

1 ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ЦИФРОВЫХ СИСТЕМАХ

1.1 СИСТЕМЫ СЧИСЛЕНИЯ

В различные исторические периоды развития человечества для подсчетов и вычислений использовались различные способы представления чисел.

Совокупность приемов и правил наименования и обозначения чисел, с помощью которых можно установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов называют СИСТЕМОЙ СЧИСЛЕНИЯ.

Существуют различные системы счисления. Примеры:

Счет предметов с помощью палочек:

												= 12 =	XII
единичная						десятичная						римская	

ДВЕНАДЦАТЕРИЧНАЯ система: английская система мер:

1 фут = 12 дюймов

1 шиллинг = 12 пенсов

ШЕСТИДЕСЯТЕРИЧНАЯ система: Вавилон

1 час = 60 минут; 1' = 60"

1° = 60' ; 1' = 60"

ДЕСЯТИЧНАЯ система - возникла в Индии, перенесена в Европу арабами, получила название арабской.

Возьмем для примера десятичное число 12 и посмотрим, каким образом оно получается в десятичной системе счисления:

"Две - на - дцать"

$$12 = 1 * 10^1 + 2 * 10^0 = 12$$

где положение чисел 1 и 2 определяется степенью числа 10.

Аналогично: $453 = 4 * 10^2 + 5 * 10^1 + 3 * 10^0 = 400 + 50 + 3$

Истинное значение каждой цифры определяется ее местом в числе, т.е. степенью числа 10 - основанием системы счисления.

Система счисления, в которой значение цифры в числе определяется ее местоположением (позицией), называется ПОЗИЦИОННОЙ.

В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ основанием является число 2. В этом случае для записи чисел используют всего две цифры: 0 и 1. Возьмем, например, число 12 и разложим его по степеням 2.

$$\text{Получим: } 12 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0$$

Число 12 в двоичной системе запишется как: $1100_2 = 12_{10}$

42	2				
- 42	21	2			
0	20	10	2		
	1	10	5	2	
		0	4	2	2
			1	2	1
				0	
V	V	V	V	V	V
0	1	0	1	0	1

] > 101010 = 42

Перевод числа из десятичной в двоичную производится методом последовательного деления числа на 2 до тех пор, пока частное от деления не станет равным 1.

Число в двоичной системе записывается в виде остатков от деления, начиная с последнего частного, справа налево:

Дробная часть представляется суммой отрицательных степеней числа 2. Например, $0.25 = 2^{-2}$.

$$0.8125_{10} = 0.5 + 0.25 + 0.0625 = 1 * 2^{-1} + 1 * 2^{-2} + 0 * 2^{-3} + 1 * 2^{-4} = 0.1101_2$$

При переводе дробей в двоичный код в большинстве случаев результат получается приближительный, поэтому необходимо задавать точность преобразования с нужным количеством знаков после запятой.

Преимуществом двоичной системы счисления является то, что она использует только две цифры. Поэтому в цифровой технике для представления логических сигналов и выполнения операций над числами в двоичной системе счисления достаточно двух значений напряжения, соответствующих уровням логического нуля и логической единицы. Микропроцессорные устройства оперируют с информацией, представляемой комбинацией двоичных цифр – **битов** (могут принимать значения 0 или 1). Группы битов называют словами. Длина слова – разрядность различна и может составлять 4, 8, 16, 32 и более битов. Слово разрядностью 8 бит называют **байтом**. Диапазон представления чисел одним байтом составляет от 0 до $2^8 - 1$, т.е. всего 256 значений.

При необходимости представления двоичных чисел со знаком старший разряд отводится для знака числа (0 – число положительное, 1 – отрицательное). В этом случае диапазон представления однобайтных чисел составляет от -127 до +127.

При написании программ на языках низкого уровня или в кодах МП и при обработке данных широко используются еще две системы счисления.

ВОСЬМЕРИЧНАЯ система в качестве основания использует число 8 и, соответственно, 8 цифр от 0 до 7. Перевод из десятичной системы в восьмеричную осуществляется по тому-же правилу, что и в случае с двоичной системой. Перевести число из двоичной в восьмеричную систему еще проще. Надо число, представленное в двоичном виде, сгруппировать справа налево по три цифры и каждую группу отдельно перевести по правилу перевода из двоичной системы в десятичную. Обратное преобразование – в обратном порядке. Например:

$$\text{число } 453_{10} = 111'000'101_2 = 705_8$$

ШЕСТНАДЦАТЕРИЧНАЯ система в качестве основания использует число 16 и, соответственно, цифры от 0 до 9 и первые 6 букв латинского алфавита A, B, C, D, E, F. При переводе числа из двоичной системы в шестнадцатеричную надо число, представленное в двоичном виде, сгруппировать справа налево по четыре цифры и каждую группу отдельно перевести по правилу перевода из двоичной системы в десятичную. Например:

$$\text{число } 453_{10} = 1'1100'0101_2 = 1C5_{16} = 1C5h = 0x1C5$$

ДВОИЧНО-ДЕСЯТИЧНАЯ система применяется в тех случаях, когда результат необходимо представить в удобном для восприятия человеком виде (на цифровом индикаторе, ЦПУ и др.). При этом каждая цифра десятичного числа отдельно переводится в двоичный код, причем результат представляется в четырех разрядах и при необходимости дополняется нужным количеством нулей. Окончательный результат получается при записи полученных двоичных чисел подряд, в том порядке, какой был в десятичном числе. Например:

$$\begin{array}{ccccccc} & 4 & & 5 & & 3 & \\ \text{число } 453_{10} = & 0100 & ' & 0101 & ' & 0011 & \end{array}$$

часто называют системой 8-4-2-1
в западной литературе система BCD

1.2 ПРАВИЛА ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ

СЛОЖЕНИЕ чисел, представленных в двоичном коде, выполняется поразрядно, начиная с младшего разряда. В результате сложения двух первых кодов слагаемых X_0 , Y_0 получается первый разряд суммы S_0 и код переноса P_0 в следующий разряд. В следующих разрядах код S_i будет определяться с учетом переноса из соседнего младшего разряда:

$$\begin{array}{r} 7 \quad 0111 \\ 5 \quad 0101 \\ + \text{---} \quad + \text{---} \\ 12 \quad 1100 \end{array}$$

ОПЕРАЦИЯ ВЫЧИТАНИЯ в ЭВМ выполняется так же как и сложение, но при этом отрицательные числа представляются в дополнительном или обратном коде. Смысл перевода отрицательных чисел из прямого в дополнительный и обратный коды поясним на примере с десятичными числами.

Допустим, требуется сложить числа $X_1=76$ и $X_2=-58$. Заменяем код отрицательного слагаемого X_2 его дополнением до 100, так чтобы $[X_2]_{\text{доп}}=100+X_2=42$. Сложив числа $X_1+[X_2]_{\text{доп}}$ получим:

$$Y=X_1+[X_2]_{\text{доп}}=76+42=118$$

Отбрасывая 1 старшего разряда получим искомый результат 18. Равенство полученного результата истинному объясняется тем, что при формировании дополнительного кода к X_2 мы прибавляли 100, а из результата вычитали 100 отбрасыванием старшего разряда.

$$Y=X_1+[X_2]_{\text{доп}}-100=X_1+[X_2+100]-100=76+[-58+100]-100=18$$

При записи двоичного числа в прямом коде в знаковом разряде ставится его знак (0 - плюс, 1 - минус), а само число записывается в естественной форме:

$$X=13_{10} \quad [X]_{\text{пр}}=01101_2$$

$$X=-13_{10} \quad [X]_{\text{пр}}=11101_2$$

ДОПОЛНИТЕЛЬНЫЙ КОД отрицательных двоичных чисел получается заменой двоичных кодов во всех разрядах на взаимно обратные (0 на 1, 1 на 0). После этого к младшему разряду числа добавляется 1. В знаковом разряде отрицательного числа записывается 1.

$$[-14]_{\text{доп}}=[-01110]_{\text{доп}}=[10001+1]=10010$$

Кроме дополнительного кода для представления отрицательных чисел используется ОБРАТНЫЙ КОД. В этом случае в знаковом разряде записывается 1, а в остальных разрядах цифры заменяются на взаимно обратные

$$[-14]_{\text{обр}} = [-01110]_{\text{обр}} = 10001$$

При выполнении арифметических операций с отрицательными числами производится поразрядное сложение слагаемых, начиная с младшего и кончая знаковым разрядом. Если используется дополнительный код, то возможная единица переноса из знакового разряда отбрасывается, при использовании обратного кода единица переноса знакового разряда суммируется с младшим разрядом полученной суммы. Результат вычисления получается в том коде, в каком были представлены слагаемые. Положительные числа в прямом, обратном и дополнительном кодах имеют одну и ту же форму записи. Например:

12	X1пр=0.1100	X1обр=0.1100	X1доп=0.1100
5	X2пр=0.0101	X2обр=1.1010	X2доп=1.1011
- 7	Sпр= 0.0111	10.0110	10.0111
		└─>─┐	<─┐
		Sобр= 0.0111	Sдоп= 0.0111
Y=5-12=-7		X1обр=0.0101	X1доп=0.0101
		X2обр=1.0011	X2доп=1.0100
		+ Sобр= 1.1000	+ Sдоп= 1.1001
		Sпр= -0.0111	Sдоп= -0.0111

УМНОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ, представленных в форме с фиксированной точкой, включает в себя определение знака и абсолютного значения произведения. Знаковый разряд произведения может быть получен суммированием знаковых разрядов сомножителей без формирования переноса (так называемое суммирование по модулю 2). Действительно, при совпадении цифр знаковых разрядов сомножителей (0 и 0, либо 1 и 1) их сумма по модулю 2 равна 0, т.е. соответствует знаковому разряду произведения двух сомножителей, имеющих одинаковые знаки; при несовпадении цифр знаковых разрядов эта сумма будет равна 1, что также соответствует знаковому разряду произведения двух сомножителей с разными знаками. Абсолютное значение произведения получается путем перемножения чисел без учета их знаков (так называемого кодового

умножения) .

Пусть производится умножение чисел $13_{10} = 1101_2$ и $11_{10} = 1011_2$

13	1101
X	X
11	1011
-----	-----
13	1101
+	+
13	1101
-----	0000
143	1101

	10001111

Как видно из примера, в процессе выполнения операции умножения формируются частичные произведения (произведения множимого на цифры разрядов множителя), которые суммируются с соответствующими сдвигами друг относительно друга. В цифровых устройствах процессу суммирования частичных произведений придают последовательный характер: формируется одно из частичных произведений, к нему с соответствующим сдвигом прибавляется следующее частичное произведение, к полученной сумме прибавляется с соответствующем сдвигом очередное частичное произведение и т.д., пока не окажутся просуммированными все частичные произведения. Этот процесс суммирования можно начинать с младшего либо старшего частичного произведения.

При умножении целых чисел для фиксации произведения в разрядной сетке должно предусматриваться число разрядов, равное сумме числа разрядов множимого и множителя.

ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ. Будем рассматривать операцию алгебраического деления чисел, представленных в форме с фиксированной точкой. при этом выполнение операции содержит действия, связанные с определения модуля частного. Знак частного может быть найден тем же приемом, что и знак произведения в рассмотренной выше операции умножения с отделением знаковых разрядов.

На рисунке 1 показана схема алгоритма нахождения частного положительных чисел a и b .

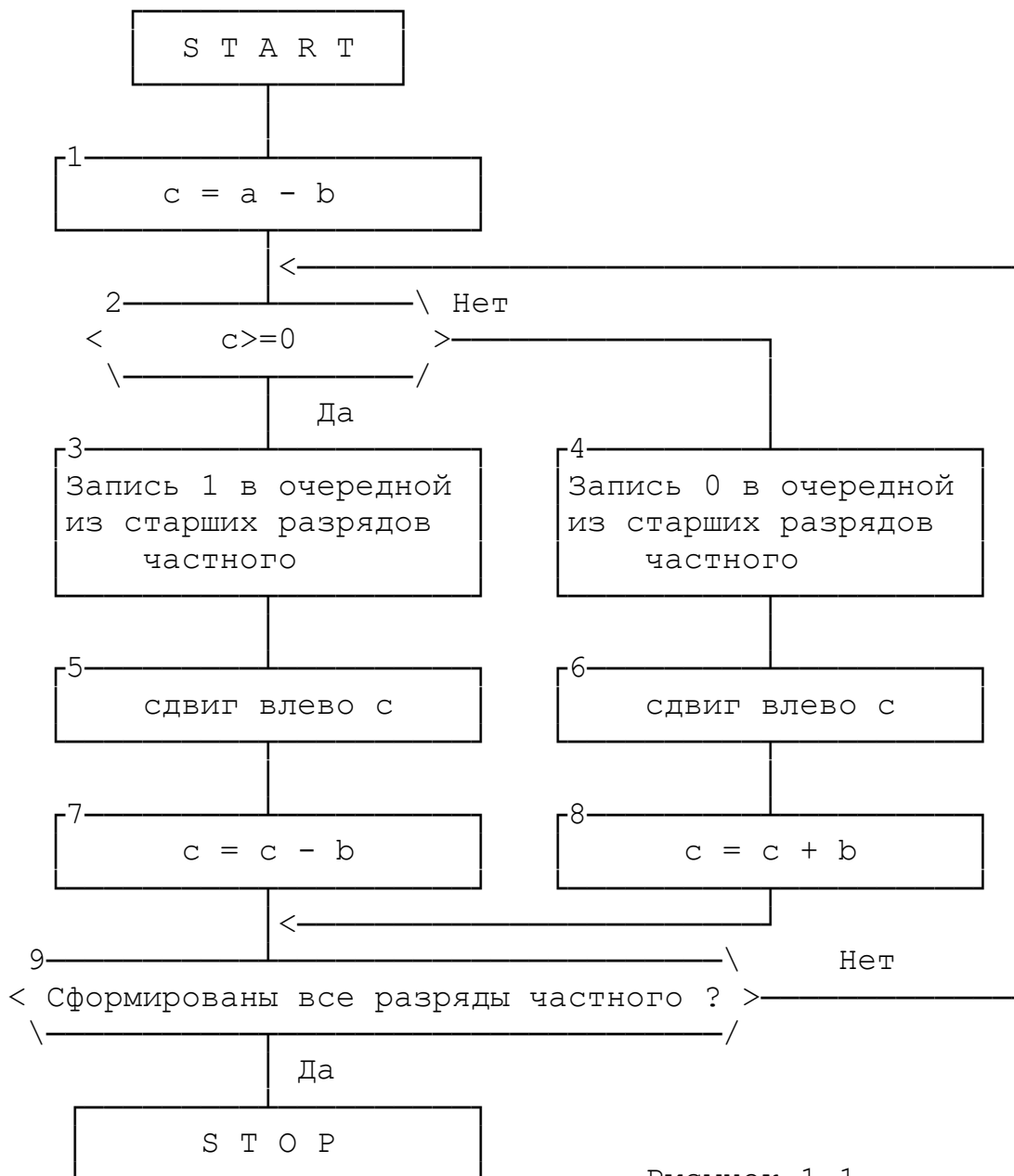


Рисунок 1.1

Покажем выполнение операции на примере. Пусть после отделения знаковых разрядов модули делимого и делителя представляются соответственно числами $a=0.10010$ и $b=0.10110$.

Встречающуюся в алгоритме операцию вычитания числа заменим прибавлением числа $-b$, представленного в дополнительном коде: $(-b)_{\text{доп}}=1.01010$.

a	0.10010		0. 1 0 1 1 0	
$(-b)_{\text{доп}}$	1.01010			
	+		0. 1 1 0 1 0	Частное
c	1.11100	$c < 0$		
Сдвиг влево	1.11000			
b	0.10110			
	+			
c	0.01110	$c > 0$		
Сдвиг влево	0.11100			
$(-b)_{\text{доп}}$	1.01010			
	+			
c	0.00110	$c > 0$		
Сдвиг влево	0.01100			
$(-b)_{\text{доп}}$	1.01010			
	+			
c	1.10110	$c < 0$		
Сдвиг влево	1.01100			
b	0.10110			
	+			
c	0.00010	$c > 0$		
Сдвиг влево	0.00100			
$(-b)_{\text{доп}}$	1.01010			
	+			
c	1.01110	$c < 0$		

Проверка: $a=0,10010_2 = 0,5625_{10}$
 $b=0,10110_2 = 0,6875_{10}$
 $a/b=0,11010_2 = 0,8125_{10}$

2 ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

Все устройства МП состоят из элементарных логических схем. Работа этих схем основана на законах и правилах алгебры логики, которая оперирует двумя понятиями: Истинности и ложности высказывания. В соответствии с такой двоичной природой высказываний их условились называть ЛОГИЧЕСКИМИ ДВОИЧНЫМИ ПЕРЕМЕННЫМИ и обозначать 1 в случае истинности и 0 в случае ложности. Примерами логических переменных являются высказывания:

$A = \text{"Трава красная"} , B = \text{"Вода мокрая"}$. На основании этих высказываний можно записать $A=0; B=1$, т.к. высказывание A ложно, а высказывание B истинно.

Высказывания могут быть простыми и сложными: простые содержат одно законченное утверждение, сложные образуются из двух или большего числа простых высказываний, связанных между собой некоторыми логическими связями наподобие «если – то».

Формализация и преобразование связей между логическими переменными осуществляется в соответствии с правилами АЛГЕБРЫ ЛОГИКИ называемой АЛГЕБРОЙ БУЛЯ.

Две логические переменные A и B принимающие значения 0 или 1, могут образовывать логические функции. Для описания логической функции используют, так называемую, таблицу истинности, в левых колонках которой отображают все возможные значения логических переменных, а справа соответствующие значения функции. Таким образом, каждая строка таблицы соответствует определенному состоянию функции или устройства, ее реализующего.

Кроме того, действие логической функции можно проиллюстрировать с помощью временных диаграмм, которые показывают изменение во времени входных логических сигналов и соответствующее им изменение выходного сигнала или сигналов.

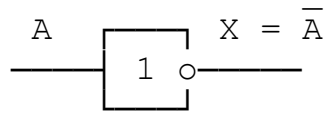
Наибольший практический интерес представляют функции отрицания, логического умножения и логического сложения которые считаются основными функциями алгебры логики. С их помощью можно реализовать любую, сколь угодно сложную, логическую функцию.

1. Логическое отрицание НЕ переменной A есть логическая функция X, которая истинна, когда A ложно, и наоборот. Инвертор.

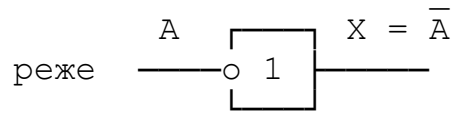
Графически функция НЕ обозначается в виде круга возле вывода (МЭК-117-15) (ГОСТ 2.743-91):

$$X = \bar{A}$$

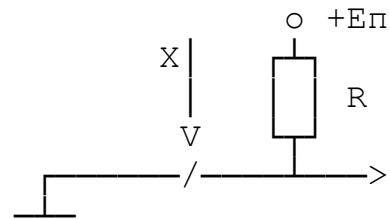
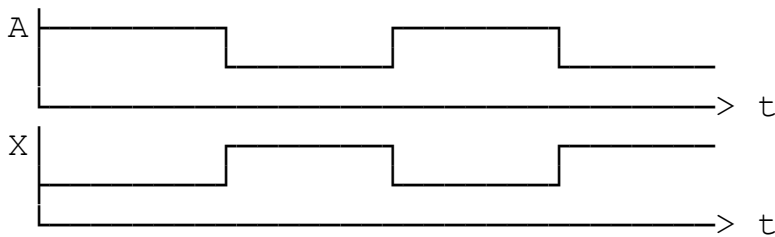
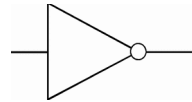
A	X
0	1
1	0



или



По американской спецификации milspec:

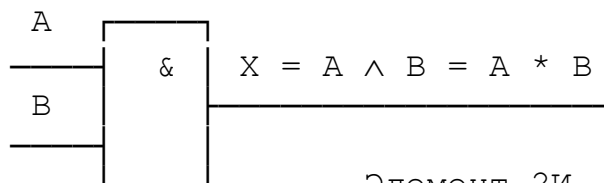


2. Логическое умножение И переменных есть логическая функция, которая истинна только тогда, когда одновременно истинны обе входные переменные. Кроме того функцию И называют КОНЪЮНКЦИЕЙ. Логическое умножение может быть функцией двух и более переменных.

$$X = A * B \text{ или } X = A \wedge B$$

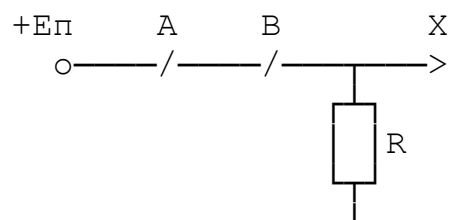
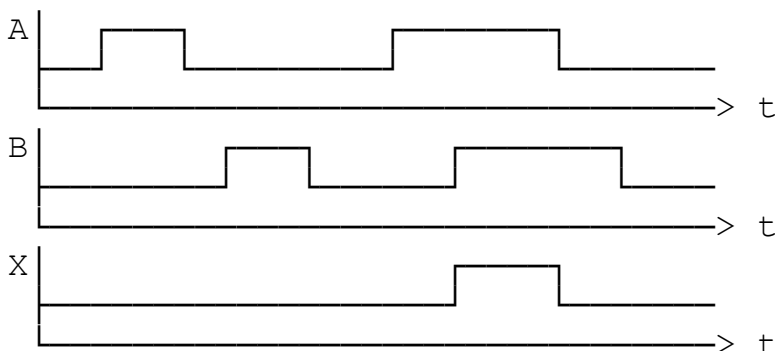
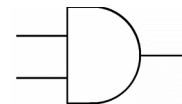
Графически функция И обозначается в виде значка «&» внутри элемента

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



Элемент 2И

По американской спецификации milspec:

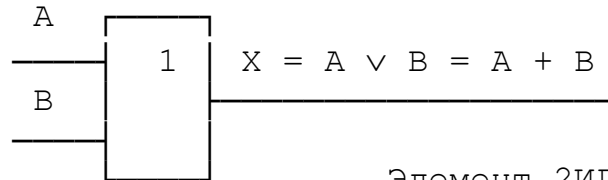


3. Логическая сумма ИЛИ переменных есть логическая функция, которая истинна, когда истинна хотя бы одна из входных переменных. Кроме того функцию ИЛИ называют ДИЗЬЮНКЦИЕЙ. Логическое сложение может быть функцией двух и более переменных.

$$X = A + B \text{ или } X = A \vee B$$

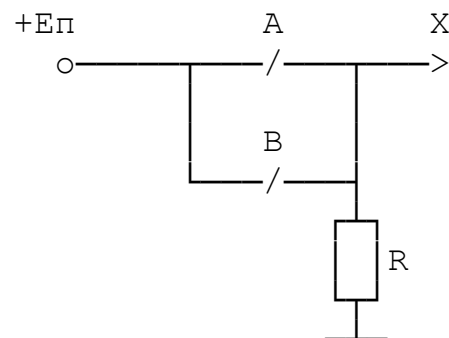
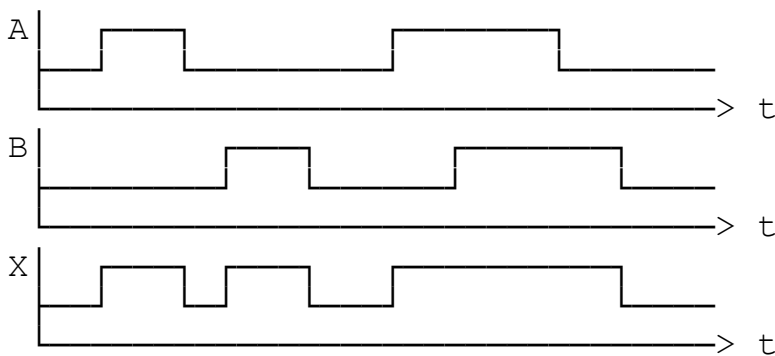
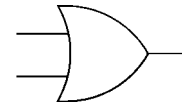
Графически функция ИЛИ обозначается в виде значка 1 внутри элемента

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

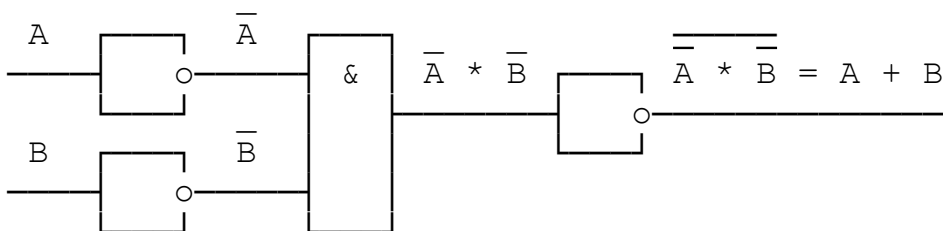


Элемент 2ИЛИ

По американской спецификации milspec:



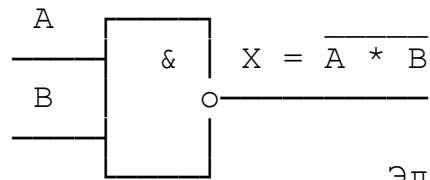
Три рассмотренных функции позволяют реализовать любую логическую комбинацию. Более того, сочетание одной из функций И либо ИЛИ с функцией НЕ также позволяет реализовать любую сложную функцию. Например, реализуем функцию ИЛИ с помощью элементов И и НЕ.



Рассмотрим еще три функции получившие широкое применение в цифровой технике:

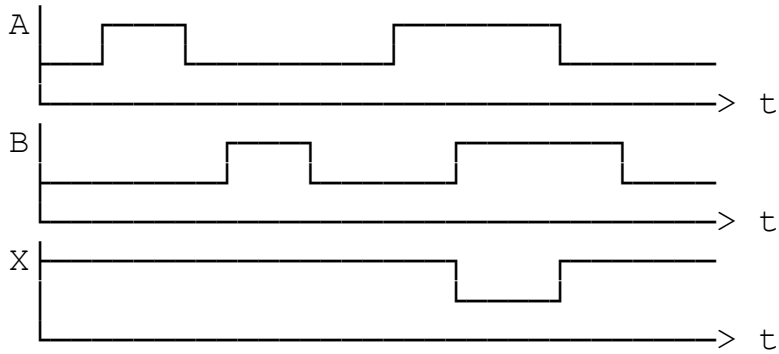
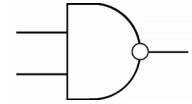
4. И-НЕ "Штрих Шеффера"

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



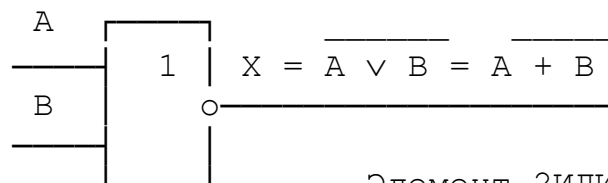
По американской спецификации milspec:

Элемент 2И-НЕ



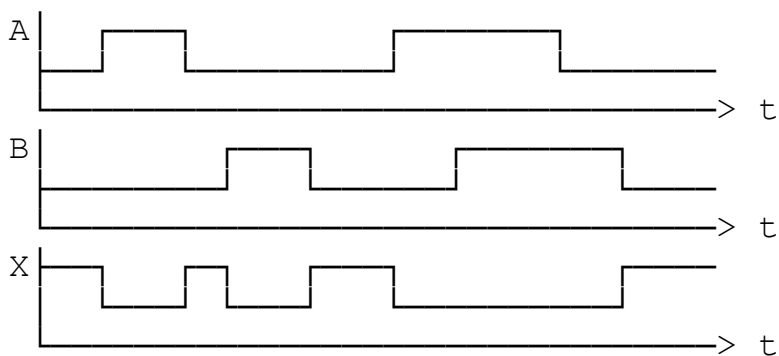
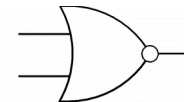
5. ИЛИ-НЕ "Стрелка Пирса"

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



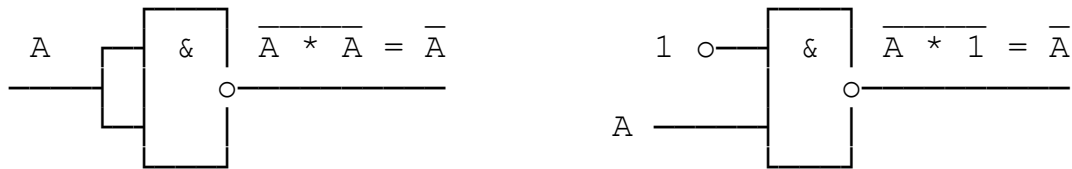
По американской спецификации milspec:

Элемент 2ИЛИ-НЕ



Две рассмотренные функции получили название базисных (или базовых) интересны тем, что являются достаточными для построения любой сложной функции, т.к. содержат в своем составе функцию И либо ИЛИ и инвертор.

Пример реализации инвертора на основе элемента 2ИЛИ-НЕ:



6. "ИСКЛЮЧАЮЩЕЕ ИЛИ" или СУММИРОВАНИЕ ПО МОДУЛЮ 2

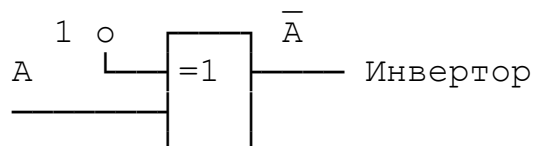
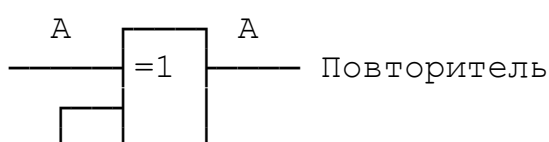
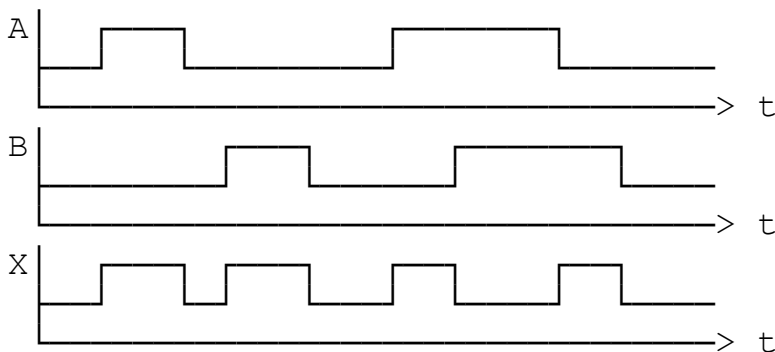
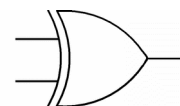
"ИСКЛЮЧАЮЩЕЕ ИЛИ" двух переменных есть логическая функция, которая истинна, когда входные переменные имеют разные логические значения.

"ИСКЛЮЧАЮЩЕЕ ИЛИ" может быть функцией только двух переменных.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{aligned}
 X &= A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B = \\
 &= \overline{A \cdot B} + \overline{\bar{A} \cdot \bar{B}} = (A+B) \cdot (\bar{A}+\bar{B}) = \\
 &= \overline{(A+B) \cdot (\bar{A}+\bar{B})}
 \end{aligned}$$

По американской спецификации milspec:



Рассмотрим наиболее распространенные правила алгебры логики на примерах с минимальным количеством переменных:

$$\begin{array}{ll}
 1: X = A * 0 = 0 & \backslash \\
 2: X = A * 1 = A & | \text{ законы} \\
 3: X = A * A = A & | \text{ конъюнкции} \\
 4: X = A * \bar{A} = 0 & / \\
 5: X = A + 0 = A & \backslash \\
 6: X = A + 1 = 1 & | \text{ законы} \\
 7: X = A + A = A & | \text{ дизъюнкции} \\
 8: X = A + \bar{A} = 1 & / \\
 9: X = \bar{\bar{A}} = A & \text{двойная инверсия} \\
 10: X = A * B = B * A & \backslash \text{ переместительный} \\
 11: X = A + B = B + A & / \text{ закон} \\
 12: X = (A * B) * C = A * (B * C) & \backslash \text{ сочетательный} \\
 13: X = (A + B) + C = A + (B + C) & / \text{ закон} \\
 14: X = A * (B + C) = AB + AC & \backslash \text{ закон дистрибутивности} \\
 15: X = A + BC = (A + B) * (A + C) & / \text{ или распределительный} \\
 16: X = A + AB = A \quad \{ A + BA = A (1 + B) \} & \backslash \\
 17: X = A (A + B) = A & | \text{ законы} \\
 18: X = A + \bar{A}B = A + B \quad \{ (1 + \bar{A}) (A + B) = A + B \} & / \text{ поглощения} \\
 19: X = AB + \bar{A}B = B & \backslash \text{ законы} \\
 20: X = (A + B) (\bar{A} + B) = B & / \text{ склеивания} \\
 21: X = \overline{AB} = \bar{A} + \bar{B} & \backslash \text{ правила Де Моргана} \\
 22: X = \overline{\bar{A} + \bar{B}} = \bar{\bar{A}} * \bar{\bar{B}} & / \text{ или законы инверсии}
 \end{array}$$

.

Пример: Задана в виде таблицы истинности логическая функция Y трех переменных A, B, C.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Функция принимает единичное значение при следующих произведениях переменных:

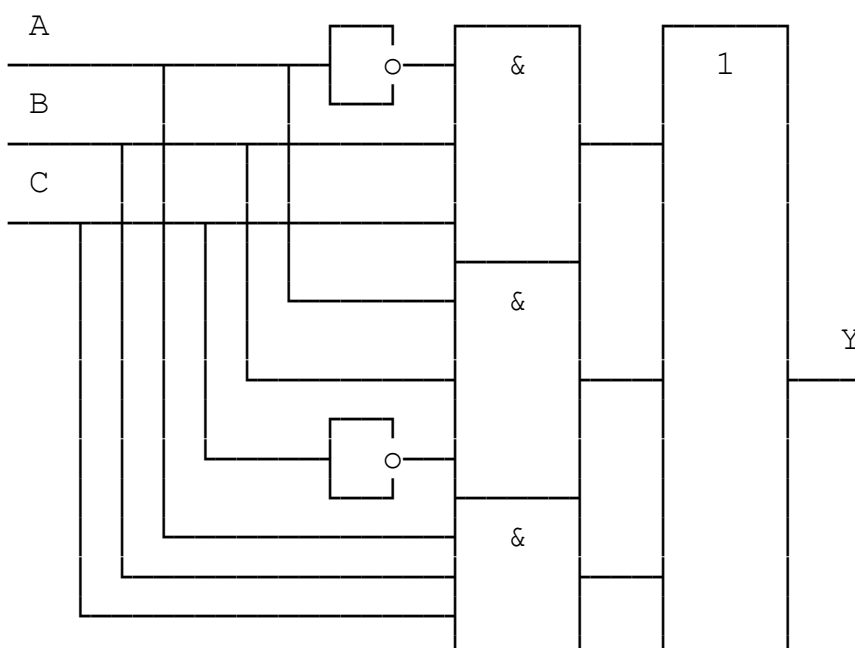
$$\bar{A}BC, A\bar{B}C, ABC$$

Каждое из произведений переменных для которых значение функции истинно, носит название МИНТЕРМА.

В соответствии с этим, функция может быть записана в виде уравнения:

$$Y = \bar{A}BC + A\bar{B}C + ABC$$

Если при такой записи каждое слагаемое содержит произведения всех переменных или их отрицаний, то такую форму представления функции называют СОВЕРШЕННОЙ ДИЗЬЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМОЙ или ПЕРВОЙ СТАНДАРТНОЙ ФОРМОЙ. Точно так же можно выделять ложные (нулевые) значения функции. Если функция дана в виде произведения (конъюнкции) сумм переменных или их отрицаний, то такую форму представления функции называют СОВЕРШЕННОЙ КОНЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМОЙ или ВТОРОЙ СТАНДАРТНОЙ ФОРМОЙ.



Для схемной реализации полученной функции потребуется:

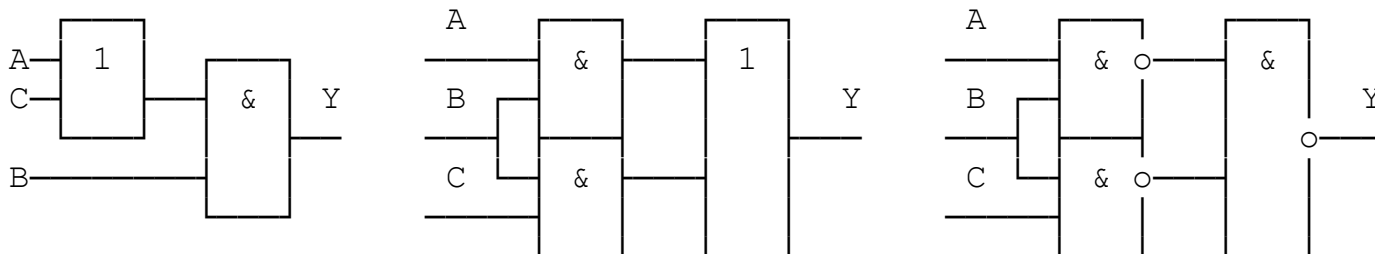
- 3 схемы, выполняющие функцию ЗИ,
- 1 схема ЗИЛИ и
- 2 схемы НЕ.

Пользуясь правилами алгебры логики упростим полученную функцию.

$$Y = \bar{A}BC + A\bar{B}C + ABC = \bar{A}BC + AB(\bar{C} + C) = \bar{A}BC + AB = B(\bar{A}C + A) =$$

$$= (\text{правило 15}) = B(\bar{A} + A)(C + A) = B(C + A) = BC + AB$$

Полученная функция и ее схемная реализация значительно проще исходных.



$$Y = \overline{\overline{Y}} = \overline{AB + BC} = \overline{AB} * \overline{BC} - \text{для построения схемы на одинаковых элементах}$$

Можно было представить ту же самую функцию в СОВЕРШЕННОЙ КОНЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЕ в виде произведения сумм:

$$Y = (A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)(\bar{A} + B + \bar{C})$$

Такая запись слишком сложна для минимизации и реализации.

Разработан метод минимизации логических функций, как бы автоматизирующий процедуру поиска "склеивающихся слагаемых - метод КАРТ КАРНО.

Карта Карно - это таблица имеющая ячейки для всех возможных минтермов функции. Можно построить карты Карно для функций, минтермы которых содержат два, три и более переменных (обычно не более 5...6). Составим карту Карно для функции 2-х переменных: Вдоль верхней грани проставлены возможные значения

	A	$\bar{A}$
B	AB	$\bar{A}B$
$\bar{B}$	$A\bar{B}$	$\bar{A}\bar{B}$

переменной A, вдоль левой боковой грани - возможные значения переменной B. В каждой клетке изображают один из возможных минтермов: AB, $\bar{A}B$, $A\bar{B}$, $\bar{A}\bar{B}$. Если какой-то из этих минтермов в совершенной дизъюнктивной

нормальной форме записи функции присутствует, то в соответствующей клетке карты Карно ставится "1", если нет - то "0".

Составим карты Карно для функций 3-х и 4-х переменных:

	--	-		-
	AB	AB	AB	AB
-	---	--	-	--
C	ABC	ABC	ABC	ABC
-	---	--	-	--
C	ABC	ABC	ABC	ABC

	--	-		-
	AB	AB	AB	AB
--	----	--	--	----
CD	ABCD	ABCD	ABCD	ABCD
-	---	--	-	--
CD	ABCD	ABCD	ABCD	ABCD
-	---	--	-	--
CD	ABCD	ABCD	ABCD	ABCD
-	---	--	-	--
CD	ABCD	ABCD	ABCD	ABCD

Склеивание осуществляется между теми минтермами, которые записаны в виде "1" в соседних клет-

ках карты (по вертикали или горизонтали). Соседними считаются клетки крайнего левого и правого, верхнего и нижнего рядов (можно представить карту как развертку цилиндра). Два минтерма, находящиеся в соседних клетках можно представить в виде одного логического произведения переменных, число которых на одну единицу меньше, чем в каждом из соседних минтермов, причем в произведении остаются общие для обоих минтермов сомножители. Если соседними окажутся сразу 4 минтерма с "1" то такую группу можно заменить произведением переменных, число которых меньше, чем в каждом минтерме, уже на два. Учитывая, что $A + A + \dots + A = A$, одну единицу, изображающую минтерм, можно объединять пары несколько раз.

Используя метод карт Карно минимизируем рассмотренную в примере функцию. Единица, изображающая минтерм ABC входит сразу в два объединения, обозначенные

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
$\bar{C}$	0	0	1	0
C	0	1	1	0

а и b. Объединение а отражает

"склеивание" минтермов ABC и $AB\bar{C}$

$$ABC + AB\bar{C} = AB (C + \bar{C}) = AB$$

объединение b отражает склеивание минтермов ABC и $\bar{A}BC$

$$ABC + \bar{A}BC = BC (A + \bar{A}) = BC$$

В результате проведенных операций "склеивания" из трех минтермов, входящих в функцию и являющихся конъюнкцией трех переменных, остались лишь слагаемые AB и BC .

Отсюда: $Y = AB + BC$.

3 ОСНОВНЫЕ КОМБИНАЦИОННЫЕ УСТРОЙСТВА

Комбинационным называют устройство, состояние которого однозначно определяется комбинацией входных сигналов и не зависит от предыдущего состояния.

3.1 Логические элементы

Выполняют операции инверсии (НЕ), конъюнкции (И), дизъюнкции (ИЛИ) и более сложные комбинации этих функций.

Логическая ИС может содержать в своем составе несколько элементов одного типа. Логические элементы в функциональном отношении являются автономными и могут применяться в устройстве самостоятельно. Цифровые ИС могут быть построены с применением различных микроэлектронных технологий, но в их основе, как правило, лежат схемы, выполняющие функцию И-НЕ либо ИЛИ-НЕ.

Параметры реальных логических схем зависят от типа микроэлектронной технологии. Для изготовления логических ИС малой и средней степени интеграции наибольшее распространение получили ТТЛШ (транзисторно – транзисторная логика с на основе биполярных транзисторов с барьером Шотки) и КМОП (комплиментарные металл – оксид – полупроводник транзисторы на основе полевых транзисторов с изолированным затвором и индуцированным каналом разной проводимости) (КМДП) технологии. Для построения быстродействующих схем применялась ЭСЛ (эмиттерно связная логика) технология на данный момент устаревшая. Интегрально-инжекционная логика (ИИЛ, И2Л) применяется при изготовлении аналого-цифровых схем – ЦАП, АЦП.

ТТЛШ (ТТЛ) –схемы характеризуются:

- достаточно высоким быстродействием $t_z = 1..5$ нс;

- значительными токами логических сигналов, что обеспечивает определенную помехозащищенность, хорошую нагрузочную способность и, как следствие, высокое энергопотребление (возможно применение в лазерной технике в условиях сильных помех);

- низкой степенью интеграции, следовательно невозможно построение сложных схем в одном корпусе МС;

Особенностью работы ТТЛШ-схем является вытекающий ток, обеспечивающий состояние логического 0 на входе ЛЭ. Это явление обусловлено применением на входе многоэмиттерного транзистора (МЭТ) (см. рисунок 3.1). Поэтому состояние неподключённого входа эквивалентно состоянию лог. 1.

По построению выходного каскада ТТЛШ – схемы различаются на устройства с двухтактным ключом – с двухстабильным состоянием (см. рисунок 3.1), с открытым коллектором и с третьим состоянием (см. рисунок 3.2). Коэффициент разветвления по выходу большинства ТТЛШ-схем – 10.

Большинство серий микросхем ТТЛШ имеют напряжение питания +5 В. Напряжение низкого уровня (лог. 0) определено в диапазоне 0..0,4 В, напряжение высокого уровня (лог. 1) определено в диапазоне 2,4..4,5 В. Выпускаются МС ТТЛШ с пониженным напряжением питания 3,3 В для сопряжения с современными микропроцессорами.

Наиболее распространенные ТТЛ-серии:

Ам. фирма Texas Instruments	отечественные
SN7400/SN5400 (60 гг)	K155/K133
SN74LS00	K555
SN74ALS00	KP1533
SN74F00	KP1531

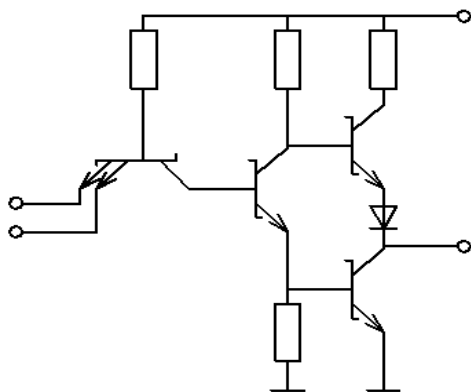
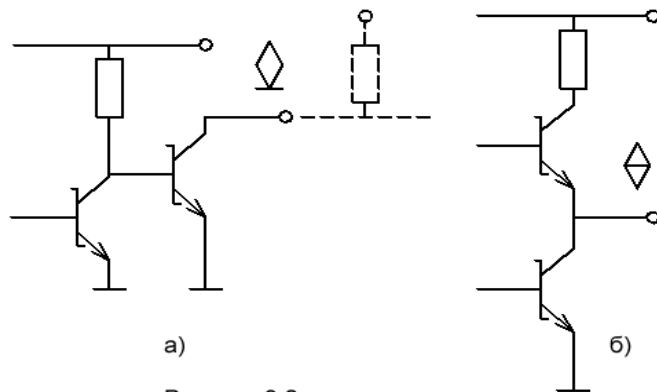


Рисунок 3.1 Элемент 2И-НЕ ТТЛШ



а)

б)

Рисунок 3.2
Выходные каскады ТТЛШ-схем
а) Открытый коллектор ОК
б) с третьим состоянием ТС

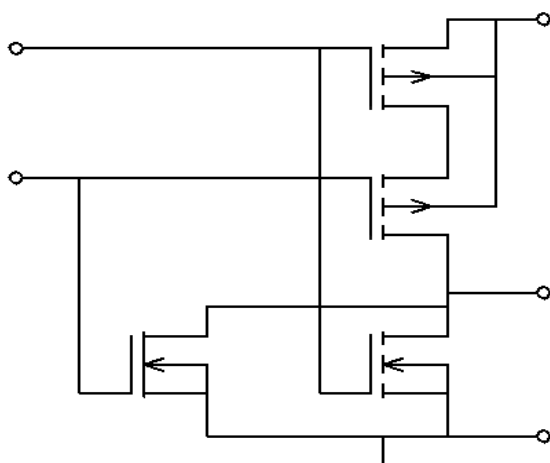


Рисунок 3.3 Элемент 2ИЛИ-НЕ КМОП

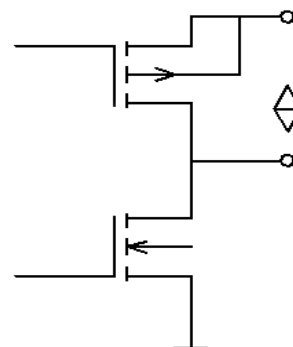


Рисунок 3.4
Выходной каскад КМОП-схемы
с третьим состоянием

КМОП-схемы характеризуются:

- средним и высоким быстродействием $t_z = 2..10$ нс;

- минимальными токами логических сигналов, что обеспечивает среднюю помехозащищенность, минимальную нагрузочную способность и, как следствие, низкое энергопотребление (возможно питание от автономных источников и др.);

- высокой степенью интеграции, следовательно возможно построение сложных схем в одном корпусе МС;

Особенностью работы КМОП-схем является высокое входное сопротивление, обусловленное применением на входе полевого транзистора (см. рисунок 3.3).

По построению выходного каскада КМОП - схемы различаются на устройства с двухтактным ключом - с двухстабильным состоянием (см. рисунок 3.3), с третьим состоянием (см. рис. 3.4) и открытым стоком. Коэффициент разветвления по выходу большинства КМОП-схем - 10.

Большинство серий микросхем КМОП работают с напряжением питания от +3 до +15 В. Напряжение низкого уровня (лог. 0) близко потенциалу общего провода, напряжение высокого уровня (лог. 1) близко потенциалу источника питания.

Наиболее распространенные КМОП-серии:

Ам. фирма RCA	отечественные
CD4000 (60 - 70 гг)	K176
CD4000A	K561
CD4000B	K564
CD4000BE	KP1561
74АС (ТТЛ-подобная)	KP1554
54НС (ТТЛ-подобная)	K1564

Для согласованной работы микросхем, изготовленных по разным технологиям применяются специальные преобразователи уровня.

Комбинируя логические элементы можно реализовать сложные логические функции. Для повышения эффективности разработок, уменьшения числа корпусов МС и повышения быстродействия цифровых устройств разработаны и выпускаются специализированные микросхемы, реализующие определенные функции.

3.2 Шифраторы

Шифраторы (кодеры) – устройства, позволяющие преобразовать цифровой код в другой код, представленный в произвольной системе счисления. Функция преобразования шифратора задается либо в виде переключательной функции, либо в виде таблицы истинности.

На практике часто используется представление входного сигнала в виде позиционного кода, то есть, в любой момент времени активный логический сигнал присутствует только на одном из нескольких входов, при этом выходной сигнал представляется в виде параллельного двоичного кода. Работу такого шифратора можно рассмотреть на основе решения задачи формирования четырёхразрядного двоичного кода, получаемого при нажатии на клавишу цифровой клавиатуры (например, калькулятора или домофона).

Входной позиционный код										Выходной двоичный код			
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	3-й разряд X3	2-й разряд X2	1-й разряд X1	0-й разряд X0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

Каждый разряд двоичного кода описывается отдельной логической функцией:

$$X0=A1+A3+A5+A7+A9;$$

$$X1=A2+A3+A6+A7;$$

$$X2=A4+A5+A6+A7$$

$$X3=A8+A9$$

Такой шифратор можно реализовать на основе диодных сумматоров по схеме на рисунке 3.5.

На выходах этой матрицы появляется двоичный код, соответствующий номеру той входной шины, которая подключена перемычкой П1 к источнику постоянного положительного напряжения $V1$. Напряжение источника $V1$ должно быть больше порога отпираания диодов $U_{д.пр}$. Выходной код определяется позициями, в которых установлены диоды $VD1...VD15$. Например, при подключении источника $V1$ к входному контакту «А5» открываются диоды $VD3$ и $VD11$. При этом на выходных шинах «Разряд 0» и «Разряд 2» шифратора появляется напряжение, равное напряжению источника $V1$ за вычетом прямого падения напряжения на диодах $U_{д.пр}$. На шинах «Разряд 1» и «Разряд 3» сохраняется нулевой потенциал. Сформированный двоичный код 101 соответствует числу 5.

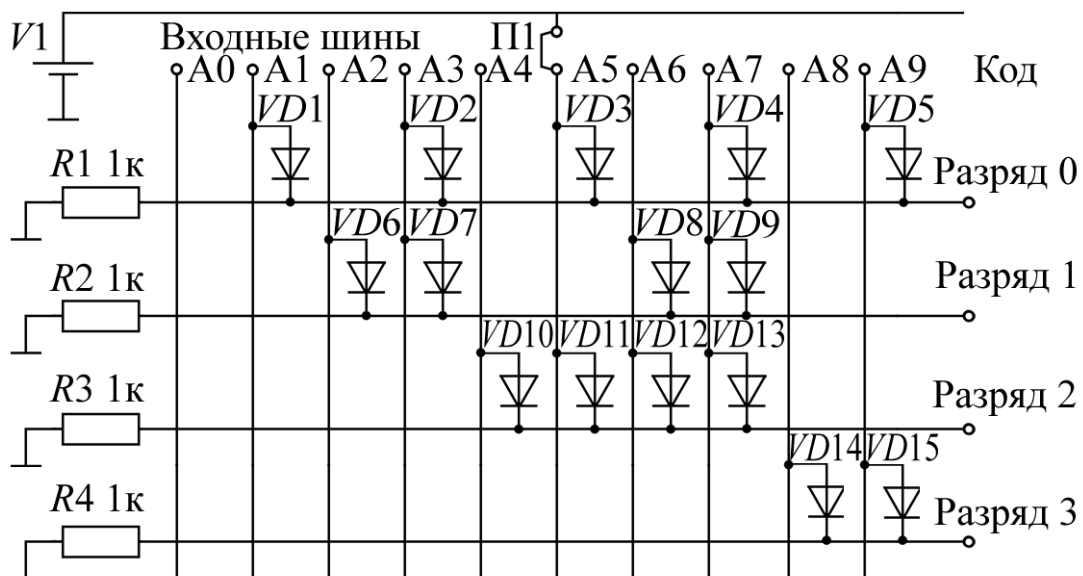


Рисунок 3.5

Операцию кодирования можно реализовать на логических элементах, выполняющих функцию И-НЕ (Рисунок 3.6).

В этой схеме управляющими являются сигналы низкого уровня. В исходном состоянии, пока все входы логических элементов $DD1$, $DD2$,

$DD3$, $DD4$ находятся под высоким потенциалом, на выходе этих элементов будет состояние логического нуля. Если присоединить одну из входных шин перемычкой П1 к «земле» (например, вход «5»), то на выходе логических элементов $DD1$ и $DD3$ (с учетом инверсии) появится логический уровень 1. Выходы $DD2$ и $DD4$ будут находиться в состоянии 0, так как все входы этих логических элементов по-прежнему остаются под высоким потенциалом. В результате на выходе схемы появится код 0101, что соответствует числу 5 в двоичном исчислении.

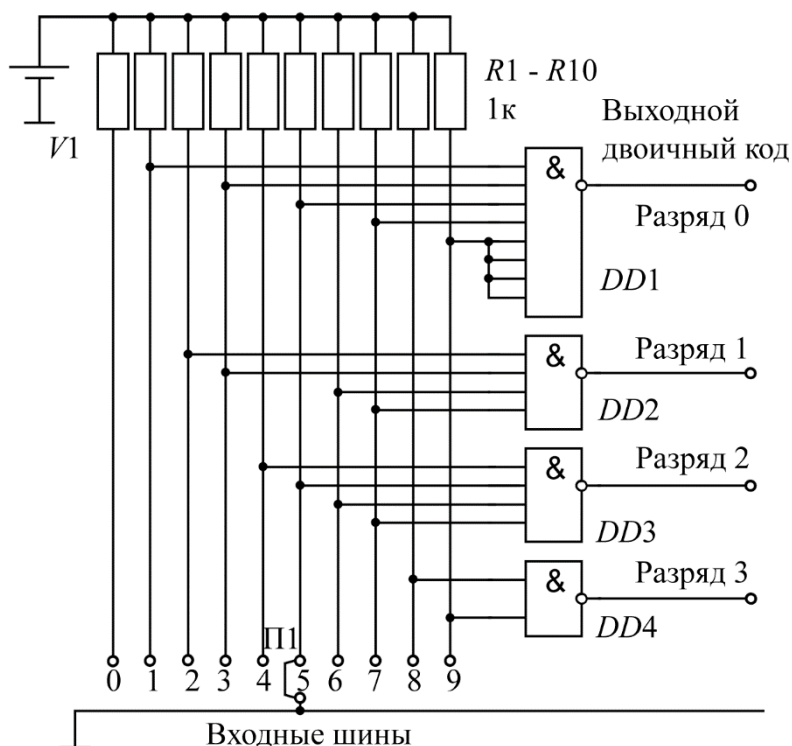


Рисунок 3.6

Выпускаются в виде функционально законченных узлов интегральные микросхемы, так называемых, приоритетных шифраторов. В них реализована функция преобразователя позиционного кода в двоичный с защитой от одновременного нажатия на две и более клавиши.

3.3 Мультиплексоры

Мультиплексором называется комбинационная схема, предназначенная для коммутации одного из 2^n входных сигналов на один выход. В состав мультиплексора входит n-разрядный адресный селектор, на входы которого подают адресный код, управляющий коммутацией.

Принцип работы мультиплексоров описывается таблицей истинности.

Сигнальные входы				Управляющие входы		Выход
«0»	«1»	«2»	«3»	A1	A0	Y
a	b	c	d	0	0	a
a	b	c	d	0	1	b
a	b	c	d	1	0	c
a	b	c	d	1	1	d

Работа мультиплексора формата 4x1 описывается логической функцией:

$$Y = \overline{A0} \cdot \overline{A1} \cdot a + \overline{A0} \cdot A1 \cdot b + A0 \cdot \overline{A1} \cdot c + A0 \cdot A1 \cdot d$$

где A0 и A1 – управляющие (адресные) сигналы,
a, b, c, d – информационные сигналы, подлежащие коммутации.

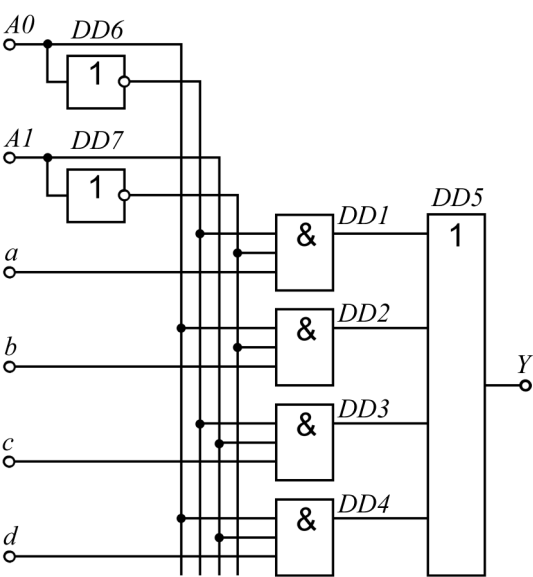


Рисунок 3.7

Логическая схема мультиплексора формата 4x1

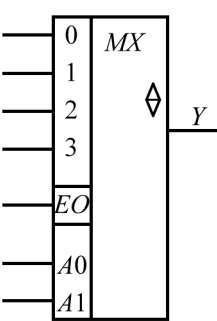


Рисунок 3.8

УГО мультиплексора формата 4x1 со входом разрешения и тремя состояниями на выходе

3.4 Дешифраторы

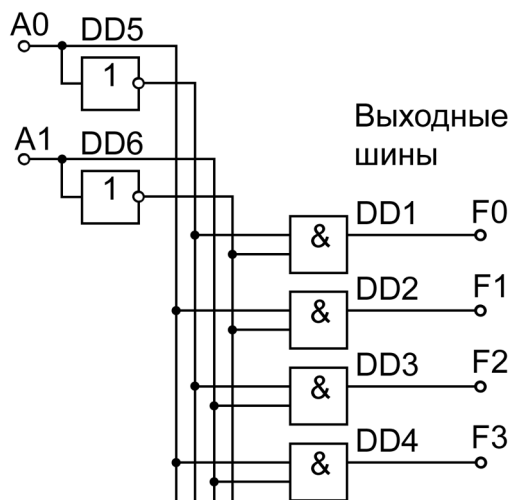
Полным дешифратором называется комбинационная схема, преобразующая входной двоичный код в унитарный и имеющая n адресных входов и 2^n информационных выходов.

Формат дешифратора обозначается как $n \times m$, где n – количество входных сигналов, а m – выходных. Работа устройства может быть представлена таблицей истинности.

Таблица истинности полного дешифратора формата 2×4

Входной двоичный код		Выходной позиционный код			
Разряд A1	Разряд A0	F0	F1	F2	F3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Логическая схема полного дешифратора формата 2×4 приведена на рисунке 3.9.



Работа дешифратора описывается логическими функциями:

$$F0 = \overline{A0} \cdot \overline{A1}$$

$$F1 = A0 \cdot \overline{A1}$$

$$F2 = \overline{A0} \cdot A1$$

$$F3 = A0 \cdot A1$$

Рис. 3.9. Логическая схема полного дешифратора формата 2×4

Схема содержит логические элементы (ЛЭ) DD1...DD4, формирующие сигналы на выходных шинах и инверторы DD5 и DD6, формирующие управляющие сигналы. Сигнал логической единицы появляется на одном из выходов F0...F3 в соответствии с кодом на входах A0 и A1.

Обычно при конструировании на логических элементах первоначально задаются логическими функциями для всех выходных переменных, на основе которых составляют принципиальную электрическую схему

устройства.

Быстродействие схем, выполненных только на комбинационных ЛЭ, определяется по задержке самой длинной логической цепочки. Оценку задержки каждого логического элемента производят по его характеристикам.

На практике дешифраторы часто оснащаются инверсными выходами и входом разрешения работы, при подаче на который сигнала логической единицы все выходные шины становятся не активны и устанавливаются в единицу.

Кроме того, рассмотренный дешифра-тор может выполнять функцию демульти-плексора. При этом вход разрешения используется как информационный (рис.3.10) .

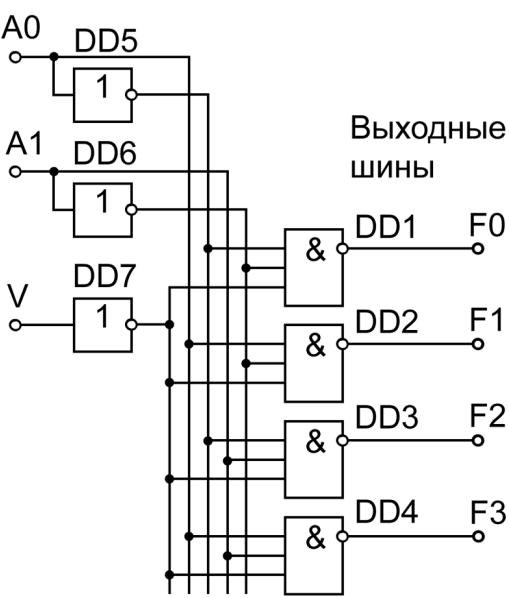


Рисунок 3.10
Схема полного дешифратора с входом разрешения

адрес				V	A1	A2	A4	0	1	2	3	4	5	6	7
	A1	DC	0	0	0	0	0	0	1	1	1	1	1	1	1
	A2		1	0	1	0	0	1	0	1	1	1	1	1	1
	A4		2	0	0	1	0	1	1	0	1	1	1	1	1
			3	0	1	1	0	1	1	1	0	1	1	1	1
			4	0	0	0	1	1	1	1	1	0	1	1	1
разре-			5	0	1	0	1	1	1	1	1	1	0	1	1
	V		6	0	0	1	1	1	1	1	1	1	1	0	1
шение			7	0	1	1	1	1	1	1	1	1	1	1	0
				1	X	X	X	1	1	1	1	1	1	1	1

Рисунок 3.11

Условное графическое обозначение и таблица истинности полного дешифратора (3×8) с инверсными выходами и входом разрешения работы

3.5 Сумматоры

Сумматором называется комбинационная схема выполняющая операцию арифметического сложения двоичных чисел.

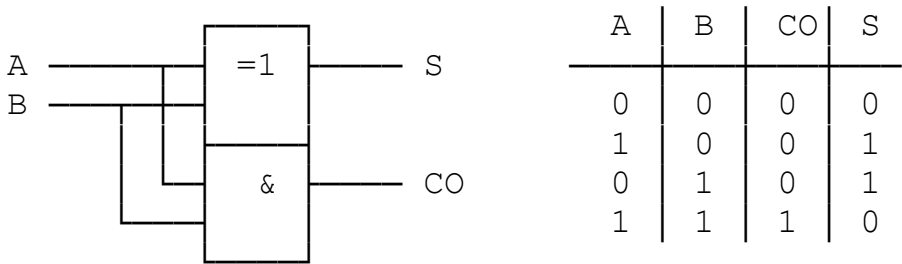


Рисунок 3.11

Полусумматор - выполняет сложение двух одноразрядных чисел A и B. На выходе S формируется сумма по модулю 2, на выходе C - перенос.



Рисунок 3.12

Полный одноразрядный сумматор выполняет операцию арифметического сложения двух одноразрядных чисел A_i и B_i с учетом переноса из младшего разряда CI . Он имеет три входа и два выхода для сигнала суммы S_i и сигнала переноса CO .

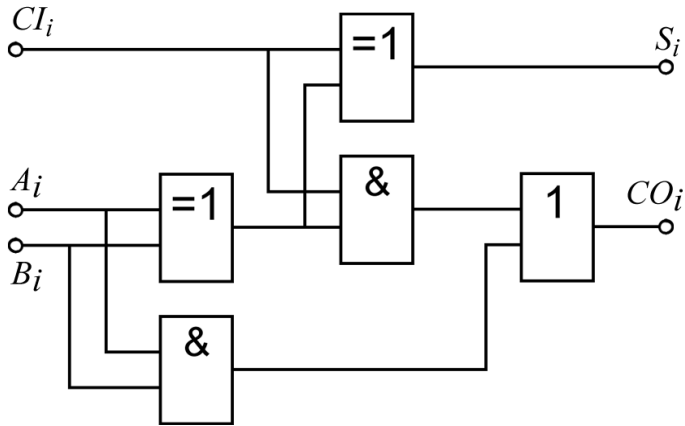


Рисунок 3.13

Логическая схема полного одноразрядного сумматора

Для выполнения сложения многоразрядных чисел одnorазрядные сумматоры включают каскадом. При этом выход переноса младшего разряда является входным переносом для следующего по старшинству разряда. Такое включение образует сумматор с последовательным переносом. Быстродействие такого сумматора зависит от числа разрядов.

Выпускаются схемы многоразрядных сумматоров (4 р). В них организована внутренняя коммутация сигнала переноса и на выходе формируется сумма с учетом переноса из младшего разряда и сигнал переноса из старшего разряда. Они обладают более высоким быстродействием благодаря тому, что имеют в своей структуре схему ускоренного формирования переноса (СУП) во все разряды одновременно. Возможно каскадирование сумматоров для обработки чисел нужной разрядности.

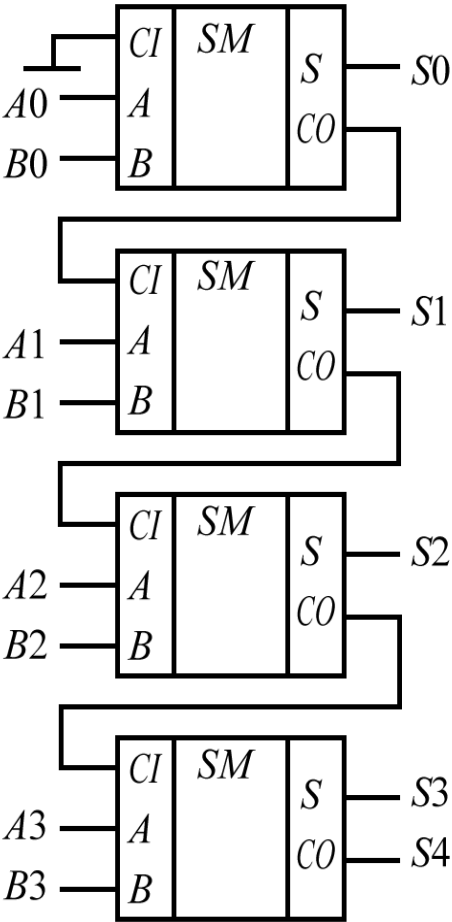


Рисунок 3.14 Каскадное включение сумматоров

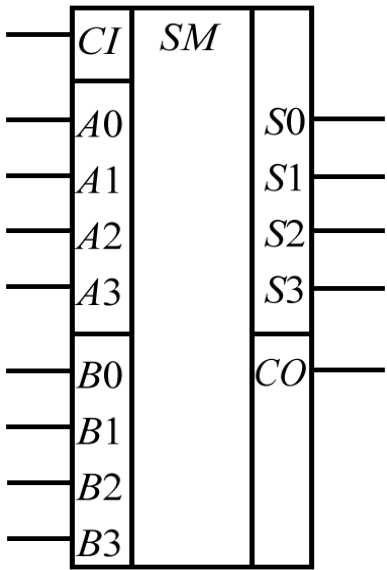


Рисунок 3.15 УГО полного четырёхразрядного сумматора

3.6 Арифметическо-логические устройства (АЛУ)

Предназначены для выполнения арифметических и логических операций над многоразрядными данными под действием управляющих сигналов.

				Функции, выполняемые ИС К555ИПЗ			
АЛУ				Управляющие сигналы S_3, S_2, S_1, S_0	Логические операции $M=1$	Арифметические операции $M=0$	
S_0		F_0		0 0 0 0	$\overline{A}$	A	
S_1		F_1		0 0 0 1	$\overline{A \vee B}$	$A \vee B$	
S_2		F_2		0 0 1 0	$\overline{A \wedge B}$	$A \vee \overline{B}$	
S_3		F_3		0 0 1 1	0	-1	
A_0		$F_{A=B}$		0 1 0 0	$A \wedge B$	$A + A \wedge \overline{B}$	
A_1				0 1 0 1	$\overline{B}$	$(A \vee B) + A \wedge \overline{B}$	
A_2		C_{n+4}		0 1 1 0	$A + B$	$A - B - 1$	
A_3				0 1 1 1	$A \wedge \overline{B}$	$A \wedge \overline{B} - 1$	
B_0		G		1 0 0 0	$\overline{A \vee B}$	$A + A \wedge B$	
B_1				1 0 0 1	$\overline{A + B}$	$\overline{A + B}$	
B_2		P		1 0 1 0	B	$(A \vee \overline{B}) + A \wedge B$	
B_3				1 0 1 1	$A \wedge B$	$A \wedge B - 1$	
C_n		$U_{н.п}$		1 1 0 0	1	$A + A$	
V		Общ		1 1 0 1	$A \vee \overline{B}$	$(A \vee B) + A$	
				1 1 1 0	$A \vee B$	$(A \vee \overline{B}) + A$	
				1 1 1 1	A	$A - 1$	

Рисунок 3.15

Вывод K является выходом встроенного компаратора, формирующего сигнал $K=1$ при $A=B$.

Примечание. 1. При условии $M=0$ ИС может одновременно выполнять как логические, так и арифметические операции. Для исключения путаницы в табл. 18.5 для обозначения операции логического сложения и умножения соответственно использованы знаки « $\vee$ » и « $\wedge$ ».

2. Знаками « $+$ » и « $-$ » обозначены операции арифметического сложения и вычитания. Поэтому выражение $A \vee B + A \wedge \overline{B}$ следует понимать как арифметическую сумму двух кодов: первый из которых получен в результате логического сложения, а второй — логического умножения исходных кодов A и B .

К564ИПЗ (МС14581А) — четырехразрядный АЛУ со СУП.

Имеет:

- восемь входов для ввода двух чисел A и B ,
- четыре выхода F_3, F_2, F_1, F_0 ,
- четыре управляющих входа,

$S_3 S_2 S_1 S_0$ код задания одной из 16 операций,

- вход M признака операций (1 — лог, 0 — ариф),
- вход переноса C_n ,
- выход компаратора $F_{A=B}$
- и выходы C_{n+4}, G, P для организации переноса при наращивании разрядности АЛУ.

К564ИП4 (СУП) позволяет объединить 4 К564ИПЗ (АЛУ) и получить 16 разрядное АЛУ.

4 ОСНОВНЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ УСТРОЙСТВА

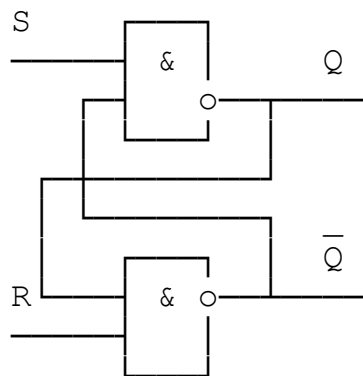
Последовательностным называют устройство, состояние которого определяется комбинацией входных сигналов и состоянием устройства в предыдущие моменты времени.

Таким образом, можно говорить о том, что последовательностные устройства обладают памятью состояния.

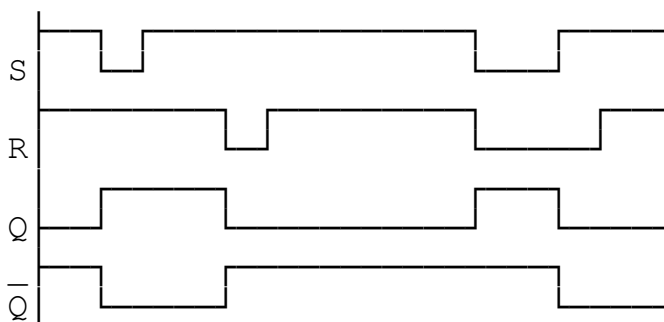
4.1 Триггеры

Триггером называется логическое устройство, имеющее два устойчивых состояния, в одно из которых оно может быть установлено под воздействием внешних управляющих сигналов и оставаться в этом состоянии после их окончания неограниченное время до следующего управляющего воздействия.

Таким образом, триггер является элементарной ячейкой памяти, способной хранить один бит двоичной информации.



- простейший R-S - триггер



S	R	Q	$\bar{Q}$	
0	0	1	1	- запрещенное
0	1	1	0	- установка «1»
1	0	0	1	- установка «0»
1	1	Q	$\bar{Q}$	- хранение

Раздельные входы для установки 0 и 1.

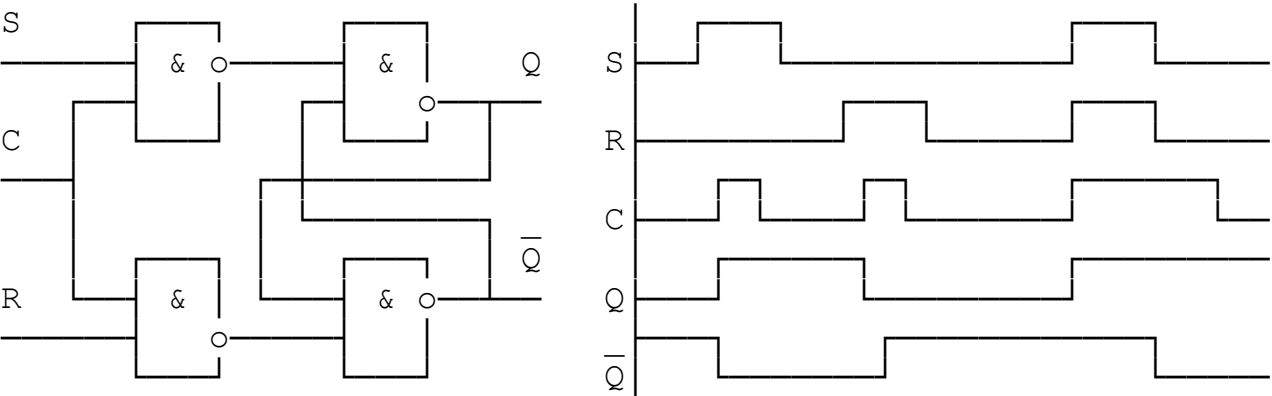
Недостаток - наличие "запрещенного"

состояния $R = 0, S = 0$.



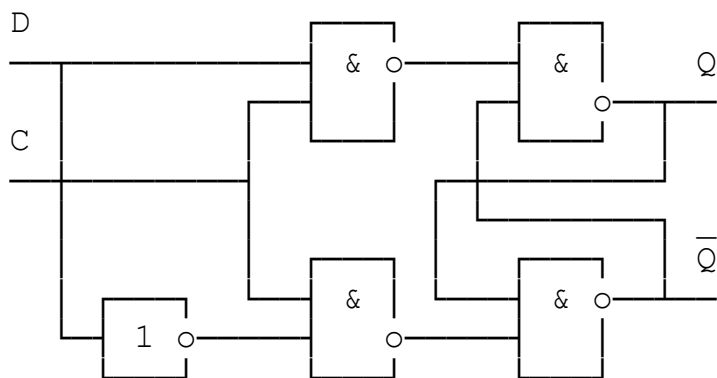
При необходимости согласования процессов установки 0 и 1 с другими процессами в системе используют тактируемые R-S триггеры, содержащие специальный вход для подачи синхроимпульса.

"Запрещенное" состояние $S = R = C = 1$

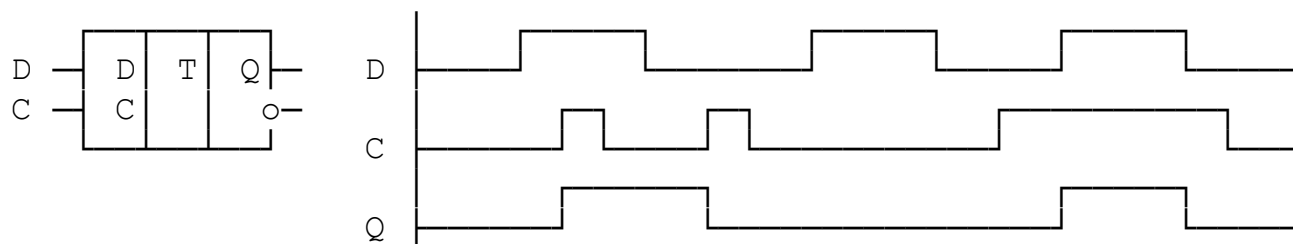


C	S	R	Q	$\overline{Q}$	В таблице "X" - безразличное состояние, не влияет на работу устройства
	1	0	1	0	- установка «1» при активном синхроимпульсе
	0	1	0	1	- установка «0» при активном синхроимпульсе
X	1	1	Q	$\overline{Q}$	- хранение
0	X	X	Q	$\overline{Q}$	- хранение
1	1	1	1	1	- запрещенное

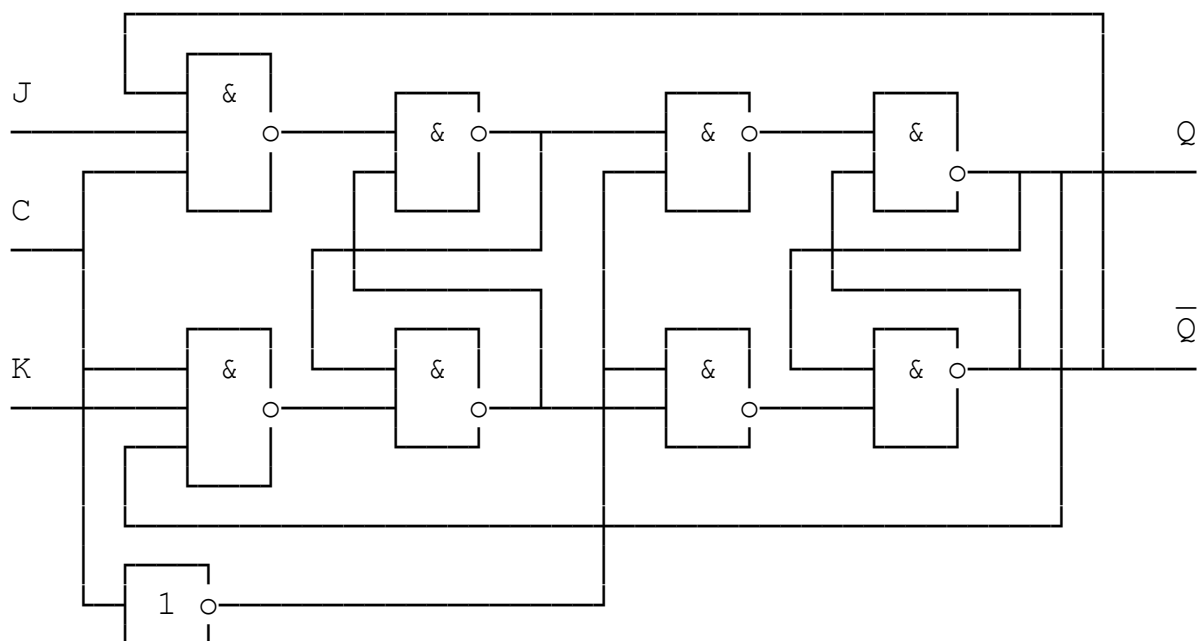
Для сохранения двоичных данных используют триггеры, построенные на основе тактируемых R-S триггеров и имеющие специальный информационный вход D (DATA). Запись информации в таких триггерах происходит только при окончании синхроимпульса ("Защелкивание") информации. Такие устройства получили название статических асинхронных D-триггеров ("Триггер-защелка"). Единичный уровень на входе синхронизации переводит триггер в режим повторителя при котором сигнал с входа "D" передаётся на выход "Q".



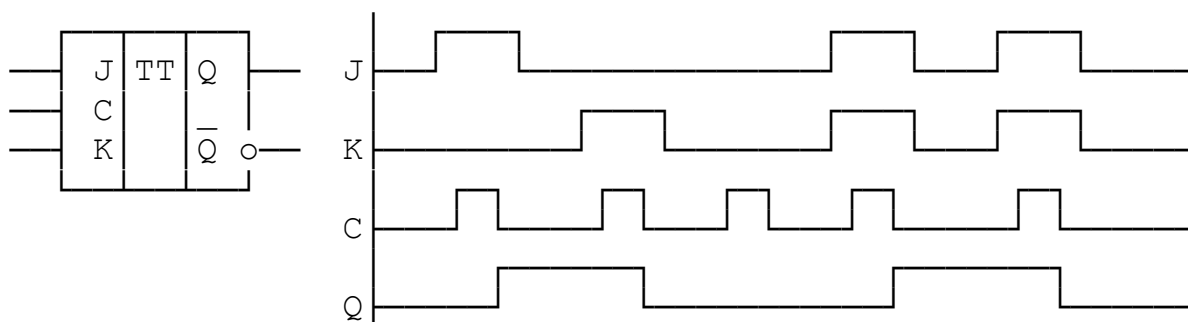
C	D	Q	$\bar{Q}$	В таблице "X" - безразличное состояние, не влияет на работу устройства
	1	1	0	- запись «1»
	0	0	1	- запись «0»
0	X	Q	$\bar{Q}$	- хранение
1	D	D	$\bar{D}$	- повторитель



Для исключения "запрещенного" состояния на основе двухступенчатых R-S триггеров с обратной связью разработан J-K триггер. Основными отличиями его от R-S триггера является отсутствие "запрещенного" состояния и запоминание информации по фронту синхроимпульса, а установка соответствующего состояния на выходе только по окончании синхроимпульса. Особенностью работы такого триггера является наличие "счетного" режима, когда триггер под действием синхроимпульса переключается в состояние противоположное предыдущему.

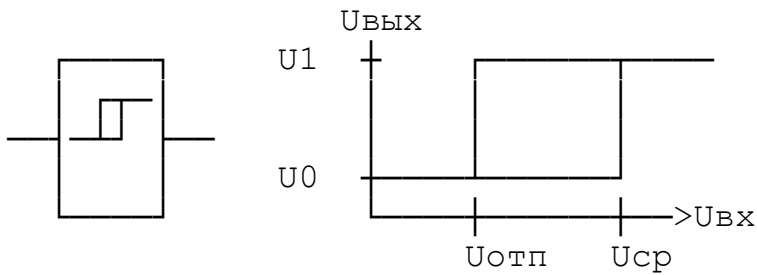


С	J	K	Q	$\bar{Q}$	В таблице "X" - безразличное состояние, не влияет на работу устройства
	1	0	1	0	- установка «1» при активном синхроимпульсе
	0	1	0	1	- установка «0» при активном синхроимпульсе
X	1	1	Q	$\bar{Q}$	- хранение
0	X	X	Q	$\bar{Q}$	- хранение
	1	1	$\bar{Q}$	Q	- счётный режим

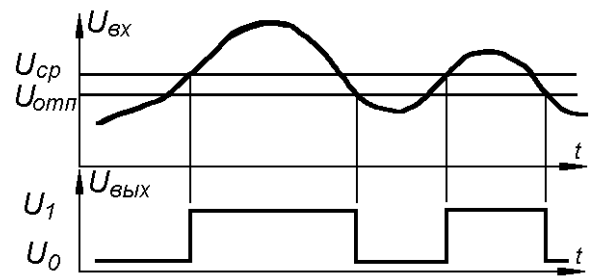


Существенно отличающимися от рассмотренных свойствами обладает триггер Шмитта. У него один вход и один выход. Передаточная характеристика триггера Шмитта имеет гистерезис, что обуславливает наличие двух пороговых уровней срабатывания и отпускания при изменении входного сигнала: при достижении напряжения на входе $U_{ср}$ триггер скачком переходит в единичное состояние, при уменьшении

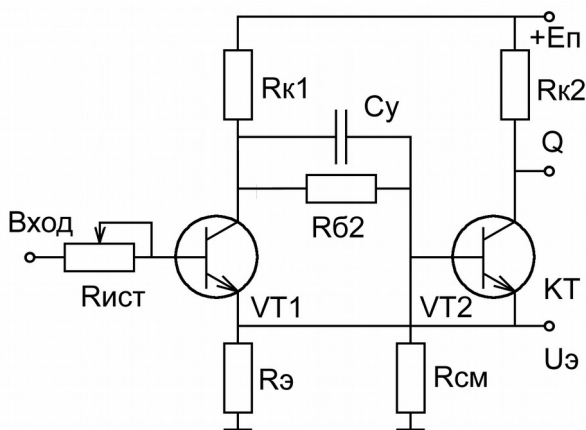
входного напряжения до $U_{отп}$ триггер возвращается в нулевое состояние. Такой принцип определяет его область применения в качестве формирователя прямоугольных импульсов из сигнала произвольной формы, одновибраторов, реле времени и др. Выполнять функции элемента памяти триггер Шмитта не может, поэтому применять его в регистрах и счетчиках нельзя.



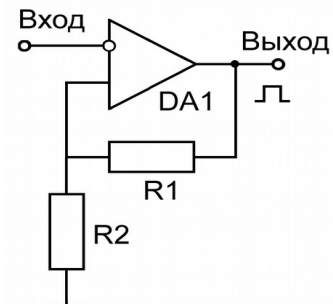
Условное графическое изображение



временные диаграммы работы



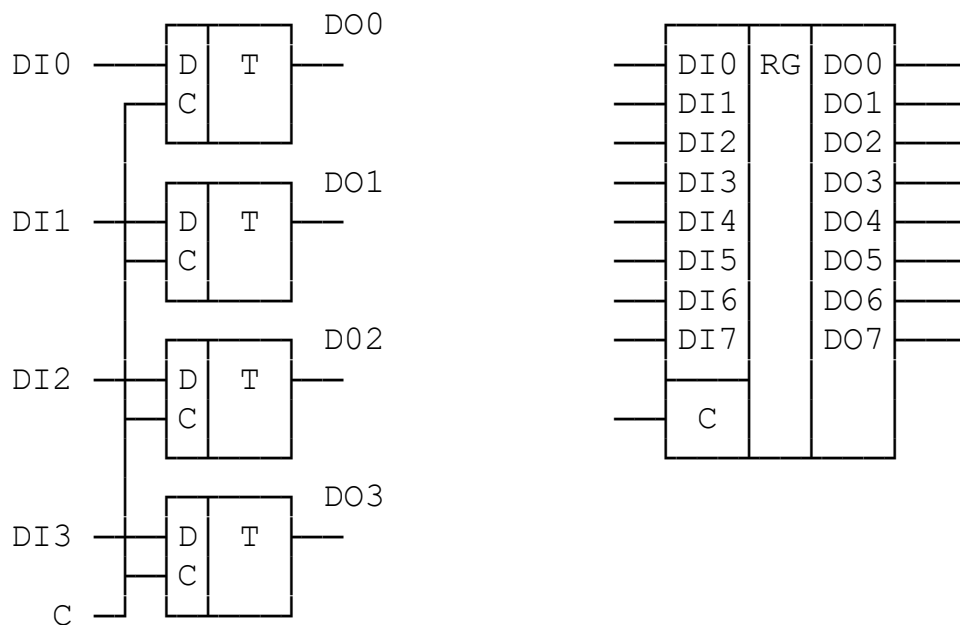
Реализация триггера Шмитта на биполярных транзисторах



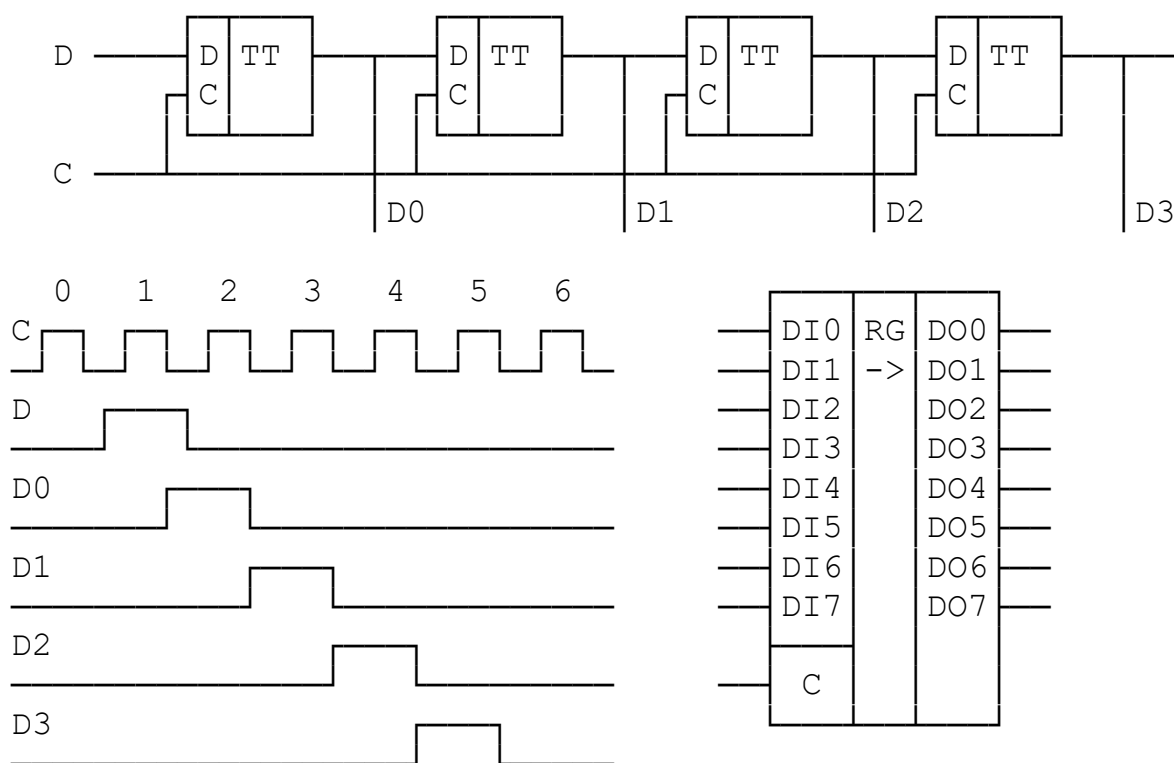
Реализация триггера Шмитта на операционном усилителе

4.2 Регистры

Регистром называют логическое устройство, предназначенное для записи и хранения двоичной информации. Помимо хранения некоторые виды регистров могут преобразовывать информацию, например, из последовательного вида в параллельный, сдвигать записанную информацию в одну или обе стороны. В соответствии с назначением различают регистры хранения и регистры сдвига.



Последовательная цепочка из n D-триггеров образует регистр с последовательным сдвигом информации.

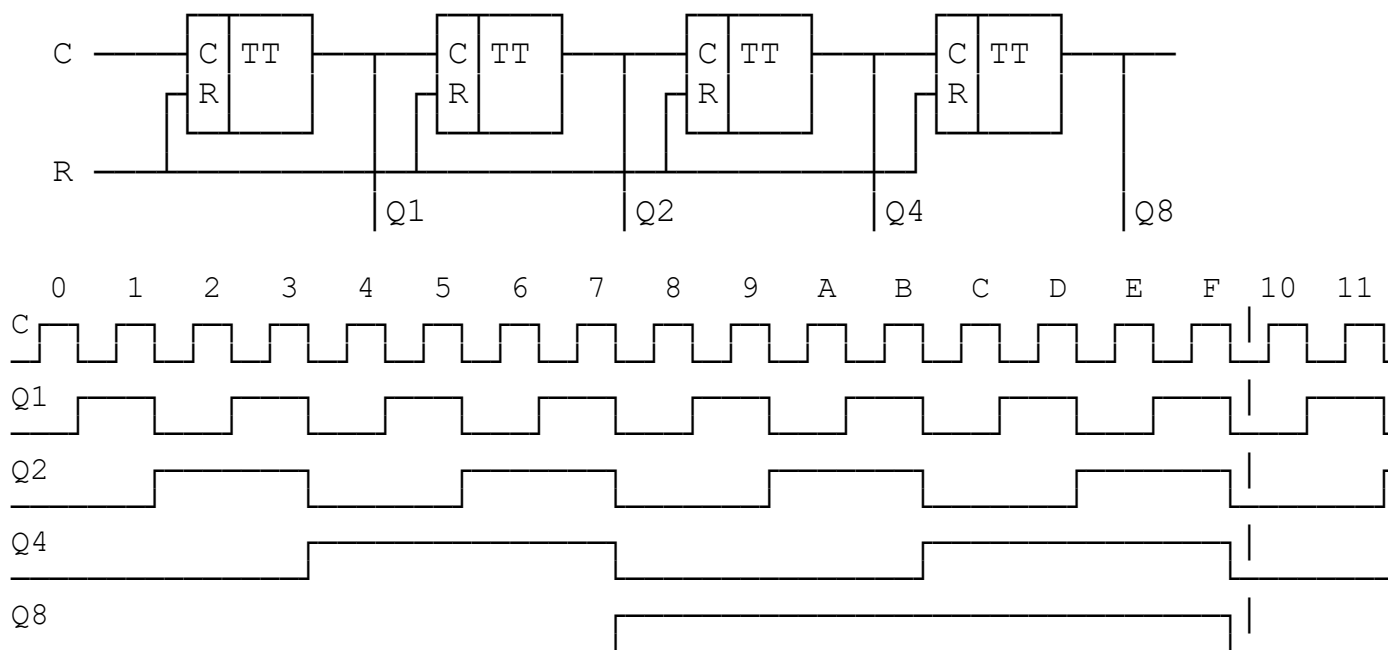


Выпускаются универсальные регистры, способные записывать и представлять информацию как в последовательном, так и в параллельном виде.

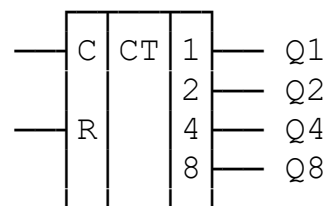
4.3 Счетчики

Цифровым счетчиком называют логическое устройство, которое осуществляет счет поступающих на его вход импульсов, формирует результат счета в заданном коде и, при необходимости, хранит его. Для построения счетчиков требуются триггеры двухступенчатой структуры.

Последовательная цепочка из n счетных триггеров образуют счетчик с последовательным переносом с коэффициентом счета $K_{сч}=2^n$.



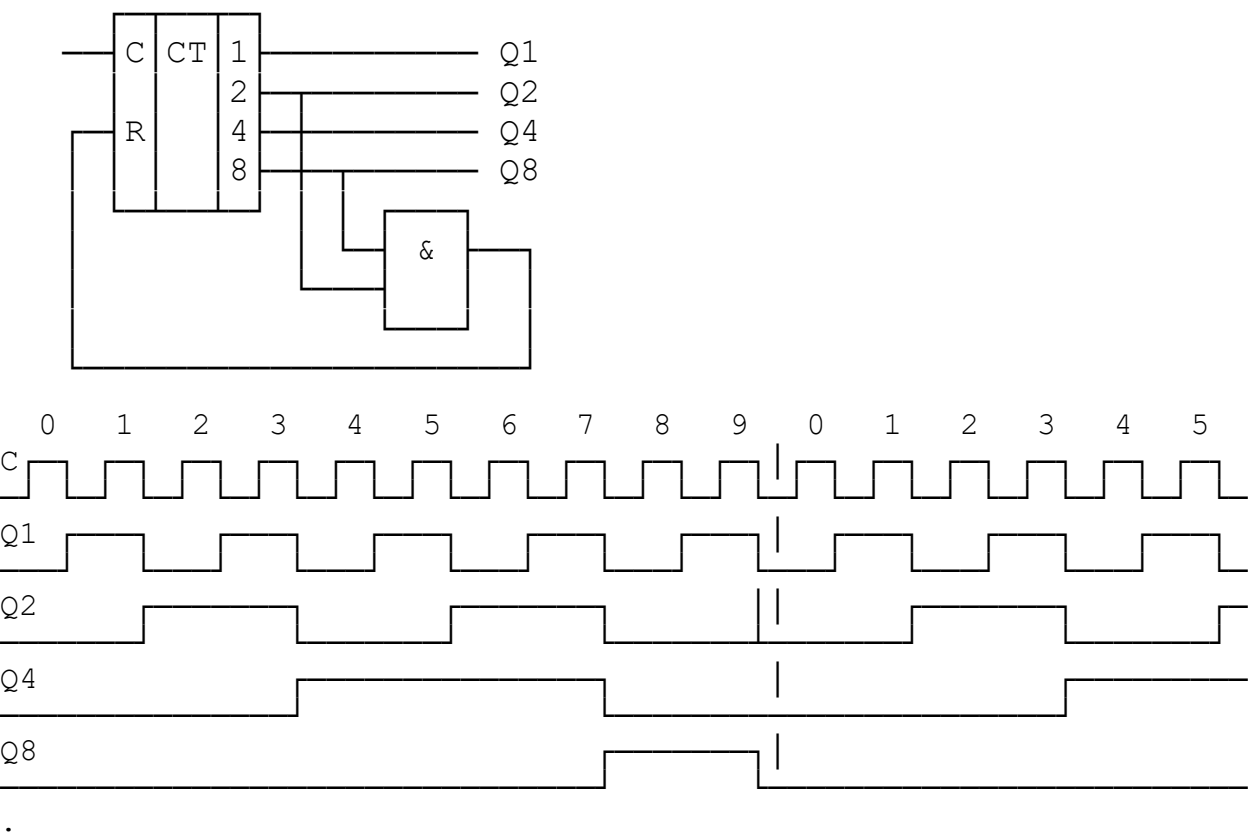
Двоичное число, сформированное на выходах Q1, Q2, Q4, Q8, соответствует числу импульсов поданных на вход C на данный момент. Увеличение этого числа на 1 происходит после окончания



очередного импульса, т.е. по заднему фронту. После окончания 2^n импульса все триггеры цепочки оказываются в нулевом состоянии. Цикл повторяется. В связи с тем, что каждый триггер вносит некоторую задержку, формирование кода на выходе происходит не мгновенно, причем, задержка увеличивается от младшего разряда к старшему. Такие счетчики называются асинхронными с последовательным переносом. Выпускаются синхронные счетчики с последовательным или параллельным переносом, у которых код во всех разрядах формируется синхронно с передним или задним фронтом входного сигнала. Такие счетчики

обеспечивают наивысшее быстродействие.

С помощью дополнительных логических элементов можно организовать цепь обратной связи и изменить коэффициент счета в пределах $2^{(n-1)} < K_{сч} < 2^n$. Для этого сигналы с выходов определенных триггеров подключаются к входам логической схемы, которая при достижении нужного числа импульсов формирует управляющий сигнал и подает его на вход R принудительного сброса триггеров в нулевое состояние. Например, счетчик до 10:



5 МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

МИКРОПРОЦЕССОРНОЙ СИСТЕМОЙ называют специализированную информационную или управляющую систему, построенную на основе МП средств.

МИКРО – ЭВМ или МИКРОКОМПЬЮТЕРОМ, называют устройство обработки данных, содержащее один или несколько МП, БИС ПЗУ и ОЗУ, БИС управления вводом – выводом информации и некоторые другие схемы.

Микро – ЭВМ в такой конфигурации часто применяют в качестве встраиваемых в различные станки, машины, технологические процессы управляющих устройств.

Микро – ЭВМ широкого назначения, используемые для выполнения вычислительных работ и управления сложными тех. процессами, оснащаются необходимыми периферийными устройствами: дисплеями, печатающими устройствами, НГМД, НЖМД, АЦП, ЦАП и др.

Микрокомпьютер с небольшими вычислительными ресурсами и упрощенной системой команд, ориентированный не на производство вычислений, а на выполнение процедур логического управления различным оборудованием называют программируемым микроконтроллером или просто МИКРОКОНТРОЛЛЕРОМ.

Логическая организация (архитектура) МП ориентирована на достижение универсальности применения, высокой производительности и технологичности.

Универсальность МП определяется возможностью их разнообразного применения и обеспечивается программным управлением МП, позволяющим производить программную настройку МП на выполнение определенных функций, магистрально – модульный принцип построения, а также специальными архитектурно – логическими средствами:

сверхоперативной регистровой памятью, многоуровневой системой прерываний, прямым доступом к памяти, программно - настраиваемыми схемами управления ввода - вывода и др. Относительно высокая производительность МП достигается использованием для их построения быстродействующих электронных схем и специальных архитектурных решений, таких как стековая память, разнообразные способы адресации, гибкая система команд.

Типичная структура МП устройства (системы)

На рисунке 5.1 представлена типичная структура МП системы, предназначенной для обработки данных или управления некоторым процессом. Примерно такую же структуру имеют микро-ЭВМ широкого назначения. Центральное место в системе занимает микропроцессор. Он выполняет арифметические и логические операции над данными, осуществляет управление процессом обработки информации

Для МП-систем характерна трехшинная структура, содержащая шину адреса ША, шину данных ШД и шину управления ШУ. Представленная схема отражает магистрально-модульный принцип построения системы. Отдельные элементы являются функционально законченными модулями со своими встроенными схемами управления. Межмодульные связи и обмен информацией осуществляется посредством шин, к которым имеют доступ все основные элементы системы. В каждый момент возможен обмен информацией только между двумя модулями системы. Таким образом, обмен информацией производится путем разделения во времени шин между модулями системы. Магистральный принцип построения МП-системы предполагает наличие информационно-логической совместимости модулей, которая реализуется путем использования единых протоколов обмена и

способа синхронизации.

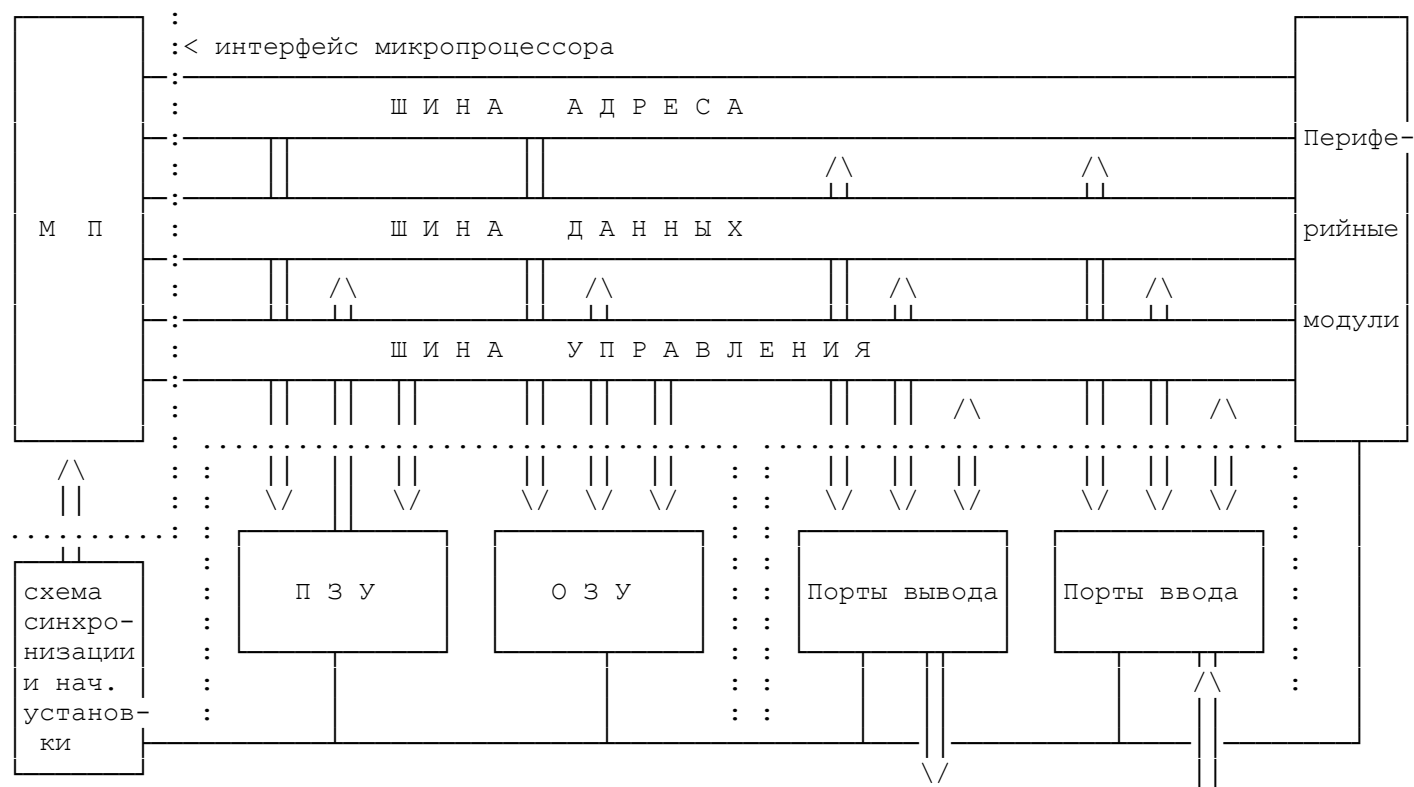


Рисунок 5.1 - Магистрально-модульный принцип построения МП системы

6 ПОНЯТИЕ И АРХИТЕКТУРА МИКРОПРОЦЕССОРА

Микропроцессор (МП) – это микросхема или совокупность небольшого числа микросхем, выполняющая над данными арифметические и логические операции и осуществляющая управление вычислительным процессом.

Любой микропроцессор можно представить как устройство, состоящее из четырех логически законченных узлов:

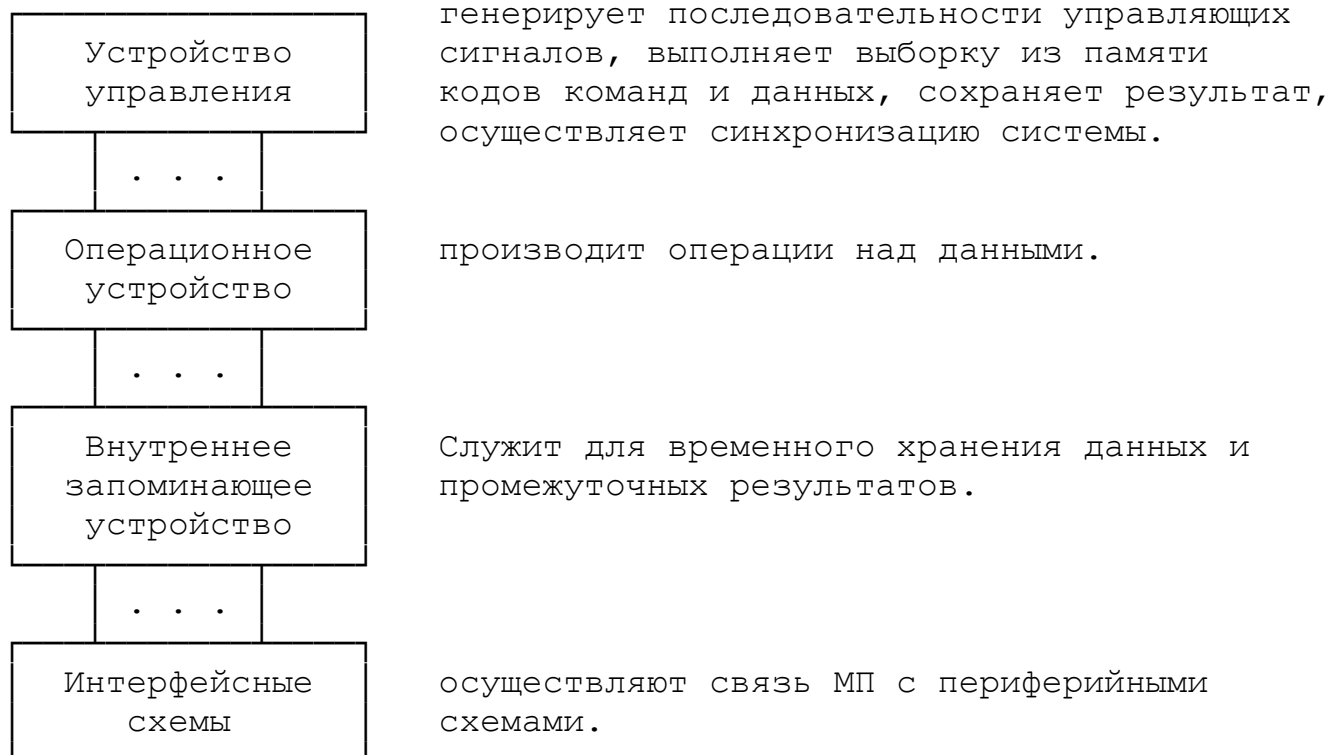


Рисунок 6.1

Связь между узлами осуществляется при помощи шин. Шины различают по характеру передаваемой информации:

- шина данных;
- шина адреса;
- шина управления.

КЛАССИФИКАЦИЯ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МП

Классификация микропроцессоров осуществляется по следующим показателям :

1. Разрядность слова данных – 4, 8, 16, 32, 64, 128;
2. Архитектура
 - а) Фон-Неймана (Принстонская)
 - б) Гарвардская
 - в) Модифицированная Гарвардская
3. Система и набор команд
 - а) Полный набор (CISC)
 - б) Ограниченный (RISC)
4. Тип управляющего устройства
 - а) схемное управление – фиксированная система команд
 - б) микропрограммное управление – изменяемая с/к;
5. Тип микрoeлектронной технологии,
Отсюда:
 - а) Быстродействие;
 - б) Энергопотребление;
4. Количество кристаллов и ядер
 - а) Однокристалльные;
 - б) Многокристалльные;
 - в) одноядерные;
 - в) многоядерные:
 - гомогенные,
 - гетерогенные,
6. Емкость адресуемой памяти, отсюда – разрядность шины адреса;
8. По назначению: центральные процессоры (CPU), однокристалльные ЭВМ (микроконтроллеры) (MCU), арифметические сопроцессоры (NPU), графические процессоры (GPU), процессоры обработки сигналов (DSP); и др. показатели.

Архитектура Фон-Неймана

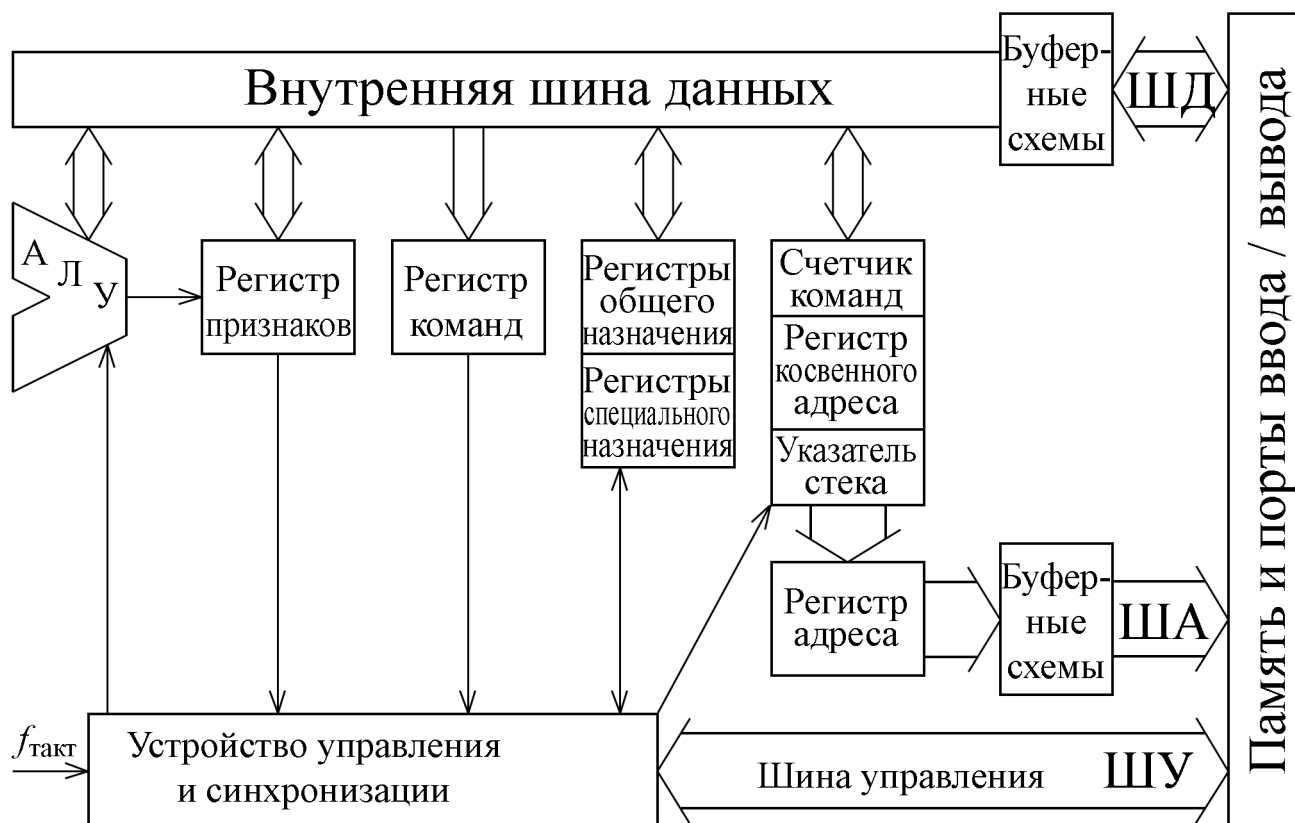


Рисунок 6.2 – Схема МП, построенного по архитектуре Фон-Неймана

Особенностью архитектуры машины Фон-Неймана является наличие единого адресного пространства для хранения как программ, так и данных. При этом для обращения к памяти используются шины адреса и данных, по которым в разные моменты времени передаются либо коды команд, либо данные и только программист определяют назначение различных областей памяти. Такая архитектура позволяет создавать достаточно простые, компактные микропроцессорные системы. Однако, использование общих шин и памяти значительно снижает быстродействие системы. Процессоры, построенные по архитектуре Фон-Неймана, обычно используются для построения универсальных систем, оперирующих большими объемам данных, со средним быстродействием. Для МП систем, построенных на основе архитектуры Фон-Неймана характерен высокий коэффициент использования памяти.

Гарвардская архитектура

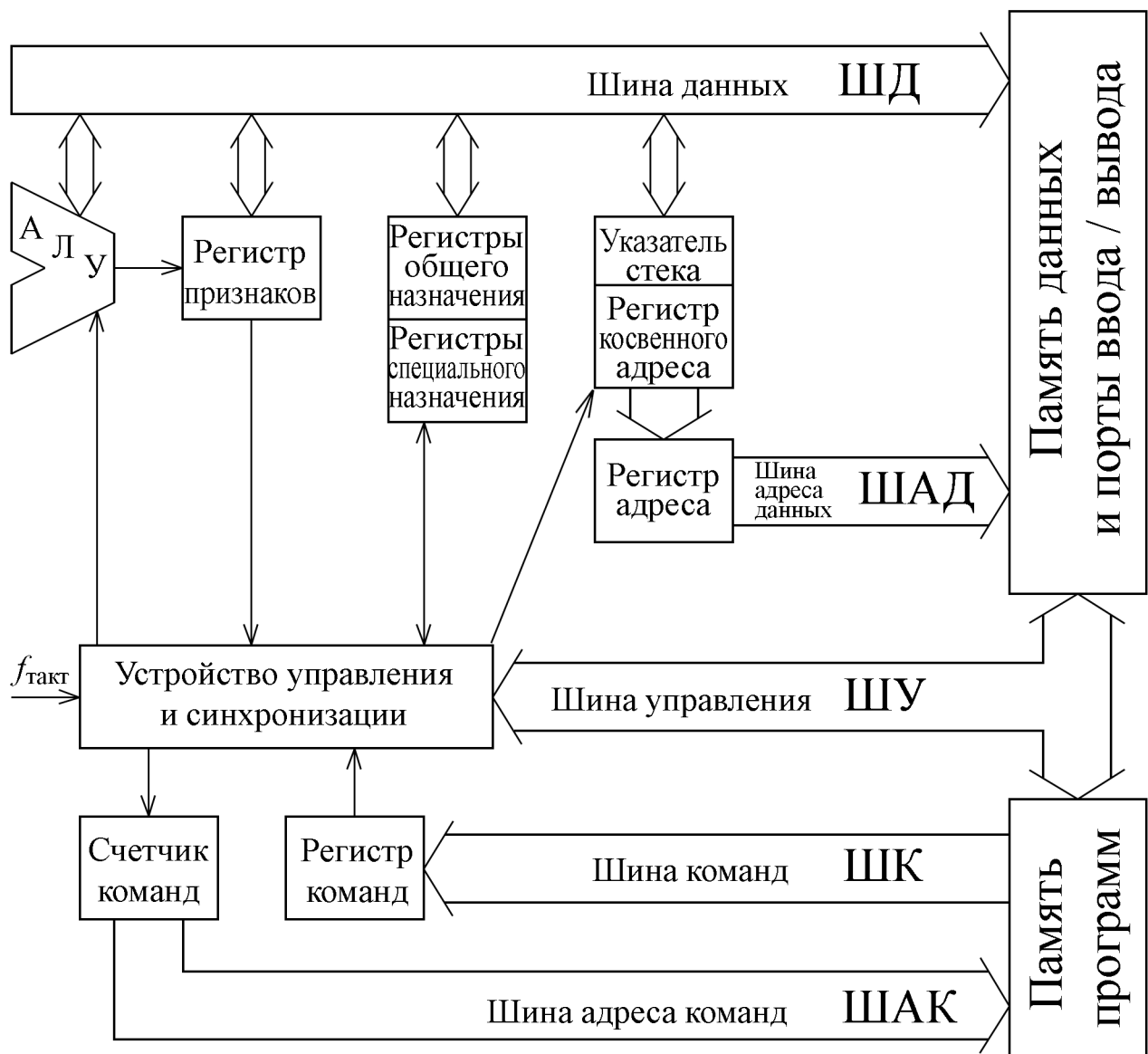


Рисунок 6.3 – Схема МП, построенного по Гарвардской архитектуре

Особенностью МП систем, построенных по Гарвардской архитектуре, является наличие отдельных групп шин и банков памяти для хранения программ и данных. При этом коды команд и данных могут передаваться одновременно, независимо друг от друга. Такая архитектура позволяет создавать системы с высоким быстродействием. Однако, использование отдельных шин значительно усложняет такие системы и делает их достаточно громоздкими. Процессоры, построенные по Гарвардской архитектуре, обычно используются для построения специализированных систем с высоким быстродействием и микроконтроллеров. Для МП систем, построенных на основе Гарвардской архитектуры характерен низкий коэффициент использования памяти. Развитием является Модифицированная Гарвардская архитектура, позволяющая использовать память программ для хранения данных..

7 ПОСТРОЕНИЕ ПРОЦЕССОРА

Существует два принципиально разных подхода к проектированию управляющего устройства МП: использование принципа схемной логики или использование принципа программируемой логики.

В первом случае при проектировании МП определяется такой порядок соединения логических схем, который обеспечивает требуемое функционирование. Устройства построенные на принципе схемной логики, способны обеспечивать наивысшее быстродействие при заданной технологии.

Во втором случае требуемое функционирование обеспечивается занесением в память устройства определенной программы или микропрограммы. В зависимости от введенной программы такое универсальное управляющее устройство способно обеспечить требуемое управление операционным устройством при решении самых разных задач. Можно построить операционное устройство с таким набором узлов и такой схемой их соединения, которые обеспечивают решение разнообразных задач. Задача, решаемая таким универсальным операционным устройством, определяется тем, какая микропрограмма хранится в управляющем устройстве. Таким образом, независимо от решаемой задачи может быть использовано одно и то же операционное устройство.

Следует однако иметь ввиду, что наивысшее быстродействие достигается в процессорах, в которых управляющее устройство строится с использованием принципа схемной логики, а операционное устройство выполняется в виде устройства, специализированного для решения конкретной задачи.

7.1 СИНТЕЗ ПРОЦЕССОРА СО СПЕЦИАЛИЗИРОВАННЫМ ОПЕРАЦИОННЫМ УСТРОЙСТВОМ

Рассмотрим методику построения процессора на примере реализации устройства, выполняющего операцию умножения двоичных чисел без знака.

Проиллюстрируем на примере умножения двух двоичных чисел.

Предусмотрим суммирование частичных произведений, начиная с младшего.

1 1 0 113

1 0 1 111

1 1 0 1

1 1 0 113

0 0 0 013

1 1 0 1

1 0 0 0 1 1 1 1143

В таблице приведена схема выполнения операции.

Множимое (R1)	Старшие разряды произведения (R3)	Множитель и младшие разряды произведения (R2)	Действие
1 1 0 1	0 0 0 0	1 0 1 1	Исходное состояние
	+ 1 1 0 1		Суммирование
	0 1 1 0 1		
	0 > 0 1 1 0	> 1 1 0 1	Сдвиг (R3) и (R2)
	+ 1 1 0 1		
	1 0 0 1 1		Суммирование
	0 > 1 0 0 1	> 1 1 1 0	Сдвиг (R3) и (R2)
	0 > 0 1 0 0	> 1 1 1 1	Сдвиг (R3) и (R2)
	+ 1 1 0 1		
	1 0 0 0 1		Суммирование
	0 > 1 0 0 0	> 1 1 1 1	Сдвиг (R3) и (R2)
	п р о и з в е д е н и е		

В исходном состоянии в R1 и R2 – множимое и множитель, R3 – в нулевом состоянии. Анализируется содержимое младшего разряда R2. Так как его значение "1", то 1-е частичное произведение равно множимому и оно прибавляется к содержимому регистра R3, используемому для накопления сумм частичных произведений. Далее производится сдвиг на один разряд вправо содержимого регистров R2 и R3, причем выдвигаемый при сдвиге из R3 младший разряд (не принимающий в дальнейшем участия при суммировании) передается в освобождающийся старший разряд R2. В младшем разряде R2 оказывается 2-й разряд множителя. Процесс повторяется. При его нулевом значении суммирование не производится, только сдвиг. Число циклов равно числу разрядов множителя.

7.2 СИНТЕЗ ОПЕРАЦИОННОГО УСТРОЙСТВА

В соответствии с описанным выше процессом, для выполнения операции умножения необходимо в операционном устройстве иметь: регистры R1,2,3, сумматор (См) и счетчик (Сч) числа повторений.

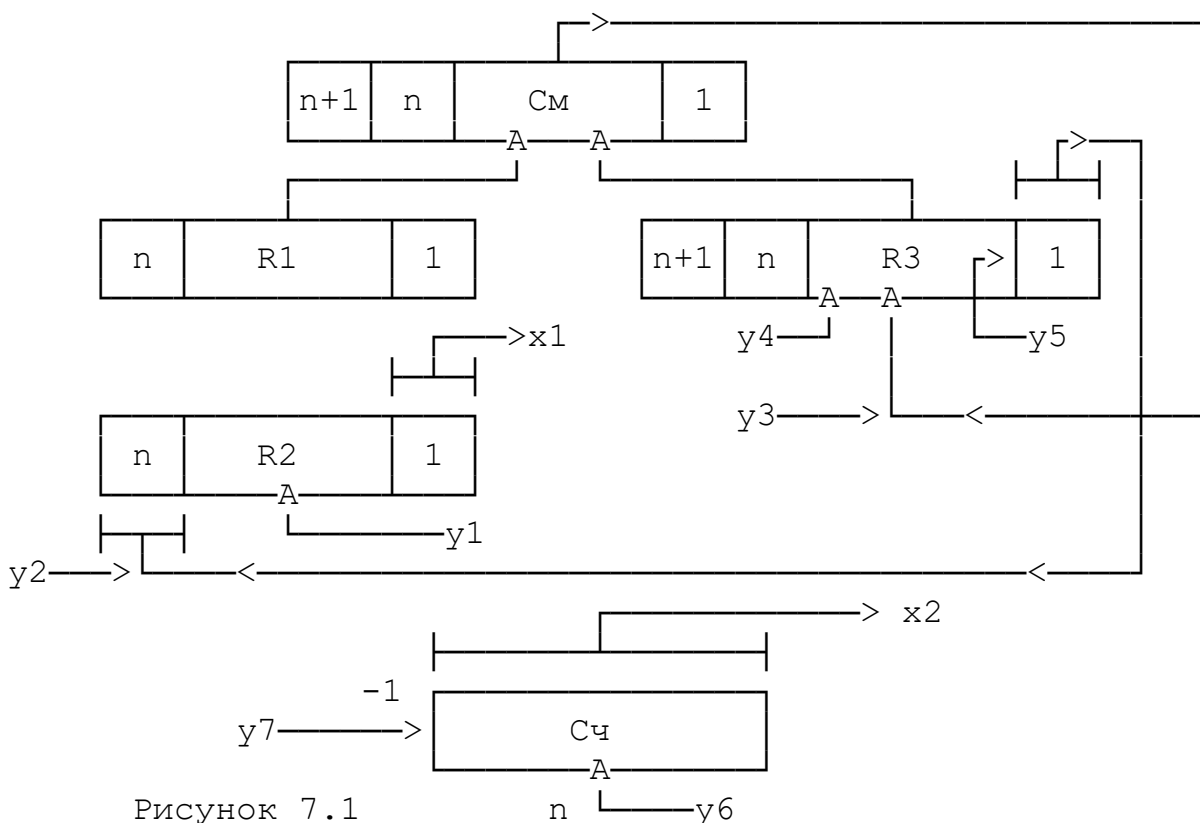


Рисунок 7.1

В R2 предусмотрена микрооперация сдвига содержимого на 1 разряд вправо, выполняемая под действием управляющего сигнала y_1 и микрооперации пересылки в старший разряд этого регистра содержимого младшего разряда R3, выполняемого под действием сигнала y_2 . Сумматор производит суммирование чисел, поступающих из R1 и R3. Для хранения переноса сумматор содержит $n+1$ разряд. Результат под действием y_3 принимается в R3, который, тоже должен иметь $n+1$ разряд. Кроме того для регистра R3 предусмотрены микрооперации установки в 0 и сдвига содержимого на 1 разряд вправо, выполняемые под действием y_4 и y_5 . При наличии y_6 счетчиком принимается установленное число n , под действием y_7 выполняется микрооперация вычитания 1.

В операционном устройстве формируются следующие признаки:

x_1 - содержимое младшего разряда R2;

x_2 - результат проверки на 0 счетчика.

Приведем в условной записи список выполняемых в узлах операционного устройства микроопераций и список формируемых признаков:

y_1 : R2 $\leftarrow$ Сдвиг вправо (R2)

y_2 : R2[n] $\leftarrow$ (R3[1])

y_3 : R3 $\leftarrow$ См

y_4 : R3 $\leftarrow$ (0)

y_5 : R3 $\leftarrow$ Сдвиг вправо (R3)

y_6 : Сч $\leftarrow$ (n)

y_7 : Сч $\leftarrow$ (Сч) - 1

x_1 : (R2[1]) = 1

x_2 : (Сч) = 0

7.3 СИНТЕЗ УПРАВЛЯЮЩЕГО УСТРОЙСТВА НА ОСНОВЕ СХЕМНОЙ ЛОГИКИ

Разобьем на 2 этапа:

1. Построение алгоритма в микрооперациях (схема на рисунке соответствует рассмотренному множительному устройству);

2. Построение алгоритма в микрокомандах.

Для формирования микрокоманд необходимо определить, какие микрооперации могут выполняться одновременно (в одни и те же тактовые периоды).

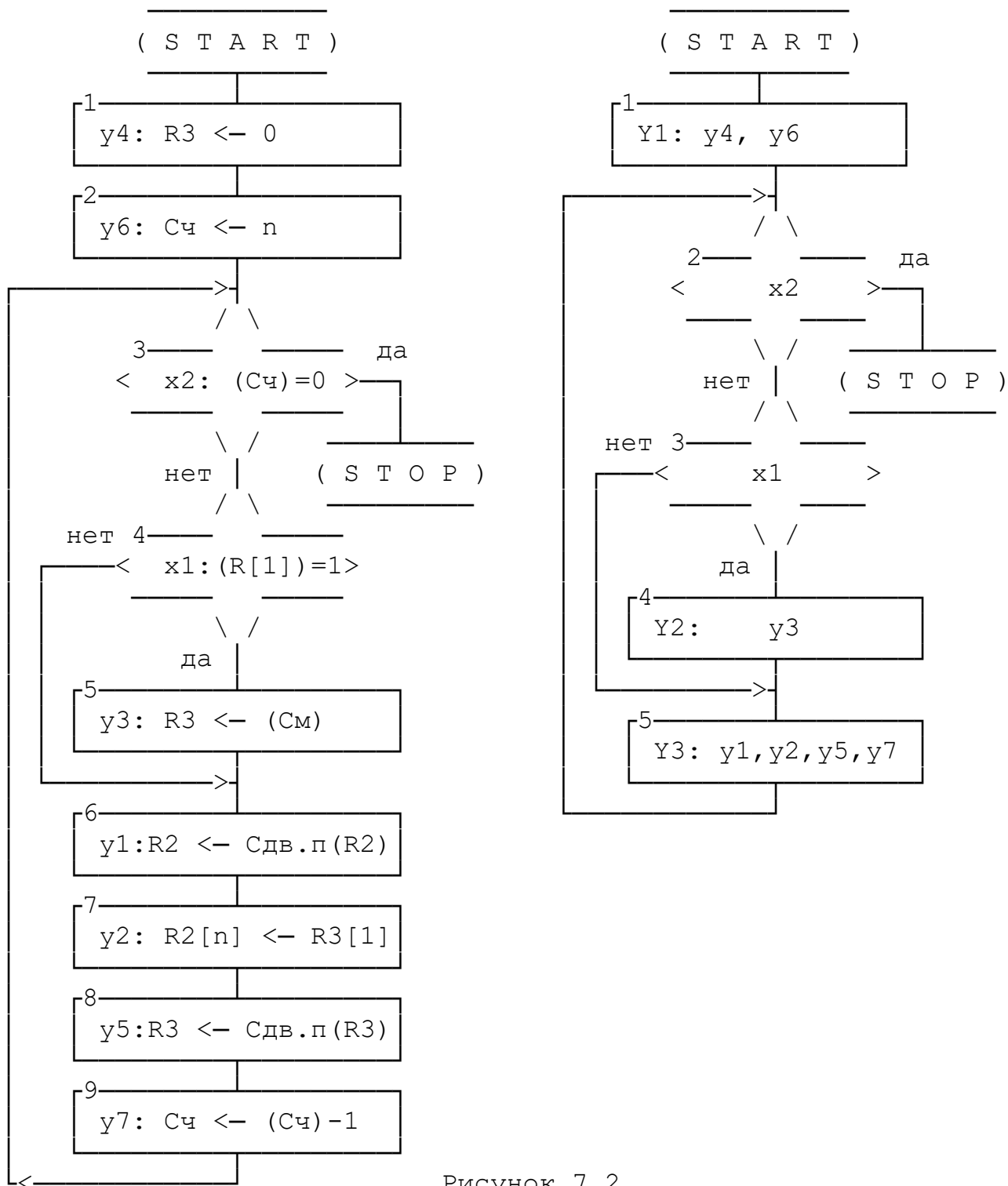


Рисунок 7.2

Очевидно, микрооперации y4 и y6 могут быть объединены в микрокоманду Y1, y3 не может быть объединена с другими

микрооперациями, она представляет микрокоманду Y2; y1, y2, y5, y7 могут выполняться одновременно и объединены в Y3. На рисунке – схема алгоритма в микрокомандах.

Управляющее устройство – это логическое устройство последовательностного типа. Микрокоманда, выдаваемая в следующем тактовом периоде, зависит от того, какая микрокоманда выдается в текущем тактовом периоде, или, иначе, от состояния в котором находится устройство.

Можно построить структурную схему управляющего устройства. Она состоит из комбинационного узла и регистра состояния.

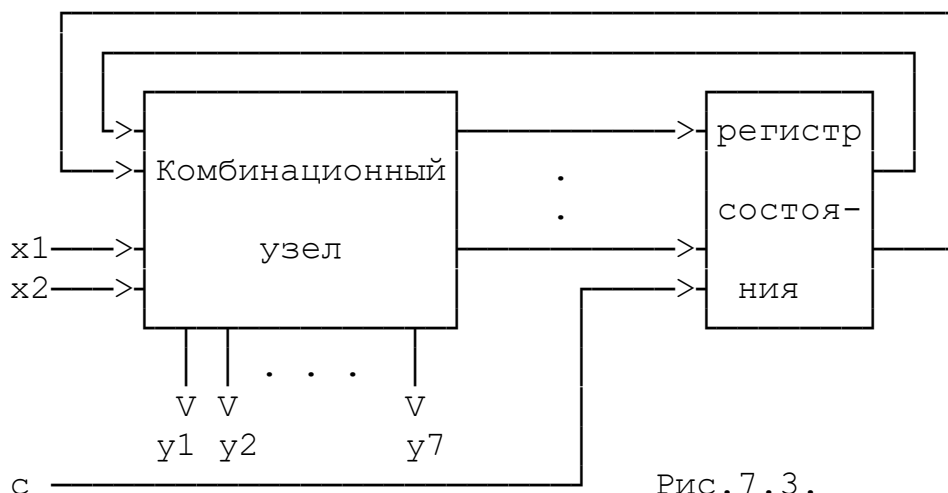


Рис.7.3.

ВЫПОЛНЕНИЕ ПРОГРАММЫ

Мы рассмотрели реализацию управляющего устройства для выполнения операции умножения. Очевидно, могут быть построены подобные устройства для выполнения других операций. И если в управляющем устройстве процессора предусмотреть такие устройства, то, включая то или иное устройство, можно обеспечить выполнение различных операций на одном и том же оборудовании операционного устройства.

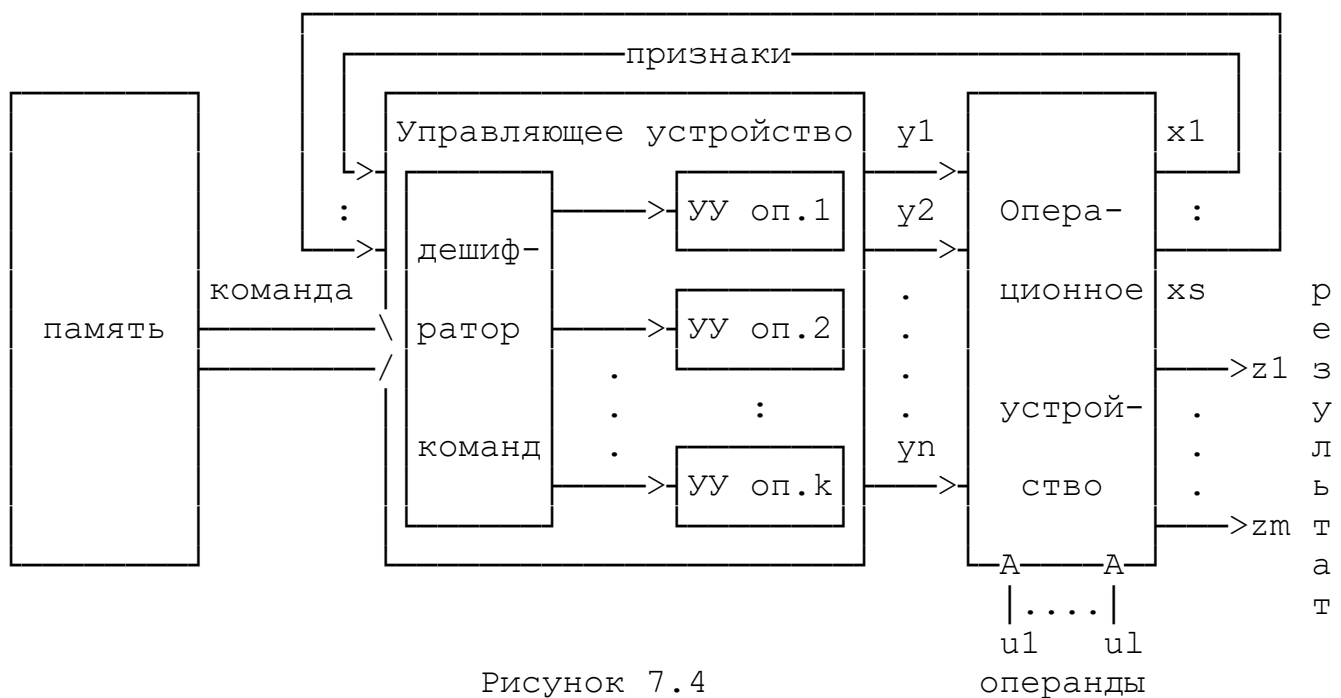


Рисунок 7.4

Операцию, подлежащую исполнению в процессоре, называют командой. С помощью дешифратора код команды можно преобразовать в сигналы, производящие включения устройств, которые управляют выполнением соответствующих операций.

При этом возникает возможность записывать алгоритм сложной задачи в виде последовательности команд, которая будет соответствовать последовательности таких выполняемых простых операций, как умножение, деление и др. Такая последовательность команд образует ПРОГРАММУ, хранимую в памяти. Считывая последовательно из памяти команды и исполняя их в процессоре можно решить сложную задачу.

7.4 СИНТЕЗ УПРАВЛЯЮЩЕГО УСТРОЙСТВА НА ОСНОВЕ ПРОГРАММИРУЕМОЙ ЛОГИКИ.

Выше показано, что выполнение операций в МП осуществляется в виде последовательности выполняемых микрокоманд. Можно предусмотреть, отличающийся от принципа схемной логики, способ формирования в управляющем устройстве управляющих сигналов, под действием которых в операционном устройстве выполняются микрокоманды.

Управляющие сигналы $y_1 \dots y_n$ на выходе управляющего устройства в каждом тактовом периоде имеют уровни логических 0 и 1. Таким образом, каждой микрокоманде на выходе управляющего устройства соответствует некоторая кодовая комбинация. Такие кодовые комбинации, называемые микрокомандами, можно хранить в управляющей памяти. Последовательность микрокоманд, предназначенную для выполнения некоторой операции, называют микропрограммой. При этом выполнение операции сводится к выборке из управляющей памяти микрокоманд микропрограммы и выдаче с их помощью управляющих сигналов $y_1 \dots y_n$ в операционное устройство.

В управляющей памяти можно хранить много микропрограмм, предназначенных для выполнения различных операций. По выбранной из оперативной памяти команде в управляющей памяти находится соответствующая команде микропрограмма. Далее путем последовательного считывания микрокоманд найденной микропрограммы и их выполнения в операционном устройстве реализуется предусматриваемая командой операция.

Такой способ реализации операций называется МИКРОПРОГРАММНЫМ способом, а построенное на этом принципе устройство – управляющим устройством с ПРОГРАММИРУЕМОЙ ЛОГИКОЙ.

На рисунке 8.5 изображена структурная схема МП с управляющим устройством, построенным на принципе программируемой логики. Функции блока микропрограммного управления (БМУ) сводятся к определению адреса очередной микрокоманды (МК) в управляющей памяти (УП). Поступающая из оперативной памяти (ОП) команда содержит адрес первой микрокоманды той микропрограммы, которая реализует предусматриваемую

командой операцию. Таким образом решается проблема поиска в УП микропрограммы, соответствующей данной команде. Адреса всех последующих микрокоманд определяются в БМУ следующим образом.

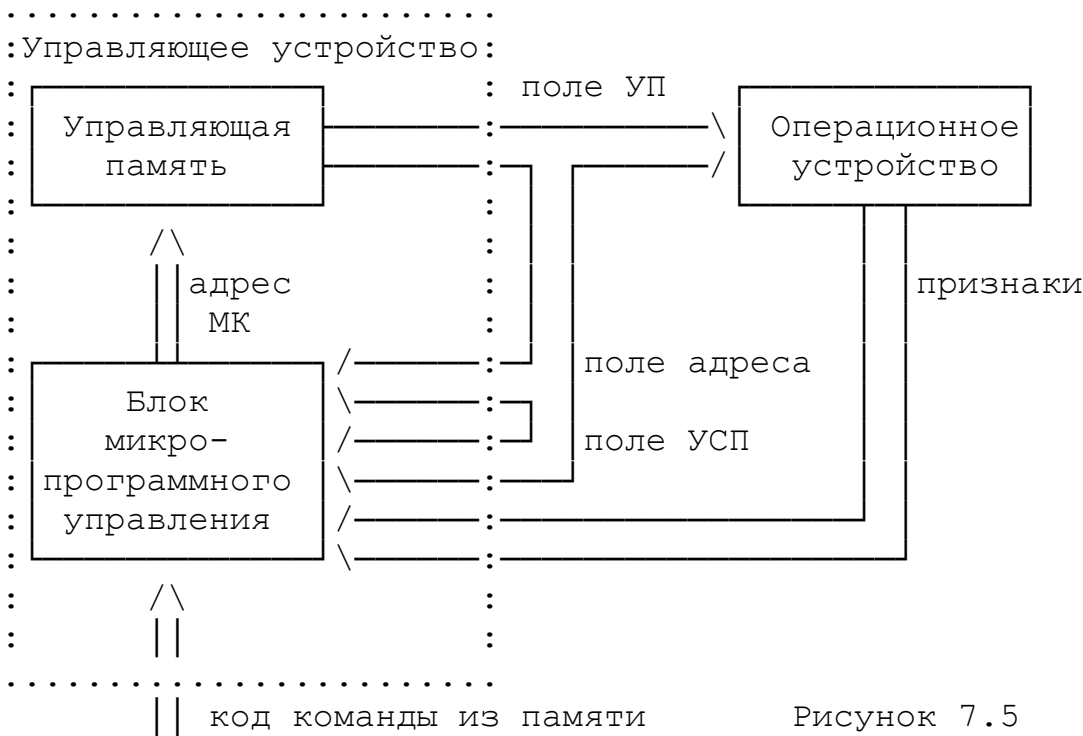


Рисунок 7.5

Рассмотрим формат микрокоманды.

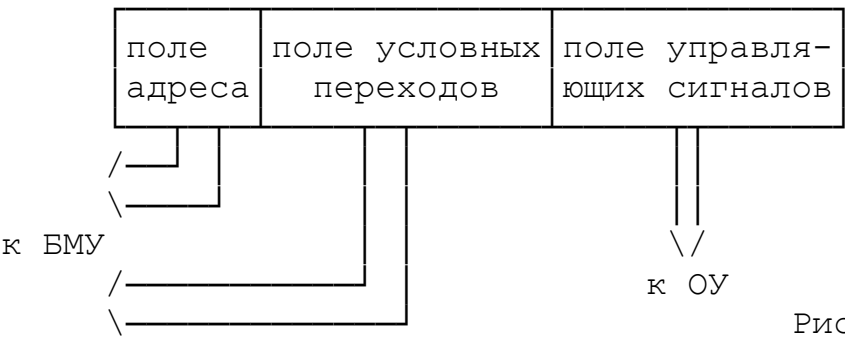


Рисунок 7.6

В формате микрокоманды предусматривается поле адреса, которое содержит адрес очередной микрокоманды. Таким образом, считав из управляющей памяти микрокоманду, по содержимому ее поля адреса узнаем адрес следующей МК, но так можно получить адрес МК при отсутствии в алгоритме разветвлений, т.е. условных переходов (УСП). Для реализации условных переходов в микрокоманде можно предусмотреть поле условных переходов, в котором указывается имеет ли место

условный или безусловный переход, и в случае условного перехода – на значение каких условий следует ориентироваться при определении адреса очередной МК.

Пусть поле условных переходов построено следующим образом: Один из разрядов поля указывает вид перехода (например: 0 – безусловный переход; 1 – условный переход) .

Для каждого условия в поле условного перехода имеется разряд, указывающий участие данного условия в определении адреса. Если условный переход осуществляется по некоторому условию, то формирование адреса очередной МК будем осуществлять замещением младшего разряда содержимого поля адреса текущей МК значением соответствующего условия. Такую операцию называют МОДИФИКАЦИЕЙ АДРЕСА.

Поле управляющих сигналов используется для подачи управляющих кодов в операционное устройство.

ПРИМЕР ПОСТРОЕНИЯ МИКРОПРОГРАММЫ

Построим микропрограмму для выполнения рассмотренной выше операции умножения. Выберем формат микрокоманды:

Поле адреса 4 разряда				поле услов- ных перех.			Поле управляющих сигналов 7 разрядов						
				П	x1	x2							

4-х разрядное поле адреса позволяет обращаться в любую ячейку управляющей памяти с 16 ячейками.

Поле условных переходов содержит 3 разряда: разряд П, наличие 1 в котором указывает на то, что имеет место условный переход; разряды x1 и x2, наличие 1 в которых определяет условие, по которому

происходит условный переход.

Поле управляющих сигналов содержит 7 разрядов и обеспечивает выдачу 7 различных микроопераций.

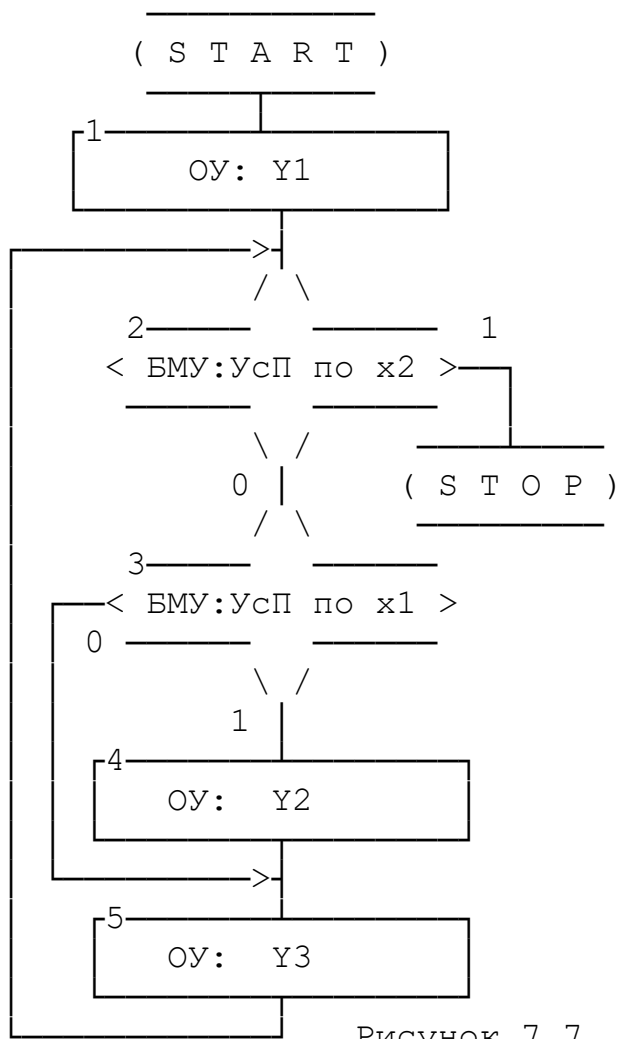


Рисунок 7.7

Для хранения составляемой микропрограммы используем ячейки управляющей памяти с нарастающими адресами 0000; 0001; 0010; ... Ориентируясь на схему алгоритма (в командах) построим схему алгоритма в микрокомандах.

Адрес упр.пам.	Содержимое ячейки (микрокоманда)
0 0 0 0 > 0 0 0 1 <	МК1 (МК ОУ:Y1; МК БМУ:ВП)
>x2=0 > 0 0 1 0 >x2=1 > 0 0 1 1	МК2 (МК ОУ: -; МК БМУ:УсП по x2)
>x1=0 > 0 1 0 0 < >x1=1 > 0 1 0 1 <	МК3 (МК ОУ: -; МК БМУ:УсП по x1)
	МК6 (продолжение)
	МК5 (МК ОУ:Y3; МК БМУ:ВП)
	МК4 (МК ОУ:Y2; МК БМК:ВП)

Микропрограмма приведена в таблице 7.1.

адрес ячейки	М и к р о к о м а н д а											
	МК БМУ				МК ОУ							
	поле адреса	поле услов. перехода			y1	y2	y3	y4	y5	y6	y7	
		П	x1	x2								
0000	0001	0	x	x	0	0	0	1	0	1	0	МК1
0001	0010	1	0	1	0	0	0	0	0	0	0	МК2
0010	0100	1	1	0	0	0	0	0	0	0	0	МК3
0011	-	-	-	-	-	-	-	-	-	-	-	МК6
0100	0001	0	x	x	1	1	0	0	1	0	1	МК5
0101	0100	0	x	x	0	0	1	0	0	0	0	МК4

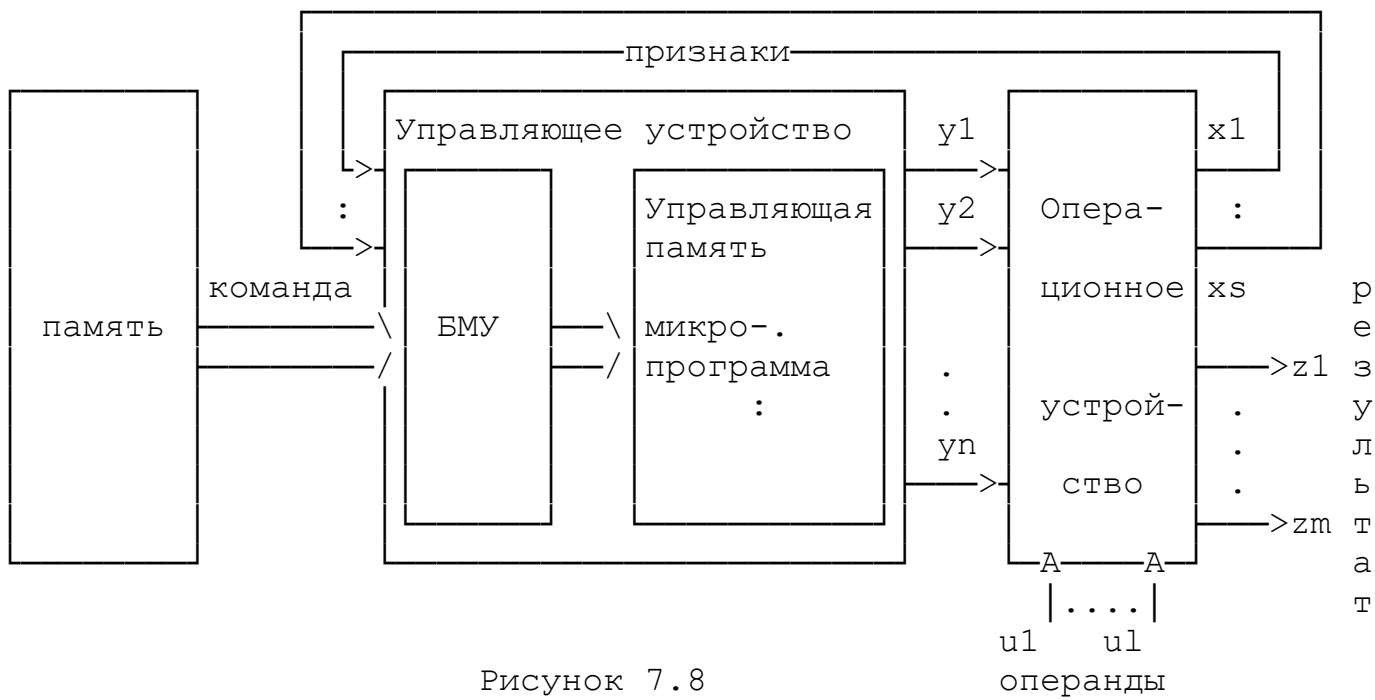


Рисунок 7.8

10. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА. МИКРОСХЕМЫ ПАМЯТИ.

Полупроводниковые запоминающие устройства (ЗУ) являются одним из основных типов ЗУ ЭВМ и устройств автоматики. Полупроводниковые ЗУ – это БИС, предназначенные для хранения и считывания двоичной информации.

Микросхемы ЗУ различают по большому набору параметров:

- Назначение (ОЗУ, ПЗУ);
- Способу хранения информации: статические,
динамические,
на основе резистивных перемычек,
на основе транзисторов с «плавающими» затворами,
на основе ферросегнетоэлектрического эффекта,
на основе мемристоров,
на цилиндрических магнитных доменах (ЦМД)
и другие физические эффекты;
- Емкости;
- Быстродействию;
- Энергопотреблению и питанию;
- Технологии изготовления;
- Типу выхода: третье состояние,
открытый коллектор,
открытый эмиттер;
- и другим параметрам.

Микросхемы памяти изготавливают по полупроводниковой технологии на основе кремния с высокой степенью интеграции компонентов на кристалле, что определяет их принадлежность к большим интегральным схемам (БИС). Конструктивно БИС ЗУ представляют собой полупроводниковый кристалл с площадью в несколько десятков квадратных миллиметров, заключенный в корпус..

10.1 АРХИТЕКТУРА БИС ЗУ

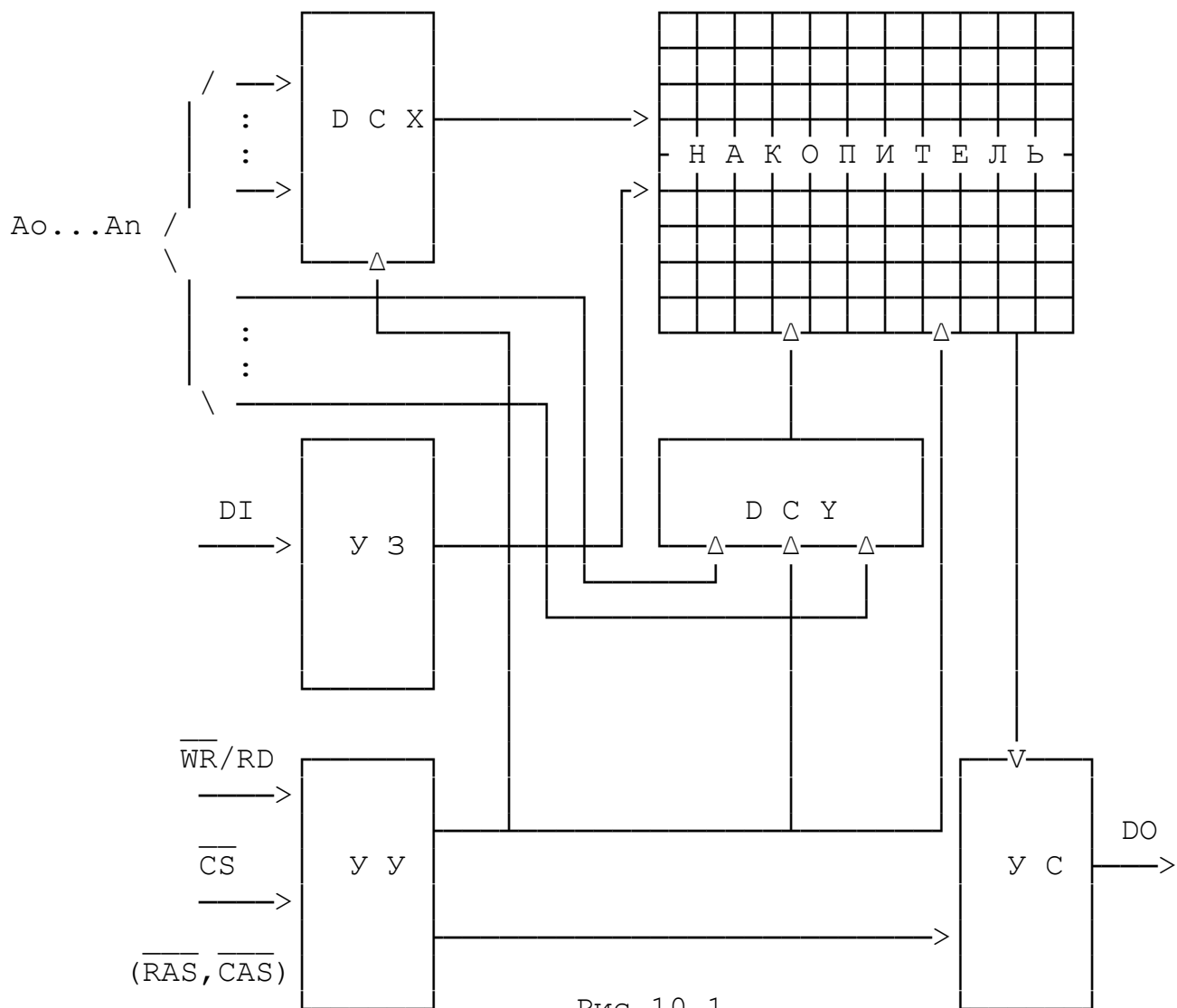


Рис.10.1.

Устройства ЗУ состоят из следующих узлов:

- Накопителя (НК)
- Дешифраторов строк и столбцов (DCX, DCY)
- Устройства записи (УЗ)
- Устройства считывания (УС)
- Устройства управления (УУ)

В зависимости от типа ЗУ те или иные типовые узлы могут в схеме отсутствовать, например БИС ПЗУ не имеют устройства записи. Основной составной частью МС ЗУ является массив элементов памяти, объединенных в матрицу накопителя. Элемент памяти (ЭП) может хранить

один бит информации. Каждый ЭП имеет свой адрес. Для обращения к ЭП необходимо выбрать его с помощью кода адреса, сигналы которого подведены к соответствующим выводам микросхемы.

Запоминающее устройство, ОЗУ или ПЗУ, которое допускает обращение по адресу к любому ЭП в произвольном порядке, называют ЗУ с произвольной выборкой (ЗУПВ).

Разрядность кода адреса m , равная числу двоичных единиц в нем определяет информационную емкость МС ЗУ, т.е. число ЭП в матрице накопителя, которое можно адресовать: оно равно 2^m .

Например, МС ЗУ, у которой число адресных разрядов $m = 10$, содержит в матрице $2^{10} = 1024$ элемента памяти, т.е. имеет информационную емкость 1024 слова. Для ввода и вывода информации служат входные и выходные выводы МС. Многие МС ЗУ имеют совмещенный вход - выход.

Для управления режимом работы микросхеме необходимы управляющие сигналы. Такими сигналами являются:

- сигнал "Запись/Считывание" ($\overline{WR}/RD$);
- сигнал "Выбор кристалла" ($\overline{CS}$)

и другие.

10.2. СТАТИЧЕСКИЕ ЗУ.

Статическое полупроводниковое ЗУ состоит из элементов памяти - триггеров на биполярных или МОП - транзисторах.

10.2.1. СТАТИЧЕСКИЕ ЗУ НА БИПОЛЯРНЫХ ТРАНЗИСТОРАХ.

Биполярный элемент памяти (рис.10.2) состоит из связанных между собой многоэмиттерных транзисторов (МЭТ), образующих RS-триггер. Выборка элемента производится с помощью двух шин выборки x и y , которые связаны с эмиттерными выводами обоих транзисторов. Третий

эмиттерный вывод служит для считывания и записи информации в элемент, в котором один из транзисторов может находиться в проводящем состоянии, а другой – закрыт. Если ячейка не выбрана, то хотя бы на одну из адресных шин, а следовательно на эмиттерные выводы, связанные с этими шинами, поступает напряжение низкого уровня. Выборка элемента происходит путем подачи напряжения высокого уровня на обе адресные шины x и y . При этом считывающий эмиттер, который связан с открытым транзистором, перехватит часть эмиттерного тока адресных эмиттеров и через шину логического "0" или логической "1" направит этот ток в один из двух считывающих усилителей. Считывание не приводит к потере информации, так как состояние элемента не изменяется под действием напряжения высокого уровня, которое появляется на адресных шинах x и y .

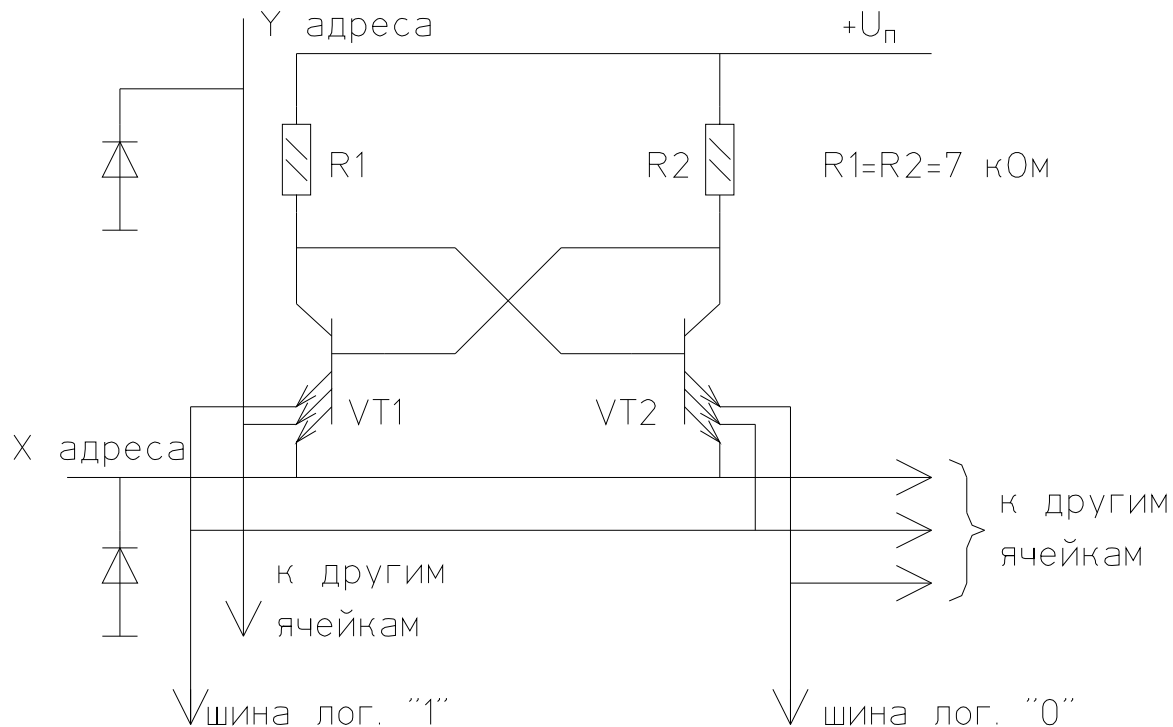


Рисунок 10.2 – Ячейка статического ЗУ на биполярных транзисторах

При записи считывающие эмиттеры служат для перевода элемента в новое состояние. На эмиттер транзистора, который в новом состоянии

должен быть открыт, подается низкое напряжение, на эмиттер транзистора, который должен запереться, - высокое напряжение. Этот транзистор запирается, когда на оба других эмиттера подается напряжение высокого уровня, как это и происходит при выборке ячейки. Другой транзистор будет открываться, так как на его базу через коллекторный резистор подается высокое напряжение.

10.2.2 ЭЛЕМЕНТ СТАТИЧЕСКОГО МОП - ЗУ

Элемент статического МОП - ЗУ (Рис. 10.3) состоит из соединенных между собой МОП - транзисторов VT1 и VT2, которые образуют триггер. При считывании информации из ячейки, которая является элементом матричной структуры, транзисторы VT5, VT7 и VT6, VT8 открываются и содержимое запоминающей ячейки попадает на активные шины логического "0" и логической "1". Информация считывается соответствующим усилителем и преобразуется в TTL - уровни для дальнейшей обработки.

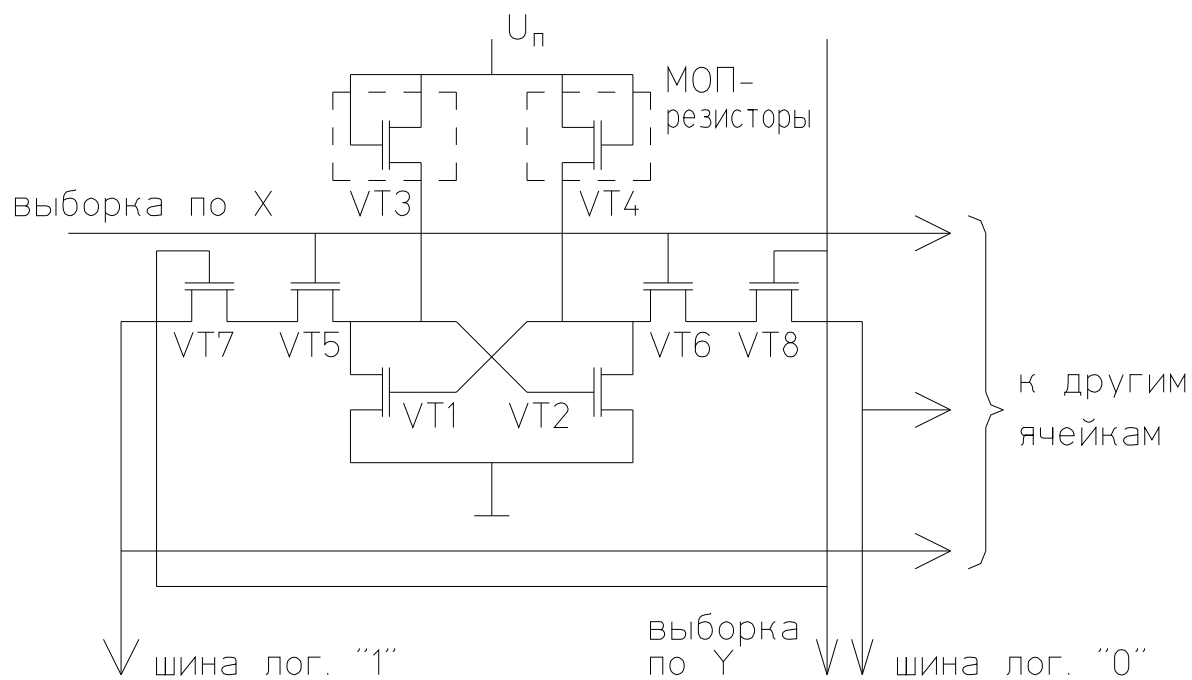


Рисунок 10.3 - Ячейка статического ЗУ на МОП - транзисторах

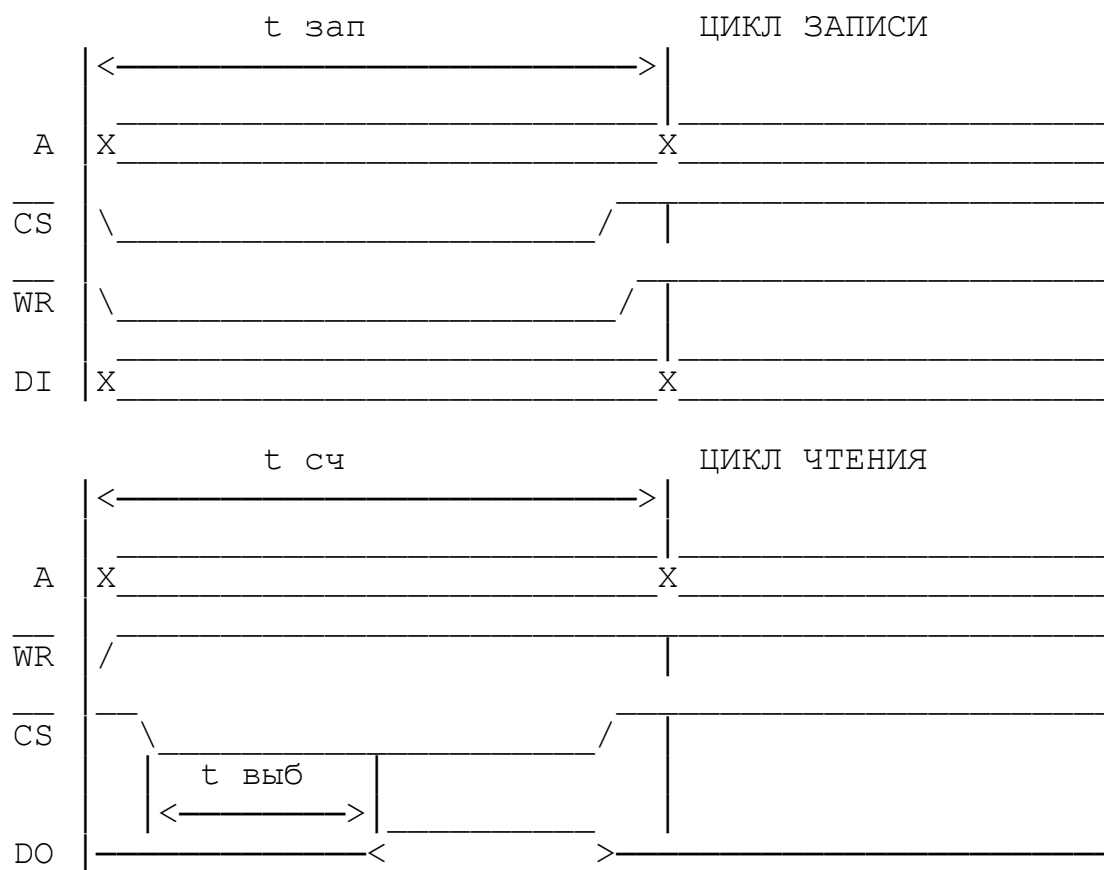


Рисунок 10.4

10.3 ДИНАМИЧЕСКИЕ ЗУ

В микросхемах памяти динамического типа функцию ячейки памяти выполняет электрический конденсатор, образованный внутри МДП - структуры. Информация представляется в виде заряда. Наличие заряда на конденсаторе соответствует логическому "0", отсутствие - "1". Поскольку время сохранения конденсатором заряда ограничено, необходимо предусмотреть периодическое восстановление (регенерацию) записанной информации. В этом состоит одна из отличительных особенностей динамических ОЗУ. Кроме того, для них необходима синхронизация, обеспечивающая требуемую последовательность включений функциональных узлов.

Для изготовления микросхем динамических ОЗУ, в основном, применяют n-МДП - технологию, которая позволяет повышать быстродействие и уровень интеграции микросхем, обеспечивать малые

токи утечки и за этот счет увеличивать время сохранения заряда на запоминающем конденсаторе.

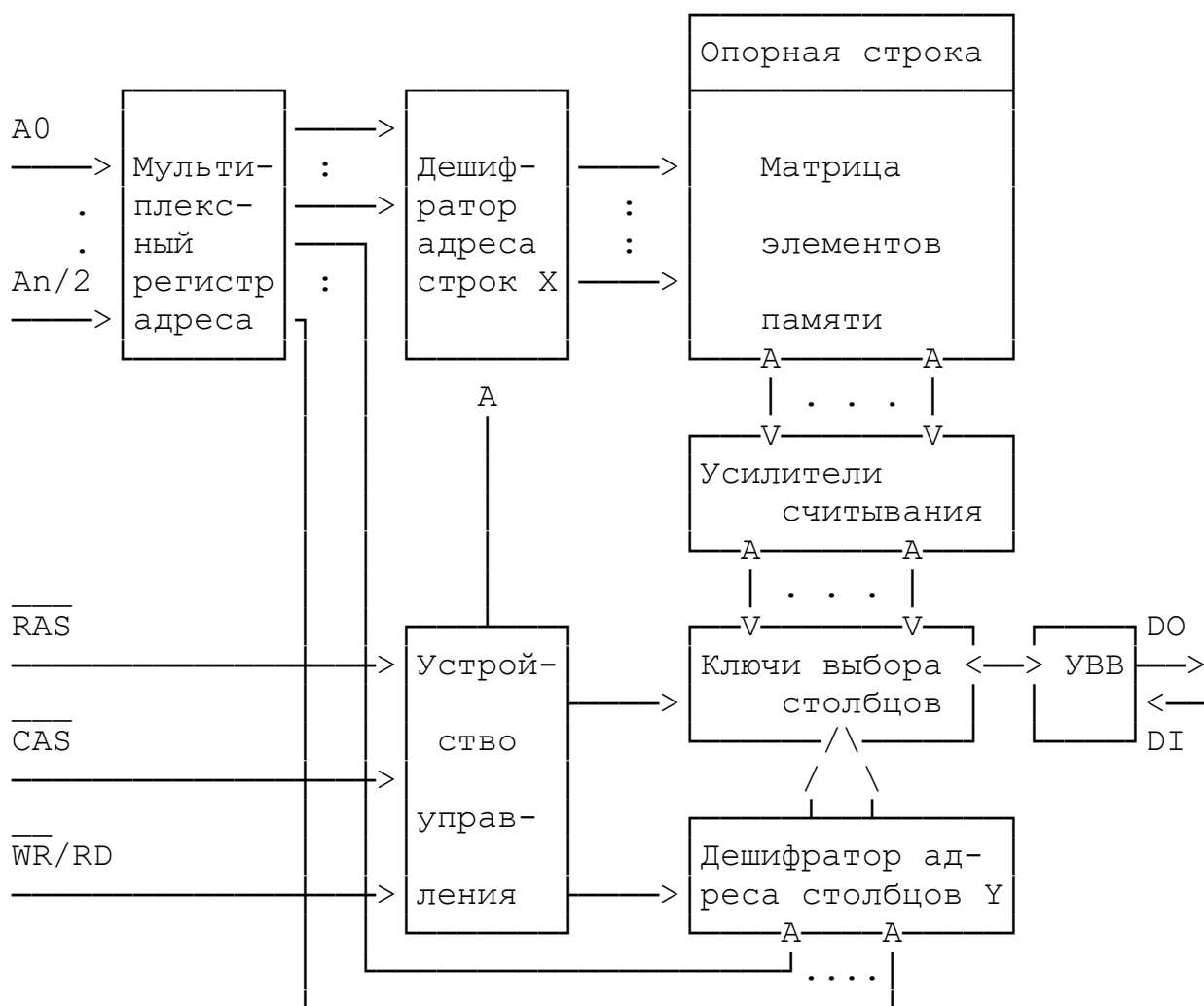


Рисунок 10.5 - Структура микросхем динамического ОЗУ

С целью уменьшения числа необходимых адресных выводов корпуса в микросхемах динамической памяти код адреса вводят по частям: вначале младшие разряды, сопровождая их стробирующим сигналом RAS, затем старшие со стробирующим сигналом CAS. Внутри микросхемы коды адреса строк и столбцов фиксируются и осуществляет выборку адресуемого ЭП (см. временные диаграммы на рисунке 10.6).

Для обеспечения надежного сохранения записанной в накопителе информации, реализуют режим принудительной регенерации. Регенерация информации в каждом ЭП должна осуществляться не реже интервала времени, называемого ПЕРИОДОМ РЕГЕНЕРАЦИИ. Период регенерации

зависит от конкретного типа микросхем памяти и обычно составляет около 2 мс. Режим регенерации заключается в последовательном опросе в режиме чтения всех строк накопителя, столбцы при этом не выбираются.

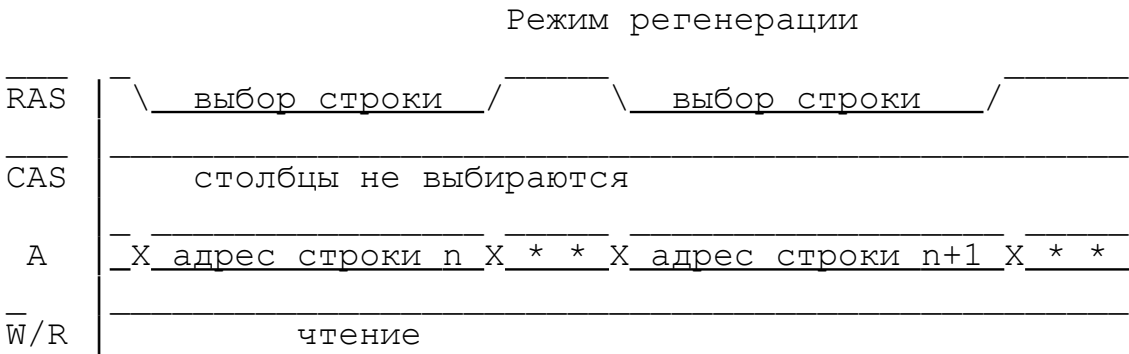
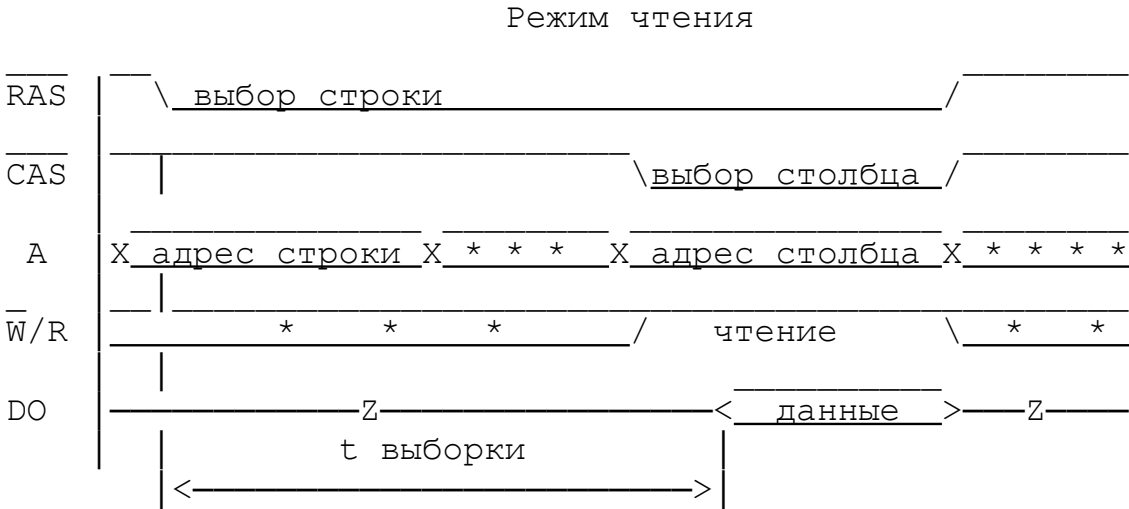
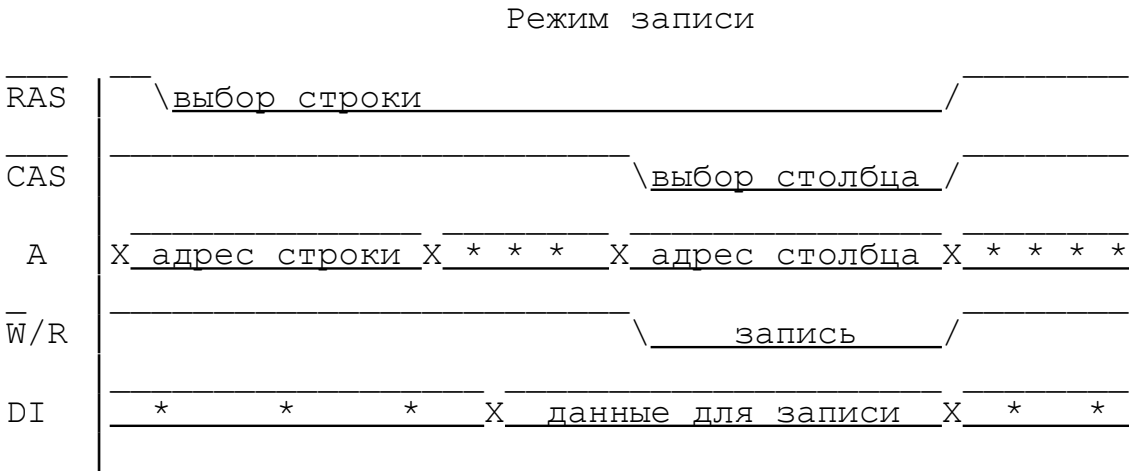


Рисунок 10.6

Общим свойством всех МС ПЗУ являются их многоразрядная (словарная) организация, режим считывания как основной режим работы и энергонезависимость.

10.4.1 МИКРОСХЕМЫ ПЗУМ

Матрица накопителя состоит из ЭП, каждый из которых расположен на пересечении строки и столбца. ЭП представляет собой резистивную или полупроводниковую перемычку включенную между строкой и столбцом через транзистор VT_{ij} (рис. 10.7). Наличие перемычки в выбранной ячейке создает высокий потенциал в шине столбца и тем самым создает на выходе напряжение логической "1". Отсутствие перемычки изолирует источник питания от информационной шины и создает на выходе напряжение логического "0". Информацию в матрицу заносят в процессе изготовления МС.

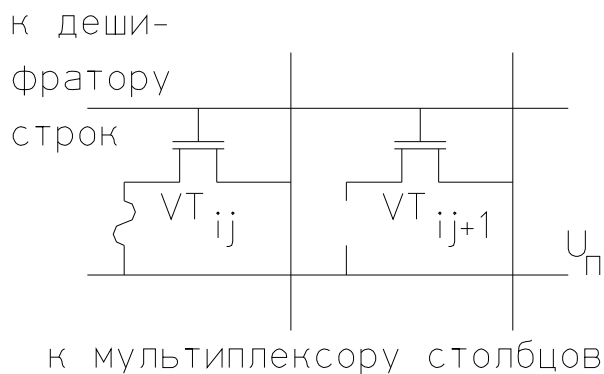


Рисунок 10.7 - Ячейка ПЗУ

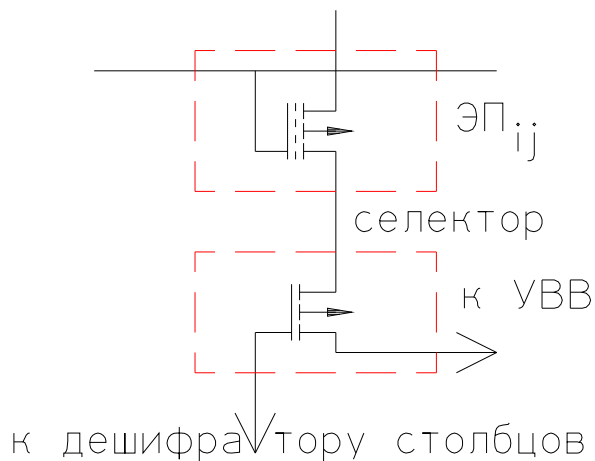


Рисунок 10.8 - Ячейка РПЗУ

10.4.2 МИКРОСХЕМЫ ППЗУ

МС ППЗУ по принципу построения и функционирования аналогичны масочным ПЗУ, но имеют существенное отличие в том, что допускают программирование на месте своего применения. Операция программирования заключается в разрушении (пережигании) части плавких

перемычек импульсами тока амплитудой 30 – 50 мА. Перемычки изготавливаются из нихрома (серия К556 и др.), из поликристаллического кремния (К541), из силицида платины и других материалов. Наличие перемычки определяет наличие логической 1, если усилитель считывания является повторителем, и логического 0, если усилитель считывания – инвертор. Следовательно, микросхема ППЗУ в исходном состоянии в зависимости от типа выходного усилителя может иметь заполнение матрицы либо логической 1 (в большинстве случаев), либо 0.

Разновидностью ППЗУ являются программируемые логические матрицы (ПЛМ). Основу ПЛМ составляют матрицы И и ИЛИ. Матрица И выполняет операции конъюнкции над входными переменными и их инверсными значениями. Требуемые логические произведения формируют пережиганием ненужных перемычек между строками и столбцами. Матрица ИЛИ выполняет операцию дизъюнкции над логическими произведениями, сформированными матрицей И. Возможности ПЛМ характеризуются емкостью матриц и числом точек коммутации.

10.4.3 МИКРОСХЕМЫ РПЗУ

Основная отличительная особенность МС РПЗУ заключается в их способности к многократному перепрограммированию самим пользователем. Это свойство МС обеспечено применением ЭП с управляемыми перемычками, функции которых выполняют транзисторы со структурой МНОП (Металл (Al) – Нитрид кремния (Si_3N_4) – Окисел кремния (SiO_2) – Полупроводник (Si)) или транзисторы n-МОП с плавающим затвором с использованием механизма лавинной инжекции заряда ЛИЗМОП. Вся номенклатуру выпускаемых МС РПЗУ можно разделить на две группы:

- РПЗУ с записью и стиранием электрическими сигналами;

- РПЗУ с записью электрическими сигналами и стиранием ультрафиолетовым излучением.

Схема ячейки РПЗУ показана на рисунке 10.8, а временные диаграммы работы всех рассмотренных ПЗУ аналогичны временным диаграммам статического ЗУ в режиме считывания (рисунок 10.4).

10.5 ЗУ НА ОСНОВЕ ЦМД

Другим классом в области энергонезависимых ЗУ является ЗУ, реализованное на цилиндрических магнитных доменах (ЦМД). Если нанести тонкий слой аморфного магнитного материала, например гадолиний - кобальта или гадолиний - железа, на немагнитную подложку, то в этой пленке появятся области, имеющие одинаковое направление намагничивания, которые похожи по форме на коромысло. Под действием внешнего магнитного поля эти области преобразуются в домены цилиндрической формы. На рисунке 10.9 показано, как это происходит. Эти домены ("пузырьки") имеют диаметр от 1 до 5 мкм. Поле, которое требуется для получения доменов такого размера, обеспечивает постоянный магнит, который располагается над подложкой с магнитной пленкой (кристаллом ЦМД - ЗУ).

В очень сильном магнитном поле домены исчезают. Для хранения информации внешнее напряжения питания для таких ЗУ не требуется, поэтому они называются энергонезависимыми.

Домены смещаются под действием магнитного поля, которое направлено по вертикали к основному полю. Для перемещения пузырьков в требуемом направлении и по определенной траектории на поверхность ЦМД - кристалла напыляются тонкие магнитные слои из пермаллоя, которые имеют форму шевронов (рисунок 10.10). Вращая вспомогательное магнитное поле, можно перемещать домены вдоль шеврона. При этом их можно перебрасывать от одного шеврона к другому. Вращающееся вспомогательное магнитное поле получается с помощью двух обмоток, которые питаются треугольными импульсами тока, сдвинутыми по фазе на 90° .

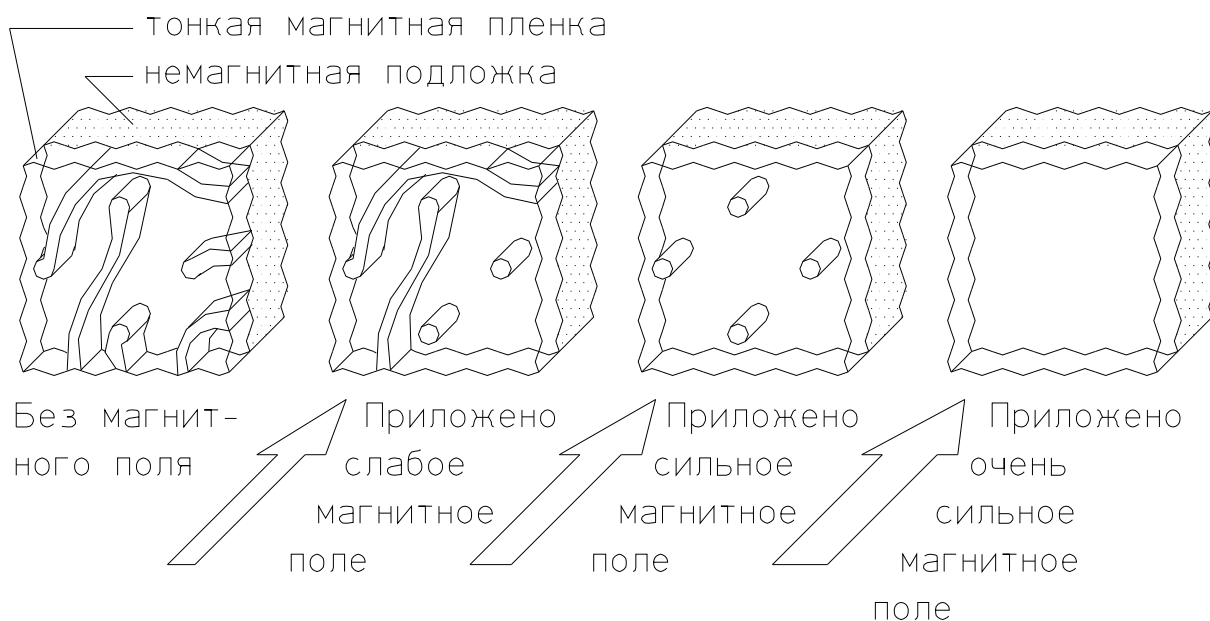


Рисунок 10.9 – Возникновение доменов в тонкой пленке из магнитного материала

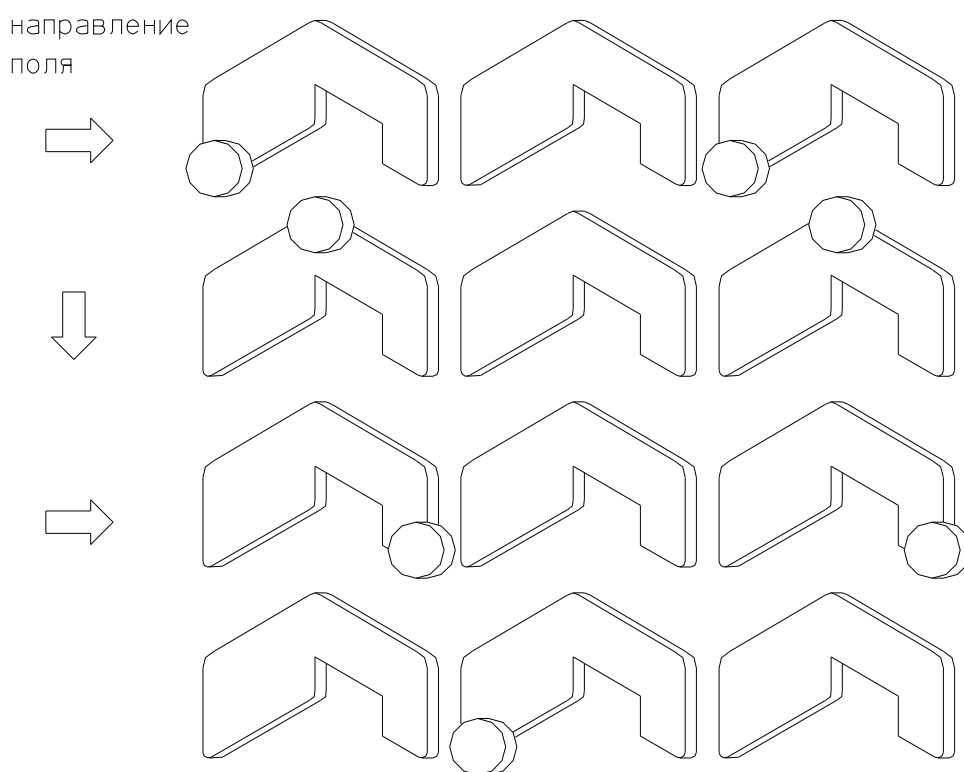


Рисунок 10.10 – Движение доменов вдоль шевронов

Домены в тонкой магнитной пленке можно сформировать с помощью тонкой алюминиевой петли, через которую пропускается ток. Магнитное поле петли накладывается на основное поле постоянного магнита.

Возникающие в пленке домены смещаются вдоль шевронов точно так

же, как и предыдущие, занимая при каждом обороте поля одно определенное положение. Пропуская домены под магниточувствительным датчиком, можно осуществить считывание данных в двоичной форме: наличие пузырька соответствует 1, отсутствие - 0. Такой метод считывания является деструктивным процессом, при котором домен пропадает. Однако, разделив каждый пузырек на две части, можно осуществить и недеструктивный метод считывания. Переместив магнитный домен под слой пермаллоя и разделив на две части, мы получаем его копию. Затем домен - оригинал продолжает свой нормальный путь, а копия направляется к двоичному детектору. Диаметр регенерированного пузырька намного превышает диаметр исходного. Увеличенный пузырек пройдет вблизи магниточувствительного датчика, который и произведет его регистрацию. Полученный в результате импульс после усиления и буферизации обеспечит на выходе двоичную информацию.

Описанное ЦМД - ЗУ представляет регистр сдвига с последовательной записью и считыванием информации. Недостатком является большое время доступа.

На практике используются ЦМД - ЗУ, состоящие из так называемых больших и малых петель шевронов, которые обеспечивают организацию типа "регистр связи - накопительные регистры".

