

Разработка приложений для СнК SmartFusion2 с использованием Libero SoC и SoftConsole.

Часть 10. Знакомьтесь, RISC-V

Петр ПОЗДНЯКОВ
Pozdnyakov@actel.ru

В статье рассмотрены предпосылки появления новой процессорной архитектуры RISC-V и пример проекта с использованием IP-ядра процессора RISC-V разработки корпорации Microsemi.

В ноябре 2016 года корпорация Microsemi представила IP-ядро CoreRISCV_AXI4 процессора новой процессорной архитектуры RISC-V. Инженеры, работающие с ПЛИС и СнК компании Microsemi, получили возможность использовать в проектах процессор новейшей архитектуры RISC-V в том числе для последнего семейства радиационно стойких микросхем линейки RTG4.

Казалось бы, на рынке уже есть целый ряд процессорных архитектур, каждая из которых представлена целым набором процессоров различных производителей. Если вспомнить ряд: x86, 8051, PowerPC, ARM Cortex-M0, ARM Cortex-A9 и другие решения, может показаться, что для каждого существующего и вновь создаваемого приложения уже выпускается процессор с нужными характеристиками.

Однако когда дело доходит до выбора процессора под конкретное приложение, оказывается, что все обстоит не так радужно. Одни процессоры недостаточно быстрые, другие требуют слишком большой электрической мощности для питания, третьи предполагают покупку дорогостоящей лицензии для их производства, четвертые плохо документированы, у пятых дорогие

инструменты разработки приложений, шестые не обладают необходимым уровнем радиационной стойкости и т. д. В результате, при всем кажущемся многообразии архитектур и их полупроводниковых воплощений, разработчику приходится выбирать из пары-тройки процессоров, как правило, жертвуя какими-то ценными качествами создаваемого устройства из-за ограничений выбранного процессора.

Но являются ли описанные трудности достаточным основанием для создания новой процессорной архитектуры? Скептики усомнятся: разработкой и производством процессоров в настоящее время занимаются такие солидные компании, как Intel, AMD, Atmel, ARM Limited. Компаниями накоплен большой опыт в столь непростом деле, по всему миру собраны лучшие специалисты. Вероятно, данные компании смогли создать шедевры в мире процессоров, чьи характеристики к тому же на протяжении десятилетий постоянно улучшались при переходе от поколения к поколению. Тратя на это миллиарды долларов, полученных от реализации процессоров предшествующих поколений. Можно ли за ограниченное время и не имея такой солидной материальной базы создать решение, сопоставимое по качеству с продукцией перечисленных гигантов индустрии?

Что ж, для начала присмотримся к шедеврам повнимательнее. Для примера обратимся к инструкции AAA процессора Intel x86, которая должна выполняться после команды процессора ADD, то есть сложения операндов. Смысл производимых процессором действий при выполнении данной инструкции представлен во множестве источников, в частности в [1, 2]. Для удобства восприятия процессором в ходе выполнения инструкции, часто представляют на языке C (рис. 1).

На человеческом языке это звучит примерно так: если в результате сложения младшая тетрада регистра AL содержит значение больше 9 и флаг AF равен 1, то необходимо:

- 1) к содержимому регистра AL прибавить 6;
- 2) к содержимому AH прибавить 1. Флаги AF и CF установить в 1.

В противном случае сбросить флаги AF и CF в 0.

После чего обнулить старшую тетраду AL.

Еще раз оговоримся, описанная последовательность действий и представленный на рис. 1 код — это не программа, написанная уставшим программистом, а инструкция процессора, то есть элемент базовой системы команд процессора, которой, по все видимости, должна быть присуща некая простота и атомарность.

Является ли система команд, включающая такие инструкции шедевром технической мысли? Можно ли сделать систему команд процессора проще? С технической точки зрения ответы на эти вопросы давно получены. Но кроме технической точки зрения есть и экономическая. Так уж сложилось исторически, что компаниям IBM и Intel удалось в нужный момент представить на рынок удачное системное решение — персональный компьютер и таким образом «поймать волну» компьютеризации в мире. После того как большое количество пользователей привыкли к IPM PC, а производители программного обеспечения начали массово писать ПО для процессоров x86, недостатки «отдельных команд» процессорной архитектуры уже мало кого интересовали. Пока удвоение производительности происходило каждые полтора года, а привязанность компьютера к розетке в стене была нормой, никто не думал менять привычный x86 на что-то иное.

Однако все когда-нибудь заканчивается. Продажи ПК на основе x86 падают из года

```
if((AL & 0xF) > 9 || AF == 1) {
    AL = AL + 6;
    AH = AH + 1;
    AF = 1;
    CF = 1
}
else {
    AF = 0;
    CF = 0;
}
AL = AL & 0xF;
```

Рис. 1. Инструкция AAA процессора Intel x86

в год (рис. 2) [3]. Аналитики уже подвели черту под эрой ПК и объявили о наступлении эры планшетных компьютеров. Для планшетного компьютера, в отличие от персонального, потребление является ключевым параметром, и наличие в системе команд процессора x86 инструкций типа AAA, которые для выполнения требуют присутствия на полупроводниковом кристалле лишних тысяч транзисторов, потребляющих драгоценные ватты мощности батарей привело к тому, что среди устройств новой эры места для процессоров x86 не нашлось. При том, что ниша портативных устройств оказалась в сотни раз более просторной, чем ниша персональных компьютеров. Продажи всех портативных устройств с процессором архитектуры ARM на борту подобралась к 16 млрд (планшетные компьютеры и смартфоны на основе ARM-процессоров). Сравните эту цифру с жалкими 365 миллионами ПК в самый удачный для персональных компьютеров 2011 год!

Но, может быть, архитектура RISC процессоров ARM, столь популярная в носимых устройствах, устранила все недостатки x86 и является пределом совершенства? С технической точки зрения все действительно довольно неплохо: процессоры архитектуры ARM обладают высокой эффективностью при довольно низком потреблении и занимают не так уж много места на кристалле. А вот с экономической точки зрения для крупных производителей электроники и для нас, потребителей, все опять «не слава богу»! Дело в том, что данная процессорная архитектура является проприетарной, то есть для изготовления процессоров данной архитектуры производителю необходимо купить лицензию у правообладателя. А правообладатель стремится увеличить свою прибыль и держит цену на оптимальном с точки зрения его бизнес-модели уровне, который часто оказывается высоким и даже непреодолимым порогом для новых проектов «Интернета вещей», требующих недорогого процессора.

Например, небольшая компания, разработала новое IoT-устройство, скажем «умную» лампу, выполняющую голосовые команды, или «умную» подушку, анализирующую ритм дыхания и сердцебиения во время сна [4], и планирует открыть серийный выпуск своего инновационного продукта. Экономический расчет проекта показал, что если стоимость процессора, применяемого в устройстве, будет меньше некоторой величины (обозначим ее условно в \$1), то стартап станет успешным, изделия разойдутся по миру миллионными тиражами, проект окупится и принесет прибыль. Если же цена процессора окажется больше \$1, то проект просто не сможет начаться.

Стоимость лицензии самого доступного процессора с архитектурой ARM IP ARM Cortex-M0 — \$40 000. Не у каждой молодой инновационной компании найдется такая сумма для запуска в производство первой партии устройств. Конечно, есть коммерче-

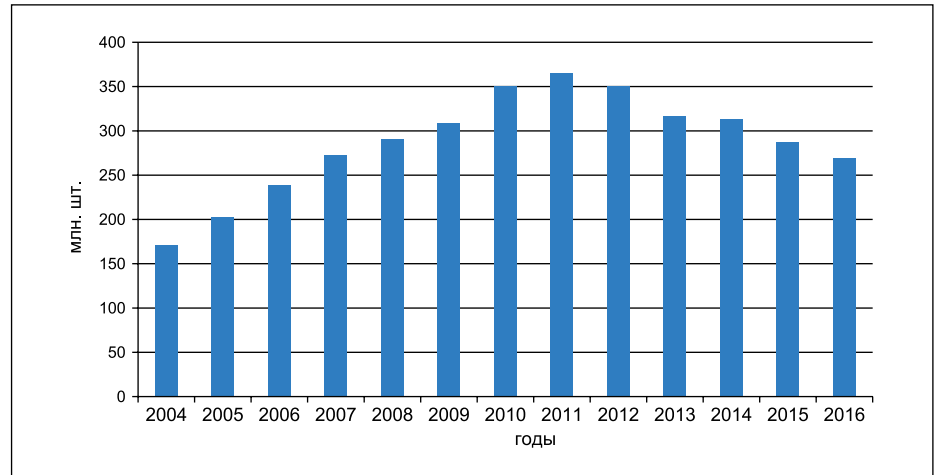


Рис. 2. Продажи персональных компьютеров с процессорами архитектуры Intel x86

ские банки, различные интернет-сообщества и проекты, позволяющие получить деньги на стартап, но очевидным остается факт, что отсутствие на рынке свободного, не проприетарного процессора тормозит появление новых гаджетов, развитие новых проектов «Интернета вещей» и таким образом задерживает технический прогресс в целом. Причем закрытость архитектуры пагубно сказывается на развитии рынка не только аппаратуры, но и программного обеспечения и создания интегрированных сред разработки для разработки ПО гаджетов. Проприетарность процессорной архитектуры ARM ведет к тому, что приобрести необходимый комплект документации для создания средств разработки ПО могут только очень состоятельные компании. В результате на мировом рынке существует очень небольшое число фирм — изготовителей интегрированных сред разработки, конкуренция между ними крайне низка, они создают платные и довольно дорогие инструменты разработки ПО, что также становится барьером для развития инноваций.

Эта проблему отчетливо осознают крупные производители коммерческой электроники. Для них проприетарность процессоров — это миллионы невыпущенных устройств и миллиарды долларов недополученной прибыли.

Проблемы с проприетарными ISA испытывают компании, выпускающие не только коммерческую электронику, но и аппаратуру промышленного и даже космического назначения. Предположим, корпорация Microsemi хочет реализовать процессор ARMxx в новой микросхеме SnK, разрабатываемой для применения в космосе. Для этого требуется повысить стойкость процессора к воздействию космической радиации. Специалисты компании определили, какие узлы и блоки процессора наиболее подвержены негативному воздействию ионизирующих излучений, выяснили, как можно увеличить их стойкость, и для проведения необходимых работ у корпорации

есть все силы и средства. Однако даже обладая всеми нужными знаниями и средствами, Microsemi не может ничего сделать. Лицензия ARM не разрешает спроектировать собственное радиационно стойкое ядро процессора на основе существующего, а лишь дает право на использование оригинального, не стойкого к воздействию радиации дизайна.

Таким образом, какую область электроники ни возьми, от бытовой электроники до космической, проприетарность ISA тормозит развитие техники. Поэтому как только в 2010 году появился коллектив разработчиков, поставивший цель создать процессор с открытой архитектурой, многие мировые гиганты, такие как Google, IBM, SAMSUNG, nVIDIA, NXP, Western Digital, и многие другие [5] его поддержали.

Хотя в исследовательскую группу вошли создатели RISC-архитектуры, в частности Дэвид Паттерсон, группа изначально не отдавала предпочтение архитектуре RISC, а начала работу с анализа достоинств и недостатков всех имеющиеся на тот момент процессорных архитектур. Прошлись от простейшей RISC до VLIW и обнаружили следующую особенность: архитектура процессора, а именно архитектура набора команд (ISA), довольно незначительно влияет на производительность. То есть если переключить производительность с разными ISA к одному уровню технологического процесса, используемого при производстве чипов, окажется, что разница между процессорами не столь велика. А если ISA не влияет на этот важнейший параметр процессора, зачем вообще нужно такое разнообразие ISA? Нельзя ли создать одну-единственную для всех процессоров, выпускаемых в мире?

Исследования группы показали, что можно и даже нужно! С технической, коммерческой и потребительской точек зрения переход на единую ISA в планетарном масштабе вполне оправдан, поскольку ведет к унификации средств разработки программного обеспечения, повышению конкуренции и качеству

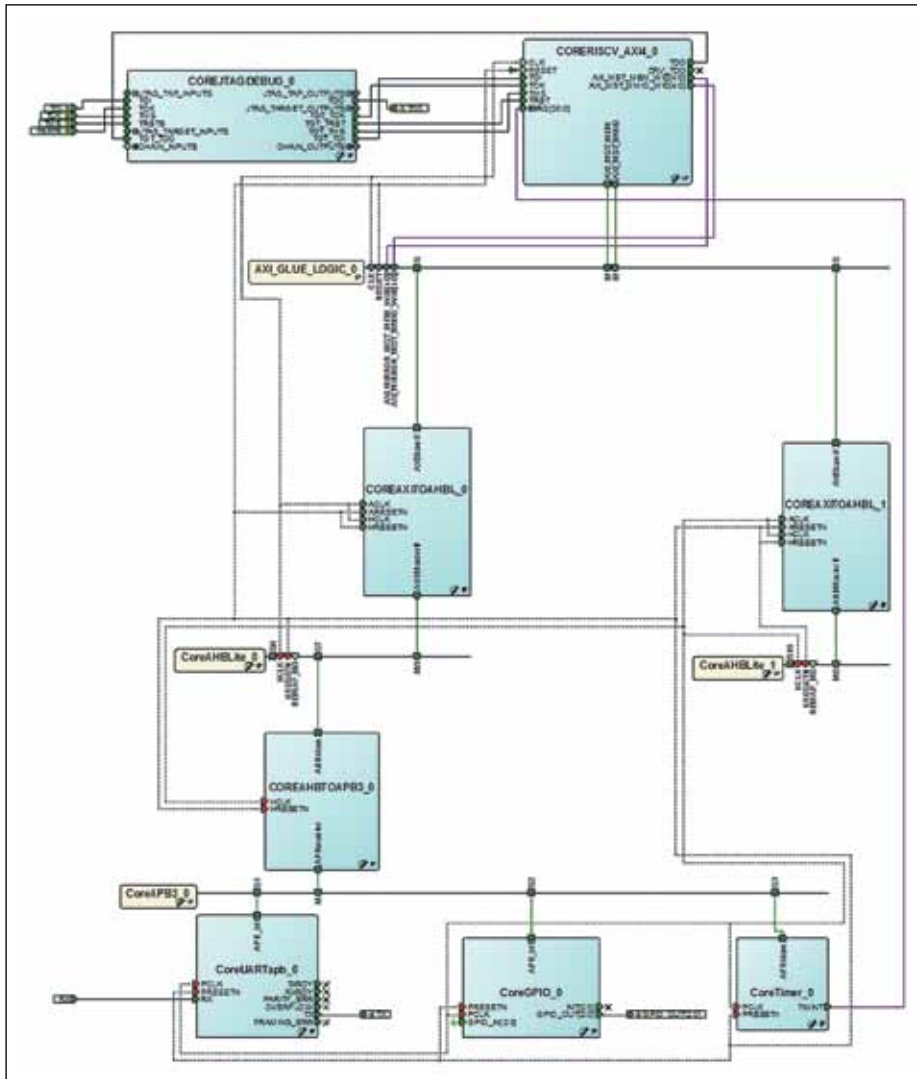


Рис. 3. Верхний уровень проекта Libero SoC с использованием процессорного ядра coreRISCV_AXI4

Третий фундаментальный принцип, который был положен в основу новой ISA, — расширяемость и масштабируемость. Следуя данному принципу, мы получим семейство процессоров от чипов с сокращенным набором 32-битных регистров общего назначения для встраиваемых приложений с ультрамалым потреблением и относительно низкой производительностью до x64-битных 128-битных процессоров средней и высокой вычислительной мощности. При работе со всеми чипами линейки можно будет использовать единую среду разработки, программатор и одни и те же приемы программирования.

Следует отметить, что целью организации RISC-V Foundation является создание не еще одного нового процессора, а спецификации архитектуры системы команд (ISA), открытой и доступной для использования всеми разработчиками оборудования и ПО в мире. При этом конкретные реализации процессора, как ASIC, так IP-ядра для ПЛИС, созданные на основе спецификации, могут быть платными или бесплатными и существенно отличаться по своим характеристикам: одни представители семейства будут иметь невысокую производительность и сверхмалое потребление, другие, напротив, обретут высокую вычислительную мощность и соответствующее потребление.

Итак, перечислим все плюсы и минусы, которые ожидаются от перехода на единую открытую архитектуру набора команд. Положительными следствиями перехода к единой ISA являются:

1. Появление промышленного стандарта для всех типов вычислительных устройств.
2. Снижение стоимости процессора и возможность сделать его подходящим для все более широкого круга задач и устройств.
3. Повышение уровня конкуренции на рынке аппаратного и программного обеспечения и снижение стоимости продукции для конечного потребителя.
4. Увеличение доступности библиотек открытых проектов Open Core Designes, а значит, и расширение возможностей повторного использования кода.
5. Упрощение процесса обучения разработке ПО и аппаратуры.
6. Ускорение поиска и устранение ошибок в коде ПО.
7. Сокращение времени выхода на рынок новых продуктов.
8. Увеличение количества инновационных продуктов в нашей повседневной жизни.
9. Появление дополнительных трудностей для правительственных организаций различных государств сделать «потайные двери» в коммерческих устройствах.
10. Увеличение «продолжительности жизни» ПО бытовых приборов. В данный момент для коммерческих устройств на основе RISC-V указывается срок службы ПО ориентировочно 50 лет.

программного кода. Если раньше разработка ПО для бортовых компьютеров космических аппаратов или самолетов была уделом узкого круга посвященных, чья исключительность основывалась на знании недочетов конкретных процессорных архитектур, допущенных к применению в данной отрасли, а также умения обходить эти недочеты, то при переходе на единую ISA разработка ПО перестает быть делом избранных. Программист, еще вчера разрабатывавший встраиваемое ПО для стиральных машин и кофеварок, сможет быстро переключиться на создание ПО для смартфона, космического аппарата и бортового компьютера авиалайнера.

Несмотря на кажущийся слишком смелым отказ от множества существующих процессорных архитектур и переход к единой ISA в глобальном масштабе, история показывает, что аналогичные обращения к единым стандартам происходили уже не раз. В качестве примера можно вспомнить историю создания единой системы железных дорог. Первые железные дороги появлялись на промышленных и горнорудных предпри-

ятиях. Параметры пути, такие как ширина колеи, размеры и даже материал рельсов и шпал, были адаптированы под условия, существовавшие на конкретном предприятии. Однако общая цель производителей и потребителей — перемещение грузов на дальние расстояния, при создании сети железных дорог вынудила перейти на единые стандарты, по крайней мере в рамках национальных государств и их союзов. Еще один пример формирования глобальных стандартов — создание Международной системы единиц (СИ).

Другой важный вывод, который был сделан командой исследователей при разработке открытой ISA, заключался в том, что поскольку ISA не является основным фактором, определяющим производительность, то для новой процессорной архитектуры необходимо взять или придумать самую простую ISA, то есть с минимальным набором команд. Воплощение такой ISA в полупроводнике будет занимать меньше места на кристалле, что приведет к удешевлению конечного продукта и положительно скажется на электропотреблении.

Негативными факторами пока можно назвать незрелость инфраструктуры разработки проектов RISC-V и связанные с этим трудности перехода на новую архитектуру.

Отчасти такой недостаток уже преодолен. Microsemi предлагает три IP-ядра процессора RISC-V и средство разработки проектов встроенного программного обеспечения SoftConsole 5.1. На популярном среди разработчиков сайте демонстрационных проектов GitHub представлена целая коллекция дизайнов для процессора с новой ISA [6]. Коллекция так разрослась, что для проектов RISC-V, выполненных на основе ПЛИС и СнК Microsemi, пришлось создавать отдельную ветку [7].

Имплементация IP-ядра CORERISCV_AXI4 требует приблизительно 10 000 логических элементов 4LUT-DFF, а с необходимой для взаимодействия с внешним миром периферией (coreUART, coreIO) — не менее 12 000. Из семейств SmartFusion2/Igloo2 подойдут чипы, имеющие более 10 тыс. логических элементов, т.е. начиная от M2S025/M2GL025 и выше.

Проекты с использованием IP-ядер RISC-V содержат многоступенчатое соединение процессорных шин AXI – AHBLight – APB3, поэтому сложны для самостоятельной сборки с нуля. Гораздо проще и быстрее создавать собственный проект, имея в качестве



Рис. 4. Результат работы программы

образца готовый работающий. Для первого знакомства с RISC-V возьмем проект, разработанный для чипа M2S090 - SmartFusion2 Evaluation Kit sample RISC-V Libero project(s) [8]. Для изучения достаточно открыть проект СнК в среде разработки Libero SoC 11.8SP2, проект встроенного ПО в среде SoftConsole 5.1 (рис. 3). Процесс разработки встроенного программного обеспечения для нового процессорного ядра практически не отличается от такового для процессора ARM Cortex-M3 СнК SmartFusion2, описанного в начальных статьях цикла [9–11]. В результате работы нашей программы в окне терминала появляется приветственное сообщение, а светодиоды начинают мигать в соответствии с алгоритмом, реализованным в программе ВПО (рис. 4).

Проекты СнК и ВПО, описанные в данной статье, доступны по ссылке [12].

Литература

1. www.club155.ru/x86cmd/AAA
2. www.fermimn.gov.it/linux/quarta/x86/aaa.htm
3. www.rationalnumbers.ru/tags/mir/
4. www.kickstarter.com/projects/modem/the-sunrise-smart-pillow-sleep-smart-wake-naturall
5. www.riscv.org/members-at-a-glance/
6. www.github.com/riscv
7. www.github.com/RISCV-on-Microsemi-FPGA
8. www.github.com/RISCV-on-Microsemi-FPGA/M2S090-Security-Eval-Kit
9. Поздняков П. Разработка приложений для СнК SmartFusion2 с использованием средств разработки Libero SoC и SoftConsole // Компоненты и технологии. 2016. № 1.
10. Поздняков П. Разработка приложений для СнК SmartFusion2 с использованием средств разработки Libero SoC и SoftConsole. Часть 2. Создание исполняемой версии прошивки // Компоненты и технологии. 2016. № 2.
11. Поздняков П. Разработка приложений для СнК SmartFusion2 с использованием средств разработки Libero SoC и SoftConsole. Часть 3. Разработка встраиваемого ПО в среде SoftConsole 4.0 // Компоненты и технологии. 2016. № 3.
12. www.drive.google.com/file/d/189_OsuZ6i0TXILFJghyHtTh9tRF5a-xt/view?usp=sharing