

# Тема 2. Комбинационные логические устройства

Это такие устройства, значение на выходе которых в данный момент времени зависит только от значений на входе в тот же момент времени. Такие устройства не имеют памяти и представляют просто отображение текущих значений на входе значениям на выходе по некоторой функции. Рассмотрим их подробнее. Материал этой лекции основан на курсе - Цифровые и микропроцессорные устройства. Китаев Ю.В. (<http://de.ifmo.ru/--books/electron/>)

## Алгебра логики

Теоретической основой проектирования цифровых устройств (ЦУ) является **алгебра логики** или булева алгебра, оперирующая логическими переменными. Для логических переменных, принимающих только два значения, существуют 4 основных операции. Операция логическое **"И"** (**AND**) **конъюнкция** или **логическое умножение**, обозначается \* или  $\wedge$ . Операция логическое **"ИЛИ"** (**OR**), **дизъюнкция** или **логическое сложение**, обозначается + или  $\vee$ . Операция логическое **"НЕ"** (**NOT**), изменение значения, **инверсия** или отрицание, обозначается чертой над логическим выражением. Инверсия иногда будет в тексте обозначаться знаком " ~ ". Операция эквивалентности обозначается "=" . Следующие соотношения являются **аксиомами**.

(1)	$0 + 0 = 0$	$1 * 1 = 1$	(1')
(2)	$1 + 1 = 1$	$0 * 0 = 0$	(2')
(3)	$1 + 0 = 0 + 1 = 1$	$0 * 1 = 1 * 0 = 0$	(3')
(4)	$\sim 1 = 0$	$\sim 0 = 1$	(4')

Из (1, 2) и (1',2') следует:  $x + x = x$  и  $x * x = x$ . (5)

Из (1, 3) и (2',3') следует:  $x + 0 = x$  и  $0 * x = 0$ . (6)

Из (2, 3) и (1',3') следует:  $1 + x = 1$  и  $x * 1 = x$ . (7)

Из (3) и (3') следует:  $x + \sim x = 1$  и  $\sim x * x = 0$ . (8)

Из (4) и (4') следует:  $\sim(\sim x) = x$ . (9)

И, наконец, из (1,1'), (2,2'), (3,3') и (4,4') следует:

$\sim(x_0 + x_1) = \sim x_0 * \sim x_1$  и  $\sim(x_0 * x_1) = \sim x_0 + \sim x_1$ . (10)

Последние выражения (10) называют **принципом двойственности** или **теоремой Де Моргана** (инверсия логической суммы равна логическому произведению инверсий и наоборот). Соотношения двойственности для n переменных, часто записывают в виде:

$\sim(x_1 + \dots + x_n) = \sim x_1 * \dots * \sim x_n$  и

$$\sim(x_1 * \dots * x_n) = \sim x_1 + \dots + \sim x_n \quad (11)$$

На функции И и ИЛИ распространяются обычные алгебраические законы - **переместительный, сочетательный и распределительный**, которые легко доказываются методом перебора:  $x_1 \text{ ор } x_0 = x_0 \text{ ор } x_1$  - переместительный,  $x_2 \text{ ор } x_1 \text{ ор } x_0 = (x_2 \text{ ор } x_1) \text{ ор } x_0$  - сочетательный и  $x_2 * (x_1 + x_0) = (x_2 * x_1) + (x_2 * x_0)$  и  $x_2 + (x_1 * x_0) = (x_2 + x_1) * (x_2 + x_0)$  - распределительный, где операция **ор** может быть, либо И, либо ИЛИ. Наряду с тремя основными логическими функциями, называемыми также переключательными, существуют и другие.

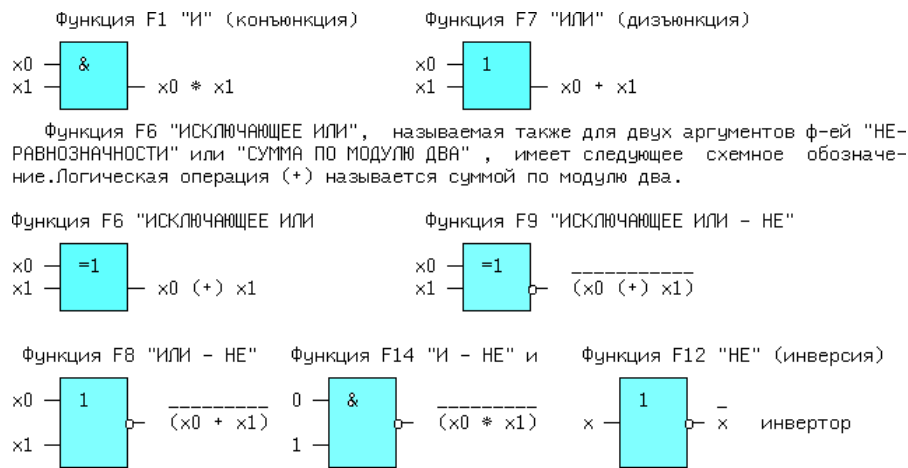
## 1.2 ПЕРЕКЛЮЧАТЕЛЬНЫЕ ФУНКЦИИ

Для  $n$  - логических переменных (аргументов) существует  $2^n$  их комбинаций или двоичных наборов. На каждом таком наборе может быть определено значение функции 0 или 1. Если значения функции отличаются хотя бы на одном наборе, функции - разные. Общее число переключательных функций (ПФ) от  $n$  аргументов равно  $N=2^{2^n}$ . Для  $n=2$ ,  $N=16$ . При  $n=3$ ,  $N=256$  и далее очень быстро растет. Практическое значение имеют 16 функций от 2-х переменных, т.к. любое сложное выражение можно рассматривать как композицию из простейших. В таблице 1 приведены некоторые из ПФ для  $n=2$ .  $i$ -номер набора входных переменных  $x_1$  и  $x_0$ .

$i$	0	1	2	3	ОБОЗНАЧЕНИЕ ФУНКЦИИ
$x_0$ $x_1$	0 0	1 0	0 1	1 1	
F0	0	0	0	0	F0 = 0, константа "0"
F1	0	0	0	1	F1 = $x_1 * x_0$ , "И"
F6	0	1	1	0	F6 = $x_1 (+) x_0$ , "ИСКЛЮЧАЮЩЕЕ ИЛИ"
F7	0	1	1	1	F7 = $x_1 + x_0$ , "ИЛИ"
F8	1	0	0	0	F8 = $\sim(x_1 + x_0)$ , "ИЛИ - НЕ"
F9	1	0	0	1	F9 = $\sim(x_1 (+) x_0)$ , "ИСКЛ. ИЛИ - НЕ"
F12	1	1	0	0	F12 = $\sim x_1$ , "НЕ"
F14	1	1	1	0	F14 = $\sim(x_1 * x_0)$ , "И - НЕ"
F15	1	1	1	1	F15 = 1, константа "1"

**ЗАПОМНИТЕ СЛЕДУЮЩИЕ ОПРЕДЕЛЕНИЯ.** Функция "И" равна единице, если равны единице ВСЕ ее аргументы. Функция "ИЛИ" равна единице, если равен единице ХОТЯ БЫ один аргумент. Функция "ИСКЛЮЧАЮЩЕЕ ИЛИ" (XOR) равна единице, если равен единице ТОЛЬКО один ее аргумент.

## 1.3 УСЛОВНЫЕ ОБОЗНАЧЕНИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ НА СХЕМАХ



### 1.4 СПОСОБЫ ПРЕДСТАВЛЕНИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ

Целью проектирования цифрового устройства является получение его логической функции (ЛФ) и соответствующей ей схемной реализации. ЛФ могут иметь различные формы представления: 1) **словесное**, 2) **графическое**, 3) **табличное**, 4) **алгебраическое**, 5) **на алгоритмическом языке (например VHDL)** и 6) **схемное**. В качестве примера, рассмотрим функцию Y от двух переменных x1 и x0, заданную словесным описанием: Y=1, если переменные НЕ РАВНЫ и Y=0, если x1=x0. Такую ЛФ удобно назвать функцией НЕРАВНОЗНАЧНОСТИ. Переходим к табличному представлению Y (таблица 2).

x0	x1	y
0	0	f0 = 0
0	1	f1 = 1
1	0	f2 = 1
1	1	f3 = 0

**Табличное представление значений ЛФ для всех наборов входных переменных называется таблицей истинности.** В общем виде переход от табличного представления к алгебраическому может осуществляться по формуле (12), одной из основных в алгебре логики.

Выражение (12) называется совершенной дизъюнктивной нормальной формой ЛФ (СДНФ).  $m_i$  - минтерм или логическое произведение всех переменных i-го двоичного набора, входящих в прямом виде, если значение этой переменной в наборе равно 1, и в инверсном виде, если ее значение равно 0.  $f_i$  - значение ЛФ на i - ом наборе. Доказательство (12) базируется на теореме разложения, в соответствии с которой любую ЛФ  $f(..)$  от n-переменных можно разложить по переменной  $x_i$  в виде:  $f(x(n-1),...,x_i, ...,x_0) = \sim x_i * f(x(n-1),...,0,...,x_0) + x_i * f(x(n-1),...,1,...,x_0)$ . Это выражение для  $x_i=0$  равно  $\sim 0 * f(x(n-$

$1), \dots, 0, \dots, x_0) + 0 * f(x_{(n-1)}, \dots, 1, \dots, x_0) = f(x_{(n-1)}, \dots, 0, \dots, x_0)$ . При  $x_i=1$  оно будет равно  $\sim 1 * f(x_{(n-1)}, \dots, 1, \dots, x_0) + 1 * f(x_{(n-1)}, \dots, 1, \dots, x_0) = f(x_{(n-1)}, \dots, 1, \dots, x_0)$ , т.е. при любых значениях  $x_i$  теорема разложения справедлива. Теореме разложения можно применить  $n$  раз и тогда ЛФ будет разложена по всем своим переменным.

В виде примера рассмотрим функцию  $F=f(x_1, x_0)$  от двух переменных. Разложение этой функции по переменной  $x_1$  даст:  $F = \sim x_1 * f(0, x_0) + x_1 * f(1, x_0)$ . Продолжая эту операцию для переменной  $x_0$ , получим:

$$F = \sim x_1 * (\sim x_0 * f(0, 0) + x_0 * f(0, 1)) + x_1 * (\sim x_0 * f(1, 0) + x_0 * f(1, 1)) = \sim x_1 * \sim x_0 * f(0, 0) + \sim x_1 * x_0 * f(0, 1) + x_1 * \sim x_0 * f(1, 0) + x_1 * x_0 * f(1, 1). \quad (12.1)$$

Выражение (12.1) позволяет записать все переключательные функции от двух переменных, используя только три основных логических операции.

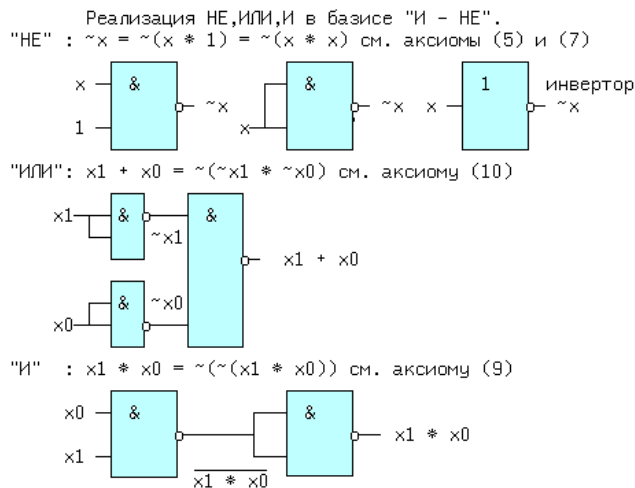
Рассмотрим разложение функций  $F_7$ -"ИЛИ" и  $F_1$ -"И", для чего необходимо обратиться к соответствующим строчкам таблицы 1. Функция И на двоичных наборах входных переменных  $x_1$  и  $x_0$  (00, 01, 10, 11) принимает значения 0, 0, 0, 1. Записывая выражение (12.1) для этих значений получим:  $F_1(x_1, x_0) = \sim x_1 * \sim x_0 * 0 + \sim x_1 * x_0 * 0 + x_1 * \sim x_0 * 0 + x_1 * x_0 * 1 = x_1 * x_0$ , что соотгласуется с ее определением. Таким же образом, находим алгебраическое выражение функции  $F_7$ -"ИЛИ", которая, соответственно, на тех же входных наборах принимает значения: 0, 1, 1, 1. Тогда, в соответствии с (12.1),  $F_7(x_1, x_0) = \sim x_1 * \sim x_0 * 0 + \sim x_1 * x_0 * 1 + x_1 * \sim x_0 * 1 + x_1 * x_0 * 1$ . Вынося за скобки в двух последних слагаемых  $x_1$ , получим  $F_7 = \sim x_1 * x_0 * 1 + x_1 * (\sim x_0 * 1 + x_0 * 1)$ . На основании аксиомы (8), выражение в скобке равно "1" и  $F_7 = \sim x_1 * x_0 * 1 + x_1$ . Применяя распределительный закон, найдем  $(\sim x_1 + x_1) * (x_0 + x_1) = x_0 + x_1$ .

Возвращаясь к таблице 2, получим  $Y = 0 * \sim x_1 * \sim x_0 + 1 * \sim x_1 * x_0 + 1 * x_1 * \sim x_0 + 0 * x_1 * x_0 = \sim x_1 * x_0 + x_1 * \sim x_0 = x_1 (+) x_0 = F_6$  (функция неравнозначности).

С помощью формулы (12) любую, сколь угодно сложную, логическую функцию можно представить в виде трех основных ЛФ: "И", "ИЛИ", "НЕ", представляющих собой логический базис.

### 1.5 ЛОГИЧЕСКИЙ БАЗИС

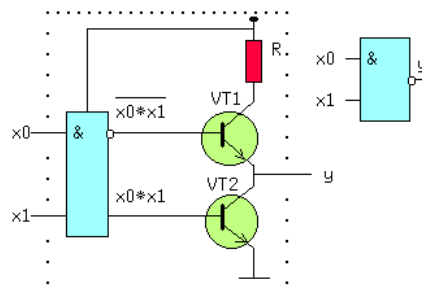
Набор простейших ЛФ, позволяющих реализовать любую другую функцию называется логическим базисом (ЛБ). Функции И, ИЛИ, НЕ не являются минимальным ЛБ, т.к. сами могут быть представлены через другие функции, например через  $F_8$ (ИЛИ -НЕ) или  $F_{14}$ (И - НЕ).



Следовательно базис "И - НЕ" является минимальным. Реализацию НЕ,И,ИЛИ в базисе "ИЛИ - НЕ" произвести самостоятельно, используя [перечисленные аксиомы](#).

### 1.6.1 БАЗОВЫЙ ЛОГИЧЕСКИЙ ЭЛЕМЕНТ

На рисунке приведена упрощенная схема И-НЕ и его условное обозначение.

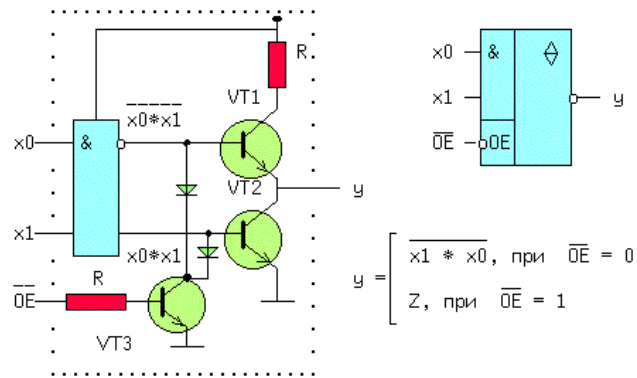


Напряжения на базах транзисторов VT1 и VT2 находятся в противофазе и, если  $x0*x1=1$ , то нижний транзистор открыт, а верхний закрыт, так как  $\sim(x0*x1)=0$ . Потенциал коллектора VT2 в этом случае примерно равен нулю и следовательно  $y=0$ . При других значениях  $x0$  и  $x1$  нижний транзистор закрыт, а верхний открыт и на выходе схемы - высокий уровень, т.е. схема работает как элемент И-НЕ. Выходы нескольких БЛЭ категорически нельзя соединять вместе, потому что, если  $n-1$  элементов находятся в состоянии "1", а  $n$ -ый в состоянии "0", то  $n-1$  транзисторов VT1 будут "сливать" (sink) токи в единственный транзистор VT2  $n$ -го элемента. Суммарный ток может превысить допустимое значение и VT2 выйдет из строя.

### 1.9 ТРИСТАБИЛЬНЫЕ ЭЛЕМЕНТЫ

Наряду с двумя логическими состояниями существует третье технологическое состояние, когда выход элемента отключается от внутренней схемы. При этом сопротивление между выходом и "землей" становится очень большим и выход микросхемы не оказывает никакого влияния на подключенные к нему выходы других микросхем. Выходы

нескольких таких элементов также могут соединяться вместе. Такое включение, разновидность "монтажного И", применяется там, где несколько источников сигналов по очереди подключаются к входам одного или нескольких приемников, не мешая друг другу. Третье состояние называют также высокоимпедансным или Z - состоянием. Схема И-НЕ с Z-состоянием выхода приведена на рис. слева, а ее условное обозначение - справа.



Если сигнал  $\sim OE=0$ , транзистор VT3 закрыт и включенные встречно диоды не оказывают влияния на логические выходы элемента И. Напряжения на базах транзисторов VT1 и VT2 находятся в противофазе и, если  $x0*x1=1$ , то верхний транзистор закрыт, а нижний открыт. Потенциал коллектора VT2 примерно равен нулю и следовательно  $y=0$ . При других значениях  $x0$  и  $x1$  нижний транзистор заперт, а верхний открыт и на выходе схемы - высокий уровень, т.е. при  $\sim OE=0$  схема работает как обычный элемент И-НЕ. Картина существенно изменится при  $\sim OE=1$ . Транзистор VT3 откроется до насыщения и на базах транзисторов VT1 и VT2 потенциал опустится примерно до нуля, запирая их. Выход "y" окажется отключенным от внутренней логической схемы. На схемах тристабильные элементы обозначаются ромбом с поперечной чертой или буквой Z.

Такие элементы используются там, где необходима передача информации по одной линии от нескольких источников к одному или нескольким приемникам. Причем, так как линия одна, то чтобы выходы пассивных источников не искажали информацию на выходе активного источника, они должны переводиться в третье состояние. Z - состояние используется по этой причине в микросхемах памяти, шинных формирователях.

Дополнительный инверсный вход относится к категории управляющих или функциональных. Функция входа зашифрована в его обозначении (Output Enable - разрешение выхода ( $\sim OE$ )), а значение активного уровня на этом входе, при котором функция выполняется, равно 1, если вход прямой, и равно 0, если вход инверсный, как на схеме.

## ДЕШИФРАТОР

Дешифратор (ДШ) преобразует двоичный код на входах в активный сигнал на том выходе, номер которого равен десятичному эквиваленту двоичного кода на входах. В полном дешифраторе количество выходов  $m = 2^n$ , где  $n$  - число входов. В неполном ДШ  $m < 2^n$ . По определению полный ДШ должен генерировать  $2^n$  выходных ЛФ, определенных на всех наборах из  $n$  - входных переменных, т.е. минтермов. Рассмотрим ДШ с  $n=2$  и  $m=4$ , называемый также дешифратором "2 в 4" и дополним его входом разрешения выходов ОЕ. Активным уровнем сигнала на прямых входах/выходах будет 1, а на инверсных - 0. По этому определению заполним таблицу истинности, где величина  $x$  может принимать любые значения.

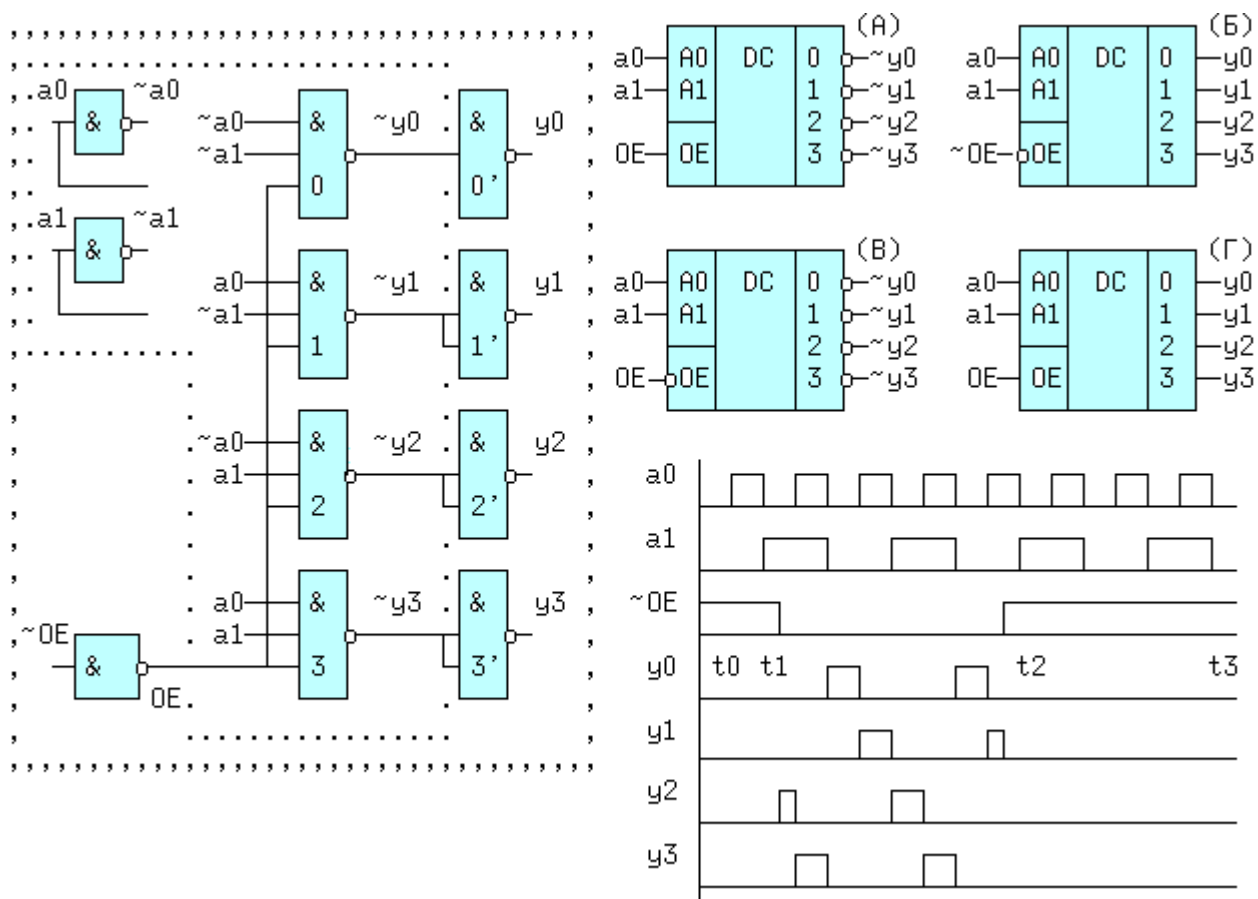
DEC число	Входы				Выходы								ЛФ
	a1	a0	OE	$\overline{OE}$	y0	y1	y2	y3	$\overline{y0}$	$\overline{y1}$	$\overline{y2}$	$\overline{y3}$	
0	0	0			1	0	0	0	0	1	1	1	$y0=OE*\sim a1*\sim a0$
1	0	1	1	0	0	1	0	0	1	0	1	1	$y1=OE*\sim a1*a0$
2	1	0			0	0	1	0	1	1	0	1	$y2=OE*a1*\sim a0$
3	1	1			0	0	0	1	1	1	1	0	$y3=OE*a1*a0$
x	x	x	0	1	0	0	0	0	1	1	1	1	$y_i=0 (\sim y_i=1)$

Таблица Карно для выхода  $y_0$  и 3-х входных переменных будет иметь вид:

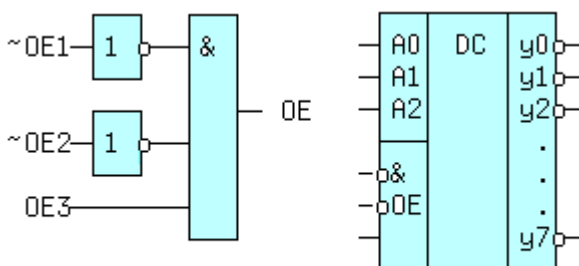
	a1a0	"y0"		
OE	00	01	11	10
0	0	0	0	0
1	1	0	0	0

Прямоугольник, составленный из 1-ных клеток содержит только одну такую клетку, поэтому логическая функция выхода  $y_0$  будет иметь вид:  $y_0 = OE*\sim a1*\sim a0$ . Аналогично получены остальные три уравнения. Преобразуем полученные для  $y_i$  уравнения с помощью аксиомы двойного отрицания к базису И-НЕ:  $y_0 = \sim(\sim(OE*\sim a1*\sim a0))$ . Решению соответствует схема на рис.13.

Схема обведенная "..." имеет условное обозначение (А), а схема в запятых - обозначение (Б). Возможны также еще 2 комбинации прямых и инверсных входов и выходов В и Г.



Пояснить работу ДШ можно с помощью временных диаграмм для схемы (Б). Во время действия сигнала  $\sim OE=1$  на нижних входах элементов И-НЕ(0..3) присутствует  $OE=0$ , и независимо от значений  $a_0, a_1$ , выходные значения  $\sim y_i=1$ , а  $y_i=0$ , что и видно из рис.13. В эти отрезки времени  $t_0..t_1$  и  $t_2..t_3$  выходы "запрещены", т.е. на прямых выходах  $y_i$  устанавливается пассивный уровень "0", а на инверсных выходах пассивная "1". В интервале  $t_1..t_2$  сигнал  $\sim OE=0$  ( $OE=1$ ) и значения  $y_i$  зависят только от переменных  $a_1, a_0$ . Если код на входах  $A1A0=10$ , что соответствует десятичной двойке, на входах второго элемента И-НЕ соберутся 3 логических "1". Сигнал  $\sim y_2=0$ , а  $y_2=1$ , что видно на диаграмме  $y_2$ . Вместо инвертора  $OE$ , может применяться более сложная схема, показанная на рис.14. Здесь  $OE=1$  в случае, когда  $\sim OE_1 = \sim OE_2 = 0$  и  $OE_3 = 1$ .

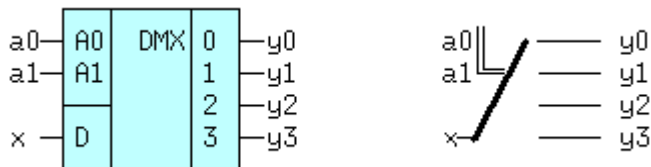


Такая схема применяется в дешифраторе "3 в 8" типа 153ЗИД7(555ИД7), условное обозначение которого приведено на рис.15. Дешифраторы широко применяются в вычислительной технике, как часть больших интегральных схем, для выбора одного из нескольких внешних устройств (ВУ) при обмене данными между ним и микропроцессором. В этом случае на входы  $a_i$  подаются сигналы, называемые адресом ВУ, а входы называются адресными.



## ДЕМУЛЬТИПЛЕКСОР

Устройство передающее сигнал с информационного входа на один из выходов, причем номер этого выхода равен десятичному эквиваленту двоичного кода на адресных входах, называется **демультиплексором (ДМ)**. В качестве ДМ может использоваться [дешифратор](#), у которого вместо сигнала ОЕ подается информационный сигнал  $x$ . Например, если на входы подать код  $a_1a_0=10(\text{BIN})=2(\text{DEC})$ , то сигнал  $x$  появится на выходе  $y_2$ , а на остальных выходах  $y_i=0$ . На рис.16. даны условное обозначение ДМ "1 в 4" и его механический аналог.



## МУЛЬТИПЛЕКСОР

Мультиплексор передает сигнал с одного из информационных входов  $x_i$  на единственный выход  $y$ , причем номер этого входа равен десятичному эквиваленту двоичного кода на адресных входах  $a_i$ . Если имеется вход разрешения выхода ОЕ, то "0" на этом входе должен перевести выход в пассивное состояние (последняя строчка таблицы). Рассмотрим мультиплексор "4 в 1", имеющий 4 информационных входа и  $\log_4 = 2$  адресных входов.

Десятич. число	Входы				Вых				ЛФ	
	$a_1$	$a_0$	ОЕ	ОЕ	$x_0$	$x_1$	$x_2$	$x_3$		$y$
0	0	0			$x_0$	$x$	$x$	$x$	$x_0$	$y=x_0 \cdot \text{ОЕ}^{\sim a_1} \cdot \sim a_0$
1	0	1	1	0	$x$	$x_1$	$x$	$x$	$x_1$	$y=x_1 \cdot \text{ОЕ}^{\sim a_1} \cdot a_0$
2	1	0			$x$	$x$	$x_2$	$x$	$x_2$	$y=x_2 \cdot \text{ОЕ} \cdot a_1 \cdot \sim a_0$
3	1	1			$x$	$x$	$x$	$x_3$	$x_3$	$y=x_3 \cdot \text{ОЕ} \cdot a_1 \cdot a_0$
x	x	x	0	1	x	x	x	x	0	$y=0$

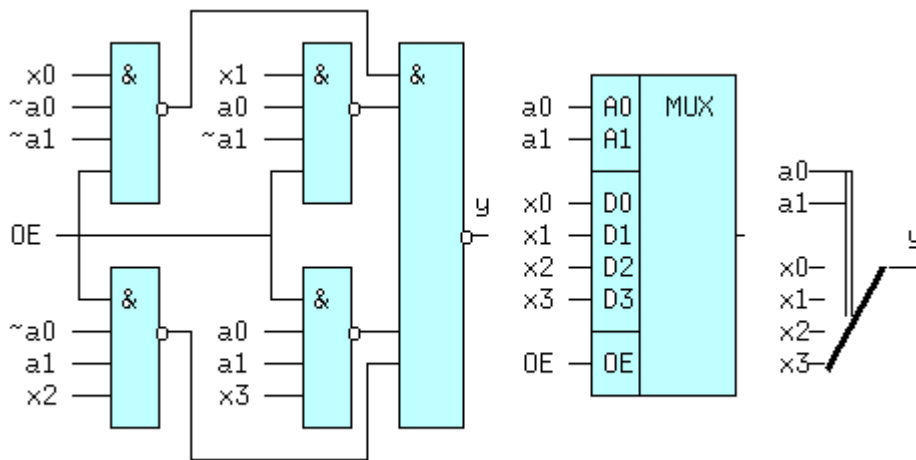
Величина  $x$  может принимать любые значения. Количество входных переменных равно 7, и таблица истинности должна иметь 128 строк. В табл в 4-х основных строках упаковано 64 исходных (с учетом значений  $x_0 \dots x_3$ ) и в последней строке, остальные 64 строчки. Анализ 0 строки, приводит к выводу, что  $y=x_0$ , если  $a_1=0$  И  $a_0=0$  И  $\text{ОЕ}=1$ , независимо от переменных  $x_1 \dots x_3$ . Поэтому для этого входного набора можно записать:  $y=x_0 \cdot \text{ОЕ}^{\sim a_1} \cdot \sim a_0$ . Аналогично записывается  $y$  для остальных трех наборов переменных. Общее решение тогда будет иметь вид:

$$y = \text{ОЕ}(x_0 \cdot \sim a_1 \cdot \sim a_0 + x_1 \cdot \sim a_1 \cdot a_0 + x_2 \cdot a_1 \cdot \sim a_0 + x_3 \cdot a_1 \cdot a_0).$$

Применяя аксиомы двойного отрицания и двойственности к правой части уравнения получим:

$$y = \sim(OE * x_0 * \sim a_1 * \sim a_0) + \dots + \sim(OE * x_0 * a_1 * a_0).$$

Предыдущему выражению соответствует схема, приведенная на рис.



Если на адресные входы подать комбинацию  $a_1a_0 = 11(\text{BIN}) = 3(\text{DEC})$ , то к выходу  $y$  будет подключен вход  $D_3$ , при условии, если  $OE=1$ .

Мультиплексоры находят широкое применение в вычислительной технике, например многие выходы  $y$  микропроцессоров "мультиплексированы", т.е. к одному выходу подключается несколько внутренних источников различных сигналов. Это могут быть сигналы линий шины данных и шины адреса, передаваемые последовательно во времени, что позволяет сократить общее число выводов микропроцессора.

## Контрольные вопросы

### Теория

1. Аксиомы и теоремы булевой алгебры. Способы представления логических функций
2. Базисные логические элементы. Цифровой выход логического элемента. Третье состояние логического сигнала.
3. Дешифратор и мультиплексор. Реализация на логических элементах.

### Задания

1. Нарисовать схему
  - Компаратора 1 битных и 2х битных чисел
  - Однобитного полного сумматора