

Б.К.Жармакин

Евразийский национальный университет им. Л.Н.Гумилева, Астана
(E-mail: zbk_60@mail.ru)

Примеры программирования элементов цифровой электроники на языке VHDL в среде XILINX

В статье приведены примеры программирования на языке VHDL различных логических элементов цифровой электроники, а также 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. Автор предлагает временные диаграммы рассматриваемых логических элементов.

Ключевые слова: цифровая интегральная схема, логический элемент, логическая переменная, логическая функция, дизъюнкция, временная диаграмма, мультиплексор, шифратор.

Целью данной статьи является объединение двух составляющих современной цифровой электроники. При изучении цифровых элементов программное обеспечение данных устройств обычно не рассматривается [1]. Также при изучении основ программирования упор делается на языки высокого уровня [2–4]. 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 умеющих 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 микроэлектронное 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».

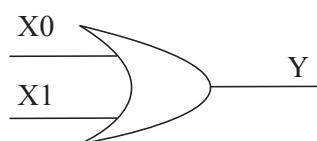
Здесь для примера рассмотрим одну из логических функций.

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»:

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

ЛЭ, реализующий функцию «Дизъюнкция», называют дизъюнктором, или ЛЭ «ИЛИ». Условное изображение и временные диаграммы ЛЭ «ИЛИ» приведены на рисунке 1 а, в, г.

а)



б)

X0	X1	Y
0	0	0
0	1	1
1	0	1
1	1	1

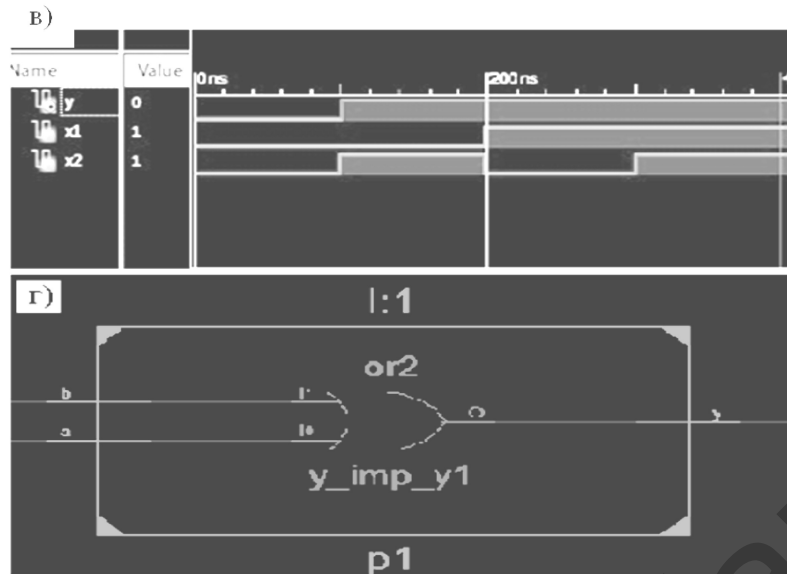
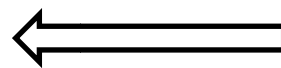


Рисунок 1. Условное графическое обозначение дизъюнктора по ANSI (а), таблица истинности инвертора (б), временная диаграмма, полученная в среде XILINX (в), обозначение в среде XILINX (г)

```
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 (x1, x2: in bit;
y: out bit);
end I;
architecture Behavioral of I is
begin
y<=a or b after 1 ns;
v) behavioral;v)

```



Описание портов ввода/вывода: **in bit** – два (x1,x2); **out bit** – выходной (y) порт дизъюнктора

Присвоение входным портам x1 и x2 значений переменных а и б, выходному порту у; значение w1 в карте порта (port map); p1 – ЛЭ (дизъюнктор)

Присвоение функции у 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 — мультиплексу.

Мультиплексором (Multiplexer) называют комбинационное устройство с М информационными (X₀, X₁, ..., X_{M-1}) и К адресными (A₀, A₁, ..., A_{K-1}) входами и одним выходом (Y), которое осуществляет передачу сигнала с заданного адресным кодом информационного входа на его выход (рис. 2).

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.

Рассмотрим мультиплексор 4×1 с инверсным входом разрешения E и прямым выходом Y , представляющий собой половину микросхемы мультиплексора КР555КП2 (рис. 3).

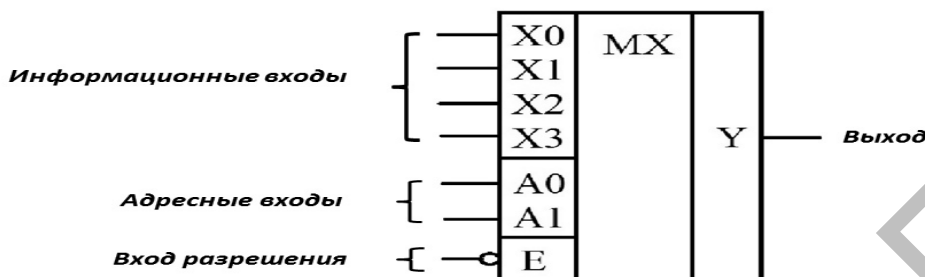


Рисунок 3. Условное графическое обозначение мультиплексора 4×1

Выражение для выходной функции такого мультиплексора можно записать в виде

$$Y = \bar{E} \wedge (X0 \wedge \bar{A0} \wedge \bar{A1} \vee X1 \wedge A0 \wedge \bar{A1} \vee X2 \wedge \bar{A0} \wedge A1 \vee X3 \wedge A0 \wedge A1). \quad (1)$$

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):

$$Y = \bar{E} \wedge (X0 \wedge \bar{A0} \wedge \bar{A1} \vee X1 \wedge A0 \wedge \bar{A1} \vee X2 \wedge \bar{A0} \wedge A1 \vee X3 \wedge A0 \wedge A1)$$

Логическое И
Логическое И

Логическое И
Логическое И

Логическое ИЛИ

Рисунок 4. Необходимые логические элементы

Для построения принципиальной схемы мультиплексора нам необходимы 4 элемента И, для информационных входов — $X0, X1, X2, X3$ и один элемент ИЛИ (рис. 4).

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

На рисунке 5 показаны 4 элемента И, 3 элемента НЕ и 1 элемент ИЛИ.

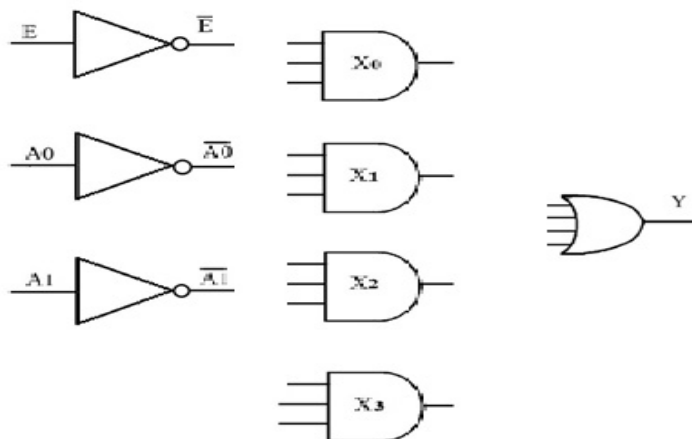


Рисунок 5. Логические элементы, необходимые для рисования схемы

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 X_0 , учитывая, что на его вход должны быть поданы 3 сигнала согласно формуле (1). Сигнал разрешения E уже у нас имеется. Сигналы с адресных входов A_0 и A_1 инверсные. Поэтому входы X_0 соединяются линиями с инверсных входов инверторов (рис. 6).

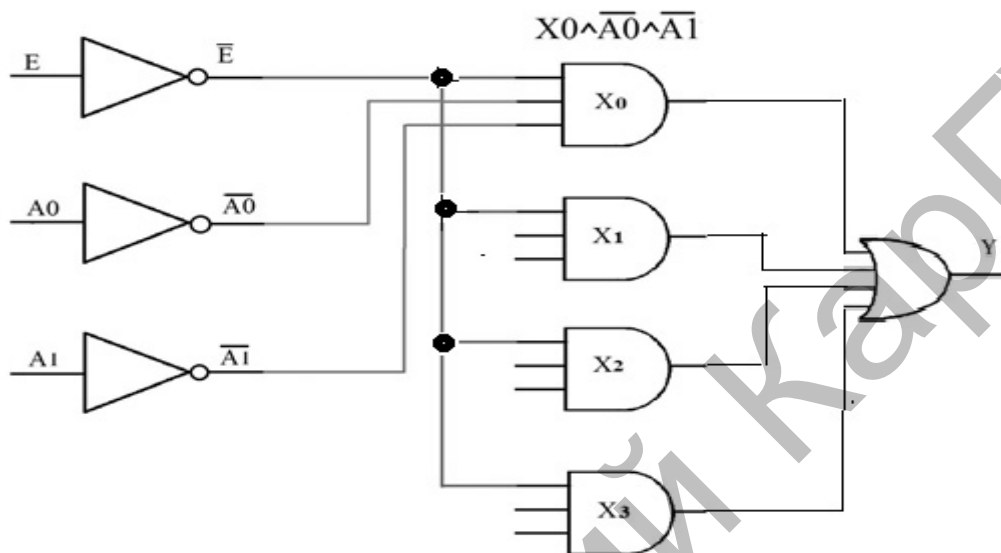


Рисунок 6. Линии соединения для информационного входа X_0

Аналогично вычерчиваем соединения для информационных входов мультиплексора X_1 , X_2 , X_3 . Далее, соединив выходы информационных входов мультиплексора X_0 , X_1 , X_2 , X_3 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.

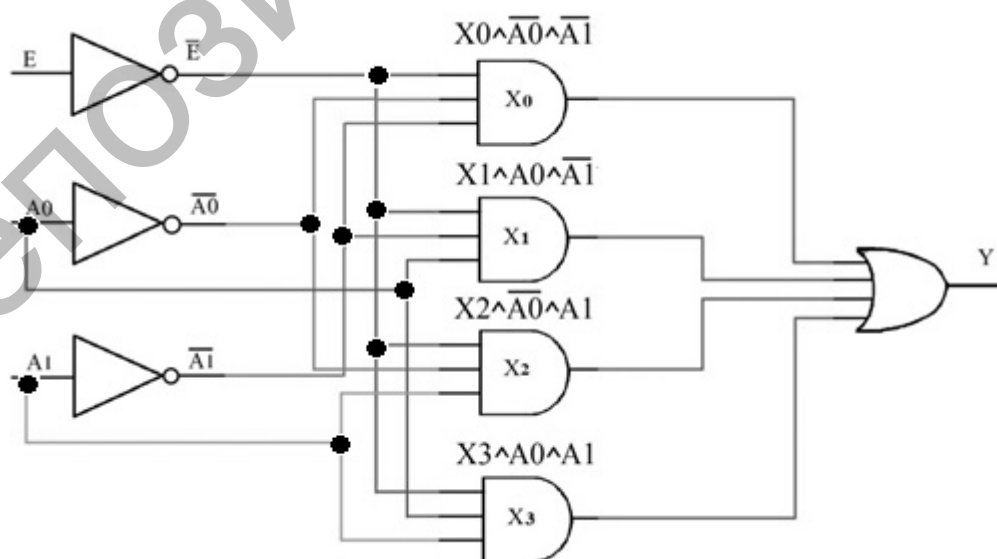


Рисунок 7. Линии соединения для информационных входов мультиплексора X_0 , X_1 , X_2 , X_3

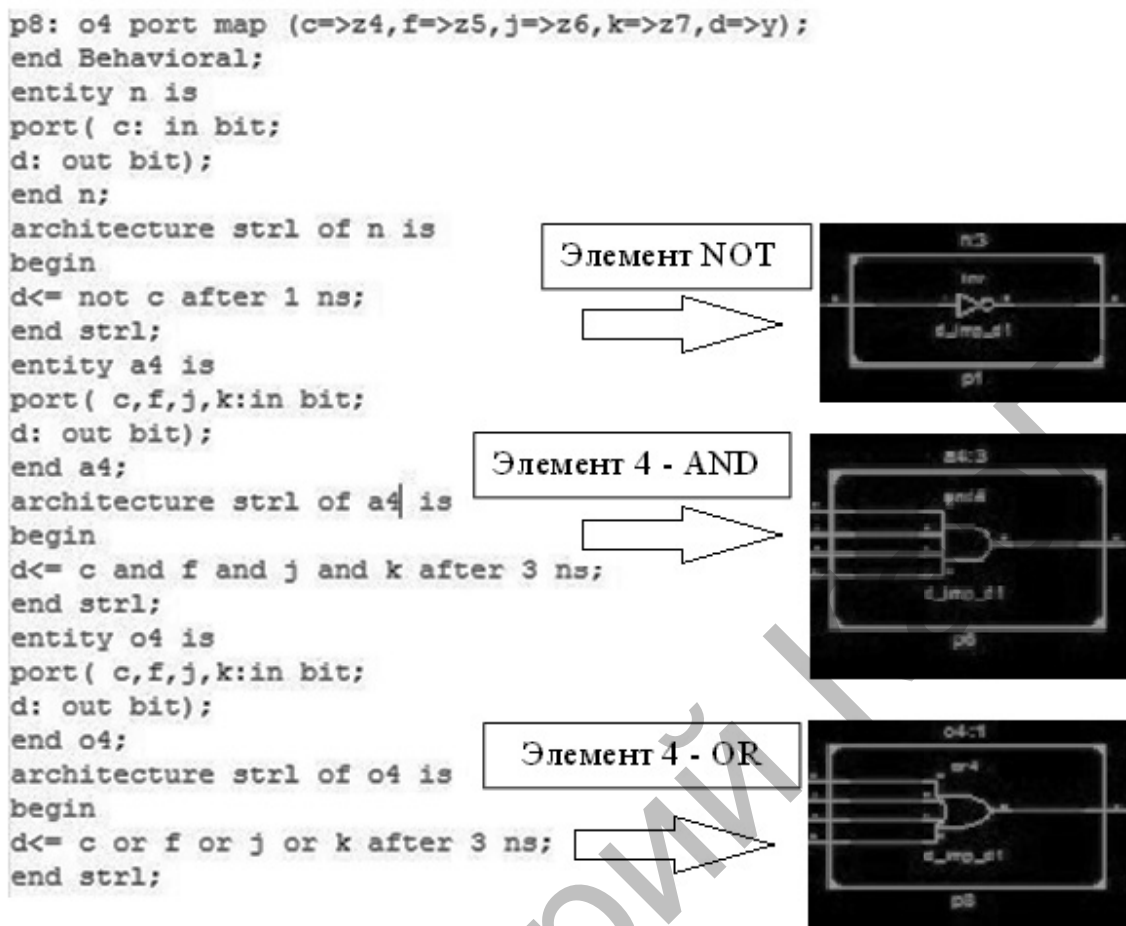


Рисунок 8. Фрагмент программы на языке VHDL для мультиплексора

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 со значениями на адресных входах a и b (A0 и A1) на входах X0 — X3 имеем сигнал логической единицы, который отражается на выходе Y.

Сравнивая данные таблицы 1 и данные временной диаграммы, можно увидеть их полное соответствие, что и ставилось целью в данной статье.

Таблица 1

Таблица истинности для мультиплексора 4×1

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

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

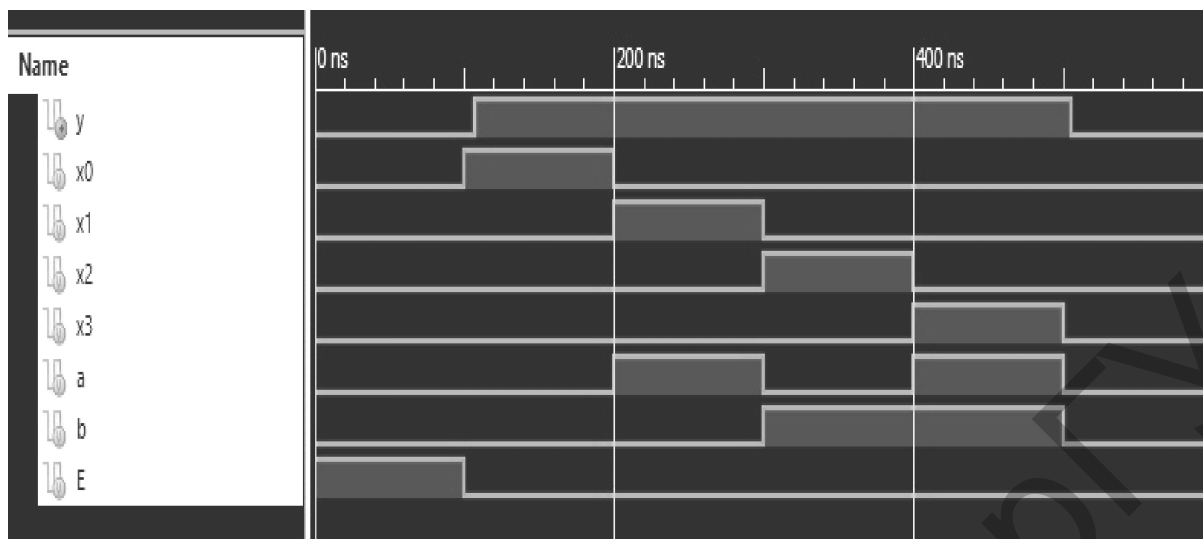


Рисунок 9. Временная диаграмма, полученная в среде XILINX

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

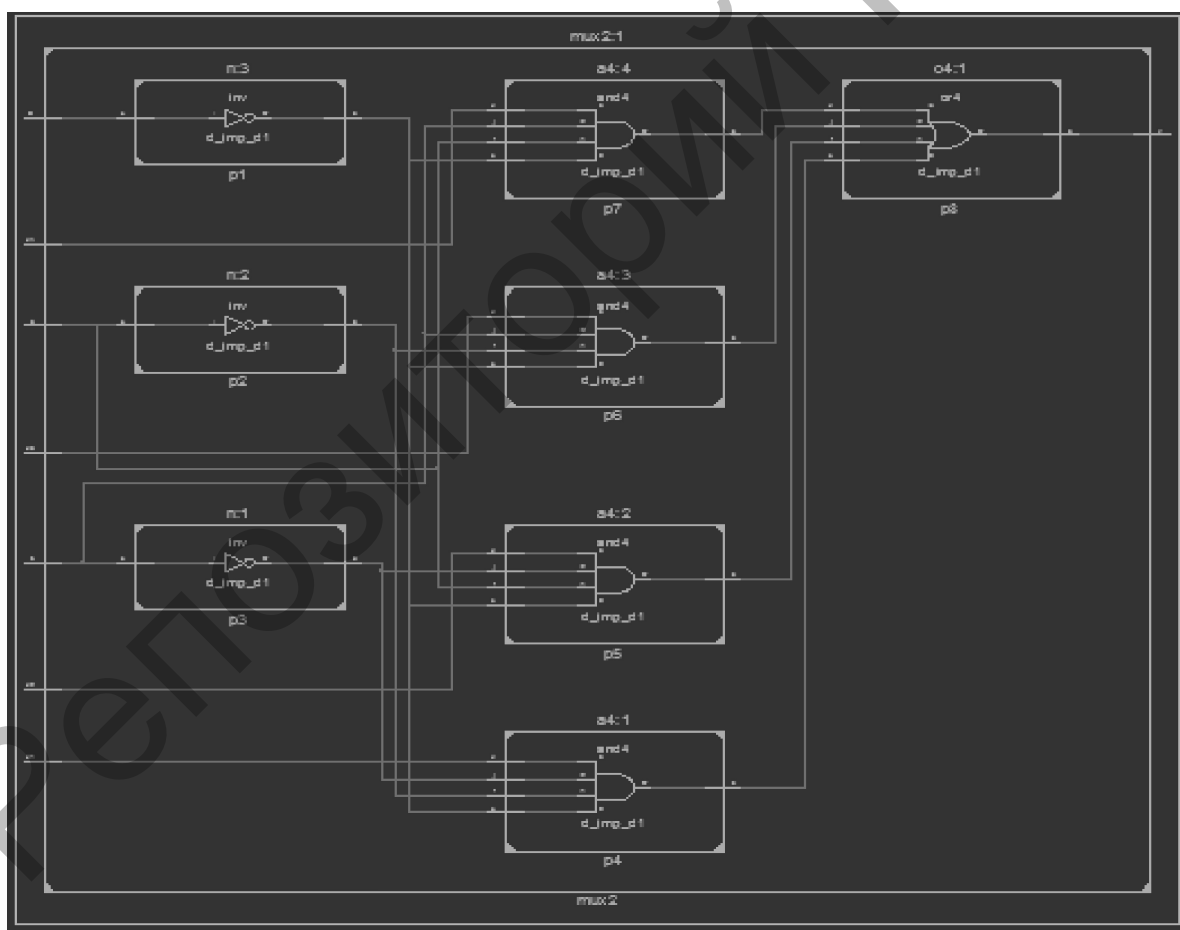


Рисунок 10. Принципиальная схема мультиплексора, полученная в среде XILINX

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.

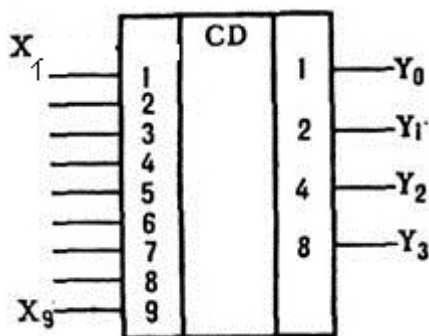


Рисунок 11. Условное обозначение шифратора

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 (рис. 11).
 Работа устройства иллюстрируется таблицей 2.

Таблица 2

Таблица истинности для шифратора

Входы									Выходы			
X9	X8	X7	X6	X5	X4	X3	X2	X1	Y3	Y2	Y1	Y0
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 можно записать в виде

$$\begin{aligned}
 Y_0 &= X_1 \wedge X_3 \wedge X_5 \wedge X_7 \vee X_9; \\
 Y_1 &= X_2 \wedge X_3 \wedge X_6 \wedge X_7; \\
 Y_2 &= X_4 \wedge X_5 \wedge X_6 \wedge X_7; \\
 Y_3 &= X_8 \wedge X_9.
 \end{aligned}
 \tag{2}$$

Для начала определим, какие логические элементы понадобятся, чтобы собрать схему шифратора. Для этого разложим на логические элементы формулу (2) (рис. 12).

$$\begin{aligned}
 Y_0 &= \underbrace{X_1 \wedge X_3 \wedge X_5 \wedge X_7 \vee X_9}_{\text{Логическое ИЛИ}} & Y_1 &= \underbrace{X_2 \wedge X_3 \wedge X_6 \wedge X_7}_{\text{Логическое ИЛИ}} \\
 Y_2 &= \underbrace{X_4 \wedge X_5 \wedge X_6 \wedge X_7}_{\text{Логическое ИЛИ}} & Y_3 &= \underbrace{X_8 \wedge X_9}_{\text{Логическое ИЛИ}}
 \end{aligned}$$

Рисунок 12. Необходимые логические элементы

Итак, 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 (рис. 13).

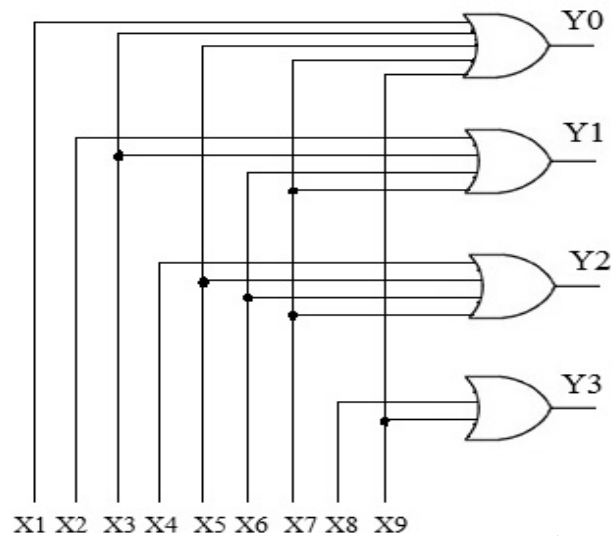


Рисунок 13. Принципиальная схема шифратора

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

```

entity shifator is
port (x1,x2,x3,x4,x5,x6,x7,x8,x9:in bit;
      y0,y1,y2,y3:out bit);
end shifator;
architecture str1 of shifator 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

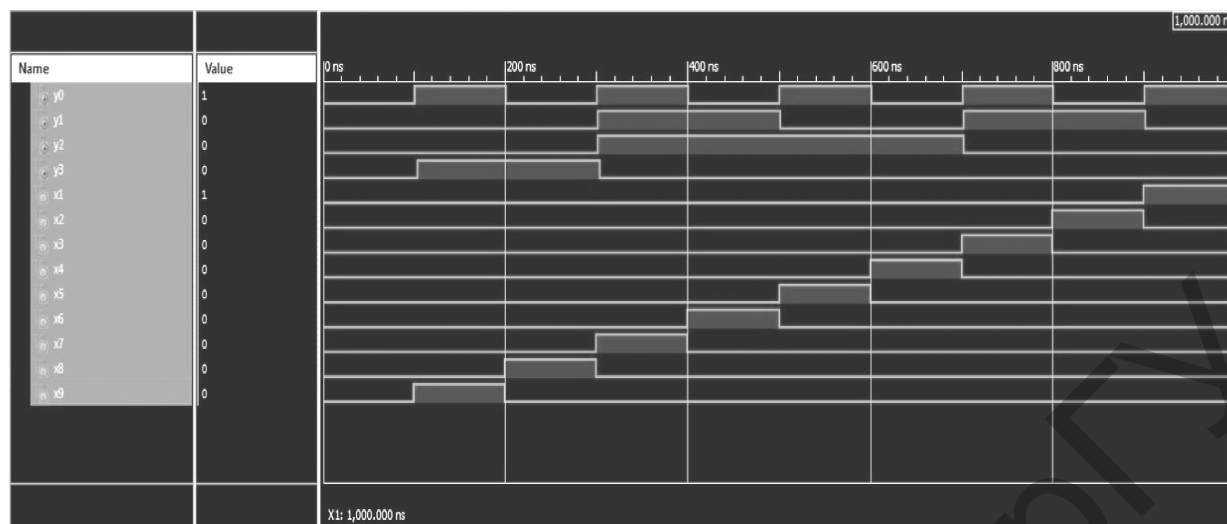


Рисунок 15. Временная диаграмма шифратора в программной среде XILINX

Вывод

Для улучшения освоения студентами дисциплин «Цифровая электроника» и «Программирование на языке VHDL» желателен симбиоз данных предметов. В данной статье нами была предпринята попытка объединения названных выше дисциплин.

Благодарен за критические замечания и приглашения к сотрудничеству в данной сфере.

Список литературы

- 1 Бибило П.Н. Основы языка VHDL. — 3-е изд., доп. — М.: Изд-во ЛКИ, 2007. — 328 с.
- 2 Калабеков Б.А. Цифровые устройства и микропроцессорные системы. — Горячая линия — Телеком. — 2003. — 336 с.
- 3 Френкель Б.С. Проектирование цифровых устройств в САПР XILINX WebPACK ISE. — Гомель: БелГУТ, 2006. — 54 с.
- 4 Программа ISE Design Suite v.14.4. — [ЭР]. Режим доступа: <http://www.xilinx.com/products/design-tools/ise-design-suite.html>

Б.К.Жармакин

XILINX ортасында сандық электроника элементтеріне VHDL тілінде бағдарлама құру мысалдары

Мақалада VHDL тілін қолданып, әр түрлі сандық логикалық элементтер үшін бағдарламаларға мысалдар келтірілген. Логикалық элементтердің шартты графикалық сұлбалары, ақиқат кестелері, бағдарламаларға мысалдар мен түсініктемелер берілген. Бағдарламалық кодтардың XILINX ортасында алынған шешімдері бар. Автор XILINX ортасында элементтердің шартты графикалық сұлбаларын келтіріліп, қарастырылған логикалық элементтердің уақыттық диаграммаларын ұсынған.

B.K.Zharmakin

Examples of programming of elements of digital electronics in language of VHDL in environment of XILINX

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.

References

- 1 Bibilo P.N. *Basics of VHDL* / Edit third, ext., Moscow: Publ. LCI, 2007, 328 p.
- 2 Kalabekov B.A. *Digital devices and microprocessor systems*, Hotline, Telecom, 2003, 336 p.
- 3 Frenkel B.S. *Design of digital devices in CAD XILINX WebPACK ISE*, Gomel: BelSUT, 2006, 54 p.
- 4 *The program of ISE Design Suite v.14.4.* [ER]. Access mode: Режим доступа: <http://www.xilinx.com/products/design-tools/ise-design-suite.html>

Репозиторий КарГУ