

Самосинхронные схемы. Принципы построения и элементная база

Алексей БУМАГИН, к. т. н.
Алексей ГОНДАРЬ
Михаил КУЛЯС
Александр РУТКЕВИЧ
Владимир СТЕШЕНКО, к. т. н.
Мехди ТАЙЛЕ
Григорий ШИШКИН

В статье рассмотрены принципы построения самосинхронных схем, дана их классификация, показаны отличия строго самосинхронных и квазисамосинхронных подходов. Приведен базис логических элементов для построения строго самосинхронных схем на основе парафазного представления сигналов. Даны примеры реализации логических элементов и конвейерных схем в этом базисе.

Введение

До настоящего времени общепринятым являлся синхронный подход к проектированию схем. При таком подходе для синхронизации элементов схемы используется глобальное тактовое дерево. С переходом к технологическим нормам 45 нм и меньше разработчики синхронных схем столкнулись с такими проблемами, как распространение тактового сигнала, дальнейшее увеличение тактовой частоты, уменьшение размеров элементов, уменьшение энергопотребления, и рядом других [1]. В самосинхронных схемах большинство этих проблем решено или вовсе отсутствует. Такие схемы обладают более низким энергопотреблением, характеризуются низким уровнем собственных шумов и электромагнитного излучения, обладают лучшей защитой от взлома (методом анализа спектра потребления мощности) и более просты для повторного использования блоков. Как свидетельство постепенного перехода проектов к самосинхронному исполнению может служить прогноз ITRS дальнейшего развития самосинхронных цепей в проектах. Так, по данным ITRS процент использования самосинхронных цепей в проектах в 2012 году достигнет 20%, а к 2020 году может быть уже 40% [2].

Классификация и принципы построения самосинхронных схем

Самосинхронная схемотехника является одним из способов локального решения проблемы синхронизации работы схемы. В отличие от синхронного подхода к проектированию, где для синхронизации работы схемы используется глобальная цепь распространения тактового сигнала с высокими требованиями к расфазировке и дрожанию фронта, в самосинхронных схемах синхронизация работы осуществляется за счет непосредственного определения моментов окончания переходных процессов в отдельных блоках схемы и использования этих сигналов для синхронизации соседних блоков. Таким образом, скорость работы схемы определяется исключительно задержками распространения сигнала в схеме и является максимально возможной для текущих условий (температуры, напряжения питания), параметров полупроводниковой структуры и обрабатываемых данных. На рис. 1 показан пример самосинхронной схемы, состоящий из двух блоков. Каждый блок состоит из асинхронного регистра (Р), осуществляющего хранение данных на время обработки, и комбинационной функции (КФ), выполняющей обработку данных. К выходу комбинационной функции подключен индикатор (И), отслеживающий момент окончания переходного процесса.

В зависимости от реализации индикатора может быть как отдельным блоком, так и частью комбинационной функции. Работа схемы происходит следующим образом: на вход блока подаются данные, сопровождаемые сигналом «Запрос». Если блок в данный момент не активен и готов к приему новых данных, схема управления формирует сигнал «Старт», который осуществляет запись данных во входной регистр и активизирует работу комбинационной функции блока, кроме это-

го, формируется сигнал «Подтверждение», показывающий, что блок готов к приему новых данных. Через какое-то время, по завершению переходных процессов, индикаторный блок формирует сигнал «Готов», показывающий, что вычисления закончены и на выходе блока находятся верные данные. Если от следующего блока приходит сигнал «Подтверждение», система управления формирует сигнал «Запрос». Время обработки данных блоком зависит от многих факторов режима работы и конкретно от обрабатываемых данных.

По способу определения момента окончания переходного процесса самосинхронные схемы можно разделить на 2 класса [1]: строго самосинхронные схемы — схемы, в которых происходит непосредственное определение момента окончания переходного процесса, и квазисамосинхронные — в них используются косвенные методы определения, и непосредственно момент времени окончания переходного процесса никак не фиксируется.

Наиболее распространенным способом построения квазисамосинхронных схем является метод модели максимальной задержки. В этих схемах для формирования сигнала окончания переходного процесса используется модель комбинационной схемы, выполненная на линии задержки, соответствующая максимальному времени распространения

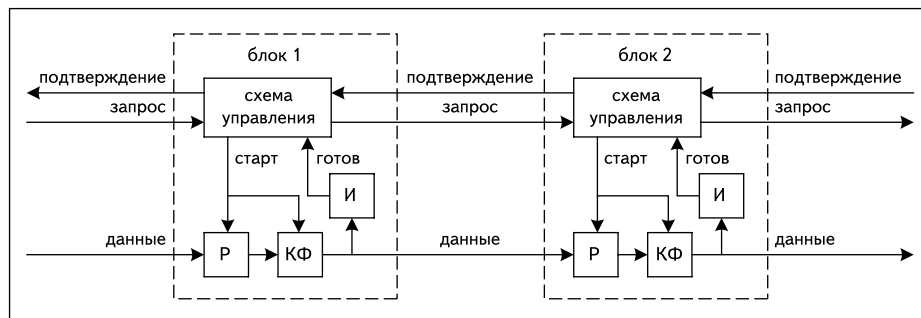


Рис. 1. Построение самосинхронной схемы

сигнала в комбинационной схеме с неким запасом, устраняющим технологический разброс и ошибки оценки времени распространения сигнала. Такой подход, в отличие от синхронной схемы, дает возможность работать схеме при максимальной скорости при данных внешних условиях и параметрах полупроводниковой структуры. Одним из недостатков такого решения являются жесткие требования к задержкам сигналов в схеме. При нарушении этих требований схема становится неработоспособной. Кроме этого, такой подход не позволяет учесть зависимость задержки в комбинационной схеме от значения сигнала на входе.

Пример квазисамосинхронной схемы показан на рис. 2. Данные между регистрами передаются с использованием специального протокола обмена и сопровождаются парой запрос/подтверждение (Req/Ack). На рис. 3 показано два варианта реализации протокола. Наиболее широко используется 4-фазный протокол, так как он требует меньше аппаратных ресурсов для своей реализации.

Для синхронизации работы схемы используется блок задержки, равной времени прохождения критического пути комбинационной схемы, и набор G-триггеров с дополнительными элементами НЕ.

G-триггер представляет собой переключающий элемент с гистерезисом, его переключение происходит, только когда на его входах присутствуют одинаковые логические уровни. Если уровни разные, то такой элемент сохраняет предыдущее состояние. Условное графическое изображение G-триггера и его таблица истинности показаны на рис. 4.

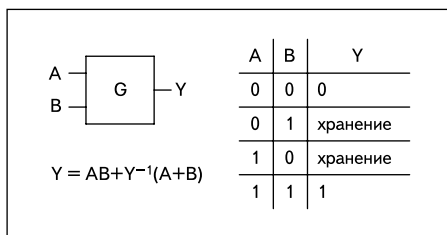


Рис. 4. G-триггер

В строго самосинхронных схемах применяется непосредственное определение момента окончания переходного процесса. Одним из способов построения таких схем является парафазное представление сигналов. Для передачи каждого бита информации

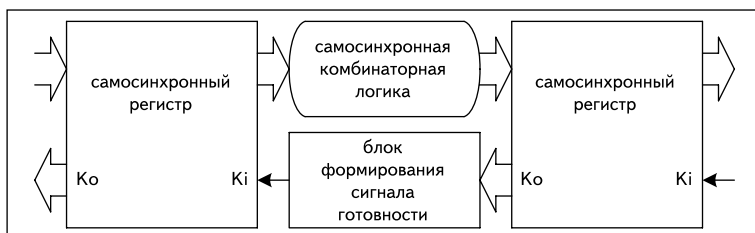


Рис. 5. Построение строгосамосинхронной схемы

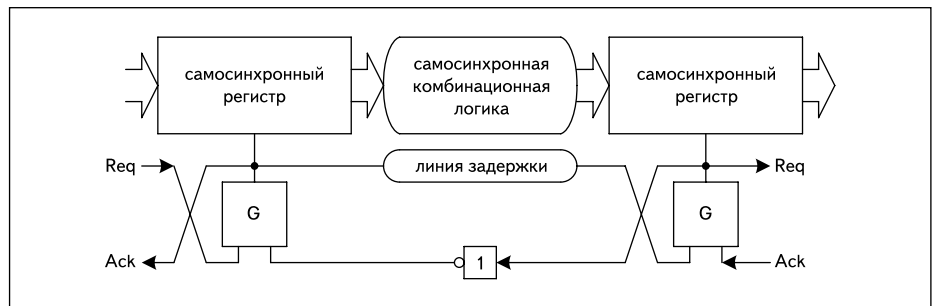


Рис. 2. Построение квазисамосинхронной схемы

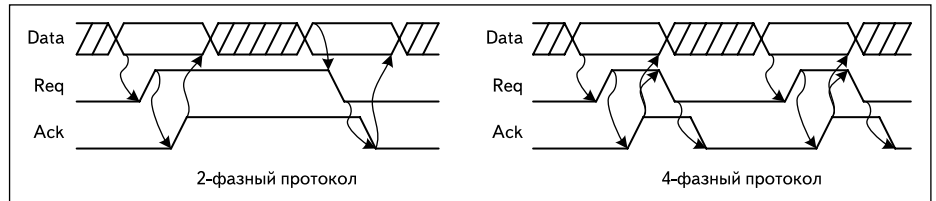


Рис. 3. Протоколы передачи данных

используется две физические линии. Это позволяет, кроме двух логических состояний, отдельно передавать разделитель, позволяющий отделять передаваемые данные во времени и определять с его помощью момент окончания переходного процесса. При проектировании таких схем к синтезу комбинационной функции предъявляются дополнительные требования на монотонность и возможность индицирования на выходе окончания всех переходных процессов внутри схемы. На рис. 5 показан пример такой схемы.

Примером реализации такого подхода является NCL-логика (NULL Convention Logic) разработанная специалистами компании Theseus Research, Inc. [4].

Класс строго самосинхронных схем имеет ряд преимуществ по сравнению с синхронными и квазисамосинхронными подходами. К ним относятся:

- Предельно возможная скорость работы схемы, так как схема сама определяет темп своей работы в зависимости от внешних условий и обрабатываемых данных.
- Независимость работоспособности схемы от задержек сигналов и логических элементов.
- Более широкий рабочий диапазон схемы, для работоспособности схемы требуется только сохранение переключаемых свойств транзисторов.

- Уменьшение энергопотребления за счет того, что при отсутствии данных в схеме не происходят переключения сигналов и остается только статическое потребление, в отличие от синхронной схемы, где без использования специальных мер, независимо от наличия данных тактовый сигнал подается на триггеры.

Базовые элементы самосинхронных схем

Основу NCL-схем составляют пороговые элементы $ТН_{mn}$, показанные на рис. 6.

Пороговый элемент $ТН_{mn}$ характеризуется двумя параметрами: количеством входов n и порогом срабатывания m , причем должно выполняться соотношение $(1 \leq m \leq n)$. Переключение выхода этого элемента в логическую «1» происходит при поступлении логических «1» на его входы m и сохраняется до тех пор, пока на всех его входах не будет логических «0». Расширением элемента $ТН_{mn}$ является пороговый элемент с весовыми коэффициентами для указанных входов $ТН_{m_1w_1m_2w_2m_3w_3m_4}$, где w_i — весовой коэффициент для входа i . На рис. 6 показан пороговый элемент $ТН_{32w_2}$ с весовым коэффициентом 2 для первого входа.

Пороговые элементы, у которых $m = n$, соответствуют G-триггерам с n - входами, а если

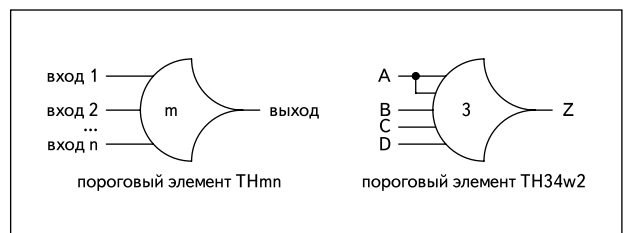


Рис. 6. Пороговые элементы

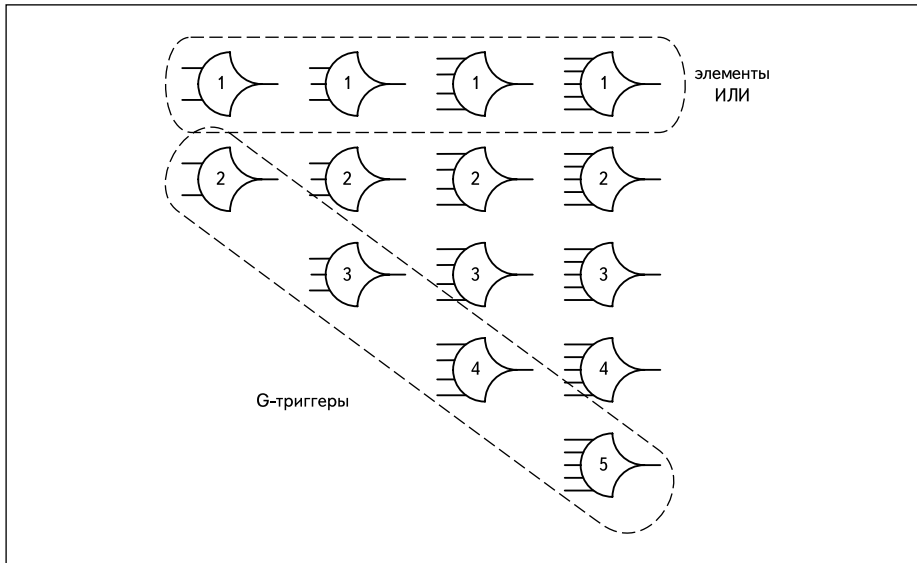


Рис. 7. Классификация пороговых элементов

$m = 1$ — элементу ИЛИ с n - входами (рис. 7). Остальные элементы являются расширением рассмотренных ранее G-триггеров и могут быть построены на их основе.

Состояние на выходе Y таких пороговых элементов определяется двумя функциями: *set* и *hold*. Функция *set* определяет условие выставления состояния логической «1» на выходе Y , а функция *hold* — условие удержания этого состояния на выходе. Общая функция переключения состояния на выходе для всех элементов $Y = set + Y^{-1} \times hold$, где Y^{-1} — это предыдущее состояние на выходе. Функция *hold* для всех элементов одинаковая и определяется как операция ИЛИ над всеми входами (например, для 4-входового элемента это $A+B+C+D$). Функция *set* определяет функ-

циональное назначение элемента. В таблице показан базовый набор из 27 таких элементов [5], который полностью определяет все

Таблица. Базовый набор пороговых элементов NCL

Элемент	Функция set
TH12	A+B
TH22	AB
TH13	A+B+C
TH23	AB+AC+BC
TH33	ABC
TH23w2	A+BC
TH33w2	AB+AC
TH14	A+B+C+D
TH24	AB+AC+AD+BC+BD+CD
TH34	ABC+ABD+ACD+BCD
TH44	ABCD
TH24w2	A+BC+BD+CD
TH34w2	AB+AC+AD+BCD
TH44w2	ABC+ABD+ACD
TH34w3	A+BCD
TH44w3	AB+AC+AD
TH24w22	A+B+CD
TH34w22	AB+AC+AD+BC+BD
TH44w22	AB+ACD+BCD
TH54w22	ABC+ABD
TH34w32	A+BC+BD
TH54w32	AB+ACD
TH44w322	AB+AC+AD+BC
TH54w322	AB+AC+BCD
THxor0	AB+CD
THand0	AD+BC+AD
TH24comp	AC+BC+AD+BD

возможные логические функции *set* для элементов с четырьмя и менее входами.

Для построения строго самосинхронной схемы используется парафазное представление сигналов, где каждый сигнал A передается по двум проводникам: $A.f$ и $A.t$. Таким образом, A может принимать три значения: «0» ($A.f = 1, A.t = 0$), «1» ($A.f = 0, A.t = 1$) и NULL ($A.f = 0, A.t = 0$). Состояние ($A.f = 1, A.t = 1$) не используется и говорит только об ошибке в работе схемы. Состояние NULL (разделитель или спейсер) необходимо для разделения отдельных данных во времени.

На рис. 8 показана реализация базовых логических элементов И, ИЛИ, НЕ и ИСКЛЮЧАЮЩЕЕ-ИЛИ в базе NCL.

На рис. 9 показан пример организации самосинхронного конвейера в NCL-базисе. Он состоит из двух регистров и комбинационной схемы. Каждый регистр состоит из двух частей — схемы запоминания состояния и индикатора, который показывает, что все переходные процессы закончены и правильные данные появились на выходе регистра. Синхронизация работы схемы осуществляется при помощи обратной связи и блока

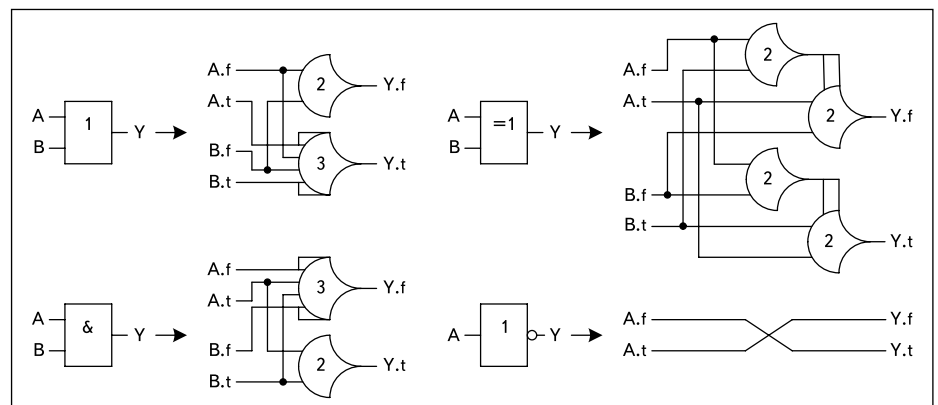


Рис. 8. Реализация логических функций в NCL-базисе

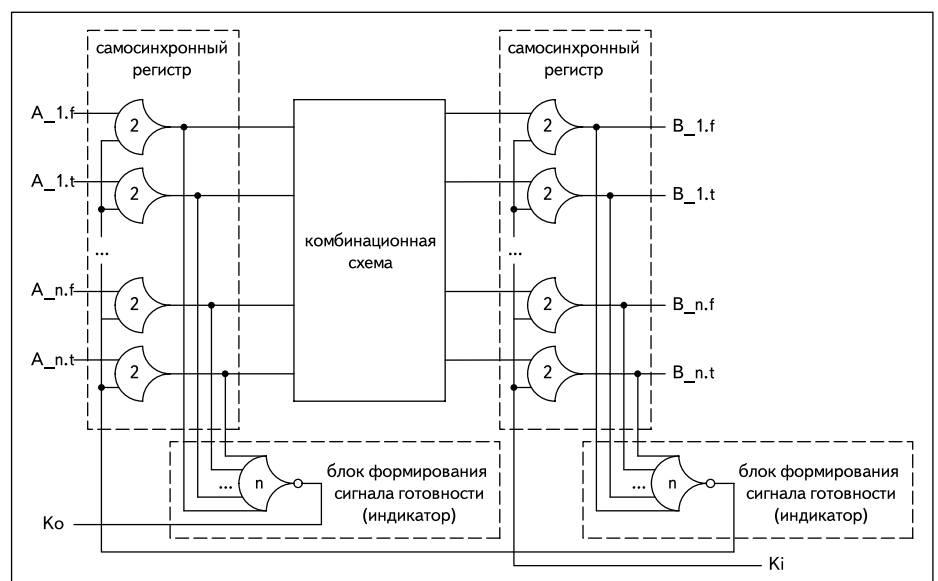


Рис. 9. Организация самосинхронного конвейера в NCL-базисе

формирования сигнала готовности, который подключен к каждому самосинхронному регистру в конвейере. Порядок прохождения данных в конвейере определяется сигналами запроса и подтверждения (K_i и K_o соответственно).

Протокол передачи данных в конвейере следующий: самосинхронные регистры принимают состояние DATA («0» или «1»), только когда приходит запрос *rfd* (request for data) на вход K_i ($K_i = 1$), и принимают состояние NULL только по запросу *frn* (request for null) на входе K_i ($K_i = 0$). Также генерируются и сигналы подтверждения K_o : когда на выходе регистра состояние DATA, на выходе K_o ($K_o = 1$) выставляется сигнал подтверждения *rfn* (ready for null), когда выходы регистра принимают состояние NULL, на выходе K_o ($K_o = 0$) выставляется сигнал подтверждения *rfd* (ready for data).

Заключение

Можно отметить широкие возможности и преимущества использования самосинхронных схем при построении различных портативных устройств и устройств, использующих алгоритмы шифрования данных, по сравнению с синхронными аналогами. Так, по данным компании Achronix [1], некоторые образцы выпускаемых ими микросхем (самосинхронные ПЛИС по технологии 90 нм с напряжением питания ядра 1,2 В) остаются работоспособным при снижении напряжения питания до 0,2 В. Кроме того, снижение напряжения

питания с 1,2 до 0,6 В приводит к 87%-ному уменьшению энергопотребления. Еще одним примером может служить самосинхронное процессорное ядро ARM996HS, разработанное фирмой Handshake Solution [1], имеющее в 2,8 раза меньшее энергопотребление по сравнению со своим синхронным аналогом ARM968E-S, при той же производительности. Такой процессор имеет в 2,4 раза меньшие токовые выбросы во время работы. ■

Литература

1. Taubin A., Cortadella J., Lavagno L., Kondratyev A., Peeters A. Design Automation of Real-Life Asynchronous Devices and Systems // Foundations and Trends(r) in Electronic Design Automation. Vol. 2, No. 1. September, 2007.
2. International Technology Roadmap for Semiconductors 2007. Design.
3. Smith S. C., DeMara R. F., Yuan J. S., Hagedorn M., Ferguson D. Delay-Insensitive Gate-Level Pipe-lining // Integration, the VLSI Journal. Vol. 30/2. October 2001.
4. Fant K. M., Logically Determined Design: Clockless System Design with NULL Convention Logic. Wiley, 2005.
5. Smith S. C., DeMara R. F., Yuan J. S., Ferguson D., Lamb D. Optimization of NULL Convention Self-Timed Circuits // Integration, The VLSI Journal. Vol. 37/3. August 2004.
6. Бумагин А., Гондарь А., Куляс М., Руткевич А., Стешенко В., Тайлеб М., Шишкин Г. Методы снижения энергопотребления в строго самосинхронных микропроцессорных схемах // Компоненты и технологии. 2009. № 9.