

С.Л. Прищепа

Конспект лекций по курсу:

«Цифровые и микропроцессорные устройства систем телекоммуникаций».

Часть I. "Цифровые устройства систем телекоммуникаций"

для специальности 45.01.01 «Многоканальные системы телекоммуникаций»

45.01.02 «Системы радиосвязи, радиовещания и телевидения»

45.01.03 «Сети телекоммуникаций»

Библиотека БГУИР

Содержание

1. Логические основы цифровой техники
 - 1.1. Основные законы и следствия Булевой алгебры
 - 1.2. Функционально полная система логических элементов
 - 1.3. Минимизация логических функций с помощью алгебраических преобразований
 - 1.4. Минимизация логических функций с помощью карт Карно
 - 1.5. Карты Карно для большого числа переменных
 - 1.6. Основные параметры цифровых схем
 - 1.7. Элемент И-НЕ в ДТЛ
 - 1.8. Элемент И-НЕ в ТТЛ
 - 1.9. ТТЛ со сложным инвертором
 - 1.10. Логические элементы с тремя устойчивыми состояниями
 - 1.11. Логические элементы с открытым коллектором
 - 1.12. Типы выходных каскадов цифровых элементов
 - 1.13. Логический выход цифровых элементов
 - 1.14. Режим неиспользуемых входов цифровых микросхем
 - 1.15. Режим неиспользуемых элементов в цифровых микросхемах
- Наращивание числа входов
2. Функциональные узлы комбинационного типа
 - 2.1. Шифратор. Построение на логических элементах
 - 2.2. Дешифратор. Построение на логических элементах
 - 2.3. Построение приоритетного двоичного шифратора
 - 2.4. Нарращивание размерности приоритетного шифратора
 - 2.5. Нарращивание размерности дешифратора
 - 2.6. Воспроизведение произвольных логических функций с помощью дешифратора
 - 2.7. Мультиплексор
 - 2.8. Демультимплексор
 - 2.9. Нарращивание размерности мультиплексора
 - 2.10. Универсальные логические модули (УЛМ) на основе мультиплексоров
 - Настройка константами. Расширение алфавита сигналов настройки
 - 2.11. Полусумматор. Синтез полного сумматора из полусумматоров
 - 2.12. Синтез полного сумматора на логических элементах
 - 2.13. Сумматор параллельного действия с последовательным переносом
 - 2.14. Параллельный сумматор с параллельным переносом
 - 2.15. Цифровые компараторы
 - 2.16. Преобразователь двоично-десятичного кода в код Айкена
 - 2.17. Преобразователь кода для семисегментной индикации
3. Последовательностные цифровые устройства
 - 3.1. Асинхронный RS-триггер с прямыми входами
 - 3.2. Асинхронный RS-триггер с инверсными входами
 - 3.3. Асинхронный JK-триггер
 - 3.4. Счётный T-триггер
 - 3.5. Функциональное назначение входов триггера
 - 3.6. Синхронные триггеры на логических элементах
 - 3.7. Синхронный RS-триггер
 - 3.8. Синхронные JK- и T-триггеры
 - 3.9. Синхронный триггер задержки (D-триггер)

- 3.10. Синхронные триггеры со статическим и с динамическим управлением
- 3.11. Синхронные MS-триггеры
- 3.12. Итоговая классификация триггеров
4. Функциональные узлы последовательностного типа
 - 4.1. Словарный метод проектирования функциональных узлов последовательностного типа
 - 4.2. Построение сдвигового регистра словарным методом
 - 4.3. Построение реверсивного регистра словарным методом
 - 4.4. Тактируемые параллельные регистры
 - 4.5. Стробируемые параллельные регистры
 - 4.6. Счётчики
 - 4.7. Суммирующий счётчик с последовательным переносом
 - 4.8. Суммирующий счётчик со сквозным переносом
 - 4.9. Суммирующий счётчик с параллельным переносом
5. Аналого-цифровые и цифроаналоговые преобразователи информации
 - 5.1. ЦАП на основе резисторных матриц R-2R
 - 5.2. АЦП последовательных поразрядных приближений
 - 5.3. АЦП параллельного действия
6. Полупроводниковые запоминающие программируемые устройства
 - 6.1. Масочные ЗУ (ПЗУ)
 - 6.2. Программируемые постоянные запоминающие устройства (ППЗУ)
 - 6.3. Программируемые ЗУ с плавкими переключателями
 - 6.4. Классификация СБИС программируемой логики по типу программируемых элементов
7. Учебно-методическая литература
8. Контрольные вопросы
9. Лабораторные работы

1. Логические основы цифровой техники.

1.1. Основные законы и следствия Булевой алгебры.

Для описания логических операций используется математический аппарат, получивший название алгебры логики, или Булевой алгебры.

В алгебре логики рассматриваются переменные, которые могут принимать только два значения: 1 и 0.

Основные логические функции:

1. Логическое отрицание НЕ (инверсия). Обозначается в виде черточки над аргументом: $y = \bar{x}$. В качестве примера цепи, реализующей функцию НЕ, можно привести размыкающий контакт реле. При срабатывании реле цепь, в которую входит такой контакт, будет размыкаться.

2. Логическое умножение И (конъюнкция). Символически обозначается:

$$y = x_1 \cdot x_2 \cdot x_3 \cdot \dots \text{ или } y = x_1 \wedge x_2 \wedge x_3 \dots$$

В качестве примера электрической цепи, реализующей функцию И, можно привести соединение последовательно замыкающих контактов нескольких реле. Цепь в этом случае будет замкнута тогда и только тогда, когда сработают все реле.

3. Логическое сложение ИЛИ (дизъюнкция). Операция обозначается выражениями: $y = x_1 + x_2 + x_3 \dots$ либо $y = x_1 \vee x_2 \vee x_3 \vee \dots$

В качестве примера электрической цепи, реализующей функцию ИЛИ, можно привести параллельное соединение замыкающих контактов нескольких реле. Цепь, в которую входят эти контакты, будет замкнута, если сработает хотя бы один контакт.

Основные законы алгебры логики:

1. *переместительный:*

$$a \cdot b = b \cdot a, \quad a + b = b + a;$$

2. *сочетательный:*

$$(a + b) + c = a + (b + c),$$

$$a(b \cdot c) = (a \cdot b) \cdot c;$$

2. *распределительный:*

$$a(b + c) = a \cdot b + a \cdot c,$$

$$a + b \cdot c = (a + b) \cdot (a + c);$$

4. *закон поглощения:*

$$a + ab = a,$$

$$\bar{a} + \bar{a} \cdot \bar{b} = \bar{a};$$

5. *закон склеивания:*

$$a \cdot b + a \cdot \bar{b} = a,$$

$$\bar{a} \cdot \bar{b} + \bar{a} \cdot b = \bar{a};$$

6. *закон отрицания или правило де Моргана:*

$$\overline{a+b} = \overline{a} \cdot \overline{b},$$

$$\overline{a \cdot b} = \overline{a} + \overline{b}.$$

Правило де Моргана справедливо для любого числа переменных:

$$\overline{a + b + c + \dots z} = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \dots \cdot \overline{z}$$

$$\overline{a \cdot b \cdot c \cdot \dots \cdot z} = \overline{a} + \overline{b} + \overline{c} + \dots \overline{z}$$

Для алгебра логики справедливы следующие соотношения:

- 1) $x \vee 0 = x$;
- 2) $x \vee x = x$;
- 3) $x \vee 1 = 1$;
- 4) $x \wedge 0 = 0$;
- 5) $x \wedge x = x$;
- 6) $x \wedge 1 = x$.

1.2. Функционально полная система логических элементов.

Функционально полная система – это такой набор элементов, используя которые можно реализовать любую сколь угодно сложную логическую функцию.

Набор ИЛИ, И, НЕ называется Булевым базисом и является функционально полной системой.

Пример: $f = a \cdot \overline{b} \vee \overline{a} \cdot b$. Для реализации этой функции необходимо: 2 элемента НЕ, 2 элемента И и 1 ИЛИ:

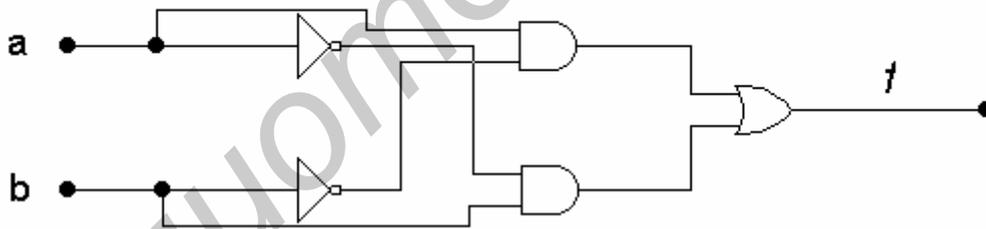


Рис.1.1. Реализация функции f на логических элементах

Функционально полные системы могут состоять и из набора элементов, реализующих логические функции, отличные от простейших. В частности, функционально полные системы могут состоять из элементов только одного типа, например, реализующих функцию И-НЕ либо ИЛИ-НЕ.

Функция Шеффера или отрицание логического умножения И-НЕ (NAND):



Обозначается: $f = \overline{a \cdot b \cdot c \cdot \dots}$

Покажем, что с помощью функции Шеффера можно реализовать все компоненты Булевого базиса.

НЕ:

$$\overline{a \cdot 1} = \overline{a};$$

И:

$$\overline{\overline{a \cdot b}} = a \cdot b;$$

ИЛИ:

$$\overline{\overline{a \cdot b}} = a + b;$$

Функция Пирса или отрицание логического сложения ИЛИ-НЕ (NOR):



Символическое обозначение: $f = \overline{a + b + c + \dots}$

Покажем, что с помощью функции Пирса можно реализовать все компоненты Булевского базиса.

НЕ:

$$\overline{a + 0} = \overline{a};$$

И:

$$\overline{\overline{a + b}} = a \cdot b;$$

ИЛИ:

$$\overline{\overline{a + b + c}} = a + b + c;$$

1.3. Минимизация логических функций с помощью алгебраических преобразований.

Минимизация логических функций применяется при синтезе комбинационных логических цепей (КЛЦ). КЛЦ – это такие цепи, выходные сигналы которых не зависят от предыстории и однозначно определяются сигналами, поступающими на их входы в рассматриваемый момент времени.

Синтез КЛЦ проводят в следующей последовательности:

1) Составляется таблица истинности. Эта таблица показывает, чему равен выходной сигнал цепи при различных комбинациях входных сигналов.

2) Исходя из таблицы истинности, записывается логическая функция.

3) Логическая функция минимизируется и преобразуется к удобному виду для реализации на логических ячейках заданного типа.

Рассмотрим работу мажоритарной ячейки на 3 входа. Строим таблицу истинности:

№	X1	X2	X3	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1

$$\overline{7} \quad 1 \quad 1 \quad 1 \quad 1$$

$$\begin{aligned}
 F &= \overline{X1} \cdot X2 \cdot X3 \vee X1 \cdot \overline{X2} \cdot X3 \vee X1 \cdot X2 \cdot \overline{X3} \vee X1 \cdot X2 \cdot X3 = \\
 &= (X1 \cdot X2 \cdot X3 \vee X1 \cdot X2 \cdot X3) \vee (X1 \cdot \overline{X2} \cdot X3 \vee X1 \cdot X2 \cdot X3) \vee (X1 \cdot X2 \cdot \overline{X3} \vee X1 \cdot X2 \cdot X3) = \\
 &= X2 \cdot X3 (X1 \vee \overline{X1}) \vee X1 \cdot X3 \vee X1 \cdot X2 = \\
 &= X2 \cdot X3 \vee X1 \cdot X3 \vee X1 \cdot X2 = \overline{\overline{X2 \cdot X3 \cdot X1 \cdot X3 \cdot X1 \cdot X3}}
 \end{aligned}$$

Построим схему по полученному выражению:

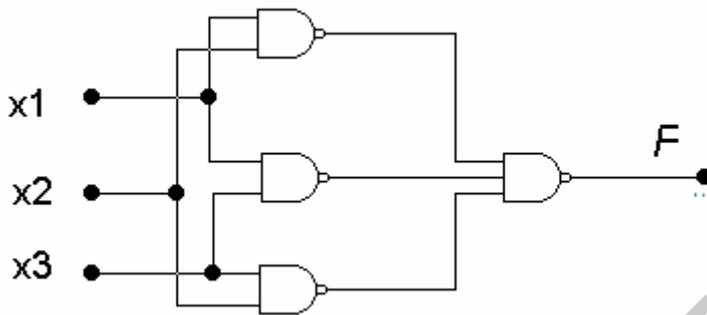


Рис. 1.2. Реализация мажоритарной ячейки на 3 входа по минимизированному выражению

1.4. Минимизация логических функций с помощью карт Карно.

При использовании карт Карно, функцию предварительно следует привести к дизъюнктивной нормальной форме (ДНФ), т.е. выразить в виде логической суммы простых конъюнкций. Простой конъюнкцией считается логическое произведение переменных, взятых с отрицанием или без него, в которых каждая переменная встречается не более одного раза. В простую конъюнкцию не должны входить суммы переменных, отрицания, функции 2 или нескольких переменных. Простая конъюнкция, в которую входят все аргументы рассматриваемых логических функций, называется минтермом. Именно в виде суммы минтермов мы получаем логическую функцию тогда, когда получаем её исходя из таблицы истинности. После того, как приведем функцию к ДНФ, следует заполнить прямоугольную таблицу. Эта таблица и есть карта Карно или диаграмма Вейча.

	$X2 \cdot X3$	$X2 \cdot \overline{X3}$	$\overline{X2} \cdot \overline{X3}$	$\overline{X2} \cdot X3$
X1	11	01	00	10
1	1	1	0	1
0	1	0	0	0

Каждой клетке ставится в соответствие определенная конъюнкция, причем это делается таким образом, чтобы в соседних клетках конъюнкции отличались не более чем на 1 сомножитель. При заполнении в клетку ставится 1, если минимизированная функция при данном наборе аргументов равна 1, в остальные клетки вписываются 0. Обводят прямоугольными контурами все 1, затем записывается минимизированная функция в виде суммы логических произведений, описывающих эти контуры.

Правила проведения контуров:

1. Контур должен быть прямоугольным.
2. Внутри контура должны быть только клетки, заполненные 1.
3. Число клеток, находящихся внутри контура, должно быть степенью числа 2.
4. Одни и те же клетки, заполненные 1, могут входить в несколько контуров.
5. При проведении контуров самая верхняя и самая нижняя строки таблицы считаются соседними. То же самое справедливо для крайних левого и правого столбцов.
6. Число контуров должно быть как можно меньше, а сами контуры - как можно большими.

1.5. Карты Карно для большого числа переменных.

Пример соседнего кодирования для трехразрядного числа:

$X_3 X_2 X_1$	
000	ось симметрии 1 ранга
001	ось симметрии 2 ранга
011	
010	ось симметрии 3 ранга
110	
111	
101	
100	ось симметрии n-ого ранга

Алгоритм:

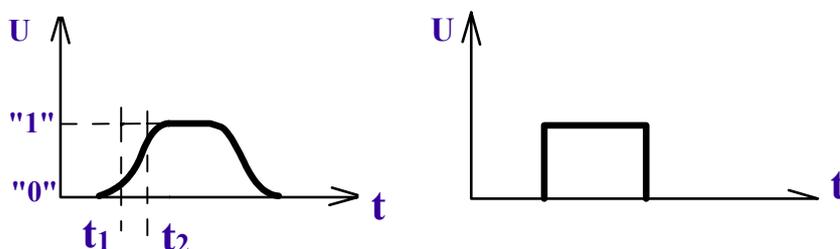
1. Под столбцом r-разрядного соседнего кодирования провести ось симметрии p+1 ранга.
2. Осуществить симметричное отображение относительно оси симметрии p+1 ранга всех r-разрядных кодов.
3. Дополнить r разрядные коды слева одним разрядом, где записаны 0 для всех кодов выше p+1 ранга и 1 для кодов ниже p+1 оси симметрии.

Такое кодирование проводится как для вертикальных, так и для горизонтальных карт Карно.

1.6. Основные параметры цифровых схем.

Основными параметрами цифровых интегральных схем является быстродействие, потребляемая мощность, коэффициент объединения по входу, коэффициент разветвления по выходу, устойчивость против внешних воздействий, помехоустойчивость, степень интеграции, надежность, пороговое напряжение.

Быстродействие определяется средней задержкой сигнала:



t_2 - уровень сигнала, соответствующий 90% от максимального значения.

t_1 – уровень сигнала, соответствующий 10% от максимума.

Время задержки: $t_2 - t_1 = \Delta t_3$.

По быстродействию схемы делятся на:

- сверхбыстродействующие $\Delta t_3 < 5$ нс;
- быстродействующие $5 < \Delta t_3 < 10$ нс;
- средне быстродействующие $10 < \Delta t_3 < 100$ нс;
- низкого быстродействия $\Delta t_3 > 100$ нс.

Схемы ТТЛ относятся к схемам среднего быстродействия. Более высокое быстродействие имеют ТТЛШ и ЭСЛ.

Потребляемая мощность характеризуется средней мощностью, равной полусумме мощностей, потребляемых системой питания, имеющей 2 устойчивых состояния.

Для схем ЭСЛ потребляемая мощность 20-80 мВт, для КМОПТЛ - от 1 до 100 мкВт.

Т.к. снижение средней задержки логических интегральных схем сопровождается ростом потребления мощности, то более распространённым является параметр, называемый работой переключения:

$$W = \Delta t * P$$
$$W = 0,5..5 \text{ пДж.}$$

Таблица. Основные параметры некоторых распространённых ИМС ТТЛ(Ш).

параметр	K155	K555	KP1533	KP533
P_{cp} , мВт	10	2	1,2	19
$\Delta t_{3,cp}$, нс	20	18	14	5
N	10	20	40	10

Помехоустойчивость. Характеризуется параметром статической помехоустойчивости – это логическое постоянное напряжение, которое, будучи добавлено к полезному входному сигналу вызовет ошибку по всей последующей цепи логической схемы.

Для ЭСЛ уровень статических помех 0.1..0,5 В. Для ТТЛ: 0.6..1,1 В, КМОП: 2..3 В (большой логический перепад напряжений).

Коэффициент объединения по входу.

$K_{об}$ - тах число входов, которые может иметь логический элемент (до 8).

Коэффициент разветвления по выходу или нагрузочная способность.

Определяется числом схем этой же серии, входы которой могут быть присоединены к выходам данной схемы без нарушения её работоспособности.

Устойчивость к внешним воздействиям.

Характеризует возможность применения интегральной схемы при изменении температуры, давления, радиации. Большую роль играет корпус. Наименьшая устойчивость к температурному воздействию – у ЭСЛ. Более устойчивы КМОП и ТТЛ.

Степень интеграции.

Характеризует достигнутый уровень производства. Численные значения определяются округлением до большего целого числа значения десятичного логарифма числа элементов в 1 кристалле.

$$\text{Lg}(12889) \approx 4.$$

Однако для потребителей ИС более важна степень интеграции не элементов, а логических функций, т.к. именно она показывает, какое число ИС (корпусов) потребуется для построения того или иного логического устройства. С этой точки зрения логические ИС делят на :

- ИС малой степени интеграции (МИС, в одном корпусе несколько инверторов или один-два триггера);
- ИС средней степени интеграции (СИС, в одном корпусе несколько сложная логическая цепь, например, сумматор или десятичный разряд счётчика);
- ИС большой степени интеграции (БИС, в одном корпусе арифметическое устройство, многоразрядный счётчик);
- ИС сверхбольшой степени интеграции (СБИС).

Надежность.

Для МИС определяется в основном отказами соединения между контактными площадками на кристалле и выводами корпуса. Для БИС и СБИС определяющими являются отказы элементов и соединений внутри самого кристалла. Интенсивность отказов у хороших схем не превышает 10^{-7} 1/ч.

Пороговое напряжение.

Высокого $U^1_{\text{пор}}$ и низкого $U^0_{\text{пор}}$ уровней. Для ТТЛ $U^1_{\text{пор}} < 2\text{В}$, $U^0_{\text{пор}} > 0,5\text{В}$.

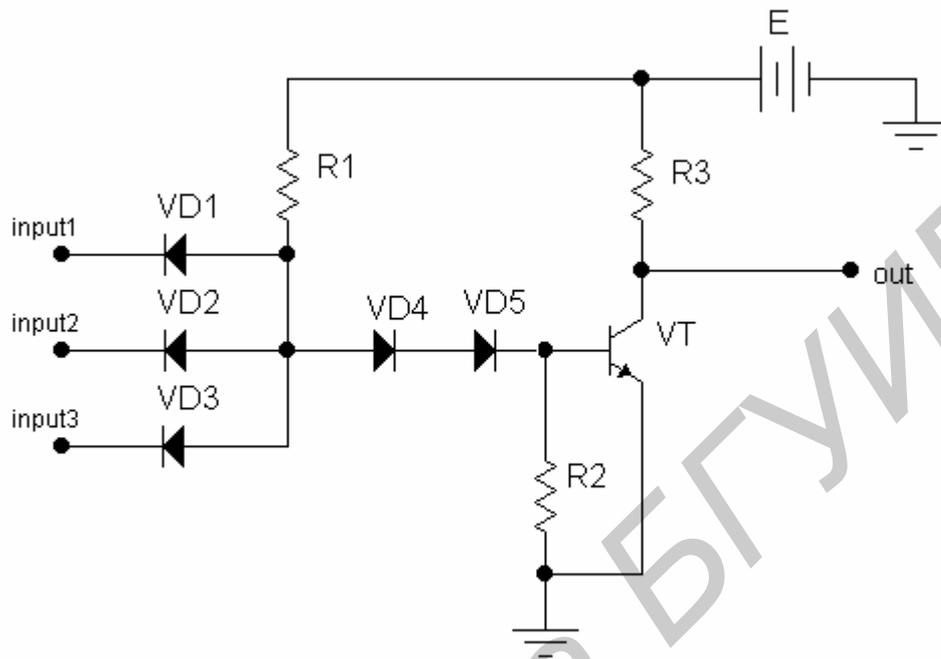
1.7. Элемент И-НЕ в ДТЛ.

Цифровые схемы могут быть построены по-разному, но в их основе, как правило, лежат схемы, выполняющие функции И-НЕ либо ИЛИ-НЕ. Поэтому интегральные схемы содержат обычно схемы И либо ИЛИ, выполненные на резисторах, диодах или транзисторах, и транзисторные инверторы. Транзисторный инвертор может быть простейшим – на одном транзисторе, включённом по схеме с общим эмиттером, или сложным – многотранзисторным с каскадным включением транзисторов в выходном каскаде.

Разберём работу схемы И-НЕ с ДТЛ, работающую от положительных сигналов (рис. 1.3). Схема состоит из двух частей. В первой входные переменные подаются на диодный элемент И. Вторая часть выполнена на транзисторе и представляет собой инвертор. Таким образом в схеме последовательно выполняется логическая операция И-НЕ. Диоды VD3, VD4 называются смещающими диодами и предназначены для надёжного закрывания транзистора.

ДТЛ-элементы обладают большим быстродействием, большим коэффициентом объединения по входу, высокой помехозащищённостью и широко используются в системах цифровой техники. Отсутствие конденсаторов и высокоомных резисторов делает их удобными для микроэлектронного исполнения. Однако, ТТЛ-элементы получили в настоящее время большее распространение. В первую очередь это связано с тем, серьёзным недостатком ДТЛ является большое количество диодов,

каждый из которых необходимо тщательно изолировать, что увеличивает площадь микросхемы.



:Рис. 1.3. Элемент И-НЕ в ДТЛ.

1.8. Элемент И-НЕ в ТТЛ.

По принципу действия, а также по основным характеристикам ТТЛ-элементы близки к ДТЛ.

Рассмотрим работу элемента ТТЛ, выполняющего логическую операцию И-НЕ. Роль изолированных диодов играет эмиттерный переход многоэмиттерного транзистора. Многоэмиттерный транзистор – чисто интегральный прибор, у которого область базы и область коллектора объединяют до 8 эмиттерных переходов. Принципиальная схема элемента И-НЕ на ТТЛ показана на рисунке 1.4.

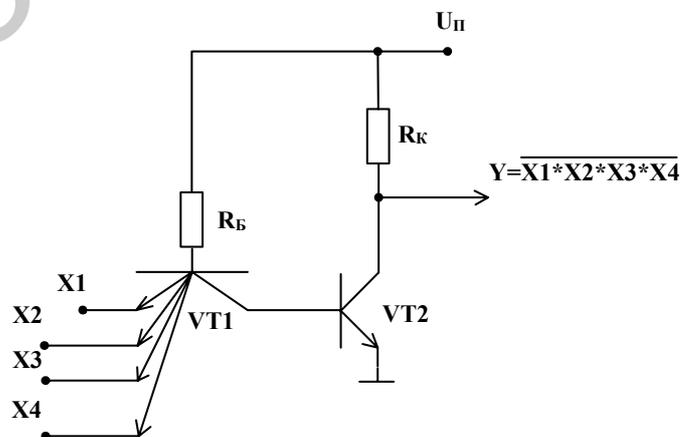


Рис. 1.4. Элемент И-НЕ в ТТЛ.

Данная схема является прототипом трёхвыходового логического элемента ТТЛ серии К155. Типичные значения параметров транзисторно-транзисторной логики следующие: Напряжение питания 5В, уровень логической единицы более 2.8В, уровень логического нуля менее 0.5В, средние времена задержки в диапазоне 10...20 нс, средняя потребляемая мощность 10...15 Вт, коэффициент разветвления по выходу 10.

Недостатком элементов ТТЛ является кратковременное увеличение мощности, потребляемой в переходном режиме, что вызывает появление в цепях питания импульсных помех.

1.9 ТТЛ со сложным инвертором.

В простой схеме И-НЕ подключая выходы схемы на входы многоэмиттерного транзистора, мы искажаем распределение тока в выходной цепи и можем исказить сигнал. Для предотвращения этого используют сложный инвертор.

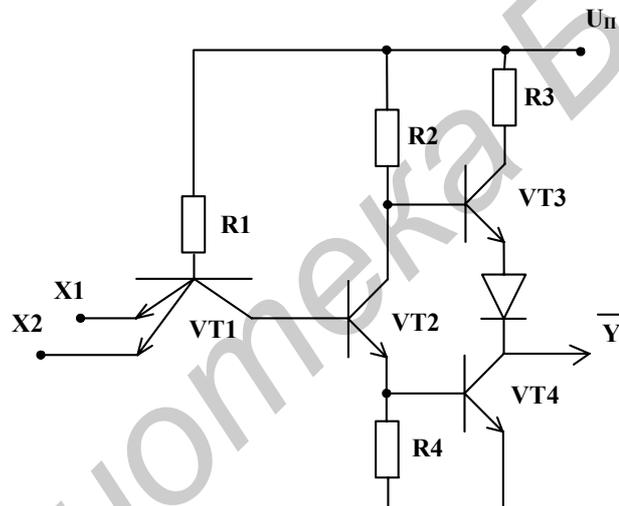


Рис. 1.5. Базовая схема серии К155 со сложным инвертором.

При наличии на входах $X1$ и $X2$ напряжения логического 0, транзистор $VT1$ находится в режиме насыщения, а транзистор $VT2$ закрыт. Следовательно, закрыт и транзистор $VT4$, так как ток через $R4$ не протекает и напряжение на базе $VT4$ равно нулю. Транзистор $VT3$ открыт, т.к. его база подключена к источнику питания через $R2$. Сопротивление $R3$ невелико и $VT3$ работает как эмиттерный повторитель. Через $VT3$ и открытый диод протекает ток нагрузки логического элемента. Напряжение на выходе соответствует уровню логической 1.

При увеличении напряжения на всех входах потенциал базы $VT2$ возрастает и при $U_{вх} > U_{пор}^0$ транзистор $VT2$ открывается, начинает протекать коллекторный ток через $R2$ и $R4$. В результате базовый ток $VT3$ уменьшается, падение напряжения на нём увеличивается и выходное напряжение в схеме снижается. Дальнейшее увеличение входных напряжений приводит к насыщению $VT2$ и $VT4$ и запираению $VT1$. $VT3$ и диод также закрыты, на выходе имеем уровень логического 0.

1.10. Логические элементы с тремя устойчивыми состояниями.

В элементах с тремя устойчивыми состояниями выходные транзисторы заперты сигналом управляющего вывода. Выходное сопротивление запертых транзисторов велико и микросхема практически полностью отключена от нагрузки. Такое состояние называется высокоимпедансным.

OE(Output Enable) =0, схема не работает. OE=1 – схема работает.

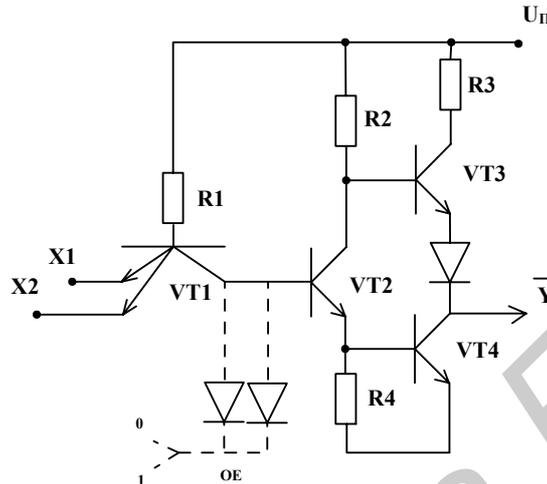


Рис. 1.6. Схема элемента И-НЕ с тремя устойчивыми состояниями

При использовании нескольких ЛЭ с тремя состояниями их выходы подключают к одной нагрузке. Управление микросхем осуществляется так, что в любой момент времени все микросхемы, кроме одной, находятся в высокоимпедансном состоянии. Таким образом удаётся по одной шине передавать в разных направлениях информацию от нескольких источников сигнала и сократить количество информационных магистралей. Вход включения третьего состояния имеет метку EZ, а выход, имеющий состояние высокого импеданса, обозначается через Z, либо V.

1.11. Логические элементы с открытым коллектором.

Обозначаются следующим образом:



ТТЛ элементы имеют небольшие значения выходных сопротивлений. Поэтому нельзя объединить между собой выходы нескольких ТТЛ ЛЭ, т.к. в случае разных выходных сигналов через выходные транзисторы ЛЭ будут протекать большие токи. Для расширения функциональных возможностей у отдельных типоминиатур на выходе ставят транзистор, коллекторная цепь которого оставлена свободной. Это ЛЭ с открытым коллектором (ОК). При использовании подобной ИС коллектор соединяют с источником напряжения через нагрузочный резистор. Роль его может выполнять резистор, обмотка реле, светодиод. Причём напряжение, к

которому подключают нагрузочное сопротивление, может быть значительно больше напряжения питания ТТЛ ЛЭ. Так, например, в МС К155ЛА11 возможно подключение нагрузки к напряжению до 30В.

Кроме того, ЛЭ с ОК позволяют осуществить непосредственное соединение между собой выходов нескольких ИС. При этом обеспечивается реализация дополнительной логической функции "монтажное И".

Наконец, подавая разные значения напряжения питания в схеме с ОК можно получить разные уровни выходного сигнала. Это позволяет осуществить согласование микросхем серии ТТЛ с другими сериями, имеющими другие значения U^0 и U^1 без использования дополнительных преобразователей уровня.

1.12. Типы выходных каскадов цифровых элементов.

Цифровые элементы могут иметь выходы следующих типов:

- логические;
- с открытым коллектором (стоком);
- с открытым эмиттером (истоком).

1.13. Логический выход цифровых элементов.

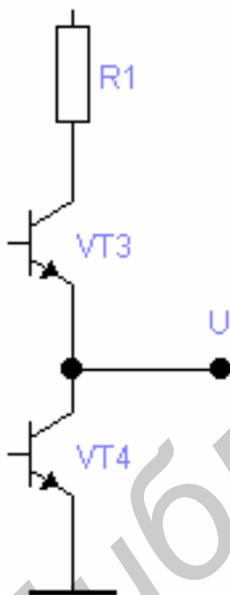
Формирует два уровня выходного напряжения: U^0 и U^1 . Входное напряжение стремятся сделать небольшим, способным развивать большие токи для перезарядки емкостных нагрузок и для получения высокого быстродействия элементов. Такой тип выходов имеют большинство вентилях, используемых в комбинационных цепях.

Рассмотрим выходной каскад элемента ТТЛ. Оба транзистора не бывают закрыты. Схемы логических выходов элементов ТТЛШ или КМОП подобны двухтактным каскадам. Транзисторы работают противофазно, что обеспечивает малое выходное сопротивление при любом направлении переключения выходов.

Особенности логических выходов:

Их нельзя соединять параллельно по следующим причинам:

- это создает логическую неопределенность, т.к. в точке соединения выходов, формирующих «1», и выходов, формирующих «0» не будет разумного результата.
- При соединении выходов, находящихся в различных логических состояниях, возникает их противоборство, уравнивающий ток может достигнуть большой величины, что может вывести из строя элементы цепи.



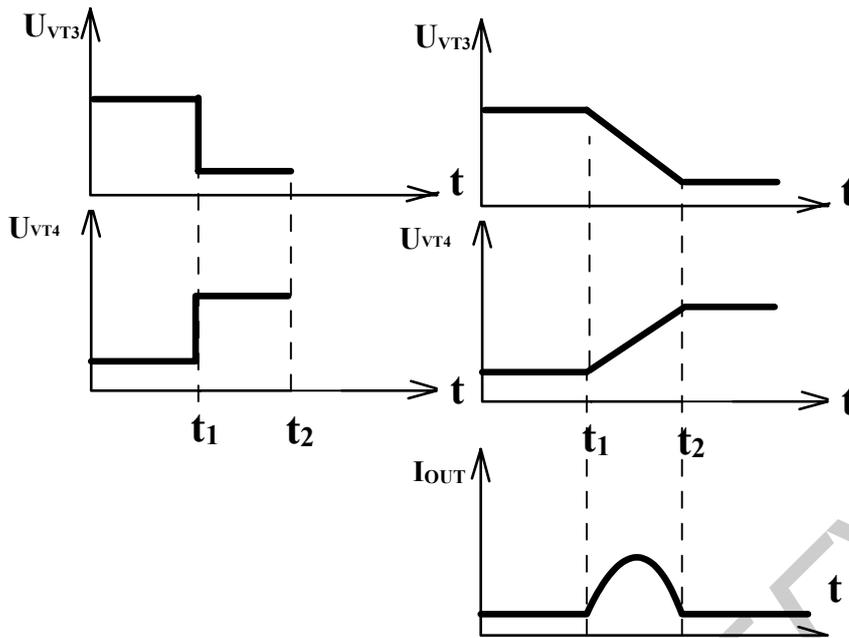


Рис. 1.7а. Идеализированные формы выходных сигналов на транзисторах VT3 и VT4

Рис. 1.7б. Реальные формы выходных сигналов на транзисторах VT3 и VT4 и временная диаграмма выходного тока

Элементы с тремя состояниями выхода

Параллельно элементы с тремя состояниями можно соединить при условии, что в любой момент времени активным может быть только один из них. В противном случае элементы могут выйти из строя, а информация теряется.

Выход с открытым коллектором.

Недостаток: большая задержка переключения из «0» в «1». При таком переключении происходит заряд емкости с малым током резистора. Сопротивление нельзя делать слишком малым, т.к. это приведёт к большим токам насыщения транзистора.

1.14. Режим неиспользуемых входов цифровых микросхем.

Типовым является случай наличия у элементов лишних входов, неиспользуемых входов, наличия в корпусах интегральных схем лишних элементов, нехватка у имеющихся элементов необходимого числа входов или нагрузочных способностей. Вопрос об использовании неиспользуемых входов решается по-разному для конкретных типов логик.

Рассмотрим пример:

Необходимо получить $F = X1 \cdot X2 \cdot X3 \cdot X4 \cdot X5$. При этом элемента 5 И-НЕ нет в наличии, зато имеется элемент 8 И-НЕ. Возможные варианты:

1. Не обращать внимания на "лишние" входы (т.е. оставить их разомкнутыми) (рис.1.8а).
2. Подсоединить их к задействованным входам (рис. 1.8в).
3. Подать на них некие константы (рис. 1.8б).

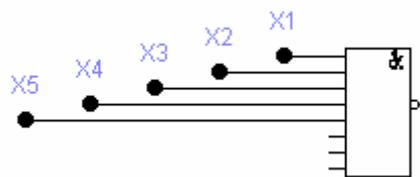


Рис. 1.8а. Режим неиспользуемых входов цифровых микросхем. "Лишние" входы

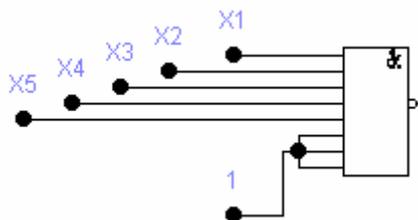


Рис. 1.8б. Режим неиспользуемых входов цифровых микросхем. Подсоединение к константам.

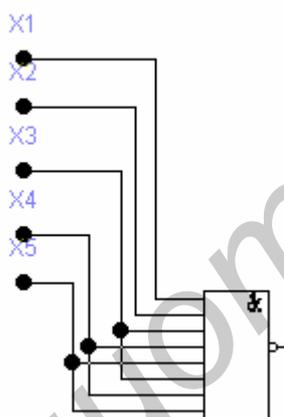


Рис. 1.8в. Режим неиспользуемых входов цифровых микросхем. Подсоединение к задействованным входам.

С точки зрения логических операций все три возможности правомерны. Но для разных схмотехнологий выбор соответствующего варианта является определяющим.

Для ЭСЛ:

Можно по 1-му варианту.

Это объясняется тем, что в схемах самих элементов уже предусмотрены резисторы, связанные с ИП, которые обеспечивают необходимые условия "лишним" входам.

Для КМОП:

По 1-му - нельзя! У КМОП ИС большие входные сопротивления, поэтому на разомкнутых входах легко наводятся паразитные потенциалы, которые могут изменить работу схемы.

Для ТТЛШ:

Строгого запрета на оставление разомкнутых входов нет, но делать это незачем, т.к. пострадает быстродействие.

Для КМОП и ТТЛШ 2-ой вариант принципиально возможен, но нежелателен в связи с тем, что это приводит к увеличению нагрузки на источник сигнала, что сопровождается снижением быстродействия источника.

Т.о. наиболее рациональный 3-ий вариант:

Для КМОП логики:



Рис. 1.9а. Вариант решения проблемы лишних входов в КМОП логике.

Для ТТЛШ схем:

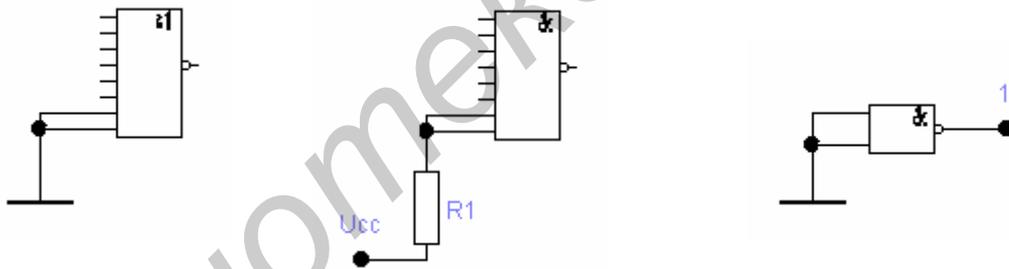


Рис. 1.9б. Вариант решения проблемы лишних входов в ТТЛШ логике.

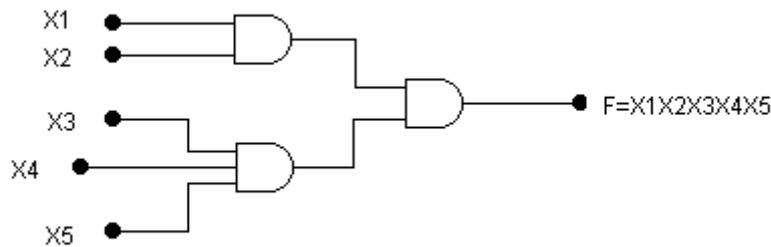
1.15. Режим неиспользуемых элементов в цифровых микросхемах. Наращивание числа входов.

Режим неиспользуемых элементов.

Если мощности, потребляемые неиспользуемыми элементами в состоянии «1» и «0» не равны, то элементы переводят в состояние потребления наименьшей мощности.

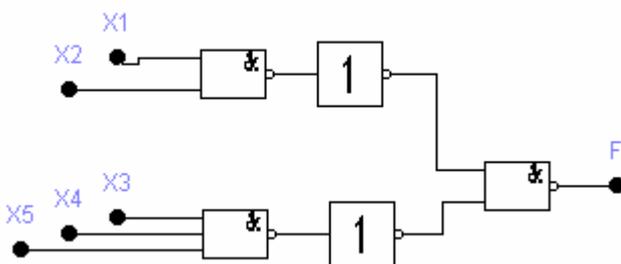
Наращивание числа входов.

Для конъюнкторов задача решается просто:



То же самое для дизъюнкции.

Если надо нарастить число входов для элементов И-НЕ или ИЛИ-НЕ то поступают следующим образом:



$$F = \overline{X1 \cdot X2 \cdot X3 \cdot X4 \cdot X5}$$

2. Функциональные узлы комбинационного типа.

Цифровым функциональным узлом называется устройство, предназначенное для выполнения определённых действий с двоичными переменными: хранение, сложение, счет, преобразование и т.д. Различают функциональные устройства комбинационного и последовательного типов.

Комбинационными называются функциональные узлы, выходные сигналы которых определяются комбинацией логических сигналов на входах, действующих в данный момент времени.

Комбинационные цифровые устройства (КЦУ) имеют в своём составе только набор логических вентилей и не обладают памятью. К ним относятся: шифраторы, дешифраторы, мультиплексоры, демультиплексоры, сумматоры, преобразователи кодов и т.д.

2.1. Шифратор. Построение на логических элементах.

Шифратор – функциональный узел, предназначенный для преобразования поступающих на его входы управляющих сигналов (команд) в n -разрядный двоичный код. В частности, десятичные числа с помощью шифратора преобразуются в двоичный код. При подаче сигнала на один из входов на выходе появляется двоичный код, численно равный номеру возбуждённого входа. Шифратор осуществляет "сжатие" информации, т.к. зашифрованная информация передаётся по меньшему числу выходных линий связи.

Шифратор, у которого при $m=2^n$ входах имеется n выходов, называется *полным*. Если $m < 2^n$, шифратор называется *неполным*.

Функционирование неполного шифратора описывается следующей таблицей:

	A3	A2	A1	A0
X0	0	0	0	0
X1	0	0	0	1
X2	0	0	1	0
X3	0	0	1	1
X4	0	1	0	0
X5	0	1	0	1
X6	0	1	1	0
X7	0	1	1	1
X8	1	0	0	0
X9	1	0	0	1

На основании этой таблицы выражения для каждого выхода имеют вид:

$$A0 = X1 \vee X3 \vee X7 \vee X5 \vee X9;$$

$$A1 = X2 \vee X3 \vee X6 \vee X7;$$

$$A2 = X4 \vee X5 \vee X6 \vee X7;$$

$$A3 = X8 \vee X9.$$

Для построения шифратора на логических элементах необходимо следующие логические вентили: 5ИЛИ, 4ИЛИ, 4ИЛИ, 2ИЛИ.

Построим неполный двоичный шифратор:

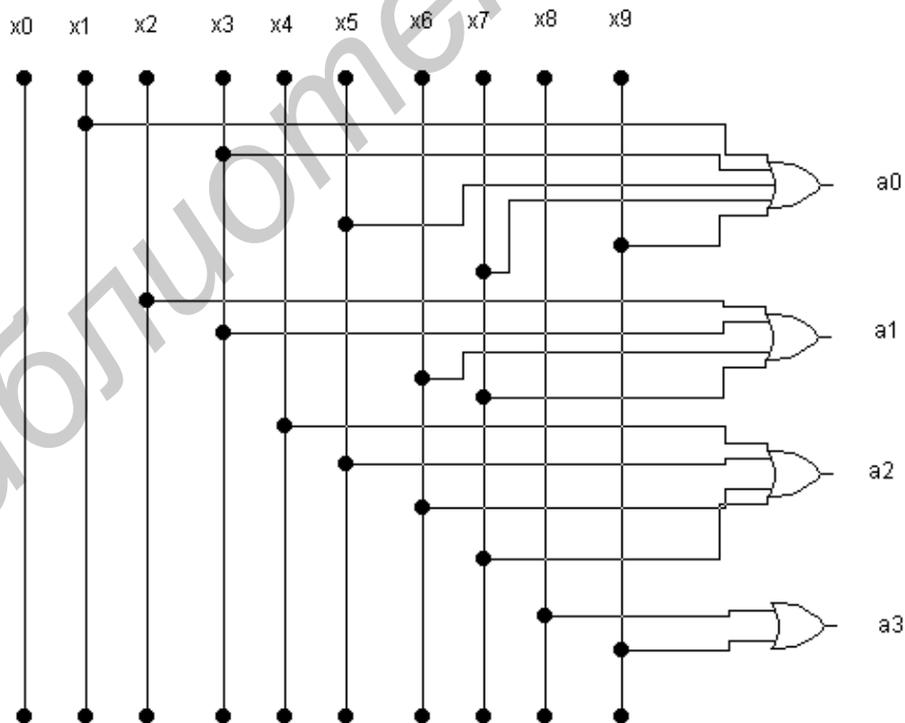


Рис. 2.1. Схема неполного двоичного шифратора на логических элементах

2.2. Дешифратор. Построение на логических элементах.

Дешифратор – функциональный узел, предназначенный для преобразования n -разрядного двоичного кода в комбинацию управляющих выходных сигналов. Если число выходов $m < 2^n$ – то это неполный дешифратор, если же $m = 2^n$ – то это полный дешифратор.

Условное обозначение: DC(decoder)

Правило его работы определяются таблицей:

A3	A2	A1	A0	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
...
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Строим карту Карно:

A2A3	A0A1			
	00	10	11	01
00	X0	X1	X3	X2
11	1	1	1	1
01	X8	X9	1	1

В соответствии с таблицей истинности функция, реализуемая по каждому выходу, имеет вид:

$$X_0 = \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot \overline{A_3};$$

$$X_1 = \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot A_3;$$

$$X_2 = \overline{A_0} \cdot \overline{A_1} \cdot A_2;$$

$$X_3 = \overline{A_0} \cdot A_1 \cdot \overline{A_2};$$

$$X_4 = \overline{A_0} \cdot A_1 \cdot A_2;$$

$$X_5 = A_0 \cdot \overline{A_1} \cdot \overline{A_2};$$

$$X_6 = A_0 \cdot \overline{A_1} \cdot A_2;$$

$$X_7 = A_0 \cdot A_1 \cdot \overline{A_2};$$

$$X_8 = \overline{A_0} \cdot A_3;$$

$$X_9 = A_0 \cdot A_3.$$

Таким образом, в состав данного дешифратора входят 10 схем И и 4 схемы НЕ. Отметим, что при любой комбинации сигналов на входах сигнал логической 1 наблюдается только на одном из выходов.

Схема будет выглядеть следующим образом:

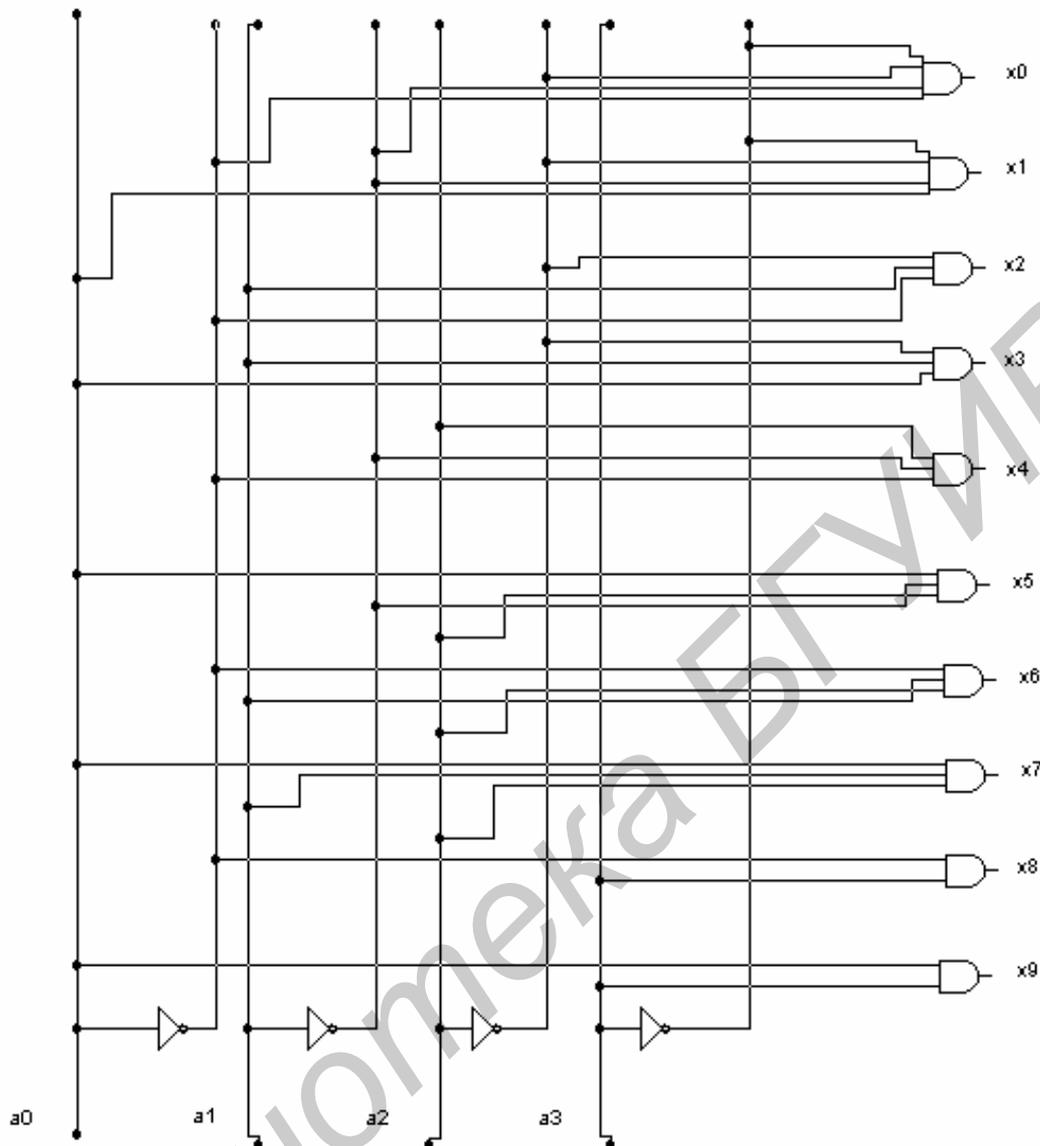


Рис. 2.2. Схема двоичного дешифратора на логических элементах.

2.3. Построение приоритетного двоичного шифратора.

Приоритетные шифраторы выполняют более сложную операцию по сравнению с обычными двоичными шифраторами. Часто в схемах систем телекоммуникаций решается задача определения приоритетного претендента на пользование каким-либо ресурсом или каналом связи. Несколько конкурентов выставляют свои запросы на обслуживание, которые не могут быть удовлетворены одновременно. Простейший вариант решения указанной задачи – присвоение каждому источнику запросов фиксированного приоритета (т.е. выстраивание очереди). Запросы (Request) обозначаются через $R_0 \dots R_7$. Приоритет будет у R_7 , а дальше приоритет будет уменьшаться. Самый младший – R_0 . Приоритетный шифратор вырабатывает на выходе двоичный номер старшего запроса. Когда имеется 1 возбужденный вход, приоритетный шифратор тождественно равен двоичному неперитетному шифратору.

EI (enable input) – сигнал разрешения работы данного шифратора. EI=1 – работа шифратора разрешена. EI=0 – независимо от состояний входов запросов все выходные сигналы шифратора равны нулю.

EO (enable output) – выходной сигнал разрешения, свидетельствующий об отсутствии возбужденных входов при включенном состоянии шифратора.

G – сигнал, свидетельствующий о наличии хотя бы одного возбужденного входа при включенном состоянии шифратора.

R₀...R₇ – запросы.

A₀..A₂ – значения разрядов выходного двоичного кода.

Функционирование приоритетного шифратора описывается таблицей:

EI	R ₇	R ₆	R ₅	R ₄	R ₃	R ₂	R ₁	R ₀	A ₂	A ₁	A ₀	G	EO
1	1	*	*	*	*	*	*	*	1	1	1	1	0
1	0	1	*	*	*	*	*	*	1	1	0	1	0
1	0	0	1	*	*	*	*	*	1	0	1	1	0
1	0	0	0	1	*	*	*	*	1	0	0	1	0
1	0	0	0	0	1	*	*	*	0	1	1	1	0
1	0	0	0	0	0	1	*	*	0	1	0	1	0
1	0	0	0	0	0	0	1	*	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	*	*	*	*	*	*	*	*	0	0	0	0	0

На основе этой таблицы выражения для выходов имеют вид:

$$A_2 = (R_7 \vee R_6 \cdot \overline{R_7} \vee \overline{R_7} \cdot \overline{R_6} \cdot R_5 \vee \overline{R_7} \cdot \overline{R_6} \cdot \overline{R_5} \cdot R_4) \cdot EI;$$

$$A_1 = (R_7 \vee \overline{R_7} \cdot R_6 \vee \overline{R_7} \cdot \overline{R_6} \cdot R_5 \cdot \overline{R_4} \cdot R_3 \vee \overline{R_7} \cdot \overline{R_6} \cdot \overline{R_5} \cdot \overline{R_4} \cdot \overline{R_3} \cdot R_2) \cdot EI;$$

$$A_0 = (R_7 \vee \overline{R_7} \cdot \overline{R_6} \cdot R_5 \vee \overline{R_7} \cdot \overline{R_6} \cdot \overline{R_5} \cdot R_4 \cdot R_3 \vee \overline{R_7} \cdot \overline{R_6} \cdot \overline{R_5} \cdot \overline{R_4} \cdot \overline{R_3} \cdot \overline{R_2} \cdot R_1) EI;$$

$$G = (R_7 \vee R_6 \vee R_5 \vee R_4 \vee R_3 \vee R_2 \vee R_1 \vee R_0) EI;$$

Применяя свойство $[a \vee f \cdot \overline{a} = a \vee f]$, получаем уравнения, определяющие внутреннюю структуру шифратора приоритета в его основной части:

$$A_2 = R_7 \vee R_6 \vee \overline{R_5} \vee \overline{R_4};$$

$$A_1 = R_7 \vee R_6 \vee \overline{R_5} \cdot \overline{R_4} \cdot R_2 \vee \overline{R_5} \cdot \overline{R_4} \cdot R_3;$$

$$A_0 = R_7 \vee \overline{R_6} \cdot R_5 \vee \overline{R_6} \cdot \overline{R_4} \cdot \overline{R_3} \vee \overline{R_6} \cdot \overline{R_4} \cdot \overline{R_2} \cdot R_1.$$

2.4. Нарращивание размерности приоритетного шифратора.

Разберём, как увеличить размерность приоритетного шифратора вдвое (от 8 до 16). На схеме нижний шифратор – старший по приоритету, его работа всегда разрешена подачей логического 0 на вход EI. Если на входах старшего шифратора есть хотя бы один запрос, то разрешения на работу младшего шифратора нет (EO=1). Выходы младшего шифратора пассивны, т.е. имеют единичные значения. При этом элементы И-НЕ, с которых снимаются сигналы A₀, A₁, A₂ играют роль инверторов для выходных сигналов старшего шифратора. Поэтому на выходах A₀, A₁, A₂ схемы в целом формируются сигналы от 0 до 7 в зависимости от номера старшего запроса в старшем шифраторе, что вместе с единицей на выходе A₃ даёт номера от 8 до 15.

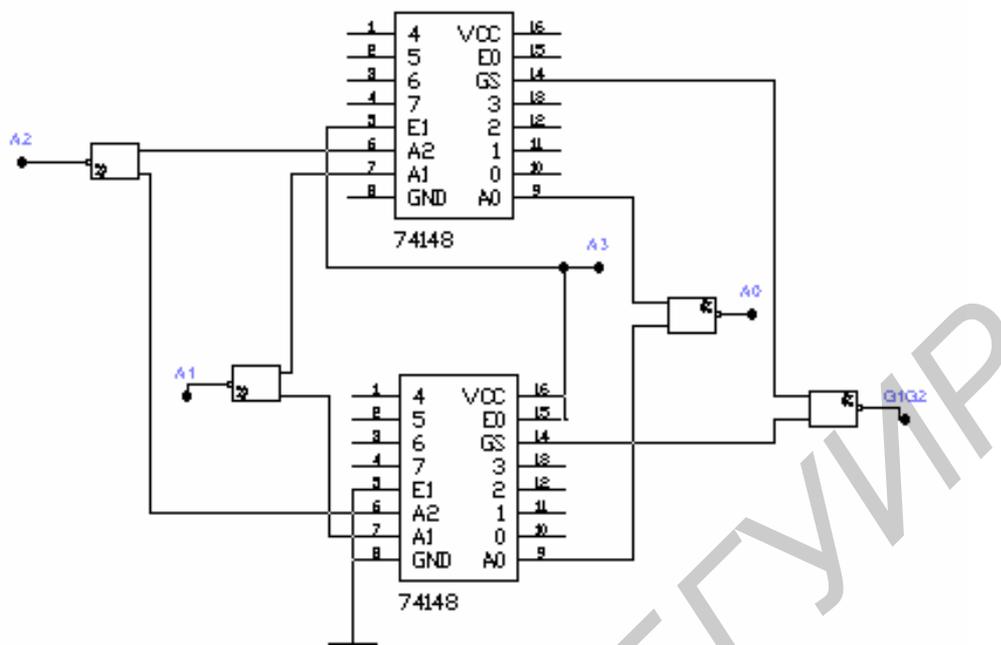


Рис. 2.3. Схема наращивания размерности приоритетного шифратора

Если на входах старшего шифратора запросов нет, он разрешает работу младшего (на схеме верхнего) шифратора, вырабатывая сигнал $EO=0$ и приводит свои выходы $A0, A1, A2$ в пассивное единичное состояние. Теперь на выходы A_i схемы в целом передаются значения выходов младшего шифратора, что вместе с нулём в разряде $A3$ соответствует номерам от 0 до 7.

Таким образом, получаем схему с 16 входами запроса с приоритетом. Выход $G1G2$ принимает значение лог.1 при наличии хотя бы одного запроса в любом из шифраторов и может использоваться как сигнал для управления последующей схемой.

2.5. Наращивание размерности дешифратора.

Из малоразрядных дешифраторов можно построить схему, эквивалентную схеме дешифратора большей разрядности. Для этого входные слова делятся на поля. Разрядность поля младших разрядов соответствует числу входов имеющегося дешифратора. Оставшиеся поля старших разрядов служат для получения сигналов разрешения работы одного из дешифраторов, декодирующих поле младших разрядов. Работу схемы рассмотрим на примере числа 11001. Это число 25 в десятичной системе счисления. На входе дешифратора первого яруса имеется код 11, его выход №3, что разрешает работу четвертого дешифратора 2 яруса. На входе дешифратора №4 действует код 001, поэтому единица появится на его первом выходе, т. е. на 25-ом выходе всей схемы. Разрешение работы дешифратора осуществляется подачей «1» на разрешающий вход дешифратора первого яруса.

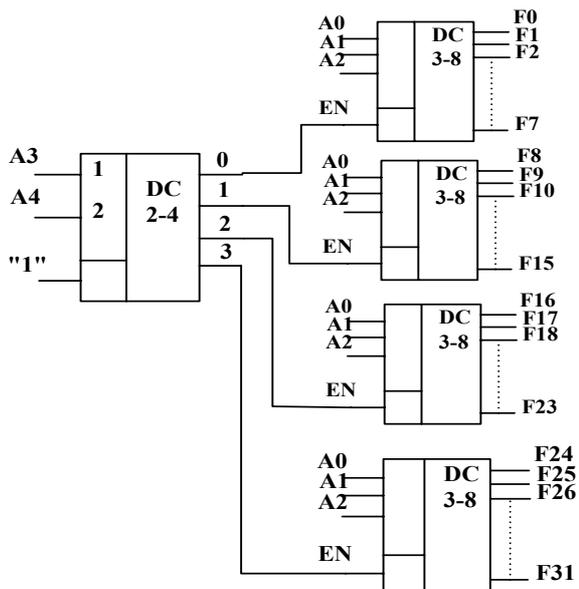


Рис. 2.4. Нарастивание размерности дешифратора

2.6. Воспроизведение произвольных логических функций с помощью дешифратора.

Дешифраторы со схемами ИЛИ можно использовать для воспроизведения произвольных логических функций. На выходах дешифратора вырабатываются все конъюнктивные термы, которые только можно составить из данного числа аргументов. Логическая функция в СДНФ есть дизъюнкция некоторого числа таких термов. Собирая нужные термы по схеме ИЛИ, можно получить любую функцию данного числа аргументов.

Разберём пример следующего соединения:

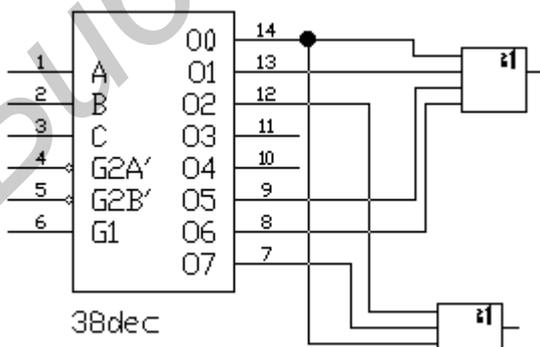


Рис. 2.5. Пример соединения дешифратора и логических вентилях ИЛИ для получения произвольных логических функций.

00: $\overline{X_3} \cdot \overline{X_2} \cdot \overline{X_1}$;
 01: $\overline{X_3} \cdot \overline{X_2} \cdot X_1$;
 02: $\overline{X_3} \cdot X_2 \cdot \overline{X_1}$;
 03: $\overline{X_3} \cdot X_2 \cdot X_1$;
 04: $X_3 \cdot \overline{X_2} \cdot \overline{X_1}$;
 05: $X_3 \cdot \overline{X_2} \cdot X_1$;
 06: $X_3 \cdot X_2 \cdot \overline{X_1}$;
 07: $X_3 \cdot X_2 \cdot X_1$.

$$F_1 = \overline{X_3} \cdot \overline{X_2} \vee X_3 \cdot X_1;$$

$$F_2 = \overline{X_3} \cdot X_2 \cdot X_1 \vee X_2 \cdot \overline{X_1}.$$

Подобное решение может быть целесообразно при необходимости выработки нескольких функций одних и тех же переменных.

2.7. Мультиплексор.

Мультиплексор – это функциональный узел, который осуществляет управляемую коммутацию информации, поступающей по N параллельным каналам, на одну выходную линию. Коммутация есть последовательный опрос и передача информации. Коммутация определённой входной линии происходит в соответствии с двоичным адресным кодом $A_{n-1}A_{n-2} \dots A_1A_0$. Входы $A_0 \dots A_{n-1}$ являются адресными. Их значение определяет одну из переменных X_i , передаваемых на выход F. Если адресный код имеет n-разрядов, то можно осуществить $N=2^n$ комбинаций адресных сигналов, каждый из которых обеспечивает коммутацию информации, поступающей по одному из N входов на выход. В простейшем случае двухразрядового адресного кода $n=2$ и максимальное число входных линий $N=4$. Таблица истинности такого мультиплексора 4-1 (Мух 4-1) выглядит следующим образом:

A_0	A_1	F
0	0	X_0
0	1	X_1
1	0	X_2
1	1	X_3

Исходя из таблицы получаем характеристическое уравнение такого мультиплексора:

$$F = \overline{A_0} \cdot \overline{A_1} \cdot X_0 \vee \overline{A_0} \cdot A_1 \cdot X_1 \vee A_0 \cdot \overline{A_1} \cdot X_2 \vee A_0 \cdot A_1 \cdot X_3.$$

По этому выражению построим мультиплексор на логических элементах. Из полученного выражения следует, что в состав структурной схемы такого мультиплексора входят два инвертора, четыре схемы И и одна ИЛИ.

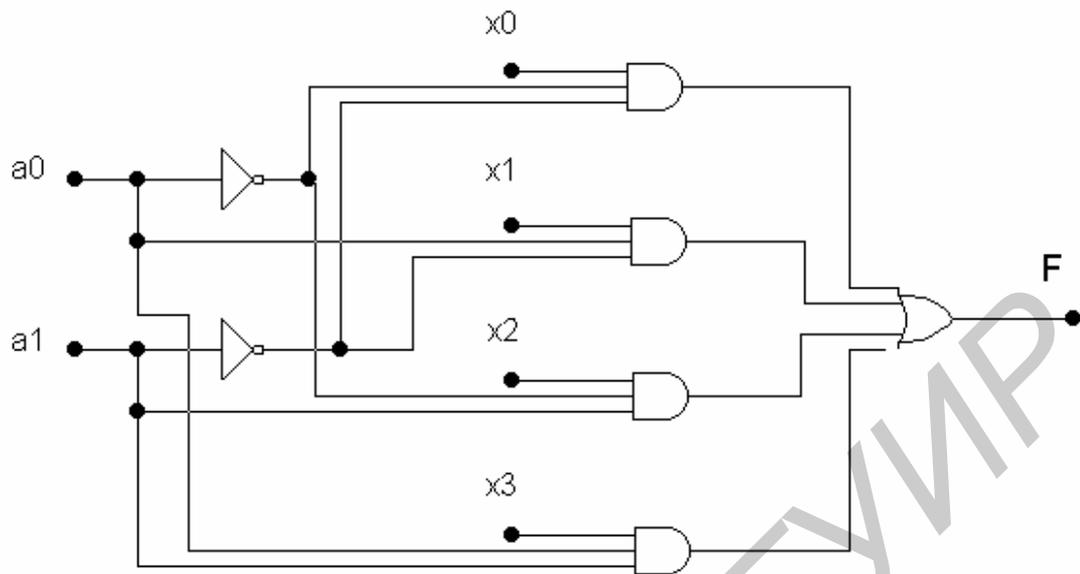


Рис. 2.6. Схема мультиплексора 4-1 на логических элементах.

Условное обозначение:

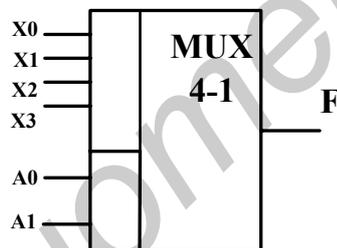


Рис. 2.7. Условное обозначение мультиплексора 4-1.

2.8. Демультимплексор.

Демультимплексор – функциональный узел, осуществляющий управляемую коммутацию информации, поступающей по одному каналу на N выходов. В общем случае число выходов линий N определяется количеством адресных входов n, т.е. $N=2^n$.

Для $n=2$ функционирование демультимплексора осуществляется в соответствии с таблицей:

A0	A1	X0	X1	X2	X3
0	0	F	0	0	0
0	1	0	F	0	0
1	0	0	0	F	0
1	1	0	0	0	F

Из таблицы следует, что информация F, в зависимости от адресных входов, направляется в разные выходные линии X_i . При этом на остальные линии

информация не поступает. Характеристические уравнения демультиплексора в соответствии с таблицей истинности будут:

$$X_0 = \overline{A_0} \cdot \overline{A_1} \cdot F;$$

$$X_1 = \overline{A_0} \cdot A_1 \cdot F;$$

$$X_2 = A_0 \cdot \overline{A_1} \cdot F;$$

$$X_3 = A_0 \cdot A_1 \cdot F.$$

Соответствующая этим функциям структурная схема выглядит следующим образом:

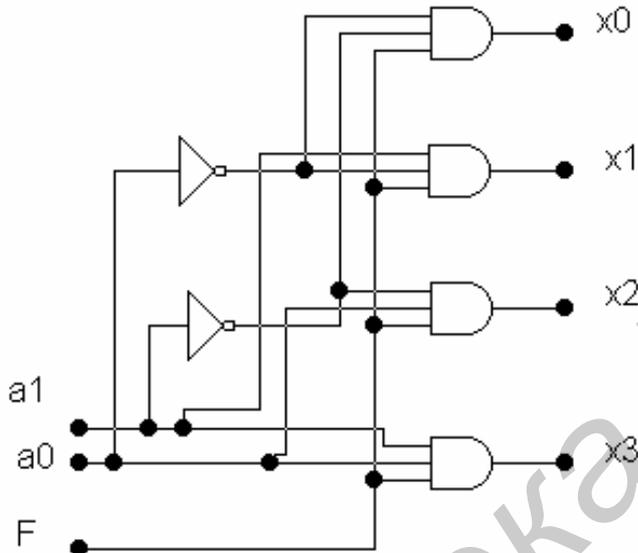


Рис. 2.8. Структурная схема демультиплексора 1-4.

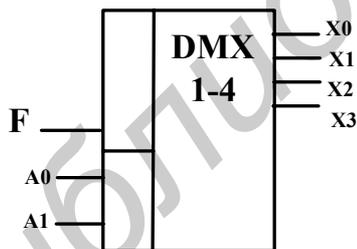


Рис. 2.9. Условное обозначение демультиплексора 1-4.

Нетрудно заметить, что дешифратор со входом разрешения (E) будет работать в режиме демультиплексора, если на вход (E) подавать информационный сигнал. Действительно, при подаче лог.1 на вход E адресация дешифратора приведёт к возбуждению соответствующего выхода, а при E=0 – нет. А это и есть передача информационного сигнала в адресованный выходной канал. Поэтому в ряде серий элементов отдельные DMX отсутствуют, а DC со входом разрешения часто называют дешифратором – демультиплексором. Обозначается: DC-DMX (Decoder-Demultiplexor).

2.9. Нарращивание размерности мультиплексора.

В стандартных сериях размерности мультиплексоров не превышают 16-1 (серия КП1). Нарращивание размерности мультиплексора возможно с помощью пирамидальной структуры из нескольких мультиплексоров. Первый ярус схемы представляет столбец, содержащий столько мультиплексоров, сколько необходимо для получения нужного числа информационных входов. Все мультиплексоры столбца адресуются одним и тем же кодом, составленным из соответствующего числа младших разрядов общего адресного кода. Старшие разряды адресного кода используются во втором ярусе, мультиплексор которого обеспечивает поочерёдную работу мультиплексоров первого яруса на общий выходной канал.

Покажем пирамидальную схему, выполняющую функции MUX 32-1 и построенную на MUX 8-1. Младшими разрядами в адресе $a_4a_3a_2a_1a_0$ будут разряды $a_2a_1a_0$.

Построение мультиплексора 32-1 на основе мультиплексоров 8-1:

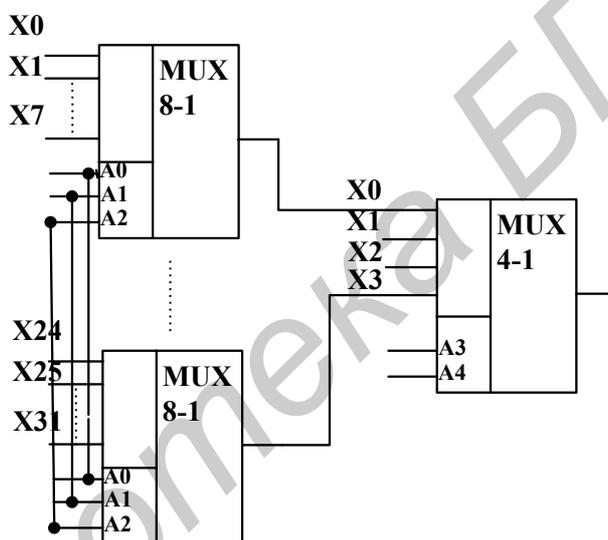


Рис. 2.10. Пример построения мультиплексора 32-1 из мультиплексоров 8-1.

2.10. Универсальные логические модули (УЛМ) на основе мультиплексоров. Настройка константами. Расширение алфавита сигналов настройки.

УЛМ на основе мультиплексоров относятся к устройствам, настраиваемым на решение той или иной задачи. Универсальность их заключается в том, что для заданного числа аргументов можно настроить УЛМ на любую функцию. Известно, что общее число функций n -аргументов выражается, как 2 в степени 2^n .

Первый способ настройки – настройка константами (фиксаций входов).

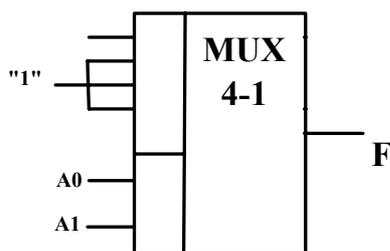


Рис. 2.11а. Пример настройки УЛМ на основе мультиплексора константами.

A0	A1	F
0	0	0
0	1	1
1	0	1
1	1	1

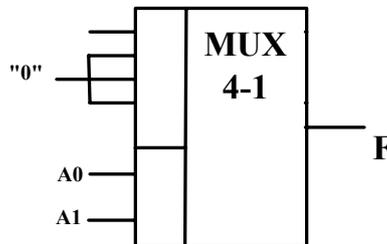


Рис. 2.116. Пример настройки УЛМ на основе мультиплексора константами.

A0	A1	F
0	0	1
0	1	0
1	0	0
1	1	0

Для этого способа справедливы следующие соотношения между числом аргументов и числом настроечных входов. Пусть число аргументов n , и требуется настройка на любую функцию. Тогда число комбинаций для кода настройки, равное числу функций, равно 2 в степени 2^n . Для двоичного кода число комбинаций связано с разрядностью кода как 2^m , где m – разрядность кода. Приравняв число воспроизводимых функций к числу комбинаций кода настройки, имеем для числа настроечных кодов выражение: $m=2^n$. Этому выражению отвечает соотношение между числом входов разного назначения для мультиплексора. При этом на адресные входы подаются аргументы функций, а на информационные входы – сигналы настройки.

$\{0,1\}$ – алфавит настройки.

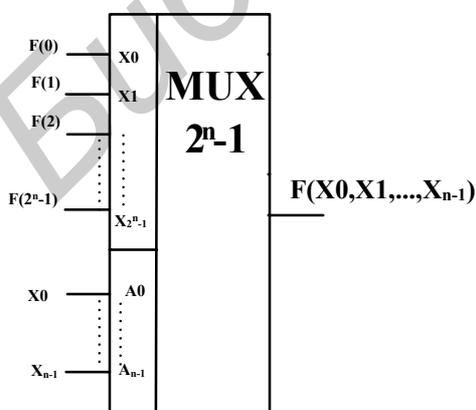


Рис. 2.12. Использование мультиплексора в качестве УЛМ

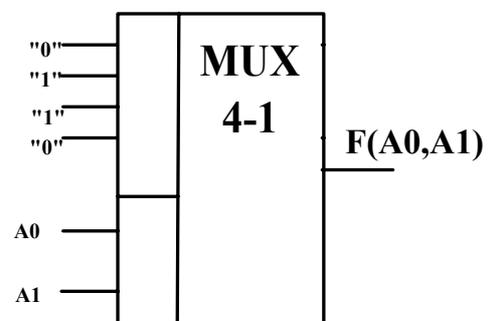


Рис. 2.13. Пример настройки УЛМ на функцию И2ЛИ

A0	A1	F
0	0	0
0	1	1
1	0	1
1	1	0

$F = \overline{A0} \cdot A1 \vee A0 \cdot \overline{A1} = A0 \oplus A1$, где \oplus - сумма по модулю два или функция неравнозначности, также называемая «исключающее ИЛИ».

Второй способ - расширение алфавита сигналов настройки - реализуется с целью уменьшения числа входов настройки. Если от алфавита $\{0,1\}$ перейти к алфавиту $\{0,1, \tilde{X}\}$, где \tilde{X} - литерал одного из аргументов, то число входов аргументов сократится на единицу, а число настроечных входов при этом сокращается в два раза.

Под *литералом* переменной понимается либо сама переменная, либо её инверсия. Перенос одного из аргументов в число сигналов настройки не влечёт за собой каких-либо схемных изменений. Для нового алфавита код настройки находится следующим образом: аргументы, за исключением литерала, подаются на адресные входы, что соответствует их фиксации в выражении для искомой функции, которая становится функцией одного аргумента \tilde{X} . Такая функция называется остаточной и её нужно подавать на настроечные входы.

$$F = X1 \cdot X2 \cdot X3 \vee \overline{X2} \cdot \overline{X3};$$

В сигнал настройки переносится тот аргумент, который меньше всего входит в термы. Для этой функции алфавит настройки выглядит следующим образом: $\{0,1, X1\}$.

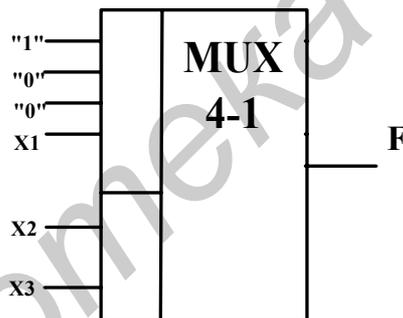


Рис. 2.14. Пример настройки УЛМ на основе мультиплексора методом расширения алфавита сигналов настройки.

X ₂	X ₃	F
0	0	1
0	1	0
1	0	0
1	1	X ₁

По пути расширения алфавита можно идти и дальше, но при этом понадобятся дополнительные логические схемы (вентили), воспроизводящие остаточные функции, которые будут зависеть больше, чем от одного аргумента. Если в сигналы настройки перевести два аргумента, то дополнительные логические схемы будут двухвходовыми вентилями, что мало усложнит схему.

$$F = \overline{X1} \cdot X2 \vee X3 \cdot \overline{X4},$$

$\{0,1, \tilde{X1}, \tilde{X2}\}$.

X ₄	X ₃	F _{ост}
0	0	X ₁ X ₂
0	1	1
1	0	X ₁ X ₂
1	1	X ₁ X ₂

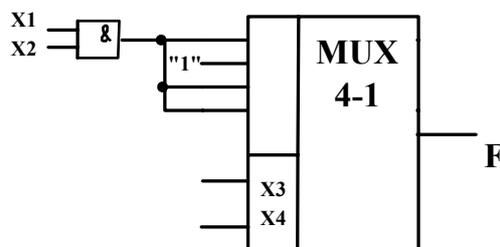


Рис. 2.15. Использование дополнительных логических вентилей на входах УЛМ на основе мультиплексора при расширении алфавита сигналов настройки.

2.11. Полусумматор. Синтез полного сумматора из полусумматоров.

Сумматор – функциональный узел, осуществляющий сложение двоичных чисел. Различают сумматор неполный (полусумматор – Half Adder) и полный (Full Adder). Простейшим является неполный одноразрядный сумматор.

Состояния такого сумматора:

A ₀	B ₀	S ₀	P ₁
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Здесь A₀ и B₀ – одноразрядные суммируемые числа, а S₀ и P₁ – значения суммы в данном разряде (частичная сумма) и перенос в следующий разряд, соответственно. Выражения для P₁ и S₀ получаются на основании таблицы:

$$P_1 = A_0 \cdot B_0;$$

$$S_0 = A_0 \oplus B_0.$$

Из чего следует, следует, что формирование переноса можно осуществить на элементе И, а частичной суммы – на элементе ИСКЛЮЧАЮЩЕЕ ИЛИ (И2ЛИ).

В полусумматоре не учитывается перенос из младшего разряда.

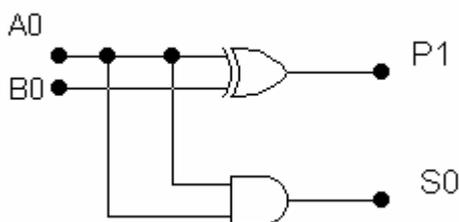


Рис. 2.16. Структурная схема полусумматора на логических элементах.

Условное изображение полусумматора:

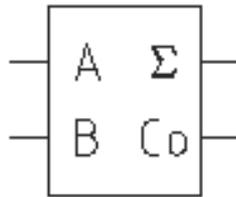


Рис. 2.17. Условное обозначение полусумматора

Если в сумматоре учитывается перенос из младшего разряда в старший, то это полный сумматор.

Операции сложения и вычитания начинаются с младших разрядов. При сложении двух единиц в одном разряде записывается 0 и образуется 1 переноса в следующий разряд, которая складывается с числами этого разряда.

$$\begin{array}{r} 10110 \\ +1001 \\ \hline 11111 \end{array}$$

Сумматор выполняет арифметическое сложение, в противовес логическому (операции дизъюнкции).

Полный сумматор может быть синтезирован на основе двух полусумматоров и логического вентиля ИЛИ:

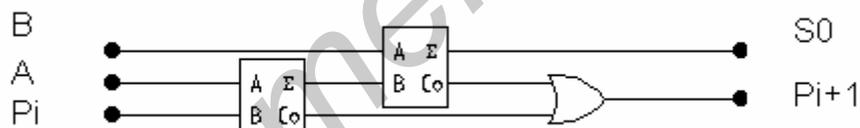


Рис. 2.18. Пример построения полного сумматора из двух полусумматоров.

2.12. Синтез полного сумматора на логических элементах.

В полусумматоре не учитывается перенос из младшего разряда. Он может применяться для сложения чисел только в нулевом разряде. Если в сумматоре учитывается перенос из младшего разряда в старший, то это полный сумматор. Он осуществляет арифметическое сложение одноразрядных двоичных чисел a_i и b_i и числа c_i переноса из младшего разряда, образуя на выходах значения суммы S_i и число переноса в старший разряд c_{i+1} . Такая операция осуществляется при сложении многоразрядных чисел.

Таблица истинности для построения сумматора:

A_i, B_i - складываемые числа;

S_i – результат арифметического сложения в данном разряде;

C_i – перенос из младшего разряда;

C_{i+1} – перенос в следующий разряд.

A_i	B_i	C_i	S_i	C_{i+1}
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

Построим карты Карно:

Для S_i :

L_i	$A_i B_i$			
	00	01	10	11
0	0	1	0	1
1	1	0	1	0

Для C_{i+1} :

C_i	$A_i B_i$			
	00	01	10	11
0	0	0	1	0
1	0	1	1	1

$$S_i = \overline{A_i} \cdot \overline{B_i} \cdot C_i \vee$$

$$\vee \overline{A_i} \cdot B_i \cdot \overline{C_i} \vee A_i \cdot \overline{B_i} \cdot \overline{C_i} \vee A_i \cdot B_i \cdot C_i = \overline{\overline{\overline{\overline{\overline{\overline{A_i \cdot B_i \cdot C_i}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{A_i \cdot B_i \cdot C_i}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{A_i \cdot B_i \cdot C_i}}}}}} \cdot \overline{\overline{\overline{\overline{\overline{\overline{A_i \cdot B_i \cdot C_i}}}}}} \\ C_{i+1} = A_i \cdot B_i \vee A_i \cdot C_i \vee \overline{B_i} \cdot C_i = \overline{\overline{\overline{\overline{\overline{\overline{A_i \cdot B_i \cdot A_i \cdot C_i \cdot B_i \cdot C_i}}}}}}.$$

Для построения полного сумматора необходимо 1 элемент 4 И-НЕ, 5 элементов 3 И-НЕ, 3 элемента 2И-НЕ и 3 инвертора, т.е. всего 12 вентилях.

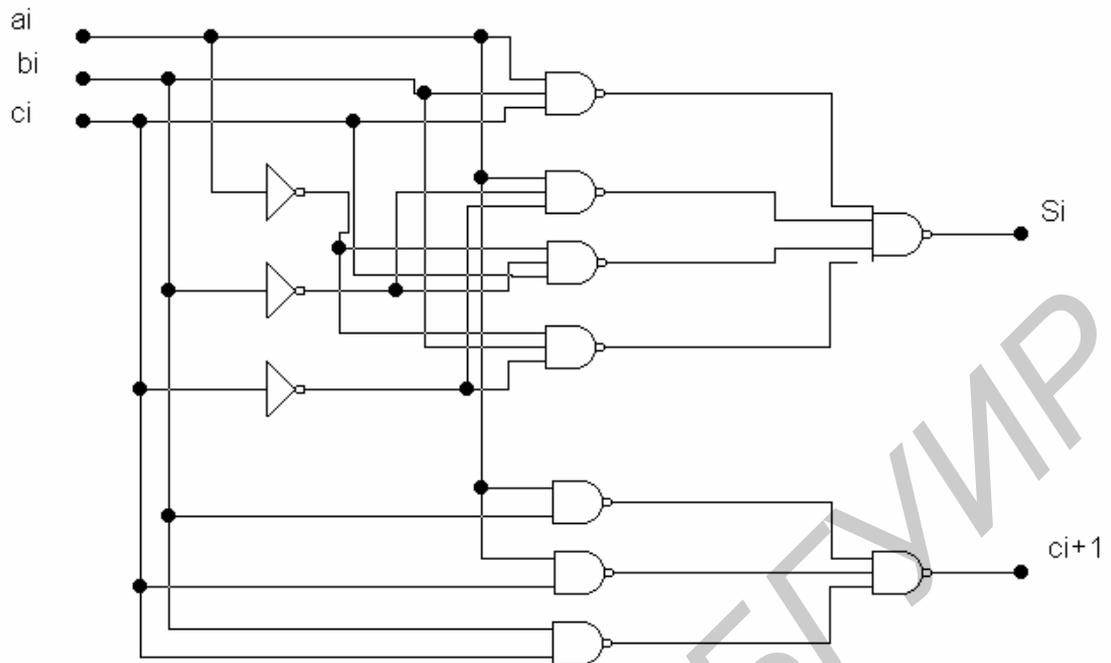


Рис. 2.19. Построение схемы полного сумматора на логических вентилях.

Однако такое построение нерационально. Из таблицы истинности для одноразрядного полного сумматора следует, что во всех строчках кроме первой и последней $S_i = \overline{C_{i+1}}$. Поэтому, при построении сумматора можно использовать уже полученное значение для C_{i+1} в качестве вспомогательного аргумента при вычислении S_i . Для S_i можно записать $S_i = \overline{C_{i+1}}(a_i \vee b_i \vee c_i) \vee a_i b_i c_i$. В этом случае схема получается проще.

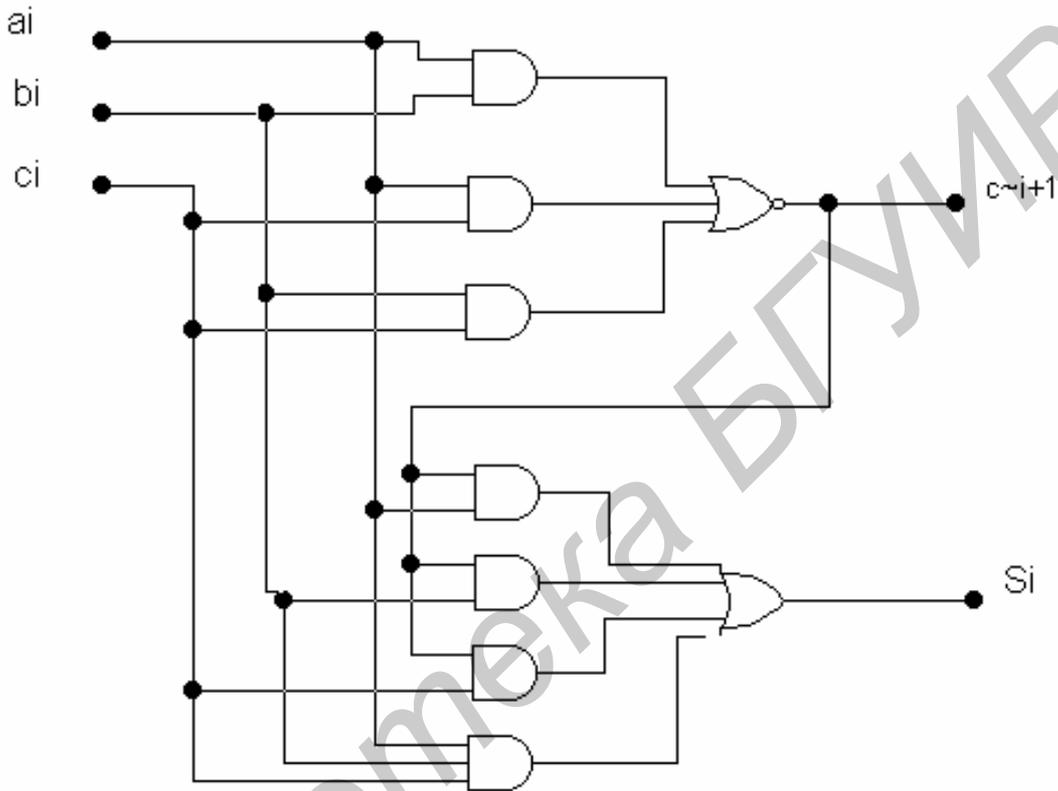


Рис. 2.20. Пример оптимизированного сумматора

Данное решение более популярно в производстве современных микросхем по сравнению с предыдущим, т.к. сокращается аппаратная сложность. В частности, имеется только 2 элемента на 3 входа и лишь 1 на 4 входа.

Из таблицы истинности видно также, что и функция суммы, и функция переноса обладают свойством самодвойственности: при инвертировании всех элементов инвертируется и значение функции.

2.13. Сумматор параллельного действия с последовательным переносом.

Из одноразрядного сумматора можно построить сумматор параллельного действия с последовательным переносом на любое количество разрядов.

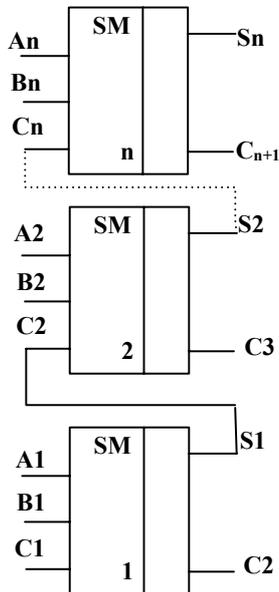


Рис. 2.21. Построение сумматора параллельного действия с последовательным переносом.

При подаче слагаемых цифры их разрядов поступают на соответствующий одноразрядный сумматор. Каждый из одноразрядных сумматоров формирует на своих выходах цифру соответствующего разряда суммы и перенос, передаваемый на вход одноразрядного сумматора, следующего, более старшего разряда.

2.14. Параллельный сумматор с параллельным переносом.

Данный сумматор разработан для получения максимального быстродействия, не имеет последовательного переноса. Во всех разрядах результаты вырабатываются

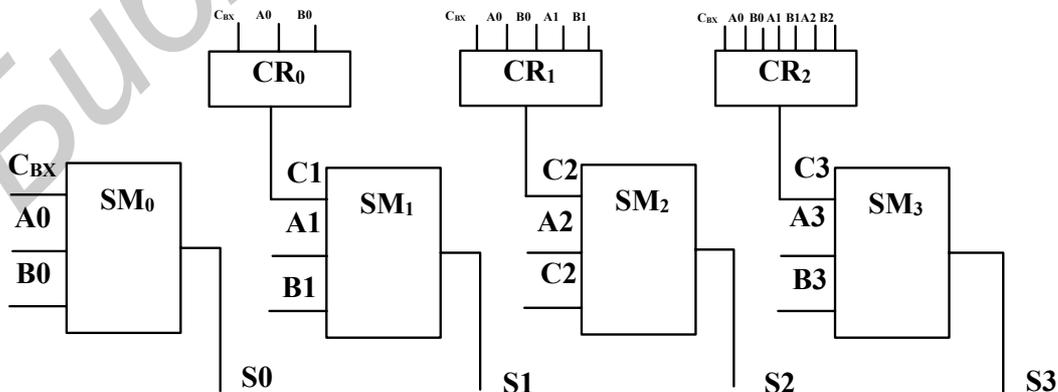


Рис. 2.22. Построение параллельного сумматора с параллельным переносом.

одновременно, параллельно во времени. Сигнал переноса для данного разряда формируется специальными схемами, на входы которых поступают все переменные, необходимые для выработки переноса. К этим величинам относятся внешний входной перенос (если он есть) и значения всех разрядов слагаемых, младших относительно данного.

Одноразрядные сумматоры для каждого разряда здесь упрощены, т.к. от них выход переноса не требуется, достаточно одного выхода сумматора.

2.15. Цифровые компараторы.

Компаратор – это устройство, предназначенное для сравнения двух чисел А и В, каждое из которых представлено в двоичной форме исчисления входными сигналами $\{a_0, a_1, \dots, a_m\}$ и $\{b_0, b_1, \dots, b_m\}$.

Простейший компаратор производит проверку равенства двух чисел. Два числа равны, если равны все соответствующие разряды этих двух чисел. Сравнение начинается со старшего разряда. Если два числа равны, то выход компаратора $y=1$; если нет, то $y=0$.

В соответствии с этим правилом переключательную функцию, реализуемую

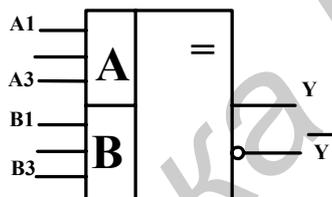


Рис. 2.23. Условное обозначение цифрового компаратора.

таким компаратором, можно представить выражением $y = \prod_{i=1}^m Z_i(A_i, B_i)$, где функция

Z_i принимает значения 1 при равенстве $A_i=B_i$, а 0 – при $A_i \neq B_i$. В дизъюнктивной нормальной форме функция $Z_i(A_i, B_i)$ имеет вид $z_i = A_i B_i \vee \overline{A_i} \overline{B_i}$. Эта функция носит название "инверсией операции ИСКЛЮЧАЮЩЕЕ ИЛИ" т.е. И2ЛИ-НЕ, или функцией равнозначности.

Функционирование компаратора описывается таблицей:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

$Z = A_i \cdot B_i \vee \overline{A_i} \cdot \overline{B_i} = \overline{A \oplus B}$ - функция равнозначности.

Построение компаратора на логических элементах:

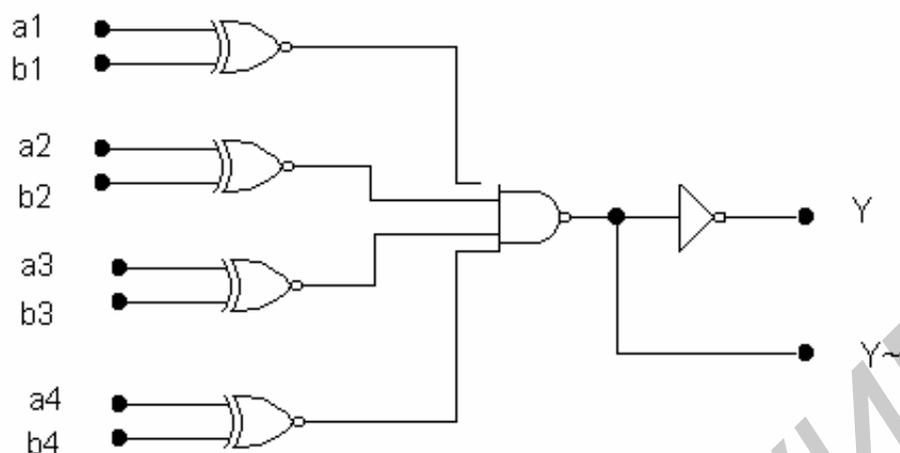


Рис. 2.24. Схема цифрового компаратора на логических элементах.

Более универсальны компараторы, которые помимо констатации равенства двух чисел могут установить, какое из них больше. Обычно такие компараторы имеют три выхода: $Y_{A=B}$, $Y_{A>B}$, $Y_{A<B}$. Простейшая задача состоит в сравнении двух одноразрядных чисел А и В. Такое сравнение реализуется переключательными функциями:

$$Y_{A=B} = AB \vee \bar{A}\bar{B},$$

$$Y_{A>B} = A\bar{B}$$

$$Y_{A<B} = \bar{A}B$$

Схема подобного компаратора на один разряд выглядит следующим образом:

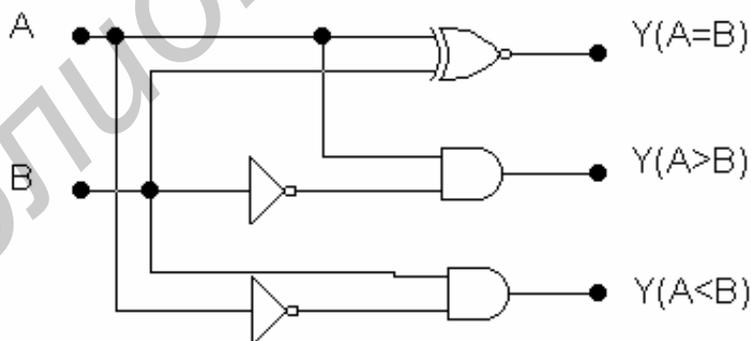


Рис. 2.25. Схема цифрового компаратора, определяющего, какое из чисел больше.

При сравнении многоразрядных чисел алгоритм такой: сначала сравнивают значения старших разрядов; если они различны, то эти разряды и определяют результат сравнения. Если же они равны, то необходимо сравнивать следующие за ним более младшие разряды и т.д.

2.16. Преобразователь двоично-десятичного кода в код Айкена.

Преобразователи кодов используются для шифрации и дешифрации цифровой информации.

8-4-2-1 – двоично-десятичный код;

2-4-2-1 – код Айкена (для чисел от 0 до 9, а не до 15).

Дес. число	8-4-2-1				2-4-2-1			
	A3	A2	A1	A0	L3	L2	L1	L0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1

В этой таблице каждую из переменных l_i можно рассматривать функцией коэффициентов a_3, a_2, a_1, a_0 двоично-десятичного кода и составить карты Карно для четырёх переменных. Итак, для каждой функции l_i ($i=0..3$) составим карты Карно:

Для l_3 :

A2A3	A0A1			
	00	10	11	01
00	0	0	0	0
10	0	1	1	1
11	*	*	*	*
01	1	1	*	*

*- факультативная единица (т.к. для 8-4-2-1 не закончились комбинации: неизвестно 0 или 1).

Объединять контур только из факультативных единиц (из *) нельзя. Необязательно, чтобы все * входили в контуры.

Для l_2 :

A2A3	A0A1			
	00	10	11	01
00	0	0	0	0
10	1	0	1	1
01	1	1	*	*

Для l_1 :

	00	10	11	01
00	0	0	1	1
10	0	1	0	0
11	*	*	*	*
01	1	1	*	*

Для l_0 :

	A0A1			
A2A3	00	10	11	01
00	0	1	1	0
10	0	1	1	0
11	*	*	*	*
01	0	1	*	*

На основании карт Карно выражения для l_i :

$$L3 = A3 \vee A2 \cdot A1 \vee A2 \cdot A0;$$

$$L2 = A3 \vee A2 \cdot A1 \vee A2 \cdot \overline{A0};$$

$$L1 = A3 \vee \overline{A2} \cdot A1 \vee A2 \cdot \overline{A1} \cdot A0;$$

$$L0 = A0.$$

Приведем к виду И-НЕ:

$$L3 = \overline{\overline{A3} \cdot \overline{A2} \cdot \overline{A1} \cdot \overline{A2} \cdot \overline{A0}};$$

$$L2 = \overline{\overline{A3} \cdot \overline{A2} \cdot \overline{A1} \cdot \overline{A2} \cdot \overline{A0}};$$

$$L1 = \overline{\overline{A3} \cdot \overline{A2} \cdot \overline{A1} \cdot \overline{A2} \cdot \overline{A1} \cdot \overline{A0}};$$

$$L0 = A0.$$

На основе этих выражений строится схема преобразователя.

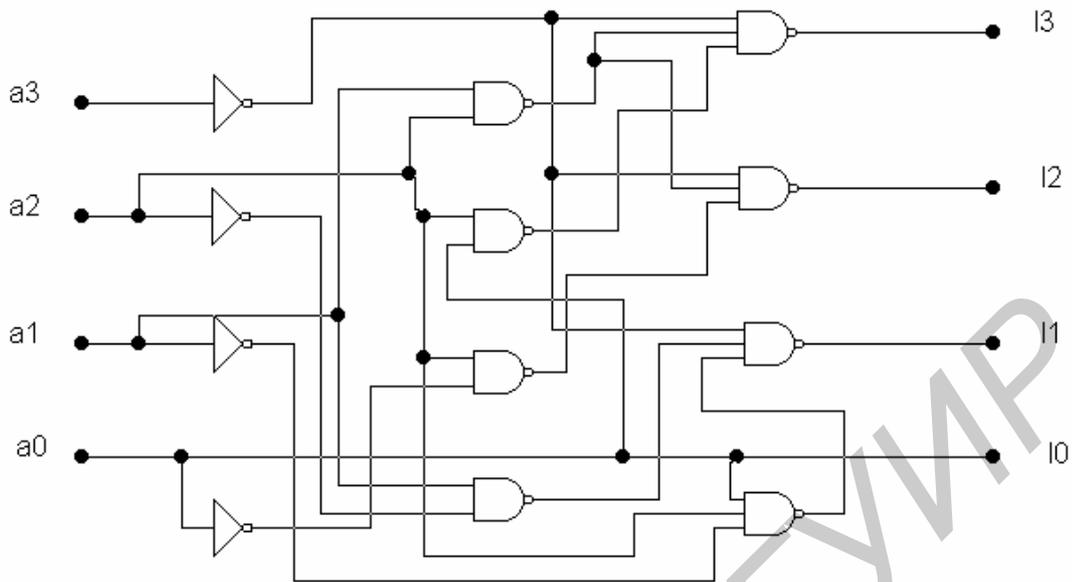
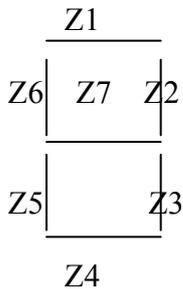


Рис. 2.26. Схема преобразователя кода 8-4-2-1 в код Айкена.

2.17. Преобразователь кода для семисегментной индикации.

Один из способов цифровой индикации состоит в следующем. Имеется семь сегментов.



Так выглядит индикатор. Изображение символов в индикаторе формируется высвечиванием необходимой комбинации сегментов. Смена изображений достигается путем соответствующей коммутации сегментов.

Десятичные цифры, отображение которых необходимо вызвать, задаются обычно в двоичном коде. Через $Y1..Y7$ обозначим управляющие импульсы. Если элемент светится, то он находится в состоянии «1», если нет – «0».

Однако управление цифровым индикатором осуществляется, как правило, по несколько иному алгоритму. Управление осуществляется следующим образом: уровень логической единицы на входе индикатора вызывает его погашение. Например, чтобы увидеть цифру «6» необходимо: $Y2=1, Z2=0$; т.е. $y = \bar{z}$.

Дес. цифра	8-4-2-1				$\overline{Z1}$	$\overline{Z2}$	$\overline{Z3}$	$\overline{Z4}$	$\overline{Z5}$	$\overline{Z6}$	$\overline{Z7}$
	A3	A2	A1	A0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

Составим карты Карно для функций Y1..Y7:

Для y_1 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	0
11	0	0	0	1
01	1	0	0	0

Для y_2 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	0
11	1	0	0	0
01	0	1	0	0

Для y_3 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	0
11	0	0	0	0
01	0	0	1	0

Для y_4 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	0
11	0	1	0	1
01	1	0	0	0

Для y_5 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	1
11	1	1	1	1
01	1	0	0	0

Для y_6 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
10	*	*	*	0
11	0	1	1	1
01	0	0	1	0

Для y_7 :

X4X1	X3X2			
	10	11	10	00
00	*	*	*	0
01	0	0	0	1

Из карт Карно выражения для Y_i :

$$y_1 = \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} = \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4}}}}$$

$$y_2 = \overline{x_1} \cdot x_2 \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot x_3 = \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4}}}}$$

$$y_3 = \overline{x_1} \cdot x_2 \cdot x_3;$$

$$y_4 = \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee x_1 \cdot x_2 \cdot x_3 = \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_1 \cdot x_2 \cdot x_3}}}}}}}}}}$$

$$y_5 = \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee x_1 = \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_1}}}}$$

$$y_6 = x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4}}}} \vee \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2}}}} = \overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_1 \cdot x_2}}}}}}}}}}$$

$$y_7 = x_1 \cdot x_2 \cdot x_3 \vee \overline{\overline{\overline{\overline{\overline{x_2 \cdot x_3 \cdot x_4}}}}} = \overline{\overline{\overline{\overline{\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_2 \cdot x_3 \cdot x_4}}}}$$

Используя эти выражения можно построить схему преобразователя кода.

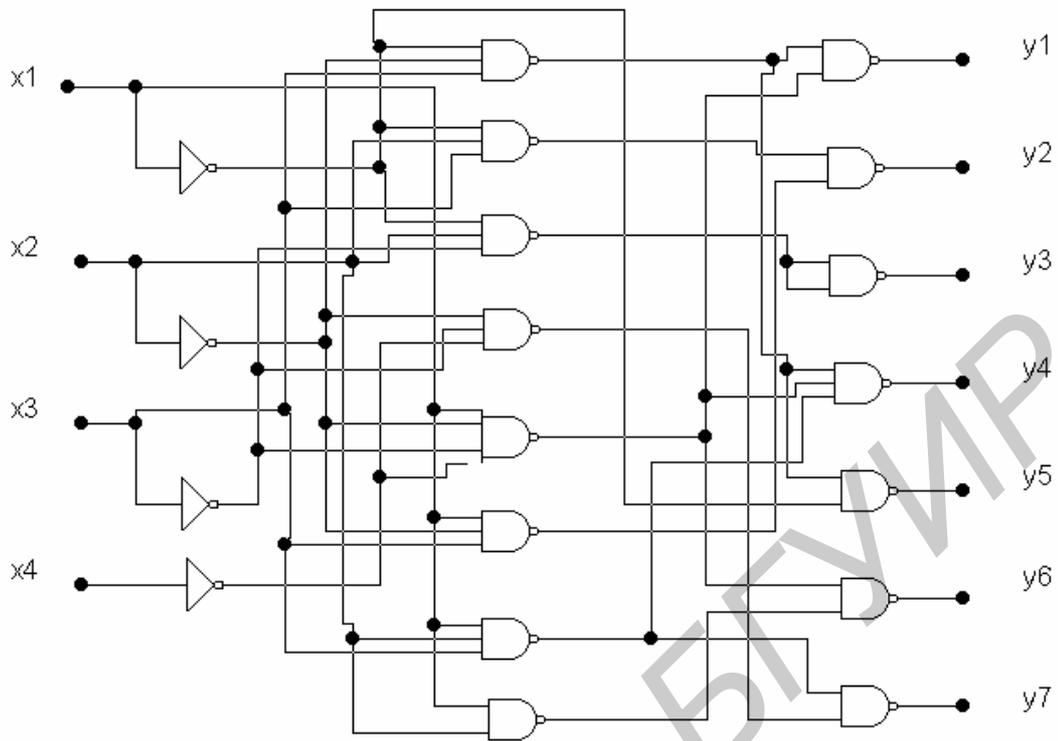


Рис. 2.27. Схема преобразователя кода для семисегментного индикатора.

3. Последовательные цифровые устройства (ПЦУ).

Последовательными называются функциональные узлы, выходные сигналы которых зависят не только от значений входных сигналов, действующих в настоящий момент времени, но и от предыдущих значений. В состав таких устройств входят логические элементы, образующие комбинационную схему (КС), и элементы памяти (ЭП). Структура последовательного функционального узла показана на рисунке.

z-выходные сигналы, их k штук.

x-входные сигналы (n штук).

у-сигналы подаются на элементы памяти – управляющие сигналы элементами памяти, вырабатываемые комбинационной схемой.

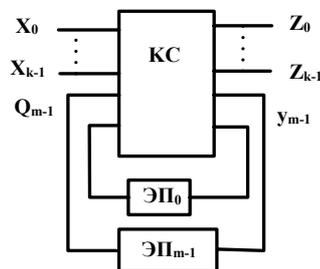


Рис. 3.1. Структура конечного автомата.

Выходные сигналы элементов памяти – q – воздействуют на комбинационную схему и называются внутренними сигналами последовательного узла.

Таким образом, выходной сигнал Z такой системы формируется как результат воздействия входных сигналов X и выходных сигналов с элементов памяти Q^n .

$$Z = \varphi(X, Q^n) \quad (1)$$

Выражение (1) называется функцией выходов, которая устанавливает зависимость значений выходных сигналов Z от значений входных сигналов X и значений внутренних сигналов Q^n в настоящий момент времени. Одновременно при воздействии входных сигналов ячейки памяти изменяют своё значение, и на выходе появляются сигналы Q^{n+1} , т.е.

$$Q^{n+1} = f(X, Q^n) \quad (2)$$

Выражение (2) называется функцией переходов, которая устанавливает зависимость значений внутренних сигналов в следующий момент времени Q^{n+1} от значений входных сигналов X и значений внутренних сигналов Q^n , действующих в данный момент времени.

Учитывая, что все сигналы могут иметь значение либо «1» либо «0», то множество входных, выходных и внутренних сигналов конечно, поэтому ПЦУ относятся к классу конечных автоматов (КА).

Конечные автоматы, закон функционирования которых задаётся соотношениями (1) и (2), называются автоматами Мили.

Иногда встречаются автоматы, у которых значения выходных сигналов не зависят от того, что подаётся на вход.

$$Z = \varphi(Q^n) \quad (3)$$

Модель конечного автомата, закон функционирования которого задаётся соотношениями (2) и (3), называется автоматом Мура. Различия между автоматами Мили и Мура: выходные сигналы в автоматах Мили возникают одновременно с подачей входных сигналов, а в автоматах Мура – с задержкой на 1 такт. Конечные автоматы существуют в дискретном времени, где время разбито на такты.

Различают автоматы:

- Синхронные – имеют генератор тактовых импульсов с постоянной частотой, и входные сигналы воздействуют на автомат только при наличии тактового импульса;
- Асинхронные – длительность такта – переменная величина, определяемая интервалами времени между моментами изменения состояния входов.

В качестве элементов памяти в конечных автоматах используются триггеры (простейшие цифровые автоматы с памятью с простой обратной связью).

Триггер – устройство, имеющее два устойчивых состояния и способное под действием управляющих сигналов скачком переходить из одного состояния в другое.

3.1. Асинхронный RS-триггер с прямыми входами.

Схема триггера, выполненного на двух логических элементах 2ИЛИ-НЕ:

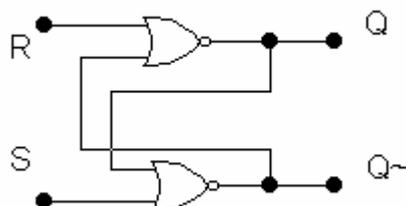


Рис. 3.2. Структурная схема асинхронного RS-триггера на логических элементах ИЛИ-НЕ.

Он содержит два информационных входа R(Reset) и S(Set), а правила его функционирования определяются таблицей:

Такт t_n		Такт t_{n+1}	
S^n	R^n	Q^{n+1}	\bar{Q}^{n+1}
0	0	Q_t	\bar{Q}_t
0	1	0	1
1	0	1	0
1	1	*	*

Из таблицы следует, что комбинация входных сигналов $S^n=R^n=0$ не изменяет состояния триггера. Комбинация входных сигналов $S^n=1, R^n=0$ переводит RS-триггер в единичное состояние. Поэтому вход S называют единичным входом: появление логической 1 на его входе гарантирует наличие логической 1 на прямом выходе триггера независимо от его первоначального состояния. Комбинация входных сигналов $S^n=0, R^n=1$ обеспечивает нулевое состояние триггера. По этой причине вход R называют нулевым входом. $S^n=R^n=1$ – запрещённая комбинация, т.к. при ней нарушается логика триггера.

Переключение триггера под действием входных сигналов описывается таблицей переключений либо характеристическим уравнением.

Таблица переключений:

S^n	R^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	*
1	1	1	*

Для состояния Q^{n+1} карта Карно выглядит следующим образом:

Q^n	$S^n R^n$			
	00	10	11	01
0	0	1	0	0
1	1	1	0	0

$$Q^{n+1} = S^n \bar{R}^n + \bar{R}^n Q^n \quad (3.1)$$

Учитывая особенность RS триггера, необходимо исключить сочетание входных сигналов $S^n=R^n=1$, приводящее к неопределённому состоянию. Поэтому соотношение (1) необходимо дополнить условием

$$S^n R^n = 0, \quad (3.2)$$

которому должны удовлетворять входные сигналы. С учётом этого полученное характеристическое уравнение для RS триггера с прямыми входами запишется в виде

$$Q^{n+1} = S^n + Q^n \bar{R}^n, \quad S^n R^n = 0 \quad (3.3)$$

3.2. Асинхронный RS-триггер с инверсными входами.

Асинхронный RS-триггер можно выполнить и на двухвходовых ЛЭ И-НЕ. В отличие от RS-триггера на ЛЭ ИЛИ-НЕ переключение данного триггера осуществляется сигналами логического 0. Такой триггер называют триггером с инверсным управлением.

Таблица функционирования RS-триггера:

Такт t_n		Такт t_{n+1}	
S^n	R^n	Q^{n+1}	\bar{Q}^{n+1}
0	0	*	*
0	1	1	0
1	0	0	1
1	1	Q_t	\bar{Q}_t

Схема:

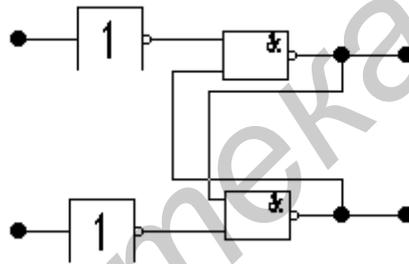


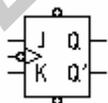
Рис. 3.3 Структурная схема асинхронного RS-триггера на логических элементах И-НЕ.

$R=S=0$ – запрещённое состояние для данной схемы;

$R=S=1$ – сохранение информации.

3.3. Асинхронный JK-триггер.

JK-триггер отличается от RS-триггера отсутствием запрещённых комбинаций информационных сигналов J и K. Обозначаются:



Построение JK триггера на элементах И-НЕ:

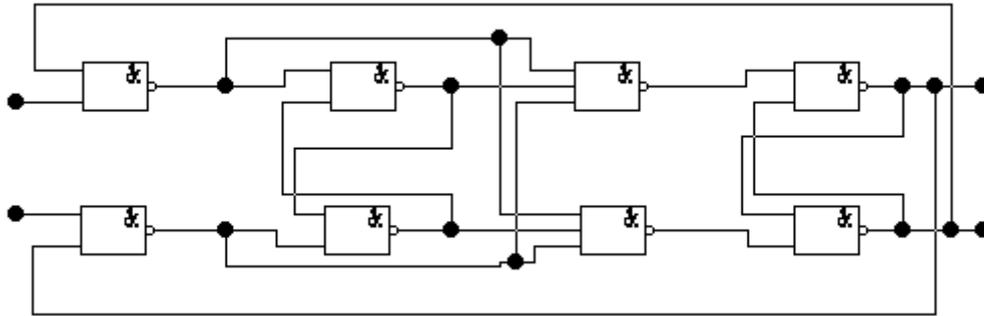


Рис. 3.4. Структурная схема JK-триггера на логических элементах И-НЕ.

Временная диаграмма такого триггера:

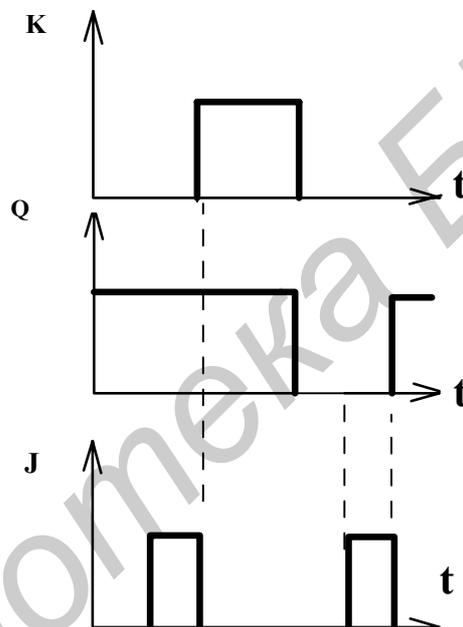


Рис. 3.5. Временные диаграммы JK-триггера

Выход данного JK-триггера переключается *после* окончания импульса. Функционирование JK-триггера описывается следующей таблицей переключений:

J^n	K^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Карта Карно для JK триггера запишется в виде:

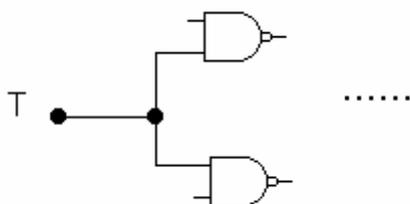
Q^n	$J^n K^n$			
	00	10	11	01
0	0	1	1	0
1	1	1	0	0

Характеристическое уравнение для JK триггера:

$$Q^{n+1} = J^n \cdot \bar{Q}^n + \bar{K}^n Q^n \quad (3.4)$$

3.4. Счётный Т-триггер.

Свойство JK триггера переходить в инверсное состояние при одновременной подаче входных сигналов $J=K=1$ позволяет создать на его основе счётный Т триггер. Для этого достаточно объединить входы J и K. Счётный Т-триггер также называется триггером со счетным запуском. Он откликается на каждую «1». В счётном триггере входы J и K объединяются между собой. Объединённый вход носит название Т вход. Поэтому схема Т-триггера будет такой же, как и JK триггера, но входы J и K объединяют.



Характеристическое уравнение счётного триггера выглядит следующим образом (в уравнении (3.4) переменные J и K заменяются на T)

$$Q^{n+1} = T^n \bar{Q}^n + Q^n \bar{T}^n \quad (3.5);$$

3.5. Функциональное назначение входов триггера:

обозначение	назначение
	Информационные входы:
S	Вход для раздельной установки RS-триггера в состояние «1».
R	Вход для раздельной установки RS-триггера в состояние «0».
J	Вход для установки JK-триггера в состояние «1».
K	Вход для установки JK-триггера в состояние «0».
T	Счётный вход триггера.
D	Вход для установки триггера в состояние «0» или «1»
	Управляющие входы:
V	Подготовительный вход для разрешения приема информации.
C	Подготовительный вход для осуществления приема информации или вход синхронизации.

3.6. Синхронные триггеры на логических элементах.

Триггеры, реализуемые на логических элементах, имеют конечное время переключения t_{sw} , определяемое суммарным средним временем задержки распространения сигнала $\sum_m t_{del\ md}$, где m – число логических элементов,

составляющих триггер. В результате выходной сигнал триггера в течение времени t_{sw} после подачи входного сигнала сохраняет значение, не соответствующее этому сигналу, т.е. является ложным. Это обстоятельство может привести к ошибкам в работе устройства обработки информации, состоящего из большого числа логических элементов. Поэтому считывание информации осуществляется в те моменты времени, когда появление ложного сигнала заведомо исключено. С этой целью на вход триггера с определённым периодом подаются тактовые импульсы, обеспечивающие его срабатывание с строго определёнными моментами времени. Такие триггеры со входом C называются тактируемыми или синхронными.

3.7. Синхронный RS-триггер

Структурная схема и условное обозначение синхронного RS-триггера выглядят следующим образом (рис. 3.6):

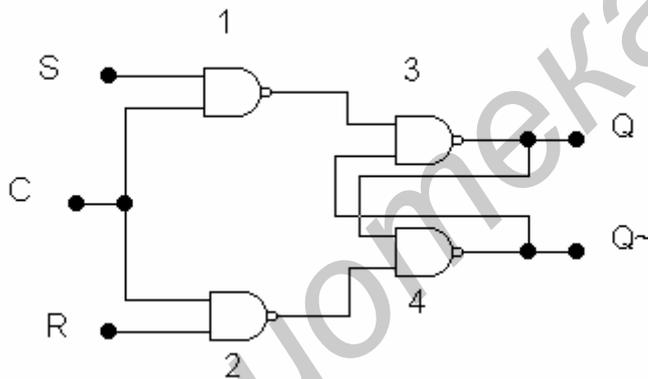


Рис. 3.6а. Структурная схема синхронного RS-триггера

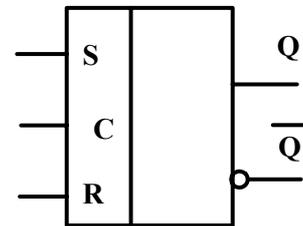


Рис. 3.6б. Условное обозначение синхронного RS-триггера

Структурная схема синхронного RS-триггера помимо элементов 3 и 4, составляющих триггер с инверсными входами, включает два входных элемента И-НЕ (1 и 2), обеспечивающих синхронный режим работы. Буквой C обозначен вход тактовых (синхронизирующих) импульсов. При наличии входного сигнала ($S=1$ или $R=1$) переключение триггера происходит только в момент поступления тактового импульса $C=1$, т.к. при этом условии на одном из входов триггера возникает сигнал лог.0. Временные диаграммы работы синхронного и асинхронного RS-триггеров приведена на следующих рисунках.

Синхронный RS-триггер также имеет запрещённую комбинацию входных сигналов: $R^n S^n C^n = 1$. Действительно, при такой комбинации входных сигналов

выходные сигналы элементов И-НЕ будут соответствовать лог.1, что для $\overline{R}\overline{S}$ -триггера недопустимо.

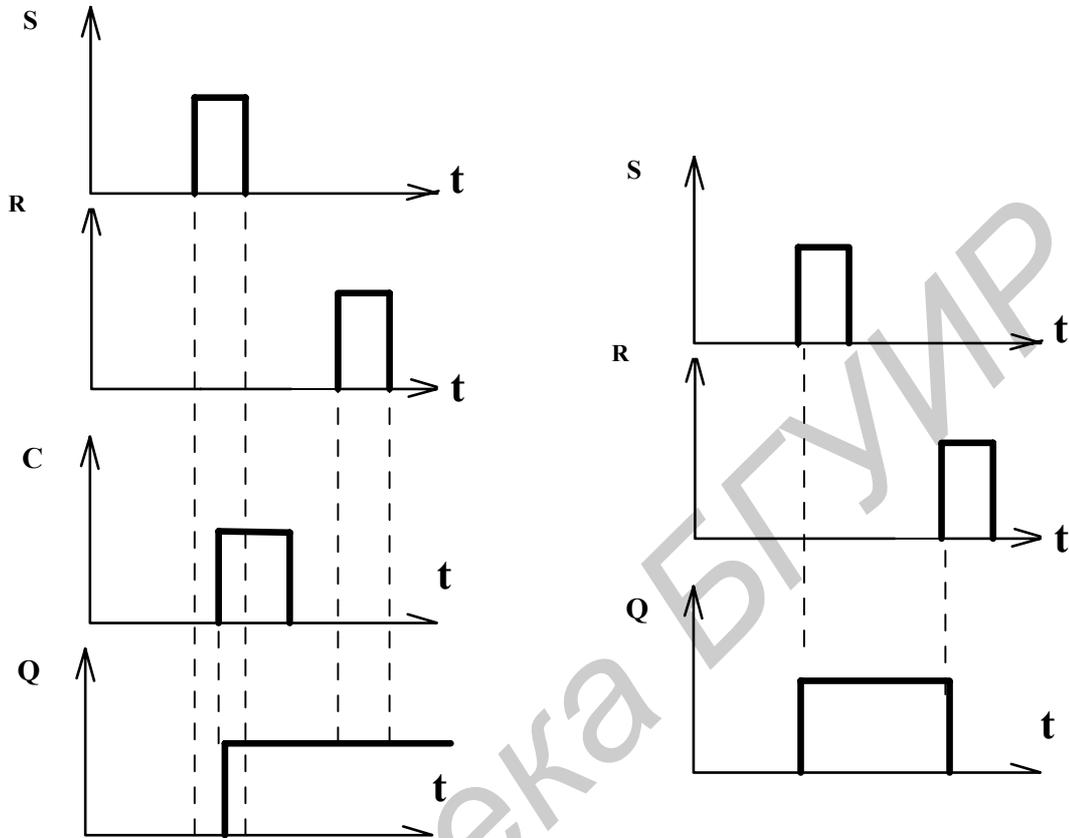


Рис. 3.7. Временные диаграммы синхронного и асинхронного RS-триггеров

Составим таблицу состояний синхронного RS-триггера.

C^n	S^n	R^n	Q^n	Q^{n+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	*

Составим карты Карно:

$C^n Q^n$	$S^n R^n$			
	00	10	11	01
00	0	0	0	0
10	0	1	*	0

Тогда характеристическое уравнение будет

$$Q^{n+1} = \bar{C}^n Q^n + S^n C^n + Q^n \bar{R}^n \quad (3.6);$$

Для исключения неопределённого состояния синхронного RS-триггера входные сигналы должны удовлетворять условию $R^n S^n C^n = 0$.

3.8. Синхронные JK- и T-триггеры.

Имеет такую же структуру, что и асинхронный. Отличие состоит в том, что входные элементы И-НЕ имеют три входа. Входы С объединены и используются для подачи тактовых сигналов.

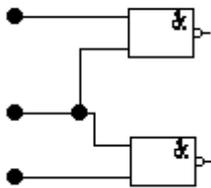


Таблица функционирования JK-триггера:

C^n	J^n	K^n	Q^{n+1}
0	0	0	Q^n
1	0	0	Q^n
1	0	1	0
1	1	0	1
1	1	1	\bar{Q}^n

Характеристическое уравнение синхронного JK-триггера:

$$Q^{n+1} = \bar{K}^n Q^n + \bar{C}^n Q^n + C^n J^n \bar{Q}^n. \quad (3.7)$$

Синхронный счётный T-триггер получают, объединяя информационные входы J и K. Управляющие сигналы T подаются на объединённый вход, а тактовые импульсы – на вход С. Характеристическое уравнение счётного триггера получается из соответствующего уравнения для JK-триггера (3.4) путем замены J и K на T:

$$Q^{n+1} = (\bar{T}^n + \bar{C}^n) Q^n + C^n T^n \bar{Q}^n. \quad (3.8)$$

3.9. Синхронный триггер задержки (D-триггер).

Этот триггер можно получить, объединяя вход J со входом K через инвертор. При таком включении независимо от значения сигнала D^n на одном из управляющих входов имеется уровень лог.1: при $D^n=1$ $J^n=1$, $K^n=0$; при $D^n=0$ $J^n=0$, $K^n=1$. Таким образом исключаются комбинации входных сигналов $J^nK^n=1$ и $J^nK^n=0$. Характеристическое уравнение D-триггера может быть получено из уравнения для синхронного JK-триггера (3.7) путём подстановки в него значений $J^n=D^n$ и $K^n=\bar{D}^n$:

$$Q^{n+1} = Q^n (D^n + \bar{C}^n) + C^n D^n \bar{Q}^n \quad (3.9)$$

Из этого соотношения следует, что при наличии синхронизирующего сигнала ($C^n=1$) на выходе триггера возникает сигнал, соответствующий входному сигналу, имевшему место в предшествующем такте $Q^{n+1}=D^n$. Таким образом, D-триггер осуществляет задержку сигнала на 1 такт.

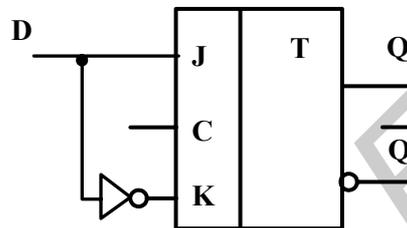


Рис. 3.8. Получение D-триггера из JK-триггера.

3.10. Синхронные триггеры со статическим и с динамическим управлением.

Синхронные триггеры делятся на два типа: со статическим и с динамическим управлением. В первых триггеры реагируют на сигналы на информационных входах во время действия сигнала на синхронизирующем входе. Поэтому изменения

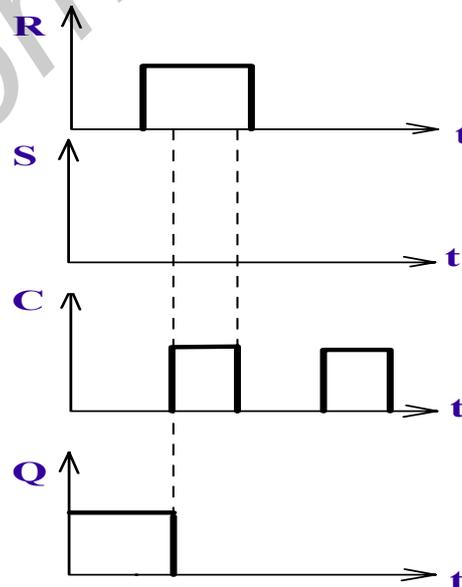


Рис. 3.8. Временные диаграммы статического RS-триггера.

информационных сигналов допускаются только при отсутствии сигнала на синхровходе.

В синхронных триггерах с динамическим управлением приём сигналов с информационных входов происходит в течение малой длительности фронта (положительного или отрицательного) сигнала на синхриводе; или на информационном входе при постоянном сигнале на синхриводе.

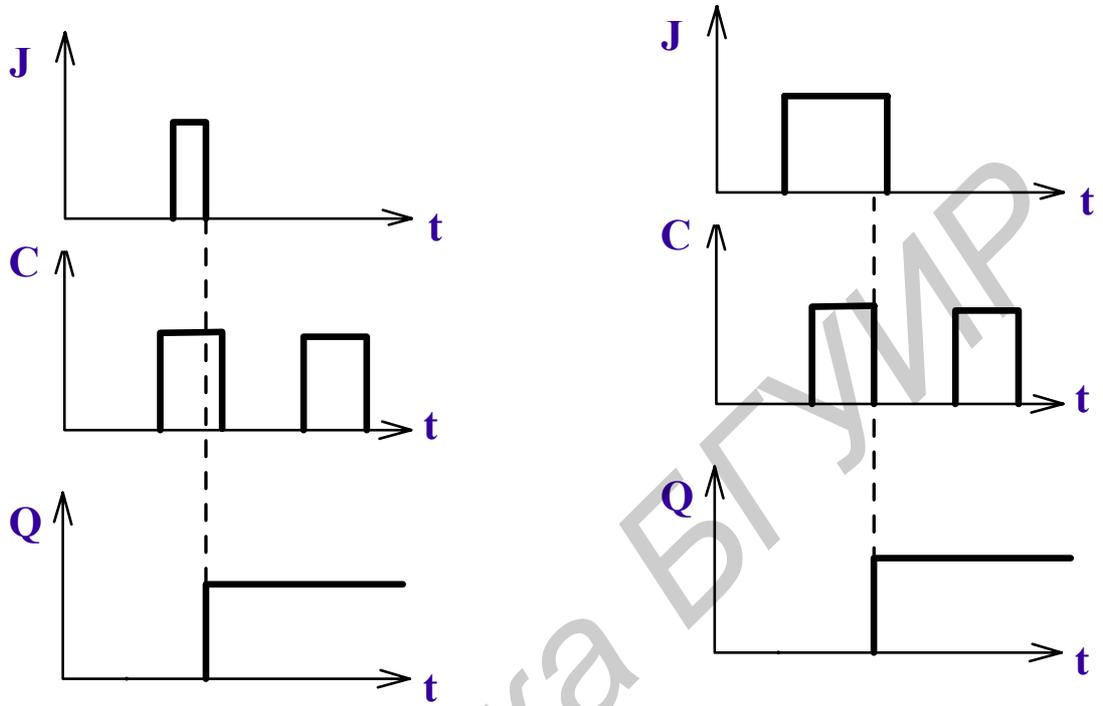


Рис. 3.9. Временные диаграммы JK-триггера с динамическим управлением.

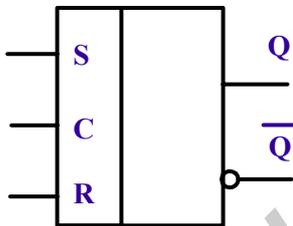


Рис. 3.10. Условное обозначение RS-триггера со статическим управлением.

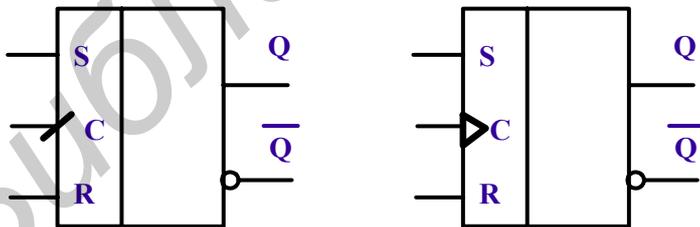


Рис. 3.11. Условное обозначение триггеров, переключающихся по положительному фронту (прямой динамический вход):



Рис. 3.12. Условное обозначение триггеров, переключающихся по отрицательному фронту (инверсный динамический вход):

3.11. Синхронные MS-триггеры.

MS-триггеры (Master – ведущий, Slave – ведомый) – триггеры с двухступенчатым запоминанием информации. Оба триггера и M, и S, функционируют как синхронные триггеры со статическим управлением.

Если на синхровход подать «1», ведущий триггер M устанавливается в новое состояние, в соответствии с сигналом, поступающим на его вход. Ведомый триггер имеет инверсный синхровход и невосприимчив к информации, поступающей на его вход с выхода триггера M. При изменении значения сигнала на синхровходе с «1» на «0», ведущий M-триггер отключается от информационных входов, а ведомый триггер S устанавливается в состояние, определённое выходом M-триггера.

Таким образом, управление в MS-триггере осуществляется двумя фронтами сигнала на синхровходе: на положительном фронте происходит установление ведущего триггера M, на отрицательном - ведомого S-триггера.

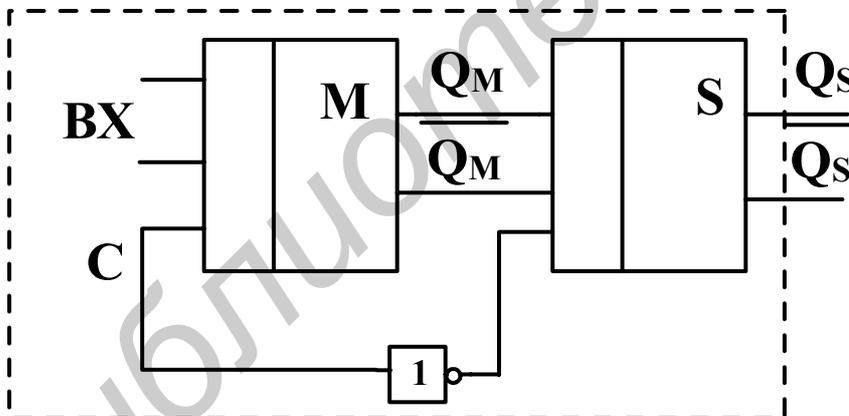


Рис. 3.13. Структурная схема двухступенчатого MS-триггера.

В качестве простейшего примера рассмотрим построение счётного триггера со статическим управлением по принципу двухступенчатого запоминания информации:

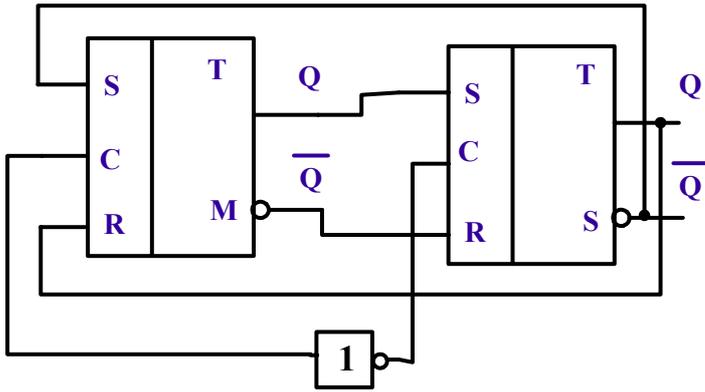
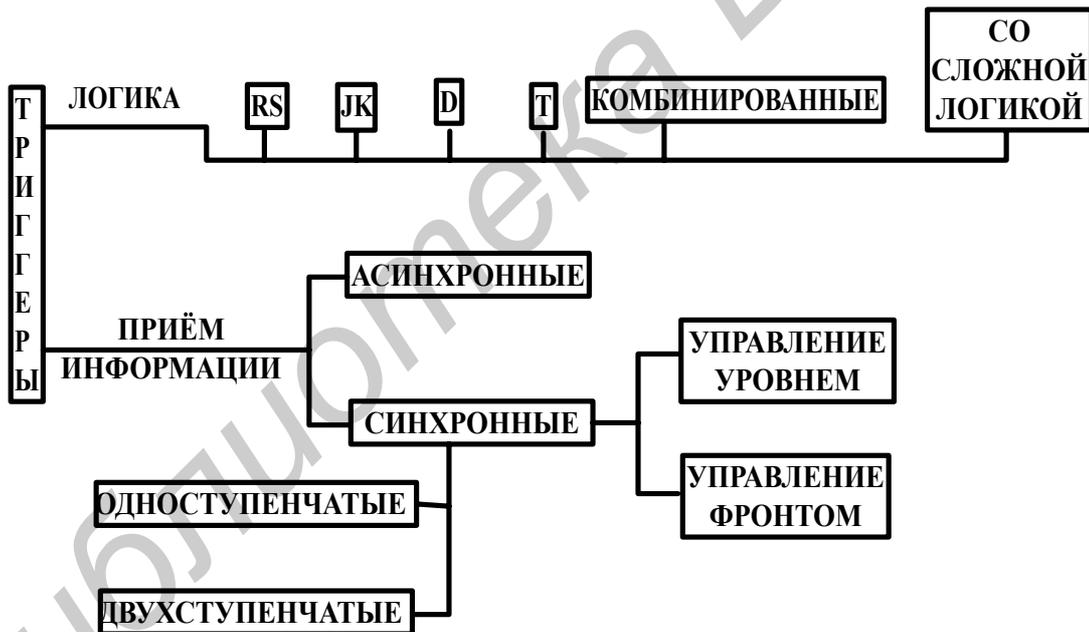


Рис. 3.14. Построение MS-триггера из двух RS-триггеров.

При положительном фронте импульса, поступающего на вход С, триггер М устанавливается в состояние, противоположное состоянию S триггера. При отрицательном фронте входного импульса С происходит передача сигнала, соответствующего состоянию триггера М, в триггер S.

3.12. Итоговая классификация триггеров.



4. Функциональные узлы последовательного типа.

4.1. Словарный метод проектирования функциональных узлов последовательного типа.

Этот метод применяется тогда, когда для элементов памяти применяются синхронизируемые фронтом триггеры.

С этой целью используется функция переходов, которой приписывают следующие значения:

$$F_Q=0, \text{ если } (Q^{n-1} \rightarrow Q^n) 0 \rightarrow 0.$$

$$F_Q=1, \text{ если } 1 \rightarrow 1.$$

$F_Q = \Delta$, если $0 \rightarrow 1$.

$F_Q = \nabla$, если $1 \rightarrow 0$.

Приведём значения F_Q для триггеров.

F_Q	$\overline{R}^n \overline{S}^n$		JK		T	D
	R	S	J	K		
0	0	X	0	X	0	0
1	X	0	X	0	0	1
Δ	1	0	1	X	1	1
∇	0	1	X	1	1	0

Использование словарного метода предусматривает на первом этапе составление таблицы состояний и кодировку состояний узла внутренними переменными. Далее производится определение функции переходов F_Q , определяется тип триггеров и осуществляется минимизация функций их входов. На последнем этапе составляется структурная схема функционального узла.

4.2. Построение сдвигового регистра словарным методом.

Регистр – функциональный узел последовательностного типа, осуществляющий прием, хранение и передачу информации.

Регистры делятся на:

- с последовательным приёмом и выдачей информации (сдвиговые);
- с параллельным приёмом и выдачей информации (регистры памяти).

В регистрах для хранения каждого разряда информации предназначен отдельный триггер.

В сдвиговых регистрах при записи многоразрядного числа используется цепочка триггеров, информация о каждом разряде последовательно переходит от одного триггера к другому. Для реализации сдвиговых регистров используются синхронизируемые фронтом D-триггеры. Определим структурную схему такого регистра. В соответствии с логикой функционирования при воздействии тактового импульса информация с выхода предыдущего триггера T_{k+1} воздействует на вход последующего триггера T_k .

Используя словарный метод, составим таблицу состояний такого узла.

Q_{k+1}	Q_k^n	Q_k^{n+1}	F_Q
0	0	0	0
0	1	0	∇
1	0	1	Δ
1	1	1	1

Q_k^n – значение сигнала на выходе T_k триггера;

Q_k^{n+1} – возможное значение выходного сигнала после воздействия входного сигнала.

Карта переходов для функции F_Q :

Q_k^n	Q_{k+1}	
	0	1
0	0	Δ
1	∇	1

Используя связь между функцией переходов и значением информационных сигналов составим карту Карно для информационного входа триггера задержки:

D_k :

Q_k^n	Q_{k+1}	
	0	1
0	0	1
1	0	1

Отсюда получим $D_k = Q_{k+1}$.

Следовательно, в сдвиговом регистре выход предыдущего триггера напрямую соединяется с информационным входом последующего триггера.

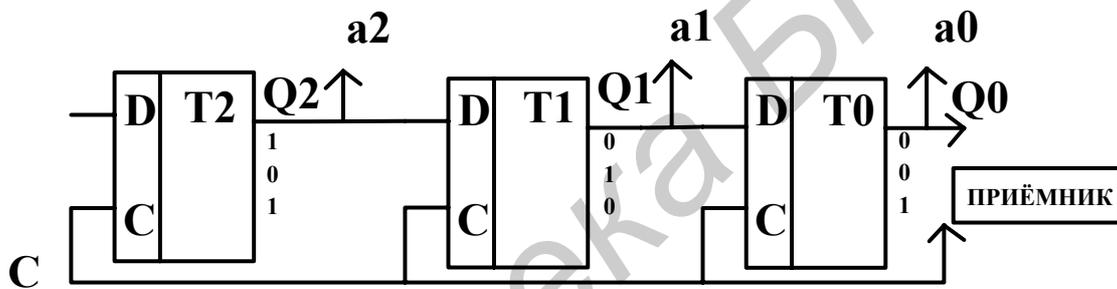


Рис. 4.1. Построение сдвигового регистра на D-триггерах.

Перед записью информации все триггеры обнуляются, для чего используется R-вход, подача сигнала на который переводит триггеры в нулевое состояние на прямых выходах. Рассмотрим процесс ввода, хранения и вывода информации на примере трёхразрядного числа 101. В начале на вход триггера T2 подаётся уровень лог.1 младшего разряда. При воздействии первого импульса синхронизации в соответствии с правилами функционирования D-триггера в момент окончания синхроимпульса на выходе T2 возникает уровень лог.1, но поскольку к этому моменту синхроимпульс закончился, то уровень лог.1 останется на входе триггера T1. В следующем такте на вход T2 подаётся лог.0 и после окончания второго тактового импульса на выходе T2 будет уровень лог.0, а на выходе T1 – уровень лог.1. К приходу третьего тактового импульса на вход T2 подаётся уровень лог.1 и после окончания тактового импульса этот уровень лог.1 перейдёт на выход T2, а поступившие ранее сигналы 0 и 1 перейдут соответственно на выходы триггеров T1 и T0.

На этом процесс записи информации заканчивается. Если теперь прекратить подачу тактовых импульсов, то регистр будет хранить записанную информацию. Вывод информации осуществляется последовательно путём подачи тактовых импульсов. При подаче трёх тактовых импульсов информация последовательно выводится из регистра. Одновременно с выводом может производиться запись новой информации.

Достоинства сдвиговых регистров: регистры могут сдвигать как слева на право, так и наоборот, используется один канал.

Недостаток: малое быстродействие.

4.3. Построение реверсивного регистра словарным методом.

С помощью сдвиговых регистров можно осуществлять операцию деления или умножения на число, кратное степени двойки, т.е. сдвигать информацию не только слева направо, но и в обратном направлении, т.е. осуществлять реверс. Это реверсивные регистры.

Управление направлением сдвига информации осуществляется путём подачи управляющих сигналов X_0 .

Осуществим словарным методом построение реверсивного регистра сдвига. Договоримся, что $X_0=1$ будет соответствовать сдвигу вправо, а при $X_0=0$ – происходит сдвиг влево. Составим таблицу состояний.

X_0	Q_{k+1}	Q_{k-1}	Q_k^n	Q_k^{n+1}	F_Q	X_0	F_Q	Q_k^{n+1}
1	0	0	0	0	0	0	0	0
1	0	0	1	0	∇	0	∇	0
1	0	1	0	0	0	0	Δ	1
1	0	1	1	0	∇	0	1	1
1	1	0	0	1	Δ	0	0	0
1	1	0	1	1	1	0	∇	0
1	1	1	0	1	Δ	0	Δ	1
1	1	1	1	1	1	0	1	1

Карты переходов для функции переходов на основе таблицы состояний:

F_Q :

$Q_{k-1}Q_k^n$	X_0Q_{k+1}			
	00	10	11	01
00	0	0	Δ	0
10	Δ	0	Δ	Δ
11	1	∇	1	1
01	∇	∇	1	∇

Тогда карта Карно для информационного входа D_k выглядит следующим образом:

$Q_{k-1}Q_k^n$	X_0Q_{k+1}			
	00	10	11	01
00	0	0	1	0
10	1	0	1	1
11	1	0	1	1
01	0	0	1	0

Откуда получаем $D_k = X_0Q_{k+1} + \bar{X}_0Q_{k-1}$;

После чего можем строить схему реверсивного регистра.

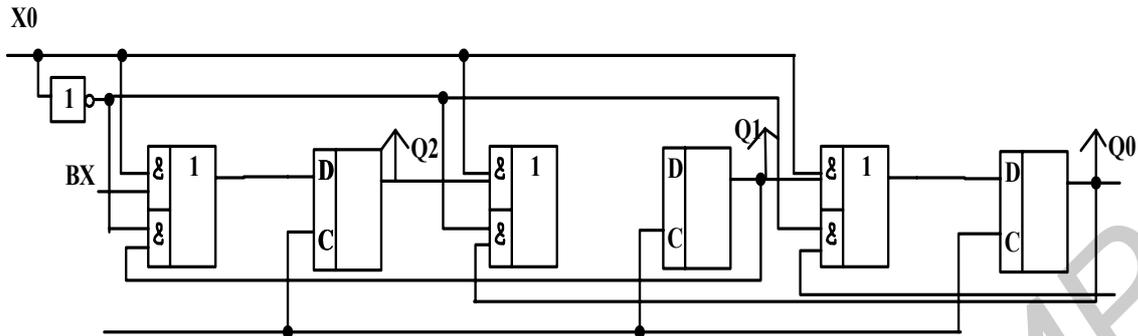


Рис. 4.2. Структурная схема реверсивного регистра.

Регистры сдвига используются также в качестве линии задержки сигнала.

4.4. Тактируемые параллельные регистры.

В параллельных регистрах каждый из триггеров имеет свой независимый информационный вход (D_i) и свой независимый информационный выход (Q_i). Тактовые входы (C) всех триггеров связаны между собой. В результате параллельный регистр представляет собой многоразрядный, многоходовой D-триггер.

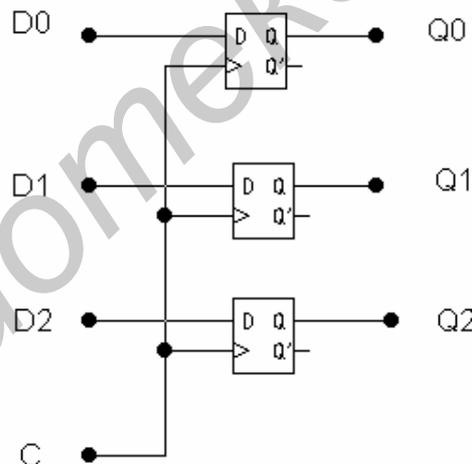


Рис. 4.3. Структурная схема параллельного регистра.

Все параллельные регистры делятся на две группы:

- регистры, срабатываемые по фронту управляющего сигнала C (тактируемые);
- регистры, срабатываемые по уровню, управляющего сигнала C (стробируемые регистры, Latch).

Чаще используются тактируемые регистры.

Принцип действия тактируемых регистров, срабатываемых по фронту, не отличается от принципа действия D-триггера. По положительному фронту тактового сигнала C каждый из выходов регистра устанавливается в тот уровень, который был в этот момент времени на входе для данного D_i . Т.е. если один триггер запоминает 1

разряд информации, то регистр сразу запоминает несколько (4,6,8,16) сигналов. Большинство регистров имеют 8 разрядов, т.е. запоминают 1 байт информации.

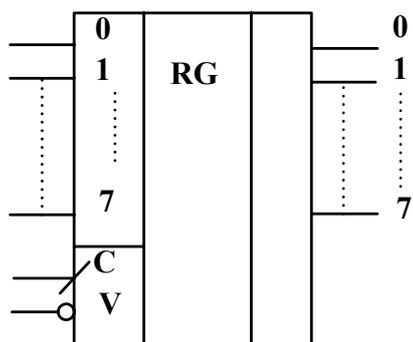


Рис. 4.4. Параллельный регистр серии ИР27.

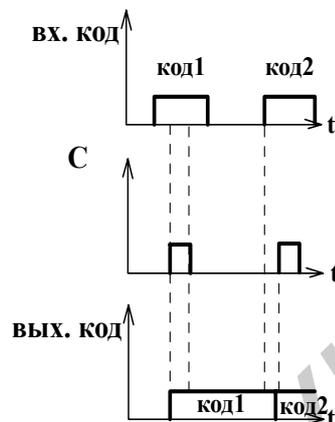


Рис. 4.5. Хранение кода в параллельном регистре

4.5. Стробируемые параллельные регистры.

Их ещё называют регистры-зашёлки (Latch). Они являются гибридом между буфером и регистром. В основе их работы лежит D-триггер, но со статическим управлением.

Когда $C=1$, такой регистр работает как буфер – он пропускает через себя входную информацию.

Когда $C=0$, регистр переходит в режим хранения последнего из пропущенных значений входного кода.

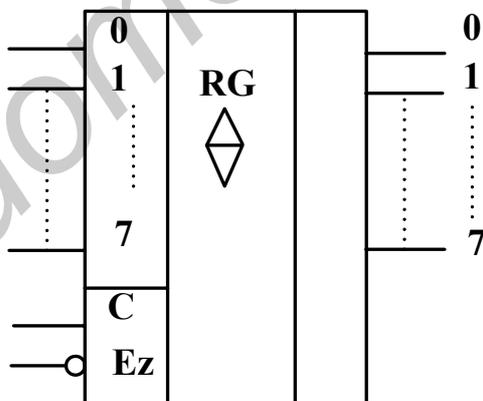


Рис. 4.6. Параллельный регистр серии ИР22.

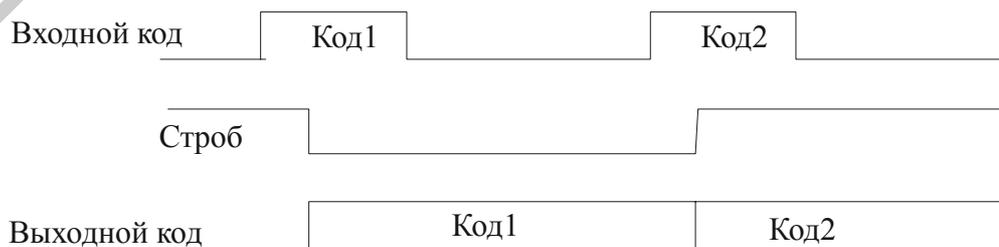


Рис. 4.7. Продление длительности входного кода с помощью регистра-зашёлки

В ряде случаев такие регистры могут заменять тактируемые регистры. Такая замена возможна в случае необходимости запоминания входного кода по сигналу С до момента прихода следующего сигнала С. Сигнал С должен в данном случае быть коротким положительным импульсом. Он обязательно должен быть "вложен" во входной код, т.е. начинаться после начала кода, а заканчиваться до конца кода. (Это так называемый вложенный цикл.)

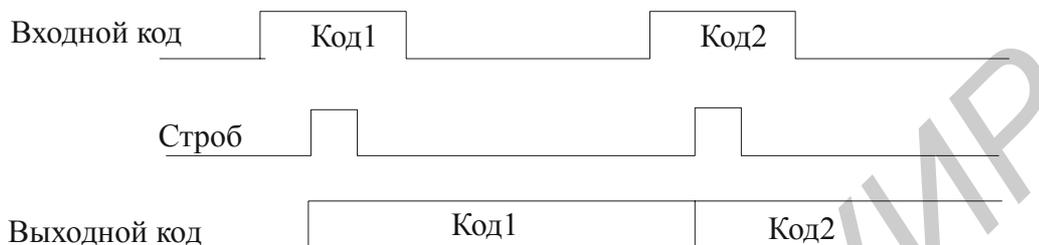


Рис. 4.8. Использование регистра-защёлки для замены регистра, срабатывающего по фронту

По переднему фронту сигнала С регистр перейдет в режим пропускания входного кода, а по заднему – в режим его хранения. Поэтому записываемый код на выходе регистра появится по положительному фронту сигнала С, то есть точно так же, как и в случае регистра, срабатывающего по фронту.

4.6. Счётчики.

Счётчиком называется функциональный узел, предназначенный для подсчета количества импульсов. Основные параметры счётчика:

коэффициент пересчета $K_{сч}$ – максимальное число входных импульсов, которое может быть сосчитано счётчиком. $K_{сч}$ зависит от разрядности;

быстродействие – характеризуется минимальным интервалом между двумя входными импульсами, на которые счетчик реагирует однозначно.

Если 2^n – разряд, то максимальное число, до которого считает счётчик, равно $2^n - 1$.

Счётчики делятся на:

- простые:
 1. суммирующие;
 2. вычитающие.
- реверсивные.

В суммирующих счётчиках каждый последующий импульс увеличивает значение счетчика на 1, а в вычитающем – уменьшает.

Реверсивные могут работать в обоих режимах.

По способу переключения триггера во время счёта импульсов, счетчики делятся на:

- асинхронные;
- синхронные.

По способу организации цепей переноса информации между разрядами:

- с последовательным переносом информации;
- со сквозным переносом;
- с параллельным переносом информации.

Если выполняется условие $K_{сч} = 2^n$ – это двоичный счетчик.

4.7. Суммирующий счётчик с последовательным переносом.

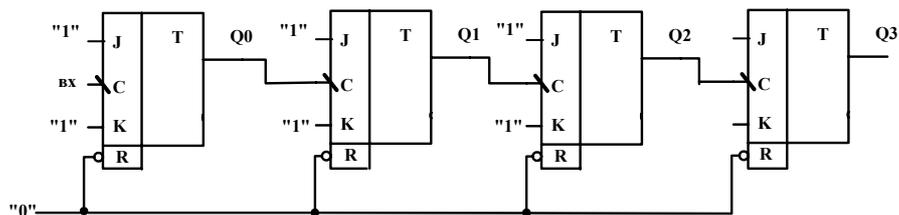


Рис. 4.9. Схема суммирующего четырёхразрядного счётчика с последовательным переносом.

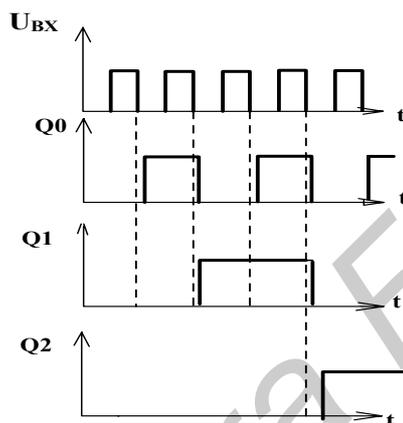


Рис. 4.10. Временные диаграммы, поясняющие работу счётчика.

Перед началом работы все разрядные счётчики устанавливаются в нулевое состояние. Первый импульс, поданный на счётный вход первого триггера, устанавливает в состояние 1 первую ступень. Вторая (основная) ступень этого триггера устанавливается в единичное состояние после окончания действия первого импульса с некоторой задержкой. Состояния остальных триггеров при этом не изменяются. Вторым счётным импульсом первый триггер переводится в нулевое состояние, которое устанавливается в нём после окончания действия второго импульса. Третьим счётным импульсом снова переводит первый триггер в единичное состояние и т.д. Таким образом, каждый нечётный импульс, поступающий на счётный вход первого триггера, переводит его в единичное состояние, а каждый чётный возвращает в нулевое. В результате на выходе триггера первого разряда образуются прямоугольные импульсы, частота которых в два раза меньше частоты входных импульсов. Эти импульсы поступают на счётный вход триггера второго разряда. Если проанализировать работу всей цепочки триггеров, то можно установить соответствие состояний триггеров количеству поступивших на вход схемы импульсов, что и составляет суть работы счётчика.

Недостаток данной схемы накопление времени задержки. Если время задержки превысит интервал между импульсами – произойдёт потеря информации.

4.8. Суммирующий счётчик со сквозным переносом.

Смысл сквозного переноса: переключение триггера любого разряда происходит лишь после завершения переходных процессов в триггерах предыдущих разрядов и при условии, что все они находились в «1» состоянии.

На входе третьего и каждого последующего триггера ставятся схемы совпадения,

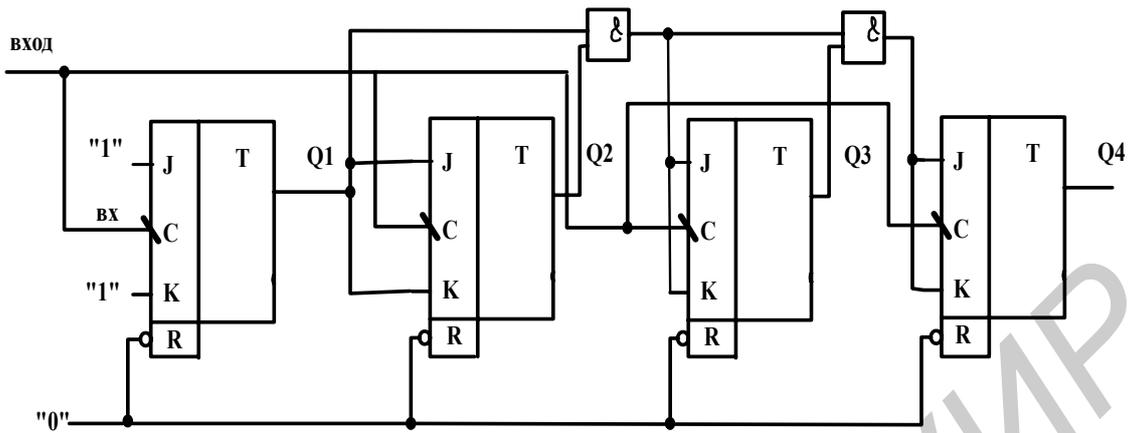


Рис. 4.11. Схема суммирующего счётчика со сквозным переносом.

представляющие из себя логический вентиль «И».

Время задержки здесь не накапливается., $t_3 = mt_{3лэ} + t_{3т}$, где m – число элементов «И».

4.9. Суммирующий счётчик с параллельным переносом.

Имеет большее быстродействие и помехоустойчивость по сравнению с другими.

На входы J и K каждого последующего триггера через логические элементы «И» подаются сигналы со входов всех предыдущих триггеров.

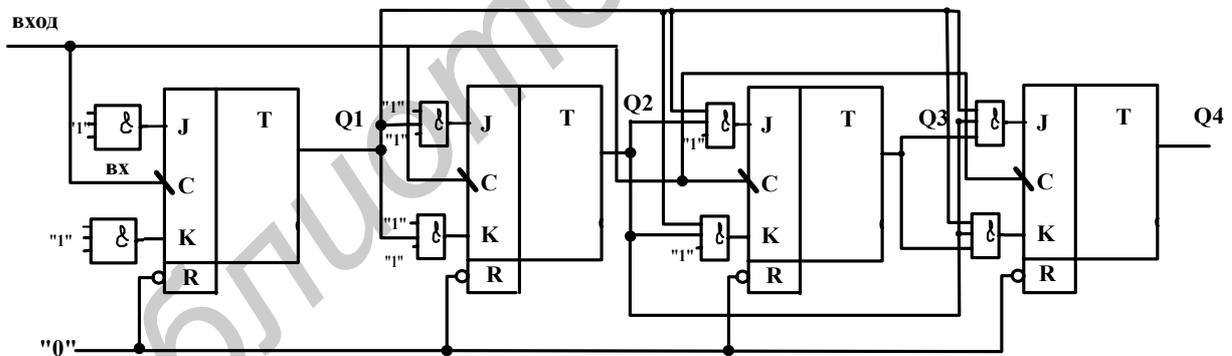


Рис. 4.12. Схема суммирующего счётчика с параллельным переносом.

Время установки первого состояния всегда равно времени переключения первого разряда. Т.к. практически все JK-триггеры имеют встроенные элементы «И», то схема такого счетчика не требует дополнительных логических элементов, но т.к. число встроенных в триггер входов «И» не больше трёх для J и не больше трёх для K, то на таких триггерах можно выполнять только четырёхразрядный счетчик с параллельным переносом. Для увеличения числа разрядов счетчик разбивают на группы по четыре разряда, в каждой группе реализуется параллельный перенос, а между группами – сквозной. Такие счетчики называют *групповыми*.

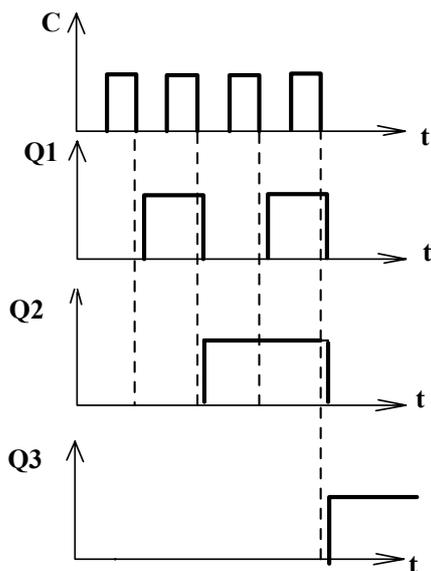


Рис. 4.13. Временная диаграмма работы суммирующего счётчика с параллельным переносом

5. Аналого-цифровые и цифроаналоговые преобразователи информации.

5.1. ЦАП на основе резисторных матриц R-2R.

Для построения ЦАП могут использоваться различные принципы. Наиболее часто используется принцип формирования токов, пропорциональных весовым коэффициентам разрядов двоичного кода с последующим их суммированием в разрядах кода, содержащих 1. Формирование указанных токов обычно производится с помощью резисторной матрицы R-2R. Схема такого формирователя токов на 3 разряда приведена на рисунке 5.1.

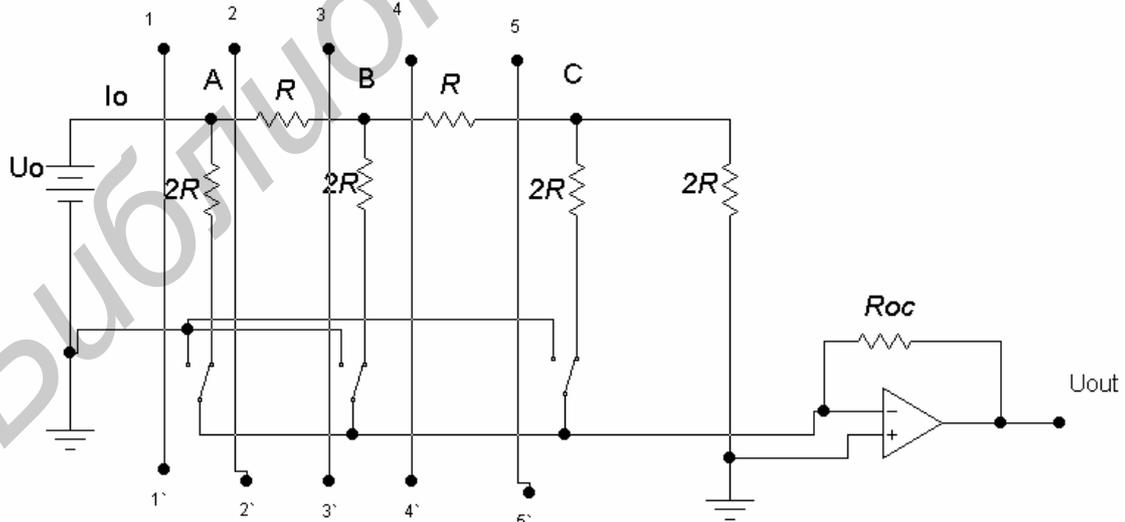


Рис. 5.1. Схема формирователя токов на 3 разряда с помощью матрицы R-2R.

Входное сопротивление схемы правее точек 5-5' равно R, правее точек 4-4' равно 2R, правее точек 3-3' равно R, правее точек 2-2' равно 2R, и, наконец, правее

точек 1-1' равно R . Таким образом, источник опорного напряжения выдаёт в матрицу ток $I_0=U_{оп}/R$. Т.к. сопротивление между точками 2-2' равно $2R$, то в точке А ток I_0 разветвляется в две ветви с равными сопротивлениями $2R$ и, следовательно, по разрядному регистру 1-ого разряда потечёт ток $I_1=I_0/2=I_0 \cdot 2^{-1}$ и ток $I_0/2$ потечёт по резистору R от точки А к точке В. Так как сопротивление правее линии 4-4' равно $2R$, то в точке В ток опять раздвоится и через регистр 2-ого разряда потечёт ток $I_2=I_0/4=I_0 \cdot 2^{-2}$. Этот же ток, $I_0 \cdot 2^{-2}$, потечёт от точки В к точке С. Здесь опять произойдёт разветвление и через регистр 3-ого разряда потечёт ток $I_3=I_0/8=I_0 \cdot 2^{-3}$. Очевидно, такую матрицу можно расширить, предусмотрев в ней любое число разрядных регистров. При этом ток в i -ом разрядном регистре будет равен $I_i=I_0 \cdot 2^{-i}$. Коэффициент 2^{-i} равен весовому коэффициенту i -ого разряда дробного двоичного числа.

При подключении к выходу схемы операционного усилителя, на аналоговом выходе ЦАП образуется напряжение

$$U_{out} = I_{out} R_{oc} = I_0 R_{oc} \sum_{i=1}^3 a_i 2^{-i} = U_0 \frac{R_{oc}}{R} \sum_{i=1}^3 a_i 2^{-i}$$

Так как $\sum_{i=1}^3 a_i 2^{-i} = N$ есть десятичное представление двоичного числа, то

$$U_{out} = U_0 \frac{R_{oc}}{R} N, \text{ т.е. на аналоговом выходе образуется напряжение,}$$

пропорциональное значению входного числа.

В ИМС используют полевые транзисторы в роли ключей, на затвор которых подается необходимая информация, позволяющая направлять необходимый ток на «землю» или на ОУ. Выходное U , снимаемое с ОУ будет пропорционально току, поступающему на его вход.

5.2. АЦП последовательных поразрядных приближений.

Рассмотрим АЦП, построенный на принципе последовательных поразрядных приближений. На рисунке 5.2 приведена функциональная схема такого АЦП.

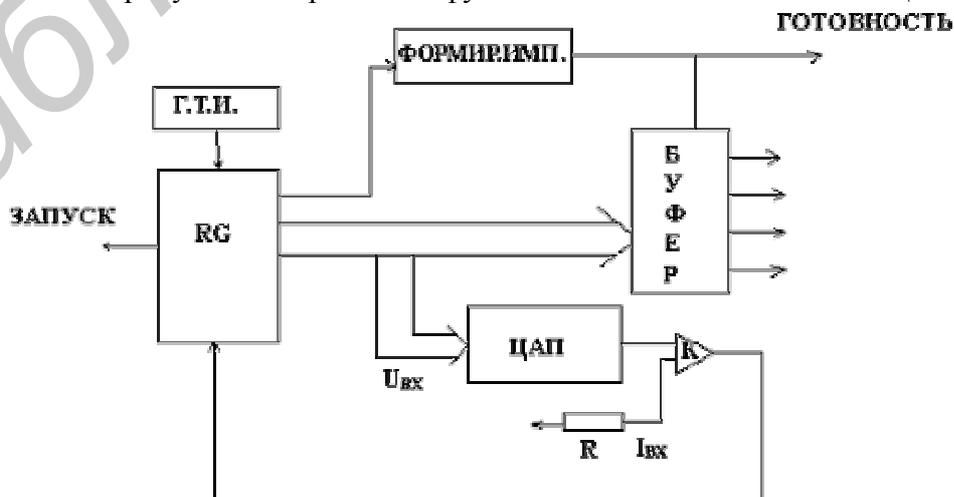


Рис. 5.2. Функциональная схема АЦП последовательных поразрядных приближений.

В преобразовании предусматривается десятиразрядный регистр, в котором последовательно, разряд за разрядом, начиная со старшего разряда, формируется двоичное число, соответствующее цифровой форме представления поданного на вход микросхемы аналогового напряжения. Процесс формирования этого числа состоит в следующем.

Регистр сбрасывается в нуль и затем записывается единица в триггер старшего (десятого) разряда регистра. Получающееся в регистре число с помощью ЦАП преобразуется в пропорциональный ему ток, который с помощью аналогового компаратора тока сравнивается с входным током, возникающим под действием входного напряжения. При выполнении неравенства $I_{вх} \geq I_{ЦАП}$ «1» в триггере старшего разряда регистра остается. При обратном неравенстве триггер старшего разряда сбрасывается в «0». Далее производится запись «1» в триггер 9-ого разряда, и вновь сравнивается $I_{вх}$ и $I_{ЦАП}$. Так производится опробование во всех 10 разрядах регистра. После чего получаемое в регистре число выдается на выход через буфер, построенный на элементах с тремя состояниями.

5.3. АЦП параллельного действия.

Более высоким быстродействием (меньшим временем преобразования) обладают АЦП параллельного действия. Структурная схема АЦП параллельного действия приведена на рисунке 5.3 и включает в себя источник опорного напряжения, резистивный делитель $R_1 \dots R_{n+1}$, компараторы напряжения, преобразователь кодов, выходной буферный регистр и генератор тактовых импульсов. Для преобразования аналогового сигнала в n-разрядный код требуется $N=2^n-1$ компараторов. При таком числе максимальное значение преобразования аналогового сигнала определяется соотношением: $U(t)=(1-1/2^n)U_{оп}$. На один вход каждого компаратора подается

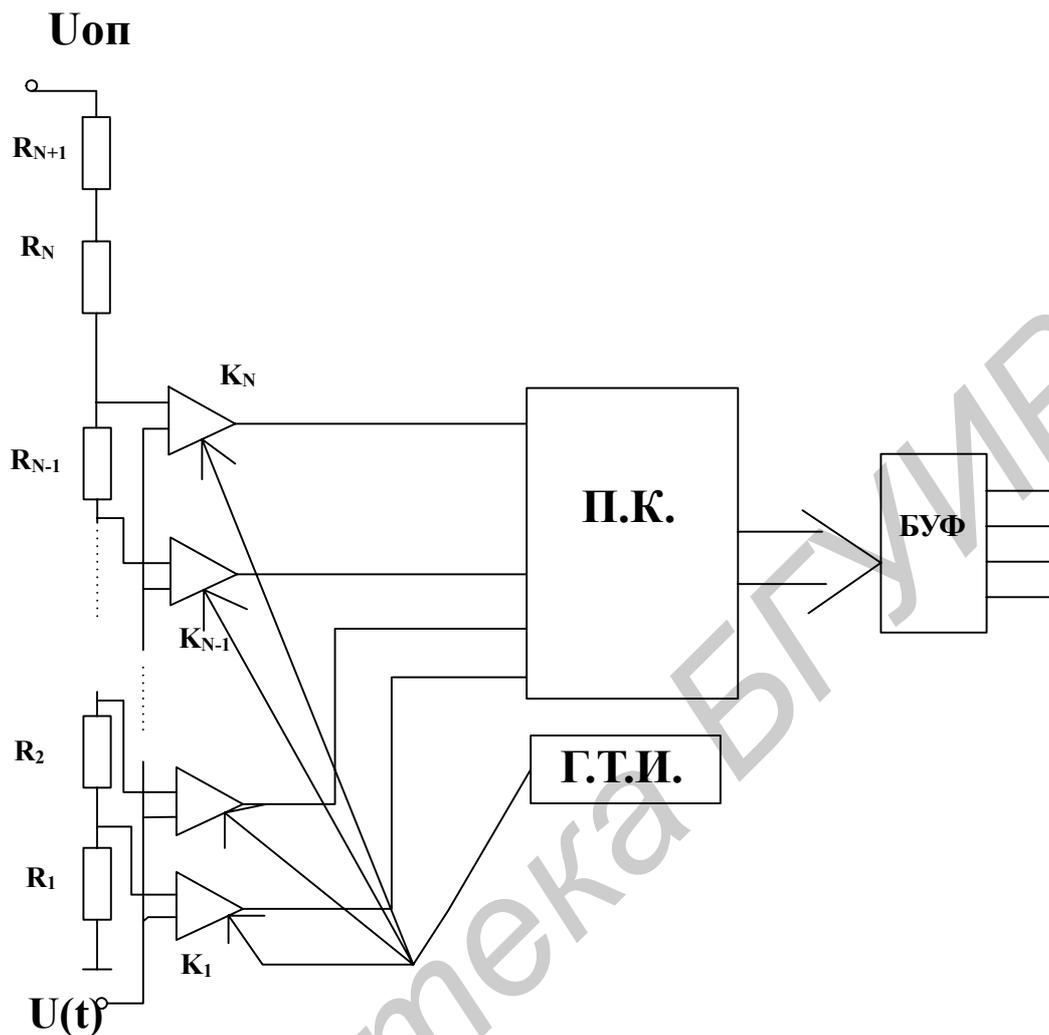


Рис. 5.3. Структурная схема АЦП параллельного действия.

напряжение с соответствующей точки делителя, а на второй вход всех компараторов одновременно поступает аналоговый сигнал $U(t)$. Таким образом на входе каждого компаратора постоянно присутствует заданный уровень напряжения, с которым осуществляется сравнение аналогового сигнала.

Если величина аналогового сигнала $U(t)$ превышает уровень U с делителя, то на его выходе вырабатывается уровень «1». Если $U(t)$ меньше – то на выходе будет уровень «0», т.е. результат получается в унитарном коде ($x_1x_2x_3\dots$). Эти сигналы поступают на вход преобразователя кодов, который преобразует унитарный код в двоичный n -разрядный код. Затем этот код записывается в буферном регистре.

Управление работой такого АЦП осуществляется тактовыми импульсами, которые одновременно подаются на компараторы, преобразователь кодов и буферный регистр. Выборка аналогового сигнала производится после появления фронта тактового импульса на управляющем входе компаратора. Преобразование унитарного кода, поступающего со всех компараторов, осуществляется с момента поступления на входы управления логических схем преобразователя кодов среза тактового импульса. Результат преобразования, т.е. n -разрядный код, записывается в буферный регистр во время появления фронта следующего импульса.

АЦП параллельного действия обладает самым высоким быстродействием (до 10 нс), что позволяет вести обработку сигналов в реальном масштабе времени. Количество разрядов выходного кода - до 8.

6. Полупроводниковые запоминающие и программируемые устройства.

6.1. Масочные ЗУ (ПЗУ).

ПЗУ – постоянное запоминающее устройство;

ППЗУ – программируемое постоянное запоминающее устройство.

ПЗУ – это масочные ЗУ, информация в которые записывается при изготовлении ИМС на предприятии с помощью шаблона маски на завершающем этапе технологического устройства.

ППЗУ – программируются после изготовления на предприятии в лаборатории потребителя без использования сложных технологических процессов, с помощью программаторов.

Рассмотрим ПЗУ – это прообраз ПЛИС. Элементами связи в масочных ЗУ могут быть диоды и транзисторы. Программирование памяти заключается в том или ином расположении элементов связи между горизонтальными и вертикальными шинами связи запоминающих элементов.

Обычно ПЗУ изображают:

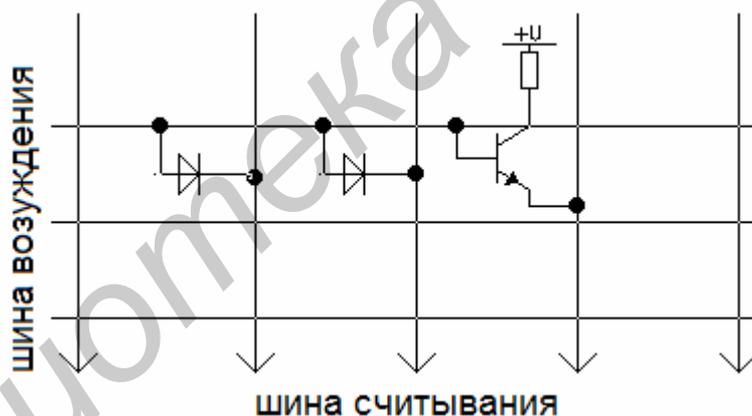


Рис. 6.1. Структура ПЗУ.

В матрице диодного ПЗУ горизонтальные линии являются – линии выбора слов, а вертикальные – линии считывания. Читаемое слово определяется расположением диодов в узлах сетки. Про наличии диода высокий потенциал выбранной горизонтальной линии передается на соответствующую вертикальную линию в данном разряде слова имеем 1.

ЗУ с матричным программированием отличается высоким уровнем интеграции. В качестве элементов связи в масочных ЗУ могут быть диоды, МОП транзисторы, биполярные транзисторы.

Элемент масочного ЗУ:

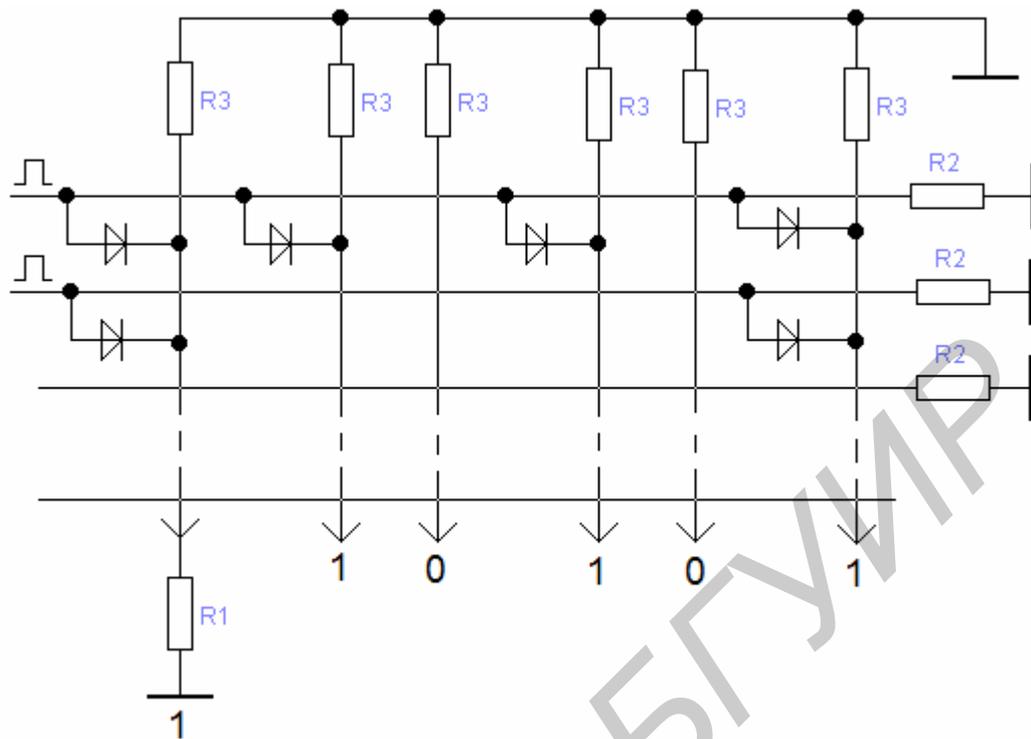


Рис. 6.2. Пример реализации масочного ЗУ на диодной матрице.

$$R3 \approx R2 \gg R1$$

При больших объемах производства масочное программирование предпочтительнее. При малой тиражности ЗУ затраты на ЗУ велики и не рентабельны. Отсюда и следует применение масочных ЗУ в качестве хранителя стандартной информации, имеющей широкий круг потребителя. Пример масочных ЗУ – прошивка кодов букв алфавитов, телефонной клавиатуры, таблиц типов функций (калькулятор, стандартное программное обеспечение).

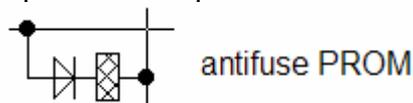
6.2. Программируемые постоянные запоминающие устройства (ППЗУ).

Программируемые постоянные запоминающие устройства делятся на два вида:

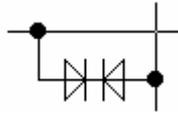
- настраиваемое на единицу;
- настраиваемое на нули.

В ЗУ типа ППЗУ микросхемы программируются путем устранения или создания специальных перемычек. В исходной заготовке имеются или отсутствуют все перемычки, после программирования остаются или возникают необходимые. Fuse – PROM – прожигание, antifuse – PROM – восстановление. Fuse – PROM может быть металлический сплав титан-вольфрам или поликристаллический кремний. В исходном состоянии логический элемент хранит уровни логических единиц, а логический ноль записывается расплавлением перемычки. Иногда перемычки создаются в виде тонких непроводимых диэлектрических слоев.

Ставят конденсатор, который потом пробивают



иногда используют следующее соединение:



на стадии изготовления – ноль.

6.3. Программируемые ЗУ с плавкими перемычками.

Реализуются с простыми аппаратными средствами.

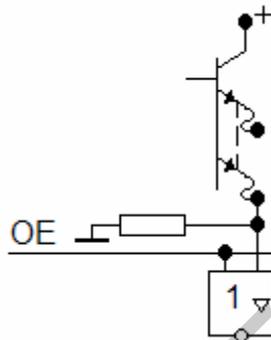


Рис. 6.2. Многоэмиттерный транзистор с плавкими перемычками в эмиттерной цепи.

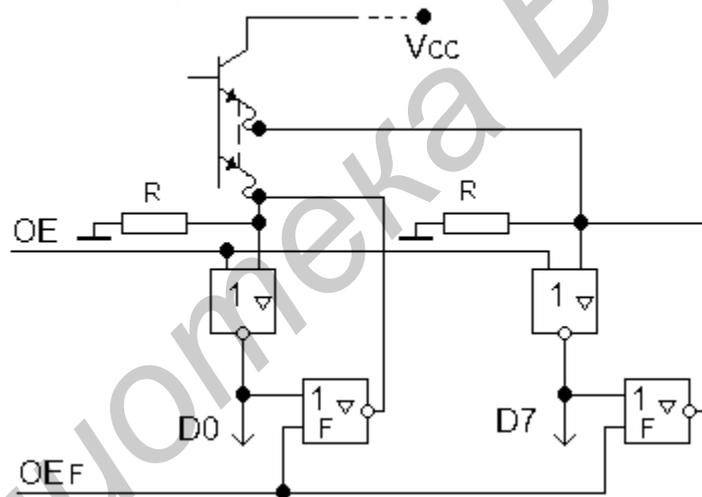


Рис. 6.3. Программирование пользователем ЗУ.

F – относится к программатору.

Для пережигания перемычки на них подают токи в 10 мА в виде серии импульсов. Среди отечественных ППЗУ ведущая роль принадлежит серии К566. Информационная емкость до 640 Кб, время доступа по адресу 90 нс. Уровень интеграции ППЗУ ниже по сравнению с ПЗУ, но простота программирования пользователем и невысокая стоимость определили широкое распространение ППЗУ.

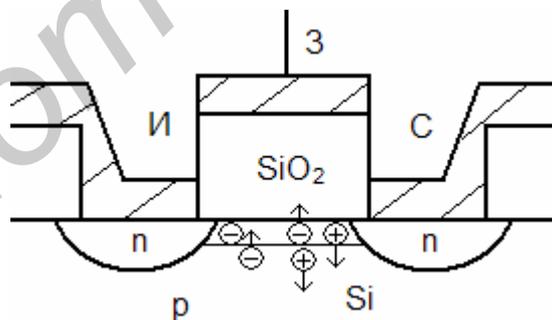
6.4. Классификация СБИС программируемой логики по типу программируемых элементов.



Простейший ключ на биполярном транзисторе:

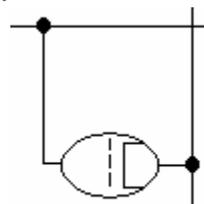


Если включать такой ключ в СБИС ПЛ, то схема будет энергозависима. Ключ на транзисторах МОП с каналом n-типа.



Как только на затворе низкий потенциал ток исток-сток исчезает, отсюда схема энергозависима.

- 10^{23} см^{-3} – металл;
- 10^{18} см^{-3} – полупроводник;
- 10^{12} и меньше см^{-3} диэлектрик.



Но можно решить эту проблему.

Это полевой транзистор с экранированной пластинкой на затворе, вследствие чего появляется паразитная емкость.

Это используют в следующих транзисторах:

МНОП (металл – нитрид – оксид – полупроводник):

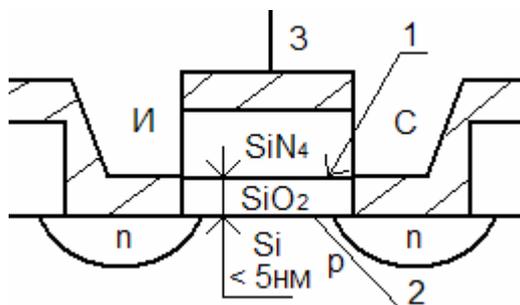


Рис. 6.4. Структура МНОП транзистора.

В диэлектрике SiO₂ находятся оборванные связи, на которых скапливается заряд. Подавая напряжение на затвор можно менять этот заряд и переместить его в 1. Напряжение при переборке 20В. Информация хранится до года. Пример транзистора МНОП - серия КР558.

ЛИЗ МОП (ЛИЗ – лавинная инжекция заряда):

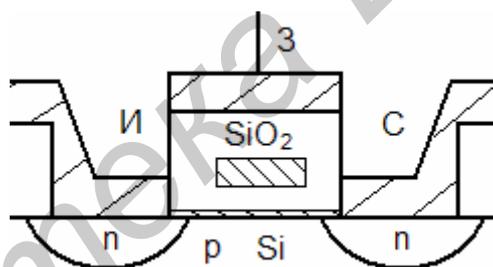


Рис. 6.5. Структура ЛИЗ МОП транзистора с двойным затвором.

В SiO₂ вносят проводящую область, она изолирована – это плавающий затвор. Если в плавающий затвор загнать электроны, то он будет экранировать управляющий затвор. Нет заряда в плавающем затворе – нет экрана и наоборот. Для перепрограммирования можно стягивать заряды из плавающего затвора в Si. При этом используют напряжение 20-25 В, переключение происходит где-то на 5 В.

Емкость определяется длиной затвора, поэтому затвор делают узким (0,13 мкм).

Перезапись информации происходит двумя способами.

Ультрафиолетовое стирание.

Поликремний и диоксид кремния прозрачны для ультрафиолета и для стирания действуют ультрафиолетом 20-30 мин., и степень интеграции больше: 256 Кбит – западные, 64 – наши. Время доступа 10 нс. Стирается сразу вся информация. Используется специальный корпус.

Электрическое стирание.

Прикладывают напряжение обратной поляризации. Электрическое стирание имеет преимущества: можно стирать не всю информацию, а выборочно (индивидуально для каждого адреса). Длительность процесса стирания значительно меньше, чем для ультрафиолетового стирания, сильно ослабляются ограничения на число циклов перепрограммирования (допускается до миллиона циклов). Кроме

того, перепрограммировать ЗУ можно, не извлекая микросхему из устройства, в котором она работает.

7. Учебно-методическая литература

1. Галкин В.И. Промышленная электроника. – Мн.: Вышэйшая школа, 1989.
2. Гусев В.Г., Гусев Ю.М. Электроника. – М.: Высшая школа, 1991.
3. Гутников В.С. Интегральная электроника в измерительных устройствах. – Ленинград: Энергоатомиздат, 1990.
4. Шило В.Л. Популярные цифровые микросхемы. – М.: Металлургия, 1988.
5. Фролкин В.Т., Попов Л.Н. Импульсные цифровые устройства. – М.: Радио и связь, 1992.
6. Быстров Ю.А., Мироненко И.Г. Электронные цепи и устройства. – М.: Высшая школа, 1999.
7. Бирюков С.А. Применение цифровых микросхем серий ТТЛ и КМОП. – М.: МК, 2000.
8. Мняян М.Г. Физика машинной памяти. – М.: Высшая школа, 1990.
9. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. – М.: Горячая линия - телеком, 2000. – 336 с.
10. Угрюмов Е.П. Цифровая схемотехника. БХВ. – Петербург, 2000.
11. Токхейм Р. Основы цифровой электроники. – М.: Мир, 1998.
12. Кучумов А.И. Электроника и схемотехника. – М.: Гелиос АРВ, 2002.
13. Новиков Ю.В. Основы цифровой схемотехники. – М. Мир, 2001.

8. Контрольные вопросы.

1. Привести пример электронной цепи, реализующей функцию ИЛИ.
2. Привести пример электронной цепи, реализующей функцию И.
3. Привести пример электронной цепи, реализующей функцию НЕ.
4. Перечислить основные законы алгебры логики.
5. Для какого числа переменных справедливо правило де Моргана?
6. Дать определение функционально полной системы логических элементов.
7. Перечислить последовательность синтеза комбинационных логических цепей.
8. Перечислить правила проведения контуров в картах Карно.
9. Перечислить правила построения карт Карно для большого количества переменных.
10. Привести классификацию интегральных схем по быстродействию.
11. Привести классификацию интегральных схем по степени интеграции
12. Что такое коэффициент разветвления по выходу ИС?
13. Какие достоинства и недостатки ДТЛ?
14. Для чего применяются схемы со сложным инвертором в ТТЛ?
15. Пояснить работу многоэмиттерного транзистора.
16. Привести условные обозначения логических элементов И, ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ, И2ЛИ, И2ЛИ-НЕ.
17. Перечислить основные преимущества логических элементов с открытым коллектором.
18. Каким образом осуществляется управление микросхем с тремя устойчивыми состояниями?
19. Каково основное предназначение логического выхода микросхем?
20. С помощью элементов 3И-НЕ реализовать функцию 7И-НЕ.
21. Каковы основные рекомендации для режима неиспользуемых входов в КМОП логике?
22. Каковы основные рекомендации для режима неиспользуемых входов в схемах ТТЛ?
23. Что такое комбинационное цифровое устройство?
24. Построить неполный двоичный шифратор 8-4 на логических элементах.
25. В каком случае приоритетный шифратор идентичен двоичному шифратору?
26. Какие выходы должны быть у микросхемы приоритетного шифратора, чтобы с её помощью можно было нарастить его размерность?
27. В каком случае дешифратор может использоваться в качестве мультиплексора?
28. Вследствие чего дешифратор со схемами ИЛИ может использоваться для реализации произвольных логических функций?
29. Пояснить суть метода настройки константами УЛМ на основе мультиплексора.
30. Пояснить суть метода расширения алфавита сигналов настройки УЛМ на основе мультиплексора.
31. Реализовать функции 3И, 3ИЛИ, 3И-НЕ с помощью УЛМ на основе мультиплексора.
32. С помощью двух полусумматоров построить полный сумматор.
33. Реализовать полный сумматор на логических элементах.
34. Записать числа от 0 до 9 в коде Айкена.
35. Объяснить алгоритм построения кодопреобразователя для семисегментной индикации.

36. Что такое последовательностное цифровое устройство?
37. Чем автомат Мили отличается от автомата Мура?
38. Постройте RS-триггер на логических элементах ИЛИ-НЕ.
39. Постройте RS-триггер на логических элементах И-НЕ.
40. Приведите схему JK-триггера на логических элементах И-НЕ.
41. Чем асинхронный триггер отличается от синхронного.
42. Приведите временную диаграмму работы синхронного JK-триггера с прямым динамическим входом.
43. Приведите временную диаграмму работы синхронного JK-триггера с инверсным динамическим входом.
44. Поясните работу синхронного MS-триггера.
45. Какие значения приписывают функции переходов?
46. Каковы отличия между сдвиговым регистром и регистром памяти?
47. Каковы отличия между тактируемым и стробируемым регистрами?
48. Что такое коэффициент пересчёта счётчика?
49. Чем характеризуется быстроедействие счётчика?
50. В чём принципиальное отличие суммирующего счётчика с последовательным и со сквозным переносом?
51. Объясните работу ЦАП на основе резисторных матриц R-2R.
52. Объясните работу АЦП последовательных поразрядных приближений.
53. Объясните внутреннее устройство АЦП параллельного преобразования.
54. В чём заключается основное отличие энергонезависимой от энергозависимой памяти?
55. Укажите основные применения масочных ЗУ.
56. Поясните процесс программирования ППЗУ с плавкими перемычками.
57. Поясните процесс программирования ПЛИС с ЛИЗМОП транзисторами.
58. Поясните суть ультрафиолетового стирания памяти.
59. Поясните суть электрического стирания памяти.

9. Лабораторные работы

1. Исследование работы основных логических элементов.
2. Исследование работы шифратора и дешифратора.
3. Исследование работы мультиплексора и демультимплексора.
4. Исследование работы сумматора.
5. Изучение принципов работы и исследования характеристик RS-, T-, и D-триггеров на логических элементах.
6. Исследование принципов работы и основных характеристик четырехразрядного регистра, кольцевого счетчика и сумматора.
7. Исследование работы делителя частоты.
8. Исследование работы ЦАП на основе резисторных матриц R-2R.

Библиотека БГУИР