```
--  Title Statement (optional)

--  Include Statement (optional)

--  Constant Statement (optional)

--  Define Statement (optional)

--  Parameters Statement (optional)
PARAMETERS
(
    NCLKSEC = 99,
    NCLKDISP = 2
);

--  Function Prototype Statement (optional)
INCLUDE "bcddecode.inc";

--  Options Statement (optional)

--  Assert Statement (optional)

--  Subdesign Section
subdesign calendar2
(
    VALUE[7..0] :input;
    SELECT[2..0] :input;
    nPRESET :input;
    CLOCK1M :input;
    SEGMENT[6..0] :output;
    OEYEAR1000 :output;
    OEYEAR0100 :output;
    OEYEAR0010 :output;
    OEYEAR0001 :output;
    OEMONTH10 :output;
    OEMONTH01 :output;
    OEDAY10 :output;
    OEDAY01 :output;
    OEHOUR10 :output;
    OEHOUR01 :output;
    OEMINUTE10 :output;
    OEMINUTE01 :output;
    OESECOND10 :output;
    OESECOND01 :output;
)
--  Variable Section (optional)
variable
    --  If Generate Statement (optional)

    --  Node Declaration (optional)
    isLeapYear :node;

    --  Instance Declaration (optional)
    ledout  : bcddecode;
    --  Register Declaration (optional)
    YEAR[15..0] :dff;
    MONTH[4..0] :dff;
    DAY[5..0] :dff;
    HOUR[5..0] :dff;
    MINUTE[6..0] :dff;
    SECOND[6..0] :dff;

    counter[19..0] :dff;
    dispcnt[15..0] :dff;
    dispsel[3..0] :dff;
    --  State Machine Declaration (optional)

    --  Machine Alias Declaration (optional)

    --  Assert Statement (optional)

--  Logic Section
begin
    --  Defaults Statement (optional)
```

```
    --  The following statements can be freely intermixed:

    --  Boolean Equation
YEAR[].clk = CLOCK1M;
YEAR[].clrn = VCC;
YEAR[].prn = VCC;
MONTH[].clk = CLOCK1M;
MONTH[].clrn = VCC;
MONTH[].prn = VCC;
DAY[].clk = CLOCK1M;
DAY[].clrn = VCC;
DAY[].prn = VCC;
HOUR[].clk = CLOCK1M;
HOUR[].clrn = VCC;
HOUR[].prn = VCC;
MINUTE[].clk = CLOCK1M;
MINUTE[].clrn = VCC;
MINUTE[].prn = VCC;
SECOND[].clk = CLOCK1M;
SECOND[].clrn = VCC;
SECOND[].prn = VCC;

counter[].clk = CLOCK1M;
counter[].clrn = VCC;
counter[].prn = VCC;
dispcnt[].clk = CLOCK1M;
dispcnt[].clrn = VCC;
dispcnt[].prn = VCC;
dispsel[].clk = CLOCK1M;
dispsel[].clrn = VCC;
dispsel[].prn = VCC;
    --  Case Statement

CASE dispsel[] IS
    WHEN 0 =>
        ledout.bcd[] = YEAR[15..12];
        SEGMENT[] = ledout.segment[];
        OEYEAR1000 = VCC;
    WHEN 1 =>
        ledout.bcd[] = YEAR[11..8];
        SEGMENT[] = ledout.segment[];
        OEYEAR0100 = VCC;
    WHEN 2 =>
        ledout.bcd[] = YEAR[7..4];
        SEGMENT[] = ledout.segment[];
        OEYEAR0010 = VCC;
    WHEN 3 =>
        ledout.bcd[] = YEAR[3..0];
        SEGMENT[] = ledout.segment[];
        OEYEAR0001 = VCC;
    WHEN 4 =>
        ledout.bcd[] = (0,0,0,MONTH[4]);
        SEGMENT[] = ledout.segment[];
        OEMONTH10 = VCC;
    WHEN 5 =>
        ledout.bcd[] = MONTH[3..0];
        SEGMENT[] = ledout.segment[];
        OEMONTH01 = VCC;
    WHEN 6 =>
        ledout.bcd[] = (0,0,DAY[5..4]);
        SEGMENT[] = ledout.segment[];
        OEDAY10 = VCC;
    WHEN 7 =>
        ledout.bcd[] = DAY[3..0];
        SEGMENT[] = ledout.segment[];
        OEDAY01 = VCC;
    WHEN 8 =>
        ledout.bcd[] = (0,0,HOUR[5..4]);
        SEGMENT[] = ledout.segment[];
        OEHOUR10 = VCC;
    WHEN 9 =>
        ledout.bcd[] = HOUR[3..0];
        SEGMENT[] = ledout.segment[];
```

```
        OEHOUR01 = VCC;
    WHEN 10 =>
        ledout.bcd[] = (0,MINUTE[6..4]);
        SEGMENT[] = ledout.segment[];
        OEMINUTE10 = VCC;
    WHEN 11 =>
        ledout.bcd[] = MINUTE[3..0];
        SEGMENT[] = ledout.segment[];
        OEMINUTE01 = VCC;
    WHEN 12 =>
        ledout.bcd[] = (0,SECOND[6..4]);
        SEGMENT[] = ledout.segment[];
        OESECOND10 = VCC;
    WHEN 13 =>
        ledout.bcd[] = SECOND[3..0];
        SEGMENT[] = ledout.segment[];
        OESECOND01 = VCC;
    WHEN OTHERS =>
        OEYEAR1000 = GND;
        OEYEAR0100 = GND;
        OEYEAR0010 = GND;
        OEYEAR0001 = GND;
        OEMONTH10 = GND;
        OEMONTH01 = GND;
        OEDAY10 = GND;
        OEDAY01 = GND;
        OEHOUR10 = GND;
        OEHOUR01 = GND;
        OEMINUTE10 = GND;
        OEMINUTE01 = GND;
        OESECOND10 = GND;
        OESECOND01 = GND;
END CASE;

IF (YEAR[7..0] == H"00") THEN
    CASE YEAR[15..8] IS
        WHEN H"00" => -- year 0000
            isLeapYear = VCC;
        WHEN H"04" => -- year 0400
            isLeapYear = VCC;
        WHEN H"08" => -- year 0800
            isLeapYear = VCC;
        WHEN H"12" => -- year 1200
            isLeapYear = VCC;
        WHEN H"16" => -- year 1600
            isLeapYear = VCC;
        WHEN H"20" => -- year 2000
            isLeapYear = VCC;
        WHEN H"24" => -- year 2400
            isLeapYear = VCC;
        WHEN H"28" => -- year 2800
            isLeapYear = VCC;
        WHEN H"32" => -- year 3200
            isLeapYear = VCC;
        WHEN H"36" => -- year 3600
            isLeapYear = VCC;
        WHEN H"40" => -- year 4000
            isLeapYear = VCC;
        WHEN H"44" => -- year 4400
            isLeapYear = VCC;
        WHEN H"48" => -- year 4800
            isLeapYear = VCC;
        WHEN H"52" => -- year 5200
            isLeapYear = VCC;
        WHEN H"56" => -- year 5600
            isLeapYear = VCC;
        WHEN H"60" => -- year 6000
            isLeapYear = VCC;
        WHEN H"64" => -- year 6400
            isLeapYear = VCC;
        WHEN H"68" => -- year 6800
            isLeapYear = VCC;
        WHEN H"72" => -- year 7200
            isLeapYear = VCC;
```

```
        WHEN H"76" => -- year 7600
            isLeapYear = VCC;
        WHEN H"80" => -- year 8000
            isLeapYear = VCC;
        WHEN H"84" => -- year 8400
            isLeapYear = VCC;
        WHEN H"88" => -- year 8800
            isLeapYear = VCC;
        WHEN H"92" => -- year 9200
            isLeapYear = VCC;
        WHEN H"96" => -- year 9600
            isLeapYear = VCC;
        WHEN OTHERS =>
            isLeapYear = GND;
    END CASE;
ELSIF (YEAR[1..0] == B"00") THEN
    isLeapYear = VCC;
END IF;
    --  For Generate Statement

    --  If Generate Statement

    --  If Then Statement
IF (!nPRESET) THEN -- see if preset strobed
    counter[].clrn = GND;
    dispcnt[].clrn = GND;
    dispsel[].clrn = GND;
    CASE SELECT[] IS
        WHEN 0 =>
            YEAR[15..8] = VALUE[7..0];
            YEAR[7..0] = YEAR[7..0];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
        WHEN 1 =>
            YEAR[7..0] = VALUE[7..0];
            YEAR[15..8] = YEAR[15..8];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
        WHEN 2 =>
            MONTH[4..0] = VALUE[4..0];
            YEAR[] = YEAR[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
        WHEN 3 =>
            DAY[5..0] = VALUE[5..0];
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
        WHEN 4 =>
            HOUR[5..0] = VALUE[5..0];
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
        WHEN 5 =>
            MINUTE[6..0] = VALUE[6..0];
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            SECOND[] = SECOND[];
        WHEN 6 =>
            SECOND[6..0] = VALUE[6..0];
```

```
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
        WHEN OTHERS =>
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
            HOUR[] = HOUR[];
            MINUTE[] = MINUTE[];
            SECOND[] = SECOND[];
    END CASE;
ELSIF (counter[] == NCLKSEC) THEN -- see if 1sec event
    counter[] = 0;
    IF (SECOND[] == B"1011001") THEN -- see if second is 59
        SECOND[] = B"0000000";
        IF (MINUTE[] == B"1011001") THEN -- see if minute is 59
            MINUTE[] = B"0000000";
            IF (HOUR[] == B"100011") THEN -- see if hour is 23
                HOUR[] = B"000000";
                IF (MONTH[] == B"00010") THEN -- see if month feb
                    IF ((isLeapYear & (DAY[] == B"101001")) #
                        (!isLeapYear & (DAY[] == B"101000"))) THEN -- see if day is 29th of
leap year or 28th
                            DAY[] = B"000001";
                            MONTH[3..0] = MONTH[3..0]+1; -- should become march 1st
                            YEAR[] = YEAR[];
                    END IF;
                ELSIF ((MONTH[] == B"00001") # (MONTH[] == B"00011") # (MONTH[] == B"00101") #
                        (MONTH[] == B"00111") # (MONTH[] == B"01000") # (MONTH[] == B"01010") #
                        (MONTH[] == B"10010")) THEN -- see if month is jan, march, may, july,
aug, oct or dec
                    IF (DAY[] == B"110001") THEN -- see if day is 31th
                        DAY[] = B"000001"; -- should become 1st
                        IF (MONTH[] == B"10010") THEN -- see if month is dec
                            MONTH[] = B"00001"; -- should become jan
                            IF (YEAR[3..0] == B"1001") THEN -- see if year is xxx9
                                YEAR[3..0] = B"0000"; -- should become xxx0
                                IF (YEAR[7..4] == B"1001") THEN -- see if year is xx99
                                    YEAR[7..4] = B"0000"; -- shoud become xx00
                                    IF (YEAR[11..8] == B"1001") THEN -- see if year is x999
                                        YEAR[11..8] = B"0000"; -- should become x000
                                        IF (YEAR[15..12] == B"1001") THEN -- see if year is
9999
                                            YEAR[15..12] = B"0000"; -- should become 0000
                                        END IF;
                                    END IF;
                                END IF;
                            ELSE
                                YEAR[15..4] = YEAR[15..4];
                                YEAR[3..0] = YEAR[3..0]+1;
                            END IF;
                        ELSE
                            MONTH[4] = MONTH[4];
                            MONTH[3..0] = MONTH[3..0]+1;
                            YEAR[] = YEAR[];
                        END IF;
                    ELSIF (DAY[3..0] == B"1001") THEN -- see if day is x9
                        DAY[3..0] = B"0000";
                        DAY[5..4] = DAY[5..4]+1;
                        YEAR[] = YEAR[];
                        MONTH[] = MONTH[];
                    ELSE
                        DAY[3..0] = DAY[3..0]+1;
                        DAY[5..4] = DAY[5..4];
                        YEAR[] = YEAR[];
                        MONTH[] = MONTH[];
                    END IF;
                ELSE -- month is arp, jun, sep, nov
                    IF (DAY[5..0] == B"110000") THEN -- see if day is 30th
                        DAY[5..0] = B"000001"; -- should become 1st
                        MONTH[4] = MONTH[4];
                        MONTH[3..0] = MONTH[3..0]+1; -- of next month
```

```
                    YEAR[] = YEAR[];
                ELSIF (DAY[3..0] == B"1001") THEN -- see if day is x9th
                    DAY[3..0] = B"0000"; -- should become x0th
                    DAY[5..4] = DAY[5..4]+1;
                    YEAR[] = YEAR[];
                    MONTH[] = MONTH[];
                END IF;
            END IF;
        ELSIF (HOUR[3..0] == B"1001") THEN -- see if hour is x9
            HOUR[3..0] = B"0000"; -- should become x0
            HOUR[5..4] = HOUR[5..4]+1;
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
        ELSE
            HOUR[5..4] = HOUR[5..4];
            HOUR[3..0] = HOUR[3..0]+1;
            YEAR[] = YEAR[];
            MONTH[] = MONTH[];
            DAY[] = DAY[];
        END IF;
    ELSIF (MINUTE[3..0] == B"1001") THEN -- see if minute is x9
        MINUTE[3..0] = B"0000"; -- should become x0
        MINUTE[6..4] = MINUTE[6..4]+1;
        YEAR[] = YEAR[];
        MONTH[] = MONTH[];
        DAY[] = DAY[];
        HOUR[] = HOUR[];
    ELSE
        MINUTE[6..4] = MINUTE[6..4];
        MINUTE[3..0] = MINUTE[3..0]+1;
        YEAR[] = YEAR[];
        MONTH[] = MONTH[];
        DAY[] = DAY[];
        HOUR[] = HOUR[];
    END IF;
ELSIF (SECOND[3..0] == B"1001") THEN -- see if second is x9
    SECOND[3..0] = B"0000"; -- should become x0
    SECOND[6..4] = SECOND[6..4]+1;
    YEAR[] = YEAR[];
    MONTH[] = MONTH[];
    DAY[] = DAY[];
    HOUR[] = HOUR[];
    MINUTE[] = MINUTE[];
ELSE
    SECOND[3..0] = SECOND[3..0]+1;
    SECOND[6..4] = SECOND[6..4];
    YEAR[] = YEAR[];
    MONTH[] = MONTH[];
    DAY[] = DAY[];
    HOUR[] = HOUR[];
    MINUTE[] = MINUTE[];
END IF;
IF (dispcnt[] == NCLKDISP) THEN
    dispcnt[] = 0;
    IF (dispsel[] == 13) THEN
        dispsel[] = 0;
    ELSE
        dispsel[] = dispsel[]+1;
    END IF;
ELSE
    dispcnt[] = dispcnt[]+1;
    dispsel[] = dispsel[];
END IF;
ELSE
    IF (dispcnt[] == NCLKDISP) THEN
        dispcnt[] = 0;
        IF (dispsel[] == 13) THEN
            dispsel[] = 0;
        ELSE
            dispsel[] = dispsel[]+1;
        END IF;
    ELSE
        dispcnt[] = dispcnt[]+1;
```

```
        dispsel[] = dispsel[];
    END IF;
    YEAR[] = YEAR[];
    MONTH[] = MONTH[];
    DAY[] = DAY[];
    HOUR[] = HOUR[];
    MINUTE[] = MINUTE[];
    SECOND[] = SECOND[];
    counter[] = counter[]+1;
END IF;

    --  In-Line Logic Function Reference

    --  Truth Table Statement

    --  Assert Statement
end;
```