

УДК 004.31

А.С. Шашков

ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ С ПОНИЖЕННЫМ ЭНЕРГОПОТРЕБЛЕНИЕМ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ UPF-ОПИСАНИЯ ПОДСИСТЕМЫ ПИТАНИЯ

В рамках проблемы проектирования энергетически эффективных цифровых систем приводятся различные методы понижения энергопотребления. Обосновывается необходимость дополнительных средств формального поведенческого описания доменной подсистемы питания. Рассматриваются элементы доменной подсистемы питания и этапы проектирования цифровой системы с учетом ее разделения на домены питания. Дается пример описания системы с применением формата UPF.

Введение

Проблема проектирования энергетически эффективных (с пониженным энергопотреблением) вычислительных систем всегда была одной из основных в индустрии электронных систем. В настоящее время эта проблема стоит наиболее остро [1]. Необходимость в удешевлении микросхем и увеличении их компактности, обусловленная как широким распространением все усложняющейся мобильной электроники, так и потребностью в высокоэффективных центрах обработки данных, поставила проблему энергосбережения на новый уровень. Так, производительность систем на кристалле стала существенно ограничиваться возможностями автономных источников питания для мобильных систем, в то время как многочисленные центры обработки данных затруднительно и затратно эксплуатировать и расширять ввиду ограниченного ресурса электроэнергии [2]. Одной из актуальных проблем является также проектирование эффективных и компактных средств отвода тепла как в мобильных, так и в стационарных системах. В таких условиях среди всех целей проектирования компьютерных систем на одном из первых мест стоит увеличение производительности, приходящейся на единицу рассеиваемой мощности. Все это способствовало активному развитию методов и технологий, позволяющих снижать энергопотребление интегральных схем при сохранении их высокой производительности [2–6].

В настоящей обзорной статье рассматриваются существующие методы снижения энергопотребления электронных систем, однако особое внимание уделяется одному из новейших методов снижения энергопотребления – использованию доменной системы питания, а также способу поведенческого проектирования системы питания с использованием средств формата, специально для этого предназначенного и названного Unified Power Format (UPF).

1. Обзор методов понижения энергопотребления для однокристалльных систем

Рассеяние мощности интегральной микросхемы состоит из двух составляющих: динамической и статической мощностей. Динамическая мощность пропорциональна квадрату напряжения питания, частоте переключений элементов и эффективной емкости элементов. Статическое энергопотребление характеризуется величиной токов утечки, которая возрастает с ростом напряжения питания, уменьшением пороговых напряжений переключения транзисторов и с увеличением ширины транзисторов [1, 5]. С другой стороны, понижение напряжения питания, увеличение пороговых напряжений, уменьшение частоты переключений приводят к снижению вычислительной производительности микросхемы.

При технологических процессах производства микросхем более 90 нм динамическая составляющая мощности существенно преобладает. Для снижения динамической мощности были разработаны следующие методы [1]:

- оптимизация цепей синхросигнала и управление пропуском синхросигнала;
- изоляция операндов;
- реструктуризация логики;
- оптимизация размеров транзисторов;
- использование буферов для минимизации времени переключения вентиляей;
- обмен выводами (выбор выводов с наименьшей емкостной нагрузкой).

Исследования показывают [1], что оптимизация цепей синхросигнала позволяет снизить динамическую мощность до 20 %. Другие методы позволяют снизить динамическую мощность до 5 % каждый.

Методы снижения динамической мощности эффективно реализуются в САПР при синтезе проекта в автоматическом режиме. Часть задач по оптимизации логики может быть решена разработчиком на уровне RTL-описаний. Таким образом, использование методов снижения динамической мощности предполагает наличие следующих входных данных:

- RTL-описание проекта;
- задания параметров оптимизации энергопотребления для САПР.

С уменьшением технологических процессов уменьшается и пороговое напряжение переключения транзисторов, что обуславливает существенный рост токов утечки, а значит, и увеличение статической мощности [3]. Современные массовые технологические процессы производства микросхем уже давно перешагнули пороговый рубеж в 65 нм, после которого статическая мощность микросхем стала превышать динамическую мощность [1]. Также с уменьшением технологических процессов значительно возросла и плотность рассеиваемой мощности кристаллов [6], что затруднило отвод тепла. В связи с этим возникла острая потребность в методах, позволяющих наряду с уменьшением динамической мощности в первую очередь уменьшать статическую мощность всей системы на кристалле. К ним относятся:

- использование нескольких доменов (областей кристалла) с различными напряжениями питания;
- использование отключаемых доменов питания;
- динамическое изменение напряжений питания;
- динамическое изменение частоты сигнала синхронизации;
- распределение памяти на блоки, расположенные в разных доменах питания (часть памяти может отключаться или переводиться в режим удержания данных без возможности чтения и записи);
- использование библиотек логических вентиляей с разными порогами переключения (реализуется автоматически в САПР);
- использование технологии смещения напряжения подложки транзисторов в режимах высокого и низкого энергопотребления (требует дополнительного источника питания).

Исследования показывают [1], что для технологических процессов от 65 нм и ниже при комбинировании всех современных технологий энергосбережения систем на кристалле наибольший вклад в сохранение энергии вносит технология отключаемых доменов питания, позволяющая сократить среднюю мощность токов утечки до 50 раз. Технологии динамического изменения напряжения питания и частоты синхросигнала позволяют снизить динамическую мощность до 70 %, а статическую мощность – до трех раз.

Доменные методы энергосбережения не представляется возможным эффективно реализовать автоматически ввиду того, что фактически эти методы являются задачами логического проектирования активной доменной сети или задачами, сопряженными с логической составляющей проекта. Таким образом, использование вышеперечисленных методов предполагает следующие входные данные для синтеза проекта:

- RTL-описание логической иерархии проекта;
- описание доменов и других элементов сети питания;
- описание логики управления активной частью сети питания (также реализуется средствами RTL).

Анализируя входные данные, необходимые для описания энергетически эффективных систем на кристалле, можно заключить, что часть описания энергетически эффективной системы можно выполнить привычными средствами HDL (Hardware Description Language, язык описания аппаратуры), такими как VHDL [7] или System Verilog [8], дополняя их выбором настроек оптимизации проекта в САПР. Однако описание активной сети доменов питания и других элементов доменной системы невозможно с помощью только VHDL или System Verilog ввиду того, что данные языки изначально не были предназначены для этой цели. Поэтому возникла проблема формального описания проектов цифровых систем с учетом обеспечения их энергетической эффективности уже на начальном этапе проектирования, т. е. на уровне разработки исходных HDL-описаний.

2. Элементы доменных систем питания

В общем случае для описания системы с управляемым питанием на поведенческом уровне можно выделить следующие основные элементы [1, 9, 10].

Описание логической составляющей проекта. Представляет собой RTL-описание модулей, составляющих логическую часть иерархии проекта.

Порты и цепи питания (Supply Ports, Supply Nets). Цепи питания и земли, подключенные к соответствующим портам питания и земли, обеспечивают питание для соответствующих элементов проекта: как логические элементы (RTL), так и элементы обеспечения энергетической эффективности имеют свои источники питания. Цепи питания могут логически характеризоваться не только параметром выключения-включения, но и уровнем напряжения. Недостаточное напряжение питания неизбежно приводит к ненормальной работе питаемого компонента системы.

Домены питания (Power Domains). Логические элементы проекта (RTL-модули) распределяются в соответствующие обособленные домены питания. Домены могут быть логически вложенными в другие домены, несколько доменов могут быть вложены в один домен. Таким образом, проект будет иметь не только иерархию RTL-модулей, но и иерархию доменов питания. Разделение системы на домены питания позволяет выделить участки системы с общей стратегией питания (с единым набором правил питания). Так, разные домены могут питаться от различных цепей питания, имеющих различные напряжения. Домены могут иметь несколько различных источников питания: основной для логических элементов и специальные для элементов обеспечения энергетической эффективности.

Переключатели питания (Power Switches). Одним из элементов стратегии питания для домена может являться возможность отключения всех логических элементов домена с помощью переключателя (переключателей). В этом случае как динамическое, так и статическое потребление энергии элементами домена сводится к минимуму. Переключатели также могут иметь несколько входных портов питания при одном выходном. Переключатели питания управляются специальными внешними логическими сигналами.

Регистры сохранения состояния для выключаемых доменов (State Retention). Отключение питания от элемента памяти (регистра) приводит к потере его состояния. Регистры сохранения по сигналам управления сохраняют и восстанавливают содержимое целевых регистров в процессе выключения-включения основного питания. Регистры сохранения состояния имеют отдельные источники питания, отличные от основных (и отключаемых) источников питания для домена, а также позволяют ускорить процесс инициализации модуля после его очередного включения. Проектировщик конкретного RTL-модуля должен выделить некоторое подмножество регистров, для которых необходимы регистры сохранения. После очередного включения домена и восстановления данных из регистров сохранения должна быть гарантирована нормальная логическая работоспособность подсистем этого домена.

Средства изоляции сигналов, исходящих из отключаемых доменов (Isolation). В процессе отключения питания все сигналы, выходящие из домена, переходят в неопределенное (плавающее) состояние. Принимающий данный сигнал модуль, входящий в состав одного из включенных доменов, может нарушить свою логическую работу из-за случайного переключения этого входного для себя сигнала. Нежелательный промежуточный уровень сигнала также мо-

жет привести к высокому потреблению тока по данной линии сигнала. Для решения этих проблем выходные сигналы отключаемого домена пропускаются через блоки изоляции. По управляющему сигналу включения блок изоляции выдает вместо входного неопределенного сигнала задаваемое разработчиком выходное значение сигнала 0, 1, Z или «защелкнутое» в регистр предыдущее определенное значение входного сигнала.

Средства изменения уровня напряжения сигналов, проходящих между доменами с различными уровнями напряжения (Level Shifters). На границах доменов, имеющих различные напряжения питания, устанавливаются блоки изменения уровня напряжения сигналов. Таким образом, напряжение единицы для выходного сигнала одного домена будет однозначно восприниматься как напряжение единицы в любом другом домене с другим напряжением питания, который принимает данный сигнал.

Средства логического управления элементами обеспечения энергетической эффективности. Как было описано выше, такие элементы доменной системы питания, как переключатели питания доменов, блоки изоляции доменов, регистры сохранения, управляются некоторыми специальными сигналами. За формирование этих сигналов может отвечать либо отдельное RTL-устройство (модуль управления питанием), либо несколько устройств. Таким образом, система управления питанием может быть централизованной или распределенной, что задается разработчиком. Для перевода домена из включенного состояния в выключенное может потребоваться определенная разрозненная во времени последовательность управляющих сигналов. Так, сначала домен необходимо изолировать от внешних для него доменов. Затем необходимо сохранить содержимое определенных регистров в соответствующие регистры сохранения. Только после этого на переключатель питания домена можно подать сигнал выключения питания. Процедура включения домена должна идти в логически обратном порядке: включение питания; возможно, сброс логических элементов домена, затем восстановление состояния регистров из регистров сохранения и отключение блоков изоляции. Также управляющая сменой режима питания домена последовательность может быть дополнена сигналами включения-выключения пропуска синхросигнала.

Список энергетических состояний системы. Стратегия управления состояниями энергопотребления характеризуется списком всех возможных для конкретной системы глобальных состояний питания (включено, сон, выключено и т. п.). Каждое состояние питания определяется неизменным во времени набором состояний всех доменов системы для такого глобального состояния. Состояние домена определяется состоянием портов и цепей питания домена (включено или выключено, текущее напряжение). Изменение списка глобальных энергетических состояний может потребовать изменения набора необходимых элементов обеспечения энергетической эффективности. Например, нет необходимости в блоках изоляции между доменами, которые отключаются одновременно для любого глобального состояния энергопотребления системы.

Представленное описание элементов системы с разделением на домены предполагает некоторую обособленность и независимость логической составляющей проекта от элементов, обеспечивающих работу подсистемы питания. Однако если разработка логических компонентов проекта осуществляется с учетом последующего внедрения в доменную структуру питания, то конечная структура логического компонента, изначально целостного, может быть впоследствии разделена на составляющие, предназначенные для разных доменов: часть структуры компонента может быть отключаемой или может питаться от другого источника питания.

В общем случае разработка поведенческого описания энергетически эффективной доменной системы включает несколько этапов [10]:

1. Спецификация логики средствами RTL и индивидуальная верификация RTL-компонентов.
2. Определение доменов питания и распределение логических элементов в них.
3. Спецификация списка режимов питания.
4. Спецификация необходимых элементов обеспечения энергетической эффективности для доменов (переключателей, изоляции, регистров сохранения и т. п.).
5. Спецификация компонентов логики управления (описанной на RTL) для элементов обеспечения энергетической эффективности.
6. Верификация всей системы с учетом всех компонентов как логической, так и энергетической составляющих проекта.

3. Основные положения технологии UPF

Представленные выше элементы, позволяющие описать доменную структуру проекта, являются, по сути, поведенческими элементами, описывающими логическую структуру энергетической подсистемы проекта. Эта структура, в свою очередь, тесно связана с логической составляющей проекта: с одной стороны, подсистема питания управляется логической составляющей, с другой стороны, состояния элементов подсистемы питания влияют на работоспособность логической составляющей проекта. Однако текущие версии языков проектирования логических схем VHDL и System Verilog не имеют никаких специальных средств для описания (а значит, верификации и синтеза) проекта с учетом цепей питания. Ввиду тесной взаимосвязанности логической (HDL) и энергетической составляющих уже на поведенческом уровне описания системы очевидна необходимость введения специального расширения языка, позволяющего описать такую систему уже на одном уровне с RTL.

Язык (формат) описания UPF, решающий данную проблему, был впервые представлен в стандарте UPF 1.0 [11] организацией Accellera Systems Initiative в 2007 г. В 2009 г. организация IEEE дополнила формат UPF в стандарте IEEE 1801–2009 (UPF 2.0) [12]. Последней версией стандарта является IEEE 1801–2013 (UPF 2.1) [13].

Формат описания UPF – это специальный формат, предназначенный для поведенческого описания энергетической составляющей проекта электронной системы или ее компонентов. Формат предоставляет возможность описания сети питания проекта: цепей питания, энергетических доменов, переключателей, средств изоляции, средств сохранения состояния и других элементов, относящихся к энергетической составляющей проекта, непосредственно задаваемой разработчиком. UPF позволяет формально описать элементы подсистемы питания в их взаимосвязи с логическими элементами, описанными на языках проектирования VHDL и System Verilog. Важно отметить, что стандарт UPF изначально не предназначен для описания всей совокупности схемотехнических и электрических подробностей цепей питания, необходимых для аналоговой симуляции и верификации синтезированного проекта: UPF описывает структуру питания на поведенческом уровне. Однако UPF-описания в связке с HDL-описаниями могут быть использованы в качестве входных данных как при симуляции и верификации проекта [9, 14], так и при его синтезе (рис. 1) [13]. При синтезе проекта абстрактные модели объектов подсистемы питания отображаются на соответствующие физические ячейки.

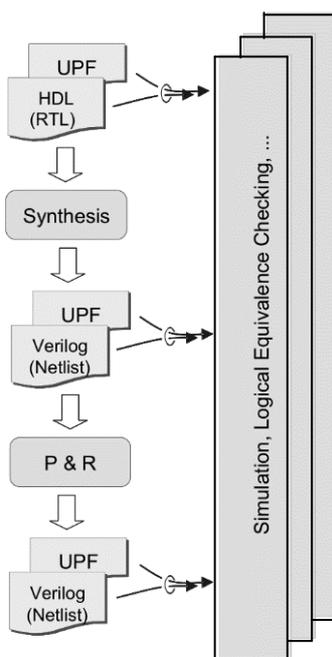


Рис. 1. Процесс проектирования системы с применением UPF-описаний

Стандарт UPF описывает четыре основные составляющие [10]:

1. *Команды описания энергетической составляющей (Commands)*. Команды языка UPF представляют собой расширение сценарного языка программирования Tcl [15]. Последовательность компактных команд-директив, записанная в файл UPF-описания с расширением .upf, поэтапно (инкрементально) описывает все элементы архитектуры энергетической составляющей проекта. Ключевой особенностью UPF-описаний является то, что данные описания никаким образом не изменяют HDL-описания логических элементов системы. Логические HDL-компоненты можно по-прежнему верифицировать отдельно от энергетической составляющей. С другой стороны, для одного и того же HDL-компонента или набора компонентов можно иметь несколько возможных UPF-описаний для различных случаев использования.

2. *Команды запроса информации о состоянии энергетической архитектуры проекта (Queries)*. Данные Tcl-команды, позволяющие облегчить процесс отладки UPF-описаний, генерируют детальные отчеты о состоянии UPF-элементов проекта.

3. *Пакет UPF (Package UPF)*. Пакет UPF, разработанный как для VHDL, так и для System Verilog, определяет типы и функции, необходимые для описания тестов, которые управляют как логическими портами, так и портами питания тестируемого модуля. Эти типы и функции также применяются для проектирования HDL-моделей, описывающих поведение энергетической составляющей проекта. Например, используя типы и функции из пакета UPF, можно описать поведенчески специфическую VHDL-модель переключателя питания, входными и выходными портами которого являются порты питания наравне с логическими портами.

Пакет UPF определяет два основных типа данных, позволяющих моделировать источники питания:

state := (OFF | UNDETERMINED | PARTIAL_ON | FULL_ON) – состояние источника питания;

supply_net_type := (state, voltage) – состояние (state) цепи питания в совокупности с уровнем напряжения (voltage).

Кроме того, пакет UPF определяет ряд функций для управления портами и цепями питания и контроля над ними.

Некоторые функции управления состоянием и напряжением порта (Supply Port) питания из теста (заголовки функций отображены в Си-подобном псевдокоде): supply_on(string port_name, real voltage), supply_off(string port_name).

Некоторые функции опроса состояния и напряжения порта питания:

get_supply_value(string port_name) – возвращает тип supply_net_type;

get_supply_state(supply_net_type arg) – возвращает тип state;

get_supply_voltage(supply_net_type arg) – возвращает тип voltage.

4. *Формат SAIF (Switching Activity Interchange Format)*. Предназначен для стандартизации представления информации об активности переключений элементов, генерируемой симулятором по окончании симуляции. Результат помещается в «обратный» backward-SAIF-файл. Этот файл представляет входные данные для программных средств анализа и оптимизации энергопотребления. Программа автоматизации проектирования может предварительно сгенерировать «прямой» forward-SAIF-файл, содержащий некоторые директивы, определяющие содержимое выходного backward-SAIF-файла. Важно отметить, что формат SAIF, являющийся частью стандарта IEEE 1801, служит лишь типовым шаблоном представления информации о переключательной активности, при этом в стандарте IEEE 1801 не указаны алгоритмы или правила для эффективного определения действительной переключательной активности: данная задача выходит за рамки стандарта IEEE 1801. Предполагается, что при разработке блока будет разработана тестовая входная последовательность, соответствующая типовым условиям функционирования этого блока; после окончания данной последовательности SAIF-совместимая САПР должна сгенерировать информацию о переключательной активности в виде файла формата SAIF. Эта информация позволит оценить динамическую мощность, рассеиваемую блоком,

с помощью любого SAIF-совместимого пакета программных средств анализа и оптимизации энергопотребления.

Важнейшей особенностью процесса симуляции проекта с UPF-описаниями является то, что все процессы HDL-логики домена с выключенным основным питанием особым образом завершают процессы симуляции: содержимое регистров становится неопределенным, а все внутренние и выходные сигналы по цепочке переходят в неопределенное состояние (X).

На сегодняшний день язык UPF поддерживается во всех ведущих системах автоматизированного проектирования и симуляции от таких компаний, как Mentor Graphics, Synopsys, Cadence Design Systems и др. Однако на данный момент UPF не является единственным поддерживаемым языком описания энергетически эффективных систем. Конкуренцию UPF составляет стандарт Common Power Format [16], который во многом совпадает с UPF, но имеет от него и некоторые отличия [3, 17].

4. Группы команд UPF

Стандарт IEEE 1801–2013 [13] содержит исчерпывающее описание команд UPF с примерами их использования, однако для лучшего понимания возможностей языка UPF целесообразно проанализировать данные команды и сгруппировать их по смысловому назначению. Так, команды описания энергетической составляющей проекта (Commands) UPF можно условно разделить по функциональному назначению на несколько групп (рис. 2) [10].

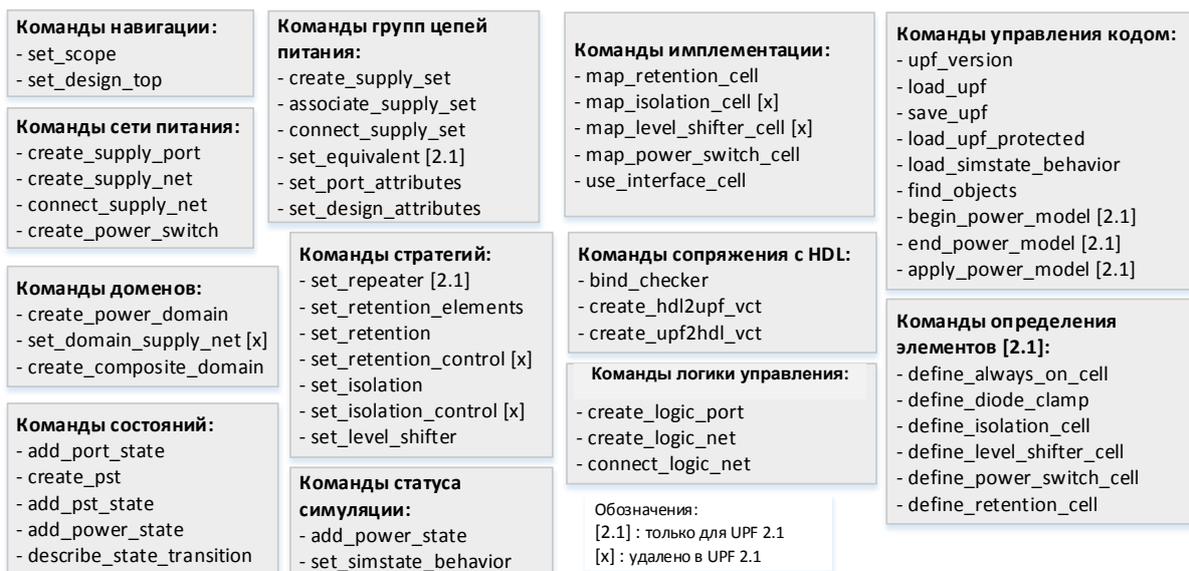


Рис. 2. Группы команд UPF

Команды навигации определяют, к какой части HDL-иерархии проекта будут относиться последующие UPF-команды.

Команды доменов используются для сегментирования проекта на отдельные участки питания, определяемые совокупностью логических компонентов (экземпляров RTL-компонентов) и совокупностью основного и специальных источников питания.

Команды описания разрешенных состояний энергопотребления используются для автоматизированной верификации энергетической системы.

Команды описания стратегий определяют правила размещения и управления элементами изоляции и элементами изменения уровня напряжения сигналов на границе домена питания, правила размещения и управления элементами сохранения состояния в пределах домена питания.

Команды имплементации определяют, каким образом абстрактные поведенческие элементы подсистемы питания отображаются на заданные библиотечные ячейки элементов питания в процессе автоматического синтеза проекта.

Команды сопряжения с HDL позволяют добавлять в проект UPF-осведомленные проверочные утверждения. Эти команды предназначены также для операций преобразования типов портов питания в логические типы HDL и наоборот.

Команды статуса симуляции (Simstate-команды) добавлены в UPF 2.0 для расширения возможностей по детальному определению статуса объектов симуляции (добавлены частично рабочие состояния доменов и сетей питания).

Команды групп и атрибутов цепей питания (Supply-Sets-команды) добавлены в UPF 2.0 для оптимизации описания и подключения сетей питания.

Команды описания логики управления добавлены в UPF 2.0 для некоторых ситуаций моделирования, в которых удобнее описывать логические сигналы управления элементами энергетической составляющей проекта в пределах UPF-файлов вместо HDL-файлов (команды не предназначены для полной замены HDL-описаний).

Важно отметить, что часть команд, представленных или дополненных в UPF 2.0, замещает по функциональности некоторые устаревшие команды из подмножества команд, представленных в UPF 1.0. В UPF 2.1 данные устаревшие команды полностью исключаются, в связи с чем обратная совместимость с предыдущими версиями невозможна. Ввиду новизны стандарта UPF 2.1 актуальной версией можно считать UPF 2.0 [10], однако использовать команды UPF 2.0, исключенные из UPF 2.1, не рекомендуется.

5. Пример разработки проекта с применением технологии UPF

Для того чтобы показать принципы проектирования энергетически эффективной системы с применением UPF-описаний, был разработан демонстрационный проект. Файлы RTL-описаний (System-Verilog-файлы `upf_demo.sv` и `tb.sv`) и UPF-описания (`upf_demo.upf`), а также файл сценария запуска `run.do` для среды Mentor Graphics QuestaSim 10.2 [18] доступны по ссылке [19]. Схемы логического проекта показаны на рис. 3. При разработке проекта использовалась методика, представленная в разд. 3. Моделирование проекта с учетом энергетической составляющей осуществлено на поведенческом уровне.

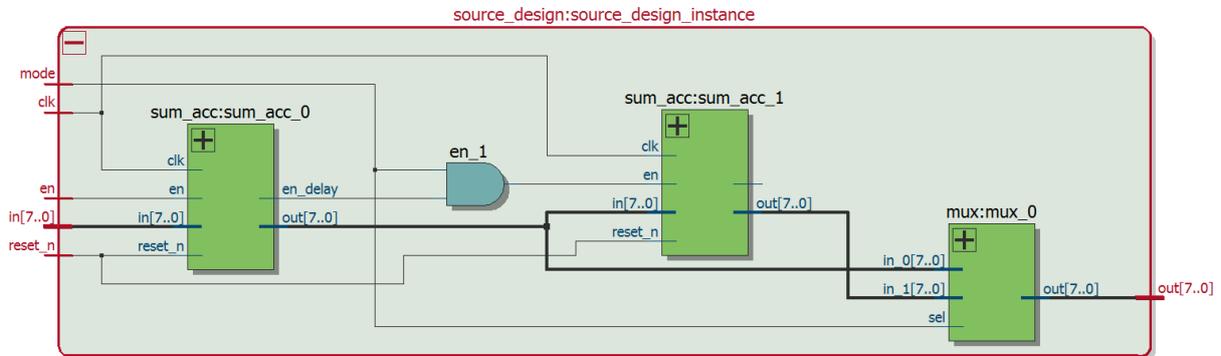
Под регистром здесь и далее понимается элемент памяти произвольной разрядности.

Основой энергетически эффективной электронной системы является иерархия ее логической части, которая некоторым оптимальным образом распределяется в отдельные домены с различными стратегиями питания. Логическая структура исходного проекта (без энергетической составляющей – системы питания) представлена модулем `source_design` (экземпляр `source_design_instance`) и вложенными в него компонентами (рис. 3, *a*). Основным логическим компонентом демонстрационного проекта выбран модуль `sum_acc`, который по фронту сигнала синхронизации при `en = 1` прибавляет входные данные порта `in` к текущему содержимому внутреннего регистра `acc[7:0]`. Содержимое регистра `acc` выводится в порт `out[7:0]`. Вывод `en_delay` является задержанным на один такт значением входного порта `en` (внутренний регистр `en_d`). Экземпляры модуля `sum_acc` `sum_acc_0` и `sum_acc_1` поставлены в каскад, где выход `sum_acc_0.out` подключен к входному порту `in` экземпляра `sum_acc_1`, а выходной порт `sum_acc_0.en_delay` подключен (через элемент И) к входному порту `en` экземпляра `sum_acc_1`. Мультиплексор `mux_0` по сигналу от порта `source_design_instance.mode` выбирает (через порт `mux_0.sel`), выход `out` какого из экземпляров `sum_acc` будет направлен на общий вывод `source_design_instance.out[7:0]`.

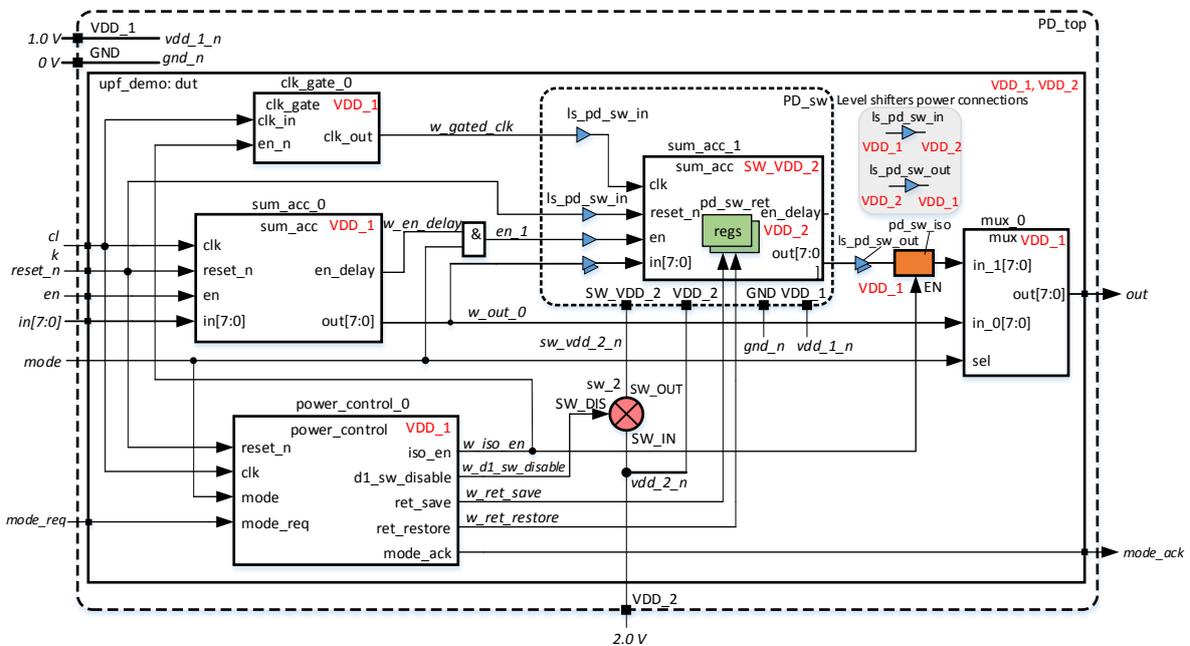
Таким образом, при `source_design_instance.mode = 1` и после изменения регистра `acc` экземпляра `sum_acc_0` в следующем такте ввиду `w_en_delay = 1` значение регистра `acc` экземпляра `sum_acc_1` будет увеличено на очередное значение регистра `sum_acc_0.acc`, а новое значение `sum_acc_1.acc` будет подано на вывод `source_design_instance.out`.

В режиме `source_design_instance.mode = 0`, при котором на вывод `source_design_instance.out` передается значение `sum_acc_0.out`, элемент И, к которому подключен

sum_acc_0.en_delay, обеспечивает сохранение состояния регистра sum_acc_1.acc. Таким образом, в режиме mode = 0 экземпляр sum_acc_1 полностью не задействован, а значит, sum_acc_1 является частным случаем компонента, который может быть отключен от питания в одном из режимов работы (при условии сохранения содержимого sum_acc_1.acc).



а)



б)

Рис. 3. Логические схемы: а) без энергетической составляющей; б) с учетом системы питания

После получения основной логической части RTL-описания проекта, представленной в модуле source_design на рис. 3, а, и соблюдения дополнительных условий его функционирования проект можно разделить на следующие домены (рис. 3, б): выключаемый домен PD_sw, содержащий экземпляр sum_acc_1, и невыключаемый домен PD_top, содержащий все остальные логические компоненты source_design. В демонстрационных целях (для обеспечения необходимости установки блоков изменения уровня сигнала) домены питания проекта с энергетической составляющей используют различные источники напряжения питания: 1.0 V для PD_top (порт питания VDD_1) и 2.0 V для PD_sw (порт питания VDD_2). Для PD_sw используется переключатель питания (ключ) sw_2 с выходным портом питания SW_OUT. Входной порт ключа SW_IN питается от напряжения 2.0 V (VDD_2). SW_OUT является основным источником питания для PD_sw, а VDD_1 – основным источником питания для PD_top. Оба домена используют общий порт земли GND.

После выделения доменов питания системы необходимо составить таблицу всех возможных одновременных состояний доменов системы, выраженных через состояния портов питания. По таблице можно выделить полный набор необходимых элементов обеспечения энергетической эффективности: блоки изоляции, сохранения, изменения уровня напряжения сигналов, блоки отключения синхросигнала. Выключение цепей питания в таблице также сигнализирует о необходимости постановки переключателей питания.

Состояние питания	VDD_1 (PD_top)	VDD_2	sw_2/SW_OUT (PD_sw)	GND
FULL_ON	ON 1.0 V	ON 2.0 V	ON 2.0 V	ON 0 V
PART_ON	ON 1.0 V	ON 2.0 V	OFF	ON 0 V
FULL_OFF	OFF	OFF	OFF	ON 0 V

В режиме питания FULL_ON домены PD_top и PD_sw питаются от различных источников напряжения питания. Для правильной интерпретации логических сигналов, связывающих эти домены (ON 1.0 V \rightarrow ON 2.0 V и ON 2.0 V \rightarrow ON 1.0 V), необходимо ввести блоки изменения уровня напряжения сигнала как для входных, так и для выходных сигналов PD_sw (ls_pd_sw_in и ls_pd_sw_out на рис. 3, б). Входы и выходы модулей изменения уровня сигнала питаются от источников питания VDD_1 и VDD_2 в соответствии с направлением изменения уровня напряжения сигналов.

В режиме питания PART_ON домен PD_sw находится в отключенном состоянии (OFF). Согласно описанным выше требованиям состояние всех внутренних регистров модуля sum_acc_1 должно сохраняться, поэтому в домен PD_sw необходимо добавить регистры сохранения для sum_acc_1.acc и sum_acc_1.en_d (pd_sw_ret на рис. 3, б). Регистры сохранения питаются от включенного порта питания VDD_2.

В режиме питания PART_ON сигналы домена PD_top передаются в отключенный домен PD_sw. Во избежание негативных эффектов от такого подключения выходной порт sum_acc_1.out подключается к блоку изоляции (сигнал от порта sum_acc_1.out принимает значение, которое предшествовало включению изоляции). Выходной порт sum_acc_1.en_delay является неподключенным, а значит, изоляция для него не требуется. Блоки изоляции питаются от порта VDD_1.

Ввиду незадействованности экземпляра sum_acc_1 в режиме PART_ON сигнал синхронизации clk для этого компонента отключается с помощью блока отключения clk_gate_0 (RTL-схема отключения пропускания синхросигнала с защелкой).

После того как для проекта определены все переключатели питания, элементы изоляции и блоки сохранения, а также блоки отключения сигнала синхронизации, необходимо описать логику управления этими элементами посредством RTL. Модуль power_control_0, показанный на рис. 3, а, является модулем управления элементами питания для всего upf_demo:dut. По сигналу mode_req конечный автомат компонента power_control_0 начинает в соответствии с таблицей изменять состояние сигнала выключения переключателя питания w_d1_sw_disable, включения изоляции w_iso_en и сигналов w_ret_save и w_ret_restore для сохранения и восстановления данных в блоках сохранения. В процессе выключения домена PD_sw модуль управления питанием сначала включает изоляцию, затем сохраняет данные в регистры сохранения и только потом выключает sw_2. По завершении процесса смены режима питания модуль сигнализирует dut.mode_ack = 1. Процесс включения PD_sw идет в обратном порядке с учетом восстановления состояний регистров sum_acc_1 из регистров сохранения. Для упрощения схемы в качестве сигнала выключения синхросигнала w_gated_clk выбран сигнал w_iso_en. Важно отметить, что для реальных проектов процессы включения и выключения домена могут занимать длительное время ввиду инертности реальных физических компонентов системы питания (в демонстрационном проекте это не показано).

```

# Определение главного модуля:
set_design_top upf_demo
set_scope .

# Создание доменов PD_top: все элементы главного
модуля
# включаются PD_top:
create_power_domain PD_top \
  -include_scope
# Создание домена PD_sw,
# перенос sum_acc_1 в домен PD_sw:
create_power_domain PD_sw \
  -elements {sum_acc_1}

# Создание портов питания:
create_supply_port VDD_1
create_supply_port VDD_2
create_supply_port GND

# Создание цепей питания:
create_supply_net vdd_1_n \
  -domain PD_top
create_supply_net vdd_2_n \
  -domain PD_top
create_supply_net gnd_n \
  -domain PD_top

# Подключение цепей питания к портам:
connect_supply_net vdd_1_n \
  -ports VDD_1
connect_supply_net vdd_2_n \
  -ports VDD_2
connect_supply_net gnd_n \
  -ports GND

# Группировка цепей питания в набор pwr_1_ss:
create_supply_set pwr_1_ss \
  -function {power vdd_1_n} \
  -function {ground gnd_n}
# Назначение группы pwr_1_ss в
# качестве основного набора питания для PD_top:
associate_supply_set pwr_1_ss \
  -handle PD_top.primary

# Группировка цепей питания в набор pwr_2_ss:
create_supply_set pwr_2_ss \
  -function {power vdd_2_n} \
  -function {ground gnd_n}

# Создание группы цепей питания
# для подключения к PD_sw:
create_supply_net sw_vdd_2_n \
  -domain PD_top
create_supply_set sw_pwr_2_ss \
  -function {power sw_vdd_2_n} \
  -function {ground gnd_n}
# Назначение группы sw_pwr_2_ss в качестве
# основного набора питания для PD_sw:
associate_supply_set sw_pwr_2_ss \
  -handle PD_sw.primary

# Создание модели ключа sw_2, управляемого
# RTL-сигналом w_d1_sw_disable:
create_power_switch sw_2 \
  -domain PD_sw \
  -input_supply_port \
    {SW_IN pwr_2_ss.power} \
  -output_supply_port \
    {SW_OUT sw_pwr_2_ss.power} \
  -control_port \
    {SW_DIS w_d1_sw_disable} \
  -on_state \
    {ON_STATE SW_IN {!SW_DIS}} \
  -off_state {OFF_STATE {SW_DIS}}

# Создание стратегии по автоматическому
# добавлению входных для PD_sw блоков изменения
# уровня напряжения сигналов:
set_level_shifter ls_pd_sw_in \
  -domain PD_sw \
  -applies_to inputs \
  -rule low_to_high \
  -location self

# Подключение питания к входным блокам изменения
# уровня напряжения сигналов:
associate_supply_set pwr_1_ss \
  -handle PD_sw.ls_pd_sw_in.input
associate_supply_set pwr_2_ss \
  -handle PD_sw.ls_pd_sw_in.output

# Создание стратегии по автоматическому
# добавлению выходных для PD_sw блоков изменения
# уровня напряжения сигналов:
set_level_shifter ls_pd_sw_out \
  -domain PD_sw \
  -applies_to outputs \
  -rule high_to_low \
  -location parent

# Подключение питания к выходным блокам изменения
# уровня напряжения сигналов:
associate_supply_set pwr_2_ss \
  -handle PD_sw.ls_pd_sw_out.input
associate_supply_set pwr_1_ss \
  -handle PD_sw.ls_pd_sw_out.output

# Создание стратегии по добавлению блоков изоляции
# с точным указанием изолируемых элементов:
set_isolation pd_sw_iso \
  -domain PD_sw \
  -clamp_value latch \
  -isolation_signal w_iso_en \
  -isolation_sense high \
  -location parent \
  -elements {sum_acc_1/out}
# Подключение питания к блокам изоляции:
associate_supply_set pwr_1_ss \
  -handle PD_sw.default_isolation

# Создание стратегии по автоматическому добавлению
# блоков сохранения для всех регистров
# домена PD_sw (управление RTL-сигналами
# w_ret_save и w_ret_restore):
set_retention pd_sw_ret \
  -domain PD_sw \
  -save_signal {w_ret_save posedge} \
  -restore_signal \
    {w_ret_restore posedge}
# Подключение питания к регистрам сохранения:
associate_supply_set pwr_2_ss \
  -handle PD_sw.default_retention

# Определение состояний портов питания
# (согласно таблице 1):
add_port_state VDD_1 \
  -state {ON_1 1.0} \
  -state {OFF_ST OFF}
add_port_state VDD_2 \
  -state {ON_2 2.0} \
  -state {OFF_ST OFF}
add_port_state sw_2/SW_OUT \
  -state {ON_2 2.0} \
  -state {OFF_ST OFF}
add_port_state GND \
  -state {ON_0 0}
# Описание таблицы 1 в формате UPF (таблица
# используется для верификации работы блока
# управления питанием):
create_pst DEMO_PST \
  -supplies {VDD_1 VDD_2 sw_2/SW_OUT GND}
add_pst_state FULL_ON -pst DEMO_PST \
  -state {ON_1 ON_2 ON_2 ON_0}
add_pst_state PART_ON -pst DEMO_PST \
  -state {ON_1 ON_2 OFF_ST ON_0}
add_pst_state FULL_OFF -pst DEMO_PST \
  -state {OFF_ST OFF_ST OFF_ST ON_0}

```

Рис. 4. Описание энергетической составляющей проекта. Файл upf_demo.upf

Следующим шагом является формальное поведенческое описание полученной выше информации об энергетической составляющей на языке UPF. Результат поведенческого проектирования (описания) доменной системы для демонстрационного проекта представлен в файле `upf_demo.upf` [19] (рис. 4). Важно отметить, что в RTL-описании модуля `upf_demo:dut` (файл `upf_demo.sv`) сигналы `w_d1_sw_disable`, `w_iso_en`, `w_ret_save` и `w_ret_restore`, выходящие из модуля `power_control_0`, являются неподключенными, но они явно подключаются к соответствующим элементам обеспечения энергетической эффективности уже в файле `upf_demo.upf`.

Моделирование демонстрационного проекта было осуществлено на поведенческом уровне в среде Mentor Graphics QuestaSim 10.2 в режиме Power Aware Simulation [20]. Временная диаграмма функционирования проекта показана на рис. 5. В ходе теста (System-Verilog-файл `tb.sv` [19]) система отработала в составе полного каскада (все домены включены). Затем был послан запрос на выключение домена PD_sw (сигнал `mode = 0 → 1`, `mode_req = 1`), после чего система отработала с отключенным от питания модулем `sum_acc_1`. Регистры `sum_acc_1.acc` и `sum_acc_1.en_d` также были отключены от питания (на диаграмме это показано диагональной решеткой). Перед выключением питания состояния этих регистров были сохранены по сигналу `w_ret_save`, а после включения питания были восстановлены по сигналу `w_ret_restore`. Сплошная прозрачная заливка для сигнала от порта `sum_acc_1.out` на диаграмме показывает включение изоляции: вывод `sum_acc_1.out` изолируется от всех изменений регистра `sum_acc_1.acc` в процессе включения и выключения питания.

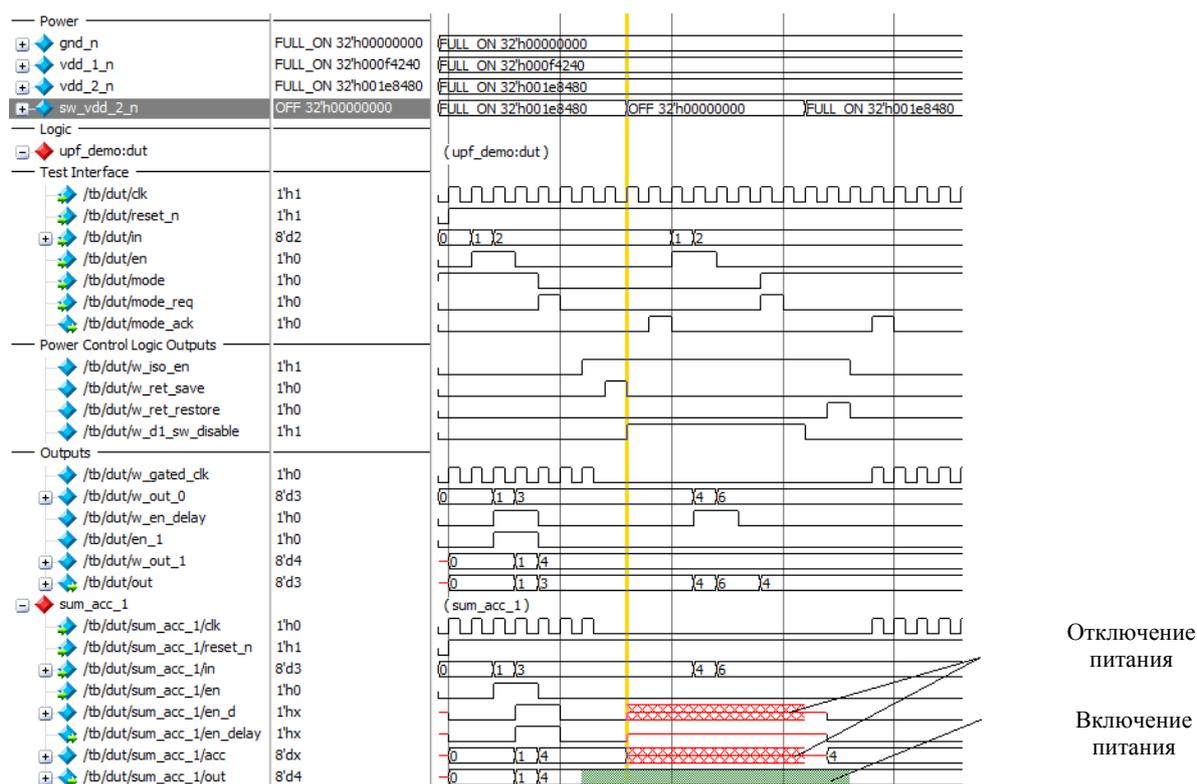


Рис. 5. Временная диаграмма симуляции проекта

Основная логика демонстрационного проекта была спроектирована таким образом, чтобы обусловить возможность разделения проекта на домены питания, причем один из доменов питания сделан отключаемым. Было осуществлено разделение системы на конкретные домены питания с различными напряжениями питания, определены все возможные состояния этих доменов через сети питания. Из таблицы состояний получен список всех переключателей питания, блоков изоляции, сохранения и изменения уровня сигнала. После этого было разработано формальное поведен-

ческое описание энергетической подсистемы проекта в виде UPF-файла. Временная диаграмма, полученная в среде симуляции, показывает специфику изменений сигналов при переключениях энергетических состояний проекта.

Дальнейшая разработка проекта предполагает синтез RTL-логики и отображение абстрактных моделей переключателей питания, изоляции, регистров сохранения и изменения уровня на реальные физические библиотечные ячейки (UPF-команды имплементации “map_”). Важно отметить, что оценить эффект от разбиения системы на домены с различными источниками питания, часть из которых является отключаемой, можно только после синтеза проекта с использованием конкретных библиотек логических элементов. Действительно, для рассмотренного демонстрационного проекта энергия токов утечки, сохраненная во время отключения домена PD_sw, для какого-то из результатов синтеза может быть меньше, чем энергия, затраченная на постоянную работу модуля управления power_control и других элементов обеспечения энергетической эффективности, которые были добавлены в систему для управления выключаемым доменом PD_sw. Однако, как правило, в реальных системах отключаемые домены питания содержат гораздо больше компонентов логики, в то время как затраты на элементы поддержки и управления отключаемыми доменами остаются незначительными [2]. Общая же методология проектирования для реальных проектов будет соответствовать методологии, показанной в данном демонстрационном проекте.

Заключение

Одной из самых эффективных мер снижения энергопотребления для систем на кристалле, производимых по современным технологическим процессам, является проектирование таких систем с активной доменной системой питания. Необходимы дополнительные средства поведенческого описания доменной подсистемы питания ввиду недостаточности возможностей языков VHDL и System Verilog. В качестве средства описания подсистемы питания был рассмотрен формат UPF, приведен пример описания системы с применением средств этого формата.

На примере схемы с отдельными доменами питания было показано, как исходное RTL-описание логической структуры проекта можно дополнить элементами доменной системы питания, используя средства формата UPF. Моделирование демонстрационного проекта было осуществлено в UPF-совместимой САПР с учетом работы компонентов системы питания.

Формат UPF помогает описывать системы с активной доменной системой питания, однако выделение доменов питания и управление ими в системе является сложной научной и инженерной задачей, лежащей на плечах разработчиков [2]. Так, для современных технологических процессов производства микросхем многие существующие сложные вычислительные блоки могут быть разделены на отдельные домены питания, что потребует тщательного пересмотра структуры таких компонентов для максимального энергосбережения. В связи с этим формат UPF является инструментом, позволяющим значительно облегчить процесс проектирования новейшего класса цифровых систем – систем с активной доменной системой питания.

Список литературы

1. A Practical Guide to Low-Power Design. User Experience with CPF [Electronic resource]. – 2009. – Mode of access : http://www.si2.org/events_dir/2009/PowerForward/LowPowerGuide09232009/pfi_lpg_chapters/lpg_sect1_06052009.pdf. – Date of access : 19.03.2015.
2. Low Power Methodology Manual: For System-on-Chip Design / D. Flynn [et al.]. – N.Y. : Springer, 2011. – 300 p.
3. Chadha, R. An ASIC Low Power Primer: Analysis, Techniques and Specification / R. Chadha, J. Bhasker. – N.Y. : Springer, 2015. – 232 p.

4. Rabaey, J. Low Power Design Essentials / J. Rabaey. – N.Y. : Springer, 2009. – 288 p.
5. Рабаи, Ж.М. Цифровые интегральные схемы / Ж.М. Рабаи, А. Чандракасан, Б. Николич. – 2-е изд. – М. : Вильямс, 2007. – 912 с.
6. Power-efficient System Design / P.R. Panda [et al.]. – N.Y. : Springer, 2011. – 253 p.
7. IEEE Standard VHDL Language Reference Manual. IEEE 1076 – 2002 [Electronic resource]. – 2002. – Mode of access : <http://ieeexplore.ieee.org/servlet/opac?punumber=7863>. – Date of access : 19.03.2015.
8. IEEE Standard for SystemVerilog: Unified Hardware Design, Specification, and Verification Language. IEEE 1800 – 2012 [Electronic resource]. – 2012. – Mode of access : <http://standards.ieee.org/findstds/standard/1800-2012.html>. – Date of access : 19.03.2015.
9. Low Power Verification Methodology Using UPF / F. Bembaron [et al.] // DVCon 2011 [Electronic resource]. – 2011. – Mode of access : http://2011.dvcon.org/file/11.3_Paper.pdf. – Date of access : 19.03.2015.
10. Marschner, E. Power Aware Verification / E. Marschner, C. Seeley // Mentor Graphics Verification Academy [Electronic resource]. – 2013. – Mode of access : <https://verificationacademy.com/courses/power-aware-verification>. – Date of access : 19.03.2015.
11. Unified Power Format (UPF) Standard. Version 1.0 [Electronic resource]. – 2007. – Mode of access : <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=B47D579C08C9E70D33FF5F2A9DA15DEC?doi=10.1.1.133.6194&rep=rep1&type=pdf>. – Date of access : 19.03.2015.
12. IEEE Standard for Design and Verification of Low Power Integrated Circuits: IEEE Std 1801 – 2009 [Electronic resource]. – 2009. – Mode of access : <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?reload=true&punumber=4809843>. – Date of access : 19.03.2015.
13. IEEE Standard for Design and Verification of Low Power Integrated Circuits: IEEE Std 1801 – 2013 [Electronic resource]. – 2013. – Mode of access : <http://standards.ieee.org/getieee/1801/download/1801-2013.pdf>. – Date of access : 19.03.2015.
14. Verification Methodology Manual for Low Power / S. Jadcherla [et al.]. – N. Y. : Springer, 2011. – 226 p.
15. Tcl/Tk Documentation // Tcl Developer Xchange [Electronic resource]. – 2015. – Режим доступа : <http://www.tcl.tk/doc/>. – Дата доступа : 19.03.2015.
16. Si2 Common Power Format (CPF) // Si2 [Electronic resource]. – 2009. – Mode of access : http://www.si2.org/openeda.si2.org/project/showfiles.php?group_id=51. – Date of access : 19.03.2015.
17. Power Intent Formats: Light at the End of the Tunnel? / S. Dobre [et al.] // EETimes [Electronic resource]. – 2012. – Mode of access : http://www.eetimes.com/document.asp?doc_id=1279413. – Date of access : 19.03.2015.
18. Questa Advanced Simulator [Electronic resource]. – 2015. – Mode of access : <http://www.mentor.com/products/fv/questa/>. – Date of access : 19.03.2015.
19. UPF Demo Project [Electronic resource]. – 2015. – Mode of access : https://www.dropbox.com/s/ogty44li1w67pxz/upf_demo.zip?dl=0. – Date of access : 19.03.2015.
20. Power Aware Simulation User's Manual. Questa SIM. Software Version 10.2c [Electronic resource]. – 2013. – Mode of access : https://www.dropbox.com/s/r43dfuqkgada6r9/questa_sim_pa.pdf?dl=0. – Date of access : 19.03.2015.

Поступила 09.06.2015

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: alexey.shashkov@gmail.com*

A.S. Shashkov

**DESIGN OF LOW-POWER ELECTRONIC SYSTEMS USING
UPF POWER INTENT SPECIFICATION TECHNOLOGY**

In this review paper, the problem of the design of energy-efficient electronic systems is considered. Different energy-saving methods are presented. The necessity of additional tools of formal behavioral description of the power subsystem is explained. The elements of the power domain energy subsystem are explained, the design stages for the system with power domains are presented. Unified Power Format (UPF) is presented as a tool to describe power intent in a low-power system. An example system with the UPF-description is presented.