

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*M. Semotjuk, E. Chichirin,
E. Sosnenko*

ORGANIZATION OF LAYERED MODELING FOR VERIFICATION OF DIGITAL PROJECTS

Expansion of functionalities of means of modeling and monitoring for development, debugging and verifications of systems of digital processing is investigated. The implementation of modeling of algorithm Viterby is resulted.

Key word: model, simulator, verification.

Досліджується розширення функціональних можливостей засобів моделювання для розробки, наладки та верифікації цифрових проектів. Розглядається реалізація моделювання алгоритму Вітербі. Ключові слова: модель, симулятор, верифікація.

Исследуются методы расширения функциональных возможностей средств моделирования и мониторинга для разработки, отладки и верификации цифровых проектов. Рассматривается реализация моделирования алгоритма Витерби.

Ключевые слова: модель, симулятор, верификация.

© В.М. Семотюк, Е.Н. Чичирин,
Е.П. Сосненко, 2012

УДК 681.3(031)

М.В. СЕМОТЮК, Е.Н. ЧИЧИРИН,
Е.П. СОСНЕНКО

ОРГАНИЗАЦИЯ МНОГОУРОВНЕВОГО МОДЕЛИРОВАНИЯ ДЛЯ ВЕРИФИКАЦИИ ЦИФРОВЫХ ПРОЕКТОВ

Введение. На этапах проектирования, отладки и верификации новых образцов микроконтроллеров и сигнальных процессоров (МП) эффективность процессов моделирования имеет первостепенное значение. Встроенные в системы автоматизированного проектирования (САПР) средства физического и функционального моделирования имеют свое продолжение в виде программных симуляторов, входящих в состав инструментальных наборов на основе серийных МП [1].

При разработке перспективных МП необходимо учитывать эффективность их использования для выполнения современных управляющих и телекоммуникационных алгоритмов уже на стадии разработки и отладки программной модели новых МП. Моделирование в этом случае обеспечивает сокращение циклов разработки, отладки, а также верификации и оптимизации структуры МП в комплексе с сопутствующими программами и приложениями. Моделирующие подсистемы при этом должны быть дополнены средствами разработки и отладки библиотек и других первоочередных программных модулей и приложений, а также содержать развитые средства верификации МП и его модели. Особое значение программное моделирование приобретает на промежуточных стадиях проектирования МП, например, на этапах конфигурирования и тестирования его блоков на базе кристаллов FPGA.

Постановка задачи. Круг требующих своего решения задач сформировался в процес-

се создания новых образцов микроконтроллеров, а также управляющих систем и приложений на их основе.

Во-первых, необходимость разработки программного обеспечения проектируемого МП (в том числе его программного симулятора) для подготовки и отладки на кросс-машине базовых программ и МП-приложений [2] формально дает возможность использовать его программную модель для последующей верификации сконфигурированного кристалла МП (и наоборот). Необходимым условием для этого кроме наличия тестирующих программ является разработка среды для их выполнения.

Во-вторых, с целью ускорения получения готового продукта необходимо разделять процесс схемотехнического проектирования МП на ряд этапов и совмещать проектирование одних блоков МП с тестированием других. Для этого необходимо решить задачу корректной работы программного симулятора в основном и вспомогательных уровнях взаимодействия с частично работоспособным МП.

В-третьих, учитывая разработку и тестирование большой и разнообразной номенклатуры разрабатываемых библиотечных программ по стандартной и специализированной цифровой обработке, возникает необходимость визуальной оценки результатов их работы. Интуитивно понятное (программистам-непрофессионалам в рассматриваемой проблемной области) визуально-графическое отображение процесса выполнения управляющих и обрабатываемых алгоритмов позволяет быстро и достаточно точно сравнить полученные результаты с ожидаемыми и локализовать причину возможных расхождений. Такой подход не требует предварительной подготовки на кросс-машине эталонных тестовых наборов и позволяет получить приемлемое качество тестирования. При этом необходимо обеспечить необходимые для адекватного представления верифицируемых процессов функциональность и производительность мониторинга МП и его модели.

Исследуемый круг вопросов близок к таким развивающимся областям верификации, как динамические и синтетические методы тестирования и мониторинга [3].

Предмет исследования. В работе исследуются методы использования имитационных программных моделей и графических моделей мониторинга динамических процессов для расширения возможностей верификации программных и аппаратных компонентов перспективных цифровых управляющих и обрабатываемых систем.

Модель. Модель проектируемого МП, альтернативная его VHDL описанию, строится на основании анализа технических требований проекта. В соответствии с рекомендациями [4] эта работа выполняется независимо от процесса построения модели МП в САПР. При последующем тестировании это дает возможность обнаружить неоднозначность понимания одного из наиболее трудно формализуемых этапов верификации всего проекта. Использование на данном этапе синтетических технологий тестирования на основе формальной модели требований и проектных решений (model based testing) позволяет частично ав-

томатизировать процесс построения оракула и оценить полноту тестового покрытия МП.

Следует отметить, что разработка тестов цифровых систем с высокой обнаруживающей и диагностирующей способностью является непростой, требующей компромиссных решений задачей. Сложность тестов подобного рода определяется в общем случае прямым произведением множества операций, специфицированных для каждого блока, множества значений исходных данных, множеств режимов исполнения операций и адресации исходных данных. С учетом существующей зависимости между последовательно выполняемыми операциями МП решение задачи его верификации путем прямого перебора всех возможных тестовых наборов невозможно. Разумная минимизация сложности теста предполагает учет лишь наиболее значимых (заложенных в проект) и наиболее вероятных (вследствие ошибок проектирования, изготовления или функционирования) зависимостей между внутривходовыми и межблочными операциями.

В конечном итоге структура теста принимает сложный, нелинейный характер и требует для своей реализации всех возможностей входного языка МП, а не только инструкций проверяемого на данном этапе блока. При не полностью отлаженном образце МП выполнение такого теста возможно на его работоспособной (как показывает практика, почти работоспособной) программной модели. При этом программная моделирующая среда должна поддерживать многоуровневое взаимодействие с МП в режиме master – slave (ведущий – ведомый), характеризующийся следующими признаками:

- тестирующее МП приложение (или его функционально законченная часть, или иное приложение МП) разрабатывается не на языке моделирующего host-компьютера, а на входном языке (ассемблере) МП;
- комплексно тестирующее приложение обрабатывается программным симулятором МП, а в МП загружаются данные и считываются реакции необходимые только на уровне верификации очередного блока;
- включающая симулятор интегрированная моделирующая среда (ИМС) содержит средства выбора в зависимости от степени готовности МП уровня взаимодействия симулятор – МП;
- полнота тестирования (тестового покрытия возможных неисправностей) определяется формируемой в симуляторе последовательностью команд, значениями их операционных и адресно-индексных полей, а также соответствующих наборов данных.

Для реализации данного подхода МП должен поддерживать возможность автономного тестирования составляющих его блоков, в частности, функционирование интерфейса с моделирующим host-компьютером на любом этапе верификации. Важным является также минимизация затрат на поддержку такого интерфейса со стороны МП. Это может быть вариант канала прямого доступа, технологический (J-Tag), либо другой, в том числе удаляемый на последнем этапе проектирования интерфейс. В целях минимизации затрат на управление уровнями в ИМС и МП целесообразно выделять в качестве кандидатов для такого тес-

тирования блоки с высокой степенью автономности, например, специализированные арифметические устройства.

Использование любого МП приложения в качестве его теста, даже в случае неработоспособности части блоков МП, расширяет возможности применения такого «внешнего» тестирования для верификации МП (и его модели) на библиотечных и базовых приложениях, создаваемых одновременно с самим МП. Улучшить и ускорить субъективную и объективную оценки внешней верификации, активно развиваемой в последнее время наряду с методами экстремального программирования [3], можно либо за счет исполнения приложения в родном контролирующем его окружении, либо за счет эмуляции основных контролируемых характеристик окружения в соответствующей подсистеме мониторинга.

Наиболее высокоуровневые функции человеко-машинного взаимодействия и мониторинга традиционно возлагаются на графический интерфейс. Графический НМИ-интерфейс ИМС выполняет функции:

- подготовки и отладки в редакторе кода базовых программ и приложений МП, а также взаимодействия оператора с ИМС на этапе выполнения подготовительных операций;

- интерактивного взаимодействия оператора с функционирующим МП и его моделью, в том числе, синхронизация и динамический мониторинг протекающих в них процессов в виде, наиболее адекватно воспринимаемом человеком.

Под адекватностью восприятия понимается разумное сочетание качества (удобства восприятия), естественной для человека информации, например, изображения или речи, и точности, с которой можно без дополнительных средств количественно оценить это качество. При этом, если приложение МП выполняет функции управления производственным или технологическим процессом, или другую абстрактную цифровую обработку, то адекватным будет ее структурирование по типам и форматам представления атрибутов в данной проблемной области.

В последнем, наиболее распространенном случае при большом количестве атрибутов используется их представление в виде таблиц, графиков, графов, диаграмм и других производных от них графических моделей (например, иерархических 3D моделей). В контексте рассматриваемой тематики – моделирования с целью верификации – графические модели, в силу своих когнитивных свойств, зачастую позволяют оценить результат проектирования при частичном отсутствии формализованных и вербализованных требований к проекту, т. е. повысить эффективность как верификации, так и валидации проектируемой системы [4].

Для получения высоких эргономических характеристик средств графической визуализации (программных виртуальных мониторов) необходимо обеспечить согласованное во времени и пространстве мультимедийное отображение в них текста, таблиц, графиков, а также аудио и видео потоков. Интерактивный характер графических моделей предполагает как вывод, так и ввод соответствующих данных в каждый из виртуальных мониторов. Возможны три основных варианта реализации интерфейса МП (или его модель) – ИМС:

- разработка в рамках ИМС алгоритмов и программ обновления информации на экранах ее мониторов с автоматическим отслеживанием ее изменения в памяти и регистрах МП и его программной модели (оптимальный вариант, не требует написания кода для МП);

- выделение общей памяти для обмена данными между ИМС и приложениями, выполняемыми на МП или его модели (требует написания программного кода для каждого приложения МП);

- разработка для МП полноценных средств ввода-вывода видеографических и аудио данных. Лучшим, но и самым сложным вариантом, по-видимому, является реализация, использующая API OpenMAX, представляющего кросс-платформенный абстрактный доступ к аудиовизуальным средствам [5].

Структурная схема выполняемых в едином процессе алгоритмов моделирования и мониторинга состояний МП и его модели показана на рис.1. Выполнение алгоритма начинается с загрузки из исполняемого файла кода программы и данных в модель МП. При задании основного MP уровня моделирования МП или подуровня sMP проверки его АЛУ код и данные загружаются в сам МП. При включенном симуляторе SIM и готовности МП к обмену на t -м шаге исполнения программы код очередной команды C_i загружается из модели МП в МП, и происходит запуск выполнения этой команды (или всей программы) в МП и его модели. При моделировании команды C_i симулятор подготавливает список $AWL_i = \{A_{ik}\}$, $k = \overline{1, L_i}$ адресов ячеек памяти, в которые производилась запись, и запоминает количество таких ячеек L_i . По окончании моделирования команды в симуляторе и МП в режиме JTG мониторинга МП или SMP – компарирования состояний МП и его модели происходит последовательный поиск адресов из списка записи AWL_i в упорядоченных AM_j списках отображения каждого из активных V_j мониторов $j = \overline{1, n}$, заданных своими минимальными $A_j \min$ и максимальными $A_j \max$ значениями. В случае совпадения адресов производится обновление ячейки памяти по адресу A_j j -го монитора соответственно данными D^* из МП или D из его модели. В режиме SMP дополнительно производится инвертирование цвета отображения данных в j -м мониторе по адресу A_j : $Color\langle A_j \rangle = inv\ Color\langle A_j \rangle$. Описанный цикл повторяется до момента выполнения команды выхода из исполняемой программы.

На схеме не представлена поддерживаемая функция обновления всех данных монитора при его включении, а также автоматическом или ручном изменении его базовых или индексных адресов сканирования, или данных, принадлежащих его области сканирования.

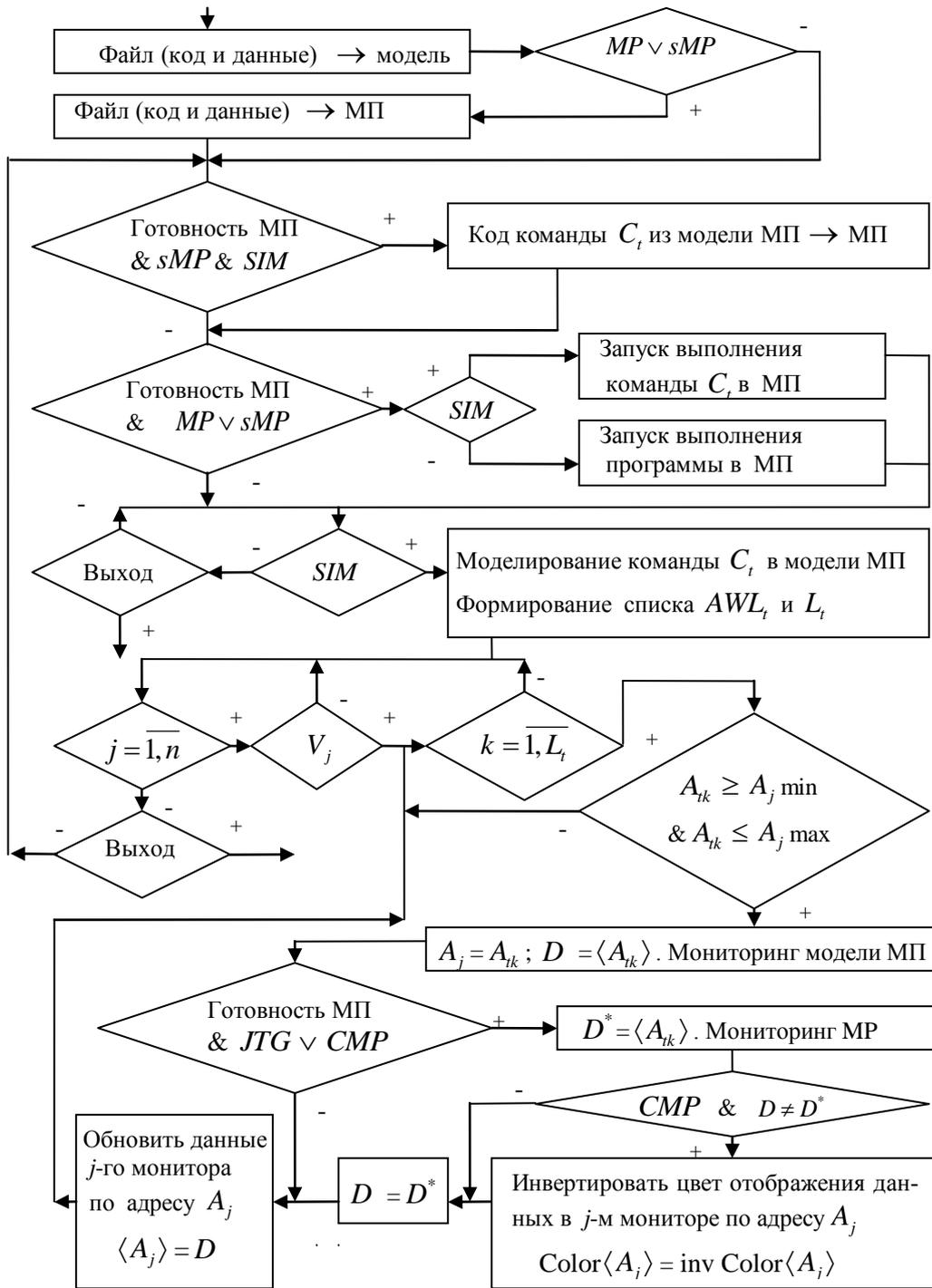


РИС. 1. Структурная схема процессов моделирования и мониторинга МП

Мультиформатные графические и табличные мониторы ИМС, обеспечивают динамическое отображение состояний непрерывных областей памяти, а также развертку на графиках и в таблицах последовательности этих состояний во времени (параметры SHistory и LHistory). Поддерживается индивидуальное изменение визуальных параметров таблиц и графиков с возможностью ручной, автоматической, следящей настройки базовых и индексных адресов сканируемых областей памяти, а также ввод графиков с помощью «мыши».

В зависимости от состояния вышеперечисленных элементов управления поддерживается более семи востребованных режимов совместного функционирования МП, его программной модели и подсистемы динамического мониторинга. Комбинации режимов функционирования мониторов различных типов существенно увеличивают это число.

Реализация. Программный симулятор, обеспечивающий базовый (независимый от работы МП) и «ведущий» уровни взаимодействия с МП, а также мультиформатная подсистема динамического мониторинга, реализованы в составе интегрированной моделирующей среды, совместно с редактором программного кода, компилятором с входного языка программирования и рядом вспомогательных утилит. Поддерживает ввод, редактирование и отладку программного кода разрабатываемого МП на языке его ассемблера. ИМС обеспечивает разнообразные режимы графической визуализации процессов преобразования данных при выполнении приложений в реальном устройстве и (или) на его программной модели. Кроме графического, поддерживается также аудио ввод-вывод с использованием средств ОС персонального компьютера.

В ИМС реализована современная технология НМІ-интерфейса: автосохранение в отдельном файле проекта всех текущих установок и списка файлов проекта, а также настроек экрана, редактора, мониторов и т. п. [6]. Имеется также возможность автозапуска ИМС при открытии какого-либо ассоциированного с ней файла.

В качестве примера рассмотрим алгоритм Витерби, широко применяющийся для декодирования сверточных кодов в современных телекоммуникационных системах связи [7].

Диаграмма состояний конечного автомата – сверточного кодера со скоростью $R = m/n = 1/2$, где $m = 1$ и $n = 2$ – длина в битах соответственно входного и выходного слова кодера, показана на рис. 2.

При декодировании принятой последовательности слов по алгоритму Витерби, из двух сходящихся в каждое из четырех состояний путей оставляется путь с минимальной метрикой (расстоянием по Хеммингу до принятой последовательности). Графический монитор ИМС позволяет представить процесс декодирования в виде динамически формируемой решетчатой структуры, образно оценить работу декодера и сопоставить ее с динамически подсвечиваемым программным кодом в текстовом редакторе. Поддерживается возможность пошагового контроля выполняемой команды в редакторе программ и мониторинга результатов ее выполнения в табличном и графическом виде.

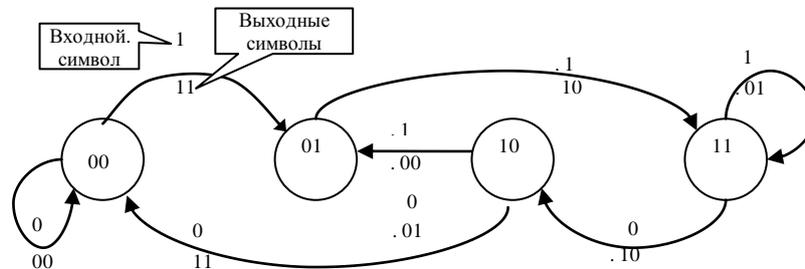


РИС. 2. Диаграмма состояний для сверточного кода с $R = 1/2$

Выводы. Исследованы вопросы построения подсистем моделирования и мониторинга для верификации и отладки перспективных средств цифровой обработки сигналов и специализированных управляющих систем. Предложены методы повышения эффективности моделирования аппаратных средств на базе микросхем FPGA для организации поблочного автономного их тестирования. Разработан мультиформатный графический интерфейс и алгоритмы динамического мониторинга проектируемых систем. Подсистемы моделирования и мониторинга интегрированы в функционально законченную среду для редактирования, компиляции и отладки программного кода приложений проектируемых цифровых систем.

1. Корнев В.В., Кисилев А.В. Современные микропроцессоры. 3-е изд. – СПб.: БХВ-Петербург, 2003. – 448 с.
2. Наливкин А.В. Микропроцессоры и ЭВМ в измерительной технике. Электронный учебник. СПбГУ ИТМО. – <http://de.ifmo.ru>. 20.09.2012.
3. Кулямин В.В. Методы верификации программного обеспечения. – М.: Ин-т системного программирования РАН, 2008. – 117 с.
4. Синицин С.В., Налютин Н.Ю. Верификация программного обеспечения. Курс лекций. – М.: МИФИ, 2006. – 157 с.
5. OpenMAX – The Standart for Media Library Portability. - <http://www.khronos.org>. 21.09.2012.
6. Палагин А.В., Семотюк М.В., Чичирин Е.Н., Сосненко Е.П. Acoustic commander – интегрированная операционная среда для измерения и расчета акустических параметров // Комп'ютерні засоби, мережі та системи. – 2009. – № 4. – С. 3 – 10.
7. Кларк Дж., мл., Кейн Дж. . Кодирование с исправлением ошибок в системах цифровой связи. – Статистическая теория связи. – Вып. 58. – М.: Радио и связь, 1987. – С. 212 – 224.

Получено 15.10.2012