PLD programmable retriggerable one-shot AN011

FEATURES

- Programmable pulse-width/delay
- Maximum 256 clock cycles
- Asynchronous TRIGGER input
- Active-High and Active-Low outputs
- Asynchronous RESET
- 20-pin package

THEORY OF OPERATION

The one-shot consists of a PLC42VA12 and an external clock which may be part of the system in which this one-shot is to work. As shown in Figures 1 and 3 the PLD is configured to have a latch and an eight-bit binary up counter which is presettable by input data to any number less than 256. Since the input data is inverted before it is loaded into the registers, counting from the complements of the input to FF will give the correct number of counts as counting from the input down to 00.

Pulse-width/delay inputs may be the outputs of another device or switches. When /RESET goes Low, flip-flops are set to all 1's (terms PB, PA, and PM0). At the rising edge of the next clock, data is latched into the registers (terms LB, LA, and LM0). When /TRIG goes Low, it is latched into the input latch formed by term # 0, 1, 2 and 13. The output O1 of the latch goes High and O2 goes Low which enables the 8-bit counting cycle. The O1 and /O1 will maintain their output levels until the end of the counting cycle at which time the counter reaches the count FF, resets the latch by term # 13, and sets O2 High. At the rising edge of the next clock, terms LA, LB, and LM0 cause data to be loaded again into the registers, and the device is ready for another /TRIG input. The output waveforms are illustrated in Figure 2.

If the /TRIG pulse-width is longer than the desired pulse-width of the one-shot, the device will react as mentioned above, and at the end of the count cycle new data will be loaded, another count cycle begins while the outputs remain set by the /TRIG input without changing throughout the change-over of one count cycle to another. O1a, on the other hand, will go Low for one clock period at the change-over. As long as the /TRIG is Low, O1a will continue to pulse Low for one clock period at the change-over of one count cycle to another. The output O2 will pulse High for one clock cycle at the change-over. Figure 2 illustrates output wave-forms for both cases. The output wave-forms are as illustrated in Figure 2.

The one-shot is implemented by programming the PLC42VA12 as shown by the SNAP listing in Figures 3 and 4.

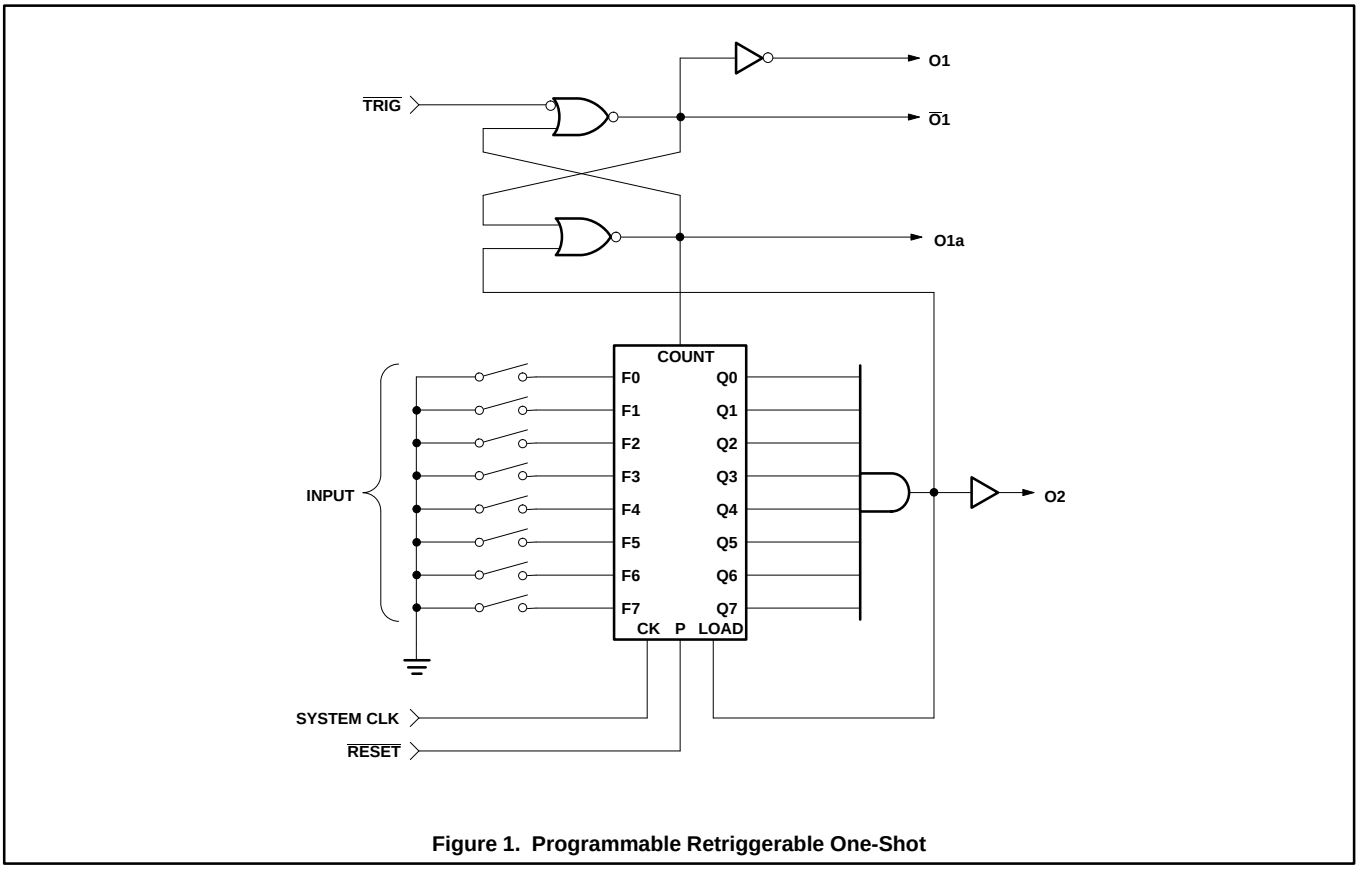
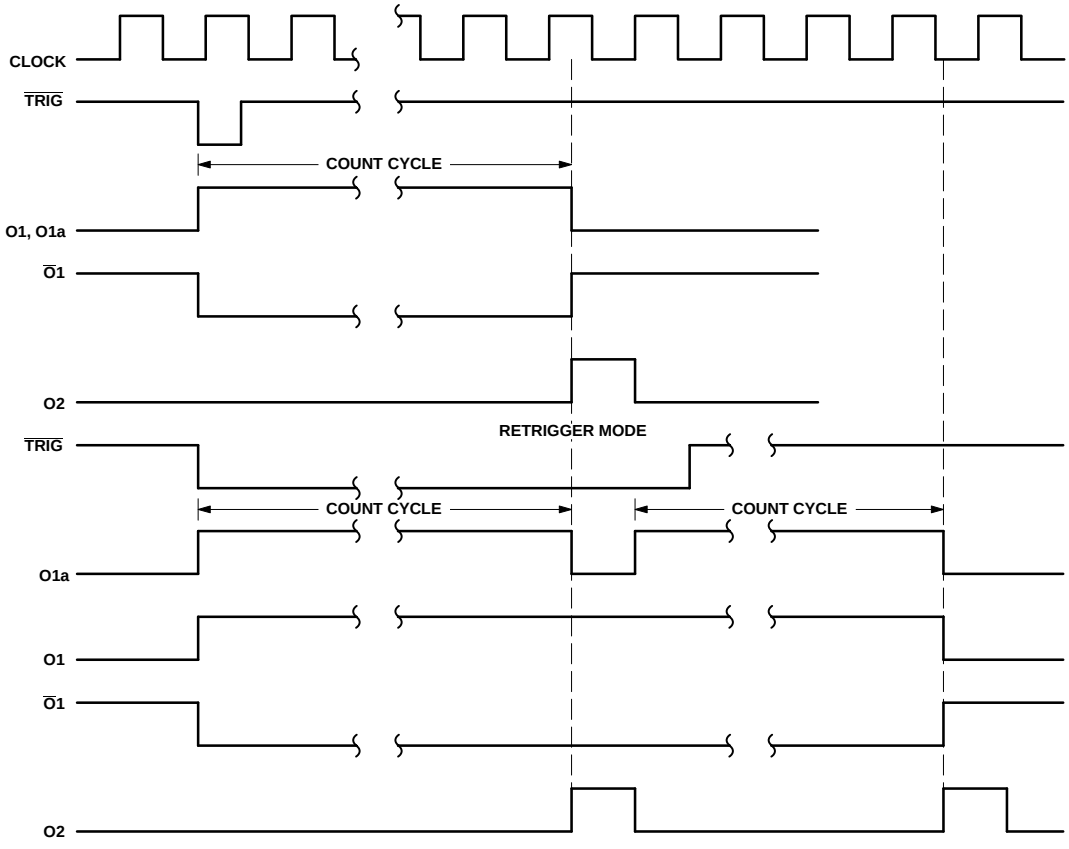


Figure 1. Programmable Retriggerable One-Shot

PLD programmable retriggerable one-shot AN011



Programmable One-Shot

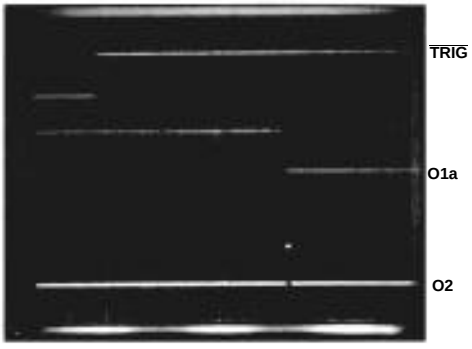


Figure 2. Timing Diagram of Programmable Retriggerable One-Shot

```
*************************************************************
*       PLC42VA12 24-Pin DIP Package Pin Layout             *
* Date: 08/10/93                          Time: 13:02:11 *
*************************************************************

                      _________   _________
                     |         \_____/         |
                     |                         |
              CLK [ 1|IO/CLK                VCC|24]
            RESET [ 2|I1                    M9|23] O2
             TRIG [ 3|I2                    M8|22] NO1
                  [ 4|I3                    M7|21] INF7
                  [ 5|I4                    M6|20] INF6
                  [ 6|I5                    M5|19] INF5
                  [ 7|I6                    M4|18] INF4
                  [ 8|I7                    M3|17] INF3
                  [ 9|I8                    M2|16] INF2
               O1 [10|B0                    M1|15] INF1
              O1A [11|B1                    M0|14] INF0
                  [12|GND               19/OE_|13]
                     |_________________________|
```

Figure 3. Pin Layout

```
"-------------------------------------"
" Programmable Retriggerable One-Shot "
"-------------------------------------"

" This design is for a PLC42VA12 device.
  A similar type of function may be programmed into
  any of the PLS155/7/9A type devices. These devices
  contain a flip-flop preload function which may be
  controlled by input pins and TTL voltage levels or
  by feedback into the array from the flip-flops outputs.

  This one-shot loads the data at the INFO-INF7
  input pins into the counter at the end of the clock
  cycle (O2 = HIGH). If TRIG input is LOW longer than
  the count cycle, output O1A will go LOW for one
  clock period and will go HIGH again for another count
  cycle. Outputs NO1 and O1 stay LOW and HIGH
  respectively until TRIG goes HIGH and the count
  cycle is completed without interruption.
"

@pinlist

clk        i;
reset      i;
trig       i;
inf[7..0]  i;
o1         o;
o2         o;
no1        b;
o1a        b;

@logic equations

"------------------------------"
" equations for latch circuit "
"------------------------------"

01     =  (no1+/trig);
o1a    = /(no1+/trig);
01a.oe = 1;
no1    = /((f7*f6*f5*f4*f3*f2*f1*f0)
       +  01a);
no1.oe = 1;
"--------------------------"
" count comparison equation "
"--------------------------"

o2     = (f7*f6*f5*f4*f3*f2*f1*f0);
```

Figure 4. SNAP Listing (1 of 2)

```
"------------------------------------"
" equations to load counter from pins "
"------------------------------------"

"use register preload feature"

temp[15..0].ld  = (f7*f6*f5*f4*f3*f2*f1*f0);
temp14          = inf7;
temp15          = /inf7;
temp12          = inf6;
temp13          = /inf6;
temp10          = inf5;
temp11          = /inf5;
temp8           = inf4;
temp9           = /inf4;
temp6           = inf3;
temp7           = /inf3;
temp4           = inf2;
temp5           = /inf2;
temp2           = inf1;
temp3           = /inf1;
temp0           = inf0;
temp1           = /inf0;

"-----------------"
" counter equations"
"-----------------"

"The counter is constructed using the toggle feature of
 JK flip-flops. Both J and K are connected to the same
 product term so only eight product erms are required to
 implement this counter. The '+tempXX' input does not require
 a product term in the 42VA12 or PLS155/7/9A type devices
 due to the wire-or register preloading feature. This
 feature is controlled internally in the device by the
 LA and LB product terms. SNAP automatically uses these
 control terms to implement the preload.
"

f0.j   = ((no1*reset)
          +temp1);
f0.k   = ((no1*reset)
          +temp0);
f1.j   = ((no1*f0*reset)
          +temp3);
f1.k   = ((no1*f0*reset)
          +temp2);
f2.j   = ((no1*f1*f0*reset)
          +temp5);
f2.k   = ((no1*f1*f0*reset)
          +temp4);
f3.j   = ((no1*f2*f1*f0*reset)
          +temp7);
f3.k   = ((no1*f2*f1*f0*reset)
          +temp6);
f4.j   = ((no1*f3*f2*f1*f0*reset)
          +temp9);
f4.k   = ((no1*f3*f2*f1*f0*reset)
          +temp8);
f5.j   = ((no1*f4*f3*f2*f1*f0*reset)
          +tem11);
f5.k   = ((no1*f4*f3*f2*f1*f0*reset)
          +tem10);
f6.j   = ((no1*f5*f4*f3*f2*f1*f0*reset)
          +tem13);
f6.k   = ((no1*f5*f4*f3*f2*f1*f0*reset)
          +tem12);
f7.j   = ((no1*f6*f5*f4*f3*f2*f1*f0*reset)
          +tem15);
f7.k   = ((no1*f6*f5*f4*f3*f2*f1*f0*reset)
          +tem14);

f[7..0].clk = clk;
f[7..0].set = reset;
```

Figure 4. SNAP Listing (2 of 2)