

УДК 681.324-192:681.325.65

САМОСИНХРОНИЗАЦИЯ – ЕСТЕСТВЕННАЯ ОСНОВА АРХИТЕКТУРЫ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

И.А. Мизин, А.В. Филин

1. ВВЕДЕНИЕ

В течение нескольких лет в ИПИ РАН выполняется фундаментально-поисковый проект, цель которого – разработка архитектуры и системной организации перспективного естественно-надежного компьютера (ЕНК) для систем реального времени (СРВ) [1, 2]. Изначально ставилась задача разработать сбалансированную систему архитектурных принципов, позволяющую конструировать архитектуры ЕНК для систем управления процессами особо ответственных (критических) объектов. В качестве отличительных признаков архитектуры ЕНК (и, соответственно, компьютеров на ее основе) были выбраны свойства: *управляемая производительность, гарантоспособность, самосинхронность, рекуррентность, параллелизм и развиваемость*. Представлялось, что реализация этих свойств в архитектуре ЕНК способна будет обеспечить им эффективную (качественную, безаварийную и долговременную) работу в качестве ядра ответственных систем управления.

К настоящему моменту практически завершен первый этап разработки проекта. Приоритетными направлениями поиска на этом этапе были:

а) определение требований к архитектуре, удовлетворяющей перечисленным выше качествам;

б) нахождение организации архитектуры, которая отвечала бы желаемым свойствам, опиралась на достоинства выбранной схемотехники и гарантировала качественные и максимально быстрые

(за счет параллелизма) вычисления в реальном времени;

в) выбор и разработка схмотехники, удовлетворяющей требованиям построения и реализации самосинхронных WSI (Wafer Scale Integration)-систем (систем с интеграцией на пластине) с указанными выше свойствами;

г) определение состава и функций инструментальных средств поддержки процессов проектирования интегральных схем для ЕНК.

В результате выполнения первого этапа:

- выбрана строго-самосинхронная (ССС) интегральная схмотехника, обладающая уникальными свойствами, особенности построения, проектирования и использования которой опубликованы в работах [3 - 7];

- предложены принципы организации рекуррентной компьютерной архитектуры, ориентированной на обеспечение быстрых (параллельных) и достоверных вычислений в реальном времени в областях использования ЕНК;

- завершается разработка САПР строго-самосинхронных интегральных схем различного уровня сложности для ЕНК (от схем базовых ячеек до уровня WSI-схем) в базисе КМОП-технологии (технологии на основе комплементарных МОП-структур); информация о САПР СССР-схем RONIS приведена в [8].

Ниже перечисляются особенности предложенной качественно новой архитектуры для ЕНК с единственной целью - дать завершенное представление о значимости принципа самосинхронизации не только для организации аппаратных средств (на уровне элементной базы), но и для выбора архитектуры компьютеров в целом. Универсальность принципа самосинхронизации дает возможность использовать его на всех уровнях организации компьютерной системы, гармонично объединить все ее части в единое целое (систему). Именно поэтому ему

уделяется особое внимание.

2. ВОЗМОЖНЫЕ ОБЛАСТИ ПРИМЕНЕНИЯ ЕНК

При разработке архитектуры одним из главных направлений исследований было обеспечение быстрых высоконадежных вычислений за счет ориентации их на параллелизм. Дело в том, что из множества требований, предъявляемых к компьютерам для СРВ, чаще всего изменяются (постоянно растут) требования к их вычислительной мощности при сохранении стабильно высоких требований к надежности. Скорость изменения этих требований фактически определяет скорость совершенствования не только самих компьютеров (а она очень высока, так как стимулируется постоянным и быстрым прогрессом в интегральных технологиях), но и систем на их основе. Повышение производительности - одна из главных тенденций в развитии компьютерных систем (КС) практически во всех областях их приложений.

Известно, что области применения устанавливают большинство основных требований к характеристикам компьютеров и других частей систем управления. В данном случае при определении требований рассматривались как возможные следующие потенциальные области использования ЕНК:

- обеспечение оптимального и безопасного функционирования подвижных технических объектов (автомобилей, судов, самолетов, космических объектов, робототехнических комплексов, лифтов в высотных зданиях, клетей в шахтах и др.);

- безаварийное управление локальными специфическими технологическими процессами длительного действия (энергетические ядерные установки, размещаемые на вездеходах, космических объектах, судах и т.п.);

- мониторинг состояния объектов, опасных в экологическом отношении (газовые и нефтяные насосные станции, буровые установки, химические, медицинские и прочие реакторы, очистные сооружения и т.п.);
- длительные натурные исследования, испытания и эксперименты;
- управление критическими технологическими процессами и агрегатами;
- управление системами оружия, войсками, связью.

Объекты управления перечисленных областей имеют много общего. Все они требуют использования высоконадежных средств контроля и управления, поскольку работают в реальном времени в *on-line*-режиме, не допускающем даже кратковременных перерывов в управлении, в жестких механо-климатических условиях эксплуатации, вызывающих более быстрое старение материалов. Ко многим объектам не всегда возможен доступ по необходимости. Поэтому к компьютерам реального времени, предназначенным для использования в таких системах, предъявляются повышенные и специфические требования. В частности, такие компьютеры должны гарантировать достаточный уровень вычислительной мощности, иметь увеличенный срок службы и другие свойства на всем цикле выполнения задания в реальном времени как в нормальных, так и в ожидаемых экстремальных условиях эксплуатации.

3. ОЖИДАЕМЫЕ СВОЙСТВА ЕНК ДЛЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Было установлено, что для работы в перечисленных областях ЕНК должны обладать следующими свойствами (перечисляются в порядке

понижения степени их важности - приоритетов):

- *предельно высоким быстродействием* (с целью создания запаса производительности на случай совпадения аномальных состояний на объекте управления с подобными состояниями компьютера);

- *безопасностью* - функциональной и параметрической, внутренней и внешней ("не навреди себе и другим");

- *живучестью* (сохранением постоянства существования) на всем периоде выполнения задания;

- *увеличенным сроком службы* (ресурсом) *электронной части систем* (пролонгированным до срока службы основного технологического оборудования);

- *"окрашенностью в зеленые цвета"* (соответствием требованиям экологического стандарта Energy Star);

- *способностью к развитию* (перекрытию диапазона производительности не менее чем на 3-4 порядка по отношению к минимальной конфигурации системы);

- *разумно быстрой экономической окупаемостью*.

Производительность компьютеров для критических СРВ постулируется в проекте как самое важное свойство, имеющее максимальный приоритет.

Надежность (безотказность, ремонтпригодность, долговечность и сохраняемость), безопасность и живучесть рассматриваются, прежде всего, как свойства, способствующие повышению интегральной (на всем периоде выполнения задания) производительности (совместно, например, с параллелизмом), и поэтому им также назначены высокие (сразу после производительности) приоритеты.

Реализация перечисленных важнейших свойств позволяет наделить ЕНК *гарантоспособностью* (по Авижиенису и Лапри [9]) - обобщающим свойством, характеризующим возможность их

применения в системах управления ответственными объектами и процессами.

Современные управляющие компьютеры (для работы в *on-line*-режиме), даже относящиеся к классу отказоустойчивых, обладают лишь некоторыми из перечисленных свойств. Очевидна необходимость в дальнейшем совершенствовании компьютеров и компьютерных систем для критических приложений.

4. ИССЛЕДОВАНИЯ В ОБЛАСТИ СОЗДАНИЯ ПЕРСПЕКТИВНЫХ КС РЕАЛЬНОГО ВРЕМЕНИ

Работы по созданию высокопродуктивных и высоконадежных КС ведутся во всем мире. В этом отношении показателен проект аэрокосмической ассоциации США (Aerospace Industries Association Systems) "Ультранадежные электронные системы" (URES - Ultrareliable Electronic Systems, 1990-2000). Основная задача проекта - создание к началу XXI века электронных КС реального времени, способных безотказно функционировать в течение полного срока их службы [10]. Проекты URES и ЕНК совпадают в своей основной цели, но расходятся в методах ее достижения и особенно в объемах финансирования (ЕНК-проект инициативный; в данное время он практически не финансируется).

Имеющиеся в большом числе на рынке отказоустойчивые КС подтвердили на практике обещанную высокую надежность, но за чрезмерно высокую стоимость. Практически все отказоустойчивые системы, предназначенные для управления в реальном времени, используют один и тот же метод повышения надежности - резервирование (удвоение, мажорирование, дублирование парами и т.п.). При этом возникают принципиальные трудности, в частности - контроль правильности работы схем контроля, резервирование

устройств, управляющих резервированием и др. Сами системы и резервируемые устройства строятся с использованием традиционной (синхронной) схемотехники, хотя известно, что около 80 % отказов аппаратуры происходит из-за рассогласования сигналов (ошибок синхронизации, состязания фронтов сигналов и операций их арбитража).

На первом этапе реализации ЕНК-проекта были изучены особенности применения КС в критических областях, а также сильные и слабые стороны известных проектов в области параллельных КС - Data Flow Machines (DFM), Reduction Machines (RM), Control Flow Machines (CFM: CISC, RISC, VLIW), Vector Processors (VP) и др. В результате удалось сформулировать ключевые особенности и принципы построения ЕНК (в том числе и требования к их элементной базе [3]), реализация которых даст возможность повысить устойчивость вычислительного процесса в реальном времени.

5. КЛЮЧЕВЫЕ ПРИНЦИПЫ И ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ ЕНК-АРХИТЕКТУРЫ

5.1 Принципиальная основа архитектуры

Предлагаемая архитектура не имеет аналогов, поскольку опирается на новую парадигму вычислительного процесса. Она ориентирована на решение задач, алгоритмы которых могут быть описаны динамическими графами (как обычных задач, так и задач, где во времени меняется алгоритм или сама структура их решения, либо и то, и другое). Примерами такого рода задач являются: административная структура предприятия, организация систем связи, снабжения, адаптивная обработка сигналов в реальном времени, распознавание и ликвидация аварийных состояний, построение номенклатуры товаров или изделий определенной категории и проч.

За основу взят *динамический подход*, отличающийся от всех известных тем, что здесь не строится статический граф алгоритма. Его в явном виде не существует. При динамическом подходе к решению задачи граф ее алгоритма представляет собой как бы "мгновенный снимок" (одномоментное событие) развития процесса его реализации в каждый конкретный (t -тый) момент времени, а развитие событий во времени - динамика - связано не с движением по графу, а с изменением самого графа. Множество одномоментных графов составляет граф в целом (образует графовую траекторию). Совокупность методов описания и изучения таких систем называют графодинамикой; суть ее заключается в следующем.

Пусть $x(t)$ - граф, существующий в момент t , и закон изменения графа во времени может быть записан в форме некоторого рекуррентного процесса:

$$x(t+1) = F[x(t)], \quad (1)$$

где F - некоторый оператор, преобразующий граф, наблюдаемый в момент времени t , в граф, который появляется в момент времени $t+1$.

Если назвать $x(0)$ начальным графом, то последовательность графов $x(t)$, возникающая из $x(0)$ в соответствии с (1), и будет графовой траекторией. Фактически выражение (1) олицетворяет собой задачу ("программу"), алгоритм которой заблаговременно был представлен в виде последовательности одномоментных графов, а затем рекуррентно свернут в "точку" $x(0)$, код которой становится условием его саморазвертки, когда это потребуется. В рекуррентно свернутой "программе", представляющей собой рекуррентно свернутый алгоритм обработки набора (потока) операндов, описывается не поток инструкций, а поток правил порождения шагов обработки операндов. Исполнение начального условия порождает последовательность одномоментных сменяющихся графов $x(1), x(2), \dots, x(t)$, образуя

динамический вычислительный процесс. Очевидно, что при рекуррентной свертке достигается мощное сжатие алгоритма и, соответственно, существенная экономия объема памяти - самого ненадежного компонента компьютеров.

Графодинамика вкуне с рекуррентной теорией самоорганизации и развития и парадигмой управления вычислительным процессом от потока активных данных (обладающего внутренней самосинхронностью вследствие синхронизации процесса от готовности данных по мере их рекуррентного саморазвертывания) составляет теоретическую основу архитектуры, отображающей организацию и развитие вычислительного процесса.

Не вдаваясь в подробности архитектуры, можно назвать ключевые принципы, являющиеся ее базисом, отличающие ее от всех известных в настоящее время компьютерных архитектур (типы их перечислены выше) и призванные обеспечить компьютерам на ее основе новые свойства и высокие технические характеристики:

1) *рекуррентность* (рекуррентное кодирование алгоритмов, рекуррентный внутренний машинный язык, рекуррентная саморазвертка инструкций и вычислительных процессов);

2) *самообеспеченность процессов, данных и единственного вычислительного потока* (отсутствуют привычные для CFM- и DFM-архитектур взаимосвязанные потоки команд и данных);

3) *повышение статуса данных до роли активных единиц* (операнды сопровождаются рекуррентными "инструкциями", способными к саморазвертке);

4) *самосборка структур процессов и данных;*

5) *естественная событийная (самосинхронная) природа взаимодействий;*

б) *естественная иерархия архитектуры;*

7) *естественно-масштабируемый параллелизм.*

Перечисленные принципы составляют концептуальную базу архитектуры и связаны между собой как взаимодополняющие. Они названы *ключевыми*, потому что распространяют свое влияние на все уровни организации архитектуры компьютера, обеспечивая ему важнейший принцип - *целостности*. Перечисленные семь качеств в единстве порождают главное качество - *архитектуру как целое* (и, соответственно, сбалансированный по характеристикам компьютер). Они же и обеспечивают полную новизну (*ноу-хау*) принимаемых решений, обеспечивая архитектуре целый ряд отличительных особенностей.

Первый принцип - стержневой для ЕНК-архитектуры, поэтому она и называется *рекуррентной*. Принцип рекуррентности тесно связан с остальными принципами, в том числе и с принципом самосинхронизации, в определении степени полезности которого и состоит цель данной работы.

5.2. Отличительные особенности архитектуры

Строящаяся на сформулированном выше принципиальном фундаменте архитектура отличается от известных сегодня архитектур, прежде всего, *нетрадиционностью*, обеспечившей свободу в выборе решений. Такая свобода снимает ограничения на достижение главной цели проекта - *улучшение производительности и других характеристик компьютеров за счет более высокого уровня организации их архитектуры*, который, в свою очередь, может быть получен, лишь опираясь на систему сбалансированных системных принципов. Один из них - принцип *самообеспеченности*, состоящий в интеграции в архитектуру (на все ее уровни) ряда системных функций с целью их самоисполнения, как-то: самосинхронного (событийного)

взаимодействия, саморазвития процессов, саморазвертки функциональных полей (рекуррентных "инструкций"), самосборки процессов, самосборки наборов данных и т.д. Принцип "*само...*" распространяется и на физический уровень архитектуры - на самосинхронную схемотехнику интегральных схем, обладающую *самопроверяемостью*, *самодиагностируемостью*, *самоорганизующимися циклами*. Таким образом, нетрадиционность архитектуры составляют следующие элементы новизны:

- *парадигма вычислений*, в основе которой лежит графодинамическое представление алгоритмов, их предварительная рекуррентная свертка и последующая саморазвертка в процессе исполнения;

- *тип организации информационного процесса*, при котором интеграция данных с рекуррентными "инструкциями" сводит традиционную двухпоточность к однопоточности; статус данных повышается, превращая их из пассивных единиц в активные, обладающие средствами системной поддержки процесса внутри себя, а не извне (в дальнейшем - "интегрированные данные");

- *концепция самоуправления ходом вычислительного процесса от потока активных интегрированных данных*: трасса вычислительного процесса динамически разворачивается как следствие предварительной рекуррентной свертки алгоритма;

- *концепция программирования на основе капсул*: капсула представляет собой самодостаточный объект, поддерживающий технологию объектно-ориентированного программирования; поскольку интегрированные данные несут в себе и "программу" их обработки, их совместное перемещение (обеспечивается перемещением капсулы) и загрузка в активную зону исключает проблему связывания данных и программ;

- *интерпретация понятия "программа"* - замена ее на единственный код начальных условий развертки (НУР), записываемый в машинное слово рядом с операндом, образуя "интегрированное данное";

- *внутренний машинный язык (ВМЯ)*, состоящий из иерархически вложенных внутренних языков, соответствующих различным уровням организации архитектуры; ВМЯ - это язык "начальных условий развертки", получаемых компилятором при трансляции и сворачивании графа взаимодействия операндов в код НУР, то есть язык самого нижнего уровня;

- *концепция представления типов машинных данных*, основанная на рекуррентной (вложенной) иерархии капсул интегрированных данных;

- *парадигма организации и использования памяти*, отличающаяся от фон-неймановской.

Поскольку здесь нет возможности раскрыть значимость каждого из ключевых принципов со всеми его взаимосвязями, остановимся на принципе, имеющем отношение к параллелизму вычислений и параллельным компьютерам, для которых обосновывается выбор принципа самосинхронизации.

Принцип естественного параллелизма (совместно с рекуррентным принципом) дает полную независимость пространственного параллелизма от архитектуры, ее индифферентность к этой проблеме (что нетривиально). Параллелизм архитектуры здесь понимается в традиционном смысле, по М.Ж.Флинну: параллелизмы SIMD- и MIMD-типов, пространственный (многопроцессорный) и временной (конвейерный или MISD) параллелизмы.

Эта универсальность есть следствие объединения потоков инструкций и данных в единый поток саморазворачивающихся

интегрированных данных, после чего кратность инструкций теряет смысл и перестает быть как классификатором, так и ограничителем параллелизма. Отсюда - легкость его масштабирования.

Хотя архитектура заявляется как параллельная и масштабируемая, ее параллелизм не является, в отличие от иерархии, обязательным отличающим компонентом. Параллелизм и масштабируемость архитектуры здесь следует понимать как потенциальную возможность, естественно адаптируемую архитектурой в различных видах. Дело в том, что каждый шаг обработки может осуществляться над множеством элементов саморазворачивающихся интегрированных данных, составляющих текущий временной срез процесса: одновременно множество операндов может селектироваться в пары аргументов, и над ними могут выполняться разные операции.

Следовательно, исходные принципы организации процессов порождают динамический параллелизм, вытекающий из структуры процесса (алгоритма) и принципиально неограниченный. (На практике ограничение определяется степенью параллелизма аппаратных средств, на которых реализуется процесс.)

В данной архитектуре степень параллелизма не специфицируется и не ограничивается архитектурой. Предлагаемая архитектура изначально ориентирована на "среднезернистый" параллелизм на уровне блоков (наборов) данных. Хотя теоретически это хуже, на деле может оказаться, что это гораздо практичнее; такой вывод напрашивается из рассмотрения принципов достижения "мелкозернистого" параллелизма в Data Flow-машинах, который к успеху так и не привел.

В рекуррентной архитектуре все перечисленные принципы играют важную роль. Но среди них особое значение имеют принципы *рекуррентности процессов* и *событийного* (самосинхронного)

взаимодействия. Кратко расшифруем их смысл и значимость для архитектуры.

Названные принципы не свойственны архитектурам традиционных компьютеров (как принстонского, так и гарвардского типов), что позволяет определить эту архитектуру как нетрадиционную, существенно отличающуюся от всех известных.

Поскольку рекуррентность, самосинхронность (самоорганизация циклов), параллелизм и другие особенности свойственны живым организмам (биологическим системам), обладающим уникальным природным свойством - *естественным интеллектом*, данную архитектуру можно рассматривать как шаг (пусть небольшой) в направлении понимания принципов создания самоорганизующихся КС.

6. ПРЕИМУЩЕСТВА РЕКУРРЕНТНО-ОРГАНИЗОВАННОЙ АРХИТЕКТУРЫ

Рекуррентная организация архитектуры, предполагающая (в начальный момент) определенную функционально-прикладную ориентацию компьютера (реализация скоростных алгоритмов обработки радиосигналов в реальном времени в соответствии со стандартами CDMA, GSM и т.п.), поддерживает весьма эффективный статико-динамический параллелизм. При этом обеспечивается эффективная работа как со скалярными, так и с векторными величинами, повышается производительность КС за счет исключения этапов выборки и декодирования инструкций (поскольку их нет), снижаются накладные расходы на сам вычислительный процесс и на поддержание параллелизма, уменьшается вероятность конфликтных ситуаций. Облегчается создание средств поддержки языков высокого уровня (ЯВУ) и применение современных технологий программирования (в частности, объектно-ориентированной). Перечисленные особенности

были определены пока в процессе качественной оценки рекуррентной архитектуры.

Можно ожидать, что в результате перехода на рекуррентную архитектуру появятся новые возможности, которые позволят:

- создавать высокопараллельные событийно-ориентированные (самосинхронные) системы, поддерживающие массовый статико-динамический параллелизм при меньших ресурсах;
- уменьшить объем памяти СБИС-систем за счет компрессионной (рекуррентной) формы представления алгоритмов (программ);
- сократить потери времени на пересылку, загрузку и хранение "программ" за счет их компрессии;
- повысить защищенность информации и алгоритмов программ за счет использования рекуррентного принципа кодирования;
- уйти от проблемы связности программ и данных, замыкая их внутри капсулы.

Однако возникают и новые проблемы. В частности, необходимо создать компилятор, способный выполнять рекуррентную свертку любых типов алгоритмов. Это задача высокого уровня сложности. Возникает нужда в языке описания преобразований графов и т.д. Естественно ожидать и других проблем, которые начнут возникать в процессе детальной проработки принципиально новой архитектуры по мере спуска от идеи к более низким уровням ее организации. Предварительная проработка показала: предлагаемая идея может быть охарактеризована как *de novo* и стоит того, чтобы ее реализовать, особенно на фоне имеющего место кризиса идей в области архитектур параллельных компьютеров.

7. ПОБУДИТЕЛЬНЫЕ ФАКТОРЫ ПЕРЕХОДА К САМОСИНХРОНИЗАЦИИ

В организации рекуррентной архитектуры принцип самосинхронизации занимает важное место, поскольку предоставляет разработчику естественный способ распределенного управления параллельными процессами, как статическими, так и динамическими. Более того, он является *универсальным* (релаксационно-активным) принципом взаимодействия, повсеместно встречающимся в природных самоорганизующихся объектах. Эта универсальность дает возможность применить его на всех уровнях системы, от архитектурного до схемотехнического, и получить дополнительный положительный эффект за счет выполнения всего проекта в едином ключе (см. [3]).

Переход к самосинхронному принципу, естественно, с отказом от синхронного и асинхронного - ответственный шаг. Сейчас разработчиками синхронных СБИС осознана необходимость перехода лишь на асинхронную схемотехнику, но это полумера. Причины выбора самосинхронного принципа, вытекающие из проблем проектирования и изготовления синхронных СБИС, описаны в [3, 4]. На уровне выбора концепции, принимаемой за основу проекта параллельного ЕНК, три основных фактора определили необходимость привлечения принципа самосинхронизации:

- создание более высокоорганизованной (и, следовательно, более эффективной) архитектуры, поддерживающей параллельную обработку данных и новую парадигму вычислений;
- естественное повышение уровня сложности аппаратуры и программ без снижения достоверности обработки информации;
- применение схемотехнической базы СБИС, понижающей сложность функционально-структурной организации верхних уровней архитектуры разгрузкой ее от функций самоконтроля, самодиагностики и пр.

Проанализируем связи принципа самосинхронизации с

основными свойствами ЕНК-архитектуры, показывающие его системность.

7.1. Рекуррентность и самосинхронизация

Предлагаемой рекуррентной архитектуре органически присуща внутренняя *самосинхронность* ("событийность") взаимодействий вследствие синхронизации вычислительного процесса от готовности данных (как в DFM) по мере их рекуррентного саморазвертывания. Это позволяет естественным образом использовать все преимущества самосинхронного принципа взаимодействия (см. [3]) и сделать естественными развитие вычислительного процесса и параллелизм системы, имея в виду:

- рекуррентную непрерывность развития процесса без внешних принуждающих воздействий;
- параллельность развития процесса как множества одновременных и независимых событий;
- событийный (самосинхронный) характер взаимодействий без ограничения длительности событий;
- автоматическую согласованность взаимодействий на всех уровнях без привязки к какой-либо принудительной тактовой системе.

7.2. Структурный параллелизм и самосинхронизация

Состояние благополучия фирм-производителей КС, базирующееся на устойчивом и быстром прогрессе в технологии кремниевых интегральных схем, постепенно сходит на нет по мере исчерпания возможностей "последовательных" принципов построения компьютеров фон-Неймана и приближения к физическому пределу микроминиатюризации транзистора в кремнии.

В результате центр тяжести усилий проектировщиков КС все больше смещается в сторону поиска новых архитектурных решений и

принципов их построения. Среди направлений поиска преобладает интерес к организации параллельной обработки данных на разных уровнях аппаратуры. Однако быстро выяснилось, что рост сложности параллельных систем и образующих их устройств потребовал, прежде всего, пересмотра требований к общей системе синхронизации. Несоответствие жесткой схемы синхронизации параллельной схеме работы аппаратных средств и программного обеспечения стало побудительным мотивом заметного увеличения числа разработок, исповедующих *асинхронные* принципы взаимодействия.

Наметившийся сейчас поворот в сторону асинхронных систем - шаг вперед в решении проблем организации эффективного взаимодействия множества элементов системы (на уровне функциональных устройств). Но поскольку не существует широкого опыта проектирования СБИС-асинхронных систем (нет отработанных методик их проектирования), представляется целесообразным сразу перейти к более перспективному принципу - самосинхронизации: ведь асинхронность - полумера, а самосинхронная работа - кардинальное решение.

Наиболее полно и последовательно принцип асинхронного взаимодействия реализуется в самосинхронных (асинхронных апериодических) элементах системы, в которых очередная фаза работы элемента инициируется не общими часами (синхронные устройства) и не границей некоторого априорно заданного интервала времени (классические асинхронные устройства), а самим устройством по завершении в нем переходных процессов, связанных с предыдущей фазой работы (см. также [3, 6]).

Изучение вопроса показало, что самосинхронизация во многом снимает проблемы проектирования WSI-систем и продвигает асинхронность на новый, более высокий уровень, поскольку является

событийным принципом организации управления параллельно протекающими асинхронными процессами.

Таким образом, переход от синхронных структур КС (с принудительной жесткой синхронизацией порядка и времен выполнения операций) к асинхронным аperiodическим структурам однозначно ведет к естественному (событийному) динамическому параллелизму, опирающемуся на концепцию самосинхронизации.

Модули таких параллельных структур должны функционировать, как минимум, асинхронно. Сами модули могут быть реализованы либо как синхронные (т.е. традиционным способом), либо как асинхронные, но наилучшие результаты в системе получаются, когда они также сконструированы в базисе самосинхронной (асинхронной аperiodической) схмотехники, обеспечивая единство реализации концепции истинного параллелизма. Можно ожидать, что систематическая самосинхронность, реализованная, прежде всего, на системном уровне архитектуры, приведет к лучшему соотношению производительности, цены и потребляемой мощности по сравнению с другими параллельными системами.

7.3. Производительность, устойчивость вычислений и самосинхронизация

Производительность компьютеров для класса СРВ является главным показателем их качества. Естественно, что и в этом классе основным методом повышения производительности (вычислительной мощности) стал параллелизм (метод обеспечения параллельных вычислений). Его высшая форма - *динамический параллелизм* (ДП), не требующий предварительного распараллеливания программы и обеспечивающий линейный рост производительности КС при линейном наращивании числа параллельно работающих операционных (процессорных) элементов. Однако на пути освоения ДП много проблем

как теоретического, так и прикладного характера, и до сих пор никому еще не удалось воплотить в компьютерах все его потенциальные преимущества.

С другой стороны, во многих приложениях, связанных, например, с обработкой сигналов, хороший результат дает *статический параллелизм* (СП) как SIMD-, так и MIMD-типа. Суть СП в том, что в исходном (как правило, последовательном) алгоритме параллелизм выявляется заранее (алгоритм преобразуется в параллельный) и программируется с учетом конкретной структуры аппаратных средств и возможностей компилятора параллельной КС. В ходе своего выполнения программа реализует предзапрограммированную (проложенную заранее) трассу вычислительного процесса. Практический же подход к реализации параллелизма в КС требует разумного сочетания обоих видов параллелизма - статического и динамического - для достижения оптимального конечного результата. Архитектура и аппаратные средства ее реализации должны эффективно воспринимать оба типа параллелизма и позволять находить разумные компромиссы. Самосинхронность и рекуррентность, присущие архитектуре и аппаратуре ЕНК, такую возможность предоставляют.

Однако реализация в КС смешанного параллелизма традиционными методами сопровождается усложнением их аппаратных и программных ресурсов, а значит - снижением устойчивости вычислительного процесса и достоверности обработки информации (из-за роста числа точек состязания сигналов, синхронизации и арбитража в аппаратуре). Игнорирование этого факта при проектировании КС как систем на одной пластине (WSI-технология) недопустимо.

Решение проблемы - в переходе к ССС-схемотехнике - наилучшей (на сегодняшний день) основе проектирования не только параллельных, но и высокодостоверных WSI-систем. Переход к ССС-схемотехнике -

это встраивание на все уровни системы информационной безопасности и гарантия повышения устойчивости вычислений. Естественная оперативная самопроверяемость и самодиагностируемость, гарантируемая информационная безопасность, параметрическая и энергетическая (по питанию) отказоустойчивость, плавная (без толчков по питанию и цепям синхронизации) экономичная работа крайне важны для компьютеров, работающих в критических условиях. Совокупность этих свойств гарантирует циркуляцию в системе более достоверной информации, что и нужно для обеспечения устойчивости вычислений и хода вычислительного процесса.

7.4. Архитектура, элементная база и самосинхронизация

Наметившийся переход в сложных СБИС-системах к параллельным асинхронным архитектурам, для которых глобальная синхронизация является противоестественной, требует адекватной схемотехнической поддержки. Такой поддержкой может служить использование ССС-схемотехники для реализации сверхкрупных (выше уровня СБИС) интегральных схем. Основанием для такого утверждения является, кроме сказанного в [3-7], также и то, что практически на всех последних международных научных конгрессах, конференциях и симпозиумах все более часто используется термин "*системы, управляемые событиями*", что на уровне схемотехнической реализации является синонимом термина "*самосинхронизация*".

Распространение принципа самосинхронизации на уровне архитектуры блочно-регистровой и системной (уровень функциональных модулей - устройств и системного интерфейса) открывает единообразный путь создания полноценных истинно параллельных асинхронных КС. Это представляется реальным потому, что в основе идеи самосинхронизации лежит естественный принцип,

состоящий в чередовании двух фаз - активной (рабочей) и релаксационной (самовосстановления - подготовки к новой рабочей фазе), с возможностью контроля их завершения. Идея самосинхронизации на схемотехническом уровне - это возможность контроля за длительностью фазового перехода (аналога синхротакта) схемы встроенной подсхемой, индицирующей момент окончания ее переходного процесса.

На регистровом уровне параллельное асинхронное поведение порождается взаимодействием фазовых сигналов (вернее, событий, состоящих в изменениях фазовых сигналов).

На блочно-модульном уровне управление параллельным асинхронным поведением, базирующееся на ССС-схемотехнике, существенно эффективнее аналогичного синхронного. Это объясняется идеологическим совпадением теоретического и модельного базиса асинхронных систем и ССС-схемотехники. Проектирование системы и поддерживающей ее схемотехники может проходить с единых методологических позиций. Асинхронное (запрос-ответное) взаимодействие элементов системы (устройств, модулей) наиболее естественно реализуется именно средствами самосинхронизации, что показывает необходимость использования их в ЕНК.

Последовательное (на всех уровнях архитектуры, программного обеспечения и схемотехники) применение принципов управления развитием процессов на основе саморазворачивающихся данных и самосинхронности обеспечат разрабатываемому ЕНК единство и целостность, а также выход на более высокий уровень общесистемной организации.

7.5. Долговечность, сохраняемость, надежность и самосинхронизация

Казалось бы, что общего между этими понятиями? Тем не менее,

имеется прямая взаимосвязь долговечности, сохраняемости и надежности с самосинхронизацией. Самосинхронные интегральные схемы, обладающие свойством параметрической устойчивости (нечувствительности к изменениям параметров связей и транзисторов при старении, колебаниях температуры и напряжений источников питания), имеют существенно лучшие показатели долговечности и сохраняемости. Для более подробного изучения причин этого вопроса рекомендуем обратиться к [11].

8. ТЕНДЕНЦИИ РАЗВИТИЯ КС И САМОСИНХРОНИЗАЦИЯ

Несмотря на очевидные преимущества, самосинхронная организация КС продолжает оставаться вне интересов ведущих производителей средств компьютерной техники. Современные синхронные компьютерные системы развиваются в соответствии с тенденциями, исторически сформировавшимися на этом рынке. Таких тенденций несколько, но среди них особо выделяются две, что позволяет считать их тенденциями-стимуляторами развития всех классов компьютеров:

1) *увеличение производительности* и, как следствие, функциональных возможностей;

2) *повышение надежности функционирования.*

Первая тенденция способствовала созданию разнообразных по быстродействию и функциональным возможностям избыточных КС - от карманных калькуляторов до суперкомпьютеров. Дальнейшее повышение производительности таких систем однозначно связывается с использованием параллелизма (параллельной обработки данных), то есть с выбором структур, в которых уровень производительности ограничивается лишь числом исполнительных элементов,

поддерживающих одновременную обработку. Перспектива - системы с высокой степенью параллелизма.

Вторая тенденция - стремление к эксплуатационному совершенству - привела к созданию класса отказоустойчивых КС, способных функционировать непрерывно в течение десятков и даже сотен тысяч часов за счет введения в их структуру аппаратной избыточности (при восстановлении отказавших съемных элементов и своевременном возврате их в систему вручную). При этом существенно повышается объем резервных функциональных устройств и подсистем, реализуемых в виде СБИС-схем.

В последние годы в работах по перспективным компьютерам явно просматривается стремление к слиянию этих тенденций для получения новых свойств, переводящих разрабатываемые КС в разряд машин нового поколения.

Принцип самосинхронизации, в силу его универсальности, удовлетворяет интеграции тенденций, рассматривая проблемы производительности и надежности как две стороны одной медали и обеспечивая тем самым их успешное решение.

9. ЭКОНОМИКА И САМОСИНХРОНИЗАЦИЯ

Проект ЕНК, помимо высокой научной значимости, имеет перспективу и в экономическом плане. Экономический эффект от разработки собственно проекта трудно подсчитать, поскольку он посвящен созданию КС, не имеющей аналогов. Экономический эффект от использования ЕНК на объектах автоматизации (и, соответственно, срок их окупаемости) будет определяться их технико-экономическими характеристиками, а также функциональными возможностями и качественно новыми свойствами, прежде всего - *гарантоспособностью*.

Гарантоспособность - свойство КС, дающее пользователю

возможность обоснованно полагаться на выполнение услуг, для которых она предназначена [9] - несомненно скажется на величине экономического эффекта от использования ЕНК. Просматриваются следующие источники его получения:

- *существенное увеличение срока службы* электроники ЕНК (в два раза и более, что уравнивает ее по сроку службы с основным технологическим оборудованием объектов управления) из-за нечувствительности ССС-схем к процессам старения, обычно вызывающим параметрические отказы в аппаратуре синхронных интегральных схем;

- *увеличение объема полезных вычислений* в связи с увеличением срока службы ЕНК и, соответственно, рабочего времени;

- *сокращение числа резервных ЕНК и затрат на замену* в связи с уравниванием сроков службы компьютера и технологического оборудования;

- *снижение потерь рабочего времени* из-за уменьшения риска получения недостоверных результатов в процессе вычислений, сокращения простоев и аварийности на объектах автоматизации по вине ЕНК.

Из сказанного следует, что самосинхронизация, являющаяся базовой составляющей гарантоспособности, может вносить существенный вклад в окупаемость СРВ на основе ЕНК.

Наличие экономической составляющей можно рассматривать как дополнительный фактор в пользу перехода к самосинхронизации в WSI-системах.

10. ВЫВОДЫ

1. Принцип самосинхронного взаимодействия (ССВ) в применении к компьютерам для СРВ исключительно плодотворен и

универсален. С его помощью можно преодолеть многие трудности проектирования ресурсов WSI-систем.

2. В связи с уникальностью ССВ-принципа была предпринята попытка всесторонней проверки его достоинств в рамках проекта ЕНК.

Результаты исследований:

а) Обоснован выбор ССС-схемотехники в качестве базы проектирования сверхкрупных (выше уровня СБИС) интегральных схем.

б) Сформулированы принципы организации архитектуры для ЕНК, начата разработка спецификации архитектуры рекуррентного процессора.

в) Существенно продвинута разработка САПР ССС-ИС RONIS.

г) Определены потенциальные области применения ЕНК (АСУ ответственными объектами и процессами в реальном времени), и сформулированы требования к свойствам ЕНК для работы в составе СРВ: производительность, безопасность применения, живучесть, увеличенный срок службы, экологическая безопасность, развиваемость, экономическая окупаемость. Интеграция перечисленных свойств должна проявиться в ЕНК в виде обобщающего свойства - гарантоспособности.

д) Выявлены факторы, указывающие на целесообразность использования ССВ-принципа в компьютерах, когда уровень сложности последних превышает уровень СБИС:

- переход к более высокоорганизованным архитектурам (структурный, задачный, командный и функциональный параллелизмы, интеграция функций непосредственно в архитектуру, событийное взаимодействие и др.);

- повышение уровня сложности аппаратуры и программ без снижения производительности, достоверности и устойчивости

вычислений и др. показателей качества;

- адекватность элементной базы уровню организации архитектуры аппаратных средств;

- согласованность с действующими тенденциями развития КС;

- необходимость поиска новых источников экономической окупаемости для систем сверхкрупной (Ultra,- Giga- и Wafer Scale Integration-уровней) интеграции.

3. Предложена нетрадиционная (рекуррентная) архитектура для ЕНК, отличающаяся новыми принципами - рекуррентности процессов и языков, единого информационного потока саморазворачивающихся данных и "инструкций", самосинхронного взаимодействия и других. Все принципы, связанные между собой как взаимодополняющие, составляют концептуальную базу архитектуры.

Самосинхронизация и рекуррентность отвечают требованиям построения WSI-систем, обеспечивая концептуальное, информационное, аппаратное и программное единство реализации.

ЛИТЕРАТУРА

1. *А.В. Филин, Ю.А. Степченков, В.С. Петрухин, Ф.И. Гринфельд.* Исследование нетрадиционных подходов к созданию компьютеров гарантированной высокой надежности. // Системы и средства информатики, вып. 5. - М.: Наука, Физматлит, 1993, с. 181-196.
2. *И.А. Мизин, А.В. Филин.* Принципиальная база архитектуры естественно надежных компьютеров. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 172-197.
3. *А.В. Филин, Ю.А. Степченков.* Схемотехника интегральной элементной базы естественно-надежных компьютеров. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 222-

- 239.
4. *Ю.А. Степченков, А.В. Филин.* Способы построения самосинхронных комбинационных схем естественно-надежных компьютеров. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 239-258.
 5. *Л.П. Плеханов.* Базовые элементы самосинхронных схем. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 258-264.
 6. *Л.П. Плеханов.* Фазовый анализ самосинхронных схем. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 264-273.
 7. *Л.П. Плеханов.* Условия самосинхронности в разветвлениях и физическая реализуемость самосинхронных схем. // Системы и средства информатики, вып. 7. - М.: Наука, Физматлит, 1995, с. 273-278.
 8. *Л.П. Плеханов.* Автоматизация разработки электронных строго самосинхронных КМДП-схем. // Наст. сб.
 9. *A. Avizienis, J.C. Laprie.* Dependable Computing: From Concepts to Design Diversity. // Proc. IEEE, vol. 74, No 5, May 1986, pp. 629-638.
 10. *B. Robinson.* An ultrareliable future. // Electronic Engineering Times, 1991, February 25, p. 32.
 11. *А.В. Филин.* Самосинхронизация - естественный путь обеспечения долгоживучести интегральных схем. // Наст. сб.