

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет радиотехники и электроники

Кафедра информационных радиотехнологий

А. А. Будько, Т. Н. Дворникова

**СИНТЕЗ ЦИФРОВЫХ АВТОМАТОВ.
СХЕМОТЕХНИКА ЦИФРОВЫХ УСТРОЙСТВ.
КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

*Рекомендовано УМО по образованию в области информатики и
радиоэлектроники в качестве пособия для специальности
1-36 04 02 «Промышленная электроника»*

Минск БГУИР 2024

УДК 004.312(076)
ББК 32.973я73
Б90

Рецензенты:

кафедра инфокоммуникационных технологий учреждения образования
«Белорусская государственная академия связи»
(протокол № 3 от 11.10.2022);

заведующий кафедрой электропривода и автоматизации промышленных
установок и технологических комплексов
Белорусского национального технического университета
кандидат технических наук, доцент С. А. Павлюковец;

профессор кафедры бизнес-анализа и математического моделирования
Международного университета «МИТСО»
кандидат физико-математических наук,
доцент Г. Е. Хурсевич

Будько, А. А

Б90

Синтез цифровых автоматов. Схемотехника цифровых устройств.
Курсовое проектирование : пособие / А. А. Будько, Т. Н. Дворникова. –
Минск : БГУИР, 2024. – 82 с. : ил.
ISBN 978-985-543-737-7.

Содержит два задания, которые состоят из 48 и 28 вариантов соответственно, и предназначено для получения практических навыков абстрактного и структурного синтеза цифровых автоматов. Содержит краткие теоретические сведения, а также примеры и ссылки на учебную и справочную литературу.

Может быть использовано для проведения практических и лабораторных работ по схемотехнике цифровых устройств для специальностей «Радиоинформатика», «Медицинская электроника», «Моделирование и компьютерное проектирование радиоэлектронных средств».

УДК 004.312(076)
ББК 32.973я73

ISBN 978-985-543-737-7

© Будько А. А., Дворникова Т. Н., 2024
© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2024

СОДЕРЖАНИЕ

1 Элементарные цифровые автоматы	4
1.1 Введение	4
1.2 Асинхронный <i>RS</i> -триггер (<i>SR-latch</i>)	5
1.3 Синхронный <i>RS</i> -триггер.....	11
1.4 Установка и сброс	15
1.5 <i>D</i> -триггер	16
1.6 Синтез <i>JK</i> -триггера	18
1.7 <i>JK</i> -триггер <i>M-S</i> -конфигурации с инвертором.....	21
1.8 <i>T</i> -триггер	23
1.9 Преобразование триггера одного типа в триггер другого типа	23
1.10 Триггеры с динамическим управлением	28
2 Типовые последовательностные схемы.....	31
2.1 Регистры и их применение.....	31
2.2 Цифровые счетчики	39
2.2.1 Асинхронные счетчики	40
2.2.2 Реверсивные счетчики.....	44
2.2.3 Счетчики с произвольным модулем счета	44
2.2.4 Интегральные схемы асинхронных счетчиков.....	46
2.2.5 Синхронные счетчики	50
2.2.6 Синтез синхронных счетчиков.....	51
2.2.7 Интегральные схемы синхронных счетчиков	54
3 Синтез последовательностных схем (конечных цифровых автоматов).....	61
3.1 Введение	61
3.2 Минимизация абстрактного автомата	66
3.3 Минимизация автоматов Мура.....	69
3.4 Задание на курсовое проектирование	79
Задание 1.....	79
Задание 2.....	80
Список использованных источников	81

1 ЭЛЕМЕНТАРНЫЕ ЦИФРОВЫЕ АВТОМАТЫ

1.1 Введение

Цифровые устройства делятся на две большие группы:

- комбинационные устройства (КУ);
- последовательностные устройства (ПУ).

Сигнал на выходе КУ в любой дискретный момент времени зависит от значения входных сигналов в этот же момент времени.

Сигнал на выходе ПУ в любой дискретный момент времени зависит от значений входных сигналов в этот же момент времени и также от значений входных и выходных сигналов в предыдущие дискретные моменты времени.

Простейшими последовательностными схемами или элементарными конечными автоматами являются триггеры.

Триггером называется такое устройство, которое имеет два устойчивых состояния и способное оставаться в любом из этих состояний неопределенный период времени. Триггер будет менять состояние, только когда управляющий сигнал будет подан на его входы. Если триггер изменил свое состояние, то он будет оставаться в новом состоянии до тех пор, пока другой управляющий сигнал не вернет триггер в исходное состояние.

Триггер имеет два выхода, которые обычно обозначаются как Q и \bar{Q} , поскольку логическое состояние одного выхода всегда инверсно (противоположно) логическому состоянию другого выхода.

Когда триггер в состоянии $Q = 1$ и $\bar{Q} = 0$, то говорят, что триггер установлен, соответственно, когда триггер в состоянии $Q = 0$ и $\bar{Q} = 1$, то говорят, что триггер сброшен.

Триггеры являются базовыми элементами последовательностных логических устройств (цифровых конечных автоматов). Используя триггеры и комбинационные устройства, можно построить любую последовательностную схему. Наиболее важными или типовыми схемами являются регистры и счетчики.

Триггеры классифицируются по логическому функционированию и по способу записи информации.

По логическому функционированию основными являются RS -, D -, JK - и T -триггеры.

По способу записи информации триггеры делятся на асинхронные (нетактируемые) и синхронные (тактируемые). В асинхронных триггерах запись информации (переключение триггера в новое состояние) происходит под действием изменений входных управляющих сигналов с момента подачи их на информационные входы. В синхронных триггерах запись информации происходит только в течение подачи синхронизирующих импульсов.

Различают синхронные триггеры со статическим и динамическим управлением записью информации.

Синхронный триггер со статическим управлением (*Latch*) изменяет свое состояние в зависимости от состояния управляющих входов во время действия активного уровня сигнала синхронизации.

Синхронный триггер с динамическим управлением (*Flip-Flop, F-F*) изменяет свое состояние в зависимости от состояния управляющих входов в момент изменения синхронизирующего сигнала.

Вход синхронизации *Clock (C, CK)* – называется прямым динамическим, если переключение триггера осуществляется положительным перепадом импульсов синхронизации, и инверсным динамическим, если переключение триггера осуществляется отрицательным перепадом импульсов синхронизации. В остальное время действия сигнала синхронизации триггер не реагирует на состояния управляющих сигналов.

1.2 Асинхронный *RS*-триггер (*SR-latch*)

Асинхронный *RS*-триггер – это устройство с двумя устойчивыми состояниями, имеющее два входа: *Set (S)* и *Reset (R)* и два выхода: Q и \bar{Q} . Асинхронный *RS*-триггер (*SR-latch*) функционирует в соответствии с таблицей истинности, или характеристической таблицей (рисунок 1.1).

t_n			t_{n+1}	
Q_n	S_n	R_n	Q_{n+1}	
0	0	0	0	Хранение информации Нет изменений в состоянии триггера
1	0	0	1	
0	1	0	1	Установка в «1»
1	1	0	1	
0	0	1	0	Сброс в «0»
1	0	1	0	
0	1	1	x	Неопределенность
1	1	1	x	

Рисунок 1.1 – Истинность *RS*-триггера

Если входы *S* и *R* оба в состоянии логического нуля, то триггер не изменяет свое состояние и говорят, что он хранит один бит информации.

Когда $S_n = 1$, а $R_n = 0$, то триггер изменит свое состояние, если он был в состоянии $Q_n = 0$ в $Q_{n+1} = 1$, и не будет изменять свое состояние, если он в состоянии $Q_n = 1$.

Когда $S_n = 0$, а $R_n = 1$, то триггер изменит свое состояние, если он был в состоянии $Q_n = 1$ в $Q_{n+1} = 0$, и не будет изменять свое состояние, если он был в состоянии $Q_n = 0$ и $Q_n = 1$.

Таким образом, комбинация $S_n = 1, R_n = 0$ всегда будет приводить схему в состояние $Q_{n+1} = 1$, а комбинация $S_n = 0, R_n = 1$ всегда будет приводить схему в состояние $Q_{n+1} = 0$ независимо от исходного состояния триггера.

И если $S_n = R_n = 1$, триггер может изменить состояние, а может не изменить, т. е. эта операция – «неопределенность».

Для синтеза RS -триггера информация из рисунка 1.1 представляется на карте Карно (рисунок 1.2).

		$S_n R_n$			
		00	01	11	10
Q_n	0	0	0	x	1
	1	1	0	x	1

Рисунок 1.2 – Карта Карно для синтеза RS -триггера

Из этой карты Карно логические выражения для выходов RS -триггера могут быть получены в конъюнктивной нормальной форме (КНФ). На карте Карно сплошной линией показано группирование логических нулей для Q_{n+1} и штриховой – для \bar{Q}_{n+1} :

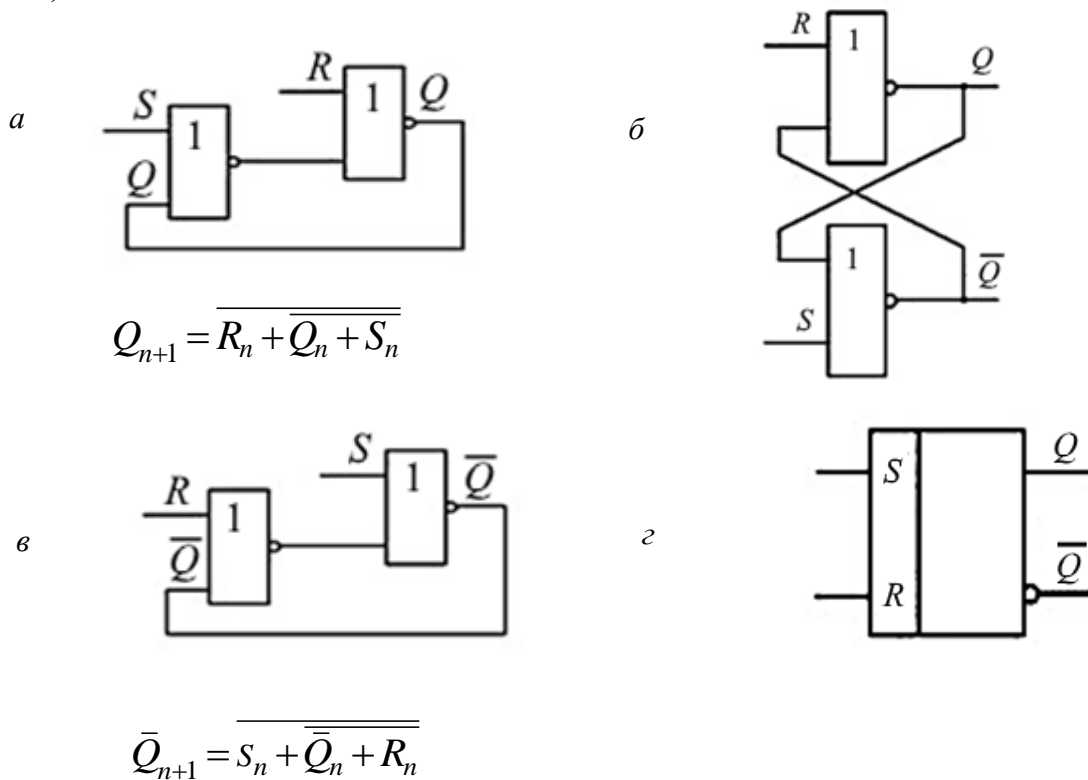
$$Q_{n+1} = \bar{R}_n \cdot (Q_n + S_n), \quad \bar{Q}_{n+1} = \bar{S}_n \cdot (\bar{Q}_n + R_n). \quad (1.1)$$

Эти выражения могут быть использованы для реализации RS -триггера на элементах ИЛИ – НЕ. Для этого дважды проинвертируем их и используем закон Де Моргана:

$$Q_{n+1} = \overline{\overline{\bar{R}_n \cdot (Q_n + S_n)}} = \overline{R_n + \bar{Q}_n + \bar{S}_n},$$

$$\bar{Q}_{n+1} = \overline{\overline{\bar{S}_n \cdot (\bar{Q}_n + R_n)}} = \overline{S_n + Q_n + R_n}. \quad (1.2)$$

Рассмотрим построение RS -триггера на элементах ИЛИ – НЕ (рисунок 1.3).



$a, б, в$ – построение RS -триггера по выражениям (см. формулу (1.2));
 $г$ – угловое обозначение RS -триггера

Рисунок 1.3 – Асинхронный RS -триггер на элементах ИЛИ – НЕ

Рассмотрим работу RS -триггера с использованием временных диаграмм (рисунок 1.4).

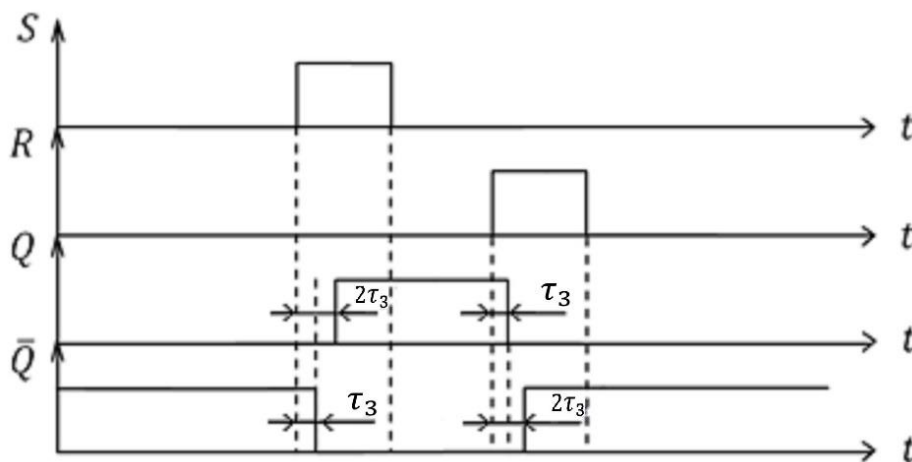


Рисунок 1.4 – Временные диаграммы работы RS -триггера

При комбинации $S_n = R_n = 1$ на обоих выходах триггера $Q_{n+1} = 0$ и $\bar{Q}_{n+1} = 0$ и далее при переходе триггера в режим хранения $S_n = R_n = 0$ триггер

установится в какое-то состояние: $Q=1, \bar{Q}=0$ или $\bar{Q}=0, \bar{Q}=1$, но в какое – неизвестно, поэтому входной комбинации $S_n = R_n = 1$ соответствует операция «неопределенность».

Временные диаграммы с рисунка 1.4 могут быть использованы для определения максимальной частоты переключения RS -триггера:

$$f_{\max} = \frac{1}{2\tau_3}, \quad (1.3)$$

где τ_3 – задержка логического элемента ИЛИ – НЕ.

Однако при такой частоте на выходах триггера длительность уровня логического нуля или единицы будет только τ_3 , а для передачи информации в последующую цепь длительность информационных нуля и единицы должна быть не менее $2\tau_3$, поэтому максимальная рабочая частота определяется как

$$f_{\max \text{ раб}} = \frac{1}{3\tau_3}. \quad (1.4)$$

При изучении работы RS -триггера и при синтезе других триггеров используется таблица переходов RS -триггера, которая определяет комбинации входных сигналов, необходимые для того или иного перехода триггера, т. е. триггер находится в каком-то исходном состоянии и определяются состояния входов, чтобы триггер перешел в определенное следующее состояние (состояние перехода). Рассмотрим таблицу 1.1.

Таблица 1.1 – Таблица переходов RS -триггера

Исходное состояние	Состояние перехода	Требуемые входы	
Q_n	Q_{n+1}	S_n	R_n
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Для реализации RS -триггера на элементах И – НЕ необходимо извлечь из карты Карно выражения для Q и \bar{Q} в дизъюнктивную нормальную форму (ДНФ) (рисунок 1.5), затем преобразовать их, используя двойную инверсию и закон Де Моргана:

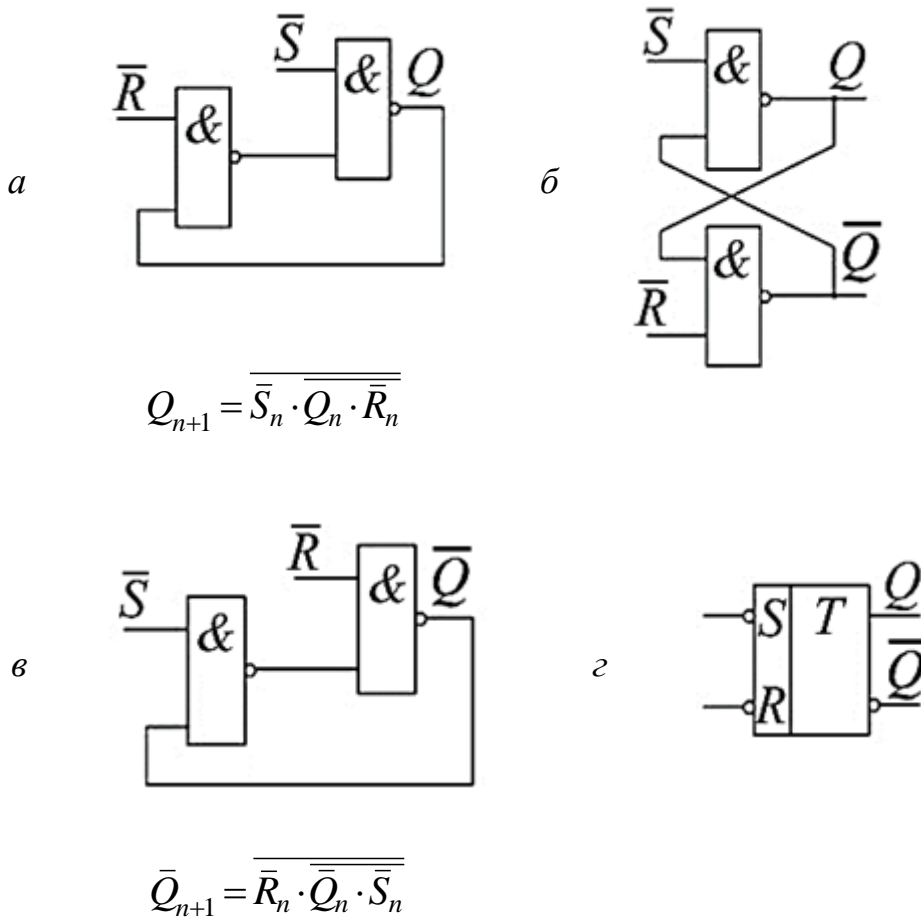
$$Q_{n+1} = S_n + Q_n \cdot \bar{R}_n = \overline{\overline{S_n + Q_n \cdot \bar{R}_n}} = \overline{\bar{S}_n \cdot \overline{Q_n \cdot \bar{R}_n}},$$

$$\bar{Q}_{n+1} = R_n + \bar{Q}_n \cdot \bar{S}_n = \overline{\overline{R_n + \bar{Q}_n \cdot \bar{S}_n}} = \overline{\bar{R}_n \cdot \overline{\bar{Q}_n \cdot \bar{S}_n}}. \quad (1.5)$$

	$S_n R_n$			
	00	01	11	10
Q_n	0	0	x	1
	1	0	x	1

Рисунок 1.5 – Карта Карно для синтеза \overline{RS} -триггера

Рассмотрим построение RS -триггера на элементах И – НЕ (рисунок 1.6).



$a, б, в$ – построение триггера; $г$ – условное обозначение
Рисунок 1.6 – Асинхронный \overline{RS} -триггер на элементах И – НЕ

Такой триггер, т. е. триггер на элементах И – НЕ, имеет инверсные входы, будем обозначать его как \overline{RS} -триггер. Таблица истинности, или характеристическая таблица, \overline{RS} -триггера представлена на рисунке 1.7.

t_n			t_{n+1}	
Q_n	S_n	R_n	Q_{n+1}	
0	1	1	0	Хранение информации
1	1	1	1	
0	0	1	1	Установка в «1»
1	0	1	1	
0	1	0	0	Сброс в «0»
1	1	0	0	
0	0	0	x	Неопределенность
1	0	0	x	

Рисунок 1.7 – Истинность \overline{RS} -триггера

Рассмотрим работу \overline{RS} -триггера, используя временные диаграммы (рисунок 1.8) и таблицу переходов (таблица 1.2).

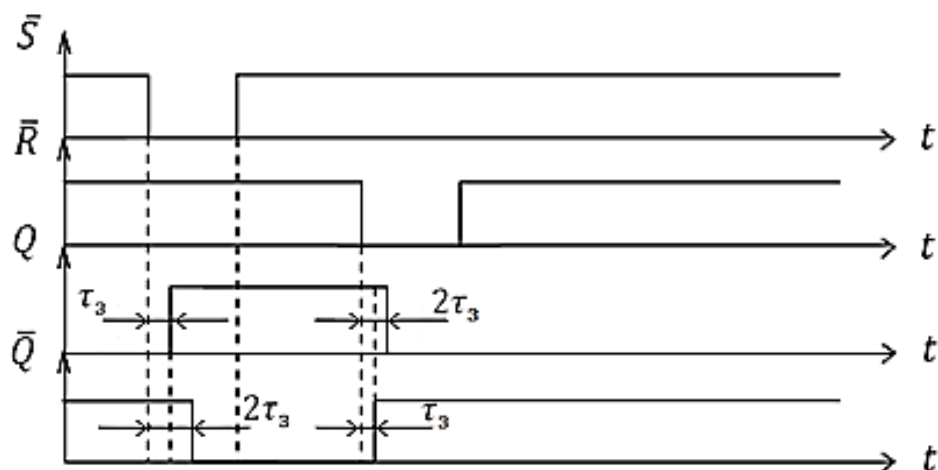


Рисунок 1.8 – Временные диаграммы работы \overline{RS} -триггера

Таблица 1.2 – Таблица переходов \overline{RS} -триггера

Исходное состояние	Состояние перехода	Требуемые входы	
		\overline{S}_n	\overline{R}_n
Q_n	Q_{n+1}		
0	0	1	x
0	1	0	1
1	0	1	0
1	1	x	1

1.3 Синхронный *RS*-триггер

Последовательностные схемы классифицируются как асинхронные и синхронные. Последовательностные схемы, чье поведение зависит от последовательности, в которой изменяются входные сигналы, называются асинхронными. Выходы таких схем изменяются, как только изменяются входы.

Последовательностные схемы, чье поведение определяется состоянием входов в дискретные моменты времени, называются синхронными.

В этих системах элементы памяти могут менять состояние только в дискретные моменты времени. Синхронизация достигается с помощью устройств, которые называются системами синхронизации и генерируют периодические последовательности импульсов (рисунок 1.9).

$$Q = \frac{T}{t_u} - \text{скважность,}$$

$$K_3 = \frac{t_u}{T} - \text{коэффициент заполнения.}$$

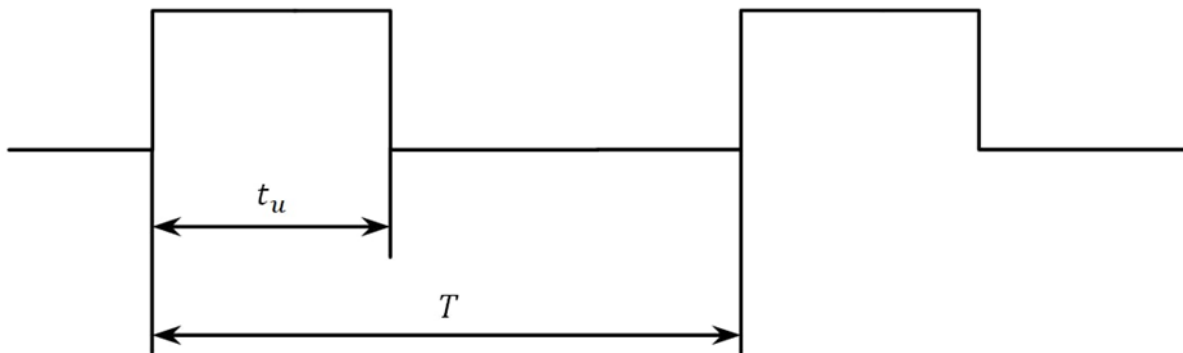


Рисунок 1.9 – Последовательность синхроимпульсов

Для синтеза *RS*-триггера построим таблицу истинности (таблица 1.3).

Таблица 1.3 – Таблица истинности синхронного RS -триггера

t_n				t_{n+1}
C_n	Q_n	S_n	R_n	Q_{n+1}
0	0	0	0	0
0	1	0	0	1
0	0	1	0	0
0	1	1	0	1
0	0	0	1	0
0	1	0	1	1
0	0	1	1	0
0	1	1	1	1
1	0	0	0	0
1	1	0	0	1
1	0	1	0	1
1	1	1	0	1
1	0	0	1	0
1	1	0	1	0
1	0	1	1	x
1	1	1	1	x

В таблице 1.3 по сравнению с таблицей истинности асинхронного RS -триггера добавлен столбец C_n для значений синхроимпульсов. Активным уровнем синхропоследовательности является высокий уровень или уровень логической единицы, поэтому эта часть таблицы соответствует таблице истинности асинхронного RS -триггера. При значениях C_n , равных логическому нулю, значение выхода триггера не изменяется. И поэтому вход C_n можно рассматривать как разрешающий.

Для минимизации выражений Q_{n+1} и \bar{Q}_{n+1} строим карту Карно (рисунок 1.10).

		$S_n R_n$			
		00	01	11	10
$C_n Q_n$	00	0	0	0	1
	01	1	1	1	1
	11	1	0	x	1
	10	0	0	x	1

Рисунок 1.10 – Карта Карно для синтеза синхронного RS -триггера

На карте Карно покрытие для функции Q_{n+1} показано сплошной линией, а для \bar{Q}_{n+1} – штриховой.

$$Q_{n+1} = \bar{C}_n \cdot Q_n + Q_n \cdot \bar{R}_n + C_n \cdot S_n,$$

$$\bar{Q}_{n+1} = \bar{C}_n \cdot \bar{Q}_n + \bar{Q}_n \cdot \bar{S}_n + C_n \cdot R_n. \quad (1.6)$$

Для построения синхронного RS -триггера используем выражение Q_{n+1} , но, поскольку в этом выражении Q_n присутствует в двух термах, его сначала модернизируем, а затем используем двойную инверсию и закон Де Моргана.

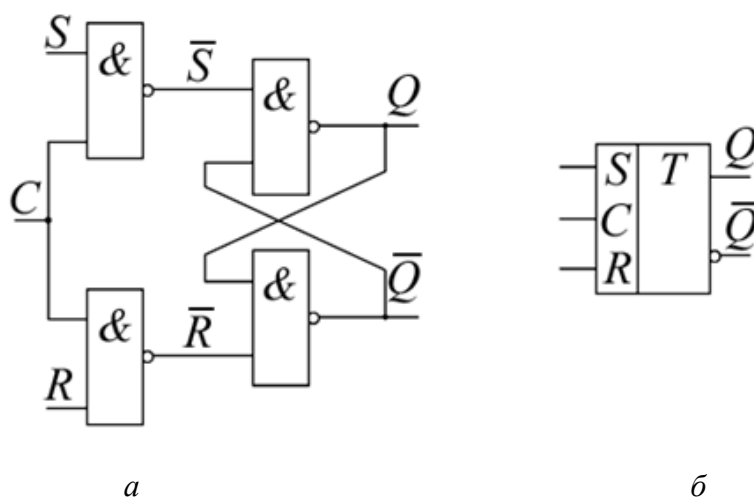
$$Q_{n+1} = Q_n(\bar{C}_n + \bar{R}_n) + C_n \cdot S_n \text{ – дистрибутивный закон,}$$

$$Q_{n+1} = Q_n \cdot \overline{C_n \cdot R_n} + C_n \cdot S_n \text{ – закон Де Моргана,}$$

$$Q_{n+1} = \overline{\overline{Q_n \cdot C_n \cdot R_n} + C_n \cdot S_n} \text{ – правило двойного отрицания,}$$

$$Q_{n+1} = \overline{\overline{Q_n \cdot C_n \cdot R_n} \cdot \overline{C_n \cdot S_n}}. \quad (1.7)$$

По выражению (1.7) строим синхронный RS -триггер (рисунок 1.11).



a – электрическая схема; *б* – условное обозначение

Рисунок 1.11 – Синхронный RS -триггер со статическим управлением

Прежде чем изучить работу синхронного RS -триггера с помощью временных диаграмм, рассмотрим другой подход при синтезе синхронного RS -триггера, который является общим не только при синтезе данного триггера. Синтез на основе асинхронного \overline{RS} -триггера заключается в определении выражений для инверсных входов \bar{S} и \bar{R} . Для этого используем таблицу переходов \overline{RS} -триггера и получим таблицу истинности для функций \bar{S}_n и \bar{R}_n для переходов в таблице истинности синхронного RS -триггера (таблица 1.4).

Таблица 1.4 – Таблица истинности синхронного

RS -триггера со значениями входов \bar{S}_n и \bar{R}_n

t_n				t_{n+1}		
C_n	Q_n	S_n	R_n	Q_{n+1}	\bar{S}_n	\bar{R}_n
0	0	0	0	0	1	x
0	1	0	0	1	x	1
0	0	1	0	0	1	x
0	1	1	0	1	x	1
0	0	0	1	0	1	x
0	1	0	1	1	x	1
0	0	1	1	0	1	x
0	1	1	1	1	x	1
1	0	0	0	0	1	x
1	1	0	0	1	x	1
1	0	1	0	1	0	1
1	1	1	0	1	x	1
1	0	0	1	0	1	x
1	1	0	1	0	1	0
1	0	1	1	x	x	x
1	1	1	1	x	x	x

Минимизируем функции \bar{S}_n^* и \bar{R}_n^* с помощью карт Карно и строим синхронный RS -триггер (рисунок 1.12).

		$S_n R_n$			
		00	01	11	10
$C_n Q_n$	00	1	1	1	1
	01	x	x	x	x
	11	x	1	x	x
	10	1	1	x	0

		$S_n R_n$			
		00	01	11	10
$C_n Q_n$	00	x	x	x	x
	01	1	1	1	1
	11	1	0	x	1
	10	x	x	x	1

$$\bar{S}_n^* = \bar{C}_n + \bar{S}_n = \overline{C_n \cdot S_n}$$

$$\bar{R}_n^* = \bar{C}_n + \bar{R}_n = \overline{C_n \cdot R_n}$$

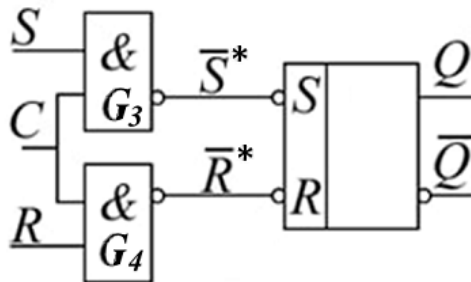


Рисунок 1.12 – Синхронный RS -триггер со статическим управлением

Теперь проиллюстрируем работу синхронного *RS*-триггера с помощью временных диаграмм (рисунок 1.13).

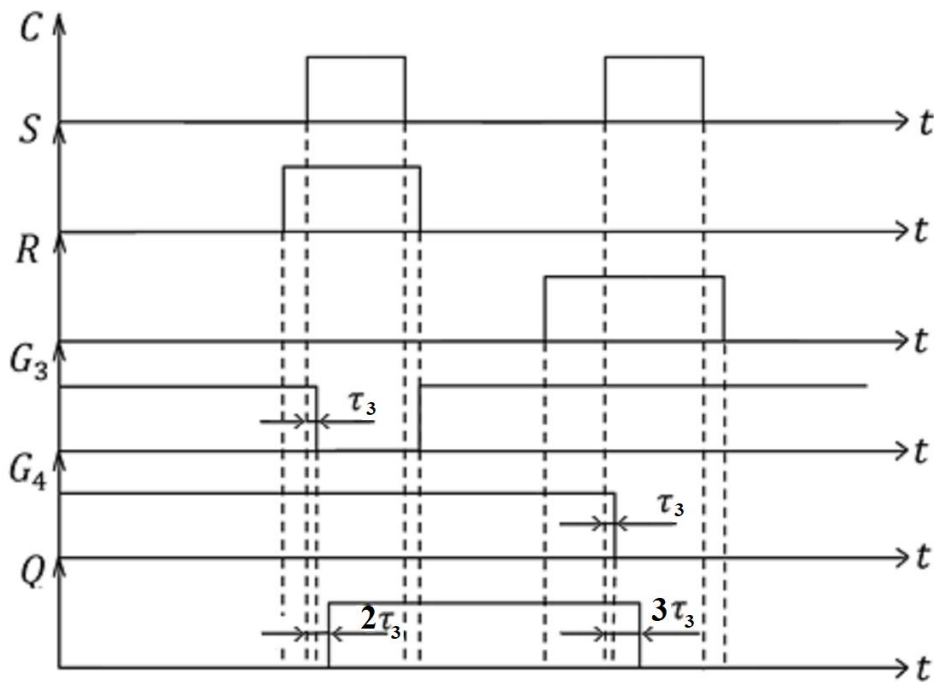


Рисунок 1.13 – Временные диаграммы

Максимальная частота переключения такого триггера $f_{\max} = \frac{1}{3\tau_x}$.

1.4 Установка и сброс

Триггеры, когда включается источник питания, устанавливаются в состояние, которое не является определенным. Это может быть $Q=1$ или $Q=0$. Во многих практических случаях желательно, чтобы триггер изначально был установлен, т. е. $Q=1$, или сброшен, т. е. $Q=0$. Это достигается путем использования специальных прямых или асинхронных входов, которые называются *Preset (Pr)* и *Clear (Cr)*.

Сигналы на эти входы могут быть поданы в любой момент времени между импульсами синхронизации (рисунок 1.14).

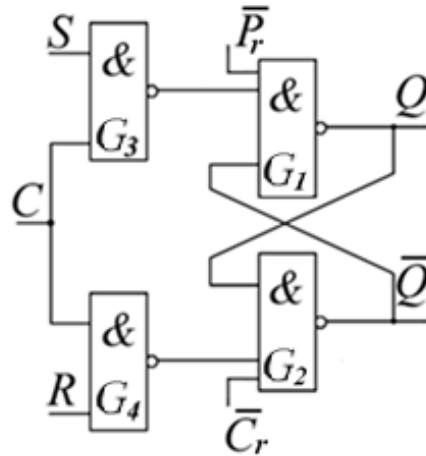


Рисунок 1.14 – Схема синхронного RS -триггера с входами $\bar{P}r$ и $\bar{C}r$

Если $\bar{P}r = 0$ и $\bar{C}r = 1$, выход $G_1(Q)$ будет единица. И, поскольку все три входа G_2 будут равны единице, на выходе $\bar{Q} = 0$. Следовательно, $\bar{P}r = 0$ устанавливает триггер.

Подобным образом, если $\bar{P}r = 1$ и $\bar{C}r = 0$, триггер сбрасывается или устанавливается в нулевое состояние или обнуляется. Таким образом, триггер устанавливается асинхронно и асинхронные входы $\bar{P}r$ и $\bar{C}r$ должны быть соединены с логической единицей прежде, чем импульс синхронизации будет подан на триггер.

Условие $\bar{P}r = \bar{C}r = 0$ не должно использоваться, поскольку приведет к неопределенному состоянию триггера.

1.5 D -триггер

D -триггер имеет два устойчивых состояния и один вход, который называется D . Рассмотрим для него таблицу истинности (таблица 1.5).

Таблица 1.5 – Таблица истинности D -триггера

t_n			t_{n+1}		
C_n	D_n	Q_n	Q_{n+1}	\bar{S}_n	\bar{R}_n
1	0	0	0	1	x
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	x	1
0	0	0	0	1	x
0	0	1	1	x	1
0	1	0	0	1	x
0	1	1	1	x	1

Синтез D -триггера осуществим на базе асинхронного \overline{RS} -триггера, для этого дополняем таблицу 1.5 входами \overline{S}_n и \overline{R}_n и минимизируем их выражения (рисунок 1.15).

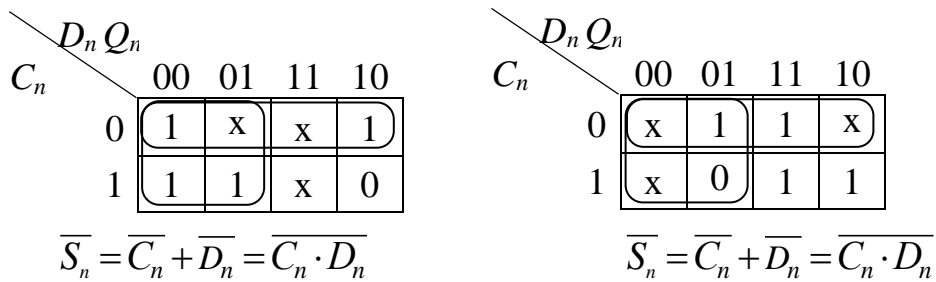


Рисунок 1.15 – Карты Карно для синтеза D -триггера

На основе полученных выражений строим D -триггер (рисунок 1.16).

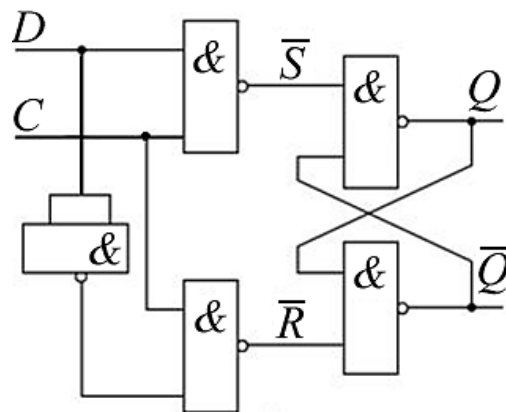


Рисунок 1.16 – Схема синхронного D -триггера со статическим управлением

Схема рисунка 1.16 может быть упрощена, если извлечь выражение для \overline{R}_n не в минимизированном виде (рисунок 1.17).

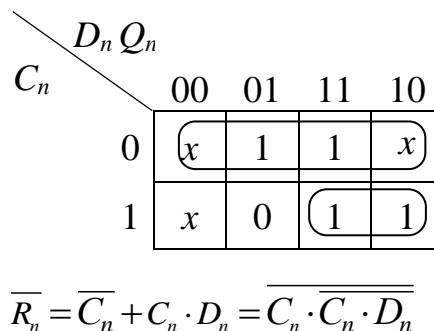


Рисунок 1.17 – Упрощение выражения для \overline{R}_n

Построим D -триггер, используя полученное выражение для \overline{R}_n (рисунок 1.18).

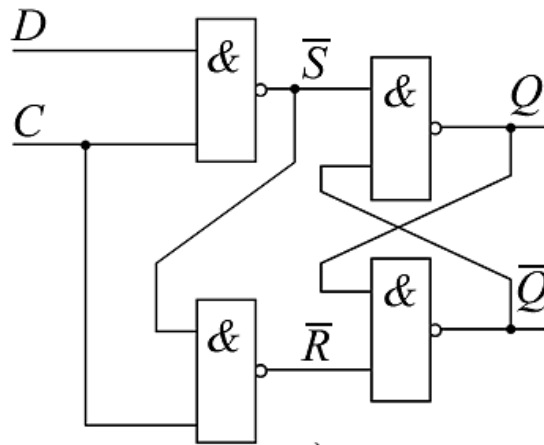


Рисунок 1.18 – Схема D -триггера со статическим управлением

1.6 Синтез JK -триггера

Операция, которую выполняет RS -триггер при входах $S_n = R_n = 1$, является «неопределенностью». Но мы можем построить триггер, который при комбинации входов $S_n = R_n = 1$ устанавливался бы, например, в состояние $Q_{n+1} = 1$. В этом случае такой триггер называется S -триггер. В случае если при комбинации входов $S_n = R_n = 1$ триггер сбрасывается в состояние $Q_n = 0$, такой триггер называется R -триггером. В случае если при комбинации $S_n = R_n = 1$ триггер не меняет свое состояние, он называется E -триггером. И в случае если при комбинации входов $S_n = R_n = 1$ триггер меняет свое состояние на противоположное, он называется JK -триггером.

Таблица 1.6 является таблицей истинности RS -, S -, R -, E - и JK -триггеров.

Таблица 1.6 – Таблица истинности RS -, S -, R -, E - и JK -триггеров

t_n			t_{n+1}				
Q_n	S_n	R_n	Q_{n+1}	Q_{n+1}	Q_{n+1}	Q_{n+1}	Q_{n+1}
0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1
1	1	0	1	1	1	1	1
0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0
0	1	1	x	1	0	0	1
1	1	1	x	1	0	1	0

Рассмотрим синтез синхронного JK -триггера. Таблица истинности, или характеристическая таблица, JK -триггера, представлена таблицей 1.7. Входы S и R в таблице 1.7 обозначены как J и K .

Таблица 1.7 – Таблица истинности JK -триггера

t_n				t_{n+1}		
C_n	Q_n	J_n	K_n	Q_{n+1}	\bar{S}_n	\bar{R}_n
1	0	0	0	0	1	x
1	1	0	0	1	x	1
1	0	1	0	1	0	1
1	1	1	0	1	x	1
1	0	0	1	0	1	x
1	1	0	1	0	1	0
1	0	1	1	1	0	1
1	1	1	1	0	1	0
0	0	0	0	0	1	x
0	1	0	0	1	x	1
0	0	1	0	0	1	x
0	1	1	0	1	x	1
0	0	0	1	0	1	x
0	1	0	1	1	x	1
0	0	1	1	0	1	x
0	1	1	1	1	x	1

Таблица 1.7 дополнена значениями требуемых входов \bar{S}_n и \bar{R}_n триггера \overline{RS} -типа, на базе которого синтезируем JK -триггер. Используя карты Карно (рисунок 1.19), минимизируем выражения для \bar{S}_n и \bar{R}_n и строим схему JK -триггера (рисунок 1.20).

		$J_n K_n$			
		00	01	11	10
$C_n Q_n$	00	x	x	x	x
	01	1	1	1	1
	11	1	0	0	1
	10	x	x	1	1

		$J_n K_n$			
		00	01	11	10
$C_n Q_n$	00	1	1	1	1
	01	x	x	x	x
	11	x	1	1	x
	10	1	1	0	0

$$\bar{S}_n = \bar{C}_n + Q_n + \bar{J}_n = \overline{C_n \cdot \bar{Q}_n J_n}$$

$$\bar{R}_n = \bar{C}_n + \bar{Q}_n + \bar{K}_n = \overline{C_n \cdot \bar{Q}_n K_n}$$

Рисунок 1.19 – Минимизация входов \bar{S}_n и \bar{R}_n

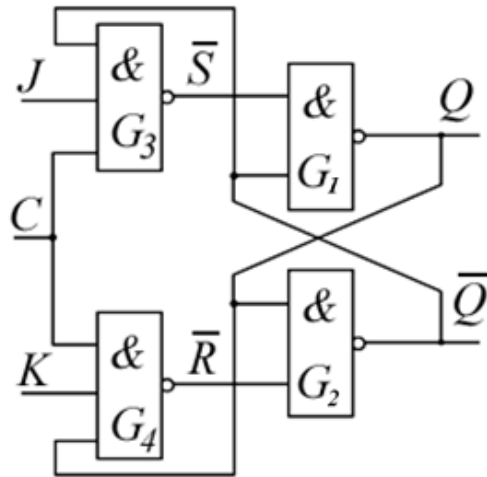


Рисунок 1.20 – Логическая схема JK -триггера

JK -триггер – это по сути модифицированный RS -триггер, в котором неопределенное состояние RS -триггера определено. Входы J и K заменяют входы S и R и осуществляют установку, сброс, хранение и изменение состояния.

Отличие в том, что при $J = K = 1$ JK -триггер переключается в противоположное состояние.

Когда J и K равны логической единице в полученной схеме, импульс синхронизации будет передаваться через один из логических элементов И – НЕ (через G_3 или G_4), один из входов которого соединен с выходом триггера, равным в данный момент логической единице. Если $Q = 1$, выход логического элемента G_4 становится равным нулю, когда поступает импульс синхронизации и триггер переключается в состояние $Q = 0$. Если $Q = 0$ и $\bar{Q} = 1$, то выход G_3 становится равным нулю при подаче импульса синхронизации и триггер устанавливается, т. е. переключается, в состояние $Q = 1$ и $\bar{Q} = 0$. В любом случае состояние триггера изменяется.

Однако ввиду присутствия обратной связи сигнал синхронизации, который остается равным единице (в то время как $JK = 1$) после того, как выход триггера изменил состояние однажды, приведет к повторному изменению состояния триггера. Рассмотрим это с помощью временной диаграммы (рисунок 1.21).

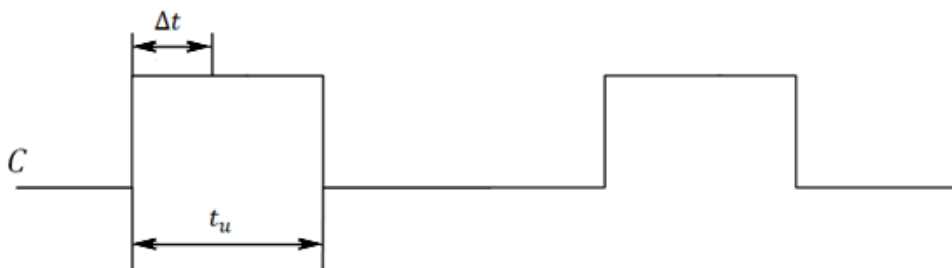


Рисунок 1.21 – Временная диаграмма синхроимпульсов

Когда $J = K = 1$ и $Q = 0$ и импульс синхронизации поступает на вход, после временного интервала Δt , равного времени задержки распространения сигнала через два элемента И – НЕ (G_3 и G_1), выход триггера изменится на $Q = 1$. Теперь при $J = K = 1$ и $Q = 1$ после другого временного интервала Δt выход триггера изменится на $Q = 0$. Следовательно, в течение длительности импульса синхронизации состояние триггера находится в автоколебательном режиме, а после окончания импульса синхронизации состояние триггера будет неизвестно. Такая ситуация называется круговым состязанием.

Этой ситуации можно было бы избежать, если бы $t_u < \Delta t < T$. Однако выполнить это невозможно в силу очень малого времени задержки распространения сигнала в ИС.

Практическим методом преодоления этой ситуации является использование M - S -конфигурации триггера (*Master-Slave*).

1.7 JK -триггер M - S -конфигурации с инвертором

JK -триггер M - S -конфигурации (рисунок 1.22) состоит из двух RS -триггеров с обратной связью с выхода второго на выход первого.

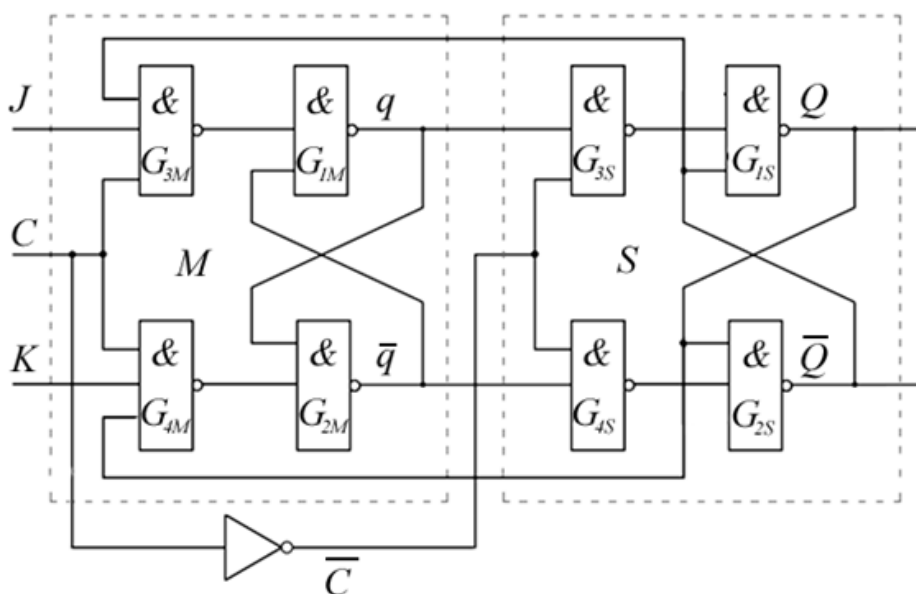


Рисунок 1.22 – JK -триггер M - S -конфигурации с инвертором

Когда сигнал синхронизации $C=1$, первый триггер имеет разрешение и выходы q и \bar{q} и зависит от состояния J и K и входов обратной связи. В то же время на второй RS -триггер подается запрет, поскольку $\bar{C}=0$. Когда сигнал синхронизации изменяется на низкий уровень $C=0$ ($\bar{C}=1$), на первый RS -триггер подается запрет, и он сохраняет свое состояние, а на второй триггер подается разрешение, поскольку теперь его синхровход $\bar{C}=1$. Поэтому выходы Q и \bar{Q} повторяют состояния q и \bar{q} соответственно. И поскольку второй триггер всегда повторяет состояние первого, то в англоязычной литературе первый называется *Master (M)*, а второй *Slave (S)*.

В этой схеме выходы и входы G_{3M} и G_{4M} не меняются в течение импульса синхронизации, поэтому условия для состязаний отсутствуют. Состояние M - S -триггера изменяется при отрицательном переходе импульса синхронизации.

Имеется другая версия JK -триггера в конфигурации M - S .

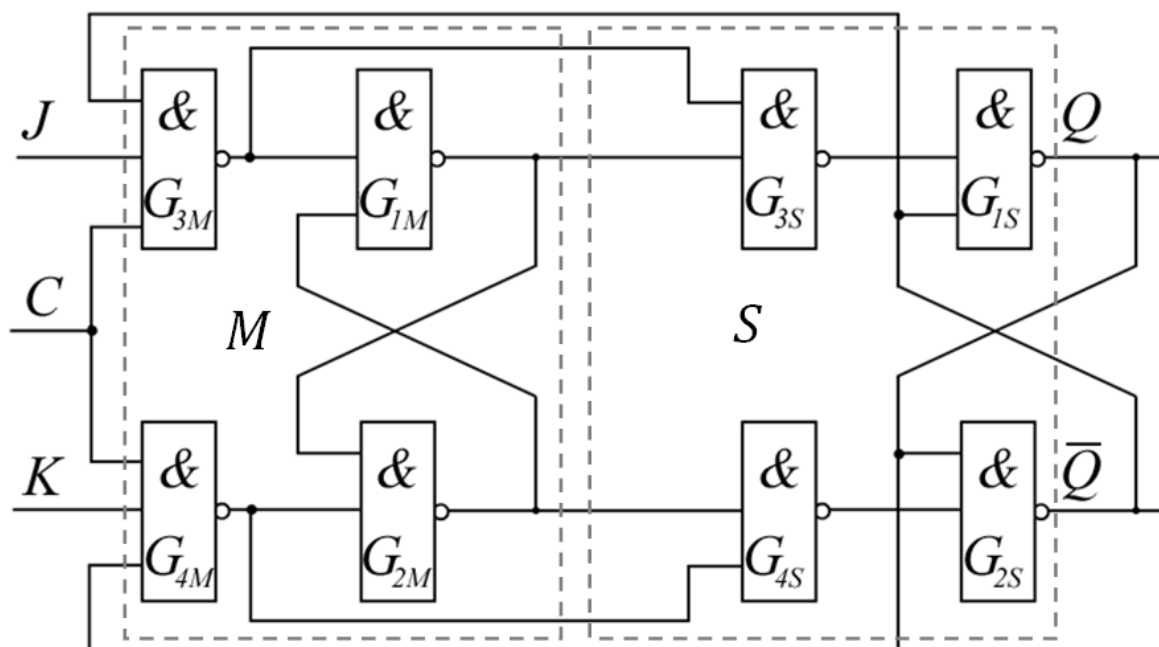


Рисунок 1.23 – JK -триггер M - S -конфигурации с запрещающими связями

Такой триггер изменяет свое состояние при $J=K=1$ только при изменении синхросигнала с состояния логической единицы на состояние логического нуля. Последовательность изменения состояния нуля и единицы на выходе в два раза меньше, чем синхросигнала.

1.8 T-триггер

T-триггер – это одноходовый вариант JK-триггера (рисунок 1.24).

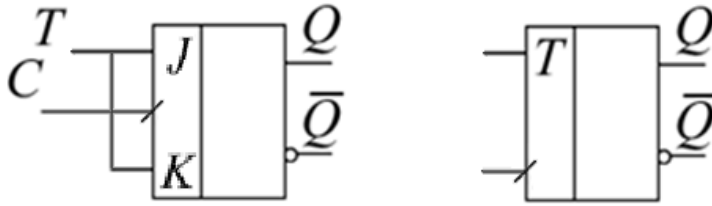


Рисунок 1.24 – T-триггер

T-триггер получается из JK-триггера (см. рисунок 1.24), если входы J и K объединить. Название триггера T происходит от слова «toggle» – «изменение состояния». Он меняется независимо от состояния, в котором находится, когда подается импульс синхронизации при логической единице на входе T и не меняет состояния, если на входе T логический нуль.

1.9 Преобразование триггера одного типа в триггер другого типа

При синтезе последовательных схем мы обычно рассматриваем ситуации, в которых исходное состояние, в котором находится схема в настоящий (данный) момент времени, и состояние схемы в следующий момент времени (состояние перехода) определены и мы должны определить входные условия, при которых желаемый переход осуществляется. Под состоянием схемы в данный момент времени и состоянием схемы в следующий момент времени мы понимаем состояние схемы до и после импульса синхронизации соответственно. Например, выход RS -триггера до прихода синхроимпульса $Q_n = 0$. Необходимо, чтобы выход не изменил состояние, когда синхроимпульс поступит на схему.

Из характеристической таблицы RS -триггера получаем следующие условия:

- 1) $S_n = R_n = 0$,
- 2) $S_n = 0, R_n = 1$.

Из этого мы принимаем, что S_n должен быть равен нулю, а R_n может быть нулем или единицей (неопределенное состояние). Подобным образом могут быть найдены входные условия для всех возможных переходов. Эти условия в табличном виде представляют собой таблицу переходов триггера, которая очень полезна при синтезе последовательных схем.

Рассмотрим таблицу переходов для триггеров RS , JK , T , D (таблица 1.8).

Таблица 1.8 – Совмещенная таблица переходов

Исходное состояние	Состояние перехода	<i>RS</i> -триггер		<i>JK</i> -триггер		<i>T</i> -триггер	<i>D</i> -триггер
		S_n	R_n	J_n	K_n	T_n	D_n
Q_n	Q_{n+1}						
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	1	0
1	1	x	0	x	0	0	1

Мы уже рассматривали преобразование *RS*-триггера в *JK*- и *D*-триггеры и *JK*- в *T*-триггер. Сейчас рассмотрим формальный подход (метод преобразования одного триггера в другой).

Для этого используем общую модель преобразования одного триггера в другой (рисунок 1.25).

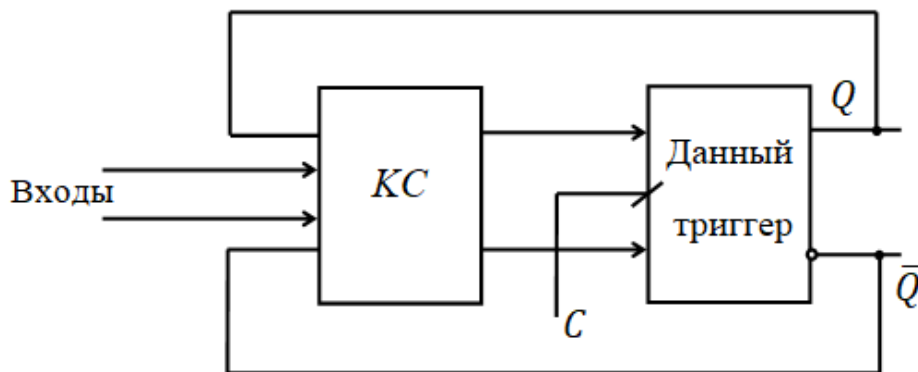


Рисунок 1.25 – Общая модель преобразования триггеров

Рассмотрим примеры преобразования триггеров.

Пример 1. Преобразовать *JK*-триггер в *D*-триггер.

Решение. В данном случае модель преобразования показана на рисунке 1.26.

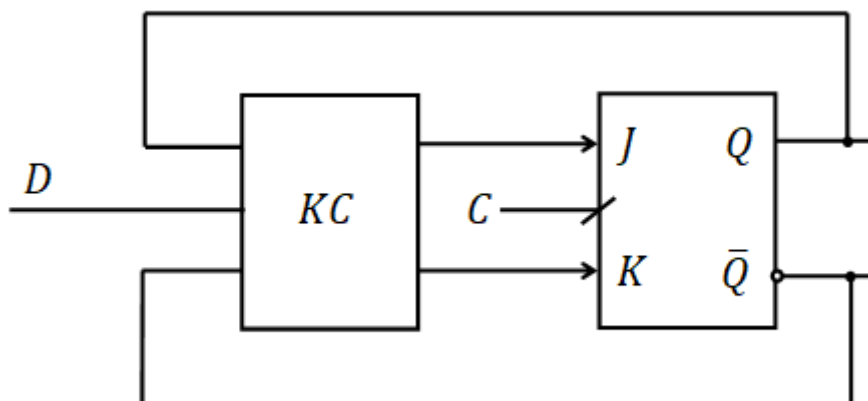


Рисунок 1.26 – Модель преобразования *JK*-триггера в *D*-триггер

Используя таблицы переходов JK -триггера и D -триггера, определяем функции J и K (рисунок 1.27).



Рисунок 1.27 – Карты Карно для определения J_n и K_n

Результат преобразования показан на рисунке 1.28.

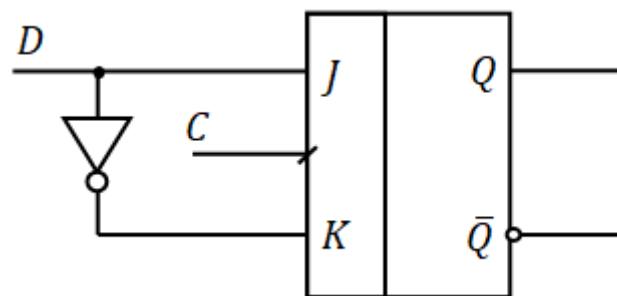


Рисунок 1.28 – D -триггер как результат преобразования JK -триггера

Пример 2. Преобразовать D -триггер в JK -триггер.

Решение. В данном случае модель преобразования показана на рисунке 1.29.

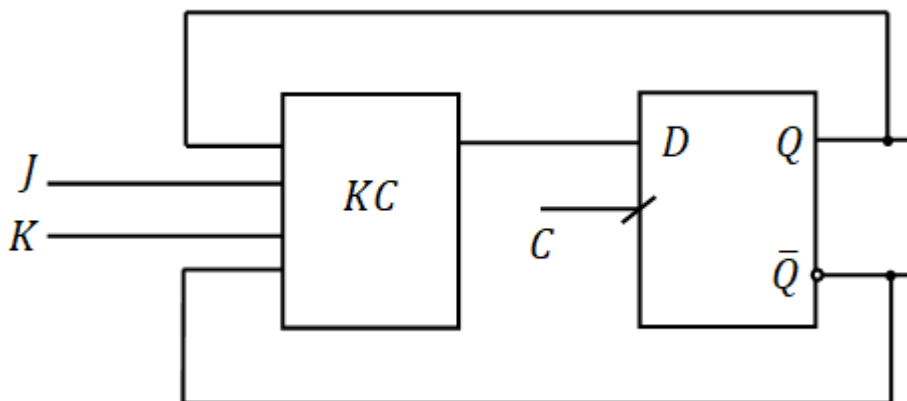


Рисунок 1.29 – Модель преобразования D -триггера в JK -триггер

Опять воспользуемся таблицами переходов для JK -триггера и D -триггера для получения карты Карно для функции D (рисунок 1.30).

		$J_n K_n$			
		00	01	11	10
Q_n	0	0	0	1	1
	1	1	0	0	1

$$D_n = \bar{Q}_n \cdot J_n + Q_n \cdot \bar{K}_n = \overline{\bar{Q}_n \cdot J_n \cdot \bar{Q}_n \cdot \bar{K}_n}$$

Рисунок 1.30 – Карта Карно для минимизации D_n

Результат преобразования показан на рисунке 1.31.

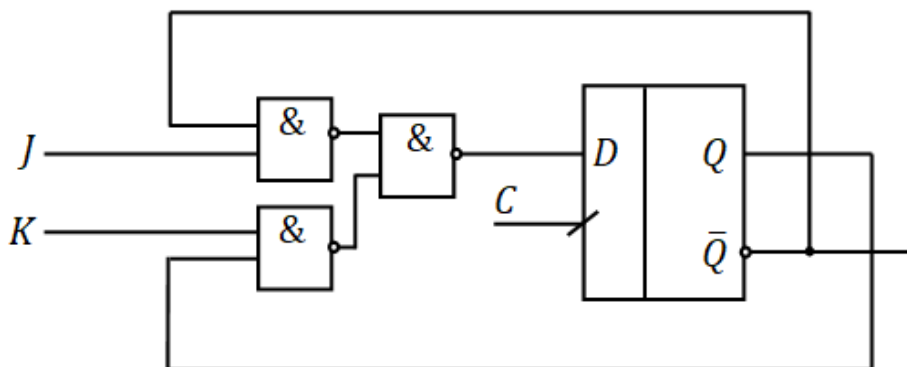


Рисунок 1.31 – JK -триггер на базе D -триггера

Пример 3. Преобразовать D -триггер в T -триггер.

Решение. Для данного случая модель преобразования показана на рисунке 1.32.

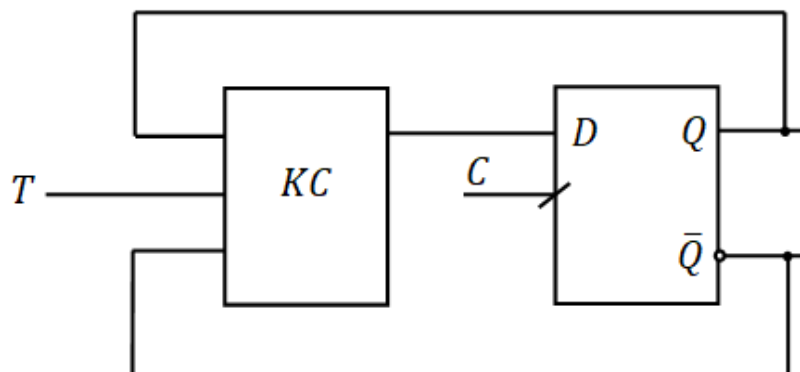


Рисунок 1.32 – Модель для преобразования D -триггера в T -триггер

Воспользуемся таблицами переходов D -триггера и T -триггера для построения карты Карно минимизации функции D (рисунок 1.33).

		T	
		0	1
Q_n	0	0	1
	1	1	0

$$D_n = Q_n \oplus T_n = \bar{Q}_n \cdot T_n + Q_n \cdot \bar{T}_n$$

Рисунок 1.33 – Карта Карно для определения D_n

Результат преобразования показан на рисунке 1.34.

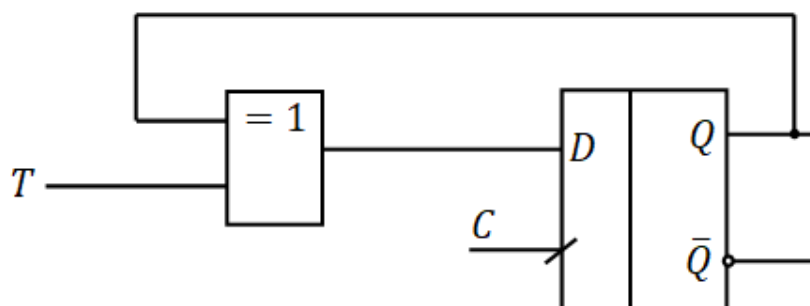


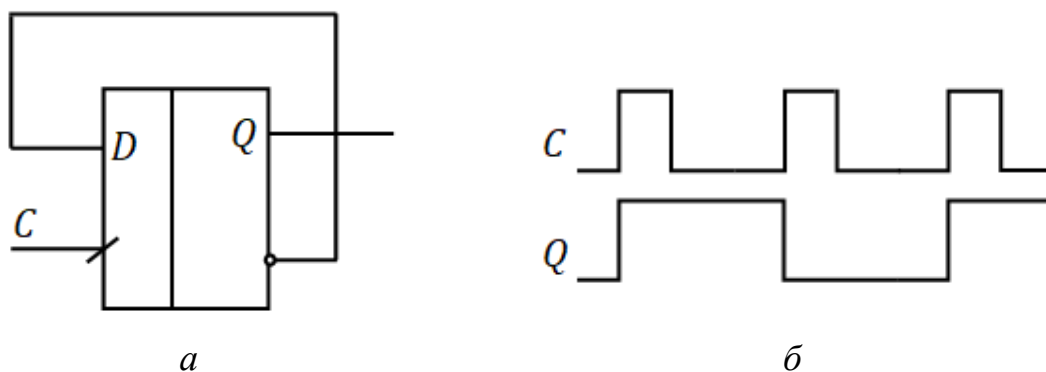
Рисунок 1.34 – Результат преобразования D -триггера в T -триггер

Если вход $T=0$, то триггер не меняет свое состояние. А если вход $T=1$, то триггер будет менять свое состояние всякий раз, когда поступает импульс синхронизации.

Если переменная T в уравнении $D_n = Q_n \oplus T_n$ равна единице, то уравнение может быть преобразовано следующим образом:

$$D_n = Q_n \oplus T_n = Q_n \oplus 1 = \bar{Q}_n.$$

Это уравнение показывает, что схему делителя на два можно построить соединением \bar{Q} с D входом (рисунок 1.35).



a – схема; b – временные диаграммы
Рисунок 1.35 – Делитель на два на базе D -триггера

1.10 Триггеры с динамическим управлением

В триггерах с динамическим управлением выходное состояние изменяется в момент перехода синхросигнала с нулевого уровня на единичный ($0 \rightarrow 1$) либо наоборот ($1 \rightarrow 0$) при достижении какого-то порогового уровня. При этом информационные входы триггера становятся нечувствительными к изменениям информационных сигналов до тех пор, пока синхросигнал не вернется в исходное нулевое состояние, и другой синхроимпульс не поступит на синхровход.

Если триггер переключается положительным перепадом синхросигнала, то вход C называется прямым динамическим, а если отрицательным перепадом – то инверсным динамическим.

Рассмотрим D -триггер (F - F) с динамическим управлением (рисунок 1.36).

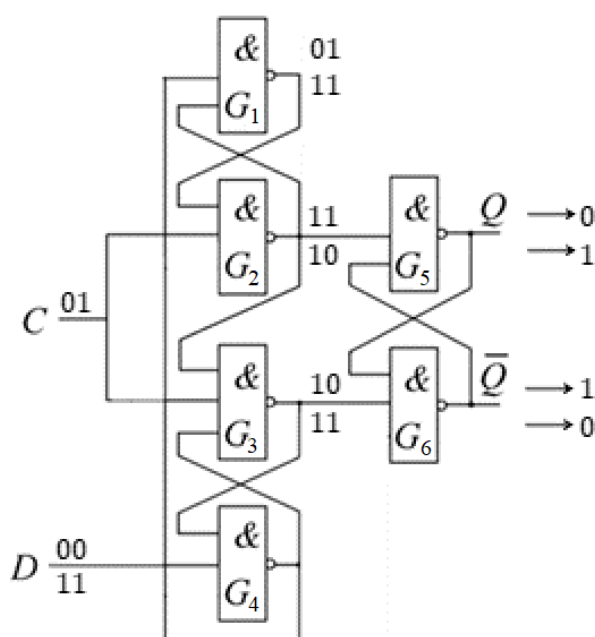


Рисунок 1.36 – Логическая схема D -триггера с динамическим управлением

Логические элементы G_1 и G_2 образуют один базовый \overline{RS} -триггер, элементы G_3 и G_4 образуют второй. Третий базовый \overline{RS} -триггер, образованный логическими элементами G_5 и G_6 , является выходом всей схемы.

Выходы \overline{S} и \overline{R} третьего триггера должны оставаться в состоянии логической единицы для того, чтобы выходы оставались в устойчивом состоянии.

Когда $\overline{S} = 0$ и $\overline{R} = 1$, выход устанавливается в $Q = 1$. Когда $\overline{S} = 1$ и $\overline{R} = 0$, выход сбрасывается в $Q = 0$. Состояния входов \overline{S} и \overline{R} определяются

состояниями выходов других базовых \overline{RS} -триггеров, которые зависят от внешних входов D и C (синхросигнала).

Теперь рассмотрим работу триггера. Состояния выходов первых двух триггеров (выходы элементов G_2 и G_3) поступают на входы элементов G_5 и G_6 и определяют состояние выхода триггера. Когда синхровход $C = 0$, вход D может быть равен нулю или единице, однако в любом случае выходы элементов G_2 и G_3 равны единице, т. е. $\overline{R} = \overline{S} = 1$, а это условие установочного состояния выхода триггера.

Когда $D = 0$, выход элемента G_4 равен единице, и это приводит к тому, что выход элемента G_1 устанавливается в нуль. Когда вход $D = 1$, выход элемента G_4 равен нулю, что приводит к тому, что на выходе G_1 устанавливается логическая единица. Это два возможных состояния, когда на синхровходе C логический нуль, исключая любые изменения на выходе триггера, независимо от значения сигнала на входе D .

Теперь рассмотрим изменения состояний схемы, когда импульс синхронизации поступает на вход. Определенное время, называемое временем установки, в течение которого на входе D должен сохраниться постоянный уровень до поступления синхронизации. Это время установки равняется времени распространения (задержки) через элементы G_4 и G_1 , поскольку изменения на входе D приводят к изменениям на выходах элементов. При этом, что уровень сигнала на входе D не изменяется в течение этого времени установки и что сигнал на синхровходе становится равным логической единице.

Если вход $D = 0$, когда синхровход становится равным логической единице, то \overline{S} остается равным единице, а \overline{R} изменяется на нуль. Это приводит к установлению на выходе \overline{Q} логической единицы и на Q – логического нуля. И даже пока $C = 1$, если произойдет изменение состояния на входе D , выход элемента G_4 будет оставаться в состоянии логической единицы (даже если на D установится логическая единица), поскольку один из входов, соединенный с \overline{R} , остается в состоянии нуля. И лишь когда синхровход C возвращается в состояние нуля, возможно изменение выхода G_4 ; однако в этом случае оба входа \overline{S} и \overline{R} становятся равными единице, запрещая любые изменения выхода триггера. Тем не менее имеется определенное время, называемое временем удержания, когда вход D должен оставаться без изменения после поступления положительного перепада синхроимпульса. Это время удержания равно времени задержки распространения элемента G_3 , чтобы \overline{R} установился в нуль и заблокировал G_4 для удержания его выхода в состоянии логической единицы, независимо от уровня на входе D .

Если вход $D = 1$, когда на вход C поступает положительный перепад, тогда \bar{S} изменяется на нуль, а \bar{R} остается равным единице, что приводит к установлению $Q = 1$. Изменения на входе D в течение времени, когда $C = 1$, не изменяют состояние входов \bar{S} и \bar{R} , поскольку на выходе G_1 сохраняется единица, потому что $\bar{S} = 0$.

Когда синхровход установится в нуль, \bar{S} и \bar{R} установятся в единицу, препятствуя любым изменениям выхода триггера.

Итак, когда на синхровходе положительный перепад, значение входа D передается на выход Q . Изменения на D -входе после того, как значения уровня на входе C установилось в единицу, не изменяет Q . Более того, отрицательный перепад синхроимпульса не изменяет состояние выхода триггера и также выход не изменяется, когда $C = 0$. В триггере с динамическим синхровходом любые проблемы, связанные с обратной связью, устраняются, как и в триггере *MS*-типа. Время установления и время удержания надо принимать во внимание при использовании данного типа триггера.

Когда используются различные типы триггеров в той же последовательной схеме, необходимо организовывать переключение триггера в одно и то же время.

2 ТИПОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ СХЕМЫ

Триггеры являются базовыми элементами последовательностных устройств. Произвольные последовательностные схемы могут быть построены с использованием триггеров и логических комбинационных схем. Типовыми и наиболее часто используемыми последовательностными устройствами являются регистры и счетчики.

Далее рассмотрим построение регистров и счетчиков и некоторые их применения.

2.1 Регистры и их применение

Триггер может хранить (запоминать) один бит цифровой информации (единицу или нуль). Его также можно назвать одноразрядным регистром. Группа триггеров для хранения двоичной информации (один триггер на каждый бит информации) называется регистром. Регистры находят применение в различных цифровых системах, включая микропроцессоры.

Данные (или двоичная информация) могут вводиться в регистр в последовательной форме (бит за битом) или в параллельной форме (все биты одновременно) и выводиться из регистра в последовательной или параллельной форме. Данные в последовательной форме также называются временным кодом, а в параллельной форме – пространственным кодом.

Регистры классифицируются в зависимости от того, в какой форме данные вводятся в регистр и в какой форме данные выводятся.

Существует четыре возможности:

- последовательный ввод и последовательный вывод;
- последовательный ввод и параллельный вывод;
- параллельный ввод и последовательный вывод;
- параллельный ввод и параллельный вывод.

Регистры могут быть построены с использованием *RS*-, *JK*-, *D*-триггеров. Они широко представлены в качестве интегральных схем (ИС) средней степени интеграции.

Регистры, в которых ввод или вывод данных производится последовательно, называются сдвигающими, поскольку биты информации сдвигаются в триггеры регистра при подаче синхроимпульсов вправо или влево.

В некоторых регистрах информация может сдвигаться вправо или влево в зависимости от управляющего сигнала. Такие регистры называются реверсивными.

Если регистр может работать во всех четырех режимах и также является реверсивным, то такой регистр называется универсальным.

Построение регистра и его работу рассмотрим на примере регистра ИС 7496 (*TTL*). 7496 – это ИС, которая содержит пятиразрядный сдвигающий регистр, построенный на *JK*-триггерах. Этот регистр может работать во всех четырех режимах. Рассмотрим рисунок 2.1.

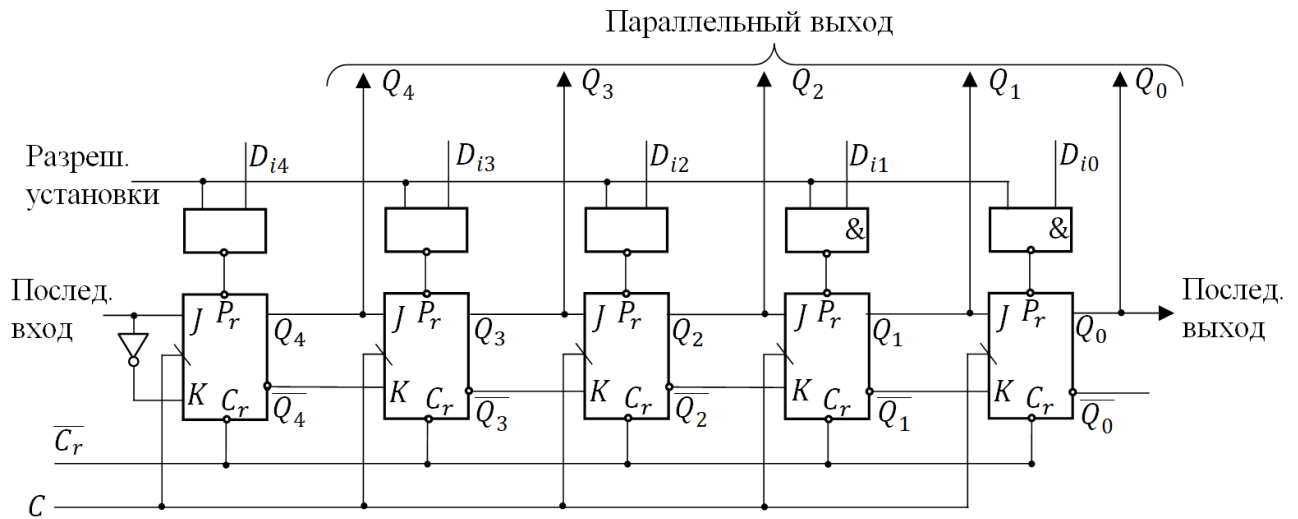


Рисунок 2.1 – Пример построения регистра (ИС 7496)

Рассмотрим работу регистра, если на последовательный вход поступают данные 10110. Для любой другой пятибитной информации работа аналогична.

Последовательный ввод информации. После обнуления регистра данные 10110 в последовательной форме подаются на последовательный вход. Это отражено на рисунке 2.2.

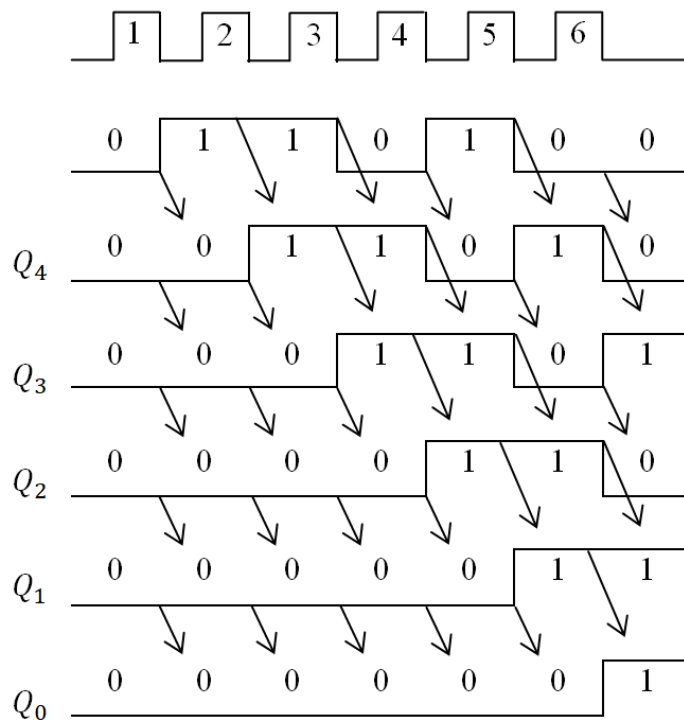


Рисунок 2.2 – Временные диаграммы работы регистра (ИС 7496)

После пяти импульсов синхронизации на выходах триггеров данные 10110 доступны в параллельной форме, т. е. это одно из возможных применений регистра: преобразование информации из последовательной формы в параллельную.

Процесс ввода данных также называется записью этих данных в регистр.

Данные, которые хранятся в регистре, могут выводиться из регистра (читаться) двумя путями: в последовательном виде и в параллельном. Данные в последовательном виде будут на выходе Q_0 при подаче синхроимпульсов. Число синхроимпульсов, требуемое для вывода данных, равно числу бит данных.

В параллельном виде данные доступны на выходах $Q_4 - Q_0$, синхроимпульсы не требуются, чтобы прочитать эту информацию.

В случае последовательного вывода информации после пяти синхроимпульсов регистр обнуляется. Это означает, что после извлечения информации в последовательном виде регистр пустой. В случае параллельного вывода содержание регистра может быть прочитано любое число раз и данные будут сохраняться до тех пор, пока новые данные не будут записаны в регистр.

Параллельный ввод. Данные могут быть записаны в регистр в параллельной форме с использованием входа «Разрешение установки». В этом случае, после обнуления регистра данные подаются на параллельные входы $D_{i4} - D_{i0}$ и при подаче высокого уровня или логической единицы на вход «Разрешение установки» эти данные записываются в регистр. Эта операция называется асинхронной загрузкой. Теперь хранимая информация может быть считана из регистра в последовательном или в параллельном виде.

Кольцевой счетчик. Если последовательный выход регистра Q_0 соединить с последовательным входом, то единичный бит, записанный в один из триггеров, будет циркулировать по регистру при подаче синхроимпульсов. Такая схема называется кольцевым счетчиком. Его временные диаграммы изображены на рисунке 2.3.

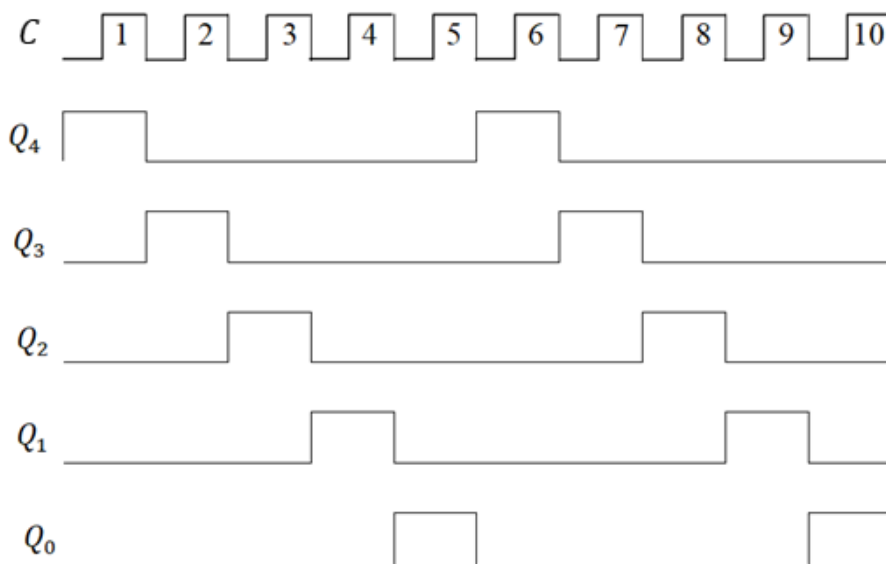


Рисунок 2.3 – Временные диаграммы кольцевого счетчика

На выходах триггеров генерируются неперекрывающиеся последовательности импульсов (рисунок 2.3), которые могут быть полезны для различных приложений (например, шаговый двигатель).

Схема может быть также использована для подсчета импульсов. Число сосчитанных импульсов определяется единичным уровнем на выходе соответствующего триггера. Эта схема может рассматриваться как делитель на $N(N:1)$.

Счетчик Джонсона. Если выход \bar{Q}_0 соединить с последовательным входом, то такая схема называется счетчиком Джонсона (*Johnson*) или *Moebius*-счетчиком. Если в такой схеме после обнуления регистра подать импульсы синхронизации (тактовые), на выходах триггеров будут генерироваться прямоугольные волны (рисунок 2.4).

Счетчик Джонсона – это делитель на $2N$ или модель счета такого счетчика по $\text{mod} = 2N$.

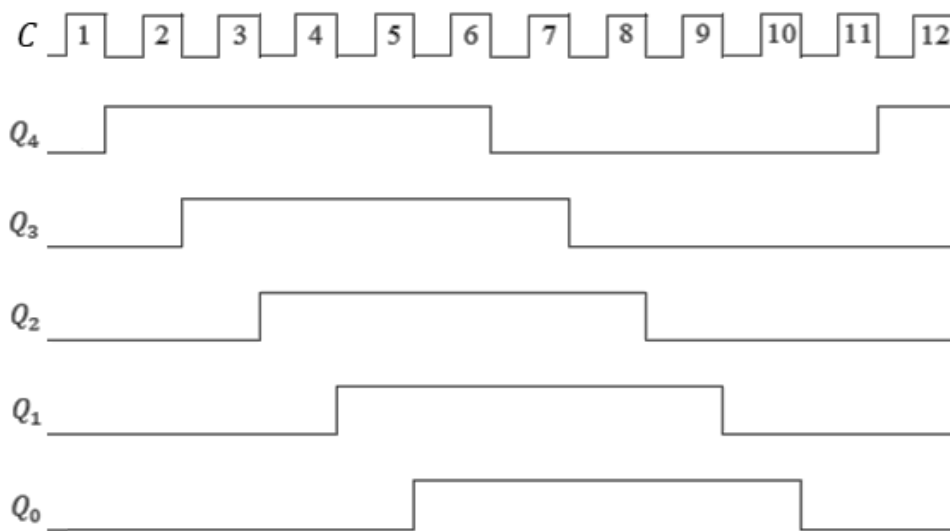


Рисунок 2.4 – Временные диаграммы счетчика Джонсона

Генераторы последовательностей. Схема, которая генерирует заданную последовательность бит синхронно с импульсами синхронизации, называется генератором последовательностей.

Такие генераторы используются:

- как счетчики;
- псевдослучайные генераторы;
- генераторы заданной последовательности и заданного периода;
- генераторы кодов.

Блок-схема генератора последовательности показана на рисунке 2.5.

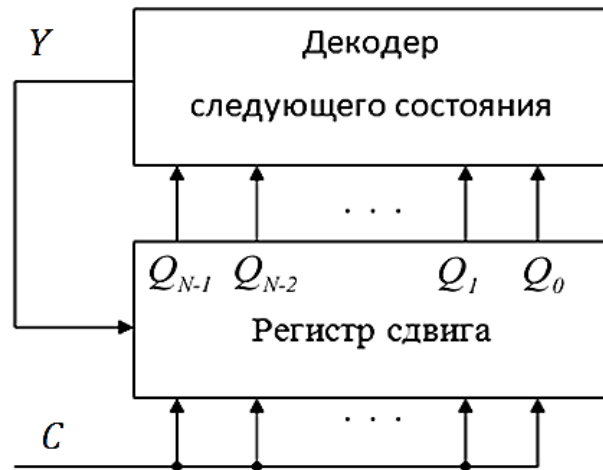


Рисунок 2.5 – Общая блок-схема генератора последовательности

Выход декодера следующего состояния Y – это функция от $Q_{n-1}Q_{n-2}\dots Q_1Q_0$. Эта схема подобна кольцевому счетчику ($Y=Q_0$) или счетчику Джонсон ($Y=\overline{Q_0}$), которые являются частными случаями генераторов последовательностей.

Синтез генераторов последовательностей рассмотрим на примерах.

Пример 1. Синтезировать генератор следующей последовательности:

...1101011... .

Решение. При синтезе генератора заданной последовательности необходимо определить число разрядов регистра сдвига и комбинационную схему декодера следующего состояния.

Возможное минимальное число триггеров N в режиме для генерирования последовательности длиной S бит определяется исходя из того, что

$$S \leq 2^N - 1 \quad \text{или} \quad N \geq \log_2(S+1). \quad (2.1)$$

В данном примере $S = 7$, поэтому минимально возможное значение $N = \log_2(7+1) = 3$. Однако это не значит, что это число триггеров является достаточным. Если данная последовательность ведет к семи различным состояниям регистра, то тогда трех триггеров будет достаточно, в противном случае число триггеров придется увеличить. Запишем состояние регистра в виде таблицы 2.1.

Таблица 2.1 – Таблица состояний регистра, $N = 3$

Число синхро- импульсов	Выходы триггеров		
	Q_2	Q_1	Q_0
1	1	1	1
2	1	1	1
3	0	1	1
4	1	0	1
5	0	1	0
6	1	0	1
7	1	1	0

Допускаем, что данная последовательность генерируется на выходе Q_2 , в таком случае на выходах Q_1 и Q_0 будет та же последовательность, только задержанная на один и два такта соответственно. Из этой таблицы видно, что не все состояния регистра являются разными, что означает, что $N = 3$ не является достаточным. Поэтому примем число $N = 4$ и нарисуем таблицу 2.2.

Таблица 2.2 – Таблица состояний регистра, $N = 4$

Число синхро- импульсов	Выходы триггеров				Y
	Q_3	Q_2	Q_1	Q_0	
1	1	1	1	0	1
2	1	1	1	1	0
3	0	1	1	1	1
4	1	0	1	1	0
5	0	1	0	1	1
6	1	0	1	0	1
7	1	1	0	1	1
8(1)	1	1	1	0	1

Поскольку в данном случае все состояния регистра сдвига являются отличными (разными), то $N = 4$ будет достаточным. Добавляем в эту таблицу колонку, в которую запишем требуемую последовательность на входе регистра при подаче тактовых импульсов.

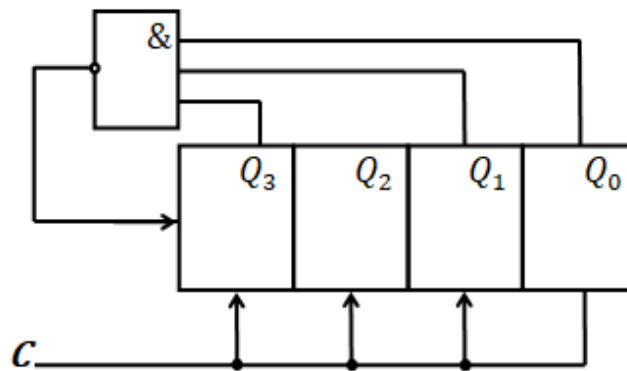
Схему декодера получим, упрощая функцию $Y(Q_3, Q_2, Q_1, Q_0)$ с помощью карты Карно (рисунок 2.6).

	Q_1Q_0			
Q_3Q_2	00	01	11	10
00	x	x	x	x
01	x	1	1	x
11	x	1	0	1
10	x	x	0	1

$$Y = \bar{Q}_3 + \bar{Q}_1 + \bar{Q}_0 = \overline{Q_3 \cdot Q_1 \cdot Q_0}$$

Рисунок 2.6 – Минимизация декодера

После минимизации упрощенная схема генератора заданной последовательности будет выглядеть, как показано на рисунке 2.7.



a

1	↘	1	↘	1	↘	1	↘	0
0	↘	1	↘	1	↘	1	↘	1
1		0		1		1		1
0		1		0		1		1
1		0		1		0		1
1		1		0		1		0
1		1		1		0		1

б

a – упрощенная схема генератора последовательности ...1101011...;
б – таблица состояний

Рисунок 2.7 – Синтез генератора последовательности

Пример 2. Синтезировать генератор последовательности ...1101001...

Решение. Определяем минимально возможное число триггеров регистра сдвига, достаточное для построения генератора заданной последовательности:

$$N = \log_2(S + 1) = \log_2(7 + 1) = 3.$$

Составляем таблицу состояний регистра и определяем, что число триггеров регистра сдвига $N = 3$ достаточно для реализации генератора заданной последовательности, поскольку все состояния регистра сдвига различны (таблица 2.3).

Таблица 2.3 – Таблица состояний регистра

Импульсы синхронизации	Выходы триггеров			Y
	Q_2	Q_1	Q_0	
1	1	1	1	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	0
5	0	1	0	1
6	1	0	1	1
7	1	1	0	1

Добавляем в таблице 2.3 столбец со значениями выходов декодера и минимизируем комбинационную схему декодера следующего состояния с помощью метода карт Карно (рисунок 2.8). Строим генератор заданной последовательности (рисунок 2.9).

		Q_1Q_0			
		00	01	11	10
Q_2	0	x	1	0	1
	1	0	1	0	1

$$Y = Q_1 \oplus Q_0$$

Рисунок 2.8 – Минимизация схемы декодера

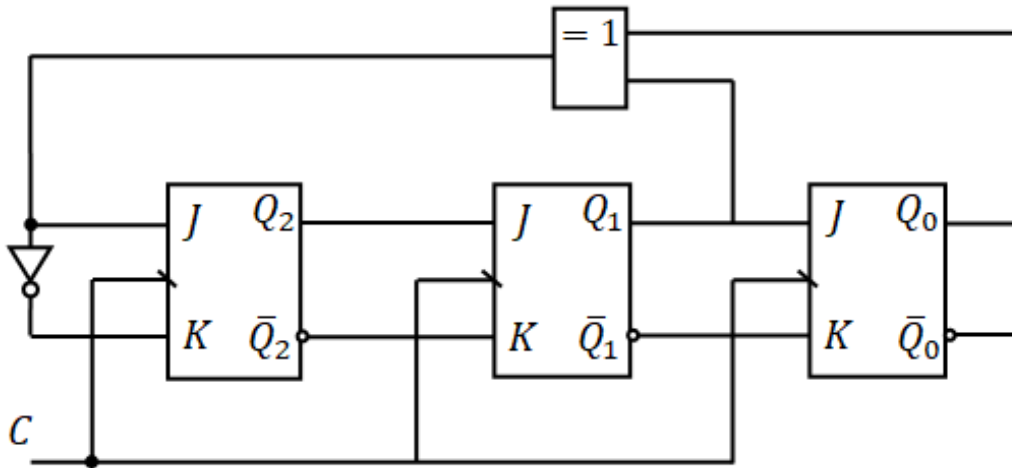


Рисунок 2.9 – Схема генератора последовательности ...1101001...

Если декодер следующего состояния представляет собой схему ИСКЛЮЧАЮЩЕЕ – ИЛИ, то такой генератор называется линейным, и если $S = 2^N - 1$, то это генератор последовательности максимальной длины.

2.2 Цифровые счетчики

Цифровой счетчик – это группа триггеров, соединенных так, чтобы считать число импульсов, поданных на вход, и фиксировать число подсчитанных импульсов в том или ином коде.

Основными характеристиками счетчика являются коэффициент (модуль) счета и быстродействие.

Коэффициент, или модуль счета, характеризует число устойчивых состояний счетчика.

Быстродействие счетчика зависит от используемой элементной базы и схемы построения.

Счетчики классифицируются по ряду признаков:

а) быстродействие и способ организации внутренних связей:

- 1) асинхронные;
- 2) синхронные;

б) направление счета:

- 1) суммирующие;
- 2) вычитающие;
- 3) реверсивные;

в) модуль счета:

- 1) двоичные;
- 2) двоично-десятичные;
- 3) другой модуль счета.

Классификационные признаки независимы и могут встречаться в разных сочетаниях.

В наиболее общем случае рассматриваются асинхронные и синхронные счетчики. В асинхронных счетчиках триггеры перебрасываются последовательно, а в синхронных счетчиках – одновременно. Основным достоинством асинхронных счетчиков является их схемная простота, а недостатком – низкое быстродействие. Основным достоинством синхронных счетчиков является их более высокое быстродействие, а недостатком – более сложная схемная реализация.

2.2.1 Асинхронные счетчики

Чтобы изучить асинхронные счетчики, рассмотрим счетную последовательность (таблица 2.4). Число состояний в такой последовательности 8, что требует использования трех триггеров ($2^3 = 8$).

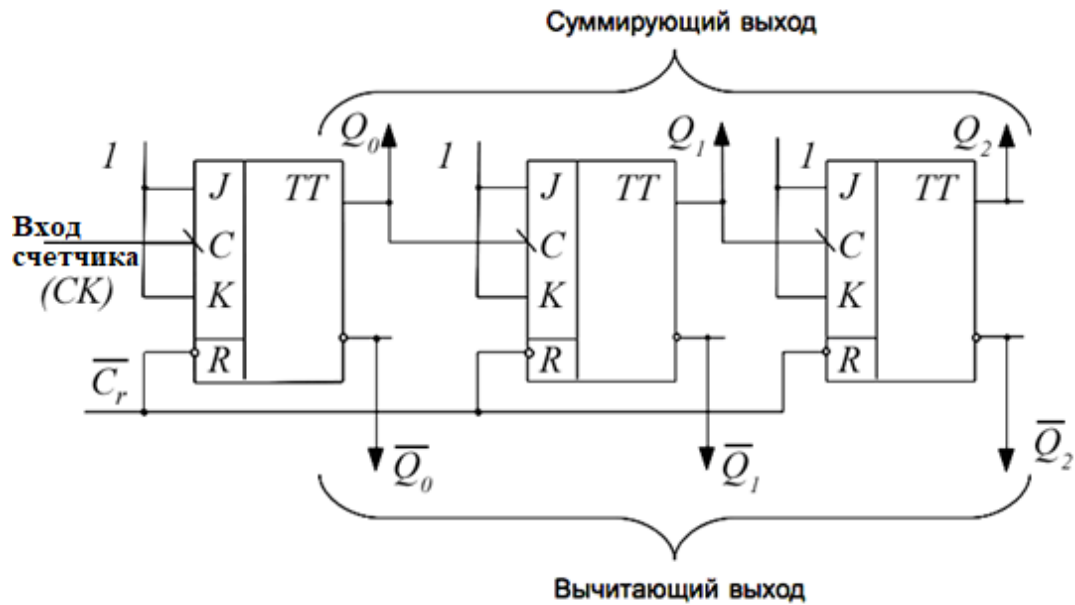
Таблица 2.4 – Счетная последовательность

Счет	Состояние счетчика		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8(0)	0	0	0

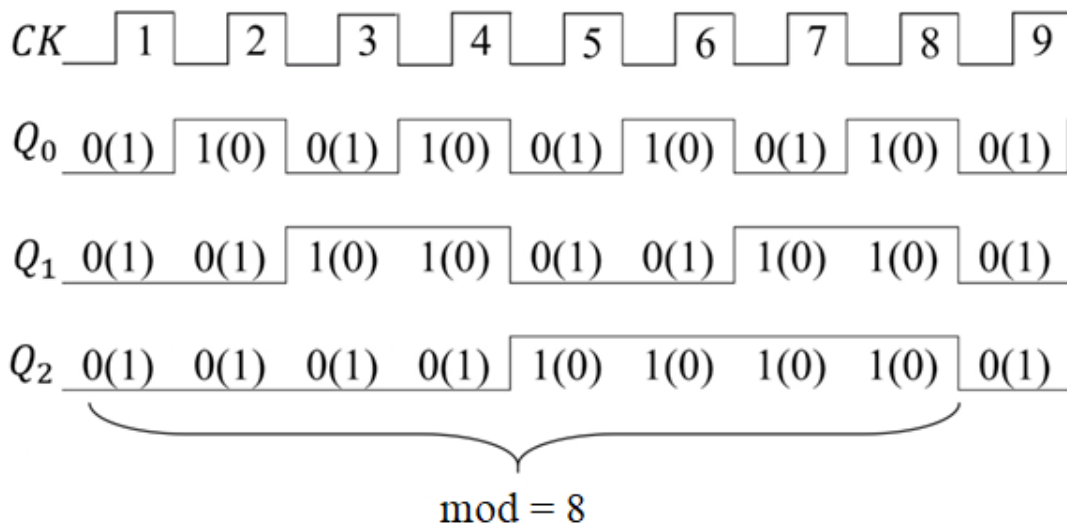
Выход Q_0 наименее значащего триггера (разряда) изменяется с каждым счетным импульсом. Это может быть реализовано путем использования триггера T -типа при $T = 1$. Состояние на выходе Q_1 меняется всякий раз, когда выход Q_0 меняется из единицы в нуль. Поэтому если выход Q_0 соединить с входом синхронизации следующего T -триггера с инверсным динамическим входом, то выход Q_1 будет менять состояние всякий раз, когда выход Q_0 осуществляет переход из единицы в нуль (отрицательный перепад синхроимпульса). Таким же образом переход из единицы в нуль с выхода Q_1 изменяет состояние Q_2 , что достигается путем соединения Q_1 с синхровходом следующего триггера. Аналогичным образом может быть построен асинхронный счетчик с большим числом разрядов или модулем счета.

На рисунке 2.10, а приведена схема асинхронного счетчика с использованием триггеров с инверсным динамическим входом ($TB6$). Входы J и K соединены вместе, образуя вход T -триггера, и на них подана логическая единица.

На рисунке 2.10, б приведены временные диаграммы, поясняющие работу счетчика. На прямых входах триггеров (Q_2, Q_1, Q_0) отражается состояние суммирующего счетчика, когда с каждым счетным импульсом состояние счетчика увеличивается. В то же самое время этот счетчик можно рассматривать как вычитающий, если информацию о состоянии счетчика снимать с инверсных выходов триггеров ($\bar{Q}_2, \bar{Q}_1, \bar{Q}_0$).



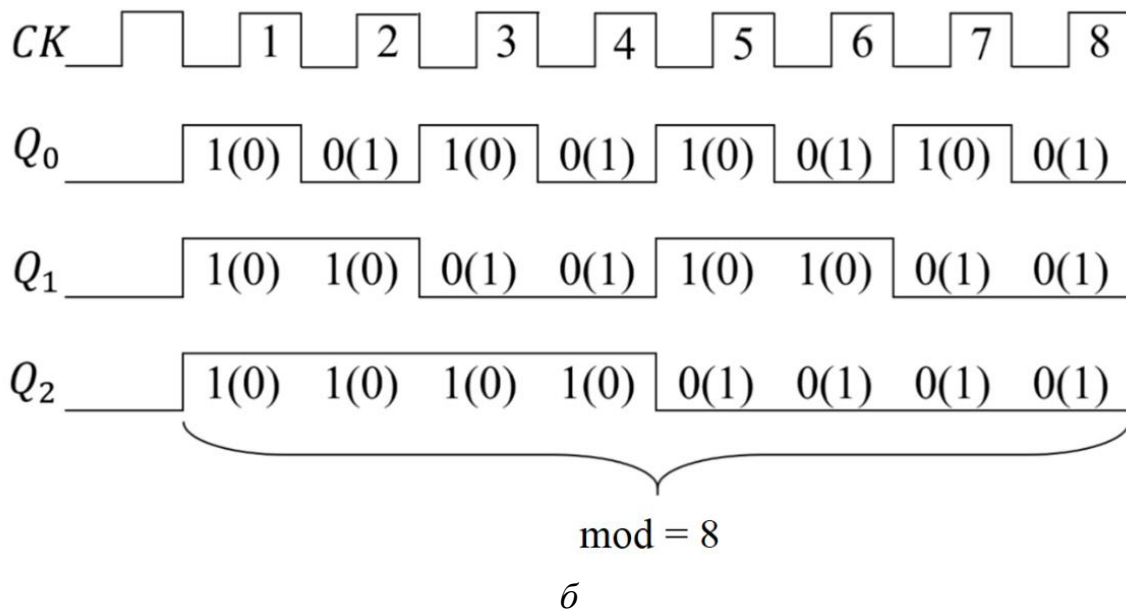
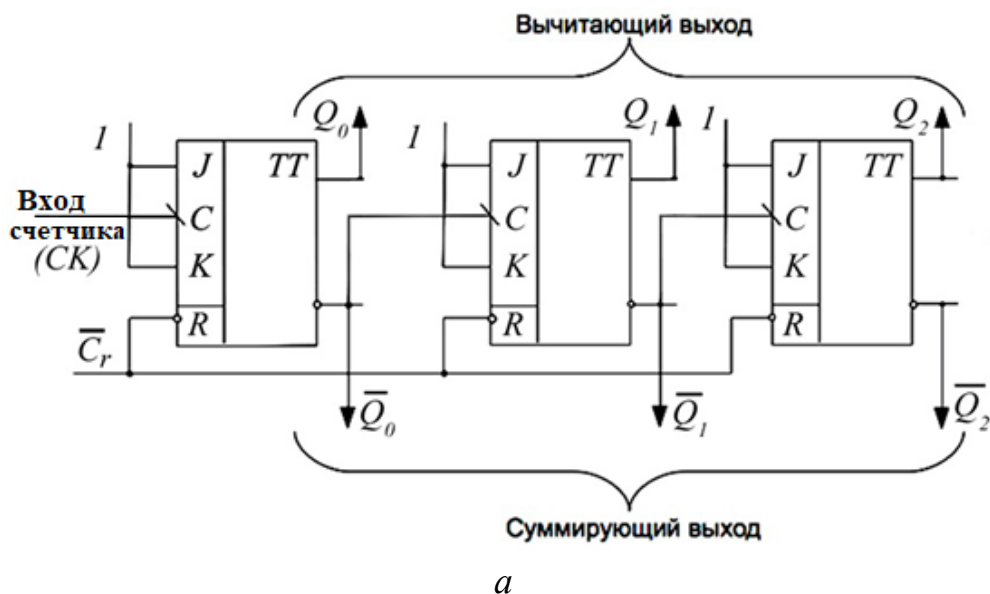
а



б

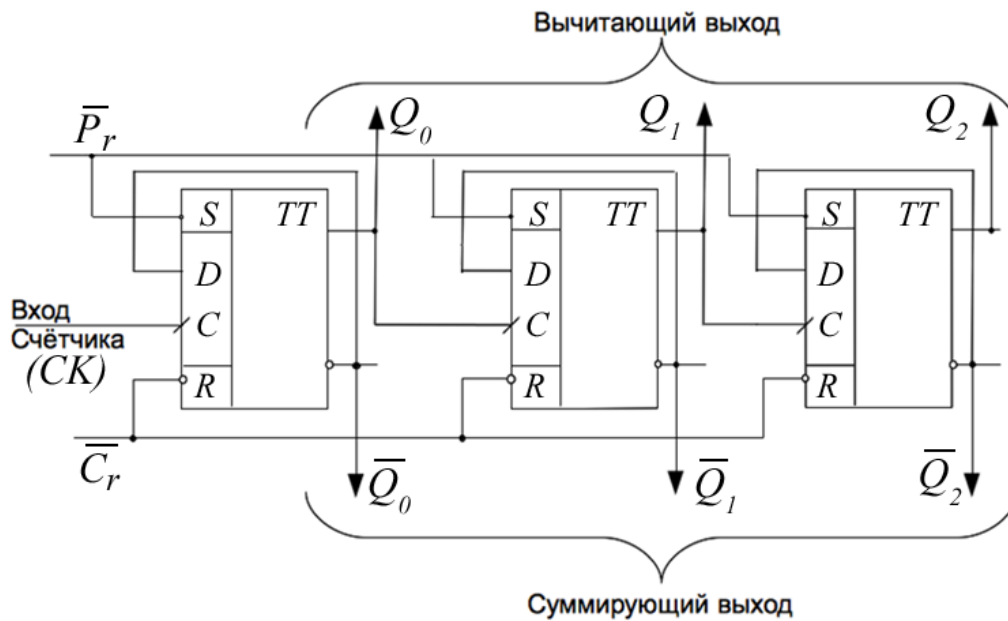
а – логическая схема счетчика; б – временные диаграммы
Рисунок 2.10 – Асинхронный суммирующий счетчик на триггерах с инверсным динамическим синхровходом

На рисунке 2.11, *a* приведен вариант схемы асинхронного счетчика на тех же триггерах с инверсным динамическим входом. На этот раз инверсные входы триггеров соединены с синхровходом последующих триггеров. В этой схеме суммирующий счетчик получается, если снимать информацию о состоянии счетчика с инверсных входов триггеров ($\bar{Q}_2, \bar{Q}_1, \bar{Q}_0$), и вычитающий счетчик, если снимать информацию с прямых выходов триггеров (Q_2, Q_1, Q_0).

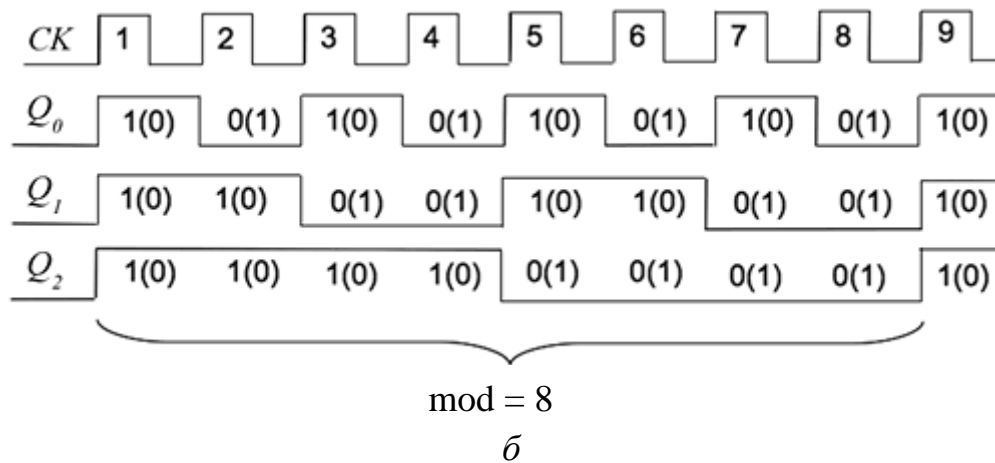


a – логическая схема счетчика; *б* – временные диаграммы
 Рисунок 2.11 – Асинхронный вычитающий счетчик на триггерах с инверсным динамическим синхровходом

Рассмотрим теперь, как строятся асинхронные счетчики на основе триггеров с прямым динамическим входом (например, ТМ2). На рисунке 2.12 приведены схемы и временные диаграммы, поясняющие работу счетчика. Если в счетчике на триггерах с прямым динамическим синхровходом прямые выходы триггеров соединены с синхровходами последующих, то суммирующий и вычитающий выходы счетчика поменяются местами.



a



b

a – логическая схема счетчика; *b* – временные диаграммы

Рисунок 2.12 – Асинхронный счетчик на триггерах с прямым динамическим синхровходом

Асинхронные счетчики, построенные на двухступенчатых триггерах *M-S*-типа, которые перебрасываются отрицательным перепадом синхроимпульса, работают аналогично счетчикам на триггерах с инверсным динамическим синхровходом.

2.2.2 Реверсивные счетчики

Счетчики, способные работать как в прямом, так и в обратном направлении, называются реверсивными. При построении реверсивного счетчика изменение направления счета достигается переключением межразрядных связей. На рисунке 2.13 приведен пример построения асинхронного реверсивного счетчика.

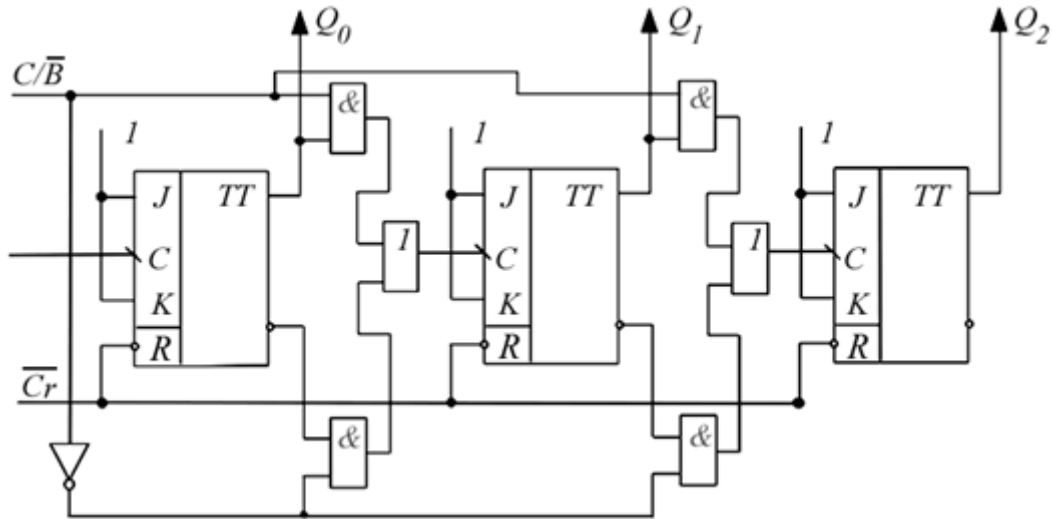


Рисунок 2.13 – Реверсивный асинхронный счетчик

В зависимости от сигнала на управляющем входе C/\bar{B} счетчик может осуществлять счет в прямом или обратном направлении.

2.2.3 Счетчики с произвольным модулем счета

Рассмотренные счетчики имеют модуль счета $\text{mod} = 2^N(2, 4, 6, 8, 16\dots)$. Однако при проектировании цифровых устройств часто возникает необходимость построить счетчик с модулем счета $\text{mod} \neq 2^N$. Принцип построения таких счетчиков состоит в исключении избыточных состояний с помощью включения обратных связей внутри счетчика либо методом управляемого сброса в нуль, когда устанавливается определенное состояние.

Итак, если нужно получить счетчик с каким-то модулем счета mod , то число необходимых при этом триггеров определяется исходя из $\text{mod} \leq 2^N$.

Например, при $N = 4$ может быть получен любой mod в пределах от 9 до 16. Если необходимо получить счетчик с $\text{mod} = 10$, то шесть состояний не используются. При этом если счет осуществляется в натуральном $B\bar{C}D$ -коде, то счетчик последовательно проходит состояния (рисунок 2.14) подобно двоичному счетчику, за исключением того, что после состояния 1001 (код десятичного числа девять) следующим состоянием будет 0000 (код десятичного числа нуль).

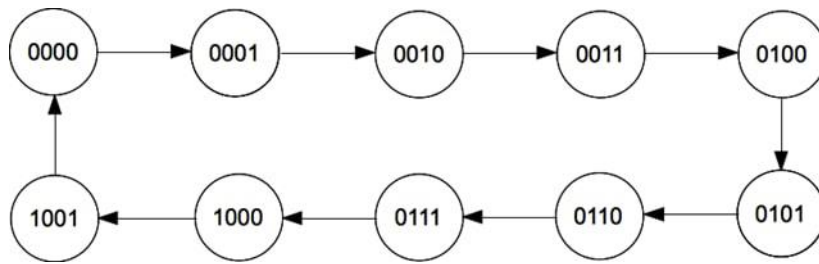
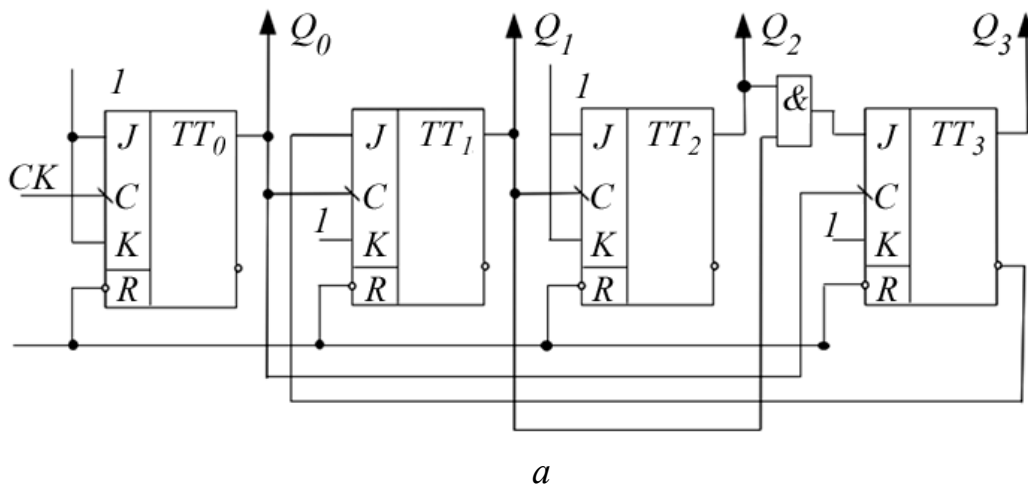


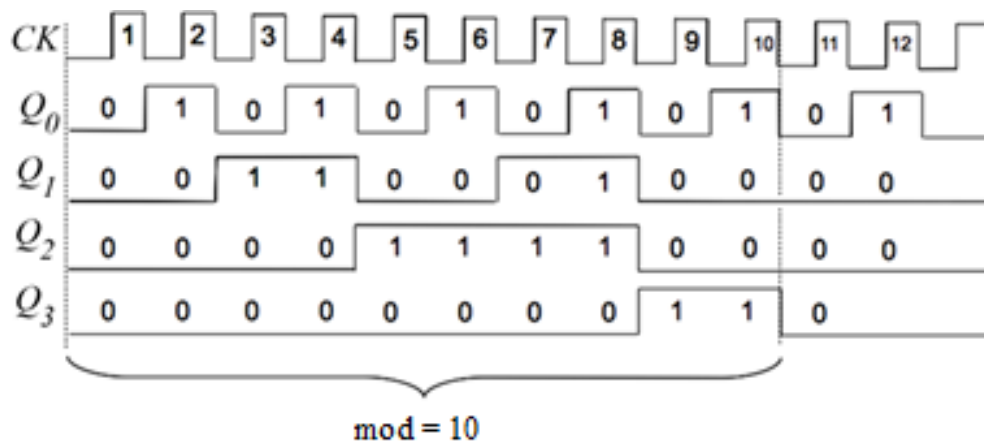
Рисунок 2.14 – Граф переходов двоично-десятичного счетчика

Синтез десятичного асинхронного счетчика, или любого другого асинхронного счетчика с $\text{mod} \neq 2^N$, или счетчика, работающего не в прямом двоичном коде, является сложной проблемой, поскольку не существует прямой процедуры синтеза.

Рассмотрим в качестве примера построение и работу двоично-десятичного асинхронного счетчика (рисунок 2.15).



a



б

a – логическая схема счетчика; *б* – временные диаграммы
Рисунок 2.15 – Асинхронный двоично-десятичный счетчик

Триггеры в счетчике (см. рисунок 2.15) перебрасываются с отрицательным перепадом синхросигнала. В схеме счетчика выход Q_0 соединен с синхровходами двух триггеров: TT_1 и TT_3 , а вход Q_1 соединен с синхровходом TT_2 . Входы J и K соединены либо постоянно с логической единицей, либо с выходами триггеров, как показано на рисунке 2.15, а. Работа счетчика может быть объяснена исходя из условий переброса триггеров. Вспомним, что если синхроимпульс имеет переход из единицы в нуль, триггер устанавливается в единичное состояние при $J = 1$ и $K = 0$, обнуляется при $J = 0$ и $K = 1$, меняет свое состояние на противоположное при $J = K = 1$ и не изменяет свое состояние при $J = K = 0$.

Из схемы рисунка 2.15, а очевидно, что:

- выход Q_0 меняет свое состояние с каждым отрицательным перепадом импульсов $СК$;

- выход Q_1 меняет свое состояние, если $Q_3 = 0$ и Q_0 имеет переход из единицы в нуль;

- выход Q_2 меняет свое состояние всякий раз, когда Q_1 имеет переход из единицы в нуль;

- выход Q_3 меняет свое состояние, когда $Q_1Q_2 = 1$ и Q_0 имеет переход из единицы в нуль. Выход Q_3 обнуляется, если Q_1 или Q_2 равны нулю и Q_0 имеет переход из единицы в нуль.

Временные диаграммы, поясняющие работу двоично-десятичного асинхронного счетчика, приведены на рисунке 2.15, б.

2.2.4 Интегральные схемы асинхронных счетчиков

Ряд асинхронных счетчиков выпускается в виде интегральных схем средней степени интеграции. Все ИС состоят из четырех $M-S$ -триггеров. Загрузка, установка и сброс (обнуление) осуществляются асинхронно, т. е. независимо от импульсов синхронизации.

В зависимости от особенностей счетчиков, связанных с загрузкой, установкой и сбросом, эти счетчики условно делятся на три группы.

К первой группе относится асинхронный счетчик 1533 ИЕ2, который имеет входы установки и сброса. Блок-схема счетчика приведена на рисунке 2.16.

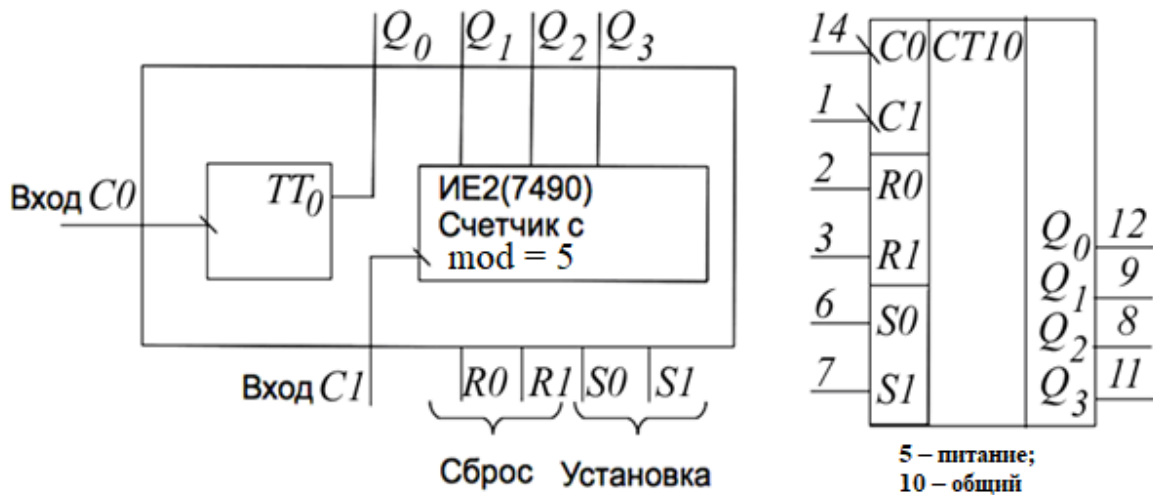


Рисунок 2.16 – Асинхронный счетчик ИЕ2

Счетчик состоит из четырех триггеров, объединенных внутри, как счетчик по $\text{mod} = 2$ и счетчик по $\text{mod} = 5$. Эти счетчики могут использоваться независимо или в комбинации. Если выход счетчика с $\text{mod} = 2$ (Q_0) соединить со входом счетчика по $\text{mod} = 5$, то образуется двоично-десятичный счетчик. Если соединить выход счетчика с $\text{mod} = 5$ (Q_3) со входом счетчика по $\text{mod} = 2$, то образуется счетчик-делитель на 10. В счетчике имеются два входа: «Сброс» R_0 и R_1 , на которые необходимо подать логическую единицу для обнуления счетчика. Когда на оба входа «Установка» S_0 и S_1 подается логическая единица, двоично-десятичный счетчик устанавливается в состояние 1001.

Внутренняя схема и временные диаграммы работы счетчика ИЕ2 соответствуют двоично-десятичному счетчику, рассмотренному ранее (см. рисунок 2.15).

Ко второй группе асинхронных счетчиков можно условно отнести счетчики, которые имеют только входы сброса. Блок-схема счетчиков 155ИЕ4, 1533ИЕ5 приведена на рисунке 2.17.

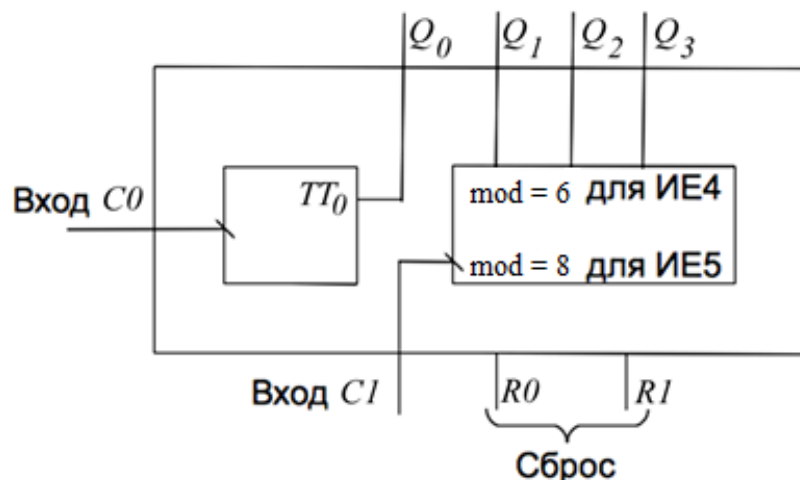


Рисунок 2.17 – Блок-схема асинхронных счетчиков ИЕ4, ИЕ5

Работа этого счетчика аналогична работе ИЕ2, за исключением того, что входы установки отсутствуют и счетчик по $\text{mod} = 6$ осуществляет счет не в натуральном двоичном коде. Эта последовательность дана в таблице 2.5.

Таблица 2.5 – Последовательность переключения счетчиков с $\text{mod} = 6$

Q_3	Q_2	Q_1
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0

ИС555ИЕ20 (74LS390) содержит два двоично-десятичных счетчика, подобных ИЕ2. Для каждого счетчика имеется один вход «Сброс» (R). ИС1533ИЕ19 (74ALS393) содержит два четырехразрядных двоичных счетчика с входом «Сброс» (R).

К третьей группе асинхронных счетчиков относятся счетчики 555ИЕ14 и 555ИЕ15, являющиеся версиями счетчиков ИЕ2 и ИЕ5 с предварительной установкой состояния. Блок-схема счетчиков показана на рисунке 2.18.

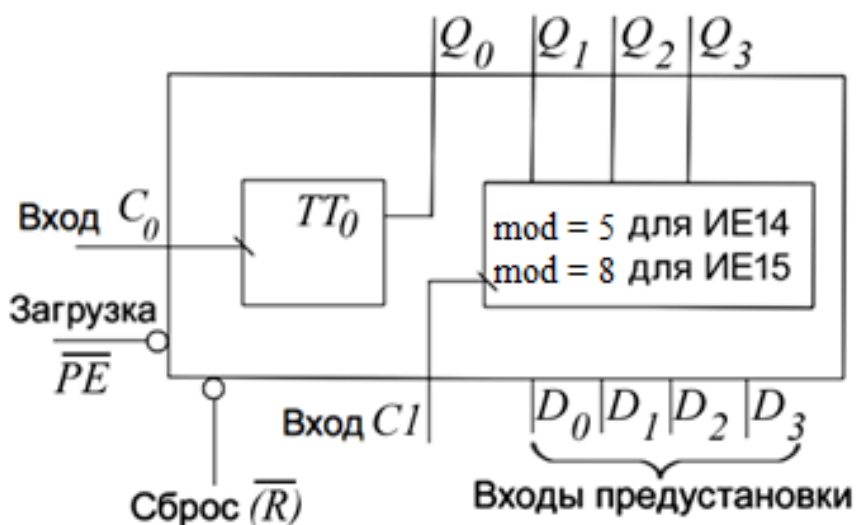


Рисунок 2.18 – Блок-схема асинхронных счетчиков ИЕ14, ИЕ15

Счетчики обнуляются при подаче логического нуля на вход \bar{R} . Подача логического нуля на вход \bar{PE} (в это время на входе «Сброс» должно быть $\bar{R} = 1$) останавливает счет, и в счетчик загружается число, установленное на входах предустановки. В режиме счета на оба входа «Сброс» и «Загрузка» должны быть поданы логические единицы.

Условное обозначение и цоколевка ИС ИЕ4, ИЕ5, ИЕ14, ИЕ15 приведены на рисунке 2.19.

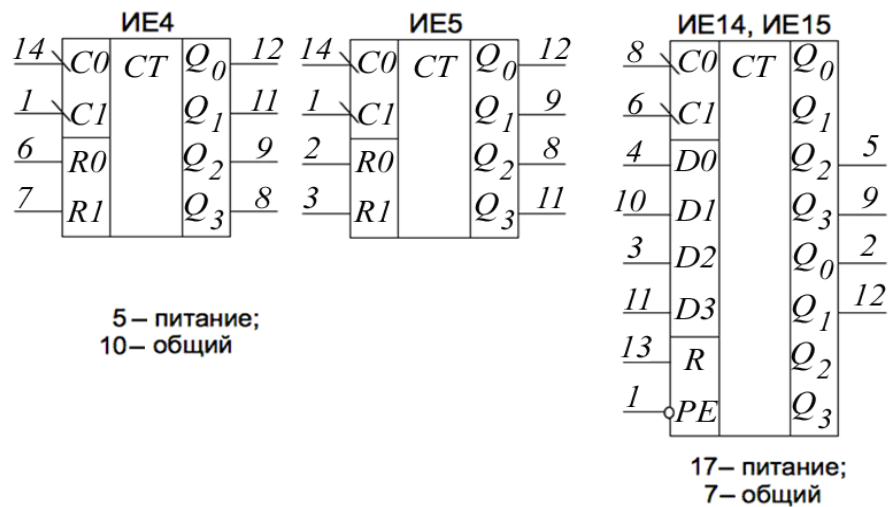
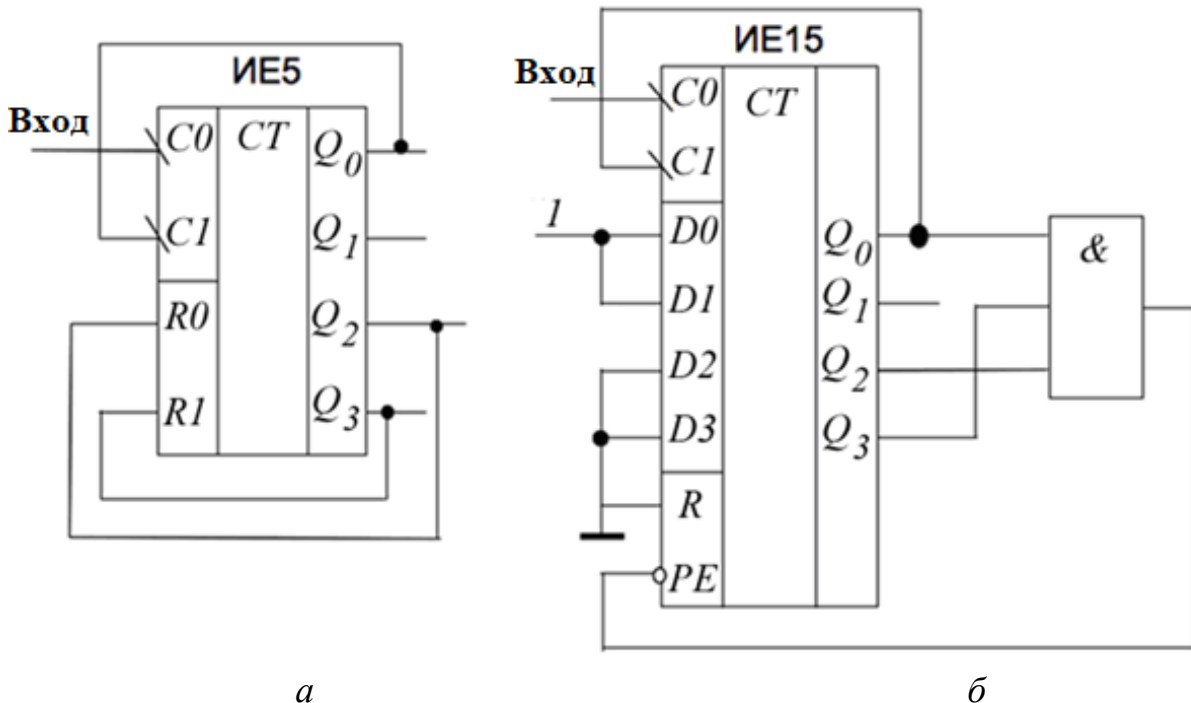


Рисунок 2.19 – Условные обозначения ИС ИЕ4, ИЕ5, ИЕ14, ИЕ15

Входы асинхронных счетчиков «Загрузка», «Установка», «Сброс» могут использоваться для изменения модуля и последовательности счета. Например, на рисунке 2.20, а приведена схема счетчика с $\text{mod} = 12$, построенная на основе ИС ИЕ5. На рисунке 2.20, б приведена схема десятичного счетчика, последовательность состояний которого соответствует коду с избытком 3.



а – для счетчика ИЕ5; б – для счетчика ИЕ15

Рисунок 2.20 – Примеры использования интегральных счетчиков

2.2.5 Синхронные счетчики

Максимальное время переключения асинхронного счетчика, когда выход счетчика последовательно изменяется 11...1 на 00...0, ограничивает его быстродействие. Быстродействие можно повысить, если все триггеры счетчика тактировать одновременно. В этом случае счетчик называется синхронным. Основу синхронного счетчика составляют, как правило, синхронные триггеры с динамическим управлением. Все синхровходы триггеров объединены и образуют счетный вход счетчика. Количество триггеров зависит от модуля счета ($\text{mod} < 2^N$). В задачу синтеза синхронного счетчика входит определение связей управляющих входов триггеров и их выходов, способствующих тому, чтобы триггеры переключались в соответствии с заданной последовательностью состояний счетчика. Синхронные счетчики могут быть синтезированы с использованием системных методов. Однако, прежде чем изучать такой метод, рассмотрим интуитивный метод. Для этого воспользуемся таблицей 2.6, в которой даны счетная последовательность и соответствующие состояния счетчика для $\text{mod} = 8$.

Таблица 2.6 – Счетная последовательность

Счет	Состояния счетчика		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Из таблицы 2.6 видно, что выход Q_0 младшего триггера счетчика изменяется с каждым счетным импульсом. Это может быть достигнуто, если использовать T -триггер с $T = 1$. Выход Q_1 изменяет свое состояние всякий раз, когда Q_0 изменяется с единицы на нуль. Поэтому если соединить выход Q_0 с входом T_1 , то Q_1 будет изменять свое состояние, когда $Q_0 = 1$ ($T_1 = Q_0 = 1$), и будет оставаться без изменения, когда $T_1 = Q_0 = 0$. Из таблицы 2.6 также видно, что Q_2 изменяет свое состояние всякий раз, когда и Q_1 , и Q_0 равны единице. Это может быть реализовано, если вход наиболее значащего триггера $TT_2 = Q_1 \cdot Q_0$. В результате получим схему синхронного счетчика ($\text{mod} = 8$), рисунок 2.21. Аналогичным образом могут быть построены синхронные счетчики с $\text{mod} = 2N$.

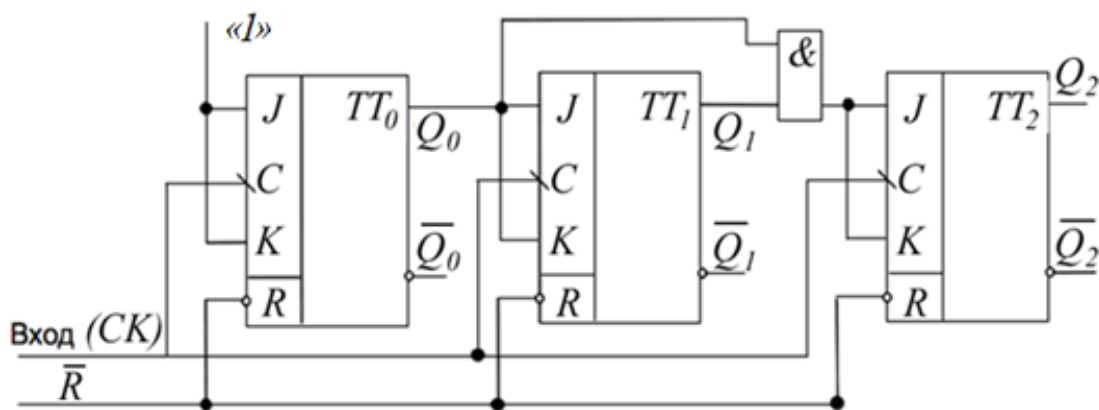


Рисунок 2.21 – Синхронный счетчик с mod = 8

2.2.6 Синтез синхронных счетчиков

Синхронный счетчик для любой заданной последовательности счета и модуля счета может быть синтезирован следующим образом:

1) необходимо определить требуемое число триггеров исходя из

$$N \geq \log_2 m,$$

где m – модуль счета;

2) записать счетную последовательность в табличном виде;

3) определить состояния входов триггеров, которые должны соблюдаться для перехода в требуемые следующие состояния исходя из настоящего состояния и таблицы переходов триггеров;

4) приготовить карты Карно для каждого входа триггеров в терминах выходов триггеров как входных переменных. Используя метод Карно, получить минимизированные выражения для каждого входа триггеров;

5) построить схему счетчика, используя триггеры и логические элементы, в соответствии с минимизированными выражениями;

Рассмотрим примеры построения счетчиков и делителей.

Пример 1. Построить двоично-десятичный счетчик, который имеет десять состояний.

Решение. Для этого необходимо ($N \geq \log_2 10$) $N = 4$ триггера. Построим таблицу состояний и таблицу истинности для входов триггеров (таблица 2.7).

Таблица 2.7 – Состояния двоично-десятичного счетчика

Состояния счетчика	Входы триггеров											
	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0 0 0 0	0	0	0	0	0	x	0	x	0	x	1	x
0 0 0 1	0	0	0	1	0	x	0	x	1	x	x	1
0 0 1 0	0	0	1	0	0	x	0	x	x	0	1	x
0 0 1 1	0	0	1	1	0	x	1	x	x	1	x	1
0 1 0 0	0	1	0	0	0	x	x	0	0	x	1	x
0 1 0 1	0	1	0	1	0	x	x	0	1	x	x	1
0 1 1 0	0	1	1	0	0	x	x	0	x	0	1	x
0 1 1 1	0	1	1	1	1	x	x	1	x	1	x	1
1 0 0 0	1	0	0	0	x	0	0	x	0	x	1	x
1 0 0 1	1	0	0	1	x	1	0	x	0	x	x	1
0 0 0 0												

Используя метод Карно, получим минимизированные выражения для всех входов J и K (рисунок 2.22). Используя эти выражения, строим синхронный двоично-десятичный счетчик (рисунок 2.23).

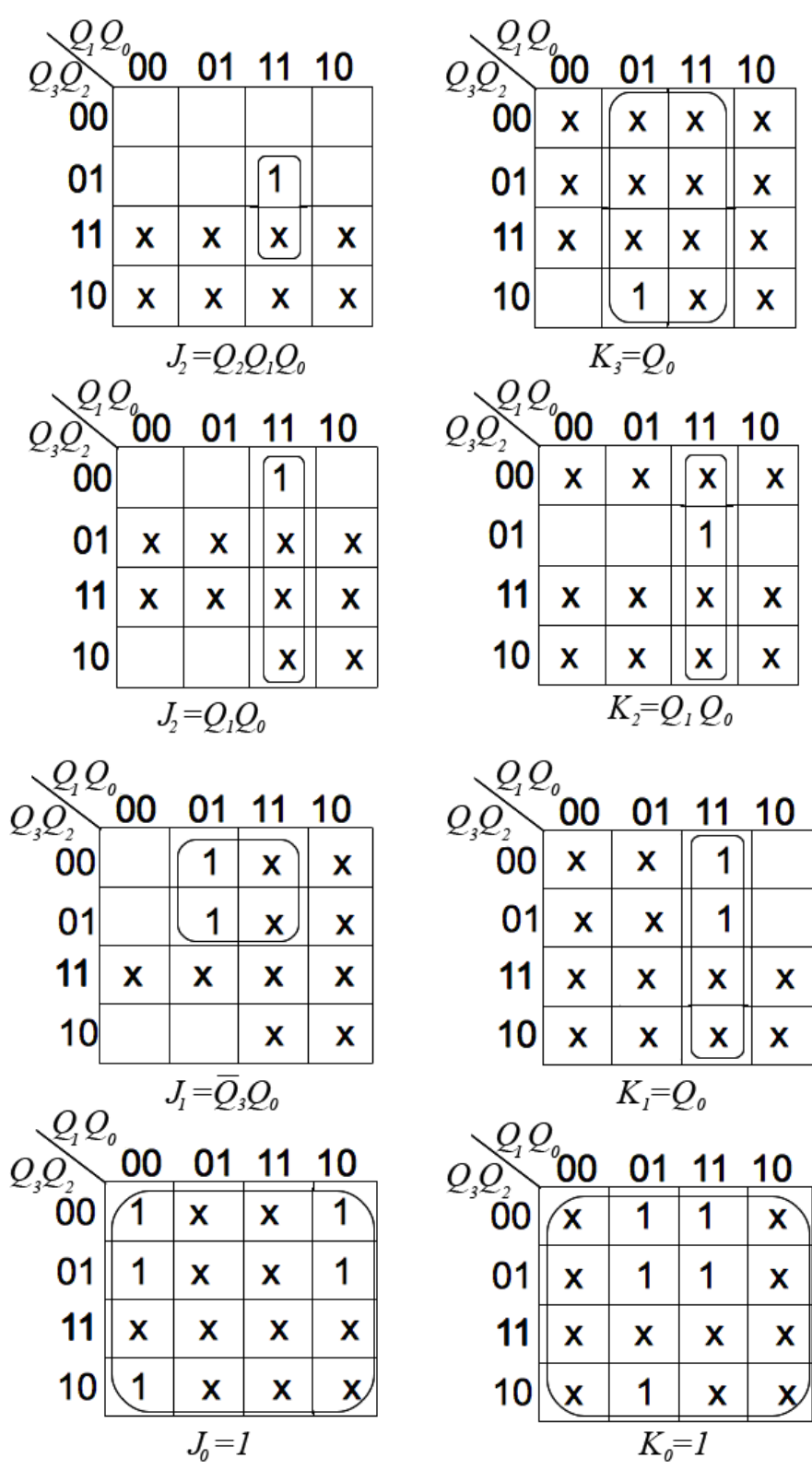


Рисунок 2.22 – Минимизация функций J и K с помощью карт Карно

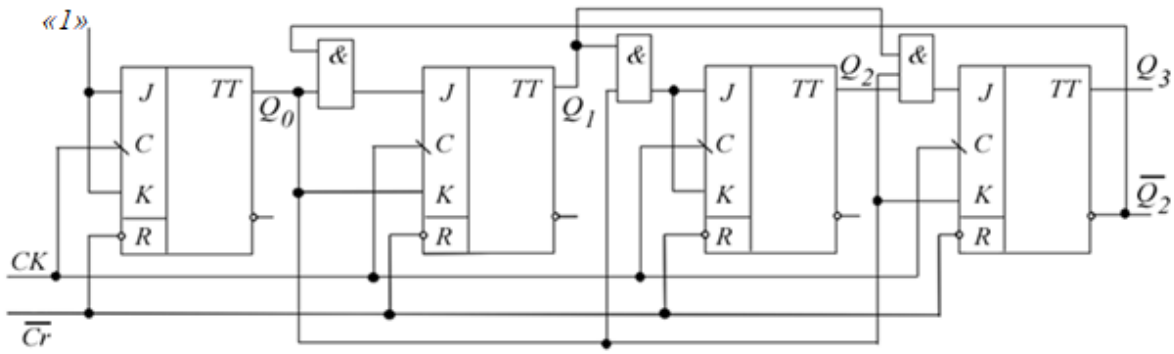


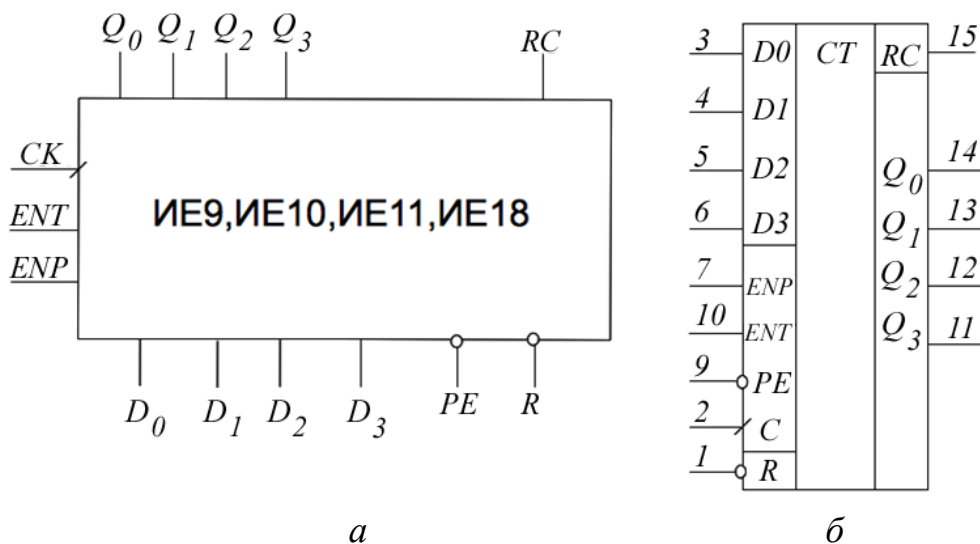
Рисунок 2.23 – Синхронный двоично-десятичный счетчик

2.2.7 Интегральные схемы синхронных счетчиков

Мы рассмотрели синтез синхронных счетчиков. Счетчики с любой счетной последовательностью и любым модулем счета могут быть синтезированы с использованием этого метода. На промышленных предприятиях выпускаются синхронные счетчики в виде интегральных схем средней степени интеграции.

Все счетчики содержат по четыре триггера с динамическим синхровходом. Изменения состояний триггеров, синхронная загрузка и синхронное обнуление происходят путем положительного перепада синхроимпульсов. В силу некоторых характерных особенностей синхронные счетчики могут быть подразделены на четыре группы.

Синхронные счетчики группы 1. К этой группе относятся счетчики ИЕ9, ИЕ10, ИЕ11, ИЕ18. Это двоично-десятичные и двоичные суммирующие счетчики с синхронной загрузкой и асинхронным сбросом. Блок-диаграмма, условное обозначение и цоколевка этих счетчиков приведены на рисунке 2.24.



a – блок-диаграмма; *б* – условное обозначение ИЕ9, ИЕ10, ИЕ11, ИЕ18

Рисунок 2.24 – Синхронные счетчики группы 1

Эти счетчики имеют два отдельных разрешающих входа *ENT* и *ENP*. При подаче на любой из этих входов логического нуля счет останавливается асинхронно.

Выход последовательного переноса *RC* находится в состоянии логического нуля и становится логической единицей всякий раз, когда счетчик достигает своего максимального значения (двоичное число 9 для двоично-десятичного счетчика и двоичное 15 для двоичного счетчика). Функционирование счетчиков группы 1 отражено в таблице 2.8.

Таблица 2.8 – Таблица функционирования счетчиков группы 1

<i>L</i>	<i>ENP</i>	<i>ENT</i>	<i>Cr</i>	<i>СК</i>	Операция
0	<i>x</i>	<i>x</i>	1	↑	Установка (загрузка)
1	0	1	1	<i>x</i>	Остановка счета
1	<i>x</i>	0	1	<i>x</i>	Остановка счета, невозможность <i>RC</i>
<i>x</i>	<i>x</i>	<i>x</i>	0	*	Сброс в нуль
1	1	1	1	↑	Прямой счет
<i>Примечание</i> – <i>x</i> для ИЕ9, ИЕ10; ↑ для ИЕ11, ИЕ18.					

Пример 2. Построить делитель на 11, используя ИЕ18. Использовать *RC*-выход и установочные входы.

Решение. Для получения делителя на 11 счетчик устанавливается в двоичное состояние 0101 (десятичное 5). Когда счет достигнет 1111, на выходе *RC* появляется логическая единица, которая используется для загрузки данных с установочных входов в счетчик (рисунок 2.25).

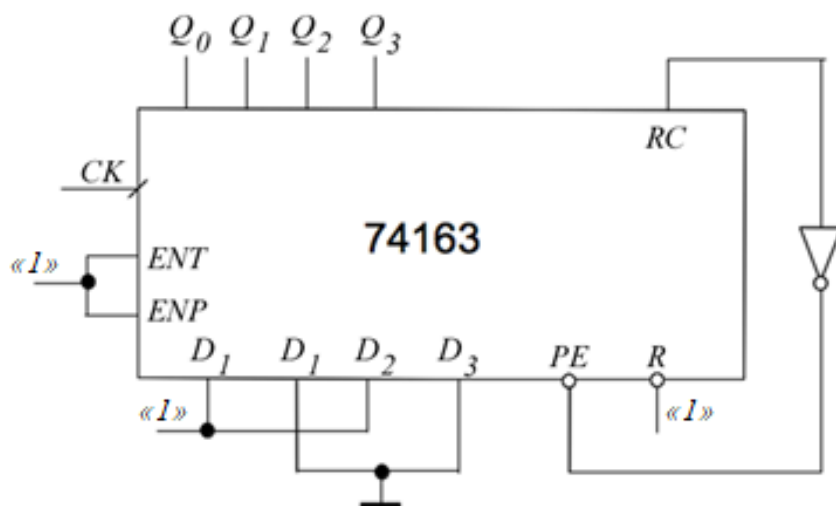


Рисунок 2.25 – Схема делителя на 11

Итак, для получения делителя на m на установочных входах должно быть: $D = 16_m$ для двоичного счетчика и $D = 10_m$ для десятичного счетчика. Входы ENT , ENP и RC могут использоваться для организации каскадного соединения счетчиков (рисунок 2.26).

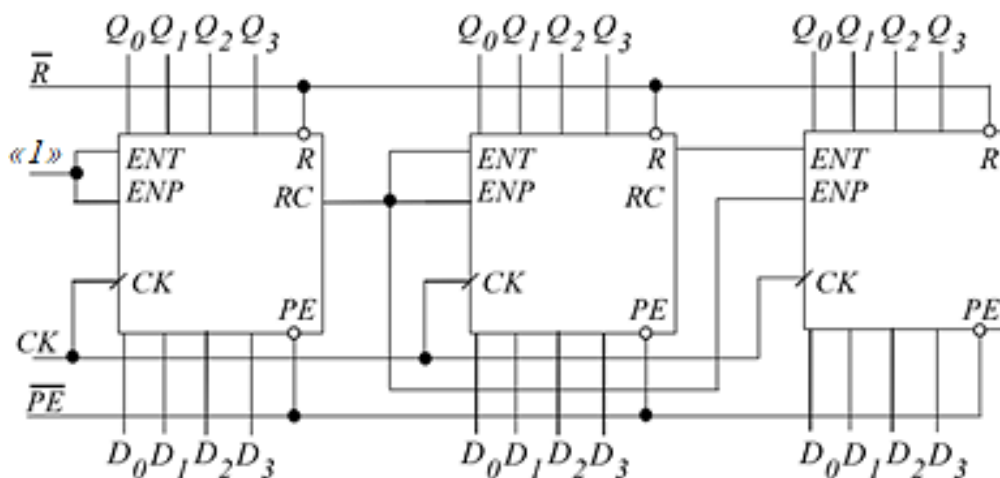
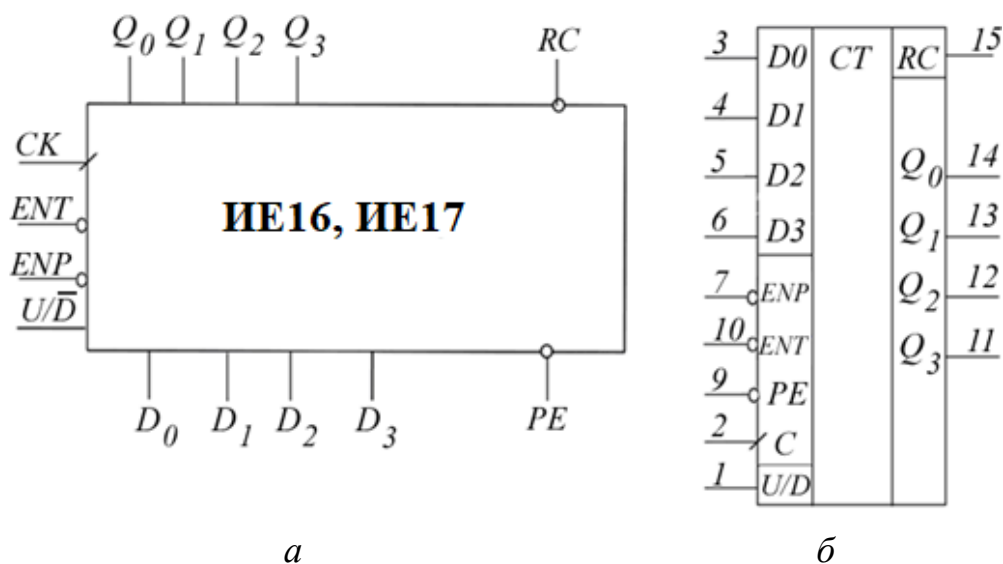


Рисунок 2.26 – Каскадное соединение счетчиков первой группы

Синхронные счетчики группы 2. К этой группе относятся счетчики ИЕ16 и ИЕ17. Это соответственно двоично-десятичный и двоичный реверсивные счетчики с синхронной установкой и без сброса (рисунок 2.27). Функция ENT и ENP та же, что и у счетчиков первой группы, но в этом случае входы активны при низком логическом уровне. Выход последовательного переноса в нормальном состоянии равен логической единице и переходит в логический ноль, когда счет достигает максимального значения при прямом счете или когда счет достигает минимального значения при обратном счете.



a – блок-диаграмма; *б* – условное обозначение ИЕ16, ИЕ17
Рисунок 2.27 – Синхронные счетчики группы 2

Функционирование счетчиков ИЕ16 и ИЕ17 отражено в таблице 2.9.

Таблица 2.9 Таблица функционирования счетчиков группы 2

PE	ENP	ENT	U/D	CK	Операция
0	x	x	x	\uparrow	Установка
1	1	0	x	x	Остановка счета
1	x	1	x	x	Остановка счета, запрет переноса
1	0	0	1	\uparrow	Прямой счет
1	0	0	0	\uparrow	Обратный счет

Сигнал на входе U/\bar{D} определяет направление счета: $U/D = 1$ для прямого счета и $U/\bar{D} = 0$ для обратного. В счетчиках этой группы отсутствует вход сброса R . Поэтому, если необходимо остановить счет до достижения максимального значения, схема И – НЕ должна быть использована для детектирования состояния, соответствующего требуемому числу, и выход схемы И – НЕ соединяется с входом установки \overline{PE} . Входы установки дают необходимое начальное состояние счетчика.

Пример 3. Построить счетчик с начальным состоянием 0011 и конечным 1100, используя ИЕ17.

Решение. Когда счетчик (рисунок 2.28) достигнет состояния 1100, на выходе элемента И – НЕ и на входе PE появится логический нуль. Следующим синхриимпульсом (положительным перепадом) счетчик установится в состояние 0011. Необходимо учитывать, что установка счетчика осуществляется синхронно.

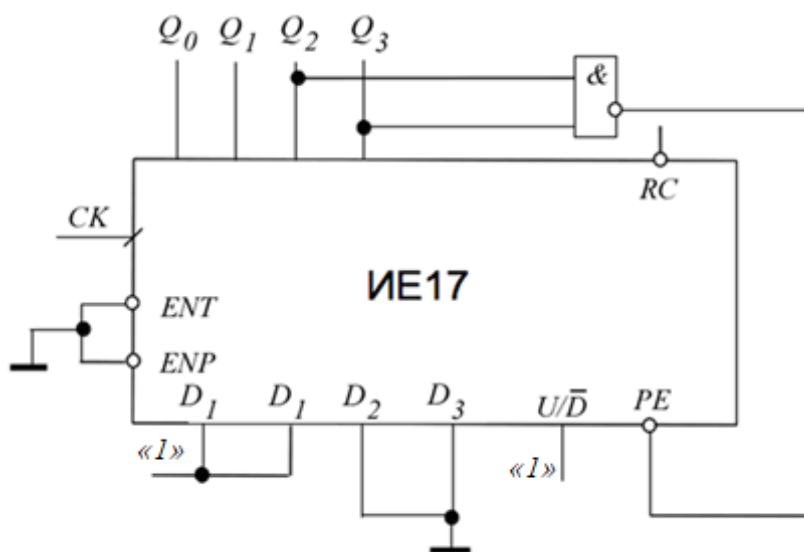
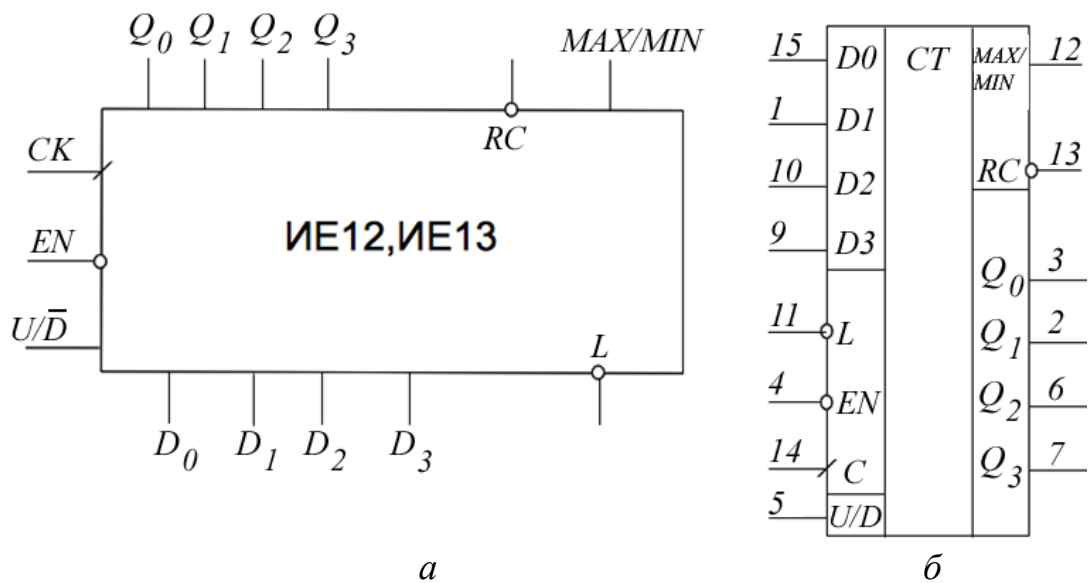


Рисунок 2.28 – Двоично-десятичный счетчик, осуществляющий счет в коде с избытком 3

Каскадное соединение счетчиков второй группы осуществляется аналогично каскадному соединению счетчиков первой группы.

Синхронные счетчики группы 3. К этой группе относятся счетчики ИЕ12 и ИЕ13. Это соответственно двоично-десятичный и двоичный реверсивные счетчики с асинхронной установкой и без сброса. Эти счетчики имеют только один низкий активный разрешающий вход *EN*. Выход *MAX/MIN* (рисунок 2.29) используется для определения максимального или минимального состояния счетчика. На этом выходе нормально логический нуль, а логическая единица появляется, когда максимальное состояние счетчика 1001 для ИЕ12 и 1111 для ИЕ13 при прямом счете или когда минимальное состояние 0000 при обратном счете.



a – блок-диаграмма; *б* – условное обозначение ИЕ12, ИЕ13
Рисунок 2.29 – Синхронные счетчики группы 3

Выход *RC* нормально в состоянии логической единицы и переходит в логический нуль, когда счетчик достигает *MAX/MIN* точки и импульс синхронизации становится равным нулю. Функционирование счетчиков ИЕ12, ИЕ13 отражено в таблице 2.10.

Таблица 2.10 – Таблица функционирования счетчиков группы

<i>L</i>	<i>ENAB</i>	<i>U/D</i>	<i>CK</i>	Операция
<i>x</i>	1	<i>x</i>	<i>x</i>	Остановка счёта
0	0	<i>x</i>	<i>x</i>	Установка счетчика
1	0	0	↑	Прямой счёт
1	0	1	↑	Обратный счёт

Синхронные счетчики третьей группы могут каскадироваться различными путями. Максимальное быстродействие возникает, если использовать параллельный перенос (рисунок 2.30).

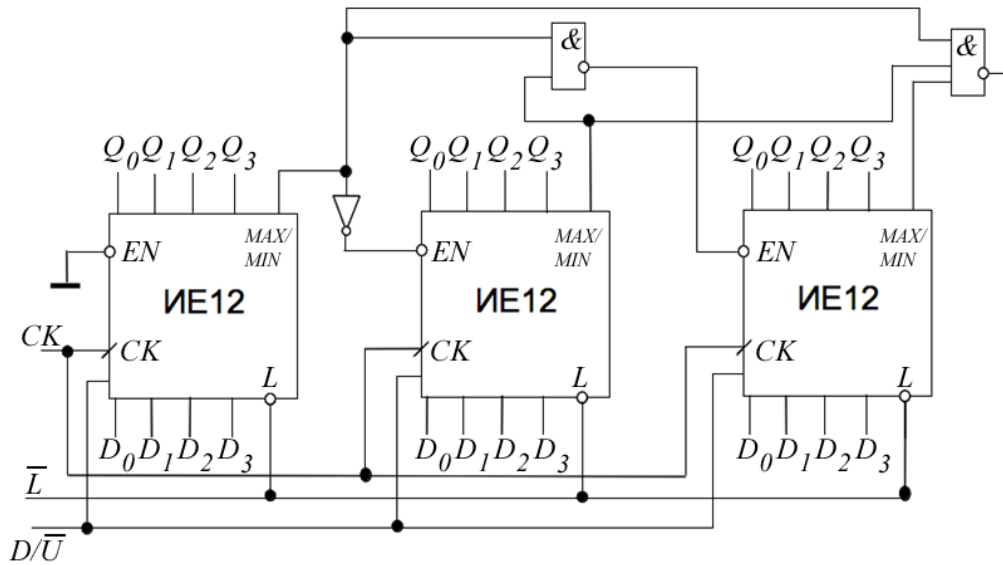


Рисунок 2.30 – Каскадные соединения счетчиков третьей группы

Синхронные счетчики группы 4. К этой группе относятся счетчики ИЕ6 и ИЕ7. Это соответственно двоично-десятичный и двоичный реверсивные счетчики с асинхронной загрузкой и сбросом. У этих счетчиков (рисунок 2.31) импульсы для прямого счета подаются на вход $C-U$, при этом вход $C-D$ соединяется с логической единицей и импульсы для обратного счета подаются на вход $C-D$, при этом вход $C-U$ соединяется с логической единицей.



а – блок-диаграмма; *б* – условное обозначение ИЕ6 и ИЕ7
Рисунок 2.31 – Синхронные счетчики группы 4

Выход C (перенос) и B (заем) обычно находятся в состоянии логической единицы. Выход переноса переходит в логический нуль, когда счетчик достигает максимального значения при прямом счете и вход $C-U$ находится в логическом нуле. Выход заема остается в логической единице, когда счетчик работает от прямого $C-U$ входа.

Функция заема при обратном счете аналогична переносу при прямом счете.

Функционирование счетчиков ИЕ6 и ИЕ7 отражено в таблице 2.11.

Таблица 2.11 – Таблица функционирования счетчиков группы 4

L	R	$C-U$	$C-D$	Операция
x	1	x	x	Сброс
1	0	\uparrow	1	Прямой счет
1	0	\uparrow	\uparrow	Обратный счет
0	0	x	x	Установка
1	0	1	1	Остановка счета

При каскадировании этих счетчиков перенос и заем каждой ступени должны быть соединены с входами $C-U$ и $C-D$ последующей ступени соответственно. Для управления импульсами на прямой и обратный счетный вход может быть использована схема, приведенная на рисунке 2.32.

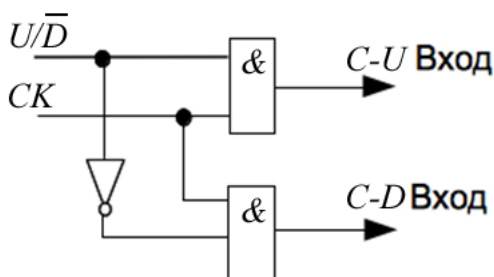


Рисунок 2.32 – Схема для управления счетными импульсами при прямом и обратном счете

3 СИНТЕЗ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ (КОНЕЧНЫХ ЦИФРОВЫХ АВТОМАТОВ)

3.1 Введение

Логические схемы подразделяются на два класса: комбинационные схемы, или автоматы без памяти, и последовательностные схемы, или автоматы с памятью. Последовательностная схема состоит из логических и запоминающих элементов. Значения выходных сигналов последовательностных схем зависят как от текущих значений входных сигналов, так и от значений входных и выходных сигналов в предыдущие моменты времени.

Синтез последовательностных схем происходит в два этапа:

- абстрактный синтез;
- структурный синтез.

Соответственно существуют абстрактные и структурные модели цифровых автоматов.

Абстрактным автоматом называют дискретный преобразователь информации с конечным входным алфавитом $Z = \{z_1, \dots, z_f, \dots, z_F\}$, конечным выходным алфавитом $W = \{w_1, \dots, w_g, \dots, w_G\}$, конечным множеством внутренних состояний $A = \{a_1, \dots, a_m, \dots, a_M\}$ и двумя характеристическими функциями: функцией переходов δ и функцией выходов λ [1]. Алфавитом называют конечное множество попарно различных символов, которые называют также буквами этого алфавита. Абстрактный автомат имеет один входной и один выходной каналы (рисунок 3.1).

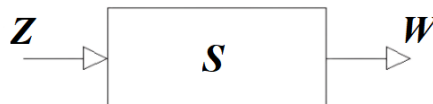


Рисунок 3.1 – Абстрактный автомат

Функционирование автомата происходит в дискретные моменты автоматного времени, $t = 0, 1, 2, \dots$. В каждый дискретный момент времени t автомат находится в определенном состоянии: $a(t) = a_m \in A$, воспринимает на входном канале некоторую букву входного алфавита: $z(t) = z_f \in Z$ (входной сигнал), выдает на выходном канале некоторую букву выходного алфавита: $w(t) = w_g \in W$ (выходной сигнал), определяемую функцией выходов λ как

$$w(t) = \lambda(a(t), z(t))$$

или

$$w_g = \lambda(a_m, z_f), \quad (3.1)$$

переключается в новое состояние $a(t+1) = a_s \in A$, которое определяется функцией переходов δ как

$$a(t + 1) = \delta(a(t), z(t))$$

или

$$a_s = \delta(a_m, z_f). \quad (3.2)$$

Автомат называется конечным, если конечны множества A , Z , W . Автомат называется полностью определенным, если для любой пары $(a_m, z_f) \in A \cdot Z$ определены функции δ и λ . У частичного автомата функции δ и λ определены не для всех пар $(a_m, z_f) \in A \cdot Z$. При рассмотрении функционирования автомата считается, что исходным состоянием в момент $t = 0$ есть $a(0) = a_1$, которое называется начальным состоянием.

Если выходной символ $w(t)$ автомата не зависит от входного символа $z(t)$, а определяется лишь внутренним состоянием автомата в момент времени t , то такой автомат называется автоматом Мура и закон функционирования задается уравнениями

$$\begin{aligned} a(t + 1) &= \delta(a(t), z(t)), \\ w(t) &= \lambda(a(t)). \end{aligned} \quad (3.3)$$

Если выходной символ $w(t)$ автомата определяется как внутренним состоянием автомата, так и значением символа $z(t)$ в момент времени t , то такой автомат называется автоматом Мили и закон функционирования задается уравнениями

$$\begin{aligned} a(t + 1) &= \delta(a(t), z(t)), \\ w(t) &= \lambda(a(t), z(t)). \end{aligned} \quad (3.4)$$

Наиболее наглядным способом задания закона функционирования автомата является задание в виде графа. Граф автомата изображается в виде совокупности вершин, каждой из которых ставится в соответствие одно из внутренних состояний автомата и дуг, соединяющих вершины графа и соответствующих переходам между состояниями. Направленность графа позволяет проследить последовательность переключений состояний автомата при подаче на его вход последовательности букв входного алфавита. В каждую вершину графа вписывается символ внутреннего состояния, которому соответствует данная вершина.

Если в автомате Мили под действием входной буквы $z_f \in Z$ осуществляется переход из состояния $a_m \in A$ в состояние $a_s \in A$ с выдачей выходной буквы $w_g \in W$, то вершины графа a_m (исходное состояние) и a_s (состояние перехода) соединяются дугой, направленной от a_m к a_s . Дуга отмечается парой символов (z_f, w_g) , как показано на рисунке 3.2.

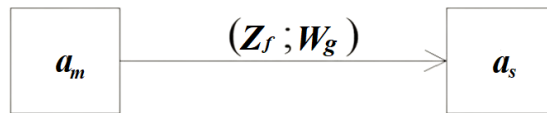


Рисунок 3.2 – Дуга между вершинами графа

Закон функционирования автоматов может задаваться также в виде таблиц переходов и выходов.

Для этого вначале устно формулируется задача, детализируются имеющиеся входы и требуемые выходы. Эту информацию можно изобразить в виде структурной схемы и привести временные диаграммы входных и выходных сигналов.

Точно сформулировать задачу трудно, и для этого обычно требуется несколько консультаций проектировщика с заказчиком. Если на этой стадии не будут полностью устранены неясности, процесс проектирования тем не менее следует продолжить, несмотря на то, что полученная аппаратная реализация схемы, возможно, не будет удовлетворять требованиям заказчика. Так процесс проектирования придется повторять несколько раз [3].

Что касается студентов, то неясности в устной формулировке задачи очень часто будят в них излишние сомнения. В стадии изучения вопроса синтеза цифровых автоматов такие сомнения не должны быть слишком серьезными. Студент должен настроиться на решение задачи и дать лишенную неточности версию ответа. Предложенный вариант ответа должен быть реализован. А при изучении поведения созданной схемы необходимо установить, подтверждается ли правильность решения.

Пример постановки задачи. Последовательностная схема имеет входной канал Z , вход синхронизации \perp и выходной канал W . Поступающие на вход Z данные делятся на группы, содержащие по четыре двоичные цифры, которые не перекрывают друг друга, проверяются логической схемой и при обнаружении комбинаций 0001, 0011, 0101 и 1001 на выходе W появляется единица. Сигнал на выходе должен иметь длительность, равную длительности синхронизирующего импульса \perp . При всех других комбинациях четырех двоичных цифр сигнал на выходе должен быть равен нулю [3].

Неопытный проектировщик может разработать древовидный граф (рисунок 3.3).

Движение по этому графу всегда будет начинаться из состояния a_1 . Из него можно попасть в состояние a_2 , если подается сигнал $\perp z_2$ ($z_1 = 0, z_2 = 1$, \perp – синхроимпульс), или в состояние a_3 , если подается сигнал $\perp z_1$. Из состояний a_2 и a_3 ведут два пути: один связан с сигналом перехода $\perp z_1$, а другой – с сигналом перехода $\perp z_2$. Всего имеется четыре пути, которые приводят к состояниям a_4, a_5, a_6, a_7 . Из состояний a_4, a_5, a_6, a_7 существует два пути, которые приведут к состояниям $a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}$.

Для каждого из этих восьми состояний существует два выходных пути, но так как следующий переход является четвертым по счету, то все шестнадцать выходных путей должны вести в начальное состояние a_1 .

Пусть $a_1 - a_3 - a_7 - a_{15} - a_1$, $a_1 - a_3 - a_7 - a_{14} - a_1$, $a_1 - a_3 - a_6 - a_{13} - a_1$ и $a_1 - a_2 - a_5 - a_{11} - a_1$ связаны соответственно с комбинациями 0001, 0011, 0101, 1001 и поэтому при переходах из состояний a_{11} , a_{13} , a_{14} и a_{15} в состояние a_1 должен генерироваться выходной сигнал, равный w_2 ($w_1 = 0, w_2 = 1$). Другие пути на графе состояний связаны с теми входными последовательностями, для которых сигнал на выходе схемы равен w_1 .

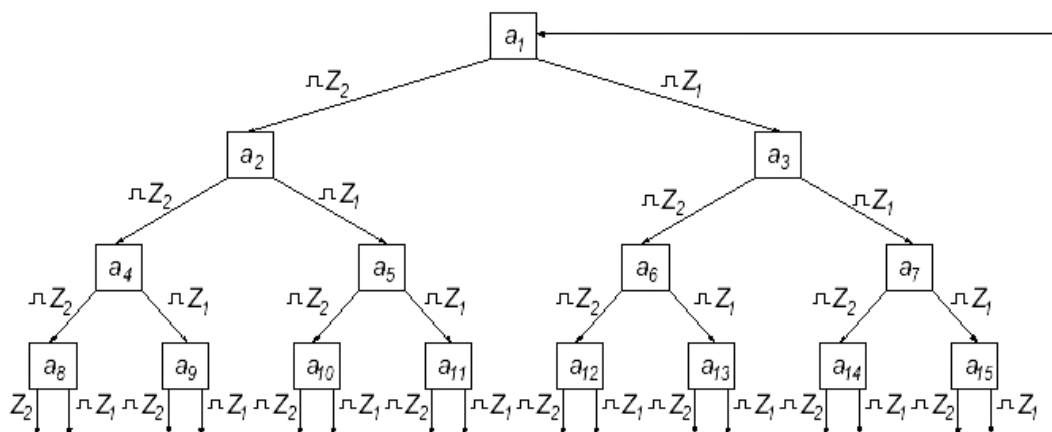


Рисунок 3.3 – Древоподобный граф

При разработке этого графа никаких сокращений не производилось. Каждая комбинация из четырех двоичных цифр отображена на графе.

При табличном задании закона функционирования автомата Мили используются таблицы переходов и таблицы выходов. Для данного примера автомат Мили, у которого $A = \{a_1, a_2, \dots, a_{15}\}$; $Z = \{z_1, z_2\}$; $W = \{w_1, w_2\}$, представлен таблицами 3.1 и 3.2.

Таблица 3.1 – Таблица переходов для закона функционирования автомата Мили

$z(t) \setminus a(t)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
z_1	a_3	a_5	a_7	a_9	a_{11}	a_{13}	a_{15}	a_1	a_1	a_1	a_1	a_1	a_1	a_1	a_1
z_2	a_2	a_4	a_6	a_8	a_{10}	a_{12}	a_{14}	a_1	a_1	a_1	a_1	a_1	a_1	a_1	a_1

Таблица 3.2 – Таблица выходов для закона функционирования автомата Мили

$z(t) \setminus a(t)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
z_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1
z_2	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_2	w_1	w_2	w_2	w_2

Входные сигналы и состояния, отмечающие строки и столбцы таблиц, относятся к моменту времени t , т. е. отражают $z(t)$ и $a(t)$. На пересечении столбца $a(t) = a_m$ и строки $z(t) = z_f$ в таблице переходов ставится состояние перехода $a(t+1) = a_s$, определяемое функцией переходов $a_s = \delta(a_m, z_f)$. В состоянии a_s автомат переключается из состояния a_m под действием сигнала z . В таблице выходов на пересечении столбца $a(t) = a_m$ и строки $z(t) = z$ ставится соответствующий этому переходу выходной сигнал $w(t) = w_g$, определяемый функцией выходов $w_g = \lambda(a_m, z_f)$.

Так же как и по графу, по таблицам 3.1 и 3.2 можно проследить последовательность работы автомата. В начальный момент $t = 0$ автомат находится в состоянии $a(0) = a_1$. Если на входе в момент $t = 0$ будет действовать, к примеру, буква $z(0) = z_2$, то автомат сформирует на выходе букву $w_1 = w(0) = \lambda(a(0), z(0))$ и в следующий момент времени $t = 1$ переключится в новое состояние $a_2 = a(1) = \delta(a(0), z(0))$. В момент времени $t = 1$ автомат, находясь в состоянии $a(1) = a_2$, может воспринять на входном канале любую букву из множества Z , например, $z(1) = z_2$. Тогда автомат сформирует на выходе букву $w_1 = w(1) = \lambda(a(1), z(1))$ и в следующий момент времени $t = 2$ переключится в новое состояние $a_4 = a(2) = \delta(a(1), z(1))$ и т. д. Таким образом, если на вход автомата, предварительно установленного в начальное состояние a_1 , подавать в последовательные моменты времени $t = 0, 1, 2, \dots$ буква за буквой некоторую последовательность букв входного алфавита, то в общем случае на выходе автомата будут последовательно появляться буквы выходного алфавита $w(0), w(1), w(2), \dots$ – выходное слово. При этом автомат будет последовательно переключаться в состояния $a(1), a(2), a(3), \dots$. Выходное слово называется также реакцией автомата на входное слово. Для рассматриваемого примера выходной сигнал, как и входной, разбивается на комбинации по четыре двоичных цифры. Причем первые три цифры во всех комбинациях всегда равны нулю, а четвертая, может быть единицей или нулем в зависимости от входной комбинации двоичных цифр.

Если функции переходов и выходов автомата определены не на всех упорядоченных парах символов $(a_m, z_f) \in A \cdot Z$, то автомат не полностью определенный, поэтому некоторые клетки таблиц будут пусты (в них ставятся прочерки).

Поскольку в автомате Мура выходной сигнал не зависит явно от входного сигнала, а зависит от состояния, автомат Мура задается отмеченной таблицей переходов, в которой каждый столбец обозначается состоянием $a(t) = a_m$ и выходным сигналом $w(t) = w_g = \lambda(a_m)$, соответствующим этому состоянию.

Получением графа автомата или таблиц переходов и выходов завершается первый этап абстрактного синтеза автомата. При этом для реализации одной и той же дискретной последовательной схемы могут быть получены автоматы с различным числом внутренних состояний. Абстрактные автоматы с

одинаковыми входными и выходными алфавитами, которые после установки их в начальное состояние a_1 на любое входное слово выдают одинаковые выходные слова, называются эквивалентными. Чем больше число внутренних состояний автомата, тем больше аппаратуры требуется для реализации схемы. По этой причине еще на этапе абстрактного синтеза необходимо провести минимизацию количества состояний автомата.

3.2 Минимизация абстрактного автомата

Рассмотрим один из методов минимизации. Основная идея этого метода состоит в разбиении всех состояний исходного абстрактного автомата на попарно непересекающиеся классы эквивалентных состояний и замене каждого класса эквивалентности одним состоянием. Таким образом, получающийся в результате минимальный автомат имеет столько состояний, на сколько классов эквивалентности разбиваются состояния исходного автомата [1; 2].

Для разбиения множества состояний исходного автомата на попарно непересекающиеся классы эквивалентных состояний сначала разбивают все состояния на классы одноэквивалентных состояний. Два состояния при этом разбиении считаются одноэквивалентными, если их реакции на всевозможные буквы входного алфавита (на всевозможные входные слова единичной длины) совпадают, т. е. соответствующие этим состояниям столбцы в таблице выходов совпадают. Далее производят разбиение состояний в пределах каждого класса одноэквивалентных состояний на классы двухэквивалентных состояний. При этом два одноэквивалентных состояния считаются двухэквивалентными, если они переводятся любым входным сигналом в одинаковые одноэквивалентные состояния. Далее производят разбиение состояний в пределах каждого класса двухэквивалентных состояний на классы трехэквивалентных состояний и т. д., пока этот процесс допустим [1; 2].

В качестве примера рассмотрим применение этого метода для минимизации автомата Мили, заданного таблицами переходов и выходов (таблицы 3.1 и 3.2).

По таблице выходов определяем множество Π_1 классов одноэквивалентных состояний путем объединения в эквивалентные классы столбцов с одинаковыми выходными сигналами $\Pi_1 = \{B_1, B_2\}$; $B_1 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{12}\}$; $B_2 = \{a_{11}, a_{13}, a_{14}, a_{15}\}$. Строим таблицу разбиений Π_1 (таблица 3.3), заменяя состояния в таблице переходов (см. таблицу 3.1) соответствующими классами одноэквивалентности.

Таблица 3.3 – Таблица разбиений

$z(t)$	B_1											B_2			
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{12}	a_{11}	a_{13}	a_{14}	a_{15}
z_1	B_1	B_1	B_1	B_1	B_2	B_2	B_2	B_1	B_1	B_1	B_1	B_1	B_1	B_1	B_1
z_2	B_1	B_1	B_1	B_1	B_1	B_1	B_2	B_1	B_1	B_1	B_1	B_1	B_1	B_1	B_1

По таблице 3.3 получаем множество Π_2 классов двухэквивалентных состояний (таблица 3.4):

$$\Pi_2 = \{C_1, C_2, C_3, C_4\}; C_1 = \{a_1, a_2, a_3, a_4, a_8, a_9, a_{10}, a_{10}, a_{12}\}; C_2 = \{a_5, a_6\}; C_3 = \{a_7\}; C_4 = \{a_{11}, a_{13}, a_{14}, a_{15}\}.$$

Таблица 3.4 – Двухэквивалентные состояния

$z(t)$	C_1								C_2		C_3	C_4			
	a_1	a_2	a_3	a_4	a_8	a_9	a_{10}	a_{12}	a_5	a_6	a_7	a_{11}	a_{13}	a_{14}	a_{15}
z_1	C_1	C_2	C_3	C_1	C_1	C_1	C_1	C_1	C_4	C_4	C_4	C_1	C_1	C_1	C_1
z_2	C_1	C_1	C_2	C_1	C_1	C_1	C_1	C_1	C_1	C_1	C_4	C_1	C_1	C_1	C_1

По таблице 3.4 получаем множество разбиений Π_3 классов трехэквивалентных состояний (таблица 3.5):

$$\Pi_3 = \{D_1, D_2, D_3, D_4, D_5, D_6\}; D_1 = \{a_1, a_4, a_8, a_9, a_{10}, a_{12}\}; D_2 = \{a_2\}; D_3 = \{a_3\}; D_4 = \{a_5, a_6\}; D_5 = \{a_7\}; D_6 = \{a_{11}, a_{13}, a_{14}, a_{15}\} \text{ и строим таблицу 3.5.}$$

Таблица 3.5 – Трехэквивалентные состояния

$z(t)$	D_1						D_2	D_3	D_4		D_5	D_6			
	a_1	a_4	a_8	a_9	a_{10}	a_{12}	a_2	a_3	a_5	a_6	a_7	a_{11}	a_{13}	a_{14}	a_{15}
z_1	D_3	D_1	D_1	D_1	D_1	D_1	D_4	D_5	D_6	D_6	D_6	D_1	D_1	D_1	D_1
z_2	D_2	D_1	D_1	D_1	D_1	D_1	D_1	D_4	D_1	D_1	D_6	D_1	D_1	D_1	D_1

По таблице 3.5 получаем множество разбиений Π_4 классов четырехэквивалентных состояний (таблица 3.6):

$$\Pi_4 = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7\}; E_1 = \{a_1\}; E_2 = \{a_4, a_8, a_9, a_{10}, a_{12}\}; E_3 = \{a_2\}; E_4 = \{a_3\}; E_5 = \{a_5, a_6\}; E_6 = \{a_7\}; E_7 = \{a_{11}, a_{13}, a_{14}, a_{15}\} \text{ и строим таблицу 3.6.}$$

Таблица 3.6 – Четырехэквивалентные состояния

$z(t)$	E_1	E_2					E_3	E_4	E_5		E_6	E_7			
	a_1	a_4	a_8	a_9	a_{10}	a_{12}	a_2	a_3	a_5	a_6	a_7	a_{11}	a_{13}	a_{14}	a_{15}
z_1	E_4	E_2	E_1	E_1	E_1	E_1	E_5	E_6	E_7	E_7	E_7	E_1	E_1	E_1	E_1
z_2	E_3	E_2	E_1	E_1	E_1	E_1	E_2	E_5	E_2	E_2	E_7	E_1	E_1	E_1	E_1

По таблице 3.6 получим множество разбиений Π_5 классов пятиэквивалентных состояний (таблица 3.7):

$$\Pi_5 = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8\};$$

$F_1 = \{a_4\}$; $F_2 = \{a_4\}$; $F_3 = \{a_8, a_9, a_{10}, a_{12}\}$; $F_4 = \{a_2\}$; $F_5 = \{a_3\}$; $F_6 = \{a_5, a_6\}$; $F_7 = \{a_7\}$; $F_8 = \{a_{11}, a_{13}, a_{14}, a_{15}\}$ и строим таблицу 3.7.

Таблица 3.7 – Пятиэквивалентные состояния

$z(t)$	F_1	F_2	F_3				F_4	F_5	F_6		F_7	F_8			
	a_1	a_4	a_8	a_9	a_{10}	a_{12}	a_2	a_3	a_5	a_6	a_7	a_{11}	a_{13}	a_{14}	a_{15}
z_1	F_5	F_3	F_1	F_1	F_1	F_1	F_6	F_7	F_8	F_8	F_8	E_1	E_1	F_1	F_1
z_2	F_4	F_3	F_1	F_1	F_1	F_1	F_2	F_6	F_3	F_3	F_8	E_1	E_1	F_1	F_1

На этом процедура разбиений заканчивается, поскольку следующее разбиение не добавляет классов эквивалентности.

Разбиение Π_5 есть разбиение множества состояний исходного автомата Мили на классы эквивалентных состояний. Далее каждому классу эквивалентных состояний приписывается один символ внутреннего состояния. Таким символом может быть, например, символ внутреннего состояния с наименьшим порядковым номером.

Возьмем в качестве множества состояний минимального автомата Мили состояния $\{a_1, a_2, a_3, a_4, a_5, a_7, a_8, a_{11}\}$ по одному из каждого класса $F_1 \dots F_8$. Теперь из первоначальной таблицы переходов (см. таблицу 3.2) выбрасываем лишние состояния $a_6, a_9, a_{10}, a_{12}, a_{13}, a_{14}, a_{15}$, в результате чего получаем минимальный автомат Мили, эквивалентный исходному автомату (таблицы 3.8, 3.9). Таблицы 3.8 и 3.9 могут быть объединены в совмещенную таблицу переходов и выходов (таблица 3.10).

Таблица 3.8 – Таблица переходов

$a(t) \backslash z(t)$	a_1	a_2	a_3	a_4	a_5	a_7	a_8	a_{11}
z_1	a_3	a_5	a_7	a_8	a_{11}	a_{11}	a_1	a_1
z_2	a_2	a_4	a_5	a_8	a_8	a_{11}	a_1	a_1

Таблица 3.9 – Таблица выходов

$a(t) \backslash z(t)$	a_1	a_2	a_3	a_4	a_5	a_7	a_8	a_{11}
z_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_1
z_2	w_1	w_1	w_1	w_1	w_1	w_1	w_1	w_2

Таблица 3.10 – Таблица переходов-выходов

$z(t) \backslash a(t)$	a_1	a_2	a_3	a_4	a_5	a_7	a_8	a_{11}
Z_1	a_3 w_1	a_5 w_1	a_7 w_1	a_8 w_1	a_{11} w_1	a_{11} w_1	a_1 w_1	a_1 w_1
Z_2	a_2 w_1	a_4 w_1	a_5 w_1	a_8 w_1	a_8 w_1	a_{11} w_1	a_1 w_1	a_1 w_2

3.3 Минимизация автоматов Мура

При минимизации автоматов Мура вводится понятие нуль-эквивалентности состояний и разбиений множества состояний на нуль-классы. Нуль-эквивалентными называются любые одинаково отмеченные состояния автомата Мура. Если два нуль-эквивалентных состояния любым входным сигналом переводятся в два нуль-эквивалентных состояния, то они называются одноэквивалентными. Все дальнейшие классы эквивалентности состояний для автомата Мура определяются аналогично, как для автоматов Мили [1; 2]. Рассмотрим граф минимизированного автомата Мили (рисунок 3.4).

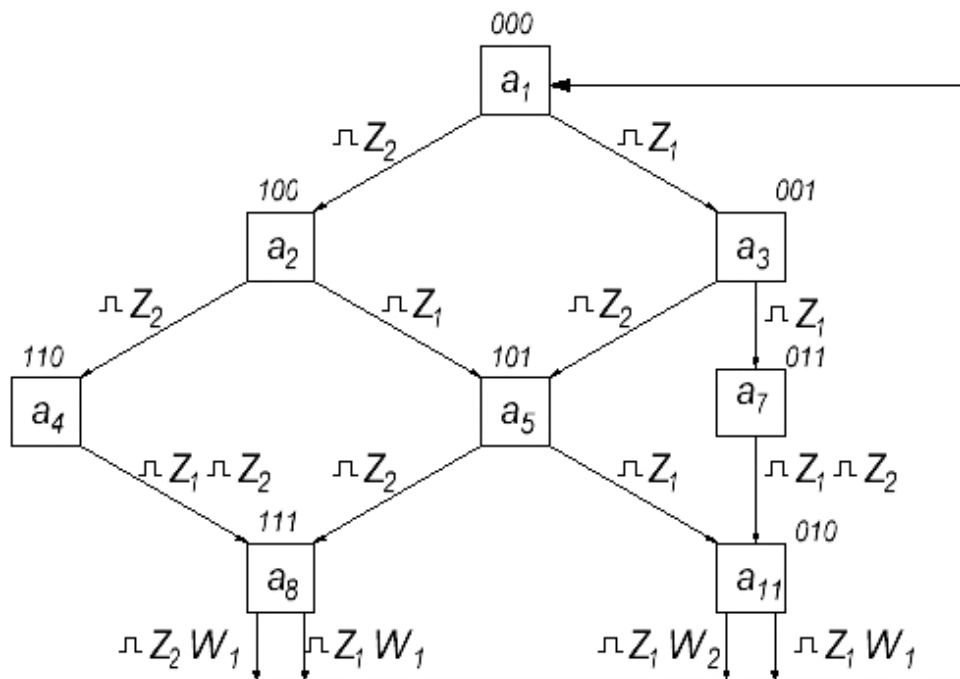


Рисунок 3.4 – Граф минимизированного автомата Мили

Преобразуем этот граф таким образом, чтобы получить автомат Мура. Для этого необходимо состояние a_1 разделить на два состояния: a_{1A} , a_{1B} . Теперь выходной сигнал ω_2 будет формироваться, когда автомат находится в

состоянии a_1 , и будет зависеть только от текущего состояния автомата (рисунок 3.5).

Необходимо отметить, что данное преобразование приводит к увеличению числа состояний и, следовательно, для реализации автомата Мура потребуется дополнительная аппаратура.

Вслед за этапом абстрактного синтеза автомата, заканчивающимся минимизацией числа состояний, следует этап структурного синтеза, целью которого является построение схемы, реализующей автомат из логических элементов заданного типа.

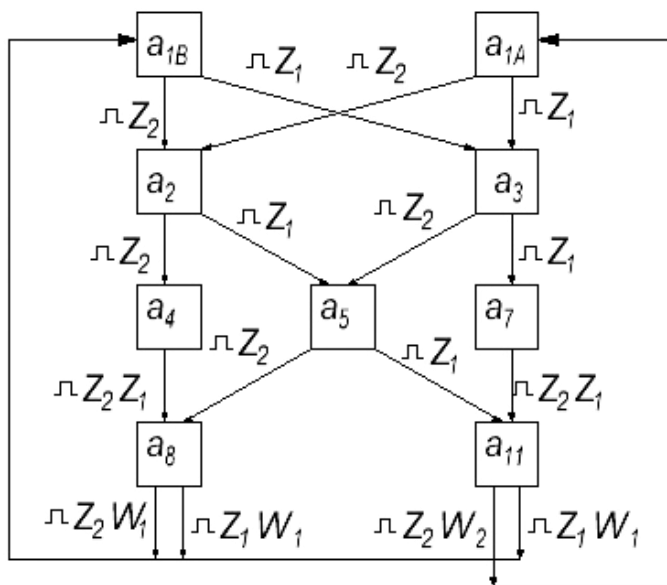


Рисунок 3.5 – Граф минимизированного автомата Мура

Структурный автомат разбивается на две взаимосвязанные части: память и комбинационную схему (рисунок 3.6).

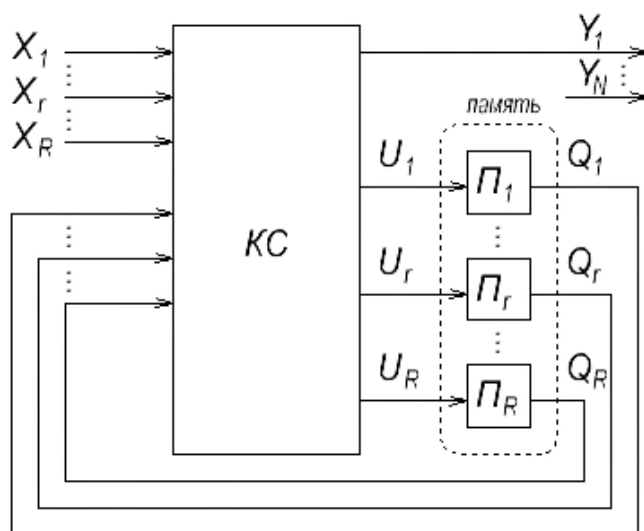


Рисунок 3.6 – Блок-диаграмма структурного автомата

Структурный синтез автомата сводится к построению такой схемы, которая функционирует в соответствии с таблицами переходов, выходов и графов минимизированного абстрактного автомата. В общем случае структурный автомат имеет L входных каналов (входов), на которые поступают входные двоичные переменные $x_1, \dots, x_l, \dots, x_L$, N выходных каналов, на которые выдаются выходные двоичные сигналы $y_1, \dots, y_n, \dots, y_N$, R элементов памяти $P_1, \dots, P_r, \dots, P_R$. Комбинационная часть структурного автомата вырабатывает выходные сигналы $u_1, \dots, u_r, \dots, u_R$, предназначенные для управления переключением элементов памяти и называемые сигналами переключения памяти. Сигналы $Q_1, \dots, Q_r, \dots, Q_R$ определяют состояние элементов памяти и, следовательно, состояние структурного автомата. Они поступают на вход комбинационной схемы, участвуя в выработке выходных сигналов автомата и сигналов переключения элементов памяти, и называются сигналами обратной связи.

Перед выполнением структурного синтеза автомата необходимо выбрать систему элементарных автоматов для построения памяти и систему логических элементов для построения комбинационной схемы. В качестве элементарных автоматов используются триггеры, которые являются элементарными автоматами Мура с двумя состояниями. Каждому состоянию соответствует свой выходной сигнал, что позволяет отождествить состояние элементарного автомата с его выходным сигналом. Это свойство элементарных автоматов означает наличие у них полноты выходов.

Полнота переходов означает, что для любого состояния найдется такой входной сигнал, который переключит триггер в противоположное состояние.

Для перехода от абстрактного автомата к его структурной схеме в общем случае необходимо:

1. Поставить в соответствие каждой букве входного алфавита Z_f , $f = 1, \dots, F$ совокупность значений двоичных сигналов $x_1, \dots, x_l, \dots, x_L$, на входных каналах, т. е. закодировать входные сигналы абстрактного автомата. Количество L физических входов структурного автомата определяется из условия $L \geq \log_2 F$ и выбирается ближайшее целое число.

2. Поставить в соответствие каждой букве выходного алфавита ω_g , $g = 1, \dots, G$ совокупность значений двоичных выходных сигналов структурного автомата $y_1, \dots, y_n, \dots, y_N$ на его выходных каналах, т. е. закодировать выходные сигналы абстрактного автомата. Количество N физических выходов структурного автомата определяется из условия $N \geq \log_2 G$ и выбирается ближайшее целое число.

3. Поставить в соответствие каждому состоянию абстрактного автомата $a_1, \dots, a_m, \dots, a_M$ совокупность состояний элементарных автоматов $Q_1, \dots, Q_r, \dots, Q_R$, т. е. закодировать состояние абстрактного автомата. Количество R элементов памяти структурного автомата определяется из условия $R \geq \log_2 M$ и выбирается ближайшее целое число.

4. Составить систему логических уравнений для функции $y_1, \dots, y_n, \dots, y_N, u_1, \dots, u_r, \dots, u_R$. Эти функции определяют комбинационную схему автоматов.

Структурный синтез основан на каноническом методе синтеза, в результате которого составляются системы логических функций для построения комбинационной схемы, формирующей выходные сигналы автомата и сигналы переключения элементов в памяти.

Для автомата Мили эта система имеет вид

$$\left. \begin{aligned} y_1 &= y_1(Q_1, \dots, Q_R, x_1, \dots, x_L), \\ &\vdots \\ y_N &= y_N(Q_1, \dots, Q_R, x_1, \dots, x_L), \\ u_1 &= u_1(Q_1, \dots, Q_R, x_1, \dots, x_L), \\ &\vdots \\ u_R &= u_R(Q_1, \dots, Q_R, x_1, \dots, x_L). \end{aligned} \right\}$$

Данная система логических функций называется канонической. Рассмотрим пример канонического метода структурного синтеза автомата, заданного таблицами переходов и выходов (см. таблицы 3.8 – 3.10) или графом (см. рисунок 3.4).

Используем в качестве элементарных автоматов J - K -триггеры, таблица переходов которых приведена ниже (таблица 3.11).

Таблица 3.11 – Таблица переходов J - K -триггера

Q^t	Q^{t+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Определяем количество входов структурного автомата $L = \log_2 2; L = 1$.

Кодируем входные сигналы: z_1 соответствует $x = 0$ и $z_2 - x = 1$.

Определяем количество выходов структурного автомата $N \geq \log_2 2; N = 1$.

Кодируем выходные сигналы: ω_1 соответствует $y = 0$ и $\omega_2 - y = 1$.

Определяем количество элементов памяти структурного автомата (J - K -триггеров) $R \geq \log_2 8; R = 3$. Кодируем состояние автомата и результаты заносим в таблицу 3.11 и в граф (рисунок 3.4).

На основании результатов кодирования входных, выходных сигналов и состояний структурного автомата преобразуем совмещенную таблицу переходов и выходов (см. таблицу 3.10) в таблицу определения входов

J - и K -триггеров (таблица 3.12). При определении значений входных сигналов триггеров используется таблица переходов J - K -триггера (см. таблицу 3.11).

Таблица 3.12 – Определение входов J - и K -триггеров

	$a(t)$	Вход	$a(t + 1)$	Входы триггеров						Выход
	$Q_2 Q_1 Q_0$	x	$Q_2^* Q_1^* Q_0^*$	J_2	K_2	J_1	K_1	J_0	K_0	Y
a_1	0 0 0	0	0 0 1	0	x	0	x	1	x	0
a_2	0 0 0	1	1 0 0	1	x	0	x	0	x	0
	1 0 0	0	1 0 1	x	0	0	x	1	x	0
a_3	1 0 0	1	1 1 0	x	0	1	x	0	x	0
	0 0 1	0	0 1 1	0	x	1	x	x	0	0
a_4	0 0 1	1	1 0 1	1	x	0	x	x	0	0
	1 1 0	0	1 1 1	x	0	x	0	1	x	0
a_5	1 1 0	1	1 1 1	x	0	x	0	1	x	0
	1 0 1	0	0 1 0	x	1	1	x	x	1	0
a_7	1 0 1	1	1 1 1	x	0	1	x	x	0	0
	0 1 1	0	0 1 0	0	x	x	0	x	1	0
a_8	0 1 1	1	0 1 0	0	x	x	0	x	1	0
	1 1 1	0	0 0 0	x	1	x	1	x	1	0
a_{11}	1 1 1	1	0 0 0	x	1	x	1	x	1	0
	0 1 0	0	0 0 0	0	x	x	1	0	x	0
	0 1 0	1	0 0 0	0	x	x	1	0	x	1

Значения входных сигналов триггеров, которые даны в таблице 3.12, определяются следующим образом. Если текущее состояние $a(t)$, $Q_2 Q_1 Q_0 = 000$ и $x = 0$, то следующим состоянием $a(t + 1)$ будет $Q_2 Q_1 Q_0 = 001$, т. е. $Q_2^t = 0$ и $Q_2^{t+1} = 0$, $Q_1^t = 0$ и $Q_1^{t+1} = 0$, $Q_0^t = 0$ и $Q_0^{t+1} = 1$. Используя таблицу переходов J - K -триггера (таблица 3.11), получаем $J_2 = 0$ и $K_2 = x$, $J_1 = 0$ и $K_1 = x$, $J_0 = 1$ и $K_0 = x$. Подобным образом получены все остальные значения входных сигналов триггеров. Значком x обозначены безразличные значения сигнала.

Логические уравнения схемы можно составить по таблице 3.12.

Минимизацию комбинационной схемы осуществим с помощью карт Карно.

Карты Карно для входных сигналов триггеров изображены ниже.

		Q_0Y			
		00	01	11	10
Q_2Q_1	00		1	1	
	01	0	0		
	11	x	x	x	x
	10	x	x	x	x

$$J_2 = \overline{Q_1} \cdot x$$

		Q_0Y			
		00	01	11	10
Q_2Q_1	00				1
	01	x	x	x	x
	11	x	x	x	x
	10		1	1	1

$$J_1 = Q_2 \cdot x + Q_0 \cdot \bar{x}$$

		Q_0Y			
		00	01	11	10
Q_2Q_1	00	1		x	x
	01	0	0	x	x
	11	1	1	x	x
	10	1		x	x

$$J_0 = Q_2 \cdot Q_1 + \overline{Q_1} \cdot \bar{x}$$

		Q_0Y			
		00	01	11	10
Q_2Q_1	00	x	x	x	x
	01	x	x	x	x
	11			1	1
	10				1

$$K_2 = Q_0(Q_1 + \bar{x})$$

		Q_0Y			
		00	01	11	10
Q_2Q_1	00	x	x	x	x
	01	1	1		
	11			1	1
	10	x	x	x	x

$$K_1 = \overline{Q_2} \cdot \overline{Q_0} + Q_2 \cdot Q_0 = \overline{Q_2} \oplus \overline{Q_0}$$

		Q_0Y			
		00	01	11	10
Q_2Q_1	00	x	x		
	01	x	x	1	1
	11	x	x	1	1
	10	x	x		1

$$K_0 = Q_1 + Q_2 \cdot \bar{x}$$

Рисунок 3.7 – Карты Карно для входных сигналов триггеров

После упрощения получаются следующие уравнения для входных сигналов триггеров:

$$J_2 = \overline{Q_1} \cdot x; \quad J_1 = Q_2 \cdot x + Q_0 \cdot \bar{x}; \quad J_0 = Q_2 \cdot Q_1 + \overline{Q_1} \cdot \bar{x};$$

$$K_2 = Q_0(Q_1 + \bar{x}); \quad K_1 = \overline{Q_2} \cdot \overline{Q_0} + Q_2 \cdot Q_0 = \overline{Q_2} \oplus \overline{Q_0}; \quad K_0 = Q_1 + Q_2 \cdot \bar{x}.$$

Уравнение, описывающее выходной сигнал, можно получить непосредственно из таблицы 3.12:

$$Y = \overline{Q_2} Q_1 \overline{Q_0} x \uparrow \downarrow.$$

Используя полученные уравнения, можно легко реализовать схему автомата (рисунок 3.8).

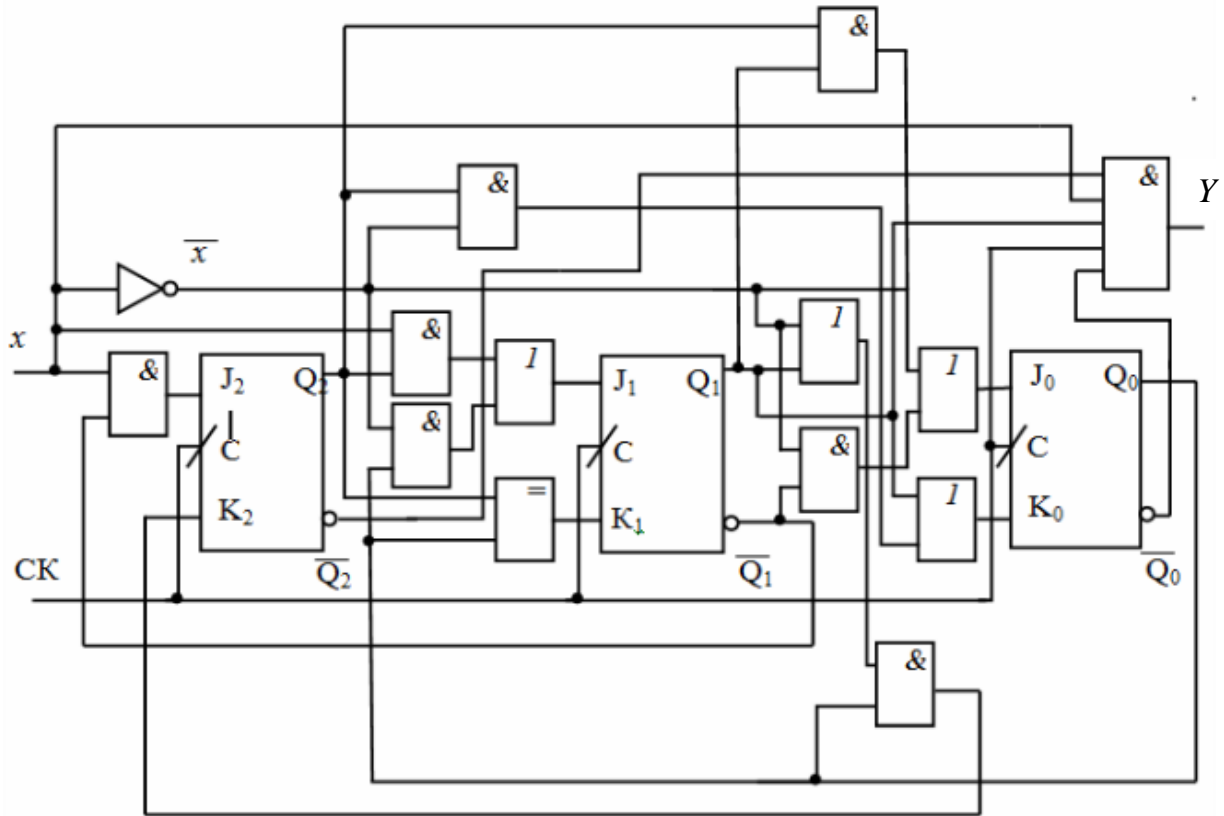


Рисунок 3.8 – Функциональная схема цифрового автомата

Пример синтеза цифрового автомата, реализующего устройство обнаружения последовательности цифр.

Постановка задачи. Двоичные данные в последовательной форме, синхронизируемые тактовыми импульсами, поступают на вход x устройства. На выходе сигнал $y = 1$ появляется только тогда, когда на вход подается последовательность цифр 1101 (причем если две последовательности перекрывают друг друга, то выходной сигнал $y = 1$ формируется дважды).

Граф работы устройства построен на рисунке 3.9.

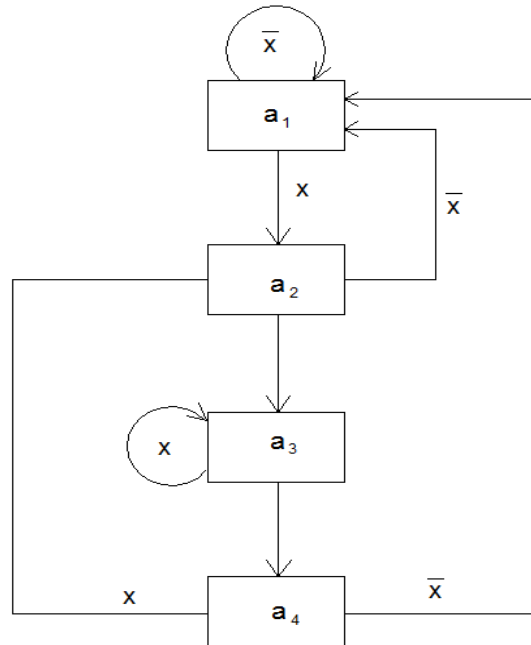


Рисунок 3.9 – Граф работы цифрового автомата

Пусть автомат находится в начальном состоянии a_1 . При подаче на вход X логического нуля, т. е. при \bar{x} , автомат остается в исходном состоянии. Если на вход X поступает логическая 1, т. е. при x автомат переходит в состояние a_2 . Из состояния a_2 автомат переходит в состояние a_3 , если на входе x , или в исходное состояние a_1 , если на входе \bar{x} . В первом случае (a_3) автомат будет продолжать анализировать поступающие на вход данные с учетом первых двух цифр 11, т. к. они соответствуют последовательности 1101. Во втором случае автомат возвращается в исходное состояние, т. к. первые две цифры не соответствуют последовательности 1101, и автомат начинает анализировать новую последовательность.

Из состояния a_3 автомат переходит в состояние a_4 , если на входе \bar{x} , или остается в состоянии a_3 , если на входе x . В первом случае (a_4) автомат будет продолжать анализировать поступающие на вход данные с учетом трех цифр 110, так как они соответствуют последовательности 1101. Во втором случае (a_3) автомат будет продолжать анализировать поступающие на вход данные, но с учетом только последних двух цифр 11, т. к. только они соответствуют последовательности 1101.

Из состояния a_4 автомат переходит в состояние a_1 , если на входе \bar{x} , или в состояние a_2 , если на входе x . В первом случае автомат возвращается в исходное состояние, т. к. поступившая на вход последовательность не соответствует ни целиком, ни частично последовательности 1101. Во втором случае автомат должен выдать выходной сигнал $Y = 1$ и продолжить анализировать поступающие данные с учетом последней цифры, т. к. возможны перекрывающиеся последовательности 1101101.

Минимизация абстрактного автомата осуществляется с помощью таблиц переходов и выходов (таблицы 3.13, 3.14).

Таблица 3.13 – Таблица переходов

	a_1	a_2	a_3	a_4
\bar{x}	a_1	a_1	a_4	a_1
x	a_2	a_3	a_3	a_2

Таблица 3.14 – Таблица выходов

a	a_1	a_2	a_3	a_4
\bar{x}	\bar{y}	\bar{y}	\bar{y}	\bar{y}
x	\bar{y}	\bar{y}	\bar{y}	\bar{y}

Анализ таблиц 3.13 и 3.14 показывает, что уменьшить количество состояний невозможно.

Структурный синтез. Определяем количество элементов памяти $R = 2$ и кодируем состояние автомата, как показано на графе. Входной и выходной сигналы закодированы при постановке задачи.

На основании результатов кодирования преобразуем совмещенную таблицу переходов и выходов автомата (таблица 3.15) в таблицу для определения входов J - и K -триггеров (таблица 3.16).

Таблица 3.15 – Таблица переходов-выходов

	a_1	a_2	a_3	a_4
\bar{x}	a_1 \bar{y}	a_1 \bar{y}	a_4 \bar{y}	a_1 \bar{y}
x	a_2 \bar{y}	a_3 \bar{y}	a_3 \bar{y}	a_2 \bar{y}

Таблица 3.16 – Определение входов J - и K -триггеров

$a(t)$	$a(t)$		$x = 0$		$x = 1$		$x = 0$		$x = 1$	
	$x = 0$	$x = 1$	J_1	K_1	J_1	K_1	J_2	K_2	J_2	K_2
	Q_1Q_2	Q_1Q_2								
00	00	01	0	x	0	x	0	x	1	\emptyset
01	00	11	0	x	1	x	x	1	x	0
11	10	11	x	0	x	0	x	1	x	0
10	00	01	x	1	x	1	0	x	1	x

Минимизацию комбинационной схемы осуществим с помощью карт Карно (рисунок 3.10).

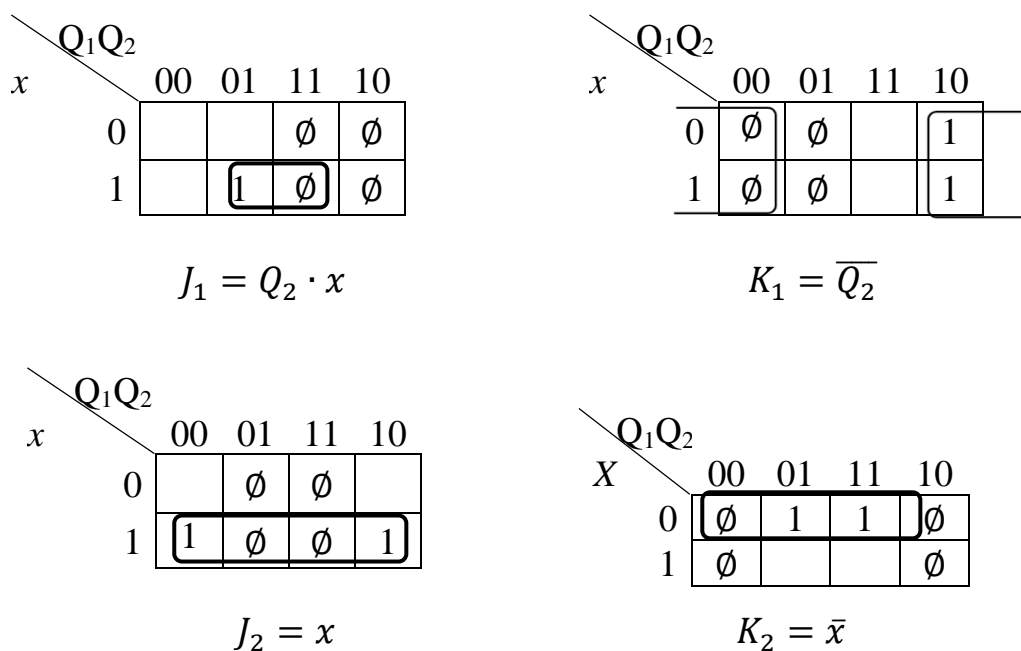


Рисунок 3.10 – Карты Карно

После упрощения получаются следующие уравнения для входных сигналов триггеров:

$$J_1 = Q_2 \cdot x, \quad K_1 = \overline{Q_2}, \quad J_2 = x, \quad K_2 = \bar{x}.$$

Уравнение, описывающее выходной сигнал:

$$Y = Q_1 \overline{Q_2} x \updownarrow.$$

Используя полученные уравнения, можно легко реализовать схему автомата (рисунок 3.11).

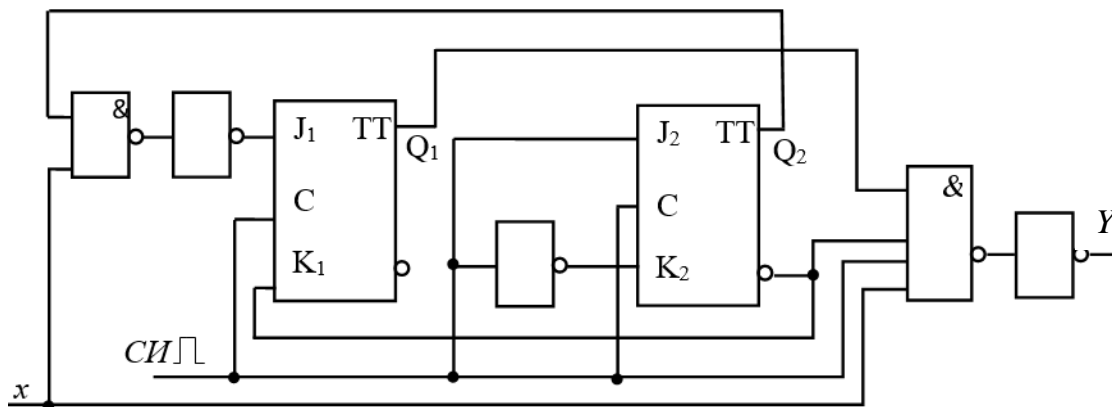


Рисунок 3.11 – Схема автомата

В качестве элементов памяти могут использоваться также *D*- и *T*-триггеры. При этом для получения таблицы выходов необходимо использовать управляющую таблицу соответствующего триггера.

3.4 Задания на курсовое проектирование

Задание 1. Последовательная схема имеет входной канал *Z*, вход синхронизации \mathbb{L} и выходной канал *W*. Поступающие на вход *Z* данные делятся на группы, содержащие по четыре двоичные цифры, которые не перекрывают друг друга, проверяются логической схемой и при обнаружении комбинаций XXXX, XXXX, XXXX и XXXX на выходе *W* появляется 1. Сигнал на выходе должен иметь длительность, равную длительности синхронизирующего импульса \mathbb{L} . При всех других комбинациях четырех двоичных цифр сигнал на выходе должен быть равен 0.

Варианты задания 1 приведены в таблице 3.17.

Таблица 3.17 – Варианты задания 1

1. 0001 0011 0101 1001	2. 0001 0011 0101 1011	3. 0001 0011 0101 1101	4. 0001 0011 0101 1111	5. 0011 0101 0111 1001	6. 0011 0101 0111 1011
7. 0011 0101 0111 1101	8. 0011 0101 0111 1111	9. 0001 0101 0111 1001	10. 0001 0101 0111 1011	11. 0001 0101 0111 1101	12. 0001 0101 0111 1111
13. 0001 0011 0111 1001	14. 0001 0011 0111 1011	15. 0001 0011 0111 1101	16. 0001 0011 0111 1111	17. 1111 1101 1011 0001	18. 1111 1101 1011 0011
19. 1111 1101 1011 0101	20. 1111 1101 1011 0111	21. 1101 1011 1001 0001	22. 1101 1011 1001 0011	23. 1101 1011 1001 0101	24. 1101 1011 1001 0111
25. 1111 1011 1001 0001	26. 1111 1011 1001 0011	27. 1111 1011 1001 0101	28. 1111 1011 1001 0111	29. 0000 0010 0101 1000	30. 0000 0010 0101 1010
31. 0000 0010 0101 1101	32. 0000 0010 0101 1110	33. 0010 0100 0110 1000	34. 0010 0100 0110 1010	35. 0010 0100 0110 1100	36. 0010 0100 0110 1110

37. 0000 0010 0110 1000	38. 0000 0010 0110 1010	39. 0000 0010 0110 1100	40. 0000 0010 0110 1110	41. 1110 1100 1010 0000	42. 1110 1100 1010 0010
43. 1110 1100 1010 0100	44. 1110 1100 1010 0110	45. 1100 1010 1000 0000	46. 1100 1010 1000 0011	47. 1100 1010 1000 0100	48. 1100 1010 1001 0110

Задание 2. Двоичные данные в последовательной форме, синхронизируемые тактовыми импульсами, поступают на вход x устройства. На выходе сигнал $y = 1$ появляется только тогда, когда на вход подается последовательность цифр XXXXXXXX (причем если две последовательности перекрывают друг друга, то выходной сигнал $y = 1$ формируется дважды).

Варианты задания 2 приведены в таблице 3.18.

Таблица 3.18 – Варианты задания 2

1. 1001011	2. 0010111	3. 0101110	4. 1011100
5. 0111001	6. 1110010	7. 1100101	8. 1010011
9. 0100111	10. 1001110	11. 0011101	12. 0111010
13. 1110100	14. 1101001	15. 0110100	16. 1101000
17. 1010001	18. 0100011	19. 1000110	20. 0001101
21. 0011010	22. 0101100	23. 1011000	24. 0110001
25. 1100010	26. 1000101	27. 0001011	28. 0010110

Курсовой проект (работа) должен быть выполнен в соответствии с требованиями, определяемыми Положением об организации и проведении курсового проектирования в БГУИР.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Баранов, С. А. Синтез микропрограммных автоматов / С. А. Баранов. – Л. : Энергия, 1974. – 232 с.

2. Проектирование цифровых вычислительных машин / С. А. Майоров [и др.] ; под ред. С. А. Майорова. – М. : Высш. шк., 1972. – 344 с.

3. Голдсуорт, Б. Проектирование цифровых логических устройств / Б. Голдсуорт. – М. : Машиностроение, 1985. – 288 с.

4. Черепанов, А. К. Микросхемотехника: учебник / А. К. Черепанов. – М. : ИНФРА-М, 2019. – 292 с.

5. Постников, А. И. Прикладная теория цифровых автоматов / А. И. Постников, О. В. Непомнящий, Л. В. Макуха. – Красноярск : федеральн. гос. автоном. образоват. учреждение высш. образования Сибирск. федеральн. ун-т, 2017. – 204 с.

Учебное издание

Будько Анатолий Антонович
Дворникова Татьяна Николаевна

**СИНТЕЗ ЦИФРОВЫХ АВТОМАТОВ.
СХЕМОТЕХНИКА ЦИФРОВЫХ УСТРОЙСТВ.
КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

ПОСОБИЕ

Редактор *Ю. В. Ляховец*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *В. М. Задоя*

Подписано в печать 08.04.2024. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 5,0. Уч.-изд. л. 5,0. Тираж 50 экз. Заказ 45.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск