

Лекция 5.

II. Комбинационные логические устройства

Комбинационная схема – логическая схема, сигнал на выходе которой определяется только уровнями сигналов на ее входах. Такой подход построения схем называется *комбинационной логикой*.

1. Шифратор – устройство, которое преобразует входной позиционный код в выходной двоичный.

Номер позиции								Выходной код		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	F_2	F_1	F_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

ИВ		С	
I_0	I_1		
I_2			
I_3			
I_4			
I_5			F_2
I_6			F_1
I_7			F_0

C – coder

$$F_2 = \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} \overline{I_1} \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} \overline{I_1} I_0 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} I_1 \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} I_1 I_0$$

$$F_1 = \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} I_1 \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} \overline{I_2} I_1 I_0 + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2 \overline{I_1} \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2 I_1 \overline{I_0}$$

$$F_0 = \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2 \overline{I_1} \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} \overline{I_3} I_2 I_1 \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} I_3 \overline{I_2} \overline{I_1} \overline{I_0} + \overline{I_7} \overline{I_6} \overline{I_5} \overline{I_4} I_3 I_2 \overline{I_1} \overline{I_0}$$

Приоритетный шифратор

Вход				Выходной код	
I_3	I_2	I_1	I_0	F_1	F_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

$$F_1 = \overline{I_3} \overline{I_2} \overline{I_1} + \overline{I_3} \overline{I_2} I_1 + I_3 \overline{I_2} \overline{I_1} + I_3 \overline{I_2} I_1 + I_3 I_2 \overline{I_1} + I_3 I_2 I_1$$

$$F_0 = \overline{I_3} \overline{I_2} I_1 + I_3 \overline{I_2} \overline{I_1} + I_3 \overline{I_2} I_1 + I_3 I_2 \overline{I_1} + I_3 I_2 I_1$$

После минимизации

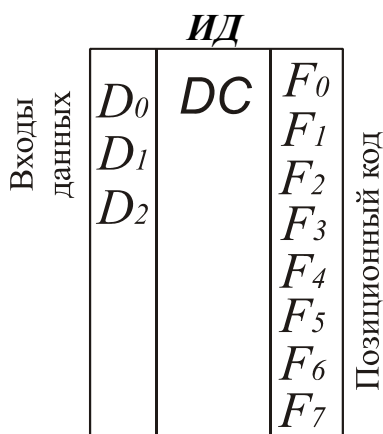
$$F_1 = I_3 + I_2 \quad F_0 = I_3 + \overline{I_2} I_1$$

I_3	I_2	I_1	F_1	F_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

$\overline{I_2} \overline{I_1}$	$I_2 \overline{I_1}$	$I_2 I_1$	$\overline{I_2} I_1$	F_1
1	1	1	1	I_3
0	1	1	0	$\overline{I_3}$

$\overline{I_2} \overline{I_1}$	$I_2 \overline{I_1}$	$I_2 I_1$	$\overline{I_2} I_1$	F_0
1	1	1	1	I_3
0	0	0	1	$\overline{I_3}$

2. Дешифратор – устройство, которое преобразует входной двоичный в выходной позиционный код.



DC – decoder

В любой момент времени дешифратор активизирует только *один* из *n* выходов. Выбор осуществляется при помощи управляющих сигналов, поступающих на соответствующие входы.

$$F_0 = \overline{x_2 x_1 x_0}$$

$$F_1 = \overline{x_2} x_1 x_0$$

$$F_2 = x_2 \overline{x_1} x_0$$

$$F_3 = x_2 x_1 \overline{x_0}$$

$$F_4 = x_2 \overline{x_1} \overline{x_0}$$

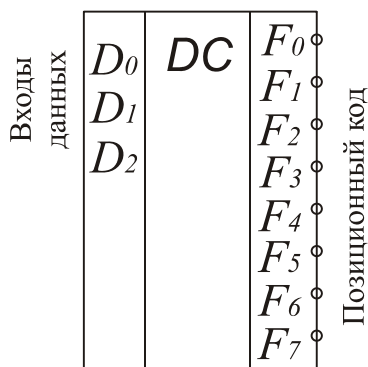
$$F_5 = x_2 x_1 \overline{x_0}$$

$$F_6 = x_2 x_1 x_0$$

$$F_7 = x_2 x_1 x_0$$

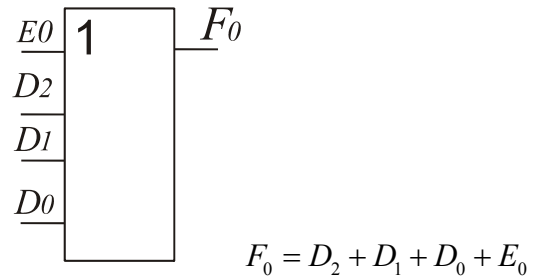
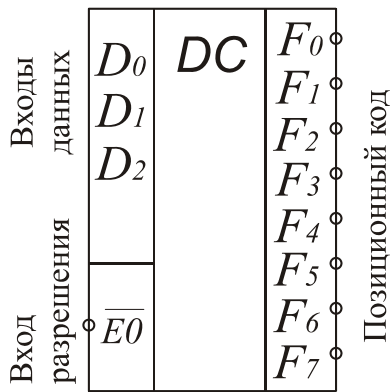
Входной код			Состояния на выходах							
x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Активное состояние часто определяется низким уровнем напряжения “0”, обозначается как

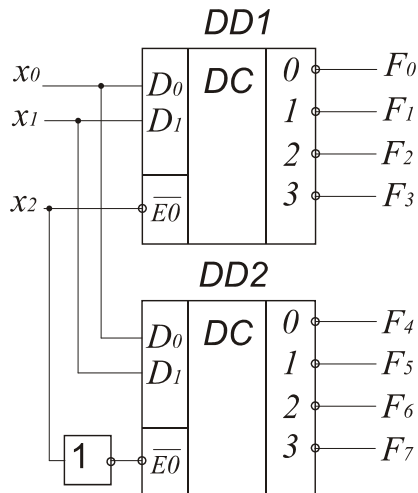


Дешифратор с входом разрешения

Входной код				Состояния на выходах							
$E0$	x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
1	x	x	x	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0



Наличие входа разрешения позволяет увеличить разрядность дешифратора.

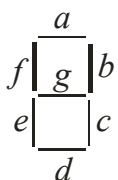


		Входной код			Состояния на выходах							
		x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
DD1	0	0	0	0	0	1	1	1	1	1	1	1
	1	0	0	1	1	0	1	1	1	1	1	1
	2	0	1	0	1	1	0	1	1	1	1	1
	3	0	1	1	1	1	1	0	1	1	1	1
DD2	4	1	0	0	1	1	1	1	0	1	1	1
	5	1	0	1	1	1	1	1	1	0	1	1
	6	1	1	0	1	1	1	1	1	1	0	1
	7	1	1	1	1	1	1	1	1	1	1	0
					DD1				DD2			

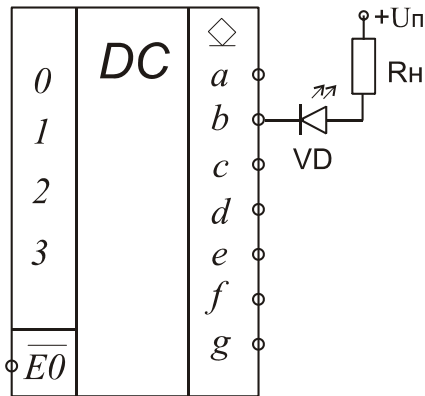
Неполные дешифраторы

Дешифраторы 1:10 иногда называют преобразователями двоично-десятичного кода в десятичное число.

		Входной код				Состояния на выходах									
		x_3	x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
DC	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
	2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
	3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
	4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
	5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
	6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
	7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
	8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
	9	1	0	0	1	1	1	1	1	1	1	1	1	1	0



Семисегментные дешифраторы используются для преобразования двоичного кода в семисегментный код. Семисегментный код необходим для отображения на цифровых индикаторах значений цифр от 0 до 9. Семисегментный, потому что цифры отображаются так называемыми сегментами, которых семь штук. Как правило, такие элементы с открытым коллектором и инверсным выходом

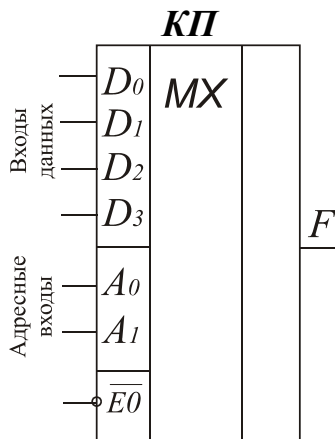


Входной код					Состояния на выходах						
dec	x_3	x_2	x_1	x_0	F_a	F_b	F_c	F_d	F_e	F_f	F_g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

+Up может быть до 60 В.

Лекция 6.

3. Мультиплексоры – схемы выборки с электронным управлением. **Мультиплексор** подключает один из n входных сигналов к *единственной* выходной линии. Выбор подключаемого входа осуществляется при помощи адреса, передаваемого по специальным линиям. Мультиплексоры иногда называют *селекторами данных*.

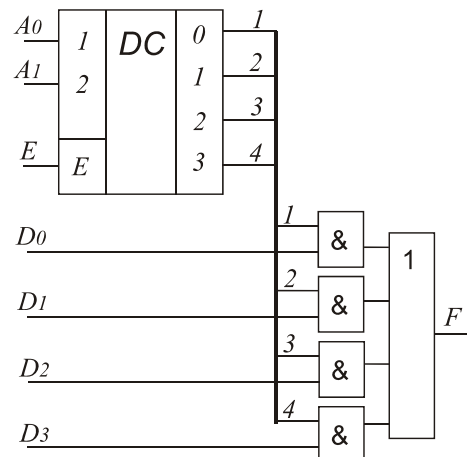
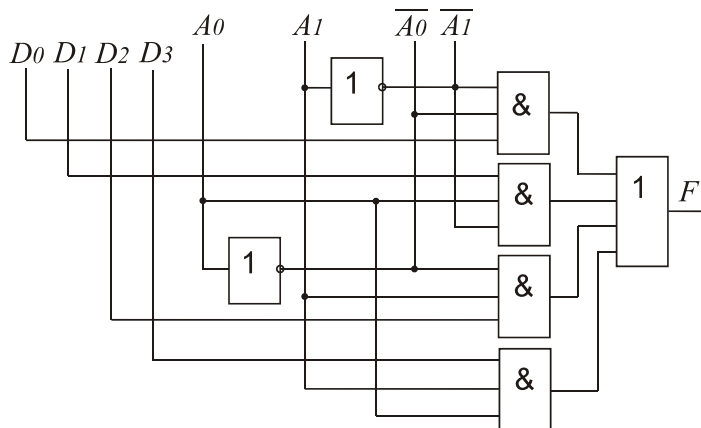


Обозначение: ***MX, MS, MUX***

A_1 и A_0 — входные линии адреса, D_3, D_2, D_1 и D_0 — входные информационные линии. При наличии активного разрешающего сигнала E на адресные линии подается двоичный код адреса. При этом на выход F будет копироваться информация с выбранного в соответствии с этим адресом информационного входа. Так, если $A_1A_0 = 00_2 = 0_{10}$, на выход F подается информация с линии D_0 ; если $A_1A_0 = 01_2 = 1_{10}$, то с линии D_1 , если $A_1A_0 = 10_2 = 2_{10}$, то с линии D_2 ; а при $A_1A_0 = 11_2 = 3_{10}$ - с линии D_3 . Таким образом, таблицу истинности данного мультиплексора можно представить в виде таблицы:

E_0	A_1	A_0	F
1	x	x	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

$$F = D_0 \overline{A_1} \overline{A_0} + D_1 \overline{A_1} A_0 + D_2 A_1 \overline{A_0} + D_3 A_1 A_0$$

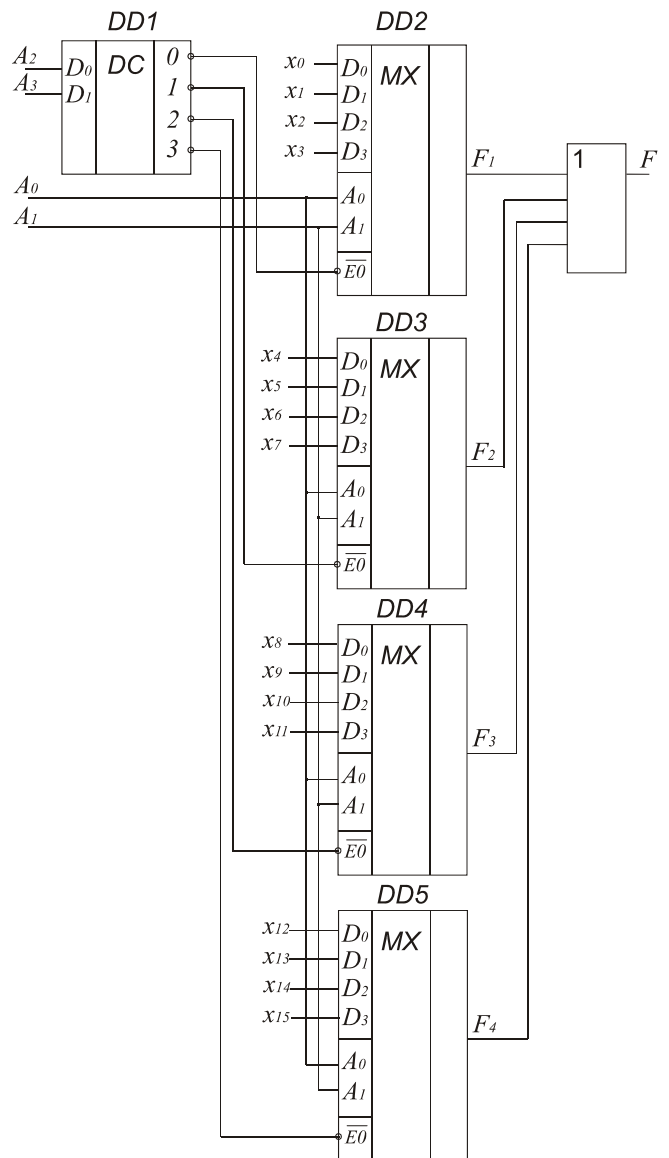
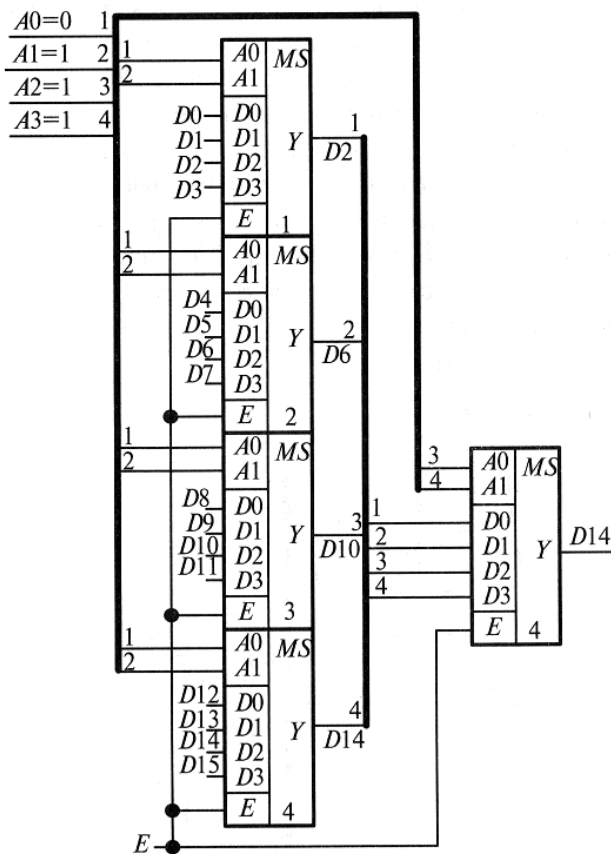


Схему мультиплексора можно упростить, используя дешифратор. Дешифратор DC осуществляет выбор информационного входа в соответствии с адресом. Он обеспечивает логическую единицу на входе логического элемента $И$, соответствующего адресу выбранного информационного входа. При наличии разрешающего сигнала $E=1$ состояние этого логического элемента $И$ определяется информацией на выбранном информационном входе. Так, при $A1A0 = 10_2 = 2_{10}$ на выход F будет передаваться информация с входа $D2$.

В случае, когда необходимо коммутировать большее количество линий, чем позволяет имеющийся мультиплексор, используют следующие *методы увеличения разрядности мультиплексора*.

Каскадное соединение мультиплексоров (последовательное соединение). Для выбора одного информационного входа из 16 ($16 = 2^4$) необходимы 4 входные линии адреса: $A3, A2, A1$ и $A0$. Четыре базовых мультиплексора «1 из 4» обеспечивают выбор в зависимости от кода, поданного на $A1$ и $A0$, соответственно: первый — одного из сигналов $D0, D1, D2$ или $D3$, второй — из сигналов $D4, D5, D6$ или $D7$, третий — из сигналов $D8, D9, D10$ или $D11$, четвертый — из сигналов $D12, D13, D14$ или $D15$. Пятый мультиплексор обеспечивает выбор одного из этих ранее выбранных сигналов в зависимости от кода, подаваемого на $A3$ и $A2$.

На входы разрешения работы E всех мультиплексоров должен быть подан активный входной уровень. Могут использоваться мультиплексоры без входа разрешения.



Каскадное соединение мультиплексоров «1 из 4»
Увеличение разрядности мультиплексоров с помощью дешифратора (справа).

Например, при подаче $A_3A_2A_1A_0 = 1110_2 = 14_{10}$ на адресные входы первых четырех мультиплексоров будет поступать двоичный код $A_1A_0 = 10_2 = 2_{10}$. Поэтому первая часть каскада будет выбирать входные информационные линии D_2, D_6, D_{10}, D_{14} . Они подключены соответственно ко входам D_0, D_1, D_2 и D_3 пятого мультиплексора. Поскольку на его адресные линии приходит код $A_3A_2 = 11_2 = 3_{10}$, на выход каскада будет передаваться информация с выхода D_3 пятого выходного мультиплексора, т. е. информация с входной информационной линии D_{14} .

Параллельное соединение мультиплексоров реализуется с помощью дешифратора (рис. справа). Мультиплексоры при этом должны иметь входы разрешения. Для выбора одного информационного входа из 16 необходимы 4 входные линии адреса: A_3, A_2, A_1 и A_0 . Старшие разряды адреса A_3 и A_2 подаются на входы данных дешифратора, который выбирает, какой из мультиплексоров DD_2-DD_5 будет *работать*. На выходах остальных мультиплексоров независимо от значений переменных x будет «0». Активный мультиплексор выбирает в зависимости от кода, поданного на A_1 и A_0 , соответственно: первый — один из сигналов x_0, x_1, x_2 или x_3 , второй — из сигналов x_4, x_5, x_6 или x_7 , третий — из сигналов x_8, x_9, x_{10} или x_{11} , четвертый — из сигналов x_{12}, x_{13}, x_{14} или x_{15} . Сигналы с выходов мультиплексоров суммируются.

Например, при подаче $A_3A_2A_1A_0 = 1110_2 = 14_{10}$ на входы данных DC будет приходить код $A_3A_2 = 11_2 = 3_{10}$, следовательно, будет разрешена работа только мультиплексора DD_5 . На выходах мультиплексоров $DD_2 - DD_4$ будет логический «0», т.е. $F_1=F_2=F_3=0$. На адресные входы всех мультиплексоров поступает двоичный код $A_1A_0 = 10_2 = 2_{10}$, то на выход F_4 DD_5 будет передаваться информация с входа D_3 , т.е. значение переменной x_{14} . После суммирования на выходе всей схемы будет $F=F_1+F_2+F_3+F_4 = 0 + 0 + 0 + x_{14} = x_{14}$.

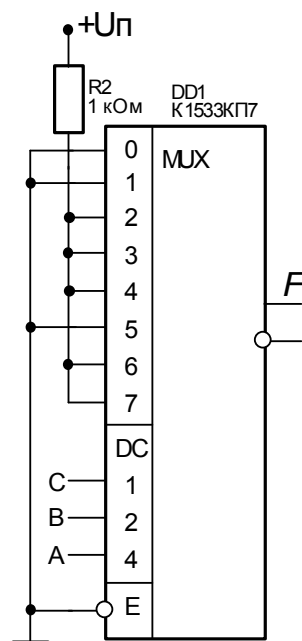
Реализация логических функций с помощью мультиплексора.

Иногда мультиплексоры используются для реализации логических функций. Для этого переменные подаются на адресные линии мультиплексора, а на входах данных устанавливается высокий или низкий уровень в соответствии с таблицей истинности.

Пример 1. Функция 3-х переменных.

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

На основе мультиплексора К1533КП7



Пример 2. Функция 4-х переменных.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Данную функцию можно реализовать с помощью одного мультиплексора с 16 входами, например, К155КП1 или с помощью 2-х мультиплексоров с 8 входами, например, К155КП7.

К155 КП1 (24 ножки)

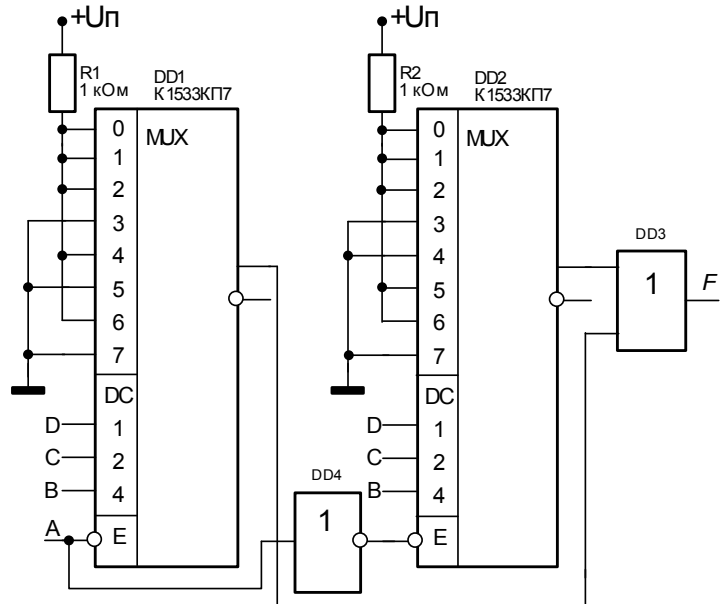
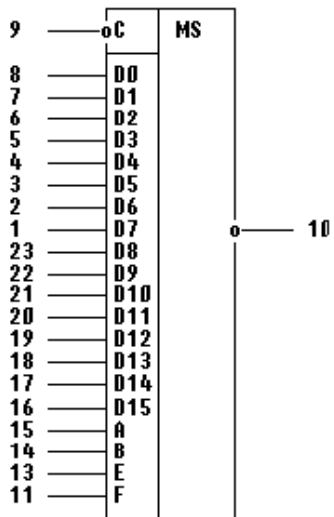
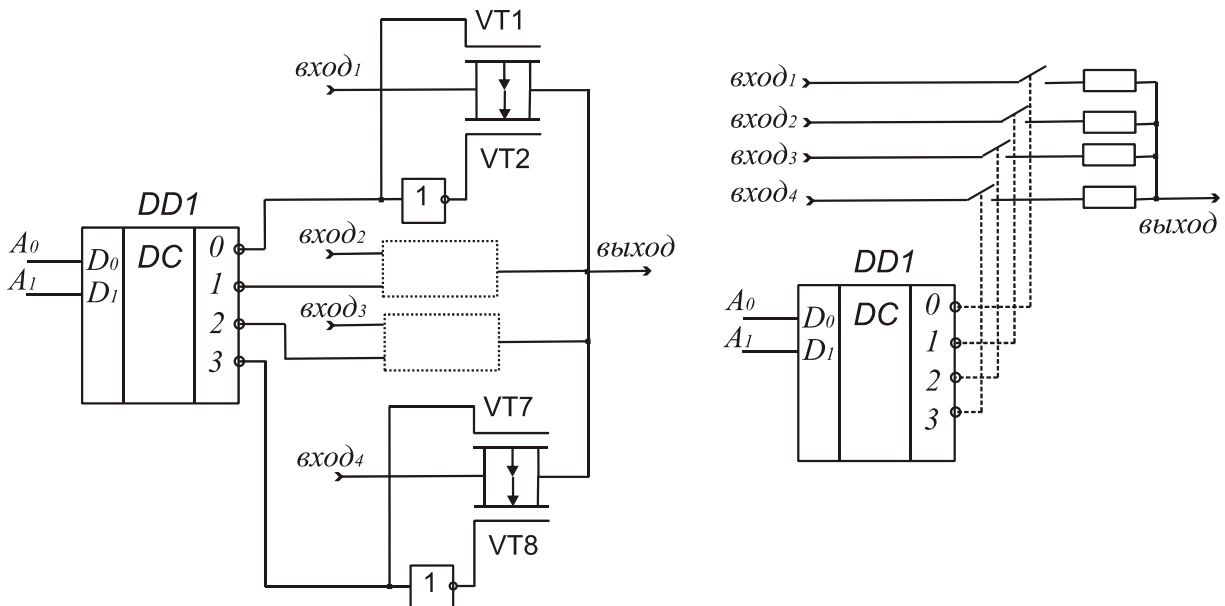


Схема мультиплексора КМОП (внутренняя структура)



Схемотехника мультиплексоров КМОП отличается тем, что они могут коммутировать аналоговые сигналы с размахом амплитуды до U_p . Пара транзисторов VT1-VT2 и др. делается для того, чтобы обеспечить одинаковое сопротивление в обоих направлениях, т.е. чтобы сигнал мог распространяться как от входа к выходу, так и от выхода к входу. В этом случае мультиплексор может использоваться в качестве демультимплексора.

4. Демультимплексор - устройство, функционально противоположное мультиплексору. Он подключает *единственный* входной сигнал к одному из n выводов, определяемому адресом.

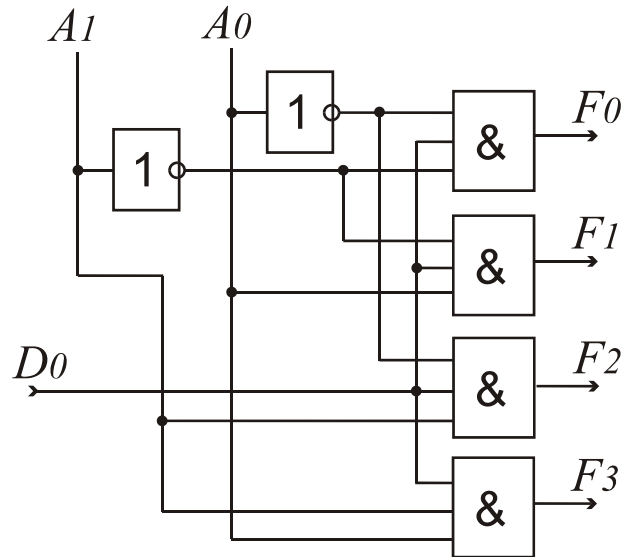
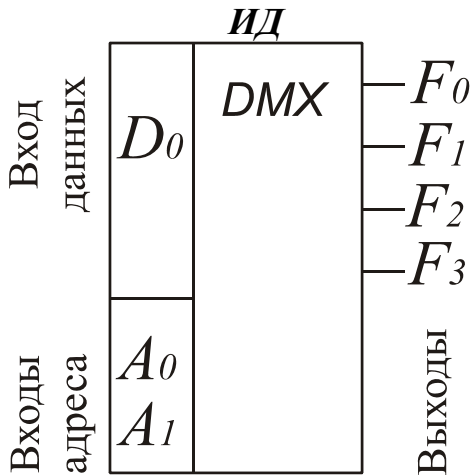
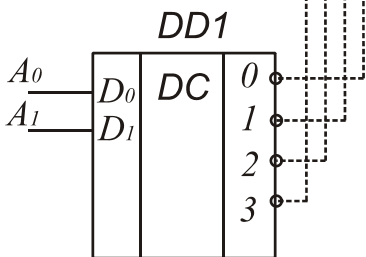
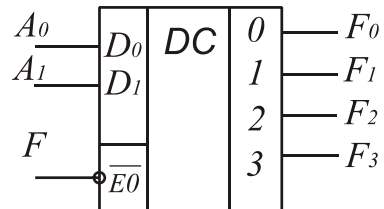
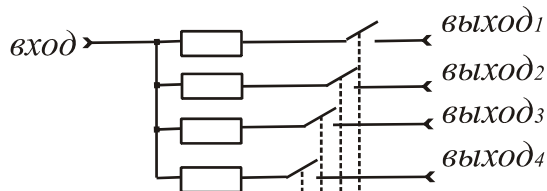


Таблица истинности демультимплексора

A_1	A_0	D_0	F_0	F_1	F_2	F_3
0	0	F	F	0	0	0
0	1	F	0	F	0	0
1	0	F	0	0	F	0
1	1	F	0	0	0	F

$$F_0 = D_0 \overline{A_1} \overline{A_0} \quad F_1 = D_0 \overline{A_1} A_0$$

$$F_2 = D_0 A_1 \overline{A_0} \quad F_3 = D_0 A_1 A_0$$



$\overline{E_0}$	A_1	A_0	F_0	F_1	F_2	F_3
F	0	0	\overline{F}	0	0	0
F	0	1	0	\overline{F}	0	0
F	1	0	0	0	\overline{F}	0
F	1	1	0	0	0	\overline{F}

В серии КМОП в качестве демультимплексора можно использовать мультиплексор. Данные подаются на выход, а выходные функции снимаются с входов.

В серии ТТЛ в качестве демультимплексора применяются дешифраторы с входом разрешения. В этом случае входные данные подаются на вход разрешения.

Лекция 7.

5. Полусумматоры и сумматоры предназначены для арифметического сложения двух переменных.

Классификация сумматоров

Классификация сумматоров может быть выполнена по различным признакам. Рассмотрим наиболее часто встречающиеся из них.

По числу выводов различают: полусумматоры, одноразрядные сумматоры, многоразрядные сумматоры.

Полусумматором называется устройство, предназначенное для сложения двух одноразрядных переменных, имеющее два входа и два выхода и формирующее из сигналов входных слагаемых сигналы суммы и переноса в старший разряд.

Одноразрядным сумматором называется устройство, предназначенное для сложения двух одноразрядных кодов (переменных), имеющее три входа и два выхода, и формирующее из сигналов входных слагаемых и сигнала переноса из младших разрядов сигналы суммы и переноса в старший разряд.

Многоразрядным сумматором называется устройство, предназначенное для сложения двух многоразрядных кодов, формирующее на выходе код суммы и сигнал переноса в случае, если результат сложения не может быть представлен кодом, разрядность которого совпадает с разрядностью кодов слагаемых.

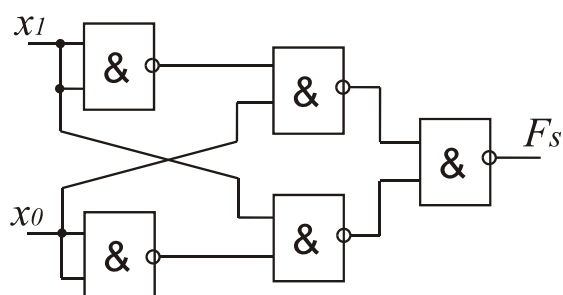
В свою очередь, многоразрядные сумматоры подразделяются на *последовательные* и *параллельные*. В *последовательных* сумматорах операция сложения выполняется последовательно разряд за разрядом, начиная с младшего. В *параллельных* сумматорах все разряды входных кодов суммируются одновременно.

Различают *комбинационные* сумматоры — устройства, не имеющие собственной памяти, и *накапливающие* сумматоры, снабженные собственной внутренней памятью, в которой аккумулируются результаты выполненной операции. При этом каждое очередное слагаемое прибавляется к уже имевшемуся в устройстве значению.

По способу тактирования различают *синхронные* и *асинхронные* сумматоры. В *синхронных* сумматорах время выполнения операции арифметического суммирования двух кодов не зависит от вида самих кодов и всегда остается постоянным. В *асинхронных* сумматорах время выполнения операции зависит от вида слагаемых. Поэтому по завершении выполнения суммирования необходимо вырабатывать специальный сигнал завершения операции.

В зависимости от используемой системы счисления различают *двоичные*, *двоично-десятичные* и другие типы сумматоров.

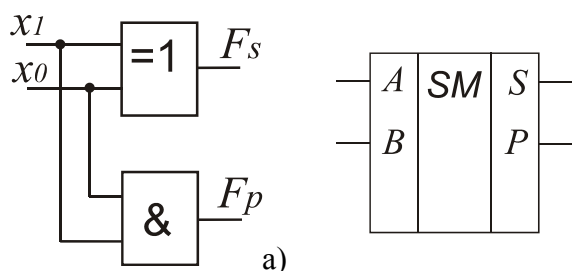
Полусумматор



x_1	x_0	F_s	F_p
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$F_s = \overline{x_1}x_0 + x_1\overline{x_0} = x_1 \oplus x_0$$

$$F_p = x_1x_0$$



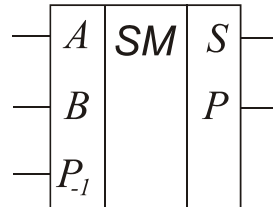
Полусумматор (а) и его графическое обозначение (б)

Время суммирования будет определяться временем выполнения операции «Исключающее ИЛИ» $t_{\Sigma} = 3t_{3,p}$, где $t_{3,p}$ - задержка распространения элемента ИЛИ-НЕ. Например для серии K155 $t_{3,p} = 9$ нс, то $t_{\Sigma} = 27$ нс.

Полусумматор применим только для сложения одноразрядных двоичных чисел или младших разрядов многоразрядных слов. Слово – величина, измеряемая в битах и равная разрядности данных, обрабатываемых цифровым устройством.

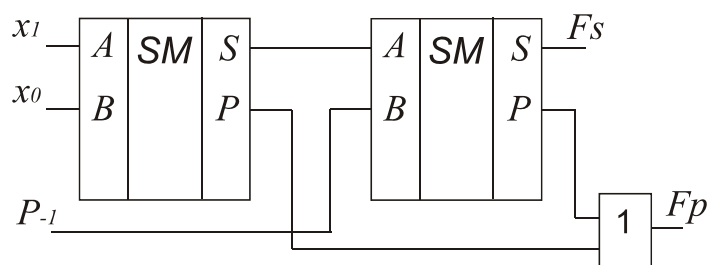
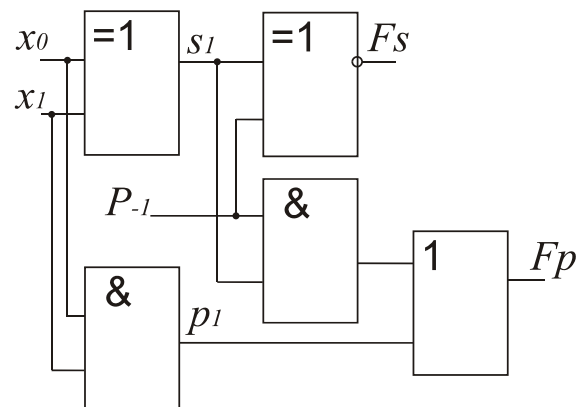
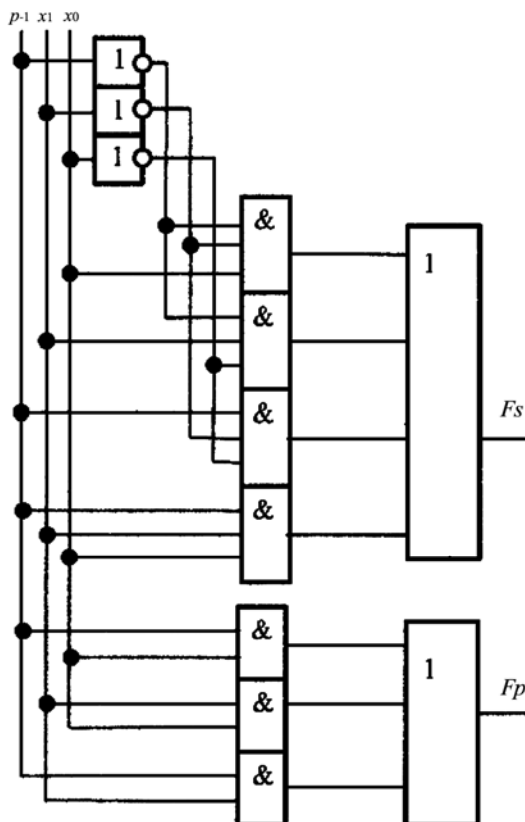
Таблица сложения старших разрядов многоразрядных двоичных слов должна быть дополнена переменной возможного переноса из более младшего разряда (p_{-1}).

x_1	x_0	p_{-1}	Fs	Fp
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1



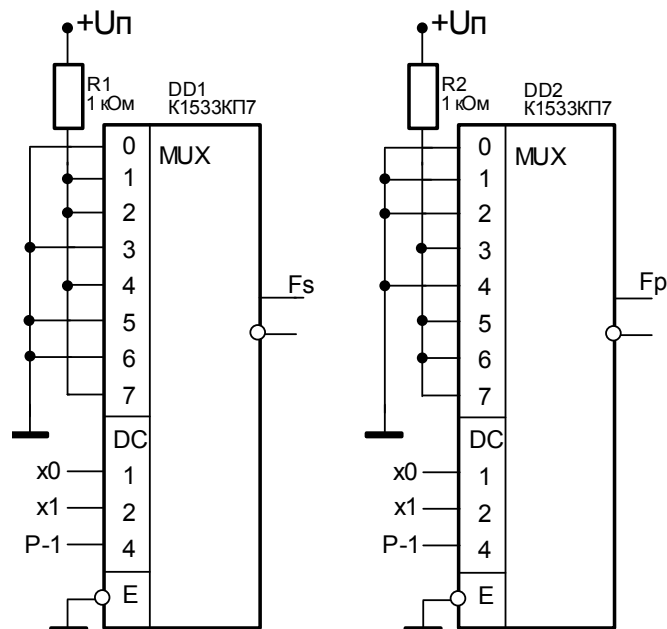
$$Fs = (\overline{x_1 x_0} + x_1 \overline{x_0}) p_{-1} + (\overline{x_1 x_0} + x_1 \overline{x_0}) p_{-1} = (x_1 \oplus x_0) \overline{p_{-1}} + (x_1 \oplus x_0) p_{-1}$$

$$Fp = x_0 x_1 + (\overline{x_1 x_0} + x_1 \overline{x_0}) p_{-1} = x_0 x_1 + (x_1 \oplus x_0) p_{-1}$$

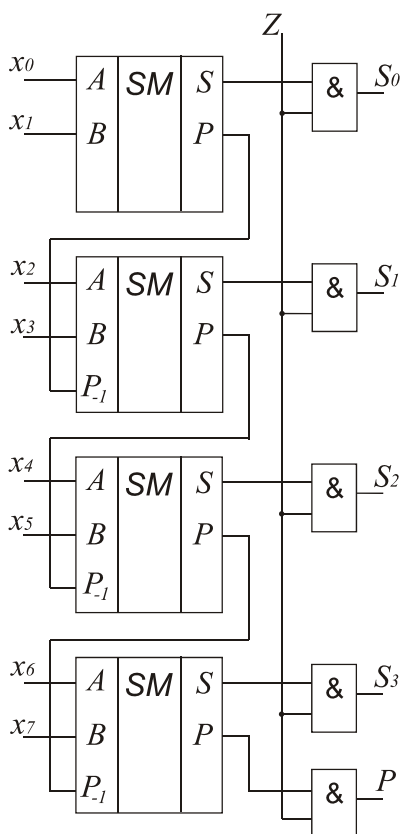


Время суммирования полного сумматора: $t_{0\Sigma} = 2 * t_{\Sigma} = 6t_{3,p}$
 Формирование сигнала переноса $t_p = 5t_{3,p}$.

Полный сумматор может быть реализован с использованием мультиплексора.



Многоразрядные сумматоры



Рассмотрим типовую структуру 4-разрядного сумматора, выполненного с использованием трех одноразрядных сумматоров и одного полусумматора. Разряды кодов слагаемых подаются на соответствующие входы сумматоров, выходы суммы которых подсоединяются к первым входам ЛЭ И, используемых в качестве выходных ключей, на вторые входы которых подается сигнал Z , определяющий момент считывания результата. Выход сигнала переноса сумматора нулевого разряда подается на вход переноса сумматора первого разряда и т. д. Для получения на выходе сигнала, равного реальной сумме входных кодов, необходимо, чтобы сигнал переноса последовательно сформировался на выходах сумматоров всех разрядов. Следовательно, не зависимо от того, что для суммирования в каждом разряде используется отдельный сумматор, реальное время выполнения операции в данной схеме определяется последовательным переносом сигнала P из разряда в разряд.

Для исключения получения ложного результата на выходе схемы установлены элементы И. Сигнал Z на входах этих элементов И должен появляться не ранее, чем после последовательной передачи сигнала переноса по всем разрядам сумматоров.

Общее время формирования сигнала результата в рассматриваемой схеме $t_{0\gamma} = t_{3,p} [6 + 2(n-1)] = 12t_{3,p}$, n — разрядность кодов слагаемых.

Выходной сигнал переноса, именуемый также сигналом переполнения, формируется за время $t_p = 11t_{3,p}$.

Схему 2-х разрядного полного сумматора на основе мультиплексов типа КП2 реализовать самостоятельно (либо на практическом занятии).

представленных в прямом двоичном коде. Наиболее часто для записи отрицательного числа используется *дополнительный код*.

Алгоритм получения дополнительного кода двоичного числа:

1. Записывают обратный код исходного числа, для чего все его разряды инвертируют (заменяют дополнениями);

2. К полученному после инвертирования коду добавляют единицу.

При сложении полученного таким образом кода вычитаемого с кодом уменьшаемого будет реализована операция вычитания. При этом знак результата определяется старшим разрядом полученного кода. Если старший разряд равен нулю, то получено *положительное* число, представленное в прямом коде. Если старший разряд равен единице, получено *отрицательное* число, представленное в дополнительном коде.

Операция увеличения какого-либо числа на единицу называется *инкрементом*, уменьшения на единицу - *декрементом*.

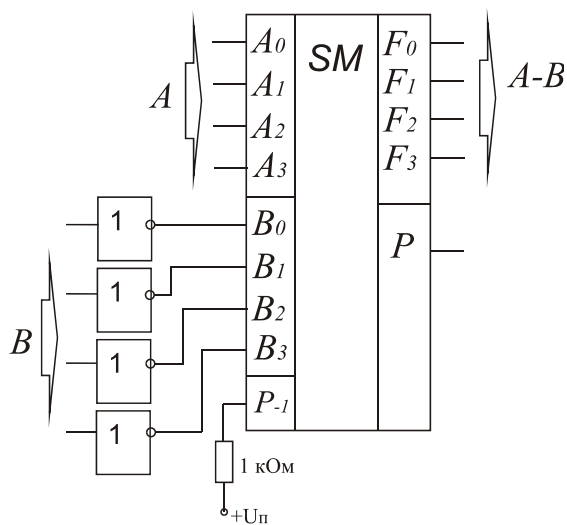
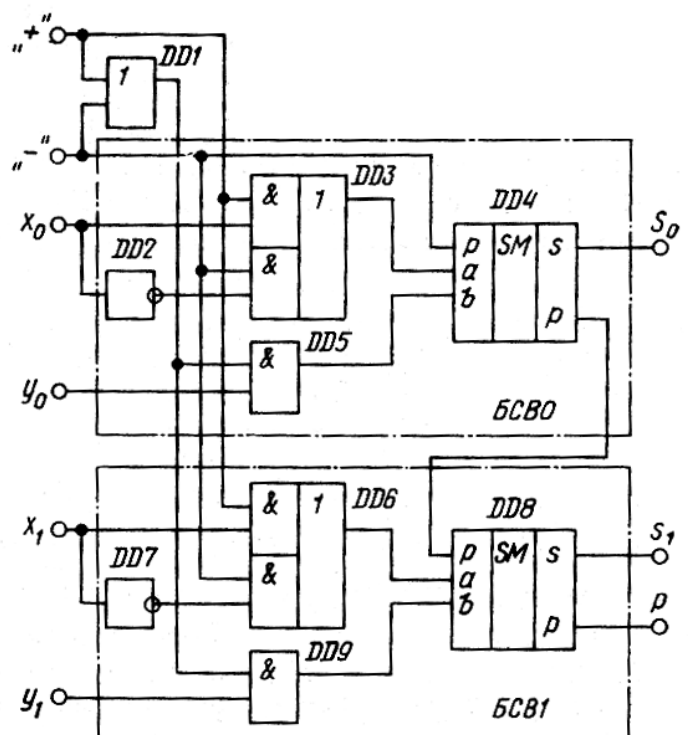


Схема аппаратной реализации вычитания в дополнительном коде.

Схема сложения-вычитания 2-разрядных кодов (БСВ – блок сложения-вычитания) →



Пример. Используя дополнительный код, найти разность $31 - 12$.

Решение. 1. Прямые двоичные коды заданных чисел: $31_{10} = 00011111_2$; $12_{10} = 00001100_2$.

2. Обратный код вычитаемого 11110011 .

3. Дополнительный код вычитаемого: $11110011 + 00000001 = 11110100$.

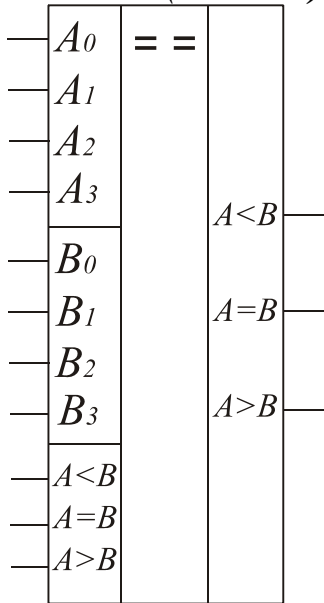
4. Сложение: $00011111 + 11110100 = 100010011$

Полученный результат представлен 9-разрядным двоичным кодом. В этом случае самый старший девятый разряд отбрасывают. Полученное двоичное число содержит нуль в старшем разряде. Поэтому результат положителен и представлен в прямом коде. Его десятичный эквивалент равен 19.

Лекция 7. Схемы сравнения (цифровой компаратор) - предназначены для сравнения двух двоичных чисел (compare - сравнивать).

Цифровой компаратор может использоваться, например, в системах автоматического контроля и регулирования. При этом первое число А является параметром некоторого процесса, а второе число В - порогом (уставкой), которого (в соответствии с условиями задачи) число А не должно превосходить или опускаться ниже.

1533СП1 (561ИИП2)



Компаратор имеет две группы входов. На одну из них поступают разряды первого числа (А), на другую группу - разряды второго числа (В). Три выхода компаратора появлением логической «1» фиксируют результат сравнения. На одном выходе она устанавливается при равенстве чисел (А=В), на другом - при А<В, на третьем - при А>В.

Значение на входах А=В, А<В, А>В влияют на результат сравнения, только если $A_3A_2A_1A_0 = B_3B_2B_1B_0$. При этом, если на вход А=В подана «1», то $F_{A=B} = 1$, значения на входах А<В и А>В могут быть произвольными. Если на вход А=В подан «0», то $F_{A=B} = 0$, а $F_{A<B}$ и $F_{A>B}$ будут определяться значением на входах А<В и А>В.

Входы					Выходы		
A_1A_0	B_1B_0	$F_{A=B}^{-1}$	$F_{A<B}^{-1}$	$F_{A>B}^{-1}$	$F_{A=B}$	$F_{A<B}$	$F_{A>B}$
00	01	x	x	x	0	1	0
00	10	x	x	x	0	1	0
00	11	x	x	x	0	1	0
01	00	x	x	x	0	0	1
01	10	x	x	x	0	1	0
01	11	x	x	x	0	1	0
10	00	x	x	x	0	0	1
10	01	x	x	x	0	0	1
10	11	x	x	x	0	1	0
11	00	x	x	x	0	0	1
11	01	x	x	x	0	0	1
11	10	x	x	x	0	0	1
00	00	1	x	x	1	0	0
01	01	0	1	0	0	1	0
10	10	0	0	1	0	0	1
11	11	0	1	1	0	0	0
		0	0	0	0	1	1

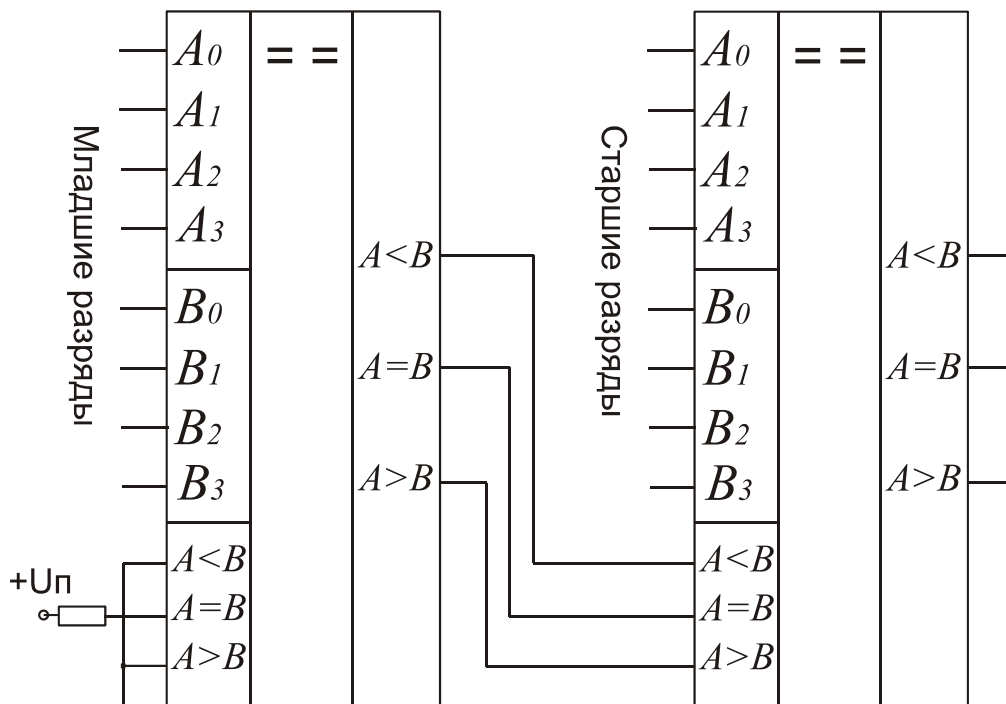
Домашнее задание:

Реализовать $F_{A>B}$, $F_{A<B}$, $F_{A=B}$ на мультиплексоре «16 на 1» (К155КП1).

Наращивание разрядности компаратора

При сравнении чисел разрядностью более четырех компараторы соединяются каскадом. Каскадирование может быть как последовательным, так и параллельным. В первом случае входы приема переноса каждого каскада соединяют с одноименными выходами предшествующего каскада. В такой схеме на вход А=В первого компаратора следует подать уровень логической «1», а состояние других входов А<В и А>В может быть произвольным.

Для повышения быстродействия компараторы могут соединяться по параллельной схеме. В этом случае входы приема переноса объединяют.



Последовательное соединение компараторов

Схема контроля четности

В системах передачи двоичной информации с целью повышения надежности передачи широко используется специфическая арифметическая операция - проверка паритета двоичных чисел. (*parity* – равенство, аналогия, соответствие).

Реализация этого метода осуществляется с помощью специальных *устройств сравнения (схем контроля четности)*.

Суть заключается в суммировании по модулю 2 всех разрядов с целью выяснения четности числа. При передаче информации по линии связи вследствие влияния помех могут возникать искажения, приводящие к искажениям информации. Если, например, передается код $1001 = 9$ и вследствие помех произойдет сбой во втором разряде слева, на приемный конец поступит $1101 = 13$.

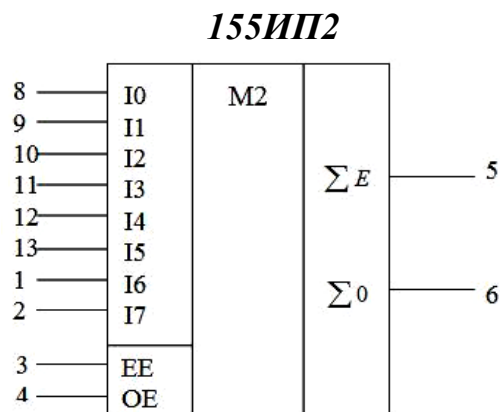
Простой и эффективный способ обнаружения ошибок основан на допущении, что в каждый момент времени ошибка может возникнуть только в одном разряде, и проявляется она в лишней единице или в потере единицы. В обоих случаях число единиц изменяется на одну. Таким образом, если передаваемое слово содержит четное число единиц по всем разрядам, а на конце линии передачи это число окажется нечетным, значит, появилась ошибка.

На передающем конце схема сравнения формирует дополнительный бит (1 или 0), так называемый паритетный или контрольный бит, который добавляется к выходной информации. Назначение паритетного бита - доводить число единиц в каждом передаваемом слове до четного или нечетного в зависимости от принятой системы кодирования. На приемном конце происходит проверка паритета поступивших сигналов. Если он правилен, разрешается прием.

Паритет может быть четным и нечетным. В случае нечетного паритета дополнительный бит формируется таким образом, чтобы сумма всех единиц в передаваемом слове, включая контрольный бит, была нечетной. Для четного наоборот. К примеру, в числе 0111 число единиц нечетно. Поэтому для нечетного паритета дополнительный бит должен быть нулем, а для четного - единицей. Передаваемое слово будет:

00111 - нечетный паритет; 10111 - четный паритет.

Здесь контрольный бит расположен слева, т. е. Занимает старший разряд передаваемого слова. На практике нечетный паритет используется чаще. Контроль нечетности позволяет фиксировать полное пропадание информации, поскольку слово из одних нулей (включая контрольный бит) противоречит нечетному паритету.



I_0 - I_7 – 8 информационных входов, два разрешающих входа для задания вида паритета: четный EE (even enable) и нечетный OE (odd enable), два выхода ΣE и $\Sigma 0$. Выходы взаимодополняющие.

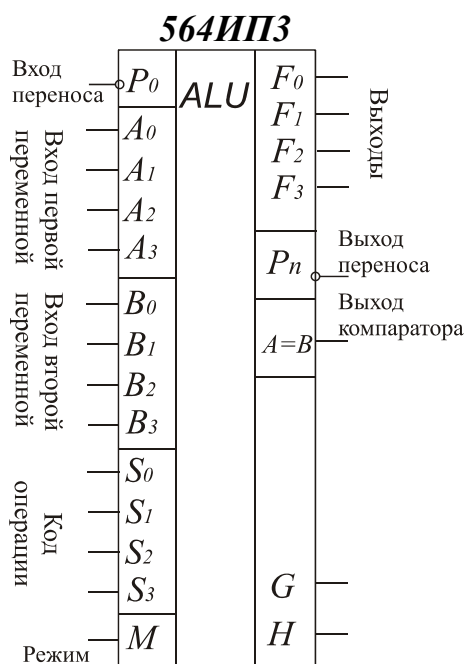
Микросхема может работать в режиме четного и нечетного паритета.

Входы				Выход	
A	B	C	D	при E = 0 нечет. парит.	при E = 1 четн. парит.
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

Домашнее задание:

Реализовать ΣE на мультиплексоре «16 на 1» (К155КП1).

Арифметико-логические устройства (АЛУ) – это многофункциональное устройство, которое выполняет над входными числами различные арифметические и логические операции.



Данная ИС оперирует с двумя 4-разрядными входными данными и способна выполнять 16 логических и 16 арифметических операций. Тип выполняемой операции (логическая или арифметическая) определяется уровнем управляющего сигнала на входе M (mode control). Если $M=1$, то в схеме блокируются все внутренние переносы и она поразрядно выполняет логические операции. При $M=0$ разблокируются внутренние переносы и ИС выполняет арифметические операции. Конкретный вид выполняемой операции определяется кодом операции - $S_3S_2S_1S_0$. Результаты выполненной операции снимаются с выводов F_3, F_2, F_1 и F_0 . Схема имеет вход P_0 и выход P_n переноса. Вывод $A=B$ является выходом встроенного компаратора K , формирующего “1” при $A=B$. Выходы G и H используются при построении многоразрядных АЛУ, с параллельным переносом.

Операции, выполняемые микросхемой 564ИПЗ

Код операции		Логические операции M=1	Арифметические операции M=0	
S ₁₆	S ₀ S ₁ S ₂ S ₃		$\bar{P}_0 = 1$	$\bar{P}_0 = 0$
0	0 0 0 0	\bar{A}	A	$A + 1$
1	0 0 0 1	$\overline{A \vee B}$	$A \vee B$	$(A \vee B) + 1$
2	0 0 1 0	$\overline{A \wedge B}$	$A \vee \bar{B}$	$(A \vee \bar{B}) + 1$
3	0 0 1 1	0	-1	0
4	0 1 0 0	$\overline{A \wedge \bar{B}}$	$A + (A \wedge \bar{B})$	$A + (A \wedge \bar{B}) + 1$
5	0 1 0 1	\bar{B}	$(A \vee B) + (A \wedge \bar{B})$	$(A \vee B) + (A \wedge \bar{B}) + 1$
6	0 1 1 0	$A \oplus B$	$A - B - 1$	$A - B$
7	0 1 1 1	$A \wedge \bar{B}$	$(A \wedge \bar{B}) - 1$	$A \wedge \bar{B}$
8	1 0 0 0	$\overline{A \vee B}$	$A + (A \wedge B)$	$A + (A \wedge B) + 1$
9	1 0 0 1	$\overline{A \oplus B}$	$A + B$	$A + B + 1$
A	1 0 1 0	B	$(A \vee \bar{B}) + (A \wedge B)$	$(A \vee \bar{B}) + (A \wedge B) + 1$
B	1 0 1 1	$A \wedge B$	$(A \wedge B) - 1$	$A \wedge B$
C	1 1 0 0	1	$A + A$	$A + A + 1$
D	1 1 0 1	$A \vee \bar{B}$	$(A \vee B) + A$	$(A \vee B) + A + 1$
E	1 1 1 0	$A \vee B$	$(A \vee \bar{B}) + A$	$(A \vee \bar{B}) + A + 1$
F	1 1 1 1	A	$A - 1$	A

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

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

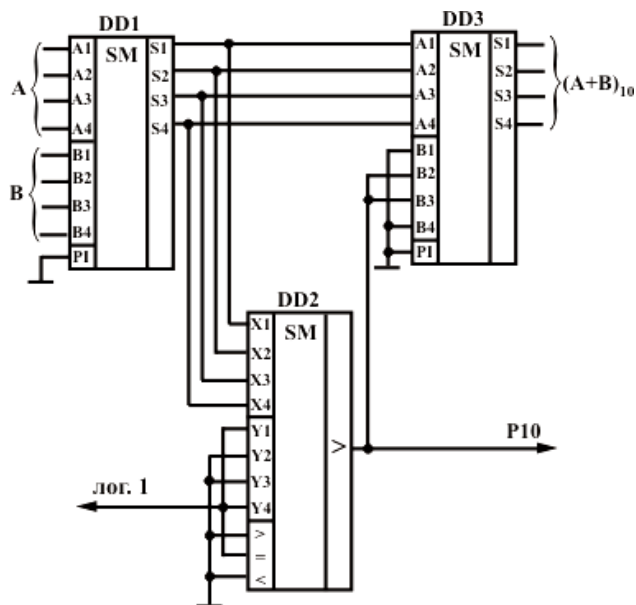
Наращивание разрядности АЛУ делается также, как и в сумматорах, т. е. выход переноса АЛУ младших разрядов подключается к входу переноса АЛУ старших разрядов. Но при большом числе разрядов время выполнения операций увеличивается из-за большого числа переносов из разряда в разряд. Для ускорения этого процесса используются специальные микросхемы ускоренного переноса (555ИП4, 564ИП4). Одна такая микросхема обслуживает до 4-х АЛУ. Получается 16-ти разрядное АЛУ.

Десятичный сумматор [дополнительно]

Кроме двоичных, в вычислительной технике часто используются так называемые двоично-десятичные коды. Они отображают выраженные в виде последовательности двоичных разрядов десятичные числа.

Двоично-десятичный код			
Двоичный код $x_3x_2x_1x_0$	Двоично-десятичный код		Десятичное число
1	2		3
0 0 0 0	0 0 0 0		0
0 0 0 1	0 0 0 1		1
0 0 1 0	0 0 1 0		2
0 0 1 1	0 0 1 1		3
0 1 0 0	0 1 0 0		4
0 1 0 1	0 1 0 1		5
0 1 1 0	0 1 1 0		6
0 1 1 1	0 1 1 1		7
1 0 0 0	1 0 0 0		8
1 0 0 1	1 0 0 1		9
1 0 1 0	0 0 0 1	0 0 0 0	10
1 0 1 1	0 0 0 1	0 0 0 1	11
1 1 0 0	0 0 0 1	0 0 1 0	12
1 1 0 1	0 0 0 1	0 0 1 1	13
1 1 1 0	0 0 0 1	0 1 0 0	14
1 1 1 1	0 0 0 1	0 1 0 1	15

Особенность двоично-десятичного кода предполагает использование для суммирования специальных логических схем. Смысл их построения состоит в том, что сначала двоично-десятичные коды суммируются как двоичные. Если результатом суммирования является несуществующий двоично-десятичный код, его необходимо уменьшить на 10, и дополнительно сформировать сигнал переноса. Уменьшение кода на 10 может выполняться его суммированием с дополнительным кодом числа 10 (0110). На рисунке приведена схема сумматора двоично-десятичных чисел на основе двоичных сумматоров.



Операцию сложения выполняет сумматор DD1. При сумме большей или равной десяти на выходе микросхемы DD2, которая является схемой сравнения входов, формируется сигнал переноса P10.

На второй вход (Y1-Y4) микросхемы DD2 подается двоичный эквивалент числа 9 (1001).

Сумматор DD3 осуществляет десятичную коррекцию результата суммирования. При отсутствии сигнала переноса на выходе микросхемы DD3 повторяется код числа, который был на выходе DD1, поскольку на входы B поданы лог. 0. При наличии сигнала переноса $P_{10}=1$ на входах B2-B3 устанавливаются лог. 1, что соответствует коду числа 6. Допустим есть числа $A=8$ и $B=4$. На выходе сумматора DD1 появляется код числа 12 ($8+4=12$). На выходе DD2 формируется сигнал переноса, сумматор DD3 выполняет операцию $12+6=18$. Числу 18 соответствует код 10010. На выходах S сумматора DD3 устанавливается код 0010 (собственный перенос микросхемы DD3 не учитывается). Поскольку на выходе P10 число 10, на выходах сумматора число 2 (0010), то в результате получается число 12.