SOURCE FILE / PAL EQUATIONS

```
#TITLE     'PNCPAL1.SRC';
#ENGINEER  'Bill Sisk';
#COMPANY   'Arecibo Observatory';
#REVISION  '0';
#PROJECT   'IP Interface, Shift Enable Counter, Memory Address Counter';
#COMMENT   '10/6/98';

INPUT multclk;
INPUT ip_clk;
INPUT wrt_inreg;
LOW_TRUE INPUT iosel;
LOW_TRUE INPUT idsel;
LOW_TRUE D_FLOP OUTPUT ack       PRESET_BY ip_reset CLOCKED_BY ip_clk;
INPUT ad3..ad1;
INPUT rw;
LOW_TRUE INPUT ip_reset;
BIPUT ip_d15..ip_d0                ENABLED_BY read;

D_FLOP OUTPUT w_ramd               RESET_BY ip_reset  CLOCKED_BY ip_clk;
LOW_TRUE D_FLOP OUTPUT we_ram      PRESET_BY ip_reset CLOCKED_BY ip_clk;
D_FLOP NODE w_com_wd               RESET_BY ip_reset  CLOCKED_BY ip_clk;
D_FLOP NODE w_se_len               RESET_BY ip_reset  CLOCKED_BY ip_clk;
D_FLOP OUTPUT w_cd_len             RESET_BY ip_reset  CLOCKED_BY ip_clk;
D_FLOP NODE w_cd_sa                RESET_BY ip_reset  CLOCKED_BY ip_clk;
D_FLOP NODE read                   RESET_BY ip_reset  CLOCKED_BY ip_clk;
D_FLOP NODE sbits[1..0]                          CLOCKED_BY ip_clk;

D_FLOP OUTPUT ramd_out3..ramd_out0 RESET_BY ip_reset  CLOCKED_BY multclk;
BIPUT ramd_io3..ramd_io0           ENABLED_BY w_ramd;

D_FLOP NODE se_start_im            RESET_BY ip_reset  CLOCKED_BY w_com_wd;
D_FLOP NODE se_stop                PRESET_BY ip_reset CLOCKED_BY w_com_wd;
INPUT trig;

D_FLOP NODE se_reg1                RESET_BY se_stop   CLOCKED_BY multclk;
D_FLOP NODE en_se                  RESET_BY se_stop   CLOCKED_BY multclk;
D_FLOP NODE ai11..ai0                                 CLOCKED_BY w_se_len;
T_FLOP NODE a11..a0                RESET_BY /en_se    CLOCKED_BY multclk;
D_FLOP NODE ld_s                   PRESET_BY /en_se   CLOCKED_BY multclk;
D_FLOP NODE se                     RESET_BY /en_se    CLOCKED_BY multclk;
D_FLOP NODE ld_sa                  PRESET_BY /en_se   CLOCKED_BY multclk;

D_FLOP NODE c_sa15..c_sa0                             CLOCKED_BY w_cd_sa;
D_FLOP NODE ci15..ci0                                 CLOCKED_BY wrt_inreg;

T_FLOP OUTPUT c15..c0              RESET_BY ip_reset  CLOCKED_BY multclk;
D_FLOP NODE cx                     RESET_BY ip_reset  CLOCKED_BY multclk;
D_FLOP NODE dx                     RESET_BY ip_reset  CLOCKED_BY multclk;
D_FLOP NODE ld_cl                  RESET_BY ip_reset  CLOCKED_BY multclk;
NODE dxx;
NODE dxxi;

[ai11..ai0] = [ip_d11..ip_d0];

se.d = ld_s;

ld_s.d = /ld_s * a0 * /+(a11..a1) + /+(ai11..ai0);
```

```
IF ld_s THEN
      [a11.T..a0.T] = [a11..a0] (+) [ai11..ai0];
ELSE
      a0.T = 1;
      a1.T = /a0;
      a2.T = /+(a1..a0);
      a3.T = /+(a2..a0);
      a4.T = /+(a3..a0);
      a5.T = /+(a4..a0);
      a6.T = /+(a5..a0);
      a7.T = /+(a6..a0);
      a8.T = /+(a7..a0);
      a9.T = /+(a8..a0);
      a10.T = /+(a9..a0);
      a11.T = /+(a10..a0);
END IF;

[c_sa15..c_sa0] = [ip_d15..ip_d0];
[ci15..ci0] = [ip_d15..ip_d0];
ld_sa.d = 0;
dxxi = /+(ci15..ci8);
dxx = /+(c15..c8);
dx.d = /ld_cl * /+(c15..c8) + /+(ci15..ci8);

IF ld_sa THEN
        ld_cl.D = /+(c7..c0) * dxx;
ELSIF se THEN
        ld_cl.D = (c0 * /+(c7..c1) * dx)  +  (/+(ci7..ci0) * dxxi);
ELSE
        ld_cl.D = ld_cl;
END IF;


IF ld_sa THEN
        cx.d = /+(c7..c0);
ELSIF se THEN
        cx.d = /ld_cl * c0 * /+(c7..c1) + ld_cl * /+(ci7..ci0);
ELSE
        cx.d = cx;
END IF;


IF ld_sa THEN
      [c7.T..c0.T] = [c7..c0] (+) [c_sa7..c_sa0];
ELSIF (ld_cl * se) THEN
      [c7.T..c0.T] = [c7..c0] (+) [ci7..ci0];
ELSIF (/ld_cl * se) THEN
      c0.T = 1;
      c1.T = /c0;
      c2.T = /+(c1..c0);
      c3.T = /+(c2..c0);
      c4.T = /+(c3..c0);
      c5.T = /+(c4..c0);
      c6.T = /+(c5..c0);
      c7.T = /+(c6..c0);
ELSE
      [c7.T..c0.T] = 0;
END IF;
```

```
IF ld_sa THEN
      [c15.T..c8.T] = [c15..c8] (+) [c_sa15..c_sa8];
ELSIF (ld_cl * se) THEN
      [c15.T..c8.T] = [c15..c8] (+) [ci15..ci8];
ELSIF (/ld_cl * se * cx) THEN
      c8.T = 1;
      c9.T = /c8;
      c10.T = /+(c9..c8);
      c11.T = /+(c10..c8);
      c12.T = /+(c11..c8);
      c13.T = /+(c12..c8);
      c14.T = /+(c13..c8);
      c15.T = /+(c14..c8);
ELSE
      [c15.T..c8.T] = 0;
END IF;

STATE_MACHINE acknowledge
      CLOCKED_BY ip_clk
      STATE_BITS  sbits[1..0]
      RESET_BY ip_reset;
      STATE zero :
            IF (iosel + idsel) THEN
                  GOTO one;
            ELSE
                  GOTO zero;
            END IF;
            ack.d = 0;
      STATE one :
            GOTO two;
            ack.d = 0;
      STATE two :
            IF (iosel + idsel) THEN
                  GOTO two;
            ELSE
                  GOTO zero;
            END IF;
            ack.d = 1;
      ELSE
            ack.d = 0;
            GOTO zero;
END acknowledge;

CASE [ip_d1..ip_d0]
        WHEN 0 =>
                se_start_im.d = se_start_im;
                se_stop.d = se_stop;
        WHEN 1 =>
                se_start_im.d = 0;
                se_stop.d = 1;
        WHEN 2 =>
                se_start_im.d = 0;
                se_stop.d = 0;
        WHEN 3 =>
                se_start_im.d = 1;
                se_stop.d = 0;
END CASE;

se_reg1.d = se_start_im + trig + se_reg1;
en_se.d = se_reg1;
```

```
w_se_len.d    = iosel * /rw * /ad3 * /ad2 * /ad1;
w_cd_len.d    = iosel * /rw * /ad3 * /ad2 *  ad1;
w_cd_sa.d     = iosel * /rw * /ad3 *  ad2 * /ad1;
w_ramd.d      = iosel * /rw * /ad3 *  ad2 *  ad1 + we_ram;
we_ram.d      = iosel * /rw * /ad3 *  ad2 *  ad1;
w_com_wd.d    = iosel * /rw *  ad3 * /ad2 * /ad1;
read.d        = iosel *  rw + read * /ack;

[ramd_io3..ramd_io0] = [ip_d3..ip_d0];
[ramd_out3..ramd_out0] = [ramd_io3..ramd_io0];

CASE [ad3..ad1]
        WHEN 0 =>
            [ip_d11..ip_d0] = [ai11..ai0];
                [ip_d15..ip_d12] = 0;
        WHEN 1 =>
            [ip_d15..ip_d0] = [ci15..ci0];
      WHEN 2 =>
            [ip_d15..ip_d0] = [c_sa15..c_sa0];
      WHEN 3 =>
            [ip_d3..ip_d0] = [ramd_io3..ramd_io0];
                [ip_d15..ip_d4] = 0;
        WHEN 4 =>
            ip_d0 = en_se;
                [ip_d15..ip_d1] = 0;
        WHEN 5 =>
            [ip_d15..ip_d0] = [c15..c0];
        ELSE
                [ip_d15..ip_d0] = 0000h;
END CASE;
```

PONOUT

Pin	Type	Signal	Pin	Type	Signal
1	GND		43	GND	
2	Vcc		44	Vcc	
3	Biput	c8	45	Biput	w_ramd
4	Biput	c9	46	Biput	we_ram
5	Biput	c10	47	Biput	ip_d15
6	Biput	c11	48	Biput	ip_d14
7	Biput	c12	49	Biput	w_cd_len
8	Biput	c13	50	Biput	ip_d13
9	Biput	c14	51	Biput	ack
10	Biput	c15	52	Biput	ip_d12
11	GND		53	GND	
12	Biput	c0	54	Biput	
13	Biput	c1	55	Biput	ip_d11
14	Biput	c2	56	Biput	
15	Biput	c3	57	Biput	ip_d10
16	Biput	c4	58	Biput	
17	Biput	c5	59	Biput	ip_d9
18	Biput	c6	60	Biput	ad3
19	Biput	c7	61	Biput	ip_d8
20	In/CLK	ip_clk	62	In/CLK	wrt_inreg
21	Vcc		63	Vcc	
22	GND		64	GND	
23	In/CLK	multclk	65	In/CLK	
24	Biput	ramd_io3	66	Biput	ad2
25	Biput	ramd_io2	67	Biput	ad1
26	Biput	ramd_io1	68	Biput	ip_d7
27	Biput	ramd_io0	69	Biput	iosel
28	Biput	ramd_out3	70	Biput	ip_d6
29	Biput	ramd_out2	71	Biput	ip_d5
30	Biput	ramd_out1	72	Biput	ip_d4
31	Biput	ramd_out0	73	Biput	idsel
32	GND		74	GND	
33	Biput		75	Biput	ip_d3
34	Biput		76	Biput	rw
35	Biput		77	Biput	ip_d2
36	Biput		78	Biput	ip_reset
37	Biput		79	Biput	ip_d1
38	Biput		80	Biput	ip_d0
39	Biput		81	Biput	
40	Biput		82	Biput	trig
41	Input		83	Input	
42	Vcc		84	Vcc	

WIRELIST

Signal	Device	Pin
multclk	MACH435_1	23
ip_clk	MACH435_1	20
wrt_inreg	MACH435_1	62
iosel	MACH435_1	69
idsel	MACH435_1	73
ack	MACH435_1	51
ad3	MACH435_1	60
ad2	MACH435_1	66
ad1	MACH435_1	67
rw	MACH435_1	76
ip_reset	MACH435_1	78
ip_d15	MACH435_1	47
ip_d14	MACH435_1	48
ip_d13	MACH435_1	50
ip_d12	MACH435_1	52
ip_d11	MACH435_1	55
ip_d10	MACH435_1	57
ip_d9	MACH435_1	59
ip_d8	MACH435_1	61
ip_d7	MACH435_1	68
ip_d6	MACH435_1	70
ip_d5	MACH435_1	71
ip_d4	MACH435_1	72
ip_d3	MACH435_1	75
ip_d2	MACH435_1	77
ip_d1	MACH435_1	79
ip_d0	MACH435_1	80
w_ramd	MACH435_1	45
we_ram	MACH435_1	46
w_cd_len	MACH435_1	49
ramd_out3	MACH435_1	28
ramd_out2	MACH435_1	29
ramd_out1	MACH435_1	30
ramd_out0	MACH435_1	31
ramd_io3	MACH435_1	24
ramd_io2	MACH435_1	25
ramd_io1	MACH435_1	26
ramd_io0	MACH435_1	27
trig	MACH435_1	82
c15	MACH435_1	10
c14	MACH435_1	9
c13	MACH435_1	8
c12	MACH435_1	7
c11	MACH435_1	6
c10	MACH435_1	5
c9	MACH435_1	4
c8	MACH435_1	3
c7	MACH435_1	19
c6	MACH435_1	18
c5	MACH435_1	17
c4	MACH435_1	16

Signal	Device	Pin
c3	MACH435_1	15
c2	MACH435_1	14
c1	MACH435_1	13
c0	MACH435_1	12