

ФУНКЦИОНАЛЬНЫЙ КОНТРОЛЬ СБИС ТИПА СнК

Л. А. Золоторевич

Рассматривается проблема обеспечения контроля современных систем на кристалле (СнК) как совокупность методов контролепригодного проектирования, функционального и тестового контроля. Функциональный контроль получил новое развитие применительно к СнК. Это стало возможным на основе языков описания, применения ассерций, расширяющих возможности функционального контроля, разработки и применения тестов, разрабатываемых на разных этапах проектирования.

Введение

Современные технологические возможности производства интегральных схем позволяют размещать на кристалле миллиарды транзисторов [1], а линейный размер транзистора составляет порядка десяти нанометров. Закон Мура о двукратном увеличении числа транзисторов на кристалле за 18 месяцев без существенного увеличения стоимости в последнем десятилетии существенно видоизменился: новые технологии требуют весьма существенных финансовых затрат. Годовой рост числа транзисторов ДОЗУ составляет 58%, а процессоров – 38% в год. На сегодняшний день продолжает усугубляться проблема отставания возможностей технологии проектирования от возможностей технологии производства современных изделий микроэлектроники по сравнению с известными (рис. 1) в литературе данными [2]. Особенно большим является отставание возможностей верификации проектов.

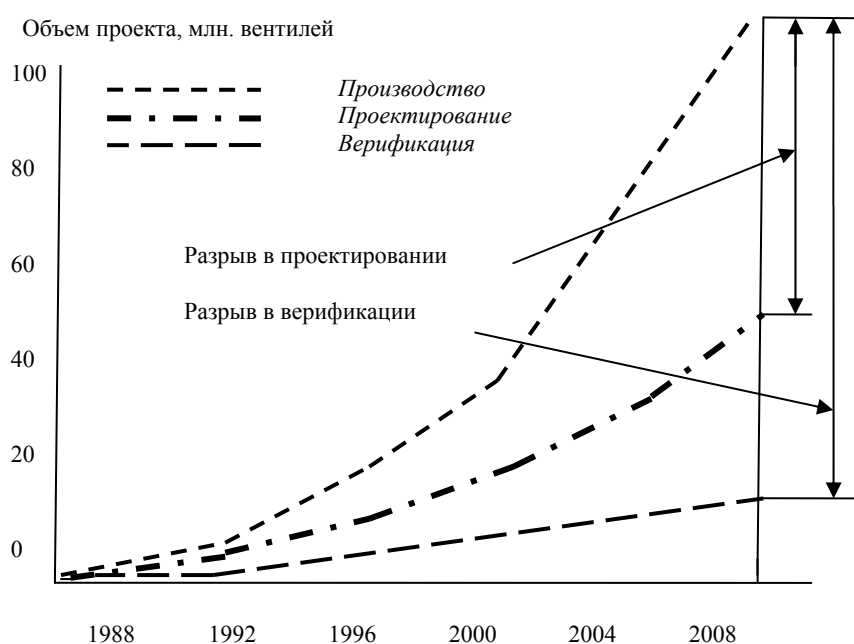


Рис. 1. Сопоставление возможностей производства и проектирования интегральных схем
Одной из весомых причин указанного отставания является высокая

стоимость средств САПР, что, с одной стороны, ограничивает их закупку фаблисс кампаниями, а с другой, снижает рентабельность развития САПР. Вследствие этого фирмы разработчики САПР пытаются параллельно работать в области проектирования и продажи готовых проектов (IP's) и не всегда заинтересованы раскрывать все тонкости технологии проектирования. В то же время из-за неполной загруженности кремниевых фабрик и благодаря полной доступности информации по готовым проектам последние также внедряются в сферу проектирования с целью получения дополнительных доходов [3]. Это приводит к тому, что многие тонкости процессов верификации проектов, контроля и построения тестов не раскрываются в деталях в публикациях и являются ноу-хау фирм разработчиков.

Известно, что большая половина всех временных и финансовых затрат при проектировании функционально- сложных СБИС типа СнК приходится на верификацию проектов и построение тестов контроля.

В докладе рассматривается проблема интеллектуального совмещения подходов к построению контролепригодных цифровых систем, методов верификации их проектов и контроля проектных решений с целью обеспечения свойств безотказности и высокой готовности.

1. Особенности применения метода функционального контроля

Задачу контроля уместно рассматривать как задачу, решаемую на всех этапах жизненного цикла изделия: на этапе проектирования необходимо исключить ошибки проектирования, на этапе производства – выполнить разбраковку готовых изделий, на этапе эксплуатации вовремя «заметить» отклонение в функционировании объекта. При проектировании задача контроля решается различными методами верификации проектов. Основная часть задач практической верификации выполняется на основе моделирования на разных этапах проектирования. Поэтому на каждом этапе для верификации необходимы тесты, гарантирующие полноту контроля. Задача развития методов и средств построения тестов контроля применительно к объектам, представленным на разных уровнях проектирования и в различных системах идентификации, продолжает оставаться актуальной как в теоретическом, так и в практическом плане. При этом с развитием технологии микроэлектроники ее актуальность возрастает.

Самым первым практически используемым методом контроля интегральных схем и устройств на их основе стал функциональный контроль, который основан на исследовании устройства на его рабочей скорости. Данный вид контроля применялся, начиная с 1958 года, когда была создана первая интегральная схема [4]. Применение функционального контроля ограничивалось недостаточным числом контрольных точек, требованием большого числа входных воздействий. В дальнейшем по мере увеличения размерности объектов контроля проявилась низкая эффективность функционального контроля, который не только не позволяет практически обеспечить требуемую полноту контроля, но и не дает возможность определить ее количественную меру. К примеру, для функционального

контроля устройства сравнения двух 32-разрядных двоичных чисел требуется 2^{64} проверок. Если положить, что время выполнения одной проверки равно одной микросекунде, то для функционального контроля данного простого объекта потребуется $T=2^{64} \cdot 10^{-6} / 3600 \cdot 24 \cdot 365$, что примерно равно 585 годам. Современные проекты по своей структуре в миллионы раз объемнее и функционально сложнее. Поэтому на определенном этапе стала понятна ограниченность области применения функционального контроля. Усилия были перенаправлены на тестовый контроль, при котором вместо контроля функций стало контролироваться наличие некоторой неисправности объекта, что оказалось существенно более эффективным. К примеру, для 4 и 32-разрядных компараторов, схемы которых приведены на рис. 2, в программной системе [5] построены тесты контроля длиной 11 наборов (для 4-разрядного компаратора) и 69 наборов (для 32-разрядного) с полнотой контроля 100% в классе константных неисправностей, что позволяет тестовый контроль эффективно проводить в режиме реального времени. Но такой подход потребовал разработки структурных тестов, что оказалось для структурных схем объектов современной сложности достаточно сложной проблемой.

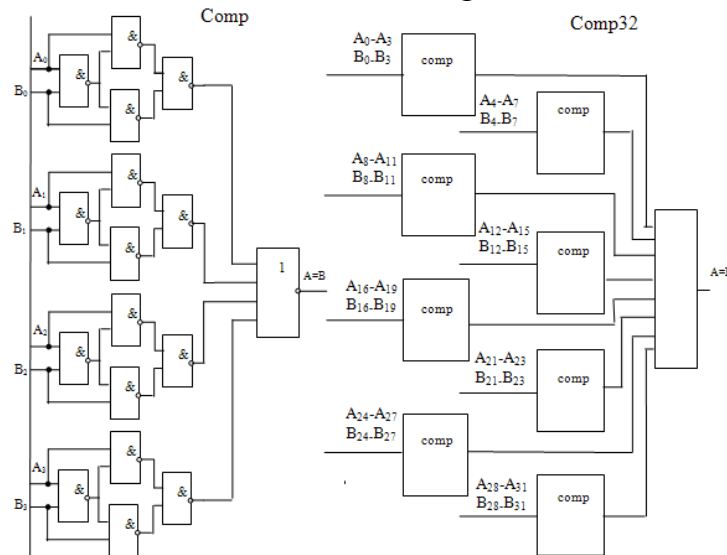


Рис. 2. Схемы сравнения 2-х чисел а – четырехразрядных; б – 32-разрядных

2. Состояние проблемы тестового контроля

Задача построения тестов цифровых устройств на всем интервале развития интегральной схемотехники является одной из наукоемких проблем, которые несмотря на прилагаемые усилия до настоящего времени не получили эффективного ни теоретического, ни практического решения. Задача алгоритмического (направленного) построения тестов принадлежит к классу NP- трудных проблем, так как ее решение для последовательной структуры вследствие зачастую необходимого полного перебора требует больших вычислительных затрат. Разработан ряд практически ориентированных подходов направленного и случайного построения тестов. Имеются программные средства, применяемые на практике. Но методы и средства направленного построения тестов, которые гарантируют построение теста, если

он существует, не нашли широкого практического применения из-за требования больших вычислительных ресурсов, ограниченных размеров объекта тестирования, и область их эффективного применения ограничена классом рассматриваемых неисправностей и объемом объекта контроля. Методы случайного построения тестов на основе моделирования неисправностей не дают желательного покрытия неисправностей для схем с памятью.

Подойти к решению задачи построения тестов для контроля СБИС оказалось возможным на основе идентификации объекта на начальных этапах проектирования, когда имеется некоторое поведенческое описание или описание объекта на уровне межрегистровых передач, которое содержит существенно меньшее число базовых примитивов, чем на структурном уровне.

В докладе рассматривается задача иерархического построения тестов контроля объекта, представленного на языке VHDL на уровне RTL. Рассматриваются функциональные неисправности, соответствующие реальным неисправностям структур, представляющих собой механизмы реализации операторов VHDL- кода объекта. Приводится один подход к построению функциональных неисправностей. Построение иерархических тестов основывается на введении функциональной неисправности в исходный код описания объекта контроля и сведении задачи построения теста для данной функциональной неисправности к решению задачи выполнимости некоторой системы КНФ функций разрешения [6]. При построении тестов в основу положена модель неисправности константного типа на структурном уровне и ее трансформация в модель функциональной неисправности.

На определенном этапе развития цифровой схемотехники считалось, что модель константной неисправности покрывает более 80 процентов всех дефектов, которые необходимо исследовать. Кроме того, предполагалось, что тест, покрывающий все одиночные неисправности константного типа, покрывает большинство множественных неисправностей. Однако с уменьшением размеров элемента, переходом в субмикронный диапазон, применением МОП – технологии, увеличением функциональной сложности интегральных схем существенно изменилось отношение к проблеме построения тестов. Утверждается новая тенденция в области построения тестов, которую можно сформулировать следующим образом: с точки зрения повышения надежности, обеспечения отказоустойчивости экономически более эффективно разрабатывать и применять тесты для неисправностей расширенного класса, так как тесты для неисправностей константного типа, разработанные для цифровых устройств на разных уровнях идентификации, не обеспечивают удовлетворительной полноты покрытия. Поэтому активно развиваются методы построения тестов для контроля неисправностей обрыва, замыкания, задержки (на логическом уровне), неисправностей типа «постоянно закрытый транзистор – ПЗТ» и др. (на переключательном уровне), которые применяются совместно с методами функционального контроля и контролепригодного проектирования. Тестовое диагностирование неисправностей задержки, в том числе методами статического статистического или детерминированного временного анализа

является важным методом обеспечения функциональной устойчивости проектов.

3. ABV как новый подход к реализации идеи функционального контроля

В последнее время методы и средства функционального контроля вышли на новый этап развития на основе широкого применения технологии утверждений ABV (assertion based verification). Функциональный контроль проводится как в течение рабочего функционирования устройства в режиме on-line, так и в режиме off-line с применением также и специально разработанных тестов с целью обеспечения необходимой полноты контроля путем совмещения функционального и тестового контроля.

Импульсом к появлению и применению этой технологии стало использование для описания и разработки изделий микроэлектроники языков высокого уровня VHDL, Verilog, в которых уже с самого начала были заложены некоторые возможности контроля. Так, в язык VHDL введен последовательный оператор *assert*, который позволяет проверить определенные результаты функционирования проектируемого объекта и при появлении некорректного поведения сообщить пользователю о результатах проверки. Оператор утверждения *assert* проверяет истинность определенного условия и сообщает об ошибке, если условие не выполняется. В форме Бэкуса-Наура оператор *assert* записывается так:

```
assertion_statement ::= [label:] assertion;  
assertion ::= assert condition [report expression] [severity expression]
```

Выполнение оператора состоит в вычислении булева выражения, определяющего условие. Если выражение имеет результат false (ложь), то это означает, что произошло нарушение утверждения. Для идентификации ситуации в структуре оператора *assert* имеется возможность указать строковое сообщение [report expression], которое будет включено в таблицу сообщений об ошибках, формируемых оператором, или по умолчанию будет опубликовано сообщение *Assertion violation*, свидетельствующее о нарушении утверждения. Кроме того, оператор может включать обозначенную пользователем категорию ошибки [severity expression], что может помочь в дальнейшем при анализе результатов моделирования.

Наиболее частое применение оператора *assert* – проверка соответствия входных сигналов, поступающих через порты, заданным требованиям или соответствие ограничениям на настроечные константы *generic*. Например, проверяется время предустановки информационного сигнала относительно фронта синхросигнала, времени его удержания, разрядность входных данных и т.п. В листинге 1 рассмотрен пример контроля корректности входных сигналов элемента памяти.

Листинг 1. Пример кода на языке VHDL

```
library IEEE;  
use IEEE.STD_LOGIC_1164.all;  
entity D_trig is  
  generic (delay1:time); (delay2:time);
```

```

port(d, clk: in std_logic;
     Q: out std_logic;
     nQ: out std_logic);
begin
  assert (clk'event and d'stable(delay1)) report "In D_trig input error" severity error;
  assert (clk='1'and d'stable(delay2)) report "In D_trig input error" severity error;
end D_trig;

```

Оператор *assert* в языке VHDL используется с успехом уже много лет. Но конструкция *assert* может содержать только булевское выражение, и варианты реакции на выполнение условия весьма ограничены. Поэтому практическое использование этих конструкций в языке VHDL представляет собой задачу, сложность которой соизмерима со сложностью реализация самого проекта.

Для проектов со сложным поведением написание ассертов при использовании только стандартных возможностей Verilog или VHDL является достаточно трудным. Существовавшие технические средства (системы моделирования) поначалу позволяли применять конструкцию *assert* практически только при написании программ Test-Bench. На данный момент расширяются возможности операторов *assert* при стандартизации и строгой формализации правил их создания. Утверждения позволяют улучшить верификационную методологию и в результате обеспечить более быструю реализацию проектов. Современные реализации ABV, а тем более методология, расширенная до понятия Assertion-Based Design (ABD), являются более строго формализованными, структурированными приложениями утверждений, хорошо приспособленными и ориентированными на задачи верификации. Ассерты позволяют легко контролировать корректное поведение внутренних сигналов тестируемого устройства. Их можно вставлять в архитектурные пары *entity architecture* VHDL-описаний, аналогично можно использовать контрольные модули (*checkers*) из библиотеки— Open Verification library (OVL) фирмы Accellera.

4. Задачи в области развития методов контроля

Классический подход к функциональному контролю, как было указано выше, оказался малоэффективным по причине большого пространства области исследований, недостаточного количества точек наблюдения за результатом контроля. Поэтому функциональный контроль не может в полной мере обеспечить решение задачи контроля. Высокая размерность разрабатываемых цифровых систем и невысокие достижения в области построения тестов структурного контроля делает все менее эффективными системы тестового диагностирования, требующие внешнего диагностического оборудования и его применения в режиме *off-line*. Можно сказать, что в случае построения «систем на кристалле» *off-line* контроль с привлечением внешнего диагностического оборудования оказался практически не эффективным. В условиях высоких требований по надежности, актуальности сокращения сроков запуска изделий в производство системы контроля применяются совокупно с различными методами контролепригодного проектирования, с применением средств встроенного самотестирования. Средства самотестирования

интегрируется в состав активной инфраструктуры непосредственно на кристалле и позволяют организовать контроль посредством ограниченного числа точек наблюдения. Таким образом, появляется возможность применять методы построения специально разработанных тестов для функциональных блоков в рамках системы контроля СнК. Интеллектуальное совмещение подходов к построению контролепригодных систем и методов их контроля позволяет обеспечивать свойства безотказности и высокой готовности современных функционально-сложных цифровых объектов. В качестве примера можно привести включение во входную псевдослучайную последовательность векторов специально разработанных тестовых блоков при реализации методов компактного тестирования (например, метода сигнатурного анализа), когда анализ результатов контроля осуществляется на основе сравнения полученной сигнатуры с эталонной. Применение методов контролепригодного проектирования позволяет упростить задачу построения тестов путем преобразования схем с памятью в комбинационные.

Верификация проектов СнК требует применения всего арсенала методов контроля, т.к. СнК – это функционально-сложные системы, разнообразие и диапазон сложности которых зависит от характера объекта, где используется встроенная система управления: управление заводом-автоматом, транспортной системой, интеллектуальным кампусом, сложным медицинским оборудованием, летательным аппаратом и т.д. Здесь решается задача непосредственного взаимодействия с физическими объектами и процессами. Большое количество элементов, ограниченное количество контрольных точек в таких объектах и, как следствие, резкое снижение управляемости и наблюдаемости внутренних переменных осложняет проблему верификации проекта и построения системы контроля. Следует заметить, что применение отработанных в плане проектной корректности многократно используемых блоков (IPs) при проектировании современных цифровых систем на кристалле не решает и даже существенно не упрощает задачу верификации проекта в целом. Объединение отлаженных отдельных функциональных блоков не дает никакой гарантии корректности полученного функционала вследствие возникающих несогласованностей, которые должны быть найдены и устранены на этапе верификации RTL- проекта.

Вместе с ренессансом функционального контроля, вызванным появлением ABV технологии верификации, появилась возможность применения тестового контроля устройств и блоков системы (рис. 3) путем повышения ее управляемости и наблюдаемости в режиме контроля. В настоящее время ABV становится господствующей методологией верификации систем на кристалле. В последние годы можно наблюдать появление большого числа языков ассерций для разных средств верификации проектов. Большая часть из них привязана к определенной программе моделирования и для его использования требует определенной непростой адаптации к применению.

Чтобы упростить разработчикам процесс верификации проектов, создана библиотека Open Verification Library (OVL), которая обеспечивает возможность

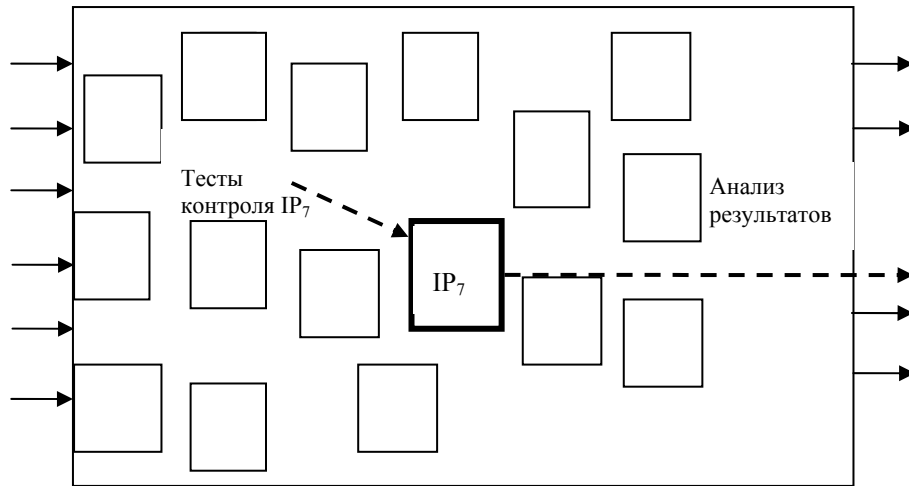


Рис. 3. Схема контроля IP-блоков в СнК

контроля ряда свойств объектов и процессов, которыми должен обладать проект. Например, правильность последовательности выполнения обозначенных условий; состояние, которое не должно возникать в системе; нахождение некоторых данных внутри заданного диапазона; контроль за состоянием данного (постоянное значение, четность или изменение внутри определенного диапазона); изменение по возрастанию или убыванию внутри диапазона значений переменной; корректность выполнения протокола передачи данных; правильная работа стека памяти (LIFO, FIFO) и т.д. Библиотека OVL содержит готовые к применению достаточно длинные коды, которые возможно и не будут учитываться при синтезе объекта и влиять на его результаты, а будут использоваться только при верификации проекта. Поэтому остается выполнить правильный выбор ассерций из библиотеки и их применение для контроля проекта.

Зачастую ассерции встраиваются и в модели и в проекты. Так, повсеместное распространение техники контроля данных по четности кода (процедура сложения по модулю два) привело к встраиванию контроля и в модель, и в реализацию проекта. Современные возможности кремниевой технологии позволяют реализовать «в кремнии» необходимые ассерции и, таким образом, повысить полноту контроля. Сложность метода ABV в том, что введение в проект ассерций является достаточно сложной, ответственной и неформальной процедурой, которая отличается от уровня проектирования, от языка описания объекта, системы моделирования и используемой библиотеки ассерций.

Эффективность верификации в значительной степени зависит от того, насколько отличаются верификационные подходы, применяемые на разных этапах проектирования. Совмещению подходов способствует тенденция распространения методов верификации, основанных на утверждениях, на большинство этапов проектирования: от разработки спецификации до выпуска готовой продукции. Библиотека OVL поддерживается и расширяется, позволяет использование в System C, Verilog, VHDL [7].

Процесс проектирования функционально-сложной СБИС является

итерационным. От качества применяемых средств и маршрута проектирования зависит число итераций, а соответственно стоимость и сроки выхода на рынок. Большая размерность задачи верификации проектов активизировала исследования в области идентификации объектов и их верификации на ранних этапах и, в частности, на системном уровне проектирования (Electronic System Level – ESL). В маршруте проектирования СБИС появился уровень транзакций (Transaction Level Modeling - TLM), который находится ниже уровня системного представления и выше уровня RTL (рис. 4), и соответственно размерность задачи верификации модели TLM существенно ниже, чем верификация на уровне RTL. На уровне TLM можно отработать функциональные аспекты проекта без учета деталей схемной реализации. При проектировании на уровне ESL часто используются средства System C и TLM 2.0. Но отладка проектов остается в большой степени проблемой разработчика, т.к. нет универсальных средств верификации проектов. Становится очевидным необходимость применения моделей TLM. В работе [8] приведен пакет ABV, содержащий средства верификации проектов на уровне TLM, который содержит возможности, заложенные в библиотеке OVL, направлен на применение при верификации RTL проектов в System C и может эффективно применяться в смешанных RTL /TLM проектах.

Метод подключения утверждений к самому проекту находит применение не только при тестовом подходе, но и в методах формальной верификации [9].

Заключение

В докладе рассматриваются вопросы интеллектуального совмещения подходов к контролю функционально-сложных цифровых СБИС типа СнК, включающих методы контролепригодного проектирования, функционального и тестового контроля, начиная с верхних уровней проектирования.

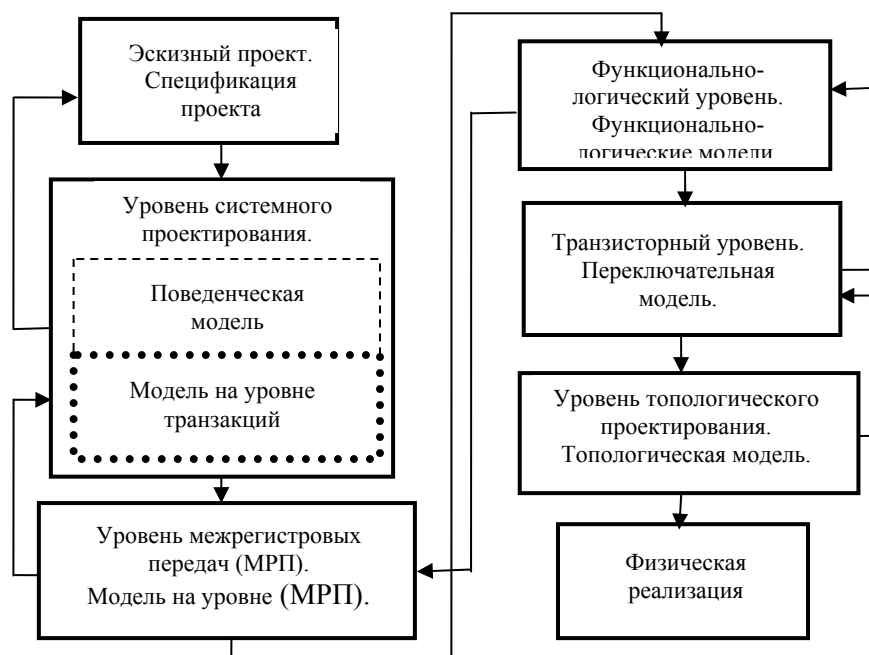


Рис. 4. Схема итерационного процесса проектирования

Список литературы

1. Transistor count in microprocessors – http://en.wikipedia.org/wiki/Transistor_count. – Дата доступа: 16.02.2016.
2. Kahng, A.B. Design technology productivity in the DSM era (invited talk) / A.B. Kahng. – Design Automation Conference. – 2001. – Proceedings of the ASP-DAC 2001. – Asia and South Pacific. – New York, NY, USA. – 2001. – ACM. – P. 443–448.
3. Турцевич, А. С. Развитие САПР микроэлектроники. САПР ОАО «ИНТЕГРАЛ» А. С. Турцевич, А. И. Белоус, С. В. Шведов. – МНПК «Современные информационные и электронные технологии». – Одесса, 27-31 мая 2013 г. – С.17-20.
4. Kilby, J. Integrated circuits invented by Jack Kilby / J. Kilby. – Texas Instruments, Dallas. – TX. – Sep.12, 1958. – Дата доступа: 18.09.2015. <http://www.ti.com/corp/docs/company/history/timeline/semicon/1950/docs/58ickilby.htm>.
5. Золоторевич, Л.А. Компьютерная инженерия в электронике: программная система для проектирования и обучения / Л.А. Золоторевич, О.М. Сидоренко, Д.И. Юхневич. – В сб. научных статей по материалам Международной научно-практической конференции «Образовательные технологии в подготовке специалистов». – Минск. – 2003. – С. 23-28.
6. Zolotorevich, L.A. Project verification and construction of superchip tests at the RTL level / L.A. Zolotorevich. – Automation and Remote Control. –USA. – NY. – Plenum Press 2013. – Vol. 74. – Issue 1. P. 113-122.
7. Wilcox, P. Professional Verification. A Guide to Advanced Functional Verification / P. Wilcox. – Kluwer Academic Publishers. – 2004.
8. Грушвицкий, Р. Проектирование в условиях временных ограничений: отладка проектов / Грушвицкий Р., Михайлов М. – Компоненты и технологии. – №6. – 2007.
9. Sohofi, H. System-level assertions: approach for electronic system-level verification / H. Sohofi, Z.Navabi. – IET COMPUTERS & DIGITAL TECHNIQUES 9(3). – 2014.

Золоторевич Людмила Андреевна, доцент кафедры электронных вычислительных машин Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, доцент, zolotorevichla@bsuir.by