

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА**

Кафедра автоматики и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

**Лабораторный практикум
по дисциплине «Теория дискретных устройств»**

Часть II

Гомель 2001

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА

Кафедра автоматики и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

Лабораторный практикум
по дисциплине «Теория дискретных устройств»

Часть II

Одобен методическими комиссиями
факультета безотрывного обучения и электротехнического факультета

Гомель 2001

УДК 656.25:681.32 (075.8)
Б 866

Бочков К. А., Березняцкий Ю. Ф.

Б 866 Анализ функциональной структуры и синтез дискретных устройств: Лабораторный практикум по дисциплине «Теория дискретных устройств». Ч. II / Белорус. гос. ун-т трансп. – Гомель: БелГУТ, 2001. – 35 с.

Рассматриваются вопросы анализа функциональной структуры и синтеза дискретных устройств с применением программы «Электронная лаборатория».

Предназначен для студентов третьего курса факультета безотрывного обучения и студентов второго курса электротехнического факультета, обучающихся по специальности Т 11.05 «Автоматика, телемеханика и связь на транспорте».

Р е ц е н з е н т – канд. техн. наук, доцент кафедры «Микропроцессорная техника и информационно-управляющие системы» БелГУТа **С. Н. Харлап**

ВВЕДЕНИЕ

Целью данного цикла лабораторных работ является изучение принципов построения, логической структуры и функциональных особенностей дискретных устройств на базе программы «Электронная лаборатория» (“*Electronics Workbench*”) фирмы “*Interactive Image Technologies*”, а также получение практических навыков анализа и синтеза дискретных устройств различного назначения.

Во второй части лабораторного практикума рассмотрены вопросы синтеза и анализа сумматоров, триггеров и регистров, а также синтеза функций алгебры логики на базе мультиплексоров.

Лабораторная работа № 1

ИЗУЧЕНИЕ ТРИГГЕРОВ

Ц е л ь р а б о т ы. Изучить виды, структуру, принципы построения и способы функционирования различных видов триггеров.

1 Сведения из теории

В современных электронных системах автоматики, телемеханики, связи и вычислительной техники в качестве основного вида элементов памяти применяют *триггеры*.

Триггер представляет собой устройство с двумя устойчивыми состояниями: 0 и 1. Он удобен для обработки двоичной информации. Двум устойчивым состояниям триггера отвечают различные значения выходного сигнала, каждый из которых соответствует логическим 0 и 1. Записанная в триггере двоичная информация (0 или 1) сохраняется до тех пор, пока состояние триггера не изменится. В случае необходимости входным сигналом триггер переводится из одного устойчивого состояния в другое, тем самым изменяется записанная в нем информация. Триггеры выпускают в интегральном исполнении в виде самостоятельных изделий, из которых собирают другие типовые цифровые устройства, или включают в состав любых дискретных устройств с памятью.

Наибольшее распространение получили статические (потенциальные) триггеры, в которых устойчивые состояния различаются уровнем постоянного потенциала. Основу статических триггеров составляет *би-стабильная ячейка* на двух логических элементах ИЛИ-НЕ (рисунок 1.1, а) или И-НЕ (рисунок 1.1, б), охваченных перекрестной положительной обратной связью. Выходы и свободные входы логических элементов образуют соответственно выходы и входы триггера.

Отличие логических элементов от других ключевых схем заключается в восстановлении логических уровней сигнала на выходе, если на вход подан сигнал, воспринимаемый им как логический 0 или 1. В сочетании с положительной обратной связью это свойство определяет наличие в схемах двух равноценных статически устойчивых состояний, в каждом из которых один логических элемент открыт, а другой закрыт. Поэтому выходные сигналы в триггере взаимно инверсны (противоположны в логическом смысле). Один выход называют *прямым*, а другой – *инверсным*, и они обозначаются соответственно Q и \bar{Q} . Состояние триггера отождествляется с сигналом на прямом выходе: говорят, что триггер находится в *единичном* состоянии, когда $Q = 1$, и в *нулевом*, когда $Q = 0$.

При отсутствии внешних сигналов бистабильная ячейка может неограниченно долго находиться в каждом из устойчивых состояний.

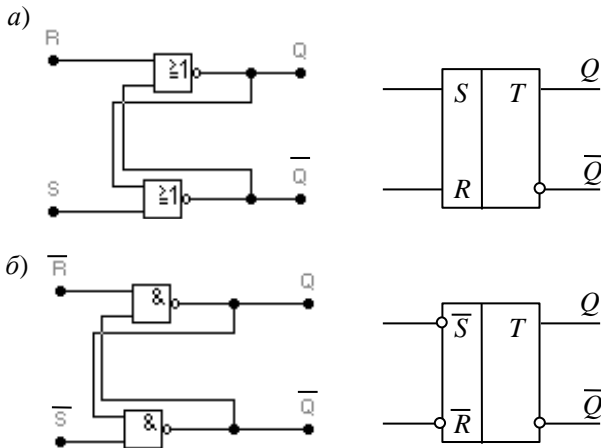


Рисунок 1.1 – Бистабильные ячейки и их функциональные схемы на элементах ИЛИ-НЕ (а) и И-НЕ (б)

Работой триггера управляют внешние сигналы. Цепь управления состоит в установке схемы в одно из двух устойчивых состояний или в изменении одного состояния на другое. Процесс переключения триггера носит

регенеративный (лавинообразный) характер.

Триггером могут управлять как единичные, так и нулевые сигналы, в соответствии с этим входы триггера называются прямыми и инверсными. Для элементов ИЛИ-НЕ переключающим сигналом является единичный сигнал, поэтому входы ячейки на дизъюнктивных элементах прямые, а для элементов И-НЕ переключающим сигналом является нулевой сигнал и входы ячеек на конъюнктивных элементах инверсные. Это находит отражение на графических обозначениях (функциональных схемах).

Термины «триггер» и «бистабильная ячейка» не являются синонимами. Бистабильная ячейка составляет основу триггера, выполняя функцию ячейки памяти. В большинстве модификаций в состав триггера входит устройство управления, преобразующее входную информацию в управляющие сигналы, непосредственно воздействующие на входы бистабильной ячейки. Применение этого устройства позволяет расширить функциональные возможности триггера и получить многочисленные модификации устройств с двумя состояниями, реализующимися алгоритмами функционирования и способами управления. В интегральном исполнении бистабильная ячейка и устройство управления образуют единое целое в схемном и конструктивном отношении, и поэтому понятием «триггер» принято обозначать всю интегральную структуру.

Устройство управления может иметь различные логические структуры, число входов, а также их назначение. В зависимости от функционального назначения входные сигналы подразделяются на три категории.

Сигналы, несущие информацию, которая подлежит записи в триггер, называются *информационными*.

Сигналы, управляющие процессом записи информации, называются *управляющими* и делятся на *подготовительные* (разрешающие) и *исполнительные* (командные).

Входы триггера классифицируются в зависимости от характера воздействующих на них сигналов и также делятся на три категории: *информационные* (логические), *предустановки* (подготовительные) и *тактовые* (синхронизирующие). На схемах входы обозначают условными знаками, которые приведены в таблице 1.1.

По способу управления триггеры бывают со статическим и динамическим управлением. Триггеры со *статическим* управлением срабатывают в момент, когда входной сигнал достигает порогового уровня. При *динамическом* управлении триггеры реагируют на перепад управляющего сигнала. Входы также делятся на статические и динамические. Инверсные статические входы и выходы триггера обозначают символом логического отрицания \bar{O} , а динамические прямые и инверсные входы – соответственно указателями \triangleright и \triangleleft . Прямые статические входы и выходы триггера указателей не имеют.

Т а б л и ц а 1.1 – **Обозначения входов триггеров**

Тип входов	Условное обозначение	Функциональное назначение
Информационные	S	Вход раздельной установки триггера в состояние 1
	R	То же в состояние 0
	J	Вход установки JK -триггера в состояние 1
	K	То же в состояние 0
	T	Вход счетного триггера (счетный вход)
	D	Информационный вход D - и DV -триггера
Управляющие	V	Подготовительный вход разрешения приема информации
	C	Исполнительный вход приема информации, вход синхронизации (тактирующий вход)

По способу ввода информации триггеры подразделяют на *асинхронные* и *синхронные*. В асинхронных триггерах ввод информации определяется изменением сигналов на логических входах. В синхронных триггерах смены информационных сигналов еще не достаточно для их переключения – нужен дополнительный командный импульс, который подается на *синхронизирующий* (тактирующий) вход C . При синхронизации триггер не реагирует на входные сигналы и в том числе на помехи в интервалах между тактовыми импульсами, что способствует повышению его помехоустойчивости.

Основным и наиболее общим классификационным признаком является функциональный, который позволяет подразделять триггеры по их функциональному назначению и числу входов.

Существование двух устойчивых состояний делает неоднозначным соотношение между логическими переменными на входах и выходах триггера. Триггеры являются простейшими последовательными устройствами, для определения выходных сигналов которых при смене информации на входах необходимо знать их состояние до этого момента.

Описание работы триггера задается в виде *таблицы переходов* (состояний), определяющей состояние триггера Q^{n+1} в момент времени t_{n+1} , если известно его состояние Q^n и значение входного сигнала X^n в момент времени t_n .

При одном входе возможны только два значения входного сигнала и две имеющие практическое значение таблицы состояний (таблицы 1.2 и 1.3).

Триггер, реализующий таблицу 1.2, называют D -триггером, а триггер, реализующий таблицу 1.3, называется T -триггером или счетным триггером. В алгебраической форме алгоритм, реализуемый D -триггером, имеет вид $Q^{n+1} = D^n$, т. е. он воспроизводит входной сигнал D со сдвигом на один такт, поэтому его называют триггером задержки.

Т а б л и ц а 1.2 – Таблица состояний *D*-триггера

Входной сигнал D^n	Состояние Q^{n+1}	
	$Q^n = 0$	$Q^n = 1$
0	0	0
1	1	1

Т а б л и ц а 1.3 – Таблица состояний *T*-триггера

Входной сигнал T^n	Состояние Q^{n+1}	
	$Q^n = 0$	$Q^n = 1$
0	0	1
1	1	0

T-триггер меняет свое состояние всякий раз, когда меняется входной *T*-сигнал с 0 на 1. В алгебраической форме алгоритм функционирования этого триггера описывается уравнением $Q^{n+1} = Q^n T^n \vee \overline{Q^n T^n}$. Это уравнение совпадает с логической функцией операции «исключающее ИЛИ», т. е. *T*-триггер выполняет операцию сложения по модулю два входной переменной, чем и объясняется его название «счетный триггер».

При двух входах возможны три или четыре значения входного сигнала. В первом случае существуют четыре существенно различные таблицы переходов, но практически используется лишь одна, соответствующая так называемому *RS*-триггеру. В этом триггере с двумя входами *R* и *S* одна из комбинаций входного сигнала (00 или 11) не изменяет его состояния, а другая (01 или 10) устанавливает в нуль, а третья (10 или 01) – в единицу. Четвертая комбинация является запрещенной. Функция переходов для *RS*-триггера с прямыми и инверсными входами приведена в таблице 1.4.

Т а б л и ц а 1.4 – Таблица состояний *RS*-триггера

Сигнал на входе				Состояние <i>RS</i> -триггера Q^{n+1}	
при прямых входах		при инверсных входах		$Q^n = 0$	$Q^n = 1$
S^n	R^n	$\overline{S^n}$	$\overline{R^n}$		
0	0	1	1	0	1
0	1	1	0	0	0
1	0	0	1	1	1

Аналитически работа *RS*-триггера описывается логическим выражением $Q^{n+1} = S^n \vee \overline{Q^n R^n}$ при условии, что $RS = 0$. При инверсных входах значения сигналов *S* и *R* заменяются инверсными, т. е. $Q^{n+1} = \overline{S^n} \vee \overline{Q^n R^n}$.

При четырех значениях входного сигнала нет запрещенных комбинаций и можно образовать только одну функцию переходов, имеющую практическое значение (таблица 1.5). Двоичные логические сигналы и входы в данном случае принято обозначать *J* и *K* и триггер называть *JK*-триггером.

В *JK*-триггере первые три комбинации имеют то же значение, что и в *RS*-триггере, но четвертая является не запрещенной, а приводит к

переключению триггера в состояние, инверсное по отношению к предыдущему.

Т а б л и ц а 1.5 – Таблица состояний JK-триггера

Входной сигнал		Состояние JK-триггера Q^{n+1}	
J^n	K^n	$Q^n = 0$	$Q^n = 1$
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	0

В алгебраической форме алгоритм работы JK-триггера имеет вид:
 $Q^{n+1} = \overline{Q^n} J^n \vee Q^n \overline{K^n}$.

В настоящее время существуют триггеры в интегральном и дискретном исполнении. Рассмотрим основные типы триггеров на интегральных компонентах.

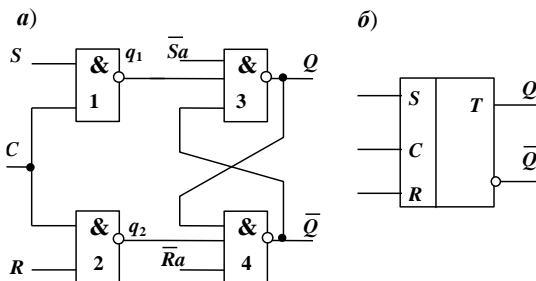
Асинхронный RS-триггер представляет собой *бистабильную* ячейку на логических элементах ИЛИ-НЕ либо И-НЕ (см. рисунок 1.1). Для реализации алгоритма работы RS-триггера нет необходимости в преобразовании входной информации, и устройство управления может отсутствовать. Входы триггера на элементах ИЛИ-НЕ (И-НЕ) закрыты при нулевом (единичном) значении входного сигнала, и его состояние определяется только внутренними сигналами. Поэтому комбинация $S = R = 0$ ($S = R = 1$) называется нейтральной или режимом хранения информации, в котором триггер может долго сохранять любое из устойчивых состояний. На этом основано использование RS-триггера в качестве ячейки памяти.

При комбинациях, когда переключающий сигнал действует на одном входе (наборы 01 или 10), триггер переключается или подтверждается существующее состояние. Если переключающие сигналы подать одновременно на оба входа ($S = R = 1$) при базисе ИЛИ-НЕ и ($S = R = 0$) при базисе И-НЕ, на обоих входах появляются одинаковые сигналы (нули, единицы), и в схеме может быть нарушен бистабильный режим работы. Поведение триггера становится непредсказуемым. Если входная комбинация сменится на нейтральную, то он может перейти как в нулевое, так и в единичное состояние. Комбинации, нарушающие *бистабильный режим*, называются запрещенными, и они не должны допускаться во время работы триггера.

Из асинхронных триггеров можно получить синхронные, если подключить к ним логические элементы, имеющие *синхронизирующий (тактовый) вход C* (рисунок 1.2). Логические элементы 3, 4 образуют ячейку памяти (бистабильную ячейку), а 1, 2 – простейшую схему управления. При отсутствии тактового сигнала ($C = 0$) элементы 1 и 2

закрываются, внутренние сигналы $q_1 = q_2 = 1$, и имеет место режим хранения информации. Информация с входов S и R может быть передана в триггер только при единичном значении сигнала на тактовом входе C . Входные сигналы инвертируются элементами 1 и 2, а входы S и R – прямые.

Рисунок 1.2 – Логическая структура синхронного RS-триггера со статическим управлением (а) и его функциональная схема (б)



Синхронные триггеры снабжают вводами, позволяющими осуществлять *асинхронное управление* (входы \bar{S}_a , \bar{R}_a показаны штриховой линией), что расширяет их функциональные возможности. При синхронном управлении на асинхронных входах \bar{S}_a , \bar{R}_a должны поддерживаться логические единицы.

В триггерах со статическим управлением возможна неоднократная смена состояния за время действия тактового импульса, что в отдельных случаях может быть недопустимо. От этого свободны триггеры с *динамическим управлением* и *двухступенчатые* триггеры.

RS-триггер с двухступенчатой или *MS*-структурой состоит из двух одинаковых синхронных RS-триггеров со статическим управлением (рисунок 1.3).

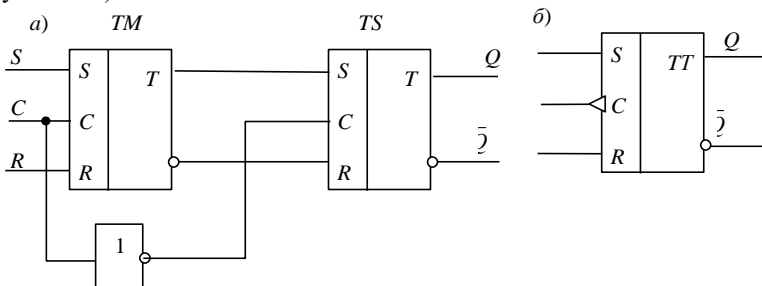


Рисунок 1.3 – Логическая (а) и функциональная (б) схемы двухступенчатого RS-триггера

При $C = 0$ информационные входы ведущего *TM* триггера заперты, и

он хранит информацию от предыдущего такта. Ведомый RS -триггер, на синхронизирующем входе которого в это время действует единица, повторяет состояние ведущей ячейки. Если $C = 1$, то в TM заносится информация с логических входов S , R , а ведомая ячейка TS блокируется. Информация из первой ступени перезаписывается во вторую по окончании действия тактового импульса. Двухступенчатая RS -структура эквивалентна триггеру с инверсным входом динамического управления C .

Функциональные возможности RS -триггера ограничены наличием у него неопределенных состояний. Эти триггеры широко используют в качестве ячеек памяти в оперативных запоминающих устройствах и в триггерах более сложной структуры, но в большинстве серий микросхем RS -триггеры как самостоятельные изделия отсутствуют.

JK -триггеры обладают универсальными функциональными возможностями. При всех значениях входного сигнала, кроме набора $J = K = 1$, они действуют подобно RS -триггеру (вход J играет роль S -входа, а вход K – роль R -входа). Комбинация $J = K = 1$ не нарушает бистабильный режим, а в каждом такте переключает триггер. Для реализации такого алгоритма в JK -триггере используют сигналы обратной связи с выходов Q и \bar{Q} для блокировки одного из информационных входов в зависимости от состояния триггера в предыдущем такте.

JK -триггеры усложняют в связи с необходимостью предотвращения явления генерации. Генерация возможна из-за того, что смена состояния триггера при $J = K = 1$ вновь приводит к комбинации сигналов на входах, вызывающей новое переключение. Просто и надежно явление генерации устраняется в двухступенчатых триггерах (рисунок 1.4).

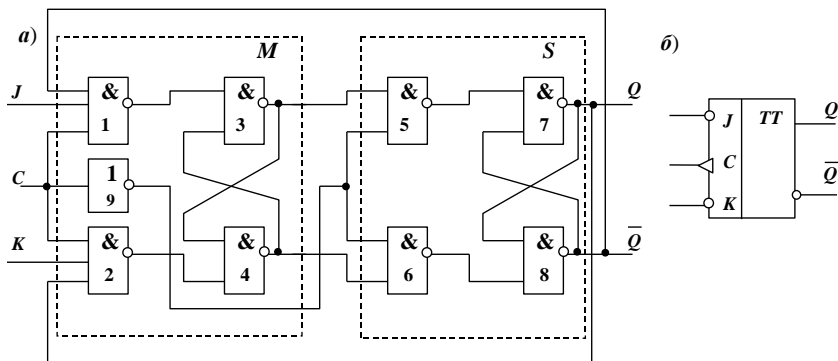


Рисунок 1.4 – Логическая (а) и функциональная (б) схема двухступенчатого JK -триггера

В нулевом положении триггера ($Q = 0$) блокирован вход K , в единичном ($Q = 1$) – вход J , и состояние первой ступени M при $C = 1$ определяется

сигналом на незаблокированном входе. При $C = 0$ блокируются оба входа J и K , а информация из предварительной ячейки M переносится в основную ячейку S . Благодаря поочередному *тактированию* ступеней входы J и K никогда не бывают открыты одновременно, и явление генерации невозможно. Наличие цепей обратной связи и поочередное блокирование ступеней являются основным условием реализации двухступенчатых JK -триггеров.

В течение первых двух тактов на входах J и K имеют место комбинации 10 и 01, и триггер функционирует подобно RS -триггеру. В следующих двух тактах $J = K = 1$ и в схеме имеет место *счетный режим*. Наконец, в последнем пятом такте $J = K = 0$, что соответствует *режиму хранения информации*.

Триггеры с одним информационным входом можно получить из RS - или JK -триггеров за счет соответствующего соединения их логических входов. Так, D -триггер (рисунок 1.5, *a* и *б*) можно получить, обеспечив в RS -триггере функциональную связь $R = \bar{S}$ между сигналами на входах R и S .

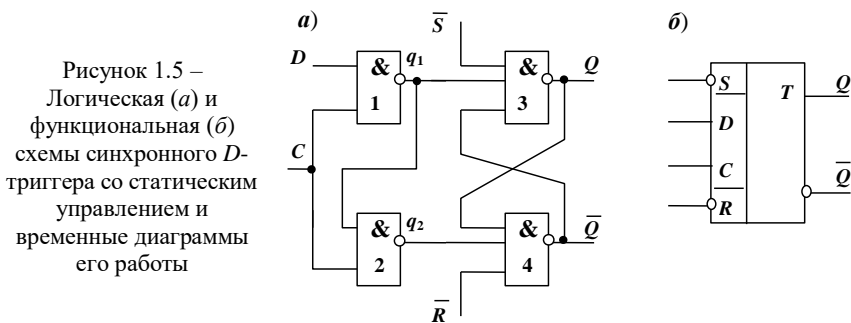


Рисунок 1.5 –
Логическая (а) и
функциональная (б)
схемы синхронного D -
триггера со статическим
управлением и
временные диаграммы
его работы

Логическая структура D -триггера состоит из *основной ячейки памяти* (элементы 3, 4), *RS-триггера* и *схемы управления* (элементы 1,2) (см. рисунок 1.5, *a*).

В паузах между тактовыми импульсами $C = 0$, управляющие элементы 1, 2 заблокированы, $q_1 = q_2$, и на входах основной ячейки действует нейтральная комбинация. При приходе синхронизирующего импульса сигналы на выходе схемы управления становятся равными $q_1 = \bar{D}$, $q_2 = D$, и триггер переключается или подтверждает его предыдущее состояние.

Хранение информации в D -триггере обеспечивается за счет сигнала синхронизации. При отсутствии тактирующего входа эта структура перестает быть триггером и превращается в устройство, не обладающее свойством памяти.

D -триггеры с динамическим управлением могут быть реализованы на

основе нескольких *RS*-триггеров (рисунок 1.6, *а* и *б*). Они состоят из трех *RS*-триггеров: основного на элементах 5, 6 и двух *коммутирующих* на элементах 1, 2 и 3, 4, образующих *схему управления*.

При $C = 0$ элементы 2 и 3 заблокированы, и имеет место режим хранения информации ($q_2 = q_3 = 1$). Сигналы q_1 и q_4 воспроизводят сигнал D : q_1 в инверсном, а q_4 в прямом виде, в результате чего предварительно записывается информация: при $D = 0$ – в ячейку на элементах 3, 4, при $D = 1$ – в ячейку на элементах 1, 2. Другая ячейка при $C = 0$ находится в неопределенном состоянии. Информация записывается в основную ячейку в момент прихода тактирующего импульса. Элементы 2, 3 деблокируются, и ячейка, находившаяся в неопределенном состоянии, принимает то же состояние, что и триггер, хранящий информацию. В результате один из сигналов q_2 (при $D = 0$) или q_3 (при $D = 1$) принимает значение логического нуля, и триггер переключается или подтверждается состояние основной ячейки.

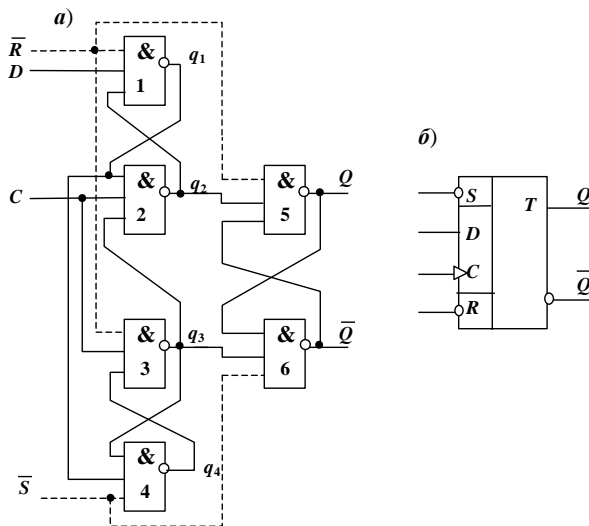


Рисунок 1.6 – Логическая (*а*) и функциональная (*б*) схемы *D*-триггера с динамическим управлением

Одновременно эти же сигналы блокируют ввод новой информации – при $q_2 = 0$ блокируется элемент 1, а при $q_3 = 0$ – элементы 2 и 4. Запись новой информации возможна только после окончания тактового импульса. Таким образом, в шестиэлементный триггер информация вводится на перепаде от 0 к 1 синхронизирующего импульса, и вход C является динамическим тактовым входом.

При асинхронном управлении состояние выходной ячейки определяется

входом, на который воздействует сигнал логического нуля. Асинхронный сигнал подается также на управляющие ячейки так, чтобы они подтверждали состояние выходного триггера. Например, при $S = 0$ блокируется элемент 4 ($q_4 = 1$), и в течение тактового импульса ($C = 1$) на выходе элемента 3 формируется логический нуль ($q_3 = 0$), который в свою очередь блокирует элемент 2 ($q_2 = 1$). При $R = 0$ блокируются элементы 1 и 3, что обеспечивает сигналы $q_2 = 0, q_3 = 1$.

Счетный триггер (T -триггер) является частным случаем JK -триггера, когда при соединении входов выполняется соотношение $K = J$. Поэтому T -триггеры также склонны к генерации и могут быть реализованы как триггеры с динамическим управлением или двухступенчатой структурой.

В виде самостоятельных изделий T -триггеры не выпускают, их получают из триггеров других типов (рисунок 1.7).

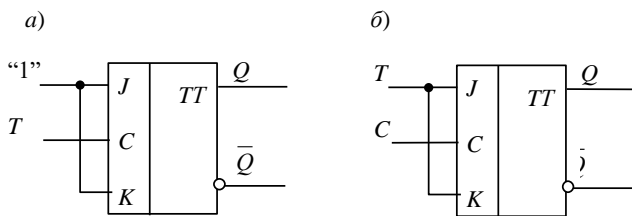


Рисунок 1.7 – Логические схемы асинхронного (а) и синхронного (б) T -триггеров, преобразованных из двухступенчатого синхронного JK -триггера

В асинхронном режиме тактирующий вход C служит в качестве счетного входа T . В синхронном режиме вход C используется по прямому назначению, а информация подается на соединенные входы J и K . Триггер меняет состояние при перепаде сигнала T от 0 к 1. T -триггер – единственный вид триггера, состояние которого определяется не информацией на входах, а его состоянием в предыдущем такте.

Триггеры D - и RS -типа также можно преобразовать в T -триггер (рисунок 1.8).

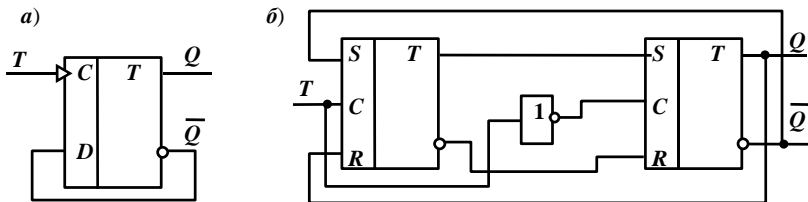


Рисунок 1.8 – Логические схемы T -триггеров, преобразованных из D -триггера с динамическим управлением (а) и двухступенчатого RS -триггера (б)

Общий принцип построения счетных триггеров состоит во введении обратной связи с выходов на входы так, чтобы обеспечивалась смена сигналов на информационных входах после каждого переключения триггера. В качестве входа T в обеих структурах используется тактирующий вход C .

Функциональные возможности триггеров дополнительно расширяются при наличии управляющего входа V (модификации DV , TV). Наличие V -входа позволяет сохранить информацию в триггере (при $V = 0$) независимо от смены сигналов на информационных входах. Запись информации возможна только при $V = 1$. Логическая функция триггера с V -входом: $Q_V^{n+1} = Q^{n+1}V^n$, где Q_V^{n+1} – логическая функция триггера без V -входа, V^n – сигнал на входе V в момент t_n .

Функционально V -вход аналогичен синхронизирующему входу C и их можно менять местами, не нарушая логику работы. Для преобразования любого триггера в модификацию с V -входом достаточно добавить еще один вход, логически связанный с входом C операцией И (рисунок 1.9).

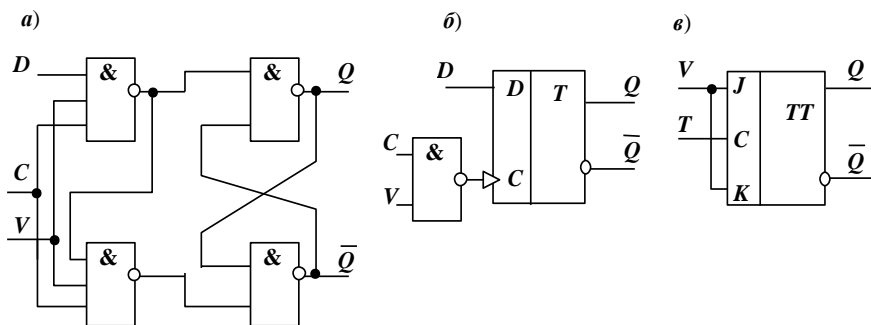


Рисунок 1.9 – Логические схемы DV -триггера (а), DV -триггера (б) и TV -триггера (в), преобразованных соответственно из D - и JK -триггера

Триггеры широко используют для построения более сложных функциональных устройств (счетчиков, регистров, распределителей и др.).

2 Порядок выполнения работы

1 Изучить теоретический материал по учебнику и данным методическим указаниям.

2 Пользуясь средствами программы «Электронная лаборатория», собрать схему триггера в соответствии с вариантом, указанным преподавателем в таблице 1.6.

3 Подавая входные сигналы в соответствии с таблицей истинности

заданного триггера, построить временную диаграмму его работы, используя логический анализатор программы «Электронная лаборатория».

4 Сделать выводы о характере и особенностях работы исследуемого триггера.

Содержание отчета

Название и цель работы; функциональные схемы всех типов триггеров, указанных в таблице 1.6; логическая схема заданного триггера и временная диаграмма его работы; ответы на контрольные вопросы (по заданию преподавателя); выводы о характере и особенностях работы исследуемого триггера.

Т а б л и ц а 1.6 – **Варианты заданий**

Вариант	Тип триггера	Наличие синхронизации	Количество ступеней	Базис реализации
1	<i>T</i>	нет	1	ИЛИ-НЕ
2	<i>D</i>	нет	1	И-НЕ
3	<i>RS</i>	да	1	И-НЕ
4	<i>RS</i>	да	2	И-НЕ
5	<i>JK</i>	да	1	И-НЕ
6	<i>RS</i>	нет	1	ИЛИ-НЕ
7	<i>JK</i>	да	2	И-НЕ
8	<i>JK</i>	нет	1	И-НЕ
9	<i>DV</i>	да	1	И-НЕ
10	<i>T</i>	да	1	И-НЕ
11	<i>TV</i>	нет	2	И-НЕ
12	<i>T из JK</i>	нет	1	И-НЕ
13	<i>RS</i>	нет	1	И-НЕ

П р и м е ч а н и е – В случае недостатка вариантов, преподавателем дополнительно может задаваться базис реализации, обратный заданному в таблице (т. е. если в таблице стоит «И-НЕ», то может быть задан вариант «ИЛИ-НЕ» и наоборот – для одинаковых прочих условий).

Контрольные вопросы

- 1 Что называется триггером?
- 2 Приведите основные типы триггеров и дайте их характеристику.
- 3 В чем заключается особенность двухступенчатых триггеров?
- 4 Приведите схемы получения *T*-триггера на базе *JK*-триггера и на базе *D*-триггера.
- 5 В чем заключается различие между статическими и динамическими триггерами?
- 6 Выполните классификацию триггеров по способу ввода информации с краткой характеристикой особенностей.

- 7 Каковы особенности счетного триггера?
- 8 Дайте характеристику особенностей управления триггерами, построенными в базах «И-НЕ» и «ИЛИ-НЕ».
- 9 Приведите общую логическую схему получения двухступенчатого триггера на базе одноступенчатого.
- 10 Опишите принцип действия двухступенчатого *RS*-триггера.
- 11 Опишите принцип действия двухступенчатого *JK*-триггера.
- 12 Опишите принцип действия *D*-триггера с динамическим управлением.
- 13 В чем заключается общий принцип построения счетных триггеров на базе обычных (несчетных) триггеров?
- 14 Какое преимущество дает наличие *V*-входа у триггеров?
- 15 Прокласифицируйте входы триггеров по функциональному назначению.

Лабораторная работа № 2

ИЗУЧЕНИЕ СУММАТОРОВ

Ц е л ь р а б о т ы. Изучить назначение, структуру и принцип действия двоичных сумматоров.

1 Сведения из теории

Двоичные сумматоры выполняют арифметическую операцию сложения двоичных чисел по модулю два (\oplus) и используются преимущественно в арифметико-логических устройствах современных ЭВМ и цифровых фильтрах.

Существует **два вида сумматоров**: с *параллельным и последовательным* действием. У сумматоров с параллельным действием сложение выполняется параллельно, сразу во всех разрядах суммируемых чисел. В сумматорах с последовательным действием имеется только одна одnorазрядная суммирующая схема и результат образуется последовательным сложением отдельных разрядов.

Рассмотрим одnorазрядный сумматор параллельного действия. Таблица истинности (таблица 2.1) такого сумматора для *i*-го разряда содержит три двоичных входа: a_i – первое слагаемое, b_i – второе слагаемое и p_{i-1} – перенос из *i*–1 разряда в *i*-й.

В результате сложения этих трех двоичных чисел образуется двухразрядное число, поэтому сумматор должен иметь два двоичных выхода. Младший выход s_i результата сложения называется частичной суммой, а старший выход p_i – переносом.

Исходя из таблицы истинности, запишем функции выходов s_i и p_i :

$$s_i = \bar{a}_i \bar{b}_i p_{i-1} \vee \bar{a}_i b_i \bar{p}_{i-1} \vee a_i \bar{b}_i \bar{p}_{i-1} \vee a_i b_i p_{i-1}; \quad p_i = \bar{a}_i b_i p_{i-1} \vee a_i \bar{b}_i p_{i-1} \vee$$

$$\vee a_i b_i \bar{p}_{i-1} \vee a_i b_i p_{i-1} = a_i b_i \vee p_{i-1} (b_i \vee a_i).$$

Т а б л и ц а 2.1 – Таблица истинности одноразрядного сумматора

Входы			Выходы	
a_i	b_i	p_{i-1}	s_i	p_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

На основе полученных формул построим схему i -го разряда параллельного сумматора в базе И-ИЛИ-НЕ, представленную на рисунке 2.1.

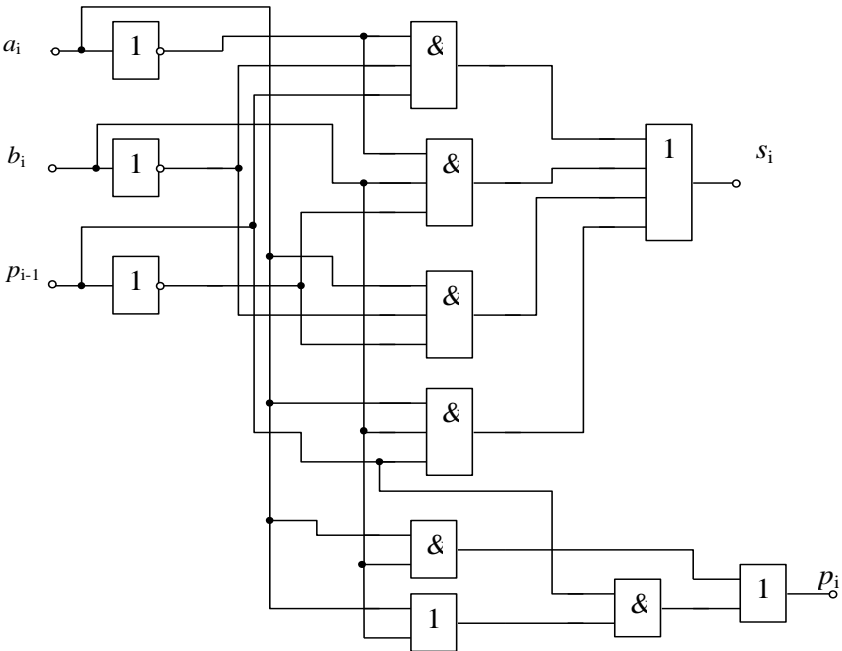


Рисунок 2.1 – Функциональная схема i -го разряда полного параллельного сумматора

Рассмотренный сумматор является полным. Если при синтезе сумматора

не учитывается перенос из предыдущего разряда, то такой сумматор называется полусумматором. Его таблица истинности имеет вид, указанный в таблице 2.2.

Функции выходов полусумматора: $s_i = \bar{a}_i b_i \vee a_i \bar{b}_i$; $p_i = a_i b_i$.

По функциям выходов полусумматора можно получить его схему, например, в базисе ИЛИ-НЕ. Для этого преобразуем выражения для s_i и p_i , используя правило де Моргана:

$$s_i = \bar{a}_i b_i \vee a_i \bar{b}_i = \overline{\overline{\bar{a}_i b_i} \cdot \overline{a_i \bar{b}_i}} = \overline{(a_i \downarrow \bar{b}_i) \downarrow (\bar{a}_i \downarrow b_i)}, \quad p_i = a_i b_i = \overline{\bar{a}_i \vee \bar{b}_i} = \bar{a}_i \downarrow \bar{b}_i.$$

Построим функциональную схему полусумматора (рисунок 2.2).

Таблица 2.2 – Таблица истинности полусумматора

Входы		Выходы	
a_i	b_i	s_i	p_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

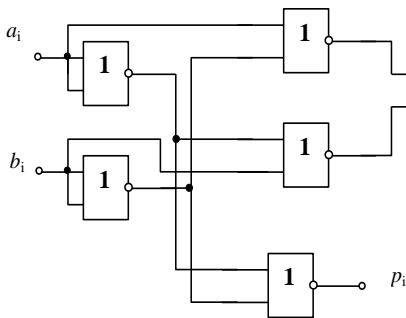


Рисунок 2.2 – Функциональная схема полусумматора

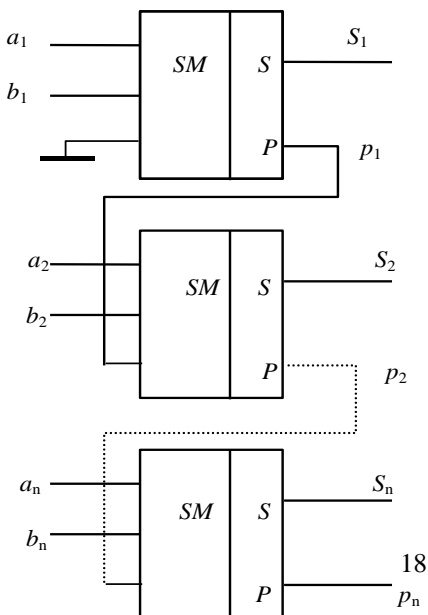


Рисунок 2.3 – Логическая схема n -разрядного сумматора

Для получения n -разрядных сумматоров соединяют n одноразрядных полных или полусумматоров (рисунок 2.3).

Как уже отмечено, сумматоры последовательного действия имеют в своем составе только одну суммирующую схему. На соответствующие входы этой схемы подаются числа a_i и b_i , начиная с младших разрядов. На выходе формируется сумма $s_i = a_i \oplus b_i$. Так как операция сложения происходит поразрядно, то на

каждом шаге работы необходимо помнить значения переноса из младшего разряда. Операция сложения в некотором i -м разряде ($i = 1, 2, \dots, n$) в общем виде задается следующей системой собственных функций:

$$\begin{cases} p_i = \varphi_1(a_i, b_i, p_{i-1}); \\ s_i = \psi_1(a_i, b_i, p_{i-1}) = a_i \oplus b_i \oplus p_{i-1}. \end{cases}$$

Сумматор последовательного действия является конечным автоматом. Роль времени t здесь играет индекс i . Схема имеет два состояния (отсутствие и наличие переноса из младшего разряда) и один выход.

Формулы для одноразрядного последовательного сумматора формально совпадают с приведенными для параллельного сумматора. Однако логическая схема имеет несколько иной вид (рисунок 2.4).

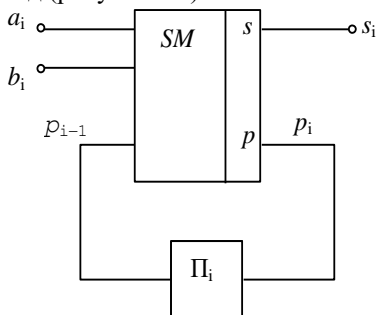


Рисунок 2.4 – Логическая схема сумматора последовательного действия (Π_i – память предыдущего состояния)

Если в качестве памяти предыдущего состояния использовать T -триггер, образованный из D -триггера, то логическая схема последовательного сумматора будет иметь вид, показанный на рисунке 2.5.

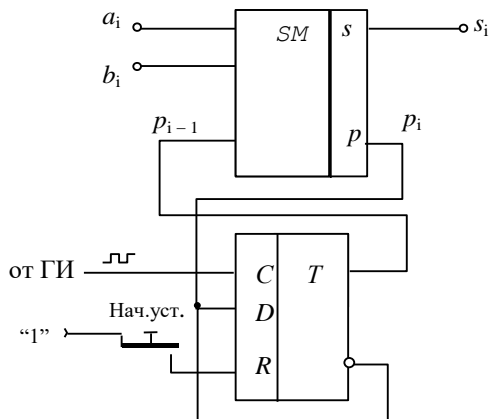


Рисунок 2.5 – Логическая схема последовательного сумматора с D -триггером в качестве памяти предыдущего состояния

Следует помнить, что при использовании триггеров в качестве элементов памяти необходимо предусматривать установку их в начальное (в частности, нулевое) состояние. Это обусловлено неопределенностью исходного состояния триггера в начальный момент работы последовательного сумматора.

Последовательный сумматор для сложения одновременно трех слагаемых в дополнительном коде, разряды которых подаются на входы сумматора один за другим, показан на рисунке 2.6. Схема состоит из двух комбинационных одноразрядных сумматоров $SM1$ и $SM2$ и двух триггеров $T1$ и $T2$. Сумматор $SM1$ и триггер $T1$ образуют последовательный сумматор для сложения первых двух слагаемых, i -е разряды которых обозначены x_i^1 и x_i^2 . Сумматор $SM2$ и триггер $T2$ образуют последовательный сумматор для сложения суммы первых двух слагаемых и третьего слагаемого, i -е разряды которых обозначены s_i' и x_i^3 .

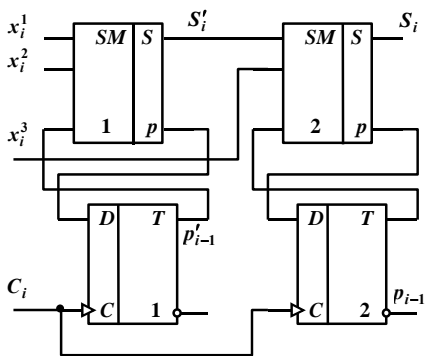


Рисунок 2.6 – Схема сумматора для сложения трех разрядов двоичного числа

Совершенно так же может быть построен последовательный сумматор для сложения четырех слагаемых, разряды которых подаются на входы сумматора один за другим (рисунок 2.7).

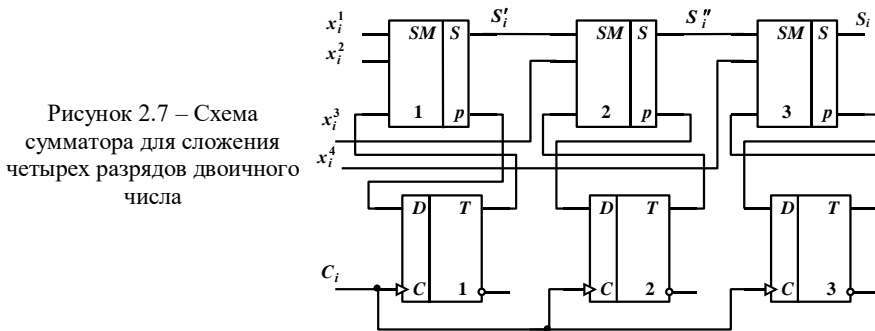


Рисунок 2.7 – Схема сумматора для сложения четырех разрядов двоичного числа

2 Порядок выполнения работы

1 Изучить теоретический материал по учебнику и данным методическим указаниям.

2 Синтезировать схему сумматора при помощи средств программы «Электронная лаборатория» в соответствии с вариантом, указанным преподавателем в таблице 2.3.

3 Получить таблицу истинности построенного сумматора.

4 Сделать вывод о правильности работы построенного сумматора на основе сравнения полученной таблицы истинности и таблицы истинности, приведенной в методических указаниях.

Таблица 2.3 – Варианты заданий

Вариант	Вид сумматора	Количество складываемых разрядов	Базис реализации
1	П полный	5	И-НЕ
2	П полный	5	ИЛИ-НЕ
3	П полный	5	И-ИЛИ-НЕ
4	П полный	7	И-НЕ
5	П полный	7	ИЛИ-НЕ
6	П полный	7	И-ИЛИ-НЕ
7	П – <i>D</i>	3	И-НЕ
8	П – <i>JK</i>	3	ИЛИ-НЕ
9	П – <i>D</i>	3	И-ИЛИ-НЕ
10	П – <i>JK</i>	4	И-НЕ
11	П полный	6	И-НЕ
12	П полный	6	ИЛИ-НЕ
13	П полный	6	И-ИЛИ-НЕ
14	П – <i>D</i>	4	ИЛИ-НЕ
15	П – <i>JK</i>	3	И-ИЛИ-НЕ
16	П полный	8	И-НЕ
17	П полный	8	ИЛИ-НЕ
18	П – <i>D</i>	4	И-НЕ
19	П – <i>JK</i>	4	ИЛИ-НЕ
20	П – <i>D</i>	5	И-ИЛИ-НЕ

Примечание – П – означает сумматор параллельного действия; П – *D* – сумматор последовательного действия с *D*-триггером в качестве памяти; П – *JK* – сумматор последовательного действия с *JK*-триггерами в качестве памяти.

Содержание отчета

Наименование и цель работы; краткие сведения из теории; схема

заданного сумматора с общей таблицей истинности для всех разрядов; ответы на заданные преподавателем контрольные вопросы; выводы по работе.

Контрольные вопросы

- 1 Для чего предназначены сумматоры?
- 2 Чем отличаются сумматор и полусумматор?
- 3 Чем отличаются сумматоры последовательного и параллельного действия?
- 4 Какие элементы для памяти, кроме триггеров, можно использовать в последовательных сумматорах?
- 5 Назовите области применения сумматоров.
- 6 Почему сумматор последовательного действия относится к конечным автоматам?
- 7 Приведите примеры стандартных интегральных микросхем сумматоров, выпускаемых промышленностью.
- 8 Каким образом в ЭВМ осуществляется операция вычитания?
- 9 На базе каких стандартных дискретных элементов могут быть реализованы сумматоры?
- 10 Для чего в сумматорах последовательного действия используется память предыдущего состояния?

Лабораторная работа № 3

СИНТЕЗ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ НА БАЗЕ МУЛЬТИПЛЕКСОРОВ

Ц е л ь р а б о т ы. Изучить методику синтеза и научиться строить схемы комбинационных дискретных устройств на асинхронных и синхронных мультиплексорах.

1 Сведения из теории

Мультиплексоры могут быть использованы при синтезе логических функций. Часто это позволяет сократить число используемых для синтеза микросхем (корпусов) по сравнению с синтезом *функций алгебры логики* (ФАЛ) на обычных логических элементах. ФАЛ k переменных может быть реализована одной микросхемой мультиплексора с k управляющими и $2k - 1$ информационными входами. В этом случае на информационные входы

подаются константы 1 или 0 в зависимости от значения ФАЛ при соответствующем наборе переменных, подаваемых на управляющие входы. На рисунке 3.1 приведена схема реализации ФАЛ, заданной таблицей истинности ТИ (таблица 3.1).

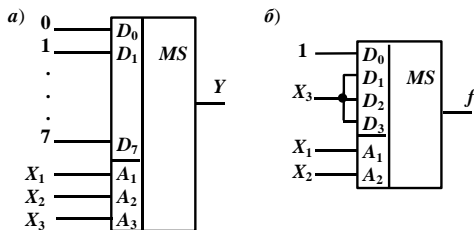
Функция трех переменных может быть реализована мультиплексором с двумя управляющими входами, если две из этих переменных подать на управляющие входы, а третью – на информационные. В этом случае заданную ФАЛ необходимо разложить по переменным, подаваемым на управляющие входы.

Разложение осуществляют с использованием теоремы разложения, в которой говорится, что любая ФАЛ $f(x_1, x_2, \dots, x_n)$ может быть преобразована к иному виду путем разложения по аргументам, например, по x_1 : $f(x_1, x_2, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) \vee \bar{x}_1 f(0, x_2, \dots, x_n)$.

Т а б л и ц а 3.1 – Таблица истинности

Входные переменные			Выходная функция
x_1	x_2	x_3	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Рисунок 3.1 – Схемы реализации функции алгебры логики трех переменных на мультиплексорах с тремя (а) и двумя (б) управляющими входами



В общем случае любую ФАЛ n переменных можно разложить по $i \leq n$ переменным (для определенности будем брать старшие переменные) и представить в виде $f(x_1, x_2, \dots, x_n) = \bar{x}_1 \bar{x}_2 \dots \bar{x}_i F_0 \vee \bar{x}_1 \bar{x}_2 \dots \bar{x}_{i-1} x_i F_1 \vee \dots \vee x_1 x_2 \dots x_i F_{2^i - 1}$, где F_j – функция алгебры логики, получающаяся из исходной ФАЛ подстановкой в нее набора значений переменных с номером $j \in \{0, 1, \dots, 2^i - 1\}$.

Пусть, например, задана функция $f = x_1 x_2 \vee \bar{x}_1 x_3 \vee \bar{x}_2 x_4$. Разложим ее

по x_1 и x_2 : $f = \bar{x}_1\bar{x}_2F_0 \vee \bar{x}_1x_2F_1 \vee x_1\bar{x}_2F_2 \vee x_1x_2F_3$; $F_0 = 0 \cdot 0 \vee 1 \cdot x_3 \vee 1 \cdot x_4 = x_3 \vee x_4$;
 $F_1 = 1 \cdot 0 \vee 1 \cdot x_3 \vee 0 \cdot x_4 = x_3$; $F_2 = 1 \cdot 0 \vee 0 \cdot x_3 \vee 1 \cdot x_4 = x_4$; $F_3 = 1 \cdot 1 \vee 0 \cdot x_3 \vee 0 \cdot x_4 = 1$.

В этом случае $f = \bar{x}_1\bar{x}_2(x_3 \vee x_4) \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_4 \vee x_1x_2$.

При разложении ФАЛ по x_1 , x_2 и x_3 будем иметь $f = \bar{x}_1\bar{x}_2\bar{x}_3x_4 \vee$
 $\vee \bar{x}_1\bar{x}_2x_3 \cdot 1 \vee \bar{x}_1x_2\bar{x}_3 \cdot 0 \vee \bar{x}_1x_2x_3 \cdot 1 \vee x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee x_1x_2\bar{x}_3 \cdot 1 \vee x_1x_2x_3 \cdot 1$.

При разложении функции $f = \bar{x}_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3$ по переменным x_1 и x_2 получим: $f = \bar{x}_1\bar{x}_2F_0 \vee \bar{x}_1x_2F_1 \vee x_1\bar{x}_2F_2 \vee x_1x_2F_3$; $F_0 = 1$; $F_1 = \bar{x}_3$; $F_2 = \bar{x}_3$;
 $F_3 = \bar{x}_3$.

Схема реализации заданной функции приведена на рисунке 3.1, б. Аналогичным способом на одной микросхеме мультиплексора с k управляющими входами можно реализовать любую ФАЛ $k + 1$ переменных.

Если число управляющих входов k_y значительно меньше числа переменных k , на информационные входы подаются функции F_i переменных $(k - k_y)$. Для реализации каждой из этих функций требуется своя схема. Функции F_i находятся в результате разложения исходной ФАЛ по k_y переменным. Следует иметь в виду, что на управляющие входы мультиплексора могут быть поданы разные группы переменных. Разлагая ФАЛ по разным группам переменных, можно получить различные по сложности схемы. Число вариантов разложения велико. В дальнейшем будем полагать, что разложение осуществляется по старшим переменным.

Приведем **алгоритм** реализации ФАЛ k переменных на мультиплексорах с k_y управляющими входами:

1) написание заданной ФАЛ в дизъюнктивной нормальной форме (ДНФ);

2) разложение ФАЛ по k_y переменным;

3) нахождение выражения функций F_i переменных $k - k_y$: $F_i(x_k - k_y, x_k - k_y + 1, \dots, x_k)$. Номер функции F_i совпадает с номером набора k_y переменных, по которым раскладывается заданная ФАЛ;

4) подача на управляющие входы мультиплексора сигналов, соответствующих k_y переменным, по которым раскладывается ФАЛ;

5) подача на информационные входы мультиплексора сигналов, соответствующих значениям функций F_i .

Пр и м е р. Построить на мультиплексорах с двумя управляющими входами схему, реализующую ФАЛ $f = \bar{x}_1\bar{x}_2x_5x_6 \vee \bar{x}_1x_3x_5 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2x_5x_6$.

Разложим функцию по x_1 и x_2 и получим: $f = \bar{x}_1\bar{x}_2(x_5x_6 \vee x_3x_5) \vee$
 $\vee \bar{x}_1x_2(x_3x_5) \vee x_1\bar{x}_2(0) \vee x_1x_2(\bar{x}_3\bar{x}_4 \vee x_5x_6)$

Для реализации ФАЛ $F_0 = x_3x_5 \vee x_5x_6$ и $F_3 = \bar{x}_3x_4 \vee x_5x_6$ будем осуществлять

разложение F_0 по x_3 и x_5 , а F_3 – по x_3 и x_4 : $F_0 = \bar{x}_3\bar{x}_5(0) \vee \bar{x}_3x_5(x_6) \vee x_3\bar{x}_5(0) \vee x_3x_5(1)$,
 $F_3 = \bar{x}_3\bar{x}_4(1) \vee \bar{x}_3x_4(x_5x_6) \vee x_3\bar{x}_4(x_5x_6) \vee x_3x_4(x_5x_6)$.

На основе полученных выражений и алгоритма строим схему (рисунок 3.2).

Использование синхронизирующего входа мультиплексора. Наличие синхронизирующего входа позволяет расширить возможности реализации ФАЛ на мультиплексорах.

Приведем *алгоритм* реализации ФАЛ k переменных с использованием двух мультиплексоров с синхронизирующими входами:

- 1) написание заданной ФАЛ в ДНФ;
- 2) разложение заданной ФАЛ по переменной с индексом i ;
- 3) нахождение выражения функций F_0 и F_1 переменных $(k - 1)$;
- 4) реализация ФАЛ F_0 и F_1 на мультиплексорах с $(k - 2)$ управляющими входами. Сигналы, соответствующие $(k - 2)$ переменным, подать параллельно на управляющие входы обоих мультиплексоров;
- 5) подача сигнала, соответствующего переменной с индексом i на стробирующий вход первого мультиплексора, реализующего ФАЛ F_1 , и через инвертор – на стробирующий вход второго мультиплексора, реализующего ФАЛ F_0 ;
- 6) объединение прямых выходов мультиплексоров, используя элемент ИЛИ.

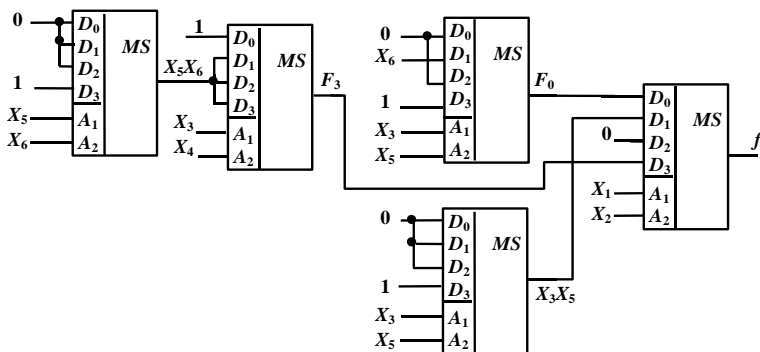


Рисунок 3.2 – Схема реализации функции алгебры логики на четырехвходовых мультиплексорах

Рассмотрим описанный алгоритм на примере.

Пример. Построить схему на мультиплексорах с двумя управляющими и стробирующими входами для ФАЛ четырех переменных $f = x_1x_2x_3 \vee x_2\bar{x}_3x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_4$.

Разложим ФАЛ по переменной x_1 : $f = \bar{x}_1F_0 \vee x_1F_1$; $F_0 = x_2\bar{x}_3x_4 \vee \bar{x}_2\bar{x}_4$;
 $F_1 = x_2x_3 \vee x_2\bar{x}_3x_4$.

Величины F_0 и F_1 являются функциями трех переменных, и могут быть

реализованы на мультиплексорах с двумя управляющими входами на основании вышеприведенного алгоритма. Разложим функции F_0 и F_1 по x_2 и x_3 : $F_0 = \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_3x_4 \vee \bar{x}_2x_3\bar{x}_4 \vee x_2x_3 \cdot 0$; $F_1 = \bar{x}_2\bar{x}_3 \cdot 0 \vee \bar{x}_2x_3 \cdot 0 \vee x_2\bar{x}_3x_4 \vee x_2x_3 \cdot 1$.

Если на синхронизирующие входы мультиплексоров, реализующих F_0 и F_1 , подать сигналы, соответствующие переменной, дизъюнкция выходных функций на прямых выходах обоих мультиплексоров реализует заданную ФАЛ (рисунок 3.3).

Мультиплексоры с числом управляющих входов k_y могут быть использованы совместно с элементами некоторого базиса для реализации ФАЛ k переменных. В этом случае на информационные входы подаются функции $(k - k_y)$ переменных, сформированные элементами выбранного базиса.

Следует учитывать, что, синтезируя дискретное устройство (реализуя ФАЛ) с использованием мультиплексоров, необходимо также построить вариант схемы без использования мультиплексора. Затем сравнить оба варианта и определить, какой из них лучше по числу используемых корпусов интегральных микросхем.

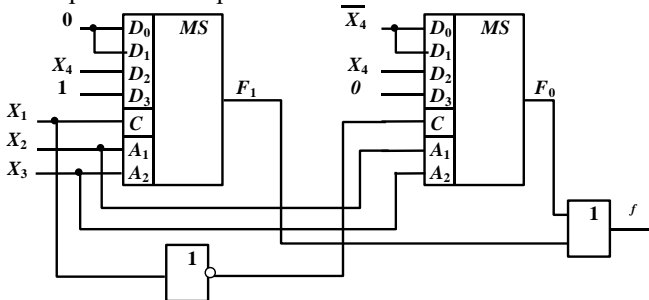


Рисунок 3.3 – Схема реализации функции алгебры логики на мультиплексорах с синхронизирующим входом

2 Порядок выполнения работы

1 Изучить теоретический материал по данным методическим указаниям и учебнику.

2 В соответствии с заданным вариантом (таблица 3.1) осуществить разложение и построить схему (используя программу «Электронная лаборатория») реализации ФАЛ на мультиплексорах.

3 Построить при помощи программы «Электронная лаборатория» таблицу истинности заданной ФАЛ по исходной формуле и проверить правильность функционирования полученной схемы на мультиплексорах путем сверки выходных значений функций по таблице истинности.

4 Построить исходную ФАЛ в программе «Электронная лаборатория» обычным способом (без использования мультиплексоров) и сравнить

варианты реализации.

5 Сделать выводы по сравнению вариантов реализации ФАЛ с использованием мультиплексоров и без их использования.

Содержание отчета

Наименование и цель работы; таблица истинности, полученная по исходной ФАЛ и таблица истинности, полученная при практической проверке схемы; схемы реализаций ФАЛ на мультиплексорах и обычным способом; ответы на контрольные вопросы; выводы по работе.

Контрольные вопросы

1 В каких целях используют разложение ФАЛ по переменным?

2 В чем заключается отличие при реализации ФАЛ на синхронных и асинхронных мультиплексорах?

3 Опишите последовательность операций при реализации ФАЛ на синхронных мультиплексорах.

4 Опишите последовательность операций при реализации ФАЛ на мультиплексорах без стробирующего входа.

5 По какому принципу выбирается вариант (обычным способом или на мультиплексорах) реализации ФАЛ?

6 Сформулируйте теорему разложения.

Таблица 3.1 – Варианты заданий

Вариант	$f = \{...\}_{X_1X_2X_3X_4}$	Вариант	$f = \{...\}_{X_1X_2X_3X_4}$
1	$f = \{0, 2, 5, 8, 14\}_{X_1X_2X_3X_4}$	17	$f = \{6, 7, 10, 15\}_{X_1X_2X_3X_4}$
2	$f = \{1, 3, 6, 9\}_{X_1X_2X_3X_4}$	18	$f = \{7, 9, 11, 14\}_{X_1X_2X_3X_4}$
3	$f = \{2, 4, 6, 7, 10\}_{X_1X_2X_3X_4}$	19	$f = \{0, 1, 8, 12\}_{X_1X_2X_3X_4}$
4	$f = \{3, 5, 8, 11\}_{X_1X_2X_3X_4}$	20	$f = \{2, 3, 9, 13\}_{X_1X_2X_3X_4}$
5	$f = \{4, 6, 9, 12\}_{X_1X_2X_3X_4}$	21	$f = \{2, 5, 9, 10, 14\}_{X_1X_2X_3X_4}$
6	$f = \{2, 5, 7, 10, 13\}_{X_1X_2X_3X_4}$	22	$f = \{0, 1, 7, 11, 12\}_{X_1X_2X_3X_4}$
7	$f = \{6, 8, 11, 14\}_{X_1X_2X_3X_4}$	23	$f = \{0, 1, 2, 14, 15\}_{X_1X_2X_3X_4}$
8	$f = \{7, 9, 12, 15\}_{X_1X_2X_3X_4}$	24	$f = \{0, 2, 3, 15\}_{X_1X_2X_3X_4}$
9	$f = \{0, 8, 10, 13\}_{X_1X_2X_3X_4}$	25	$f = \{2, 3, 4, 14\}_{X_1X_2X_3X_4}$
10	$f = \{1, 4, 9, 11, 14\}_{X_1X_2X_3X_4}$	26	$f = \{3, 4, 5, 12\}_{X_1X_2X_3X_4}$
11	$f = \{2, 10, 12, 15\}_{X_1X_2X_3X_4}$	27	$f = \{5, 7, 8, 11\}_{X_1X_2X_3X_4}$
12	$f = \{1, 5, 10, 15\}_{X_1X_2X_3X_4}$	28	$f = \{9, 10, 11, 15\}_{X_1X_2X_3X_4}$
13	$f = \{2, 6, 11, 15\}_{X_1X_2X_3X_4}$	29	$f = \{0, 13, 14, 15\}_{X_1X_2X_3X_4}$
14	$f = \{0, 3, 7, 12\}_{X_1X_2X_3X_4}$	30	$f = \{1, 2, 10, 12\}_{X_1X_2X_3X_4}$
15	$f = \{4, 8, 10, 13\}_{X_1X_2X_3X_4}$	31	$f = \{1, 3, 10, 15\}_{X_1X_2X_3X_4}$

16	$f = \{2, 5, 9, 14\}_{X_1 X_2 X_3 X_4}$	32	$f = \{1, 5, 6, 7, 9\}_{X_1 X_2 X_3 X_4}$
<p>Примечания</p> <p>1 – Запись вида $f(0, 1, 6, 8, 12, 15)_{X_1 X_2 X_3 X_4}$ означает, что функция $f(X_1, X_2, X_3, X_4)$ на нулевом, первом, шестом, восьмом, двенадцатом и пятнадцатом наборах принимает значение логической единицы, а на всех остальных – логического нуля.</p> <p>2 – Для четных вариантов задания при синтезе схем выбираются двухвходовые мультиплексоры, а для нечетных – трехвходовые.</p> <p>3 – Для вариантов, кратных 2^n, при синтезе выбираются синхронные мультиплексоры, а для остальных – асинхронные мультиплексоры.</p>			

Лабораторная работа № 4

СИНТЕЗ КОМБИНАЦИОННЫХ УСТРОЙСТВ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ МАТРИЦАХ

Ц е л ь р а б о т ы. Изучить основы структуры программируемых логических матриц и принципы построения комбинационных дискретных устройств на их базе.

1 Сведения из теории

В устройствах управления современных микропроцессоров и различных комбинационных структурах широкое применение находят программируемые логические матрицы (ПЛМ), которые во многих случаях представляют собой комбинационные устройства с большим числом входов и выходов. На входы ПЛМ подается в парафазном коде n -разрядное двоичное число, содержащее информацию о коде подлежащей выполнению команды, а также о кодах дополнительных признаков и условий. Входное число должно быть преобразовано в m -разрядное выходное число, являющееся кодом подлежащей выполнению микрокоманды. Обычно длина выходного числа значительно больше длины входного. В рассмотренном случае ПЛМ функционирует как преобразователь кодов.

Методы построения в значительной степени определяются следующими особенностями ПЛМ: сравнительно большой длиной входных и выходных чисел (до нескольких десятков бит); возможностью изменения связей между логическими элементами в процессе изготовления ПЛМ с целью реализации различных модификаций устройств. Обычно ПЛМ содержит несколько групп уровней логических элементов; связи между группами программируются, а необходимые соединения выполняются дополнительными технологическими операциями металлизации.

На рисунке 4.1 изображена структура ПЛМ, содержащая логические элементы трех уровней. При использовании универсальных логических элементов И-НЕ для преобразования входного числа в произвольное выходное на основе, например, дизъюнктивной нормальной формы (ДНФ),

достаточно иметь два уровня элементов. Количество логических элементов каждого уровня в такой структуре резко возрастает при увеличении разрядности входных и выходных чисел. Применение дополнительного уровня инверторов на выходе позволяет сократить общее число логических элементов.

В рассматриваемой структуре сигналы со второго уровня попадают на выход либо непосредственно, либо через соответствующий инвертор. Выбор необходимых связей между логическими элементами различных уровней представляет собой достаточно трудоемкую задачу и выполняется обычно с помощью ЭВМ.

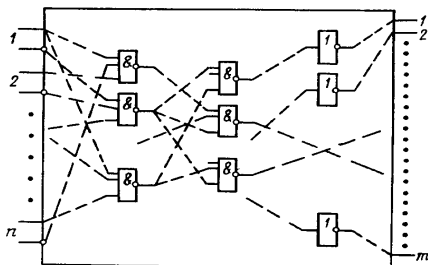


Рисунок 4.1 – Структура ПЛМ

В принципе, ПЛМ является универсальным элементом, который может быть запрограммирован на реализацию любой функции алгебры логики (ФАЛ) определенной сложности. Это устройство содержит матрицы $M1$ и $M2$ (рисунок 4.2), входной буфер, формирующий парафазный код от входных переменных x_1, x_2, \dots, x_s , выходной буфер, усиливающий и формирующий выходные сигналы z_1, z_2, \dots, z_t . Программируемая логическая матрица реализует систему ФАЛ:

$$z_1 = f_1(x_1, x_2, \dots, x_s); \quad z_2 = f_2(x_1, x_2, \dots, x_s); \quad \dots; \quad z_t = f_t(x_1, x_2, \dots, x_s).$$

Матрица $M1$ формирует конъюнкции F_i входных переменных, матрица $M2$ – дизъюнкции конъюнкций F_i , необходимые для реализации соответствующих ФАЛ выходов.

Примерное построение ПЛМ показано на рисунке 4.3. В узлах матриц включают МОП-транзисторы. Программирование матриц заключается в изъятии тех или иных транзисторов, оставшиеся транзисторы обеспечивают реализацию заданных ФАЛ. На рисунке 4.3 узлы матриц, в которых не изъятые транзисторы, обозначены точкой.

В зависимости от принципа реализации матриц и используемой технологии ПЛМ может программироваться в процессе изготовления ее на заводе с применением масок, выполняемых по заданию заказчика или самим заказчиком с использованием специального вспомогательного оборудования.

На каждом выходе матриц $M1$ и $M2$ реализуется функция ИЛИ-НЕ тех

переменных, шины которых через транзисторы соединены с соответствующей выходной шиной. В частности, для матрицы $M1$ (см. рисунок 4.3) шина F_1 соединена с шинами переменных x_1 и \bar{x}_2 . При подаче высокого потенциала (сигнал логической 1) на затвор транзистора $VT1$ или $VT2$ один из них открывается, и на шине F_1 появляется низкий потенциал (сигнал логического 0), так как напряжение источника питания E падает на резисторе $R1$.

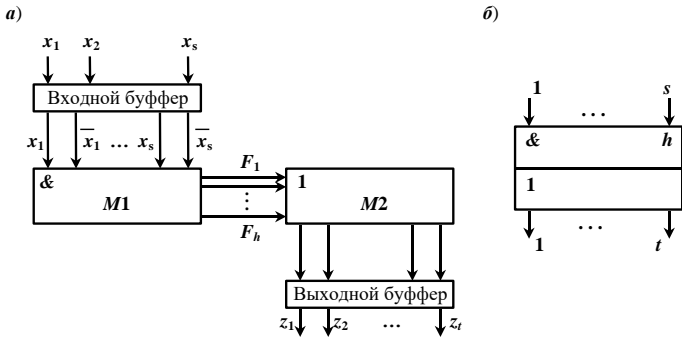


Рисунок 4.2 – Структурная схема (а) и условное обозначение (б) программируемой логической матрицы

Таким образом, ФАЛ для выхода F_1 имеет вид: $F_1 = \overline{x_1 \vee \bar{x}_2} = \bar{x}_1 x_2$;

Аналогично для F_2 и F_3 получим: $F_2 = \overline{\bar{x}_2 \vee x_3} = x_2 \bar{x}_3$; $F_3 = \overline{x_1 \vee x_3} = \bar{x}_1 \bar{x}_3$.

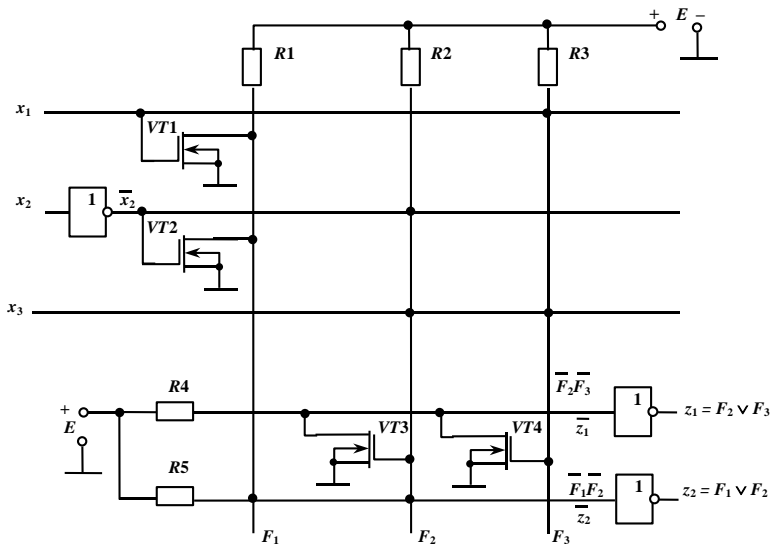


Рисунок 4.3 – Принципиальная схема программируемой логической матрицы

Для матрицы $M2$ входными переменными являются функции F_1, F_2 и F_3 . При показанных на рисунке 4.4 местах включения транзисторов реализуются следующие логические функции: $\bar{z}_1 = F_2 \vee F_3 = \bar{F}_2 \bar{F}_3$; $z_1 = \overline{\bar{F}_2 \bar{F}_3} = F_2 \vee F_3$; $\bar{z}_2 = \overline{F_1 \vee F_2} = \bar{F}_1 \bar{F}_2$; $z_2 = \overline{\bar{F}_1 \bar{F}_2} = F_1 \vee F_2$.

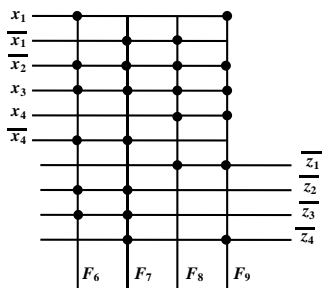


Рисунок 4.4 – Точечное начертание программ ПЛМ

Реализация ФАЛ на ПЛМ осуществляется по следующему алгоритму:

- 1) написание заданной ФАЛ в ДНФ;
- 2) постановка в соответствие каждой конъюнкции F_i , входящей в ДНФ, определенной шины матрицы $M1$;
- 3) инвертирование переменных, входящих в конъюнкцию F_i ;
- 4) сохранение транзисторов в узлах матрицы $M1$, соответствующих пересечению шины конъюнкции F_i с цепями инверсных значений

переменных, входящих в конъюнкцию F_i ;

5) сохранение транзисторов в узлах матрицы M_2 , соответствующих пересечению цепи выходной ФАЛ с шинами конъюнкций F_i , входящих в ДНФ.

Функцию алгебры логики при реализации на ПЛМ не обязательно приводить к минимальной форме. Преобразование ФАЛ проводится лишь в случаях недостаточного числа выходных цепей матриц M_1 и M_2 . Выпускаемые в настоящее время ПЛМ имеют следующие параметры: число входов $s = 14 - 20$; число выходов $t = 8 - 24$; число конъюнкций $h = 48 - 96$. При необходимости можно расширить ПЛМ по выходам и конъюнкциям композицией нескольких ПЛМ.

П р и м е р. Построить преобразователь из кода 2421 в код 8421 на ПЛМ для цифр в десятичном эквиваленте от трех до девяти.

Как известно, при синтезе комбинационных дискретных устройств пользуются таблицами истинности, которые задают алгоритм функционирования каждого конкретного дискретного устройства. Поэтому, для получения выражений функций преобразователя, построим его таблицу истинности (ТИ) (таблица 4.1).

На основе таблицы 4.1 и с учетом свойства независимости входов и выходов можно найти функции алгебры логики выходов матриц M_1 и M_2 .

Функции на выходе матрицы M_1 :

$$F_1 = \bar{x}_1 \bar{x}_2 x_3 x_4 = x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4 = x_1 \downarrow x_2 \downarrow \bar{x}_3 \downarrow \bar{x}_4; \quad F_2 = x_1 \downarrow \bar{x}_2 \downarrow x_3 \downarrow x_4;$$

$$F_3 = x_1 \downarrow \bar{x}_2 \downarrow x_3 \downarrow \bar{x}_4; \quad F_4 = x_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3 \downarrow x_4; \quad F_5 = x_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3 \downarrow \bar{x}_4;$$

$$F_6 = \bar{x}_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3 \downarrow x_4; \quad F_7 = \bar{x}_1 \downarrow \bar{x}_2 \downarrow \bar{x}_3 \downarrow \bar{x}_4.$$

Т а б л и ц а 4.1 – ТИ преобразователя кодов на ПЛМ

Десятичный эквивалент	Входной код 2421				Выходной код 8421				Выходы M_1
	x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	F_i
3	0	0	1	1	0	0	1	1	F_1
4	0	1	0	0	0	1	0	0	F_2
5	0	1	0	1	0	1	0	1	F_3
6	0	1	1	0	0	1	1	0	F_4
7	0	1	1	1	0	1	1	1	F_5
8	1	1	1	0	1	0	0	0	F_6
9	1	1	1	1	1	0	0	1	F_7

На выходе матрицы M_2 : $\bar{y}_1 = \bar{F}_6 \vee \bar{F}_7 = F_6 \downarrow F_7; \quad \bar{y}_2 = F_2 \downarrow F_3 \downarrow F_4 \downarrow F_5;$
 $\bar{y}_3 = F_1 \downarrow F_4 \downarrow F_5; \quad \bar{y}_4 = F_1 \downarrow F_3 \downarrow F_5 \downarrow F_7.$

Как видно, получаются инверсные выходные значения и поэтому их следует инвертировать при окончательном построении схемы.

В соответствии с полученными уравнениями осуществляется программирование ПЛМ. Узлы матрицы, в которых необходимо сохранить

транзисторы, отмечены на рисунке 4.5 точками.

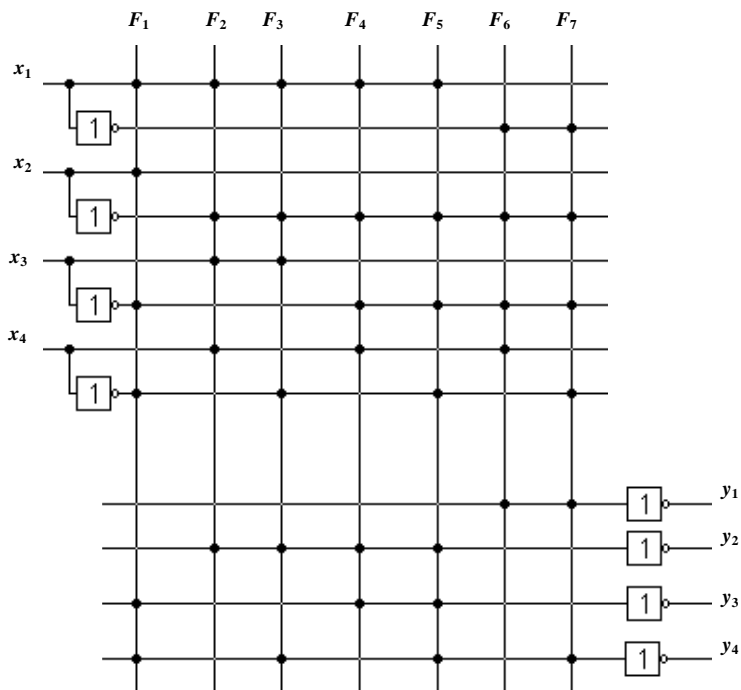


Рисунок 4.5 – Точечное начертание матрицы преобразователя кодов

В местах пересечения линий F_i, x_i и y_i ставятся транзисторы аналогично тому, как это изображено на рисунке 4.3. В результате, после добавления резисторов получается рабочая схема преобразователя кодов.

2 Порядок выполнения работы

1 Изучить теоретический материал по учебнику и данным методическим указаниям.

2 В соответствии с приведенным алгоритмом синтеза и вариантом задания, указанным в таблице 4.2, построить комбинационное дискретное устройство, пользуясь программой «Электронная лаборатория».

3 Проверить правильность его функционирования в соответствии с таблицей истинности.

4 Сделать выводы по работе.

Содержание отчета

Название и цель работы, краткие сведения из теории, функции на выходах матриц $M1$ и $M2$, точечное начертание матрицы заданного устройства, принципиальная схема реализованного на ПЛИМ устройства, письменные ответы на заданные преподавателем контрольные вопросы, полученные выводы о соответствии работы устройства его таблице истинности.

Т а б л и ц а 4.2 – Варианты заданий

Вариант	Тип устройства для цифр в десятичном эквиваленте	Код	
		на входе	на выходе
1	ПК от 0 до 5	8421	2 из 5
2	Ш от 3 до 7	Десятичный	с избытком 3
3	ДШ от 3 до 9	8421	Десятичный
4	ПК от 2 до 9	7421	3а+2
5	Ш от 0 до 6	Десятичный	7421
6	ДШ от 2 до 8	7421	Десятичный
7	ПК от 1 до 7	2421	с избытком 3
8	Ш от 0 до 5	Десятичный	2421
9	ДШ от 0 до 5	2421	Десятичный
10	ПК от 2 до 7	2 из 5	8421
11	Ш от 2 до 9	Десятичный	2 из 5
12	ДШ от 2 до 9	2 из 5	Десятичный
13	ПК от 4 до 9	3а+2	7421
14	Ш от 3 до 9	Десятичный	3а+2
15	ДШ от 1 до 8	3а+2	Десятичный
16	ПК от 1 до 8	2 из 5	3а+2
17	Ш от 1 до 8	Десятичный	2421
18	ДШ от 1 до 6	2421	Десятичный
19	ПК от 1 до 6	3а+2	8421
20	Ш от 1 до 6	Десятичный	3а+2
21	ДШ от 0 до 6	3а+2	Десятичный
22	ПК от 0 до 6	2421	с избытком 3
23	Ш от 4 до 9	Десятичный	с избытком 3
24	ДШ от 0 до 4	с избытком 3	Десятичный
25	ПК от 0 до 4	2 из 5	3а+2
26	Ш от 0 до 4	Десятичный	2 из 5
27	ДШ от 0 до 7	2 из 5	Десятичный
28	ПК от 0 до 7	с избытком 3	7421
29	Ш от 0 до 7	Десятичный	2421
30	ДШ от 0 до 4	8421	Десятичный

Контрольные вопросы

1 В чем заключается основное отличие ПЛИМ от других комбинационных

устройств?

2 Каково различие между мультиплексором и ПЛМ?

3 Где преимущественно применяются ПЛМ?

4 Каково назначение матрицы $M1$ в ПЛМ?

5 Для чего используется матрица $M2$?

6 В чем состоят недостатки ПЛМ?

7 Какие преимущества у ПЛМ перед обычными дискретными устройствами?

8 В каких случаях требуется преобразование функций алгебры логики при синтезе схем на ПЛМ?

9 Каким образом осуществляется расширение количества входов ПЛМ?

Рекомендуемая литература

1 Букреев И. Н. и др. Микроэлектронные схемы цифровых устройств. М.: Советское радио, 1975. 264 с.

2 Алексеенко А. Г., Шагури И. И. Микросхемотехника. М.: Радио и связь, 1982. 414 с.

3 Автоматизация систем энергоснабжения: Учебник для вузов ж.-д. трансп. /Ю. И. Жарков, В. Я. Овласюк, Н. Г. Сергеев, Н. Д. Сухопрудский, А. С. Шилов; Под ред. Н. Д. Сухопрудского. М.: Транспорт, 1990. 359 с.

4 Сапожников В. В. и др. Дискретные устройства железнодорожной автоматики, телемеханики и связи. Учебник для вузов ж.-д. трансп. М.: Транспорт, 1988. 255 с.

5 Поспелов Д. А. Логические методы анализа и синтеза схем. М.: Энергия, 1974. 368 с.

О Г Л А В Л Е Н И Е

Введение	3
<i>Лабораторная работа № 1.</i> Изучение триггеров	3
<i>Лабораторная работа № 2.</i> Изучение сумматоров	16
<i>Лабораторная работа № 3.</i> Синтез функций алгебры логики на базе мультиплексоров.....	22
<i>Лабораторная работа № 4.</i> Синтез комбинационных устройств на программируемых логических матрицах.....	27
Рекомендуемая литература.....	34

Учебное издание

Константин Афанасьевич БОЧКОВ
Юрий Федорович БЕРЕЗНЯЦКИЙ

Анализ функциональной структуры и синтез дискретных устройств

Лабораторный практикум по дисциплине «Теория дискретных устройств»

Часть II

Редактор Т. М. Р и з е в с к а я
Технический редактор В. Н. К у ч е р о в а
Корректоры Н. А. Д а ш к е в и ч, О. В. З а н и н а

Подписано в печать 10.01.2001 г. Формат бумаги 60x84_{1/16}.
Бумага газетная. Гарнитура Таймс. Печать офсетная.
Усл. печ. л. 2,09. Уч.-изд. л. 2,02. Тираж 200 экз.
Зак. № 2365. Изд. № 3581

Редакционно-издательский отдел БелГУТа, 246653, г.Гомель, ул. Кирова, 34.
Лицензия ЛВ № 57 от 22.10.97 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.
Лицензия ЛП № 360 от 26.07.99 г.