

О формальной спецификации цифровых систем

А.К. Скуратов

Российский государственный аграрный университет – МСХА им. К.А.Тимирязева, г. Москва
akskuratov@rgau-msha.ru

Аннотация — Рассмотрены вопросы составления функциональной спецификации на проектируемую цифровую специализированную систему с целью последующего формирования набора тестовых примеров для проверки правильности и отладки проекта, задаваемого схемой технических средств и текстом программного обеспечения. Предполагается, что сам процесс отладки осуществляется на компьютерной функционально-логической модели проекта цифровой системы. В качестве модели технических средств цифровой системы используется семейство стационарных динамических систем с конечнзначным представлением логических сигналов и непрерывным временем. Формирование тестовых примеров для отладки проекта предлагается проводить на основе последовательностей функций, выполняемых цифровой системой при управлении оборудованием. Для формирования таких последовательностей предлагается использовать иерархические сети Петри.

Ключевые слова — отладка проектов цифровых систем, функционально-логическое моделирование, сети Петри, формирование отладочных тестовых примеров, спецификация набора функций.

I. ВВЕДЕНИЕ

Начиная с последнего десятилетия прошлого века и до настоящего времени перспективным и обязательным становится внедрение информационных технологий во все направления деятельности современного общества [1, 2]. Существенное развитие получают как теоретические аспекты информационных технологий: методы искусственного интеллекта, теория знаний, формализация различных аспектов человеческой деятельности, так и их практическое использование для развития телекоммуникаций, образования, промышленности и управления [3-8].

Внедрение цифровой техники в управление различными производственными процессами требует создания большого количества специализированных цифровых систем. Подобные цифровые системы, часто их называют контроллерами, проектируются на различных предприятиях разработчиками, обладающими различной квалификацией и опытом, в связи с чем рассматриваемая проблема является актуальной. Проект управляющей цифровой системы включает принципиальную схему соединения блоков и элементов и целевые программы или микропрограммы, совместно реализующие функции устройства управления, заданные в техническом задании. Набор функций соответствует автоматизируемому производственному процессу.

В области анализа правильности проектов цифровых управляющих систем существуют следующие направления: формальные методы доказательства корректности проекта; макетирование; моделирование на ЭВМ, предполагающее использование в качестве объекта исследования программную модель проектируемого устройства.

Формальные методы математических доказательств, являясь наиболее строгой проверкой правильности проекта, носят чаще теоретический характер. Метод макетирования наиболее доступен и широко распространен вследствие своей простоты и наличия разнообразных средств отладки и технической диагностики, однако, изготовление макета связано со значительными материальными и временными затратами. Метод моделирования на ЭВМ наиболее предпочтителен вследствие своей универсальности относительно архитектуры устройств, проекты которых анализируются, что особенно важно для цифровых микроэлектронных систем [9-11].

Анализ проектов цифровых систем рассматриваемого в данной работе класса проводится на функционально-логическом уровне. В качестве инструмента анализа используют системы функционально-логического моделирования [12-14]. При эксплуатации таких систем пользователи-разработчики специализированных цифровых систем формируют тестовые примеры на основе своего опыта и интуиции. Для эффективного использования систем функционально-логического моделирования необходима разработка формализованных методик и соответствующих им средств, позволяющих формировать тестовые примеры и подавать их автоматически на модель.

Цель функционально-логического моделирования может быть достигнута только при использовании в качестве исходной информации (для построения тестовых примеров) формального, независимого от конкретной проектной реализации, описания работы специализированной цифровой системы – функциональной спецификации. Формирование тестовых примеров на основе функциональной спецификации не только дает возможность анализа любой проектной реализации специализированной цифровой системы, которая объявляется способной выполнять функции, оговоренные в спецификации, но и позволяет сократить продолжительность самого анализа, поскольку тестовые примеры могут быть получены сразу после утверждения функциональной спецификации.

II. ТЕХНИЧЕСКОЕ ДИАГНОСТИРОВАНИЕ НА ЭТАПЕ ПРОЕКТИРОВАНИЯ

Рассмотрим анализ проекта цифровой системы с точки зрения технического диагностирования на этапе проектирования. Техническое диагностирование цифровых систем на этапе проектирования заключается в определении соответствия проекта цифровой системы ее функциональной спецификации [15-18]. Под объектом технического диагностирования на этапе проектирования понимается проект цифровой системы, заданный совокупностью принципиальной электрической схемы и целевых программ или микропрограмм. Под нормативно-технической документацией на этапе проектирования понимается функциональная спецификация. Таким образом анализ правильности проекта цифровой системы является составляющей технического диагностирования и заключается в обнаружении и устранении ошибок проектирования. Проверяющий тест для анализа исправности (соответствия функциональной спецификации) проекта цифровой системы назовем тестовым примером.

Виды тестового контроля цифровых устройств и систем подразделяются на структурный, структурно-функциональный и функциональный. Методы структурного и структурно-функционального видов контроля, использующие в качестве исходной информации принципиальную или структурную схему цифрового устройства или системы, не приемлемы для целей анализа правильности их проекта, поскольку схема сама входит в объект диагностирования. Поэтому единственно возможным видом тестового контроля на этапе проектирования является функциональный, использующий функциональное описание цифровой системы. Ему соответствует способ формирования тестовых входных воздействий по заданному рабочему алгоритму специализированной цифровой системы [19-21].

При выборе метода, вида, способа и подхода тестового контроля для целей анализа правильности проекта специализированной цифровой системы учитывалась особенность данного анализа в двух аспектах: отличие специализированной цифровой системы от универсального процессора, заключающегося в многообразии архитектур, и в целевом назначении; отличие тестового контроля на этапе проектирования от тестового контроля на этапах изготовления и эксплуатации.

Стратегия анализа правильности проекта специализированной цифровой системы состоит в следующем: составление неформализованной функциональной спецификации; анализ полноты и непротиворечивости неформализованной функциональной спецификации коллективом экспертов; запись функциональной спецификации в формализованной виде; анализ корректности функциональной спецификации формальными методами, соответствующими формализмам, положенным в ее основу; составление тестовых примеров по функциональной спецификации параллельно с разработкой функционально-логической модели цифровой системы;

проведение эксперимента с моделью цифровой системы с использованием построенных тестовых примеров.

III. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ СПЕЦИАЛИЗИРОВАННЫХ ЦИФРОВЫХ СИСТЕМ

Для реализации предложенной стратегии необходимо разработать математическую модель функционирования специализированных цифровых систем как формализм их функциональной спецификации. На основе требований к функциональной спецификации, которые диктуются особенностью специализированных цифровых систем управления и предъявляются к функциональным спецификациям в целом, предлагается следующая модель. В качестве основы для исследования выбираем математическую модель поведения цифровой системы в виде семейства стационарных динамических систем с конечными множествами входных, выходных и внутренних состояний и непрерывным временем [22-24]. В результате исследования множества допустимых взаимодействий цифровой системы с внешней средой с учетом изменений значений внутренних переменных получена математическая модель функционирования специализированных цифровых систем для целей формализации их функциональных спецификаций: $S = (P, T, K, D, E, F)$, где P – конечное множество переменных цифровой системы, включающее подмножество переменных, соответствующих входным P^1 , выходным P^2 и двунаправленным P^3 линиям цифровой системы, и множество переменных, соответствующих регистрам и триггерам цифровой системы P^4 ; множество действительных неотрицательных чисел $T = \{t | t \geq 0\}$; K – множество взаимодействия цифровой системы с внешней средой с учетом результирующих событий по внутренним переменным; D – конечное множество временных ограничений на входные взаимодействия; E – конечное множество временных ограничений на выходные взаимодействия; F – конечное множество функций цифровой системы.

$K = (M, R, \alpha)$, где M – множество входных взаимодействий, элемент $\mu_i \in M$ определяется как совокупность событий входного воздействия, выходного воздействия и начальных значений переменных подмножества P^4 ; R – множество реакций цифровой системы, элемент $r_i \in R$ определяется как совокупность событий по переменным подмножества P^2 и событий, характеризующих внутреннее состояние цифровой системы в момент окончания рассматриваемого взаимодействия с внешней средой; $\alpha: M \times R \rightarrow \{0, 1\}$ – отношение причины, предикат $P_\alpha(\mu_i, r_i) = 1$ в том и только в том случае, если входное взаимодействие μ_i приводит к реакции r_i .

Элементы $\mu_i \in M$ и $r_i \in R$, задаваемые последовательностью событий, являются континуальными множествами. В частности, вследствие того, что перед осуществлением взаимодействия цифровая система может находиться в одном из нескольких мгновенных состояний, возможен различный допустимый порядок сверше-

ния событий; как исходное, так и результирующее значение переменных в отдельных событиях могут представлять собой множества; момент совершения события, как правило, лежит в интервале действительных чисел.

Введем множество $\mathbf{F} = (f_i | k_1^i l_1^i k_2^i l_2^i \dots k_p^i l_p^i)$, где $l_j^i \in \mathbf{L}$ – множество логических функций. Определим отображение следования $\xi: (\mathbf{K} \cup \mathbf{F}) \rightarrow (\mathbf{K} \cup \mathbf{F})$. Выделим подмножество \mathbf{K}_n начальных взаимодействий. Функцию, выполняемую цифровой системой, определим как последовательность элементов множества \mathbf{F} , получающуюся при выполнении отображения ξ первоначально над элементом $k_n^i \in \mathbf{F}$ и далее над $\xi(k_n^i), \xi(\xi(k_n^i))$ и так далее, до тех пор, пока в качестве результата не будет получен любой из уже включенных в последовательность элементов множества \mathbf{F} . Следовательно, $\Phi = (\mathbf{F}, \mathbf{K}_n, \xi)$. Множество функций, имеющих одинаковое $k_n \in \mathbf{K}_n$, будем рассматривать как режим работы цифровой системы.

IV. ИСПОЛЬЗОВАНИЕ ИЕРАРХИЧЕСКИХ СЕТЕЙ ПЕТРИ ПРИ ОПИСАНИИ ФУНКЦИОНИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ

Для построения функциональной спецификации на специализированную цифровую систему выберем модельное, комбинированное средство, а именно, иерархическую сеть Петри, элементы которой нагружены процедурными лексическими конструкциями. Сеть Петри представляется в виде системы логических уравнений. Множество логических функций, явно и неявно используемых при этом, определяет ее логический базис.

Для целей дальнейших исследований введем ряд новых понятий в теории сетей Петри. Сформулируем принцип преобразований сетей Петри: для того, чтобы сеть Петри А можно было преобразовать в сеть Петри В, необходимо, чтобы логический базис сети А мог быть представлен средствами логического базиса сети В, и достаточно, чтобы логический базис сети В был бы функционально полным. Сеть Петри определим как функционально полную, если ей соответствует функционально полный базис логических функций. Сформулируем критерий эквивалентности сети Петри машине Тьюринга-Поста, а именно: любая функционально полная сеть Петри и только она позволяет моделировать или описывать машину Тьюринга-Поста, а следовательно, любой алгоритмический процесс. Введем понятие чисто потоковых, сходящихся, расходящихся сетей Петри. Расширим понятие ограниченности сетей Петри и введем свойство 0-безопасности: позиция сети Петри является 0-безопасной, если при любом выполнении данной сети фишка может оказаться в позиции только один раз.

Множество функций специализированной цифровой системы Φ , а также ранее определенные множества $\mathbf{K}, \mathbf{D}, \mathbf{E}$ представим в виде иерархической, функционально полной, маркированной сети Петри $\mathbf{C}_1 = (\mathbf{V}_1, \mathbf{K}, I_1, \mathbf{O}_1, \mathbf{L}, \lambda^+, \lambda^-, \mathbf{M}_1^0)$, где \mathbf{V}_1 – конечное множе-

ство позиций, \mathbf{K} – конечное множество переходов, соответствующее множеству взаимодействий цифровой системы с внешней средой; $I_1: \mathbf{V}_1 \rightarrow \mathbf{K}$; $\mathbf{O}_1: \mathbf{V}_1 \rightarrow \mathbf{K}$ – расширенные входная и выходная функции соответственно, определенные таким образом, что $\#(k_j, I_1(b_i)) = \#(b_i, \mathbf{O}_1(k_j))$; $\#(k_j, \mathbf{O}_1(b_i)) = \#(b_i, I_1(k_j))$; \mathbf{L} – функционально полный базис логических функций; λ^+ – входное отображение, ставящее каждому переходу $k_i \in \mathbf{K}$ в соответствие логическое выражение типа $b_1^{l_1^i} l_1^i b_2^{l_2^i} l_2^i b_3^{l_3^i} \dots l_{r-1}^i b_r^{l_r^i}$, возможно включающее скобки, где $\forall l_j^i \in \mathbf{L}, \forall b \in I_1(k_i)$; λ^- – выходное отображение, ставящее каждому переходу $k_i \in \mathbf{K}$ в соответствие логическое выражение указанного выше типа, однако в котором $\forall b_j^{l_j^i} \in \mathbf{O}_1(k_i)$; \mathbf{M}_1^0 – множество начальных маркировок сети \mathbf{C}_1 , причем для $\forall \mathbf{M}_i^{01} \in \mathbf{M}_1^0: \mathbf{M}_i^{01}: \mathbf{V}^i \rightarrow N, \mathbf{V}^i \in I(k_n)$.

Каждый $k_v \in \mathbf{K}$ сети \mathbf{C}_1 представим в виде совокупности сетей второго уровня иерархии: \mathbf{C}_2^{v1} – сходящаяся чисто потоковая сеть, описывающая вычисление логического выражения $\lambda^+(k_v)$; \mathbf{C}_2^{v2} – расходящаяся чисто потоковая сеть, описывающая порядок попадания фишек в позиции множества $\mathbf{O}_1(k_v)$; \mathbf{C}_2^{v3} – чисто потоковая, временная, маркированная сеть, описывающая элементарные взаимодействия в пределах данного взаимодействия, названного алгоритмическим. Сеть \mathbf{C}_3^{v1} – третьего уровня иерархии описывает ситуацию, когда перед выполнением взаимодействия k_v цифровая система может находиться в одном из нескольких мгновенных состояний. Отображение $\tau_2^{v3}: \mathbf{V}_2^{v3} \rightarrow INT_2$, определяющее временной интервал хранения фишек в позициях сети \mathbf{C}_2^{v3} , является основой для построения сети \mathbf{C}_3^{v2} , при выполнении которой фиксируется конкретное время наступления событий, происходящих в процессе функционирования цифровой системы. Сеть Петри четвертого уровня иерархии \mathbf{C}_4^v описывает различные допустимые комбинации изменений значений переменных цифровой системы при свершении рассматриваемого события.

Таким образом, предложен формальный принцип описания функционирования цифровой системы при помощи иерархической сети Петри. При этом функция цифровой системы является 0-безопасным фрагментом выполнения максимальной длины построенной иерархической сети Петри. Для получения множества функций цифровой системы необходимо различным образом выполнить сеть Петри при заданной начальной маркировке, которая определяет режим работы цифровой системы.

V. МЕТОДИКА СОСТАВЛЕНИЯ ФУНКЦИОНАЛЬНОЙ СПЕЦИФИКАЦИИ И ПОДХОД К ФОРМИРОВАНИЮ НАБОРА ТЕСТОВЫХ ПРИМЕРОВ

Методика составления функциональной спецификации на специализированную цифровую систему следующая:

- задать множество функций специализированной цифровой системы посредством определения множе-

ства алгоритмических взаимодействий с внешней средой, условий наступления и возможных продолжений каждого из них, что фактически формирует сеть Петри верхнего уровня иерархии; на данном этапе возможно применение иерархического подхода;

- описать переменные цифровой системы посредством использования специального абстрактного вопросника;

- задать каждое из алгоритмических взаимодействий; такое задание заключается в кодировании соответствующей временной диаграммы; в этих целях разработаны необходимые конструкции сети Петри в описывающие их лексические средства;

- задать множество режимов работы цифровой системы посредством формирования и использования абстрактного вопросника, что соответствует заданию начальной маркировки сети Петри на всех уровнях иерархии (заданию начальных значений).

При этом используется язык функциональных спецификаций, основной концепцией построения которого является объединение элементов и свойств языков спецификаций, применяемых в проектировании программного обеспечения вычислительных систем (абстрактный уровень), и языков уровня регистровых передач, применяемых в проектировании цифровых устройств (уровень кодирования временных диаграмм). Оба уровня языка спецификаций основаны на единых математических формализмах аппарата теории сетей Петри.

Такими образом, структура функциональной спецификации специализированной цифровой системы включает следующее.

1. Раздел функций.
2. Раздел описания переменных.
3. Раздел алгоритмических взаимодействий.
4. Раздел режимов работы.

Определим формальные критерии корректности функциональной спецификации цифровой системы: удовлетворение свойств 0-безопасности и К-ограниченности сети Петри, отсутствие петель некоторых типов, отсутствие конфликтов и пассивных переходов. Формальный анализ корректности функциональной спецификации проводится при составлении каждого из ее разделов, а также в процессе формирования тестовых примеров.

Сетевое представление функциональной спецификации обуславливает метод формирования тестовых примеров на ее основе. Отдельному тестовому примеру поставим в соответствие функцию цифровой системы в том виде, как она определена выше.

Процесс формирования тестовых примеров включает три алгоритма.

1. Алгоритм построения функций для заданного режима работы специализированной цифровой системы, задающий последовательность выполнения алгоритмических взаимодействий: строится функция как одно из

выполнений иерархической сети Петри; функция, построенная на предыдущем шаге алгоритма, просматривается снизу вверх с целью нахождения нового выполнения иерархической сети Петри; при достижении начальной маркировки и невозможности по новому выполнить сеть, алгоритм завершает работу.

Исходными данными для алгоритма является раздел функций спецификации, а результатом – последовательность выполняемых алгоритмических взаимодействий. Свойство 0-безопасности сети Петри, которое определяет функцию системы, обеспечивает конечность ее построения.

2. Алгоритм формирования параметризованного тестового примера на уровне выполнения алгоритмических взаимодействий. В качестве параметров тестовых примеров рассматриваются значения на информационных шинах и во внутренних регистрах данной цифровой системы, а также значения временных задержек между событиями. Составление параметризованного тестового примера для отдельной функции цифровой системы заключается в выполнении событий тех алгоритмических взаимодействий, которые задаются алгоритмом построения функций. Тем самым осуществляется выполнение сети Петри, описывающей временные диаграммы и заданной языковыми средствами уровня регистровых передач.

Предусмотрена база данных, которая является внутренним представлением функциональной спецификации и содержит исходные данные для формирования параметризованного тестового примера. В результате работы данного алгоритма формируются строки значений переменных цифровой системы. Каждая строка соответствует совершению события. Некоторые строки содержат вместо значения информационный переменный параметр. Необходимость введения параметров обусловлена тем, что выделение в множестве переменных подмножеств управляющих и информационных переменных условно разделяет цифровую систему на управляющую и операционную части. Известно, что для операционной части, непосредственно выполняющей простейшие вычислительные операции, такие как сложение, сдвиг и т.п., метод построения функционального теста не существует. Для выбора конкретного экземпляра тестового воздействия для операционной части требуется вмешательство пользователя.

3. Эвристический алгоритм выбора значений параметров тестового примера. Данный алгоритм основан на методологиях, предложенных Г.Майерсом для целей тестирования программного обеспечения, а именно: эквивалентное разбиение, применение функциональных диаграмм, анализ граничных значений и предположение об ошибке [51-54]. Указанные методологии адаптированы для целей задания параметров формируемых тестовых примеров посредством учета особенностей функционирования специализированных цифровых систем.

VI. ЗАКЛЮЧЕНИЕ

Приведенные алгоритмы обеспечивают полную проверку управляющей части специализированной цифровой системы, а полноту проверки операционной части определяет пользователь путем задания параметров тестовых примеров в диалоговом режиме.

ЛИТЕРАТУРА

- [1] Тихонов А.Н., Иванников А.Д. Информатизация российского образования и общества в целом // *Международное сотрудничество*. 1997. № 4. С. 1.
- [2] Горохов Ю.П., Жевнов И.И., Иванников А.Д., Татарников Ю.А. Основные направления программы информатизации высшего образования // *Педагогическая информатика*. 1993. № 2. С. 38.
- [3] Иванников А.Д. Тематические интернет-порталы как средство агрегации электронного контента в заданной предметной области // *Информационные технологии*. 2014. №3. С. 43-48.
- [4] Стемпковский А.Л., Амербаев В.М., Соловьев Р.А. Принципы рекурсивных модулярных вычислений // *Информационные технологии*. 2013. № 2. С. 22-27.
- [5] 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.
- [6] 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.
- [7] Климов А.В., Левченко Н.Н., Окунев А.С., Стемпковский А.Л. Методы адаптации параллельной потоковой вычислительной системы под задачи отдельных классов // *Информационные технологии и вычислительные системы*. 2009. №3. С. 12-21.
- [8] Скуратов А.К., Сухарева Н.А. Информационные технологии дистанционного обучения // *Университетское управление: практика и анализ*. 2000. № 1. С. 37-42.
- [9] Слинкин Д.И. Анализ современных методов тестирования и верификации проектов сверхбольших интегральных схем // *Программные продукты и системы*. 2017. Т. 30. № 3. С. 401-408.
- [10] Ivannikov A., Kulagin V., Romanov A., Pozdneev B. Algebraic models of digital system design debugging decomposition. *Proceedings of 2016 IEEE East-West Design and Test Symposium, EWDTS 2016*. 2016. P. 7807712.
- [11] Стемпковский А.Л., Гаврилов С.В., Глебов А.Л. Методы повышения эффективности временного анализа СБИС // *Информационные технологии*. 2006. № 12. С. 2-12.
- [12] 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 нанoeлектронных систем (МЭС)*. 2020. № 4. С. 90-94.
- [13] 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 нанoeлектронных систем (МЭС)*. 2020. № 3. С. 2-8.
- [14] 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.
- [15] Камашев М.А. Специализированные модели для разработки программных приложений на основе алгебраических спецификаций и средств функционального программирования // *Системы управления и информационные технологии*. 2010. № 4(42). С. 73-78.
- [16] Поляков Г.А., Лысых В.В., Толстолужская В.В. Функциональный синтез параллельных неперестраиваемых спецпроцессоров с использованием аппарата структур семантико-числовой спецификации // *Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика*. 2012. № 13 (132). С. 142-149.
- [17] Иванников А.Д. Формирование отладочного набора тестов для проверки функций цифровых систем управления объектами // *Мехатроника, автоматизация, управление*. 2017. Т. 18. №. 12. С. 795-801.
- [18] Гаврилов С.В., Иванова Г.А., Рыжова Д.И., Соловьев А.Н., Стемпковский А.Л. Методы синтеза помехозащищенных комбинационных блоков // *Информационные технологии*. 2015. Т. 21. № 11. С. 821-826.
- [19] Nguen M.D. Hardware/software formal co-verification using hardware verification techniques // *Fourth Int. Conf. on Communications and Electronics (ICCE)*. 2012. Pp. 465-470.
- [20] 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.
- [21] Sokhatski A.A.. Use of Formal Methods to Resolve Actual Problems of ASIC Design Verification // *Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС)*. 2020. № 3. С. 22-27.
- [22] Иванников А.Д., Стемпковский 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 нанoeлектронных систем (МЭС)*. 2014. № 2. С. 123-128.
- [23] Скуратов А.К. Математическая модель функционирования специализированного микропроцессорного устройства как основа для составления его функциональной спецификации // *Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС)*. 2020. № 3. С. 65-72.
- [24] Gavrilov S.V., Ivannikov A.D., Stempkovsky A.L. Method of mathematical description for digital system blocks logical models // *Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС)*. 2019. № 2. С. 8-11.

On the Formal Specification of Digital Systems

A.K. Skuratov

Russian State Agrarian University – Moscow Timiryazev Agricultural Academy, Moscow
akskuratov@rgau-msha.ru

Abstract — Functional specification development for digital specialized system designs is considered. The main idea is the subsequent formation of a set of test examples for checking the correctness and debugging of the design specified by the hardware circuit diagram and the text of the software. It is assumed that the debugging process itself is carried out on a computer functional-logical model of a digital system design. A family of stationary dynamic systems with finite-value representation of logical signals and continuous time is used as a model of digital system hardware. The test examples generation for the design debugging is proposed to be carried out on the basis of the sequences of functions performed by the digital system when controlling the equipment. To form such sequences, it is proposed to use hierarchical Petri nets.

Keywords — digital systems design debugging, functional-logic simulation, Petri nets, debugging test set generation, function set specification

REFERENCES

- [1] Tikhonov A.N., Ivannikov A.D. Informatization of Russian education and society as a whole // International cooperation. 1997. No. 4. P. 1.
- [2] Gorogov Y.P., Jevnov I.L., Ivannikov A.D., Tatarnikov Yu. A. Main streams of higher education informatization program // Pedagogicheskaya informatika. 1993. № 2. P. 38.
- [3] 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.
- [4] Stempkovskiy A.L., Amerbaev V.M., Solovyev R.A.. Principles of recursive modular arithmetic. //Information technologies. 2013. No.2. Pp. 22-27.
- [5] 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.
- [6] 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.
- [7] Klimov A.V., Levchenko N.N., Okunev A.S., Stempkovskiy A.L. Parallel dataflow computing system adaptation for specific task classes // Journal of Information Technologies and Computing Systems. 2009. No. 3. Pp. 12-21.
- [8] Skuratov A.K., Sukhareva N.A. Information technologies of distance education // University management: practice and analysis. 2000. No. 1. Pp. 37-42.
- [9] Slinkin D.I. Analysis of Modern VLSI Project Testing and Verification Methods // Programmnie Produkti i Systemi. 2017. T. 30. № 3. Pp. 401-408.
- [10] Ivannikov A., Kulagin V., Romanov A., Pozdneevev B. Algebraic models of digital system design debugging decomposition. Proceedings of 2016 IEEE East-West Design and Test Symposium, EWDTs 2016. 2016. P. 7807712.
- [11] Stempkovskiy A.L., Gavrilov S.V., Glebov A.L. Methods for increasing the efficiency of VLSI time analysis // Information technologies. 2006. No. 12. Pp. 2-12.
- [12] 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. Pp. 90-94.
- [13] Ivannikov A.D., Stempkovskiy 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. Pp. 2-8.
- [14] 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.
- [15] Kamashev M.A. Specialized models for developing software applications based on algebraic specifications and functional programming tools // Control systems and information technologies. 2010. No. 4(42). Pp. 73-78.
- [16] Polyakov G.A., Lisikh V.V., Tolstolujskaya V.V. Functional synthesis of parallel non-tunable special processors using the apparatus of semantic-numerical specification structures // Belgorod state university scientific bulletin. Economics, Information technologies. 2012. No. 13(132). Pp. 142-149.
- [17] 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.
- [18] Stempkovskiy A.L., Gavrilov S.V., Ivanova G.A., Ryzhova D.I., Soloviev A.N. Metody sinteza pomekhozashchishchennykh kombinatsionnykh blokov // Informatsionnye tekhnologii, 2015. V. 21. No.11. Pp. 821-826.
- [19] Nguen M.D. Hardware/software formal co-verification using hardware verification techniques // Fourth Int. Conf. on Communications and Electronics (ICCE). 2012. Pp. 465-470.
- [20] Ivannikov A., Stempkovskiy 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.
- [21] Sokhatskiy A.A.. Use of Formal Methods to Resolve Actual Problems of ASIC Design Verification // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 3. Pp. 22-27.
- [22] Ivannikov A.D., Stempkovskiy 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.
- [23] Skuratov A.K. Mathematical model of the functioning of a specialized microprocessor device as a basis for drawing up its functional specification // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 3. Pp. 65-72.
- [24] Gavrilov S.V., Ivannikov A.D., Stempkovskiy A.L. Method of mathematical description for digital system blocks logical models // Problems of Perspective Micro- and Nanoelectronic Systems Development. 2019. Issue 2. Pp. 8-11.