# ABEL: The ABEL Hardware Description Language

## ABEL.1 WHEN Synthesis

Just for fun, Table ABEL-1 shows the equations that the ABEL compiler produces for the entire `when` example program in Table 5-5.

## ABEL.2 Don't-Care Inputs

Some versions of the ABEL compiler have a limited ability to handle don't-care inputs. As mentioned previously, ABEL equations specify input combinations that belong to the on-set of a logic function; the remaining combinations are assumed to belong to the off-set. If some input combinations can instead be assigned to the d-set, then the program may be able to use these don't-care inputs to do a better job of minimization.

The ABEL language defines two mechanisms for assigning input combinations to the d-set. In order to use either mechanism, you must include the compiler directive `@DCSET` in your program or include "`dc`" in the `istype` property list of the outputs for which you want don't-cares to be considered.

*`@DCSET`*
*`dc`*

The first mechanism is the *don't-care unclocked assignment operator*, *`?=`*. This operator is used instead of = in equations to indicate that input combinations matching the righthand side should be put into the d-set instead of the on-set. Although this operator is documented in the ABEL compiler that I use, unfortunately it seems to be broken, so I'm not going to talk about it any more.

*`?=`don't-care unclocked assignment operator*

The second mechanism is the truth table. When don't-care processing is enabled, any input combinations that are not explicitly listed in the truth table are put into the d-set. Thus, the prime BCD-digit detector described at the beginning of Section Min.2 can be specified in ABEL as shown in Table ABEL-2. A don't-care value is implied for input combinations 10–15 because these combinations do not appear in the truth table and the `@DCSET` directive is in effect.

It's also possible to specify don't-care combinations explicitly, as shown in the second truth table. As introduced at the very beginning of this section, ABEL recognizes `.X.` as a special 1-bit constant whose value is "don't-care." In Table ABEL-2, the identifier "`X`" has been equated to this constant just to make it easier to type don't-cares in the truth table. The minimized equations resulting from Table ABEL-2 are shown in Table ABEL-3. Notice that the two equations for `F` are not equal; the compiler has selected different values for the don't-cares.

**Table ABEL-1** Synthesized equations file produced by ABEL for program in Table 5-5.

```
ABEL 6.30

Design whenex created Wed Dec 2 1998

Title: WHEN Statement Examples

 P-Terms  Fan-in Fan-out Type Name
--------- ------ ------- ---- -----
   2/3       4       1   Pin  X1
   2/3       4       1   Pin  X1A
   6/3       6       1   Pin  X2
   6/3       6       1   Pin  X2A
   3/4       6       1   Pin  X3
   3/4       6       1   Pin  X3A
   2/3       4       1   Pin  X4
   1/3       3       1   Pin  X5
   2/3       4       1   Pin  X6
   1/3       3       1   Pin  X7
   4/4       5       1   Pin  X8
   2/2       3       1   Pin  X9
   2/4       5       1   Pin  X10
=========
  36/42     Best P-Term Total: 30
                Total Pins: 19
              Total Nodes: 0
       Average P-Term/Output: 2
```

```
Equations:

X1 = (C & !D & !A
     # C & !D & B);

X1A = (C & !D & !A
     # C & !D & B);

X2 = (D & A & B
     # C & A & B
     # !B & E
     # !A & E
     # !B & F
     # !A & F);

X2A = (D & A & B
     # C & A & B
     # !B & E
     # !A & E
     # !B & F
     # !A & F);

X3 = (C & !A & !B & F
     # !A & B & E
     # D & A);

X3A = (C & !A & !B & F
     # !A & B & E
     # D & A);

X4 = (D & A & B
     # !A & E);

X5 = (D & A & B);

X6 = (A & !B & E
     # !C & !A & E);

X7 = (C & !A & F);

X8 = (D & A & E & F
     # A & B
     # !A & !E
     # !D & !A);

X9 = (D & !A
     # D & !B);

X10 = (C & D & !A
     # A & !B & E);
```

```
Reverse-Polarity Eqns:

!X1 = (A & !B
     # D
     # !C);

!X1A = (A & !B
     # D
     # !C);

!X2 = (!C & !D & A & B
     # !B & !E & !F
     # !A & !E & !F);

!X2A = (!C & !D & A & B
     # !B & !E & !F
     # !A & !E & !F);

!X3 = (!C & !A & !B
     # !A & B & !E
     # !D & A
     # !A & !B & !F);

!X3A = (!C & !A & !B
     # !A & B & !E
     # !D & A
     # !A & !B & !F);

!X4 = (A & !B
     # !D & A
     # !A & !E);

!X5 = (!A
     # !D
     # !B);

!X6 = (A & B
     # C & !A
     # !E);

!X7 = (A
     # !C
     # !F);

!X8 = (A & !B & !F
     # D & !A & E
     # A & !B & !E
     # !D & A & !B);

!X9 = (!D
     # A & B);

!X10 = (A & B
     # !D & !A
     # !C & !A
     # A & !E);
```

```
module DontCare
title 'Dont Care Examples'
@DCSET

" Input and output pins
N3..N0, A, B                    pin;
F, Y                           pin istype 'com';

NUM = [N3..N0];
X = .X.;

truth_table (NUM->F)
               0->0;
               1->1;
               2->1;
               3->1;
               4->0;
               5->1;
               6->0;
               7->1;
               8->0;
               9->0;

truth_table ([A,B]->Y)
             [0,0]->0;
             [0,1]->X;
             [1,0]->X;
             [1,1]->1;

end DontCare
```

**Table ABEL-2**
ABEL program using don't-cares.

```
Equations:
F = (!N2 & N1
     # !N3 & N0);
Y = (B);

Reverse-Polarity Equations:
!F = (N2 & !N0
     # N3
     # !N1 & !N0);
!Y = (!B);
```

**Table ABEL-3**
Minimized equations derived from Table ABEL-2.