

Методы снижения энергопотребления в строго самосинхронных микропроцессорных схемах

Алексей БУМАГИН,
к. т. н.
Екатерина ГЛАДКОВА
Алексей ГОНДАРЬ
Михаил КУЛЯС
Александр РУТКЕВИЧ
Владимир СТЕШЕНКО,
к. т. н.
Мехди ТАЙЛЕБ
Григорий ШИШКИН

В статье рассмотрена методика снижения потребляемой мощности строго самосинхронных микропроцессоров. Показаны отличия в подходах к энергосбережению по сравнению с обычными синхронными микропроцессорными системами. Рассмотрены алгоритмы динамического управления напряжением питания ядра микропроцессора при решении различных типов задач. Результатом использования рассмотренных методик является рациональное распределение вычислительной мощности и уменьшение энергопотребления.

Благодаря интенсивному развитию субмикронных технологий, позволяющих проектировать процессоры с плотностью до 10^9 транзисторов в кристалле размером 300 мм^2 , появилась возможность повысить функциональность электронных устройств. Однако вместе с тем увеличение плотности транзисторов в кристаллах интегральных схем (ИС), а следовательно, и удельного тепловыделения, предопределили необходимость применения кардинальных мер по снижению энергопотребления. Существует большое количество способов снижения энергопотребления синхронных процессорных систем, связанных с управлением напряжением питания и частотой работы, но их возможности ограничены. Данная проблема особенно актуальна для бортовых и портативных систем, для которых энергопотребление становится одним из факторов, сдерживающих дальнейшее развитие таких систем.

Одним из кардинальных путей снижения энергопотребления является переход к самосинхронной методике проектирования. Из-за особенностей данного подхода строго самосинхронные микропроцессоры (ССМ) обладают меньшим энергопотреблением по сравнению с широко применяемыми синхронными вычислительными устройствами при соизмеримой производительности. Это обусловлено, в первую очередь, отсутствием необходимости использования глобальных цепей тактовой синхронизации, являющихся дополнительными потребителями [1]. Кроме того, строго самосинхронные схемы при отсутствии изменения данных на входе находятся в статическом состоянии, что позволяет автоматически снижать потребляемую мощность по сравнению с синхронными аналогами.

Кроме снижения энергопотребления за счет использования особенностей самосинхронного подхода, необходимо использовать дополнительные методы энергосбережения ССМ, которые будут рассмотрены в этой статье.

Особенности строго самосинхронной схемотехники

Особенностью самосинхронных схем является отсутствие глобальных тактовых сигналов, используемых в синхронной схемотехнике. При использовании данного подхода каждый комбинационный блок после завершения переходных процессов должен вырабатывать сигнал готовности приема следующих данных, по которому осуществляется синхронизация предыдущего блока, тем самым обеспечивая логическое упорядочивание событий в схеме. Данные между логическими блоками передаются с применением некоторого протокола передачи, который обеспечивает синхронизацию информации. В задачу протокола входит сопровождение передаваемых данных парой «запрос/подтверждение», которая обеспечивает синхронизацию на локальном участке схемы.

Один из вариантов построения строго самосинхронных схем основан на использовании парафазного представления сигналов.

Для передачи одного бита информации D используется два бита — D0 и D1 (табл. 1). Такое представление позволяет, кроме обычных логических состояний — «0» и «1», определять спейсер (состояние перехода, NULL).

Таблица 1. Избыточное представление логического сигнала

Сигнал	Бит 0	Бит 1
Спейсер N	0	0
Логический «0»	1	0
Логическая «1»	0	1
Ошибка E	1	1

На рис. 1 показаны временные диаграммы представления сигнала в парафазном коде. При передаче биты данных разделяются спейсером, позволяющим отличать переходные процессы от стационарных состояний сигналов. Кроме того, не требуется отдельная передача сигнала «запрос», так как он непосредственно встроен в данные.

Состояние ошибки недопустимо для исправной схемы. В реальной схеме оно не долж-

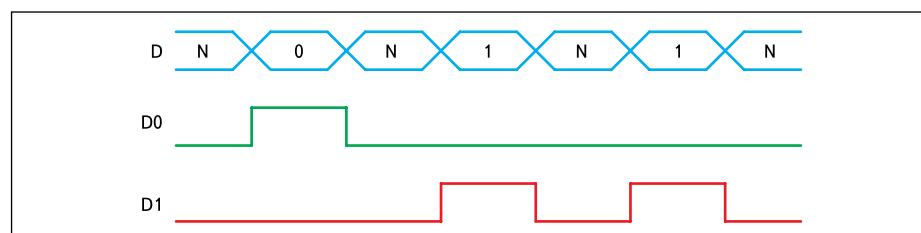


Рис. 1. Временные диаграммы парафазного представления сигнала

но встречаться никогда, поэтому данное состояние может, например, использоваться для контроля исправности работы схемы.

Парафазное кодирование позволяет определять момент окончания переходного процесса, непосредственно анализируя данные на выходе блока, без использования дополнительных блоков индикации. Сейчас это решение наиболее часто используется, оно позволяет строить строго самосинхронные схемы, используя стандартные логические элементы. К ограничениям на построение схем относятся требования индицируемости и монотонности комбинационных частей схемы. Эти два условия сводятся к возможности по выходам комбинационной схемы определить момент окончания всех переходных процессов внутри схемы и требование, чтобы внутри схемы не было «логических гонок» вне зависимости от моментов поступления входных сигналов.

Основой строго самосинхронных схем является G-триггер (гистерезисный триггер, С-элемент Мюллера). Выход G-триггера представляет собой копию входных сигналов, если они имеют одинаковое значение, и сохраняет свое предыдущее состояние, если они отличаются друг от друга. Триггер может иметь произвольное количество входов. На рис. 2 показано условное графическое изображение двухвходового элемента, а в таблице 2 приведена его таблица истинности.

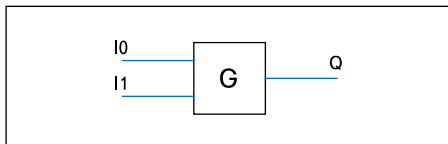


Рис. 2. Условное графическое изображение G-триггера

Таблица 2. Таблица истинности G-триггера

I0	I1	Q (n+1)
0	0	0
0	1	Q (n)
1	0	Q (n)
1	1	1

На рис. 3 показан пример простейшей строго самосинхронной схемы. Ее основой являются асинхронные защелки и индикаторы окончания переходного процесса, которые вместе образуют асинхронные регистры. Схема работает следующим образом. В начальном состоянии все линии данных находятся в состоянии спейсера N, а выходы блоков индикации — в состоянии логической «1», что говорит о том, что асинхронные защелки готовы принимать данные. При поступлении на вход схемы D данных они проходят через G-триггеры первой асинхронной защелки и далее поступают на вход комбинационной схемы и первого индикатора. Наличие данных на входе индикатора вызовет переключение его выхода в состоя-

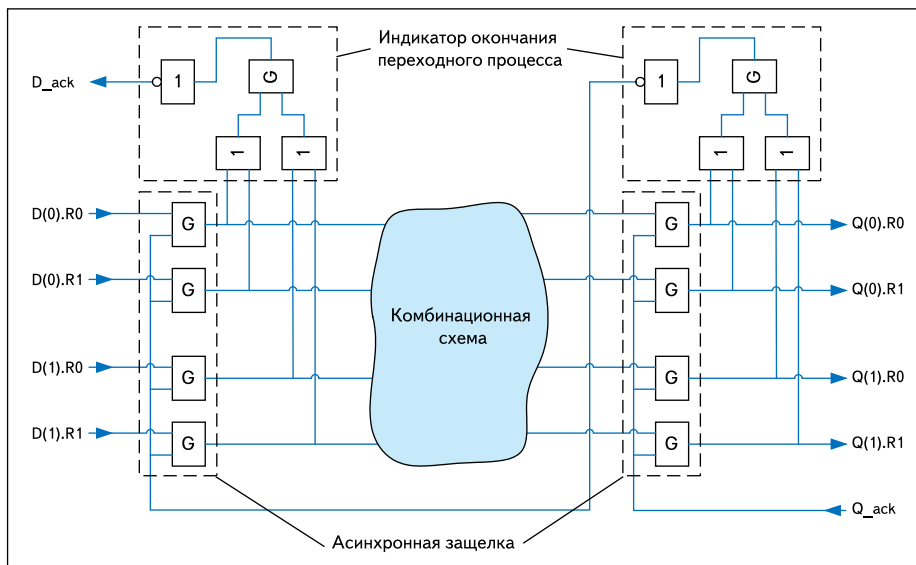


Рис. 3. Пример строго самосинхронной схемы

ние логического «0», что показывает готовность первой асинхронной защелки принять следующий за данными спейсер. При его поступлении на вход схемы он не сможет пройти через первую асинхронную защелку, так как выход второго индикатора находится в состоянии логической «1», и первая асинхронная защелка сохраняет на выходе свое предыдущее состояние.

Через некоторое время, равное задержке прохождения сигналов через комбинационную схему, данные появляются на ее выходе. Для того чтобы данные прошли через вторую асинхронную защелку, необходимо, чтобы сигнал Q_ack был в «1», показывая, что следующая часть схемы готова принять данные. Если это условие выполняется, данные пройдут вторую асинхронную защелку и вызовут переключение второго индикатора. Это в свою очередь разрешит первой защелке пропустить состояние спейсера через себя, что приведет к переключению первого индикатора. Когда состояние спейсера пройдет через комбинационную схему, оно поступит на вход второй защелки, которая, пока сигнал Q_ack находится в «1», будет сохранять на своем выходе пришедшие ранее данные.

Разработчикам синхронных схем не нужно отслеживать порядок прохождения данных через блоки: это делается автоматически при использовании глобального тактового сигнала и проверке соблюдения ограничений проекта. Если в синхронных схемах разработчик имел дело с отдельными сигналами, изменения которых происходили синхронно во всей схеме, то при самосинхронном подходе нет синхронизации между отдельными событиями в схеме. Необходимо строго контролировать порядок прохождения данных через модули. Это требует дополнительных аппаратных затрат: введения дополнительных маркеров данных, сигналов синхронизации работы отдельных блоков и принци-

пиально отличающегося подхода к проектированию схемы. Задачи синхронизации обработки данных могут решаться как разработчиком при создании описания схемы, так и с помощью специальных программных средств на этапе синтеза схемы — из ее высокоуровневого описания на специальном языке описания аппаратуры, предназначенном для описания самосинхронных схем.

Методы снижения энергопотребления в процессорных системах

Для КМОП-технологий значительную долю рассеиваемой мощности имеет динамическая составляющая, которая определяется паразитным током заряда емкости нагрузки логического элемента и сквозным током от источника питания к общему проводу в момент изменения логического уровня.

Для синхронных микропроцессоров, если частота переключения сигнала равна f , то рассеиваемая паразитным током заряда входной емкости мощность определяется выражением [2]:

$$P = C_L U^2 f,$$

где C_L — емкость нагрузки, U — напряжение питания.

Для ССМ эта формула выглядит иначе:

$$P = C_L U^2 s,$$

где s — мгновенная скорость процессора.

Из этого выражения следует, что снижение потребляемой мощности ССМ возможно за счет уменьшения емкости нагрузки, напряжения питания и мгновенной скорости s .

Вторая составляющая рассеиваемой мощности КМОП-схем — сквозной ток. Величина сквозного тока прямо пропорциональна дли-

тельности фронтов входного сигнала и обратно пропорциональна емкости нагрузки (C_L). Мощность, рассеиваемая КМОП-схемой вследствие протекания сквозного тока, может составлять от 10 до 60% общей динамической мощности. Ее величина зависит от параметров технологии изготовления микросхемы, таких как вольт-амперные характеристики транзисторов, длительность переднего и заднего фронтов входного сигнала, входной емкости и напряжения питания.

Кроме динамического, необходимо учитывать статическое энергопотребление микропроцессорной системы. Его причина — токи утечки КМОП-транзисторов и обратные токи p - n -переходов.

Номинальные значения токов утечки «сток/исток» лежат в диапазоне 1–10 нА, причем величина тока экспоненциально зависит от величины порогового напряжения. Поэтому с увеличением разброса пороговых напряжений средние значения также возрастают экспоненциально. Единственным способом уменьшения статического энергопотребления является снижение напряжения питания вплоть до отключения неиспользуемых блоков.

Таким образом, факторы, влияющие на энергопотребление ССМ, можно разделить на две группы — технологические, зависящие только от технологии изготовления микросхемы, и внешние, которыми можно управлять во время работы. Первые определяются только выбором соответствующих библиотек элементов и в данной статье не рассматриваются.

Анализ динамического энергопотребления микросхем, построенных по КМОП-технологии, показывает, что удельное энергопотребление (Вт/МГц) пропорционально квадрату напряжения питания, то есть при уменьшении напряжения питания снижается не только производительность ССМ, но и удельное энергопотребление. Таким образом, с точки зрения энергосбережения наиболее выгодной является работа ССМ на предельно низкой частоте. В реальном устройстве это невозможно, так как выполнение вычислительных задач необходимо проводить за ограниченное время.

На основании вышесказанного можно выделить следующие способы управления энергопотреблением ССМ:

- отключение неиспользуемых функциональных блоков для уменьшения статического тока утечки;
- снижение напряжения питания во время работы до минимально возможного уровня, достаточного для решения требуемых вычислительных задач с целью уменьшения динамической составляющей энергопотребления.

Дополнительно стоит отметить, что в строго самосинхронных схемах переключение логических элементов, вызывающие динамическую составляющую энергопотребления, происходит только при обработке данных,

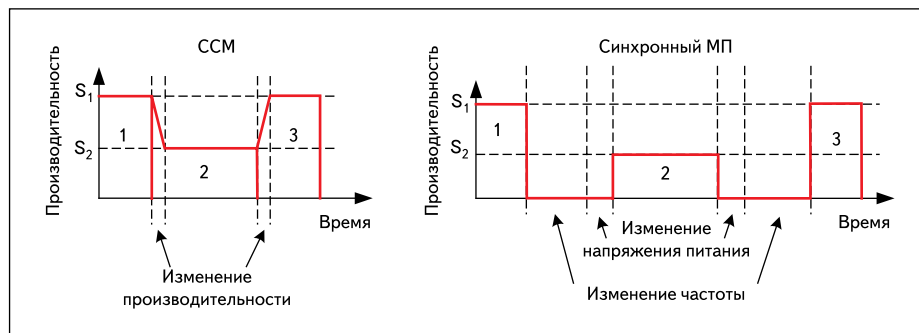


Рис. 4. Процесс изменения производительности ССМ и синхронного МП

при простом схеме находится в статическом состоянии. В синхронных микропроцессорах для уменьшения динамической составляющей энергопотребления во время простоя необходимо использовать специальные технические решения, осуществляющие понижение или полное отключение тактовой частоты у некоторых блоков.

Для перехода в режим пониженного энергопотребления и возврата из него синхронным процессорам требуется дополнительное время, в течение которого процессор простаивает. Это время может быть значительным, если необходимо перестраивать синтезатор частоты (PLL). В отличие от этого, в самосинхронных процессорах переход в такой режим происходит автоматически. Потребление энергии происходит только в процессе обработки данных. В остальное время ССМ находится в режиме ожидания, пока не произойдет запрос или прерывание. При этом не нужно дополнительного времени на включение неиспользуемых блоков. Эта особенность строго самосинхронной схемотехники позволяет постоянно использовать режим пониженного энергопотребления без применения дополнительных аппаратных средств, за исключением случая управления напряжением питания.

Особенности управления производительностью строго самосинхронных микропроцессоров

Рассмотрим процесс изменения производительности самосинхронного микропроцессора. Так как мы анализируем неидеальную систему, изменение производительности происходит за конечное время, которое определяется параметрами стабилизатора напряжения и нагрузочной емкостью цепей питания процессора.

Для синхронного МП управление производительностью можно осуществлять изменением напряжения питания и тактовой частоты. Это приводит к тому, что процесс изменения производительности должен состоять из двух этапов: при понижении скорости необходимо выполнить снижение тактовой частоты, затем уменьшить напряжение пита-

ния; при повышении выполнить эти этапы в обратном порядке: повысить напряжение питания, затем тактовую частоту [3].

Время изменения частоты может быть как постоянным (в процессорах StrongARM от Intel), так и зависеть от начальной и конечной частоты (в процессорах Crusoe производства Transmeta).

На рис. 4 показан пример процесса изменения скорости работы ССМ и синхронного микропроцессора. Изменение напряжения питания ССМ не требует остановки выполнения программы, в отличие от этого, для синхронного процессора необходимо остановить программу на время изменения частоты и напряжения для предотвращения возможных сбоев. Кроме того, необходимо учитывать дополнительное потребление энергии во время изменения напряжения и частоты, которого нет в ССМ.

Единственным способом управления производительностью ССМ, как было показано выше, является изменение напряжения питания U_{dd} . Напряжение питания может изменяться только в ограниченном диапазоне $U_{\min} < U_{dd} < U_{\max}$, где U_{\min} — минимальное рабочее напряжение, которое определяется переключательной способностью транзисторов во всем диапазоне рабочих температур, разбросов технологических параметров и старения, U_{\max} — максимально допустимое рабочее напряжение схемы, зависящее не только от физических параметров полупроводниковой структуры, но и от максимальной отводимой тепловой мощности от кристалла.

Ограничение тепловой нагрузки ССМ

При использовании ССМ возникают ситуации, когда тепловыделение микросхемы при максимальном напряжении питания превышает потенциальный отвод тепла. Это возможно для высокопроизводительных мобильных, автономных вычислительных устройств с жесткими требованиями по габаритам. Повышенное тепловыделение может привести к перегреву и в дальнейшем — к тепловому разрушению полупроводниковой структуры. Один из способов решения

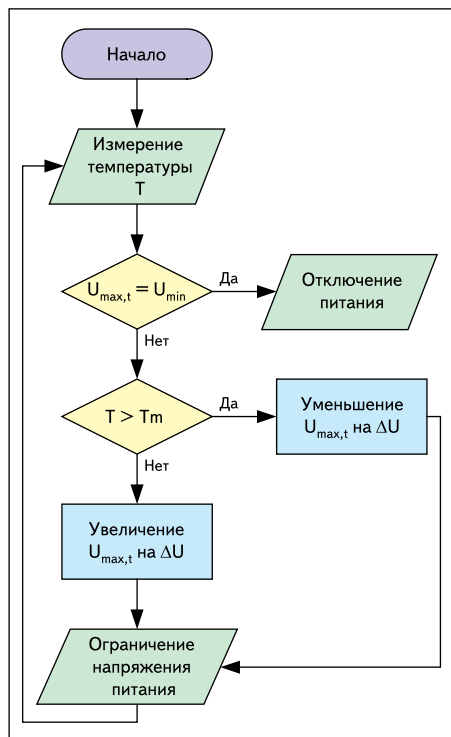


Рис. 5. Блок-схема алгоритма ограничения максимального напряжения питания

данной проблемы — уменьшение максимального напряжения питания U_{\max} до уровня, соответствующего возможностям отвода тепла. Такое решение приводит к занижению возможной максимальной производительности и может найти применение только в простейших вычислительных системах. Наиболее оптимальным является ограничение максимального напряжения питания U_{\max} в зависимости от температуры кристалла. Для этого измерение температуры необходимо проводить с помощью термодатчика, размещенного непосредственно в полупроводниковой структуре, в месте с наибольшей тепловой нагрузкой. На рис. 5 представлен алгоритм ограничения максимального напряжения питания в зависимости от температуры.

Для работы алгоритма вводится порог T_m , соответствующий температуре максимального перегрева кристалла, при которой необходимо снижать напряжение питания. При температуре выше порога T_m происходит снижение максимального напряжения при данных условиях: $U_{\max,t}$ на ΔU вплоть до значения U_{\min} . При достижении минимального значения необходимо осуществлять дополнительное охлаждение либо отключить питание процессора и, возможно, всей вычислительной системы — для предотвращения возможного перегрева и теплового разрушения, так как даже при минимальном значении рабочего напряжения питания невозможно обеспечить требуемое охлаждение микросхемы.

При использовании данного алгоритма необходимо обеспечить изменение напряже-

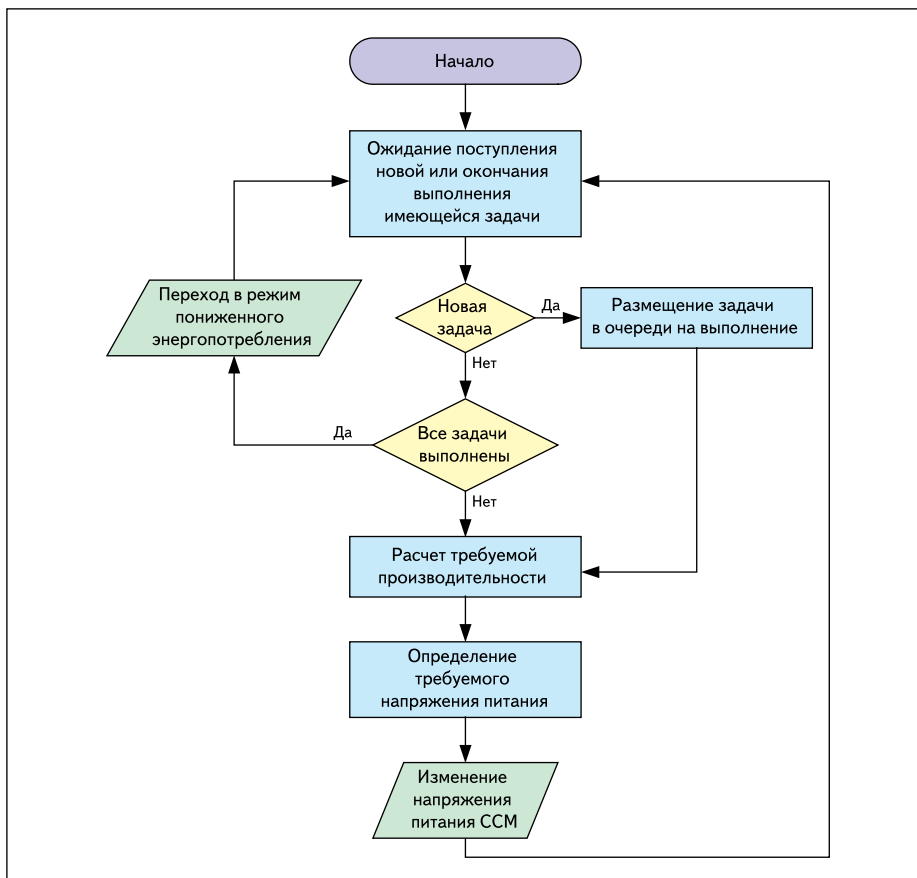


Рис. 6. Блок-схема алгоритма управления напряжением питания ССМ при выполнении потока случайных задач

ния питания процессора в диапазоне $U_{\min} < U_{dd} < U_{\max,t}$. Конкретные значения параметров T_m , ΔU , U_{\min} и U_{\max} зависят от используемой технологии изготовления микросхемы и определяются в процессе разработки.

Использование данного алгоритма позволяет получить максимальную пиковую производительность ССМ за счет теплоемкости элементов конструкции с последующим их охлаждением во время простоя.

Алгоритм динамического управления производительностью ССМ

Рассмотрим алгоритмы управления энергопотреблением ССМ, основанные на анализе и прогнозировании производительности.

Задачи, выполняемые ССМ, можно разделить на периодически повторяющиеся (периодические) и выполняемые при наступлении какого-либо случайного одиночного события (случайные). Периодические задачи встречаются значительно чаще случайных и характерны, например, при обработке поступающих данных от внешних источников сигнала (интерфейсного блока, АЦП и т. п.) или обновлении текущего состояния системы.

Каждая задача может быть описана тремя параметрами: количеством инструкций в задаче, требуемым временем завершения и периодом повторения. Для описания алгорит-

ма управления питанием введем следующие допущения:

- Задачи являются независимыми с известными параметрами.
- Время выполнения каждой периодической задачи меньше, чем период ее повторения; периоды повторения задач разные.
- Производительности ССМ при максимальном напряжении питания достаточно для выполнения всех задач.

Кроме того, не будем учитывать дополнительные вычислительные ресурсы, требуемые для контекстного переключения задач и управления напряжением питания, считая, что они пренебрежимо малы по сравнению с основными затратами на выполнение вычислительных задач. Первые можно учесть как дополнительную вычислительную задачу (или дополнительные инструкции), сопровождающую основную вычислительный процесс.

Управление напряжением питания ССМ при выполнении случайного потока задач

Рассмотрим ССМ, выполняющий только поток случайных задач. Для упрощения модели предположим, что в момент поступления задача готова и известны ее параметры; поступившие ранее задачи выполняются

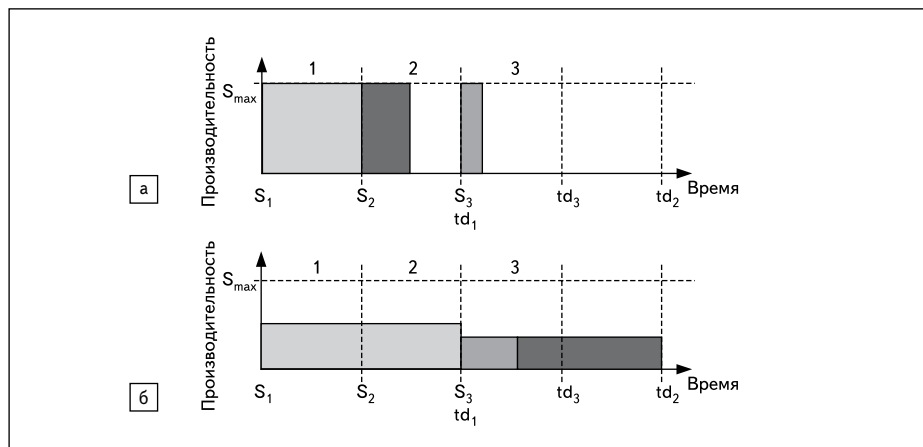


Рис. 7. Примеры функционирования процессоров: а) без изменения производительности; б) работающего в соответствии с рассмотренным алгоритмом

без превышения установленного для этого времени. Это требование может быть описано следующим выражением:

$$\left(\sum_{j \leq t} \frac{\bar{N}_j}{td_j - t} \right) \leq S_{\max}$$

где \bar{N}_j — среднее количество инструкций, необходимых для выполнения j -й задачи; td_j — время окончания задачи; t — текущее время; S_{\max} — максимальная производительность процессора (при максимально возможном напряжении питания).

На рис. 6 представлена блок-схема алгоритма управления напряжением питания ССМ при выполнении потока случайных задач.

Принцип работы алгоритма основан на оценке производительности в соответствии с параметрами задач, находящихся в очереди на выполнение.

Изменение производительности процессора происходит при поступлении новой задачи или при окончании выполнения имеющейся. При поступлении новой задачи последняя размещается в очереди на выполнение, причем таким образом, чтобы первая задача в очереди имела наименьшее время выполнения.

Далее производится оценка требуемой производительности ССМ по формуле:

$$S = \max_{i \in Q} \left\{ \frac{\sum_{j \neq i, P_j \leq P_i} \bar{N}_j}{td_i - t} \right\}$$

где Q — поток случайных задач в очереди на выполнение, P_i — приоритет i -й задачи в очереди на исполнение.

После выполнения оценки производится изменение напряжения питания ССМ в соответствии с расчетной производительностью. Напряжение питания выбирается из диапазона $U_{\min} < U_{dd} < U_{\max}$.

Производительность процессора изменяется каждый раз, когда добавляется или за-

вершается задача в очереди на выполнение. При окончании выполнения всех задач и отсутствии новых процессор переходит в состояние пониженного энергопотребления.

При реализации алгоритма необходимо учитывать: процесс изменения производительности является инерционным, что имеет значение при частой смене задач.

Несмотря на простоту данного алгоритма, он может быть не менее эффективен, чем более сложные алгоритмы управления производительностью ССМ [4]. Этот алгоритм избегает переоценки требуемой производительности, что и приводит к большей экономии энергии. На рис. 7 показаны примеры функционирования процессоров.

Предположим, необходимо выполнение трех задач. В случае, когда отсутствует управление напряжением питания (рис. 7а), процессор работает с производительностью S_{\max} и потребляет мощность P_{\max} . Для уменьшения энергопотребления процессор может быть остановлен (переведен в состояние пониженного энергопотребления) — по окончании выполнения задачи 2, вновь запущен при поступлении задачи 3 и по ее окончании вновь остановлен.

Так как ССМ не нужно дополнительного времени на остановку и запуск (рис. 7б), все три задачи будут непрерывно выполняться в течение требуемого времени. Для синхронных МП такой подход не эффективен, так как случайные задачи имеют произвольное время поступления, и время остановки предсказать трудно. Кроме того, остановка и запуск синхронных процессоров требует дополнительно расхода времени и энергии.

Управление напряжением питания ССМ при выполнении потока периодических задач

Рассмотрим алгоритм управления напряжением питания при выполнении потока периодических задач. Блок-схема алгоритма управления энергопотреблением для данного процесса представлена на рис. 8.

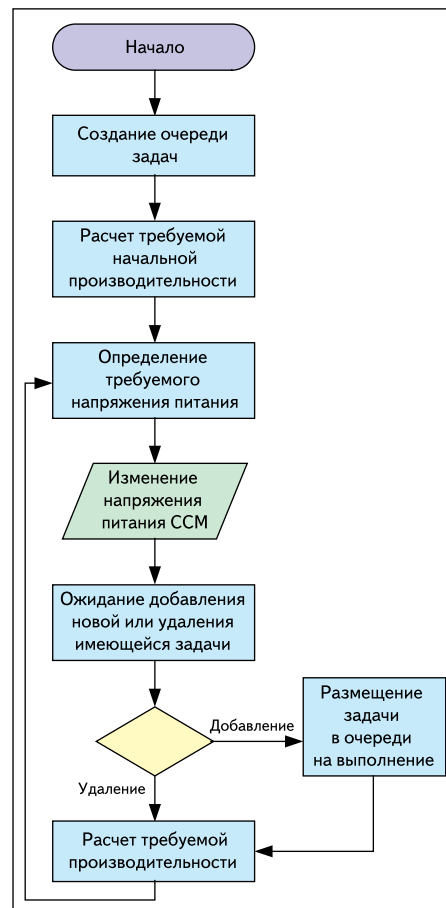


Рис. 8. Блок-схема алгоритма управления энергопотреблением при выполнении потока периодических задач

После начала работы необходимо создать очередь задач. Согласно требованиям приоритетности, в начале очереди должна располагаться задача, обладающая наименьшим временем исполнения. Необходимая начальная производительность процессора вычисляется следующим образом:

$$S = \sum_{j=1}^n \frac{N_j}{t_j}$$

где t_j — время выполнения j -й задачи; n — количество задач в очереди.

После чего определяется напряжение питания ССМ.

При поступлении новой задачи последняя размещается в очереди согласно приоритету, и производится пересчет требуемого быстродействия процессора:

$$S = \frac{N_{n+1}}{t_{n+1}} + \sum_{j=1}^n \frac{\bar{N}_j}{td_j - t}$$

Затем определяется и корректируется напряжение питания процессора в соответствии с расчетным быстродействием. Если задача удаляется из очереди, вновь производится перерасчет производительности.

В заключение отметим, что несмотря на имеющиеся у строго самосинхронных микропроцессоров преимущества, связанные с более низким энергопотреблением по сравнению с синхронными аналогами, проблема снижения энергопотребления для них также актуальна. Рассмотренные в статье алгоритмы позволяют уменьшить энергопотребление ССМ за счет оптимального перераспределения вычислительной мощности в зависимости от решаемой задачи. Наиболее оптимальным является использование комбинации рассмотренных алгоритмов. Они могут быть реализованы как программно, например, средствами операционной системы, так и аппаратно, с помощью специальных блоков оценки и прогнозирования.

Описанные алгоритмы управления напряжением питания позволяют повысить пиковую производительность вычислительной системы, не допуская перегрева микросхемы.

Результаты моделирования алгоритмов управления показывают возможность снижения потребляемой мощности более чем на 80%. ■

Литература

1. Рабаи Ж. М., Чандракасан А., Николич Б. Цифровые интегральные схемы. Методология проектирования / Пер. с англ. М.: ИД Вильямс, 2007.
2. Белоус А. И., Мурашко И. А., Сякерский В. С. Методы минимизации энергопотребления при проектировании КМОП БИС // Технология и конструирование в электронной аппаратуре. 2008. № 2.
3. Pering T., Burd T., Brodersen R. Voltage scheduling in the IpARM microprocessor system. Proc. Int. Symp. on Low-Power Electronic Design. Rapallo, Italy. July 2000.
4. Piquet C. Low-Power processors and systems on chip. Taylor & Francis Group, LLC, 2006.