



Звіт

про виконання лабораторної роботи №1
з дисципліни:
“Моделювання комп’ютерних систем”

Виконав: ст. гр. КІ-202
Ключко Д.С.
Прийняв:
Старший викладач Козак Н.Б.

Мета: ознайомлення з середовищем розробки Xilinx ISE та побудова дешифратора 3-7.

Лабораторна робота №1

**Інсталяція та ознайомлення з середовищем розробки Xilinx ISE.
Ознайомлення зі стендом Elbert V2 – Spartan 3A FPGA.**

Етапи роботи:

1. Інсталяція Xilinx ISE.
2. Побудова дешифратора 3->7 за допомогою ISE WebPACK™ Schematic Capture та моделювання його роботи за допомогою симулятора ISim.
3. Генерування .bit файла та тестування за допомогою стенда Elbert V2 – Spartan 3A FPGA.

Виконання завдання

Згідно завдання було розроблено схему дешифратора 3 в 7, використовуючи компоненти з бібліотеки, у новому, доданому до проекту Schematic файлу

My2To4Decoder.sch

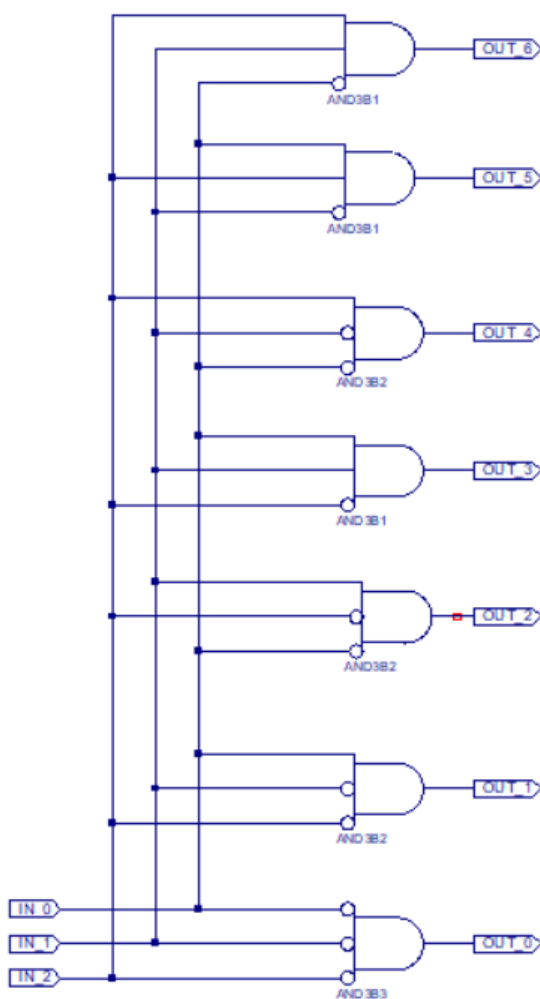


Рис.1. Схема дешифратора 3 в 7

Згодом було додану до проекту User Constraint файл, в якому було додано змінений вміст файлу elbertv2.ucf.

ConstraintsF.ucf

```

8  #*****
9  #                               UCF for ElbertV2 Development Board
10 #*****
11 CONFIG VCCAUX = "3.3" ;
12
13 # Clock 12 MHz
14 #NET "Clk"                LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
15
16
17 #####
18 #                               LED
19 #####
20
21 NET "OUT_0"                LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
22 NET "OUT_1"                LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
23 NET "OUT_2"                LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
24 NET "OUT_3"                LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
25 NET "OUT_4"                LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
26 NET "OUT_5"                LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
27 NET "OUT_6"                LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
28 |#NET "OUT_7"              LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
29
30 #####
31 #                               DP Switches
32 #####
33
34 NET "IN_0"                 LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
35 NET "IN_1"                 LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
36 NET "IN_2"                 LOC = P68 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
37 #NET "DPSwitch[3]"         LOC = P64 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
38 #NET "DPSwitch[4]"         LOC = P63 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
39 #NET "DPSwitch[5]"         LOC = P60 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
40 #NET "DPSwitch[6]"         LOC = P59 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
41 #NET "DPSwitch[7]"         LOC = P58 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

```

Рис.2. User Constraint файл

Результат симуляції моделювання дешифратора

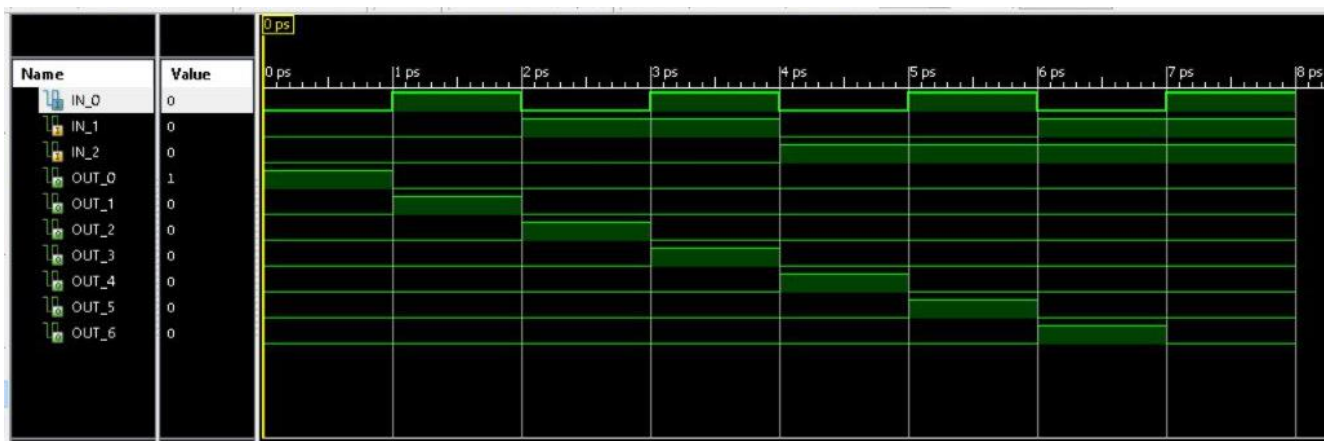


Рис.3. Запуск симуляції

Згодом було створено конфігураційний файл і були послідовно запуснені процеси, щоб переконатися що всі процеси успішно виконалися.

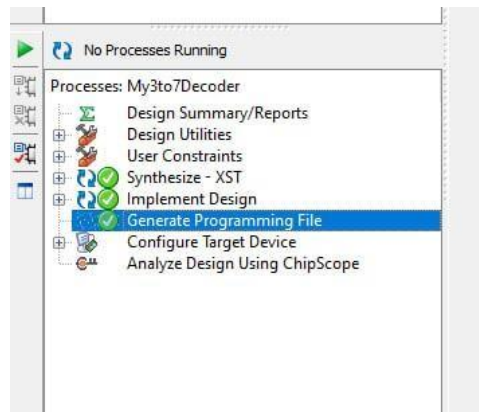


Рис.4. Виконання процесів

Далі під'єднуємо ПЛІС Elbert V2 Spartan 3A до нашого комп'ютера за допомогою спеціального роз'єма. Після цього кроку, заходимо в диспетчер пристроїв та шукаємо наш ПЛІС у диспетчері пристроїв.

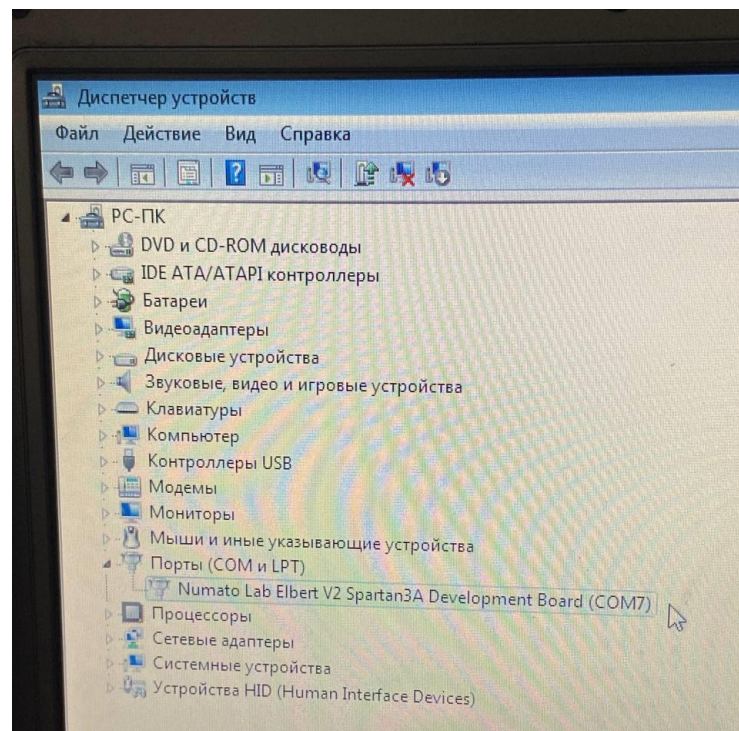


Рис.5. ПЛІС в диспетчері пристроїв

Далі підготуємо нашу ПЛІС до завантаження прошивки. Для цього відкриємо програму – прошивальник та виберемо СОМ порт, який відображений в диспетчері пристроїв (В даному випадку СОМ7) також не забуваємо про створений раніше файл з розширенням bin його ми завантажувемо через спеціальне вікно програми. Виконавши всі ці етапи програмуємо нашу ПЛІС.

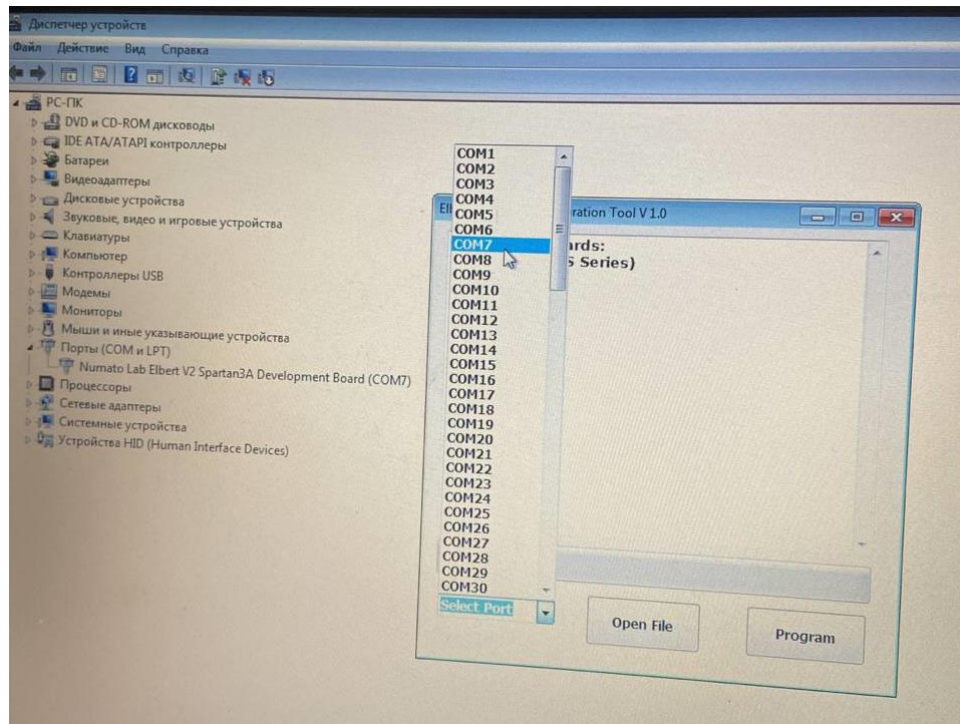


Рис.6. Вибір COM порту та завантаження bin-файлу з прошивкою

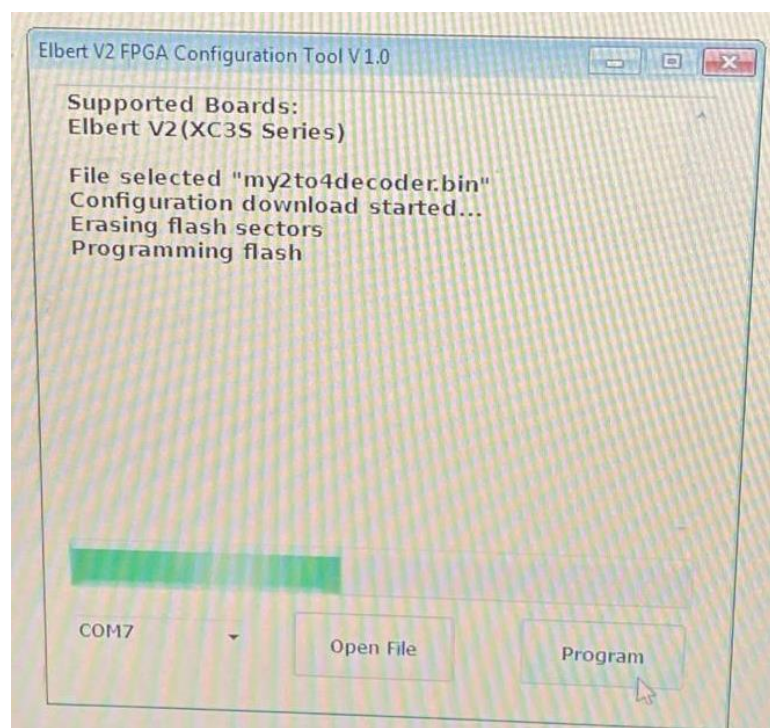


Рис.7. Процес прошивання Elbert V2 Spartan 3A

Висновок: Під час виконання даної лабораторної роботи я ознайомився з середовищем розробки **Xilinx ISE**, навчився працювати з засобами управління проектом. Також створив схему дешифратора **3->7** та перевірів його роботу в режимі симуляції. Ще пригадав, як працювати з логічними елементами **AND**.