УДК 621.3.049.77
DOI: 10.22213/2410-9304-2021-4-88-97

Способы уменьшения электропитания ПЛИС программными методами

А. Н. Копысов, кандидат технических наук, доцент, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
А. Ю. Шаимов, аспирант, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
А. Ю. Белоусов, аспирант, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия

Статья посвящена решению вопроса снижения электропотребления программируемых логических интегральных схем (далее ПЛИС) программными способами. Решение данного вопроса особенно актуально для различного рода радиотехнических систем 5-го и 6-го поколений, в состав которых входят автономные и портативные станции.

В начале статьи приведены теоретические основы потребления ПЛИС: типы потребления (статическое и динамическое) и влияющие параметры (переключение логики, физические основы интегральной схемы и других внутренних ресурсов). Пояснена работа и приведены методические указания для специализированной встроенной утилиты PowerAnalyzer программной среды разработки Quartus II, разъяснены основные показатели, с которыми работает утилита: toggle rate (частота переключения) и static probability (вероятность нахождения сигнала в устойчивом состоянии «0» или «1»).

Далее изложен метод Pipelinning для устранения эффекта гонок (эффект, при котором сигналы, проходящие через элемент/кристалл, доходят до точки назначения с разными временными задержками). Суть метода заключается в том, что на выходе элемента, создающего задержку, устанавливается регистр. Регистр сохраняет значение на входе по тактовой частоте CLK, поэтому случайные переключения предыдущего элемента схемы будут игнорироваться. Тем самым устраняется эффект гонок и снижается динамическое потребление схемы.

В конце статьи проведены результаты снижения энергопотребления при использовании приведенных методов. Наибольший выигрыш наблюдается при минимизации потребления, тогда, когда задействовано большое количество логических элементов на кристалле ПЛИС, а также когда происходит работа на высокой частоте. Обусловлено это тем, что при использовании методов отключения тактовой частоты мы избавляемся от большого количества переключений между состояниями логических элементов (чем больше частота, тем больше таких переключений состояний в единицу времени). При количестве занятых элементов больше 50 % мы получаем снижение потребляемой мощности более чем на 5 %, а если количество занимаемых элементов 75 %, то мы получаем выигрыш около 14 %.

Ключевые слова: снижение электропотребления, портативные устройства, ПЛИС, Quartus II, Power Analyzer.

Введение

Вопрос уменьшения электропотребления портативных устройств (таких, как радиостанции) играет важную роль в процессе проектирования, реализации и их дальнейшего использования. Рациональный подход к решению данного вопроса позволяет создавать автономные по питанию системы, уменьшать размер аккумуляторной батареи, снижать перегрев внутренней электроники, что позволяет, во-первых, облегчить устройство (сократить дополнительные места под систему охлаждения и различных радиаторов), во-вторых, увеличить его сроки службы [1, 2].

В данной статье рассматриваются программные методы снижения потребляемой мощности (электропотребление в единицу времени) для программируемой логической интегральной схемы (ПЛИС), входящей в портативное устройство [3]. Целью работы является определение численной оценки снижения потребляемой мощности при использовании программных методов.

Потребление ПЛИС

Разделяется на два типа: статическое и динамическое. К первому типу относят: потребление безотносительного переключения логических уровней в проекте, потребление конкретной модели устройства ПЛИС; складывается из задействованных банков ввода/вывода, используемой логики, температуры охлаждающей системы (ОС), наличия/отсутствия кулера и/или теплопровода, температуры контактов, использования I/Os (контактов входа/выхода). Второй тип характеризуется: потреблением при перезарядке внутренних емкостей и зависимостью от частоты работы логики, количества и типа используемых ресурсов трассировки (routing) [4].

В среде разработки Quartus II имеется встроенная утилита Power Analyzer, которая оценивает динамическое и статическое потребление для

готового проекта [3]. В разрабатываемой структурной модели на ПЛИС следует производить расчет питания исходя из следующих соображений:

1. В Assignments => Settings => Operating Settings and Conditions => Temperature указываем в параметре «Use cooling solution» => «No heat sink with still air» (т. е. без использования теплоотвода и кулера).

2. Использование «Auto compute junction temperature using cooling solution» (автоматический расчет температуры контактов при том или ином способе охлаждения). Выставляем температуру окружающей среды 80 °С (например).

3. Необходимо учитывать поведение каждого сигнала в проекте. Для этого вводятся два термина – toggle rate (частота переключения) и static probability (вероятность нахождения сигнала в устойчивом состоянии «0» или «1»).

4. Правильно задаем все временно-частотные характеристики.

5. Правильно задаем все параметры проекта Quartus II в настройках Pin Planner – это напряжение, логическая структура, I/O банк (порты ввода/вывода).

Поясним важные введенные понятия.

Toggle rate – среднее число переключений (1 => 0 или 0 => 1) сигнала в течение определенного промежутка времени. Измеряется в N (число переключений)/секунда.

Static probability – доля времени, в течение которого сигнал находится в состоянии логической «1». Считается за время анализа работы прибора, например за сутки. Находится в пределах от 0 (всегда на уровне напряжения «земли») до 1 (всегда на уровне напряжения питания).

Динамическое потребление – мощность, которая увеличивается линейно с увеличением параметра toggle rate из-за более частого перезаряда емкостной нагрузки [5]. Static probability влияет на статическое потребление из-за токов утечки, что учитывается в Power Play Analyzer'e.

Для всех контактов ввода/вывода (I/O pins) желательно произвести расчет Toggle Rate и Static probability и вносить данные об этом в Assignment Editor. Ошибочно указанные данные по Toggle Rate вносят наибольший вклад в ошибку расчета энергопотребления ПЛИС. При расчете не стоит путать time-averaged toggle rate, т. е. усредненный за некоторое время параметр с highest possible toggle rate. Highest possible toggle rate – это параметр, используемый на этапе имплементации и I/O Assignment Analysis для проверки целостности сигнала и критичности перекрестных помех. Для расчета энергопотребления используется time-averaged toggle rate, рассчитываемый как количество переключений за секунду, например, если полсекунды сигнал активен, а полсекунды нет, то количество переключений должно учитывать этот факт. Эти параметры задаются для контактов информационных входов (рис. 1).

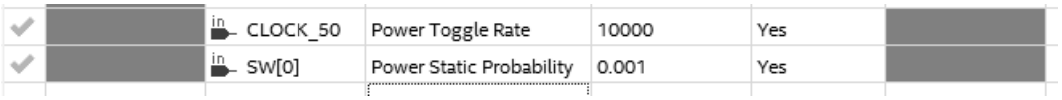


Рис. 1. Пример указания параметров Toggle Rate и Static Probability в Assignments Editor
Fig. 1. Example of specifying the parameters Switching speed and Static Probability in the Assignment Editor

Данные о контактах тактовых частот задаются в .SDC-файле проекта для временного анализа. Данные о внутренних регистрах рассчитываются на основе синхронного дизайна по заданным частотам в Time Quest Analyzer (внутреннее приложение Quartus II) [6].

Для оптимизации электропотребления используют методы:

1. Clock gating/muxing – выключение тактовых частот для частей или модулей проекта.

2. Задание нескольких тактовых частот, отладка проекта на них, переключение между ними в соответствии с режимом работы.

3. Использование сигналов Enable для логики LABs/Les, когда это возможно.

(Назначить высокий уровень на вход Enable, а действительный сигнал Enable на вход тактовой частоты CLK. Переключение входа тактовой частоты и работа всего блока происходит только во время записи или чтения.)

4. Изменение кода – это стили описания конечных автоматов, настройки IP-ядер, размеры блоков памяти.

5. Использование методов Pipelining и retiming – остановка случайных переключений.

6. Использование настройки компилятора, задание правильных параметров портов ввода/вывода и банков питания в Assignments Editor.

Для мультиплексирования тактовых входов (clk) используется IP-ядро ALTCLKCTRL. На рис. 2 представлен пример настройки блока.

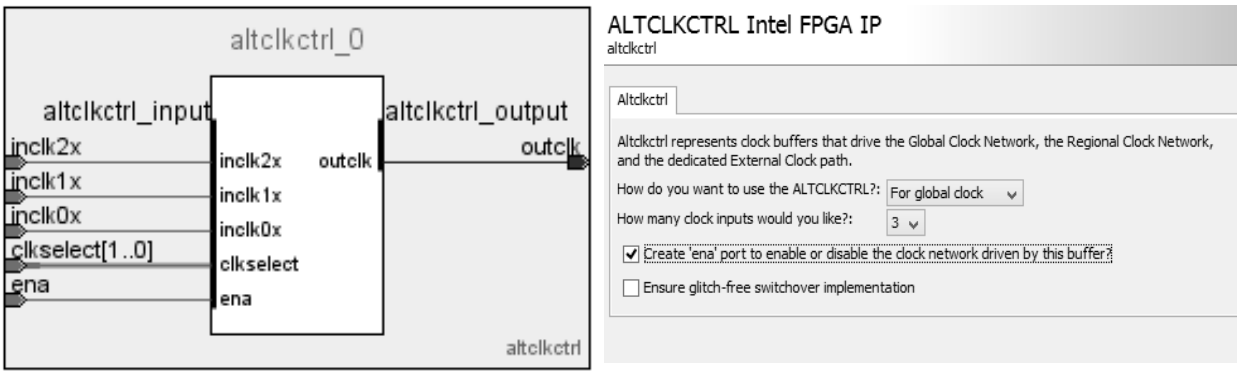


Рис. 2. IP-ядро ALTCLKCTRL и его параметры
Fig. 2. ALTCLKCTRL IP core and its parameters

Этот блок использует специализированные линии соединений для тактовых частот, что уменьшает нежелательные последствия. В случае когда описываем мультиплексирование частоты синтаксисом языка, то задействуем логические блоки LE и линии, неспециализированные для тактовых частот, появляются джиттер (частотная нестабильность) и задержки [7].

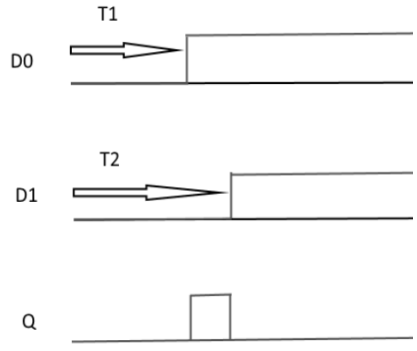
Метод Pipelinning для устранения эффектов гонок

При проектировании на ПЛИС необходимо учитывать время распространений сигналов по кристаллу [8]. При этом возникает эффект гонок, когда сигналы доходят до точки назначения с разными временными задержками. С этим эффектом связаны нежелательные переключения состояния элементов, а следовательно, возрастает динамическое потребление схемы. Наиболее подверженный этому явлению элемент «исключающий ИЛИ» (XOR), таблица истинности представлена на рис. 3, *а*. При эффекте гонок, даже если состояние элемента должно остаться без изменения, происходит переключения состояния выхода. На рис. 3, *б* представлен пример временной диаграммы состояния элемента «исключающий ИЛИ», где: D0 и D1 – входы элемента, Q – выход.

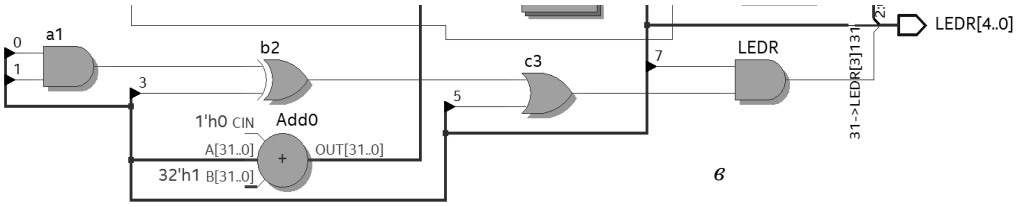
На рис. 3, *в* в качестве примера представлен участок схемы, включающий в себя элемент XOR b2, схема получена при компиляции проекта в среде Quartus II в утилите RTL Viewer. При переключении элемента b2 произойдет и переключение последующих элементов схемы.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

а



б



в

Рис. 3. Явление эффекта гонок с использованием элемента XOR
Fig. 3. The phenomenon of the racing effect using the XOR element

Для устранения паразитического динамического потребления используется метод pipelinning. При этом на выходе элемента устанавливается регистр. Регистр сохраняет значение на входе по тактовой частоте CLK, поэтому случайные переключения предыдущего элемента схемы будут игнорироваться. Тем самым устраняется эффект гонок и снижается динамическое потребление схемы. На рис. 4 представлен пример исправленной схемы с добавлением регистра.

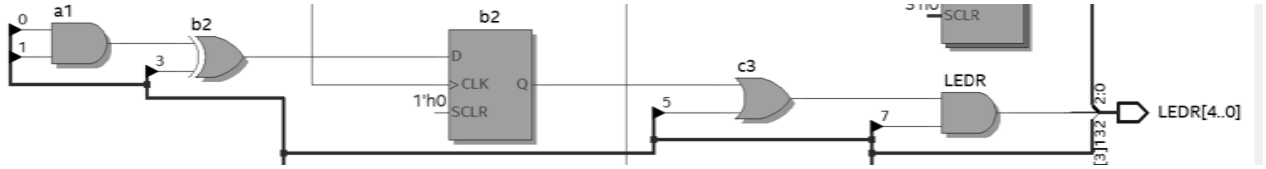


Рис. 4. Применения метода Pipelinning
Fig.4. Applications of the Pipelining Method

Этот метод особенно актуален для схем, включающих в себя большое количество элементов комбинаторной логики. Но при этом необходимо учитывать внесенные задержки регистром в логике работы схемы [9].

Работа с Power Analyzer

Для запуска Power Analyzer необходимо выполнить следующий порядок: Processing –> Start –> Start PowerPlay Power Analyzer. Это необходимо сделать, так как Power Analyzer является частью компилятора, но по умолчанию выключен [10]. Есть возможность самостоятельно добавить его подключение к проекту при настройках потока компиляции. Отчет после компиляции этапа Power Analyzer приведен на рис. 5.

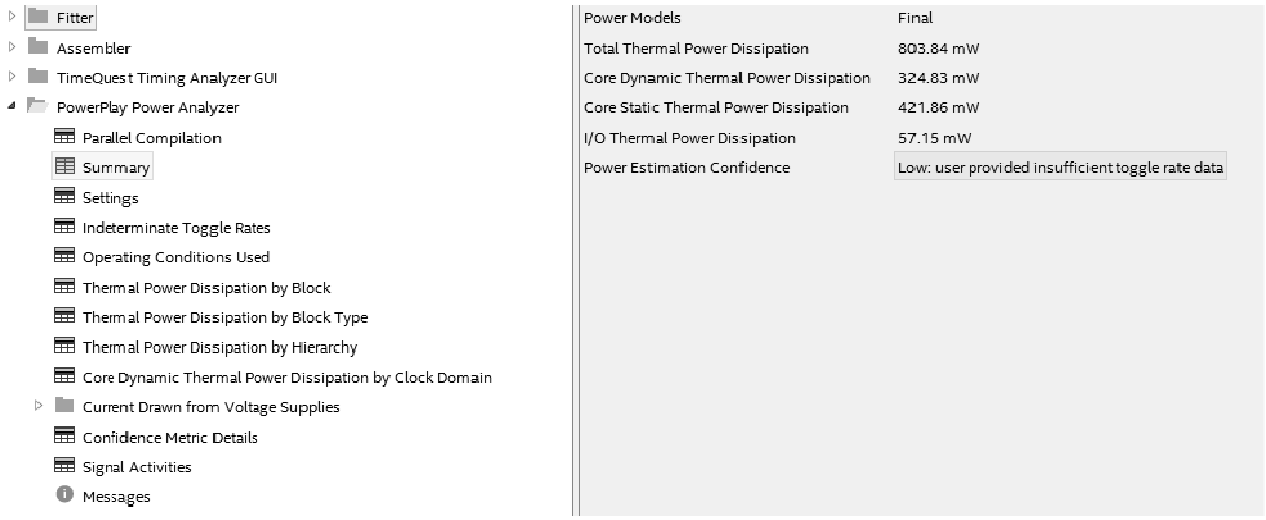


Рис. 5. Окно отчета Power Analyzer
Fig. 5. Power Analyzer Report Window

Итоговый отчет состоит из общего отчета, настройки, выбранных условий, потребления по блокам, типам блоков, иерархии блоков, по доменам тактовых частот и т. д.

Поставленный эксперимент и полученные результаты

Рассмотрим, как изменяется электропотребление (мощность) ПЛИС в зависимости от различных параметров. В качестве устройства выберем плату Cyclone IV EP4CE115F29C7 фирмы Intel. Основные характеристики платы представлены на рис. 6 (количество логических элементов, количество входных/выходных разъёмов, размер внутренней памяти, количество синтезаторов тактовых частот PLLs) [11].

Посмотрим, как влияют toggle rate и static probability.

Посмотрим, как влияют toggle rate и static probability. Сперва приведем результаты компиляции и анализа проекта без описания параметров для входов toggle rate и static probability (рис. 7).

Оценим влияние toggle rate и static probability (рис. 8, 9).

Flow Status	Successful - Wed May 19 11:06:11 2021
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	proj_stat
Top-level Entity Name	proj_stat
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	244 / 114,480 (< 1 %)
Total registers	0
Total pins	98 / 529 (19 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	18 / 532 (3 %)
Total PLLs	0 / 4 (0 %)

Рис. 6. Окно отчета компиляции и анализа проекта, показывающее количество задействованных элементов ПЛИС
Fig. 6. The project compilation and analysis report window showing the number of FPGA elements involved

Power Analyzer Status	Successful - Wed May 19 11:06:11 2021
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	proj_stat
Top-level Entity Name	proj_stat
Family	Cyclone IV E
Device	EP4CE115F29C7
Power Models	Final
Total Thermal Power Dissipation	172.42 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	98.61 mW
I/O Thermal Power Dissipation	73.81 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Рис. 7. Окно отчета компиляции и анализа проекта с использованием Power Analazer без описания параметров для входов toggle rate и static probability
Fig.7. The report window for compiling and analyzing a project using Power Analyzer without describing parameters for the toggle rate and static probability inputs

	tatu	From	To	Assignment Name	Value	Enabled	Entity
1	✓		a	Power Toggle Rate	500000	Yes	
2	✓		b	Power Toggle Rate	500000	Yes	
3	✓		reset	Power Toggle Rate	10	Yes	
4	✓		clk_1	Power Toggle Rate	50000000	Yes	
5	✓		a	Power Static Probability	0.5	Yes	
6	✓		b	Power Static Probability	0.5	Yes	
7	✓		reset	Power Static Probability	0.001	Yes	
8	✓		clk_1	Power Static Probability	0.5	Yes	
9	✓		a	I/O Standard	3.3-V LVTTL	Yes	proj_stat
10	✓		b	I/O Standard	3.3-V LVTTL	Yes	proj_stat
11	✓		reset	I/O Standard	3.3-V LVTTL	Yes	proj_stat
12	✓		clk_1	I/O Standard	3.3-V LVTTL	Yes	proj_stat

Рис. 8. Описание параметров toggle rate и static probability для входов
Fig.8. Description of the toggle rate and static probability parameters for inputs

Power Analyzer Status	Successful - Wed May 19 11:25:17 2021
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	proj_stat
Top-level Entity Name	proj_stat
Family	Cyclone IV E
Device	EP4CE115F29C7
Power Models	Final
Total Thermal Power Dissipation	178.79 mW
Core Dynamic Thermal Power Dissipation	0.92 mW
Core Static Thermal Power Dissipation	98.63 mW
I/O Thermal Power Dissipation	79.24 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Рис. 9. Окно отчета компиляции и анализа проекта с использованием Power Analazer с учетом входов toggle rate и static probability
Fig. 9. The window of the project compilation and analysis report using Power Analyzer, taking into account the toggle rate and static probability inputs

Как видно из рис. 7 и 8, показания выходной мощности изменились, но не сильно, так как в проекте небольшое количество задействованных элементов (всего меньше процента, рис. 8).

Приведем в табл. 1 результаты компиляции и анализа влияния изменения рабочей частоты и разного количества задействованных элементов на итоговое потребление мощности. Также сравним результаты с использованием комплекса специальных методик, представленных выше по тексту (использование тактовых частот только по необходимости, метод Pipelining и retiming, грамотное описание кода под специализированные блоки и т. д.).

Таблица 1. **Потребляемая мощность проекта, исходя из значения рабочей частоты и количества задействованных логических элементов**
Table 1. **The power consumption of the project, based on the value of the operating frequency and the number of logic elements involved**

Процент задействованных логических элементов из113 480, %	Потребляемая мощность на частоте 50 МГц, мВт	Потребляемая мощность на частоте 100 МГц, мВт	Потребляемая мощность на частоте 150 МГц, мВт	Потребляемая мощность на частоте 200 МГц, мВт
5	192,10	298,02	361,42	408,34
15	256,75	556,67	756,07	919,56
25	326,61	814,23	1112,71	1403,43
35	391,49	1089,03	1500,04	1924,92
45	453,95	1334,26	1870,26	2396,86
55	517,14	1551,96	2216,06	2823,40
65	574,46	1785,03	2557,24	3302,74
75	643,25	2028,24	2974,65	3812,74

Для наглядности представим результаты в виде графиков (рис. 10).

В табл. 2 приведены результаты с учетом использования программных методов по снижению электропотребления (существует и лучший вариант минимизации) [12]. В скобках написан процент уменьшения потребления в сравнении с результатами табл. 1.

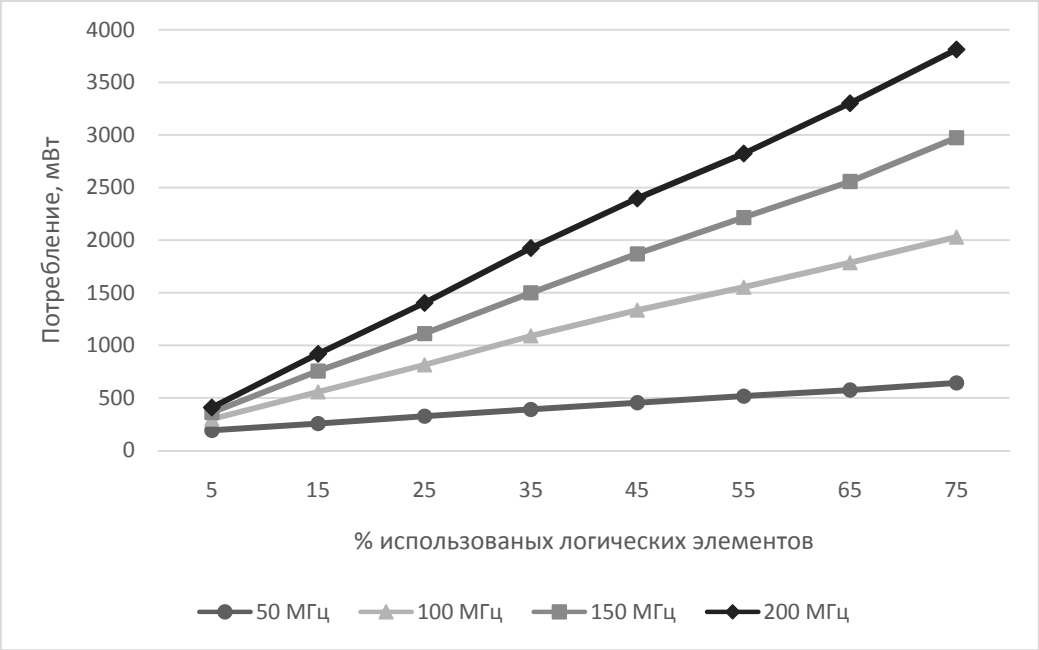


Рис.10. Графики потребляемой мощности в зависимости от рабочей частоты и количества используемых логических элементов
Fig.10. Graphs of power consumption depending on the operating frequency and the number of logic elements used

Таблица 2. **Потребляемая мощность проекта исходя из значения рабочей частоты и количества задействованных логических элементов с применением программных методов по снижению мощности**
Table 2. **The power consumption of the project, based on the value of the operating frequency and the number of logic elements involved using software methods to reduce power**

Процент задействованных логических элементов из 114 480, %	Потребляемая мощность на частоте 50 МГц, мВт	Потребляемая мощность на частоте 100 МГц, мВт	Потребляемая мощность на частоте 150 МГц, мВт	Потребляемая мощность на частоте 200 МГц, мВт
5	186,63 *(2,85 %)*	289,08 *(3,00 %)*	350,77 *(2,95 %)*	397,12 *(2,75 %)*
15	248,23 *(3,32 %)*	534,4 *(4,00 %)*	726,68 *(3,89 %)*	882,73 *(4,01 %)*
25	313,55 *(4,00 %)*	776,66 *(4,61 %)*	1069,2 *(3,91 %)*	1320,51 *(5,91 %)*
35	375,88 *(3,99 %)*	1031,58 *(5,28 %)*	1424,34 *(5,05 %)*	1790,67 *(6,97 %)*
45	431,74 (*4,89 %)*	1254,2 *(6,00 %)*	1751,07 *(6,37 %)*	2210,25 *(7,97 %)*
55	490,39 *(5,17 %)*	1443,75 *(6,97 %)*	2050,94 *(7,45 %)*	2563,76 *(9,20 %)*
65	531,76 *(7,43 %)*	1625,38 *(8,94%)*	2327,02 *(9,00 %)*	2916,49 *(11,69 %)*
75	580,06 *(9,82 %)*	1805,85 *(10,96 %)*	2617,12 *(12,02 %)*	3256,34 *(14,59 %)*

Как видно из табл. 2, наибольший выигрыш при минимизации потребления приходится тогда, когда используется много ресурсов ПЛИС (логических элементов) и чем выше рабочая частота. Обусловлено это тем, что при использовании методов отключения тактовой частоты мы избавляемся от большого количества переключений между состояниями логических элементов (тем больше частота, тем больше таких переключений состояний в единицу времени) [13]. При количестве занятых элементов больше 50 % мы получаем снижение потребляемой мощности более чем 5 %.

На рис. 11 наглядно отобразим, как происходит снижение потребления для случая работы проекта на частоте 150 МГц.

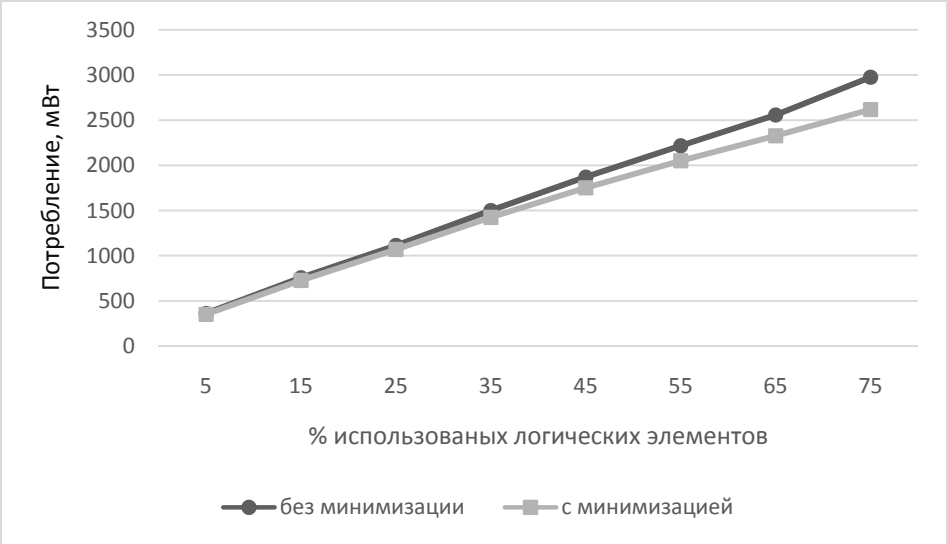


Рис. 11. Графики потребляемой мощности в зависимости от применения программных способов минимизации
Fig. 11. Graphs of power consumption depending on the application of software minimization methods

Отметим, что результат снижения электропотребления после применения методов не прогнозируем заранее и зависит от опыта разработчика [14].

Анализ полученных результатов

Можно заметить, что потребление ПЛИС существенно изменяется для одной и той же интегральной микросхемы в зависимости от загружаемого проекта. ПЛИС с большим задействованием количества логических элементов (свыше 55 %) и высоким уровнем быстродействия (рабочей частотой выше 150 МГц) потребляют не менее 2000 мВт, и с дальнейшим увеличением параметров потребляемая мощность продолжает резко возрастать. При разработке проекта крайне нежелательно постоянно возвращаться к вопросу питания цифровых компонентов, поэтому имеет целесообразность предусмотреть определенный запас мощности источника питания на начальных этапах разработки.

Заключение

Потребляемая мощность является критически важным параметром для разработчиков встраиваемых систем, особенно портативных устройств с батарейным питанием. Правильный выбор архитектуры ПЛИС, используемых для построения системы, позволяет уменьшить энергопотребление и улучшить такие характеристики разрабатываемого приложения, как гибкость, стоимость и габаритные размеры [15].

Встроенные программные возможности среды разработки позволяют оценить потребляемую мощность на разных этапах проектирования системы и создать схемное и топологическое решение, позволяющее снизить энергопотребление разрабатываемого устройства.

Для анализа потребляемой мощности в Quartus II используется утилита Power Analyzer. Чтобы получить предварительную оценку потребляемой мощности, необходимо правильно задавать все временные характеристики тактовых частот и сигналов, а также параметры Toggle Rate и Static Probability в Assignments Editor. При неправильно заданных параметрах, результат работы Power Analyzer будет абсолютно некорректным. Также для снижения электропотребления ПЛИС обычно уменьшают динамическое потребление, используют мультиплексирование тактовых частот и отключение подачи частоты в незадействованные в текущий момент времени блоки. Численная оценка при использовании программных методов снижения потребления: при количестве занятых элементов больше 50 % кристалла ПЛИС мы получаем снижение потребляемой мощности более чем 5 %.

Библиографические ссылки

1. *Зинченко М. Ю., Левадний А. М., Гребенко Ю. А.* Реализация LDPC декодера на ПЛИС и оптимизация потребляемой мощности // T-COMM: телекоммуникации и транспорт. 2020. Т. 14, № 3. С. 4–10. eISSN: 2072-8743.

2. *Vipin K.* FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications / K. Vipin, S. A. Fahmy // ACM Computing Surveys. 2018. Vol. 51. No. 4. P. 3193827. DOI 10.1145/3193827.

3. Wei G. J., Hou Y. Z., Cui Q. M., Deng G., Tao X. F. YOLO acceleration using FPGA architecture // International conference on communications in China. 2018. Pp. 734-735. ISSN: 2377-8644.

4. *Городков П. С.* Особенности разработки HDL-проектов для реализации в базисе ПЛИС серии

5578ТС // Электроника: наука, технология, бизнес. 2017. № 5. С. 50–59. eISSN: 1992-4186.

5. Quartus II // Altera [Сайт]. URL: http://altera.ru/soft_quartus.html (дата обращения 20.06.2021).

6. Lara-Nino C. A., Morales-Sandoval M., Diaz-Perez A. Small lightweight hash functions in FPGA // 9th IEEE American symposium on circuits and systems. 2018. pp. 55-58. ISSN: 2330-9954.

7. Firmansyah I., Wijayanto Y. N., Yamaguchi Y. 2D stencil computation on Cyclone V SoC FPGA using OpenCL // Internatioanal conference on radar, antenna, microwave, electronics and telecommunications. 2018. pp. 121-124. IEEE 345 E47TH ST, NewYork, NY 10017 USA.

8. Al-Shorman M. Y., Al-Kofahi M. M., Al-Kofahi O. M. A practical microwatt-meter for electrical energy measurement in programmable devices // Measure Contr. 2018. Vol. 51. № 9-10. Pp. 383-395. DOI: 10.1177/0020294018794350.

9. Kulkarni V. A., Udupi G. R. A simplified method for instruction level energy estimation for embedded system // European Journal of engineering and technology research. 2017. Vol. 2, № 5. Pp. 56-59. DOI: 10.24018/ejers.2017.2.5.359.

10. СБИС ПЛ подсемейства Cyclone 5 // ALTERA intel FPGA [Сайт]. URL: http://altera.ru/sbis-pl-cyclone-IV-E.html (дата обращения: 10.05.2019).

11. *Тюрин С. Ф., Вихорев Р. В.* Усовершенствованный метод реализации в FPGA систем логических функций, заданных в СДНФ // Инженерный вестник Дона. 2017. № 1. 39 с. eISSN: 2073-8633.

12. *Матвеев Д. А., Бальзамов А. Ю.* Разработка системы управления полупроводниковым преобразователем электроэнергии на ПЛИС // Практическая силовая электроника. 2017. № 4 (68). С. 18–26.

13. *Волобуев С. В., Евдокимов А. П., Рябцев В. Г.* Применение аппаратных и программных средств для изменения режимов работы цифровых устройств, реализованных на ПЛИС // Инженерный вестник Дона. 2017. № 4(47). С. 74.

14. Решение задач трассировки межсоединений с ресинтезом для реконфигурируемых систем на кристалле / С. В. Гаврилов, Д. А. Железников, В. М. Хватов // Известия высших учебных заведений. Электроника. 2017. Т. 22, № 3. С. 266-275. DOI 10.24151/1561-5405-2017-22-3-266-275.

15. Real-time dispersion interferometry for density feedback in fusion devices / K. J. Brunner, M. Hirsch, J. Knauer [et al.] // Journal of Instrumentation. 2018. Vol. 13. No. 9. P. P09002. DOI 10.1088/1748-0221/13/09/P09002.

References

1. Zinchenko M.Yu., Levadnij A.M., Grebenko Yu.A. [Implementation of LDPC decoder on FPGA and optimization of power consumption]. *T-COMM: telekommunikacii i transport,* 2020, vol. 14, no. 3, pp 4-10 (in Russ.). eISSN: 2072-8743.

2. Vipin K. [FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications]. *ACM Computing Surveys,* 2018, vol. 51, no. 4, p. 3193827. DOI 10.1145/3193827.

3. Wei G.J., Hou Y.Z., Cui Q.M., Deng G., Tao X.F. [YOLO acceleration using FPGA architecture]. *International conference on communications in China*, 2018, pp. 734-735. ISSN: 2377-8644.

4. Gorodkov P.S. [Features of the development of HDL projects for implementation in the basis of FPGA series 5578TS]. *E`lektronika: nauka, texnologiya, biznes*, 2017, no. 5, pp. 50-59 (in Russ.). eISSN: 1992-4186.

5. *Quartus II, Altera* [Quartus II, Altera]. Available at: http://altera.ru/soft_quartus.html (acessed: 20.06.2021).

6. Lara-Nino C. A., Morales-Sandoval M., Diaz-Perez A. [Small lightweight hash functions in FPGA]. *9th IEEE American symposium on circuits and systems,* 2018, pp. 55-58. ISSN: 2330-9954.

7. Firmansyah I., Wijayanto Y. N., Yamaguchi Y. [2D stencil computation on Cyclone V SoC FPGA using OpenCL]. *Internatioanal conference on radar, antenna, microwave, electronics and telecommunications*, 2018, pp. 121-124. IEEE 345 E47TH ST, NewYork, NY 10017 USA.

8. Al-Shorman M.Y., Al-Kofahi M.M., Al-Kofahi O.M. [A practical microwatt-meter for electrical energy measurement in programmable devices]. *Measure Contr*, 2018, vol. 51, no. 9-10. Pp. 383-395. DOI: 10.1177/0020294018794350.

9. Kulkarni V. A., Udupi G. R. [A simplified method for instruction level energy estimation for embedded system]. *European Journal of engineering and technology research*, 2017, vol. 2, no.5. pp. 56-59. DOI: 10.24018/ejers.2017.2.5.359.

10. *SBIS PL podsemeistva Cyclone 5, ALTERA intel FPGA* [SBIS PL family Cyclone 5, ALTERA intel FPGA]. Available at: http://altera.ru/sbis-pl-cyclone-IV-E.html (acessed: 10.05.2019).

11. Tyurin S.F., Vikhorev R.V. [An improved method for implementing logical functions specified in the SDNF in FPGA systems]. Inzhenernyi vestnik Dona, 2017, no. 1, pp. 39 (in Russ.). eISSN: 2073-8633.

12. Matveev D.A. [Development of a control system for a semiconductor electric power converter on an FPGA]. *Prakticheskaya silovaya elektronika*, 2017, no. 4(68), pp. 18-26 (in Russ.).

13. Volobuev S.V., Evdokimov A.P., Ryabtsev V.G. [Application of hardware and software tools for changing the operating modes of digital devices implemented on FPGAs]. *Inzhenernyi vestnik Dona*, 2017, no. 4, pp. 74 (in Russ.).

14. Gavrilov S.V., Zheleznikov D.A., Khvatov V.M. [Reshenie zadach trassirovki mezhsoedinenii s resintezom dlya rekonfiguriruemykh sistem na kristalle]. *Izvestiya vysshikh uchebnykh zavedenii. Elektronika*, 2017, vol. 22, no. 3, pp. 266-275 (in. Russ.). DOI 10.24151/1561-5405-2017-22-3-266-275.

15. Brunner K.J., Hirsch M., Knauer J. [Real-time dispersion interferometry for density feedback in fusion devices]. Journal of Instrumentation, 2018, vol. 13, no. 9, p. P09002. DOI 10.1088/1748-0221/13/09/P09002.

* * *

Ways to Reduce FPGA Power Supply by Software Methods

A. N. Kopysov, PhD in Engineering, Associate Professor, Kalashnikov ISTU, Izhevsk, Russia
A. Yu. Shaimov, Post-graduate, Kalashnikov ISTU, Izhevsk, Russia
A. Yu. Belousov, Post-graduate, Kalashnikov ISTU, Izhevsk, Russia

The paper is devoted to solving the problem of reducing the power consumption of programmable logic integrated circuits (hereinafter FPGAs) by software methods. The solution of this issue is especially relevant for various types of radio engineering systems, which include autonomous and portable stations.

At the beginning of the paper, the theoretical foundations of FPGA consumption are given: types of consumption (static and dynamic) and influencing parameters (switching logic, the physical foundations of the integrated circuit and other internal resources). The work is explained and methodological guidelines are given for the specialized built-in PowerAnalyzer utility of the Quartus II software development environment, the main indicators with which the utility works are explained: toggle rate (switching frequency) and static probability (the probability of finding a signal in a stable state "0" or "1").

The following is a Pipelinning method for eliminating the racing effect (the effect in which signals passing through an element/crystal reach the destination point with different time delays). The essence of the method is that a register is set at the output of the element that creates the delay. The register stores the value at the input at the CLK clock frequency, so random switching of the previous circuit element will be ignored. This eliminates the effect of racing and reduces the dynamic consumption of the circuit.

At the end of the paper, the results of reducing energy consumption using the above methods are presented. The greatest gain is observed when minimizing consumption, when a large number of logic elements on the FPGA chip are involved, as well as when high-frequency operation occurs. This is due to the fact that when using the methods of switching off the clock frequency, we get rid of a large number of switches between the states of logic elements (the higher the frequency, the more such state switches per unit of time). If the number of occupied elements is more than 50%, we get a reduction in power consumption of more than 5%, and if the number of occupied elements is 75%, then we get a gain of about 14%.

Keywords: minimization of power consumption, portable devices, FPGA, Quartus II, PowerAnalyzer.

Получено: 16.09.2021