

А. А. Лялинский, ст. науч. сотр., e-mail: zelyal@inbox.ru,
Институт проблем проектирования в микроэлектронике РАН, Зеленоград

Распараллеливание в процессах характеризации библиотек цифровых элементов

Рассмотрены различные аспекты (запуск задач, мониторинг исполнения, сбор данных, их визуализация, формирование итоговых таблиц) задачи характеризации библиотек цифровых элементов, используемых в процессе создания и производства интегральных схем. Проанализировано применение распараллеливания на различных этапах задачи характеризации. Описана реализация предложенных методов на примере разработки системы веб-характеризации библиотек цифровых ячеек.

Ключевые слова: параллельные процессы, характеризация библиотек ячеек, веб-системы САПР

Введение

Сегодня невозможно представить разработку проекта электронного устройства, имеющего цифровой блок хотя бы небольшого размера, без заранее разработанной и протестированной библиотеки цифровых ячеек. Собственно процесс тестирования получил название "характеризация библиотеки". Его суть — это проведение на схемотехническом уровне расчета каждой ячейки библиотеки в целях получения основных функциональных характеристик ячейки (времени задержки, времени формирования выходных фронтов сигнала, потребляемой мощности; кроме того, для элементов памяти дополнительно вычисляются параметры, связанные с временем установки сигнала после переключения). Расчеты проводятся для определенного пользователем диапазона значений нагрузок на выходе элемента и для диапазона фронта входного сигнала. Кроме того, дополнительно рассчитываются варианты для различных значений температуры окружающей среды и при определенном разбросе напряжения питания. Результаты сохраняются в табличной форме. В простейшем случае аргументы таблицы — это значения нагрузки и наклона входного сигнала, значения таблицы — различные вычисленные функциональные параметры ячейки. Полученные таким образом таблицы позволяют при дальнейшем проектировании устройства использовать программы логического моделирования, имеющие

дело с макромоделями ячеек. Поведение ячейки определяется заложенным в нее функционалом, а характеристики прохождения сигнала и форма выходного импульса определяются из рассчитанных ранее таблиц. Такой подход позволяет на два-три порядка ускорить проектирование устройства в целом.

Следует отметить два важных момента при разработке библиотеки цифровых ячеек:

- определение состава библиотеки;
- вычисление характеристик входящих в ее состав ячеек (или кратко "характеризация библиотеки").

1. **Состав** библиотеки определяется как возможностями получения максимума различных логических функций и типов цифровых ячеек, так и стремлением охватить требуемый диапазон по быстродействию и нагрузочной способности ячеек. Отметим, что функциональное многообразие комбинационного раздела библиотеки не связано с технологическими ограничениями, накладываемыми процессом изготовления кристалла, а определяется только временем и ресурсами, выделенными на создание библиотеки. Понятно, что чем больше набор логических функций библиотеки, тем больше возможностей для создания оптимального устройства в целом предоставляется его разработчику.

2. Особенностью **процесса характеризации** является его большая временная продолжительность. Библиотека ячеек может содержать несколько сотен различных вариантов цифро-

вых ячеек. Вариативность обусловлена как различными логическими функциями, реализуемыми в библиотечных ячейках, так и необходимостью иметь для одной и той же функции линейку вариантов, различающихся быстродействием и нагрузочной способностью. Для каждой ячейки необходимо вычислить параметры как минимум для худшего, номинального и лучшего условий эксплуатации аппаратуры, под которыми понимаются температура окружающей среды и напряжение питания.

В соответствии с общепринятым в настоящее время форматом хранения данных в виде Liberty-файла [1] для каждой ячейки в зависимости от ее типа — комбинационная или ячейка памяти — необходимо вычислить набор табличных параметров, описывающих ее временные и мощностные характеристики. Каждая из таблиц вычисляется на сетке соответствующих параметров (например, вычисляется таблица значений задержки прохождения сигнала в зависимости от нагрузочной емкости и наклона входного сигнала). Для комбинационных схем временные затраты растут с увеличением сложности реализованной логической функции. Для ячеек памяти их сложность не оказывает такого существенного влияния, так как ее разброс не столь велик, как для комбинационных схем. Главным является то, что для них вычисляется большее число параметров, причем очень часто вычисление ряда параметров ячеек памяти связано с итерационными циклами. Поэтому временные затраты на характеризацию ячеек памяти существенно больше по сравнению с комбинационными схемами.

Совокупность перечисленных выше факторов приводит к тому, что для полноценного расчета всей библиотеки требуются 1...2 недели постоянной работы компьютера. Таким образом, задача ускорения характеризации библиотеки цифровых ячеек становится достаточно актуальной.

Одним из способов ускорения вычислений является подключение к процессу всего доступного пула компьютеров. Оценим возможности ускорения. Достаточно обычным является наличие во внутренней сети организации нескольких десятков вычислительных серверов. Предположим, что для выполнения задачи характеризации из них доступны хотя бы 4...6 серверов. Обычно каждый такой сервер имеет не менее 24 ядер. Тогда в предельном случае можно задействовать для расчетов примерно 100...150 независимых вычислительных ядер. Простой прямолинейный способ ускорения — вручную раскидать все задачи характеризации

на доступные ядра — имеет ряд существенных недостатков. Во-первых, подготовка данных для задачи характеризации является достаточно трудоемким и тонким в настройке процессом. Во-вторых, отслеживание состояния запущенных задач и сбор результатов выполнения для такого огромного числа задач само по себе является непростой задачей. Как результат, становится актуальным исследование возможности распараллеливания задачи характеризации цифровой библиотеки на пуле вычислительных ядер.

В данной работе исследуются вопросы выделения отдельных задач из процесса характеризации в целом, а также оптимального построения системы характеризации, работающей с различными уровнями распараллеливания процессов.

Структура процесса характеризации

Как описано выше, процесс характеризации цифровых библиотек имеет несколько слоев (рис. 1).

Уровень L1 — это процесс характеризации библиотеки в целом, распараллеливание отсутствует.

Уровень L2 — получение всех параметров во всех точках сетки для отдельной ячейки, распараллеливание путем одновременного расчета всех ячеек.

Уровень L3 — вычисление всех параметров в одной точке сетки для отдельной ячейки, дополнительно к уровню L2 распараллеливание путем одновременного вычисления узлов сетки.

Уровень L4 — вычисление отдельного параметра в одной точке сетки для отдельной ячейки, дополнительно к уровню L3 распараллеливание путем одновременного вычисления всех требуемых параметров ячейки.

К счастью, все процессы на всех уровнях выполняются независимо друг от друга, поэтому их можно распараллелить вплоть до самого нижнего уровня.

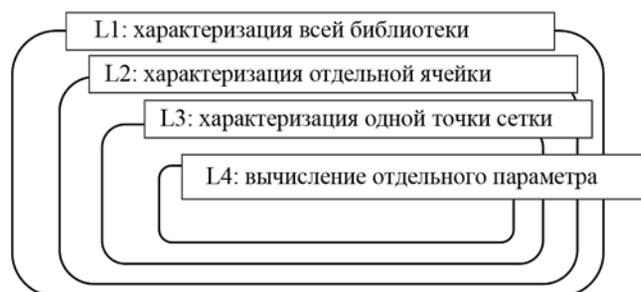


Рис. 1. Структура процесса характеризации библиотеки цифровых ячеек

Система веб-характеризации

В качестве примера для сравнения возьмем используемую много лет пакетную версию системы характеризации цифровых библиотек [2]. Некоторые из ее блоков вошли во вновь разработанную веб-систему характеризации библиотек цифровых ячеек, представляемую в этой работе.

В текущей версии эта веб-система ориентирована на библиотеку полузаказных ячеек [3] (все ячейки имеют одну и ту же топологию, выбор функциональности осуществляется путем подачи определенной совокупности сигналов на входы "прошивки"), в дальнейшем предполагается ее расширение для библиотек произвольных ячеек. Система реализована на основе управляющего веб-сервера, взаимодействующего с пулом из вычислительных ядер (рис. 2). Обмен данными осуществляется через общее дисковое пространство.

Система частично использует некоторые блоки ранее разработанной пакетной версии. Можно считать, что расчет на уровне L1 — это характеризация, проводимая в системе с пакетным исполнением. В предлагаемой системе веб-характеризации предусмотрены возможности распараллеливания на уровнях L2...L4 (уровни L2 и L3 реализованы, уровень L4 — в проекте будущих работ).

Последовательность операций при проведении процесса моделирования на уровнях L2...L4 состоит из следующих шагов:

1. Прием исходной информации от пользователя и формирование списка задач.
2. Получение списка доступных вычислительных ядер.
3. Работа блока подготовки и запуска отдельных задач. Новые задачи запускаются при соблюдении следующих условий:

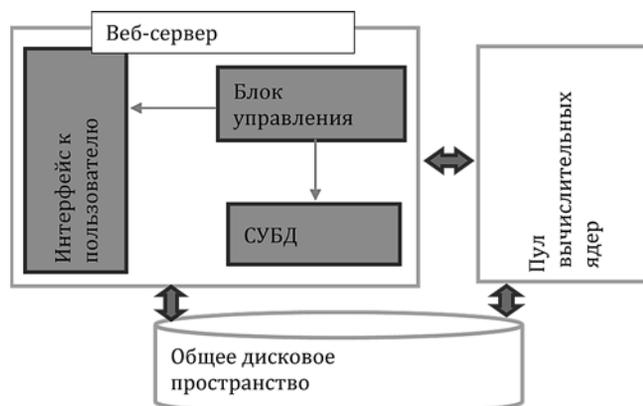


Рис. 2. Структура системы веб-характеризации

- имеется свободное вычислительное ядро;
- не превышено максимальное время отклика на HTTP-запрос.

4. Передача управления в модуль визуализации и контроля исполнения. Проверка состояния запущенных задач. Показ текущих полученных результатов. Если есть хотя бы одна невыполненная задача, то передача управления на шаг 3.

5. Если все задачи выполнены, то перейти к формированию итоговых таблиц и завершению работы алгоритма.

Основная цель данного алгоритма — постараться выполнить все намеченные задачи в минимально возможное время на доступном ресурсе вычислительных ядер с максимальным подержанием собственной работоспособности.

Особенности разработанной системы

Имеет смысл описать две особенности системы, обусловленные ее реализацией в виде набора скриптов, размещенных на веб-сервере.

1. Ограничения работоспособности блоков системы

Реализация системы в виде веб-сайта накладывает определенные требования к рабочим характеристикам составляющих ее блоков [4]. Важнейшими являются три следующих, перечисленных по мере уменьшения их значимости:

- 1) ограничение времени работы блока;
- 2) ограниченный объем памяти ОЗУ;
- 3) ограниченный объем данных, которыми можно обмениваться между блоками.

Первое ограничение (на время работы) вытекает из протокола обмена по HTTP-запросам — ответ может ожидаться не более 30 с (есть два способа увеличить этот порог максимум до 5 мин, но отменить его нельзя). Ограничение продиктовано необходимостью недопущения ситуации, когда скрипт на веб-сервере по каким-либо причинам зацикливается и не может самостоятельно прекратить свое выполнение. Но у скриптов разработанной системы есть достаточно большое число ситуаций, когда объем данных требует неизвестное заранее время на выполнение задачи. Для решения этой проблемы для критических блоков вводится самоконтроль времени исполнения и организация самостоятельного повторного вызова блока с предварительным сохранением и последующим восстановлением оперативных данных (аналогию можно найти с понятием организации "критических точек").

Второе ограничение (на объем памяти) вытекает из реализации блоков системы в виде PHP-скриптов, для которых всегда существует максимальный объем используемой динамической памяти ОЗУ. Опять же, этот предел можно самостоятельно увеличить, но нельзя отменить. Полноценного решения этой проблемы нет, при разработке любого скрипта необходимо держать в уме это ограничение. В критических случаях приходится отказываться от хранения всех данных в ОЗУ и вводить комплекс обменных операций с рабочими файлами. Частичным решением проблемы может быть установка в PHP-сервер расширения Memcached [5], позволяющего работать с неограниченной памятью ОЗУ, которое на самом деле заменяет хранение данных в памяти компьютера их хранением в файлах.

Третье ограничение (на объем обмениваемых данных) скорее имеет вторичный характер — слишком большой объем обрабатываемых данных приведет к тому, что вступят в действие ограничения по времени или памяти. Формальный момент — достаточно удобный способ обмена через глобальную PHP-переменную `$_SESSION` также имеет некое (но достаточно большое) ограничение. Выход — контролировать операции блока и не допускать долгих операций разработки с большими блоками данных. В критических случаях необходимо разделять операции на более мелкие процессы, выполняя их последовательно друг за другом, возможно, с организацией циклов.

2. Борьба с возникновением некорректных данных

В процессе отладки и эксплуатации системы были выявлены два источника некорректных данных.

1. При практически одновременном запуске большого числа небольших задач (например, это нормальная ситуация при характеристике комбинационных схем, когда в списке на запуск стоят рядом задачи расчета простейших комбинационных ячеек типа AND2) некоторая (до 5 %) часть задач получает некорректные исходные текстовые данные. Обычно это файл моделей, общий для всех задач. Исследование задач с некорректными данными показывает, что они получают файл с "обрубленной" хвостовой частью. Иными словами, часть задач отправляется на выполнение с неполностью полученным исходным файлом. Результатом является отказ программы Spice-моделирования рассчитывать схему с выдачей соответствующей диагностики.

Выявить такую ситуацию до начала моделирования не представляется возможным, но исправить это достаточно легко. Если при анализе результатов завершенной задачи мы видим сообщение об аварийном завершении, то данная задача ставится на повторное исполнение. Практика показывает, что запущенная при иных обстоятельствах, она удачно решается. Во избежание заикливания число повторов одной и той же задачи ограничено.

2. При расчете схем памяти могут возникать ситуации, когда из-за некорректно поставленных исходных данных (например, не совсем точно заданных напряжения питания или параметров измерительной сетки) не удастся получить корректные данные для части точек сетки. Пока это случалось достаточно редко, но теоретически такая ситуация возможна. Одним из выходов представляется подготовка результирующих данных с неполной сеткой, пустоты в сетке могут заполняться интер- и экстраполяцией с соседних узлов. Данная проблема требует своего исследования.

Описание основных блоков системы

Рассмотрим вкратце основные блоки. Кроме описанных ниже в системе имеется еще более десятка блоков, осуществляющих различные подготовительные и сервисные операции, их описание в данную статью не включено.

1. Интерфейс к пользователю

Данная система изначально спроектирована как интерактивная, ее интерфейс построен на основе веб-страниц, что позволяет вести общение с системой через любой веб-браузер (рис. 3, см. третью сторону обложки). Это позволяет иметь доступ к системе практически с любого компьютера в той сети, где доступна данная система, предустановки какого-либо программного обеспечения не требуется.

В данном блоке на основе полученной от пользователя информации формируется список намеченных к выполнению задач. Понятие "задача" зависит от выбранного текущего уровня распараллеливания. Это может быть полный расчет ячейки библиотеки, расчет одной точки сетки для одной ячейки или расчет одного типа параметров.

2. Блок подготовки задания на моделирование

Этот блок и описываемый за ним блок визуализации текущей информации вызывают друг

друга в цикле до тех пор, пока не закончится список намеченных к выполнению задач. На рис. 4 (см. третью сторону обложки) представлено стартовое меню задачи характеристики. Пользователь выбирает режим распараллеливания и набор ячеек для характеристики. Под конкретный случай расчета параметров формируются набор входных сигналов [6], файлы, определяющие вычисление параметров (обычно с помощью Spice-директивы measure) и, в завершение, исходный файл, после этого запускается Spice-моделирование.

В начале работы блока анализируется список разрешенных к использованию вычислительных серверов, определяется число вычислительных ядер на каждом сервере, после чего все ядра ранжируются по степени их текущей общей загрузки. Из полученного списка выбирается столько наименее загруженных ядер, сколько невыполненных задач имеется на данный момент. Для каждой задачи готовится полный пакет исходных файлов, и задача запускается на исполнение на выбранном ядре.

Отметим два важных момента:

1. Алгоритм выбора вычислительных ядер старается равномерно загрузить вычислительные серверы, что позволяет достичь максимальной производительности системы в целом.

2. Ведется контроль времени исполнения блока. При достижении отметки в 25 с запуск задач прекращается, и управление передается в блок визуализации текущей информации. Такой самоконтроль позволяет не допускать аварийного завершения работы данного скрипта, вызванного превышением разрешенного времени исполнения.

3. Блок визуализации текущей информации и контроля исполнения задания

Данный блок работает в тесной связке с предыдущим блоком и выполняет две важных задачи:

1. Анализируя различные факторы (предыдущее и текущее состояния задачи в CPU, наличие результирующих файлов), определяет "пользовательский" статус каждой задачи — стартовала ли она, в процессе выполнения или завершена. Если задача завершена, то анализируется корректность выходных данных, и если обнаружены ошибки, то данная задача ставится на повторное выполнение. Если выходные данные корректны, то в зависимости от уровня распараллеливания проводится их первичная обработка и сохранение во временном хранилище.

Степень текущей загрузки компьютеров: (compNN = 192.168.77.NN)					
в очереди на загрузку: 388					
	comp33	comp35	comp37	comp38	всего
индекс загрузки	0.93	0.89	0.97	0.82	
#по плану	23	21	23	25	92 из 480
#готовых	21	20	21	23	85 из 480
#в CPU	1	1	1	2	5

Отсутствует файл результатов - рестарт задачи al_and3fft_A0A2
Повторный старт вычислений 1 точки.

Рис. 5. Меню загрузки пула вычислительных ядер

Количество точек рассчитываемой сетки (для каждой маски): 40						
Примерное время завершения расчетов: 17:11:16						
N	маска	#заверш.	старт	заверш.	затрач.	результаты действие
задачи текущего проекта Almaz14_v22						
пакет в целом	201	16:55:00			6:07	
1	al_and2	40	16:55:16	17:01:07	5:51	☰
2	al_and2ft	0				—
3	al_and3	40	16:55:01	16:58:09	3:08	☰
4	al_and3fft	40	16:55:42	16:58:51	3:09	☰
5	al_and3fft	40	16:57:51	17:00:34	2:43	☰
6	al_ao21	10	16:59:35		2:14	—
7	al_nand3fft	31	17:00:41		1:08	—
8	al_nand3	0				—
9	al_nand2ft	0				—
10	al_nand2	0				—
11	al_mux2l	0				—
12	al_mux2h	0				—

N	маска	время расчета одной точки сетки (сек)			
		мин	среднее	макс	предельное
1	al_and2	5	28.7	62	2:35 мин
3	al_and3	7	21.6	53	2:12 мин
4	al_and3fft	1	21.1	42	1:45 мин
5	al_and3fft	7	21.4	49	2:02 мин
6	al_ao21	2	37.9	48	2 мин
7	al_nand3fft	9	18.1	26	1:05 мин

Рис. 6. Меню состояния вычисления задач

На рис. 5 и 6 представлены фрагменты выдачи текущей информации о выполнении задачи.

2. Достаточно большая часть работы этого блока связана с формированием данных для итогового хранения уже завершенных задач.

Кроме того, данный блок обеспечивает текущую визуализацию процесса — информирует о степени загрузки пула вычислительных ядер и состоянии всех намеченных к обработке задач.

4. Блок сохранения и визуализации итоговых данных

Данный блок работает после завершения расчетов всех задач и формирует данные для визуализации окончательных расчетов, позволяющих пользователю принять решение о сохранении полученных результатов в качестве Liberty-файлов.

Сравнение способов подготовки исходной информации

Элементы	Существующая система (пакетное исполнение)	Новая интерактивная веб-система
Подготовка входных сигналов для измерения параметров		
Комбинационные ячейки	В ручном режиме	Полностью автоматизировано
Ячейки ввода-вывода		
Преобразователи уровня		Автоматизировано, реализованы частично
Ячейки памяти	Заранее приготовленные тесты	
Подготовка скриптов для измерения характеристик		
Комбинационные ячейки	Заранее подготовленные скрипты	Более быстрая собственная измерительная часть
Ячейки ввода-вывода		
Преобразователи уровня		Заранее приготовленные тесты
Ячейки памяти	Заранее приготовленные тесты	
Выбор множества ячеек для текущей задачи		
	В ручном режиме (в текстовом редакторе)	Интерактивно из предложенного списка

Результаты моделирования

Таблица 2

Сравнение системы характеристики с аналогами имеет смысл проводить по трем признакам:

- удобство подготовки исходной информации;
- скорость выполнения задачи в целом;
- удобство работы с полученными результатами.

Сравнение проведем с упомянутым выше пакетным вариантом системы характеристики, так как достаточно полной информации о других системах характеристики в открытом доступе не найдено.

1. Подготовка исходной информации

В табл. 1 сведены воедино различные аспекты, связанные с подготовкой исходных данных для задачи характеристики. Видно, что во многих случаях новая система удобнее за счет предоставления интерактивного режима. Наиболее заметно это для комбинационных схем, где за счет более высокого уровня формализации описания удалось достичь более высокого уровня автоматизации процессов.

2. Скорость выполнения задачи в целом

В табл. 2 дано сравнение скорости выполнения задачи характеристики. Для старой системы приведены оценочные данные, базирующиеся на имеющемся опыте эксплуатации этой системы. Из таблицы мы видим ускорение процесса вычислений в 10...20 раз.

Сравнение скорости выполнения задачи характеристики

Элементы	Существующая система (пакетное исполнение) Примерная оценка	Новая интерактивная веб-система
33 комбинационные ячейки	Более суток	2 ч
24 ячейки памяти	От 1,5 до 2 суток	2 ч

Таблица 3

Варианты представления выходной информации

Существующая система (пакетное исполнение)	Новая интерактивная веб-система
Основная выходная информация	
Liberty-файл	<ul style="list-style-type: none"> • Таблицы и 3D-изображения полученных данных • Liberty-файл
Служебная информация (протоколы моделирования и создания Liberty-файла)	
Доступ из Unix-каталога	Интерактивный доступ из меню системы
Варианты формирования составных Liberty-файлов	
Указанием списка в текстовом редакторе	Интерактивный выбор из меню системы
Сервисные функции	
Отсутствуют	<ul style="list-style-type: none"> • Оценка однородности полученных результатов • Оценка статистического разброса данных

3. Удобство работы с полученными результатами

В табл. 3 приведено сравнение различных способов обработки выходной информации данных систем. Видно, что новая система за счет интерактивности обеспечивает больше удобств и возможностей для пользователя. Результаты вычисления параметров ячейки представлены на рис. 7 в виде таблицы и на рис. 8 (см. третью сторону обложки) в виде цветного трехмерного изображения.

Заключение

Представлена новая веб-система характеристики библиотек цифровых схем. Описаны методы и приемы распараллеливания вычислительных процессов, что позволило ускорить данную работу в целом на 1...2 порядка. Показаны приемы, позволяющие преодолеть ограничения, вызванные реализацией данной системы в виде веб-сервера.

Список литературы

1. **Описание** формата Liberty-файла на сайте Synopsis. URL: <https://www.synopsys.com/> (дата проверки: 22.05.2020).

input slew, ps	cap							
	c_out, pF							
	1	20	50	75	100	125	150	200
20	0.00392106	0.00392106	0.00392106	0.00392106	0.00392106	0.00392106	0.00392106	0.00392106
200	0.00513706	0.00513706	0.00513706	0.00513706	0.00513706	0.00513706	0.00513706	0.00513706
500	0.0055124	0.0055124	0.0055124	0.0055124	0.0055124	0.0055124	0.0055124	0.0055124
1000	0.00560405	0.00560405	0.00560405	0.00560405	0.00560405	0.00560405	0.00560405	0.00560405
1200	0.00560612	0.00560613	0.00560613	0.00560613	0.00560613	0.00560613	0.00560613	0.00560613

Рис. 7. Таблица результатов вычисления задержки ячейки

2. **Хватов В. М., Гарбулина Т., Лялинская О. В.** Методы формирования и верификации библиотек стандартных элементов в составе маршрута проектирования ИС на базе ПЛИС отечественного производства // Проблемы разработки перспективных микро- и нанoeлектронных систем. 2018. Вып. 1. С. 57–62.

3. **Ian Poole**, ASIC basics tutorial. URL: https://www.electronics-notes.com/articles/electronic_components/programmable-logic/what-is-an-asic-application-specific-integrated-circuit.php (дата обращения: 03.02.2020).

4. **Лялинский А. А.** Особенности построения прикладных программ с веб-доступом // Информатика. № 1(37). Январь-март 2013. С. 76–83. Минск. ОИПИ НАН Беларуси. 2013.

5. **Документация** по PHP. URL: <https://www.php.net/manual/ru/intro.memcached.php> (дата обращения: 22.05.2020).

6. **Лялинский А. А.** Автоматизированное формирование тестов при характеристике цифровых ячеек с использованием веб-доступа // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2012. Сборник трудов / Под общ. ред. академика РАН А. Л. Стемпковского. М.: ИППМ РАН, 2012. С. 95–100.

A. A. Lyalinsky, Senior Researcher, e-mail: zelyal@inbox.ru,
Institute for Design Problems in Microelectronics of RAS, Moscow

Parallelization in Characterization of Digital Cells Libraries

Various aspects (starting tasks, monitoring execution, collecting data, visualizing, forming final tables) of the problem of characterization of digital elements libraries used in the process of creating and manufacturing integrated circuits are considered. The analysis of parallelization application at various stages of the characterization problem is carried out. The implementation of the proposed methods is described using the example of developing a web-based characterization system for digital cell libraries.

Keywords: parallel processes, characterization of cell libraries, web-based CAD systems

DOI: 10.17587/it.26.618-624

References

1. **Description** of format of Liberty-file on Synopsis site, available at: <https://www.synopsys.com/> (access date: 22.05.2020) (in Russian).

2. **Khvatov V. M., Garbulina T., Lyalinskaya O. V.** Formation and Verification of Standard Element Libraries in the Design Flow for the Domestic FPGAs, *Problems of Perspective Micro- and Nanoelectronic Systems*, Development, 2018, iss. 1, pp. 57–62 (in Russian).

3. **Ian Poole**, ASIC basics tutorial, available at: [\[grammable-logic/what-is-an-asic-application-specific-integrated-circuit.php\]\(https://www.electronics-notes.com/articles/electronic_components/programmable-logic/what-is-an-asic-application-specific-integrated-circuit.php\) \(access date: 03.02.2020\).](https://www.electronics-notes.com/articles/electronic_components/pro-</p>
</div>
<div data-bbox=)

4. **Lyalinsky A.** Osobennosti postroeniya prikladnykh programm s veb-dostupom, *Informatika*, Jan.-March 2013, no. 1 (37), pp. 76–83, Minsk, OIPI NAN of Belarus, 2013 (in Russian).

5. **PHP** documentation, available at: <https://www.php.net/manual/ru/intro.memcached.php> (access date: 22.05.2020) (in Russian).

6. **Lyalinsky A. A.** Web-based automatic generation of input patterns at characterization of digital cells, *Problems of Perspective Micro- and Nanoelectronic Systems Development*, 2012, Proceedings, edited by A. Stempkovsky, Moscow, IPPM RAS, 2012, pp. 95–100 (in Russian).

Рисунки к статье А. А. Лялинского
 «РАСПАРАЛЛЕЛИВАНИЕ В ПРОЦЕССАХ ХАРАКТЕРИЗАЦИИ
 БИБЛИОТЕК ЦИФРОВЫХ ЭЛЕМЕНТОВ»

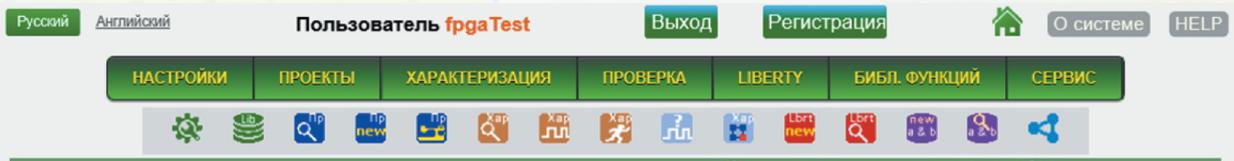


Рис. 3. Меню команд системы веб-характеризации

режим распараллеливания: по схемам по схемам и узлам сетки по всему

- есть данные для создания Liberty-файла
 - есть Liberty-файл
 - обрабатываемая
 - в списке ожидания

Флип-флоп триггеры

прошивка	функция	прошивка	функция
al_dff	!/clk & q /clk & data	al_dfflci	(!clk & q) (clk & !data) clr
al_dffc	(!/clk & q & !clr) (/clk & data & !clr)	al_dffli	!clk & q clk & !data
al_dffci	(!/clk & q) (/clk & !data) clr	al_dffls	(!clk & q) (clk & data) stn
al_dffi	!/clk & q /clk & !data	al_dfflsi	(!clk & q & !stn) (clk & !data & !stn)
al_dffl	!clk & q clk & data	al_dffs	(!/clk & q) (/clk & data) stn
al_dfflc	(!clk & q & !clr) (clk & data & !clr)	al_dffsi	(!/clk & q & !stn) (/clk & !data & !stn)

Рис. 4. Стартовое меню задачи характеризации

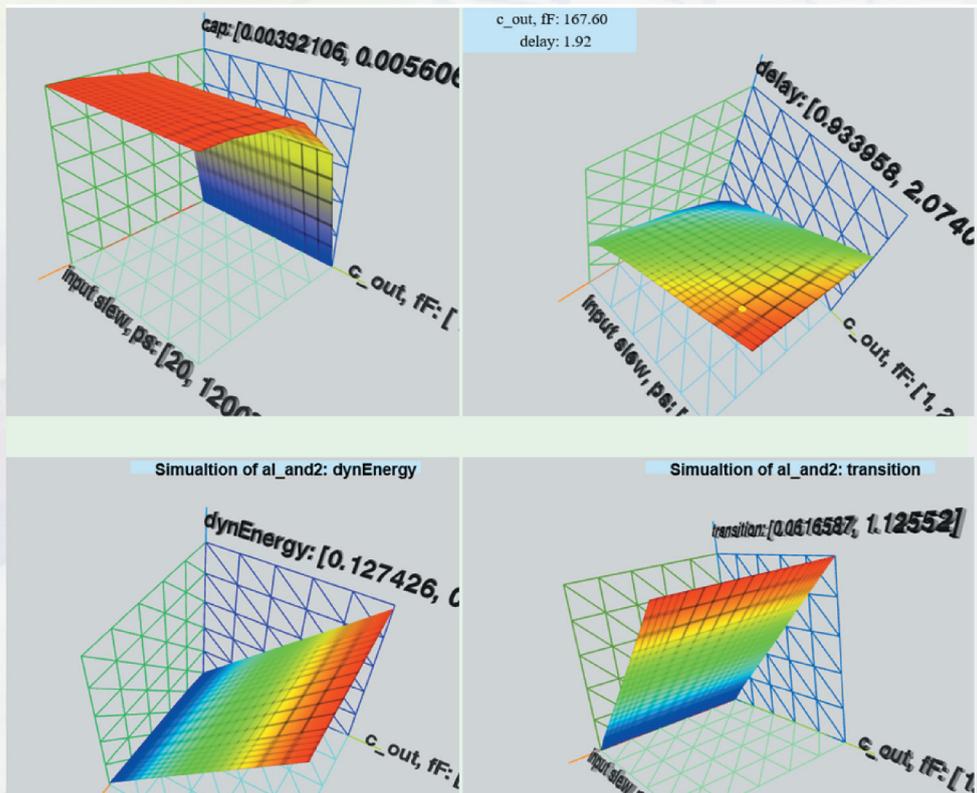


Рис. 8. 3D-плоскости результатов вычисления параметров ячейки