

# Архитектура, технологические особенности и методы проектирования систем на кристалле ПЛИС

Алексей Шматок (Москва)

**В статье обсуждаются архитектура, технологические особенности и методы проектирования систем на кристалле (СнК) ПЛИС, рассматриваются способы повышения эффективности реализации приложений.**

Разработка эффективных систем – это всегда поиск компромисса в рамках технологических возможностей и ограничений элементной базы, средств и методов проектирования. Сложившаяся тенденция использования универсальных средств проектирования и высокоуровневых языков описания аппаратуры (HDL) подразумевает сокрытие особенностей аппаратной реализации, что негативно сказывается на эффективности. Явное использование

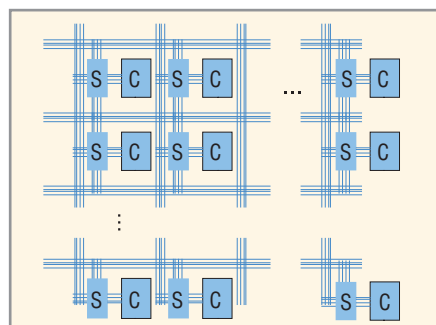


Рис. 1. Пространственная коммутационная система

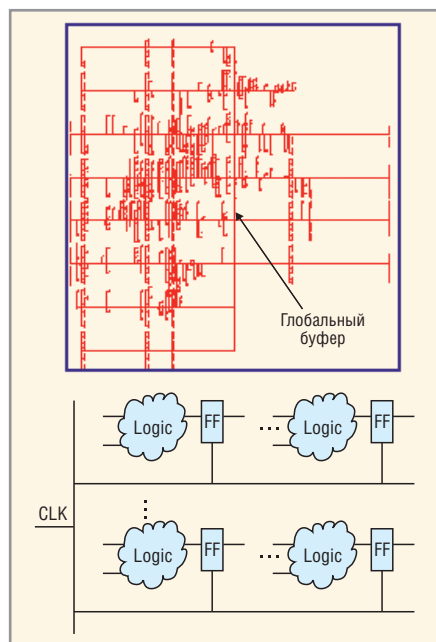


Рис. 2. Синхронизация от глобальных тактовых сигналов

архитектурных особенностей позволяет создавать более производительные решения.

## Языки описания аппаратуры

Наиболее широко используемыми языками описания аппаратуры являются VHDL и Verilog; на их основе существует множество специфических расширений и дополнений [1, 2]. Также практикуется использование традиционных языков C++, Java и т.д. для программирования описания аппаратных компонентов (генерации HDL), для моделирования, верификации и т.п., например, в виде библиотеки классов C++ SystemC [3].

В последние годы не наблюдается качественно нового развития языков и средств описания аппаратуры (HDL), несмотря на то что степень интеграции микросхем неуклонно растёт и требуются более эффективные решения [4]. В частности, время генерации конфигурационного потока для микросхем ПЛИС высокой ёмкости может занимать от нескольких часов до нескольких суток; в таких условиях отладка проекта может продолжаться многие месяцы. Решением проблемы может быть прямое программирование конфигураций ПЛИС без использования универсальных средств разработки, поставляемых производителями ПЛИС.

## Архитектура современных ПЛИС

Внутреннее устройство микросхем ПЛИС организовано на базе двумерной пространственной коммутационной среды (2D-решётки), к которой через конфигурируемые коммутационные блоки (S) подключаются различные компоненты (C), такие как логические блоки (LB), блоки памяти (BRAM), блоки арифметических операций (DSP) и многие другие, что схематично отображено на рисунке 1.

Внутренние компоненты располагаются в виде столбцов и имеют свои пространственные координаты (XY). Для следующих поколений ПЛИС анонсирована технологическая возможность использования третьего измерения для линий распространения сигналов (многослойность).

Традиционно синхронизация работы внутренних компонентов выполняется от одного или нескольких тактовых сигналов, которые распространяются либо от глобальных буферов, расположенных в центре кристалла, либо от региональных буферов, расположенных в соответствующих областях. Глобальные тактовые сигналы формируются в специальных блоках, в которых входящий тактовый сигнал умножается, делится и сдвигается по фазе. В современных кристаллах ПЛИС могут быть синтезированы тактовые сигналы с частотами порядка 1 ГГц и выше.

## Распространение сигналов и синхронизация СнК

Основополагающим элементом архитектуры ПЛИС является механизм синхронизации распространения сигналов. То, как реализована синхронизация, определяет все дальнейшие решения и существенным образом влияет на временные задержки. На рисунке 2 показан пример, в котором линии распространения глобальных тактовых сигналов расходятся по всему кристаллу от глобального буфера. Чем дальше от глобального буфера распространения тактовых сигналов (центра кристалла) располагаются компоненты, тем ниже возможная частота работы компонентов. Более эффективным является использование региональных буферов, тактовые сигналы от которых локализованы и распространяются в относительно небольшой области. Для получения максимально возможной частоты работы применяются конвейерные схемы передачи данных, в которых синхронизация выполняется между соседними компонентами (см. рис. 3).

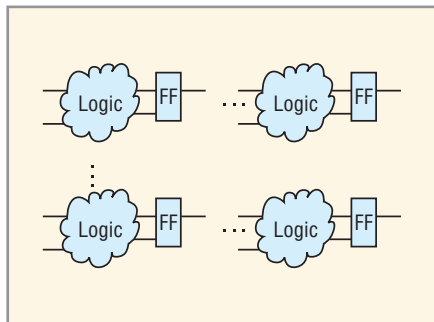


Рис. 3. Конвейерная синхронизация

### ИСПОЛЬЗОВАНИЕ ОГРАНИЧЕНИЙ

Важным инструментом повышения эффективности реализации СнК ПЛИС являются пользовательские ограничения (user constraints). Наиболее используемыми являются ограничения на время распространения сигналов и пространственные ограничения. Последние фактически уменьшают временные задержки, и их зачастую комбинируют с временными ограничениями.

### РАЗРАБОТКА ПРИЛОЖЕНИЙ

Как правило, разработка сложных приложений занимает много времени. Это время можно сократить, если использовать библиотеки компонентов, специальные средства синтеза прикладной части СнК из исходных кодов последовательных программ или из специальных упрощённых языковых конструкций. С этой целью выпускаются программные продукты Catapult-C, Mitrion-SDK, Impulse-C и др. [5–7]. Их существенным недостатком является генерация высокоуровневых HDL-описаний, без достаточных механизмов эффективной реализации: не генерируются какие-либо пользовательские ограничения; используется глобальная синхронизация; используется стандартный маршрут проектирования, занимающий достаточно долгое время; генерируемые решения не масштабируются. Имеющиеся высокоуровневые средства описания аппаратуры позволяют относительно быстро получать работоспособные, но далеко не самые эффективные решения.

### ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПРИЛОЖЕНИЙ

С повышением интеграции микросхем ПЛИС трудоёмкость разработки эффективных СнК будет возрастать, поэтому создание новых средств и методов проектирования весьма актуально. Одной из перспективных разработок является разработка открытого стандарта промежуточного языка описания

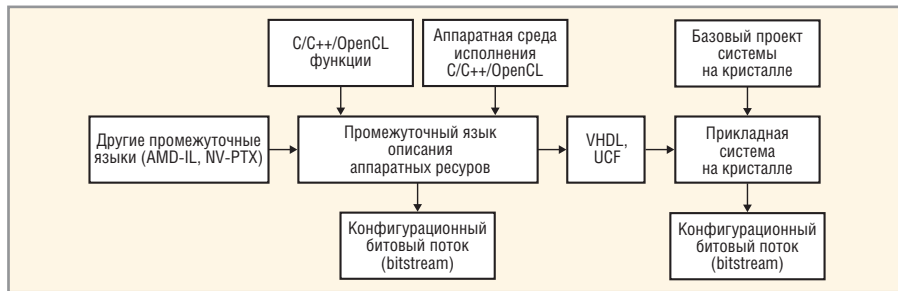


Рис. 4. Схема получения аппаратных решений (bitstream) на базе промежуточного языка описания аппаратуры (IHDL)

аппаратуры (Open Intermediate Hardware Description Language (Open-IHDL)) [8]. Промежуточный язык не скрывает особенностей реализации, а, наоборот, позволяет использовать все аппаратные возможности, при этом поддерживается масштабируемость и переносимость приложений между различными типами и поколениями ПЛИС. Прикладные задачи могут как разрабатываться непосредственно на промежуточном языке, так и создаваться путём генерации из высокоуровневого описания алгоритмов на языках C/C++/OpenCL с соответствующей аппаратной поддержкой среды исполнения. Вместе с тем обеспечивается возможность интеграции с существующими форматами и средствами разработки через трансляцию (IHDL-to-VHDL/UCF), что схематично отражено на рисунке 4.

### НОВЫЕ ПОКОЛЕНИЯ РЕКОНФИГУРИРУЕМЫХ УСТРОЙСТВ

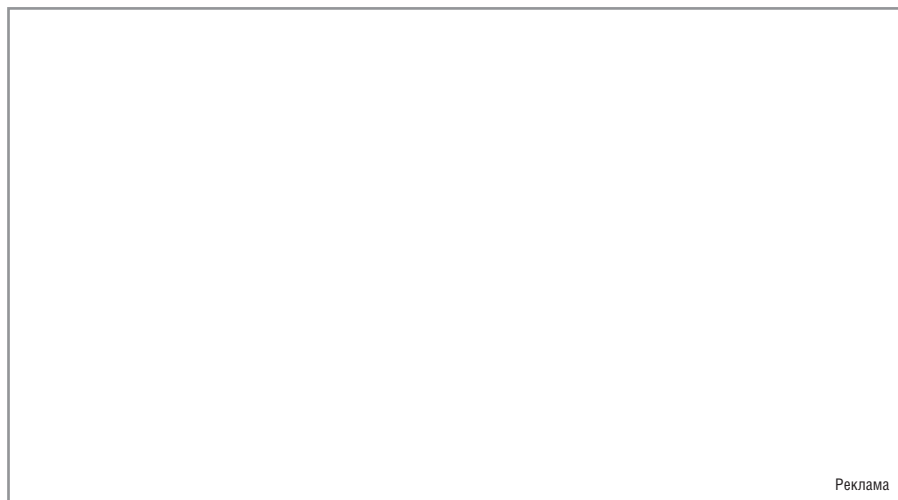
Создание новых поколений реконфигурируемых устройств происходит по мере освоения всё меньших проектных норм. На сегодняшний день анонсированы ПЛИС Xilinx и Altera с 28-нм техпроцессом [9, 10], а также заявлены 22-нм ПЛИС Achronix на техпроцессе Intel [11, 12]. Большая степень интегра-

ции приводит к усложнению внутренней архитектуры ПЛИС, изменяется схема распространения сигналов, происходит отказ от использования глобальных тактовых сигналов, вместо которых находят применение конвейерные схемы передачи данных. Развитие технологий позволяет создавать устройства, работающие на частоте порядка 1 ГГц и выше.

Перспективные разработки по созданию стандарта открытого промежуточного языка описания аппаратных ресурсов направлены на повышение продуктивности и эффективности создания приложений для ПЛИС не только сегодняшнего дня, но и будущих поколений.

### ЛИТЕРАТУРА

1. <http://en.wikipedia.org/wiki/VHDL>.
2. <http://en.wikipedia.org/wiki/Verilog>.
3. <http://www.systemc.org>.
4. [http://ru.wikipedia.org/wiki/Закон\\_Мура](http://ru.wikipedia.org/wiki/Закон_Мура).
5. <http://www.mentor.com/esl/catapult/overview>.
6. <http://www.mitrionics.com/>.
7. <http://www.impulsecelerated.com/>.
8. <http://www.recompute.ru>.
9. <http://www.xilinx.com>.
10. <http://www.altera.com>.
11. <http://www.intel.com>.
12. <http://www.achronix.com>.



Реклама