

Лабораторная работа №2

Изучение устройств ввода/вывода. Мультиплексоры, дешифраторы.

В данной лабораторной работе будут рассмотрены стандартные устройства ввода-вывода, входящие в состав учебного стенда Altera DE1, а также применение некоторых из них на примере их использования совместно с дешифраторами и мультиплексорами.

Описание лабораторной работы.

Учебный стенд Altera DE1 имеет в своём составе следующий набор устройств, а также цифровых и аналоговых портов ввода-вывода информации, подключенных к ПЛИС Cyclone II:

- MIC, микрофонный аудио вход;
- LINE IN, линейный аудио вход;
- LINE OUT, линейный аудио выход;
- VGA, видео разъем;
- RS-232, разъем последовательного интерфейса;
- PS2, разъем интерфейса PS/2;
- SD CARD, разъем карты памяти SD;
- LEDG, 8 зеленых светодиодов;
- KEY, 4 кнопочных переключателя;
- SW, 10 переключателей;
- LEDR, 10 красных светодиодов;
- HEX, четыре 7-сегментных индикатора;
- SDRAM, модуль динамической оперативной памяти SDRAM объемом 8 МБайт;
- SRAM, модуль статической оперативной памяти SRAM объемом 512 КБайт;
- FLASH, модуль энергонезависимой флэш-памяти объемом 4 МБайт.

Коммутация между устройствами, обработка получаемой информации и установка выходных сигналов производится посредством конфигурирования ПЛИС.

При выполнении лабораторной работы вам будет необходимо сконфигурировать ПЛИС для работы с переключателями *SW*, светодиодами *LED* и блоком семисегментных индикаторов *HEX*.

Лабораторное задание.

Разработать в среде Quartus II Verilog-описание блоков мультиплексора и дешифраторов, выполнить моделирование и сконфигурировать ПЛИС для проверки функциональности блоков *LED*, *SW* и *HEX* учебного стенда.

Часть 1. Использование устройств ввода/вывода. Настройка, компиляция и проверка работоспособности проекта с помощью временных диаграмм.

На стенде представлены 10 переключателей *SW*₉₋₀, 10 светодиодов *LEDR*₉₋₀ и 8 светодиодов *LEDG*₇₋₀, которые могут быть использованы как устройства ввода и вывода соответственно. Каждый из переключателей *SW* подключен к ПЛИС и имеет два положения: «логическая единица» и «логический ноль», определяемые напряжением на выходе

переключателя. Высокое напряжение (от 1,8 до 3,3 В), определяемое напряжением портов ввода/вывода ПЛИС – логическая «единица», низкое (~ 0 В) – логический «ноль». Ниже описана простая конструкция на языке Verilog, использующая переключатели SW , состояние которых отображается на красных светодиодах $LEDR$:

```
assign LEDR[9] = SW[9];
assign LEDR[8] = SW[8];
...
assign LEDR[0] = SW[0];
```

Количество переключателей и индикаторов одинаково, поэтому удобно представить их в виде векторов в Verilog-описании. В этом случае необходимо использовать только одну команду присваивания, что будет эквивалентно 10-ти индивидуальным присваиваниям.

```
input [9:0] SW;           // переключатели
output [9:0] LEDR;        // красные светодиоды
assign LEDR = SW;
```

ПЛИС Cyclone II, установленная на стенде, аппаратно подключена к переключателям и светодиодам. Чтобы использовать $SW_{9:0}$ и $LEDR_{9:0}$ необходимо корректно назначить выводы ПЛИС в соответствии со схемой подключения. Например, согласно спецификации на стенд, SW_0 подключен к выводу $L22$ ПЛИС, а $LEDR_0$ — к выводу $R20$. Корректное назначение выводов описано в файле DE_pin_assignments.csv, который необходимо добавить в проект. Важно помнить, что имена выводов, определённые файле *.csv должны совпадать с именами, используемыми в Verilog-описании. В файле DE_pin_assignments.csv заданы имена $SW[0]$... $SW[9]$ и $LEDR[0]$... $LEDR[9]$ для переключателей и диодов.

```
// Simple module that connects the SW switches to the LEDR lights
module part1 (SW, LEDR);
input [9:0] SW;
output [9:0] LEDR; // red LEDs
assign LEDR = SW;
endmodule
```

Выполните следующие пункты для моделирования и конфигурации ПЛИС, соответствующей модулю part1:

1. Запустите среду Quartus II. Выберите *File -> New Project Wizard*.
2. В появившемся окне нажмите *Next*. Измените при необходимости текущую рабочую директорию проекта и укажите имя проекта, нажмите *Next -> Next*.
3. В выпадающем списке *Device Family: Family* выберите *Cyclone II*, в перечне *Available devices* выберите *EP2C20F484C7*. Нажмите *Finish*.
4. Выберите *File -> New*. В появившемся окне выберите *Verilog HDL File*, нажмите *OK*. В проект будет добавлен новый файл с расширением *.v и откроется текстовый редактор этого файла.
5. Добавьте код описанного выше модуля в тело файла и выберите *File -> Save*. Сохраните файл в рабочую папку проекта под именем *part1.v*. Внимание, имя файла

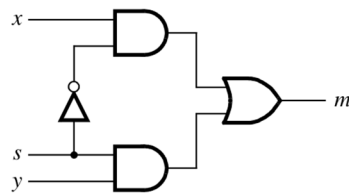
- должно совпадать с именем модуля, описанного в нём!
6. Скомпилируйте проект: *Processing* -> *Start Compilation*. При наличии ошибок, исправьте их и повторно скомпилируйте проект.
 7. Промоделируйте работу модуля, построив временную диаграмму средствами Quartus II (в последующих лабораторных работах будут рассмотрены возможности пакета ModelSim для построения временных диаграмм). Выберите *File* -> *New*. В появившемся окне выберите *Vector Waveform File* и нажмите *OK*.
 8. В открывшемся редакторе временных диаграмм щелкните правой кнопкой мыши в поле списка узлов (*Name*) и выберите *Insert* -> *Insert Node or Bus*. В появившемся окне нажмите *Node Finder*. В выпадающем списке *Filter* выберите *Pins: all* и нажмите кнопку *List*. Перенесите необходимые узлы из поля *Nodes Found* в поле *Selected Nodes*. Нажмите *OK* -> *OK*.
 9. В пункте меню *Edit* -> *End Time* и *Edit* -> *Grid Size* можно выбрать продолжительность моделирования и размер сетки. Установите *End Time* = 30 us, *Grid Size* = 1us.
 10. Изменяя значение входных сигналов SW, вы моделируете работу переключателей. Результат моделирования - выходные сигналы LEDG имитируют состояние светодиодов. Задайте входные сигналы SW и сохраните файл *.vwf в папку с проектом. Выберите *Assignments* -> *Settings*. *Category: Simulator Settings*, в поле *Simulation input* укажите пусть до сохраненного файла *.vwf. Таким образом, вы указываете входные параметры моделирования для симулятора работы ПЛИС.
 11. Если временная диаграмма-результат моделирования корректна, переходите к следующему пункту. Иначе отредактируйте Verilog-модуль для получения корректной диаграммы.
 12. Добавьте файл DE_pin_assignments.csv в проект. Для этого выберите *Assignments* -> *Import Assignments*. В появившемся окне укажите путь до файла DE_pin_assignments.csv и нажмите *OK*.
 13. Скомпилируйте проект: *Processing* -> *Start Compilation*. При наличии ошибок, исправьте их и повторно скомпилируйте проект.
 14. Как только будут устранены все ошибки, сконфигурируйте ПЛИС: выберите *Tools* -> *Programmer*. В появившемся окне нажмите кнопку *Hardware Setup* и в выпадающем списке *Currently selected hardware* выберите *USB-Blaster*. Нажмите *Close*.
 15. Кнопка *Start* должна стать активной. Нажмите её – начнётся процесс конфигурации ПЛИС.
 16. По окончании процесса конфигурации проверьте работу стенда: при изменении положения переключателей SW соответствующие диоды должны загораться.

Часть 2. Мультиплексоры

Мультиплексором называют комбинационную схему, которая обеспечивает коммутацию сигнала с одного из множества входов на единственный выход. Выбор коммутируемого сигнала происходит с помощью управляющего сигнала.

На рисунке 1 а показана схема, реализующая мультиплексор 2-1 с управляющим сигналом s . Если $s=0$, на выход m мультиплексора поступает сигнал с входа x , иначе — сигнал с входа y .

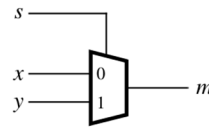
Также на рисунке 1 б представлена таблица истинности данной схемы и её символ (графический примитив) в Quartus II (см. рис. 1 с).



a) Circuit

s	m
0	x
1	y

b) Truth table



c) Symbol

Рисунок 1. Мультиплексор 2-1

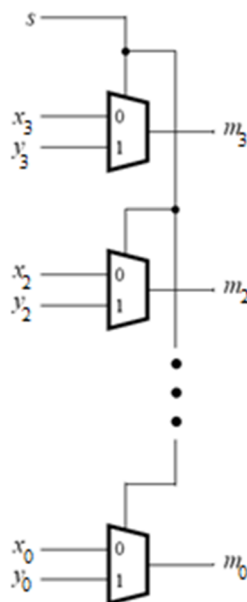
а) схема, б) таблица истинности, в) графический примитив

При помощи языка Verilog мультиплексор может быть описан таким образом:

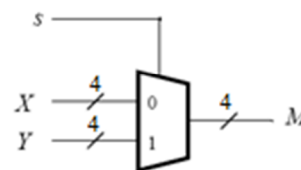
assign $m = (\sim s \ \& \ x) \mid (s \ \& \ y);$

Напишите Verilog-модуль, который опишет приведённую ниже схему.

На схеме представлены 4-х битные входы X и Y , 4-х битный выход M и управляющий сигнал s . Если $s=0$, то $M=X$, иначе (если $s=1$) $M=Y$. В упрощенном варианте можно использовать только один символ мультиплексора, с подключенными к нему 4-х битными шинами X , Y и M , а также сигналом s . Схемы на рисунках 2а и 2б – эквивалентны.



a) Circuit



b) Symbol

Рисунок 2 - Мультиплексирование 4-разрядных шин

Теперь опишите представленную схему, используя язык Verilog. В качестве сигнала s используйте переключатель SW_8 , в качестве X – переключатели SW_{3-0} , а в качестве Y – переключатели SW_{7-4} . Выход m подключите к зелёным светодиодам $LEDG_{7-4}$, а

переключатели SW к красным светодиодам $LEDR$.

Нужное описание может выглядеть следующим образом:

```
module mul2_1 (SW, LEDG, LEDR);  
...  
assign LEDG[7:4] = ( {4{~SW[8]}} & SW[3:0]) | ({4{SW[8]}} & SW[7:4]);
```

Обратите внимание на приведённый в этом описании способ «размножения» сигналов $SW[8]$ и $\sim SW[8]$, который реализуется с помощью оператора объединения $\{ \}$.

Скомпилируйте и промоделируйте проект, а также сконфигурируйте ПЛИС как описано в части 1. Проверьте результат работы мультиплексора по состоянию светодиодов $LEDG$ в зависимости от управляющего сигнала SW_8 и положения $SW_{7:0}$.

Часть 3. Дешифраторы

Двоичным дешифратором (*англ.* **decoder**) называют комбинационную схему, которая ставит в соответствие каждому возможному входному вектору $[x_{n-1} \dots x_0]$ заранее определенный выходной вектор y , разрядность которого не более 2^n . В случае если подключены все 2^n выходов, дешифратор называют **полным**, в противном случае – **неполным**. Дешифраторы активно применяются при разработке цифровых устройств. В большинстве цифровых устройств в явном или неявном виде можно встретить дешифратор.

Так, например, одним из наиболее распространённых устройств является дешифратор для преобразования 4-х разрядного кода в значение для отображения на 7-сегментном индикаторе. На рисунке 3 представлен дешифратор для выполнения необходимого преобразования с 4-х битным входом $c_{3:0}$. На основании входного сигнала декодер вырабатывает 7-ми разрядный сигнал для отображения символов на 7-сегментном индикаторе. Таблица 1 иллюстрирует символы, которые должны быть отображены на дисплее в зависимости от сигнала c .

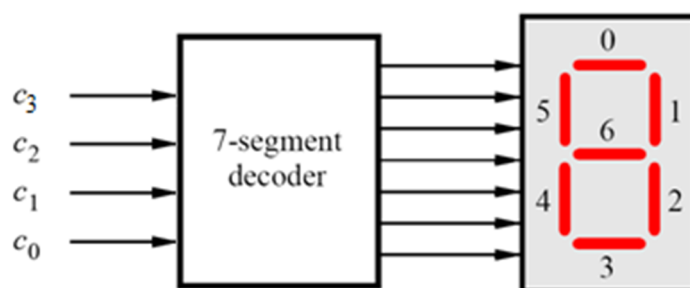


Рисунок 3 – 7-сегментный индикатор с дешифратором

Таблица 1. Таблица декодирования для 7-сегментного индикатора

$c_3c_2c_1c_0$	Символ	7-ми разрядный код
0000	0	7'b1000000
0001	1	7'b1111001
0010	2	7'b0100100
0011	3	7'b0110000
0100	4	7'b0011001
0101	5	7'b0010010
0110	6	7'b0000010
0111	7	7'b1111000
1000	8	7'b0000000

1001	9	7'b0010000
1010	A	7'b0001000
1011	b	7'b0000011
1100	C	7'b1000110
1101	d	7'b0100001
1110	E	7'b0000110
1111	F	7'b0001110

Дешифратор рекомендуется выполнить при помощи конструкций *always* и *case*.

Сегменты индикатора проиндексированы в соответствии с рисунком 3. Каждый сегмент горит при логическом «0» на входе, при логической «1» - не горит. В качестве c_{3-0} используйте SW_{3-0} . Подключите 7-сегментный индикатор следующим образом:

output reg [6:0] HEX0;

Для конфигурации ПЛИС выполните действия, описанные в части 1. Проверьте результат работы мультиплексора по состоянию дисплея *HEX0* в зависимости от положения переключателей SW_{3-0} .

Индивидуальное задание.

В качестве индивидуального задания предлагается сделать 4 дешифратора.

На стенде имеются четыре кнопочных переключателя KEY0 - KEY3, а также четыре 7-сегментных дисплея HEX0 – HEX3. Необходимо, чтобы на дисплеях отображались первые четыре буквы имени по нажатию соответствующей кнопки. Для HEX0 – первая буква и KEY0 и т.д. Добавьте возможность отображения сразу всех символов имени с помощью SW[9]. Обратите внимание, что значения с кнопок – инверсные, т.е. при нажатии на кнопку соответствующий ей сигнал KEY принимает значение логического нуля.

Требования к отчету.

В качестве отчета о выполнении лабораторной работы вы должны представить проект, содержащий Verilog-описания предлагаемых к разработке модулей, временные диаграммы с результатом моделирования, а также продемонстрировать работу разработанных модулей на учебном стенде и ответить на вопросы преподавателя.