

КОМПЬЮТЕРЫ БЕЗ СИНХРОНИЗАЦИИ

А.В. Филин, Ю.А. Степченко

ВВЕДЕНИЕ

С момента появления компьютеров их производительность повышается, в основном, за счет усложнения *структуры устройств* (повышения степени параллелизма) и увеличения *степени интеграции* и *тактовой частоты* элементной базы. В последние годы частота синхронизации стала для компьютеров "числом Маха" - той единственной характеристикой, которая позволяет различать компьютеры, идентичные по всем другим показателям. Ввиду этой символической значимости частоты синхронизации кажется невозможным представить себе компьютер без традиционной (принудительной) синхронизации.

Однако рост степени интеграции ИС и связанные с ним проблемы проектирования и реализации синхронных схем с уровнем интеграции, превышающим СБИС-уровень сложности (*Ultra-, Giga- и Wafer Scale Integration*), заставляют разработчиков СБИС-компьютеров (компьютеров на сверхбольших интегральных схемах) искать пути их создания, не использующие принцип синхронного взаимодействия (*C-принцип*), который является ключевым в организации функционирования современных (*синхронных*) СБИС-компьютеров всех видов. К таким проблемам, в частности, могут быть отнесены: повышение вероятности сбоев вследствие роста числа "точек" электронного арбитража сигналов; увеличение временных потерь из-за перекоса ("расползания") синхросигналов в линиях; отрицательное воздействие параметрической нестабильности элементов; трудность

выполнения требований к точности и стабильности частоты синхронизации и ряд других.

Отказ от С-принципа означает исключение из структуры компьютера подсистемы *принудительной* синхронизации и обеспечение взаимодействия остальных подсистем и устройств каким-либо другим способом. На сегодняшний день известны, по крайней мере, три альтернативных принципа (и, соответственно, способа) координации взаимодействия компонентов системы:

- *асинхронный* (А-принцип);
- *квазисамосинхронный* (КСС-принцип).
- *асинхронный аperiodический* или *строго самосинхронный* (ССС-принцип).

А-принцип базируется на асинхронной логике, следствие которой - асинхронная схемотехника, определяющая правила проектирования асинхронных СБИС (А-СБИС). Аналогично, ССС-принцип базируется на собственной теории *полумодулярных* (самосинхронных) *схем* и самосинхронной схемотехнике, позволяющей проектировать как строго самосинхронные, так и квазисамосинхронные СБИС. КСС-принцип занимает промежуточное положение между А- и ССС-принципами. Он также базируется на теории полумодулярных схем, однако в некоторых случаях допускает смягчение ее жестких правил. Такой подход позволяет уменьшить аппаратные затраты, требующиеся для реализации ССС-схем, но при этом теряются некоторые уникальные характеристики, свойственные ССС-схемам.

Асинхронная логика в применении к полупроводниковым СБИС мало известна большинству разработчиков, которые десятилетиями полагались и продолжают полагаться на синхронную схемотехнику. В настоящее время разработки асинхронных архитектур переросли чисто

научно-исследовательские рамки, и наряду с экспериментальными разработками [1, 2] делаются попытки применить А-логику в коммерческих СБИС (пример: Manchester University, проекты Amulet, Amulet 1 и Amulet 2e по разработке асинхронного RISC-микропроцессора, полностью совместимого с синхронным RISC-микропроцессором ARM6 [3, 4, 5]).

1. ОСОБЕННОСТИ СПОСОБОВ СИНХРОНИЗАЦИИ (СОПОСТАВЛЕНИЕ НА КАЧЕСТВЕННОМ УРОВНЕ)

Несмотря на кажущуюся очевидность различий рассматриваемых подходов, в литературе нет четкости относительно применяемых в них терминов "синхронный" (timing, synchronous), "асинхронный" (asynchronous) и "самосинхронный" (self-timing, self-synchronous). Например, в статических ЗУПВ-ИС [6, 7] *синхронные* схемотехнические решения объявляются *самосинхронными* по причине использования собственного внутрикристального тактового генератора в дополнение к управлению от системного (внекристального) генератора. Действительно, такое техническое решение позволяет упростить ряд проблем, связанных с использованием системной (глобальной) синхронизации, но не исключить их полностью. В [8] *асинхронная* реализация конвейера с использованием встроенных задержек объявляется как *самосинхронная*. Таких примеров можно привести множество. Поэтому уточним смысл терминов, обозначающих принципы синхронизации.

Все устройства современного С-компьютера (процессор, память, дисковая подсистема, последовательный порт, различные контроллеры) работают в такт с главным системным генератором. Такая схема организации операций обмена информацией и сигналами между

устройствами и блоками в СБИС-системах стала *традиционной* или *классической*.

Попробуем понять своеобразие С-, А- и ССС-принципов на примере взаимодействия устройств по общему для них интерфейсу в рамках операций обмена "один - одному", "один - всем" и "все - одному".

Образно С-принцип взаимодействия участников обмена может быть представлен в виде, например, езды автомобилей по однополосной (не допускающей обгона) кольцевой дороге, регулируемой светофорами. Независимо от индивидуальных скоростных возможностей участников движения, время езды по кольцу (длительность цикла) определяется возможностью самого медленного участника. (Из этой аналогии понятна правомерность утверждения, что С-принцип вынужденно ориентируется на худший случай.) Число светофоров (временных стоянок) выбирается из числа необходимых фаз обмена (подциклов). Обмен информацией между участниками происходит на этих временных стоянках без диалога, по умолчанию. Предполагается, что раз участник достиг стоянки, то необходимая ему информация уже готова. Скорость на разных участках дороги может быть разной (не совпадать со скоростью на других участках), но постоянной и определяется возможностью самого медленного участника на каждом из участков. Участники, которые не могут выдерживать заданный темп движения, из процесса исключаются.

Рассмотренная картина движения, естественно, отличается от реальной картины, однако помогает понять, почему теоретически синхронный принцип - самый медленный и более энергоемкий, чем альтернативные принципы. В каждом конкретном цикле обмена в число его возможных участников, как правило, попадает лишь небольшая их

часть, например, два участника в режиме "один - одному". Однако для того, чтобы эти два участника смогли произвести обмен информацией, все потенциальные участники вынуждены принимать сигналы синхронизации, которые вызывают переключения их элементов, а значит и расход энергии. Даже когда необходимости в обмене нет, все устройства С-системы вынуждены принимать в нем какое-то участие (совершать холостые ходы).

Быстродействующий участник мог бы скорее достичь цели, но он "притормаживается" самым медленным участником. В результате потенциальное быстродействие участников недоиспользуется, и к тому же перерасходуется энергия, которая могла бы пойти на поддержание полезной, а не холостой, работы.

В отличие от синхронных, каждая из асинхронных и самосинхронных схем работает со своей собственной скоростью, управляемая собственным "таймером" и готовностью данных, необходимых для начала работы.

А- и ССС-принципы, в рамках нашего примера, сравнимы с многополосной кольцевой дорогой, где число полос совпадает с числом участников. До тех пор, пока не появится необходимость, движение по трассе не начнется, хотя моторы участников включены, и они готовы к началу движения (вполне возможен вариант и с выключенными моторами, но с потерей времени на их включение и вхождение в режим готовности к движению).

А- и ССС-принципы отличаются от С-принципа *отсутствием пустых пробегов*. Как только появляется инициатор обмена, все участники А-системы начинают движение до первого "светофора" - конца первого участка движения (эквивалент адресной фазы цикла обмена). Аналогично С-системе, на этом участке все участники

движутся с одной и той же скоростью, определяемой самым медленным участником. В конце первого участка определяется, кто из участников будет выполнять обмен: кто-либо один (при операциях "один - одному"), какая-то группа участников (при операциях "один - некоторым") или все (при операциях "один-всем").

Невыбранные участники возвращаются на исходную позицию (начало первого участка), а выбранные - продолжают свое движение дальше, каждый со своей скоростью. В конце второго участка все участники ждут самого медленного участника и продолжают движение дальше. На каждом из "светофоров" между участниками происходит диалог - подтверждение готовности двигаться дальше. Теоретически средняя скорость движения по такой "многополосной дороге" будет выше, чем по "однополосной", и будет определяться быстродействием участников. Необходимая для полезной работы энергия также будет меньше из-за отсутствия холостых пробегов (переключений элементов) у тех участников, которые не привлекаются к текущему обмену.

Следует подчеркнуть, что в А-системе *нет строгого контроля окончания переходных процессов*, и приход участника к концу какого-либо участка не всегда реален. А-принцип допускает использование гипотезы ограниченного сверху времени протекания отдельных процессов. Запустив какой-либо процесс, А-система позволяет формирование сигнала об его окончании по истечении определенного времени (временной уставки), хотя реально процесс может быть завершён раньше или может не уложиться в это время. В рамках нашего абстрактного примера это означает, что на трассах одного или нескольких участников могут встретиться участки с постоянной скоростью движения, определяемой их дорожными знаками. Если С-системы вынужденно ориентируются на худший случай в глобальном

плане (для всех участников и во всех фазах), то А-системы - только в локальном плане (для тех участников и тех участков, где используется гипотеза ограниченного времени протекания процессов).

ССС-принцип базируется на контролируемости моментов окончания во времени всех процессов и не допускает использования гипотезы об их ограниченности. Длительность любого процесса здесь может быть сколь угодно большой, но конечной величиной. Контроль достигается использованием специальных способов передачи информации о движении и слежением за моментами окончания переходных процессов. В СССР-системе отсутствуют участки без контроля действительного окончания движения участников.

По существу, это единственное отличие между А- и СССР-системами. Но оно кардинально меняет методологию построения СССР-аппаратуры и программных средств автоматизации их проектирования.

В КСС-системах на каких-то участках (достаточно протяженных, но редко используемых) применяется контроль действительного окончания событий (аналогично СССР-системам), а на других (достаточно коротких и часто используемых) - действует гипотеза ограниченного времени протекания событий (аналогично А-системам).

Подводя итог сопоставлению принципов координации событий, можно констатировать следующие различия:

- С-система использует *совокупность генераторов* - общесистемных и локальных, обходясь без обмена сигналами-квитанциями между взаимодействующими устройствами;
- А-система использует *совокупность встроенных задержек*;
- КСС-система использует *комбинацию встроенных задержек и индикаторов окончания переходных процессов*.

- ССС-система содержит *совокупность индикаторов окончания переходных процессов*.

В последних двух системах взаимобмен сигналами-квитанциями обязателен, что в ряде случаев может снижать быстродействие обмена.

В реальных компьютерных системах, как правило, одновременно используются разные принципы (и, соответственно, схемотехнические решения). Можно встретить смешанные решения: синхронно-асинхронные, асинхронно-самосинхронные, синхронно-квазисамосинхронные и т. д.

2. СРАВНЕНИЕ С-, А- И ССС-СПОСОБОВ ВЗАИМОДЕЙСТВИЯ НА ПРИМЕРЕ ИНТЕГРАЛЬНОЙ СХЕМОТЕХНИКИ

Проблема координации событий и согласования работы во времени различных частей компьютеров (от отдельных компонентов ИС до подсистем) является одной из основных.

С-способ, наиболее широко применяемый в реализации СБИС, связан с использованием механизма тактирования и жесткого ограничения времени всех фаз обмена (T) взаимодействующих компонентов - фазы установления связи (T_c), собственно фазы обмена (T_e) и фазы разъединения связи (T_d). В качестве примера, иллюстрирующего обмен на основе С-способа, на рис. 1а представлена временная диаграмма обмена информацией между устройствами - *задатчиком* (инициатором обмена) и *исполнителем* - по системному интерфейсу магистрального типа (например, Multibus II) с использованием системы тактовых генераторов $\{T_i\}$, в данном случае двух - T_1 и T_2 . Длительность синхросигналов T_1 и T_2 , определяющая быстродействие реализуемой схемы (скорость обмена), выбирается в расчете на худший случай, учитывающий максимально возможные

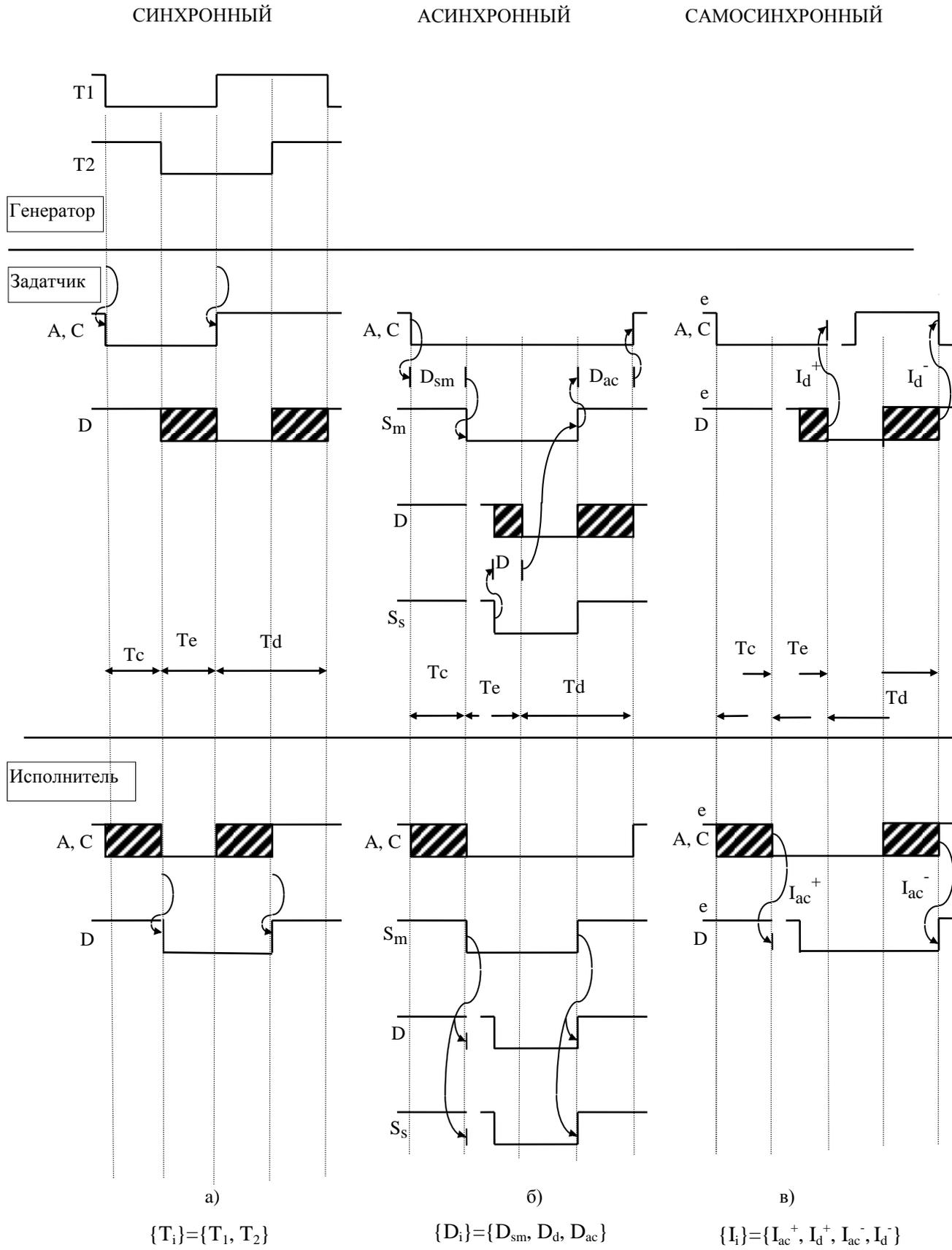


Рис. 1. Операция "чтение" между устройствами: задатчиком и исполнителем

времена переключения элементов и неблагоприятные сочетания условий их функционирования (напряжений питания, температуры, длины соединений и т.п.). Поэтому синхронные схемы работают, как правило, медленнее, чем могли бы, если исходить из реального быстродействия элементов и реальных условий применения.

Контроль переходных процессов в С-схемах не осуществляется.

В случае А-способа один из компонентов T , а именно T_e , становится асинхронным (длительность фазы обмена не ограничена сверху) и определяется быстродействием элементов исполнителя. Вероятность обмена недостоверной информацией уменьшается, но не исчезает, так как длительности фаз обмена и разъединения связи ограничены (T_c и T_d остаются синхронными, то есть не контролируемыми по времени параметрами). Асинхронный способ характеризуется использованием системы встроенных задержек $\{D_i\}$ для координации событий.

Полное устранение ограничений на длительность всех фаз обмена достигается в ССС-схемотехнике, которую можно рассматривать в качестве альтернативы обоим традиционным подходам. В ССС-схемотехнике система тактовых генераторов ($\{T_i\}$ - при синхронном подходе) и система фиксированных задержек ($\{D_i\}$ - при асинхронном подходе) заменены системой логических индикаторов окончания переходных процессов $\{I_i\}$. Индикаторы реализуют механизм обнаружения и фиксации моментов завершения переходных процессов в ССС-схеме, вызванных изменением входного воздействия.

На рис. 1в видно, что все компоненты фаз ССС-обмена не лимитированы и определяются временем реакции партнера по обмену (временем окончания в нем переходных процессов). Длительность каждой из фаз обмена может быть произвольной, но конечной

величиной. Здесь показан пример использования одного из видов СС-кода - *парафазного кода со спейсером* [9] - одного из наиболее простых по пониманию и универсальности (но не по эффективности). Каждому из разрядов используемых шин (адреса и данных) соответствует два провода. Из четырех возможных комбинаций кодов на этих проводах два ("все единицы" или "все нули") используются как *спейсер* - некоторый служебный (нерабочий) набор, по которому определяется факт окончания переходного процесса в обмене.

Быстродействие ССС-обмена определяется реальными задержками элементов и при прочих равных условиях выше, чем при синхронном и асинхронном обменах. Контроль переходных процессов всех фаз ССС-обмена обеспечивает повышенную достоверность обмена информацией.

Таким образом, используемый в реальной практике проектирования асинхронный принцип обмена таковым на самом деле не является. Термин "асинхронный" применим только к одной из трех фаз - собственно фазе обмена *Te*. Другие же фазы, по существу, являются синхронными, в реализации которых вместо системы тактовых генераторов используется система фиксированных задержек.

При ССС-обмене понятие времени исключено; все фазы обмена - асинхронные и могут иметь любую, но конечную длительность. С этой точки зрения ССС-принцип является правильной и полной (распространяется на все фазы обмена) реализацией А-принципа.

На рис. 2 и 3 приведены диаграммы изменений - сигнальные графы обмена информацией между задатчиком и исполнителем - с использованием А- и ССС-способов, соответственно.

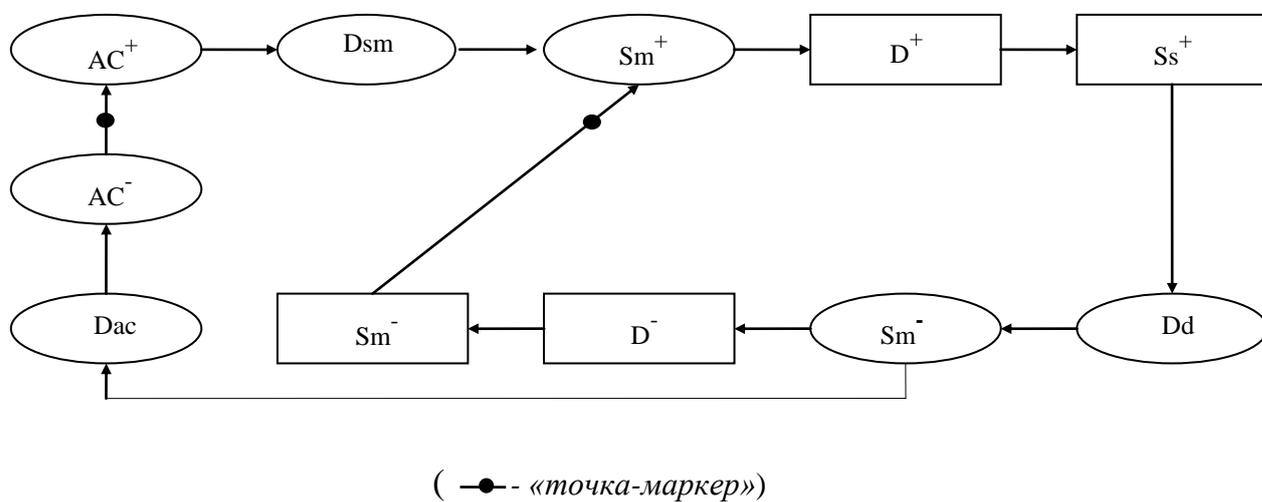


Рис. 2. Граф изменений при асинхронном подходе (операция "чтение")

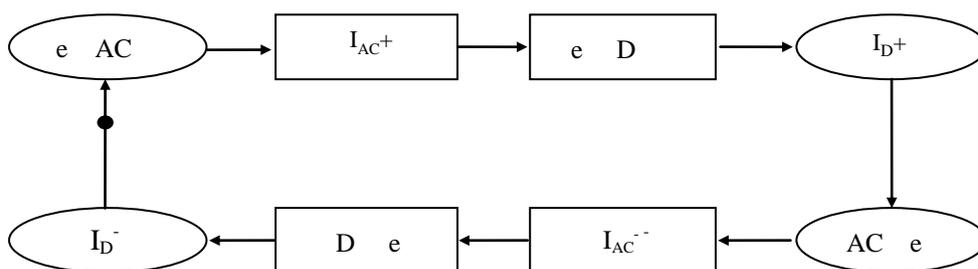


Рис. 3. Граф изменений при строго самосинхронном подходе (операция "чтение")

Сигнальный граф - средство описания взаимодействия процессов, в данном случае - протокола обмена посредством системной магистрали. На диаграммах переходы задатчика изображены овалами, а переходы исполнителя - прямоугольниками.

Каждая из дуг сигнального графа может быть маркирована любым целым числом "точек-маркеров" [6]. Правила срабатывания вершин графа аналогичны правилам срабатывания переходов в сетях Петри. Дугам маркированного графа соответствуют позиции сети Петри. Каждой вершине маркированного графа ставится в соответствие одно из следующих событий:

- изменение состояния некоторой моделируемой переменной;
- обозначение определенной встроенной задержки, компенсирующей, например, перекося в распространении сигналов по параллельным линиям связи и т.п.;
- "чистые синхронизаторы" - дополнительные вершины, не имеющие семантической интерпретации.

Надстрочный знак "+" в названии переменной используется для обозначения установки сигнала на линию обмена, знак "-" - для обозначения его снятия. Вершины графа, моделирующие встроенные задержки, обозначены через D_k , где k принимает значения sm , d и ac . Исходная маркировка сигнального графа на рис. 2 соответствует состоянию линий связи, при котором на шинах адреса (A), управления (C) и данных (D) информации нет, линии синхронизации (S_m и S_s) сброшены, временные задержки выключены.

На рис. 3 представлен сигнальный граф, соответствующий временной диаграмме ССС-обмена на рис. 1. Переход e -- $AC(D)$ означает смену спейсера сообщением, задаваемым кодовым набором СС-кода (парафазным); $AC(D)$ -- e - обратный переход. Из рис. 3 видно,

что обмен осуществляется без использования фиксированных задержек и без тактовых генераторов. Служебные сигналы индикаторов I_{ac} и I_d формируются и используются внутри взаимодействующих устройств. Между устройствами нет обмена этими сигналами. Граф симметричен, так как каждое устройство, независимо от своего статуса, одинаково выполняет свою часть обмена. Это - "чистый" обмен.

3. КВАЗИСАМОСИНХРОНИЗАЦИЯ

Полностью реализовать все преимущества ССС-схем возможно только на пути строгого соблюдения гипотезы Маллера [9,10]. Одно из положений гипотезы определяет требование к характеру задержек: элемент самосинхронной схемы характеризуется инерциальной задержкой произвольной, но конечной длины. Нарушение этого требования позволяет уменьшить "цену" ССС-реализации (см. [11]). Схемы, работоспособные при *определенных соотношениях задержек* некоторых элементов, получили название *квазисамосинхронных* схем.

Ослабление требований к соотношениям задержек расширяет класс схем, которые рассматриваются как самосинхронные. Естественно, что реализация таких (квазисамосинхронных) схем проще. Однако вопрос о том, компенсирует ли выигрыш сопутствующие потери, остается открытым. Например, отказ от строгой самосинхронности - переход к учету "разумных соотношений" задержек элементов - фактически означает отказ от самодиагностических свойств схем (полная информация о завершении переходных процессов в элементах уже не доставляется к выходу схемы). Диагностика неисправностей и отказоустойчивость квазисамосинхронных схем становятся более сложными проблемами, чем для схем с произвольными соотношениями задержек.

Разработка квазисамосинхронных схем требует использования проблемно-ориентированных методов их анализа и синтеза, поскольку применение для них методов строгой самосинхронизации невозможно. Традиционно существует мощный комплекс моделей и методов описания, анализа и синтеза синхронных и классических асинхронных схем со всеми их преимуществами и недостатками, базирующийся на гипотезе ограниченной ("паспортной") задержки. Более предпочтительна разработка методов анализа и синтеза квазисамосинхронных схем на основе событийных моделей типа *диаграмм изменений*.

Квазисамосинхронные схемы можно рассматривать как переходной этап на пути к строго самосинхронным схемам.

4. ПРЕИМУЩЕСТВА АСИНХРОННОЙ РЕАЛИЗАЦИИ

Несмотря на то, что С-схемы проще, более экономичны в разработке, и опыт их проектирования огромен, они обладают рядом принципиальных недостатков: они достаточно медленны, плохо контролируемы и энергорасточительны. По мере роста рабочей частоты С-СБИС проблемы и ограничения усиливаются. Пропорционально растет интерес к асинхронной схемотехнике.

Переход на асинхронную логику в СБИС кажется привлекательным, поскольку должен сопровождаться:

- снижением рассеиваемой мощности;
- устранением проблем перекоса синхросигналов;
- повышением гибкости в проектировании;
- упрощением задачи отвода тепла от кристалла схемы.

Малая рассеиваемая мощность - главная причина интереса к А-схемам. Асинхронное устройство (например, процессор), в отличие от своего синхронного аналога, должно затрачивать столько энергии,

сколько нужно для работы в данный момент. Любая схема на кристалле, не используемая текущей инструкцией, находится в состоянии покоя и не потребляет мощности. В то же время она не выключается и готова начать работать, как только поступят данные. Схемы таким свойством не обладают.

Придумано немало приемов, с помощью которых разработчики могут уменьшать потребляемую мощность. Наиболее популярный состоит в использовании специальных схем, которые замедляют или даже останавливают генератор системы, когда ее процессор простаивает. Такой подход помогает экономить энергию, но факт, что синхронным устройствам *органически* присуща растрата мощности, остается фактом. Даже когда схемы не участвуют в полезной работе, синхронизация заставляет их "маршировать на месте" и тратить энергию. Попытки достичь "мелкозернистого" (например, на уровне функциональных узлов) уровня экономии энергии в синхронных устройствах влекут за собой столь сложные схемы сравнения и переключения, что результат получается прямо противоположным.

Другое преимущество асинхронной работы схем видится в том, что исключается проблема разводки по кристаллу СБИС глобальных и, как правило, самых быстрых синхросигналов. Устранение их перекосов - серьезнейшая проблема, усугубляющаяся с возрастанием частоты. Кроме того, на кристалле высвобождается достаточно большая площадь, занимаемая системой синхронизации. Например, на кристалле RISC-процессора Alfa фирмы DEC более 25 % площади кремния занимает логика синхронизации.

Еще одно преимущество А-логики - она предоставляет разработчикам больше возможностей для улучшения схем. Например, А-сумматор может быть построен так, что скорость его работы будет

меняться в зависимости от числа переносов, порождаемых операндами, в то время как С-сумматор проектируется на худший случай - на возникновение переносов в каждом разряде. Ожидается, что А-схемы в среднем будут быстрее С-схем хотя бы потому, что они управляются внутрисхемными событиями. Поэтому асинхронная логика отлично подходит для построения конвейерных схем и схем передачи данных между устройствами, различными по быстродействию.

Снижение мощности стало весьма актуально не только из-за роста интереса к портативным, блокнотным и бортовым компьютерам и устройствам, но и по другим, менее очевидным причинам. В частности, когда появятся СБИС с сотнями миллионов транзисторов, во весь рост встанет задача отвода тепла от кристалла. Если иметь в виду, что рассеяние тепла - серьезная проблема уже текущего поколения технологии, кристаллы большей плотности утяжелят ее в еще большей степени.

5. ПРЕИМУЩЕСТВА САМОСИНХРОННОЙ РЕАЛИЗАЦИИ

Теоретические основы самосинхронизации изложены в [9]. Способ самосинхронизации, когда схема (устройство) сама сигнализирует об окончании очередного этапа своей работы, обеспечивает таким схемам определенные преимущества перед обычно применяемыми схемами.

Преимущества, достигаемые применением ССС-схем в компьютерной технике (сравнительно с С-схемами):

1) *увеличение быстродействия до предельных значений* (не менее чем в 2 раза) - за счет работы ССС-схем по реальным задержкам на всех уровнях (от вентиля до системы в целом);

2) *повышение устойчивости функционирования* - в силу нечувствительности ССС-схем к перекосам и гонкам сигналов, к

разбросу параметров из-за старения материалов, изменения температуры и напряжения источника питания (до 50% в обе стороны);

3) *повышение внутренней безопасности функционирования* - прекращение работы ССС-схем в момент возникновения неисправности; поскольку самоконтроль и локализация ошибок являются естественными свойствами ССС-схем, в их механизме самоконтроля не существует "проблемы сторожей" (контроля схем контроля);

4) *повышение достоверности обработки информации* - снижение риска получения недостоверных результатов благодаря оперативному обнаружению константных неисправностей (100 %) и отсутствию шумопроизводящей системы тактовой синхронизации;

5) *увеличение срока службы СБИС-приборов* (в несколько раз) - благодаря свойству ССС-схем сохранять работоспособность вплоть до момента разрушения топологических структур (транзисторных каналов, проводников и т.д.);

б) *умеренное энергопотребление* (сравнительно с самопроверяющейся и отказоустойчивой синхронной схемой-аналогом) - за счет исключения постоянно потребляющей системы синхронизации и способности ССС-схем автоматически переходить в энергосберегающий режим в пассивном состоянии;

7) *продление срока работы систем с батарейным питанием* - в силу способности ССС-схем функционировать при пониженном напряжении питания и увеличения, таким образом, времени использования батарей;

8) *повышение технологичности изготовления* - из-за исключения проблемы разводки по кристаллу цепей глобальной синхронизации (обычно занимают около четверти площади кристалла синхронной

СБИС), увеличения выхода годных схем, прямого автомасштабирования масок при переходе к другим технологическим нормам;

9) *снижение расходов на производство* - достаточно функционального тестирования, поскольку ССС-схемы являются самопроверяющимися;

10) *улучшение обслуживаемости и снижение расходов на восстановление систем после отказов* - из-за полной оперативной самопроверяемости и самодиагностируемости ССС-схем относительно константных неисправностей.

Как видим, ССС-схемотехника способна устранить ряд проблем, препятствующих переходу на новые уровни интеграции - к созданию ULSI-, GSI- и WSI-схем. Наиболее значительный из ее недостатков: ССС-схемы содержат большее число транзисторов и трасс соединений, что удорожает их стоимость.

6. КОЛИЧЕСТВЕННАЯ ОЦЕНКА ДОСТОИНСТВ СХЕМОТЕХНИК

Для подтверждения объявленных достоинств ССС-схем на количественном уровне были проведены модельные испытания специально выбранных тестовых функциональных устройств (ТФУ). Они должны были удовлетворять целому ряду требований и, в первую очередь, быть функционально законченными, расширяемыми по разрядности и достаточно простыми в реализации, чтобы можно было рассмотреть разные варианты их исполнения: *традиционный* (несамопроверяющийся, безызбыточный), *самопроверяющийся* и *отказоустойчивый*.

Все варианты схем ТФУ различной разрядности сравнивались между собой по затратам на реализацию (по числу транзисторов) и

быстродействию. Чтобы сравнительные оценки схемотехнических решений были корректны, использовался один и тот же КМОП-технологический базис, для которого имелся полный набор необходимых топологических и технологических параметров. В результате моделирования удалось выявить предпочтительные области применения ССС-схемотехники.

Результаты моделирования различных вариантов комбинационной и регистровой С- и ССС-схем ТФУ представлены в [11]. Там же содержатся и выводы по итогам моделирования.

ЗАКЛЮЧЕНИЕ

1) *Переход на ССС-схемотехнику - эффективное решение проблем синхронных систем.*

С переходом на субмикронную технологию, ростом архитектурной сложности и быстродействия цифровых КС быстро усугубляются трудности глобальной синхронизации от внешнего источника. С другой стороны, наметившийся переход в СБИС-системах к параллельным архитектурам, для которых противоестественна общая (системная) принудительная синхронизация, требует адекватной схемотехнической поддержки. Синхронный подход с этими проблемами плохо согласуется.

Выход из складывающейся ситуации состоит в использовании в элементной базе КС реального времени ССС-схемотехники. Построение ИС любой степени сложности на принципе самосинхронизации позволяет создавать компьютеры, свободные от недостатков принудительной синхронизации и обладающие уникальными характеристиками.

2) *Переход на ССС-схемотехнику в ИС с уровнем интеграции выше СБИС - гарантия создания информационно безопасных КС реального времени.*

Элементная база на основе ССС-схемотехники позволяет естественным образом наделять компьютеры рядом уникальных надежностных свойств (без потери производительности), недостижимых в случаях традиционных схемотехник, и, следовательно, отвечает требованиям, предъявляемым к интегральной элементной базе КС критических применений.

Самосинхронным реализациям ИС любой сложности присущи следующие дополнительные уникальные свойства, входящие в новое качество надежности - *гарантоспособность*:

- естественная самопроверяемость и самодиагностируемость;
- гарантированная информационная отказобезопасность;
- повышенная параметрическая отказоустойчивость;
- экстремально возможное быстроедействие.

Перечисленные свойства крайне важны для критических КС. Они автоматически реализуются в ССС-ИС, разгружая разработчиков от необходимости решения соответствующих проблем на верхних уровнях архитектуры. Совокупность указанных свойств обеспечивает гарантию использования в КС достоверной (неискаженной) информации.

Подобные системы попадают в разряд так называемых гарантоспособных систем, где термин "гарантоспособность" выступает как обобщенный показатель надежности. Показатели готовности, безопасности, ремонтоспособности, живучести и другие рассматриваются как частные количественные характеристики различных проявлений одного и того же свойства системы - ее гарантоспособности.

3) *Переход на ССС-схемотехнику и принцип самосинхронного взаимодействия в компьютерных системах достаточно подготовлен.*

Анализ общих свойств и характеристик ССС-схемотехники приводит к следующим выводам:

а) Не существует серьезных причин, препятствующих более широкому развитию ССС-схемотехники и использованию в любых классах систем, но начинать применение следует с создания систем, где ее новые свойства особенно важны: с компьютеров для чрезвычайных (критических) областей применения.

б) Для получения максимума преимуществ от применения ССС-схемотехники целесообразно использовать ее "насквозь" - от источника до приемника информации.

в) При встраивании ССС-схем в традиционные цифровые системы эффект зависит от конкретного окружения, поэтому величину его следует оценивать заранее.

г) Для внедрения принципа самосинхронизации в практику создания КС необходимо интенсифицировать исследования и разработки в областях:

- теории методов и алгоритмов проектирования сверхкрупных строго самосинхронных схем и самосинхронных архитектур для них;
- параметризованных библиотек ССС-элементов общего назначения;
- коммерческих САПР сквозного проектирования сверхкрупных строго самосинхронных схем;
- программных инструментальных систем поддержки процессов проектирования самосинхронных архитектур для КС на основе самосинхронных ИС;

- строго самосинхронных библиотечных базовых и макроэлементов для оперативной и постоянной памяти и интерфейсов.

ЛИТЕРАТУРА

1. *A.J. Martin et al.* The design of an Asynchronous Microprocessor. Tech. Report Caltech-CS-TR-89-2, California Inst. of Technology, Pasadena, Calif., 1989.
2. *K.-R. Cho, K. Okura, K. Asada.* Design of a 32-bit Fully Asynchronous Microprocessor (FAM). Proc. 35th Midwest Symp. Circuit and System, IEEE Press, Piscataway, N.J., 1992, pp. 1,500-1,503.
3. *N.C. Paver.* The Design and Implementation of an Asynchronous Microprocessor. A Thesis Submitted to the University of Manchester for the Degree of Doctor of Philosophy in the faculty of science. Department of Computer Science, 1994.
4. *J.V. Woods et al.* AMULET1: An Asynchronous ARM Microprocessor. IEEE Transactions on Computers, vol. 46, No 4, April 1997, pp. 385-397.
5. *T. Werner, V. Akella.* Asynchronous Processor Survey. IEEE Computer Society, November 1997, vol. 30, No 11, pp. 67-75.
6. *D. Bursky.* Advanced Self-timed SRAM Panes Access Time to 5 ns. Electronic Design, vol. 38, No 4, February 22, 1990, p. 101-108.
7. *D. Pryce.* Self-timed SRAMs offer 5- and 10-nsec cycle times for high-speed systems. EDN, 1990, vol. 35, No 5, 129 p.
8. *J. Compton, A. Albicki.* Self-timed Pipeline with Adder. Proceedings GLSV'92, Second Great Lakes Symposium on VLSI Design Automation of VLSI Systems in the 90's, February 28-29, 1992, Kalamazoo, MI, pp. 109-113.
9. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах. / Под ред. В.И. Варшавского. М.: Наука, 1986, 400 с.
10. *Л.П. Плеханов.* Базовые элементы самосинхронных схем. // Системы и средства информатики. Вып. 7. М.: Наука, Физматлит, 1995, с. 258-264.
11. *Ю.А. Степченко, Ю.Г. Дьяченко, В.С. Петрухин, А.В. Филин.* Цена реализации уникальных свойств самосинхронных схем. // Наст. сб.