Типовые комбинационные схемы

Перязева Юлия Валерьевна

Доцент кафедры ВТ

Теория автоматов
Дискретные преобразователи без памяти

Дешифратор. Шифратор

Содержание

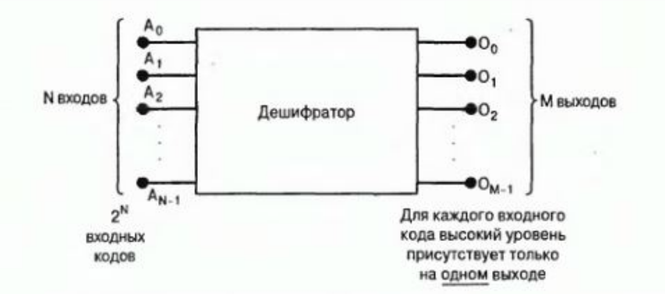
1. Дешифратор. Шифратор

2. Мультиплексор

3. Арифметические схемы

Дешифратор

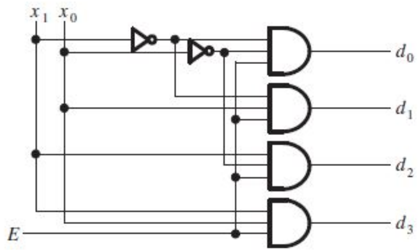
Дешифратор преобразует входной двоичный n-разрядный код, в двоичное число, содержащее не более одной единицы, в номер выходного сигнала (дешифрирует код). При этом входное n-разрядное двоичное число обычно совпадает с номером выхода.



Дешифратор

Некоторые дешифраторы имеют один или несколько разрешающих входов, которые используются для управления работой дешифратора.

Inputs			Outputs			
E	x_1	x_0	d_0	d_1	d_2	d_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



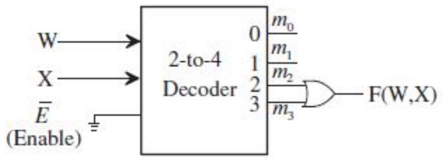
Дешифратор. Пример

Спроектируем схему функции $f(w, x) = w$ с использование 2-4 дешифратора и минимально возможного набора логических элементов.

Дешифратор. Пример

Спроектируем схему функции $f(w,x) = w$ с использование 2-4 дешифратора и минимально возможного набора логических элементов.

$f(w,x) = w = xw \vee \overline{x}w$



Дешифратор. Пример

Спроектируем схему с использование 3-8 дешифратора и минимально возможного набора логических элементов для следующей системы булевых функций:

$f_0(x, y, z) = \overline{x}yz \vee xz$

$f_1(x, y, z) = y(x \vee \overline{z})$

$f_2(x, y, z) = xy \vee \overline{x}yz$

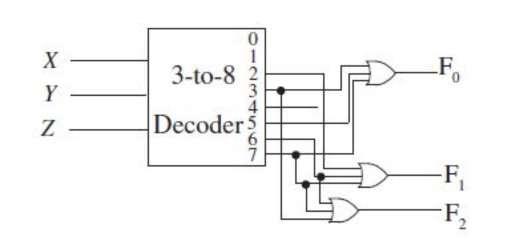
Дешифратор. Пример

Спроектируем схему с использование 3-8 дешифратора и минимально возможного набора логических элементов для следующей системы булевых функций:

$f_0(x, y, z) = \overline{x}yz \vee xz$

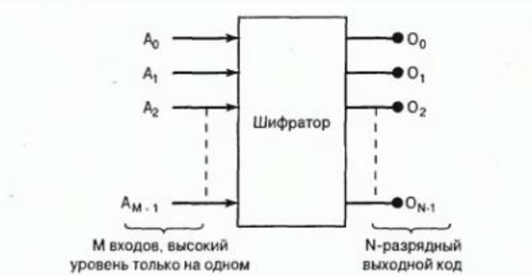
$f_1(x, y, z) = y(x \vee \overline{z})$

$f_2(x, y, z) = xy \vee \overline{x}yz$



Шифратор

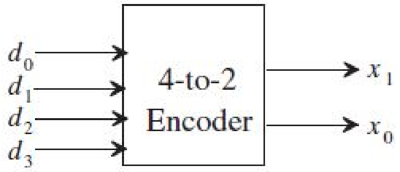
Шифратор преобразует номер входного сигнала в выходной двоичный код (шифрует номер входного сигнала).



Различают приоритетные и неприоритетные шифраторы. Неприоритетный шифратор – такой, у которого при появлении активного сигнала более чем на одном входе выходная информация становится недостоверной. У приоритетного шифратора каждому входу присвоен свой уровень приоритета. Если активный сигнал появится на нескольких входах одновременно, то на выходах появится номер того входа, приоритет которого выше всех остальных.

Шифратор

Inputs				Outputs	
d_0	d_1	d_2	d_3	x_1	x_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



Шифратор

10 десятичных цифр от 0 до 9 могут быть представлены их соответствующими 4-битными двоичными числами. Такой код называют двоично-десятичными, BCD или 8421.

n	Код 8421
	$x_8 x_4 x_2 x_1$
0	0 0 0 0
1	1 0 0 0
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

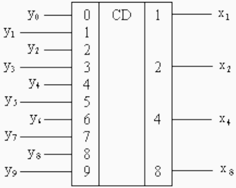


рис. 17

Преобразователи кодов

Преобразователем кодов называется цифровое устройство, осуществляющее преобразование слов входного алфавита $(x_1, x_2, ..., x_n)$ в слова выходного алфавита $(y_1, y_2, ..., y_k)$, при этом соотношения между n и k могут быть любыми: $n = k, n > k, n < k$.
Преобразователи кодов можно разделить на два типа:

- с весовым преобразователем кодов (десятичные коды в двоичные, двоично-десятичные в двоичные);
- с невесовым преобразователем кодов (двоично-десятичного кода в код семисегментного индикатора десятичных цифр, двоичного весового кода в код Грея).

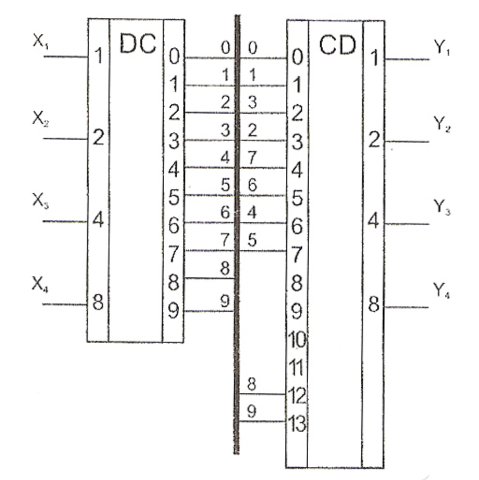
Преобразователи кодов

Преобразование кода 8421 в код Грея.

Код 8421	Код Грея
$x_4x_3x_2x_1$	$y_4y_3y_2y_1$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 0
0 1 0 0	0 1 1 0
0 1 0 1	0 1 1 1
0 1 1 0	0 1 0 1
0 1 1 1	0 1 0 0
1 0 0 0	1 1 0 0
1 0 0 1	1 1 0 1

Одним из весьма распространенных путей реализации преобразователей кодов является метод последовательного соединения дешифратора и шифратора, например, преобразователь 8421 – код Грея:

n	Код 8421		Код Грея
	$x_4x_3x_2x_1$		$y_4y_3y_2y_1$
0	0 0 0 0	0	0 0 0 0
1	0 0 0 1	1	0 0 0 1
2	0 0 1 0	3	0 0 1 1
3	0 0 1 1	2	0 0 1 0
4	0 1 0 0	6	0 1 1 0
5	0 1 0 1	7	0 1 1 1
6	0 1 1 0	5	0 1 0 1
7	0 1 1 1	4	0 1 0 0
8	1 0 0 0	12	1 1 0 0
9	1 0 0 1	13	1 1 0 1



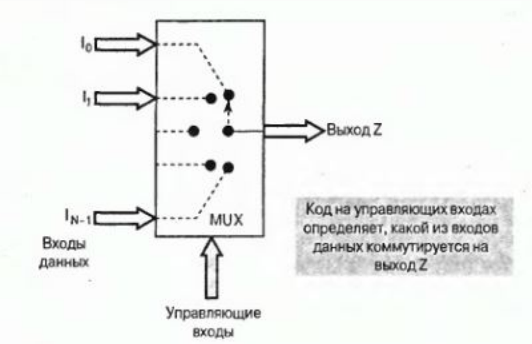
Мультиплексор

Содержание

1. Дешифратор. Шифратор
2. Мультиплексор
3. Арифметические схемы

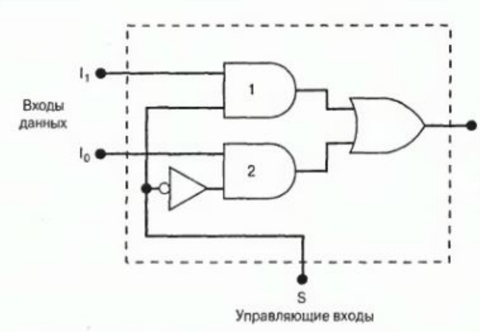
Мультиплексор (multiplexer, MUX) – это комбинационная схема, имеющая n адресных входов S, 2^n информационных входов $I_0 \dots I_{2^n-1}$ и один выход f_n. Если на адресные входы подать n-значное двоичное число i, то выход f_n подключиться к i-му информационному входу. Функция описывающая полный мультиплексор:

$$f_n = I_0 \overline{S_{n-1}}\,\overline{S_{n-2}} \dots \overline{S_1}\,\overline{S_0} \vee I_1 \overline{S_{n-1}}\,\overline{S_{n-2}} \dots \overline{S_1}\, S_0 \vee \dots \vee I_{2^n-1} S_{n-1}\, S_{n-2} \dots S_1\, S_0$$



Мультиплексор с двумя входами данных

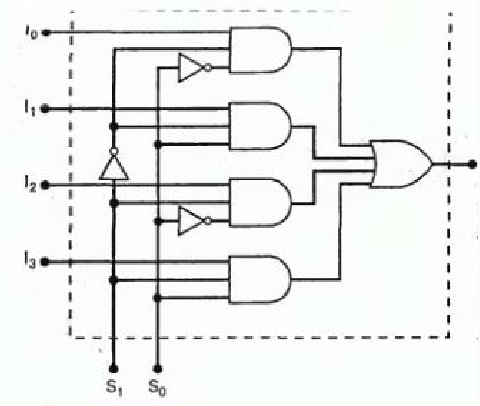
S – адресный, управляющий вход, при $S = 0$ на выходе I_0, при $S = 1$ на выходе I_1. При $n = 2$: $f_2 = I_0\overline{S} \vee I_1 S$



Мультиплексор с четырьмя входами данных

При $n = 2$ $f_2 = I_0\overline{S_1}\,\overline{S_0} \vee I_1\overline{S_1}\,S_0 \vee I_2 S_1\,\overline{S_0} \vee I_3 S_1\,S_0$

S_1	S_0	Выход
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$



Мультиплексор

С математической точки зрения, мультиплексор реализует операцию нахождения производной от булевой функции, описывающей структуру этого мультиплексора, если производную брать по I_k. Например, для

$$f_2(S_0, S_1, I_0, I_1, I_2, I_3) = I_0\overline{S_1}\,\overline{S_0} \vee I_1\overline{S_1}\,S_0 \vee I_2 S_1\,\overline{S_0} \vee I_3 S_1\,S_0,$$

найдем производную по I_2

$$f_2(S_0, S_1, I_0, I_1, I_2, I_3)'_{I_2} = f_2(S_0, S_1, I_0, I_1, I_2, I_3)^0_{I_2} \oplus f_2(S_0, S_1, I_0, I_1, I_2, I_3)^1_{I_2} =$$

$$(I_0\overline{S_1}\,\overline{S_0} \vee I_1\overline{S_1}\,S_0 \vee I_3 S_1\,S_0) \oplus (I_0\overline{S_1}\,\overline{S_0} \vee I_1\overline{S_1}\,S_0 \vee S_1\,\overline{S_0} \vee I_3 S_1\,S_0) = S_1\,\overline{S_0}$$

Если $S_1\,\overline{S_0} = 1$, то $f_2 = I_2$, т.е. функция меняет свое значение одновременно с изменением значения аргумента.

Демультиплексор

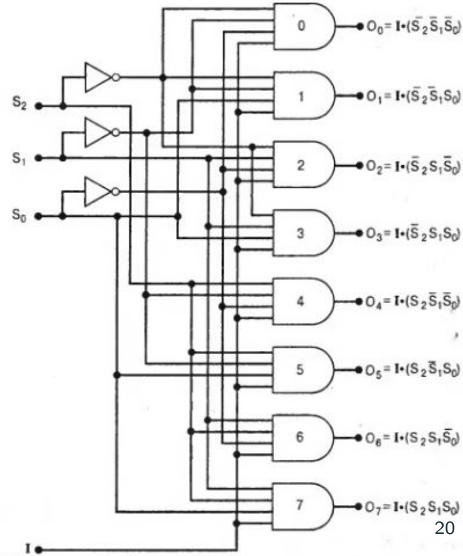
Демультиплексор (DEMUX) берет один входной сигнал и распределяет на несколько выходов, поступает один сигнал от источника данных, который проходит на один из выходов:

Демультиплексор

Демультиплексор с одним входом и 8 выходами:

Управляющий код			Выходы							
S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

Примечание. I — вход данных



Арифметические схемы

Содержание

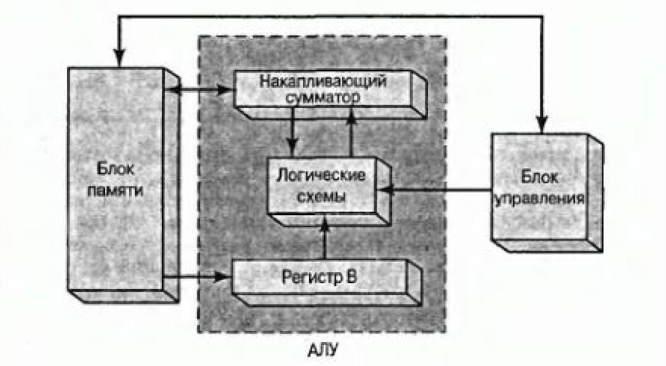
1. Дешифратор. Шифратор

2. Мультиплексор

3. Арифметические схемы

Арифметические схемы

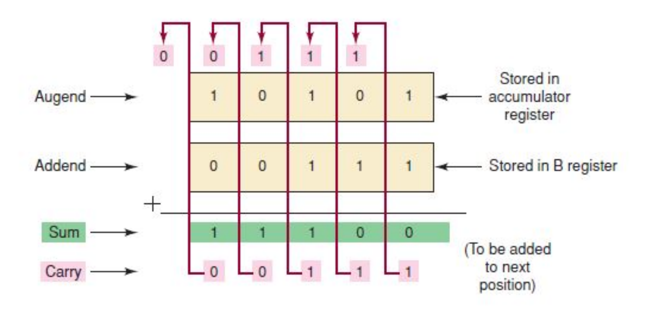
Одной из наиболее важных функций большинства компьютеров является выполнение арифметических операций. В компьютерах эти операции осуществляются с помощью арифметико-логических устройств (АЛУ). Схема типичного АЛУ:



Арифметические схемы

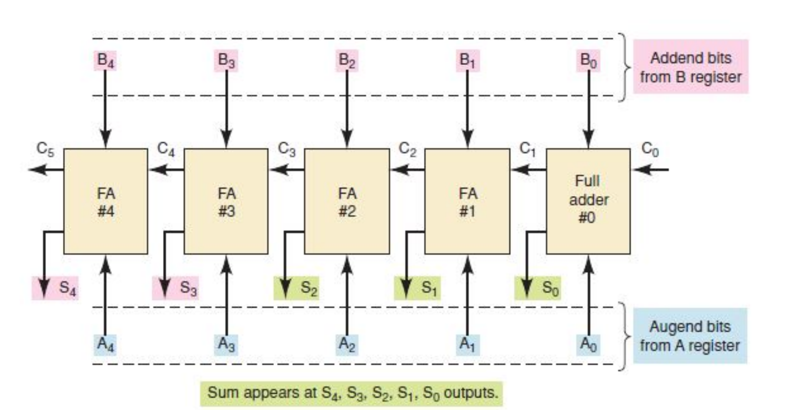
Пусть есть два n-разрядных числа, записанные в двоичной системе счисления. Тогда их сумму можно вычислять «в столбик» (x_i, y_i – разряды чисел, а c_i – единицы переноса):

$$\begin{array}{cccccc} C_{n+1} & C_n & C_{n-1} & \dots & C_1 & C_0 \\ & X_n & X_{n-1} & \dots & X_1 & X_0 \\ & Y_n & Y_{n-1} & \dots & Y_1 & Y_0 \\ \hline & S_n & S_{n-1} & \dots & S_1 & S_0 \end{array}$$



Арифметические схемы

Параллельный пятибитовый сумматор, в каждой позиции данного устройства используется схема полного сумматора:



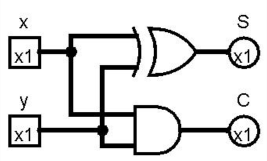
Полусумматор

Когда складывают два бита x и y, генерируется бит суммы и бит переноса, такая комбинационная схема называется полусумматор.

1. Таблица истинности:

x y	S	C
0 0	0	0
0 1	1	0
1 0	1	0
1 1	0	1

2. Получение формулы и минимазация:
   $S(x, y) = x \oplus y = (x \vee y)\overline{xy}$
   $C(x, y) = x \& y$
3. Построение схемы в базисе $\{\&, \oplus, -\}$:

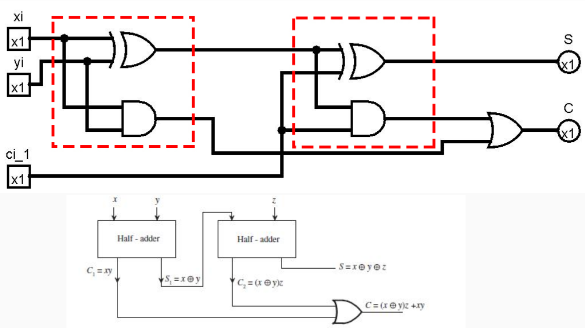


Полный сумматор

Таблица истинности:

$x_i y_i c_i$	S	C
0 0 0	0	0
0 0 1	1	0
0 1 0	1	0
0 1 1	0	1
1 0 0	1	0
1 0 1	0	1
1 1 0	0	1
1 1 1	1	1

Схема:



Формула:

$$S = x_i \oplus y_i \oplus c_i$$

$$C = \overline{x_i} y_i c_i \vee x_i \overline{y_i} c_i \vee x_i y_i \overline{c_i} \vee x_i y_i c_i = x_i y_i(\overline{c_i} \vee c_i) \vee c_{i-1}(\overline{x_i} y_i \vee x_i \overline{y_i}) = x_i y_i \vee c_i(x_i \oplus y_i)$$

Вычитание

Большинство современных компьютеров используют дополнительный код для представления отрицательных чисел и осуществления операции вычитания. Положительные и отрицательные величины, включая знаковые биты, могут быть сложены вместе с помощью простейшего параллельного сумматора, если предварительно перевести отрицательные числа в дополнительный код.

Вычитание

Можно построить схему параллельного сумматора в которой вычитаемое будет заменяться дополнительным кодом: