10H/10020EV8 high-speed (4.4ns) ECL PLD AN043

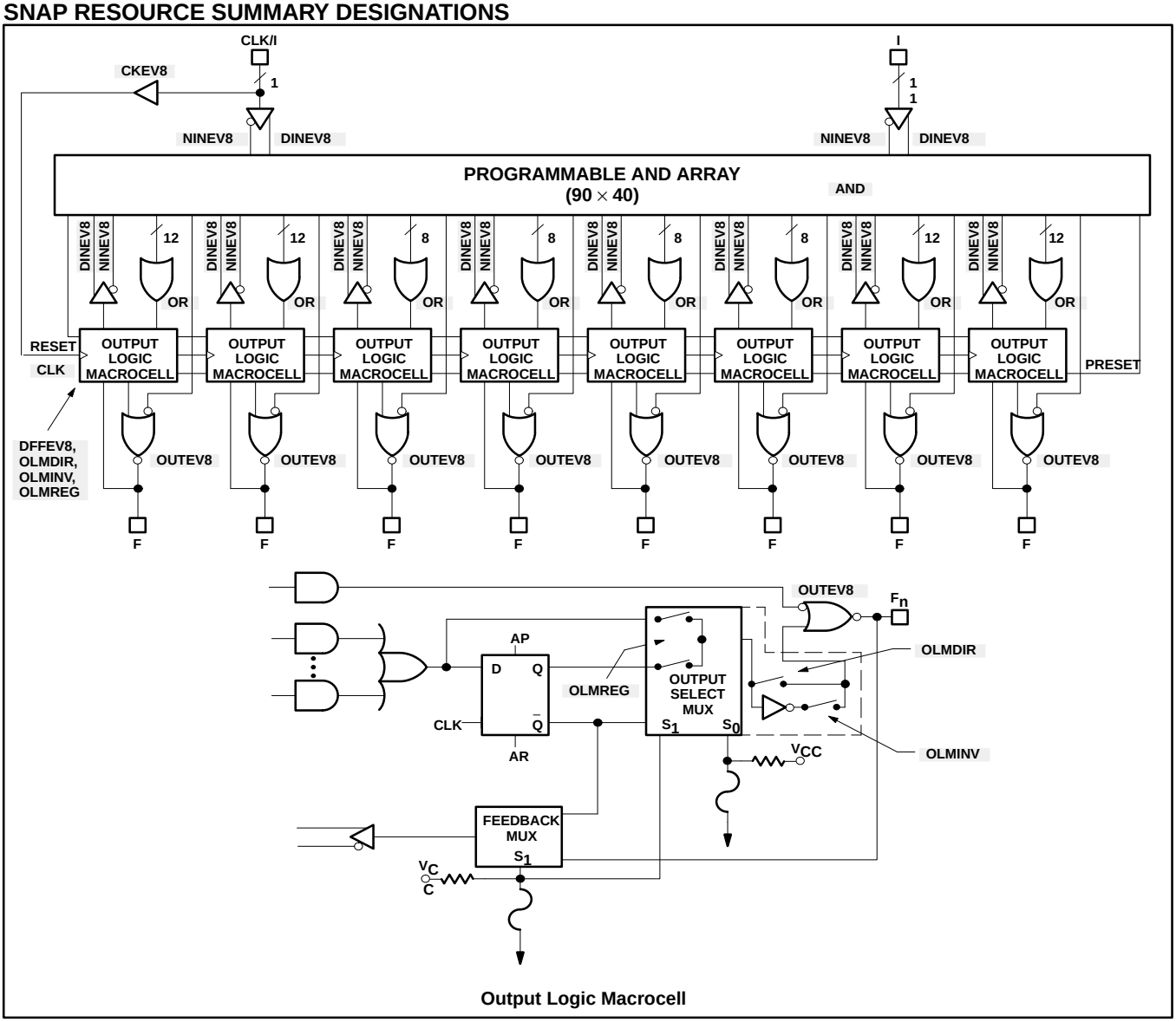
INTRODUCTION

ECL designers have never had enough chips to complete high performance designs. Period. The TTL and CMOS designers always had full MSI and LSI catalogs to rely on, and the ECL guys only had a handful of chips, to solve the hardest technological problems yet! But, that's all changed with the introduction of the Philips Semiconductors PAL™ 20EV8. Here's how.

Philips Semiconductors 10H/10020EV8 PAL devices are two 24-pin PLDs which can be configured to perform a whole catalog of parts in just two ICs. Whether you need to do simple decoding, multiplexing, counting, shifting or form complex state machines, these extraordinary parts can do it all. You can think of them as equivalents to these valuable functions, or as an empty canvas to paint your own catalog on. Don't be frustrated by the nonavailable entries in your ECL catalog, go ahead and roll your own!

Designers don't have time to read lengthy application notes, so this one was created to show you exactly how to make the basic functions you'll probably need. First, there is a single page description of the parts. Then, there is a series of four application briefs showing the exact equations, pinning, internal resource usage and simulation files for the basic mux, decoder, shifter and counter. These design files are created using Philips Semiconductors SNAP and SLICE syntax, which is simple: ∗=AND, +=OR and /=INVERT. These products are also supported on Data I/O ABEL™, which has a similar syntax.

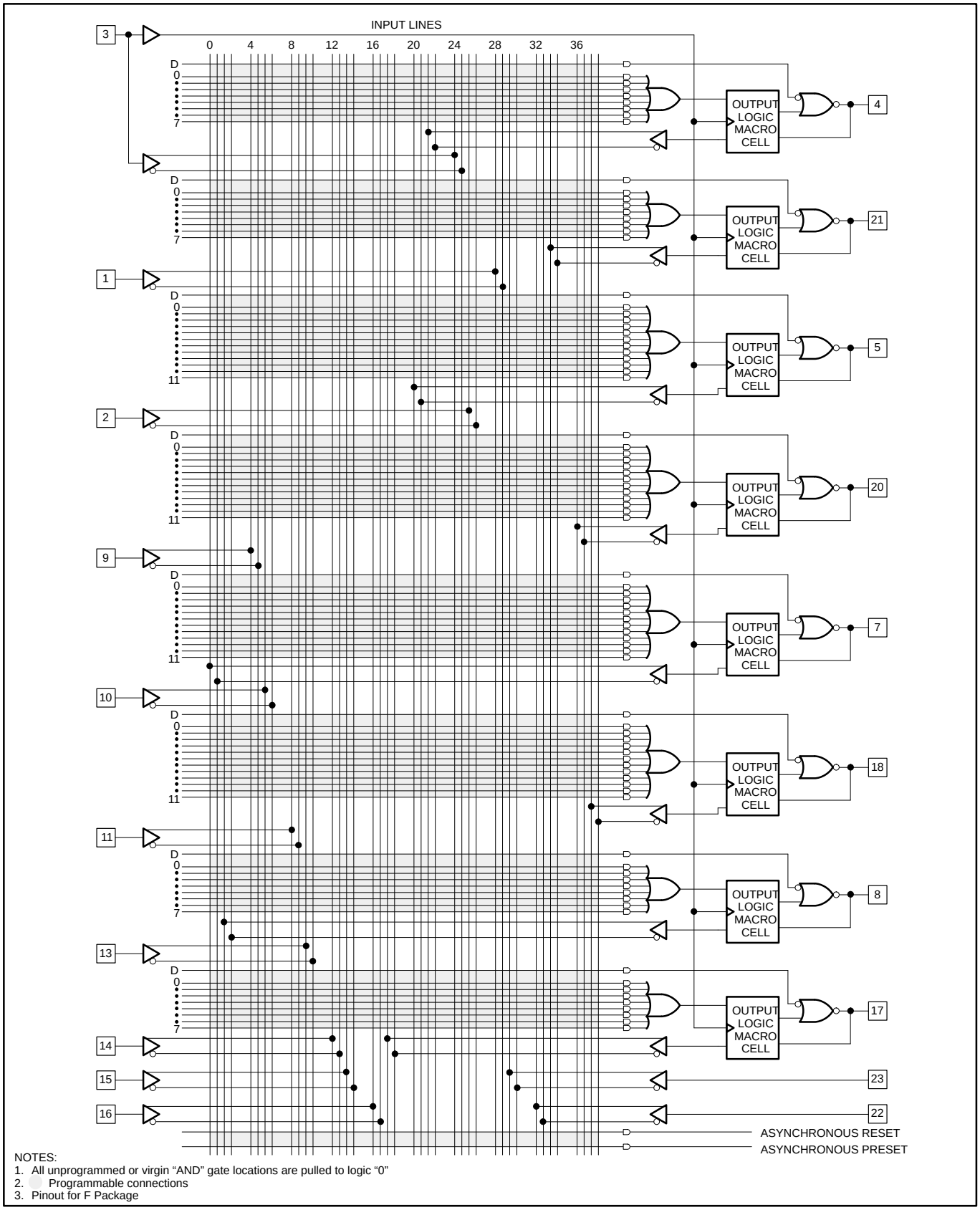
SNAP RESOURCE SUMMARY DESIGNATIONS



Output Logic Macrocell

PAL is a registered trademark of Advanced Micro Devices.
ABEL is a trademark of Data I/O Corporation.

10H/10020EV8 LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin “AND” gate locations are pulled to logic “0”
2. Programmable connections
3. Pinout for F Package

Design Brief #1: Three-To-Eight Decoder Using SNAP/SLICE

```
@PINLIST
A0 I;A1 I;A2 I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4
 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0 =/      (  /A2*  /A1*  /A0);
OUT1 =/      (  /A2*  /A1*   A0);
OUT2 =/      (  /A2*   A1*  /A0);
OUT3 =/      (  /A2*   A1*   A0);
OUT4 =/      (   A2*  /A1*  /A0);
OUT5 =/      (   A2*  /A1*   A0);
OUT6 =/      (   A2*   A1*  /A0);
OUT7 =/      (   A2*   A1*   A0);
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device	=	0X20EV8
Pin1	=	A0
Pin2	=	A1
Pin4	=	OUT0
Pin5	=	OUT1
Pin7	=	OUT2
Pin8	=	OUT3
Pin9	=	A2
Pin17	=	OUT4
Pin18	=	OUT5
Pin20	=	OUT6
Pin21	=	OUT7

DESIGN FROM DECODE.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	0 / 1	0%
DINEV8	3 / 28	10%
NINEV8	3 / 28	10%
AND	16 / 90	17%
OR	8 / 8	100%
OLMDIR	8 / 8	100%
OLMINV	0 / 8	0%
OLMREG	0 / 8	0%
DFFEV8	0 / 8	0%
OUTEV8	8 / 8	100%

Design Brief #1: Three-To-Eight Decoder Using ABEL

```
module_decode;

decode device 'ec20ev8a';

a0,a1,a2             pin 1,2,9;
out0,out1,out2,out3  pin 4,5,7,8;
out4,out5,out6,out7  pin 17,18,20,21

inputs = [a2,a1,a0];
H,L = 1,0;

equations
out0 = ! (inputs==0);
out1 = ! (inputs==1);
out2 = ! (inputs==2);
out3 = ! (inputs==3);
out4 = ! (inputs==4);
out5 = ! (inputs==5);
out6 = ! (inputs==6);
out7 = ! (inputs==7);

test vectors
( [inputs ]  ->   [out0 ,out1 ,out2,  out3  ,out4 ,out5   ,out6  ,out7 ]);
  [   0   ]  ->   [  L  ,  H  ,H   ,  H     ,H    ,H      ,   H   ,  H  ];
  [   1   ]  ->   [  H  ,  L  ,H   ,  H     ,H    ,H      ,   H   ,  H  ];
  [   2   ]  ->   [  H  ,  H  ,L   ,  H     ,H    ,H      ,   H   ,  H  ];
  [   3   ]  ->   [  H  ,  H  ,H   ,  L     ,H    ,H      ,   H   ,  H  ];
  [   4   ]  ->   [  H  ,  H  ,H   ,  H     ,L    ,H      ,   H   ,  H  ];
  [   5   ]  ->   [  H  ,  H  ,H   ,  H     ,H    ,L      ,   H   ,  H  ];
  [   6   ]  ->   [  H  ,  H  ,H   ,  H     ,H    ,H      ,   L   ,  H  ];
  [   7   ]  ->   [  H  ,  H  ,H   ,  H     ,H    ,H      ,   H   ,  L  ];
  [   0   ]  ->   [  L  ,  H  ,H   ,  H     ,H    ,H      ,   H   ,  H  ];

end
```

Design Brief #2: Dual Four-to-One MUX with SNAP/SLICE

```
@PINLIST
A0 I;A1 I;B0 I;B1 I;C0 I;C1 I;D0 I;D1 I;SEL I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0 =    A0*SEL   +   A1*/SEL;      "MUX A0–D0 ON SEL, A1–D1 ON /SEL"
OUT1 =    B0*SEL   +   B1*/SEL;
OUT2 =    C0*SEL   +   C1*/SEL;
OUT3 =    D0*SEL   +   D1*/SEL;
OUT4 =    A1*SEL   +   A0*/SEL;      "DATA SWAP FROM ABOVE METHOD"
OUT5 =    B1*SEL   +   B0*/SEL;
OUT6 =    C1*SEL   +   C0*/SEL;
OUT7 =    D1*SEL   +   D0*/SEL;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device	=	0X20EV8
Pin1	=	A0
Pin2	=	A1
Pin4	=	OUT0
Pin5	=	OUT1
Pin7	=	OUT2
Pin8	=	OUT3
Pin9	=	B0
Pin10	=	B1
Pin11	=	C0
Pin13	=	C1
Pin14	=	D0
Pin15	=	D1
Pin16	=	SEL
Pin17	=	OUT4
Pin18	=	OUT5
Pin20	=	OUT6
Pin21	=	OUT7

DESIGN FROM MUX.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	0 / 1	0%
DINEV8	9 / 28	32%
NINEV8	1 / 28	3%
AND	24 / 90	26%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	8 / 8	100%
OLMREG	0 / 8	0%
DFFEV8	0 / 8	0%
OUTEV8	8 / 8	100%

Design Brief #2: Dual Four-to-One MUX Using ABEL

```
module_mux;

mux device 'ec20ev8a';

sel                          pin 16;
a0,b0,c0.d0                  pin 1,9,11,14;
a1,b1,c1,d1                  pin 2,10,13,15;
out0,out1,out2,out3          pin 4,5,7,8;
out4,out5,out6,out7          pin 17,18,20,21;


"mux a0-d0 to out0-out3 on sel==1
"mux a1-d1 to out4-out7 on sel==1
"mux a0-d0 to out4-out7 on sel==0
"mux a1-d1 to out0-out3 on sel==0

outl = [out0..out3];
outh = [out4..out7];
inl  = [a0,b0,c0,d0];
inh  = [a1,b1,c1,d1];



equations

when (sel==1) then outl=inl;
when (sel==1) then outh=inh;
when (sel==0) then outl=inh;
when (sel==0) then outh=inl;

test_vectors
( [sel,inl,inh  ]   ->    [outl,outh]);
  [1, ^h0 ,^hf  ]   ->    [^h0 , ^hf];
  [1, ^ha ,^h5  ]   ->    [^ha ,  ^h5];
  [0, ^h0 ,^hf  ]   ->    [^h0 ,  ^hf];
  [0, ^ha ,^h5  ]   ->    [^ha ,  ^h5];
  [1, ^h1 ,^h8  ]   ->    [^h1 ,  ^h8];


end
```

Design Brief #3: 8-bit Counter Using SNAP

```
@PINLIST
CLOCK I;RESET I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0.D =     /OUT0;
OUT1.D =     /OUT0*OUT1
             + OUT0*/OUT1;
OUT2.D =     /OUT2*OUT1*OUT0
             + OUT2*(/OUT1 + /OUT0);
OUT3.D =     /OUT3*OUT2*OUT1*OUT0
             + OUT3*(/OUT2 + /OUT1 + /OUT0);
OUT4.D =     /OUT4*OUT3*OUT2*OUT1*OUT0
             + OUT4*(/OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT5.D =     /OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
             + OUT5*(/OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT6.D =     /OUT6*OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
             + OUT6*(/OUT5 + /OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT7.D =     /OUT7*OUT6*OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
             + OUT7*(/OUT6 + /OUT5 + /OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT0.CLK =   CLOCK;OUT1.CLK = CLOCK;OUT2.CLK = CLOCK;OUT3.CLK = CLOCK;
OUT4.CLK =   CLOCK;OUT5.CLK = CLOCK;OUT6.CLK = CLOCK;OUT7.CLK = CLOCK;
OUT0.RST =   RESET;OUT1.RST = RESET;OUT2.RST = RESET;OUT3.RST = RESET;
OUT4.RST =   RESET;OUT5.RST = RESET;OUT6.RST = RESET;OUT7.RST = RESET;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device = 0X20EV8
Pin1 = RESET
Pin3 = CLOCK
Pin4 = OUT0
Pin5 = OUT5
Pin7 = OUT7
Pin8 = OUT3
Pin17 = OUT4
Pin18 = OUT1
Pin20 = OUT6
Pin21 = OUT2

DESIGN FROM COUNTER.N2
FOR DEVICE 10X20EV8

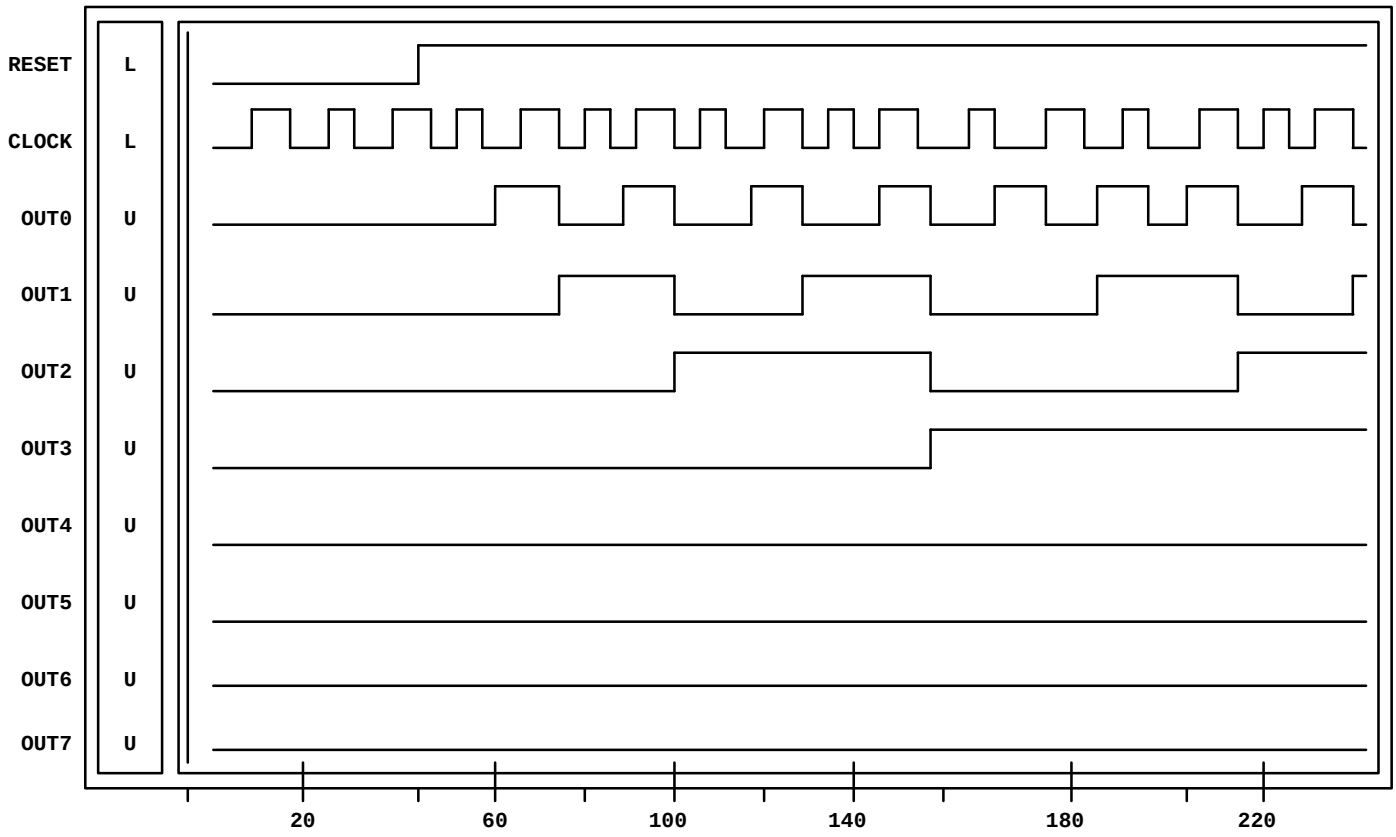
Cell Name	Used/Total	%
CKEV8	1 / 1	100%
DINEV8	8 / 28	28%
NINEV8	9 / 28	32%
AND	45 / 90	50%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	0 / 8	0%
OLMREG	8 / 8	100%
DFFEV8	8 / 8	100%
OUTEV8	8 / 8	100%

Design Brief #3: 8-bit Counter Using SNAP (continued)

```
                Output of Waveform Version 1.80
              Date: 01/20/92        Time: 17:26:48
              Input File Name      :    COUNTER.SCL
              Rule File Name :     Scl Rule
              Output File Name     :    COUNTER.SCL

P RESET, CLOCK, OUT0, OUT1, OUT2, OUT3, OUT4, OUT5, OUT6, OUT7
PCO
S 0 (40) RESET
S 0 (7,14, ETC) CLOCK
SU time = 5220
F

File:  counter.res          Delay = 0ns        Marker = 0ns        Sec/Div = 20ns
   (Model)
```




```
PHILIPS © 1992      WFA 1.80        MODE = 1                    F1 : HELP, F10 : EXIT
```

Design Brief #3: 8-bit Counter Using ABEL

```
module_count;

countb device 'ec20ev8a';

clock,reset           pin 3,1;
out0,out1,out2,out3   pin 4,18,21,8;
out4,out5,out6,out7   pin 17,5,20,7;


count = [out7..out0];
H,L,C = [1,0,.C.;


equations

count := count + 1;
count.ar = !reset;

test_vectors
( [clock  ,reset ]-->      count);
  [   L  , L     ]-->      0;
  [   C   ,L     ]-->      0;
  [   C   ,H     ]-->      1;
  [   C   ,H     ]-->      2;
  [   C   ,H     ]-->      3;
  [   C   ,H     ]-->      4;
  [   C   ,H     ]-->      5;
  [   C   ,H     ]-->      6;
  [   C   ,H     ]-->      7;
  [   C   ,H     ]-->      8;
  [   C   ,H     ]-->      9;
  [   C   ,H     ]-->      10;
  [   C   ,H     ]-->      11;
  [   C   ,H     ]-->      12;
  [   C   ,L     ]-->      0;

end
```

Design Brief #4: Octal Shifter Using SNAP

```
@PINLIST
CLOCK I;DATIN I;RESET I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0.D =     DATIN;
OUT1.D =     OUT0;
OUT2.D =     OUT1;
OUT3.D =     OUT2;
OUT4.D =     OUT3;
OUT5.D =     OUT4;
OUT6.D =     OUT5;
OUT7.D =     OUT6;
OUT0.CLK =   CLOCK;OUT1.CLK = CLOCK;OUT2.CLK = CLOCK;OUT3.CLK = CLOCK;
OUT4.CLK =   CLOCK;OUT5.CLK = CLOCK;OUT6.CLK = CLOCK;OUT7.CLK = CLOCK;
OUT0.RST =   RESET;OUT1.RST = RESET;OUT2.RST = RESET;OUT3.RST = RESET;
OUT4.RST =   RESET;OUT5.RST = RESET;OUT6.RST = RESET;OUT7.RST = RESET;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device = 0X20EV8
Pin1 = RESET
Pin2 = DATIN
Pin3 = CLOCK
Pin4 = OUT0
Pin5 = OUT1
Pin7 = OUT2
Pin8 = OUT3
Pin17 = OUT4
Pin18 = OUT5
Pin20 = OUT6
Pin21 = OUT7

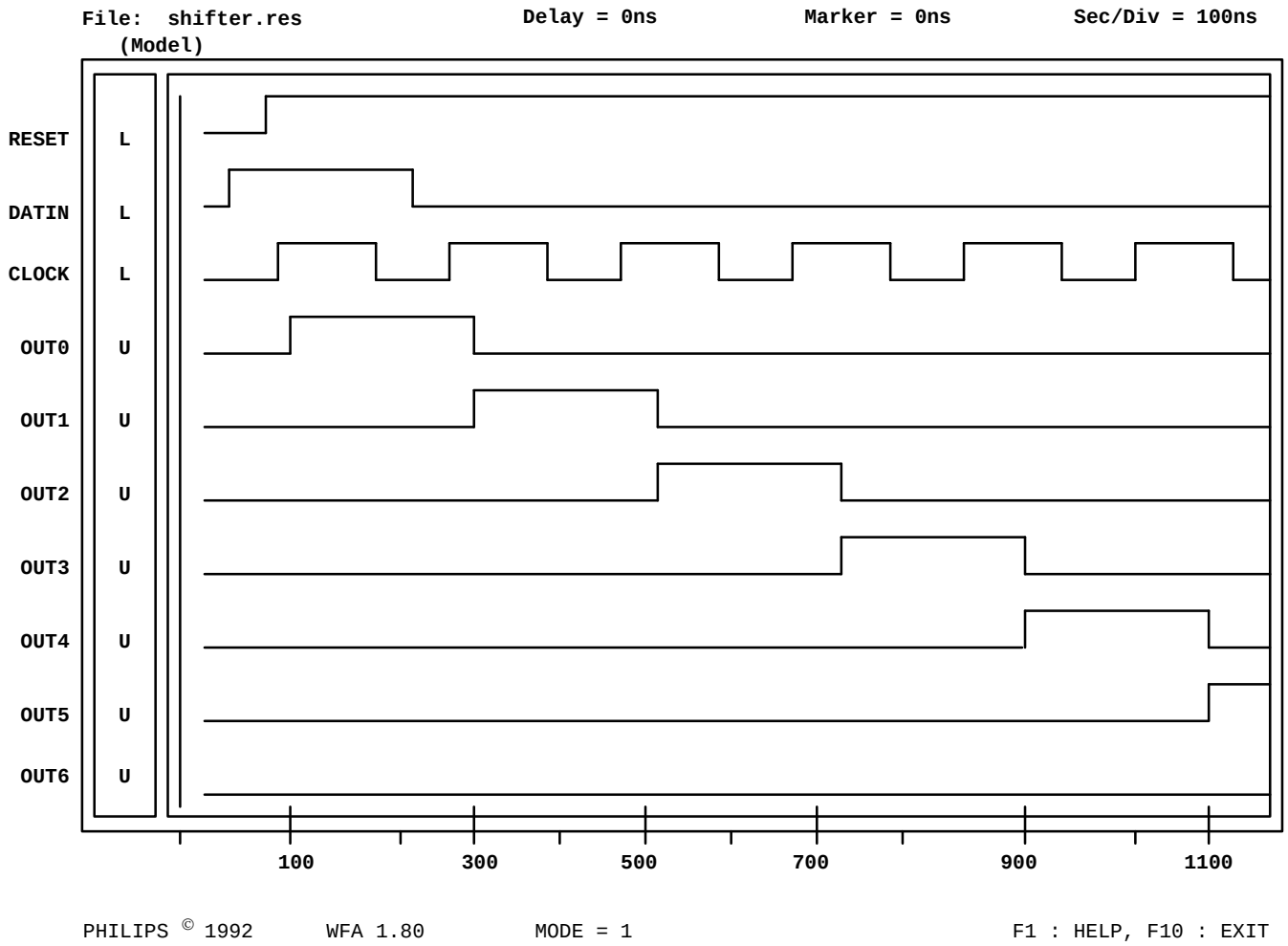
DESIGN FROM SHIFTER.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	1 / 1	100%
DINEV8	1 / 28	3%
NINEV8	8 / 28	28%
AND	17 / 90	18%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	0 / 8	0%
OLMREG	8 / 8	100%
DFFEV8	8 / 8	100%
OUTEV8	8 / 8	100%

Design Brief #4: Octal Shifter using SNAP (continued)

```
                Output of Waveform Version 1.80
              Date: 01/20/92        Time: 17:15:42
              Input File Name      :    SHIFTER.SCL
              Rule File Name :     Scl Rule
              Output File Name     :    SHIFTER.SCL
```

```
P RESET, DATIN, CLOCK, OUT0, OUT1, OUT2, OUT3, OUT4, OUT5, # OUT6, OUT7
PCO
S 0 (80) RESET
S 0 (40,220,1640) DATIN
S 0 (100,200,ETC) CLOCK
S 1 (5880) VCC
S 0 (5880) GND
SU time = 5880
F
```




```
PHILIPS © 1992      WFA 1.80        MODE = 1                F1 : HELP, F10 : EXIT
```

Design Brief #4: Octal Shifter Using ABEL

```
module_shift;

shifter device 'ec20ev8a';

clock,reset,datain    pin 3,1,2;
out0,out1,out2,out3   pin 4,5,7,8;
out4,out5,out6,out7   pin 17,18,20,21;
output = [out7..out0];
H,L,C = [1,0,.C.;


output istype 'buffer';


equations
out0.d = datain;
out1.d = out0;
out2.d = out1;
out3.d = out2;
out4.d = out3;
out5.d = out4;
out6.d = out5;
out7.d = out6;

[output].ar = !reset;

test_vectors
(  [clock,reset,datain    ]   ⟶   [out0  ,out1  ,out2  ,out3,out4,out5    ,out6  ,out7  ]);
   [   0   ,   0   ,   0    ]   ⟶   [   L   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   0   ,   1    ]   ⟶   [   L   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   0    ]   ⟶   [   L   ,   H   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   L   ,   H   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   0    ]   ⟶   [   L   ,   H   ,   L   ,   H   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   L   ,   H   ,   L   ,   H   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   0    ]   ⟶   [   L   ,   H   ,   L   ,   H   ,   L   ,   H    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   L   ,   H   ,   L   ,   H   ,   L    ,   H   ,   L   ];
   [   C   ,   1   ,   0    ]   ⟶   [   L   ,   H   ,   L   ,   H   ,   L   ,   H    ,   L   ,   H   ];

   [   C   ,   0   ,   1    ]   ⟶   [   L   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   L   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   H   ,   L   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   H   ,   H   ,   L    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   H   ,   H   ,   H    ,   L   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   H   ,   H   ,   H    ,   H   ,   L   ];
   [   C   ,   1   ,   1    ]   ⟶   [   H   ,   H   ,   H   ,   H   ,   H   ,   H    ,   H   ,   H   ];
   [   C   ,   0   ,   1    ]   ⟶   [   L   ,   L   ,   L   ,   L   ,   L   ,   L    ,   L   ,   L   ];

end
```