

УДК 681.3

ПРОЕКТИРОВАНИЕ СРЕДСТВ СИНХРОНИЗАЦИИ БЛОКОВ ГЛОБАЛЬНО АСИНХРОННЫХ СИСТЕМ С ПРОИЗВОЛЬНОЙ ЛОКАЛЬНОЙ СИНХРОНИЗАЦИЕЙ

В. Б. Мараховский,

доктор техн. наук, профессор

В. Ф. Мелехин,

доктор техн. наук, профессор

Санкт-Петербургский государственный политехнический университет

Рассматриваются подходы к организации взаимодействия блоков в многопроцессорных системах с параллельно протекающими процессами, а также методика проектирования на базе самосинхронных схем устройства (синхростратума), координирующего взаимодействие блоков в глобально асинхронной системе.

Ключевые слова — синхронизация, логическое время, физическое время, синхронные и асинхронные автоматы, массив автоматов, синхростратум, проектирование.

Введение

Синхронизация является одной из важных проблем в цифровых вычислительных системах. Специалисты в области проектирования аппаратуры и программного обеспечения понимают и решают ее по-разному на различных уровнях системной иерархии.

Термин «синхронизация» понимают как координацию событий (сигналов, операций или процессов) во времени. Координация событий отражает причинно-следственные отношения между ними и определяется частичным порядком на множестве событий, выделенных в системе. Такое определение синхронизации соответствует концепции *логического времени*, течение которого связано с наступлением того или иного события. Концепцию логического времени используют разработчики программного обеспечения.

Разработчикам системы и ее аппаратного обеспечения необходимо учитывать, что любая система функционирует в непрерывном *физическом времени*, которое может быть представлено непрерывной независимой переменной. На процессы в аппаратуре существенное влияние оказывают задержки в элементах и связях, а не только события, рассматриваемые в логическом времени. Поэтому при реализации вычислительных систем проектировщику необходимо обес-

печить соответствие следования событий в реальном физическом времени тому частичному порядку, который был заложен в логическом времени.

Группа ученых, возглавляемая профессором В. И. Варшавским (в состав группы входит и В. Б. Мараховский, один из авторов данной статьи), в 1990-е годы проводила исследования по рассматриваемой теме и получила ряд интересных результатов. В тот период исследователи работали и публиковались за рубежом, и эти результаты мало знакомы специалистам в России. Кроме того, данная группа известна по исследованиям асинхронных и, в частности, самосинхронных схем, не получивших должного развития в современных технологиях проектирования цифровой аппаратуры и недостаточно поддержанных инструментальными средствами автоматизации проектирования.

Цель статьи — рассмотреть практические задачи разработки средств координации событий в глобально асинхронных системах, подсистемы которых чаще всего являются синхронными, хотя могут быть и асинхронными, с использованием результатов, полученных специалистами в «асинхронике». А также в том, чтобы обратить внимание отечественных специалистов на публикации группы В. И. Варшавского, которые приведены в списке литературы.

Классификация вычислительных систем по способам синхронизации работы подсистем

С использованием теории автоматов любую вычислительную систему можно представить как сеть конечных автоматов. Функционирование конечного автомата как математической модели некоторого устройства описывается в шагах алгоритма, и сама модель не содержит понятия времени. С другой стороны, любое устройство функционирует в реальном физическом времени. Время вводится на этапе структурного синтеза автомата как физического устройства. Реальное физическое время квантуется, т. е. представляется в виде последовательности квантов (тактов). Такое квантованное физическое время обычно называют *дискретным временем*. На выполнение шага алгоритма автомата выделяется квант времени, или такт. Таким образом, последовательность шагов алгоритма выполняется за время прохождения соответствующего количества тактов. В зависимости от того, как непрерывное физическое время отображается на последовательность тактов в дискретном времени, автоматы как физические устройства делятся на два класса: синхронные и асинхронные.

В *синхронных автоматах* физическое непрерывное время делится на такты сигналами от генератора тактовых импульсов. Длительность такта постоянна и определяется частотой генератора (тактовой частотой). Генератор выполняет функцию часов. Для привязки события ко времени говорят о номере такта, когда произошло событие.

В *асинхронных автоматах* начало выполнения очередного шага алгоритма связано с изменением состояния входа (с событием на входе), и физическое время квантуется сигналами окончания переходных процессов в схеме автомата. Такие асинхронные автоматы называются *согласованными*, в отличие от традиционных асинхронных автоматов, в которых, как и в синхронных, применяется тактирование входных сигналов, рассчитанное на худший случай. В дальнейшем под асинхронным автоматом будем понимать только согласованный автомат. В согласованных автоматах длительность такта не является постоянной, что приводит к сложности определения реальных моментов физического времени выполнения шагов алгоритма, реализуемого автоматом.

При построении системы, содержащей много параллельно работающих блоков, возникает проблема организации требуемого взаимодействия блоков, т. е. координации их функционирования, или синхронизации. Когда говорят, что некоторые события происходят синхронно, то обычно

имеют в виду, что эти события являются следствиями одних и тех же причин. Такое понимание синхронности можно использовать как в логическом, так и в физическом времени. Когда говорят о *синхронной системе*, то обычно подразумевают, что все события в системе могут происходить в том или ином такте. События происходят синхронно, если они происходят в одном такте. Такое понимание синхронности используется специалистами по аппаратным средствам при построении синхронных систем.

С увеличением сложности вычислительных систем и ростом их быстродействия возрастает сложность синхронизации всех блоков от одного генератора (одних внешних часов). Это можно объяснить, по крайней мере, двумя причинами:

— события во внешних часах не имеют причинно-следственной связи с событиями в системе, что в результате приводит к нарушению общей причинной семантики поведения системы;

— система доставки сигналов от часов в различные точки системы из-за задержек в связях функционирует с ограниченной точностью. Она может правильно функционировать только в случае, когда ее точность соответствует требуемой точности синхронизации. Обеспечение этих требований за счет увеличения длительности такта в настоящее время существенно снижает быстродействие в сравнении с теми возможностями, которые предоставляет современная микроэлектроника.

Термин «асинхронность» тоже часто трактуется по-разному специалистами в области программного обеспечения и аппаратуры. Программисты обычно трактуют асинхронность как зависимость события от числа шагов алгоритма, необходимых для получения результатов из исходных данных. Такой подход оказывается полезным только для оценки эффективности программы. В параллельном программировании подразумевается, что шаги алгоритма могут выполняться одновременно. В этом случае при спецификации программы используются причинно-следственные отношения между операциями в виде их частичного упорядочения, что эквивалентно введению в алгоритм логического времени.

На уровне аппаратных средств течение логического времени обычно измеряется в единицах физического времени (числом тактов). Используя операцию *ждать* и хорошо известные способы и средства синхронизации (вместе с внешними часами), можно решить проблемы поведения параллельной системы во времени даже в случае, когда локальные процессы внутри системы являются асинхронными. Это используется, например, при организации обменов по системной шине в компьютере.

При отказе от общих часов мы приходим к системам, которые являются асинхронными в физическом времени. Существуют подходы к проектированию асинхронных аппаратных средств, которые не используют общих часов при координации событий. Однако все они, включая подход самосинхронизации, приводят к решениям более высокой сложности, чем синхронные средства. Они требуют специальных методов спецификации поведения аппаратуры, существенно ограничивают возможности использования развитых инструментальных средств автоматизации проектирования.

В работе [1] приведена классификация вычислительных систем по способу координации поведения блоков аппаратных средств во времени:

ПС — полностью синхронные (синхронизация от общих часов);

ПА — полностью асинхронные (например, самосинхронные);

ЛАГС — локально асинхронные, глобально синхронные;

ГАЛС — глобально асинхронные, локально синхронные;

ГАЛП — глобально асинхронные, локально произвольные.

Полностью синхронные и ЛАГС требуют системы доставки синхронизирующих сигналов от общих часов. Недостатки таких систем обсуждались выше.

Полностью асинхронные системы обладают рядом важных преимуществ, но они очень сложны и не соответствуют принятой парадигме эволюционного развития средств и методов проектирования аппаратуры, не позволяют использовать в проектах большинство из уже разработанных узлов, представленных в виде IP-модулей.

Наиболее привлекательными являются ГАЛС-системы, так как они объединяют преимущества как синхронного, так и асинхронного подходов.

Системы ГАЛП являются расширением ГАЛС-систем. В таких системах блоки структуры могут относиться как к классу синхронных (ПС), так и к классу асинхронных (ПА). При анализе их работы в физическом времени асинхронность трактуется как непредсказуемые вариации длительностей переходных процессов. В логическом времени асинхронность трактуется как вариации количества дискретных шагов в процессах. Физическое время может быть трансформировано в логическое время посредством введения общих часов или использования принципов самосинхронизации. Будем говорить о *глобальном* логическом времени, если функционирование структурных блоков системы координируется в логическом времени. *Локальное* логическое время протекает внутри блоков структуры системы.

Концепция времени и ее интерпретации при проектировании вычислительных систем

Применительно к рассматриваемым вопросам синхронизации процессов в системах необходимо уточнить, что понимается под словом «время».

Современная наука говорит, что время (как и материя) имеет двойственную природу. С одной стороны, время трактуется как число (Платон и Аристотель) или как момент, с другой стороны, — как отражение причинно-следственного отношения между событиями.

Известно высказывание Аристотеля: «Если ничего не происходит, то нет и времени». Традиция греков была продолжена Лейбницем, который трактовал время как отношение. С этой точки зрения, время определяется как частичный порядок на событиях, устанавливающий причинно-следственное отношение между ними. McGraw-Hill Encyclopedia of Science & Technology определяет синхронизацию как «процесс поддержания одной операции в шагах другой». Здесь время рассматривается как дискретная величина.

Концепция аналогового времени связана с именем Исаака Ньютона, который продолжил и развил античную индоевропейскую трактовку времени как непрерывной величины, называя ее физическим временем. Заметим, что все существующие определения физического времени в той или иной степени связаны с процедурой измерения времени, которая может быть выполнена только с определенной степенью точности. Процедура измерения обеспечивает связь между физическим и логическим временем. Основным измерительным прибором при этом являются часы; в процессе измерения они сравнивают временной интервал между двумя дискретными событиями с числом также дискретных событий, с помощью которых осуществляется измерение. Одновременность двух событий, строго говоря, может трактоваться только как логическая абстракция, достигаемая также с определенной степенью точности.¹

¹ С некоторой степенью спекуляции можно сказать, что минимальный интервал при временной дискретизации определяется выражением для неопределенности в форме времени $\tau = \hbar/\Delta E$, где τ — среднее время перехода от одного квантового состояния к другому с разницей энергетических уровней ΔE ; \hbar — постоянная Планка. Если $\Delta E = 1 \text{ eV} \approx 600 \text{ eV}$ (характеристика энергии для переключения продвинутых микроэлектронных устройств), то $\tau \approx 10^{-3} \text{ пс}$. Практически то же значение ($\tau \approx 2 \cdot 10^{-3} \text{ пс}$) дается для среднего времени переключения квантового устройства, когда испускается один фотон с $\lambda \approx 900 \text{ нм}$ (нормальная частота для оптоволокна). Эти значения можно рассматривать как приблизительную нижнюю границу для квантования времени.

Все шаги проектирования, начиная с формулирования исходной спецификации и кончая физическим проектированием, касаются в том числе и организации временного поведения проектируемых устройств. Таким образом, проектирование системы синхронизации можно интерпретировать как реализацию системного времени.

Несмотря на то что все устройства являются физическими объектами, функционирующими в реальном физическом времени, специалисты в области проектирования систем на разных этапах проектирования имеют дело с различными типами времени и по-разному трактуют системное время, так же как и связанные с ними термины синхронный и асинхронный.

Специалисты в области программирования и архитектуры имеют дело с логическим временем. Они трактуют время как число шагов в процессе с дискретными состояниями (число переходов из одного дискретного состояния в другое). Асинхронность трактуется как вариации (например, в зависимости от данных) числа шагов в процессе (алгоритме) от его инициализации до получения результата. Например, деление без восстановления остатка является синхронным процессом, в то время как деление с восстановлением остатка — асинхронным процессом.

Специалисты в области микроэлектроники и вычислительной техники имеют дело с физическими процессами, которые протекают в физических устройствах, и, следовательно, с аналоговым физическим временем. Для них асинхронность связана с неуправляемыми вариациями времени распространения сигналов по сети элементов и перехода из одного дискретного состояния в другое. Некоторые факторы таких вариаций могут быть постоянными (например, разброс технологических параметров), некоторые — переменными (такие как изменения условий функционирования: температуры, питающих напряжений и др.), а также зависящими от данных (например, время завершения переносов в сумматоре).

Проблемы синхронизации можно рассматривать как проблемы реализации интерфейса между физическим и логическим временем. Простейшим и общепринятым способом обеспечения такого интерфейса в синхронных системах является использование системных часов (общего генератора тактовых импульсов). Интервал физического времени между двумя тактовыми импульсами (длительность такта) должен быть достаточно большим, чтобы маскировать все возможные вариации длительностей переходных процессов. При этом можно полностью исключить из рассмотрения физическое время и рассматривать работу только в логическом времени. Это является

основным преимуществом синхронного подхода, использующего общие часы.

Несмотря на все преимущества, синхронный подход имеет недостатки, которые становятся все более ощутимыми по мере совершенствования технологии производства интегральных схем в направлении повышения степени интеграции и производительности. Наибольшие трудности связаны с проектированием системы доставки синхросигналов в точки их потребления.

Альтернативой синхронному подходу являются полностью асинхронные модели (им соответствуют несколько направлений в схемотехнике: самосинхронные схемы; схемы, не зависящие от скорости элементов; схемы, не чувствительные к задержкам). В них системное время определяется частичным порядком, накладываемым на дискретные события. Время вводится с помощью спецификации причинно-следственных отношений между событиями; интерфейс между физическим и логическим временем обеспечивается определенными механизмами фиксации моментов окончания переходных процессов в системных компонентах.

Рассмотрим основные *методологические различия* между синхронным и асинхронным подходами.

В синхронном подходе механизм, обеспечивающий системное время, полностью отделен от модели поведения системы. В асинхронном подходе он встроен в модель поведения системы. Его следует разрабатывать вместе с исходной поведенческой спецификацией.

Асинхронное проектирование оказывается более сложным, чем синхронное. Основная причина этого заключается в том, что поведенческие алгоритмы обычно формулируются как синхронные. Перевод синхронных алгоритмов в асинхронные является одной из сложных задач, требующей большой изобретательности.

Как правило, асинхронные реализации превосходят синхронные по сложности в 2,5–4 раза и являются по этой причине малоэффективными, хотя имеют определенные положительные свойства. Поэтому в последние 10 лет бурно развивается ГАЛС-подход к проектированию аппаратных средств. Он сочетает в себе преимущества синхронного и асинхронного подходов и свободен от многих недостатков, присущих обоим.

Любую асинхронную систему можно представить в виде композиции двух подсистем (слоев) [2–5]: синхростратума (*stratum* — слой, ярус) и процессорного стратума. Такая декомпозиция позволяет абстрагироваться от поведенческих аспектов блоков системы и сосредоточить основное внимание на проблеме их глобальной синхронизации с помощью полностью асинхронного

синхростратума. Кроме того, такой подход (с выделением синхростратума) позволяет строить глобально асинхронные системы по их синхронным прототипам. Подход применим для ГАЛС- и ГАЛП-систем.

Проектирование синхростратумов

В представлении системы в виде композиции синхростратума и процессорного стратума синхростратум функционирует как распределенные асинхронные часы, с помощью которых осуществляется глобальная синхронизация блоков системы. Он может быть построен по аналогии с системой синхронизации от общих часов.

Отсутствие внешних часов требует асинхронного взаимодействия синхростратума с блоками системы. Это может быть организовано за счет использования запрос-ответного взаимодействия (хендшейка), которое требует от блоков процессорного стратума формирования сигналов о завершении в них переходных процессов. Такая задача может быть решена многими способами, зависящими от типа и структуры блока, его размеров, возможных вариаций длительностей переходных процессов и т. д. Например, сигнал завершения переходного процесса может быть сформирован с помощью задержки, включенной параллельно блоку, использованием самосинхронных блоков, уже имеющих такой сигнал, или применением старт-стопного генератора, запускаемого сигналом *запрос* и останавливаемого по переполнению счетчика числа тактов. При организации сложных систем возможно также использование синхростратумов второго и более высокого уровней иерархии.

В дальнейшем изложении структура и функции блоков процессорного стратума системы не рассматриваются. Взаимодействие блоков с синхростратумом рассматривается на уровне сигналов инициации и завершения переходных процессов. Основное внимание будет уделено способам трансформации системы синхронизации синхронного прототипа в спецификацию синхростратума, с помощью которого должна осуществляться глобальная синхронизация системных блоков.

Синхронный прототип

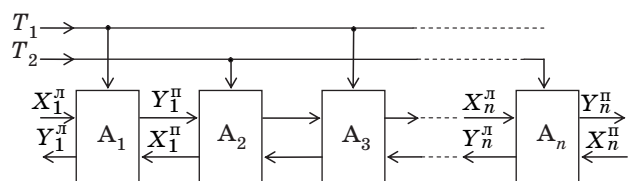
Основная идея синхронизации посредством тактовых импульсов T от внешних часов (генератора синхроимпульсов) связана с организацией взаимодействия блоков по принципу «ведущий—ведомый».

Для реализации памяти синхронных автоматов используются триггеры, синхронизируемые фронтом (либо спадом) синхроимпульса [6]. В них

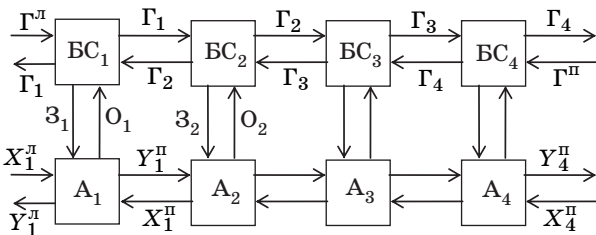
встроен механизм «ведущий—ведомый». В одном синхронном триггере используются два асинхронных триггера: основной и вспомогательный. При одном значении сигнала T , соответствующем первой фазе работы синхронного триггера, происходит запись нового состояния в основной триггер, состояние вспомогательного при этом не изменяется и соответствует старому значению состояния автомата. При другом значении сигнала T (смене значения T и соответствует фронт либо спад) новое состояние из основного триггера копируется во вспомогательный. В этой фазе состояние основного триггера не меняется, несмотря на изменения внешних сигналов. Такая организация памяти позволяет исключать критические состязания в автомате, связанные с разбросом задержек срабатывания отдельных триггеров.

Рассмотрим в качестве синхронного прототипа одномерный массив линейно связанных синхронных автоматов. Массивы автоматов синхронизируют с помощью различных систем синхронизации в зависимости от способов передачи сигналов между автоматами и логического базиса реализации автоматов. В простейшем случае используется однопроводная синхронизация. При этом механизм реализации двухфазной работы, соответствующей принципу «ведущий—ведомый», встроен в каждый синхронный триггер, как уже было отмечено. Поскольку наша задача — рассматривать организацию синхростратума, также координирующего взаимодействие автоматов по принципу «ведущий—ведомый», то будем рассматривать в качестве синхронного прототипа массив синхронных автоматов с двухпроводной двухфазной синхронизацией (рис. 1).

В нем механизм «ведущий—ведомый» вынесен в двухпроводную систему синхронизации, что удобно для дальнейшего рассмотрения синхростратума. По характеру информационных связей между автоматами массив представляет собой цепь. Каждый автомат A_i связан с соседом слева A_{i-1} и с соседом справа A_{i+1} . У каждого автомата A_i имеются два входных вектора — слева X_i^I и справа X_i^II , и два выходных вектора Y_i^I и Y_i^II . В таком массиве информационные потоки могут передаваться в обоих направлениях и процессы могут быть иницированы через свободные порты обоих оконечных автоматов, не вызы-



■ Рис. 1. Одномерный массив синхронных автоматов с двухфазной синхронизацией



■ Рис. 2. Одномерный массив автоматов с асинхронным синхростратумом

вая нежелательных коллизий. Другими словами, может быть реализован любой волновой алгоритм, так как синхросигналы T_1 и T_2 обеспечивают правильное взаимодействие между соседними автоматами.

Пример одномерного массива взаимодействующих автоматов с асинхронным синхростратумом показан на рис. 2.

Структура соединений блоков синхростратума BC_1, \dots, BC_4 изоморфна структуре соединений автоматов в массиве. Блоки синхростратума BC_i взаимодействуют с соответствующими автоматами по принципу «запрос — ответ». При этом выходной сигнал «запрос» Z_i блока BC_i синхростратума является сигналом локальной синхронизации автомата, инициирующим начало его работы. После завершения процесса в инициированном автомате A_i он формирует ответный сигнал O_i . При этом соответствующий блок BC_i посылает своим соседям сигналы готовности Γ_i . При такой организации синхростратума каждый автомат получает сигнал локальной синхронизации лишь при условии завершения переходных процессов в автоматах, являющихся его соседями по графу связей между автоматами. В этом заключается отличие от синхронного прототипа, в котором условием срабатывания каждого автомата является завершение переходных процессов во всех автоматах, которые были инициированы предыдущим синхросигналом.

Анализ взаимодействия автоматов в синхронном прототипе

Структура на рис. 1 предполагает, что автоматы взаимодействуют по принципу «ведущий — ведомый». (При этом в каждом автомате, как отмечалось выше, также может использоваться этот принцип, механизм реализации которого встроен в триггеры памяти). Для организации взаимодействия автоматов используются две последовательности синхросигналов T_1 и T_2 . Пусть эти сигналы изменяют свои значения следующим образом:

$$(T_1, T_2) = (0, 0) \rightarrow (1, 0) \rightarrow (0, 0) \rightarrow (0, 1) \rightarrow (0, 0) \rightarrow \dots \quad (*)$$

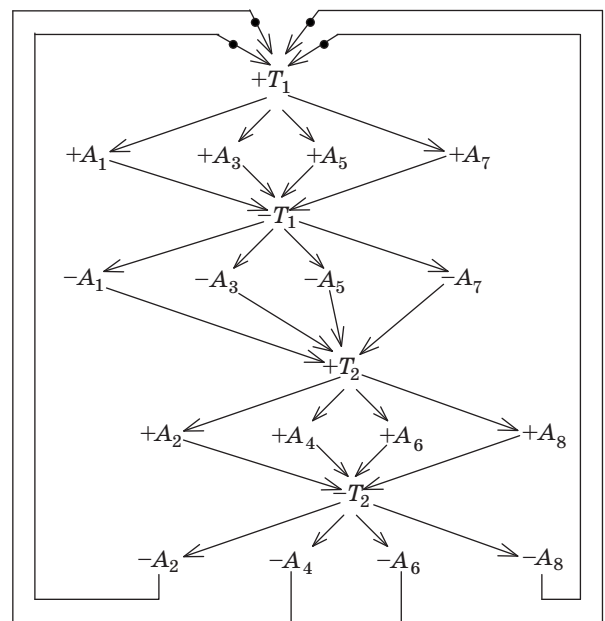
При $(T_1, T_2) = (0, 1)$ нечетные автоматы передают информацию своим соседям — четным автоматам, которые выполняют переход в новое состояние. Таким образом, нечетные автоматы являются ведущими, а четные — ведомыми. При $(T_1, T_2) = (1, 0)$ нечетные автоматы получают информацию от своих соседей — четных автоматов и выполняют переход, т. е. в этом такте нечетные автоматы являются ведомыми, а четные — ведущими.

Рассмотрим множество событий, связанных с приведенным описанием работы автоматов на рис. 1, и дискретизацию физического времени, а также связь дискретного физического времени с логическим. Такими событиями являются $\pm T_1$ и $\pm T_2$ — переключения сигналов синхронизации T_1 и T_2 ; события $+A_i$ и $-A_i$ имеют смысл начала и завершения переходных процессов в автоматах A_i .

Сигнальный граф параллельной двухфазной синхронизации в дискретном времени одномерного массива (см. рис. 1) из восьми автоматов с дисциплиной синхронизации (*) изображен на рис. 3. Граф определяет частичный порядок на множестве событий.

Синхронизация происходит в узловых вершинах графа: вершины $-T_1, +T_2$ ожидают сигналов окончания переходных процессов в нечетных автоматах, а $-T_2, +T_1$ — от четных автоматов.

Заметим, что в рассмотренной структуре синхронизации каждый шаг k в логическом времени, соответствующий одному шагу выполняемого автоматами алгоритма, занимает два такта: T_1 и T_2 . Таким образом, приведенный на рис. 3 граф



■ Рис. 3. Сигнальный граф параллельной двухфазной синхронизации одномерного массива из 8 автоматов

соответствует одному шагу алгоритма (логического времени), раскрывая его во введенных с помощью событий шагах дискретного времени.

Проектирование асинхронного синхростратума для одномерного массива автоматов

С использованием приведенного анализа организации взаимодействия в синхронном прототипе (см. рис. 1) рассмотрим задачу проектирования асинхронного синхростратума (см. рис. 2) для того же процессорного стратума. В общем случае длительность переходных процессов в параллельно работающих автоматах массива может быть разной и не известна в физическом времени. Поэтому их взаимодействие с синхростратумом должно быть организовано по принципу «запрос—ответ» (хендшейка). При этом не имеет значения, какой принцип лежит в основе построения автоматов: свой генератор синхроимпульсов, самосинхронная схема, встроенные задержки и др. Важно только то, что автоматы формируют сигналы O_i об окончании в них переходных процессов в ответ на получение ими локальных сигналов инициации работы Z_i от синхростратума (см. рис. 2).

Возможны два подхода к организации параллельной работы автоматов (как и обработки данных в вычислительной системе): многоэлементная (параллельная) и конвейерная работа. При многоэлементной работе каждый автомат выполняет свою работу от начала до конца параллельно с другим. При конвейерной (многостадийной) работе процесс разбивается на несколько стадий. Автомат выполняет одну из стадий обработки. Завершив свою работу, автомат передает результаты соседнему автомату (например, правому соседу), а затем опять начинает выполнять свою стадию работы с новыми входными данными, полученными от левого соседа. Конвейерная работа автоматов очень похожа на процесс распространения волны, поэтому иногда будем называть процесс организации такой работы волновой синхронизацией. В приведенной на рис. 2 системе волна может распространяться как слева направо, так и справа налево.

Применительно к задаче проектирования синхростратума рассмотрим сначала *организацию конвейерной работы* (волновую синхронизацию).

На вход конвейера поступает последовательность порций обрабатываемых данных. Размер порции (слово, полуслово, двойное слово и пр.) определяется особенностями процессорного стратума и для организации взаимодействия автоматов не имеет значения.

Шагом алгоритма (шагом k логического времени) будем считать полный цикл обработки пор-

ции данных в конвейере. Для конкретизации дальнейшего рассмотрения примем, что массив на рис. 2 содержит 8 автоматов.

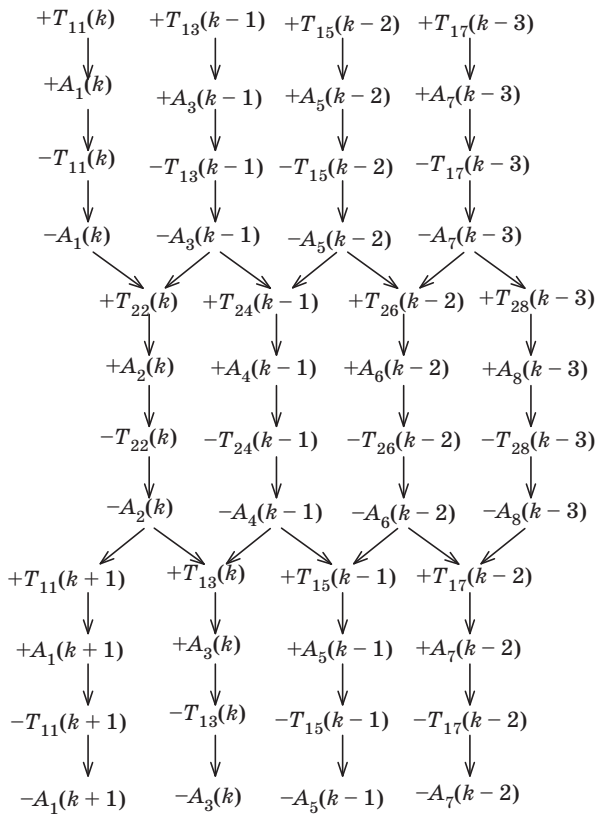
Напомним, что для организации бесконфликтной работы массива автоматов используется принцип «ведущий—ведомый». По аналогии с рис. 1 и 3 будем различать работу автоматов на рис. 2 в нечетном T_1 и четном T_2 тактах. Но теперь эти такты не связаны с внешним генератором. Можно говорить только об аналогии с двухфазной синхронизацией. Для первого автомата A_1 (см. рис. 2) дискретизация непрерывного физического времени связана со следующими событиями: поступление от БС₁ запроса Z_1 , начало работы автомата $+A_1$, завершение работы автомата $-A_1$, формирование ответа O_1 .

Поскольку инициация работы автоматов не связана с внешними часами и может происходить в различные моменты физического времени, для четных и нечетных тактов введем двойную индексацию: T_{11}, T_{13}, \dots — нечетные такты в 1-м, 3-м автомате; T_{22}, T_{24}, \dots — четные такты во 2-м, 4-м автоматах. При этом событие $+T_{11}$ соответствует $+Z_1$, $-T_{11}$ соответствует $-Z_1$, $-A_1$ соответствует O_1 .

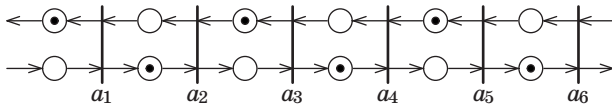
С использованием принятых обозначений на рис. 4 показан фрагмент развертки сигнального графа для структуры, представленной на рис. 2 и содержащей 8 автоматов. В этом графе $\pm T_{ij}(k)$ означает событие, заключающееся в k -м переходе синхросигнала T_{ij} , $i \in \{1, 2\}$, $j \in \{1, n\}$ (шаг k логического времени), которое наступает в j -м автомате массива. Дискретным моментам физического времени соответствуют строки, моменты логического времени указаны в скобках.

Анализируя рис. 4, легко увидеть, что один и тот же момент логического времени существует в различных автоматах массива в различные моменты физического времени и в один и тот же момент физического времени представлены различные моменты логического времени. Рис. 4 представляет общее отношение между сигналами запросов и ответов и может служить спецификацией для проектирования синхростратума. В общем случае реализация синхростратума требует введения в спецификацию дополнительных переменных и специального проектирования. Одно из возможных решений выглядит следующим образом.

Рассмотрим поток событий, связанных с взаимодействием автоматов в массиве рис. 2 при конвейерной работе. Пусть a_j — событие, соответствующее полному циклу синхронизации j -го автомата, т. е. $a_j = +T_{ij} \rightarrow +A_j \rightarrow -T_{ij} \rightarrow -A_j$. Тогда, следуя графу на рис. 4, необходимую координацию событий в потоке опишем маркированной сетью Петри (рис. 5). Рис. 5 — это спецификация простого конвейера.



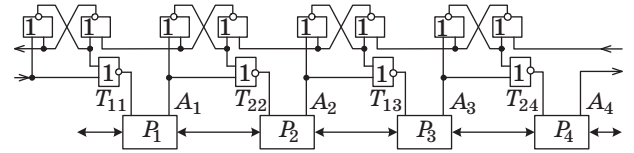
■ Рис. 4. Фрагмент развертки сигнального графа, описывающего волновую логическую синхронизацию в одномерном массиве



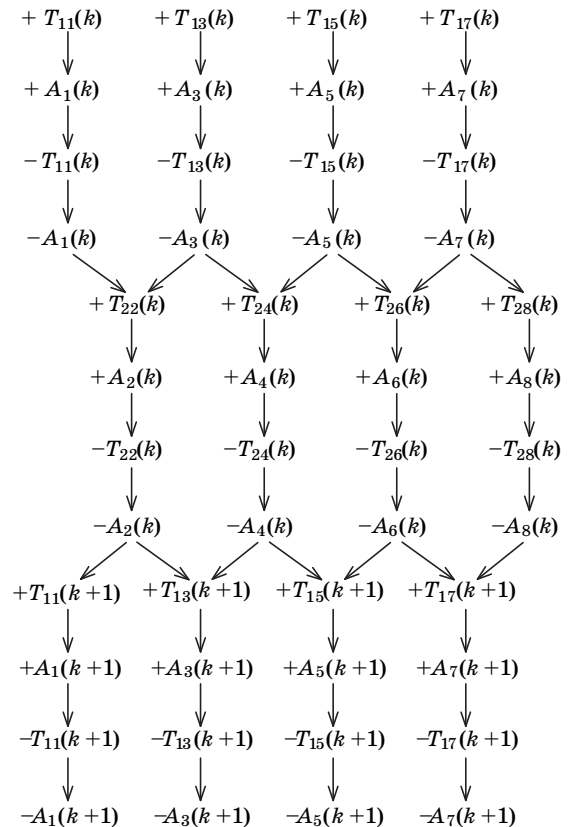
■ Рис. 5. Маркированная сеть Петри, описывающая конвейерное взаимодействие автоматов в одномерном массиве

В работах [7–9] рассмотрена методика перехода от спецификации к реализации на простых распределительных ячейках. Такой переход возможен по той причине, что в конвейере на распределительных ячейках выходной сигнал каждой j -й ячейки совершает полный цикл события a_j и только после этого начинает работать следующая $(j + 1)$ -я ячейка. В результате для рассматриваемого примера получаем достаточно простую схему синхростратума (рис. 6), где сигнал T_{11} соответствует сигналу Z_1 на рис. 2, а сигнал A_1 — сигналу O_1 .

Теперь рассмотрим **организацию параллельной работы** автоматов массива (см. рис. 2) при многоэлементной обработке данных. На рис. 7 показан фрагмент развертки сигнального графа, который описывает параллельную синхронизацию, в отличие от ранее рассмотренного подхода, основанного на распространении волн.



■ Рис. 6. Синхростратум на распределительных ячейках



■ Рис. 7. Фрагмент развертки сигнального графа для параллельной синхронизации одномерного массива

Этот фрагмент отличается от фрагмента на рис. 4 тем, что все параллельные синхросигналы (сигналы T_{11}, T_{13}, \dots) в одной строке (в одинаковый момент физического времени) соответствуют одному и тому же моменту логического времени. Даже беглого взгляда на рис. 4 и 7 достаточно, чтобы сделать заключение, которое кажется тривиальным и в то же время неожиданным.

Эти графы отличаются только маркировкой моментов логического времени (в скобках). Следовательно, одна и та же схема синхростратума в зависимости от принятого соглашения о маркировке моментов логического времени (т. е. в зависимости от функций перехода и выхода автоматов) может обеспечить либо параллельную, либо волновую систему глобальной синхронизации.

При синтезе схемы синхростратума по сигнальному графу, представляющему исходную

спецификацию, этот граф, согласно разработанной методике [7–9], должен быть приведен к корректному виду посредством введения в него дополнительных переменных¹. С их помощью устраняются конфликты кодирования состояний схемы. Было найдено удивительно простое решение (рис. 8).

В этом графе сигналы $\pm X_i$ и $\pm Y_i$ являются дополнительными переменными. Полученный сигнальный граф реализуется очень простой схемой (рис. 9).

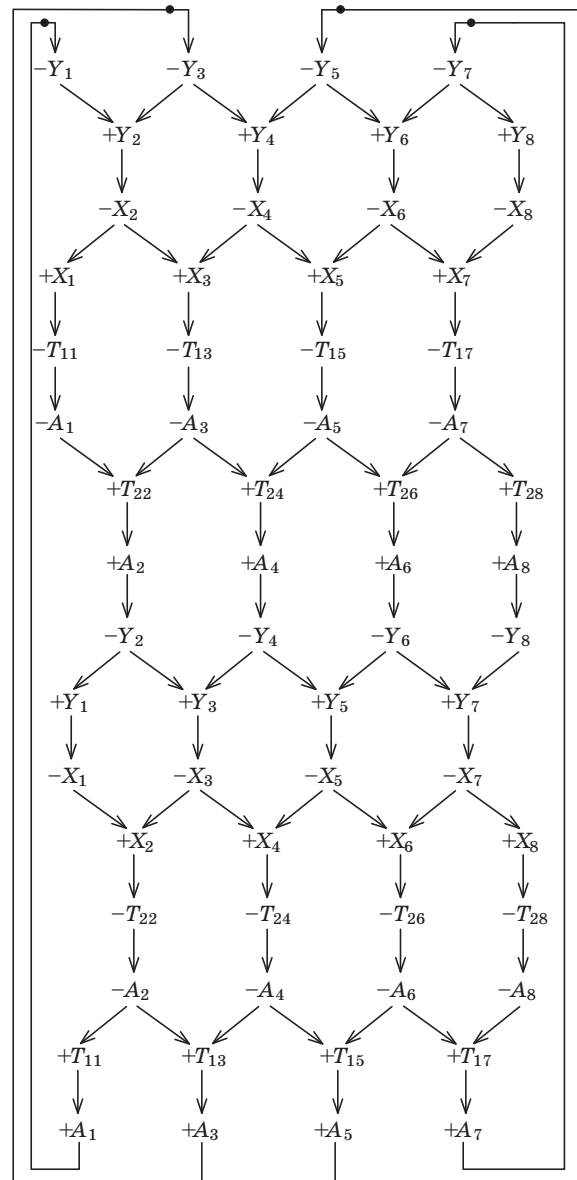
Теперь рассмотрим синхростратум для системы из автоматов, помещенных в вершины произвольного графа. Дуги графа соответствуют связям между автоматами. Согласно принятой дисциплине синхронизации (с двумя синхронизирующими системами T_{1j} и T_{2j}), граф связи должен быть двудольным (графом Кёнига) [10].

Сведение произвольного графа связей к графу Кёнига тривиально; например, это можно сделать вставкой буферных регистров во все связи. Заметим, что это вообще не должно влиять на общую стратегию синхронизации, так как сама стратегия синхронизации, как упоминалось выше, инвариантна семантике поведения синхронизированного автомата. Вставка буферных регистров может быть полезной независимо от типа графа соединений автоматов, особенно если цикл работы автомата значительно длиннее цикла записи в буферные регистры. При таком подходе сигналы $T_{1j} = 1$ инициируют активность автоматов, а сигналы $T_{2j} = 1$ инициируют циклы записи в буферные регистры.

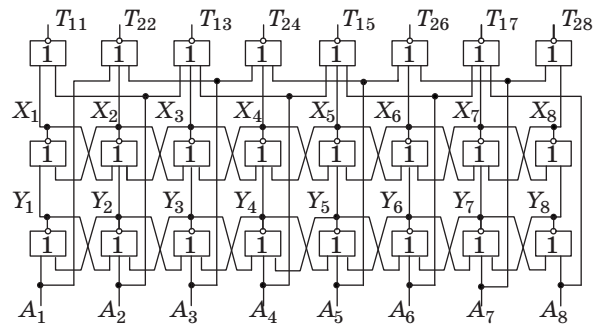
Для достижения большей степени параллелизма могут быть использованы автоматы с более сложной внутренней организацией. Однако эти проблемы связаны со стратегией формирования сигналов окончания переходных процессов в автоматах и методами внутри автоматной синхронизации, что выходит за рамки исследуемой проблемы.

Еще раз подчеркнем, что рассматриваемый здесь подход, связанный с декомпозицией системы на синхростратум и процессорный (автоматный) стратум, позволяет разделить проблемы глобальной синхронизации и алгоритмического поведения автоматов.

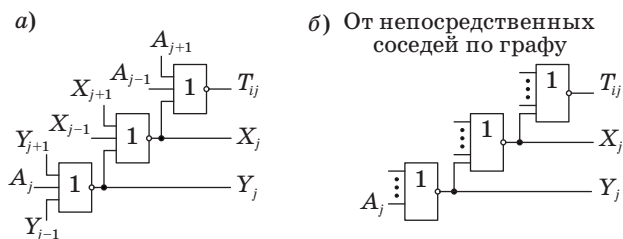
¹ Процедура приведения спецификаций асинхронных схем к корректному виду является сложной научной проблемой. Разработанные алгоритмы реализованы во многих системах проектирования асинхронных схем, например в системе **Petrify**. От способа введения в сигнальный граф дополнительных переменных, необходимых для устранения конфликтов состояний в схеме, зависит сложность реализации. Рассматриваемые здесь схемы синхростратумов построены ручным способом непосредственно по их сигнальным графам, что требует определенного навыка проектирования.



■ Рис. 8. Правильный сигнальный граф для параллельной синхронизации одномерного массива



■ Рис. 9. Синхростратум для параллельной синхронизации одномерного массива из 8 автоматов



■ Рис. 10. Схема элемента синхростратума: а — для одномерного массива; б — для произвольного графа (Кёнига) сети автоматов

Возвратимся к рис. 9 и рассмотрим локальные свойства схемы синхростратума, гарантирующие правильность ее поведения.

Во-первых, в слое вентилях, с выходов которых снимаются сигналы T_{1j} и T_{2j} , связи между соседними вентилями (через автоматы A_j) обеспечивают переключение вентиля из состояния 0 в состояние 1, если и только если выходные сигналы всех соседних вентилях равны 0.

Во-вторых, переключение вентиля с выходом T_{ij} из состояния 0 в состояние 1 должно быть детерминированным. Следовательно, необходима память для сохранения его предыдущего состояния; такая память организована на двух слоях вентилях с выходами X_j и Y_j . Каждый из этих выходов может переключиться в состояние 1 не раньше, чем два его боковых соседа переключат-

ся в 0. Поэтому связи вентилях X_j и Y_j со своими боковыми соседями аналогичны связям между вентилями T_{ij} .

Схема элемента синхростратума (см. рис. 9) изображена на рис. 10, а. Соблюдая требования к локальному поведению вентилях, с учетом числа соседей можно построить, по аналогии с этой схемой, схему элемента синхростратума для произвольного графа Кёнига (рис. 10, б).

Заключение

Представленный результат по глобальной синхронизации массивов многоблочных (многопроцессорных) систем соответствует случаю существования синхронного прототипа с двухфазной синхронизацией, использующей принцип «ведущий—ведомый».

Показано, что для любой глобально синхронной системы можно спроектировать систему асинхронной глобальной синхронизации (координации). Система может иметь любую топологию связей. Для этого достаточно снабдить блоки системы сигналами окончания в них переходных процессов и построить синхростратум из модулей, например, типа модулей рис. 10.

Интересные результаты могут быть получены при использовании в качестве синхронного прототипа не двухфазной, а многофазной синхронизации. Эта тема требует отдельного рассмотрения.

Литература

1. Varshavsky V., Marakhovsky V. GALA (Globally Asynchronous — Locally Arbitrary) Design // Concurrency and Hardware Design, Advances in Petri Nets: LNCS 2549 / Eds. J. Cortadella, A. Yakovlev, G. Rozenberg. Springer, 2002. P. 61–107.
2. Угрюмов Е. П. Цифровая схемотехника. СПб.: — БХВ-Петербург, 2007. — 800 с.
3. Varshavsky V. I., Chu T.-A. Self-Timing — Tools for Hardware Support of Parallel, Concurrent and Event-Driven Process Control: Proc. of the Conf. on Massively Parallel Computing Systems (MPCS). May 1994. P. 510–515.
4. Varshavsky V. I., Marakhovsky V. B., Chu T.-A. Logical Timing (Global Synchronization of Asynchronous Arrays) // Parallel Algorithm / Architecture Synthesis: Intern. Symp., Aizu-Wakamatsu, Japan. IEEE CS Press, Mar. 1995. P. 130–138.
5. Varshavsky V. I., Marakhovsky V. B., Chu T.-A. Asynchronous Timing of Arrays with Synchronous Prototype: Proc. of the Second Intern. Conf. on Massively Parallel Computing Systems (MPCS'96), Ischia, Italy, May 1996. P. 47–54.
6. Varshavsky V. I., Marakhovsky V. B. Global Synchronization of Asynchronous Arrays in Logical Time // Parallel Algorithm / Architecture Synthesis: Proc. of the Second Intern. Symp., Aizu-Wakamatsu, Japan. IEEE CS Press, Mar. 1997. P. 207–215.
7. Varshavsky V. Hardware Support of Parallel Asynchronous Processes. Helsinki University of Technology, Digital Systems Laboratory. Ser. A: Research Reports: Sept. 1987. N 2.
8. Варшавский В. И., Кишиневский М. А., Мараховский В. Б. и др. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под ред. В. И. Варшавского. — М.: Наука, 1986. — 308 с.
9. А. с. 1064461 СССР. Асинхронный распределитель / Варшавский В. И., Кишиневский М. А., Мараховский В. Б. и др. (СССР). 1983.
10. König D. Theorie der Endlichen und Unendlichen Graphen. Leipzig: Akad. Verlag M. B. H., 1936, 258SS; N. Y.: Chelsea, 1950. Zbl, 15, 375.