

Организация процесса отладки проектируемых цифровых микроэлектронных систем

А.Д. Иванников

Институт проблем проектирования в микроэлектронике РАН, г. Москва

adi@ippm.ru

Аннотация — В работе рассматриваются методы отладки, то есть выявления и исправления ошибок в проектах цифровых систем, заданных схемой технических средств и текстом программного или микропрограммного обеспечения. Рассматривается отладка на изготовленном макете технических средств и на программной модели цифровой системы, а также на сочетании макета отдельных блоков и программной модели остальной части системы. Анализируются методы организации отладочных режимов для всех рассматриваемых методов. Указывается, что для цифровых систем на кристалле наиболее эффективным методом является отладка проектов на программной модели.

Ключевые слова — проектирования цифровых систем, отладка проектов, организация отладочных режимов

I. ВВЕДЕНИЕ

Еще в 90-ые году во многих работах указывалось на перспективность использования информационных технологий в различных областях современной человеческой деятельности: коммуникациях, образовании, науке и технике [1-4]. В последующие тридцать лет информационные технологии постоянно развивались и находили применение во всех областях человеческой деятельности [5-9].

Применение информационных технологий возможно при наличии средств вычислительной техники и встраиваемых цифровых микроэлектронных систем управления, разработка которых опять же требует применения систем автоматизированного проектирования [10-12].

При проектировании цифровых систем, результатом которого является схема соединения блоков и элементов цифровых систем, а также текст программного или микропрограммного обеспечения, важным этапом разработки является отладка, то есть выявление и исправление ошибок проектирования [13, 14]. При постоянно возрастающей сложности современных цифровых систем этап отладки схемы технических средств, программного обеспечения, а также их совместной работы может занимать существенную часть всего этапа проектирования.

II. СОСТАВЛЯЮЩИЕ ПРОЦЕССА ОТЛАДКИ

Для проведения отладки проектов цифровых систем необходимы следующие составляющие.

1. Объект отладки, позволяющий по заданному входному воздействию определить реакцию на выходе. Объектом отладки может являться как макет или прототип цифровой системы, так и ее программная модель.

2. Конечный набор конечных по времени входных воздействий, назовем их отладочными тестами, которые подаются на объект отладки во время отладочных экспериментов. Набор отладочных тестов должен обладать свойством полноты, то есть из правильности реакции объекта отладки на все тесты набора должна следовать правильность его работы при любых допустимых входных воздействиях.

3. Критерий определения правильности выходной реакции объекта отладки на поданное входное воздействие.

4. Возможность локализации ошибок в цифровой системе при неправильной реакции объекта отладки на заданное входное воздействие.

5. Средства для внесения исправлений в объект отладки и повторения отладочного эксперимента.

Проведем анализ возможных методов отладки цифровых систем на этапе проектирования, рассмотрим поочередно составляющие процесса отладки.

III. ОБЪЕКТ ОТЛАДКИ И МЕТОДЫ ОТЛАДКИ ЦИФРОВЫХ СИСТЕМ НА ЭТАПЕ ПРОЕКТИРОВАНИЯ

Отладка цифровой системы может быть проведена как на изготовленном макете цифровой системы, так и на некоторой модели последней [15, 16]. В соответствии с различным представлением разработанной цифровой системы можно выделить четыре метода отладки, возможных на этапе проектирования, а именно:

- метод полного макетирования;
- метод частичного макетирования при использовании типового ядра технических средств;
- метод моделирования на ЭВМ;
- метод моделирования на ЭВМ с использованием реальных блоков технических средств.

При методе полного макетирования начало отладки технических средств и программно-микропрограммного обеспечения и их совместной работы возможно только после изготовления макета,

что требует материальных затрат и занимает при современной структуре разрабатывающих организаций достаточно длительные сроки.

В процессе анализа работы цифровой системы и локализации ошибок проектирования принципиальной схемы и программно-микропрограммного обеспечения необходимо следить за изменениями содержимого регистров системы и ячеек памяти, а также за логическими сигналами на шинах и линиях технических средств, иметь возможность останавливать и возобновлять выполнение программы или микропрограммы, задавать и модифицировать содержимое внутренних регистров системы и ячеек ЗУ. Если слежение за изменениями логических сигналов на внешних выводах системы в макете может быть осуществлено с помощью логических анализаторов [17], то контроль за изменениями содержимого внутренних регистров и ячеек ЗУ, не имеющих внешних выводов, останов и повторный запуск программы или микропрограммы, задание и изменение содержимого внутренних регистров и ячеек ЗУ, то есть отладочные режимы, невозможны, если не предусмотрены специальные средства внутри макета. Эти операции могут быть осуществлены только при наличии в макете дополнительных аппаратных затрат, специального программного обеспечения и подключения интерактивных устройств ввода-вывода, например, системы JTAG [18, 19], которая часто предусматривается в разрабатываемой цифровой системе. Встраиваемые во цифровую систему дополнительные средства во многом облегчают процесс отладки, но не в полной мере. Кроме того, разработка дополнительных аппаратно-программных средств для эффективной организации отладочных режимов на макете является нетривиальной.

Отладка цифровой системы на макете технических средств при конкретных значениях динамических параметров блоков не гарантирует работоспособность технических средств системы при любых допустимых по ТУ значениях динамических параметров блоков.

Отладка макета совместно с управляемым объектом в реальном времени часто затруднена в связи с невозможностью создания реальных, например, аварийных ситуаций на объекте. Такая отладка может быть осуществлена путем совместного моделирования на ЭВМ цифровой системы и поведения управляемого объекта в одинаковом масштабе времени.

Метод частичного макетирования предполагает разделение технических средств разрабатываемой системы на типовое ядро (центральную часть), одинаковое для данного класса цифровых систем, и часть, например, схемы сопряжения с конкретным объектом управления, присутствующую только в разрабатываемой системе. Типовое ядро, включающее аппаратно-программные средства организации отладочных режимов, может быть изготовлено и отлажено один раз, после чего в сочетании с макетом нетиповой части используется при отладке различных цифровых определенного класса.

Так, во многих простейших цифровых системах управления (контроллерах) основная по объему центральная часть (процессорный блок, ЗУ, сопряжение с типовыми внешними устройствами, типовые каналы ввода-вывода) является типовой, в связи с чем метод частичного макетирования для таких систем применим в большей степени, чем для цифровых систем других классов [20, 21].

В случае микропрограммируемых систем, для которых характерно большое разнообразие структур и архитектур, в типовое ядро включаются блок микропрограммного управления и ЗУ микропрограмм большой разрядности, а блок обработки данных вместе со схемами сопряжения с обслуживаемым объектом макетируется в соответствии со схемой технических средств разрабатываемой системы. При этом макетируемая часть системы существенно больше, чем в случае систем на основе вычислителей с фиксированной системой команд.

Ввиду фиксированной организации блока микропрограммного управления типового ядра метод частичного макетирования не позволяет получить объекты отладки для всего разнообразия микропрограммируемых цифровых систем.

В случае многопроцессорных цифровых систем метод частичного макетирования может быть также использован. В этом случае макетируемая часть представляет собой нестандартные схемы сопряжения с обслуживаемым объектом и, возможно, обрабатывающие блоки, аппаратно реализующие определенные операции, доля которых в общем объеме технических средств, как правило, не велика. При нетиповой организации многопроцессорной системы применение метода частичного макетирования затруднено в связи с невозможностью выделения существенного по объему типового ядра технических средств.

Таким образом, характеризуя метод частичного макетирования в целом, можно сказать следующее.

1. Метод требует меньших затрат на изготовление макета, чем метод полного макетирования.

2. Метод может быть эффективно использован при проектировании систем, имеющих существенное по объему типовое ядро технических средств. Применение метода при нетиповой структуре и архитектуре разрабатываемой цифровой системы невозможно.

3. Для метода частичного макетирования характерен ряд недостатков, свойственных методу макетирования вообще, а именно:

- трудоемкость внесения изменений в принципиальную схему в процессе отладки;

- невозможность гарантировать работоспособность технических средств при любом допустимом разбросе задержек блоков;

- трудность отладки макета системы совместно с обслуживаемым объектом в реальном времени в свя-

зи с невозможностью искусственного создания ряда ситуаций на объекте.

Метод моделирования на ЭВМ предполагает использование в качестве объекта отладки программной модели проектируемой системы [22-26]. Этот метод является универсальным в том смысле, что модель может быть получена для цифровой системы любой структуры и архитектуры, в том числе для нетиповых микропрограммируемых и мультипроцессорных систем.

Проверка работоспособности технических средств цифровой системы для всех допустимых по ТУ значениях динамических параметров блоков одновременно возможна только методом моделирования на ЭВМ. Этот метод принципиально позволяет проводить отладку путем совместного моделирования цифровой системы и обслуживаемого объекта во всех возможных режимах работы последнего. Использование метода моделирования сокращает сроки и стоимость процесса отладки в связи с отсутствием необходимости изготовления макета до окончания отладки проекта.

Недостатком метода моделирования является меньшее относительно макета быстродействие программной модели отлаживаемой системы. Так, при моделировании на уровне системы команд на эмуляцию одной команды процессорного элемента цифровой системы приходится от нескольких десятков до нескольких сотен команд инструментальной ЭВМ. При функционально-логическом моделировании системы на уровне блоков затраты машинного ресурса, как правило, еще больше.

Тем ни менее, использование машинных моделей цифровых систем в качестве объекта отладки весьма эффективно вследствие своей универсальности относительно типа отлаживаемых систем, отсутствия необходимости изготовления макета, возможности параллельного анализа нескольких технических решений.

Применение **метода моделирования на ЭВМ с использованием реальных блоков** обусловлено следующим. Как указывалось выше, недостатком использования машинных моделей цифровых систем является большое время моделирования. При моделировании на функционально-логическом уровне обычно каждому блоку соответствует своя программная модель, а модель технических средств в целом генерируется путем комбинации моделей блоков в соответствии с заданной схемой соединения блоков. Модели цифровых блоков сверхвысокой степени интеграции сложны в разработке, занимают большой объем памяти и требуют больших затрат машинного времени при моделировании. В связи с этим известно использование автоматизированных рабочих мест для проектирования, в которых для отладки используется функционально-логическое моделирование на уровне блоков, а центральное вычислительное ядро подключается к системе физически. При этом, однако, возникает

две проблемы: отсутствие в процессе отладки доступа к внутренним регистрам физически подключаемого блока и существенно разные скорости работы физического блока и программной модели.

Таким образом, метод моделирования на ЭВМ с использованием реальных блоков позволяет уменьшить затраты времени на моделирование. Тем ни менее, этот метод не гарантирует работоспособность отлаживаемой системы при любых допустимых значениях динамических параметров аппаратно подключаемых блоков. Кроме того, поскольку скорость работы объекта отладки в этом случае определяется моделирующей частью и существенно меньше скорости работы реальной системы, трудно гарантировать работоспособность системы при реальном режиме работы.

Метод моделирования с использованием реальных блоков следует рассматривать как расширение метода моделирования.

IV. ЛОКАЛИЗАЦИЯ ОШИБОК В ПРОЕКТАХ ЦИФРОВЫХ СИСТЕМ

Единственным универсальным методом локализации ошибок как в схеме технических средств, так и в программно-микропрограммном обеспечении на этапе отладки проекта является анализ разработчиком внутренних переменных: содержимого ячеек памяти и регистров, а также логических сигналов на шинах и линиях цифровой системы в процессе ее работы [27]. Избирательный и детальный анализ указанных внутренних переменных в сочетании со знанием требуемого функционирования разрабатываемой системы позволяет разработчику предположить и проверить наличие конкретных ошибок в проекте.

Таким образом, возможность удобного и детального анализа изменения внутренних переменных цифровой системы является основным средством локализации ошибок проектирования.

При использовании **метода полного макетирования** слежение за логическими сигналами на шинах системы и выводах блоков осуществляется с помощью логических анализаторов различных типов, в том числе встроенных в проектируемую систему. Эти приборы позволяют проверить логические диаграммы узлов принципиальной схемы при возникновении заданных условий.

Слежение за изменением содержимого внутренних регистров блоков и ячеек ЗУ, а также организация отладочных режимов программно-микропрограммного обеспечения при методе полного макетирования осуществляется с помощью загружаемых в ОЗУ системы отладочных мониторов. Однако, без дополнительных аппаратных средств организации отладочных режимов выполнение программного обеспечения осуществляется медленнее, чем в реальной аппаратуре. Отладочные мониторы, ориентированные на слежение за содержимым ячеек ЗУ и регистров типовой части, не позволяют анализировать со-

стояние регистров нетиповых схем сопряжения с внешними устройствами.

В случае нетиповых микропрограммируемых и мультипроцессорных систем использование типовых отладочных мониторов невозможно. Создание же мониторов для каждой конкретной системы дорого и является нетривиальной задачей.

Метод частичного макетирования предполагает использование в качестве типового ядра специальных аппаратно-программных отладочных комплексов, которые представляют собой универсальные цифровые системы с большим объемом ЗУ (включая внешнее ЗУ на магнитных носителях), широким набором внешних устройств, в том числе логических анализаторов, и резидентной системой разработки и отладки программного обеспечения. В отладочных комплексах предусматриваются специальные аппаратные и программные средства организации отладочных режимов, позволяющие выполнять останов в нужных точках программы или микропрограммы, следить за изменениями и модифицировать содержимое ячеек ЗУ и регистров типового ядра системы.

Существенно расширяет возможности отладочных комплексов наличие внутрисхемных эмуляторов – аппаратных блоков, включающих процессор, архитектурно эквивалентной используемому в отлаживаемой системе, и аппаратно реализованные схемы слежения за логическими сигналами на шине системы, которые подключаются к макету технических средств отлаживаемой системы вместо микропроцессора. Использование внутрисхемных эмуляторов позволяет проводить совместную отладку технических средств и программного обеспечения в реальном времени, а также подключать макет цифровой системы к обслуживаемому объекту для совместной отладки. При этом возможно поэтапное наращивание в отладка технических средств путем последовательной передачи функций от плат типового ядра, расположенного в отладочном комплексе, добавляемым платам макета. В сочетании с логическими анализаторами аппаратно-программные отладочные комплексы являются мощным средством отладки цифровых систем.

В случае цифровых систем, использующих микропрограммирование, отладочный комплекс, как правило, включает ОЗУ, моделирующее ПЗУ микрокоманд, а также универсальную ЭВМ для управления работой комплекса. Наличие типового блока микропрограммного управления позволяет создать системное программное обеспечение, обеспечивающее организацию отладочных режимов выполнения микропрограмм и слежения за логическими сигналами на шине системы, что в случае отладки микропрограммируемых систем особенно необходимо. Однако, такие комплексы могут использоваться при отладке микропрограммируемых цифровых систем только со стандартной структурой шин и стандартным протоколом обмена информацией по шинам, а также типовой микроархитектурой, что ограничивает применение отладочных ком-

плексов для микропрограммируемых систем типовыми или близкими к типовым системами.

В случае мультипроцессорных систем аппаратно-программные отладочные комплексы также используются. Ряд известных аппаратно-программных отладочных комплексов имеют внутрисхемные эмуляторы, позволяющие одновременно следить за работой двух и более процессорных блоков, входящих в сосредоточенную мультипроцессорную систему.

Применение аппаратно-программных отладочных комплексов, используемых одновременно как стандартное ядро с встроенными средствами организации отладочных режимов и как автоматизированное рабочее место для разработки программного и, в ряде случаев, микропрограммного обеспечения является одним из основных факторов, обуславливающих широкое применение метода частичного макетирования при отладке цифровых систем.

При использовании для отладки **метода моделирования** слежение как за изменениями содержимого ячеек ЗУ и внутренних регистров всех блоков системы, так и за логическими значениями сигналов во всех узлах схемы не вызывает трудностей, так как они представляются переменными в моделях. Удобство задания любых отладочных режимов, условий вывода информации и самой информации о функционировании модели определяется организацией системы моделирования и языком управления ее работой.

В последние годы в системах моделирования цифровых систем нашли применение методы работы с большими данными, то есть не выделение наиболее значимых переменных при моделировании и поиске ошибок, а запоминание при моделировании системы на уровне регистровых передач всех данных, не зависимо от их значимости. При этом при обнаружении ошибки в каких-либо переменных: логических сигналах или содержимых регистров, - возможен обратный просмотр хода моделирования для обнаружения причины ошибки.

При применении **метода моделирования с использованием реальных блоков** в качестве последних необходимо выбирать те блоки, слежение за внутренними регистрами которых не является обязательным для отладки.

V. ВНЕСЕНИЕ ИСПРАВЛЕНИЙ В ПРОЕКТ

Внесение исправлений в текст программно-микропрограммного обеспечения, находящегося в ОЗУ, осуществляется обычными способами и трудностей не вызывает. Внесение же изменений в схему технических средств при методе полного или частичного макетирования требует физического изменения схемы: изменения подключения блоков к узлам схемы, удаления, добавления или модификации блоков. Это требует определенных временных затрат, особенно при существенных изменениях в схеме. Внесение изменений в моделируемую схему технических средств производится путем редактирования машин-

ной модели, что осуществляется разработчиком в интерактивном режиме и практически не требует временных затрат.

VI. ОТЛАДОЧНЫЕ ТЕСТЫ И ОПРЕДЕЛЕНИЕ ПРАВИЛЬНОСТИ ВЫХОДНОЙ РЕАКЦИИ

Выбор набора отладочных тестов и определение правильности выходной реакции является общей задачей для всех методов отладки цифровых систем.

Задача выбора отладочных тестов существенно отличается от задачи выбора проверяющих и диагностических тестов цифровой аппаратуры, используемых на этапе производства и обслуживания, тем, что при отладке цифровых систем на этапе проектирования отсутствуют модели возможных ошибок. Задача выбора отладочных тестов как для технических средств, так и для цифровых систем в целом по своей постановке в определенной степени близка задаче синтеза тестовых примеров для отладки программ. Отладочные тесты должны выбираться в первую очередь исходя из функций, выполняемых цифровой системой и определенных техническим заданием на ее разработку, а также, возможно, структуры разработанной системы [28-30].

Независимо от метода выбора тестов при отладке необходимо определять правильность или ошибочность реакции объекта отладки. Наиболее удобным, но не всегда возможным в практике проектирования является случай, когда разработчик в точности знает требуемую реакцию объекта отладки на любой тест. Часто разработчик заранее в точности не знает правильную реакцию на отладочный тест, но после получения результата и, возможно, анализа промежуточных данных может сказать, соответствует ли полученная реакция объекта отладки требуемой. В любом случае возможность разработчика определить правильность реакции объекта отладки на поданное входное воздействие является необходимым условием проведения отладки вообще.

VII. ЗАКЛЮЧЕНИЕ

Таким образом, на основе анализа методов отладки цифровых систем можно сказать, что потенциально только метод моделирования на ЭВМ является универсальным относительно структуры и архитектуры отлаживаемых цифровых систем. Этот метод наиболее широко применяется при разработке современных цифровых систем в микроэлектронном исполнении. Однако, и другие методы также находят применение.

ЛИТЕРАТУРА

- [1] Юсупов Р.М. Информатизация и наука // Проблемы информатизации. 1994. № 1-2. С. 22-28.
- [2] Тихонов А.Н., Иванников А.Д. Информатизация российского образования и общества в целом // Международное сотрудничество. 1997. № 4. С. 1.
- [3] Горохов Ю.П., Жевнов И.И., Иванников А.Д., Татарников Ю.А. Основные направления программы информатизации высшего образования // Педагогическая информатика. 1993. № 2. С. 38.
- [4] Стемповский А., Шилин В. КМОП-фотодиодные СБИС. Перспективная элементная база однокристалльных систем приема и обработки информации // Электроника: Наука, технология, бизнес. 2003. № 2(44). С. 14-21.
- [5] Тихонов А.Н., Иванников А.Д., Цветков В.Я. Образовательные услуги как инструмент качества образования // Международный журнал прикладных и фундаментальных исследований. 2009. № 3. С. 94-96.
- [6] Абрамов А.Г., Булакина М.Б., Иванников А.Д., Кривошеев А.О., Шмелькова Л.В. Система дополнительного профессионального образования Российской Федерации: ключевые статистические показатели по результатам автоматизированного сбора данных // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2014. № 1. С. 133-144.
- [7] Иванников А.Д. Тематические интернет-порталы как средство агрегации электронного контента в заданной предметной области // Информационные технологии. 2014. №3. С. 43-48.
- [8] Стемповский А.Л., Амербаев В.М., Соловьев Р.А. Принципы рекурсивных модулярных вычислений // Информационные технологии. 2013. № 2. С. 22-27.
- [9] Климов А.В., Левченко Н.Н., Окунев А.С., Стемповский А.Л. Суперкомпьютеры, иерархия памяти и потоковая модель вычислений // Программные системы: теория и приложения. 2014. Т. 5. № 1(19). С. 15-36.
- [10] Tiunov I.V., Lipatov I.A., Zheleznikov D.A. Digital circuits resynthesis approach for FPGAs based on logic cell with built-in flip-flop. // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2019. № 3. С. 33-36.
- [11] Иванников А.Д., Стемповский А.Л. Математическая модель отладки проектов сложных цифровых схем и микросистем на основе представления последних в виде семейства стационарных динамических систем // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2014. № 2. С. 123-128.
- [12] Иванников А.Д. Теоретические основы выбора множества отладочных тестов цифровых систем на основе алфавита выполняемых функций // Информационные технологии. 2019. Т.25. № 11. С. 657-662.
- [13] Слинкин Д.И. Анализ современных методов тестирования и верификации проектов сверхбольших интегральных схем // Программные продукты и системы. 2017. Т. 30. № 3. С. 401-408.
- [14] Иванников А.Д. Формирование отладочного набора тестов для проверки функций цифровых систем управления объектами // Мехатроника, автоматизация, управление. 2017. Т. 18. № 12. С. 795-801.
- [15] Ilias Alexopoulos. How to Debug Embedded Systems // <https://www.EDN.com>, Dec. 11, 2012.
- [16] Абрамов Е.М., Егоров А.В., Козлов А.О., Поперечный П.С., Путря Ф.М., Фролова С.Е. Выбор платформ прототипирования для СФ-блоков и подсистем СНК // Вопросы радиоэлектроники. 2017. № 8. С. 76-83.
- [17] Зотов В. Инструментальные средства разработки и отладки цифровых устройств и встраиваемых микропроцессорных систем, проектируемых на основе ПЛИС FPGA фирмы XILINX серии Kintex-7 // Компоненты и технологии. 2012. № 4 (129). С. 124-132.
- [18] Жезлов К.А., Колбасов Я.С., Козлов А.О., Николаев А.В., Путря Ф.М., Фролова С.Е. Автоматизация процесса создания тестовых окружений, обеспечивающая сквозной маршрут разработки, верификации и исследования СФ-блоков и СНК // Проблемы разработки пер-

- спективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 46-53.
- [19] X. Hu, Y. Jin and Z. Li. A Parallel JTAG-based Debugging and Selection Scheme for Multi-core Digital Signal Processors // *2018 IEEE International Conference of Safety Production Informatization (ICSPI)*. Chongqing, China. 2018. Pp. 527-530.
- [20] Иванов С.А., Орешкин Д.М. Аппаратно-программный комплекс для разработки и отладки электронных систем на цифровых микропроцессорных модулях // *Новая наука: от идеи к результату*. 2016. № 8-1 (96). С. 27-29.
- [21] Стрелец А.И., Протопопова Ю.Д., Иванников В.С. Универсальная система для тестирования цифровых устройств // *Научный журнал*. 2018. № 5 (28). С. 41-44.
- [22] Nguen M.D. Hardware/software formal co-verification using hardware verification techniques // *Fourth Int. Conf. on Communications and Electronics (ICCE)*. 2012. Pp. 465-470.
- [23] Romanov A.Yu., Ivannikov A.D., Romanova I.I. Simulation and synthesis of network-on-chip by using NOCSIPM HDL library. 2016 IEEE 36th International Conference on Electronics and Nanotechnology, ELNANO 2016 – Conference Proceedings. 36. 2016. Pp. 300-303.
- [24] Стемпковский А.Л., Тельпухов Д.В., Соловьев Р.А., Соловьев А.Н., Мячиков М.В. Моделирование возникновения неисправностей для оценки надежных характеристик логических схем // *Информационные технологии*. 2014. № 11. С. 30-36.
- [25] Иванников А.Д., Стемпковский А.Л. Анализ итерационных методов решения систем логических уравнений и их использование при моделировании цифровых систем // *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*. 2020. № 3. С. 2-8.
- [26] Гаврилов С.В., Иванова Г.А., Стемпковский А.Л. Теоретико-графовая модель сложно-функциональных блоков для КМОП технологий с трехмерной структурой транзистора // *Известия ЮФУ. Технические науки*. 2014. № 7 (156). С. 58-68.
- [27] Стемпковский А.Л., Тельпухов Д.В., Соловьев Р.А., Тельпухова Н.В. Исследование вероятностных методов оценки логической уязвимости комбинационных схем // *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*. 2016. № 4. С. 121-126.
- [28] Никитин С.А., Николаев А.В., Путря Ф.М., Неклюдов И.А. Автоматизация маршрута функциональной верификации на основе стандарта IP-XACT // *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*. 2020. № 4. С. 90-94.
- [29] Ivannikov A., Stempkovsky A., Romanov A. Set-theoretic model of digital systems functioning. 2016 International Siberian Conference on Control and Communications. SIBCON 2016 – Proceedings. Moscow. 2016. P. 7491726.
- [30] Гаврилов С.В., Глебов А.Л., Стемпковский А.Л. Анализ помехоустойчивости цифровых схем на основе логических импликаций // *Известия высших учебных заведений. Электроника*. 2002. № 5. С. 60.

Organization of Debugging Process for Digital Microelectronic System Designs

A.D. Ivannikov

Institute for design problems in microelectronics of RAS, Moscow

Abstract — The paper discusses methods of debugging, that is, identifying and correcting errors in digital system designs, specified by the circuit diagram of hardware and the text of software or firmware. Debugging on a manufactured prototype of hardware and on a software model of a digital system, as well as on a combination of a real hardware blocks and a software model of the rest of the system is considered. Methods of organization of debugging modes for all considered methods are analyzed. It is indicated that for digital systems on a chip the most effective method is to debug a design on a software model.

Keywords — digital systems design, design debugging, debugging modes implementation.

REFERENCES

- [1] Usupov R.M. Informatization and Science // *Problemi informatizacii*. 1994. № 1-2. P. 22-28.
- [2] Tikhonov A.N., Ivannikov A.D. Informatization of Russian education and society as a whole // *International cooperation*. 1997. No. 4. P. 1.
- [3] Gorogov Y.P., Jevnov I.I., Ivannikov A.D., Tatarnikov Yu. A. Main streams of higher education informatization program // *Pedagogicheskaya informatika*. 1993. № 2. P. 38.
- [4] Stempkovsky A., Shilin V. CMOS Photodiode LSI as an Advanced Base for a Single Chip Receiver and Processor System // *Electronics: Science, Technology, Business*. 2003. No. 2(44), Pp. 14-21.
- [5] Tikhonov A.N., Ivannikov A.D., Cvetkov V.Ja. Obrazovatel'nye uslugi kak instrument kachestva obrazovaniya // *Mezhdunarodnyj zhurnal prikladnyh i fundamental'nyh issledovanij*. 2009. No 3. Pp. 94-96.
- [6] Abramov A.G., Bulakina M.B., Ivannikov A.D., Krivosheev A.O., Shmelkova L.V. The system of additional professional education of the Russian Federation: key statistical indicators based on the results of automated data collection // *Bulletin of the Peoples' Friendship University of Russia. Series: Informatization of education*. 2014. No. 1. Pp. 133-144.
- [7] Ivannikov A.D. Subject Internet portals as the means of aggregating electronic content in a given subject area // *Information technologies*. 2014. No. 3. Pp. 43-48.
- [8] Stempkovsky A.L., Amerbaev V.M., Solov'yev R.A.. Principles of recursive modular arithmetic. // *Informacionnie tehnologii*. 2013. No.2. Pp. 22-27.
- [9] Klimov A.V., Levchenko N.N., Okunev A.S., Stempkovskij A.L. Supercomputers. Memory Hierarchy and Dataflow Computation Model // *Program Systems: Theory and Applications*. 2014. V. 5. No. 1(19). Pp. 15-36.

- [10] Tiunov I.V., Lipatov I.A., Zheleznikov D.A. Digital circuits resynthesis approach for FPGAs based on logic cell with built-in flip-flop. // *Problems of Perspective Micro- and Nanoelectronic Systems Development*. 2019. Issue 3. Pp. 33-36.
- [11] Ivannikov A.D., Stempkovsky A.L. Complex Digital Systems and Microsystems Design Debugging Mathematic Model on the Basis of Stationary Dynamic System Family Presentation // *Problems of Perspective Micro- and Nanoelectronic Systems Development*. 2014. Issue 2. Pp. 123-128.
- [12] Ivannikov A.D. Theoretical Basis for the Selection of Design Debugging Tests Set for Digital Systems Based on the Alphabet of Functions Performed // *Information Technologies*. 2019. Vol. 25. No. 11. Pp. 657-662.
- [13] Slinkin D.I. Analysis of Modern VLSI Project Testing and Verification Methods // *Programmnie Produkti i Systemi*. 2017. T. 30. № 3. Pp. 401-408.
- [14] Ivannikov A.D. Debugging Input Set Generation for Testing of Control Digital Systems Functions // *Mekhatronika, Avtomatizatsiya, Upravlenie*. 2017. Vol. 18. No.12. Pp. 795-801.
- [15] Ilias Alexopoulos. How to Debug Embedded Systems // <https://www.EDN.com>, Dec. 11, 2012.
- [16] Abramov E.M., Egorov A.V., Kozlov A.O., Poperechniy P.S., Putrya F.M., Frolova S.E. Prototype Platform Choosing for IP Blocks and SoC Subsystems // *Voprosi Radioelektroniki*. 2017. № 8. Pp. 76-83.
- [17] Zotov V. Design and Debugging Means for Digital Devices and In-Circuit Microprocessor Systems on the Bases of XILINX FPGA Kintex-7 Series // *Komponenti i Technologii*. 2012. № 4 (129). Pp. 124-132.
- [18] Zhezlov K.A., Kolbasov Y.S., Kozlov A.O., Nikolaev A.V., Putrya F.M., Frolova S.E. Automation of verification environments development process providing a through design flow for design, verification and research of IP-blocks and SoC // *Problems of Perspective Micro- and Nanoelectronic Systems Development* - 2016. Issue 2. P. 46-53.
- [19] X. Hu, Y. Jin and Z. Li. A Parallel JTAG-based Debugging and Selection Scheme for Multi-core Digital Signal Processors // *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*. Chongqing, China. 2018. Pp. 527-530.
- [20] Ivanov S.A., Oreshkin D.M. Hardware-Software Complex for Development and Debugging of Electronic Systems Based on Digital Microprocessor Modules // *Novaya Nauka: ot Idei kResultatu*. 2016. № 8-1 (96). Pp. 27-29.
- [21] Strelec A.I., Protopopova U.D., Ivannikov V.S. Universal System for Digital Device Testing // *Nauchniy Jurnal*. 2018. № 5 (28). Pp. 41-44.
- [22] Nguen M.D. Hardware/software formal co-verification using hardware verification techniques. Fourth Int. Conf. on Communications and Electronics (ICCE). 2012. Pp. 465-470.
- [23] Romanov A.Yu., Ivannikov A.D., Romanova I.I. Simulation and synthesis of network-on-chip by using NOCSIPM HDL library. 2016 IEEE 36th International Conference on Electronics and Nanotechnology, ELNANO 2016 – Conference Proceedings. 36. 2016. Pp. 300-303.
- [24] Stempkovskiy A.L., Telpukhov D.V., Solovyev R.A., Solovyev A.N., Myachikov M.V. Fault simulation technique for logic circuits reliability characteristics evaluation // *Informacionnie Technologii*. 2014. No. 11. Pp. 30-36.
- [25] Ivannikov A.D., Stempkovsky A.L. Analysis of Iterative Methods for Solving Logical Equation Systems and their Use in Digital System Simulation // *Problems of Perspective Micro- and Nanoelectronic Systems Development* - 2020. Issue 3. P. 2-8. doi:10.31114/2078-7707-2020-3-2-8.
- [26] Gavriliv S.V., Ivanova G.A., Stempkovsky A.L. Theoretical Graph Model of Complex Functional Blocks for CMOS Technology with Three Dimension Transistor Structure. *Izvestiya UFU. Tehnicheskie Nauki*. 2014. No. 7 (156). Pp.58-68.
- [27] Stempkovskiy A.L., Telpukhov D.V., Solovyov R.A., Telpukhova N.V. Probabilistic Methods for Reliability Evaluation of Combinational Circuits // *Problems of Perspective Micro- and Nanoelectronic Systems Development*. 2016. Issue 4. Pp. 121-126.
- [28] Nikitin S.A., Nikolaev A.V., Putrya F.M., Neklyudov I.A. Route automation of Functional Verification based on IP-XACT standard // *Problems of Perspective Micro- and Nanoelectronic Systems Development*. 2020. Issue 4. P. 90-94.
- [29] Ivannikov A., Stempkovsky A., Romanov A. Set-theoretic model of digital systems functioning. 2016 International Siberian Conference on Control and Communications. SIBCON 2016 – Proceedings. Moscow. 2016. P. 7491726.
- [30] Gavrilov S.V., Glebov A.L., Stempkovskiy A.L. Digital circuits noise immunity analysis on logical implication base // *Izvestiya Vysshikh Uchebnykh Zavedenii. Elektronika*. 2002. No. 5. P. 60.