

В. Ф. Романов

Лекции по теории автоматов

Часть 3

Структурный синтез автоматов

Владимир 2009

Учебное пособие для студентов очной и заочной форм обучения специальностям в области вычислительной техники, информатики и управления.

ОГЛАВЛЕНИЕ

Часть 3. Структурный синтез автоматов.....	3
3.1. Канонический метод структурного синтеза.....	3
3.2. Триггеры.....	4
3.3. Регистры.....	13
3.4. Сумматоры.....	15
3.5. Синтез схемы автомата Мура по заданному графу	16
3.6. Счетчики.....	20
3.7. Кодирование состояний автомата. Пример синтеза автомата Мили.....	22
Задачи и упражнения.....	26
Литература.....	26

ЧАСТЬ 3. СТРУКТУРНЫЙ СИНТЕЗ АВТОМАТОВ

3.1. Канонический метод структурного синтеза

Как известно (см. часть 1), абстрактный автомат имеет один вход, один выход и конечные алфавиты для входных, выходных символов и символов, кодирующих состояния (мощности этих алфавитов обозначались m , p , n , соответственно). Структурный автомат имеет M входов и P выходов, причем каждый из этих параметров закодирован в структурном (двоичном) алфавите; состояния автомата закодированы двоичными векторами длины N . Значения M , P и N вычисляются исходя из соотношений:

$$M \geq \lfloor \log_2 m \rfloor + 1; P \geq \lfloor \log_2 p \rfloor + 1; N \geq \lfloor \log_2 n \rfloor + 1.$$

Если нумерация элементов алфавитов абстрактного автомата начинается с нуля (соответственно, x_0 , y_0 , q_0), то удобно использовать равенства:

$$M = \lceil \log_2 m \rceil; P = \lceil \log_2 p \rceil; N = \lceil \log_2 n \rceil.$$

Существует общий конструктивный прием – *канонический метод структурного синтеза*, – позволяющий свести задачу построения автомата с памятью к задаче синтеза комбинационных схем (КС) [1]. Этот метод основан на представлении автомата схемами, приведенными на рис. 1 (для автомата Мили) и рис. 2 (для автомата Мура).

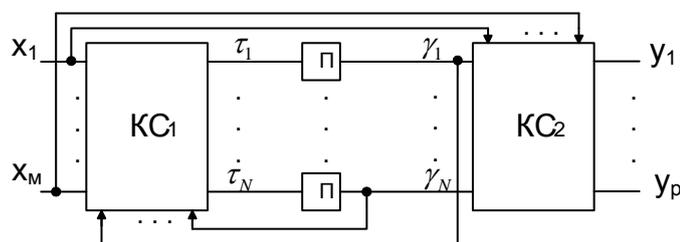


Рис. 1. Каноническая структура автомата Мили

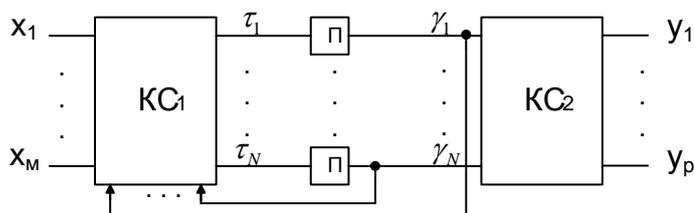


Рис. 2. Каноническая структура автомата Мура

Каждый из автоматов построен на двух комбинационных схемах – КС1 и КС2, и двоичных элементах памяти, обозначенных через П, число которых равно N . В качестве элементов памяти мы будем далее использовать *триггеры*; совокупности двоичных значений их выходов (на рисунках это $\gamma_1, \dots, \gamma_N$), образуют двоичные векторы, кодирующие состояния. На основе векторов $X(x_1, \dots, x_M)$ и $\Gamma(\gamma_1, \dots, \gamma_N)$, поступивших на входы КС1 в момент t дискретного времени, последняя формирует *сигналы возбуждения элементов памяти* (τ_1, \dots, τ_N), переводящие автомат в новое состояние в момент $t + 1$. КС2 служит для формирования выходных сигналов на основе векторов X и Γ (для автомата Мили) или только вектора Γ (для автомата Мура).

Функции переходов автоматов Мили и Мура:

$$\left. \begin{aligned} \tau_1 &= \tau_1(x_1, \dots, x_M, \gamma_1, \dots, \gamma_N) \\ \dots \\ \tau_N &= \tau_N(x_1, \dots, x_M, \gamma_1, \dots, \gamma_N) \end{aligned} \right\}$$

Функции выходов автомата Мили:

$$\left. \begin{aligned} y_1 &= y_1(x_1, \dots, x_M, \gamma_1, \dots, \gamma_N) \\ \dots \\ y_P &= y_P(x_1, \dots, x_M, \gamma_1, \dots, \gamma_N) \end{aligned} \right\}$$

Функции выходов автомата Мура:

$$\left. \begin{aligned} y_1 &= y_1(\gamma_1, \dots, \gamma_N) \\ \dots \\ y_P &= y_P(\gamma_1, \dots, \gamma_N) \end{aligned} \right\}$$

Таким образом, работа схем полностью согласуется с функционированием автоматов, описанным в разделе 1.1 части 1 настоящего пособия.

3.2. Триггеры

Триггер – элементарный автомат Мура, предназначенный для приема, хранения и выдачи одного бита информации. Функция памяти достигается введением обратных связей в простейшую КС, построенную на логических элементах. Триггер имеет два выхода – *прямой и инверсный*, значение сигнала на прямом выходе отображает *состояние триггера*. Существует множество видов триггеров [2]; рассмотрим последовательно основные из них.

Асинхронный RS-триггер (Reset – сброс, Set – установка)

На рис. 3 и 4 приведены соответственно схема триггера, построенного на двух элементах И-НЕ, и его условное обозначение. Инверсные записи входных сигналов показы-

вают, что в данной схеме установка триггера в "1" осуществляется нулевым сигналом: $S = 0$ (см. таблицу 1 переходов триггера).

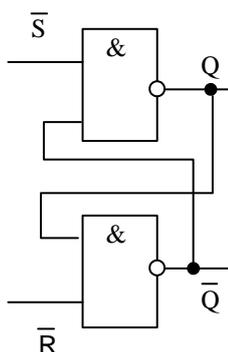


Рис. 3. Асинхронный RS -триггер с инверсными входами

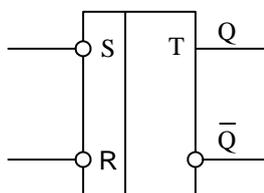


Рис. 4. Условное обозначение RS -триггера

Таблица 1

Таблица переходов RS -триггера с инверсными входами

t		$t + 1$	Примечание
S	R	Q	
1	1	$Q(t)$	Хранение информации
0	1	1	Установка «1»
1	0	0	Установка «0»
0	0	–	Запрещенная комбинация

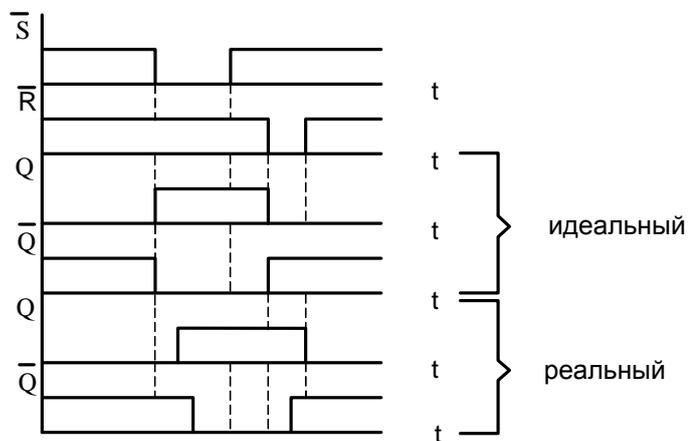


Рис. 5. Временная диаграмма переходов RS -триггера

На рис. 5 отражена условная задержка сигнала τ на одном логическом элементе при рассмотрении реальных схем (см. пунктирные линии). В данной схеме триггера задержка момента установки триггера в новое состояние по отношению к моменту подачи входных сигналов составляет 2τ . При *запрещенной комбинации* на входе триггер "распадается" на два независимых инвертора, поэтому такое сочетание входных сигналов не должно применяться.

Модификацией рассмотренного триггера является триггер, изображенный на рис. 6. Схема построена таким образом, что установка триггера в "1" осуществляется сигналом $S = 1$ (см. таблицу 2). Условное обозначение триггера приведено на рис. 7.

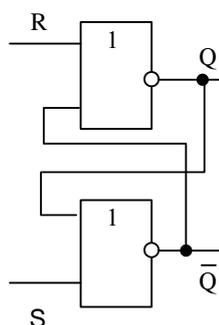


Рис. 6. Модифицированный асинхронный *RS*-триггер

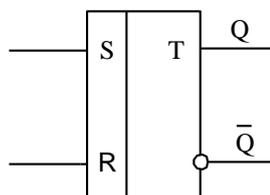


Рис. 7. Условное обозначение асинхронного *RS*-триггера

Таблица 2

Таблица переходов асинхронного *RS*-триггера

t		$t + 1$	Примечание
S	R	Q	
0	0	$Q(t)$	Хранение информации
0	1	1	Установка «0»
1	0	0	Установка «1»
1	1	–	Запрещенная комбинация

Аналитически функционирование *RS*-триггера можно описать уравнением

$$Q(t+1) = S(t) \vee Q(t) \bar{R}(t).$$

Синхронизируемые триггеры.

Схема рис. 3 при небольшом изменении превращается в схему одноконтного *синхронизируемого* (или *синхронного*) *RS*-триггера (рис. 8). Очевидно, что при сигнале синхронизации $t = 0$ происходит хранение информации, а при $t = 1$ осуществляются переходы в соответствии с таблицей 2 для асинхронного *RS*-триггера. На рис. 9 и далее вход синхронизации обозначен символом *C*.

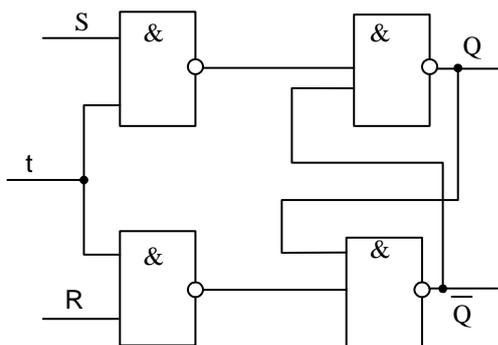


Рис. 8. Синхронный *RS*-триггер

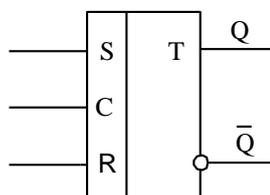


Рис. 9. Условное обозначение синхронного *RS*-триггера

Двухтактный *RS*-триггер. Устойчивая работа одноконтных *RS*-триггеров в произвольной схеме возможна только в случае, если занесение в триггер информации осуществляется после завершения передачи информации о прежнем его состоянии в другой триггер. Это достаточно просто обеспечивается при использовании двух серий находящихся в противофазе синхросигналов.

Такой принцип обмена информацией реализован в двухтактных *RS*-триггерах.

Простейшая схема двухвходового двухтактного *RS*-триггера показана на рис. 10. Она состоит из двух одноконтных *RS*-триггеров и инвертора в цепи синхронизации. При поступлении на вход *RS*-триггера сигнала $t = 1$ входная информация заносится в первый одноконтный *RS*-триггер, а второй будет при этом хранить информацию, относящуюся к предыдущему периоду представления. По окончании действия сигнала синхронизации,

когда $t = 0$, а $\bar{t} = 1$, первый RS -триггер перейдет в режим хранения, а второй переписет из него новое значение выходного сигнала. Рис. 11 иллюстрирует переключения во времени первого и второго триггеров при поступлении на вход t (вход синхронизации) синхроимпульсов (обозначаемых здесь и далее сокращением $СИ$). На рис. 12 представлено условное обозначение двухтактного RS -триггера.

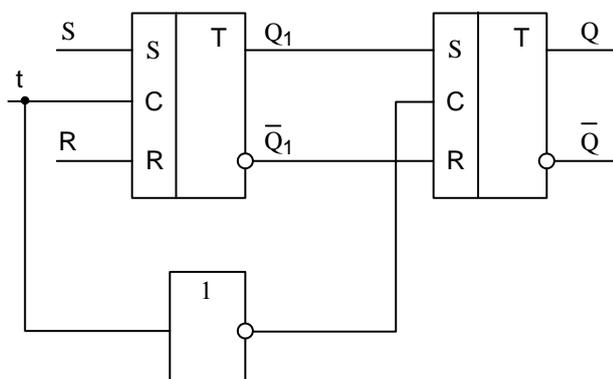


Рис. 10. Синхронный двухтактный RS -триггер

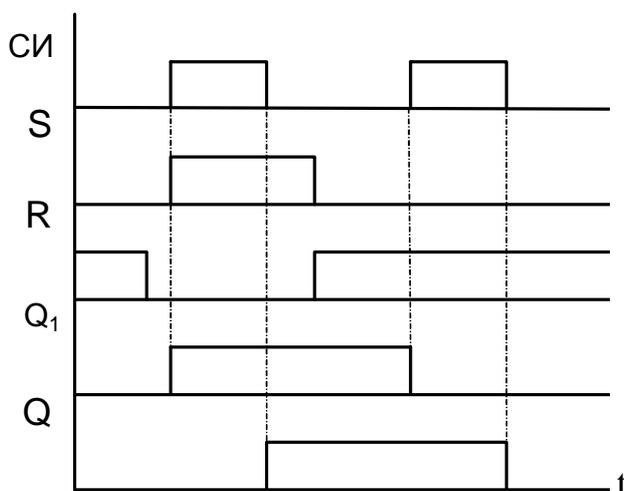
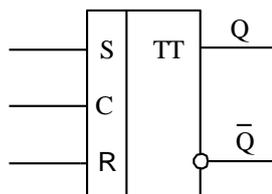


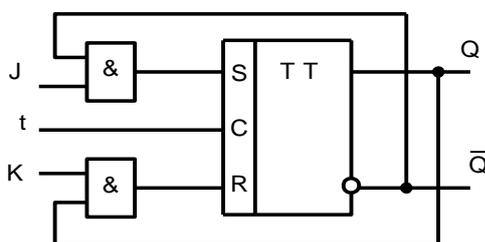
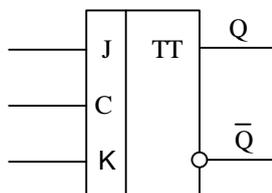
Рис. 11. Временная диаграмма переключений двухтактного триггера

Двухтактный триггер изменяет свои состояния только по окончании действия сигнала синхронизации $t = 1$ (переход в режим хранения информации). Поэтому из двухтактных триггеров можно строить произвольные схемы, в том числе подавать сигналы с выхода триггера на его вход.

Схемы RS -триггеров составляют основу для построения других триггерных схем, таких, как T -, D - и JK -триггеры.

Рис. 12. Условное обозначение двухтактного RS -триггера

JK -триггер. Распространенным типом триггера в системах интегральных логических элементов является универсальный двухтактный JK -триггер, схема которого показана на рис.13, а условное обозначение на рис.14.

Рис. 13. Двухтактный JK -триггерРис. 14. Условное обозначение двухтактного JK -триггера

Из таблицы переходов (табл. 3) следует, что для входов JK -триггера отсутствует запрещенная комбинация; сочетание $S = 1, K = 1$ может использоваться для выполнения полезного действия – инверсии (т. е. переключения триггера). При остальных сочетаниях значений входов легко увидеть соответствие входов J и K входам S и R RS -триггера.

Таблица 3Таблица переходов JK -триггера

t		$t + 1$	Примечание
J	K	Q	
0	0	$Q(t)$	Хранение информации
0	1	1	Установка «0»
1	0	0	Установка «1»
1	1	$\bar{Q}(t)$	Инверсия

Временная диаграмма переключения состояний триггера в зависимости от входных информационных сигналов и $СИ$ (рис. 15) показывает, что переключения двухтактного триггера происходят по "заднему фронту" синхроимпульса.

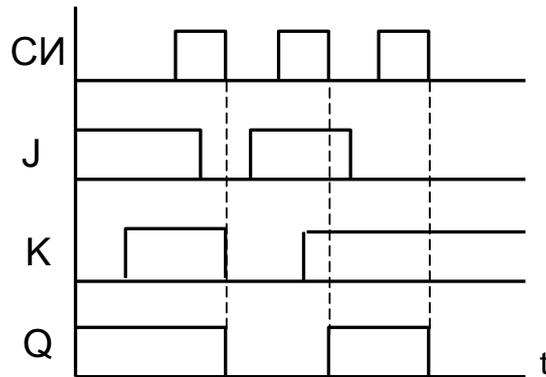


Рис. 15. Временная диаграмма переключения состояний JK -триггера

У рассматриваемого триггера, как и у триггеров других типов могут присутствовать входы асинхронной (несинхронизируемой) установки R и S (рис. 16), с помощью которых при $C = 0$ триггер может быть установлен в состояние 1 путем подачи значений $R = 1$ и $S = 0$, либо в состояние 0 путем подачи $R = 0$ и $S = 1$. При подаче сигналов $R = S = 1$, не изменяющих состояние схемы, работа триггера осуществляется под воздействием синхронизируемых входов ($C = 1$). В этом случае функционирование триггера описывается таблицей 3. На рис. 16 изображены также встроенные конъюнктивные входы, расширяющие возможности схемы, при этом $J = J_1 J_2 J_3$ и $K = K_1 K_2 K_3$.

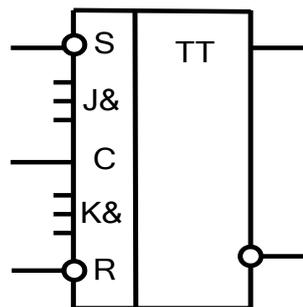


Рис. 16. Условное обозначение JK -триггера с входами асинхронной установки

Переходы JK -триггера описываются булевой функцией

$$Q(t+1) = \bar{K}(t)Q(t) \vee \bar{K}(t)J(t) \vee \bar{Q}(t)J(t),$$

при условии, что $S \& R = 1$ (если в триггере имеются входы асинхронной установки).

JK -триггер удобен тем, что при различных вариантах подключения его входов можно получить схемы, функционирующие как RS , T и D -триггеры.

T-триггер. Триггер со счетным входом (*T*-триггер) и простейшем случае может быть построен с использованием двухтактного синхронизируемого *JK*-триггера (рис. 17).

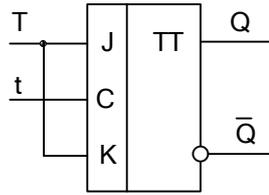


Рис. 17. Двухтактный *T*-триггер, построенный с использованием *JK*-триггера

Из таблицы 3 следует, что при $T = 0$ триггер хранит информацию, а при $T = 1$ происходит переключение состояния триггера по заднему фронту каждого синхроимпульса (рис. 18). Фиксация значения $T = 1$ приводит к тому, что триггер фактически имеет один вход (счетный вход t), и триггер становится асинхронным. На рис. 19 показано его условное обозначение.

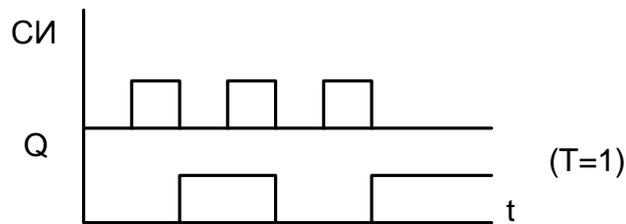


Рис. 18. Временная диаграмма переключений асинхронного *T*-триггера

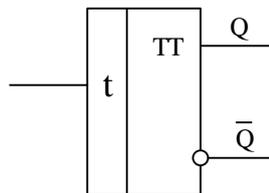


Рис. 19. Условное обозначение асинхронного *T*-триггера

Рисунки 20 и 21 относятся к синхронному *T*-триггеру, в котором возможность управления режимами (хранение информации или счет) сохранена.

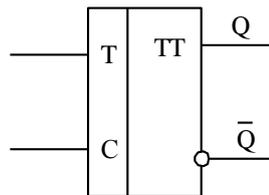


Рис. 20. Условное обозначение синхронного *T*-триггера

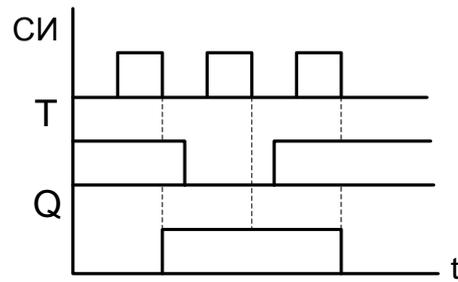


Рис. 21. Временная диаграмма переключений синхронного T -триггера

D -триггер. Название триггера происходит от слова *delay* (задержка). Этот триггер (рис. 22, 23) повторяет на выходе значение сигнала, поданного на вход, с задержкой, определяемой расположением во времени информационного и синхронизирующего сигналов.

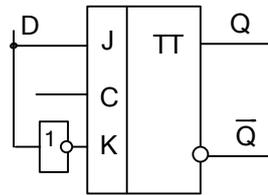


Рис. 22. D -триггер, построенный с использованием JK -триггера

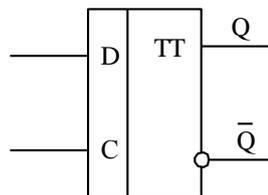


Рис. 23. Условное изображение D -триггера

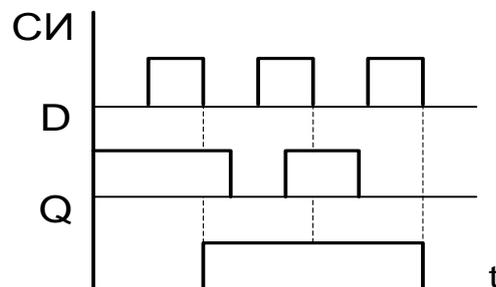


Рис. 24. Временная диаграмма переключений D -триггера

Характеристические таблицы. При синтезе схем, содержащих триггеры, удобно использовать таблицы, определяющие входные сигналы как функцию заданных переключений.

чений триггеров. Такие таблицы (названные *характеристическими*, см. ниже) строятся на основе описанных выше таблиц переходов.

Таблица 4

RS-триггер

$Q(t)$	$Q(t+1)$	S	R
0	0	0	*
0	1	1	0
1	0	0	1
1	1	*	0

Таблица 5

JK-триггер

$Q(t)$	$Q(t+1)$	J	K
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

Таблица 6

T-триггер

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

Таблица 7

D-триггер

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

3.3. Регистры

Регистром называется устройство, предназначенное для запоминания слова, а также для выполнения над словом некоторых логических преобразований. Регистр представляет собой совокупность триггеров, число которых соответствует числу разрядов в слове, и вспомогательных схем, обеспечивающих выполнение некоторых операций, среди которых могут быть:

- установка регистра в 0 (сброс);
- прием слова из другого устройства;
- передача слова в другой регистр;
- сдвиг слова вправо или влево на требуемое число разрядов;
- преобразование последовательного кода слова в параллельный и наоборот;
- поразрядные логические операции.

На рис. 25 показана схемная реализация приема разрядов входного слова в регистр (по синхросигналу) и передачи слова из регистра на внешнее устройство (по управляющему сигналу y). Здесь $a_1, \dots, a_n, b_1, \dots, b_n$ – разряды соответственно входного и выходного слов.

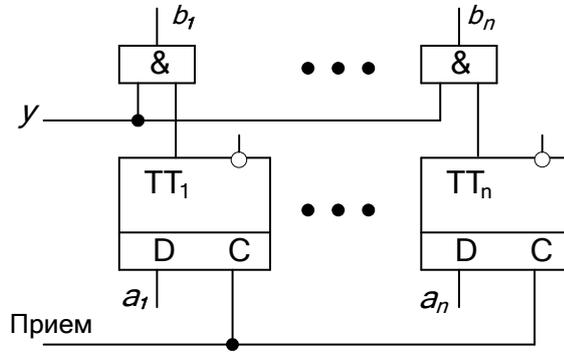


Рис. 25. Прием и передача информации в регистре

На рис. 26 приведены схемы для выполнения поразрядных логических операций:

$$r := r \vee r^*; \quad r := r \& r^*; \quad r := r \oplus r^*,$$

осуществляемых по синхросигналу.

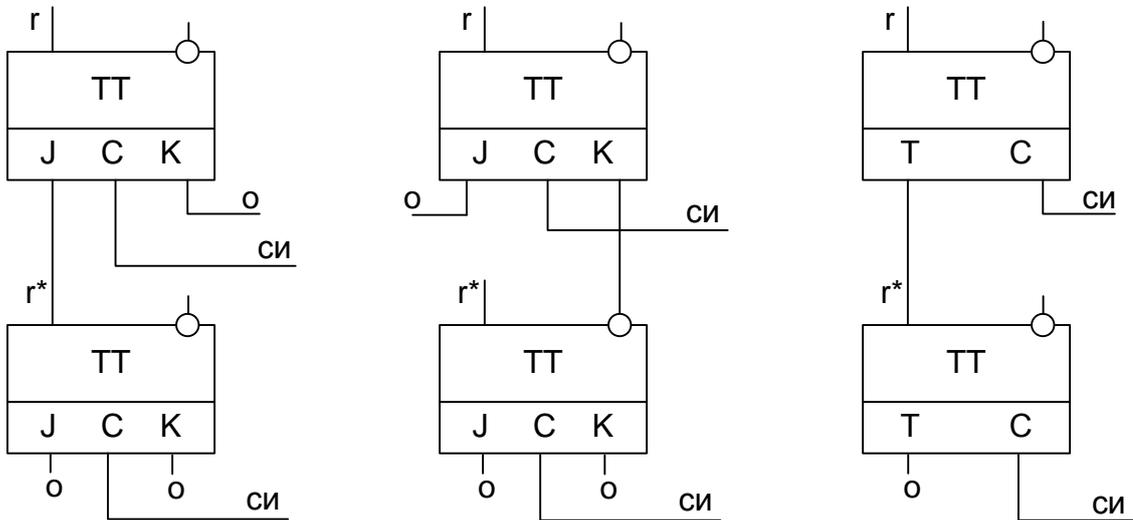


Рис. 26. Реализация поразрядных логических операций

На рис. 27 приведен пример условного изображения многофункционального регистра. Горизонтальные линии снизу и сверху выделяют разряды слов, которые подлежат приему и передаче (Y_1 и Y_2 – управляющие сигналы).

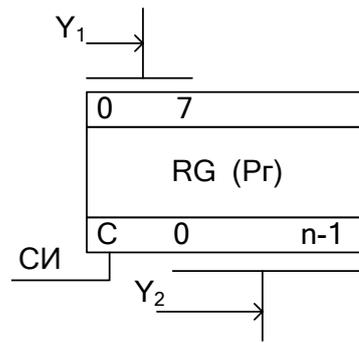


Рис. 27. Условное изображение многофункционального регистра

3.4. Сумматоры

Сумматором называется узел вычислительного устройства, выполняющий арифметическое суммирование кодов чисел. Обычно сумматор представляет собой комбинацию одноразрядных суммирующих схем. В таблице 8 приведены восемь возможных вариантов, возникающих при сложении двух двоичных чисел в одном разряде.

Таблица 8

Сложение двух двоичных чисел в одном разряде.

p_i	a_i	b_i	s_i	p_{i+1}
0	0	0	0	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	0	1	0	1
1	1	1	1	1

Обозначения:

p_i – цифра переноса из предыдущего разряда; a_i – первое слагаемое; b_i – второе слагаемое; s_i – сумма; p_{i+1} – цифра переноса в старший разряд.

Логические функции для описания одноразрядного сумматора:

$$s_i = a_i \bar{b}_i \bar{p}_i \vee \bar{a}_i b_i \bar{p}_i \vee a_i \bar{b}_i p_i \vee a_i b_i p_i;$$

$$p_{i+1} = a_i b_i \vee a_i p_i \vee b_i p_i.$$

Поскольку записанные соотношения реализуются комбинационными схемами, такие сумматоры получили название комбинационных. Условное обозначение рассмотренного сумматора приведено на рис. 28.

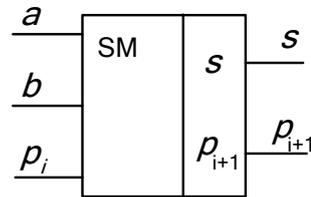


Рис. 28. Условное обозначение одноразрядного комбинационного сумматора

Параллельный (многоразрядный) сумматор может быть составлен из одноразрядных сумматоров, число которых равно числу разрядов слагаемых, путем соединения выхода, на котором формируется сигнал переноса данного разряда, с входом для сигнала переноса соседнего старшего разряда. При этом схемы сумматоров прямого, дополнительного и обратного кодов будут различаться [3].

Упражнение 1. Построить схемы названных многоразрядных сумматоров.

После того как выработан результат сложения на выходах комбинационных схем формирования суммы, он обычно запоминается в отдельном триггерном регистре.

3.5. Синтез схемы автомата Мура по заданному графу

Рассмотрим пример (рассчитанный на обобщение) синтеза схемы автомата Мура по заданному графу (рис. 29).

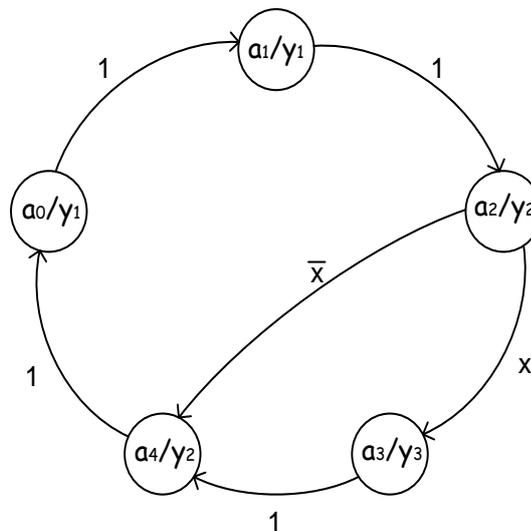


Рис.29. Граф автомата Мура

Для управления переходами автомата здесь используется один двоичный входной сигнал $x \in \{0, 1\}$ (соответственно \bar{x} и x). Символ ‘1’ при дуге графа обозначает любое значение входного сигнала – при этом переход в новое состояние происходит по синхросигналу.

Для синтеза целесообразно использовать следующие шаги.

1. Определение числа триггеров.

В данном примере $n = \lceil \log_2 5 \rceil = 3$.

2. Определение типа триггеров.

Тип триггеров в зависимости от постановки задачи может задаваться или выбираться исходя из некоторых соображений целесообразности. Пусть по заданию должны использоваться JK-триггеры.

3. Кодирование состояний автомата.

Закодируем состояния автомата трехразрядными двоичными числами в их естественном порядке (такой способ кодирования не является универсальным; ниже будут рассмотрены правила кодирования состояний, оптимизирующие схему автомата, построенного на триггерах различных типов)

4. Составление структурной таблицы.

Таблица 9

Структурная таблица

№ перех.	Сост. a_i	Код состояния			Сост. a_{i+1}	Код состояния			Вход. сигналы	Сигналы возбуждения						Выход. сигналы
		Q_3	Q_2	Q_1		Q_3	Q_2	Q_1		J_3	K_3	J_2	K_2	J_1	K_1	
1	a_0	0	0	0	a_1	0	0	1	1	0	*	0	*	1	*	y_1
2	a_1	0	0	1	a_2	0	1	0	1	0	*	1	*	*	1	y_1
3	a_2	0	1	0	a_3	0	1	1	x	0	*	*	0	1	*	y_2
4	a_2	0	1	0	a_4	1	0	0	\bar{x}	1	*	*	1	0	*	y_2
5	a_3	0	1	1	a_4	1	0	0	1	1	*	*	1	*	1	y_3
6	a_4	1	0	0	a_0	0	0	0	1	*	1	0	*	0	*	y_2

5. Получение минимизированных функций возбуждения триггеров, переключающих их в новое состояние:

$$J_i = J_i(Q_1, Q_2, Q_3, x); K_i = K_i(Q_1, Q_2, Q_3, x); (i = 1, 2, 3).$$

Для этой цели удобно использовать карты Карно (рис. 30).

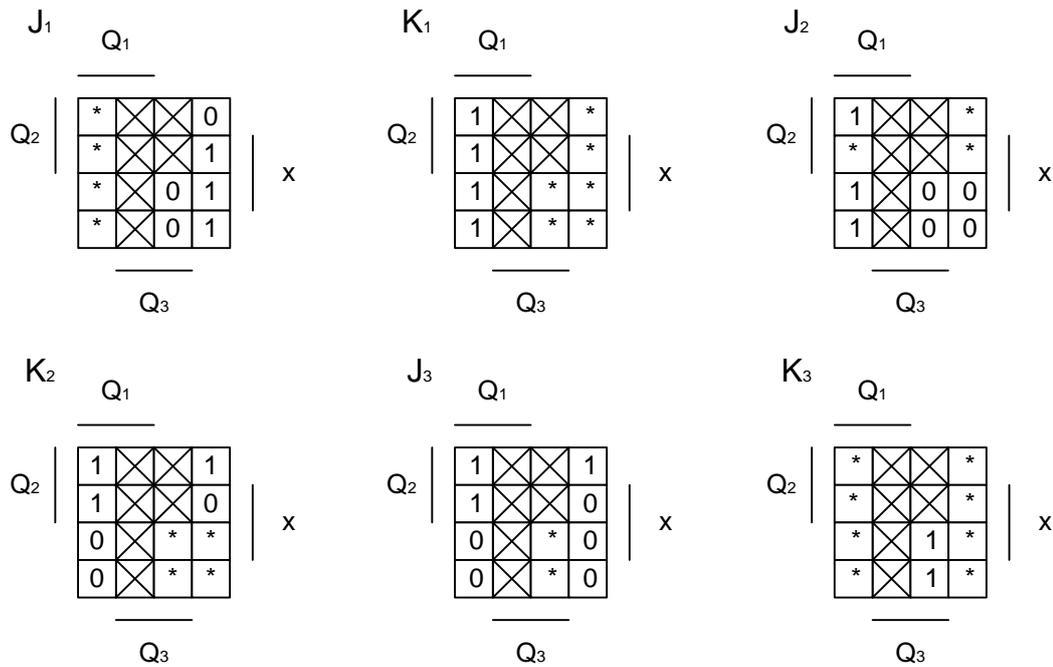


Рис. 30. Карты Карно для минимизации функций возбуждения

На картах Карно знаком "крест" отмечены клетки, соответствующие неиспользуемым кодам состояний; разумеется, они могут включаться в контуры наряду со звездочками, обозначающими неопределенные значения сигналов.

Результаты минимизации представлены в базисе $\{\&, \neg\}$ следующими выражениями:

$$J_1 = \overline{x} Q_2 \overline{Q}_3; K_1 = 1;$$

$$J_2 = Q_1; K_2 = x \overline{Q}_1 Q_2;$$

$$J_3 = K_2; K_3 = 1.$$

6. Получение минимизированных функций, выражающих выходные сигналы:

$$y_i = y_i(Q_1, Q_2, Q_3); (i = 1, 2, 3).$$

Напомним, что в автомате Мура выходные сигналы зависят явно только от состояния, поэтому переменная x не будет присутствовать в выражениях, описывающих y_i .

Упражнение 2. Построить карты Карно для минимизации функций y_i .

Результаты минимизации представлены в базисе $\{\vee, \&, \neg\}$ выражениями:

$$y_1 = \overline{Q}_2 \overline{Q}_3; y_2 = \overline{Q}_1 Q_2 \vee Q_3; y_3 = Q_1 Q_2.$$

7. Построение структурной схемы автомата.

Возможны два варианта схемы:

а) на основе соотношений, полученных при выполнении шагов 5 и 6 (рис. 31):

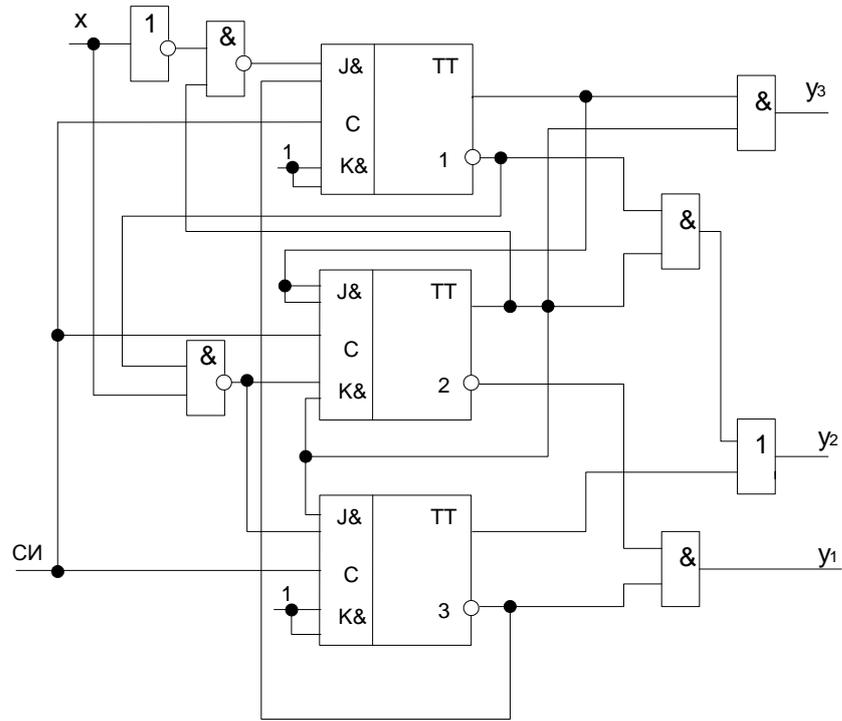


Рис. 31. Схема автомата Мура

б) с использованием дешифратора состояний автомата (рис. 32):

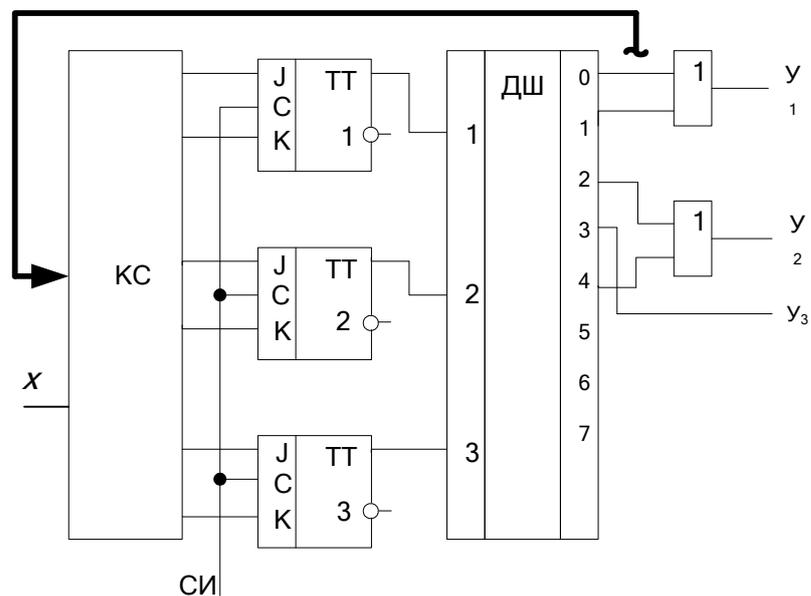


Рис. 32. Построение схемы автомата с использованием дешифратора

В варианте (б) функции возбуждения и выходные функции выражаются через дополнительные переменные, соответствующие состояниям автомата и по этой причине обозначенные символами состояний. Соответствующие выражения записываются на основе анализа строк структурной таблицы:

$$J_1 = a_0 \vee a_2 x; K_1 = a_1 \vee a_3;$$

$$J_2 = a_1; K_2 = a_2 \bar{x} \vee a_3;$$

$$J_3 = K_2; K_3 = a_4.$$

При подаче на вход дешифратора сигналов с прямых выходов триггеров переменные $a_0 - a_4$ присутствуют на выходах дешифратора, обозначенных соответствующими номерами.

Упражнение 3. Изобразить полную комбинационную схему, представленную на рис. 31 условным блоком (КС). В формате канонической структуры автомата это комбинационная схема КС1.

В заключение приведем временную диаграмму действия автомата (рис. 33).

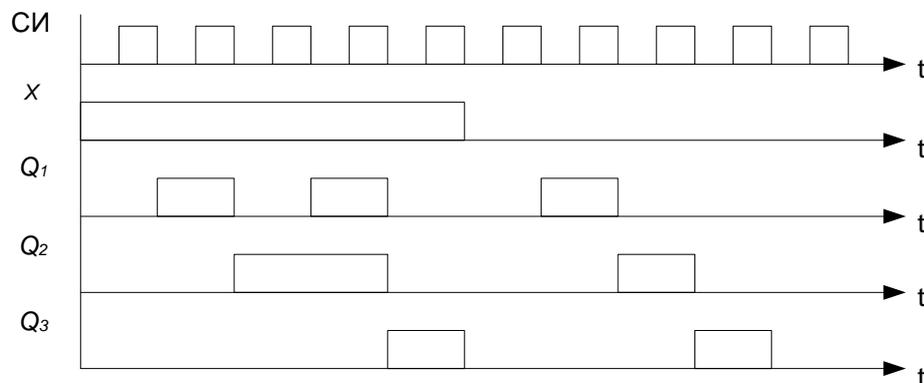


Рис. 33. Временная диаграмма работы автомата Мура

3.6. Счетчики

Счетчиком называется типовой узел ВТ, предназначенный для подсчета числа входных сигналов. Счетчики используются в ЭВМ для образования последовательностей адресов команд, для счета количества циклов выполнения операций и т. п.

Счетчики принято подразделять на суммирующие, вычитающие и реверсивные.

По другой характеристике классификации – *коэффициенту пересчета* K^* – они подразделяются на двоичные ($K^* = 2^m$), десятичные (K^* кратен 10) и счетчики с произвольным коэффициентом пересчета.

Кроме того, счетчики могут быть синхронизируемыми или асинхронными.

Типовые схемы счетчиков приведены в учебной и справочной литературе [2]. Счетчик по построению является автоматом Мура, так как результат подсчета входных сигналов представлен кодом, образованным выходами триггеров, и, следовательно, выходной сигнал зависит явно только от состояния автомата.

В данном разделе мы уделим внимание построению счетчика с произвольным коэффициентом пересчета (на примере, рассчитанном на обобщение).

Пусть необходимо синтезировать счетчик с $K^* = 5$, с использованием JK -триггеров.

Последовательность шагов синтеза любого автомата Мура, в том числе и счетчика, описана выше. В рассматриваемом примере шаги 1 – 3 очевидны, и можно сразу перейти к структурной таблице.

Таблица 10

Структурная таблица для построения счетчика

№ перех.	Сост.	Код состояния			Сост.	Код состояния			Вход. сигналы	Сигналы возбуждения						Выход. сигналы
	a_i	Q_2	Q_1	Q_0	a_{i+1}	Q_2	Q_1	Q_0		J_2	K_2	J_1	K_1	J_0	K_0	
1	a_0	0	0	0	a_1	0	0	1	СИ	0	*	0	*	1	*	Выходной код счетчика
2	a_1	0	0	1	a_2	0	1	0		0	*	1	*	*	1	
3	a_2	0	1	0	a_3	0	1	1		0	*	*	0	1	*	
5	a_3	0	1	1	a_4	1	0	0		1	*	*	1	*	1	
6	a_4	1	0	0	a_0	0	0	0		*	1	0	*	0	*	

Обратим внимание на столбцы, относящиеся к входным и выходным сигналам. В нашем случае в качестве подсчитываемых сигналов используются синхриимпульсы.

Минимизация функций возбуждения по картам Карно приводит к следующим результатам (рассматривать этот шаг как самостоятельное **Упражнение 4**):

$$J_0 = K_0 = \bar{Q}_2; J_1 = K_1 = Q_0; J_2 = Q_0 Q_1; K_2 = 1.$$

На основе полученных выражений строится схема счетчика (рис. 34). Временная диаграмма работы счетчика показана на рис. 35.

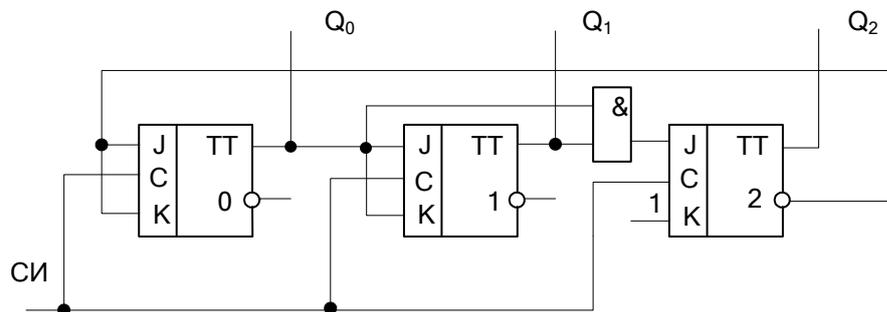


Рис. 34. Схема счетчика

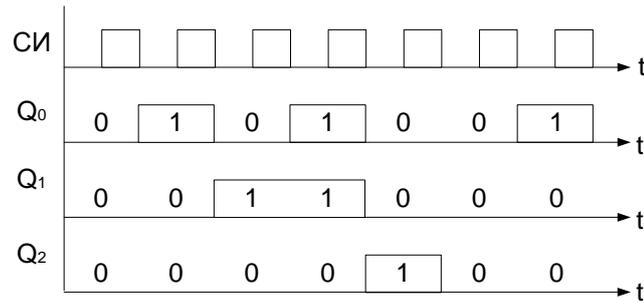


Рис. 35. Временная диаграмма работы счетчика

3.7. Кодирование состояний автомата. Пример синтеза автомата Мили

При переходе от абстрактного автомата, заданного графом, к структурному, возникает необходимость двоичного кодирования состояний. Выбранный способ кодирования существенно влияет на сложность КС синтезируемого автомата. Универсального способа оптимального кодирования для произвольного автомата в теории не существует. Однако имеется ряд практических способов кодирования состояний, которые используются для упрощения комбинационной части синтезируемых схем.

При синтезе автоматов на D -триггерах установлена целесообразность использования следующего алгоритма кодирования состояний.

1. Каждому состоянию q_i автомата сопоставляется целое число N_i , равное числу переходов в это состояние, т. е. числу входящих дуг в соответствующую вершину графа ($i = 1, \dots, n$).

2. Числа N_1, \dots, N_n упорядочиваются по невозрастанию.

3. Состоянию q_j с наибольшим значением N_j присваивается код $00\dots 0$.

4. Для кодирования следующих состояний в упорядоченном списке сначала используются все коды, содержащие одну единицу, затем все коды, содержащие две единицы, и т. д. Вопрос равноправного выбора кодов для некоторых состояний решается произвольно, при условии, что никакие коды не повторяются.

При синтезе автоматов на триггерах других типов (T , RS , JK) предпочтительным является принцип *соседнего кодирования* состояний. При таком кодировании на каждом из переходов переключается только один из триггеров запоминающей части автомата (напомним, что соседние коды различаются значением только одного разряда), что, в частности, повышает надежность работы схемы за счет устранения *гонок* триггеров (см. ниже). В идеале каждые два состояния, соединенные на графе автомата дугой, должны иметь соседние коды. Добиться этого для большинства схем без их усложнения практически не-

возможно, поэтому целью реального кодирования обычно является решение задачи в приближенном варианте (удобным средством для этого служат карты Карно, в клетках которых размещаются коды состояний). Более подробно этот вопрос раскрыт в методических указаниях М. Ю. Монахова [5], в частности в этом пособии сформулирован критерий эффективности кодирования состояний.

Гонки в автоматах. Термин *гонки* характеризует ситуацию неалгоритмической работы схемы, вызванную различной скоростью переключений триггеров в запоминающей части (ЗЧ) автомата. В качестве примера рассмотрим случай смены состояний автомата, содержащего в ЗЧ три триггера (таблица 11). Во второй строке правой части таблицы 11 приведено неалгоритмическое состояние a'_i автомата, возникшее по причине опережающего переключения второго триггера. Из этого состояния автомат может перейти в состояние a_{i+1} , и тогда результирующий переход будет алгоритмическим, а возникшие гонки *некритическими*, т. е. не нарушающими правильную работу автомата (строка 3). Но *в худшем случае* возможен переход из a'_i в некоторое не согласующееся с функцией перехода состояние, например, в a'_{i+1} (*критические гонки* – строка 4). Последствия такого перехода могут вызвать некорректное функционирование автомата. Эффект гонок иллюстрируется также рисунком 36.

Таблица 11

Алгоритмический переход

Q_1	Q_2	Q_3	Сост.
0	0	0	a_i
1	1	1	a_{i+1}

Неалгоритмический переход

Q_1	Q_2	Q_3	Сост.
0	0	0	a_i
0	1	0	a'_i
1	1	1	a_{i+1}
0	1	1	a'_{i+1}

Способы борьбы с критическими гонками:

1. Соседнее кодирование состояний.
2. Специальное противогоночное кодирование [5], приводящее обычно к увеличению числа состояний автомата и, следовательно, к его усложнению.
3. Применение двухтактных схем. Эти схемы обеспечивают завершение переходных процессов при переключениях триггеров в течение действия синхросигнала.

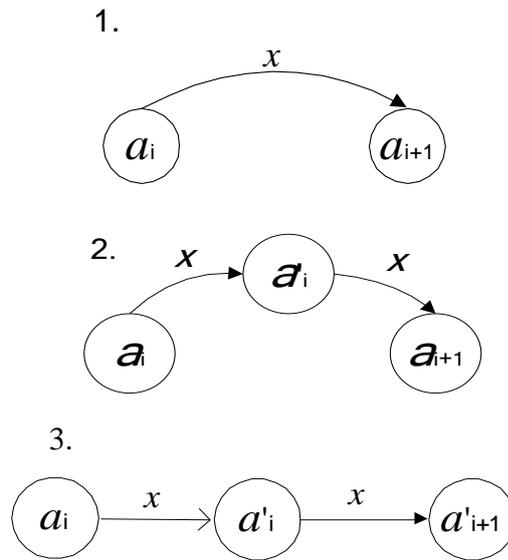


Рис. 36. Эффект гонок в автоматах:

- 1) алгоритмический переход; 2) некритические гонки; 3) критические гонки

Заметим, что автомат Мили в общем случае в большей степени подвержен влиянию гонок по причине зависимости выходного сигнала не только от состояния, но и от входного сигнала который может измениться в течение такта. Поскольку выходной сигнал в автомате Мили вырабатывается при переходе из одного состояния в другое, эффективным средством повышения надежности схемы служит “стробирование” выходного сигнала достаточно узким импульсом, т. е. ограничение времени присутствия сигнала на выходе схемы.

Пример синтеза автомата Мили. Построим несложный автомат Мили по заданному графу. Целью рассмотрения примера являются особенности, присущие этому виду автомата, а также использование принципа кодирования состояний, изложенного в данном разделе.

Исходный граф представлен рисунком 37.

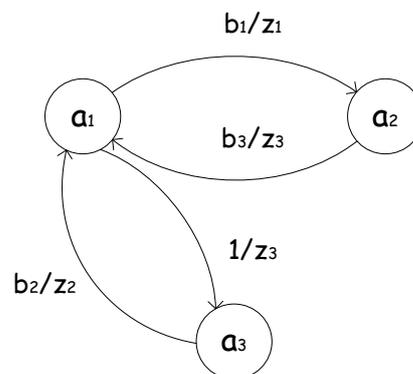


Рис. 37. Граф автомата Мили

Замечание. Если автомат является не полностью определенным, то подразумевается, что он не изменяет состояния под действием сигналов, при которых переход не определен.

Выполним стандартные шаги синтеза.

1. Определение числа триггеров.

В данном примере $n = \lceil \log_2 4 \rceil = 2$.

2. Определение типа триггеров.

Используем D -триггеры.

3. Кодирование состояний автомата.

Закодируем состояния автомата двухразрядными двоичными числами с использованием принципа, сформулированного для D -триггеров.

Одновременно применим двоичное кодирование входных и выходных сигналов согласно каноническому методу структурного синтеза. Выбор кодов выполним исходя из соображений, подобных принятым при кодировании состояний: чем чаще встречается некоторый символ, тем больше нулей присутствует в его коде. Таблицы кодирования, сведенные в общую таблицу 12, приведены ниже.

Таблица 12

Кодирование состояний и сигналов автомата

	Q_1	Q_2		x_1	x_2		y_1	y_2
a_1	0	0	b_1	0	0	z_1	0	1
a_2	0	1	b_2	1	0	z_2	1	0
a_3	1	0	b_3	0	1	z_3	0	0

Заметим, что для рассматриваемого автомата исходные двоичные входные и выходные сигналы можно оставить и без перекодирования; канонический метод для этой цели актуален при достаточно мощных алфавитах абстрактного автомата.

4. Составление структурной таблицы.

Таблица 13

Структурная таблица

№ перех.	Сост. a_i	Код состояния		Сост. a_{i+1}	Код состояния		Вход. наборы	Сигналы возбуждения		Выход. сигналы	
		Q_1	Q_2		Q_1	Q_2		D_1	D_2	y_1	y_2
1	a_1	0	0	a_2	0	1	$\bar{x}_1 \bar{x}_2$	0	1	0	1
2	a_2	0	1	a_1	0	0	$\bar{x}_1 x_2$	0	0	0	0
3	a_1	0	0	a_3	1	0	$x_1 \bar{x}_2$	1	0	1	0
4	a_3	1	0	a_1	0	0	1	0	0	0	0

5. Получение функций возбуждения триггеров.

На основе анализа структурной таблицы искомые функции записываются в виде:

$$D_1 = a_1 x_1 \bar{x}_2 = \bar{Q}_1 \bar{Q}_2 x_1 \bar{x}_2;$$

$$D_2 = a_1 \bar{x}_1 \bar{x}_2 = \bar{Q}_1 \bar{Q}_2 \bar{x}_1 \bar{x}_2.$$

Подобные простые выражения не требуют минимизации.

6. Получение функций, выражающих выходные сигналы.

Из таблицы следует, что в данном частном случае (не обобщать этот факт!):

$$y_1 = D_1 \text{ и } y_2 = D_2.$$

7. Построение структурной схемы автомата (рис. 38).

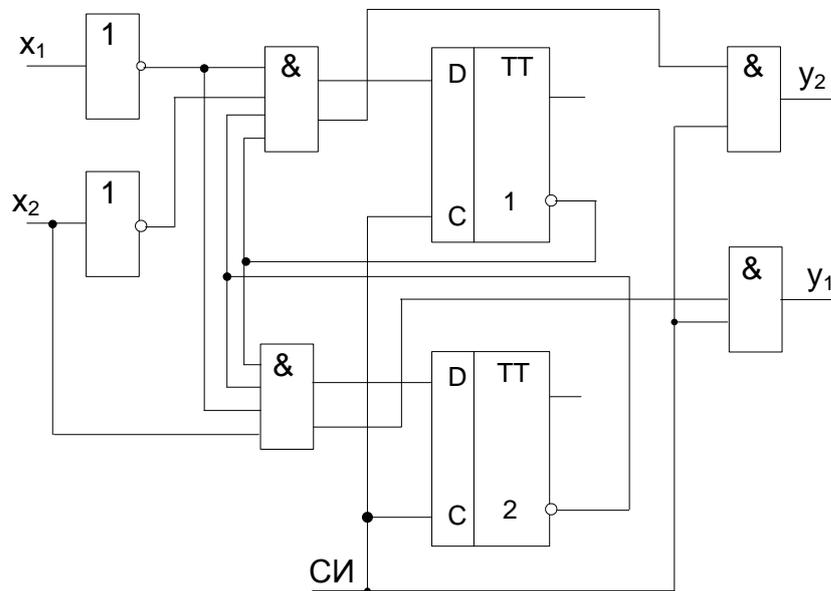


Рис. 38. Схема автомата Мили

Упражнение 5. Построить автомат (рис. 37) с использованием *T*-триггеров и дешифратора состояний.

ЗАДАЧИ И УПРАЖНЕНИЯ

Упражнения 1 – 5, сформулированные в разделах 3.4, 3.5, 3.6, 3.7.

Литература

1. Майоров С. А., Новиков Г. И. Структура электронных вычислительных машин. – Л., «Машиностроение», 1979. - 134 с.
2. Каган Б. М. Электронные вычислительные машины и системы. – М., "Энергия", 1979. - 527 с.

3. Савельев А. Я. Прикладная теория цифровых автоматов. *Учебник для ВУЗов* – М., ВШ, 1987. - 270 с.
4. Карпов Ю. Г. Теория автоматов. *Учебник для ВУЗов*. – СПб.: ПИТЕР, 2002. - 206 с.
5. Монахов М. Ю. Прикладная теория цифровых автоматов. Методические указания к курсовой работе. – Владимир, ВПИ, 1991. - 44 с.