Application Note AN2249

PRS User Module as a One-Shot Pulse Width Discriminator and Debouncer

Author: ILYA Mamontov
Associated Project: Yes
Associated Part Family: CY8C27xxx
PSoC Designer Version: 4.2
Associated Application Notes: AN2108, AN2231

Abstract

Are you experiencing delays while shaping software pulses? If so, this Application Note is for you! By using the PRS User Module and its flexible configuration capabilities, you can find interesting solutions for your tasks. This document describes how to implement the one-shot pulse width discriminator and one-shot debouncer.

Introduction

Sometimes a device must produce a single pulse in response to an input signal. Such devices are called one-shots (or "univibrators" or "monostable multivibrators") and are used to delay and reshape input pulses.

One-shots are also used as debouncers. The source of the bounce does not always originate from mechanical switches.

The input signal has curved rising and falling edges. A comparator responds to an incoming pulse by triggering a signal as shown in Figure 1.

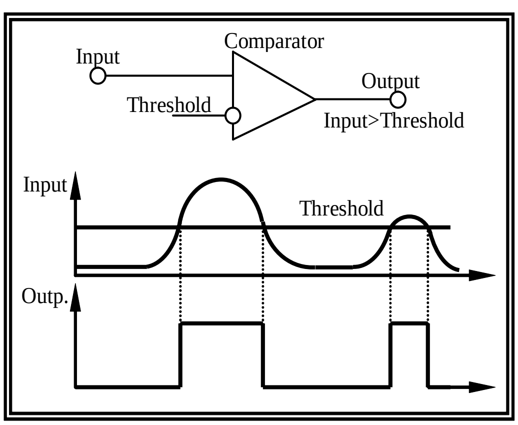

Figure 1. Ideal Operation of a Comparator

In reality, the input pulses (and thresholds) have a noise component. The comparator also has its own noise. As a result, it responds with multiple triggers when the signal crosses the threshold (Figure **2**). This phenomenon is called "bouncing."

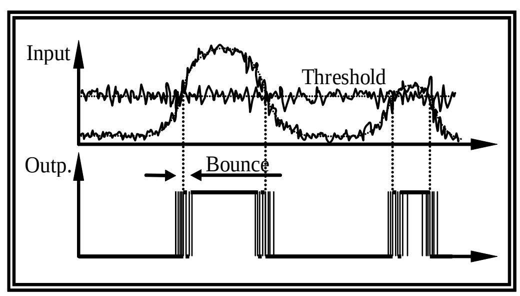

Figure 2. Reality Operation

False triggers will be counted as the noise in the signal introduces extra threshold crossings.

One way to eliminate false triggers is to add a hysteresis to the comparator. Application Note AN2108 - Standard - "Hysteresis Comparator with PSoC" describes this method. However, it may not always be possible to correctly route the signals because of the chip's architecture (for example, see Application Note AN2231 - Standard - "Ratemeter with a Precise Pulse Discriminator for Spectrometry").

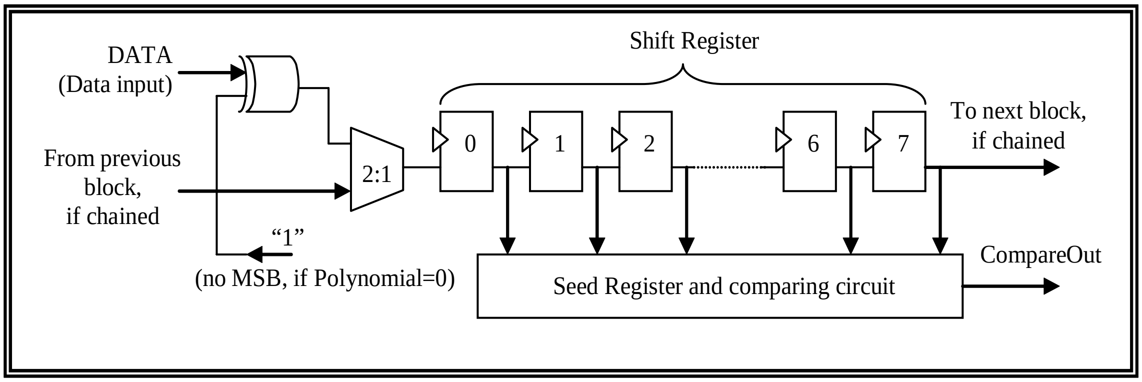

Figure 3. PRS Block Structure (Polynomial Register=00000000)

An alternative way is to use one-shots. When the first pulse is sent out for a specified period of time, the device ignores any other input pulses even if one is reactivated by a bounce. As a result, very clean output pulses are received.

PSoC Implementation

While researching the possibilities of the PRS User Module (Pseudo Random Sequence Generator) as a kind of Shift register, I discovered and tried to utilize an interesting feature. This feature appeared **only when a value of "0" was set in the Polynomial register**.

The internal block structure of the PRS with a zero polynomial value is shown in Figure 3.

Here, the PRS Shift register has no real data output (titled "OutputBitStream"), and the data input is inverted (by XOR with "1"). Note that the PRS also has no configurable data input because it is forced to "0" for proper operation.

To route and correct the data input (in other words, invert), the control registers should be programmed. Refer to Code 1.

```
M8C_SetBank1
;prepare (clear) 7..4 bits
and reg[PRS8_1_INPUT_REG],0x0F
;connect to Row_N_Input_3
or  reg[PRS8_1_INPUT_REG],0xF0
;the Data Input should be inverted
;(for normal operation)
or  reg[PRS8_1_FUNC_REG],0x80
M8C_SetBank0
```

Code 1. Programming the Control Registers

`PRS8_1_INPUT_REG` is an alias for the DxBxxIN register.

Table 1 lists all possible connections and code values for the "OR" command.

Table 1. Possible Connections

Code	Connection To:
00h	Low Level (0)
10h	High Level (1)
20h	Row Broadcast Net
30h	Chain Function To Previous Block
40h	Analog Column Comparator 0
50h	Analog Column Comparator 1
60h	Analog Column Comparator 2
70h	Analog Column Comparator 3
80h	Row Output 0
90h	Row Output 1
A0h	Row Output 2
B0h	Row Output 3
C0h	Row Input 0
D0h	Row Input 1
E0h	Row Input 2
F0h	Row Input 3

Note that the value of the Seed register should be set to 00000000 or 11111111 and the type of compare function must be "Equal."

Operation of the One-Shot

Operation of the one-shot is as follows:

1. With a clock tick, the Shift register is filled by a uniform sequence of 0s or 1s (defined by data input).

2. The comparing circuit sets the output to 0 or 1 (defined in the Seed register). This is a “stable” (or initial) state of the one-shot.

3. With varying input data, the Shift register is filled and the comparing circuit reacts to the combinations by outputting a 1 or 0.

4. The input then returns to initial state and the Shift register is filled by the initial sequence (returned to “stable” state).

The operating variants of the 8-bit PRS with different parameters are listed in Table 2. The operating diagrams are shown in Figure 4, Figure 5, Figure 6 and Figure 7.

The configuration code for the PRS can be found in the *main.asm* file in the associated project.

Table 2. Operating Variants for the 8-Bit PRS

Initial Data Input	Seed Register Value	Initial Output	Operating as One-Shot	Operating as Pulse Width Discriminator	Operating as Debouncer
0	00000000	1	First “1” triggers the output to low for 8 ticks, every subsequent “1” prolongs the low state for 8 ticks (Figure 4).		At bouncing, the output is active, there is an 8-tick “relax time” after bouncing (Figure 6).
0	11111111	0	Only 8 or more adjacent “1s” set the output to high. First incoming “0” triggers the output to low (Figure 5).	Only 8 or more ticks of “1” produce the output pulse (Figure 5).	At bouncing, the output is in initial state, there is an 8-tick “latent time” (delay between first non-bouncing pulse and output trigger). (Figure 7.)
1	00000000	0	Only 8 or more adjacent “0s” set the output to high. First incoming “1” triggers the output to low.	Only 8 or more ticks of “0” produce the output pulse.	At bouncing, the output is in initial state, there is an 8-tick “latent time” (delay between first non-bouncing pulse and output trigger).
1	11111111	1	First “0” triggers the output to low for 8 ticks, every subsequent “0” prolongs the low state for 8 ticks.		At bouncing, the output is active, there is an 8- tick “relax time” after bouncing.

Important Notes

1. All reactions are delayed by 1 tick of the PRS clock as shown in figures 4, 5, 6 and 7 (a feature of the internal comparing circuit).
2. The input is sampled by a rising edge of the PRS clock.
3. The input is inverted by setting bit 7 in the DxBxxFN register (alias, PRS_FUNC_REG) to “1” in order to obtain “normal” (i.e., transparent) input logic. You must not modify this register to obtain the “inverted” input logic.
4. Also, you can use the “Less Than” compare function with a Seed Value=11111111 to obtain inverted output logic.

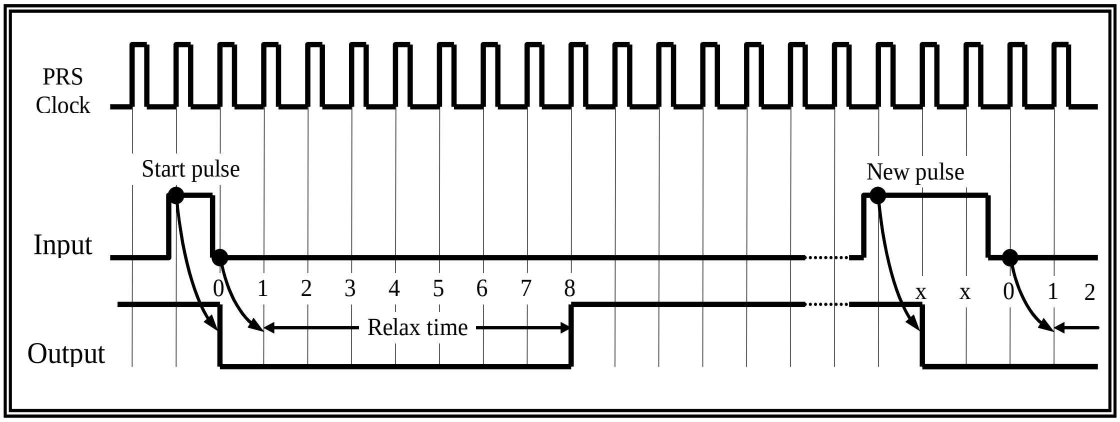

Figure 4. One-Shot Operation (Seed Register=00000000)

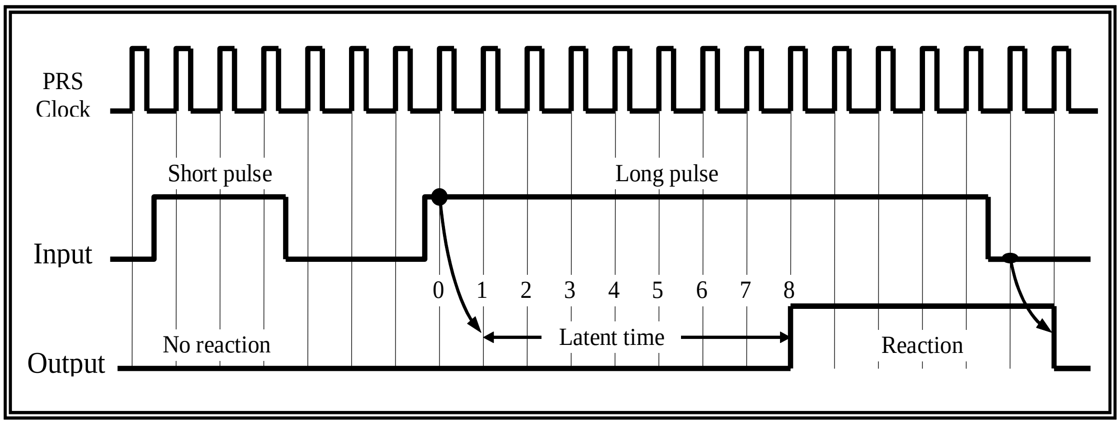

Figure 5. Pulse Width Selection (Seed Register=11111111)

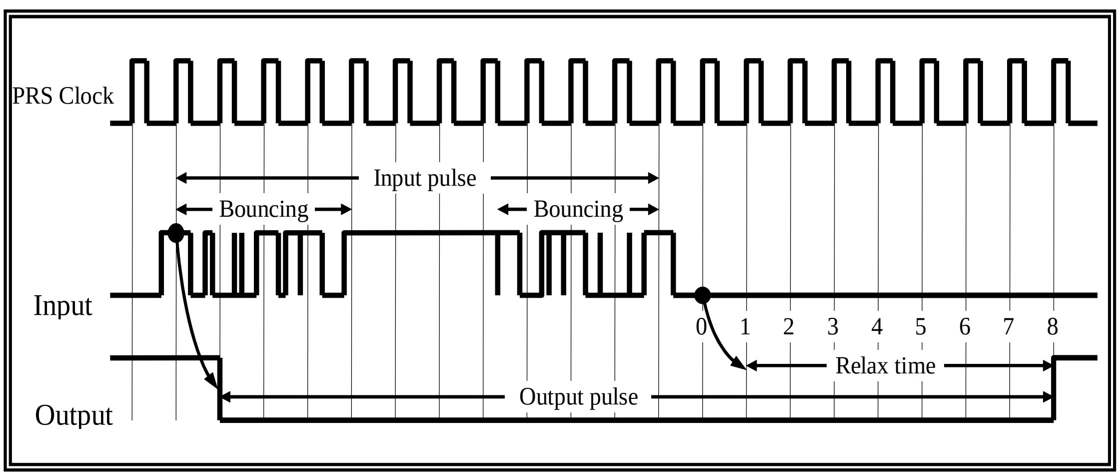

Figure 6. Pulse Width Debouncing (Seed Register=00000000) Output Pulse is “Longed”

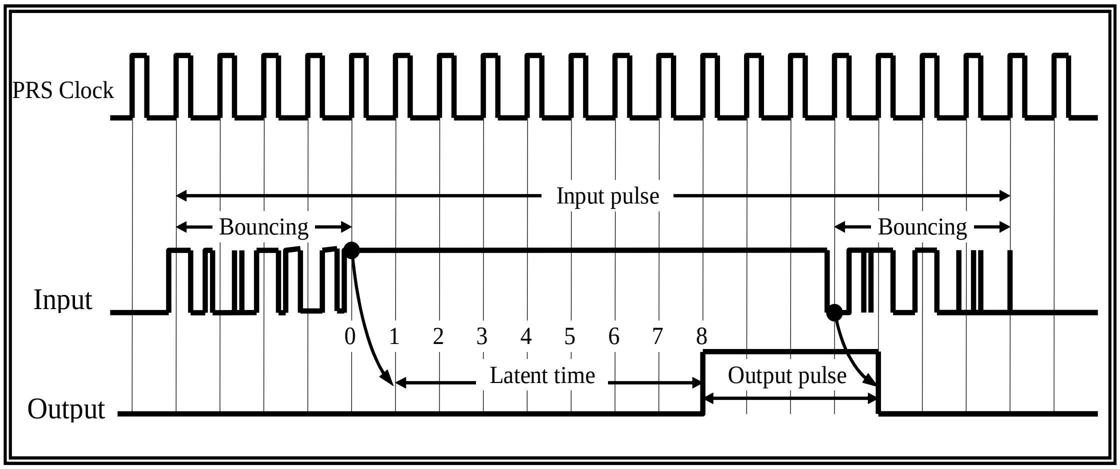

Figure 7. Pulse Width Debouncing (Seed Register=11111111) Output Pulse is "Shorted"

Notes

Why is the PRS User Module used instead of the other CRC- and PRS-based user modules such as the CRC16 User Module or the DigBuf User Module?

The answer is that the PRS User Module allows the user to choose 8-, 16-, 24- or 32-bit capacity to obtain a required clock reaction time.

Of course, if you are going to use a 16-bit PRS, the better choice is a CRC16 Generator User Module. (The DigBuf is exclusively 8 bits.) With the CRC, the data input can easily be reconfigured in the Device Editor.

Also note that a special instruction sequence (in, for instance, *main.asm*) can eliminate the long initial (power-on) output pulse of the one-shot with a Seed register value of 11111111. But the one-shot with a Seed register value of 00000000 always starts with a one-tick reaction. The user should be prepared to implement this, if applicable.

One important note, data input is sampled by a rising edge of the PRS clock. If the input pulse is too short and does not coincide with a rising edge, it is ignored. This is not a problem if you use a PRS clock-based (or slower) input signal. In other cases, you should anticipate probable losses.

A second important note, a CRC- and PRS-based user module should not be placed in the high adjacent position of the PSoC digital array. This type of user module uses the internal, common MSB tri-state bus and will disturb one-shot operation.

Testing

I demonstrated the one-shot operation in the best traditions of Dr. Robert Wood, the original American Experimental Physicist. Do not worry. I did not use a cat as a spectrometer detector or an electrostatic power source; the cat only observed the LED's blinking and gave me a response.

To test, create the circuit as shown in figures 8 and 9 then program the part according to the associated project. I used a 4-position switch, tricolor LEDs with resistors, the proposed circuit board, and a socket.

The additional inverter between PWM8_1 and the output is needed to make the polarity of the output pulses identical. The sequences from ports P2[7], P2[5], P2[3] and P2[1] have a sample rate of 8 Hz and are 16 bits in length. The "bouncing" with a sub-pulse period of 0.25s can easily be seen.

P2[7] Output – Short Pulses
`10000000000000001000000000000000...`

P2[5] Output – Long Pulses
`11111111000000001111111100000000...`

P2[3] Output – Short Pulses with "Bouncing"
`10110100000000001011010000000000...`

P2[1] Output – Long Pulses with "Bouncing"
`10101111111100001010111111110000...`

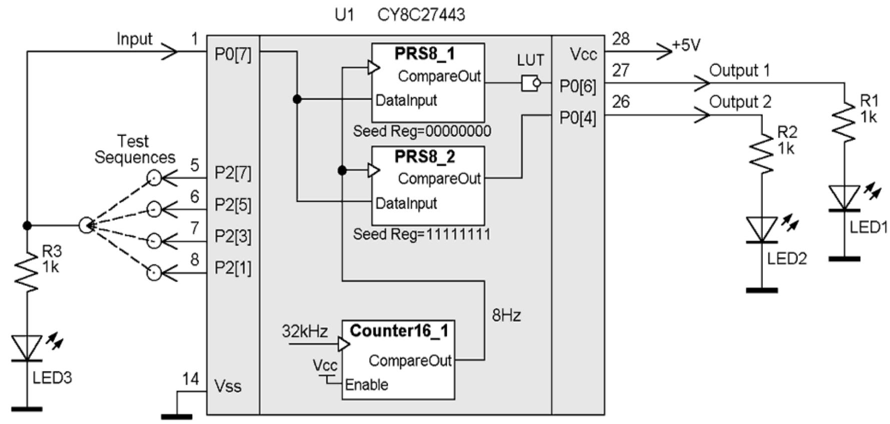

Figure 8. PSoC Internals and Schematic

Test Results

The reactions to the test sequences were as follows:

- P2[7] - LED1 responded with an 8-tick flash duration. LED2 had no response (input pulses are too short).
- P2[5] - LED1 responded with a 15-tick flash duration. LED2 responded with a 1- tick flash duration.
- P2[3] - LED1 responded with a 13-tick flash duration. LED2 had no response (input pulses are too short).
- P2[1] - LED1 stayed lit continuously (the intervals between pulses were too short). LED2 responded with a 1-tick flash duration.

Note that the PRS responded with a single pulse, despite the signal bouncing (i.e., false triggers).

Conclusion

The CRC- and PRS-based user modules as a one-shot pulse width discriminator and debouncer allow the user to reshape pulses in different applications. Of course, this is for such applications where high-frequency clocks are usually used.

Note that there are a few disadvantages:

- Only 8 bits may be used.
- CRC- and PRS-based user modules should not be placed in the high adjacent position of the digital block array.
- The skipping of short pulses and pauses does not coincide with the clock's rising edge.

Figure 9. Testing

About the Author

Name: ILYA Mamontov
Title: Electronic Engineer
Background: Design, modernization, repair and technical service of measurement devices.
Contact: ilka@elsite.ru Subject: PSOC AN
or
illinoys@narod.ru Subject: PSOC AN

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641

http://www.cypress.com/