Оптимизация энергопотребления устройств на базе микроконтроллеров EFM32 Wonder Gecko с ядром Cortex-M4F

Ксения КОНДРАШОВА
xk@efo.ru

Статья посвящена способам снижения энергопотребления устройств сбора и обработки данных на базе микроконтроллеров с ядром Cortex-M4F. Приводятся основные отличительные характеристики ядра, их роль при организации эффективных алгоритмов математической обработки данных. В качестве примера рассматривается решение от компании Silicon Labs — микроконтроллеры EFM32 Wonder Gecko, обладающие высокой производительностью и сбалансированным комплексом технологий энергосбережения.

Введение

Возможности современных микроконтроллеров нередко позволяют перенести часть вычислительных функций распределенной системы на узлы, где осуществляется сбор первичных данных. Такой подход имеет ряд преимуществ, например, повышение устойчивости, характерное для децентрализованных систем, разгрузка каналов связи за счет предварительной обработки и сжатия данных. Обработка данных «на месте» оправдана при регулярном опросе аналоговых датчиков, в задачах, где входные данные имеют потоковый характер, особенно при приеме аудио или изображений. В таких приложениях накладные расходы на передачу неинформативных массивов данных на узел-обработчик исключаются в пользу передачи, сохранения или вывода лишь значимых результатов преобразования.

В то же время для организации локальной обработки данных нужно наращивать функциональность и производительность устройства, а это не может быть сделано без побочных эффектов. Для приборов с батарейным питанием самым весомым отрицательным эффектом является повышение энергопотребления.

Наиболее популярной платформой для разработки устройств сбора и обработки данных остаются микроконтроллеры на базе ядра ARM Cortex-M3, однако в линейке Cortex-M существует гораздо более эффективное с точки зрения математического аппарата процессорное ядро Cortex-M4, ассортимент микроконтроллеров с которым постепенно расширяется. В статье показано, какую выгоду по производительности и потребляемой мощности можно получить при использовании микроконтроллеров с ядром Cortex-M4 и его модификацией Cortex-M4F.

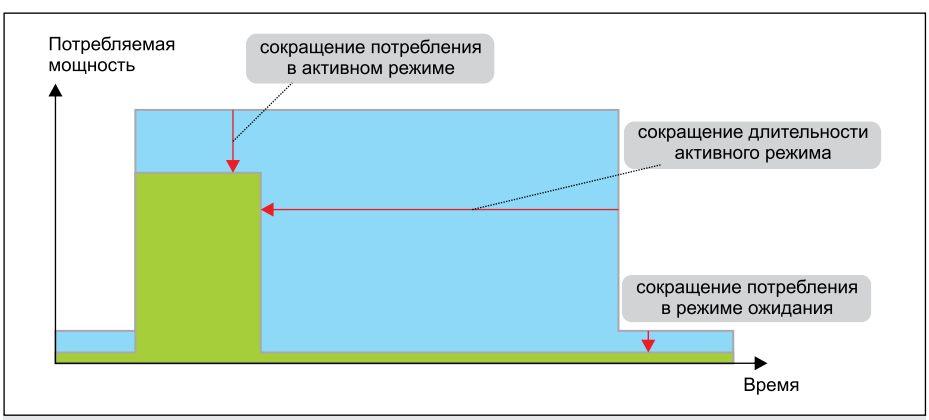

Рис. 1. Энергопоребление кристалла

Энергопотребление микроконтроллера

При реализации устройства с батарейным питанием оптимизация энергопотребления кристалла определяется тремя параметрами (рис. 1):

- расход энергии в активном режиме работы;
- расход энергии в режиме ожидания;
- длительность активного режима работы.

Если в устройстве сбора данных обработка входных данных проводится с использованием математических алгоритмов, то наиболее затратной по энергопотреблению частью цикла работы устройства становится активный режим. На рис. 1 энергопотреблению во время обработки данных соответствует площадь прямоугольника, ограниченного отрезком времени работы в активном режиме и значением тока потребления. Уменьшение этой площади возможно либо за счет сокращения времени работы в активном режиме, либо путем снижения потребляемого при этом тока. Уменьшение величины потребляемого тока без потери производительности — задача, над которой непрерывно трудятся производители микроконтроллеров. Разработчику конечного устройства предоставляется возможность наращивать скорость вычислений. Чтобы это не влекло пропорционального увеличения энергопотребления, используются оптимизированные математические алгоритмы, многие из которых могут эффективно исполняться только с аппаратной поддержкой со стороны процессорного ядра. В таком ключе и рассмотрим особенности ядра Cortex-M4F, сравнивая его с популярным ядром Cortex-M3.

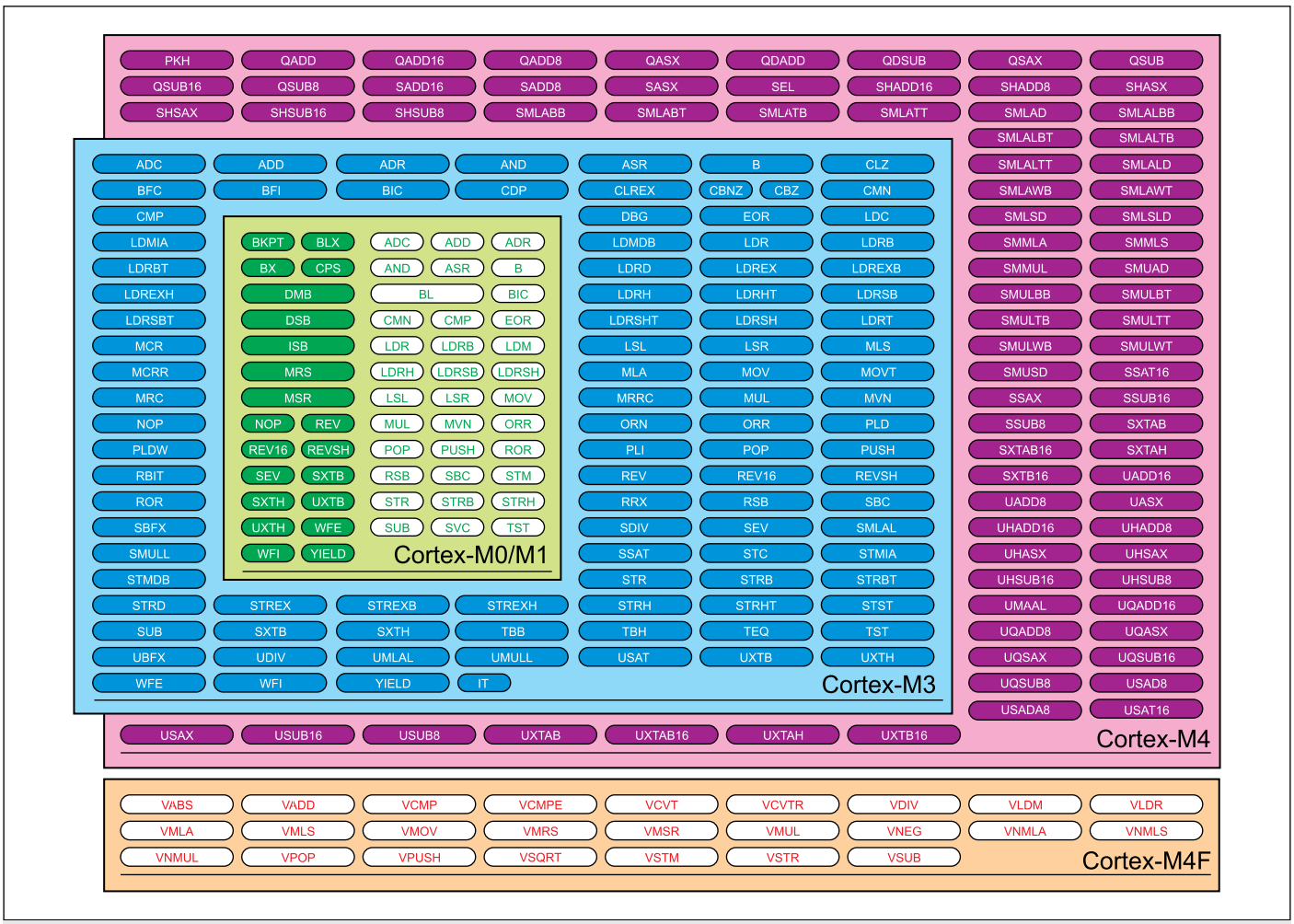

Рис. 2. Система команд процессорных ядер Cortex-M

Ядро Cortex-M4F

Система команд Cortex-M4 состоит из набора команд ядра Cortex-M3, расширенного DSP-инструкциями, а система команд Cortex-M4F дополнена командами блока FPU (рис 2). Дополнительные команды предназначены для увеличения производительности вычислений и могут быть разделены на четыре основные группы.

Умножение с накоплением (Single-cycle Multiply Accumulate, MAC)

Умножение с накоплением описывается формулой:

$$S = S+A\times B.$$

Соответствующие команды описывают умножение двух регистров с суммированием результата в аккумулятор и смежные операции: умножение с вычитанием результата из аккумулятора, умножение без использования аккумулятора и т. д. Операции предусмотрены для 16- и 32-разрядных переменных и играют важную роль во многих типовых алгоритмах цифровой обработки сигналов. Например, КИХ-фильтр описывается уравнением:

$$y_n = \sum_{i=0}^{N} c_i x_{n-i},$$

то есть представляет собой лишь последовательность операций умножения с накоплением, а значит, скорость работы фильтра напрямую определяется скоростью выполнения умножения с накоплением. Все MAC-инструкции в микроконтроллерах с ядром Cortex-M4(F) выполняются за один машинный цикл.

Параллельная обработка данных (Single Instruction Multiple Data, SIMD)

Команды группы SIMD позволяют оптимизировать обработку данных за счет параллелизма вычислений. Независимые переменные попарно помещаются в одни и те же регистры большей размерности для одновременного выполнения арифметических операций. Приведем пример: SIMD-команда SADD16 R0, R1, R2 подразумевает, что в регистрах R1 и R2 записано по две независимые переменные (разряды [15:0] содержат первую пару переменных, разряды [31:16] — вторую пару). Переменные из младших разрядов будут сложены с сохранением результата в младших разрядах регистра R0, одновременно результат сложения переменных из старших разрядов будет сохранен в старших разрядах регистра R0 (рис. 3).

Операция сложения SIMD, как и стандартное сложение двух 32-разрядных переменных, исполняется за один машинный цикл, но за этот цикл выполняется два сложения вместо одного. Поскольку в Cortex-M4(F) регистры общего назначения имеют разрядность 32 бит, в каждый из них можно поме-

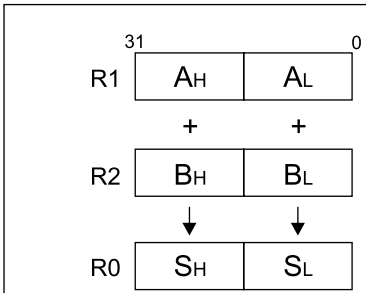

Рис. 3. Сложение с использованием SIMD-команды

стить по две 16-разрядных переменных или до четырех 8-разрядных.

Ядро Cortex-M4(F) поддерживает множество SIMD-команд, все они исполняются за один машинный цикл. Несколько команд SIMD включают умножение с накоплением, что еще больше увеличивает производительность вычислений. Например, выражение:

$$S_{64} = S_{64}+A_{16}\times B_{16}+C_{16}\times D_{16}$$

может быть выполнено за один машинный цикл.

Команды операций с насыщением (Saturating instructions)

Если результат какой-либо операции имеет большую разрядность, чем регистр, его хранящий, то говорят о переполнении регистра. При использовании стандартных команд регистр «перезагружается» с нуля после достижения переменной максимума, допустимого разрядностью, и, чтобы избежать значительной ошибки, требуется предусматривать проверки флага переполнения в коде программы.

При использовании команд с насыщением в аналогичной ситуации результат фиксируется на максимуме, допустимом разрядностью (рис. 4). Применение операций с насыщением существенно ускоряет и упрощает вычисления за счет отсутствия проверок на переполнение.

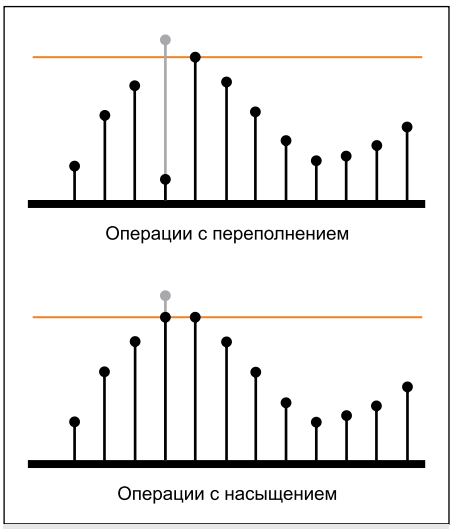

Рис. 4. Операции с переполнением и с насыщением

Вычисления с плавающей точкой (Floating Point Unit, FPU)

Команды вычислений с плавающей точкой — это команды блока FPU, позволяющие выполнять операции над вещественными числами с максимальной производительностью. Для представления вещественных чисел используется два формата — с фиксированной и плавающей точкой. В первом случае количество разрядов для записи целой и дробной частей зафиксировано и вычисления сводятся к операциям над целыми числами. Число с плавающей точкой представляется как совокупность знакового бита, нескольких разрядов для порядка и мантиссы:

$$\pm m\times 2^{-e},$$

где m — мантисса; e — порядок.

Использование формата с плавающей точкой предпочтительно при обработке сигналов за счет гораздо более широкого диапазона значений, его применение также избавляет разработчика от необходимости следить за разрядностью. Формат чисел с плавающей точкой одинарной точности описываются стандартом IEEE 754 (рис 5), это представление используется в микроконтроллерах с ядром Cortex-M4F. Диапазон допустимых значений составляет (10^{-38}–10^{38}) при приблизительном пересчете в десятичные числа.

В таблице указана длительность исполнения нескольких операций с вещественными числами, выполненных с использованием блока FPU.

Выигрыш по производительности, получаемый при использовании описанных выше расширений системы команд, может быть продемонстрирован через сравнение длительности выполнения типовых алгоритмов на микроконтроллерах с ядрами Cortex-M4F и Cortex-M3. Рассмотрим быстрое преобразование Фурье, КИХ-фильтр и матричные вычисления для различных типов данных. На графиках (рис. 6) f32 — формат числа с плавающей точкой одинарной точности, q15 и q31 — формат числа с фиксированной точкой с 15- и 31-разрядной мантиссой соответственно.

Быстрое преобразование Фурье (рис. 6а) для данных в формате с фиксированной точ-

Таблица. Длительность исполнения команд FPU в машинных циклах

Команда	Описание	Длительность в машинных циклах
VADD.F32	сложение	1
VSUB.F32	вычитание	1
VMUL.F32	умножение	1
VFMA.F32	умножение с последующим сложением	3
VDIV.F32	деление	14
VSQRT.F32	извлечение квадратного корня	14
VMOV	загрузка вещественной константы в регистр	1
VCMP.F32	сравнение двух вещественных чисел	1
VCVT.F32	преобразование между типами данных (целые, с фиксированной и плавающей точкой)	1

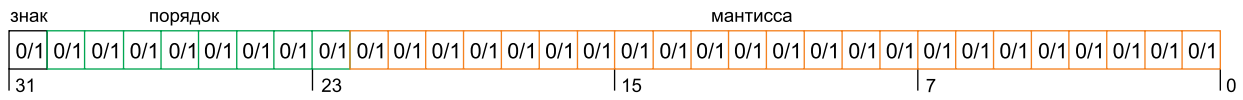

Рис. 5. Формат числа с плавающей точкой одинарной точности

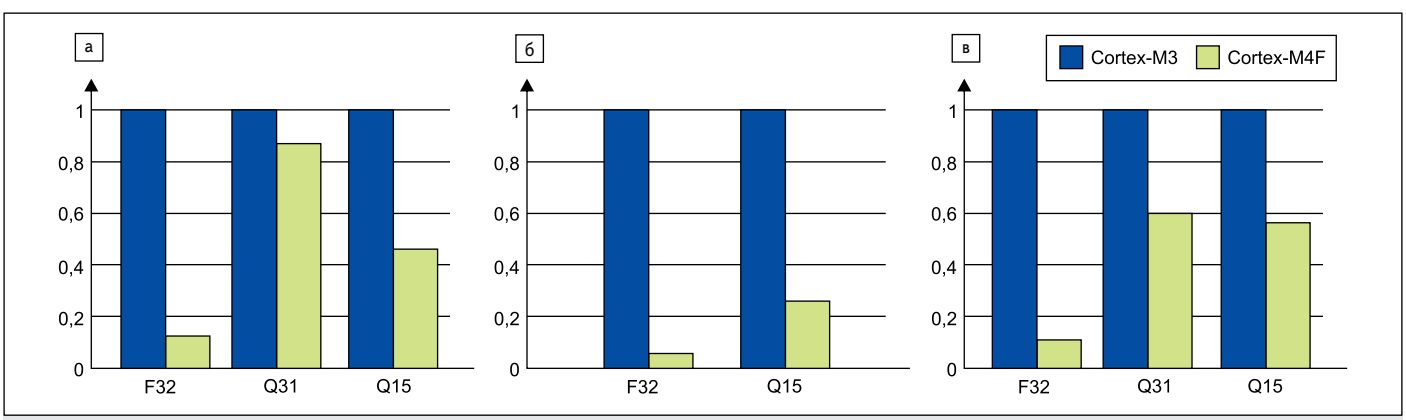

Рис. 6. Выполнение алгоритма: а) быстрого преобразования Фурье; б) КИХ-фильтра; в) матричного вычисления

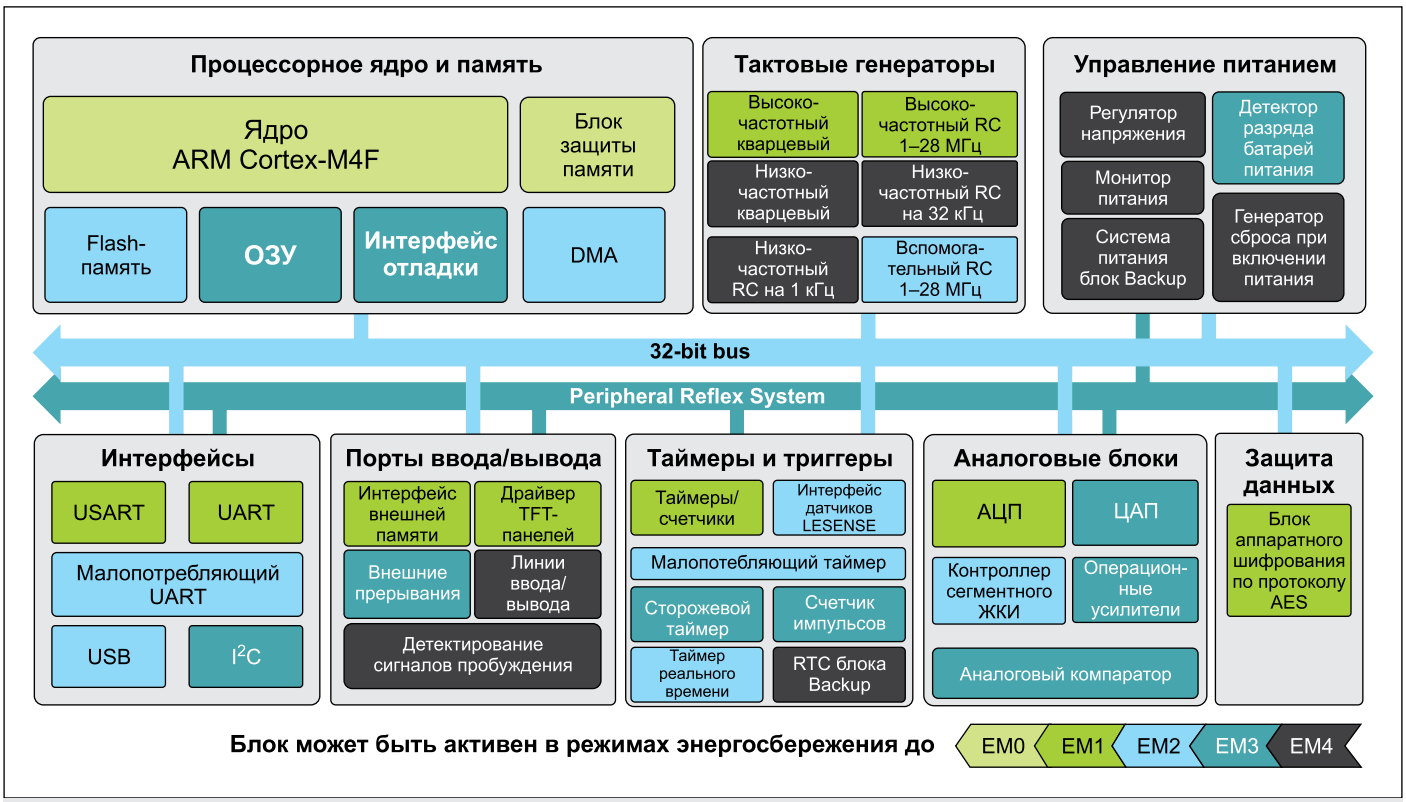

Рис. 7. Состав микроконтроллеров серии EFM32 Wonder Gecko

кой выполняется быстрее на Cortex-M4F благодаря использованию оптимизированных инструкций SIMD. Самое заметное повышение производительности видно, конечно, для формата f32, то есть при аппаратной реализации вычислений с плавающей точкой на ядре Cortex-M4F вместо программной на Cortex-M3.

Сокращение длительности обработки данных влечет снижение энергопотребления. Так при выполнении алгоритма БПФ для 512 выборок дважды за секунду с заменой ядра Cortex-M3 на Cortex-M4F среднее значение энергопотребления понижается с 550 до 170 мкА, то есть более чем в три раза.

Скорость КИХ-фильтрации (рис. 6б) на ядре Cortex-M4F возрастает для всех рассмотренных типов данных благодаря MAC-инструкциям. При использовании формата с плавающей точкой алгоритм выполняется в 16,6 раза быстрее по сравнению с реализацией на ядре Cortex-M3.

При обработке данных достаточно часто применяются матричные вычисления, особенно при сопоставлении показателей разных датчиков. Например, комбинация показаний GPS, акселерометра и гироскопа используется для точного определения местоположения. Распространенный способ сопоставления показаний — фильтр Калмана, реализация которого основана на умножении матриц. На рис. 6в показано сравнение скорости выполнения десяти умножений матриц размерности 10×10 с использованием различных форматов данных. Во всех случаях заметно увеличение производительности при использовании расширенной системы команд Cortex-M4F.

Все приведенные тесты выполнялись компанией Silicon Labs для программ, скомпилированных в среде IAR EWARM 6.50 с включенной функцией оптимизации кода по скорости. Быстрое преобразование Фурье вычислялось для выборок объемом 1024, КИХ-фильтр — для 320 выборок и 30 коэффициентов. Производительность Cortex-M4F оценивалась на примере микроконтроллеров серии EFM32 Wonder Gecko — одного из наиболее удачных решений среди универсальных микроконтроллеров на базе Cortex-M4F для приложений с батарейным питанием. Расскажем об этой серии более подробно.

Микроконтроллеры EFM32 Wonder Gecko

Серия EFM32 Wonder Gecko является развитием популярной линейки малопотребляющих контроллеров EFM32 Gecko, в которую также входят кристаллы на базе ядер Cortex-M0 и Cortex-M3. Подобно им, Wonder Gecko обладают хорошими показателями по потребляемой мощности процессорного ядра и отдельных периферийных модулей. Энергопотребление процессорного ядра у этих кристаллов составляет 225 мкА/МГц, максимальная тактовая частота — 48 МГц. Для реализации систем сбора данных предусмотрены механизмы для дополнительного снижения энергопотребления.

EFM32 Wonder Gecko поддерживают четыре режима сна, на рис. 7 для каждого из них цветом отмечены доступные периферийные устройства. Функциональность и энергопотребление кристалла в каждом режиме могут быть дополнительно настроены, для этого служит гибкая система тактирования. В кристаллах EFM32 Wonder Gecko разработчику доступно четыре встроенных RC-генератора, два кварцевых генератора и несколько независимых делителей частоты. Большое количество источников и типов тактовых сигналов, а также возможность автоматического и программного отключения тактирования отдельных периферийных блоков позволяют настроить кристалл так, чтобы энергия потреблялась только работающими блоками. Разработчик может не только отключить питание периферийных блоков, но и настроить неиспользуемые линии ввода/вывода в специальный режим с отключением входного триггера Шмита и выходного драйвера для каждой «ножки». При необходимости можно отключить не использующиеся блоки ОЗУ.

Для сбора данных на микроконтроллерах EFM32 Wonder Gecko может применяться 12-разрядный АЦП с частотой дискретизации до 1 млн отсчетов в секунду при потреблении всего 350 мкА. Важно отметить, что преобразование может быть инициировано без участия процессорного ядра средствами Peripheral Reflex System (PRS). PRS — инструмент для организации сценариев прямого взаимодействия периферийных устройств (например, старт АЦП по срабатыванию тай-

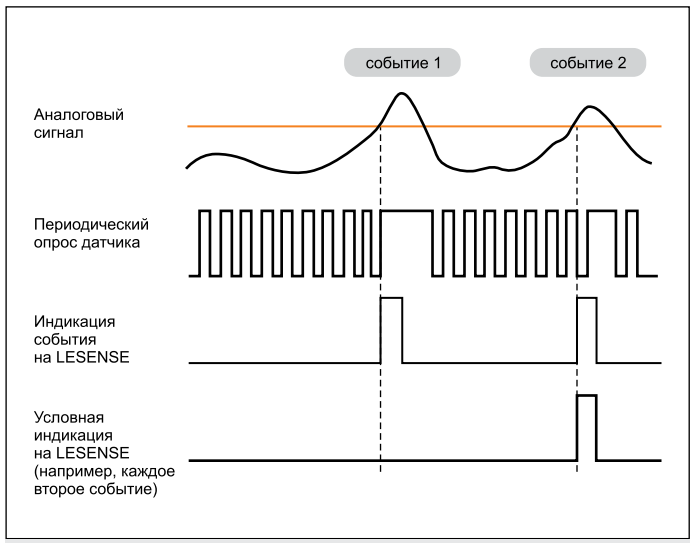

Рис. 8. Принцип работы блока LESENSE

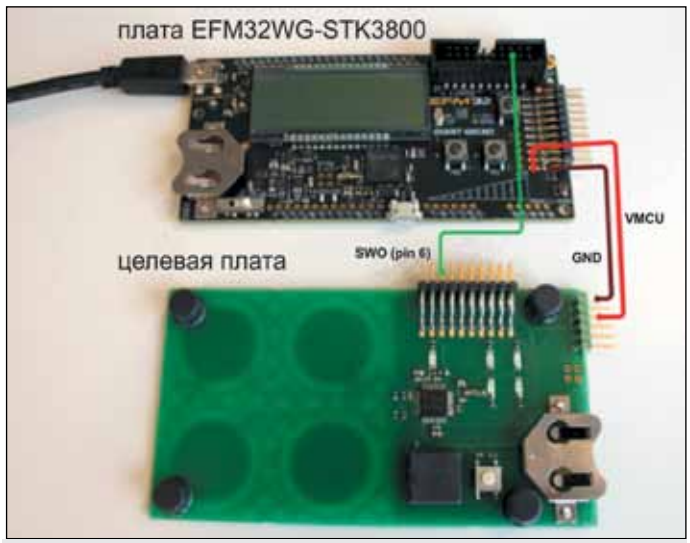

Рис. 9. Подключение EFM32WG-STK3800 для анализа потребления сторонней платы

мера). Ядро может находиться в режиме сна пока идет сбор данных, переходя в активный режим только когда данные уже доступны в памяти и готовы к обработке.

На автономную работу и малую потребляемую мощность ориентировано также множество других цифровых и аналоговых периферийных модулей, среди которых ЦАП, таймеры/счетчики, аналоговые компараторы, счетчики импульсов, операционные усилители, последовательные и параллельные интерфейсы. Отдельно можно отметить блок Low Energy Sensor Interface (LESENSE). LESENSE — это интерфейс для работы с резистивными, индуктивными и емкостными датчиками, одновременно может использоваться до 16 датчиков. Он позволяет организовать циклическое включение датчиков, обеспечивая их активность только во время измерений. Блок может использоваться в режимах сна вплоть до EM2, а его собственное энергопотребление не превышает 2 мкА. Конечный автомат блока можно настроить так, чтобы ядро «просыпалось» только при детектировании заданной последовательности событий (рис. 8), например при скольжении пальца по сенсорной панели.

Кроме уменьшения роли ядра при сборе данных средствами DMA, PRS и автономных периферийных блоков, производитель рекомендует использовать другие приемы для оптимизации потребления в активном режиме:

- выполнение инструкций из ОЗУ вместо flash-памяти (режимы EM3 и выше);
- применение режимов энергосбережения с выходом по прерыванию вместо программной реализации ожидания события (циклов типа while);
- настройка потребления аналоговых блоков через токи смещения в счет потери точности и др.

Поддержка разработок производителем

Для всех микроконтроллеров EFM32 Gecko программная поддержка разработок осуществляется через платформу Simplicity Studio. Платформа распространяется бесплатно и содержит IDE, набор инструментов для конфигурирования кристалла и контроля его энергопотребления, документацию, а также набор программных библиотек и примеров готовых устройств. Таким образом, все описанные ранее способы снижения энергопотребления контроллера могут быть опробованы и внедрены. Приведем несколько примеров: графический конфигуратор периферии и выводов кристалла позволит отключить питание неиспользуемых блоков, утилита Energy Aware Battery — осуществить оценку длительности работы микроконтроллера при питании от типовых элементов с учетом конфигурации контроллера. Для эффективной реализации алгоритмов цифровой обработки данных можно задействовать встроенную в Simplicity Studio библиотеку CMSIS от компании ARM Ltd. Библиотека содержит готовые и оптимизированные для EFM32 Wonder Gecko функции DSP: базовые векторные и матричные операции, операции с комплексными числами, тригонометрические и вероятностные функции, различные фильтры, преобразователи и т. д. — всего порядка шестидесяти функций.

На отладочной плате EFM32WG-STK3800 (рис. 9) можно сразу опробовать готовые примеры из Simplicity Studio, среди них обработка аудиофайлов, КИХ-фильтрация, измерение частоты работы источника света и т. д. Сама отладочная плата, помимо прочего, содержит интегрированный измеритель тока, который в паре с утилитой Energy Aware Profiler из Simplicity Studio позволяет отслеживать потребление кристалла по ходу исполнения программы. Такой программно-аппаратный комплекс можно использовать в том числе для изменения энергопотребления сторонних плат, обеспечивая оптимизацию по потреблению во всем цикле проектирования устройства.

Выводы

Эффективным решением для разработки устройств с батарейным питанием для сбора и обработки данных «на месте», является использование микроконтроллеров с ядром Cortex-M4F. Компания Silicon Labs предлагает микроконтроллеры EFM32 Wonder Gecko и полный комплекс их программной и аппаратной поддержки для создания устройств, обеспечивающих компромисс производительности и энергопотребления.

Литература

1. Energy Optimization AN0027 — Application Note.
2. Introducing the EFM32 Wonder Gecko: An Energy Friendly MCU with Signal Processing Capabilities.
3. Digital Signal Processing with the EFM32 AN0051 — Application Note.
4. EFM32WG Reference Manual.