

CY8C29/27/24/22xxx and CY8C26/25xxx Data Sheet

Resources	PSoC™ Blocks			API Memory (Bytes)		Pins
	Digital	Analog CT	Analog SC	Flash	RAM	
CY8C29/27/24/22xxx						
8-bit	1	0	0	8	0	2
CY8C26/25xxx						
8-bit	1	0	0	8	0	2

Features and Overview

- Data input clocking up to 48 MHz
- Data Input can be inverted
- 3 variations: One-shot Classic, Pulse Width Discriminator and Pulse Width Discriminator with Inverted Output

A one-shot (also known as a “univibrator” or “monostable multivibrator”) is a device that produce a single pulse in response to an input signal. It can be used for reshaping of input pulses and generation of single pulses with a required duration.

A Pulse Width Discriminator produces an output pulse in response to an input pulse of a certain duration. Input pulses less than the specified duration are ignored.

Both the One-shot and Pulse Width Discriminator can be used for de-bouncing. Usually the source of bounce is a comparator that creates multiple triggers when the input levels are crossing. If this signal is routed to a Counter User Module, false triggers will be counted. Typical usage of a One Shot User Module is between a Comparator’s output and a Counter’s input.

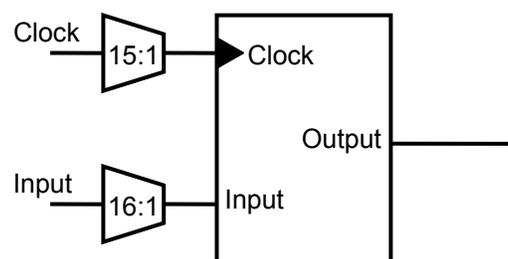


Figure 1: OneShot Block Diagram

Functional Description

Note. Incoming pulses are sampled by the rising edge of the clock. If the input pulse is not coincident with a clock rising edge then it will be ignored.

Operation of the One-shot Classic is shown in **Figure 2**. The incoming data to the Input is sampled by the rising edge of the Clock. The first incoming "0" triggers the output low for 8 cycles; then each following "0" prolongs the low state for 8 cycles. All reactions are delayed by one cycle.

Operation of the Pulse Width Discriminator is shown in **Figure 3 and 4**. The incoming data to the Input is sampled by a rising edge of Clock. Only eight or more of adjacent "0" set the output to low, the first incoming "1" triggers the output to high. All reactions are delayed by one cycle.

Operating of One-shot Classic and Pulse Width Discriminator as a de-bouncing unit is shown in **Figure 5 and 6**

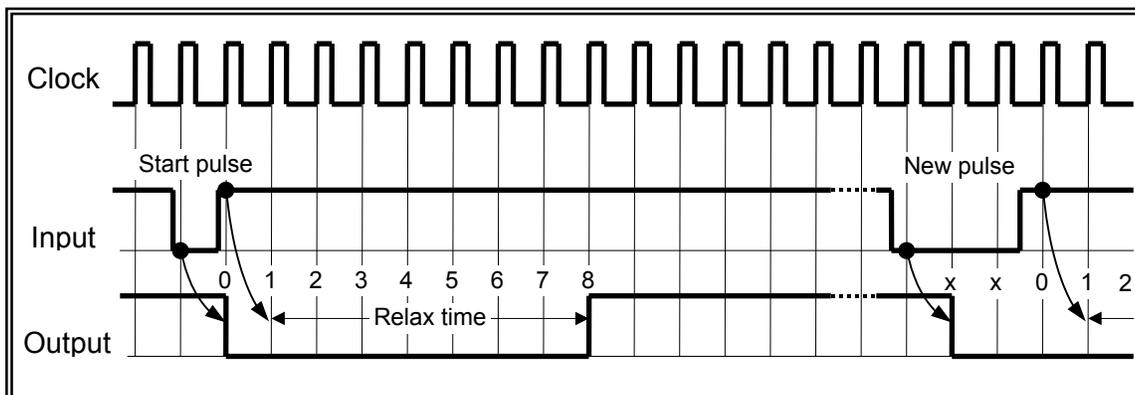


Figure 2: "One-shot Classic" Mode. Output pulse is longer than input pulse.

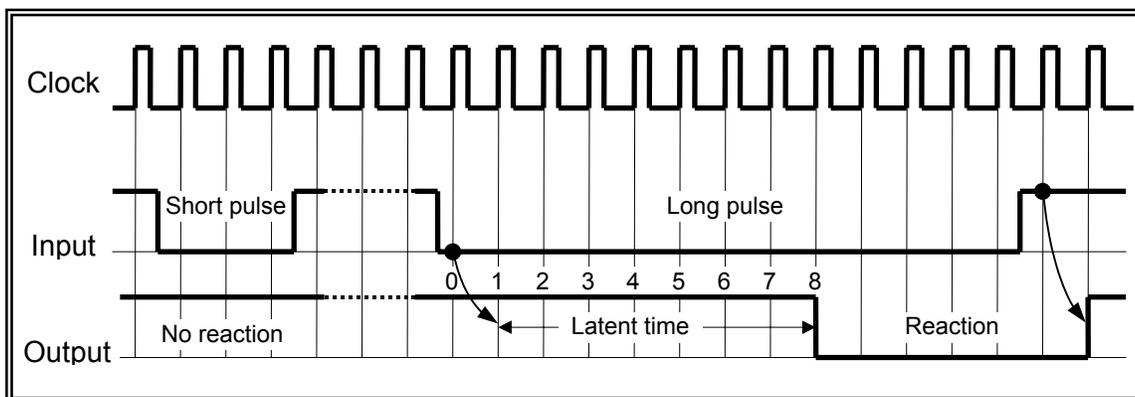


Figure 3: "Pulse Width Discriminator" Mode. Output pulse is shorter than input pulse.

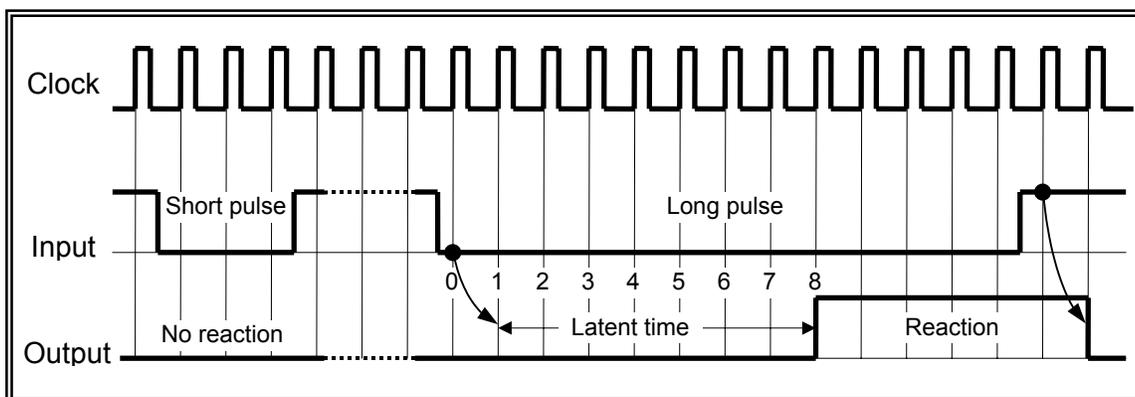


Figure 4: "Pulse Width Discriminator (Inverted)" Mode. Same as previous, output is inverted.

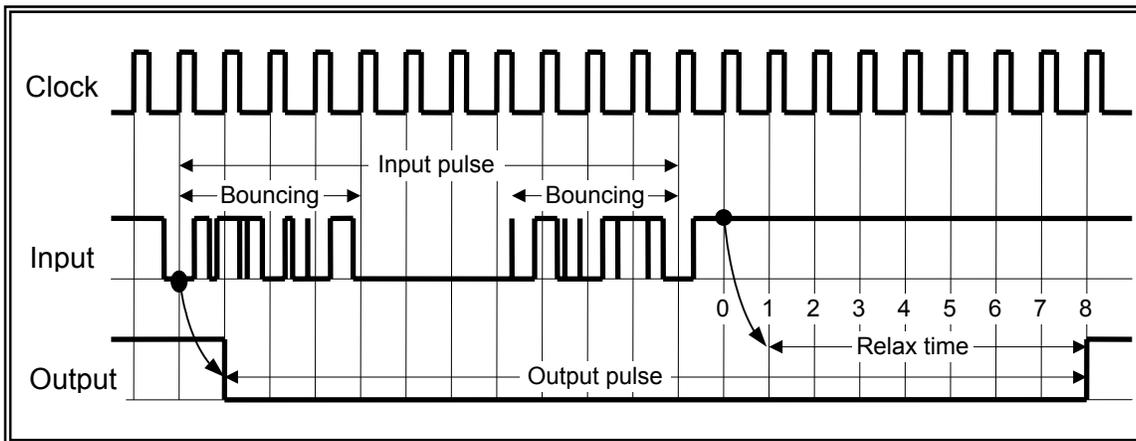


Figure 5: Pulse De-bouncing Diagram ("One-shot Classic" Mode. Output pulse is prolonged)

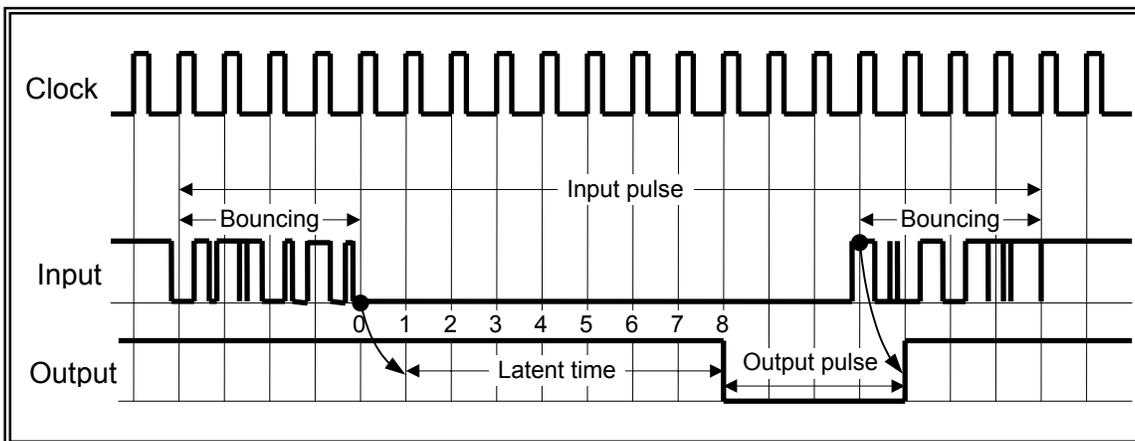


Figure 6: Pulse De-bouncing Diagram ("Pulse Width Discriminator" Mode. Output pulse is shorted)

Placement

The One-shot consumes one Digital PSoC block per 8 bits of resolution. Each block is given a symbolic name by the Device Editor during placement. The API qualifies all register names with the user assigned name.

An additional consideration, in selecting a placement location, is that the OneShot User Module should not be placed in a low adjacent position near any CRCPRS-based module (PRS, CRC16 or another OneShot).

Parameters and Resources

Once a One-shot User Module has been selected and placed using the Device Editor, the following parameters may be set:

Clock

The Clock parameter is selected from 1 of 15 sources. These sources include the 48 MHz oscillator (5.0V operation only), lower frequencies (24V1 and 24V2) divided down from the 24 MHz system clock, other PSoC blocks, and external inputs routed through global inputs and outputs.

ClockSync

This parameter sets the ClockSync type "SysClk Direct", "Sync to SysClk", "Sync to SysClk*2" or "Unsynchronized".

Input

The Input parameter is selected from 1 of 16 sources.

Output

The Output may be routed to one of four Global Output buses or disabled.

Type

This parameter sets the operating type "One-shot Classic", "Pulse Width Discriminator" or "Pulse Width Discriminator (Inverted)" as described in the functional description above.

InvertInput

This parameter gives the user the option to invert the Input.

Application Programming Interface

The Application Programming Interface (API) routines are provided as part of the user module to allow the designer to deal with the module at a higher level. This section specifies the interface to each function together with related constants provided by the "include" files.

8-Bit One-Shot API

OneShot8_Start

Description:

Enables the OneShot8 User Module for operation.

C Prototype:

```
void OneShot8_Start(void)
```

Assembler:

```
call OneShot8_Start
```

Parameters:

None

Return Value:

None

Side Effects:

Produces a single output pulse.

OneShot8_Stop

Description:

Disables the OneShot8 User Module.

C Prototype:

```
void OneShot8_Stop(void)
```

Assembler:

```
call OneShot8_Stop
```

Parameters:

None

Return Value:

None

Side Effects:

None