

Метод статического временного анализа с учетом трассировочных ресурсов для схем на базе реконфигурируемых систем на кристалле

С. В. Гаврилов, В. М. Хватов, Д. А. Железников, Т. В. Гарбулина

Институт проблем проектирования в микроэлектронике Российской академии наук, г. Москва

s.g@ippm.ru, khvatov_v@ippm.ru, zheleznikov_d@ippm.ru, garbulina_t@ippm.ru

Аннотация — Существующие средства оценки быстродействия цифровых схем, проектируемых в базе ПЛИС и РСнК, используют либо полное моделирование полученной после трассировки схемы, либо заранее рассчитанные задержки межсоединений в совокупности с задержками логических элементов.

Полное моделирование схемы подразумевает использование Spice-моделей транзисторов и требует больших затрат машинного времени, что сильно зависит от объемов схемы и количества использованных коммутационных ресурсов. Расчет задержек всех путей, по которым может пройти сигнал в конкретной схеме, вследствие большого количества возможных вариантов коммутации цепей также требует больших временных затрат. Следовательно, требуется быстрая оценка быстродействия схемы с динамическим расчетом задержек цепей еще до этапа временной верификации с помощью Spice.

В статье показана необходимость быстрой оценки производительности проектируемых схем по окончании различных этапов проектирования: логического синтеза в базе элементов системы на кристалле, а также после размещения элементов и трассировки межсоединений. В работе представлен поэтапный процесс внедрения оценки быстродействия в общий автоматизированный маршрут проектирования и описан разработанный маршрут быстрой оценки быстродействия после трассировки.

Данный маршрут использует программное обеспечение, выполняющее статический временной анализ с использованием библиотеки характеризованных логических элементов в формате Liberty и емкостных характеристик цепей, полученных в результате экстракции паразитных элементов топологии.

Ключевые слова — ПЛИС (программируемая логическая интегральная схема), РСнК (реконфигурируемая система-на-кристалле), САПР, статический временной анализ, трассировка, оценка быстродействия.

I. ВВЕДЕНИЕ

Одним из основных этапов маршрута проектирования интегральных схем (ИС) на базе ПЛИС и РСнК (схемы, объединяющие на одном кристалле сложно-функциональные блоки, аналоговые и цифровые компоненты и матрицы программируемой

логики) является временная верификация с целью оценки быстродействия разрабатываемого устройства.

Временная верификация может быть выполнена с помощью двух видов анализа – временного моделирования и статического временного анализа (СВА) (также известный под аббревиатурой STA – Static Timing Analysis). Временное моделирование представлено двумя видами – на транзисторном уровне (SPICE-моделирование) и на уровне логических элементов (Verilog-моделирование). Для Verilog-моделирования используется библиотека логических элементов на языке Verilog и файлы задержек в формате SDF (Standard Delay Format) [1]. Для SPICE-моделирования используется только схемотехническое описание схемы на языке Spice. Оно наиболее точное, но требует больших объемов вычислительных ресурсов и в некоторых случаях может быть невыполнимо вследствие огромных затрат машинного времени.

СВА не выполняет логическое моделирование схемы и не рассматривает логические функции её составляющих. На базе библиотек элементов в формате Liberty, в СВА оцениваются задержки всех возможных путей распространения сигналов в схеме и на их основе выполняется поиск временных нарушений. Использование библиотек с заранее рассчитанными задержками является основным преимуществом СВА перед полным моделированием, так как такой анализ позволяет очень быстро оценить быстродействие схемы и тем самым - её работоспособность.

При проектировании схем на базе ПЛИС или РСнК необходимо выполнять процедуру СВА несколько раз. Во-первых, в начале проектирования – после этапа логического синтеза. Это позволяет оценить быстродействие результатов логического синтеза и более качественно провести размещение схемы на кристалле, учитывая разветвление по выходу для каждого логического элемента. Во-вторых, по окончании процесса проектирования – после завершения этапов размещения и трассировки. Это позволяет определить общую работоспособность схемы и её максимальную рабочую частоту с учетом межсоединений.

На данный момент в открытых источниках отсутствует информация о технической реализации оценки быстродействия схем с учетом межсоединений

и трассировочных ресурсов. Методы проведения такой оценки в коммерческих САПР для ПЛИС и РСнК недоступны пользователю, так как являются интеллектуальной собственностью и коммерческой тайной [2]. Общедоступные методы не рассчитывают задержки элементов и межсоединений между ними. Они рассчитывают только результирующие задержки путей, основываясь на данных, представленных пользователем [3, 4]. Предполагается, что у пользователя есть информация о временных ограничениях и он учтет задержку физических межсоединений и коммутационных элементов при определении задержки логического элемента. Но единовременный статический расчет задержек всех межсоединений не применим к реальным ПЛИС и РСнК вследствие большого количества возможных вариантов коммутации цепей. Следовательно, требуется рассчитывать задержки соединений динамически, после проведения процедуры трассировки.

Дальнейшее содержание работы включает следующие разделы:

- раздел II, в котором представлен разработанный метод оценки быстродействия ИС и поэтапный процесс внедрения данного метода в общий маршрут проектирования, описаны основные принципы и понятия, используемые при статическом временном анализе. Показаны особенности выполнения СВА до и после выполнения физического синтеза.
- раздел III, содержащий численные результаты оценки быстродействия схем.

II. РАЗРАБОТАННЫЙ МЕТОД ОЦЕНКИ БЫСТРОДЕЙСТВИЯ СХЕМ ПОСЛЕ ТРАССИРОВКИ

В данной работе предложен метод оценки быстродействия цифровых схем после трассировки, который может быть интегрирован в общий автоматизированный маршрут проектирования ИС на ПЛИС и РСнК (рис 1).

Он включает в себя логический синтез, физический синтез, состоящий из планировки, размещения и трассировки межсоединений, а также проведение верификации результатов каждого этапа [4].

Метод позволяет проводить оценку быстродействия схем на базе ПЛИС и РСнК с любой архитектурой и заменять используемое в данной работе программное обеспечение (ПО) на любое другое с соответствующим функционалом.

Для проведения этапа логического синтеза в данной работе было использовано открытое ПО Yosys с программным модулем ABC, являющимся его составной частью [6]. Для выполнения размещения и трассировки был использован программный модуль XCAD, разработанный в ИППМ РАН. СВА выполняется программой STAT, также разработанной в ИППМ РАН. В качестве альтернативы для выполнения СВА в предложенном маршруте можно использовать

как открытое ПО – например, OpenTimer [7], так и коммерческое – например, Synopsys PrimeTime или Synopsys Design Compiler (функция получения временных отчетов без логического синтеза).

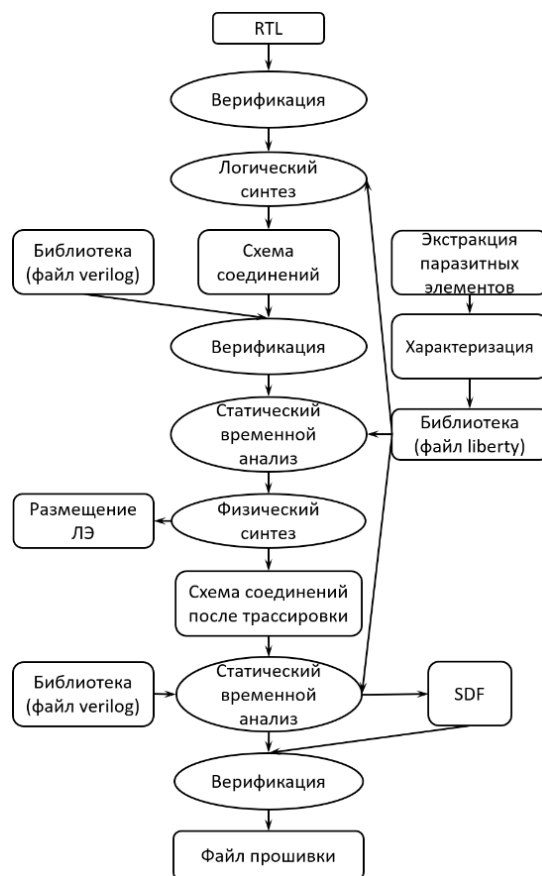


Рис. 1. Маршрут проектирования схем на базе ПЛИС и РСнК

Разработанный метод включает в себя следующий ряд действий:

A. Экстракция паразитных элементов.

Данный процесс представляет собой расчет емкостей и сопротивлений, образованных между проводниками и элементами схемы в разработанной топологии ПЛИС или РСнК [8]. Экстракция, как правило, выполняется с помощью коммерческих САПР, таких как Calibre от Mentor Graphics или StarRC от Synopsys.

B. Характеризация библиотек

Характеризацией называется поиск зависимостей выходной задержки, длительности фронта выходного сигнала и энергии от интервала длительности входного фронта и интервала нагружающих выход емкостей [9].

Для повышения точности расчетов и для их приближения к реальным значениям характеристика проводится с использованием списка соединений с результатами экстракции. Перед характеристикой определяется максимальная емкость (на основе данных, полученных после процедуры экстракции), нагружающая выходы каждого из типов используемых

элементов. Её значение задает верхний предел интервала емкостей, на основании которых будет выполняться характеристика. Результатом характеристики является набор библиотек в формате Liberty.

Типы элементов, которые используются в дальнейшем маршруте и требуют выполнения характеристики, следующие:

- *Логический Элемент (ЛЭ)* – элемент, выполняющий логическую функцию разной сложности, реализуемый в ПЛИС и РСнК в виде конфигурируемого логического блока или LUT-элемента.
- *Ячейка ввода-вывода (ЯВВ)* – буферный элемент, соединяющий периферию с обрабатывающими компонентами ПЛИС и РСнК.
- *Трассировочный элемент (ТЭ)* – элемент, конфигурируемый в процессе трассировки, позволяющий выполнять маршрутизацию сигналов между ЛЭ. В данной работе для проведения СВА после трассировки требуется характеристика таких типов ТЭ, как буферы (рис. 2а), инверторы (рис. 2б) и мультиплексоры (рис. 2в).
- *Сложно-функциональные блоки (СФ-блоки)* – блоки, специально разработанные для выполнения ряда конкретных сложных функций и занимающие фиксированное положение на схеме. Примерами таких блоков являются многоадресные умножители, блоки памяти, блоки фазовой автоподстройки частоты и др.

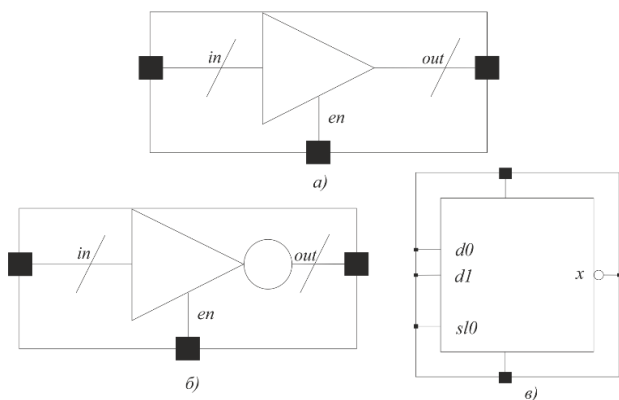


Рис. 2. Трассировочные элементы, требующие характеристики

С. Формирование Verilog-библиотек

Для последующего временного моделирования заранее формируются библиотеки с описанием всех используемых типов элементов на языке Verilog. Библиотеки могут быть сгенерированы с помощью ПО, выполняющего характеристику, или разработаны вручную [10].

Д. Загрузка паразитных элементов в САПР

Паразитные элементы, полученные в ходе экстракции, преобразуются в формат, необходимый программе, выполняющей размещение и трассировку.

В рассматриваемом маршруте с помощью Mentor Graphics Calibre генерируется файл с полными емкостями цепей, который далее преобразуется в синтаксические конструкции языка Tcl.

Пример емкостей, подготовленных к загрузке в программу, представлен ниже:

```
set_load 0.249229 { ILAB0101_net39756 }
set_load 0.098835 { ILAB0101_net27710 }
set_load 0.274796 { ILAB0201_net39105 }
set_load 0.246603 { ILAB0201_net37974 }
```

Полученные емкости загружаются в САПР и добавляются в трассировочный граф. В коммутационном графе загруженная емкость преобразуется в специальный весовой коэффициент для последующего выполнения процедуры трассировки с помощью алгоритма PathFinder [11].

Е. Расчет эквивалентных емкостей

Для проведения СВА необходимы список соединений и библиотеки характеризованных элементов, включая ЛЭ, ЯВВ и ТЭ. В ПЛИС и РСнК к трассировочным элементам относятся как буферы, инверторы и мультиплексоры, так и проходные ключи, реализованные с помощью МОП-транзисторов. Так как такой ключ является аналоговым элементом, к нему не могут быть применены методы характеристики цифровых схем и модели описания их задержек, в том числе NLDM (Non-Linear Delay Model – нелинейная модель задержки). Следовательно, данный тип элементов не может быть добавлен в характеризованные библиотеки.

Для решения этой проблемы в данной работе предложен метод построения математической модели эквивалентной емкости коммутационных соединений ПЛИС и РСнК без прямого использования проходных ключей.

В данном методе после завершения процесса трассировки коммутационный граф обходится от приемников сигнала к его источнику. Обход пути каждой из проектных цепей завершается после достижения выхода ЛЭ, ЯВВ или буферизирующего ТЭ. Все паразитные сопротивления игнорируются, а проходные ключи и их емкости, встречающиеся на пути распространения сигнала, заменяются на эквивалентную емкость.

Предложенная модель учитывает цепочки ключей, расположенных как последовательно, так и параллельно. Полученные формулы для расчета эквивалентной емкости представлены ниже.

Формула расчета эквивалентной емкости для последовательно расположенных ключей (рис 3):

$$C_{eq} = C_{src} + \sum_{i=1}^{n-1} (\alpha * C_{swi}) + C_{swn} , \quad (1)$$

где C_{eq} – эквивалентная емкость, C_{src} – нагрузочная емкость рассматриваемого буферизующего элемента, C_{swi} – нагрузочные емкости промежуточных ключей, C_{swn} – нагрузочная емкость крайнего ключа из рассматриваемого отрезка пути, α – балансирующий коэффициент, n – количество ключей в ветке.

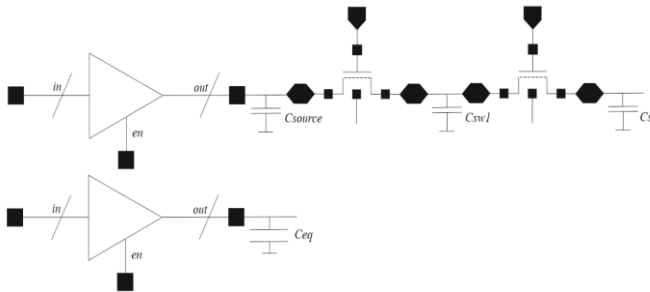


Рис. 3. Цепочка проходных ключей и емкостей и их замена эквивалентной емкостью

Формула расчета эквивалентной емкости параллельно стоящих цепочек ключей (рис. 4):

$$C_{eq} = C_{src} + \sum_{j=1}^m \left(\sum_{i=1}^{n-1} (\alpha * C_{swij}) \right) + C_{swnj} , \quad (2)$$

где C_{src} – нагрузочная емкость рассматриваемого буферизующего элемента, C_{swij} – нагрузочные емкости промежуточных ключей из каждой параллельной ветки, C_{swnj} – емкости крайних ключей из каждой параллельной ветки, α – балансирующий коэффициент, m – общее количество параллельных веток одной цепи, n – количество ключей в ветке.

Для получения формулы расчета конечной эквивалентной емкости без использования паразитных сопротивлений и коэффициента коррекции было проведено исследование буферных элементов ПЛИС и РСнК с разной нагрузочной емкостью.

Исследование проводилось для технологии SOI 180 нм на основании результатов SPICE-моделирования буферных элементов. За образец были взяты результаты моделирования с использованием нагрузочной емкости в виде последовательно соединенных проходных ключей (от одного до шести) и паразитными емкостями на их выходах (рис. 3).

Для калибровки модели эквивалентной емкости сравнивались четыре зависимых от неё выходных параметра: время задержки переключения выхода относительно входа по фронту и спаду, а также время нарастания и убывания выходного сигнала. Задержка измерялась на уровнях 40% (для фронта) и 60% (для спада) от напряжения питания. Время нарастания выходных фронтов измерялось на уровнях 20% (для фронта) и 80% (для спада).

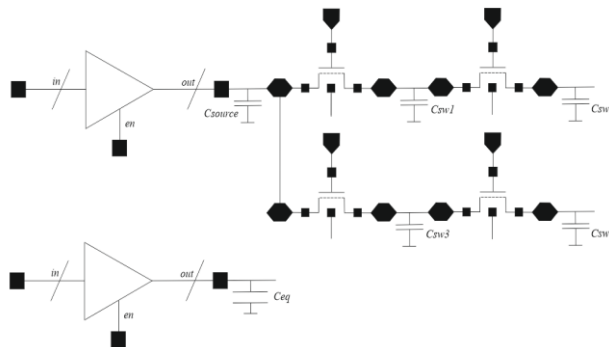


Рис. 4. Параллельные цепочки подряд стоящих нагрузочных ключей и их емкостей и замена эквивалентной емкостью

В результате исследования было получено оптимальное значение корректирующего коэффициента $\alpha = 2$ для технологии SOI 180 нм.

На рис. 5 изображены временные диаграммы, показывающие зависимость выходного сигнала от нагружающей его цепочки элементов, принятой за образец – сплошная линия, и зависимость выходного сигнала от рассчитанной емкости с полученным значением α – пунктирная линия.

По результатам сравнения – при замене реальной выходной нагрузки трассировочных элементов на эквивалентную – разница в задержках и времени убывания не превысила 10%. Время нарастания сигнала отличается на ~20%, что вследствие малой величины элементов не сильно влияет на определение конечной задержки нагружаемого элемента и зависит от того, при каких значениях напряжения выходного сигнала были проведены измерения.

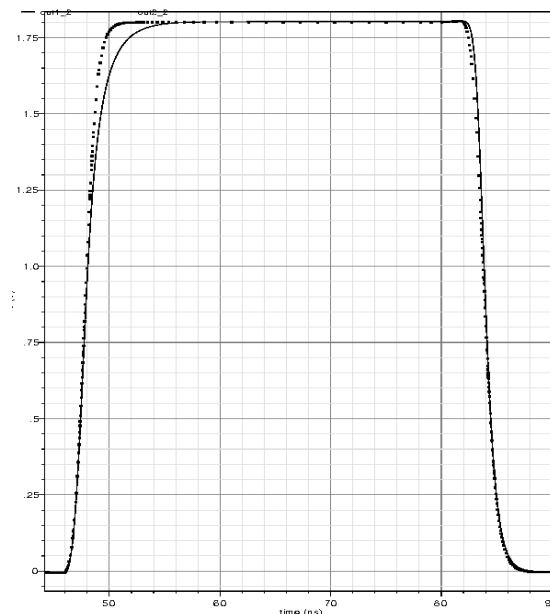


Рис.5. Выходной сигнал трассировочного элемента при нагрузке выхода ключами с первоначальной и эквивалентной емкостью

Точность расчета эквивалентной емкости варьируется в зависимости от топологических параметров транзисторов выходного каскада коммутационного элемента (ширина – W, длина – L, умножение – m). Для повышения точности расчетов задержки элемента в дальнейшем планируется разработать метод, позволяющий разработчику ПЛИС задать данные особенности трассировочных элементов и добавить их в разработанную модель.

F. Генерация файлов с Verilog-описаниями схемы

Перед каждым запуском СВА генерируется файл с Verilog-описанием схемы. После этапа логического синтеза формируется файл с соединенными между собой логическими элементами, после этапа трассировки на межсоединениях логических элементов добавляются трассировочные элементы. Пример Verilog-описания после этапа трассировки приведен ниже:

```
io_pad_inp io_buf (.a(IN), .x(out_io));
re_buf buf (.a(out_io), .en(vdd!), .x(out_b));
le_nand2 nand (.a(out_b), .b(out_c), .y(out_nand));
```

I. Интеграция и проведение СВА

СВА основан на следующих основных понятиях:

A. Required Time (RT)

RT – требуемое время прибытия сигнала. Для комбинационной логики – это время, за которое сигнал должен дойти от входа до выхода. Для последовательностной логики RT – это время, за которое данные должны установиться относительно необходимого им переключения тактового сигнала. Это ограничение рассчитывается по формуле:

$$RT = T_{per} * n - T_{set}, \quad (3)$$

где n – номер переключения тактового сигнала, T_{per} – время периода CLK, T_{set} – время установки.

T_{per} передается процедуре пользователем с помощью опции, T_{set} – автоматически загружается из описания триггера библиотеки .lib.

B. Arrival Time (AT)

Фактическое или реальное время, за которое сигнал достигает выхода.

C. Slack

Slack может быть как положительным, так и отрицательным. Положительный slack считается нарушением. Отрицательный slack называется запасом [12]. Slack рассчитывается по формуле:

$$slack = AT - RT, \quad (4)$$

где AT и RT – значения фактического и требуемого времени прибытия, описанные выше.

Все процедуры и функции, выполняющиеся на различных этапах рассмотренного маршрута проектирования, связаны между собой с помощью

управляющего скрипта, написанного на языке Tcl. Структура скрипта позволяет передать при запуске необходимые параметры и использовать их при запуске любой процедуры.

Для интеграции СВА в общий маршрут была написана специальная процедура, которая обрабатывает входные параметры, проверяет их корректность и непосредственно запускает СВА. Данная процедура вызывается в управляющем скрипте в соответствии с рис. 1 – после этапа логического синтеза и после этапа трассировки.

Запуск STA выполняется из процедуры следующей командой:

```
sta_run -i <ckt_name> -f <file_name>
-num_path <n_path> -clk <clk_list> -per <per_list>
-max_del <max_del> -tran_f <tran_f> -cap_f <cap_f>
-lib <lib_list>
```

где:

<ckt_name>	- имя топового модуля в файле
<file_name>	- имя файла для анализа
<n_path>	- количество критических путей, выведенных в отчет
<clk_list>	- список тактовых сигналов в проекте
<per_list>	- список значений периодов тактовых сигналов
<max_delay>	- значение максимальной задержки от входа до выхода
<tran_f>	- файл с задержками элементов, межсоединений, входов и выходов;
<cap_f>	- файл с емкостями цепей
<lib_list>	- список библиотек, используемых в загруженном Verilog-описании

III. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для подтверждения работоспособности разработанного метода оценки быстродействия схем после трассировки и разработанной модели эквивалентной емкости представлены экспериментальные данные для 3 схем: комбинационных – c17 из набора ISCAS'89, multiplier_2x2 и схемы счетчика с использованием триггеров – gray counter. Результаты STA с использованием эквивалентной емкости сравнивались с результатами Spice-моделирования в Cadence Spectre с паразитными емкостями без удаления ключей.

Сравнение результатов показало, что разработанный метод позволяет достаточно точно определить задержки критических путей и определить быстродействие схем на ПЛИС и РСнК. Полученная погрешность метода находится в пределах 5%. Результаты сравнения представлены в таблице 1.

Таблица 1

Сравнение задержек, полученных с помощью SPICE-моделирования и разработанного метода

Имя схемы	Задержки на критическом пути, нс		Погрешность, %
	Spectre	STA	
c17	22.1876	21.938	1.13%
Multiplier 2x2	23.185	22.232	4.11%
Gray Counter	35.333	35.017	0.89%

IV. ЗАКЛЮЧЕНИЕ

В данной работе представлен метод статического временного анализа с учетом трассировочных ресурсов, позволяющий быстро оценить быстродействие ИС на основе ПЛИС и РСнК. Учет трассировочных ресурсов в представленном методе позволяет использование библиотеки характеризованных элементов трассировки и их нагрузочных паразитных емкостей. Паразитные сопротивления в методе не учитывались. Для учета проходных ключей, входящих в состав коммутационных ресурсов ПЛИС, была разработана эквивалентная модель емкости, позволяющая исключить ключ из анализируемого списка соединений, учесть его проходную емкость и вклад в общую задержку цепи.

Разработанный метод позволяет быстро оценить быстродействие схемы, используя только конечный вариант трассировки соединения между элементами, что позволяет не характеризовать логический вентиль с каждым возможным вариантом соединения с соседними элементами.

Метод также помогает определить работоспособность схемы без проведения SPICE-моделирования, которое имеет большие временные затраты и сильно зависит от размеров схемы и количества задействованных коммутационных ресурсов. Рассчитанные временные ограничения могут быть выведены в файл формата SDF (Standard Delay Format), который в дальнейшем может быть использован для функционального Verilog-моделирования.

ЛИТЕРАТУРА

- [1] IEEE Standard for Standard Delay Format (SDF) for the Electronic Design Process (1497-2001) // IEEE. [Электронный ресурс]. Системные требования: Adobe Acrobat Reader. Режим доступа: <https://ieeexplore.ieee.org/document/972829> (дата обращения: 15.02.2020).
- [2] Vivado Design Suite User Guide Design Analysis and Closure Techniques (2018). [Электронный ресурс]. Системные требования: Adobe Acrobat Reader. Режим доступа: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug906-vivado-design-analysis.pdf (дата обращения: 27.11.2019).
- [3] Lu J., Xu N., Yu J., Weng T., Research of timing graph traversal algorithm in static timing analysis based on FPGA. 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, 2017, pp. 334-338.
- [4] Verilog-to-Routing. FPGA Architecture Description. Architecture Reference. Wire Segments. (2016). [Электронный ресурс]. Режим доступа: <https://docs.verilogtorouting.org/en/latest/arch/reference> (дата обращения: 27.11.2019).
- [5] Gavrilov S.V., Zheleznikov D.A., Zapletima M.V., Khvatov V.M., Chochaev R. Zh., Enns V.I. Layout Synthesis Design Flow for Special-Purpose Reconfigurable Systems-on-a-Chip. Russian. Microelectronics, 2019, Vol. 48, No. 3, pp. 176-186.
- [6] ABC. A System for Sequential Synthesis and Verification. [Электронный ресурс]. Режим доступа: <https://people.eecs.berkeley.edu/~alanmi/abc/> (дата обращения: 21.03.2020).
- [7] Huang T., Wong M. D. F. OpenTimer: A high-performance timing analysis tool. 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, 2015, pp. 895-902.
- [8] Rabaey Jon M., Chandrakasan A., Nicolic B. Digital Integrated Circuits: A Design Perspective 2nd Edition. Upper Saddle River, New Jersey, Pearson Education Inc., 2003, 914 p.
- [9] Стенковский А.Л., Гаврилов С.В., Глебов А.Л. Методы логического и логико-временного анализа цифровых КМОП СБИС. М.: Наука, 2007. 220 с.
- [10] Khvatov V. M., Garbulina T. V., Zheleznikov D. A. Development and Verification of Various Formats of Functional Blocks Libraries as a Part of the Design Flow for FPGAs. 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus), Moscow, 2019, pp. 1687-1691.
- [11] McMurchie L., Ebeling C. PathFinder: a negotiation-based performance-driven router for FPGAs // Proceedings of the 3rd International ACM Symposium on FPGAs, Napa Valley, CA, 1995. PP. 111-117.
- [12] Bhasker J., Chadha R. *Static Timing Analysis for Nanometer Designs: A Practical Approach* (1st ed.). Berlin: Springer Publishing Company, 2009. 572 p.

Static Timing Analysis Method with Routing Resources Estimation for Reconfigurable System-on-a-Chip

S. V. Gavrilov, V. M. Khvatov, D. A. Zheleznikov, T. V. Garbulina

Institute for Design Problems in Microelectronics of RAS, Moscow

sergey_g@ippm.ru, khvatov_v@ippm.ru, zheleznikov_d@ippm.ru, garbulina_t@ippm.ru

Abstract — The well-known methods for evaluating the performance of digital circuits designed on the basis of Field-Programmable Gate Array (FPGA) or Reconfigurable System-on-a-Chip (RSoC) use the full post-routing circuit simulation or pre-calculated interconnect delays in conjunction with the logic elements delays.

Full circuit simulation involves the use of transistors Spice-models and can be time-consuming. Simulation time is highly dependent on the circuit size and the amount of used routing resources. The delays calculation of each path also requires a lot of time due to large number of possible nets switching options. Therefore, a fast circuit performance estimation with a dynamic calculation of net delays before the stage of timing verification using Spice is required.

This article shows the need for a fast circuit performance estimation at the end of various design stages: after logical synthesis in the basis of system-on-chip elements and after element placement and interconnect routing. The paper presents a step-by-step process of adding performance estimation in a whole RSoC automated design flow and describes the developed flow for a post-routing performance estimation.

This method uses software that performs static time analysis using a library of characterized logic elements in Liberty format and net's capacitive characteristics resulting from parasitics extraction.

Keywords — Field-Programmable Gate Array, FPGA, Reconfigurable System-on-Chip, RSoC, Static Timing Analysis, Routing, Performance Estimation

REFERENCES

- [1] IEEE Standard for Standard Delay Format (SDF) for the Electronic Design Process (1497-2001) Available at: <https://ieeexplore.ieee.org/document/972829> (accessed date: 15.02.2020).
- [2] Vivado Design Suite User Guide Design Analysis and Closure Techniques (2018). Available at: https://www.xilinx.com/support/documentation/sw_manual/s/xilinx2018_1/ug906-vivado-design-analysis.pdf (accessed date: 27.11.2019).
- [3] Lu J., Xu N., Yu J., Weng T., Research of timing graph traversal algorithm in static timing analysis based on FPGA. 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, 2017, pp. 334-338.
- [4] Verilog-to-Routing. FPGA Architecture Description. Architecture Reference. Wire Segments. (2016). Available at: <https://docs.verilogtorouting.org/en/latest/arch/reference> (accessed date: 27.11.2019).
- [5] Gavrilov S.V., Zheleznikov D.A., Zapletina M.V., Khvatov V.M., Chochaev R. Zh., Enns V.I. Layout Synthesis Design Flow for Special-Purpose Reconfigurable Systems-on-a-Chip. Russian. Microelectronics, 2019, Vol. 48, No. 3, pp. 176–186.
- [6] ABC. A System for Sequential Synthesis and Verification. Available at: <https://people.eecs.berkeley.edu/~alanmi/abc/> (accessed date: 21.03.2020).
- [7] Huang T., Wong M. D. F. OpenTimer: A high-performance timing analysis tool. 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, 2015, pp. 895-902.
- [8] Rabaey Jon M., Chandrakasan A., Nicolic B. Digital Integrated Circuits: A Design Perspective 2nd Edition. Upper Saddle River, New Jersey, Pearson Education Inc., 2003, 914 p.
- [9] Stempkovskiy A.L., Gavrilov S.V., Glebov F.L. Metody logicheskogo i logiko-vremennogo analiza tsifrovyykh KMOP SBIS [Methods of logical and logical-timing analysis of digital CMOS VLSI]. Moscow, Nauka Publ., 2007. 220 p.
- [10] Khvatov V. M., Garbulina T. V., Zheleznikov D. A. Development and Verification of Various Formats of Functional Blocks Libraries as a Part of the Design Flow for FPGAs. 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, 2019, pp. 1687-1691.
- [11] McMurchie, L. Ebeling C. PathFinder: a negotiation-based performance-driven router for FPGAs // Proceedings of the 3rd International ACM Symposium on FPGAs, Napa Valley, CA, 1995. PP. 111–117.
- [12] Bhasker J., Chadha R. Static Timing Analysis for Nanometer Designs: A Practical Approach (1st ed.). Berlin, Springer Publishing Company, 2009, 572 p.