Two Examples of ABEL Language Programming of a PLD: With and Without Finite State Machine Syntax

```
Module   ctrpal2a  "Note: a 'Module' line followed by project name is
                        "required.  This statement marks the beginning of
                        "a section that must terminate with an 'END' statement

                        "Also the quote character marks the beginning of
                        "a comment that extends until the next quote or the
                        "end of the line.

Title       'PAL 2-Bit Counter in 16V8 by ABEL'

ctrpal2a DEVICE  'P16V8';  / / This lines specifies the device
                                                / / architecture

@Alternate;          / / Changes the Boolean operator set to what I like.  You
                     / / can leave it out and use the operators ! & # $ !$ instead.
                     / / The double '/ /' starts a comment the same way a quote
                     / / does, but the comment ends with the end of line regardless of
                     / / any slashes or quotes in the line.

Q1, Q0   PIN      18, 16    ISTYPE   'REG';    / / Assigns the signal names Q1 and Q0 to
                                                / / pins 18 and 16 respectively.  Also
                                                / / directs that these are registered
                                                / / (D-flipflop) outputs.

GOING    PIN      13        ISTYPE   'COM';    / / Combinational output called GOING
                                                / / on pin 13.

CLOCK    PIN      1         ;
/OE      PIN      11        ;
UP       PIN      2         ;
HALT     PIN      3         ;

EQUATIONS

[Q1..Q0].CLK     =         CLOCK;   / / The notation [Q1..Q0] specifies all the
                                    / / signals in the "range" Q1 to Q0.  This does
                                    / / not save much space here but would help,
                                    / / for example, with [Q10..Q0] meaning Q10,
                                    / / Q9, Q8, Q7,...Q1, Q0.

[Q1..Q0].OE      =         OE;      / / Sets output enable of outputs from OE input.
GOING.OE         =         OE;

Q1       :=      HALT*Q1.FB + /HALT*(UP*Q0.FB + /UP*(/Q1.FB*/Q0.FB
                        + Q1.FB*Q0.FB));
Q0       :=      HALT*Q0.FB + /HALT*/Q0.FB;

GOING    =       /HALT;

END;
```

```
Module     ctrpal2a     "Note: a 'Module' line followed by project name is
                              "required.  This statement marks the beginning of
                              "a section that must terminate with an 'END' statement

                              "Also the quote character marks the beginning of
                              "a comment that extends until the next quote or the
                              "end of the line.

Title      'PAL 2-Bit Counter in 16V8 by ABEL'

ctrpal2a   DEVICE   'P16V8';    / / This lines specifies the device
                                                    / / architecture

@Alternate;              / / Changes the Boolean operator set to what I like.  You
                         / / can leave it out and use the operators ! & # $ !$ instead.
                         / / The double ' / / ' starts a comment the same way a quote
                         / / does, but the comment ends with the end of line regardless of
                         / / any slashes or quotes in the line.

Q1, Q0     PIN      18, 16     ISTYPE    'REG';    / / Assigns the signal names Q1 and Q0 to
                                                    / / pins 18 and 16 respectively.  Also
                                                    / / directs that these are registered
                                                    / / (D-flipflop) outputs.

GOING      PIN      13         ISTYPE    'COM';    / / Combinational output called GOING
                                                    / / on pin 13.
ROLLING    PIN      14         ISTYPE    'COM';    / / Combinational output indicating
                                                    / / counter rollover.

CLOCK      PIN      1          ;
/ OE       PIN      11         ;
UP         PIN      2          ;
HALT       PIN      3          ;

"Declare the present state vector.

PRES_STATE = [Q1..Q0];   / / The notation [Q1..Q0] specifies all the
                         / / signals in the "range" Q1 to Q0.  This does
                         / / not save much space here but would help,
                         / / for example, with [Q10..Q0] meaning Q10,
                         / / Q9, Q8, Q7,...Q1, Q0.

"Define the state values -- constants.  Since this is a counter, the
"state and the output are the same.  Label by binary value of the count.

OUT0 = [0, 0];           / / Q1 = 0, Q0 = 0 for this state -- count = 0.
OUT1 = [0, 1];
OUT2 = [1, 0];
OUT3 = [1, 1];

STATE_DIAGRAM   PRES_STATE          "STATE_DIAGRAM is a reserved word to label
                                    "the state machine transition table or
                                    "diagram.  This is the up / down logic.

STATE      OUT0:     IF (GOING == 0) THEN OUT0 ELSE IF (UP == 1)
           THEN OUT1 ELSE OUT3;
STATE      OUT1:     IF (GOING == 0) THEN OUT1 ELSE IF (UP == 1)
           THEN OUT2 ELSE OUT0;
STATE      OUT2:     IF (GOING == 0) THEN OUT2 ELSE IF (UP == 1)
           THEN OUT3 ELSE OUT1;
STATE      OUT3:     IF (GOING == 0) THEN OUT3 ELSE IF (UP == 1)
           THEN OUT0 ELSE OUT2;

EQUATIONS          "Clock declaration (and output enable declaration if used)
                   "and signals derived from the state or input bits by logical
                   "operations.

PRES_STATE.CLK  =        CLOCK;    "Pin 1 is only possible clock for 16V8
                                   "architecture

ROLLING = UP*Q1.FB*Q0.FB + / UP* / Q1.FB* / Q0.FB    "HIGH for the clock cycle
                                                     "before counter rolls over.
GOING      =        / HALT;

END;
```