



**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И
СВЯЗИ**

Кафедра
электроники

ИНТЕГРАЛЬНАЯ И МИКРОПРОЦЕССОРНАЯ СХЕМОТЕХНИКА

Конспект лекций для студентов специальности
5В071600 - Приборостроение

Алматы 2015

СОСТАВИТЕЛЬ: А.М.Ауэзова. Интегральная и микропроцессорная схемотехника. Конспект лекций для студентов специальности 5В071600 - Приборостроение. - Алматы: АУЭС, 2015.- 77 с.

Настоящие конспекты лекций включают в себя краткое содержание теоретического материала по дисциплине «Интегральная и микропроцессорная схемотехника» в рамках дисциплин по выбору для студентов всех форм обучения специальности 5В071600 - Приборостроение.

Ил.-41, табл.-7, библиогр.-7

Рецензент: доц. Б.К. Курпенов.

Печатается по дополнительному плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015г.

© НАО «Алматинский университет энергетики и связи», 2015 г.

Содержание

1 лекция. Микропроцессоры и микрокомпьютеры. Типовая структура микропроцессорной системы.....	4
2 лекция. Системы счисления и арифметика. Булева алгебра и логические схемы.....	9
3 лекция. Теоремы булевой алгебры. Карты Карно.....	15
4 лекция. Логические элементы.....	20
5 лекция. Типовые логические узлы. Триггеры.....	22
6 лекция. Счетчики импульсов.....	27
7 лекция. Счетчики с коэффициентом счета $K_{сч} \neq 2^N$	30
8 лекция. Регистры.....	33
9 лекция. Комбинационные схемы. Дешифраторы и селекторы.....	39
10 лекция. Сумматоры-вычитатели.....	42
11 лекция. Запоминающие устройства. Оперативное запоминающее устройство.....	48
12 лекция. Постоянные запоминающие устройства. Стеки.....	53
13 лекция. Понятие интерфейса.....	59
14 лекция. Архитектура микропроцессоров.....	63
15 лекция. Цифро-аналоговые преобразователи. Аналогово-цифровые преобразователи. Дискретизация и квантование сигналов в микропроцессоре.....	71
Список литературы.....	76

1 лекция. Микропроцессоры и микрокомпьютеры. Типовая структура микропроцессорной системы

Содержание лекции: основные понятия и определения микропроцессорной системы.

Цель лекции: изучить основные блоки микропроцессорной системы, определения и понятия современных микропроцессоров.

Микропроцессоры и микрокомпьютеры.

Одно из достижений технологий полупроводников заключается в создании больших интегральных схем (БИС), т.е. схем, позволяющих размещать большое число транзисторов, скажем 1000 или более, на одной кремневой подложке (на одном «кристалле»). Это называют высокой степенью интеграции. Собственно, высокая степень интеграции и привела к микропроцессорам.

Микропроцессор, грубо говоря, это программируемое логическое устройство, изготовленное на БИС - технологии. Сам по себе он не может решить ту или иную конкретную задачу. Чтобы решить задачу, его нужно запрограммировать и соединить с другими устройствами. В их число обычно входят память и устройства ввода/вывода, нацеленные на выполнение некоторой четко определенной функции, микрокомпьютером или микропроцессорной системой.

Хотя микрокомпьютеры обладают всеми свойствами обычных ЭВМ, замечательная их особенность состоит в относительно низкой стоимости и малом размере. Именно этому они обязаны своей популярностью и успехом. Большие ЭВМ и миникомпьютеры обладают большой вычислительной мощностью, но не для всех приложений эта мощь оказывается необходимой. Микропроцессоры открывают возможность для применения программируемых устройств в тех логических системах, для которых фактор стоимости оказывается важнее, чем скорость и разнообразие вычислений.

Каждый день открываются новые области приложений для микропроцессоров. В настоящее время они используются в контрольно-измерительных приборах, в кассовых аппаратах магазинов, в интеллектуальных терминалах и карманных калькуляторов. Они оказываются жизненно важным элементом в управлении станками, химическими процессами, периферийными устройствами больших компьютеров, в управлении уличным движением, сложными бытовыми электроприборами и автомобилями. Наконец, в области развлечений и досуга благодаря этим устройствам возникло хобби - строить свои собственные компьютеры [1].

Типовая структура микрокомпьютера.

Типовая компьютерная система включает пять функциональных блоков: устройство ввода, память, арифметическое устройство, устройство управления и устройство вывода.

Физические компоненты и схемы, составляющие микрокомпьютер, - это его аппаратура (hardware). Аппаратура способна выполнять только ограниченный набор элементарных операций. Все прочие функциональные возможности микрокомпьютера достигаются программным путем. Программа - это определенным образом организованная совокупность элементарных машинных операций, называемых командами или инструкциями, с помощью которых осуществляется обработка информации или данных. Программы, написанные для компьютера, образуют его программное обеспечение (software).

Программа и данные сначала накапливаются в памяти, куда они поступают через устройство ввода. Затем отдельные команды программы одна за другой автоматически поступают в устройство управления, которое их расшифровывает и выполняет. Для выполнения операции обычно требуется, чтобы данные поступили в арифметическое устройство (АУ), содержащее все необходимое для их обработки схемы. В процессе вычислений или после их завершения полученные результаты направляются в устройство управления (УУ). АУ и УУ вместе обычно называются центральным процессорным устройством (ЦПУ) или центральным процессором (ЦП). Центральный процессор в микрокомпьютерной системе это и есть микропроцессор.

Не только память, но и другие устройства ЭВМ способны хранить информацию. Информация запоминается как содержимое групп двоичных разрядов - битов - на запоминающих устройствах - регистрах. По существу, любую операцию в ЭВМ можно рассматривать как серию передач. Группа двоичных цифр, обрабатываемых одновременно, называется машинным словом, а число двоичных цифр, составляющих слово, называется длиной слова. Слово является базовой логической единицей информации в компьютере. Команды или данные обычно состоят из одного или несколько слов. Типичные микропроцессоры имеют длину слова 4, 8, 12 или 16 двоичных разрядов. В силу особой распространенности слово длиной 8 бит имеет специальное название - байт.

Создание интегральных микросхем, состоящих из 10-15 до 25-100 схемных элементов (компонентов), явилось первым этапом на пути существенного расширения функциональных возможностей электронной аппаратуры и улучшение ее количественных и качественных показателей. Дальнейшее развитие микроэлектроники направлено на создание больших интегральных схем (БИС), состоящих из тысяч и десятков тысяч компонентов.

Количество компонентов N в кристалле полупроводника характеризует степень интеграции k микросхемы. Её определяют по формуле:

$$k = \lg N.$$

В соответствии с этим к первой степени интеграции относят микросхемы, содержащие до 10 компонентов, ко второй степени интеграции - от 11 до 100 компонентов, к третьей степени интеграции - от 101 до 1000 и т.д.

С появлением БИС началось реальное слияние процесса создания интегральных компонентов с производством электронной аппаратуры.

БИС представляют ряд типовых узлов и схем цифровых устройств: счетчики, регистры, дешифраторы и т.д. На их основе реализуются блоки, а также целые электронные устройства.

Микропроцессор (МП), грубо говоря, это программируемое логическое устройство, изготовленное по БИС - технологии.

В конструкцию МП заложена большая гибкость. Сам по себе он не может решить ту или иную конкретную задачу. Чтобы решить задачу, его нужно запрограммировать и соединить с другими устройствами (рисунок 1.1). В их число обычно входят память и устройства ввода-вывода (УВВ).

Вообще говоря, некоторая совокупность соединенных друг с другом системных устройств, включающая МП, память и УВВ, нацеленная на выполнение некоторой четко определенной функции, называется микрокомпьютером или микропроцессорной системой (МПС). Хотя микрокомпьютеры обладают свойствами обычных ЭВМ, замечательная их особенность состоит в относительно низкой стоимости и малом размере. Именно этому они обязаны своей популярностью и успехом.

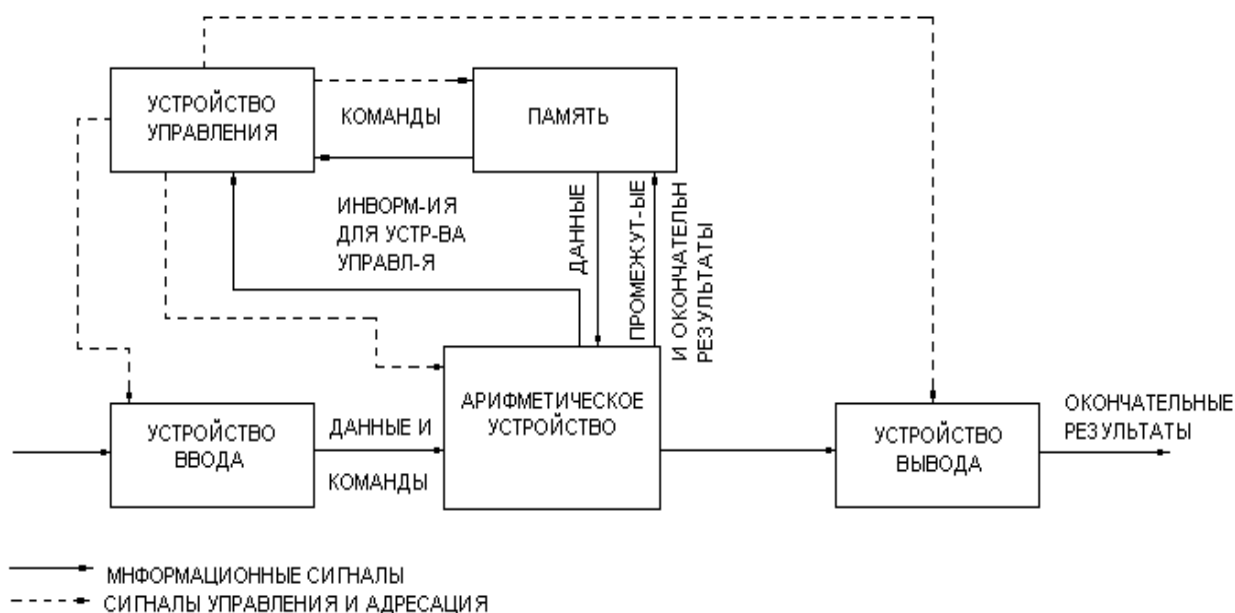


Рисунок 1.1 - Принципиальная организация компьютера

Программа и данные сначала накапливаются в памяти, куда они поступают через устройство ввода. Затем отдельные команды программы одна за другой автоматически поступают в устройство управления (УУ), которое их расшифровывает и выполняет.

Для выполнения операции обычно требуется, чтобы данные поступили в арифметическое устройство (АУ), содержащее все необходимые для их обработки схемы.

В процессе вычислений или после их завершения, полученные результаты направляются в устройство вывода.

Память. Запоминание больших объемов информации происходит в памяти или точнее, в запоминающем устройстве (ЗУ). Этот функциональный блок компьютера подразделяется на подблоки, называемые регистрами, и каждый из которых способен хранить одно машинное слово. Каждый такой регистр, или ячейка памяти, имеет свой адрес.

Адрес - это просто целое число, однозначно идентифицирующее ячейку.

Слово, хранящееся в ячейке, называют содержимым этой ячейки.

Итак, как данные, так и программа хранятся в памяти. Это важное обстоятельство приводит к двум основным концепциям проектирования компьютера.

Первая заключается в том, что компьютер имеет два отдельных и четко различающихся вида памяти. Программа находится всегда в одной памяти (ЗУП - запоминающее устройство памяти), а данные в другой (ЗУД - запоминающее устройство данных).

В соответствии со второй концепцией различие между программной памятью и памятью для данных не проводится. В них программа может размещаться в любом месте общей памяти, и задача программиста - следить за тем, чтобы данные и программа обрабатывались по - разному.

Преимущество второй концепции в возможности трактовать программу как данные, что позволяет компьютеру изменять свои собственные команды. И во втором случае возможно два вида памяти:

- память, из которой возможно только считывание (ROM- read- only- memory) или постоянная память;

- память со считыванием и записью (RWM- read- write- memory) или память с произвольной выборкой.

Изменить информацию, однажды записанную в постоянную память, сложно, если вообще возможно.

Память этого типа благодаря своей низкой стоимости используется для хранения программ и постоянных данных; изменяющаяся информация хранится в памяти со считыванием и записью.

Русский аналог: ПЗУ – постоянное запоминающее устройство (ROM); ОЗУ - оперативное запоминающее устройство (RWM).

Арифметическое устройство (АУ). АУ осуществляет обработку данных, выполняя операции вычитания, сложения, сравнения, операции И, ИЛИ над двумя числами (операндами) с выдачей результата по одному выходу. Вид операции задается программными командами.

Обычно главный регистр в АУ называется аккумулятором. В нем, как правило, находится один из операндов перед выполнением операции, и в него же помещается её результат.

АУ часто содержит еще несколько вспомогательных регистров, названных рабочими; они упрощают составление программ.

АУ содержит также признаковые БИТы или флажки. Эти БИТы содержат информацию, характеризующую состояние МП, которая важна для выбора дальнейшего пути вычисления.

Устройство управления (УУ). УУ управляет работой компьютера. Оно автоматически, последовательно по одной, получает команды из памяти, декодирует каждый из них и генерирует необходимые сигналы.

Для того чтобы получить команду из памяти, УУ прежде всего должно знать её адрес. Обычно команды выбираются из последовательных ячеек памяти и их адреса указываются программным счетчиком, находящимся в УУ.

Чтобы иметь возможность декодировать и выполнять текущую команду, ее нужно где-то запомнить. Этой цели в УУ служит регистр команды.

Для того чтобы быть правильно проинтерпретированной УУ, команда должна иметь определенную структуру, которую называют форматом команды. У разных МП форматы команд различны.

Следующая функция УУ - это синхронизация работы отдельных блоков компьютера. Она осуществляется с помощью генератора тактовых импульсов или тактового генератора.

Обработка команды занимает несколько периодов тактового генератора. Команду нужно выбрать из памяти, декодировать и затем выполнить. Выборка, декодирование и выполнение распадаются на несколько временных интервалов. Каждый из этих интервалов, включающих один или более число периодов тактового генератора, представляет собой так называемый машинный цикл. Совокупное время для выполнения одной команды (выборка, декодирование и выполнение) образует командный цикл.

Код операции - это совокупность двоичных цифр, которая однозначно определяет операцию, выполняемую в процессе интерпретации команды.

Адресная часть команды (если она присутствует) указывает на ячейку (например, в памяти), в которую надо обратиться, выполняя команду.

Устройства ввода/вывода (УВВ). Устройства ввода/вывода - это устройства, осуществляющие контакт компьютера с внешним миром и преобразующие информацию с тех языков и тех скоростей, на которых работает компьютер к тем, которые воспринимает человек или другая связанная с компьютером система.

Устройство ввода получает из внешнего мира данные команды, которые поступают в память. Устройство вывода получает вычисленные результаты и передает их человеку, оператору или другой системе. УВВ представляет собой периферийные устройства машины (внешние или граничные устройства). Точки контакта между УВВ и МП называются портами ввода/вывода. Порты имеют свои адреса, так что к одному МП может быть подключено несколько УВВ.

Шины. Отдельные блоки МПС связаны друг с другом с помощью шин.

Шина представляет собой совокупность линий, по которым передается информация от любого из нескольких источников к любому из нескольких приемников. Существуют адресные шины, шины данных и шины управления.

Адресная шина однонаправленная, т.е. информация передается только в одном направлении (от МП к памяти, вводному или выводному устройству).

Шина данных двунаправленная, т.е. информация по ней может передаваться в обоих направлениях, и она служит для передачи данных.

Шина управления состоит из линии, по которой передаются тактовые, синхронизирующие сигналы, а также информация о состоянии (статусе) устройств [1]. Часть линий в управляющей шине однонаправленные, часть двунаправленные. Поэтому на рисунках, как правило, направление этой шины не указано.

2 лекция. Системы счисления и арифметика.

Содержание лекции: системы счисления, кодирование, коррекция.

Цель лекции: изучить основные понятия и арифметические операции систем счисления, изучить основы кодирования.

Системы счисления и арифметика.

Десятичная система счисления - это лишь одна из многих позиционных систем счисления по основанию.

В этих системах используется конечный набор различных символов. Каждый символ называется *цифрой* и обозначает некоторое количество. Число различных символов в наборе называется *основанием*.

Для величин больших тех, которым соответствуют индивидуальные символы, цифры выписывают рядом, образуя *число*. Например: десятичное число 536.4 можно считать сокращенным обозначенным полинома $5 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1}$.

Значит, в позиционной системе счисления по некоторому основанию число N:

$$N = d_{n-1}d_{n-2}d_{n-3} \dots d_1d_0 \text{ } d_{-1}d_{-2} \dots d_{-m},$$

можно считать обозначением полинома:

$$N = d_{n-1}b^{n-1} + d_{n-2}b^{n-2} + \dots + d_{-m}b^{-m}.$$

В этой общей форме:

d_i - цифры, лежащие в диапазоне $0 < d_i < b$;

n - число цифр левее разделительной или позиционной точки;

m - число цифр правее разделительной или позиционной точки;

b - основание системы счисления (таблица 2.1).

Т а б л и ц а 2.1 – Основание и символы систем счисления

Основание	Цифровые символы
2 (двоичная)	0, 1
3 (троичная)	0, 1, 2
8 (восьмеричная)	0, 1, 2, 3, 4, 5, 6, 7
10 (десятичная)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (шестнадцатеричная)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Из упомянутых систем три представляют особый интерес: двоичная, восьмеричная, шестнадцатеричная (таблица 2.2).

Пример:

1011.101₂

372.46₈

C65F.B3₁₆

Индекс - это основание системы счисления.

В случае двоичной системы цифры 0 и 1 называют БИТами, как сокращённое от binary digits (двоичные цифры).

Перевод в десятичную систему.

Это вычисление можно выполнить следующим образом:

1) Записываем число в виде полинома:

$$d_{n-1} d_{n-2} \dots d_1 d_0 . d_{-1} d_{-2} \dots d_{-m} = d_{n-1} b^{n-1} + d_{n-2} b^{n-2} + \dots + d_0 b^0 + d_{-1} b^{-1} + \dots + d_{-m} b^{-m},$$

где b - основание системы, выраженное в десятичной форме, а буквы заменяются десятичными эквивалентами A=10, D =11, C=12 и т.д.

2) Вычисляем значение полинома, пользуясь десятичной арифметикой.

Пример 1:

$$1110.1_2 \rightarrow 14.5_{10}$$

$$1^{2^3} 1^{2^2} 1^{2^1} 0^{2^0} . 1_2^{-1} = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} = 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 + 1 \cdot \frac{1}{2} = 8 + 4 + 2 + 0.5 = 14.5$$

Пример 2:

$$D3F.4_{16} \rightarrow 3391.25_{10}$$

$$13^{16^2} 3^{16^1} 15^{16^0} . 4^{16^{-1}}_{16}$$

$$13 \cdot 16^2 + 3 \cdot 16 + 15 \cdot 16^0 + 4 \cdot \frac{1}{16} = 3328 + 48 + 15 + 0.25 = 3391.25_{10}$$

Т а б л и ц а 2.2 – Системы счисления

10 (десятичная)	2 (двоичная)	8 (восьмеричная)	16 (шестнадцатеричная)
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F

Перевод из десятичной системы.

В процессе преобразования приходится порознь обрабатывать целую и дробную части числа.

Рассмотрим преобразование целого десятичного числа: N_I в систему счисления с основанием b (b - целое положительное число).

Любое число можно записать в системе с основанием b в виде полинома:

$$N_I = d_{n-1} b^{n-1} + \dots + d_1 b^1 + d_0 b^0.$$

Теперь нужно найти цифры: $d_{n-1}; d_{n-2}; \dots d_1; d_0$.
 Для этого разделим обе части на b .
 Получим целое частное и остаток:

$$N_1' = d_{n-1}b^{n-2} + d_{n-2}b^{n-3} + \dots + d_2b^1 + d_1b^0.$$

Остаток(N_1/b)= d_0 .

Таким образом, остаток равен младшей цифре числа в системе счисления с основанием b , то есть d_0 .

Если процесс деления повторить для целого частного N_1' , мы получим снова целое частное:

$$N_1'' = d_{n-1}b^{n-3} + \dots + d_2b^0$$

и остаток:

$$\left(\frac{N_1'}{b} \right) = d_1.$$

Повторяя описанный процесс, получим все цифры d_i (остаток следует каждый раз представлять цифрой в системе счисления с основанием b).

Пример:

$52 \mid 2$	$d_0=0$
$26 \mid 2$	$d_1=0$
$13 \mid 2$	$d_2=1$
$6 \mid 2$	$d_3=0$
$3 \mid 2$	$d_4=1$
$1 \mid 1$	$d_5=1$

$$52_{10} \rightarrow 110100_2$$

Ответ: $52_{10} \rightarrow 110100_2$.

Процедура перевода правильной десятичной дроби в системе счисления с основанием b должно быть несколько иной.

Обозначим через N_F десятичную дробь, соответствующую полиному:

$$N_F = d_1b^{-1} + d_2b^{-2} + \dots + d_{-m}b^{-m}. \quad (2.1)$$

Умножив обе части на b , получим целую часть d_1 и дробную часть N_F' :

$$b N_F = d_{-1} + N_F' \quad (2.2)$$

Целая часть эквивалентна старшей цифре исходной дроби в системе счисления с основанием b .

Повторяя этот процесс, найдем все цифры. Процесс приостанавливается при достижении требуемой точности, иначе процесс может быть бесконечным.

Пример 1:

$$0.6875_{10} \rightarrow 0.1011_2$$

$2 \times 0.6875 = 1.3750$	$d_{-1}=1$
$2 \times 0.3750 = 0.750$	$d_{-2}=0$
$2 \times 0.750 = 1.50$	$d_{-3}=1$
$2 \times 0.5 = 1$	$d_{-4}=1$

Пример 2:

$$0.8435_{10} \rightarrow 0.D7EF9_{16}$$

$16 \times 0,8435 = 13.496$	$d_{-1}=D$
$16 \times 0.496 = 7.936$	$d_{-2}=7$
$16 \times 0.936 = 14.976$	$d_{-3}=E$
$16 \times 0.976 = F.616$	$d_{-4}=F$
$16 \times 0.616 = 9.856$	$d_{-5}=9$

Применение двоичной системы позволяет существенно уменьшить общее количество аппаратуры и создает большие удобства в эксплуатации цифровых устройств, так как для представления в цифровом устройстве одного разряда двоичного числа требуется компонент с двумя устойчивыми состояниями (например, триггер), а для представления одного разряда десятичного числа - более сложный компонент с десятью устойчивыми состояниями. Этим объясняется преимущественное применение двоичной системы.

Код «8421» взвешенный, так как отражает значение весовых множителей, приписываемых соответствующим БИТам.

Код с избытком «3» не взвешенный, получается прибавлением комбинации $0011=3_{10}$ к каждой цифре в коде «8421».

Достоинство этого кода: замена нулей на единицы и единиц на нули превращает любую цифру x в $(9 - x)$, поэтому этот код еще называют самодополнительным (таблица 2.3).

$$x = 4$$

$$4 = 0111$$

$$(9 - 4) = 5$$

$$5 = 1000$$

Т а б л и ц а 2.3 - Кодирование

Десятичный	Код «8421»	Код с избытком «3»	Код «2 из 5»
0	0000	0011	11000
1	0001	0100	00011
2	0010	0101	00101
3	0011	0110	00110
4	0100	0111	01001
5	0101	1000	01010
6	0110	1001	01100
7	0111	1010	10001
8	1000	1011	10010
9	1001	1100	10100

Код «2 из 5» не взвешенный, обозначается C^2_5 , состоит из пяти БИТов, два из которых единицы, а три - нули. Относится к кодам с обнаружением ошибок. Правильность принятых кодовых комбинаций определяется путем подсчета количества единиц, и если, например, в коде C^2_5 приняты не две единицы, а одна или три, то из этого следует, что в передаче произошла ошибка (все одиночные ошибки могут быть обнаружены, за исключением одномоментного искажения единицы в ноль и нуля в единицу - называется смещением или двойным искажением [1,2]).

Коррекция
Пример 1

$$\begin{array}{r}
 27_{10} = 0010 \quad 0111 \\
 + \\
 36_{10} = +0011 \quad 0110 \\
 \hline
 63_{10} \quad 0101 \quad 1101
 \end{array}$$

Коррекция $6_{10}=0110_2$ (так как используется только 10 из 16 комбинаций).
Коррекция:

$$\begin{array}{r}
 \phantom{6_{10}} + \quad 0101 \quad 1101 \\
 6_{10} + \quad 0000 \quad 0110 \\
 \hline
 63_{10} = 0110.0011
 \end{array}$$

Пример 2

$$\begin{array}{r} 28_{10} = 0010 \quad 1000 \\ + 59_{10} = +0101 \quad 1001 \\ \hline 87_{10} \quad 1000 \quad 0001 \end{array}$$

Коррекция:

$$\begin{array}{r} 0000 \quad 0110 \\ 87_{10} = 1000 \quad 0111 \end{array}$$

Десятичное вычитание:

$$\begin{array}{r} 61_{10} = 0110 \quad 0001 \\ - 38_{10} = -0011 \quad 1000 \\ \hline 0010 \quad 1001 \end{array}$$

Коррекция b_{10} :

$$\begin{array}{r} 0010 \quad 1001 \\ - 0000 \quad 0110 \\ \hline \text{разность } 23_{10} = 0010 \quad 0011 \end{array}$$

3 лекция. Теоремы булевой алгебры. Карты Карно

Содержание лекции: булева алгебра, карты Карно, недоопределенные условия.

Цель лекции: изучить теоремы алгебры логики, графический метод карты Карно.

Теоремы булевой алгебры.

Цель булевой алгебры - описание поведения структуры логических схем.

Логическая схема, которую можно полностью описать таблицей истинности (булевыми выражениями), называется комбинационной схемой.

Комбинационная схема – это такая схема, в которой значения входных переменных в текущий момент времени определяет значения выходных переменных.

Теоремы булевой алгебры

1а $\bar{0} = 1$	1б $\bar{1} = 0$	
2а $x + 0 = x$	2б $x \cdot 1 = x$	
3а $x + 1 = 1$	3б $x \cdot 0 = 0$	
4а $x + x = x$	4б $x \cdot x = x$	закон идемпотентности
5а $x + \bar{x} = 1$	5б $x \cdot \bar{x} = 0$	
6а $\overline{(\bar{x})} = x$	6б	закон двойного отрицания
7а $x + y = y + x$	7б $xy = yx$	закон коммутативности
8а $x + x y = x$	8б $x(x + y) = x$	закон поглощения
9а $x + \bar{x}y = x + y$	9б $x(\bar{x} + y) = xy$	
10а $\overline{(x + y)} = \bar{x}\bar{y}$	10б $\overline{(xy)} = \bar{x} + \bar{y}$	законы де Моргана
11а $(x+y)+z=x+(y+z)=x+y+z$	11б $(xy)z = x(yz) = xyz$	закон ассоциативности
12а $x+yz=(x+y)(x+z)$	12б $x(y + z) = xy + xz$	закон дистрибутивности

Пример 1.

$$\overline{(x_1 + x_3)(x_1 + x_3)(\bar{x}_2 + x_3)} = (x_1 + x_3\bar{x}_3)(\bar{x}_2 + x_3) = (x_1 + 0)(\bar{x}_2 + x_3) = x_1(\bar{x}_2 + x_3)$$

Пример 2.

$$\begin{aligned} \overline{x_1} \overline{x_2} + \overline{x_1} x_2 + x_1 \overline{x_2} + x_1 x_2 + \overline{x_2} x_3 &= \overline{x_1} x_2 + \overline{x_1} \overline{x_2} + \overline{x_1} x_2 + x_1 \overline{x_2} + x_1 x_2 + \overline{x_2} x_3 = \\ \overline{x_2} \overline{x_1} + \overline{x_2} x_1 + x_1 \overline{x_2} + x_1 x_2 + \overline{x_2} x_3 &= \overline{x_2} (\overline{x_1} + x_1) + x_1 (\overline{x_2} + x_2) + \overline{x_2} x_3 = \\ \overline{x_2} \cdot 1 + x_1 \cdot 1 + \overline{x_2} x_3 &= \overline{x_2} + x_1 + \overline{x_2} x_3 = \overline{x_2} + x_1 + x_3 \end{aligned}$$

Другой класс логических схем составляют схемы с внутренней памятью, для них значения выходных переменных определяются не только значениями текущих входных переменных, но и значениями в предыдущий момент времени [1,3].

Карты Карно.

Мы рассмотрели, как с помощью теорем булевой алгебры осуществляют тождественные преобразования булевых выражений. Так как получающиеся при этом выражения эквивалентны, то и комбинационные схемы, описываемые ими, будут эквивалентны.

Возникает вопрос, как определить и как отыскать в том или ином смысле простейшие выражения? Один из методов используют карты Карно.

Карты Карно - это графическое представление таблицы истинности, то есть всех возможных комбинаций переменных.

Карту Карно можно рассматривать как графическое представление всех минтермов заданного числа переменных. Каждый минтерм изображается на карте в виде клетки.

1)

x	y	f
0	0	0
0	1	1
1	0	1
1	1	1

		y	
		0	1
x	0	0	1
	1	1	1

$$f = x + y$$

2)

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

		yz			
		00	01	11	10
x	0	0	1	1	1
	1	0	0	0	1

$$f = \bar{x} + yz$$

3)

w	x	y	z	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

		yz			
		00	01	11	10
wx	00	1	0	1	1
	01	1	0	1	1
	11	1	1	0	0
	10	1	0	0	0

$$f = \bar{y}\bar{z} + wx + \bar{w}y$$

Карта образуется путем такого расположения клеток, при котором минтермы соседних клеток отличаются только значением одной переменной.

В связи с указанным, соседними считаются так же крайние клетки каждого столбца или строки. Символ «1» характеризует прямое значение переменной, а «0» ее инверсное значение.

Минтермы минимизированной функции отличаются единицами в соответствующих клетках карты. Минтермы, не входящие в функцию, отличаются нулями или оставляют клетки пустыми. На основании распределительного закона и аксиом два минтерма, находящихся в соседних клетках, могут быть заменены одним логическим произведением. Если соседними являются две пары минтерма, то такая группа минтермов может быть заменена произведением, содержащим уже на две переменные меньше и т.д. В общем случае наличие единиц в 2^n соседних клетках позволяет исключить «n» переменное.

На карте Карно нужно определить соседние минтермы (клетки) и объединить их в минимальное количество групп соседних минтермов (получение минимальной суммы). Карта Карно позволяет также произвести минимизацию той же функции с помощью макстермов по нулевым значениям функции (то есть получить минимальное произведение). Одна из мер степени сложности булевых выражений определяется числом букв, входящих в выражение (переменные и их инверсии называют литералами).

Пример 1.

		yz				
		00	01	11	10	
wx	{	00	1	1	0	1
	{	01	0	0	0	0
	{	11	0	0	0	1
	{	10	1	1	0	1

Минимальная сумма:

$$f = \bar{x}\bar{y} + \bar{x}z + wyz = \bar{x}\bar{y} + yz \quad 7 \text{ литералов.}$$

Минимальное произведение:

$$\bar{f} = x\bar{y} + \bar{w}x + yz$$

$$f = (\bar{x} + y)(w + \bar{x})(\bar{y} + z) \quad 6 \text{ литералов.}$$

Пример 2.

Минимальная сумма:

$$f = \overline{\overline{w}x\overline{y}} + w\overline{x}\overline{y} + w\overline{y}z + w\overline{y}z + \overline{w}xyz \quad 16 \text{ литералов.}$$

Минимальное произведение:

$$\overline{f} = \overline{w}\overline{x}\overline{y} + \overline{w}x\overline{y} + \overline{w}y\overline{z} + w\overline{y}z + \overline{w}xyz.$$

		yz			
		00	01	11	10
wx	00	1	1	0	0
	01	0	0	1	0
	11	1	1	0	1
	10	0	1	0	1

$$f = (w + \overline{x} + y)(w + x + \overline{y})(w + \overline{y} + z)(\overline{w} + \overline{y} + z)(\overline{w} + x + y + z).$$

Недоопределенные условия.

Рассмотрим еще один вопрос, связанный с существованием функции с недоопределенными условиями. В некоторых случаях известно, что какие-то комбинации на входе появиться не могут, или же, если они появляются, то значение на выходе не существенно. Для таких ситуаций нет необходимости определять выходные значения схемы, то есть функции таблицы истинности.

Соответственно функции, определенные не для всех комбинаций входных переменных, называют частичными функциями.

На карте Карно недоопределенное условие обозначается прочерком «-» в соответствующей ячейке [1].

Такие ячейки могут произвольным образом включаться в группы при построении минимальных сумм и произведений. Любую из них можно включать как в группу единичных, так и в группу нулевых ячеек, более того, их можно вообще никуда не включать.

Пример 3.

		yz			
		00	01	11	10
wx	00	1	-	0	1
	01	-	1	0	1
	11	0	0	-	-
	10	0	0	0	1

Минимальная сумма $f(w, x, y, z) = \bar{y}z + \bar{w}x\bar{y}$.

		yz			
		00	01	11	10
wx	00	1	-	0	1
	01	-	1	0	1
	11	0	0	-	-
	10	0	0	0	1

Минимальное произведение: $f(w, x, y, z) = (y + z)(\bar{w} + y)(\bar{y} + \bar{z})$.

4 лекция. Логические элементы

Содержание лекции: основные логические функции.

Цель лекции: изучить логические функции, основные параметры логических элементов.

К основным параметрам логических элементов (логических микросхем) относятся: функциональные возможности элемента, быстродействие, потребляемая мощность помехоустойчивость.

1. Функциональные возможности логических элементов определяется коэффициентом разветвления «n» по выходу и коэффициентом объединения «m» по входу:

а) под коэффициентом разветвления «n» логических элементов понимают, количество входов аналогичных элементов, которое может быть подключено к его выходу;

б) под коэффициентом объединения «m» – число входов, которое может иметь данный элемент.

Иными словами, коэффициент «n» характеризует нагрузочную способность микросхемы. Чем выше коэффициенты «n» и «m», тем меньшее количество микросхем потребуется. Препятствием к увеличению коэффициента «n» является ухудшение других показателей логического элемента или нарушение нормального режима его работы.

Существующие микросхемы позволяют иметь $n = 4 \div 10$, а с использованием так называемых буферных усилителей $n = 20 \div 50$. Коэффициент $m=2 \div 6$, а с использованием логических расширителей $m=10$ и более.

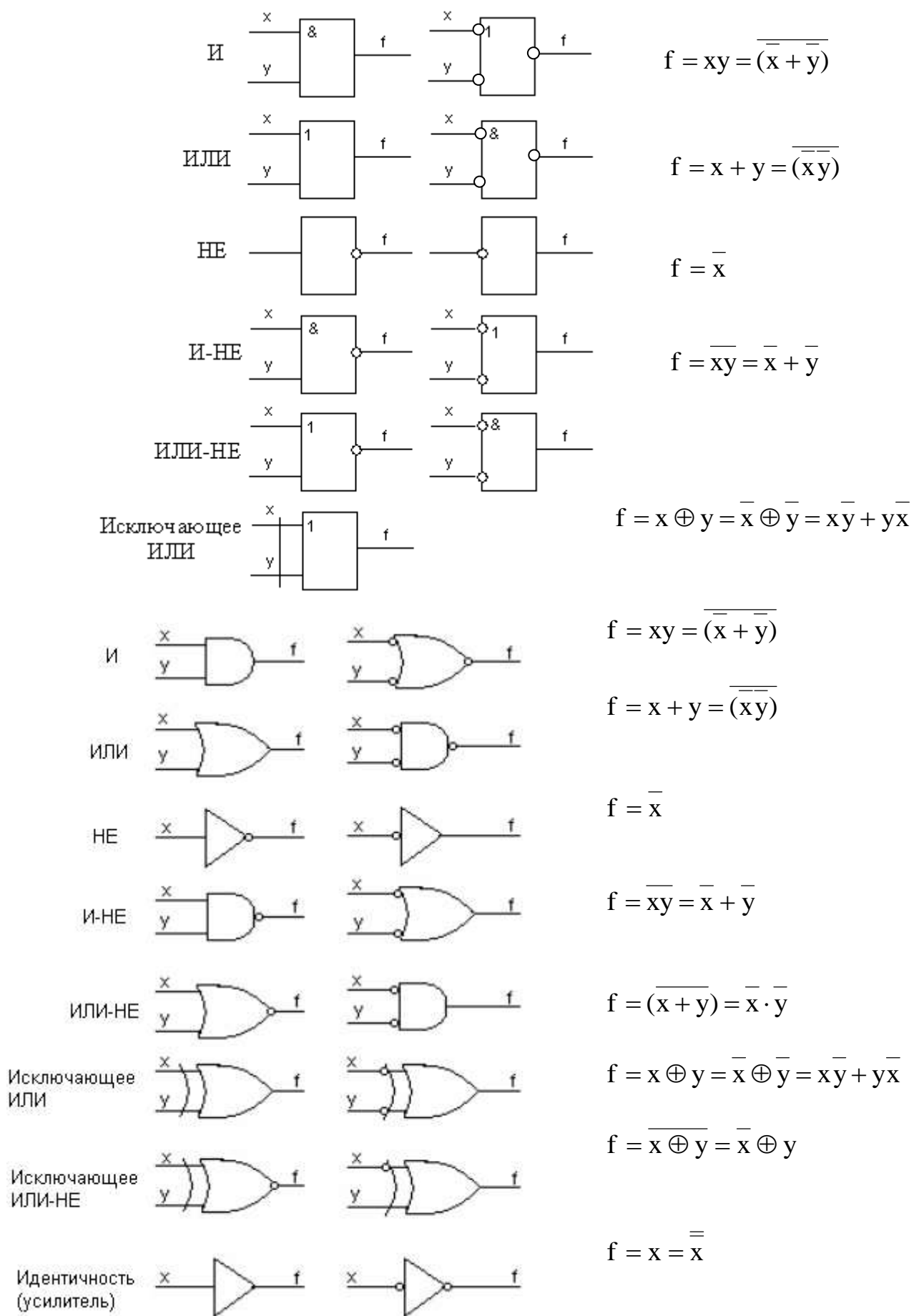


Рисунок 4.1 – Условные обозначения логических элементов

2. Быстродействие характеризует время реакции логического элемента на изменение сигналов на входах. Показателем быстродействия является среднее время задержки прохождения сигнала через логический элемент:

$$t_{3,c} = (t_3^+ + t_3^-) / 2, \quad (4.1)$$

t_3^+ – задержка переключения из состояния «0» в «1»;

t_3^- – задержка переключения из состояния «1» в «0».

Логические микросхемы подразделяют:

- 1) Сверхбыстродействующие $t_{3,ср} < 0,01$ мкс.
- 2) Быстродействующие $t_{3,с} \leq 0.01 \div 0.03$ мкс
- 3) Среднего быстродействия $t_{3,с} < 0.03 \div 0.3$ мкс
- 4) Низкого быстродействия $t_{3,с} \geq 0.3$ мкс

3. Потребляемая мощность от 250 мВт до 1 мкВт. Схемы, потребляющие большую мощность, обладают большим быстродействием [2].

4. Помехоустойчивость характеризует меру невосприимчивости логического элемента к изменению своих состояний под воздействием напряжения помех.

Помехи бывают статические и импульсные, поэтому статическая помехоустойчивость характеризуется максимальным напряжением помехи $U_{п.стат}$, которая может быть подана на вход логического элемента, не вызывая при этом его ложного срабатывания, а импульсная помехоустойчивость характеризуется напряжением импульса $U_{им}$, величина которого зависит от формы и длительности импульса [3].

5 лекция. Типовые логические узлы. Триггеры

Содержание лекции: основные триггерные схемы.

Цель лекции: изучить схемы: RS-триггеры, D-триггеры, T-триггеры.

RS-триггеры.

В зависимости от способа управления различают асинхронные и тактируемые RS-триггеры.

Асинхронные RS-триггеры являются простейшими, но они служат основой триггеров других типов и весьма широко применяются в импульсной и цифровой технике. Для их построения требуется два двухвходовых логических элемента типа И-НЕ (рисунок 5.1) или ИЛИ-НЕ (рисунок 5.2).

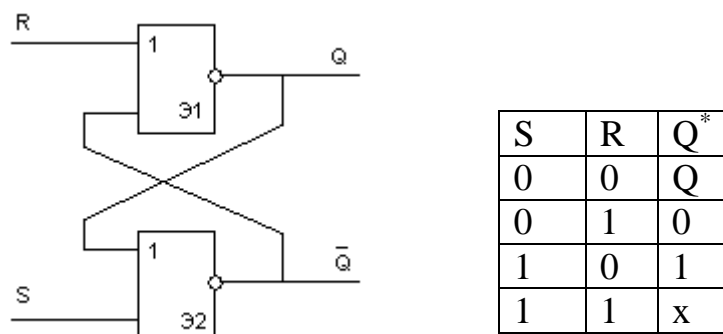


Рисунок 5.1 - RS-триггер на основе логического элемента ИЛИ-НЕ

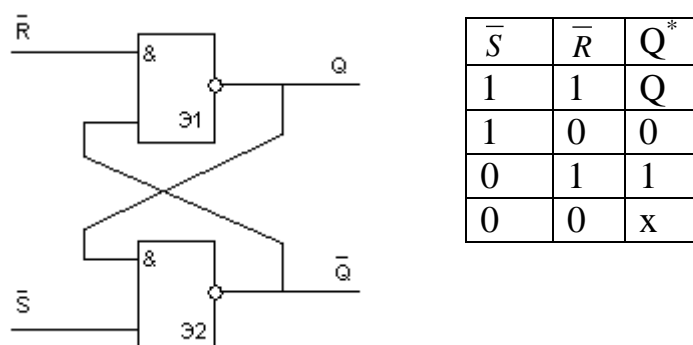
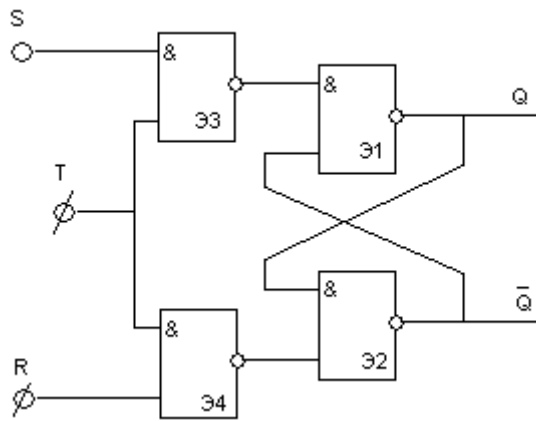


Рисунок 5.2 - RS-триггеры на основе логического элемента И-НЕ

Вход S-установочный (set-установить), а вход R-вход сброса (reset-вновь установить, сбросить). Схема имеет два выхода Q-прямой, \bar{Q} -инверсный. По информационному входу \bar{S} производится установка триггеров в состояние логической «1», а по информационному входу \bar{R} установка (перевод) триггера в исходное состояние логического «0».

Этот триггер управляется не прямыми значениями сигналов, а инверсными значениями входных сигналов. Из двух рассмотренных схем асинхронных RS-триггеров триггер на логическом элементе И-НЕ нашёл наибольшее применение в виду большей распространённости этих элементов в сериях интегральных микросхем.

Переключения в тактируемом триггере возможны лишь при наличии разрешающего сигнала (импульса такта), подаваемого на вход «Т» (рисунок 5.3). Здесь применяются два дополнительных управляющих элемента И-НЕ ($\mathcal{E}_3, \mathcal{E}_4$).



S	R	T	Q*
0	0	0	Q
1	0	0	Q
1	0	1	1
0	1	1	0
0	1	0	Q
1	1	1	X
0	0	1	Q
1	1	0	Q

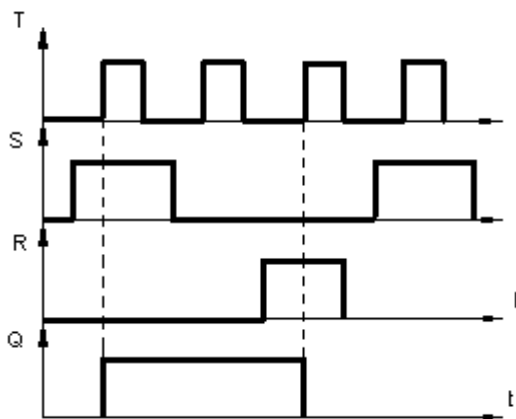


Рисунок 5.3 - Тактируемый RS-триггер на логических элементах И-НЕ

Тактируемые RS-триггеры нашли широкое применение в устройствах цифрового действия для хранения двоичной информации, в течение времени большего её существования в исходном источнике, например, для хранения промежуточной информации, передаваемой от счётчика импульсов и регистра.

D-триггеры.

D-триггеры имеют один информационный вход (рисунок 5.4). На практике наибольшее применение получили одноктактные и двухтактные D-триггеры. Их обозначение (delay- задержка) обусловлено свойством сохранять состояние логической «1», после снятия входного сигнала до прихода очередного тактового импульса. D-триггер широко используют при построении регистров. Предположим, что к моменту прихода входного сигнала D-триггер находится в состоянии логического «0» ($Q=0$). В интервале времени t_1 - t_2 , когда действует входной сигнал D, состояние триггера не изменится, так как при этом $T=0$ и сигналы на входах элементов Э₁, Э₂, $\bar{S} = \bar{R} = 1$. Действие сигнала $T=1$ в момент t_2 обуславливает на выходе элемента Э₃, $\bar{S} = 0$, а на выходе элемента Э₄, $\bar{R} = 1$. Это приводит к переключению

триггера в состояние «1», которое не изменяется до момента следующего синхроимпульса.

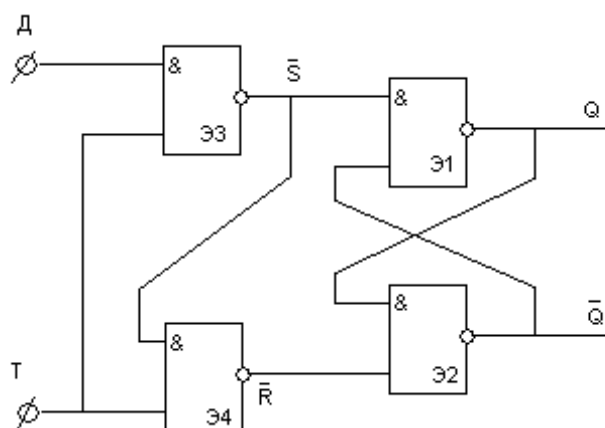


Рисунок 5.4 - Схема однотактного D-триггера, выполненная на основе асинхронного RS-триггера

Двухтактный D-триггер: реализуется на комбинированных логических элементах (2И-ИЛИ). Последовательности тактовых импульсов имеет фазовый сдвиг на 180° .

Записи «1» в триггере производятся при одновременном действии сигналов T_2 , D на входах элемента I_2 , а считывание производится по входу T_1 элемента I_1 .

До момента времени t_1 триггер находится в состоянии «0». При наличии в момент времени t_1 сигналов на обоих входах элемента I_2 происходит изменение состояния триггера ($Q=1$). Это состояние сохраняется до t_2 с помощью логической единицы на обоих входах I_1 .

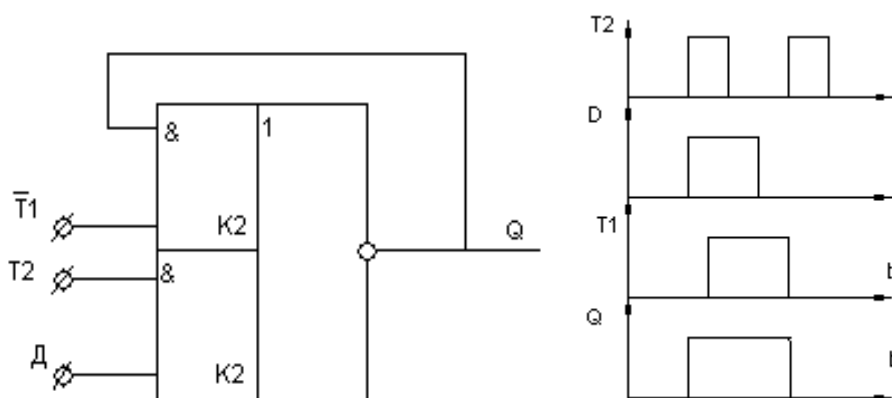


Рисунок 5.5 - Двухтактный D-триггер

В момент t_2 $\bar{T}=0$, триггер переключается в состояние логического «0», так как при $T=0$ сигналах на входах асинхронного триггера $\bar{S} = \bar{R} = 1$.

Появление в момент t_3 сигнала $T=1$ дает $\bar{R}=0$; $\bar{S}=1$ и вызывает переключение триггера в состояние логического «0» [1,4].

T-триггер.

T-триггер: характерным свойством T-триггера является его переключение в противоположное состояние с приходом каждого очередного входного импульса. (tumble-опрокидываться). Его широко применяют в счетчиках импульса и называют - счетным триггером (рисунок 5.6).

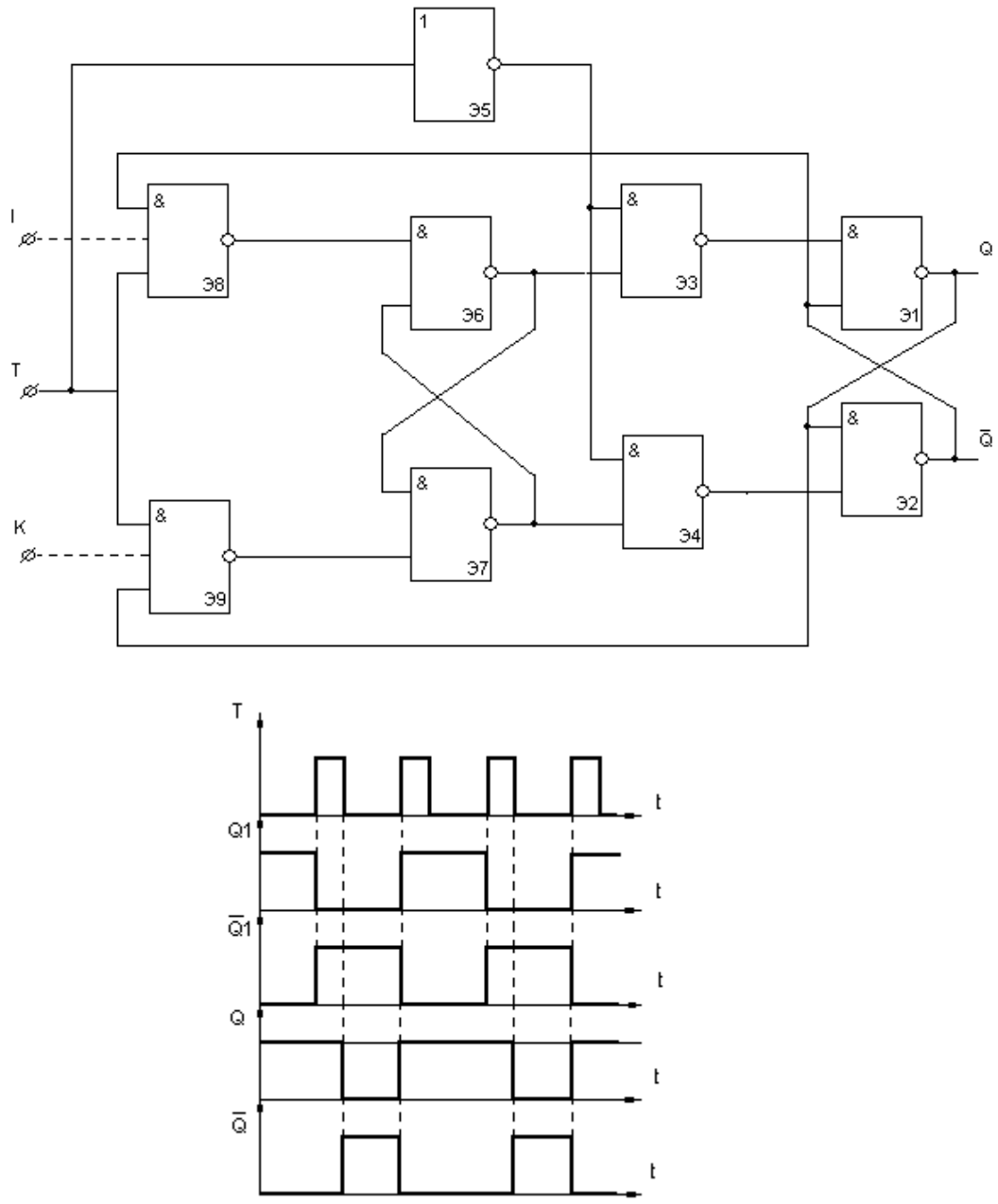


Рисунок 5.6 - T-триггер на базе двух асинхронных RS – триггеров и временная диаграмма схемы

T-триггер выполняют на базе двух асинхронных RS-триггеров (MS-схема), один из которых называют основным (master-хозяин), а другой - вспомогательный (slave) ОТ и ВТ.

По MS-схеме реализуется несколько вариантов T-триггера. Особенность данной схемы - наличие в ней дополнительного инвертора (элемент 5).

Последовательность переключения RS-триггеров, входящих в T-триггер, такая: на этапе фронта входного импульса переключается основной триггер ОТ, а по окончании длительности t_n входного импульса (на этапе среза) переключается ВТ. Иногда этот T-триггер называют триггер с внутренней задержкой. К моменту времени t_1 триггеры ОТ и ВТ находятся в состоянии «1». В момент t_1 $T=1$ и триггер ОТ переходит в «0», однако при этом на входе \mathcal{E}_5 действует «0», а на выходах ВТ ситуация не изменяется. В момент времени t_2 $T=0$ и на выходе \mathcal{E}_5 действует «1».

На обоих входах \mathcal{E}_4 присутствует «1», сигнал на выходе \mathcal{E}_4 станет равен «0», что вызывает переключение в состояние «0» и ВТ.

С момента t_3 действие входного импульса направлено на переключение триггера в состояние «1». В начале (в t_3) переключается ОТ в состояние «1», а затем (в t_4) - ВТ - в «1».

6 лекция. Счетчики импульсов

Содержание лекции: суммирующие, вычитающие и реверсивные счетчики.

Цель лекции: изучить основные схемы простых и реверсивных счетчиков.

Подсчет числа импульсов является наиболее распространенной операцией в устройствах цифровой обработки информации.

В этих устройствах, измеряемый параметр (угол поворота, перемещение, скорость, частота, время, температура и т.д.), преобразуется в импульсы напряжения, число которых в соответствующем масштабе характеризует значение данного параметра. Эти импульсы подсчитываются счетчиками импульсов и выражаются в виде цифр.

По целевому назначению счетчики подразделяют на простые и реверсивные. Простые счетчики, в свою очередь, подразделяют на суммирующие и вычитающие. Суммирующие счетчики предназначены для выполнения счета в прямом направлении, то есть для сложения. С приходом очередного импульса на вход счетчика его показание увеличивается на единицу. Вычитающий счетчик служит для осуществления счета в обратном направлении, то есть для вычитания. Каждый счетный импульс, поступающий на вход вычитающего счетчика, уменьшает его показание на единицу.

Реверсивные счетчики предназначены для выполнения операции счета как в прямом, так и в обратном направлении, то есть они могут работать в режиме сложения и вычитания.

Основными показателями счетчиков являются модуль счета (коэффициент счета K) и быстродействие.

Коэффициент счета определяет число импульсов, которое может быть сосчитано счетчиком. Быстродействие счетчика характеризуется максимальной частотой $f_{сч}$ следования счетных импульсов и связанным с ней временем $t_{уст}$ установки счетчика. Величина $t_{уст}$ определяет время протекания переходных процессов во всех разрядах счетчика с поступлением на вход очередного счетного импульса.

Счетчики импульсов выполняются на основе триггеров. Счет числа поступающих импульсов производится с использованием двоичной системы счисления.

Двоичные суммирующие счетчики с непосредственной связью.

Двоичные счетчики производят счет поступающих импульсов в двоичной системе счисления.

Основным узлом двоичного счетчика является триггер со счетным запуском, осуществляющий подсчет импульсов по модулю 2 (он же является и разрядом).

Многоразрядные двоичные суммирующие счетчики с непосредственной связью выполняются путём последовательного соединения счетных триггеров.

Состояние триггера изменяется на изменение состояния предыдущего триггера, а не на общий внешний сигнал (асинхронные).

Синхронные имеют многовходовый элемент, и все триггеры меняют свое состояние одновременно.

Счетные импульсы подаются на счетный вход первого триггера (рисунок 5.7). Счетные входы последующих триггеров связаны непосредственно с прямыми выходами предыдущих триггеров.

Рассмотрим четырёх разрядный счетчик на счетных (T_1) - триггерах с внутренней задержкой.

Перед поступлением счетных импульсов все разряды счетчика устанавливаются в состояние «0» ($Q_1=Q_2=Q_3=Q_4=0$) подачей импульса «установка нуля».

При поступлении первого счетного импульса первый разряд подготавливается к переключению в противоположное состояние и после окончания действия входного импульса переходит в состояние $Q=1$.

В счетчик записывается число 1. Уровень 1 с выхода Q_1 воздействует на счетный вход второго разряда, подготавливая его к переключению.

По окончании второго счетного импульса первый разряд счетчика переходит в состояние «0», а второй разряд переключается в состоянии «1».

В счетчике записывается число 2-е кодом 0010 и так далее.

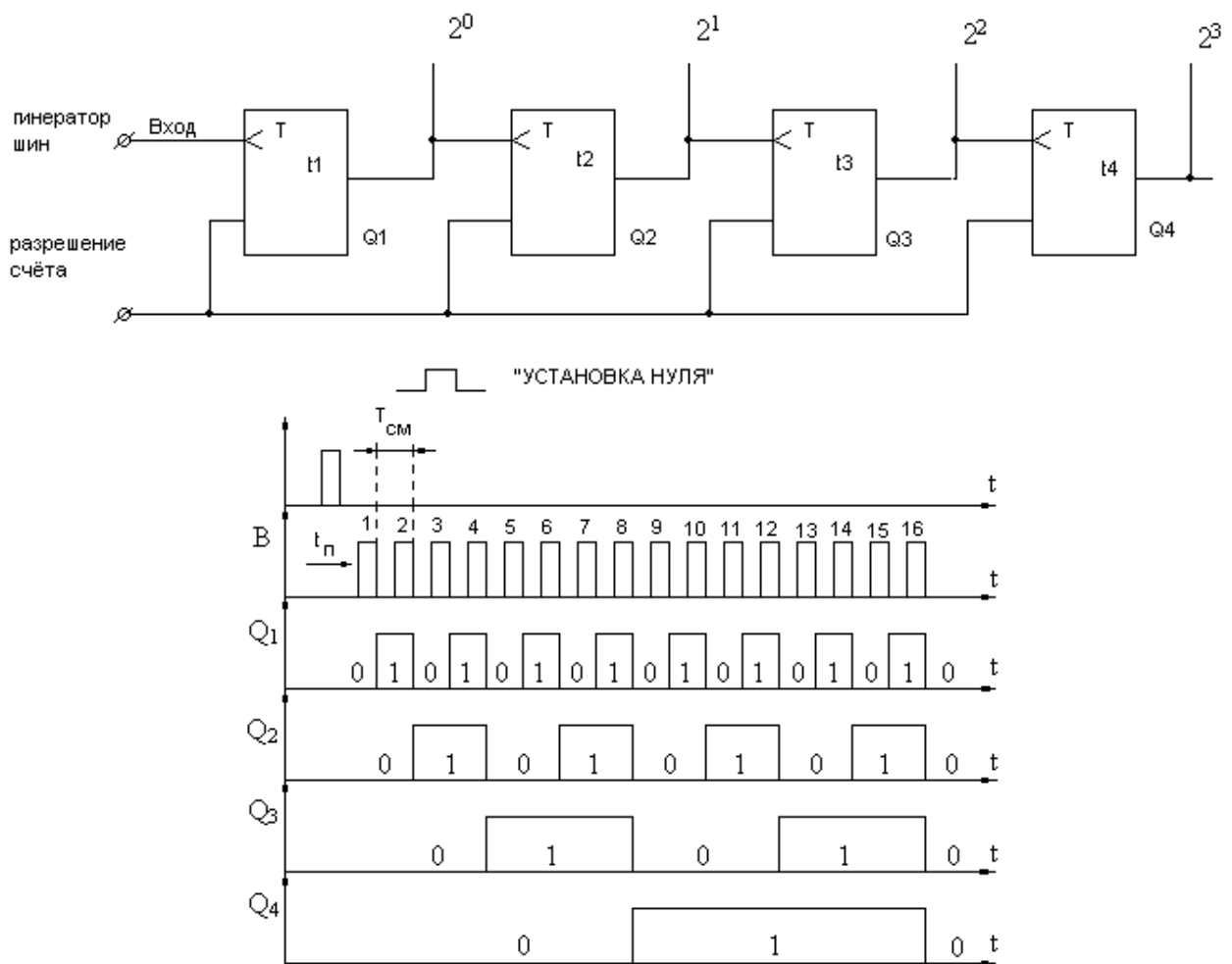


Рисунок 5.7 - Асинхронный двоичный суммирующий счётчик

Модуль счета $K_{сч}=2^N$, где N- число разрядов счетчика.

$$K_{сч}=2^N.$$

В процессе работы двоичного счетчика, частота следования импульсов на выходе каждого последующего каскада в два раза меньше, по сравнению с частотой его входных импульсов.

Это свойство схемы используют для построения делителей частоты.

Выходная и входная частоты связаны соотношением.

$$f_{\text{ВЫХ}}=f_{\text{ВХ}}/K_{сч}.$$

$t_{\text{уст max}}$ - максимальное время установки, в двоичных счетчиках с непосредственной связью характеризуется суммарной задержкой в последовательной передаче информации от младшего к старшему разряду.

$$t_{\text{уст max}}=N t_{з т},$$

где $t_{3\tau}$ - задержка переключения T_t триггера после окончания счетного импульса.

Максимальная частота следования счетных импульсов ограничивается величиной:

$$f_{\text{вх}}=1/(t_{\text{и}}+t_{\text{уст. max}}).$$

При работе счетчика в режиме деления частоты его предельная частота определяется предельной частотой переключения триггера первого разряда [1,4].

$$f_{\text{вх}}=1/(t_{\text{и}}+t_{\text{уст 1}}).$$

Двоично - кодированные десятичные числа.

Популярность двоичной системы счисления в вычислительной технике объясняется простотой и надежностью электронных схем, работающих с числами, имеющими только две цифры. Для человека эта система неудобна, и поэтому иногда желательно, чтобы вычислительная система работала с десятичными числами.

Схема кодирования, при которой каждая десятичная цифра представляется двоично-кодированной группой, называется двоично-кодированной десятичной схемой.

Существует несколько способов кодирования десятичных цифр.

Наиболее известна схема 8421BCD (8421 Binary - Coded Decimal), которую иногда называют взвешенным кодом [1].

7 лекция. Счетчики с коэффициентом счета $K_{\text{сч}} \neq 2^N$

Содержание лекции: типовые схемы счетчиков с коэффициентом счета $K_{\text{сч}} \neq 2^N$.

Цель лекции: изучить основные схемы счетчиков с коэффициентом счета $K_{\text{сч}} \neq 2^N$.

В рассмотренных двоичных счетчиках, коэффициент счёта $K_{\text{сч}}$ связан определенной зависимостью с числом разрядов (триггеров) счетчика.

$K_{\text{сч}}=2^N$ и может быть равен: 2,4,8,16,32,64...

Однако на практике возникает необходимость в счетчиках, коэффициент счёта которых не соответствует указанным значениям.

Например:

$K_{\text{сч}}=3,5,10$ и так далее.

Их можно выполнить на основе двоичных.

Общий принцип их построения основывается на исключении у счетчика с $K_{сч}=2^N$ соответствующего числа «избыточных» состояний. Число запрещенных состояний S определяется разностью.

$$S=2^N-K_{сч},$$

2^N - количество состояний двоичного счетчика;

$K_{сч}$ – коэффициент счёта требующийся.

Например:

При построении счетчика с $K_{сч}=10$ на четырех триггерах требуется исключить шесть состояний 8.

$$S=2^4-10=16-10=6.$$

Для построения счетчика с $K_{сч} \neq 2^N$ можно применить способ (рисунок 7.1):

- 1) Принудительной установки «0» всех разрядов двоичного счетчика.
- 2) Способ принудительного начеса.

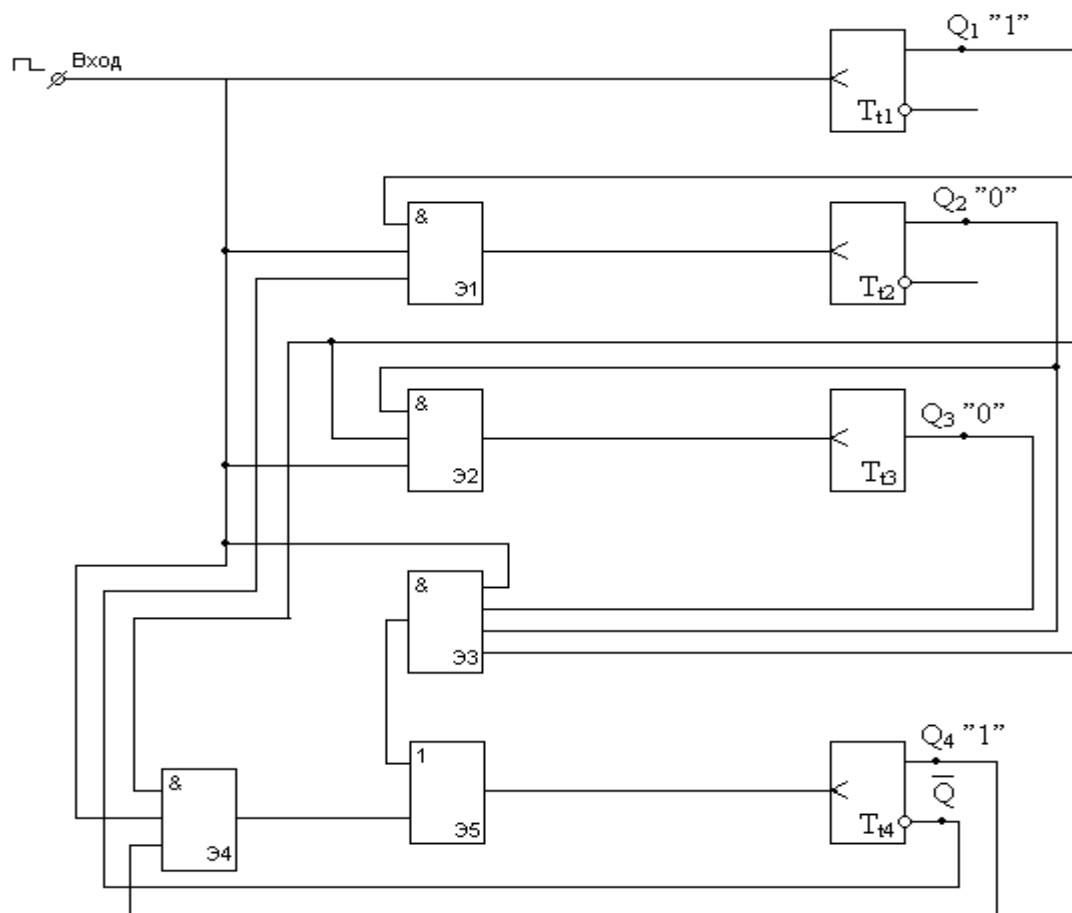


Рисунок 7.1 - Схема декадного счетчика с естественным порядком счета

По первому способу реализуются счетчики с естественным порядком счета, у которых порядок счета такой же, как в двоичных счетчиках.

Отличие заключается в том, что путём введения дополнительных связей, счет заканчивается раньше значения 2^N (таблица 7.1). Так, у счетчика с $K_{сч}=10$, переход разрядов в состояние «0» будет происходить с приходом не шестнадцатого, а десятого счетного импульса (система 16-6).

Счетчик содержит четыре T_t - триггера и пять элементов \mathcal{E}_1 - \mathcal{E}_5 , управляющих переключением триггеров T_{12} - T_{14} .

Т а б л и ц а 7.1 - Состояние триггеров счётчика

Число входных импульсов	Состояние триггеров счётчика			
	T_4	T_3	T_2	T_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Запуск триггера T_{11} осуществляется непосредственно счетными импульсами, а запуск триггеров T_{12} - T_{14} - счетными импульсами, проходящими через элементы \mathcal{E}_1 - \mathcal{E}_4 .

До наступления десятого счётного импульса последовательность переключения триггеров та же, что и у двоичного счетчика.

Действительно, к приходу счётного импульса на один из входов элементов \mathcal{E}_1 - \mathcal{E}_4 подается логический «0» с выхода $Q_1=0$ и все они закрыты для пропускания первого счетного импульса на входы T_{12} - T_{14} .

Первый счетный импульс переключает только триггер T_{11} первого декадного счетчика в состояние «1».

К приходу второго счетного импульса подготовлен к пропуску счетного импульса элемент \mathcal{E}_1 (на левом его входе присутствует «1» с выхода $Q_1=1$, а на правом - «1» с выхода $\overline{Q_4}=1$). Элемент \mathcal{E}_2 закрыт по входу $Q_2=0$, элемент \mathcal{E}_3 - по входам $Q_2=Q_3=0$, а элемент \mathcal{E}_4 - по входу $Q_4=0$.

Второй счетный импульс переключает в состояние «0» триггер T_{11} и в состояние «1» триггер T_{12} .

К приходу третьего счетного импульса элементы \mathcal{E}_1 - \mathcal{E}_4 закрыты по одному из их входов ($Q_1=0$). Третий счетный импульс переключает только

триггер T_{11} . И так далее, с приходом четвёртого-девятого счетного импульсов в соответствии с таблицей [1,2].

После девятого счетного импульса триггеры счетчика принимают следующие состояния $Q_1=Q_4=1; Q_2=Q_3=0$.

Сигналы $Q_2=Q_3=\overline{Q_4}=0$ закрывают элементы $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$, а сигнал $Q_4=1$ подготавливает элемент \mathcal{E}_4 к отпираанию, при поступлении десятого счетного импульса.

Поступающий десятый счетный импульс переводит триггеры T_{11} и T_{14} в состояние «0», обеспечивая нулевое исходное состояние всех разрядов счетчика.

В счетчиках с принудительным насчетом исключение избыточных состояний двоичного счетчика достигается путем принудительной установки отдельных его разрядов в состояние «1» в процессе счета (с произвольным порядком счета).

Счетчики с $K_{сч}=10$ называют десятичными или декадными [4].

Они нашли широкое применение для регистрации числа импульсов с последующим визуальным отображением результата. Декадные счетчики часто включают последовательно (рисунок 7.2).

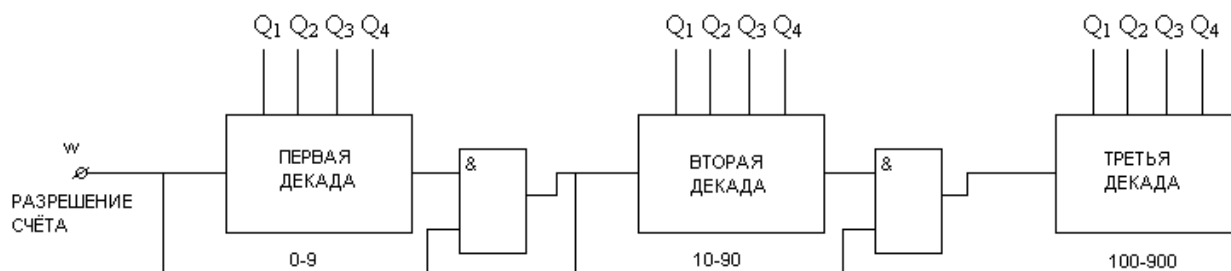


Рисунок 7.2 - Последовательное соединение декадных счётчиков

8 лекция. Регистры

Содержание лекции: параллельные и последовательные регистры.

Цель лекции: изучить основные схемы регистров.

Регистрами называют функциональные узлы, предназначенные для приема, хранения, передачи и преобразования информации.

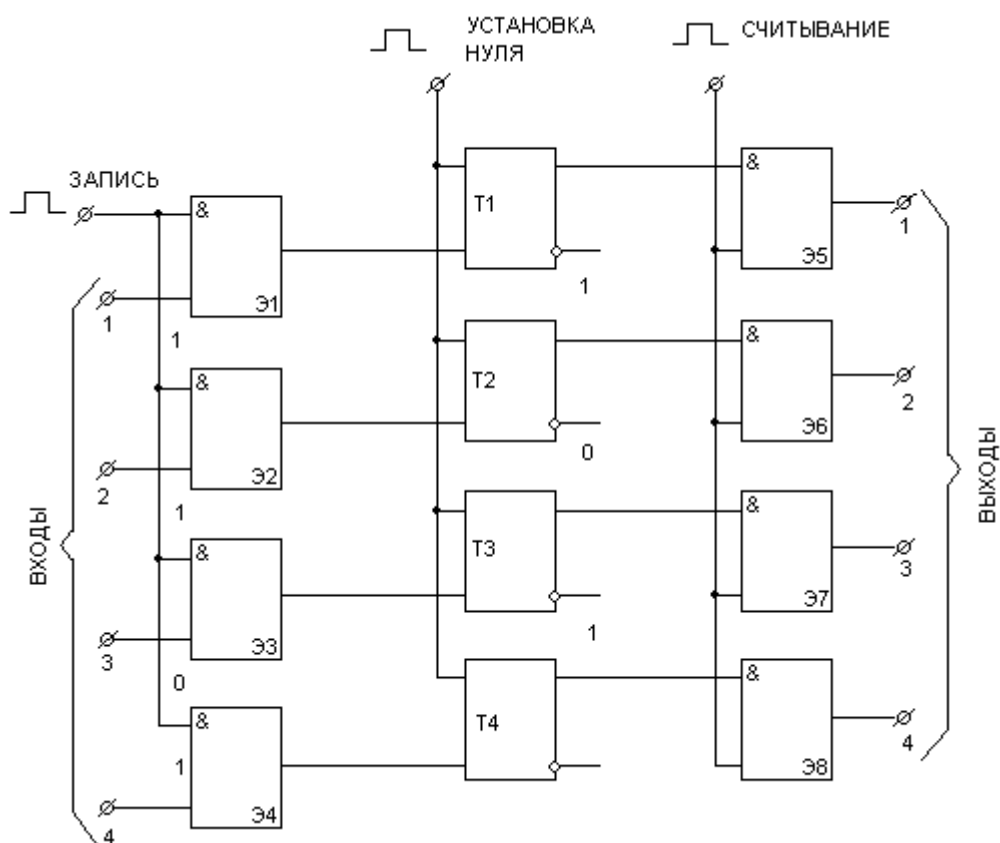
В зависимости от способа записи информации (кода числа) различают параллельные, последовательные, и параллельно – последовательные регистры.

Параллельные регистры.

Запись двоичного числа (слова) осуществляется параллельным кодом, то есть во все разряды регистра одновременно (рисунок 8.1).

Их функция сводится только к приему, хранению и передаче информации. В связи с этим параллельные регистры часто называют регистрами памяти.

Параллельный N- разрядный регистр состоит из N- триггеров, каждый из которых состоит из входов, соответствующих количеству источников информации.



T_1 - триггер младшего разряда; T_4 - триггер старшего разряда; Э₁-Э₄-элементы, управляющие записью информации в регистры; Э₅-Э₈- элемент, управляющие считыванием информации из регистра.

Рисунок 8.1 - Структурная схема параллельных регистров при одном канале информации

Если источник цифровой информации один, то каждый триггер имеет один вход. При двух и трех параллельных каналах информации триггер разряда выполняется на два и три входа.

Запись цифровой информации осуществляется по цепи управления регистром.

Перед записью двоичного числа все триггеры устанавливаются в состояние «0» подачей импульса по входу «Установка нуля». Для записи в

регистр входной информации подают импульс записи, открывающий входные элементы И. Код записывается в регистр (например: $1011_2 - 11_{10}$).

По окончании записи, записанная информация сохраняется, хотя входная информация может изменяться. Для считывания информации подают импульс по входу «Считывание». Тогда на выходные шины регистра передается код числа. При этом число, записанное в регистр, сохраняется. Для получения новой информации описанные операции повторяются. При построении параллельных регистров наибольшее применение получили D-триггеры (триггер - «защелка»), позволяющие выполнять параллельные регистры на малом количестве элементов (рисунок 8.2).

Запись информации по каналу a производится тактовым импульсом T_a , а по каналу b -тактовым импульсом T_b .

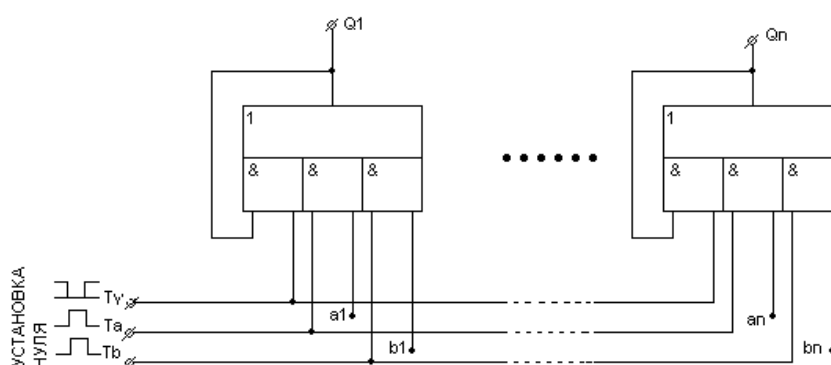


Рисунок 8.2 - Схема параллельного регистра на однофазных D-триггерах (без считывания) с записью по двум каналам (a_1, b_1)

Последовательные регистры.

Регистры сдвига характеризуются записью числа последовательным кодом (рисунок 8.3). Регистр состоит из последовательно соединенных ячеек памяти, состояния которых передаются (сдвигаются) на последующие ячейки под действием тактовых импульсов.

Тактовые импульсы управляют работой регистра.

Если регистры сдвига управляются одной последовательностью тактовых импульсов, то регистры называют одноктактными. Если управляются двумя, тремя и так далее последовательностями тактовых импульсов соответственно, то называются двух-, трехтактными (или в общем случае многотактными).

Частота следования тактовых импульсов обычно неизменна. В многотактных регистрах последовательности тактовых импульсов следуют с взаимным фазовым сдвигом $2\pi/m$, m - количество последовательностей тактовых импульсов.

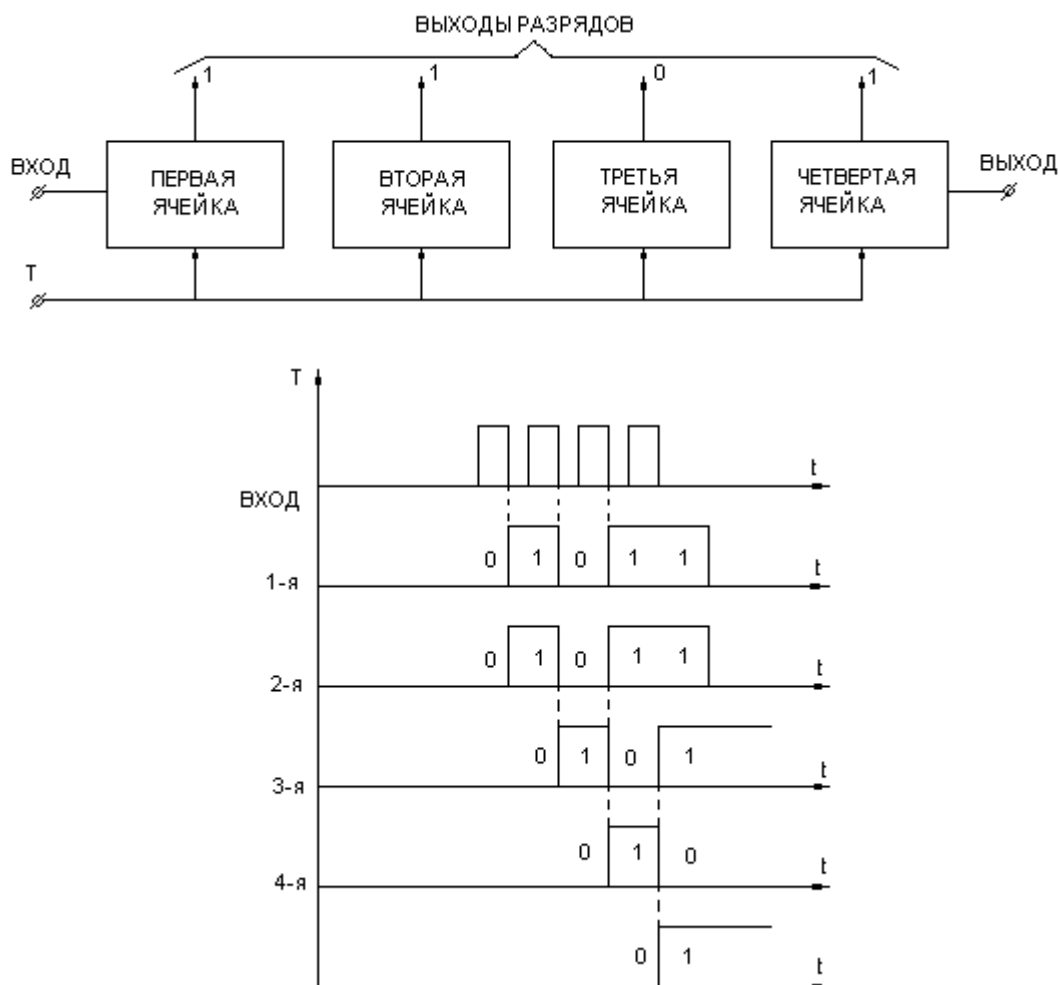


Рисунок 8.3 - Структурная схема последовательных одноклаптивных регистров

Перед записью информации триггер устанавливают в состояние «0». Для этого в отсутствие сигнала на входе подается серия тактовых импульсов с числом импульсов, равным количеству разрядов в регистре. При записи информации одновременно с поступлением кода числа подаются тактовые импульсы. Тактовыми импульсами осуществляется продвижение информации от младшего разряда к старшему.

Первая ячейка относится к младшему разряду регистра. Четвертая - к старшему.

Тактовые импульсы подаются на все триггеры ячеек одновременно.

В результате, после четвертого тактового импульса ячейки регистра принимают состояния, соответствующие коду принятого четырехразрядного числа. (Временные диаграммы показывают запись числа 1011_2 с продвижением кода от младшего разряда к старшему).

Считывание информации из последовательного регистра может быть осуществлено в параллельном или последовательном коде. Для передачи информации в параллельном коде используют выходы разрядов регистра.

Таким образом, последовательный регистр позволяет осуществить операцию преобразования последовательного кода в параллельный.

Считывание информации в последовательном коде реализуется подачей серии тактовых импульсов.

При реализации одноктактных регистров сдвига необходимо учитывать следующее обстоятельство.

В этих регистрах тактовые импульсы воздействуют на перевод в состоянии «0» триггеров всех разрядов одновременно, поэтому в одноктактных регистрах должна быть решена задача разделения во времени (по меньшей мере на длительность тактовых импульсов t_i) операций считывания единицы с триггера каждого разряда, её переписи в триггер следующего разряда. В противном случае перепись единицы в следующий разряд не будет произведена.

Разнесение во времени операций считывания и переписи единицы осуществляют схемными средствами, например: выполнение ячеек на триггерах с внутренней задержкой (R-S_i; J-K_i; D_i-триггера).

В этом случае разделение во времени операции считывания и переписи осуществляется автоматически.

Другим методом построения одноктактных регистров сдвига является использование двухтактного режима их работы.

В каждую ячейку таких регистров входит два триггера (основной и дополнительный). Управление производится одной последовательностью тактовых импульсов.

Разделение во времени операций считывания единицы из предыдущей ячейки и её переписи в последующую ячейку достигается тем, что считывание единицы из основных триггеров осуществляется в момент поступления тактового импульса, а запись единицы в них по окончании тактового импульса [4].

В схемном отношении рассматриваемый метод построения одноктактных регистров сдвига наиболее просто реализуется с применением тактируемых D-триггеров (рисунок 8.4).

В исходном состоянии все триггеры регистра установлены в состояние «0». При наличии сигнала на входе первым тактовым импульсом, через открытый элемент I_2 , записывается единица в основной триггер первого разряда (ячейки). Сигнал $Q_1=1$ передается на информационный вход дополнительного триггера первой ячейки, но поскольку на выходе её инвертора действует сигнал $\bar{T}=0$, элемент I_2 закрыт и дополнительный триггер остается в состоянии «0» ($Q_1^1=0$).

По окончании первого тактового импульса на обоих входах основного и дополнительного триггеров действует сигнал $T=0$, на элементе I_2 дополнительного триггера первой ячейки присутствует единица, он переключается в состояние $Q_1^1=1$.

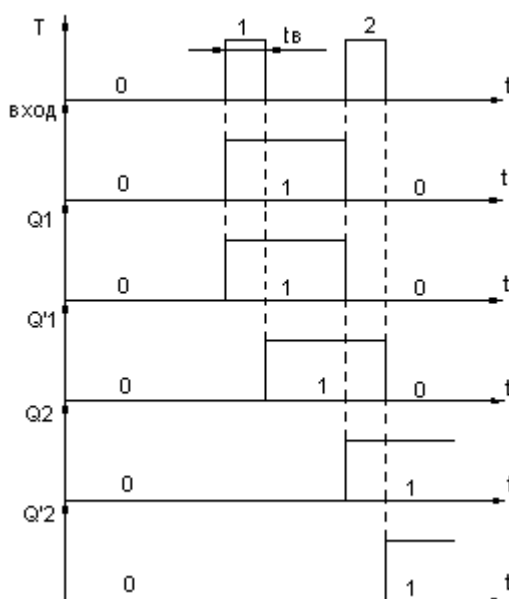
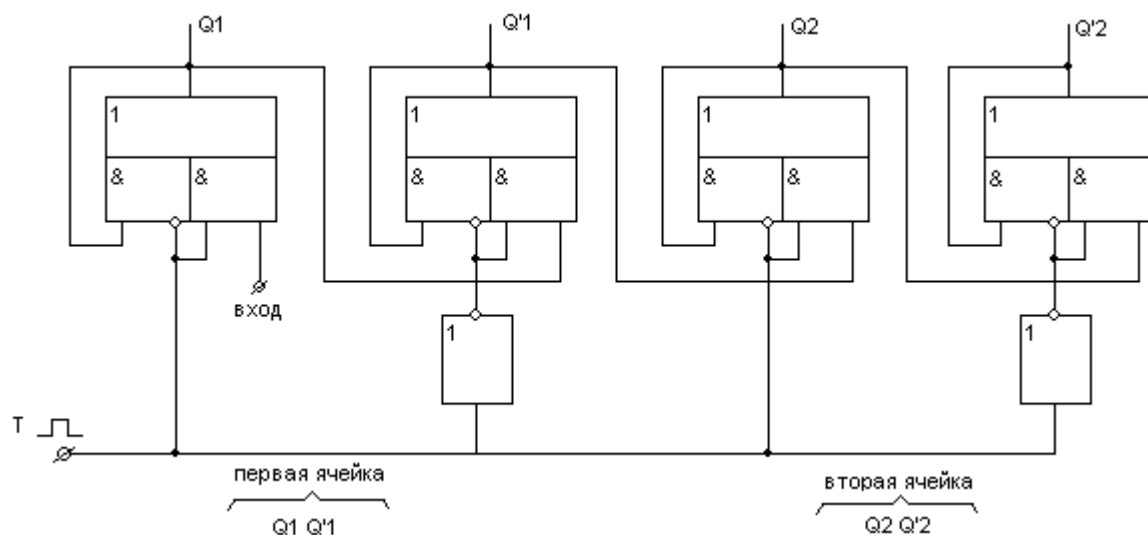


Рисунок 8.4 - Схема одноклоного двухразрядного последовательного регистра на D-триггерах

Состояние «1» основного триггера сохраняется по входам элемента I_1 ($Q_1=1; T=0$), а дополнительного- по входам элемента I_2 ($Q_1=1; \bar{T}=1$).

Если сигнал на входе равен нулю, то поступление второго тактового импульса приводит к заперению элементов I_1, I_2 основного триггера первой ячейки и переключению его в состояние «0». При этом основной триггер второй ячейки по входам элемента I_2 ($Q_1^1=1; T=1$) переключается в состояние $Q_2=1$. На этапе действия второго тактового импульса дополнительный триггер первой ячейки поддерживается в состоянии $Q_1^1=1$ по обоим входам элемента I_1 , а дополнительный триггер второй ячейки остается в состоянии $Q_2^1=0$ по обоим входам его элементов I_1, I_2 .

После окончания второго тактового импульса элементы I_1 , I_2 дополнительного триггера первой ячейки закрываются, что вызывает переключение этого триггера в состояние $Q_1^1=0$. Дополнительный триггер второй ячейки переводится в состояние $Q_2^1=1$ наличием единиц на входах I_2 .

Таким образом, происходит перемещение входной информации во второй разряд регистра.

9 лекция. Комбинационные схемы. Дешифраторы и селекторы

Содержание лекции: основные комбинационные схемы.

Цель лекции: изучить схемы дешифраторов и селекторов.

Дешифраторы.

При разработке различного рода цифровых управляющих устройств часто необходимо решать задачу, когда управляющее воздействие определяется значениями входных сигналов только в данный момент времени и не зависит от их значений в предыдущие моменты времени, то есть, выходной сигнал зависит только от наличия соответствующей комбинации сигналов на входах устройств.

Такие схемы называют комбинационными схемами или автоматами с нулевой памятью.

Принцип проектирования комбинационных схем заключается в следующем. По требуемому алгоритму работы схемы находят управляющее воздействие (функцию) от входных сигналов (переменных). Затем по найденной функции синтезируют логическую схему её реализации.

Задачу нахождения функции связывают с необходимостью построения схемы с минимальным содержанием в ней логических элементов. Для этого функция предварительно проходит стадию минимизации, то есть приведения её к наиболее простому виду.

Примером комбинационных схем является дешифраторы (рисунок 9.1).

Дешифратором называют комбинационную логическую схему, в которой каждая из комбинаций сигналов на входах соответствует сигнал только на одном из его входов.

Широко распространены дешифраторы для преобразования кодов, например, двоичного или двоично-десятичного в десятичный. Они находят применение и в управляющих системах для удобного представления информации.

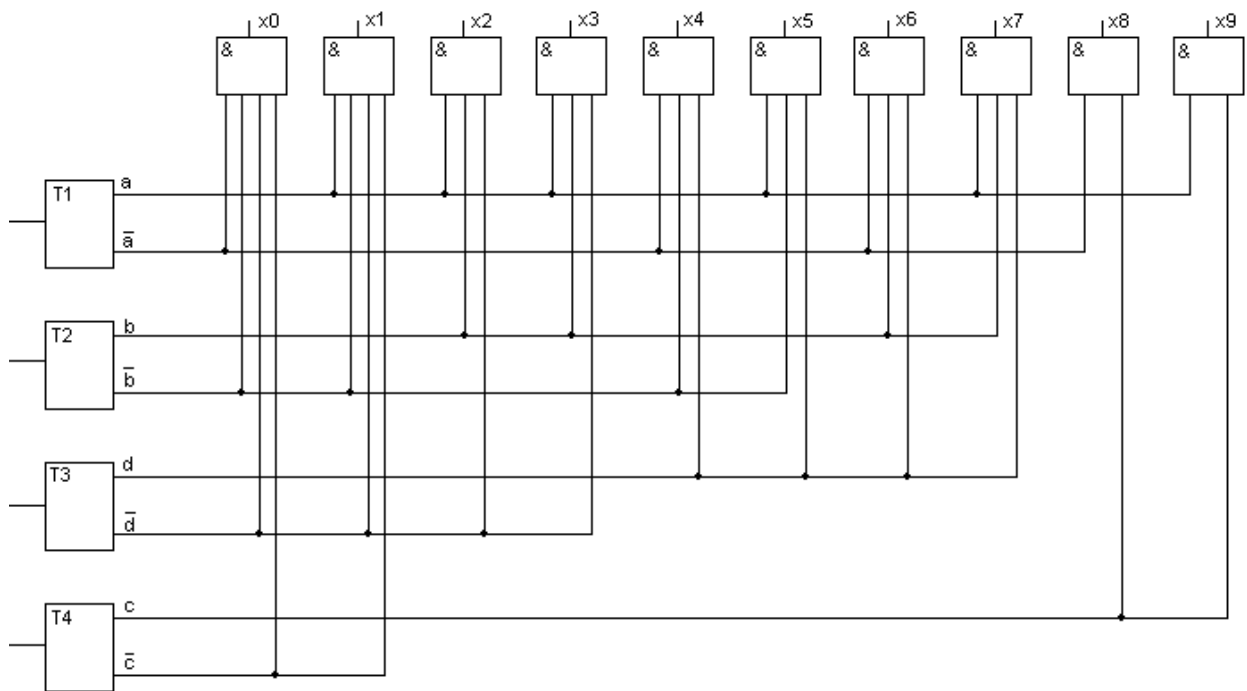


Рисунок 9.1 - Схема дешифратора для перевода показаний двоично-десятичного счётчика в десятичную систему счёта

Дешифраторы других типов.

Кроме рассмотренного дешифратора «1 из $(2^n - 6) = 10$ », существует много других специальных дешифраторов. Например, преобразователь в семисегментный код (рисунок 9.2).

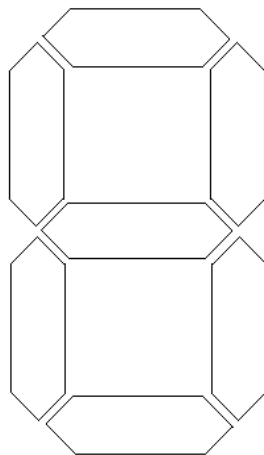


Рисунок 9.2 - Семисегментный индикатор

Этот преобразователь получает десятичную цифру в четырехбитовом двоично - кодированном представлении и формирует значения на семи выходных линиях, которое используется для управления семью сегментами светового индикатора.

Задача преобразователя- выбрать нужные сегменты, чтобы получить изображение десятичной цифры, представленной в коде двоичными цифрами.

Преобразователь является схемой, реализующей в своей главной части заданные в таблице функции. Однако реально в неё обычно добавляются компоненты для того, чтобы обеспечить нужный уровень напряжения на выходах, для управления индикатором и другие вспомогательные функции.

Преобразователь является схемой реализующей в своей главной части заданные в таблице функции (таблица 9.1). Однако реально в неё обычно добавляются компоненты для того, чтобы обеспечить нужный уровень напряжения на выходах, для управления индикатором и другие вспомогательные функции.

Т а б л и ц а 9.1 - Таблица истинности для преобразователя из кода 8421 BCD в семисегментный код

Десятич ная цифра	BCD				Сегменты						
	x ₃	x ₂	x ₁	x ₀	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Селекторы.

Функцию выбора данных от одного из нескольких источников выполняют устройства, изготавливаемые в виде модулей и называемые селекторами (рисунок 9.3).

Как правило, селектор - это комбинационная схема с одной выходной линией для данных, 2^n входными линиями для данных и n управляющими линиями.

Каждая из 2^n входных линий соответствует одной из 2^n возможных комбинаций сигналов на n входных управляющих линиях. При заданной комбинации на входных управляющих линиях значения соответствующей входной линии данных передается на выходную линию данных.

Дешифратор выбирает лишь какой-то один вентиль И (1 из 8), и через него данные с входной линии поступают на выходную схему селектора, можно расширить для многобитовых слов, добавив вентили И и ИЛИ, в соответствии с числом разрядов в слове можно выбирать целое из слов, а не только один бит [1].

Селекторы данных фирмы - изготовители называют также мультиплексорами.

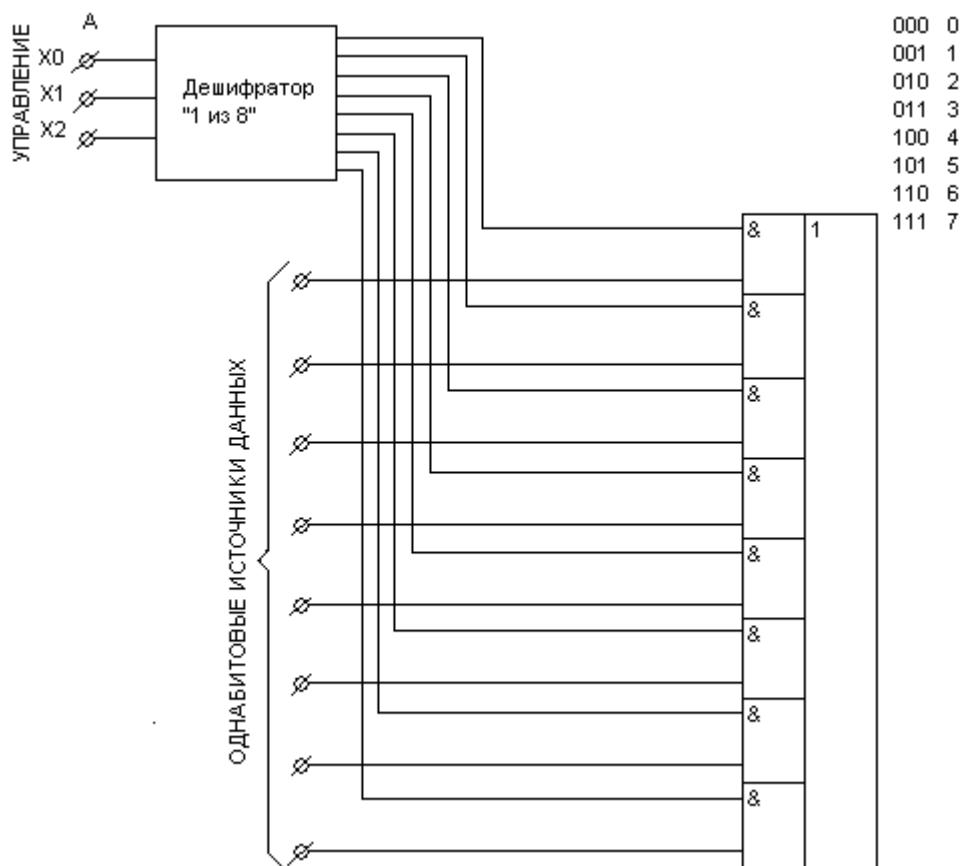


Рисунок 9.3 - Однобитовый селектор с восемью линиями данных

10 лекция. Сумматоры-вычитатели

Содержание лекции: основные схемы сумматоров и вычитателей.

Цель лекции: изучить схемы сумматоров и вычитателей на основе карт Карно и алгебры логики.

Сумматор с последовательным переносом.

Устройства, выполняющие основные операции над числами, сложение и вычитание, называются сумматоры-вычитатели.

Их можно реализовать с помощью комбинационных схем.

Рассмотрим схему сумматора, основанного на поразрядном процессе. Обозначим два складываемых числа через A и B:

$$A = a_{n-1} a_{n-2} \dots a_1 a_0 ;$$

$$B = b_{n-1} b_{n-2} \dots b_1 b_0 .$$

Мы выше уже установили, что при сложении двоичных чисел значения цифр в каждом двоичном разряде д. б. сложены между собой и с переносом из предыдущего разряда.

Если результат при этом превышает 1, то возникает перенос в следующий разряд.

В логической схеме информацию о переносе в разряд i можно представить в виде бита переноса C_i , равного «единице», если перенос из предыдущего разряда есть, и «0»- в противном случае.

Операция, которую нужно выполнить для каждого разряда i , будет тогда заключаться в сложении трёх битов a_i , b_i , c_i , получении значения бита суммы S_i и переноса в следующий разряд C_{i+1} .

Фактически C_{i+1} и S_i представляют собой просто старший и младший разряды соответственно в двухразрядной сумме битов a_i , b_i , c_i (таблица 10.1).

Т а б л и ц а 10.1 - Двухразрядная сумма битов a_i , b_i , c_i

Сумма							
a_i	b_i	c_i	a_i	b_i	c_i	C_{i+1}	S_i
0	0	0	0	0		0	0
0	0	1	0	1		0	1
0	1	0	0	1		0	1
0	1	1	1	0		1	0
1	0	0	0	1		0	1
1	0	1	1	0		1	0
1	1	0	1	0		1	0
1	1	1	1	1		1	1

По функциям, заданным в таблице, можно построить соответствующие логические выражения.

Одно из возможных выражений для S_i - это каноническая сумма минтермов.

$$S_i = \overline{a_i} \overline{b_i} c_i + \overline{a_i} b_i \overline{c_i} + a_i \overline{b_i} \overline{c_i} + a_i b_i c_i.$$

		$b_i c_i$			
		00	01	11	10
a_i	{	0	1	0	1
	}	1	0	1	0

Её можно преобразовать следующим образом

$$S_i = \overline{a_i} (\overline{b_i} c_i + b_i \overline{c_i}) + a_i (\overline{b_i} \overline{c_i} + b_i c_i) = \overline{a_i} (b_i \oplus c_i) + a_i (\overline{b_i} \oplus \overline{c_i}) = a_i \oplus (b_i \oplus c_i) = a_i \oplus b_i \oplus c_i.$$

Аналогично каноническая сумма минтермов для C_{i+1} есть

$$C_{i+1} = \bar{a}_i \bar{b}_i c_i + a_i \bar{b}_i c_i + a_i b_i \bar{c}_i + a_i b_i c_i.$$

		$b_i c_i$			
		00	01	11	10
a_i	0	0	0	1	0
	1	0	1	1	1

$$C_{i+1} = a_i b_i + a_i c_i + b_i c_i.$$

Тогда схема полного сумматора реализуется следующим образом (рисунок 10.1):

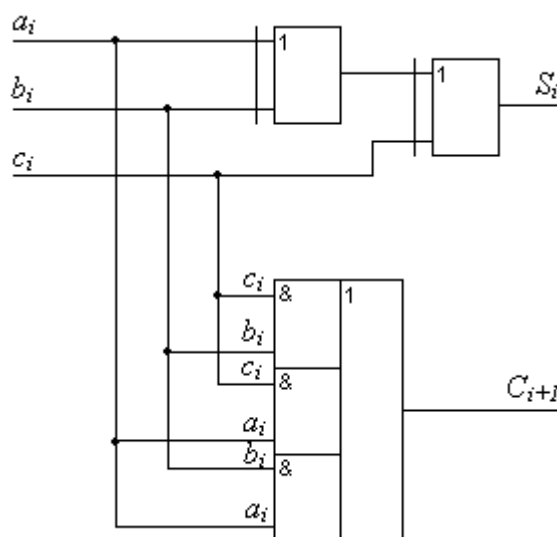


Рисунок 10.1 - Схема полного сумматора

Полный сумматор.

На основе этой схемы можно построить схему для сложения n-разрядных чисел, покаскадно соединив n полных сумматоров.

Разряды двух складываемых чисел подаются на входы a_i b_i , а результат появляется на выходах S_i (таблица 10.2). Последний перенос C_n является старшим разрядом (n+1) разрядной суммы. Входная линия переноса в младший разряд C_0 является ещё одним входом всей схемы. Существуют сумматоры с параллельным переносом, которые работают существенно быстрее, но схемная реализация гораздо сложнее (рисунок 10.2).

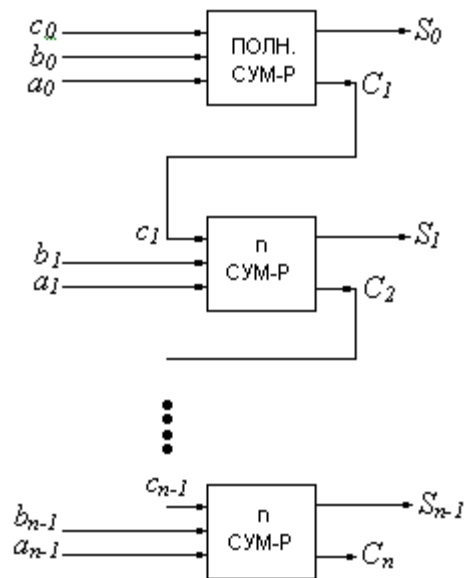


Рисунок 10.2 - Двоичный сумматор с последовательным переносом

Т а б л и ц а 10.2 - Двухразрядная сумма битов a_i , b_i , c_i

a_i	b_i	c_i	c_{i+1}	S_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Вычитатель с последовательным переносом строится по тому же принципу (рисунок 10.3).

Пусть теперь:

$$A = a_{n-1}a_{n-2}\dots a_1 a_0 \text{ -уменьшаемое;}$$

$$B = b_{n-1}b_{n-2}\dots b_1 b_0 \text{ - вычитаемое.}$$

Вычитатель выполняет операцию А-В. Заём в соседнем разряде можно считать отрицательным переносом. Таким образом, разряд переноса C_i при вычитании может обозначать наличие заема из предыдущего разряда (таблица 10.2).

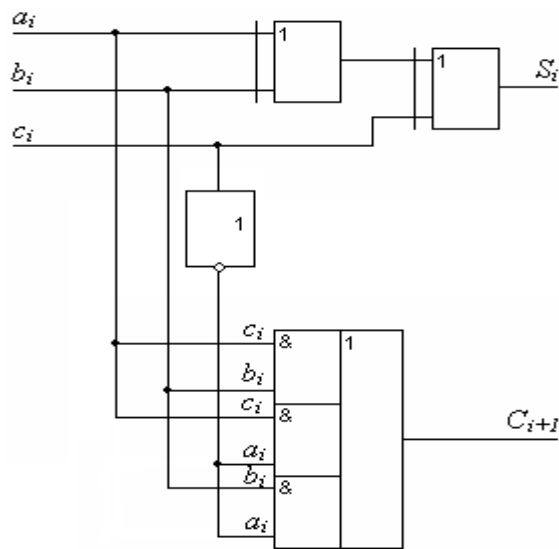


Рисунок 10.3 - Вычитатель с последовательным переносом

Действительно, если сравнить колонки S_i для вычитателя и сумматора, они совпадают. А для C_{i+1} можно составить следующее выражение:

$$C_{i+1} = \bar{a}_i \bar{b}_i + \bar{a}_i c_i + b_i c_i.$$

		$b_i c_i$			
		00	01	11	10
a_i	0	0	1	1	1
	1	0	0	1	0

Легко подметить, что если \bar{a}_i заменить на a_i , это выражение совпадает с выражением для сумматора.

Сумматоры – вычитатели.

Пользуясь сходством булевых выражений для сумматора и вычитателя, можно довольно просто построить комбинированную схему, которая сможет и складывать и вычитать (рисунок 10.4).

Для этого достаточно предусмотреть управляющий вход, с помощью которого избирательно инвертировать a_i в зависимости от требуемой операции. Пусть на управляющей линии K будет значение, равное логическому 0 для сложения и логической 1 для вычитания. Тогда для избирательной инверсии a_i можно применить вентиль ИСКЛЮЧАЮЩЕЕ ИЛИ. Тогда [1,2]:

$$K \oplus a_i = K \bar{a}_i + \bar{K} a_i,$$

если $K=0$, то a_i ;

если $K=1$, то $\overline{a_i}$, что соответствует избирательной инверсии.

Для сумматора: $C_{i+1} = a_i b_i + \overline{a_i} \overline{b_i} + b_i c_i$.

Для выключателя: $C_{i+1} = \overline{a_i} \overline{b_i} + \overline{a_i} b_i + b_i c_i$.

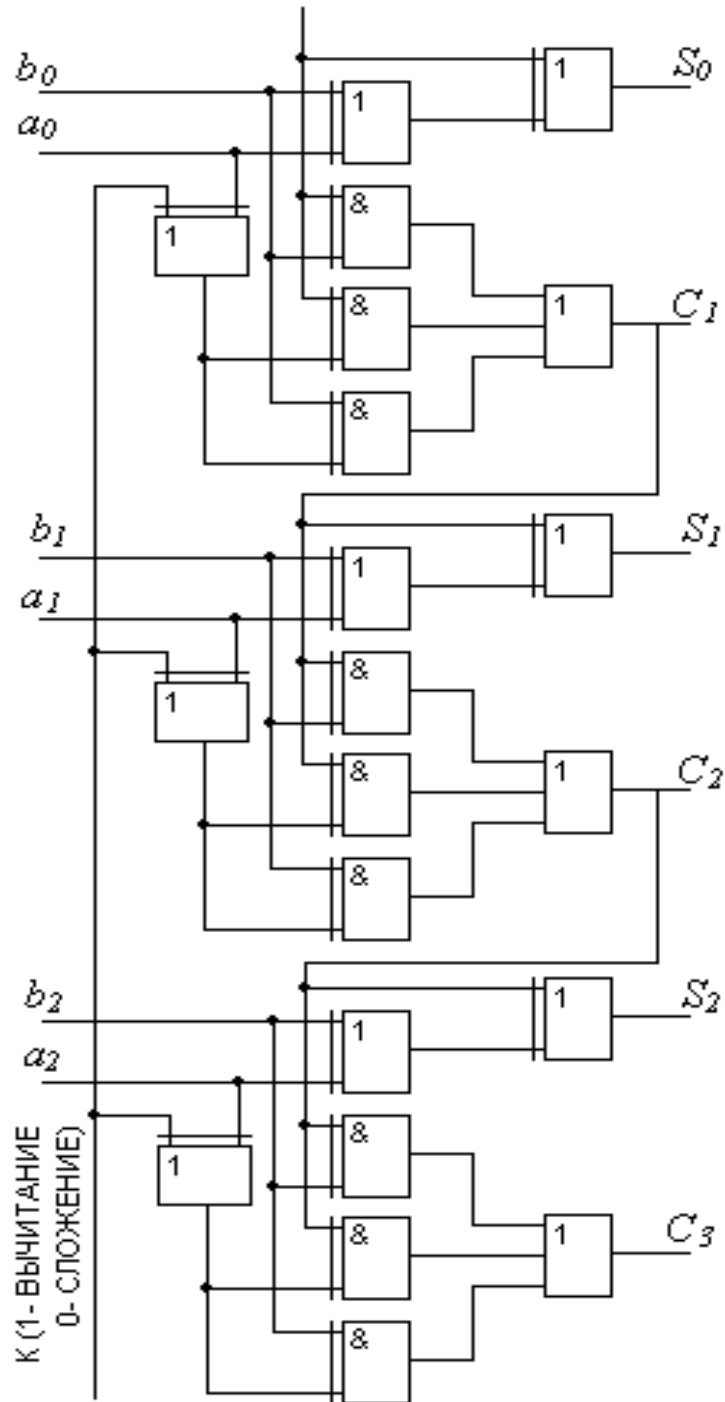


Рисунок 10.4 - Трёхразрядный сумматор - вычитатель, составленный из полных сумматоров – вычитателей

11 лекция. Запоминающие устройства. Оперативное запоминающее устройство

Содержание лекции: типовые структуры запоминающих устройств.

Цель лекции: изучить структуру оперативного запоминающего устройства.

Оперативное запоминающее устройство.

Оперативная память является одним из важнейших элементов компьютера. Именно из нее процессор берет программы и исходные данные для обработки, в нее он записывает полученные результаты. Название «оперативная» эта память получила потому, что она работает очень быстро, так что процессору практически не приходится ждать при чтении данных из памяти или записи в память. Однако содержащиеся в ней данные сохраняются только пока компьютер включен. При выключении компьютера содержимое оперативной памяти стирается. Часто для оперативной памяти используют обозначение RAM (Random Access Memory, то есть память с произвольным доступом).

Трудно недооценить все значение и всю важность этих небольших по своим размерам плат. Сегодняшние программы становятся все требовательнее не только к количеству, но и к быстродействию ОЗУ. Однако до недавнего времени эта область компьютерной индустрии практически не развивалась (по сравнению с другими направлениями). Взять хотя бы видео, аудиоподсистемы, производительность процессоров и. т. д.

Усовершенствования были, но они не соответствовали темпам развития других компонентов и касались лишь таких параметров, как время выборки, был добавлен кэш непосредственно на модуль памяти, конвейерное исполнение запроса, изменен управляющий сигнал вывода данных, но технология производства оставалась прежней, исчерпавшей свой ресурс. Память становилась узким местом компьютера, а, как известно, быстродействие всей системы определяется быстродействием самого медленного ее элемента. И вот несколько лет назад волна технологического бума докатилась и до оперативной памяти. Стали появляться новые типы RAM микросхем и модулей. Встречаются такие понятия, как FPM RAM, EDO RAM, DRAM, VRAM, WRAM, SGRAM, MDRAM, SDRAM, SDRAM II (DDR SDRAM), ESDRAM, SLDRAM, RDRAM, Concurrent RDRAM, Direct Rambus. Большинство из этих технологий используются лишь на графических платах, и в производстве системной памяти компьютера используются лишь некоторые из них.

Организация ЗУ с произвольным доступом.

Память с произвольным доступом – это такое ЗУ, в котором элемент данных, запомненный в ячейке, может быть непосредственно считан. Время, необходимое для выборки данной ячейки, оказывается примерно тем же, что и

для любой другой ячейки. Каждая ячейка содержит фиксированное число запоминающих элементов и имеет свой идентифицирующий номер. Идентифицирующий номер, состоящий из фиксированного числа битов, называется адресом ячейки. Наличие адресов позволяет различать ячейки при обращении к ним для выполнения операции записи и чтения. В общем случае ЗУ с произвольным доступом состоит из нескольких блоков, или модулей. Для полупроводников памяти модуль обычно реализуется в виде отдельных интегральных схем. Состав и функции внешних сигнальных линий выбирается с таким расчетом, в число таких линий входят линии для задания адреса слова, к которому производится обращение, линии, по которым передаются данные в модуль или из модуля, и несколько управляющих линий, позволяющих задать нужную операцию (запись или чтение). Существует довольно много разновидностей модулей с набором внешних сигналов, соответствующих различным типам шин, однако большинство модулей можно отнести к одной из двух основных моделей, показанных на рисунке 11.1. Различаются они только линиями данных. У модели на рисунке 11.1 (а) имеется лишь один комплект линий данных, по которым передаются как поступающие данные при записи, так и выдаваемые данные при чтении. Модуль на рисунке 11.1 (б) содержит два отдельных комплекта линий для поступающих и выдаваемых данных.

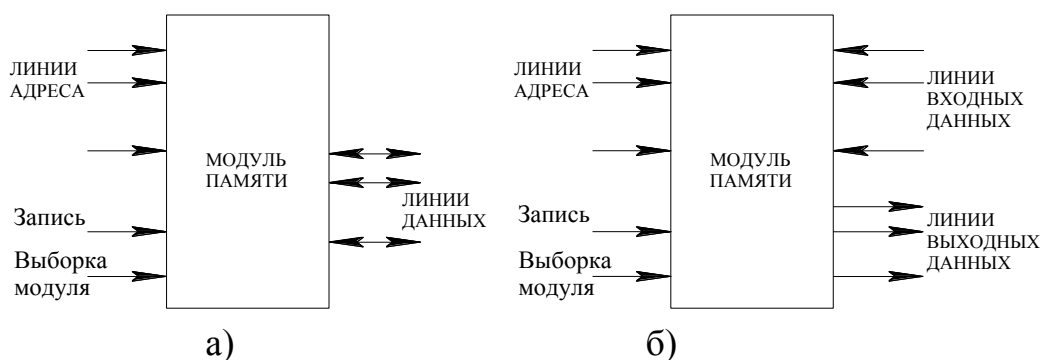


Рисунок 11.1 - Линии внешних сигналов в модулях памяти с произвольным доступом: модуль с двунаправленными линиями данных (а); модуль с отдельными линиями для поступающих и выдаваемых данных (б)

Обе модели имеют комплект адресной линий, сигналы на которых определяют ячейку, к которой осуществляется обращение для считывания или записи данных. В обеих моделях управляющая линия «запись» задает режим записи («запись»=1) или режим чтения («запись»=0). Наконец, управляющая линия «выборка модуля» в обоих случаях либо разрешает выполнение операции чтения/записи в данном модуле («выборка модуля»=1), либо запрещает ее выполнение («выборка модуля»=0).

Для того чтобы модули памяти, соответствующие любой из двух моделей, можно было включать в многоугольную систему ЗУ с шиной

организацией, на линиях выдачи данных (т.е. на линиях данных в модели рисунке (а) и на линиях выходных данных в модели на рисунке (б)) используются вентили, допускающие монтажную логику. Модуль памяти любой модели выдает данные на выходные линии только при чтении; во всех остальных ситуациях этими линиями могут пользоваться другие модели памяти или устройства, подключенные к линиям. Это особенно важно для случая на рисунке (а), поскольку при записи для передачи данных ЗУ должны использоваться те же линии.

Состояние линий выходных данных, когда в модуле не выполняется чтение, конечно, зависит от типа монтажной логики. Например, для монтажного И на схемах транзисторно-транзисторной логики (ТТЛ) с открытым коллектором состояние на свободной линии должно соответствовать логической 1, поскольку в данном случае 0 доминирует над 1 и, следовательно, другое устройство или модуль памяти сможет задать нужное ему состояние на линии. Для монтажного ИЛИ свободное состояние должно быть логическим 0. Для тристабильной монтажной логики свободное состояние с высоким импедансом, эквивалентное отсутствию соединения с линией. В любом случае линия выходных данных оказывается в свободном состоянии, когда либо подана логическая 1 на линию «запись», либо подан логический 0 на линию «выборка модуля».

Внутренняя организация ЗУ с произвольным доступом. Одна из возможных схем, позволяющих выбирать нужную ячейку и осуществлять передачу информации в нее или из нее, представлена на рисунке 11.2. В запоминающем элементе для хранения одного бита слова применяется асинхронный RS-триггер. Помимо триггера, каждый запоминающий элемент содержит вентили для передачи информации между триггером и внутренними линиями данных.

Одна из двух операций, чтение или запись, выполняется одновременно для всех элементов одной «строки». Каждая «строка» представляет собой ячейку для хранения слова и имеет свой адрес. Дешифратор «1 из 2^n » служит для выбора ячейки по заданному адресу. На n входных линиях дешифратора возможны 2^n линий дешифратора называются линиями выборки слова. В зависимости от поданной на вход дешифратора комбинации какая-то одна линия выборки слова получает значение логической 1, а все остальные - логического 0. Каждая линия выборки слова используется как линия, разрешающая операцию чтения или записи во всех элементах строки.

В каждом столбце имеются по две внутренние линии: одна - для передачи данных в память (вход), а другая - из памяти (выход). Состояние линии «выход» определяется состоянием запоминающего элемента в выбранной строке. Для этого выход триггера в каждом элементе логически умножается на сигнал «выборка слова», а логическая сумма всех результатов поступает в линию «выход» (рисунок 11.3).

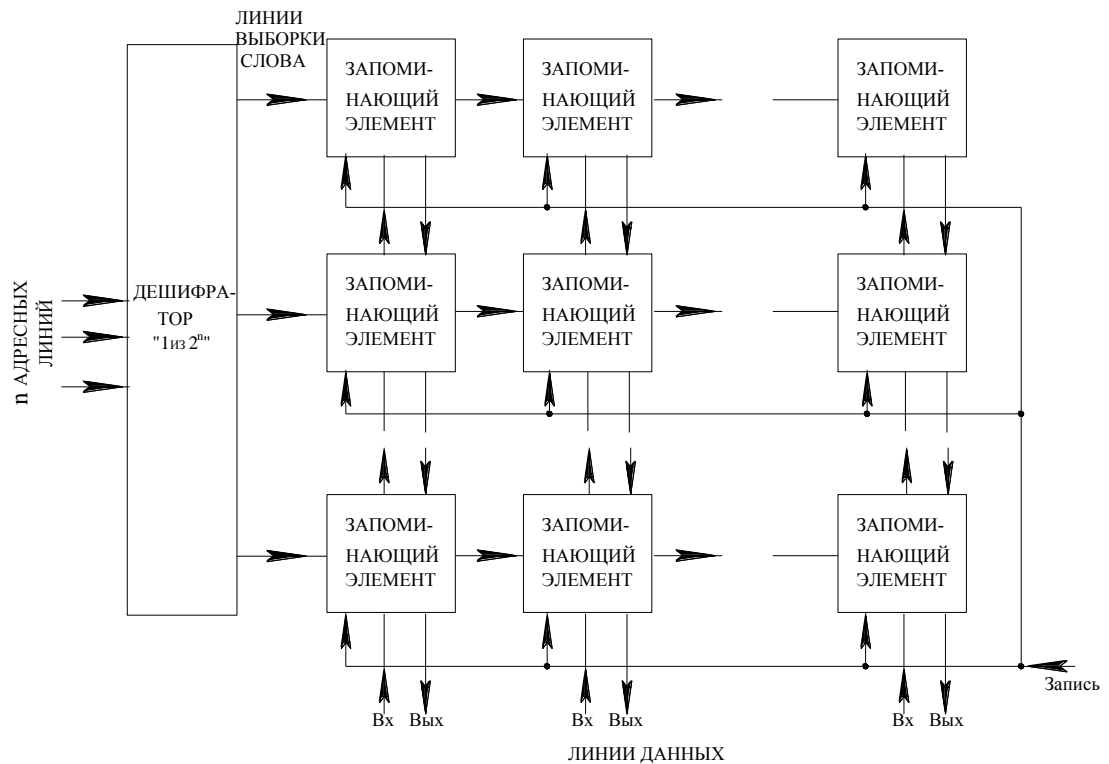


Рисунок 11.2 - Модуль памяти с произвольным доступом

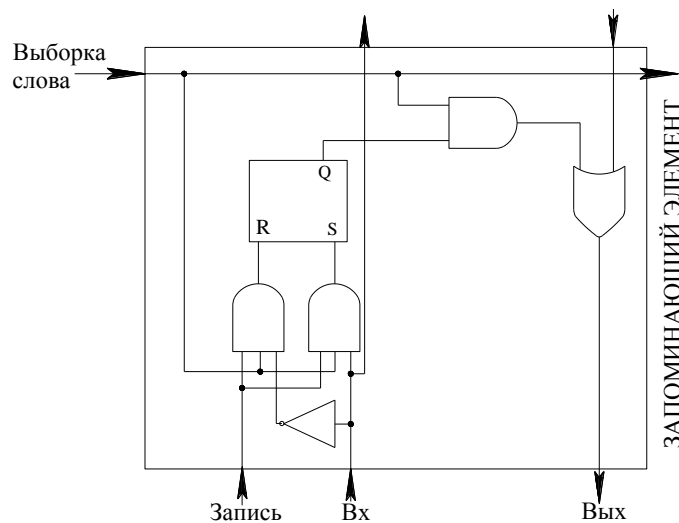


Рисунок 11.3 - Внутренняя организация модуля памяти с произвольным доступом

Таким образом, чтение слова из памяти осуществляется подачей адреса нужного слова на вход дешифратора и наблюдением состояния на линиях «ВЫХОД».

Линия «вход» каждого столбца используется для передачи информации в запоминающий элемент выбранной строки при выполнении операции записи. Это осуществляется с помощью двух вентилях. И на каждый

запоминающий элемент, который при наличии сигналов «выборка слова» и «запись» передает сигнал из линии «вход» на вход S триггера, а его дополнение – на вход R. Таким образом, запись слова в память производится заданием адреса нужной ячейки на входе дешифратора, а записываемого слова – на линии «вход» и затем – логической 1 на управляющей линии «запись» [1,5].

Нужно обратить внимание, что в приведенном выше описании фигурировал сигналы на линиях «запись», «вход» и «выход», являющиеся внутренними для модуля памяти. Эти сигналы связаны с внешними по отношению к модулю сигналами с помощью соответствующих вентилях и буферных схем. Такая связь может быть реализована различными способами, в зависимости от требуемых внешних характеристик модуля.

Двумерная адресация.

Недостаток рассмотренной схемы ЗУ произвольного доступа связан с большим размером адресного дешифратора. Например, при $n = 10$ адресный дешифратор должен иметь $2^{10} = 1024$ выхода. Каждый выход должен быть соединен с запоминающим элементом в строке. Количество связей можно уменьшить, если часть функций дешифратора передать самому запоминающему элементу.

Чтобы показать, каким образом это может быть сделано, возьмем один столбец ЗУ с организацией, приведенной на рисунке 2, и построим из него прямоугольную матрицу. При $n = 10$ число запоминающих элементов равно 210. Можно построить матрицу размером $2^5 \times 2^5$ или в общем случае $2^m \times 2^{n-m}$, как показано на рисунке 11.4. Каждый элемент исходного столбца теперь лежит на пересечении новой строки и столбца прямоугольной матрицы. Возьмем два дешифратора с $n/2$ входными и $2^{n/2}$ выходными линиями в каждом (будем считать, что n четно и матрица квадратная). При $n=10$ число $2^{n/2}$ равно 32. Выходные линии одного дешифратора будут служить для выборки строки, а другого — для выборки столбца матрицы. Общее число выходных линий в дешифраторах при таком подходе равно $2^{n/2} + 2^{n/2} = 2^{n/2+1}$, что существенно меньше $2n$ линий при одном адресном дешифраторе. В каждом запоминающем элементе должна выполняться операция И над линиями выборки строки и столбца. Эта операция даст логическую 1 только для одного элемента, находящегося на пересечении выбранных строки и столбца, логический 0 для всех остальных элементов, поскольку для них либо сигнал выборки столбца, либо строки будет равен 0. Выходной сигнал вентиля И выполняется функции сигнала выборки бита, который разрешается чтение или запись в данном элементе. Аналогичные двумерные матрицы нужны для всех остальных битов в слове. Матрицы можно считать лежащими в параллельных плоскостях. Оба адресных дешифратора работают сразу на все плоскости, таким образом, все биты

ячейки адресуются одновременно [1,5]. Полезно обратить внимание на то, что число выходных линий в дешифраторах при этом не увеличивается.

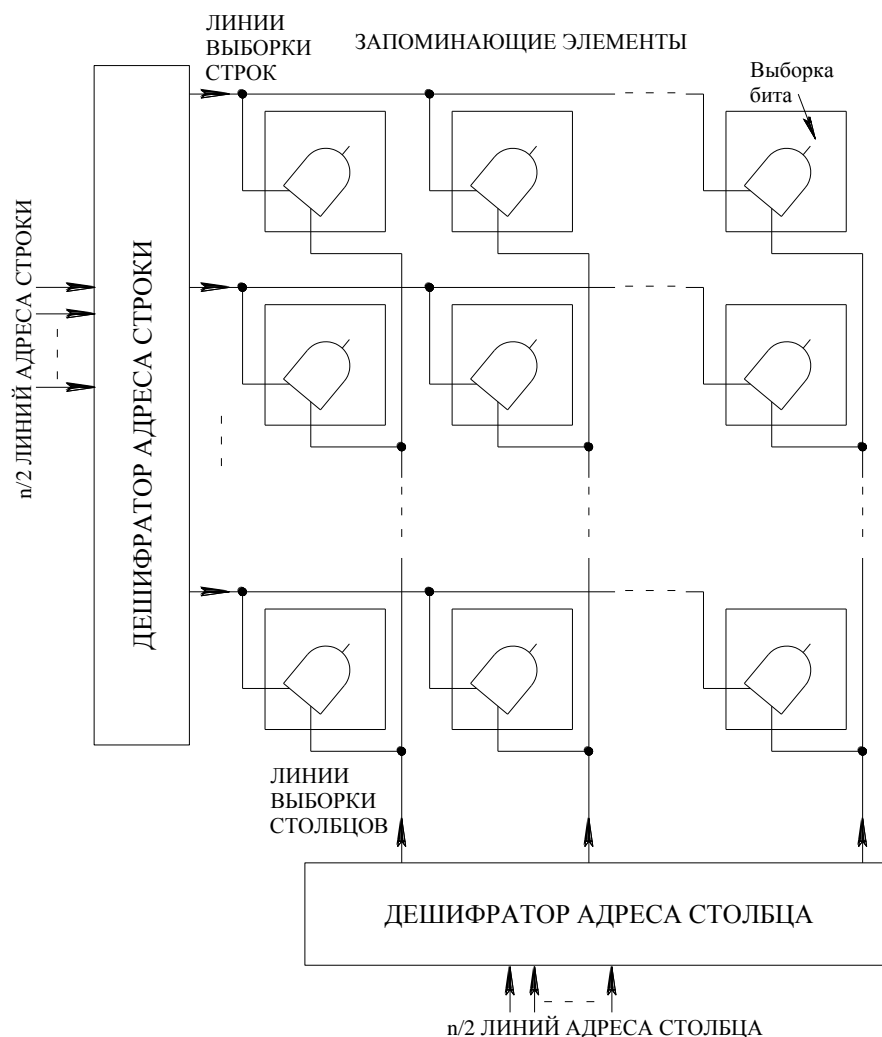


Рисунок 11.4 - Двумерная адресация ЗУ с произвольным доступом

12 лекция. Постоянные запоминающие устройства. Стеки

Содержание лекции: типовые структуры запоминающих устройств.

Цель лекции: изучить структуру постоянного запоминающего устройства.

В компьютерах и других цифровых системах память часто служит источником информации, остающейся неизменной. Примерами могут служить списки констант, таблицы для преобразования данных и постоянные программы. В таких случаях используются модули памяти, в которых изменить записанную информацию невозможно средствами самой использующей модуль системы. Такие модули выпускаются и называются

постоянными ЗУ (ПЗУ) или памятью типа ROM (Read - Only Memory - память только со считыванием).

Постоянные запоминающие устройства (ПЗУ) делятся на четыре типа:

- масочные, программируемые на заводе-изготовителе с применением специальных масок;
- однократно программируемые потребителем путем пережигания нихромовых или поликремневых перемычек;
- многократно программируемые потребителем со стиранием записанной информации ультрафиолетовым излучением;
- многократно программируемые потребителем с электрическим стиранием информации.

Применение ПЗУ для хранения неизменяющейся информации в некоторых приложениях дает значительные преимущества, поскольку информация не разрушается даже при выключении питания. Кроме того, в ПЗУ можно достичь много большей плотности упаковки информации, поскольку запоминающие элементы существенно проще.

Базовую структуру ПЗУ можно представлять себе состоящей из дешифратора адреса и совокупности подключенных к нему вентилях ИЛИ, как представлено на рисунке 12.1.

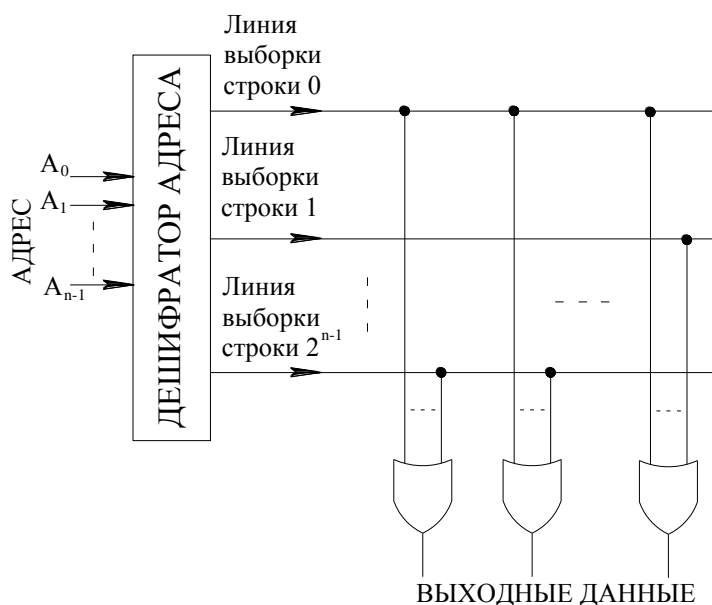


Рисунок 12.1 - Структура ПЗУ с произвольным доступом

Как и для ОЗУ, выходные линии дешифраторы находятся во взаимно-однозначном соответствии с ячейками ПЗУ. Число вентилях ИЛИ определяется разрядностью слова.

Память с последовательным доступом.

Термин «последовательный доступ» относится к запоминающим устройствам, в которых позиции слов становятся доступными для чтения или записи только в определенном порядке (рисунок 12.2). В ЗУ с последовательным доступом каждое хранящееся слово не привязывается к конкретным запоминающим элементам, а скорее к своему положению относительно других хранящихся слов. Например, слова могут перемещаться по запоминающим элементам, но сохранять при этом свою взаимную упорядоченность. В таком случае достаточно обеспечить средства для чтения только некоторых запоминающих элементов. То или иное конкретное слово считывается в тот момент, когда в процессе перемещения по памяти оно окажется на запоминающих элементах, из которых может производиться чтение. Аналогично информация записывается в позицию слова, когда в процессе перемещения эта позиция окажется на элементах, снабженных средствами записи. В некоторых видах ЗУ с последовательным доступом перемещаются сами запоминающие элементы, а позиции механизмов чтения и записи остаются постоянными. Рассмотрим ЗУ, в которых слова перемещаются относительно запоминающих элементов. По тому, как перемещаются слова, эти ЗУ делятся на два основных типа. В ЗУ первого типа слова перемещаются всегда в одном направлении вдоль цепочки из групп элементов, причем каждая группа вмещает одно слово. Информация записывается на группу элементов в начале цепочки и считывается в ее конце. Таким образом, данные становятся доступными для считывания в том порядке, в каком проводилась запись.

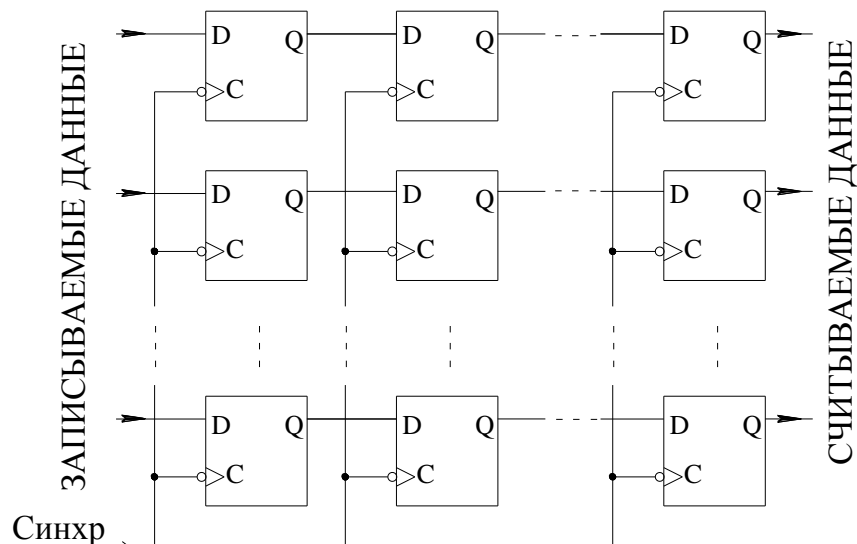


Рисунок 12.2 - Модель последовательного ЗУ в виде сдвиговых регистров

В ЗУ второго типа слова могут перемещаться по цепочке в любом направлении, и информация записывается и считывается на одной и той же группе элементов. Нетрудно видеть, что в ЗУ второго типа слова считываются в порядке, противоположном порядку записи. ЗУ этого типа называются стеками, и их мы рассмотрим в следующем разделе. Простейшим примером последовательного ЗУ первого типа может служить сдвиговый регистр. На самом деле в качестве модели ЗУ первого типа со словом из t битов лучше рассматривать группу из m сдвиговых регистров, работающих согласованно, так сказать, «в унисон». Как показано на рисунке 12.3, сдвиговые регистры состоят из синхронных В-триггеров, синхронизирующие входы которых соединены вместе. Информация записывается на левом конце и считывается на правом. По каждому синхроимпульсу информация продвигается вправо на одну позицию, а слева записывается новая информация. То, что записано в крайнем правом столбце, при этом теряется.

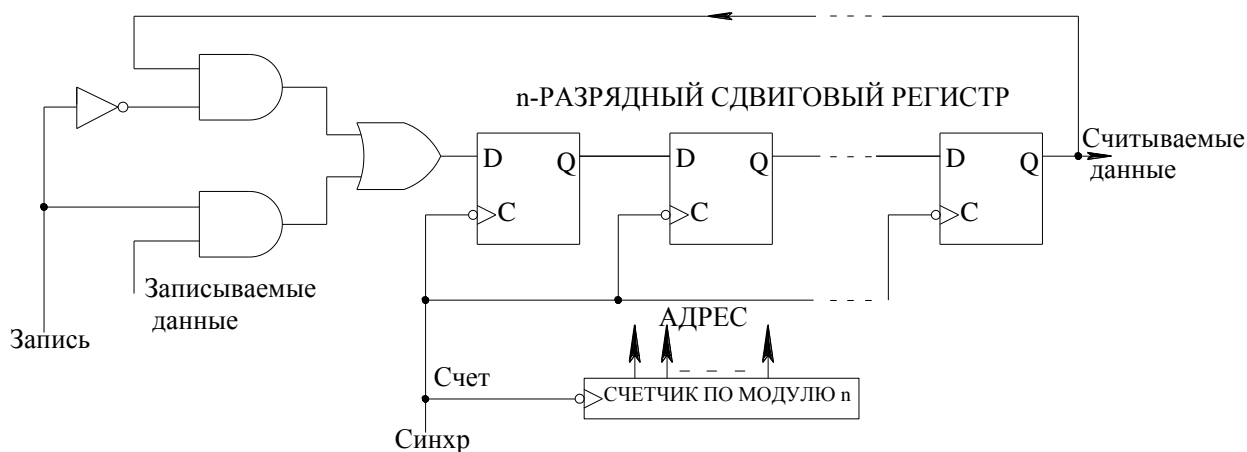


Рисунок 12.3 - Последовательная память с циркуляцией

Подобное устройство станет более удобным во многих приложениях, если предусмотреть возврат информации, выталкиваемой на правом конце, и запись ее на левом, в тех случаях, когда не требуется замещать старые данные новыми. Это можно реализовать с помощью схемы, показанной на рисунке 6 и состоящей из одного регистра. На схеме селектор из двух вентилях И и одного ИЛИ осуществляет выбор между внешними данными и данными, поступающими с правого триггера, и результат выбора подает на вход левого триггера. Управляет селектором сигнал «запись». Когда этот сигнал равен логической 1, записываются внешние данные; в противном случае осуществляется циркуляция данных. Вообще говоря, в последовательном ЗУ нужно каким-то образом знать о положении каждого слова. Как показано на рисунке 6, эту проблему можно решить с помощью счетчика, на вход которого подаются синхроимпульсы, осуществляющие сдвиг. Счетчик должен считать по модулю n , где n — число триггеров в каждом сдвиговом регистре. Таким образом, на счетчике будет одно и то же значение всякий раз, когда

некоторое слово будет попадать в некоторую фиксированную позицию. Следовательно, текущее состояние счетчика можно связать, например, со словом в позиции считывания. При таком подходе нужное слово для чтения можно выбрать, просто подождать, пока номер его позиции появится на счетчике. Можно также в эту позицию записать новые данные извне, установив логическую 1 на линии «запись» до прихода следующего синхроимпульса. Таким образом, содержимое счетчика может служить адресом как при чтении, так и при записи. Доступ к конкретному слову в последовательной памяти, очевидно, требует в среднем существенно больше времени, чем в памяти с произвольным доступом. И тем не менее ЗУ с произвольным доступом играют важную роль в микрокомпьютерах, поскольку благодаря более простой структуре они обладают большей плотностью упаковки информации и дешевле в изготовлении.

Стеки.

Стеком называется запоминающее устройство с последовательным доступом, в котором слова считываются в порядке, обратном записи, т.е. по правилу LIFO (Last-in First-out – последним вошел – первым вышел). Стек можно представить себе в виде вертикально расположенного массива запоминающих ячеек, как показано на рисунке 12.4.

Доступ осуществляется всегда к вершине ячейке – вершине стека. Когда слово записывается в вершину стека, слово, занимавшее ее, и все нижележащие слова сдвигаются вниз на одну ячейку, а содержимое нижней ячейки теряется. При этом говорят, что стек опускается. Существует обратная операция, при которой стек поднимается. Когда стек поднимается, то слово на вершине стека удаляется и его место занимает и тем самым становится доступным слово из второй сверху ячейки.

С точки зрения внутренней структуры стек можно представить группой работающих «в унисон» регистров со сдвигом в обоих направлениях, по одному регистру на каждый столбец битов. Один такой регистр показан на рисунке 12.5. Он состоит из группы триггеров, управляемых сигналом «направление сдвига» и синхросигналом. По каждому синхроимпульсу содержимое каждого триггера передается на соседний триггер в направлении, заданном сигналом «направление сдвига». Опускание стека соответствует одному направлению, подъем – другому.

Во многих приложениях применение стека оказывается очень удобным и естественным. Для таких приложений характерно использование памяти с выборкой не по произвольным адресам слов, а на основе взаимной их упорядочности. Например, нужные в процессе вычислений операнды можно было бы загрузить в стек в порядке, обратном их использованию [1,5]. Подъем слов из стека будет тогда поставлять операнды в нужном порядке.

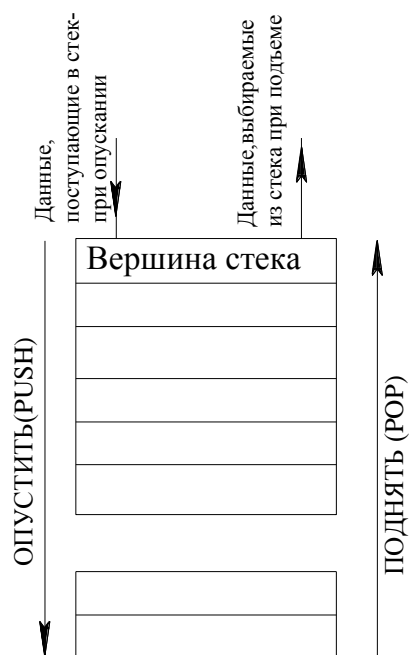


Рисунок 12.4 - Схема работы стека

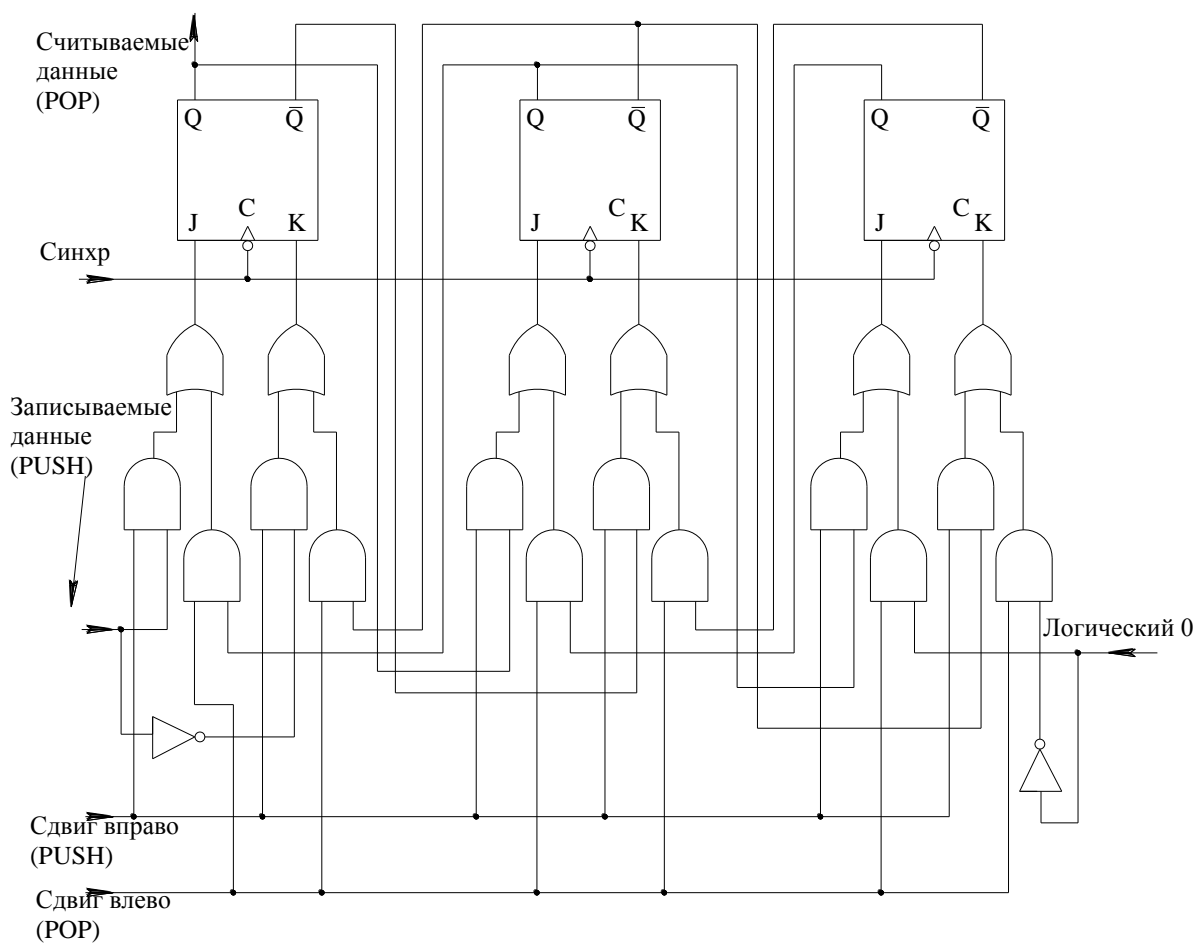


Рисунок 12.5 - Двухнаправленный сдвиговый регистр, соответствующий однобитовому столбцу в стеке

Стек играет важную роль в микрокомпьютерах как средство сохранения адресов возврата и состояния данных для подпрограмм.

Его использование приводит к существенным упрощениям, когда одна подпрограмма вызывает другую, которая в свою очередь может вызвать третью, и т.д. В таких случаях при каждом вызове адрес возврата из текущей подпрограммы и другая необходимая информация в нужном порядке выбирается из стека.

13 лекция. Понятие интерфейса

Содержание лекции: интерфейс, порты ввода-вывода.

Цель лекции: изучить понятие интерфейса, схему порта ввода и порта вывода в иллюстративном микропроцессоре.

Порты ввода/вывода.

В соответствии с общепринятым соглашением направление потоков вводной и выводной информации рассматривается относительно микропроцессора. Поэтому портом ввода называется любой источник данных, например, адресуемый регистр, подключенный к шине данных микрокомпьютера. Он выдает слово в микропроцессор, когда к нему происходит обращение. Портом вывода называется приемник данных, например, адресуемый регистр, подключенный к шине данных микропроцессора. Он получает слово от микропроцессора, когда последний к нему обращается. В большинстве микропроцессоров для адресации портов (т.е. для выборки нужного порта) используется адресная шина или ее часть. Довольно часто адреса портов ввода отличаются от портов вывода и от адресов памяти не значениями, а сигналами на соответствующих управляющих линиях.

Порты ввода/вывода иллюстративного микропроцессора.

На рисунке 13.1 показаны типичные схемы порта ввода и порта вывода для нашего микропроцессора. Этим двум портам присвоен уникальный 8-битовый код (адрес устройства), по которому микропроцессор может отличать их от других портов. В нашем случае обоим портам присвоен один код (01100101), что вполне допустимо, поскольку между собой они будут отличаться сигналами в управляющих линиях «ввод» и «вывод». При выполнении команд ввода или вывода на младшую половину адресной шины подается код устройства, взятый из второго байта команды. Специальный вентиль И на восемь входов служит для распознавания кода данного устройства. Входы вентиля подключены к 8 младшим линиям адресной шины, причем в зависимости от кода часть сигналов инвертируется. Выход вентиля И («выборка») используется для выборки обоих портов.

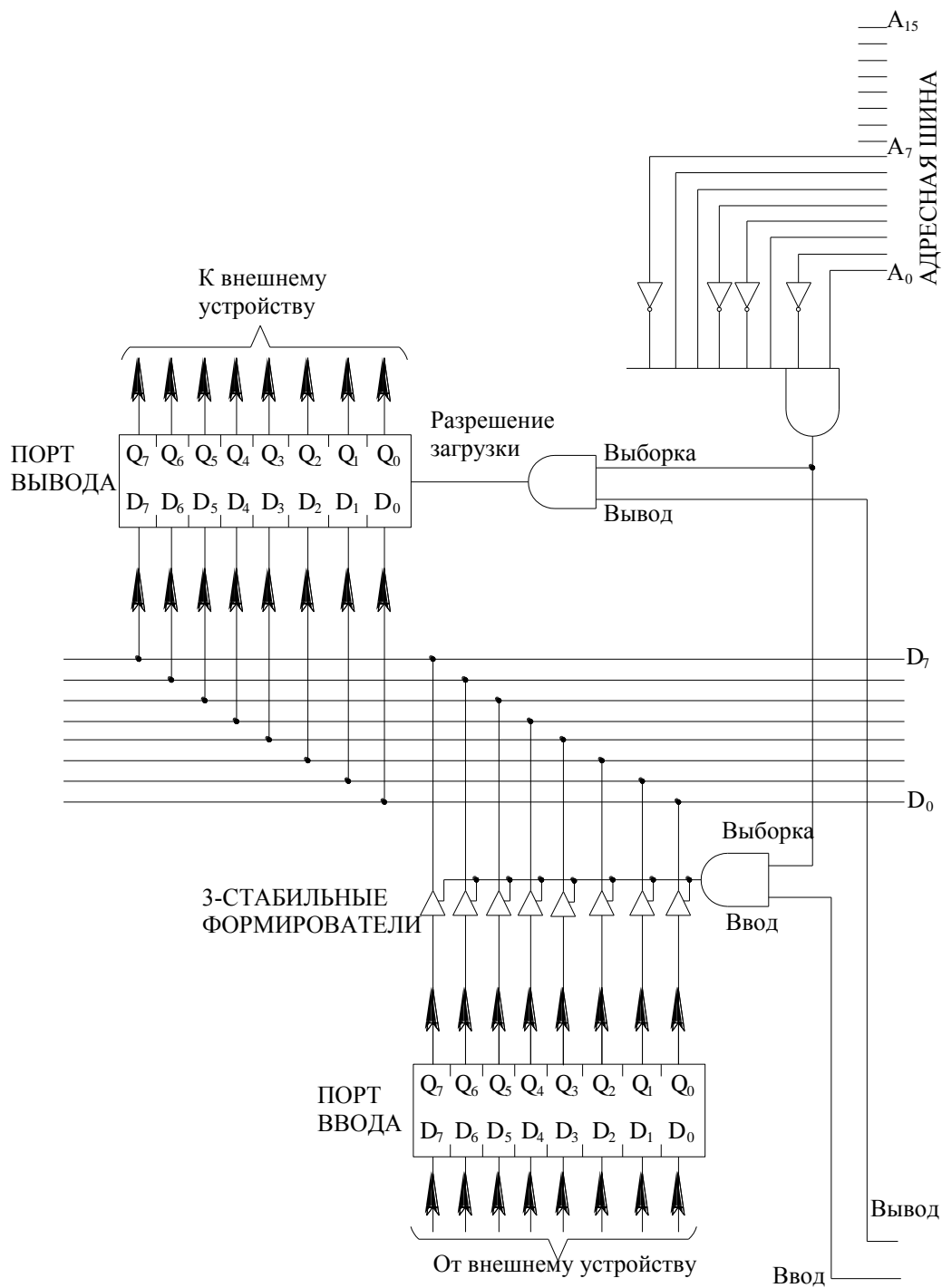


Рисунок 13.1 - Схема порта ввода и порта вывода в иллюстративном микропроцессоре

Порт ввода имеет 8-битовый регистр, на который поступает информация от некоторого внешнего устройства. Выходы этого регистра подключены к шине данных через восемь тристабильных формирователей. Сигнал разрешения на эти формирователи есть результат логического И сигнала «выборка» и управляющего сигнала «ввод» от микропроцессора.

Содержимое регистра подается на шину данных, когда (1) «выборка» равна логической 1, т. е. задан код устройства, относящийся к данному порту, и (2) «ввод» равен логической 1, т.е. выполняется команда ввода, и в ее фазе выполнения наступил момент ввода данных. Задача порта – только поместить информацию на шину данных. Дальнейший ее путь к месту назначения (к аккумулятору в данном случае) определяется микропроцессором. Передача данных осуществляется в третьем машинном цикле команды ввода. Временные диаграммы сигналов в этом цикле показаны на рисунке 13.2. В машинный цикл входит два три синхроимпульса.

В начале первого синхроимпульса T_1 микропроцессор подает код соответствующего порта на младшую половину адресной шины. По концу T_1 подается логическая 1 на линию «ввод», которая остается в этом состоянии до переднего фронта импульса T_3 . При этом предполагается, что на шину данных в ответ на сигнал «ввод» будут поданы данные с регистра порта ввода, и они будут доступны в интервале между передними фронтами T_2 и T_3 . Стробирование передачи данных на аккумулятор осуществляется по концу T_2 .

Нужно отметить, что описанная логика работы порта ввода налагает два ограничения на время срабатывания схем. Одно ограничение связано с тем, что информация должна быть подана на шину данных не позднее, чем через полтакта после прихода сигнала «ввод». Это означает, что задержка в тристабильных формирователях плюс задержка в двухвходовом венти́ле И, управляющем усилителями, не должна превышать одного такта.

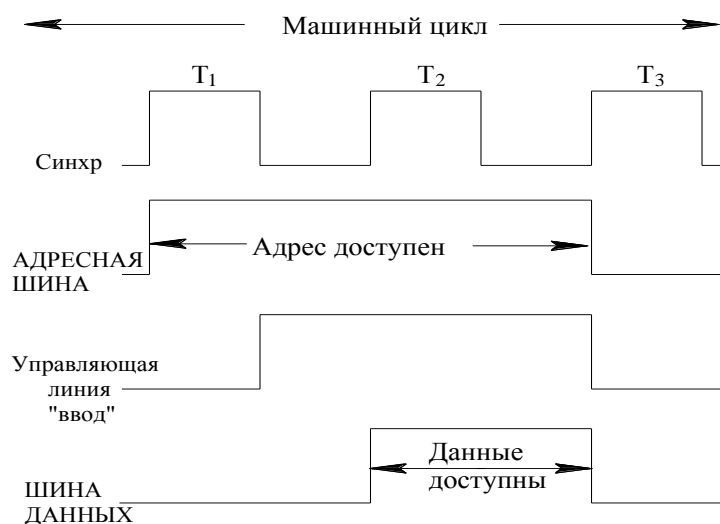


Рисунок 13.2 - Временные диаграммы ввода данных

В конкретных микропроцессорах мы можем встретить схемы ввода данных, отличающиеся несколько от описанной, однако аналогичные временные ограничения существуют и для них. Как правило, к этим требованиям во внешних компонентах удовлетворить легко, поскольку

элементы, из которых строятся порты ввода/вывода, обычно оказываются более быстродействующими по сравнению с элементами, из которых строится микропроцессор. Например, в портах ввода/вывода можно использовать порты TTL-схемы средней степени интеграции, более быстрые по сравнению с большими интегральными МОП - схемами, применяемыми в микропроцессоре.

Порт вывода, показанный на рисунке 13.1, имеет 8-битавый регистр, используемый для передачи информации внешнему устройству. Регистр построен из синхронных фиксаторов D-типа, управляемых общей линией разрешения загрузки. Входы D-триггеров подсоединены непосредственно к шине данных. Сигнал «разрешение загрузки» формируется на двухвходовом вентиле И из сигналов «выборка» и «вывод». Таким образом, информация на шине данных, поступившая из аккумулятора микропроцессора, передается в регистр порта вывода, когда (1) «выборка» равна логической 1, т.е. выполняется команда вывода и в ее фазе выполнения наступил момент вывода данных.

Передача данных осуществляется в третьем машинном цикле команды вывода. Временные диаграммы сигналов в этом цикле показаны на рисунке 13.3.

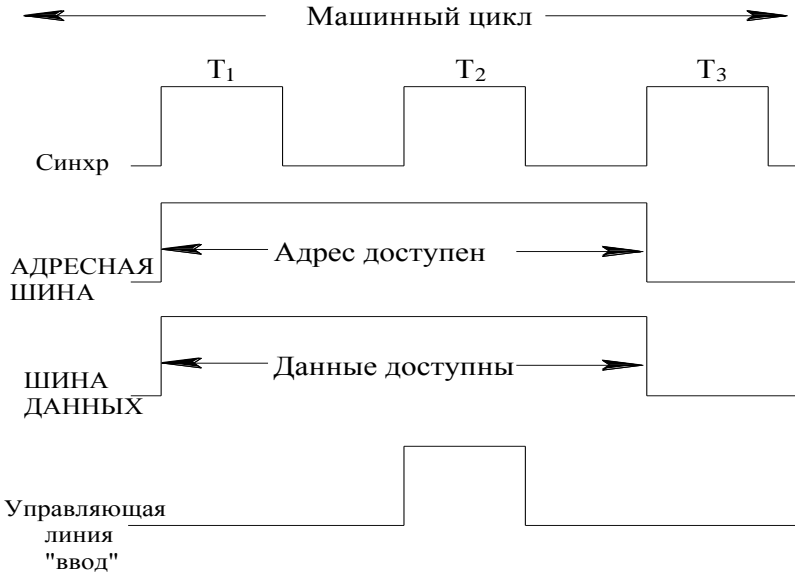


Рисунок 13.3 - Временные диаграммы вывода данных

По переднему фронту синхроимпульса T₁ микропроцессор подает код устройства на младшую половину адресной шины и данные на шину данных. Затем в момент T₂ выдается сигнал «вывод», стробирующий передачу данных на регистр вывода.

Схемы порта вывода так же, как и порта ввода, должны удовлетворять определенным временным ограничениям, и так же, как для портов ввода, эти

ограничения обычно нетрудно соблюсти благодаря относительно высокому быстродействию схем ввода/вывода.

Варианты схем ввода/вывода.

Различия между только что рассмотренной логикой работы ввода/вывода в иллюстративном микропроцессоре и логикой работы в некоторых промышленных микропроцессорах в основном касаются временных соотношений и смысла управляющих сигналов. В нашем микропроцессоре сигналы «ввод» и «вывод» выполняли и управляющие, и стробирующие функции. Во многих промышленных микропроцессорах сигналы, аналогичные сигналам «ввод» и «вывод», получают как комбинация других управляющих сигналов.

В некоторых случаях предусматривается управляющая линия, которая задает направление передачи по шине данных. Это линия, назовем ее, скажем, «вывод данных», при значении 1 задает вывод, а при значении 0 – ввод. Для задания же момента передачи служит либо синхросигнал, либо другая управляющая линия. Интерфейс портов ввода/вывода с таким микропроцессором можно реализовать аналогично интерфейсу с иллюстративным микропроцессором, сформировав сигналы «ввод» и «вывод» следующим образом:

$$\begin{aligned} \text{Ввод} &= (\text{Вывод данных}) \text{ Строб}; \\ \text{Вывод} &= (\text{Вывод данных}) \text{ Строб}, \end{aligned}$$

где Строб – это сигнал, определяющий момент передачи.

В некоторых схемах вначале машинного цикла по шине данных посылается управляющее слово, которое определяет тип предстоящей передачи: ввод или вывод, чтение из памяти или запись в память. Это управляющее слово запоминается во внешнем регистре и в комбинации с синхросигналами или с другими управляющими сигналами используется для образования сигналов, аналогичных сигналам «ввод», «вывод»; «чтение» и «запись» нашего иллюстративного микропроцессора. При таком подходе некоторые управляющие линии становятся ненужными [1,2,4].

14 лекция. Архитектура микропроцессоров

Содержание лекции: архитектура микропроцессоров.

Цель лекции: изучить архитектуру иллюстративного микропроцессора.

В настоящее время выпускается много моделей микропроцессоров с очень широким спектром параметров. Мы не будем пытаться изучить их все, а опишем гипотетический микропроцессор с архитектурой (т. е. структурой и

языком), обладающей типичными чертами. Таким образом, мы сможем лучше осветить наиболее существенные аспекты функционирования микропроцессоров вообще, не вдаваясь в детали, характерные для тех или иных конкретных моделей. Хотя наш гипотетический микропроцессор и не выпускается промышленностью, он близок к некоторым реально существующим моделям, например, к процессорам серии 8000 фирмы «Интел».

Микрокомпьютер — это система с шинной организацией, состоящая из модулей, или блоков, реализованных в виде больших интегральных схем. Эти модули обрабатывают информацию, управляют потоком и интерпретацией команд, управляют работой шин, хранят информацию и осуществляют взаимодействие между компьютером и его окружением (рисунок 14.1).

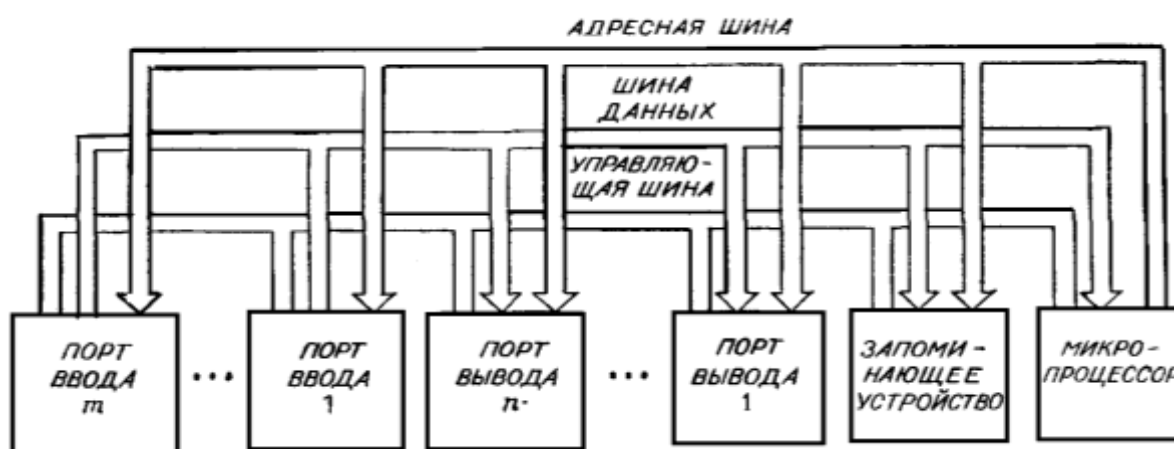


Рисунок 14.1 - Структура микрокомпьютера с шинной организацией

Первые четыре функции обычно выполняются одним функциональным блоком — микропроцессором. Функции хранения информации осуществляет запоминающее устройство. В него могут входить как постоянная память, так и оперативная. Наконец, внешние коммуникации осуществляют блоки, называемые портами ввода/вывода. Каждый такой порт является интерфейсом между микропроцессором и каким-либо внешним устройством, например, терминалом, внешней памятью для хранения больших объемов информации, контроллером технологических процессов или измерительным прибором.

Структура микрокомпьютера, состоящего из подобных блоков, приведена на рисунке. Взаимодействие блоков осуществляется при помощи шин трех типов: адресных шин, шин данных и управляющих шин. В дальнейшем мы будем рассматривать именно такую структуру шин, хотя возможны и другие варианты.

Структура микропроцессора. Общая структура нашего гипотетического иллюстративного микропроцессора представлена на рисунке 7.2. Входящие в его состав компоненты говорят о том, что он является

процессором с программным управлением. Некоторые компоненты, а именно: программный счетчик (счетчик команд), стек и регистр команд - служат для обработки команд. Такие компоненты, как АЛУ, триггер переноса, общие регистры (или рабочие) и регистр адреса данных, служат для обработки данных. Все остальные компоненты, а именно: дешифратор команд и блок управления и синхронизации (БУС) - управляют работой других компонентов. Взаимодействие компонентов осуществляется по внутренним каналам передачи данных. Связь микропроцессора с другими блоками (ЗУ и устройствами ввода/вывода) происходит по адресной шине, шине данных и управляющей шине.

Микропроцессор работает со словами, состоящими из 8 битов. Такие слова, называемые байтами, удобны при выполнении арифметических и логических операций и используются в большинстве выпускаемых микропроцессоров. Поэтому мы будем предполагать, что все данные представляются именно такими словами. Если в расчетах встречаются числа «большой длины», то применяются специальные программы для вычислений с «двойной точностью» «тройной точностью» и. т. д.

С другой стороны, адрес из 8 битов позволяет прямо адресовать только $2^8=256$ ячеек памяти. Для реальных задач этого, конечно, мало. Поэтому для задания адреса памяти обычно используется 16 разрядов (два байта), и это позволяет прямо адресовать $2^{16}=65\,536$ ячеек.

Информация к микропроцессору и от него передается по шинам. Шины данных в соответствии с длиной слова состоят из 8 линий, тогда как адресная шина состоит из 16 линий. Адресная шина однонаправленная, а шина данных двунаправленная. Управляющая шина состоит из 5 линий, ведущих к блоку управления и синхронизации, и 8 выходящих из него линий. По этим линиям передаются управляющие и тактирующие сигналы между компонентами микропроцессора и между микропроцессором и другими блоками микрокомпьютера (рисунок 14.2).

Счетчик команд состоит из 16 бит и содержит адрес очередного байта команды, считываемого из памяти. Он автоматически увеличивается на единицу после чтения каждого байта. Существует связь между счетчиком команд и вершиной стека из 64 регистров. Одна из функций стека — сохранение адреса возврата из подпрограммы. В стеке могут также сохраняться данные из верхних трех общих регистров и триггера переноса.

В то время как слово данных всегда состоит из одного байта, команда может состоять из одного, двух или трех байтов. Первый байт любой команды поступает из памяти по шине данных на регистр команд. Этот первый байт подается на вход дешифратора команд, который определяет ее смысл. В частности, дешифратор определяет, является ли команда однобайтовой или она состоит из большего числа байтов. В последнем случае дополнительные байты передаются по шинам данных из памяти и принимаются или на регистр адреса данных, или на один из общих регистров.

Регистр адреса данных содержит адрес операнда для команд, обращающихся к памяти, адрес порта для команд ввода/вывода или адрес следующей команды для команд перехода.

Пятнадцать 8-битовых общих регистров содержат операнды для всех команд, работающих с данными. Для указания этих регистров используются 4-битовые коды от 0000 до 1110. Регистр 0000 называется аккумулятором (АК) и участвует во всех арифметических и логических операциях. В частности, он содержит один из операндов перед выполнением операции и получает результат после ее завершения. Обычно обращения к общим регистрам осуществляются при помощи R-селектора или r-селектора. R-селектор позволяет обращаться к любому регистру, тогда как через r-селектор доступны только регистры 0000, 0001 и 0010.

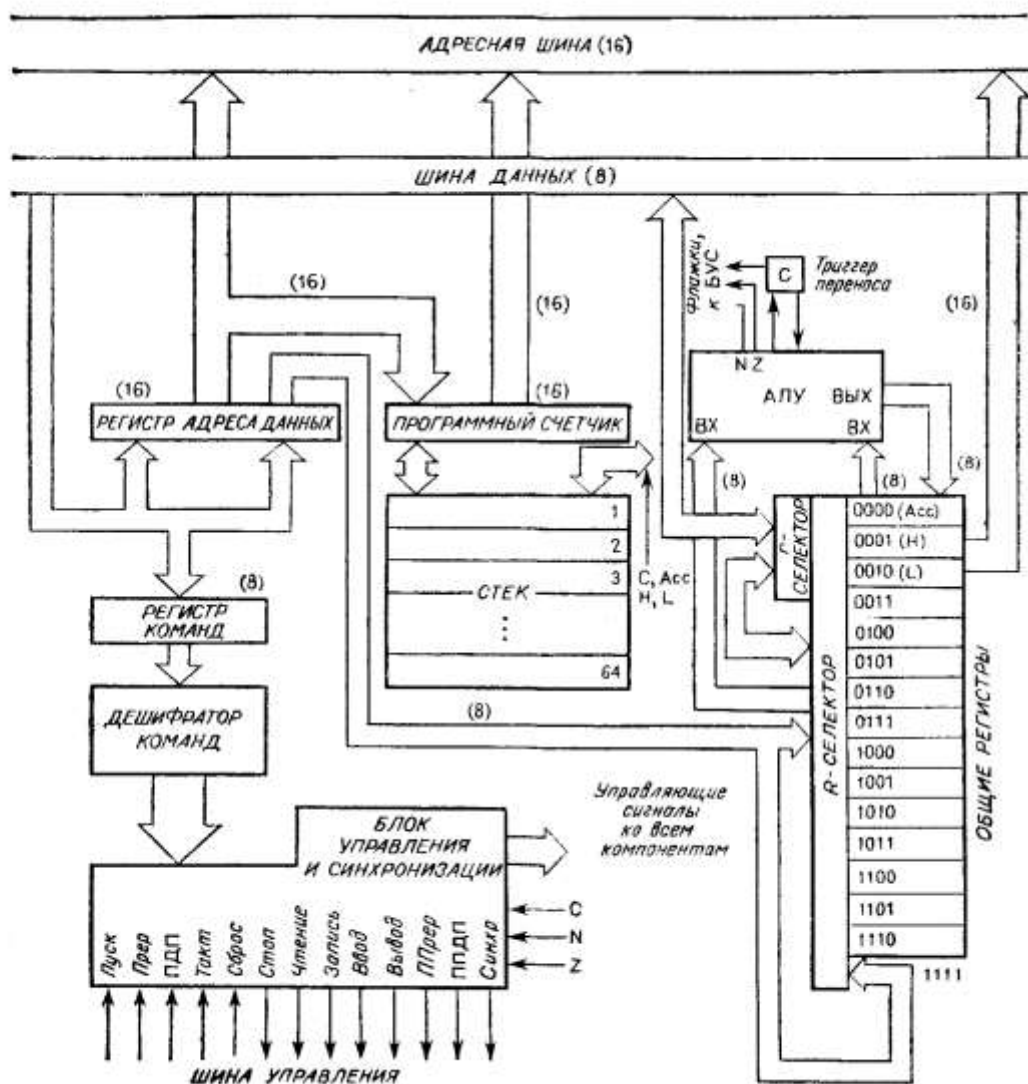


Рисунок 14.2 - Гипотетический микропроцессор (числа в скобках указывают количество линий)

Очень полезная возможность, присутствующая во многих машинах— это косвенная адресация. Некоторый вариант косвенной адресации встроены и в наш микропроцессор. Задание несуществующего регистра общего назначения 1111 используется как указание на то, что нужно обратиться к байту памяти по 16-разрядному адресу, который получается комбинированием содержимого двух фиксированных общих регистров, а именно: старшие 8 разрядов адреса — из регистра 0001, а младшие 8 разрядов адреса — из регистра 0010 (в дальнейшем эти два регистра мы будем называть H и L соответственно; H — от слова Higher, высший, или старший, L — от Lower, низший, или младший). Чтобы пояснить подобную косвенную адресацию, предположим, что регистр 0001, т. е. регистр H, содержит 1011101, а регистр 0010, т. е. регистр L, содержит 00101011. Тогда любая команда, ссылающаяся на регистр 1111, вызовет выборку операнда из ячейки памяти с адресом 101110100101011.

Все арифметические и логические операции выполняются в арифметико-логическом устройстве (АЛУ). Входами АЛУ служат две 8-битовые шины. Одна из них идет от аккумулятора (регистр 0000), а другая — от R-селектора, который выбирает либо один из регистров общего назначения от 0000 до 1110, либо ячейку памяти, если задана косвенная адресация. Еще одна входная линия поступает в АЛУ от триггера переноса C, который участвует в некоторых арифметических и логических операциях.

Результаты из АЛУ передаются в аккумулятор по выходной 8-битовой шине. Существуют еще две линии, идущие от АЛУ к блоку управления и синхронизации; они передают информацию о наличии или отсутствии двух особых условий: аккумулятор содержит нули (линия Z) и старший разряд аккумулятора равен 1 (линия N). Вторая линия очень удобна при работе с числами в дополнительном коде, когда старший разряд знаковый, причем 1 соответствует отрицательным числам. Триггер переноса и обе линии состояния АЛУ Z и N называются флажками и используются в командах условного перехода.

Последний компонент микропроцессора — это блок управления и синхронизации (БУС). Он получает сигналы от дешифратора команд, который анализирует команду. Как уже упоминалось, в БУС из АЛУ и от триггера переноса поступают сигналы, по которым определяются условия для передач управления. Все остальные компоненты микропроцессора получают от БУС управляющие и синхронизирующие сигналы, необходимые для выполнения команды. С помощью 13 внешних линий реализуется интерфейс устройства управления с другими модулями.

Блок управления и синхронизации. Вообще говоря, микропроцессор состоит из триггеров и комбинационных вентилей и работает как синхронная последовательностная схема. Триггеры всех регистров и флажков микропроцессора получают синхроимпульсы по общей линии от внешнего генератора. Поэтому любые изменения содержимого регистров или флажков могут происходить лишь в строго определенные моменты времени.

Новые состояния определяются комбинационными схемами по текущим состояниям триггеров и флажков, а также по значениям внешних сигналов. Общая схема работы показана на рисунке 14.3. Комбинационные схемы выполняют функции, связанные с передачей данных между регистрами, с обработкой данных, формированием информации о состоянии, выбором пути при передачах управления и. т. п. Поскольку некоторые из этих функций довольно сложны, реализующие их комбинационные схемы оказываются многоуровневыми и, следовательно, дают заметные задержки при распространении сигнала. Частота синхроимпульсов выбирается такой, чтобы в интервале между импульсами успевали переключиться триггеры и сработать комбинационные схемы в расчете на самую большую задержку.

На рисунке 14.3 показаны также и внешние по отношению к микропроцессору компоненты. В их число входят запоминающие устройства и устройства ввода/вывода. Информация между внешними компонентами и микропроцессором передается по шинам также в синхронном режиме.

В нашем микропроцессоре частота от тактового генератора подается на входную управляющую линию такта. На обмен данными с внешним устройством уходит два такта, т. е. два периода импульсов синхронизации. В них входит и время срабатывания шин, и время отклика внешней компоненты. Поскольку шины участвует во всех внешних обменах, именно они фактически оказываются «узким местом», ограничивающим быстродействие микропроцессора. Поэтому вся синхронизация в микропроцессоре строится вокруг внешних обменов.

Последовательность синхронизирующих импульсов разбивается на так называемые машинные циклы: по три такта в каждом цикле. При работе микропроцессора в каждом машинном цикле происходит один внешний обмен. Первый и второй такты тратятся на обмен и иногда на обработку данных; третий такт весь выделяется для той переработки данных, которую нужно выполнить между последовательными внешними обменами.

При всех внешних обменах используются шины адресов и данных. Передачи данных делятся на два класса: входные и выходные (по отношению к микропроцессору). Каждый класс характеризуется своими временными диаграммами сигналов на шинах. Выходные обмены бывают двух типов. Обмены первого типа, в которых участвует главная память, называются записью; второго, в которых участвует устройство вывода, называются выводом. В классе входных обменов три типа. Два из них называются чтением и вводом, и в них участвуют соответственно главная память и устройство ввода.

Четыре выходных управляющих шины — «запись», «вывод», «чтение» и «ввод» — служат для задания типа и момента обмена. На рисунке 14.4 (а) приведены временные диаграммы сигналов за машинный цикл, в котором выполняется выходной обмен. Микропроцессор задает как состояние адресной шины, подавая на нее либо 16-разрядный адрес памяти для операции

записи, либо 8-разрядный код устройства и операции вывода, так и состояние шины данных.

Предполагается, что информация на шинах правильно установлена и неизменна - от переднего фронта импульса T_1 до переднего фронта импульса T_3^1 . При операции записи импульс на управляющую линию «запись» подается одновременно с T_2 , и он используется для стробирования передачи содержимого шины данных в ячейку главной памяти с адресом, заданным на адресной шине. При операции вывода импульс одновременно с T_2 подается на управляющую линию «вывода» и используется для стробирования передачи содержимого шины данных в регистр устройства вывода, код которого задан на младшей половине адресной шины.

На рисунке 14.4 (б) показаны временные диаграммы для входных обменов. Содержимое адресной шины задается в те же моменты и теми же способами, что и для выходных обменов. Содержимое шины данных на этот раз определяется внешним компонентом. Микропроцессор в интервале между задним фронтом T_3 и передним фронтом T_3 устанавливает логическую 1 на управляющую линию «чтение» или «ввод», и это служит запросом к внешнему компоненту поместить данные на шину данных.

Линия «чтение» используется главной памятью, а линия «ввод» - устройствами ввода. В обоих случаях предполагается, что информация на шине данных достоверна между передними фронтами импульсов T_2 и T_3 и стробирование передачи ее на соответствующий регистр микропроцессора выполняется одновременно с задним фронтом T_2 .

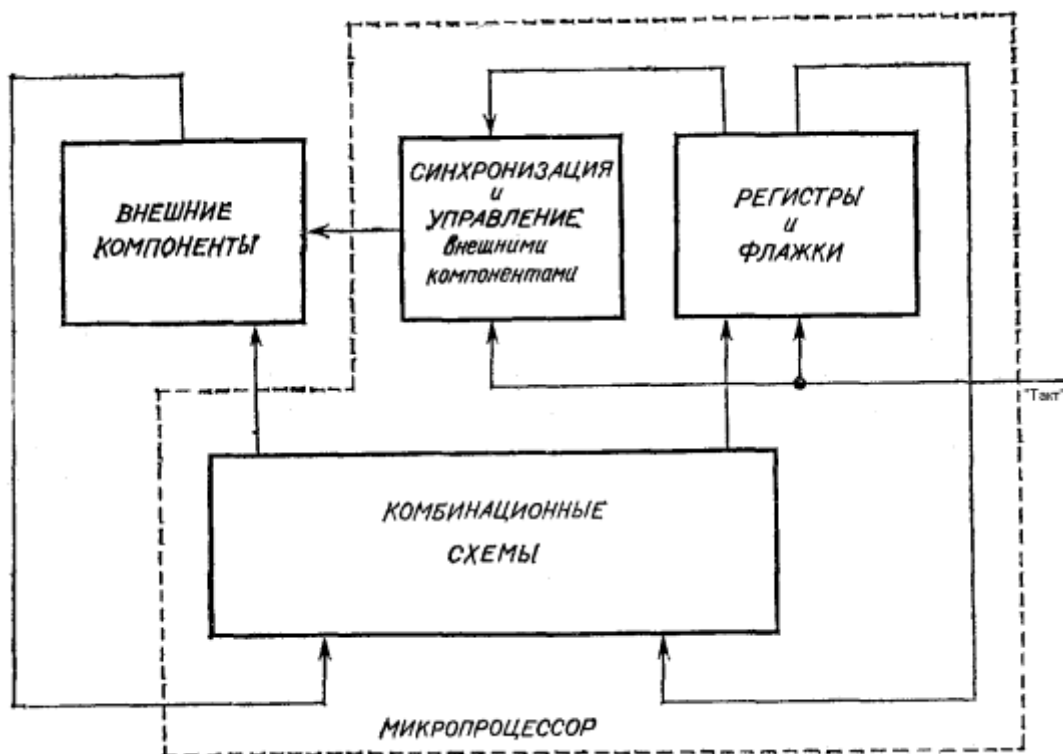


Рисунок 14.3 - Общая схема работы микропроцессора

Кроме рассмотренных выше управляющих линий, существует ещё восемь: Пуск, Сброс, Стоп, Синхр., Прер., ППрер., ПДП и ППДП. Четыре последние относятся к прерываниям программы и прямому доступу в память. Мы остановимся на первых четырех. «Пуск» — это входная линия, которая устанавливает режим автоматической работы микропроцессора, а именно: появление импульса в этой линии приводит к тому, что процессор по первому же синхроимпульсу начнет выполнять программу с той команды, адрес которой находился на счетчике команд.

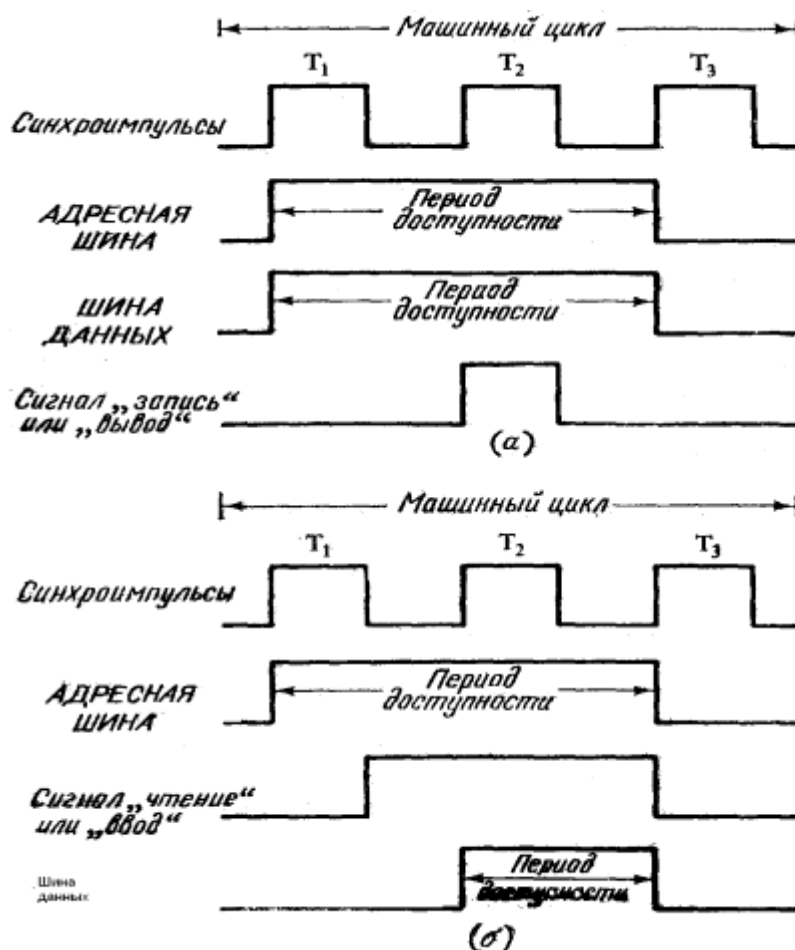


Рисунок 14.4 - Временные диаграммы обменов в иллюстративном микропроцессоре: диаграмма выходных обменов (а); диаграмма входных обменов (б)

«Сброс» — это входная линия, с помощью которой микропроцессор устанавливается в начальное состояние. В частности, по импульсу в линии «сброс» прекращается выполнение программы, счетчик команд устанавливается в нуль, а флажки прерываний сбрасываются. «Стоп» — это выходная линия, на которой поддерживается состояние логической 1 все время, пока микропроцессор не выполняет программы (остановлен). И наконец, «Синхр.» — это выходная линия, на которую подается импульс одновременно с третьим синхроимпульсом каждого машинного цикла и

которая используется в некоторых случаях для согласования работы некоторых внешних компонентов с внутренним машинным циклом микропроцессора.

Система команд. Обычно микропроцессор выполняет команды из некоторого фиксированного набора, называемого системой команд. Разные микропроцессоры имеют различные системы команд. Описание системы команд содержит символическое обозначение каждой команды, формат, а также символическое и словесное описание выполняемого действия. Никаких других команд наш микропроцессор выполнять не может, следовательно, решение любой задачи должно быть представлено в виде последовательности этих команд [5].

Формат каждой команды полностью определяет представление команды в памяти компьютера. Таким образом, команда в памяти хранится как последовательность из нулей и единиц. С другой стороны, для человека много удобнее символические обозначения. Поэтому при написании программ можно пользоваться символической формой, преобразуя ее в машинный двоичный формат непосредственно перед вводом в машину.

15 лекция. Цифро-аналоговые преобразователи. Аналогово-цифровые преобразователи

Содержание лекции: цифро-аналоговые и аналогово-цифровые преобразователи.

Цель лекции: изучить принцип действия цифро-аналоговых и аналогово-цифровых преобразователей.

Аналого-цифровой преобразователь.

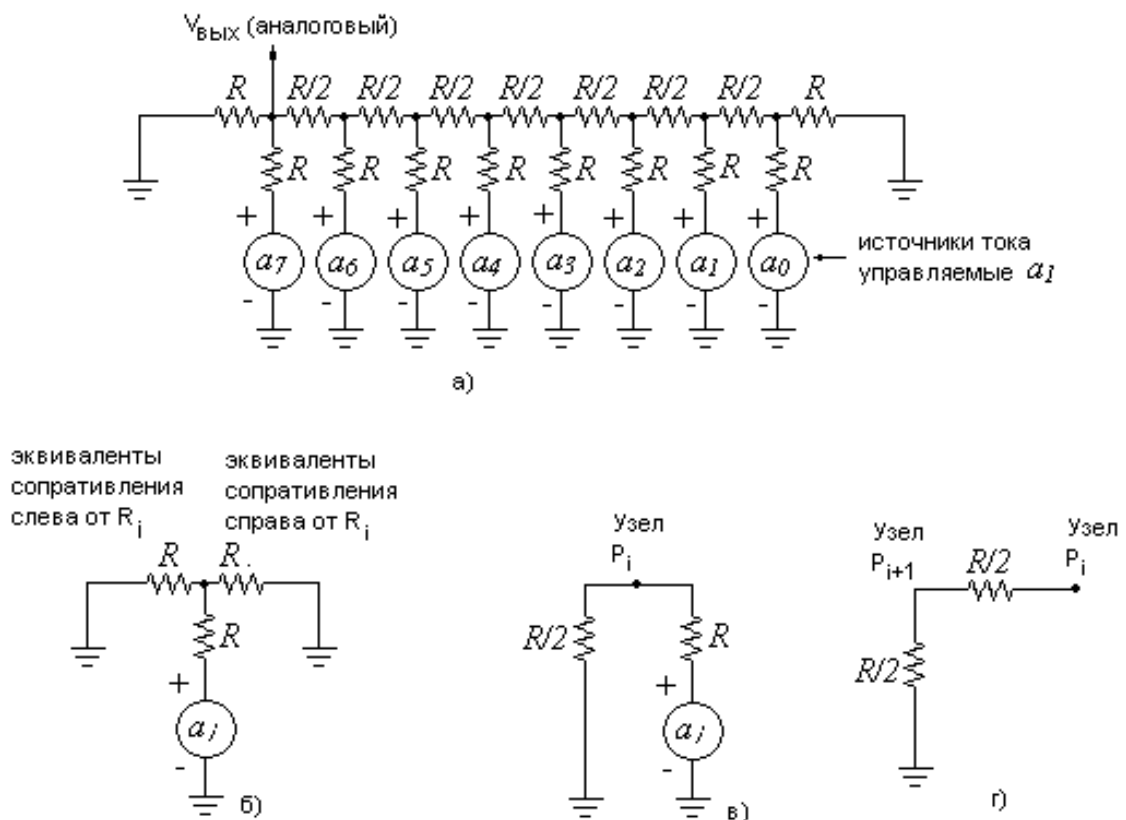
Во многих приложениях МП должны работать с непрерывными физическими параметрами - это задача управления механическими устройствами. Обычно мы сталкиваемся с необходимостью опрашивать или управлять такими параметрами, как скорость, координаты в пространстве, сила. В конечном счёте, эти параметры представляются значениями электрических величин, которые мы будем считать напряжениями. Напряжение, которое представляет физический параметр, мы будем называть аналоговой величиной. Аналоговые напряжения генерируются соответствующими датчиками (например: тахометром для скорости, датчиком давления для давления и силы, потенциометром для координаты в пространстве). При управлении параметрами их физические значения, генерируются по соответствующему напряжению с помощью различных приводов (моторов для сил, скоростей или координат, нагреватели для температуры и т.п.). Микропроцессор по самой своей природе устройства цифровые и работают с цифровым представлением величин, чтобы

микропроцессоры могли работать с аналоговыми величинами, нужны средства преобразования цифровых величин в аналоговые и обратно.

Цифровое двоичное представление величины N состоит из k -битовой цепочки. Значение величины определяется полиномом:

$$N = a_{k-1} 2^{k-1} + a_{k-2} 2^{k-2} + \dots + a_1 2 + a_0.$$

Вычисление полинома сводится к домножению отдельных битов на весовые коэффициенты, являющиеся степенями двойки, и в сложении получаемых членов. Это соответствует вычислению линейной комбинации битов. Подобные линейные комбинации реализуются схемами из резисторов, если мы располагаем напряжениями представляющими значения отдельных битов, то с помощью схемы из резисторов мы можем получить напряжение соответствующее значению всего двоичного числа. Одна из таких схем по внешнему виду напоминающая «лестницу» (в технической литературе называемая аттенюатором) представлена на рисунке 15.1.



а) «лестница» резисторов; б) эквивалентные сопротивления левой и правой части схемы относительно узла P_i ; в) эквивалентный делитель напряжения, определяющий действие a_i на напряжение в узле P_i ; г) эквивалентный делитель напряжения, определяющий напряжение в узле P_{i+1} в зависимости от напряжения в узле P_i .

Рисунок 15.1 - Цифро-аналоговый преобразователь на резисторах

Число битов предполагается равным 8, к схеме подключаются 8 идеальных генераторов напряжения, соответствующих отдельным битам a_i , каждый генератор даёт напряжение либо 0 В или 1 В, в зависимости от значения соответствующего бита (значение напряжения на генераторе численно равно значению бита). Выходное напряжение $V_{\text{вых}}$ пропорционально сумме значений напряжения на генераторах, взвешенных по степеням 2. Таким образом «лестница» резисторов выполняет функцию цифро-аналогового преобразователя числа N.

Напряжение в узле:

$$P_7 = \left(\frac{1}{2}\right)^{7-i} \frac{a_i}{3}.$$

Аналогово-цифровой преобразователь (АЦП).

При любом способе всё начинается с некоторого исходного испытуемого числа, которое преобразуется ЦАП в аналоговую форму и сравнивается с заданным аналоговым напряжением (рисунок 15.2).



Рисунок 15.2 - Аналого-цифровой преобразователь с суммирующим счётчиком

По результатам сравнения испытуемое число корректируется, затем попытка повторяется с новым числом и т.д., пока не будет подобрано число соответствующее заданному напряжению, при этом предполагается, что входное аналоговое напряжение промасштабировано и соответствует диапазону ЦАП. Простейший метод подбора - это начать с наименьшего числа в диапазоне (с 0- при положительных числах) и последовательно увеличивать его на 1 до тех пор, пока преобразованное напряжение не превысит входное и не сравняется с ним.

На рисунке 15.2 показано, как это можно реализовать с помощью суммирующего счётчика. Выходы счётчика поданы на ЦАП. Аналоговый выход этого преобразователя и заданное входное напряжение поданы на компаратор (схему сравнения), на выходе которого появляется логическая «1», если заданное напряжение превышает полученное ЦАП. Выход от компаратора и сигналы от генератора импульсов подаются на вентиль И, выход последнего подаётся на счётный вход счётчика. Процесс аналогово-цифрового преобразования начинается со сброса счётчика в 0, после этого счётчик начинает увеличиваться, пока на выходе компаратора не появится логический «0». Это произойдёт, когда содержимое счётчика окажется достаточным для того, чтобы выход ЦАП превысил или сравнялся с входным напряжением. Достигнутое к этому моменту содержимое счётчика и берётся в качестве выходного преобразованного числа [5].

Дискретизация и квантование сигналов в микропроцессоре.

Квантование. Передача телемеханических сообщений осуществляется как непрерывными, так и дискретными сигналами. По ряду причин, в телемеханике все чаще используют дискретные сигналы. В частности, передача дискретными сигналами обеспечивает более высокую помехоустойчивость и воспроизведение информации с большей точностью. В то же время первичные величины, которые снимаются с датчиков и подлежат передаче системами телеизмерения, являются во многих случаях непрерывными и их необходимо преобразовать в дискретные. Замена непрерывной величины дискретной осуществляется с помощью квантования.

Виды сообщений и квантование. Величины, характеризующие производственный процесс, обычно принимают случайные значения, т. е. являются случайными.

Если случайная величина может принимать конечное число значений, то ее называют дискретной или, точнее, дискретной по множеству. Если случайная величина может принимать бесконечное число значений, то ее называют непрерывной или непрерывной по множеству. Так как при управлении и контроле передаются сообщения о случайной величине, то и передаваемые сообщения о различных показателях производственного процесса по своему характеру можно разбить на две большие группы: непрерывные и дискретные. Например, уровень продукта (бензин, нефть и др.), который хранится в резервуарах нефтеперевалочных баз и отпускается потребителям, желательнее измерять с максимальной точностью, так как ошибка в 1 мм при диаметре цистерны 10-12 м и вместимости 2000-5000 т влечет за собой недоотпуск или перерасход, продукта примерно на 0,5 т. Такая большая точность требует передачи, как правило, непрерывных сообщений.

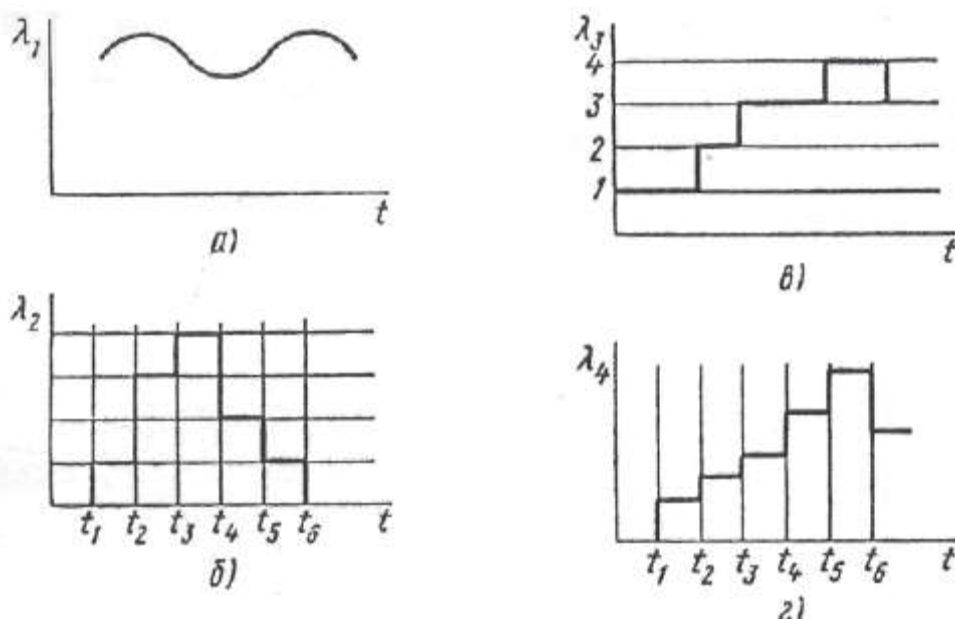
Если же, например, об уровне продукта нужно передать лишь несколько значений - минимальное (резервуар пуст), максимальное (резервуар полон) и ряд промежуточных, то такие сообщения будут дискретными [1,5]:

- непрерывные, по множеству и времени или просто непрерывные (рисунок 15.3, а). Функция $\lambda_1(t)$, описывающая такие сообщения, имеет непрерывное множество значений и изменяется непрерывно во времени; в телемеханике к таким сообщениям относятся так называемые текущие телеизмерения;

- дискретные, по множеству и времени, или просто дискретные (рисунок 15.3,б). Функция $\lambda_2(t)$ может принимать в заранее заданные моменты времени (t_1, t_2, \dots, t_n) только определенные значения. Частным случаем таких сообщений являются двухпозиционные сообщения в телемеханике. Например, «открыто – закрыто», «пусто – наполнено» (телесигнализация) или «включить — отключить» (телеуправление), если изменение состояний происходит в дискретные моменты времени;

- непрерывные, по времени и дискретные по множеству (рисунок 15.3,в). Функция $\lambda_3(t)$ может принимать заранее заданные значения и изменять их в произвольные моменты времени. Других значений, кроме 1, 2, 3, ..., т.е. находящихся между этими точками, функция принимать не может;

- непрерывные, по множеству и дискретные по времени (рисунок 15.3,г). Функция $\lambda_4(t)$ изменяет свое значение в определенные фиксированные моменты времени (t_1, t_2, \dots, t_n) и может принимать любые значения.



а – непрерывные по множеству и времени; б – дискретные по множеству и времени; в – непрерывные по времени и дискретные по множеству; г – непрерывные по множеству и дискретные во времени.

Рисунок 15.3 - Виды сообщений в системах управления

Список литературы

1 Ауэзова А.М. Интегрирование цифровой техники в системы автоматизации управления. Методические указания к выполнения лабораторных работ для магистрантов специальности 6М070200 – Автоматизация и управление. – Алматы: АУЭС, 2014. – 45 с.

2 Карлащук В.И. Электронная лаборатория на IBM PC. ЕWB. – М.: Солон - Пресс, 2005. - 512 с

3 Кабасова Ж.К., Шакенова Ж.К., Ауэзова А.М., Самуратова Б.Н. Электроника. – Жезказган: ЖезУ, 2004. - 77 с.

4 Мюллер С. Модернизация и ремонт ПК. – М.: Вильямс, 2008. – 1184 с.

5 Ауэзова А.М. Элементы и устройства автоматики. Учебное пособие. – Алматы: АУЭС, 2013. – 116 с.

Алма Мухамбетжановна Ауэзова

ИНТЕГРАЛЬНАЯ И МИКРОПРОЦЕССОРНАЯ СХЕМОТЕХНИКА

Конспект лекций для студентов специальности
5В071600 - Приборостроение

Редактор Л.Сластикина
Специалист по стандартизации Н.К.Молдабекова

Подписано в печать __. __. __.
Тираж 50 экз.
Объем 4,8 уч.-изд. л.

Формат 60x84 1/16
Бумага типографская № 1
Заказ ____ .Цена 2400 тенге.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, ул. Байтурсынова, 126