

Делители частоты

Знакомство со средой САПР БИС «Ковчег 3.04».....	1
Комбинационные схемы	2
Триггерные устройства	3
Параллельные регистры	4
5 Делители частоты.....	5
Синхронные счётчики.....	6
Асинхронные счётчики.....	7
Пересчётные устройства.....	8

Лабораторная работа 5: Делители частоты

5.1. Теоретические сведения	5-2
5.1.1. Организация цепей приёма и выдачи	5-2
5.1.2. Организация цепей сдвига	5-2
5.1.3. Организация цепей сдвига в реверсивных регистрах.....	5-4
5.1.4. Организация цепей сдвига на несколько разрядов за один такт	5-15
5.1.5. Полные графы переходов для сдвиговых регистров с числом разрядов n от 1 до 4.....	5-17
5.1.6. Операции, выполняемые сдвиговым регистром	5-25
5.2. Лабораторное задание	5-30
5.2.1. Пример индивидуального задания	5-30
5.2.2. Порядок выполнения работы	5-31
5.3. Перечень индивидуальных заданий	5-38

Цель работы: изучить виды и состав последовательных регистров; овладеть методами синтеза делителей частоты на базе сдвиговых регистров; приобрести навыки в создании, отладке и экспериментальном исследовании регистровых структур в среде САПР БИС «Ковчег 3.04».

5.1. Теоретические сведения

Последовательный (сдвиговый или сдвигающий) регистр — это регистр, содержимое которого при подаче тактового сигнала (управляющего сигнала «сдвиг») может сдвигаться в сторону старших или младших разрядов. Если регистр выполняется на непрозрачных триггерах, то управление сдвигом осуществляется однофазной синхронизацией. Сдвиговый регистр, выполненный на одноступенчатых защёлках, должен иметь двухфазную систему синхронизации.

В качестве основной элементарной операции сдвига слова применяется сдвиг на один разряд. Если необходимо слово сдвинуть на большее число разрядов, эта операция повторяется соответствующее число раз либо вводится специальная элементарная операция с использованием сдвигателей или других схемных решений. С помощью операции сдвига можно осуществить последовательную передачу слова: поразрядный ввод и поразрядную выдачу слова.

5.1.1. Организация цепей приёма и выдачи

Организация цепей приёма и выдачи данных в сдвиговых регистрах не отличается от процедур, аналогичных для параллельных регистров. Это относится как к последовательной, так и к параллельной загрузке и выдаче данных.

5.1.2. Организация цепей сдвига

Рассмотрим сначала организацию цепей сдвига для случая однофазной синхронизации в предположении, что сдвиг производится влево (в сторону старших разрядов) на один разряд за один такт. Функционирование цепей сдвига представлено в табл. 5.1 в обобщённом виде для двух различных типов триггеров (D и JK). Заполнение этой таблицы осуществляется с использованием соответствующих характеристических таблиц этих триггеров. Из рис. 5.1 следует:

$$D_i^* = Q_{i-1}; \quad (5.1)$$

$$\begin{aligned} J_i^* &= Q_{i-1}; \\ K_i^* &= \bar{Q}_{i-1}. \end{aligned} \quad (5.2)$$

Схемы соответствующих 4-разрядных регистров сдвига приведены на рис. 5.2. Если сдвиговый регистр выполняется на одноступенчатых тактируемых триггерах, то он должен иметь двухфазную систему синхронизации. На рис. 5.3 в качестве примера приведена схема сдвигового регистра на D -триггерах, осуществляющая сдвиг в сторону младших разрядов (вправо).

Следует отметить, что в приведённой схеме при наличии существенных топологических задержек в цепях синхронизации, сравнимых с собственной задержкой триггера, возможны ложные срабатывания регистра. Более подробно об этом указано в книге 1 «Методология проектирования и освоение производства» серии практических пособий «Полузаказные БИС на БМК серий 5503 и 5507» (раздел 5) [5.1].

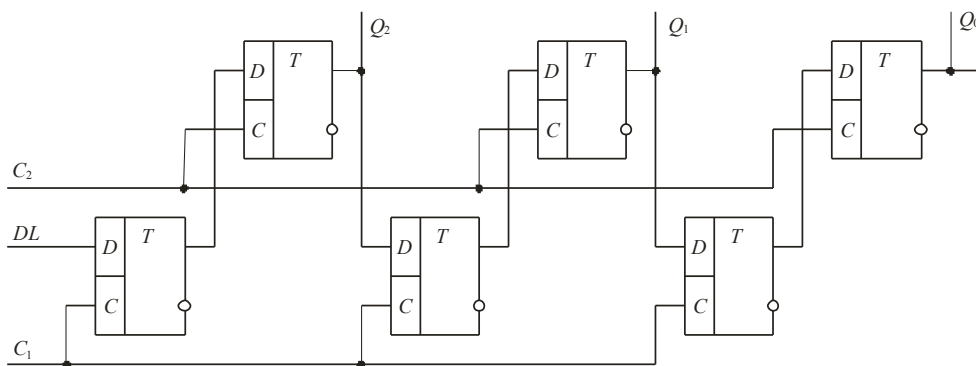


Рис. 5.3. Сдвиговой регистр с двухфазной системой синхронизации

5.1.3. Организация цепей сдвига в реверсивных регистрах

При двунаправленном сдвиге появляются два последовательных входа: вход, принимающий входные данные, вдвигаемые в регистр со стороны младшего разряда DR (*dataright* — данные справа), и вход со стороны старшего разряда DL (*dataleft* — данные слева). Эти же входы используются и при наращивании разрядов регистра: последовательный вход DR старшей секции общего регистра подключается к выходу старшего разряда соседней младшей секции общего регистра, последовательный вход DL младшей секции общего регистра подключается к выходу младшего разряда соседней старшей секции общего регистра.

Кроме последовательных входов сдвиговые регистры могут иметь входы параллельной загрузки за один такт и обязательные управляющие входы, определяющие режим работы сдвигового регистра. Для экономного использования выводов микросхемы часто управляющие сигналы не вводят непосредственно, а формируют их с помощью дешифратора режимов. Другой приём упрощения структуры сдвигового регистра заключается в реализации цепей сдвига только в одном направлении, а для сдвига данных в другом направлении используется режим параллельной загрузки, для чего необходимо осуществить соответствующую коммутацию внешними соединениями между выходами разрядов регистра и входами параллельной загрузки.

Рассмотрим несколько примеров организации сдвиговых регистров.

Пример 5.1. Разработать реверсивный сдвиговой регистр на универсальных D -триггерах. Так как регистр выполняет две микрооперации, то код микрооперации (управляющий сигнал a) будет одноразрядным. Примем, что при $a = 0$ обеспечивается сдвиг вправо (к младшим разрядам), а при $a = 1$ — влево (к старшим разрядам). Функционирование такого регистра отражено в табл. 5.2, где Q_i — про-

извольный разряд сдвигового регистра, Q_{i+1} и Q_{i-1} — соответственно ближайший старший и младший разряды, D_i^* — информационный вход i -го разряда.

Из рис. 5.4 следует:

$$D_i^* = aQ_{i-1} + \bar{a}Q_{i+1}. \quad (5.3)$$

Фрагмент реверсивного сдвигового регистра на D -триггерах приведен на рис. 5.5.

Таблица 5.2. Реализация реверсивного сдвигового регистра (к примеру 5.1)

Номер набора	a	Q_{i+1}^t	Q_i^t	Q_{i-1}^t	Q_i^{t+1} D^*
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

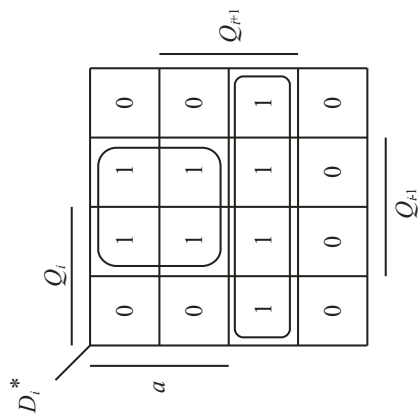


Рис. 5.4. Карта Карно для D_i^*

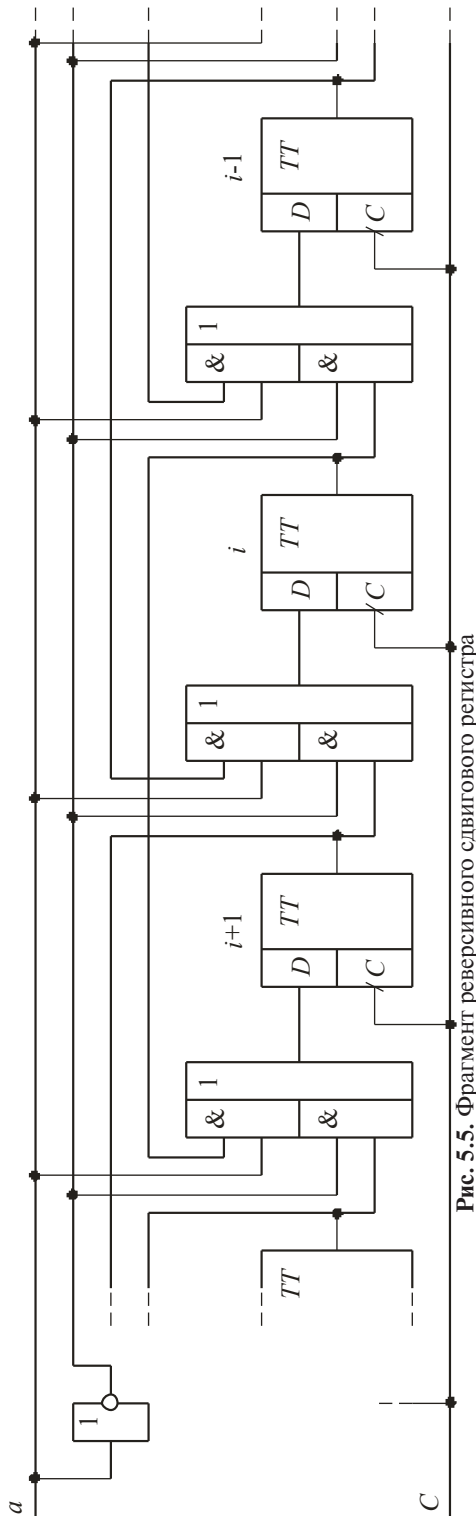


Рис. 5.5. Фрагмент реверсивного сдвигового регистра

Пример 5.2. Разработать реверсивный сдвиговый регистр на универсальных JK-триггерах. Примем, что при $a = 0$ обеспечивается сдвиг влево (к старшим разрядам), а при $a = 1$ — вправо (к младшим разрядам). Функционирование такого регистра отражено в табл. 5.3. Из рис. 5.6 следует:

$$\begin{aligned} J_i^* &= aQ_{i+1} + \bar{a}Q_{i-1}; \\ K_i^* &= a\bar{Q}_{i+1} + \bar{a}\bar{Q}_{i-1}. \end{aligned} \quad (5.4)$$

Таблица 5.3. Реализация реверсивного сдвигового регистра (к примеру 5.2)

Номер набора	a	Q_{i+1}^t	Q_i^t	Q_{i-1}^t	Q_i^{t+1}	J^*	K^*
0	0	0	0	0	0	0	×
1	0	0	0	1	1	1	×
2	0	0	1	0	0	×	1
3	0	0	1	1	1	×	0
4	0	1	0	0	0	0	×
5	0	1	0	1	1	1	×
6	0	1	1	0	0	×	1
7	0	1	1	1	1	×	0
8	1	0	0	0	0	0	×
9	1	0	0	1	0	0	×
10	1	0	1	0	0	×	1
11	1	0	1	1	0	×	1
12	1	1	0	0	1	1	×
13	1	1	0	1	1	1	×
14	1	1	1	0	1	×	0
15	1	1	1	1	1	×	0

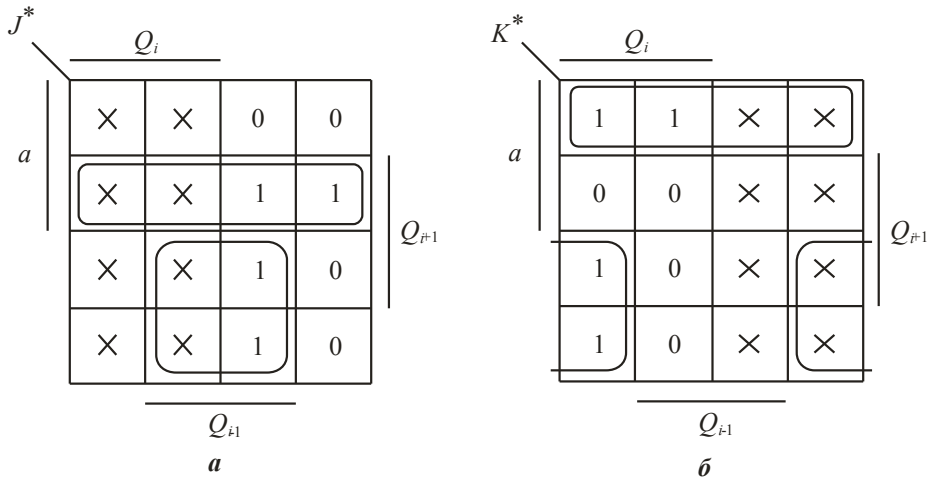


Рис. 5.6. Карты Карно: *a* — для J^* ; *b* — для K^*

Упростим выражение для K_i^* , проделав очевидные преобразования:

$$\begin{aligned}
 K_i^* &= \overline{aQ_{i+1}} + \overline{aQ_{i-1}} = \overline{aQ_{i+1}} \cdot \overline{aQ_{i-1}} = \\
 &= \overline{(a + Q_{i+1})(a + Q_{i-1})} = \overline{aQ_{i+1}} + \overline{aQ_{i-1}} = \overline{J^*}.
 \end{aligned}
 \tag{5.5}$$

Фрагмент реверсивного сдвигового регистра на JK -триггерах приведен на рис. 5.7.

Пример 5.3. Разработать на JK -триггерах регистр, работающий в двух режимах: 1) при управляющем сигнале $V = 0$ должна осуществляться операция сдвига в сторону старших разрядов, 2) при $V = 1$ должна осуществляться параллельная загрузка регистра внешними данными.

Функционирование такого регистра отражено в табл. 5.4, где D_i — i -й разряд параллельного входного кода.

Из рис. 5.8 следует:

$$\begin{aligned}
 J_i^* &= VD_i + \overline{V}Q_{i-1}; \\
 K_i^* &= V\overline{D}_i + \overline{V}Q_{i-1}.
 \end{aligned}
 \tag{5.6}$$

Учитывая преобразования, проделанные в (5.5), можно записать

$$K_i^* = \overline{J_i^*}.
 \tag{5.7}$$

По уравнениям (5.6) и (5.7) реализована схема универсального регистра типа К155ИР1 (зарубежный аналог SN7495). На рис. 5.9 приведено функциональное обозначение этой ИС с указанием номеров выводов микросхемы, в табл. 5.5 приведены режимы функционирования, на рис. 5.10 — структурная схема регистра, а на рис. 5.11 — реализация регистра К155ИР1 в базе библиотеки 5503, выполненная средствами САПР БИС «Ковчег 3.04». Дополнительной особенностью ИС типа 155ИР1 является реализация независимых входов синхронизации для

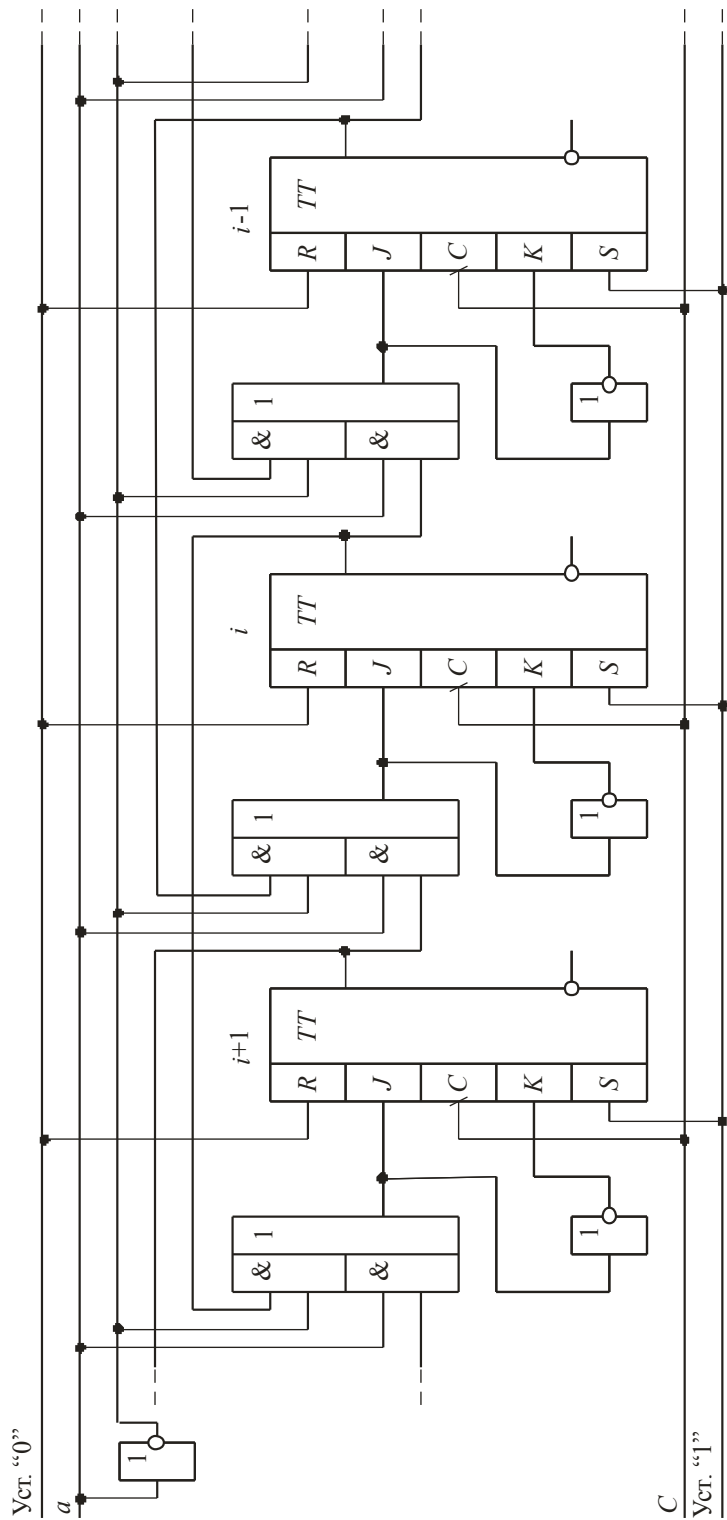


Рис. 5.7. Фрагмент реверсивного сдвигового регистра

режима сдвига C_1 и для режима параллельной загрузки C_2 , коммутация этих входов осуществляется входом установки режима V . Для большинства практических применений $C_1 = C_2$, т.е. используется один синхросигнал.

Таблица 5.4. Функционирование регистра, работающего в двух режимах (к примеру 5.3)

Номер набора	V	D_i	Q_{i1}	Q'_i	Q_i^{r+1}	J^*	K^*
0	0	0	0	0	0	0	×
1	0	0	0	1	0	×	1
2	0	0	1	0	1	1	×
3	0	0	1	1	1	×	0
4	0	1	0	0	0	0	×
5	0	1	0	1	0	×	1
6	0	1	1	0	1	1	×
7	0	1	1	1	1	×	0
8	1	0	0	0	0	0	×
9	1	0	0	1	0	×	1
10	1	0	1	0	0	0	×
11	1	0	1	1	0	×	1
12	1	1	0	0	1	1	×
13	1	1	0	1	1	×	0
14	1	1	1	0	1	1	×
15	1	1	1	1	1	×	0

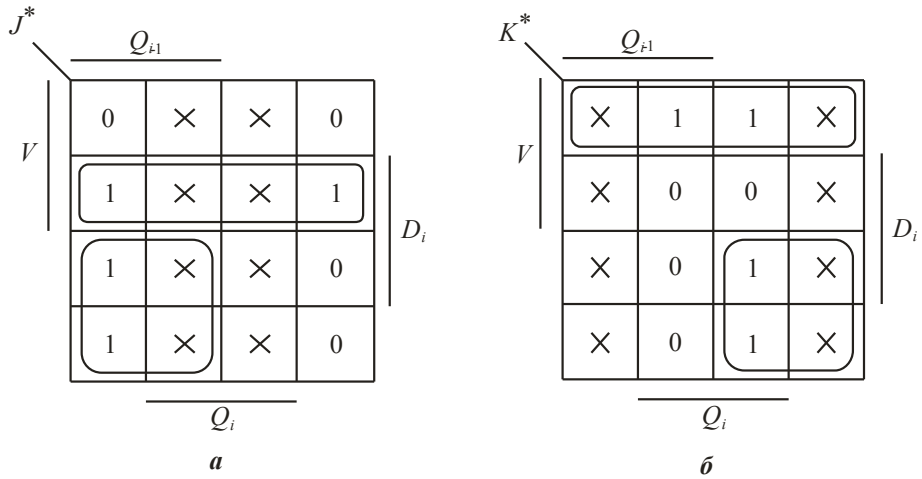


Рис. 5.8. Карты Карно: a — для J^* ; b — для K^*

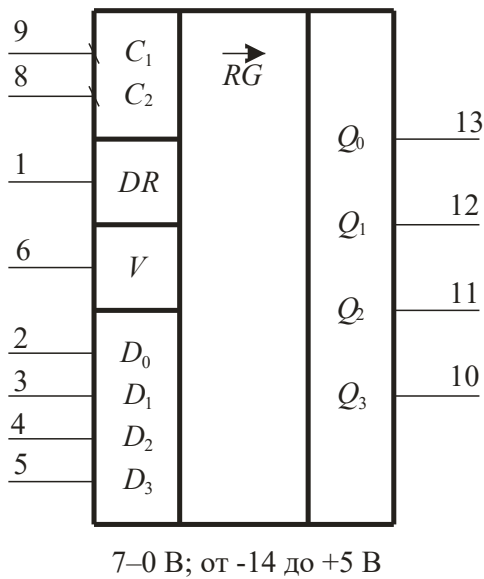


Рис. 5.9. Функциональное обозначение ИС типа К155ИР1, 555ИР1 и т.п.

Таблица 5.5. Режимы функционирования ИС К155ИР1

Режим работы	C_1	C_2	DR	V	Примечание	
Хранение	0	×	×	0	-	
	×	0	×	1		
	0	0	×	×		
Запись параллельного кода	×	⌋	×	1	Код подаётся на входы D_i	
Сдвиг влево	⌋	×	Вход	0	Допускаются указанные ниже соединения	
Сдвиг вправо	×	⌋	×	1	Соединить: 2-12; 3-11; 4-10. Вход - D_3	
Делитель на:	2	⌋	×	y_2	0	$y_2 = \overline{Q_0}$
	3			y_3		$y_3 = \overline{Q_1 Q_0}$
	4			y_4		$y_4 = \overline{Q_1}$
	5			y_5		$y_5 = \overline{Q_2 Q_1}$
	6			y_6		$y_6 = \overline{Q_2}$
	7			y_7		$y_7 = \overline{Q_3 Q_2}$
	8			y_8		$y_8 = \overline{Q_3}$
	...					

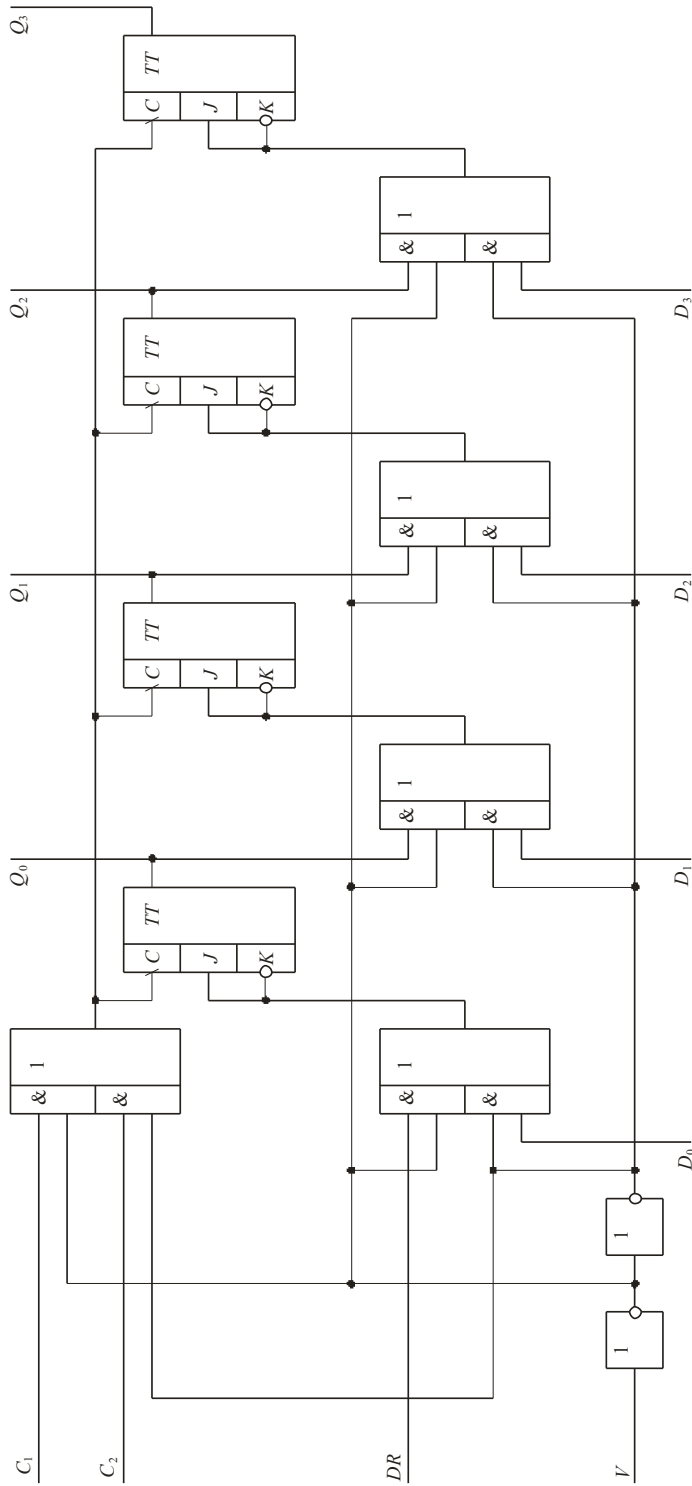


Рис. 5.10. Структурная схема универсального сдвигового регистра типа К155ИР1, 555ИР1 и т.п.

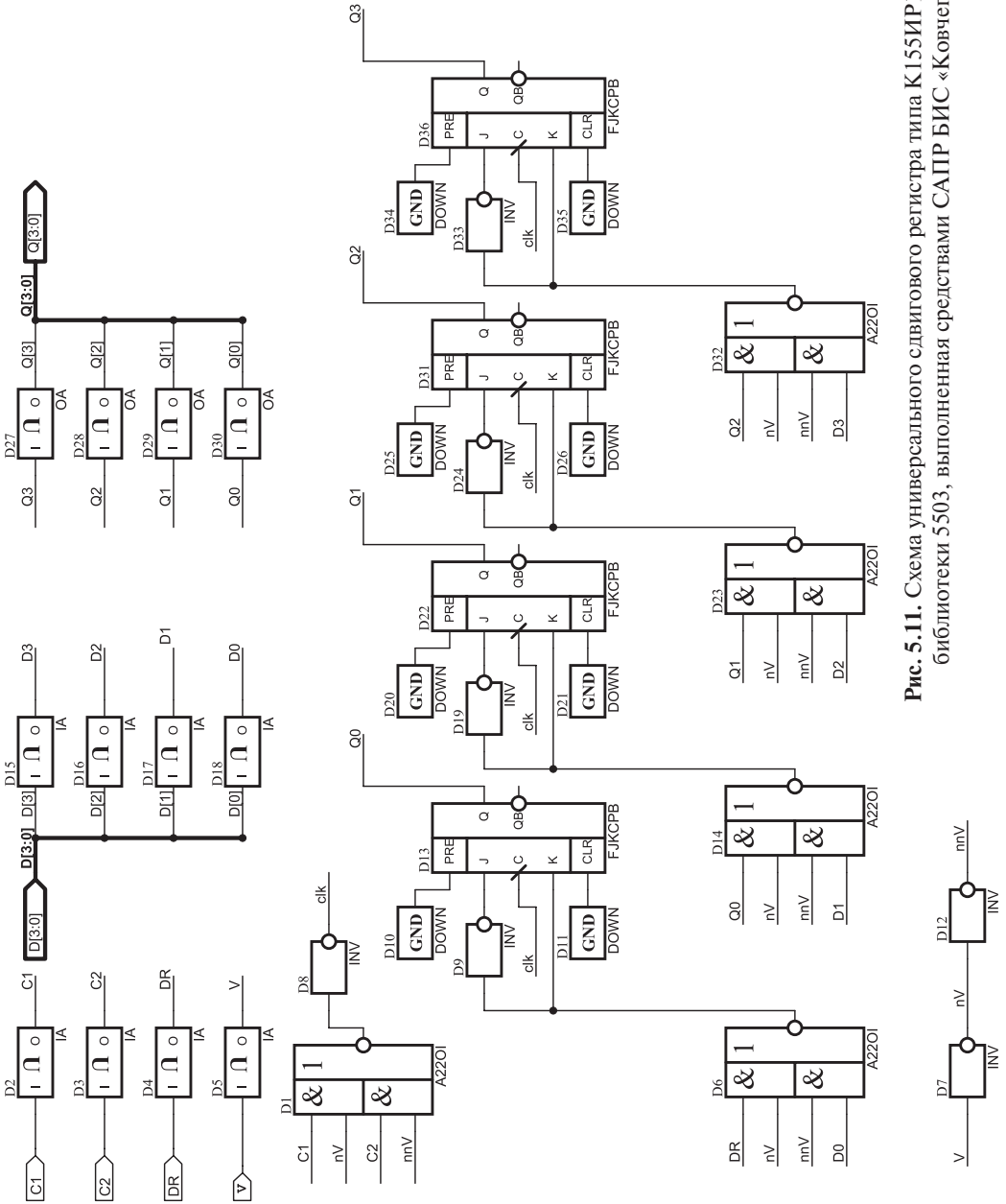


Рис. 5.11. Схема универсального сдвигового регистра типа К155ИР1 в базе библиотеки 5503, выполненная средствами САПР БИС «Ковчег 3.04»

5.1.4. Организация цепей сдвига на несколько разрядов за один такт

Несмотря на большое разнообразие сдвиговых регистров, в ряде случаев их функциональных возможностей оказывается недостаточно. В первую очередь это относится к необходимости обеспечения сдвига за один такт на два или более разрядов. Последнее становится особенно необходимым при реализации арифметических операций в системах счисления с основанием, кратным двум, например в четверичной или шестнадцатеричной.

Аппаратная реализация сдвига данных на два и более разряда за один такт основана на использовании одного из следующих трёх методов: пространственная перестановка разрядов, применение сдвигателей, применение мультиплексоров.

При использовании метода пространственной перестановки разрядов для построения регистров сдвига одновременно на k разрядов необходимо, чтобы общая длина регистра L была кратна произведению mk , где m — разрядность составляющих его регистров. Тогда процедура построения такого регистра будет следующая: обозначим через i номер входа регистра, а через i_m — номер того же входа по модулю m . В результате множество входных шин разбивается на k групп. Входные шины из одной группы подключаются в порядке следования их исходных номеров ко входам составляющего регистра с числом входов L/k . В свою очередь, последний может состоять из отдельных ИС, связанных по цепям сдвига последовательно между собой. ИС, соответствующие различным группам входов, связываются последовательными входами так же, как и в схеме сдвига на один разряд, но данные вводятся последовательным кодом, в котором разряды также пространственно переставлены. Номера выходов в регистре соответствуют номерам входов на нем. На рис. 5.12 показан пример выполнения восьмиразрядного регистра со сдвигом на два разряда за один такт в сторону старших разрядов (влево).

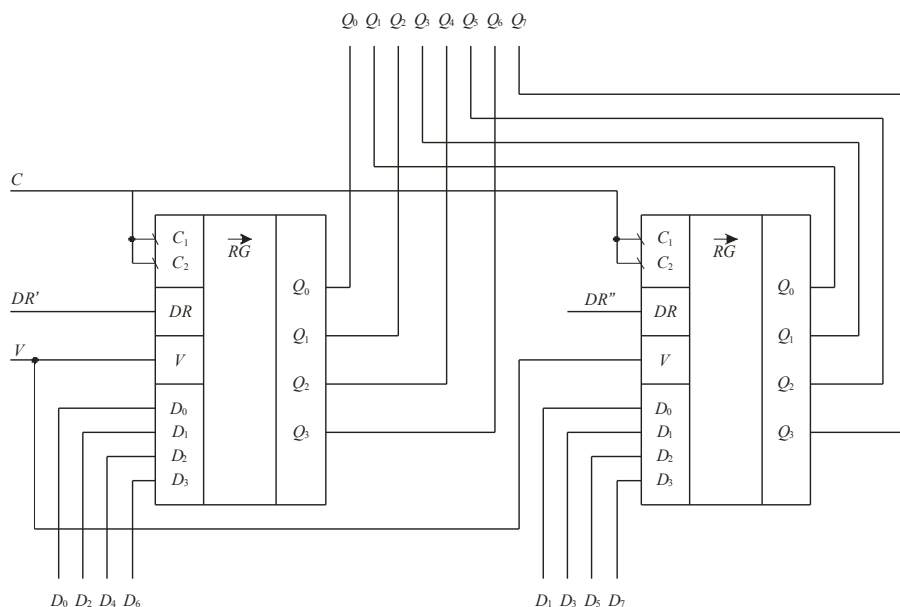


Рис. 5.12. Восьмиразрядный регистр сдвига со сдвигом на два разряда за один такт в сторону старших разрядов (влево)

Сдвигателем называется комбинационный узел ЭВМ, выполняющий микрооперацию сдвига данных на любое число разрядов за один такт. На рис. 5.13 приведена обобщённая схема с использованием сдвигателя, где $RG1$ — регистр, в котором хранится n -разрядное число, m — максимальное число разрядов, на которое сдвигатель может сдвинуть данные за один такт; $RG2$ — регистр, в который записывается сдвинутое число с разрядностью $n + m$; $RG3$ — регистр, в котором хранится код с разрядностью $\log_2(m + 1)$, определяющий число разрядов сдвига за один такт. Дешифратор DC преобразует последний код в $m + 1$ управляющие сигналы, обеспечивающие работу сдвигателя. Направление сдвига определяется либо соответствующим пространственным подключением, либо мультиплексированием входных цепей $RG2$.

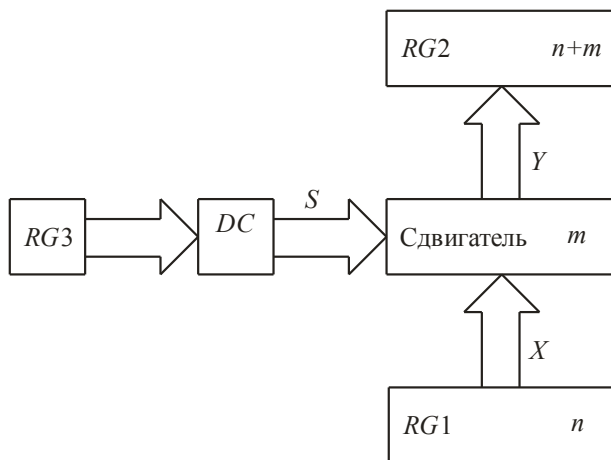


Рис. 5.13. Обобщённая схема сдвига данных на основе сдвигателя

Уравнения для сдвигателя, предназначенного для сдвига от нуля до трёх разрядов 4-разрядного числа, имеют вид

$$\begin{aligned}
 y_0 &= S_0 x_0; \\
 y_1 &= S_0 x_1 + S_1 x_0; \\
 y_2 &= S_0 x_2 + S_1 x_1 + S_2 x_0; \\
 y_3 &= S_0 x_3 + S_1 x_2 + S_2 x_1 + S_3 x_0; \\
 y_4 &= S_1 x_3 + S_2 x_2 + S_3 x_1; \\
 y_5 &= S_2 x_3 + S_3 x_2; \\
 y_6 &= S_3 x_3.
 \end{aligned} \tag{5.8}$$

Как видно из (5.8), сдвигатель представляет собой совокупность элементов И-ИЛИ с различным числом входов по ИЛИ.

Сдвигатели не изготавливаются в виде специализированных ИС и СИС, так как их функции легко реализуются на мультиплексорах, хотя в ряде серий микро-

процессорных наборов есть так называемые «арифметические расширители», которые могут выполнять и функции сдвигателя.

Для реализации уравнений (5.8) потребуется семь мультиплексоров $4 \rightarrow 1$, уравнение для выходного сигнала каждого из которых имеет вид

$$y = \bar{a}_1 \bar{a}_0 D_0 + \bar{a} a_0 D_1 + a_1 \bar{a}_0 D_2 + a_1 a_0 D_3. \quad (5.9)$$

Закодировав число разрядов сдвига за один такт (как показано в табл. 5.6), запишем систему уравнений:

$$\left. \begin{aligned} y_0 &= \bar{a}_1 \bar{a}_0 x_0 + \bar{a} a_0 D_1 + a_1 \bar{a}_0 D_2 + a_1 a_0 D_3 \\ y_1 &= \bar{a}_1 \bar{a}_0 x_1 + \bar{a} a_0 x_0 + a_1 \bar{a}_0 D_2 + a_1 a_0 D_3 \\ y_2 &= \bar{a}_1 \bar{a}_0 x_2 + \bar{a} a_0 x_1 + a_1 \bar{a}_0 x_0 + a_1 a_0 D_3 \\ y_3 &= \bar{a}_1 \bar{a}_0 x_3 + \bar{a} a_0 x_2 + a_1 \bar{a}_0 x_1 + a_1 a_0 x_0 \\ y_4 &= \bar{a}_1 \bar{a}_0 D_0 + \bar{a} a_0 x_3 + a_1 \bar{a}_0 x_2 + a_1 a_0 x_1 \\ y_5 &= \bar{a}_1 \bar{a}_0 D_0 + \bar{a} a_0 D_1 + a_1 \bar{a}_0 x_3 + a_1 a_0 x_2 \\ y_6 &= \bar{a}_1 \bar{a}_0 D_0 + \bar{a} a_0 D_1 + a_1 \bar{a}_0 D_2 + a_1 a_0 x_3 \end{aligned} \right\} \quad (5.9)$$

Таблица 5.6. Кодирование числа разрядов сдвига «за один такт»

Код числа m		m
a_1	a_0	
0	0	0
0	1	1
1	0	2
1	1	3

На незадействованные входы D_0-D_3 мультиплексоров D_3-D_5 и D_7-D_9 подаются «0», «1» или сдвигатель замыкается в кольцо в зависимости от алгоритма обработки данных. Соответствующая схема приведена на рис. 5.14.

5.1.5. Полные графы переходов для сдвиговых регистров с числом разрядов n от 1 до 4

Если последовательный вход DR рассматривать как функцию алгебры логики от переменных на выходах разрядов сдвигового регистра, то можно реализовать большое число различных по функциональному назначению схем. Ряд таких схем может быть получен с использованием полного графа переходов для сдвигового регистра. На рис. 5.15 приведены такие графы для n от 1 до 4 для случая сдвига данных влево (в сторону старших разрядов). Для случая сдвига данных вправо (в сторону младших разрядов) постройте полные графы переходов самостоятельно.

Граф переходов служит для представления всех сведений о последовательной схеме, подлежащей разработке. Вершины графа представляют состояния регистра. Для удобства работы с графом рядом с состоянием записан номер состояния, т.е. десятичный эквивалент соответствующего «двоичного числа». Переходы между состояниями изображаются в виде направленных линий. Стрелки указывают направление последовательности событий.

Входной сигнал DR может либо поступать от внешних источников, либо быть функцией состояний сдвигового регистра; он проставляется рядом с линией, по которой под его действием осуществляется переход.

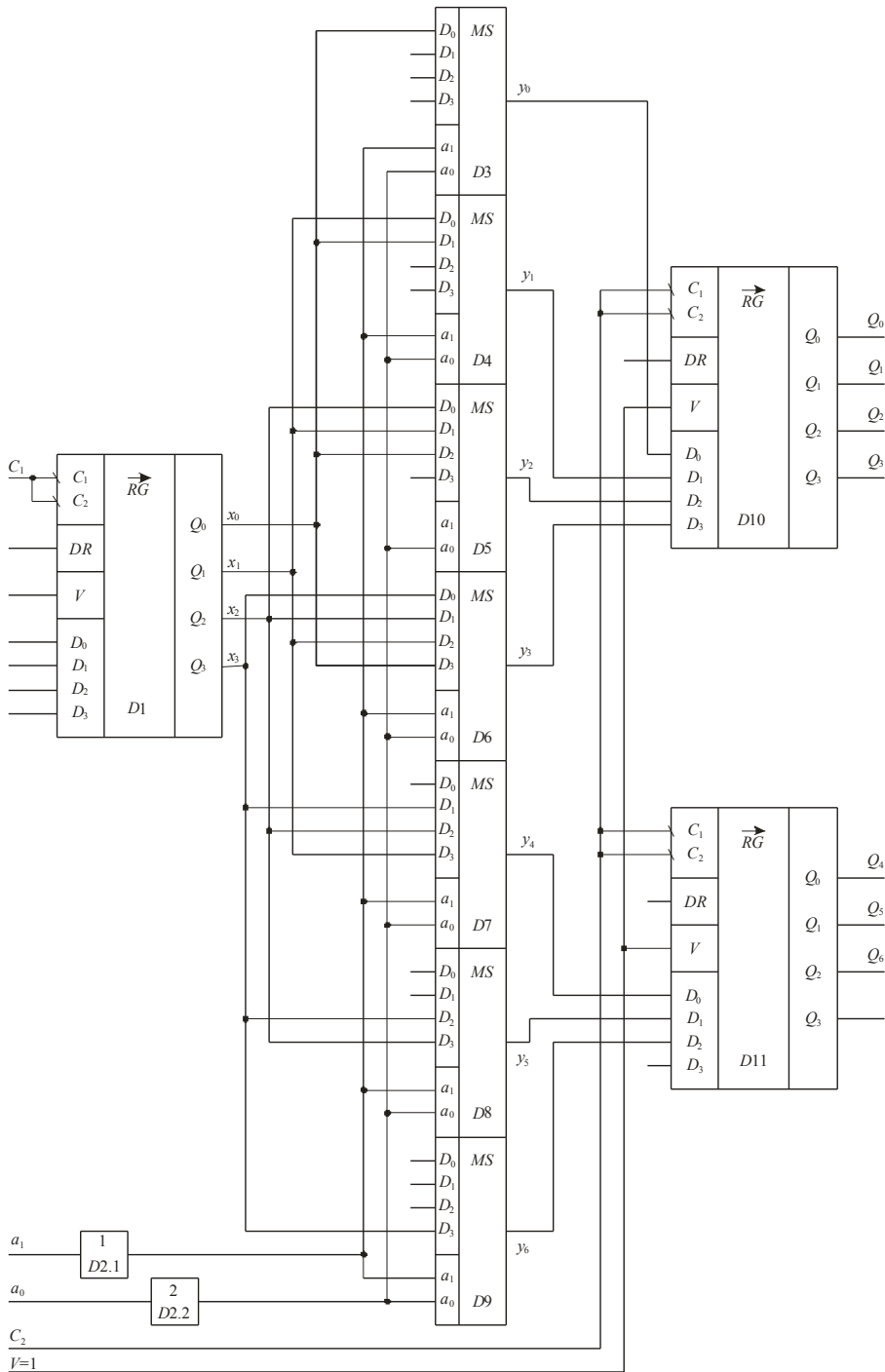


Рис. 5.14. Схема сдвига 4-разрядного числа от нуля до трёх разрядов на мультиплексорах

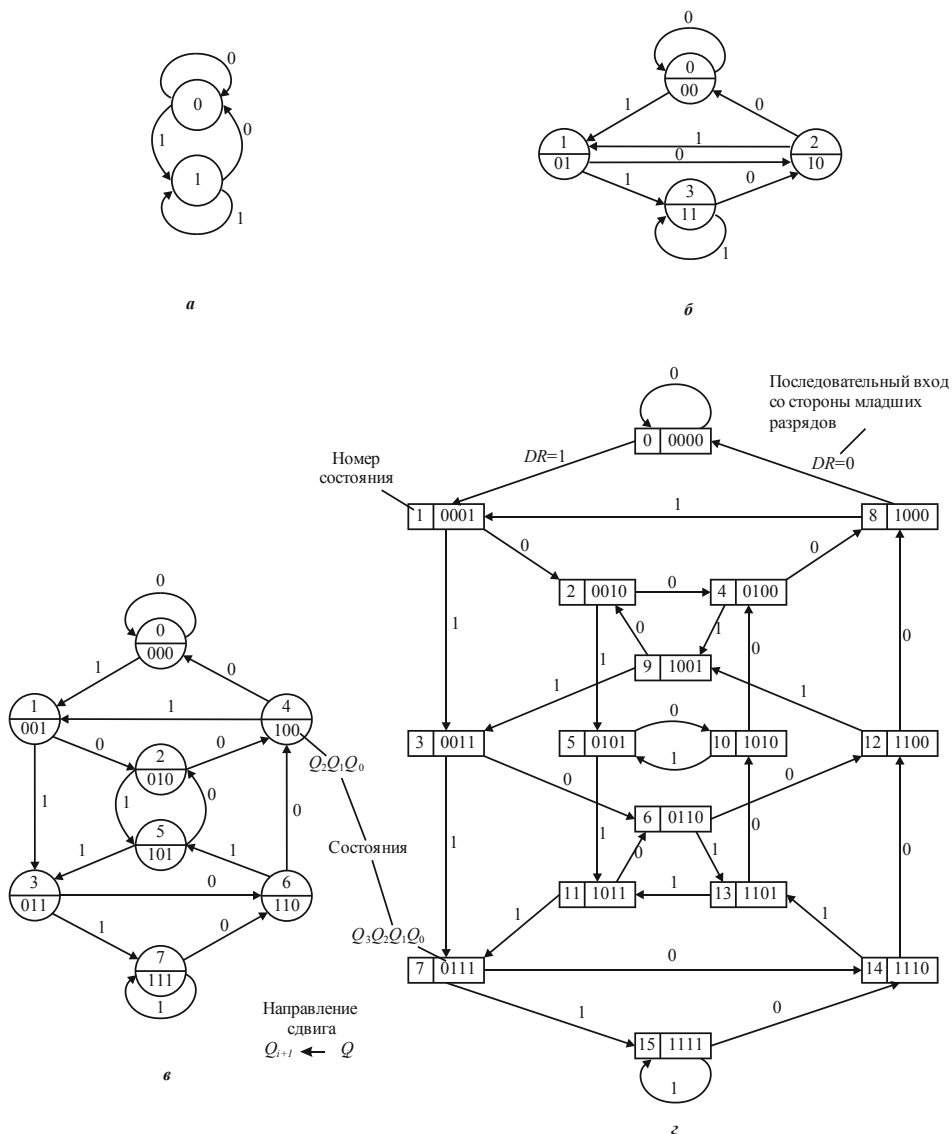


Рис. 5.15. Полные графы переходов для сдвиговых регистров:
 а — $n = 1$; б — $n = 2$; в — $n = 3$; г — $n = 4$

Рассмотрим несколько примеров построения схем с использованием полных графов переходов сдвиговых регистров.

Пример 5.3. Разработать на двухразрядном сдвиговом регистре (рис. 5.16) делитель частоты на три. Разработка включает в себя два этапа: 1) создание автомата, имеющего три рабочих состояния в цикле; 2) формирование выхода делителя частоты. Сигнал с входной частотой подаётся на объединённые тактовые входы всех триггеров, на которых построен сдвиговый регистр.

Из графа на рис. 5.15, б видно, что три состояния в цикле работы сдвигового регистра можно обеспечить двумя способами: 1) переходами 0-1-2-0-... и 2) переходами 1-3-2-1-.... Какое состояние выбрано в любом из этих циклов за начальное, не имеет значения — все они «равноправны». Рассмотрим оба эти способа. Функционирование регистра отображено соответственно в табл. 5.7 и 5.8. Поясним порядок заполнения табл. 5.7. Когда регистр находится в состоянии $Q_1Q_0 = 00$, он готов перейти в состояние $Q_1Q_0 = 01$, для чего на входе DR_1 должно быть сформировано значение «1» в состоянии 00. Нижний индекс 1 у DR говорит о том, что реализуется схема, в которой используется первый из двух упомянутых выше переходов. После перехода в состояние 01 регистр готов перейти в состояние 10, для чего в состоянии 01 должно быть сформировано значение сигнала $DR_1 = 0$ и т.д. Состояние 11 в нормально работающей схеме с переходами 0-1-2-0-... никогда не будет присутствовать, поэтому если оно возникнет при первичном включении или под воздействием помехи, то сигнал DR_1 в этом состоянии должен быть равным «0», в противном случае регистр зависнет в состоянии 11.

Состояние 00 в нормально работающей схеме с переходами 1-3-2-1-... никогда не будет присутствовать, поэтому если оно возникнет при первичном включении или под воздействием помехи, то сигнал DR_2 в этом состоянии должен быть равным «1», в противном случае регистр зависнет в состоянии 00.

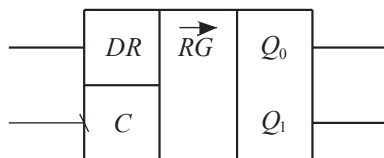


Рис. 5.16. 2-разрядный сдвиговый регистр

Таблица 5.7. Функционирование регистра в режиме делителя частоты на три (цикл: 0-1-2-0-...)

Номер набора	Q_1	Q_0	DR
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	×

Таблица 5.8. Функционирование регистра в режиме делителя частоты на три (цикл: 1-3-2-1-...)

Номер набора	Q_1	Q_0	DR
0	0	0	×
1	0	1	1
2	1	0	1
3	1	1	0

Карты Карно для сигналов DR_1 и DR_2 приведены на рис. 5.17, а и б, из которых следует:

$$DR_1 = \bar{Q}_1 \bar{Q}_0; \tag{5.10}$$

$$DR_2 = \bar{Q}_1 + \bar{Q}_0 = \overline{Q_1 Q_0}. \tag{5.11}$$

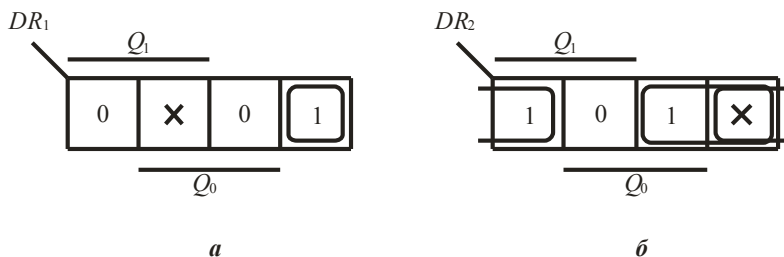


Рис. 5.17. Карты Карно: *a* — для DR_1 ; *б* — для DR_2

Схемы и соответствующие им временные диаграммы приведены на рис. 5.18. Реальные графы работы схем рис. 5.17, *a* и *б* приведены на рис. 5.18. В качестве выходного сигнала разработанных делителей частоты можно рассматривать любой из сигналов: Q_1 , Q_0 , DR . Обратите внимание, что совокупность всех этих трёх сигналов образует распределитель импульсов.

Если сравнить выражение (5.11) с выражением для y_3 в табл. 5.5, они совпадают. Это означает, что если четырёхразрядный регистр К155ИР1 (зарубежный аналог SN7495) используется в режиме делителя на 3, то собственно делитель реализуется на двух разрядах: Q_1 и Q_0 , а разряды Q_2 и Q_3 будут выполнять функцию сдвигового 2-разрядного регистра для сигнала Q_1 . Если эти сдвинутые сигналы не нужны разработчику, то можно сделать общий вывод, что реализация делителя частоты на $\leq 2^{n-1}$ нерациональна на n -разрядном сдвиговом регистре.

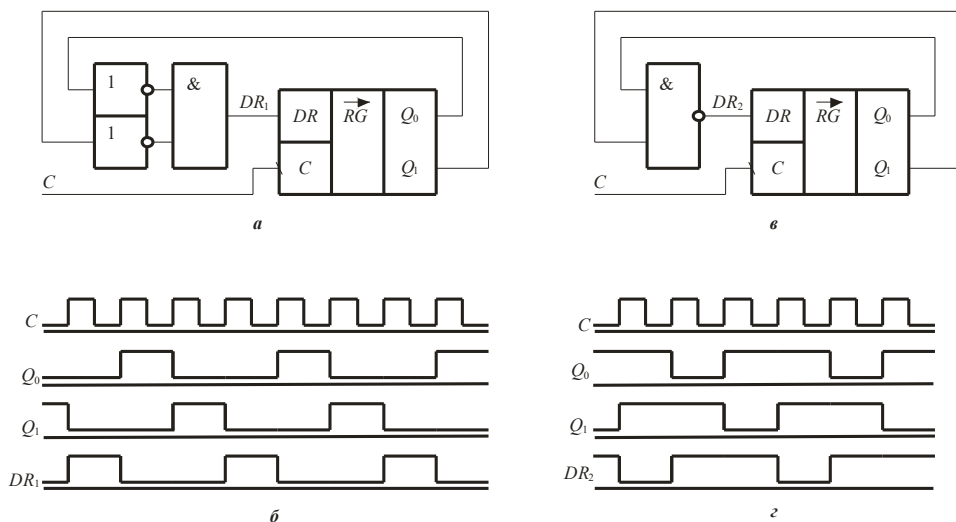


Рис. 5.18. Схемы делителей на три: *a* — 1-й вариант; *б* — временные диаграммы для 1-го способа; *в* — 2-й способ; *г* — временные диаграммы для 2-го способа

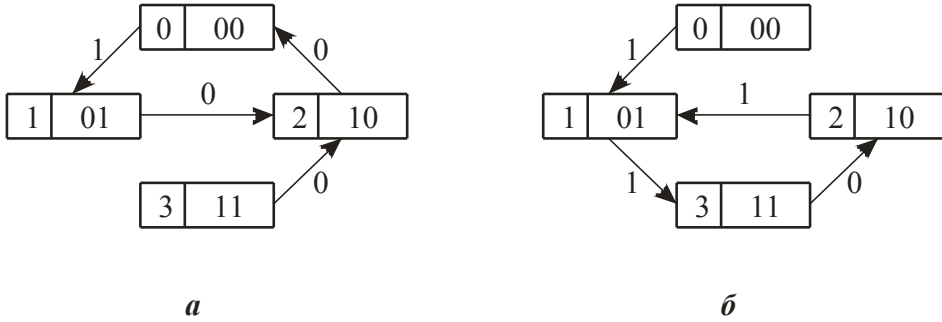


Рис. 5.19. Графы схем делителей на три: *a* — для 1-го способа; *b*— для 2-го способа

Пример 5.4. Разработать на 4-разрядном сдвиговом регистре типа К155ИР1 делитель частоты на 15.

Один из 32 возможных вариантов циклов работы (с числом состояний, равным 15), которые могут быть получены из графа на рис. 5.14, *г*, имеет вид 1-2-4-9-3-6-13-10-5-11-7-15-14-12-8-1.... Функционирование регистра в данном случае отображено в табл. 5.9, которая составляется с использованием графа рис. 5.14, *г*. Из рис. 5.20 следует:

$$DR = Q_3\bar{Q}_2 + \bar{Q}_3Q_2 = Q_3 \oplus Q_2. \tag{5.12}$$

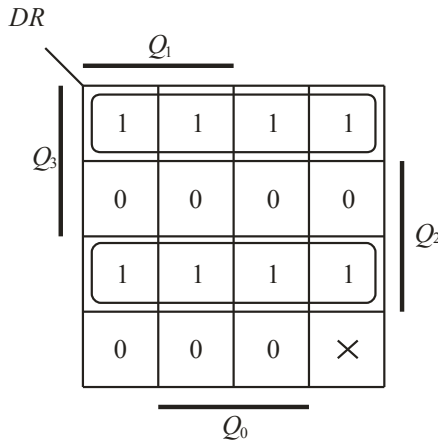


Рис. 5.20. Карта Карно для *DR*

Таблица 5.9. Функционирование регистра в режиме делителя частоты на 15

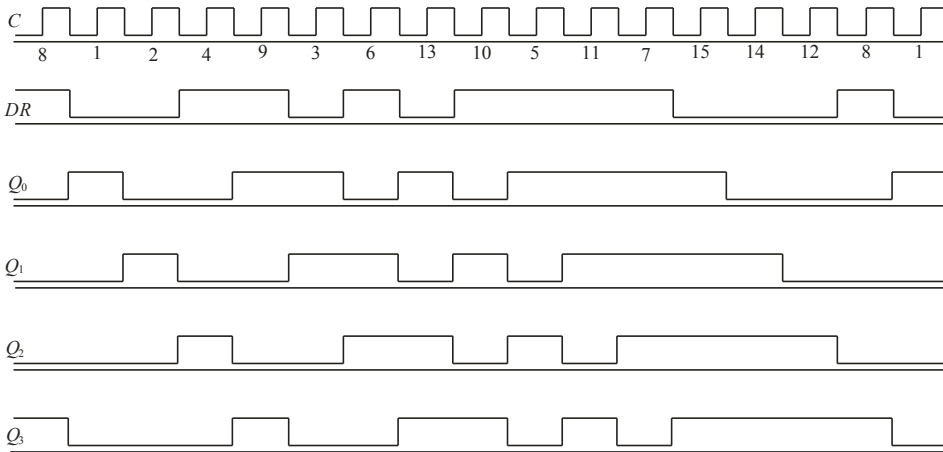
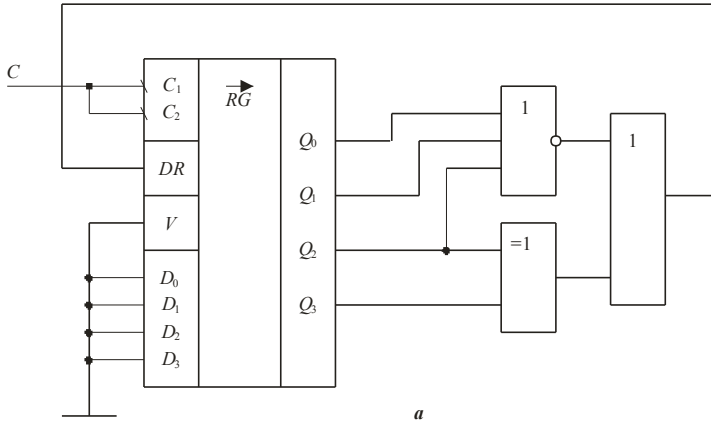
Номер набора	Q_3	Q_2	Q_1	Q_0	DR
0	0	0	0	0	×
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Прежде чем оформлять схему, рассмотрим, как будет вести себя регистр, если он по каким-либо причинам установится в состояние 0, которое отсутствует в рабочем цикле. При написании выражения (5.12) значение DR на наборе (состоянии) 0 доопределено как «0». Ясно, что схема зациклится в состоянии 0, что равносильно катастрофическому отказу. Следовательно, если регистр попадет в это состояние, то на входе DR нужно выработать значение «1», которое переведет схему в состояние 1 и далее она будет работать в нормальном цикле. Из рис. 5.20 следуют три возможных варианта реализации $DR = 1$ в состоянии 0:

$$\left. \begin{aligned} DR_1 &= \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} = \overline{Q_3 + Q_2 + Q_1 + Q_0} \\ DR_2 &= \overline{Q_2} \overline{Q_1} \overline{Q_0} = \overline{Q_2 + Q_1 + Q_0} \\ DR_3 &= \overline{Q_3} \overline{Q_1} \overline{Q_0} = \overline{Q_3 + Q_1 + Q_0} \end{aligned} \right\} \quad (5.13)$$

На рис. 5.20, а показана схема, в которой сигнал DR реализуется следующим образом:

$$DR = Q_3 \oplus Q_2 + \overline{Q_2 + Q_1 + Q_0} \quad (5.14)$$



б

Рис. 5.21. Делитель частоты на 15: а — схема; б — временные диаграммы

Из рис. 5.20, б следует, что ни один из разрядов регистра, ни сигнал DR не могут являться выходом делителя частоты, так как все они выдают четыре импульса

в течение рабочего цикла вместо требуемого одного. Выход делителя частоты может быть сформирован одним из следующих способом:

- дешифрированием любого одного рабочего состояния сдвигового регистра, например, если дешифрируется рабочее состояние 11, то $y_{\text{ВЫХ}} = y_{11} = Q_3 \bar{Q}_2 Q_1 Q_0$ для дешифратора с активной единицей выхода или $y_{\text{ВЫХ}} = y_{11} = \bar{Q}_3 \bar{Q}_2 Q_1 Q_0$ для дешифратора с активным нулём выхода;

- дешифрированием любого одного рабочего состояния сдвигового регистра с возможностью понижения ранга произведений для $y_{\text{ВЫХ}}$, если воспользоваться наличием нерабочих состояний. В рассматриваемом случае имеется только одно нерабочее состояние 0, поэтому можно сформировать следующие четыре варианта сигнала $y_{\text{ВЫХ}}$ (для формирования активной 1 выхода): $y_{\text{ВЫХ}} = y_1 = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1$; $y_{\text{ВЫХ}} = y_2 = \bar{Q}_3 \bar{Q}_2 Q_0$; $y_{\text{ВЫХ}} = y_4 = \bar{Q}_3 Q_1 \bar{Q}_0$; $y_{\text{ВЫХ}} = y_8 = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$. Эти четыре произведения соответствуют двум клеткам карты Карно с наборами 1 и 0; 2 и 0 и т.д.;

- дешифрированием нескольких рабочих состояний, если в рабочем цикле они расположены рядом во времени, в том числе и с учётом нерабочих состояний;

- формированием осведомительного сигнала в зависимости от скважности выходного импульса по отношению к длительности рабочего цикла. Например, если требуется сформировать выходной сигнал, принимающий значение 1 на восьми рядом расположенных во времени состояниях и значение 0 на остальных семи рядом расположенных состояниях, то одним из возможных решений может быть формирование, например, следующего выходного сигнала, представленного в числовой форме СДНФ: $y = V(1,2,4,9,3,6,13,10; \times; 0)$. После минимизации (например, с помощью карт Карно) получаем $y_{\text{ВЫХ}} = \bar{Q}_3 \bar{Q}_2 + \bar{Q}_3 \bar{Q}_0 + \bar{Q}_2 \bar{Q}_0 + Q_3 Q_2 \bar{Q}_1 Q_0$. Если модуль деления частоты — четное число, то легко сформировать выходной сигнал со скважностью 2.

Для устранения рисков сбоя на выходе логического элемента, формирующего сигнал $y_{\text{ВЫХ}}$, рекомендуется подключить конденсатор, емкость которого не должна превышать величину, указанную в технических условиях на используемую серию логических элементов.

Отметим также, что схема, показанная на рис. 5.21, а, более известна как генератор цифровой псевдослучайной последовательности длительностью $2n - 1$ при $n = 4$.

5.1.6. Операции, выполняемые сдвиговым регистром

Сдвиговые регистры как многофункциональные узлы ЭВМ могут выполнять различные операции над данными, наиболее важными среди которых являются:

- преобразование параллельного кода в последовательный;
- преобразование последовательного кода в параллельный;
- хранение чисел в ОЗУ с небольшим объемом данных с последовательной выборкой;
- логические и арифметические сдвиги и др.

При преобразовании параллельного кода в последовательный данные загружаются в регистр параллельным кодом. Затем регистр переводится в режим сдвига и данные выдаются в последовательном коде. Выдача данных может осуществляться старшими или младшими разрядами вперед в зависимости от алго-

ритма их обработки. Типичные примеры использования такого преобразования: построение передатчиков для последовательных интерфейсов; сравнение чисел, представленных последовательными кодами; ввод контролирующих кодов в БИС с ограниченным числом контактных площадок и т.д.

При преобразовании последовательного кода в параллельный данные загружаются в регистр через последовательный вход, после чего они могут быть считаны со всех разрядов регистра одновременно. Типичные примеры использования: построение приёмников для последовательных интерфейсов; вывод контролирующих кодов в БИС с ограниченным числом контактных площадок и т.д.

Хранение чисел в ОЗУ с небольшим объемом данных с последовательной выборкой основано на замыкании сдвигового регистра в кольцо, в котором данные могут записываться, храниться и считываться.

Логические и арифметические сдвиги относятся к командам и выполняются регистрами-аккумуляторами, входящими в состав центральных процессорных элементов всех микропроцессорных комплектов.

Команды выполняются для данных (чисел) без знака и чисел со знаком (рис. 5.22). В большинстве микропроцессоров реализуются восемь команд сдвига, представленных на рис. 5.23. Для всех команд сдвига используется бит переноса C регистра состояний, который является как бы расширением операнда битом 9. Далее под операндом подразумеваются данные без знака либо числа со знаком. Четыре из восьми команд сдвигают операнд, а другие четыре вращают или циклически сдвигают операнд. Команды сдвига и циклического сдвига образуют две группы.

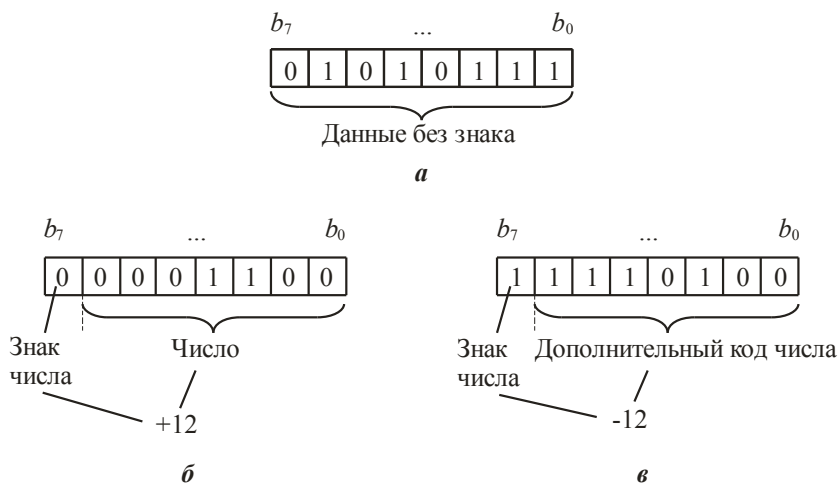


Рис. 5.22. Представление данных в формате 1 байт (8 бит):
a — без знака; *б* — положительных чисел; *в* — отрицательных чисел

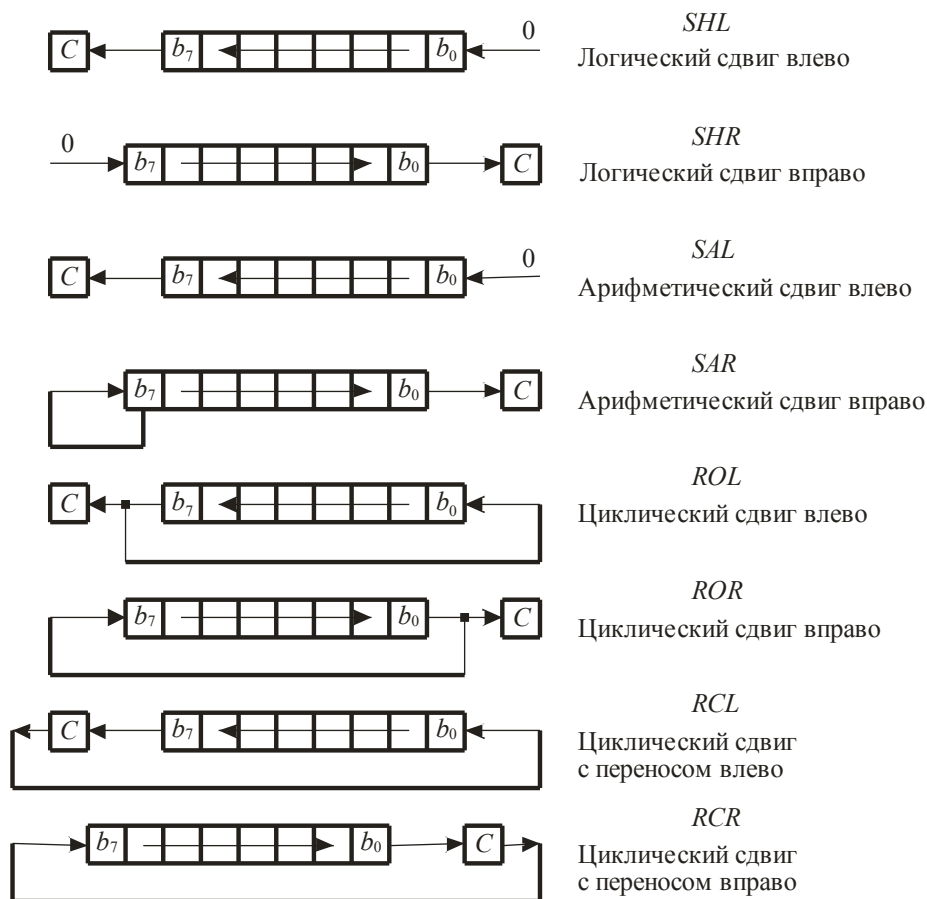


Рис. 5.23. Команды сдвига

Логические команды сдвигают операнд, не считаясь с его знаком, или они используются для действий над числами без знака и над нечисловыми значениями, например над масками. Сдвиг осуществляется по всем позициям регистра.

Арифметические команды сохраняют знаковый (старший) разряд операнда и используются для действия с числами со знаком; отметим, что арифметический сдвиг влево совпадает с логическим сдвигом влево.

Отметим несколько обстоятельств, которые необходимо иметь в виду, работая с командами простого и циклического сдвига. Во-первых, любая из выполняемых при этом операций эквивалентна умножению (сдвиг влево) или делению (сдвиг вправо) числа на 2 при сдвиге на один разряд. Во-вторых, при использовании команд простого сдвига данные, сдвигаемые за пределы регистра, теряются. В-третьих, команды циклического сдвига сохраняют сдвинутые за пределы регистра биты, помещая их обратно в регистр. В-четвертых, команды сдвига взаимодействуют с флагами регистра состояния. Разряд переноса *C* регистра состояния участвует в выполнении команд, и его значение изменяется по мере передачи данных в него или из него.

Рассмотрим подробнее все эти операции (команды).

Логический (SHL — shiftlogicalleft), арифметический (SAL — shiftarithmeticleft) сдвиг влево — эта команда часто используется для реализации операции умножения. Как видно из рис. 5.22, при выполнении этих команд содержимое старшего разряда регистра передаётся в разряд переноса C регистра состояния. В младший разряд b_0 регистра загружается при этом 0. Содержимое каждого разряда регистра перемещается в соседний старший разряд, а содержимое разряда переноса C (независимо от его значения) теряется. Команда SAL не сохраняет знак операнда, но заносит 1 во флаг переполнения V регистра состояния в случае изменения знака операнда.

Логический сдвиг вправо (SHR — shiftlogicalright) — эта команда часто используется для операции деления. Как видно из рис. 5.22, при выполнении этой команды содержимое младшего разряда b_0 регистра передаётся в разряд переноса C регистра состояния. В старший разряд b_7 регистра загружается при этом 0. Содержимое каждого разряда регистра перемещается в соседний младший разряд, а содержимое разряда переноса C теряется.

Арифметический сдвиг вправо (SAR — shiftarithmeticright) — содержимое младшего разряда передаётся в разряд переноса C регистра состояний, содержимое всех остальных разрядов числа сдвигается вправо на одну позицию. Отличительной особенностью команды является то, что старший (знаковый) разряд сохраняет свое значение, это необходимо при выполнении некоторых арифметических операций в дополнительном коде. Содержимое разряда переноса C теряется.

Циклический сдвиг влево (ROL — rotateleft) — содержимое старшего разряда b_7 регистра загружается в разряд переноса C регистра состояния. Все разряды операнда, находящегося в регистре, сдвигаются на одну позицию влево, т.е. содержимое каждого разряда перемещается в позицию соседнего старшего разряда, а содержимое разряда b_7 загружается в разряд b_0 .

Циклический сдвиг вправо (ROR — rotateright) — содержимое младшего разряда b_0 регистра загружается в разряд переноса C регистра состояния. Все разряды операнда, находящегося в регистре, сдвигаются на одну позицию вправо, т.е. содержимое каждого разряда перемещается в позицию соседнего младшего разряда, а содержимое разряда b_0 загружается в разряд b_7 .

Циклический сдвиг с переносом влево (RCL — rotateleftthroughcarry) — этой командой все данные, находящиеся в регистре и разряде переноса, сдвигаются на одну позицию влево, причём содержимое b_7 загружается в разряд переноса C , а последний — в разряд регистра b_0 .

Циклический сдвиг с переносом вправо (RCR — rotaterightthroughcarry) — этой командой все данные, находящиеся в регистре и разряде переноса, сдвигаются на одну позицию вправо, причём содержимое b_0 загружается в разряд переноса C , а последний — в разряд регистра b_7 .

При использовании команд циклического сдвига не происходит потерь данных, поскольку данные перемещаются по замкнутому контуру.

Пример 5.5. Спроектировать операционный блок, выполняющий указанные операции сдвига, на реверсивном восьмиразрядном сдвиговом регистре. В качестве разряда переноса C регистра состояний использовать отдельный триггер. Сигнал m направления сдвига закодировать так: $m = 0$ — сдвиг влево, $m = 1$ —

сдвиг вправо. Триггер переноса и регистр должны иметь единую систему синхронизации.

Так как всего выполняется восемь операций, закодируем их 3-разрядным кодом $a_2a_1a_0$. Функционирование операционного блока представлено в табл. 5.10. Рассматривая $C^{t+1}, b_7^{t+1}, b_0^{t+1}$ и m как функции алгебры логики (ФАЛ) от переменных $a_2, a_1, a_0, C^t, b_7^t$ и b_0^t , из табл. 2.22 получаем выражения в СДНФ:

$$D_c^{*t} = C^{t+1} = \bar{a}_2\bar{a}_1\bar{a}_0b_7 + \bar{a}_2\bar{a}_1a_0b_0 + \bar{a}_2a_1\bar{a}_0b_7 + \bar{a}_2a_1a_0b_0 + a_2\bar{a}_1\bar{a}_0b_7 + a_2\bar{a}_1a_0b_7 + a_2a_1\bar{a}_0b_7 + a_2a_1a_0b_0;$$

$$DL^t = b_7^{t+1} = \bar{a}_2a_1a_0b_7 + a_2\bar{a}_1a_0b_0 + a_2a_1a_0C;$$

$$DR^t = b_0^{t+1} = a_2\bar{a}_1\bar{a}_0b_7 + a_2a_1\bar{a}_0C;$$

$$m = \bar{a}_2\bar{a}_1a_0 + \bar{a}_2a_1a_0b_0 + a_2\bar{a}_1a_0 + a_2a_1a_0.$$

Таблица 5.10. Функционирование операционного блока, реализующего восемь сдвиговых команд

Операция	Номер набора	a_2	a_1	a_0	C^{t+1}	b_7^{t+1}	b_0^{t+1}	m
SHL	0	0	0	0	b_7^t	×	0	0
SHR	1	0	0	1	b_0^t	0	×	1
SAL	2	0	1	0	b_7^t	×	0	0
SAR	3	0	1	1	b_0^t	b_7^t	×	1
ROL	4	1	0	0	b_7^t	×	b_7^t	0
ROR	5	1	0	1	b_0^t	b_0^t	×	1
RCL	6	1	1	0	b_7^t	×	C^t	0
RCR	7	1	1	1	b_0^t	C^t	×	1

После минимизации получаем

$$\left. \begin{aligned} D_c^{*t} = C^{t+1} &= \bar{a}_0b_7 + a_0b_0; \\ DL^t = b_7^{t+1} &= (\bar{a}_2a_1b_7 + a_2\bar{a}_1b_0 + a_2a_1C)a_0; \\ DR^t = b_0^{t+1} &= a_2\bar{a}_0(\bar{a}_1b_7 + a_1C); \\ m &= a_0. \end{aligned} \right\} (5.15)$$

Уравнения (2.38) определяют структуру операционного блока, которая и приведена на рис. 5.24. 8-разрядный сдвиговый регистр показан в обобщённом виде, так как может быть реализован в различных вариантах. На рис. 5.24 обозначено: V — вход выбора режима (например, $V=0$ — параллельная загрузка, $V=1$ — сдвиг); DR (*dataright*) — данные справа (со стороны младших разрядов), на этот вход подаётся сигнал b_0^{t+1} ; DL (*dataleft*) — данные слева (со стороны старших разрядов), на этот вход подаётся сигнал b_7^{t+1} .

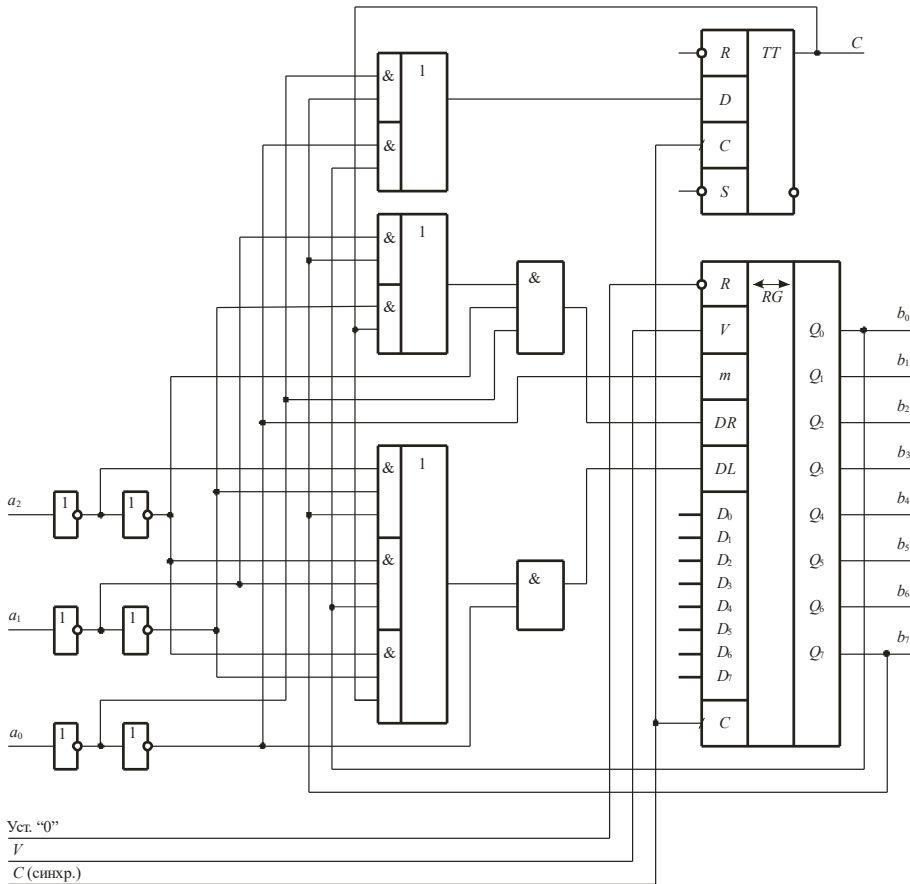


Рис. 5.24. Схема операционного блока

5.2. Лабораторное задание

Выполнить синтез структур, заданных в индивидуальном задании, построить временные диаграммы их работы.

5.2.1. Пример индивидуального задания

Используя сдвиговый регистр на JK -триггерах, разработать самовосстанавливающийся делитель частоты на 13.

5.2.2. Порядок выполнения работы

Для синтеза делителя частоты с использованием сдвигового регистра на *JK*-триггере необходимо выделить на графе переходов (рис. 5.25) замкнутый контур, содержащий столько вершин, на какое число необходимо синтезировать делитель частоты. В нашем случае — контур с 13 вершинами (13 переходов). Один из возможных контуров изображён на рис. 5.26.

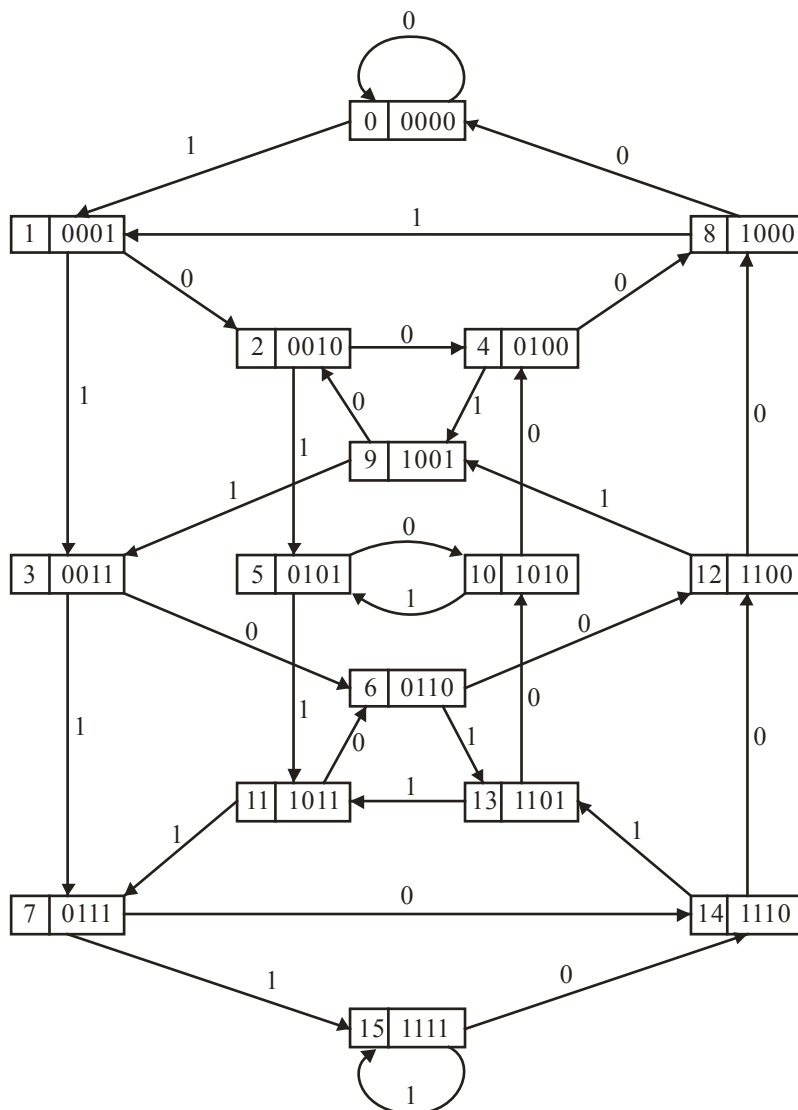


Рис. 5.25. Полный граф переходов 4-разрядного сдвигового регистра

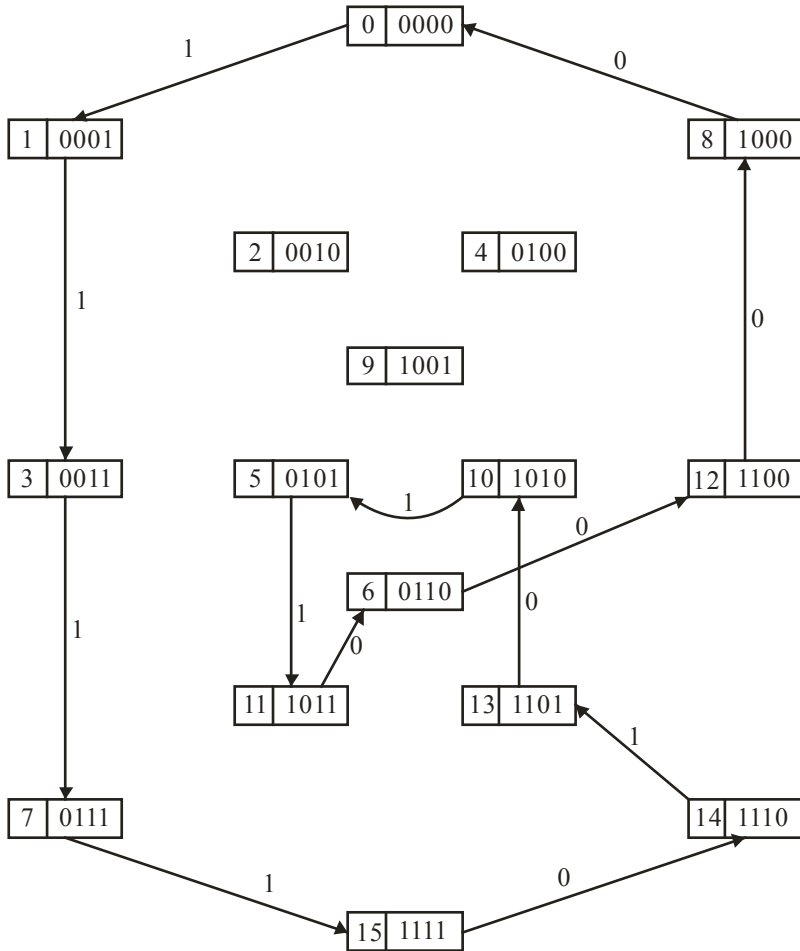


Рис. 5.26. Замкнутый путь из 13 вершин на графе переходов

Если сдвиговый регистр, реализующий выбранный цикл, в исходном состоянии находится в какой-либо вершине графа, то ровно за 13 тактов он вернётся в эту же вершину. Поведение сдвигового регистра полностью определяется состоянием самого младшего триггера, поэтому после выбора маршрута на графе переходов необходимо задать логику работы для входов младшего триггера сдвигового регистра.

Воспользуемся характеристической таблицей для JK-триггера (табл. 5.11).

Таблица 5.11. Характеристическая таблица JK-триггера

$Q^n \rightarrow Q^{n+1}$	J_0	K_0
0 0	0	×
0 1	1	×
1 0	×	1
1 1	×	0
C: 1→0		

Используя её, заполним таблицу переходов (табл. 5.12). Согласно выбранному маршруту, показанному на рис. 5.25, сдвиговый регистр из состояния «0» (0000) должен перейти в состояние «1» (0001), следовательно, младший триггер должен принять значение $Q_0^{t+1} = 1$. Из характеристической таблицы видно, что для перехода $0 \rightarrow 1$ входы $J_0 K_0$ должны принять значения «1» и «×» соответственно, что и записано в табл. 5.12 в строчке с номером набора «0».

Таблица 5.12. Таблица переходов для младшего JK -триггера сдвигового регистра

Номер набора	Q_3	Q_2	Q_1	Q_0	Q_0^{t+1}	J_0	K_0
0	0	0	0	0	1	1	×
1	0	0	0	1	1	×	0
3	0	0	1	1	1	×	0
7	0	1	1	1	1	×	0
15	1	1	1	1	0	×	1
14	1	1	1	0	1	1	×
13	1	1	0	1	0	×	1
10	1	0	1	0	1	1	×
5	0	1	0	1	1	×	0
11	1	0	1	1	0	×	1
6	0	1	1	0	0	0	×
12	1	1	0	0	0	0	×
8	1	0	0	0	0	0	×

Перенесём J_0 и K_0 на карты Карно в целях минимизации функций алгебры логики J_0 и K_0 . Выбираем покрытие, показанные на рис. 5.27.

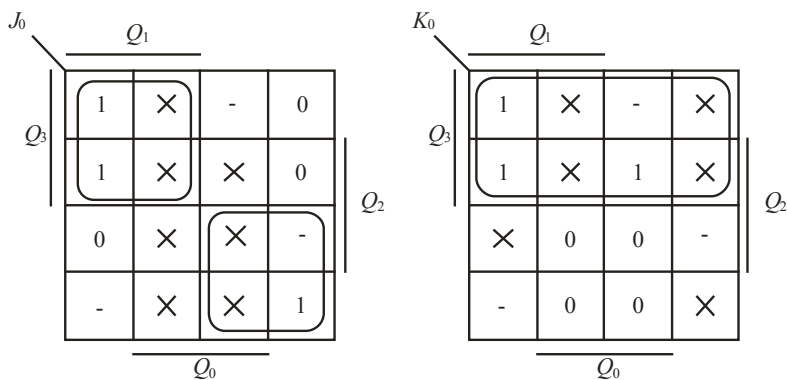


Рис. 5.27. Карты Карно: а — для J_0 ; б — для K_0

В результате минимизации получаем

$$J_0 = Q_3 Q_1 + \overline{Q_3} \overline{Q_1};$$

$$K_0 = Q_3.$$

Теперь, когда выбраны покрытия, определяющие значения входов J_0 и K_0 во всех состояниях, включая нерабочие, следует выполнить проверку, подтверждающую, что сдвиговый регистр является самовосстанавливающимся, т.е. за конечное число тактов он войдёт в рабочий цикл. Так как в нашем рабочем цикле используется 13 различных состояний из 16 возможных, необходимо проверить оставшиеся $16 - 13 = 3$ нерабочих состояния: «2», «4» и «9».

Начнём с состояния «2». Так как эта клетка не попала ни в один из контуров в покрытиях карт Карно на рис. 3, то состояния сигналов J_0 и K_0 в этом состоянии доопределились до $J_0 = \langle 0 \rangle$ и $K_0 = \langle 0 \rangle$. По определению работы JK -триггера (см. раздел 3), если на входы J, K подать комбинацию «0», «0», то он сохранит своё значение выхода. В состоянии «2» (0010) Q_0 принимает значение логического «0», значит, после перехода в новое состояние он также должен принять значение логического «0». Из графа переходов на рис. 1 видно, что это соответствует переходу из вершины «2» в вершину «4», которая также является нерабочей.

Рассуждая аналогично, для «4» состояния видим, что в этом состоянии J_0 и K_0 доопределились до $J_0 = \langle 1 \rangle$ и $K_0 = \langle 0 \rangle$. Это приведёт к тому, что выход триггера установится в логическую «1». Это соответствует переходу из вершины «4» в «9», которая тоже не является рабочей.

В вершине «9» входы J_0 и K_0 доопределились до $J_0 = \langle 0 \rangle$ и $K_0 = \langle 1 \rangle$. Это означает сброс младшего триггера после прихода тактового импульса. Это обеспечит переход из вершины «9» в вершину «2».

В итоге получился паразитный цикл, показанный на рис. 5.28.

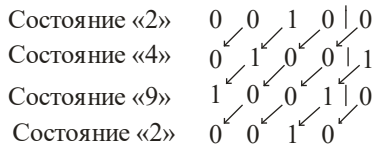


Рис. 5.28. Паразитный цикл, приводящий к неработоспособности делителя частоты

Подобное поведение эквивалентно отказу, поэтому необходимо скорректировать логику работы сдвигового регистра так, чтобы обеспечить вход в рабочий цикл. Для этого скорректируем цикл так, чтобы, например, из состояния «9» произошёл переход в состояние «3», которое является рабочим (рис. 5.29). В связи с этим изменим J_0 и K_0 в картах Карно для состояния «9», внося туда значения «x», «0». В результате изменений выбираем новые покрытия для функций J_0 и K_0 (рис. 5.30).

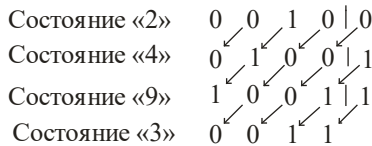


Рис. 5.29. Скорректированная логика работы сдвигового регистра

Избавившись от паразитных циклов, получаем окончательные выражения для информационных входов младшего триггера сдвигового регистра:

$$J_0 = Q_3Q_1 + \overline{Q_3} \cdot \overline{Q_1} ;$$

$$K_0 = Q_3Q_1 + Q_3Q_2 .$$

Следует помнить, что делитель частоты — это двухполюсник, у которого есть вход тактовых импульсов (ТИ) и выход. На выходе делителя частоты должны появляться переходы в 13 раз реже, чем на входе (делитель частоты на 13). Для синтеза функции выхода достаточно сформировать осведомительный сигнал о том, что сдвиговый регистр находится в каком-либо состоянии из рабочего цикла. Выберем для этого, например, состояние «13» (1101₂). Выбранное состояние «13» никак не связано со значением делителя частоты, можно было выбрать состояние «1», «7» или «15», так как они присутствуют в рабочем цикле. Недопустимо выбирать состояния «2», «4» и «9», так как они являются нерабочими. Очевидно, что $F_{\text{вых}} = Q_3Q_2\overline{Q_1}Q_0$ будет принимать значение логической «1» только в «13» состоянии. Таким образом, окончательная схема делителя частоты показана на рис. 5.31.

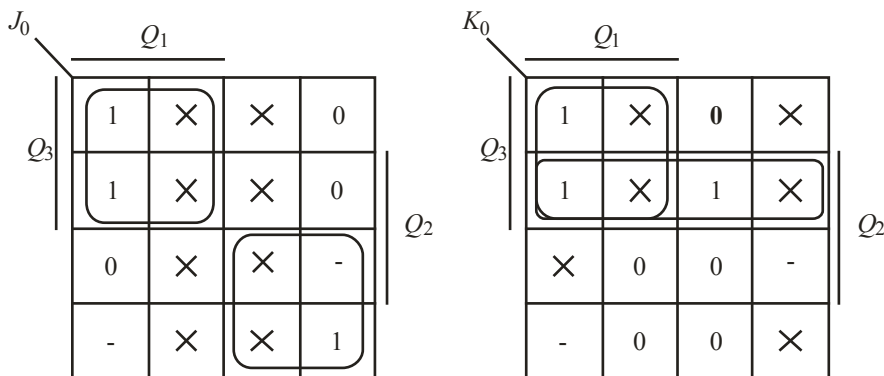


Рис. 5.30. Модифицированные карты Карно: а — для J_0 ; б — для K_0

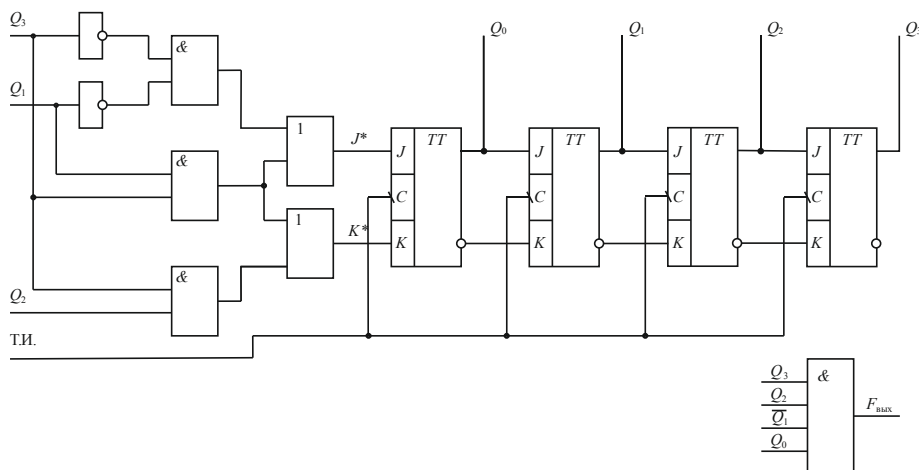


Рис. 5.31. Делитель частоты на D-триггерах

Временная диаграмма работы делителя частоты показана на рис. 5.32. На ней основными сигналами являются ТИ (входной сигнал) и $F_{\text{ВЫХ}}$ (выходной сигнал). Необязательные сигналы Q_3, Q_2, Q_1, Q_0 представлены для пояснения работы сдвигового регистра.

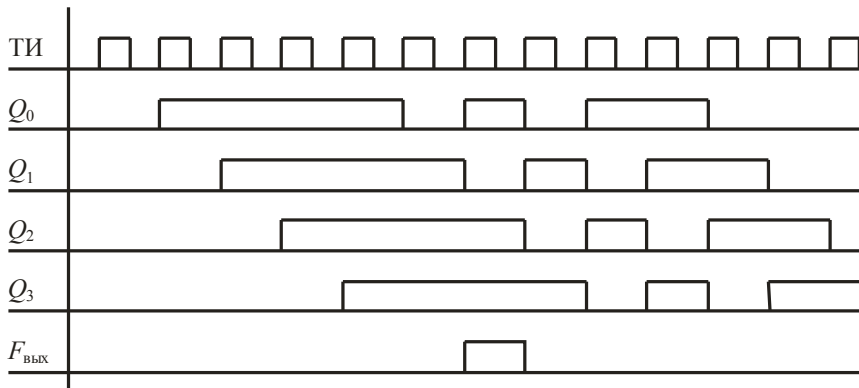


Рис. 5.32. Временная диаграмма работы делителя частоты

5

Используя методические указания к предыдущим работам, создайте новый проект «lab5» с головной схемой DIV13JK. Приступим к практической реализации схемы делителя частоты (рис. 5.31) в среде САПР БИС «Ковчег 3.04». Создадим схему делителя частоты, добавив необходимые входные и выходные сигналы, не забыв про функцию выхода $F_{2KE} = Q_3Q_2Q_1Q_0$ (рис. 5.33).

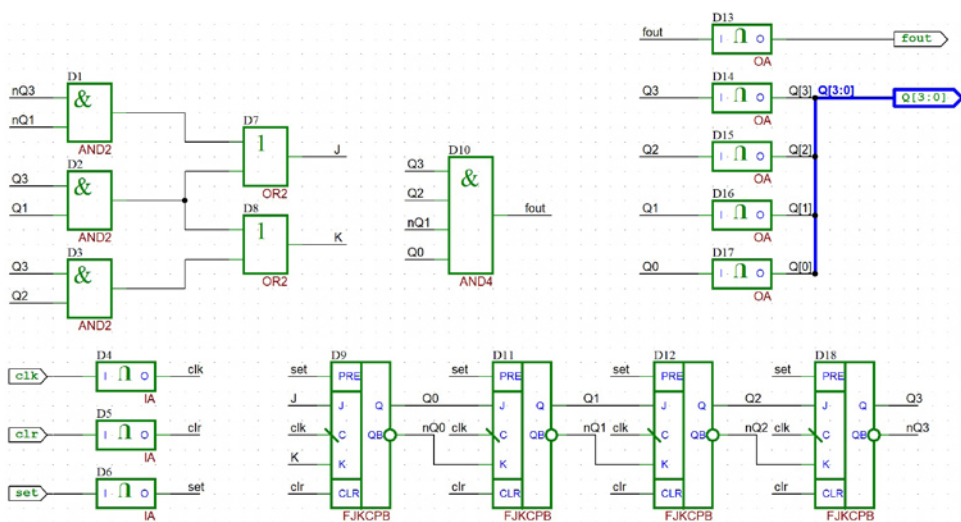


Рис. 5.33. Схема делителя частоты на 13 в среде САПР БИС «Ковчег 3.04»

Выполним трансляцию схемы и убедимся, что она прошла без ошибок, после чего необходимо написать тест для собранной схемы. Необходимо проверить работу делителя частоты на 13, поэтому количество тактовых импульсов должно

быть больше 26, чтобы можно было увидеть периодичность работы делителя. Исходный текст файла с тестами представлен ниже:

1. «Тест для сдвигового регистра»
2. DIVIDER_TEST;
- 3.
4. DIVIDER_TEST:
5. clk = +:30;
6. clr = 1,1,0;
7. set = 0;

Тест выполняется за 30 тактовых импульсов *clk*, первые два такта выделены на сброс *JK*-триггеров. Транслируем тесты и убеждаемся, что процедура прошла без ошибок. Теперь необходимо задать контрольные точки. Для проверки работы делителя частоты нам необходимо отслеживать выбранную последовательность состояний на графе переходов 4-разрядного сдвигового регистра — шина *bus*, а также сам выходной сигнал делителя — *fout*. Исходный текст файла контрольных точек представлен ниже:

1. «Контрольные точки»
2. ALL, bus, fout;
3. ALL: *;
- 4.
5. bus: Q{Q[3], Q[2], Q[1], Q[0]}D;
6. fout: fout;

После успешной трансляции контрольных точек необходимо с помощью команды «Задать параметры контактов», находящейся в подменю «Моделирование» области панелей быстрого доступа, указать длительность импульса *clk*, например 500 нс. После чего можно запустить процесс моделирования и увидеть следующую временную диаграмму (рис. 5.34).

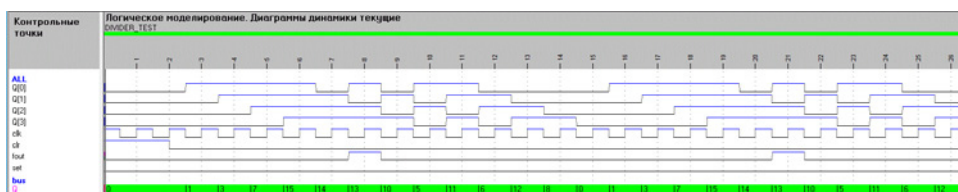


Рис. 5.34. Симуляция работы делителя частоты

Полученная диаграмма (рис. 5.34) в случае правильного проектирования должна совпасть с расчётной (рис. 5.32). При несовпадении временных диаграмм с заданием необходимо найти ошибки синтеза или ошибки в схеме и исправить их, после чего снова выполнить симуляцию работы проекта.

Следует отметить, что выходной сигнал *fout* сформирован на комбинационной ячейке 4И, что с учетом топологических задержек приведет к дребезгу на переднем и заднем фронтах сигнала. Избавиться от этого можно, применив дополнительный триггер *D*-типа с синхронизацией по заднему фронту тактового сигнала *clk*.

5.3. Перечень индивидуальных заданий

1. Используя сдвиговой регистр, разработать самовосстанавливающийся делитель частоты на N (тип триггера и N указаны в индивидуальном задании).

2. Оформить схемы отдельными компонентами в среде САПРС БИС «Ковчег 3.04». Промоделировать работу делителя частоты, сравнить временные диаграммы с расчётными.

Вариант 1

Тип триггера: JK . $N = 10$.

Вариант 2

Тип триггера: JK . $N = 11$.

Вариант 3

Тип триггера: JK . $N = 12$.

Вариант 4

Тип триггера: JK . $N = 13$.

Вариант 5

Тип триггера: JK . $N = 9$.

Вариант 6

Тип триггера: D . $N = 10$.

Вариант 7

Тип триггера: D . $N = 11$.

Вариант 8

Тип триггера: JK . $N = 12$.

Вариант 9

Тип триггера: D . $N = 13$.

Вариант 10

Тип триггера: D . $N = 9$.

Вариант 11.

Тип триггера: D . $N = 10$.

Вариант 12

Тип триггера: D . $N = 11$.

Вариант 13

Тип триггера: D . $N = 12$.

Вариант 14

Тип триггера: D . $N = 13$.

Вариант 15

Тип триггера: D . $N = 9$.

Вариант 16

Тип триггера: JK . $N = 10$.

Вариант 17

Тип триггера: JK . $N = 11$.

Вариант 18

Тип триггера: JK . $N = 12$.

Вариант 19

Тип триггера: JK . $N = 13$.

Вариант 20

Тип триггера: JK . $N = 9$.

Вариант 21

Тип триггера: JK . $N = 10$.

Вариант 22

Тип триггера: JK . $N = 11$.

Вариант 23

Тип триггера: JK . $N = 12$.

Вариант 24

Тип триггера: JK . $N = 13$.

Вариант 25

Тип триггера: JK . $N = 9$.

Вариант 26

Тип триггера: D . $N = 10$.

Вариант 27

Тип триггера: D . $N = 11$.

Вариант 28

Тип триггера: D . $N = 12$.

Вариант 29

Тип триггера: D . $N = 13$.

Вариант 30

Тип триггера: D . $N = 9$.