```
PAL16R8                                          PAL DESIGN SPECIFICATION
0400M100                                         BOB BRUCE 5-MAY-82
OMEGA 440 MEMORY CYCLE CONTROL PAL 1

CLK DWZ D C QA QB QC QD WEPPL GND
OE COLENL CASPL ROWENL NC1 VSRSO WEPH WEPOL RASPL VCC

/RASPL := /QA * /QB * /C
        + /QA * /QB * /D
        + QA * /QB * /QC
        + QA * /QB * /QD
        + /QA * QB * /QC
        + /QA * QB * /QD
        + QA * QB

/CASPL := QA * /QC * /QD
        + QB * /QD

/ROWENL := /QA * /QB * D
         + QA * QD
         + QB * QD

/COLENL := /QA * /QB * /D
         + QA * /QD
         + QB * /QD

/VSRSO := QD
        + /DWZ
        + DWZ * /QB * /QD
        + DWZ * /QA * QB * /QD

/WEPH := QD
       + /QB * QC
       + /QA * /QB * /QC

/WEPOL := QA * /QC * /QD * /WEPPL
        + QB * /QD * /WEPPL

DESCRIPTION:
```

```
PAL16R4                                 PAL DESIGN SPECIFICATION
G440M200                                BOB BRUCE 24-MAY-82
OMEGA 440 MEMORY CYCLE CONTROL 2

CLK CMAPL PADRL WSRSO WALL QB PROCWL QD BRL GND
OE CLIPH WEPPL PRCLK3 PRCLK1 PRCLK2 PRCLK4 WEENL CASMSKL VCC

IF(VCC) /WEPPL = WALL * /PROCWL * /PADRL * /CLIPH

IF(VCC) /WEENL = /WALL * /PROCWL * /PADRL * /CLIPH

IF(VCC) /CASMSKL = /PROCWL * /PADRL

/PRCLK1 := PADRL * PROCWL * BRL * QB * CMAPL * /WSRSO
         + /QD * /PADRL * BRL * QB

/PRCLK2 := PADRL * PROCWL * BRL * QB * CMAPL * /WSRSO
         + /QD * /PADRL * BRL * QB

/PRCLK3 := PADRL * PROCWL * BRL * QB * CMAPL * /WSRSO
         + /QD * /PADRL * BRL * QB

/PRCLK4 := PADRL * PROCWL * BRL * QB * CMAPL * /WSRSO
         + /QD * /PADRL * BRL * QB

DESCRIPTION:
Revision History:
  jrp 6-1-83    Added @ more proccliks for better buffering. One will be
                used for I/O boards exclusively.
```

```
DISPLAY MODULE MEMORY CONTROLLER
;DISPLAY MODULE MEMORY CONTROL STATE MACHINE
;
;           BOB BRUCE
;          16-DEC-81
;
;THIS PROGRAM REPLACES AN EARLIER VERSION "MEMC1".
;CORRECTS ERRORS IN THE DEFINITION OF A, B, DO, D1
;THEY SHOULD BE SET TO THE VALUE REQUIRED BY THE
;NEXT STATE, NOT THE CURRENT STATE
;
;THIS PROGRAM DEFINES THE STATE TRANSITIONS FOR
;THE DISPLAY MODULE MEMORY CONTROLLER.  THE MEMORY
;CONTROLLER DETERMINES A SEQUENCE OF MEMORY HALF-
;CYCLES TO ALLOW IMAGE MEMORY TO BE ACCESSED FOR
;SCREEN REFRESHES, DYNAMIC RAM REFRESHES AND BIT
;SLICE DRAWING PROCESSOR READS AND WRITES.  THERE
;ARE FIVE INPUT SIGNALS: 1. HSYNCH IS THE HORIZ-
;ONTAL SYNCH SIGNAL.  IT IS USED TO INITIATE REFRESH CYCLES.
;NOTE THAT BLOL CANNOT BE USED FOR THIS PURPOSE BECAUSE
;STAYS LOW DURING VERTICAL BLANKING AND WOULD PREVENT PROPER
;REFRESH.  2.  BLOL IS THE COMPOSITE BLANKING SIGNAL
;WHICH INDICATES WHEN THE CONTROLLER SHOULD ALLOW REFRESH
;AND PROCESSOR ACCESS CYCLES.  3.  RCYLH INDICATES WHEN
;A FULL RAS-CAS CYCLEIS REQUIRED DURING PROCESSOR ACCESS.
;IT IS USED BY THE CONTROLLER TO SELECT RAS-CAS OR PAGE MODE
;CYCLES.  4.  NOZOOMH INDICATES THAT A 1X HORIZONTAL ZOOM
;FACTOR IS IN USE.  THIS ALLOWS THE CONTROLLER TO ELIMINATE
;DYNAMIC RAM REFRESH CYCLES AND THUS IMPROVES DRAWING SPEED.
;5.  PWRUPL INDICATES THAT A POWERUP INIT HAS OCCURRED.
;UNTIL THIS SIGNAL IS CLEARED BY THE DRAWING PROCESSOR, THE
;MEMORY CONTROLLER ALLOWS ONLY PROCESSOR RAS-CAS CYCLES TO
;AVOID HANGING OR ERRATIC OPERATION AT POWERUP BEFORE THE
;CRT CONTROLLER HAS BEEN INITIALIZED AND HSYNCH AND BLOL
;MAY BE INVALID.
        ;
        ;
        ;INPUT SIGNAL DEFINITIONS
        HSYNCH, BLOL, RCYLH, NOZMH, PWRUPL
        ;
        ;OUTPUT SIGNAL DEFINITIONS
        A, B, DO, D1
        ;
        ;STATE DEFINITIONS
        SCRAS, SCCAS, PRAS0, PCAS0, RRAS0, RCK0, RRAS1, &
        RCK1, RRAS2, RCK2, PRAS1, PCAS1, PCAS2, WAIT, &
        PURAS, PUCAS
        ;
        ;
        ;BEGIN WITH STATE 0 TRANSITIONS
        ;SCRAS ALWAYS FOLLOWED BY SCCAS
        SCRAS, PWRUPL=X> SCCAS, A=1, B=0, D1=0, DO=0
        ;
        ;SCCAS STATE HAS THREE POSSIBLE TRANSITIONS:
        ;TO SCRAS, TO PRAS0 AND TO PURAS
        SCCAS, BLOL=1, PWRUPL=1> SCRAS, A=1, B=0, D1=1, DO=1
        ;
        SCCAS, BLOL=0, PWRUPL=1> PRAS0, A=0, B=0, D1=1, DO=1
        ;
        SCCAS, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
        ;
        ;PRAS0 HAS TWO POSSIBLE TRANSITIONS: ONE TO PCAS0 AND
        ;ONE TO PCAS2
        PRAS0, BLOL=0> PCAS0, A=0, B=0, D1=0, DO=0
        ;
        PRAS0, BLOL=1> PCAS2, A=0, B=0, D1=0, DO=0
```

```
;
;PCAS0 HAS SIX TRANSITIONS: TO PCAS2, TO WAIT,
;TO RRAS0, TO PURAS, TO PRAS0 AND TO ITSELF
;NOTE THAT THERE ARE ACTUALLY TWO SEPARATE
;CONDITIONS FOR THE TRANSITIONS TO ITSELF AND
;TO PRAS0: ONE FOR HSYNCH=0 AND ONE FOR NOZOOM=1
;WITH HSYNCH=X
PCAS0, BLOL=1, RCYLH=0, PWRUPL=1> PCAS2, A=0, B=0, &
D1=0, DO=1
;
PCAS0, BLOL=1, RCYLH=1, PWRUPL=1> WAIT, A=0, B=0, &
D1=1, DO=0
;
PCAS0, BLOL=0, HSYNCH=1, NOZMH=0, PWRUPL=1> RRAS0, &
A=1, B=1, D1=1, DO=1
;
PCAS0, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
PCAS0, BLOL=0, HSYNCH=0, RCYLH=1, PWRUPL=1> PRAS0, A=0, &
B=0, D1=1, DO=1
;
PCAS0, BLOL=0, HSYNCH=1, NOZMH=1, RCYLH=1, PWRUPL=1> &
PRAS0, A=0, B=0, D1=1, DO=1
;
PCAS0, BLOL=0, HSYNCH=0, RCYLH=0, PWRUPL=1> PCAS0, A=0, &
B=0, D1=0, DO=1
;
PCAS0, BLOL=0, HSYNCH=1, NOZMH=1, RCYLH=0, PWRUPL=1> &
PCAS0, A=0, B=0, D1=0, DO=1
;
;RRAS0 HAS AN UNCONDITIONAL TRANSITION TO RCK0
RRAS0, PWRUPL=X> RCK0, A=0, B=1, D1=1, DO=0
;
;RCK0 HAS TWO TRANSITIONS: ONE TO RRAS1 AND
;ONE TO PURAS
RCK0, PWRUPL=1> RRAS1, A=1, B=1, D1=1, DO=1
;
RCK0, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;RRAS1 HAS AN UNCONDITIONAL TRANSITION TO RCK1
RRAS1, PWRUPL=X> RCK1, A=0, B=1, D1=1, DO=0
;
;RCK1 HAS TWO TRANSITIONS: ONE TO RRAS2 AND
;ONE TO PURAS
RCK1, PWRUPL=1> RRAS2, A=1, B=1, D1=1, DO=1
;
RCK1, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;RRAS2 HAS AN UNCONDITIONAL TRANSITION  TO RCK2
RRAS2, PWRUPL=X> RCK2, A=0, B=1, D1=1, DO=0
;
;RCK2 HAS TWO TRANSITIONS: ONE TO PRAS1 AND
;ONE TO PURAS
RCK2, PWRUPL=1> PRAS1, A=0, B=0, D1=1, DO=1
;
RCK2, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;PRAS1 HAS AN UNCONDITIONAL TRANSITION TO PCAS1
PRAS1, PWRUPL=X> PCAS1, A=0, B=0, D1=0, DO=0
;
;PCAS1 HAS FIVE TRANSITIONS: ONE TO PRAS1,
;ONE TO ITSELF, ONE TO PRAS0, ONE TO PCAS0 AND
;ONE TO PURAS
PCAS1, PWRUPL=1, HSYNCH=1, RCYLH=1> PRAS1, A=0, B=0, &
D1=1, DO=1
;
```

```
PCAS1, PWRUPL=1, HSYNCH=1, RCYLH=0> PCAS1, A=0, B=0, &
D1=0, DO=1
;
PCAS1, PWRUPL=1, HSYNCH=0, RCYLH=1> PRASO, A=0, B=0, &
D1=1, DO=1
;
PCAS1, PWRUPL=1, HSYNCH=0, RCYLH=0> PCASO, A=0, B=0, &
D1=0, DO=1
;
PCAS1, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;PCAS2 HAS TWO TRANSITIONS: ONE TO SCRAS AND ONE
;TO PURAS
PCAS2, PWRUPL=1> SCRAS, A=1, B=0, D1=1, DO=1
;
PCAS2, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;WAIT HAS TWO TRANSITIONS: ONE TO SCRAS AND
;ONE TO PURAS
WAIT, PWRUPL=1> SCRAS, A=1, B=0, D1=1, DO=1
;
WAIT, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
;PURAS HAS ONE TRANSITION: TO PUCAS
PURAS, PWRUPL=X> PUCAS, A=0, B=0, D1=0, DO=0
;
;PUCAS HAS TWO TRANSITIONS: ONE TO PURAS
;AND ONE TO PRASO
PUCAS, PWRUPL=0> PURAS, A=0, B=0, D1=1, DO=1
;
PUCAS, PWRUPL=1> PRASO, A=0, B=0, D1=1, DO=1
;
;END OF PROGRAM
```

```
PAL16L8                        PAL DESIGN SPECIFICATION
G400BR00
OMEGA 440 MICROCODE BRANCH CONTROL
BOB BRUCE MAY 24, 1982
NC1 CC1S4 CC1 NC2 BS2 BS1 BS0 CC2 CC1S3 GND
NC3 S0 CC2S2 S1 CN OR1 OR0 FE ST VCC

IF(VCC) /S0 = BS1 * /BS2 * CC1 * /CC1S4
            + BS1 * /BS2 * /CC1 * CC1S4
            + /BS1 * BS2 * CC1 * /CC1S4
            + /BS1 * BS2 * /CC1 * CC1S4
            + BS1 * BS2

IF(VCC) /S1 = /BS0
            + BS1 * CC1 * /CC1S4
            + BS1 * /CC1 * CC1S4
            + BS2 * CC1 * /CC1S4
            + BS2 * /CC1 * CC1S4

IF(VCC) /FE = /BS1 * BS2 * CC1 * CC1S4
            + /BS1 * BS2 * /CC1 * /CC1S4
            + BS0 * BS1 * BS2 * CC1 * CC1S4
            + BS0 * BS1 * BS2 * /CC1 * /CC1S4

IF(VCC) /OR0 = BS1
             + BS2
             + CC1 * /CC1S4
             + /CC1 * CC1S4

IF(VCC) /OR1 = BS1
             + BS2
             + CC2 * /CC2S2
             + /CC2 * CC2S2

IF(VCC) /CN  = /BS0 * BS0

IF(VCC) /ST = CC1S3

DESCRIPTION:
This PAL decodes branch instructions from the microcode
into the appropriate control signals for the 2909-11
microsequencer.  The three BS inputs provide eight instructions:
        0       fourway branch using R
        1       fourway branch using D
        2       conditional branch using R
        3       conditional branch using D
        4       conditional branch to subroutine using R
        5       conditional branch to subroutine using D
        7       return from subroutine
```