

## Вторая жизнь асинхронных процессоров

Леонид Черняк

Тактовая частота стала настолько обязательным атрибутом в имидже процессоров, но частота далеко не единственный показатель производительности, особенно теперь, когда на рынке осталось лишь считанное количество различных процессорных архитектур.

Если вы хотите заставить массу людей целенаправленно передвигаться в заданном направлении, нет ничего лучше, чем, воспользовавшись вековым армейским опытом, принудить толпу шагать в ногу стройными рядами. При необходимости ускориться, можно скомандовать «шире шаг» или увеличить частоту (правда, последнее действие для армейского строя возможно только в теории — частота шага лимитирована собственной частотой колебания маятника-ноги). И все же ускорение имеет предел, определяемый не физическими возможностями идущих строем, а синхронностью. Каждый из них имеет потенциал для более быстрого движения, но на жесткой связке это сделать невозможно. Для того чтобы реализовать потенциал, нужно перейти на асинхронный шаг или бег — при условии, что будет предложен некий альтернативный механизм управления. Еще одним ограничением для синхронного перемещения является численность идущих строем людей; понятно, что сверх какого-то предела всем шагать в ногу не удастся. Эти банальные рассуждения относительно ограничений, накладываемых синхронностью, как ни удивительно, вполне можно отнести к процессорам. Мы привыкли к тому, что компоненты большинства процессоров «шагают в ногу», но есть основания предположить, у традиционной синхронной схемы есть альтернатива в лице асинхронной.



Практически все известные процессоры синхронны. Для современного процессора работать в синхронном режиме также естественно, как нам с вами «говорить прозой». В процессоре в роли командира, задающего темп, выступают тактовый генератор или системные часы. Сегодня только и слышишь: «частота, частота...». Тактовая частота стала настолько обязательным атрибутом в имидже процессоров, что даже школьники младших классов называют соответствующую цифру, говоря о своих домашних «компах», едва ли понимая ее смысл.

Перекокс всеобщего интереса в сторону частоты, которая является, несомненно, важным, но далеко не единственным показателем производительности, весьма характерен, особенно теперь, когда на рынке осталось лишь считанное количество различных процессорных архитектур. Но не всегда выбор был так ограничен; на заре компьютерной эры богатство архитектурных стилей было больше, хотя и тогда основная масса компьютеров имела центральный тактовый генератор. Впрочем, на общем «синхронном» фоне были отдельные альтернативные проекты. В качестве удачного примера компьютера «без часов» можно привести построенный в начале 50-х годов в Иллинойском университете ORDVAC (Ordnance Variable Automated Computer), который с успехом применялся для управления артиллерийским огнем. В истории компьютеров оставил заметный след асинхронный проект IAS, выполнявшийся под руководством самого Джона фон Неймана. В течение долгих лет асинхронными компьютерами занимались в Манчестерском университете, где был построен MU5. Однако позже идеи построения асинхронных процессоров оказались вне сферы интересов крупных производителей; о них почти забыли.

В середине 90-х годов интерес к почти забытым асинхронным компьютерам пробудился вновь. Импульсом к реинкарнации послужил программный документ с внушительным названием National Roadmap for Semiconductors, подготовленный американской

Ассоциацией производителей полупроводников в 1997 году. Этот документ содержал предупреждение о грядущем кризисе, который может возникнуть спустя какое-то время и затронуть всю полупроводниковую индустрию. По мнению большинства членов Ассоциации, угроза кроется не в таких наиболее муссируемых и, на первый взгляд, действительно самых важных и широко обсуждаемых факторах, как достижение теоретического предела ширины проводника или той же тактовой частоты.

Физика и технологии демонстрируют свои возможности преодолевать многие, казалось бы, непреодолимые барьеры. Однако основным пределом роста производительности в будущем может стать именно синхронность — точнее, не сама синхронность, а вызываемая ею сложность архитектуры. Эта сложность неподвластна ни физике, ни технологиям. Конечно же, у современных полупроводниковых технологий резерв для роста производительности еще очень велик, впадать в панику не стоит, активные антикризисные мероприятия пока еще не актуальны. Кризис, если и разразится, то определенно не завтра. Между тем, архитектурный кризис нужно решать архитектурными средствами. Поэтому уже сегодня ведутся авангардные проекты по созданию альтернативных асинхронных процессоров. Ряд академических лабораторий и коммерческих компаний исследуют асинхронные решения, предлагаемые для замены решений синхронных; уже производятся или готовятся к производству реальные образцы подобных процессоров.

Координацию деятельности в области разработки асинхронных процессоров осуществляет симпозиум International Symposium on Advanced Research in Asynchronous Circuits and Systems, который с 1994 года проводит ежегодные встречи. Последняя из них, Async2002, состоялась в Манчестере.

## **Синхронность и булева логика**

Главная беда синхронности заключается в том, что она требует для своей реализации весьма сложный механизм управления, который может занимать от 10 до 50% общего числа транзисторов на микросхеме. Стоит задуматься: почти половина от всего наполнения чипа, оказывается, занято непродуктивной работой, играет роль координатора, не более того.

Это имеет целый ряд негативных последствий — и, прежде всего, сложность в двух ее ипостасях. Во-первых, сложность самого изделия как такового, и, во-вторых, что может быть даже важнее, постоянно растущая сложность и трудоемкость проектирования, вызванная необходимостью совместить два потока информации, обрабатываемый и управляющий. Когда количество транзисторов на кристалле исчисляется десятками миллионов, процесс проектирования становится в подлинном смысле запредельно сложным. Немногочисленные крупные производители, способные обеспечить проектирование на требуемом уровне, используют самые совершенные системы автоматизации проектирования, однако сложное вхождение в жизнь некоторых семейств 64-разрядных процессоров, постоянные переносы сроков выпуска свидетельствуют, что этого недостаточно. Дальше непредвиденных осложнений, видимо, будет еще больше. Закону Мура, в его упрощенной трактовке постулирующему удвоение производительности каждые 18 месяцев, угрожают не технологические барьеры, а сложность проектирования, в том числе и проблемы, связанные с тестированием. (О неминуемом повышении энергопотребления, о физическом пределе роста частоты, о роли электромагнитного излучения, особенно в мобильных устройствах, и других подобных вещах за всей их очевидностью, говорить не будем.)

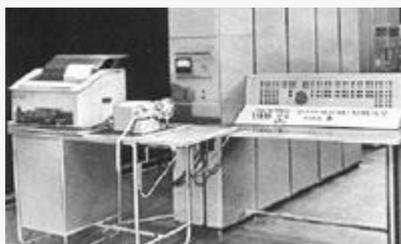
Рост сложности требует новых подходов к методологии проектирования. Уже очень давно было показано, что можно создать систему эффективного проектирования процессоров, аналогичную по смыслу системе программирования на языках высокого уровня. Если на некотором языке описать процессор, затем скомпилировать описание на уровень логических схем, то теоретически получится проект готового изделия. Подобную

методологию обычно называют синтезом. В СССР работы по синтезу были сконцентрированы в киевском Институте кибернетики, руководил ими академик Виктор Глушков. Были написаны статьи и книги, но практического выхода у исследований, посвященных синтезу, не оказалось; они разбивались, главным образом, о рифы синхронного управления.

Если кому-то удастся избавиться от механизма синхронизации, то после этого вполне можно будет возвратиться к идее синтеза, можно будет использовать серьезный математический аппарат, что откроет путь к созданию новых, куда более мощных процессоров. Когда статья была практически закончена, мне удалось найти в Internet ссылку, подтверждающую эту мысль. Оказывается, сотрудниками Манчестерского университета Дагом Эдвардсом и Эндрю Бэрдсли разработана система Balsa («an asynchronous hardware synthesis system»), предназначенная для описания и синтеза схем путем прямой компиляции.

Откуда же происходит сохраняющаяся приверженность синхронности, если она так плоха? Есть простенькое маркетинговое объяснение: чем тяжеловеснее проектирование или производство, тем выше уровень монополизации отрасли; понятно, кто оказывается в выигрыше. Но существует и более серьезное объяснение генезиса проблемы. Синхронность в современных процессорах есть прямое следствие использования исключительно двоичной булевой логики. Изобретенная Джорджем Булем в середине XIX века, эта логика оставалась в неизвестности почти век, до тех пор, пока юный в ту пору студент Клод Шеннон не открыл ее возможностей для конструирования электронных схем, по своей физической природе в основном двоичных. (К слову, несколько раньше него инженер В. Валтат тоже предложил использовать двоичную систему счисления, но только для механических машин.) С тех пор к двоичной системе сложилось культовое отношение (как у индусов к священным коровам) дошли до того, что первоклашки изучают двоичную систему счисления. Если позволите, «врожденный порок», которым страдает двоичная логика, состоит в том, что она не включает в себя условие завершения выполнения функции; она может существовать только на фоне соответствующей временной диаграммы, развертывающей процесс выполнения во времени. Иными словами, булева логика не представляет собой символически законченную логическую систему (symbolically completed), поскольку ей для завершения недостаточно собственного потока логических переменных. В данном случае «completed» можно перевести и как «завершенная» (еще точнее будет довольно странное словосочетание «способная к завершению», но может быть, лучше было бы назвать ее «самодостаточной»).

Булеву логику можно сделать законченной с привлечением внешнего управления, на этом и построены все современные процессоры. Для практической реализации к основной логике требуется добавить «небольшой довесок» в виде управления во времени, осуществляющего синхронизацию. Понятно, что информационные входные сигналы поступают не одновременно, задержка может быть разной, следовательно, на какой-то момент возникает ситуация неопределенности, которую можно разрешить, если, например, регламентировать срабатывание логической схемы по управляющему синхросигналу. Синхронизация реализуется посредством дополнительных сигналов (управляющих импульсов, или «стробов»), поступающих от тактового генератора или системных часов. Иначе говоря, у двоичной схемы «и/или», имеющей два информационных входа, на самом деле обязан быть третий управляющий вход, который синхронизирует два остальных. Выходит, двоичная схема не является вполне двоичной, она не может существовать без управления за пределами простейших релейных схем.



ЭВМ «Сетунь» с троичной системой представления чисел была разработана в проблемной лаборатории вычислительного центра Московского государственного университета. Серийное производство было поручено Казанскому заводу математических машин. Всего было выпущено 50 машин, 30 из них работали в высших учебных заведениях СССР

Этого врожденного недостатка лишена троичная логика. В свое время были выполнены достаточно интересные альтернативные работы с использованием троичной логики, например, в МГУ под руководством Николая Брусенцова были созданы оригинальные троичные компьютеры «Сетунь» и «Сетунь 70». Но обратим внимание на то, что эти компьютеры строились на специфической троичной элементной базе, где имеют место трехзначные сигналы и элементы памяти с тремя устойчивыми состояниями («трит»); байту здесь соответствует «тайт» — шестерка тритов. Троичная логика использовалась и для моделирования динамических процессов, происходящих в компьютере; такими работами занимался мой коллега по НИИ-5 Аркадий Бураковский, к сожалению, по молодости ни я, ни мои коллеги не могли понять глубины его идей. Теперь, спустя годы, об этом стоит пожалеть.

Как бы не были интересны машины на троичных элементах, построенные в 60-х — 70-х годах, они остались без будущего. Бурный прогресс кремниевой индустрии не оставил места троичной элементной базе, которая строилась главным образом на феррит-транзисторных элементах, поэтому серьезного развития троичные машины не получили. Вместе с оставшейся в истории ферритной технологией оказалась почти забытой и задача создания асинхронного процессора. На долгое время ее отставили в сторону, поскольку она оказалась бессмысленной с прагматических позиций. Так с водой выплеснули и ребенка.

В итоге те, кто продолжал думать об асинхронных процессорах, были поставлены перед дилеммой. Для того чтобы строить асинхронные процессоры, требуется выйти за ограничения, накладываемые простой булевой логикой. Но при этом нельзя забывать, что инвестиции в современные двоичные по своей природе кремниевые технологии таковы, что даже предположить появление чего-то принципиально иного, каких-то новых троичных элементов немислимо. На уровне технологий производства все должно остаться в существующем русле. Следовательно, нужно каким-то образом совместить двоичную природу кремниевых вентилях с логикой более сложной, нежели булева. На это и оказались направлены усилия нескольких инженерных коллективов.

Исследования в этом направлении ведут несколько групп; их работы постепенно выходят из подполья и начинают привлекать к себе серьезное внимание. Показательным в этом отношении оказалось подведение итогов 2001 года журналом *Microprocessor Report*. В январском номере этого авторитетнейшего журнала в области микропроцессорной техники названы победители по десяти номинациям. В самой престижной номинации, «За выдающиеся достижения в области цифровой обработки» были отмечены четыре компании-призера: Intel за «Hyper-Threading Technology», Proceler за «Dynamically VArIable Instruction seT Architecture», Sun Microsystems за «Asynchronous Design Technology» и Theseus Logic за «NULL Convention Logic». Так вот, из четырех работ две последние непосредственно связаны с созданием асинхронных процессоров. Помимо Sun Microsystems и Theseus Logic активные работы по созданию асинхронных процессоров возобновил с 1990 года Манчестерский университет силами AMULET Group; есть еще несколько исследовательских групп в университетах Соединенных Штатов и Западной Европы.

## NULL Convention Logic

С математической точки зрения, пожалуй, наиболее изящной является разработка альтернативной логики NULL Convention Logic (NCL), выполненная специалистами из небольшой компании Theseus Logic.

Стоит сказать несколько слов о компании, в чье название символично включено имя античного героя Тезея, который, применив нестандартную логику смог победить Минотавра в его дворце Лабиринте. Theseus Logic образована ветеранами, имеющими за плечами десятки лет работы в крупнейших корпорациях, в том числе, в блиставшей когда-то Honeywell. В качестве главного идеолога и теоретика NCL выступает Кард Фант, основатель Theseus. Это его вторая компания, в названии которой встречается имя Тезея. Решение, предлагаемое Theseus, состоит из двух частей: собственно математического аппарата и его реализации. Логику, предложенную Фантом, можно назвать «расширенной двоичной логикой», поскольку помимо двух основных сигналов TRUE и FALSE она оперирует еще двумя, NULL и INTERMEDIATE. Логика NCL существует в двух вариантах, как троичная, где используются TRUE и FALSE + NULL, и как четверичная, где используются все четыре значения.

Избавление от обязательных часов и необходимости в синхронизации в NULL Convention Logic достигается путем интеграции данных и управления в одном логическом выражении, в одном логическом потоке, где NULL, ее называют переменной-разделителем, выполняет синхронизирующую функцию. Информативные сигналы, нули и единицы, и NULL передаются по разным проводам, поэтому иногда такой подход образно называют «dual rail», подразумевая под этим, что для передачи сигнала используется «два рельса» (точнее, два провода). Важно заметить разницу в смысловом значении сигнала: проводник сам по себе не несет в себе логического значения, изменение напряжения на нем свидетельствует только о том, что сигнал поступил. Сигнал приобретает логический смысл лишь тогда, когда по параллельному проводнику поступает NULL; такое разделение обеспечивает самосинхронизацию. Логику NCL можно представить как второй логический уровень над булевой логикой, который позволяет воспроизвести троичные элементы с использованием принятых сегодня технологий.

В качестве элементной базы, на которой реализуется NCL, применяются пороговые вентили с гистерезисом. В отличие от «простого» полупроводникового вентиля этот имеет два состояния — DATA или NULL. Исходное состояние — всегда NULL, вентиль сохраняет NULL. Переход в состояние DATA возможен, когда на всех входах будет значение DATA, каждое подтвержденное своим разделителем NULL. На рисунке представлены упрощенные реализации логических функций NCL средствами обычной двоичной логики. Использование четвертого сигнала INTERMEDIATE и «четырёхрельсовой» логики позволяет сделать конструкцию вентиля и схемы в целом более эффективной. Очевидно, расплачиваться за избавление от синхронизации приходится увеличенным числом проводников и транзисторов.

В качестве одной из наиболее реальных областей применения асинхронных микросхем на основе NCL называют «системы на кристалле» (system-on-chip — SoC). В 2000 году Theseus выпустила 8-разрядный микроконтроллер NCL08, изготавливаемый по 0,25-микронной технологии и насчитывающий 1 млн. транзисторов. С 1999 года существует стратегическое партнерство с корпорацией Motorola, которая перепроектировала свой микропроцессор STAR08 на основании предложений от Theseus. Но, может быть, самое убедительное доказательство перспективности NCL можно найти в подписании пятилетнего контракта между Theseus и Sanders, дочерней компании корпорации Lockheed Martin. В соответствии с ним Sanders и Theseus начинают совместно работать по проекту, спонсируемому тем самым агентством перспективным оборонных проектов DARPA, которое в свое время финансировало создание Internet, и самой Lockheed. Цель этого проекта заключается в создании процессоров для перспективного тактического истребителя F-22 Raptor.

## Работы Asynchronous Design Group в Sun Microsystems

Вторая из отмеченных журналом *Microprocessor Report* работ по асинхронным процессорам ведется в исследовательской лаборатории Sun Microsystems под руководством Айвэна Сазерленда. За свои работы по приданию процессорам асинхронности он еще в 1989 году получил премию имени Тьюринга, одну из высших наград в области computer science. Сазерленду принадлежит идея микроконвейера, функционирование которого организована на принципах, названных «рукопожатием» (handshaking). Рукопожатие в данном случае — ни что иное, как реализация обратной связи путем обмена служебными сигналами. Этот принцип получил большое распространение, он существует в разных версиях, активное и пассивное, двух- и четырехфазное.

Однако сам Сазерленд на этом не успокоился и сейчас развивает идею асинхронности, но уже на уровне полупроводниковых переходов. Свои новые идеи он реализует в проекте под названием FleetZero. Как утверждает, экспериментальный процессор демонстрирует заметно более высокую производительность, чем синхронные схемы, а эффективность использования полупроводниковых ресурсов повышается примерно в двадцать раз. Впрочем, пока проект FleetZero находится на начальной фазе и появления коммерческих результатов можно ожидать не ранее 2005 года. В рамках этого проекта реализуется технология управления потоком команд GasP, которая должна хорошо согласовываться с логикой работы компиляторов.

## MiniMYPs и AMULET

Несколько компаний и исследовательских групп ставят своей целью переделать в асинхронные стандартные синхронные процессоры. Наиболее интересной представляется работа, которая проводилась в Калифорнийском технологическом институте в Пасадене. Здесь исследователям, взяв в качестве прообраза процессор MIPS R3000, удалось сделать его асинхронный аналог MiniMIPS. Сегодня на основе этих исследований действует начинающая компания Fulcrum Microelectronics. Аналогичное направление работ избрала для себя английская группа AMULET Group, выпускающая с 1994 года процессоры AMULET. Эти процессоры родственны синхронному процессору StrongARM. Кроме того, в этом исследовательском центре ведутся работы по использованию нейронных сетей в асинхронных процессорах.

## Взгляд скептика

Среди видных специалистов в области процессоров нет единодушия во взглядах на асинхронные процессоры. Гордон Белл, легендарный создатель архитектур PDP и VAX, относится к числу скептически настроенных. Он признает, что, быть может, в какой-то очень далекой перспективе у асинхронных процессоров действительно есть будущее, однако все современные попытки их построить он считает неудачными. Он приводит следующую аналогию: проектировщики первых стальных мостов, не желая того, повторяли то, как строили мосты из камня. Потребовалось длительное время, чтобы понять возможности нового материала и создавать совершенно новые проекты. Нынешние попытки создания асинхронных процессоров, по мнению Белла, слишком привязаны к тому, что было сделано за последние десятилетия, им просто не хватает радикализма.

*Леонид Черняк ([cherniak@osp.ru](mailto:cherniak@osp.ru)).*

---

## Уроки механического пианино

К проблеме синхронности можно подойти с еще одной неожиданной позиции. Вспомним начала кибернетики. Одно из важнейших достижений Норберта Винера заключается в

том, что он показал значение обратной связи в системах с саморегулированием в живом мире и в технике. Существует альтернатива: либо управление по обратной связи, либо программное управление, третьего не дано. Обратная связь дает элементу системы возможность оценить сделанное (например, свое положение в некотором пространстве параметров), критически его оценить посредством регулятора с отрицательной обратной связью и внести коррекцию во входные параметры, чтобы приблизиться к требуемому положению. Типичный пример программного управления — стрельба из артиллерийских орудий, пример управления с обратной связью — боеголовка с самонаведением. Первыми программными автоматами были механические игрушки, шарманки и тому подобное; они не меняют своего положения, поэтому программного управления оказывается достаточно. Процессор, работающий по заданной временной диаграмме, — точно такая же синхронная машина с конечным числом состояний, что и механическое пианино. Какими бы сложными не были архитектуры современных процессоров: суперскалярными, конвейерными и прочее, прочее — все они попадают в эту простейшую модель. Если вернуться к аналогии с армейским строем, то в асинхронном процессоре ответственность за совместность действий переносится на его участников, они должны поглядывать на остальных и регулировать свое движение по отношению к ним. Внедрение асинхронного управления в процессор означает частичный или полный отказ от программного управления и переход к управлению по обратной связи. На рисунке представлены два подхода: синхронный и асинхронный. Наполнение блока «Комбинационная Логика» может различаться в зависимости от избранного решения. При всем разнообразии решений с точки зрения управления, все они различаются между собой глубиной проникновения обратной связи — иначе говоря, тем, какой объем логики предполагается вместить в прямоугольник, обозначенный как «Комбинационная Логика». Обратную связь можно поместить на самый нижний уровень, реализуемый непосредственно на кристалле С-MOS, если использовать так называемые элементы С или элементы Мюллера и синтезировать процессор с использованием аппарата сетей Петри. Более высокий, средний уровень предполагает включение дополнительных сигналов в поток данных, что позволяет реализовать обратную связь на уровне элементарных логических функций. К среднему уровню также относят и технологию, которая называется микроконвейерной. Наконец, третий, верхний уровень предполагает охватывание петель обратной связи более крупных логических компонентов, включающих в себя регистры целиком.

