

Спиридонов С.Б.

"Схемотехника дискретных устройств"

Учебно-методическое пособие

2020 г.

УДК
БКК

Рассмотрены основные понятия базовых логических элементов входящих в основные узлы и блоки вычислительных устройств широкого класса. Изложены принципы описания функционирования, методологии проектирования узлов и блоков на основе логических элементов. Рассмотрен раздел булевой алгебры, нашедший применение на этапах синтеза схемотехнических решений узлов и устройств. Рассмотрены основные методы проектирования и оптимизации решений.

Для студентов вузов, обучающихся по направлению подготовки «Информатика и вычислительная техника» специальности «Автоматизированные системы обработки информации и управления».

Оглавление

Предисловие.....	6
Задачи и результаты изучения теоретического материала.....	9
Модуль 1. Базовые элементы дискретных устройств АСОИУ.....	12
Глава 1. Логические элементы.....	12
1.1 Базовые схемы реализации логических функций.....	12
1.2. Типовые примеры проектных решений.....	19
1.3. Контрольные вопросы.....	20
1.4. Задачи для самостоятельного решения.....	22
Глава 2. Применение булевой алгебры в проектировании комбинационных схем.....	26
2.1. Основные понятия и определения булевой алгебры.....	26
2.2 Типовые примеры проектных решений.....	41
2.3. Контрольные вопросы.....	41
2.4. Задачи для самостоятельного решения.....	42
Глава 3. Математический аппарат проектирования комбинационных схем дискретных устройств.....	44
3.1 Синтез комбинационных схем.....	44
3.2. Типовые примеры проектных решений.....	53
3.3 Контрольные вопросы.....	59
3.4. Задачи для самостоятельного решения.....	60
Глава 4. Схемотехника коммутирующих и кодирующих устройств.....	65
4.1. Мультиплексоры, демультиплексоры, дешифраторы, шифраторы, преобразователи кодов.....	65
4.2. Типовые примеры проектных решений.....	80
4.3. Контрольные вопросы.....	84
4.4. Задачи для самостоятельного решения.....	85
Глава 5. Триггеры	89
5.1. Назначение, классификация и типовое устройство триггеров.....	89
5.2. Триггеры RS, D, JK, T типа.....	94

5.3. Применение триггеров в схемах управления и переключения.....	113
5.4. Типовые примеры проектных решений с использованием триггеров.....	116
5.5. Контрольные вопросы.....	118
5.6. Задачи для самостоятельного решения.....	121
Глава 6. Задания и вопросы для самостоятельной подготовки к рубежному контролю по модулю 1.....	125
Модуль 2. Базовые узлы и блоки дискретных устройств АСОИУ.....	137
Глава 7. Регистры.....	137
7.1. Назначение, классификация, параметры и типовое устройство регистров.....	137
7.2. Регистр с приёмом информации параллельным кодом.....	138
7.3. Сдвигающие регистры.....	140
7.4. Реверсивные сдвигающие регистры.....	141
7.5. Универсальный регистр.....	142
7.6. Применение регистров.....	143
7.7. Типовые примеры проектных решений с использованием регистров.....	144
7.8. Контрольные вопросы.....	145
7.9. Задачи для самостоятельного решения.....	145
Глава 8. Счетчики и делители частоты.....	146
8.1. Назначение, классификация, параметры и типовое устройство счётчиков.....	146
8.2. Синтез счетчиков и пересчетных схем на универсальных D – триггерах и JK – триггерах.....	147
Итоги изучения главы 8.	
8.2. Типовые примеры проектных решений.....	155
8.3. Контрольные вопросы.....	156
8.4. Задачи для самостоятельного решения.....	156
Глава 9. Арифметические устройства.....	157
9.1. Назначение и классификация арифметических устройств.....	157

9.2. Классификация и основные типы вычитателей.....	166
9.3. Сумматоры накапливающего типа.....	169
9.4. Двоично-десятичный сумматор.....	172
9.5 Универсальный многоразрядный сумматор-вычитатель.....	173
9.6. Сумматоры дополнительного кода и сумматоры обратного кода.....	175
9.7. Методы ускорения распространения переносов в сумматорах.....	177
9.8. Инкременторы и декременторы.....	178
9.9. Цифровые компараторы.....	180
Итоги изучения главы 9.....	184
9.10. Типовые примеры проектных решений арифметических устройств..	185
9.11 Контрольные вопросы.....	185
9.12. Задачи для самостоятельного решения.....	186
Глава 10. Запоминающие устройства.....	187
10.1 Назначение и классификация запоминающих устройств.....	187
10.2. Постоянные и оперативные запоминающие устройства.....	187
Итоги изучения главы 10.....	201
10.4. Контрольные вопросы.....	202
Глава 11. Задания и вопросы для самостоятельной подготовки к рубежному контролю по модулю 2.....	203
Глоссарий. Список сокращения и обозначений.....	205
Литература.....	206

Предисловие

Бурное развитие интегрированных схемотехнических решений во всех областях использования вычислительной техники и системах управления, требует потребности в базовых знаниях основ проектирования базовых элементов и устройств. Большинство структурных и схемотехнических решений, применяемых в компьютерах, микропроцессорных системах, микроконтроллерах и т.д. являются универсальными и типовыми.

Схемотехника дискретных устройств – учебная дисциплина, в которой рассматривают схемотехнический базис устройств, лежащих в основе проектных и конструкторских решений при создании средств вычислительной техники, математический аппарат булевой алгебры, применяемый для анализа, проектирования и оптимизации архитектуры схемотехнических решений.

Учебное пособие охватывает часть, соответствующую первому модулю «Комбинационные схемы и устройства» программы курса по дисциплине «Схемотехника дискретных устройств».

В настоящем пособии изложены основные понятия схемотехнического базиса устройств вычислительной техники и математический аппарат, лежащий в основе проектирования широкого класса элементов и узлов вычислительных устройств.

Опыт преподавания студентам основ проектирования показал, что теоретический материал прочно и быстро усваивается студентами только при практическом выполнении задач. Часть этой проблемы решается в ходе лабораторного практикума и при практическом выполнении домашнего задания. Помимо этого в данное учебное пособие внесены разделы для самостоятельного контроля знаний студентов. Изложенные в пособии тестовые примеры и задачи служат целью подготовить студента к контрольным мероприятиям по данному курсу. В приведённых типовых

примерах обосновывается использование рассмотренных теоретических методов и подходов.

Пособие состоит из пяти глав и приложения.

В первой главе рассмотрены основные схемы самого нижнего базового уровня, на котором создаются самые основные узлы и блоки вычислительных устройств. Даны основные вводные понятия о логических элементах и зависимостях выходных сигналов от множества входных сигналов. Обосновывается тождественность понятий двоичной арифметики, булевой алгебры и теории переключательных схем.

Во второй главе приведено описание той части понятий, теорем, законов булевой алгебры, которая применима для задач схемотехнического проектирования устройств вычислительной техники.

Рассмотрены булевы функции, которые имеют однозначное соответствие функционированию типовых элементов, реализованных на основе полупроводниковых приборов. Рассмотрено применение законов преобразования булевых функций и показаны практические примеры альтернативных вариантов решения одного и того же узла на различной элементной базе. В пособии значительное место уделено одному из удобных методов минимизации булевых функций, имеющих практическую цель сокращения размерности разрабатываемых узлов и устройств, при сохранении их функциональных возможностей. На основе применения этого метода минимизации сформулировано большое количество примеров и задач, приведённых в последующих разделах.

В третьей главе изложена технология синтеза, то есть определение структуры и архитектуры схем на основе описания их работы средствами булевой алгебры. Подробно рассмотрены этапы этого проектного подхода. Рассмотрены примеры практической направленности. Показано, что синтез комбинационных схем лежит в основе проектирования более сложных по функционированию устройств вычислительной техники.

В четвёртой главе рассмотрена архитектура и схемная реализация дешифраторов, шифраторов, мультиплексоров и демультимплексоров. Приведены примеры их применения и функционального назначения.

В пятой главе рассмотрена классификация триггеров, являющихся основой схемотехнической базы подавляющего большинства устройств вычислительной техники. Рассматриваются триггеры RS, D, JK и E - типа. Рассматриваются варианты схемотехнической архитектуры триггеров. Приводятся примеры использования триггеров в схемах управления и селекции сигналов. Приводятся контрольные вопросы и задачи для самостоятельного решения.

В шестой главе приводятся примеры практического выполнения тех теоретических основ и принципов, которые рассмотрены в предыдущих главах.

Далее приведены контрольные вопросы для самоконтроля усвоения теоретического материала данного курса, сформулированы тестовые задания и задачи для самостоятельного решения, а также приведены ответы и подробные решения предложенных задач.

Самостоятельное выполнение тестовых заданий и задач даст студенту возможность успешно подготовиться к выполнению контрольных мероприятий по дисциплине.

Пособие предназначено для студентов вузов, обучающихся по направлению подготовки 230100 «Информатика и вычислительная техника» специальности «Автоматизированные системы обработки информации и управления». Отдельные главы учебного пособия будут полезны слушателям второго высшего образования и кругу специалистов, занимающихся проектированием специализированных вычислительных средств автоматизации и управления.

Автор выражает благодарность рецензентам пособия за замечания и полезные рекомендации, сделанные в процессе детального изучения учебного пособия.

Задачи и результаты изучения теоретического материала модулей 1 и 2.

1. Цель преподавания дисциплины состоит в содействии формированию у студента базовых знаний по составу, назначению и принципам работы основных электронных и логических элементов и узлов современных компьютеров, ЭВМ и вычислительных систем. .

2. Задачами преподавания дисциплины являются:

- изучение теоретических основ и используемых математических методов при построении архитектуры современных компьютеров, ЭВМ и вычислительных систем,

- изучение базовых электронных компонентов, основных логических элементов, типовых узлов современных компьютеров, ЭВМ и вычислительных систем,

- изучение программных средств, позволяющих проводить моделирование функционирования и исследования различных характеристик электронных компонентов, логических элементов, комбинационных схем и типовых узлов современных компьютеров, ЭВМ и вычислительных систем.

3. Изучение дисциплины предполагает предварительное освоение следующих дисциплин учебного плана:

1. Дискретная математика.
2. Электротехника.
3. Электроника.
4. Информатика.

После освоения дисциплины студент должен приобрести следующие знания, умения и владения соответствующие компетенциям по СУОС:

Согласно СОК-12:

1. Знать и понимать назначение и применение логических элементов в архитектуре вычислительных устройств.
2. Уметь проводить оценку возможности совместного применения различных логических элементов для создания комбинационных схем.

Согласно СОПК-2:

1. Знать и понимать возможности программ моделирования электронных и логических схем с целью проверки работоспособности спроектированной комбинационной схемы или узла вычислительного устройства.

2. Продемонстрировать знания пользовательского интерфейса программ моделирования.

3. Уметь создавать моделируемые схемы в среде программ моделирования.

4. Владеть способом отображения элементов электронной схемы библиотечными компонентами программ моделирования и настройки их параметров.

Согласно СОПК – 7:

1. Помнить основные теоремы и законы алгебры логики.

Понимать различие поведения булевых функций.

2. Продемонстрировать знания по минимизации булевых выражений.

3. Уметь применять аппарат булевых функций для описания поведения электронных комбинационных схем, анализировать возможность преобразования схем из одного элементного базиса в другой. Создавать оптимальные по структуре комбинационные схемы.

4. Владеть методами и способами создания комбинационных схем по формализованному в терминах булевых выражений закону их работы.

В учебном пособии в завершении материала глав применены четыре категории оценки знаний, получаемых при изучении данной учебной дисциплины согласно уточненной таксономии Блума:

Категория 1. Помнить – Извлекать необходимую информацию из памяти.

Категория 2. Понимать – Создавать значения на базе учебных материалов или опыта.

Категория 3. Применять – использовать процедуру действий.

Категория 4. Анализировать и оценивать – сравнивать, делать суждения, основанные на критериях и стандартах.

Категория 5. Создавать – проектировать и разрабатывать функциональные узлы АСОИУ на основе базовых элементов.

В дальнейшем материале по данным категориям сгруппированы контрольные вопросы.

Модуль 1. Базовые элементы дискретных устройств АСОИУ

Глава 1. Логические элементы.

1.1. Базовые схемы реализации логических функций.

Понятие логического элемента. Логические операции над двоичными переменными реализуются схемами, которые называются логическими элементами. Число входов логического элемента соответствует числу аргументов воспроизводимой им булевой функции.

В цифровой вычислительной технике (ЦВТ) вся информация, необходимая для вычислительного процесса, представляется в виде набора дискретных сигналов. Каждый из сигналов может принимать одно из двух возможных значений, обозначаемых «1» и «0». Символ «1» обозначает наличие сигнала, «0» – его отсутствие.

В схемах цифровых вычислительных устройств переменные и соответствующие им сигналы изменяются и воспринимаются не непрерывно, а лишь в дискретные моменты времени, обозначаемые целыми положительными числами.

$$t_i = 0, 1, \dots, i, \dots, n \quad (1.1)$$

При потенциальном способе представления информации при положительной логике двум значениям переменной “1” и “0” соответствует высокий и низкий уровни напряжения. Потенциальный сигнал сохраняет

постоянный уровень (нулевой или единичный) в течение периода представления информации (такта).

Понятие о комбинационной схеме и цифровом автомате.

В ЦВТ преобразование информации производится электронными устройствами двух классов: комбинационными устройствами (схемами) и последовательностными устройствами (цифровыми автоматами или автоматами с памятью).

В комбинационных схемах (КС), называемых также автоматами без памяти, совокупность выходных сигналов (выходное слово Y) в дискретный момент времени t_i однозначно определяется входными сигналами (входным словом X), поступившим на входы в тот же дискретный момент времени.

Реализуемый в этих схемах способ обработки информации называется комбинационным, т.к. результат обработки информации зависит от комбинации входных сигналов и вырабатывается сразу после подачи на входы входной информации.

Закон функционирования КС определен, если задано соответствие между входными словами и её выходными словами в табличной или аналитической форме.

$$Y_i = f_i(x_1, x_2, \dots, x_n) \quad (1.2)$$

В алгебре логики (булевой алгебре) обычно все X_i и Y_i могут принимать только два значения: 0 и 1. В этом случае функции $f_1 \dots f_m$ называются функциями алгебры логики (булевыми или двоичными функциями).

Другой, более сложный, класс преобразователей цифровой информации составляют цифровые автоматы. Цифровой автомат, в отличие от логической схемы, имеет некоторое конечное число различных внутренних состояний.

$$Q = \{q_0, q_1, \dots, q_k\} \quad (1.3)$$

Под воздействием входного слова цифровой автомат переходит из одного состояния в другое и выдает выходное слово. Выходное слово на выходе цифрового автомата в дискретный момент времени определяется входным словом, поступившим в этот момент времени на вход автомата, и

внутренним состоянием автомата, которое явилось результатом воздействия на автомат входных слов в предыдущие моменты времени.

Цифровой автомат обязательно содержит память, состоящую из запоминающих элементов (триггеров, элементов задержки и др.), фиксирующих состояние, в котором он находится.

Комбинационная схема не содержит запоминающих элементов, поэтому её называют автоматом без памяти или “примитивным автоматом”.

Возможность применения алгебры логики к задачам проектирования цифровых устройств обусловлена аналогией понятий и категорий алгебры логики и двоичной системы счисления.

Основные логические элементы, воспроизводящие булевы функции двух переменных.

Логический элемент «И».

На рис. 1.1 приведено условно-графическое изображение логического элемента «И» на два входа, в табл.1.1 Представлено значение выходного сигнала Y на его выходе в зависимости от комбинации входных сигналов.

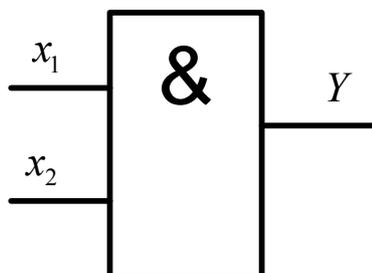


Рис. 1.1 Условно-графическое изображение логического элемента «И» на два входа.

Таблица 1.1

Таблица функции конъюнкции.

x_1	x_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Выражение булевой функции «конъюнкция» или логического умножения

$$Y = x_1 \wedge x_2 = x_1 x_2 \quad (1.4)$$

Логический элемент «ИЛИ».

На рис.1.2 приведено условно-графическое изображение логического элемента «ИЛИ» на два входа, в табл.1.2 представлено значение выходного сигнала Y на его выходе от комбинации входных сигналов.

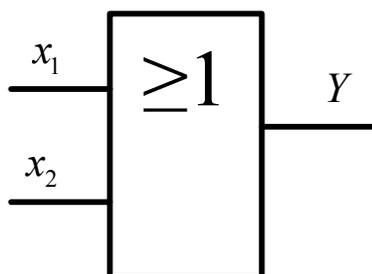


Рис. 1.2 Условно-графическое изображение логического элемента «ИЛИ» на два входа.

Таблица 1.2

Таблица функции дизъюнкции.

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Выражение булевой функции «дизъюнкция» или логического сложения

$$Y = x_1 \vee x_2 = x_1 + x_2 \quad (1.5)$$

Логический элемент «НЕ».

На рис. 1.3 приведено условно-графическое изображение логического элемента «НЕ», реализующего функцию отрицания (инверсии). В табл. 1.3 представлено значение выходного сигнала Y на его выходе в зависимости от входного сигнала.

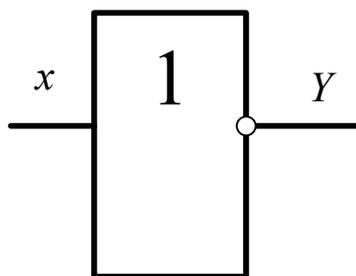


Рис.1.3 Условно-графическое изображение логического элемента «НЕ»

Таблица 1.3.

Таблица функции отрицания (инверсии).

x	Y
0	1
1	0

Выражение булевой функции «отрицание (инверсия)»

$$Y = \bar{x} \quad (1.6)$$

Логический элемент «И-НЕ».

На рис.1.4 приведено условно-графическое изображение логического элемента «И-НЕ» на два входа, реализующего булеву функцию «Штрих Шеффера», в табл. 1.4 представлено значение выходного сигнала Y на его выходе в зависимости от комбинации входных сигналов.

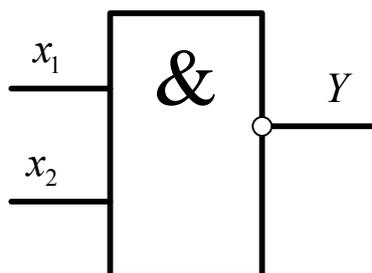


Рис. 1.4 Условно-графическое изображение логического элемента «И-НЕ» на два входа.

Таблица 1.4.

Таблица функции «И-НЕ»

x_1	x_2	Y
0	0	1
0	1	1
1	0	1
1	1	0

Выражение булевой функции «Штрих Шеффера» (И-НЕ).

$$Y = \overline{x_1 \wedge x_2} = \overline{x_1 x_2} \quad (1.7)$$

Логический элемент «ИЛИ-НЕ».

На рис.1.5 приведено условно-графическое изображение логического элемента «ИЛИ-НЕ» на два входа, реализующего булеву функцию «Стрелка Пирса», в табл. 1.5 представлено значение выходного сигнала Y на его выходе в зависимости от комбинации входных сигналов.

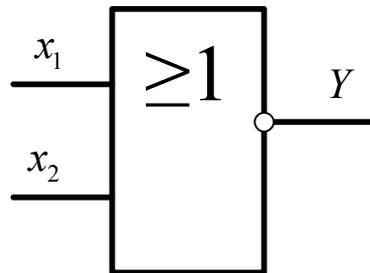


Рис. 1.5 Условно-графическое изображение логического элемента «ИЛИ-НЕ» на два входа.

Таблица 1.5.

Таблица функции «Стрелка Пирса» (ИЛИ-НЕ).

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Выражение булевой функции «Стрелка Пирса».

$$Y = \overline{x_1 \vee x_2} = \overline{x_1 + x_2} \quad (1.8)$$

Логический элемент «Исключающее ИЛИ».

На рис. 1.6 приведено условно-графическое изображение логического элемента «Исключающее ИЛИ» на два входа, в табл. 1.6 представлено значение выходного сигнала Y на его выходе в зависимости от комбинации входных сигналов.

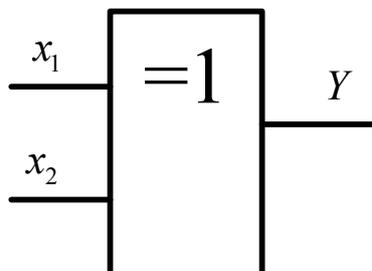


Рис. 1.6 Условно-графическое изображение логического элемента «Исключающее ИЛИ» на два входа.

Таблица 1.6.

Таблица функции «Исключающее ИЛИ».

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Выражение булевой функции «Исключающее ИЛИ».

$$Y = x_1 \oplus x_2 \quad (1.9)$$

Логический элемент «Эквивалентность или логическая равнозначность».

На рис. 1.7 приведено условно-графическое изображение логического элемента «Эквивалентность» на два входа, в табл. 1.7 представлено значение выходного сигнала Y на его выходе в зависимости от комбинации входных сигналов.

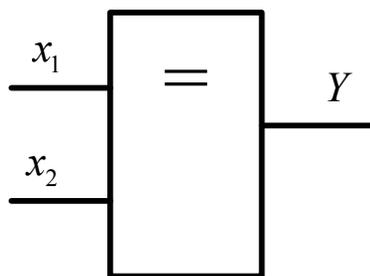


Рис. 1.7 Условно-графическое изображение логического элемента «Эквивалентность» на два входа.

Таблица 1.7.

Таблица функции «Эквивалентность (логическая равнозначность)».

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

Выражение булевой функции «Эквивалентность (логическая равнозначность)».

$$Y = x_1 \equiv x_2 \quad (1.10)$$

Приобретаемые компетенции после изучения главы 1.

Помнить:

- наименования логических элементов, их условно-графические отображения (УГО) и таблицы истинности функций, воспроизводимые логическими элементами.

Понимать:

- схемные особенности для воспроизводства булевых функций на примерах элементарных переключательных схем.

Применять:

- управление включением в переключательных схемах и схемах, моделируемых с помощью специализированных программ на компьютерах, согласно формулируемым правилам и дисциплинам включений и управления.

1.2. Типовые примеры проектных решений

1.2.1. Подобрать логический элемент для реализации функции включения лампы, соответствующей электрической схеме на рис. 1.8

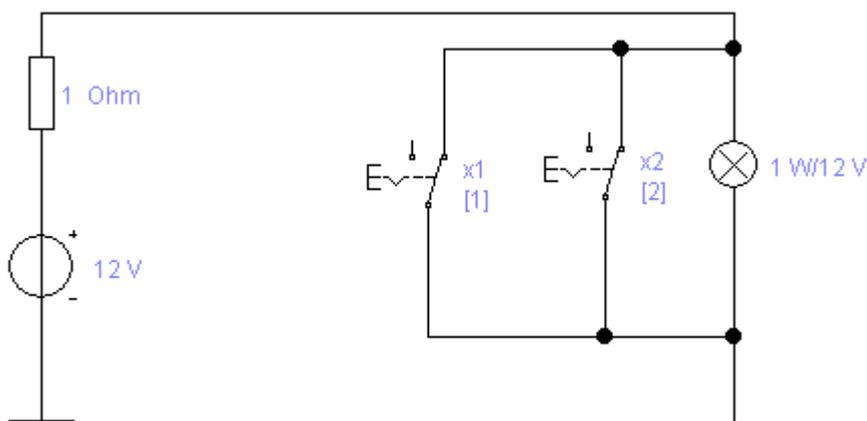


Рис. 1.8 Электрическая схема включения лампы, управляемая двумя выключателями.

Составим таблицу горения лампы (табл. 1.8) от комбинации состояния выключателей.

За разомкнутое состояние переключателя примем значение $x_i = 0$, за замкнутое $x_i = 1$. Лампа будет загораться только при двух одновременно разомкнутых переключателях. Обозначим переменной $Y = 0$ состояние, когда лампочка не горит, а $Y = 1$ состояние горящей лампочки.

Таблица 1.8

Таблица включения лампочки.

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Данная таблица совпадает с булевой функцией «стрелка Пирса» и может быть реализована логическим элементом «И-НЕ».

Схема включения индикаторного светодиода (эквивалента лампочки) может быть смоделирована эквивалентной рисунку 1.8 моделируемой схеме в пакете Electronics Workbench в следующем виде рис. 1.9

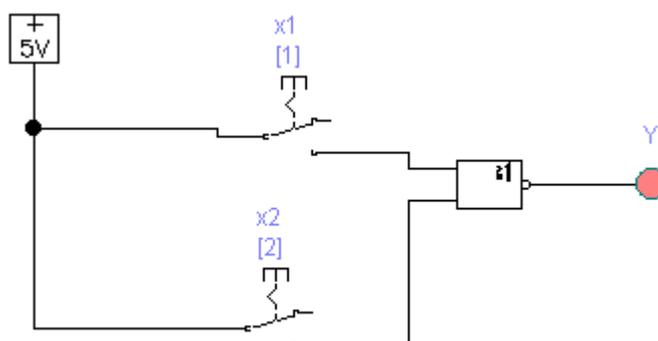


Рис. 1.9. Эквивалентная схема включения светодиода, совпадающая с логикой включения лампочки в схеме на рис. 1.8

1.3. Контрольные вопросы.

Вопросы категории 1. «Помнить».

1.3.1 Запишите выражение функции конъюнкции и приведите её таблицу истинности для трёх переменных.

1.3.2 Запишите выражение функции дизъюнкции и приведите её таблицу истинности для трёх переменных.

Вопросы категории 2. «Понимать».

1.3.3 На рис. 1.10. представлена диаграмма подачи входных сигналов x_1, x_2 на неизвестный логический элемент и диаграмма поведения выходного

сигнала. Какую логическую функцию воспроизводит данный логический элемент?

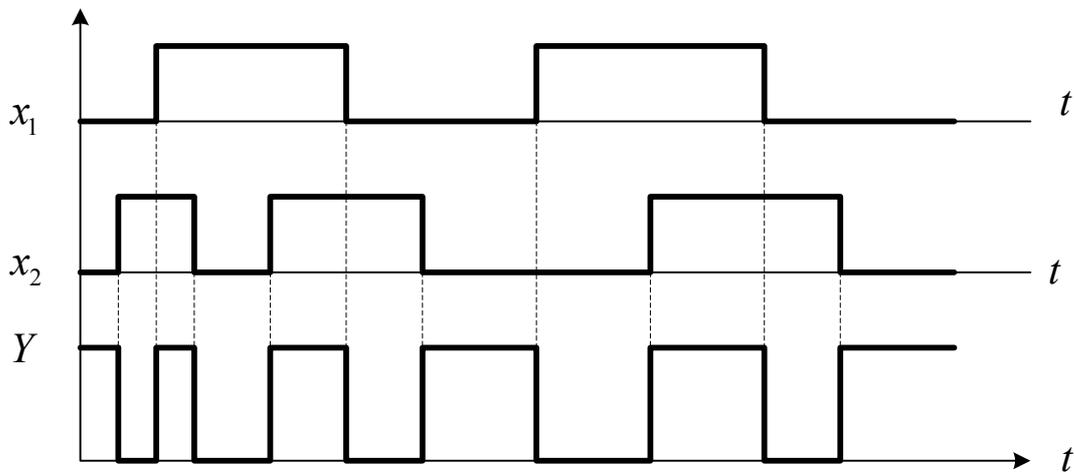


Рис. 1.10 Диаграмма подачи входных сигналов на неизвестный логический элемент.

Вопросы категории 3. «Применять».

1.3.4 Постройте таблицу истинности для логического элемента ИЛИ с тремя входами. Выходной сигнал обозначим символом Y , входные сигналы соответственно x_1, x_2, x_3 .

Ответы на контрольные вопросы.

1.3.1 Таблица истинности для трёх переменных примет вид:

Таблица 1.9.

Таблица истинности для логического элемента «И» с тремя входами.

x_1	x_2	x_3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

1.3.3 Для решения вопроса целесообразно результаты в графическом варианте представить в виде таблицы истинности табл. 1.10.

Таблица 1.10

Таблица истинности поведения неизвестного логического элемента.

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Таблица истинности 1.10. соответствует логическому элементу «ИЛИ-НЕ».

1.3.4 Таблица истинности для трёх переменных примет вид:

Таблица 1.11

Таблица истинности для логического элемента «ИЛИ» с тремя входами.

x_1	x_2	x_3	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

1.4. Задачи для самостоятельного решения.

1.4.1 Записать таблицу истинности для электрической схемы рис. 1.11

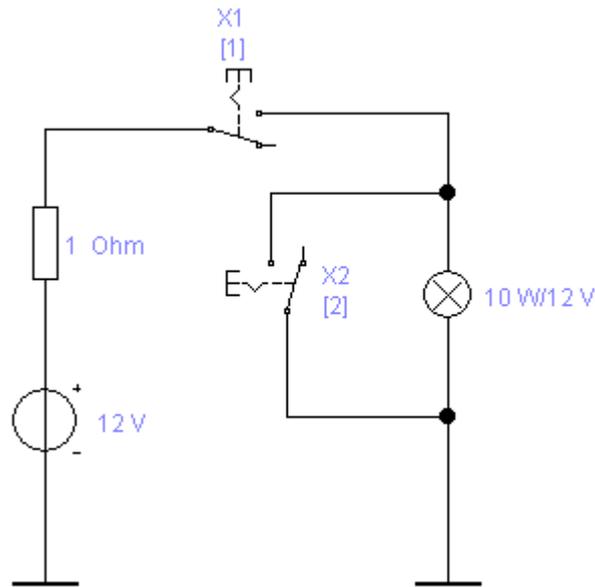


Рис. 1.11 Электрическая схема с двумя ключами.

Собрать комбинационную схему, функционирующую по полученной таблице истинности.

Решение:

Замкнутое состояние ключа X1 приведёт к протеканию тока через лампочку, если ключ X2 разомкнут. При замыкании ключа X2 лампочка окажется замкнутой накоротко и гореть не будет. При отключении ключа X1 лампочка гореть не будет и при замкнутом и при разомкнутом состоянии ключа X2. Таблица истинности получается следующей (табл.1.12).

Таблица 1.12

Таблица истинности для схемы на рис.1.11.

x_1	x_2	Y
0	0	0
0	1	0
1	0	1
1	1	0

Горение лампочки обозначим $Y = 1$, не горящая лампочка соответствует функции $Y = 0$.

По таблице истинности записывается выражение для функции Y (горение лампочки).

$$Y = x_1 * \overline{x_2} \quad (1.11)$$

Этой функции соответствует следующая комбинационная схема, реализующая булеву функцию «запрет по x_2 » (рис. 1.12.)

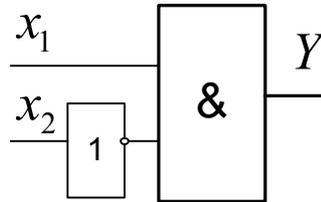


Рис. 1.12. Комбинационная схема, реализующая функцию «запрет по x_2 »

1.4.2. Сформируйте электрическую схему, подобную рис. 1.11, функционирование которой соответствует булевой функции «стрелка Пирса» («ИЛИ-НЕ»).

Решение.

Функции «стрелка Пирса» («ИЛИ-НЕ») соответствует следующая таблица истинности табл. 1.13

Таблица. 1.13

Таблица истинности функции для задания 1.4.2.

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

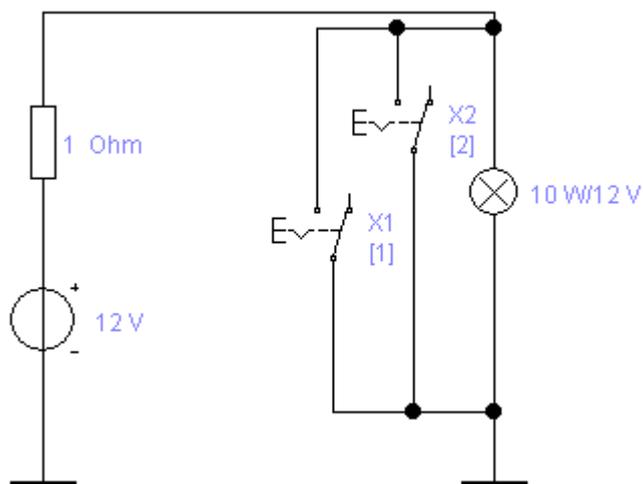


Рис. 1.13 Электрическая схема, соответствующая таблице истинности табл.1.13

Горение лампочки возможно только в случае двух разомкнутых ключей. Во всех остальных комбинациях хотя бы один ключ замыкает накоротко лампочку.

1.4.3. Сформируйте электрическую схему, подобную рис.1.11, функционирование которой соответствует булевой функции «штрих Шеффера» («И-НЕ»).

Решение:

Функции «штрих Шеффера» («И-НЕ») соответствует следующая таблица истинности табл. 1.13

Таблица. 1.14.

Таблица истинности функции «штрих Шеффера» («И-НЕ»).

x_1	x_2	Y
0	0	1
0	1	1
1	0	1
1	1	0

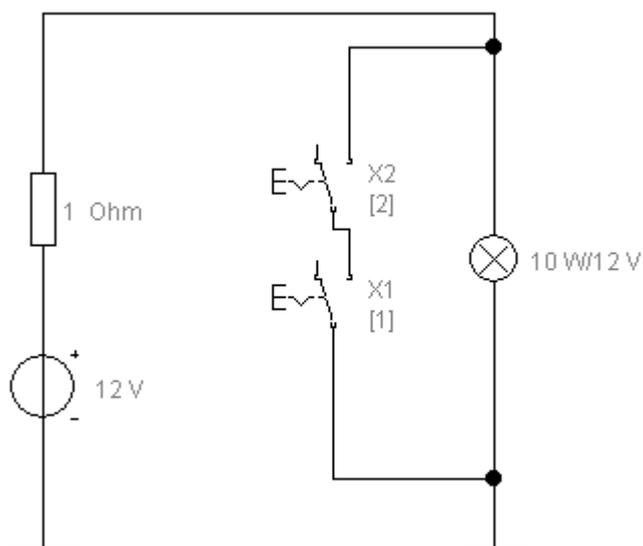


Рис. 1.14. Электрическая схема, соответствующая таблице истинности табл. 1.14.

Горение лампочки возможно только в случае двух разомкнутых ключей, либо одного из двух. Два одновременно замкнутых ключа замыкают одной цепью лампочку накоротко, лампочка не горит.

Глава 2. Применение булевой алгебры в проектировании комбинационных схем.

2.1 Основные понятия и определения булевой алгебры логики.

Логика- наука о формах и законах мышления (в общем понимании).

Математическая логика- наука о применении математических методов для решения различного рода логических задач.

Основное понятие алгебры логики - *высказывание*.

Простое высказывание - некоторое предложение, о котором можно утверждать, что оно истинно или ложно.

Сложным высказыванием является предложение, состоящее из нескольких простых предложений (т.е. простых высказываний), связанных между собой какими либо логическими связями.

Под логическими связями понимаются грамматические союзы типа «НЕ», «И», «ИЛИ», «ЕСЛИ, ТО».

Любое высказывание можно обозначить символом x и считать, что $x = 1$, если высказывание истинно, а $x = 0$, если высказывание ложно.

Логическая (булева) переменная – такая величина x , которая может принимать только два значения: $x = \{0, 1\}$.

Высказывание абсолютно истинно, если соответствующая ему логическая величина принимает значение $x = 1$ при любых условиях.

Высказывание абсолютно ложно, если соответствующая ему логическая величина принимает значение $x = 0$ при любых условиях.

Определение булевой функции:

Под булевой функцией (БФ) понимают сложное высказывание.

Эта функция принимает лишь два значения: 0 или 1. Булева функция всегда конечна.

Простые высказывания, входящие в булеву функцию, называют переменными (или булевыми переменными).

Булева (двоичная) функция – это двоичная переменная Y , значение которой зависит от её двоичных переменных (аргументов функции).

Чтобы задать булеву функцию надо каждому из возможных сочетаний аргументов x_1, x_2, \dots, x_n поставить в соответствие 0 или 1 (т.е. значение функции).

Количество возможных булевых функций N при количестве переменных p , определяется по формуле:

$$N = 2^{2^p} \quad (2.1)$$

БФ одной переменной называется симвилярной функцией (табл. 2.1)

Симвилярная БФ.

Таблица 2.1

Симвилярные булевы функции.

функция	$x = 0$	$x = 1$	обозначение	Название
$Y_1 = 0$	0	0	0	Константа нуль
$Y_2 = x$	0	1	x	Повторение
$Y_3 = \bar{x}$	1	0	\bar{x}	Отрицание (инверсия)
$Y_4 = 1$	1	1	1	Константа единица

Бинарная булева функция.

Булева функция от двух переменных называется бинарной. (Табл.2.2)

Таблица 2.2.

Бинарные булевы функции

x_1	x_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0

x_1	x_2	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1

Наименования бинарных булевых функций:

Y_0 - константа 0

$$Y_0 = 0$$

Y_1 – функция Пирса

$$Y_1 = \overline{x_1 + x_2}$$

Y_2 – запрет по x_1

$$Y_2 = \bar{x}_1 * x_2$$

Y_3 – переменная не x_1

$$Y_3 = \bar{x}_1$$

Y_4 – запрет по x_2

$$Y_4 = x_1 * \bar{x}_2$$

Y_5 – переменная

$$Y_5 = \bar{x}_2$$

Y6 – искл. ИЛИ (сложение по модулю 2)	
Y7 – функция Шеффера	$Y_7 = \overline{(x_1 * x_2)}$
Y8 – конъюнкция	$Y_8 = x_1 * x_2$
Y9 – эквивалентность (логическая равнозначность)	$Y_9 = x_1 \equiv x_2$
Y10- перемен. x2	$Y_{10} = x_2$
Y11- импликация $x_1 \rightarrow x_2$	$Y_{11} = \overline{x_1} + x_2$
Y12 – переменная x1	$Y_{12} = x_1$
Y13 – импликация $x_2 \rightarrow x_1$	$Y_{13} = x_1 + \overline{x_2}$
Y14 – дизъюнкция	$Y_{14} = x_1 + x_2$
Y15 – константа единицы	$Y_{15} = 1$

Определение логической функции.

Логическая функция алгебры логики – функция $f(x_1, x_2, \dots, x_n)$, принимающая значение, равное 0 или 1 на наборе логических переменных x_1, x_2, \dots, x_n .

Возможность применения алгебры логики к задачам проектирования вычислительных устройств обусловлена аналогией понятий и категорий алгебры логики и двоичной системы счисления.

Техническим аналогом булевой функции является комбинационная схема, выполняющая соответствующее этой функции преобразование информации.

Постоянные уровни напряжения, соответствующие принятому в схеме представлению 0 и 1, могут рассматриваться как технические аналоги функции «ложь» и «истина».

Множество элементов, которые рассматриваются в алгебре логики равно 2. Эти элементы получили название двоичных переменных. Для них в алгебре логики определены:

– отношение эквивалентности, обозначаемое символом равенства “=”,

– три операции:

1) операция логического сложения (дизъюнкции), обозначаемая символом “ \vee ” или “+”,

2) операция логического умножения (конъюнкции), обозначаемая символом “ \wedge ” или “&” или “.”,

3) операция логического отрицания (инверсии), обозначаемая черточкой над двоичной переменной “ \bar{X} ”.

В качестве постулатов или аксиом принимается, что при выполнении перечисленных операций отношения эквивалентности имеют следующий вид: а) логическое сложение, б) логическое умножение, в) инверсия

$$\text{а) } 0+0=0 \quad \text{б) } 0*0=0 \quad \text{в) } \bar{0} = 1$$

$$0+1=1 \quad 0*1=0 \quad \bar{1} = 0$$

$$1+0=1 \quad 1*0=0$$

$$1+1=1 \quad 1*1=1$$

Возможна и другая система постулатов. На основании постулатов выводятся соотношения или законы алгебры логики для двоичных переменных.

Законы и аксиомы алгебры логики.

Закон одинарных элементов:

$$x + 1 = 1; x * 1 = x$$

$$x + 0 = x; x * 0 = 0$$

Законы отрицания:

- закон двойного отрицания

$$\overline{\bar{x}} = x$$

- закон дополнительности

$$x + \bar{x} = 1 \quad x * \bar{x} = 0$$

Закон двойственности Де Моргана

$$\overline{x_1 + x_2} = \bar{x}_1 * \bar{x}_2 \quad \overline{x_1 * x_2} = \bar{x}_1 + \bar{x}_2$$

Из этих выражений следует следствие:

$$x_1 + x_2 = \overline{\overline{x_1 * x_2}} \quad x_1 * x_2 = \overline{\overline{x_1 + x_2}}$$

Комбинационные законы

Закон тавтологии:

$$x + x + x + \dots + x = x;$$

$$x * x * x * \dots * x = x$$

Переместительный (коммутативный) закон:

$$x_1 + x_2 = x_2 + x_1; \quad x_1 * x_2 = x_2 * x_1$$

Сочетательный (ассоциативный) закон:

$$(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3);$$

$$(x_1 * x_2) * x_3 = x_1 * (x_2 * x_3)$$

Распределительный закон:

$$x_1(x_2 + x_3) = x_1 * x_2 + x_1 * x_3;$$

$$x_1 + (x_2 * x_3) = (x_1 + x_2)(x_1 + x_3)$$

Закон поглощения (абсорбции):

$$x_1 + x_1 * x_2 = x_1$$

$$x_1(x_1 + x_2) = x_1$$

Закон склеивания:

$$x_1 * x_2 + x_1 * \bar{x}_2 = x_1$$

$$(x_1 + x_2) * (x_1 + \bar{x}_2) = x_1$$

Теорема Шеннона.

Шеннон доказал справедливость закона Де Моргана для нескольких переменных.

«Операция инвертирования произвольной комбинации двоичных переменных, связанных знаками дизъюнкции и конъюнкции эквивалентна замене в этой комбинации исходных значений двоичных переменных их инверсными значениями при одновременной замене знаков дизъюнкции и конъюнкции»

Пример применения теоремы Шеннона:

$$\begin{aligned} Y &= a * b + c = \overline{\overline{a + b}} + c = \\ &= \overline{\overline{a + b}} * \bar{c} = \overline{a + b} * \bar{c} \end{aligned} \quad (2.2.)$$

Определение Основного Функционально полного набора.

Набор функций дизъюнкции, конъюнкции и инверсии, который соответствует трём операциям булевой алгебры-логики, получил название Основного функционально-полного набора.

Двумя другими функционально полными наборами являются функции Пирса и Шеффера.

Функционально полная система логических элементов

Систему логических элементов называют функционально полной, если есть возможность создать любые заданные переключательные выходные функции.

Понятие « базис».

Система переключательных функций, образующую функционально полную систему, логических функций называется базисом.

Переключательная функция.

Переключательной функцией называется математическое выражение, связывающее между собой элементарные двоичные логические переменные, принимающие значения «0» и «1».

Дизъюнктивно-нормальная форма (ДНФ)

Если логическая функция выражена посредством логической суммы элементарных конъюнкций, то считается, что она задана своей ДНФ.

$$Y = A * B + C * D = (A \wedge B) \vee (C \wedge D) \quad (2.3)$$

Элементарной конъюнкцией называется логическое произведение двоичных переменных и их отрицаний, причём, каждая переменная в произведении должна встречаться, только один раз. Например:

$$x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4 \wedge \overline{x_5} \quad (2.4)$$

Рангом конъюнкции называется число двоичных переменных, составляющих элементарную конъюнкцию.

Например : $x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4 \wedge \overline{x_5}$ - это конъюнкция 5-го ранга, так как составлена из произведения пяти переменных и их отрицаний.

Конъюнктивно-нормальная форма (КНФ).

Если логическая функция выражена посредством логического произведения элементарных дизъюнкций, то считается, что она задана своей КНФ.

$$Y = (A + B) * (C + D) = (A \vee B) \wedge (C + D) \quad (2.5)$$

Элементарной дизъюнкцией n-го ранга называется логическая сумма двоичных переменных и их отрицаний, причём, каждая переменная в сумме должна встречаться, только один раз. Например:

$$x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee \overline{x_5} \quad (2.6)$$

Рангом дизъюнкции называется число двоичных переменных, составляющих элементарную дизъюнкцию.

Например: $x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee \overline{x_5}$ - это дизъюнкция 5-го ранга, так как составлена из логической суммы пяти переменных и их отрицаний.

Одна и та же логическая функция может быть представлена как своей ДНФ так и КНФ, путём эквивалентных преобразований.

Из множества этих нормальных форм функций выделяют одну совершенную дизъюнктивную (СДНФ) и одну совершенную конъюнктивную (СКНФ) формы.

Совершенной дизъюнктивно-нормальной формой логической функции от n двоичных переменных называется такая ДНФ логической функции, в которой:

- все конъюнкции имеют один и тот же ранг;
- нет двух одинаковых конъюнкций;
- каждая конъюнкция содержит либо прямое, либо инверсное значение двоичной переменной;
- ни одна конъюнкция не содержит двух одинаковых двоичных переменных.

Конъюнкции n -го ранга, составляющие СДНФ функции и обращающие функцию в 1 при определённом наборе переменных, получили название минтермы.

Совершенной конъюнктивно-нормальной формой логической функции от n двоичных переменных, называется такая КНФ логической функции в которой:

- все дизъюнкции имеют один и тот же ранг;
- каждая дизъюнкция содержит либо прямое, либо инверсное значение двоичной переменной;
- нет двух одинаковых дизъюнкций;
- ни одна дизъюнкция не содержит двух одинаковых двоичных переменных.

Дизъюнкции n -го ранга, составляющие СКНФ функции и обращающие функцию в 0 при определённом наборе переменных, получили название минтермы.

Примеры СКНФ и СДНФ.

СДНФ:

$$Y = (x_1 * \bar{x}_2 * x_3) + (\bar{x}_1 * \bar{x}_2 * \bar{x}_3) + (x_1 * x_2 * x_3) \quad (2.7)$$

СКНФ:

$$Y = (x_1 + \bar{x}_2 + x_3) * (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) * (x_1 + x_2 + x_3) \quad (2.8)$$

Пример. Табличное задание булевой функции приведено в табл. 2.3

Таблица 2.3

Таблица задания булевой функции.

x_3	x_2	x_1	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Составить булевы функции в СДНФ и СКНФ.

Решение:

Составление СДНФ по табличному заданию булевой функции сформулировано в следующих правилах:

соответствующие минтермам элементарные конъюнкции объединить знаками дизъюнкций,

в элементарных конъюнкциях переменная=1 записывается прямым значением, а переменная=0, своим инверсным значением (с черточкой над именем переменной).

Итоговая запись СДНФ

$$Y = (\bar{x}_3 * x_2 * x_1) + (x_3 * \bar{x}_2 * x_1) + (x_3 * x_2 * \bar{x}_1) + (x_3 * x_2 * x_1) \quad (2.9)$$

Составление СКНФ по табличному заданию булевой функции сформулировано в следующих правилах:

Соответствующие минтермам элементарные дизъюнкции объединить знаками конъюнкций, в элементарных дизъюнциях переменная, равная нулю, записывается прямым значением, а переменная, равная единице, своим инверсным значением (с черточкой над именем переменной).

$$Y = (x_3 + x_2 + x_1) * (x_3 + x_2 + \overline{x_1}) * (x_3 + \overline{x_2} + x_1) * (\overline{x_3} + x_2 + x_1) \quad (2.10)$$

Минимизация булевых функций.

Основная задача минимизации состоит в получении такой формы, которой соответствует логическая функция с минимальным числом элементов. Различают несколько методов минимизации булевых функций.

При эвристических методах преобразования логических функций, использующих законы алгебры логики. Конечный вид минимизируемой функции в значительной степени зависит от квалификации и опыта разработчика цифровых устройств. К самым распространённым методам относятся:

Методы Квайна и Мак-Класки используются, вследствие четко сформулированных правил проведения отдельных операций, для минимизации сложных функций по разработанным алгоритмам с использованием ЭВМ.

Метод карт Карно или карт Вейча, отличающихся способом обозначения строк и столбцов таблицы истинности, нашел применение при минимизации логических функций с числом двоичных переменных не более пяти-шести.

Основным методом минимизации логических функций, представленных в виде СДНФ или СКНФ является операция попарного неполного склеивания

и элементарного поглощения. Операция попарного склеивания осуществляется между двумя термами (членами), содержащими одинаковые переменные, вхождения которых (прямые и инверсные) совпадают для всех переменных, кроме одной. В этом случае все переменные, кроме одной, можно вынести за скобки, а оставшиеся в скобках прямое и инверсное вхождение одной переменной подвергнуть склейке.

Главной задачей при минимизации СДНФ и СКНФ является поиск термов, пригодных к склейке с последующим поглощением, что для больших форм может оказаться достаточно сложной задачей. Карты Карно предоставляют наглядный способ отыскания таких термов.

Как известно, булевы функции N переменных, представленные в виде СДНФ или СКНФ могут иметь в своём составе $2N$ различных термов. Все эти члены составляют некоторую структуру, топологически эквивалентную N -мерному кубу, причём любые два терма, соединённые ребром, пригодны для склейки и поглощения.

Метод карт Карно.

Карту Карно можно рассматривать как графическое представление совокупности всех наборов переменных для данного числа переменных.

Карты Карно были изобретены в 1952 Эдвардом В. Вейчем и усовершенствованы в 1953 Морисом Карно, физиком из «Bell Labs», и были призваны помочь упростить цифровые электронные схемы.

Каждый набор переменных изображается на карте в виде клетки. Таким образом, при $n=3$ карта имеет 8 клеток, а при $n=6$ – 64 клетки, рис.2.1 и рис.2.2 соответственно.

x_1x_2	00	01	11	10
x_3				
0				
1				

Рис. 2.1 Карта Карно для трёх переменных.

$x_1x_2x_3$	000	001	011	010	110	111	101	100
$x_4x_5x_5$								
000								
001								
011								
010								
110								
111								
101								
100								

Рис.2.2. Карта Карно для шести переменных.

Карта Карно образуется путем такого расположения клеток, при котором наборы переменных, находящиеся в соседних клетках, отличаются значением одной переменной. В картах Карно соседними считаются также крайние клетки каждого столбца или строки. Расположенные в них наборы переменных отличаются значением одной переменной.

Минтермы логической функции, т.е. наборы двоичных переменных, при которых эта функция равна 1, отмечаются единицами в соответствующих клетках. Для наборов переменных не входящих в логическую функцию соответствующие им клетки остаются пустыми.

Логическая функция, записанная в СДНФ или заданная в виде таблицы истинности, переносится на карту Карно. Затем карта покрывается контурами. В контур может входить 2^n рядом расположенных клеток, содержащих единичное значение логической функции, т.е. 2,4,8 и т.д. точек. Допускается пересечение контуров.

Два минтерма, находящиеся в соседних клетках, т.е. в одном контуре, могут быть заменены одним логическим произведением, содержащим на одну переменную меньше. Исключается та переменная, которая меняет своё значение при переходе из одной клетки в другую. Если соседними являются

две пары минтермов, то такая группа из четырех минтермов может быть заменена конъюнкцией двоичных переменных, содержащих на две переменных меньше. В общем случае, наличие единиц в 2^n соседних клетках позволяет исключить n переменных.

При минимизации с помощью карт Карно рекомендуется следовать следующему порядку действий:

Необходимо образовывать контура, в которые входило бы максимально возможное количество клеток с минтермами - произведение будет наиболее простым. Контуров должно быть как можно меньше, чтобы было меньше слагаемых.

После покрытия карты контурами производится их анализ с точки зрения уменьшения числа переменных. На основе анализа контуров записывается минимизированная ДНФ (МДНФ) логической функции в виде логической суммы логических произведений двоичных переменных. При этом двоичные переменные, имеющие единичное значение записываются без инверсии, а имеющие нулевое значение с инверсией.

Минимизацию с помощью карт Карно можно использовать и для логических функций представленных в СКНФ. В этом случае, наборы двоичных переменных, при которых логическая функция равна 0 (макстермы), отмечаются нулями в соответствующих клетках карты. Аналогично образуются контура, охватывающие клетки с макстермами, далее контура анализируются, и записывается минимальная КНФ (МКНФ) логической функции в виде логического произведения логических сумм двоичных переменных, в которых двоичные переменные, имеющие нулевое значение, записываются без инверсии, а имеющие единичное значение с инверсией.

Крайние квадраты карты являются соседними при ее скручивании. Это значит, что они тоже подлежат минимизации. На плоскости можно изобразить карту Карно для 4-х переменных. Для 5 и более переменных необходимы объемные фигуры.

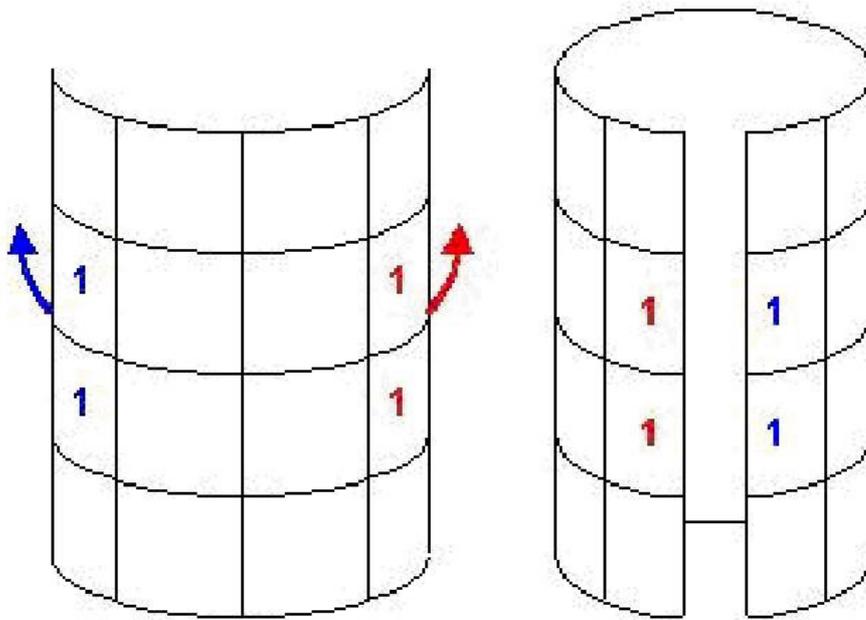


Рис. 2.3 Скручивание карты Карно.

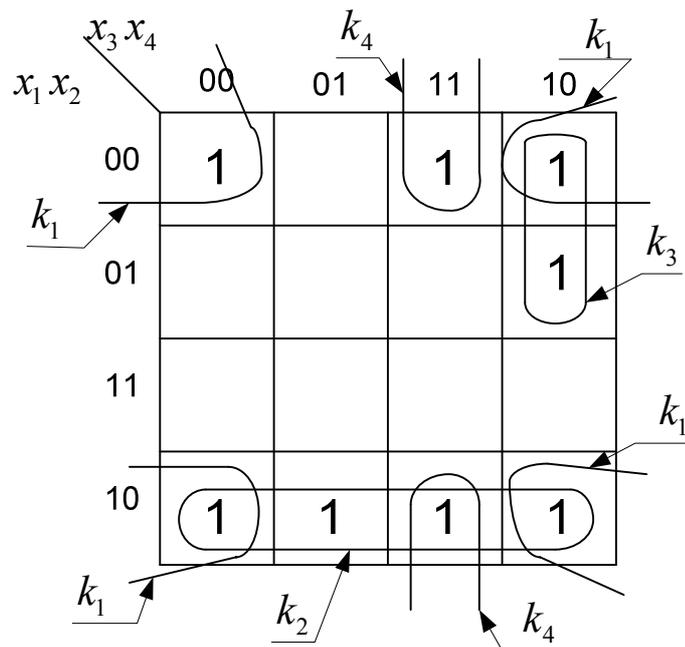


Рис. 2.4 Пример карты Карно для 4-х переменных.

Приобретаемые компетенции после изучения главы 2.

Помнить:

- законы и теоремы алгебры логики, основные определения и термины.

Понимать:

- различия и особенности дизъюнктивных и конъюнктивных форм булевых функций.

- особенности совершенных дизъюнктивных и конъюнктивных форм булевых функций.

Применять:

- законы и теоремы булевой алгебры логики с целью получения упрощённых (минимизированных) выражений.

- методы минимизации булевых функций.

Анализировать и оценивать:

- возможность применения аппарата булевой алгебры в проектных решениях переключательных и комбинационных схем.

2.3. Контрольные вопросы.

Вопросы категории 1. «Помнить»

2.3.1 С помощью какого закона можно сделать преобразование булевой функции из одного функционально-полного набора в другой?

2.3.2 Какие два значения может принимать булева переменная?

2.3.3 Какое число булевых переменных может входить в булеву функцию?

2.3.4 Какие два значения может принимать булева функция?

2.3.5 Перечислите три функционально-полных набора логических функций.

Вопросы категории 2. «Понимать»

2.3.6 Сформулируйте правила, характеризующие нормальную дизъюнктивную и нормальную конъюнктивную форму булевой функции.

2.3.7 Сформулируйте правила, характеризующие совершенную нормальную дизъюнктивную и совершенную нормальную конъюнктивную форму булевой функции.

2.3.8 Какую информацию содержит таблица истинности булевой функции?

2.3.9 Дайте определение понятиям элементарная конъюнкция и элементарная дизъюнкция.

2.3.10. Дайте определение терминам: ранг элементарной конъюнкции и ранг элементарной дизъюнкции.

Вопросы категории 3. «Применять»

2.3.11 Каким свойством обладает функционально полный набор логических элементов?

2.3.12 Поясните термин «минтерм», применительно к функции, заданной таблицей истинности.

Вопросы категории 4. «Анализировать и оценивать»

2.3.13 Какие цели достигаются при минимизации булевых функций?

2.3.14 На каких подходах основаны эвристический метод минимизации булевых функций?

Ответы на контрольные вопросы.

2.3.1 Преобразование из одного функционально полного набора в другой возможно выполнить используя закон Де Моргана и следствия из этого закона.

2.3.2 Булева переменная может принимать значения либо 0 (ложь), либо 1 (истина).

2.3.3 В булеву функцию может входить любое конечное значение булевых переменных.

2.3.4 Булева функция может принимать значения либо 0 (ложь), либо 1 (истина).

2.3.5 К трём функционально полным наборам относятся:

- конъюнкция, дизъюнкция, инверсия, второй набор состоит из функции «штрих Шеффера», третий набор состоит из функции «стрелка Пирса».

2.3.6 Если логическая функция выражена посредством логической суммы элементарных конъюнкций, то считается, что она задана своей дизъюнктивно-нормальной формой (ДНФ). Если логическая функция выражена посредством логического произведения элементарных

дизъюнкций, то считается, что она задана своей конъюнктивно нормальной формой (КНФ).

2.3.7 Совершенная дизъюнктивно-нормальная форма логической функции от n двоичных переменных называется такая ДНФ логической функции, в которой:

- все конъюнкции имеют один и тот же ранг,
- нет двух одинаковых конъюнкций,
- каждая конъюнкция содержит либо прямое, либо инверсное значение двоичной переменной,
- ни одна конъюнкция не содержит двух одинаковых двоичных переменных.

2.3.8 Таблица истинности булевой функции содержит перечень соответствий значению булевой функции от каждого из возможных наборов (сочетаний) значений составляющих её переменных.

2.3.9 Элементарная конъюнкция это логическое произведение n -го количества булевых переменных, при условии, что переменная, либо её инверсное значение входит в это произведение только один раз. Элементарная дизъюнкция это логическая сумма n -го количества булевых переменных, при условии, что переменная, либо её инверсное значение входит в эту логическую сумму только один раз.

2.3.10 Рангом элементарной конъюнкции называется количество булевых переменных, входящих в элементарную конъюнкцию. Рангом элементарной дизъюнкции называется количество переменных, входящих в элементарную дизъюнкцию.

2.3.11 Функционально-полный набор логических функций позволяет выразить любую булеву функцию.

2.3.12 Минтермом называют набор значений переменных, обращающих в 1 значение булевой функции, заданной в СДНФ, и обращающий в 0 значение булевой функции, заданной в СКНФ.

2.3.13 Уменьшение количества логических элементов, входящих в комбинационную схему, реализующую данную булеву функцию.

2.3.14 Метод основан на использовании аксиом, теорем и законов алгебры логики.

Глава 3. Математический аппарат проектирования комбинационных схем дискретных устройств.

3.1. Синтез комбинационных схем.

Цифровые электронные схемы на логических элементах нашли очень широкое применение в конструкции многих узлов вычислительных устройств, устройств автоматики и устройствах управления. Под синтезом схемы понимают её проектирование (разработку). Данные электронные схемы получили наименование комбинационных схем (КС).

Функционирование комбинационной схемы описывается булевой функцией.

Постоянные уровни напряжения, соответствующие принятому в схеме представлению 0 и 1, могут рассматриваться как технические аналоги функции «ложь» и «истина».

Существуют различные способы задания или представления булевых функций:

1. Словесное представление функций.

Например: функция от трех аргументов принимает значение 1, если два любых аргумента или все три равны 1. Во всех других случаях функция равна 0.

Этим высказыванием значения выходной функции соответствующей схемы полностью задано.

2. Табличный способ.

При этом способе функция представляется в виде таблицы истинности, в которой записываются все возможные наборы аргументов и для каждого набора устанавливается значение функции 0 и 1.

3. Алгебраический способ.

От таблицы истинности можно перейти к алгебраической форме представления функции. В такой форме удобно производить различные преобразования функций, например, с целью их минимизации.

Наиболее удобным является задание булевой функции с помощью таблицы истинности. Таблица истинности однозначно определяет, как будет работать синтезируемая комбинационная схема. Далее необходимо определить, на основе каких логических элементов можно построить КС. КС должна быть как можно проще и состоять из логических элементов, имеющихся в наличии.

В процессе синтеза КС необходимо попытаться минимизировать булеву функцию, применив один из известных методов минимизации, и сделать преобразования к заданному одному из функционально полных наборов.

Типовой порядок проектирования комбинационных схем состоит из следующих этапов:

1. Назначение входных и выходных переменных и присвоение значений 0 и 1.
2. Определение табличных значений поведения булевой функции синтезируемой комбинационной схемы.
3. Составление СДНФ (либо СКНФ) по минтермам табличной записи булевой функции.
4. Упрощение, то есть минимизация СДНФ (или СКНФ) и получение минимальной ДНФ (СКНФ).
5. Переход от минимальной ДНФ (либо КНФ) к минимизированной форме в каком-либо базисе функционально-полного набора.
6. Составление комбинационной схемы из логических элементов, входящих в указанный базис.

Дизъюнктивная нормальная форма (ДНФ) представляет собой логическую сумму элементарных логических произведений, в каждое из которых аргумент или его отрицание входят не более одного раза.

$$\text{Например: } f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 + x_2 \bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 \quad (3.1)$$

Если каждое слагаемое содержит все переменные или их отрицания, то в этом случае логическая функция представлена в совершенной дизъюнктивной нормальной форме (СДНФ).

Конъюнктивная нормальная форма (КНФ) представляет собой логическое произведение элементарных логических сумм, в каждую из которых аргумент или его отрицание входят не более одного раза.

$$\text{Например: } f(x_1, x_2, x_3) = (\bar{x}_1 + \bar{x}_2)(x_2 + \bar{x}_3)(\bar{x}_1 + x_2 + \bar{x}_3) \quad (3.2)$$

Переход от таблицы истинности к СДНФ можно осуществить следующим путем. Для каждого набора, на котором функция равна единице, записывается произведение всех аргументов, причем, если аргумент в этом наборе принимает значение "0", то пишется его отрицание. Затем производится логическое сложение этих элементарных произведений.

Для перехода от таблицы истинности к СКНФ логической функции, по каждому набору двоичных переменных, на котором функция принимает значение "0", записывается дизъюнкция всех переменных, и полученные дизъюнкции логически перемножаются. При записи логических сумм инвертируются те переменные, которые в таблице истинности имеют значение единицы.

Пример написания СДНФ и СКНФ логической функции. Пусть логические функции Y_1 и Y_2 заданы в виде таблицы истинности, табл.3.1.

Таблица.3. 1.

Типовая таблица истинности

x_1	x_2	x_3	y_1	y_2
0	0	0	0	0

0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	1	0

Тогда СДНФ и СКНФ логических функций y_1 и y_2 запишутся следующим образом:

$$Y_1^{СДНФ} = x_1x_2x_3 + x_1x_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 \quad (3.3)$$

$$Y_1^{СКНФ} = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3) \quad (3.4)$$

$$Y_2^{СДНФ} = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1x_2\bar{x}_3 \quad (3.5)$$

$$Y_2^{СКНФ} = (x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \quad (3.6)$$

Комбинационные схемы, реализующие вышеприведенные СДНФ и СКНФ логических функций, должны содержать соответственно:

$Y_1^{сднф}$ —четыре трехвходовые схемы И и одна четырехвходовая схема ИЛИ,

$Y_1^{скнф}$ —четыре трехвходовые схемы ИЛИ и одна четырехвходовая схема «И»,

$Y_2^{сднф}$ —шесть трехвходовых схем И и одна шестивходовая схема ИЛИ,

$Y_2^{скнф}$ —две трехвходовые схемы ИЛИ и одна двухвходовая схема И.

В качестве примера минимизации с помощью карт Карно взяты логические функции, приведенные в табл.3.2.

Карта Карно для функции Y_1 приведена на рис.3.1.

$X_3 \backslash X_1 X_2$	00	01	11	10
0	0	1	0	0
1	1	1	1	1

Рис.3.1 Карта Карно для функции Y_1

Карта Карно для функции Y_2 приведена на рис. 3.2.

$X_3 \backslash X_1 X_2$	00	01	11	10
0	0	1	1	1
1	1	1	1	1

Рис. 3.2 Карта Карно для функции Y_2

После минимизации с помощью карт Карно получают следующие минимальные дизъюнктивная и конъюнктивная нормальные формы логических функций y_1 и y_2 :

$$y_1^{мднф} = x_3 + \bar{x}_1 x_2, \quad (3.7)$$

$$y_1^{мкнф} = (\bar{x}_1 + x_3) * (x_2 + x_3), \quad (3.8)$$

Применение закона Де Моргана, позволяет преобразовать выражения в базис «Шеффера» и «Пирса».

$$y_1^{мшефф} = \overline{\overline{x_3 * \overline{\overline{x_1 x_2}}}} \quad (3.9)$$

$$y_1^{мпирс} = \overline{\overline{(\bar{x}_1 + x_3) + (x_2 + x_3)}} \quad (3.10)$$

Схемы реализации функций y_1 на элементах Шеффера и Пирса приведены на рис.3.3. и на рис.3.4.

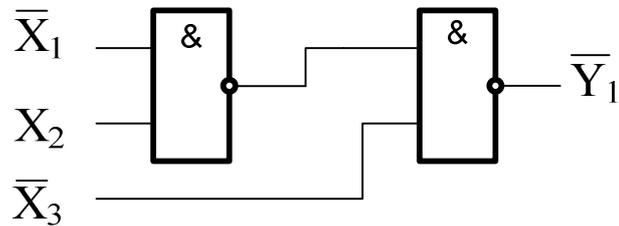


Рис.3.3 Реализация функции Y_1 на элементах Шеффера.

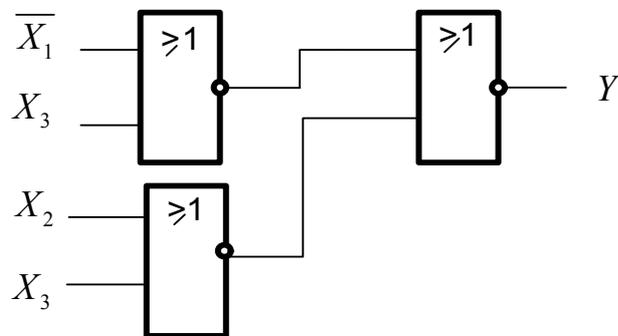


Рис.3.4. Реализация функции Y_1 на элементах Пирса.

Рассмотрим ещё один пример синтеза комбинационной схемы. Необходимо спроектировать схему, разрешающую движение лифта. Описание входных и выходных переменных:

x_1 - дверной контакт, при замкнутом контакте (дверь закрыта) $x_1 = 1$, при разомкнутом контакте $x_1 = 0$ (дверь открыта).

x_2 - перегрузка, $x_2 = 1$ - перегрузка, $x_2 = 0$ - нет перегрузки,

x_3 - кнопка этажа, $x_3 = 1$ -кнопка этажа нажата, $x_3 = 0$, кнопка этажа не нажата,

Y - сигнал движения лифта: $Y = 1$ - лифт может двигаться, $Y = 0$ движение лифта блокируется.

При трёх переменных таблица истинности должна иметь 8 различных вариантов переменных. С учётом логики движения лифта составляем таблицу истинности (табл. 3.2 .)

Таблица 3.2.

Таблица истинности сигнала управления лифтом.

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Записываем СДНФ: $Y = x_1 * \overline{x_2} * x_3$, этой функции соответствует комбинационная схема рис. 16.

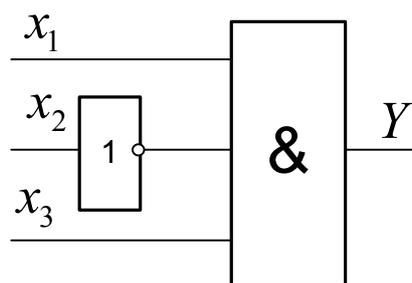


Рис. 3.5. Комбинационная схема безопасности пуска лифта.

Если необходимо реализовать данную схему в другом базисе, например «ИЛИ-НЕ», то путём преобразований на основе теоремы Шеннона можно получить другое выражение булевой функции:

$$Y = x_1 * \overline{x_2} * x_3 = \overline{\overline{x_1 + x_2 + x_3}} \quad (3.13)$$

Комбинационная схема, соответствующая преобразованному выражению представлена на рис. 3.6 .

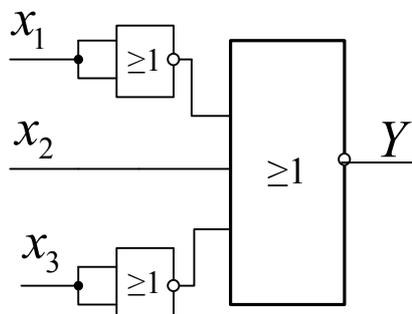


Рис. 3.6 . Комбинационная схема безопасности пуска лифта на элементах «ИЛИ-НЕ».

Пример синтеза схемы аварийного отключения « два из трёх».

В системах, связанных с повышенным риском, строятся управляющие отключающие электронные схемы с повышенной надёжностью срабатывания. В аварийных датчиках, ответственных за отключение, могут быть ложные срабатывания. Поэтому в каждом критическом месте ставят три одинаковых аварийных датчика. Отключение должно происходить, когда сработали не менее двух датчиков из трёх. Аварийный датчик принимает значение «1» при срабатывании, отключение системы должно происходить, когда значение выходного сигнала управляющей схемы Y равно 1. Аварийным датчикам соответствуют булевы переменные x_1, x_2, x_3 .

Таблица истинности для синтеза схемы аварийного отключения (табл.3.3):

Таблица 3.3.

Таблица истинности для синтеза схемы аварийного отключения.

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	0

0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

СДНФ по таблице истинности имеет следующий вид:

$$Y = \overline{x_1} * x_2 * x_3 + x_1 * \overline{x_2} * x_3 + x_1 * x_2 * \overline{x_3} + x_1 * x_2 * x_3 \quad (3.14)$$

Проверим возможность минимизации полученной функции с помощью карт Карно. Карта Карно для данной функции представлена на рис. 3.7 и имеет три контура. Анализ контуров даёт возможность записать минимизированную нормальную форму:

$$Y = x_2 * x_3 + x_1 * x_3 + x_1 * x_2 \quad (3.15)$$

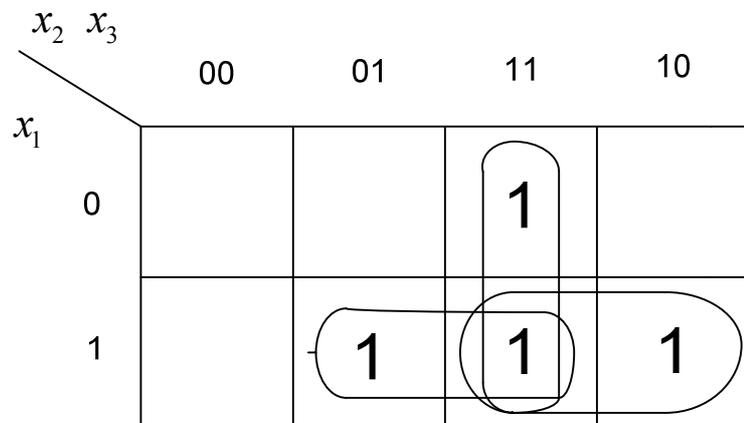


Рис. 3.7 Карта Карно для функции (3.15)

Выражению (3.15) соответствует комбинационная схема управления аварийным отключением рис.3.8

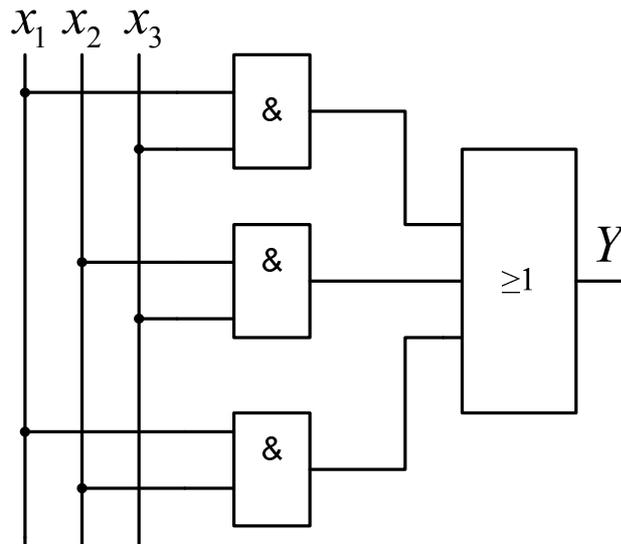


Рис. 3.8 Комбинационная схема комбинационная схема управления аварийным отключением.

3.2. Типовые примеры проектных решений.

3.2.1. Синтез комбинационной схемы контроля чётности.

Для обнаружения однократных ошибок в линиях передачи данных в двоичных кодах применяется схема, в которой выход равен 1, когда в составе кода, подаваемого на вход этой схемы, содержится чётное количество единиц. Такая схема получила название схемы контроля чётности.

Будем считать, код, подлежащий контролю четырёхразрядный. То есть количество входов у схемы контроля чётности равно четырём, и она имеет один выход. Построим таблицу истинности для проектируемой схемы (табл. 3.4.)

Таблица 3.4.

Таблица функции проектируемой схемы.

x_4	x_3	x_2	x_1	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

По таблице истинности получается СДНФ:

$$\begin{aligned}
 Y = & (\overline{x_1} * \overline{x_2} * \overline{x_3} * \overline{x_4}) + (x_1 * x_2 * \overline{x_3} * \overline{x_4}) \\
 & + (x_1 * \overline{x_2} * x_3 * \overline{x_4}) + (\overline{x_1} * x_2 * x_3 * \overline{x_4}) \\
 & + (x_1 * \overline{x_2} * \overline{x_3} * x_4) + (\overline{x_1} * x_2 * \overline{x_3} * x_4) \\
 & + (\overline{x_1} * \overline{x_2} * x_3 * x_4) + (x_1 * x_2 * x_3 * x_4)
 \end{aligned}
 \tag{3.16}$$

Составим карту Карно для проверки возможности минимизации (рис. 3.9.)

		$x_3 x_4$			
		00	01	11	10
$x_1 x_2$	00	1		1	
	01		1		1
	11	1		1	
	10		1		1

Рис. 3.9. Карта Карно к таблице истинности табл. 3.4.

Исследования карты Карно показывают невозможность объединения контуров. Следовательно, выражение для СДНФ упрощено быть не может.

Комбинационная схема контроля чётности будет иметь вид (рис. 3.10)

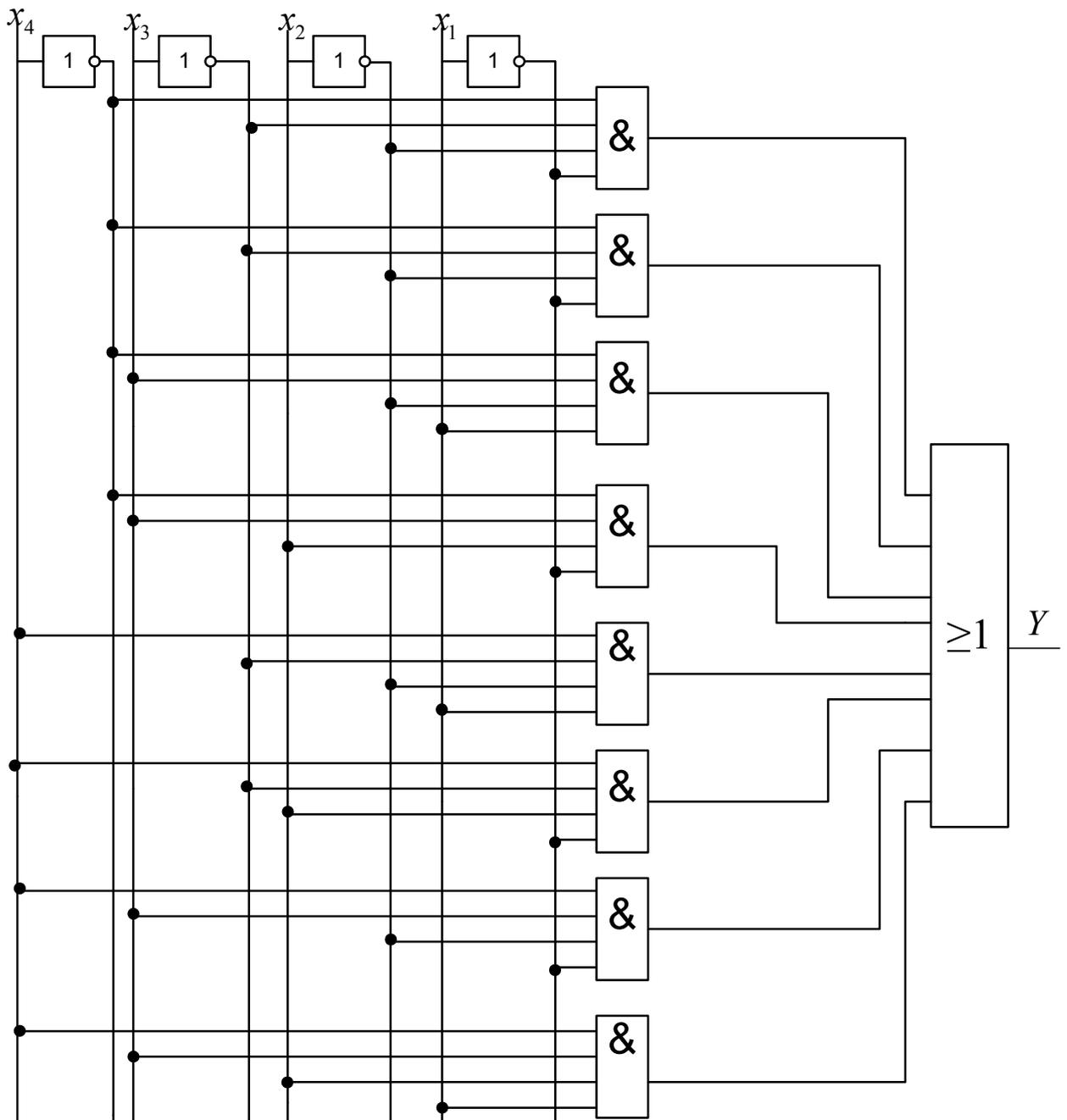


Рис. 3.10. Комбинационная схема контроля чётности.

3.2.2. Компьютерный зал оснащён двумя кондиционерами разной мощности для поддержания заданных температурных условий. Необходимо спроектировать комбинационную схему включения кондиционеров один из которых малой мощности, другой – большой мощности.

Схема включения кондиционеров получает данные о текущей температуре в зале от трёх температурных датчиков: датчика превышения температуры больше 20 градусов ($t_{20} = 1$), датчика превышения температуры больше 25 градусов ($t_{25} = 1$), и датчика превышения температуры больше 30 градусов ($t_{30} = 1$). Сигналом включения кондиционера малой мощности является выход комбинационной схемы включения Y_1 ($Y_1 = 1$), сигналом включения кондиционера большой мощности является выход комбинационной схемы включения Y_2 ($Y_2 = 1$).

При температуре в зале меньше 20 градусов оба кондиционера выключены.

В диапазоне от 20 градусов до 25 градусов достаточно мощности малого кондиционера. В диапазоне от 25 до 30 градусов достаточно мощности одного большого кондиционера. При повышении температуры в зале выше 30 градусов должны работать оба кондиционера.

Согласно перечисленным условиям работы составим таблицу истинности для сигналов включения кондиционеров (табл. 3.5.).

Таблица 3.5.

Таблица управления кондиционерами.

t_{30}	t_{25}	t_{20}	Y_1	Y_2
0	0	0	0	0
0	0	1	1	0
0	1	0	*	*
0	1	1	0	1
1	0	0	*	*
1	0	1	*	*
1	1	0	*	*
1	1	1	1	1

Состояния, помеченные * являются неопределёнными, так как такого сочетания сигналов при исправном состоянии датчиков быть не должно.

Выражения для сигналов Y_1 и Y_2 в СНДФ получаются следующие:

$$Y_1 = \overline{t_{30}} * \overline{t_{25}} * t_{20} + t_{30} * t_{25} * t_{20}; \quad (3.17)$$

$$Y_2 = \overline{t_{30}} * t_{25} * t_{20} + t_{30} * \overline{t_{25}} * t_{20} \quad (3.18)$$

На рис. 3.11 представлена комбинационная схема управления включением кондиционеров.

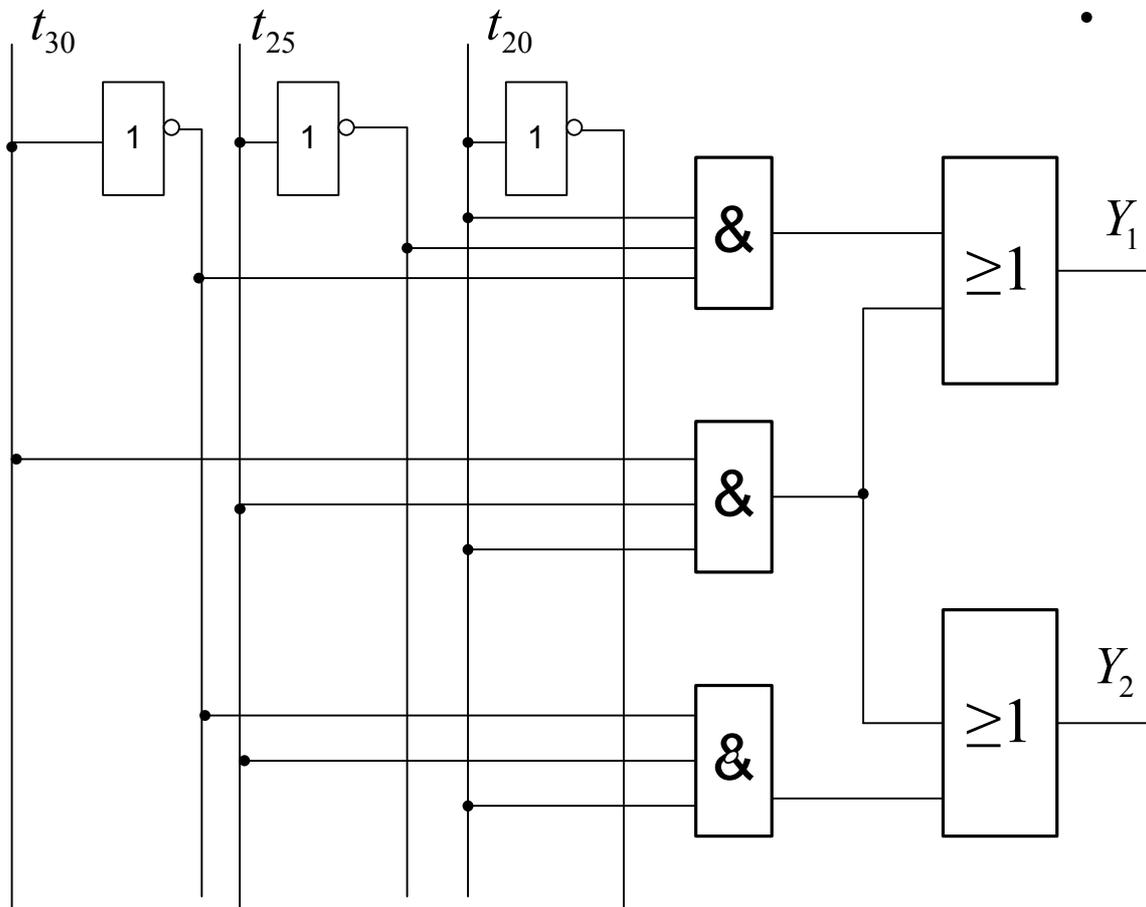


Рис. 3.11. Комбинационная схема управления включением кондиционеров.

Приобретаемые компетенции после изучения главы 3.

Помнить:

- правила составления карт Карно и методы анализа контуров с целью получения минимизированной записи булевого выражения.

Понимать:

- различия и особенности дизъюнктивных и конъюнктивных форм булевых функций, минимизируемых с помощью карт Карно.

Применять:

- методы синтеза комбинационных схем, функционирование которых задаётся таблицей истинности булевой функции, соответствующей правилам работы искомой комбинационной схемы.

Оценивать:

- возможность применения аппарата булевой алгебры в проектных решениях переключательных схем.

- выбирать базис логических элементов комбинационных схем в зависимости от поставленной в проектном задании задачи и дополнительных условий.

- уметь обосновать тождественность нескольких решений топологии комбинационных схем в зависимости от использования дизъюнктивной или конъюнктивной формы записи булевой функции.

3.3. Контрольные вопросы.

Вопросы категории 1. «Помнить»

3.3.1 Напишите формулы, выражающие следствие, вытекающее из закона Де Моргана.

3.3.2 Напишите формулу, используемую для расчёта числа переменных, которые можно объединять в контур на карте Карно?

3.3.3 Какие значения переменных объединяются в контура при минимизации дизъюнктивной формы?

3.3.4 Сформулируйте правило обозначений последовательностей кодов переменных в карте Карно.

Вопросы категории 2. «Понимать»

3.3.5 Поясните отличие комбинационной схемы от цифрового автомата?

3.3.6 Сформулируйте понятие «соседние клетки» применительно к карте Карно.

3.3.7 Какие значения переменных объединяются в контуры при минимизации конъюнктивной формы?

Вопросы категории 3. «Применять»

3.3.8 Какие диагональные клетки считаются соседними в прямоугольной карте Карно при минимизации булевой функции?

Ответы на контрольные вопросы.

3.3.1. Следствие из закона Де Моргана:

$$x_1 + x_2 = \overline{\overline{x_1} * \overline{x_2}} \quad (3.19)$$

$$x_1 * x_2 = \overline{\overline{x_1} + \overline{x_2}} \quad (3.20)$$

3.3.2. В карту Карно можно объединять количество единиц (или нулей) согласно формуле:

$$k = 2^n \quad (3.21)$$

3.3.3. При минимизации дизъюнктивной формы в карту Карно заносятся единицы.

3.3.4. Последовательность обозначений в строках и столбцах карты Карно должна соответствовать коду Грея.

3.3.5. У комбинационных схем значение сигнала на её выходе полностью определяется (зависит) от множества сигналов на её входе, а у цифрового автомата сигналы на его выходах зависят не только от входных сигналов, но и внутреннего состояния цифрового автомата, в котором он находился в момент подачи входных сигналов.

3.3.6. В квадратных картах Карно кроме внутренних соседних клеток, соседними считаются четыре диагональных клетки и клетки по краям карты. В прямоугольной карте диагональные клетки не являются соседними.

3.3.7. При минимизации конъюнктивной формы в контура объединяются нули.

3.3.8. Никакие клетки по диагонали в прямоугольной карте Карно не являются соседними.

3.4 Задачи для самостоятельного решения.

3.4.1. Из приведённых на рис. 3.12 трёх комбинационных схем, выберите схему соответствующей таблице истинности (табл. 3.6.).

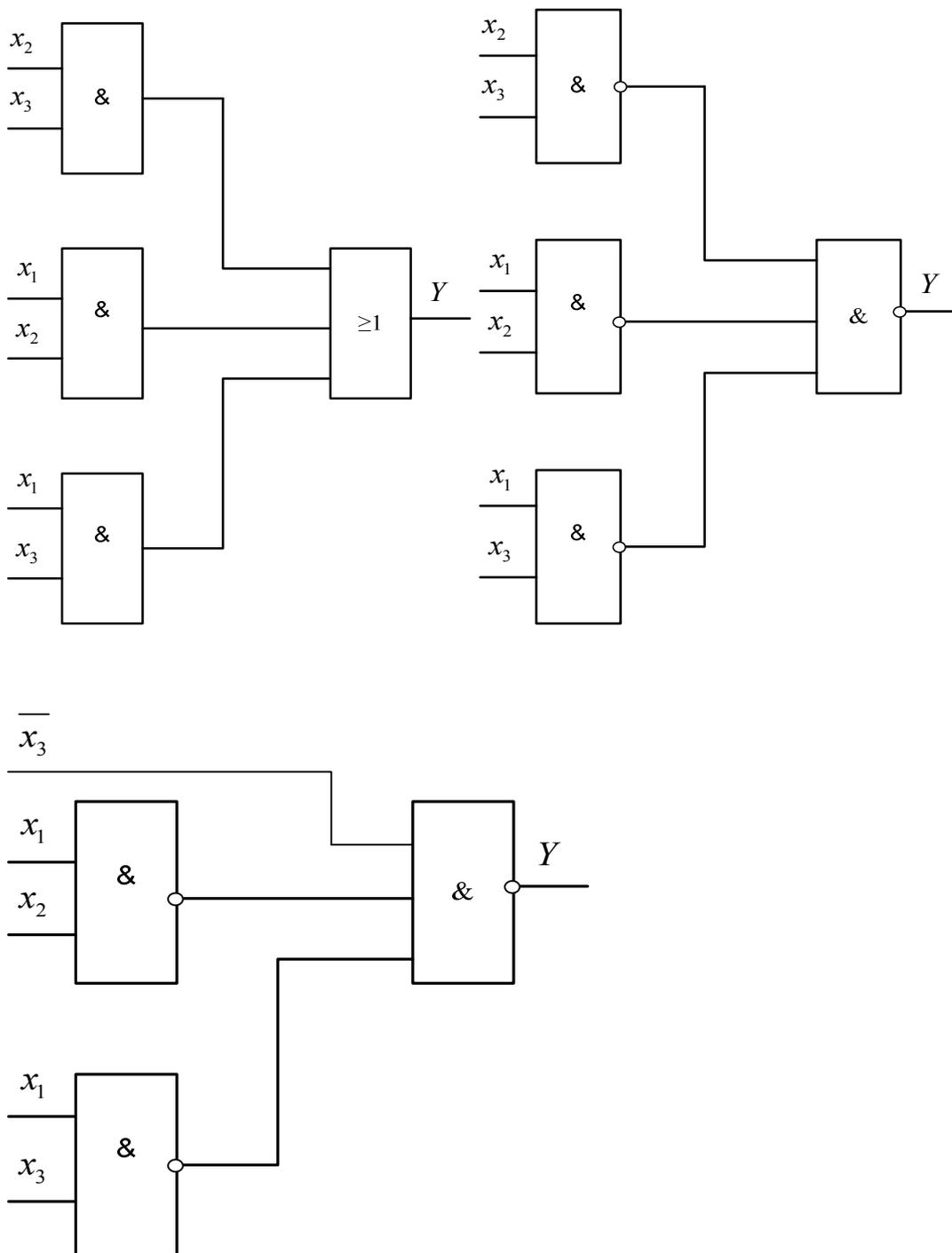


Рис. 3.12 Для получения правильной комбинационной схемы необходимо произвести следующие действия:

- синтезировать комбинационную схему, работа которой описывается булевой функцией, заданной следующей таблицей истинности (табл. 3.6);

Таблица 3.6

Таблица истинности булевой функции к заданию 3.4.1

x_1	x_2	x_3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- произвести минимизацию булевой функции с помощью карты Карно;
- изобразить структуру комбинационной схемы по минимизированной записи булевой функции в базисе на основе элементов Шеффера.

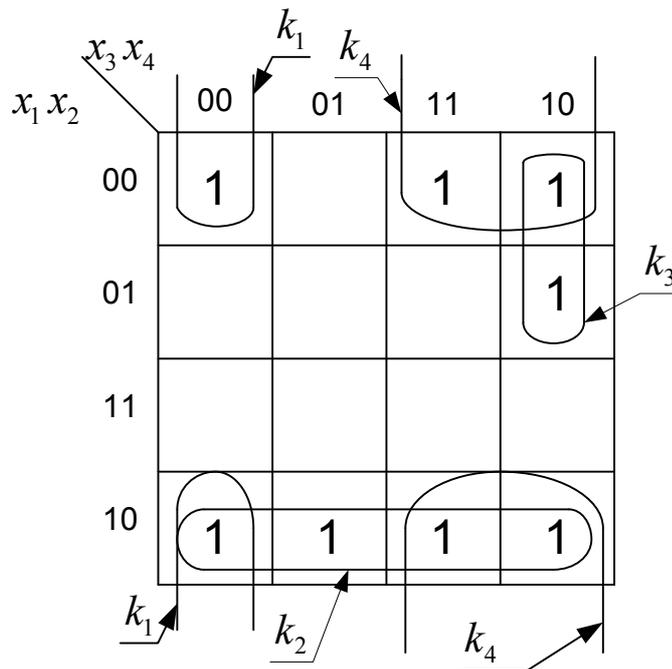
Правильный вариант комбинационной схемы соответствует рис. 3.12. б)

3.4.2. Выберите вариант правильного объединения в контура, соответствующими приведённой заполненной исходной карте Карно (рис. 3.12.)

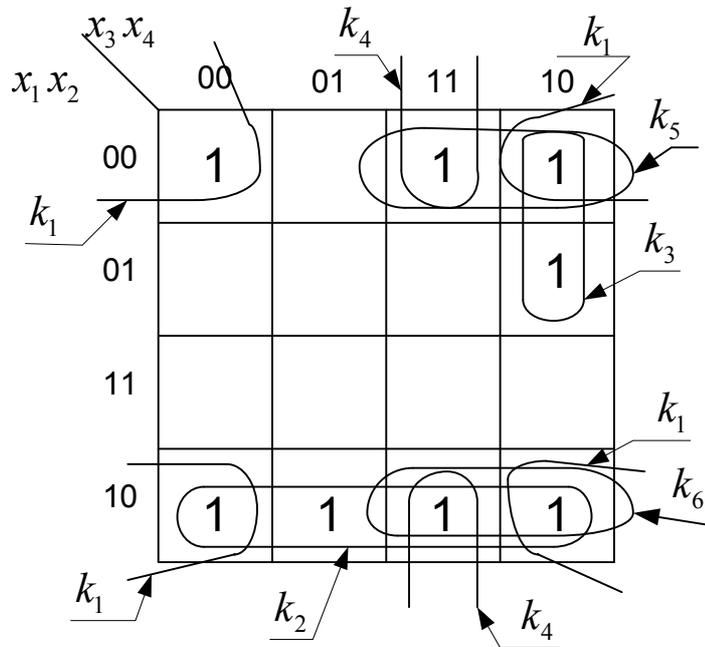
		$x_3 x_4$			
		00	01	11	10
$x_1 x_2$	00	1		1	1
	01				1
	11				
	10	1	1	1	1

Рис. 3.13. Заполненная единицами карта Карно.

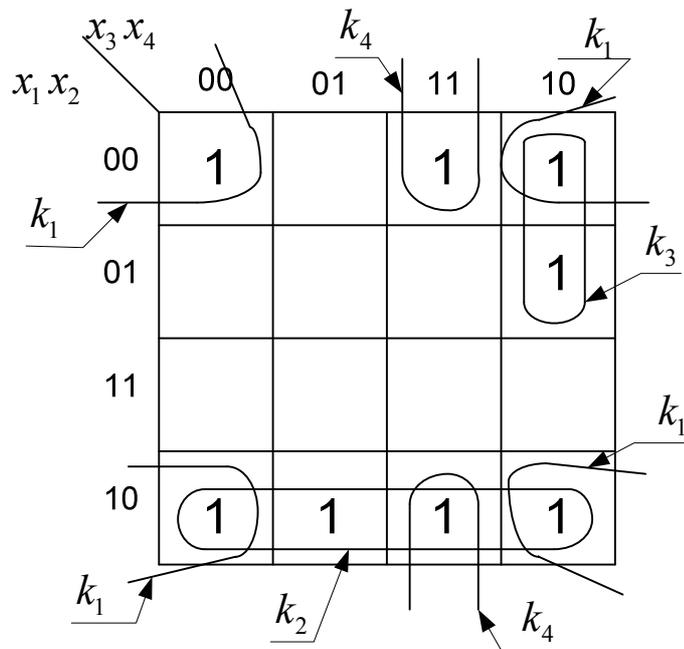
По таблице истинности (табл. 3.6.) запишите соответствующее ей выражение булевой функции в совершенной конъюнктивной нормальной форме.



a)



б)



в)

Рис. 3.14. Варианты объединения в контура. Правильный вариант А.

Глава 4. Схемотехника коммутирующих и кодирующих устройств.

4.1. Мультиплексоры и демultipлексоры.

Мультиплексор - это устройство, являющееся комбинационной схемой, служащей для передачи данных с одного из её входов на один единственный выход под управлением адресующего кода. Мультиплексоры могут быть одноразрядными и многоразрядными. Мультиплексор – это цифровой коммутатор с разделением по времени.

Демultipлексор - это устройство, являющееся комбинационной схемой, служащее для передачи данных с единственного входа на один их выходов под управлением адресующего кода. Демultipлексор может быть одноразрядным и многоразрядным.

Эти два устройства именуют цифровыми коммутаторами. Управляющие входы этих двух устройств именуют адресными входами.

Для мультиплексора существует соотношение зависимости количества информационных входов от количества разрядов адреса. Это соотношение определяется формулой $m = 2^n$, где n – разрядность адреса, m – количество информационных входов. Двоичный код на входах адреса определяет номер того информационного входа данных, с которого информация проходит на выход мультиплексора в этот момент.

Большинство устройств, как правило, имеют «вход разрешения», то есть сигнала переводящего устройство в работу. Обозначим этот сигнал символом E , информационные входы обозначим переменными x_1, x_2, \dots, x_m , адресные входы – переменными a_1, a_2 , выходной сигнал мультиплексора Y . Тогда работа мультиплексора будет соответствовать таблице истинности (табл. 4.1).

Таблица 4.1.

Таблица истинности мультиплексора.

E	a_1	a_0	Y
1	0	0	x_0
1	0	1	x_1
1	1	0	x_2
1	1	1	x_3
0	*	*	0

Согласуясь с таблицей истинности можно записать выражение:

$$Y = E * \overline{a_1} * \overline{a_0} * x_0 + E * \overline{a_1} * a_0 * x_1 + E * a_1 * \overline{a_0} * x_2 + E * a_1 * a_0 * x_3 \quad (4.1)$$

При любом значении адресующего кода все слагаемые, кроме одного, равны нулю. Ненулевое слагаемое равно x_i , где i — значение текущего адресного кода.

На основании выражения (4.1) комбинационная схема мультиплексора в базисе основного функционального набора имеет следующий вид (рис.4.1.)

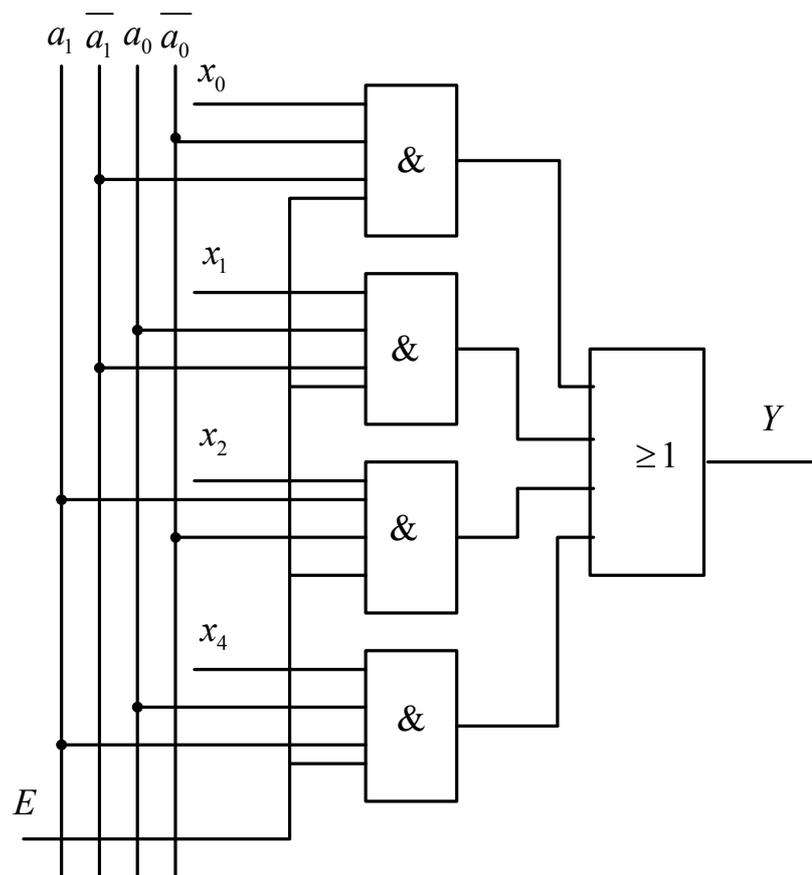


Рис. 4.1. Схема мультиплексора на 4 информационных входа с разрешающим входом E в базисе основного функционально-полного набора.

Выражение (4.2) можно преобразовать согласно теореме Шеннона в базис Шеффера («И-НЕ»).

$$Y = \overline{\overline{(E * \overline{a_1} * \overline{a_0} * x_0)} * \overline{(E * \overline{a_1} * a_0 * x_1)} * \overline{(E * a_1 * \overline{a_0} * x_2)} * \overline{(E * a_1 * a_0 * x_3)}} \quad (4.2)$$

На основании выражения (4.2) комбинационная схема мультиплексора в базисе функционального набора Шеффера («И-НЕ») имеет следующий вид (рис.4.2)

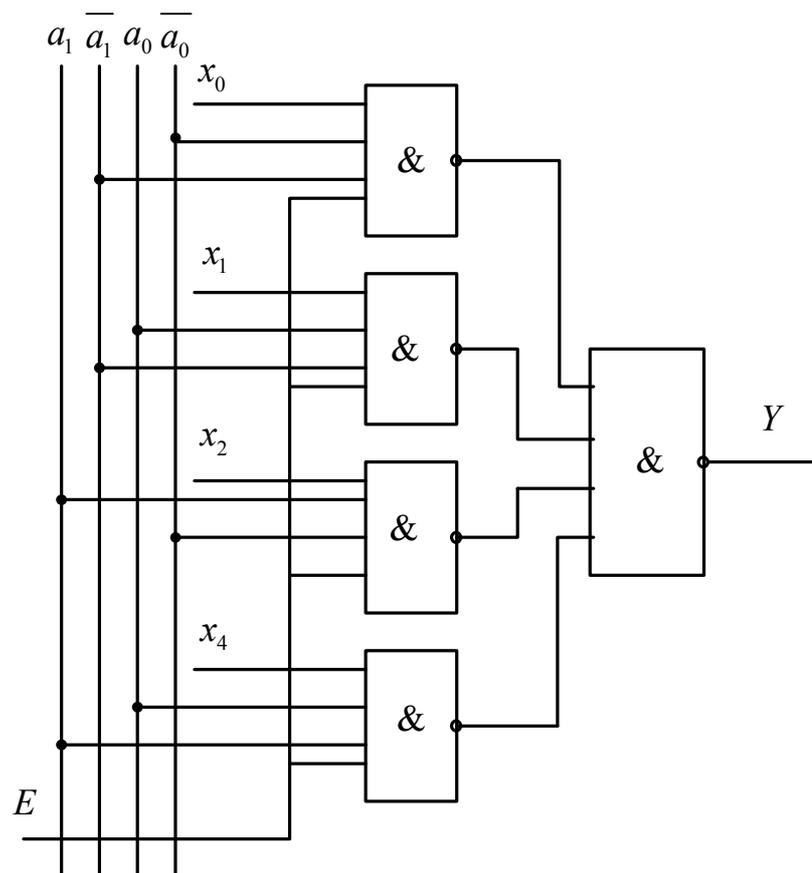


Рис. 4.2. Схема мультиплексора на 4 информационных входа с разрешающим входом E в базисе Шеффера.

При необходимости реализации количества информационных входов, превышающее возможности конкретной реализации мультиплексора, реализуют каскадные схемы соединения мультиплексоров.

Рассмотрим пример реализации управления 16-ю информационными входами, с помощью мультиплексоров, имеющих по 4 информационных входа. Для решения данной задачи необходимо 5 мультиплексоров на 4 информационных входа.

Схема соединения пяти мультиплексоров по 4 информационных входа для реализации мультиплексора по каскадной схеме представлена на рис. 4.3.

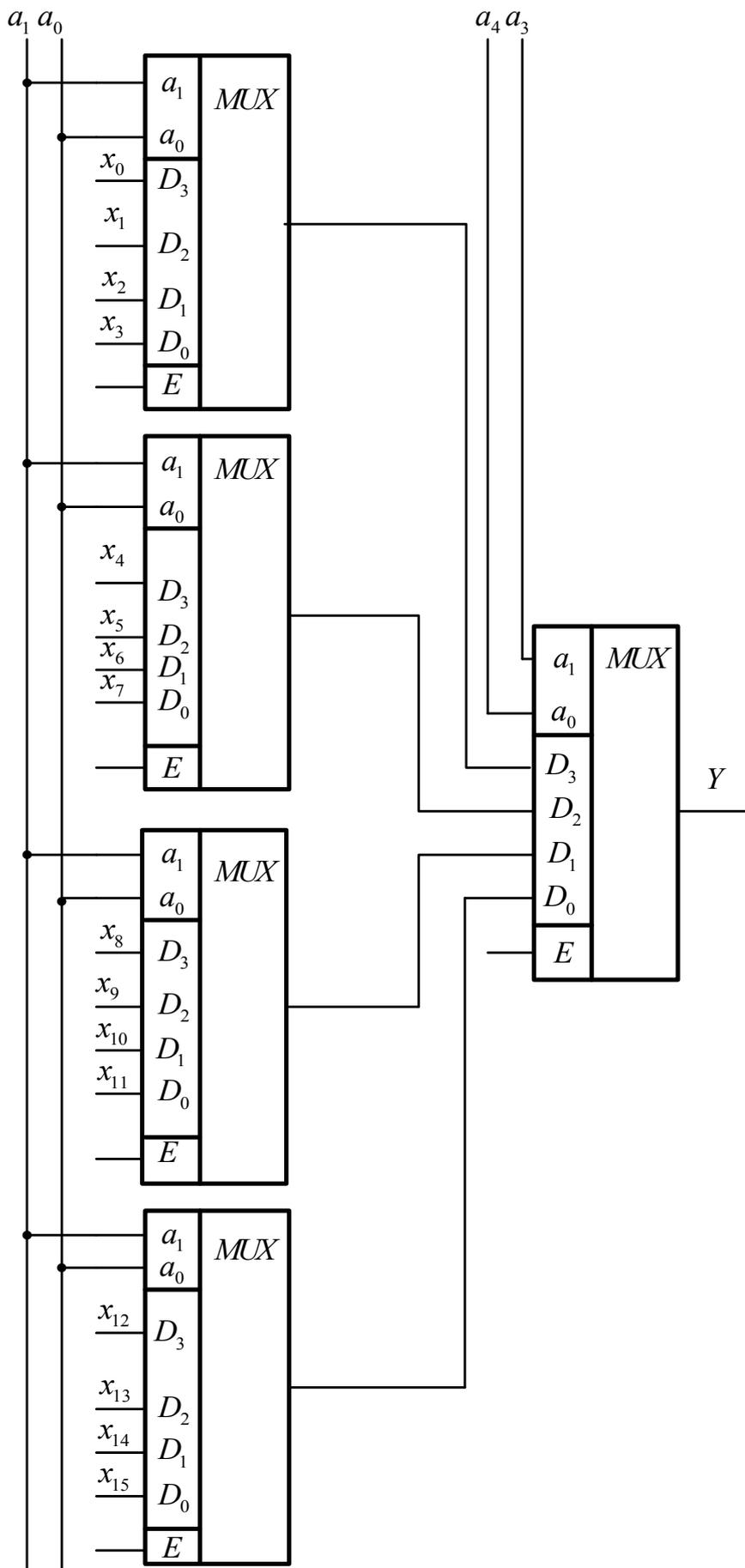


Рис. 4.3 Схема каскадного мультиплексора на 16 информационных входов.

Одним из распространённых применений мультиплексора является управление передачей кода в прямом или обратном виде в зависимости от управляющих сигналов. Для этого в каждом разряде на выходе регистра входы мультиплексора подключаются к прямому и инверсному выходу триггера регистра источника. Пример схемы, применённой для каждого разряда иллюстрирует рис.4.4

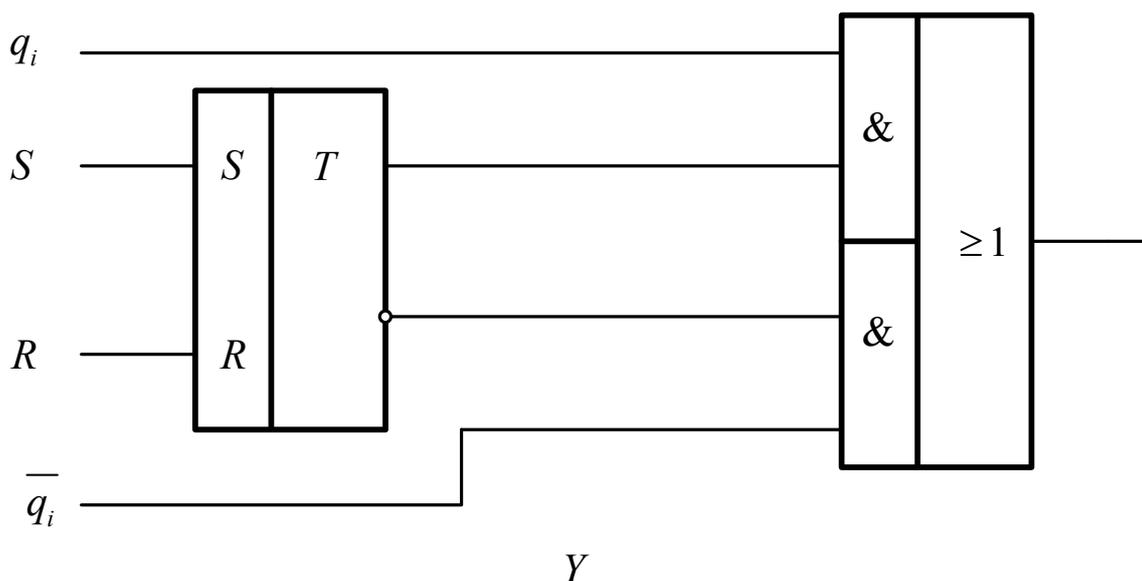


Рис.4.4 Мультиплексор в качестве инвертора кода слова.

Для демultipлексора существует соотношение зависимости количества информационных выходов от количества разрядов адреса. Это соотношение определяется формулой $m = 2^n$, где n – разрядность адреса, m – количество информационных выходов. Двоичный код на входах адреса определяет номер того информационного выхода, на который входная информация поступает в данный момент.

Тогда работа демultipлексора будет соответствовать таблице истинности (табл.4.2).

Таблица 4.2.

Таблица истинности демультиплексора.

X	a_1	a_0	E	Y_0	Y_1	Y_2	Y_3
x	0	0	1	x	0	0	0
x	0	0	1	0	x	0	0
x	1	0	1	0	0	x	0
x	1	1	1	0	0	0	x
x	*	*	0	0	0	0	0

Значения сигналов на выходе демультиплексора будет определяться из следующих выражений:

$$\begin{aligned} Y_0 &= x * \overline{a_1} * \overline{a_0} * E; \\ Y_1 &= x * \overline{a_1} * a_0 * E; \\ Y_2 &= x * a_1 * \overline{a_0} * E; \\ Y_3 &= x * a_1 * a_0 * E \end{aligned} \quad (4.3)$$

Исходя из выведенных выражений, синтезируется схема демультиплексора (рис. 4.5.)

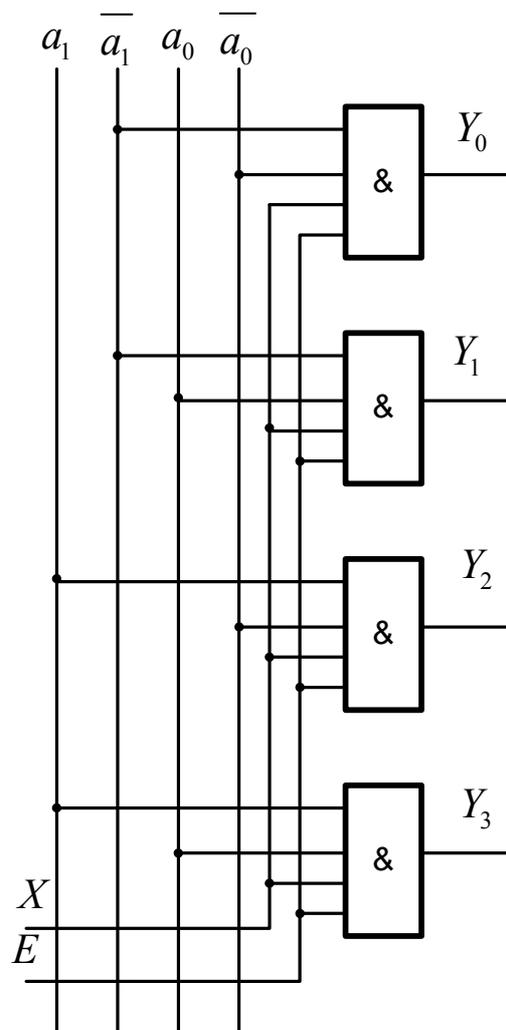


Рис.4.5 . Схема демультиплексора с четырьмя выходами и двумя разрядами адреса и одним информационным входом.

Аналогичный приём наращивание размерности демультиплексора применим при реализации каскадной схемы построения демультиплексора. Каскадный демультиплексор имеет 16 выходов и строится на основе пяти демультиплексоров по четыре выхода. Пространство разрядов адресов подразделяется на два множества: адреса управления входящим демультиплексором (младшие разряды) и адреса управления выходными демультиплексорами (старшие разряды). Управление для селекции работающего выходного демультиплексора основывается на использовании сигнала разрешения, который определяется начальным демультиплексором.

Схема каскадного демультиплексора приведена на рис. 4.6.

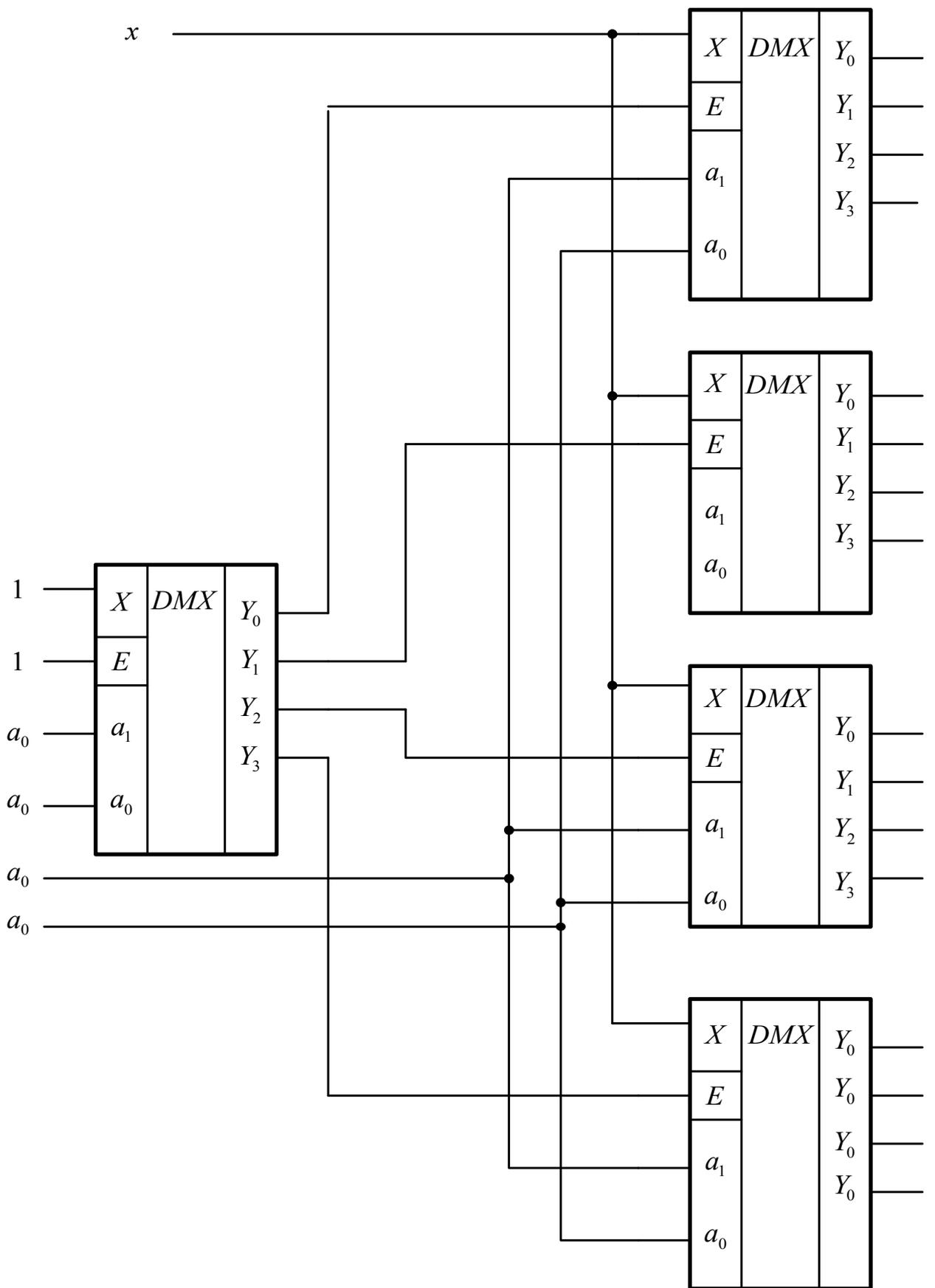


Рис. 4.6 . Схема каскадного демультиплектора на 16 информационных выходов.

4.2 Дешифраторы и шифраторы

Дешифратором называется кодирующее устройство, представляющее собой комбинационную логическую схему, преобразующее многоразрядный входной код в выходной (чаще всего в унарный) код.

Примером унарного кода является изображение десятичных чисел 0,1,2.....9.

Дешифраторы относят к кодирующим устройствам.

Если на входы дешифратора подаются двоичные переменные, то на одном из выходов дешифратора вырабатывается сигнал 1, а на остальных выходах сохраняются нули.

Число входов и выходов в так называемом полном дешифраторе связано соотношением $m = 2^n$,

где n — число входов, а m — число выходов.

Если в работе дешифратора используется неполное число выходов, то такой дешифратор называется неполным. Так, например, дешифратор, имеющий 4 входа и 16 выходов, будет полным, а если бы выходов было только 10, то он являлся бы неполным.

На выходах дешифратора вырабатываются значения булевых функций соответственно:

$$\begin{aligned} Y_0 &= \overline{x_3} * \overline{x_2} * \overline{x_1}; & Y_1 &= \overline{x_3} * \overline{x_2} * x_1; \\ Y_2 &= \overline{x_3} * x_2 * \overline{x_1}; & Y_3 &= \overline{x_3} * x_2 * x_1; \end{aligned} \quad (4.4)$$

Данные выражения позволяют построить линейный дешифратор на 4 выхода

рис.4.7.

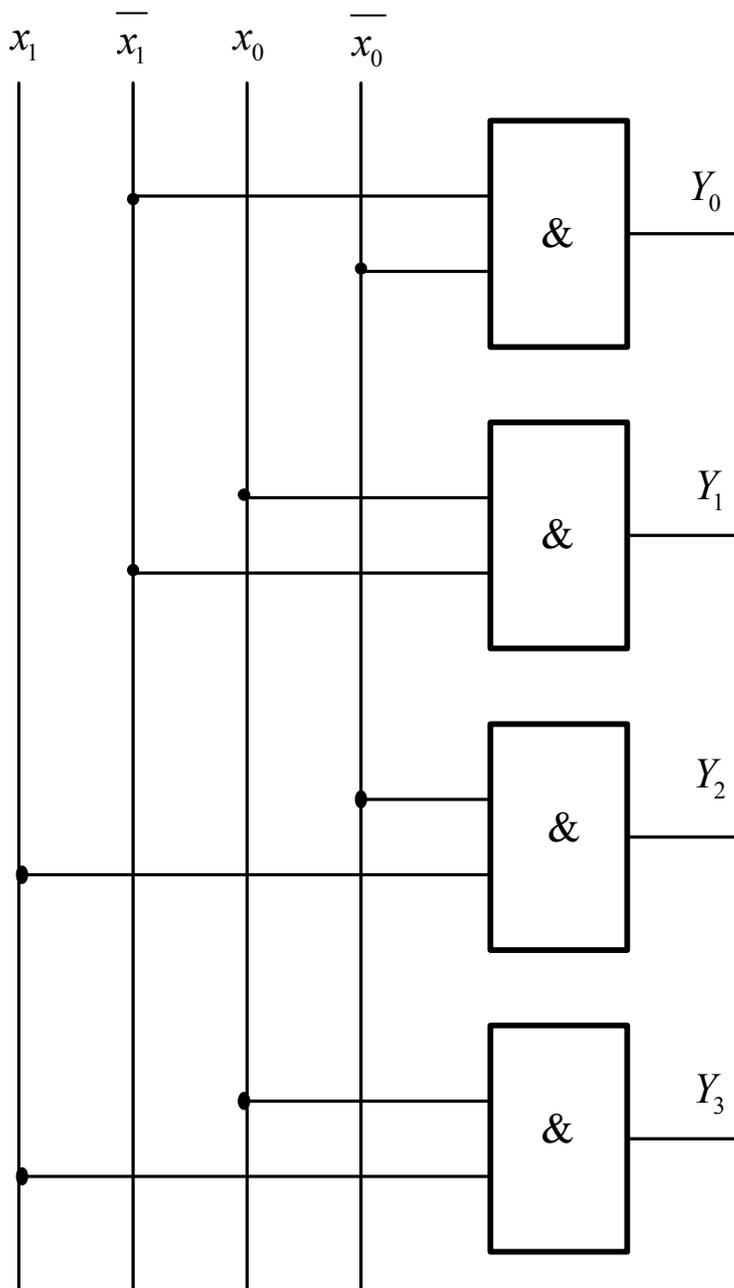


Рис. 4.7. Линейный дешифратор на 4 выхода.

Согласно ГОСТ 2.743-91 условно графическое обозначение дешифратора на 8 выходов изображено на рис. 4.8.

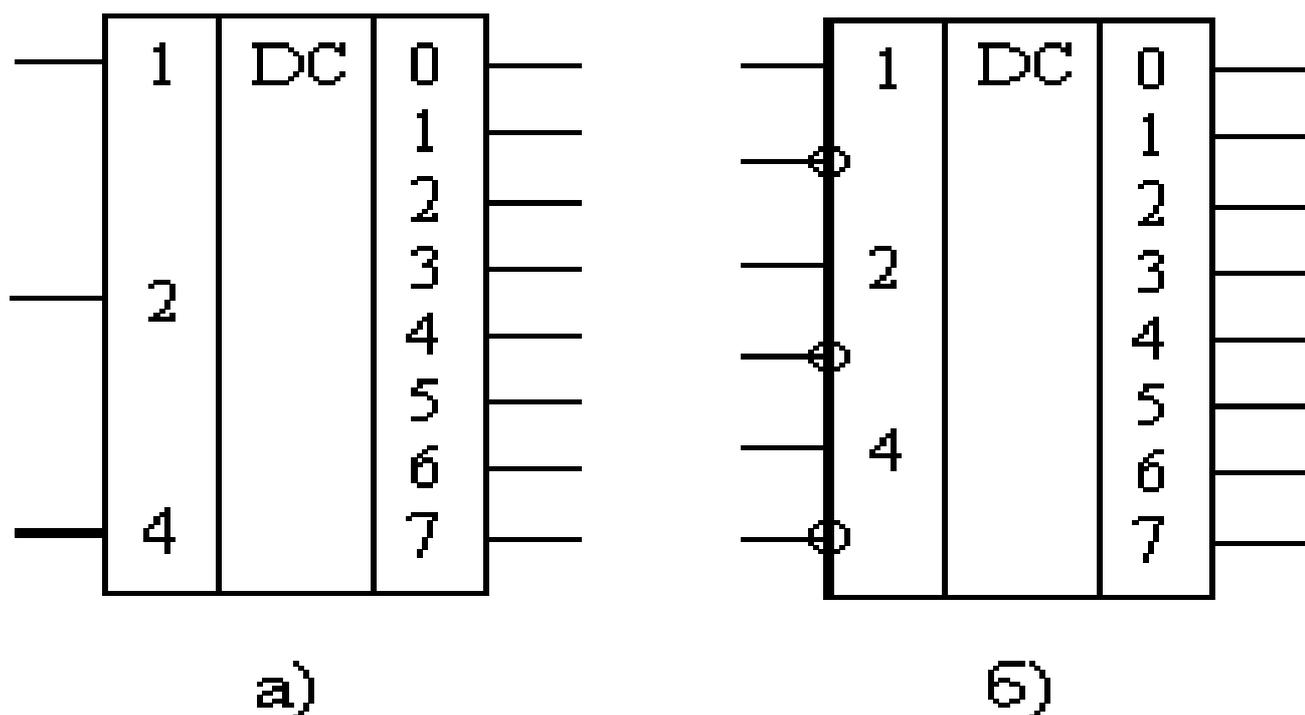


Рис. 4.8. Условно-графическое изображение дешифратора с прямыми входами а), с комбинацией прямых и инверсных входов б).

Шифратор — это комбинационное устройство, преобразующее унарный код (например, десятичные числа) в двоичную систему счисления, причем каждому входу может быть поставлено в соответствие десятичное число, а набор выходных логических сигналов соответствует определенному двоичному коду. Шифратор иногда называют «кодером» (от англ. coder) и используют, например, для перевода десятичных чисел, набранных на клавиатуре кнопочного пульта управления, в двоичные числа.

Если количество входов настолько велико, что в шифраторе используются все возможные комбинации сигналов на выходе, то такой шифратор называется полным, если не все, то неполным. Число входов и выходов в полном шифраторе связано соотношением $n = 2^m$, где n — число входов,

m — число выходов.

Так, для преобразования кода кнопочного пульта в четырехразрядное двоичное число достаточно использовать лишь 10 входов, в то время как полное число возможных входов будет равно 16, поэтому шифратор 10x4 (из 10 в 4) будет неполным.

Рассмотрим пример построения шифратора для преобразования десятиразрядного единичного кода (десятичных чисел от 0 до 9) в двоичный код. При этом предполагается, что сигнал, соответствующий логической единице, в каждый момент времени подается только на один вход. Таблица истинности такого шифратора представлена табл. 4.3.

Таблица 4.3.

Таблица истинности шифратора на 10 входов.

Код клавиши	Y_3	Y_2	Y_1	Y_0
K_0	0	0	0	0
K_1	0	0	0	1
K_2	0	0	1	0
K_3	0	0	1	1
K_4	0	1	0	0
K_5	0	1	0	1
K_6	0	1	1	0
K_7	0	1	1	1
K_8	1	0	0	0
K_9	1	0	0	1

Используя данную таблицу соответствия, запишем логические выражения, включая в логическую сумму те входные переменные, которые соответствуют единице некоторой выходной переменной. Так, на выходе Y_0 , будет логическая «1» тогда, когда логическая «1» будет или на входе K_1 , или K_3 , или K_5 , или K_7 , или K_9 , т. е.:

$$\begin{aligned} Y_0 &= K_1 + K_3 + K_5 + K_7 + K_9; \\ Y_1 &= K_2 + K_3 + K_6 + K_7; \\ Y_2 &= K_4 + K_5 + K_6 + K_7; \\ Y_3 &= K_8 + K_9 \end{aligned} \tag{4.5}$$

Данные булевы выражения для выходов шифратора позволяют синтезировать его схему рис.4.9.

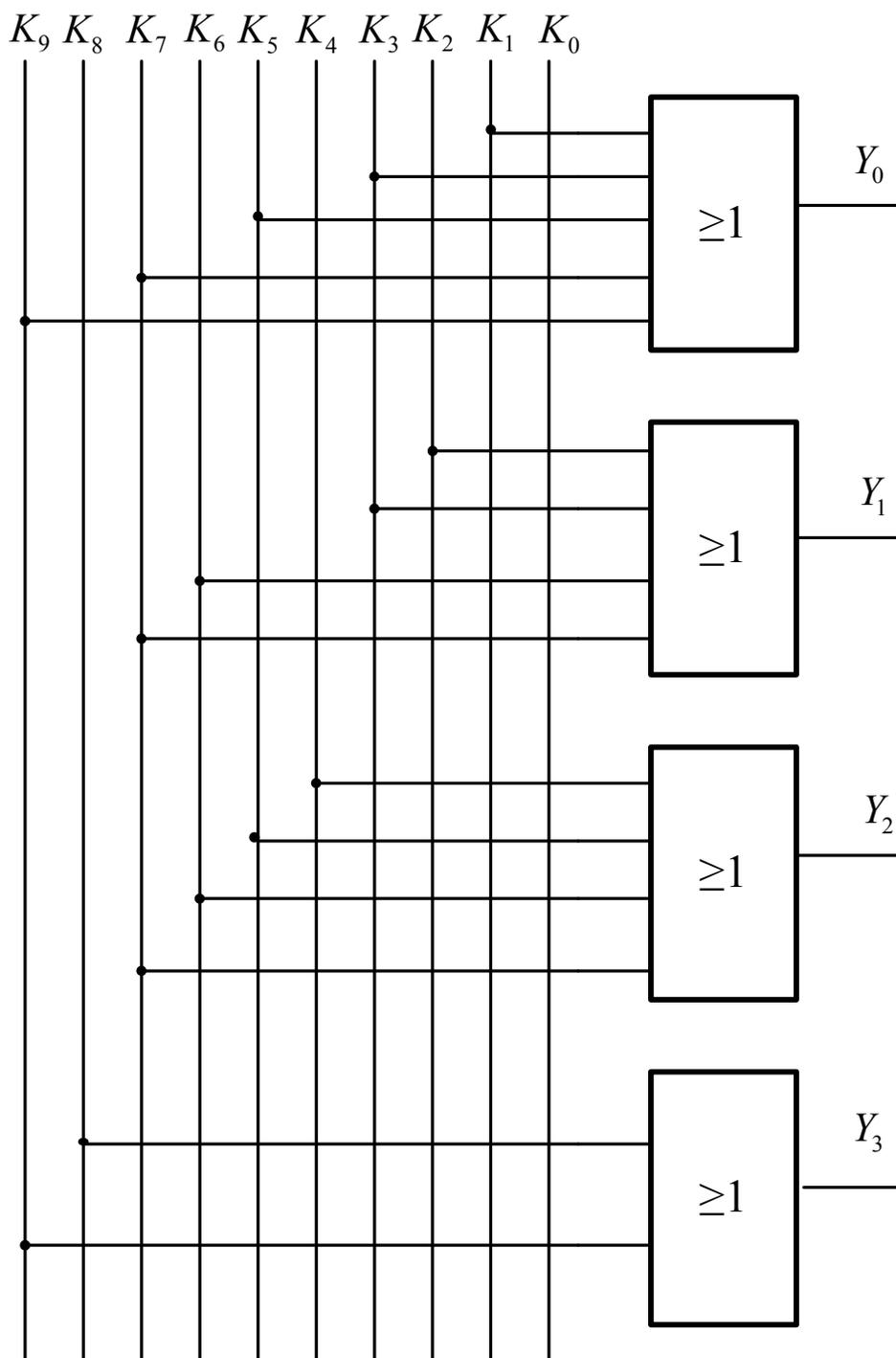


Рис. 4.9. Схема шифратора 10x4.

4.2. Типовые примеры проектных решений.

4.2.1 Пример использования мультиплексора в качестве универсальной комбинационной схемы.

Например, вместо синтеза комбинационной схемы по таблице истинности (табл. 3.1.) можно использовать мультиплексор с 8-ю информационными входами и тремя разрядами адресного кода. Технология создания заменяющего устройства состоит в подключении к адресным входам значений переменных, а на соответствующий вход мультиплексора подаётся значение либо логического нуля, либо единицы в зависимости от значения, принимаемого функцией при данном наборе трёх переменных (рис. 4.10.)

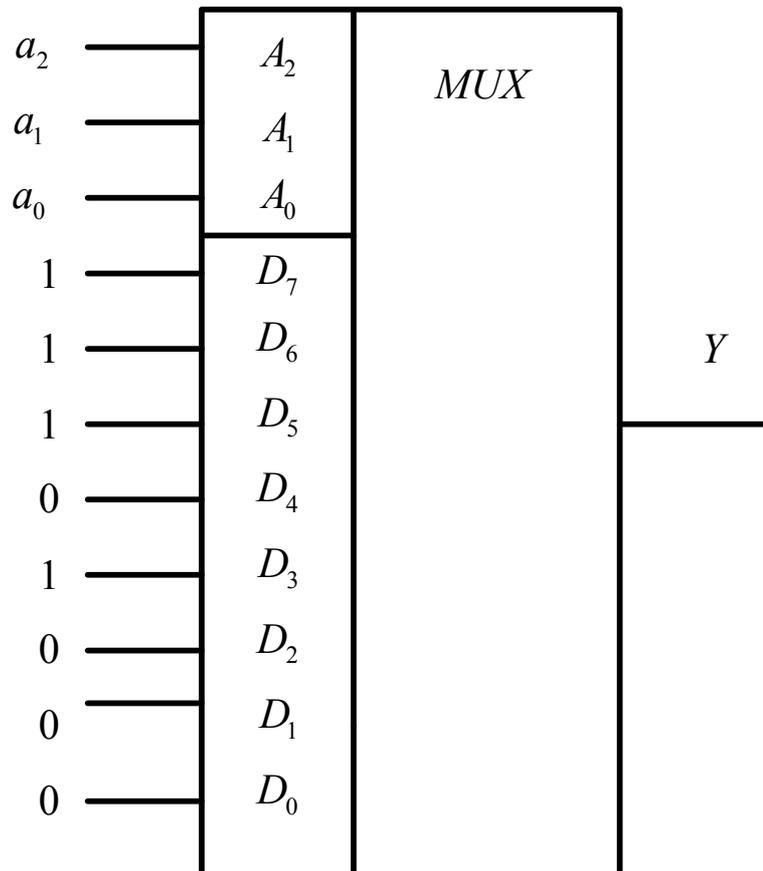


Рис. 4.10 . Мультиплексор в качестве универсальной комбинационной схемы.

4.2.2 Синтез шифратора для кода Грея.

Особенностью кода Грея является его большая устойчивость к ошибкам при передаче по линиям связи. При переходе от тетрады к тетраде в коде Грея всегда только один разряд меняется с 0 на 1 или с 1 на 0. То есть изменяется всегда только один бит в тетраде.

Коды такого типа получили название одношаговых кодов. Одношаговые коды также называются прогрессивными кодами.

Таблица 4.4 даёт соответствие четырёхразрядных комбинаций кода Грея десятичному (унарному) коду.

Таблица 4.4

Соответствие десятичного унарного кода и кода Грея

Десятичный Унарный код	a_3	a_2	a_1	a_0
0 K_0	0	0	0	0
1 K_1	0	0	0	1
2 K_2	0	0	1	1
3 K_3	0	0	1	0
4 K_4	0	1	1	0
5 K_5	0	1	1	1
6 K_6	0	1	0	1
7 K_7	0	1	0	0
8 K_8	1	1	0	0
9 K_9	1	1	0	1
10 K_{10}	1	1	1	1

Тогда для каждого выхода шифратора $a_0 \dots a_3$ можно записать булево выражение по данной таблице истинности.

$$a_0 = K_1 + K_2 + K_5 + K_6 + K_9 + K_{10} + K_{14}$$

$$a_1 = K_2 + K_3 + K_4 + K_5 + K_{10} + K_{11} + K_{12} + K_{13}$$

$$a_2 = K_4 + K_5 + K_6 + K_7 + K_8 + K_9 + K_{10} + K_{11} + K_{12} + K_{13}$$

$$a_3 = K_8 + K_9 + K_{10} + K_{11} + K_{12} + K_{13} + K_{14} + K_{15}$$

По приведённым булевым выражениям строится схема шифратора в код Грея (рис.4.10.)

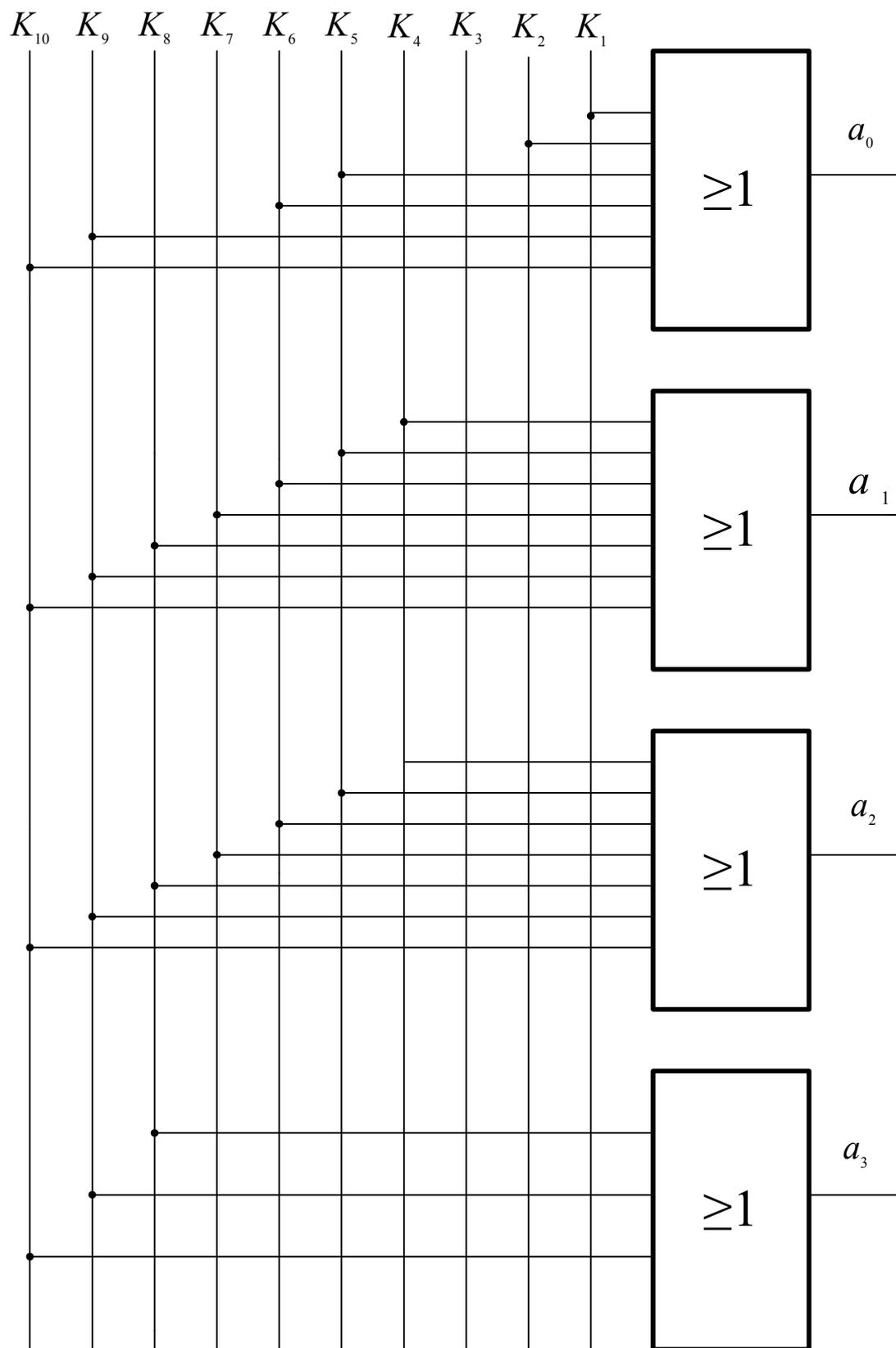


Рис. 4.10. Шифратор в код Грея.

Приобретаемые компетенции после изучения главы 4.

Помнить:

- определение и функциональное назначение дешифраторов, шифраторов, мультиплексоров, демультиплексоров.

Понимать:

- логику синтеза типовых схем дешифраторов, шифраторов, мультиплексоров, демультиплексоров на основе логических элементов.
- различать и понимать особенности в схемных решения перечисленных функциональных устройств.

Применять:

- дешифраторы, шифраторы, мультиплексоры, демультиплексоры для решения задач связи, селектирования сигналов и линий передачи двоичных кодов.
- строить схемы на основе мультиплексоров, замещающие типовые комбинационные схемы, построенные на основе таблиц истинности функций, описывающих дисциплину работы.

Оценивать:

- возможности применения дешифраторов, шифраторов, мультиплексоров, демультиплексоров для построения узлов с другим функциональным назначением.

4.3 Контрольные вопросы.

Вопросы категории 1. «Помнить»

4.3.1 Напишите формулу зависимости количества выходов дешифратора от имеющегося количества разрядов входного двоичного кода.

4.3.2 На основе, каких логических элементов может быть построен дешифратор?

Вопросы категории 2. «Понимать»

4.3.3 Сформулируйте различие в схемах построения полного и неполного линейного дешифратора.

4.3.4 На каком схемном принципе строится каскадный дешифратор и чем вызвана необходимость в такой архитектуре дешифратора?

4.3.5 На каком схемном принципе строится матричный дешифратор и чем вызвана необходимость в такой архитектуре дешифратора?

4.3.6 Поясните разницу в архитектуре полного и неполного шифратора.

4.3.7 Какое назначение у адресных входов мультиплексора?

Вопросы категории 3. «Применять»

4.3.8 На основе, каких логических элементов может быть построен шифратор?

4.3.9 Какие логические элементы входят в состав мультиплексоров?

4.3.10 Какая цель достигается при каскадировании мультиплексоров?

Ответы на контрольные вопросы.

4.3.1 Число входов и выходов в так называемом полном дешифраторе связано соотношением $m = 2^n$, где n — число входов, а m — число выходов.

4.3.2 Дешифратор может быть построен на основе логических элементов «И».

4.3.3 Если в работе дешифратора используется неполное число выходов, то такой дешифратор называется неполным.

4.3.4 Каскадный дешифратор строится на основе нескольких дешифраторов меньшей размерности. Для определения дешифратора, с которого будет выдан выходной сигнал, зависит от кодов старших разрядов, подаваемых на дешифратор старших разрядов входного кода.

4.3.5 Матричный дешифратор, как и каскадный позволяет использовать дешифраторы, имеющие меньшее количество входных разрядов кода вместо линейного дешифратора увеличенной размерности. В схемах матричного дешифратора потребуется некоторое количество логических элементов «И», определяемое количеством выходов итогового дешифратора.

4.3.6 В отличие полного шифратора, неполный шифратор имеет ограниченное количество значений выходных кодов.

4.3.7 Адресные входы мультиплексора воспринимают значения номера входа информационного сигнала.

4.3.8 Шифратор может быть построен на основе логических элементов «ИЛИ».

4.3.9 В состав мультиплексора входят логические элементы «И», «ИЛИ», «НЕ».

4.3.10 Каскадирование мультиплексоров позволяет построить составной мультиплексор из мультиплексоров с меньшим количеством информационных входов.

4.4. Задачи для самостоятельного решения.

4.4.1 Спроектировать схему преобразователя двоичного кода в код Грея, используя дешифратор и шифратор.

Решение:

Сформируем таблицу соответствия двоичного кода коду Грея (табл. 4.5).

Таблица 4.5. Таблица соответствия двоичного кода коду Грея.

Двоичный код				Код Грея			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0

1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Применим дешифратор на 16 выходов и шифратор на 16 входов.

Далее по данной таблице осуществим коммутацию выходов дешифратора, руководствуясь, что стандартный шифратор при подаче сигнала на его один из входов даст значение двоичного кода. Рассмотрим последнюю строчку данной таблицы. Двоичному коду 1111 соответствует код Грея 1000, значит, если последний выход дешифратора соединить с восьмым (по десятичному счислению) входом шифратора, на его выходе будет установлен код 1000.

Схема коммутации представлена на рис. 4.11.

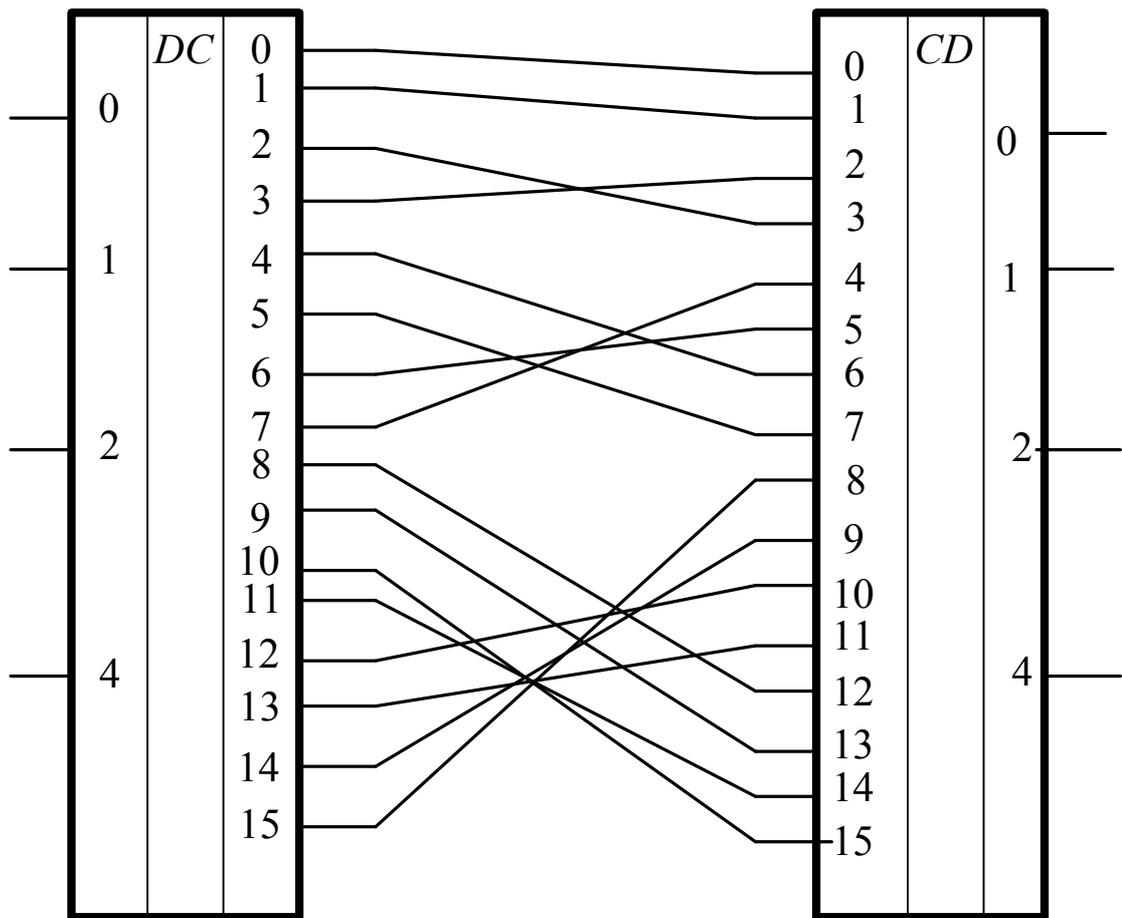


Рис.4.11. Схема преобразователя двоичного кода в код Грея.

4.4.2 Доказать, что шифратор может быть построен и на логических элементах «ИЛИ» и на логических элементах «И-НЕ».

Решение:

Выражение для разрядов шифратора в базисе «И-ИЛИ-НЕ» преобразуется по формуле Де Моргана в базис Шеффера.

$$Y_0 = K_1 + K_3 + K_5 + K_7 + K_9;$$

$$Y_1 = K_2 + K_3 + K_6 + K_7;$$

$$Y_2 = K_4 + K_5 + K_6 + K_7;$$

$$Y_3 = K_8 + K_9$$

4.4.3 Как надо видоизменить схему линейного дешифратора, чтобы получить демультиплексор?

Решение:

В схему дешифратора добавляется сигнал разрешения на все элементы «И».

Этот вход и будет являться информационным входом демультиплексора, а остальные входы будут адресными входами.

Глава 5. Триггеры.

5.1. Назначение, классификация и типовое устройство триггеров.

Триггером называют логическую схему с положительной обратной связью, имеющую два устойчивых состояния.

История изобретения триггера относится к началу 20 века. Первая схема триггера была реализована на октальных радиолампах и имела название «катодное реле». Это изобретение принадлежит М.А.Бонч-Бруевичу, известному советскому радиотехнику, через два года стало известно об У. Г. Икклизе (англ.) и Ф. У. Джордане (англ.), получившим патент на схему триггера в Великобритании.

Триггер содержит элемент памяти (собственно триггер) и схему управления, выполненную, как правило, с помощью комбинационной схемы

(КС). Схема управления преобразует поступающую на ее входы информацию (x_1, x_2, \dots, x_m) в комбинацию сигналов, действующих непосредственно на входы собственно триггера.

Триггеры принадлежат к классу устройств, именуемых цифровыми автоматами (ЦА). В отличие от комбинационной схемы (КС), цифровой автомат содержит входную комбинационную схему и элемент памяти (запоминающую ячейку (ЗЯ)). Функциональную схему цифрового автомата можно иллюстрирует рис. 5.1.

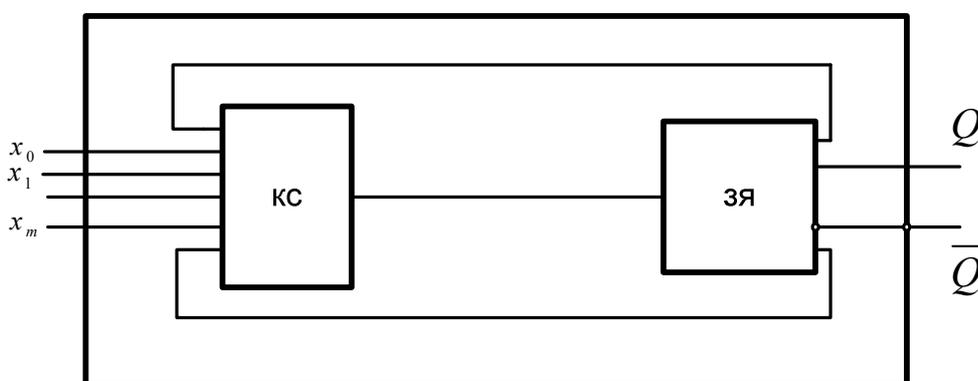


Рис. 5.1. Функциональная схема цифрового автомата.

Значение сигнала на выходе цифрового автомата в общем виде можно выразить следующим выражением:

$$Q = f(x_1, x_2, x_3, \dots, x_m, S(t))$$

где $x_1, x_2, x_3, \dots, x_m$ - множество входных сигналов, $S(t)$ - состояние запоминающей ячейки в момент подачи новой комбинации входных сигналов.

Следовательно, в отличие от определения комбинационной схемы, определение цифрового автомата можно сформулировать следующим образом:

Значения сигналов на выходе цифрового автомата зависят не только от множества сигналов на его входе, но и от состояния в котором находилась запоминающая ячейка (ЗЯ) цифрового автомата.

Схема управления преобразует поступающую на ее входы информацию (x_1, x_2, \dots, x_m) в комбинацию сигналов, действующих непосредственно на входы собственно триггера. При этом информационные входы триггера отождествляются с входными переменными и имеют следующие обозначения:

S (от английского Set - установка) - вход для асинхронной установки триггера в состояние «1» (S - вход);

R (от английского Reset - сброс) - вход для асинхронной установки триггера в состояние «0» (R - вход);

D (от английского Delay - задержка) - информационный вход для установки триггера в состояние «1» или «0» (D - вход);

T (от английского Toggle - кувырнуться) - счетный вход (T - вход);

J - вход для синхронной установки состояния «1» в универсальном J-K - триггере (J - вход);

K - вход для синхронной установки состояния «0» в универсальном J-K - триггере (K - вход);

C - вход синхронизации (C - вход).

Асинхронные входы R и S могут быть прямыми и инверсными. Инверсные входы обозначаются знаком инверсии. Для инверсных входов активным сигналом является уровень логического "0" (низкий уровень). На уровень логической "1" (высокий уровень) триггер не реагирует. Для прямых входов наоборот активным сигналом является уровень логической "1".

Выход триггера принято отождествлять с его внутренним состоянием и обозначать символом Q . Подавляющее число схем триггеров имеет два выхода: прямой и инверсный \bar{Q} . В установившемся состоянии всегда, если $Q=1$, то $\bar{Q}=0$ и, если $Q=0$, то $\bar{Q}=1$. При этом считается, что триггер находится в состоянии "1", если на прямом выходе Q имеется высокий уровень напряжения ($Q=1, \bar{Q}=0$), и в состоянии "0", если на прямом выходе имеется низкий уровень напряжения ($Q=0, \bar{Q}=1$).

Обозначение триггеров на функциональных схемах.

Примеры условного обозначения триггеров на функциональных схемах в соответствии с ГОСТ 2.743-91 (ЕСКД) показаны на рис.1.

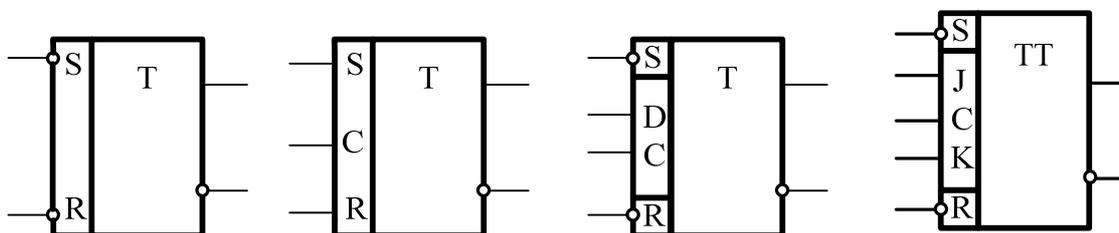


Рис. 5.2. Условные обозначения триггеров.

Если триггер содержит входную логику, управляющую процессом занесения в него информации, то в прямоугольнике, условно изображающем триггер, отделяется чертой левее дополнительное поле, а в нем показывается вход "С" синхронизирующего сигнала и отмечаются функциональные назначения информационных входных сигналов x_1, x_2, \dots, x_m .

В основном поле прямоугольника ставится символ T для обозначения триггера. Дополнительное поле может быть разделено на две части: асинхронную и синхронную. В первой проставляются символы S и R входов асинхронной установки триггеров в «1» и «0», во второй на местах x_1, x_2, \dots, x_m - символы, относящие данный триггер к тому или иному функциональному типу.

Задание законов функционирования триггеров.

Законы функционирования конечных автоматов вообще, и в частности, триггеров могут быть заданы различными способами. Часто такой закон задают таблицей истинности. Таблица истинности отражает процесс перехода триггера из одного устойчивого состояния в другое, и поэтому ее чаще называют таблицей переходов. В таблице переходов содержатся значения информационных и синхронизирующих сигналов на входе триггера, а также значения выходных сигналов (внутренних состояний триггера) после окончания действия синхронизирующего сигнала. Закон функционирования

триггера может быть задан и в виде характеристического уравнения логической функции вида

$$Q^{t+1} = f(Q^t, X_i^t), \quad i = 1, 2, \dots, m \quad (5.1)$$

где Q^{t+1} – состояние триггера после окончания действия синхронизирующего сигнала,

Q^t – состояние триггера до прихода синхронизирующего сигнала,

X^t – значение сигнала на информационном входе в момент времени “ t ”.

Характеристическое уравнение связывает логические переменные на входах и выходах триггера до и после его перехода из одного состояния в другое.

Между таблицей переходов и характеристическим уравнением существует взаимно однозначное соответствие, т.е. от таблицы переходов всегда можно перейти к характеристическому уравнению и наоборот.

Классификация триггеров.

В настоящее время в интегральной микросхемотехнике наиболее распространенными являются триггеры и логические элементы потенциального типа. В основу классификации этих триггеров положены два основных признака:

1) *Функциональный* – этот признак определяет назначение триггера и в ряде случаев является решающим при выборе типа триггера для проектируемого вычислительного устройства или узла. По указанному признаку различают триггеры RS-, D-, T-, JK- и др. типов.

2) *Способ записи информации в триггер* – этот признак характеризует способ записи информации и временную диаграмму работы триггера, т.е. определяет ход процесса записи информации в триггер. По этому признаку триггеры подразделяются на две группы:

а) асинхронные;

б) синхронные;

Запись информации в асинхронный триггер осуществляется в произвольный момент времени непосредственно с поступлением информационного сигнала на вход триггера.

Синхронные триггеры помимо информационных входов содержат один или несколько синхронизирующих входов (вход «С» на рис.1). Запись информации в такие триггеры осуществляется только при подаче синхронизирующего импульса (СИ). В свою очередь, синхронные триггеры подразделяются на триггеры, работающие по уровню СИ (без задержки) и на триггеры с внутренней задержкой. В первых срабатывание происходит одновременно с поступлением СИ, а во-вторых – после окончания действия СИ.

Схемотехнические параметры триггеров.

Схемотехнические параметры триггеров являются определяющими для расчёта реальных времён переключений (срабатываний) триггеров и оценки задержки появления нового значения сигнала на его выходе.

Наиболее важными являются следующие:

1. Нагрузочная способность – характеристика, определяющая количество входных каскадов устройств, подключаемых к выходу триггера при сохранении его работоспособности и сохранения параметров работы в допустимом диапазоне значений,

2. Коэффициент объединения по входу,

3. Быстродействие, характеризующиеся следующими параметрами:

- задержкой переключения триггера t_t ,

- минимально допустимой длительностью сигнала записи информации в триггер t_i ,

- рабочей частотой и максимальной частотой переключения триггера f_p и f_m .

Специфические параметры триггеров:

- разрешающее время триггера $t_{разр}$, представляет наименьший интервал времени между входными сигналами минимальной длительности, вызывающие переключение триггера. Тогда максимально допустимая частота работы триггера определяется по формуле

$$f_{\max} = 1/t_{разр}$$

Аномальные состояния триггеров.

Аномалии – это в большинстве случаев неустранимые явления, вызванные разбросом параметров полупроводниковых компонентов из которых собирается либо формируется по интегральной технологии схема того или иного триггера. Как правило, аномалии возникают при работе с асинхронными сигналами. Аномалии бывают метастабильные, когда выходные напряжения на выходах триггеров близки к пороговым, и колебательные, когда напряжение выходного сигнала триггера осуществляет колебания вокруг пороговых значений.

5.2 Триггеры RS, D, JK, T типа.

Триггеры RS – типа.

Асинхронный RS-триггер

Асинхронным триггером RS-типа (RS -триггером) называется логическое устройство с двумя устойчивыми состояниями, имеющее два информационных входа R и S , такие что, при $S = 1$ и $R = 0$ триггер принимает состояние 1 ($Q = 1$), а при $R = 1$ и $S = 0$ триггер принимает состояние 0 ($Q = 0$). Закон функционирования RS – триггера с прямыми входами отображен в таблице переходов, табл.5.1

Таблица 5.1

Таблица истинности RS – триггера.

R^t	S^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1

0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	*
1	1	1	*

При значениях $R^t=S^t=1$ состояние триггера является неопределенным. Перенеся на карту Карно (рис.2.) таблицу переходов RS-триггера и доопределив значения Q^{t+1} единицами можно получить минимизированную дизъюнктивную нормальную форму (МДНФ) характеристического уравнения RS-триггера.

$Q^t \backslash R^t S^t$	00	01	11	10
0		1	*	
1	1	1	*	

Рис. 5.3 Карта Карно асинхронного RS – триггера.

$$Q^{t+1} = S^t + \bar{R}^t Q^t \quad (5.2)$$

Исходя из характеристического уравнения триггера, его можно построить на логических элементах «И-НЕ», «ИЛИ-НЕ» и других. Чтобы получить структурное уравнение триггера, необходимо произвести двойное инвертирование характеристического уравнения.

$$Q^{t+1} = \overline{\overline{S^t * \overline{\overline{R^t Q^t}}}} \quad (5.3)$$

Реализация этого уравнения на элементах «И-НЕ» представлена на рис. 5.4, а условное графическое обозначение (УГО) на рис. 5.5

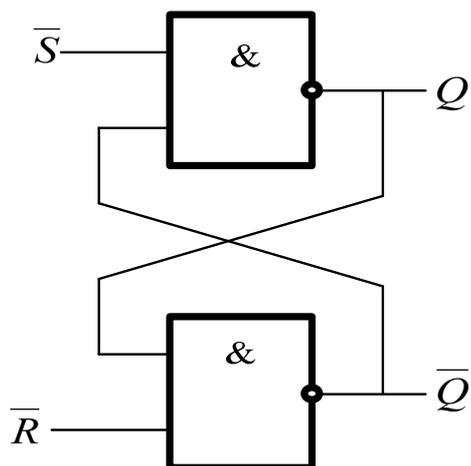


Рис. 5.4 Асинхронный R-S триггер с инверсными входами на элементах «И-НЕ».

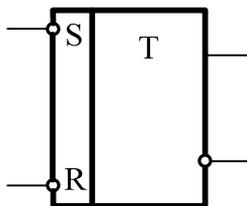


Рис. 5.5 Условно-графическое обозначение асинхронного R-S триггера с инверсными входами.

Для такого триггера активным сигналом является уровень логического 0, а запрещенной комбинацией $R^t=S^t=0$, при которой состояние триггера будет неопределенным.

МКНФ характеристического уравнения RS-триггера в соответствии с картой Карно (рис.5.6.) имеет следующий вид

$Q^t \backslash R^t S^t$	00	01	11	10
0	0		*	0
1			*	0

Рис.5.6 Карта Карно для RS – триггера с прямыми входами.

$$Q^{t+1} = \overline{R^t} * (S^t + Q^t) \quad (5.4)$$

После инвертирования уравнения (4) получится структурное уравнение для реализации RS-триггера с прямыми входами на элементах «ИЛИ-НЕ».

$$Q^{t+1} = \overline{\overline{R^t + (S^t + Q^t)}} \quad (5.5)$$

Реализация уравнения (5) на элементах «ИЛИ-НЕ» показана на рис.5.7, а УГО асинхронного RS-триггера с инверсными входами на рис.5.8

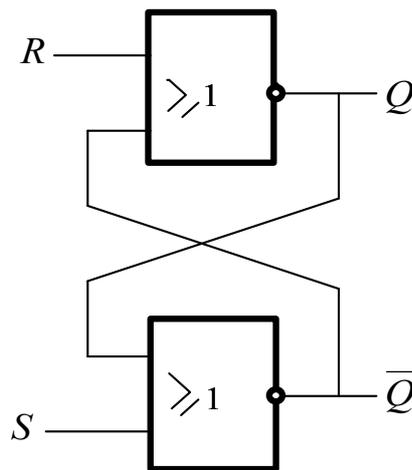


Рис.5.7. RS-триггер с прямыми входами на элементах «ИЛИ-НЕ».

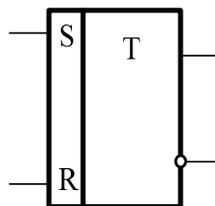


Рис.5.8. Условно-графическое обозначение асинхронного R-S триггера с прямыми входами.

Временная диаграмма асинхронного R-S триггера представлена на рис. 5.9.

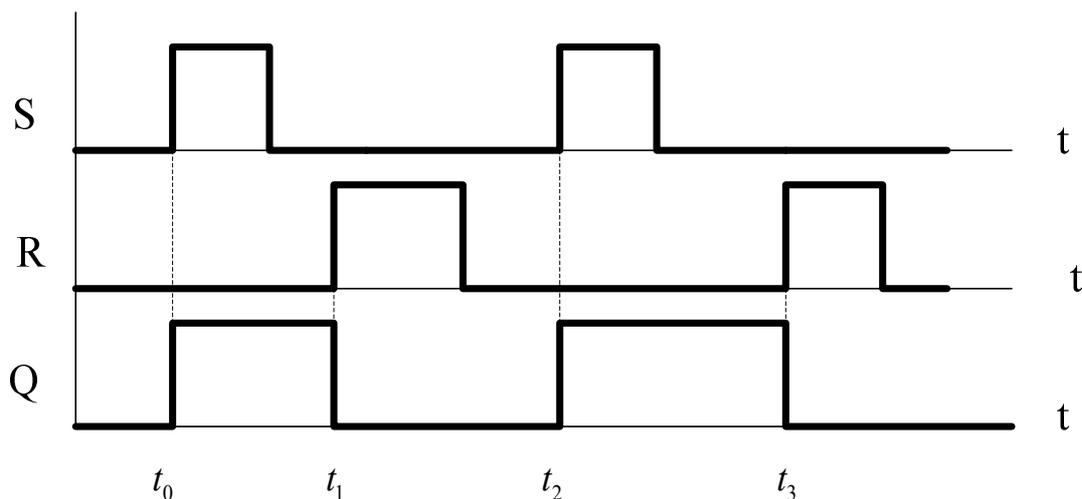


Рис. 5.9 Временная диаграмма асинхронного R-S триггера.

Синхронный R-S -триггер с прямыми входами.

В отличие от асинхронного этот триггер на каждом информационном входе имеет дополнительные схемы совпадения, первые входы которых объединены и на них подаются синхронизирующие сигналы. Вторые входы схем совпадения являются информационными. Таким образом, наличие схем совпадения определяет то обстоятельство, что триггер будет срабатывать от сигналов R и S только при наличии синхронизирующего импульса.

При добавлении синхросигнала таблица переходов синхронного RS-триггера имеет следующий вид (табл. 5.2).

Таблица 5.2. Таблица переходов синхронного

RS-триггера

C^t	S^t	R^t	Q^t	Q^{t+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1

1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	*
1	1	1	1	*

Карта Карно для логической функции Q^{t+1} имеет следующий вид, рис.5.10

$C^t Q^t \backslash R^t S^t$	00	01	11	10
00				
01	1	1	1	1
11	1	1	*	
10		1	*	

Рис.5.10 Карта Карно синхронного RS- триггера для логической функции выходного сигнала Q^{t+1}

Доопределяя функцию Q^{t+1} единицами, МДНФ характеристического уравнения синхронного RS-триггера примет вид:

$$Q^{t+1} = \overline{C^t} + Q^t \overline{R^t} + C^t S^t \quad (5.6)$$

При построении схемы синхронного RS-триггера на элементах «И-НЕ» МДНФ его характеристическое уравнение преобразуется в

структурное (7).

$$Q^{t+1} = \overline{\overline{S^t C^t} * Q^t * R^t C^t} \quad (5.7)$$

Реализация структурного уравнения синхронного RS-триггера с прямыми входами на элементах «И-НЕ» представлена на рис.5.11, а его УГО на рис. 5.12.

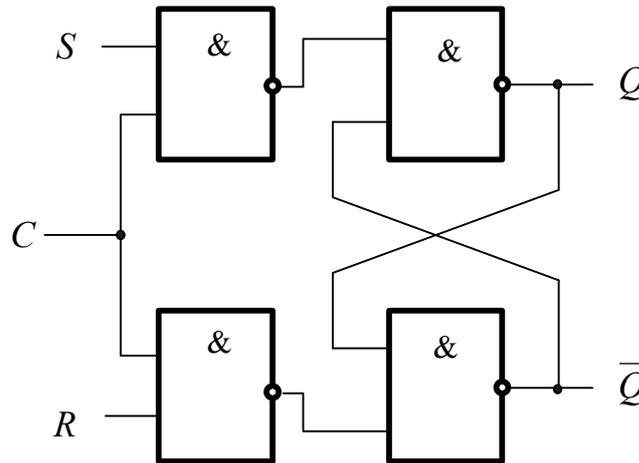


Рис. 5.11 Синхронный RS-триггер с прямыми входами на элементах «И-НЕ».

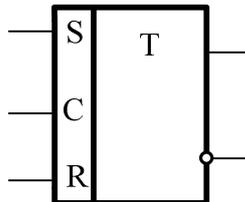


Рис. 5.12. Условно-графическое изображение синхронного RS-триггера.

Временная диаграмма синхронного RS-триггера представлена на рис. 5.13.

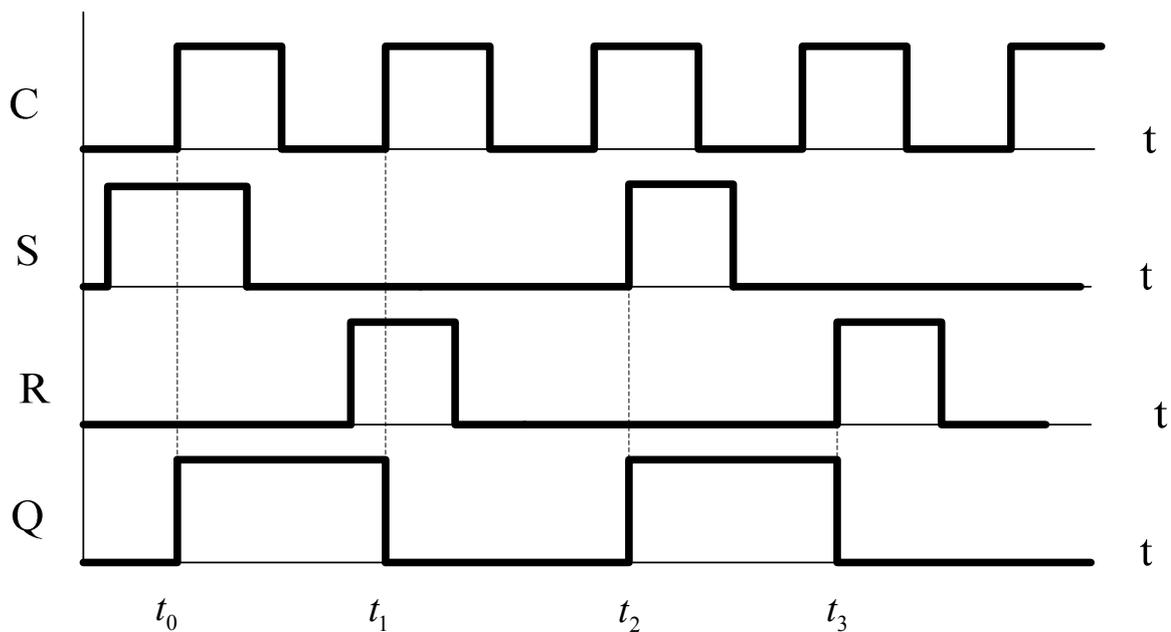


Рис. 5.13. Временная диаграмма синхронного RS-триггера.

Двухступенчатый (типа MS) синхронный RS-триггер.

Устойчивая работа одноступенчатых RS-триггеров в произвольной схеме возможна только в случае, если занесение в триггер информации осуществляется после завершения передачи информации о прежнем его состоянии в другой триггер. Такую работу триггера можно обеспечить при использовании двух серий находящихся в противофазе сигналов, используя свойство внутренней задержки. В триггерах с внутренней задержкой новая информация на выходе устанавливается только после окончания действия синхронизирующего импульса.

Подобный принцип обмена информацией реализован в двухступенчатых RS-триггерах. Такой тип триггеров получил ещё альтернативное наименование: – триггер с задержкой. Простейшая схема двухступенчатого RS-триггера с управляющим инвертором показана на рис. 5.14, а его УГО на рис. 5.15.

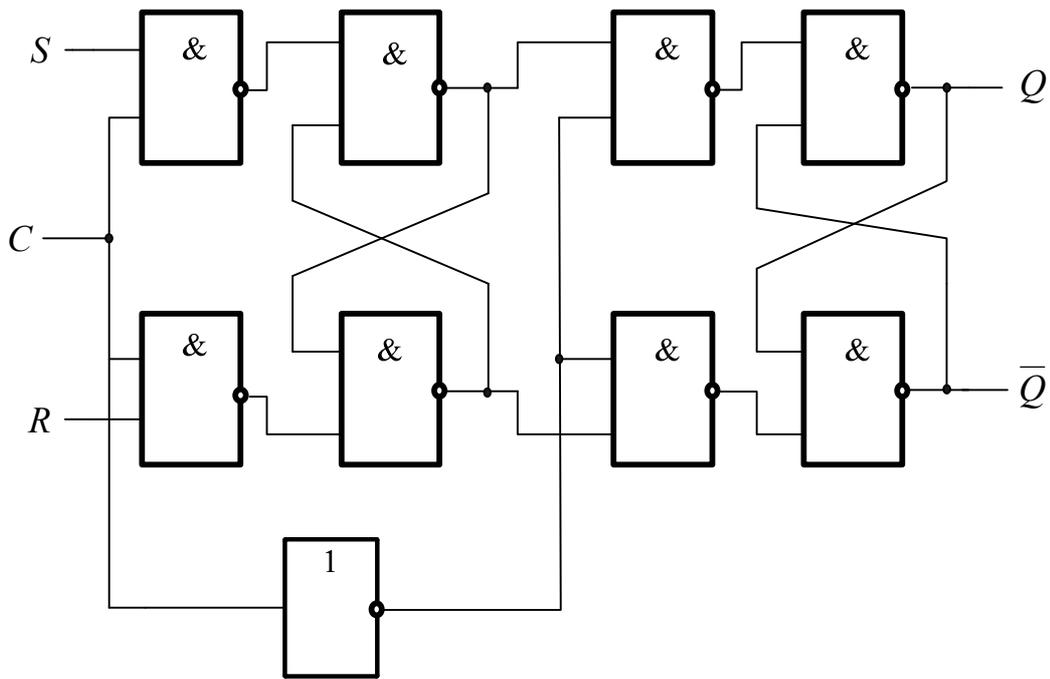


Рис. 5.14 Схема двухступенчатого синхронного RS-триггера.

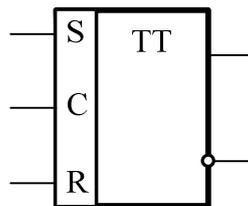


Рис.5.15 Условно-графическое изображение двухступенчатого RS-триггера.

Схема двухступенчатого RS-триггера состоит из двух одноступенчатых RS-триггеров и инвертора в цепи синхронизации. При поступлении единичного синхросигнала входная информация заносится в первый одноступенчатый RS-триггер. Второй RS-триггер при этом будет хранить информацию, относящуюся к предыдущему такту.

По окончании действия синхросигнала, когда $C=0$, а $\bar{C}=1$, первый триггер перейдет в режим хранения, а второй перепишет из него новое значение выходного сигнала.

Двухступенчатый триггер изменяет свои состояния только после окончания действия сигнала синхронизации. Поэтому из двухступенчатых триггеров можно строить произвольные схемы, в том числе подавать сигналы с выхода триггера на его вход.

На схеме показаны асинхронные R-, S- входы. Они являются приоритетными, т.е. при подаче на них активных информационных сигналов - синхронные R-, S- входы блокируются. Для работы триггера в синхронизируемом режиме на асинхронные входы должны быть поданы единичные сигналы.

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

Временная диаграмма двухступенчатого RS-триггера представлена на рис. 5.16.

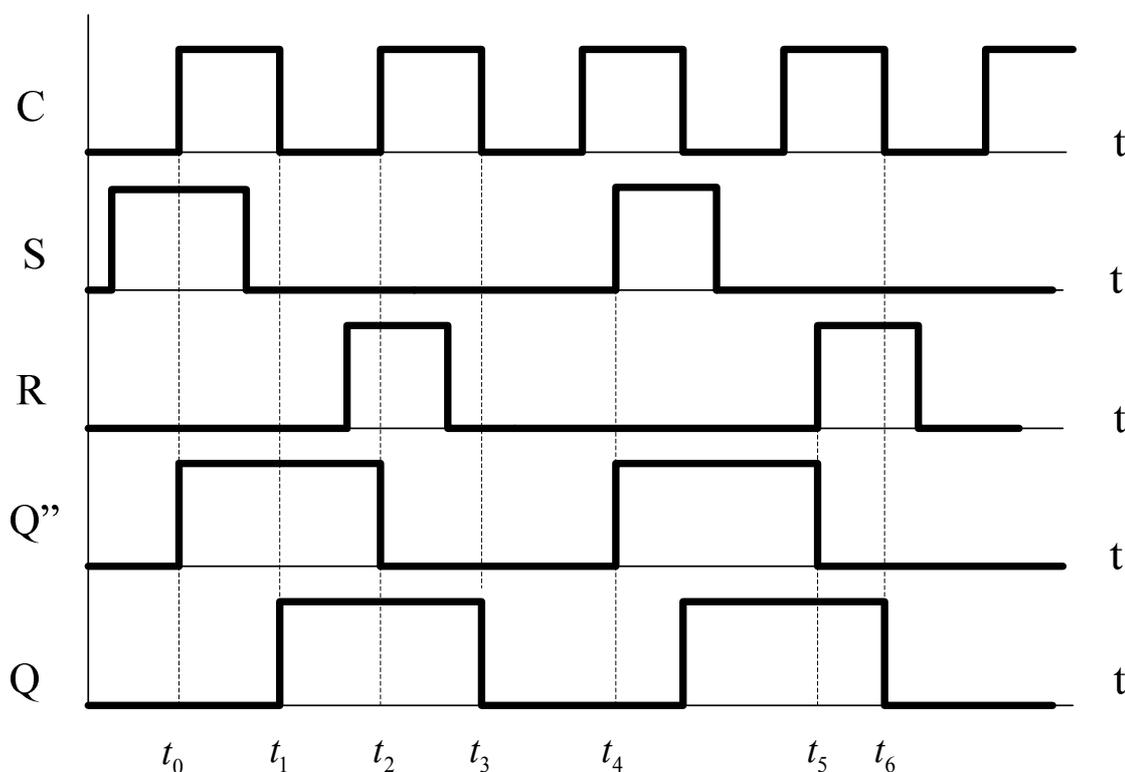


Рис. 5.16. Временная диаграмма двухступенчатого RS-триггера.

Синхронный D-триггер.

Триггером D-типа называется логическое устройство с двумя устойчивыми состояниями и одним информационным входом D и входом синхроимпульса C.

Таблица переходов синхронного D-триггера имеет вид, табл. 5.4.

Таблица 5.4

Таблица истинности синхронного D – триггера.

C^t	D^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Характеристическое уравнение синхронного D-триггера после несложных преобразований примет вид (5.8).

$$Q^{t+1} = C^t Q^t + C^t D^t \quad (5.8)$$

Из уравнения следует, что при наличии синхросигнала ($C=1$) триггер переходит в состояние, предписанное D-входом, т.е. $Q^{t+1} = D^t$. При отсутствии синхросигнала ($C=0$) триггер сохраняет предыдущее состояние. Таким образом, триггер задерживает прохождение поступившего по D-входу информационного сигнала до появления синхросигнала. Отсюда название D-триггера (delay-задержка).

Характеристическое уравнение, преобразованное в структурное для его реализации на элементах И-НЕ, примет вид (5.9).

$$Q^{t+1} = \overline{\overline{C^t D^t} * Q^t * C^t D^t} \quad (5.9)$$

Реализация этого структурного уравнения синхронного D-триггера на элементах «И-НЕ» приведена на рис. 5.17, а УГО D – триггера на рис. 5.18.

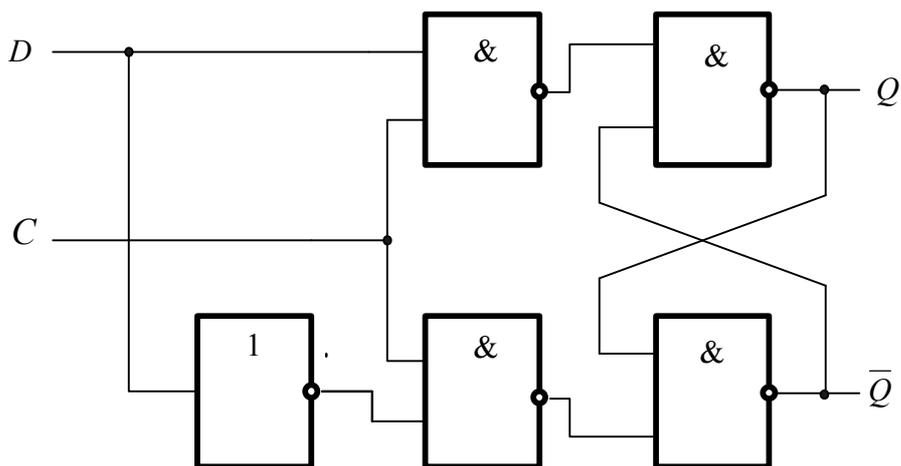


Рис. 5.17 Схема синхронного D-триггера на элементах И-НЕ.

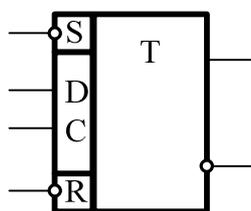


Рис. 5.18 Условно-графическое изображение синхронного D-триггера.

Временная диаграмма синхронного D-триггера представлена на рис. 5.19

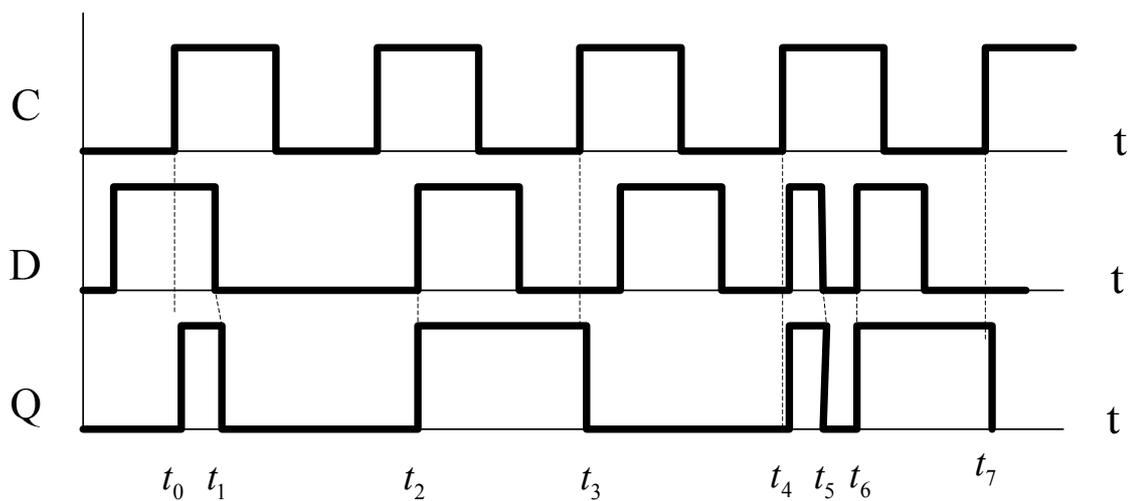


Рис. 5.19. Временная диаграмма синхронного D-триггера.

Синхронные триггеры динамического типа.

Рассмотренные выше типы синхронных триггеров относились к потенциальному типу. Особенность триггеров потенциального типа состоит в возможности многократного изменения состояния триггеров при установленном в единицу сигнале синхроимпульса. В отличие от них синхронные триггеры динамического типа изменяют своё состояние только в момент изменения синхроимпульса из нуля в единицу, либо из единицы в ноль. При установившемся значении единичного синхроимпульса динамический триггер не меняет своё состояние при подаче новой комбинации установочных сигналов.

На рис. 5.20 и рис. 5.21 приведены функциональные схемы динамических синхронных RS и D триггеров.

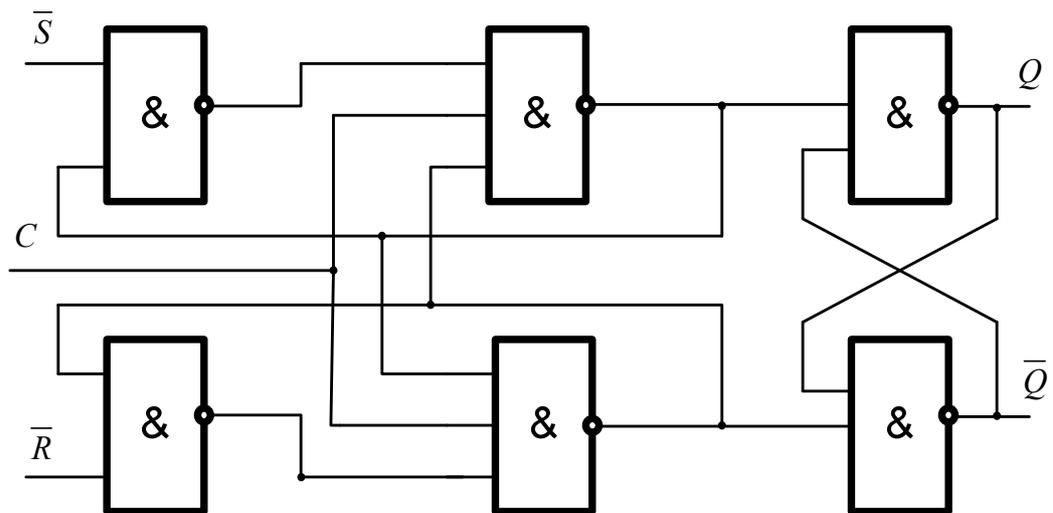


Рис. 5.20. Схема синхронного динамического RS-триггера.

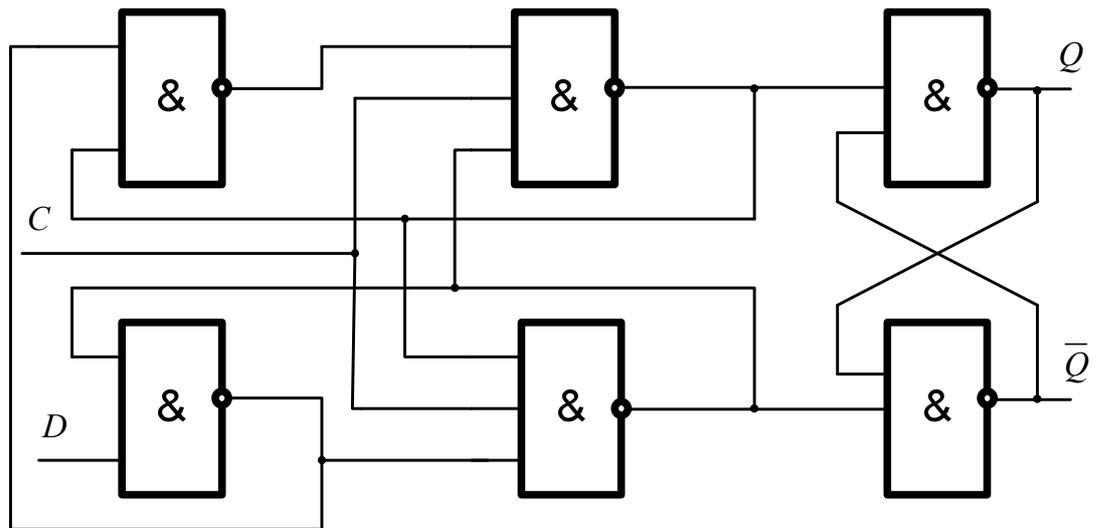


Рис. 5.21. Схема синхронного динамического D-триггера.

Условно-графические обозначения динамических RS и D триггеров приведены на рис. 5.22. Признаком динамического типа триггера является наличие обозначения стрелочки на входе синхроимпульса C . Если в обозначении динамического триггера стрелочка на входе синхроимпульса направлена к триггеру, то установка триггера в новое состояние происходит по фронту синхроимпульса, а, если стрелочка направлена от обозначения триггера, то установка триггера в новое состояние происходит по спаду синхроимпульса. Для перевода динамического триггера в противоположное состояние обязательно необходимо подать следующий синхроимпульс.

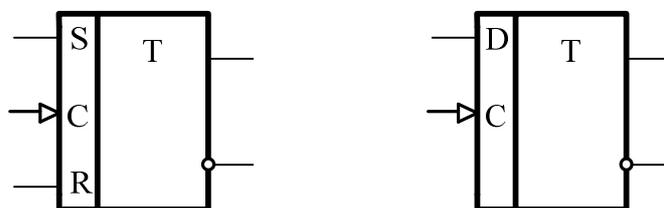


Рис. 5.22 Условно-графические обозначения динамических RS и D триггеров.

Временная диаграмма динамического синхронного R-S триггера представлена на рис. 5.23

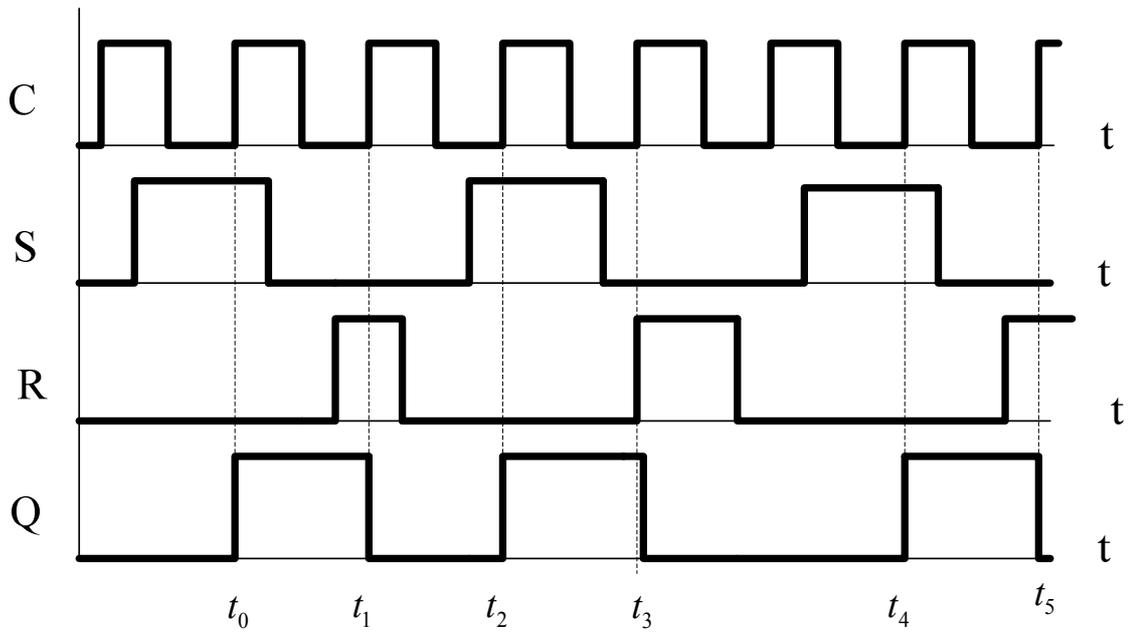


Рис. 5.23 Временная диаграмма динамического синхронного R-S триггера.

Универсальный JK-триггер.

Триггером JK-типа называется логическое устройство с двумя устойчивыми состояниями и двумя информационными входами. Триггер функционирует в соответствии со следующей таблицей переходов, табл. 5.5

Таблица 5.5

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

Карта Карно с перенесенной на неё таблицей переходов JK-триггера изображена на рис. 5.24

$J^t K^t \backslash C^t Q^t$	00	01	11	10
00		1	1	
01		1		
11		1		1
10		1	1	1

Рис. 5.24 Карта Карно для уравнения JK триггера.

Из карты Карно следует, что МДНФ характеристического уравнения JK-триггера имеет вид (5.10).

$$Q^{t+1} = \overline{C^t} Q^t + C^t Q^t \overline{K^t} + C^t \overline{Q^t} J^t \quad (5.10)$$

При $C = 0$, $Q^{t+1} = Q^t$, а при $C=1$,

$$Q^{t+1} = Q^t \overline{K^t} + \overline{Q^t} J^t \quad (5.11)$$

Если принять, что $K = R$, а $J = S$, то при $C = 1$ JK-триггер ведёт себя как RS-триггер, за исключением комбинации $J = K = 1$, при которой триггер меняет своё состояние на обратное по отрицательному фронту (спаду) синхросигнала.

Построить JK-триггер можно на основе двухступенчатого RS-триггера с входом синхронизации, соединив перекрёстным образом инвертирующими обратными связями входы и выходы триггера, рис.5.25 Указанные перекрёстные связи и являются отличительной особенностью JK-триггера от RS-триггера с задержкой (типа MS).

УГО JK-триггера представлено на рис.5.25

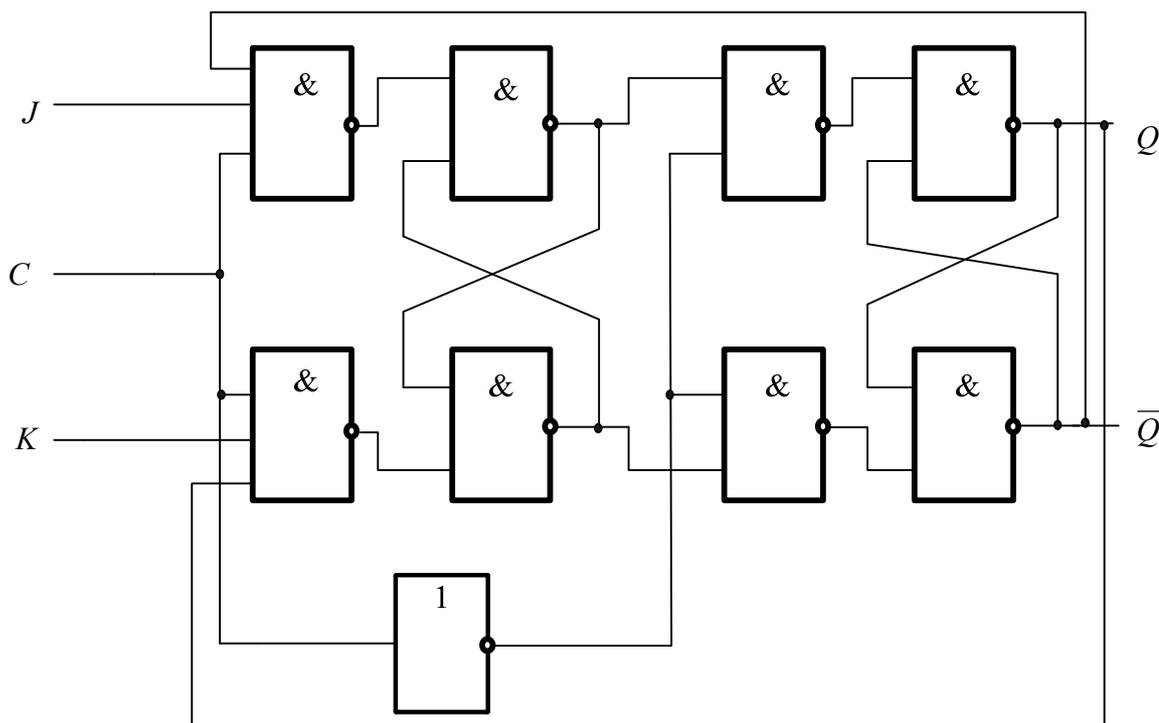


Рис. 5.25 Схема двухступенчатого JK-триггера.

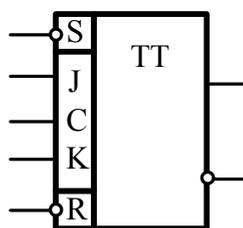


Рис. 5. 26 Условно-графическое изображение универсального JK- триггера.

Триггер JK - типа относится к разряду универсальных триггеров, поскольку на его основе путем несложных внешних коммутаций входов триггера можно получить схемы, выполняющие функции асинхронного RS-триггера, синхронного RS-триггера, синхронного D-триггера и Т-триггера. Временная диаграмма JK – триггера представлена на рис. 5.27.

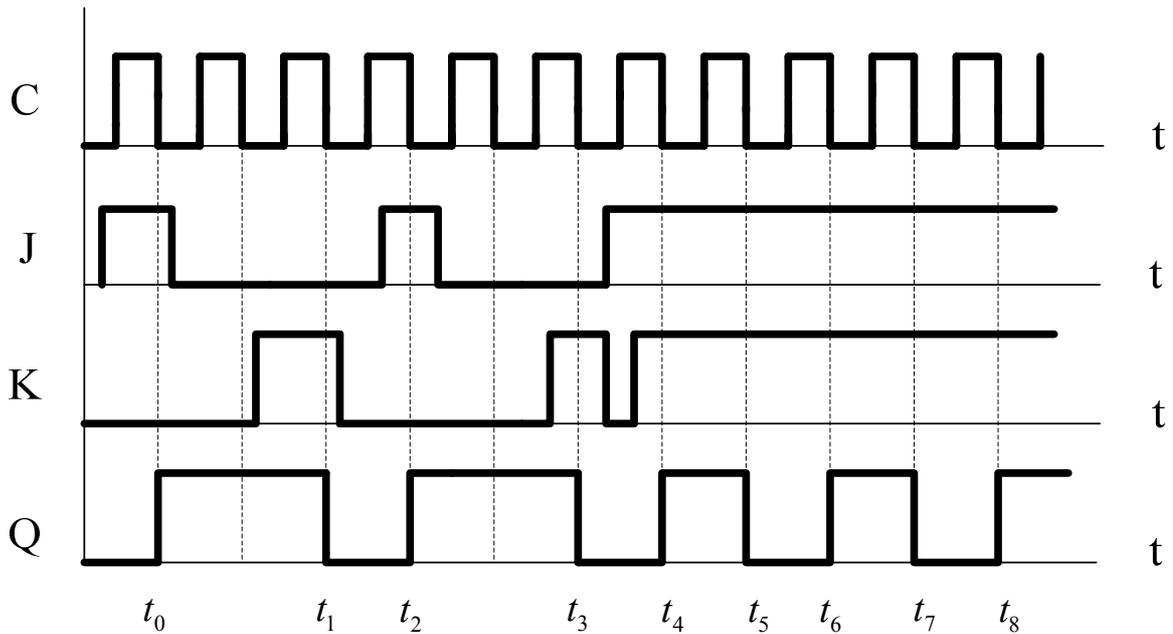


Рис. 5.27 Временная диаграмма работы JK – триггера.

Триггеры со счетным входом (Т-триггер).

Триггером Т-типа называют логическое устройство с двумя устойчивыми состояниями и с одним информационным Т-входом. Т-триггер меняет своё состояние на противоположное всякий раз, когда на Т-вход поступает управляющий (счетный) сигнал. Асинхронный Т-триггер функционирует в соответствии со следующей таблицей переходов, табл. 5.6.

Таблица 5.6
Таблица истинности счётного триггера.

T^t	Q^t	Q^{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Характеристическое уравнение Т-триггера, исходя из таблицы переходов, будет иметь вид (12).

$$Q^{t+1} = \overline{Q^t}T^t + Q^t\overline{T^t} \tag{5.23}$$

Это уравнение отображает операцию суммирования по модулю два сигнала состояния триггера Q и входного сигнала T.

Синхронный Т-триггер функционирует в соответствии со следующей таблицей переходов, табл.5.7.

Таблица 5.7

C^t	T^t	Q^t	Q^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Из таблицы переходов вытекает следующее характеристическое уравнение Т-триггера (14).

$$Q^{t+1} = \overline{C^t} \overline{Q^t} \overline{T^t} + \overline{C^t} Q^t T^t + C^t \overline{Q^t} T^t + C^t Q^t \overline{T^t} \quad (5.14)$$

или
$$Q^{t+1} = \overline{C^t} (Q^t \overline{T^t} + Q^t T^t) + C^t (\overline{Q^t} T^t + Q^t \overline{T^t}) \quad (5.15)$$

Из (5.15) следует, что при $C^t = 1$ триггер работает как асинхронный Т-триггер, а при $C^t = 0$, $Q^{t+1} = Q^t$, т.е. триггер сохраняет предыдущее состояние.

Построить Т-триггер можно на основе JK-триггера и синхронного динамического D-триггера. Два варианта счётного триггера, преобразованного из JK-триггера представлены на рис. 5.28 и рис. 5.29 соответственно. На рис. 5.30. изображена схема синхронного Т-триггера на основе D-триггера. Схема счётного триггера на рис. 5.30 работает по положительному (переднему) фронту синхросигнала, а схема триггера на рис. 5.32 по отрицательному (заднему) фронту синхросигнала.

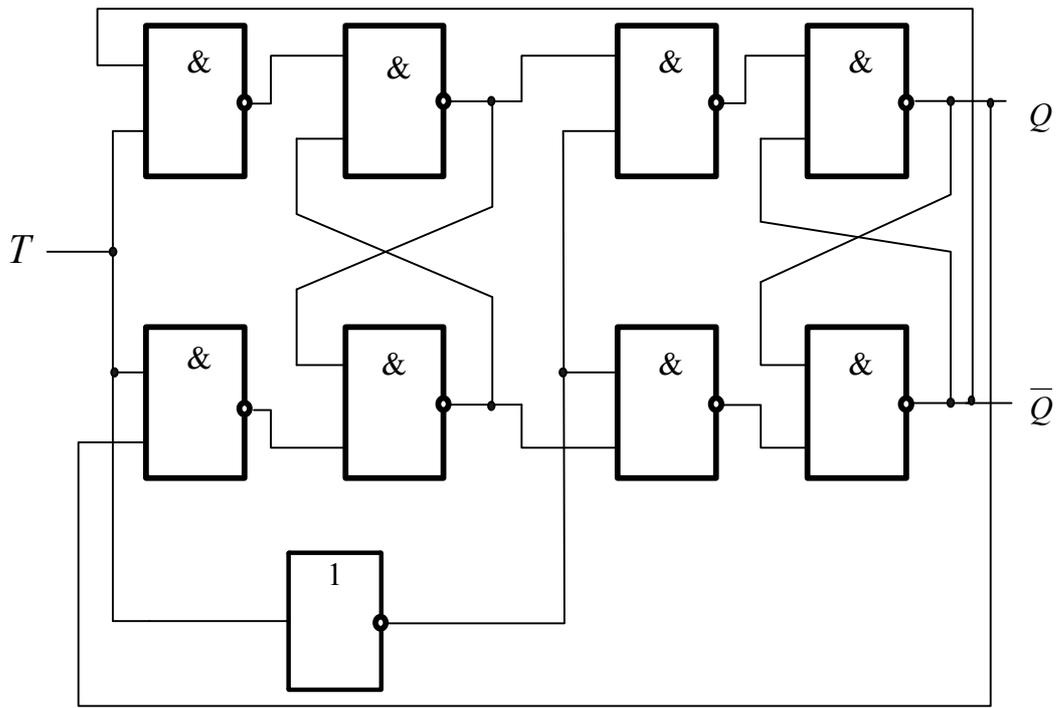


Рис. 5.28. Асинхронный двухступенчатый Т-триггер.

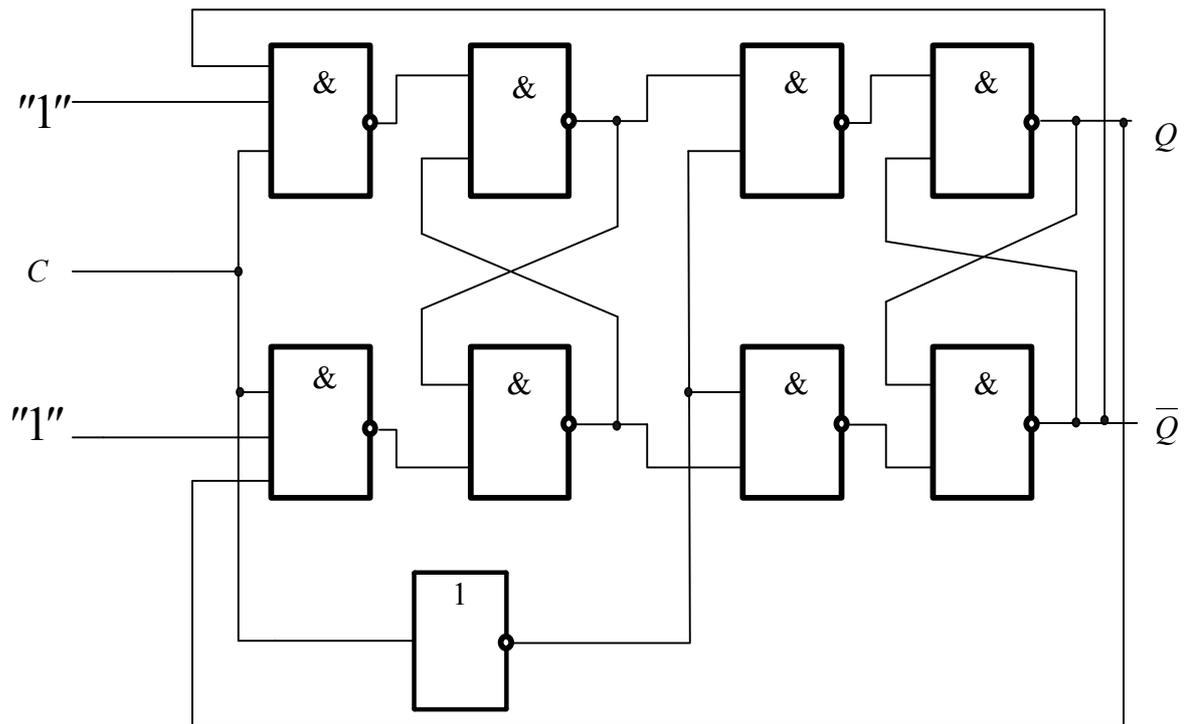


Рис. 5.29. Синхронный двухступенчатый Т-триггер.

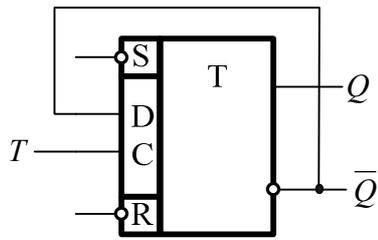


Рис. 5.30 Т-триггер на основе синхронного D-триггера.

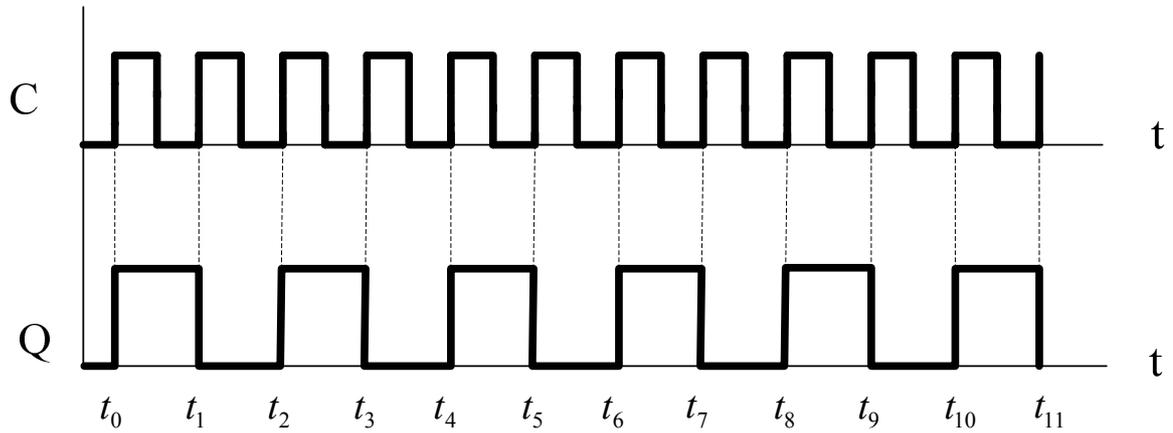


Рис. 5.31. Временная диаграмма Т-триггера на основе синхронного D-триггера.

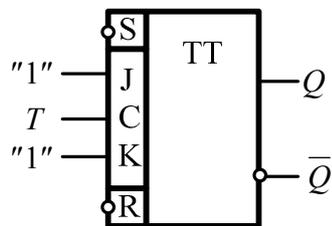


Рис. 5.32 Т-триггер на основе универсального JK-триггера.

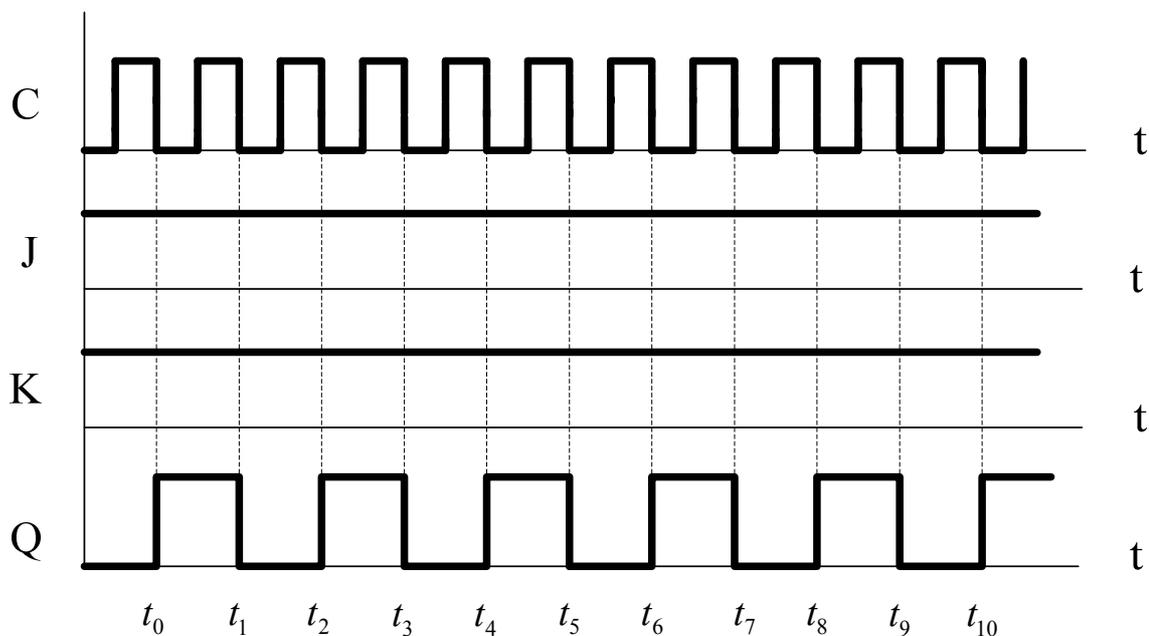


Рис. 5.33 Временная диаграмма Т-триггер на основе универсального JK-триггера.

На УГО триггеров две буквы ТТ в основном поле обозначения триггера указывают, что триггер двухступенчатый и, следовательно, он работает по заднему фронту синхросигнала.

Как видно из рис. 5.30 для получения счетного триггера из D-триггера необходимо соединить инверсный выход D-триггера с его D-входом.

При таком соединении входа и выхода триггера в характеристическое уравнение D-триггера (9) нужно подставить $D^t = Q^t$ и тогда уравнение D-триггера приобретает вид характеристического уравнения Т-триггера (5.16)

$$Q^{t+1} = \overline{C^t} Q^t + C^t \overline{Q^t} \quad (5.16)$$

5.3. Применение триггеров в схемах управления и переключения.

Триггеры нашли применение не только как запоминающие элементы для временного хранения двоичного одноразрядного кода, иными словами одного бита информации, но и широко применяются как электронные ключи.

В качестве примера можно рассмотреть несколько таких схем, имеющих широкое распространение в разных по назначению устройствах.

Пример схемы 1. При использовании механических ключей, или электромеханических реле, возникает так называемый «дребезг контактов». При этом явлении возникают упругие колебания с длительностью от 1 до 10 мс. Такой сигнал нельзя подавать в цифровое устройство, так как он может создать множество ложных переключений.

Схема механического ключа с использованием триггера представлена на рис. 5.34.

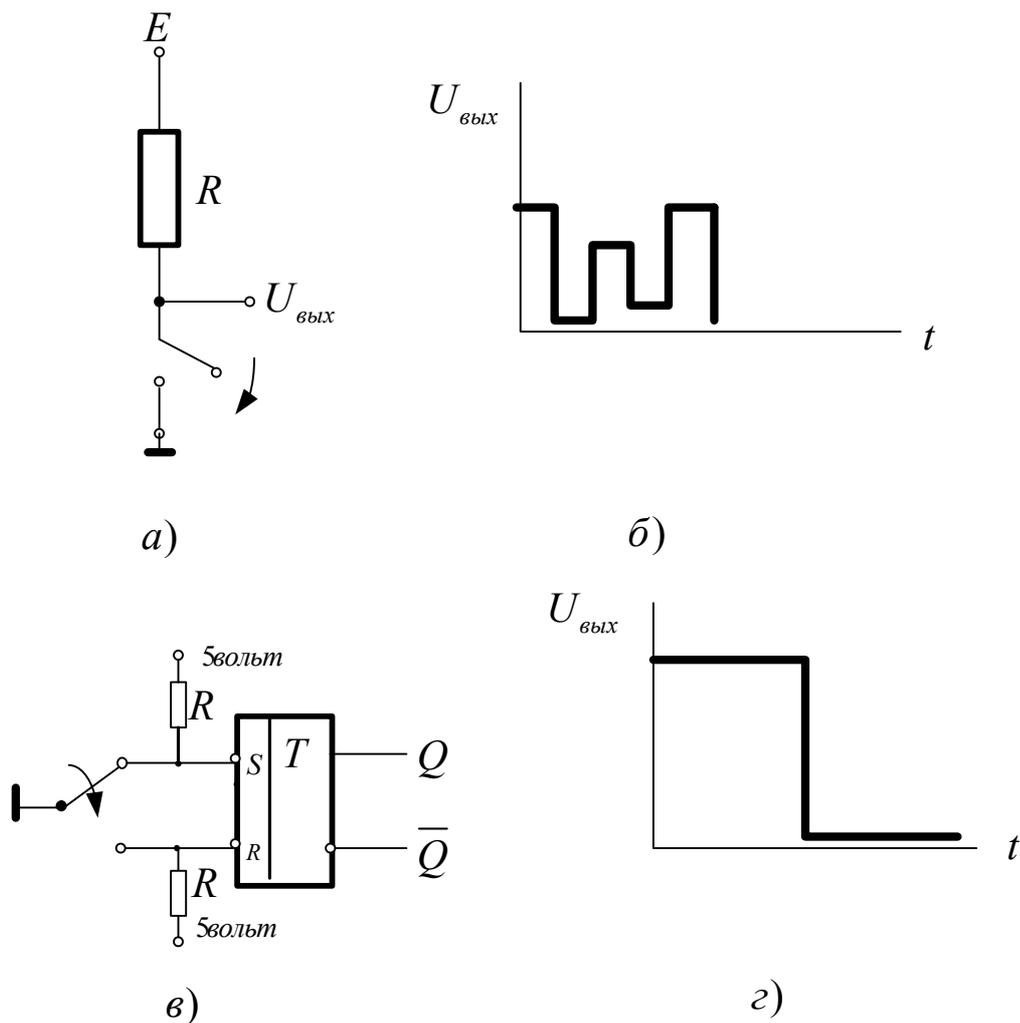


Рис. 5.34. Классический ключ а), диаграмма дребезга контакта б), ключ, управляемый триггером г), диаграмма выходного сигнала с выхода триггера г).

С помощью триггера выходное напряжение $U_{\text{вых}}$ очищается от паразитных колебаний. Первое соударение ключа приводит триггер в

соответствующее состояние, а при отскоке ключа оба входа триггера получают пассивные сигналы логической единицы (высокого уровня).

Пример схемы 2. Необходимо создать устройство выделения одиночного импульса Y длительностью равного тактирующему импульсу T при наступлении события x . На рис. 5.35 представлены временные диаграммы, согласно которым должно работать проектируемое устройство.

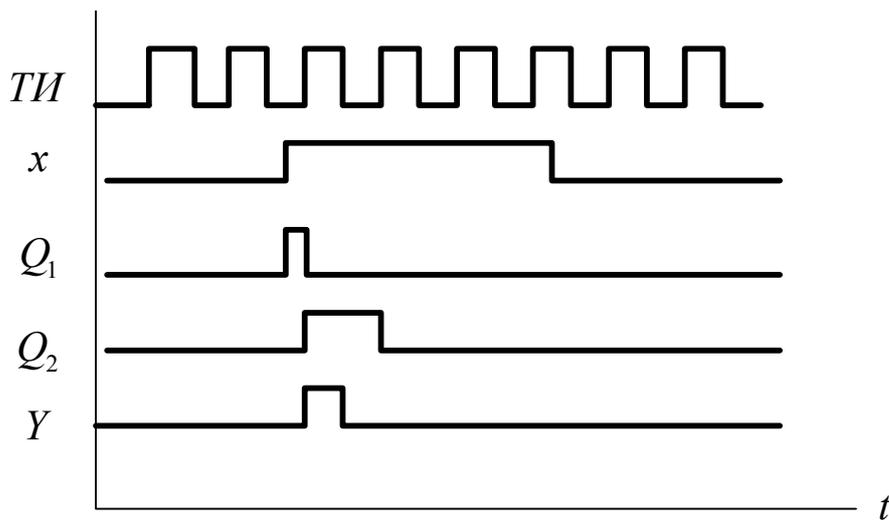


Рис. 5.35 Временные диаграммы устройства выделения одиночного импульса.

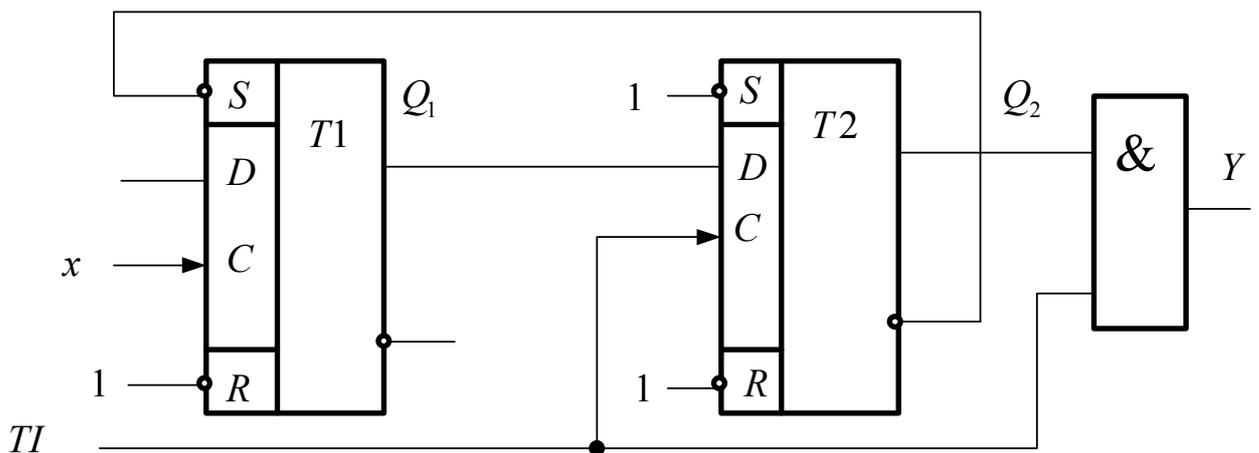


Рис. 5.36 Схема устройства выделения одиночного импульса.

В исходном состоянии триггеры $T1$ и $T2$ находятся в нулевом состоянии. Нулевое значение сигнала Q_2 , блокирует прохождение

тактовых импульсов через логический элемент «И». Приходящий сигнал x переведёт триггер $T1$ в единичное состояние по ближайшему начальному фронту тактового импульса $TИ$. Следующий тактовый импульс переведёт триггер $T2$ в нулевое состояние, что и приведёт к блокировке логического элемента «И». В итоге сигнал Y , будет выглядеть, как на рис. 5.36.

5.4. Типовые примеры проектных решений схем триггеров.

5.4.1. Необходимо реализовать дополнительное управление работой D-триггера сигналом разрешения V . Такие триггеры иногда именуют DV-триггерами. Сигнал разрешения должен блокировать воздействие сигнала установки D . Приведите возможные схемы реализации.

Пример решения:

В первом варианте сигналом разрешения можно заблокировать происхождение установочного сигнала D , предварительно пропустив его через логический элемент «И», управляемый сигналом V рис. 5.38.

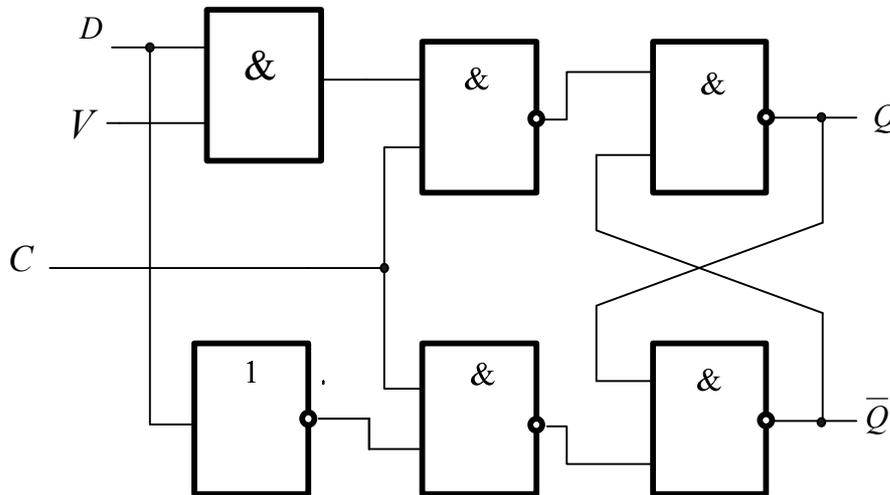


Рис. 5.38. Схема DV- триггера.

Сигнал разрешения должен блокировать воздействие сигнала установки D . Во втором варианте сигнал разрешение V осуществляет блокировку сигнала синхроимпульсов C рис. 5.39.

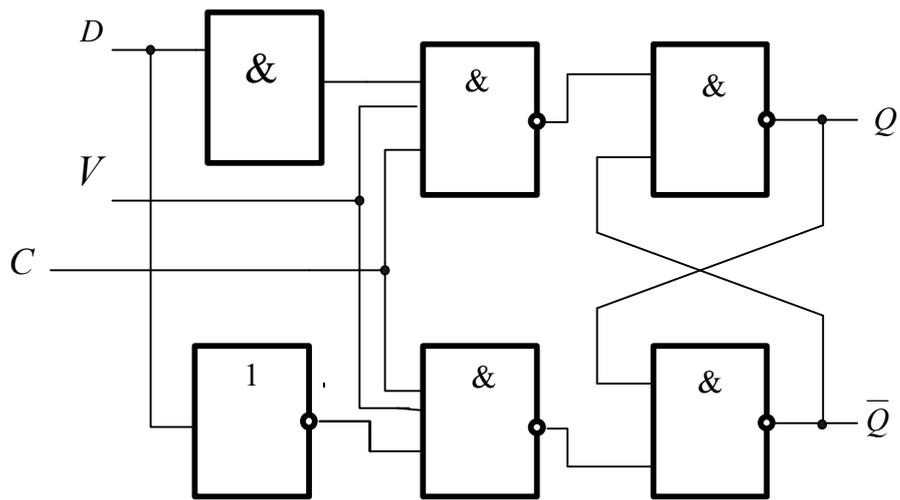


Рис. 5.39 Схема DV- триггера, реализованного по второму варианту.

5.4.2 Необходимо получить последовательность импульсов, обозначенную ТИ2 на временной диаграмме рис. 5.40, состоящую только из «чётных» сигналов синхропоследовательности ТИ1.

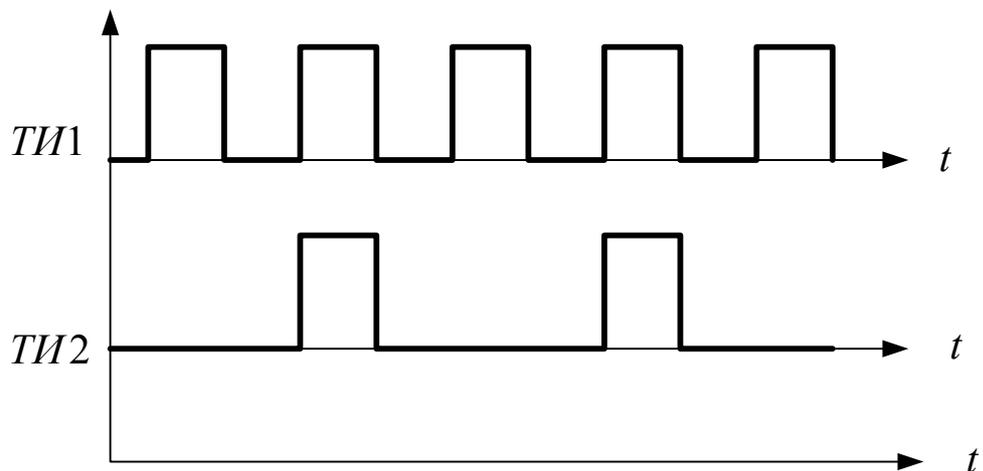


Рис. 5.40 Временная диаграмма выделения чётного импульса.

Пример решения:

В качестве устройства селекции (выделения) подойдёт счётный триггер, работающий по начальному фронту синхроимпульса ТИ1.

При работе счётного триггера его состояние равно 1 при поступлении нечётных импульсов, после подачи чётного синхроимпульса счётный триггер принимает нулевое значение. Следовательно, для пропуска сигнала через логический элемент «И» надо использовать инверсный выход триггера рис. 5.41.

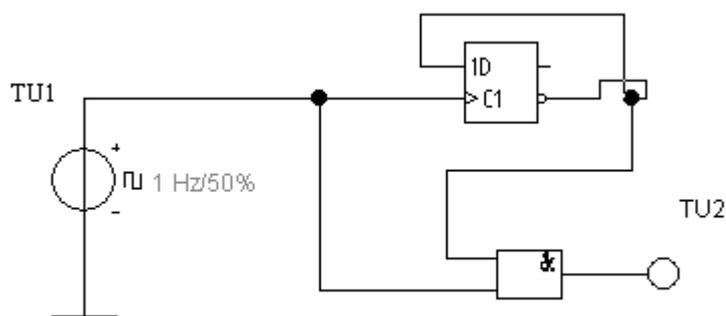


Рис. 5.41 Схема выделения чётных импульсов синхропоследовательности.

Компетенции, приобретаемые после изучения главы 5.

Помнить:

- классификацию триггеров по ряду сравнительных критериев и основное определение триггера.
- методику синтеза асинхронного RS-триггера.

Понимать:

- различия функционирования триггера в зависимости от его схемного построения.
- назначение входных сигналов и управляющее их воздействие на состояние триггера.
- роль синхроимпульса в работе синхронных триггеров.

Применять:

- схемы триггеров для реализации схем включения и выработки сигналов.
- уметь строить функционально более сложные схемы из однотипных триггеров.

Оценивать:

- возможные аномалии, вызываемые естественным разбросом параметров триггеров.
- возможность взаимозаменяемости триггеров различного типа.

5.5 Контрольные вопросы.

Вопросы категории 1. «Помнить»

5.5.1. Сформулируйте назначение сигнала синхронизации в синхронных триггерах.

5.5.2. Перечислите основные временные параметры триггеров и их физический смысл.

5.5.3. Перечислите разновидности триггеров MS-типа.

Вопросы категории 2. «Понимать»

5.5.4. Схемотехническая особенность построения триггеров MS-типа.

5.5.5. Чем отличается триггер потенциального типа от динамического?

5.5.6. В чём состоит различие RS-триггера, реализованного на логических элементах «И-НЕ» и «ИЛИ-НЕ»?

5.5.7. В чём состоят отличия временных диаграмм счётных триггеров, преобразованных из динамического D-триггера и JK-триггера?

5.5.8. Как определить по временной диаграмме динамический триггер от потенциального.

5.5.9. Чем могут быть вызваны аномалии в триггерах?

Вопросы категории 3. «Применять»

5.5.10. Сформулируйте роль перекрёстных обратных связей в схеме построения триггера JK-типа.

5.5.11. Какие установочные входы имеют высший приоритет у универсального JK- триггера?

5.5.12. Как преобразовать схему синхронного RS-триггера в D-триггер.

Вопросы категории 4. «Оценивать»

5.5.13. В чём состоит универсальность JK-триггера?

5.5.14. В чём состоит принципиальное отличие динамического типа триггеров, от потенциальных триггеров?

5.5.15. Поясните термин «асинхронные установочные входы» триггера.

5.5.16. При какой комбинации входных сигналов JK-триггер работает как счётный триггер.

Ответы на контрольные вопросы.

5.5.1 Триггер динамического типа устанавливается только при воздействии начального фронта синхроимпульса.

5.5.2 Основные временные характеристики триггеров: время переключения триггера и минимально допустимая длительность установочных сигналов.

5.5.3 Триггерами MS - типа могут быть синхронные RS – триггеры, синхронные D – триггеры, JK – триггеры, T – триггеры.

5.5.4 Триггер MS-типа представляет собой два соединённых друг за другом RS-триггеров, и управляемы прямым и инверсным значением синхроимпульсов.

5.5.5 Синхронный триггер потенциального типа может менять своё состояние под воздействием установочных сигналов много раз во время установившегося в единицу сигнала синхроимпульса. Синхронный динамический триггер изменяет своё состояние под воздействием установочных сигналов только в момент перехода сигнала синхроимпульса из нуля в единицу.

5.5.6 Триггер, реализованный на логических элементах «И-НЕ» имеет инверсные установочные входы, а реализованный на логических элементах «ИЛИ-НЕ» имеет прямые установочные входы.

5.5.7 Переключение счётного триггера, преобразованного из D –триггера, осуществляется по фронту синхроимпульса, а счётный триггер, преобразованный из JK – триггера переключается по спаду синхроимпульса.

5.5.8 На временной диаграмме динамический триггер отличается переключением только по фронту синхроимпульса.

5.5.9 Аномалии в триггерах могут быть вызваны длительностью установочных сигналов, короче допустимой минимальной длительности.

5.5.10 Перекрёстные обратные связи в JK-триггере снимают запрет на одновременную подачу единиц по J и K входу.

5.5.11 У универсального JK – триггера наивысший приоритет имеют асинхронные установочные входы.

5.5.12 Для преобразования достаточно вместо входа R подать инвертированный сигнал с входа S рис. 5.32.

5.5.13 Термин универсальный для JK-триггера обозначает возможность его использования в качестве:

- асинхронного и синхронного RS-триггеров,
- синхронного D- триггера, счётного T-триггера.

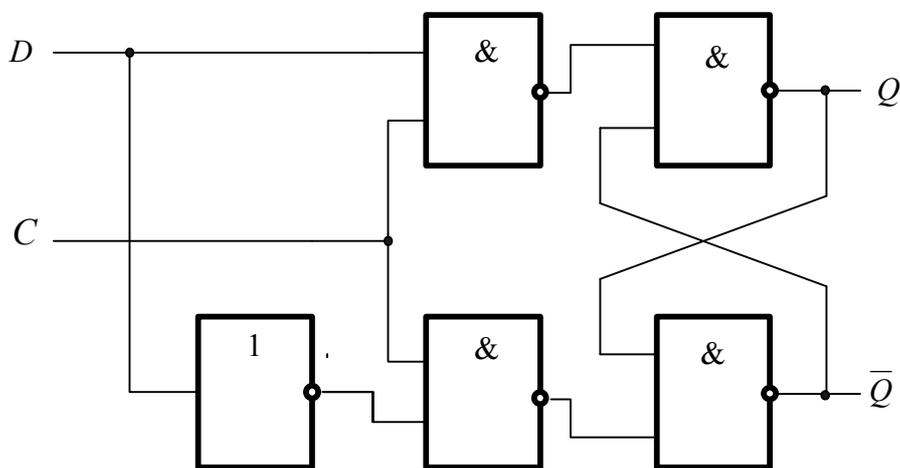


Рис. 5.32. Преобразование в D-триггер.

5.5.14 Отличие динамического типа триггеров от потенциальных триггеров, состоит в воздействии установочного сигнала на триггер только в момент перехода синхроимпульса из нуля в единицу.

5.5.15 Эти входы получили термин асинхронные из-за того, что имеют приоритет воздействия на установку триггера независимо от подаваемого синхроимпульса.

5.5.16 Переход работы JK-триггера в счётный режим происходит при установке сочетания сигналов $J = K = 1$.

5.6. Задачи для самостоятельного решения.

5.6.1 Преобразуйте D-триггер динамического типа в счётный триггер?

Решение: Для данного преобразования достаточно соединить инверсный выход триггера с входом D. (рис. 5.33).

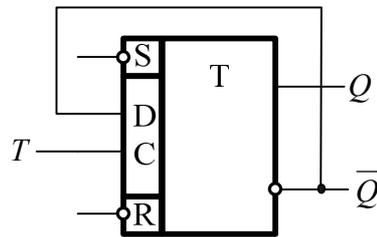


Рис. 5.33 D- триггер, преобразованный в счётный.

5.6.2 Нарисуйте схему преобразования RS- триггера, управляемого задним фронтом синхросигнала, в JK- триггер.

Решение: изначально данная схема триггера соответствует триггеру типа MS (ведущий-ведомый) рис.5.34., преобразование состоит в добавлении перекрёстных обратных связей рис. 5.35.

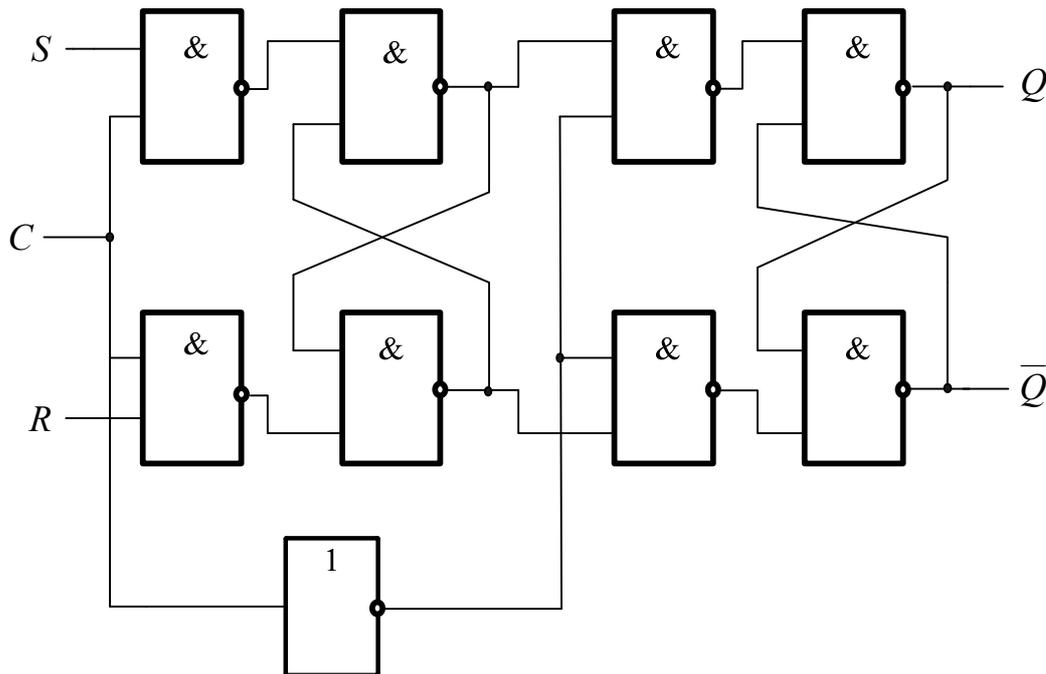


Рис. 5.34 Исходная схема RS- триггера, управляемая задним фронтом

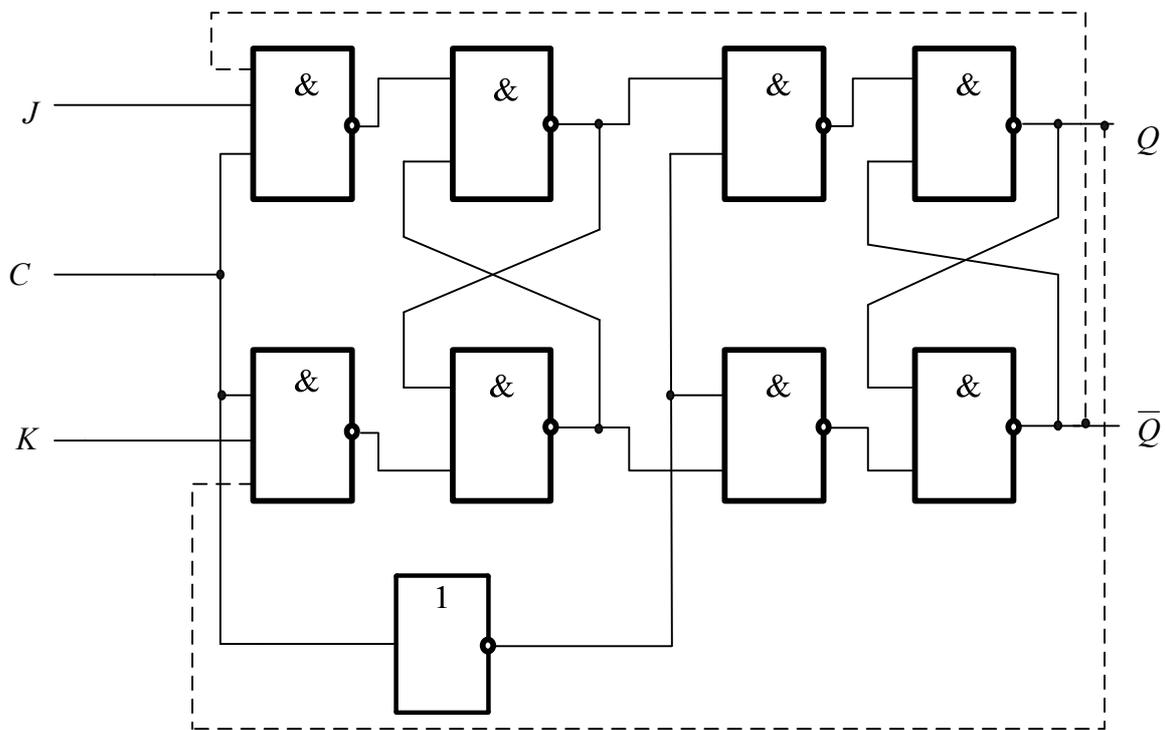


Рис. 5.35 Преобразованная схема триггера за счёт добавления перекрёстных обратных связей.

5.6.3 Как работает синхронный RS- триггер с асинхронным прямым входом S ? Изобразите условно-графическое обозначение временную диаграмму.

Решение:

Условно-графическое отображение (УГО) представлено на рис. 5.36.

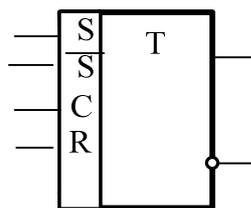


Рис. 5.36. УГО RS-триггера с асинхронным прямым входом S ?

Временная диаграмма изображена на рис. 5.37.

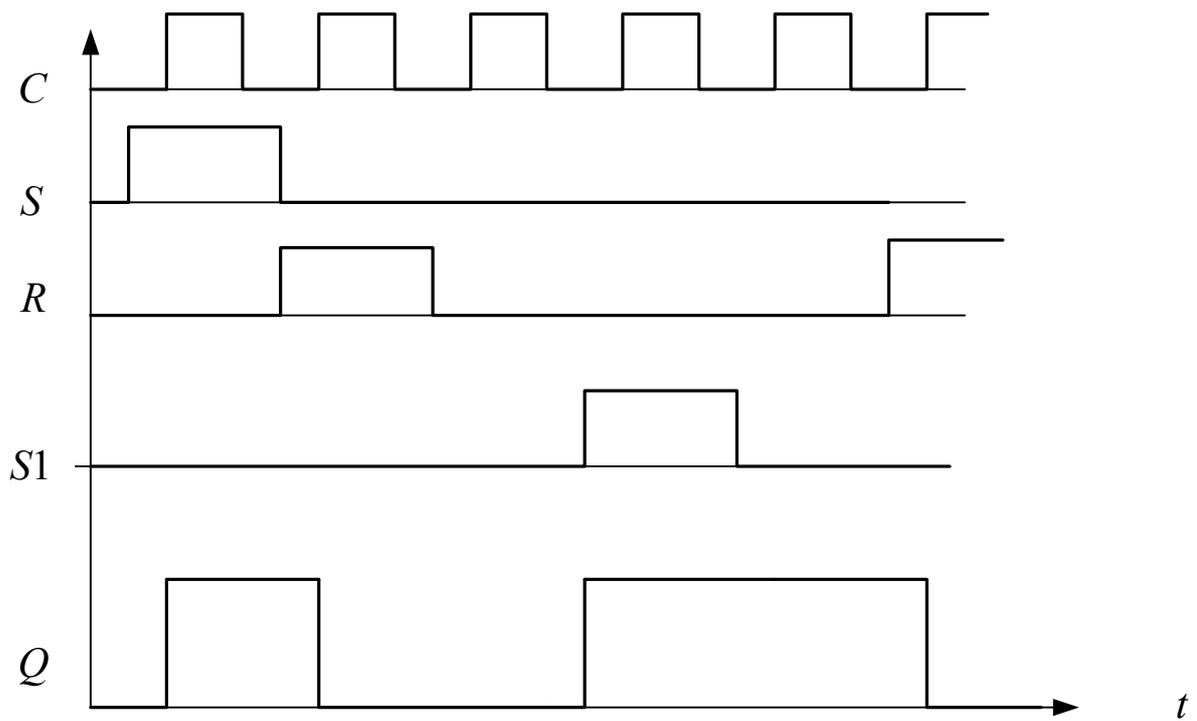


Рис. 5.37. Временная диаграмма для триггера рис. 5.36.

5.6.5. Спроектируйте схему пропуска синхроимпульсов согласно временной диаграмме рис. 5.38.

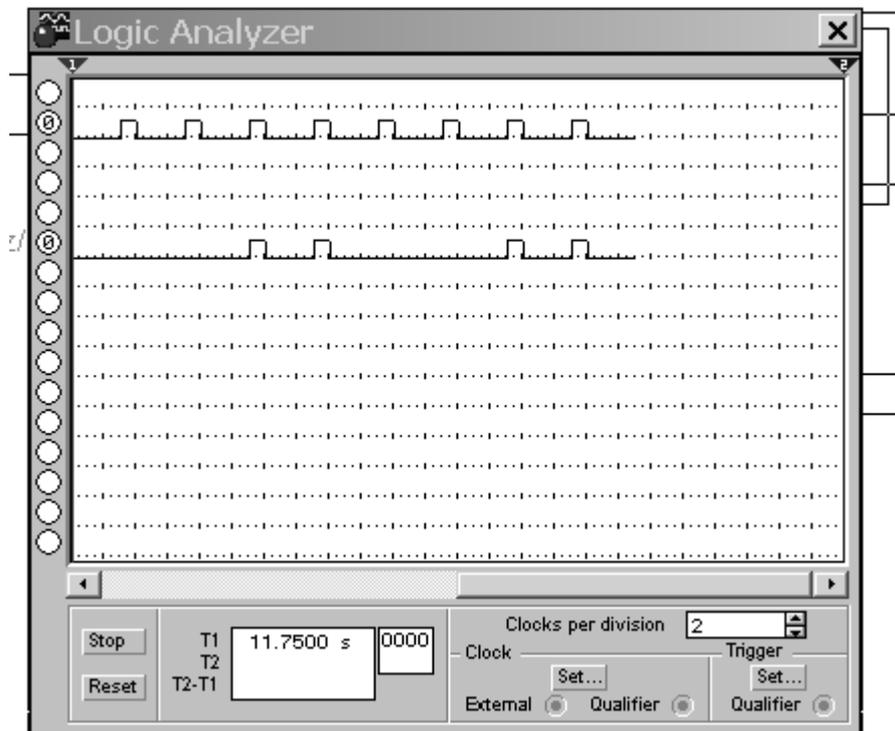


Рис. 5.38 Временная диаграмма выделения 3 и 4 импульса.

Требуемая схема представлена на рис. 5.39.

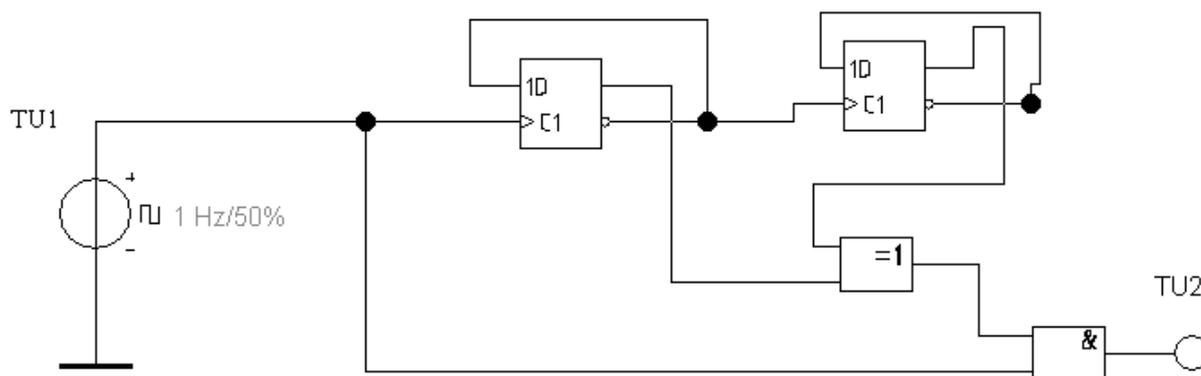


Рис. 5.39. Схема выделения двух импульсов через два.

5.6.6. Объясните назначение входов и принцип действия изображённого на рис. 5.40. триггера.

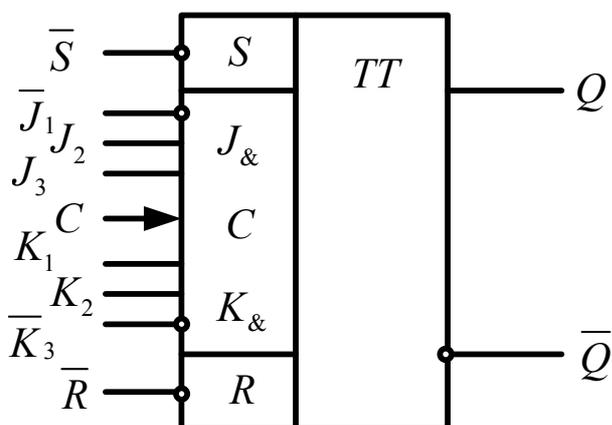


Рис. 5.40. JK- триггер с несколькими установочными входами.

Особенностью данного JK-триггера является наличие нескольких установочных входов, объединённых по функции «И». Триггер установится в 1 при $\bar{J}_1 J_2 J_3 = 1$, и установится в 0 при $K_1 K_2 \bar{K}_3 = 1$

Глава 6. Задания и вопросы для самостоятельной подготовки к рубежному контролю по модулю 1.

Задание 1.

Синтезировать комбинационную схему, реализующую следующую булеву функцию

$$Y = \overline{\overline{x_1 + x_2 + x_3}} + \overline{(x_1 + x_3)} \quad (6.1)$$

Решение:

Для реализации комбинационной схемы сначала необходимо выяснить потребность в логических элементах и количество входов у них. Рассмотрим поэтапно данное выражение функции. Для получения результата операции

$\overline{\overline{x_1 + x_2 + x_3}}$ необходим трёхвходовый элемент «ИЛИ-НЕ»;

для получения инверсных значений $\overline{x_1}$, $\overline{x_2}$, $\overline{x_3}$ необходимо три логических элемента «НЕ»; для получения результата операции $\overline{(x_1 + x_3)}$ необходим двухвходовый элемент «ИЛИ»; завершающий двухвходовый логический элемент «ИЛИ» принимает результаты двух операций: $x_1 + x_2 + x_3$ и $\overline{(x_1 + x_3)}$.

Итоговая комбинационная схема выглядит следующим образом (рис.6.1.)

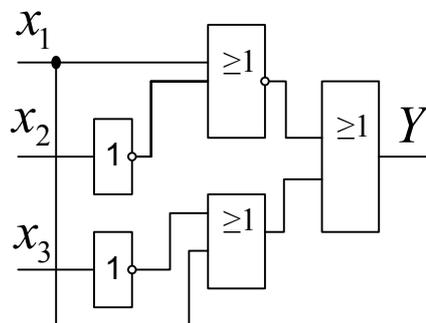


Рис. 6.1 . Комбинационная схема по булевому выражению (6.1).

Задание 2.

Синтезировать комбинационную схему, реализующую следующую булеву функцию

$$Y = \overline{x_1} * x_2 * \overline{\overline{x_1 * x_2}} * x_3 \quad (6.2)$$

Решение:

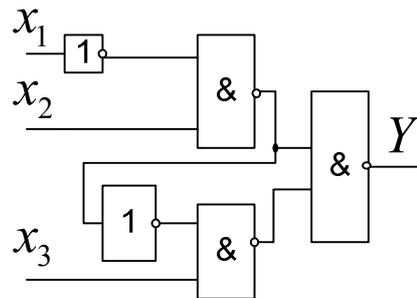
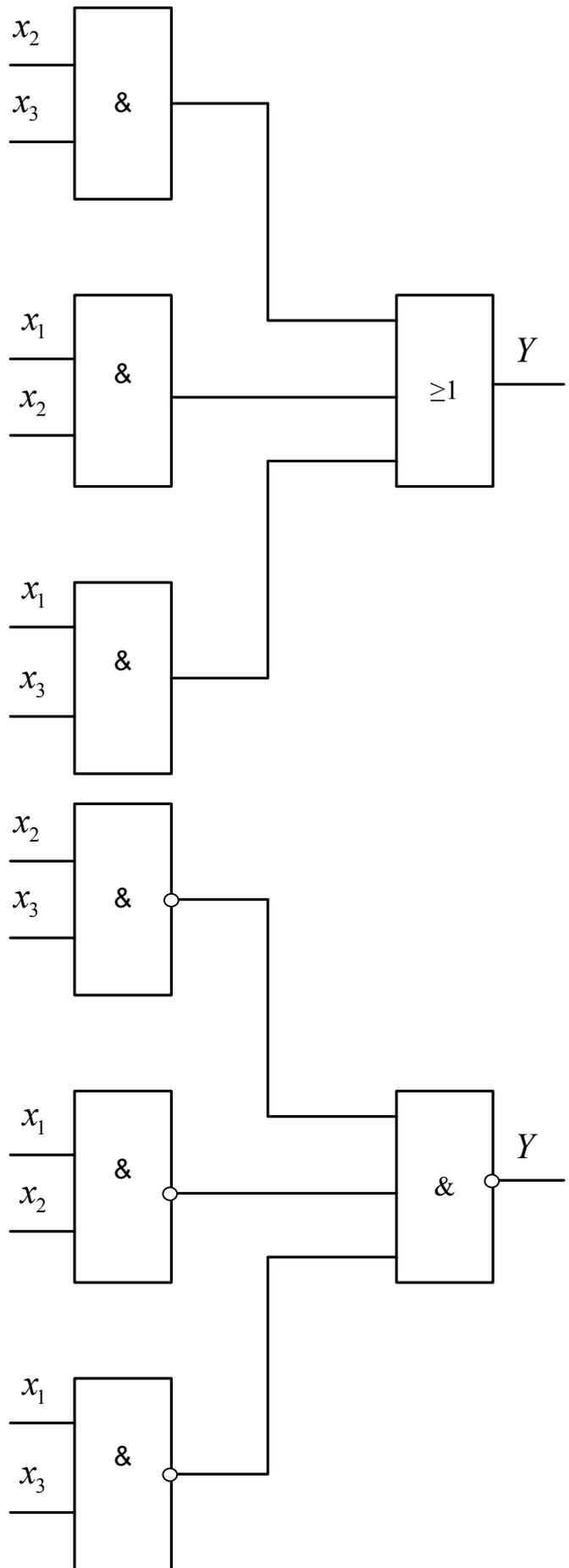


Рис. 6.2 Комбинационная схема по булевому выражению (6.2).

Задание 3.

Из приведённых на рис. 6.3. трёх комбинационных схем, выберите схему соответствующей таблице истинности (табл.6.1.).



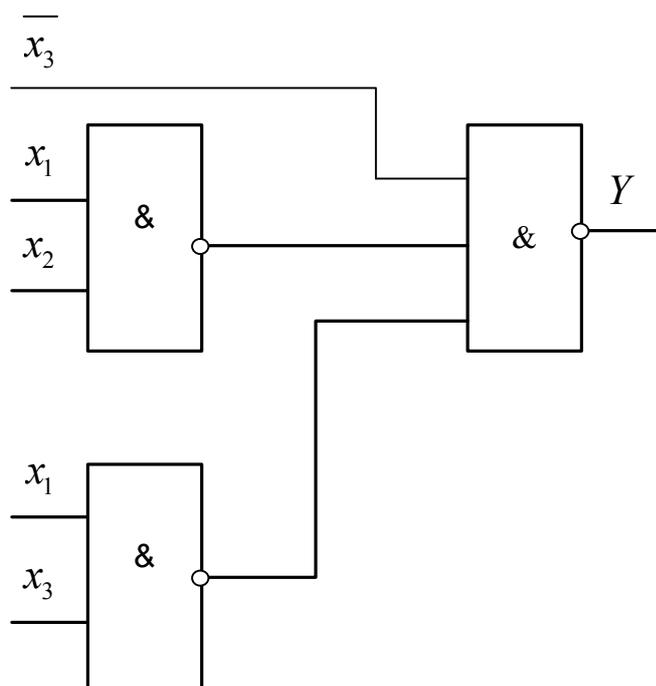


Рис. 6.3. Три варианта комбинационных схем для задания 3.

Решение:

Для получения правильной комбинационной схемы необходимо произвести следующие действия:

- синтезировать комбинационную схему, работа которой описывается булевой функцией, заданной следующей таблицей истинности (табл.6.1);

Таблица 6.1 . Таблица истинности для функции задания 3.

x_1	x_2	x_3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- произвести минимизацию булевой функции с помощью карты Карно;

- изобразить структуру комбинационной схемы по минимизированной записи булевой функции в базе на основе элементов Шеффера.

Задание 4.

Выберите вариант правильного объединения в контура (рис 6.5 а, б, в), соответствующими приведённой заполненной исходной карте Карно (рис. 6.4).

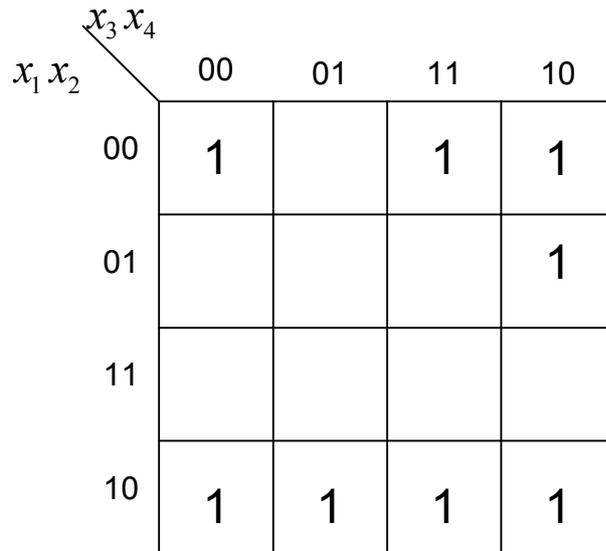
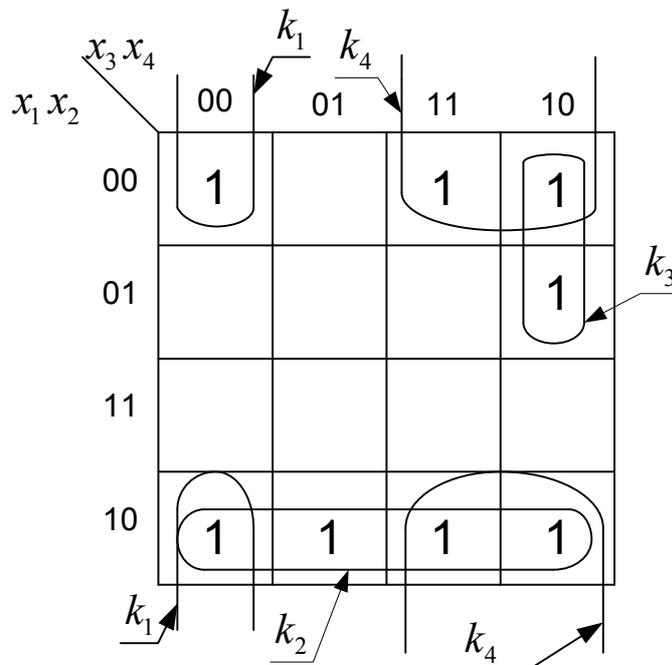


Рис. 6.4 Исходная карта Карно для синтеза минимизированной КНФ.



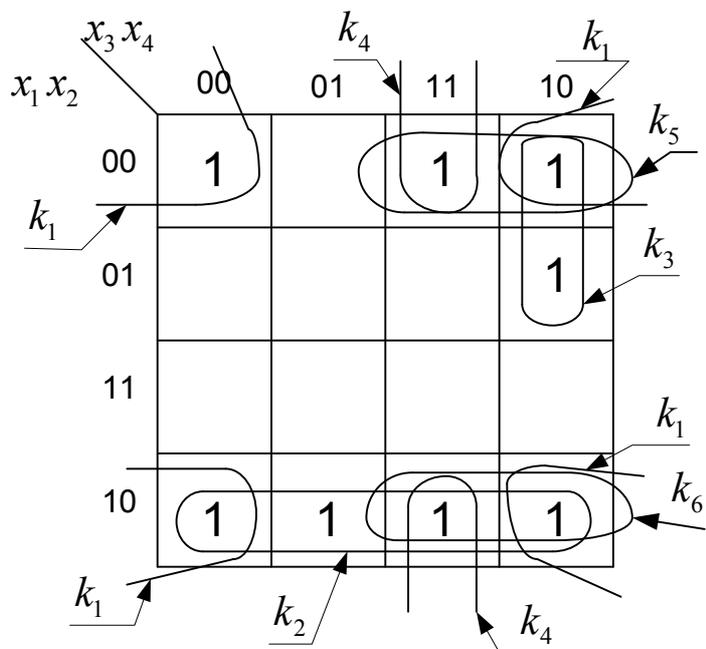
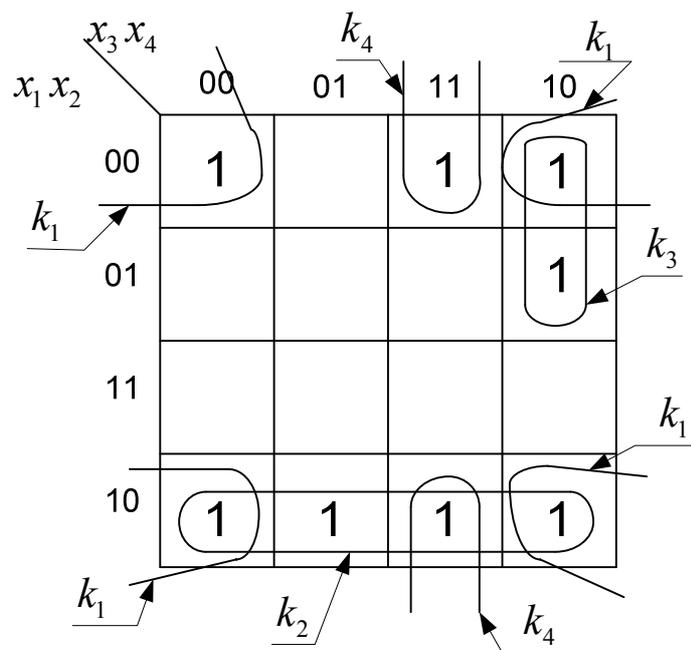


Рис. 6.5. Варианты объединения в контура для Карты Карно рис. 6.5.

Задание 5.

Записать СДНФ для функций, приведённых в таблице 6.2.

Таблица 6.2. Варианты булевых функций.

X ₁	X ₂	X ₃	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉	Y ₁₀	Y ₁₁
0	0	0	1	1	0	0	1	0	1	0	0	0	0
0	0	1	1	1	1	1	0	0	1	0	0	0	0
0	1	0	1	0	0	1	1	1	0	0	0	0	0
0	1	1	0	0	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	0	1	0	1	0	1	0
1	0	1	0	0	0	1	1	1	0	1	0	0	1
1	1	0	0	1	1	1	0	0	0	1	1	1	1
1	1	1	0	1	0	0	1	0	0	0	1	1	1

Решение:

$$Y_1 = \overline{x_1} * \overline{x_2} * \overline{x_3} + x_1 * x_2 * \overline{x_3} + x_1 * \overline{x_2} * x_3$$

$$Y_2 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_3 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_4 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_5 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_6 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_7 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * \overline{x_3}$$

$$Y_8 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * x_3$$

$$Y_9 = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * x_2 * x_3$$

$$Y_{10} = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * \overline{x_2} * x_3$$

$$Y_{11} = \overline{x_1} * \overline{x_2} * x_3 + \overline{x_1} * x_2 * \overline{x_3} + x_1 * x_2 * x_3$$

Задание 6.

Составить две карты Карно для функций Y_5, Y_9 (из табл. 6.2.) от трёх переменных. Занести в карты Карно единицы в соответствующие

координатам переменных клетки. Создать и проанализировать контуры с единицами. Записать минимизированные выражения в базисе основного функционально полного набора («И», «ИЛИ», «НЕ») и базисе функции Шеффера («И-НЕ») для функций Y_5, Y_9 .

Решение:

$X_3 \backslash X_1 X_2$	00	01	11	10
0	1	1		
1			1	1

Рис. 6.6. Карта Карно для функции Y_3 .

$$Y_3 = (x_1 + x_3) * (\bar{x}_1 + \bar{x}_3);$$

$$Y_3 = \overline{\overline{(x_1 + x_3)} + \overline{(\bar{x}_1 + \bar{x}_3)}}$$

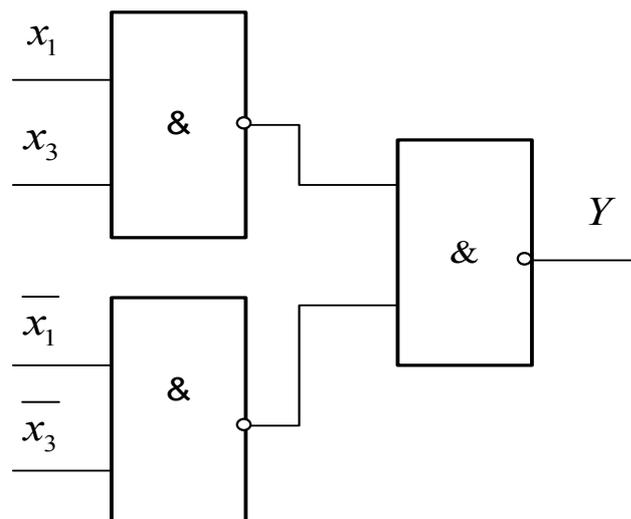


Рис. 6.7. Комбинационная схема для функции Y_5 .

Задание 7.

Составить две карты Карно для функций Y_3, Y_6 от трёх переменных. Занести в карты Карно нули в соответствующие координатам переменных клетки. Создать и проанализировать контура с нулями. Записать минимизированные выражения в базисе основного функционально полного набора («И», «ИЛИ», «НЕ») и базисе функции Пирса («ИЛИ-НЕ») для функций Y_3, Y_6 .

Задание 8.

Изобразите комбинационную схему из логических элементов, соответствующих выражению булевой функции Y_6 в базисах «И»-«ИЛИ»-«НЕ» и «ИЛИ»-«НЕ».

Решение:

$X_3 \backslash X_1 X_2$	00	01	11	10
0	0		0	
1	0		0	

Рис. 6.8. Карта Карно для функции Y_6 .

$$Y_6 = (x_1 + x_2) * (\bar{x}_1 + \bar{x}_2);$$

$$Y_6 = \overline{\overline{(x_1 + x_2)} + \overline{\overline{(x_1 + x_2)}}}$$

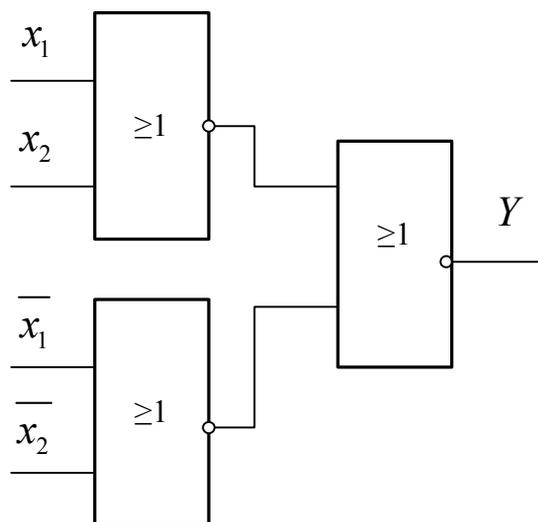


Рис.6.9. Комбинационная схема для функции Y_6 .

Задание 9.

Докажите, что функцию отрицания можно реализовать на элементах логических элемента Шеффера и Пирса.

Решение:

По законам тавтологии: $\overline{x + x} = \bar{x}$, т.е. элемент «НЕ» получается из элемента «ИЛИ-НЕ» подачей на оба входа переменной x .

По законам тавтологии: $\overline{x * x} = \bar{x}$ т.е элемент «НЕ» получается из элемента «И-НЕ» подачей на оба входа переменной x .

Задание 10.

Какая логическая функция реализована следующей комбинационной схемой (рис. 6.10.)?

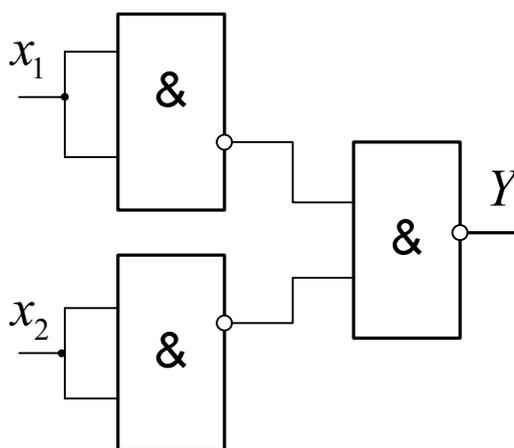


Рис. 6.10. Комбинационная схема к заданию 10.

Решение:

Данная комбинационная схема реализует логическую функцию ИЛИ (дизъюнкцию), это доказывается следствием из закона Де Моргана.

$$Y = \overline{x_1 * x_2} = x_1 + x_2$$

Задание 11.

Преобразуйте следующие функции так, чтобы реализующая их комбинационная схема состояла только из элементов «И-НЕ»:

$$Y = (x_1 * x_2 * x_3) + (x_4 * x_5 * x_6);$$

Решение:

$$Y = (x_1 * x_2 * x_3) + (x_4 * x_5 * x_6) = \overline{\overline{(x_1 * x_2 * x_3)} * \overline{(x_4 * x_5 * x_6)}}$$

Задание 12.

Преобразуйте следующую функцию так, чтобы реализующая комбинационная схема состояла только из элементов «ИЛИ-НЕ»:

$$Y = (x_1 + x_2 + x_3) * (x_3 + x_4 + x_5) * (x_6 + x_7)$$

Решение:

$$Y = \overline{\overline{(x_1 + x_2 + x_3)} + \overline{(x_3 + x_4 + x_5)} + \overline{(x_6 + x_7)}}$$

Задание 13.

Составьте таблицу истинности для схемы рис. 6.11 .

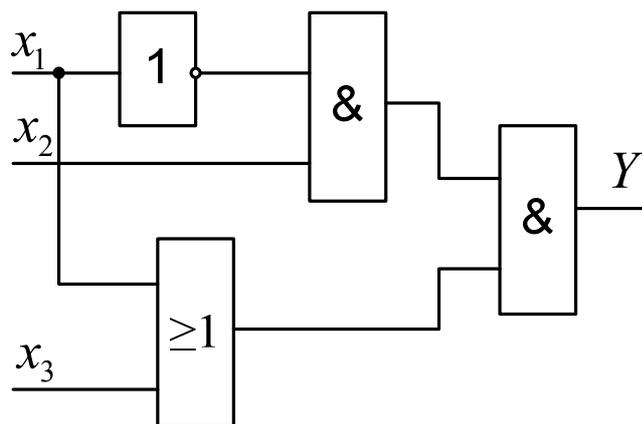


Рис. 6.11. Комбинационная схема к заданию 13.

Решение:

Таблица 6. Таблица истинности к заданию 13.

x_1	x_2	x_3	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Модуль 2. Базовые узлы и функциональные блоки дискретных устройств АСОИУ.

Глава 7. Регистры.

7.1. Назначение, классификация, параметры и типовое устройство регистров.

Регистром называется устройство, предназначенное для приема, кратковременного хранения и выдачи кодов слова, а также выполняющее логические преобразования над кодом слова.

Регистр – функциональный узел, объединяющий несколько однотипных триггеров. Помимо триггеров регистр может содержать вспомогательные логические схемы, количество и схема соединений которых зависит от количества разрядов в коде и набора операций, выполняемых регистром.

Помимо хранения кода слова регистры, в общем случае, обеспечивают выполнение следующих операций:

- установка регистра в нулевое состояние «сброс»;
- прием кода слова из другого устройства;
- передача кода слова в другое устройство;
- преобразование кода числа:
- преобразование прямого кода в обратный код и наоборот;
- преобразование последовательного кода слова в параллельный код и наоборот;
- сдвиг кода слова вправо или влево на требуемое число разрядов;
- выполнение некоторых логических операций.

Регистры, как правило, строятся на D-триггерах, т.к. эти триггеры позволяют производить запись информации однофазным кодом без предварительного "обнуления" по входу D.

Приём информации в регистр и выдача может осуществляться параллельно и последовательно. В первом случае слово представляется в виде параллельного кода.

Классификация регистров.

Различают следующие типы регистров:

- *регистры параллельного занесения;*
- *сдвигающие регистры;*
- *сдвигающие реверсивные регистры;*
- *универсальные регистры*

При записи и при считывании все разряды кода слова передаются одновременно, каждый разряд по своей кодовой шине.

При последовательной передаче кода слова, все разряды кода слова передаются последовательно во времени один за другим и строго в определенные дискретные моменты времени, совпадающие с управляющими сигналами.

Регистр с приёмом информации параллельным кодом.

Эти регистры предназначены для приёма, хранения и выдачи информации кода одного m -разрядного слова. Приём и выдача информации в регистре осуществляется в параллельном коде. Этот тип регистров ещё называют регистрами хранения. Регистры хранения очень распространены в схемах вычисления и управления.

Такой, m -разрядный, регистр можно представить как совокупность одноразрядных регистров, имеющих общие шины управления.

На рис.7.1. представлена схема 4-х разрядного регистра с приёмом информации параллельным однофазным кодом.

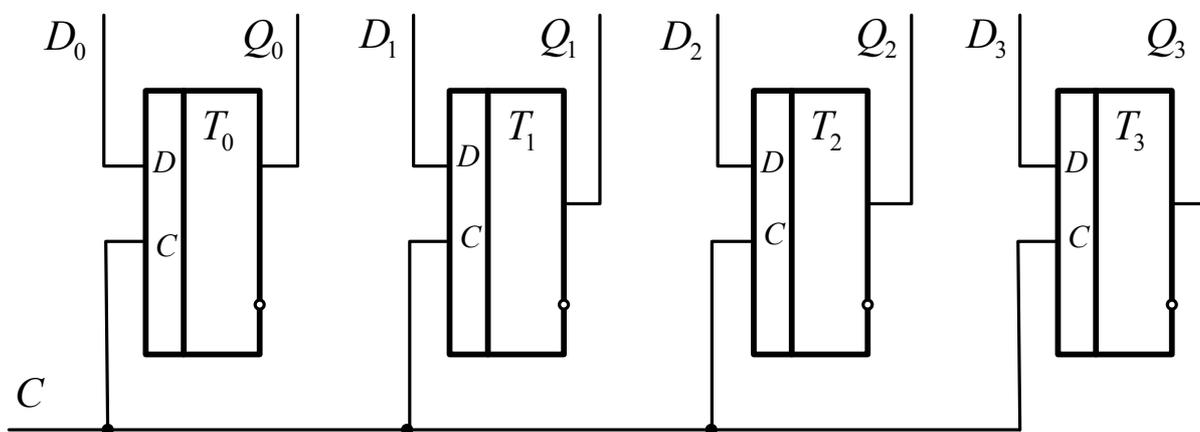


Рис.7.1 Схема 4-х разрядного регистра с приёмом информации параллельным однофазным кодом.

Регистр параллельного занесения кода с выдачей прямого, либо обратного кода представлен на рис.7.2

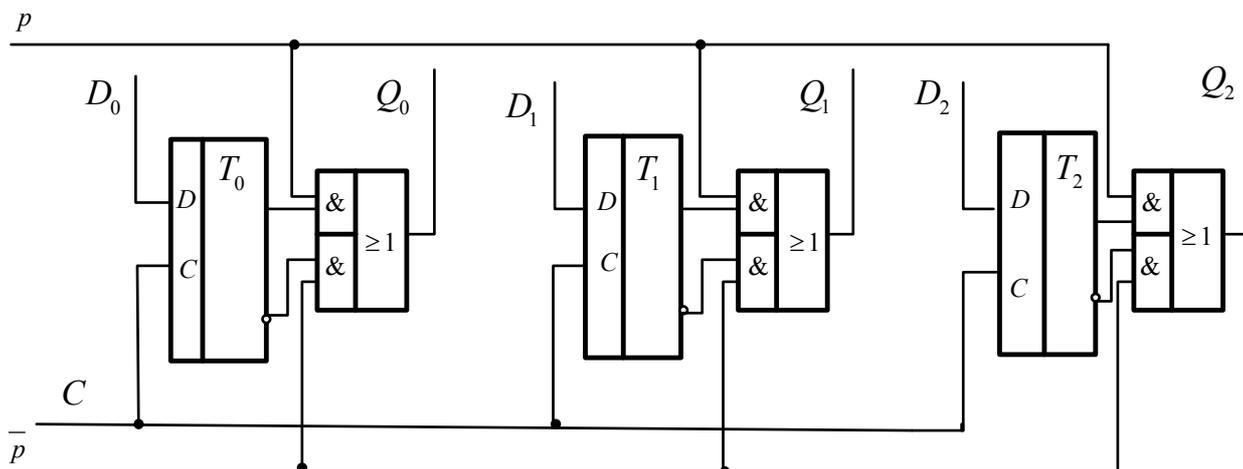


Рис. 7.2. Регистр параллельного занесения кода с выдачей прямого, либо обратного кода.

Схема данного регистра имеет выходную логику на элементах «2И-ИЛИ», которая позволяет считывать информацию из регистра, как в прямом, так и обратном коде в зависимости от значений разрешающего сигнала P .

Регистр выполнен на D - триггерах $T_0 - T_2$. Запись информации осуществляется по входам D в соответствии с уравнением D-триггера:

$$Q^{t+1} = \overline{C^t} Q^t + C^t D^t$$

Входное слово $X = x_3 x_2 x_1$ поступает на входы D- триггеров и при $C = 1$ записывается в регистр.

Считывание выходного слова $Y = y_3 y_2 y_1$ производится через логические элементы 2И-НЕ - с прямых выходов $Q_3 Q_2 Q_1$ или с инверсных выходов. При $P = 1$, считывание происходит в прямом коде, а при $P = 0$, считывание происходит в обратном коде.

Сдвигающие регистры.

В регистрах этого типа осуществляется **сдвиг слова влево или вправо** на заданное число разрядов. За один такт происходит сдвиг на один разряд.

На рис. 7.3 представлена схема сдвигающего регистра вправо, выполняющего операцию преобразования последовательного кода в параллельный код.

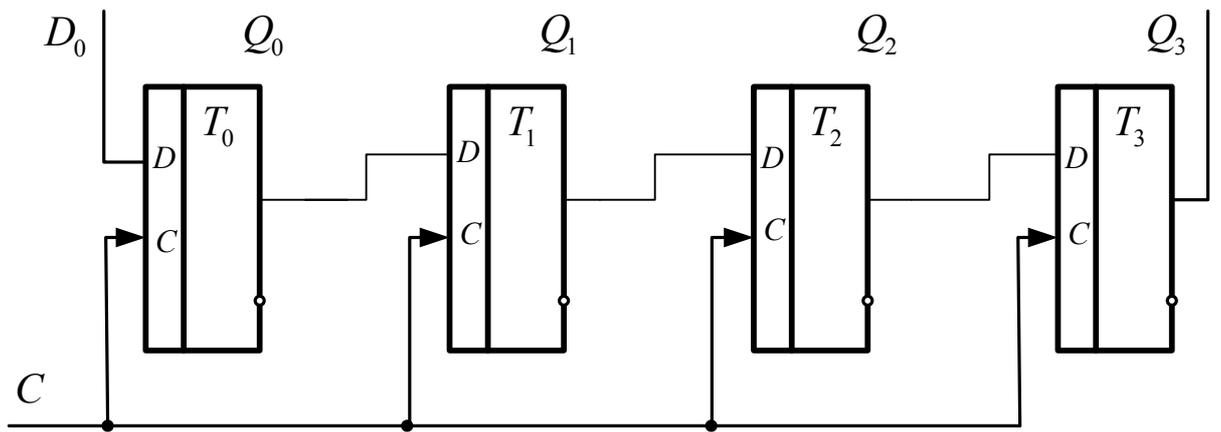


Рис. 7.3. Схема сдвигающего регистра вправо.

Число $X = x_1x_2x_3x_4$ в последовательном коде поступает на вход D-триггера T_1 младшим разрядом вперёд.

При поступлении 1-го ТИ x_4 записывается в T_1 . По второму ТИ x_4 переписывается в T_2 , а в T_1 записывается x_3 и т.д.

Таким образом, после четырёх тактирующих импульсов всё число окажется записанным в регистр и может быть считано в параллельном коде с выходов Y_1, Y_2, Y_3, Y_4 .

На рис. 7.4. изображена временная диаграмма занесения четырёхразрядного кода 1111. При этом на вход D_0 подаётся логическая единица в течение четырёх синхроимпульсов.

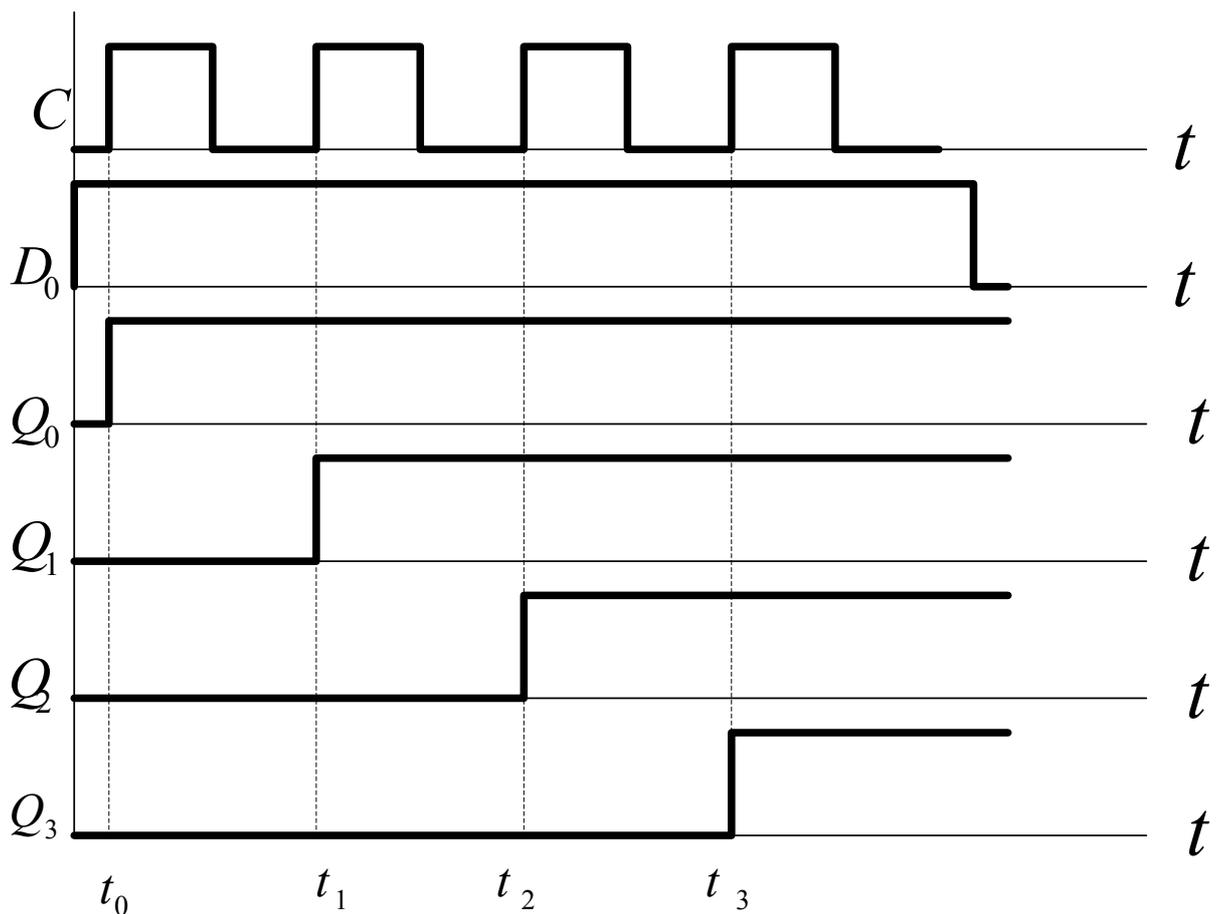


Рис. 7.4. Временная диаграмма занесения четырёхразрядного кода 1111 в регистр сдвига.

Сдвигающий регистр, выполняющий преобразование параллельного кода в последовательный.

Входное слово $X=x_3x_2x_1$ при $P1=[TI]=1$ и $P2=0$ записывается в $T_3T_2T_1$.

Затем при $P2=[TI]=1$ и $P1=0$ происходит перезапись информации из T_3 в T_2 , из T_2 в T_1 и т.д.

Таким образом, на выходе последнего разряда Q_1 будут последовательно появляться x_3, x_2, x_1 , т.е. входное слово X , представленное в параллельном коде будет преобразовано в выходное слово Y , представленное последовательным кодом. На рис. 7.5. представлена схема регистра сдвига влево.

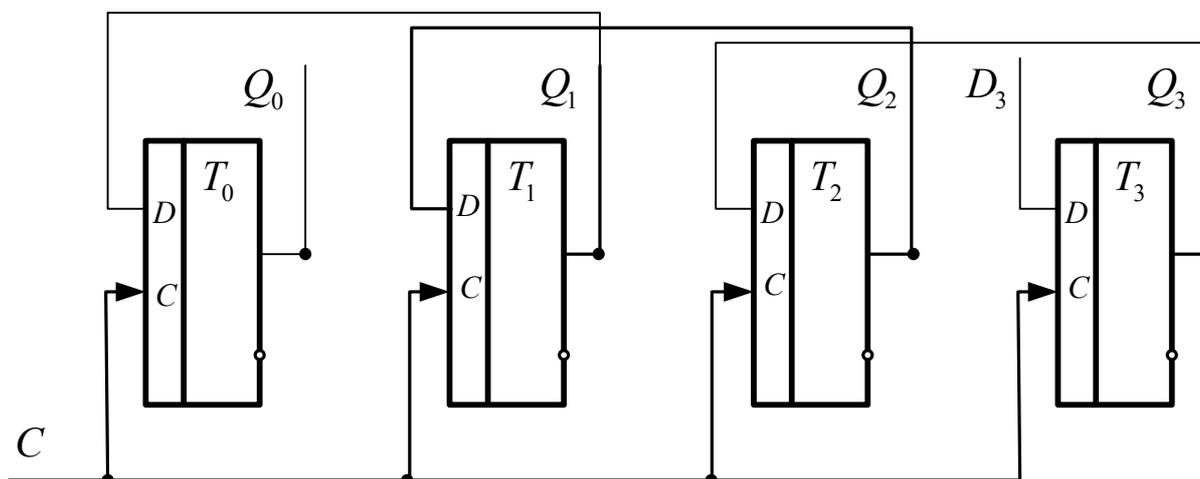


Рис.7.5. Схема регистра сдвига влево.

Реверсивные сдвигающие регистры.

Реверсивные регистры осуществляют *сдвиг кода либо вправо, либо влево* в зависимости от значений дополнительных управляющих сигналов.

Поскольку транзисторы и логические элементы способны передавать сигналы только в одном направлении с входа на выход (слева направо), то, для сдвига информации влево, необходимо информацию с выхода последующих триггеров по специально созданным цепям подавать на входы предыдущих триггеров и записывать их следующим тактовым сигналом. Это эквивалентно сдвигу информации влево.

Схема реверсивного регистра сдвига приведена на рис. 7.6.

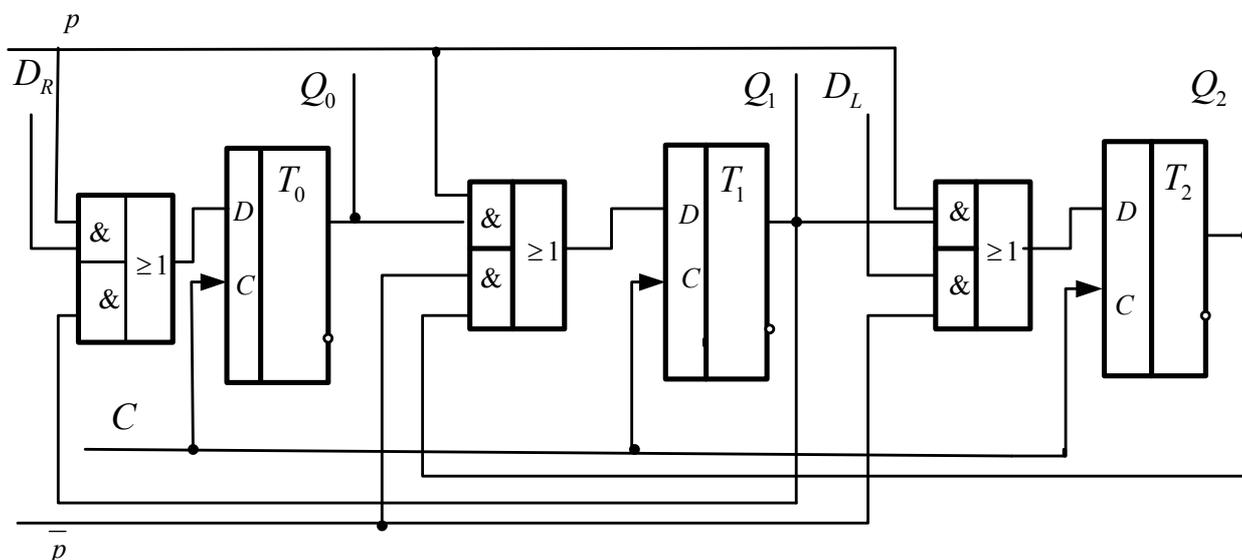


Рис. 7.6. Схема реверсивного регистра сдвига.

Если сигнал на входе направления сдвига $N=1$, то потенциал на входе « D_i » триггера определяется выходом Q триггера, стоящего слева от него. Если $N=0$, то – выходом триггера, стоящего справа.

Таким образом, при $P=1$ тактовые импульсы производят сдвиг информации вправо, а при $P=0$ – сдвиг информации влево.

7.5. Универсальный регистр.

Универсальный регистр объединяет в себе функции регистра параллельного занесения и реверсивного регистра.

Наиболее распространённой схемой управления режимами данного регистра является мультиплексор на 4-ре информационных входа, которым дополняется каждый разрядный триггер универсального регистра рис. 7.7.

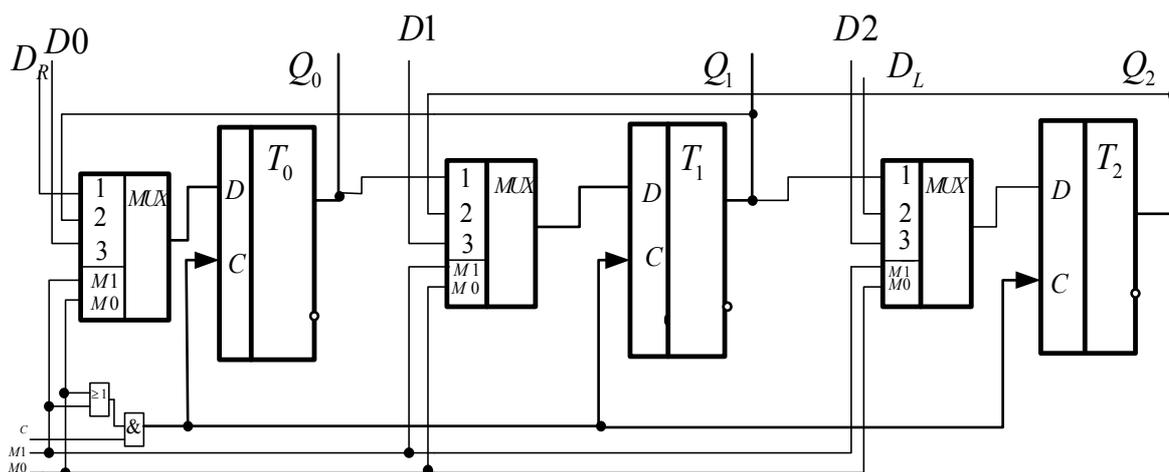


Рис. 7.7. Схема универсального регистра.

Значение управляющих сигналов универсального регистра представлены в таблице. 7.1.

Таблица 7.1. Значение управляющих сигналов универсального регистра

M1	M0	Режим регистра
0	0	Считывание
0	1	Сдвиг вправо
1	0	Сдвиг влево
1	1	Параллельное занесение

В режиме считывания происходит блокирование подачи сигнала синхроимпульса на все разряды с помощью с помощью комбинационной схемы, состоящей из логического элемента «ИЛИ», на входы которого подаётся кодовая комбинация сигналов управления режимами: $M1=0$ и $M0=0$. Нулевое значение выхода этого логического элемента блокирует прохождение сигнала C с помощью логического элемента «И».

7.6. Применение регистров.

Регистры применяются и как быстродействующая память в блоках регистровой памяти и как регистры управления, содержащие отдельные разряды как флаги условий и признаков. Наибольшее распространение получили универсальные регистры, реализуемые в большом количестве БИС.

Подобные регистры получили наименование многорежимных (многофункциональных), они способны выполнять набор микроопераций. Многорежимность достигается добавлением комбинационных схем, необходимых для выполнения различных операций.

Регистры, имеющие разнотипные вход и выход, служат основными блоками преобразователей параллельных кодов в последовательные и обратно.

7.7. Типовые примеры проектных решений с использованием регистров.

Во многих выходных каскадах портов применяются протоколы последовательных интерфейсов, поэтому очень часто на выходе шинного параллельного интерфейса возникает задача преобразования параллельного выводимого кода данных в последовательный код. Типовая схема преобразования может быть реализована со сдвигающим регистром, имеющим входы параллельной загрузки кода. На рис. 7.8. представлена схема преобразователя параллельного кода в последовательный.

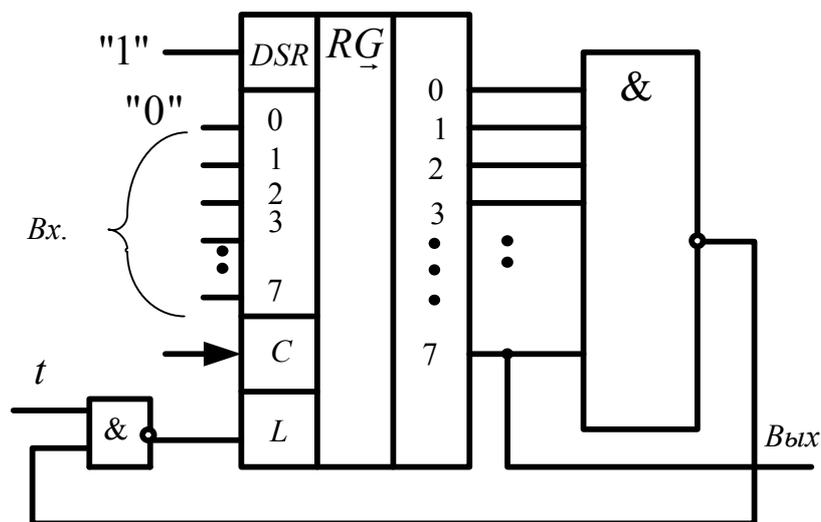


Рис. 7.8. Схема преобразователя параллельного кода в последовательный код.

Регистр принимает параллельный код с внутренней шины параллельным кодом. Затем, под управлением тактовых синхроимпульсов C осуществляется сдвиг слова вправо (для условного обозначения сверху вниз). Сдвиги выводят последовательно код на выход № 7. Вслед за информационными разрядами идёт ноль, после которого цепочка единиц. Пока ноль не выведен из регистра, на выходе элемента «И» действует единичный сигнал. После вывода нуля все входы элемента «И» становятся единичными, его выход приобретает нулевое значение и иницируется загрузка нового параллельного кода.

7.8. Контрольные вопросы.

Вопросы категории «Помнить»:

- 7.8.1. Перечислите функции, выполняемые регистрами.
- 7.8.2. На основе каких устройств строятся регистры?

Вопросы категории «Понимать»:

- 7.8.3. Возможна ли реализация сдвигающего регистра на синхронных динамических RS-триггерах?
- 7.8.4. Какие устройства позволяют реализовать схему реверса у реверсивных регистров?

7.8.5. Являются ли синонимами наименования операций: «сброс регистра» и «обнуление регистра»?

Вопросы категории «Применять»:

7.8.6. Изобразите схему 3-х разрядного регистра для последовательного ввода данных и возможностью ввода данных параллельным кодом. При построении использовать D-триггеры.

7.8.7. Каким схемным решением обеспечивается неизменность хранимого в регистре кода при его выдаче с выходов регистра, притом, что на входе регистра может быть сигнал наведённой помехи?

Ответы на контрольные вопросы.

7.8.1. Регистры выполняют следующие основные действия:

установка регистра в нулевое состояние «сброс»;

прием кода слова из другого устройства;

передача кода слова в другое устройство;

преобразование кода числа:

преобразование прямого кода в обратный код и наоборот;

преобразование последовательного кода слова в параллельный код и наоборот;

сдвиг кода слова вправо или влево на требуемое число разрядов;

7.8.2. Регистры строятся на основе синхронных триггеров.

7.8.3. Реализация сдвигающего регистра возможна при организации связей по прямому и инверсному выходу разрядов триггеров.

7.8.4. Режим реверса обеспечивается переключением связей между разрядами сдвигающего регистра мультиплексорами.

7.8.5. Сброс регистра и обнуление регистра обозначает одно и тоже действие.

7.8.6.

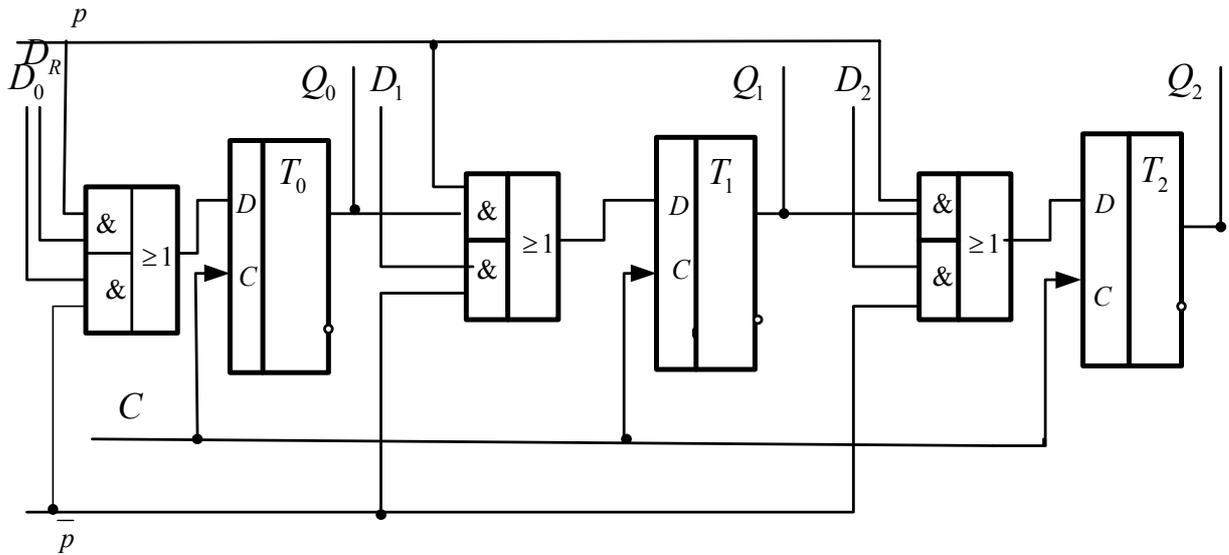


Рис. 7.9. Схема регистра для последовательного ввода данных и возможностью ввода данных параллельным кодом.

7.8.7. Для гарантии сохранения содержимого регистра задействована схема блокировки прохождения синхроимпульса.

7.9. Задачи для самостоятельного решения.

7.9.1. Изобразите временную диаграмму занесения в сдвигающий регистр кода 1001. На диаграмме показать синхросигнал, сигнал на входе D крайнего триггера регистра, сигналы на выходах четырёх разрядов регистра.

Решение:

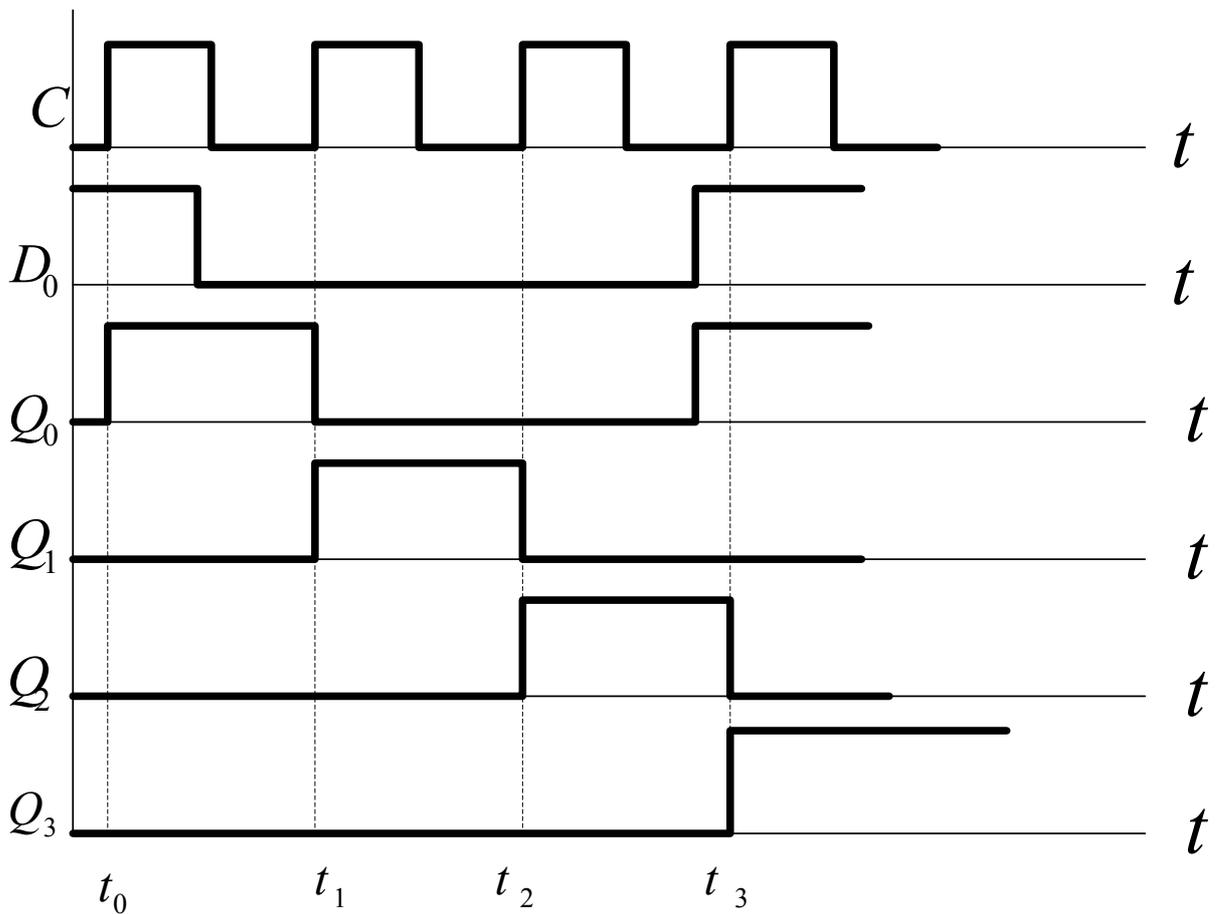


Рис. 7.9. Временная диаграмма занесения в сдвигающий регистр кода 1001.

7.9.2. Как выполнить поразрядное логическое ИЛИ с помощью регистра, если он составлен из триггеров RS-типа?

Для этой операции необходимо добавить к каждому разряду регистра комбинационную схему для выработки последовательности подачи управляющих сигналов.

Решение:

Вход S и R каждого разряда регистра дополняется несложными комбинационными схемами. Регистр приобретает вид рис. 7.10.

Порядок выполнения операции поразрядного сложения следующий:

Сначала по значению установочного кода записывается первый аргумент (режим D-триггера, т.е. парафазным кодом при воздействии синхроимпульса C1. По синхроимпульсу C2 заносится второй аргумент B. Если B=1 и триггер был в состоянии «0», триггер установится в 1. Если триггер был в 1, его состояние меняться не будет.

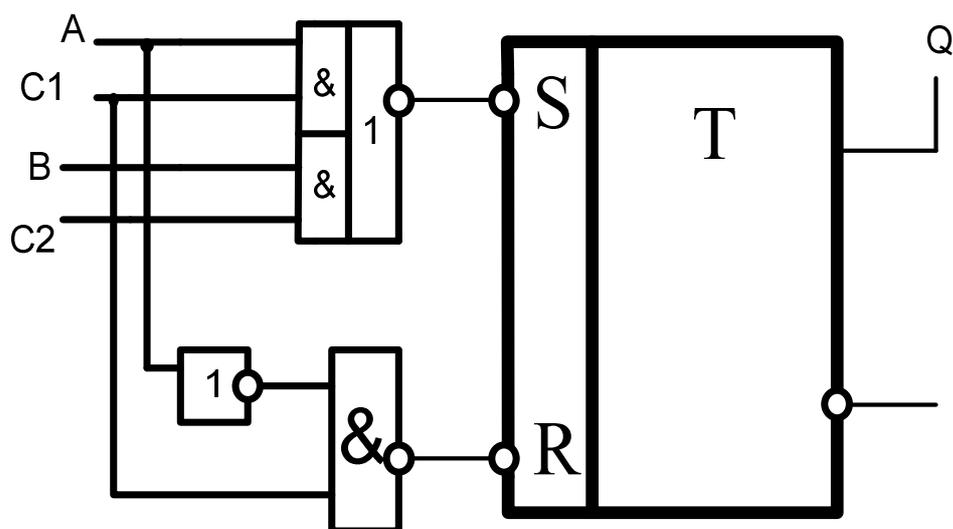


Рис. 7.10. Один из разрядов регистра с дополненными комбинационными схемами.

Глава 8. Счетчики.

8.1. Назначение, классификация, параметры и типовое устройство счётчиков.

Одной из наиболее распространенных операций, выполняемых в вычислительных устройствах цифровой обработки информации, является подсчет (или счёт) числа сигналов импульсного или потенциального вида.

Счётом называется повторяемое прибавление (или вычитание) единицы к предыдущему числу. Начальное значение, которое в большинстве вариантов начинается с нуля, прибавляется (или вычитается) единица до достижения счётчиком граничного конечного значения счёта.

Узел вычислительных устройств, предназначенный для подсчета числа входных сигналов, называется счетчиком.

Счетчики широко применяются почти во всех цифровых устройствах автоматики и вычислительной техники. В вычислительных устройствах счетчики используются: для подсчета шагов программы, для подсчета циклов сложения и вычитания при выполнении арифметических

операций, для преобразования кодов, в делителях частоты и распределителях сигналов и т.д.

Для любых задач, связанных со счётом, могут быть синтезированы электронные счётчики. Счётчики, работающие в двоичной системе счисления, получили наименование бинарных, либо двоичных счётчиков. Счётчики, работающие с двумя, тремя, четырьмя и большим количеством входных сигналов, не имеют практического значения. Счётчики могут работать в различных системах счислений и с различными кодами.

Классификация счетчиков по основным признакам:

- по системе счисления счетчики делятся на: двоичные, двоично-десятичные, десятичные, счетчики с основанием системы счисления неравным 2 и 10 (пересчетные схемы).
- по реализуемой операции счетчики подразделяются на: суммирующие, вычитающие и реверсивные. Реверсивные счётчики имеют управляемое переключение направление счёта (увеличение или уменьшение).
- по моменту переключения составляющих счётчик триггеров, счётчики подразделяются на синхронные и асинхронные счётчики.

Основные параметры счетчиков :

- модуль счета или коэффициент пересчета счетчика « $K_{сч}$ » характеризует число (количество) устойчивых состояний, в которых может находиться n - разрядный счетчик, т. е. предельное число входных сигналов, которое может быть подсчитано счетчиком.

Двоичный n - разрядный счетчик имеет 2^n различных состояний.

Число разрядов двоичного счетчика можно определить из выражения:

$$n \geq \log_2 K_{сч} \quad (8.1)$$

где $K_{сч}$ - коэффициент пересчета;

n - ближайшее целое число, удовлетворяющее данному неравенству.

- максимальная частота поступления входных сигналов $f_{\text{сч max}}$ - это частота, при которой счетчик еще сохраняет работоспособность.

Она определяется, как правило, максимально допустимой частотой переключения триггера младшего разряда счетчика.

8.2. Синтез счетчиков и пересчетных схем на универсальных

D– триггерах и JK – триггерах.

8.2.1. Синтез асинхронных счётчиков.

Простейшим счетчиком является триггер со счетным входом, считающий сигнал по модулю 2, т.е. осуществляющий подсчет и хранение результата подсчета не более 2-х сигналов. Соединяя определенным образом несколько счетных триггеров, можно получить схему многоразрядного счетчика.

В настоящее время в составе большинства самых распространенных серий логических микросхем 133, K133, 155, K155 и др. имеются широко используемые D и JK -триггеры.

При использовании D - триггеров в качестве счетных, его инверсный выход соединяют со своим входом D. Счётный режим возможен только у D-триггеров динамического типа.

Представление счётчика цепочкой счётных триггеров справедливо как для суммирующего, так и для вычитающего вариантов, поскольку закономерность по соотношению частот переключения разрядов сохраняется как при просмотре таблицы сверху, так и снизу. Различия при этом состоят в направлении переключения предыдущего разряда, вызывающего переключение следующего.

При прямом счёте (суммирование) следующий разряд переключается при переходе предыдущего в направлении из 1 в 0, а при обратном счёте – при переключении из 0 в 1.

Суммирующий асинхронный счетчик на D-триггерах получается, если инверсный выход предыдущего триггера \bar{Q} соединить со входом C последующего триггера.

Схема асинхронного 4-х разрядного суммирующего счетчика на D-триггерах приведена, на рис. 8.1.

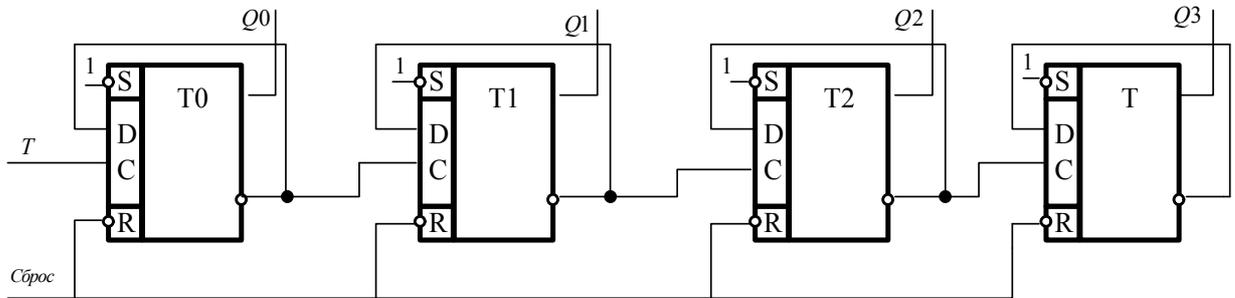


Рис. 8.1. Схема асинхронного 4-х разрядного суммирующего счетчика на D-триггерах.

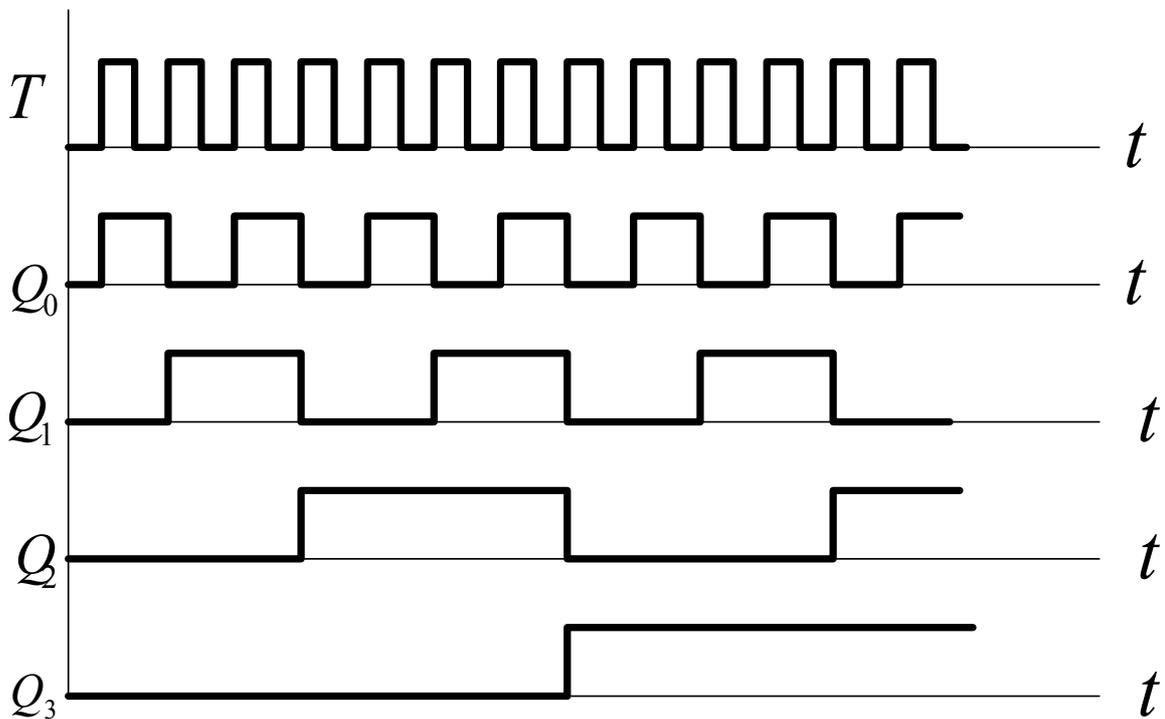


Рис. 8.2. Временная диаграмма асинхронного 4-х разрядного суммирующего счетчика на D-триггерах.

Для построения вычитающего счетчика на D-триггерах прямой выход предыдущего триггера соединяют со входом синхроимпульса C последующего триггера.

Схема асинхронного 4-х разрядного вычитающего счетчика на D-триггерах приведена на рис. 8.3.

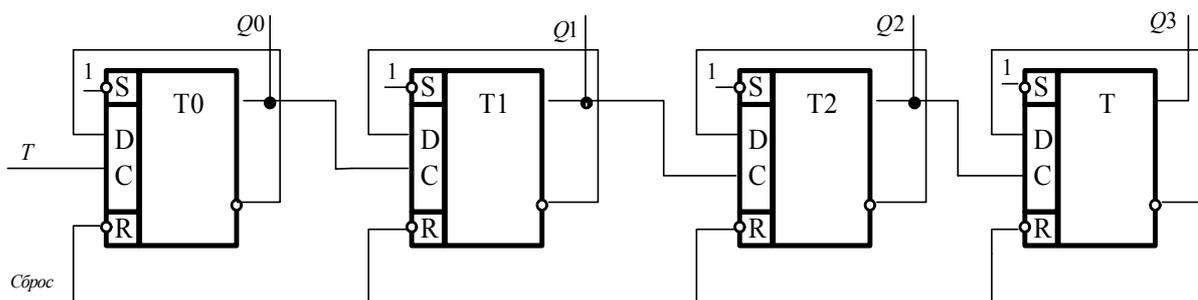


Рис. 8.3. Схема асинхронного 4-х разрядного вычитающего счетчика на D-триггерах.

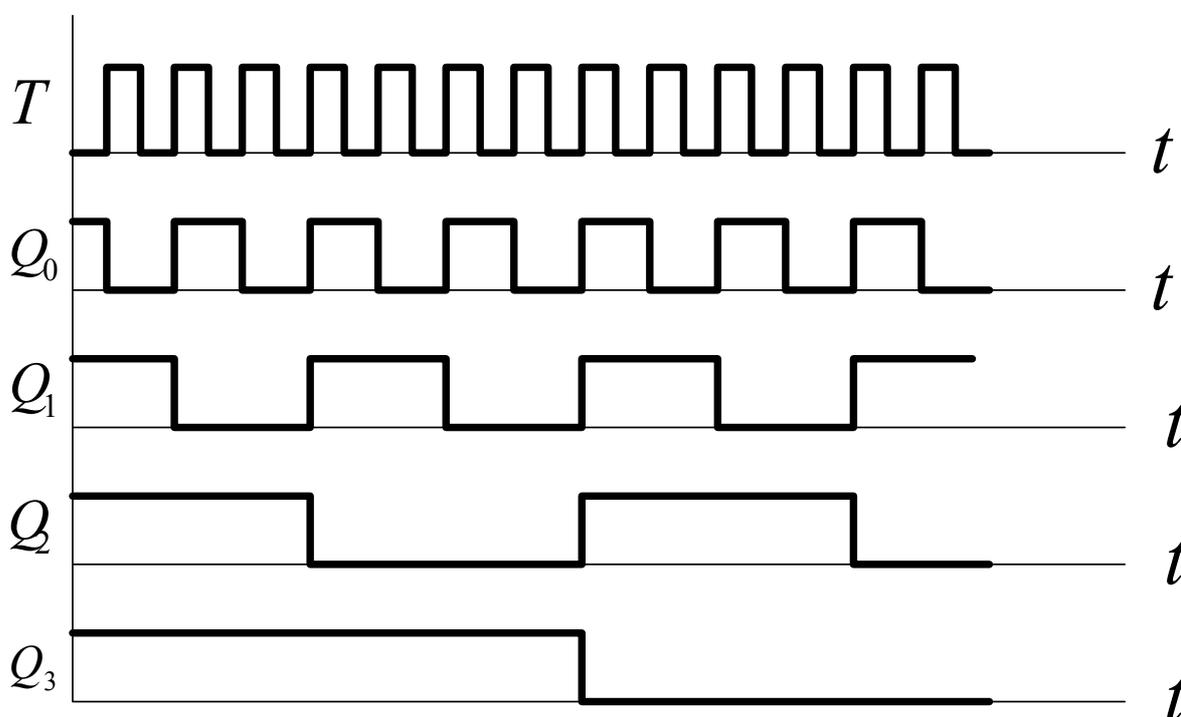


Рис. 8.4. Временная диаграмма асинхронного 4-х разрядного вычитающего счетчика на D-триггерах.

Для построения суммирующего асинхронного счетчика на JK-триггерах необходимо соединить прямые выходы предыдущих триггеров со входом синхроимпульса С последующих триггеров.

JK-триггер должен находиться в счётном режиме, при котором должны подаваться сигналы логической единицы одновременно на входы J и K. Схема асинхронного 4-х разрядного суммирующего счетчика на JK-триггерах приведена на рис. 8.5.

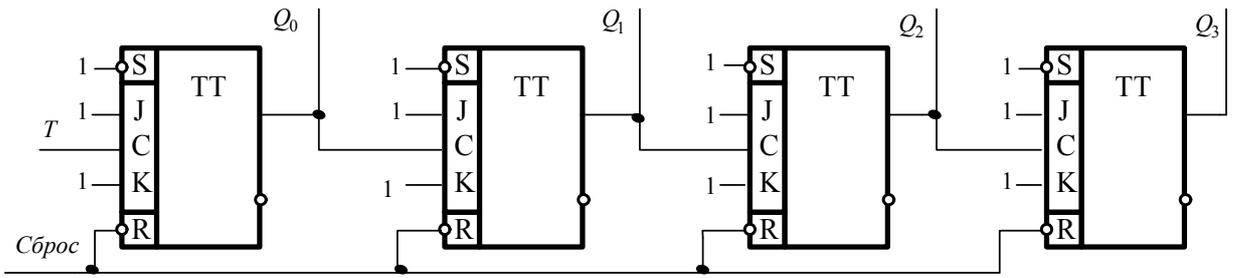


Рис. 8.5. Схема асинхронного 4-х разрядного суммирующего счетчика на JK – триггерах.

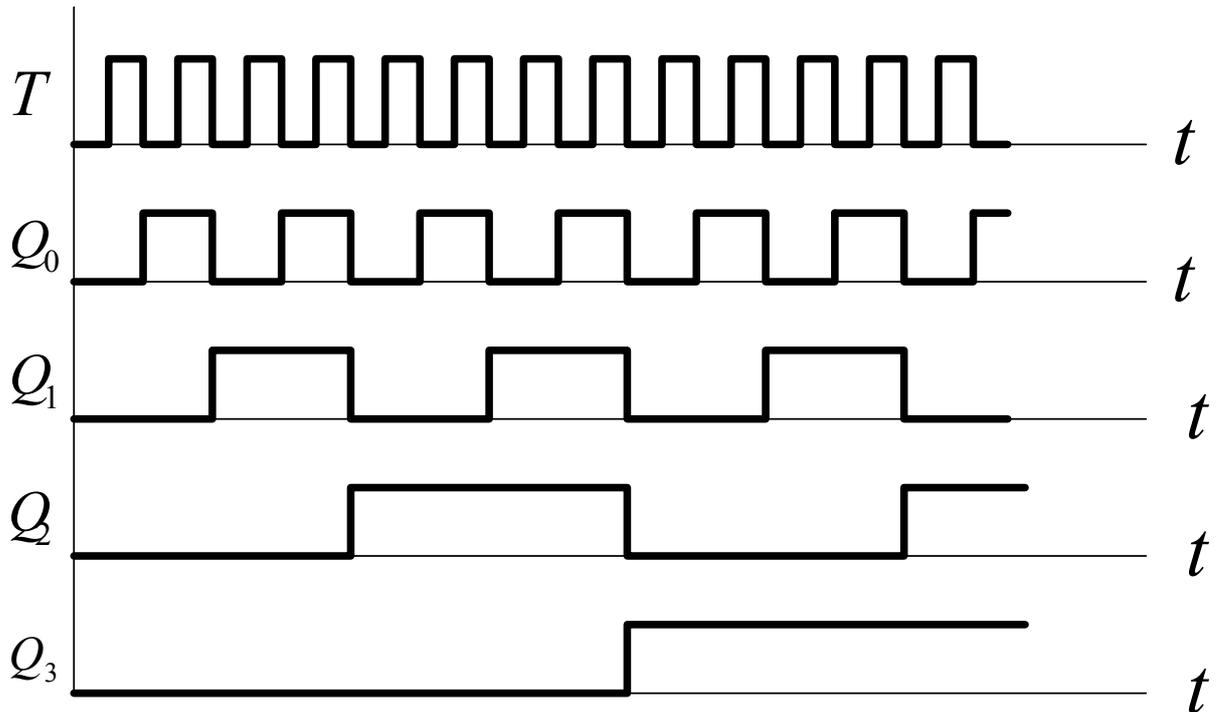


Рис. 8.6. Временная диаграмма асинхронного 4-х разрядного суммирующего счетчика на JK – триггерах.

Вычитающий асинхронный счетчик на JK-триггерах можно получить, если инверсный выход предыдущего триггера соединить со входом синхроимпульса C последующего триггера. Схема такого счётчика представлена на рис. 8.6.

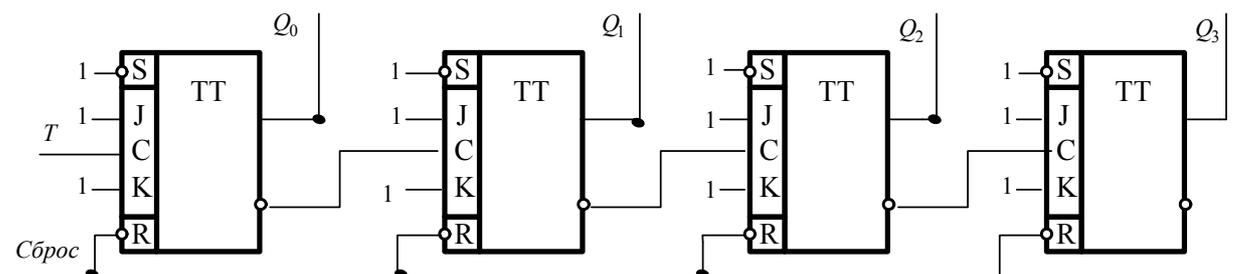


Рис. 8.7. Схема асинхронного 4-х разрядного вычитающего счетчика

на JK –триггерах.

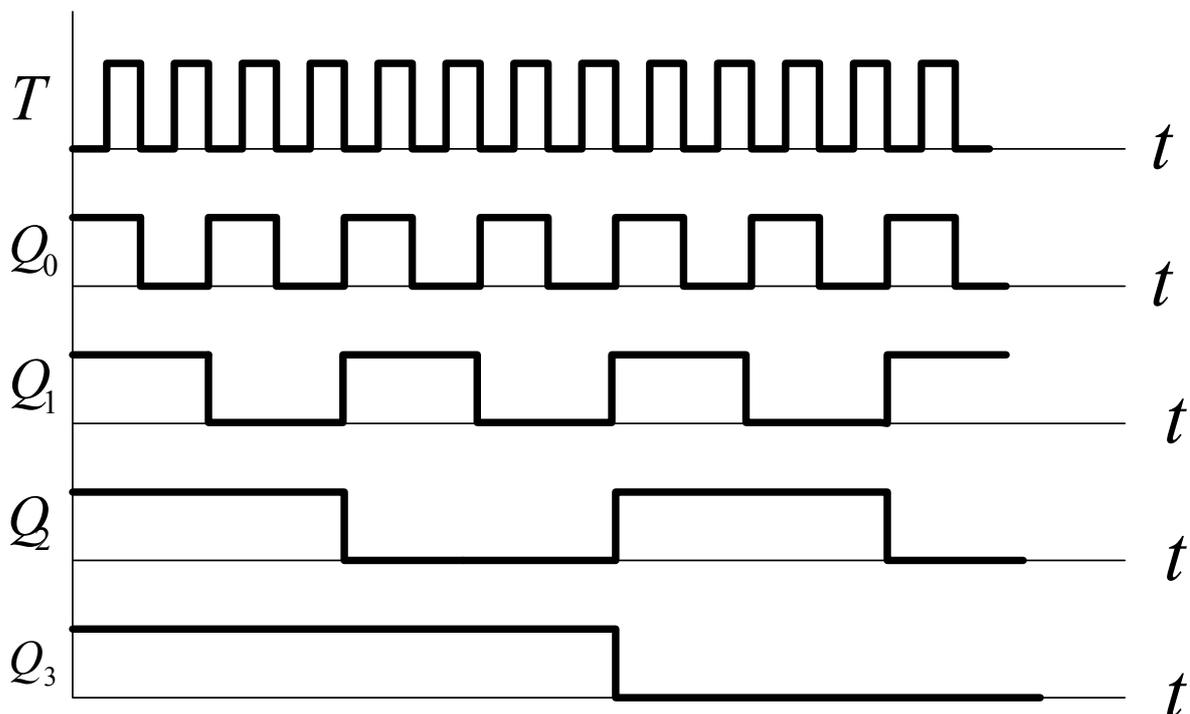


Рис. 8.8. Временная диаграмма асинхронного 4-х разрядного вычитающего счетчика на JK –триггерах.

Реверсивные счетчики осуществляют подсчет сигналов как в прямом, так и в обратном направлении, т.е. они могут работать в режиме сложения и в режиме вычитания сигналов.

Для построения реверсивных счетчиков необходимо предусмотреть схемы, пропускающие сигналы на вход следующих триггеров либо с инверсионного (при суммировании), либо с прямого (при вычитании) выходов предыдущего триггера.

Схема асинхронного 3-х разрядного реверсивного счетчика на D-триггерах со схемой управления прямым и обратным счетом приведена на рис.8.9.

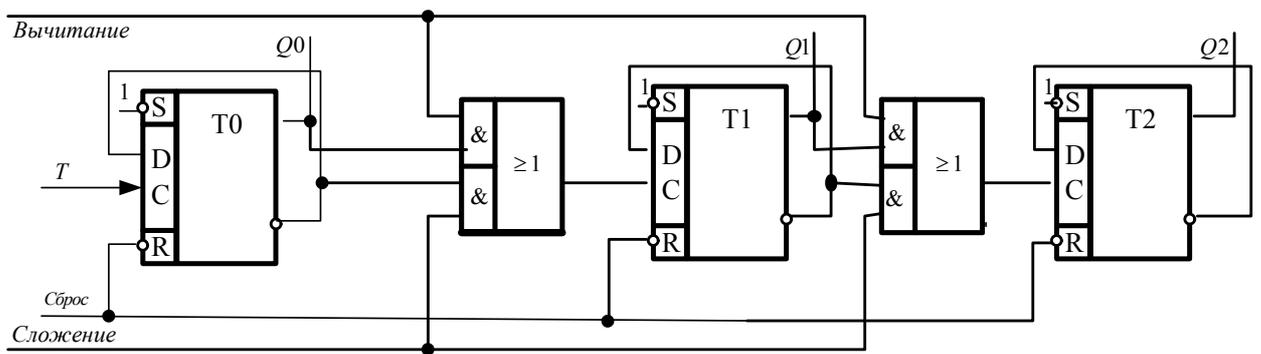


Рис. 8.9. Схема асинхронного 3-х разрядного реверсивного счетчика на D-триггерах со схемой управления прямым и обратным счетом.

Асинхронные реверсивные последовательные счетчики на **JK**-триггерах строятся аналогично реверсивным счетчикам на **D**-триггерах.

Схема асинхронного последовательного реверсивного счетчика на **JK** - триггерах со схемой управления прямым и обратным счетом на **JK**-триггере представлена на рис. 8.10.

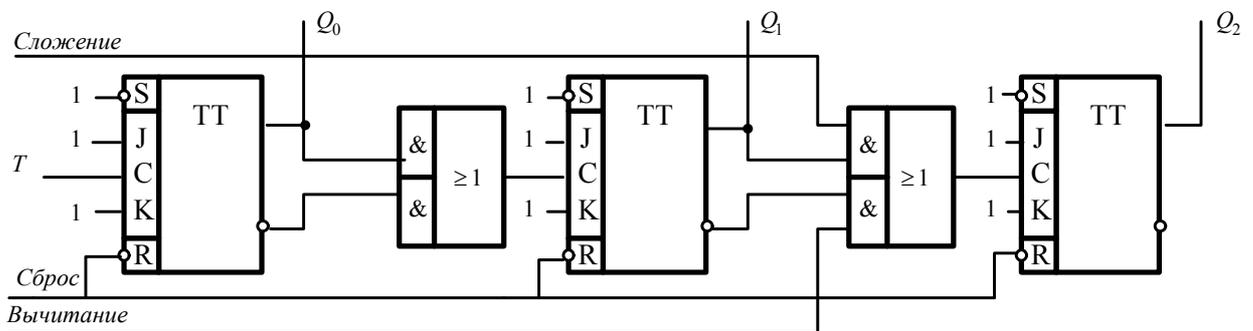


Рис. 8.10. Схема асинхронного последовательного реверсивного счетчика на **JK** - триггерах со схемой управления прямым и обратным счетом на **JK**-триггере.

Рассмотренные выше счетчики имели коэффициент пересчета равный 2^n , где n - число разрядов счетчика. Однако на практике возникает необходимость в счетчиках, коэффициент пересчета которых отличен от значения 2^n . Очень часто, например, применяются счетчики с $K_{сч} = 3, 10$ и т.д., т.е. счетчики, имеющие соответственно 3, 10 и т.д. устойчивых состояний.

Принцип построения таких счетчиков заключается в исключении “лишних” устойчивых состояний у счетчика с $K_{сч} = 2^n$, т.е. в организации схем, запрещающих некоторые состояния.

Число запрещенных состояний для любого счетчика можно определить из следующего выражения:

$$m = 2^n - K_{сч} \quad (8.2)$$

где m - число запрещенных состояний;

$K_{сч}$ - требуемый коэффициент счета;

2^n - число устойчивых состояний двоичного счетчика.

Так, например, у счетчика с $K_{сч} = 3$, который строится на двух счетных триггерах, и у счетчика с $K_{сч} = 10$, который строится на четырех счетных триггерах, число избыточных состояний согласно вышеприведенной формуле равно 1 и 6 соответственно.

Рассмотрим способ построения счетчика с естественным порядком счета, у которого уменьшение числа устойчивых состояний достигается за счет сбрасывания счетчика в нулевое состояние при записи в него заданного числа сигналов.

В соответствии с этим способом к счетчику добавляется логическая схема, проверяющая условие: «код на счетчике изображает число равное $K_{сч}$ и в зависимости от результата проверки направляет входной сигнал либо на шину "установка 0", либо на суммирование к записанному коду.

Это условие может быть проверено n -входной схемой И, связанной с прямыми выходами тех триггеров, которые при записи в счетчике числа $K_{сч}$ должны находиться в состоянии «1» и с инверсными выходами триггеров, которые в этом случае должны находиться в состоянии «0».

Практически число входов элемента "И" можно сократить, связав его с прямыми выходами, т.к. сочетание единиц в записи кода числа $K_{сч}$ может повториться только в недозволённых кодах больших $K_{сч}$.

Рассмотрим синтез схемы подобного счетчика на примере $K_{сч}=10$, т.е. счетчик должен иметь 10 состояний - от 0 до 9 в десятичной системе и от 0000 до 1001 в двоичной системе. Сначала определяется разрядность счетчика

$$n \geq \log_2 K_{сч} = \log_2 10 \approx 3.35 \quad (8.3)$$

Полученное значение n округляется до ближайшего целого числа, т.е. $n=4$. Затем определяется какие разряды счетчика будут находиться в единичном состоянии при записи в счетчик $K_{сч}$.

$$K_{сч} = 10_{10} = 1010 = Q_4 \overline{Q_3} Q_2 \overline{Q_1} \quad (8.4)$$

Прямые выходы этих разрядов заводятся на входы логической схемы "И" и далее в цепь установки "0". Таким образом, при достижении счетчиком значения $K_{сч}$ он автоматически возвращается в состояние 0000 и счет начинается снова.

На рис. 8.11 приведена схема асинхронного суммирующего счетчика с $K_{сч}=10$ построенного на JK - триггерах.

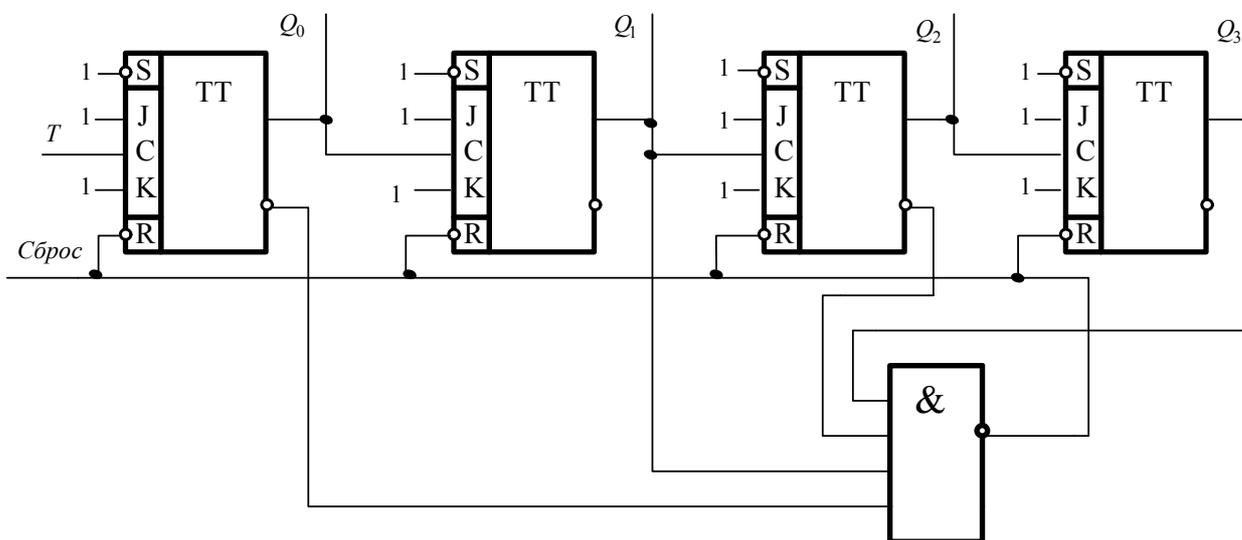


Рис. 8.11. Схема асинхронного суммирующего счетчика с $K_{сч}=10$, построенного на JK - триггерах.

8.2.2. Синтез синхронных счетчиков на основе JK - триггера.

Особенностью синхронного счётчика является то, что счётный сигнал подаётся по шине одновременно на синхровходы триггеров всех разрядов.

Логика переключения счётчика управляется переводом триггера данного разряда в счётное состояние (логические единицы на входах J и K), либо иное состояние, когда изменение состояния триггера не меняется при наличии счётного сигнала.

Во многих схемах J-K триггер реализуется с наличием нескольких входов J и входов K, объединённых конъюнктурами (логический элемент «И»).

Из анализа последовательности двоичных состояний счётчика можно сделать вывод, что i - тый разряд счётчика должен поменять своё состояние при приходе очередного счётного импульса только тогда, когда все разряды более младшие находятся в состоянии логических единиц.

Самый младший разряд счётчика должен менять своё состояние на каждый счётный импульс. Счётное состояние первого разряда обеспечивают логические единицы на всех трёх входах J и K.

На рис. 8.12. показана схема трёхразрядного синхронного суммирующего счётчика.

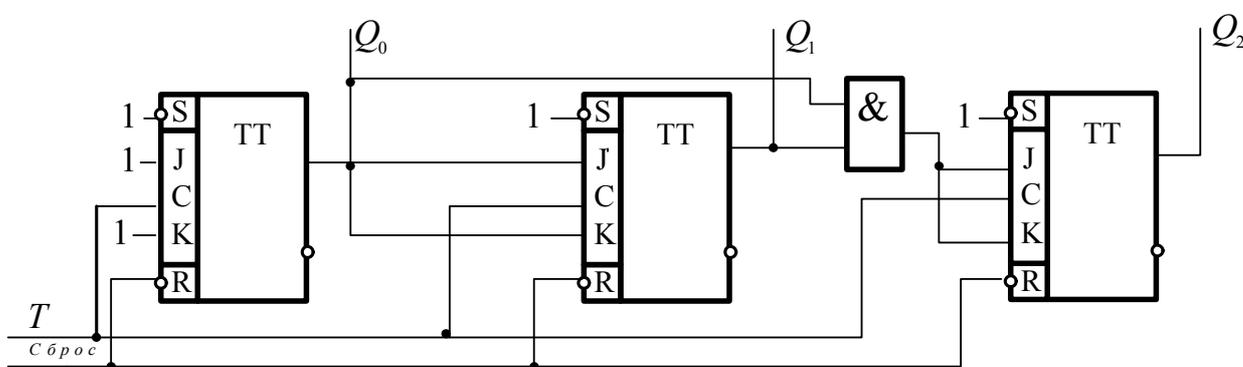


Рис. 8.12. Схема трёхразрядного синхронного суммирующего счётчика.

В случае построения синхронного вычитающего счётчика, i - тый разряд счётчика должен поменять своё состояние при приходе очередного счётного импульса только тогда, когда все разряды более младшие находятся в состоянии логических нулей, поэтому анализируются на совпадение по логике «И» инверсные выводы триггеров. Схема вычитающего счётчика представлена на рис. 8.13.

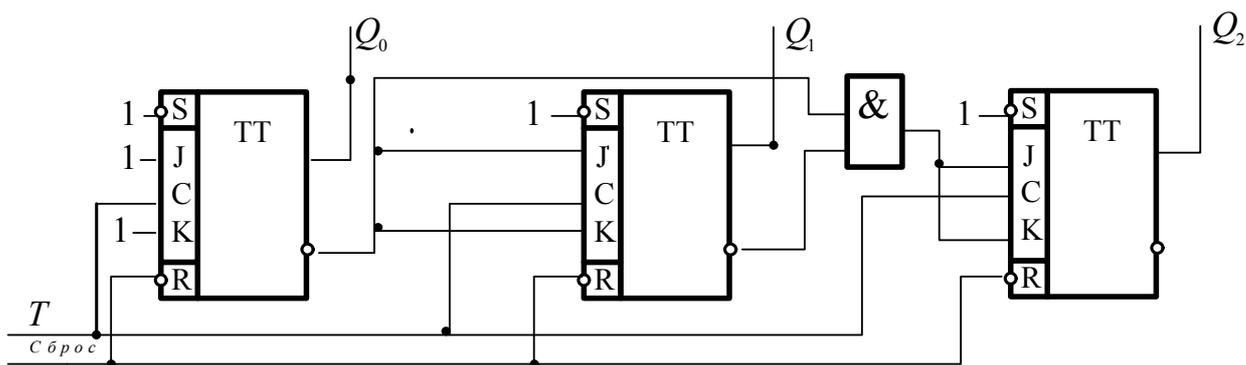


Рис. 8.13. Схема трёхразрядного синхронного вычитающего счётчика.

Итоги изучения главы 8.

Помнить:

- наименования типов счётчиков, их условно-графические отображения и основные схемотехнические отличия.

Понимать:

- логику применения счётных триггеров и топологию соединения триггеров в многоразрядных счётчиках.
 - различия в управлении переключением разрядов счётчиков асинхронного и синхронного типов.

Применять:

- счётчики для включения в топологии схем более сложного уровня для реализации подсчёта импульсов и сигналов.

8.3. Типовые примеры проектных решений.

8.2.1. Необходимо реализовать часы с индикацией часов. Будем полагать наличие отсчётного сигнала, подаваемого один раз в час. Индикацию необходимо вывести на два шестнадцатеричных индикатора.

Для решения данной задачи целесообразно применить два десятичных счётчика для удобства отображения цифр младшего и следующего десятичного разрядов на индикаторах без применения дешифраторов.

Счётчик младшего десятичного разряда должен показывать числа от 1 до 9, если старший десятичный разряд равен 0 и перебирать состояния кодов от 0 до 2, если старший разряд стал равен 1. По достижению значения 1 на старшем счётчике и 2 на младшем счётчике старший переходит в 0, а младший в 1 и счёт начинается снова. На рис. 8.13. приведена структура часов, смоделированная с помощью пакета Electronics Workbench.

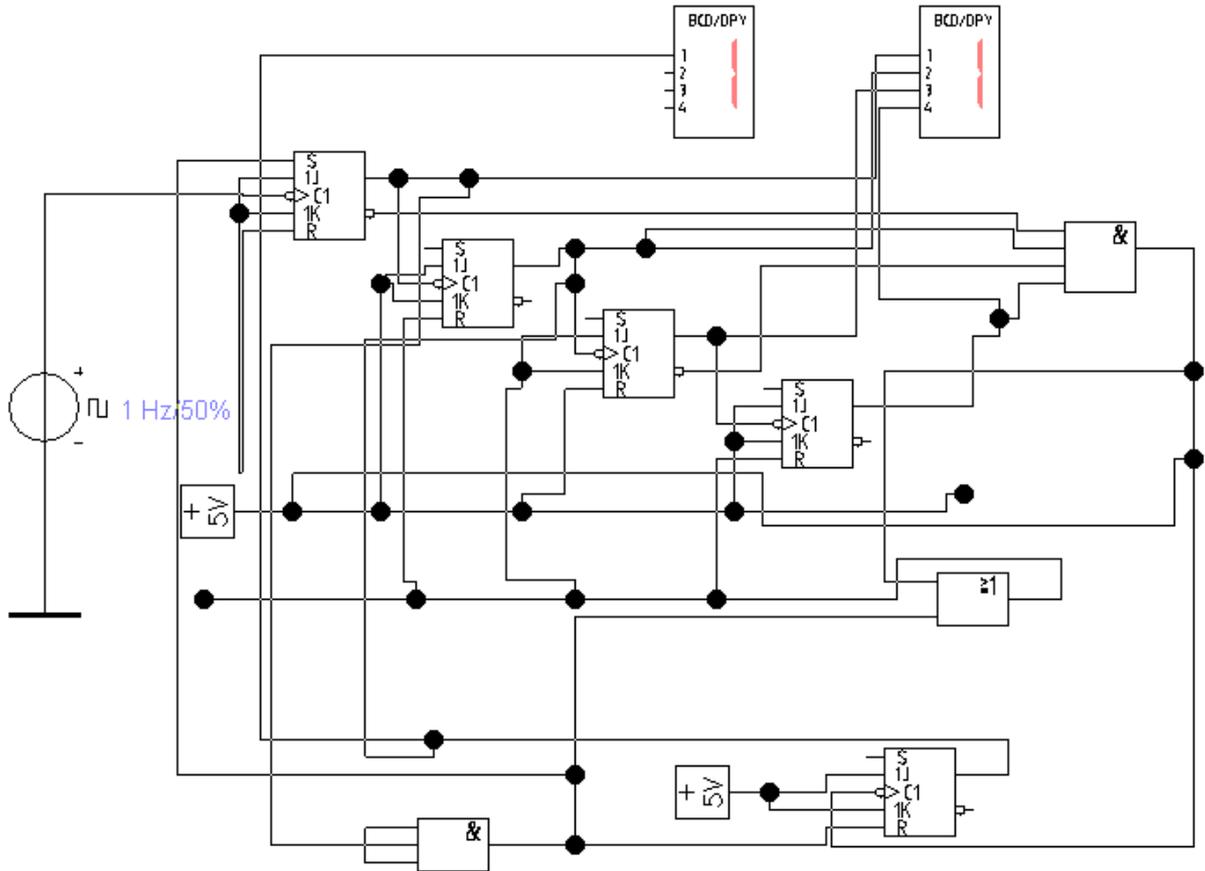


Рис. 8.13. Схема часов с индикацией часов от 0 до 12.

8.4. Контрольные вопросы.

Вопросы категории 1. «Помнить».

- 8.4.1. Сформулируйте определение понятия «модуль счётчика».
- 8.4.2. Поясните свойство счётчика, если он именуется реверсивным.

Вопросы категории 2. «Понимать».

- 8.4.3 Чем различаются асинхронные и синхронные счётчики?

8.4.4. Как преобразовать 4-х разрядный суммирующий счётчик в счётчик по модулю 10 ?

8.4.5. На какие входы счётчика подаётся значения для начального отсчёта, если оно отлично от нуля?

Вопросы категории 3. «Применять».

8.4.6. Изобразите схему асинхронного счётчика по модулю 13, на основе счётных триггеров, работающих по началу фронта счётного импульса.

8.4.7. Переделайте схему счётчика из п.8.3. , чтобы он начинал счёт с числа 12 до значения 0.

Ответы на контрольные вопросы.

8.4.1. Термин «модуль счётчика» определяет количество состояний, которое принимает счётчик.

8.4.2. Реверсивный счётчик работает как в суммирующем (увеличивающемся) режиме, так и в вычитающем.

8.4.3. Синхронные счётчики получают тактовых синхросигнал на все разряды в отличии от асинхронного счётчика, в котором синхроимпульс одновременно является и счётным сигналом и подаётся на младший разряд счётчика.

8.4.4. Один из вариантов такого ограничения модуля счётчика можно реализовать с помощью элемента «И-НЕ», на входы которого собран код, равный значению модуля, снимаемый с прямых и инверсных выходов разрядов счётчика. Схема ограничения может быть следующей (рис. 8.14)

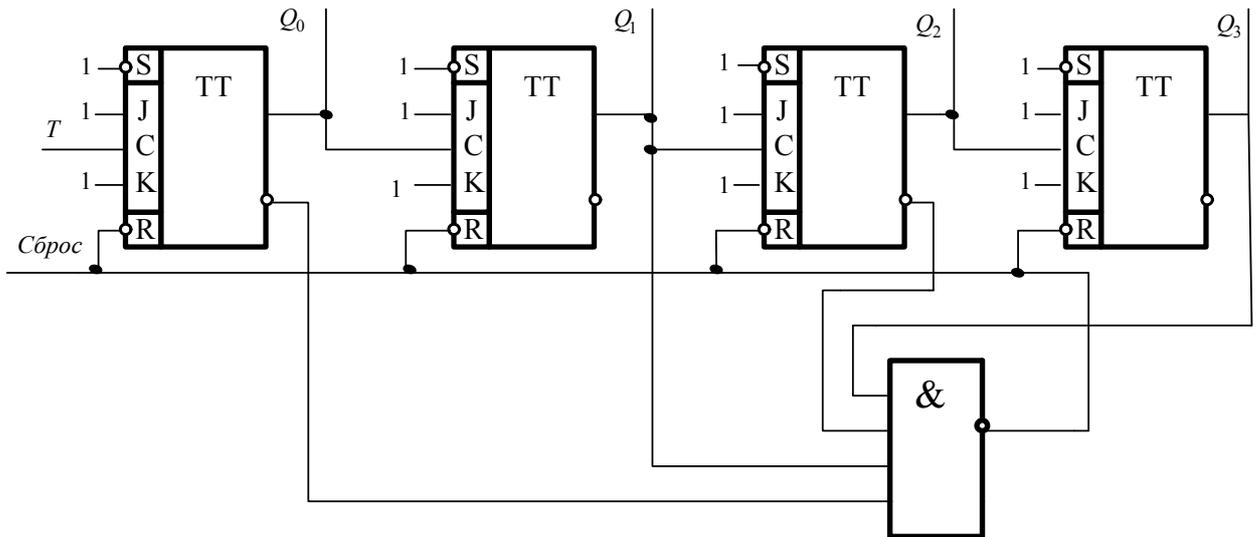


Рис. 8.14.. Схема асинхронного суммирующего счетчика с значением модуля $K_{сч}=10$.

8.4.5. Начальное значение отсчёта в счётчик загружается через асинхронные установочные входы разрядов счётчика.

8.4.6.

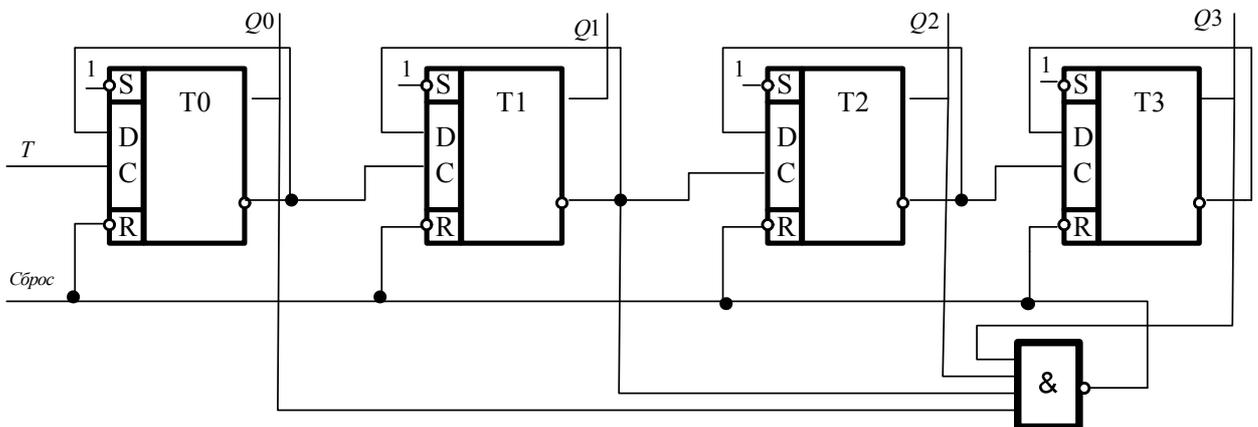


Рис. 8.15.. Схема асинхронного суммирующего счетчика по модулю 13.

8.4.7.

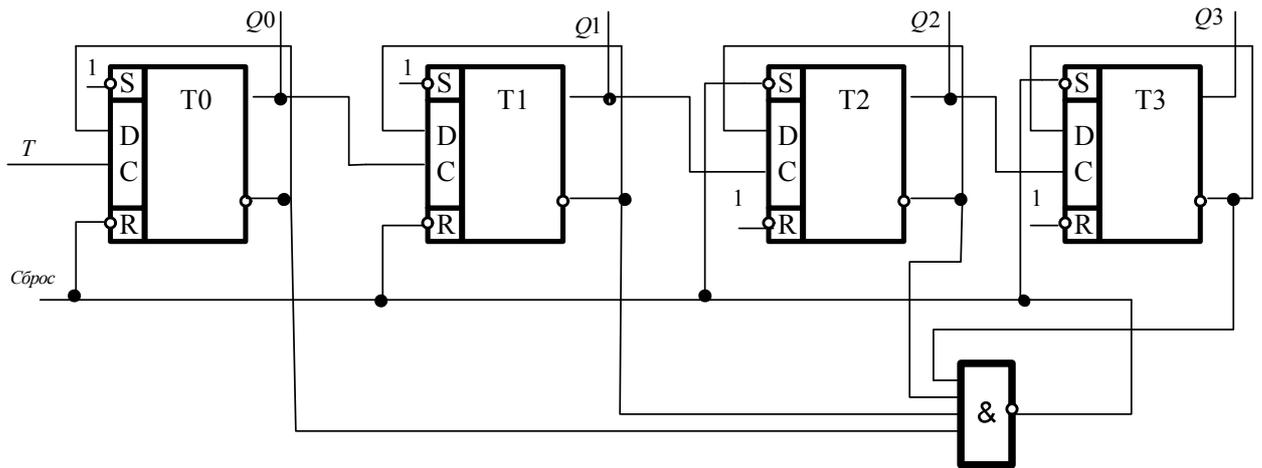


Рис. 8.16.. Схема асинхронного вычитающего счетчика с числа 12 до значения числа 0.

8.5. Задачи для самостоятельного решения.

8.5.1. Изобразите схему асинхронного 8-разрядного суммирующего счётчика на базе JK-триггеров, управляемых прямым (начальным) фронтом.

8.5.2. Как работает схема счётчика, приведённая на рис. 8. 17.

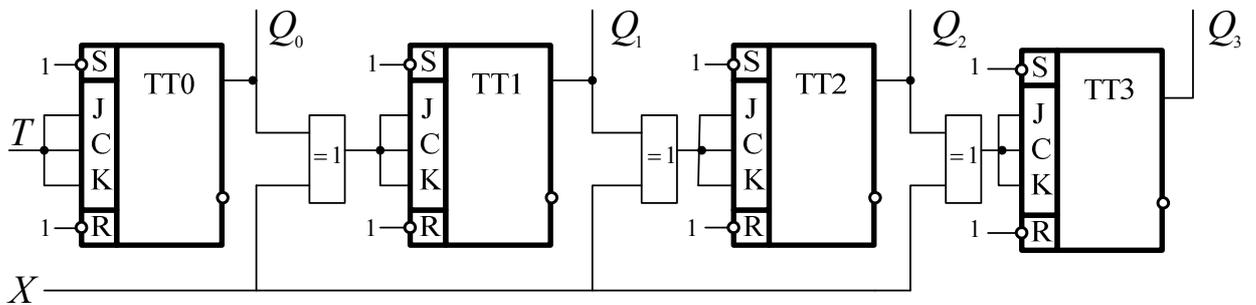


Рис. 8.17. Схема счётчика к заданию 8.14.

Глава 9. Арифметические устройства.

9.1. Назначение и классификация арифметических устройств.

Арифметические устройства предназначены для выполнения самых распространённых операций: сложения, вычитания, умножения и сравнения.

Основные устройства, выполняющие перечисленные операции, это сумматоры, вычитатели, множительные блоки и компараторы. Они входят в состав более сложных функциональных устройств вычислительных устройств, таких как, например, арифметико-логическое устройство (АЛУ).

Арифметическое сложение считается основной элементарной операцией, выполняемой над кодами чисел в цифровых устройствах.

9.1.1. Назначение и классификация сумматоров.

Сумматор - это электронное устройство, выполняющее арифметическое сложение кодов двух чисел.

Сумматоры применяются и для выполнения операции вычитания, но для этого осуществляются дополнительные преобразования кодов чисел.

Сумматоры классифицируют по различным признакам:

В зависимости от системы счисления различают:

- двоичные сумматоры;
- двоично-десятичные;
- десятичные;
- и другие.

По количеству одновременно обрабатываемых разрядов складываемых чисел сумматоры бывают:

- одноразрядные;
- многоразрядные.

По числу входов и выходов одноразрядных двоичных сумматоров различают:

- четверть сумматоры (элементы "сумма по модулю 2", то есть "исключающее ИЛИ"), имеют два входа для двух одноразрядных чисел и одним выходом, на котором реализуется их арифметическая сумма;

- полусумматоры, характеризующиеся наличием двух входов, на которые подаются одноимённые разряды двух чисел и двух выходов: на одном реализуется арифметическая сумма в данном разряде, а на другом - перенос в следующий (более старший разряд);

- полные одноразрядные двоичные сумматоры, характеризующиеся наличием трёх входов, на которые подаются одноимённые разряды двух складываемых чисел и перенос из предыдущего (более младшего) разряда и двумя выходами:

на одном реализуется арифметическая сумма в данном разряде,
а на другом - перенос в следующий (более старший разряд).

По способу представления и обработки складываемых чисел многоразрядные сумматоры подразделяются на:

- последовательные, в которых обработка чисел ведётся поочерёдно, разряд за разрядом на одном и том же оборудовании;

- параллельные, в которых слагаемые складываются одновременно по всем разрядам, и для каждого разряда имеется своё оборудование.

По способу выполнения операции сложения и возможности сохранения результата выделяются два типа сумматоров:

- комбинационный сумматор, выполняющий микрооперацию " $S=A+B$ ", в котором результат выдаётся по мере его образования;

- накапливающий сумматор, на вход которого операнды подаются последовательно с некоторой задержкой.

По способу организации межразрядных переносов делят на:

сумматоры с последовательным переносом,

сумматоры с параллельным переносом,

сумматоры с условным переносом,

сумматоры со сквозным переносом

9.1.2. Основные параметры сумматоров.

Разрядность.

Статические параметры: $U_{\text{вх.}}$, $U_{\text{вых.}}$.

$I_{\text{вх.}}$, то есть обычные параметры интегральных схем.

Динамические параметры:

- задержка распространения от подачи входного переноса до установления всех выходов суммы при постоянном уровне на всех входах слагаемых,
- задержка распространения от одновременной подачи всех слагаемых до установления всех выходов суммы при постоянном уровне на входе переноса,
- задержка распространения от подачи входного переноса до установления выходного переноса при постоянном уровне на входах слагаемых,
- задержка распространения от подачи всех слагаемых до установления выходного переноса при постоянном уровне на входах слагаемых.

9.1.3. Основные типы сумматоров.

Четвертьсумматор.

Четвертьсумматор является простейшим двоичным суммирующим элементом. Наиболее известны названия для схем, реализующих четвертьсумматор: логический элемент «исключающее ИЛИ» и логический элемент «сумма по модулю 2». Эти два элемента имеют идентичную таблицу истинности (табл. 9.1.).

Четверть сумматор имеет два входа двоичных аргументов a и b и один выход суммы S . На рис. 9.1. представлены два логических элемента, реализующих четверть сумматор.

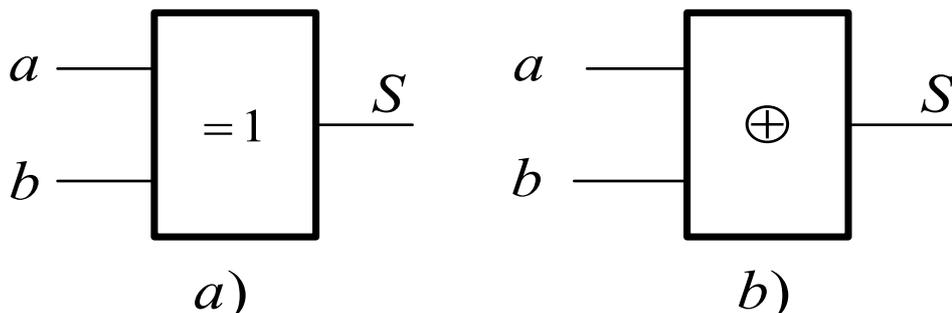


Рис. 9.1. Варианты реализации четвертьсумматора на логических элементах.

Таблица 9.1. Таблица истинности четвертьсумматора.

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

По таблице истинности можно составить булеву функцию для суммы S .

$$S = \bar{a}b + a\bar{b} = \overline{\overline{\bar{a}b} \wedge \overline{a\bar{b}}} \quad (9.1)$$

По выражению 9.1 синтезируется схема четвертьсумматора на элементах «И-НЕ» рис. 9.2.

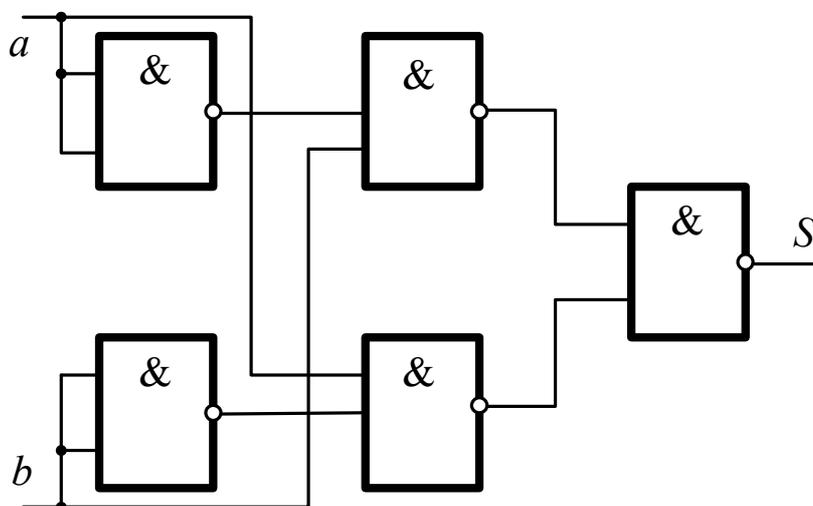


Рис.9.2. схема четвертьсумматора на элементах «И-НЕ».

Полусумматор.

Полусумматор - это комбинационная схема, которая вырабатывает сигналы суммы S и переноса C_o при сложении двух двоичных чисел a и b . Полусумматор функционирует согласно таблице истинности табл. 9.2.

Таблица 9.2. Таблица истинности полусумматора.

a	b	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Из таблицы 9.2 следует, что функция суммы реализуется логическим элементом «Исключающее ИЛИ», а функция переноса реализуется логическим элементом «И». Типовая схема полусумматора на элементах «Исключающее ИЛИ» и «И» представлена на рис. 9.3.

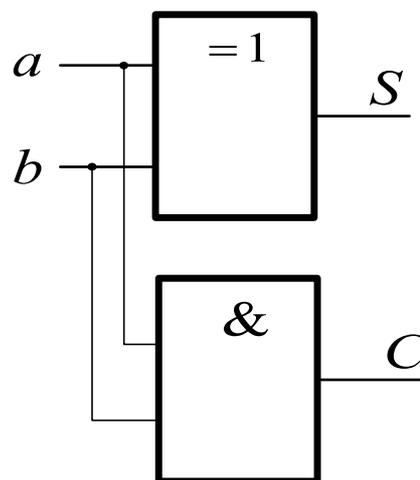


Рис. 9.3. Схема полусумматора на элементах «Исключающее ИЛИ» и «И».

Полный сумматор или просто сумматор.

В отличие от полусумматора должен воспринимать 3 входных сигнала: 2 слагаемых и сигнал переноса с предыдущего разряда.

Сумматором называется операционный узел ЭВМ, выполняющий операцию арифметического сложения двух чисел.

Работа полного сумматора соответствует следующей таблице истинности табл. 9.3.

Таблица 9.3. Таблица истинности полного сумматора.

a	b	c_i	S	c_o
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

Из таблицы истинности полного сумматора видно, что, если отсутствует перенос из младшего разряда, то перенос в старший разряд может быть только в одном случае, когда оба числа равны единице.

Если же имеется перенос из младшего разряда, то перенос в старший разряд будет всегда, кроме одного случая, когда оба слагаемых равны нулю.

Схема сумматора может быть реализована на двух полусумматорах.

Соответствие работы этой схемы и таблицы функционирования можно проверить перебором всех возможных вариантов.

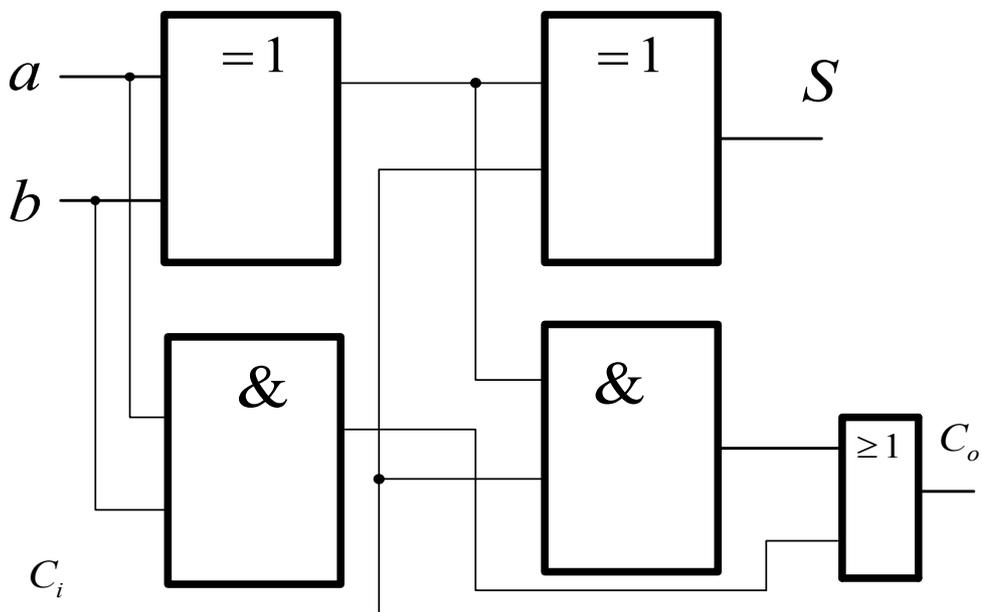


Рис. 9.4. Схема полного сумматора из двух полусумматоров.

Для сложения двух многоразрядных двоичных чисел на каждый разряд необходим один полный сумматор. Только в младшем разряде можно

обойтись полусумматором. Согласно ГОСТ 2.743-91 условно-графическое изображение сумматора выглядит следующим образом (рис. 9.5)

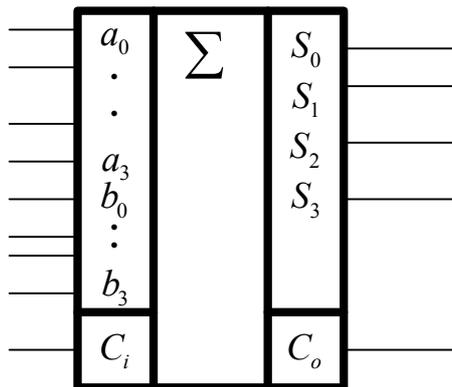


Рис. 9.5. Условно-графическое изображение комбинационного сумматора на 4 двоичных разряда.

Реализовать сумматор на 4 разряда можно из одноразрядных сумматоров (рис. 9.4.) соединив их связями для реализации последовательного переноса (рис. 9.6.).

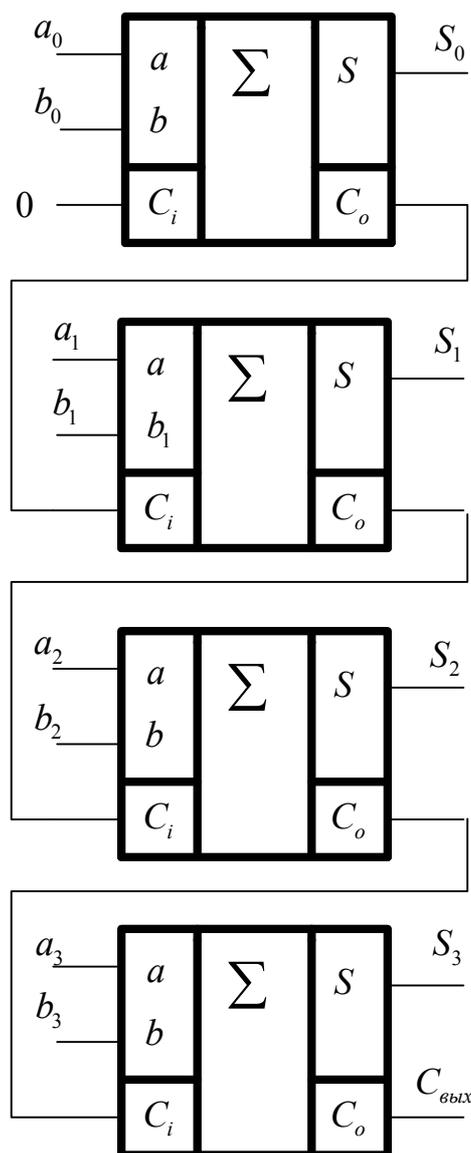


Рис. 9.6. Четырёхразрядный комбинационный сумматор с последовательным переносом.

Одноразрядный комбинационный сумматор можно реализовать и в другом базисе, например на логических элементах основного функционально-полного набора («И-ИЛИ-НЕ»). Для этого необходимо сделать синтез схемы для функции суммы S и функции переноса C_o .

Для этого применим формальный приём, в котором выходной перенос C_o будем считать аргументом для вычисления функции суммы S . Тогда таблица 9.3 примет иной вид (табл. 9.3.).

Таблица 9.3.

a	b	c_i	c_o	S
0	0	0	0	0
0	0	0	1	X
0	0	1	0	1
0	0	1	1	1
0	1	0	0	X
0	1	0	1	X
0	1	1	0	X
0	1	1	1	0
1	0	0	0	1
1	0	0	1	X
1	0	1	0	X
1	0	1	1	0
1	1	0	0	X
1	1	0	1	0
1	1	1	0	X
1	1	1	1	1

Результатом синтеза являются характеристические выражения для суммы и выходного переноса.

$$S = C_i \overline{C_o} + abC_i + aC_o + b\overline{C_o} = abC_i + C_i + 1(a + b + C_i)$$

$$C_o = \overline{a}bC_i + a\overline{b}C_i + ab\overline{C_i} + abC_i = ab + C_i(a + b)$$

Согласно выведенным выражениям схема одноразрядного полного сумматора на элементах «И-ИЛИ-НЕ» выглядит следующим образом (рис. 9.7.)

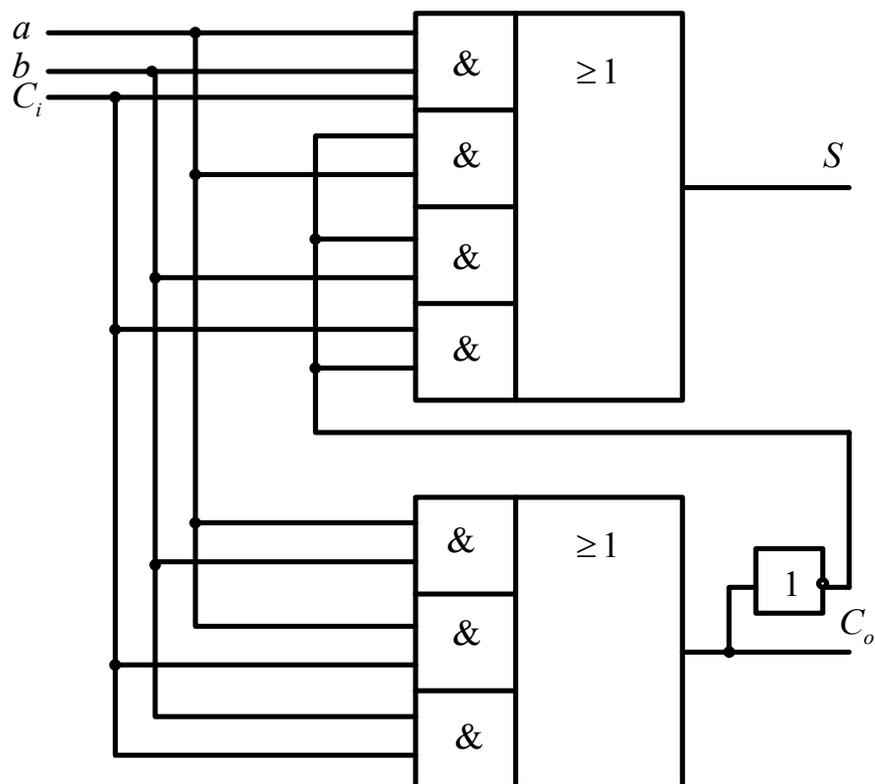


Рис. 9.7. Схема одноразрядного полного сумматора на элементах «И-ИЛИ-НЕ».

9.2. Классификация и основные типы вычитателей.

Вычитатель – устройство комбинационного типа, предназначенное для выполнения операции вычитания двоичных чисел.

9.2.1. Полувычитатель.

Полувычитатель - это комбинационная схема, которая вырабатывает сигналы разности D и заем Z_o при вычитании двух двоичных чисел a и b . Полувычитатель функционирует согласно таблице истинности табл. 9.4.

Таблица 9.4. Таблица истинности полувычитателя.

a	b	D	Z_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Из таблицы 9.4. следует, что функция разности реализуется логическим элементом «Исключающее ИЛИ», а функция заёма реализуется логическим элементом «Запрет». Схема полувычитателя представлена на рис. 9.8

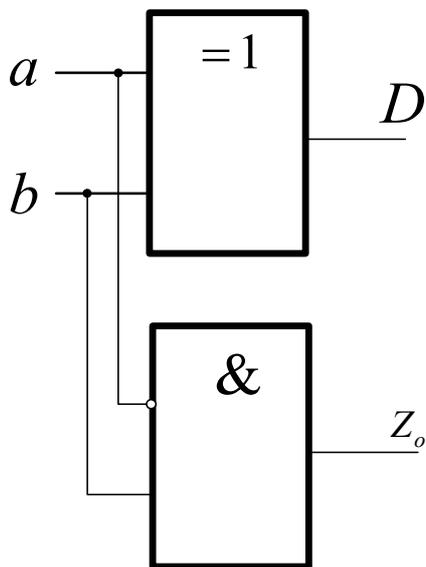


Рис. 9.8. Схема полувычитателя.

9.2.2. Полный вычитатель.

Полный вычитатель имеет помимо входов аргументов a и b третий вход входного заёма из младшего разряда Z_i . Типовая схема полного вычитателя на элементах «Исключающее ИЛИ» и «Запрет» представлена на рис. 9.9.

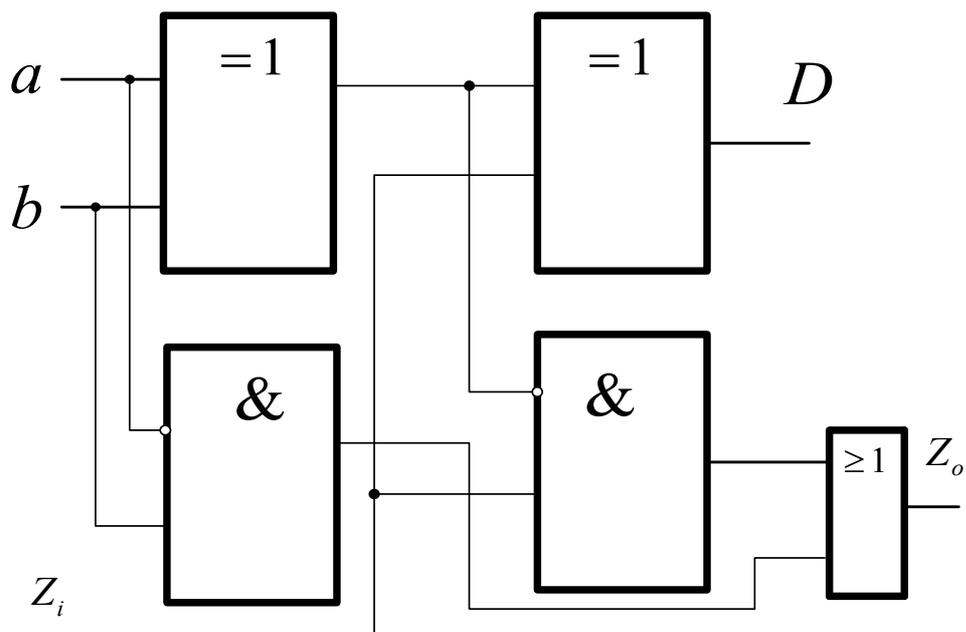


Рис. 9.9. Схема полного вычитателя на логических элементах «Исключающее ИЛИ», «Запрет» и «ИЛИ».

Согласно ГОСТ 2.743-91 условно-графическое изображение вычитателя выглядит следующим образом (рис. 9.10)

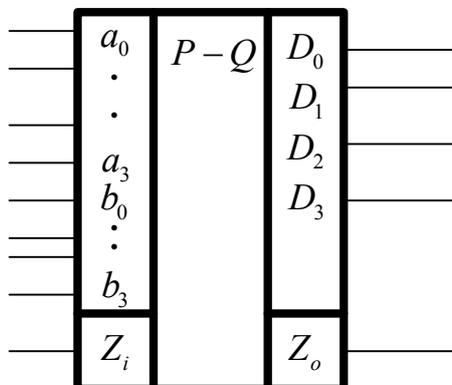


Рис. 9.10. Условно-графическое изображение вычитателя.

Аналогично сумматору параллельного действия с последовательным переносом можно построить схему параллельного многоразрядного вычитателя с последовательным заёмом рис. 9.11.

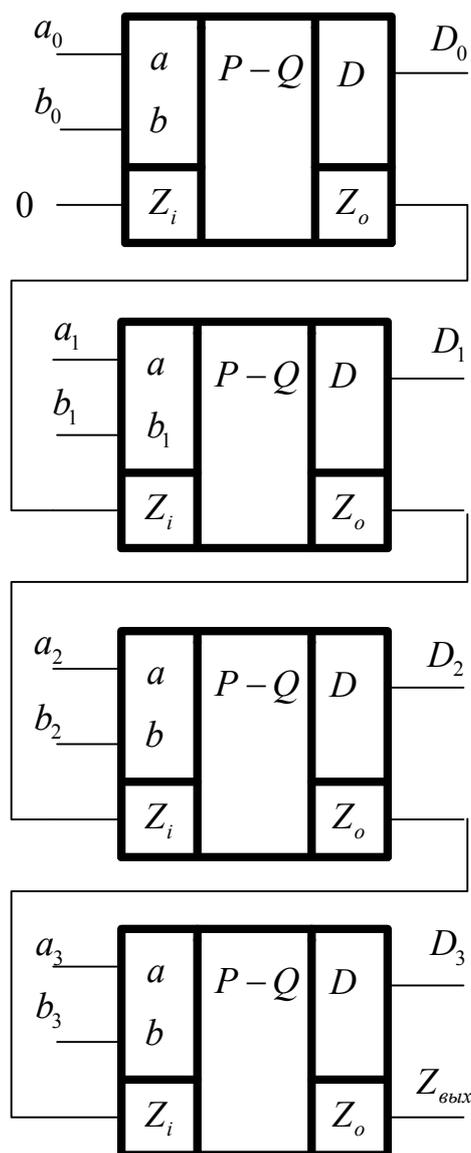


Рис. 9.11. Схема параллельного многоразрядного вычитателя с последовательным заёмом.

9.3. Сумматоры накапливающего типа.

В архитектурах вычислительных устройств встречаются сумматоры, не относящиеся к сумматорам комбинационного типа. Одними из них являются накапливающие сумматоры. Под термином «накапливающие» встречаются сумматоры двух разновидностей:

- сумматоры, построенные на основе счётного триггера и логических элементов,
- сумматор со структурой «комбинационный многоразрядный сумматор плюс регистр хранения». Сумматор первого типа является медленным вследствие двухтактности операции сложения и сложностью управления и

обнуления. Во второй разновидности сумматора можно реализовать схему с накоплением результата « $S := S \text{ плюс } A$ ».

Первая разновидность накапливающего сумматора на базе счётного Т-триггера представлена на рис. 9.12.

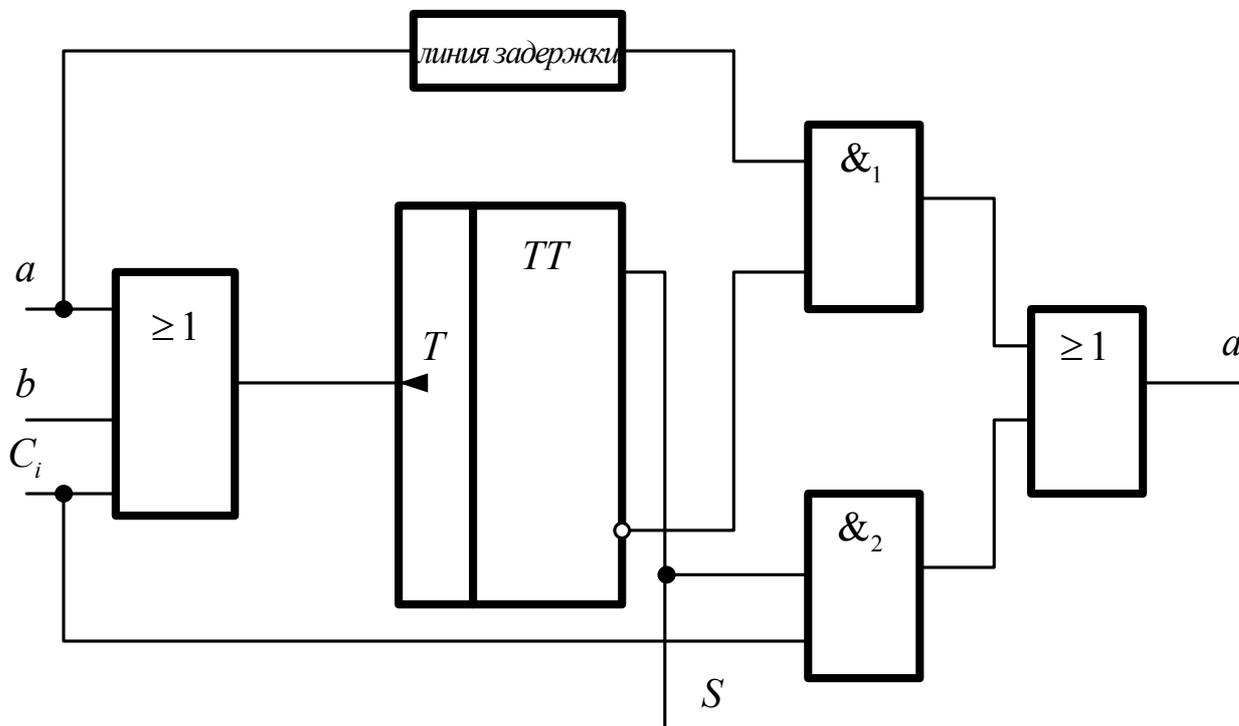


Рис. 9.12. Схема накапливающего сумматора на базе счётного Т-триггера.

Особенностью является необходимость последовательной подачи во времени аргументов и входного переноса. Это требование вытекает из конечного времени переключения триггера. Последовательность подачи входных сигналов иллюстрирует временная диаграмма рис. 9.13.

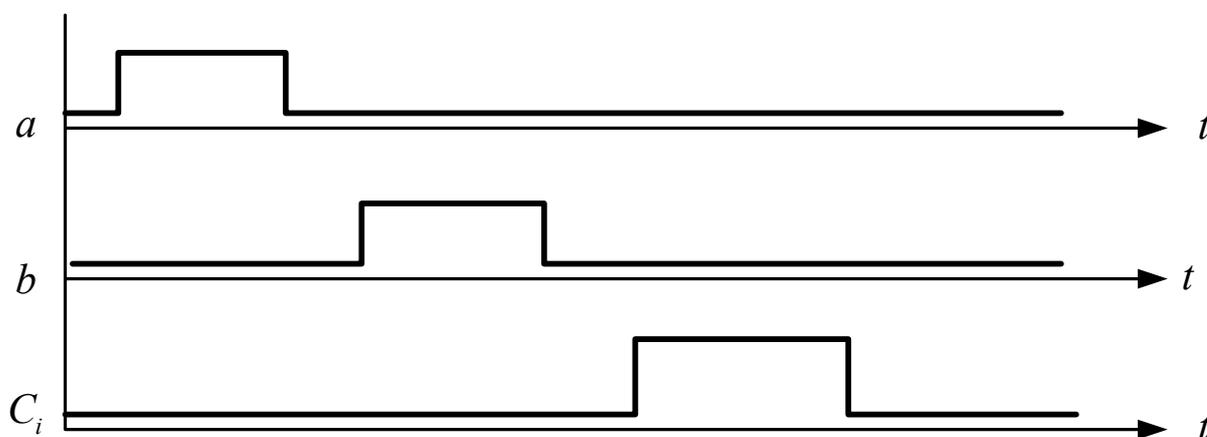


Рис. 9.13 Временная диаграмма подачи входных сигналов на схему накапливающего сумматора.

Таблица 9.5. иллюстрирует получаемые значения суммы S и выходного переноса C_o .

Таблица 9.5. Таблица работы сумматора на основе счётного триггера.

a	b	C_i	S	C_o и по каким сигналам
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1 (aub)
1	0	0	1	0
1	0	1	0	1 (auc_i)
1	1	0	0	1(aub)
1	1	1	1	1($aubuc_i$)

В основе суммирования лежит свойство счётного триггера, которые меняет своё состояние на противоположное при каждой поступающей логической единицей на его счётный вход. Двоичный одноразрядный код аргументов и входящего переноса поступают на счётный вход триггера последовательно согласно временной диаграмме рис. 9.13. Значение суммы снимается с прямого выхода триггера после поступления перечисленных трёх сигналов.

Логические элементы «И» с номерами 1 и 2 формируют значения переноса, возникающее из-за сложения двух единичных аргументов или при сложении одного из единичного аргумента и единичного значения входного переноса. Линия задержки нужна для того, чтобы передать предыдущее состояние триггера в момент поступления второго аргумента для порождения выходного переноса. Логический элемент «И» с номером 2 порождает

входной перенос при наличии входного переноса и одного единичного операнда.

Вторая разновидность накапливающего сумматора рис. 9.14. состоит из многоразрядного комбинационного сумматора, выходы которого соединяются со входами регистра хранения. Регистр хранения коммутируется своими выходами на вход второго аргумента сумматора.

Аргумент A подаётся на вход первого аргумента сумматора. При каждом тактовом импульсе регистр обновляет своё содержимое новой промежуточной суммой $S + A$.

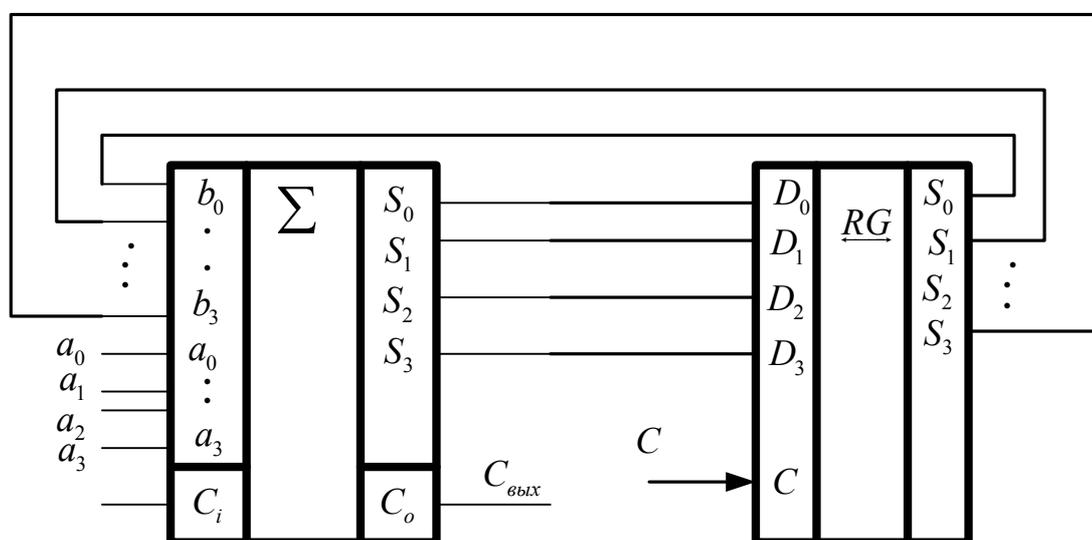


Рис. 9.14. Накапливающий сумматор за счёт регистра хранения.

9.4. Двоично-десятичный сумматор.

Двоично-десятичные сумматоры выполняют действия над десятичными числами, разряды которых заданы двоичными тетрадами. Способ построения двоично-десятичного сумматора основывается на суммировании двоичных тетрад аргументов и коррекцию результата на дополнительном сумматоре, если сумма двух двоичных тетрад дала результат, превышающий число 9. Если при сложении тетрад и входного переноса результат лежит в диапазоне от 10 до 15, необходимо организовать перенос в соседнюю тетраду с одновременным уменьшением полученного числа на 10. Вычитание числа 10 можно заменить сложением с дополнительным кодом числа 10, имеющим двоичное представление 0110. Второй корректирующий

сумматор в этом случае должен прибавить корректирующую поправку 0110 и обеспечить порождение переноса в старшую тетраду.

Схема двоично-десятичного сумматора представлена на рис. 9.15.

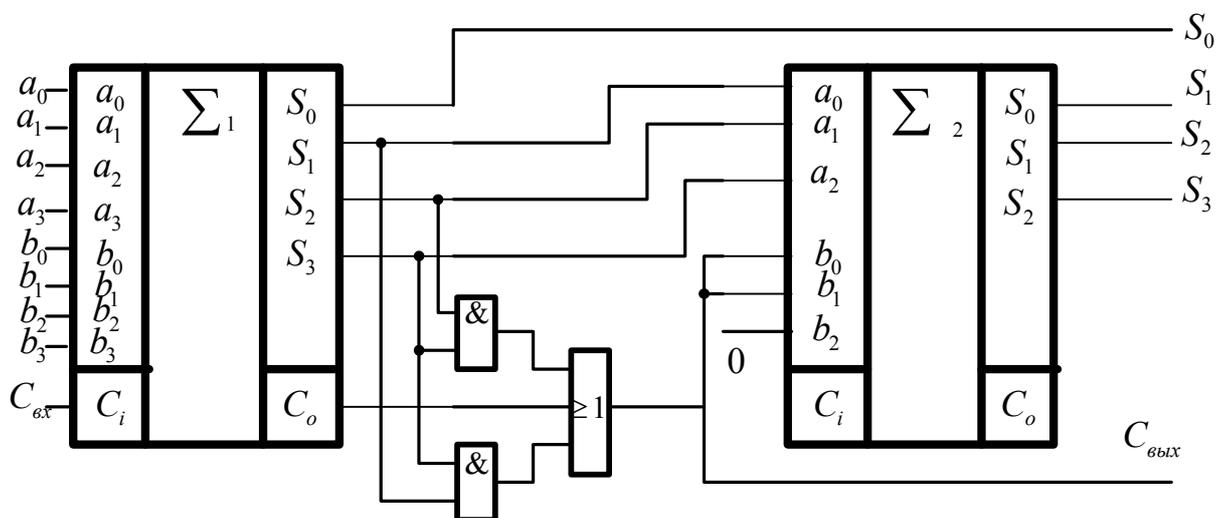


Рис. 9.15. Схема двоично-десятичного сумматора.

Сумматор с индексом 1 формирует значения аргументов, которые могут лежать в диапазоне от 0 до 31. Для обнаружения сумм, превышающих число 9 введены два логических элемента «И», которые в этих комбинациях выдают логическую единицу, поступающую на логический элемент «ИЛИ». Выход элемента «ИЛИ» формирует выходной перенос в следующую тетраду. Этот же сигнал логической единицы порождает добавление кода 0110 (по входам b_0 и b_1). При результате сложения на первом сумматоре больше числа 15 выдаётся выходной перенос, который поступает на следующую тетраду.

9.5 Универсальный многоразрядный сумматор-вычитатель.

Особенность данного сумматора состоит в том, что в нём предусмотрено переключение между операцией суммирование и вычитание. Ограничение в его применения заключается в получении не прямого, а дополнительного кода при выполнении операции $c = a - b$, если $a \leq b$. Схема построения

универсального многоразрядного сумматора-вычитателя приведена на рис. 9.16.

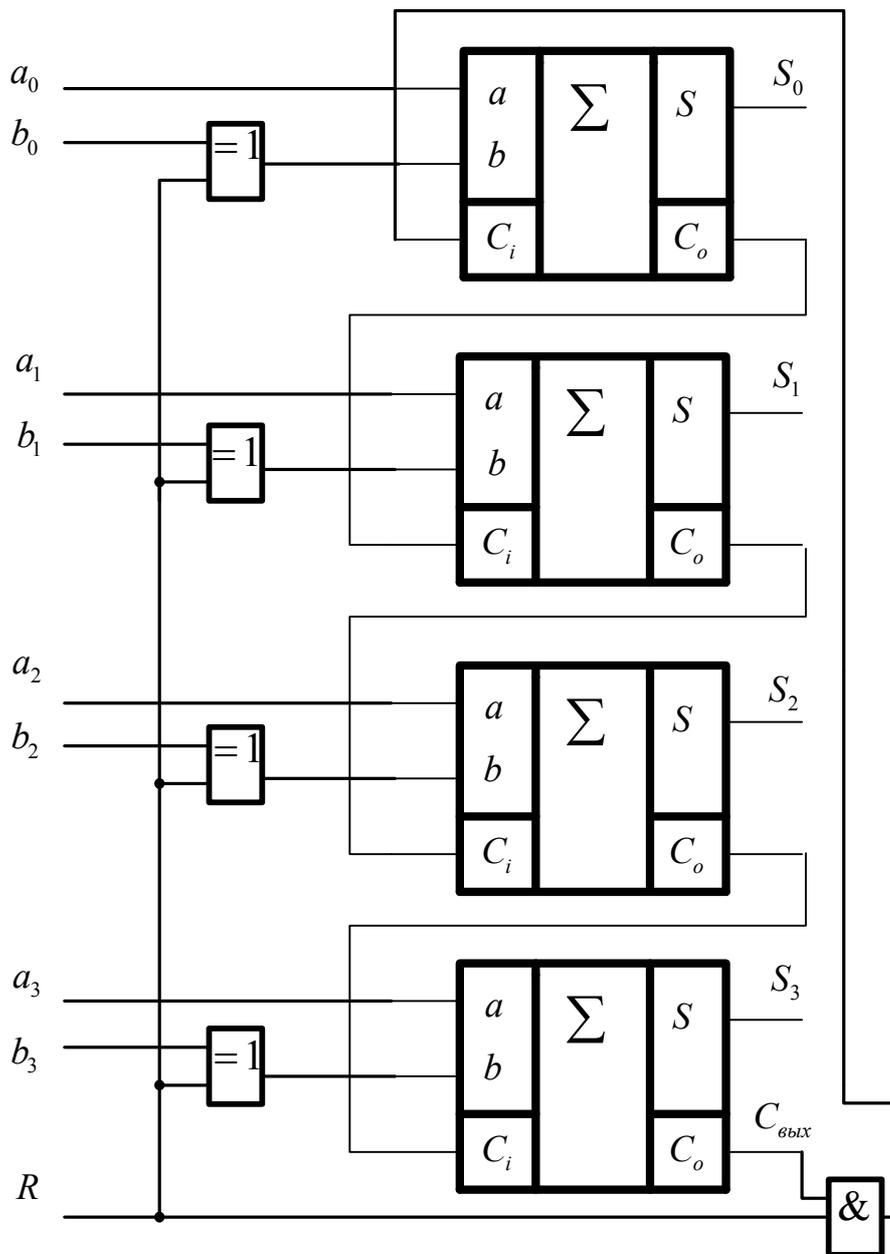


Рис. 9.16. Схема универсального сумматора-вычитателя.

Управления переключением режимов реализовано двоичным сигналом R .

При $R = 0$ выполняется суммирование. Аргумент b проходит логические элементы «Исключающее ИЛИ» без изменений. При $R = 1$, аргумент b попадает на входы сумматоров в инверсном коде. При этом возникающий перенос в выхода старшего разряда подаётся через открытый элемент «И» на вход переноса младшего разряда.

9.6. Сумматоры дополнительного кода и сумматоры обратного кода.

Применение модифицированного дополнительного кода и модифицированного обратного кода.

В большинстве вычислительных устройств используется представление целых чисел со знаком. Старший разряд используется для кодирования знака: 0 обозначает положительное число, 1- отрицательное число. Использование представления чисел вместе со знаками в двоичном коде позволяет осуществлять арифметические операции суммирования и вычитания, получая результат с готовым знаком числа результата, применяя тот же классический комбинационный многоразрядный сумматор. Применение дополнительного и обратного кода позволяет представлять отрицательные числа в этих кодах и, подавая их на сумматор фактически выполняя операцию суммирования, получать разность.

Перед рассмотрением примеров использования дополнительного и обратного кода необходимо остановится на следующих аксиомах и правилах:

1. Сумма двух чисел в дополнительном (или обратном) коде есть дополнительный (или обратный) код.
2. Дополнительный (обратный) код положительного числа совпадает с его прямым кодом.
3. Дополнительный код отрицательного числа со знаком формируется путём инвертирования разрядов числовой части целого числа, исключая знак и арифметическое сложение инвертированного значения с единицей.
4. Обратный код отрицательного числа со знаком формируется путём инвертирования разрядов числа, кроме разряда знака.

Сумматор дополнительного кода выглядит как многоразрядный комбинационный сумматор рис. 9.6. Самый старший одноразрядный сумматор в таком сумматоре складывает двоичные коды знаков. Выходной перенос из знакового разряда отбрасывается. Ниже приведены

несколько примеров сложения чисел со знаками в различных сочетаниях у аргументов.

Примеры:

9.6.1. $C=A+B=(+9)+(-3)$

В прямом коде: A 0 1001 ; B 1 0011 .

В дополнительном коде A 0 1001 ; B 1 1101 .

Результат сложения:

$$\begin{array}{r} 0\ 1001 \\ + \\ 1\ 1101 \\ \hline \end{array}$$

C= $\overline{0\ 0110}$ наличие нуля в знаковом разряде

говорит о положительном результате. То есть дополнительный код результата равен прямому коду и это есть окончательный результат.

9.6.2 $C=A+B=(+4)+(-9)$

В прямом коде: A 0 0100 ; B 1 1001

В дополнительном коде: A 0 0100 ; B 0111

Результат сложения:

$$\begin{array}{r} 0\ 0100 \\ + \\ 1\ 0111 \\ \hline \end{array}$$

C=1 1011 (отрицательный результат в

дополнительном коде). После преобразования в прямой код C= 1 0101.

Операция сложения со знаками возможна и в обратных кодах.

Особенностью сумматора обратного кода является наличие связи по переносу между сумматором знаков и сумматором младших разрядов. Схема сумматора обратного кода представлена на рис. 9.17.

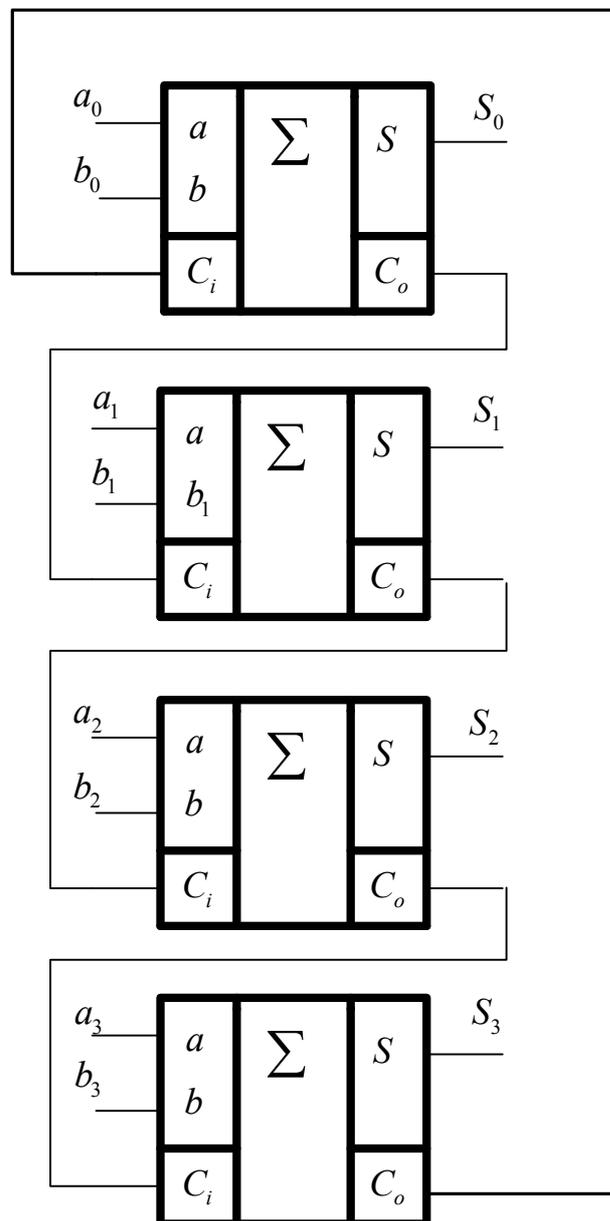


Рис. 9.17. Схема сумматора обратного кода.

Применение модифицированных кодов.

При выполнении операций в дополнительном и обратном коде включая знаки чисел происходит нарушение правильного значения знака результата из-за переполнения разрядной сетки числовой части и поступления вызванного переполнением переноса в сумматор знаков. Аппаратно определить данную ситуацию не представляется возможным. Для решения данной проблемы было предложено использовать модифицированный дополнительный и модифицированный обратный код. Отличие состоит в дублировании битов, обозначающих код знаков слагаемых. Благодаря такому дублированию, возникающий перенос искажает только один из

знаков результата в модифицированном коде. Признаком переполнения является несовпадение знаков результата выполнения операции сложения. Это несовпадение легко обнаруживается при помощи логического элемента «Исключающее ИЛИ».

9.7. Методы ускорения распространения переносов в сумматорах.

Недостатком схемы многоразрядного сумматора с последовательным переносом (рис. 9.6) является пропорциональная зависимость роста задержки распространения переноса в зависимости от разрядности. Задержку распространения переноса можно определить из выражения:

$T_c = t_{ic} * (n - 1)$, где t_{ic} - задержка появления переноса на одноразрядном сумматоре, n - количество разрядов сумматора.

Наибольшее распространение в схемах сумматоров получила технология организация параллельного переноса. Сумматоры с параллельным переносом не имеют последовательного распространения переноса вдоль разрядной сетки. Во всех разрядах сигналы переносов вырабатываются одновременно, параллельно во времени. Сигналы переноса для данного разряда формируются специальными схемами, на входы которых поступают все переменные, необходимые для выработки переносов, т.е. те от которых зависит его наличие или отсутствие. Это внешний входной перенос на входе младшего разряда (если он есть) и значения всех разрядов слагаемых, младших относительно данных. На рис. 9.18 приведена функциональная архитектура сумматора с параллельным переносом.

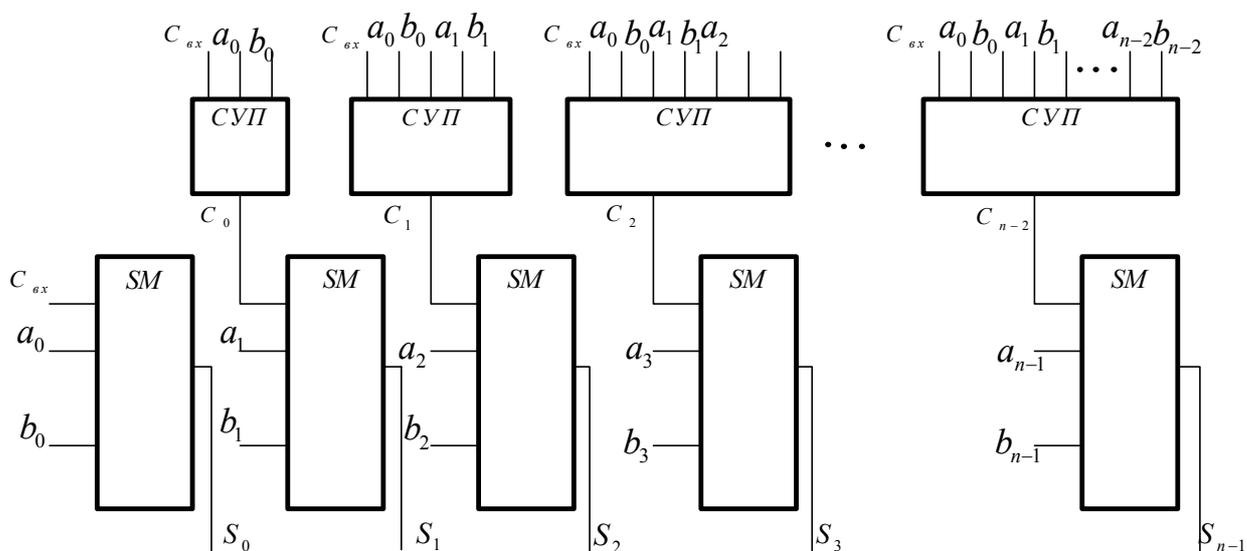


Рис. 9.18 Сумматор с параллельным переносом.

9.8. Инкременторы и декременторы.

Инкрементор - это электронное устройство, выполняющее арифметическое сложение многоразрядного кода с единицей младшего разряда.

Декрементор - это электронное устройство, выполняющее арифметическое вычитание из многоразрядного кода единицы младшего разряда.

Эти устройства строятся на основе полусумматоров и полувычитателей в силу того, что входы второго аргумента, начиная со второго разряда задействованы под приём переноса (или заёма) из предыдущего младшего разряда. На вход второго аргумента младшего разряда подаётся единица.

На рис. 9.18. представлена схема инкрементора, а на рис. 9.19. схема декрементора.

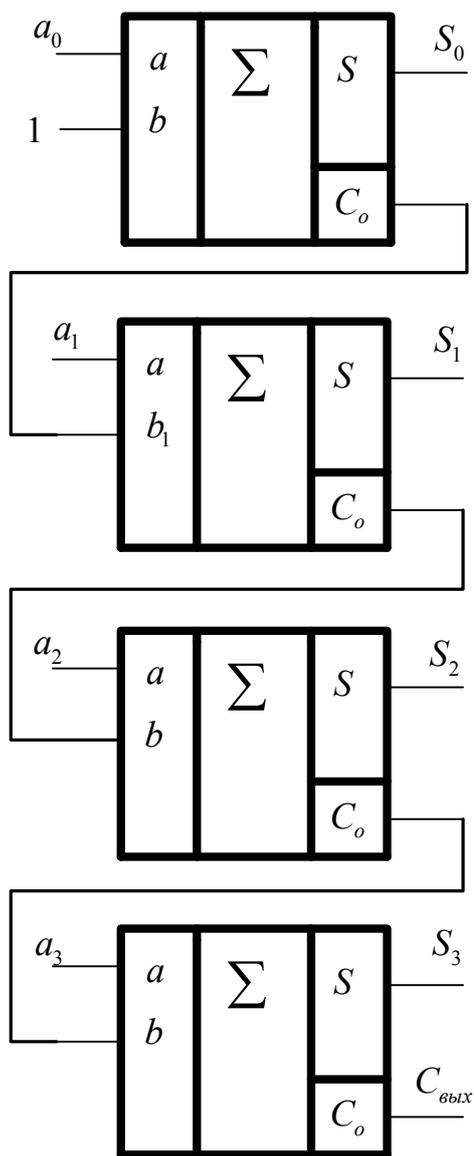


Рис. 9.18. Схема инкремента.

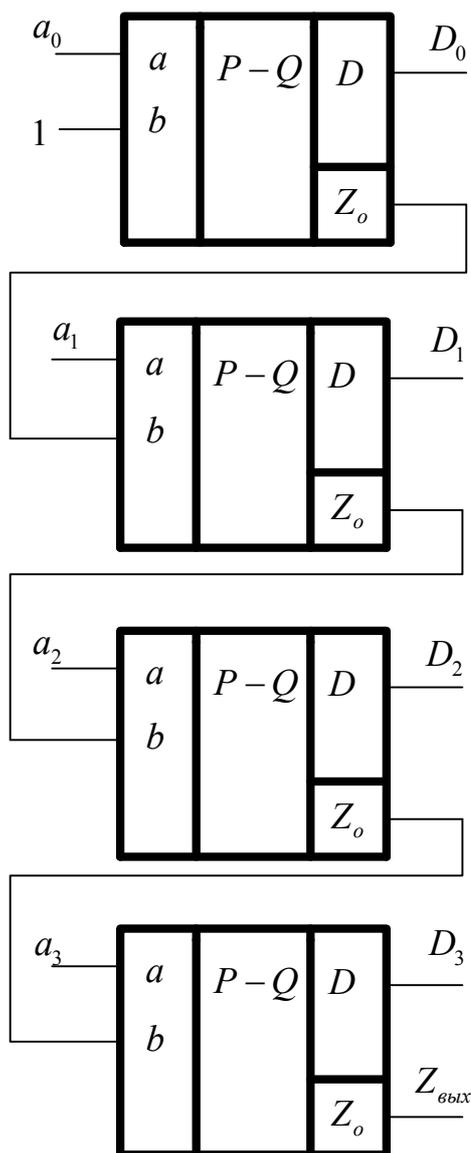


Рис. 9.19. Схема декрементора.

9.9. Цифровые компараторы.

Компаратор – это электронное устройство, выполняющее арифметическое сравнение двух многоразрядных кодов A и B и сообщает является ли $A > B$, $A = B$ или $A < B$. Компараторы относятся к арифметическим устройствам и, как правило, входят в состав арифметико-логических устройств.

В вычислительной технике компараторы применяются:

- для выявления нужного числа в потоке информации,
- для отметок времени в часовых приборах,

- для выполнения условного перехода в вычислительных устройствах.

В устройствах обработки информации и автоматики компараторы применяются:

- для сигнализации о выходе величины за пределы допуска,

- в приводах следящих систем для определения направления воздействия, ликвидирующего рассогласование,

- при построении счётчиков и сумматоров по произвольному основанию.

9.8.1 Синтез одноразрядного компаратора. Для синтеза комбинационной схемы одноразрядного компаратора воспользуемся таблицей 9.6

Таблица 9.6. Таблица истинности одноразрядного компаратора.

ВХОДЫ		ВЫХОДЫ		
a	b	$F(a > b)$	$F(a = b)$	$F(a < b)$
1	1	0	0	0
1	0	1	0	0
0	1	0	0	1
0	0	0	1	0

На соответствующем выходе компаратора появляется логическая 1, когда сравниваемые входные двухразрядные числа находятся в нужном соотношении. По таблице истинности можно записать соотношения:

$$F(a > b) = a * \bar{b}, F(a = b) = a * b + \bar{a} * \bar{b}, F(a < b) = \bar{a} * b$$

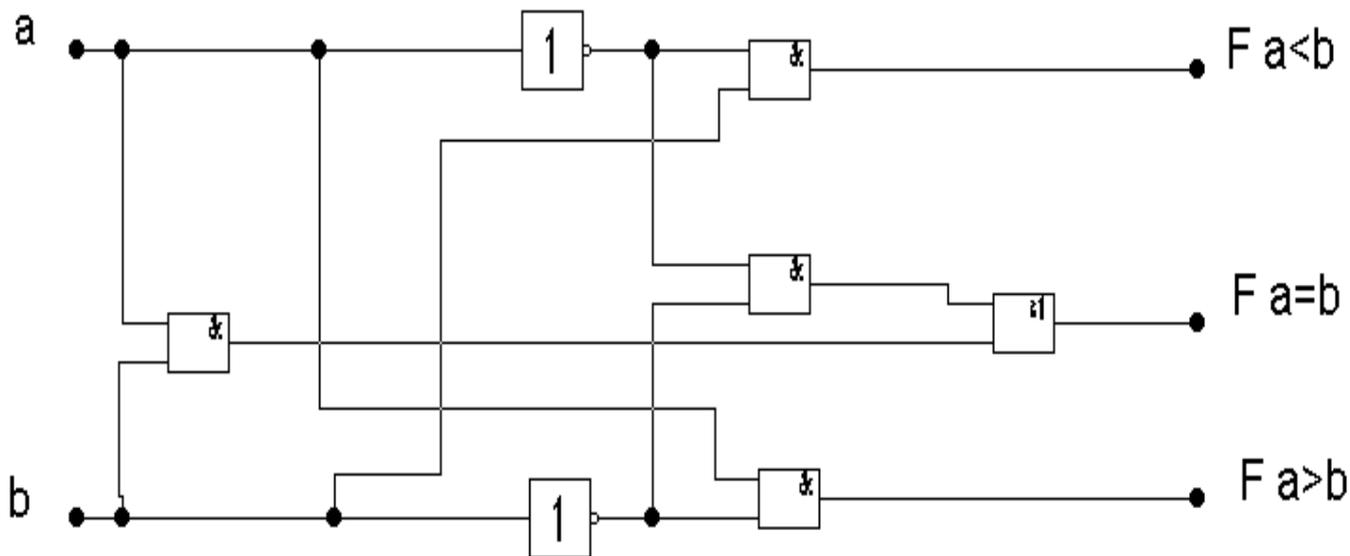


Рис. 9.20 Схема одноразрядного компаратора.

Можно реализовать многоразрядные компараторы, выдающие выходной сигнал одной из функций сравнения или всех функций сравнения.

Трёхразрядный компаратор, выполняющий сравнение двух трёхразрядных кодов на равенство представлен на рис. 9.21.

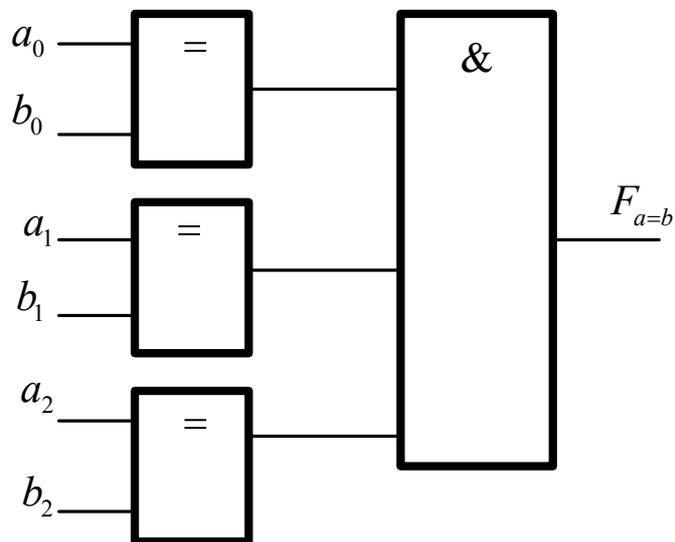


Рис. 9.21. Трёхразрядный компаратор, выполняющий сравнение двух трёхразрядных кодов на равенство.

При сравнении разряда на выходе логического элемента «равнозначность» выдаётся логическая единица. Выходы всех элементов

«равнозначность» сведены на входы логического элемента «И». На выходе логического элемента «И» логическая единица выдаётся при равенстве всех разрядов.

Синтез многоразрядного компаратора основывается на анализе влияния анализируемого разряда на выдаваемого функцию сравнения. Рассмотрим пример синтеза компаратора на три разряда. Часть схемы компаратора, выдающая сигнал равенства двух трёхразрядных чисел соответствует схеме рис. 9.21. Если осуществить запись выражений для одного из неравенств, то для определения второго сигнала неравенства можно принять, что это будет при нулевом значении противоположного неравенства и равенства. Если выходы этих функций подать на вход логического элемента «ИЛИ-НЕ» получим единичное значение искомой функции неравенства.

Разобьем синтез схемы для получения, например, функции $F_{a>b}$ на три этапа анализа влияния каждого из трёх разрядов:

- этап 1. Сравниваются наиболее значимые биты, т.е. a_2 с b_2 . Если $a_2 > b_2$, то и $A > B$. Следовательно, схема для этого случая синтезируется на основе выражения $F2_{a>b} = a_2 \bar{b}_2$;

- этап 2. Если старшие разряды равны $a_2 = b_2$, то $F_{a>b}$ выполнится при $a_1 > b_1$. Следовательно, схема для этого случая синтезируется на основе выражения $F1_{a>b} = (a_2 b_2 + \bar{a}_2 \bar{b}_2) * a_1 \bar{b}_1$;

- этап 3. Если оба разряда $a_2 = b_2$ и $a_1 = b_1$, то $F_{a>b}$ выполнится при $a_0 > b_0$. Следовательно, схема для этого случая синтезируется на основе выражения $F0_{a>b} = (a_2 b_2 + \bar{a}_2 \bar{b}_2) * (a_1 b_1 + \bar{a}_1 \bar{b}_1) * a_0 \bar{b}_0$.

Все эти три выражения по трём этапам связываются функцией дизъюнкцией:

$$F_{a>b} = F2_{a>b} + F1_{a>b} + F0_{a>b}.$$

Согласно выведенным выражениям схема трёхразрядного компаратора для трёх функций условий сравнения примет вид рис.9.22.:

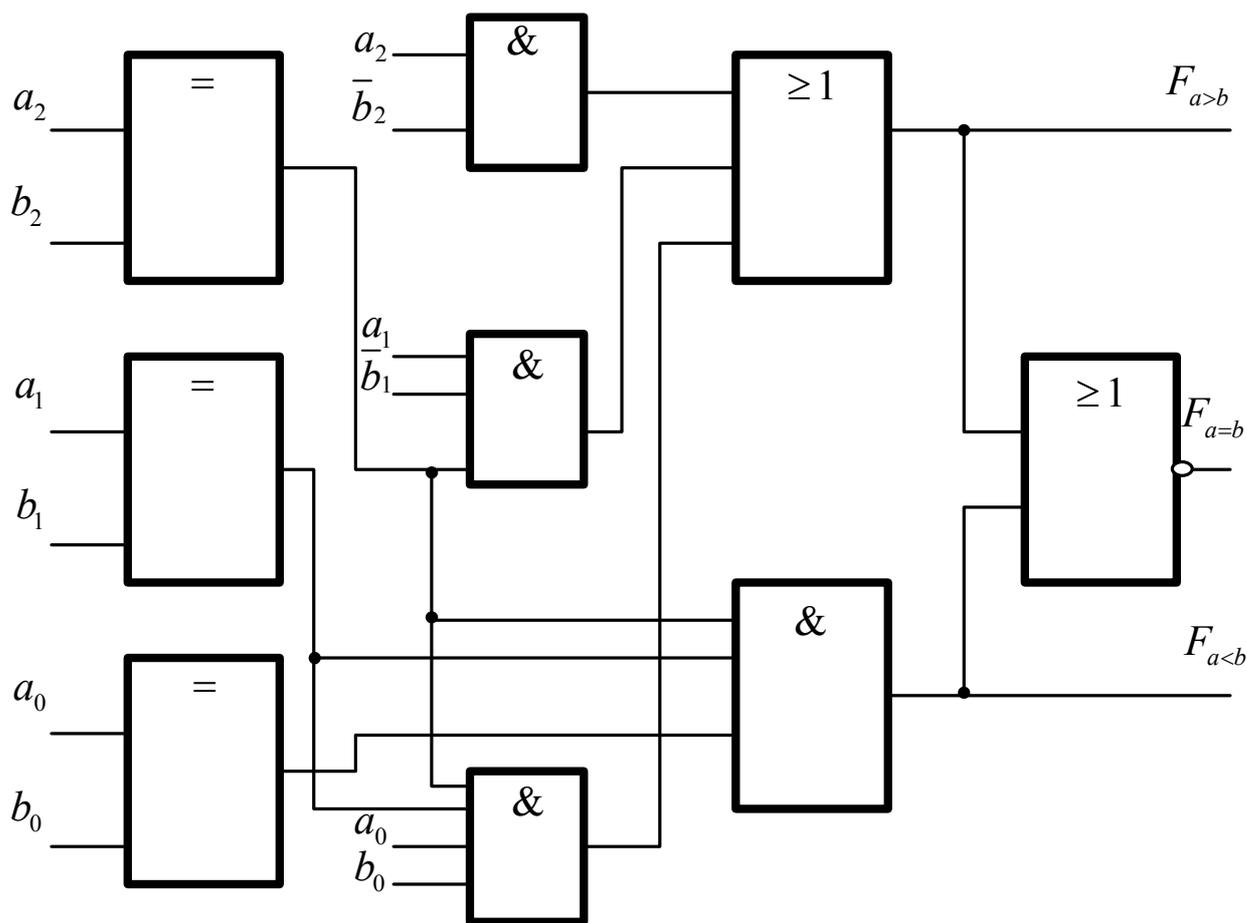


Рис. 9.22. Схема трехразрядного компаратора с тремя выходами результатов сравнения.

Итоги изучения главы 9.

Помнить:

- наименования типовых арифметических устройств, применяемых в вычислительной технике, их условно-графические отображения и основные схемотехнические отличия.

Понимать:

- как осуществляются арифметические операции сложения и вычитания на сумматорах и вычитателях;
- различия в использовании дополнительного и обратного кода в операциях сложения чисел со знаками;
- отличия схем организации переносов в сумматорах;

Применять:

- дополнительный и модифицированный дополнительный код для операций сложения чисел со знаками;
- уметь обнаруживать переполнение при использовании модифицированного кода.

9.10. Типовые примеры проектных решений арифметических устройств.

9.9.1. Спроектировать вычислительное устройство сложения и вычитания одноразрядных десятичных чисел, набираемых на клавиатуре. Работу устройства протестировать с применением пакета Electronics Workbench.

9.11 Контрольные вопросы.

Вопросы категории 1. «Помнить».

- 9.10. Чем отличается понятие «арифметическое сложение» от понятия «логическое сложение»?
- 9.10. Сформулируйте отличия сумматоров дополнительного и обратного кода от обычных сумматоров.

Вопросы категории 2. «Понимать»

- 9.10. Сформулируйте признаки, определяющие четвертьсумматор, полусумматор, и полный сумматор.
- 9.10. Сформулируйте признаки, определяющие четвертьвычитатель, полувычитатель, полный вычитатель.
- 9.10. Поясните, как используется сигнал переноса, возникающий в каком-либо разряде сумматора?
- 9.10. Сформулируйте схмотехнические особенности построения инкрементатора.
- 9.10. Сформулируйте схмотехнические особенности построения декрементатора.
- 9.10. В чём состоит особенность сумматора накапливающего типа. Приведите две возможные реализации.
- 9.10. Поясните функцию переключения универсального сумматора-вычитателя.

9.10. Объясните принцип действия одноразрядного компаратора.

Вопросы категории 3. «Применять»

9.10. Какой логический элемент позволяет реализовать четвертьсумматор?

9.10. На основании каких двух логических элементов можно сформировать схему полусумматора?

9.10. Какой логический элемент позволяет реализовать четвертьвычитатель.

9.10. Какие два логические элементы позволяют сформировать из них полувычитатель?

9.10. Какой приём позволяет использовать сумматор для выполнения операции вычитания?

9.10. Какой приём позволяет обнаружить переполнение при использовании сумматоров?

9.10. Приведите пример использования модифицированных кодов при обнаружении переполнения.

Вопросы категории 4. «Оценивать»

9.10. В каком коде поступает второй аргумент на входы сумматора-вычитателя при выполнении операции вычитания?

9.10. Поясните недостаток дополнительного и обратного кода, возникающий при переполнении на операциях с аргументами со знаками на обычном сумматоре.

9.12. Задачи для самостоятельного решения.

9.11.1. Выполните арифметическую операцию $C = a + b$, где $a = -9$, $b = 3$ в дополнительном модифицированном коде.

9.11.2. Выполните п. 9.11.1. в обратном модифицированном коде.

9.11.2. Компаратор, представленный на рис. 9.21., реализуйте на логических элементах «Исключающее ИЛИ» и «ИЛИ-НЕ».

Глава 10. Запоминающие устройства.

10.1 Назначение и классификация запоминающих устройств.

Запоминающие устройства (ЗУ) предназначены для хранения многоразрядных данных в двоичном коде.

Классификация ЗУ проводится по некоторым разным критериям.

По типу материала ЗУ подразделяются на:

- полупроводниковые,
- магнитные,
- конденсаторные,
- оптические,
- криогенные

По способу организации обращения ЗУ подразделяются на:

- адресные,
- последовательные,
- стековые,
- ассоциативные.

Адресные ЗУ в свою очередь подразделяются на:

- постоянные ЗУ,
- оперативные ЗУ.

В настоящем учебнике рассматриваются только схемы ЗУ полупроводникового типа. Другие типы ЗУ входят в программы курса «Вычислительные средства АСОИУ».

10.2. Постоянные и оперативные запоминающие устройства.

Основной функциональный признак постоянного ЗУ (ПЗУ) - хранение записанной информации в течение продолжительного периода времени при отключённом электропитании. Постоянные ЗУ имеют широкое назначение в вычислительных устройствах, начиная от хранения программ первоначальной загрузки до внешних постоянных ЗУ, используемых различными пользователями.

ПЗУ в свою очередь подразделяются:

- ПЗУ масочного типа (непрограммируемые),
- ПЗУ однократно программируемые.
- Перепрограммируемые ПЗУ.

10.2.1. Масочные ПЗУ. Масочные ПЗУ могут быть реализованы на основе диодов, биполярных транзисторов, МОП-транзисторов. Масочные ПЗУ используются при изготовлении большой серии, и их прошивка осуществляется на этапе изготовления.

На рис. 10.1 представлена матрица диодных элементов масочного ПЗУ. На линиях считывания в пересечениях с адресными линиями, там где «защита» логическая единица диод соединяет адресную линию с разрядной линией считывания. На пересечениях где, защит ноль, диод не подключён.

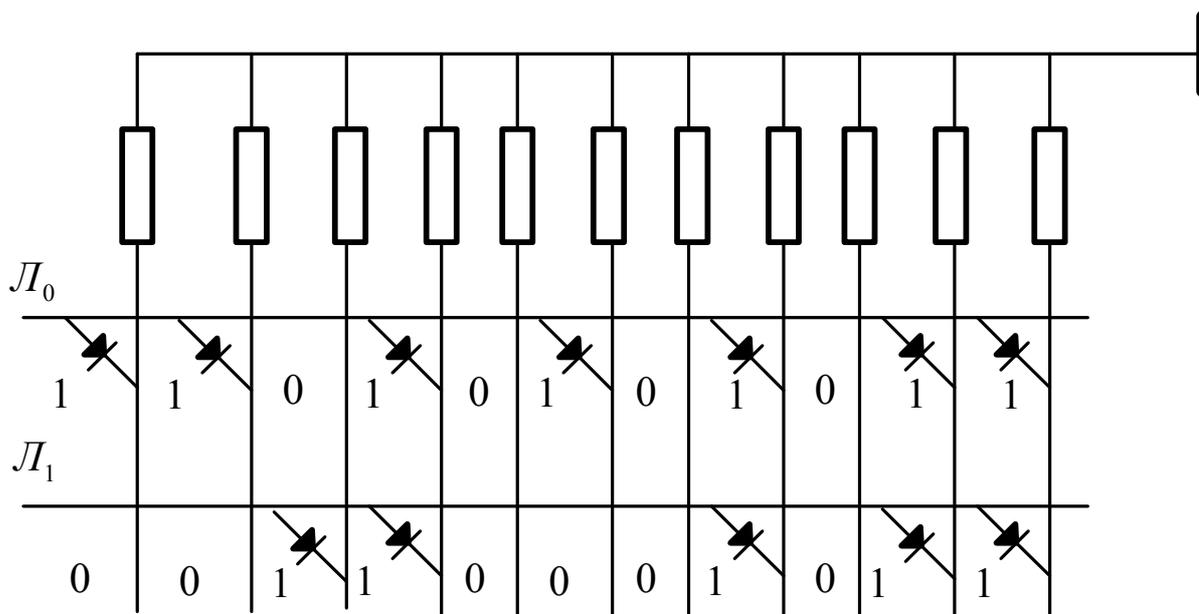


Рис.10.1 Схема масочного ПЗУ на диодах.

Информация в подобное масочное ЗУ заносится в процессе изготовления путём металлизации промежутков, позволяющих соединить через диоды соответствующие линии строк и столбцов. Каждая прошивка требует своего фотошаблона. Масочные ЗУ наиболее надёжны и допускают высокую плотность записи информации.

На рис. 10.2 представлено масочное ЗУ на основе униполярных МОП-транзисторов.

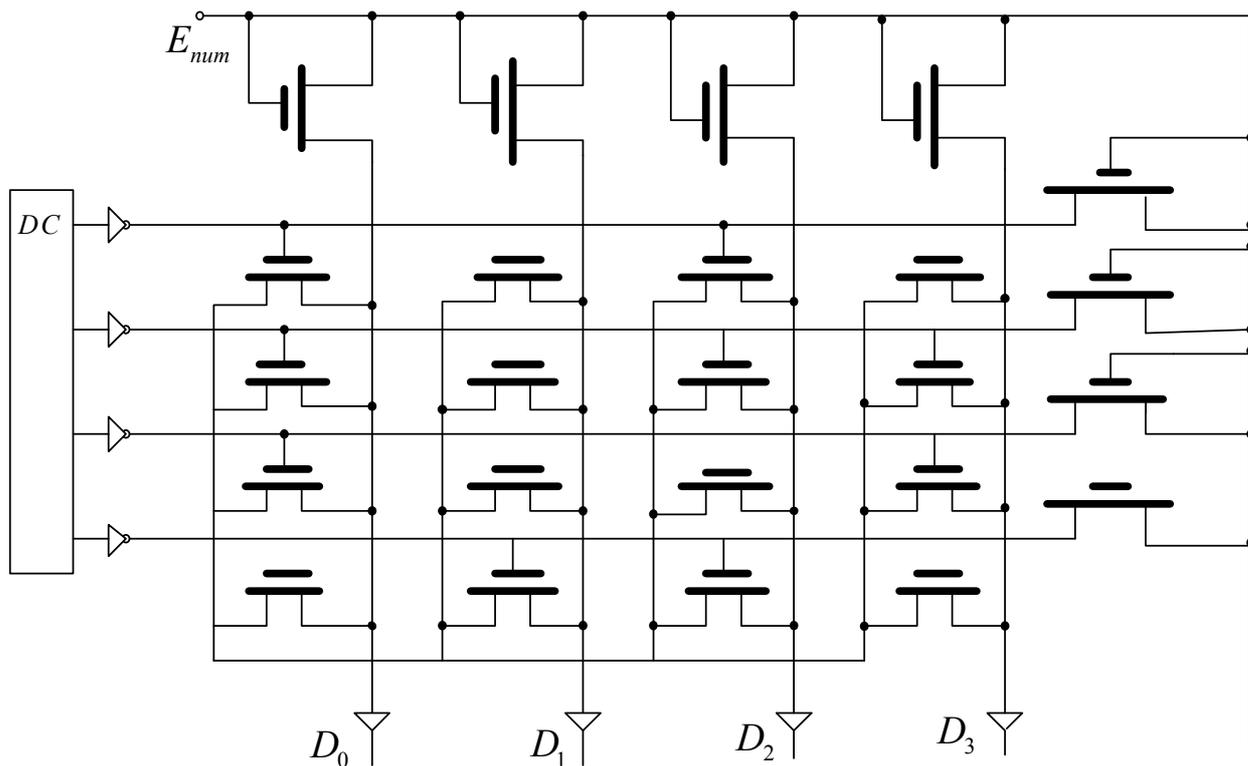


Рис. 10.2. Масочное 3У основе униполярных МОП-транзисторов.

Программирование заключается в подключении затвора к адресной линии в тех местах матрицы памяти, где должна быть записана единица. Там, где должен быть записан 0, коммутация затвора на адресную линию отсутствует.

10.2.2. Однократно-программируемое 3У.

3У типа PROM программируются пользователем устранением или созданием перемычек рис.10.3.

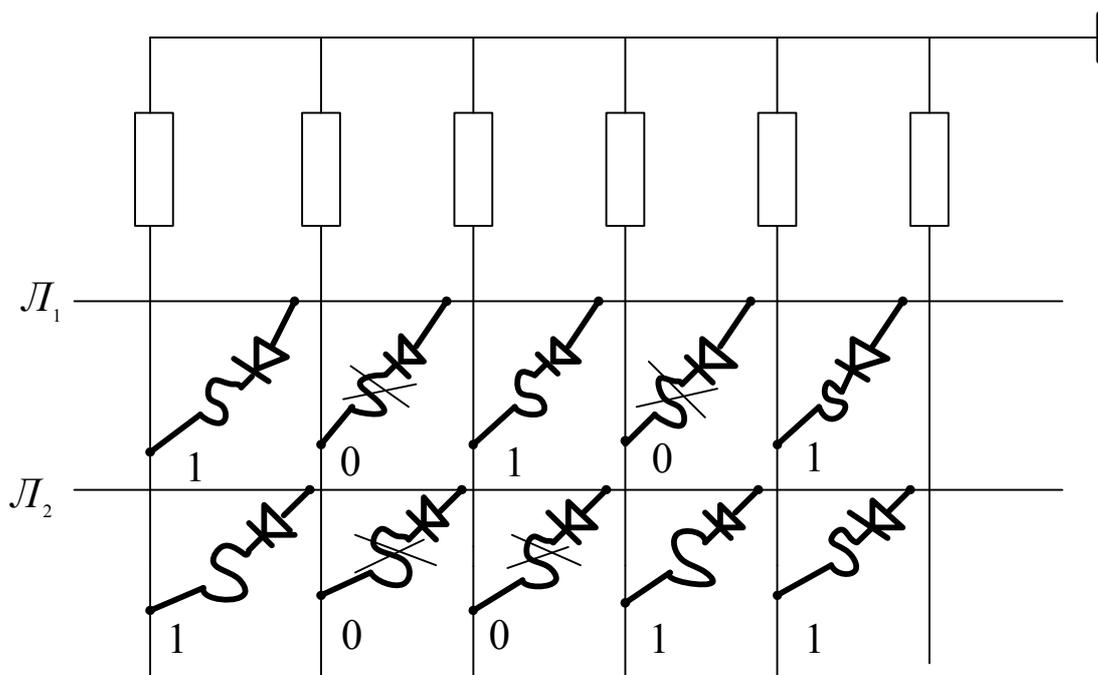


Рис. 10.3. Схема однократно-программируемого ЗУ с пережигаемыми перемычками.

В исходном состоянии ЗУ имеет все перемычки, а при программировании часть их ликвидируется путем расплавления импульсами тока (большой амплитуды и длительности). Эти перемычки включаются в электроды диодов или транзисторов. Перемычки изготавливаются металлическими (нихром) и поликристаллическими (кремниевыми).

Другой тип перемычки - два встречно включенных диода рис.10.4. В исходном состоянии цепь можно считать разомкнутой. Для записи единицы к диодам прикладывается высокое напряжение, пробивающее диод, смещенный в обратном направлении. Диод пробивается с образованием в нем короткого замыкания.

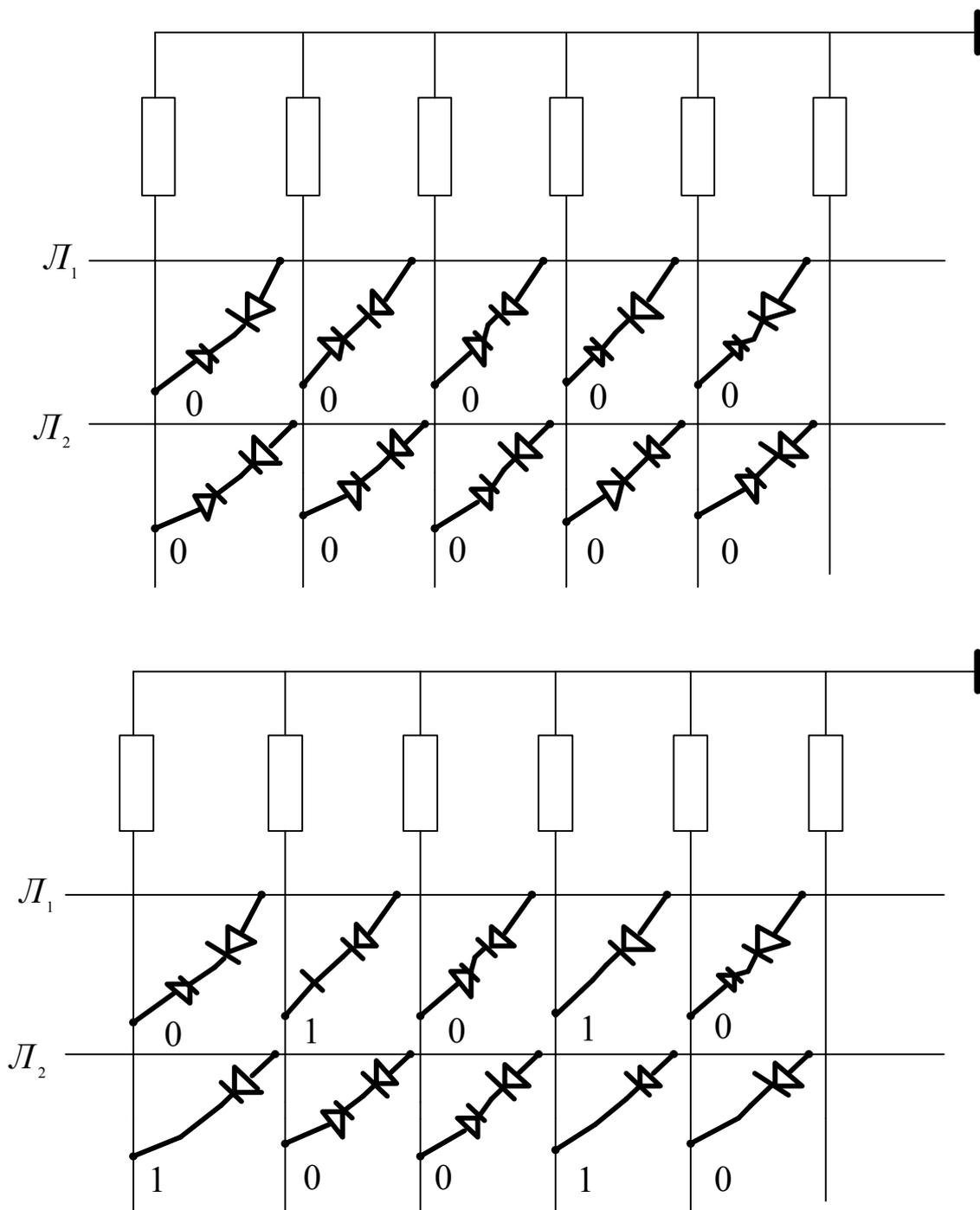


Рис. 10.4. Однократно-программируемое ЗУ на основе двух встречно включенных диодов до программирования и после программирования.

Выпускаемые также схемы с тонкими пробиваемыми диэлектрическими переключками (типа antifuse) более компактны и совершенны. Применяются в ПЛИС.

Представителем ЗУ с плавкими переключками является микросхема K155PE3 (ТТЛ).

Плавкие переключки занимают довольно много места, в связи с этим уровень (степень) интеграции ниже, чем у масочных ЗУ. При этом имеют невысокую стоимость, т.к. изготовитель выпускает микросхему без учета конкретного содержимого ЗУ. Программирует ЗУ пользователь.

Среди отечественных PROM ведущее место занимают микросхемы серии K556. Емкость 1-64 Кбит и $t_{\text{доступа}}=70-90$ нс. Рассмотренную структуру имеют масочные – запрограммированными изготовителем (ROM), однократно программируемые пользователем (PROM или OTP) постоянные запоминающие устройства.

10.2.3. Перепрограммируемые ПЗУ.

Перепрограммируемые ПЗУ позволяют многократно осуществлять прошивку памяти. В настоящее время наибольшее распространение получили ПЗУ с электрическим стиранием предыдущей записанной в ПЗУ информации. Ячейка хранения такого ПЗУ строится на основе полевого транзистора МОП-типа с двойным подзатворным диэлектриком. Эффект хранения бита информации основывается на двух состояниях полевых транзисторов. Транзистор может иметь «плавающий» заряд в подзатворном пространстве или не иметь его. Если в транзисторе создан заряд, то для перевода транзистора в открытое состояния поданного на затвор напряжения будет недостаточно. Транзистор останется в высокоимпедансном состоянии. Если заряд не создан, то полевой транзистор переведётся в насыщенное (открытое) состояние и тем самым обеспечит соответствующее напряжение

на линии считывания. На рис. 10.5. представлена структура МОП-транзистора с двойным подзатворным диэлектриком.

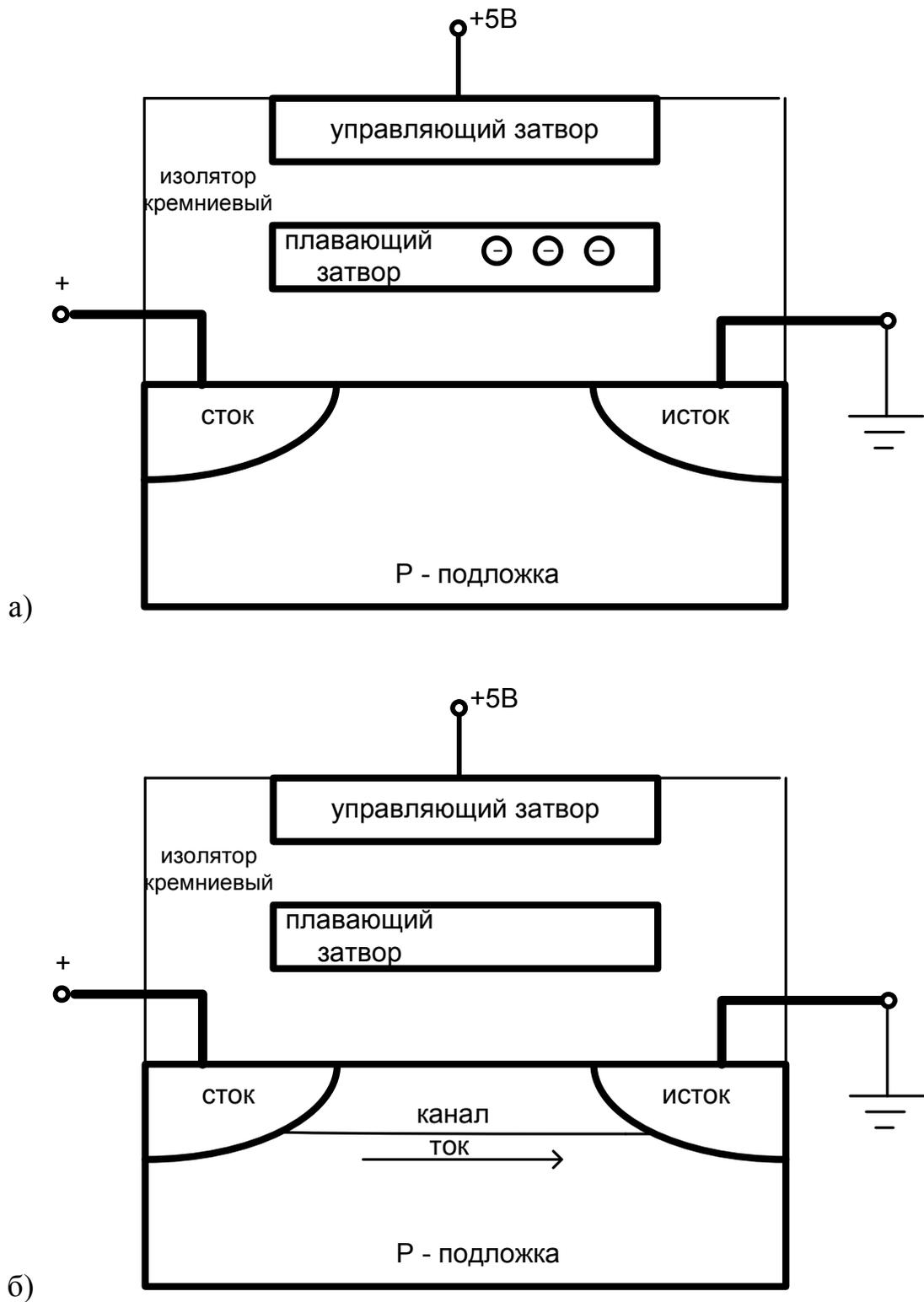


Рис.10.5. Структура МОП-транзистора с двойным подзатворным диэлектриком: а) плавающий затвор с созданным зарядом препятствует

созданию проводящего канала; б) плавающий затвор не имеет заряда, проводящий канал создаётся, между стоком и истоком протекает ток.

Чтобы записать в плавающий затвор информацию создаём большую разность потенциалов между стоком и истоком и естественно положительный на затвор.

Возникает канал. В транзисторе в этом случае протекает большой ток, причём такой силы, что, слегка пробивает изолятор и заносит электроны в плавающий затвор. При резком выключении тока электроны остаются в плавающем затворе, так как покинуть его уже не могут ведь энергию для преодоления изолятора даёт большой ток. В общем, плавающий затвор остаётся отрицательно заряжен при полном отключении напряжения. Ток через такой транзистор не пойдёт в дальнейшем, так как отрицательный заряд эффектом поля закрывает канал. Чтобы стереть заряд на плавающем затворе подаём на затвор большое напряжение "сгоняющее" с плавающего затвора электроны и, следовательно, заряд.

При отсутствии заряда на плавающем затворе и при привычной подаче положительного потенциала на затвор канал открывается и ток протекает, создавая падение напряжения, соответствующее логической единице..

Как мы видим, есть два устойчивых состояния:

1)ток не идёт через транзистор, есть заряд на плавающем затворе

2)ток идёт, нет заряда на плавающем затворе,

а два устойчивых состояния дают право такому транзистору использоваться в качестве запоминающего устройства типа флэш-память.

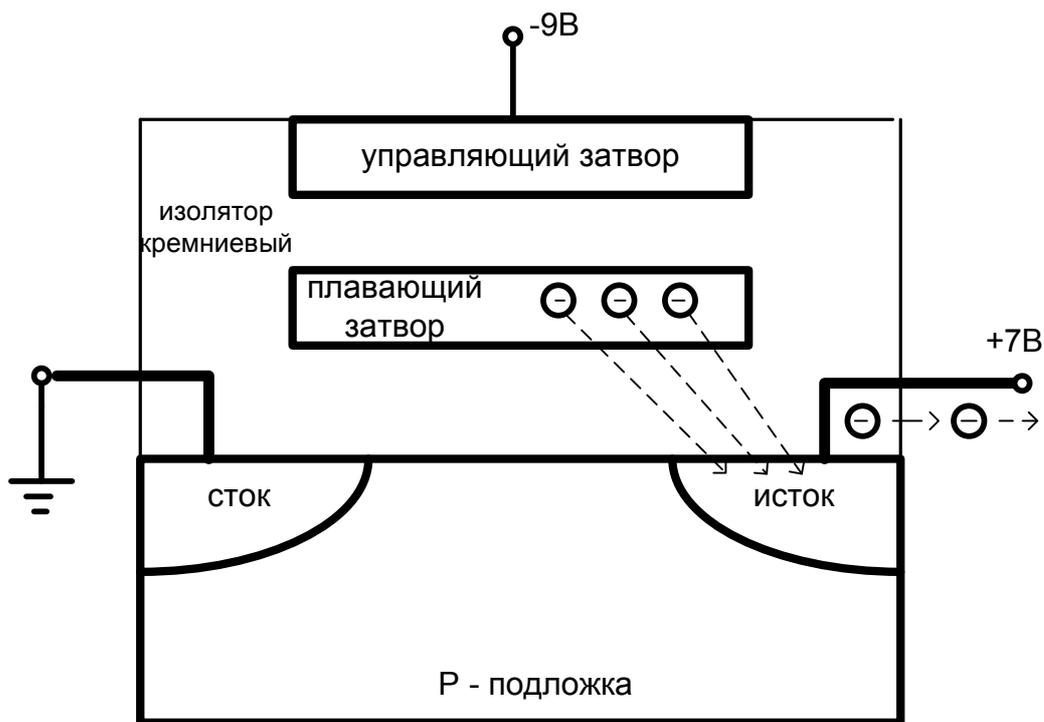


Рис. 10.6. Рассасывание плавающего заряда в МОП-транзисторе.

Заряд может сохраняться не менее 15-25 тысяч часов во включённом состоянии и до 100 тысяч часов (более 10 лет) в выключенном состоянии. Стирание информации производится напряжением противоположной полярности напряжения, при этом происходит восстановление прежнего порога включения полевого транзистора.

10.3. Оперативные запоминающие устройства.

Основной признак оперативного ЗУ – хранение информации только при наличии питающего напряжения. После повторного включения питания восстановить содержимое оперативного ЗУ можно только повторив операцию записи. Оперативные ЗУ подразделяются на:

- статические ОЗУ;
- динамические ОЗУ.

10.3.1. Статические оперативные запоминающие устройства.

Область применения относительно дорогостоящих статических ОЗУ определяется их высоким быстродействием. В частности, они используются в специализированной «быстрой» кэш-памяти. Данная память при сравнительно небольшой ёмкости, должна иметь максимальное быстродействие. Схемотехническую основу ОЗУ статического типа составляют регистры на основе триггеров. Современные схемы ОЗУ строятся на полевых транзисторах. Среди транзисторных технологий наиболее распространённой стала схема триггера, выполненная по К-МОП технологии. На рис. 10. 7. представлена запоминающая ячейка (RS-триггер) для хранения одного бита информации на основе n-МОП транзисторов.

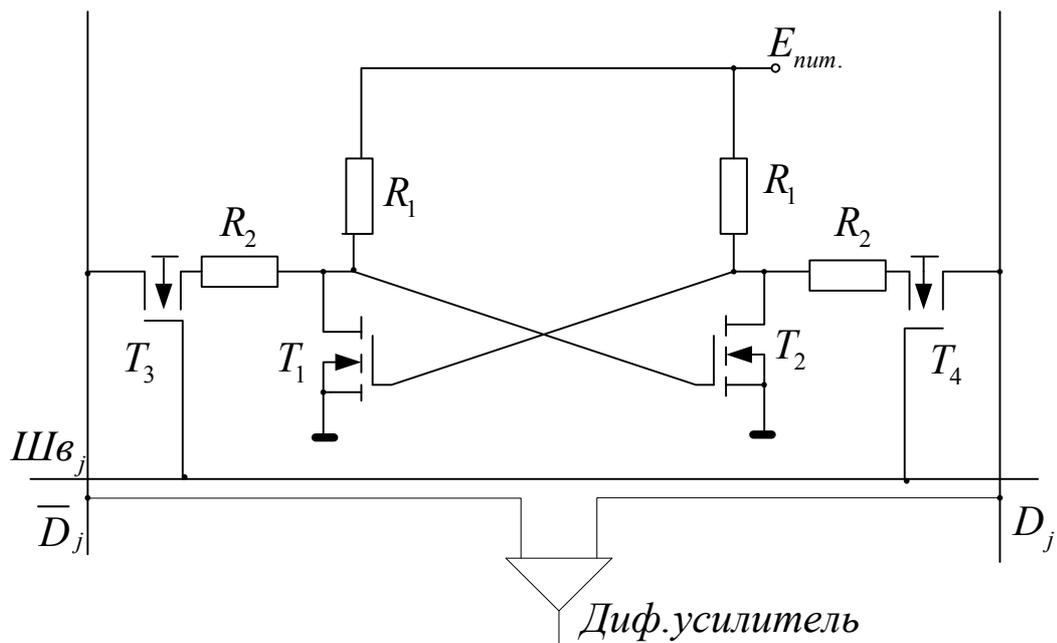


Рис. 10.7. Схема триггерного запоминающего элемента статического ОЗУ на n-МОП транзисторах.

На $\overline{Шв}_j$ поступает высокий потенциал от дешифратора адреса для выборки строки памяти, который открывает вспомогательные транзисторы T_3 и T_4 у всех триггеров, подключённых к данной шине выборки. Через

столбцовые шины выборки \overline{D}_j и D_j и вспомогательный дифференциальный усилитель можно считать состояние запоминающей ячейки (RS – триггера). Столбцовые шины позволяют через них установить состояние триггера. Установочный сигнал снижает стокое напряжение запертого основного транзистора хранения, что приводит к переводу противоположного транзистора в запертое состояние. Резисторы R_1 служат для уменьшения ёмкостных токов. Вместо сопротивления R_1 , как правило, используется n-МОП транзистор с нулевым напряжением затвора. Дифференциальный усилитель необходим из-за режима очень малых токов через столбцовые шины.

Структурная схема статического ОЗУ приведена на рис. 10.8. Вход и выход ОЗУ в этой схеме объединены при помощи шинного формирователя. Естественно, что схемы реальных ОЗУ будут отличаться от приведенной на этом рисунке. Тем не менее, приведенная схема позволяет понять, как работает реальное ОЗУ.

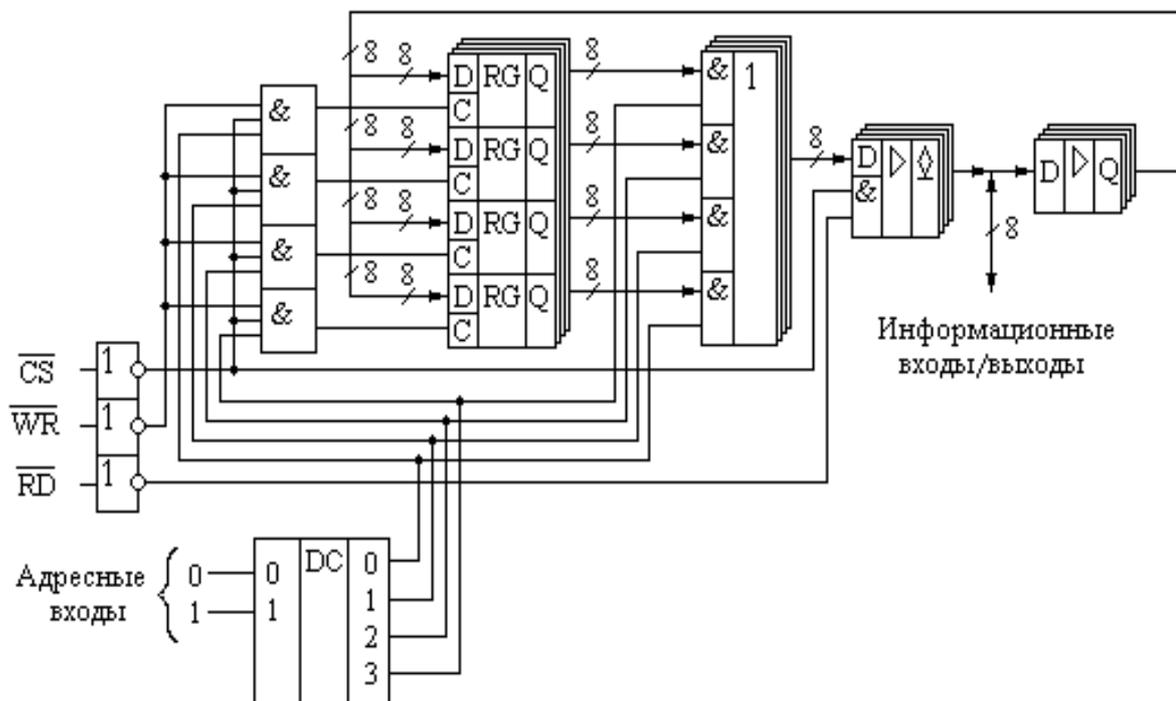


Рис. 10.8. Функциональная схема статического ОЗУ.

Условно-графическое обозначение ОЗУ на принципиальных схемах приведено на рис. 10.9.

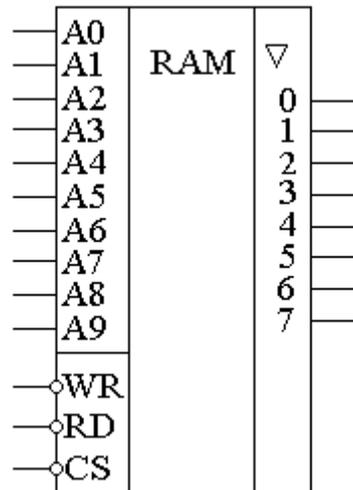
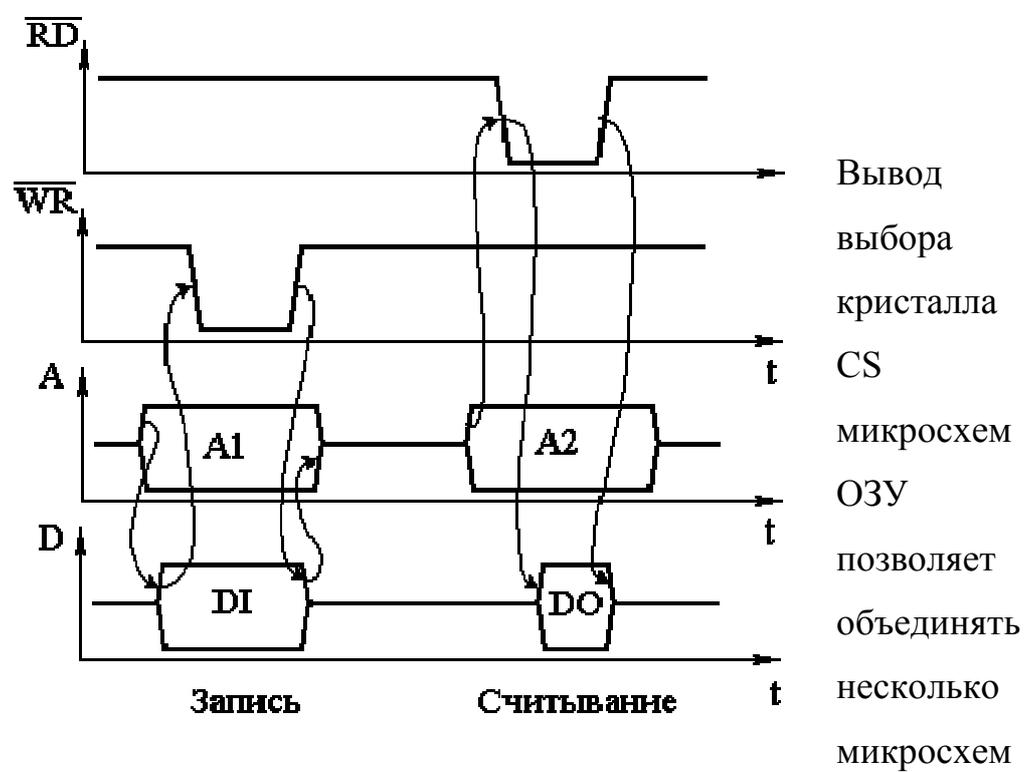


Рис. 10.9. Условно-графическое изображение статического ОЗУ.

Сигнал записи WR позволяет записать логические уровни, присутствующие на информационных входах во внутреннюю ячейку ОЗУ (RAM). Сигнал чтения RD позволяет выдать содержимое внутренней ячейки памяти на информационные выходы микросхемы. В приведенной на рис. 10.7. схеме невозможно одновременно производить операцию записи и чтения, но обычно это и не нужно.

Конкретная ячейка ОЗУ выбирается при помощи двоичного кода — адреса ячейки. Объем памяти ОЗУ (RAM) зависит от количества ячеек, содержащихся в ней или, что то же самое, от количества адресных проводов. Количество ячеек в ОЗУ можно определить по количеству адресных проводов, возводя 2 в степень, равную количеству адресных выводов в микросхеме:

$$M = 2^n$$



для увеличения объема памяти ОЗУ. Статические ОЗУ требуют для своего построения большой площади кристалла, поэтому их ёмкость относительно невелика. Статические ОЗУ применяются для построения микроконтроллерных схем из-за простоты построения принципиальной схемы и возможности работать на сколь угодно низких частотах, вплоть до постоянного тока. Кроме того статические ОЗУ применяются для построения кэш-памяти в универсальных компьютерах из-за высокого быстродействия статического ОЗУ. Временные диаграммы чтения из статического ОЗУ совпадают с временными диаграммами чтения из ПЗУ. Временные диаграммы записи в статическое ОЗУ и чтения из него приведены на рис. 10.10.

Рис.10.10. Временные диаграммы работы статического ОЗУ.

10.3.2. Динамические оперативные ЗУ.

Динамические ОЗУ обладают рядом преимуществ по сравнению со статическими ОЗУ, благодаря чему получили распространение и применение в вычислительной техники. Как и ОЗУ статического типа

они теряют свой хранимый код при отключении электропитания. Принцип хранения данных в динамическом ЗУ основан на создании заряда на микроконденсаторе в составе полевого транзистора МОП-типа, являющегося вместе с конденсатором запоминающей ячейкой динамического ОЗУ. Одно из преимуществ динамического ЗУ – в 4-5 раз меньшая площадь запоминающей ячейки по сравнению со статическим ЗУ. Однако из-за малой ёмкости конденсатора в 0.1...0.2 пф. Его необходимо перезаряжать через несколько миллисекунд. Эта технология получила термин: регенерация памяти. Однако этот недостаток не повлиял на бурное развитие и совершенствование схем подобной динамической памяти. Есть различные варианты реализации схем запоминающей ячейки подобного ЗУ, но лидерами по компактности являются однотранзисторные ЗУ. На рис. 10.11. приведена схема запоминающей ячейки и на рис. 10.12 структура полевого транзистора с микроконденсатором в цепи стока.

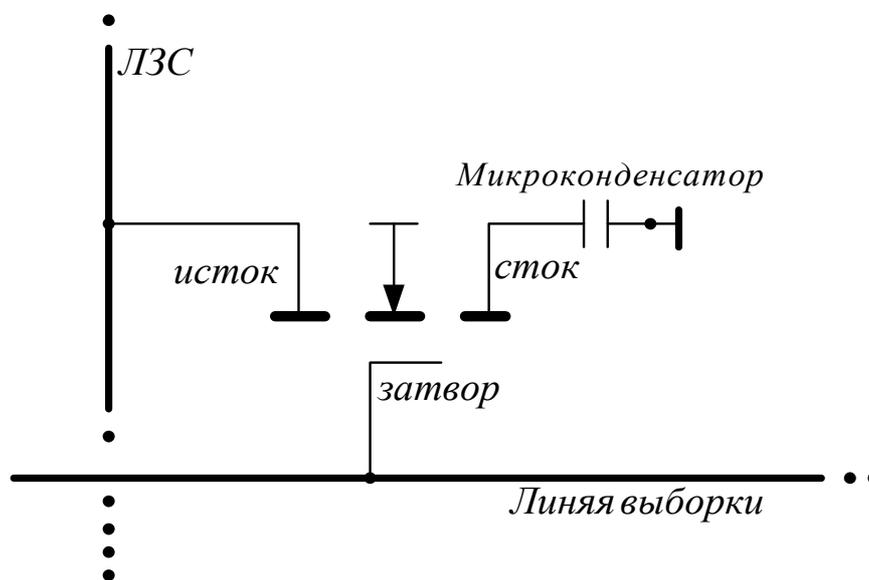


Рис.10.11. Схема запоминающей ячейки динамического ОЗУ на основе МОП-транзистора и конденсатора.

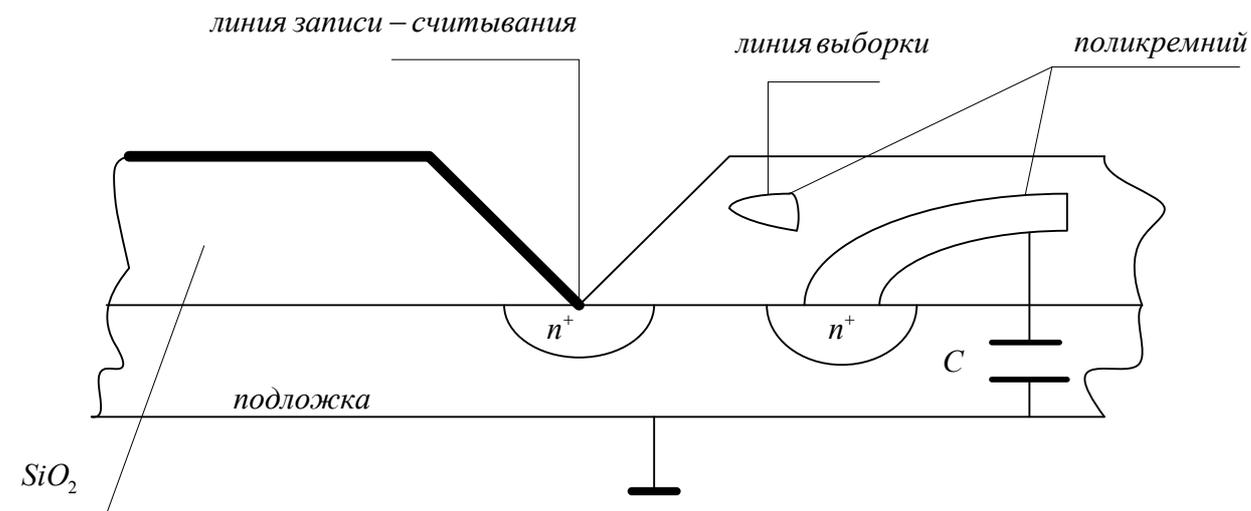


Рис.10.12. Структура полевого МОП-транзистора с встроенным конденсатором.

Ключевой транзистор запоминающей ячейки с встроенным в структуру стока микроконденсатором подключается к линии записи считывания посредством подключения к ней истока. Сток транзистора не имеет внешнего вывода и образует одну из обкладок конденсатора. Другая обкладка конденсатора соединяется с заземлённой подложкой. Между обкладками располагается тонкий слой поликремния SiO_2 . В режиме хранения транзистор заперт. При выборке запоминающего элемента на затвор подаётся напряжение, отпирающее транзистор. Запоминающая ёмкость (конденсатор) через проводящий канал транзистора подключается к линии записи-считывания. При различном состоянии конденсатора состояние потенциала линии записи-считывания изменяется по-разному.

Если конденсатор хранил заряд, то этот заряд, стекая на ЛЗС, увеличивает её номинальное значение потенциала. Если конденсатор не был заряжен, то потенциал ЛЗС понизится, т.к. произойдёт зарядка конденсатора. Данный вид считывания является «разрушающим», т.е. требует восстановления данных после их считывания (регенерации). Рис. 10.13. поясняет состояние изменения потенциала на линии записи-считывания.

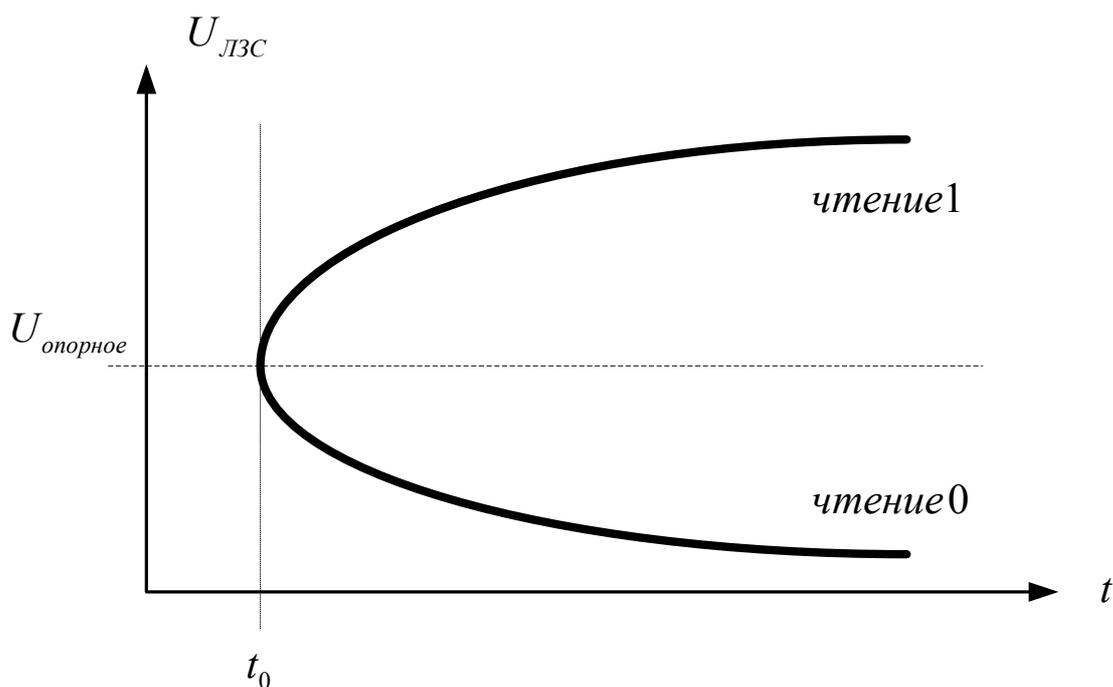


Рис.10.13. Пояснение изменения потенциала ЛЗС при считывании 1 и 0.

Итоги изучения главы 10.

Помнить:

- наименования типов памяти, применяемых в вычислительных устройствах, их условно-графические отображения и основные схемотехнические отличия.

Понимать:

- технологические аспекты операций «программирования» памяти различного типа.
- различия в физических принципах считывания хранимого в различных типах памяти кода.

Применять:

- принимать решения по выбору подходящего типа памяти для конкретных вычислительных устройств и вспомогательных блоков.

10.4. Контрольные вопросы.

Вопросы категории 1. «Помнить».

10.4.1. Сформулируйте различия между постоянной и оперативной памятью.

10.4.2. Каков порядок времени сохранения заряда микроконденсатора в динамическом ОЗУ?

10.4.3. Каков порядок ёмкости микроконденсатора в динамическом ОЗУ?

Вопросы категории 2. «Понимать».

10.4.4. На каком физическом принципе основана запись информации в ПЗУ перемычного типа?

10.4.5. На каком физическом принципе основана запись информации в ПЗУ с встречными диодами?

10.4.6. На каком физическом принципе основана запись информации в ПЗУ с электрическим стиранием?

10.4.7. Чем отличается структура полевого МОП-транзистора от МОП-транзистора, используемого во флэш-памяти?

Вопросы категории 3. «Применять».

10.4.8. Какого направления электрическое поле должен создавать «плавающий» заряд в перепрограммируемом ПЗУ для уверенного распознавания записанной в ПЗУ информации?

10.4.9. Как доказать, что считывание оперативного ОЗУ является разрушающим?

Ответы на контрольные вопросы.

10.4.1. Постоянная память (ПЗУ) сохраняет информацию при отключении питания, оперативная память (ОЗУ) не сохраняет информацию при отключенном питании.

10.4.2. Период регенерации микроконденсатора в динамическом ОЗУ составляет 1-10 миллисекунд.

10.4.3. Ёмкость микроконденсатора динамического ОЗУ составляет 0.1-0.2 пф.

10.4.4. Запись кода производится пережиганием плавких перемычек между горизонтальной адресной линией и вертикальной разрядной линии считывания.

10.4.5. При программировании ПЗУ со встречно расположенными диодами производится пробой одного из диодов, тем самым диод превращается в проводник.

10.4.6. Принцип программирования ПЗУ с электрическим стиранием состоит в создании плавающего заряда в подзатворном пространстве полевого транзистора.

10.4.7. У МОП-транзисторов, используемых во флэш-памяти по иному реализовано подзатворное пространство, имеющее отделённую двумя диэлектриками область для накопления «плавающего» заряда.

10.4.8. Направление электрического поля плавающего заряда должно быть противоположно вектору электрического поля, создаваемого напряжением, приложенным к затвору.

10.4.9. Фактор «разрушения информации» при чтении кода из данного типа памяти обуславливается разрядом либо зарядом конденсатора во время считывания.

Глава 11. Задания и вопросы для самостоятельной подготовки к рубежному контролю по модулю 2.

11.1. С какой логической функцией совпадает функция переноса одноразрядного сумматора?

11.2. С помощью каких устройств осуществляется управление направлением счёта в реверсивном счётчике?

11.3. Сформулируйте конструктивную особенность в схеме построения сумматора дополнительного кода.

11.4. Синтезируйте схему полного компаратора для сравнения двух одноразрядных чисел.

11.5. На каких типах триггеров реализуется сдвигающий регистр?

11.6. Перечислите известные вам схемы организации переносов, назовите самую быстродействующую схему организации переноса.

- 11.7. Сформулируйте отличительную особенность в схеме построения синхронного счётчика. Поясните принцип управления переключения разрядов.
- 11.8. Поясните методику синтеза схемы одноразрядного четвертьсумматора на элементах второго функционально-полного набора (И - НЕ).
- 11.9. Сформулируйте конструктивную особенность в схеме построения сумматора обратного кода.
- 11.10. Схемотехническая особенность в схемах построения асинхронных счётчиков. Поясните принцип переключения состояния разрядов счётчика.
- 11.11. Статические и динамические запоминающие устройства. Сформулируйте отличия и область применения.
- 11.12. Сформулируйте различия в управлении переключением разрядов в асинхронном и синхронном счётчике.
- 11.13. Поясните термин «накапливающий» применительно к сумматорам. Приведите пример.
- 11.14. Сформулируйте отличия декрементора и инкрементора и их отличия в схемах построения от сумматора и вычитателя.
- 11.15. Какую связь необходимо изменить для преобразования суммирующего асинхронного счётчика на JK-триггерах в вычитающий счётчик?
- 11.16. Перечислите функции, выполняемые компараторами.
- 11.17. На каком принципе основано хранение бита информации в динамическом ОЗУ.
- 11.18. На каком принципе основано хранение и запись бита информации в перепрограммируемом ПЗУ?
- 11.19. Сформулируйте отличия модифицированного дополнительного кода от дополнительного кода и его назначение по применению.
- 11.20. К каким последствиям приводит переполнение при применении дополнительного кода при сложении чисел со знаками?

- 11.21. На каком принципе основана работа схемы универсального сумматора вычитателя?
- 11.22. Каким недостатком обладают асинхронные счётчики?
- 11.23. В чём состоит различие между инкрементором и счётчиком?
- 11.24. Сформулируйте принцип организации переноса в сумматорах с параллельным переносом.
- 11.25. С какой логической функцией совпадает логика сигнала заём в полувычитателе?
- 11.26. На базе каких типов триггеров строятся счётчики?
- 11.27. Возможно ли построить синхронный счётчик с использованием D-триггеров?
- 11.28. Перечислите разновидности схем однократно-программируемых ПЗУ.
- 11.29. Поясните понятие «модуль счётчика».
- 11.30. Какие элементы лежат в основе схем статического ОЗУ?

Ответы на контрольные вопросы для самостоятельной подготовки к рубежному контролю по модулю 2.

- 11.1 Функция переноса одноразрядного сумматора совпадает с булевой функцией «конъюнкция».
- 11.2 Для организации переключения направления счёта в реверсивном счётчике используются поразрядные схемы мультиплексоров, имеющих два информационных входа и одноразрядный адресный вход.
- 11.3 В сумматоре дополнительного кода предусмотрен перенос из старшего сумматора числовой части в сумматор кодов знака.
- 11.4 Синтез компаратора для одноразрядных кодов, выдающих результат операции «равно», «меньше», «больше» изложен в разделе 9.9 на стр. 192-194.
- 11.5 Сдвигающий регистр может быть реализован на D – триггерах динамического типа, JK – триггерах.
- 11.6 Из известных схем организации переносов наиболее распространены: схема последовательного переноса, схема параллельного переноса, схема

сквозного переноса, схема условного переноса, схема параллельного группового переноса, которая является и самой быстродействующей.

11.7 Особенностью синхронного счётчика является то, что счётный сигнал подаётся по шине одновременно на синхровходы триггеров всех разрядов. Логика переключения счётчика управляется переводом триггера данного разряда в счётное состояние (логические единицы на входах J и K), либо иное состояние, когда изменение состояния триггера не меняется при наличии счётного сигнала.

11.8 Варианты реализации четвертьсумматора на логических элементах.

Таблица истинности четвертьсумматора.

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

По таблице истинности можно составить булеву функцию для суммы S .

$$S = \bar{a}b + a\bar{b} = \overline{\overline{\bar{a}b} \wedge \overline{a\bar{b}}}$$

По выражению синтезируется схема четвертьсумматора на элементах «И-НЕ» рис. 11.1.

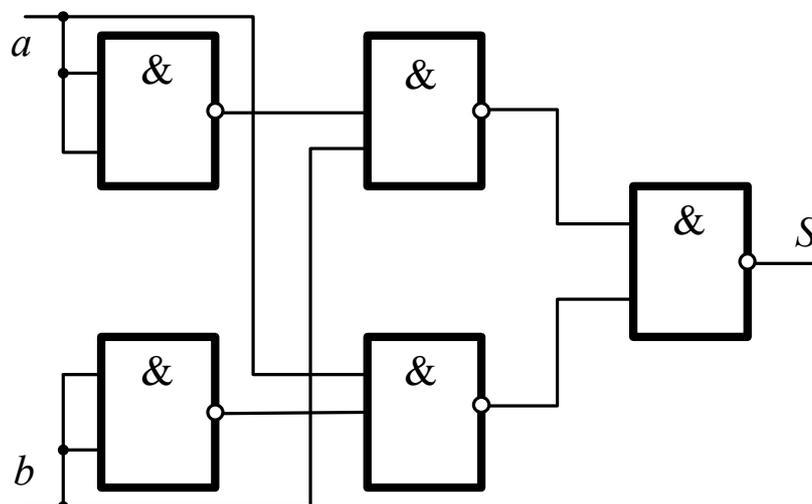


Рис. 11.1. схема четвертьсумматора на элементах «И-НЕ».

11.9 Особенностью сумматора обратного кода является наличие связи по переносу между сумматором знаков и сумматором младших разрядов.

11.10 В асинхронных счётчиках реализуется связь прямого, либо инверсного выхода триггера младшего разряда асинхронного счётчика с счётным входом следующего (старшего) триггера счётчика. Этот сигнал и переключает триггер по счётному входу.

11.11 Статические ЗУ строятся на триггерах и представляют самую «быструю» оперативную память, а динамические ЗУ в качестве хранилища бита информации используют встроенный микроконденсатор. Эти устройства памяти требуют регенерации и восстановления информации после считывания.

11.12 В отличие от асинхронного счётчика в котором переключение определённого разряда полностью предопределено изменением сигнала на выходе соседнего младшего разряда, у синхронного счётчика переключение данного разряда «разрешается» наличием единиц (либо нулей) на всех разрядах младше данного.

11.13 . Под термином «накапливающие» встречаются сумматоры двух разновидностей:

- сумматоры, построенные на основе счётного триггера и логических элементов,

- сумматор со структурой «комбинационный многоразрядный сумматор плюс регистр хранения».

11.14 Инкрементор - это электронное устройство, выполняющее арифметическое сложение многоразрядного кода с единицей младшего разряда. Декрементор - это электронное устройство, выполняющее арифметическое вычитание из многоразрядного кода единицы младшего разряда.

Эти устройства строятся на основе полусумматоров и полувычитателей в силу того, что входы второго аргумента, начиная со второго разряда

задействованы под приём переноса (или заёма) из предыдущего младшего разряда. На вход второго аргумента младшего разряда подаётся единица.

11.15 Компараторы выполняют сравнение многоразрядных кодов с выдачей значений: «равно», «больше», «меньше».

11.16 Для преобразования суммирующего асинхронного счётчика на JK – триггерах в вычитающий необходимо сделать перекоммутацию выходных связей с чётных входов во всех разрядах с прямого выхода JK – триггера на инверсные выходы.

11.17 Принцип хранения данных в динамическом ЗУ основан на создании заряда на микроконденсаторе в составе полевого транзистора МОП-типа. являющегося вместе с конденсатором запоминающей ячейкой динамического ОЗУ.

11.18 Перепрограммируемые ПЗУ позволяют многократно осуществлять прошивку памяти. Ячейка хранения такого ПЗУ строится на основе полевого транзистора МОП-типа с двойным подзатворным диэлектриком. Эффект хранения бита информации основывается на двух состояниях полевых транзисторов. Транзистор может иметь «плавающий» заряд в подзатворном пространстве или не иметь его. Если в транзисторе создан заряд, то для перевода транзистора в открытое состояния поданного на затвор напряжения будет недостаточно. Транзистор останется в высокоимпедансном состоянии. Если заряд не создан, то полевой транзистор переведётся в насыщенное (открытое) состояние и тем самым обеспечит соответствующее напряжение на линии считывания.

11.19 В модифицированных дополнительном и обратных кодах используются под знак два разряда, вместо одного как в дополнительном и обратных кодах. Это позволяет аппаратно определить факт переполнения при операции сложения по несовпадению значений кодов в двух знаковых разрядах.

11.20 При выполнении операции суммирования чисел со знаками в дополнительном и обратных кодах в случае переполнения происходит

перенос в знаковый разряд, что искажает значение результирующего кода знака.

11.21 Особенность данного сумматора состоит в том, что в нём предусмотрено переключение между операцией суммирование и вычитание. Ограничение в его применения заключается в получении не прямого, а дополнительного кода при выполнении операции $c = a - b$, если $a \leq b$.

11.22 У асинхронных счётчиков из-за задержки срабатывания разрядных триггеров имеются небольшие интервалы в которых счётчик показывает неправильный по последовательности переключения код.

11.23 В отличие от сумматора в инкременторе первый аргументом является многоразрядный код, а вторым аргументом – целое число единица.

11.24 Сумматоры с параллельным переносом не имеют последовательного распространения переноса вдоль разрядной сетки. Во всех разрядах сигналы переносов вырабатываются одновременно, параллельно во времени. Сигналы переноса для данного разряда формируются специальными схемами, на входы которых поступают все переменные, необходимые для выработки переносов, т.е. те от которых зависит его наличие или отсутствие.

11.25 Функция сигнала заём в полувычитателе совпадает с булевой функцией «запрет по аргументу а».

11.26 Счётчики строятся на основе триггеров счётного типа, преобразованных из триггеров JK – типа, или динамических D – триггеров с обратной связью с инверсного выхода.

11.27 Для построения синхронного счётчика триггеры D – типа с обратной связью не подходят, так как не имеют управляющего входа блокировки счётного режима.

11.28 Это схемы с пережигаемыми перемычками, с встречно расположенными диодами и биполярными транзисторами.

11.29 Модуль счета или коэффициент пересчета счетчика «К_{сч}»

характеризует число (количество) устойчивых состояний, в которых может находиться N - разрядный счетчик, т. е. предельное число входных сигналов, которое может быть подсчитано счетчиком.

11.30 Статическое ОЗУ строится на основе регистров, построенных на триггерах различного типа.

Список сокращения и обозначений

БФ - булева функция

ДНФ - дизъюнктивно-нормальная форма

ЗУ – запоминающее устройство

КНФ – конъюнктивно-нормальная форма

КС – комбинационная схема

ЛЗС – линия записи-считывания

МДНФ- минимальная дизъюнктивно-нормальная форма

МДП – металл-диэлектрик-полупроводник

МКНФ – минимальная конъюнктивно-нормальная форма

МОП – металл-окисел-полупроводник

ОЗУ – оперативное запоминающее устройство

ПЗУ – постоянное запоминающее устройство

СДНФ – совершенная конъюнктивно-нормальная форма

СКНФ – совершенная дизъюнктивно-нормальная форма

ЦВТ – цифровая вычислительная техника

Список используемых источников.

1. Бойт К. Цифровая электроника. Москва: Техносфера, 2007. -472 с. ил. - (Мир электроники).
2. В.И. Карлащук. Электронная лаборатория на IBM PC. М., "СОЛОН-Р", 2001.
3. Музылева И.В. Элементная база для построения цифровых систем управления: Учебное пособие для вузов. – М.: Техносфера, 2006. – 137 с.: ил. – (Мир электроники).
4. Потёмкин И.С. Функциональные узлы цифровой автоматики. - М.: Энергоатомиздат, 1988. - 320 с.
5. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: Учебное пособие для вузов. - СПб.: Политехника, 1996. - 885 с.
6. Савельев А.Я. Арифметические и логические основы цифровых автоматов: Учебник. - М.: Высшая школа, 1980. -255 с.
7. Угрюмов Е.П. Цифровая схемотехника. - СПб.: БХВ - Санкт-Петербург, 2000 - 528 с.: ил.
8. Электротехника и электроника в экспериментах и упражнениях: Практикум на Electronics Workbench: В 2-х томах /Под общей редакцией Д.И.Панфилова. М.: ДОДЭКА,2000.

