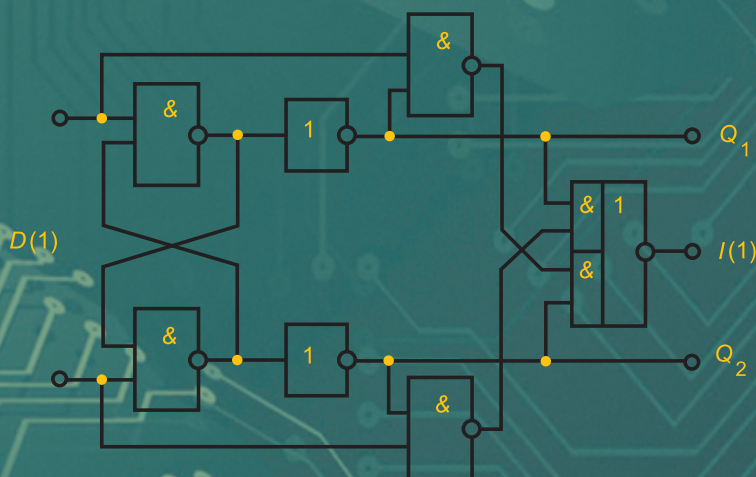


Л. П. Плеханов

# ОСНОВЫ САМОСИНХРОННЫХ ЭЛЕКТРОННЫХ СХЕМ



Л. П. Плеханов

# ОСНОВЫ САМОСИНХРОННЫХ ЭЛЕКТРОННЫХ СХЕМ

ЭЛЕКТРОННОЕ ИЗДАНИЕ



Москва  
БИНОМ. Лаборатория знаний  
2013

УДК 621.38  
ББК 32.844  
П38

*Электронный аналог печатного издания:* Основы самосинхронных электронных схем / Л. П. Плеханов. — М. : БИНОМ. Лаборатория знаний, 2013. — 208 с. : ил.

**Плеханов Л. П.**

П38 Основы самосинхронных электронных схем [Электронный ресурс] / Л. П. Плеханов. — Эл. изд. — М. : БИНОМ. Лаборатория знаний, 2013. — 208 с. : ил.

ISBN 978-5-9963-2191-9

В монографии подробно представлены начальные понятия, принципы работы, свойства, поведение и построение самосинхронных схем. Приведены примеры комбинационных и последовательностных схем, а также результаты экспериментальной проверки их уникальных свойств. Изложение ведется с позиций нового, функционального, подхода, основанного на исследовании логических функций, описывающих элементы схем, без привлечения методов теории автоматов. Аналогов данного подхода ни в отечественной, ни в зарубежной литературе не отмечено.

Для специалистов по дискретной электронике, аспирантов и студентов этого направления. Может также использоваться как для начального ознакомления, так и для учебного процесса.

**УДК 621.38  
ББК 32.844**

**По вопросам приобретения обращаться:  
«БИНОМ. Лаборатория знаний»  
Телефон: (499) 157-5272  
e-mail: binom@Lbz.ru, <http://www.Lbz.ru>**

ISBN 978-5-9963-2191-9

© БИНОМ. Лаборатория знаний,  
2013

# Интерактивное оглавление

Предисловие .....	6
<b>Глава 1. Введение в самосинхронику.....</b>	<b>12</b>
1.1. Проблемы работы и особенности цифровых схем.....	12
1.2. Самосинхронные схемы и их свойства.....	15
1.3. Экспериментальная проверка уникальных свойств СС-схем .....	18
1.4. Физический подход к достижению самосинхронности	22
1.5. Об используемой терминологии .....	23
<b>Глава 2. Теоретические основы самосинхронных схем .....</b>	<b>25</b>
2.1. Принципы построения и работы СС-схем.....	25
2.1.1. Индикация.....	26
2.1.2. Двухфазная дисциплина .....	29
2.1.3. Обратная связь .....	31
2.2. Самосинхронизирующиеся коды.....	33
2.3. Основа теории СС-схем — модель и теория Маллера .	41
2.3.1. Модель Маллера и независимость от задержек.	41
2.3.2. Применимость модели Маллера для элементов и их соединений .....	44
2.4. Базовые элементы СС-схем .....	45
2.4.1. Общая структура базовых элементов для СС- схем.....	46
2.4.2. Самосинхронные базовые элементы КМДП- технологии.....	48
2.5. Событийный и функциональный подходы в самосин- хронике .....	53
2.5.1. Типы сигналов и интерфейс СС-схем в функ- циональном подходе .....	54
2.5.2. Индицируемость — необходимое свойство разо- мкнутых СС-схем .....	56
2.5.3. Полная и частичная индицируемость.....	58
<b>Глава 3. Основы построения самосинхронных схем.....</b>	<b>59</b>
3.1. Подходы к созданию СС-схем .....	59
3.1.1. Методы событийного подхода.....	60
3.1.2. NCL-методология.....	61

3.1.3. Особенности разработки СС-схем в функциональном подходе .....	63
3.2. Способы индикации в СС-схемах .....	64
3.2.1. Глобальная индикация .....	65
3.2.2. Схемы редукции .....	65
3.3. Построение комбинационных СС-схем .....	72
3.3.1. Парафазное преобразование .....	73
3.3.2. Связь ПФС-преобразования и индикации .....	75
3.3.3. СС-секции .....	77
3.3.4. Общий порядок построения комбинационных СС-схем .....	79
3.3.5. Синтез СС-секций .....	81
3.4. Правила манипулирования сигналами в СС-схемах ..	99
3.4.1. Контрольные сигналы .....	100
3.4.2. ПФС-сигналы .....	100
3.4.3. Бистабильные сигналы .....	101
3.5. Создание последовательностных СС-схем .....	102
3.5.1. Использование бистабильных ячеек .....	102
3.5.2. Самосинхронные запоминающие ячейки .....	103
3.5.3. Самосинхронные автоматы .....	107
3.6. Самосинхронные конвейеры .....	110
3.7. Проблема задержек трасс после разветвлений .....	114
<b>Глава 4. Примеры самосинхронных схем .....</b>	<b>119</b>
4.1. Комбинационные схемы .....	120
4.1.1. Дешифратор .....	120
4.1.2. Мультиплексор .....	121
4.1.3. Демультимплексор .....	122
4.1.4. Схема равнозначности .....	123
4.1.5. Полусумматор .....	124
4.1.6. Сумматор одноразрядный .....	124
4.2. Схемы с памятью .....	125
4.2.1. Индикаторные триггеры .....	125
4.2.2. Информационные триггеры .....	129
4.2.3. Параллельные регистры .....	135
4.2.4. Последовательные регистры .....	135
4.2.5. Счетчики .....	140
4.3. Преобразование моносигналов в самосинхронные ..	142
<b>Глава 5. Анализ схем на самосинхронность .....</b>	<b>145</b>
5.1. Начальные и основные состояния .....	145
5.2. Дисциплина входных наборов .....	147

5.3.	Согласованное замыкание . . . . .	148
5.3.1.	Константное замыкание . . . . .	149
5.4.	Событийные методы анализа (замкнутых схем) . . . . .	151
5.4.1.	Прямой метод по диаграммам переходов (ДП) . . . . .	152
5.4.2.	Метод диаграмм изменений . . . . .	157
5.5.	Полнота анализа . . . . .	160
5.5.1.	Обеспечение полноты в событийном подходе . . . . .	162
5.6.	Функциональный метод анализа разомкнутых схем (ФМА) . . . . .	163
5.6.1.	Задачи анализа разомкнутых схем . . . . .	165
5.6.2.	Формирование входных значений, обеспечение полноты . . . . .	166
5.6.3.	Проверка индицируемости элементов . . . . .	167
5.6.4.	Определение параметров взаимодействия с окружением . . . . .	170
5.6.5.	Проверка состязаний . . . . .	171
5.6.6.	Анализ других логических неисправностей . . . . .	175
5.7.	Иерархический метод анализа (ИМА) . . . . .	176
5.7.1.	Задачи ИМА . . . . .	177
5.7.2.	Проверка правильности соединений фрагментов . . . . .	178
5.7.3.	Проверка индицируемости сигналов . . . . .	179
5.7.4.	Проверка соблюдения дисциплины БС-сигналов . . . . .	179
5.7.5.	Получение параметров интерфейса главной схемы . . . . .	185
<b>Глава 6.</b>	<b>Автоматизация проектирования сс-схем . . . . .</b>	<b>186</b>
6.1.	Система ФОРСАЖ (группа В. И. Варшавского) . . . . .	187
6.2.	Отдельные программы анализа (ИПИРАН) . . . . .	188
6.3.	САПР СС-схем промышленного назначения РОНИС (ИПИРАН) . . . . .	189
	Послесловие . . . . .	192
	Литература . . . . .	195
	Список сокращений . . . . .	200
	Словарь терминов . . . . .	201
	Приложение. Решение логических уравнений и систем. . . . .	207
П. 1.	Одно уравнение . . . . .	207
П. 2.	Система уравнений с одной неизвестной . . . . .	207
П. 3.	Система уравнений с несколькими неизвестными . . . . .	208

# ОГЛАВЛЕНИЕ

---

<b>Предисловие</b> . . . . .	6
<b>Глава 1. Введение в самосинхроннику</b> . . . . .	12
1.1. Проблемы работы и особенности цифровых схем . . . . .	12
1.2. Самосинхронные схемы и их свойства . . . . .	15
1.3. Экспериментальная проверка уникальных свойств СС-схем . . . . .	18
1.4. Физический подход к достижению самосинхронности . . . . .	22
1.5. Об используемой терминологии . . . . .	23
<b>Глава 2. Теоретические основы самосинхронных схем</b> . . . . .	25
2.1. Принципы построения и работы СС-схем . . . . .	25
2.1.1. Индикация . . . . .	26
2.1.2. Двухфазная дисциплина . . . . .	29
2.1.3. Обратная связь . . . . .	31
2.2. Самосинхронизирующиеся коды . . . . .	33
2.3. Основа теории СС-схем — модель и теория Маллера . . . . .	41
2.3.1. Модель Маллера и независимость от задержек . . . . .	41
2.3.2. Применимость модели Маллера для элементов и их соединений . . . . .	44
2.4. Базовые элементы СС-схем . . . . .	45
2.4.1. Общая структура базовых элементов для СС-схем . . . . .	46
2.4.2. Самосинхронные базовые элементы КМДП-технологии . . . . .	48
2.5. Событийный и функциональный подходы в самосинхронике . . . . .	53
2.5.1. Типы сигналов и интерфейс СС-схем в функциональном подходе . . . . .	54
2.5.2. Индицируемость — необходимое свойство разомкнутых СС-схем . . . . .	56
2.5.3. Полная и частичная индицируемость . . . . .	58
<b>Глава 3. Основы построения самосинхронных схем</b> . . . . .	59
3.1. Подходы к созданию СС-схем . . . . .	59
3.1.1. Методы событийного подхода . . . . .	60
3.1.2. NCL-методология . . . . .	61
3.1.3. Особенности разработки СС-схем в функциональном подходе . . . . .	63

3.2.	Способы индикации в СС-схемах . . . . .	64
3.2.1.	Глобальная индикация . . . . .	65
3.2.2.	Схемы редукции . . . . .	65
3.3.	Построение комбинационных СС-схем . . . . .	72
3.3.1.	Парафазное преобразование . . . . .	73
3.3.2.	Связь ПФС-преобразования и индикации . . . . .	75
3.3.3.	СС-секции . . . . .	77
3.3.4.	Общий порядок построения комбинационных СС-схем. . . . .	79
3.3.5.	Синтез СС-секций . . . . .	81
3.4.	Правила манипулирования сигналами в СС-схемах . . . . .	99
3.4.1.	Контрольные сигналы . . . . .	100
3.4.2.	ПФС-сигналы . . . . .	100
3.4.3.	Бистабильные сигналы. . . . .	101
3.5.	Создание последовательностных СС-схем. . . . .	102
3.5.1.	Использование бистабильных ячеек. . . . .	102
3.5.2.	Самосинхронные запоминающие ячейки. . . . .	103
3.5.3.	Самосинхронные автоматы . . . . .	107
3.6.	Самосинхронные конвейеры. . . . .	110
3.7.	Проблема задержек трасс после разветвлений. . . . .	114
<b>Глава 4.</b>	<b>Примеры самосинхронных схем . . . . .</b>	<b>119</b>
4.1.	Комбинационные схемы . . . . .	120
4.1.1.	Дешифратор . . . . .	120
4.1.2.	Мультиплексор . . . . .	121
4.1.3.	Демультиплексор . . . . .	122
4.1.4.	Схема равнозначности . . . . .	123
4.1.5.	Полусумматор . . . . .	124
4.1.6.	Сумматор одноразрядный . . . . .	124
4.2.	Схемы с памятью . . . . .	125
4.2.1.	Индикаторные триггеры. . . . .	125
4.2.2.	Информационные триггеры . . . . .	129
4.2.3.	Параллельные регистры . . . . .	135
4.2.4.	Последовательные регистры . . . . .	135
4.2.5.	Счетчики . . . . .	140
4.3.	Преобразование моносигналов в самосинхронные. . . . .	142
<b>Глава 5.</b>	<b>Анализ схем на самосинхронность . . . . .</b>	<b>145</b>
5.1.	Начальные и основные состояния. . . . .	145
5.2.	Дисциплина входных наборов . . . . .	147
5.3.	Согласованное замыкание . . . . .	148
5.3.1.	Константное замыкание . . . . .	149
5.4.	Событийные методы анализа (замкнутых схем). . . . .	151
5.4.1.	Прямой метод по диаграммам переходов (ДП) . . . . .	152
5.4.2.	Метод диаграмм изменений. . . . .	157



5.5. Полнота анализа . . . . .	160
5.5.1. Обеспечение полноты в событийном подходе . . . . .	162
5.6. Функциональный метод анализа разомкнутых схем (ФМА) . . . . .	163
5.6.1. Задачи анализа разомкнутых схем . . . . .	165
5.6.2. Формирование входных значений, обеспечение полноты . . . . .	166
5.6.3. Проверка индицируемости элементов. . . . .	167
5.6.4. Определение параметров взаимодействия с окружением . . . . .	170
5.6.5. Проверка состязаний. . . . .	171
5.6.6. Анализ других логических неисправностей. . . . .	175
5.7. Иерархический метод анализа (ИМА) . . . . .	176
5.7.1. Задачи ИМА . . . . .	177
5.7.2. Проверка правильности соединений фрагментов . . . . .	178
5.7.3. Проверка индицируемости сигналов . . . . .	179
5.7.4. Проверка соблюдения дисциплины БС-сигналов. . . . .	179
5.7.5. Получение параметров интерфейса главной схемы . . . . .	185
<b>Глава 6. Автоматизация проектирования СС-схем . . . . .</b>	<b>186</b>
6.1. Система ФОРСАЖ (группа В. И. Варшавского) . . . . .	187
6.2. Отдельные программы анализа (ИПИРАН) . . . . .	188
6.3. САПР СС-схем промышленного назначения РОНИС (ИПИРАН) . . . . .	189
<b>Послесловие . . . . .</b>	<b>192</b>
<b>Литература . . . . .</b>	<b>195</b>
<b>Список сокращений . . . . .</b>	<b>200</b>
<b>Словарь терминов . . . . .</b>	<b>201</b>
<b>Приложение. Решение логических уравнений и систем . . . . .</b>	<b>207</b>
П. 1. Одно уравнение . . . . .	207
П. 2. Система уравнений с одной неизвестной . . . . .	207
П. 3. Система уравнений с несколькими неизвестными. . . . .	208

## Предисловие

Дискретные электронные схемы отличаются огромным разнообразием, но все они могут быть разделены на два больших класса: *синхронные* и *асинхронные* (здесь и далее выделяются курсивом термины, поясняемые либо прямо в тексте, либо в глоссарии в конце книги).

Синхронные схемы появились и развивались первыми, продолжая господствовать в промышленности и в настоящее время. Асинхронные схемы имеют ряд преимуществ перед синхронными и получили большое развитие в последние 2—3 десятилетия. Их исследование и построение оформились в отдельное направление электроники — *асинхроннику*. Публикации в этом направлении в последние годы растут лавинообразно. По сложившейся традиции все компромиссные (при тех или иных ограничениях) и смешанные (синхронно-асинхронные) решения относят также к асинхронике.

Тем не менее в силу производственной и, как следствие, ментальной инерции асинхронные схемы не достигли сопоставимого с синхронными уровня промышленного развития.

*Самосинхронные* схемы относятся к асинхронным схемам (детали существующей не очень строгой терминологии приведены в книге далее). Они реализуют «чистый», бескомпромиссный подход в асинхронном построении схем и потому обладают свойствами, недостижимыми при смешанных и компромиссных подходах.

Одни отрасли техники развиваются эволюционно, последовательно, благодаря усилиям многих специалистов. Развитие других можно назвать революционным, происходящим благодаря единичным, прорывным публикациям, открывающим неизвестные ранее пути.

Развитие самосинхронных схем (первоначально и они назывались просто асинхронными) относится к революционному типу. Оно связано с разработками двух выдающихся ученых, которых можно считать классиками в асинхронике. Конечно, они имели сотрудников и соавторов, но, несомненно, были центральными фигурами в своих прорывных исследованиях. Это американский профессор Дэвид Маллер (David E. Muller) и российский профессор Виктор Ильич Варшавский.

В далекие уже 50-е годы XX века дискретные схемы назывались «переключательными» и строились на основе ламповых, транзисторных (транзисторы только появились) и даже релейно-контактных элементов. Хотя для элементов дискретных схем существовали и асинхронные прототипы, практические схемы строились с обязательной синхронизацией, так как иные способы тогда не были известны.



Дэвид Е. Маллер (1924—2008).

Фото 1959 г. (с разрешения архива Иллинойского университета)

В этих условиях 30-летний профессор прикладной математики Иллинойского университета (США) Д. Маллер задался вопросом об альтернативном способе построения переключательных схем. Тогда практической потребности в этом не было, но сыграл свою роль исследовательский дух. В результате в 1957 году появилась «Теория асинхронных схем» [1] (опубликована в 1959 году), ставшая одним из фундаментов современной асинхроники.

По прошествии лет можно утверждать, что это действительно выдающаяся, классическая работа. Простой и строгий язык, отточенные формулировки и доказательства, полная завершенность. Открыта (доказана) возможность существования асинхронных схем, определены и исследованы их свойства и поведение, сформулированы критерии их распознавания. Не оставлено никаких неясностей, на все возможные теоретические вопросы есть ответы. В современную асинхроннику вошли термины «теория Маллера», «гипотеза задержек Маллера», «модель Маллера», «С-элемент Маллера». Разработанный в теории способ определения отсутствия состязаний сигналов остается и поныне базовым критерием для таких схем.

Следует отметить, что «Теория асинхронных схем» — чисто математическая работа. Возможно, поэтому она долгое время не была востребована в электронике.

Д. Маллер участвовал в одном из проектов ILLIAC — серии ЭВМ Иллинойского университета в 50-е годы. Однако асинхронные схемы там не пошли — как из-за их необычности, так и оттого, что производственная база экономически не была готова для реализации таких схем. Первый фактор, в отличие от второго, действует и по сей день.

В этой теории Маллер опередил свое время. В дальнейшем в своих исследованиях он перешел к другим математическим проблемам теории автоматов, формальных языков и иным.

Теория асинхронных (самосинхронных) схем «зависла» на долгие 20 лет. Причина заключалась в том, что из этой теории никак не следовали способы построения конкретных схем, даже простейших, типа самосинхронного элемента И-НЕ или счетного триггера.

Такие способы были разработаны В. И. Варшавским с сотрудниками. В начале 70-х годов Варшавский был уже известным специалистом в теории автоматов. Столкнувшись однажды с трудностью формального синтеза, казалось бы, простого триггера, Варшавский и сотрудники предприняли широкий поиск литературы, в процессе которого и «открыли» для себя теорию Маллера. Эта теория стала основой их исследований и разработок в области уже конкретных самосинхронных схем. В итоге были предложены методы целенаправленного построения практических схем, названных самосинхронными, и множество типовых решений.

Результаты этих работ изложены в книгах [2, 3] и многих статьях. Варшавский со своей группой получили около 180 авторских свидетельств и патентов по самосинхронным схемам, за что сам руководитель был удостоен звания «Изобретатель СССР».

Плодотворная деятельность группы Варшавского, увы, остановилась на этапе внедрения. Группа выиграла всесоюзный конкурс Министерства электронной промышленности на разработку и изготовление самосинхронных микросхем, но произошло это летом 1991 года. Последовавшие политические и экономические преобразования сделали реализацию проекта невозможной.

Через некоторое время костяк группы во главе с Варшавским отбыл в международный университет Аизу (Япония). Но и там, в силу упомянутой промышленной инерции, не удалось реализовать какие-либо проекты конкретных микросхем.

В начале нового века члены группы рассеялись по разным странам и городам, и группа как целое перестала существовать. Последняя большая публикация группы — книга [4].



Виктор Ильич Варшавский (1933—2005)

В России на данный момент направление самосинхронных схем развивается единственной группой специалистов в составе Института проблем информатики РАН (ИПИРАН). Эта группа на рубеже 80—90-х годов сотрудничала (в качестве заказчика) с группой Варшавского и хорошо знакома со всеми ее работами. Усилия группы ИПИРАН направлены на проектирование конкретных самосинхронных схем, разработку методов и программных средств их создания. На счету группы полтора десятка патентов на самосинхронные схемы, публикации и доклады в научных изданиях и на конференциях. С работой группы можно ознакомиться на ее сайте <http://www.samosinhron.ru> (дочернем сайте ИПИРАН).

В настоящее время из-за все большей коммерциализации электроники среди фирм-изготовителей стало модным называть свои изделия самосинхронными (в том или ином варианте термина) с целью привлечения клиентов и заказчиков. Оценить «самосинхронность» таких изделий не представляется возможным, так как схемы их не публикуются. Приводимые же иногда фрагменты схем, «объясняющие» самосинхронность, недостаточны, поскольку это свойство нелокально; более того, порой такие фрагменты показывают как раз обратное — невозможность самосинхронности.

В мире выходит огромное количество публикаций по асинхронным схемам. Подавляющее большинство их не связано с асинхронностью по Маллеру: главным признаком асинхронности считается просто отсутствие глобальной синхронизации. Признаком самосинхронности многие авторы считают запрос-ответное взаимодействие блоков схем. Такое взаимодействие улучшает некоторые характеристики схем, но не является достаточным условием самосинхронности.

Публикуются также исследования схем с истинной самосинхронностью. Это работы в русле Маллера—Варшавского, имеющие явно выраженный теоретический характер и относящиеся в основном к теории автоматов и сетям Петри. Работы Д. Маллера и группы Варшавского также относятся к области теории автоматов. И те, и другие оказались слишком трудны для практических разработчиков электронных схем. В результате сложилась ситуация, когда разработчики электронной аппаратуры либо совсем не знают, либо слабо представляют, как работают самосинхронные схемы и чем они отличаются как от синхронных, так и от других асинхронных схем.

Единственной методологией, позволяющей разрабатывать самосинхронные схемы, является методология *Null Convention Logic* (NCL) [5] фирмы Theseus Research Incorporated. NCL-методология появилась в конце 1990-х годов и изложена без упоминания основополагающих работ Маллера и Варшавского. Методология, по-видимому, претендующая на новизну, содержит ряд далеко не новых приемов. Одна из главных ее основ — двухфазная дисциплина — была предложена группой Варшавского в [2] за 20 лет до появления NCL. В последней предлагается жесткая конвейерная структура построения схем с запрос-ответным взаимодействием, также давно известная, и ограниченный набор базовых элементов. С одной стороны, это позволило избежать больших проблем проектирования, но, с другой стороны, получаемые схемы весьма далеки от оптимальности. По разным оценкам, имеющимся в литературе, такие схемы в 2,5—4 раза избыточнее по числу транзисторов сравнительно с другими возможными решениями и соответственно неоптимальны по быстродействию и энергопотреблению. Следует отметить также, что фирма только пропагандирует методологию, но изделий по ней не выпускает, заказы на проектирование не принимает и программных средств не предоставляет. Более подробно NCL-методология рассмотрена далее в книге.

Таким образом, при немалом числе теоретических работ по *самосинхронике* весьма велик дефицит литературы, доступной большинству разработчиков схем и систематически излагающей все аспекты создания самосинхронных схем — от теории до практики.

Настоящая книга предназначена по возможности заполнить этот пробел.

# Глава 1

## ВВЕДЕНИЕ В САМОСИНХРОНИКУ

---

### 1.1. Проблемы работы и особенности цифровых схем

Отличительные особенности самосинхронных схем (СС-схем) связаны с надежностью их функционирования. Термины, относящиеся к надежности, определены в стандарте [6] и ниже будут выделены кроме курсива жирным шрифтом.

Одна из главных проблем работы цифровых электронных схем — возникновение **ошибок**. Ошибки в данном случае — это несоответствие значений сигналов внутри схемы и на ее выходах правильным значениям, предусмотренным при их создании.

Ошибки могут порождаться внешними воздействиями и внутренними причинами.

Внешние воздействия связаны с такими факторами, как электромагнитные наводки, удары энергичных элементарных частиц, влажность, перегрузки и другие подобные явления. Вопрос защиты от внешних воздействий составляет отдельное направление исследований и не является предметом данной книги.

Внутренние причины ошибок обусловлены способом построения схемы и условиями, в которых она работает. Таких внутренних причин две.

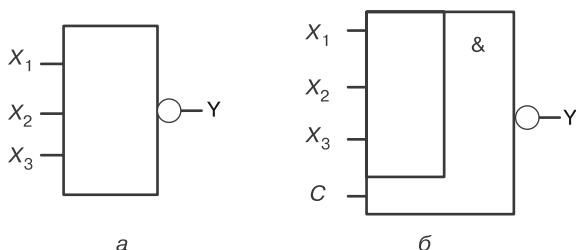
1. **Состязания** сигналов (*гонки*) на элементах, т. е. такие изменения входов элементов, которые вызывают ошибочные изменения их выходов и, как результат, выходов всей схемы.

2. Возникновение **отказов** схемы. Отказы происходят при потере физической работоспособности внутренних структур, вызванной условиями работы: температурой, напряжением питания, а также старением и другими процессами.

На примере состязаний можно объяснить синхронный и асинхронный принципы работы схем. Собственно, синхронный принцип и возник как способ избежать состязаний.



Доказано (и это практически очевидно), что любой логический элемент, имеющий два или более входов, подвержен состязаниям. Рассмотрим логический элемент на рис. 1.1, а.



**Рис. 1.1.** Логический элемент (а) и элемент с блокирующим сигналом  $C$  (б)

В любом элементе всегда найдется пара входов, например  $x_1$  и  $x_2$ , обладающая следующим свойством: если  $x_1$  меняется первым, а  $x_2$  — вторым, то выход  $y$  не меняется; если же сначала меняется  $x_2$ , а затем  $x_1$  (с некоторой задержкой), то на выходе  $y$  появится импульс, как правило, нежелательный. Этот эффект и составляет сущность гонок.

Причины неодновременного, рассогласованного изменения сигналов известны: разброс параметров и, соответственно, задержек схемы, старение, влияние нагрузок, температуры, напряжения питания и других факторов.

Поскольку при произвольном порядке изменений входных сигналов элементов избежать гонок невозможно, единственным путем борьбы с ними будет упорядочение этих изменений, т. е. введение дисциплины сигналов.

Дисциплина сигналов состоит в том, что в работе схемы или ее частей чередуются два периода: в одном периоде изменения входов допустимы (разрешенный период), а в другом — нет (запрещенный период).

Одним из простейших способов организации дисциплины является введение блокирующего сигнала (рис. 1.1, б, сигнал  $C$ ). Разрешенный и запрещенный периоды определяются его значениями: на рисунке при  $C = 1$  — запрещенный, при  $C = 0$  — разрешенный период.

Синхронный принцип реализуется тем, что блокирующий сигнал подается от тактового генератора, который и определяет оба периода. Как правило, тактовый генератор является общим (глобальным) для всей схемы.

Асинхронный принцип отличается от синхронного тем, что либо блокирующего сигнала нет и дисциплина организуется другим способом, либо, если он есть, формируется локально и не связан с глобальным тактированием.

Разнообразие возможных дисциплин при асинхронном принципе порождает и большое разнообразие асинхронных схем.

Например, широко известна так называемая *соседняя дисциплина*, когда в разрешенный период (см. рис. 1.1, *a*) допускается изменение только одного входного сигнала. При изменении одного сигнала гонок не возникает. Запрещенный период в этом случае необходим для подготовки схемы к следующему циклу.

Возможность возникновения гонок сильно зависит от способа формирования разрешенного и запрещенного периодов.

В *синхронном принципе* эти периоды выбираются априорно, исходя из худшего случая и «с запасом», когда можно считать, что изменения сигналов (переходные процессы) закончились. В этом принципе текущие (фактические) изменения задержек элементов схемы, по сути, не учитываются, что и является причиной возможных гонок.

В *асинхронном принципе* периоды, как правило, не постоянны, и принимаются те или иные меры согласования их длительностей с текущими задержками. В ряде асинхронных реализаций для этого используются определенные допущения и предположения. Например, в соседней дисциплине запрещенный период должен превышать длительность переходных процессов, инициированных в разрешенном периоде. Другой пример — «моделирование» задержки некоторого фрагмента схемы цепочкой инверторов. Здесь предполагается, что при одинаковых внешних условиях задержка фрагмента и цепочки инверторов меняется одинаково. Тем не менее во всех подобных случаях избежать несоответствия априорных оценок характеру действительных процессов не удастся, и эти несоответствия могут порождать гонки.

Принципиальным отличием *самосинхронного принципа* является то, что оба периода определяются не предположительно, а по *фактическому окончанию* переходных процессов. И как бы ни менялись задержки элементов в этом случае, гонки возникнуть не могут.

Вернемся к проблеме ошибок. Радикальным решением этой проблемы было бы устранение двух названных их причин.

В существующих подходах (помимо самосинхронного) гонки не могут быть устранены полностью. Можно устранить их в каком-либо блоке, но в окружающих его схемах надо обеспечить дисциплину сигналов для этого блока. Окружающие схемы, однако, для ликвидации гонок тоже требуют обеспечения дисциплины в своих окружениях и т. д. Задача оказывается нереальной.

А каково идеальное решение для второй причины ошибок? Предвидеть отказы при работе схемы невозможно, однако ошибок не будет, если сразу после отказа схема остановится и не выдаст на выходах

непредусмотренных значений. В книге [3] данное свойство называется самопроверяемостью (в действующем стандарте [6] — *самопроверка*). Однако более близкий стандартный термин, отражающий это свойство, — *отказобезопасность*, определяемая как «свойства изделия, ориентированные на сохранение безопасности в случае отказа». Это общее понятие конкретизируется для рассматриваемого свойства в следующем разделе.

Уникальность и замечательность самосинхронных схем состоят в том, что они реализуют практически идеальное решение: гонки в них отсутствуют полностью и отказобезопасность также обеспечена, поскольку схемы останавливаются при возникновении отказов по широкому набору наиболее значимых причин.

## 1.2. Самосинхронные схемы и их свойства

Название «самосинхронные схемы» впервые появилось в работах Варшавского с сотрудниками. В книге [3] такие схемы еще назывались аperiодическими или самосинхронизирующимися, но позже авторы перешли на термин «самосинхронные». Теория самосинхронных схем основана на теории Маллера [1].

Условия применения теории для таких схем получили название *модели Маллера*. В ней принято, что задержки элементов приведены к их выходам, а задержки межсоединений после разветвлений пренебрежимо малы по сравнению с задержками элементов. Под элементом понимается устройство, имеющее один выход и описываемое одним логическим уравнением. Задержка элемента может иметь любое конечное значение. Более подробно об элементах СС-схем рассказано в главе 2.

В теории Маллера рассматриваются схемы, удовлетворяющие данной модели и имеющие свойство *полумодулярности* [1]; этот термин поясняется в разделе 2.3. Подобные схемы были названы *speed-independent*, что по-русски можно передать как независимые от задержек элементов по Маллеру (НЗЭМ).

В работах Варшавского с сотрудниками [2, 3] доказано, что самосинхронные схемы кроме свойства НЗЭМ имеют еще и диагностические свойства. По мнению авторов, «решающим достоинством аperiодических схем являются их самодиагностические свойства».

Более точное описание самосинхронности приведено в главе 5, а с внешней, потребительской точки зрения можно дать такое определение [7].

*Самосинхронной схемой (СС-схемой) называется схема, обладающая двумя свойствами безошибочной работы:*

- *отсутствием гонок при любых конечных задержках элементов;*
- *отказобезопасностью.*

Отсутствие гонок обеспечивается свойством НЗЭМ.

Отказобезопасность СС-схем понимается как остановка (прекращение изменений сигналов), вызванная константными **неисправностями** типа *залипания* на 0 и 1 на выходах (КНЗ-01), т. е. сохранением постоянных значений 0 или 1 на выходах одного или нескольких элементов. В технической диагностике цифровых схем они считаются наиболее частыми и практически наиболее значимыми. Действительно, такую неисправность можно толковать как задержку бесконечной величины, а СС-схема, ожидая переключения элементов, в этом случае останавливается.

Остановка СС-схем на выходные неисправности элементов гарантируется их построением. Кроме того, эти схемы могут останавливаться и по отношению к ряду других *логических неисправностей*, в частности залипаний на входах элементов.

Безопасность СС-схем в случае отказов, во-первых, означает отсутствие непредусмотренных значений, во многих случаях опасных, на их выходах. Во-вторых, она позволяет строить схемы, способные к *самопроверке* и *самотестированию*, т. е. обнаруживающие ошибки и оценивающие свое состояние во время работы; дает возможность *диагностирования неисправностей* и в конечном итоге получения надежных схем.

Следует подчеркнуть, что СС-схемы, работая по фактическим задержкам, тем самым реализуют предельно высокое быстродействие, возможное в данных конкретных условиях. В отличие от схем других типов, у них не предусмотрен «запас быстродействия», и элементы «сами себя синхронизируют». Такая известная операция, как «разгонка» компьютеров, для СС-схем неприменима, поскольку они и так уже «разогнаны» до предела по способу построения.

Совокупность приведенных свойств СС-схем является уникальной и присуща только им. Эти свойства обеспечивают целый ряд практически важных преимуществ подобных схем.

1. Отсутствие состязаний сигналов.
2. Устойчивость функционирования (правильность работы) в предельно широком диапазоне внешних условий (температуры и напряжения питания), ограниченном только физиче-

ской работоспособностью транзисторных и топологических структур.

3. Как следствие предыдущего, способность работать при пониженном питании. Это дает возможность снижения энергопотребления и продления срока работы при батарейном питании.
4. Нечувствительность к разбросу и изменению параметров из-за старения элементов, т. е. увеличение срока службы.
5. Достоверность обработки информации (отсутствие ошибок).
6. Сравнительно простое построение надежных схем — самопроверяемых и отказоустойчивых.
7. Увеличенный выход годных чипов за счет нечувствительности к разбросу параметров.
8. Равномерность токопотребления по сравнению с синхронными схемами за счет отсутствия бросков тока, порождаемых синхроимпульсами.
9. Упрощение тестирования — функциональные тесты одновременно являются и диагностическими на выходные КНЗ–01.

Платой за перечисленные ценные качества СС-схем являются повышенные затраты по числу транзисторов при их реализации. В зависимости от особенностей схем эти затраты могут быть на 10—100% больше, чем для схем других типов.

Заметим также, что быстродействие СС-схем определяется двумя взаимно противоположными факторами. Работа по фактическим задержкам увеличивает их быстродействие, однако в их рабочем цикле, наряду с рабочей фазой, присутствует и вспомогательная фаза, что снижает этот показатель. В результате быстродействие таких схем может быть как больше, так и меньше, чем у синхронных аналогов, что зависит от качества проектирования СС-схем. Имеющийся опыт показал, что во многих реальных проектах СС-схем удалось сделать их быстродействие лучше синхронных аналогов.

Противоположными факторами обусловлено и энергопотребление СС-схем. Отсутствие тактовых генераторов и цепей разводки синхроимпульсов уменьшает его, но наличие дополнительных транзисторов для обеспечения самосинхронности увеличивает. Все же, как правило, энергопотребление СС-схем ниже, чем их синхронных аналогов.

При сравнении синхронных и самосинхронных реализаций надежных схем преимущество последних проявляется по всем параметрам [8] (модельная оценка).

### 1.3. Экспериментальная проверка уникальных свойств СС-схем

С практической точки зрения, крайне интересен и важен вопрос, оправдываются ли теоретически заявленные уникальные свойства в реальных самосинхронных схемах. Об этих свойствах писал и говорил еще Варшавский, но проверить их экспериментально ему не удалось ни в России, ни за рубежом.

Доступные публикации по этой теме очень фрагментарны, и обнаружить сообщения о более или менее представительных испытаниях реальных асинхронных схем также не удастся. В лучшем случае, как, например, в [9], приводятся данные при изменении температуры и номинальном напряжении 5 В и при изменении напряжения от 3,5 до 6 В и номинальной температуре. Причем проверяемая схема была асинхронной, но не самосинхронной.

Полноценные испытания самосинхронных схем проводились в 2005—2006 годах в ИПИРАН совместно с НПО «Физика» (г. Москва) и Технологическим центром МИЭТ (г. Зеленоград) [10].

Для начала была разработана простая самосинхронная схема, содержащая запоминающую часть (трехразрядный счетчик), комбинационную часть (трехразрядный сумматор) и Г-триггеры — типичные элементы самосинхронных схем (рис. 1.2).

Схема работает автономно и циклически. Счетчик увеличивает свое содержимое на единицу каждый рабочий цикл. Сигналы с разрядов счетчика подаются на сумматор и суммируются с внешними сигналами  $N_0$ — $N_2$ , постоянными для каждого эксперимента. Эти внешние (одиночные) сигналы для суммирования преобразуются по правилам построения СС-схем.

С помощью Г-триггеров формируется общий сигнал индикации схемы *Ind*. В эксперименте этот сигнал должен быть замкнут внешней связью с входным управляющим сигналом СС для обеспечения режима самогенерации — рабочего режима всех самосинхронных схем.

Другие внешние сигналы схемы:  $Y_0$ — $Y_3$  — выходы сумматора и его перенос, *Reset* — установка счетчика, *IR* — индикатор установки счетчика.

Приведенная схема была реализована на двух разных предприятиях по двум независимым КМДП-технологиям — 3 мкм (НПО «Интеграл», г. Минск, по заказу НПО «Физика») и 1,6 мкм (НПК «Технологический центр» МИЭТ).

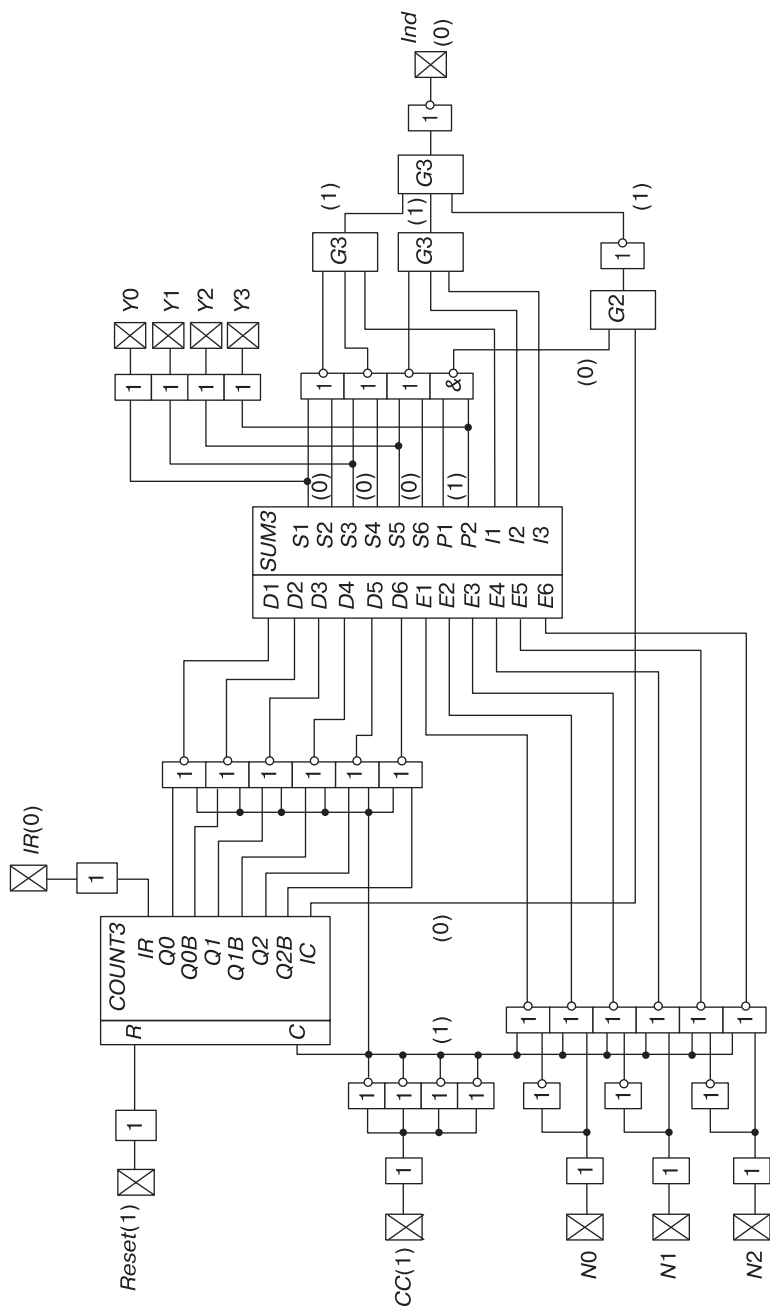


Рис. 1.2. Экспериментальная схема

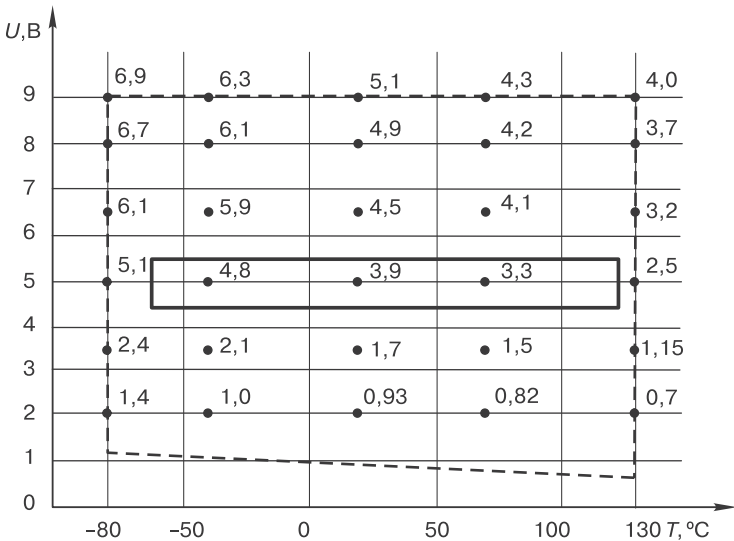
Изготовленные экземпляры СС-схемы помещались в климатическую установку. Эксперименты проводились с помощью простых приборов: регулируемого источника питания, осциллографа и логического анализатора.

В экспериментах проверялась правильность работы схем: в разомкнутом состоянии — по статическим значениям выходов сумматора, в замкнутом состоянии — по характерной периодической картине этих сигналов на осциллографе. Также измерялась собственная частота в режиме самогенерации для оценки быстродействия.

На рис. 1.3 и 1.4 приведены результаты экспериментов по проверке работоспособности обеих реализаций (по несколько экземпляров) данной схемы.

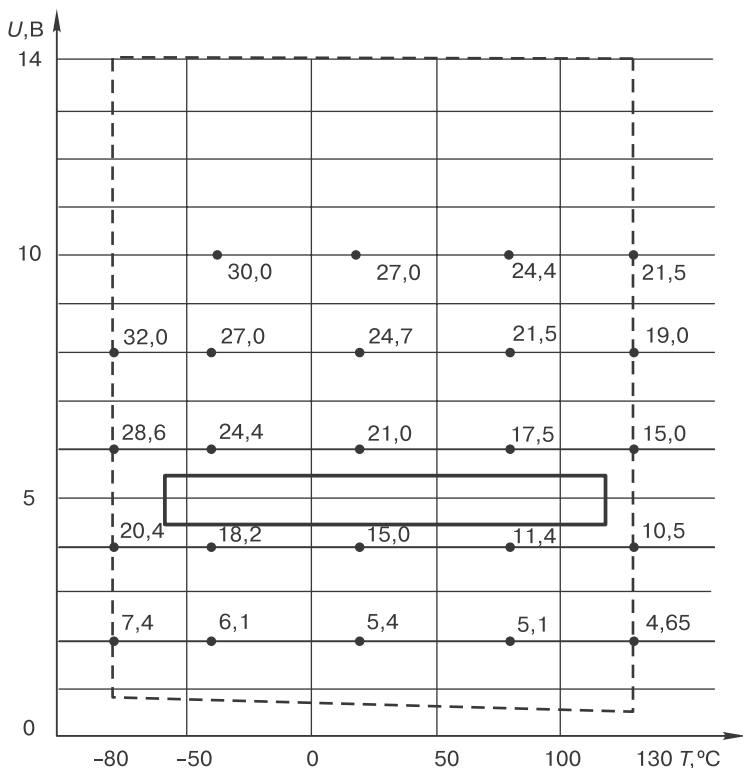
Внутренний прямоугольник на рисунках показывает паспортную область работоспособности обычных синхронных микросхем данных типов. Следует отметить, что для синхронных реализаций и в паспортной области нет полной гарантии отсутствия состязаний.

В отмеченных точках на рисунках указаны частоты самогенерации (в МГц). Обращает на себя внимание широкий диапазон их значений. Это объясняется тем, что самосинхронные схемы работают без ожидания каких-либо тактовых сигналов и потому с быстродействием, максимально возможным в данных внешних условиях.



**Рис. 1.3.** Область правильной работоспособности первой реализации микросхемы (3,0 мкм)





**Рис. 1.4.** Область правильной работоспособности второй реализации микросхемы (1,6 мкм)

Левая и правая (температурные) границы областей определялись только возможностями климатической установки. По всей видимости, эти границы могли бы быть расширены, особенно в сторону низких температур.

В области низких питающих напряжений образцы СС-схем вели себя в полном соответствии с теорией: при наступлении некоторого порогового напряжения они прекращали работу — самогенерация останавливалась, изменения выходов прекращались. При повышении напряжения самогенерация самопроизвольно возобновлялась.

На верхних границах областей, показанных на рисунках, все экземпляры были работоспособны. При дальнейшем повышении напряжения они безвозвратно выходили из строя (также с прекращением самогенерации), в зависимости от экземпляра, в диапазоне от 9,5 до 13,5 В для первой реализации и от 14,5 до 15,5 В — для второй.

На втором этапе экспериментов испытывалось полноценное вычислительное устройство «Микроядро» [11], реализующее основные команды известного 8-битного микроконтроллера PIC18XX и изготовленное по той же 1,6-мкм технологии в ТЦ МИЭТ. Результаты экспериментов по определению диапазонов работоспособности полностью совпали с рис. 1.4. Этот факт позволяет предположить, что область работоспособности зависит скорее не от размера схемы, а от переключаемых свойств транзисторных структур, определяемых конкретной технологией.

Таким образом, впервые в России (а возможно, и в более широком регионе) проведено прямое экспериментальное исследование ряда теоретически утверждаемых свойств самосинхронных схем: независимости правильной работы от задержек элементов, безопасности функционирования, беспрецедентно широкого диапазона работоспособности.

## 1.4. Физический подход к достижению самосинхронности

Претендентами на самосинхронность иногда считались и считаются схемы с токовыми индикаторами, например [12]. (Подобные работы велись и в группе Варшавского, но доступных ссылок найти не удалось.)

Данная идея относится к КМДП-схемам и состоит в том, что ток в таких схемах (не считая утечек) протекает только во время переходных процессов. Предлагается для определения окончания переходных процессов в некоторой схеме поставить индикатор общего протекающего через нее тока.

Такой подход, однако, не гарантирует самосинхронности. Токовая индикация не обеспечивает отсутствие гонок, например, при соединении параллельных ветвей схемы. Отказобезопасность также не гарантируется: если в измеряемой схеме появляется неисправность, то индикатор не обязательно отреагирует достаточным изменением тока, а во многих случаях будет продолжать работать.

По названным причинам подобную физическую реализацию, наряду с «индикатором» в виде параллельной цепочки инверторов, следует отнести к *квази-самосинхронным* (разновидность асинхронных) решениям.

Дополнительным недостатком токовых индикаторов становится сам факт введения аналоговых компонентов в цифровые схемы,

что ухудшает их эксплуатационные свойства, уменьшает диапазон работоспособности, удорожает производство и т. п.

## 1.5. Об используемой терминологии

По аналогии с существующим термином *асинхроника* далее для краткости будем называть работы, посвященные СС-схемам, *самосинхроникой*.

В зарубежной и отечественной литературе используется несколько названий для схем, в той или иной мере относящихся к самосинхронике.

*Asynchronous* — асинхронные. Схемы, у которых отсутствует внешняя (глобальная) синхронизация или общий тактовый генератор.

*Speed-independent (SI)* — независимые от скорости. Термин, впервые введенный Маллером, означает независимость конечного поведения от выходных задержек элементов схемы. В наше время термин выглядит устаревшим, так как непонятно, какая скорость имеется в виду. Однако широко используется в зарубежной литературе как дань традиции.

*Delay-insensitive (DI)* — нечувствительные к задержкам. Здесь имеются в виду как задержки элементов, так и задержки в «проводах» (межсоединениях). В точном смысле термин применим только к схемам «на чипе», реализованным физически. На этапе же разработки логической схемы, когда он обычно и применяется в зарубежной литературе, этот термин не вполне корректен по двум причинам. Во-первых, как следует из теории Маллера, создать схему, независимую от любых задержек межсоединений после разветвлений (то есть *приведенных ко входам* элементов), невозможно. Во-вторых, для реализации такой независимости в конкретных случаях требуются сведения о топологии (длины трасс), технологии (величина *эквифронной зоны*) и др. Одна и та же электрическая схема при разных реализациях может зависеть или не зависеть от задержек после разветвлений.

Как компромисс при использовании этого термина обычно применяют «слабые условия» независимости [13], означающие, что трассы после разветвлений находятся в эквифронной зоне. Но и в таком случае эти условия должны контролироваться на этапе топологии.

*Self-timed (ST)* — самотактируемые, или самосинхронные. В зарубежной литературе термин применяется к разнообразным асинхронным схемам, имеющим какую-либо обратную связь, например запрос-ответное взаимодействие. Следует помнить, что простое

наличие обратной связи отнюдь не гарантирует независимость от скорости по Маллеру. При употреблении этого термина, как правило, о свойствах отсутствия гонок и особенно отказобезопасности не упоминается.

Иногда к названным терминам, в случаях их ограниченной применимости, добавляют приставку *quasi-*.

*Самосинхронные*. В русскоязычной литературе этот термин впервые введен в работах Варшавского с сотрудниками (см. выше).

В работах ИПИРАН иногда применялся термин *строго самосинхронные*, полностью равнозначный термину *самосинхронные* и призванный подчеркнуть отличие от нестрогого зарубежного термина *self-timed*. В дальнейшем от данного термина отказались.

К сожалению, как в зарубежных, так и в отечественных публикациях зачастую содержатся небрежности, прямые ошибки в употреблении указанных терминов (например, в Wikipedia), а то и спекуляции на их «престижности» в рекламных и иных целях.

С недавних пор (с 2008 года) в (рецензируемых!) журналах «Электронные компоненты» ([www.elcp.ru](http://www.elcp.ru)), «Автоматизация в промышленности» ([www.avtprom.ru](http://www.avtprom.ru)), «Компоненты и технологии» ([www.kit-e.ru](http://www.kit-e.ru)) начали появляться статьи по самосинхронике одной и той же группы авторов, от трети до половины скопированные (иногда с ошибками) с материалов сайта фирмы Theseus Research Inc по NCL-методологии [5]. В этих статьях термины *самосинхронные* и *строго самосинхронные* трактуются произвольно, без понимания смысла, а «собственные» материалы авторов повторяют общие соображения об энергопотреблении КМДП-схем (известные более 40 лет) и о надежности самосинхронных схем (известные 25 лет, с работ Варшавского). Являясь плагиатом и компиляцией, статьи лишены научной новизны, ценности не представляют и лишь вводят в заблуждение неискующих читателей.

## Глава 2

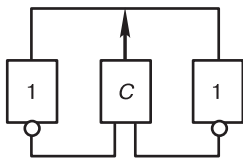
# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ САМОСИНХРОННЫХ СХЕМ

---

В отличие от синхронных, а также других типов асинхронных схем самосинхронные схемы могут работать при любых задержках своих элементов. Задержки элементов будем считать произвольными, конечной величины. В этих условиях самосинхронные схемы должны быть построены специальным образом.

### 2.1. Принципы построения и работы СС-схем

СС-схемы можно строить по-разному. Во многих теоретических работах приводятся схемы, иллюстрирующие теорию, но не имеющие практического применения. Одна из первых самосинхронных схем приведена в работе Маллера и Бартки [1] (рис. 2.1 — в современном начертании).



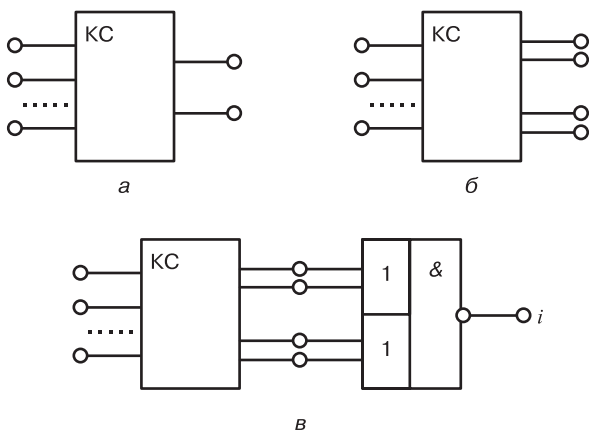
**Рис. 2.1.** Самосинхронная (speed-independent) схема из работы [1]:  
С—С-элемент Маллера

Схема демонстрирует два параллельных канала (процесса) из двух инверторов и объединяющий их С-элемент Маллера. Именно этот элемент позволяет корректно, без гонок, соединить два процесса.

Однако теория Маллера дает только инструмент анализа СС-схем. Для их практического построения (синтеза) необходимо рассматривать разомкнутые схемы, имеющие значимые для них входы и выходы.

ды. Соответствующие методы построения и разработаны В. И. Варшавским с сотрудниками [2—4].

Далее для примера возьмем схему с двумя выходами (рис. 2.2, а).



**Рис. 2.2.** Комбинационная схема (КС):

а — исходная схема; б — схема в кодах; в — схема в кодах с индикатором;  
i — индикаторный сигнал

### 2.1.1. Индикация

Для построения любой дискретной схемы принципиально важно знать моменты окончания переходных процессов в различных частях схемы или, что равнозначно, задержки ее элементов. В синхронных схемах это знание учитывается априорно и определяет период повторения тактовых сигналов.

Допуская произвольные задержки и отказываясь от синхронного принципа, мы должны прямо измерять моменты завершения переходных процессов в различных точках схемы. Как это можно сделать?

Определить такие моменты на выходах схемы — значит, сформировать некоторый двоичный сигнал, сообщающий об окончании переходного процесса в ответ на изменение входов схемы. Такой сигнал будем называть *индикаторным (И-сигналом)*, а схему его формирования — *индикатором*.

В зависимости от построения схемы существуют два варианта индикации. В первом выходные сигналы индицируются непосредственно, что свойственно комбинационным схемам. Второй вариант — ячейки памяти — требует другого подхода.

Рассмотрим сначала первый вариант.

### **А. Кодовая индикация.**

Два выхода схемы на рис. 2.2, *а* независимы друг от друга и могут принимать такие сочетания значений (назовем их *рабочими наборами*): 00, 01, 10, 11. Предположим, что при некотором изменении входов выходные сигналы должны перейти от набора 01 к набору 10.

Такой переход может произойти следующими путями:  $01 \rightarrow 10$ ,  $01 \rightarrow 00 \rightarrow 10$  или  $01 \rightarrow 11 \rightarrow 10$ . Каков бы ни был путь перехода, любой следующий набор после исходного 01 также является рабочим, и мы лишены возможности узнать, закончился переходный процесс или еще будет продолжаться.

Таким образом, в «обычной» схеме, с «обычными» выходами не удастся найти способ определения окончания переходных процессов.

Единственным решением этой задачи в рамках двоичной логики является следующее. Необходимо расширить кодовое пространство выходных сигналов, введя кроме рабочих еще некоторое множество *переходных наборов*. Эти два множества не должны пересекаться, чтобы быть легко различимыми с помощью простых электронных схем. Другими словами, надо применить подходящее (избыточное) кодирование сигналов схемы.

Возьмем следующий способ кодирования: исходное значение 1 кодируется как 01, а значение 0 — как 10. В этом случае от схемы на рис. 2.2, *а* мы переходим к эквивалентной схеме на рис. 2.2, *б*. Для нее рабочими будут четыре набора: 0101, 0110, 1001 и 1010, а остальные двенадцать из шестнадцати возможных — переходными.

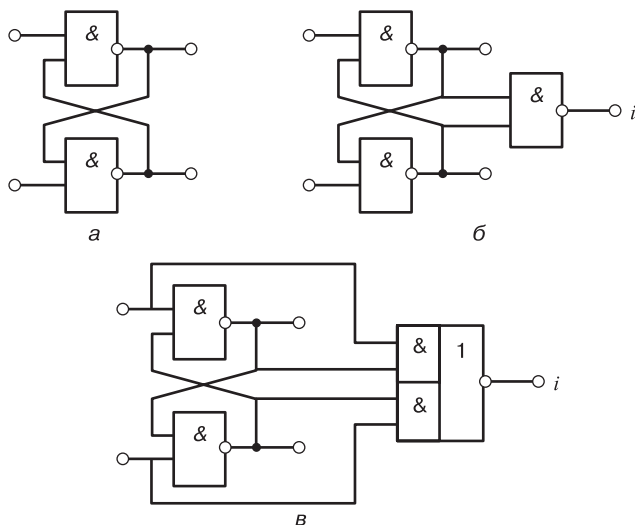
Далее будем считать, что все переходы между рабочими наборами происходят только через переходные наборы (реализация этого условия изложена в следующем разделе).

В принятых условиях переход  $01 \rightarrow 10$  исходной схемы может реализоваться в эквивалентной схеме, например, так:  $0110 \rightarrow 0010 \rightarrow 0011 \rightarrow 0001 \rightarrow 1001$ . Видно, что здесь первый и последний наборы — рабочие, остальные — переходные. Поскольку два множества наборов четко разделены, мы можем построить схему индикации переходов и, значит, окончания переходных процессов (рис. 2.2, *в*).

Кодовые системы, пригодные для такой индикации, получили название *самосинхронизирующихся кодов* [3]. Более подробно они описаны в одном из следующих разделов.

### **Б. Индикация ячеек памяти.**

Рассмотрим простейшую ячейку памяти — бистабильную ячейку (БСЯ) (рис. 2.3, *а*). В ней значения обоих выходов в стабильном состоянии противоположны друг другу, а в переходном — совпадают.



**Рис. 2.3.** Бистабильная ячейка:  
*а* — исходная; *б* — с индикацией по выходам;  
*в* — с индикацией по входам и выходам;  
*i* — индикаторный сигнал

Есть две идеи индикации БСЯ [3].

Первая идея — индикация с использованием только выходов ячейки (рис. 2.3, *б*). Предположим, в исходном состоянии входы ячейки имеют значения 11, выходы — 01 (по порядку сверху вниз), сигнал индикации — 1. Подадим на входы значения 01. Тогда выходы будут изменяться так: 01 → 11 → 10, а И-сигнал соответственно 1 → 0 → 1. Таким образом, значение 1 И-сигнала показывает стабильное состояние БСЯ, а 0 — переходное.

Однако, чтобы использовать индикацию в СС-схемах, необходимо согласовать входы ячейки и выход индикатора с фазой работы (см. ниже). Для БСЯ это означает, что И-сигнал в случае одинаковых входов ячейки должен отличаться от его значения при противоположных значениях входов.

Для такого способа индикации согласование достигается только в режиме гашения (на рисунке входы должны быть равны 00). Но в этом режиме записанная ранее в ячейку информация уничтожается, и применение БСЯ теряет смысл. По этой причине первая идея не нашла применения в СС-схемах.

Вторая идея предусматривает запоминание информации и согласование фаз (рис. 2.3, *в*). Она состоит в установлении факта соответ-



ствия между входами и выходами БСЯ. Если выходы соответствуют входам, то переходные процессы закончились, иначе — они не завершены. На рисунке И-сигнал для этого способа изменяется как  $0 \rightarrow 1 \rightarrow 0$ , где 0 — стабильное состояние, 1 — переходное.

Принципиально важным моментом в обоих вариантах индикации является то, что по построению выход индикаторного элемента индицирует окончание переходных процессов не только в основной схеме, но также и в самом себе. Другими словами, нет необходимости строить специальную схему индикации для индикатора, что не обязательно справедливо для других подходов, например физического.

В схеме может быть много И-сигналов. Все они схемотехнически комбинируются по несложным правилам (изложенным ниже) в один сигнал, общий для всей схемы. Таким образом построенный общий И-сигнал индицирует все внутренние элементы и сигналы схемы.

Индикация переходных процессов является важнейшим принципом построения практических СС-схем. Реализация именно этого принципа организует работу таких схем по факту завершения переходных процессов, а не по предположению, как при построении схем других типов.

Индикация обеспечивает свойство отказобезопасности СС-схем, так как при возникновении КНЗ-01 индикатор не переключится. Она же делает работу схем максимально быстрой в текущих условиях.

Однако индикация не гарантирует отсутствия состязаний в индицируемых элементах, в частности в триггерах.

### 2.1.2. Двухфазная дисциплина

Вернемся к примеру комбинационной схемы на рис. 2.2. Пусть ее некодированные выходы по-прежнему делают переход  $01 \rightarrow 10$ . Так как задержки в схеме по предположению произвольны, при некотором их соотношении может возникнуть коллизия в схеме на рис. 2.2, в. Например, выходные кодированные сигналы реализуют переход  $0110 \rightarrow 0010 \rightarrow 1010 \rightarrow 1011 \rightarrow 1001$ . В этом случае первая пара уже закончила свой переход в новое состояние, а вторая пара, в силу больших задержек, еще не успела измениться. В результате в середине перехода возникло состояние 1010, относящееся к рабочему множеству, и индикатор должен показать завершение переходного процесса, хотя фактически тот не окончен.

Для исключения подобных коллизий, порождающих состязания сигналов, налагается требование недопустимости появления при переходе других рабочих наборов, кроме начального и конечного, т. е. соблюдения определенной *дисциплины* переходов.

Опыт показал, что наиболее эффективной и простой дисциплиной является следующая. В множестве переходных наборов выделяется один или несколько специальных наборов, называемых *спейсерами*. На все переходы сигналов налагается требование, чтобы они обязательно совершались через множество спейсеров. В этом случае каждый переход осуществляется в два этапа, или *фазы*: от начального рабочего набора через промежуточные к спейсеру и от спейсера — к конечному рабочему набору, причем также через промежуточные.

Такая дисциплина получила название *двухфазной*. Фазу перехода от рабочего набора к спейсеру принято называть *спейсерной*, а от спейсера к рабочему набору — *рабочей* фазой.

В рабочей фазе реализуются те функциональные преобразования сигналов, для которых схема предназначена (логические операции, суммирование и т. п.). В спейсерной фазе — вспомогательной — происходит подготовка к следующим рабочим преобразованиям. Каждая фаза начинается только после того, как полностью закончится предыдущая.

При двухфазной дисциплине задача индикации существенно упрощается по сравнению с общим случаем: поскольку прохождение через спейсер обязательно, индикатору нет необходимости отличать рабочие наборы от всех переходных. Ему достаточно распознать только один спейсер, который является «полномочным представителем» всего множества переходных наборов в данном месте схемы.

Для максимального упрощения индикации целесообразно выбирать спейсерные наборы, максимально легко отличимые от рабочих. Поэтому в качестве спейсеров обычно берут наборы с одинаковыми значениями связанных кодовых сигналов, например 0000 или 1111.

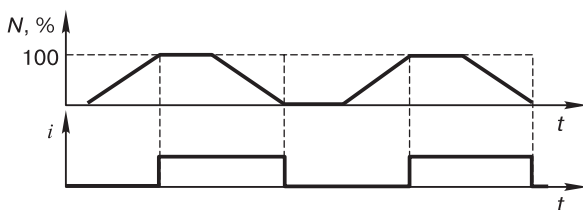
Можно отметить, что задача индикации фаз эквивалентна задаче индикации (различения) рабочих и переходных наборов. В практических схемах последняя задача без использования двухфазной дисциплины является нереальной, что делает эту дисциплину необходимым условием создания СС-схем.

На рис. 2.4 показана качественная картина индикации схемы при двухфазной дисциплине.

Важным следствием двухфазной организации работы является следующее.

*Любой сигнал СС-схемы в каждой фазе меняется не более одного раза.*

Наряду с другими свойствами это следствие обеспечивает отсутствие состязаний. Двухфазная дисциплина также упрощает индикацию.

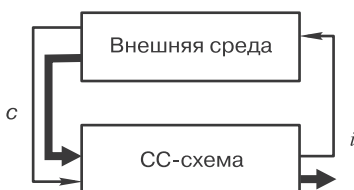


**Рис. 2.4.** Индикация схемы в двухфазной дисциплине:  $N$  — количество сигналов, находящихся в рабочей фазе (остальные — в переходном состоянии);  $i$  — индикаторный сигнал;  $i = 0$  — спейсерная фаза;  $i = 1$  — рабочая фаза

### 2.1.3. Обратная связь

В общем случае на вход СС-схемы наряду с *информационными сигналами (инфосигналами)* подается и сигнал, инициирующий ее переход в очередную фазу. По отношению к СС-схеме этот сигнал будет управляющим. Управляющий сигнал обеспечивается окружающими схемами (внешней средой). На выходе СС-схемы должен присутствовать ее И-сигнал.

Взаимодействие схемы с одним управляющим сигналом иллюстрируется на рис. 2.5.



**Рис. 2.5.** Взаимодействие СС-схемы с внешней средой;  $c$  — управляющий сигнал;  $i$  — индикаторный сигнал

При взаимодействии с внешней средой могут возникнуть соστάзания на входах схемы, и для их исключения необходима определенная дисциплина. Такая дисциплина в данном случае приводит к двум правилам:

- 1) управляющий сигнал на схему переходит в очередную фазу только после перехода в ту же фазу входных инфосигналов схемы;
- 2) входные инфосигналы схемы могут меняться только в одной из фаз. В другой фазе, когда инфосигналы обрабатываются, они должны оставаться неизменными.

Предположим, что значения 0 управляющего и индикаторного сигналов соответствуют спейсеру, а значения 1 — рабочей фазе. Схема

работает циклически, по четыре шага в цикле. Примем за исходную фазу системы спейсер.

**Шаг 1.** Установка средой на входах схемы новых инфосигналов, переход  $c = 0 \rightarrow 1$  — разрешение внешней среды на обработку сигналов схемой.

**Шаг 2.** Рабочая фаза схемы: обработка входных сигналов и получение выходных, переход  $i = 0 \rightarrow 1$  — индикация окончания переходных процессов в схеме.

**Шаг 3.** Прием внешней средой сигнала окончания обработки, переход  $c = 1 \rightarrow 0$ .

**Шаг 4.** Спейсерная фаза схемы: возврат всех сигналов в спейсерные значения, переход  $i = 1 \rightarrow 0$  — индикация окончания фазы и запрос схемы на новые входные инфосигналы.

Легко заметить, что приведенный процесс полностью аналогичен известному механизму запрос-ответного взаимодействия, широко используемому при асинхронном обмене информацией. Для СС-схем такое взаимодействие является естественным и единственно возможным механизмом, обеспечивающим совокупность их свойств.

Таким образом, СС-схема взаимодействует с окружающей средой по принципу отрицательной обратной связи по фазе работы: получив от схемы сигнал окончания очередной фазы, внешняя среда формирует на входе схемы сигналы противоположной фазы. Такой режим является режимом самогенерации и свойствен СС-схемам при конечном использовании.

Этот же режим, в силу обязательности индикации, является и режимом ожидания всех необходимых переключений элементов схемы и тем самым обеспечивает безопасность ее работы (отсутствие ошибок) при отказах. Действительно, при возникновении константных неисправностей на выходе элементов схема, ожидая переключения, останавливается, самогенерация прекращается, и ошибочная информация не проходит на выход. Остановку работы схемы можно использовать как признак наличия неисправностей для организации их диагностирования, самотестирования и саморемонта.

Из порядка запрос-ответного взаимодействия видно, что внешняя среда взаимодействует с СС-схемой симметрично, по одинаковому механизму. Поэтому внешней средой СС-схемы может быть другая СС-схема, что позволяет строить составные схемы типа конвейеров. Но окружением для СС-схемы могут быть и устройства другой природы, только с обязательной обратной связью. Например, если СС-схема сформировала сигнал для исполнительного механизма — шагового двигателя, то она должна получить в ответ

сигнал, что требуемое перемещение совершено. Тем самым в контур обратной связи включается и исполнительный механизм — внешняя среда, и теперь работа системы СС-схема — шаговый двигатель не зависит от задержек не только элементов СС-схемы, но и шагового двигателя.

Особый случай применения СС-схем — в устройствах, где главным фактором является время, например в таймерах. В них всегда имеется задающий генератор, что накладывает естественное ограничение на величины задержек любой схемы, в том числе и самосинхронной. При выполнении этих ограничений задающий генератор для СС-схемы является элементом внешней среды, обеспечивающим замыкание обратной связи и не влияющим на внутреннюю синхронизацию в СС-схеме.

Значение принципа обратной связи в СС-схемах состоит в организации их естественной циклической работы. Кроме того, в течение каждого цикла, наряду с обработкой информации, происходит попутное тестирование элементов схемы — переключились ли они в очередной фазе, что обеспечивает отказобезопасность схемы.

## 2.2. Самосинхронизирующиеся коды

Самосинхронизирующиеся коды описаны в [3] и предназначены для определения окончания переходных процессов в каком-либо месте схемы чисто логическим способом.

Пусть имеется  $N$  сигналов (переменных)  $z_j$ ,  $j = 1, \dots, N$ , представляющих кодированную информацию. Совокупность значений, принимаемых этими сигналами, называется *набором*. Изменение значений от одного набора к другому называется *переходом*.

Назовем *правильным переходом* из набора  $a$  в набор  $b$  такой переход, когда каждая переменная  $z_j$  изменяется не более одного раза.

При заданных наборах  $a$  и  $b$  рассмотрим множество *переходных наборов*  $[a, b]$ . Оно состоит из наборов  $t$ , таких что:

- 1)  $t \neq a, t \neq b$ ;
- 2) значения  $t$  совпадают со значениями  $a$  и  $b$  в тех разрядах, в которых совпадают значения  $a$  и  $b$ ;
- 3) остальные разряды произвольны.

Например, если  $a = 001$ ,  $b = 100$ , то переходное множество  $[a, b]$  состоит из двух наборов: 101 и 000.

Приведенные определения показывают, что все правильные переходы из  $a$  в  $b$  происходят через переходные наборы.

Из всех  $2^N$  наборов сигналов  $z_j$  рассмотрим подмножество  $Z$  — *кодovou систему* (код).

Назовем *допустимым* переход из набора  $a$  в набор  $b$  в системе  $Z$  при  $a \neq b$ , если выполняются следующие три условия:

- 1)  $a$  принадлежит  $Z$ ,  $b$  принадлежит  $Z(a)$ , где  $Z(a)$  — множество всех наборов из  $Z$ , которые могут появиться вслед за  $a$ ;
- 2) переход из  $a$  в  $b$  — правильный;
- 3) для любого  $t$ , принадлежащего  $[a, b]$ ,  $t$  не принадлежит  $Z(a)$ .

**Основное определение.** Кодовая система  $Z$ , в которой каждый возможный переход — допустимый, называется *самосинхронизирующимся кодом* (СС-кодом).

Таким образом, центральным моментом в определении СС-кода является допустимость переходов.

В первом условии допустимости рассматриваются только наборы, входящие в  $Z$ . Кроме того, выделяется подмножество «последовательностей» набора  $a$ . Именно это подмножество  $Z(a)$  и должно быть использовано для индикации окончания перехода из  $a$  в  $b$ .

Второе условие также необходимо. Представим себе случай, когда некий сигнал  $z_j$  «залипает» на 0, т. е. из-за константной неисправности принимает постоянное значение 0. Если допустить неправильные переходы, то такую ситуацию невозможно отличить от случая, когда сигнал  $z_j$  меняется дважды: из 0 в 1 и из 1 в 0.

Третье условие означает, что если переход из  $a$  в  $b$  происходит через промежуточные наборы, то ни один из этих промежуточных наборов не попадает в множество  $Z(a)$ , по которому оценивается окончание перехода, т. е. гарантируется отсутствие ложной индикации. Промежуточные наборы могут принадлежать или не принадлежать множеству  $Z$ , но обычно они ему не принадлежат.

Важно отметить, что в определении СС-кодов отсутствует понятие времени. Момент окончания перехода однозначно фиксируется при попадании набора в заранее известное множество наборов независимо от любых задержек. В синхронных же, а также в других типах асинхронных схем неизбежны априорные предположения типа: «переход скорее всего (или в большинстве случаев) закончится через столько-то наносекунд».

Определенных таким образом СС-кодов существует множество. Но не все они одинаково пригодны для практической схемотехники. Ряд кодов требует сложных алгоритмов кодирования и декодирования, а для получения сигнала индикации требуется громоздкая схема.

Практическим целям наиболее подходят модификации кодовых систем, получившие название *двухфазных СС-кодов*.

Наборы  $a$  и  $b$ , принадлежащие  $Z$ , называются *сравнимыми*, если один из них можно получить из другого заменой только нулей на единицы или только единиц на нули. Например, наборы 01010 и 11011 — сравнимы, а наборы 01010 и 11000 — несравнимы.

В двухфазных СС-кодах все наборы  $Z$  делятся на две части: наборы  $S$  — *спейсеры* и наборы  $W$  — *рабочие*. Все спейсеры должны быть сравнимы со всеми рабочими наборами, а все рабочие должны быть попарно несравнимы между собой. Рабочие наборы кодируют исходную информацию, а спейсеры играют служебную роль.

В двухфазных СС-кодах изменения кодовых наборов происходят в две фазы. В каждой из них все промежуточные наборы должны быть сравнимы как с начальным, так и с конечным кодом. При этом соблюдаются все условия допустимости переходов и гарантируется отсутствие ложной индикации. Множеством для индикации окончания перехода из рабочей фазы в спейсер служит множество спейсеров  $Z(a) = S$ , а таким же множеством при переходе в рабочую фазу — множество  $Z(a) = W$ .

Хотя возможны коды с несколькими спейсерными наборами, на практике применяют СС-коды с одним спейсером, поскольку в противном случае получаются неоправданно сложные схемы. Поэтому в дальнейшем мы будем рассматривать только коды с одним спейсером.

Рассмотрим некоторые конкретные СС-коды.

Пусть исходная информация представлена  $n$  битами (сигналами). Требуется построить СС-код, кодирующий все  $2^n$  исходных наборов.

#### А. Парафазный код со спейсером (ПФС-код).

В парафазном коде каждый из  $n$  исходных битов кодируется двумя битами кода. Например, 0 кодируется как 01, а 1 — как 10. Роль спейсера играет один из двух наборов — или все нули, или все единицы.

Возьмем пример двухбитовой исходной информации:

Исходные наборы	Кодовые наборы
00	0101
01	0110
10	1001
11	1010

Спейсером в этом примере может служить любой из двух наборов — 0000 или 1111.

Таким образом, кодовая длина ПФС-кода  $N = 2n$ .

**Б. Оптимальный равновесный код (ОРК).**

В равновесном коде все кодовые наборы содержат одинаковое количество нулей и соответственно одинаковое количество единиц. Из соображений оптимальности, число нулей (единиц) в ОРК должно составлять половину (или ближайшее к половине число) длины кода.

Кодовая длина ОРК  $N$  выбирается из условия:

$$C_{N-1}^{(N-1)/2} < 2^n \leq C_N^{N/2}.$$

При  $n = 2$  ОРК совпадает с ПФС-кодом.

Приведем пример ОРК для  $n = 3$ .

Исходные наборы	Кодовые наборы
000	11000
001	01100
010	00110
011	00011
100	10100
101	01010
110	00101
111	10001

Спейсером может служить любой из двух наборов: 00000 или 11111.

**В. Код с идентификатором (КИ).**

В этой кодовой системе применяются наборы, состоящие из  $n$  основных и  $k$  дополнительных разрядов. Число  $k$  равно ближайшему сверху целому к числу  $\log_2(n+1)$ . Основные разряды повторяют исходные, дополнительные (идентификатор) служат для обеспечения свойств СС-кодов.

При  $n = 2$  КИ также совпадает с ПФС-кодом с точностью до перестановки разрядов.

Пример КИ для  $n = 3$ :

Исходные наборы	Кодовые наборы
000	00011
001	00110
010	01010
011	01101
100	10010



Исходные наборы	Кодовые наборы
101	10101
110	11001
111	11100

Спейсером также может служить любой из двух наборов: 00000 или 11111.

### Г. Единичный код (ЕК).

Существует примитивный код, который будем называть *единичным* по аналогии с единичной системой счисления. Такой код (и только он) применяется в NCL-методологии [5].

Исходный объект (сигнал или группа сигналов), принимающий всего  $N$  значений, кодируется  $N$  битами (сигналами). Каждое значение  $i$  ( $0 \leq i \leq N - 1$ ) представляется набором из  $N$  бит, на  $i$ -м месте которого стоит единица, а остальные биты равны нулю. Спейсер кодируется нулевым набором. (По принципу двойственности возможен и второй ЕК с взаимной заменой единиц и нулей.)

В [5] этот код не имеет общего названия и применяется в основном в модификациях при  $N = 2$  («double-rail») и  $N = 4$  («quad-rail»).

Пример ЕК для  $N = 4$ :

Исходные значения	Кодовые наборы
0	1000
1	0100
2	0010
3	0001

Код удовлетворяет всем требованиям самосинхронности. При  $N = 2$  он совпадает с ПФС-кодом. При больших  $N$  код ЕК характеризуется чрезвычайной громоздкостью, в чем можно убедиться, посмотрев в [5] пример для  $N = 16$ .

### Д. Другие виды кодов.

В ряде случаев могут применяться и другие коды, удовлетворяющие условиям двухфазных СС-кодов. Например, для схемы распределения четырех сигналов можно построить такой «сдвиговой» код с любым спейсером — нулевым или единичным:

Исходные значения	Кодовые наборы
0	1100
1	0110

Исходные наборы	Кодовые наборы
2	0011
3	1001

### Е. Сравнение самосинхронизирующихся кодов.

Как видно из предыдущего, длина любого СС-кода всегда больше длины исходных наборов, минимально необходимой для представления информации. Другими словами, все СС-коды избыточны по кодовой длине. Теоретически доказано в [3], что наименее избыточен ОРК. Наиболее избыточными являются ЕК, затем ПФС, затем остальные коды.

Оценим применимость различных СС-кодов для построения СС-схем. Основным критерием сравнения будет сложность схем, под которой понимается общее число транзисторов, а также связей между ними. Зависимость быстродействия схем от применяемого кода оценить трудно из-за опосредованного характера такой зависимости.

Главным достоинством ОРК является его минимальная избыточность. Это дает минимальное количество связей между частями схемы по сравнению с другими кодами. Однако этот код имеет серьезный схемотехнический недостаток — потерю структурности — поразрядной связи между исходной информацией и кодом. Как, например, выделить в ОРК младший разряд для округления или старший разряд для переноса? Ответ один: необходимо строить специальные схемы.

Рассмотрим пример исходных некодированных связей между элементами — прототипа, который требуется реализовать в СС-схеме с помощью ОРК (рис. 2.6).

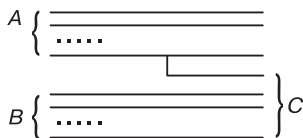


Рис. 2.6. Исходные связи (прототип)

В закодированном виде таких связей, как на рис. 2.6, не будет. В СС-схеме, реализующей прототип, необходимо:

- из кодированного вектора  $A_{\text{ОРК}}$  выделить (декодировать) один исходный сигнал с помощью специальной схемы;
- декодировать вектор  $B_{\text{ОРК}}$  с помощью специальной схемы;
- закодировать исходный вектор  $C$  в ОРК с помощью специальной схемы.

Приведенный пример является простейшим. В реальных схемах такие и еще более сложные ситуации будут встречаться постоянно.

Отметим также, что упомянутые специальные схемы кодирования-декодирования тоже должны быть самосинхронными, что еще более усложняет задачу.

Таким образом, минимальная избыточность ОРК с лихвой перекрывается сложностью получаемых схем.

ЕК, хотя и обладает некоторой структурностью (в частности, младший разряд выделить просто, а старший — наоборот, сложно), но из-за большой избыточности его применение представляется нецелесообразным.

В ПФС-коде сохраняется полная связь с прототипом — структурность. Это преимущество, несмотря на избыточность, оказалось решающим для того, чтобы именно этот код использовать при построении практических СС-схем.

КИ занимает промежуточное положение между ОРК и ПФС-кодом. Для него связь с прототипом сохраняется (по основным разрядам). Однако свойством структурности он не обладает из-за дополнительной части кода — идентификатора. Этот код в силу сложности построения идентификаторов также не находит применения на практике.

Однако и ОРК, и КИ имеют свою нишу в СС-схемах — там, где малая избыточность оказывается решающей. Это, например, обмен по шинам. В вычислительной технике существует устойчивая тенденция к увеличению размера шин. Не редкость уже шины в 128 и 256 разрядов; видимо, и в дальнейшем они будут расти.

Количество трасс здесь становится ограничивающим фактором, и в случае использования СС-схем преимущества КИ и ОРК очевидны. Ниже дано сравнение основных кодов по длине.

Исходный код (бит)	ПФС (бит)	КИ (бит)	ОРК (бит)
32	64	38	35
64	128	71	68
128	256	136	132
256	512	265	261
512	1024	522	517
1024	2048	1035	1030

Число битов КИ и ОРК лишь незначительно увеличивается по сравнению с исходным кодом.

В практической СС-схемотехнике, не считая шин, удобнее всего использовать ПФС-код.

### Ж. Коды в изменениях.

Коды в изменениях представляют собой пример приспособления СС-кодов для практики с целью избавиться от их недостатков.

При передаче по шине наборов в СС-кодах необходимо последовательно передавать и рабочие наборы, и спейсеры. Спейсер не несет информации, и его передача составляет системные издержки. Идея состоит в том, чтобы передавать по шине не сами СС-наборы, а их изменения. Так часто поступают и в обычных шинах и каналах связи. При передаче СС-кодов таким способом удастся избежать передачи спейсера.

Для передачи изменений на передающем и приемном концах шины ставится по регистру памяти и схеме преобразования (поразрядное суммирование по модулю 2).

Платой за выигрыш времени передачи СС-кодов являются задержки в передающих и приемных преобразователях. Обозначим задержку передачи одного набора по шине как  $T_{ш}$ , задержки преобразователей передачи и приема соответственно  $T_{пер}$  и  $T_{пр}$ . Тогда цикл передачи одного набора в СС-коде (рабочий набор и спейсер) будет  $2T_{ш}$ , а в изменениях —  $T_{ш} + T_{пер} + T_{пр}$ . Сравнивая задержки, находим, что коды в изменениях более эффективны при больших задержках шины при условии:

$$T_{ш} > T_{пер} + T_{пр}.$$

Сами коды в изменениях не являются СС-кодами. Однако они легко приводятся к СС-кодам с учетом *переменных памяти* в регистрах. Самосинхронность при этом обеспечивается схемотехнически.

### 3. Пример простой схемы в ПФС-коде.

На рис. 2.7 приведен пример схемы исключаящего ИЛИ (суммы по модулю 2) с индикацией выходных сигналов. Следует заметить, что входные сигналы являются выходами других элементов, не показанных на схеме, и тоже должны где-то индцироваться.

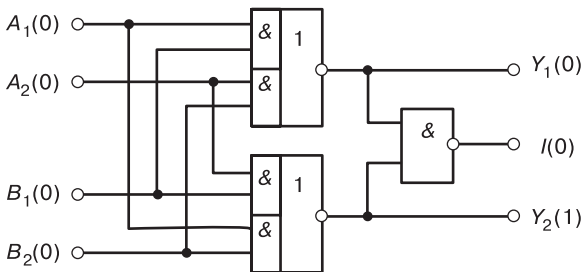


Рис. 2.7. Схема исключаящего ИЛИ в ПФС-коде

Первые два элемента схемы реализуют исходную функцию в ПФС-коде, третий осуществляет индикацию выходов.

В таких схемах обычно в скобках после имен переменных записывают значения их спейсеров (таков же смысл цифр в скобках и на рис. 1.2). Как видно, в разных частях схемы могут быть разные спейсеры. Сигналы  $A_1, A_2, B_1, B_2, Y_1$  и  $Y_2$  — парафазные (закодированные в ПФС-коде), сигнал  $I$  — индикаторный.

Таблица истинности этой схемы:

Исходные коды	Входы в ПФС		Выходы в ПФС	Индикатор
	$A_1, A_2$	$B_1, B_2$	$Y_1, Y_2$	
$AB$				$I$
00	01	01	01	1
01	01	10	10	1
10	10	01	10	1
11	10	10	01	1
(спейсер)	00	00	11	0

Значение  $I = 0$  показывает, что выходные сигналы перешли в спейсер ( $Y_1 = Y_2 = 1$ ), значение  $I = 1$  — что они находятся в рабочем состоянии.

Единичное состояние входов  $A_1 = A_2 = B_1 = B_2 = 1$  при нормальном функционировании — запрещенное, оно не соответствует ни рабочему состоянию, ни спейсеру, ни промежуточным значениям. Соответственно, в исправной схеме значения  $Y_1 = Y_2 = 0$  никогда не возникают.

## 2.3. Основа теории СС-схем — модель и теория Маллера

Теория асинхронных схем Маллера [1] разрабатывалась обобщенно, для  $k$ -значных элементов любой физической природы. Нас будет интересовать ее применение только к цифровым электронным схемам двузначной логики ( $k = 2$ ). В этом мы будем следовать также ее интерпретации по Варшавскому [3].

Важность модели Маллера для самосинхронной схмотехники состоит в том, что на ее основе удастся точно сформулировать условия самосинхронности и создать методы анализа и синтеза СС-схем.

### 2.3.1. Модель Маллера и независимость от задержек

Условием применения модели считается то, что каждый логический элемент схемы имеет один выходной сигнал.

*Моделью Маллера* называется система булевых уравнений, описывающих выходные сигналы элементов:

$$z_i^+ = f_i(z_1, \dots, z_i, \dots, z_N), \quad i = 1, \dots, N. \quad (2.1)$$

В левой части уравнений стоят будущие значения переменных (выходов элементов). Для каждой переменной, присутствующей в правых частях равенств, должно быть записано одно и только одно уравнение с этой переменной в левой части. Тем самым не допускаются монтажные соединения нескольких выходов — наличие нескольких уравнений с одинаковой левой частью.

В существующей литературе моделью Маллера обычно описывают замкнутые схемы, и может показаться, что только такие схемы и могут быть представлены его моделью. Однако это не так. Разомкнутые схемы тоже описываются моделью Маллера, и на них тоже распространяется его теория. Например, разомкнутый инвертор, показанный ниже на рис. 2.14, описывается уравнениями:

$$\begin{aligned} z_1^+ &= z_1, \\ z_2^+ &= \hat{z}_1. \end{aligned} \quad (2.2)$$

Здесь и далее символом « $\hat{\phantom{x}}$ » обозначается операция отрицания.

Первое уравнение в (2.2) показывает, что вход инвертора не меняется (является константой). Второе уравнение — обычное уравнение инвертора.

Множество  $A = a_1, \dots, a_N$  двоичных значений переменных  $z_1, \dots, z_N$  называется *состоянием* схемы.

Если в состоянии  $A$  для переменной  $z_i$  выполняется условие  $a_i = f_i(A)$ , то будем говорить, что эта переменная и соответствующий ей элемент *устойчивы* в состоянии  $A$ . Если в результате изменения входов элемента выполняется условие  $a_i \neq f_i(A)$ , то элемент *возбужден* в этом состоянии. В возбужденном элементе начинается процесс перехода выходного сигнала от одного значения к противоположному (процесс перезаряда нагрузки).

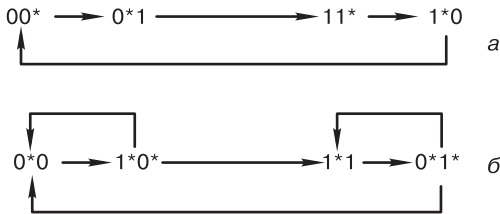
Задержки элементов, таким образом, учитываются в теории Маллера путем введения понятия возбуждения.

Состояние возбуждения элемента может прекратиться двумя путями:

- 1) выход элемента перешел в новое значение; это ожидаемая ситуация срабатывания элемента;
- 2) выход элемента не успел измениться, а входы изменились так, что опять  $a_i = f_i(A)$ ; это непредвиденное развитие событий порождает состязания входов элемента.

Сами длительности возбуждений (величины задержек) элементов никак не фигурируют в модели. Важен только факт возникновения и прекращения возбуждения. Поэтому модель описывает элементы с произвольными задержками.

Одним из наглядных способов представления работы модели Маллера является *диаграмма переходов* (ДП). Она изображается в виде направленного графа с состояниями в узлах. Если какая-либо переменная возбуждена в некотором состоянии, то ее значение в этом состоянии помечается звездочкой справа. На рис. 2.8 показаны примеры ДП для схемы из двух элементов.



**Рис. 2.8.** Примеры диаграмм переходов:  
*a* — полумодулярная; *b* — неполумодулярная с двумя конфликтными переходами

Переход в диаграмме называется *конфликтным*, если хотя бы одна переменная в исходном состоянии (до перехода) помечена звездочкой, а после перехода имеет такое же значение, но без звездочки. Конфликтный переход как раз соответствует второму пути снятия возбуждения элемента (состязанию).

Диаграмма переходов называется *полумодулярной* (*ПМ-диаграммой*), если она не содержит конфликтных переходов.

Термин «полумодулярный» происходит из теории алгебраических структур — решеток, использованной в работе [1]. Не углубляясь в теорию, можно дать следующие нестрогие пояснения термина. Каждому уравнению (элементу) схемы ставится в соответствие число — счетчик переключений. При возбуждении элемента счетчик увеличивается на единицу, при снятии возбуждения из-за изменения входов (по второму пути) счетчик уменьшается на единицу. При ожидаемом переходе (по первому пути) счетчик не изменяется. Совокупность всех счетчиков схемы и образует алгебраический объект — решетку. Свойство полумодулярности решетки означает, что ни одно число из совокупности в процессе переключений не уменьшается. Это и свидетельствует об отсутствии конфликтных переходов.

Для одной и той же модели вид ДП и свойство полумодулярности зависят от начальных значений переменных. Поэтому о полумодулярности модели (2.1) можно говорить только относительно заданного *начального состояния*.

Схема, описываемая системой (2.1), называется *полумодулярной при данном начальном состоянии*, если в этом случае она описывается ПМ-диаграммой.

Маллером и Бартки доказано, что полумодулярная схема является независимой от задержек элементов. Ими же установлено, что поведение таких схем развивается по одному из двух сценариев.

По первому схема, пройдя из начального состояния через несколько промежуточных, останавливается в состоянии, называемом *тупиковым*.

Во втором сценарии схема проходит несколько неповторяющихся состояний (*начальный участок*), а затем бесконечно проходит по состояниям из некоторого циклического множества — *финального цикла* (включающего не менее двух состояний).

Главный смысл применения на практике модели Маллера и определения полумодулярности состоит в том, что процессы, описываемые ПМ-диаграммами, и не зависят от задержек элементов, и не имеют состязаний.

Итак, если электронную схему при определенных условиях можно описать ПМ-диаграммой, то при этих условиях она будет свободной от состязаний и ее работа не будет зависеть от задержек элементов. Многие специфические методы проектирования СС-схем базируются на модели Маллера и свойстве полумодулярности.

### **2.3.2. Применимость модели Маллера для элементов и их соединений**

В современной электронной технике применяются весьма разнообразные электронные компоненты и схемные решения, не все из которых допускают описание моделью Маллера. Поэтому целесообразно сформулировать в явном виде требования, позволяющие применять эту модель.

Требования и условия применимости модели Маллера к электронным схемам таковы.

1. Значения, принимаемые переменными (сигналами), — только двоичные. Высокий импеданс (третье состояние) и другие значения расширенных алфавитов не допускаются.



2. Каждый элемент схемы может иметь несколько входов и только один выход. Не допускается объединение выходов, так как в системе (2.1) не может быть нескольких уравнений с одинаковыми левыми частями.

3. Элементы в модели — пассивные, их выходы могут меняться лишь в ответ на возбуждение, которое может возникнуть только при изменении входов. При этом одно возбуждение может породить только одно изменение выхода.

4. Входные задержки элементов отсутствуют, т. е. возбуждение любого элемента начинается сразу, как только на его входах появляется комбинация значений, приводящая к изменению выхода. При необходимости учета входной задержки ее описывают внешним элементом — повторителем.

5. Задержка выходного сигнала считается задержкой всего элемента.

6. Задержки в соединениях элементов (проводах) после разветвлений считаются нулевыми (или пренебрежимо малыми по сравнению с задержками соответствующих элементов-последователей).

Особо следует остановиться на последнем требовании. Внутри кристаллов большой и сверхбольшой степеней интеграции и особенно при соединении кристаллов (микросхем) между собой оно может нарушаться. В подобных случаях в модели Маллера необходимо вводить повторители, которые имитируют задержки. Однако после введения повторителей полумодулярность исходной схемы, как правило, нарушается.

Задержки в проводах после разветвлений составляют одну из проблем создания СС-схем на этапе перехода от логических описаний к топологии. Тем не менее с учетом топологической реализации существуют способы, и довольно простые, получения СС-схем с независимостью от задержек не только в элементах, но и в проводах после разветвлений. Этот вопрос обсуждается далее в книге.

## 2.4. Базовые элементы СС-схем

Один из первых вопросов, возникающих при создании СС-схем, — на каких элементах их можно корректно реализовать. Знание базиса реализации необходимо на самых ранних этапах проектирования, поскольку свойство самосинхронности существенно зависит от каждого элемента схемы. Оно также важно для разработки библиотек

как базовых матричных кристаллов (БМК), так и заказных элементов для СС-схем.

Вопрос базиса можно рассматривать с двух точек зрения — логической и практической.

По логическому базису для полумодулярных схем (а значит, и для СС-схем) установлено следующее [14, 3].

1. Они могут быть корректно реализованы в неограниченном базисе И-НЕ (ИЛИ-НЕ), а также в минимальном базисе 2И-НЕ (2ИЛИ-НЕ) с коэффициентом разветвления по выходу, равным 2.

2. Они могут быть корректно реализованы в неограниченном базисе И-ИЛИ-НЕ (ИЛИ-И-НЕ).

Если иметь в виду, что любая функция представима в базисе И-ИЛИ-НЕ (ИЛИ-И-НЕ), т. е. в дизъюнктивной (конъюнктивной) нормальной форме (ДНФ, КНФ), то, с логической стороны, для реализации СС-схем нет никаких ограничений.

С практической же точки зрения, необходимо решить, какой сложности и какого типа элементы следует использовать для реализуемых схем. Такое исследование опубликовано в работах [15, 16].

Одно и то же уравнение, входящее в модель Маллера, можно реализовать по-разному. Например, уравнение И-НЕ с четырьмя входами можно реализовать одним элементом 4И-НЕ, а можно тремя элементами 2И-НЕ и инвертором. Эти реализации функционально эквивалентны, и формально условия модели Маллера соблюдены в обоих случаях. Однако во второй реализации возможны состязания типа короткого импульса на выходе. Такая ситуация недопустима для СС-схем и возникает из-за некорректности выбора базиса. Естественно потребовать, чтобы элементы, описываемые одним уравнением Маллера, были в определенном смысле простыми, а не составными.

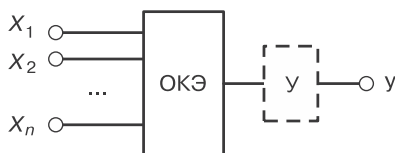
### 2.4.1. Общая структура базовых элементов для СС-схем

В любом из основных типов интегральных технологий (ТТЛ, ЭСЛ, хМДП) существуют *однокаскадные элементы*. Под однокаскадным понимается элемент, ток нагрузки которого переключается транзисторами, чьи управляющие электроды непосредственно соединены со входами элемента. Однокаскадный элемент описывается одним логическим уравнением и удовлетворяет всем условиям применимости модели Маллера. Такой элемент невозможно разбить на составные части, которые могли бы быть описаны логическими уравнениями. Внутренние состязания в однокаскадном элементе не возникают.

К однокаскадным элементам относятся, в частности, инверторы и повторители (но не их последовательные цепочки).

Часто для увеличения нагрузочной способности к выходам однокаскадных элементов подключают инверторы и/или повторители. Такое подключение не нарушает применимости модели Маллера, поскольку в соответствующем уравнении оно (подключение) определяет только общее отрицание (или его отсутствие) всей правой части. Оно также не вносит дополнительных соизязаний и потому может быть допущено в базовые элементы для СС-схем.

Назовем *самосинхронным базовым элементом (ССБЭ)* однокаскадный элемент (ОКЭ), на выходе которого может быть последовательно подключено любое количество инверторов и/или повторителей (усилительная часть). Структура ССБЭ показана на рис. 2.9.



**Рис. 2.9.** Структура самосинхронного базового элемента: ОКЭ — однокаскадный элемент;  $Y$  — необязательная усилительная часть

Общим свойством ССБЭ является то, что они выполняют либо *изотонные*, либо *антитонные* логические функции по всем переменным. (Признаком изотонной функции является существование ее ДНФ без вхождений переменных с отрицаниями, а антитонной — ДНФ со всеми вхождениями переменных с отрицанием.) Оба этих вида функций будем для краткости называть *монотонными*. Данное свойство монотонности является достаточным для базиса построения СС-схем, так как в общем случае [3] требуется только монотонность по каждой переменной в отдельности.

Формально ССБЭ, имеющий усилительную часть, описывается двумя логическими уравнениями. Но условия модели Маллера требуют выполнения правила: один элемент — одно уравнение. Уравнения ССБЭ с усилительной частью легко сводятся к одному уравнению путем подстановки. Это либо уравнение ОКЭ, если в цепочке усилительной части содержится четное число инверторов, либо уравнение ОКЭ с общим отрицанием, если число инверторов нечетное. Поэтому в дальнейшем будем всегда считать, что каждый ССБЭ описывается одним логическим уравнением.

Возможные множества ССБЭ будут определяться технологией изготовления и свойственными ей схмотехническими ограничениями.

Мы рассмотрим только элементы КМДП-технологии как наиболее распространенной в настоящее время (основные сведения о ней приведены, например, в [17]).

### 2.4.2. Самосинхронные базовые элементы КМДП-технологии

Основными характеристиками элементов, используемыми в дальнейшем, будут количество транзисторов в элементе (т. е. затраты) и быстродействие.

Назовем *индексом задержки (ИЗ)* максимальное число последовательно соединенных транзисторов в цепи перезаряда нагрузки элемента.

Как показано в [18], задержку однокаскадного КМДП-элемента можно оценить двумя слагаемыми — нагрузочным и внутренним (паразитным):

$$T = r_l I C_l + r_s (I + I^2) C_s, \quad (2.3)$$

где  $r_l$  и  $r_s$  — нагрузочный и паразитный коэффициенты соответственно (зависящие от реализации);  $I$  — индекс задержки;  $C_l$  — емкость нагрузки;  $C_s$  — средняя паразитная емкость внутренних точек соединений транзисторов в элементе.

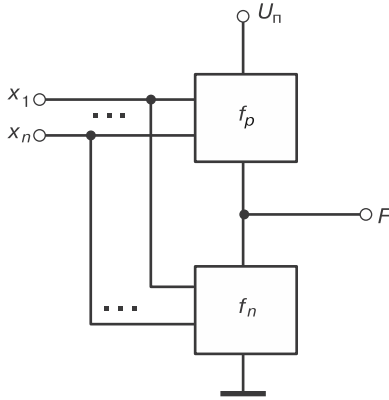
Как видно из (2.3), структурным параметром, определяющим быстродействие однокаскадного элемента, является индекс задержки. Этот параметр и будет использоваться как характеристика быстродействия элемента: чем больше ИЗ, тем медленнее элемент, и наоборот.

При малых ИЗ вторым слагаемым в (2.3) можно пренебречь [18] из-за малости паразитных параметров. С увеличением ИЗ паразитная составляющая (2.3) быстро растет, и элемент становится неоптимальным по быстродействию: его целесообразнее заменить двумя-тремя элементами с меньшей суммарной задержкой. По этой причине на практике однокаскадные элементы имеют  $ИЗ \leq 4$ .

Исходя из структуры ССБЭ, приведенной на рис. 2.9, можно выделить два их подкласса: вентильные и запоминающие.

#### 2.4.2.1. Вентильные ССБЭ

Однокаскадный элемент вентильного ССБЭ представляет собой обычный вентиль типа push-pull, в каждом плече которого находятся параллельно-последовательные цепочки транзисторов, управляемых входными сигналами. Структуру ОКЭ можно представить в виде, показанном на рис. 2.10.



**Рис. 2.10.** Однокаскадный элемент вентиляющего ССБЭ

На этом рисунке буквами  $p$  и  $n$  обозначены цепи, содержащие, соответственно,  $p$ - и  $n$ -транзисторы.

Для описания транзисторной структуры элемента удобно пользоваться символическими функциями  $f_p$  и  $f_n$ . Символическая функция — это логическая функция без отрицаний, показывающая соединение транзисторов: логическое И означает последовательное, логическое ИЛИ — параллельное соединение.

Здесь и далее постфиксным символом «\*» будет обозначаться двойственная логическая функция.

Для вентиля должно выполняться условие:

$$f_n = f_p^* \quad (2.4)$$

Данное условие обеспечивает как отсутствие сквозных токов, так и отсутствие третьего состояния (высокого импеданса) на выходе.

При этом условии функция, выполняемая ОКЭ, равна

$$F = \hat{f}_n \quad (2.5)$$

В зависимости от усилительной части каждый ОКЭ может породить два ССБЭ — с функцией  $F$  и с инверсной ей функцией.

Рассмотрим вопрос о множестве вентилях с  $ИЗ \leq 4$ , которое определит и множество возможных вентиляльных ССБЭ. Расчеты на ЭВМ показали, что число таких вентилях очень велико: найдено более 250 тысяч, а общее число должно быть еще больше.

Представление о числе вентиляльных ССБЭ могут дать табл. 2.1—2.3 количеств порождающих функций. Каждая порождающая функция  $f$  дает два вентиля, т. е. четыре ССБЭ со следующими функция-

ми:  $f, \hat{f}, f^*, \hat{f}^*$  (очень редкие самодвойственные функции порождают два элемента).

В таблицах обозначено:

$N$  — число аргументов (входов) — верхняя строка;

$M$  — число импликант в ДНФ функции — левый столбец;

$X$  — функции предположительно есть, но пока не найдены.

Таблица 2.1

### Число всех порождающих функций с ИЗ = 2

$M \backslash N$	2	3	4
1	1		
2		1	1
3		1	1

Таблица 2.2

### Число порождающих функций с ИЗ = 3

$M \backslash N$	3	4	5	6	7	8	9	10
1	1							
2		3	2	1				
3		3	13	13	7	2	1	
4		2	26	82	81	33	5	
5			24	267	434	200	30	1
6			10	490	1625	713	88	1
7			4	717	3130	1543	193	X
8			1	816	4691	2343	X	X

Таблица 2.3

### Число порождающих функций с ИЗ = 4

$M \backslash N$	4	5	6	7	8	9	10	11	12
1	1								
2		4	3	2	1				
3		7	37	47	38	17	7	2	1
4		9	177	613	982	960	614	282	99
5		6	590	6045	20 060	32 820	33 110	X	X
6		1	1367	23 397	44 394	X	X	X	X
7		1	2224	32 768	X	X	X	X	X
8			2616	61 963	X	X	X	X	X

Для примера приведем все порождающие функции с ИЗ = 2:

$$\begin{aligned}
 F_1 &= AB, \\
 F_2 &= A \vee BC, \\
 F_3 &= AB \vee AC \vee BC, \\
 F_4 &= AB \vee CD, \\
 F_5 &= AB \vee AC \vee BD.
 \end{aligned}$$

В частности, функция  $F_2$  порождает следующие функции ССБЭ:

$$\begin{aligned}
 f_1 &= a \vee bc, \\
 f_2 &= ab \vee ac, \\
 f_3 &= \hat{a} \vee \hat{b} \hat{c}, \\
 f_4 &= \hat{a} \hat{b} \vee \hat{a} \hat{c}.
 \end{aligned}$$

Приведенные данные показывают, что число возможных вентиляльных ССБЭ очень велико, и этот логический базис достаточно широк для построения СС-схем.

#### 2.4.2.2. Запоминающие ССБЭ

На данный тип элементов для теоретически корректной реализации СС-схем обращено внимание в [16]. Хотя в зарубежной литературе известна реализация  $S$ -элемента Маллера, относящаяся к данному типу, в публикациях группы Варшавского такого рода реализации отсутствуют, а функциональным эквивалентом  $S$ -элемента служит  $\Gamma$ -триггер на вентильных элементах (см. ниже). В публикациях последних лет по NCL-методологии [5] такие элементы применяются, но «интуитивно», без теоретического обоснования.

Запоминание в таких элементах происходит во внутренней точке. Структура элемента показана на рис. 2.11, в которой усилительная часть представлена  $Q$ -элементом.

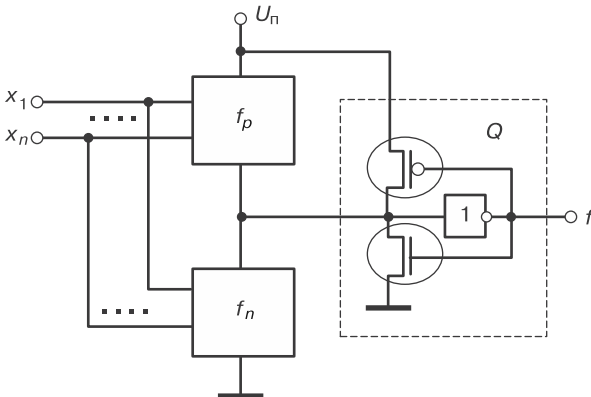


Рис. 2.11. Запоминающий ССБЭ

$Q$ -элемент представляет собой обычный инвертор, в обратной связи которого стоит «слабый» инвертор. Транзисторы в кружках на рисунке — «слабые», т. е. имеют в открытом состоянии сопротивление, в несколько раз большее аналогичного сопротивления обычного транзистора. Элемент предназначен для компенсации токов утечки на входе этого элемента, т. е. сохранения потенциала внутренней точки в случаях, когда со стороны ОКЭ устанавливается высокий импеданс. В остальных случаях элемент работает как инвертор.

Поскольку в элементе существенно используется третье состояние в средней точке, ни однокаскадная часть, ни  $Q$ -элемент по отдельности не описываются логическими уравнениями. Однако элемент в целом имеет логическое описание, удовлетворяющее требованиям модели Маллера.

Для нормальной работы запоминающего ССБЭ необходимо обеспечить выполнение двух условий: отсутствие сквозного тока и наличие третьего состояния:

$$f_n \wedge f_p^* = 0, \quad \wedge f_n f_p^* \neq 0. \quad (2.6)$$

При этих условиях элемент выполняет триггерную функцию

$$f = f_n \vee qf_p^*, \quad (2.7)$$

где  $q$  — предыдущее значение  $f$ .

Условия (2.6) — более слабые, чем условие (2.4), поэтому возможное количество запоминающих ССБЭ будет больше, чем вентилярных, т. е. тоже очень велико. Пример двух простейших элементов этого типа приведен на рис. 2.12.

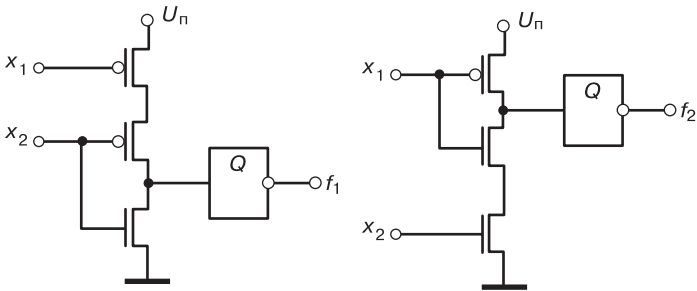


Рис. 2.12. Простейшие запоминающие ССБЭ

Функции этих элементов имеют вид:

$$f_1 = x_2 \vee qx_1, \quad f_2 = x_1x_2 \vee qx_1.$$



Другие примеры ССБЭ этого типа приведены в главе 4.

Анализ других возможностей базовых элементов со структурой, приведенной на рис. 2.9, показывает, что двумя рассмотренными вариантами эти возможности исчерпываются. Таким образом, существует только два типа базовых элементов КМДП-технологии для СС-схем, удовлетворяющие условиям применимости модели Маллера, — вентильные и запоминающие на внутренней точке.

## 2.5. Событийный и функциональный подходы в самосинхронике

Традиционно, начиная с работ Маллера, в литературе по самосинхронике господствует подход, пришедший из математики и теории автоматов. Основная особенность этого подхода — описание работы СС-схем как процесса изменений (переключений) выходных значений элементов — *событий*. Соответственно и для представления работы схем применяются диаграммы переходов (как в теории Маллера), диаграммы изменений, сети Петри, сигнальные графы и другие графические представления, удобные для отображения событий. В связи с этим данный подход будем называть в дальнейшем *событийным*.

Другая особенность событийного подхода — замкнутость исследуемых схем. Она естественно необходима как нормальный режим функционирования СС-схем.

Однако проектирование СС-схем, как и любых других цифровых схем, первоначально происходит в иных представлениях. Схемы практической сложности разрабатываются по частям. Схема целиком и ее фрагменты являют собой устройства преобразования входных данных в выходные. Наличие входов и выходов для них является неотъемлемым условием, и разомкнутое их представление неизбежно. Но в рамках традиционного событийного подхода каждый фрагмент и схему в целом требуется представлять в замкнутом виде для имитации поведения внешней среды. Поэтому приходится неоднократно выполнять подходящее замыкание и размыкание как фрагментов схемы, так и схемы целиком.

Для большинства практических разработчиков электронных схем такой подход и необычен, и сложен из-за сложности математического аппарата, необходимости замыкания-размыкания и других трудностей. Например, оптимизировать схему по быстродействию или затратам, имея только событийное описание, невозможно.

В этих условиях представляется весьма полезной возможность проектирования СС-схем в категориях, привычных для большинства разработчиков обычных схем. С этой целью был предложен [19] и развит *функциональный подход*.

В функциональном подходе приняты следующие основные принципы:

1) рассматриваются разомкнутые схемы, имеющие входы и выходы;

2) элементы схем описываются в виде логических функций и уравнений, событийные описания не привлекаются;

3) поведение схемы на любом уровне иерархии рассматривается в связи с поведением ее окружения, но без явного замыкания.

Замыкание, вообще говоря, потребуется только для схемы в целом и только на конечном этапе, при анализе ее работы в целевом изделии.

В дальнейшем оказалось, что функциональный подход открывает в проектировании СС-схем также новые, недоступные при событийном подходе возможности, которые описаны далее.

### 2.5.1. Типы сигналов и интерфейс СС-схем в функциональном подходе

По функциональному назначению сигналы СС-схем можно разделить на три группы: информационные, контрольные и вспомогательные.

**Информационные сигналы (инфосигналы)** представляют информацию, обрабатываемую в схеме. К ним относятся сигналы, кодированные в СС-кодах со спейсером, и некодированные сигналы.

Частным, но наиболее важным случаем кодированных информационных сигналов являются сигналы в ПФС-коде — *ПФС-сигналы*.

К группе некодированных информационных сигналов относятся бистабильные (*БС-сигналы*) — выходы бистабильных ячеек и *унарные* (одионочные) инфосигналы. Последние по поведению в СС-схемах аналогичны бистабильным сигналам, поэтому условно будем называть их *унарными стабильными* — *УС-сигналами*.

УС-сигналом, как правило, может быть либо независимый сигнал из несамосинхронного окружения, либо один из пары БС-сигналов. Для использования в СС-схемах эти сигналы должны быть сразу преобразованы в парные сигналы, т. е. в ПФС- или БС-сигналы.

Существующие в схемотехнике многофазные и многостабильные сигналы — выходы соответствующих ячеек — могут быть учтены аналогично, но для простоты рассматриваться не будут.

**Контрольные сигналы** предназначены для организации правильных переходов из одной фазы в другую. К ним относятся индикаторные (*И-сигналы*) и управляющие (*У-сигналы*). И-сигналы показывают завершение перехода соответствующей части схемы в очередную фазу — рабочую или спейсер. У-сигналы управляют переключениями и обеспечивают правильность изменений сигналов в необходимых точках схемы при фазовых переходах. Обычно И-сигналы в одной части схемы являются У-сигналами для другой части схемы.

Контрольные и ПФС-сигналы, в силу их природы, имеют одно из значений спейсера — 0 или 1 и показывают текущую фазу работы. Поэтому вместе они называются *фазовыми сигналами*. В отличие от них выходы бистабильных ячеек — БС-сигналы, а также УС-сигналы — не характеризуются каким-либо значением спейсера.

**Вспомогательные сигналы** предназначены для вспомогательных целей, например предустановки внутренних устройств, задания режима и др.

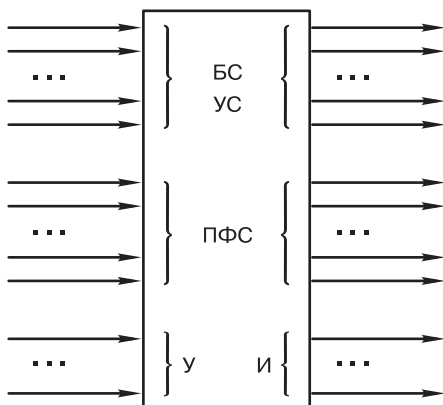
Считается, что СС-схема подключена к внешним окружающим СС-схемам, поэтому ее интерфейс (входы и выходы) должен иметь определенные свойства.

1. На входах схемы могут присутствовать сигналы типов БС, УС, ПФС и У. При этом необходимо наличие хотя бы одного фазового сигнала для организации правильных переходов внутри схемы.

2. На выходах схемы могут присутствовать сигналы типов БС, УС, ПФС и И. При этом также обязательно наличие хотя бы одного фазового сигнала для указания внешним схемам текущей фазы данной схемы.

3. Вспомогательные сигналы могут присутствовать и на входах, и на выходах. Далее для простоты эти сигналы показываться не будут.

На рис. 2.13 приведен типовой интерфейс СС-схемы в функциональном подходе.



**Рис. 2.13.** Типовой интерфейс разомкнутой СС-схемы (вспомогательные сигналы не показаны)

### 2.5.2. Индицируемость — необходимое свойство разомкнутых СС-схем

Есть два схожих по названию термина для СС-схем: индикация и индицируемость. Индикация, как указывалось ранее, — это способ определения окончания переходных процессов в каком-либо фрагменте схемы.

*Индицируемость* сигналов — это свойство СС-схем, обеспечивающее правильность смены фаз и отказобезопасность (остановку схем при возникновении неисправностей).

Индицируемость — одно из основных понятий при проектировании СС-схем, впервые введенное в [3] для разомкнутых схем. В последующих публикациях группы Варшавского понятие индицируемости непосредственно не использовалось, поскольку авторы развивали событийный подход в применении к замкнутым схемам.

Приведем в менее формализованном виде следствия из теории, касающиеся индицируемости [3], важные как для функционального подхода, так и для понимания работы СС-схем.

Пусть задана разомкнутая СС-схема и все требования теории для нее соблюдены.

В условиях двухфазной дисциплины индицируемость сигналов необходимо рассматривать отдельно в каждой фазе работы — спейсере и рабочей. В схеме может содержаться множество сигналов, которые меняются только в одной из фаз. В другой фазе, поскольку они не меняются, говорить об их индицируемости не имеет смысла.

*Будем говорить, что некоторый сигнал  $X$  индицируется сигналом  $Y$  (или на сигнале  $Y$ ), если оба сигнала в данной фазе меняются, а изменение сигнала  $Y$  происходит только при условии изменения сигнала  $X$ .*

Наглядно индицируемость некоторого сигнала можно представить следующим образом. Рассмотрим состояние схемы в такой фазе и с такими значениями других сигналов, что данный сигнал должен измениться. Задержим это изменение до тех пор, пока завершатся другие переходные процессы в схеме (ситуация законная, так как задержки в СС-схеме могут быть любыми). Далее позволим выбранному сигналу измениться. Если в ответ на такое изменение изменятся какие-либо фазовые сигналы, то это значит, что эти фазовые сигналы индицируют данный сигнал.

Остановимся на особенностях индицируемости.

1. Индицируемость обладает свойством транзитивности: если сигнал  $A$  индицируется на  $B$ , а  $B$  — на  $C$ , то  $A$  индицируется на  $C$ . Например, вход инвертора всегда индицируется на выходе, что легко проверить. То же относится и к любой последовательной цепочке инверторов и повторителей: все сигналы цепочки индицируются на ее выходе. Это свойство, например, позволяет нам подсоединять или исключать усилительную часть базовых элементов.

2. Благодаря транзитивности на одном сигнале может индицироваться любое количество других сигналов. Например, типичным является случай, когда один сигнал индицирует все остальные сигналы СС-схемы. В схеме на рис. 2.3, в таким сигналом является сигнал  $i$ , индицирующий остальные ее сигналы.

3. Для схем, имеющих типовой интерфейс, выходными индицирующими сигналами могут быть только фазовые сигналы, поскольку только они меняются в каждой фазе.

*Будем говорить, что схема имеет свойство индицируемости, если в каждой из фаз все изменяющиеся выходы ее элементов индицируются на фазовых выходах схемы (хотя бы на одном).*

Индицируемость схемы обеспечивает как правильную смену фаз, так и отказобезопасность. Действительно, если в противном случае изменение выхода некоторого элемента не отражается на выходных фазовых сигналах, определяющих момент перехода к следующей фазе, то и задержка сигнала на этом выходе вплоть до константного «залипания» не будет замечена, что при реальной работе схемы приведет или к состязанию, или к нарушению отказобезопасности.

Таким образом, индицируемость является необходимым условием самосинхронности для любой разомкнутой СС-схемы. Достаточные условия обсуждаются в главе 5.

### 2.5.3. Полная и частичная индицируемость

Рассмотрим, какие варианты индицирования внутренних и внешних сигналов возможны и необходимы в СС-схемах с типовым интерфейсом на рис. 2.13.

1. Все выходы элементов схемы, которые меняются в текущей фазе, должны индицироваться в этой фазе на выходных фазовых сигналах.

2. Входные ПФС-сигналы и У-сигналы могут, но не обязательно, индицироваться на выходных фазовых сигналах.

3. Выходные БС- и УС-сигналы, как выходы внутренних элементов, должны индицироваться на выходных И-сигналах.

4. Выходные ПФС-сигналы могут индицироваться либо не индицироваться на выходных И-сигналах.

5. Выходные И-сигналы могут индицироваться на других выходных фазовых сигналах схемы. Но, как правило, это не делается, так как подобная индикация избыточна.

Если не все входные фазовые сигналы индицируются на выходных (см. пункт 2), то будем говорить о *частичной индицируемости*. В этом случае, при использовании данной схемы, неиндицированные входы должны быть индицированы отдельно, в окружающей схеме.

Если, согласно тому же пункту, все входные фазовые сигналы индицируются на выходных, то имеет место *полная индицируемость*, и отдельной индикации этих входов не требуется.

Заметим, что в книгах [2, 3] рекомендуется всегда делать отдельную индикацию входных фазовых сигналов, что во многих случаях избыточно.

В заключение приведем простейшую СС-схему — инвертор (рис. 2.14). В ней отсутствуют инфосигналы, вход и выход — фазовые, значения спейсеров на входе и выходе противоположны, и вход индицируется на выходе (полная индицируемость).

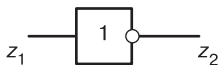


Рис. 2.14. Простейшая СС-схема

## Глава 3

# ОСНОВЫ ПОСТРОЕНИЯ САМОСИНХРОННЫХ СХЕМ

---

В общем виде можно сформулировать три основные задачи, решение которых должен обеспечить разработчик, проектируя любую цифровую схему:

- 1) реализовать желаемое функционирование схемы;
- 2) создать схему в некотором заданном базисе;
- 3) придать схеме определенные качества — быстродействие, приемлемые затраты и др.

### 3.1. Подходы к созданию СС-схем

Существует большое число подходов и методов построения асинхронных схем и их разновидностей. Описанные в литературе способы создания асинхронных схем, за исключением работ В. И. Варшавского с сотрудниками и NCL-методологии (см. ниже), не обеспечивают самосинхронности, поэтому получаемые схемы следует отнести к *квази-самосинхронным*.

Например, можно упомянуть две группы подобных методов. В одной группе с помощью некоторых формальных процедур получают схемы, обеспечивающие взаимодействие типа «запрос-ответ» между блоками. Сами блоки при этом не являются независимыми от задержек. Примерами таких подходов могут служить работы [20, 21].

Методы другой группы используют некоторые априорные условия типа «суммарная задержка трех последовательных элементов всегда больше задержки двух элементов» и связаны с конструированием соответствующей внутренней синхронизации в схеме, например [22, 23]. Правильность функционирования таких схем целиком зависит

от соотношения задержек между элементами, и поэтому они не являются независимыми от этих задержек.

Достоинством квази-самосинхронных подходов обычно считается то, что с приемлемыми транзисторными затратами реализуются схемы, имеющие в определенных пределах целевое свойство, например независимость от задержек. Недостаток таких подходов — необходимость обеспечения заявленных соотношений задержек (что очень непросто при изготовлении), а также отсутствие полной гарантии от ошибок даже при выполнении соотношений.

Рассмотрим теперь существующие методы, обеспечивающие самосинхронность.

### 3.1.1. Методы событийного подхода

В рамках событийного подхода наиболее глубоко и последовательно методы создания полумодулярных схем разрабатывались В. И. Варшавским и его группой [2, 3, 24]. Достоинства их подхода и методов следующие. Впервые созданы регулярные методы анализа и синтеза полумодулярных схем на функционально-логическом уровне. Методы имеют универсальный характер по отношению к способу СС-кодирования информации, наличию или отсутствию памяти, дисциплине смены наборов и другим исходным характеристикам. На основе разработанных методов была создана, также впервые, программная система анализа и синтеза СС-схем ФОРСАЖ (см. главу 6). Был предложен также ряд способов построения комбинационных разомкнутых самосинхронных схем, правда, не автоматизированных. Создано большое количество типовых элементов и ячеек СС-схем.

Помимо «ручного» способа, один из основных регулярных методов создания СС-схем, предлагаемый группой Варшавского, — синтез на основе диаграмм изменений (ДИ) [24], который для краткости назовем ДИ-синтезом. В качестве исходных данных для синтеза разработчик должен задать замкнутую диаграмму (с константными входами и выходами) аperiodического процесса, описывающего схему. Метод синтеза заключается в проверке ДИ на корректность и переводе ее описания в логические уравнения — модель Маллера.

Метод имеет несколько серьезных недостатков. Прежде всего, это сложность и необычность задачи для разработчика электронных схем, привыкшего иметь дело с функциональным описанием, временными диаграммами и таблицами истинности. Основная сложность состоит в том, что исходная диаграмма, которую должен подготовить разработчик, уже должна иметь совокупность свойств, эквивалентных



полумодулярности. Процедура синтеза лишь проверяет ее корректность по шести критериям и, если корректность соблюдена, выполняет преобразование исходного представления в систему логических уравнений. Данный метод было бы логичней назвать конверсией, так как целевое свойство схемы — полумодулярность — не появляется в результате процедуры, а его обеспечение возложено на разработчика.

Подготовка исходной ДИ также сопряжена с трудностями. Например, для комбинационных схем ДИ получаются очень громоздкими, поскольку в ДИ должны быть представлены все комбинации значений комбинационных входов. Использование переменных памяти и начальных состояний в методе также не является простой задачей.

Другим недостатком ДИ-синтеза является отсутствие блочности и иерархичности проектирования, т. е. ранее разработанные фрагменты не могут в неизменном виде использоваться в новой схеме, а требуют значительной переработки. В результате резко снижаются возможности проектирования практических схем.

Базис реализации в ДИ-синтезе не задается, т. е. является свободным (по факту это И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ, ИЛИ-И-НЕ). Такая ситуация, во-первых, может привести к неприемлемым для практики элементам типа многовходовых (более четырех) И-НЕ и другим. Во-вторых, проектирование в конкретных ограниченных базисах дешевой полузаказной технологии (БМК) становится нереальным.

Никакой оптимизации в ДИ-синтезе не предусмотрено, если не считать стандартной процедуры минимизации ДНФ. Для КМДП-технологии такой процедуры явно недостаточно, так как она не минимизирует непосредственно ни быстродействие, ни затраты. Во многих случаях, особенно включающих комбинационные преобразования, в результате синтеза получаются очень громоздкие схемы.

Таким образом, ДИ-синтез решает до конца только одну задачу проектирования — обеспечение желаемого функционирования схемы.

Тем не менее на основе ДИ-синтеза его авторами получено много схем небольшого объема, которые можно с успехом использовать в практических разработках.

### 3.1.2. NCL-методология

Методология Null Convention Logic (NCL-методология) [5] фирмы Theseus Research Inc появилась в конце 1990-х годов и была запатентована авторами. Фирма поддерживает только методологию, но конечным проектированием по ней не занимается. Хотя в ма-

териалах фирмы упоминаются разработки сторонних производителей микросхем с использованием этой методологии, никаких схем не приводится, что не позволяет оценить реальные результаты проектирования.

NCL-методология — единственный из зарубежных подходов, обеспечивающий самосинхронность схем.

Методология имеет ряд существенных недостатков, лишь отчасти отмеченных в литературе. Рекламные же заявления фирмы, как водится, трактуют о полном и окончательном решении проблемы. Поэтому целесообразно критически оценить данный подход.

Методология не опирается на основополагающую работу Маллера и Бартки [1], а лишь ссылается на одну из более поздних работ Маллера прикладного, а не фундаментального характера. Авторы не знакомы и с работами В. И. Варшавского с сотрудниками, хотя специалисты по асинхронике эти работы хорошо знают. Более того, судя по всему, даже сами эти фамилии авторам методологии неизвестны. В результате во многом интуитивного подхода получилась хотя и реализованная, но довольно «корявая» методология.

Вот основные характеристики и оценки методологии и получаемых схем.

1. Для кодирования информации используется ЕК (единичный код, см. раздел 2.2), в основном в модификациях при  $N = 2$  («double-rail») и  $N = 4$  («quad-rail»). Как уже отмечалось, данный тип кода наиболее избыточен из всех возможных.

2. Каждый каскад схемы должен быть выполнен и выполняется как звено конвейера. Обычно конвейеры применяются в специальных случаях, здесь же методология навязывает такую структуру во всех случаях, нужна она или нет. Данная структура совместно с ЕК обеспечивает индицируемость и в целом самосинхронность схем, но приводит к еще большим затратам по числу транзисторов.

3. Базовыми элементами NCL-методологии служат так называемые пороговые элементы. Конкретно было взято 27 логических функций для реализации элементов от двух до четырех входов. Каждый элемент может быть реализован в двух видах: как запоминающий ССБЭ (см. раздел 2.4.2.2) либо как вентильный элемент с локальными обратными связями.

Базовый элемент способен запоминать предыдущее состояние и имеет порог — количество изменяемых входов (в любом сочетании), необходимых для срабатывания. Использование таких элементов диктуется необходимостью индицирования. Очень часто базовые элементы применяются с переключением входов. Все эти конструк-

ции сложны, неоптимальны и увеличивают транзисторные затраты реализации.

4. Затраты реализации избыточны в 2,5—4 раза по осторожной оценке бывших сотрудников Варшавского [25].

5. Схемы работают в две фазы: NULL (спейсерная), давшая название методологии, и DATA (рабочая). Такое построение схем, как уже говорилось, было впервые предложено Варшавским с соавторами в [5] задолго до появления методологии. Используется только нулевое значение спейсера. Введение такого фиксированного значения уменьшает гибкость проектирования и также приводит к избыточности.

6. Хотя авторы методологии много пишут об оптимизации, эта оптимизация происходит лишь в пределах отмеченных выше особенностей и имеет характер минимизации ДНФ по картам Карно. Наличие избыточности и сплошной конвейеризации снижает быстродействие схем и увеличивает энергопотребление.

Таким образом, NCL-методология позволяет создавать СС-схемы, которые характеризуются большими избыточными затратами и соответственно неоптимальным быстродействием и энергопотреблением. Тем не менее, она остается единственной зарубежной методологией создания СС-схем, альтернативной подходам на основе работ Маллера и Варшавского.

### 3.1.3. Особенности разработки СС-схем в функциональном подходе

При разработке СС-схем в рамках функционального подхода, параллельно с общими задачами проектирования, упомянутыми в начале главы, требуется решить две главные проблемы, специфичные для этих схем: индикацию всех сигналов и отсутствие состязаний.

На практике реализация этих требований приводит к построению схем по принципу «снизу вверх». Таким путем наиболее удобно обеспечиваются и функционирование схемы, и требования самосинхронности. Эти требования проверяются на каждом этапе путем анализа на самосинхронность.

Вопросы функционирования и обеспечения самосинхронности тесно связаны. Практика показала, что при совместном решении этих вопросов на всех уровнях можно значительно снижать затраты и улучшать быстродействие схем.

Типовой порядок проектирования СС-схем состоит в следующем.

1. Создание самосинхронных функциональных фрагментов нижнего уровня и получение первичных И-сигналов. Здесь це-

лесообразно использовать многие ранее полученные схемные решения.

2. Соединение фрагментов схем с обеспечением самосинхронности.

3. Понижение количества И-сигналов (для СС-схем конечного применения — до одного).

4. Переход к более высокому уровню иерархии.

Ниже рассматривается ряд конкретных задач проектирования СС-схем в функциональном подходе.

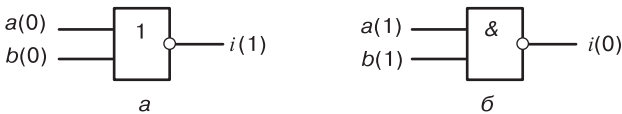
### 3.2. Способы индикации в СС-схемах

Общее правило индикации состоит в том, что каждый сигнал (выход элемента) с необходимостью должен индицироваться на выходных фазовых сигналах схемы.

Индикация влияет и на качество схем, и на процесс проектирования, поскольку требует определенных затрат и вносит свой вклад в общую задержку [26].

Индикацию можно рассматривать на двух уровнях: локальном и глобальном. Локальный уровень — это уровень функциональных преобразований и получения первичных И-сигналов. Глобальный уровень — это уровень соединений между фрагментами.

Простейшим видом локальной индикации будет индикация ПФС-сигналов элементарными индикаторами (рис. 3.1).



**Рис. 3.1.** Индикаторы ПФС-сигналов:

$a$  — с нулевым спейсером на входе;  $б$  — с единичным спейсером на входе

Другие способы индикации на локальном уровне будут рассмотрены ниже совместно с вопросами синтеза.

При обеспечении правила индикации в СС-схемах появляется множество И-сигналов и возникает задача снижения их числа.

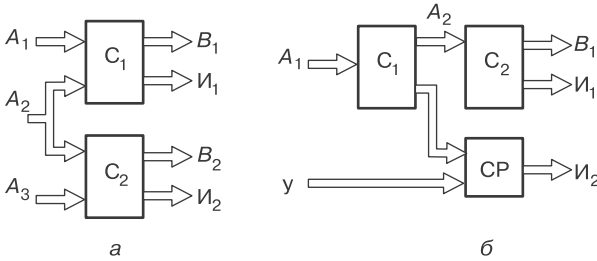
Уменьшение количества И-сигналов с сохранением индицируемости и отсутствия гонок назовем *редукцией*, а схемы, реализующие такое понижение, — *схемами редукции*.

Далее будут рассмотрены общие вопросы глобальной индикации и построения схем редукции.

### 3.2.1. Глобальная индикация

Для оптимизации затрат и быстродействия при соединениях СС-схем между собой целесообразно следовать нескольким правилам.

1. При параллельном соединении СС-схем (рис. 3.2, а) каждый их общий вход следует индцировать только в какой-то одной схеме.



**Рис. 3.2.** Соединение СС-схем:

а — параллельное; б — последовательное;

$C_k$  — СС-схемы; СР — схема редукции;  $A_k$  — любые входы;  $B_k$  — инфосигналы;  $Y$  — управляющие сигналы;  $I_k$  — индикаторные сигналы

Например, на рис. 3.2, а сигналы  $A_2$  могут быть индцированы либо в схеме  $C_1$ , либо в схеме  $C_2$ . Индикация в обеих схемах не обязательна и приводит к лишним затратам.

2. При последовательном соединении СС-схем (рис. 3.2, б) с целью уменьшения задержек следует максимально совмещать во времени процессы функциональных преобразований и редукции.

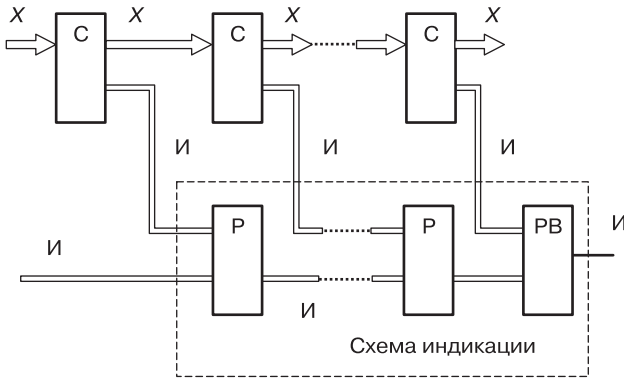
Например, на рис. 3.2, б процессы переключений в схемах  $C_2$  и СР проходят параллельно по времени.

В результате можно рекомендовать общую структуру глобальной индикации СС-схем, изображенную на рис. 3.3.

Из рис. 3.3 видно, что схемы редукции Р в параллельном канале в зависимости от своих задержек могут влиять или не влиять на общую задержку схемы, а выходная схема редукции РВ дает прямой вклад в общую задержку схемы.

### 3.2.2. Схемы редукции

Схемы редукции (СР) были рассмотрены в [2] («многовходовые индикаторы»), однако приведенные там оценки неприменимы к КМДП-схемам. В других литературных источниках, российских и зарубежных, анализ таких схем не обнаружен.



**Рис. 3.3.** Фрагмент схемы со структурно оптимизированной индикацией:  
 С — СС-схемы; Р — схемы редукции в параллельном канале; РВ — выходная схема редукции;  
 X — инфосигналы; И — индикаторные сигналы

Все входы СР являются У-сигналами, выходы — одним или несколькими И-сигналами. Будем считать, что все входы СР приведены к одному и тому же значению спейсера. При необходимости это легко делается с помощью инверторов.

С теоретической и практической точек зрения, интересен и важен вопрос, какие типы схем редукции вообще возможны.

### 3.2.2.1. Возможные схемы редукции

Воспользуемся методом составления и решения логических уравнений [27, Приложение]. Будем исследовать СР с одним, двумя и более выходами.

Первый вариант — схема с одним выходом.

Пусть  $x_i$  ( $i = 1, \dots, n$ ) — входы СР,  $y$  — ее выход.

Назовем *нулевым* и *единичным наборами* значения  $x_i = 0$  и  $x_i = 1$  ( $i = 1, \dots, n$ ) соответственно, оба вместе — *крайними*, а *промежуточным набором* — любой набор значений входов, не являющийся крайним.

В каждой фазе переход от одного крайнего набора к другому должен происходить монотонно, т. е. только с увеличением числа единиц (нулей). В противном случае нарушается допустимость переходов (см. главу 2). Это требование накладывается как на входы, так и на выходы СР. Монотонность переходов гарантирует отсутствие состязаний на элементах СР.

Без уменьшения общности предположим, что схемы редукции — не инвертирующие. В этом случае на нулевом входном наборе выход

СР должен быть нулем, на единичном наборе — единицей, что дает два уравнения:

$$\begin{aligned} \hat{x}_1 \hat{x}_2 \dots \hat{x}_n &\rightarrow \hat{y}, \\ x_1 x_2 \dots x_n &\rightarrow y, \end{aligned} \quad (3.1)$$

где стрелка означает логическое следование (импликацию).

Покажем, что на промежуточных наборах выход СР должен иметь значение  $y_p$ , равное значению выхода на ближайшем предыдущем крайнем входном наборе. Предположим обратное. Это значит, что выход СР меняет свое значение на противоположное не на крайнем, а на промежуточном наборе. Но по назначению СР такое изменение должно сигнализировать о том, что все И-сигналы  $x_i$  уже пришли к другому крайнему набору, что противоречит действительности.

Уравнение для промежуточных наборов поэтому выглядит так:

$$(x_1 \vee x_2 \vee \dots \vee x_n)(\hat{x}_1 \vee \hat{x}_2 \vee \dots \vee \hat{x}_n) \rightarrow (y = y_p). \quad (3.2)$$

Решая систему уравнений (3.1), (3.2) в соответствии с методом, изложенным в [27, Приложение], приходим к заключению, что решение системы существует всегда, оно единственно и имеет вид:

$$y = x_1 x_2 \dots x_n \vee (x_1 \vee x_2 \vee \dots \vee x_n) y_p. \quad (3.3)$$

Это известное уравнение *гистерезисного триггера* ( $\Gamma$ -триггера) [2].

В результате получен следующий важный вывод.

*Существует единственный тип схем редукции с одним выходом —  $\Gamma$ -триггер.*

Второй вариант — возможные СР с двумя выходами. Аналогично предыдущему должны быть записаны три уравнения: два для крайних наборов и одно — для промежуточных. На промежуточных наборах оба выхода должны быть разными и постоянными — для соблюдения монотонности по выходным наборам.

$$\begin{aligned} \hat{x}_1 \hat{x}_2 \dots \hat{x}_n &\rightarrow \hat{y}_1 \hat{y}_2, \\ x_1 x_2 \dots x_n &\rightarrow y_1 y_2, \\ (x_1 \vee x_2 \vee \dots \vee x_n)(\hat{x}_1 \vee \hat{x}_2 \vee \dots \vee \hat{x}_n) &\rightarrow (y_1 = C)(y_2 = \hat{C}), \end{aligned} \quad (3.4)$$

где  $C$  — константа в пределах текущей фазы.

Решая систему (3.4) относительно  $y_1$  и  $y_2$ , заключаем, что решение существует всегда и имеет вид:

$$\begin{aligned} y_1 &= x_1 x_2 \dots x_n \vee (x_1 \vee x_2 \vee \dots \vee x_n) C, \\ y_2 &= x_1 x_2 \dots x_n \vee (x_1 \vee x_2 \vee \dots \vee x_n) \hat{C}. \end{aligned} \quad (3.5)$$

Придавая  $C$  различные значения, можно получить различные варианты решений и схем.

Анализ (3.5) показывает, что все варианты, имеющие смысл в данном контексте, сводятся к двум.

1. Вариант  $C = y_{1p}$  — предыдущее значение  $y_1$ . Тогда первое решение в (3.5) совпадает с  $\Gamma$ -триггером. Реализация второй функции практически бессмысленна, так как первая уже решает задачу, а вторая лишь обеспечивает свойство монотонности и является излишней для индикации. Данный вариант в результате приводится к полученной ранее СР с одним выходом.

2. Вариант  $C = 0$  (или, для симметричного решения,  $C = 1$ ) дает существенно другие решения:

$$\begin{aligned} y_1 &= x_1 x_2 \dots x_n, \\ y_2 &= x_1 \vee x_2 \vee \dots \vee x_n. \end{aligned} \quad (3.6)$$

Это уравнения схемы, названной в [2] *схемой параллельного сжатия (СПС)*.

Если допустить в схемах редукции инвертирующие элементы (уравнения), то аналогично можно заключить, что вместо функций И и ИЛИ в СПС можно использовать многовыходовые функции И-НЕ и ИЛИ-НЕ. Таким образом, справедлив следующий вывод.

*Существует только один тип схем редукции с двумя выходами, не сводимый к  $\Gamma$ -триггерам, — схемы параллельного сжатия.*

Рассмотрим СР с тремя и более выходами. Уравнения для крайних наборов запишутся подобно (3.1). На промежуточных входных наборах выходные переменные также должны отличаться от своих крайних наборов. Решения уравнений в этом случае будут не единственными. Варианты таких решений будут повторять или комбинировать предыдущие. Практическая реализация подобных вариантов нецелесообразна также и потому, что СР с тремя и более выходами неэффективны — имеют малый коэффициент редукции — отношение числа входов к числу выходов. Это заключение будет подтверждено далее при рассмотрении оптимизации СР.

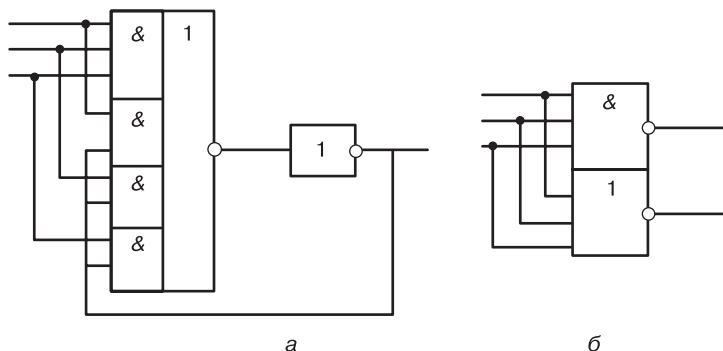
Сказанное позволяет сделать такой вывод: практическое значение имеют только два типа схем редукции:  $\Gamma$ -триггеры и СПС.

Примеры реализации таких схем приведены на рис. 3.4.

### 3.2.2.2. Оптимизация схем редукции

Уравнения (3.3) и (3.6) реализуются в КМДП-технологии с помощью элементов с числом входов не более четырех. Поэтому СР с большим числом входов должны строиться путем каскадного сое-





**Рис. 3.4.** Примеры схем редукции с тремя входами:  
*a* — Г-триггер; *б* — СПС

динения простейших схем. Как доказано в [3], каскадное соединение подобных схем также является схемой редукции и обеспечивает правильную индикацию.

Назовем *редуктором* элементарную СР, т. е. СР, состоящую из минимального количества базовых элементов.

Пусть имеется однородная (состоящая из редукторов одного типа) схема СР, включающая  $L$  каскадов, каждый из которых имеет коэффициент редукции  $k_i$  ( $i = 1, \dots, L$ ). Уменьшение по всем каскадам равно

$$K = k_1 k_2 \dots k_L. \quad (3.7)$$

Будем оценивать быстродействие вентиля с помощью задержки, отнесенной к единице нагрузки  $\tau = T/(rC_n)$ , где  $T$  — задержка вентиля;  $r$  — параметр, зависящий от сопротивления открытого транзистора;  $C_n$  — емкость нагрузки вентиля. Относительную задержку вентиля (без учета паразитных емкостей) можно просто выразить через индекс задержки  $I$  из формулы (2.3):

$$\tau = I.$$

Для обобщения формул введем структурный параметр  $S$ , учитывающий тип редуктора:  $S = 1$  для Г-триггера,  $S = 2$  для СПС. Тогда связь между коэффициентом редукции и ИЗ в каскаде будет иметь вид:  $I_i = S k_i$ .

С учетом того что Г-триггер состоит из двух последовательно соединенных вентилях, а СПС — нет, общая относительная задержка СР равна

$$T = \sum_{i=1}^L \tau_i = S \sum_{i=1}^L k_i + (2-S)L. \quad (3.8)$$

### А. Оптимизация быстродействия.

Задача ставится следующим образом: определить, при каких условиях задержка СР (3.8) минимальна, если задан общий коэффициент снижения И-сигналов (3.7).

Решая задачу относительно  $k_i$  с помощью множителя Лагранжа, найдем, что минимум достигается при  $k_i = k$ , т. е. коэффициенты редукции во всех каскадах должны быть одинаковыми. Тогда  $K = k^L$ . Подставив отсюда  $L$  в (3.8), получим

$$T = \frac{\ln K}{\ln k} (Sk + 2 - S). \quad (3.9)$$

Выражение (3.9) имеет минимум по  $k$ . Этот минимум достигается при следующих значениях:

$$\begin{aligned} k &= e = 2,72 \text{ для СПС,} \\ k &= 3,59 \text{ для Г-триггера.} \end{aligned} \quad (3.10)$$

Заметим, что коэффициент редукции для СПС ограничен числом 2, так как при этом число входов ее вентилях уже достигает четырех. Кроме того, СР, построенная только на СПС, доводит конечное число И-сигналов до двух, и требуется еще дополнительный Г-триггер, чтобы получить один И-сигнал на выходе.

Сделаем сравнение СР с различными параметрами на нескольких примерах. Пусть требуется редуцировать И-сигналы от  $N$  до одного. В таблице 3.1 приведены относительные задержки некоторых вариантов реализаций СР.

Таблица 3.1

Относительные задержки схем редукции

Вариант СР	$N = 16$	$N = 32$	$N = 64$
СПС + конечный Г-триггер, $k = 2$	15	19	23
Г-триггер, $k = 2$	12	15	18
Г-триггер, $k = 3$	11	13	16
Г-триггер, $k = 4$	10	13	15

Учитывая, что паразитные емкости сильнее влияют на задержку Г-триггера с  $k = 4$ , чем с  $k = 3$ , можно сделать следующий вывод.

*Оптимальными по быстродействию будут схемы редукции, выполненные на Г-триггерах с  $k = 3$  или  $k = 4$  (значения, ближайšie к оптимальному).*

### Б. Оптимизация затрат по числу транзисторов.

Определим условия минимальности числа транзисторов в СР.

Пусть необходимо редуцировать И-сигналы от  $N$  до одного и СР состоит из одинаковых редукторов с коэффициентом  $k$ . Оценивать затраты будем величиной *удельных затрат редукации*  $Z$  — числом транзисторов на один входной И-сигнал.

Введем обозначения:  $z_p$  — число транзисторов в одном редукторе,  $n$  — число входов одного редуктора,  $M$  — число редукторов. Для СПС  $n = 2k$ , а для Г-триггеров  $n = k$ .

Общие затраты на редукацию составят

$$ZN = Mz_p + D, \quad (3.11)$$

где  $D$  — поправка, учитывающая реализацию схемы. Она появляется при нарушении однородности СР, когда приходится использовать редуктор с отличающимися от остальных параметрами. Например, в схеме на СПС конечным редуктором должен быть Г-триггер. Поскольку нарушений обычно мало,  $D$  не превышает нескольких единиц.

Число редукторов является суммой геометрической прогрессии:

$$M = \sum_{i=1}^L N / (nk^{i-1}) = \frac{Nk-1}{n(k-1)}. \quad (3.12)$$

Подставляя это выражение в (3.11), найдем

$$Z = \frac{kz_p}{n(k-1)} + \frac{1}{N} \left( D - \frac{z_p}{n(k-1)} \right). \quad (3.13)$$

Таким образом, удельные затраты на индикацию определяются в основном параметрами применяемого редуктора. Назовем *затратным показателем редуктора* (ЗПР) величину

$$Q_p = \frac{kz_p}{n(k-1)}. \quad (3.14)$$

Она является асимптотической (для больших  $N$ ) оценкой удельных затрат на индикацию. С помощью ЗПР удобно также сравнивать редукторы между собой — чем меньше этот показатель, тем эффективнее редуктор.

С точки зрения затрат, имеют значение две реализации Г-триггера: на обычных вентилях и на элементах с внутренним третьим состоянием (см. главу 2). Вторую реализацию обозначим как ГЗ-триггер. В литературе она имеет и другое название — С-элемент Маллера (схемы приведены в главе 4).

В таблице 3.2 приведены величины ЗПР для различных редукторов.

Таблица 3.2

### Затратные показатели редукторов

Число входов $n$	СПС	Г-триггер	ГЗ-триггер
2	—	12	8
3	12	6	5
4	8	6,7	4

Расчеты конкретных СР показывают, что при  $N \geq 16$  величина ЗПР (3.14) оценивает сверху удельные затраты на редукцию с погрешностью не более 6,3%.

При сравнении ЗПР видно, что СПС проигрывает по эффективности Г-триггерам, а наилучшим редуктором для СР является схема Г-триггера с внутренним третьим состоянием. Следует также добавить, что, хотя площадь на кристалле, занимаемая слабым инвертором, входящим в ГЗ-триггер, больше площади обычного инвертора, общая площадь ГЗ-триггеров меньше, чем площадь Г-триггеров с соответствующим числом входов.

Резюмируя сказанное, можно утверждать следующее.

*Из двух типов схем редукции, имеющих реальный смысл, СПС и Г-триггеров, по всем показателям — быстродействию и затратам транзисторов — наиболее эффективны Г-триггеры и особенно ГЗ-триггеры. Применение СПС в практических схемах нецелесообразно.*

### 3.3. Построение комбинационных СС-схем

Вопросы проектирования комбинационных самосинхронных схем в литературе специально рассмотрены только в книге [3] и в NCL-методологии [5] (см. раздел 3.1.2). Как уже говорилось, в последнем случае схемы строятся как конвейеры и приводят к очень громоздким по затратам решениям.

Если попытаться воспользоваться событийным ДИ-синтезом, то сразу возникает проблема исходного задания, построить которое, кроме тривиальных случаев, крайне сложно. Помимо того, что требуется описать замкнутый корректный (эквивалентный полумодулярному) процесс, необходимо в этом процессе пройти все возможные комбинации значений информационных входов. Например, для 2-разрядного комбинационного сумматора на два операнда таких комбинаций 16, а для 4-разрядного — уже 256.

Далее построение комбинационных схем будет рассмотрено с позиций функционального подхода. Исходное задание будет содержать набор функций, определяющих только функциональные преобразования схемы, без каких-либо требований со стороны самосинхронности. Напомним также (см. главу 2), что в практических СС-схемах, не связанных с приемом-передачей данных по шинам и каналам связи, используется парафазное со спейсером кодирование — ПФС-коды.

Исходные некодированные логические функции (информационные), в отличие от функций в парафазном представлении — ПФС-функций, будем называть *моно-функциями*, или *М-функциями*.

Исходными данными для построения комбинационных СС-схем будут: описание функционирования в виде множества комбинационных (без запоминания) М-функций; базис реализации, фиксированный или свободный, в виде ССБЭ; критерии оптимизации — быстродействие или затраты.

Строго оптимальное построение СС-схем, за исключением простейших, практически нереально. Поэтому, как и во всех подобных случаях, схема подходящим образом разбивается на фрагменты, каждый из них синтезируется отдельно и по возможности оптимально.

### 3.3.1. Парафазное преобразование

При ПФС-кодировании информации для соблюдения двухфазной дисциплины и допустимости переходов каждая ПФС-функция, реализуемая одним элементом, должна быть монотонной по каждой переменной [2]. При реализации в КМДП-технологии монотонность (изотонность или антитонность по всем переменным) обеспечивается СС-базовыми элементами.

Переход от М-функций к соответствующим монотонным ПФС-функциям назовем *парафазным преобразованием со спейсером* (ПФС-преобразованием).

Рассмотрим М-функцию от  $N$  переменных  $F(x_1, \dots, x_N)$ . В ПФС-коде каждой переменной  $x_i$  соответствует сопряженная переменная  $y_i$ . Функция  $F$  должна быть преобразована в пару функций  $U$  и  $V$  от  $2N$  переменных  $x_i$  и  $y_i$ .

Обозначим  $s_1$  и  $s_2$  — значения входного и выходного спейсеров (константы). Тогда условиями перехода будут:

1) в фазе спейсера:

$$\begin{aligned} x_i &= y_i = s_1, & i = 1, \dots, N, \\ U &= V = s_2. \end{aligned} \quad (3.15)$$

2) в рабочей фазе:

$$\begin{aligned} x_i \neq y_i, \quad i = 1, \dots, N, \\ U = F, \quad V = \wedge F. \end{aligned} \quad (3.16)$$

Введем функции спейсера и рабочей фазы для входных наборов:

$$\begin{aligned} S = s_1 x_1 y_1 \dots x_N y_N \vee \wedge s_1 \wedge x_1 \wedge y_1 \dots \wedge x_N \wedge y_N, \\ R = r_1 \dots r_N, \end{aligned} \quad (3.17)$$

где  $r_i = x_i \oplus y_i$ .

Тогда из условий (3.15) и (3.16) можно записать уравнения для ПФС-преобразования:

$$\begin{aligned} S \rightarrow (U = s_2)(V = s_2), \\ R \rightarrow (U = F)(V = \wedge F), \end{aligned} \quad (3.18)$$

где стрелка означает логическое следование (импликацию).

Из этих уравнений необходимо найти функции  $U$  и  $V$ . Формально, в соответствии с методом решения логических уравнений [27, Приложение], условием существования их решений будет

$$SR = 0. \quad (3.19)$$

Оно означает, что спейсер и рабочая фаза не должны иметь общих наборов. По определению фаз, это справедливо всегда, что также следует и из (3.17).

Сами решения (3.18) будут иметь вид:

$$\begin{aligned} U = s_2 S \vee FR \wedge S \vee h \wedge S \wedge R, \\ V = s_2 S \vee \wedge FR \wedge S \vee g \wedge S \wedge R, \end{aligned} \quad (3.20)$$

где  $h$  и  $g$  — произвольные величины. Они могут быть либо константами, либо произвольными функциями от  $x_i$  и  $y_i$ .

Из решений (3.20) видно, что функции  $U$  и  $V$  являются неполностью определенными. Они не определены (произвольны) на наборах, отличных от рабочих и спейсера. Путем доопределения на этих наборах можно получить все множество возможных ПФС-функций. Однако из всего этого множества нас интересуют только монотонные функции. Покажем, что монотонные решения среди (3.20) всегда существуют: при одинаковых спейсерах  $s_1 = s_2$  — изотонные, при разных спейсерах  $s_1 \neq s_2$  — антитонные.

Пусть для определенности  $s_1 = s_2 = 0$ , и рассмотрим функцию  $U$ . Используем известное разложение функции  $F$  по переменной  $x_i$ :  $F = x_i A_i \vee \wedge x_i B_i$ , где  $A_i$  и  $B_i$  не зависят от  $x_i$ , а также свойство идемпотентности для второго равенства из (3.17)  $R = r_i R$ :

$$\begin{aligned} U &= (x_i A_i \vee \wedge x_i B_i) r_i R \wedge S \vee h \wedge S \wedge (r_i R) = \\ &= (x_i \wedge y_i A_i \vee \wedge x_i y_i B_i) R \wedge S \vee h \wedge S (x_i y_i \vee \wedge x_i \wedge y_i \vee \wedge R). \end{aligned} \quad (3.21)$$

Положим  $h = x_i y_i (A_i \vee B_i) R \wedge S \vee h_1 \wedge R$  ( $h_1$  — также произвольная величина) и получим

$$U = G_i R \wedge S \vee h_1 \wedge S \wedge R. \quad (3.22)$$

Функция  $G_i = x_i A_i \vee y_i B_i$  уже не имеет вхождения  $x_i$  с отрицаниями. Полученное выражение (3.22) по форме совпадает с (3.20) для  $U$ , а величина  $h_1$ , в силу произвольности, не отличается от  $h$ . Повторяя процесс для остальных переменных, на последнем этапе получим

$$U = G_N R \wedge S \vee h \wedge S \wedge R, \quad (3.23)$$

где  $G_N$  — изотонная функция от  $2N$  переменных. Возьмем теперь  $h = G_N$ , тогда  $U = G_N \wedge S$ . В принятых условиях  $\wedge S$  представляет собой дизъюнкцию всех переменных  $x_i$  и  $y_i$  (без отрицаний). Поэтому по правилам поглощения  $U = G_N$ .

Аналогично можно получить изотонную функцию  $V$ , а также антитонные решения при  $s_1 \neq s_2$ .

Изложенная процедура показывает, что монотонные решения ПФС-преобразования существуют всегда. Она также дает простой способ такого преобразования. Назовем этот способ *МПФС-процедурой*. Он состоит в том, что в выражениях функций  $F$  и  $\wedge F$  для получения изотонного преобразования все вхождения переменных  $\wedge x_i$  заменяются на сопряженную переменную  $y_i$ , а для антитонного преобразования — все вхождения  $x_i$  на  $\wedge y_i$ .

Однако множество решений ПФС-преобразования не исчерпывается только теми, которые получены МПФС-процедурой. Например, М-функции  $F = x_1 \vee \wedge x_2$  при разных спейсерах на входе и выходе соответствуют как одна, так и другая следующие функции:

$$\begin{aligned} U &= \wedge y_1 \vee \wedge x_2, \\ U &= \wedge x_1 \wedge x_2 \vee \wedge y_1 \wedge x_2 \vee \wedge y_1 \wedge y_2. \end{aligned}$$

Наличие, как правило, большого выбора монотонных решений облегчает синтез комбинационных схем в ограниченном и неограниченном базисах.

### 3.3.2. Связь ПФС-преобразования и индикации

Как указывалось выше, в СС-схемах входные ПФС-сигналы должны быть так или иначе индцированы на выходных фазовых сигналах. Традиционно в таких случаях применяются отдельные индикаторы,

показанные на рис. 3.1. Однако во многих случаях некоторая часть входных ПФС-сигналов индицируется на выходах автоматически, в силу выполняемых элементами схемы функций, и для таких сигналов нет необходимости делать специальные индикаторы. Например, схема, реализующая две М-функции 2-И и 2-ИЛИ, в ПФС-кодах описывается уравнениями (при  $s_1 = 0, s_2 = 1$ ):

$$\begin{aligned} U_1 &= \hat{y}_1 \hat{y}_2, & U_2 &= \hat{y}_1 \vee \hat{y}_2, \\ V_1 &= \hat{x}_1 \vee \hat{x}_2, & V_2 &= \hat{x}_1 \hat{x}_2. \end{aligned}$$

В такой схеме все входные ПФС-сигналы уже без индикаторов индицируются на выходных.

Рассмотрим связь между видом исходных М-функций и индикацией при ПФС-преобразовании. Сначала проанализируем индикацию перехода в спейсер.

Индикация в данном случае эквивалентна тому, что булева производная хотя бы одного из двух выходов по заданному входу равняется единице в состоянии спейсера.

Напомним, что булева производная функции  $F$  по переменной  $x$  равна:  $\Delta F(x) = F(x=0) \oplus F(x=1)$ .

**Утверждение 1.** Если исходная М-функция представима в одной из ниже приведенных форм, то имеет место индикация перехода в спейсер следующих ПФС-переменных.

а) В формах  $F = x \vee F_1$  и  $F = \hat{x}F_1$  (функция  $F_1$  не зависит от  $x$ ):

- при  $s_1 = 0$  индицируется переменная  $x$ ,
- при  $s_1 = 1$  индицируется переменная  $y$ .

Для доказательства возьмем первую форму и применим МПФС-процедуру для  $s_1 = s_2$ . Тогда  $U = x \vee G_1, V = yG_2$  ( $G_1$  и  $G_2$  не зависят ни от  $x$ , ни от  $y$ ).

$$\begin{aligned} \Delta U(x) &= G_1 \oplus 1 = \hat{G}_1 = 1 \text{ при } s_1 = s_2 = 0. \\ \Delta V(y) &= G_2 = 1 \text{ при } s_1 = s_2 = 1. \end{aligned}$$

Для  $s_1 \neq s_2$  имеем:  $U = \hat{y} \vee G_1, V = \hat{x}G_2$  ( $G_1$  и  $G_2$  не зависят ни от  $x$ , ни от  $y$ ).

$$\begin{aligned} \Delta U(y) &= G_1 \oplus 1 = \hat{G}_1 = 1 \text{ при } s_1 = 1, s_2 = 0. \\ \Delta V(x) &= G_2 = 1 \text{ при } s_1 = 0, s_2 = 1. \end{aligned}$$

Для второй формы доказательство аналогично.

б) В формах  $F = xF_1$  и  $F = \hat{x} \vee F_1$  (функция  $F_1$  не зависит от  $x$ ):

- при  $s_1 = 0$  индицируется переменная  $y$ ,
- при  $s_1 = 1$  индицируется переменная  $x$ .

Доказательства также аналогичны.



**Утверждение 2.** Если исходная М-функция представима в форме  $F = x \oplus F_1$  (функция  $F_1$  не зависит от  $x$ ), то имеет место индикация перехода в рабочую фазу обеих сопряженных ПФС-переменных, соответствующих  $x$ , при любых сочетаниях спейсеров.

Для доказательства возьмем  $s_1 = s_2 = 0$ , поскольку в остальных случаях оно аналогично. Исходные и ПФС-функции примут вид:

$$F = \hat{x}F_1 \vee xF_1, \quad \hat{F} = xF_1 \vee \hat{x}\hat{F}_1, \\ U = yG_1 \vee xG_2, \quad V = xG_1 \vee yG_2.$$

Функции  $G_1$  и  $G_2$  получились соответственно из  $F_1$  и  $\hat{F}_1$  и потому в рабочей фазе не равны:  $G_1 \neq G_2$ . Далее,

$$\Delta U(x) = yG_1 \oplus (G_2 \vee yG_1) = G_2 \oplus yG_1G_2 = G_2, \\ \Delta V(x) = yG_2 \oplus (G_1 \vee yG_2) = G_1 \oplus yG_1G_2 = G_1.$$

Либо одна, либо другая булева производная в рабочей фазе отлична от нуля. Такой же результат получается и для переменной  $y$ , что и доказывает утверждение.

Пользуясь приведенными признаками, можно заранее сказать, что для совокупности трех М-функций

$$F_1 = \hat{(x_1x_2x_3)}, \\ F_2 = \hat{(x_1 \vee x_2 \vee x_3)}, \\ F_3 = x_1 \oplus x_2 \oplus x_3$$

после ПФС-преобразования все входные ПФС-переменные индицируются на выходных при всех сочетаниях спейсеров, и специальных индикаторов для входных сигналов не требуется.

### 3.3.3. СС-секции

Рассмотрим вопрос о том, как из множества исходных моно-функций получить СС-схему (соответствующие логические уравнения).

Из-за особенностей ПФС-кодирования и индикации М-функции целесообразно преобразовывать в СС-схемы не поодиночке, а некоторыми наборами одновременно, когда за счет общей индикации можно получать более оптимальные решения.

Будем называть *М-секцией* набор из одной или нескольких М-функций, подлежащих одновременному преобразованию в СС-схему.

Назовем также *СС-секцией* СС-схему с частичной или полной индикацией, реализующую М-секцию и удовлетворяющую следующим условиям.

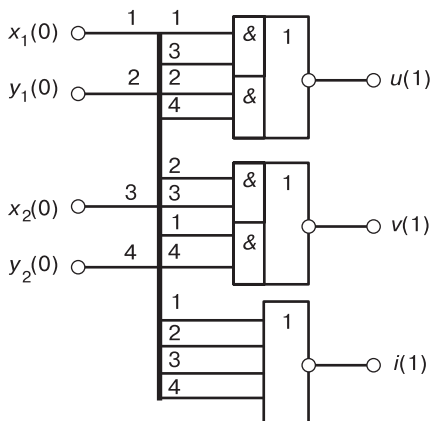
1. Она выполняет функциональные преобразования в ПФС-кодах. Ее входами являются ПФС-сигналы, выходами — ПФС-сигналы и И-сигналы.

2. Каждая ПФС-функция и каждый индикатор должны реализовываться одним ССБЭ, не допускается каскадного соединения элементов.

3. Для всех ПФС-входов секции должно быть единое значение входного спейсера, а для всех выходов — единое значение выходного спейсера.

Из двух последних условий следует, что логические функции, описывающие СС-секцию и реализуемые СС-базовыми элементами, должны быть все изотонными или все антитонными по всем переменным.

На рис. 3.5 приведен пример СС-секции, для которой М-секция состоит из одной функции — исключаяющего ИЛИ. В этой секции все входы индицируются на выходах.



**Рис. 3.5.** СС-секция исключаяющего ИЛИ: каждая СС-секция может быть исследована и синтезирована независимо от других секций в схеме; СС-секции достаточно легко объединять между собой в параллельные и последовательные структуры при реализации необходимой внешней индикации

Рассмотрим вопрос о возможных входных наборах СС-секций. Их можно разделить на четыре непересекающихся множества:

$S$  — спейсер,

$R$  — рабочие наборы,

$P$  — промежуточные наборы,

$Z$  — запрещенные наборы.

Промежуточные наборы — это наборы, принимаемые сигналами при переходе из спейсера к рабочим и наоборот.  $S$ -,  $R$ - и  $P$ -наборы составляют множество допустимых наборов, которые реализуются в СС-секциях. Напротив, запрещенные наборы в исправных схемах никогда не могут возникать. Запрещенные наборы — это наборы, в которых хотя бы одна пара сопряженных сигналов принимает значения, инверсные спейсеру.

В таблице 3.3 приведены данные о количестве возможных наборов в зависимости от числа входов  $M$ -секции ( $M$ -сигналов) (спейсер составляет один набор).

Таблица 3.3

Количество входных наборов СС-секции

Исходные $M$ -сигналы	ПФС-сигналы	$R$	$P$	Всего допустимо	$Z$	Всего наборов
1	2	2	—	3	1	4
2	4	4	4	9	7	16
3	6	8	18	27	37	64
4	8	16	64	81	175	256
5	10	32	210	243	781	1024
6	12	64	664	729	3367	4096
7	14	128	2058	2187	14 191	16 388
8	16	256	6304	6561	58 975	65 536

Из таблицы видно, что СС-секции работают с большим количеством промежуточных наборов по отношению к рабочим и спейсеру. Как будет показано ниже, это обстоятельство улучшает возможности синтеза и оптимизации СС-секций.

### 3.3.4. Общий порядок построения комбинационных СС-схем

Построение (синтез) комбинационной схемы состоит из нескольких этапов. Рассмотрим особенности каждого из них.

#### А. Каскадирование и балансировка.

На этом этапе исходное множество  $M$ -функций эквивалентно преобразуется с целью разбить его подходящим образом на последовательные каскады.

Здесь надо иметь в виду, что каждая исходная  $M$ -функция в дальнейшем будет преобразована в две ПФС-функции, и каждая из последних должна реализовываться только одним ССБЭ. Это обстоятельство накладывает ограничение на вид исходных  $M$ -функций.

Обозначим  $I_M$  — максимальное число букв в импликантах ДНФ М-функции и ее инверсии. Из выражений (3.20) и, в частности, из МПФС-процедуры можно видеть, что выполняется неравенство для ИЗ ПФС-функций:

$$I_{\text{ПФС}} \geq I_M. \quad (3.24)$$

Поэтому величина  $I_M$  М-функций не должна превышать числа 4.

В остальном характер связи между М-функциями и параметрами реализации — быстродействием и затратами — такой же, как и для обычных комбинационных схем. Например, нецелесообразно без необходимости представлять М-функции двухвходовыми вариантами, так как это приводит к излишним затратам. Оптимальными значениями  $I_M$  будут 3 или 4, в обеспечении чего и состоит смысл балансировки.

Интересно отметить, что инверторы в М-функциях при переходе к ПФС-функциям исчезают, так как в ПФС-кодах инверсию можно реализовать, переменяя местами сопряженные выходы.

На этом же этапе производится выявление и реализация одинаковых подфункций для снижения общих затрат.

### **Б. Секционирование исходного описания.**

Задача этого этапа — разбить функции, полученные на первом этапе, на М-секции — группы функций, которые будут затем синтезированы в СС-секции. Как и предыдущий, данный этап во многом определяет окончательные характеристики СС-секций после синтеза.

По определению секция реализует один каскад (ярус) функционального преобразования. Исходное описание поэтому надо предварительно рассортировать по ярусам. Ярус — это множество М-функций, расположенных «на одинаковом расстоянии» по числу функциональных преобразований от входов (или выходов) описания. Группировать М-функции в М-секции далее необходимо в пределах одного яруса.

При разбиении следует учитывать, что маленькие секции, как правило, приводят к увеличению общих затрат, так как чем больше функций в секции, тем больше возможностей реализации и индцирования по информационному каналу. Однако процедура синтеза для больших секций, естественно, сложнее.

В каждом ярусе необходимо обеспечить индикацию каждого входного сигнала на выходах яруса. Поэтому если один и тот же сигнал попадает в разные секции, то его достаточно индцировать в какой-то одной из них.

### **В. Синтез СС-секций.**

На этом этапе каждая выделенная М-секция должна быть синтезирована в СС-секцию.

### **Г. Синтез схемы редукции И-сигналов (при необходимости).**

При практическом синтезе возможны итеративные возвраты на предыдущие этапы для корректировки получаемых решений.

Этапы А и Б связаны только с преобразованием логических функций и их анализом, и по способам реализации не являются специфическими для СС-схем. Этап Г описан выше, поэтому далее будет рассматриваться только этап В.

### **3.3.5. Синтез СС-секций**

Практически все существующие схемотехнические способы, подходящие для данной цели, изложены в [2] и кратко повторены в [3]. Эти способы описаны как идеи и демонстрируют принципиальную возможность вариантов реализации комбинационных СС-схем. Они могут быть использованы при синтезе, но имеют существенные недостатки. Рассмотрим их подробнее в терминах данной книги.

«Реализация по минимальным формам» и «Реализация на элементах И-ИЛИ-НЕ» — это просто переход к ПФС-функциям с помощью МПФС-процедуры и отдельная индикация всех входов. Способы характеризуются повышенными затратами из-за неучета индицирующих свойств информационного канала.

«Реализация по ортогональным формам» предусматривает двухъярусное воплощение ДНФ после ее ортогонализации — приближения к совершенной ДНФ. Затраты при этом способе весьма велики, и требуется отдельная индикация всех входов обоих ярусов. Кроме того, для КМДП-элементов такой способ не имеет смысла, так как они могут реализовать монотонную ДНФ в одном ярусе.

«Реализация с коллективной ответственностью» также ориентирована на двухъярусную реализацию ДНФ. Для КМДП-схем она также не имеет смысла. Способ характеризуется повышенными затратами и пригоден только при наличии дополнительного управляющего сигнала.

«Реализация на элементах типа гистерезисных триггеров» представляет собой реализацию с внутренней памятью. Индикация всех входов делается либо на каждом выходе, либо на каждой паре сопряженных выходов. В обоих случаях индикация избыточна, что и дает повышенные затраты.

«Огульная» индикация, характерная для упомянутых способов, приводит к повышенным затратам не только за счет элементов для получения сигналов индикации, но и за счет последующих неизбежных схем редукации.

Другим общим недостатком перечисленных способов является отсутствие каких-либо возможностей оптимизации, не считая вариантов представления ДНФ.

Таким образом, предложенные способы, на наш взгляд, могут найти лишь ограниченное применение в синтезе.

Ниже предлагается табличный метод синтеза СС-секций [28].

Табличный метод позволяет одновременно выбирать подходящее ПФС-преобразование, детально отслеживать индикацию в секции и даже ее «тонкую структуру» — отдельно индикацию перехода в спейсер и в рабочую фазу. Также в этом методе ясно видны возможности оптимизации как по быстродействию, так и по затратам.

Основной недостаток метода — экспоненциальная зависимость размера таблиц от числа входов секции. Однако, как известно, в обычном синтезе табличные представления, явно или неявно, при том же недостатке используются весьма широко.

В зависимости от числа функций в М-секциях размер таблиц растет линейно. Оценки показывают, что синтез М-секций с числом входов 8—10 и любым числом выходов (функций) может быть успешно выполнен на ПЭВМ среднего быстродействия (требования по памяти невелики). Как следует из опыта, для практических целей таких максимальных параметров М-секций обычно достаточно.

Исходными данными для синтеза будут:

- М-секция, т. е. набор М-функций;
- значения входного и выходного спейсеров;
- указание индикации ПФС-переменных: какие из них должны индексироваться и как (полностью, в спейсере или в рабочей фазе);
- критерий оптимизации: быстродействие или затраты.

### 3.3.5.1. Табличный метод

Табличный метод основан на теоремах из [3], относящихся к комбинационным схемам. Там рассмотрены комбинационные схемы, у которых выходы всех элементов являются и выходами схемы. СС-секции относятся к схемам именно такого типа. Опуская подробности, скажем, что условиями самосинхронности будут: монотонность функций и переходов входных наборов, несравнимость входных и выходных наборов с промежуточными и индексированность входов.

Эти условия, кроме последнего, выполнить легко, и основная задача табличного метода состоит в обеспечении индицируемости входов.

Будем рассматривать специально построенную таблицу истинности  $T$ , в левой части которой будут находиться наборы входных ПФС-сигналов, а в правой — значения на этих наборах искомым ПФС-функций и функций индикаторов.

Для унификации метода примем значения спейсеров  $s_1 = s_2 = 0$ . Другие варианты спейсеров приводятся к данному путем простых преобразований исходных функций и результатов синтеза. Эти преобразования описаны ниже как этап синтеза.

Пусть  $N$  — число входов  $M$ -секции,  $M$  — число ее выходов (функций).

Обозначим:

$x_i$  — входная переменная  $M$ -функций и первая ПФС-переменная,  $i = 1, \dots, N$ ;

$y_i$  — ПФС-переменная, сопряженная  $x_i$ ;

$U_m$  — ПФС-функция — выход СС-секции,  $m = 1, \dots, M$ ;

$V_m$  — функция, сопряженная  $U_m$ , — также выход секции;

$I_k$  — функция индикатора — выход СС-секции,  $k = 1, \dots, K$ .

Будем различать следующие множества (зоны) входных наборов:

$S$  — спейсерная зона (все фазовые входы равны нулю);

$DS$  — шаговая зона спейсера;

$DR$  — шаговая рабочая зона;

$R$  — рабочая зона.

Шаговая зона спейсера представляет собой наборы, отличающиеся от спейсерного «на один шаг», т. е. по одной переменной. Например, для двух ПФС-переменных ( $N = 1$ )  $DS$ -зона будет состоять из двух наборов: 01 и 10.

Таким образом, число наборов (мощность)  $DS$ -зоны составляет  $W(DS) = 2N$ .

Рабочая зона — это все наборы, для которых  $x_i \neq y_i$ . Ее мощность  $W(R) = 2^N$ . Например, при  $N = 2$  это будут наборы 0101, 1001, 0110 и 1010.

Шаговая рабочая зона — это наборы, отличающиеся «на один шаг» от рабочих. Зона состоит из  $N$  групп наборов, соответствующих каждой паре ПФС-переменных. В группу  $DR_i$  входят наборы, для которых  $x_i = y_i = 0$ ,  $x_k \neq y_k$  при  $k \neq i$ ,  $k = 1, \dots, N$ . Например, при  $N = 2$  первая группа содержит наборы 0001 и 0010, вторая — 0100 и 1000. Мощность одной группы  $W(DR_i) = 2^{N-1}$ , мощность всей зоны  $W(DR) = N2^{N-1}$ .

В таблицу  $T$  для синтеза достаточно поместить только зоны  $DS$  и  $DR$ . В зону  $DS$  помещаются только наборы (строки), соответствующи-

шие переменным, подлежащим индикации в спейсере. Аналогично в зону  $DR$  помещаются только группы строк, соответствующих индицируемому в рабочей фазе переменным.

Таким образом, максимальная мощность (сумма строк) таблицы ограничена:

$$W(T) \leq 2N + N2^{N-1}. \quad (3.25)$$

Принципиальным условием табличного метода является монотонность всех ПФС-функций. При принятых нулевых значениях входного и выходного спейсеров все ПФС-функции будут изотонными.

Пусть  $A$  — некоторое множество строк таблицы. Обозначим через  $E(A)$  максимум сумм единиц в строках правой части таблицы  $T$  для этого множества. Из изотонности функций и из того факта, что суммы единиц в наборах каждой из зон в последовательности  $S — DS — DR — R$  не убывают, вытекает справедливость соотношений:

$$E(S) = 0 \leq E(DS) \leq E(DR) \leq E(R) = M. \quad (3.26)$$

Теперь можно сформулировать основные критерии метода — табличные условия индикации переменных. Эти условия являются табличной формой выражения булевых производных выходов секции по переменным в соответствующих фазах.

1. *Условие индикации перехода в спейсер ПФС-переменной  $d$  — наличие хотя бы одной единицы в строке этой переменной в зоне  $DS$ :*

$$E(DS(d)) > 0. \quad (3.27)$$

2. *Условие индикации перехода в рабочую фазу обеих переменных  $x_i$  и  $u_i$  — сумма единиц в группе  $DR_i$  (для каждой строки) меньше такой же суммы в зоне  $R$ :*

$$E(DR_i) < E(R). \quad (3.28)$$

Процесс конструирования выходных функций будет связан с добавлением и удалением единиц в правой части таблицы. Поэтому полезно установить свойства этих действий, важные для синтеза.

Будем говорить, что строка таблицы *поглощает* другую строку, если ее входной набор поглощает входной набор другой строки или, что то же самое, если набор второй строки содержит все единицы и на тех же местах, что и первый набор. Например, строка с набором 0010 поглощает строку с набором 0110. В нашем случае строки в одной какой-либо зоне могут поглощать только строки в других нижележащих зонах (внутри зон строки друг друга не поглощают).



При манипуляциях с единицами в таблице необходимость сохранения свойства изотонности функций влечет за собой такие следствия.

**Следствие 1.** Добавление единицы в какой-либо зоне  $DS$  или  $DR$  в строке  $A$  для любой функции  $Y$  требует добавления единиц для этой же функции в нижележащих зонах  $DR$  и  $R$  в строках  $A_k$ , поглощаемых строкой  $A$ .

Действительно, суммы единиц наборов  $E(A) < E(A_k)$ , и при изотонности  $Y$  должно выполняться  $Y(A) \leq Y(A_k)$ . Но, добавляя единицу, мы делаем  $Y(A) = 1$ , откуда  $Y(A_k) = 1$ .

Добавление единицы упрощает вид функции в форме ДНФ (так как поглощается ряд импликант) и уменьшает затраты при ее реализации, но ухудшает условия индикации в зоне  $DR$ .

**Следствие 2.** Удаление единицы функции в зоне  $DR$  в строке  $A$  требует удаления единиц во всех строках  $A_k$  вышележащей зоны  $DS$ , поглощающих строку  $A$  этой функции.

Удаление единицы в зоне  $DS$  не вызывает удаления других единиц.

Удаление единиц в целом усложняет вид функций, но улучшает условия индикации в зоне  $DR$ .

Сущность табличного метода заключается в нахождении такой конфигурации единиц для ПФС-функций и индикаторов в правой части таблицы, чтобы при соблюдении необходимых ограничений обеспечить необходимую индикацию и возможную оптимизацию.

Далее в разделах 3.3.5.2 и 3.3.5.3 рассмотрены секции, в которых все входы являются ПФС-сигналами. Для них зона  $S$  всегда заполнена нулями, и в таблицу эту зону включать не обязательно.

### 3.3.5.2. СС-секции без памяти

В этом варианте секция будет состоять из вентильных ССБЭ без обратных связей. Пусть задана  $M$ -секция из функций  $F_m$  ( $m = 1, \dots, M$ ).

Для наглядности будем сопровождать изложение простым примером синтеза со следующими исходными данными:

- $M$ -секция:  $F_1 = x_1x_2$ ,  $F_2 = x_2x_3$ ;
- спейсеры  $s_1 = 0$ ,  $s_2 = 1$ ;
- все входы должны быть полностью индцированы.

Синтез производится по шагам.

#### **А. Выполнение изотонной МПФС-процедуры.**

Для удобства расчетов примем такой порядок расположения сопряженных ПФС-функций: первая из них всегда будет соответствовать исходной  $M$ -функции.

Сначала необходимо привести исходные данные к нулевым спейсерам, т. е. из исходных  $M$ -функций получить функции, требуемые для синтеза. При различных комбинациях входных и выходных реальных спейсеров для табличного синтеза следует брать такие пары функций для ПФС-преобразования:

$$\begin{aligned} 00 &- (F_m, \hat{F}_m), \\ 01 &- (\hat{F}_m, F_m), \\ 10 &- (\hat{\Phi}_m, \Phi_m), \\ 11 &- (\Phi_m, \hat{\Phi}_m), \end{aligned}$$

где  $\Phi_m$  — функция, двойственная к  $F_m$ .

Далее, в соответствии с ПФС-преобразованием, необходимо перейти к изотонным начальным для синтеза ПФС-функциям.

В примере начальные ПФС-функции будут иметь вид:

$$\begin{aligned} U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\ U_2 &= y_2 \vee y_3, & V_2 &= x_2 x_3. \end{aligned} \quad (3.29)$$

### **Б. Получение начальной таблицы — таблицы начальных ПФС-функций.**

*Здесь и далее в пустых клетках таблиц подразумеваются нули.*

В таблице 3.4 необязательная зона  $R$  приведена для иллюстрации.

Из начальной таблицы (без учета индикатора) видно, что переменные  $x_1$ ,  $x_2$  и  $x_3$  не индицируются в спейсере, и ни одна из пар переменных не индицируется в рабочей фазе.

При модификации таблиц следует иметь в виду ограничение, вытекающее из Следствия 1: нельзя добавлять единицу в строку  $A$  зон  $DS$  и  $DR$  какой-либо функции, если в зоне  $R$  хотя бы в одной поглощаемой строке  $A_k$  содержится ноль этой функции. В противном случае нарушается правильность ПФС-кодирования.

Из этого следует, что в зоне  $DS$  не может быть других единиц ПФС-функций, кроме имеющихся в начальной таблице.

Далее все последующие шаги могут итеративно повторяться.

### **В. Обеспечение индикации спейсера.**

Этот вид индикации необходимо обеспечивать в первую очередь, так как он связан с добавлением единиц в зоне  $DS$ , что наиболее сильно влияет на дальнейший синтез.

Из ограничения, диктуемого Следствием 1, вытекает, что недостающие для индикации спейсера единицы можно добавлять только в функции индикаторов.

Таблица 3.4

## Пример синтеза СС-секции с одним индикатором

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$	$U_1$	$V_1$	$U_2$	$V_2$	$I_1$
$DS$	1										1
		1					1				
			1								1
				1			1		1		
					1						1
						1			1		
$DR_1$			1		1					1	1
			1			1			1		1
				1	1		1		1		1
				1		1	1		1		
$DR_2$	1				1						1
	1					1			1		1
		1				1	1				1
		1				1	1		1		
$DR_3$	1		1					1			1
	1			1			1		1		1
		1	1				1				1
		1		1			1		1*		
$R$	1		1		1			1		1	1
	1		1			1		1	1		1
	1			1	1		1		1		1
	1			1		1	1		1		1
		1	1			1	1			1	1
		1	1			1	1		1		1
		1		1	1		1		1		1
		1		1		1	1		1		1

Напомним, что введение каждого дополнительного индикатора требует увеличения затрат на редукцию согласно табл. 3.2.

Функции индикаторов в зоне  $R$  должны иметь только единицы. Действительно, назначение индикатора состоит в том, чтобы различать спейсер и рабочую фазу. В данном случае

$$\begin{aligned} I(S) &= 0, \\ I(R) &= 1, \end{aligned}$$

а на промежуточных наборах возможны варианты.

В таблице 3.4 для индикации всех переменных в спейсере в каждой строке зоны  $DS$  должно быть хотя бы по одной единице. В этой зоне недостает трех единиц. Значит, надо вводить от одного до трех индикаторов:  $K = 1, 2, 3$ .

В примере показан один индикатор. Добавление в его функцию трех единиц зоны  $DS$  и соответствующее (согласно Следствиям 1 и 2) добавление единиц в другие зоны позволило корректно обеспечить индикацию спейсера.

#### Г. Обеспечение всей требуемой индикации.

На этом шаге необходимо добиться индикации рабочей фазы, не нарушая индикации спейсера и соблюдая ограничения реализации.

По свойству ПФС-функций и индикаторов, сумма единиц строк в рабочей зоне таблиц одинакова и равна  $E(R) = N + K$ . Поэтому из (3.27) и (3.28) условие индикации для любой строки  $A_k$  в зонах  $DS$  и  $DR$  будет иметь вид

$$0 < E(A_k) < N + K. \quad (3.30)$$

Выполнить это условие можно, манипулируя единицами в таблице и добавляя при необходимости индикаторы. Задача имеет конечное решение, когда таблица — начальная,  $K = N$ , а функции индикаторов равны  $I_i = x_i \vee y_i$  (полная индикация входов).

В таблице 3.4 — один индикатор. Удалим единицы, выделенные курсивом, и задача с одним индикатором решена. Выпишем результат:

$$\begin{aligned} U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\ U_2 &= y_3 \vee y_1 y_2 \vee x_1 y_2 x_3, & V_2 &= x_2 x_3, \\ I_1 &= x_1 \vee x_2 \vee x_3 \vee y_1 y_2 y_3. \end{aligned} \quad (3.31)$$

#### Д. Минимизация и подсчет критериев.

Многие строки таблиц, получаемых в процессе синтеза, удовлетворяют условиям индикации с запасом, что открывает возможность минимизации. Одновременная (совместная) минимизация всех полученных функций практически невозможна из-за большого объема вычислений, и минимизировать приходится каждую функцию отдельно. Поскольку функции зависят друг от друга согласно условию индикации (3.30), порядок минимизации оказывается важным. Здесь возможны различные стратегии. Одна из них — минимизировать функции по порядку убывания сложности.

Для минимизации какой-либо функции  $G$  из полученного результата будем задавать две функции:  $G_E$  — функция единиц,  $G_N$  — функ-

ция нулей и определять  $G_H$  — функцию произвольных значений. Каждая из них равна единице на соответствующих наборах функции  $G$ :

$$G = G_E \vee \wedge G_N G_H. \quad (3.32)$$

Функция произвольных значений расширяет возможности склеивания единиц при минимизации.

В нашем случае функции нулей и единиц определяются по таблице.

Назовем *критическими нулями* зоны  $DR$  наборы  $A_k$  этой зоны, на которых функция  $G(A_k) = 0$  и выполняется условие  $E(A_k) = N + K - 1$ . На месте критических нулей не могут появляться единицы.

При формировании функции  $G$  обязательными единицами для нее будут единицы зон  $DS$  и  $R$ . Обязательными нулями для нее будут нули зоны  $R$  и критические нули зоны  $DR$ . Обязательность объясняется тем, что единицы зоны  $DS$  и критические нули  $DR$  обеспечивают индикацию, а единицы и нули зоны  $R$  — правильность ПФС-функций.

По обязательным единицам и нулям записываются функции  $G_E$  и  $G_N$ , а функция  $G_H$  находится среди промежуточных наборов  $P$ :

$$\begin{aligned} G_H &= \wedge G_E \wedge G_N P, \\ P &= \wedge S \wedge R \wedge Z, \\ Z &= x_1 y_1 \vee \dots \vee x_N y_N, \end{aligned} \quad (3.33)$$

где  $S$  и  $R$  берутся из (3.17) при  $s_1 = 0$ , а  $Z$  — функция запрещенных наборов также при  $s_1 = 0$ .

Для обеспечения изотонности есть две возможности. Первая — поместить в функцию  $G_N$  все промежуточные наборы (не только из зон  $DR$  и  $DS$ ), могущие нарушить (3.30). Тогда (3.33) даст искомые наборы для минимизации. Вторая возможность — в  $G_N$  включаются только обязательные нули функции  $G$ . Тогда необходимо проверять все наборы, определяемые (3.33) на условие (3.30) и отбрасывать неподходящие.

В примере есть смысл минимизировать функции  $U_2$  и  $I_1$ .

Для  $U_2$  функция  $U_{2H} = \wedge x_1 y_1 \wedge x_2 y_2 \wedge x_3 y_3$  (единица в таблице помечена звездочкой, она может быть удалена или сохранена). Минимизация не меняет результата (3.31).

Для индикатора  $I_{1H} = 0$ , что означает отсутствие возможности минимизации.

По результатам (3.31) подсчитаем критерии — максимальный индекс задержки и затраты в транзисторах  $L$ . Затраты на редукцию по табл. 3.2 примем равными 6.

$$\begin{aligned} I_{\max} &= 4, \\ Z &= 34 + 6 = 40. \end{aligned} \quad (3.34)$$

Рассмотрим другой вариант минимизации: переместим единицу, помеченную звездочкой в табл. 3.4, из функции  $U_2$  в функцию  $I_1$ , что не нарушает никаких условий. Получим результат с меньшими затратами:

$$\begin{aligned} U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\ U_2 &= y_3 \vee y_1 y_2 x_3 \vee x_1 y_2 x_3, & V_2 &= x_2 x_3, \\ I_1 &= x_1 \vee x_2 \vee x_3 \vee y_1 y_2. \\ I_{\max} &= 4, \\ L &= 32 + 6 = 38. \end{aligned} \quad (3.35)$$

### Е. Приведение к реальным значениям спейсеров и получение конечного результата.

Для перехода от расчетных нулевых спейсеров к реальным следует последовательно выполнить такие действия:

- если входной спейсер равен 1, каждую функцию результата синтеза заменить ее двойственной;
- если входной и выходной спейсеры различны, инвертировать все полученные функции.

Отметим, что реализация КМДП-элемента для выполнения заданной функции требует одновременной реализации и ее двойственной функции ( $n$ -часть и  $p$ -часть элемента). Поэтому при переходе от результирующих функций к двойственным значения ранее полученных затрат по числу транзисторов сохраняются. Однако, если входной и выходной спейсеры одинаковы, необходимо использовать выходные инверторы и соответственно корректировать критерии.

Для завершения примера приведем результаты синтеза с двумя и тремя индикаторами.

Таблица 3.5 описывает секцию с двумя индикаторами. Клетки для минимизации помечены звездочками. Результат синтеза:

$$\begin{aligned} U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\ U_2 &= y_2 \vee y_3, & V_2 &= x_2 x_3, \\ I_1 &= x_1 \vee y_1, \\ I_2 &= x_2 \vee x_3 \vee y_2 y_3, \\ I_{\max} &= 3, \\ L &= 28 + 12 = 40. \end{aligned} \quad (3.36)$$

Для трех индикаторов (отдельная полная индикация всех входов) результат получается непосредственно:

$$\begin{aligned} U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\ U_2 &= y_2 \vee y_3, & V_2 &= x_2 x_3, \end{aligned} \quad (3.37)$$

$$\begin{aligned}
 I_1 &= x_1 \vee y_1, \\
 I_2 &= x_2 \vee y_2, \\
 I_3 &= x_3 \vee y_3, \\
 I_{\max} &= 2, \\
 L &= 28 + 18 = 46.
 \end{aligned}$$

Как видим, все три варианта (3.35—3.37) имеют различные характеристики, и выбор какого-либо из них зависит от основного критерия оптимизации.

Таблица 3.5

Пример синтеза СС-секции с двумя индикаторами

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$	$U_1$	$V_1$	$U_2$	$V_2$	$I_1$	$I_2$
$DS$	1										1	
		1					1				1	
			1									1
				1			1		1			
					1							1
						1				1		
$DR_1$			1		1					1		1
			1			1			1			1
				1	1		1		1			1
				1		1	1		1			*
$DR_2$	1				1						1	1
	1					1			1		1	*
		1				1	1				1	1
		1					1		1		1	
$DR_3$	1		1					1			1	1
	1			1			1		1		1	
		1	1				1				1	1
		1		1			1		1		1	
$R$	1		1		1			1		1	1	1
	1		1			1		1	1		1	1
	1			1	1		1		1		1	1
	1			1		1	1		1		1	1
		1	1			1	1			1	1	1
		1	1				1		1		1	1
		1		1	1		1		1		1	1
		1		1		1	1		1		1	1

### 3.3.5.3. СС-секции с внутренней памятью

Идея такого рода секции состоит в использовании свойств запоминающих ССБЭ (см. главу 2). Выходные сигналы секции могут запоминаться на период перехода между крайними входными наборами, а сами переходы из спейсера в рабочую фазу и обратно могут обеспечиваться отдельно  $p$ - и  $n$ -частями базовых элементов.

Будем проводить синтез при значениях спейсеров  $s_1 = s_2 = 0$  с применением табличного метода. Уравнения элементов в этом случае всегда изотонны и будут иметь вид триггерных уравнений:

$$W = W_1 \vee W^- W_2, \quad (3.38)$$

где  $W_1$  — функция, определяющая переход из спейсера в рабочую фазу;  $W^-$  — предыдущее значение функции;  $W_2$  — функция, определяющая переход из рабочей фазы в спейсер.

Синтез будет проводиться отдельно для функций  $W_1$  и  $W_2$ .

На совокупность функций  $W_1$  секции накладывается условие

$$0 \leq E(DS) \leq E(DR) < E(R) = M, \quad (3.39)$$

а на совокупность функций  $W_2$  — условие

$$0 < E(DS) \leq E(DR) \leq E(R) = M. \quad (3.40)$$

Эти условия — более слабые, чем пара неравенств (3.27) и (3.28). Отличие состоит в том, что условия (3.27) и (3.28) накладываются на одни и те же функции, а условия (3.39) и (3.40) — на разные функции (каждое на свои). В остальном табличный метод здесь используется так же, как и для секций без запоминания.

Остановимся подробнее на синтезе функций  $W_2$ . В условии (3.40) зона  $DS$  — шаговая зона спейсера, поэтому функции  $W_2$  всегда будут иметь вид дизъюнкций входных ПФС-переменных (не обязательно всех).

Функцию  $W_2$  в (3.38) назовем *спейсерной частью* ( $S$ -частью) функции  $W$ .

Совокупность требований к  $S$ -частям функций секции позволяет сформулировать условие индикации спейсера, гарантирующее выполнение крайней левой части неравенства (3.40).

**Условие индикации спейсера:** каждая входная ПФС-переменная должна входить хотя бы в одну из  $S$ -частей тех функций, которые равны единице в рабочей фазе.

Выполнить это условие можно по-разному. Самый простой способ — включить все переменные в каждую  $S$ -часть. Однако это избыточно и в большинстве случаев нереализуемо из-за ограничений на базовые элементы.



Более экономный способ состоит в следующем. В рабочей фазе одна из каждой пары сопряженных ПФС-функций равна единице. Переменные можно помещать в  $S$ -части обеих функций пары, что дает основание следующему правилу.

**Правило 1.** Для индицирования какой-либо переменной в спейсере достаточно поместить ее в обе  $S$ -части одной пары сопряженных ПФС-функций. При этом данную переменную можно исключить из остальных  $S$ -частей.

Возможен и другой способ индицирования в спейсере.

**Правило 2.** Для индицирования всех переменных в спейсере достаточно поместить в  $S$ -часть каждой функции (3.38) те переменные, от которых зависит выражение  $W_1$  этой функции.

Докажем правило. Поскольку любая переменная  $x$  и ее сопряженная  $y$  индицируются в рабочей фазе, в секции всегда есть пара ПФС-функций вида (3.38):

$$\begin{aligned} U &= U_1 \vee U^-U_2, \\ V &= V_1 \vee V^-V_2, \end{aligned} \quad (3.41)$$

для которых

$$\begin{aligned} U_1 &= xG_1 \vee yG_2, \\ V_1 &= yG_1 \vee xG_2 \end{aligned} \quad (3.42)$$

и в рабочей фазе выполняется условие  $G_1 \oplus G_2 = 1$ .

Таким образом, для любой переменной  $x$  в секции всегда найдется функция, от нее зависящая и равная единице в рабочей фазе. Это будет функция  $U$ , если в рабочей фазе  $G_1 = 1$ , или  $V$ , если там же  $G_2 = 1$ .

Если теперь, согласно Правилу 2, поместить переменную  $x$  в  $S$ -часть каждой функции, от нее зависящей, то тем самым будет выполнено условие индикации спейсера для этой переменной. Аналогичное рассуждение справедливо для всех остальных входных переменных, что и доказывает справедливость Правила 2.

Наконец, в ряде случаев полезны следующие два правила.

**Правило 3.** Если функция  $W_1$  представима в виде  $W_1 = z \vee W_{11}$ , где  $W_{11}$  не зависит от  $z$ , то переменную  $z$  можно исключить из остальных  $S$ -частей секции.

Справедливость правила следует из того, что в этом случае функция  $W_1$  обеспечивает также и индикацию переменной  $z$  при переходе в спейсер.

**Правило 4.** Добавление переменных в дизъюнкцию  $W_2$  не ухудшает индикацию, поэтому оно всегда допустимо.

При синтезе правила 1—4 можно комбинировать.

Рассмотрим синтез секции с памятью по шагам и с тем же примером.

### А. Выполнение изотонной МПФС-процедуры.

Этот шаг совпадает с таким же шагом для секции без памяти, и в результате получаются уравнения (3.29).

### Б. Получение начальной таблицы.

Аналогично предыдущему примеру, это таблица ПФС-функций, определяющих переход из спейсера в рабочую фазу ( $W_1$  в форме (3.38)).

### В. Обеспечение индикации рабочей фазы.

На правую часть таблицы соответственно будет накладываться только условие (3.40). Производится изотонная минимизация при этом условии. Правила предыдущего раздела, связанные с изотонностью, справедливы и здесь.

В соответствии с (3.40), обязательными единицами для минимизации являются только единицы зоны  $R$ . Обязательными нулями являются нули зоны  $R$  и критические нули зоны  $DR$ .

В таблице 3.6 звездочками помечены клетки для минимизации, а рядом показан результат этой минимизации.

Искомые функции примера для рабочей фазы выглядят так:

$$\begin{aligned} U_1 &= x_1 x_2, & V_1 &= x_1 y_2 \vee y_1 x_2 \vee y_1 y_2, \\ U_2 &= x_2 x_3, & V_2 &= y_3 \vee y_2 x_3. \end{aligned} \quad (3.43)$$

### Г. Обеспечение индикации перехода в спейсер.

На этом шаге необходимо сконструировать подходящие функции  $W_2$  формы (3.38) (дизъюнкции переменных) по Правилам 1—4. Поскольку правила просты, никакой таблицы не требуется.

Алгоритм действий также достаточно прост: для каждой функции  $W$  по Правилу 2 выписываются переменные, от которых зависят полученные на предыдущем шаге выражения  $W_1$ . Далее по остальным правилам производятся упрощения с учетом критерия оптимизации.

Перечни переменных для функций по (3.43) имеют вид:

$$\begin{aligned} U_1: & x_1 x_2, \\ V_1: & x_1 x_2 y_1 y_2, \\ U_2: & x_2 x_3, \\ V_2: & y_2 x_3 y_3. \end{aligned}$$

По Правилу 1 удалим  $x_2$  из списка функции  $U_2$  и туда же добавим  $y_2$ . Далее по тому же правилу удалим  $y_2$  из списка функции  $V_1$ .

Таблица 3.6

Пример СС-секции с внутренней памятью (для рабочей фазы)

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$	$U_1$	$V_1$	$U_2$	$V_2$
$DS$	1									
		1						*0		
			1							
				1				*0		*0
					1					
						1				*1
$DR_1$			1		1				*1	
			1			1				*1
				1	1			*0		*1
				1		1		*0		*1
$DR_2$	1				1					
	1					1				*1
		1			1			*0		
		1				1		*0		*1
$DR_3$	1		1				*1			
	1			1				*1		*0
		1	1					*1		
		1		1				*1		*0
$R$	1		1		1		1		1	
	1		1			1	1			1
	1			1	1			1		1
	1			1		1		1		1
		1	1		1			1	1	
		1	1			1		1		1
		1		1	1			1		1
		1		1		1		1		1

По Правилу 3 из списка  $V_2$  удаляем переменную  $y_3$  (впрочем, в данном случае она автоматически исчезнет в конечном выражении из-за поглощения). В результате получим решение:

$$\begin{aligned}
 U_1 &= x_1 x_2 \vee U_1^-(x_1 \vee x_2), \\
 V_1 &= x_1 y_2 \vee y_1 x_2 \vee y_1 y_2 \vee V_1^-(x_1 \vee x_2 \vee y_1), \\
 U_2 &= x_2 x_3 \vee U_2^-(y_2 \vee x_3), \\
 V_2 &= y_3 \vee y_2 x_3 \vee V_2^-(y_2 \vee x_3).
 \end{aligned}
 \tag{3.44}$$

При реализации на запоминающих ССБЭ необходимо обеспечить условие отсутствия сквозных токов, что в данном случае приводит к еще одному правилу.

**Правило 5.** Для отсутствия сквозных токов в запоминающих ССБЭ необходимо наличие в  $S$ -части каждой функции  $W_2$  хотя бы по одной переменной из каждой импликанты (конъюнкции), входящей в выражение  $W_1$ .

В нашем примере в решении (3.44) это условие выполняется везде, кроме последней функции, в  $S$ -часть которой надо добавить переменную  $y_3$ :

$$V_2 = y_3 \vee y_2 x_3 \vee V_2^-(y_2 \vee x_3 \vee y_3).$$

В результате реализация на запоминающих ССБЭ дает:

$$\begin{aligned} I_{\max} &= 4, \\ L &= 38. \end{aligned} \tag{3.45}$$

Отметим, что если применить способ «реализации на элементах типа Г-триггеров», описанный в [2], то затраты составят 68 транзисторов.

Таким образом, табличный метод синтеза комбинационных СС-секций позволяет точно отследить и обеспечить индикацию переходов отдельно в рабочую фазу и спейсер и на основе этого оптимизировать секции по быстродействию и затратам.

#### 3.3.5.4. СС-секции со смешанными входами

Можно построить СС-секции, в которых одна часть входов — фазовые сигналы, а другая — бистабильные. Будем называть их секциями со смешанными входами, или *смешанными СС-секциями*. Вообще говоря, без таких секций можно обойтись, преобразовав БС-сигналы в ПФС-сигналы (см. далее в разделе 3.4.3 рис. 3.8) и используя предыдущие способы. Однако в смешанной секции совмещаются и вычисление функций, и преобразование БС — ПФС, что в большинстве случаев снижает затраты по числу транзисторов. Особенно актуальны смешанные секции при создании СС-автоматов (см. раздел 3.5.3), в которых всегда требуется участие БС-сигналов при вычислении функций.

Для смешанных СС-секций должны выполняться несколько необходимых условий. Очевидными условиями будут следующие. Во-первых, выходы секции по-прежнему должны быть ПФС-сигналами, иначе они не могут индексироваться в дальнейшем. Во-вторых, на входах секции должен присутствовать хотя бы один фазовый сигнал для обеспечения чередования фаз.

Еще одно условие связано с монотонностью функций и входных переходов. При переходе из спейсера в рабочую фазу и обратно в каждой паре ПФС-сигналов меняется один и только один сигнал, и все изменения во всех парах идут в одну сторону: из 0 в 1 или из 1 в 0 (монотонность переходов). Но в паре БС-сигналов в одной из фаз меняются оба сигнала — один из 0 в 1, другой из 1 в 0, что нарушает монотонность. Для правильной работы секции необходимо сделать так, чтобы в той фазе, когда входные БС-сигналы меняются, выходы секции были заблокированы (не могли изменяться). Такой фазой в данном случае будет спейсер, а блокирующими сигналами будут фазовые сигналы. Таким образом, должно выполняться следующее правило.

*БС-сигналы на входе смешанной СС-секции должны меняться только в спейсере.*

Это условие должно обеспечиваться внешней схемой. Блокировка выходов будет обеспечена подбором функций секции с помощью табличного метода.

Выполнение приведенного условия означает, что при построении таблицы БС-сигналы могут считаться константами, и для них должен быть сделан полный перебор значений.

В отличие от предыдущих, в таблицах для смешанных секций должна явно присутствовать зона  $S$ . Также для каждого сочетания значений ПФС-входов должны отдельными строками присутствовать все сочетания значений БС-входов.

Рассмотрим предыдущий пример секции с уравнениями (3.29), когда переменные  $(x_3, y_3)$  являются БС-входами. Начальное распределение единиц по уравнениям (3.29) приведено в табл. 3.7.

Поскольку ПФС-пар имеется только две, зоны  $DS$  и  $DR$  в таблице совпадают.

Из таблицы видно, что в зоне  $S$  присутствует единица, что недопустимо. Для устранения единиц в подобных случаях есть только один способ: умножить функцию, дающую единицу, на дизъюнкцию сигналов какой-либо ПФС-пары. При этом единица в спейсерной зоне исчезает, а функционирование, определяемое единицами рабочей зоны, не искажается.

В данном случае лучше умножить функцию  $U_2$  на дизъюнкцию  $(x_2 \vee y_2)$ , что дает несколько меньшие затраты, чем при использовании другой ПФС-пары. В результате в таблице исчезнут единицы, отмеченные звездочками.

Далее, таблица показывает, что зона  $DS-DR$  имеет нулевые строки, и необходимо использовать индикаторы. В данном примере

Таблица 3.7

## Исходные данные смешанной СС-секции

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$	$U_1$	$V_1$	$U_2$	$V_2$
$S$					1					
						1			1*	
$DS$ $DR$			1		1					1
			1			1			1	
				1	1		1		1	
				1		1	1		1	
	1					1				
	1								1*	
		1				1		1		
$R$			1		1			1		1
	1		1			1		1	1	
	1			1	1		1		1	
	1			1		1	1		1	
		1	1			1	1			1
		1	1				1		1	
		1			1	1		1		1

достаточно одного индикатора, так как зона  $DS-DR$  допускает, с учетом индикатора, или по одной, или по две единицы в каждой строке.

Окончательное расположение единиц примера приведено в табл. 3.8, а уравнения секции имеют вид:

$$\begin{aligned}
 U_1 &= y_1 \vee y_2, & V_1 &= x_1 x_2, \\
 U_2 &= y_2 \vee x_2 y_3, & V_2 &= x_2 x_3, \\
 I_1 &= x_1 \vee y_1.
 \end{aligned}
 \tag{3.46}$$

Наконец, рассмотрим предельный случай, когда все информационные входы секции — БС-сигналы. Тогда на входе необходимо присутствие хотя бы одного  $У$ -сигнала. Правильность построения секции обеспечивается, если правые части всех уравнений секции умножаются на  $У$ -сигналы.

В предыдущем примере введем один входной  $У$ -сигнал  $c$ . Уравнения такой смешанной СС-секции запишутся так:

$$\begin{aligned}
 U_1 &= (y_1 \vee y_2)c, & V_1 &= x_1 x_2 c, \\
 U_2 &= (y_2 \vee y_3)c, & V_2 &= x_2 x_3 c.
 \end{aligned}
 \tag{3.47}$$

Таблица 3.8

**Окончательные данные смешанной СС-секции**

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$	$U_1$	$V_1$	$U_2$	$V_2$	$I_1$
<i>S</i>					1						
						1					
<i>DS</i> <i>DR</i>			1		1					1	
			1			1			1		
				1	1		1		1		
				1		1	1		1		
	1					1					1
	1						1				1
		1				1					
<i>R</i>		1				1		1			1
		1		1				1	1		1
		1			1	1		1			1
		1			1		1		1		1
			1	1		1				1	1
			1	1			1		1		1
			1		1	1			1		1
			1		1		1		1		1

Таблицу для этого случая приводить нет смысла. Она будет содержать зону *S* со всеми нулями и зону *R*, такую же, как и в других таблицах. Дополнительного индикатора здесь не требуется.

**3.4. Правила манипулирования сигналами в СС-схемах**

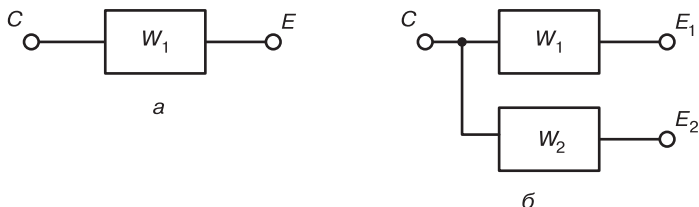
На практике постоянно возникают ситуации, когда сигналы в схеме необходимо усилить, разветвить, направить по линии с существенной задержкой. В обычных синхронных и многих асинхронных схемах нет принципиальных ограничений на подобные операции.

В СС-схемах, однако, такие ограничения существуют, что связано с условиями модели Маллера и требованиями самосинхронности. Можно сформулировать ряд правил, при соблюдении которых требования самосинхронности не нарушаются. Эти правила имеют локальный характер и облегчают построение СС-схем.

В правилах используются свойства индикации (см. раздел 2.5.2).

### 3.4.1. Контрольные сигналы

Контрольные сигналы могут быть инвертированы и/или усилены, как показано на рис. 3.6, *а*. В этом случае обязательной индикации подлежит сигнал  $E$ .



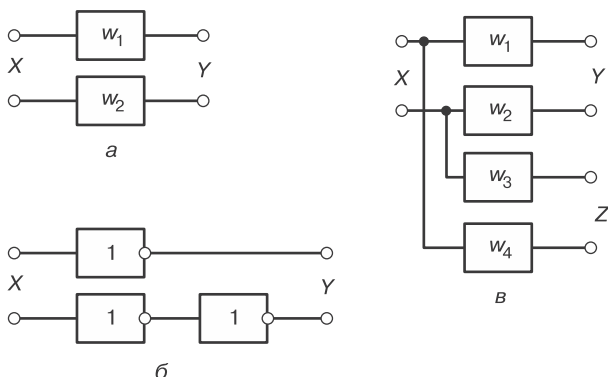
**Рис. 3.6.** Контрольные сигналы с инверсией/усилением:  
 $w_i$  — любое количество инверторов/повторителей

При разветвлении с инверсией/усилением, показанном на рис. 3.6, *б*, индцироваться должны сигналы  $E_1$  и  $E_2$ .

В обоих случаях, по свойству транзитивности, индикация сигнала  $C$  не обязательна. Необходимо учитывать также возможное изменение значения спейсеров выходных сигналов.

### 3.4.2. ПФС-сигналы

На рис. 3.7 показаны возможные действия с ПФС-сигналами.



**Рис. 3.7.** Схемы с ПФС-сигналами:  
*а* — с симметричным усилением/повторением; *б* — с несимметричным усилением/повторением; *в* — разветвление;  
 $w_i$  — инверторы и/или повторители

Блоки  $W_1$  и  $W_2$  на рисунке должны быть оба либо инверторами, либо повторителями, так же, как и блоки  $W_3$  и  $W_4$ . Несимметричная



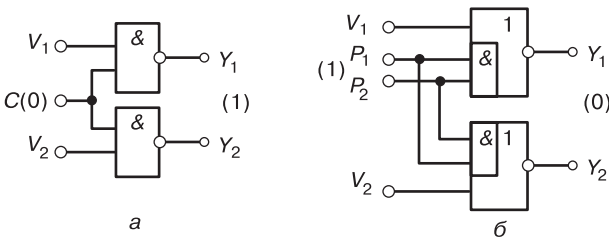
схема  $\bar{b}$ , хотя и может использоваться, но для индикации в один из каналов придется добавить инвертор, т. е. опять получить симметричную схему. Поэтому несимметричные схемы рассматривать не имеет смысла. В схемах  $a$  и  $\bar{b}$  индикации подлежат ПФС-сигналы  $Y$  и  $Z$ , например, с помощью локальной индикации (см. рис. 3.1). Сигнал  $X$  индцировать не обязательно.

### 3.4.3. Бистабильные сигналы

БС-сигналы имеют две особенности, в корне отличающие их от ПФС-сигналов с точки зрения самосинхронности.

Во-первых, БС-сигналы, в отличие от ПФС-сигналов, не несут информации о текущей фазе — в любой фазе они могут иметь любые установившиеся (взаимно противоположные) значения. Если в цепи БС-сигналов включить дополнительные элементы — инверторы, усилители или задержки, то выходы таких дополнительных элементов уже не могут быть индцированы, поскольку также не несут информации о фазе (сами исходные БС-сигналы индцируются в составе БСЯ, которая их порождает, — см. раздел 2.1.1).

Из-за этой особенности БС-сигналы не допускают простых операций: инвертирования, усиления или задержек. Но на практике они в ряде случаев необходимы и могут быть решены схемотехнически. Одно из решений — специальная схема триггера с усиленными выходами (см. главу 4). Другое простое решение — преобразование БС-сигнала в ПФС-сигнал (рис. 3.8).



**Рис. 3.8.** Преобразователи БС-сигнала в ПФС-сигнал:

$a$  — с блокирующим У-сигналом;  $\bar{b}$  — с блокирующим ПФС-сигналом;  
 $V_1, V_2$  — БС-сигналы;  $P_1, P_2$  и  $Y_1, Y_2$  — ПФС-сигналы;  $C$  — У-сигнал

Во-вторых, БС-сигналы отличаются характером их изменения. Состояние, в котором компоненты БС-сигнала равны (*транзитное состояние*), является неустойчивым, и обе компоненты меняются в течение одной фазы (*транзитной*). (Напомним, что у ПФС-сигнала в каждой фазе меняется лишь один из его компонентов.)

Если позволить элементам, к которым подключены БС-сигналы (их последователям), изменяться в транзитной фазе, то это будет порождать гонки. Поэтому при подаче БС-сигнала на какие-либо элементы необходимо блокировать их в транзитной фазе БС-сигнала. Блокирующими должны быть сигналы, имеющие фазу, — управляющие или ПФС-сигналы.

Иллюстрацией применения этого правила могут служить схемы преобразования БС-сигнала в ПФС-сигнал (см. рис. 3.8). На рисунке транзитные фазы БС-сигналов — это спейсеры соответствующих фазовых входов.

### 3.5. Создание последовательностных СС-схем

В книгах [2, 3] и других публикациях тех же авторов довольно подробно рассмотрены вопросы построения последовательностных схем: дано много схем с памятью, приведены запоминающие СС-ячейки, варианты схем автоматов на их основе («стандартные реализации»). Однако, с точки зрения практической схемотехники, эти материалы недостаточны, а иногда и неприемлемы. Недостаточна систематизация основных ячеек памяти, не определены их особенности в СС-схемотехнике, вследствие чего неясно, какие из них и в каких случаях следует применять. Отсутствует анализ схем по критериям оптимальности. Ряд схем, например, на основе ячеек с гашением либо вообще не могут использоваться в двухфазной дисциплине, либо неэффективны по быстродействию и/или затратам.

Сказанное приводит к необходимости более детального рассмотрения запоминающих СС-схем с практической точки зрения.

Приводимые в этом разделе схемы представляют один из двух возможных — двойственных — вариантов реализации. Любой из этих вариантов может быть получен из другого заменой всех операций И на ИЛИ и наоборот, в результате чего значения спейсеров и транзитных состояний БСЯ заменятся на противоположные. Из-за принципиального различия таких вариантов будет приводиться только один из них.

#### 3.5.1. Использование бистабильных ячеек

В основе самосинхронных запоминающих ячеек лежат БСЯ (см. рис. 2.3, *a*). В зависимости от входных сигналов БСЯ может находиться в одном из трех режимов: записи, хранения и гашения. При переходе

в режим гашения информация ячейки теряется. Первые два режима используются всегда, гашение может применяться дополнительно.

На ячейках с гашением реализован ряд СС-схем в [2, 3]. При построении подобных схем приходится учитывать две трудности. Во-первых, в режиме гашения информация теряется и для сохранения ее надо куда-то переписать. Во-вторых, три режима БСЯ должны уложиться в две фазы работы СС-схемы. Это приводит к необходимости реализации триггера с гашением на трех БСЯ с обратными связями между ними, что делает схемы более сложными и менее быстродействующими. Так, *D*-триггер с гашением ([3, рис. 4.11, *a*]) имеет затраты по числу транзисторов на 26% больше, чем триггер без гашения, а индекс задержки — 13 против 8.

Сравнить эти два типа БСЯ можно также на примере счетных триггеров с гашением и без гашения [3, рис. 4.12, *a*, *b* и рис. 4.18, 4.19]. Затраты транзисторов в обоих вариантах триггеров одинаковы, однако быстродействие схемы с гашением в 1,5 раза хуже, чем схемы без гашения.

Использование БСЯ без гашения представляется более естественным: они имеют только два режима — записи и хранения, что идеально согласуется с двумя фазами работы СС-схем. По основным критериям — быстродействию и затратам — схемы без гашения превосходят таковые с гашением. Это обстоятельство необходимо учитывать при создании СС-схем.

Следует сказать и о терминах «*RS-триггер*» и «*D-триггер*» в СС-схемотехнике. Легко убедиться, что при парафазном кодировании *D*-триггер функционально идентичен триггеру, в обычной схемотехнике называемому *RS*-триггером. Поэтому в СС-схемотехнике название «*RS*-триггер» теряет смысл, хотя и употребляется как дань традиции.

### 3.5.2. Самосинхронные запоминающие ячейки

К самосинхронным запоминающим ячейкам (ССЗЯ) предъявляются следующие основные требования.

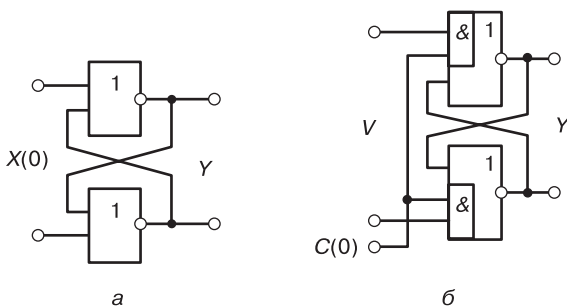
1. ССЗЯ должна хранить один бит информации после окончания фазы, в которой произошла запись этого бита.

2. Окончание переходных процессов должно индицироваться в обеих фазах — рабочей и спейсере.

3. ССЗЯ должна иметь максимальное быстродействие и /или минимальную сложность (в транзисторах) среди возможных вариантов.

Оказалось, что выбор по последнему требованию однозначен: все представленные ниже ячейки являются наилучшими и по быстродействию, и по затратам из известных на данный момент вариантов.

Суммируя сказанное выше о режимах БСЯ и БС-сигналах, можно заключить, что в запоминающих ячейках целесообразно использовать две показанные на рис. 3.9 разновидности БСЯ — неуправляемую и управляемую отдельным сигналом. Входные и выходные сигналы этих БСЯ будут индексироваться в составе триггеров, куда эти ячейки входят.



**Рис. 3.9.** Бистабильные ячейки:

*a* — неуправляемая; *б* — управляемая;

$X$  — ПФС-сигнал;  $V$  — БС-сигнал;  $C$  —  $Y$ -сигнал;  $Y$  — БС-сигнал

Поскольку БС-сигнал не несет информацию о фазе, управляющий сигнал нужен для того, чтобы разрешить запись в ячейку в требуемой фазе и предотвратить гонки.

Таким образом, управляемая и неуправляемая ячейки альтернативны по использованию: неуправляемая применяется для запоминания ПФС-сигналов, управляемая — для БС-сигналов.

Перейдем теперь к построению ССЗЯ на основе БСЯ.

Функционально ССЗЯ являются триггерами, имеющими на входе один инфосигнал. Существует два типа таких триггеров —  $D$  и  $T$ . Самосинхронный  $T$ -триггер (как и обычный асинхронный его аналог) легко получается из  $D$ -триггера с помощью обратных связей. Другие, более сложные триггеры также могут быть построены как автоматы на основе  $D$ -триггеров. Таким образом, за элементарную ССЗЯ можно принять  $D$ -триггер.

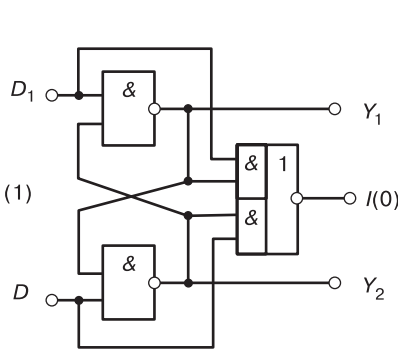
Все преобразования инфосигналов в СС-схемах происходят в рабочей фазе, в том числе и запись информации в ССЗЯ. Хранение же, в зависимости от типа ячейки, реализуется либо только в последующей фазе — спейсере, либо также и в следующей за спейсером рабочей фазе.

ССЗЯ можно классифицировать по основным признакам, существенным для СС-схемотехники.

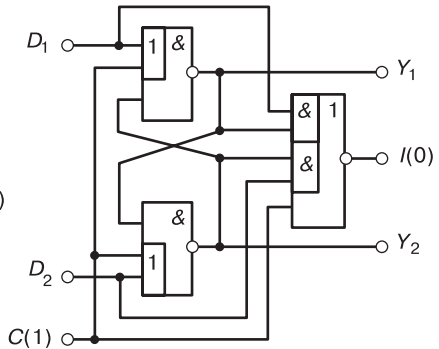
1. Длительность хранения записанной информации — в каких последующих фазах она хранится: только в спейсере (одноступенчатая ячейка) или в спейсере и рабочей фазе (двухступенчатая ячейка).
2. Наличие управляющего сигнала.
3. Значение входного спейсера.

В ряде применений достаточно, чтобы информация, записанная в ячейку в рабочей фазе, хранилась только в течение последующей фазы спейсера, а в очередной рабочей фазе вновь менялась. В частности, эти применения связаны с использованием комбинационных схем в конвейерах, а также с выходом на внешние устройства: исполнительные механизмы, индикаторы (световые, звуковые) и т. д. Для подобных применений можно использовать одноступенчатые ССЗЯ.

Одноступенчатая ячейка представляет собой БСЯ, снабженную индикатором. Выходной спейсер И-сигнала в одноступенчатых ячейках всегда противоположен входному. Схемы одноступенчатых ячеек — неуправляемого и управляемого  $D$ -триггеров [3] приведены соответственно на рис. 3.10 и 3.11.



**Рис. 3.10.** Одноступенчатый  $D$ -триггер с ПФС-входами



**Рис. 3.11.** Одноступенчатый  $D$ -триггер с БС-входами и  $Y$ -сигналом

Поскольку функциональные преобразования в ячейках происходят в рабочей фазе, для реализации автоматных функций с памятью необходимо хранить информацию в ССЗЯ не только в последующем спейсере, но и в следующей за ним рабочей фазе. Для этого необходимо использовать двухступенчатые ячейки.

Двухступенчатая ячейка строится из двух одноступенчатых по принципу *Master — Slave* с объединенным индикатором. В рабочей фазе информация со входов триггера записывается в первую ступень. Входы второй ступени при этом блокируются. В фазе спейсера бло-

кируются входы первой ступени, информация из нее переписывается во вторую ступень и хранится там в последующей рабочей фазе.

На рис. 3.12 и 3.13 показаны неуправляемая и управляемая двухступенчатые ячейки [3] — двухступенчатые  $D$ -триггеры.

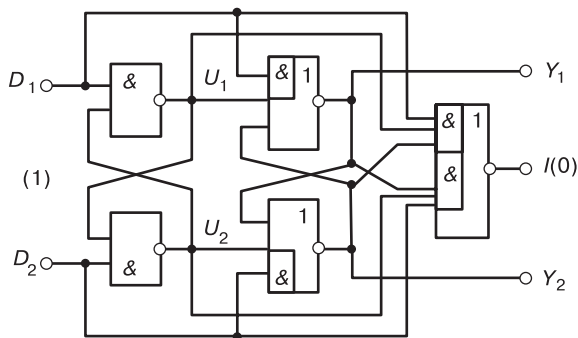


Рис. 3.12. Двухступенчатый  $D$ -триггер с ПФС-входами

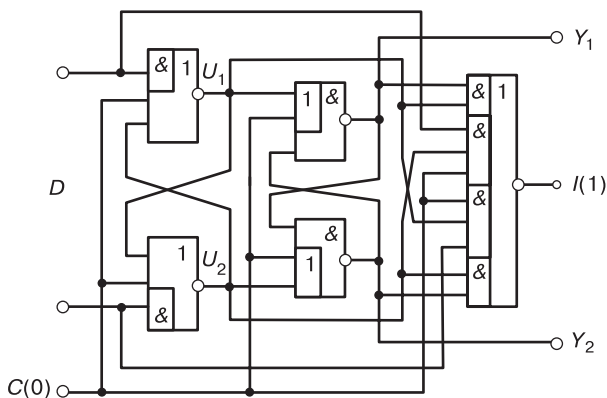


Рис. 3.13. Двухступенчатый  $D$ -триггер с БС-входами и  $U$ -сигналом

Во всех представленных выше триггерах входные фазовые сигналы индицируются на индикаторном выходе и отдельной индикации не требуют. Выходные БС-сигналы также индицируются на индикаторе. Что касается входных БС-сигналов, то они должны индицироваться во внешних ячейках, их порождающих.

Описанные триггеры служат основными ячейками для построения СС-схем с памятью. Другие разновидности ССЗЯ могут быть получены из показанных путем модификации. Примеры можно увидеть в главе 4.

### 3.5.3. Самосинхронные автоматы

Самосинхронные автоматы рассмотрим в форме автомата Мура [29], в котором функциональные преобразования происходят на входе, до запоминания, а выходами автомата являются выходы запоминающих ячеек.

Структура Мура (как и Мили) является универсальной, по ней можно реализовать любые автоматы. Функции автомата синтезируются известными методами, а переход к СС-реализации делается на последнем этапе разработки.

В книге [3] приводятся обобщенные структуры автоматов Мура, однако ряд вопросов, существенных при их практической реализации, не упоминается.

Для наглядности рассмотрим особенности построения СС-автоматов на примере — аналоге *JK*-триггера, прочно вошедшего во все учебники по электронике. Традиционное применение *JK*-триггера связано с тем, что, комбинируя его входы, можно реализовать функции или *RS*-, или *T*-триггера, и подобное применение давно утратило практический смысл. В нем входы *J* и *K* взаимосвязаны, что не характерно для автоматов. Поэтому в данном примере входы *J* и *K* будут считаться независимыми, а сам автомат назовем *JK-автоматом*.

Характеристическая функция *JK*-автомата будет иметь вид:

$$Q^+ = J \wedge Q \vee \bar{K} Q, \quad (3.48)$$

где  $Q$  и  $Q^+$  — предыдущее и последующее хранимые значения.

В СС-реализации к уравнению (3.48) надо применить ПФС-преобразование (см. раздел 3.3.1), и сигналы *J*, *K* и *Q* будут представлены в парафазном виде, т. е. парами сигналов.

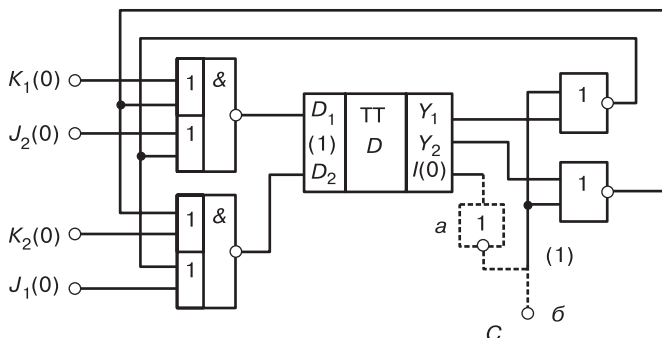
При двухфазной дисциплине функциональное преобразование происходит только в одной из фаз — рабочей, поэтому понятие «предыдущее значение» относится к предыдущей рабочей фазе. Это значит, что ССЗЯ в СС-автоматах должны помнить информацию от одной рабочей фазы через спейсер до другой рабочей, т. е. быть двухступенчатыми (см. 3.5.2).

Существенная особенность СС-реализаций функций типа (3.48) состоит в том, что запомненные сигналы  $Q$  являются, во-первых, БС-сигналами и, во-вторых, сигналами обратной связи в схеме.

Правую часть (3.48) необходимо реализовывать как комбинационную СС-секцию одним из описанных выше способов (см. раздел 3.3.5). Эти способы зависят от того, какого типа сигналы поступают на входы секции — ПФС или БС.

Сначала рассмотрим случай, когда сигналы  $J$  и  $K$  закодированы в ПФС-коде, а сигнал  $Q$  (представленный выходами триггера) преобразуется из БС в ПФС-форму, как показано на рис. 3.8,  $a$ .

Результатом будет схема, показанная на рис. 3.14.



**Рис. 3.14.**  $JK$ -автомат с ПФС-сигналами (некорректный):

$a$  — вариант преобразования с использованием И-сигнала ССЗЯ;

$б$  — вариант преобразования с внешним И-сигналом;

ТТ — двухступенчатый  $D$ -триггер (см. рис. 3.12);  $J_1, J_2$  и  $K_1, K_2$  — ПФС-входы;  $C$  — внешний  $Y$ -сигнал;  $Y_1, Y_2$  — выходы триггера и автомата

Преобразовать БС-выходы триггера в ПФС-сигналы здесь можно двумя путями: с использованием либо И-сигнала самого триггера, либо независимого  $Y$ -сигнала — соответственно варианты  $a$  и  $б$  на рисунке (в варианте  $a$  необходим согласующий инвертор).

Покажем, что в первом варианте схема попадает в тупик. Предположим, она находится в фазе спейсера. Чтобы перейти в рабочую фазу, необходимо, чтобы в нее перешли все входы комбинационной секции, а потом и входы триггера. Но секция не может перейти в рабочую фазу ибо сигналы обратной связи находятся в фазе спейсера до тех пор, пока не перейдет в рабочую фазу триггер. А входы триггера остаются в спейсере из-за комбинационной части. Схема неработоспособна.

При использовании независимого  $Y$ -сигнала  $C$  схема не попадает в тупик. По принципу самосинхронности внешние фазовые сигналы не зависят друг от друга, и сигнал  $C$  может измениться как раньше, так и позже информационных входов. Предположим, схема установилась в рабочую фазу. Пусть информационные входы перешли в спейсер, а сигнал  $C$  — еще нет. Входы триггера переходят в спейсер, и информация внутри него переписывается из первой ступени во вторую. Сигналы  $Y_1$  и  $Y_2$  — выходы второй ступени — изменяются. Сигнал  $C$  пока остается в рабочей фазе, т. е. равным нулю, и позволяет меняться сигналам обратной связи на входе комбинационной



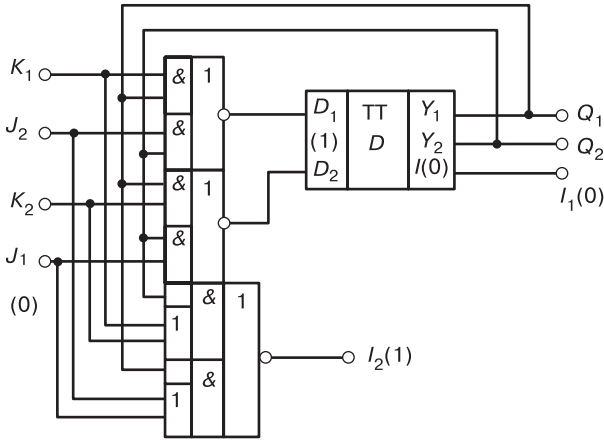
части. Уже эта ситуация порождает гонки в комбинационной части. Но в какой-то момент должен измениться и сигнал  $C$ . Один из сигналов обратной связи, успевший перейти из 0 в 1, будет возвращен обратно в 0, что означает гонку в преобразователе БС-ПФС. Второй способ преобразования также оказывается неприемлемым.

Таким образом, можно сделать следующий вывод.

*Использование в СС-автоматах сигналов обратной связи, преобразованных из БС в ПФС-форму, приводит к некорректной реализации.*

Корректные реализации следует получать на основе смешанных секций (см. раздел 3.3.5.4).

На рис. 3.15 приведена корректная схема  $JK$ -автомата с входными ПФС-сигналами и смешанной СС-секцией.



**Рис. 3.15.** Корректный  $JK$ -автомат с ПФС-входами; ТТ — двухступенчатый  $D$ -триггер (см. рис. 3.12);  $J_1, J_2$  и  $K_1, K_2$  — ПФС-входы;  $Q_1, Q_2$  — БС-выходы;  $I_1, I_2$  — И-сигналы

Приведем для сравнения еще две корректные реализации  $JK$ -автомата (в виде уравнений), отличающиеся типом входных сигналов.

Если входы ( $J_1, J_2$ ) — ПФС, а ( $K_1, K_2$ ) — БС-сигналы, то уравнения комбинационной части записываются так:

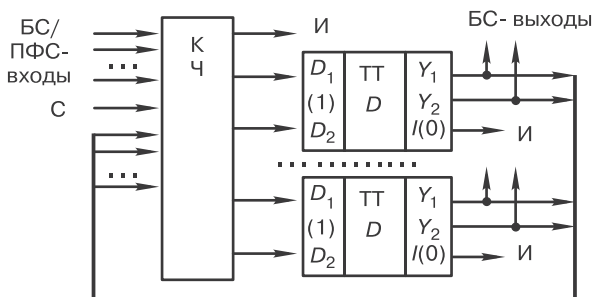
$$\begin{aligned} D_1 &= \wedge[(K_1 Q_1 \vee J_2 Q_2)(J_1 \vee J_2)], \\ D_2 &= \wedge[(J_1 Q_2 \vee K_2 Q_1)(J_1 \vee J_2)]. \end{aligned} \quad (3.49)$$

Если все входы автомата — БС-сигналы, то необходим внешний управляющий сигнал  $C$ :

$$\begin{aligned} D_1 &= \wedge[(K_1 Q_1 \vee J_2 Q_2)C], \\ D_2 &= \wedge[(J_1 Q_2 \vee K_2 Q_1)C]. \end{aligned} \quad (3.50)$$

В этих реализациях фазовые входы автомата индицируются на выходах комбинационной части и далее — на И-сигнале триггера, поэтому никакой дополнительной индикации входов схем не требуется.

Резюмируя изложенное выше, общую структуру СС-автомата Мура можно представить в следующем виде (рис. 3.16).



**Рис. 3.16.** Структура СС-автомата Мура:

КЧ — комбинационная часть; С — У-сигнал; И — И-сигналы;  
ТТ — двухступенчатые D-триггеры (см. рис. 3.12)

Комбинационная часть в структуре не должна содержать явных преобразований приходящих БС-сигналов в ПФС-форму. Она выполняет заданные для автомата функции на основе смешанных СС-секций (см. раздел 3.3.5.4).

### 3.6. Самосинхронные конвейеры

Конвейеры в цифровых схемах применяются для повышения производительности устройств, когда за счет «пространственного» — по схеме — распределения информации осуществляется одновременная обработка разных ее порций.

В обычной синхронной схемотехнике конвейер — весьма сложное устройство, требующее тщательного согласования (синхронизации) как его элементов между собой, так и всего конвейера с другими устройствами. Поэтому синхронные конвейеры используются довольно редко, в наиболее ответственных частях схемы, сдерживающих производительность устройства целиком. К таким частям относятся, например, аппаратные умножители, устройства управления процессоров, согласующие буферы и др.

Напротив, органически свойственный СС-схемам способ запрото- ответного взаимодействия идеально подходит для организации конвейеров при низких накладных расходах. Это позволяет более широко

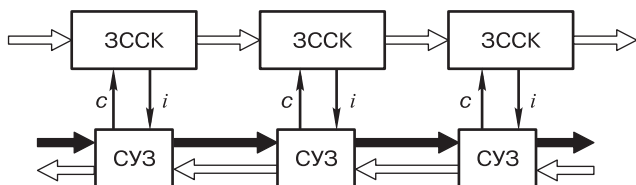
использовать СС-конвейеры в различных устройствах, где они обычно не применяются, и это перспективное направление дальнейшего развития СС-схемотехники.

СС-конвейер представляет собой последовательное соединение звеньев. Для правильного взаимодействия каждого звена с предшествующим и последующим необходимо организовать для них запрос-ответное взаимодействие (см. главу 2). Для этого каждое звено должно иметь определенные свойства, а для управления звеном необходима специальная схема.

Из процедуры запрос-ответа видно, что звеном самосинхронного конвейера должна быть СС-схема, имеющая, кроме информационных входов и выходов, один и только один индикаторный и один управляющий сигнал.

Схема управления звеном (СУЗ) будет одинаковой для всех звеньев одного конвейера, поскольку реализует одну и ту же дисциплину.

На рис 3.17 показана общая структура СС-конвейера. Из нее видно, что в конвейере есть два канала прохождения информации: по информационному каналу данные идут от начала конвейера к концу, по управляющему каналу служебная информация распространяется в обе стороны. Поэтому работа каждого звена конвейера в конечном итоге зависит от работы всех других звеньев.



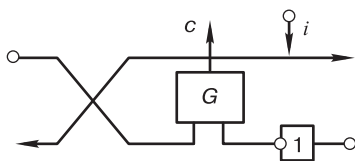
**Рис. 3.17.** Общая структура СС-конвейера:  
ЗССК — звено самосинхронного конвейера; СУЗ — схема управления звеном;  $i$  — И-сигналы;  $c$  — У-сигналы

Специфический признак, по которому можно разделить СС-конвейеры на два класса, — информационная плотность [2, 3]. Она зависит от способа организации фазовых переходов в соседних звеньях. В *неплотном конвейере* не допускается ситуация, когда два соседних звена находятся в рабочей фазе с разными порциями информации (задачами). Между такими звеньями должно быть хотя бы одно звено, находящееся в спейсере. В *плотном конвейере* такого ограничения нет. В результате рабочая фаза в плотном конвейере может установиться «плотно» — во всех звеньях, в неплотном — не плотнее, чем через одно звено.

Отметим, что плотность или неплотность конвейера не влияет на скорость прохождения через него каждой данной задачи, так как задача должна пройти через все звенья, и каждое звено при этом должно побывать и в спейсере, и в рабочей фазе.

Разница между неплотным и плотным конвейерами с одинаковым числом звеньев проявляется только тогда, когда предшествующая схема успевает подавать на вход конвейера очередную задачу сразу, как только наступает готовность к ее приему. В этом случае в плотном конвейере будет находиться больше разных задач в рабочей фазе, чем в неплотном, и его производительность будет соответственно выше.

Схема управления звеном **неплотного конвейера** приведена на рис. 3.18 [2].



**Рис. 3.18.** СУЗ неплотного СС-конвейера:  
G — Г-триггер

На рис. 3.19 показаны графики работы двух последовательных звеньев неплотного конвейера при условии, что соседние звенья обеспечивают немедленную выдачу и прием очередных порций данных.

На рис. 3.19 обозначено:

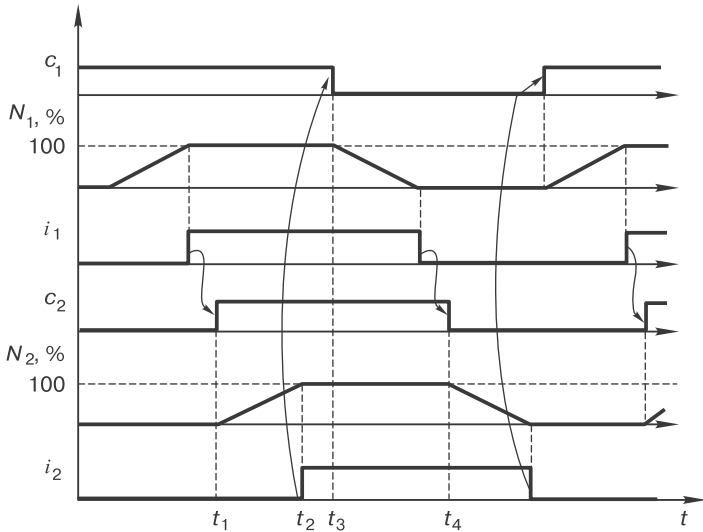
$t_1$  — начало перехода второго звена в рабочую фазу — обработка входных данных;

$t_2$  — завершение вторым звеном обработки входных данных;

$t_3$  — начало перехода первого звена в спейсер;

$t_4$  — начало перехода второго звена в спейсер.

Таким образом, в период  $t_1 - t_4$  У-сигнал  $c_2$  держит на входе второго звена состояние рабочей фазы. За это время первое звено с необходимостью переходит из рабочей фазы в спейсер. Если при этом выходные инфосигналы первого звена тоже будут меняться, то произойдет нарушение допустимости переходов для второго звена. Действительно, в одной и той же фазе при  $c_2 = 1$  в период  $t_1 - t_2$  сигналы второго звена монотонно переходили из спейсера в рабочую фазу, а затем, после момента  $t_3$ , будут переходить обратно в спейсер из-за изменения информационных входов. Поэтому можно сделать следующий важный вывод.



**Рис. 3.19.** Взаимодействие последовательных звеньев неплотного конвейера:

$N$  — число внутренних сигналов, находящихся в рабочей фазе;  
 $c$  — У-сигналы;  $i$  — И-сигналы; значения спейсеров — единичные

*Значения инфосигналов, связывающих два звена неплотного конвейера, должны сохраняться неизменными в рабочей фазе второго звена (определяемой его У-сигналом).*

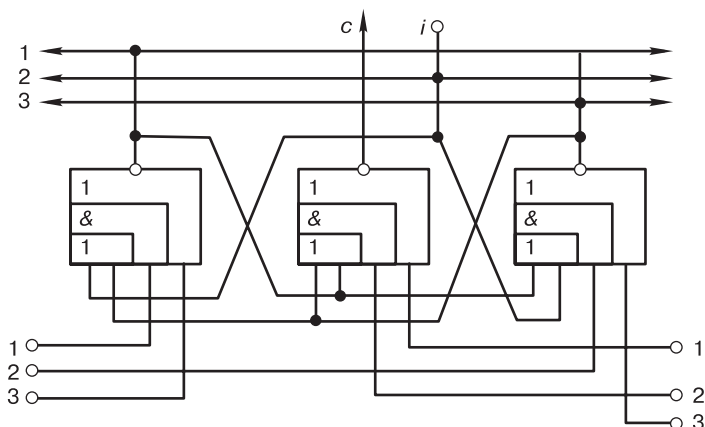
Данное требование не выполняется, например, для комбинационного предыдущего звена, когда связывающие инфосигналы — это сигналы типа ПФС.

Названное требование будет выполнено, если выходы первого звена являются выходами ССЗЯ. Для неплотного конвейера это могут быть одноступенчатые ячейки. Значения на их выходах будут сохраняться неизменными в спейсере первого звена (после момента  $t_3$ ), что гарантирует выполнение требования.

Рассмотрим особенности **плотных конвейеров**. СУЗ плотного конвейера [3] показана на рис. 3.20.

Управляющая схема плотного конвейера сложнее подобной схемы неплотного конвейера в 2,4 раза по числу транзисторов и в 3 раза по числу трасс, соединяющих эти схемы.

В плотном конвейере инфосигналы предыдущего звена должны оставаться неизменными в течение не только спейсера, но и рабочей фазы этого звена. Поэтому для запоминания таких выходных сигналов необходимо использовать только двухступенчатые ячейки.



**Рис. 3.20.** СУЗ плотного конвейера:  
 $c$  — У-сигнал;  $i$  — И-сигнал

Приведенные факты показывают, что плотные конвейеры сложнее неплотных, и их следует строить лишь в особых случаях, когда пиковая информационная емкость является решающей характеристикой. На настоящий момент видится только один подобный практический случай — буферный конвейерный регистр *FIFO* (first-in, first-out). В этом частном случае, однако, нет смысла применять управляющую схему общего вида, так как, используя специфику задачи, конвейер можно упростить. Пример подобной схемы есть в главе 4.

### 3.7. Проблема задержек трасс после разветвлений

Одним из условий корректного построения СС-схем на функционально-логическом этапе является соблюдение ограничений модели Маллера (см. раздел 2.3.2). Наиболее обсуждаемое из них — задержки реальных трасс после разветвлений. Если такие задержки различаются более чем на величину инерционных задержек последующих элементов, то ранее использованное логическое описание схемы теряет адекватность из-за нарушения условий модели Маллера. Свойство самосинхронности для таких реальных схем не может быть обеспечено. В то же время на практике подобные ситуации с задержками могут встречаться нередко — при передаче сигналов, например, по кабелям, через печатную плату между микросхемами, да и внутри микросхемы, если трассы достаточно длинны. Поэтому проблема задержек после разветвлений обсуждается с самого появления самосинхронных схем.

Простые соображения показывают, что решить проблему на логическом уровне, т. е. создать схему, нечувствительную ко всем задержкам после разветвлений, невозможно. Если требуется учесть задержку после разветвления, то ее надо каким-то образом измерить (оценить, сравнить), для чего приходится сделать ответвление на измеритель. Число разветвлений при этом увеличивается, и процесс не имеет конца.

Таким образом, для решения проблемы надо привлекать дополнительные данные, как минимум, из топологии и технологии.

Одна из попыток топологического решения принадлежит В. И. Варшавскому [4]. Им предложено «ветвление полем» — специальная конструкция транзистора, содержащая ветвление в нем самом, и специальная технология для ее реализации. Триггерная ячейка на таких транзисторах действительно не имеет разветвлений в трассах. Однако идея не получила развития по нескольким причинам. Усложнение транзисторов и технологии удорожает производство микросхем, но дело не только в этом. Проблема не решается до конца даже в принципе: разветвления существуют не только в триггерах, но и в других местах схемы, а для них ничего не предложено. И уж совсем непонятно, как быть с соединениями между микросхемами.

Другая идея, тоже топологическая, состоит в том, чтобы на этапе трассировки сделать трассы после разветвлений примерно одинаковой длины. Реализация идеи, однако, в настоящий момент представляется нереальной, так как процесс трассировки столь сложен и громоздок (это фактически переборная задача, а трасс — «тьма»), что дополнительные требования не позволят его завершить. Соединения между микросхемами — также вне рамок идеи.

Возникает вопрос, можно ли в принципе построить корректную СС-схему и с приемлемыми затратами решить проблему задержек после разветвлений.

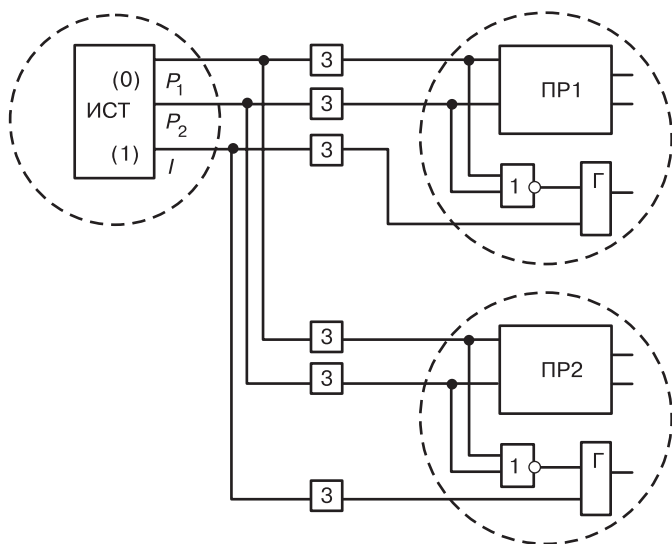
При исследовании этой проблемы обычно используется понятие эквихронной зоны.

*Эквихронная зона* — область на кристалле, внутри которой задержки трасс не превышают минимальных задержек элементов. Для любой СС-схемы, целиком лежащей в эквихронной зоне, условие задержек модели Маллера выполняется.

Предположим теперь, что источник и приемники сигналов находятся каждый в своей эквихронной зоне, а трассы связи разветвляются вне всех этих зон.

Проблему разветвлений можно решить, используя для связи между эквихронными зонами только фазовые сигналы. Такое реше-

ние представляется естественным, так как именно фазовые сигналы и предназначены для согласования работы устройств в СС-схемах. Бистабильные сигналы следует преобразовать в ПФС-сигналы с помощью простых схем (см. рис. 3.8) в той же эквихронной зоне, где эти бистабильные сигналы созданы. На всех приемниках, расположенных в других эквихронных зонах по отношению к источнику, пришедшие фазовые сигналы необходимо индицировать, и тогда самосинхронность будет обеспечена независимо от задержек после разветвлений (рис. 3.21). Межсоединения могут быть любыми — на кристалле, между кристаллами, блоками и стойками.



**Рис. 3.21.** К решению проблемы задержек после разветвлений:  
 ИСТ — источник сигналов; ПР1, ПР2 — приемники;  
 $P_1$ ,  $P_2$  — ПФС-сигналы;  $I$  — И-сигнал; 3 — задержки;  
 Г — Г-триггеры; пунктиром отмечены эквихронные зоны

Далее, для выяснения принципиальной возможности решения проблемы требуется оценить, уместятся ли минимальные законченные СС-схемы в эквихронные зоны. Величину эквихронных зон можно оценить по открытым публикациям, содержащим данные о конкретных технологических и топологических параметрах.

В экспериментальных данных и оценочных расчетах обычно указывают задержки на вентиль, под которым понимается инвертор (2 транзистора). В этих единицах и будем сравнивать величины схем и эквихронных зон.



Минимальные законченные СС-схемы представляют собой:

- ССЗЯ для схем с памятью (см. раздел 3.5.2);
- Г-триггеры (см. 3.2.2 и 4.2.1);
- СС-секции для комбинационных схем (см. 3.3.3).

Затраты любой ССЗЯ не превышают 24 вентиля. Г-триггер в любой реализации с числом входов до 4 содержит не более 10 вентиляей.

Найдем минимальный размер СС-секции. Исходные комбинационные уравнения всегда можно представить в базисе 2И-НЕ (2ИЛИ-НЕ), и тогда минимальная СС-секция, реализующая одну базисную функцию, будет содержать 8 вентиляей.

С учетом дополнительных затрат на преобразование сигналов БС — ПФС и индикации согласно рис. 3.21, можно принять оценку величины минимальной СС-схемы в пределах одной эквихронной зоны в 40 вентиляей (с запасом). Из таких схем в принципе можно создать любую СС-схему, нечувствительную к задержкам после разветвлений.

Величину эквихронной зоны оценим следующим образом. Трассы на кристалле идут под прямыми углами. Пусть эквихронная зона имеет форму квадрата, и сигнал проходит максимальный путь из одного угла в другой по двум сторонам. Время прохождения этого пути должно быть не больше задержки одного вентиля. Количество вентиляей внутри квадрата и будет величиной эквихронной зоны.

Из публикации [30] можно видеть, что если ширина медной трассы оптимальна, то задержка на ней составляет 20—22 пс/мм для всех технологий с размерами 90—22 нм. Некоторые другие источники дают задержки для чипов 90 нм равными 5—10 пс/мм, но примем худший случай — 22 пс/мм.

Задержка на вентиль, измеренная экспериментально, по разным источникам, составляет 5—10 пс/вент для технологических норм 90 нм. Для худшего расчетного случая примем ее значение 5 пс. При уменьшении норм задержка уменьшается не быстрее, чем линейно. Поэтому для норм 45 нм примем задержку в 2,5, а для 32 нм — 1,8 пс/вент.

Наконец, число вентиляей на единице площади кристалла зависит от типа технологии [31]. Для технологии 90 нм это около 1,4 млн, для 45 нм — 1,7 млн, для 32 нм — 2,4 млн вент/мм<sup>2</sup>.

В результате расчетов получаем величины эквихронных зон:

- для 90 нм — 18 тыс. вентиляей;
- для 45 нм — 5,5 тыс. вентиляей;
- для 32 нм — 4 тыс. вентиляей.

Таким образом, при существующих технологиях величины эквихронных зон на порядки превышают минимально необходимые размеры СС-схем.

В результате этого оценочного анализа можно утверждать, что проблема задержек трасс после разветвлений носит не принципиальный, а технический характер — это проблема разработки схем с учетом их размещения на кристалле.

Отметим также, что в сходных условиях проблема синхронизации в обычных синхронных схемах не менее, а может быть и более, сложна.

Вообще, задержки в трассах представляют собой общую проблему микроэлектроники, и по мере развития технологий удельные задержки уменьшаются. В перспективных разработках, например [30], рассматриваются оптические межсоединения, с появлением которых эквихронные зоны еще увеличатся. В этом смысле СС-схемы имеют хорошие перспективы.

# Глава 4

## ПРИМЕРЫ САМОСИНХРОННЫХ СХЕМ

---

Множество примеров схем, претендующих на свойство самосинхронности, содержится в патентах (включая сюда и авторские свидетельства СССР). Анализ патентных решений выходит за рамки книги, однако следует сделать несколько замечаний.

Зарубежные схемы, как правило, не являются самосинхронными и относятся к квазисамосинхронным решениям. Немногие доступные схемы, созданные по NCL-методологии, специфичны по составу элементов, избыточны по числу транзисторов и здесь рассматриваться не будут.

Отечественные патенты представлены двумя группами авторов — В. И. Варшавским с сотрудниками и специалистами ИПИРАН.

На счету группы Варшавского около 180 российских патентов. Далеко не все они обладают достаточными качествами для практического применения. С точки зрения логики, они в основном удовлетворяют теории самосинхронности (хотя есть ошибки, выявленные, в том числе, и самими авторами). Тем не менее перед использованием какой-либо схемы ее для уверенности следует проанализировать с учетом полноты (см. раздел 5.5), так как проблемой полноты данная группа не занималась.

Главный недостаток большинства патентов группы Варшавского — неэффективность реализации. Многие схемы относятся к пМДП-технологии, т. е. с токопотреблением в статике, что почти не востребовано на практике. Некоторые схемы допускают сквозные токи, т. е. неработоспособны. Наконец, много схем, элементы которых имеют индексы задержки (число последовательных транзисторов между нагрузкой и питанием/землей) более четырех, что делает их неэффективными по задержкам в КМДП-технологии. В ряде схем наблюдается чрезмерно большая нагрузка по некоторым вхо-

дам и на выходах элементов. В результате лишь небольшая часть патентов группы может быть подходящей для практического использования.

Специалисты ИПИРАН на данный момент имеют 11 патентов и несколько заявок на рассмотрении. Самосинхронность с учетом полноты и эффективность реализации были одними из главных условий при разработке схемных решений. Поэтому схемы этих патентов могут быть с уверенностью рекомендованы для применения.

Большое разнообразие СС-триггеров, а также некоторых комбинационных СС-элементов можно найти в публикации ИПИРАН [32].

Ниже в данной главе приводятся относительно простые самосинхронные схемы (в том числе из опубликованных материалов — книг, патентов и др.), которые могут служить как образцами для построения подобных устройств, так и базовыми конструкциями для создания более сложных схем.

Устройства выполнены в парафазной кодировке и с использованием двухфазной дисциплины (рабочая фаза и спейсер).

Все схемы включают необходимые индикаторы входных сигналов в предположении, что соответствующие сигналы не индицируются во внешних схемах. Если некоторые входы будут индицироваться вовне, как нередко бывает на практике, то их индикаторы можно из схем исключить.

Во всех схемах выбрана следующая кодировка значений: 0 кодируется как 01 в ПФС-коде, 1 — как 10. Спейсер кодируется одинаковыми значениями ПФС-сигналов, 0 или 1, в зависимости от места сигналов в схеме. Значения спейсеров указаны на рисунках в скобках.

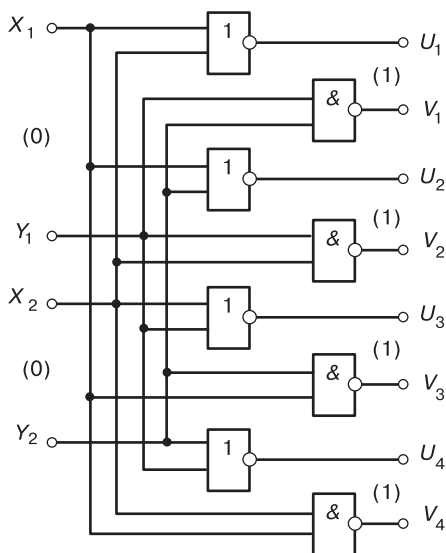
## 4.1. Комбинационные схемы

Все приведенные примеры комбинационных схем построены с использованием табличного метода синтеза (см. раздел 3.3.5) с оптимизацией.

### 4.1.1. Дешифратор

СС-дешифраторы относятся к тем редким СС-схемам, в которых все ПФС-входы индицируются на информационных выходах, и отдельной индикации входов не требуется.

Пример дешифратора на 2 входа и 4 выхода (информационных) приведен на рис. 4.1.



**Рис. 4.1.** Дешифратор 2—4:

$X_i, Y_i$  — ПФС-входы;  $U_j, V_j$  — ПФС-выходы

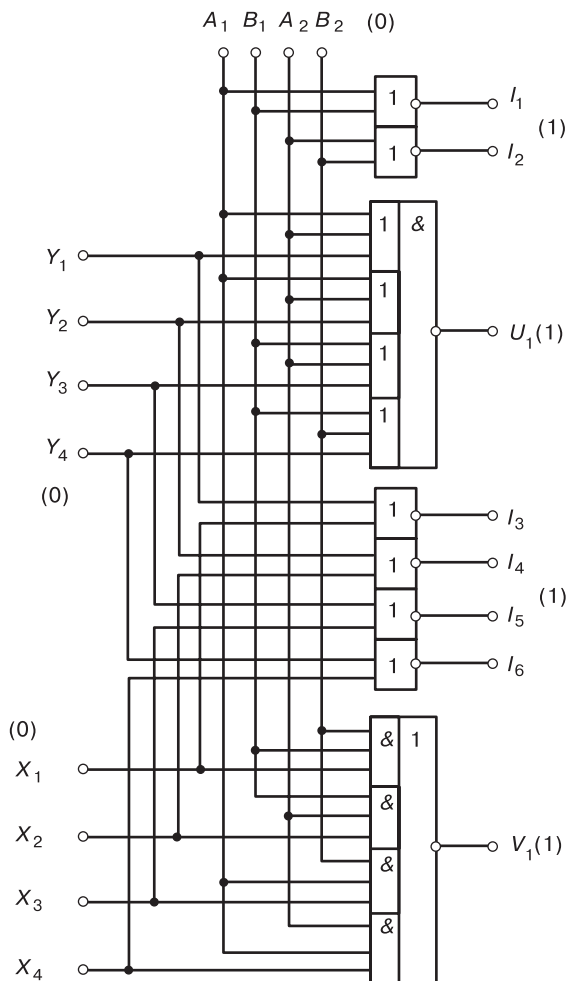
Схема работает следующим образом. В исходном состоянии — спейсере — входные и выходные ПФС-сигналы попарно равны. После установления на входе рабочих наборов (попарно неравных) выходы также установятся в рабочую фазу. При этом на ПФС-выходе, номер которого соответствует входному двоичному коду, установится 1, на остальных выходах установится 0.

#### 4.1.2. Мультиплексор

В отличие от дешифраторов, в СС-мультиплексорах ни один из входных ПФС-сигналов не индицируется на информационных выходах. Поэтому все ПФС-входы должны индицироваться отдельно.

Пример мультиплексора на 4 информационных входа, 2 адресных входа и 1 информационный выход показан на рис. 4.2.

При наступлении рабочего состояния на ПФС-выходе мультиплексора появится значение, повторяющее значение информационного ПФС-входа, номер которого закодирован адресными ПФС-входами.



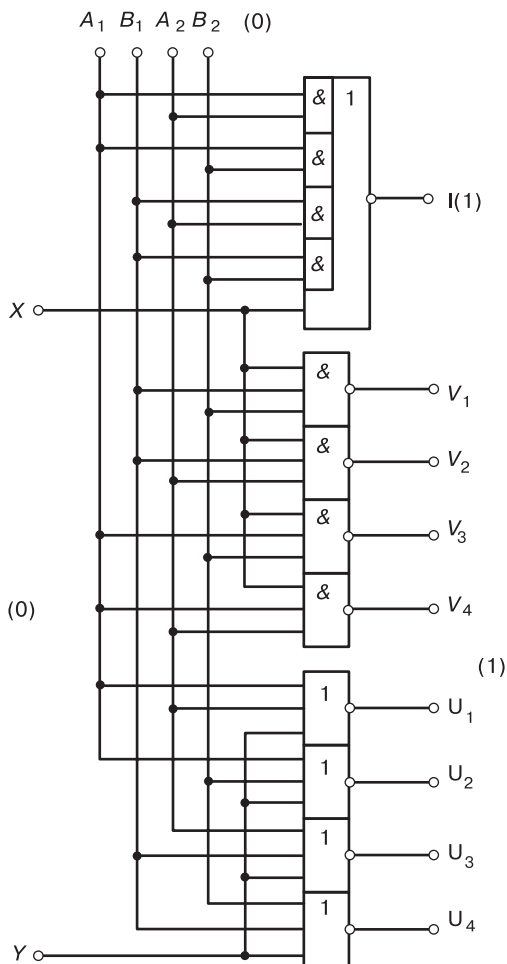
**Рис. 4.2.** Мультиплексор 4—1:

$A_i, B_i$  — адресные ПФС-входы;  $X_j, Y_j$  — информационные ПФС-входы;  
 $U, V$  — информационный выход;  $I_k$  — И-сигналы

### 4.1.3. Демультимплексор

Демультимплексор на 1 информационный вход, 2 адресных входа и 4 информационных выхода показан на рис. 4.3.

При наступлении рабочей фазы значение информационного ПФС-входа установится на том ПФС-выходе, номер которого закодирован значениями адресных ПФС-входов.

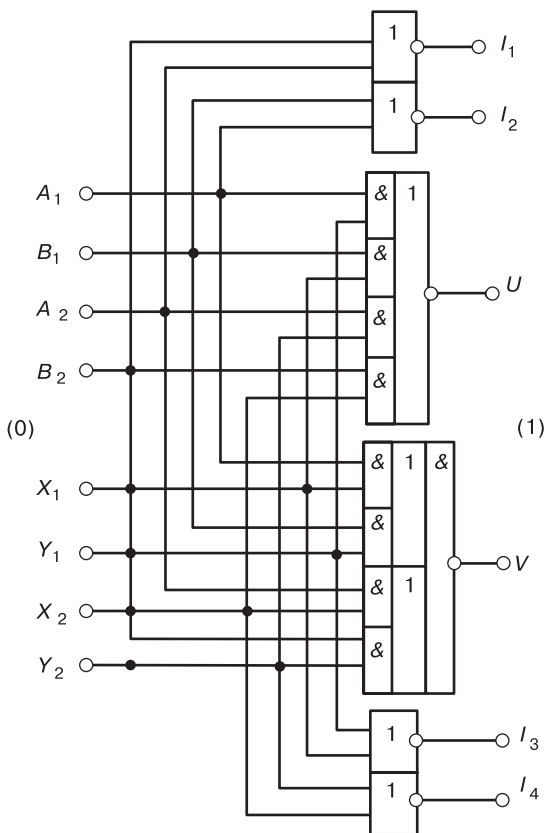


**Рис. 4.3.** Демультимплексор 1—4:  
 $X, Y$  — информационный ПФС-вход;  $A_i, B_i$  — адресные ПФС-входы;  $U_i, V_i$  — ПФС-выходы;  $I$  — И-сигнал

#### 4.1.4. Схема равнозначности

В схемах равнозначности все ПФС-входы также приходится индексировать отдельно.

На рис. 4.4 приведена схема равнозначности двух 2-битовых кодов.



**Рис. 4.4.** Схема равнозначности двух 2-битовых кодов:  
 $A_i, B_i$  — первый код;  $X_i, Y_i$  — второй код;  
 $U, V$  — выход равенства кодов

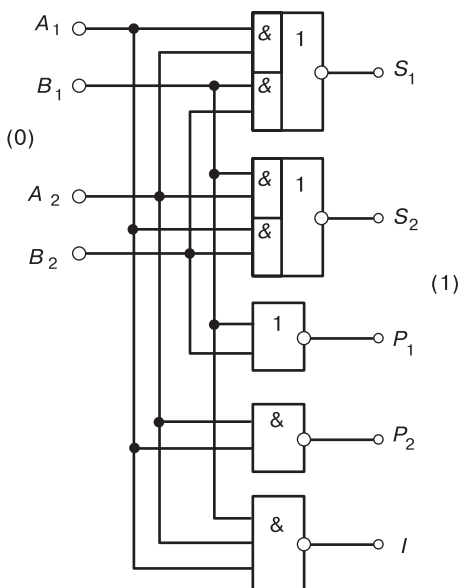
#### 4.1.5. Полусумматор

Полусумматор показан на рис. 4.5.

#### 4.1.6. Сумматор одноразрядный

Одноразрядный сумматор можно построить либо на основе двух полусумматоров, либо специально. В последнем случае схема несколько выигрывает как по быстродействию, так и по числу транзисторов. Именно в этом варианте схема показана на рис. 4.6.





**Рис. 4.5.** Полусумматор:

$A_1, A_2, B_1, B_2$  — операнды;  $S_1, S_2$  — выход суммы;  $P_1, P_2$  — выход переноса

## 4.2. Схемы с памятью

### 4.2.1. Индикаторные триггеры

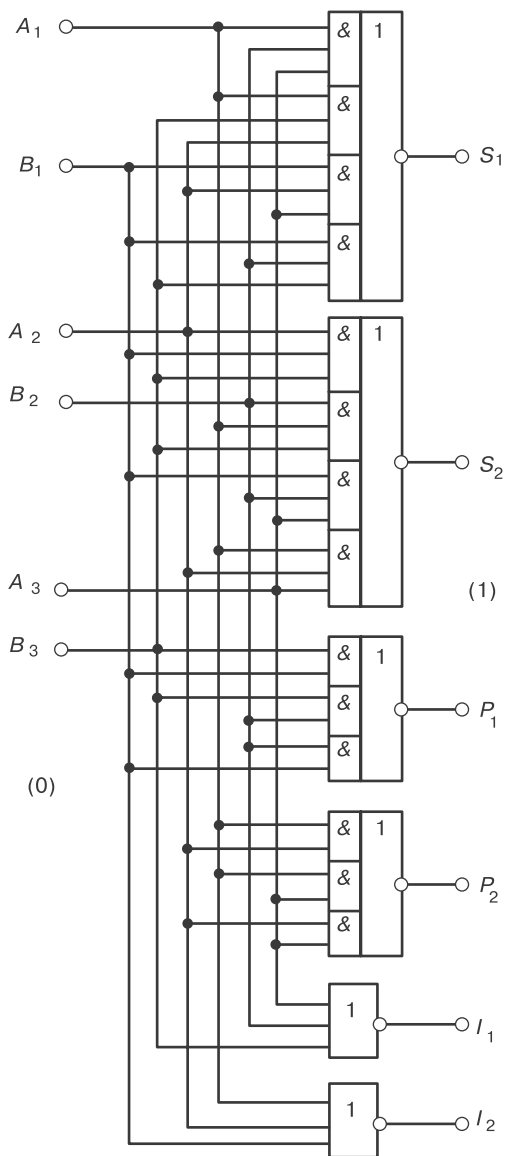
Индикаторные триггеры предназначены для редукции — понижения числа индикаторных сигналов в схеме и /или для индикации группы ПФС-сигналов, также с редукцией.

Индикаторный триггер (редуктор) с тремя унарными входами показан на рис. 3.4, а. По такому же принципу строятся триггеры на 2 и 4 унарных входа. Каждый из этих редукторов описывается уравнением вида (3.3). Редукторы с унарными входами обладают симметрией: их можно использовать как при входном спейсере 0, так и при спейсере, равном 1.

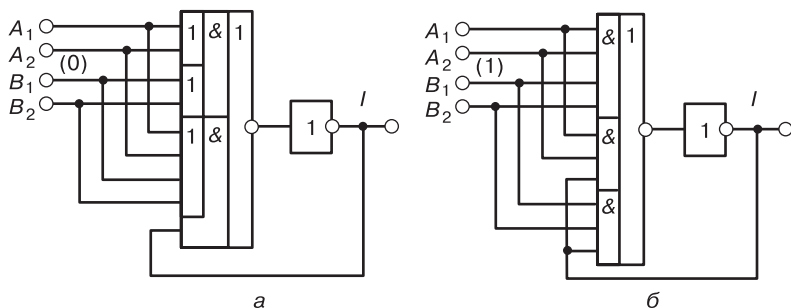
Индикацию двух ПФС-сигналов с одновременным понижением индикаторных сигналов до одного можно реализовать на индикаторных триггерах, приведенных на рис. 4.7. Симметрия в данном случае с очевидностью отсутствует.

Уравнения триггеров на рис. 4.7 имеют соответственно вид:

$$\begin{aligned}
 I &= (A_1 \vee A_2)(B_1 \vee B_2) \vee (A_1 \vee A_2 \vee B_1 \vee B_2)I^-, \\
 I &= A_1A_2B_1B_2 \vee (A_1A_2 \vee B_1B_2)I^-.
 \end{aligned}
 \tag{4.1}$$

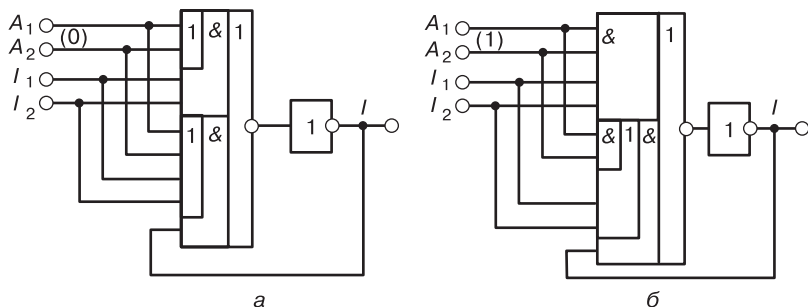


**Рис. 4.6.** Сумматор одноразрядный:  
 $A_i, B_i$  — три слагаемых;  $S_1, S_2$  — выход суммы;  
 $P_1, P_2$  — выход переноса



**Рис. 4.7.** Индикаторные триггеры для двух ПФС-сигналов: *а* — с входным спейсером 0; *б* — с входным спейсером 1

Существуют и комбинированные индикаторные триггеры — для ПФС- и И-сигналов. Примеры представлены на рис. 4.8.



**Рис. 4.8.** Комбинированные индикаторные триггеры для одной пары ПФС-сигналов и двух И-сигналов: *а* — с входным спейсером 0; *б* — с входным спейсером 1

Эти триггеры описываются соответственно уравнениями:

$$\begin{aligned}
 I &= (A_1 \vee A_2)I_1I_2 \vee (A_1 \vee A_2 \vee I_1 \vee I_2)I^-, \\
 I &= A_1A_2I_1I_2 \vee (A_1A_2 \vee I_1 \vee I_2)I^-.
 \end{aligned}
 \tag{4.2}$$

Индикаторные триггеры можно построить и на запоминающих ССБЭ (см. главу 2).

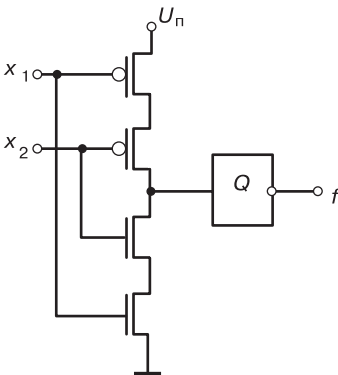
Схемы на запоминающих ССБЭ с унарными входами, выполняющие функцию Г-триггера вида (3.3), в зарубежной литературе получили название *С*-элементов Маллера (Маллер использовал уравнения таких устройств для иллюстрации своей теории). На практике, по соображениям ограничения максимального индекса задержки, применяются *С*-элементы с числом входов не более четырех.

На рис. 4.9 показан редуктор ( $C$ -элемент Маллера) с двумя входами (схема  $Q$ -элемента приведена в разделе 2.4.2.2).

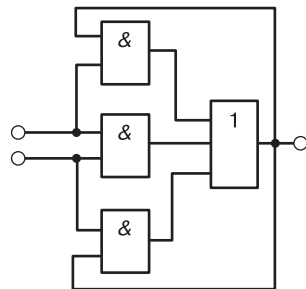
Поведение этого элемента описывается уравнением

$$f = x_1 x_2 \vee f^-(x_1 \vee x_2). \quad (4.3)$$

Увы, поверхностное знакомство с теорией приводит к ошибкам в публикациях. В статье [33] наряду с правильной схемой (рис. 4.9) показана схема под названием «вентильная реализация  $C$ -элемента» (рис. 4.10). Та же схема опубликована и в Wikipedia [34]. Здесь допущено сразу несколько ошибок. Во-первых, под  $C$ -элементами всегда имелись и имеются в виду схемы со слабым инвертором. Во-вторых, нарушены условия модели Маллера, подразумевающей триаду: один элемент — одна задержка — одно уравнение. Схема на рис. 4.10 имеет четыре таких триады, и ее поведение описывается не одним уравнением, а четырьмя. Аналогичная схема на элементах И-НЕ приведена в [3, рис. 4.25] именно как некорректная реализация двухвходового  $\Gamma$ -триггера. Там же показано нарушение самосинхронности схемы (аналогичные нарушения присущи и схеме на рис. 4.10). Таким образом, схема на рис. 4.10 является обычным асинхронным триггером с характеристическим (обобщенным) уравнением (4.3) и, кроме уравнения, не имеющим никакого отношения ни к  $C$ -элементу, ни к теории Маллера.

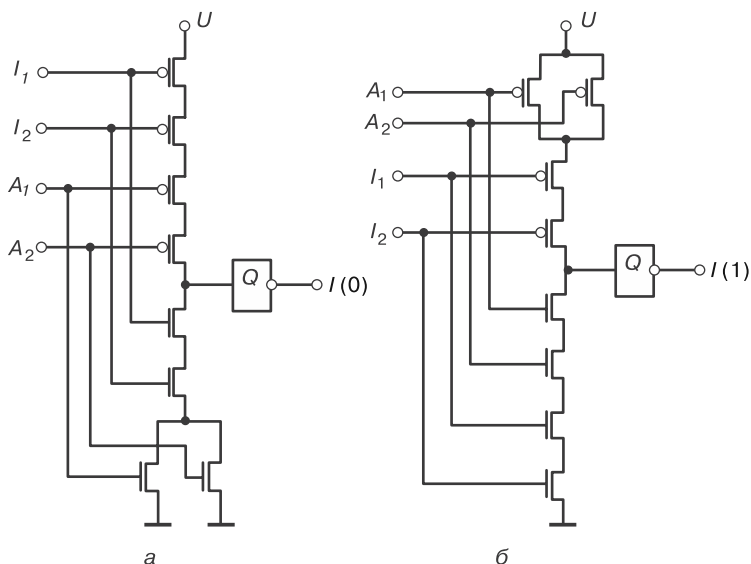


**Рис. 4.9.**  $C$ -элемент Маллера с двумя входами



**Рис. 4.10.** Некорректная «реализация»  $C$ -элемента Маллера

На запоминающих ССБЭ можно реализовать и комбинированные индикаторные триггеры. Примеры приведены на рис. 4.11.



**Рис. 4.11.** Индикаторные триггеры на запоминающих ССБЭ для одного ПФС-сигнала и двух И-сигналов:  
 а — с входным спейсером 0; б — с входным спейсером 1

## 4.2.2. Информационные триггеры

Информационные триггеры предназначены для запоминания информации. Запись информации в триггер происходит в одной из фаз, как правило, рабочей (так и будем считать далее). В одноступенчатых триггерах информация сохраняется только в последующей фазе спейсера, в двухступенчатых — и в спейсере, и в следующей рабочей фазе.

Далее будут рассмотрены примеры  $D$ - и  $T$ -триггеров.

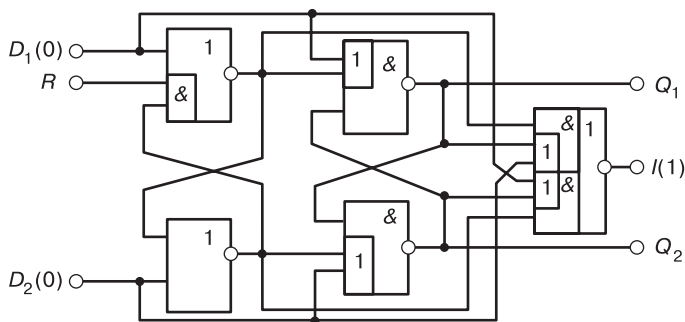
### 4.2.2.1. $D$ -триггеры

Основные схемы  $D$ -триггеров показаны на рис. 3.10—3.13. Ниже приводится несколько их модификаций для типовых случаев практического применения. Некоторые модификации традиционно называются в литературе  $RS$ -триггерами, что для СС-схем не совсем корректно (см. раздел 3.4.1).

#### А. Двухступенчатый $D$ -триггер с асинхронной предустановкой (сбросом).

Схема триггера показана на рис. 4.12. Сброс осуществляется сигналом  $R = 0$  при спейсерных значениях входов  $D_1 = D_2 = 0$ . Асинхронность предустановки состоит в том, что установочный сигнал подает-

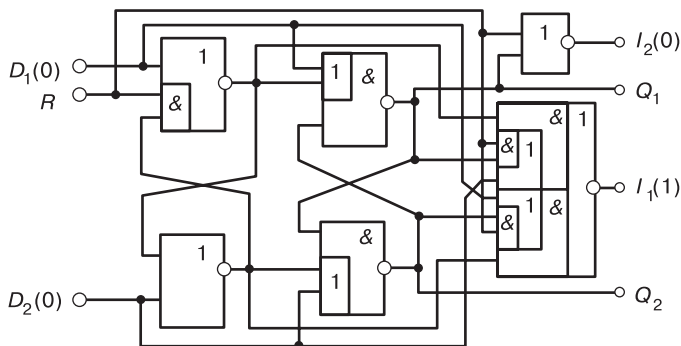
ся на некоторый период, предположительно больший длительности переходных процессов этой предустановки, после чего возвращается в единичное значение.



**Рис. 4.12.** Двухступенчатый  $D$ -триггер с асинхронной предустановкой (сбросом):  $D_1, D_2$  — ПФС-вход;  $R$  — вход сброса;  $Q_1, Q_2$  — БС-выход;  $I$  — И-сигнал

### Б. Двухступенчатый $D$ -триггер с самосинхронной предустановкой (сбросом).

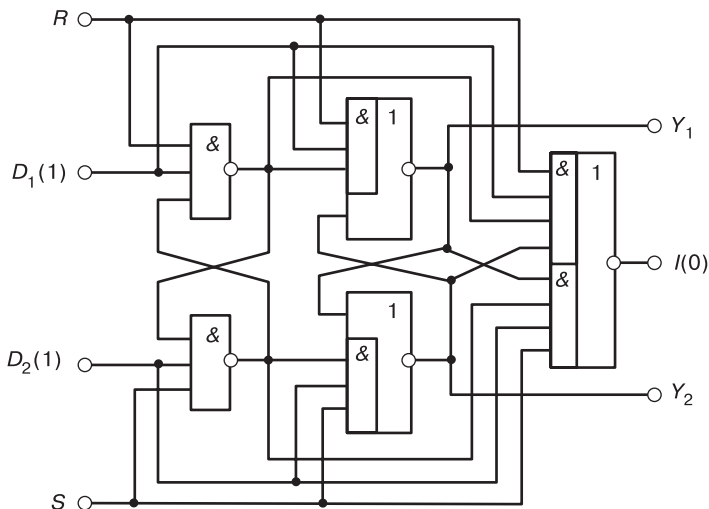
Схема этого триггера приведена на рис. 4.13. Предустановка производится также в спейсерной фазе. В отличие от предыдущей схемы, завершение предустановки индицируется сигналом  $I_2 = 1$ , после чего необходимо снять установочный сигнал и дождаться индикации спейсера  $I_2 = 0$ . Такой порядок предустановки повышает общую надежность схем с использованием триггеров этого типа.



**Рис. 4.13.** Двухступенчатый  $D$ -триггер с самосинхронной предустановкой (сбросом):  $D_1, D_2$  — ПФС-вход;  $R$  — вход сброса;  $Q_1, Q_2$  — БС-выход;  $I_1$  — И-сигнал триггера;  $I_2$  — И-сигнал сброса

### В. Двухступенчатый $D$ -триггер с самосинхронными установкой и сбросом.

Схема триггера показана на рис. 4.14. Предустановка производится в спейсерной фазе  $D_1 = D_2 = 1$ . Для сброса надо подать сигнал  $R = 0$ , для установки —  $S = 0$  (подавать оба нулевых сигнала запрещено). Завершение любой предустановки индицируется общим индикаторным сигналом  $I = 1$ , после чего необходимо снять установочный сигнал и дождаться индикации спейсера  $I = 0$ .



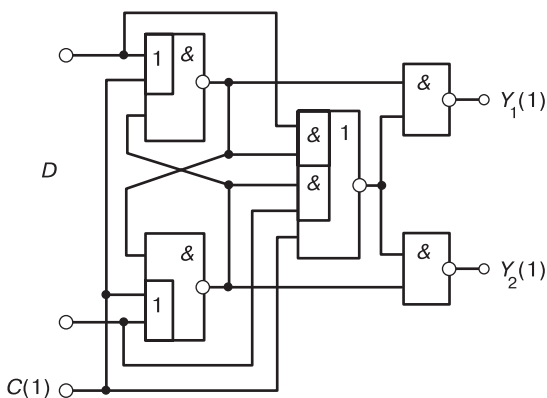
**Рис. 4.14.** Двухступенчатый  $D$ -триггер с самосинхронными установкой и сбросом:  $D_1, D_2$  — ПФС-вход;  $S$  — вход установки;  $R$  — вход сброса;  $Q_1, Q_2$  — БС-выход;  $I$  — общий И-сигнал

### Г. Одноступенчатый $D$ -триггер с преобразованием выхода в ПФС-сигнал.

Особенность этого триггера в том, что выход схемы (рис. 4.15) является ПФС-сигналом, поскольку использование БС-сигналов в ряде случаев ограничено — их нельзя усилить или задержать. Задержка может возникнуть, например, если попытаться послать этот сигнал на длинную линию или усилить. Приведенная схема как раз и призвана снять подобные ограничения [35].

### Д. Одноступенчатый $D$ -триггер с усиленными выходами.

Во многих случаях полезно иметь триггеры с усиленными выходами. В традиционных синхронных и асинхронных схемах вопрос решается просто: на выходы ставятся усиливающие инверторы. Для

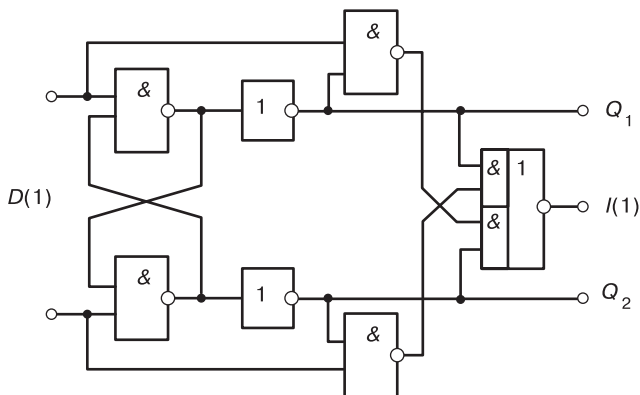


**Рис. 4.15.** Одноступенчатый  $D$ -триггер с преобразованием выхода в ПФС-сигнал:  
 $D$  — БС-вход;  $C$  —  $Y$ -сигнал;  $Y_1, Y_2$  — ПФС-выход

самосинхронных схем такое простое решение недопустимо (см. раздел 3.4.3).

Для соблюдения самосинхронности необходимо принимать специальные меры, одна из которых — преобразовать БС-выход в ПФС-сигнал (см. рис. 4.15) и затем усилить ПФС-выходы.

Другое решение — специальная схема усиленного триггера (без преобразования выходов) [36] — показано на рис. 4.16. Поскольку усиленные выходы остаются БС-сигналами, для их индикации использовался принцип сравнения со входами (см. раздел 2.1.1).



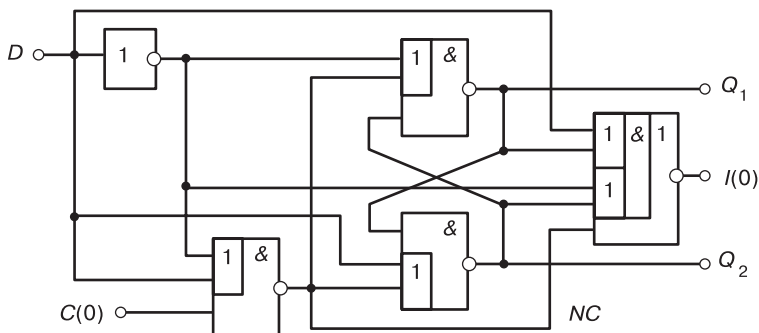
**Рис. 4.16.** Одноступенчатый  $D$ -триггер с ПФС-входом и усиленными выходами:  
 $D$  — ПФС-вход;  $Q_1, Q_2$  — БС-выход;  $I$  — И-сигнал



Аналогичным способом строятся и двухступенчатые  $D$ -триггеры с усиленными выходами.

### Е. Одноступенчатый $D$ -триггер с унарным информационным входом.

Пример триггера с унарным входом [37] приведен на рис. 4.17.



**Рис. 4.17.** Одноступенчатый  $D$ -триггер с унарным входом:  
 $D$  — УС-вход;  $C$  — У-сигнал;  $Q_1, Q_2$  — БС-выход;  
 $I$  — И-сигнал;  $NC$  — вспомогательный сигнал

Такой триггер может использоваться для корректного преобразования унарного сигнала в БС-сигнал. Унарный вход в общем случае должен подчиняться той же дисциплине, что и БС-сигналы (см. раздел 3.4.3), т. е. не должен меняться при  $C = 1$ . Однако можно улучшить быстродействие преобразования, применив способ, указанный в разделе 4.3.

#### 4.2.2.2. Счетные триггеры ( $T$ -триггеры)

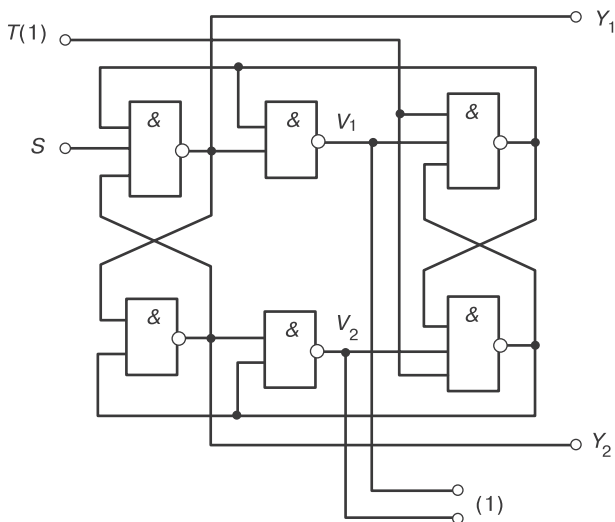
Счетные триггеры могут реализовываться на ячейках с гашением и без гашения. Как уже указывалось (см. раздел 3.4.1), при одинаковых затратах триггер с гашением характеризуется худшим быстродействием, чем триггер без гашения. Тем не менее триггеры с гашением широко использовались в схемах группы Варшавского.

Особенность триггеров данного типа в том, что их основной вход — фазовый управляющий сигнал, а выходы переключаются в противоположные значения в каждом следующем цикле спейсер — рабочая фаза.

#### А. $T$ -триггер с гашением и асинхронной установкой.

$T$ -триггер с гашением [3, рис. 4.12, б] и асинхронной установкой показан на рис. 4.18. Он работает таким образом. При  $T = 0$  последняя БСЯ погашена:  $U_1 = U_2 = 1$ . При переходе в  $T = 1$  информация от сиг-

налов  $V_1$  и  $V_2$  записывается в эту последнюю БСЯ, что по обратной связи приводит к изменению выходов схемы  $Y_1$  и  $Y_2$  на противоположные. При этом промежуточные парафазные сигналы становятся равными:  $V_1 = V_2 = 1$ . После обратного перехода  $T=0$  последняя БСЯ гасится, сигналы обратной связи разрешают сигналам левой БСЯ установиться в значения  $V_1 \neq V_2$  — рабочее состояние. Далее процесс повторяется.



**Рис. 4.18.** Счетный триггер с гашением:  
 $T$  — счетный вход;  $S$  — сигнал установки;  
 $Y_1, Y_2$  — БС-выход;  $V_1, V_2$  — ПФС-выход (вариант)

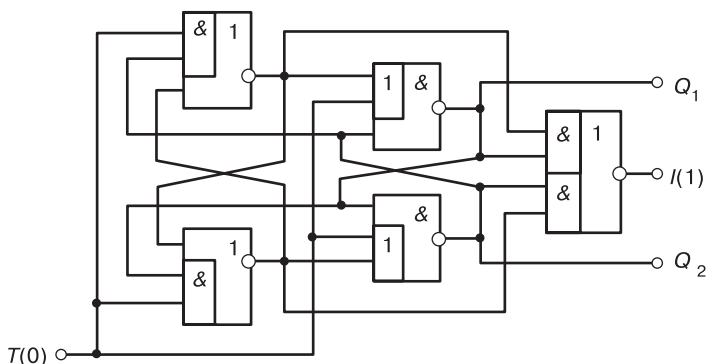
Установка триггера производится сигналом  $S = 0$  при  $T = 0$ . Окончание процесса установки в этой схеме не контролируется, что и отражено в названии «асинхронная установка» (в отличие от самосинхронной). Сигнал  $S = 0$  необходимо удерживать на время, «достаточное для завершения установки».

Некоторым схемотехническим преимуществом данного триггера по сравнению с триггером без гашения может служить универсальность: есть и БС-выход, и ПФС-выход с единичным спейсером.

### **Б. $T$ -триггер без гашения.**

Счетный триггер на основе двухступенчатого  $D$ -триггера [3] (см. рис. 3.12) показан на рис. 4.19. Он построен по тому же принципу, что и аналогичный триггер в обычной схематехнике, — с помощью обратных связей с выходов на входы. По этой причине его индикатор

проще исходного индикатора, так как входы и выходы совмещены. Каждый цикл изменения входа  $T$  за счет перекрестных обратных связей приводит к изменению значений выходов  $Y_1$  и  $Y_2$  на противоположные. В этой схеме входной спейсер может быть любым — 0 или 1, а выходной противоположен входному.



**Рис. 4.19.** Счетный триггер без гашения на основе двухступенчатого  $D$ -триггера:  $T$  — счетный вход;  $Q_1, Q_2$  — БС-выход;  $I$  — И-сигнал

Асинхронную или самосинхронную предустановку этого триггера можно сделать по аналогии с триггерами на рис. 4.12—4.14.

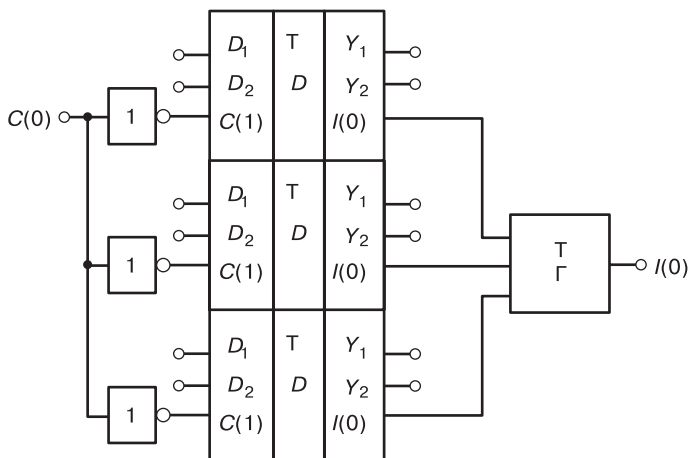
### 4.2.3. Параллельные регистры

Параллельные регистры применяются для временного хранения многоразрядных сигналов. Единственная особенность параллельных СС-регистров состоит в необходимости редукции индикаторных сигналов (см. раздел 3.2.2). При большом числе разрядов может потребоваться каскадная схема редукции.

На рис. 4.20 показан трехразрядный параллельный одноступенчатый регистр на элементах рис. 3.11.

### 4.2.4. Последовательные регистры

В данном разделе приводятся примеры регистров сдвига и последовательного конвейерного регистра. Разница между ними состоит в разном взаимодействии соседних разрядов. В обычном сдвиговом регистре информация в разрядах сдвигается при переходе всей схемы из одной фазы в другую. В конвейерном регистре реализуется запрос-ответное взаимодействие, и соседние разряды могут находиться в разных фазах.



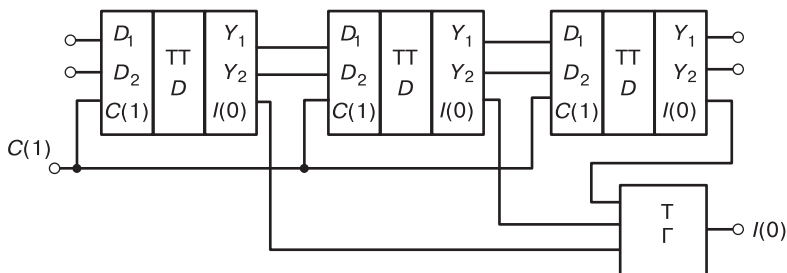
**Рис. 4.20.** Параллельный одноступенчатый регистр:  
Г — Г-триггер, С — У-сигнал, I — И-сигнал

Отметим, что все изобретения группы Варшавского, имеющие в названии слова «регистр сдвига», являются конвейерными схемами, что не соответствует традиционному пониманию регистра сдвига.

#### 4.2.4.1. Регистры сдвига

СС-регистры сдвига строятся на двухступенчатых триггерах. В одной из фаз, обычно рабочей, информация переписывается из одного триггера в другой, а в другой фазе она переписывается внутри триггеров из первой ступени во вторую. Таким образом, за каждый цикл рабочая фаза — спейсер информация сдвигается на один разряд.

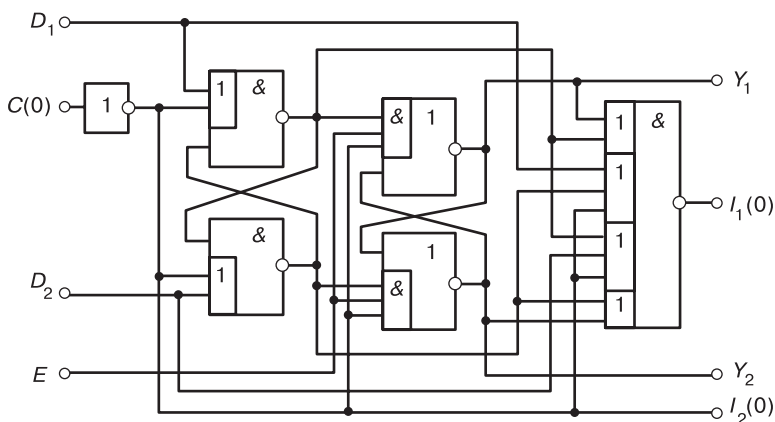
Простейший регистр сдвига можно построить на элементах рис. 3.13 с общим управляющим сигналом, как показано на рис. 4.21.



**Рис. 4.21.** Регистр сдвига с общим У-сигналом

Регистры данного типа характеризуются большой нагрузкой на вход  $C$ . Но ставить усиливающие инверторы (или повторители) на управляющий вход каждого триггера нельзя из-за возникновения гонок в этом случае (см. главу 5). Поэтому предельное число разрядов при практической реализации таких схем будет невелико.

Одним из решений проблемы усиления управляющих сигналов и создания практичных регистров любой длины может быть использование специальной ячейки с дополнительной блокировкой [38] (рис. 4.22).



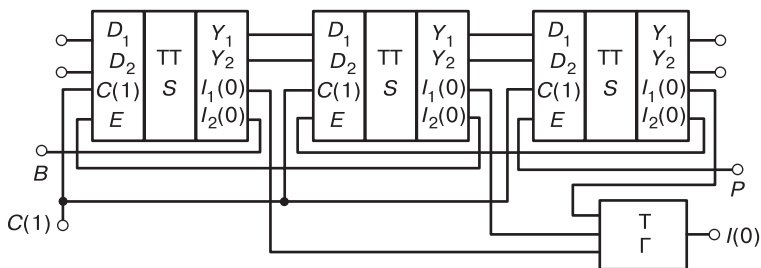
**Рис. 4.22.** Разряд самосинхронного регистра сдвига:  
 $D_1, D_2, Y_1, Y_2$  — входные и выходные БС-сигналы;  $C$  — У-сигнал;  
 $I_1, I_2$  — И-сигналы;  $E$  — вход дополнительной блокировки

В этой схеме управляющий сигнал  $C$  уже снабжен усилительным инвертором, но для обеспечения самосинхронности требуется дополнительная блокировка (подробно о необходимых блокировках рассказано в разделе 5.7).

Сигнал  $I_2$  индицируется на И-сигнале  $I_1$ , и во внешней схеме его индикации не требуется. Сигнал  $E$ , напротив, в схеме не индицируется (частичная индицируемость, см. раздел 2.5.3) и должен индицироваться вне этой схемы.

Регистр сдвига на этой ячейке строится следующим образом (рис. 4.23).

Обратные связи в этой схеме предотвращают возникновение гонок. Сигнал  $E$  каждого разряда подключен к выходу  $I_2$  последующей ячейки и потому индицируется автоматически (сигнал  $P$ , однако, должен индицироваться во внешней схеме).

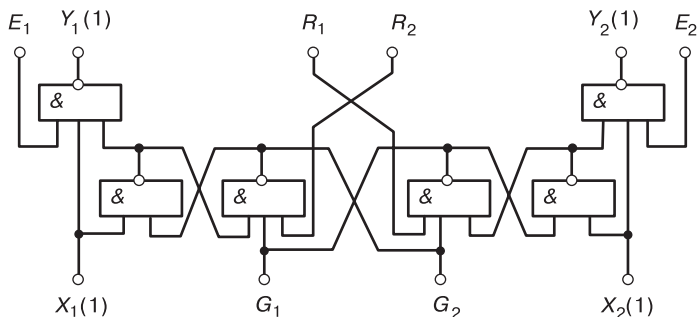


**Рис. 4.23.** Регистр сдвига на специальных ячейках:  
 $B$  — сигнал блокировки для предыдущей БСЯ (используется при необходимости);  $C$  —  $Y$ -сигнал;  $P$  — блокирующий сигнал от последующей БСЯ;  $I$  — И-сигнал

#### 4.2.4.2. Буферный конвейерный регистр FIFO

Регистры типа FIFO (first-in, first-out) применяются для буферного (временного) хранения информации при согласовании взаимодействия источников и приемников данных.

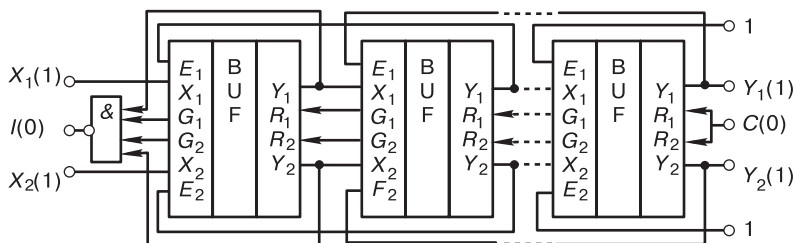
Одной из наиболее эффективных схем для одного разряда такого регистра является изобретение [39] (рис. 4.24).



**Рис. 4.24.** Одноразрядная ячейка буферного конвейерного регистра FIFO:  
 $X_1, X_2$  — ПФС-входы;  $Y_1, Y_2$  — ПФС-выходы;  $R_1, R_2$  — ответные входы;  
 $G_1, G_2$  — ответные выходы;  $E_1, E_2$  — входы дополнительных блокировок

Ячейка регистра состоит из двух триггерных полуячеек — левой и правой. Левая полуячейка предназначена для хранения и передачи информационных единиц, правая — для информационных нулей. Перекрестные связи между полуячейками предотвращают одновременную запись в обе полуячейки и тем самым сохраняют порядок следования информации в конвейере.

На рис. 4.25 показана схема одноразрядного неплотного регистра на таких ячейках.



**Рис. 4.25.** Одноразрядный буферный конвейерный регистр FIFO:  
 $X_1, X_2$  — ПФС-входы;  $Y_1, Y_2$  — ПФС-выходы;  
 $C$  — управляющий сигнал (запрос считывания);  
 $I$  — индикаторный сигнал (запрет записи)

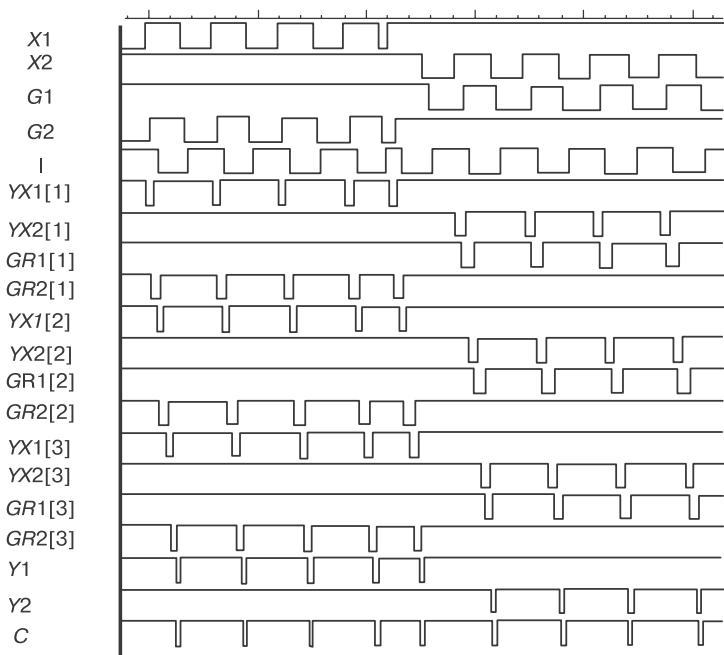
Схема работает следующим образом. При включении питания в разрядах устанавливаются произвольные значения. Поэтому для очистки регистра необходимо многократно переключить сигнал «запрос на считывание»  $C$ .

В процессе работы, в ответ на сигнал  $I = 0$  источник информации должен установить на входах ( $X_1, X_2$ ) рабочий парафазный код 01 или 10. После этого первая ячейка срабатывает и выдает сигнал запрета записи  $I = 1$ . В ответ на этот сигнал источник должен перейти в состояние спейсера и выдать код 11. В этой ситуации, если вторая и третья ячейки свободны (конвейер — неплотный!), информационный бит переписывается во вторую ячейку, а первая переходит в режим ожидания и формирует сигнал  $I = 0$ . Далее принятый бит перемещается к выходу регистра в зависимости уже не от источника, а от заполнения соседних ячеек.

В конце регистра, если сигнал  $C = 0$ , бит останавливается в предпоследней ячейке, а выходы ( $Y_1, Y_2$ ) находятся в спейсере 11. При поступлении запроса на считывание  $C = 1$  бит переходит в последнюю ячейку, и выходы регистра получают соответствующие значения 01 или 10. После принятия информации приемник должен перевести сигнал  $C$  в значение 0.

Если считывания в конце регистра не происходит, регистр заполняется информацией и формирует запрет записи  $I = 1$ . Если регистр пуст, то в ответ на запрос  $C = 1$  выходы ( $Y_1, Y_2$ ) остаются в спейсере 11 до прихода какого-либо бита в предпоследнюю ячейку.

На рис. 4.26 показан результат моделирования такого регистра, состоящего из четырех ячеек, в случае, когда приемник оперативно принимает приходящие сигналы (сигнал  $C$  меняется достаточно быстро после изменения  $Y_1$  или  $Y_2$ ).



**Рис. 4.26.** Переходные процессы 4-разрядного конвейерного регистра FIFO при незамедлительном приеме информации в приемнике

На рисунке наглядно видны особенности как СС-схем, так и состоящих из них конвейеров: информационные импульсы продвигаются от начала к концу регистра без какого-либо внешнего тактирования. В первой половине временной диаграммы передвигаются единицы по верхнему каналу рис. 4.25 по цепочке:  $X_1 \rightarrow YX1 [1] \rightarrow YX1 [2] \rightarrow YX1 [3] \rightarrow Y_1$ . Во второй половине передвигаются нули по нижнему каналу:  $X_2 \rightarrow YX2 [1] \rightarrow YX2 [2] \rightarrow YX2 [3] \rightarrow Y_2$ .

На рис. 4.27 показан случай, когда приемная сторона не принимает сигналы ( $C = 0$ ).

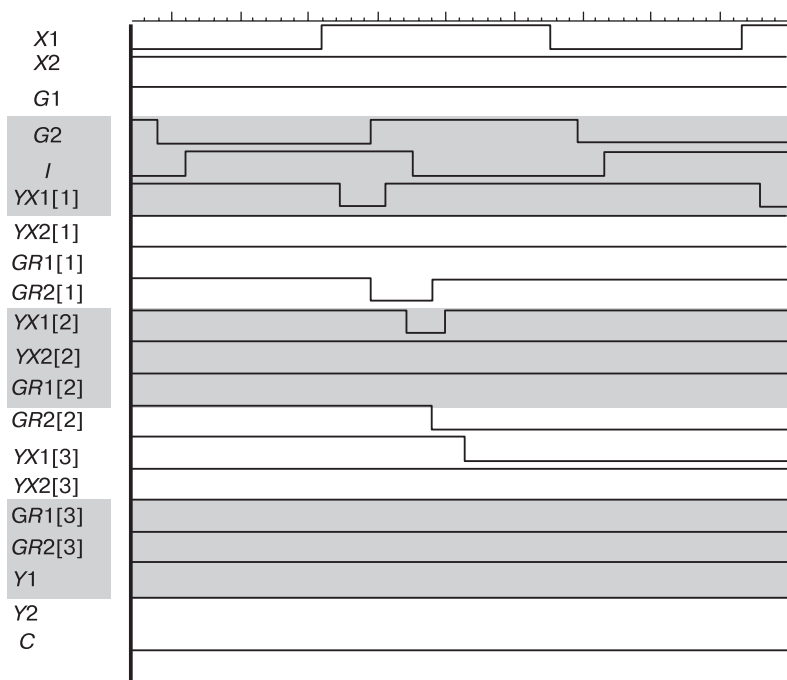
В этом случае первая посланная единица достигает третьей ячейки ( $YX1 [3] = 0$ ) и останавливается. Следующая единица останавливается уже в первой ячейке ( $YX1 [1] = 0$ ), так как конвейер неплотный.

#### 4.2.5. Счетчики

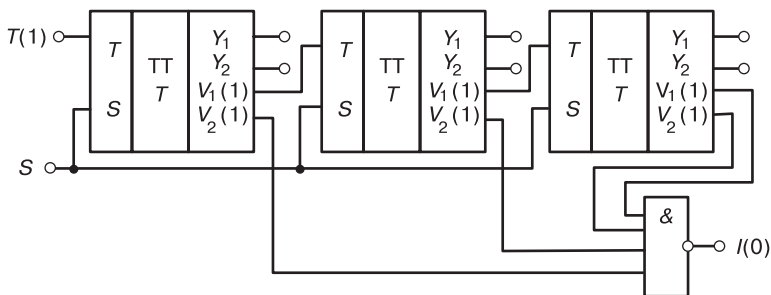
Счетчики строятся на основе  $T$ -триггеров.

На рис. 4.28 показан трехразрядный счетчик на ячейках с гашением, изображенных на рис. 4.18.





**Рис. 4.27.** Поведение 4-разрядного конвейерного регистра FIFO при отсутствии приема информации

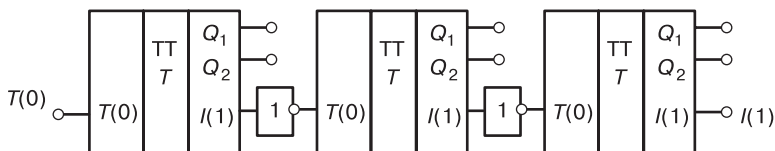


**Рис. 4.28.** Счетчик на триггерах с гашением:

$T$  — счетный вход;  $S$  — вход асинхронной предустановки;  $I$  — И-сигнал

В этой схеме счетный вход  $T$  каждой ячейки индицируется на ПФС-выходе ( $V_1, V_2$ ), поэтому для общей индикации счетчика вместо Г-триггера достаточно использовать элемент И-НЕ.

На рис. 4.29 показан также трехразрядный счетчик на  $T$ -триггерах без гашения (рис. 4.19).



**Рис. 4.29.** Счетчик на  $T$ -триггере без гашения:  
 $T$  — счетный вход;  $I$  — И-сигнал

В этой схеме счетный вход  $T$  каждой ячейки индицируется на И-сигнале ячейки  $I$ , поэтому весь счетчик индицируется на его последнем И-сигнале.

### 4.3. Преобразование моносигналов в самосинхронные

Схемы преобразования М-сигналов в самосинхронные могут потребоваться в ряде практических применений. Например, при сопряжении синхронных или асинхронных схем с самосинхронными, при включении сигналов датчиков в самосинхронную схему, при получении сигналов с удаленного источника и др.

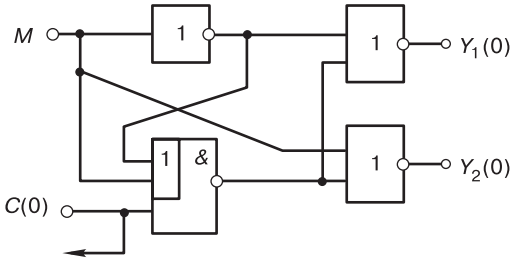
В литературе вопрос такого преобразования кратко обсуждается в [3, с. 355], однако представленная там схема многозарядного преобразователя [3, рис. 11.19] содержит ошибки и не является самосинхронной.

Проблема преобразования М-сигналов в самосинхронные состоит в том, что СС-схема работает в двухфазном режиме, и во избежание ошибок, связанных с состязаниями, изменения М-сигналов должны быть запрещены в некоторые периоды времени, определяемые принимающей СС-схемой. Иными словами, источники М-сигналов должны управляться сигналами, приходящими из СС-схемы. Источники М-сигналов, применяемые в обычных схемах, как правило, управляются специальными сигналами, и в роли этих сигналов могут использоваться сигналы от СС-схемы.

Простейшая схема преобразования одного М-сигнала в ПФС-сигнал [40] показана на рис. 4.30. Стрелкой обозначен сигнал для управления источником.

Дисциплина для этой схемы состоит в том, что сигнал  $M$  не должен меняться в рабочей фазе схемы, т. е. при  $C = 1$ .

Длительность рабочей фазы зависит не только от преобразователя, но и от всей СС-схемы, в которую преобразователь входит. Поэтому



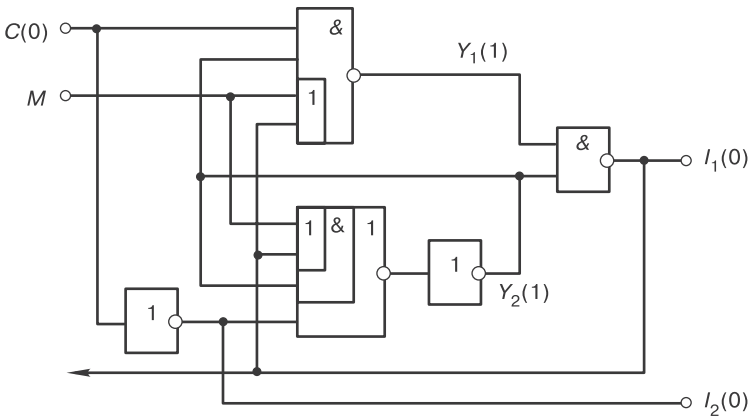
**Рис. 4.30.** Преобразователь М-сигнала в ПФС-сигнал: М — моносигнал; С — У-сигнал; (Y<sub>1</sub>, Y<sub>2</sub>) — ПФС-сигнал

источник и приемник (СС-схема) с таким преобразователем работают последовательно, с минимальным быстродействием.

Улучшить быстродействие при совместной работе можно за счет частичного распараллеливания работы источника и приемника, применяя более сложные схемы преобразования.

На рис. 4.31 показана схема с улучшенным взаимодействием источника и приемника [41]. В этой схеме преобразование начинается по сигналу  $C = 1$  и заканчивается, когда сигнал  $I_1$  станет равным 1, после чего источник может формировать новое значение моносигнала, не дожидаясь окончания рабочей фазы.

В приведенных схемах преобразованный сигнал не запоминается. Если его необходимо запомнить, то следует использовать триггеры с унарным входом, одноступенчатые или двухступенчатые.

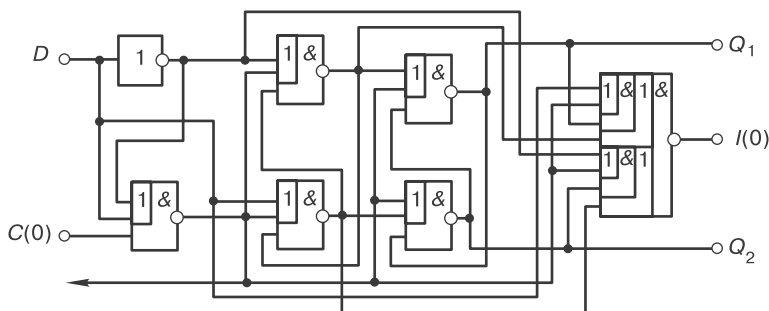


**Рис. 4.31.** Преобразователь М-сигнала в ПФС-сигнал с улучшенным взаимодействием источника и приемника: М — моносигнал; С — У-сигнал; (Y<sub>1</sub>, Y<sub>2</sub>) — ПФС-сигнал; I<sub>1</sub>, I<sub>2</sub> — И-сигналы

Одноступенчатый  $D$ -триггер на рис. 4.17 [37] может использоваться как преобразователь. Он запоминает преобразованный БС-сигнал только в течение последующей фазы спейсера. Для улучшенного взаимодействия с источником можно использовать сигнал, обозначенный  $NC$  на указанном рисунке.

Во многих других случаях требуется запоминать выходные сигналы на последующий полный цикл спейсера — рабочая фаза, для чего следует применять двухступенчатые триггеры.

Пример такого триггера [42] показан на рис. 4.32. Для управления источником также целесообразно использовать сигнал, обозначенный стрелкой.



**Рис. 4.32.** Двухступенчатый  $D$ -триггер с унарным входом для преобразования М-сигнала в БС-сигнал:

$D$  — моносигнал;  $C$  — У-сигнал;  $Q_1, Q_2$  — БС-сигнал;  $I$  — И-сигнал

Многоразрядные преобразователи можно построить как параллельные регистры из приведенных выше ячеек с необходимой индикацией.

# Глава 5

## АНАЛИЗ СХЕМ НА САМОСИНХРОННОСТЬ

---

Анализ схем на самосинхронность представляет собой один из главных этапов проектирования СС-схем. Анализ выполняется на ЭВМ с помощью программ, реализующих специальные алгоритмы и математические методы. Определить самосинхронность по уравнениям схемы или по ее графическому изображению «вручную», кроме тривиальных случаев, практически нереально.

Одна из основных трудностей анализа состоит в том, что самосинхронность является свойством схемы в целом, и любое изменение хотя бы одного ее элемента или связи требует повторного анализа всей схемы.

Другая трудность анализа заключается в необходимости учета всех возможных состояний схемы и допустимых переходов сигналов, возникающих при ее реальной работе. Эта трудность далее обсуждается в виде проблемы *полноты анализа*.

Ввиду сказанного анализ схем реальной сложности (десятки элементов и более) существующими методами требует очень больших вычислительных затрат, как правило, практически неприемлемых. Эти трудности можно преодолеть методами иерархического анализа (см. раздел 5.7).

Изложение вопросов анализа начнем с традиционной (не иерархической) постановки, когда схема представлена системой логических уравнений.

### 5.1. Начальные и основные состояния

Рассмотрим сначала вопрос о *начальных состояниях* в разомкнутых схемах.

Начальные состояния возникают при включении питания схемы. Они бывают *равновесными* и *неравновесными*. Равновесные состояния — это состояния, которые сохраняются неограниченно долго при неизменных входных сигналах. Неравновесные состояния без внешних воздействий со временем переходят в равновесные. Примером неравновесных состояний служат такие, в которых входы и выходы инверторов имеют одинаковые значения.

Неравновесные начальные состояния в процессе их переходов в равновесные часто порождают гонки. И вообще, период установления схемы в стабильное состояние после включения питания — это особый период, который обычно не считается рабочим. Поэтому такие состояния, как правило, на практике исключают из рассмотрения. Более того, опыт показал, что попытка анализа схем с неравновесными начальными состояниями практически всегда диагностирует нарушение самосинхронности, и это «засоряет» анализ. По указанным причинам будем учитывать в дальнейшем только равновесные начальные состояния.

Любое равновесное состояние разомкнутой схемы в каждой фазе — спейсере и рабочей — определяется множеством значений переменных, состоящим из двух наборов.

Первый набор порождается входными информационными сигналами — инфосигналами.

Каждый инфосигнал — пара ПФС- или БС-сигналов — порождает одну независимую переменную, поскольку сигналы в паре связаны условиями: БС-сигналы взаимно-противоположны (в статике) всегда, а ПФС-сигналы противоположны в рабочей фазе. Задавая независимой переменной произвольное значение, можно получить соответствующие значения сигналов пары. Входные У-сигналы не дают независимых информационных переменных, так как их значения определяются фазой работы.

Второй набор получается в процессе вычисления значений внутренних сигналов по выбранному первому набору. При этом вычислении некоторые сигналы могут оставаться произвольными, например один из выходов триггера. Такие сигналы также порождают свободные переменные, которые будем называть *переменными памяти*. Таким образом, в общем случае переменные памяти зависят от конкретных значений входных инфосигналов.

Задавая произвольные значения инфосигналам и соответствующим им переменным памяти, мы получим набор значений, однозначно определяющий состояние схемы. Каждый возможный такой набор будем называть *фактором состояния*. В соответствии

с фазой входных сигналов факторы состояния будут своими в каждой фазе.

Состояния схемы, определяемые факторами состояния, будем называть *основными*.

Реальная работа СС-схемы (с общей внешней обратной связью) происходит следующим образом. После того как схема установилась в одно из основных состояний текущей фазы, переключается общий индикаторный сигнал схемы, который через внешнее окружение инициирует поступление на вход нового набора значений, соответствующего другой фазе. Переменные памяти в этот момент сохраняются от предыдущей фазы. Далее схема начинает переходить из прежнего основного состояния через ряд промежуточных неравновесных состояний (связанных с переключениями элементов) в основное состояние новой фазы, с изменением переменных памяти. Затем процесс повторяется.

Таким образом, в процессе реальной работы СС-схема проходит поочередно основные состояния спейсера и рабочей фазы. Эти состояния являются равновесными для разомкнутой схемы (без общей обратной связи) в каждой фазе. Между основными состояниями схема проходит промежуточные неравновесные состояния.

## 5.2. Дисциплина входных наборов

Уточним понятие фазы работы с точки зрения динамики. Будем считать *фазой работы схемы* переход от основного состояния в предыдущей фазе к основному состоянию в следующей (новой) фазе. Фаза начинается с началом изменений входов, соответствующих новой фазе, и завершается установлением всех выходных фазовых сигналов в соответствующие новой фазе значения.

В силу СС-кодирования сигналов и двухфазной работы не всякий входной набор и не всякое изменение наборов являются допустимыми. Правила, определяющие допустимость, будем называть *дисциплиной входных наборов*.

Как указывалось ранее, СС-схема взаимодействует с внешней средой согласованно, по тем же принципам, как и работает сама. Поэтому считается, что входы СС-схемы являются выходами некоторой внешней СС-схемы, имеющей, как и всякая подобная схема, производные задержки своих сигналов.

Дисциплина входных наборов должна обеспечиваться внешней схемой, и правила этой дисциплины состоят в следующем.

1. У-сигналы на входе должны принимать значения, соответствующие наступающей фазе.

2. Для ПФС-входов оба сигнала пары в фазе спейсера должны принимать одинаковые спейсерные значения, а в рабочей фазе один и только один из них может принять противоположное значение.

3. Каждая пара БС-входов, понимаемая как выходы внешней БСЯ, может меняться только в одной из фаз.

Это требование объясняется тем, что только в таком случае удастся обеспечить монотонность переходов внутри любой СС-схемы и, соответственно, их правильность по Варшавскому [3] (отсутствие гонок). Собственно, это и было одной из целей введения двухфазной работы СС-схем: если допустить изменение пары БС-сигналов в каждой фазе, то обеспечить монотонность в обеих фазах при таких условиях невозможно.

Фазу работы, в которой разрешены изменения пары входных БС-сигналов, будем называть *транзитной фазой* для этой пары. Транзитные фазы для разных пар могут быть разными.

4. Для каждой пары БС-сигналов ее последователи (элементы, на входы которых поступают сигналы пары) на время изменения сигналов пары должны быть заблокированы от переключений другими, фазовыми входами схемы. В противном случае будут возникать состязания.

5. При выполнении предыдущего правила моменты изменений остальных У-входов, ПФС-входов, БС-входов как внутри пар, так и между парами не зависят друг от друга.

Таким образом, необходимым требованием к включению СС-схемы во внешнюю схему и, соответственно, к анализу ее самосинхронности будет соблюдение дисциплины входных наборов.

Во избежание излишних вычислений начинать анализ целесообразно не с произвольного состояния, которое может оказаться неравновесным, а с одного из основных состояний.

### 5.3. Согласованное замыкание

Как указывалось ранее, СС-схема работает с внешней средой согласованно, по принципу отрицательной обратной связи по фазе. Однако на начальном этапе проектирования СС-схема и ее фрагменты представлены в разомкнутом виде. Необходимость замыкания возникает в двух случаях. Во-первых, СС-схема конечного назначения должна



быть замкнута — это ее нормальный способ применения. Во-вторых, практические СС-схемы разрабатываются по частям, и для анализа ее фрагментов событийными методами необходимо эти фрагменты правильно замкнуть.

*Согласованное замыкание* предназначено для имитации работы внешней среды совместно с исследуемой разомкнутой схемой. Это замыкание должно обеспечить правильность переходов входных сигналов и дисциплину их изменения.

Переход от разомкнутых схем к замкнутым в литературе по самосинхронике не рассматривался, и согласованное замыкание поэтому нигде явно не описано. В [3] содержится похожий термин «согласованная реализация» для случая одного У-сигнала на входе и одного И-сигнала на выходе. В этом частном случае согласование схемы и ее окружения тривиально.

В известных по литературе примерах событийного анализа схема предъявляется для анализа уже в замкнутом виде, т. е. замыкание выходит за рамки собственно анализа. Кроме того, ни в одном из найденных примеров замкнутых схем не было информационных входов.

Рассмотрим простейший вид согласованного замыкания — с информационными константами (*константное замыкание*).

### 5.3.1. Константное замыкание

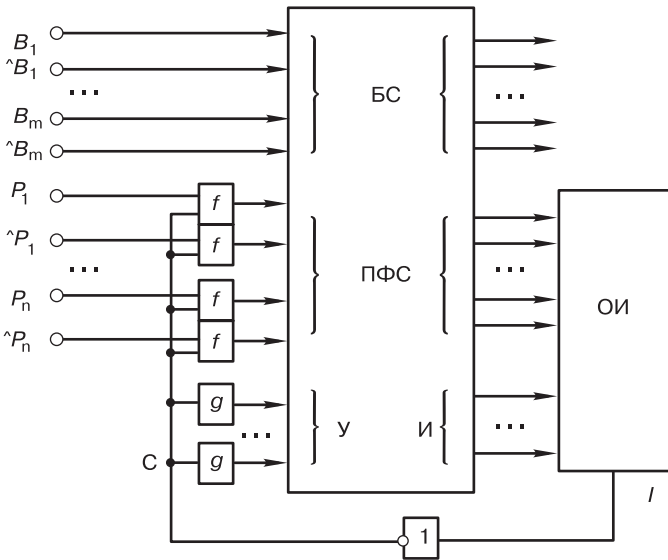
При константном замыкании все информационные входы считаются неизменными (константами), т. е. выступают как *параметры анализа*. В этом случае согласованное замыкание строится, как показано на рис. 5.1.

В схеме общей индикации ОИ все ее входы индицируются на одном выходе  $I$  (см. главу 3). На рисунке значение сигнала  $I$  соответствует завершившейся фазе схемы, а инвертор иллюстрирует отрицательность обратной связи (значения спейсеров сигналов  $I$  и  $C$  одинаковы).

Константное замыкание строится следующим образом.

1. Входные БС-сигналы заменяются взаимно-обратными двоичными числами 0 или 1 согласно константам  $B_k$  ( $k = 1, \dots, m$ ),  $m$  — число пар БС-сигналов.

2. На ПФС-входах появляются сигналы от констант  $P_k$  ( $k = 1, \dots, n$ ),  $n$  — число пар ПФС-сигналов. Значения спейсеров ПФС-сигналов  $P$  обозначим как  $S_p$ .



**Рис. 5.1.** Согласованное замыкание с информационными константами:  $B_1 \dots B_m$  — независимые константы БС-входов;  $P_1 \dots P_n$  — то же для ПФС-входов; ОИ — схема общей индикации;  $I$  — общий индикаторный сигнал;  $C$  — общий согласующий сигнал

Функции  $f(c, p)$  обеспечивают правильные (согласованные) рабочие и спейсерные значения ПФС-сигналов и реализуются в зависимости от спейсеров сигналов  $C$  и  $P$  по табл. 5.1.

Таблица 5.1

**Функции согласования**

	$S_c = 0$	$S_c = 1$
$S_p = 0$	$f = cp$	$f = \wedge cp$
$S_p = 1$	$f = \wedge c \vee p$	$f = c \vee p$

- Для  $Y$ -сигналов замыкание задается элементами с функцией  $g$ , которая определяется в зависимости от спейсера  $Y$ -сигнала  $S_u$  и спейсера  $S_c$ :

$$g(x) = x \text{ (повторение), если } S_u = S_c,$$

$$g(x) = \wedge x \text{ (инверсия), если } S_u \neq S_c.$$

Важно отметить, что *если имеется несколько  $Y$ -сигналов, то повторители и инверторы не могут быть ни удалены, ни объединены*, так как в противном случае нарушается независимость по моментам изменений  $Y$ -сигналов.

Построенное константное замыкание удовлетворяет условиям модели Маллера и обеспечивает допустимость переходов и правильное чередование фаз, т. е. корректно моделирует поведение внешней среды при неизменности информационных входов.

В общем случае такое замыкание недостаточно, так как в реальной работе информационные входы меняются. Согласованное замыкание с изменением инфовходов существенно сложнее приведенного выше не только из-за необходимости соблюдения дисциплины входов, но и перебора в определенном порядке входных информационных наборов. Данный вопрос связан с проблемой полноты анализа и рассмотрен ниже (см. раздел 5.5).

Константное замыкание поэтому может использоваться лишь как инструмент в рамках более общих методов анализа, а также для иллюстрации работы СС-схем.

Остановимся сначала на существующих методах анализа в событийном подходе.

## 5.4. Событийные методы анализа (замкнутых схем)

Как уже указывалось в разделе 2.5, в имеющейся литературе по анализу самосинхронности и независимости от задержек, начиная с работ Д. Маллера и В. И. Варшавского с коллегами и до настоящего времени (не считая публикаций автора), исследуются и обсуждаются исключительно событийные методы, традиционно описываемые в рамках теории автоматов.

Событийные методы анализа основаны на теории Маллера [1]. Постановка задачи в этих методах состоит в определении полумодулярности предъявленной замкнутой схемы (см. раздел 2.3.1).

При практической разработке СС-схем такая постановка неудобна из-за необходимости замыкания и размыкания схем и их фрагментов. Имеются и другие недостатки общего характера (см. раздел 2.5).

Характерными особенностями событийных методов являются замкнутость анализируемых схем и явное задание начального состояния схемы (значений выходов всех элементов).

Результат анализа событийными методами — заключение о полумодулярности, т. е. о наличии или отсутствии конфликтных переходов сигналов — состязаний на элементах. Полумодулярность, таким образом, есть свойство не только схемы, но и заданного начального состояния. Это свойство является более узким, чем самосинхронность, так как для самосинхронности необхо-

дима полумодулярность по всем реальным состояниям и переходам схемы.

Одна и та же схема при одних начальных состояниях может быть полумодулярной, а при других — нет. Поэтому для анализа следует брать только состояния, в которые схема попадает при нормальной работе, когда закончатся однократные переходы, могущие появиться при включении схемы. Правильный порядок действий здесь таков:

- разомкнуть общую обратную связь, например цепь на выходе инвертора на рис. 5.1;
- задать на вход разрыва одно из значений: 0 или 1, которое определит начальную фазу анализа;
- задать информационные константы;
- вычислить один из факторов состояния разомкнутой схемы;
- по выбранному фактору взять основное состояние разомкнутой схемы как начальное для анализа замкнутой.

Опыт показал, что при произвольном выборе начального состояния можно попасть в состояние, неравновесное для разомкнутой схемы. В этом случае анализ покажет, что самопроизвольный переход из неравновесного состояния в равновесное и переходы, порождаемые изменением обратной связи, будут конфликтовать, вызывая нарушение полумодулярности. В итоге пригодная схема может быть забракована только из-за неудачного выбора начального состояния.

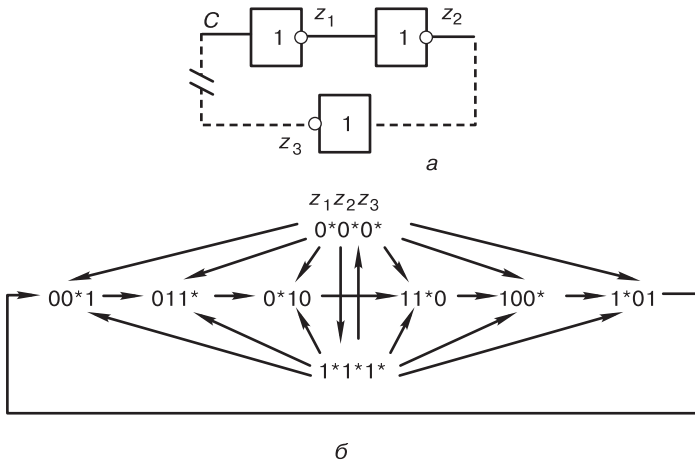
#### 5.4.1. Прямой метод по диаграммам переходов (ДП)

Данный метод непосредственно следует из теории Маллера, и потому исторически стал первым методом, пригодным для анализа. Он позволяет проверять свойство полумодулярности — отсутствие конфликтных переходов в диаграмме переходов (см. раздел 2.3.1).

Для иллюстрации метода рассмотрим простую схему — два последовательных инвертора. Согласованным замыканием для нее будет один инвертор в обратной связи (рис. 5.2, *а*).

Эта схема имеет восемь состояний, ее полная диаграмма переходов приведена на рис. 5.2, *б*.

Если для анализа взять какое-либо из двух состояний 000 или 111, то такой выбор будет неудачным: эти состояния не равновесны для разомкнутой схемы. В замкнутом состоянии они либо переходят друг в друга, либо переход из них в любое другое состояние конфликтен. Результат анализа в этом случае — нарушение полумодулярности — для практики неинтересен.



**Рис. 5.2.** Схема из двух инверторов с замыканием (а) и ее диаграмма переходов (б)

Для подходящего выбора начального состояния следует схему разомкнуть (место размыкания показано), задать сигналу  $C$  одно из возможных значений 0 или 1 и вычислить значения остальных сигналов. Расчет дает два основных начальных состояния: 010 и 101. При любом из них схема не совершает конфликтных переходов, т. е. полумодулярна.

В общем случае в соответствии с теорией Маллера (см. раздел 2.3.1) схема проходит некоторый начальный участок и попадает либо в тупик, либо в бесконечный финальный цикл.

Состояние тупика (остановки) замкнутой СС-схемы свидетельствует о ее неработоспособности и является признаком наличия в ней неисправностей (см. раздел 1.2). При анализе исправной схемы нормальная череда переходов должна привести ее в финальный цикл. Поэтому если тупик возник при анализе предположительно исправной схемы, то этот факт указывает либо на ошибку в схеме (обычно неправильную индикацию), либо на ошибку в замыкании.

В дальнейшем будем считать, что схема при анализе не попадает в тупик. Для схемы на рис. 5.2 видно, что любое из рассчитанных начальных состояний попадает сразу в финальный цикл, и начальный участок отсутствует.

В общем случае ни начальный участок, ни финальный цикл заранее не известны и определяются путем поиска. Поиск удобнее проводить не по отдельным состояниям, а по их множествам на каждом шаге процесса.

Множество состояний, в которых может находиться схема на данном шаге, получило название *слоя*.

Первый слой состоит из одного состояния — начального.

Каждое состояние в текущем слое проверяется на наличие возбуждений элементов. Возбуждение элемента происходит, если его значение (на выходе) не соответствует значениям входов. Например, если выход инвертора совпадает по значению со входом, инвертор возбужден.

Если хотя бы в одном состоянии возбуждений нет, то это значит, что схема находится в тупике, и анализ прекращается.

Далее по текущему слою строится следующий слой. Рассматривается каждое состояние текущего слоя. Предположим, что в некотором состоянии текущего слоя имеется  $n$  возбужденных элементов. Данное состояние добавляет в следующий слой  $n$  состояний, в каждом из которых один из возбужденных элементов (поочередно) переключается, а остальные остаются возбужденными. Переключение элемента при таком переходе воздействует на его последователи. Если это переключение снимает возбуждение какого-либо из оставшихся  $n - 1$  возбужденных элементов текущего состояния, то возникает конфликт, означающий нарушение полумодулярности. Анализ прекращается. Это же переключение элемента может возбудить другие элементы в том состоянии, которое добавляется в следующий слой.

Полученное новое состояние может совпадать с одним из состояний следующего слоя. Во избежание увеличения вычислений следует проводить проверку на совпадение и не допускать повторных вхождений состояний в слое.

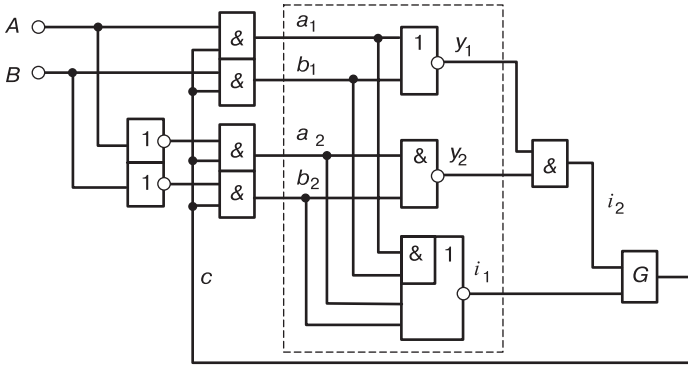
Цикличность обнаруживается путем запоминания некоторого слоя, называемого *контрольным*, и сравнения с ним последующих слоев.

Стратегия поиска финального цикла состоит в следующем.

В начале анализа проходится несколько слоев (предполагаемый начальный участок), запоминается контрольный слой и задается целое число  $N$ , предположительно превышающее длину цикла. Последующие  $N$  слоев сравниваются с контрольным. Если какой-то слой совпал с контрольным, анализ завершается (цикл найден). В противном случае запоминается новый контрольный слой (так как не известно, пройден ли начальный участок), и число  $N$  увеличивается. Процедура затем повторяется.

Для примера рассмотрим комбинационную схему с константным замыканием, реализующую в парафазном коде функцию ИЛИ-НЕ (а заодно и функцию ИЛИ, если поменять местами пара-

фазные выходы  $y_1$  и  $y_2$ ) (рис. 5.3). Собственно схема с индикатором показана в пунктирной рамке, остальные элементы — согласованное замыкание.



**Рис. 5.3.** Комбинационная схема ИЛИ-НЕ (ИЛИ) в парафазном коде с константным замыканием и информационными константами  $A$  и  $B$

Буквой  $G$  на рисунке обозначен  $\Gamma$ -триггер (см. главу 3), имеющий уравнение:

$$c = i_1 i_2 \vee c^-(i_1 \vee i_2), \tag{5.1}$$

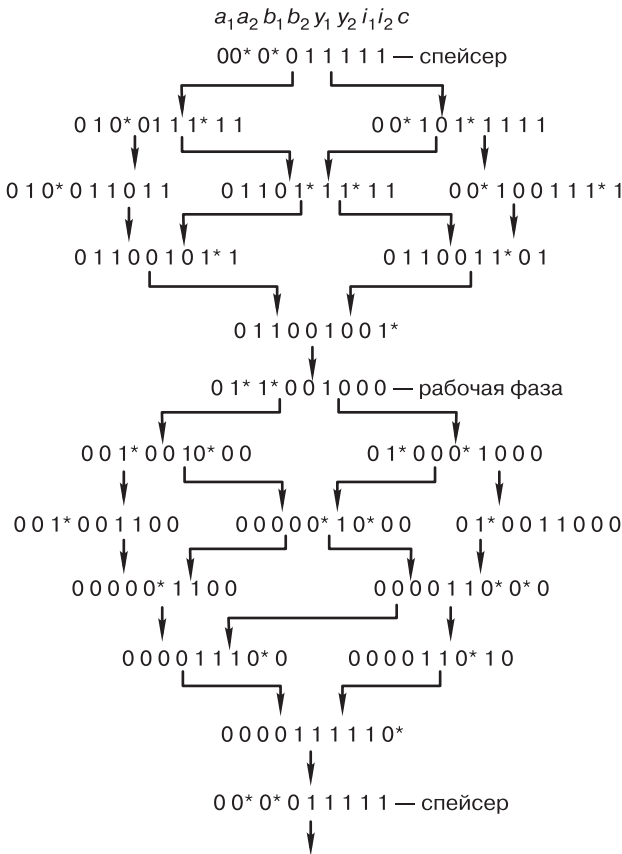
где  $c^-$  — предыдущее значение  $c$ .

На рис. 5.4 показана диаграмма переходов этой схемы при  $A = 0$ ,  $B = 1$ . Каждый слой записан в отдельной строке. Первый слой, включающий одно начальное состояние, — спейсер, был взят как контрольный. Как видно из диаграммы, начальный участок при таком выборе отсутствует (что характерно для комбинационных схем).

Метод ДП очень трудоемок по вычислительным затратам. Если для схемы на рис. 5.2 каждый слой содержит по одному состоянию, то для не очень сложных схем, имеющих 15—20 элементов (уравнений), слои могут содержать тысячи, сотни тысяч и более состояний. Вычислительная сложность метода оценивается как экспонента от числа элементов.

Рассмотрим вопрос, каким образом проверка полумодулярности замкнутой схемы обеспечивает и проверку безошибочности ее работы (см. раздел 1.2).

Конфликтный переход в ДП — это потенциальная гонка в реальной схеме. Поэтому полумодулярность — отсутствие конфликтных переходов — гарантирует отсутствие гонок при любых задержках элементов, или независимость от задержек по Маллеру.



**Рис. 5.4.** Диаграмма переходов схемы, изображенной на рис. 5.3

Теперь о связи полумодулярности со свойством отказобезопасности — остановке схемы при появлении неисправности типа КНЗ-01. Этот вопрос непосредственно связан с индицируемостью элементов (см. раздел 2.5.2). Предположим, что некоторый выход элемента не индицируется на общем индикаторном сигнале схемы, идущем в обратную связь (индицируемость определяется предварительно, при разомкнутой обратной связи). Пусть рассматриваемый элемент из-за изменений его входов возбудился, но не переключился в значение, соответствующее текущей фазе (например, не успел либо «залип» в предыдущем состоянии из-за неисправности), т. е. остался возбужденным. Так как сигнал не индицируется, общий индикатор «не заметит» отсутствия переключения и по обратной связи инициирует переход в другую фазу. В тече-



ние следующей фазы возбуждение элемента будет снято, что означает конфликт и нарушение полумодулярности.

В реальной работе схемы если элемент индицируется, то общий индикатор будет ожидать его переключения, и при неисправности на этом элементе схема остановится.

Таким образом, обнаружение конфликтного перехода элемента в методе ДП показывает, что либо на элементе возможна гонка (если элемент индицируется на общем сигнале), либо схема не остановится при появлении КНЗ-01 на выходе этого элемента (если элемент не индицируется). В результате оба свойства безошибочности работы проверяются единым способом — вычислением конфликтов.

Реальная работа СС-схемы происходит в замкнутом виде, поэтому приведенные рассуждения позволяют сделать и такой вывод: благодаря обратной связи (с корректным замыканием) неиндицируемые элементы подвержены гонкам. По этой причине все элементы разомкнутой СС-схемы должны быть индицируемыми, что еще раз подтверждает тезис, изложенный в разделе 2.5.2: индицируемость — необходимое свойство разомкнутых СС-схем.

Остановимся еще на одном аспекте связи разомкнутых и замкнутых схем.

Как следует из [3], наличия бесконечного финального цикла и полумодулярности еще недостаточно для самосинхронности замкнутой схемы. Требуется определить отсутствие *фиктивных классов эквивалентности (ФКЭ)* в ДП.

ФКЭ — это замкнутый цикл, в котором какая-либо переменная находится в одном и том же состоянии и возбуждена. Если в элементе, соответствующем этой переменной, возникает константная неисправность типа «залипания», то реальная схема не остановится по этой неисправности. ФКЭ не нарушает полумодулярность ДП, но схема не самосинхронна.

ФКЭ появляются, если в разомкнутой схеме есть самогенерация [3]. Выявить наличие ФКЭ в ДП довольно сложно, более целесообразно проверять устойчивость разомкнутой схемы. Например, при расчете начальных состояний по разомкнутой схеме, как он описан в начале раздела 5.4, если статических состояний не существует (их невозможно вычислить), то это говорит о наличии самогенерации.

#### 5.4.2. Метод диаграмм изменений

Как и предыдущий, данный метод предназначен для анализа полумодулярности замкнутых схем. Исходная схема здесь также нуждается в подготовке к анализу (замыканию).

Метод *диаграмм изменений* (ДИ) разработан В. И. Варшавским с сотрудниками [24] для преодоления вычислительной сложности предыдущего метода. Метод основан на связи диаграмм изменений с соответствующими диаграммами переходов схемы, т. е. базируется на теории Маллера.

Диаграмма изменений представляет изменения отдельных переменных схемы графически. В простой форме диаграммы в вершинах графа указываются изменения переменных: знаком плюс — из 0 в 1, знаком минус — из 1 в 0. Сплошные входящие дуги графа означают, что событие в вершине происходит, когда произойдут все предшествующие данной вершине события (условия типа И), пунктирные дуги — если произойдет хотя бы одно из предшествующих событий (условия типа ИЛИ). Для интересующих нас замкнутых схем ДИ будет циклической.

Диаграммы изменений — значительно более компактное представление процессов в схемах, чем диаграммы переходов.

В качестве примера на рис. 5.5 показана ДИ для схемы, приведенной на рис. 5.3.

Для некоторых исходных уравнений схемы требуется вводить фиктивные переменные. В данном примере такие переменные порождаются уравнением (5.1) и описываются равенствами:

$$f_1 = i_1 i_2, \quad f_2 = c i_1, \quad f_3 = c i_2. \quad (5.2)$$

В отличие от диаграмм переходов, где состояния схемы строго переходят от предыдущего слоя к последующему, ДИ не отображают такой очередности. События в ДИ могут обгонять или ожидать друг друга. Поэтому возникают проблемы одновременности переключений, достижимости и ряд других, объединенных понятием *корректности* ДИ. Например, корректность по переключениям состоит в требовании поочередности положительных и отрицательных переходов каждой переменной.

В. И. Варшавским с сотрудниками доказаны важные утверждения относительно ДИ, в частности следующие:

- 1) для анализа свойств ДИ достаточно пройти не более трех первых ее циклов;
- 2) если ДИ корректна, она эквивалентна полумодулярной диаграмме переходов;
- 3) вычислительная сложность метода оценивается как полином степени не более 3 от числа переменных.

Таким образом, анализ по ДИ сводится к проверке корректности некоторого ограниченного множества событий, порожденных при прохождении ДИ.

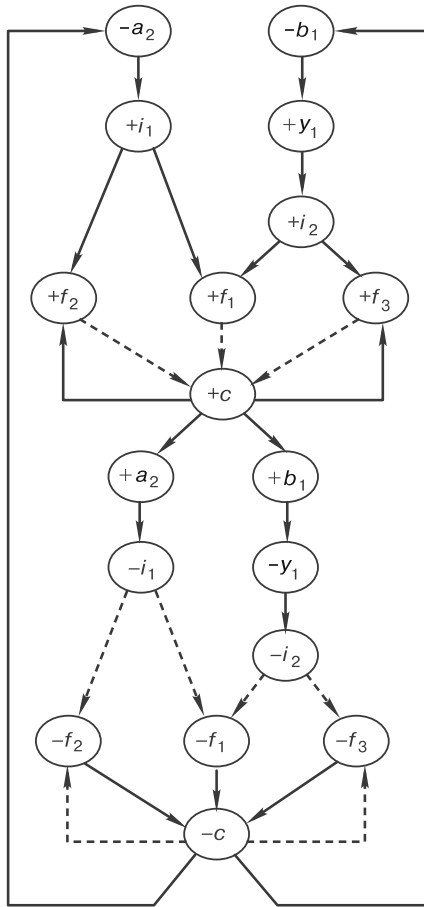


Рис. 5.5. Диаграмма изменений схемы, изображенной на рис. 5.3

В процессе анализа по уравнениям замкнутой схемы, поурядно, строится ДИ и одновременно проверяется ее корректность. В конечном итоге выявляются ситуации, когда два одновременных события (переключения) влекут за собой и положительный, и отрицательный переходы одной и той же переменной. Такие ситуации означают нарушение полумодулярности.

Из упомянутого утверждения 3 видно, что метод весьма быстродействующий, реализующие его программы работают на порядки быстрее, чем по методу диаграмм переходов. Основные трудности метода состоят в переходе от системы логических уравнений к ДИ, интерпретации и обработке ряда условий типа ИЛИ.

Приведенные оценки сложности обоих событийных методов анализа относятся к случаю одной комбинации информационных констант и переменных памяти. Полный же анализ требует изменения этих параметров, а такой вопрос ранее не исследовался.

## 5.5. Полнота анализа

Введем понятие *сеанс анализа* — однократный расчет самосинхронности (или полумодулярности) всей схемы.

Для событийных методов сеансом анализа будет анализ на полумодулярность при одном факторе состояния (см. раздел 5.1).

До недавнего времени считалось, что способ «прямого перебора» — выполнения всех сеансов анализа с константным замыканием и всеми возможными факторами состояния — достаточен для полного анализа. Однако выяснилось, что это не так, и причина состоит в следующем.

Для полноты анализа требуется не только пройти все основные состояния разомкнутой схемы (что обеспечивается прямым перебором), но и проверить все реализуемые переходы между основными состояниями. Последнее условие требует изменения входов, т. е. выхода за рамки константного замыкания, чего при прямом переборе не происходит.

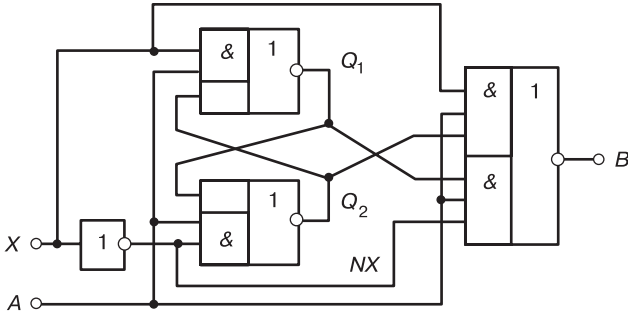
Неучет изменений инфовходов в сеансе даже с полным перебором факторов состояний приводит к ошибочным выводам, что было впервые замечено при практическом проектировании СС-схем в ИПИРАН (Ю. А. Степченков).

Кроме того, во избежание ошибок следует учитывать, что не все переходы между основными состояниями реализуются. Возьмем пример схемы, содержащей сдвиговый регистр разрядностью более единицы. Два основных состояния этой схемы — одно со всеми единицами в регистре, другое со всеми нулями — никогда друг в друга не переходят. Если формально подобный переход проанализировать, то может выявиться конфликт, и работоспособная СС-схема будет забракована.

Под *полнотой анализа* СС-схем будем понимать проверку в одном сеансе анализа всех основных состояний и всех реальных переходов между ними с контролем самосинхронности.

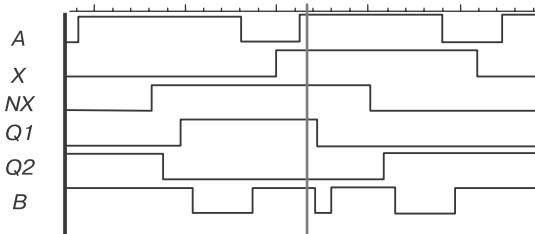
В событийном подходе, и вообще в литературе по самосинхронике, подобная задача полного анализа до публикации автора [43] не рассматривалась.

Например, схема, показанная на рис. 5.6, приведена в [3, рис. 4.11, в] как самосинхронная. Действительно, если применить константное замыкание, в сеансе анализа сигнал  $X$  (унарный информационный вход) не меняется. При любых его значениях анализ показывает соблюдение полумодулярности, и схема формально самосинхронна.



**Рис. 5.6.** Триггер с нарушением самосинхронности:  
 $X$  — УС-вход;  $A$  — У-сигнал;  $B$  — И-сигнал.

В реальных условиях именно допустимое изменение сигнала  $X$  создает условия для возникновения гонки в триггере. Предположим, схема установлена в такое состояние в фазе спейсера:  $A = 0$ ,  $B = 1$ ,  $X = 0$ ,  $Q_1 = 1$ ,  $Q_2 = 0$ ,  $NX = 1$ . По принципу самосинхронности задержки элементов могут быть любыми. Пусть задержки инвертора и нижнего плеча триггера достаточно велики, так что сигналы  $NX$  и  $Q_2$  не успевают измениться до изменения сигнала  $Q_1$ . В соответствии с дисциплиной в фазе спейсера разрешено изменение сигнала  $X$ , после чего может меняться сигнал  $A$  для перехода в рабочую фазу. После перехода сигнала  $X$ , а затем и  $A$  в единицу в нижней части индикаторного элемента появятся три единицы, и сигнал  $B$  перейдет в нуль. Далее по условию изменится сигнал  $Q_1$ , который вернет  $B$  в единицу. На рис. 5.7 показан результат моделирования схемы в описанных условиях.



**Рис. 5.7.** Моделирование несамосинхронного триггера с унарным входом

Маркер на рисунке (вертикальная линия) отмечает ситуацию непосредственно перед гонкой. Справа от маркера виден ложный импульс сигнала  $B$  — результат состязания, обусловленного нарушением самосинхронности.

(Корректная схема триггера с унарным инфовходом приведена на рис. 4.17.)

Таким образом, для обеспечения полноты анализа возникает задача изменения инфовходов в сеансе, т. е. выбора количества и последовательности входных наборов, а также специальной организации процесса анализа.

Задачу полноты удобно рассмотреть как последовательность решения следующих подзадач.

1. Последовательно устанавливать схему в каждое из основных состояний.
2. Проверять все реальные переходы из установленного состояния в другие основные.

### 5.5.1. Обеспечение полноты в событийном подходе

Как показывают приведенные выше примеры, константного замыкания, представленного на рис. 5.1, явно недостаточно для обеспечения полноты анализа в событийных методах.

Отметим, что в управляющих схемах (без информационных входов) все основные состояния проходятся автоматически, и проблема полноты не возникает при условии выбора начального состояния из множества основных (до замыкания).

В общем же случае потребуется специальная организация процедуры анализа, например следующая.

1. Фиксируется (очередной) набор значений инфовходов и вычисляются все факторы состояния, соответствующие этому набору по разомкнутой схеме (см. раздел 5.1).
2. Выбирается очередной фактор из предыдущего пункта и по нему вычисляется начальное состояние.
3. Проводится множество сеансов анализа с константными замыканиями. В каждом сеансе задается выбранное начальное состояние, а входные константы принимают одно из возможных значений инфовходов, включая и исходное.
4. В соответствии с принципом перебора фактор меняется и действия повторяются начиная с пункта 2.
5. Вся описанная процедура повторяется по всем возможным наборам значений инфовходов начиная с пункта 1.

Оценим возможное количество сеансов для полного анализа. Пусть  $N$  — число независимых переменных инфовходов,  $M$  — число переменных памяти. Общее число состояний не будет превышать величины  $2^{N+M}$ . Для каждого состояния надо выполнить сеансы анализа со всеми возможными входными комбинациями, что дает  $2^N$  сеансов. В итоге верхняя оценка количества сеансов  $S$  будет иметь вид:

$$S = 2^{2N+M}. \quad (5.3)$$

К этим вычислительным затратам следует добавить затраты на поиск подходящих начальных состояний, соответствующих всем возможным комбинациям входных наборов.

Как обычно, в некоторых частных случаях фактическое количество необходимых сеансов будет меньше оценки, например, для параллельного регистра.

## 5.6. Функциональный метод анализа разомкнутых схем (ФМА)

Функциональный метод анализа основан на принципах, изложенных в разделе 2.5. Метод впервые схематично описан в неполном виде в [44] и более подробно в [45].

При применении ФМА требуется, чтобы разомкнутая схема была устойчивой в обеих фазах, иначе она не работоспособна.

В отличие от анализа замкнутых схем, где наличие общей обратной связи переводит анализ индицируемости в анализ состязаний, для разомкнутых схем эти два вида анализа необходимо делать отдельно.

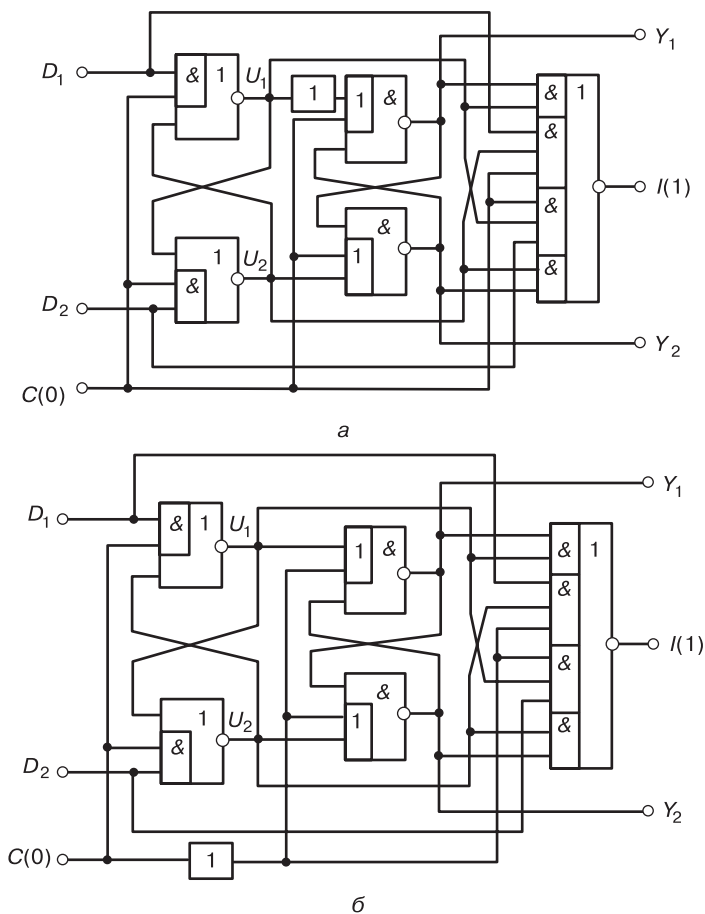
Исходя из сказанного в разделе 1.2, критерий самосинхронности разомкнутых схем в функциональном подходе, при соблюдении соглашений модели Маллера, можно сформулировать так.

*Разомкнутая схема, имеющая типовой интерфейс (см. рис. 2.13 в разделе 2.5.1), будет самосинхронной, если в каждой фазе (рабочей и спейсерной) обеспечиваются с исчерпывающей полнотой два условия:*

- 1) *индицируемость всех элементов схемы;*
- 2) *отсутствие гонок на всех элементах при любых задержках (независимость от задержек).*

Этот критерий аналогичен результату из [3, Определение 4.8], сформулированному для конечных автоматов.

Различие между анализом разомкнутых и замкнутых схем можно проиллюстрировать примерами на рис. 5.8.



**Рис. 5.8.** Схемы с нарушением самосинхронности:  
*a* — полумодулярная, но не индицируемая;  
*б* — индицируемая, но не полумодулярная

Эти примеры отличаются от прототипа на рис. 3.13 тем, что в них введены повторители.

В первой схеме гонок не возникает (она полумодулярна в разомкнутом состоянии), но сигнал на выходе повторителя не индицируется.

Во второй схеме все сигналы индицируются, но задержка повторителя порождает гонку во второй БСЯ триггера.

Если же в каждой из этих схем сделать константное замыкание и проанализировать каким-либо событийным методом, то в обоих случаях будет обнаружено нарушение полумодулярности.



Технологически ФМА имеет ряд отличий от событийных методов. Главное из них — работа с логическими функциями, а не с состояниями (в двоичных числах) и переходами между ними.

Еще одно отличие ФМА от событийных методов — явный учет параметров взаимодействия схемы с ее окружением. В замкнутых схемах все особенности такого взаимодействия «спрятаны» в замыкании и в самом анализе не фигурируют. Для разомкнутых схем окружение явно представлено интерфейсом, и без учета взаимодействия анализ провести невозможно.

Кроме того, ФМА в функциональном подходе рассматривается как этап на пути к анализу схем любого размера (состоящих из любого количества элементов) — иерархическому (см. раздел 5.7). Проведение такого анализа требует предоставления информации о взаимодействии анализируемой схемы с ее окружением на более высокий уровень иерархии. Поэтому одной из целей ФМА принято получение необходимых данных для иерархического анализа.

Исходными данными для анализа должны быть следующие.

1. Система логических уравнений всех элементов.
2. Указание групп входных и выходных сигналов и их самосинхронный тип (СС-тип) согласно типовому интерфейсу (см. раздел 2.5.1).
3. Для входных фазовых сигналов — значение спейсеров, 0 или 1.

Эти исходные данные минимально необходимы и достаточны для однозначности расчетов.

### 5.6.1. Задачи анализа разомкнутых схем

Необходимым условием индицируемости по [3] является монотонность функций элементов.

Монотонность по [3] отличается от традиционного определения, приводимого в учебниках. Традиционное понятие монотонности касается всех аргументов функции, и ее признаком является существование ДНФ функции, в которую все аргументы входят без отрицаний. Понятие монотонности в книге [3] более детализировано — оно относится к каждому аргументу в отдельности. Признаком монотонности функции по некоторому аргументу будет существование ее ДНФ, в которую этот аргумент входит либо без отрицаний (изотонная функция), либо только с отрицаниями (антитонная функция). Например, функция  $f = a \wedge b \vee b \wedge c$  немонотонна в традиционном смысле, но изотонна по аргументу  $a$  и антитонна по аргументу  $c$  по книге [3].

Монотонность функций в заданных логических уравнениях, вообще говоря, можно не проверять — ошибки выявятся при анализе самосинхронности. Но с целью возможно более раннего обнаружения ошибок сделать это все же целесообразно, тем более, что такая проверка легко выполнима.

Основные задачи анализа следующие.

1. Проверка индицируемости всех элементов схемы.
2. Определение параметров взаимодействия с окружением.
3. Проверка связей на всех элементах.

Задачи 1 и 3 есть собственно анализ самосинхронности схемы. Задача 2 предназначена для иерархического анализа. Полученная в задаче 2 информация частично используется и в задаче 3.

### 5.6.2. Формирование входных значений, обеспечение полноты

По заданным входам схемы, в соответствии с типовым интерфейсом (рис. 2.13), фиксируется набор независимых информационных переменных (см. раздел 5.1): каждая пара ПФС- и БС-сигналов, а также каждый УС-сигнал порождают по одной независимой переменной.

Обозначим исходные независимые входные переменные:

$A_i$  — для ПФС-входов  $a_i$  и  $b_i$ ,

$D_j$  — для БС-входов  $d_j$  и  $e_j$  (для УС — только  $d_j$ ), всего  $N$  переменных.

Индексы  $i$  и  $j$  пробегают множества соответствующих типов входов.

Для ПФС-входов, если текущая фаза — спейсер, оба сигнала пары получают одинаковые значения — спейсерные ( $S_i$  — константа 0 или 1):

$$a_i = b_i = S_i, \quad (5.4)$$

а в рабочей фазе ПФС-входы должны различаться:

$$a_i = A_i, \quad b_i = \bar{A}_i. \quad (5.5)$$

БС-входы в любой фазе получают значения (для УС-входов — только  $d_j$ ):

$$d_j = D_j, \quad e_j = \bar{D}_j. \quad (5.6)$$

Для проведения расчетов в какой-либо фазе (текущей) требуется знать начальные условия — значения входов и установившиеся значения элементов для предыдущей фазы. Поэтому делается предварительный расчет предыдущей фазы: на входах устанавливаются константы в зависимости от типа фазы и равенств (5.4)—(5.6).

Все расчеты как для предыдущей, так и для текущей фазы производятся в общем виде, как расчеты функций входных независимых переменных. В процессе расчета определяются и переменные памяти (ПП) — переменные (сигналы), значения которых в данном расчете произвольны. (При нормальной работе значения ПП остаются от предыдущей фазы — запоминаются.) Эти переменные добавляются к независимым переменным. Набор ПП для разных фаз обычно бывает разным.

В новой — текущей — фазе на входы схемы могут прийти другие значения, отличные от тех, что были на входах в предыдущей фазе. Для обеспечения полноты будем устанавливать в текущей фазе на входы новые значения: в соответствии с типом текущей фазы и равенствами (5.4)—(5.6), но с заменой  $A_i$  на  $A_i^*$  и  $D_j$  на  $D_j^*$ .

Смысл введения новых независимых переменных состоит в имитации изменений внешних сигналов схемы при ее работе. Действительно, наличие двух независимых переменных для инфовходов моделирует все четыре возможных перехода между старыми и новыми значениями: 0—0, 1—1, 0—1, 1—0. В результате будут учтены все входные состояния: и те, в которых схема начинает работу, и те, в которых она может оказаться как при неизменных, так и при изменяющихся входах.

Таким образом, полнота анализа обеспечивается ценой удвоения количества независимых переменных для инфовходов ( $2N$ ) и соответствующего увеличения вычислительной сложности.

Вычисляемые далее функции выходов элементов в текущей фазе будут зависеть от переменных  $A_i$ ,  $A_i^*$ ,  $D_j$ ,  $D_j^*$ , а также от  $M_m$  — ПП, полученных в предыдущей фазе. Значения перечисленных переменных в рамках текущих расчетов будут оставаться неизменными, поэтому будем называть их текущими *параметрами анализа*, и их набор обозначать буквой  $G$ :

$$G = \{A_i, A_i^*, D_j, D_j^*, M_m\}. \quad (5.7)$$

Заметим, что параметры анализа будут разными для разных фаз.

### 5.6.3. Проверка индицируемости элементов

Индицируемость элементов проверяется согласно правилам индицирования (см. раздел 2.5.3) в каждой фазе по отдельности. Проверка ведется прямым методом — имитацией неисправностей КНЗ-01 на выходах элементов и сравнением результатов расчетов исправной и неисправной схем.

После предварительного расчета предыдущей фазы в текущей фазе становятся известными значения всех элементов в этой предыдущей фазе. Далее должны быть изменены входы схемы в соответствии с переходом от предыдущей фазы к текущей и с учетом полноты анализа. Так как индицируемость есть свойство установившихся режимов, дисциплина здесь учитывается тем, что на входы устанавливаются значения текущей фазы. После этого проводится расчет текущей фазы.

В каждой фазе по полученным функциям фазовых выходов схемы надо сформировать общий виртуальный индикаторный сигнал  $W$ , который будем называть *оценочным*, с помощью уравнений  $\Gamma$ -триггеров (3.3), (4.1) или (4.2).

Оценочный сигнал — это сигнал, индицирующий текущую фазу. При правильном построении схемы он должен быть равным 0 или 1 и различным в разных фазах.

Пусть значение оценочного сигнала в текущей фазе равно  $W_i$  (это 0 или 1).

Рассмотрим элемент схемы с индексом  $k$ . Функцию его выхода в предыдущей фазе обозначим как  $F_{kp}(G)$ , а в текущей фазе — как  $F_{kt}(G)$ . При одних параметрах анализа выход элемента может меняться при переходе в текущую фазу, а при других — оставаться неизменным. Индицироваться, по определению, может только меняющийся сигнал. Поэтому необходимо ввести условие изменчивости  $k$ -го выхода:

$$F_{kp}(G) \oplus F_{kt}(G) = 1. \quad (5.8)$$

Для проверки индицируемости элемента сделаем расчет состояния схемы и оценочного сигнала, имитируя «залипание» (одиночную выходную КНЗ-01 — см. раздел 1.2) на  $k$ -м выходе. В этом расчете выход элемента будет оставаться неизменным и равным выражению, полученному в предыдущей фазе. Пусть в таких условиях вычислено выражение для оценочного сигнала  $W_k(G)$ .

Теперь с учетом изменчивости можно записать условие индицируемости  $k$ -го элемента:

$$[W_k(G) \equiv W_i][F_{kp}(G) \oplus F_{kt}(G)] = 0. \quad (5.9)$$

Действительно, если выход элемента не меняется, выражение во вторых квадратных скобках обращается в нуль. Если выход меняется, то оценочные выражения в первых квадратных скобках должны быть взаимно обратными, что тоже обратит (5.9) в нуль.

Если условие (5.9) выполняется тождественно (при любых параметрах анализа),  $k$ -й элемент индицируется. И наоборот, любая ком-

бинация параметров, обращающая левую часть (5.9) в единицу, покажет условия нарушения индицируемости — при каких значениях ПП и каких переходах входных инфосигналов это нарушение происходит. Если, например, после всех сокращений и приведения к ДНФ левая часть (5.9) содержит импликанту  $A_i \wedge A_i^*$ , то это значит, что нарушение индицируемости произошло при переходе из предыдущей фазы в текущую, когда сигнал  $a$  изменился из 1 в 0.

Рассмотрим пример двухступенчатого триггера на рис. 5.8, а. Обозначим выход повторителя буквой  $V$  и проверим его индицируемость в рабочей фазе.

Приведем исправные установившиеся значения сигналов в предыдущей фазе (спейсере) и текущей (рабочей) фазе:

Спейсер	Рабочая фаза
$C = 0$	1
$D_1 = D$	$D^*$
$D_2 = \wedge D$	$\wedge D^*$
$U_1 = \wedge M$	$\wedge D^*$
$U_2 = M$	$D^*$
$V = \wedge M$	$\wedge D^*$
$Y_1 = M$	$M$
$Y_2 = \wedge M$	$\wedge M$
$I = 1$	0,

где  $D$  — значение БС-входа  $D_1$  в предыдущей фазе (спейсере);  $D^*$  — значение этого входа в текущей (рабочей) фазе;  $M$  — переменная памяти первой ступени триггера в спейсере.

Вместе значения  $D$ ,  $D^*$  и  $M$  будут параметрами анализа в данном случае.

В этом примере роль оценочного сигнала выполняет сигнал  $I$ , и для него  $W_i = 0$ . Для выхода  $V$  имеем (его индекс опущен):  $F_p(G) = \wedge M$ ,  $F_i(G) = \wedge D^*$ .

В новом расчете рабочей фазы используем «залипшее» значение сигнала  $V$  (оставшееся от спейсера)  $\wedge M$ . В результате находим выражение оценочного сигнала  $W_k(G) = I = 0$ . Подставляя полученные выражения в (5.9), определяем условие индицируемости выхода  $V$ :

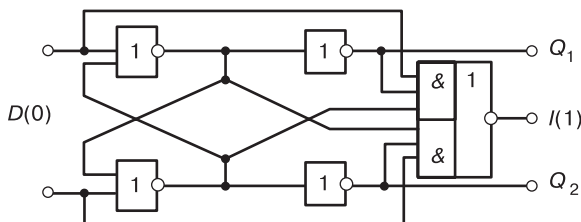
$$D^* \wedge M \vee \wedge D^* M = 0. \quad (5.10)$$

Это условие означает, что новое значение БС-входов, изменившееся в спейсере (транзитной фазе для входов этого примера), должно

совпадать с тем, что было запомнено в первой ступени в предыдущем цикле «рабочая фаза — спейсер». В свою очередь, это значит, что входы  $D_1$  и  $D_2$  не должны изменяться. Условие (5.10) и реализуется при константном замыкании.

В реальности, когда входы меняются,  $D^* \neq M$ , условие (5.10) нарушается, и схема на рис. 5.8, *a* оказывается не самосинхронной.

Еще один пример — попытка простым способом построить из триггера на рис. 3.10, *a* самосинхронный триггер с усиленными выходами (рис. 5.9).



**Рис. 5.9.** Триггер с усиленными выходами (с нарушением самосинхронности):

$D$  — ПФС-вход;  $Q_1, Q_2$  — БС-выход;  $I$  — И-сигнал.

Анализ этой схемы с константными замыканиями даже с полным перебором факторов состояний (их здесь 4 — по двум переменным: входу и памяти) приводит к заключению о полумодулярности схемы. Если же проанализировать схему с обеспечением полноты, например, функциональным методом, то обнаруживаются нарушения индикации сигналов  $Q_1$  и  $Q_2$  при изменении входов схемы с условием, аналогичным (5.10).

(Корректная схема триггера с усиленными выходами приведена на рис. 4.16.)

#### 5.6.4. Определение параметров взаимодействия с окружением

В этой задаче определяются и запоминаются индицируемость и необходимая дисциплина внешних сигналов схемы.

Для выходных фазовых сигналов — каждого И-сигнала и каждой пары ПФС-сигналов — заводится список индицируемых ими внешних входов и выходов схемы. В список не должны попадать входные БС-сигналы, так как они индицируются во внешней схеме.

Для правильного функционирования анализируемой схемы в составе окружения необходимо констатировать следующее.

*Дисциплина внешних сигналов определяется анализируемой схемой (каков должен быть порядок переключений без нарушения самосинхронности), а обеспечивается внешним окружением.*

Дисциплина ПФС-сигналов обеспечивается автоматически при правильном их подключении: значения спейсеров соединяемых сигналов должны совпадать.

Все проблемы возникают с дисциплиной БС-сигналов (см. раздел 5.2). Напомним, что во избежание состязаний перед началом изменения БС-входов те элементы, к которым эти входы подключены, должны быть заблокированы фазовым входом или входами (обычно это У-сигналы, но могут быть и ПФС). Данный порядок должна обеспечить внешняя схема.

Так как выходные БС-сигналы могут быть подключены к входам внешней схемы, необходимо выявить и их параметры переключений.

Итак, должны быть определены следующие параметры взаимодействия с окружением:

- 1) для фазовых выходов — списки индицируемых ими внешних сигналов;
- 2) для БС-входов — транзитная фаза, блокирующие фазовые входы и их значения, осуществляющие блокировку;
- 3) для БС-выходов — транзитная фаза, фазовые входы и их значения, разрешающие изменение этих выходов.

Кроме того, как показано в разделе 5.7, для БС-сигналов требуется знать, осуществляется ли блокировка непосредственно фазовыми входами или через посредство промежуточных элементов.

Списки индицирования вычисляются попутно, на этапе проверки индицируемости. Параметры изменения БС-сигналов определяются по уравнениям схемы.

Вопрос взаимодействия с окружением более подробно изложен в разделе 5.7.

### **5.6.5. Проверка состязаний**

Существуют способы обнаружения состязаний, основанные на моделировании в расширенных алфавитах (троичном и более), описанные, например, в [46] (троичный алфавит). Способы с использованием троичного алфавита не решают задачу до конца, а выявляют только «кандидатов» на состязания, которые должны быть дополнительно проверены другими методами, какими — неясно. Способы с использованием пятеричного алфавита не позволяют обобщить их на случай параметрического задания функций, необходимого для рассматриваемой задачи.

Вместе с тем СС-схемы имеют свою специфику — самосинхронное кодирование и двухфазный порядок работы. Эти особенности позволяют применить другие способы определения состязаний. Одна из важных особенностей СС-схем — каждый элемент в каждой фазе может переключаться не более одного раза.

В предлагаемом методе используются известные свойства монотонных логических функций. Пусть все аргументы функции меняются из 0 в 1 в любом порядке. Тогда, следуя [3], будем называть функцию *изотонной*, если она либо остается неизменной, либо меняется только из 0 в 1, и *антитонной*, если она не меняется или меняется только из 1 в 0.

Важно отметить, что если аргументы монотонной функции изменяются в одном направлении, состязания в ней не возникают при любом порядке изменения аргументов.

Признаком изотонной функции является существование ее ДНФ со всеми вхождениями аргументов без отрицаний, а признаком антитонной — то же со всеми вхождениями с отрицаниями.

Проверка состязаний проводится также отдельно в каждой фазе — спейсерной и рабочей.

Исходные данные и параметры анализа будут те же, что и при проверке индицируемости: результаты расчетов установившихся значений всех сигналов в предыдущей и текущей фазах.

Специальной заменой переменных строятся *проверочные функции* каждого элемента, свойства которых связаны с наличием состязаний в схеме.

Введем обозначения:

$G$  — множество параметров анализа,

$P_k(G)$  — выражение  $k$ -го сигнала в предыдущей фазе,

$T_k(G)$  — то же в текущей фазе,

$h_k$  — переменная, меняющаяся из 0 в 1 (*изо-переменная*),

$H$  — множество изо-переменных,

$V_k = P_k(G) \wedge h_k \vee T_k(G) h_k$  — вспомогательная функция сигнала. Вспомогательная функция меняется от выражения предыдущей фазы к текущей, когда изо-переменная меняется из 0 в 1.

Сделаем новый расчет схемы в текущей фазе, но вместо истинных выражений сигналов будем подставлять их вспомогательные функции.

Рассмотрим один из элементов схемы. В результате нового расчета выражение его выхода можно представить в виде  $F(G, H)$ . Когда все изо-переменные в этом выражении перейдут из 0 в 1, тогда и выражение перейдет от предыдущей фазы к текущей.



В зависимости от параметров анализа выход элемента в текущей фазе может либо остаться неизменным, либо измениться из 0 в 1, или из 1 в 0, и только один раз. Начальное и конечное значения определяются выражениями  $P(G)$  и  $T(G)$ .

Для анализа состязаний на элементе сформируем три проверочные функции:

$$\begin{aligned} F_1 &= [\wedge P(G)T(G)]F(G, H), \\ F_2 &= [P(G)\wedge T(G)]F(G, H), \\ F_3 &= [P(G)\equiv T(G)][F(G, H)\oplus P(G)]. \end{aligned} \quad (5.11)$$

Выражения в левых квадратных скобках обращаются в единицу в следующих случаях: в первой функции — когда выход элемента переходит из 0 в 1, во второй — при переходе из 1 в 0, в третьей — если выход не изменяется. В остальных случаях указанные выражения равны нулю, и проверочные функции становятся константами (и монотонными).

В соответствии с определением монотонности и введенным изо-переменным можно сформулировать критерии отсутствия состязаний по проверочным функциям.

*Функция  $F_1$  должна быть изотонной по всем изо-переменным при всех параметрах анализа.*

*Функция  $F_2$  должна быть антитонной по всем изо-переменным при всех параметрах анализа.*

*Функция  $F_3$  должна быть равна нулю.*

Смысл последнего критерия заключается в том, чтобы обнаружить возможный ложный импульс, когда начальные и конечные выражения совпадают.

Если какой-либо критерий нарушается, параметры анализа, при которых произошло нарушение, дадут необходимую диагностику анализа.

Рассмотрим пример триггера с унарным входом на рис. 5.6.

Установившиеся значения его сигналов в предыдущей фазе (спейсере) и текущей (рабочей) фазе таковы:

	спейсер	рабочая фаза
$A$	0	1
$X$	$D$	$D^*$
$NX$	$\wedge D$	$\wedge D^*$
$Q_1$	$\wedge M$	$\wedge D^*$
$Q_2$	$M$	$D^*$
$B$	1	0,

где  $D$  — значение унарного входа в предыдущей фазе;  $D^*$  — значение этого входа в текущей фазе;  $M$  — переменная памяти триггера в спейсере.

Рассмотрим сигнал  $B$  — выход индикаторного элемента. Для него первая и третья функции (5.11) равны нулю, а вторая имеет вид:

$$F_2 = \hat{h}_A \vee D \wedge M \wedge h_{NX} \wedge h_{Q1} \vee \hat{D}^*(D \wedge h_{NX} \vee M \wedge h_{Q1} \vee M \wedge h_{Q2}) \vee \vee D^*(\hat{M} h_{NX} \wedge h_{Q1} \vee \hat{M} h_{Q1} \wedge h_{Q2} \vee \hat{D} M \wedge h_{NX} h_{Q2} \vee \hat{D} \wedge M \wedge h_{NX} h_{Q1}). \quad (5.12)$$

Для отсутствия гонок приведенная функция должна быть анти-тонной по всем изо-переменным, от которых она зависит. Однако по ее выражению видно, что при некоторых параметрах анализа она может быть изотонной по переменным  $h_{NX}$ ,  $h_{Q1}$  и  $h_{Q2}$ , что нарушает приведенный выше критерий. Соответствующие этим переменным сигналы и будут непосредственно порождать гонки.

Определим, например, условия возникновения гонок на индикаторном элементе от сигнала  $Q_1$ . Для этого надо найти такое сочетание параметров и остальных изо-переменных, при котором функция (5.12) становится изотонной по этой переменной и зависит только от нее, т. е.  $F_2 = h_{Q1}$ . Искомые условия запишутся так:

$$\hat{D} D^* \wedge M \wedge h_{NX} h_A = 1. \quad (5.13)$$

Действительно, если подставить численные значения переменных, обращающих левую часть (5.13) в единицу, в правую часть выражения (5.12), то функция  $F_2$  станет изотонной.

Условие возникновения гонки, т. е. обращение левой части (5.13) в единицу, интерпретируется так. Вход  $X$  в фазе спейсера делает разрешенное изменение  $0 \rightarrow 1$  и остается единицей в рабочей фазе. Переменная памяти  $M = 0$ . Обращение изо-переменной  $h_{NX}$  в нуль в (5.13) означает, что сигнал  $NX$  еще не успел измениться в текущей фазе и остался равным единице. Сигнал  $A$ , напротив, должен измениться из 0 в 1. В этих условиях изменение сигнала  $Q_1$  из 1 в 0 порождает гонку. Именно такое развитие событий иллюстрируется на рис. 5.7.

Определение монотонности функции предполагает, что ее аргументы независимы друг от друга. Но изо-переменные — аргументы проверочных функций (5.11) — порождены входами анализируемого элемента, которые в общем случае не являются независимыми. Независимыми в любой схеме будут только изо-переменные, порожденные входными информационными сигналами схемы. Например, если в схеме на рис. 5.6 в рабочей фазе  $h_A = 0$ , т. е. вход  $A$  еще не изменился, то и  $h_{Q1} = h_{Q2} = 0$ . Вследствие этого не всег-

да удается обнаружить нарушение монотонности непосредственно по функциям (5.11) (хотя в примере это удалось). Поэтому при анализе необходимо учитывать зависимость между изо-переменными путем подстановок или другими способами. Для схем реальной сложности это делает процедуру проверки состязаний весьма трудоемкой и непригодной к «ручному» применению. (Впрочем, подобная ситуация характерна и для любых других методов обнаружения гонок.) Требуется использование машинных алгоритмов, подробности которых выходят за рамки книги.

### 5.6.6. Анализ других логических неисправностей

В разделе 5.6.3 отмечено, что для расчета индицируемости элемента необходимо имитировать его выходную КНЗ-01 тем, что выходное значение остается равным значению в предыдущей фазе («залипает»). Это обязательное условие самосинхронности.

Но ничто не мешает имитировать и другие неисправности тем же прямым методом, лишь бы неисправность выражалась логической формулой (в технической диагностике такие неисправности называют логическими [47]). Как исключение — к нелогическим неисправностям приводят отказы замыкания трасс, поскольку возникающие монтажные соединения не описываются двоичной булевой алгеброй.

Следует отметить, что в событийных методах отсутствуют возможности фиксации каких-либо других неисправностей, кроме КНЗ-01 для выходов элементов.

Имеющиеся в ФМА средства анализа прочих логических неисправностей позволяют расширить возможности проектирования надежных СС-схем. Это становится возможным на этапах как создания самих схем, так и создания проверяющих тестов, когда главной задачей тестов становится покрытие максимального числа неисправностей.

К наиболее значимым дополнительным логическим неисправностям СС-схем отнесем входные КНЗ элементов и *мутантные неисправности*.

Как одиночные, так и кратные входные КНЗ можно проверить, задавая входам элементов значения от предыдущей фазы (см. раздел 5.6.3.).

Важность входных КНЗ состоит в том, что они позволяют имитировать входные задержки элементов, в том числе после разветвлений. В последнем случае можно задать одному из входов после разветвления КНЗ, а другим входам — нет.

Проблема задержек после разветвлений (см. раздел 3.7) считается одной из главных в самосинхронике. Анализ этих задержек в ФМА может служить одним из инструментов решения проблемы.

*Мутантные отказы* (приводящие к мутантным неисправностям), в отличие от КНЗ, меняют значение сигнала в момент возникновения. Одним из примеров мутантной неисправности может быть ситуация, когда выход элемента, имеющий значение 1, «пробивается» на землю. Еще одна ситуация может возникнуть в паре ПФС-сигналов. При нормальной работе эта пара не может оказаться в состоянии равенства обоих значений, противоположных спейсеру. Но в результате пробоя такая ситуация может реализоваться. Диагностика мутантных неисправностей в событийном подходе рассмотрена в [3].

Процедура прямой имитации неисправностей в ФМА позволяет диагностировать практически любые константные логические неисправности.

## 5.7. Иерархический метод анализа (ИМА)

Как уже говорилось, методы анализа СС-схем, изложенные выше, сталкиваются с большими вычислительными трудностями, а начиная с числа элементов в несколько десятков эти трудности становятся практически непреодолимыми. Такое положение объясняется тем, что все методы предполагают необходимость так или иначе пройти и проверить как все возможные рабочие состояния схемы, так и переходы между ними.

Единственным выходом для создания схем практической сложности может быть их проектирование по частям (как это и делается в электронике).

Известны работы, нацеленные на построение одной полумодулярной схемы из нескольких других [3, 48]. Они относятся к формальному синтезу, основаны на событийных методах, очень громоздки и в основном ориентированы на управляющие схемы (не имеющие информационных сигналов). Методы так и не вышли за рамки теоретических исследований. В частности, они требуют событийных спецификаций работы схем, которые для большинства схем крайне объемны. Убедиться в сложности таких спецификаций можно, например, попытавшись создать событийное описание поведения хотя бы четырехразрядного умножителя. Важная для практики проблема полноты в этих методах даже не упоминается.

Необходимым и важным условием полноценного проектирования СС-схем является проведение анализа схем любого размера. В рамках событийных методов найден только один возможный путь такого анализа — в работе [4]. Здесь предлагается использовать ДИ, ранее построенные для блоков нижнего уровня, во взаимодействии с ДИ анализируемой схемы. Этот путь представляет собой лишь набор теорем с доказательствами и не получил практического развития. Например, каждая ДИ соответствует одному начальному состоянию. Следовательно, каждый блок нижнего уровня должен описываться множеством ДИ, число которых экспоненциально зависит от числа инфовходов и переменных памяти блока. Позволит ли предлагаемый путь избежать экспоненциальной сложности вычислений при практическом анализе, остается неясным.

Функциональный подход позволяет естественно подойти к иерархическому анализу: схемы, как правило, создаются иерархически, снизу вверх, на этапе проектирования они разомкнуты и имеют понятный интерфейс.

Существо функционального ИМА состоит в том, что анализируемая схема, состоящая из нескольких фрагментов, не раскрывается до элементов (уравнений), как это требуется для методов, изложенных выше, а проверяются только внешние описания (интерфейсы) фрагментов схемы и соединения фрагментов. Вычислительная сложность ИМА линейно зависит от количества фрагментов, что позволяет анализировать снизу вверх схемы любого размера.

Функциональный метод иерархического анализа впервые представлен в [49]. Отметим, что других публикаций по иерархическим методам (не подходам) анализа СС-схем, пригодным для практики, не найдено.

Впервые в самосинхронике указанный метод позволяет решить главную проблему проектирования СС-схем, во многом тормозящую их развитие, — анализ схем любого размера.

### 5.7.1. Задачи ИМА

Исходной для анализа будет разомкнутая схема, имеющая типовой интерфейс (см. раздел 2.5.1). Схема должна состоять из фрагментов, заранее прошедших через ФМА (СС-фрагментов). Исключения делаются для инверторов и повторителей, для которых не требуется специальных описаний.

Как и при анализе функциональным методом (см. раздел 5.6), в интерфейсе схемы верхнего уровня должно быть указано следующее:

- 1) группы входных и выходных сигналов и их СС-тип согласно типовому интерфейсу;
- 2) для входных фазовых сигналов — значение спейсеров, 0 или 1.

Схема представляется в виде списка имен фрагментов и соединений между ними, как это делается в языках описания аппаратуры (типа VHDL).

Кроме того, отдельно должны быть представлены описания фрагментов. В описаниях каждого фрагмента должны присутствовать следующие параметры интерфейса:

- 1) группы входных и выходных сигналов и их СС-тип согласно типовому интерфейсу;
- 2) для входных и выходных фазовых сигналов — значения спейсеров;
- 3) для выходных фазовых сигналов — списки индицируемых ими входов и выходов фрагмента;
- 4) для входных и выходных БС-сигналов — транзитная фаза и информация о блокирующих фазовых сигналах (подробнее в разделе 5.7.4.3).

Как и любая разомкнутая СС-схема, анализируемая схема должна быть проверена на индицируемость и отсутствие состязаний (в ИМА это делается косвенно). Кроме того, как и в ФМА, должна быть получена интерфейсная информация для следующего верхнего уровня иерархии. С учетом сказанного определяются следующие задачи, которые необходимо решать в ИМА:

- 1) проверка правильности соединений фрагментов;
- 2) проверка индицируемости сигналов;
- 3) проверка соблюдения дисциплины БС-сигналов;
- 4) получение параметров интерфейса главной схемы.

### 5.7.2. Проверка правильности соединений фрагментов

В данной задаче проверяются соединения фрагментов с точки зрения самосинхронности. И-сигналы соединяются с У-сигналами, БС — с БС-сигналами, ПФС — с ПФС-сигналами. Соединяемые фазовые сигналы должны иметь одинаковые значения спейсеров.

Необходимо проверять отсутствие разбаланса фаз на входах каждого фрагмента, т. е. следить, чтобы значения фазовых сигналов, подключенных к одному фрагменту, соответствовали одной и той же фазе.

Транзитные фазы соединяемых БС-сигналов должны быть согласованы со значениями спейсеров индицирующих их фазовых сигналов.

Также необходимо следить, чтобы БС-выходы одних фрагментов соединялись с БС-входами других без каких-либо промежуточных элементов, так как уже установлено, что в противном случае это приводит к нарушению самосинхронности.

### 5.7.3. Проверка индицируемости сигналов

На этом этапе определяется, какие сигналы главной схемы индицируются на ее фазовых выходах. На индикацию проверяются все сигналы, внешние и внутренние, кроме вспомогательных. Носителями информации об индикации выступают фазовые сигналы (см. разделы 2.5.2 и 2.5.3). При этом существенно используется свойство транзитивности.

Каждому фазовому сигналу сопоставляется список сигналов, индицируемых этим сигналом. Изначально и сам сигнал записывается в его список.

Проверка производится от входов схемы к выходам. По ходу проверки каждый фрагмент получает на свои фазовые входы списки индицирования. По полученным спискам и параметрам интерфейса (спискам индицируемых фрагментом своих входов и выходов) путем объединения формируются списки выходных фазовых сигналов фрагмента. Процесс заканчивается на фазовых выходах схемы.

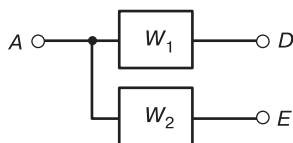
Обязательным для успешной проверки является присутствие всех внутренних сигналов схемы хотя бы в одном из выходных списков индицирования. Присутствие внешних фазовых сигналов не обязательно, так как они могут быть индицированы в схемах верхнего уровня (частичная индицируемость рассматриваемой схемы).

### 5.7.4. Проверка соблюдения дисциплины БС-сигналов

Данный этап предназначен для обнаружения состязаний в главной схеме. При условии успешного прохождения предыдущих этапов анализа источниками состязаний могут быть нарушения дисциплины входов фрагментов, т. е. несогласованность переключений БС-сигналов фрагментов и связанных с ними фазовых сигналов.

Предварительно введем понятие очередности изменений сигналов. В самосинхронике, из-за произвольности задержек, вместо понятия времени имеет смысл понятие очередности.

На рисунке 5.10  $W_1$  и  $W_2$  — цепочки любых элементов, имеющие не менее одного элемента.



**Рис. 5.10.** К понятию очередности переключений

По условиям теории, задержки элементов хотя и произвольны, но не равны нулю. Поэтому справедливо утверждение, что изменения сигналов  $D$  и  $E$  всегда будут происходить *после* изменения сигнала  $A$ .

Назовем сигнал  $A$  *инициатором* для сигналов  $D$  и  $E$ , а сигналы  $D$  и  $E$  — *континуаторами* (продолжателями) сигнала  $A$ . Таким образом, изменение инициатора всегда будет предшествовать изменению любого его континуатора.

Очевидно и свойство транзитивности: континуатор континуатора будет континуатором их общего инициатора.

Будем называть также *конкурентными* сигналы, моменты изменений которых по отношению друг к другу произвольны. Конкурентными могут быть как сигналы, имеющие общий инициатор, например  $D$  и  $E$ , так и сигналы с разными инициаторами (независимые).

Отметим, что индикаторные сигналы всегда являются континуаторами тех сигналов, которые они индицируют.

#### 5.7.4.1. Правила дисциплины при соединении БС-сигналов

Рассмотрим схематично соединение БС-сигналов двух фрагментов. В принимающем фрагменте БС-сигналы могут поступать либо на БСЯ, либо на другие схемы, например преобразователь БС-ПФС. Схему, принимающую БС-сигнал, будем называть далее *приемником*.

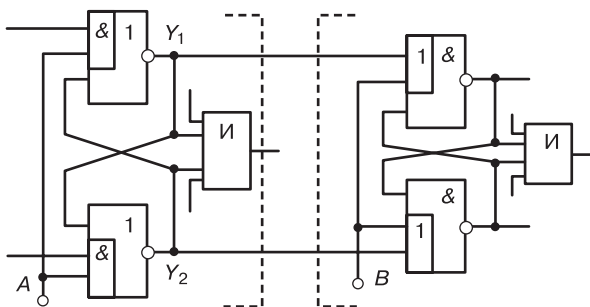
На рисунке 5.11 показана выходная БСЯ одного фрагмента и входная БСЯ (приемник) другого.

В соответствии с правилами, изложенными в разделе 5.2, основное правило дисциплины БС-сигналов при соединении фрагментов состоит в следующем.

*На период времени, когда выходные БС-сигналы ( $Y_1$ ,  $Y_2$ ) могут меняться, сигнал  $B$  должен запретить изменение сигналов приемника (заблокировать его входы).*

Рассмотрим последовательность изменений сигналов в соединении, показанном на рисунке. Все дальнейшие выводы справедливы для любого приемника БС-сигналов.





**Рис. 5.11.** Соединение БС-сигналов СС-фрагментов:

$Y_1, Y_2$  — БС-сигналы;  $A$  — разрешающий сигнал выхода;  $B$  — разрешающий сигнал входа; И — индикаторы; пунктир — границы фрагментов

Выходные БС-сигналы меняются в одной из фаз — транзитной. В этой фазе сигнал  $B$  блокирует подключенные БС-входы. В следующей, нетранзитной фазе сигнал  $B$  открывает БС-входы, и фаза заканчивается, когда переходные процессы во входной БСЯ завершились, а сигнал  $B$  еще остается в разрешающем состоянии.

Как видно, в нетранзитной фазе дисциплина соблюдается, и состязания не возникают.

В наступающей далее транзитной фазе изменяются и сигнал  $B$ , и выходные БС-сигналы. Чтобы не возникли состязания, должен соблюдаться порядок этих изменений, и предыдущее правило можно теперь сформулировать так.

*В транзитной фазе сначала должен измениться сигнал  $B$ , заблокировав входы, и лишь затем могут меняться выходные БС-сигналы.*

Данное правило уже можно приводить к схемотехническим терминам.

На рисунке 5.11 БС-сигналы ( $Y_1, Y_2$ ) являются континуаторами сигнала  $A$ , а выход индикатора, в свою очередь, есть континуатор этих БС-сигналов.

Поскольку в нетранзитной фазе отношение очередности рассматриваемых сигналов отсутствует, правило дисциплины можно распространить на обе фазы. Окончательно оно формулируется так.

*В соединении БС-сигналов фрагментов выходные сигналы ( $Y_1, Y_2$ ) и блокирующий сигнал  $B$  не могут быть конкурентными. Сигнал  $B$  должен быть инициатором сигналов ( $Y_1, Y_2$ ), а те, в свою очередь, должны быть континуаторами сигнала  $B$ .*

Это правило дисциплины будет основным при проверке возможных состязаний сигналов в ИМА. Для краткости будем называть его *БС-правилом*.

Отметим также, что разрешающие и блокирующие сигналы не могут быть никакими иными, кроме фазовых.

#### 5.7.4.2. Типы управления соединенными ячейками

Рассмотрим возможные типы управления соединенными ячейками в реальных схемах. Основных (некомбинированных) вариантов здесь всего четыре.

##### А. Единый управляющий сигнал.

Это наиболее простой тип управления (рис. 5.12).

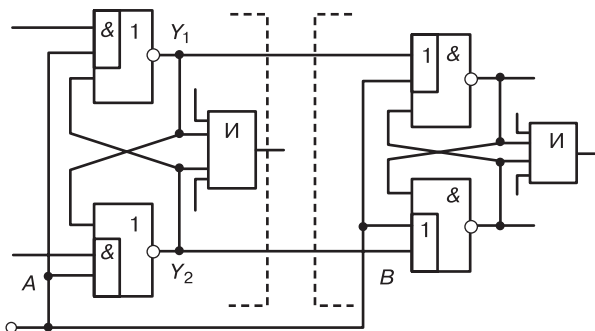


Рис. 5.12. Корректное управление с помощью единого сигнала

Так как сигналы  $A$  и  $B$  совпадают, выходы ( $Y_1$ ,  $Y_2$ ) являются континуаторами сигнала  $B$ , и БС-правило соблюдается.

Именно такое управление используется между ячейками внутри двухтактных триггеров (примеры на рис. 3.13 и 4.19). Пример внешнего управления данного типа можно видеть на рис. 4.21.

##### Б. Управление «с левой задержкой».

Такой вариант управления показан на рис. 5.13, где  $W$  — цепочка элементов. Обычно такими цепочками бывают повторители или инверторы.

В этом случае выходы ( $Y_1$ ,  $Y_2$ ) также остаются континуаторами сигнала  $B$ , и потому управление корректно.

##### В. Управление «с правой задержкой».

Этот тип управления показан на рис. 5.14.

При таком способе сигналы ( $Y_1$ ,  $Y_2$ ) и  $B$  являются конкурентными с общим инициатором  $A$ , и БС-правило нарушается.

##### Г. Управление с обратной связью.

Во многих случаях на практике требуется усиление в цепи сигнала  $B$ , что неизбежно вносит в цепь задержку. Выполнение БС-

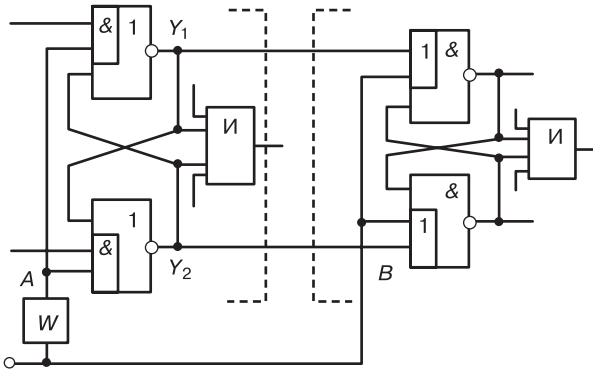


Рис. 5.13. Управление «с левой задержкой» (корректное)

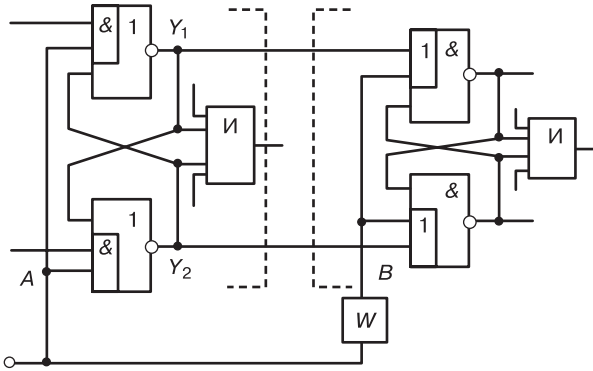


Рис. 5.14. Управление «с правой задержкой» (некорректное)

правила в подобных случаях можно обеспечить за счет обратной связи от фрагмента-приемника БС-сигнала к фрагменту-источнику. Обратная связь может быть локальной или глобальной.

Способ управления с локальной обратной связью показан на рис. 5.15. Обратная связь заводится с сигнала *B* (показана стрелками) либо непосредственно, либо через посредство других элементов (показано пунктиром). В последнем случае должна обеспечиваться индицируемость сигнала *B* на сигнале *A*.

Пример использования управления такого типа можно видеть на рис. 4.22 и 4.23.

Применение глобальной обратной связи (которая реализует общее замыкание главной схемы) иллюстрируется на рис. 5.16. Контур обратной связи с произвольной задержкой показан пунктиром в нижней части схемы.

По принципу самосинхронности сигнал общей обратной связи индицирует все внутренние сигналы схемы, в том числе сигнал  $B$ . По тому же принципу все внутренние фазовые сигналы схемы должны индировать этот общий сигнал, приходящий на вход главной схемы. Таким образом, все фазовые сигналы через общую обратную связь являются континуаторами сигнала  $B$ , в том числе и сигнал  $A$ . Из этого следует, что и сигналы  $(Y_1, Y_2)$  являются континуаторами сигнала  $B$ , и второе условие БС-правила соблюдается.

По первому условию БС-правила сигнал  $B$  не должен быть конкурентным с сигналами  $(Y_1, Y_2)$ . С учетом указанных выше свойств индирования через общую обратную связь есть только одна воз-

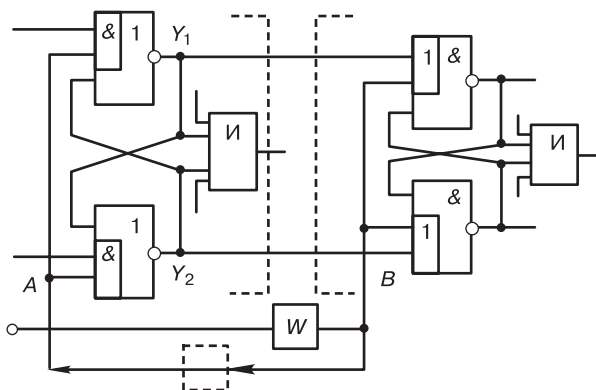


Рис. 5.15. Управление с помощью локальной обратной связи

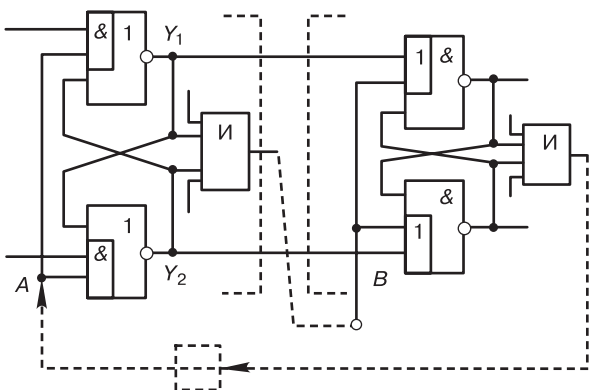


Рис. 5.16. Управление с помощью общей обратной связи

возможность выполнить это условие — сделать сигнал  $B$  континуатором сигналов  $(Y_1, Y_2)$ . Любое другое подключение  $B$  приводит к его конкурентности с сигналами  $(Y_1, Y_2)$ .

На рисунке 5.16 пунктиром показан пример корректного подключения сигнала  $B$ .

Кроме перечисленных основных вариантов возможны их комбинации, а также соединения с несколькими управляющими сигналами в БСЯ (например, это могут быть ПФС-сигналы). В последнем случае блокировка и разрешение производятся аналогично несколькими значениями этих управляющих сигналов.

### 5.7.4.3. Проверка соединений

Из изложенного выше видно, что в описании интерфейсов СС-фрагментов должна содержаться информация о выходных БСЯ и их входных приемниках: какими входами фрагментов и какими значениями они блокируются и существуют ли задержки в цепях блокировки. Эта информация готовится на более низких уровнях иерархии с помощью ИМА, а на нижнем уровне — ФМА.

Проверка дисциплины БС-сигналов заключается в установлении варианта соединения и правильности блокировки соответствующими значениями фазовых сигналов, а также проверке индикации разрешающих сигналов по ранее полученным спискам индикации.

### 5.7.5. Получение параметров интерфейса главной схемы

На этом шаге готовится информация для следующего верхнего уровня иерархии об индикации всех внешних сигналов и параметрах соединений внешних БС-сигналов схемы.

Из полученных ранее списков индикации фазовых выходов выбираются внешние сигналы главной схемы, которые и составляют внешние списки индикации.

Внешние БС-сигналы, в силу специфики СС-схем, соединяются с БС-входами внутренних СС-фрагментов непосредственно, без промежуточных элементов. Поэтому параметры внешних БС-сигналов и сигналов, их блокирующих, определяются по описаниям интерфейсов подсоединенных СС-фрагментов.

Полученная информация записывается в соответствующее описание интерфейса главной схемы.

Данный шаг завершает анализ самосинхронности на текущем уровне иерархии.

## Глава 6

# АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ СС-СХЕМ

---

В предыдущих главах рассмотрены теоретические и практические аспекты поведения и проектирования СС-схем. Уникальные свойства СС-схем обусловлены спецификой их построения. Главной особенностью создания СС-схем является необходимость отслеживания и учета каждого состояния, которое проходит схема в реальной работе. «Вручную» это удастся лишь для схем, состоящих не более чем из 4-5 базовых элементов.

Очевидно, что без привлечения специальных программных средств проектирование СС-схем невозможно. Поэтому с самого начала разработки таких схем для них создавались и программные системы.

Первые программные средства для СС-схем были разработаны в России группой В. И. Варшавского для ИПИРАН на рубеже 1980-х — 90-х годов (система ФОРСАЖ). В дальнейшем эта группа (уже за рубежом) выпустила обновленные версии этих программ (1993—1994 гг.). Известно также, что специалисты группы создали некоторые программы для исходных описаний в форме сетей Петри. Эти программы носят исследовательский характер и отсутствуют в открытом доступе.

В последующие годы и по настоящее время специальные программные средства для СС-схем, как следует из доступных источников, разрабатываются только в ИПИРАН.

Что касается зарубежных работ, единственная компания, рассматривающая СС-схемы и упоминавшаяся выше (Предисловие и раздел 3.1.2), Theseus Research Inc. [5], продвигает только предложенный ею подход NULL Convention Logic. Однако, как видно из ее сайта, она не занимается конкретными разработками схем и не предлагает никаких программ.

Ниже приводятся краткие сведения об известных программах проектирования СС-схем.

## 6.1. Система ФОРСАЖ (группа В. И. Варшавского)

В конце 80-х и начале 90-х годов группой Варшавского была впервые разработана специализированная система для СС-схем ФОРСАЖ (FORCAGE) (первая версия — по заказу ИПИРАН) [50, 4].

Хотя упоминания о системе существуют в Интернете (в англоязычных названиях) и в настоящее время, система не поддерживается и в Интернете недоступна (ссылочные сайты отсутствуют). Версия FORCAGE-3.0 существует как приложение в книге [4], сама книга, однако, труднодоступна.

Система состоит из следующих основных подсистем.

**ТРАНАЛ (TRANAL)** — анализ полумодулярности по методу диаграмм переходов (ДП).

Программа принимает описание замкнутой схемы с одним начальным состоянием. Язык описания — собственный (расширение файлов — ‘.cir’ — от слова circuit). Программа позволяет анализировать описания, ограниченные по количеству уравнений (т. е. элементов) из-за экспоненциальной зависимости времени счета от их числа.

**ТРАСПЕК (TRASPEC)** — анализ дистрибутивности (частный случай полумодулярности) по методу диаграмм изменений (ДИ).

Исходные данные программы — те же, что и у ТРАНАЛ. Время счета программы оценивается как полиномиальное от числа уравнений.

Программа способна анализировать довольно узкий подкласс полумодулярных схем — дистрибутивных. Со стороны внешнего описания дистрибутивные схемы — это схемы, в уравнениях которых есть операции И и нет операций ИЛИ. Для практики это, конечно, совершенно недостаточно.

**ТРАСИН (TRASYN)** — синтез по диаграммам изменений (ДИ).

Как отмечалось ранее в книге, данная программа скорее относится к типу конвертора из одной формы описания (ДИ) в другую (схемное описание). Все требования корректности исходных данных (по существу — требования полумодулярности) должен заранее выполнить пользователь. Программа делает проверку этих требований и отбраковывает некорректные описания.

Все упомянутые программы принимают только описания замкнутых схем с одним начальным состоянием. Вопрос полноты анализа не упоминается.

В системе используются собственные форматы описаний исходных данных, не совместимые с промышленными стандартами.

Большим неудобством характеризуется диагностика ошибок, которая в большинстве случаев малоинформативна.

В целом систему ФОРСАЖ можно охарактеризовать как систему для исследовательских, а не промышленных целей. В настоящее время система не развивается.

## 6.2. Отдельные программы анализа (ИПИРАН)

Научная группа в составе ИПИРАН, имевшая тесные контакты с группой В. И. Варшавского, с 90-х годов продолжила создание программных средств, уделяя большое внимание практической стороне проектирования СС-схем.

Разработаны и поддерживаются несколько действующих программ.

**БТРАН** — анализ полумодулярности по методу ДП.

Программа создана на основе программы ТРАНАЛ и была предназначена для исправления ошибок и улучшения потребительских качеств исходной программы.

**АСИАН** — анализ полумодулярности по методу ДП [51, 52].

Это модернизация и дальнейшее развитие программ ТРАНАЛ и БТРАН. Заново разработана организация памяти, введена визуальная диагностика нарушений. Резко увеличен объем анализируемых схем по сравнению с предыдущими программами.

Программа принимает описания замкнутых схем с одним начальным состоянием на языке системы ФОРСАЖ. Полнота анализа не рассматривается.

**АСПЕКТ** — анализ полумодулярности по методу ДИ [53].

Программа имеет существенно большие возможности, чем предшествующий аналог ТРАСПЕК. Анализируется дистрибутивность и полумодулярность. Значительно увеличены производительность анализа и объем анализируемых схем по сравнению с программами по методу ДП.

Программа также принимает описания замкнутых схем с одним начальным состоянием на языке системы ФОРСАЖ. Полнота анализа не учитывается.

**САМАН** — анализ полумодулярности по методу ДП [54].

Программа может использоваться для анализа небольших схем типа библиотечных элементов, а также для начального знакомства с СС-схемами и учебных целей.

В отличие от предыдущих программ ее исходные данные — описание разомкнутой схемы с использованием «обычного» языка алгебры



логики. Замыкание схемы для анализа осуществляется самой программой. Новая версия программы позволяет анализировать схемы различными способами: с одним начальным состоянием и константным замыканием, полным перебором входных констант и начальных состояний (без полноты), а также с обеспечением исчерпывающей полноты.

По планам группы программа САМАН будет подготовлена для открытого использования и в 2013 году представлена на сайте <http://www.samosinhron.ru> (дочернем сайте ИПИРАН).

### **6.3. САПР СС-схем промышленного назначения РОНИС (ИПИРАН)**

Система РОНИС [55] создана и развивается как дополнение к САПР общего назначения в сквозном маршруте проектирования микросхем.

САПР РОНИС предназначена для функционального проектирования СС-схем и выполнения специфических операций для схем этого класса, не поддерживаемых САПР электронных схем общего назначения. Предполагается, что при сквозном проектировании СС-схем будут использоваться известные прикладные системы: ввода схем (например, OrCAD), логического моделирования (например, Cadence LDV, Modelsim) и др.

Язык описания схем в системе РОНИС — VHDL (версия 1993 г.), поддерживаемый всеми разработчиками промышленных САПР.

В настоящее время в системе РОНИС имеются следующие компоненты.

#### **ОПИСАНИЕ БИБЛИОТЕКИ БАЗОВЫХ ЭЛЕМЕНТОВ СС-СХЕМ.**

Библиотека элементов для СС-схем [56] разработана ИПИРАН в сотрудничестве с НПК «Технологический Центр» МИЭТ (г. Зеленоград) для базовых матричных кристаллов (БМК) центра. Текущая версия библиотеки содержит около 150 элементов. В составе библиотеки есть как СС-элементы, так и элементы общего назначения, используемые в составных библиотечных СС-элементах.

Описание библиотеки на языке VHDL [57, 58] включено в систему РОНИС.

Пример описания СС-элемента библиотеки приведен на рис. 6.1.

СИНТАБИБ — синтез простых комбинационных схем на заданной библиотеке элементов [28, 59].

```

Library IEEE; use IEEE. STD_LOGIC_1164. all;
Library RONIS; Use RONIS. RonisPack. all; Use WORK. SAMOS_55APack. all;
Entity R011 is
    generic (Load_Q, Load_QB, Load_I: real:= Load_typ);
    port (S, R: in ron_logic; Q, QB, I: out ron_logic);
attribute Pin_load of I: signal is Load_Fact;
attribute Pin_load of S, R: signal is Load_Fact * 2.0;
attribute Pin_load of Q, QB: signal is Load_Fact * 3.0;
attribute SSS_main of S, R: signal is ParaPhase;
attribute SSS_main of Q, QB: signal is BiStable;
attribute SSS_main of I: signal is UnarPhase;
attribute SPACER of I: signal is '0';
attribute SPACER of S: signal is '0';
attribute DetailInd of R011: entity is "I<S>S R:I<W>Q QB S R";
attribute TranBlock of R011: entity is "Q<W>";
end R011;

Architecture Complex of R011 is
    attribute Ventil of Complex: architecture is 7;
    attribute BaseElem of Complex: architecture is true;
    signal U, UB, V, VB, QU, QBU: ron_logic;
begin
DD1: NOR2 generic map (2.0* Load_Fact) port map (R, U, UB);
DD2: NOR2 generic map (2.0* Load_Fact) port map (UB, S, U);
DD3: INV generic map (2.0* Load_Fact+ Load_QB) port map (UB, QBU);
DD4: INV generic map (2.0* Load_Fact+ Load_Q) port map (U, QU);
DD5: NAN2 generic map (Load_Fact) port map (R, QBU, VB);
DD6: NAN2 generic map (Load_Fact) port map (S, QU, V);
DD7: O22AI generic map (Load_I) port map (QBU, V, VB, QU, I);
Q <= QU;
QB <= QBU;
-----
Assert (S and R) /= '1'
report "Self-timed violation (anti-spacer S=R=1): " & R011'path_name
severity Warning;
end Complex;

```

**Рис. 6.1.** Один из библиотечных элементов системы РОНИС

В отличие от программы ТРАСИН здесь задается разомкнутая система уравнений в моно-функциях, т. е. в исходном, не самосинхронном, представлении на языке VHDL.

Программа использует табличный метод, описанный в данной книге. Схема может быть оптимизирована по одному из критериев — быстродействию или числу транзисторов. Результат выдается на языке VHDL. Программа использует элементы перебора, и время счета может существенно зависеть от числа уравнений.

ФАЗАН — анализ самосинхронности функциональным методом [44, 60].

Программа осуществляет анализ разомкнутых схем нижнего уровня, т. е. с полным раскрытием описаний до уравнений сигналов.

Исходные данные программы — множество файлов описаний анализируемой схемы, ее составных частей, компонентов, пакетов и библиотек на языке VHDL.

Программа обеспечивает исчерпывающую полноту анализа. Размер анализируемой схемы ограничен из-за экспоненциальной зависимости вычислительных затрат при проверке полноты.

При обнаружении нарушений самосинхронности выдается подробная диагностика с указанием места, сигналов и условий нарушений.

Программа также вычисляет данные, необходимые для иерархического анализа на верхнем уровне и (опционально) записывает их в интерфейс анализируемой схемы.

ЛИМАН (на стадии разработки) — иерархический анализ самосинхронности схем функциональным методом [49].

Программа предназначена для анализа самосинхронности схем любого размера. Анализ проводится на основе информации, записанной в интерфейсах составляющих схему блоков, полученной на предыдущем уровне иерархии, и взаимосвязей этих блоков.

# ПОСЛЕСЛОВИЕ

---

Самосинхронные схемы обладают уникальным набором свойств, что обосновано теоретически и проверено экспериментально. Однако они не получили такого признания и распространения, которого заслуживают.

Конечно, уникальные свойства не даются даром. Самосинхронные схемы требуют увеличения затрат по числу транзисторов, что сейчас считается недостатком. Однако опыт показывает, что при грамотном проектировании СС-схем можно добиться уровня превышения затрат только в 1,5—2,5 раза больше синхронных аналогов. Попытка повторить структуру синхронной схемы при реализации самосинхронного аналога обычно приводит к неоптимальной СС-схеме (которую можно улучшить). Проектирование СС-схем имеет свои особенности и часто диктует иные способы организации вычислений. Например, «хорошо спроектированный» самосинхронный умножитель будет иметь другую внутреннюю структуру, чем его синхронный функциональный аналог.

Вообще, бывший в широком ходу ранее и сохраняющийся до сих пор тезис «самосинхронные схемы требуют больших затрат по сравнению с синхронными» несостоятелен. Эти схемы, синхронные и самосинхронные, — просто разные, и так прямолинейно сравнивать их некорректно. Возьмем простейший пример для обеих схем — реализацию функции И-НЕ. Синхронная схема в данном случае — это единственный элемент, самосинхронная состоит из нескольких элементов. Затраты самосинхронной действительно больше затрат синхронной. Но синхронная схема реализует одну задачу для использования в окружающей схеме: выполняемую функцию, а самосинхронная — две: выполняемую функцию и обеспечение индикации окончания переходного процесса.

Такова же ситуация и в другом классе простых схем — триггерах. Простейшие триггеры — бистабильная ячейка или асинхронный двухтактный триггер — имеют минимальные затраты, так как выполняют одну функцию: запоминание значения. Но в синхронных схе-

мах широко применяются более сложные триггеры — с различными блокировками, обратными связями и т. п., призванные решать и дополнительные задачи, например повышение помехозащищенности. Затраты таких триггеров, имеющих уже две функции, сравнимы с затратами самосинхронных триггеров, имеющих также две функции, что можно проверить по приведенным в книге схемам.

Точно так же некорректно сравнивать синхронные и самосинхронные схемы по быстродействию. К тому же практика показывает, что во многих случаях удается создать самосинхронные схемы со сравнимым и даже лучшим быстродействием, чем синхронные, выполняющие одинаковые логические преобразования. В то же время есть и проблемы, связанные с быстродействием. Они возникают в многоуровневых устройствах, когда требуется одновременная индикация многих сигналов и в нескольких местах. Каскадные схемы, приведенные в книге, задачу решают, но общее быстродействие устройства нежелательно снижается. Это одна из проблем самосинхронности, требующая исследования и разработки методов преодоления.

Традиционно считается, что самосинхронные схемы трудны для понимания. Действительно, теория, да и методы их реализации созданы специалистами по прикладной математике. Известная литература слишком математизирована, что, очевидно, тормозит освоение этих схем «рядовыми» разработчиками. Налицо, конечно, явный дефицит (если не сказать — полное отсутствие) литературы по самосинхронной проблематике в целом, более понятной специалистам по электронике. Автор книги стремился изложить материал с точки зрения двух понятий — индикации и состязаний. Индикация имеет вполне понятный физический смысл, а состязания — давно известная проблема в электронике. Есть надежда, что это изложение привлечет внимание разработчиков к самосинхронным схемам и подвигнет их для начала хотя бы поэкспериментировать с ними.

Для самосинхронных схем, не имеющих состязаний и отказобезопасных по широкому классу неисправностей, есть очевидные области применения: устройства повышенной надежности для производства, медицины и других отраслей, автономные системы вплоть до космических, устройства, работающие в нестабильных внешних условиях.

Но и для «обычных» применений путь открыт: плохо ли иметь вычислительное устройство, работающее без ошибок (сбоев)? В широчайшем диапазоне температур и напряжений? В случае батарейного питания — до предельного истощения источника? Ответ — конечно, хорошо. Затраты по транзисторам, да и быстродействие во множестве случаев не критичны.

С экономической точки зрения, повышенные затраты самосинхронных схем в транзисторах могут с успехом компенсироваться повышением выхода годных кристаллов (чипов) с пластин при производстве. Одним из видов контроля схем является проверка на динамических тестах. Главная причина отбраковки чипов на этих тестах — состязания. Для самосинхронных схем такой вид брака должен свестись практически к нулю.

В книге В. И. Варшавского с сотрудниками [3] высказаны соображения о скором развитии самосинхронной схемотехники. Эти соображения оказались слишком оптимистичными. Альтернативный синхронному путь, которому пока следует индустрия, — компромиссный, до точной самосинхронности не доходящий. Но уникальные свойства самосинхронных схем никуда не делись и объективно должны быть востребованы.

Именно по последней причине группа специалистов ИПИРАН сохраняет оптимизм и работает на развитие и распространение самосинхронных схем.

# ЛИТЕРАТУРА

---

1. *Muller D. E., Bartky W. S.* A Theory of Asynchronous Circuits // Proceedings of the International Symposium on the Theory of Switching. Part 1. Harvard University Press, 1959. P. 204—243.
2. Аперiodические автоматы/Под. ред. В. И. Варшавского. — М.: Наука, 1976. 424 с.
3. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах/Под. ред. В. И. Варшавского. — М.: Наука, 1986. 398 с.
4. *Kishinevsky M., Kondratyev A., Taubin A., Varshavsky V.* Concurrent Hardware: The theory and Practice of Self-Timed Design. — London: John Wiley and Sons, 1993. 388 p.
5. <http://www.theseusresearch.com/NCLPaper01.htm>.
6. ГОСТ Р 53480—2009. Надежность в технике. Термины и определения. — М.: Стандартинформ, 2010.
7. *Плеханов Л. П.* О свойстве самосинхронности цифровых электронных схем // Системы и средства информатики. Вып. 21. № 1. — М.: Торус Пресс, 2011. С. 84—91.
8. *Степченков Ю. А., Дьяченко Ю. Г., Петрухин В. С., Филин А. В.* Цена реализации уникальных свойств самосинхронных схем // Системы и средства информатики. Вып. 9. — М.: Наука, 1999. С. 261—292.
9. *Woods J. V., Day P., Furder S. B., Garside J. D., Paver N. C., Temple S.* AMULET1: an asynchronous ARM microprocessor // IEEE Transactions Computers. 1997, V. 46, № 4, April. P. 385—397.
10. *Плеханов Л. П., Степченков Ю. А.* Экспериментальная проверка некоторых свойств строго самосинхронных схем // Системы и средства информатики. Вып. 16. — М.: Наука, 2006. С. 476—485.
11. *Соколов И. А., Степченков Ю. А., Петрухин В. С., Дьяченко Ю. Г., Захаров В. Н.* Самосинхронная схемотехника — перспективный путь реализации аппаратуры // Системы высокой доступности. 2007, № 1—2, Т. 3. С. 61—72.
12. *Izosimov O. A., Shagurin I. I., Tsylyov V. V.* Physical Approach to CMOS Module Self-Timing // Electronic Letters. 1990, V. 26, № 22. P. 1835—1836.
13. *Seitz C. L.* «System Timing,» in Introduction to VLSI Systems. — ed. by Carver Mead and Lynn Conway, Reading, Mass., Addison-Wesley, 1980. P. 242—262.
14. *Цирлин Б. С.* Базисные реализации полумодулярных схем // Техническая кибернетика. 1983, № 5. С. 194.

15. Плеханов Л. П. Базовые элементы самосинхронных схем // Системы и средства информатики. Вып. 7. — М.: Наука, 1995. С. 258—264.
16. Плеханов Л. П. Базовые элементы самосинхронных схем КМДП-технологии // Системы и средства информатики. Вып. 11. — М.: Наука, 2001. С. 316—320.
17. Файзуллаев Б. Н., Кармазинский А. Н. Сверхскоростная элементная база на основе КМДП-схемотехники // Эл. выч. техн. Сб. статей, вып. 1. — М.: Радио и связь, 1987. С. 134—143.
18. Кармазинский А. Н. Методы синтеза и средства композиционного проектирования элементов быстродействующих цифровых КМДП микросхем: Диссертация на соискание ученой степени д. т. н.. — М.: МИФИ, 1990.
19. Плеханов Л. П. Проблемы функционального подхода в проектировании самосинхронных схем // Системы и средства информатики. Вып. 15. — М.: Наука, 2005. С. 329—337.
20. Berkel K. V., Kessels J., Ronchen M. and oth. The VLSI-Programming Language Tangram and its Translation into Handshake Circuits // Proc. of the EDAC-91. Los Alamitos, IEEE Comp. Ass. Press. 1991. P. 384—389.
21. Martin A. J. A Synthesis Method for Self-Timed VLSI Circuits // IEEE Intern. Conf. on Comp. Design: VLSI in Computers & Processors. 1987. P. 224—229.
22. Berkel K. V., Saeijs R. Compilation of Communicating Processes into Delay-Insensitive Circuits // Proc. of the IEEE Int. Conf. on Computer Design, IEEE. 1988. P. 157—162.
23. Brzozowski J. A., Ebergen J. C. Recent Developments in the Design of Asynchronous Circuits // Lecture Notes in Comp. Sci. 1989, V. 380. P. 78—94.
24. Варшавский В. И., Кишиневский М. А., Кондратьев А. Ю., Розенблюм Л. Я., Таубин А. Р. Модели для спецификации и анализа процессов в асинхронных схемах // Техническая кибернетика. 1988, № 2. С. 171—190.
25. Taubin A., Cortadella J., Lavagno L., Kondratyev A. and Peeters A. Design Automation of Real-Life Asynchronous Devices and Systems // Foundations and Trends in Electronic Design Automation. — Now Publishers Inc., 2007, V. 2, no 1. P 1—133.
26. Плеханов Л. П. Индикация в самосинхронных электронных схемах. Обоснование и оптимизация // Системы и средства информатики. Вып. 12. — М.: Наука, 2002. С. 290—297.
27. Плеханов Л. П. Логические уравнения в разработке цифровых микроэлектронных устройств // Электронная техника. Сер. 10. Микроэлектронные устройства. 1989, Вып. 5 (77). С. 25—28.
28. Плеханов Л. П. Синтез комбинационных самосинхронных электронных схем // Системы и средства информатики. Вып. 14. — М.: Наука, 2004. С. 292—304.
29. Шоломов Л. А. Основы теории дискретных логических и вычислительных устройств. — М.: Наука, 1980. 400 с.



30. *Chen G., Chen H. and oth.* On-Chip Cooper-Based vs. Optical Interconnects: Delay Uncertainty, Latency, Power and Bandwidth Density Comparative Predictions. — Rochester University, Cornell University, USA, 2006. (<http://www.csl.cornell.edu/~albonesi/research/papers/iitc06.pdf>)
31. [http://en.wikipedia.org/wiki/Transistor\\_count](http://en.wikipedia.org/wiki/Transistor_count).
32. *Степченков Ю. А., Денисов А. Н., Дьяченко Ю. Г., Гринфельд Ф. И., Филимоненко О. П.* Библиотека самосинхронных элементов для проектирования полужаказных микросхем серий 5503 и 5507. — М.: ИПИРАН, 2008. 237 с.
33. *Chen C.-J., Cheng W.-M., Tsai H.-Y., Wu J.-C.* A Quasi-Delay-Insensitive Microprocessor Core Implementation for Microcontrollers // *Journal of Information Science and Engineering*. 2009, 25. P. 543—557 ([http://www.iis.sinica.edu.tw/page/jise/=2009/=2009=03\\_13.pdf](http://www.iis.sinica.edu.tw/page/jise/=2009/=2009=03_13.pdf))
34. <http://en.wikipedia.org/wiki/C-element>.
35. *Плеханов Л. П., Степченков Ю. А., Денисов А. Н. и др.* Самосинхронный триггер для связи с удаленным приемником: Патент на изобретение 2382487 (РФ). Приоритет от 23.10.2008.
36. *Sokolov I. A., Stepchenkov Y. A., Dyachenko. Y. G.* Self-timed RS-trigger with the enhanced noise immunity International application number PCT/RU 2010/000279. International publication date 28.05. 2010.
37. *Соколов И. А., Степченков Ю. А., Дьяченко Ю. Г., Захаров В. Н.* Самосинхронный одноктактный D-триггер с высоким активным уровнем сигнала управления: Патент на изобретение 2362266 (РФ). Приоритет от 20.07.2009.
38. *Степченков Ю. А., Дьяченко Ю. Г., Рождественский Ю. В., Рождественскене А. В.* Разряд самосинхронного регистра сдвига: Патент на изобретение 2319232 (РФ). Приоритет от 10.03.2008.
39. *Булгаков С. С., Варшавский В. И., Лазуткин В. И. и др.* Асинхронный регистр сдвига: Авторское свидетельство СССР № 1138834. // Б. И. № 5, 1985.
40. *Степченков Ю. А., Дьяченко Ю. Г., Шнейдер А. Ю., Прокофьев А. А.* Формирователь парафазного сигнала с высоким активным уровнем сигнала управления: Заявка на выдачу патента РФ на изобретение. Рег. № 2011129014, 13.07.2011.
41. *Плеханов Л. П., Степченков Ю. А., Дьяченко Ю. Г.* Преобразователь унарного информационного сигнала в парафазный: Заявка на выдачу патента РФ на изобретение (на регистрации). 2012.
42. *Степченков Ю. А., Дьяченко Ю. Г., Плеханов Л. П., Гринфельд Ф. И., Степченков Д. Ю.* Самосинхронный двухтактный D-триггер с высоким активным уровнем сигнала управления: Патент на изобретение 2365031 (РФ) // Б. И. № 23, 2009.
43. *Плеханов Л. П.* Полнота анализа электронных схем на самосинхронность // *Системы и средства информатики*. Вып. 20, № 1. — М.: Торус Пресс, 2010. С. 48—58.

44. Плеханов Л. П. Анализ самосинхронности электронных схем функциональным методом // Системы и средства информатики. Вып. 18. — М.: Наука, 2008. С. 225—233.
45. Плеханов Л. П. Проектирование самосинхронных схем: функциональный подход/Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС-2010)». Сборник трудов. 2010, № 1. С. 26—31.
46. Малышенко Ю. В. Техническая диагностика: конспект лекций. Ч I. 6. Состязания сигналов в цифровых схемах // [http://abc.vvsu.ru/Books/Teh\\_diag/page0017.asp](http://abc.vvsu.ru/Books/Teh_diag/page0017.asp).
47. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики. — М.: Энергия, 1981. 320 с.
48. Карнов С. А. Метод и средства проектирования самосинхронных схем: Автореферат диссертации на соискание ученой степени кандидата технических наук. — СПб.: Государственный электротехнический университет, 1993. 16 с.
49. Плеханов Л. П. Иерархический метод анализа самосинхронных электронных схем // Системы и средства информатики. Т. 22. Вып. 1. — М.: Торус Пресс, 2012. С. 62—73.
50. Варшавский В. И., Карнов С. А., Кондратьев А. Ю. и др. Инструментальные средства автоматизации проектирования самосинхронных схем // Системы и средства информатики. Вып. 5. — М.: Наука, 1993. С. 196—214.
51. Рождественский Ю. В., Морозов Н. В., Степченков Ю. А., Рождественскене А. В. Универсальная подсистема анализа самосинхронных схем // Системы и средства информатики. Вып. 16. — М.: Наука, 2006. С. 463—475.
52. Морозов Н. В., Рождественский Ю. В., Степченков Ю. А., Рождественскене А. В. Средство анализа системы булевых уравнений на полумодулярность и дистрибутивность АСИАН // Свидетельство об официальной регистрации программы для ЭВМ № 2006613257. — 22.08.2007.
53. Рождественский Ю. В., Морозов Н. В., Рождественскене А. В. Подсистема событийного анализа самосинхронных схем АСПЕКТ/Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС-2010)». Сборник трудов. 2010, № 1. С. 26—31.
54. Плеханов Л. П. Программа анализа описаний цифровых схем на строгую самосинхронность САМАН. Версия 2 // Свидетельство об официальной регистрации программы для ЭВМ № 2008615059. — 21.10.2008.
55. Плеханов Л. П. САПР строго самосинхронных электронных схем РОНИС/Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС-2006)»: Сборник трудов. 2006. С. 155—157.
56. Степченков Ю. А., Денисов А. Н., Дьяченко Ю. Г., Гринфельд Ф. И., Филимоненко О. П., Фомин Ю. П. Библиотека элементов базовых матричных

- кристаллов для критических областей применения // Системы и средства информатики. Вып. 14. — М.: Наука, 2004. С. 318—361.
57. Плеханов Л. П., Степченко Ю. А., Дьяченко Ю. Г. Библиотека самосинхронных элементов на базе 55 серии БМК САМОС\_55 // Свидетельство об официальной регистрации программы для ЭВМ № 2007613547. — 22.08.2007.
58. Плеханов Л. П., Степченко Ю. А., Дьяченко Ю. Г. Библиотека самосинхронных элементов семейств БМК 5503 и 5507 САМОС\_55А // Свидетельство об официальной регистрации программы для ЭВМ № 2007615060. — 21.10.2008.
59. Плеханов Л. П. Программа синтеза комбинационных самосинхронных схем на заданной библиотеке элементов СИНТАБИБ // Свидетельство об официальной регистрации программы для ЭВМ № 2007613665. — 27.08.2007.
60. Плеханов Л. П. Программа анализа самосинхронных схем функциональным методом (ФАЗАН) // Свидетельство о государственной регистрации программы для ЭВМ № 2011611102. — 01.02.2011
61. Пухальский Г. И. Логическое проектирование цифровых устройств радиотехнических систем. — Л.: ЛГУ, 1976. 257 с.

# СПИСОК СОКРАЩЕНИЙ

---

БМК	— базовый матричный кристалл
БС	— бистабильный
БСЯ	— бистабильная ячейка
Г-триггер	— гистерезисный триггер
ДИ	— диаграмма изменений
ДНФ	— дизъюнктивная нормальная форма
ДП	— диаграмма переходов
ЕК	— единичный код
ЗПР	— затратный показатель редуктора
И	— индикаторный (сигнал, элемент)
ИЗ	— индекс задержки
ИМА	— иерархический метод анализа
КИ	— код с идентификатором
КМДП	— комплементарная металл-диэлектрик-полупроводник (структура транзистора)
КНЗ-01	— константная неисправность типа залипания на 0 или 1
КНФ	— конъюнктивная нормальная форма
МПФС	— моно-парафазная со спейсером (процедура)
М-функция	— моно-функция
НЗЭМ	— независимые от задержек элементов по Маллеру (схемы)
ОКЭ	— однокаскадный элемент
ОРК	— оптимальный равновесный код
ПМ	— полумодулярный (свойство)
ПП	— переменная памяти
ПФС	— парафазный со спейсером (код, сигнал)
СПС	— схема параллельного сжатия
СР	— схема редукции
СС	— самосинхронный (свойство)
ССБЭ	— самосинхронный базовый элемент
ССБЯ	— самосинхронная базовая ячейка
ССЗЯ	— самосинхронная запоминающая ячейка
ССК	— самосинхронизирующий код
СУЗ	— схема управления звеном (конвейера)
У	— управляющий (сигнал)
УС	— унарный стабильный (сигнал)
ФКЭ	— фиктивный класс эквивалентности (в диаграмме переходов)
ФМА	— функциональный метод анализа

# СЛОВАРЬ ТЕРМИНОВ

---

## **Асинхроника**

область исследования схем, не имеющих внешней (глобальной) синхронизации

## **Гонки — см. Состязания**

## **Диагностирование неисправности**

действия, проводимые с целью установления наличия неисправности, локализации неисправности и определения причин ее появления (ГОСТ Р 53480—2009)

## **Диаграмма изменений**

графическое изображение изменений сигналов — выходов элементов схемы

## **Диаграмма переходов**

графическое изображение изменений вектора состояний схемы — значений всех сигналов (выходов элементов)

## **Дисциплина сигналов**

определенный порядок следования сигналов

## **Залипание**

отказ, заключающийся в сохранении значения выхода элемента, когда он должен измениться из-за изменения входов

## **Замыкание константное**

согласованное замыкание разомкнутой схемы, при котором внешняя среда не меняет значений ее входов

## **Замыкание согласованное**

введение обратной связи для разомкнутой схемы, корректно имитирующей взаимодействие этой схемы с внешней средой

## **Зона эквихронная**

область на кристалле, внутри которой задержки трасс не превышают минимальной задержки элементов

## **Изо-переменная**

переменная, меняющаяся из 0 в 1

## **Импликанта**

конъюнктивный член ДНФ

**Индикация**

логический способ определения окончания переходного процесса

**Индицируемость сигнала**

свойство сигнала в схеме: сигнал А индицируется на сигнале В (или сигнал В индицирует сигнал А), если любое изменение сигнала А вызывает (в той же фазе) изменение сигнала В

**Индицируемость схемы**

свойство разомкнутой схемы, когда в каждой фазе каждый изменяющийся выход элемента индицируется хотя бы на одном из фазовых выходов схемы

**Индицируемость схемы полная**

свойство разомкнутой схемы, когда в каждой фазе на выходных фазовых сигналах индицируются как все внутренние сигналы, так и все фазовые входы схемы

**Индицируемость схемы частичная**

свойство разомкнутой схемы, когда в каждой фазе на выходных фазовых сигналах индицируются все внутренние сигналы, а индикация фазовых входов схемы не обязательна

**Инициатор**

независимый изменяющийся сигнал

**Конвейер неплотный**

конвейер, в установившемся режиме которого передаваемая информация располагается неплотно — через одно звено

**Конвейер плотный**

конвейер, в установившемся режиме которого передаваемая информация располагается плотно — в каждом звене

**Континуатор**

сигнал, определяемый в связи с инициатором: может меняться только после изменения его инициатора

**Моно-функция (М-функция)**

исходная, некодированная логическая функция, в дальнейшем подлежащая самосинхронному кодированию

**М-секция**

набор из одной или нескольких комбинационных (без запоминания) М-функций, имеющих общие аргументы, не зависящих друг от друга и подлежащих одновременному преобразованию в СС-секцию

**Неисправность**

состояние изделия, характеризующееся неспособностью выполнить требуемую функцию (ГОСТ Р 53480—2009)

**Неисправность логическая**

неисправность, которая может быть описана логической функцией или уравнением

**Неисправность мутантная**

неисправность, возникшая в результате мутантного отказа

**Отказ**

потеря способности изделия выполнить требуемую функцию — событие, приводящее к состоянию неисправности (ГОСТ Р 53480—2009)

**Отказ мутантный**

отказ, изменяющий значение сигнала на противоположное (в отличие от залипания)

**Отказобезопасность**

свойства изделия, ориентированные на сохранение безопасности в случае отказа (ГОСТ Р 53480—2009)

**Ошибка**

несоответствие между вычисленным, наблюдаемым или измеренным значением или состоянием и истинным, заданным или теоретически правильным значением или состоянием (ГОСТ Р 53480—2009)

**Параметры анализа**

набор постоянных величин, определяющих информационные входы и переменные памяти схемы в процессе анализа

**Переменная памяти**

переменная в описании схемы, значение которой произвольно; при нормальной работе это значение запоминается от предыдущей фазы

**Переход конфликтный**

переход в диаграмме переходов, в котором возбуждение элемента снимается из-за изменения его входов, а не после окончания переходного процесса (по Маллеру); означает наличие состязания

**Подход событийный**

совокупность методов исследования самосинхронных схем, основанных на представлении поведения в виде событий — изменений сигналов (выходов элементов)

**Подход функциональный**

совокупность методов исследования самосинхронных схем, основанных на анализе систем логических уравнений и функций

**Полнота анализа**

проверка в одном сеансе анализа всех основных состояний и всех реальных переходов между ними с контролем самосинхронности

**Полумодулярность**

свойство схемы, в диаграмме переходов которой отсутствуют конфликтные переходы (по Маллеру)

**Преобразование парафазное со спейсером (ПФС-преобразование)**

построение парафазных функций со спейсером (ПФС-функций) по заданным моно-функциям (М-функциям)

**Редуктор**

минимальная схема редукции

**Редукция**

схемотехническое понижение числа индикаторных сигналов с сохранением самосинхронности

**Самопроверка**

обнаружение ошибок непосредственно при выполнении требуемой функции (ГОСТ Р 53480—2009)

**Самосинхронизирующиеся коды**

способы кодирования информации для определения окончания переходных процессов чисто логическим путем

**Самосинхроника**

область исследования самосинхронных схем как часть асинхроники

**Сеанс анализа**

однократный расчет самосинхронности (или полумодулярности) всей схемы

**Сигнал оценочный**

виртуальный индикаторный сигнал, индицирующий всю схему

**Сигнал унарный**

одионый независимый сигнал

**Сигнал унарный стабильный**

одионый информационный сигнал (по аналогии с бистабильными сигналами)

**Сигналы информационные (инфосигналы)**

сигналы, несущие содержательную информацию: парафазные со спейсером (ПФС), бистабильные (БС), унарные стабильные (УС)

**Сигналы конкурентные**

сигналы, моменты изменений которых по отношению друг к другу произвольны

**Сигналы контрольные**

унарные фазовые сигналы, предназначенные для организации переходов фаз: индикаторные (И) или управляющие (У)



**Сигналы фазовые**

сигналы, значения которых определяется фазой работы: индикаторные (И), управляющие (У) или парафазные со спейсером (ПФС)

**Состояние (схемы)**

множество двоичных значений всех сигналов схемы — выходов ее элементов

**Состояние начальное**

состояние схемы в начале ее работы или анализа

**Состояние неравновесное**

неустойчивое состояние схемы, самопроизвольно переходящее в другое состояние

**Состояние равновесное**

устойчивое состояние схемы, не меняющееся без изменения внешних воздействий

**Состязания (сигналов на элементе)**

такие изменения входов элемента, которые могут вызвать ошибочное изменение его выхода, например ложный импульс

**Спейсер — см. Фаза спейсерная****СС-секция**

однокаскадная комбинационная самосинхронная схема (полученная из исходной М-секции)

**СС-секция смешанная**

СС-секция, на входе которой наряду с обязательными фазовыми сигналами присутствуют бистабильные (БС) сигналы

**Схема асинхронная**

схема, не имеющая тактовых генераторов и сигналов

**Схема полумодулярная**

схема, диаграмма переходов которой не имеет конфликтных переходов (по Маллеру)

**Схема самосинхронная**

схема, для которой при всех реальных состояниях и переходах выполняются два условия: отсутствие гонок при любых задержках элементов (независимость от задержек) и отказобезопасность (остановка при возникновении одиночных и кратных неисправностей типа залипаний на 0 или 1 выходов элементов)

**Схема синхронная**

схема, работа которой управляется импульсами тактового генератора

**Фаза рабочая**

одна из двух фаз работы СС-схем, в которой, как правило, происходит обработка информации

**Фаза спейсерная (спейсер)**

одна из двух фаз работы СС-схем — промежуточная, необходимая для обеспечения самосинхронности

**Фаза транзитная**

для бистабильных (БС) сигналов — одна из двух фаз, в которой эти сигналы могут изменяться

**Фактор состояния**

набор независимых значений информационных входов схемы и переменных памяти, однозначно определяющий состояние схемы

**Функция антитонная**

логическая функция, значение которой не увеличивается при изменении аргументов из 0 в 1

**Функция изотонная**

логическая функция, значение которой не уменьшается при изменении аргументов из 0 в 1

**Функция монотонная**

логическая функция, значение которой при изменении аргументов из 0 в 1 либо только не убывает (изотонная), либо только не возрастает (антитонная)

**Функция проверочная**

функция, специально построенная для выявления состязаний на элементе

**Элемент однокаскадный**

элемент, ток нагрузки которого переключается транзисторами, управляющие электроды которых непосредственно соединены со входами элемента

# Приложение

## Решение логических уравнений и систем

### П. 1. Одно уравнение

Рассмотрим одно логическое уравнение в форме ДНФ:

$$ay \vee b \wedge y = 0, \quad (\text{П1})$$

где  $y$  — неизвестная, которую надо определить;  $a, b$  — любые переменные.

Согласно [61], решением этого уравнения будет:

$$y = b \vee \wedge ah, \quad (\text{П2})$$

где  $h$  — произвольная переменная. Эта переменная может принимать значения 0 или 1 или быть любой функцией от переменных, от которых зависят  $a$  и  $b$ .

Решение (П2) существует только при условии

$$ab = 0, \quad (\text{П3})$$

так как в противном случае исходное уравнение (П1) будет противоречивым.

### П. 2. Система уравнений с одной неизвестной

Пусть имеется несколько уравнений, из которых надо определить переменную  $y$ :

$$a_i y \vee b_i \wedge y = 0, \quad i = 1, \dots, N. \quad (\text{П4})$$

Согласно [27], решение будет иметь вид:

$$y = \bigvee_{i=1}^N b_i \vee \bigwedge_{i=1}^N \wedge a_i h, \quad (\text{П5})$$

Условием существования решения будет либо одно условие

$$\bigvee_{i=1}^N (Va_i) \bigvee_{i=1}^N (Vb_i) = 0, \quad (\text{П6})$$

либо эквивалентная система из  $N^2$  условий:

$$a_i b_j = 0, \quad i, j = 1, \dots, N. \quad (\text{П7})$$

Если левая часть любого из уравнений (П6) или (П7) обращается в единицу, система не имеет решения.

### П. 3. Система уравнений с несколькими неизвестными

Все переменные, входящие в систему, следует разбить на три группы [27]:

*неизвестные* — переменные, которые требуется найти;

*аргументы* — независимые переменные, от которых будут зависеть неизвестные;

*промежуточные* — переменные, которые необходимо исключить из системы.

Решение проводится в три этапа.

#### 1. Исключение промежуточных переменных.

Переменные исключаются по одной. Из исходной системы следует выбрать все уравнения, содержащие исключаемую переменную. Для этой переменной надо вычислить условия (П6) или (П7). Выбранные уравнения удаляются из системы, а вместо них включаются полученные условия.

#### 2. Разрешение неизвестных.

Неизвестные определяются поочередно. Также выбираются уравнения, содержащие очередную неизвестную; для нее вычисляются выражение (П5) и условия (П6) или (П7). Выбранные уравнения удаляются, а условия включаются в систему.

По ходу решения условия вида  $0 = 0$  удаляются, а при возникновении условия вида  $1 = 0$  решение прекращается ввиду противоречивости.

В результате либо не останется ни одного уравнения, либо оставшаяся система будет зависеть только от аргументов. Эта оставшаяся система и будет выражать условия существования решений (условия совместности исходной системы).

Каждое полученное выражение для очередной неизвестной будет в общем случае зависеть от других, еще оставшихся неразрешенными неизвестных. Последнее выражение уже будет зависеть только от аргументов и произвольных переменных.

#### 3. Обратные подстановки.

Последующие неизвестные, в порядке их разрешения на предыдущем этапе, подставляются в выражения предыдущих, и тем самым получается окончательное решение исходной системы уравнений.

*Научное электронное издание*

**Плеханов** Леонид Петрович

## **ОСНОВЫ САМОСИНХРОННЫХ ЭЛЕКТРОННЫХ СХЕМ**

Редактор *Т. Г. Хохлова*

Художественный редактор *Н. А. Новак*

Технический редактор *Е. В. Денюкова*

Корректор *Е. Н. Клитина*

Компьютерная верстка: *В. И. Савельев*

Подписано 02.04.13. Формат 60×90/16.

Усл. печ. л. 13.

Издательство «БИНОМ. Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: binom@Lbz.ru, <http://www.Lbz.ru>

*Системные требования:* процессор *Intel* или совместимый с ним с тактовой частотой от 1,3 ГГц и выше; операционная система *Microsoft Windows XP, Vista, 7* или *8*; от 256 Мб оперативной памяти; от 260 Мб свободного пространства на жестком диске; разрешение экрана не ниже 1024×768; программа *Adobe Reader* не ниже X.