

Аппаратные ускорители приложений на базе ПЛИС

Алексей Шматок (Москва)

В статье приведён краткий обзор архитектуры платформ, развиваемых компаниями AMD и Intel, современных технологий передачи данных PCI Express, Hypertransport (HT) и Front Side Bus (FSB), затронуты вопросы проектирования систем на кристаллах (СнК) ПЛИС, особенности использования ОС Linux. Рассматриваются средства и методы быстрой разработки аппаратных решений, основанные на высокоуровневых средствах проектирования.

Современные программируемые логические интегральные схемы (ПЛИС) позволяют реализовать ряд задач на аппаратном уровне намного эффективнее по параметрам производительности, пропускной способности, латентности и т.д., нежели их программная реализация даже на самых современных микропроцессорах.

ПЛАТФОРМА

В платформах на базе архитектуры как AMD, так и Intel на сегодняшний

день имеются достаточно широкие возможности интеграции различных устройств. После появления платформ с несколькими процессорами стало возможным создание новых продуктов, устанавливаемых непосредственно в процессорный разъём, при этом возросла актуальность понятия сопроцессора.

На рисунке 1 приведена схема подключения основных элементов платформы AMD, на которой обозначены различные варианты подключения

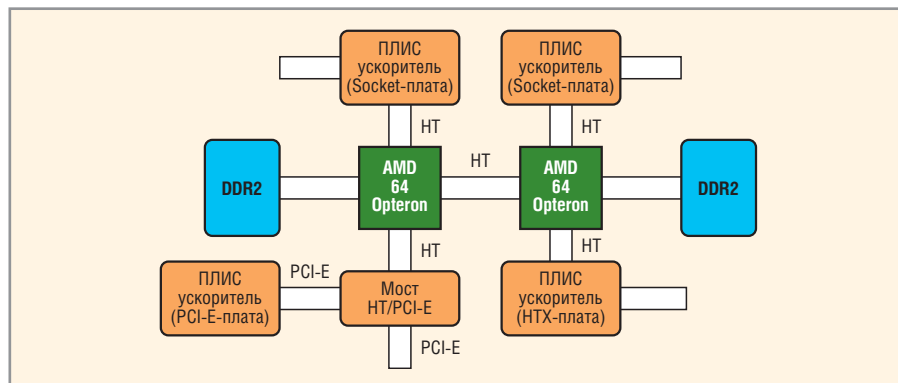


Рис. 1. Платформа AMD Torrenza

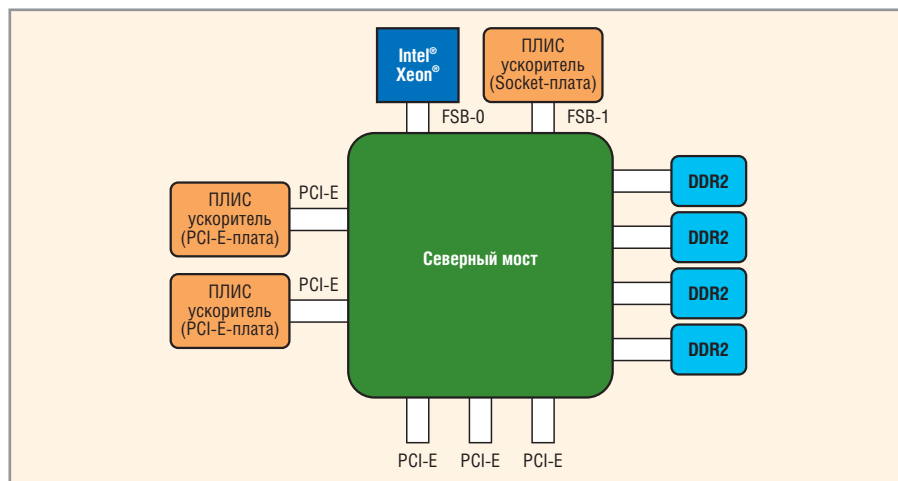


Рис. 2. Платформа Intel, технология QuickAssist AAL

ПЛИС-ускорителей [1, 2]. Для данной платформы возможны:

- плата с интерфейсом HT, которая устанавливается непосредственно в процессорный разъём (socket);
- плата с интерфейсом HT, которая устанавливается в разъём HTX [3];
- плата с интерфейсом PCI Express [4].

На рисунке 2 приведена схема подключения основных элементов платформы Intel [5]. Для данной платформы возможны следующие варианты:

- плата с интерфейсом FSB, которая устанавливается непосредственно в процессорный разъём (socket);
- плата с интерфейсом PCI Express.

Существует ряд других возможностей, таких как использование менее скоростных интерфейсов PCI, PCI-X, Ethernet и др. Очевидно, что целесообразность интеграции и выбор варианта исполнения напрямую зависит от круга задач, решаемых ускорителем на базе ПЛИС.

Другой немаловажный аспект, касающийся платформ, – сильная конкурентная борьба и, как следствие, движение в сторону максимальной гибкости, открытости стандартов и эффективности. Компании Intel и AMD стремятся привлечь на свою платформу максимальное число сторонних разработчиков.

В ближайшем будущем массовые универсальные процессоры будут состоять из десятков и даже сотен ядер на кристалле, часть из которых будет выполнять разнообразные специализированные функции. Судя по инициативам сегодняшнего дня, таким как AMD Torrenza и технологии Intel QuickAssist Accelerator Abstraction Layer (AAL) для FSB, уже сейчас «зарезервировано» достаточно возможностей для аппаратных решений третьих фирм. Потому следующий шаг – ядра внутри кристалла процессора!

ТИПОВОЕ РЕШЕНИЕ

Многие созданные продукты имеют некоторую типовую структуру. На рисунке 3 схематично отображена

типовая структура платы аппаратного ускорителя на базе ПЛИС. Основное место на плате занимают один-два кристалла ПЛИС, как правило, устанавливаются несколько кристаллов дополнительной памяти, микросхема Flash и SPROM. В ряде случаев также используются контроллеры системной памяти DDR2 (установленной на материнской плате), контроллеры внешних интерфейсов. Как показывает практика, для системных/интерфейсных ядер требуется ощутимая часть ресурсов ПЛИС, поэтому по возможности используются два кристалла – один «системный», другой «пользовательский».

ИЗВЕСТНЫЕ РЕШЕНИЯ

Среди известных на сегодняшний день решений на рынке аппаратных ускорителей на базе ПЛИС следует отметить следующие:

Для платформы AMD с интерфейсом Hypertransport:

- RCNTX-XV4, НТХ-плата [6], с двумя ПЛИС Xilinx Virtex 4;
- XD1000, НТ-плата [7], с одной ПЛИС Altera Stratix II;
- DRC RPU, НТ-плата [8], с одной ПЛИС Xilinx Virtex 4.

Для платформы Intel с интерфейсом FSB пока только анонсировано начало разработок.

Платы с интерфейсом PCI Express/PCI-X представлены достаточно давно и широко, например, семейство PCI-X плат [9] используется в качестве аппаратных ускорителей целого ряда задач.

ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ

Современные технологии передачи данных основываются на низковольтных дифференциальных сигналах LVDS [10] и позволяют достигать пропускной способности в десятки гигабайт в секунду. Использование дифференциальных сигналов повлекло за собой создание новых высокоскоростных последовательных интерфейсов, призванных заменить параллельные интерфейсы.

В современных кристаллах ПЛИС аппаратно поддерживаются технологии передачи данных на основе дифференциальных сигналов; во многих кристаллах практически все внешние проводники могут использоваться парами (N, P) для создания диффе-

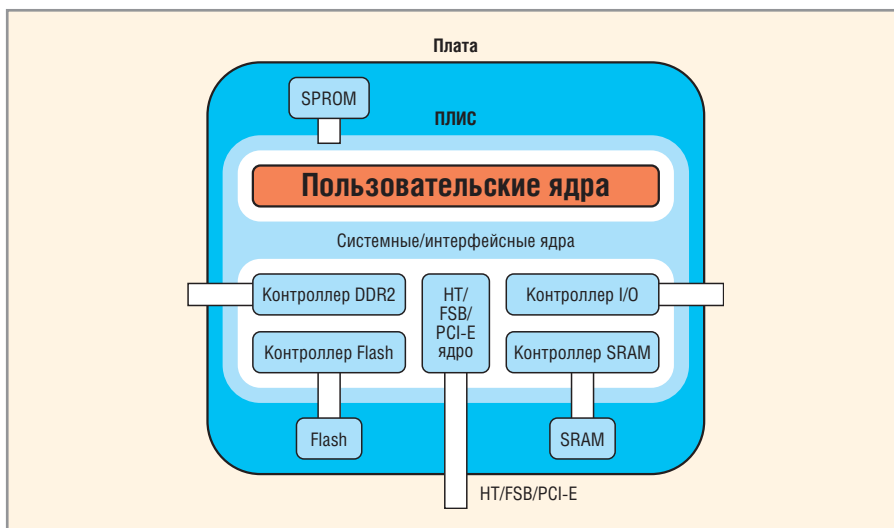


Рис. 3. Типовая структура платы ускорителя на базе ПЛИС

ренциальных линий передачи данных. В ПЛИС некоторых серий, например Xilinx Virtex 5 LXT, реализованы дополнительные, «усиленные» гигабитные приёмопередатчики (gigabit transceivers (GT)), для которых только по одной паре возможна передача до 3,125 Гб/с.

Технология передачи данных Hypertransport ориентирована на содинения типа процессор-процессор и процессор-периферия. Для НТ-соединения используются высокоскоростные дифференциальные сигналы:

- CLK, тактовый;
- CTL, управляющий (определяет тип передаваемого пакета);
- CAD[n-1:0] – (command, address, data); если CTL=1, то передаётся команда, в остальных случаях – данные и адреса. Параметр n определяет ширину канала; возможные варианты: 2, 4, 8, 16 и 32 бита; ширина канала в разных направлениях может быть различной.

Также используются низкоскоростные сигналы сброса, статуса, режима stand-by.

На рисунке 4 приведена схема соединения двух НТ-устройств, на которой RX обозначает приёмник, TX – передатчик. На базе соединений точка-точ-

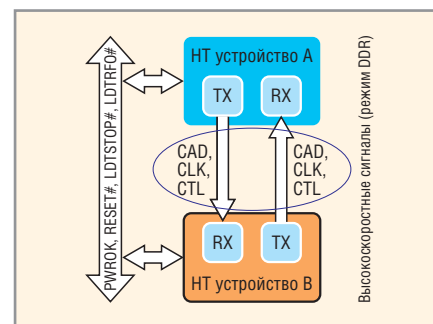


Рис. 4. Схема соединения двух НТ-устройств

ка, кроме конечных устройств, строятся туннели, коммутаторы, мосты и т.п.

Передача данных выполняется пакетами, при этом поддерживаются различные протоколы взаимодействия конечных устройств, такие как:

- отложенные транзакции (posted);
- немедленные транзакции (non-posted);
- атомарные операции (atomic operation);
- транзакции барьерной синхронизацией (flush/fence transaction).

Пример типовой архитектуры системы для обработки сетевого трафика приведён на рисунке 5, где процессорные элементы с высокой эффективностью реализуются на ПЛИС.

Для создания систем, использующих шину Hypertransport, кроме «покупных» IP-ядер, можно применить

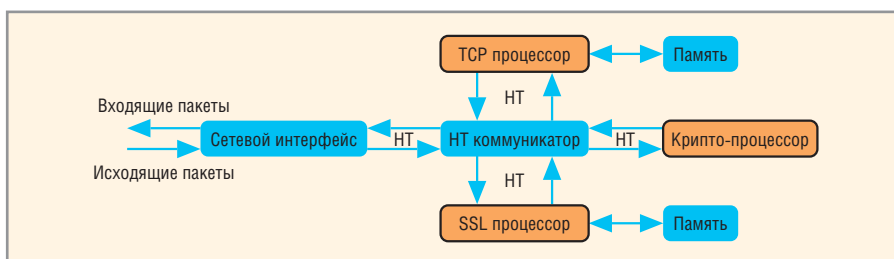


Рис. 5. Архитектура системы обработки сетевого трафика

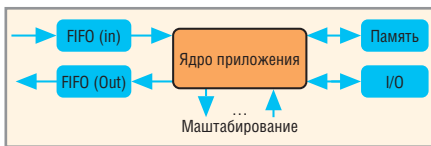


Рис. 6. Типовая схема сопроцессора

открытые реализации HT-ядер, например [11, 12]. Реализация открытого HTX-ядра конечного устройства (HT-save) на кристалле ПЛИС Virtex4 fx60 занимает порядка 16% логических ячеек.

Технология передачи данных PCI Express по ряду объективных причин получила значительно большее распространение, нежели Hypertransport, при этом обе технологии имеют существенное сходство и могут быть эффективно использованы для решения многих прикладных задач. Однако, как было видно из предыдущих схем, отображенных на рисунках 1, 2, устройство с PCI-E-интерфейсом может быть интегрировано в любую платформу, что делает использование PCI-E более универсальным решением.

В некоторых современных кристаллах ПЛИС, например Virtex 5 LXT, уже имеется встроенное ядро (hard core) контроллера PCI-E, работающее через гигабитные приёмопередатчики GT. В отличие от ядер, синтезируемых на логических ячейках (soft cores), встроенные аппаратные блоки в минимальной степени требуют использования ресурсов ПЛИС.

Для эффективной реализации прикладных задач требуется высокоскоростная среда передачи данных с низкой латентностью, подразумевающая интенсивный обмен данными между основным вычислительными компонентами системы (процессором, памятью и сопроцессором). Для таких целей, несомненно, лучше использовать технологии Hypertransport (платформа AMD) и FSB (Intel).

РАЗРАБОТКА ПРИЛОЖЕНИЙ

Приобрести или создать адекватную современным технологиям аппаратную платформу недостаточно, поскольку прикладную часть необходимо эффективно запрограммировать, и на это, как правило, отводится очень мало времени и ресурсов. В связи с этим развиваются высокоуровневые средства «быстрой» разработки.

В большинстве случаев, для разработки аппаратных систем уже

имеется «эталонная» программная реализация (Reference Code) если не всей задачи, то некоторых её фрагментов. Полезным инструментом становится транслятор из программного описания на процедурном языке в аппаратное software-to-hardware или, как в большинстве случаев, C-to-HDL. На сегодняшний день используются такие инструментальные средства, как: Impulse C [13], Catapult C [14], Mitrion C [15], Handel-C [16] и др.

Стоит отметить использование программных расширений для таких мощных пакетов, как MATLAB, которые позволяют быстро создавать и отлаживать схемотехнические решения, не вникая в HDL-программирование.

ОТКРЫТЫЙ ИСХОДНЫЙ КОД

Приступая к созданию систем на кристаллах ПЛИС, необходимо иметь программную реализацию её функционала (желательно, на языке Си). К счастью, ситуация сложилась так, что для большинства задач можно найти проект с открытым исходным кодом (Open Source). Существуют различные поисковые системы непосредственно по исходным кодам, например <http://www.google.com/code-search>; найти подобные «поисковики» можно по запросу «source code search».

ОС Linux

Операционная система Linux создавалась разработчиками и для разработчиков, и неважно, создаётся новый программный продукт или аппаратно-программный комплекс, – лучшей платформы для процесса разработки и отладки трудно найти. Например, работа с платами RSP-5XX [17] с PCI-X интерфейсом происходит практически без затруднений: используя утилиты lspci, lshw, setpci и несколько простых скриптов, можно реализовать минимально достаточный функционал для загрузки конфигураций пользовательских ПЛИС, организации ввода/вывода и проведения тестирования. Для более сложных режимов работы используются модули ядра (драйверы).

Следует отметить, что в ядре Linux не так давно появилась поддержка ряда аппаратных устройств, предназначенная для ускорения работы некоторых функций самого ядра (пока, в основном, связанных с шифрова-

нием), но тенденция наметилась, и в скором будущем «тяжелые» системные задачи будут решаться сопроцессором.

ЭФФЕКТИВНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЙ

Современные тенденции развития информационных технологий не только определяют появление новых задач и приложений, но, как показывает практика, уже известные задачи иногда получают более эффективное решение на новом технологическом уровне.

Компании – разработчики аппаратных ускорителей на базе ПЛИС обычно приводят списки самых разнообразных приложений [9]; иногда приводятся цифры по ускорению в 10, 100 и даже 1000 раз.

Резюмируя изученные материалы, можно обобщить некоторые рекомендации по разработке и программированию ускорителей на базе ПЛИС:

- RTL-уровень прикладных IP-ядер должен реализовывать параллельно-конвейерную схему обработки потока данных/инструкций с минимальной временной задержкой между ступенями конвейера;
- обработка потока данных/инструкций внутри прикладных IP-ядер должна происходить с интенсивностью, не меньшей интенсивности их поступления;
- вероятность переполнения буферов входящих/выходящих пакетов данных должна стремиться к нулю;
- реализация приложения должна быть легко масштабируемой и обеспечивать соответствующий прирост производительности вычислительной системы при увеличении аппаратных ресурсов.

На рисунке 6 приведена типовая схема сопроцессора, иллюстрирующая приведённые рекомендации.

Выводы и рекомендации

Время массового использования аппаратных ускорителей на основе ПЛИС пришло. Современные технологии, основанные на низковольтных дифференциальных сигналах (LVDS), позволяют достигать скорости передачи данных в десятки гигабит/с. Широко распространённая платформа x86 открыта для интеграции аппаратных решений третьих фирм. Современные кристаллы ПЛИС позволяют создавать сложные

цифровые схемы и поддерживают практически все известные высокоскоростные интерфейсы. Высокоуровневые средства разработки аппаратно-программных решений, а также большое число проектов с открытым исходным кодом позволяют значительно уменьшить сроки создания конечных продуктов.

ОС Linux является эффективной платформой, как для отладки и тестирования разрабатываемых аппаратно-программных комплексов, так и для создания готовых решений.

Имея в своей системе ускоритель на базе ПЛИС, пользователь получает

не просто ускоритель определённого приложения, но и эффективно расширяемую аппаратную платформу.

ЛИТЕРАТУРА

1. http://www.hypertransport.org/docs/pres/HTX_&_Torrenza_01-28-07.pdf.
2. http://www.xtremedatainc.com/pdf/NathanWoods_FPGA_2007_presentation.pdf.
3. http://www.hypertransport.org/tech/tech_htx_main.cfm?m=9.
4. <http://www.pcisig.com/specifications/pci-express>.
5. http://www.intel.com/products/processor/xeon5000/index.htm?iid=servproc+body_xeon5000.
6. <http://www.celoxica.com/techlib/files/CEL-W06112119BY-517.pdf>.
7. <http://www.xtremedatainc.com>.
8. <http://www.drccomputer.com>.
9. <http://www.sensornetworks.com/Products/Acceleration>.
10. <http://en.wikipedia.org/wiki/LVDS>.
11. http://opencores.gds.tuwien.ac.at/projects.cgi/web/ht_tunnel.
12. <http://www.ra.informatik.uni-mannheim.de/coeht>.
13. <http://www.impulsec.com>.
14. <http://www.mentor.com>.
15. <http://www.mitrion.com>.
16. <http://www.celoxica.com>.
17. <http://www.rosta.ru>.

