

**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.

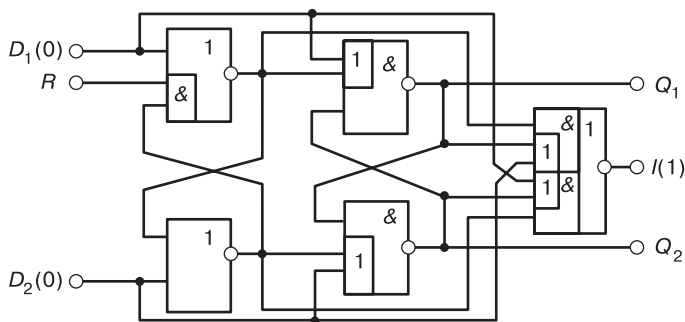
### 4.2.2.1. *D*-триггеры

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-

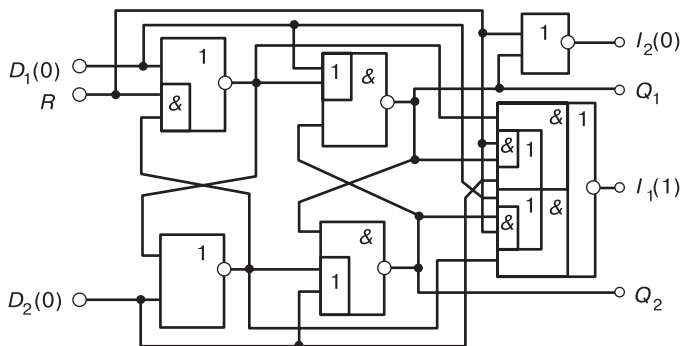
ся на некоторый период, предположительно больший длительности переходных процессов этой предустановки, после чего возвращается в единичное значение.



**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).

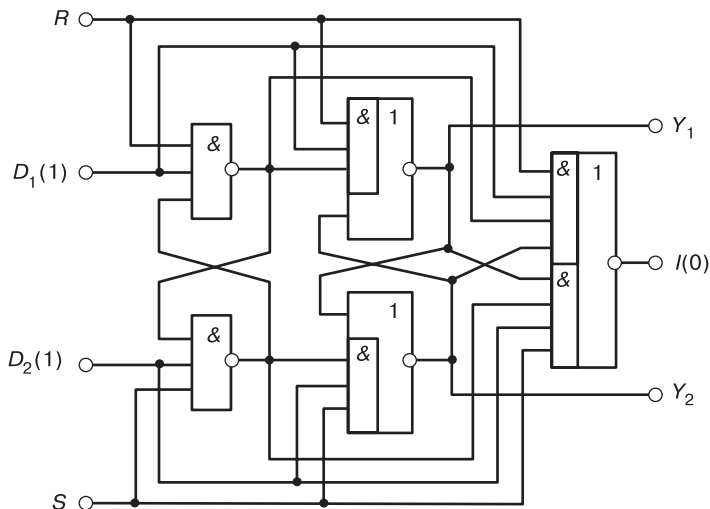
Схема этого триггера приведена на рис. 4.13. Предустановка производится также в спейсерной фазе. В отличие от предыдущей схемы, завершение предустановки индицируется сигналом  $I_2 = 1$ , после чего необходимо снять установочный сигнал и дождаться индикации спейсера  $I_2 = 0$ . Такой порядок предустановки 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  $I = 0$ .



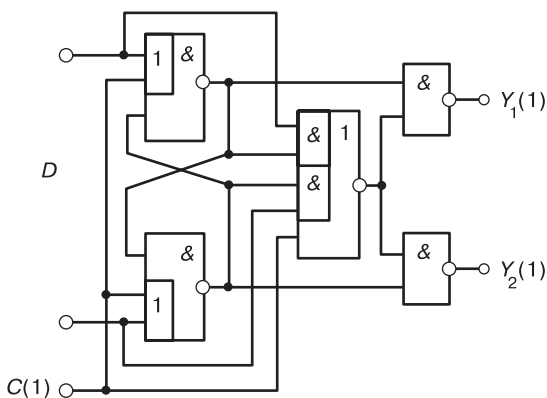
**Рис. 4.14.** Двухступенчатый  $D$ -триггер с самосинхронными установкой и сбросом:  $D_1, D_2$  — ПФС-вход;  $S$  — вход установки;  $R$  — вход сброса;  $Q_1, Q_2$  — БС-выход;  $I$  — общий И-сигнал

### 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

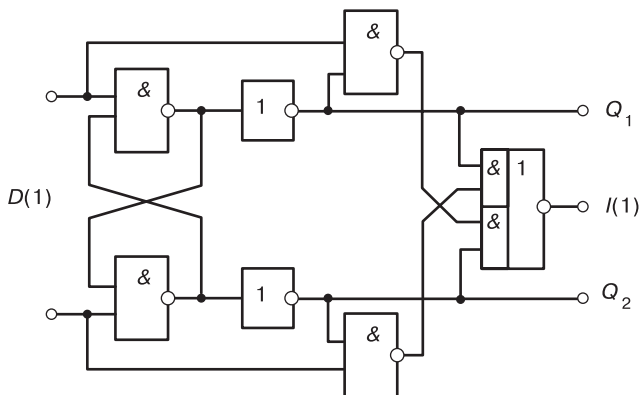


**Рис. 4.15.** Одноступенчатый  $D$ -триггер с преобразованием выхода в ПФС-сигнал:  
 $D$  — БС-вход;  $C$  —  $Y$ -сигнал;  $Y_1, Y_2$  — ПФС-выход

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).

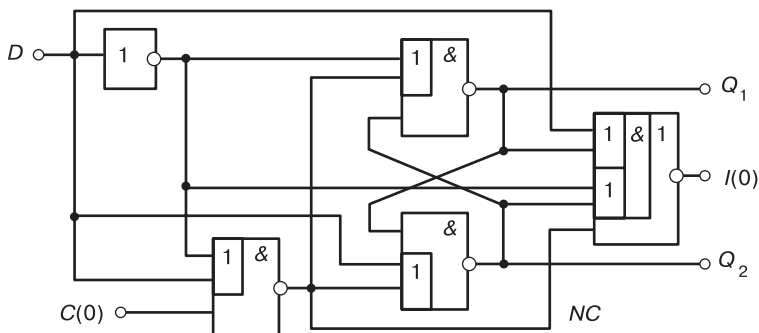


**Рис. 4.16.** Одноступенчатый  $D$ -триггер с ПФС-входом и усиленными выходами:  
 $D$  — ПФС-вход;  $Q_1, Q_2$  — БС-выход;  $I$  —  $I$ -сигнал

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)

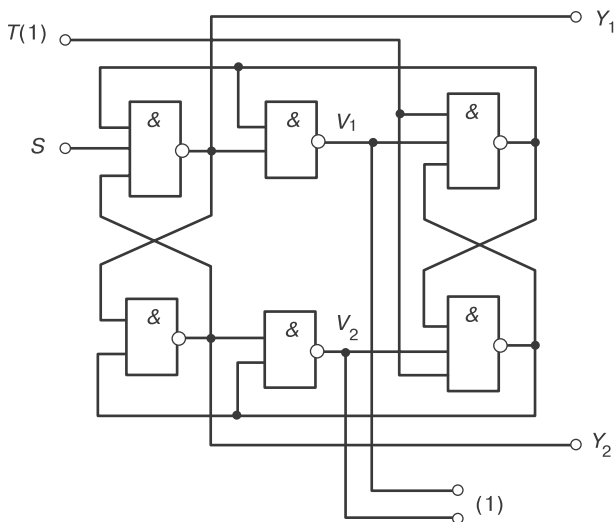
Счетные триггеры могут реализовываться на ячейках с гашением и без гашения. Как уже указывалось (см. раздел 3.4.1), при одинаковых затратах триггер с гашением характеризуется худшим быстродействием, чем триггер без гашения. Тем не менее триггеры с гашением широко использовались в схемах группы Варшавского.

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)

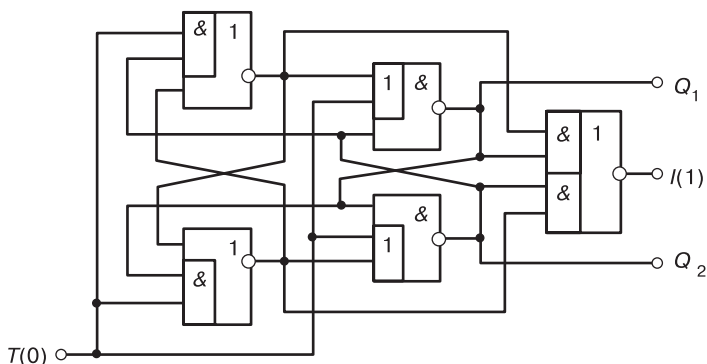
Установка триггера производится сигналом  $S = 0$  при  $T = 0$ . Окончание процесса установки в этой схеме не контролируется, что и отражено в названии «асинхронная установка» (в отличие от самосинхронной). Сигнал  $S = 0$  необходимо удерживать на время, «достаточное для завершения установки».

Некоторым схмотехническим 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 схмотехнике, — 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.

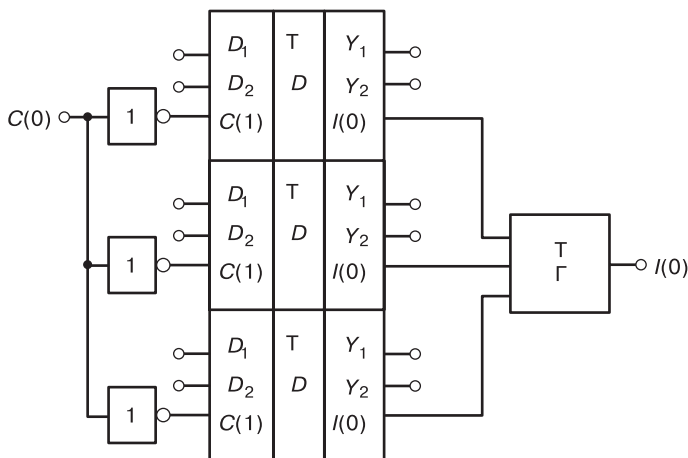
### 4.2.3. Параллельные регистры

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.



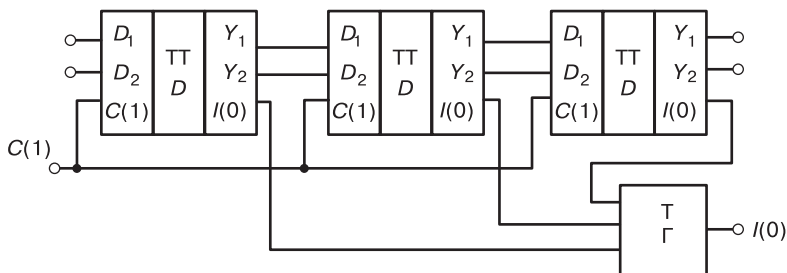
**Рис. 4.20.** Параллельный одноступенчатый регистр:  
Г — Г-триггер, С — У-сигнал, I — И-сигнал

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.

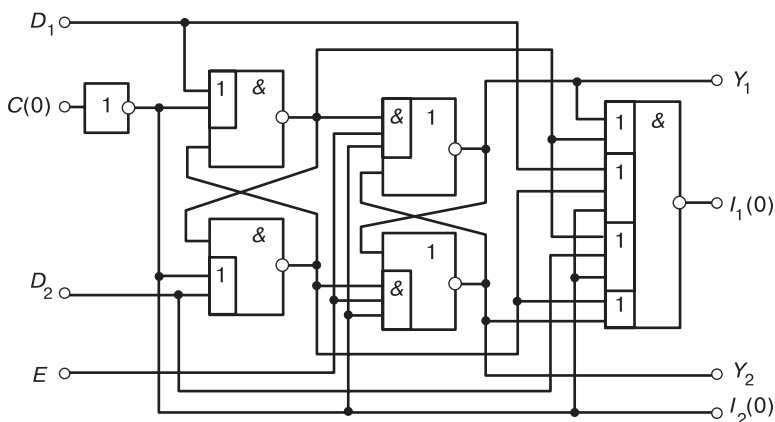
Простейший регистр сдвига можно построить на элементах рис. 3.13 с общим управляющим сигналом, как показано на рис. 4.21.



**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).



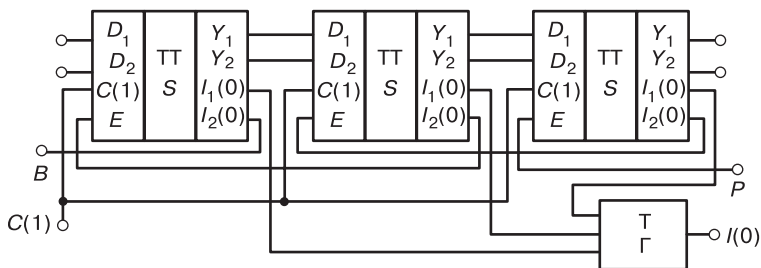
**Рис. 4.22.** Разряд самосинхронного регистра сдвига:  
 $D_1, D_2, Y_1, Y_2$  — входные и выходные БС-сигналы;  $C$  — У-сигнал;  
 $I_1, I_2$  — И-сигналы;  $E$  — вход дополнительной блокировки

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).

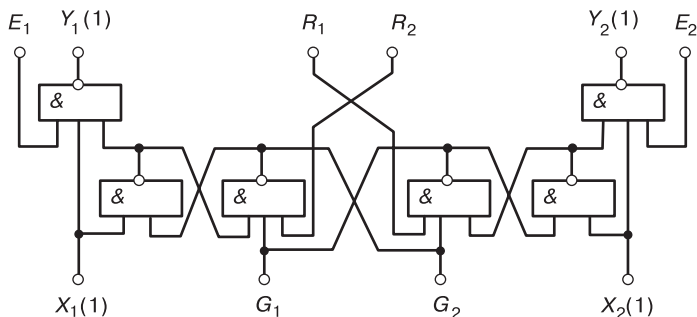


**Рис. 4.23.** Регистр сдвига на специальных ячейках:  
 $B$  — сигнал блокировки для предыдущей БСЯ (используется при необходимости);  $C$  —  $Y$ -сигнал;  $P$  — блокирующий сигнал от последующей БСЯ;  $I$  — И-сигнал

#### 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).

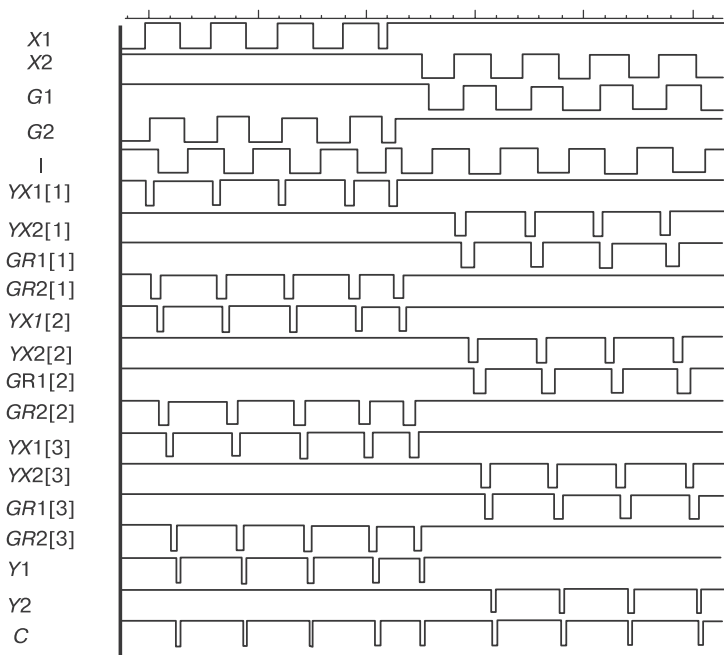


**Рис. 4.24.** Одноразрядная ячейка буферного конвейерного регистра FIFO:  
 $X_1, X_2$  — ПФС-входы;  $Y_1, Y_2$  — ПФС-выходы;  $R_1, R_2$  — ответные входы;  
 $G_1, G_2$  — ответные выходы;  $E_1, E_2$  — входы дополнительных блокировок

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. 4.26.** 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:  $X_2 \rightarrow YX2 [1] \rightarrow YX2 [2] \rightarrow YX2 [3] \rightarrow Y_2$ .

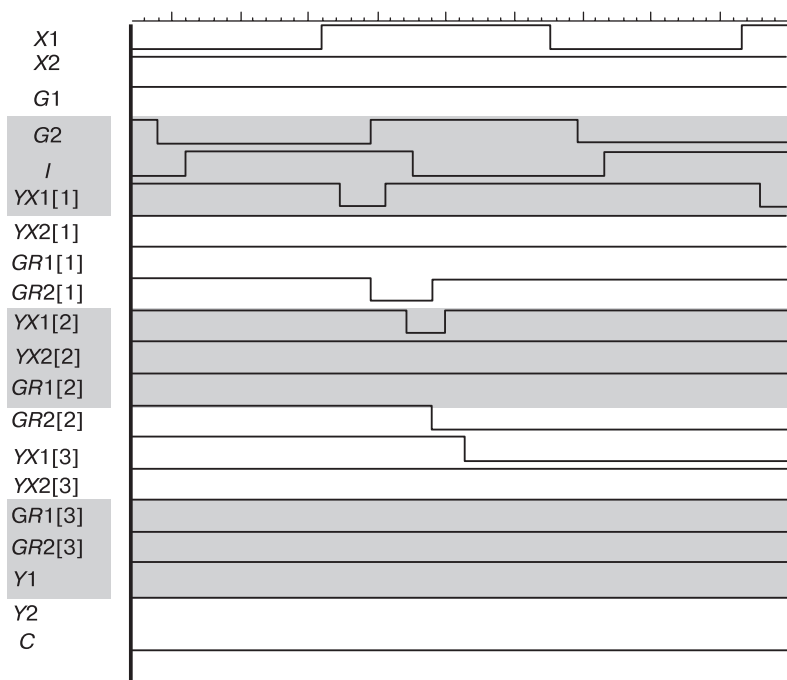
На рис. 4.27 показан случай, когда приемная сторона не принимает сигналы ( $C = 0$ ).

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.

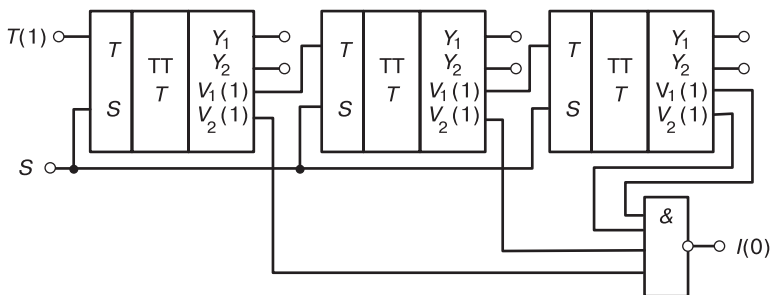
#### 4.2.5. Счетчики

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

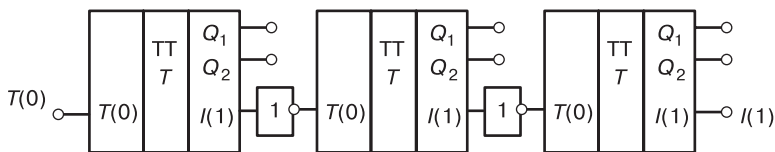


**Рис. 4.28.** Счетчик на триггерах с гашением:

$T$  — 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.

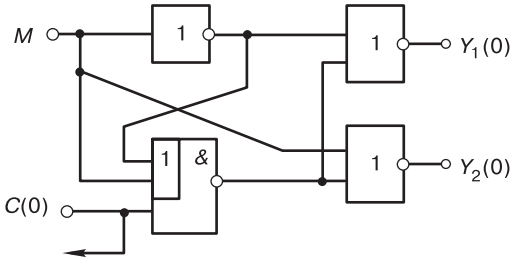
В литературе вопрос такого преобразования кратко обсуждается в [3, с. 355], однако представленная там схема многозарядного преобразователя [3, рис. 11.19] содержит ошибки и не является самосинхронной.

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



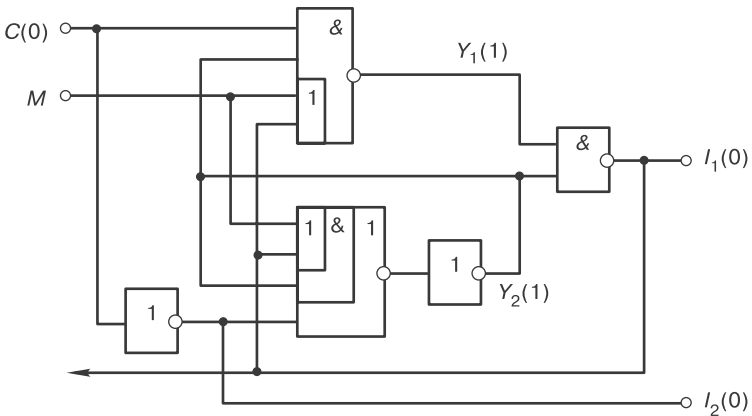
**Рис. 4.30.** Преобразователь М-сигнала в ПФС-сигнал:  $M$  — моносигнал;  $C$  — У-сигнал;  $(Y_1, Y_2)$  — ПФС-сигнал

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.

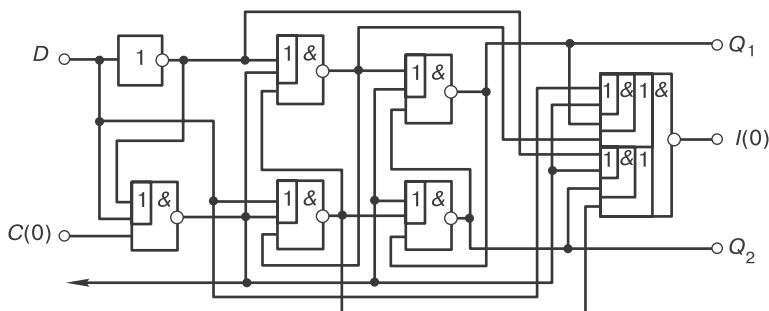


**Рис. 4.31.** Преобразователь М-сигнала в ПФС-сигнал с улучшенным взаимодействием источника и приемника:  $M$  — моносигнал;  $C$  — У-сигнал;  $(Y_1, Y_2)$  — ПФС-сигнал;  $I_1, I_2$  — И-сигналы

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

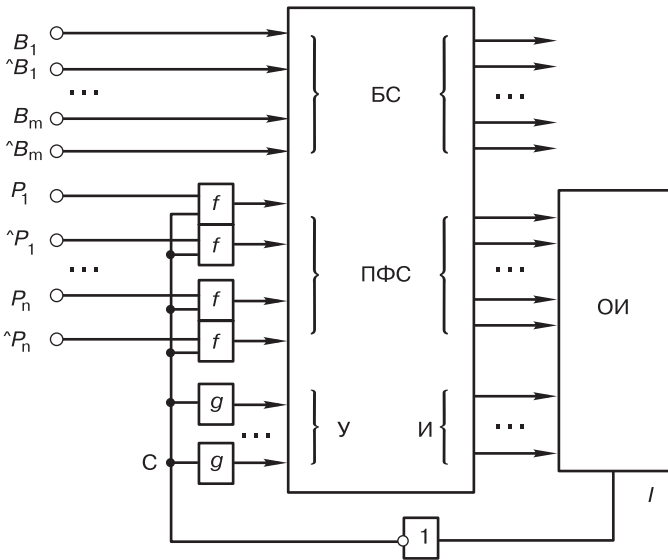
При константном замыкании все информационные входы считаются неизменными (константами), т. е. выступают как *параметры анализа*. В этом случае согласованное замыкание строится, как показано на рис. 5.1.

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.

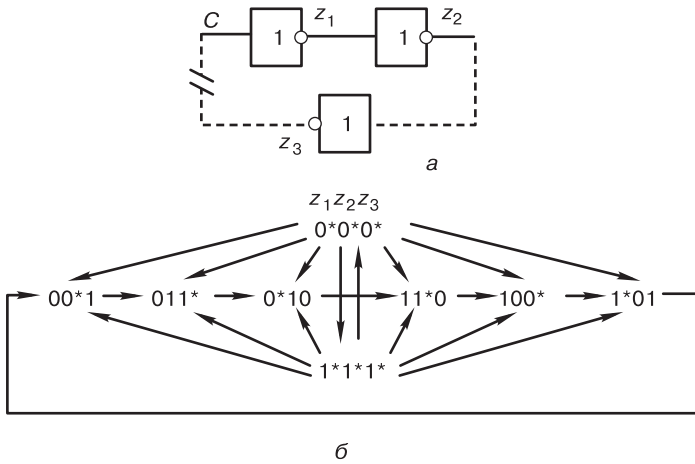
#### 5.4.1. Прямой метод по диаграммам переходов (ДП)

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.

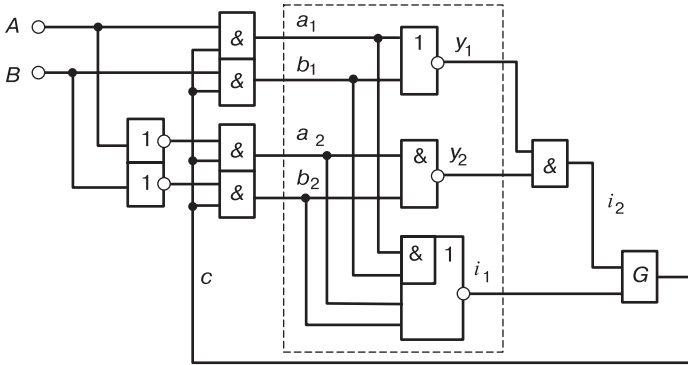
Цикличность обнаруживается путем запоминания некоторого слоя, называемого *контрольным*, и сравнения с ним последующих слоев.

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 Г-триггер (см. главу 3), имеющий уравнение:

$$c = i_1 i_2 \vee c^-(i_1 \vee i_2), \tag{5.1}$$

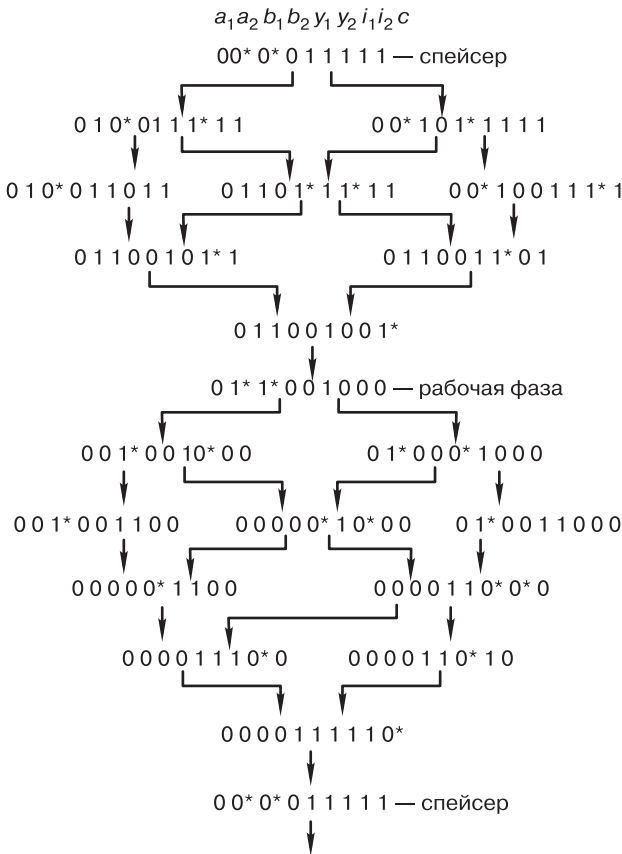
где  $c^-$  — предыдущее значение  $c$ .

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.

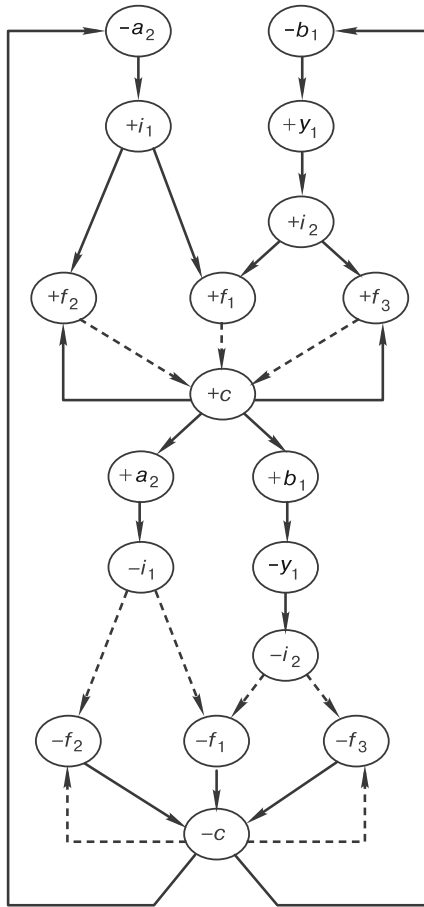


Рис. 5.5. Диаграмма изменений схемы, изображенной на рис. 5.3

В процессе анализа по уравнениям замкнутой схемы, поуровню, 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.

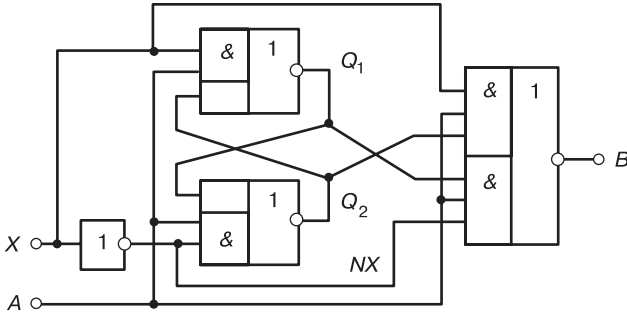
Неучет изменений инфовходов в сеансе даже с полным перебором факторов состояний приводит к ошибочным выводам, что было впервые замечено при практическом проектировании СС-схем в ИПИРАН (Ю. А. Степченков).

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.

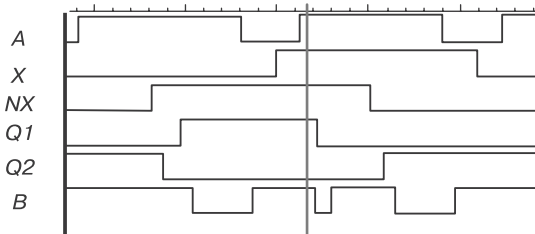
В событийном подходе, и вообще в литературе по самосинхронике, подобная задача полного анализа до публикации автора [43] не рассматривалась.

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.

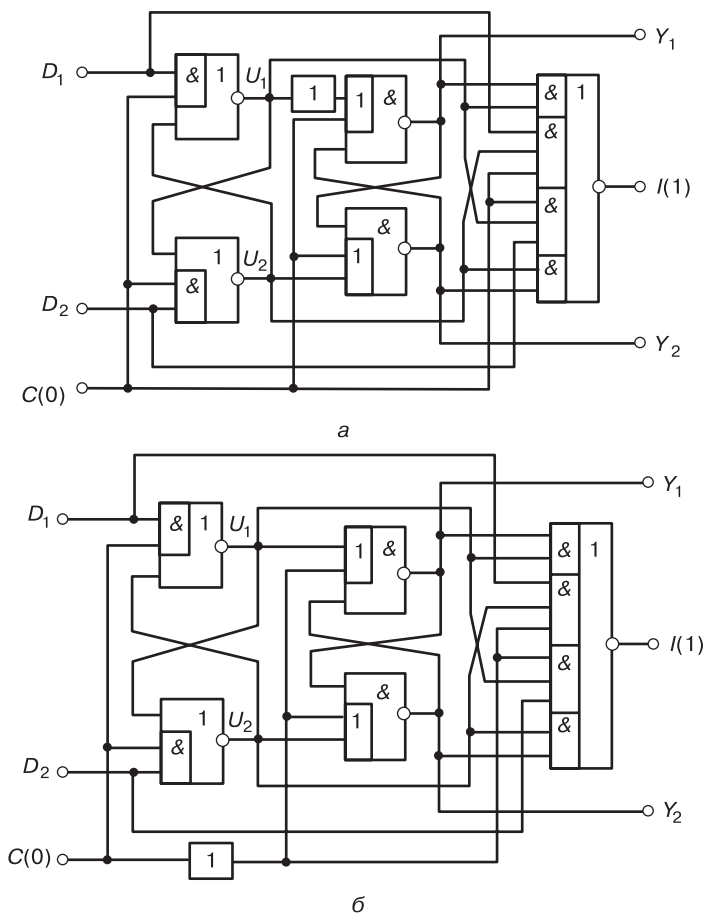
Исходя из сказанного в разделе 1.2, критерий самосинхронности разомкнутых схем в функциональном подходе, при соблюдении соглашений модели Маллера, можно сформулировать так.

*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:  
*а* — 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  $V$ :

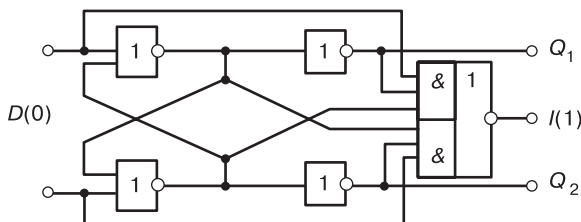
$$D^* \wedge M \vee \wedge D^* M = 0. \quad (5.10)$$

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: входа 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 антитонной 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.

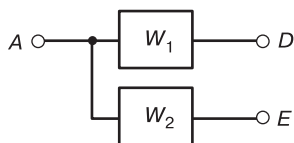
Обязательным для успешной проверки является присутствие всех внутренних сигналов схемы хотя бы в одном из выходных списков индицирования. Присутствие внешних фазовых сигналов не обязательно, так как они могут быть индицированы в схемах верхнего уровня (частичная индицируемость рассматриваемой схемы).

### 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

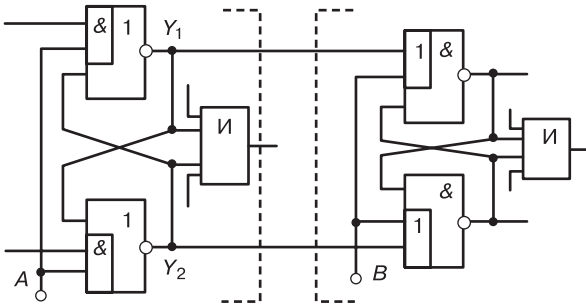
Рассмотрим схематично соединение БС-сигналов двух фрагментов. В принимающем фрагменте БС-сигналы могут поступать либо на БСЯ, либо на другие схемы, например преобразователь БС-ПФС. Схему, принимающую БС-сигнал, будем называть далее *приемником*.

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  $V$  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).

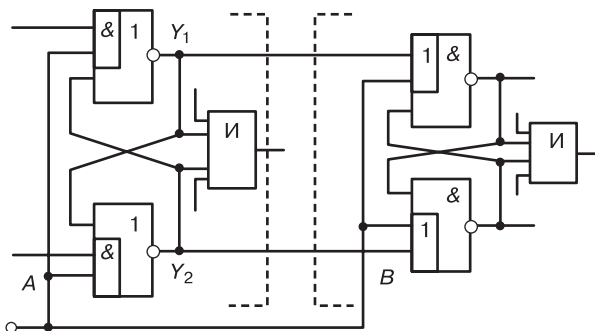


Рис. 5.12. Корректное управление с помощью единого сигнала

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.

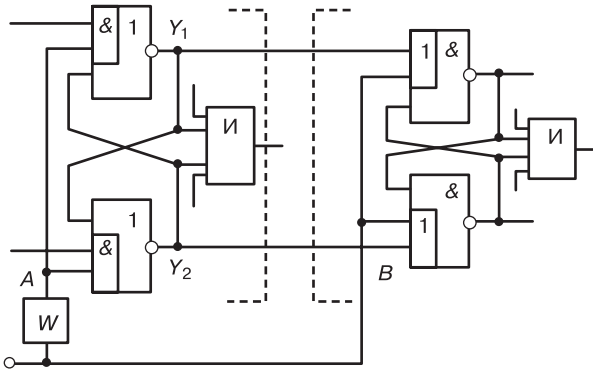
#### В. Управление «с правой задержкой».

Этот тип управления показан на рис. 5.14.

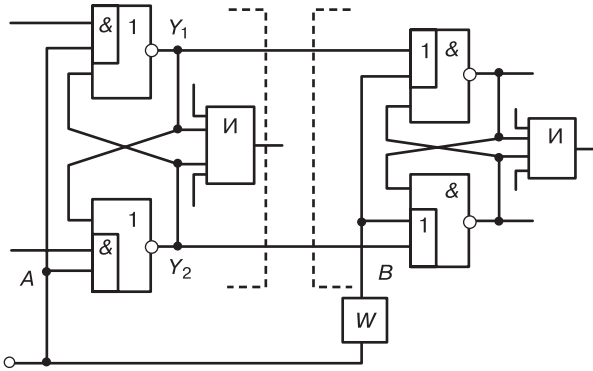
При таком способе сигналы ( $Y_1$ ,  $Y_2$ ) и  $B$  являются конкурентными с общим инициатором  $A$ , и БС-правило нарушается.

#### 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-

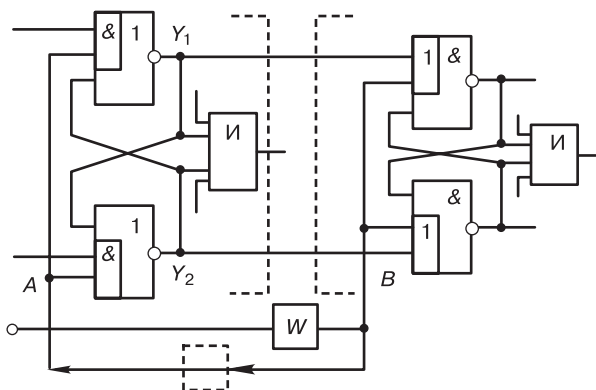


Рис. 5.15. Управление с помощью локальной обратной связи

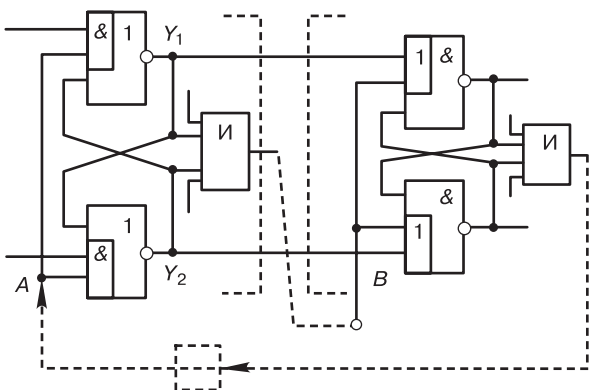


Рис. 5.16. Управление с помощью общей обратной связи

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**

одионочный независимый сигнал

**Сигнал унарный стабильный**

одионочный 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

$$\bigvee_{i=1}^N (Va_i) \vee \bigwedge_{i=1}^N (Vb_i) = 0, \quad (\text{П6})$$

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.