

**ПРИМЕРЫ ПРОГРАММИРОВАНИЯ ЭЛЕМЕНТОВ ЦИФРОВОЙ ЭЛЕКТРОНИКИ НА ЯЗЫКЕ VHDL В СРЕДЕ XILINX**

**EXAMPLES OF PROGRAMMING OF ELEMENTS OF DIGITAL ELECTRONICS IN LANGUAGE OF VHDL IN ENVIRONMENT OF XILINX**

**XILINX ОРТАСЫНДА САНДЫҚ ЭЛЕКТРОНИКА ЭЛЕМЕНТТЕРІНЕ VHDL ТІЛІНДЕ БАҒДАРЛАМА ҚҰРУ МЫСАЛДАРЫ**

Жармакин Болатхан Кайкенович., м.т.н., преподаватель  
Zharmakin Bolatkhan, Master of technical science, teacher

Евразийский национальный университет им. Л.Н. Гумилева, г. Астана, [zbk\\_60@mail.ru](mailto:zbk_60@mail.ru).  
L.N. Gumilyov Eurasian National University, Astana city

**АНДАТПА**

Бұл мақалада VHDL тілін қолданып әртүрлі сандық логикалық элементтер үшін бағдарламаларға мысалдар келтірілген. Логикалық элементтердің шартты графикалық сұлбалары, ақиқат кестелері, бағдарламаларға мысалдар мен түсініктемелер келтірілген. Бағдарламалық кодтардың XILINX ортасында алынған шешімдері бар. XILINX ортасында элементтердің шартты графикалық сұлбалары келтірілген. Осы мақалада қарастырылған логикалық элементтердің уақыттық диаграммалары көрсетілген.

**АННОТАЦИЯ**

В статье приводятся примеры программирования на языке VHDL различных логических элементов цифровой электроники. Приведены условные графические обозначения, таблицы истинности, примеры программирования и комментарии к ним. Показаны результаты программного кода в среде XILINX. Отдельно выведены графические обозначения элементов в среде XILINX. Показаны временные диаграммы рассматриваемых в данной статье логических элементов.

**ARTICLE**

In this article programming examples are given for various logical elements of digital electronics in VHDL language. There are also listed conditional graphic symbols, tables of the validity examples of programming and comments to them. The program code results are shown out as temporal diagrams in XILINX medium. Graphic symbols of elements have been separately displayed in the XILINX medium. Timing charts of logical elements have been presented as well.

**Ключевые слова:** Цифровая интегральная схема (ИС), логический элемент (ЛЭ), логическая переменная, логическая функция, дизъюнкция, временная диаграмма, мультиплексор, шифратор.

Целью данной статьи является объединение двух составляющих современной цифровой электроники. Зачастую, при изучении цифровых элементов программное обеспечение данных устройств обычно не рассматривается. Также при изучении основ программирования упор делается на языки высокого уровня. Языки низкого уровня в виде Ассемблера или VHDL изучаются слабо. Требования же к современному инженеру со стороны работодателей за последнее время изменились. Сейчас на производстве нуждаются в инженерах хорошо знающих электронику и умеющие неплохо программировать. Целью данной статьи является пробудить у студентов интерес к данной проблематике.

Итак, элементной базой современных цифровых устройств и систем являются цифровые интегральные схемы.

Цифровая интегральная схема (ИС) – это микросэлектронное изделие, выполняющее определенную функцию преобразования дискретных (цифровых) сигналов. Простейшие преобразования над цифровыми сигналами осуществляют цифровые ИС, получившие названия логических элементов (ЛЭ).

Для описания работы цифровых ИС, а следовательно и устройств, построенных на их основе, используется математический аппарат алгебры логики или булевой алгебры.

Основными понятиями булевой алгебры являются понятия логической переменной и логической функции.

Логической переменной называется величина, которая может принимать одно из двух возможных состояний, одно из которых обозначается символом “0”, другое – “1”. Сами двоичные переменные чаще обозначают символами  $x_1, x_2, \dots$

Логической (булевой) функцией (обычное обозначение –  $y$ ) называется функция двоичных переменных (аргументов), которая также может принимать одно из двух возможных состояний: “0” или “1”.

Здесь для примера рассмотрим одну из логических функций.

Функция “дизъюнкция” – это функция двух или большего числа аргументов (другие названия функции: логическое сложение, логическая связь ИЛИ). Функция равна 1, если хотя бы один из ее аргументов равен 1 (рис. 1, б). Обозначение функции “Дизъюнкция”:

$$y = x_1 + x_2 \text{ или } y = x_1 \vee x_2.$$

ЛЭ, реализующий функцию “дизъюнкция”, называют дизъюнктором или ЛЭ “ИЛИ”. Условное изображение и временные диаграммы ЛЭ “ИЛИ” приведены на рис. 1 а, г, в.

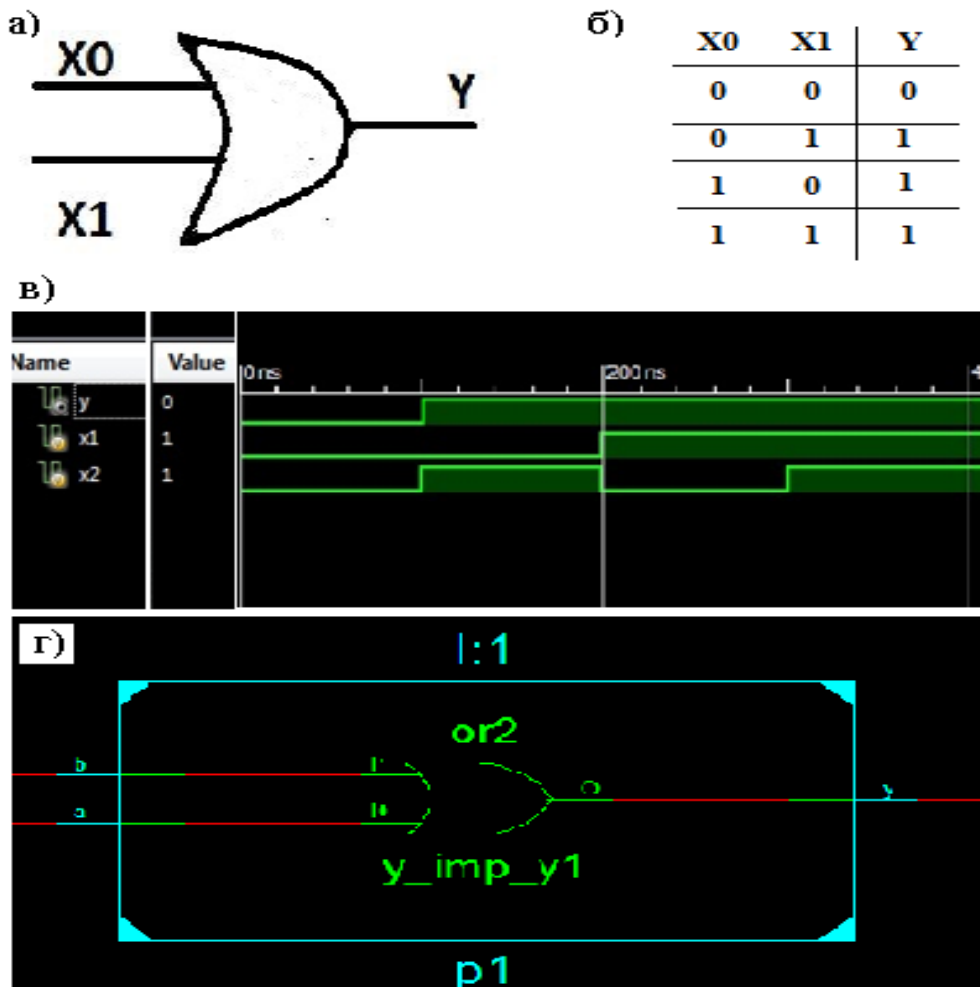


Рисунок 1 - Условное графическое обозначение дизъюнктора по ANSI (а), таблица истинности инвертора (б), временная диаграмма полученная в среде XILINX (в), обозначение в среде XILINX (г)

```

entity An is
port (x1,x2:in bit;
y:out bit);
end An;
architecture Behavioral of An is
component I
port (a,b:in bit;
y:out bit);
end component;
signal w1:bit;
begin
p1:I port map (a=>x1,b=>x2,y=>w1);
y<=w1;
end Behavioral;
entity I is
port (a,b:in bit;
y:out bit);
end I;
architecture Behavioral of I is
begin
y<=a or b after 1 ns;
end Behavioral;

```



Описание портов ввода/вывода: **in bit** - два (x1, x2) входных порта, **out bit** - выходной (y) порт дизъюнктора.

Присвоение входным портам x1 и x2 значений переменных a и b, выходному порту y значение w1 в карте порта (port map). p1 - ЛЭ (дизъюнктор).

Присвоение функции y значение операции сложения (a or b) через 1(одну) наносекунду (ns)

Рисунок 2 - Описание дизъюнктора на языке программирования VHDL

Следующий пример посвящен более сложному элементу - мультиплексору.

Мультиплексором (Multiplexer) называют комбинационное устройство с M информационными (X<sub>0</sub>, X<sub>1</sub>, ..., X<sub>M-1</sub>), и K адресными (A<sub>0</sub>, A<sub>1</sub>, ..., A<sub>K-1</sub>) входами и одним выходом (Y), которое осуществляет передачу сигнала с заданного адресным кодом информационного входа на его выход.

Кроме информационных и адресных входов, мультиплексор содержит вход разрешения (E - enable), при подаче на который активного уровня (в нашем случае уровень логического «0») мультиплексор переходит в активное состояние.

Мы рассмотрим мультиплексор 4×1 с инверсным входом разрешения E и прямым выходом Y, представляющий собой половину микросхемы мультиплексора КР555КП2.

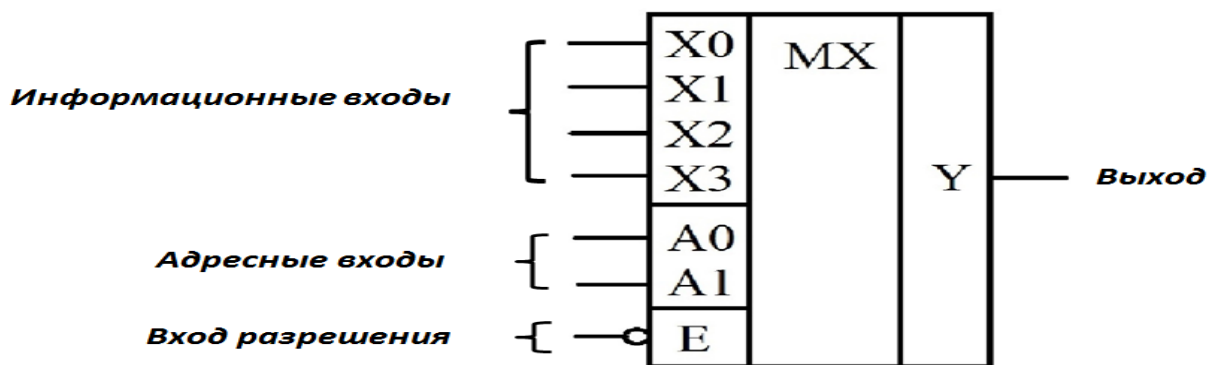


Рисунок 3- Условное графическое обозначение мультиплексора 4×1

Выражение для выходной функции такого мультиплексора можно записать в виде:

$$Y = \bar{E} \wedge (X_0 \wedge \bar{A}_0 \wedge \bar{A}_1 \vee X_1 \wedge A_0 \wedge \bar{A}_1 \vee X_2 \wedge \bar{A}_0 \wedge A_1 \vee X_3 \wedge A_0 \wedge A_1), \quad (1)$$

Нарисуем принципиальную схему мультиплексора, подчиняющегося закону, показанному в формуле (1) с помощью программы PAINT.

Сперва определим, какие логические элементы понадобятся, чтобы собрать схему мультиплексора. Для этого разложим на логические элементы формулу (1):

$$Y = \bar{E} \wedge (X_0 \wedge \bar{A}_0 \wedge \bar{A}_1 \vee X_1 \wedge A_0 \wedge \bar{A}_1 \vee X_2 \wedge \bar{A}_0 \wedge A_1 \vee X_3 \wedge A_0 \wedge A_1),$$

Рисунок 4 – Необходимые логические элементы

Итого, нам для построения принципиальной схемы мультиплексора необходимы 4 элемента И для информационных входов – X0, X1, X2, X3 и один элемент ИЛИ.

Также, для сигналов разрешения E и адресных входов A0 и A1 необходимы элемент НЕ (инвертор), т. к. данные элементы используют в формуле (1) как прямые, так и инверсные значения.

На рисунке 5 таким образом мы размещаем 4 элемента И, 3 элемента НЕ и 1 элемент ИЛИ.

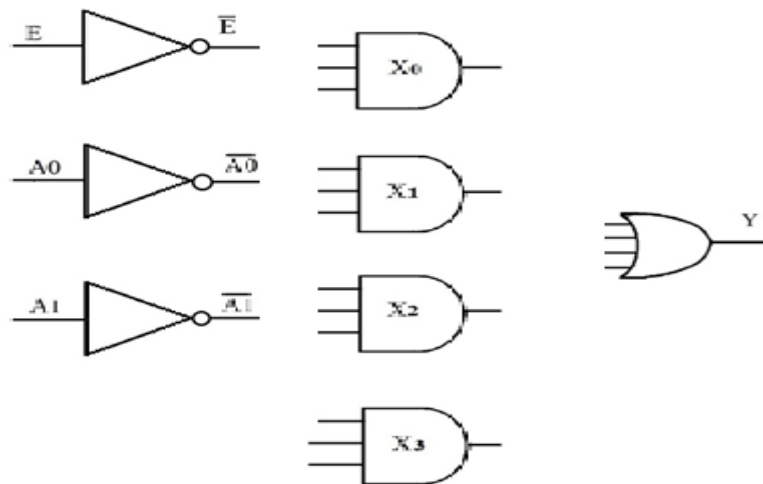


Рисунок 5 - Логические элементы необходимые для рисования схемы

Начнем с сигнала разрешения E. На рисунке 3 данный сигнал изображен со входной инверсией, который указывает что данный сигнал имеет активный уровень, равный логическому нулю (лог. «0»). Поэтому сигнал разрешения берется с инверсного выхода и соединяется со всеми информационными входами мультиплексора.

Далее, вычерчиваем соединения для информационного входа мультиплексора X0, учитывая что на его вход должны быть поданы 3 сигнала согласно формуле (1). Сигнал разрешения E уже у нас имеется. Сигналы с адресных входов A0 и A1 инверсные. Поэтому входы X0 соединяются линиями с инверсных входов инверторов.

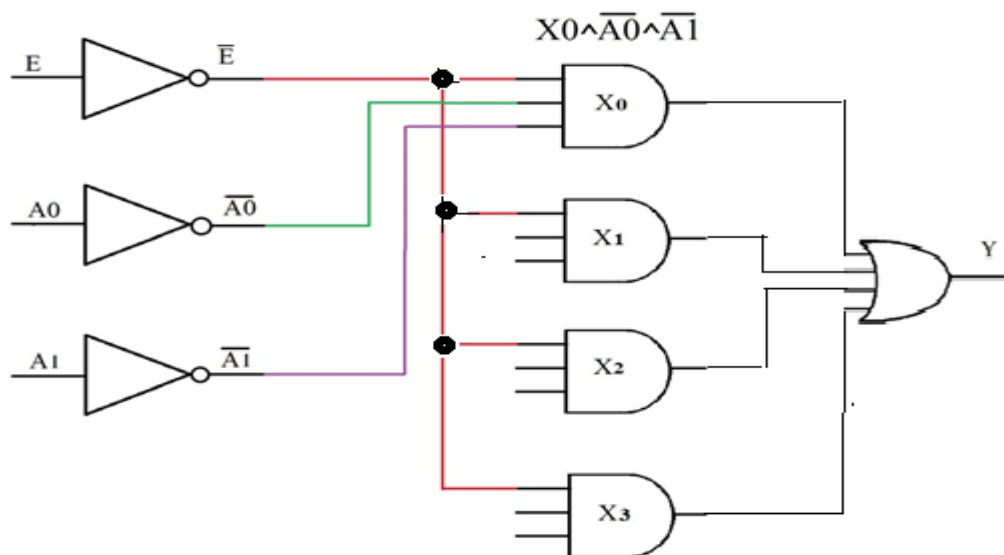


Рисунок 6 - Линии соединения для информационного входа X0

Аналогично вычерчиваем соединения для информационных входов мультиплексора X1, X2, X3. Далее, соединив выходы информационных входов мультиплексора X0, X1, X2, X3 в дизъюнктор, получаем полную схему мультиплексора с выходом Y (рис. 7). Таким образом, мы собрали принципиальную схему мультиплексора.

Теперь перейдем к описанию модели мультиплексора на языке VHDL в среде XILINX. На рисунке 8 показан фрагмент программы на языке VHDL и элементы описанные на рисунке 7.

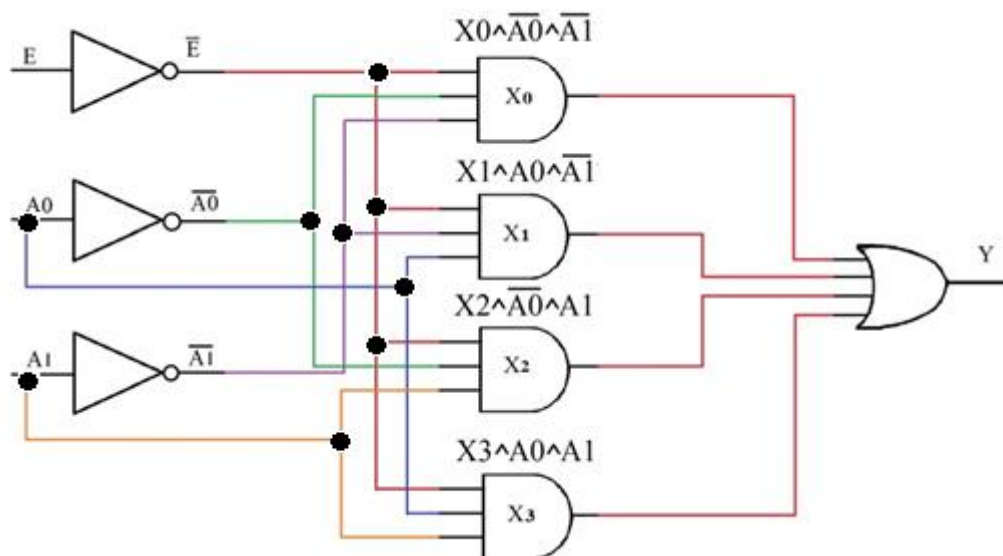


Рисунок 7 - Линии соединения для информационных входов мультиплексора X0, X1, X2, X3

```

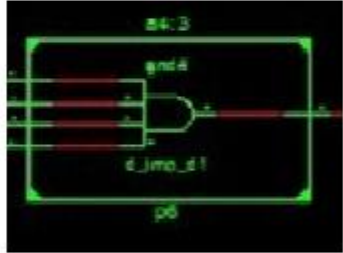
p8: o4 port map (c=>z4,f=>z5,j=>z6,k=>z7,d=>y);
end Behavioral;
entity n is
port( c: in bit;
d: out bit);
end n;
architecture str1 of n is
begin
d<= not c after 1 ns;
end str1;
entity a4 is
port( c,f,j,k:in bit;
d: out bit);
end a4;
architecture str1 of a4 is
begin
d<= c and f and j and k after 3 ns;
end str1;
entity o4 is
port( c,f,j,k:in bit;
d: out bit);
end o4;
architecture str1 of o4 is
begin
d<= c or f or j or k after 3 ns;
end str1;

```

Элемент NOT



Элемент 4 - AND



Элемент 4 - OR

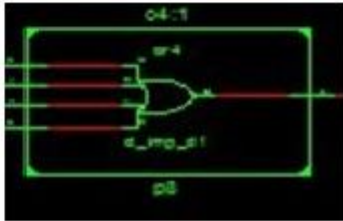
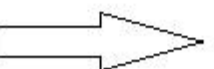


Рисунок 8 - Фрагмент программы на языке VHDL для мультиплексора

Работа мультиплексора описана в таблице 1. Запустив программу мы получаем временную диаграмму показанную на рисунке 9.

Пояснения к рисунку 9. Работа мультиплексора начинается с 100 – й наносекунды, когда сигналу разрешения E присваивается значение логического нуля. Сигнал A0 задан буквой a , сигнал A1- буквой b. Для наглядности, в соответствии значениям на адресных входах a и b (A0 и A1) на входах X0 – X3 имеем сигнал логической единицы, которые отражаются на выходе Y.

Сравнивая данные с таблицы 1 и данные временной диаграммы можно увидеть их полное соответствие, что хотели и показать в данной статье.

Таблица 1 – Таблица истинности для мультиплексора 4×1

E	A1	A0	X3	X2	X1	X0	Y
1	×	×	×	×	×	×	0
0	0	0	×	×	×	0	0
0	0	0	×	×	×	1	1
0	0	1	×	×	0	×	0
0	0	1	×	×	1	×	1
0	1	0	×	0	×	×	0
0	1	0	×	1	×	×	1
0	1	1	0	×	×	×	0
0	1	1	1	×	×	×	1

Здесь символ × указывает на то, что состояние соответствующего сигнала не имеет значение, т.е. не влияет на состояние выхода.



Рисунок 9- Временная диаграмма, полученная в среде XILINX

На рисунке 10 показана принципиальная схема мультиплексора, полученная в среде XILINX. Сравнивая данный рисунок с рисунком 7 мы видим их полное соответствие.

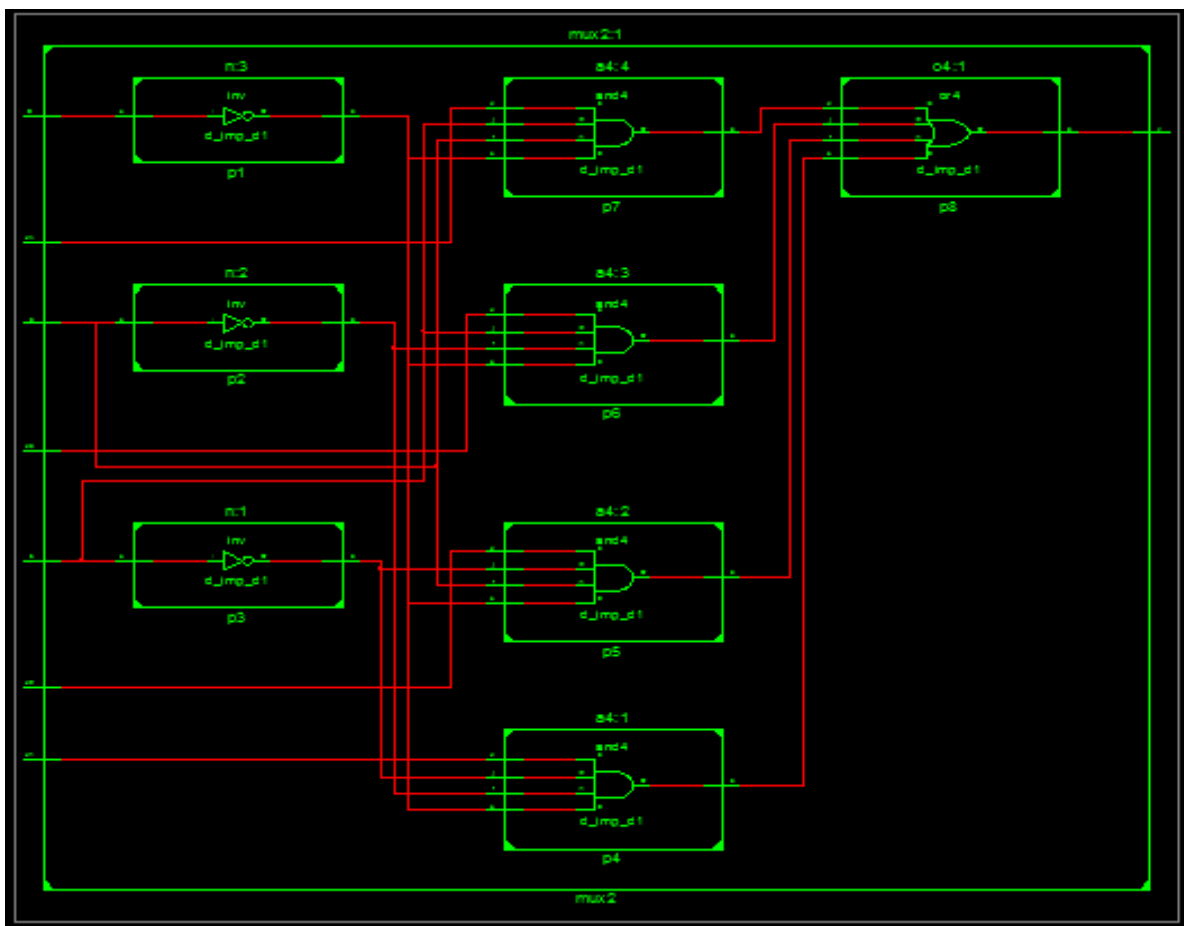


Рисунок 10 - Принципиальная схема мультиплексора, полученная в среде XILINX

Теперь рассмотрим работу шифратора.

Шифратором (Coder – CD)  $M \times N$  называют комбинационное устройство с  $M$  входами и  $N$  выходами, преобразующее  $M$  - разрядный унитарный код в  $N$  – разрядный двоичный код.

Рассмотрим работу шифратора на примере преобразователя десятичных чисел от 0 до 9 в двоично-десятичный код.

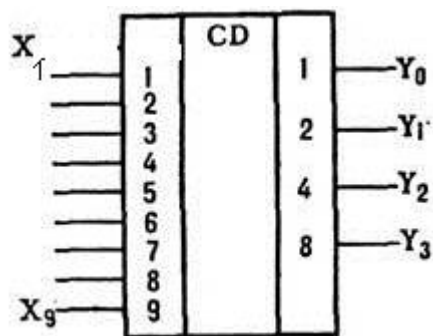


Рисунок 11 - Условное обозначение шифратора

Данная интегральная микросхема имеет следующий набор входных и выходных сигналов:

- девять информационных входов  $X_1, \dots, X_9$ ;
- четыре информационных выхода  $Y_0, Y_1, Y_2, Y_3$ .

Работа устройства иллюстрируется таблицей состояний на таб. 2.

Таблица 2 – Таблица истинности для шифратора

Входы									Выходы			
$X_9$	$X_8$	$X_7$	$X_6$	$X_5$	$X_4$	$X_3$	$X_2$	$X_1$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0	1	0	0	0

Выражения для выходных функций такого можно записать в виде:

$$Y_0 = X_1 \wedge X_3 \wedge X_5 \wedge X_7 \vee X_9$$

$$Y_1 = X_2 \wedge X_3 \wedge X_6 \wedge X_7 \quad (2)$$

$$Y_2 = X_4 \wedge X_5 \wedge X_6 \wedge X_7$$

$$Y_3 = X_8 \wedge X_9$$

Для начала определим, какие логические элементы понадобятся, чтобы собрать схему шифратора. Для этого разложим на логические элементы формулу (2).



$$Y_0 = X_1 \wedge X_3 \wedge X_5 \wedge X_7 \vee X_9$$

.Логическое ИЛИ

$$Y_1 = X_2 \wedge X_3 \wedge X_6 \wedge X_7$$

.Логическое ИЛИ

$$Y_2 = X_4 \wedge X_5 \wedge X_6 \wedge X_7$$

.Логическое ИЛИ

$$Y_3 = X_8 \wedge X_9$$

.Логическое ИЛИ

Рисунок 12 – Необходимые логические элементы

Итого, нам для построения принципиальной схемы шифратора необходимы 4 элемента ИЛИ для информационных входов –  $X_1, X_2, \dots, X_9$ .

В результате получаем принципиальную схему шифратора:

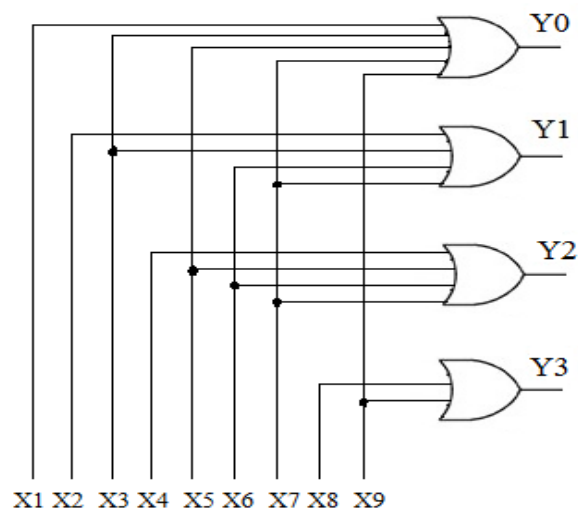


Рисунок 13 -Принципиальная схема шифратора

Согласно полученной принципиальной схеме шифратора, приступим к работе в программной среде XILINX (Рис. 14). По окончании программы получаем временную диаграмму (Рис. 15).

Вывод: Для улучшения освоения студентами дисциплин «Цифровая электроника» и «Программирование на языке VHDL» желателен симбиоз данных предметов. В данной статье мной была сделана попытка объединение вышеназванных дисциплин. Буду очень благодарен за критические замечания и приглашения по сотрудничеству в данной сфере.

```

entity shifrator is
port (x1,x2,x3,x4,x5,x6,x7,x8,x9:in bit;
y0,y1,y2,y3:out bit);
end shifrator;
architecture str1 of shifrator is
component o5
port (a,b,c,d,e:in bit;
y:out bit);
end component;
component o4
port (a,b,c,d:in bit;
y:out bit);
end component;
component o
port (a,b:in bit;
y:out bit);
end component;
signal w1,w2,w3,w4:bit;
begin
p1:o5 port map(a=>x1, b=>x3, c=>x5, d=>x7, e=>x9, y=>w1);
p2:o4 port map(a=>x2, b=>x3,c=>x6,d=>x7,y=>w2);
p3:o4 port map (a=>x4, b=>x5,c=>x6, d=>x7,y=>w3);
p4:o port map (a=>x8, b=>x9,y=>w4);

```

Имеются 9 входных сигналов (X1 – X9) и 4 выходных сигнала (Y0 – Y3)

Здесь описываются компоненты: O5 означает схему OR (ИЛИ) с 5 входами и переменными (a, b, c, d, e) и т.д.

Здесь в карте порта происходит присвоение значений переменных (a, b, c, d, e) входным сигналам X1, X3, X5, X7 и X9 соответственно

Рисунок 14 - Фрагмент программы на языке VHDL для шифратора



Рисунок 15 - Временная диаграмма шифратора в программной среде XILINX

#### References:

1. Bibilo PN "Basics of VHDL». Ed. Third, ext. - M.: Publishing LCI, 2007. - 328 p.
2. Kalabekov BA "Digital devices and microprocessor systems" - Hotline - Telecom, 2003. – 336 p.
3. Frenkel BS "Design of digital devices in CAD XILINX WebPACK ISE» - Gomel BelSUT, 2006. - 54 p.
4. The program of ISE Design Suite v.14.4.

Публикация:

ISSN 0142 - 0843

Қарағанды университетінің хабаршысы. МАТЕМАТИКА сериясы. № 4 (80) / 2015

Вестник Карагандинского университета. Серия МАТЕМАТИКА. № 4 (80) /2015