

Формирование высокоуровневых моделей цифровых ячеек с использованием веб-доступа

А.А. Лялинский

Институт проблем проектирования в микроэлектронике РАН, zelyal@inbox.ru

Аннотация — Предлагается система генерации Verilog-моделей стандартных цифровых ячеек. В качестве исходных данных используются логическое выражение, описывающее работу ячейки, и, если это необходимо, её spice-описание. Система имеет веб-доступ и размещена на сайте <http://www.ippm.ru/FastVerMod/>. Благодаря интерактивному доступу значительно ускоряется разработка и отладка высокоуровневых моделей. Иерархия модели по уровню сложности позволяет выбрать ту, которая в наибольшей степени соответствует потребностям пользователя системы.

Ключевые слова — высокоуровневые модели, Verilog, цифровые схемы, веб-доступ.

I. ВВЕДЕНИЕ

Широкое распространение сети Интернет и возросшее количество высокоскоростных каналов связи дают хорошие предпосылки к интеграции всемирной паутины и систем автоматизированного проектирования (САПР), используемых для разработки изделий электронной промышленности. Среди множества направлений исследований по этой тематике [1] наибольший интерес с точки зрения максимального вовлечения новых интернет-технологий представляют следующие:

- построение систем с распределением вычислительных и программных ресурсов по сети компьютеров;
- добавление нового доступа к системам проектирования через веб-страницы или веб-порталы.

Первое направление работ связано как со стремлением снизить стоимость программных комплексов САПР путем размещения их различных частей на нескольких серверах с возможностью оперативного обмена данными, так и с попытками продвинуть программное обеспечение (ПО) САПР в «облачные вычисления» [2], обеспечивающие удобный сетевой доступ к общему пулу вычислительных ресурсов. Цель работ по второму направлению – повысить доступность тех блоков САПР, которые выполняют задачи сравнительно небольшого объема (как по времени, так и по затрачиваемых серверным ресурсам).

В данной работе рассматриваются некоторые аспекты исследований, ведущихся по каждому из этих направлений, и предлагается система генерации высокоуровневых моделей, осуществляющая ввод-вывод данных посредством веб-доступа.

II. САПР НА ОСНОВЕ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Используемые в настоящее время системы проектирования в микроэлектронике состоят из большого количества блоков различного назначения (средства ввода и визуализации данных, моделирования, компоновки и трассировки и т.п.). Соответственно, стоимость большинства таких программных комплексов достаточно велика. С другой стороны, в конкретных маршрутах проектирования интегральных схем (ИС) часто бывает задействовано всего 10-20% имеющегося программного обеспечения [3], что говорит о не очень высокой эффективности финансовых вложений в этой области. Поэтому возникает закономерный вопрос: нельзя ли создать такую программную оболочку, которая позволяла бы построить маршрут проектирования изделия с привлечением только необходимых блоков, снизив тем самым затраты на лицензионного ПО и уменьшив нагрузку на отдельные вычислительные сервера?

Для локально размещенных САПР такие оболочки, представляющие собой командные надстройки над отдельными блоками комплекса проектирования и позволяющие реализовывать заранее заданный маршрут проектирования, известны уже достаточно давно. Например, ADAM (Advanced Design Automation System) [6], OASIS [7], Cadweld [8] и многие другие (подробный обзор можно найти в [5]). Основные трудности возникают при попытке связать с помощью такой оболочки блоки ПО, размещенные на различных, не связанных между собой локальной сетью, серверах. Проблема в данном случае в том, что связь между блоками приходится организовывать не на уровне вызовов функций, а посредством обмена сообщениями через глобальную сеть, что и сложнее в плане программирования, и медленнее из-за связей между удаленными серверами. Среди работ по этому направлению отметим: [9], ExPro [10], WELD [11] и др..

Развитие технологий программирования сближает эти подходы. Так технология «.NET Remoting» [12], включенная в состав пакета «.NET Framework 1.0», и пришедшая ей на смену WCF (Windows Communication Foundation) [13], являющаяся частью пакета «.NET Framework 3.0», позволяют приложению создать объект в другом процессе, который может быть расположен на другом компьютере, соединенном сетью с сервером основного процесса. В результате запросы к удаленным объектам для клиентского кода ничем не отличаются от локальных вызовов. Среда исполнения сама выполняет связь объектов между клиентским и серверным домена-

ми приложения. Стоит, однако, отметить, что существенно упрощающий программирование обмена данными при разработке новых САПР, этот подход неприменим для уже существующего ПО, и, кроме того, остаются вопросы снижения скорости работы из-за обмена данными по сети.

Пока нет сведений о коммерчески успешных случаях построения таких оболочек, хотя существуют системы с достаточно глубокой проработкой схемы обмена данными между блоками. Можно предположить несколько причин неудач таких систем:

- нежелание основных разработчиков САПР для микроэлектроники разрабатывать параллельные версии одного и того же ПО (для локальной и распределенной версий), в связи с чем подобные разработки ведутся в основном в университетских центрах;
- потенциально более низкая производительность таких систем, обусловленная необходимостью дополнительных временных затрат на обмен данными, что также снижает интерес к таким системам;
- при интеграции в единое целое блоков от различных производителей возникают дополнительные проблемы с лицензированием программных продуктов разных фирм.

III. БЛОКИ САПР С ВЕБ-ДОСТУПОМ

Оснащение блока САПР веб-сервисом подразумевает использование канала связи для доступа клиента к удаленному приложению. Это изменяет требования, предъявляемые к блокам САПР. Две черты отличают стандартный канал, используемый для связи интернет-узлов, от канала в локальной сети компьютеров – ограничение по времени ожидания отклика и объем передаваемых данных за одну транзакцию. Первое приводит к тому, что при наличии задач с заведомо большим временем исполнения (более одной минуты) веб-сервис должен быть построен не на стандартных синхронных подходах (после выдачи запроса к серверной части клиентская часть останавливает свою работу в ожидании завершения обработки запроса), а с использованием асинхронных технологий типа AJAX [15] (после выдачи запроса клиентская часть продолжает свою работу, периодически опрашивая серверную часть на предмет окончания обработки запроса). Второе ограничение связано с тем, что мы не можем за одну транзакцию передать на сервер объем данных свыше 2, 4, 8, или более Мбайт (в зависимости от настроек конкретного сервера). К счастью, в интерактивной работе, это требуется не так часто, так как большая часть данных, передаваемых от клиента к серверу – это команды на исполнение тех или иных операций.

Еще одним ограничением является более бедный набор средств разработки графического интерфейса, предоставляемый при разработке веб-страниц.

Из известных реальных систем проектирования с веб-доступом отметим систему barcelonadesign.com фирмы Barcelona Design Inc [14], предлагавшую проек-

тирование операционных усилителей, преобразователей данных и других устройств на основе вводимых пользователем параметров будущей схемы.

В настоящей работе предлагается имеющая веб-доступ система генерации высокоуровневых моделей цифровых схем. Рассматриваются особенности ее построения и варианты создаваемых моделей.

IV. ОСОБЕННОСТИ ВЕБ-ДОСТУПА

Вначале рассмотрим, что нам дает веб-доступ и какие он накладывает ограничения.

Наиболее существенное преимущество – практически неограниченный доступ к использованию системы. Кроме того в связи с размещением системы в одном месте, снимается проблема с распространением и регистрацией данного ПО. Автор системы может довольно легко вести учет запусков системы, что позволяет достаточно быстро анализировать ее работу и оперативно устранять слабые места.

Из недостатков веб-доступа отметим:

- возможные проблемы с доступом, обусловленные низкой пропускной способностью каналов связи или перегруженностью сервера;
- потенциальная опасность хакерских атак на сайт;
- ограничение по времени исполнения задачи из-за того, что ПО, написанное на скриптовых языках, работает медленнее, чем на компилируемых.

V. НАЗНАЧЕНИЕ И СТРУКТУРА СИСТЕМЫ

Данная система предназначена для быстрого получения HDL-моделей стандартных ячеек цифровых библиотек по заданной логической функции. Имеется иерархия моделей по уровню сложности. *Простая* модель содержит описание портов ввода-вывода и логическое выражение, определяющее работу ячейки. *Сложная* HDL-модель дополнительно имеет блок временных параметров, позволяющих определить задержки прохождения сигналов и их фронты. Система может работать как автономно, так и в качестве основы для веб-сервиса, «отдающего» по запросу клиента HDL-модель.

Доступ к системе осуществляется через графическую оболочку, различные части которой можно подключать и отключать как по запросу пользователя, так и автоматически в зависимости от выполняемой задачи. Графическая оболочка состоит из следующих блоков (рис. 1):

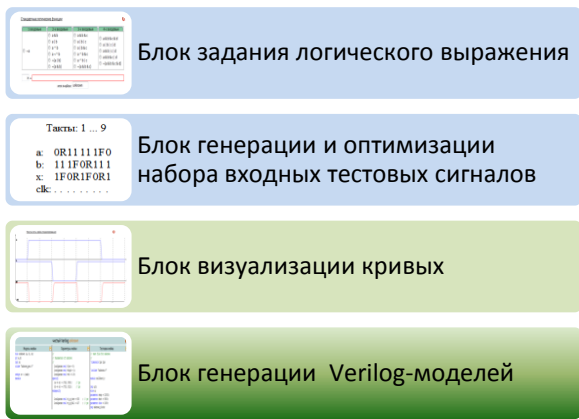


Рис. 1. Структура графической оболочки системы генерации Verilog-моделей

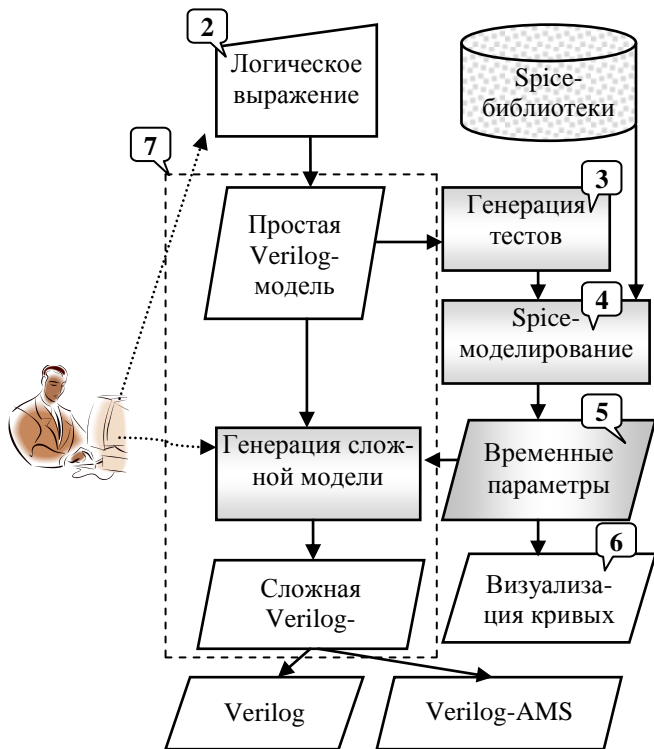


Рис. 2. Структура системы генерации Verilog-моделей

В программном обеспечении системы в целом можно выделить следующие структурные части (рис.2):

1. Графическая оболочка (на рисунке не показана).
2. Блок чтения входной информации. Проверяется наличие и корректность полей ввода.
3. Блок генерации и оптимизации набора входных тестовых сигналов для получения временных параметров ячейки с помощью Spice-моделирования.
4. Блок схемотехнического («Spice») моделирования. Используется внутренняя стандартная программа Spice-моделирования. Допустимо использование любого Spice-симулятора.

5. Блок вычисления фронтов и задержек по результатам Spice-моделирования.
6. Блок визуализации кривых.
7. Блоки генерации Verilog-моделей.

Все части системы, кроме блока Spice-моделирования, реализованы на языке PHP. Spice-моделирование – это завершённый загрузочный модуль, написанный на языке C.

VI. ВАРИАНТЫ ПОСТРОЕНИЯ VERILOG-МОДЕЛЕЙ

В описании высокоуровневой модели можно выделить три основных блока информации:

- интерфейсная часть (описание портов),
- поведенческая часть,
- временные параметры модели.

Сочетание этих блоков и их наполнение определяют тип формируемой модели. В данной системе допускаются формирование HDL-моделей четырех типов:

- Verilog-модель без временных параметров (см. далее А),
- Verilog-модель с временными параметрами (В),
- Verilog-AMS модель со стандартным оформлением параметров (С),
- Verilog-AMS модель с оформлением параметров с помощью «paramset» (D).

Рассмотрим эти варианты подробнее. Далее при записи логического выражения используется нотация языка Verilog.

A. Простая Verilog-модель

Пример: при задании логической функции для ячейки NAND2:

$$x = (\sim a \& b)$$

получим следующую модель:

```
module nand2 (a, b, x);
input a,b;
output x;
assign x = ~(a&b);
endmodule
```

Как мы видим, из заданного логического выражения извлекаются переменные, определяющие входы и выход ячейки, а само логическое выражение без изменений переносится в тело модели.

B. Verilog-модель с временными параметрами

В данном варианте входы-выходы и логическая функция определяются так же, как и в предыдущей модели. Временные параметры определяются с помощью Spice-моделирования. Технологическую библиотеку параметров, необходимую для spice-моделирования, можно либо загрузить свою, либо воспользоваться

включенной по умолчанию. Система предоставляет возможность просмотра выбранной библиотеки.

Для включения временных параметров в Verilog-модель используется конструкция «sparam». В отличие от конструкции #delay (например, $x = \#1 (\sim a \& b)$), использование данной конструкции позволяет указывать отдельные значения rise- и fall-событий, и, кроме того, указывать не только задержку сигнала, но и величину его фронтов, что, в целом, повышает точность модели.

В качестве примера зададим ту же логическую функцию $x = (\sim a \& b)$. Для одной модели будут сформированы два файла: собственно модель и файл ее временных параметров, подключаемый в основную модель с помощью оператора include:

```
// файл модели
module nand2 (a, b, x);
input a,b;
output x;
`include "nand2_par.v"

assign x = ~(a&b);
endmodule

/// файл параметров модели =====
// Parameters of nand2
//
localparam real vlow = 0;
localparam real vhigh = 1;
localparam real vth = 0.5;
specify
(a => x) = (725, 535); // ps
(b => x) = (733, 512); // ps
endspecify

localparam real tt_x_rise = 633 ; // ps
localparam real tt_x_fall = 417 ; // ps
```

C. Verilog-AMS модель

Модель данного типа позволяет, оставаясь на уровне более высоком, чем схемотехнический, контролировать времена и уровни переключения сигналов с точностью, доступной при схемотехническом моделировании. Модель построена на основе компонентов, входящих в аналоговую часть языка Verilog-AMS. Изменение состояния сигналов модели определяется моментами пересечения заданных уровней напряжения.

Так же, как и в предыдущем примере, будут сформированы два файла (модель и параметры). Результаты для той же тестовой функции имеют вид:

```
// файл модели
`include "disciplines.vams"

module nand2 (a, b, x);
input a,b;
output x;
voltage a,b,x;
integer _a,_b,_x;
```

```
real tt; // transition time
real td; // delay to start of output
`include "nand2_par.vams"

analog begin
@(cross(V(a)-vth,0) or initial_step) begin
if (V(a) > vth) begin
_a = 1;
if (V(x) > vth)
td = td_a_x_rf; // rise a --> fall x
else
td = td_a_x_rr; // rise a --> rise x
end else begin
_a = 0;
if (V(x) > vth)
td = td_a_x_ff; // fall a --> fall x
else
td = td_a_x_fr; // fall a --> rise x
end
end
@(cross(V(b)-vth,0) or initial_step) begin
if (V(b) > vth) begin
_b = 1;
if (V(x) > vth)
td = td_b_x_rf; // rise b --> fall x
else
td = td_b_x_rr; // rise b --> rise x
end else begin
_b = 0;
if (V(x) > vth)
td = td_b_x_ff; // fall b --> fall x
else
td = td_b_x_fr; // fall b --> rise x
end
end
if (V(x) > vth)
tt = tt_x_fall; // fall x
else
tt = tt_x_rise; // rise x
_x = ~(_a&_b);
V(x) <+ transition (_x ? vhigh : vlow, td, tt);
end
endmodule
```

```
/// файл параметров модели =====
// Parameters of nand2
//
localparam real vlow = 0;
localparam real vhigh = 1;
localparam real vth = 0.5;
localparam real td_a_x_rf = 0.535705853;
localparam real td_b_x_fr = 0.733871322;
localparam real td_b_x_rf = 0.512686825;
localparam real td_a_x_fr = 0.725908179;

localparam real tt_x_rise = 0.633372761;
localparam real tt_x_fall = 0.417369735;
```

D. Verilog-AMS модель и «paramset»

Этот вариант модели основан на предложении К.Кундерта использовать для описания параметров модели конструкцию типа paramset [4]. Включение параметров модели посредством «paramset» значительно облегчает перенос библиотек технологических файлов, имеющих формат языка Spice (с модификациями, зависящими от используемого варианта Spice), на язык Verilog-AMS. На настоящий момент данная конструкция пока не включена в какой-либо стандарт языка Verilog-AMS.

Файл модели в данном варианте не изменяется, а файл параметров модели имеет вид:

/// файл параметров модели =====

```
paramset env nand2;
.vlow = 0;
.vhigh = 1;
.vth = 0.5;
.td_a_x_rf = 0.53570585377493;
.td_b_x_fr = 0.73387132256361;
.td_b_x_rf = 0.51268682565029;
.td_a_x_fr = 0.72590817962035;
```

```
localparam real tt_x_rise = 0.63337276161129;
localparam real tt_x_fall = 0.417369735408;
endparamset
```

На рис.3...5 представлены примеры ввода данных и представления полученных результатов.

Исходные параметры spice-моделирования

Рекомендации по правильному вычислению временных параметров

Verilog модель ячейки кроме функции, описывающей ее логическое поведение, содержит также блок, определяющий временные параметры ячейки (задержки, фронты). Они вычисляются на основе spice-моделирования. Spice-описание ячейки и библиотека моделей берутся по умолчанию либо загружаются пользователем.

Таким цветом выделены списки ранее использованных наборов параметров

Внешние параметры характеристики ячейки:		Внешние параметры портов ячейки:	
t, °C: 27	VDD, В: 1	t=27 vdd=1	slew, ns: 1 Cload, pF: 0.03
Уровни логических сигналов:		Длительность такта входной тестовой последовательности:	
low: 0	high: 100	clock: 10 ns	
<input checked="" type="radio"/> % от VDD <input type="radio"/> абсолютное значение			

Параметры измерения сигналов:					
(все величины указаны в процентах разницы между верхним и нижним уровнями сигнала)					
Задержка между уровнями:		Фронт входного сигнала между уровнями:		Фронт выходного сигнала между уровнями:	
начало:	конец:	нижний:	верхний:	нижний:	верхний:
50	50	20	80	20	80

Spice-описания	
Ячейка	Имя ячейки в библиотеке ячеек: nand2_1 see ".subckt cell_name ..."
	Имя файла библиотеки ячеек: fvm_cells.lib Обзор...
Модель транзистора	Имя файла библиотеки моделей: fvm_models.lib Обзор...
Общее	Узлы 'земли' и питания: ground: 0 supply: vdd

Порядок узлов в описании ячейки	
Стандартные списки => комбинационная ячейка: outputs inputs комбинационная ячейка: inputs outputs флип-флоп триггер: outputs CLK inputs CLR PRE inputs триггер-защелка: outputs CLK inputs CLR PRE inputs буфер с 3 состояниями: outputs inputs EN	Текущий список: outputs inputs Могут быть использованы только следующие ключевые слова: outputs - один или более выходных узлов inputs - один или более входных узлов (для триггеров - D-входы) OUT - один выходной узел IN - один входной узел EN - узел "enable" (защелка или буфер с 3 состояниями) CLK - узел "clock" (флип-флоп триггер) PRE - узел "preset" (защелка или флип-флоп триггер) CLR - узел "clear" (защелка или флип-флоп триггер)

Рис. 3. Задание параметров, необходимых для проведения spice-моделирования и последующего формирования временных параметров

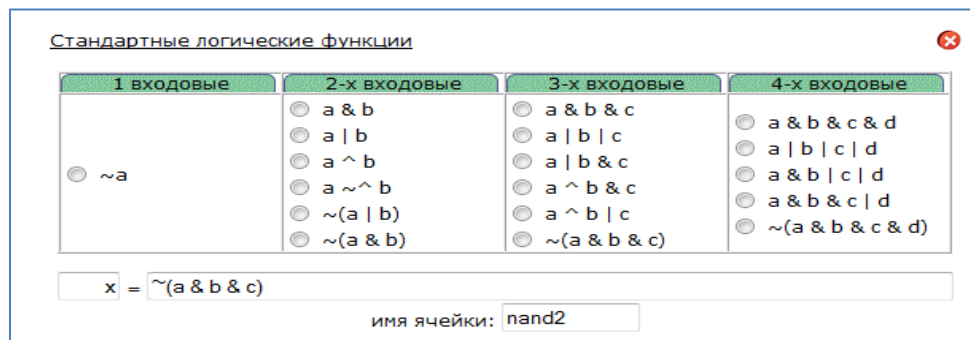


Рис. 4. Ввод логического выражения: можно выбрать одну из стандартных функций или задать произвольное выражение

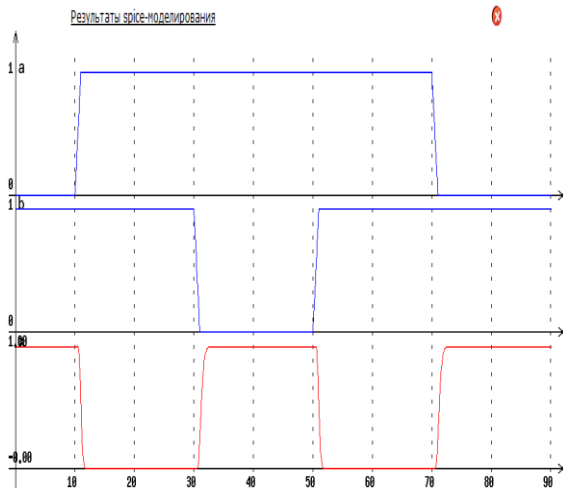


Рис. 5. Отображение результатов spice-моделирования

VII. ВЫВОДЫ

Разработано математическое и программное обеспечение системы автоматической генерации высокоуровневых моделей цифровых ячеек. Веб-доступ к системе можно получить на сайте <http://www.ippm.ru/FastVerMod/>.

ЛИТЕРАТУРА

- [1] Анисимов В.И., Гридин В.Н. Методы построения систем автоматизированного проектирования на основе интернет-технологий и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. 2009. № 1. С. 3-7.
- [2] Antonopoulos Nick, Gillam Lee Cloud Computing: Principles, Systems and Applications // L.: Springer, 2010. 379 p. (Computer Communications and Networks). ISBN 9781849962407.
- [3] Гридин В.Н., Дмитриевич Г.Д., Анисимов Д.А. Построение систем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. №5. С. 23-26.
- [4] Ken Kundert. "paramset: A Verilog-A/MS Implementation of Spice .model Statements". URL: <http://www.designers-guide.org/verilogams/paramset.pdf> (дата обращения 23.04.2012).
- [5] Computer Aided Design and Design Automation. The Circuits and Filters Handbook. Third Edition. Edited by Wai-

Kai Chen. CRC Press. Taylor & Francis Group. 6000 Broken Sound Parkway NW, Suite 300. Boca Raton, FL 33487-2742. ISBNNumber-13: 978-1-4200-5918-2. Chapter 10. 2002.

- [6] Knapp David, Parker Alice The ADAM design planning engine // in book "Artificial intelligence in engineering design (Vol. II)". Academic Press Professional, Inc. San Diego, CA, USA, 1992. pp. 263-285: ISBN 0-12-660562-9.
- [7] OASIS, OASIS Users Guide and Reference Manual. MCNC, Research Triangle Park, NC. 1992.
- [8] Daniell J., Director S.W. An Object Oriented Approach to CAD Tool Control // IEEE Trans. on CAD. June 1991. vol. 10, pp. 698-713.
- [9] Lavana H., Khetawat A., Brglez F., Kozminski K. Executable workflows: A paradigm for collaborative design on the Internet // 34th ACM-IEEE Design Automation Conference. ACM Press, Anaheim Convention Center, Anaheim, CA. June 9-13, 1997. pp. 553-558.
- [10] Rastogi P., Koziki M., Golshani F. ExPro: An expert system based process management system // IEEE Trans. Semiconductor Manuf. 1993. № 6 (3). pp. 207-218.
- [11] Chan F.L., Spiller M.D., Newton A.R. WELD — An environment for web-based electronic design // 35th ACM - IEEE Design Automation Conference. ACM Press, San Francisco, CA. June 1998. pp. 146-151.
- [12] Scott McLean, James Naftel, Kim Williams. Microsoft .NET Remoting. Microsoft Press, 2002.
- [13] Сибраро Пабло, Клайс Курт, Коссолито Фабио, Грабнер Йохан. WCF 4: Windows Communication Foundation и .NET 4 для профессионалов = Professional WCF 4: Windows Communication Foundation with .NET 4. М.: Диалектика, 2011. С. 464. ISBN 978-5-8459-1713-3.
- [14] Mar Hershenson. Op-Amp Design for Switched-Capacitor Stages in Pipelined ADCs // Online Symposium for Electronics Engineers (OSEE). October 2007. URL: <http://www.techonline.com/> (дата обращения 23.04.2012).
- [15] Крейн Дейв, Паскарелло Эрик, Джеймс Даррен. AJAX в действии: технология - Asynchronous JavaScript and XML = Ajax in Action. М.: Вильямс, 2006. С. 640. ISBN 1-932394-61-3.