X  H HH HHHH
LL LHLL X XXX X  L HH HHHH
LL LHLH X LHX X  L HH HHHH
LL LHLH X LLL X  L HH HHHH
LL LHLH X LLH H  L LH HHHH
LL LHLH X LLH L  L LL HHHH
LL LHLH H HHX X  L HH HHHH
LL LHLH H HLH X  L HH HHHH
LL LHLH H HLL L  L HH HHHL
LL LHLH L HHH L  L HH HHHH
LL LHLH L HHL L  L HH LHHH
LL LHLH L HLH L  L HH HLHH
LL LHLH L HLL L  L HH HHLH
LL LHLH H HXX H  L HH HHHH
```

DESCRIPTION

This device generates all strobes directed to X35 video adapter.

```
Nov 11 10:45 1985 pat8001 Page 1


PAL18R8
PAT8001          **-5176-**                    mk,85-10-02
Sequencer for X35 video adapter

clk rfc pak uds lds a20 a19 a11 vsp GND
 OE dak bds snc das hle va0 lle vdi VCC

;equations:
/snc     :=  snc*/vsp*/a20* rfc* vdi
          + /snc*/vsp*/a20*/a19
          + /snc*/vsp*/a20* a19* a11
          + /snc*/vsp*/a20* vdi
          + /snc*/vsp*/a20*/rfc
/vdi     :=  vdi*/vsp*/a20*/snc* rfc*/uds
          +  vdi*/vsp*/a20*/snc* rfc*/lds
          + /vdi*/vsp*/a20*/a19
          + /vdi*/vsp*/a20* a19* a11
          + /vdi*/vsp*/a20*/snc
          + /vdi*/vsp*/a20*/rfc
          + /vdi*/vsp*/a20*/dak
/va0     :=  va0*/vsp*/a20*/snc*/uds* rfc* das* lle* dak
          + /va0*/vsp*/a20*/rfc
          + /va0*/vsp*/a20*/das
/das     :=  das*/vsp*/a20*/vdi* lle* dak
          +  das*/vsp*/a20*/vdi*/a19*/uds*/lds* va0* hle*/lle*/rfc* dak
          + /das*/vsp*/a20* hle* lle* a19
          + /das*/vsp*/a20*/a19* pak
          + /das*/vsp*/a20*/rfc
/bds     :=  bds*/vsp*/a20*/das* dak
          + /bds*/vsp*/a20* hle* lle* a19
          + /bds*/vsp*/a20*/rfc
          + /bds*/vsp*/a20*/a19* pak
/hle     :=  hle*/vsp*/a20*/a19*/va0* lle* dak
          +  hle*/vsp*/a20*/va0* rfc*/das* a19* a11* dak
          +  hle*/vsp*/a20*/va0* rfc*/bds* a19*/a11*/snc* dak
          + /hle*/vsp*/a20*/a19* pak
          + /hle*/vsp*/a20*/rfc
/lle     :=  lle*/vsp*/a20*/a19*/vdi*/lds* dak
```

```
            +  lle*/vsp*/a20* a19*/uds*/lds*/va0*/hle* rfc* dak
            +  lle*/vsp*/a20* va0* rfc*/das* a19* a11* dak
            +  lle*/vsp*/a20* va0* rfc*/bds* a19*/a11*/snc* dak
            + /lle*/vsp*/a20* snc* a19
            + /lle*/vsp*/a20*/a19*/hle
            + /lle*/vsp*/a20*/a19* pak
            + /lle*/vsp*/a20*/rfc
/dak     :=  dak*/vsp*/a20* a19*/uds* lds*/bds*/hle* rfc
            +  dak*/vsp*/a20*/a19* hle*/lle*/pak* rfc
            +  dak*/vsp*/a20*/a19*/hle* lle*/pak* rfc
            +  dak*/vsp*/a20* a19*/lds*/bds*/lle* rfc
            + /dak*/vsp*/a20

 FUNCTION TABLE
 clk  OE a20 a19 a11 vsp uds lds rfc pak
snc vdi va0 das bds hle lle dak

;c a aa v ul r p  svv db hl d
;102 11 s dd f a  nda ad ll a
;kE0 91 p ss c k  ci0 ss ee k
-----------------------------
CHX XX X XX X X  ZZZ ZZ ZZ Z ;initialize
CLH XX X XX X X  HHH HH HH H
CLX XX H XX X X  HHH HH HH H
CLX XX H XX X L  HHH HH HH H ;crt controller high byte access
CLL HL L XX H X  LHH HH HH H
CLL HL L XX L X  LHH HH HH H
CLL HL L LH H X  LLL HH HH H
CLL HL L LH L X  LLL LH HH H
CLL HL L LH H X  HLL LL HH H
CLL HL L LH L X  HLL LL HH H
CLL HL L LH H X  HHL LL HH H
CLL HL L LH L X  HHL LL HH H
CLL HL L LH H X  LHL LL HH H
CLL HL L LH L X  LHL LL HH H
CLL HL L LH H X  LLL LL LH H
CLL HL L LH L X  LLL LL LH H
CLL HL L LH H X  HLL HH HH L
```

```
    CLL HL L LH L X  HLL HH HH L
    CLL HL L LH H X  HLH HH HH L
    CLX XX H XX L X  HHH HH HH H
    CLL XX L XX H X  LHH HH HH H ;crt controller low byte access
    CLL HL L XX L X  LHH HH HH H
    CLL HL L HL H X  LLH HH HH H
    CLL HL L HL L X  LLH LH HH H
    CLL HL L HL H X  HLH LL HH H
    CLL HL L HL L X  HLH LL HH H
    CLL HL L HL H X  HHH LL HH H
    CLL HL L HL L X  HHH LL HH H
    CLL HL L HL H X  LHH LL HH H
    CLL HL L HL L X  LHH LL HH H
    CLL HL L HL H X  LLH LL HL H
    CLL HL L HL L X  LLH LL HL H
    CLL HL L HL H X  HLH HH HH L
    CLL HL L HL L X  HLH HH HH L
    CLL HL X HL H X  HXH HH HH X
    CLX XX H XX L X  HHH HH HH H
    CLL XX L LL H X  LHH HH HH H ;crt controller word access
    CLL HL L LL L X  LHH HH HH H
    CLL HL L LL H X  LLL HH HH H
    CLL HL L LL L X  LLL LH HH H
    CLL HL L LL H X  HLL LL HH H
    CLL HL L LL L X  HLL LL HH H
    CLL HL L LL H X  HHL LL HH H
    CLL HL L LL L X  HHL LL HH H
    CLL HL L LL H X  LHL LL HH H
    CLL HL L LL L X  LHL LL HH H
    CLL HL L LL H X  LLL LL LH H
    CLL HL L LL L X  LLL LL LH H
    CLL HL L LL H X  HLL HH HL H
    CLL HL L LL L X  HLL HH HL H
    CLL HL L LL H X  HHH HH HL H
    CLL HL L LL L X  HHH HH HL H
    CLL HL L LL H X  LHH HH HL H
    CLL HL L LL L X  LHH HH HL H
    CLL HL L LL H X  LLH HH HH H
```

```
   CLL HL L LL L X  LLH LH HH H
   CLL HL L LL H X  HLH LL HH H
   CLL HL L LL L X  HLH LL HH H
   CLL HL L LL H X  HHH LL HH H
   CLL HL L LL L X  HHH LL HH H
   CLL HL L LL H X  LHH LL HH H
   CLL HL L LL L X  LHH LL HH H
   CLL HL L LL H X  LLH LL HL H
   CLL HL L LL L X  LLH LL HL H
   CLL HL L LL H X  HLH HH HH L
   CLL HL L LL L X  HLH HH HH L
   CLL HL L LL H X  HLH HH HH L
   CLL HL X LL L X  HXH HH HH X
   CLX XX H XX H X  HHH HH HH H
   CLX LX H XX L X  HHH HH HH H ;picture memory high byte access
   CLL LX L XX H X  LHH HH HH H
   CLL LX L XX L X  LHH HH HH H
   CLL LX L LH H X  LLL HH HH H
   CLL LX L LH L H  LLL LH LH H
   CLL LX L LH H H  LLL LL LH H
   CLL LX L LH L H  LLL LL LH H
   CLL LX L LH H H  LLL LL LH H
   CLL LX L LH L L  LLL LL LH H
   CLL LX L LH H X  LLL HH HH L
   CLL LX L LH L X  LLL HH HH L
   CLL LX X XX H X  XXX HH HH X
   CLX XX H HH L X  HHH HH HH H
   CLL LX L XX H X  LHH HH HH H ;picture memory low byte access
   CLL LX L XX L X  LHH HH HH H
   CLL LX L HL H H  LLH HH HH H
   CLL LX L HL L H  LLH LH HL H
   CLL LX L HL H H  LLH LL HL H
   CLL LX L HL L H  LLH LL HL H
   CLL LX L HL H H  LLH LL HL H
   CLL LX L HL L L  LLH LL HL H
   CLL LX L HL H L  LLH HH HH L
   CLL LX X XX L X  XXH HH HH X
   CLX XX H HH H X  HHH HH HH H
```

```
   CLX XX H XX L X  HHH HH HH H ;picture memory word access
   CLL LX L XX H X  LHH HH HH H
   CLL LX L XX L X  LHH HH HH H
   CLL LX L LL H X  LLL HH HH H
   CLL LX L LL L H  LLL LH LL H
   CLL LX L LL H H  LLL LL LL H
   CLL LX L LL L L  LLL LL LL H
   CLL LX L LL H L  LLL HH HL H
   CLL LX L LL L H  LLL HH HL H
   CLL LX L LL H H  LLH HH HL H
   CLL LX L LL L H  LLH LH HL H
   CLL LX L LL H H  LLH LL HL H
   CLL LX L LL L L  LLH LL HL H
   CLL LX L LL H X  LLH HH HH L
   CLL LX L LL L X  LLH HH HH L
   CLL LX X XX H X  XXH HH HH X
   CLX XX H XX L X  HHH HH HH H
   CLL HH L XX H X  LHH HH HH H ;mover high byte access
   CLL HH L XX L X  LHH HH HH H
   CLL HH L LH H X  LLL HH HH H
   CLL HH L LH L X  LLL LH HH H
   CLL HH L LH H X  LLL LL LH H
   CLL HH L LH L X  LLL LL LH H
   CLL HH L LH H X  LLL HH HH L
   CLL HH L LH L X  LLL HH HH L
   CLH HH X XH H X  XXH HH HH X
   CLX XX H XH L X  HHH HH HH H
   CLL HH L XX H X  LHH HH HH H ;mover low byte access
   CLL HH L XX L X  LHH HH HH H
   CLL HH L HL H X  LLH HH HH H
   CLL HH L HL L X  LLH LH HH H
   CLL HH L HL H X  LLH LL HL H
   CLL HH L HL L X  LLH LL HL H
   CLL HH L HL H X  LLH HH HH L
   CLX HH H XX L X  HHH HH HH H
   CLL HH L XX H X  LHH HH HH H ;mover word access
   CLL HH L XX L X  LHH HH HH H
   CLL HH L LL H X  LLL HH HH H
```

```
   CLL HH L LL L X  LLL LH HH H
   CLL HH L LL H X  LLL LL LH H
   CLL HH L LL L X  LLL LL LH H
   CLL HH L LL H X  LLL HH HL H
   CLL HH L LL L X  LLL HH HL H
   CLL HH L LL H X  LLH HH HH H
   CLL HH L LL L X  LLH LH HH H
   CLL HH L LL H X  LLH LL HL H
   CLL HH L LL L X  LLH LL HL H
   CLL HH L LL H X  LLH HH HH L
   CLL HH L LL L X  LLH HH HH L
   CLL HH X XX H X  XXX HH HH X
   CLX HH H XX L X  HHH HH HH H
-------------------------------
```

DESCRIPTION

This device is a state generator for byte and word communication between X37 CPU board and X35 graphic controller. Different pulse trains are generated depending on access to 6845 CRT controller, mover block or picture memory.

```
Dec 16 15:00 1985 pat8003 Page 1


PAL16R8
PAT8003  **-Luxor X37 CPU board-** mk,85-05-29
SCC decoder and clock multiplexor control

clk ba2 zwr d07 scc ba5 ba4 ba3 rst GND
 OE ce0 c0a c1b ce1 c1a c2b ce2 c2a VCC
;equations:

/ce0 :=  rst*/scc*/ba5*/ba4*/ba3

/ce1 :=  rst*/scc*/ba5* ba4*/ba3

/ce2 :=  rst*/scc* ba5*/ba4*/ba3

/c0a := /rst
      + /scc*/ba5*/ba4* ba3* ba2*/zwr*/d07
      + /c0a* scc
      + /c0a* zwr
      + /c0a* ba5
      + /c0a* ba4
      + /c0a*/ba3
      + /c0a*/ba2


 /c1b := /rst
      + /scc*/ba5* ba4* ba3*/ba2*/zwr*/d07
      + /c1b* scc
      + /c1b* zwr
      + /c1b* ba5
      + /c1b*/ba4
      + /c1b*/ba3
      + /c1b* ba2

/c1a := /rst
      + /scc*/ba5* ba4* ba3* ba2*/zwr*/d07
      + /c1a* scc
      + /c1a* zwr
      + /c1a* ba5
```

```
        + /c1a*/ba4
        + /c1a*/ba3
        + /c1a*/ba2

/c2b := /rst
        + /scc* ba5*/ba4* ba3*/ba2*/zwr*/d07
        + /c2b* scc
        + /c2b* zwr
        + /c2b*/ba5
        + /c2b* ba4
        + /c2b*/ba3
        + /c2b* ba2

/c2a := /rst
        + /scc* ba5*/ba4* ba3* ba2*/zwr*/d07
        + /c2a* scc
        + /c2a* zwr
        + /c2a*/ba5
        + /c2a* ba4
        + /c2a*/ba3
        + /c2a*/ba2

FUNCTION TABLE

clk  OE rst scc zwr d07 ba3 ba5 ba4 ba2
ce2 ce1 ce0 c2a c2b c1a c1b c0a

;c r sz d b bbb   ccc ccccc
;l0s cw 0 a aaa   eee 22110
;kEt cr 7 3 542   210 ababa
--------------------------
 CHX XX X X XXX   ZZZ ZZZZZ
 CLL XX X X XXX   HHH LLLLL
 CLH HX X X XXX   HHH LLLLL
 CLH LX X L LLX   HHL LLLLL
 CLH LX X L LHX   HLH LLLLL
```

```
CLH LX X L HLX   LHH LLLLL
CLH LX X L HHX   HHH LLLLL
CLH LL X H LLL   HHH LLLLL
CLH LL H H LLH   HHH LLLLH
CLH LL H H LHL   HHH LLLHH
CLH LL H H LHH   HHH LLHHH
CLH LL H H HLL   HHH LHHHH
CLH LL H H HLH   HHH HHHHH
CLH LL X H HHX   HHH HHHHH
CLH HX X X XXX   HHH HHHHH
CLH XH X X XXX   XXX HHHHH
CLH XX X H XXX   HHH HHHHH
CLH LL L H HLH   HHH LHHHH
CLH LL L H HLL   HHH LLHHH
CLH LL L H LHH   HHH LLLHH
CLH LL L H LHL   HHH LLLLH
CLH LL L H LLH   HHH LLLLL
-------------------------
```

DESCRIPTION:

This device generates three decoded CE signals for SCC circuits on X37 CPU board and five latched clock multiplexor controls for serial adapters.

Inputs: rst - reset
ba(n) - buffered address line n
d07 - data bit 7
scc - access to the SCC device group
zwr - write pulse

Outputs:ce(n) - SCC(n) chip enable
c(n)a - mux control corr. to SCC(n) port a
c(n)b - " " " " " b

```
PAL16L8
PAT8031          **-1110-**              MK, 84-11-08
DS60 MAPPER CONTROL

FC2 A23 A22 A08 A07 DIR BGA CB6 CB7 GND
EAS PTE SWC PWC CXE TPT POE SOE STE VCC

EQUATIONS:

IF(VCC) /PTE = FC2* A23*/A22*/A08* A07* EAS* BGA          ;PAGE ACC.

IF(VCC) /SWC = FC2* A23*/A22*/A08*/A07* EAS*/DIR* BGA     ;SEGMENT WRITE

IF(VCC) /PWC = FC2* A23*/A22*/A08* A07* EAS*/DIR* BGA +   ;PAGE WRITE
               /CB6*/A23*/A22* EAS +                      ;RAM ACC
               /CB6* A23* A22*/A08* EAS                   ;I/O ACCESS

IF(VCC) /CXE =/FC2 +                                      ;USER
               FC2* EAS* A23                              ;SYS CTRL OR I/O

 IF(VCC) /TPT = FC2*/A23*/A22* CB7* CB6 +                 ;RAM ACC LOW LEVEL
               FC2* A23* A22*/A08* CB7* CB6               ;I/O ACC  "    "

IF(VCC) /POE = FC2* A23*/A22*/A08* A07* EAS* DIR* BGA +   ;PAGE READ
              /CB6*/A23*/A22 +                            ;RAM ACCESS
              /CB6* A23* A22*/A08 +                       ;I/O   "
              /CB7* CB6*/A23*/A22 +                       ;RAM ACCESS - TEST
              /CB7* CB6* A23* A22*/A08                    ;I/O   "       "

IF(VCC) /SOE = FC2* A23*/A22*/A08*/A07* EAS* DIR* BGA +   ;SEGMENT READ
               FC2* A23*/A22*/A08* A07 +                  ;PAGE ACCESS
              /FC2 +                                      ;USER MODE
               FC2*/A23*/A22*/CB7 +                       ;RAM - RAM LEVEL
               FC2*/A23*/A22* CB7*/CB6 +                  ;RAM - HIGH BOOT
               FC2* A23* A22*/A08                         ;SYSTEM I/O

 IF(VCC) /STE = FC2* A23*/A22*/A08*/A07* EAS* BGA         ;SYSTEM SEGMENT ACC
```

FUNCTION TABLE:

```
BGA CB7 CB6 FC2 A23 A22 A08 A07 DIR EAS
TPT CXE SOE POE PWC SWC STE PTE

;B CC F AA AA D E   T C S P P S S P
;G BB C 22 00 I A   P X 0 0 W W T T
;A 76 2 32 87 R S   T E E E C C E E
-----------------------------------
;BOOT LOW LEVEL
 X HH H LL XX X X   L H H H H H H H      ;1
 X HH H HH LX X H   L L L H H H H H
 X HH H LH XX X X   H X H H H H H H
 L XX X HL XX X X   X X X X H H H H
 H XX H HL LL L H   H L H H H L L H      ;5
 H XX H HL LL H H   H L L H H H L H
 H XX H HL LH L H   H L L H L H H L
 H XX H HL LH H H   H L L L H H H L
 H XX H HL LX X L   H H X X H H H H
;BOOT HIGH LEVEL
 X HL H LL XX X H   H H L L L H H H     ;10
 X HL H HH LX X H   H L L L L H H H
;RAM AND TEST LEVEL
 X LX X LL XX X H   H X L L L H H H
 X LX X HH LX X H   H X L L L H H H
 X LL L XX XX X H   H L L L L H H H
-----------------------------------
```

DESCRIPTION:

THIS DEVICE DECODES MAPPER ACCESSES AMONG COMMON PROCESSOR ACTIVITY AND GENERATES APPROPRIATE MAPPER CONTROLS.

INPUT SIGNALS:

FC2 - SYSTEM ACCESS (ACTIVE HIGH)

AXX - ADDRESS BITS

```
DIR - TRANSFER TO MAPPER                "    LOW
BGA - DMA ACTIVITY                      "     "
CB6 - CONTROL BIT 6
CB7 -    "      "  7
EAS - EXTENDED ADDRESS STROBE           "    HIGH

OUTPUT SIGNALS:
PTE - PAGE TRANSCEIVER ENABLE           "    LOW
SWC - SEGMENT WRITE CMD                 "     "
PWC - PAGE       "    "                 "     "
CXE - CONTEX ENABLE                     "     "
TPT - TRANSPARENT PAGE TABLE            "     "
POE - PAGE OUTPUT ENABLE                "     "
SOE - SEGMENT OUTPUT ENABLE             "     "
STE - SEGMENT TRANSCEIVER ENABLE        "     "
```

```
PAL16L8
PAT8032          **-1110-**              MK, 84-12-06
DS60 MAIN FUNCTION ENCODER

FC2 FC1 AT1 AT0 A23 A22 A17 A16 A08 GND
CB7 RAM FPP ERR CDE CD2 CD1 CD0 ACK VCC

;EQUATIONS:

IF(VCC) /RAM = /A23*/A22*/CB7 +                                  ;RAM:BOOT OFF
               /A23*/A22* CB7* A16                                ;RAM:BOOT ON

IF(VCC) /FPP =  FC2*/FC1* A23*/A22* A17*/A16* A08                 ;FPP

IF(VCC) /ERR =  FC1* A23*/A22 +                                  ;EXEC MAPPER
                FC1* A23* A22*/A08 +                             ;EXEC I/O
               /FC2*/FC1* A23* A22*/AT1*/A08 +                   ;PROT I/O
               /FC2*/FC1* A23* A22*/AT0*/A08 +                   ;  "
               /FC2* A23*/A22                                    ;USER ACC

IF(VCC) /ACK = /A23*/A22*/CB7 +                                  ;RAM:BOOT OFF
               /A23*/A22* CB7* A16 +                             ;RAM:BOOT ON
                FC2*/FC1* A23*/A22*/A08 +                        ;MAPPER
                FC2*/FC1* A23*/A22*/A17* A16* A08 +               ;EDC
                FC2*/FC1* A23*/A22* A17*/A16* A08                ;FPP

IF(VCC) /CD2 =  FC2* A23* A22* A17* A16* A08 +                   ;INTACK
                FC2*/FC1* A23*/A22* A08                          ;CIO, EDC, DMA

IF(VCC) /CD1 =  FC2* A23* A22* A17* A16* A08 +                   ;INTACK
                FC2*/FC1* A23*/A22*/A17* A16* A08 +               ;EDC
               /FC2*/FC1* A23* A22*/A17*/A16* AT1* AT0*/A08 +     ;SCC GROUP
               /FC2*/FC1* A23* A22* A17* AT1* AT0*/A08 +          ;4680 GROUP
                FC2*/FC1* A23* A22*/A17*/A16*/A08 +               ;SCCG SYS.
                FC2*/FC1* A23* A22* A17*/A08                      ;4680 SYS

IF(VCC) /CD0 =  FC2* A23* A22* A17* A16* A08 +                   ;INTACK
                FC2*/FC1* A23*/A22*/A17*/A16* A08 +               ;CIO
```

```
                /FC2*/FC1* A23* A22*/A17* AT1* AT0*/A08 +      ;SCC, MMEM GROUP
                 FC2*/FC1* A23* A22*/A17*/A08                  ;    "     SYS.

IF(VCC) /CDE = /A23*/A22* CB7*/A16 +                           ;BOOT PROM
                 FC2*/FC1* A23*/A22*/A17* A08 +                ;EDC OR CIO
                 FC2* A23* A22* A17* A16* A08 +                ;INTACK
                 FC2*/FC1* A23*/A22* A17* A16* A08 +           ;DMA
                /FC2*/FC1* A23* A22*/A08* AT1* AT0 +           ;8-BIT I/O
                 FC2*/FC1* A23* A22*/A08                       ;8-BIT I/O SYS.

FUNCTION TABLE:

CB7 FC2 FC1 AT1 AT0 A23 A22 A17 A16 A08
RAM FPP ERR CDE CD2 CD1 CD0 ACK

;C FF AA AA AA A    R F E  C CCC A
;B CC TT 22 11 0    A P R  D DDD C
;7 21 10 32 76 8    M P R  E 210 K
-----------------------------------
 H XX XX LL XH X    L H H  H HHH L    ;RAM - BOOT ON
 L XX XX LL XX X    L H H  H HHH L    ;     "      OFF
 X HL XX HL HL H    H L H  H LHH L    ;FPP
 X XH XX HL XX X    H H L  H HHH H    ;ERR
 X XH XX HH XX L    H H L  H HHH H    ; "
 X LL LX HH XX L    H H L  H HHH H    ; "
 X LL XL HH XX L    H H L  H HHH H    ; "
 X LX XX HL XX X    H H L  H HHH H    ; "
 X HL XX HL XX L    H H H  H HHH L    ;ACK MAPPER
 X HX XX HH HH H    H H H  L LLL H    ;INTACK
 X HL XX HL LH H    H H H  L LLH L    ;EDC
 X HL XX HL LL H    H H H  L LHL H    ;CIO
 X HL XX HL HH H    H H H  L LHH H    ;DMA
 X LL HH HH LL L    H H H  L HLL H    ;SCCG
 X LL HH HH HX L    H H H  L HLH H    ;4680
 X LL HH HH LH L    H H H  L HHL H    ;MMG
 H XX XX LL XL X    H H H  L HHH H    ;BOOT
 X XX XX LH XX X    H H H  H HHH H    ;EXT ACC
 X HL XX HH XX L    H H H  L XXX H    ;SYS I/O ACC
```

DESCRIPTION:

THIS DEVICE PROVIDES DECODED SIGNALS AS WELL AS CODES
FOR MOST SYSTEM OPERATIONS. "RAM","FPP","ERR" AND "ACK"
ARE DECODED SIGNALS, WHILE "CDE","CD2 - CD0" ARE CODES
ADDRESSING EIGHT MAIN SYSTEM DEVICES OR DEVICE GROUPS.

AT1 AND AT0 ARE ATTRIBUTE BITS FROM MAPPER WHICH PROVIDE
FOLLOWING ACCESS PROTECTION:

AT1	AT0	
0	0	READ-ONLY MEMORY ACCESS
0	1	READ/WRITE " "
1	0	NONE " "
1	1	READ/WRITE I/O "

THE LOGICAL ADDRESS SPACE IS DIVIDED INTO FOUR MAIN AREAS
ENCODED BY ADDRESS BITS A23 AND A22. THE AREAS ARE:

A23	A22	
0	0	SYSTEM RAM (ADDRESSED AND PROTECTED BY MAPPER)
0	1	16-BIT EXPANSION (I/O, MEMORY, DMA ETC)
1	0	SYSTEM CONTROL (MAPPER,DMA,CIO,EDC)
1	1	8-BIT I/O DEVICES (PROT. BY MAPPER)

WITHIN SYSTEM CONTROL SPACE, MAPPER IS ACCESSED IF A08=0. IF A08=1,
ACCESS IS DIRECTED TO CIO (A17=0, A16=0), EDC STATUS REGISTER
(A17=0,A16=1) , FLOATING POINT PROCESSOR (A17=1, A16=0) OR DMA (A17=1, A16=1)

WIHTIN 8-BIT I/O DEVICE GROUP, ADDRESS BITS A17 AND A16 ARE USED
FOR DECODING OF **4680** INTERFACE (A17=1), SERIAL COMM. CHANNELS
(A17=0, A16=0) AND MASS MEMORY CHANNELS SUCH AS 5'WINCHESTER INTERFACE,
SASI INTERFACE AND FLOPPY INTERFACE. MASS MEMORY GROUP WILL BE
ADDRESSED BY A17=0, A16=1.

```
DEVICE OUTPUTS CD2-CD0 ENCODE FOLLOWING:
0 - INTERRUPT ACKNOWLEDGE CYCLE
1 - EDC STATUS ACCESS
2 - CIO ACCESS
3 - DMA   "
4 - SCC GROUP ACCESS
5 - 4680      "
6 - MASS MEM.   "
7 - BOOT PROM   "

DEVICE OUTPUT CDE ENABLES OUTPUTS CD2 - CD0, WHEN LOW.
```

```
PAL16R4
PAT8033              **-1110-**              MK,85-02-11
DS60 RAM REFRESH AND MAPPER WRITE CONTROL
  1                                          10
CLK AS' REQ UDS LDS BGA PWC SWC PTE GND
 OE DAS PWL SWR RAS OFF SAS PWH ECK VCC
 11              14  15  16  17          20
;EQUATIONS:

IF(VCC) /PWL   =/PWC*/AS'*/DAS*/PTE* BGA*/LDS*/UDS

        /SWR := SWR*/SWC*/SAS*/DAS* OFF* RAS*/AS'*/LDS* BGA +
                /SWR* ECK* OFF* RAS

        /RAS := RAS*/OFF* ECK*/REQ* SAS +
                /RAS*/REQ*/OFF* SAS +
                /RAS* ECK*/OFF* SAS

        /OFF := OFF*/REQ*/SAS* AS'* ECK* BGA +
                 OFF*/REQ* SAS* AS'*/ECK* BGA +
                /OFF*/REQ* SAS +
                /OFF*/RAS* SAS +
                /OFF*/AS'* BGA* UDS* LDS* SAS +
                /OFF* ECK* SAS

        /SAS := SAS* OFF* RAS*/AS'*/ECK +
                /SAS*/AS'* OFF* RAS +
                /SAS*/ECK* OFF* RAS

IF(VCC) /PWH   =/PWC*/AS'*/DAS*/UDS +
                /PWC*/AS'*/DAS*/LDS

FUNCTION TABLE:

CLK  OE BGA REQ PTE PWC SWC UDS LDS AS' DAS ECK
SAS OFF RAS SWR PWH PWL
```

```
   ------------------------------
   ;C B R PPS ULAD  E  SOR  SPP
   ;LOG E TWW DDSA  C  AFA  WWW
   ;KEA Q ECC SS'S  K  SFS  RHL
   ;------------------------------INITIALIZE
    CLH H HHH HHHH  H  XXX  XHH ;1
    CLH H HHH HHHH  L  HXH  HHH
    CLH H HHH HHHH  H  HXH  HHH
    CLH H HHH HHHH  L  HHH  HHH
    CLH H HHH HHHH  H  HHH  HHH
    CLH H HHH HHHH  L  HHH  HHH
   ;------------------------------PAGE WRITE CMD (TYPE 1)
    CLX H HXH XXXH  H  HHH  HHH
    CLX H HLH LLLH  L  LHH  HHH
    CLX H HLH LLLH  H  LHH  HHH
    CLX H HLH LLLL  L  LHH  HLH ;10
    CLX H HLH LLLL  H  LHH  HLH
    CLH H HXH XXXL  L  LHH  HXH
    CLH H HHH HHHH  H  HHH  HHH
    CLH H HHH HHHH  L  HHH  HHH
   ;------------------------------PAGE WRITE CMD (TYPE 2)
    CLH H XXH XXXH  H  HHH  HHH
    CLH H LLH LLLH  L  LHH  HHH
    CLH H LLH LLLH  H  LHH  HHH
    CLH H LLH LLLL  L  LHH  HLL
    CLH H LLH LLLL  H  LHH  HLL
    CLH H XXH XXXL  L  LHH  HXX ;20
    CLH H HHH HHHH  H  HHH  HHH
    CLH H HHH HHHH  L  HHH  HHH
   ;------------------------------SEGMENT WRITE CMD
    CLH H HHX XXXH  H  HHH  HHH
    CLH H HHL LLLH  L  LHH  HHH
    CLH H HHL LLLH  H  LHH  HHH
    CLH H HHL LLLL  L  LHH  LHH
    CLH H HHL LLLL  H  LHH  LHH
    CLH H HHX XXXL  L  LHH  HHH
    CLH H HHH HHHH  H  HHH  HHH
    CLH H HHH HHHH  L  HHH  HHH ;30
```

```
;----------------------------REFRESH SEQUENCES
 CLH L HHH HHHH  H  HHH  HHH
 CLH L HHH HHHH  L  HLH  HHH
 CLH L HHH XXXH  H  HLL  HHH
 CLH L HHH LLLH  L  HLL  HHH
 CLH H HHH LLLH  H  HLL  HHH
 CLH H HHH LLLH  L  HLH  HHH
 CLH H HHH LLLH  H  HLH  HHH
 CLH H HHH LLLH  L  HHH  HHH
 CLH L HHH LLLH  H  HHH  HHH
 CLH L HHH LLLH  L  LHH  HHH ;40
 CLH L HHH LLLH  H  LHH  HHH
 CLH L HHH XXXL  L  LHH  HHH
 CLH L HHH HHHH  H  HLH  HHH
 CLH L HHH HHHH  L  HLH  HHH
 CLH L HHH HHHH  H  HLL  HHH
 CLH L HHH HHHH  L  HLL  HHH
 CLH H HHH HHXH  H  HLL  HHH
 CLH H HHH HHLH  L  HLH  HHH
 CLH H HHH HHLH  H  HLH  HHH
 CLH H HHH HHLH  L  HLH  HHH ;50
 CLH H HHH XXLH  H  HLH  HHH
 CLH H HHH LLLH  L  HHH  HHH
 CLH H HHH LLLH  H  HHH  HHH
 CLH H HHH LLLH  L  LHH  HHH
 CLH H HHH LLLH  H  LHH  HHH
 CLH H HHH XXXL  L  LHH  HHH
 CLH H HHH HHHH  H  HHH  HHH
 CLL L HHH HHHH  L  HHH  HHH
 CLL L HHH HHHH  H  HHH  HHH
 ----------------------------


A RAM REFRESH CYCLE PROVIDES "OFF"(CPU ADDRESS STROBE OFF)
AND "RAS" PULSES UPON REQUEST INPUT. THIS SEQUENCE IS
SYNCRONIZED WITH CPU ACTIVITY INDICATED BY ADDRESS STROBE
"AS'". SYNCRONIZED ADDRESS STROBE "SAS" IS USED BY ANOTHER
CIRCUITS IN ORDER TO EXTEND CPU CYCLE IN CASE OF CONFLICT
```

BEETWEN CPU OR DMA AND REFRESH.
MAPPER WRITE OUTPUTS SYNCRONIZE WRITE CONTROLS AND PROVIDE
ACCESS PROTECTION.

INPUT SIGNALS:

AS' - CPU OR DMA ADDRESS STROBE ACTIVE LOW
REQ - REFRESH REQUEST " "
UDS - UPPER DATA STROBE " "
LDS - LOW " " " "
BGA - DMA CYCLE " "
PWC - PAGE TABLE WRITE COMMAND " "
SWC - SEGMENT TABLE WRITE COMMAND " "
PTE - PAGE TABLE TRANSCEIVER ENABLED " "
DAS - DELAYED ADDRESS STROBE " "
ECK - EARLY CLOCK

OUTPUT SIGNALS:

PWL - PAGE WRITE LOWER PART " "
SWR - SEGMENT TABLE WRITE " "
RAS - REFRESH RAS STROBE " "
OFF - CPU ADDRESS OFF " "
SAS - SYNCRONIZED ADDRESS STROBE " "
PWH - PAGE WRITE HIGH PART " "

Nov 11 10:49 1985 pat8034 Page 1

```
PAL16R4
PAT8034          **-1110-**                    mk,85-05-29
DS60 CIO, SCC, FDC, and bootprom controller

clk rst lds dir int eck scc cio dly GND
 OE zrd zwr pck zia ba0 cff dta fdc VCC

;equations:

if(VCC)          /zrd :=  dir*/scc*/lds* dly
                        +  dir*/cio*/lds
                        +  dir*/fdc*/lds
                        + /zia*/ba0*/lds* cff
                        + /rst

                 /zwr := /dir*/scc*/cff
                        + /dir*/cio*/cff
                        + /dir*/fdc*/cff
                        + /rst

                 /pck := /rst*/lds
                        +  pck* eck
                        + /pck*/eck

                 /zia :=  rst* zia*/int*/lds* pck* eck
                        +  rst*/zia*/lds
                        +  rst*/zia*/eck

                 /cff :=  rst* cff*/scc*/lds* ba0* pck* eck* dly
                        +  rst* cff*/cio*/lds* ba0* pck* eck
                        +  rst* cff*/fdc*/lds* ba0* pck* eck
                        +  rst* cff*/zia* ba0* pck* eck
                        +  rst*/cff* ba0
                        +  rst*/cff* zia* fdc* pck
                        +  rst*/cff* cio* scc*/pck
                        +  rst*/cff*/zia*/eck

                 /ba0 :=  rst* ba0*/cff* scc* cio* pck* eck
```

```
                        +  rst* ba0*/cff*/scc* eck
                        +  rst* ba0*/cff*/cio* eck
                        +  rst*/ba0*/lds
                        +  rst*/ba0*/eck

if(/ba0)          /dta  = /ba0*/zia* cff* pck*/lds
                        + /ba0*/zia*/lds*/dta
                        + /ba0* zia*/lds

FUNCTION TABLE

clk  OE rst dir lds int cio scc fdc dly eck
pck zia zrd zwr cff ba0 dta

;c r dl icsf d  e p  z zz cb d
;l0s id nicd l  c c  i rw fa t
;kEt rs tocc y  k k  a dr f0 a
-----------------------------
CHL XL XXXX X  L Z  Z LL ZZ X
CLL XL XXXX X  H L  H LL HH Z
CLH XH HHHH X  L L  H HH HH Z
CLH XH HHHH X  H H  H HH HH Z
CLH XX HHHH X  L H  H HH HH Z
CLH XX HHHH X  H L  H HH HH Z
;intack sequence
CLH HL LHHH X  L L  H HH HH Z
CLH HL LHHH X  H H  H HH HH Z
CLH HL LHHH X  L H  H HH HH Z
CLH HL LHHH X  H L  L HH HH Z
CLH HL LHHH X  L L  L HH HH Z
CLH HL LHHH X  H H  L HH HH Z
CLH HL LHHH X  L H  L HH HH Z
CLH HL LHHH X  H L  L HH LH Z
CLH HL LHHH X  L L  L HH LH Z
CLH HL LHHH X  H H  L HH LH Z
CLH HL LHHH X  L H  L HH LH Z
CLH HL LHHH X  H L  L HH LL H
CLH HL LHHH X  L L  L HH LL H
```

```
    CLH HL LHHH X  H H  L HH LL H
    CLH HL LHHH X  L H  L HH LL H
    CLH HL LHHH X  H L  L LH HL H
    CLH HL LHHH X  L L  L LH HL H
    CLH HL LHHH X  H H  L LH HL L
    CLH HL LHHH X  L H  L LH HL L
    CLH HL LHHH X  H L  L LH HL L
    CLH HL LHHH X  L L  L LH HL L
    CLH HX LHHH X  H H  L XH HL X
    CLH HH HHHH X  L H  L HH HL H
    CLH HH HHHH X  H L  H HH HH Z
;fdc read and write seqences
    CLH HH HHHH X  L L  H HH HH Z
    CLH HX HHHH X  H H  H HH HH Z
    CLH HL HHHL X  L H  H LH HH Z
    CLH HL HHHL X  H L  H LH LH Z
    CLH HL HHHL X  L L  H LH LH Z
    CLH HL HHHL X  H H  H LH LH Z
    CLH HL HHHL X  L H  H LH LH Z
    CLH HL HHHL X  H L  H LH LL L
    CLH HL HHHL X  L L  H LH LL L
    CLH HL HHHL X  H H  H LH LL L
    CLH HL HHHL X  L H  H LH HL L
    CLH HX HHHL X  H L  H XH HL X
    CLH HH HHHH X  L L  H HH HL H
    CLH HH HHHH X  H H  H HH HH Z
    CLH HH HHHH X  L H  H HH HH Z
    CLH LH HHHH X  H L  H HH HH Z
    CLH LH HHHL X  L L  H HH HH Z
    CLH LX HHHL X  H H  H HH HH Z
    CLH LL HHHL X  L H  H HH HH Z
    CLH LL HHHL X  H L  H HL LH Z
    CLH LL HHHL X  L L  H HL LH Z
    CLH LL HHHL X  H H  H HL LH Z
    CLH LL HHHL X  L H  H HL LH Z
    CLH LL HHHL X  H L  H HL LL L
    CLH LL HHHL X  L L  H HL LL L
    CLH LL HHHL X  H H  H HL LL L
```

```
   CLH LX HHHL X  L H  H HH HL X
   CLH XH HHHH X  H L  H HH HH Z
 ;scc read and write cycles
   CLH HH HHHH X  L L  H HH HH Z
   CLH HX HHHH L  H H  H HH HH Z
   CLH HL HHLH L  L H  H HH HH Z
   CLH HL HHLH L  H L  H HH HH Z
   CLH HL HHLH L  L L  H HH HH Z
   CLH HL HHLH H  H H  H LH HH Z
   CLH HL HHLH H  L H  H LH HH Z
   CLH HL HHLH H  H L  H LH LH Z
   CLH HL HHLH H  L L  H LH LH Z
   CLH HL HHLH H  H H  H LH LL L
   CLH HL HHLH H  L H  H LH LL L
   CLH HL HHLH H  H L  H LH LL L
   CLH HL HHLH H  L L  H LH HL L
   CLH HX HHLH X  H H  H XH HL X
   CLH HX HHLH X  H H  H XH HL X
   CLH HH HHLH X  L H  H HH HL H
   CLH HH HHHH X  H L  H HH HH Z
   CLH HH HHHH X  L L  H HH HH Z
   CLH LL HHLH L  H H  H HH HH Z
   CLH LL HHLH L  L H  H HH HH Z
   CLH LL HHLH L  H L  H HH HH Z
   CLH LL HHLH H  L L  H HH HH Z
   CLH LL HHLH H  H H  H HH HH Z
   CLH LL HHLH H  L H  H HH HH Z
   CLH LL HHLH H  H L  H HL LH Z
   CLH LL HHLH H  L L  H HL LH Z
   CLH LL HHLH H  H H  H HL LL L
   CLH LL HHLH H  L H  H HL LL L
   CLH LL HHLH H  H L  H HL LL L
   CLH LL HHLH H  L L  H HH HL L
   CLH XH HHXH X  H H  H HH HH Z
 ;cio read and write cycles
   CLH HH HHHH X  L H  H HH HH Z
   CLH HL HLHH X  H L  H LH LH Z
   CLH HL HLHH X  L L  H LH LH Z
   CLH HL HLHH X  H H  H LH LL L
```

```
   CLH HL HLHH X  L H  H LH LL L
   CLH HL HLHH X  H L  H LH LL L
   CLH HL HLHH X  L L  H LH HL L
   CLH HH HXHH X  H H  H HH HH Z
   CLH HH HLHH X  L H  H HH HH Z
   CLH XH HHHH X  H L  H HH HH Z
   CLH LH HHHH X  L L  H HH HH Z
   CLH LX HLHH X  H H  H HH HH Z
   CLH LL HLHH X  L H  H HH HH Z
   CLH LL HLHH X  H L  H HL LH Z
   CLH LL HLHH X  L L  H HL LH Z
   CLH LL HLHH X  H H  H HL LL L
   CLH LL HLHH X  L H  H HL LL L
   CLH LL HLHH X  H L  H HL LL L
   CLH LL HLHH X  L L  H HH HL L
   CLH LL HLHH X  H H  H HH HL L
   CLH XX HLHH X  L H  H HH HL X
   CLH HH HHHH X  H L  H HH HH Z
```

DESCRIPTION:

This device generates RD, WR, and INTACK signals for 8530 SCC and 8536 CIO circuits, RD and WR for FD1797, A0 for bootstrap PROM, and DTACK for the CPU.

```
Nov 11 10:50 1985 pat8035 Page 1

PAL16L8
PAT8035          **-1110-**          MK,85-10-09
DS60 RAM DATA PATH CONTROL

AT1 FAS ES1 LLE ULE ESO DEF UDS LDS GND
ATO UDE LDE RAM DAS AEL DEL DIR BER VCC

;EQUATIONS:

IF(VCC)        /UDE = /RAM*/DAS*/FAS* DIR +
                      /RAM*/UDS* LDS*/ESO* ULE*/DIR +
                      /RAM* UDS* LDS* ULE* LLE*/FAS*/DIR +
                      /RAM*/UDS*/LDS* ULE* LLE*/FAS*/DIR

IF(VCC)        /LDE = /RAM*/DAS*/FAS* DIR +
                      /RAM* UDS*/LDS*/ESO* LLE*/DIR +
                      /RAM* UDS* LDS* ULE* LLE*/FAS*/DIR +
                      /RAM*/UDS*/LDS* ULE* LLE*/FAS*/DIR

IF(VCC)        /AEL =  AT1*/FAS*/RAM*/DAS +
                      /ATO*/DIR*/FAS*/RAM*/DAS

IF(VCC)        /DEL = /DEF*/RAM*/FAS*/DAS +
                      /DEL*/RAM*/DAS

IF(/RAM*/FAS) /BER = /AEL +
                      /DEL

FUNCTION TABLE:

RAM DIR FAS DAS AT1 ATO UDS LDS ESO ULE LLE DEF
UDE LDE AEL DEL BER

;R D FD AA UL E UL D   UL AD B
;A I AA TT DD S LL E   DD EE E
;M R SS 10 SS O EE F   EE LL R
-----------------------------
 H X XX XX XX X XX X   HH HH Z
```

```
   L  X  HH  XX  XX  X  XX  H     HH  HH  Z
   L  H  LH  XX  XX  X  XX  H     HH  HH  H
   L  H  LL  LX  XX  X  XX  H     LL  HH  H
   L  H  LL  LX  XX  X  XX  L     LL  HL  L
   L  H  LL  LX  XX  X  XX  H     LL  HL  L
   L  H  LL  HX  XX  X  XX  H     LL  LL  L
   L  H  LL  HX  XX  X  XX  H     LL  LL  L
   L  H  HL  HX  XX  X  XX  H     HH  HL  Z
   L  H  HH  XX  XX  X  XX  H     HH  HH  Z
   L  L  HH  XX  HH  H  HH  H     HH  HH  Z
   L  L  LH  LH  HH  H  HH  H     LL  HH  H
   L  L  LL  LH  LL  H  HH  H     LL  HH  H
   L  L  LL  LH  LH  H  LL  H     HH  HH  H
   L  L  LL  LH  LH  L  LL  H     HH  HH  H
   L  L  LL  LH  LH  L  HX  H     LH  HH  H
   L  L  LL  LH  HL  H  LL  H     HH  HH  H
   L  L  LL  LH  HL  L  LL  H     HH  HH  H
   L  L  LL  LH  HL  L  XH  H     HL  HH  H
   L  L  LL  LL  HL  L  XL  H     HH  LH  L
   L  L  LL  LL  HL  L  XL  H     HH  LH  L
   L  L  HL  LL  HL  L  XL  H     HH  HH  Z
   L  H  HH  XX  XX  X  XX  X     HH  HH  Z
   ----------------------------
```

DESCRIPTION:

THIS DEVICE CONTROLS DATA TRANSCEIVERS LOCATED
BETWEEN CPU DATA BUS AND MEMORY DATA BUS.
IT PERFORMS ACCESS PROTECTION AS WELL, GENERATING
"BUS ERROR" SIGNAL ON PROHIBITED MEMORY ACCESSES.

INPUT SIGNALS:

FAS - FIXED ADDRESS STROBE (ACTIVE LOW)
DAS - DELAYED " " " "
ES0 - EDAC CONTROL S0 (INVERTED)
ES1 - " " S1
ULE - UPPER MEMORY LATCH ENABLED " "

LLE - LOWER " " " " "
DEF - UNCORRECTABLE ERROR FLAG " "
UDS - UPPER DATA STROBE (FROM CPU) " "
LDS - LOWER " " " " " "
DIR - DIRECTION TO MEMORY " "
RAM - RAM CYCLE " "
AT1 - ATTRIBUTE BIT 1 (EXISTING) " "
AT0 - " " 0 (WRITE PROTECTED) " "

OUTPUT SIGNALS:

UDE - UPPER DATA ENABLE " "
LDE - LOWER " " " " "
AEL - ACCESS ERROR LATCH " "
DEL - DOUBBLE MEMORY ERROR LATCH " "
BER - BUS ERROR " "

```
PAL16R6
PAT8036          **-1110-**          MK,85-04-18
DS60 RAM CONTROL SEQUENCER

CLK ECK FAS SEF ERR RAM LDS UDS DIR GND
 OE CFE LLE ULE ESO CAS HLD ES1 MS1 VCC

;EQUATIONS:

      /LLE:=/RAM*/FAS* ERR*/CAS*/ECK* DIR* ESO +                    ;READ - ASSERT AFTER CAS
            /RAM*/FAS* ERR*/CAS*/ECK*/DIR* ESO*/UDS +               ;BYTE WR - AFTER CAS IF /UDS
            /RAM*/FAS* ERR*/ES1*/ESO* CFE* DIR +                    ;EDC MODE - AT CORR.
            /RAM*/FAS* ERR*/ES1*/ESO* CFE*/DIR*/UDS +               ;EDC MODE - WR. BYTE
            /RAM*/FAS* ERR*/LLE*/ESO +                              ;KEEP TO END OF WRITE
            /RAM*/FAS* ERR*/LLE* ES1* ESO*/UDS +                    ;KEEP TO END OF CYCLE
            /RAM*/FAS* ERR*/LLE* ES1* ESO*/LDS                      ;  "      "      "

      /ULE:=/RAM*/FAS* ERR*/CAS*/ECK* DIR* ESO +                    ;READ - ASSERT AFTER CAS
            /RAM*/FAS* ERR*/CAS*/ECK*/DIR* ESO*/LDS +               ;BYTE WR - AFTER CAS IF /LDS
            /RAM*/FAS* ERR*/ES1*/ESO* CFE* DIR +                    ;EDC MODE - AT CORR.
            /RAM*/FAS* ERR*/ES1*/ESO* CFE*/DIR*/LDS +               ;EDC MODE - WR. BYTE
            /RAM*/FAS* ERR*/ULE*/ESO +                              ;KEEP TO END OF WRITE
            /RAM*/FAS* ERR*/ULE* ES1* ESO*/UDS +                    ;KEEP TO END OF CYCLE
            /RAM*/FAS* ERR*/ULE* ES1* ESO*/LDS                      ;  "      "      "

      /CAS:=/RAM*/FAS* ERR* CAS* ES1* ULE* LLE*/UDS*/ECK* DIR +     ;READ
            /RAM*/FAS* ERR* CAS* ES1* ULE* LLE*/LDS*/ECK* DIR +     ;  "
            /RAM*/FAS* ERR* CAS* ES1* ULE* LLE*/UDS*/ECK*/DIR*/ESO + ;WRITE OR PRERD
            /RAM*/FAS* ERR* CAS* ES1* ULE* LLE*/LDS*/ECK*/DIR*/ESO + ;  "
            /RAM*/FAS* ERR* CAS* ES1*/ESO*/HLD*/ECK +               ;BYTE WR. OR CORR.
            /CAS* ECK                                               ;KEEP ONE CYCLE

      /ESO:=/RAM*/FAS* ERR*/DIR* ULE* LLE* ESO*/ECK* CAS +          ;WR - START
            /RAM*/FAS* ERR*/DIR* ULE* LLE* UDS* LDS*/ECK* CAS +     ;WAIT FOR DATA STROBES
            /RAM*/FAS* ERR*/DIR* ULE* LLE*/UDS*/LDS*/ECK* CAS +     ;WR WORD - CONTINUE
            /RAM*/FAS* ERR*/ESO* ECK +                              ;KEEP ONE CLOCK
            /RAM*/FAS* ERR*/ULE*/HLD*/ECK +                         ;BYTE WR OR CORRECTION
```

```
        /RAM*/FAS* ERR*/LLE*/HLD*/ECK +                                 ; "                "
        /RAM*/FAS* ERR*/ES1* CFE                                        ;DETECTION

  /HLD:=/RAM*/FAS* ERR*/DIR*/ESO*/ECK*/UDS* LDS* ULE* LLE* HLD +        ;BYTE WRITE
        /RAM*/FAS* ERR*/DIR*/ESO*/ECK* UDS*/LDS* ULE* LLE* HLD +        ; "     "
        /RAM*/FAS* ERR*/SEF* HLD*/ES1*/ECK +                            ;CORRECTABLE ERROR
        /RAM*/FAS* ERR*/HLD* ECK +                                      ;KEEP ONE CLOCK
        /RAM*/FAS* ERR*/HLD*/DIR* ESO +                                 ;EXTEND AT BYTE WRITE
        /RAM*/FAS* ERR*/HLD*/DIR*/ES1                                   ;  "          "     "

  /ES1:=/RAM*/FAS* ERR* CFE* ES1* ESO*/CAS*/ECK*/UDS +                  ;DETECTION - AFTER CAS
        /RAM*/FAS* ERR* CFE* ES1* ESO*/CAS*/ECK*/LDS +                  ;   "
        /RAM*/FAS* ERR*/ES1* ECK                                        ;EXTEND ONE CYCLE

IF(VCC)/MS1 =/RAM*/FAS* ERR* CFE* ES1* ESO*/CAS*/ECK*/UDS +             ;DETECTION - AFTER CAS
        /RAM*/FAS* ERR* CFE* ES1* ESO*/CAS*/ECK*/LDS +                  ;   "
        /RAM*/FAS* ERR*/ES1* ECK                                        ;EXTEND ONE CYCLE
```

FUNCTION TABLE:

```
CLK  OE RAM CFE FAS ERR SEF DIR UDS LDS ECK
ULE LLE CAS MS1 ES1 ESO HLD
-----------------------------
;C RCF ES D UL  E  UL C MEE H
;LOAFA RE I DD  C  LL A SSS L
;KEMES RF R SS  K  EE S 110 D
;-----------------------------INITIALIZE
 CHHXX XX X XX  H  ZZ Z XZZ Z ;1
;-----------------------------RD WORD OR LOW BYTE (EDC MODE, CHK OFF)
 CLXLH HH H HH  L  HH H HHH H
 CLLLX HH H XX  H  HH H HHH H
 CLLLL HH H XL  L  HH L HHH H
 CLLLL HH H XL  H  HH L HHH H
 CLLLL HH H XL  L  LL H HHH H
 CLLLL HH H XL  H  LL H HHH H
 CLLLX HH H XX  L  LL H HHH H
 CLXLH HH H HH  H  HH H HHH H
```

```
;------------------------------RD WORD OR HIGH BYTE (EDC MODE, CHK ON)
 CLXHH HH H HH  L  HH H HHH H ;10
 CLXHX HH H XX  H  HH H HHH H
 CLLHL HH H LX  L  HH L LHH H
 CLLHL HH H LX  H  HH L HHH H
 CLLHL HH H LX  L  LL H HLH H
 CLLHL HH H LX  H  HH H LLL H
 CLLHX HH H XX  L  LL H HHL H
 CLXHH HH H HH  H  HH H HHH H
 CLXHH HH H HH  L  HH H HHH H
;------------------------------RD WORD OR BYTE DMA (EDC MODE, CHK ON)
 CLXHX HH H XX  H  HH H HHH H ;20
 CLLHL HH H LL  L  HH L LHH H
 CLLHL HH H LL  H  HH L HHH H
 CLLHL HH H LL  L  LL H HLH H
 CLLHL HH H LL  H  HH H LLL H
 CLLHL HH H LL  L  LL H HHL H
 CLLHL HH H LL  H  LL H HHH H
 CLLHX HH H XX  L  LL H HHH H
 CLXHH HH H HH  H  HH H HHH H ;30
 CLXHH HH H HH  L  HH H HHH H
;------------------------------RD WORD OR BYTE  DMA -  CORRECTION CYCLE
 CLLHX HH H XX  H  HH H HHH H
 CLLHL HH H XL  L  HH L LHH H
 CLLHL HH H XL  H  HH L HHH H
 CLLHL HH H XL  L  LL H HLH H
 CLLHL HH H XL  H  HH H LLL H
 CLLHL HL H XL  L  LL H HHL L
 CLLHL HL H XL  H  LL H HHL L
 CLLHL HH H XL  L  LL L HHL H
 CLLHL HH H XL  H  LL L HHL H ;40
 CLLHL HH H XL  L  LL H HHH H
 CLLHL HH H XL  H  LL H HHH H
 CLLHX HH H XX  L  LL H HHH H
 CLXHH HH H HH  H  HH H HHH H
;------------------------------RD - ACCESS ERROR
```

```
 6
    CLXXH HH H HH  L  HH H HHH H
 8  CLLHX HH H XX  H  HH H HHH H
    CLLHL HH H LX  L  HH L LHH H
10  CLLHL LX H LX  H  HH L HHH H
    CLLHL LX H LX  L  HH H HHH H
12  CLLHL LX H LX  H  HH H HHH H ;50
    CLLHX XX H XX  L  HH H HHH H
14  CLXXH HH H HH  H  HH H HHH H
   ;-----------------------------WRITE WORD (ALL MODES)  CPU CYCLE AFTER REFRESH
16  CLXXH HH H HH  L  HH H HHH H
    CLLXL HH X XX  H  HH H HHH H
18  CLLXL HH L LL  L  HH H HHL H
    CLLXL HH L LL  H  HH H HHL H
20  CLLXL HH L LL  L  HH L HHL H
    CLLXL HH L LL  H  HH L HHL H
22  CLLXX HH L XX  L  HH H HHH H
    CLLXH HH X HH  H  HH H HHH H ;60
24  CLXXH HH H HH  L  HH H HHH H
   ;-----------------------------WRITE WORD (ALL MODES) DMA  CYCLE
26  CLLXX HH X HH  H  HH H HHH H
    CLLXL HH L HH  L  HH H HHL H
28  CLLXL HH L HH  H  HH H HHL H
    CLLXL HH L HH  L  HH H HHL H
30  CLLXL HH L XX  H  HH H HHL H
    CLLXL HH L LL  L  HH L HHL H
32  CLLXL HH L LL  H  HH L HHL H
    CLLXL HH L LL  L  HH H HHH H
34  CLLXL HH L LL  H  HH H HHH H ;70
    CLLXX HH L XX  L  HH H HHL H
36  CLXXH HH X HH  H  HH H HHH H
    CLXXH HH H HH  L  HH H HHH H
38 ;-----------------------------WR BYTE (EDC MODE, CHK OFF) DMA STD
    CLLLX HH X HH  H  HH H HHH H
40  CLLLL HH L HH  L  HH H HHL H
    CLLLL HH L HH  H  HH H HHL H
42  CLLLL HH L HH  L  HH H HHL H
    CLLLL HH L HX  H  HH H HHL H
44  CLLLL HH L HL  L  HH L HHH L
```

```
    CLLLL HH L HL  H  HH L HHH L ;80
    CLLLL HH L HL  L  LH H HHH L
    CLLLL HH L HL  H  LH H HHH L
    CLLLL HH L HL  L  LH H HHL L
    CLLLL HH L HL  H  LH H HHL L
    CLLLL HH L HL  L  LH L HHL H
    CLLLL HH L HL  H  LH L HHL H
    CLLLL HH L HL  L  LH H HHH H
    CLLLL HH L HL  H  LH H HHH H
    CLLLX HH L HX  L  LH H HHH H
    CLXLH HH X aiHH  H  HH H HHH H ;90
;-----------------------------WR BYTE (EDC MODE, CHK ON) CPU AFTER REFRESH
    CLLHH HH H HH  L  HH H HHH H
    CLLHX HH X XH  H  HH H HHH H
    CLLHL HH L LH  L  HH H HHL H
    CLLHL HH L LH  H  HH H HHL H
    CLLHL HH L LH  L  HH L LHH L
    CLLHL HH L LH  H  HH L HHH L
    CLLHL HH L LH  L  HL H HLH L
    CLLHL HX L LH  H  HH H LLL L
    CLLHL HX L LH  L  HL H HHL L
    CLLHL HH L LH  H  HL H HHL L ;100
    CLLHL HH L LH  L  HL L HHL H
    CLLHL HH L LH  H  HL L HHL H
    CLLHX HH L XH  L  HL H HHH H
    CLXHH HH X HH  H  HH H HHH H
;-----------------------------ADDRESS ERROR CYCLE
    CLXXH HH H HH  L  HH H HHH H
    CLLXX HH X HH  H  HH H HHH H
    CLLXL HH L HH  L  HH H HHL H
    CLLXL HH L HH  H  HH H HHL H
    CLLXL HH L HH  L  HH H HHL H
    CLLXL HH L HH  H  HH H HHL H ;110
    CLLXX HH L HH  L  HH H HHL H
    CLXXH HH X HH  H  HH H HHH H
;-----------------------------WRITE ACCESS ERROR CYCLE (WRITE PROTECTED)
    CLXXH HH H HH  L  HH H HHH H
    CLLXX HH X XX  H  HH H HHH H
```

```
 CLLXL HH L LX  L  HH H HHL H
 CLLXL HH L LX  H  HH H HHL H
 CLLXL LH L LX  L  HH H HHH H
 CLLXL LH L LX  H  HH H HHH H
 CLLXX LH L XX  L  HH H HHH H
 CLXXH HH X HH  H  HH H HHH H ;120
-------------------------------
DESCRIPTION:

THIS DEVICE GENERATES CONTROL STROBES FOR MEMORY
LATCHES, CAS AND WRITE STROBES FOR RAM MATRIX, HOLD
SIGNAL FOR  CPU CLOCK CONTROL AND STROBES FOR EDAC
CHECK CIRCUITS.

INPUT SIGNALS:

ECK- EARLY CLOCK
FAS- FORMATTED ADDRESS STROBE              (ACTIVE LOW)
SEF- SINGLE ERROR FLAG                     "     "
ERR- ACCESS ERROR (PROTECT OR PARITY)      "     "
RAM- RAM CYCLE                             "     "
LDS- LOWER DATA STROBE                     "     "
UDS- UPPER   "     "                       "     "
DIR- DIRECTION TO MEMORY                   "     "
CFE- CHECK FUNCTION ON                     "    HIGH

OUTPUT SIGNALS:

LLE- LOWER MEMORY LATCH ENABLE             "    LOW
ULE- UPPER   "      "     "                "     "
ESO- EDAC SO AND MEMORY WRITE STROBE       "     "
CAS- MEMORY CAS STROBE                     "     "
HLD- HOLD CPU CLOCK                        "     "
ES1- EDAC S1 OR PARITY TEST STROBE         "     "
MS1-    "    "      "   MIRROR             "     "
```

```
Nov 11 10:53 1985 pat8037 Page 1

PAL16L8
PAT8037              **-1110-**                MK, 84-12-28
DS60 SASI INTERFACE CONTROL

DMC MMG DIR A13 A07 ADS BRC HLC REQ GND
STB BRQ BWR BHE BLE CNT TRE DLY EXC VCC

;EQUATIONS:

IF(VCC) /BRQ =  REQ*/BRC*/STB +
                REQ* BRC* STB

IF(VCC) /BWR = /DMC* DIR*/ADS +
               /MMG*/A13*/A07*/DIR*/ADS +
                REQ* STB*/BRC*/DLY

IF(VCC) /BHE = /DMC*/ADS +
               /MMG*/A13*/A07*/ADS +
                REQ* STB*/HLC*/BRC*/DLY +
                REQ*/STB*/HLC* BRC

IF(VCC) /BLE = /DMC*/ADS +
               /MMG*/A13*/A07*/ADS +
                REQ* STB* HLC*/BRC*/DLY +
                REQ*/STB* HLC* BRC

IF(VCC) /CNT = /DMC +
               /MMG*/A13*/A07 +
                REQ* HLC* DLY

IF(VCC) /TRE = /DMC*/ADS +
               /MMG*/A13*/A07*/ADS

IF(VCC) /EXC =  REQ*/STB* BRC +
                REQ* STB*/BRC

FUNCTION TABLE:
```

```
STB BRC MMG A13 A07 DIR ADS DMC HLC REQ DLY
BRQ BWR BHE BLE CNT TRE EXC

;SB MAA DA D HRD  B BBB CT E
;TR M10 ID M LEL  R WHL NR X
;BC G37 RS C CQY  Q REE TE C
-------------------------
 XX XXX XX X XLX  H XXX XX H
 LL XXX XX X XHX  L XXX XX H
 LH XXX XX X XHX  H XXX XX L
 HL XXX XX X XHX  H XXX XX L
 HH XXX XX X XHX  L XXX XX H
 XX HXX XX H XLX  X HHH HH H
 XX LHX XH H XLX  X HHH HH H
 XX LLL HL H XLX  X HLL LL H
 XX LLL LL H XLX  X LLL LL H
 XX HXX HL L XXX  X LLL LL X
 XX HXX LL L XXX  X HLL LL X
 HL HXX XX H HHL  H LHL HH L
 HL HXX XX H LHL  H LLH HH L
 LH HXX XX H HHL  H HHL HH L
 LH HXX XX H LHL  H HLH HH L
 XX HXX XX H HHH  X HHX LH X
-------------------------
```

DESCRIPTION:

INPUT SIGNALS:

DMC - DMA CYCLE — ACTIVE LOW
MMG - MASS MEMORY GROUP — " "
DIR - DATA BUS DIRECTION OUT — " "
A13, A07 - CPU ADDRESS BITS
ADS - ADDRESS STROBE — " "
BRC - BUFFER READY CONTROL FF
HLC - HIGH/LOW BYTE CONTROL FF
REQ - SASI REQUEST — " HIGH
STB - SASI TO BUFFER — " "

DLY - ACKNOWLEDGE DELAY " "

OUTPUT SIGNALS:

BRQ - BUFFER REQUESTS TRANSFER " LOW
BWR - " WRITE STROBE " "
BHE - " HIGH BYTE CE " "
BLE - " LOW BYTE CE " "
CNT - ADVANCE BUFFER POINTER " "
TRE - BUS TRANSCEIVER ENABLE " "
EXC - EXECUTE CYCLE (REQUEST RESPOND) " "

```
PAL16L8
PAT8038          **-1110-**          MK,85-05-08
DS60 PARITY GENERATION/DETECTION CONTROL

NC1 NC2 NC3 ES1 ESO MPI ODD NC4 NC5 GND
TST LDR EDR PIN PBL NC6 ER1 MPO ERR VCC

;EQUATIONS:

IF(VCC)      /LDR =  ES1                        ;LATCH DATA

IF(VCC)      /EDR =  ES1*/ESO                   ;WRITE - ENABLE DR TO BUS

IF(VCC)      /PBL = /MPI*/ES1*/TST +            ;PARITY BIT TRANSPARENT
                    /PBL* ES1*/TST +            ;  "      "  LATCHED
                    /MPI*/PBL*/TST              ;NO HAZARD

IF(VCC)      /PIN = /PBL* ES1 +                 ;READ TEST - LATCHED PARITY
                    /ESO*/ES1 +                 ;WRITE - ALLWAYS LOW
                    /MPI*/ES1* ESO              ;NOT WRITE - PREPARE

IF(/ESO)     /MPO = /ODD                        ;WRITE - PARITY BIT TO MEMORY

IF(VCC)      /ERR = /ES1 +                      ;LOW IF NO TEST
                     ES1*/ODD                   ;ERROR IF WRONG PARITY

IF(VCC)      /ER1 = /ES1 +                      ;LOW IF NO TEST
                     ES1*/ODD                   ;ERROR IF WRONG PARITY
FUNCTION TABLE:

TST ES1 ESO MPI ODD LDR EDR PBL PIN MPO ERR ER1

;T EE MO   L E PP M E E
;S SS PD   D D BI P R R
;T 10 ID   R R LN O R 1
------------------------
 H HH HH   L H HH Z H H
 H LH HL   H H HH Z L L
```

```
 H HL XL    L L HH L L L
 H HL XH    L L HH H H H
 L HX HX    L X HH X X X
 L LL LX    H H LL X L L
-------------------------
```

DESCRIPTION:

THIS CIRCUIT PROVIDES CONTROL OF PARITY ADAPTER IN ORDER TO
SIMULATE EDAC COMPATIBILITY.

INPUT SIGNALS:

ES0 - EDAC S0 CONTROL
ES1 - EDAC S1 CONTROL
MPI - MEMORY PARITY BIT (INPUT)
ODD - PARITY GENERATOR OUTPUT
TST - TEST INPUT (FOR TEST PATTERN ONLY)

OUTPUT SIGNALS:

LDR - LATCH DATA REGISTER
EDR - ENABLE DATA REGISTER OUTPUTS
PIN - PARITY GENERATOR INPUT
PBL - LATCHED PARITY BIT FROM MEMORY
MPO - MEMORY PARITY BIT (OUTPUT)
ERR - PARITY ERROR

```
PAL16R4
PAT8039              **-1110-**                MK,85-02-11
DS60 NS32081 FLOATING POINT PROCESSOR INTERFACE

CLK WRT FAS FPP A07 A06 NC1 NC2 NC3 GND
 OE SPC NC4 C01 C00 EPC DON NC5 D00 VCC

;EQUATIONS:

        /C01 := /C00
        /C00 := /FPP*/FAS*/A07* C01

        /EPC := /FPP*/FAS*/WRT*/A07* C01 +
                /FPP*/FAS* WRT*/A07 +
                /EPC*/C00* C01 +
                /EPC*/C00* WRT

        /DON :=  EPC*/SPC +
                 EPC*/DON +
                /WRT*/DON

IF(/EPC) /SPC  = /C00* C01*/WRT +
                 /C00* WRT

IF(/FPP*/FAS* WRT* A07) /D00 = DON

FUNCTION TABLE:

CLK OE FPP WRT A07 FAS SPC C00 C01 EPC DON D00

--------------------
;C  FWA F S  CC ED D
;LO PRO A P  00 P0 0
;KE PT7 S C  01 CN 0
;--------------------INITIALIZE
 CH XXX H Z  ZZ ZZ Z
 CL HXX H X  HX XX Z
 CL HHX H X  HH HX Z
```

```
   CL HHH H H   HH HX Z
;--------------------READ
   CL LHL L L   LH LX Z
   CL LHL L L   LL LH Z
   CL LHL L H   HL LH Z
   CL XHX H H   HH HH Z
;--------------------WRITE
   CL HXX H L   HH HL Z
   CL LLL L L   LH LL Z
   CL LLL L H   LL LL Z
   CL LLL L H   HL HL Z
   CL XXX H H   HH HL Z
;--------------------CATCH SPC AND READ STATUS
   CL XXX H L   HH HL Z
   CL XHX H H   HH HL Z
   CL LHH L H   HH HL H
   CL LHH L H   HH HL H
   CL LHH L H   HH HL H
   CL XHH H H   HH HL Z
;--------------------READ
   CL LHL L L   LH LL Z
   CL LHL L L   LL LH Z
   CL LHL L H   HL LH Z
   CL XHL H H   HH HH Z
;--------------------READ STATUS AGAIN
   CL LHH L H   HH HH L
--------------------
```

DESCRIPTION:

A07 = 1 -> DONE FF ACCESS
A07 = 0 -> FPP REGISTER ACCESS

```
Nov 11 10:56 1985 pat8040 Page 1

PAL16R8
PAT8040              **-5173-**              MK, 85-02-19
DS60 4680 I/O BACKPLANE CONTROL

CLK ACK PHB PHA FCK A06 A04 DIR CYC GND
 OE DBE LA6 STB CSB EXP CSS TRE DTA VCC

;EQUATIONS:

/DBE :=  DBE*/CYC* ACK* FCK* PHB* PHA +
         DBE*/CYC*/ACK* FCK*/PHB*/PHA +
        /DBE*/CYC

/LA6 :=  LA6*/CYC* ACK* FCK*/CSS*/A06 +
        /LA6*/CYC* DBE +
        /LA6*/CYC*/FCK +
         LA6*/CYC*/DBE* FCK* PHB +
        /LA6*/CYC*/DBE*/STB +
        /LA6*/CYC*/DBE*/CSB +
        /LA6*/CYC*/DBE*/EXP +
        /LA6*/CYC*/DBE*/TRE

/STB :=  STB*/CYC* ACK* FCK*/DBE*/LA6* DTA* CSB* EXP +
         STB*/CYC* ACK* FCK*/DBE*/DIR* DTA +
        /STB*/CYC*/FCK +
        /STB*/CYC* DTA +
        /STB*/CYC*/DTA* DIR*/PHB*/PHA

/CSB :=  CSB*/CYC* ACK* FCK*/DBE* LA6*/A04* DTA* DIR* STB* EXP +
        /CSB*/CYC*/FCK +
        /CSB*/CYC* DTA +
        /CSB*/CYC*/DTA* DIR*/PHB*/PHA

/EXP :=  EXP*/CYC* ACK* FCK*/DBE* LA6* A04* DTA* DIR* STB* CSB +
        /EXP*/CYC*/FCK +
        /EXP*/CYC* DTA +
        /EXP*/CYC*/DTA* DIR*/PHB*/PHA
```

```
/CSS :=  CSS*/CYC* ACK* FCK* DBE*/PHB*/PHA +
         /CSS*/CYC*/FCK +
         /CSS*/CYC*/PHB* PHA

/TRE :=  TRE*/CYC*/ACK* FCK* DBE*/PHB*/PHA +
         /TRE*/CYC*/FCK +
         /TRE*/CYC* DTA +
         /TRE*/CYC*/DTA* DIR*/PHB*/PHA

/DTA :=  DTA*/CYC* FCK*/DBE*/LA6* DIR* PHB* PHA +
         DTA*/CYC* FCK*/DBE*/LA6*/DIR*/PHB*/PHA*/STB +
         DTA*/CYC* FCK*/DBE*/LA6*/DIR*/PHB*/PHA*/TRE +
         /DTA*/CYC*/FCK +
         /DTA*/CYC* DIR*/PHB*/PHA +
         /DTA*/CYC*/DIR*/PHB* PHA

FUNCTION TABLE:

CLK  OE CYC ACK DIR A06 A04 PHB PHA FCK
CSS LA6 DBE STB CSB EXP TRE DTA

;C C AD AA PP  F  CLD SCET D
;LOY CI OO HH  C  SAB TSXR T
;KEC KR 64 BA  K  S6E BBPE A
-----------------------------
 CLH XX XX LL  H  XXX XXXX X ;1   INITIALIZE
 CLH XX XX LL  L  HHH HHHH H
 CHH XX XX LL  H  ZZZ ZZZZ Z
;-----------------------------;    READ OPERATION
 CLL HH LX LL  L  HHH HHHH H
 CLL HH LX LL  H  LHH HHHH H ;5
 CLL HH LX LH  L  LHH HHHH H
 CLL HH LX LH  H  LLH HHHH H
 CLL HH LX HL  L  LLH HHHH H
 CLL HH XX HL  H  HLH HHHH H
 CLL HH XX HH  L  HLH HHHH H ;10
 CLL HH XX HH  H  HLL HHHH H
 CLL HH XX LL  L  HLL HHHH H
```

```
    CLL HH XX LL  H  HHL LHHH H
    CLL HH XX LH  L  HHL LHHH H
    CLL HH XX LH  H  HHL LHHH H ;15
    CLL HH XX HL  L  HHL LHHH H
    CLL HH XX HL  H  HLL LHHH H
    CLL HH XX HH  L  HLL LHHH H
    CLL HH XX HH  H  HLL LHHH L
    CLL HH XX LL  L  HLL LHHH L ;20
    CLL HH XX LL  H  HLL LHHH L
    CLL HH XX LH  L  HLL LHHH L
    CLL HH XX LH  H  HLL HHHH H
    CLL HH XX HH  L  HLL HHHH H
    CLH HH XX HH  H  HHH HHHH H ;25
;-------------------------;    WRITE OPERATION
    CLL HL XX LL  L  HHH HHHH H
    CLL HL XX LL  H  LHH HHHH H
    CLL HL XX LH  L  LHH HHHH H
    CLL HL XX LH  H  LXH HHHH H
    CLL HL XX HL  L  LXH HHHH H ;30
    CLL HL XX HL  H  HXH HHHH H
    CLL HL XX HH  L  HXH HHHH H
    CLL HL XX HH  H  HXL HHHH H
    CLL HL XX LL  L  HXL HHHH H
    CLL HL XX LL  H  HHL LHHH H ;35
    CLL HL XX LH  L  HHL LHHH H
    CLL HL XX LH  H  HHL LHHH H
    CLL HL XX HL  L  HHL LHHH H
    CLL HL XX HL  H  HLL LHHH H
    CLL HL XX HH  L  HLL LHHH H ;40
    CLL HL XX HH  H  HLL LHHH H
    CLL HL XX LL  L  HLL LHHH H
    CLL HL XX LL  H  HLL LHHH L
    CLL HL XX LH  L  HLL LHHH L
    CLL HL XX LH  H  HLL HHHH L ;45
    CLL HL XX HL  L  HLL HHHH L
    CLL HL XX HL  H  HHL HHHH H
    CLH HL XX HH  L  HHH HHHH H
;-------------------------;    DMA READ CYCLE
```

```
   CLL LH XX LL  H  HHL HHHL H
   CLL LH XX LH  L  HHL HHHL H ;50
   CLL LH XX LH  H  HHL HHHL H
   CLL LH XX HL  L  HHL HHHL H
   CLL LH XX HL  H  HLL HHHL H
   CLL LH XX HH  L  HLL HHHL H
   CLL LH XX HH  H  HLL HHHL L ;55
   CLL LH XX LL  L  HLL HHHL L
   CLL LH XX LL  H  HLL HHHL L
   CLL LH XX LH  L  HLL HHHL L
   CLL LH XX LH  H  HLL HHHH H
   CLL LH XX HL  L  HLL HHHH H ;60
   CLH XX XX XX  H  HHH HHHH H
;--------------------------;    DMA WRITE CYCLE
   CLL LL XX LL  L  HHH HHHH H
   CLL LL XX LL  H  HHL HHHL H
   CLL LL XX LH  L  HHL HHHL H
   CLL LL XX LH  H  HHL HHHL H ;65
   CLL LL XX HL  L  HHL HHHL H
   CLL LL XX HL  H  HLL HHHL H
   CLL LL XX HH  L  HLL HHHL H
   CLL LL XX HH  H  HLL HHHL H
   CLL LL XX LL  L  HLL HHHL H ;70
   CLL LL XX LL  H  HLL HHHL L
   CLL LL XX LH  L  HLL HHHL L
   CLL LL XX LH  H  HLL HHHH L
   CLL LL XX HL  L  HLL HHHH L
   CLL LL XX HL  H  HHL HHHH H ;75
   CLH XX XX XX  L  HHH HHHH H
;--------------------------;    CSB CYCLE
   CLL HH HL LL  L  HHH HHHH H
   CLL HH HL LL  H  LHH HHHH H
   CLL HH HL LH  L  LHH HHHH H
   CLL HH HL LH  H  LHH HHHH H ;80
   CLL HH HL HL  L  LHH HHHH H
   CLL HH HL HL  H  HHH HHHH H
   CLL HH HL HH  L  HHH HHHH H
   CLL HH HL HH  H  HHL HHHH H
```

```
   CLL HH HL LL  L  HHL HHHH H ;85
   CLL HH HL LL  H  HHL HLHH H
   CLL HH HL LH  L  HHL HLHH H
   CLL HH HL LH  H  HHL HLHH H
   CLL HH HL HL  L  HHL HLHH H
   CLL HH HL HL  H  HLL HLHH H ;90
   CLL HH HL HH  L  HLL HLHH H
   CLL HH HL HH  H  HLL HLHH L
   CLL HH HL LL  L  HLL HLHH L
   CLL HH HL LL  H  HLL HLHH L
   CLL HH HL LH  L  HLL HLHH L ;95
   CLL HH HL LH  H  HLL HHHH H
   CLL HH HL HH  L  HLL HHHH H
   CLH HH HL HH  H  HHH HHHH H
;-------------------------;    EXP CYCLE
   CLL HH HH LL  H  LHH HHHH H
   CLL HH HH LH  L  LHH HHHH H ;100
   CLL HH HH LH  H  LHH HHHH H
   CLL HH HH HL  L  LHH HHHH H
   CLL HH HH HL  H  HHH HHHH H
   CLL HH HH HH  L  HHH HHHH H
   CLL HH HH HH  H  HHL HHHH H ;105
   CLL HH HH LL  L  HHL HHHH H
   CLL HH HH LL  H  HHL HHLH H
   CLL HH HH LH  L  HHL HHLH H
   CLL HH HH LH  H  HHL HHLH H
   CLL HH HH HL  L  HHL HHLH H ;110
   CLL HH HH HL  H  HLL HHLH H
   CLL HH HH HH  L  HLL HHLH H
   CLL HH HH HH  H  HLL HHLH L
   CLL HH HH LL  L  HLL HHLH L
   CLL HH HH LL  H  HLL HHLH L ;115
   CLL HH HH LH  L  HLL HHLH L
   CLL HH HH LH  H  HLL HHHH H
   CLL HH HH HH  L  HLL HHHH H
   CLH HH HL HH  H  HHH HHHH H
---------------------------
```

DESCRIPTION:

THIS DEVICE GENERATES APPROPRIATE STROBE AND GATE TIMING
FOR DS60 4680 I/O INTERFACE

INPUT SIGNALS:

ACK - HD68450 CHANNEL 2 CYCLE ACTIVE LOW
PHB - TIMING GENERATOR PHASE B
PHA - " " " A
FCK - CPU CLOCK
A06 - ADDRESS BIT 6
A04 - ADDRESS BIT 4
DIR - DIRECTION WRITE " "
CYC - 4680 CYCLE " "

OUTPUT SIGNALS:

DBE - DATA BUS ENABLE " "
LA6 - LATCHED ADDRESS 6
STB - INPUT/OUTPUT STROBE " "
CSB - CSB READ STROBE " "
EXP - I/O EXPANDER READ STROBE " "
CSS - CHANNEL SELECT STROBE " "
TRE - DMA TRANSFER ENABLE STROBE " "
DTA - DATA ACKNOWLEDGE RESPONSE " "