УДК 621.382

МЕТОД СНИЖЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЯ НА ОСНОВЕ НЕПРЕРЫВНОГО УПРАВЛЕНИЯ НАПРЯЖЕНИЕМ ПИТАНИЯ В СБИС ДЛЯ ПРИЛОЖЕНИЙ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Ковалев А.В., Коноплев Б.Г.

ФГАОУ ВО «Южный федеральный университет», Ростов-на-Дону, e-mail: avkovalev@mail.sfedu.ru

Описывается реализация синхронно-асинхронной архитектуры динамического управления напряжением питания вычислительных блоков. Предлагается подход к плавному управлению питанием функциональных блоков СБИС, позволяющему минимизировать энергопотребление за счет динамического расчета времени вычислений и точной подстройки уровней напряжений при сохранении заданного соотношения успешно завершенных операций к их общему количеству. Нами предложен алгоритм, позволяющий статистически обеспечивать необходимую степень завершения операций Q при существенном снижении энергопотребления. Экономия энергии непрерывной моделью, по сравнению с оптимальным уровнем (при расчетном оптимальном напряжении), возможна только при меньшем (относительно расчетного) количестве операций без нарушения лимита времени. Для проверки энергоэффективности проведено моделирование популярных приложений ЦОС предлагаемого подхода. Непрерывная модель экономит до 74% энергии по сравнению с традиционной четырехуровневой дискретной моделью.

Ключевые слова: энергоэффективность, асинхронные схемы, граф, алгоритмы

THE METHOD TO REDUCE POWER OF DSP APPLICATIONS IN VLSI BASED ON CONTINUOUS CONTROL SUPPLY VOLTAGE

Kovalev A.V., Konoplev B.G.

Southern Federal University, Rostov-on-Don, e-mail: avkovalev@mail.sfedu.ru

The paper describes the implementation of synchronous-asynchronous architecture of the dynamic voltage control of computational units. An approach to a smooth power management of VLSI IP-cores to minimize the power consumption by dynamically calculating the computation time and fine tuning voltage levels while keeping the ratio of successfully completed problems to their total number. We have proposed the algorithm which statistically provide the necessary degree of completion of operations Q while substantially reducing power consumption. Energy saving continuous model, as compared to the optimal level (estimated when the optimum voltage) is possible only at a lower (relative to the reference) number of operations without violating the time limit. The simulations of popular DSP applications ware passed to test the efficiency of the proposed approach. Continuous model saves up to 74% energy compared with traditional 4-level discrete model.

Keywords: energy efficiency, asynchronous circuits, graph, algorithms

При проектировании портативных устройств электронной техники, работающих на автономном питании, одним из важнейших факторов, который должны учитывать разработчики, является эффективность энергопотребления. При этом точная продолжительность обработки потоков данных, поступающих с перерывами, труднопредсказуема во время работы приложения, однако она укладывается в определенный диапазон. Причем в ряде вычислительных задач, например кодирование/декодирование звука или изображений, допустимы потери качества, неразличимые для органов восприятия человека.

Сегодня большое число проектов строится на принципе гарантирования выполнения всех вычислительных задач, даже если временные показатели становятся неудовлетворительными (худшие случаи). Однако и в этих проектах, помимо того, что практически нет экономии мощности, случаются ошибки предсказания времени вычислений, поскольку в реальных приложениях худшие случаи наступают нечасто, от этого возникают непродуктивные простои или выполнение ненужных расчетов. За счет пропуска «несущественных» и удлинения времени выполнения «необходимых» вычислительных задач можно добиться сокращения энергопотребления процессора или специализированной СБИС в портативном электронном устройстве.

В синхронных схемах часто применяют дополнительные управляющие схемы и технологически предопределенный набор уровней питающих напряжений.

Нами предлагается метод сокращения энергопотребления СБИС, использующий возможности пропуска необязательных вычислений и плавного «растягивания» необходимых расчетов за счет применения динамического непрерывного управления напряжением питания в сочетании с асинхронными схемами, которые изначально предназначены для поддержания

непрерывной работоспособности в широком диапазоне напряжений и рабочих температур, а также разбросе технологических параметров.

Выбор в операционном режиме баланса между качеством вычислений (производительностью) и энергопотреблением – достаточно трудная и неоднозначно решаемая задача. В данной статье описана технология использования подобной информации для эффективного динамического непрерывного управления напряжением питания с целью сокращения энергопотребления кристалла.

Существующие аналогичные подходы

Динамическая потребляемая мощность в сравнении со статической занимает значительно большую долю в общем энергопотреблении кристалла. Поскольку уровень рассеиваемой энергии пропорционален количеству переключений элементов и квадрату питающего напряжения, то сокращение этих двух параметров потенциально ведет к существенной экономии мощности.

В работе [6] приводится анализ энергопотребления в высокопроизводительных процессорах, описываются различные подходы к снижению мощности и делается вывод о том, что управление энергопитанием дает наилучшие результаты.

В последнее время стали популярны методы и алгоритмы переключения блоков схем на различные заранее предопределенные уровни напряжений (как правило, 2–4 возможных уровня). В [7] впервые описали системы с многоуровневым напряжением на вентильном уровне. Эффективность подобных подходов зависит от оптимальности подбора уровней напряжений [5]. Исследователи в [2] предложили методологию проектирования систем на кристалле со встроенным аппаратным динамическим управлением уровнями напряжения.

В вышеописанных исследованиях недостаточно исследованы синхронно-асинхронные архитектурные решения и алгоритмы планирования с плавным изменением напряжений на системном уровне. Часто используется лишь несколько дискретных уровней питающего напряжения (как правило, не более 4-х), что приводит к погрешностям задания времени окончания периодов вычислений и, как следствие, к снижению либо числа «успешных» операций, либо к повышению общего объема затрачиваемой энергии.

В данной статье предлагается подход к плавному управлению питанием функциональных боков СБИС, позволяющему минимизировать энергопотребление за счет динамического расчета времени вычислений и точной подстройки уровней напряжений при сохранении заданного соотношения успешно завершенных периодов к их общему количеству.

Формулировка задачи

Пусть для выполнения приложения имеется граф вычислений

$$G = (V, E),$$

где V – набор вершин, представляющих собой отдельные вычислительные задачи (функции); E – направленные ребра, соединяющие вершины. С каждым ребром связан поток данных между функциями.

Задача вершины v_i может быть выполнена за конечное время, которое укладывается в набор известных длительностей

$$T_i = \left\{ t_{i1}\left(p_{i1}\right), t_{i2}\left(p_{i2}\right), \ldots, t_{ic_i}\left(p_{ic_i}\right) \right\},$$

где p_{ic_i} – вероятность выполнения v_i за время t_{ic_i}; c_i – количество элементов набора длительностей для вершины v_i. События (длительности) данного набора являются взаимоисключающими.

В данной работе рассматриваются следующие системы динамического управления напряжениям:

1. Идеальная модель: идеальная система управления напряжением может мгновенно и произвольно изменить рабочее напряжение (скорость процессора может варьироваться от 0 to ∞).

2. Реализуемые модели: реализуемая система управления напряжением может менять напряжение между $V_{\min}$ и $V_{\max}$ с максимальной скоростью K.

3. Мультивольтная модель: система мультинапряжений имеет ряд доступных дискретных уровней напряжения питания, имеющая возможность переключаться между ними мгновенно (без остановки вычислений).

Исследуем выполнение набора приложений (задач) в описанных системах динамического управления напряжением.

Приложение, выполняемое процессором, состоит из множества маршрутов (в графе G) $M = \{M_1, M_2, \ldots, M_N\}$, где N – число маршрутов вычислений. Для каждого маршрута заранее задается максимальное время его выполнения $LT(M_i)$, превышение которого признается как неуспешное завершение вычислений. Длительность выполнения всех вычислительных задач, т.е. прохождения заданного маршрута M_i на графе G, определяется суммой реальных временных отрезков r_i для каждой пройденной вершины:

$$t\left(M_i, h\right) = \sum_{i=1}^{h} r_i, \quad \text{при } h = N.$$

В связи с этим вводится индикатор качества результатов выполнения приложения в виде отношения «успешных» маршрутов к общему их числу:

$$Q = \frac{N - f}{N}, \quad (1)$$

где f – количество прерванных операций (маршрутов).

Показатель Q для различных типов приложения может различаться и задаваться априори.

Для заданного набора маршрутов $\{M_1, M_2, ..., M_N\}$ управляющий блок определяет системное напряжение, при котором будет выполняться текущее приложение. Каждый маршрут приложения M характеризуется следующими параметрами: время начала T_s, максимальное время выполнения d.

Общая потребляемая энергия, которую необходимо минимизировать:

$$E = \int_{T_s}^{T_s + d} P(t)dt, \quad (2)$$

где $P(t)$ – потребляемая мощность.

Таким образом, решение задачи снижения энергопотребления сводится к следующему: для данного набора маршрутов M на графе G с ограничениями LT необходимо найти такую вычислительную архитектуру и стратегию управления напряжением питания, чтобы энергопотребление приложения с заданным Q было минимальным.

Стратегия управления питанием состоит в определении оптимального уровня напряжения для каждой вершины графа G, а вычислительная архитектура должна на аппаратном уровне обеспечивать эффективную реализацию данной стратегии.

Оптимальное управление в описанном смысле может быть только в идеальной системе, в которой уровень напряжения питания изменяется мгновенно и непрерывно (бесступенчато) [5]. Однако в реальности возможно только приближение к такой идеальной системе, изменения напряжения будут в лучшем случае квазибесступенчатые с конечным временем установки желаемого уровня.

Поставленная задача снижения энергопотребления решается новым подходом к разработке схемной структуры на основе синхронно-асинхронной системы, реализующей алгоритм динамического управления непрерывным (квазибесступенчатым) питающим напряжением.

Архитектура управления питанием вычислительных модулей

Квазибесступенчатый характер изменения напряжения питания наиболее эффективно может быть реализован с помощью асинхронных схем, которые за счет присущей им возможности изменения напряжения в широком диапазоне без необходимости подстройки тактовой частоты потенциально могут обеспечить более низкое энергопотребление по сравнению с синхронными схемами при прочих равных условиях.

В рамках предложенного подхода предлагается построение системы по типу «глобально синхронная – локально асинхронная». Это позволит согласовать традиционное синхронное окружение с асинхронным ядром, которое будет питаться управляемым источником напряжения.

Структура для управления напряжением питания асинхронного ядра приведена на рисунке. Вычислительные каскады (блоки комбинационной логики с регистрирующими элементами) выдают в последовательном режиме сигналы, индицирующие о завершении локальных вычислений. Данные сигналы воспринимаются блоком управления напряжением, который по заданному алгоритму (описан ниже) генерирует команды на изменение напряжения питания. Тактовый генератор в данной системе необходим для привязки к реальному времени, используемому в алгоритме управления.

Задание временной модели вычислительных блоков

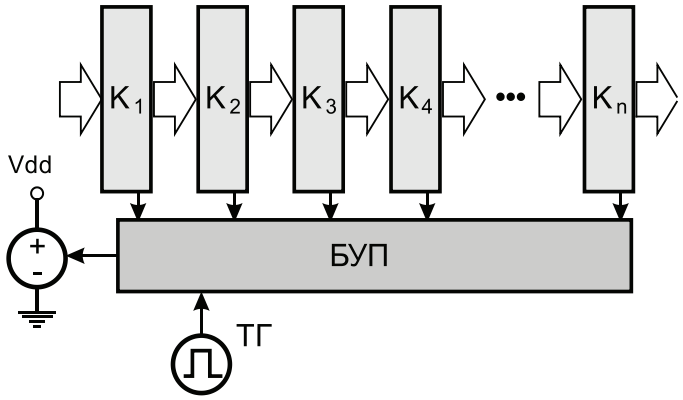
Экспериментальные данные, полученные путем моделирования асинхронной системы, показали плавную зависимость производительности от непрерывно изменяющегося напряжения питания. Характер данной зависимости сохраняется не только в статических режимах, но и при динамическом изменении напряжении с относительно невысокой частотой. Как можно видеть на графике, система работоспособна в определенных пределах: снизу ограничена пороговым напряжением V_{th}, сверху – после некоторого «насыщения» скорости, уровнем V_m, выше которого начинаются сбои в работе.

Для описания зависимости задержки от напряжения питания требуется подобрать аппроксимирующую функцию достаточной степени точности. Класс аппроксимирующей функции определен как степенной.

Задержка прохождения сигналов через элемент оценивается следующим выражением:

$$t = \frac{kV_{dd}}{(V_g - V_{th})^{\alpha}}, \quad (3)$$

где V_{dd} – напряжение питания; V_g – напряжение на затворе; V_{th} – пороговое напряжение; k и α – коэффициенты.



Структура, реализующая алгоритм управления (Обозначения: Ki – вычислительные каскады, ТГ – тактовый генератор, БУП – блок управления напряжением)

Для конкретной технологической реализации необходимо определить лишь значения постоянных коэффициентов, входящих в функцию. Причем для коэффициента α в (3) задается условие: $1 \le \alpha \le 2$.

Для обеспечения приемлемой точности аппроксимации применен метод наименьших квадратов, который позволяет добиться наименьшей невязки в произвольном числе точек, не связанном с числом неизвестных коэффициентов.

С целью решения обратной задачи определения уровня напряжения, соответствующего требуемому времени выполнения вычислений, получим дополнительную функцию при напряжении на затворе, равном питающему, и значении α в (3), равному 2:

$$V(t) = \frac{2tV_{th} + k + \sqrt{4ktV_{th} + k^2}}{2t}. \quad (4)$$

Алгоритм предсказания нагрузки и управления напряжением питания

В реальной системе частота и напряжение питания должны устанавливаться динамически для того, чтобы удовлетворять меняющиеся запросы вычислительной среды на требуемую производительность. Такого рода запросы, как правило, формирует операционная система, которая не всегда имеет информацию о планируемой вычислительной нагрузке конкретного приложения. Поэтому необходим определенный механизм предсказания, например на основе мониторинга текущей загруженности и оценки будущих запросов. Самым простым способом предсказания нагрузки является экстраполяция, однако в реальных системах более подходящим будет некий эвристических алгоритм.

В подобной нетривиальной задаче в случае ошибочного предсказания возрастают потери полезной мощности. В исследованиях [1, 4] показано, что избыточность по энергопотреблению в ряде случаев может быть существенной.

В рамках нашей работы предлагается модификация данного подхода, в котором реальная нагрузка (ненулевая) в предыдущем окне влечет за собой пропорциональное изменение напряжения для последующего окна.

Большинство операционных систем, однако, не могут точно определить объем недоделанной работы в предыдущем окне из-за недостатка оперативной информации. Это не является проблемой для приложений с постоянной нагрузкой, но в большинстве случаев сложные приложения имеют резко меняющиеся потребности в производительности. Например, моделирование декодера потока MPEG показало, что дополнительные 36% сокращения энергопотребления вполне достижимы при улучшении предсказания загрузки [3].

Алгоритм управления напряжением

Обозначения: N – число вершин; V_i – текущее напряжение питания; $V()$ – функция, возвращающая шаг изменения напряжения на основе (4) (аргументы – предыдущее значение V_{i-1} и разность между расчетным и текущим временем); CH – флаг, отражающий наличие корректировки.

1. Задание общего лимита времени LT (Limit of Time).

2. Распределение лимитов времени для каждой операции $LM = \{lm_1, lm_2, ..., lm_N\}$ при заданных вероятностях.

3. Расчет оптимального напряжения, соответствующего выполнению всех операций за время LT. Используется выражение (4).

4. Для каждой операции рассчитывается реально затраченное время выполнения, которое сравнивается с лимитами, полученными на шаге 2.

5. Если происходит расхождение между расчетным и реальными временами больше чем на определенную константную величину K, то производится корректировка напряжения питания в ту или иную сторону с учетом известных вероятностей.

6. Если расхождения более чем на K не наблюдается, устанавливается напряжение с предыдущей операции на один шаг назад или два, в зависимости от истории изменений.

7. Если происходит превышение текущего лимита, маршрут прерывается. Иначе переход к следующей итерации.

Экспериментальные результаты показателей эффективности снижения энергопотребления

Предложенная архитектура реализует непрерывную модель управления напряжением, поэтому следует ее сравнить с традиционным подходом (мультивольтная модель), а также с методом «идеальной» подстройки напряжения питания под динамически (случайно) меняющуюся нагрузку.

Данные методы управления напряжением моделировались на большом количестве итераций (от 100 тыс. до 1 млн) с различными реальными тестовыми графами задач, применяемых в популярных приложениях ЦОС.

Для каждой пары (LT, Q) мы имитировали выполнение каждого приложения на основе трех моделей и отслеживали соотношение завершения вычислений Q и количества потребленной энергии.

Проведено моделирование при двух вариантах изменения вычислительной нагрузки:

1) нагрузка меняется случайно, но равномерно по всем итерациям (Equal);

2) нагрузка меняется случайно и независимо по всем итерациям (Random).

Также рассматривались варианты модели с дискретными напряжениями (от 2 до 10 уровней).

Результаты сравнения предложенной непрерывной модели с двумя другими (gWo и gWst) приведены в таблице.

Проценты соотношения потребленной мощности моделей gWo и gWst к модели gWa.

Метод	Тип изменения нагрузки	Количество уровней напряжения для gWst				
		2	4	6	8	10
gWo	Equal	+18,56	+18,56	+18,56	+18,56	+18,56
	Random	+21,48	+21,63	+21,48	+21,48	+21,48
gWst	Equal	–92,03	–76,87	–55	–21,46	–1,628
	Random	–91,84	–74,56	–54,84	–36,69	–12,168

Из представленных данных можно сделать вывод о том, что предложенный алгоритм и непрерывная модель «проигрывает» по затраченной энергии идеальной подстройке напряжений приблизительно от 18 до 21% в зависимости от характера изменения вычислительной нагрузки. Однако он значительно превосходит по энергосбережению традиционный подход с дискретным набором напряжений. Для типичного случая с 4-мя уровнями выигрыш составляет примерно 74%. Конечно, с ростом числа уровней в модели gWst разрыв в результатах сохранения энергии сокращается, но паритет с gWa наступит только в бесконечном пределе.

Заключение

Предложен алгоритм, позволяющий статистически обеспечивать необходимую степень завершения операций Q при существенном снижении энергопотребления. Экономия энергии непрерывной моделью (gWa), по сравнению с оптимальным уровнем (при расчетном оптимальном напряжении), возможна только при меньшем (относительно расчетного) количестве операций без нарушения лимита времени.

Для проверки энергоэффективности проведено моделирование популярных приложений ЦОС предлагаемого подхода. Непрерывная модель экономит до 74% энергии по сравнению с традиционной 4-хуровневой дискретной мультивольтной моделью.

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации (гос. задание № 8.797.2014/к).

Список литературы

1. Govil K., Chan E. и Wasserman H. Comparing algorithms for dynamic speed-setting of a low-power CPU // Proceedings of the 1st annual international conference on Mobile computing and networking. – ACM. – 1995. – С. 13–25.

2. Hong I. [и др.] Power Minimization of Variable Voltage Core-Based Systems // 35th ACM/IEEE Design Automation Conference. – 1998 г. – С. 176–181.

3. Pering T., Burd T. и Brodersen R. The simulation and evaluation of dynamic voltage scaling algorithms // Proceedings of the 1998 international symposium on Low power electronics and design. – ACM. – 1998 г. – С. 76–81.

4. Pering T., Burd T.D. и Brodersen R.W. Voltage Scheduling in the IpARM Microprocessor System // ISLPED'00: International Symposium on Low Power Electronics and Design. – July 2000 г. – С. 96–101.

5. Qu G. What is the Limit of Energy Saving by Dynamic Voltage Scaling? // IEEE/ACM International Conference on Computer-Aided Design. – 2001 г. – С. 560–563.

6. Tiwali V. и al et Reducing Power in High-Perfomance Microprocessors // 35th ACM/IEEE Design Automation Conference. – 1998 г. – С. 732–737.

7. Usami K. и Horowitz M. Clustered Voltage Scaling Technique for Low-Power Design // ISLPED'95: International Symposium on Low Power Electronics and Design. – April 1995 г. – С. 3–8.

References

1. Govil K., Chan E. и Wasserman H. Comparing algorithms for dynamic speed-setting of a low-power CPU // Proceedings of the 1st annual international conference on Mobile computing and networking. ACM. 1995. pp. 13–25.

2. Hong I. [и др.] Power Minimization of Variable Voltage Core-Based Systems // 35th ACM/IEEE Design Automation Conference. 1998 г. pp. 176–181.

3. Pering T., Burd T. и Brodersen R. The simulation and evaluation of dynamic voltage scaling algorithms // Proceedings of the 1998 international symposium on Low power electronics and design. ACM. 1998 г. pp. 76–81.

4. Pering T., Burd T.D. и Brodersen R.W. Voltage Scheduling in the IpARM Microprocessor System // ISLPED'00: International Symposium on Low Power Electronics and Design. July 2000 г. pp. 96–101.

5. Qu G. What is the Limit of Energy Saving by Dynamic Voltage Scaling? // IEEE/ACM International Conference on Computer-Aided Design. 2001 г. pp. 560–563.

6. Tiwali V. и al et Reducing Power in High-Perfomance Microprocessors // 35th ACM/IEEE Design Automation Conference. 1998 г. pp. 732–737.

7. Usami K. и Horowitz M. Clustered Voltage Scaling Technique for Low-Power Design // ISLPED'95: International Symposium on Low Power Electronics and Design. April 1995 г. pp. 3–8.

Рецензенты:

Малюков С.П., д.т.н., профессор, директор Научно-образовательного центра «Лазерные технологии», Южный федеральный университет, г. Таганрог;

Лысенко И.Е., д.т.н., доцент, директор ООО «Нанотехнологии», г. Таганрог.

Работа поступила в редакцию 16.12.2014.