Application Note AN2xxx

Генерирование абсолютно случайных последовательностей

Автор: Илья Мамонтов
Сопровождающий проект: Есть
Семейство микросхем: CY8C22/24/25/26/27/29xxx
Среда разработки: PSoC Designer 4.2
Ссылки на дополнительные документы: AN2041, AN2224

Если хочешь рассмешить Бога – расскажи ему, что изобрел Генератор Случайных Чисел!
Перефразированная поговорка

Резюме

В этой статье описывается способ построения генератора абсолютно случайной последовательности на основе аналогового блока PSoC. Этот новый модуль может быть вовлечен в любой проект как источник случайных чисел, и как аппаратный генератор сигнала для аналоговых и цифровых блоков. Он может использоваться как криптостойкая альтернатива модулю PRS.

Введение

Получение случайных чисел в заведомо предсказуемых системах? Это невозможно, скажете вы и будете на 99,73% правы, если речь идет о традиционных микроконтроллерах. Я наблюдал обсуждение этой проблемы на разных интернет-форумах, где предлагалось использовать измерение нестабильности встроенных систем ФАПЧ, результатов АЦП, нестабильность времени зарядки/разрядки внешнего конденсатора и т.д.

Используя же «продвинутые» микроконтроллеры PSoC, сделать это очень просто!

Реализация

Классическим способом получения случайного сигнала считается взять какой-нибудь источник шумового сигнала и усилить его до нужного значения. Таковым может быть тепловой шум резистора или, например, специфический шум полупроводников.

Подробно об источниках шума в чипах PSoC можно почитать в документе AN2224 – «Lower Noise Continuous Time Signal Processing with PSoC». Замечу, что это вредное по своей природе явление, в данном случае будет работать нам на пользу. Поэтому делайте все противоположное, что рекомендуется этим документом в качестве мер снижения шума. На самом деле это шутка, поскольку на кристалле могут располагаться другие Модули Пользователя, требующие минимизации шумов и, соответственно, применения этих правил.

Основная проблема, которую необходимо решить, – получить шумовой сигнал достаточной амплитуды.

Напряжение шума аналоговых блоков (CTB и SCB) невелико даже при установке максимальных коэффициентов усиления этих блоков. Если же соединить каскадом несколько усилительных модулей, то ошибка смещения первого каскада, многократно усилясь, приведет к насыщению последнего каскада, что потребует применения специальных антимер (усложнения топологии, применения внешних компонентов).

Я предлагаю использовать более элегантный способ (см. Рисунок 1), основанный на особой конфигурации модуля SCBLOCK (Generic Switched Capacitor Block).

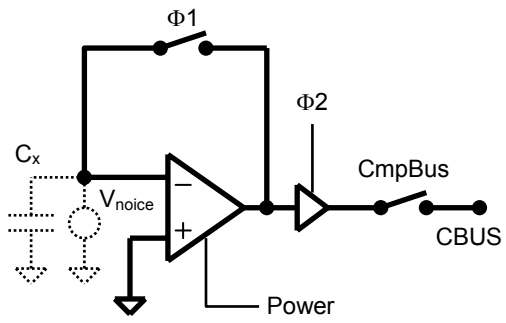

Рисунок 1. Внутренняя структура генератора случайной последовательности

Чтобы получить такую конфигурацию, разместите Модуль Пользователя SCBLOCK в любой свободной позиции той колонки, где не задействована шина Analog Comparator Bus (CBUS). Далее задайте параметры модуля в соответствии с Таблицей 1 (красным помечены те параметры, которые существены для работы модуля в качестве генератора случайной последовательности; значения же остальных параметров можно задать любыми).

Таблица 1. Параметры модуля SCBLOCK

SCBLOCK_2

User Module Para	Value
FCap	32
ClockPhase	Norm
ASign	Pos
ACap	0
AMux	ACB01
BCap	0
AnalogBus	Disable
CompBus	ComparatorBus_1
AutoZero	On
CCap	0
ARefMux	AGND
FSW1	Off
FSW0	Off
BSW	Off
BMux	ACB01
Power	High

Выходной сигнал, снимаемый с шины Comparator Bus, представляет собой случайную последовательность нулей и единиц. Использовать ее можно трояко:

- программно опрашивать состояние шины Comparator Bus для считывания случайных значений;
- как входной сигнал для цифровых Модулей Пользователя;
- как входной сигнал для аналоговых Модулей Пользователя, в качестве модулирующего сигнала (справедливо для модулей, в состав которых входят SC блоки типа «C».

Кроме того, в устройствах серии CY8C24x94 возможна прямая подача сигнала с шины Comparator Bus на шину Global Output Bus. Об особенностях использования выходного сигнала будет рассказано ниже.

Как же работает такой модуль? Вы вероятно уже знаете, что SC блок управляется двумя неперекрывающимися сигналами-фазами: Φ1 и Φ2. В течении фазы Φ1, называемой подготовительной (или «опросной»), производится заряд/разряд внутренних конденсаторов блока, а в течение фазы Φ2, называемой «маршевой», происходит перераспределение заряда между конденсаторами с формированием результирующего напряжения. Более подробно о работе SC блока можно почитать в документе AN2041 – «Understanding Switched Capacitor Blocks».

В нашем случае все конденсаторы отключены, за исключением внутренних паразитных емкостей, которые условно приведены ко входу в виде единой C_x. Также все источники шума приведены ко входу в виде единого генератора V_{noice}.

В итоге мы имеем операционный усилитель с очень большим коэффициентом усиления, к выходу которого подключен дополнительный компаратор, работающий на шину CBUS. Малое смещение выходного напряжения обеспечивается цепью автоматической коррекции нуля на основе ключа Φ1 и паразитной емкости C_x.

Усиленный шум, среднее значение которого «подтянуто» к уровню «аналоговой земли» (AGND) заставляет компаратор переключаться случайным образом, давая на выходе желаемую случайную последовательность.

В Сопровождающем Проекте приведен пример такой конфигурации для четырех блоков, расположенных в четырех различных колонках кристалла CY8C27443 (см. Рисунок 2). Выходные двоичные сигналы выведены на ножки P0[2], P0[3], P0[4] и P0[5] и имеют битрейт F=500 кбит в секунду. Ножки выбраны из соображения легкости клонирования проекта под 8-выводные модификации чипов. Имея четыре источника случайного сигнала, вы легко можете убедиться в отсутствии их взаимной корреляции.

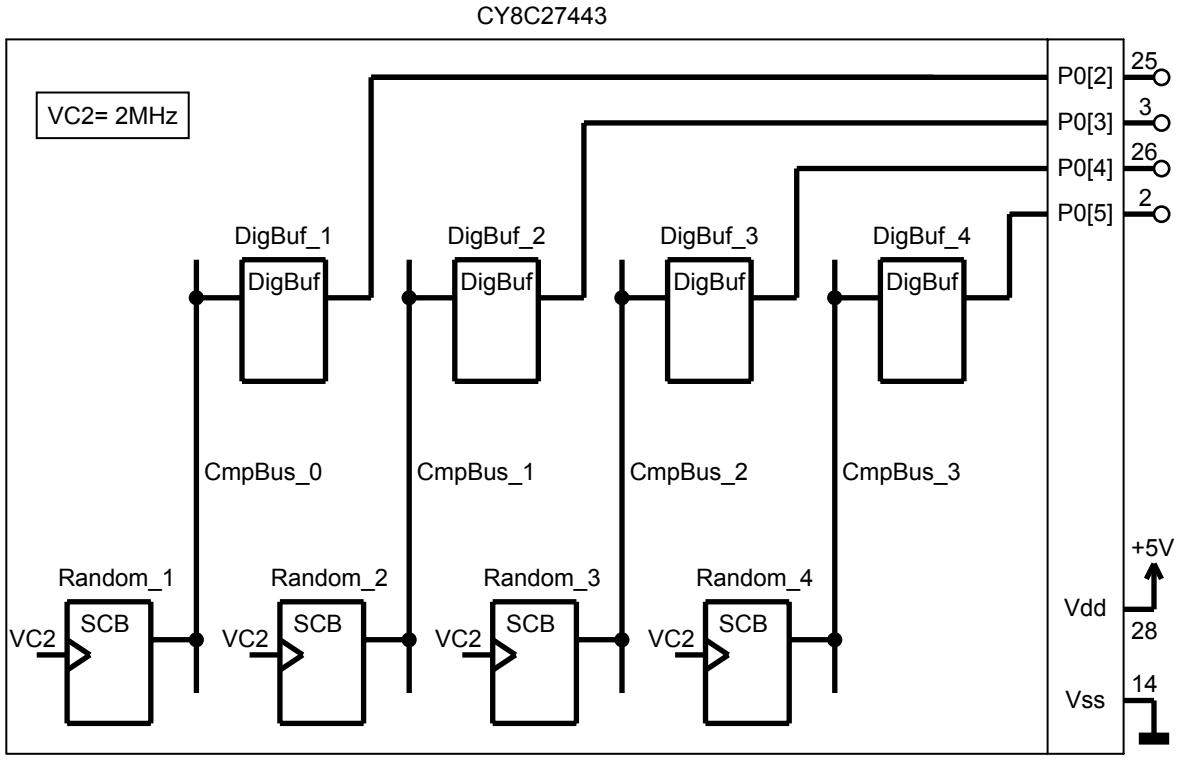

Рисунок 2. Внутренняя конфигурация CY8C27443 для тестирования модулей «Random»

Выходные сигналы имеют стандартные ТТЛ уровни, а их спектр показан на Рисунке 3. Я испытал несколько экземпляров CY8C27143 и CY8C27443; все они стабильно работали на скоростях до F=500 кбит в секунду (соответствует тактовой частоте SC блока f_{clk}=2МГц). На более высоких тактовых частотах начинает сказываться разброс параметров даже внутри одного кристалла, и модуль, хорошо работающий на одной позиции, плохо работает, либо вообще не работает на другой (нули и единицы не равновероятны).

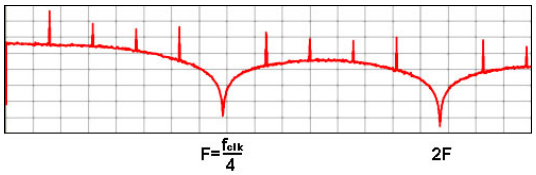

Рисунок 3. Фурье-спектр выходного сигнала (для случая F=7,8 кГц)

Параноикам читать дальше!

Если мы занимаемся криптосистемами, то можно долго спорить о том, является ли полученный сигнал абсолютно случайным или нет. Судя по игольчатым пикам на спектре (Рисунок 3) – нет. Я подозреваю, что на тепловые шумы блока накладываются шумы со стороны прочих узлов кристалла – процессора, цифровых и аналоговых модулей, вспомогательных узлов и т.д., которые по своей природе более детермистичны.

Я также провел несколько экспериментов для разных источников тактовых частот (VC1, VC2, VC3, либо взятых со счетчиков Counter) и заметил, что при некоторых комбинациях коэффициентов деления балланс между средним количеством нулей и единиц в выходном сигнале несколько нарушается. Это обнаружилось при рассматривании записанного сигнала, а также по измерению его среднего напряжения. Наилучший балланс обеспечивался при коэффициентах деления, кратных степени 2, т.е. 2, 4, 8, 16…

Тем не менее существует простой способ дополнительного «ослучаивания» исходной последовательности.

Наиболее легко это сделать с помощью Модуля Пользователя CRC16, который имеет конфигурируемый вход данных. Соедините этот вход с шиной компаратора, на которой присутствует наш случайный сигнал, а в Регистр Полинома этого модуля (Polynomial register) запишите какое-либо значение из таблицы образующих полиномов, например 0xD008, 0x8810 или другие. Тактовый вход модуля, вообще говоря, надо было бы подключать к источнику поделенной на 4 тактовой частоты, которая используется для тактирования той колонки, где расположен Генератор Случайной Последовательности. Но в принципе, подойдет любая другая частота, например частота тактирования соответствующей колонки Это только незначительно повысит предсказуемость результирующей последовательности.

Чтобы считывать 16-битные случайные числа в программу, можно воспользоваться API-шной функцией ***CRC16_ReadCRC***.

Если вам жалко тратить два «посадочных места» под модуль CRC16, воспользуйтесь 1-блочным модулем PRS8 («Генератор Псевдослучайной Последовательности»). К сожалению, этот модуль не имеет конфигурируемого в среде разработки PSoC Designer входа данных, поэтому соединяйте этот вход программно, с помощью изменения битов 7…4 в регистре управления DxBxxIN. Пример кодирования на ассемблере можно посмотреть в Листинге 1, где имя «PRS8_1_INPUT_REG» является альтернативным именем регистра DxBxxIN.

```
M8C_SetBank1
;prepare (clear) 7..4 bits
and reg[PRS8_1_INPUT_REG],0x0F
;connect to Analog column comparator 0
or  reg[PRS8_1_INPUT_REG],0x40
M8C_SetBank0
```

Листинг 1. Программирование входа данных модуля PRS8

В Таблице 1 перечислены коды для команды «OR», используемые для подключения к другим источникам сигнала.

Значения полиномов для модуля PRS8 должны выбираться, естественно, 8-битные.

Таблица 2. Варианты подключений

Код	Подключение к:
40h	Analog column comparator 0
50h	Analog column comparator 1
60h	Analog column comparator 2
70h	Analog column comparator 3

Интересное замечание. Если для генерации псевдослучайной последовательности на основе сдвигающего регистра с обратными связями кодовая комбинация «все 0» является запрещенной, то в нашем случае регистр может принимать любые значения от 00000000 до 11111111. Для чтения 8-битного случайного числа можно ипользовать функцию ***PRS8_bReadPRS***.

С использованием этих дополнительных мер спектр выходного сигнала сглаживается и становится очень близок к теоретическому (смотри Рисунок 4).

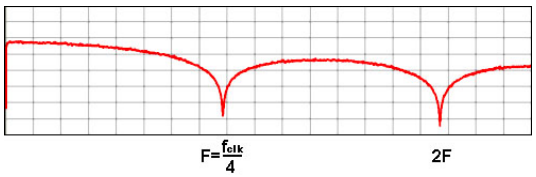

Рисунок 4. Фурье-спектр сигнала, пропущенного через модуль PRS8 (для случая F=7,8 кГц)

Можно вообще отказаться от использования цифровых Модулей Пользователя и считывать случайные числа побитно с шины Comparator Bus, как это сделано в Code 2.

```
;test state of Comparator Bus 0
;use mask 0x20, 0x40 or 0x80 for
;Comparator Bus 1, 2 or 3
tst reg[CMP_CR0],0x10
jz label_0
;random bit is 1.
    {Insert your code here}
label_0:
;random bit is 0.
    {Insert your code here}
```

Листинг 2. Получение случайных битов

Прямая подача сигнала на аналоговые блоки

Единственный способ напрямую подать случайный сигнал на аналоговые блоки – это использовать модулятор, присутствующий в блоках SCB типа «C». Так как среда разработки PSoC Designer не позволяет манипулировать этим входом напрямую, требуются некоторые усилия по программированию управляющих регистров Модулятора (AMD_CR0 и AMD_CR1). Пример программирования SC блока, расположенного в колонке 0, приведен в Листинге 3.

```
M8C_SetBank1
;Connect Modulator to Comparator Bus 0
;use code 0x02, 0x04 or 0x08 for
;Comparator Bus 1, 2 or 3
mov reg[AMD_CR0],0x04
M8C_SetBank0
```

Листинг 3. Программирование регистра Модулятора

Для блоков, расположенных в колонках 0 и 2 должен использоваться регистр AMD_CR0 (биты 3…0 и 7…4 соответственно), а для блоков, расположенных в колонках 1 и 3 – регистр AMD_CR1 (биты 3…0 и 7…4).

Если на вход такого блока подать какой-либо сигнал, то он окажется промодулированным бинарной случайной последовательностью. Чтобы просто получить двуполярную (относительно аналоговой земли AGnd) случайную последовательность, соедините вход блока с источником постоянного напряжения (например с RefHi, либо с выходом другого Модуля Пользователя, на котором присутствует постоянное напряжение).

Заключение

Генератор абсолютно случайной (либо трудно предсказуемой) последовательности можно легко реализовать средствами PSoC. В простейшем случае для этого достаточно задействовать только один блок.

Если вас очень волнуют вопросы случайности, вы можете добавить дополнительные цифровые блоки на основе сдвигающих регистров с обратными связями разрядностью до 32 бит, а также скомбинировать сигналы с нескольких «случайных» генераторов, например, используя блоки LUT с функцией «исключающее ИЛИ».

About the Author

Name: Ilya Mamontov
Title: Electronic Engineer
Background: Ilya graduated from Moscow Aviation Institute (Russian Federation) in 1989 and worked 6 years for spacecraft electronic engineering. He presently working for atomic industry, his region of interests is design, modernization, repair and technical service of measurement instruments.
Contact: ilka@elsite.ru subj: PSOC AN
or
illinoys@narod.ru subj: PSOC AN

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98087
Phone: 800.669.0557
Fax: 425.787.4641

http://www.cypress.com/