

1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ЦИФРОВЫХ СИСТЕМАХ.

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

ДВЕНАДЦАТЕРИЧНАЯ система: английская система мер:

1 фут = 12 дюймов

1 шиллинг = 12 пенсов

ШЕСТИДЕСЯТЕРИЧНАЯ система: Вавилон

1 час = 60 минут; $1' = 60''$

$1^{\circ} = 60'$; $1' = 60''$

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: $342 = 3 * 10^2 + 4 * 10^1 + 2 * 10^0 = 300 + 40 + 2$

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.

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
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, $0.25 = 2^{-2}$.

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

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

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

ВОСЬМИРИЧНАЯ система в качестве основания использует число 8 и, соответственно, 8 цифр от 0 до 7. Перевод из десятичной системы в восьмиричную осуществляется по тому-же правилу, что и в случае с двоичной системой. Перевести число из двоичной в восьмиричную 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:

число $453_{10} = 111.000.101_2 = 705_8$

ШЕСТИНАДЦАТЕРИЧНАЯ 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:

$$\begin{array}{ccc} 4 & 5 & 3 \\ \text{число } 453_{10} = 0100.0101.0011_{2-10} \end{array}$$

.

2. ПРАВИЛА ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ.

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:

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

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	$X1_{\text{пр}} = 0.1100$	$X1_{\text{обр}} = 0.1100$	$X1_{\text{доп}} = 0.1100$
5	$X2_{\text{пр}} = 0.0101$	$X2_{\text{обр}} = 1.1010$	$X2_{\text{доп}} = 1.1011$
- 7	$S_{\text{пр}} = 0.0111$	10.0110	10.0111
		└─>─┘	<─┘
		$S_{\text{обр}} = 0.0111$	$S_{\text{доп}} = 0.0111$
$Y = 5 - 12 = -7$		$X1_{\text{обр}} = 0.0101$	$X1_{\text{доп}} = 0.0101$
		$X2_{\text{обр}} = 1.0011$	$X2_{\text{доп}} = 1.0100$
		$S_{\text{обр}} = 1.1000$	$S_{\text{доп}} = 1.1001$
		$S_{\text{пр}} = -0.0111$	$S_{\text{пр}} = -0.0111$

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
+	+ 1101
13	0000
-----	1101
143	-----
	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 соответствующим 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 определением 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 .

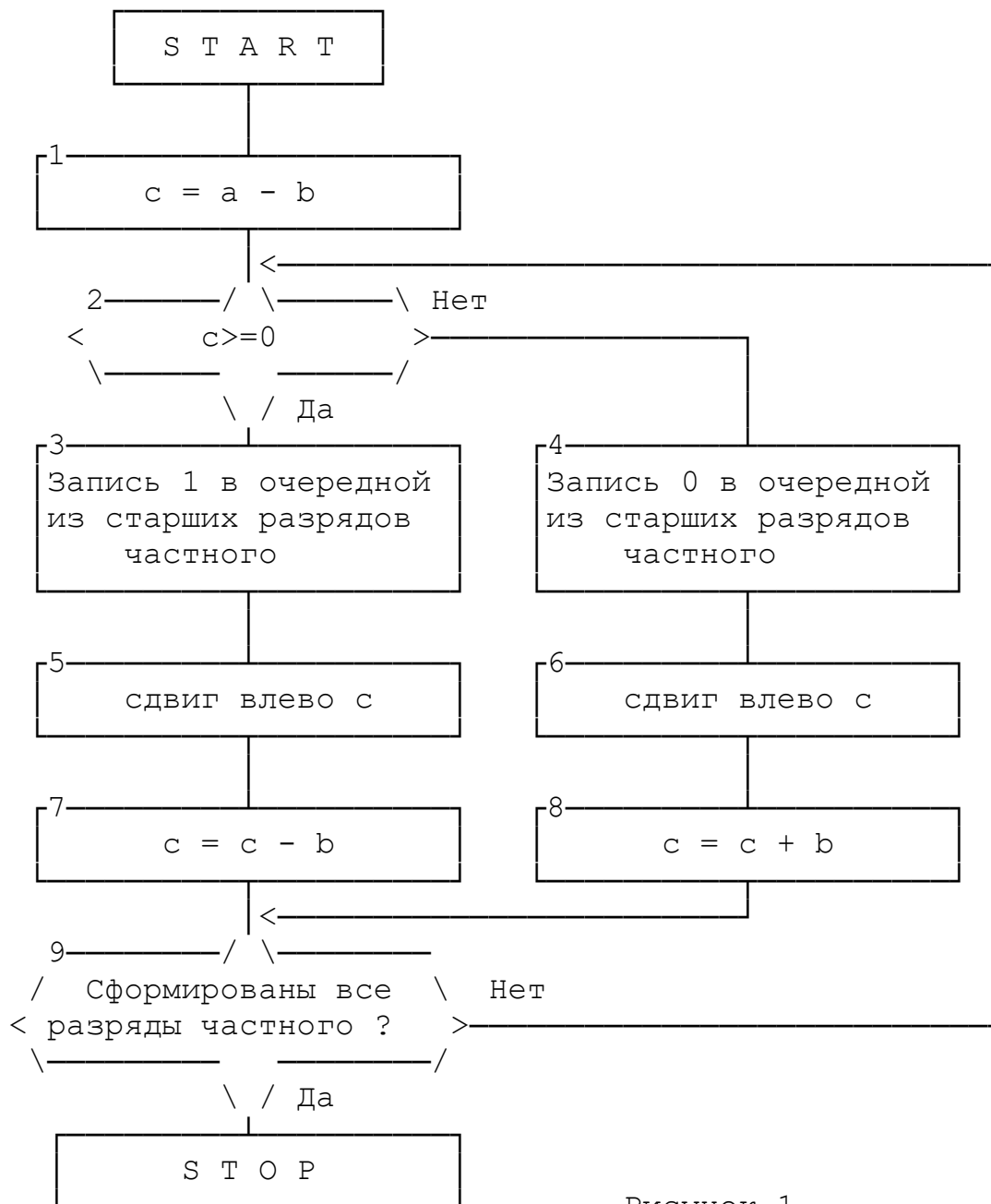


Рисунок 1.

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

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

3. 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:

$A = \text{"Земля плоская"} , B = \text{"Парта черная"} .$ 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.

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.

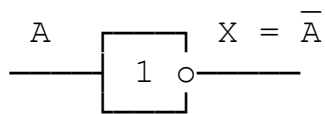
Наибольший практический интерес представляют функции отрицания, логического умножения и логического сложения..

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

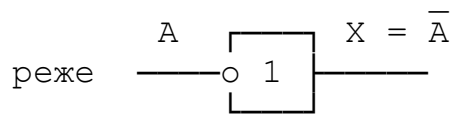
Графически функция НЕ обозначается в виде круга возле вывода (МЭК-117-15) (ГОСТ 2.743-91):

$$X = \bar{A}$$

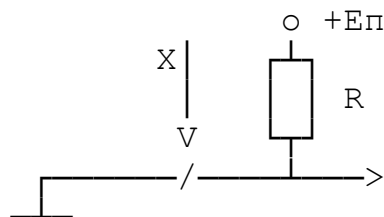
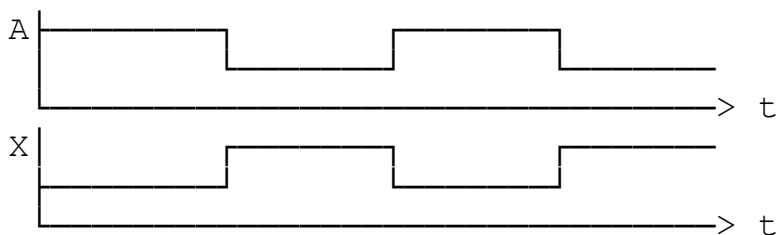
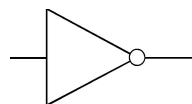
A	X
0	1
1	0



или



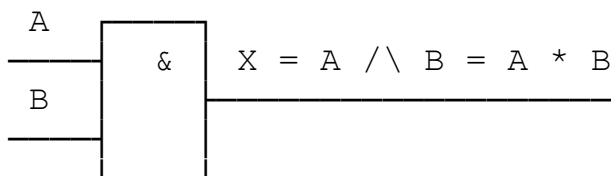
По американской спецификации milspec:



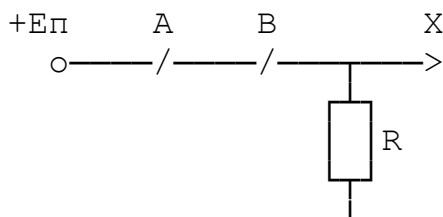
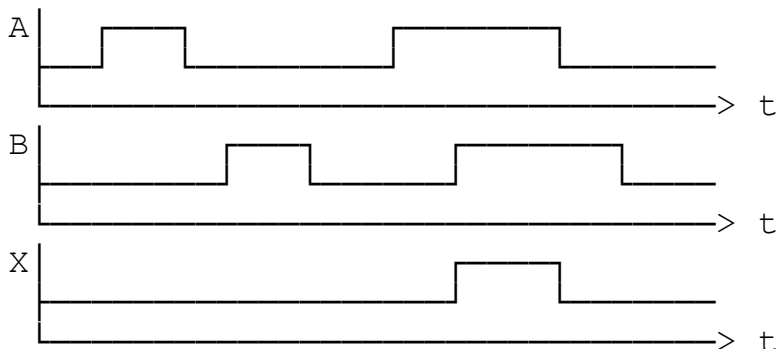
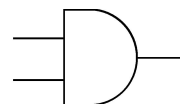
2. Логическое умножение И двух переменных A и B есть логическая функция X, которая истинна только тогда, когда одновременно истинны обе входные переменные. Кроме того функцию И называют КОНЪЮНКЦИЕЙ.

$X = A * B$ 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



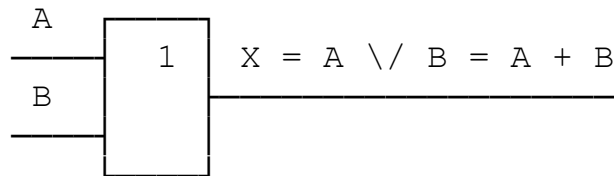
По американской спецификации milspec:



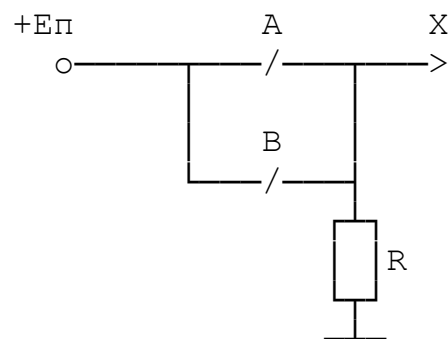
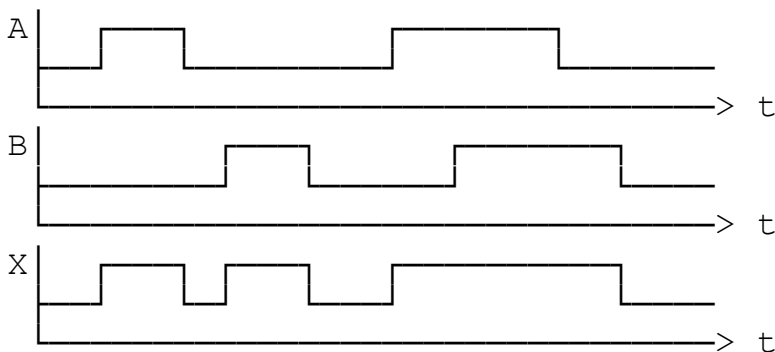
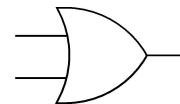
3. Логическая сумма ИЛИ двух переменных А и В есть логическая функция X, которая истинна, когда истинна хотя бы одна из входных переменных. Кроме того функцию ИЛИ называют ДИЗЬЮНКЦИЕЙ.

$X=A+B$ или $X=A \vee B$ Графически функция ИЛИ обозначается в виде значка 1 внутри элемента

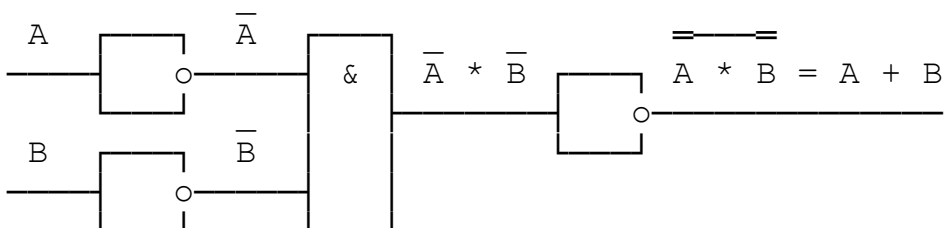
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



По американской спецификации milspec:



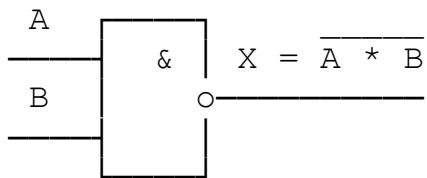
Три рассмотренных функции позволяют реализовать любую логическую комбинацию. Более того, сочетание одной из функций И либо ИЛИ с функцией НЕ также позволяет реализовать любую сложную функцию. Например, реализуем функцию ИЛИ с помощью элементов И и НЕ.



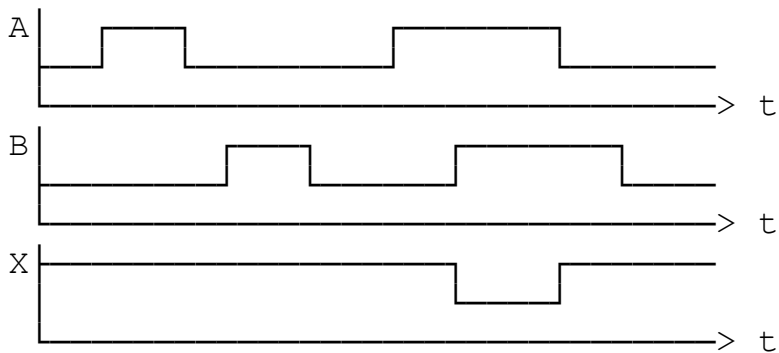
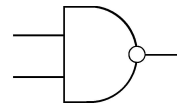
Рассмотрим еще три функции получившие широкое применение в цифровой технике:

4. И-НЕ "Штрих Шеффера"

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

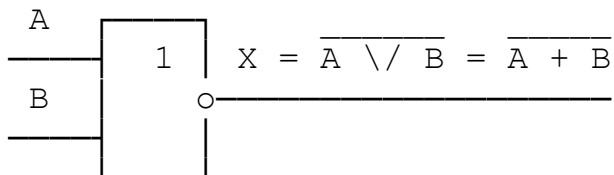


По американской спецификации milspec:

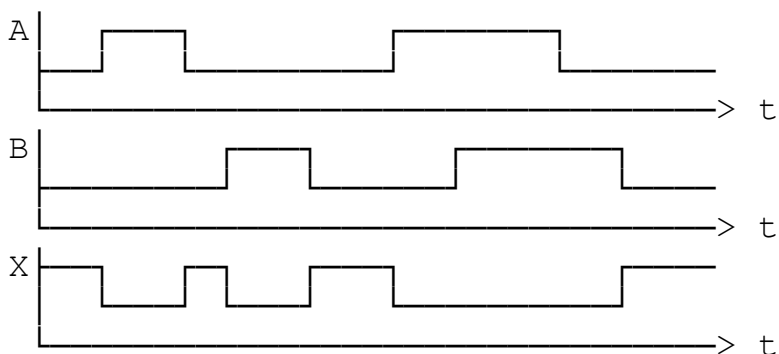
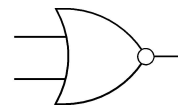


5. ИЛИ-НЕ "Стрелка Пирса"

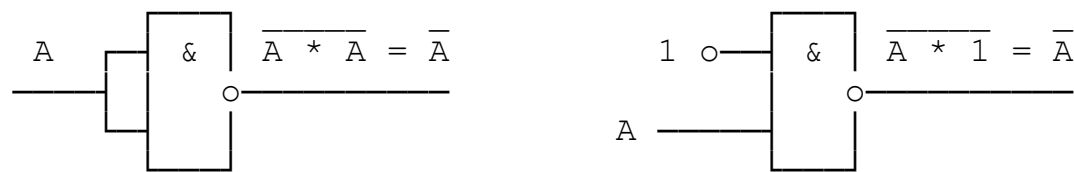
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



По американской спецификации milspec:

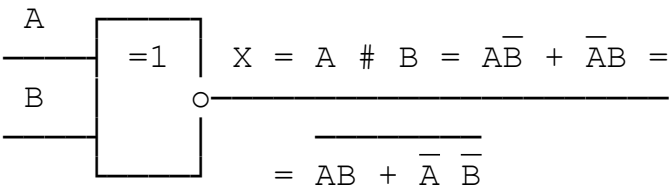


Две рассмотренные функции интересны тем, что являются достаточными для построения любой сложной функции, т.к. содержат в своем составе функцию И либо ИЛИ и инвертор.

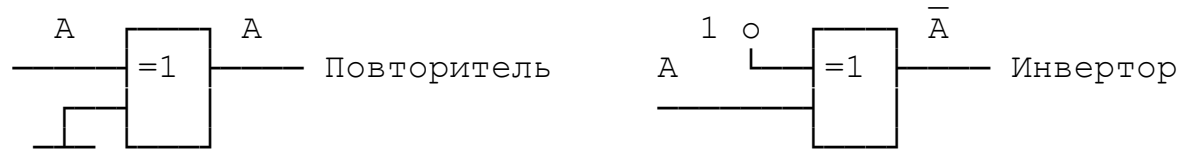
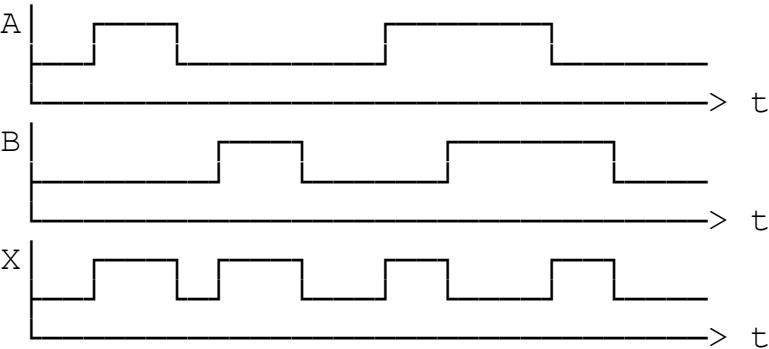
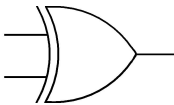


6. "ИСКЛЮЧАЮЩЕЕ ИЛИ" или СУММИРОВАНИЕ ПО МОДУЛЮ 2

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



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:

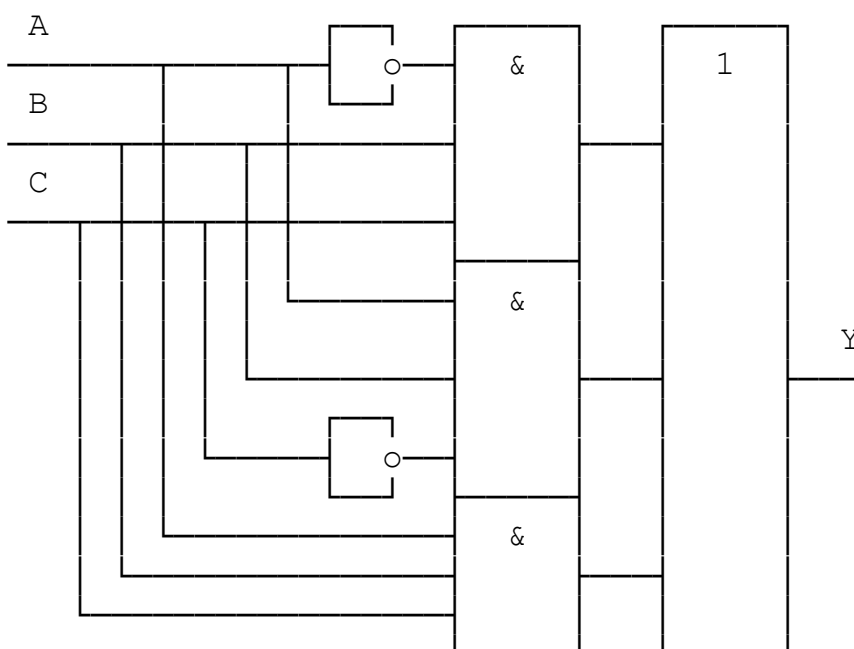
$$\bar{A}BC, A\bar{B}\bar{C}, ABC$$

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

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

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

Если при такой записи каждое слагаемое содержит произведения всех переменных или их отрицаний, то такую форму представления функции называют СОВЕРШЕННОЙ ДИЗЪЮНКТИВНОЙ 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:

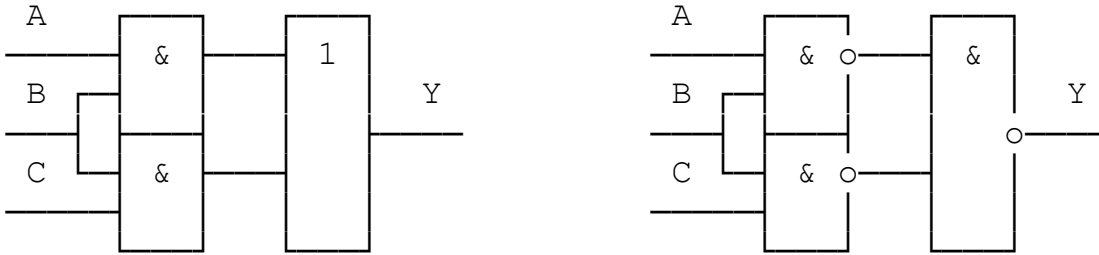
- 3 схемы, выполняющие функцию ЗИ,
- 1 схема ЗИЛИ и
- 2 схемы НЕ.

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.



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

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

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

а и b. Объединение а отражает "склеивание" минтермов ABC и $A\overline{B}C$

$$ABC + A\overline{B}C = AB(C + \overline{C}) = AB$$

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

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

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

Отсюда: $Y = AB + BC$.

4. ОСНОВНЫЕ КОМБИНАЦИОННЫЕ УСТРОЙСТВА.

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

4.1. 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 диодами Шоттки) и КМОП (комплиментарные металл - оксид - полупроводник транзисторы) (КМДП) технологии. Для построения особо быстродействующих схем применяется ЭСЛ (эмиттерно связная логика) технология. Интегрально инжекционная логика (ИИЛ, И2Л) применяется при изготовлении аналого-цифровых схем - ЦАП, АЦП и некоторых микропроцессорных серий.

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 4.1). Поэтому состояние неподключенного входа эквивалентно состоянию лог. 1.

По построению выходного каскада ТТЛШ - схемы различаются на устройства с двухтактным ключом - с двухстабильным состоянием (см. рисунок 4.1), с открытым коллектором и с третьим состоянием (см. рисунок 4.2). 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

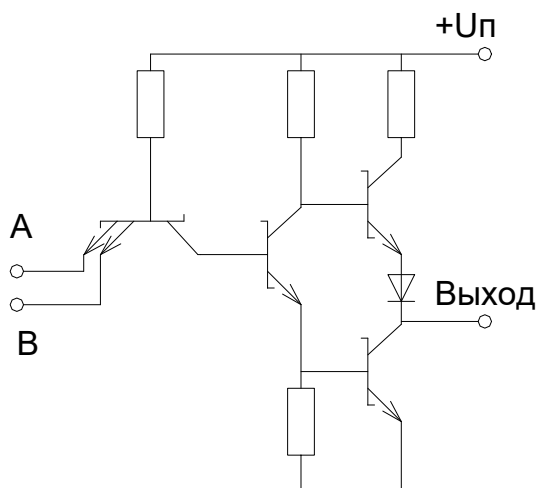


Рисунок 4.1. Элемент 2И-НЕ ТТЛШ

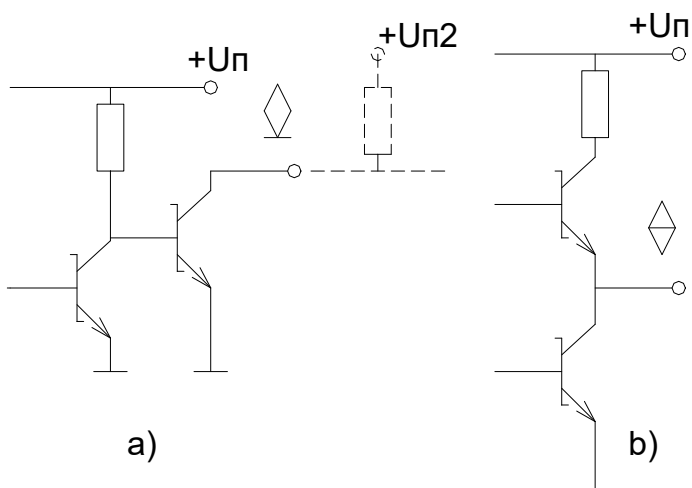


Рисунок 4.2. Выходные каскады ТТЛШ-схем

- а) Открытый коллектор ОК
- б) с Третьим состоянием

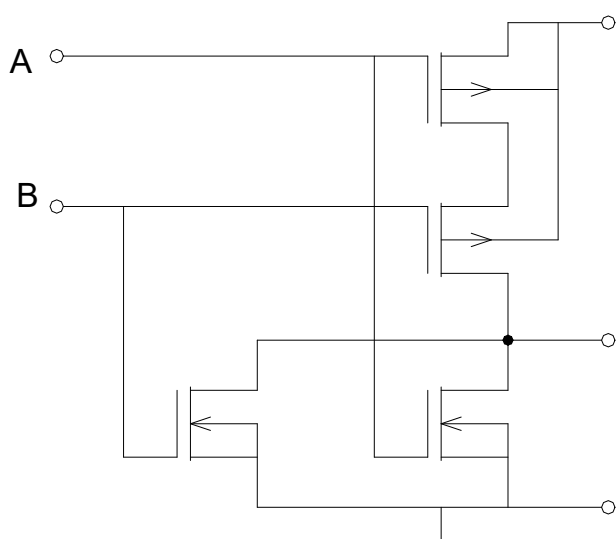


Рисунок 4.3. Элемент 2ИЛИ-НЕ КМОП

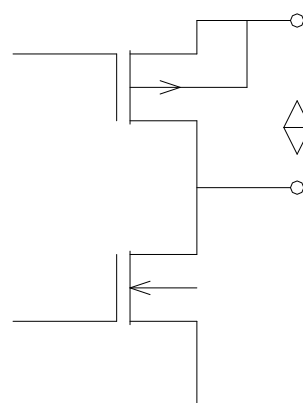


Рисунок 4.4. Выходной каскад КМОП-схемы с ТС

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 4.3).

По построению выходного каскада КМОП - схемы различаются на устройства с двухтактным ключом - с двухстабильным состоянием (см. рисунок 4.3), и с третьим состоянием (см. рисунок 4.4). 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:

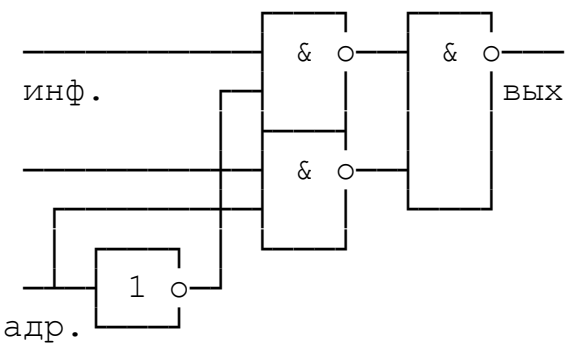
Ам. фирма RCA	отечественные
CD4000 (60 - 70 гг)	K176
CD4000A	K561
CD4000B	K564
CD4000BE	KP1561
74AC	KP1554
54HC	K1564

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.

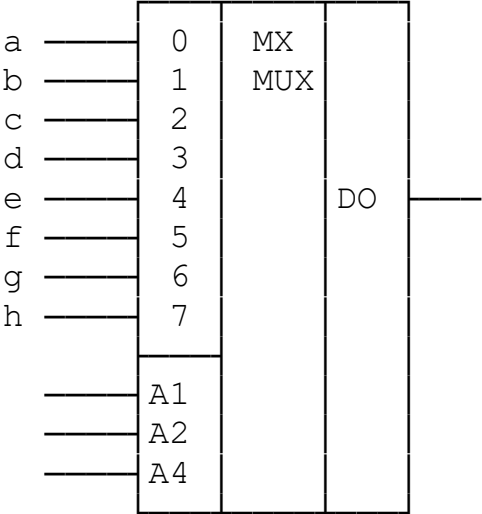
4.2. Мультиплексоры

Мультиплексором называется комбинационная схема, предназна-
n
ченая для коммутации одного из 2 входных сигналов на один



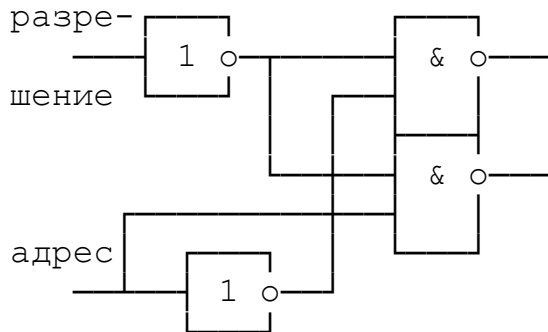
выход. В состав мультиплексора входит n-разрядный адресный селектор, на входы которого подают адресный код, управляющий коммутацией.

A1	A2	A4	a	b	c	d	e	f	g	h	X
0	0	0	a	b	c	d	e	f	g	h	a
1	0	0	a	b	c	d	e	f	g	h	b
0	1	0	a	b	c	d	e	f	g	h	c
1	1	0	a	b	c	d	e	f	g	h	d
0	0	1	a	b	c	d	e	f	g	h	e
1	0	1	a	b	c	d	e	f	g	h	f
0	1	1	a	b	c	d	e	f	g	h	g
1	1	1	a	b	c	d	e	f	g	h	h

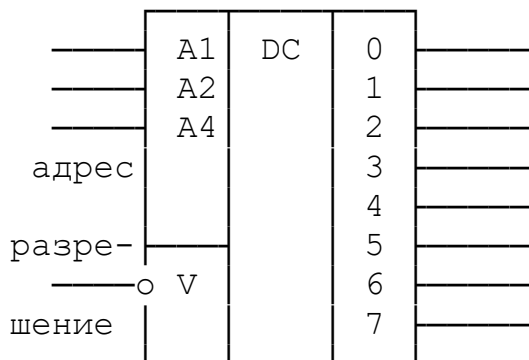


4.3. Дешифраторы

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



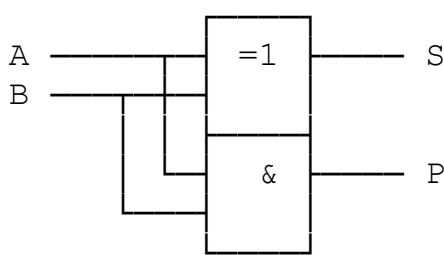
Кроме того, рассмотренный дешифратор может выполнять функцию демультиплексора. При этом вход разрешения используется как информационный.



V	A1	A2	A4	0	1	2	3	4	5	6	7
0	0	0	0	0	1	1	1	1	1	1	1
0	1	0	0	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0
1	X	X	X	1	1	1	1	1	1	1	1

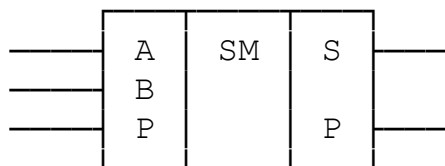
4.4. Сумматоры

Сумматором называется комбинационная схема выполняющая операцию сложения двоичных чисел.



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

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



A _i	B _i	P _{i-1}	S _i	P _i
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Полный одноразрядный сумматор выполняет операцию арифметического сложения двух одноразрядных чисел A_i

и B_i с учетом переноса из младшего разряда P_{i-1}. Он имеет три входа и два выхода для сигнала суммы S_i и сигнала переноса P_i.

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. Возможно каскадирование сумматоров для обработки чисел нужной разрядности.

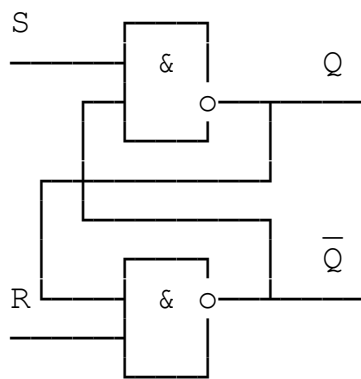
5. ОСНОВНЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ УСТРОЙСТВА.

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

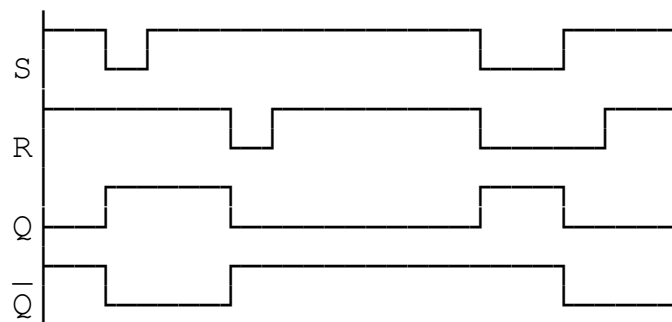
Таким образом, можно говорить о том, что последовательные устройства обладают памятью состояния.

5.1. 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.



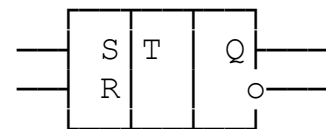
- простейший R-S - триггер



Раздельные входы для установки 0 и 1.

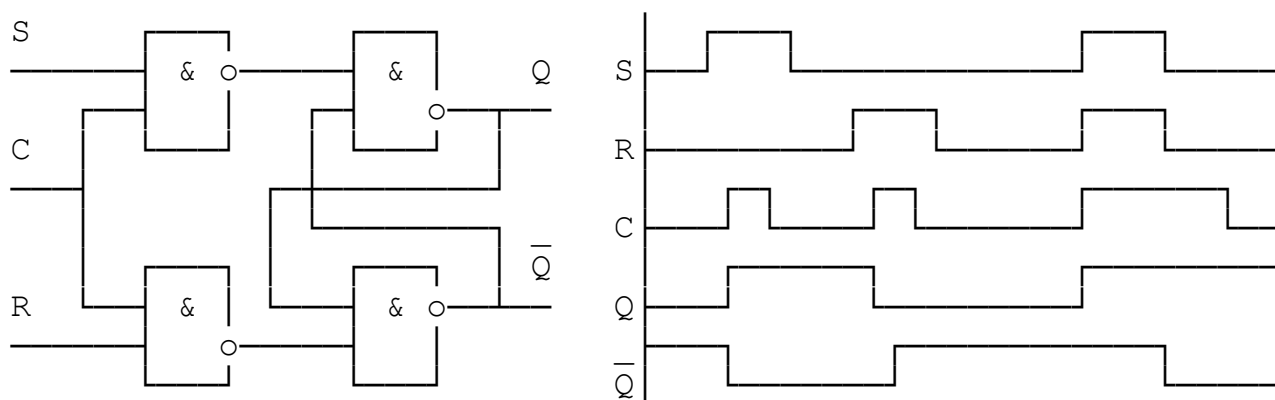
Недостаток - наличие "запрещенного"

состояния $R = 0, S = 0$.

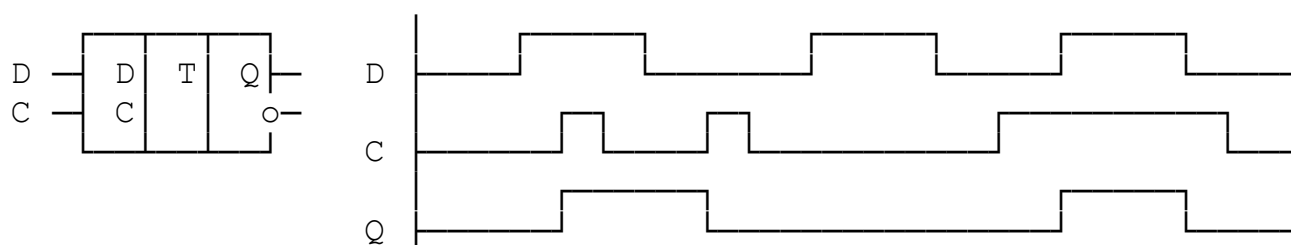


При необходимости согласования процессов установки 0 и 1 с другими процессами в системе используют тактируемые R-S триггеры, содержащие специальный вход для подачи синхроимпульса.

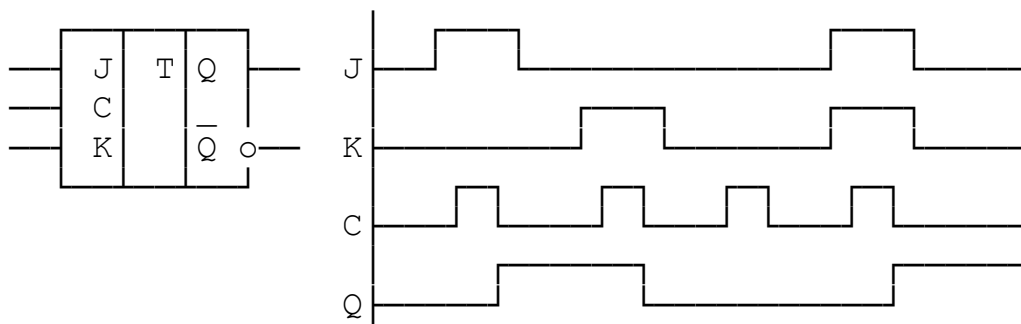
"Запрещенное" состояние $S = R = 1$



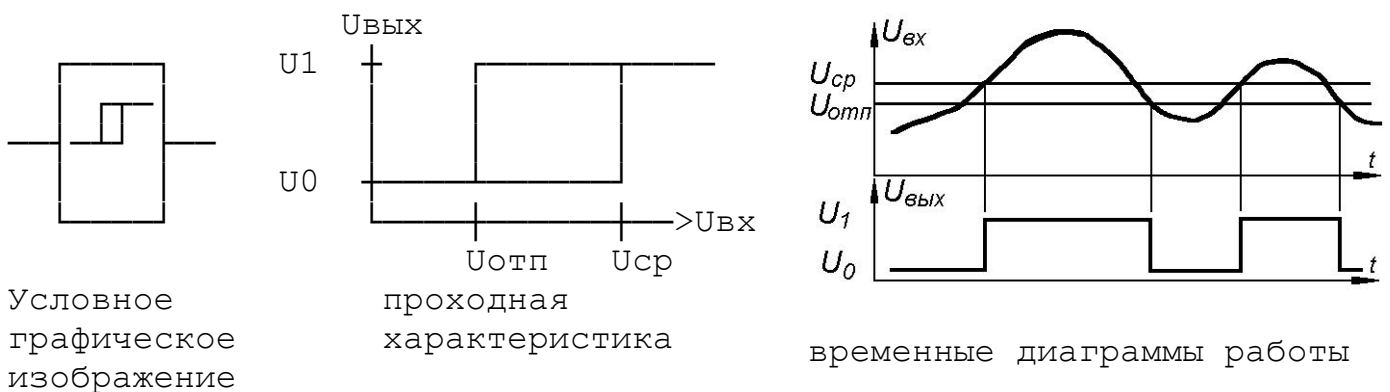
Для сохранения двоичных данных используют триггеры, построенные на основе тактируемых R-S триггеров и имеющие специальный информационный вход D (DATA). Запись информации в таких триггерах происходит только при окончании синхроимпульса ("Защелкивание") информации. Такие устройства получили название статических асинхронных D-триггеров ("Триггер-защелка").



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.



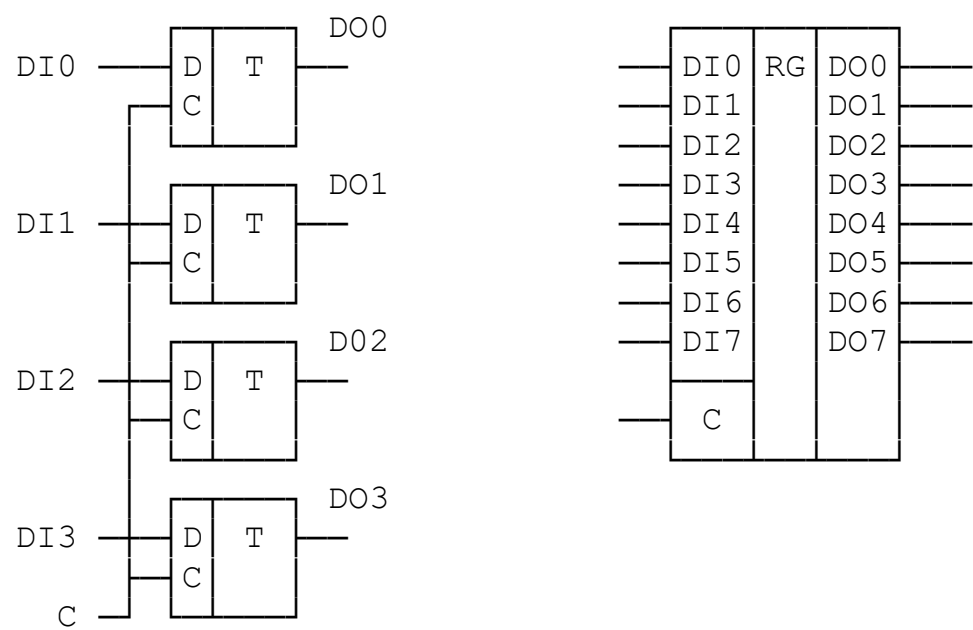
Существенно отличающимися от рассмотренных свойствами обладает триггер Шмитта. У него один вход и один выход. Передаточная характеристика триггера Шмитта имеет гистерезис, что обуславливает наличие двух пороговых уровней срабатывания и отпущения при изменении входного сигнала: при достижении напряжения 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.



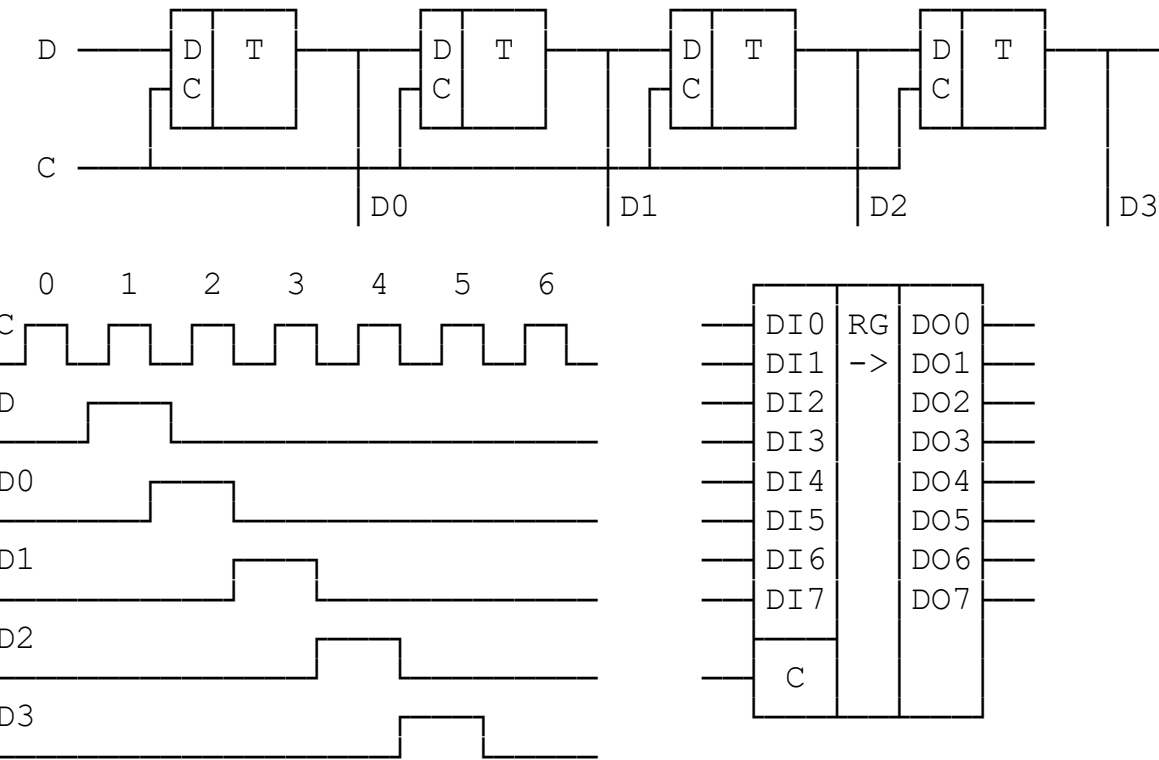
5.3. Регистры

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.



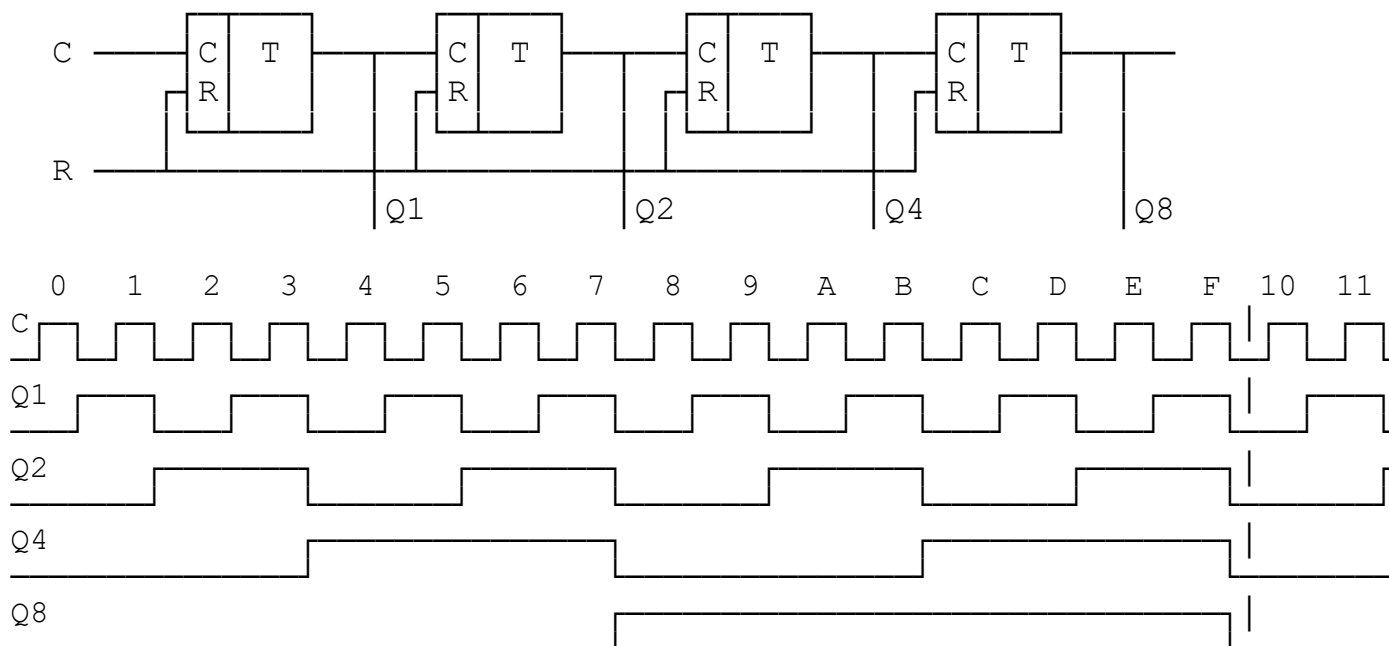
Выпускаются универсальные регистры, способные записывать и представлять информацию как в последовательном, так и в параллельном

виде.

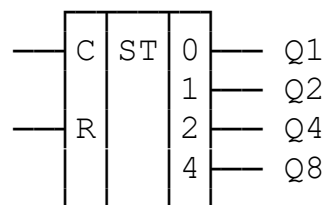
5.3. Счетчики

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 $K_{\text{сч}}=2^n$.



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. Существуют синхронные счетчики с параллельным переносом, у которых код во всех

6. 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 6.1 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

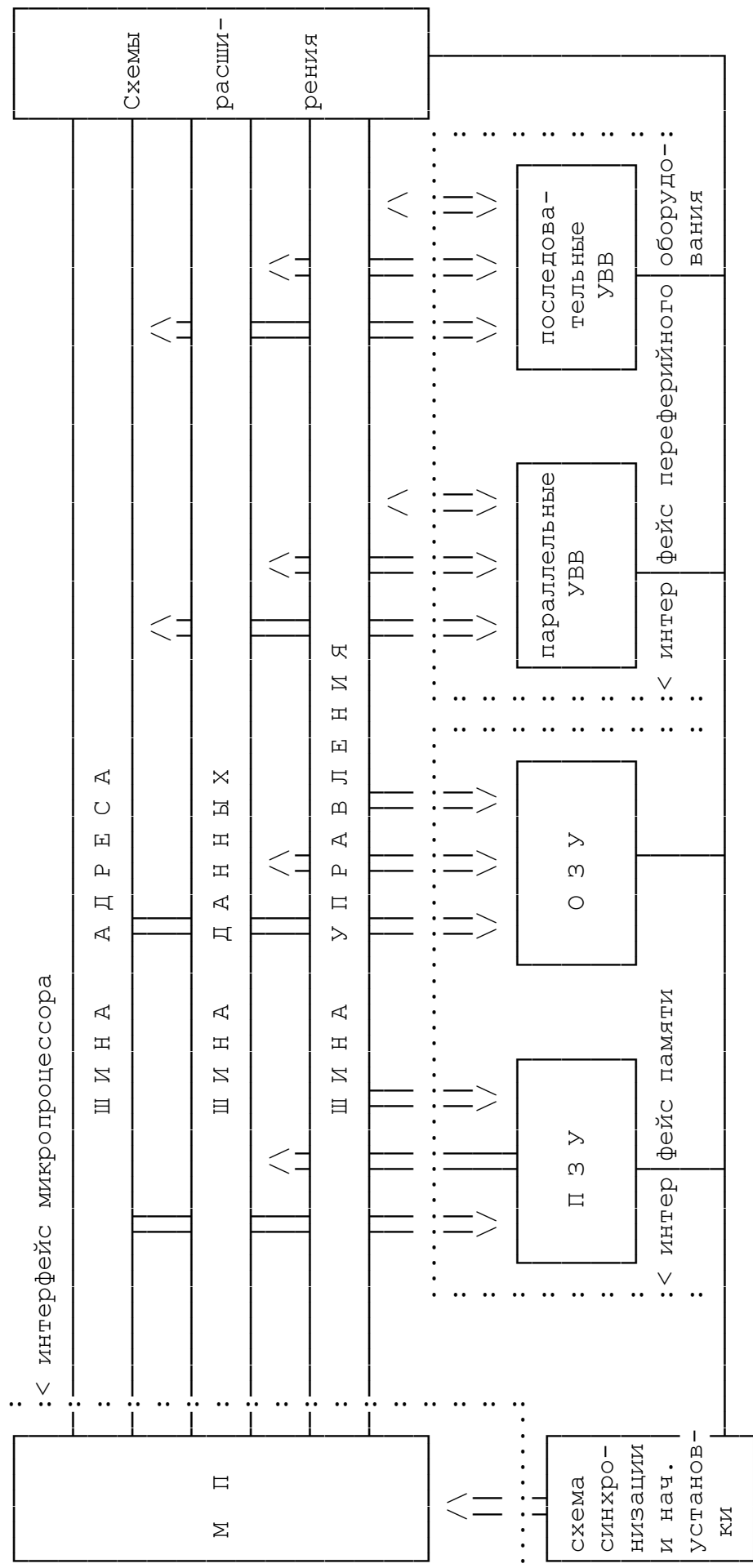


Рис.6.1. Магистрально - модульный принцип построения МП системы.

7. 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:

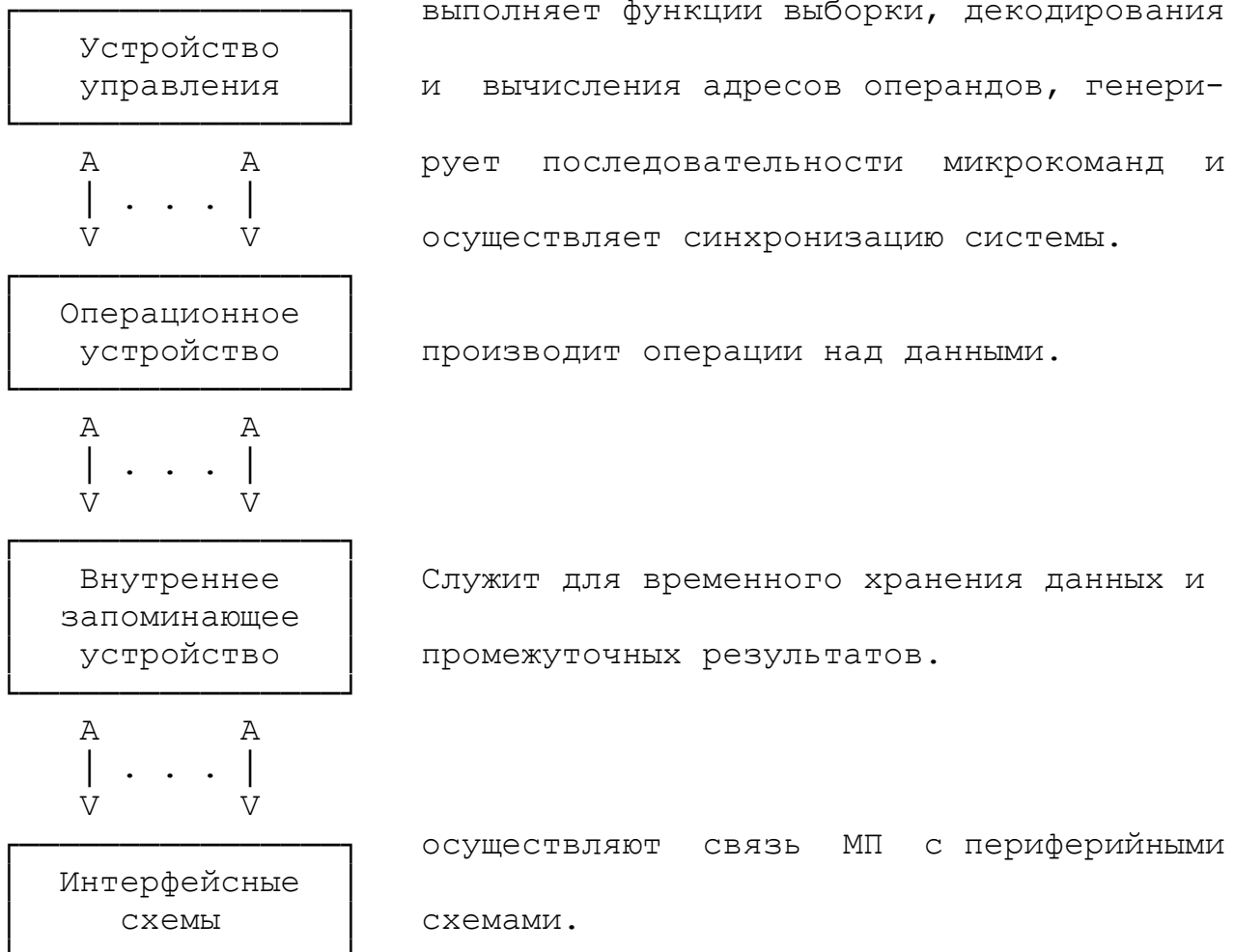


Рис.7.1.

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

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

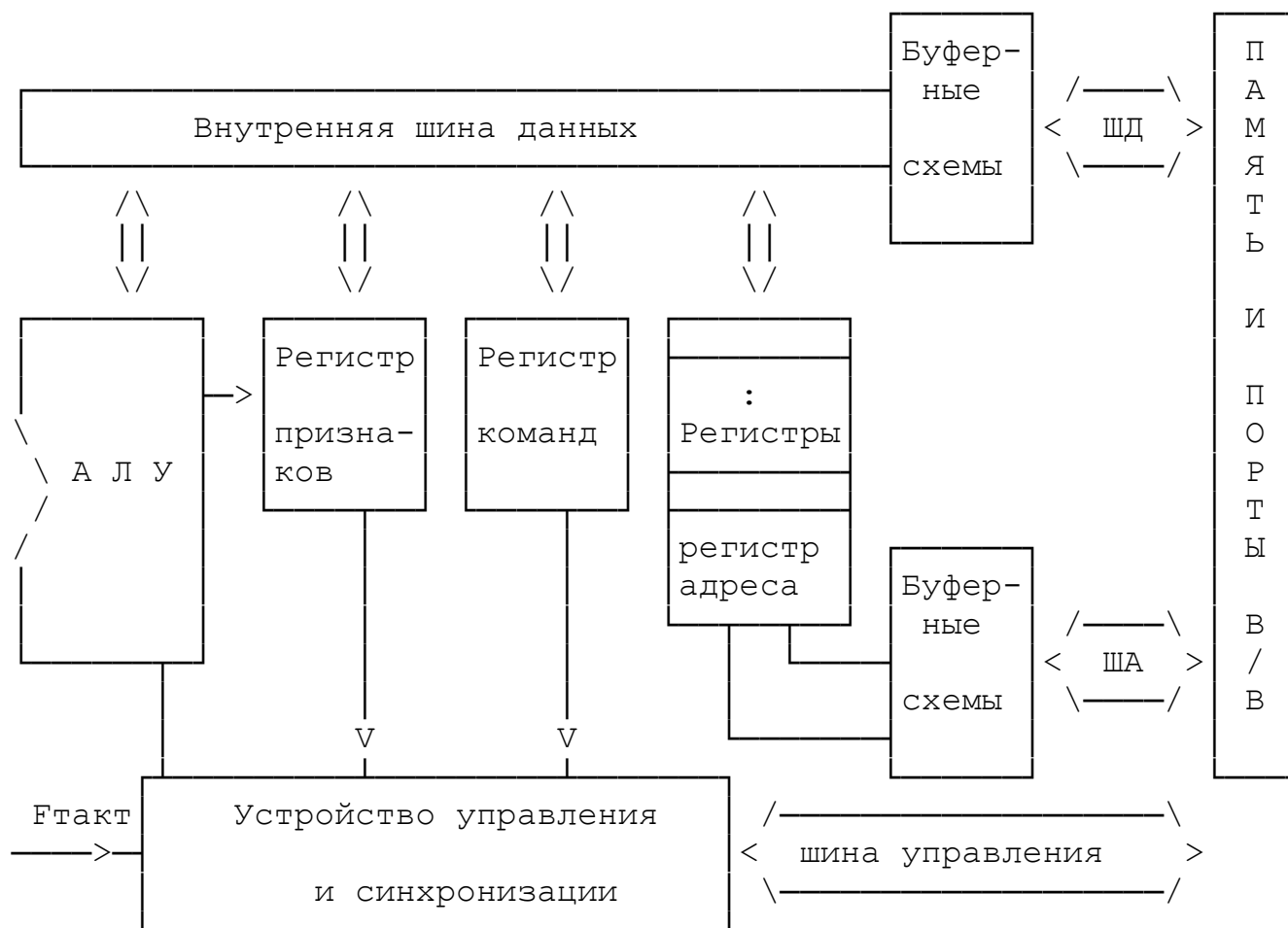


Рис.7.2. Схема МП, построенного по архитектуре Фон-Неймана.

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.

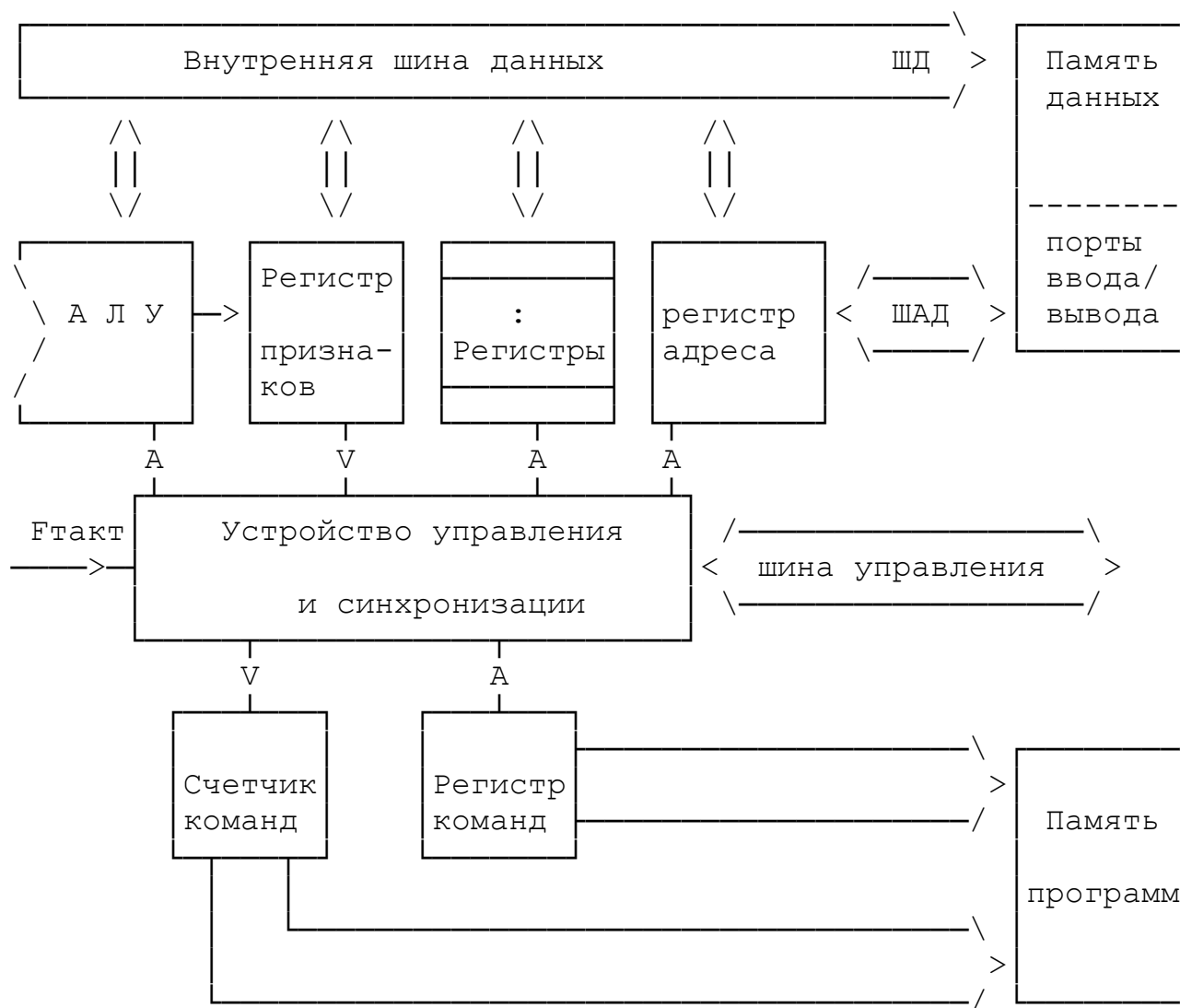


Рис.7.3. Схема МП, построенного по Гарвардской архитектуре.

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 раздельных 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.

КЛАССИФИКАЦИЯ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МП.

Классификация микропроцессоров осуществляется по следующим показателям (см. Приложение 1) :

1.Архитектура а) Фон-Неймана

б) Гарвардская

в) Модифицированная Гарвардская

2.Система и набор команд

а) Полный набор (CISC)

б) Ограниченный (RISC)

3.Тип микроэлектронной технологии.

Отсюда: а) Быстродействие;

б) Энергопотребление;

4.Количество кристаллов

а) Однокристалльные;

б) Многокристалльные;

5.Разрядность слова данных;

6.Емкость адресуемой памяти;

7.Тип управляющего устройства

а) схемное управление – фиксированная система команд

б) микропрограммное управление – изменяемая с/к;

8.Микропроцессоры и однокристалльные ЭВМ;

и др. показатели.

.

8. 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.

8.1. 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.

8.2. 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.

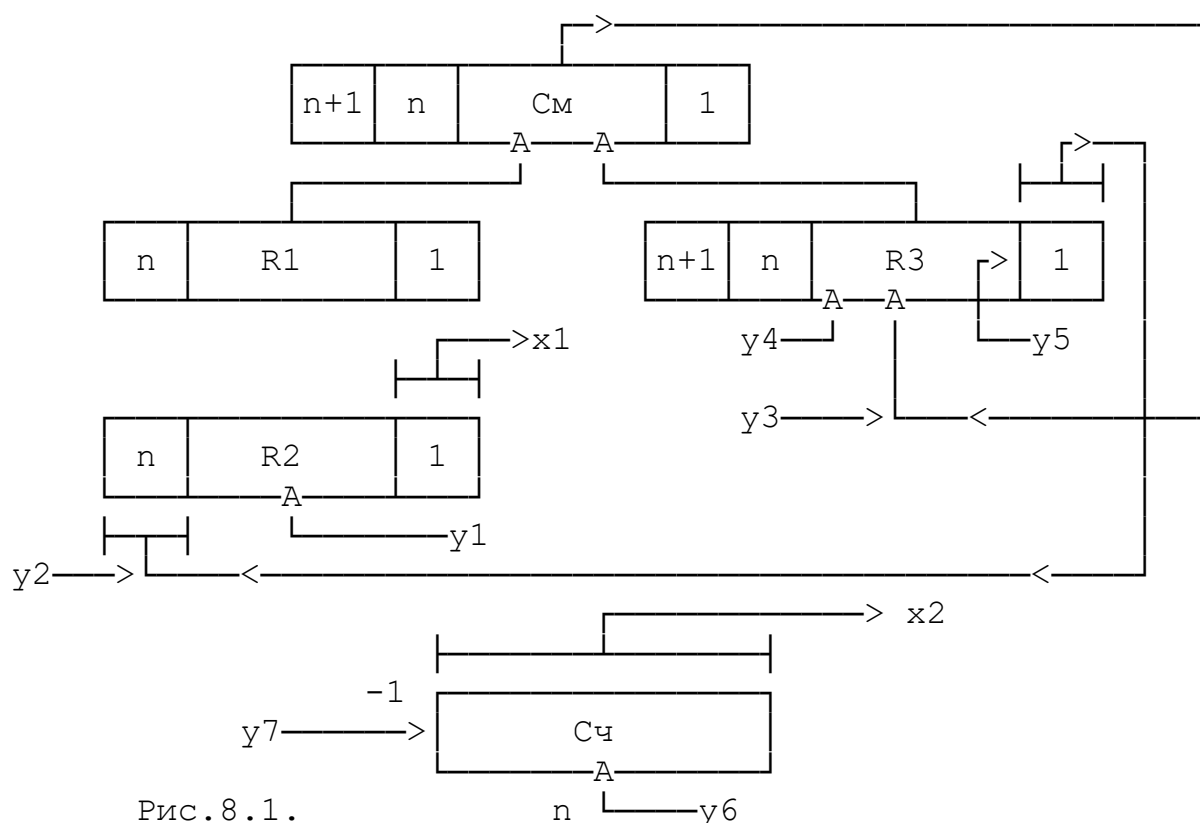


Рис.8.1.

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

8.3. 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) .

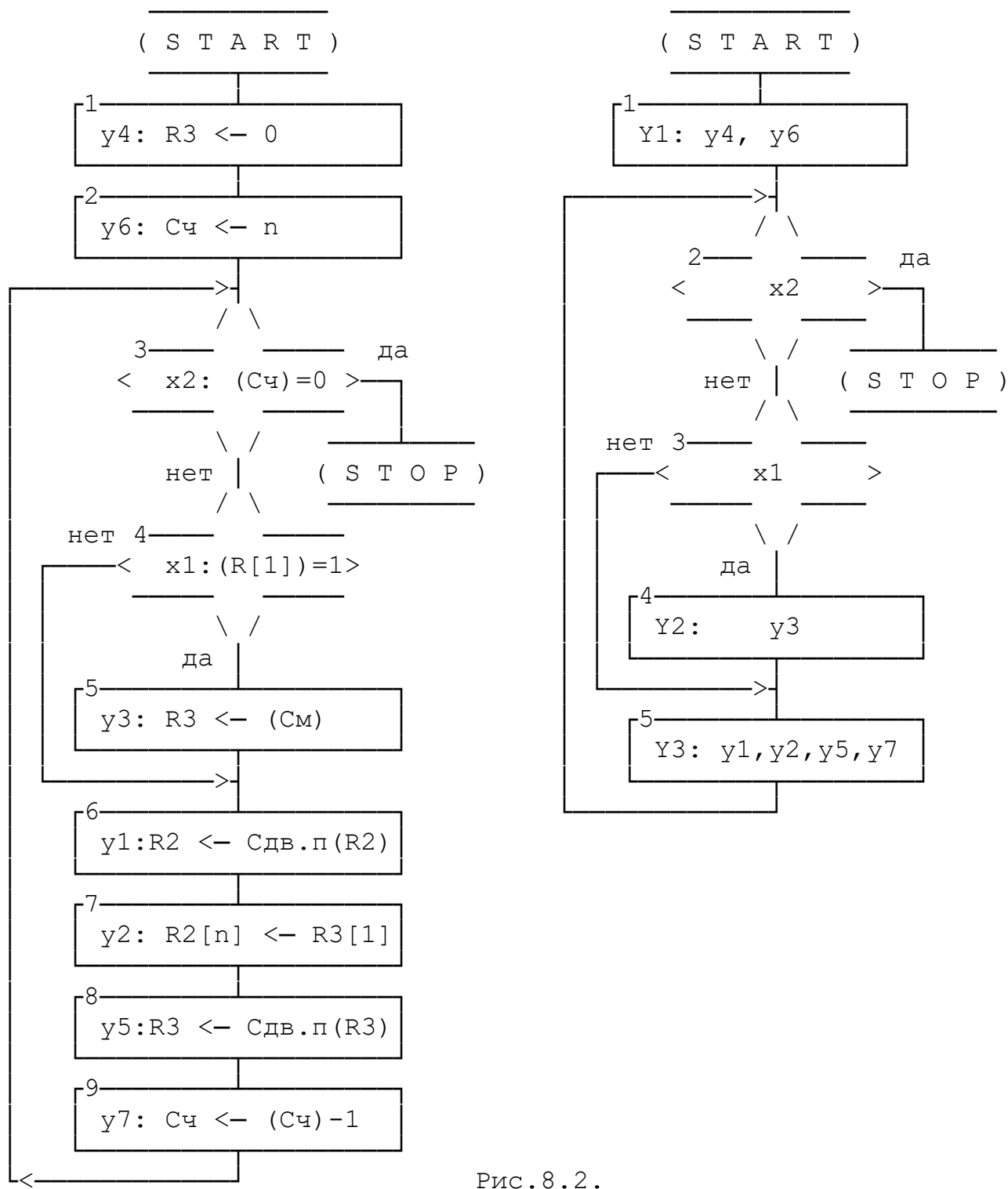


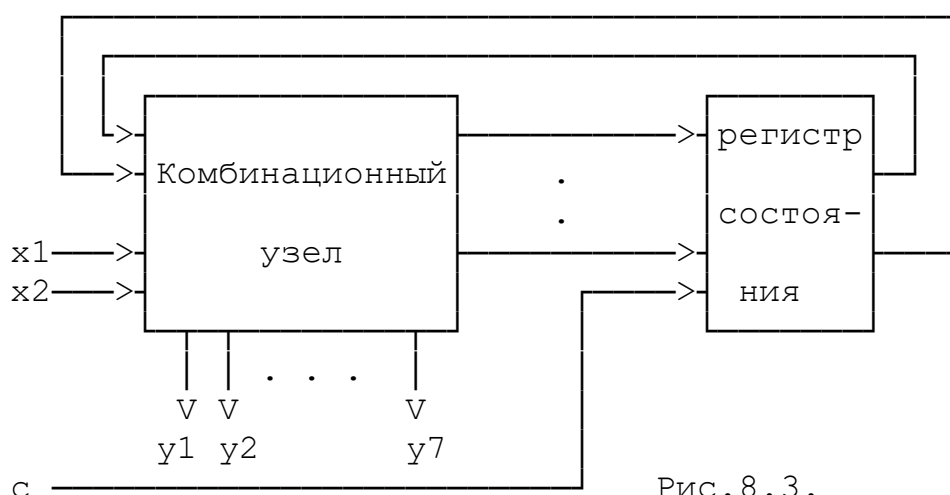
Рис.8.2.

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.



8.4. 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.

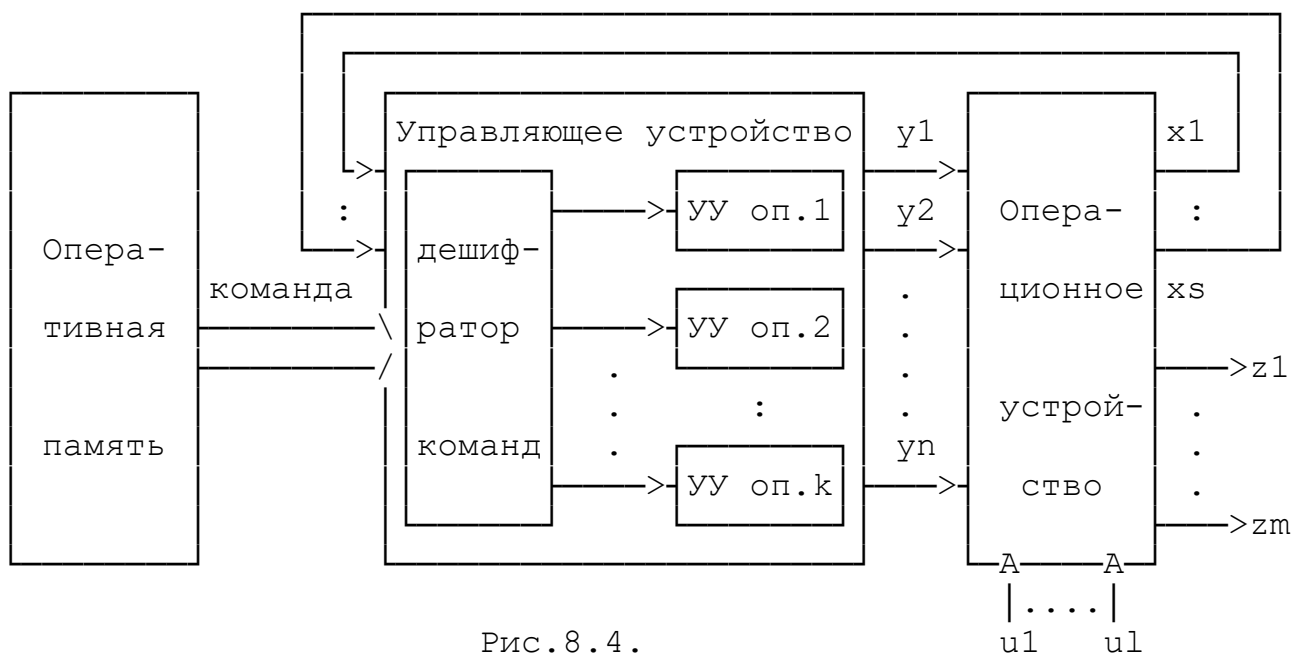


Рис.8.4.

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.

8.5. 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

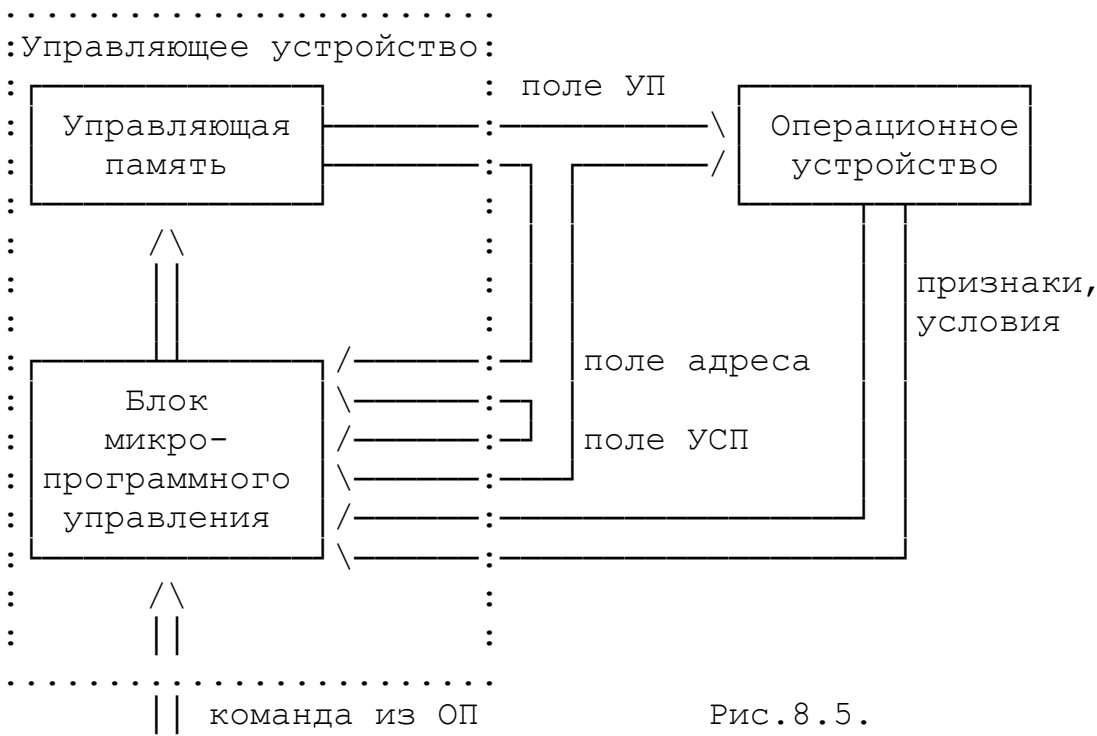
каждом тактовом периоде имеют уровни логических 0 и 1. Таким образом, каждой микрокоманде на выходе управляющего устройства соответствует некоторая кодовая комбинация. Такие кодовые комбинации, называемые микрокомандами, можно хранить в управляющей памяти. Последовательность микрокоманд, предназначенную 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.

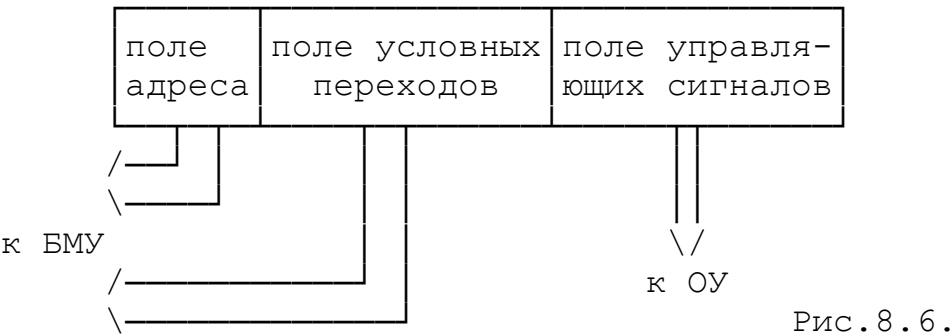
Такой способ реализации операций называется МИКРОПРОГРАММНЫМ способом, а построенное на этом принципе устройство – управляющим устройством с ПРОГРАММИРУЕМОЙ ЛОГИКОЙ.

На рисунке 8 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.



Рассмотрим формат микрокоманды.



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.

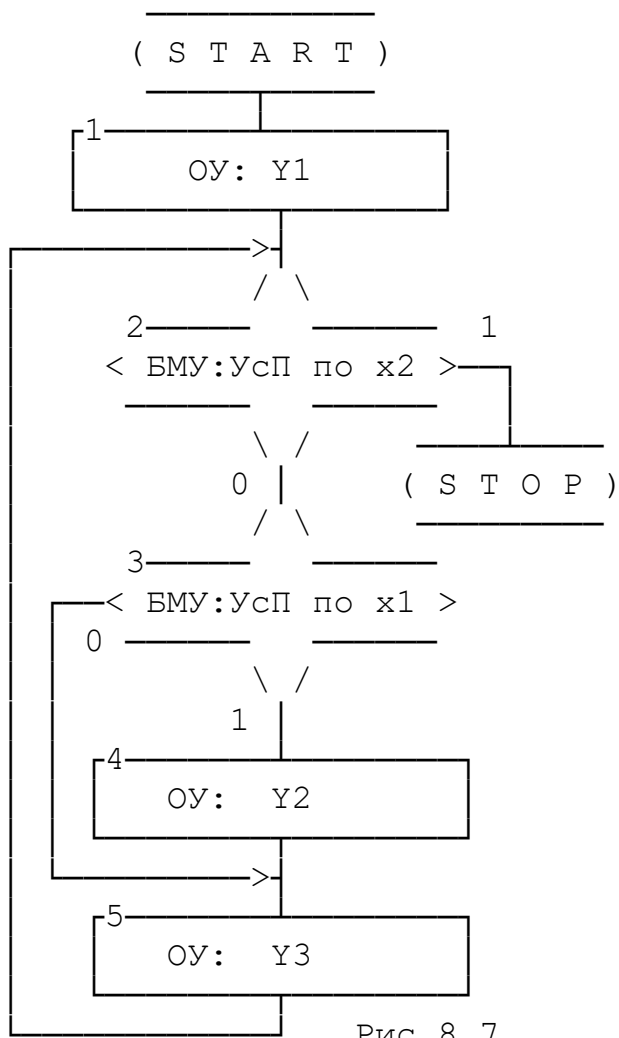


Рис.8.7.

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)
<div> <div>0 0 0 0</div> <div> <div>> 0 0 0 1 <</div> <div> <div>> x2=0 > 0 0 1 0</div> <div> <div>> x2=1 > 0 0 1 1</div> <div> <div>> x1=0 > 0 1 0 0 <</div> <div>> x1=1 > 0 1 0 1</div> </div> </div> </div> </div> </div>	<div>МК1 (МК ОУ:Y1; МК БМУ:БП)</div> <div>МК2 (МК ОУ: -; МК БМУ:УсП по x2)</div> <div>МК3 (МК ОУ: -; МК БМУ:УсП по x1)</div> <div>МК6 (продолжение)</div> <div>МК5 (МК ОУ:Y3; МК БМУ:БП)</div> <div>МК4 (МК ОУ:Y2; МК БМУ:БП)</div>

Микропрограмма приведена в таблице 8.1.

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	0	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

9. ОРГАНИЗАЦИЯ ОДНОКРИСТАЛЬНЫХ 8 - РАЗРЯДНЫХ МИКРОПРОЦЕССОРОВ

9.1. ОБЩИЕ СВЕДЕНИЯ О МП К580

Микросхема КР580ВМ80А представляет собой функционально законченный однокристалльный параллельный 8 - разрядный микропроцессор архитектуры Фон-Неймана, с управляющим устройством, построенным по принципу схемной логики. Изготовлен по n-МОП технологии, содержит около 5 тыс. транзисторов. БИС микропроцессора реализована на кремниевом кристалле размером около 30 мм², заключенном в корпус с 40 выводами. Длина слова микропроцессора - 8 разрядов. Тактовая частота МП до 2 МГц. Уровни напряжения питания +5, -5, +12 В. Применяется в качестве центрального процессора в устройствах обработки данных и управления.

МП К580 имеет отдельные 16 - разрядный канал адреса и 8-разрядный канал данных. Канал адреса обеспечивает прямую адресацию внешней памяти с общей емкостью до 64 Кбайт, 256 устройств ввода и 256 устройств вывода.

8 -разрядное арифметическо-логическое устройство микропроцессора обеспечивает выполнение арифметических и логических операций над двоичными данными, представленными в прямом, дополнительном и двоично - десятичном кодах.

Организация МП К580 отмечена следующими основными особенностями:

- трехшинной структурой с шинами данных, адреса и управления;
- магистральным принципом связей, реализованным в виде связывающей основные узлы МП двунаправленной шиной данных, имеющей разрядность, равную длине слов, обрабатываемых МП;
- наличием регистровой памяти, образованной программно -

доступными общими и специализированными регистрами, а также регистрами временного хранения;

- наличием средств организации стековой памяти ;

- наличием 16 - разрядной шины адреса, обеспечивающей возможность прямой адресации любого байта в памяти емкостью 64 КБайт;

- наличием операций над двухбайтными словами, что обеспечивается специальными командами, тандемными передачами;

- использованием трех форматов команд и разнообразных способов адресации, позволивших при коротком 8 - разрядном слове иметь достаточно гибкую систему команд;

- возможностью реализации векторного многоуровневого приоритетного прерывания путем подключения к МП специальной дополнительной БИС контроллера прерываний;

- возможностью реализации в МП режима прямого доступа к памяти путем подключения специальной дополнительной БИС контроллера прямого доступа;

- наличием эффективных средств работы с подпрограммами и обработки запросов прерываний.

.

9.2. СТРУКТУРА МП К580.

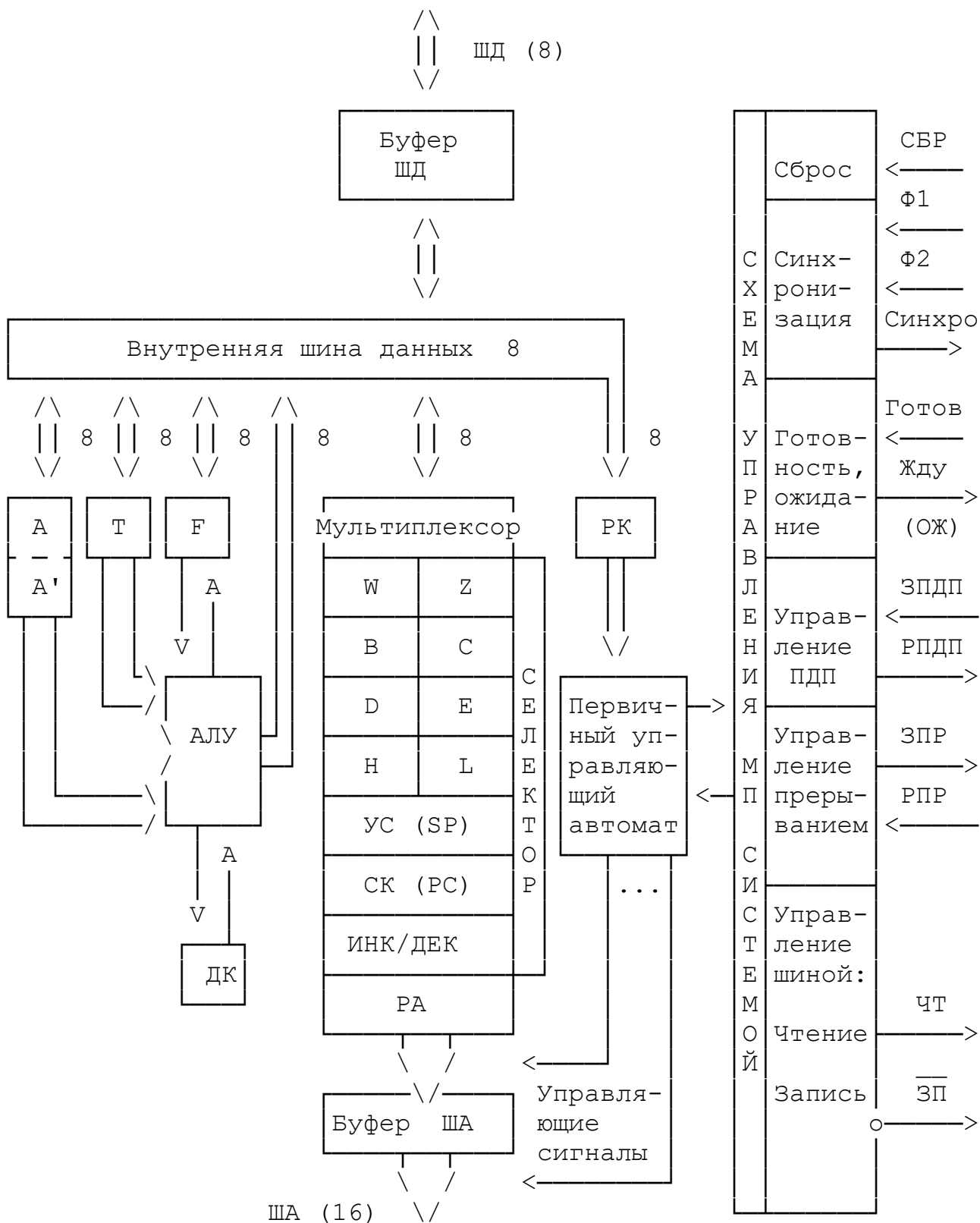


Рисунок 9.1. Структура однокристалльного 8 – разрядного микропроцессора КР580ИК80

МП имеет три шины: 8 - разрядную двунаправленную ВНУТРЕНнюю шину данных (ШД), 16 - разрядную шину адреса (ША) и шину управления (ШУ).

Внутренняя ШД является магистралью, по которой могут

обмениваться данными все подключенные к ней блоки и узлы МП. Таким образом, узлы МП, подсоединенные к ШД разделяют эту шину во времени.

ШУ содержит линии для передачи управляющих сигналов, признаков состояния МП и периферийных устройств, в том числе линии: синхронизации передачи и идентификации информации, передаваемой по ШД; сигналов, информирующих МП о готовности периферийных устройств; сигнала запроса прерывания от периферийных устройств; сигнала разрешения прерывания и др.

При рассмотрении структуры МП можно выделить следующие части: блок регистров, АЛУ, буферные схемы, управляющее устройство.

9.2.1. БЛОК РЕГИСТРОВ

МП К580 содержит программно - доступные 8-разрядные регистры:

- регистр - аккумулятор А;
- регистры общего назначения (РОНы) В, С, D, Е;
- регистр признаков F

и 16-разрядные специализированные регистры:

- счетчик команд СК (PC);
- регистр - указатель стека УС (SP);
- сдвоенный регистр косвенного адреса HL

H - регистр старшего полуадреса

L - регистр младшего полуадреса.

Кроме того, имеются непосредственно недоступные программе регистры:

- 8-разрядные регистры временного хранения T, W, Z;
- 8-разрядный регистр команды;
- 16-разрядный регистр адреса PA.

Имеется возможность использования содержимого пар регистров В и С,

D и E, H и L как составных слов двойной длины.

РОН используются для хранения операндов, промежуточных и конечных результатов, а также адресов и индексов при косвенной и индексной адресациях.

Блок регистров имеет в своем составе регистр - АККУМУЛЯТОР (A). A используется в качестве источника одного из операндов и места, где фиксируется результат операции. В команде A в явном виде не адресуется. На использование A в операции указывает код команды. Иначе говоря, в отношении A применяется подразумеваемая адресация, что позволяет применять одноадресные команды, имеющие сравнительно короткий формат.

Для того, чтобы аккумулятор мог одновременно являться регистром операнда и регистром результата операции, он строится на основе двухступенчатых регистров. (Некоторые МП содержат несколько A).

По отношению к другим программно - доступным регистрам применяется или подразумеваемая или укороченная адресация, задаваемая коротким номером регистра.

Использование A и РОН позволяет при выполнении команд уменьшить число обращений к памяти и тем самым повысить быстродействие.

Наличие в блоке регистров специализированного регистра косвенного адреса HL позволяет иметь команды с подразумеваемой адресацией, т.е. без указания в команде номера регистра, хранящего исполнительный адрес.

Особенностью блока регистров МП является наличие в его составе схемы инкрементатора/декрементатора, которая производит над содержимым регистров (без привлечения АЛУ) операцию

прибавления/вычитания 1. Схема инкрементатора/декрементатора позволяет реализовать процедуры автоматического задания приращений при операциях с адресами не только в регистре – указателе стека, но и в счетчике команд.

При выполнении операций в МП возникает потребность в кратковременном хранении операндов и результатов выполнения операций. Для этой цели служат регистры временного хранения данных T, W и Z. Использование регистров временного хранения позволяет МП за один цикл реализовать, например, такую операцию, как обмен содержимым двух регистров.

Буферные регистры временного хранения построены по простейшему принципу динамического хранения данных на емкостях МОП – транзисторов. Они позволяют простейшими аппаратными средствами выполнять довольно сложные операции перетасовки данных не только между РОН, но и между регистрами, аккумулятором и памятью.

Для повышения эффективности операций со словами двойной длины и операций формирования и пересылок двухбайтных адресов имеется возможность оперировать с содержимым пар регистров В и С, D и E, H и L как с составными словами двойной длины. При этом реализуются так называемые тандемные пересылки, состоящие в передаче в цикле выполнения команды последовательно друг за другом 2 байт информации.

В состав блока регистров входит регистр – защелка адреса памяти РА. Собственно регистр адреса недоступен программисту. Однако любая пара регистров (BC, DE, HL) может быть использована для задания адресов команд и данных в программе. Этот адрес под воздействием соответствующих команд не только может быть загружен в регистр –

защелку, но и модифицирован (посредством схемы инкрементатора/декрементатора) в процессе загрузки. Регистр - защелка адреса передает код адреса в буферную схему и далее в ША.

9.2.2. АРИФМЕТИЧЕСКО - ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ)

8 - разрядная комбинированная схема АЛУ выполняет арифметические и логические операции над 8 - разрядными числами в процессе межрегистровых пересылок. К одному из входов АЛУ всегда подключен А, к другому через регистр Т может быть подключен любой из РОН.

Регистр временного хранения позволяет избежать сбоя и потери информации, когда один из регистров используется в качестве регистра - операнда и регистра - результата.

АЛУ непосредственно связано с регистром признаков F, в соответствующих разрядах которого фиксируются особенности выполнения каждой операции:

- нулевой результат в аккумуляторе - Z,
- перенос и старшего разряда - CY,
- знак результата - S,
- паритет - P,
- вспомогательный перенос из младшего полубайта - AC.

Наличие в МП регистра признаков упрощает осуществление программных переходов в зависимости от состояния триггеров признаков. АЛУ позволяет в процессе межрегистровых "пересылок с перекосом" выполнять операции сдвига на один разряд вправо или влево. Многократный сдвиг реализуется последовательностью одноразрядных сдвигов.

В состав АЛУ входит комбинационная схема десятичного корректора ДК, назначение которого состоит в том, чтобы под воздействием

специальной команды интерпретировать результат выполнения двоичной операции как результат десятичной арифметики.

АЛУ реализует простейшие арифметические и логические операции (сложение, вычитание, сдвиги, сравнение, логическое умножение и другие операции). Все более сложные операции (умножение, деление, вычисление элементарных функций и др.) выполняются по подпрограммам.

9.2.3. СТЕКОВАЯ ПАМЯТЬ

В МП К580 организуется стековая память, реализующая безадресное задание операндов. В общем случае стек представляет собой группу последовательно пронумерованных ячеек памяти, снабженных указателем стека, в котором автоматически при записи и считывании устанавливается адрес последней занятой ячейки стека - вершины стека. При операции занесения в стек слово помещается в следующую по порядку свободную ячейку стека, а при считывании из стека извлекается последнее поступившее в него слово. Таким образом в стеке реализуется принцип обслуживания "последний пришел - первый ушел". Этот порядок при обращении к стеку реализуется автоматически. Поэтому при операциях со стеком возможно безадресное задание операнда - команда не содержит адреса ячейки стека, но содержит адрес ячейки памяти или регистра, откуда слово передается в стек или куда загружается из стека.

Таким образом, стек является устройством памяти с последовательным доступом. Невозможно считать информацию, не считав стоящую перед ней.

В рассматриваемом МП используется "перевернутый" стек, т.е. при передаче в стек слова значение УС (адрес вершины стека) уменьшается,

а при извлечении – увеличивается.

Непосредственно в МП К580 из оборудования стековой памяти содержится только регистр – указатель стека и соответствующие цепи управления. Сам стек реализуется в виде группы последовательных ячеек оперативной памяти.

Стековая адресация широко используется при работе с подпрограммами и в процедурах обработки прерывания.

9.2.4. БУФЕРНЫЕ СХЕМЫ

Двунаправленный буфер шины данных предназначен для логического и электрического разделения внутрипроцессорной шины данных и внешней, системной ШД. Буфер состоит из регистра – защелки и выходной схемы с тремя состояниями, т.е. схемы, обеспечивающей на выходе состояния 0, 1 и полное электрическое отключение от нагрузки (высокоимпедансное или "третье" состояние). Наличие в МП буферных схем, отключающихся от общей системной шины, обеспечивает реализацию магистрального принципа межмодульных связей в МП – системе.

В режиме ввода информации внутренняя шина данных подсоединяется к регистру – защелке, загрузку которого из внешней шины производит буферная схема под управлением команды.

В режиме вывода информации буферная схема передает во внешнюю шину данных содержимое буферного регистра – защелки, на вход которого по внутренней шине данных с одного из регистров загружен код, подлежащий выдаче.

Во время выполнения операций в МП, не связанных с процедурами обмена с внешними по отношению к МП устройствами, буферная схема отключается от ШД, т.е. переходит в высокоимпедансное состояние.

Буферная схема шины адреса (ША) – однонаправленная, обеспечивает передачу адресов команд и данных, а также номеров периферийных устройств от МП в систему. Выход буферной схемы ША, точно так же как и буфера данных может переходить в третье состояние. Подобный режим необходимо иметь в системе, в которой к памяти могут обращаться по системной шине не только МП, но и другие периферийные устройства (контроллер ПДП, пульт оператора, контроллеры накопителей и др.)

9.3. УПРАВЛЯЮЩЕЕ УСТРОЙСТВО МП.

Управляющее устройство МП состоит из двух относительно независимых частей:

1. первичного автомата, управляющего процессами внутри МП;
2. схемы, обрабатывающей внешние сигналы и генерирующей управляющие сигналы МП – системе.

9.3.1. АЛГОРИТМ УПРАВЛЯЮЩЕГО АВТОМАТА МП.

Рассмотрим примерную схему алгоритма функционирования управляющего автомата в течение рабочего цикла выполнения команды (рис.9.3). Выполнение рабочего цикла команды начинается с опроса триггера прерывания. Если запрос прерывания имеет место, то автомат передает управление специальной подпрограмме обработки прерывания и выполняется соответствующая программа обслуживания данного запроса прерывания. При отсутствии прерывания управляющий автомат генерирует управляющие сигналы выборки команды из памяти и формирование в СК адреса следующей команды. Затем управляющий автомат дешифрирует код операции в команде и генерирует соответствующую коду операции серию управляющих сигналов, обеспечивающих реализацию в МП заданной операции.

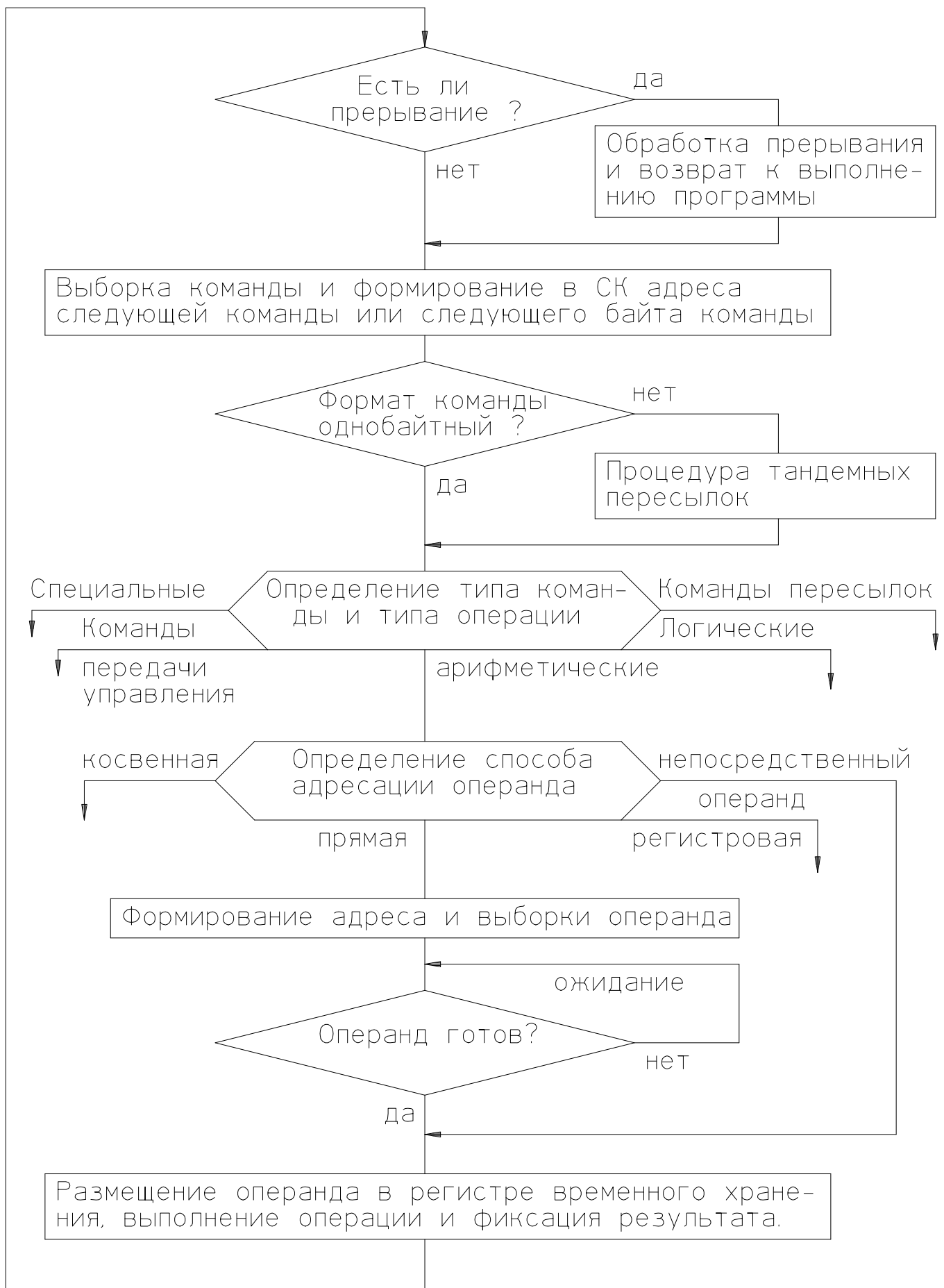


Рисунок 9.2 Алгоритм управляющего автомата МП

Существенная особенность работы первичного автомата состоит в том, что его алгоритм содержит условный оператор ожидания готовности операнда. Появление такого оператора в алгоритме объясняется тем, что МП приспособлены для работы с различными типами внешней памяти (с непосредственным или последовательным доступом), имеющими разные времена обращения. Кроме того, МП может обращаться за операндом не только к памяти, но и к медленно действующим УВВ. Наличие в схеме алгоритма первичного автомата оператора ожидания готовности операнда является одной из причин того, что последовательность управляющих сигналов, реализующая некоторую команду генерируется первичным автоматом не только на основе кода команды, но и под воздействием набора внешних управляющих сигналов.

Схема управления системой в зависимости от кода текущей команды, состояния первичного управляющего автомата, а также от значений внешних сигналов в шине управления вырабатывает управляющие сигналы, которые реализуют процедуры системного обмена информацией.

9.3.2. ФОРМИРОВАНИЕ И ОБРАБОТКА ВНЕШНИХ СИГНАЛОВ.

В режиме временного мультиплексирования шины данных МП выдает информацию о своем внутреннем состоянии в начале каждого МЦ. Для согласованной работы МП и периферийного оборудования этого недостаточно. Поэтому в МП для управления системой используется комбинация прямых управляющих линий в шине управления собственно МП, которая подсоединяется к внешним схемам. Сигналы на этих управляющих линиях (РПР, ЧТ, ЗП, ЖДУ) изменяются на выходе МП в каждом такте. т.е. они отражают переходы первичного автомата от машинного такта T_i к такту T_{i+1} . Такая комбинация: слово состояния МП, передаваемого в

регистр состояния один раз за МЦ, и набор управляющих сигналов, которые изменяются в каждом такте, позволяют организовать шину управления МП – системы.

Временная диаграмма определяет основной цикл команды МП в условиях, когда присутствует внешний управляющий сигнал ГОТОВ, информирующий о готовности периферийного оборудования к обмену с МП (рисунок 9.3). В первом такте синхронизации T1 МП выставляет на шине адреса адрес очередной команды A0 – A15. Начинается цикл выборки команды. Одновременно на линии синхронизации СИНХРО появляется единичный сигнал, который, во-первых, идентифицирует информацию на шине данных D0 – D7 как слово состояния процессора (ССП) и загружает его во внешний регистр состояния, а во вторых свидетельствует о начале машинного цикла ВЫБОРКА.

По окончании сигнала синхронизации буферная схема ШД переводит ШД в режим ввода, о чем свидетельствует сигнал на линии ЧТ ШУ. В такте T2 МП осуществляет проверку готовности внешнего устройства к обмену в том случае, если адаптер внешнего устройства или память генерирует сигнал ГОТОВ. Первичный автомат переходит в состояние ожидания. В этом состоянии МП будет находиться до тех пор, пока на линии управления ГОТОВ не появится единичный сигнал, который будет свидетельствовать, что память или периферийное оборудование готовы к обмену. Пока первичный автомат находится в состоянии ожидания Tw МП простаивает. Из состояний T2 или Tw первичный автомат переходит в состояние T3, в котором производится чтение или запись слова в память. Состояния T4 и T5 отводятся для реализации операции, заданной кодом команды. Выполнение некоторых сложных команд может потребовать от

первичного автомата неоднократного прохода по циклу состояний от T2 до T5.

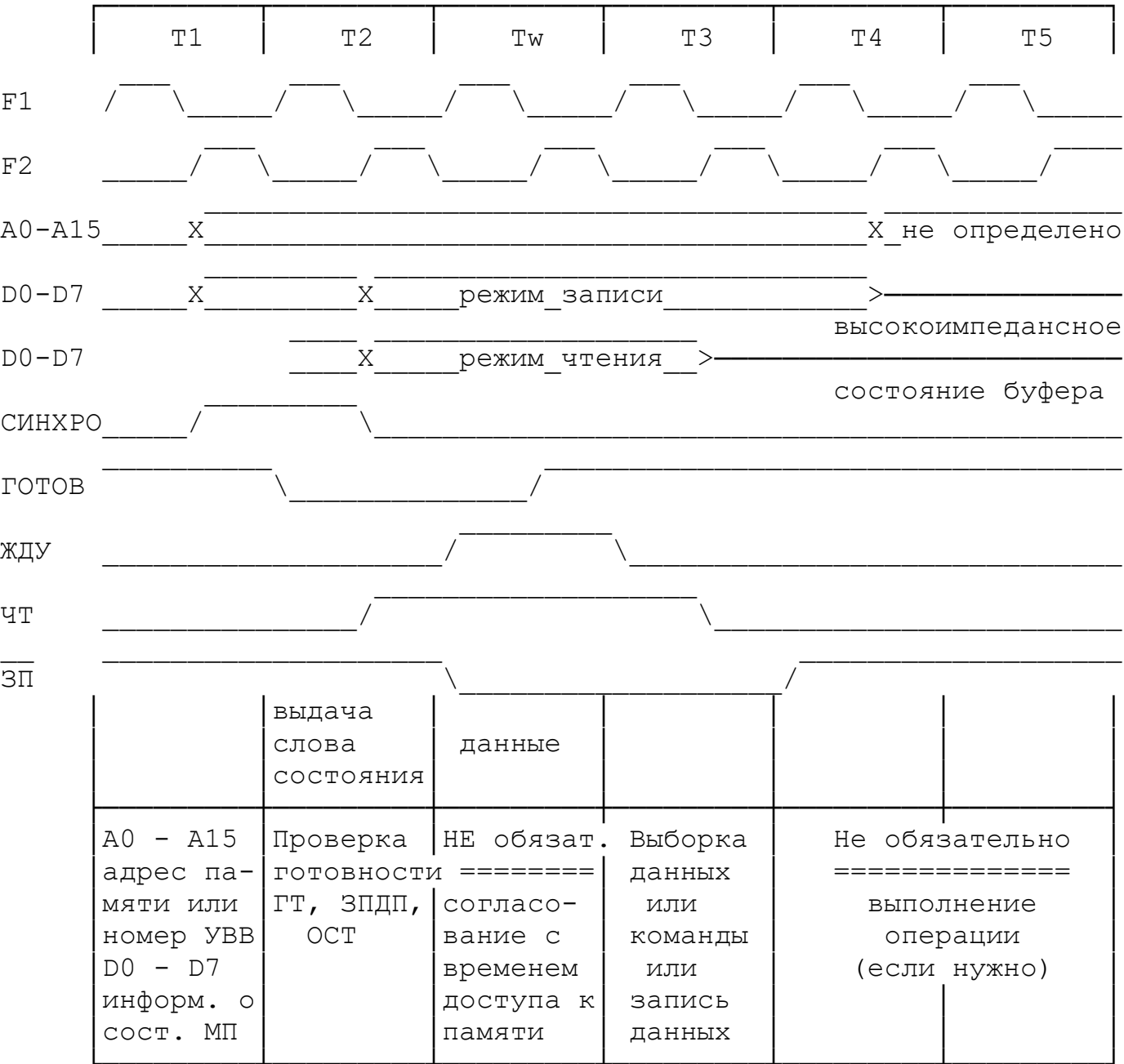


Рисунок 9.3

9.3.3. ТАКТИРОВАНИЕ МП И СИНХРОНИЗАЦИЯ МП - СИСТЕМЫ.

В МП управляющий автомат в зависимости от сложности команды реализует ее цикл за несколько внутренних машинных циклов. Цикл команды в МП реализуется за 1 - 5 машинных циклов. Один машинный цикл требуется МП для одного обращения к памяти или УВВ. Выборка байта команды, байта адреса или данных требует одного машинного цикла.

Аналогичность операций, выполняемых в этих циклах, несмотря на то что они расположены в различных фрагментах схемы алгоритма работы управляющего устройства, позволяет их реализовать в течение цикла команды на одном и том же оборудовании первичного автомата, что приводит к усложнению управления МП – системой.

Эффективность работы управляющего автомата достигается за счет того, что машинные циклы (МЦ) могут быть переменной длины. Так в МП К580 каждый МЦ может состоять из 3 – 5 тактов.

Рассмотрим временные диаграммы тактирования МП от внешнего генератора.

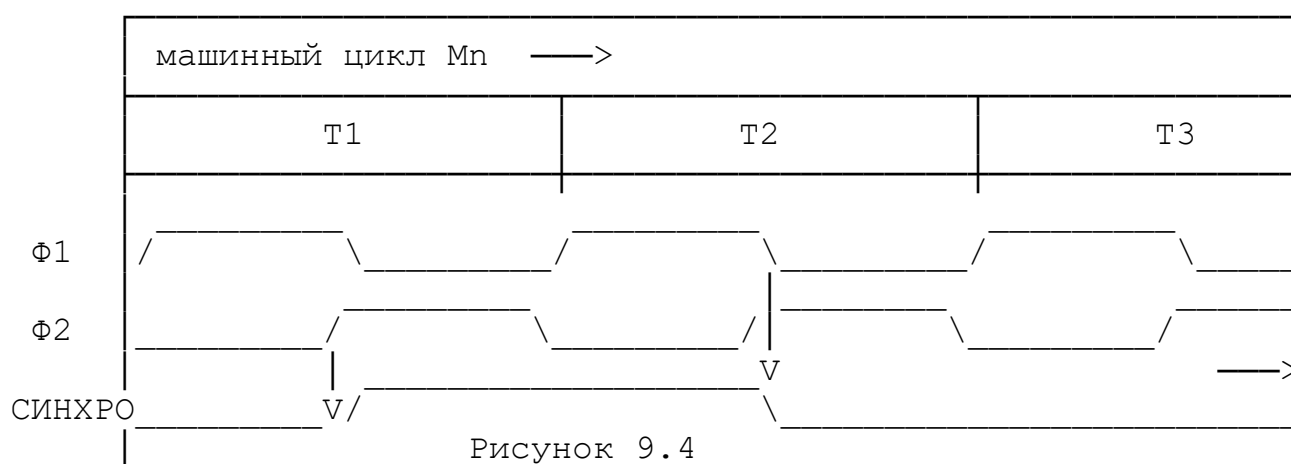


Рисунок 9.4

Сигнал "СИНХРО" идентифицируется МП – системой как начало машинного цикла.

Каждый такт машинного цикла образует пара сигналов тактирования Φ_1 и Φ_2 , поступающих от внешнего генератора. В начале каждого машинного цикла первичный автомат генерирует сигнал синхронизации МП – системе "СИНХРО". Каждому такту T соответствует отдельное состояние первичного автомата управляющего устройства МП. Все такты имеют одинаковую длительность. Существуют три исключения из этого состояния:

- Состояние "ЖДУ", в котором МП находится в состоянии ожидания операнда;

- Состояние "ПДП", в которое МП переходит под воздействием внешних сигналов управления МП - системой;
- Состояние "ОСТАНОВ", в которое МП может быть введен посредством команды останова.

Эти три состояния МП не связаны с тактовой частотой сигналов $\Phi 1$ и $\Phi 2$ и их продолжительность неопределенная, так как зависит от внешних по отношению к МП событий. Эти состояния длятся целое число тактов и выход МП из этих состояний тактируется.

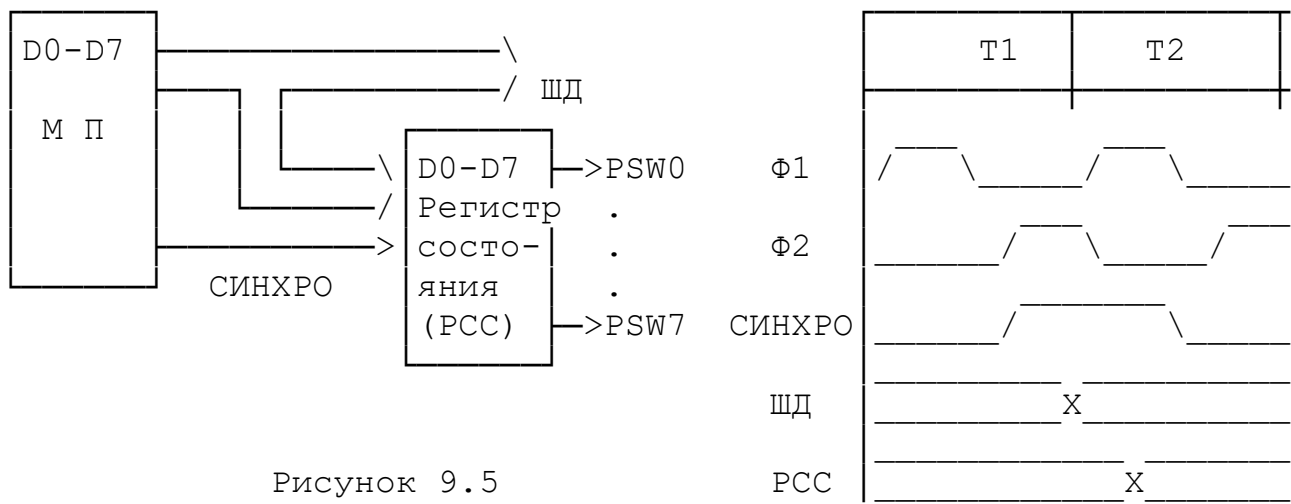
Из состояния ОСТАНОВ МП может быть выведен двумя способами: или поступившим внешним сигналом прерывания и соответствующим разрешением на него ЗПР - РПР, или сигналом системной установки в исходное состояние СБРОС, который переводит первичный автомат в начальной состоянии.

Таким образом, каждый период тактирования МП К580 соответствует особому состоянию первичного автомата. В стандартном МЦ может быть от трех до пяти состояний автомата. В зависимости от сложности операций, определяемых командой, цикл команды может выполняться от четырех до восемнадцати тактов.

9.3.4. СЛОВО - СОСТОЯНИЯ МП.

Для нормального функционирования МП - системы недостаточно управляющих сигналов, генерируемых МП. МП - система в каждом МЦ должна получать более полную информацию о состоянии МП. В условиях "узкого" интерфейса МП, когда внешних выводов для индикации внутреннего состояния (слова состояния) МП недостаточно, эта задача решается с использованием мультиплексирования шины данных и представления внутреннего состояния МП на внешнем по отношению к МП

регистре слова состояния РСС.



Процессор в первом такте каждого МЦ генерирует на шине данных слово состояния СС (PSW), которое содержит информацию о процессах, происходящих в МП. Так как сигнал синхронизации СИНХРО вырабатывается в начале каждого МЦ, то он используется в качестве сигнала, идентифицирующего информацию, представленную на шине данных как СС.

СС загружается во внешний регистр РСС под воздействием сигнала синхронизации СИНХРО. Сигналы, которые представлены на выходе 8 - разрядного регистра состояния, используются в качестве сигналов управления периферией МП - системы. Всего в МП К580 десять типов машинного цикла и соответственно десять кодов СС, идентифицирующих цикл. Каждый разряд слова состояния заводится на соответствующие управляющие входы адаптеров или схем сопряжения с УВВ, определяя тем самым их режим функционирования в соответствии с данным текущим состоянием МП.

Раз-	Сигнал	Вы-	Чте-	Зап-	Чте-	Зап-	Чте-	Зап-	Пре-	Ос-	Преры-
------	--------	-----	------	------	------	------	------	------	------	-----	--------

ряд	состояния	борка команды	ние из памяти	ись в память	ние из стека	ись в стек	ние из УВВ	ись в УВВ	ры- ва- ние	та- нов	вание при оста- нове
PSW0	Подтв. прерыв.	0	0	0	0	0	0	0	1	0	1
PSW1	Запись Вывод	1	1	0	1	0	1	0	1	1	1
PSW2	Стек	0	0	0	1	1	0	0	0	0	0
PSW3	Подтв. остан.	0	0	0	0	0	0	0	0	1	1
PSW4	Вывод	0	0	0	0	0	0	1	0	0	0
PSW5	M1	1	0	0	0	0	0	0	0	0	1
PSW6	Ввод	0	0	0	0	0	1	0	0	0	0
PSW7	Чтение	1	1	0	1	0	0	0	0	1	0

Таким образом в условиях "узкого" интерфейса МП, управление МП – системой осуществляется генерацией управляющих воздействий на двух уровнях:

1. На уровне управляющих сигналов по шине управления собственно МП в каждом такте работы первичного автомата;

2. На уровне сигналов слова состояния в каждом МЦ.

Выходы регистра слова состояния и управляющие линии корпуса МП образуют ШУ МП – системы со сложным, многофункциональным периферийным оборудованием. При этом использование временного мультиплексирования шины данных для вывода на внешний регистр состояния сигналов управления МП – системой снижает общую производительность системы.

9.3.5. РАБОТА УПРАВЛЯЮЩЕГО УСТРОЙСТВА В РЕЖИМЕ ПРЕРЫВАНИЯ

Периферийное оборудование МП – системы может запросить прерывание текущей программы у МП путем подачи сигнала ЗПР на вход прерывания. Сигнал прерывания может возникнуть в любой момент цикла команды. Обработка прерываний организована таким образом, что запрос прерывания фиксируется во внутреннем триггере запроса прерывания

только при переходе первичного автомата к циклу M1, т.е. к начальному циклу очередной команды, и только в случае, если программным путем было разрешено прерывание, т.е. внутренний триггер разрешения прерывания РПР находится с состоянием "1". Выполнение этих условий приведет к тому, что следующий МЦ M1 будет циклом обработки запроса прерывания. Машинный цикл прерывания, который начинается в такте T1 в условиях разрешенного прерывания, в основных чертах повторяет МЦ ВЫБОРКА.

9.3.6. РАБОТА УПРАВЛЯЮЩЕГО УСТРОЙСТВА В РЕЖИМЕ ПРЯМОГО ДОСТУПА К ПАМЯТИ.

Управляющее устройство МП позволяет выполнять операции с прямым доступом к памяти (ПДП). При этом обмен данными осуществляется без участия процессора. Режим ПДП наиболее целесообразен при обмене блоками данных. В этом случае обмен данными между оперативной памятью и периферийным устройством производится быстрее, чем при программно - управляемом обмене. Выставляя на соответствующей линии запрос ПДП, периферийное устройство приостанавливает вычислительный процесс в МП и осуществляет управление по шинам адреса и данных. Первичный автомат при этом вырабатывает управляющие сигналы, переводящие буферные схемы шин адреса и данных МП в высокоимпедансное состояние. Таким образом реализуется принцип захвата шины МП - системой для ввода - вывода данных. Т.к. в режиме ПДП обмен данными происходит без участия МП, содержимое всех регистров МП остается неизменным.

9.4. СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА К580.

Для лучшего понимания особенностей системы команд МП воспользуемся программистской моделью, соответствующей упрощенной структуре МП.

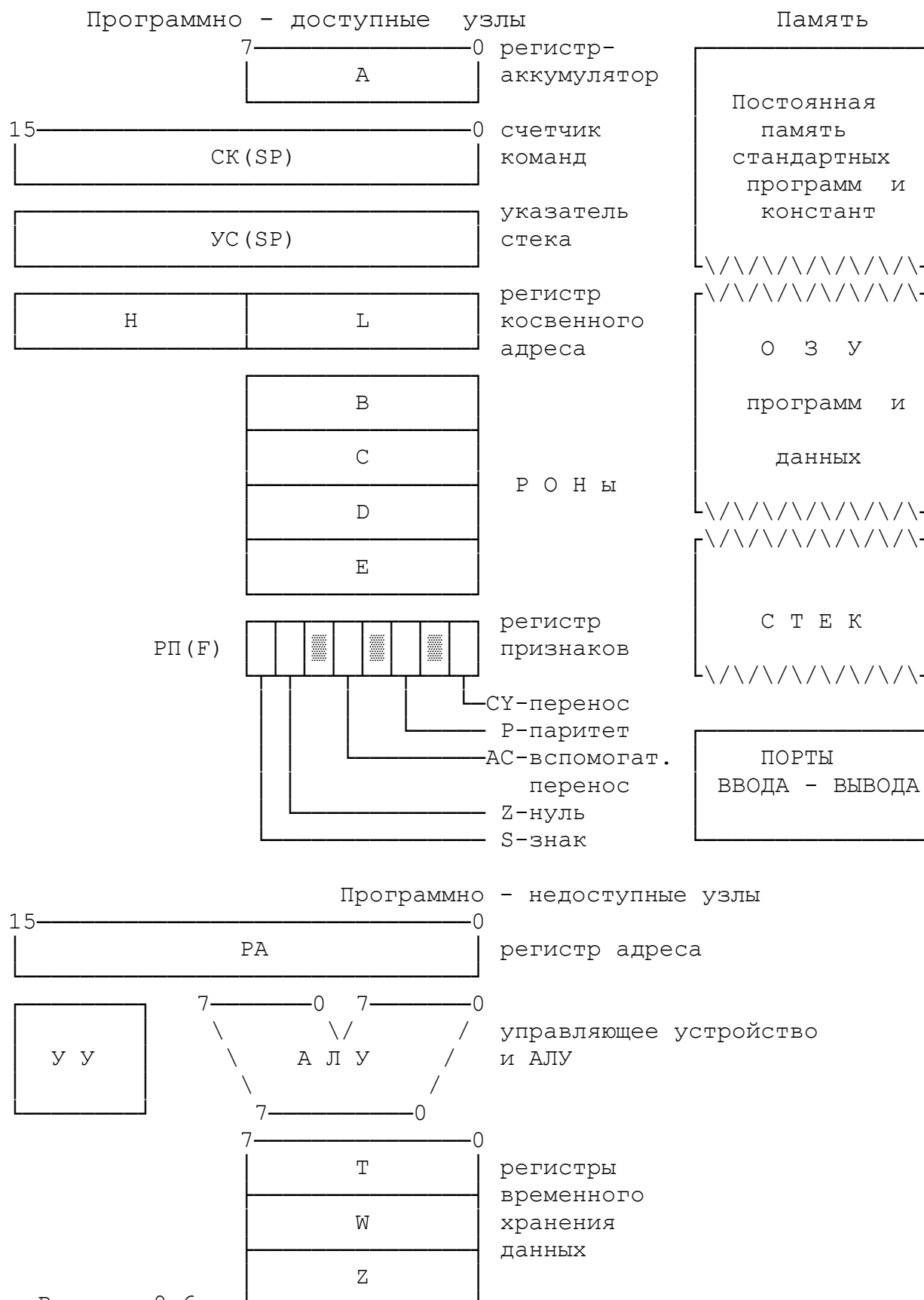
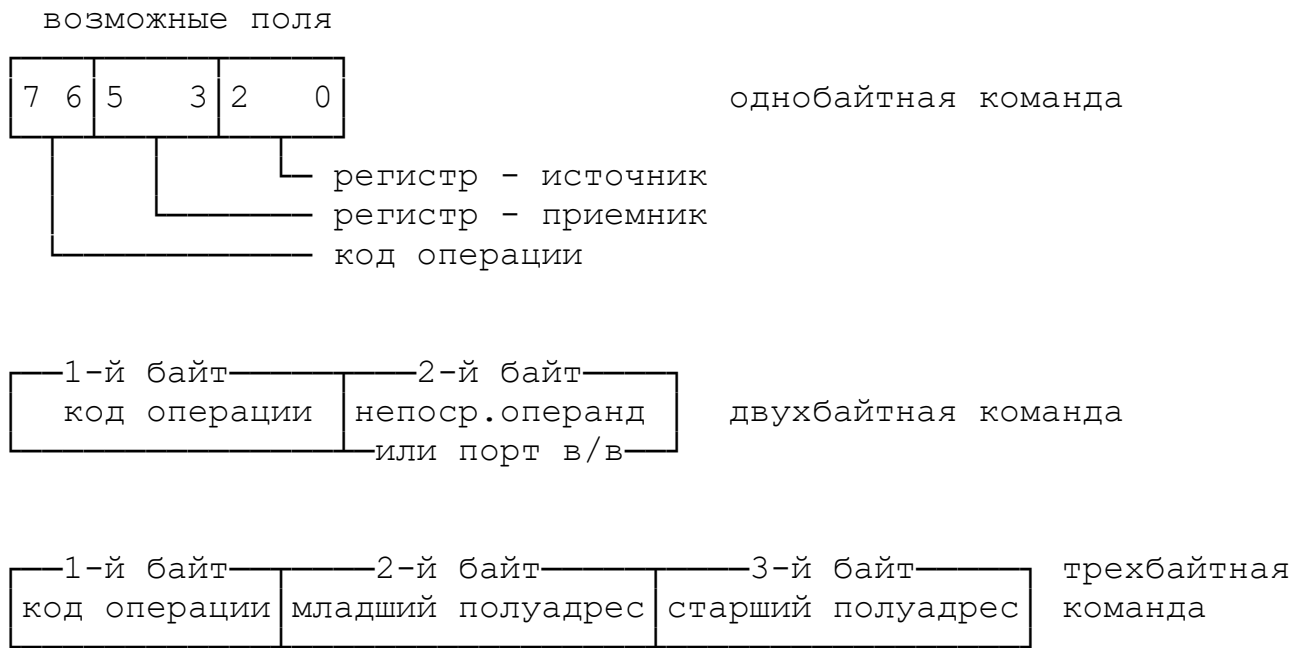


Рисунок 9.6.













Модель содержит только узлы, наиболее важные для понимания программистом процесса работы МП.

В системе команд МП имеются однобайтные, двухбайтные и трехбайтные команды. Формат команды и тип адресации задаются в команде неявно кодом операции. Адрес команды определяется адресом ее первого байта.






.


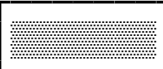



Варианты однобайтных команд

 D D D S S S	 - поле кода операции
 D D D 	DDD - трехбайтный номер регистра - приемника
 S S S	SSS - трехбайтный номер регистра - источника
 R P 	RP - двухразрядный указатель адресующий пару регистров
	CND - трехразрядный указатель условия передачи управления
 C N D 	NUM - трехразрядный код, определяющий адрес вектора прерывания
 N U M 	

Варианты двухбайтных команд

 D D D 	непоср. операнд
	непоср. операнд или порт в/в

Варианты трехбайтных команд

 R P 	младший полуадрес	старший полуадрес
	младший полуадрес	старший полуадрес
 C N D 	младший полуадрес	старший полуадрес

9.4.1. ГРУППЫ КОМАНД МП

Систему команд принято разделять на группы команд в соответствии с их функциональным назначением. По этому признаку выделяют пять групп:

- 1) пересылки информации;
- 2) арифметических операций;
- 3) логических операций;
- 4) ветвления;
- 5) управления.

1) ГРУППА КОМАНД ПЕРЕСЫЛКИ.

Группу команд пересылки информации образуют команды организации двустороннего обмена между внешней памятью и блоком РОН или между различными регистрами блока РОН. Признаки состояния МП не изменяются в процессе выполнения этих команд.

MOV R1, R2 (mov register) - передать из регистра в регистр

0	1	D	D	D	S	S	S
---	---	---	---	---	---	---	---

R1 ← R2

Содержимое регистра R2 копируется в регистр R1. При этом старое содержимое R1 уничтожается, а содержимое R2 не меняется.

DAX RP (load A indirect) - загрузить A с косвенной адресацией

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

A ← (RP)

Содержимое ячейки памяти, адрес которой содержится в паре адресуемых командой регистров, загружается в Аккумулятор.

STAX RP (Store A indirect)

0	0	R	P	0	0	1	0
---	---	---	---	---	---	---	---

(RP) ← A

Содержимое Аккумулятора загружается в ячейку памяти, адрес которой содержится в указанной паре регистров.

PUSH RP - загрузить в стек

1	1	R	P	0	1	0	1
---	---	---	---	---	---	---	---

(SP) ← RP;
SP ← SP - 2

Содержимое пары регистров загружается в стек. При этом значение указателя стека уменьшается на 2.

2) ГРУППА КОМАНД АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ.

Группа команд арифметических операций выполняет такие операции, как суммирование ADD, вычитание SUBtract, добавление единицы - INCrement или вычитание единицы - DECrement. Эта группа команд адресует только один из операндов, второй операнд всегда находится в аккумуляторе. В результате выполнения команд этой группы формируется ряд признаков.

ADD R (add register) - сложить с регистром

1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

$$A \leftarrow A + R$$

Содержимое Аккумулятора суммируется с содержимым регистра. Результат сохраняется в Аккумуляторе. Предыдущее значение A уничтожается, а содержимое регистра не меняется.

ADD M (add memory) - сложить с памятью

1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

$$A \leftarrow A + (HL)$$

Содержимое A суммируется с содержимым ячейки памяти по адресу, содержащемуся в паре HL.

ADI data (add immediate) - сложить с непосредственным операндом

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

D	A	T	A
---	---	---	---

$$A \leftarrow A + (\text{byte } 2)$$

Содержимое A суммируется с непосредственным операндом, содержащемся во втором байте команды.

ADC R (add register with carry) - сложить с регистром и с переносом

1	0	0	0	1	S	S	S
---	---	---	---	---	---	---	---

$$A \leftarrow A + R + CY$$

Содержимое A суммируется с содержимым регистра R и с содержимым разряда CY регистра признаков.

SBI data (subtract immediat with borrow) - вычесть непосредственный операнд с заемом

1	1	0	1	1	1	1	0
---	---	---	---	---	---	---	---

D	A	T	A
---	---	---	---

$$A \leftarrow A - (\text{byte } 2) - CY$$

DAD (add register pair to H and L) - сложить содержимое регистров с двойной точностью

0	0	R	P	1	0	0	1
---	---	---	---	---	---	---	---

$$HL \leftarrow HL + RP$$

3) ГРУППА КОМАНД ЛОГИЧЕСКИХ ОПЕРАЦИЙ.

Группа команд логических операций содержит команды, реализующие операции И (AND), ИЛИ (OR), исключающее ИЛИ (EXCLUSIVE OR), а так же сравнение (COMPARE) над данными. Так же как команды арифметических операций, эти команды адресует только один из операндов, второй операнд всегда находится в аккумуляторе. Результат операции образуется в аккумуляторе.

ANA R – поразрядное логическое сложение с A

1	0	1	0	0	S	S	S
---	---	---	---	---	---	---	---

 (A) <- (A) AND (R)

CMR M – Сравнение с памятью

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

 (A) <- M(HL)

Содержимое ячейки памяти, адрес которой хранится в HL вычитается из содержимого аккумулятора. Результат нигде не сохраняется. Изменяется только содержимое регистра признаков. Операция служит для сравнения двух чисел и организации условных переходов по этому признаку.

RLC – Циклический сдвиг содержимого A влево

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

 A <- Сдв.лев A

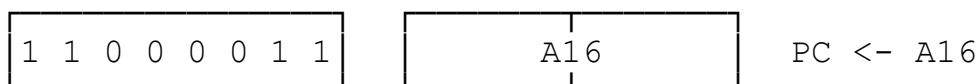
Содержимое A сдвигается влево. Содержимое старшего разряда перемещается в младший.

4) ГРУППА КОМАНД ВЕТВЛЕНИЯ.

Группа команд ветвления образуется набором условных и безусловных команд перехода JUMP и JUMP IF, команд вызова подпрограмм CALL и команд возврата RETURN. Группа команд JUMP IF используется тогда, когда по результату выполнения некоторой операции выбирается направление дальнейшего хода программы. Команды CALL и RETURN служат для организации связи основной программы с подпрограммами, т.е. для

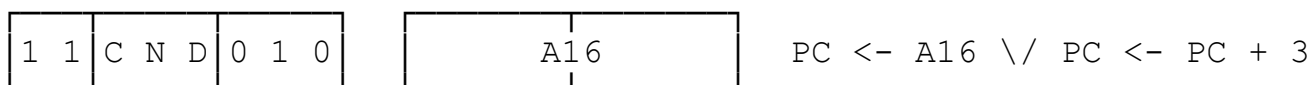
входа в подпрограмму и последующего возврата к основной программе. В МП системах команды этой группы получили широкое применение, так как по подпрограммам выполняются как арифметические операции типа умножения, деления, вычисления некоторых функций, так и операции ввода - вывода.

JMP A16 - безусловный переход



Происходит переход по адресу A16, указанному в команде (т.е. значение A16 загружается в счетчик команд)

J(COND) A16 - переход по условию

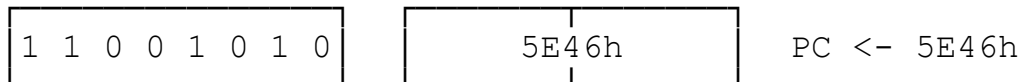


Если условие COND (Z, NE и др.) выполняется, то происходит переход по адресу A16, указанному в команде (т.е. значение A16 загружается в счетчик команд), если условие не выполняется, то выполняется следующая по очереди команда.

В командах условного перехода 3-х разрядный код CND задает условие передачи управления по адресу, указанному в команде. Возможны задания 8 вариантов условия переходов:

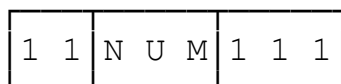
- по наличию переноса
(C) ;
- отсутствию переноса
(NC) ;
- нулевому результату
(Z) ;
- ненулевому результату
(NZ) ;
- положительному результату
(P) ;
- отрицательному результату
(M) ;
- четному числу единиц результата
(PE) ;
- нечетному числу единиц результата
(PO) ;

Например: условный переход по нулевому результату JZ 5E46h



Если признак z установлен в 1 (т.е. был нулевой результат), то происходит переход по адресу 5E46h, указанному в команде (т.е. значение 5E46h загружается в счетчик команд). Если признак z сброшен в 0 (т.е. был ненулевой результат), то выполняется следующая по очереди команда (значение счетчика команд увеличивается на три).

RST N - обслужить программное прерывание

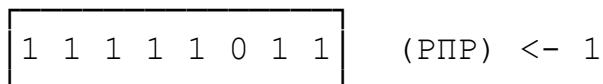


Текущее значение счетчика команд загружается в стек, а управление передается по адресу равному $N \times 8$, где N - номер прерывания, число от 0 до 7.

5) ГРУППА КОМАНД УПРАВЛЕНИЯ.

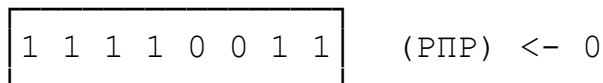
Группу команд управления образуют команды, управляющие процедурами ввода - вывода информации, операциями со стеком, и команды проверки текущего состояния процессора.

EI - разрешить прерывание



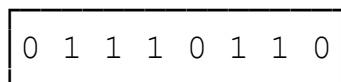
Разрешает прерывание программы. Устанавливает триггер разрешения прерывания РПР в 1.

DI - запретить прерывание



Запрещает прерывание программы. Устанавливает триггер разрешения прерывания РПР в 0.

HLT - Останов



Переводит процессор в режим останова.

NOP – Пустая операция

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Не выполняет никаких действий. Используется для организации программных задержек или для замещения других команд, подлежащих удалению.

9.5. СОСТАВ МПК КР580

МПК КР580, выполненный по n-МДП и ТТЛШ – технологии, характеризуется архитектурным единством, которое обеспечивается автономностью и функциональной законченностью отдельных микросхем, унификацией их интерфейса, программируемостью, их логической и электрической совместимостью. Восемьразрядная организация, фиксированный набор команд, большой выбор периферийных микросхем различного назначения, относительно высокое быстродействие, умеренное потребление мощности обеспечивают МПК широкое применение при создании средств вычислительной техники: устройств локальной автоматики, контроллеров измерительных приборов и периферийных устройств, микро – ЭВМ для управления технологическими объектами и измерительными системами.

МС КР580ВМ80А – функционально законченный однокристальный параллельный 8 – разрядный МП с фиксированной системой команд, применяется в качестве центрального процессора в устройствах обработки данных и управления.

МС КР580ВВ51А – универсальный синхронно – асинхронный приемо – передатчик (УСАПП), предназначен для аппаратной реализации последовательного протокола обмена между МП КР580ВМ80 или другим устройством, способным запрограммировать данную микросхему на

требуемый режим работы, и каналами последовательной передачи дискретной информации.

МС УСАПП преобразует параллельный код, полученный от ЦП, в последовательный поток символов со служебными битами и выдает этот поток в последовательный канал связи с заданной скоростью, а так же выполняет обратное преобразование: последовательный поток символов в параллельное 8 - разрядное слово. Передаваемая и принимаемая информация при необходимости может контролироваться на четность или нечетность.

МС УСАПП программируется на выполнение почти всех применяющихся в настоящее время протоколов последовательной передачи данных и работает в двух режимах: синхронном и асинхронном. Программирование МС на тот или иной режим работы выполняется записью в соответствующие регистры слов инструкции режима, служебных синхросигналов и инструкции команды.

Максимальная скорость приема/передачи информации по последовательному каналу 64 кБод, минимальная не ограничена и определяется внешними устройствами.

Рассмотрим один из наиболее распространенных стандартов последовательной передачи данных - интерфейс стандарта RS232 (отечественный аналог "Стык С2").

Последовательный интерфейс RS232.

Стандартный формат асинхронной последовательной передачи данных. Используемый в ЭВМ и ВУ формат содержит n пересылаемых битов информации и 3 - 4 дополнительных бита: стартовый бит, бит контроля по четности (или нечетности) и 1 или 2 стоповых бита (рисунок).

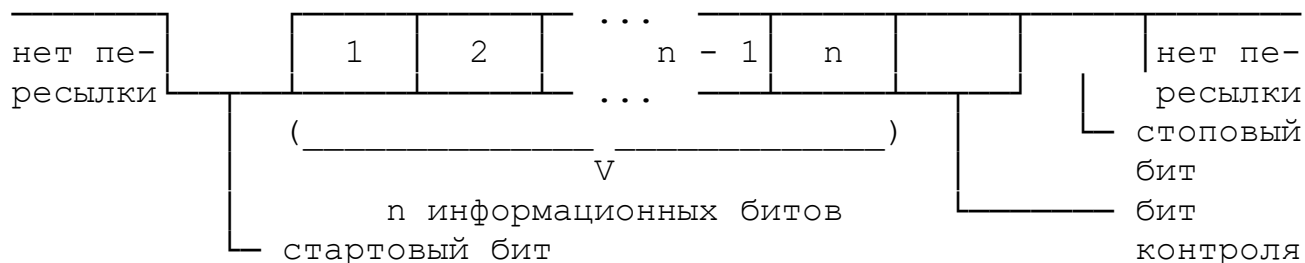


Рисунок 9.7

Когда передатчик бездействует, на линии сохраняется уровень сигнала логической 1. Передатчик может начать пересылку информации в любой момент времени посредством генерирования стартового бита, т.е. перевода линии в состояние логического 0 на время, точно равное времени передачи бита. Затем происходит передача информационных битов начиная с младшего значащего бита, за которыми может следовать дополнительный бит контроля. Далее с помощью стопового бита линия переводится в состояние лог. 1. При единичном бите контроля стоповый бит не изменяет состояние линии. Состояние логической 1 должно поддерживаться в течении промежутка времени, равного 1 или 2 времени передачи бита.

Промежуток времени от начала стартового бита до конца стопового бита называется кадром. Новый стартовый бит может быть послан в любой момент после окончания стопового бита. В линиях передачи данных передатчик и приемник должны быть согласованы по всем параметрам формата, включая номинальное время передачи бита. Для этого в приемнике устанавливается генератор синхроимпульсов, частота которого должна совпадать с частотой генератора передатчика. Кроме того для обеспечения оптимальной защищенности сигнала от искажения, шумов и разброса частоты синхроимпульсов приемник должен считывать принимаемый бит в середине его длительности.

предназначен для организации работы микропроцессорных систем в режиме реального времени. МС формирует сигналы с различными временными параметрами.

МС КР580ВВ55А - программируемое устройство ввода - вывода параллельной информации, применяется в качестве элемента ввода - вывода общего назначения, сопрягающего различные типы периферийных устройств с магистралью данных систем обработки информации. Условное графическое обозначение приведено на рисунке 9.8.

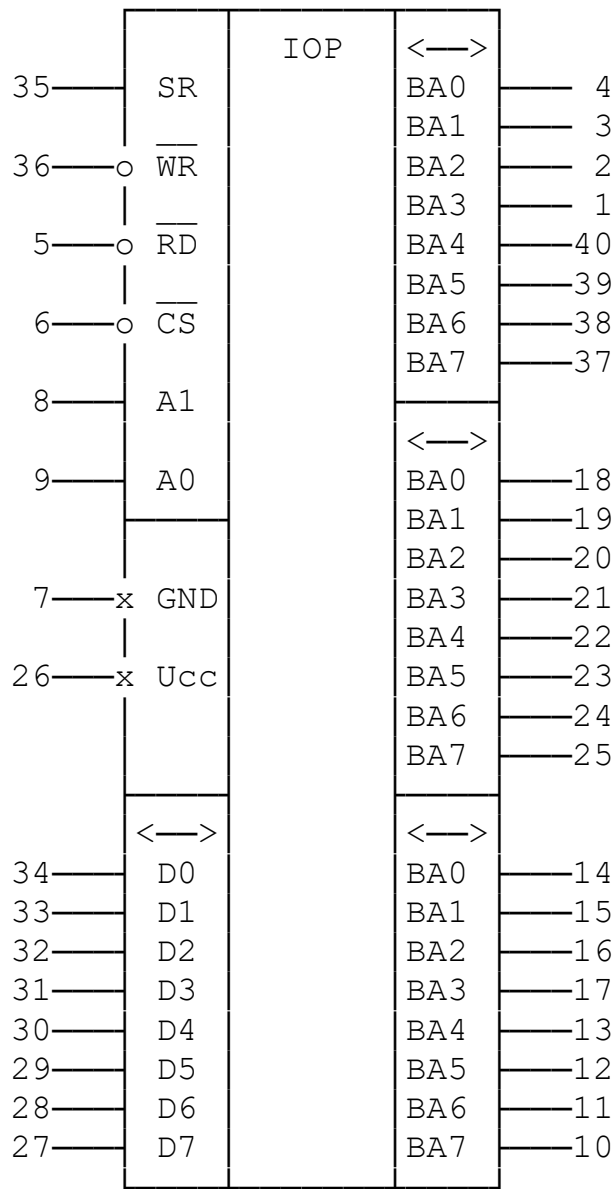


Рисунок 9.8. Условное графическое обозначение КР580ВВ55А

Назначение выводов KP580BB55A

Вывод	Обозначение	Тип вывода	Функ. назначение выводов
1 - 4	BA3 - BA0	Входы/Вы-	Информационный
37 - 40	BA7 - BA4	ходы	канал А
5	RD	Вход	Чтение информации
6	CS	Вход	Выбор микросхемы
7	GND	-	Общий
8, 9	A1, A0	Входы	Младшие разряды адреса
10 - 11	BC7 - BC4	Входы/Вы-	Информационный
	BC0 - BC3	ходы	канал С
18 - 25	BB0 - BB7	Входы/Вы-	Информационный
		ходы	канал В
26	Ucc	-	Напряжение питания + 5 В
27 - 34	D7 - D0	Входы/Вы-	Канал
		ходы	данных
35	SR	Вход	Установка в исходное сост.
36	WR	Вход	Запись информации

Структурная схема KP580BB55A

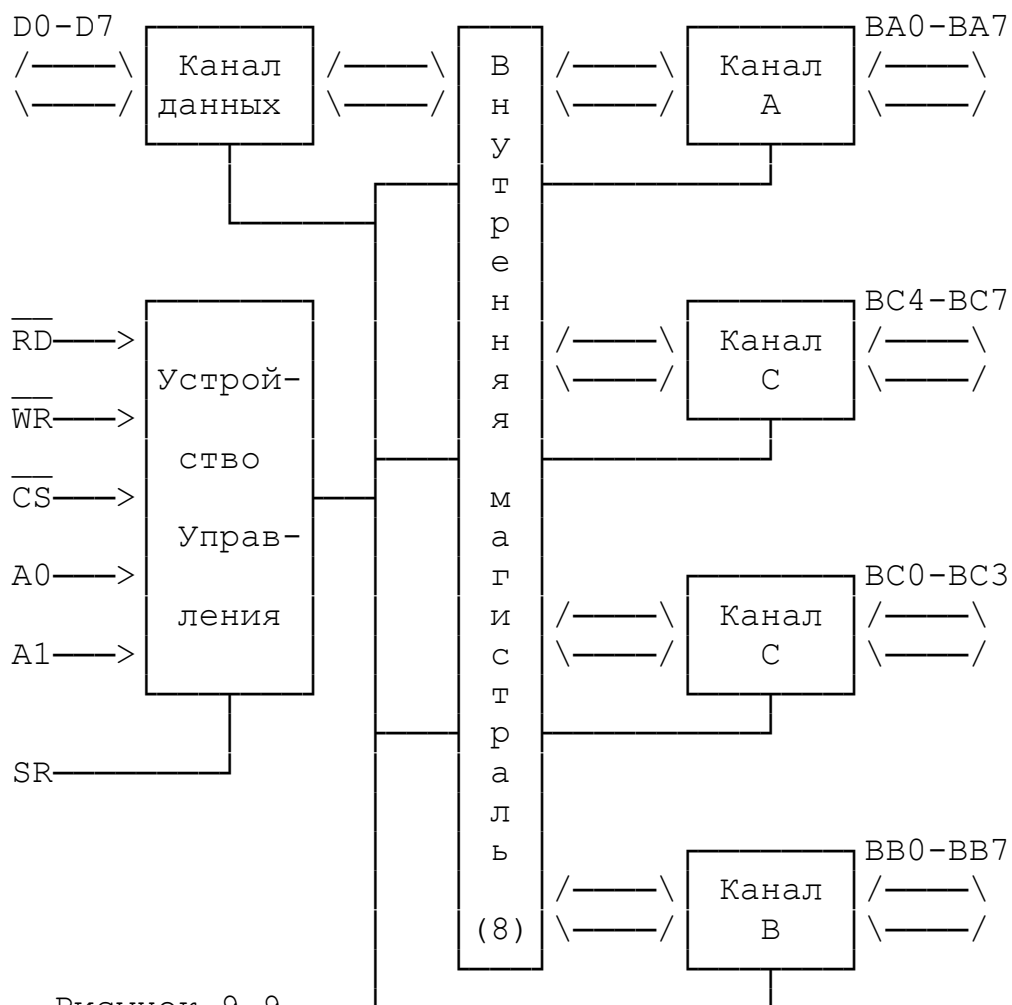


Рисунок 9.9.

Обмен информацией между магистралью данных систем и микросхемой КР580ВВ55А осуществляется через трехстабильный канал данных (D). Для связи с периферийными устройствами используются 24 линии ввода - вывода, сгруппированные в три 8-разрядных канала ВА, ВВ, ВС, направление передачи которых определяются программным способом. Микросхема может функционировать в трех основных режимах. В режиме 0 обеспечивается возможность синхронной программно - управляемой передачи данных через два независимых 8-разрядных канала ВА и ВВ и два 4-разрядных канала ВС.

В режиме 1 обеспечивается возможность ввода или вывода информации в или из периферийного устройства через два независимых 8-разрядных канала ВА и ВВ. При этом линии канала С используются для приема и выдачи сигналов управления обменом.

В режиме 2 обеспечивается возможность обмена информацией с периферийными устройствами через двунаправленный 8-разрядный канал ВА. Для передачи и приема сигналов управления обменом используются пять линий канала ВС. Выбор соответствующего канала и направление передачи информации через канал определяются сигналами А0, А1, соединяемые обычно с младшими разрядами шины адреса системы, и сигналами RD, WR, CS.

Режим работы каждого из каналов ВА, ВВ, ВС определяется содержимым регистра управляющего слова (РУС). Произведя запись управляющего слова в РУС, можно перевести микросхему в один из трех режимов работы: режим 0 - простой ввод/вывод;

режим 1 - стробируемый ввод/вывод;

режим 2 - двунаправленный канал.

При подаче сигнала SR РУС устанавливается в состояние, при котором все каналы настраиваются на работу в режиме 0 для ввода информации. Режим работы каналов можно изменять как в начале, так и в процессе выполнения программы, что позволяет обслуживать различные периферийные устройства в определенном порядке одной микросхемой. При изменении режима работы любого канала все входные и выходные регистры каналов и триггеры состояния сбрасываются.

В дополнение к основным режимам работы микросхема обеспечивает возможность программной независимой установки в 1 и сброса в 0 любого

из разрядов регистра канала ВС.

Рассмотрим один из наиболее распространенных стандартов параллельной передачи данных - интерфейс стандарта "Centronics" (отечественный аналог "ИППР-М").

Параллельный интерфейс "Centronics".

Основные характеристики:

- скорость передачи данных до 1000 байт/сек;
- синхронизация внешним импульсом строба;
- ответ сигналом BUSY или ASK;
- логические уровни напряжения совместимы с ТТЛ.

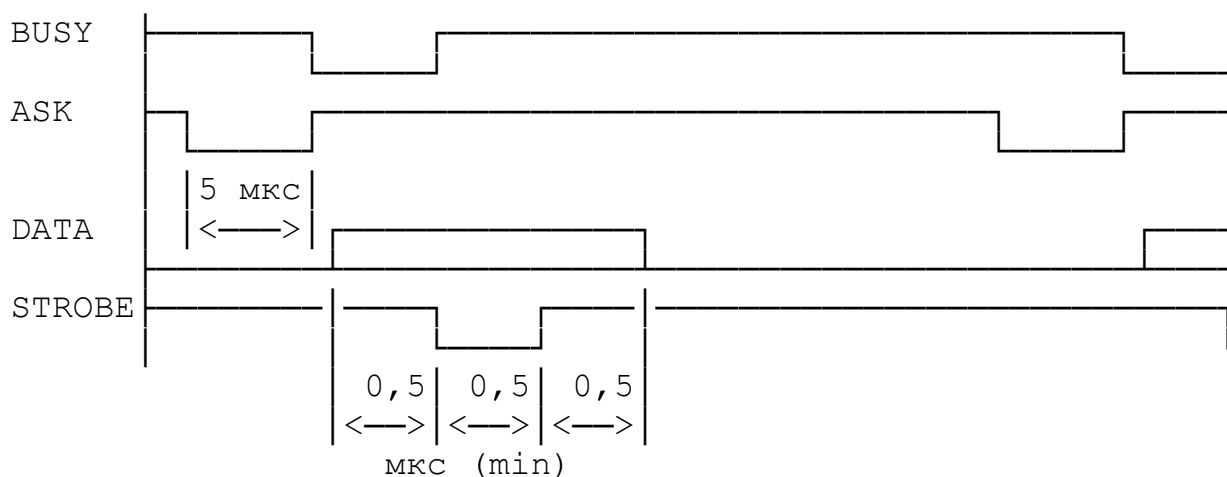


Рисунок 9.10. Временная диаграмма одного цикла передачи байта параллельным интерфейсом "Centronics".

BUSY - высокий уровень означает неготовность принтера (или другого приемника) к приему данных, низкий - готовность. Сигнал становится высоким во время ввода данных, во время печати, во время нахождения принтера в автономном режиме или в состоянии сбоя.

ASK - (Acknowledge) - переход с высокого уровня на низкий означает завершение ввода данных.

DATA - Данные D0..D7.

STROBE - синхронизирующий сигнал для записи данных в приемник.

Кроме того имеется ряд дополнительных управляющих сигналов.

МС КР580ВТ57 - четырехканальный программируемый контроллер прямого доступа к памяти (ПДП), предназначен для высокоскоростного обмена между памятью системы и периферийными устройствами путем генерации массива последовательных адресов памяти по требованию периферийного устройства. МС осуществляет двунаправленный обмен

данными между памятью и периферийными устройствами путем формирования в адресном канале микропроцессорной системы параметров заданного массива адресов ячеек памяти и управляющих сигналов. Массив адресов, по которым происходит обмен данными между периферией и памятью, характеризуется начальным адресом, т.е. первым адресом начала обмена и числом циклов обращений к памяти. После предоставления системной шины со стороны микропроцессора микросхема может осуществить обмен массивом данных между памятью и периферийными устройствами без дальнейшего вмешательства процессора.

Каждый из четырех каналов микросхемы обеспечивает адресацию, путем инкрементирования выработанного адреса, внешней памяти массивами объемом до 16 Кбайт с возможностью задания любого из 64 К начальных адресов.

МС КР580ВН59 – программируемый контроллер прерываний, обслуживает до восьми запросов на прерывание микропроцессора, поступающих от внешних устройств. Микросхема позволяет сократить средства программного обеспечения и реальные затраты времени при обработке прерываний в системах с приоритетами многих уровней. Алгоритм задания приоритета устанавливается программным путем. Приоритеты, закрепленные за внешними устройствами, могут быть изменены в процессе выполнения программы. В микросхеме предусмотрена возможность расширения числа обслуживаемых запросов до 64 путем каскадного соединения микросхем.

МС КР580ГФ24 – генератор тактовых сигналов С1, С2, предназначен для синхронизации работы микропроцессора КР580ВМ80.

Генератор формирует:

две фазы С1, С2 с положительными импульсами, сдвинутыми во времени, амплитудой 12 В и частотой 0,5 – 3,0 МГц;

тактовые сигналы опорной частоты амплитудой напряжения уровня ТТЛ; стробирующий сигнал состояния;

тактовые сигналы С, синхронные с фазой С2, амплитудой уровня ТТЛ.

МС КР580ИР82 и КР580ИР83 – 8-разрядные адресные регистры, предназначенные для связи микропроцессора с системной шиной; обладают повышенной нагрузочной способностью. МС КР580ИР82 – 8-разрядный D-регистр-"защелка" без инверсии и тремя состояниями на выходе, МС

КР580ИР83 - 8-разрядный D-регистр-"защелка" с инверсией и тремя состояниями на выходе (рисунок 9.11).

МС КР580ВА86 и КР580ВА87 - двунаправленные 8-разрядные шинные формирователи, предназначенные для обмена данными между микропроцессором и системной шиной; обладают повышенной нагрузочной способностью. МС КР580ВА86 - формирователь без инверсии и тремя состояниями на выходе, МС КР580ВА87 - формирователь с инверсией и тремя состояниями на выходе (рисунок 9.11).

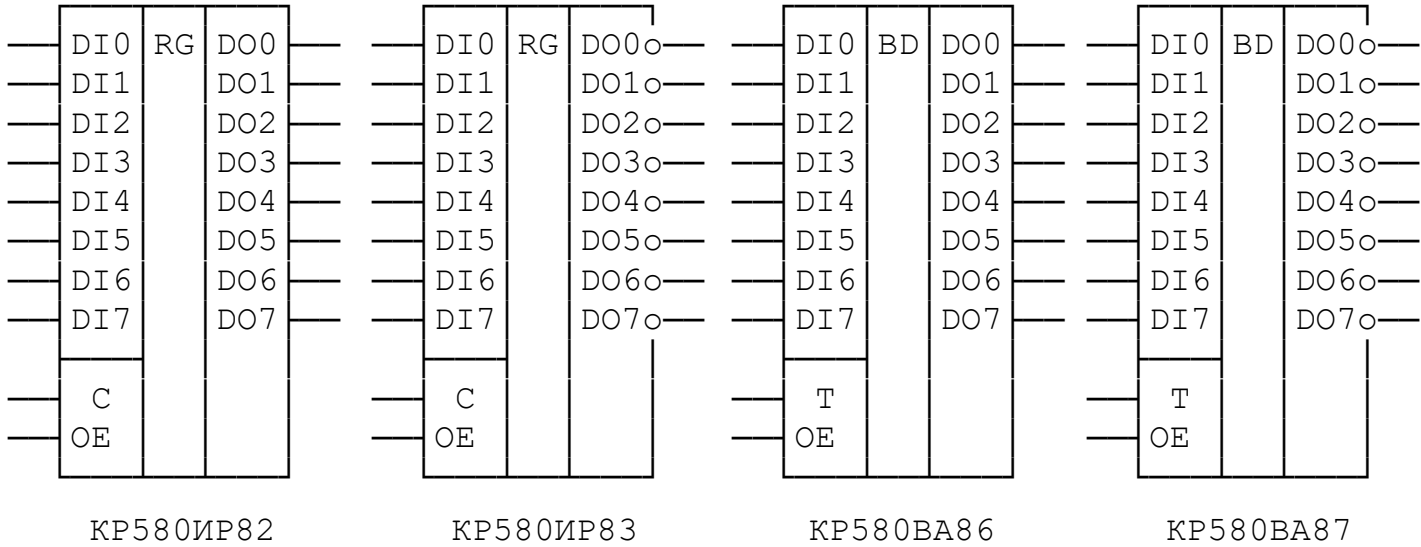


Рисунок 9.11

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 ЦМД;

Емкости;

Быстродействию;

Энергопотреблению и питанию;

Технологии изготовления;

Типу выхода (открытый коллектор, третье состояние, 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 БИС.

Конструктивно БИС ЗУ представляют собой п/п кристалл с площадью в несколько десятков квадратных миллиметров, заключенный в корпус.

10.1. АРХИТЕКТУРА БИС ЗУ

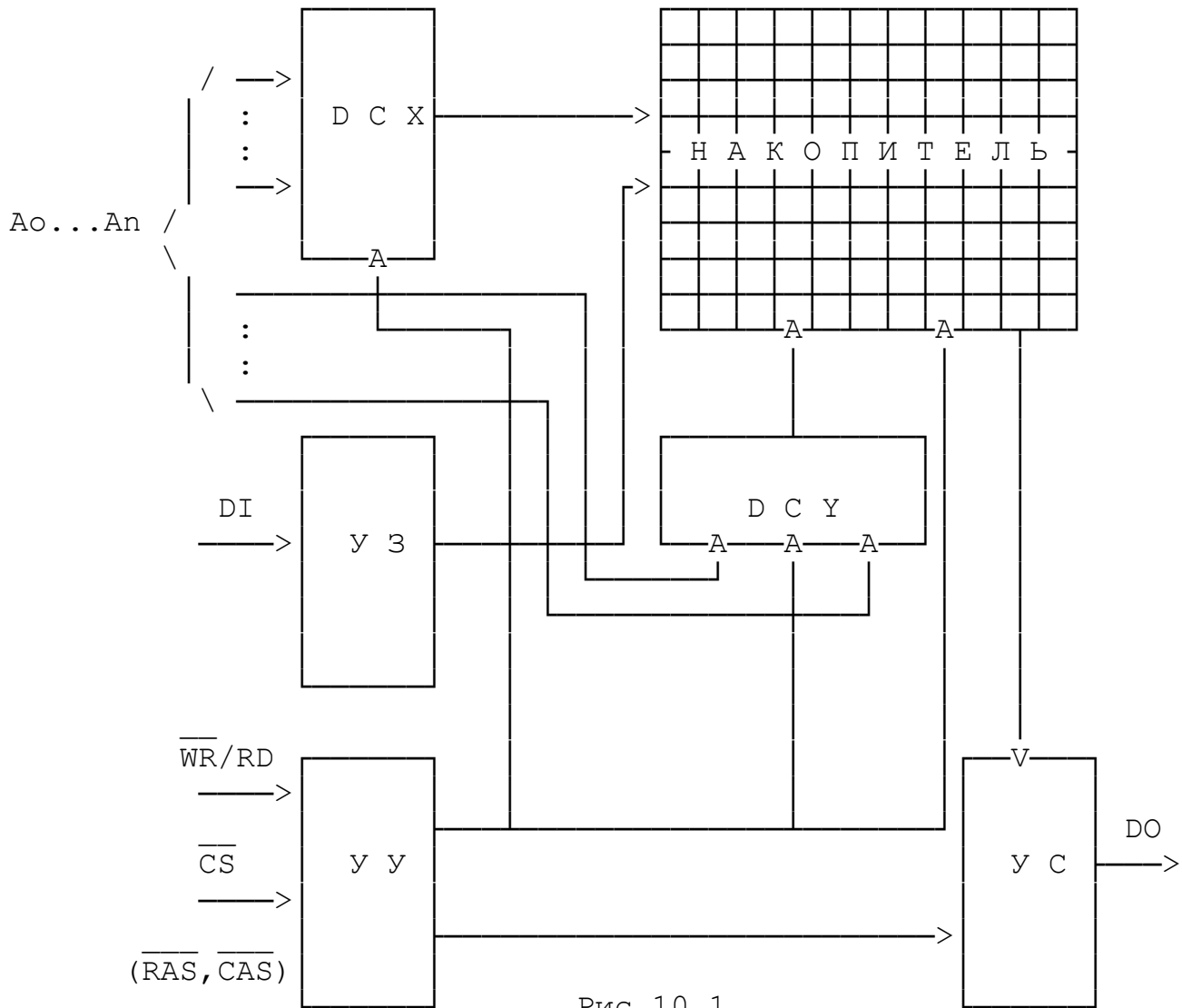


Рис.10.1.

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). 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 .

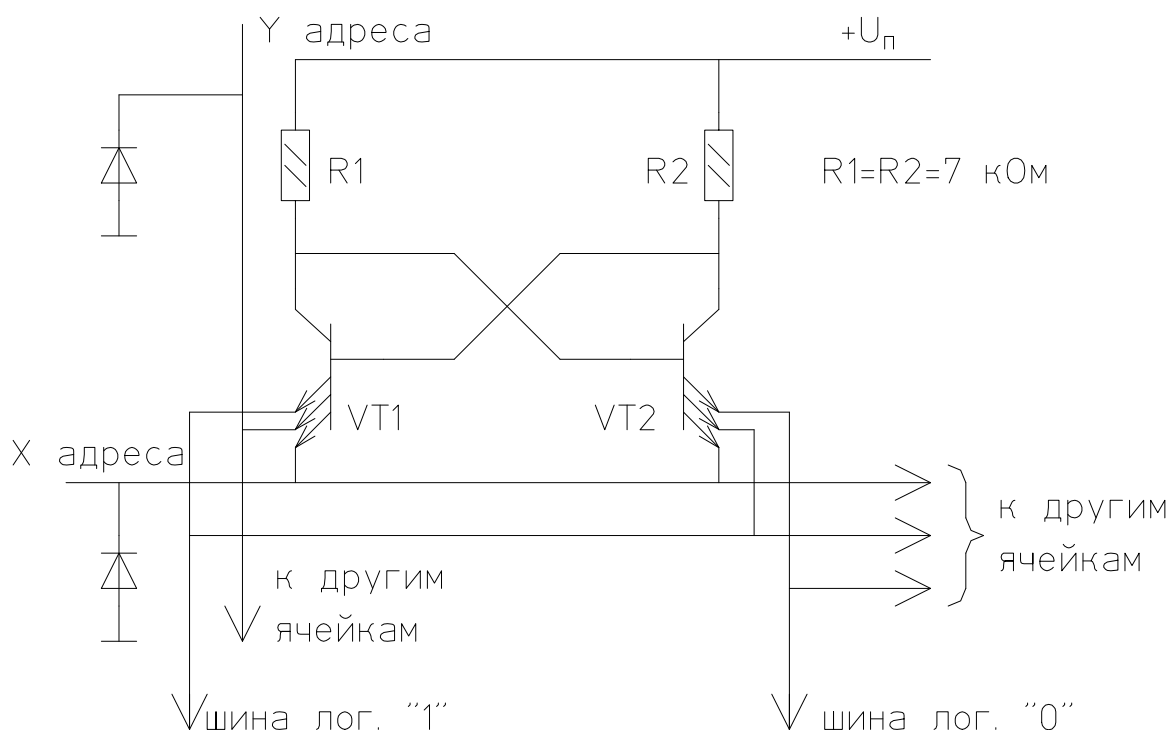


Рисунок 10.2 Ячейка статического ЗУ на биполярных транзисторах

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.

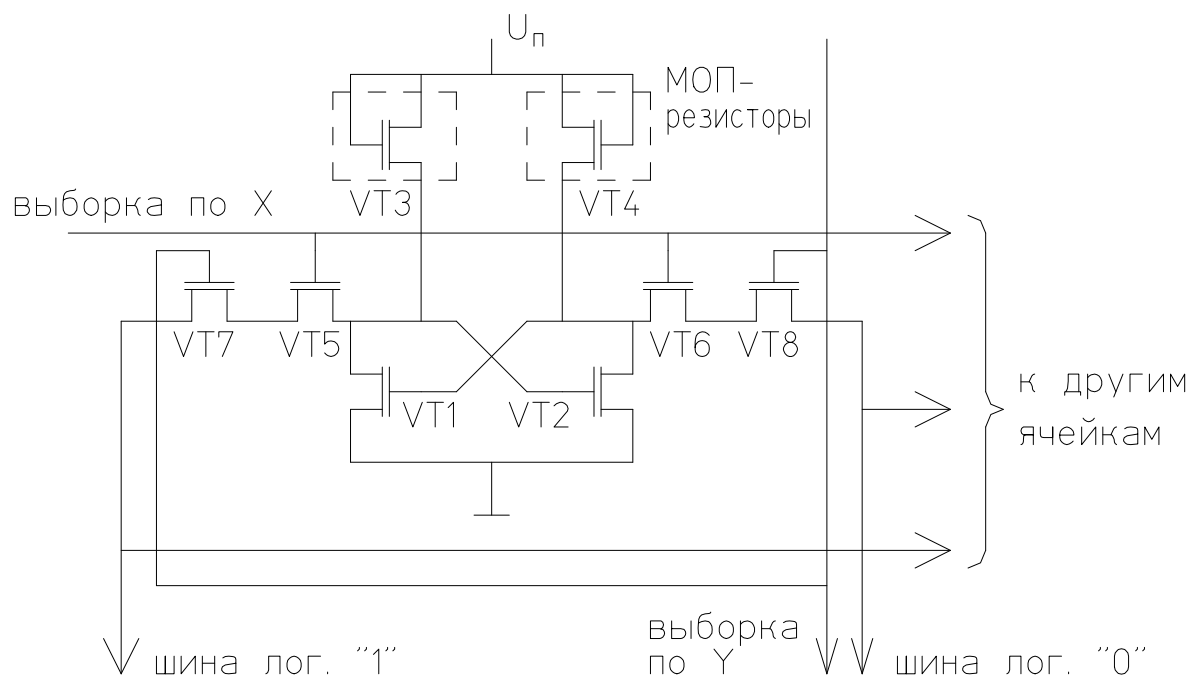


Рисунок 10.3 Ячейка статического ЗУ на МОП - транзисторах

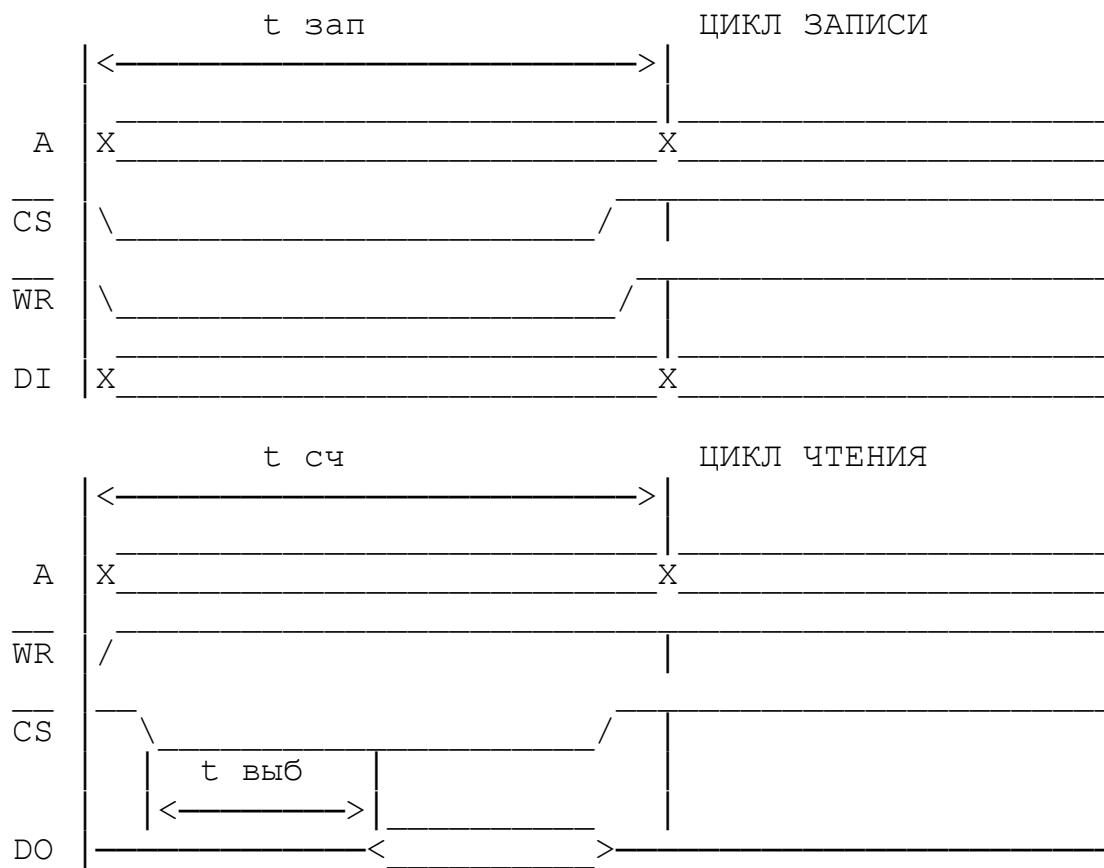


Рисунок 10.4.

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.

Структура микросхем динамического ОЗУ.

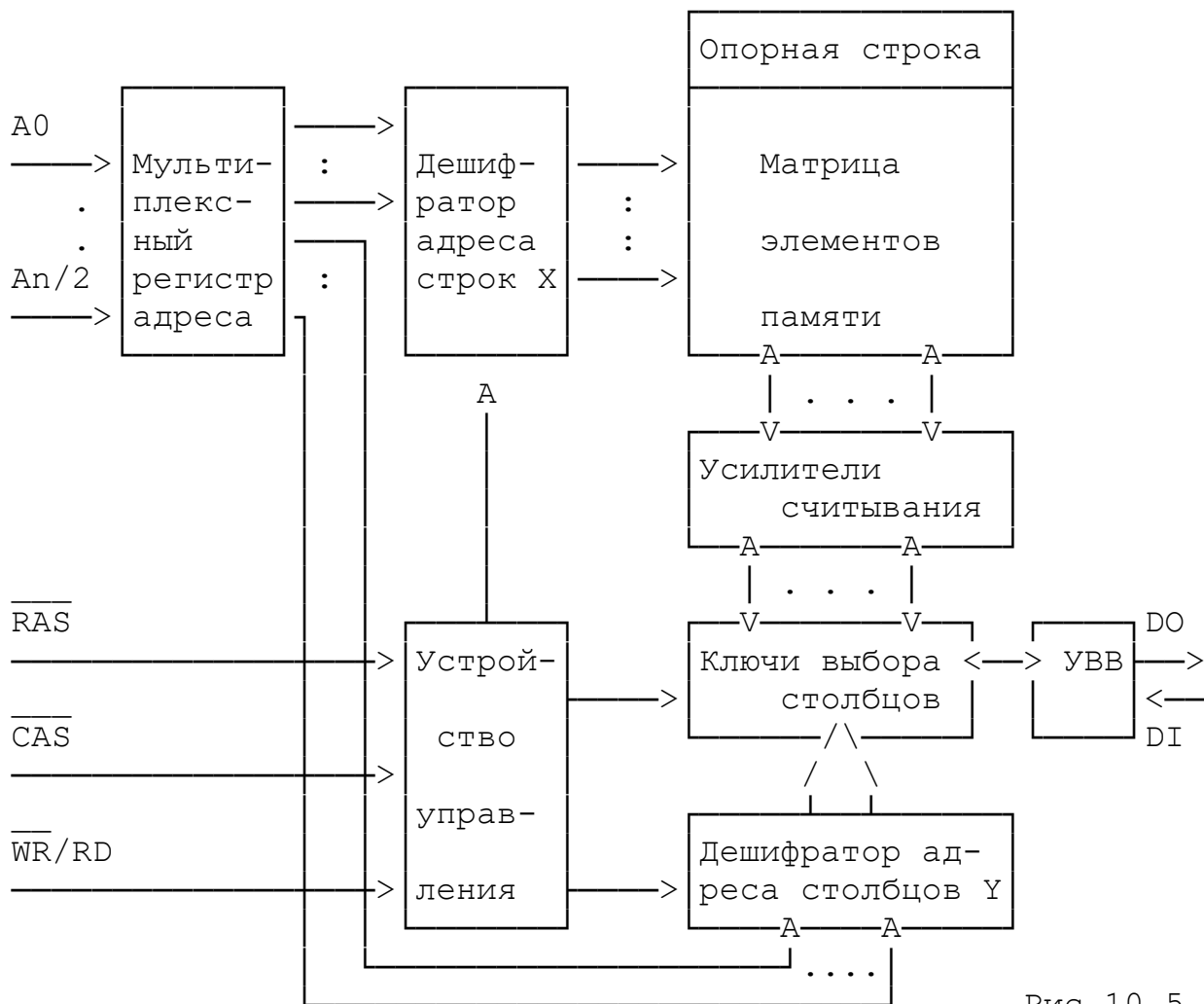


Рис.10.5.

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 рис. 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.

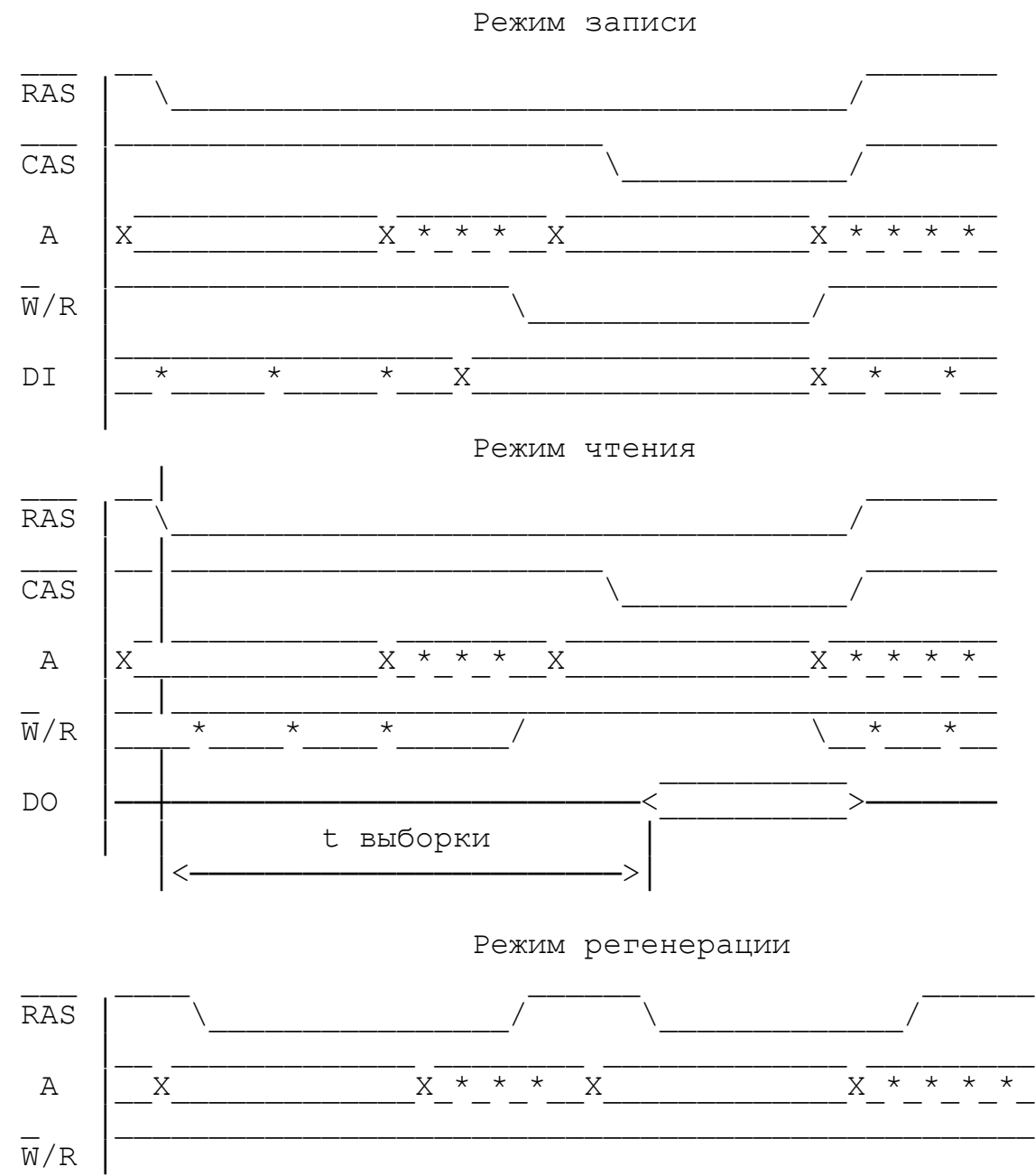


Рис. 10.6.

10.4. МИКРОСХЕМЫ ПОСТОЯННЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ (ПЗУ)

Микросхемы ПЗУ по способу программирования, т.е. занесения в них информации, подразделяют на 3 группы:

- 1. ПЗУ, однократно программируемые изготовителем по способу заказного фотошаблона (маски) – масочные ПЗУ (ПЗУМ, ROM);

2. ПЗУ, однократно программируемые пользователем по способу пережигания плавких перемычек на кристалле (ППЗУ, PROM);

3. ПЗУ, многократно программируемые пользователем, репрограммируемые ПЗУ (РПЗУ, EPROM).

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.

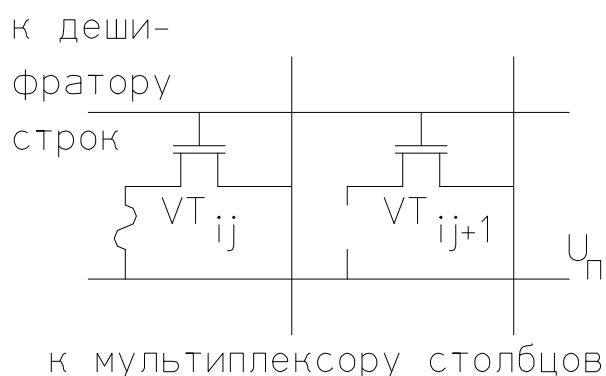


Рисунок 10.7 Ячейка ПЗУ

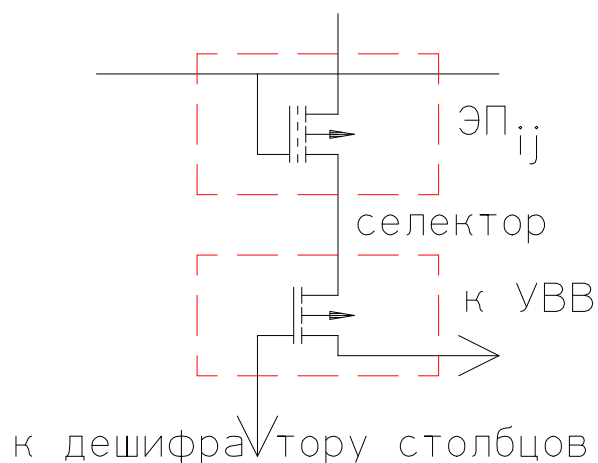


Рисунок 10.8 Ячейка РПЗУ

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. Всю 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 рис.10.8, а временные диаграммы работы всех рассмотренных ПЗУ аналогичны временным диаграммам статического ЗУ в режиме считывания (рис. 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 рис.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 (рис. 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

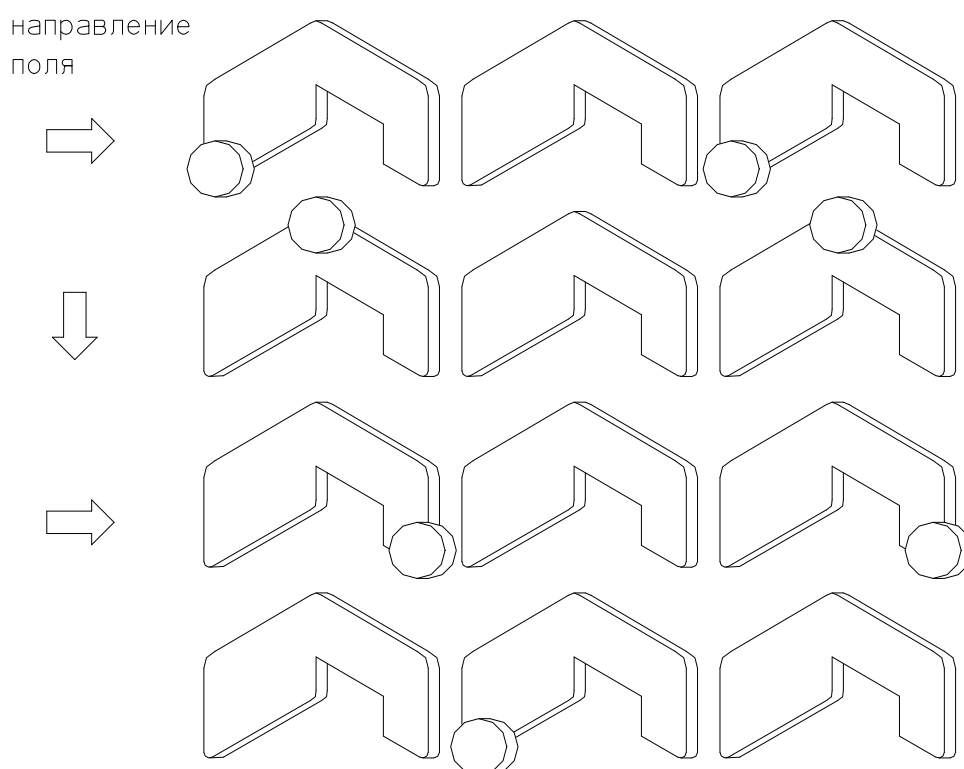


Рисунок 10.10 Движение доменов вдоль шевронов

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".

.

11. ПРЕОБРАЗОВАТЕЛИ

Основными параметрами преобразователя являются:

- число разрядов цифрового кода n ;
- максимальное входное напряжение U_{max} ;

$$\delta = \frac{1}{2^n - 1};$$

- относительная разрешающая способность
- абсолютная разрешающая способность $h = \delta \cdot U_{max}$ (или $\Delta U_{кв}$),

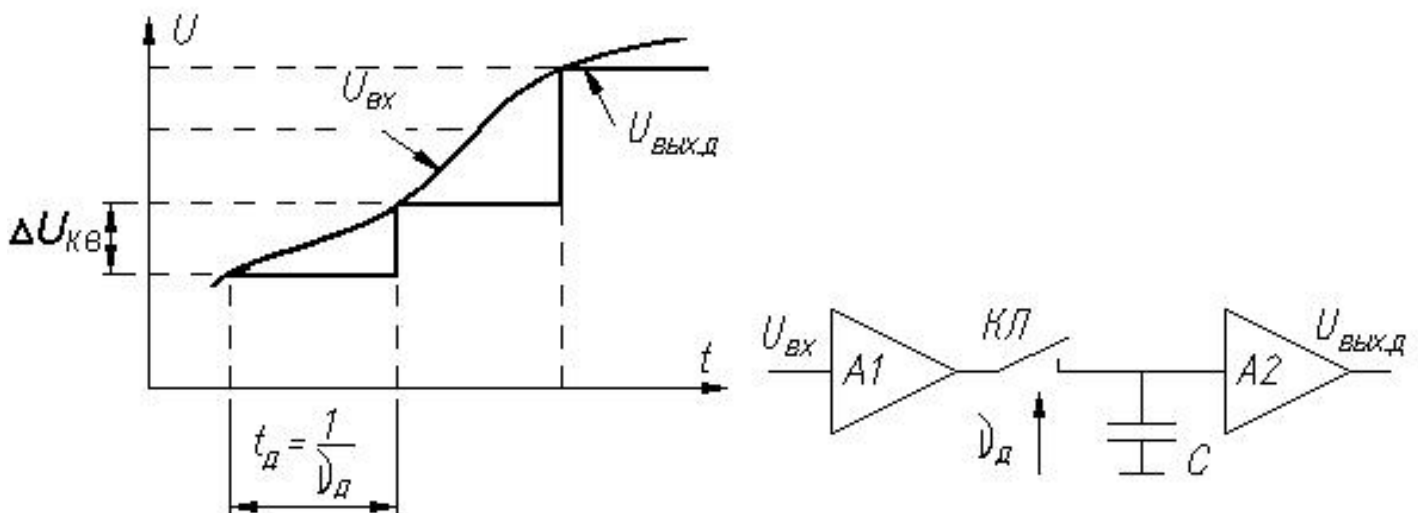
$$\varepsilon = \frac{|h \cdot k - U(k)|}{U_{max}};$$

- относительная нелинейность преобразования

где $U(k)$ входное напряжение, соответствующее коду k ;

- время преобразования $T_{пр}$ - интервал времени от подачи аналогового напряжения на вход АЦП до получения цифрового кода на выходе;

- максимальная частота преобразования $\nu_{пр}$ - наибольшая частота запуска, при которой параметры АЦП соответствуют заданным значениям.



11.1. Преобразователи код - напряжение (ПКН)

ПКН - функциональный узел, позволяющий получать на выходе напряжение, пропорциональное цифровому коду на входах ПКН.

ПКН часто называют цифро - аналоговыми преобразователями - ЦАП. Существует два принципа получения напряжения, пропорционального коду: последовательный и параллельный.

11.1.1 Последовательные ЦАП

В последовательных ЦАП формирование выходного сигнала осуществляется при последовательном преобразовании входного кода, поэтому время преобразования определяется числом n разрядов кода и элементарным тактом $t_{эл}$:

$$t_{пр} = n t_{эл}$$

Существует множество типов последовательных ЦАП. В основе всех типов этих ЦАП лежит идея накопления напряжения на конденсаторе (реже на индуктивности). Рассмотрим работу ЦАП этого типа на примере схемы с двумя накопительными конденсаторами (Рис. 11.1)

Входной код заполняет регистр сдвига либо параллельно либо последовательно. В каждом элементарном такте происходит анализ текущего разряда младшими разряда вперед на выходе регистра сдвига. При наличии 1 в данном разряде кода разрешается замыкание ключа $K1$, при наличии 0 в данном разряде разрешается замыкание ключа $K3$. В первой половине элементарного такта и при 1 в разряде кода замкнут $K1$ и разомкнут $K2$. $C1$ заряжается до напряжения $E_{эт}$. Во второй половине такта $K1$ и $K3$ разомкнуты и замкнут $K2$. Напряжение между $C1$ и $C2$ ($C1=C2$) перераспределяется поровну ($E/2$). В следующем такте при наличии 1 в коде вновь происходит перераспределение разностного заряда между емкостями поровну (т.е. на $C2$ будет $E/2 + E/4$) и т.д. Если в данном разряде кода присутствует 0, то $C1$ разряжается полностью через $K1$ в первой половине такта, а $C2$ отдает половину своего заряда конденсатору $C1$ во второй половине такта. После n тактов на $C1$ будет напряжение, пропорциональное входному коду. Максимальное напряжение на выходе равно

$$U_{\max} = E - E/2^n = (1 - 1/2^n) E = (1 - 2^{-n}) E$$

Последовательные ЦАП имеют низкое быстродействие, но просты и легко позволяют осуществить многоканальную работу при малом числе активных элементов схемы.

11.1.2. Параллельные ЦАП

Существует два основных метода построения параллельных ЦАП:

- 1) с суммированием напряжений
- 2) с суммированием токов

1) ЦАП с суммированием напряжений содержит n источников э.д.с., значения которых образует геометрическую прогрессию с показателем 2 (рис. 32) и n ключей, подключающих соответствующий источник э.д.с. при наличии 1 в соответствующем разряде кода. Выходное напряжение схемы равно:

Т.к. цифровой код на входе ЦАП выражается числом N :

то нетрудно видеть, что $U_{\text{вых}} = e N$, т.е. пропорционально входному коду.

2) ЦАП с суммированием токов содержит n ключей и n генераторов тока (рис. 33), выходные токи которых либо замыкаются на землю, либо суммируются на низком входном сопротивлении буферного усилителя:

Хорошим приближением к идеальному усилителю с нулевым входным сопротивлением является операционный усилитель, охваченный параллельной отрицательной обратной связью по напряжению для которого $R_{\text{вх}} \sim R_{\text{ос}}/K$. Буферный усилитель преобразует входной то-

ковый сигнал в выходной сигнал в виде напряжения.

Естественно, что в реальных системах из-за невозможности построения идеальных генераторов тока и напряжения, не всегда удастся точно классифицировать тип ЦАП. Ниже рассматриваются те варианты ЦАП, в которых легче всего обнаруживаются характерные признаки этих двух принципов построения ЦАП.

ЦАП с суммированием напряжений на матрице резисторов R-2R.

В качестве переключателей могут использоваться ключи на МОП, КМОП транзисторах (рис. 34). Для ключевых схем лестничный делитель R-2R имеет входное сопротивление равное $3R$ независимо от номера отвода. Коэффициент деления каждой ступени делителя равен 2. Поэтому, при замыкании i -го ключа на $E_{\text{эт}}$ на выходе схемы создается прирост напряжения $U_{\text{вых}i} = E_{\text{эт}} \cdot 1/2^i$, т.е. младшим значащим разрядом будет a_0 , старшим - a_{n-1} . Принципиальная схема ключа может быть реализована, например, на КМОП транзисторах, как показано на рисунке.

Быстродействие ЦАП такого типа определяется временем включения ключа $t_{\text{вкл}}$ и временем установления напряжения в матрице R-2R ($t_{\text{уст}}$). Последнее обусловлено наличием паразитных емкостей нагрузки, ключей и монтаже. Заряд паразитных емкостей в худшем случае происходит от 0 до $U_{\text{выхmax}}$. поэтому $t_{\text{уст}}$ бывает значительным. Снижение $t_{\text{уст}}$ достигается уменьшением значений резисторов R-2R. Типовые значения $n = 10..12$, $t_{\text{преоб}} = 5..10$ мкс.

11.1.3. ЦАП с суммированием токов.

Строятся с пассивными генераторами тока. Пассивные генераторы весовых токов реализуются с помощью эталонных э.д.с. $E_{\text{эт}}$ и набора весовых эталонных резисторов, номиналы которых образуют геометрическую прогрессию с показателем $1/2$ (рис. 35).

Переключатель тока в простейшем случае строится на диодах. При наличии логической 1 в i разряде диод D_{li} будет зак-

рыт высоким потенциалом на катоде диода. В этом случае выходной ток данного разряда будет равен $E_{эт}/(R/2) = E_{эт}/R * 2$, и будет протекать через диод $D2i$. При наличии логического 0 в i -м разряде на катоде $D1i$ будет низкий потенциал за счет диода смещения $D_{см}$ и весовой ток будет протекать по цепи: $E_{эт} - R_{o/2} - D1i - R_{эi} - (-E_{э})$, а диод $D2i$ будет закрыт. Весовой ток данного разряда на выходе будет равен только обратному току $D2i$. Т.к. суммирование токов происходит на очень низком входном сопротивлении буферного усилителя, то изменение выходного потенциала будет ничтожным, а следовательно влияние пере заряда паразитных емкостей незначительно. Быстродействие такой схемы в основном определяется временем установления буферного усилителя.

Активные генераторы и переключатели тока выполняются на транзисторах (рис. 36).

$T1$ и $T2$ - переключатель тока на базе дифференциального каскада. $T3$ - генератор стабильного тока (ГСТ) $I_o = U_{эт}/R_{эi}$. Номиналы резисторов $R_{эi}$ образуют геометрическую прогрессию с показателем 2.

Типовые параметры: $n=10..14$, $t_{преобр}=0,1..5$ мкс.

11.2. Преобразователи напряжение - код.

Преобразователи напряжение - код (ПНК) или иначе аналого - цифровые преобразователи (АЦП) - функциональные устройства, выходной код которых пропорционален амплитуде входного сигнала.

Т.к. большинство физических сигналов, амплитуда которых преобразуется в цифровой код, представляют непрерывные функции времени, а цифровые устройства - устройства дискретного действия, то входные аналоговые сигналы предварительно дискретизируют. Процесс дискретизации заключается в фиксации (запомина-

нии) мгновенного значения входного сигнала в определенные моменты времени и сохранение зафиксированного значения до следующей выборки из входного сигнала (рис.37). Интервал дискретизации t_d или частота дискретизации f_d (временного квантования) должны удовлетворять теореме Котельникова:

$f_d > 2f_v$, где f_v – высшая частота в спектре преобразуемого сигнала. При невыполнении условия теоремы однозначное восстановление исходного сигнала из дискретизированного (обратная операция) невозможно. Практически f_d превышает $2f_v$ в 1,5..3 раза, т.е. $f_d \sim 3..6 f_v$, что значительно упрощает последующее восстановление аналогового сигнала после цифровой обработки.

Операция дискретизации производится обычно с помощью устройств выборки – хранения (УВХ), состоящих из ключа КЛ (рис. 38), запоминающего конденсатора С и буферных усилителей А1 и А2. Усилитель А2 должен иметь высокое входное сопротивление, а ключ в разомкнутом состоянии – малые токи утечки. В этом случае обеспечивается длительное хранение запомненной информации. Низкое выходное сопротивление усилителя А1 и низкое сопротивление замкнутого ключа – условия, необходимые для быстрого заряда емкости.

Преобразование амплитуды в код начинается обычно с момента размыкания ключа в УВХ. Существует множество способов А/Ц преобразования.

- 1) АЦП с последовательным счетом;
- 2) АЦП с поразрядным кодированием;
- 3) АЦП параллельного действия;
- 4) сигма – дельта АЦП.

11.2.1 АЦП последовательного счета

Наиболее медленные, однако они допускают варианты построения с интегрированием входного сигнала, что резко снижет

действие шумов, помех и наводок. Разрядность таких АЦП доходит до 14...16 (84...96 дБ). Характерным для этих АЦП является преобразование амплитуды входного сигнала в длительность временного интервала и счет числа импульсов образцового генератора за этот временной интервал (рис. 39).

По сигналу "Пуск" ГПН начинает формировать линейно возрастающее напряжение, которое поступает на один из входов компаратора и сравнивается с входным сигналом. В это же время происходит счет импульсов генератора КГ счетчиком СТ. В тот момент, когда напряжение ГПН становится равным напряжению входного сигнала, счет прекращается. Код на выходе счетчика пропорционален амплитуде входного сигнала. При этом время преобразования зависит от амплитуды преобразуемого сигнала ($t_{преобр} \sim 1..10 \text{ мс}$).

11.2.2. АЦП поразрядного уравнивания (последовательного приближения)

Обеспечивают существенно большее быстродействие и умеренную точность. число разрядов обычно не превышает 12 (рис. 40).

При запуске схемы старший разряд регистра RG2 устанавливается в 1, а все остальные разряды сбрасываются в 0. В регистре RG также записывается 1 в старший разряд. С выхода ЦАП снимается $U_{цап}$, равное весу старшего разряда $U_{цап} = U_{max}/2$. Если U_x превышает это значение, то с выхода схемы сравнения напряжений СС поступает 0 на выбранную первую схему совпадения И1 и триггер an регистра RG2 останется в состоянии 1. Следующим тактом генератора импульсов ГИ "1" в регистре RG сдвигается вниз подготавливая И2 и устанавливая разряд an-1 регистра RG2 в 1. Если U_x оказывается меньше этого значения, то с выхода СС будет поступать 1 на схему И2, что приведет к сбросу в 0 разряда an-1 регистра RG2. За n тактов будут испытаны все разряды кода, а в регистре RG2 будет храниться цифровой эквивалент

входного сигнала, а $U_{цп}$ с точностью до единицы младшего разряда будет равно U_x . Типовые времена преобразования таких АЦП $t_{преобр} \sim 1..20$ мкс.

11.2.3. АЦП параллельного действия.

Наиболее быстродействующие, однако разрядность обычно не превышает 6 (рис. 41).

Напряжение входного сигнала одновременно сравнивается с 2^n уровнями напряжения сформированными эталонным резистивным делителем из Уэт. Для этого АЦП должен содержать 2^n компараторов. Сигналы с компараторов поступают на шифратор, где преобразуются в двоичный код.

Типовое время преобразования $t_{преобр} = 20..10$ нс.

11.2.4. Сигма - дельта АЦП.

Принцип действия основан на компенсации входного среднего тока током внутреннего источника (рис. 42).

Входное напряжение подается на интегратор, выходной сигнал которого сравнивается с фиксированным напряжением, например, с нулем. Импульсы тока фиксированной длительности (то есть с фиксированным приростом заряда) на каждом такте подключаются, в зависимости от состояния выхода компаратора, либо к суммирующей точке интегратора, либо к земле. В суммирующей точке поддерживается нулевой средний ток, то есть преобразование основано на принципе уравнивания. Счетчик подсчитывает количество импульсов, которые поступают на суммирующую точку за интервал времени, определяемый путем отсчета заданного числа тактовых импульсов. Число, полученное в счетчике за это время, пропорционально среднему значению входного сигнала и может быть использовано в качестве выходного кода.

В сигма - дельта АЦП для формирования импульсов тока также можно использовать резистор и стабилизированный источник опорного напряжения, поскольку суммирующая точка фактически

находится под потенциалом земли. В этом случае сопротивление замкнутого ключа должно быть меньше сопротивления резистора R .

.

12 МИКРОПРОЦЕССОРЫ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

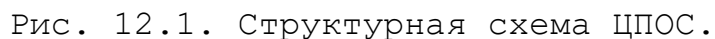
В связи с широким внедрением МП в системы реального времени, в микропроцессорной технике заметна тенденция создания высокопроизводительных приборов в ущерб универсальности. Наиболее мощным классом специализированных приборов являются микропроцессоры для цифровой обработки сигналов (ЦОС). Системы ЦОС нашли широкое применение при решении задач, связанных с обработкой радиолокационной и телеметрической информации, звуковых сигналов и изображений и других сложных задач, решаемых на основе МП. Значительная эффективность систем ЦОС достигается за счет специализации МП. МП, ориентированные на ЦОС, получили название цифровых процессоров обработки сигналов (ЦПОС).

Такие процессоры характеризуются:

- 1) небольшим числом типов используемых команд;
- 2) функциональной законченностью;
- 3) наличием в своем составе:
 - тактового генератора;
 - нескольких видов памяти;
 - интерфейсных схем;
 - многоканальных ЦАП и АЦП;
 - схем выборки и хранения.

В настоящее время распространены несколько различных семейств ЦПОС: семейство TMS320 фирмы Texas Instruments, семейство ADSP-2100 фирмы Analog Devices, 8015 фирмы Motorola и серии K1813 и K1827 выпускаемые отечественной промышленностью.

Микросхема КМ1813ВЕ1 представляет собой однокристалльную ЭВМ с встроенными ЦАП и АЦП, архитектура и система которой преимущественно ориентированы на решение задач цифровой фильтрации сигналов.



В структуре ЦПОС КМ1813ВЕ1 можно выделить аналоговую и цифровую части, функционирующие под управлением программы

хранящейся в РПЗУ. Аналоговая часть представляет собой интерфейс для ввода - вывода сигналов и содержит узлы для многоканального аналого - цифрового и цифро - аналогового преобразования:

- входной мультиплексор на четыре входа (М);
- входное устройство выборки - хранения (УВХ);
- компаратор (К);
- девятиразрядный ЦАП;
- выходной демультиплексор на восемь выходов (ДМ);
- выходные УВХ с усилителями на каждый выход.

Аналого - цифровое преобразование выполняется методом последовательного приближения и результат получается в специальном регистре DAR (digital - analog register), адресуемом как ячейка ОЗУ с адресом 40. Уровень преобразуемого напряжения запоминается на конденсаторах УВХ, который является общим для всех входов и подключается к конкретному входу только в момент преобразования. Для заряда емкости с требуемой точностью до уровня входного напряжения необходимо выполнить последовательно до 10 команд ввода $In(k)$. После этого отсчет преобразуется в его цифровой эквивалент командой $CVT(n)$, начиная со знакового и кончая младшим разрядом. Для полного девятиразрядного преобразования при максимальной тактовой частоте требуется 34 команды, что соответствует времени преобразования около 20 мкс. Входной отсчет может быть представлен и с меньшей точностью, тогда требуемое число команд уменьшается. Аналоговые инструкции декодируются одновременно с цифровыми, т.е. реали-

зация аналоговых функций, как правило не приводит к снижению возможностей цифровой обработки. Результат аналого - цифрового преобразования хранится в регистре DAR, из которого может быть переписан в любую ячейку ОЗУ, использован в операциях АЛУ или через ЦАП выдан на любой из восьми выходов.

Цифро - аналоговое преобразование данных осуществляется методом поразрядного взвешивания двоичного кода числа, содержащегося в DAR. Номер выхода задается инструкцией OUT(k). Для заряда емкости выходного УВХ требуется несколько таких инструкций (от 4 до 7). Диапазон входного и выходного напряжений определяется напряжением внешнего опорного источника (1..2В), подключаемого к выводу Uref и аналоговой земле GRDA. Входные и выходные сигналы могут быть также цифровыми с уровнями TTL. Управление режимом входов - выходов осуществляется по входам M1 и M2.

Цифровая часть построена на основе Гарвардской архитектуры, особенностью которой является наличие отдельных блоков памяти программ и данных.

Память программ представляет собой РПЗУ с ультрафиолетовым стиранием емкостью 192 слова по 24 разряда. При работе ЦПОС доступ ко всем ячейкам памяти команд последовательный. Счетчик команд РПЗУ (СчК) возвращает нулевое состояние после выполнения команды с адресом 191 или когда в поле команды встречается инструкция "Конец программы" или "Возврат по условию". Каждая команда выполняется за четыре такта задающего генератора. Команды из РПЗУ считываются четверками команд в бу-

ферный 96 - разрядный регистр. Чтение из памяти каждой четверки команд сопровождается сигналом на выходе CCLK. Цифровая часть включает двухпортовое ОЗУ данных и констант, масштабирующее устройство и АЛУ. Каждый из этих узлов получает инструкцию или код адреса из РПЗУ.

Данные, поступающие в АЛУ, обрабатываются с использованием 25 - разрядной арифметики в дополнительном коде. Старший разряд является знаковым, значения переменных лежат в пределах $-1...+1$, а ошибка округления результата имеет величину 2^{-24} . Переполнение отображается импульсом на выходе \overline{OF} и может обрабатываться логически с целью коррекции результата.

Память данных включает матрицу ОЗУ статического типа с произвольной выборкой одновременно двух 25 - разрядных слов и порты - регистры для хранения операндов А и В. Емкость ОЗУ составляет 40 слов. Свободная область шестиразрядного адреса используется для задания констант в программе.

Система команд ЦПОС КМ1813ВЕ1 включает группу цифровых и аналоговых команд. В свою очередь, цифровые команды разделяются на безусловные и условные. Все цифровые команды могут выполняться со сдвигом операнда А или без сдвига.

Все команды ЦПОС используют прямую адресацию ячеек ОЗУ. При этом значения адреса от 0 до 39 соответствуют регистрам общего назначения, а значения адреса более 47 обозначают константы, которые можно рассматривать как непосредственный операнд.

Позднее была разработана модификация ЦПОС КМ1813ВЕ1,

имеющая цель расширить функциональные возможности системы при сохранении программной совместимости и функциональной завершенности. Основным отличием модифицированного кристалла является введение цифрового параллельного порта и связанного с ним блока регистров. Цифровой ввод - вывод существенно расширил область применения ЦПОС, обеспечив возможность реализации многопроцессорных структур и использование ЦПОС в качестве периферийного устройства цифровых систем. Цифровой порт D7 - D0 имеет сигналы управления \overline{WR} , \overline{RDY} , аналогичные по функциям сигналам МПК КР580. Емкость ПЗУ программ расширена более чем вдвое и имеется возможность работы с внешней памятью программ. Для этого из кристалла выведены входы регистра команд P0 - P7.

Точность ЦАП и АЦП увеличена до 11 разрядов. Увеличено до 8 число аналоговых входов.

Однокристалльные микроЭВМ КМ1827ВЕ3 и КМ1827ВЕ4 предназначены для цифровой обработки сигналов звукового диапазона частот в реальном масштабе времени. Микросхемы выполнены по n-МОП технологии и имеют одинаковую архитектуру. Различие заключается в реализации ПЗУ команд. ВЕ3 имеет масочную память, программируемую при изготовлении, а ВЕ4 - ПЗУ, программируемую пользователем. БИС обрабатывают 16 - разрядные числа, представленные в дополнительном коде с фиксированной точкой. Диапазон представления чисел $\pm 2^{-15}$.

Сопряжение БИС с внешним устройством осуществляется посредством программируемых 8/16 - разрядных параллельного и последовательного портов ввода - вывода. Параллельный порт согла-

суется по сигналам интерфейса с универсальными микропроцессорами серий КР580, К1810, К1816, К1821 и др.

12.2 ЦПОС с фиксированной точкой фирмы Analog Devices

Семейство этих процессоров имеет обозначение ADSP-21XX. Первый процессор этого семейства ADSP-2100 (86 г.) имеет 80 - наносекундный цикл и выполнен по 1 - микронной CMOS - технологии. Архитектура семейства оптимизирована под алгоритм цифровой обработки сигналов, что повышает эффективность вычислений. Процессоры семейства отличаются друг от друга внутренними устройствами, содержащимися в микросхеме. Ознакомимся с базовой архитектурой ADSP-21XX (рис. 12.2).

12.2.1 Базовая архитектура

Семейство процессоров ADSP-21XX используют модифицированную Гарвардскую архитектуру, где шины данных и команд разделены. При этом память данных содержит данные, а память команд содержит как команды, так и данные. Процессор содержит ОЗУ и/или ПЗУ на кристалле (кроме ADSP-2100), так что часть адресного пространства памяти находится в нем. Быстродействие памяти на кристалле позволяет процессору в течение цикла считать 2 операнда из памяти данных и команду из памяти команд.

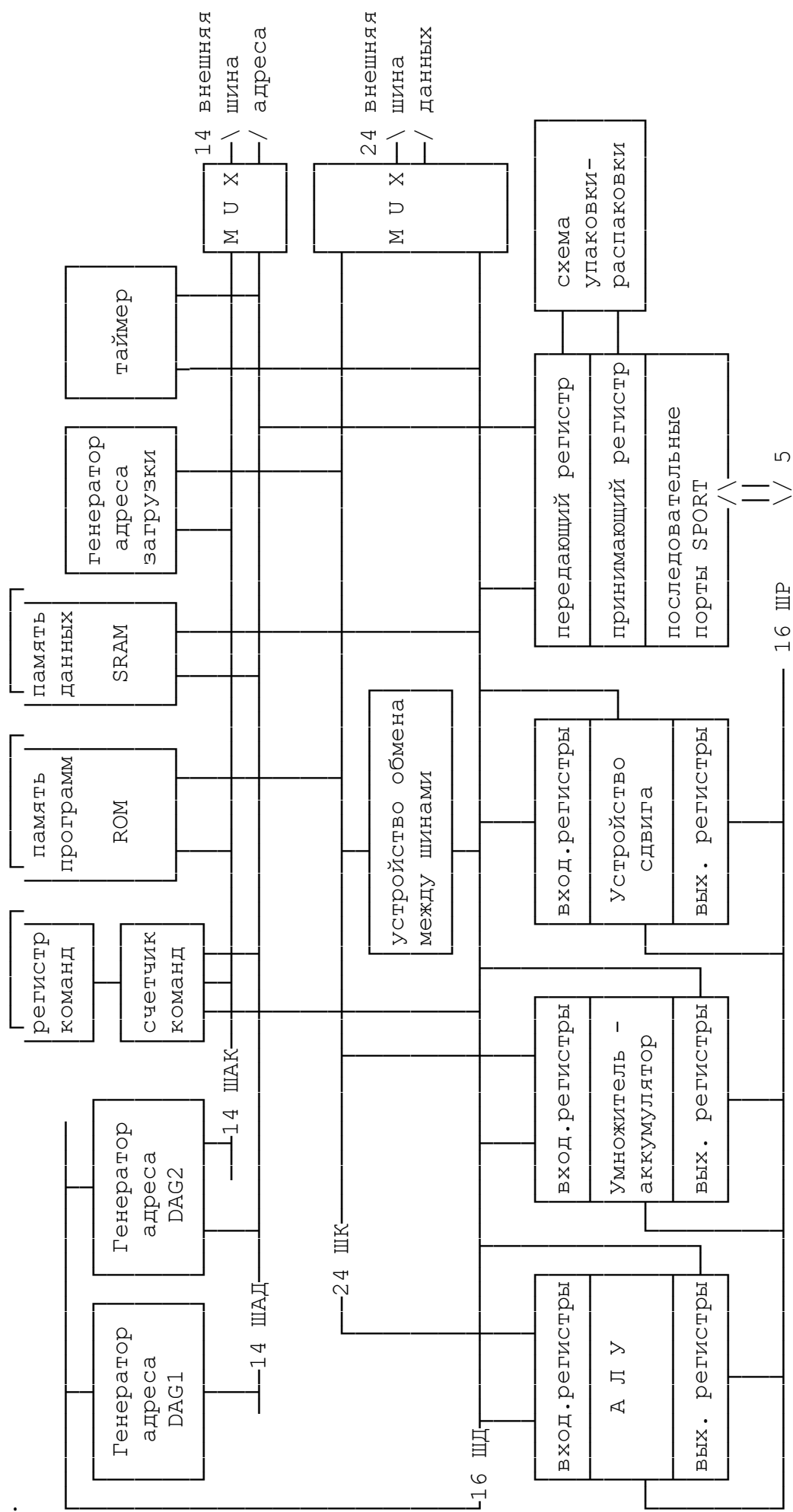


Рис. 12.2. Внутренняя архитектура процессоров семейства ADSP-21XX.

Внутренние устройства МП связываются пятью шинами. 14 - разрядная шина адреса данных (ШАД) служит для указания адресов данных и обеспечивает доступ к 16 Кб данных. 16 - разрядная шина данных (ШД) обеспечивает пересылку содержимого любого регистра в любой регистр или в память/ из памяти в течение одного цикла. Адрес памяти данных формируется из абсолютного значения, записанного в инструкции (абсолютная адресация), или из генератора адресов данных (косвенная адресация). Только косвенная адресация возможна для адресации данных в памяти программ. 14 - разрядная шина адреса команд (ШАК) обеспечивает доступ к 16 Кб команд и данных. 24 - разрядная шина команд (ШК) обеспечивает загрузку 24 - битного кода команды. 16 - разрядная внутренняя шина результата (ШР) используется для обмена данными между тремя вычислительными устройствами.

В процессорах, которые имеют внутреннюю память, внутренняя шина адреса памяти команд (ШАК) и внутренняя шина адреса памяти данных (ШАД) мультиплексированы в единую шину адреса, а внутренняя шина данных памяти команд (ШК) и внутренняя шина данных памяти данных (ШД) мультиплексированы в единую шину данных. Эти шины выведены на внешние выводы кристалла.

Процессор содержит три полнофункциональных независимых вычислительных блока: арифметико - логическое устройство, умножитель - аккумулятор и устройство сдвига. Эти три устройства обеспечивают выполнение трех операций с 16 - разрядными словами и имеют аппаратную поддержку для работы с числами повышенной точности. АЛУ осуществляет стандартные арифметические и логические операции; умножитель -

аккумулятор производит одноцикловое умножение, умножение с суммированием и умножение с вычитанием; устройство сдвига выполняет операции арифметического и логического сдвига, нормализацию, денормализацию и действия с экспонентой. Каждое из перечисленных устройств включает в себя двойной набор входных регистров, которые могут загружаться с шины данных и команд, а также выходные регистры.

Результаты работы любого устройства могут быть операндами любого другого устройства в следующем цикле. Шина внутренних результатов (ШР) прямо соединяет вычислительные устройства с этой целью.

Генераторы адреса (DAG1, DAG2) позволяют одновременно выбирать два операнда и по взаимодействию с счетчиком команд способствуют эффективному выполнению команд. Все команды одноцикловые. 24 - битовые командные слова допускают высокую степень параллелизма. Возможно выполнение трех операций за один командный цикл.

Счетчик команд формирует адреса инструкций для памяти программ. Он управляет регистром инструкций, который содержит исполняемую в данный момент команду. Регистр команд буферизирует исполнение программы. Команды загружаются в регистр команд в течение одного цикла, а исполняются в течение следующего, одновременно с загрузкой следующей команды. Чтобы минимизировать циклы ожидания, счетчик команд выполняет условные переходы, вызовы и возвраты из подпрограмм за один цикл. Он имеет внутренний счетчик вложенностей циклов и стек циклов, что позволяет выполнять циклы без потерь времени.

12.2.2 Периферийные устройства

Процессоры семейства ADSP-21XX содержат на кристалле периферийные устройства, обеспечивающие работу процессора и связь с внешними устройствами.

Программируемый интервальный таймер обеспечивает периодическую генерацию прерываний. 8-битный масштаб позволяет ему декрементировать содержимое регистра - счетчика в диапазоне от каждого цикла до каждого 256-го цикла процессора. Прерывание генерируется когда регистр - счетчик обнуляется.

Большинство процессоров семейства имеют 2 двунаправленных последовательный порта (SPORT) с двойной буферизацией. Эти порты используют синхронную передачу данных и кадровые сигналы, чтобы контролировать поток данных. Каждый порт может тактироваться от внутреннего таймера или от внешней частоты. Сигналы кадровой синхронизации могут быть сгенерированы самим портом, так и получены извне. Длина слова может меняться от 3 бит до 16.

Процессоры серии ADSP-21msp5x содержат в своем составе аналоговый интерфейс. Он состоит из входных усилителей и 16-битных сигма - дельта АЦП и ЦАП, а также дифференциального входного усилителя. Набор регистров, отображенных на память процессора служат для управления операциями аналоговой части и для передачи данных между аналоговой частью и остальными устройствами процессора.

Каждый из процессоров семейства поддерживает один или более входов для внешних прерываний. Внешние прерывания имеют свой уровень приоритета, могут быть индивидуально замаскированы. Процессор имеет отдельный вход прерывания IRQ2, а IRQ0 и IRQ1 могут быть сконфигурированы на входах порта SPORT1.

12.2.3 Набор команд

В системе команд ADSP-21XX используются 24-битные команды, которые исполняются за один цикл. Исключением являются команды ожидания прерывания, команды, для выполнения которых требуется 2 обращения к внешней памяти, и если доступ к памяти требует циклов ожидания. Язык ассемблера использует математическую мнемонику для простоты написания и читаемости программ.

Например:

$MX0=1234;$ - запись в регистр $MX0$ значения 1234

$DM(0x3FFE)=AX0;$ - копирование содержимого регистра $AX0$ в ячейку памяти с адресом $0x3FFE$

$JUMP(I4);$ - косвенный переход по адресу, хранящемуся в $I4$

Для использования параллелизма процессора в системе команд присутствуют многофункциональные команды, которые обеспечивают комбинацию пересылок данных, чтения - записи памяти и вычислений за один цикл. Например:

$MR=MR+MX0*MY(SS), MX0=DM(I0, M0), MY0=PM(I4, M5);$

┌—1 часть—┐ ┌—2 часть—┐ ┌—3 часть—┐

Первая часть этой многофункциональной инструкции суммирует предыдущее значение регистра MR с произведением регистров $MX0$ и $MY0$, причем оба операнда считаются знаковыми (SS). Вторая и третья части загружают два новых операнда. Один из них считывается из памяти данных (DM), указатель на данные находится в индексном регистре $I0$, после загрузки происходит пост - модификация указателя значением, содержащимся в регистре $M0$. Другой считывается из памяти инструкций (PM), указатель

на данные находится в индексном регистре I4, после загрузки происходит пост - модификация указателя значением, содержащимся в регистре M5.

Арифметические операции могут быть включены в условные операторы:
IF AC AR=AX0+AY0+C;

Условное выражение IF AC (которое может отсутствовать) проверяет флаг переноса АЛУ (AC). Если предыдущая команда сгенерировала флаг переноса, выполняется данная арифметическая команда, в противном случае происходит выполнение команда NOP и исполнение переходит к следующей команде. Алгебраическое выражение AR=AX0+AY0+C означает, что содержимое регистра AR будет содержать AX0 и AY0 плюс значение флага переноса.

Команда IDLE заставляет процессор ожидать при пониженном потреблении энергии сигнала прерывания.

К средствам разработки и отладки устройств на основе процессоров семейства ADSP-21XX относятся как программные (системный конфигуратор, ассемблер, редактор связей, ПЗУ-кодер, симулятор и Си-компилятор), так и аппаратные средства (различные эмуляторы и т.п.).

Процессоры рассматриваемого семейства находят широкое применение в системах управления, сжатия аудиоданных, сотовой телефонии, модемах, медицинской ультразвуковой аппаратуре, системах распознавания и обработки речи, а также во многих других областях электронной индустрии.