

## Модуль PRS как одновибратор, селектор импульсов и подавитель «звона»

Автор: Илья Мамонтов  
 Сопровождающий проект: Есть  
 Семейство микросхем: CY8C27xxx  
 Версия PSoC Designer: 4.2

Ссылки на дополнительные документы: AN2108, AN2231

### Резюме

Не хватает скорости для программного формирования импульсов? Тогда эта статья для вас! Используя модуль PRS с особой конфигурацией, вы получаете интересные решения для ваших задач. В этом документе описывается способ реализации одновибратора, селектора импульсов по длительности и устройства подавления «звона» с помощью аппаратных средств PSoC.

### Введение

Иногда при разработке устройств бывают необходимы узлы, формирующие в ответ на входной сигнал одиночный выходной импульс. Такие узлы называются одновибраторами и используются для получения задержек и формирования выходных импульсов.

Одновибраторы также используются и для подавления «звона». Источником последнего являются не только механические переключатели и кнопки. Взгляните на **рис. 1** и **2**.

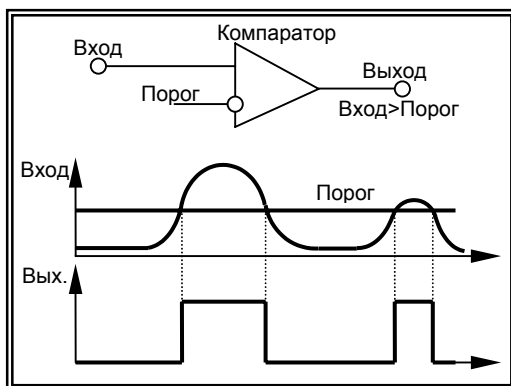


Рис. 1 Работа идеального компаратора

Входной сигнал имеет относительно пологие передние и задние фронты. Компаратор должен среагировать на приходящий импульс

одиночным переключением, как это показано на **рис. 1**.

В реальности же входному сигналу (а также напряжению порога) присуща шумовая компонента, и кроме того, компаратор имеет собственный шум. В результате, когда уровни сигналов пересекаются, компаратор многократно переключается туда-сюда (**рис. 2**). Это явление называют «звоном».

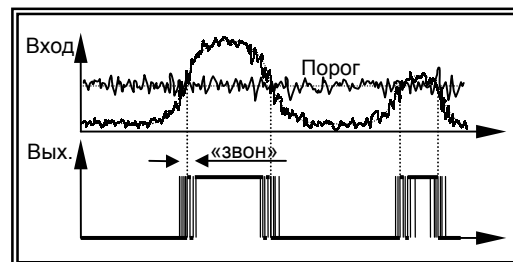


Рис. 2 То, что получается на практике

Если следующий узел – счетчик, то эти ложные переключения будут подсчитаны.

Одним из способов подавления ложных переключений является добавление гистерезиса. Документ AN2108 («Гистерезис для компаратора в PSoC») описывает этот метод, однако трудности трассировки внутри PSoC не всегда позволяют им воспользоваться (например, чтобы улучшить

работу дискриминатора импульсов, описанного в AN2231).

Альтернативный способ – использовать одновибратор. Когда приходит первый импульс, такое устройство формирует выходной сигнал заданной длительности, во время которого любые входные импульсы игнорируются, даже если устройство будет повторно активировано «звоном». В результате мы получим на выходе очень чистый сигнал.

## Реализация на PSoC

Изучая возможности работы модуля PRS в качестве сдвигового регистра, я однажды обнаружил интересные особенности и попытался их как-то использовать. Эти

особенности возникают только **при нулевом значении регистра Полинома (Polynomial Register)**.

При нулевом значении внутренняя структура модуля PRS становится такой, как это показано на **рис. 3**.

Здесь сдвиговый регистр (Shift Register) уже не имеет выхода данных (обозначаемый как «OutputBitStream»), а вход данных DATA проинвертирован элементом «исключающее ИЛИ» (из-за того, что на втором входе этого элемента образуется лог. 1). Замечу, что модуль PRS не имеет отдельного конфигурируемого входа данных, так как этот вход всегда принудительно подключен к лог.0 для правильной работы модуля в исходном варианте.

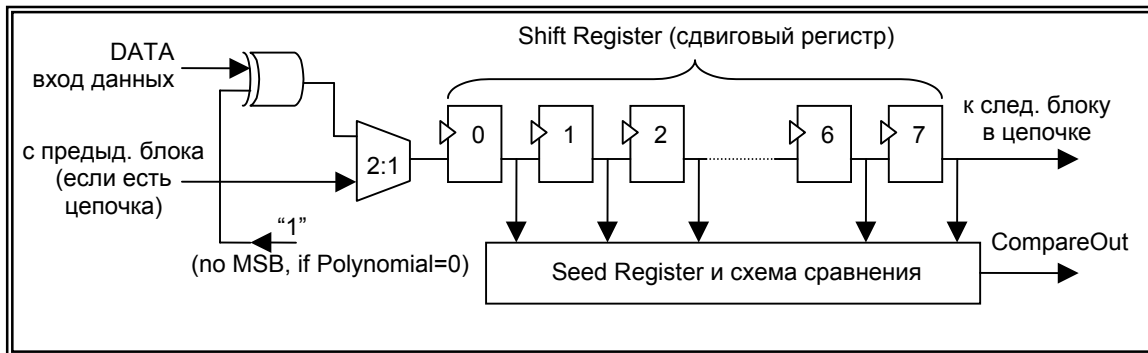


Рис. 3 Структура CRCPRS блока (Polynomial Register=00000000)

Чтобы подключить вход данных и исправить его инверсность, надо запрограммировать соответствующие регистры, например:

```
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 operating)
or reg[PRS8_1_FUNC_REG],0x80
M8C_SetBank0
```

где PRS8\_1\_INPUT\_REG – это альтернативное имя регистра «DxBxxIN».

Вот все возможные варианты подключений (с кодовыми значениями для команды «OR»):

Код	Подключение
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

Также в Регистр Сравнения («Seed Register») должно быть записано значение **00000000** или **11111111**, а тип функции сравнения («Compare Function») должен быть установлен «равно» («Equal»).

Одновибратор работает следующим образом:

1. По тактовым импульсам сдвиговый регистр («Shift Register») заполняется однородной последовательностью нулей или единиц (определяется состоянием входа данных).

2. Схема сравнения устанавливает на своем выходе 0 или 1 (определяется значением регистра сравнения «Seed Register»). Это есть «стабильное» (иначе говоря – «исходное») состояние одновибратора.

3. В зависимости от меняющихся входных данных, сдвиговый регистр будет заполняться

по-разному, а схема сравнения будет реагировать на кодовые комбинации выставлением 1 или 0.

4. После того, как на входе опять установится исходный логический уровень, сдвиговый регистр снова заполнится исходной последовательностью (возврат в «стабильное» состояние).

Различные варианты работы 8-битного PRS-модуля представлены в табл. 1, диаграммы работы – на рис 4, 5, 6 и 7.

Таблица 1 Варианты работы 8-битного PRS-модуля

Вход (исх. сост.)	Seed Register	Выход (исх. сост.)	Работа в качестве одновибратора	Работа в качестве селектора	Работа в качестве подавителя «звона»
0	00000000	1	Первая же 1 переключает выход в «0» на 8 тактов, каждая следующая 1 продлевает это состояние на 8 тактов (рис. 4)	–	При «звоне» выход активен, есть также 8-тактовое «время восстановления» после окончания «звона» (рис. 6)
0	11111111	0	Только 8 и более подряд единиц утанавливают выход в «1», первый же 0 сбрасывает выход в исх. состояние (рис. 5)	Только высокий уровень длиной 8 тактов и более вызывает вых. импульс (рис. 5)	При «звоне» выход остается в исх. состоянии, есть 8-тактовое «латентное время» (задержка до момента перехода в активное состояние) (рис. 7)
1	00000000	0	Только 8 и более подряд нулей утанавливают выход в «1», первая же 1 сбрасывает выход в исх. состояние	Только низкий уровень длиной 8 тактов и более вызывает вых. импульс	При «звоне» выход остается в исх. состоянии, есть 8-тактовое «латентное время» (задержка до момента перехода в активное состояние)
1	11111111	1	Первый же 0 переключает выход в «0» на 8 тактов, каждый следующий 0 продлевает это состояние на 8 тактов	–	При «звоне» выход активен, есть также 8-тактовое «время восстановления» после окончания «звона»

Примечания.

1. Все выходные перепады задерживаются на 1 такт (особенность работы схемы сравнения, см. рис. 4, 5, 6 и 7).
2. Состояние входа опрашивается по фронту тактового сигнала PRS Clock.
3. Вход данных проинвертирован записью лог.1 в бит 7 регистра DxVxxFN (альтернативное имя – PRS\_FUNC\_REG), чтобы получить «прозрачную» (т.е. без инверсий) логику работы. Чтобы использовать инверсный входной сигнал, вы можете не модифицировать этот бит.
4. Также со значением регистра Seed Register=11111111 вы можете использовать функцию сравнения «меньше чем» («Less Than»), чтобы получить инверсный выходной сигнал.

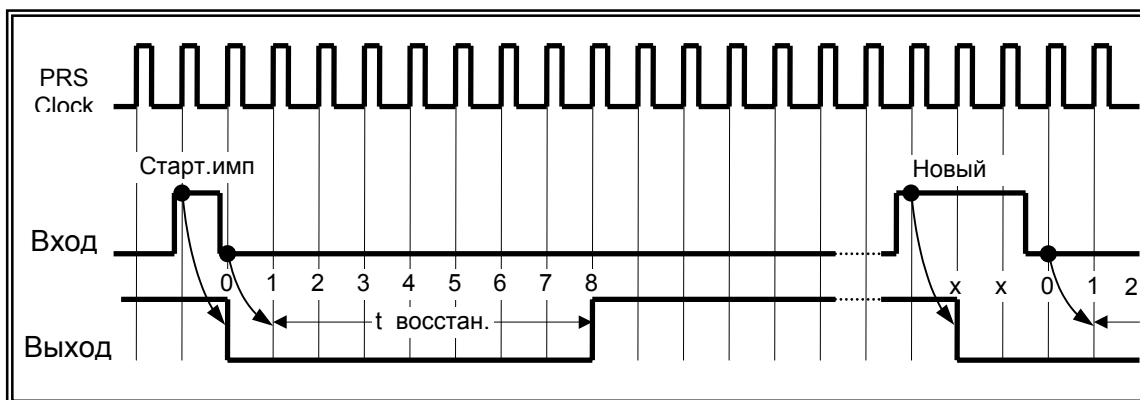


Рис. 4 Диаграмма работы PRS в качестве одновибратора (Seed Register=00000000)

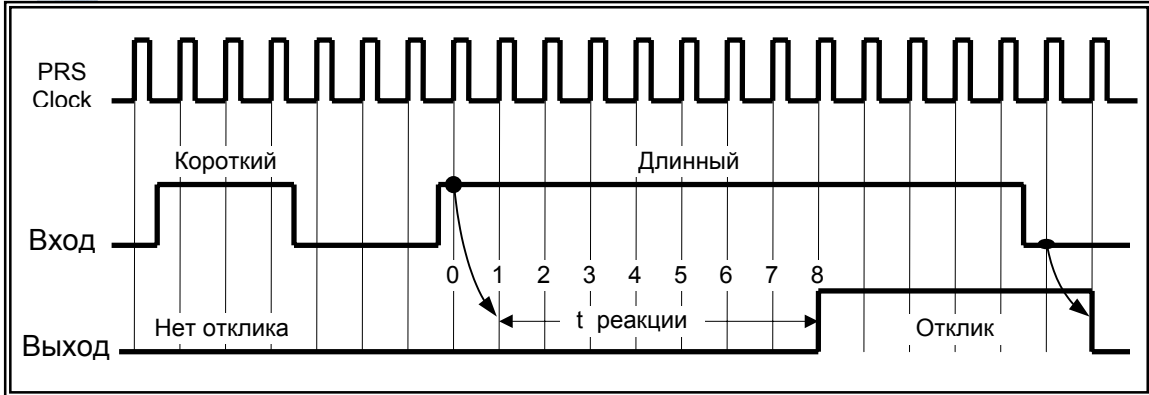


Рис. 5 Работа PRS в качестве дискриминатора импульсов разной длительности (Seed Register=11111111)

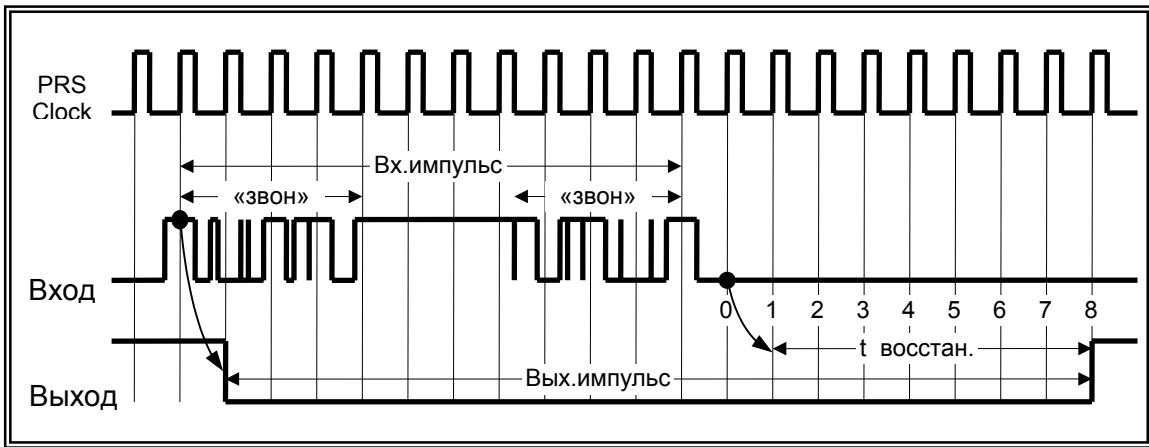


Рис. 6 Работа PRS в качестве подавителя «звона», Seed Register=00000000 (вых. импульс удлиняется)

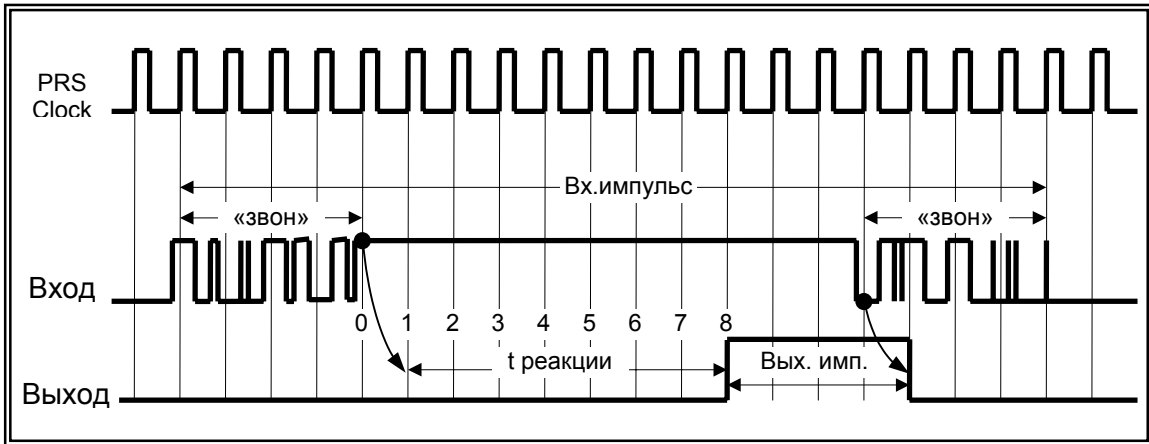


Рис. 7 Работа PRS в качестве подавителя «звона», Seed Register=11111111 (вых. имп. укорачивается)

Полный код для конфигурирования модуля PRS вы можете посмотреть в файле *main.asm* Сопровождающего Проекта.

## Примечания

Почему использованы модули PRS, а не другие, основанные на CRCPRS-функции, например CRC16 или DigBuf?

Ответ таков: семейство этих модулей позволяет выбирать разрядность 8, 16, 24 и 32 бит, исходя их получения требуемой продолжительности реакции (в тактах).

Конечно, если вы хотите использовать разрядность 16 бит, то лучший выбор – модуль CRC16. Модуль DigBuf имеет 8 разрядов и позволяет легко конфигурировать вход данных средствами редактора PSoC Designer, но сам модуль должен быть переконфигурирован из подрежима «Pass function» в режим PRS.

Замечу также, что с помощью определенной последовательности конфигурирования (как это сделано в файле *main.asm*) можно добиться отсутствия длинного вых. импульса при подаче питания на микросхему (в случае, когда SeedRegister = 11111111), но одновибратор со значением SeedRegister = 00000000 стартует с коротким (однотактовым) импульсом. Будьте готовы к такому поведению.

Важный момент: состояние входа данных опрашивается по переднему фронту тактового сигнала PRS Clock. Если входной импульс будет настолько короткий, что на него не попадет фронт тактового сигнала, то такой импульс будет пропущен. Если входные импульсы получаются из тактовых сигналов или более длительны, то проблем быть не должно. В других случаях оцените последствия возможных пропусков.

Второй важный момент – в соседних, более старших позициях матрицы цифровых блоков не должно находиться никаких модулей, основанных на CRCPRS-функции. Эти модули используют совместную шину «MSB tri-state bus» и будут нарушать работу наших узлов.

## Тестирование

Я попытался продемонстрировать работу получившегося одновибратора в лучших традициях д-ра Роберта Вуда. Не беспокойтесь, я не собираюсь использовать кошку в качестве прочистителя спектрометров или как электростатический источник питания. Он (кот) будет просто наблюдать за миганием светодиодов и подавать мне знаки.



Рис. 8 Тестирование

Для тестирования соберите схему по рис. 8 и 9 и запрограммируйте PSoC микросхему согласно Сопровождающему Проекту. Я использовал 4-позиционный переключатель, три цветных светодиода с резисторами, плату Invention Board и контактные панельки.

Дополнительный инвертор между модулем PWM8\_1 и выходом микросхемы необходим для того, чтобы получить одинаковую полярность выходных импульсов. Тестовые последовательности с ножек P2[7], P2[5], P2[3] и P2[1] имеют темп 8 Гц и длину 16 бит. «Звон» с периодом 0,25 сек. визуально легко различим.

Выход P2[7] – короткие импульсы:  
**1**000000000000000**1**000000000000000...  
 Выход P2[5] – длинные импульсы:  
**11111111**00000000**11111111**00000000...  
 Выход P2[3] – короткие импульсы со «звоном»:  
**101101**00000000**101101**00000000...  
 Выход P2[1] – длинные импульсы со «звоном»:  
**101011111111**0000**101011111111**0000...

Реакция на тестовые последовательности:

P2[7] – светодиод LED1 загорается на время, равное 8 тактам, LED2 молчит (вх. импульсы слишком короткие);

P2[5] – LED1 загорается на время 15 тактов, LED2 загорается на 1 такт;

P2[3] – LED1 загорается на время 13 тактов, LED2 молчит (вх. импульсы короткие);

P2[1] – LED1 постоянно горит (интервалы между импульсами слишком короткие), LED2 загорается на 1 такт.

Заметьте, что модули PRS выдают одиночные импульсы, несмотря на то, что входные импульсы имеют несколько ложных переключений.

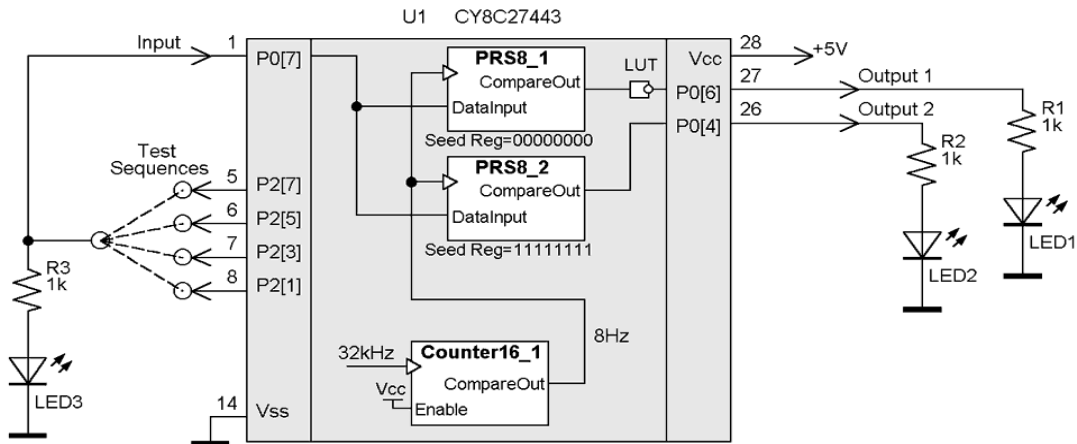


Рис. 9 Внутренняя структура PSoC и электрическая схема для тестирования

## Заключение

Модули, основанные на CRCPRS функции, позволяют строить узлы для различных применений, где требуется аппаратное формирование импульсов, например, для задания задержек, для селекции импульсов по длительности, а также для подавления «звона». Естественно – для тех приложений, где в основном используются мегагерцовые частоты.

У предлагаемых решений существуют также некоторые недостатки:

- может быть использована только разрядность, кратная 8;
- не разрешается размещение в соседних (старших) позициях модулей, основанных на CRCPRS функции,
- пропуск коротких импульсов (или пауз между ними), если они не приходятся на передний фронт тактового сигнала.

---

## About the Author

Name: ILYA MAMONTOV  
Title: Electronic Engineer  
Background: Design, modernisation, repair and technical  
service of measurement devices  
Contact: [illinoys@narod.ru](mailto:illinoys@narod.ru) subj: PSOC AN

---

Cypress MicroSystems, Inc.  
2700 162<sup>nd</sup> Street SW, Building D  
Lynnwood, WA 98037  
Phone: 800.669.0557  
Fax: 425.787.4641

<http://www.cypress.com/> / [http://www.cypress.com/aboutus/sales\\_locations.cfm](http://www.cypress.com/aboutus/sales_locations.cfm) / [support@cypressmicro.com](mailto:support@cypressmicro.com)

Copyright © 2003 Cypress MicroSystems, Inc. All rights reserved.

PSoC™ (Programmable System-on-Chip™) is a trademark of Cypress MicroSystems, Inc.

All other trademarks or registered trademarks referenced herein are property of the respective corporations.

The information contained herein is subject to change without notice.