

# Роль и проблемы высокоуровневого этапа проектирования встраиваемых вычислительных систем

Алексей ПЛАТУНОВ,  
к. т. н.  
platonov@lmt.ifmo.ru

**Условия и характер проектирования встраиваемых систем требуют сегодня серьезного пересмотра методик и технологий их создания. Одним из путей решения проблемы может стать использование системы архитектурных абстракций, в которой объединяются традиционные аппаратная и программная составляющие вычислительной системы.**

## Введение

Стремительный рост потребности во встроженных вычислительных системах различного назначения на современном этапе заставляет разработчиков активно совершенствовать методы и средства проектирования. Определить встраиваемые системы и сети (ВсС) можно как специализированные (заказные) микропроцессорные системы, непосредственно взаимодействующие с объектом контроля или управления и объединенные с ним конструктивно. Диапазон ВсС уже сегодня чрезвычайно широк и простирается от простейших контроллеров бытовой техники до сложнейших и ответственных распределенных систем транспортной, промышленной, телекоммуникационной автоматизации.

Активно растет доля ВсС со сложной внутренней организацией, где сложность проявляется в таких особенностях, как многопроцессорная гетерогенная архитектура, распределенный характер вычислений, широкий диапазон потенциально доступных разработчику вычислительных ресурсов, который определяется разнообразием и быстрой сменяемостью элементной базы.

Ключевой особенностью создания ВсС следует считать комплексный подход к проектированию, охватывающий практически все уровни организации вычислительной системы (ВС). Разработчик реально сталкивается с необходимостью анализа большого числа разнообразных потенциально возможных вариантов архитектурной организации и целевой системы, а также инфраструктуры проекта в целом (речь идет, в первую очередь, о технологиях проектирования и инструментарии). Именно эта комплексность определяет коренные отличия в подходах, методах и средствах создания ВсС по сравнению с циклами проектирования иных классов ВС, где

удается в значительной степени сократить число анализируемых вариантов за счет использования готовых вычислительных платформ и крупномасштабных компонентов различного уровня. В информационной индустрии сегодня хорошо проработаны и освоены технологии схемно-конструкторского этапа проектирования аппаратуры и этапа программного кодирования, существуют развитые технологии создания программного продукта в целом. Активно ведутся работы по развитию технологий программирования систем реального времени и мобильных систем. Однако по-прежнему остаются открытыми вопросы создания эффективных средств комплексного (сквозного) проектирования специализированных микропроцессорных систем, каковыми являются ВсС.

В преодолении указанных проблем первоочередное значение приобретает развитие методов и средств высокоуровневого (архитектурного) проектирования ВсС, где центральное место занимает формирование цельного взгляда на организацию ВС и вычислительного процесса как, собственно, цели проектирования.

Поясним последнее утверждение. Во многом шаблонный характер сегодняшнего проектирования ВС, при всех его достоинствах, использует решения (frameworks, паттерны), которые получены в большинстве своем эмпирически. Цепочки организации вычислительного процесса включают такие элементы, как компиляторы, интерпретаторы, виртуальные машины, аппаратные программируемые процессоры, специальные функциональные аппаратные блоки, и многое другое. Выбирая ту или иную технологию, разработчик, часто неосознанно, распределяет элементы вычислительного процесса внутри инструментальной (design-time) и исполнительской (run-time) фаз проекта. Успешное создание

перспективных ВсС с учетом их разнообразия и специфики возможно на основе широкого анализа и осознанного выбора решений в обеих фазах существования проектируемой системы.

## Программируемость встраиваемых систем

Стремительное развитие цифровой элементной базы, стилей и технологий программирования, вычислительных архитектур и парадигм проектирования определили необходимость выделения большей части современных ВС в специальную группу, для которой изначально подчеркивается доминирующее значение программных технологий на всех уровнях организации системы. Такие ВС принято называть “Software Intensive Systems” [1]. Применительно к ВС общего назначения, в которых создание целевых приложений (то есть собственно решение прикладных задач) достаточно четко отделено от создания вычислительной платформы (то есть самой универсальной ВС), данный термин означает в первую очередь то, что акцент в решении задачи смещен в область программной разработки. В подавляющем большинстве случаев такое программирование выполняется в традиционном стиле, без вторжения в область аппаратно-зависимых частей и без использования специализированной программируемой аппаратуры или методов кремниевой компиляции.

Для области ВсС ситуация выглядит иначе. Здесь, наоборот, вычислительная платформа и прикладная надстройка не просто связаны друг с другом, во многих случаях между ними вообще сложно провести четкую границу [2]. Создание ВсС обязательно затрагивает все уровни организации системы. На практике в равной мере используются сценарии пол-

ностью заказного и полужаказного проектирования наряду с проектированием на готовой вычислительной платформе. Используются весь арсенал современных технологий и элементной базы.

Естественным следует считать появление для области ВСС уточненного варианта термина, отражающего значение и место программных технологий. Сегодня такие ВСС принято называть системами с преимущественно программной реализацией (SW-dominated), которые строятся на основе вычислительных платформ с большой долей программируемых средств (highly programmable platforms — «глубоко» программируемых платформ).

Смысл приведенной выше «двухзвенной» характеристики программируемости современных ВСС состоит в следующем.

Отнесение ВСС к системам с преимущественно программной реализацией объясняется следующими моментами. Во-первых, широким использованием на уровне аппаратуры программируемых процессоров — последовательных и параллельно-последовательных программных интерпретаторов (речь идет обо всем многообразии современных микропроцессоров и микроконтроллеров). Во-вторых, использованием языковых методов представления решаемой задачи (представление вычислительного процесса в виде программ того или иного стиля) с последующей реализацией средствами всего арсенала доступных технических решений (от последовательной программной интерпретации аппаратными процессорами до исполнения специально созданной аппаратуры). Естественным является использование иерархии аппаратно и программно реализованных интерпретаторов (виртуальных машин) в сочетании со специальными аппаратными средствами.

Перечислим основные проблемы, порождаемые этой частью «программируемости» ВСС:

- противоречие между ростом желаемого разработчиком уровня абстрактности в представлении все более сложных задач ВСС и необходимой «прозрачностью» (контролируемостью) реализации;
- гетерогенность вычислителей (процессоров — как в составе вычислительного узла, так и в сетевых конфигурациях);
- сложность функциональной и пространственной декомпозиции задачи;
- специфический «портрет специалиста», необходимого для работы в области программирования ВСС (охват многих уровней в организации ВСС; необходимость отслеживания таких «аспектов» проекта, как реальное время, надежность, энергосбережение и другие);
- неадекватность в части надежности большинства современных технологий программирования требованиям проектирования ВСС [3];

- отставание темпов роста эффективности проектирования от потребностей отрасли;
- низкий коэффициент повторного использования результатов проектирования.

Вторая часть характеристики проистекает из особенностей современной элементной базы. Наряду с традиционными программируемыми процессорными ядрами (микропроцессорами) разработчик имеет дело с большим числом программируемых спецпроцессоров, с конфигурируемыми функциональными расширителями и контроллерами, со схемами программируемой логики (ПЛИС, ПСнК). Современная цифровая и цифро-аналоговая элементная база конфигурируется в очень широком диапазоне: настройка временных параметров и режимов обмена элементов памяти, приемопередатчиков интерфейсов, контроллеров интерфейсов, АЦП, ЦАП, усилителей, источников питания, супервизоров и схем контроля. Сложность конфигурирования и программирования таких элементов может быть значительной.

В сумме это расширяет возможности разработчика, одновременно резко увеличивая риск ошибки и трудоемкость низкоуровневого проектирования. Попытки вообще уйти от низкоуровневого проектирования ВСС пока успехом не увенчались, так как навязывание разработчику ограниченного числа шаблонов конфигурирования резко ухудшает качество проектирования, а диапазон уровней организации системы, который вынужден охватывать разработчик для качественного управления аппаратурой, чрезвычайно широк.

Важным шагом в области проектирования ВСС явилось определение термина «встроенное программное обеспечение» (embedded software, ВПО) [3], который должен обозначить границы и особенности отрасли программирования ВСС.

Таким образом, «двойственная программируемость» современных ВСС порождает ряд методологических, методических и инструментальных проблем в проектировании, требующих сегодня решения. Результатом их решения должны стать качественно новые САПР, охватывающие системные уровни проектирования ВСС и предоставляющие разработчику возможность целенаправленного поиска вариантов во всем многообразии доступных вычислительных архитектур и маршрутов проектирования.

### Встроенное программное обеспечение

Обсудим некоторые варианты определения ВПО, вытекающие из них определения зон ответственности разработчика и охватываемые технологии.

Если ВПО трактовать как совокупность конечного кода, который определяет функцио-

нирование всех категорий программно-настраиваемых и программно-управляемых физических элементов в составе ВСС, то по стилям (методикам) создания целесообразно выделять область конфигурационного обеспечения (кода) и область программного обеспечения (кода).

Такой взгляд на ВПО обладает рядом достоинств, из которых основным можно считать простое и понятное (упрощенное) неискусственному специалисту в области ВСС представление о месте, роли, способах создания ВПО. Приемлемое качество проектирования ВПО в такой идеологии достигается только в рамках небольших и несложных проектов ВСС, где возможно использование шаблонных решений.

Недостаток такой упрощенной модели состоит в искусственном отделении конечного продукта (кода ВПО) от технологий и средств его создания. Это, по сути, исключает возможность эффективного поиска решений на системном уровне, препятствует использованию технологий на основе многоуровневой организации программного обеспечения (например, технологии виртуальных машин), ограничивает использование технологий проектирования с высоким уровнем абстракций, вступает в противоречие с перспективными способами проектирования и применения многих категорий элементной базы ВСС.

Вторым («обобщающим») вариантом трактовки ВПО может выступать определение ВПО как всей совокупности алгоритмического обеспечения ВСС. Этот подход хорошо укладывается в модель единого сквозного цикла проектирования ВСС, в которой важнейшее место занимают этапы высокоуровневого проектирования, широко используются абстракции организации вычислительного процесса и унифицированный взгляд на создание HW и SW компонентов системы.

Данная позиция в части создания ВПО, по мнению автора, является наиболее продуктивной и перспективной.

Отметим основные варианты трактовки ВПО, ранжированные по уровню общности (или что относят к категории программно-обеспечения различные группы разработчиков):

- Программное обеспечение только то, что «работает» в режиме run-time внутри ВСС на последовательных интерпретаторах.
- Все то, что на этапе проектирования технологически представлялось как программный продукт и было в дальнейшем реализовано в ВСС в произвольном виде (в виде аппаратных блоков, конфигурационного обеспечения, традиционного программного кода).
- Вся совокупность деятельности и продуктов по созданию алгоритмического наполнения ВСС, которые присутствуют на этапах design-time и run-time.

Перечисленные варианты идеологической позиции разработчика ВСС в области ВПО предполагают серьезные различия в организуемой технологической цепи, удельном весе этапов проектирования, влияют на требуемый состав команды разработчиков, доступные модели процесса проектирования, следовательно, на масштаб решаемых задач и эффективность решения.

### Проблемы этапа высокоуровневого проектирования ВСС

Перечислим основные проблемы, характерные для области проектирования ВСС как в части ВПО, так и в целом:

- необходимость увеличения степени повторного использования;
- совместное проектирование (co-design) ПО и аппаратуры;
- создание средств моделирования нефункциональных свойств;
- усиление роли архитектурного уровня в проектировании систем и их ПО;
- проверка достоверности и верификация;
- адаптация ПО и аппаратуры посредством использования реконфигурируемых архитектур и компонент "plug and play";
- разработка общей технологии и стандартов процесса проектирования в части семантики проектирования.

По мнению ведущих специалистов, сегодня на первый план выходят следующие направления в проектировании ВСС:

- формальная верификация;
- развитие технологии HW/SW co-design;
- повторное использование и интеграция компонентов (это было выдвинуто на первый план как главная потребность);
- совместное проектирование (co-design) вычислительной платформы и целевой функциональности ВСС;
- объединенные потоки проектирования HW/SW;
- совместное проектирование (co-design) в предметной области ВСС и архитектуры самой ВСС.

Все перечисленные выше проблемы и направления исследований лежат в области высокоуровневых этапов проектирования ВСС, к которым относятся исходное специфицирование, архитектурное и структурное проектирование, необходимые фазы верификации и оценки качества решений, формирование спецификаций для этапа реализации системы.

Важнейшая проблема, которую необходимо решить, — связь между функциональностью системы и программируемыми платформами. Для этого нужно начинать проектирование с абстракций высокого уровня, адекватно отражающих функциональность системы. Такие абстракции должны быть полностью независимо реализованными от элементной базы и способов традиционно-

го программирования (кодирования) и основаны на солидных теоретических фундаментах, которые позволят выполнять формальный анализ [4]. Необходимо научиться выбирать вычислительную платформу, которая может поддерживать функциональность, отвечающую физическим ограничениям, наложенным на окончательную реализацию ВСС.

Один из конкретных шагов в этом направлении, бесспорно, появление стандарта IEC61499, декларирующего для систем промышленной автоматизации и ВСС в качестве базового принципа аппаратно независимое проектирование (hardware-independent engineering) функциональности и активное развитие соответствующих инструментальных средств [5].

Тем не менее, очень важным является то, что большая часть перечисленных выше проблем и задач, типичных для ВПО и ВСС, недооценивается сообществом специалистов по программному обеспечению. В частности, из этого вытекает проблема «программистского перекося» в проектировании ВСС [4], которую кратко можно определить как проектирование ВСС в целом с позиций и методами проектирования программного продукта. При этом в поле зрения разработчика попадает только часть возможных вариантов построения системы, ограничения технического задания по многим аспектам выполняются путем использования неэффективных технических решений.

### Заключение

Этапы высокоуровневого проектирования ВСС приобретают сегодня все большее значение, покрывая в ряде случаев до 70% общей трудоемкости работ. Однако степень их формализации и автоматизации остается исключительно низкой. Это во многом определяется недостаточным развитием средств описания и ранжирования вычислительных технологий, фрагментарностью существующих методик проектирования ВСС.

Таким образом, в области проектирования ВСС важнейшей задачей является создание системы архитектурных абстракций, в рамках которой объединяются традиционные аппаратная и программная составляющие ВСС, при этом охватываются все уровни технических решений, которыми оперируют специалисты по вычислительной технике. Область действия архитектурных абстракций ВСС должна распространяться от нижней границы организации ВСС посредством системотехники до верхней границы проекта. При этом в качестве нижней границы выступают вычислительные механизмы, отчетливо демонстрирующие организацию вычислительного процесса в анализируемой структуре. Верхней границей

является техническое задание, содержащее вычислительную функциональность проекта в совокупности с нефункциональными составляющими (надежность, энергопотребление, массо-габаритные показатели, инструментальные и технологические требования и многое другое).

Следует признать, что в России подавляющее большинство коллективов разработчиков ВСС сегодня недостаточно высоко оценивают роль и трудоемкость этапов высокоуровневого проектирования. У них отсутствует адекватный технический язык для общения на этом уровне и корректность разграничения зон ответственности разработчиков, страдает взаимопонимание, нет должной интеграции. Они оперируют не понятием вычислительных механизмов в общем, а лишь их конкретными реализациями (по аналогии с «ассемблерными кодами», в которых трудно или невозможно проследить концептуальные моменты и решения).

Система архитектурных абстракций может позволить создать эффективный язык общения в области проектирования ВСС, качественно продвинуть вперед технологии и САПР [6], резко улучшить качество технической документации. Однако для этого необходимы усилия не только со стороны действующих разработчиков вычислительной техники, но и поддержка высшей школы в части модернизации учебного процесса на профильных кафедрах университетов [7, 8].

### Литература

1. Ferrari A., Sangiovanni-Vincentelli A. System Design: Traditional Concepts and New Paradigms // Proceedings of the 1999 Int. Conf. On Comp. Des., Austin, Oct. 1999.
2. Sangiovanni-Vincentelli A. Defining platform-based design // EEDesign, Feb. 2002.
3. Lee E. A. Embedded Software // Technical Memorandum UCB/ERL M01/26, University of California, Berkeley, CA 94720, November 1, 2001.
4. Платунов А. Е., Постников Н. П. Единое проектное пространство плюс аспектная технология — перспективная парадигма проектирования встроенных систем // Научно-технический вестник СПбГУ ИТМО. Вып. 11. Актуальные проблемы анализа и синтеза сложных технических систем. СПб.: СПбГУ ИТМО, 2003.
5. [www.nxtcontrol.com](http://www.nxtcontrol.com)
6. Sangiovanni-Vincentelli A., Lee E.A. A framework for comparing models of computation // IEEE Trans. Computer-Aided Design Integrated Circuits, 17(12):1217-1229, Dec. 1998.
7. Платунов А. Е. Подготовка специалистов в области встраиваемых вычислительных систем // Труды первого Санкт-Петербургского конгресса «Профессиональное образование, наука, инновации в XXI веке». СПб.: СПбГУ ИТМО, 2007.
8. <http://embedded.ifmo.ru>