ECE2810 Digital Systems Design Laboratory

Lab 6

FPGA and VHDL

School of Science and Engineering

The Chinese University of Hong Kong, Shenzhen

2024, Spring term

1. Objectives

In Lab 6, we will spend the 3-week session on the following:

- Learn to programme Xilinx FPGA for digit logic function.
- Learn to programme Xilinx FPGA for multiple 7-segment outputs simultaneously displayed.
- Learn to programme Xilinx FPGA to divide high-frequency clocks into low-frequency clocks.
- Learn to programme Xilinx FPGA for a count-down timer.

2. Introduction

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL).

FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

The FPGA development board we will use is produced by E-Elements, with Xilinx ARTIX-7 FPGA IC.

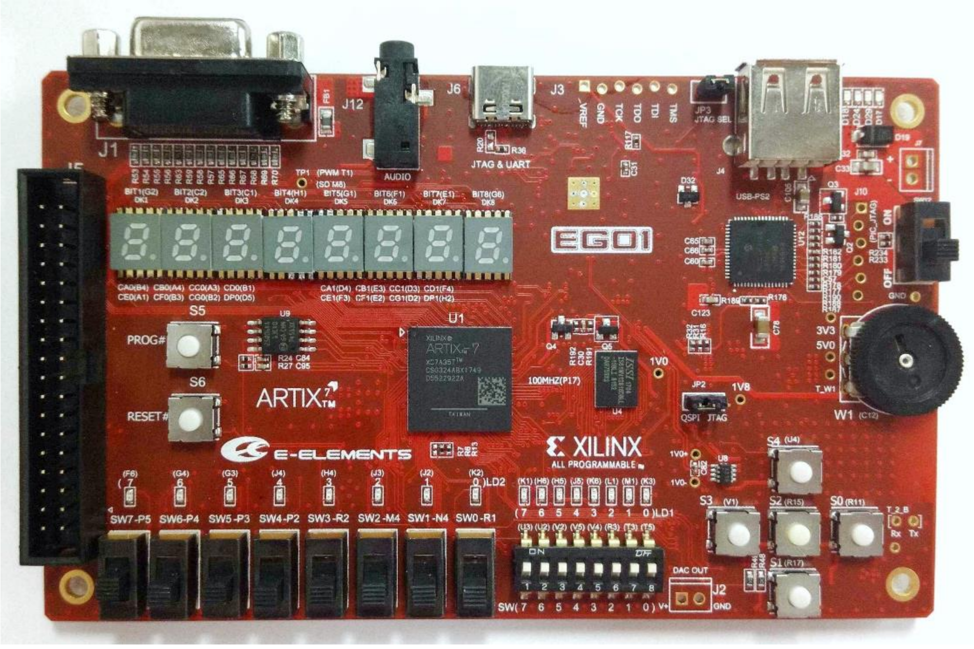


Fig. 1 EGO1 Xilinx FPGA

3. Experiments

3.1 **Experiment 1:** Hands-on Practice

The first experiment is a hands-on practice to build up a simple 8-channel AND gate.

1) Find and double click the shortcut of "Vivado" on the desktop.

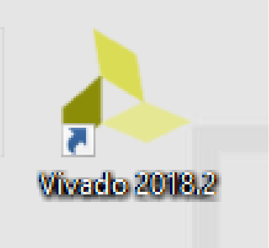


Fig. 2

2) It takes a few seconds to launch the GUI as below.

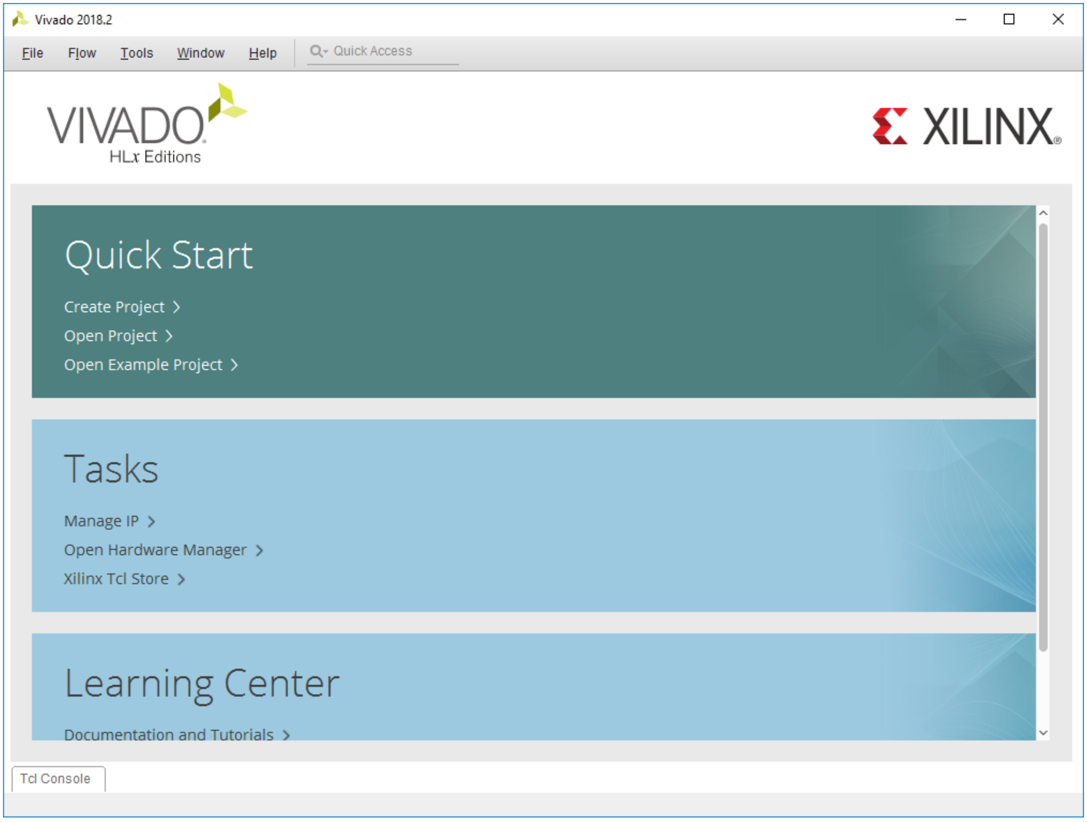
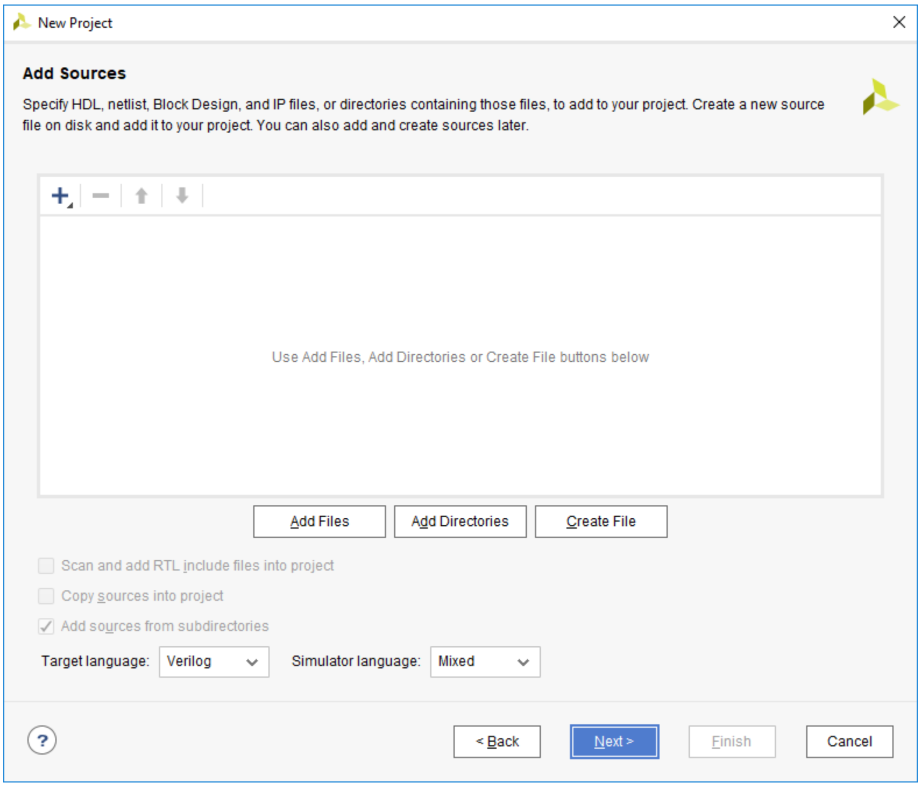


Fig. 3

3) Left-click “Create Project” => “Next” => input the project name, e.g. “MyAndGate”=> keep the default items => “Next”, and you will see the dialogue box below.



New Project

Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Use Add Files, Add Directories or Create File buttons below

☐ Scan and add RTL include files into project

☐ Copy sources into project

☑ Add sources from subdirectories

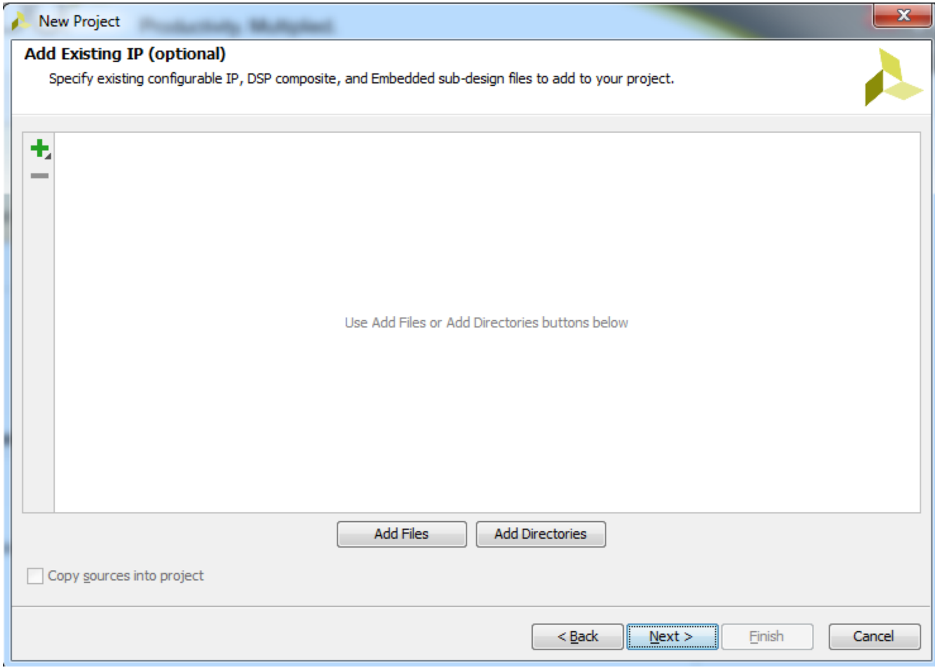
Target language: Verilog

Simulator language: Mixed

Fig. 4

4) Change the “Target language” to “VHDL”, and Click “Create File” => Input the file name, e.g. “MyAndGate” => “Next”.

5) In the “Add Existing IP (optional)”, click “Next”.



New Project

Add Existing IP (optional)

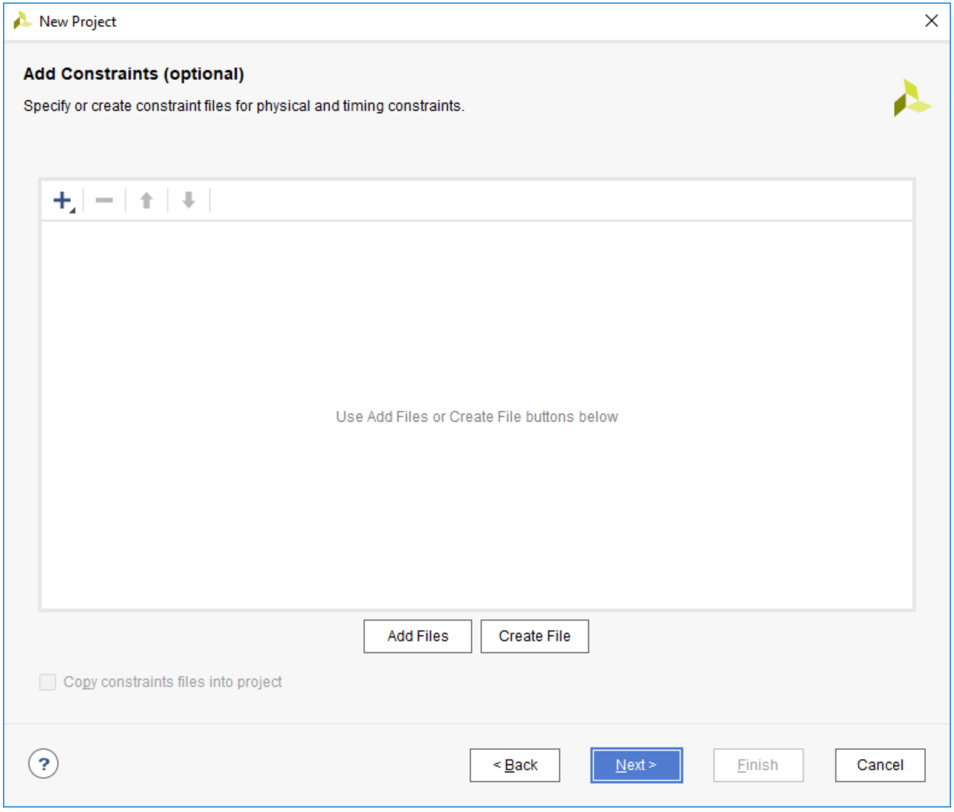
Specify existing configurable IP, DSP composite, and Embedded sub-design files to add to your project.

Use Add Files or Add Directories buttons below

☐ Copy sources into project

Fig. 5

6) In the “Add Constraint (Optional)”, click “Next”.



New Project

Add Constraints (optional)

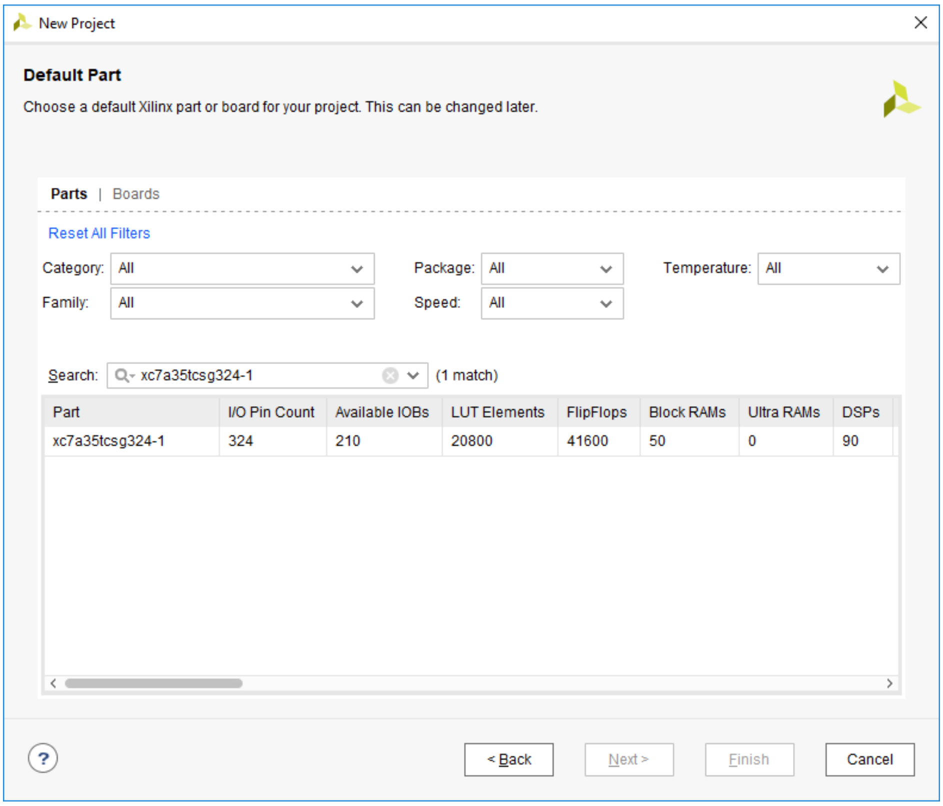
Specify or create constraint files for physical and timing constraints.

Use Add Files or Create File buttons below

□ Copy constraints files into project

Fig. 6

7) In the “Search” textbox, input “xc7a35tcsg324-1”, and then select the corresponding chip. Click “Next” => “Finish”.



New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Category: All

Family: All

Package: All

Speed: All

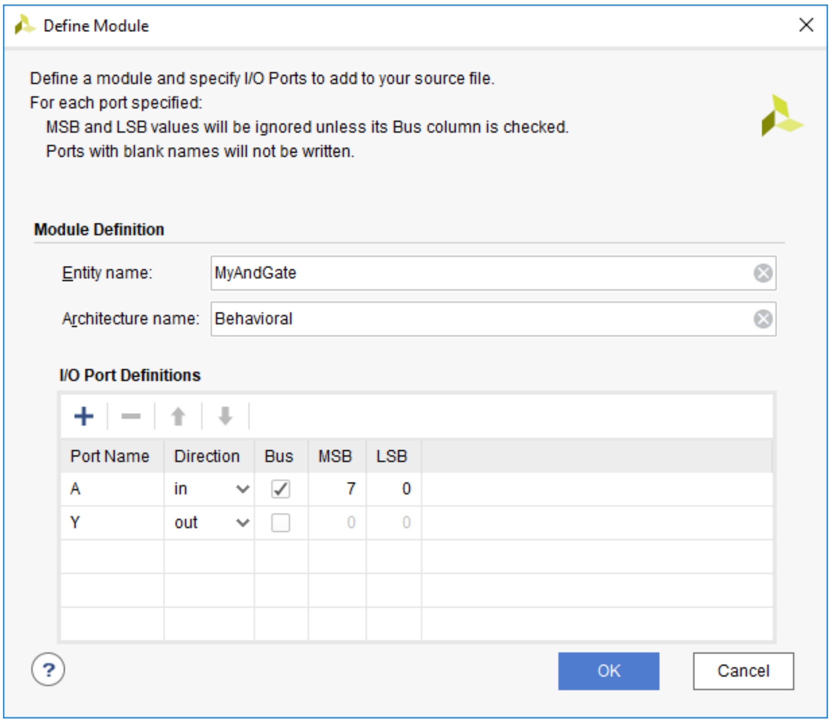
Temperature: All

Search: xc7a35tcsg324-1 (1 match)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7a35tcsg324-1	324	210	20800	41600	50	0	90

Fig. 7

8) Input the port names, directions, bus choice, MSB and LSB as below. Then click "OK".



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Entity name: MyAndGate

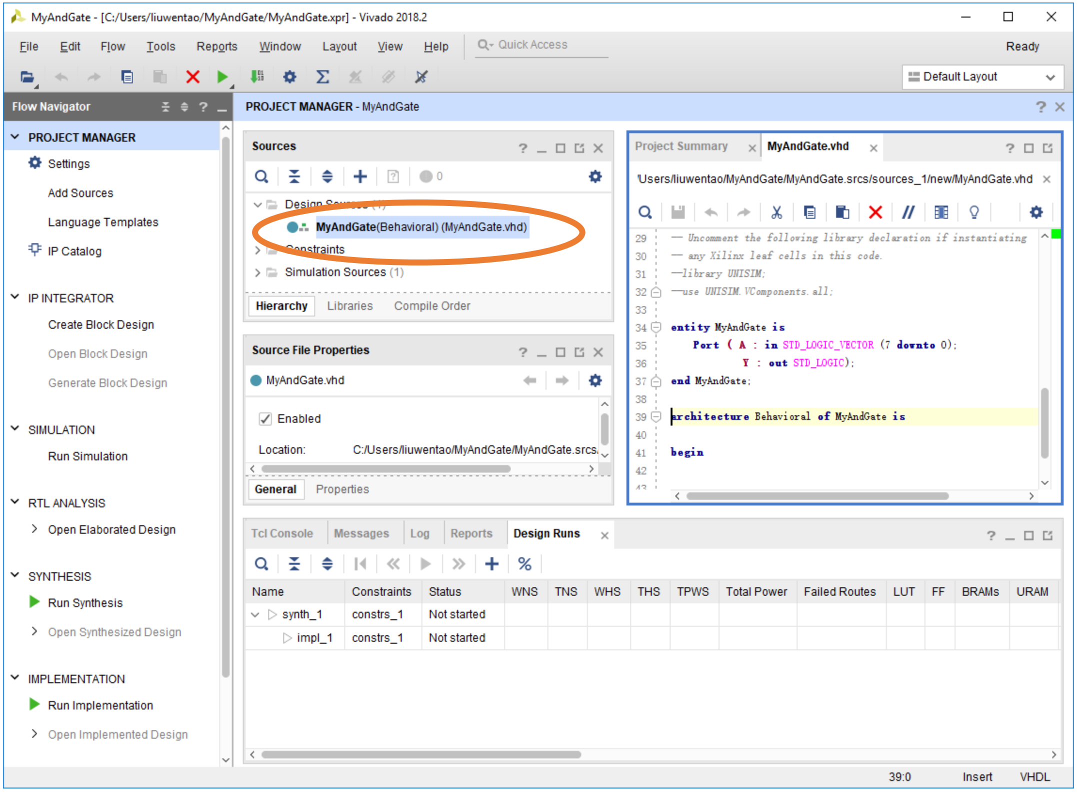
Architecture name: Behavioral

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
A	in	☑	7	0
Y	out	□	0	0

Fig. 8

9) Double click "MyAndGate" in the "Sources" panel, you will see the automatically generated coding in the window.



MyAndGate - [C:/Users/liuwentao/MyAndGate/MyAndGate.xpr] - Vivado 2018.2

☑ Enabled

Location: C:/Users/liuwentao/MyAndGate/MyAndGate.srcs

'Users/liuwentao/MyAndGate/MyAndGate.srcs/sources_1/new/MyAndGate.vhd

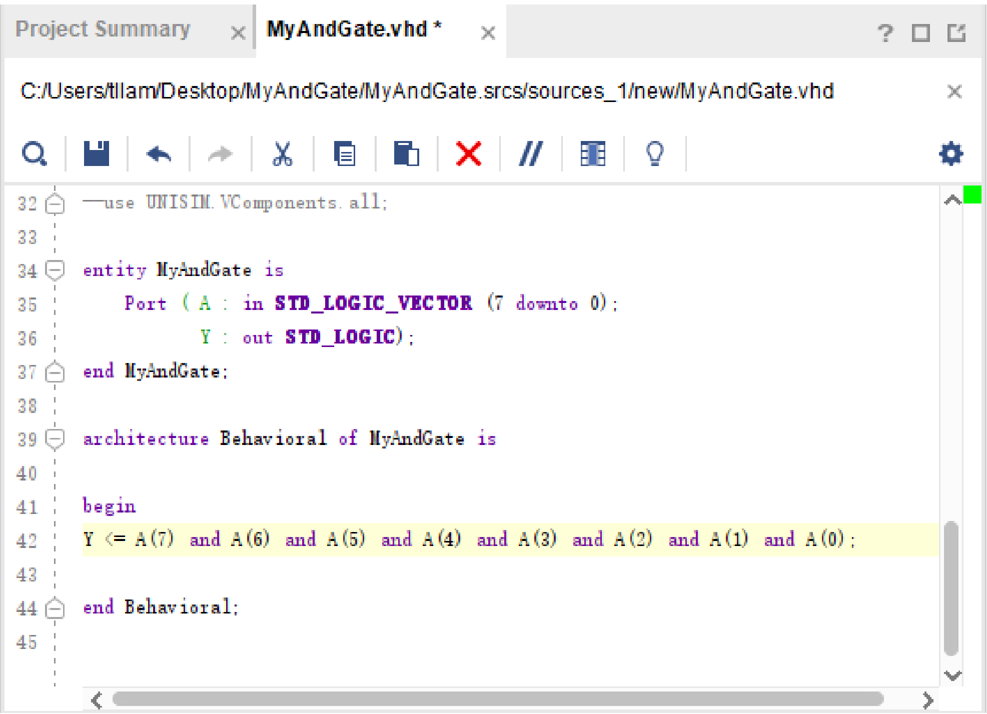
```
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity MyAndGate is
    Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
           Y : out STD_LOGIC);
end MyAndGate;

architecture Behavioral of MyAndGate is

begin
```

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM
synth_1	constrs_1	Not started											
impl_1	constrs_1	Not started											

39:0 Insert VHDL

Fig. 9

10) Add the coding describing the behavior as below. You can use “Ctrl+s” to save the coding.



C:/Users/tllam/Desktop/MyAndGate/MyAndGate.srcs/sources_1/new/MyAndGate.vhd

```
--use UNISIM.VComponents.all;

entity MyAndGate is
    Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
           Y : out STD_LOGIC);
end MyAndGate;

architecture Behavioral of MyAndGate is

begin
Y <= A(7) and A(6) and A(5) and A(4) and A(3) and A(2) and A(1) and A(0);

end Behavioral;

```

Fig. 10

11) Click the icon for “Run synthesis”, keep the default and Click OK.

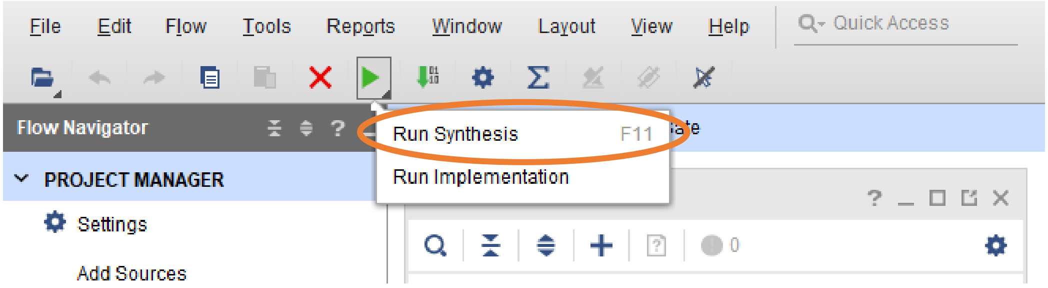
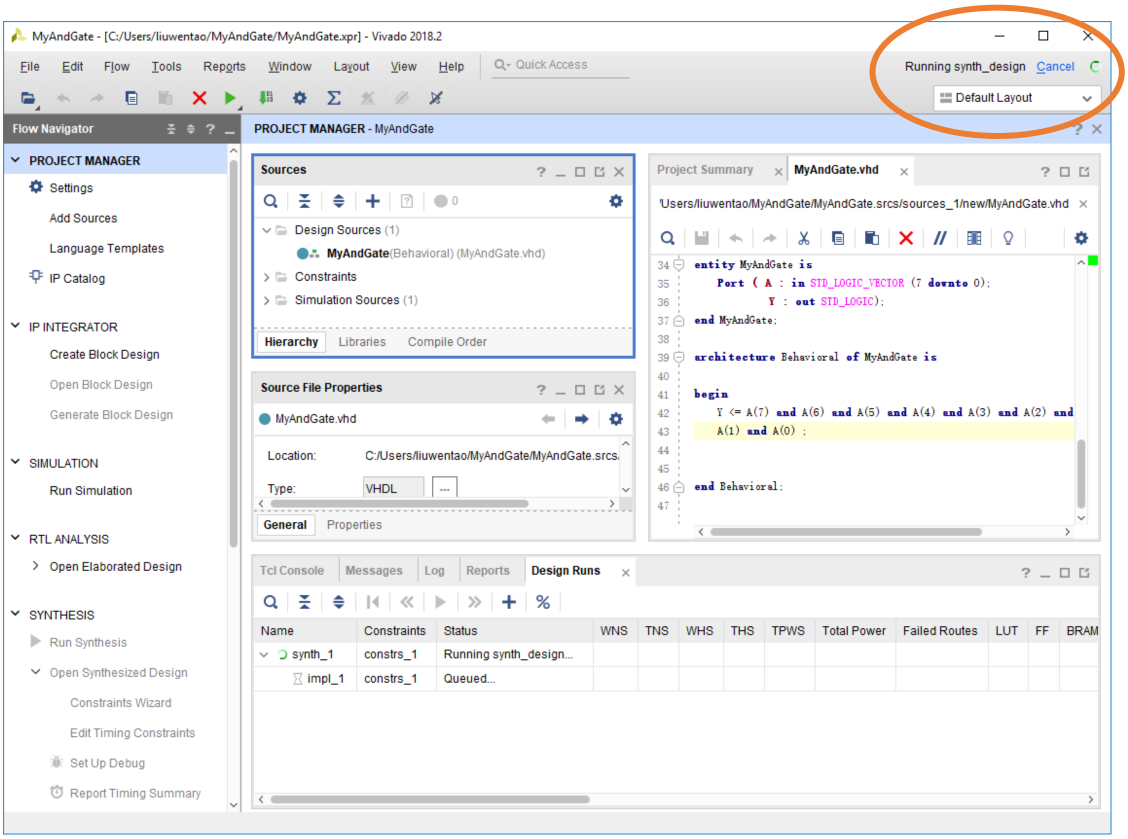


Fig. 11

There will be a progression bar on the right upper corner for a few seconds, indicating Vivado synthesis is running.

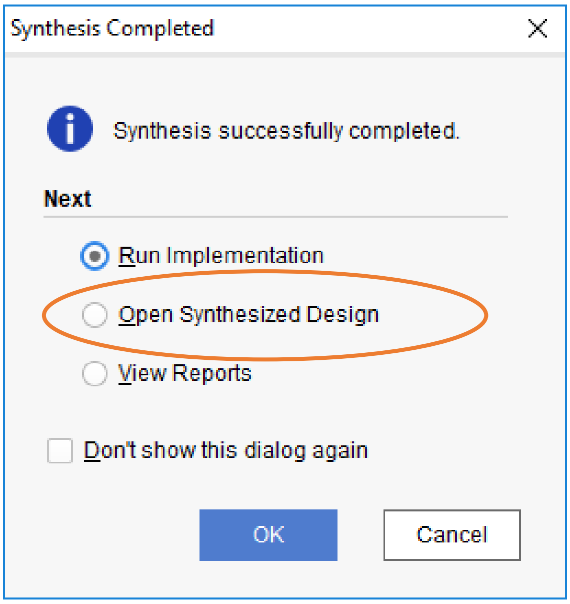


```
entity MyAndGate is
    Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
           Y : out STD_LOGIC);
end MyAndGate;

architecture Behavioral of MyAndGate is

begin
    Y <= A(7) and A(6) and A(5) and A(4) and A(3) and A(2) and
    A(1) and A(0) ;


end Behavioral;
```

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM
synth_1	constrs_1	Running synth_design...										
impl_1	constrs_1	Queued...										

Fig. 12

Then, the dialogue box below shows that the synthesis is completed. Choose "Open synthesized design" and click "OK".


Synthesis Completed

Synthesis successfully completed.

Next

● Run Implementation

○ Open Synthesized Design

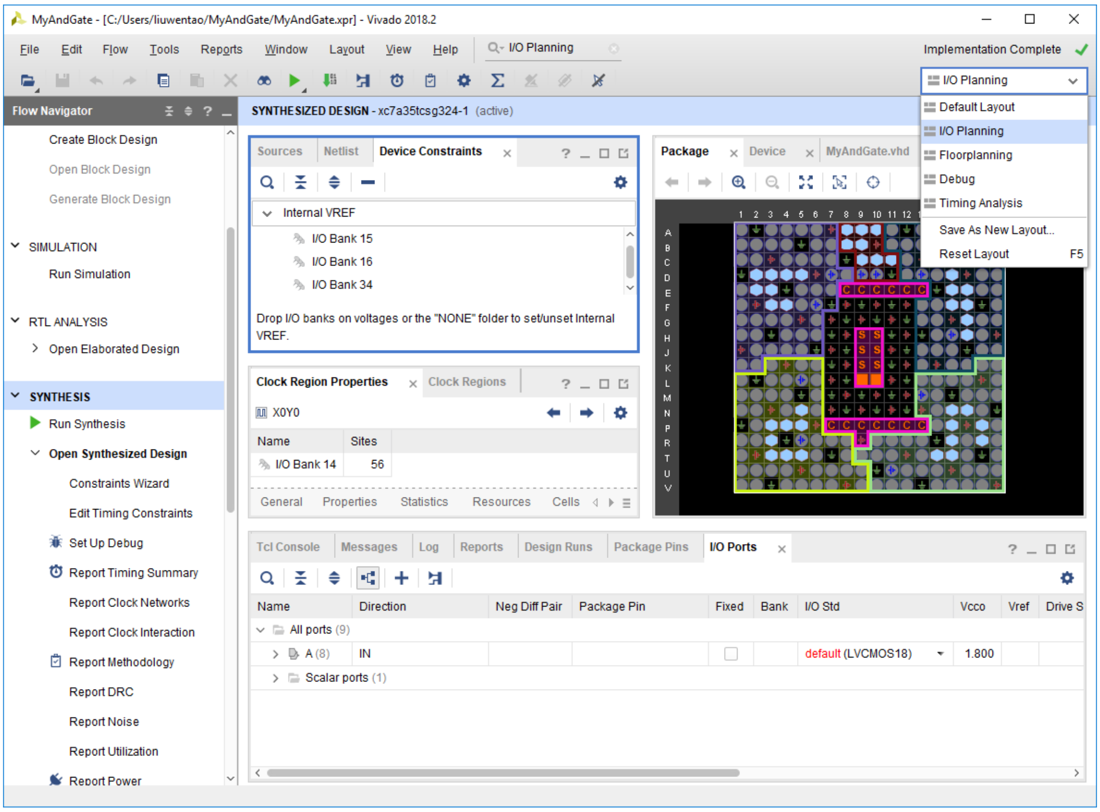
○ View Reports

□ Don't show this dialog again

OK Cancel

Fig. 13

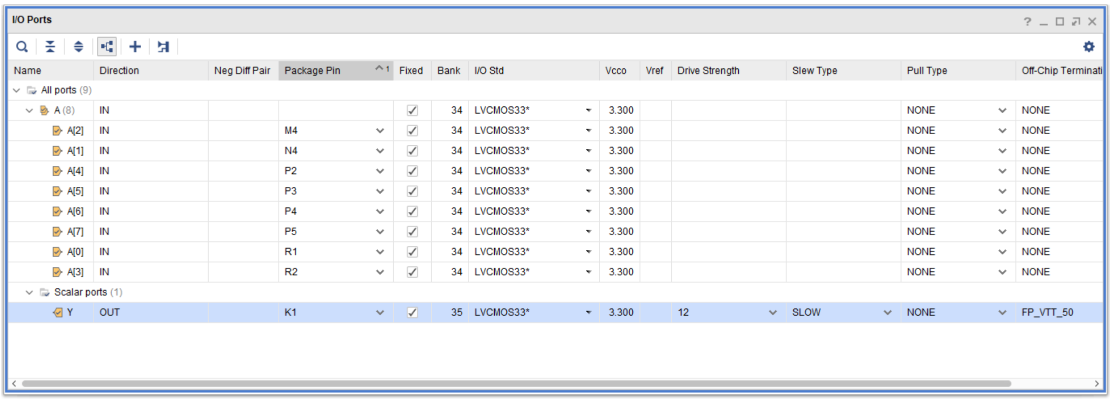
12) From the dropdown menu, select "I/O Planning". Now we are going to connect the input and output pins to the switches and LEDs.



Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive S
All ports (9)									
A (8)	IN			□		default (LVCMOS18)	1.800		
Scalar ports (1)									

Fig. 14

13) In the Synthesized Design, assign the "Site" of each input pins from A[7]-A[0] to P5, P4, P3, P2, R2, M4, N4, and R1, and the output pin to K1. Also set the "I/O Std" to "LVCMOS33*". This indicates the voltage level is CMOS 3.3V.



I/O Ports

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Terminati
All ports (9)												
A (8)	IN			☑	34	LVCMOS33*	3.300				NONE	NONE
A[2]	IN		M4	☑	34	LVCMOS33*	3.300				NONE	NONE
A[1]	IN		N4	☑	34	LVCMOS33*	3.300				NONE	NONE
A[4]	IN		P2	☑	34	LVCMOS33*	3.300				NONE	NONE
A[5]	IN		P3	☑	34	LVCMOS33*	3.300				NONE	NONE
A[6]	IN		P4	☑	34	LVCMOS33*	3.300				NONE	NONE
A[7]	IN		P5	☑	34	LVCMOS33*	3.300				NONE	NONE
A[0]	IN		R1	☑	34	LVCMOS33*	3.300				NONE	NONE
A[3]	IN		R2	☑	34	LVCMOS33*	3.300				NONE	NONE
Scalar ports (1)												
Y	OUT		K1	☑	35	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50

Fig. 15

P5, P4, P3, P2, R2, M4, N4, R1, and K1 are connected to the I/O of the FPGA chip. You can also check the pins' names by the text on the development board. For other PINs, you can take reference to "EGO1 User Manual".

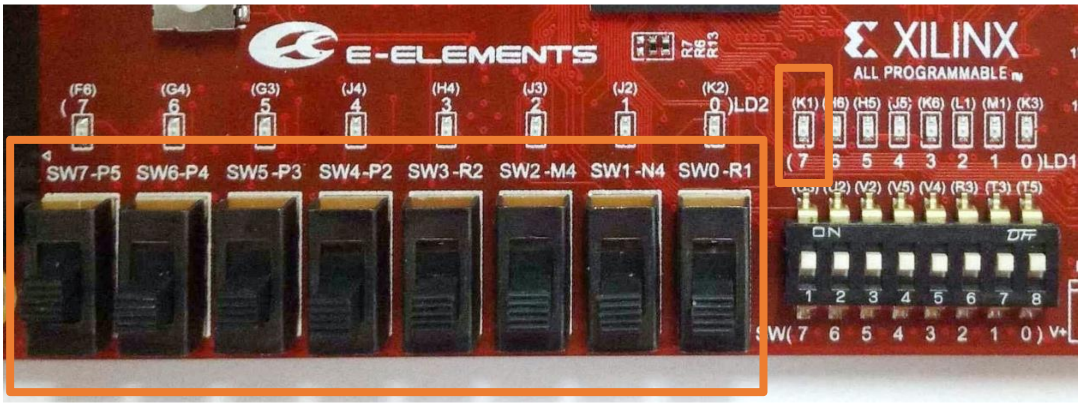
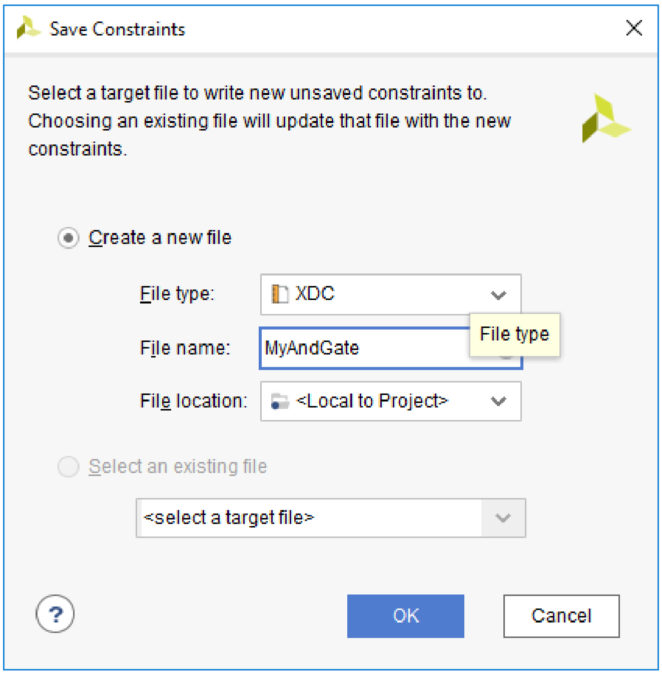


Fig. 16

14) Use "Ctrl+s" to save the constraint file. In the dialogue box, input the "File name", e.g. "MyAndGate", and click "OK".


Save Constraints

Select a target file to write new unsaved constraints to. Choosing an existing file will update that file with the new constraints.

● Create a new file

File type: XDC

File name: MyAndGate

File location: <Local to Project>

○ Select an existing file

<select a target file>

Fig. 17

15) Click the icon for “Run implementation”. It will ask you to run synthesis again. Click “OK”.

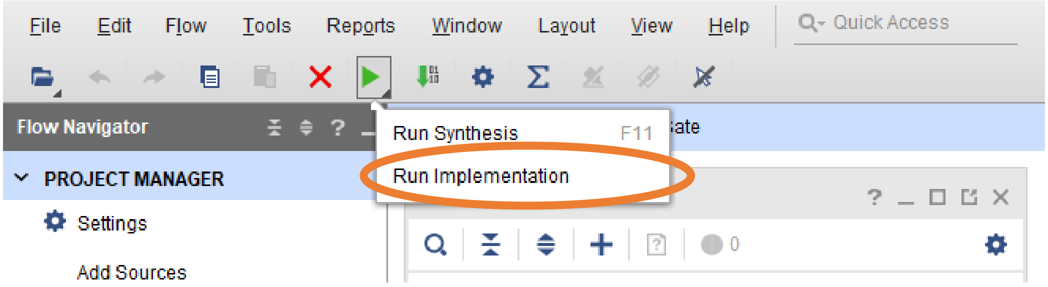
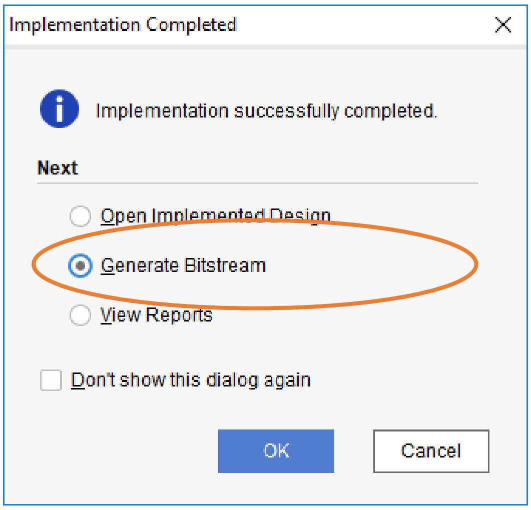


Fig. 18

16) This synthesis and implementation step might take a longer time, about 10s of seconds varying based on computers. When the following dialogue box prompts out, choose “Generate Bitstream” and click “OK”.


Implementation Completed

Implementation successfully completed.

Next

○ Open Implemented Design

● Generate Bitstream

○ View Reports

□ Don't show this dialog again

Fig. 19

It will take a few seconds, and the progress bar will be shown in the right upper corner.

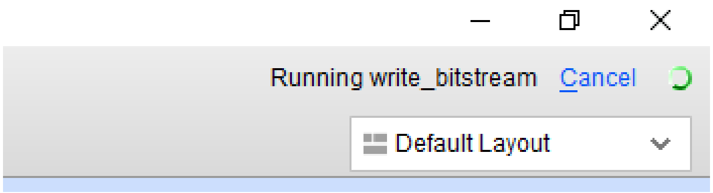
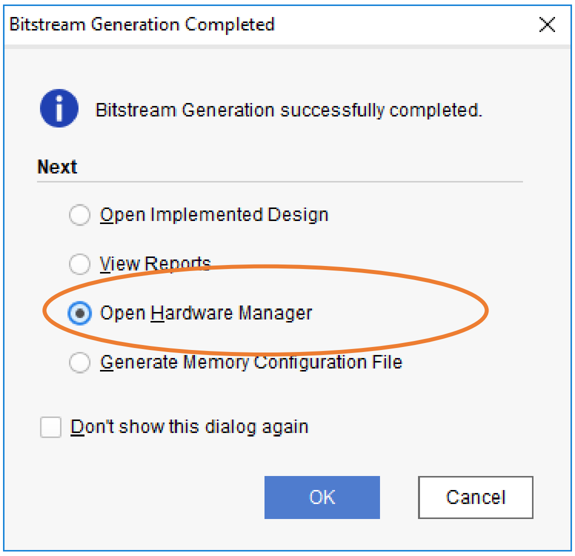

Running write_bitstream

Fig. 20

17) When it is completed, select “Open Hardware Manager” and click “OK”.



Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

○ Open Implemented Design

○ View Reports

● Open Hardware Manager

○ Generate Memory Configuration File

□ Don't show this dialog again

Fig. 21

18) Connect the J6 port of the development board with a computer through USB, and then turn on the switch. You will see the 7-Segments are all 0, with other 4 LEDs on the upper part of the board on.

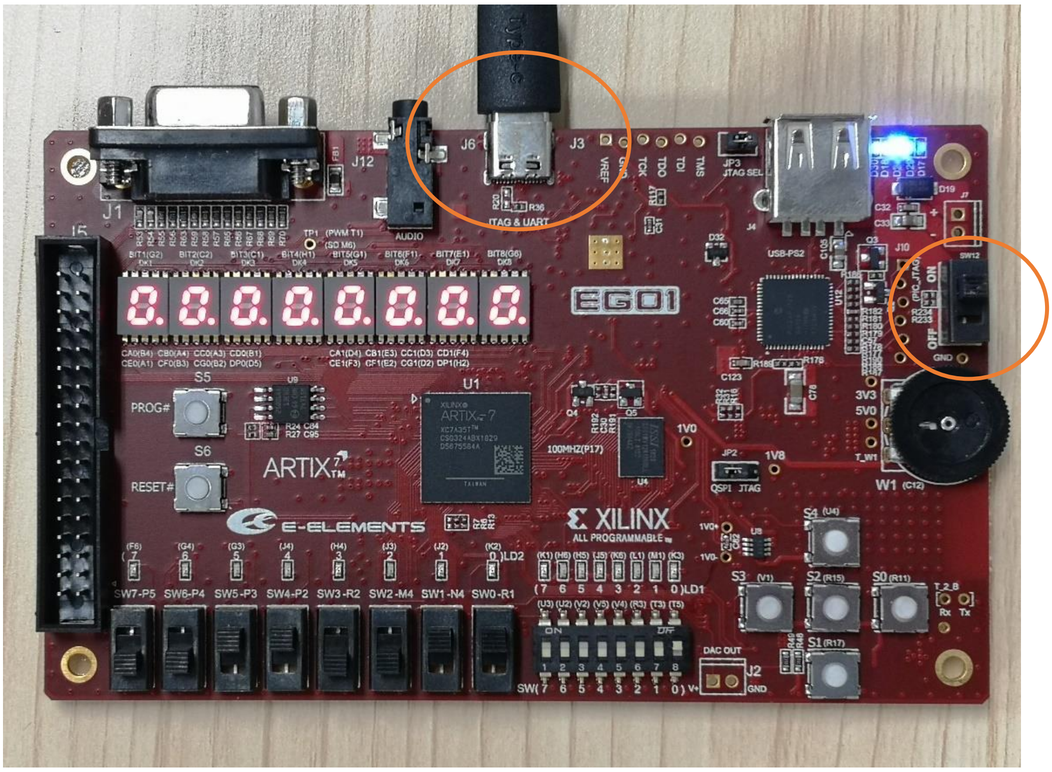
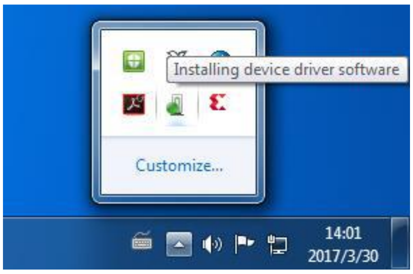


Fig. 22

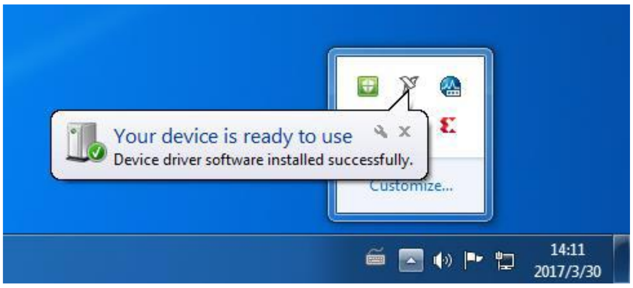
The driver needs to be installed.



Installing device driver software

Fig. 23

It will take from a few seconds to a few minutes to install the driver. When you see the indicating text below, the drivers are installed successfully.

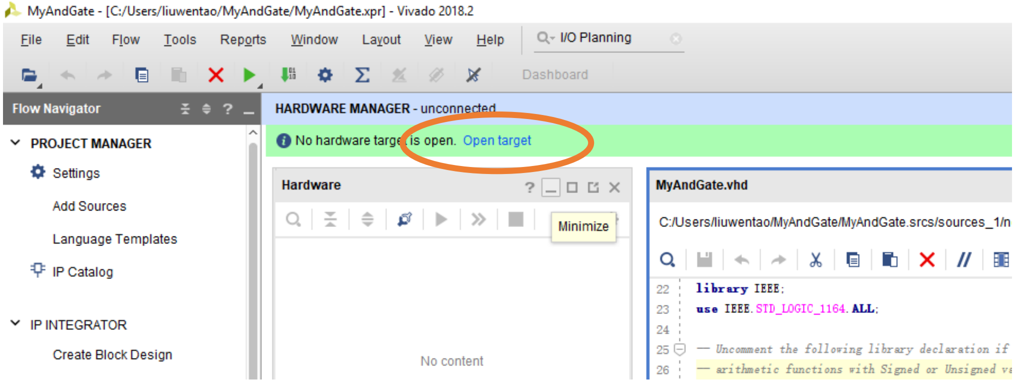


Your device is ready to use

Device driver software installed successfully.

Fig. 24

19) Click the “Open target” in the “Hardware Manager”, and then click “Auto Connect”.



MyAndGate - [C:/Users/liuwentao/MyAndGate/MyAndGate.xpr] - Vivado 2018.2

HARDWARE MANAGER - unconnected

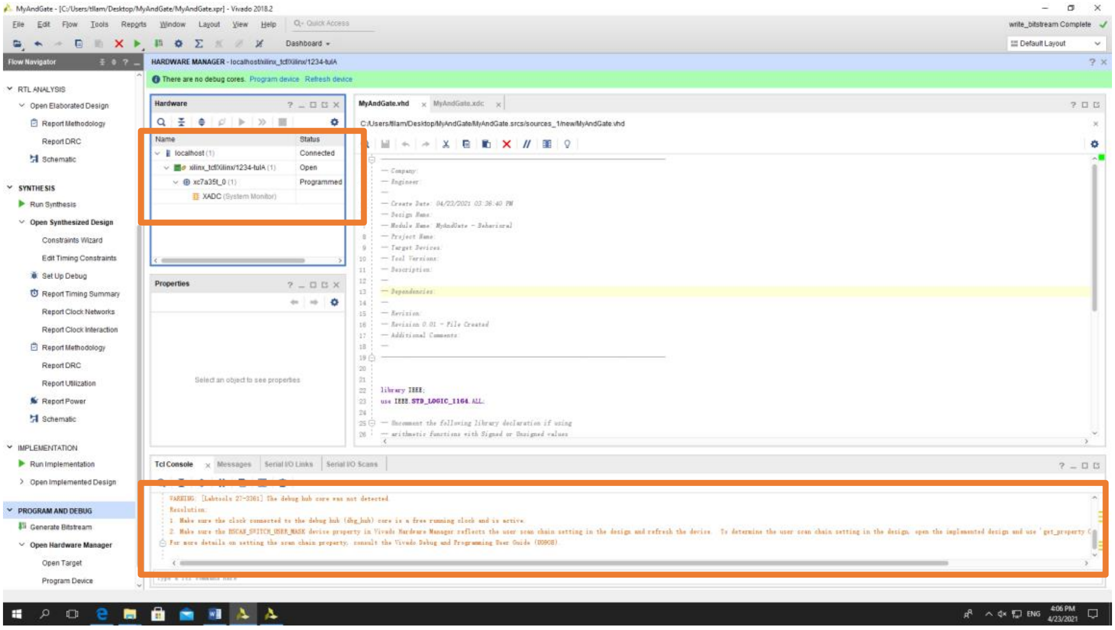
No hardware target is open. Open target

C:/Users/liuwentao/MyAndGate/MyAndGate.srcs/sources_1/n

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if
-- arithmetic functions with Signed or Unsigned v
```

Fig. 25

When you see the interface below, it indicates that the hardware connection is successful.



HARDWARE MANAGER - localhost/xilinx_tcf/Xilinx/1234-tulA

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Xilinx/1234-tulA (1)	Open
xc7a35t_0 (1)	Programmed
XADC (System Monitor)	

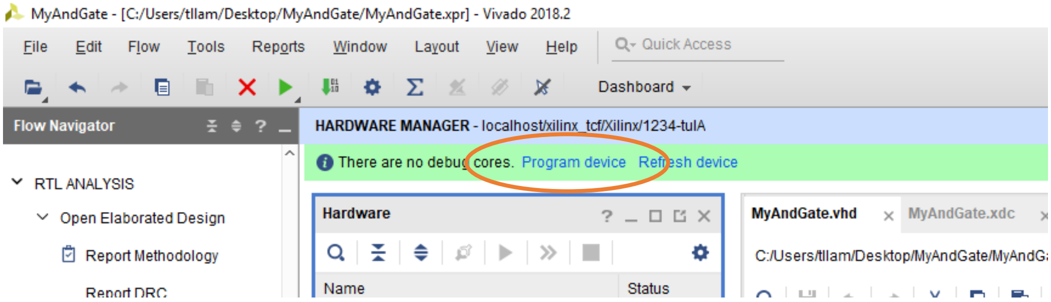
C:/Users/tllam/Desktop/MyAndGate/MyAndGate.srcs/sources_1/new/MyAndGate.vhd

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/23/2021 03:36:40 PM
-- Design Name:
-- Module Name: MyAndGate - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
```

```
WARNING: [Labtools 27-3361] The debug hub core was not detected.
Resolution:
1. Make sure the clock connected to the debug hub (dbg_hub) core is a free running clock and is active.
2. Make sure the BSCAN_SWITCH_USER_MASK device property in Vivado Hardware Manager reflects the user scan chain setting in the design and refresh the device.  To determine the user scan chain setting in the design, open the implemented design and use 'get_property C
For more details on setting the scan chain property, consult the Vivado Debug and Programming User Guide (UG908).
```

Fig. 26

20) Click “Program device” to download coding the FPGA.



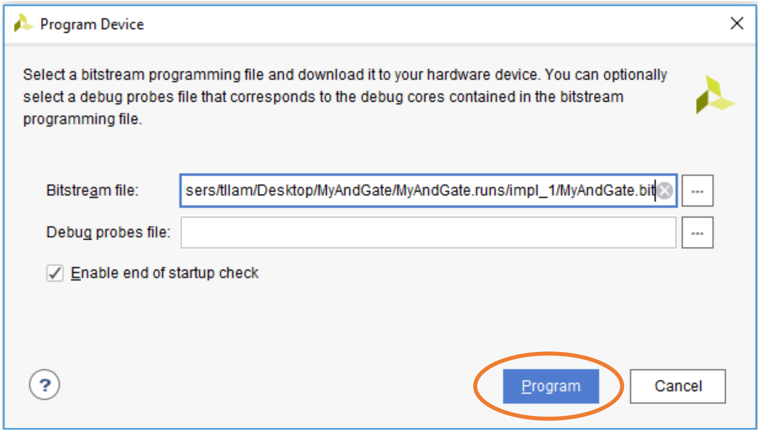
MyAndGate - [C:/Users/tllam/Desktop/MyAndGate/MyAndGate.xpr] - Vivado 2018.2

HARDWARE MANAGER - localhost/xilinx_tcf/Xilinx/1234-tulA

There are no debug cores. Program device Refresh device

Fig. 27

Click “Program”.



Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: sers/tllam/Desktop/MyAndGate/MyAndGate.runs/impl_1/MyAndGate.bit

Debug probes file:

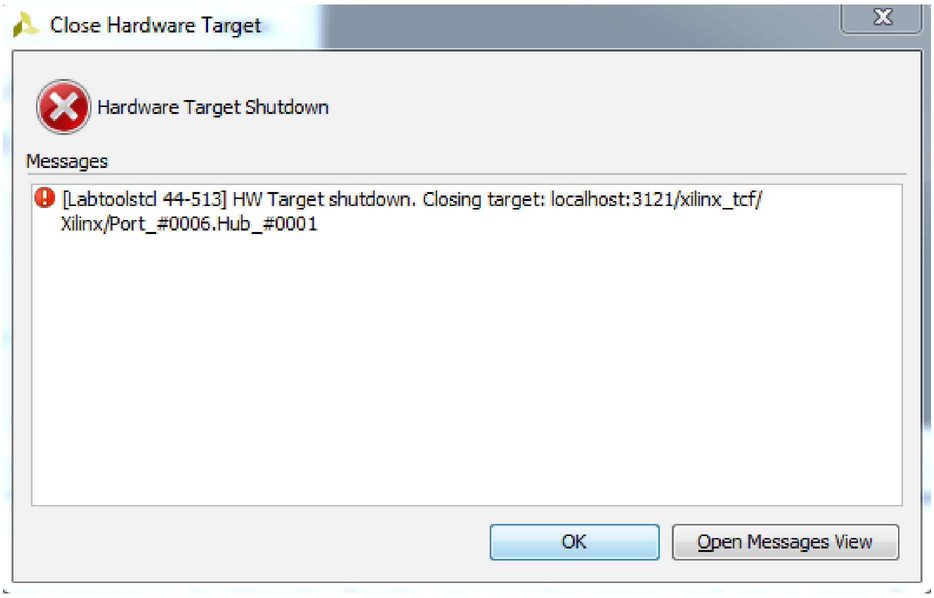
☑ Enable end of startup check

Program | Cancel

Fig. 28

21) You will be able to test the AND gate by the 8 switches now.

22) When you would like to unplug the USB or turn off the development board, always remember to close "Hardware Manager" first. Otherwise, you will meet the error below.



[Labtoolstcl 44-513] HW Target shutdown. Closing target: localhost:3121/xilinx_tcf/Xilinx/Port_#0006.Hub_#0001

Fig. 29

Follow these steps to recover from the error: close the Hardware Manager => turn off the power switch => disconnect the usb cable => reconnect the usb cable => turn on the power switch => wait for driver installation (about 10 seconds) => run Vivado and get back to the Hardware Manager => click "Open target" and "Auto Connect".

23) Now you can test the logic functions of an 8-input AND gate.

24) Redesign a new project for a 2-input-1-output XOR. Use "MyXORGate" as the file name.

[DEMONSTRATION] Show instructor or TA when you have completed the 8-input AND gate and step 2-input-1-output XOR gate.

[IN REPORT] Include your coding and all test results.

[QUESTION] Write the entity and architecture for the combinational logic of a 4-bits plus 4-bits adder, with no input carry and 1-bit output carry.

3.2 **Experiment 2:** Count-down timer

You are required to design a count-down timer based on FPGA. The functional requirements are:

A. The 4 LEDs from LD1(4-7) are flashing at the frequency of 8Hz, 4Hz, 2Hz, and 1Hz.

B. Set the BCD codes of 2 digits (from decimals 99 to 0) by the switches SW0-SW7. SW0 and SW3 are the LSB and MSB of the 1s digit, respectively. SW4 and SW7 are the LSB and MSB of the 10s digit, respectively. Any BCD code exceeding 0b1001 should be regarded as illegal, and then will be regarded as 0b1001. Press the button S4 on the board, the initialized count-down time will be shown on the left two 7-segment displays. If the 10s digit is 0, the left 7-segment display will show nothing.

C. When turn on the button SW(0) on the board, the timer starts immediately, and the 2-digit display counts down at the frequency of 1Hz. At the end of the process, the 1s digit stops and keeps “0”.

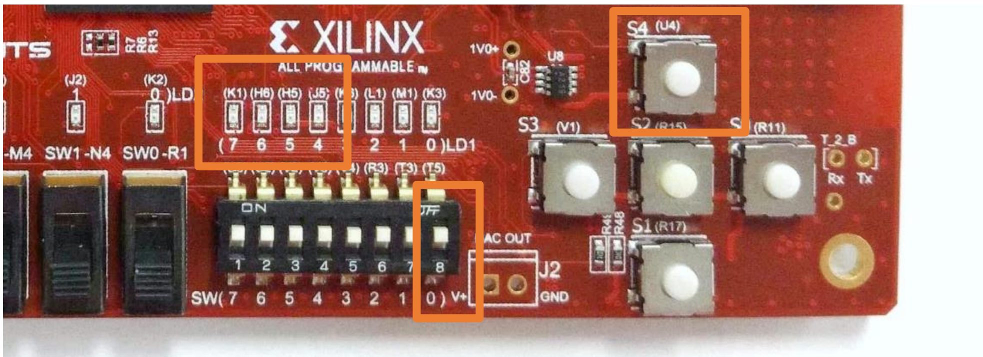


Fig. 30 Location of LD1(4-7), S4, and SW(0)

To realize this timer, there are a number of steps and questions to think about. We provide some hint, and you are expected to design most of the coding by yourselves.

1) How to generate a counter at the frequency of 1Hz (i.e. make LED LD1(7) flashing at 1Hz)? You are advised to realize 8Hz on LED LD1(4) first, and then divide the frequency to 4Hz, 2Hz, and 1Hz.

There is a 100MHz clock in the development board. The PIN is P17. Try to use it as the input. In the port, define an input, e.g. ClkFPGA. Connect it to P17. Check the rising edge by:

```
if rising_edge(ClkFPGA) then
```

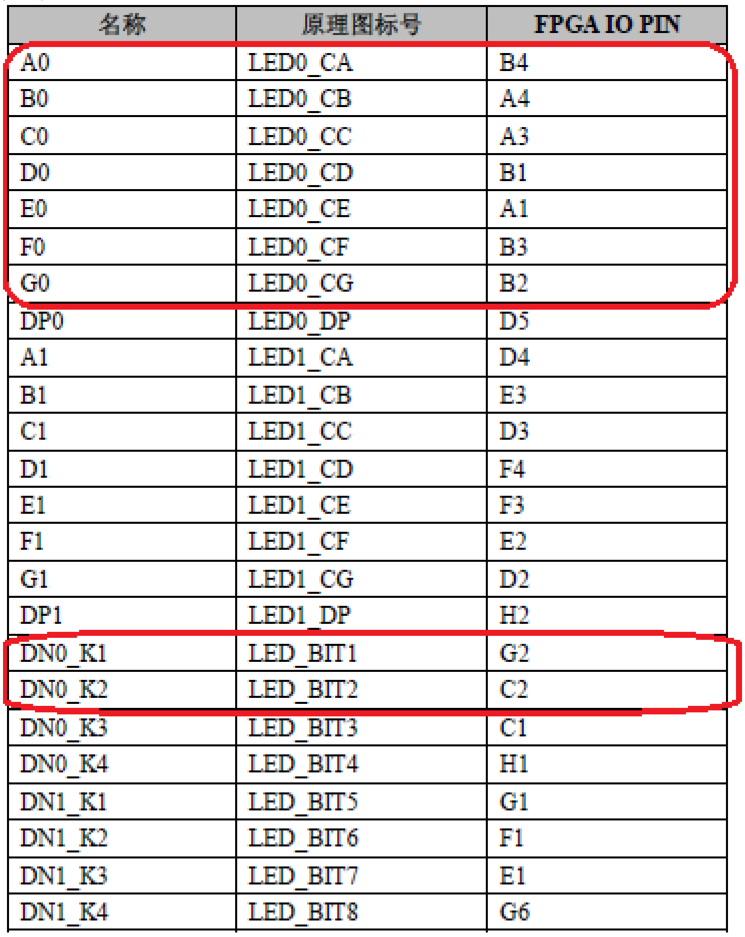
2) How to input SW0~7 into the initial BCD codes? How to change BCD codes exceeding 0b1001 (e.g. 0b1010) to 0b1001? You can use the button S4 to indicate the reading command of the SW0~7. When S4 is pressed, the BCD codes will be shown to the 8 LEDS, i.e. LD2(0-7).

3) How to show the 2-digits simultaneously in the first two 7-segment displays? Try to show the two BCD numbers onto the 1st and 2nd 7-segment displays from the left simultaneously.

The 7-segment display in the development board is different from the hardware we use in the previous labs. They have the same cathodes (negative terminals), and thus high voltage should be supplied to activate it. The first four 7-segment displays share the same A~G pins, while the other 4 share the other same A-G pins. For each of the 7-segment, there is a selection pin, DN0_K1~K4 for the 7-segments in the left, and

DN1_K1~K4 for those in the right. A high voltage should be provided to one selection pin to activate it. For example, if you would like to put a “7” onto the 2nd display from the left, set DN0_K2 = “1” and A~G = “1110000”.

To show both the two digits onto two 7-segment, it is a bit tricky. You can iteratively show different values of A~G numbers onto the 1st and 2nd 7-segments, and use DN0_K1=“1” or DN0_K0=“1” to switch from them at a frequency that your eyes can not identify.



名称	原理图标号	FPGA IO PIN
A0	LED0_CA	B4
B0	LED0_CB	A4
C0	LED0_CC	A3
D0	LED0_CD	B1
E0	LED0_CE	A1
F0	LED0_CF	B3
G0	LED0_CG	B2
DP0	LED0_DP	D5
A1	LED1_CA	D4
B1	LED1_CB	E3
C1	LED1_CC	D3
D1	LED1_CD	F4
E1	LED1_CE	F3
F1	LED1_CF	E2
G1	LED1_CG	D2
DP1	LED1_DP	H2
DN0_K1	LED_BIT1	G2
DN0_K2	LED_BIT2	C2
DN0_K3	LED_BIT3	C1
DN0_K4	LED_BIT4	H1
DN1_K1	LED_BIT5	G1
DN1_K2	LED_BIT6	F1
DN1_K3	LED_BIT7	E1
DN1_K4	LED_BIT8	G6

Fig. 31 PIN table for 7-segment displays in EGO1

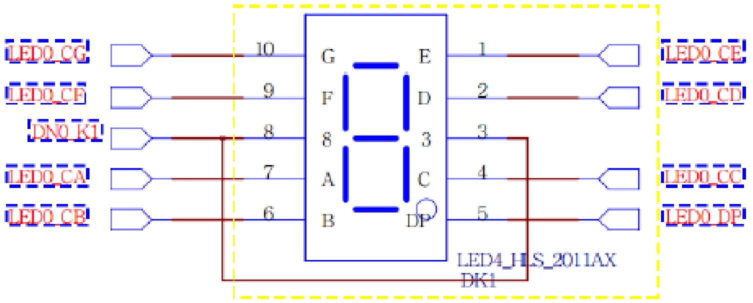


Fig. 32 PINs of one 7-segment display

4) How to count down and stop at "0"? Try to design the coding by yourself.

[DEMONSTRATION] Show instructor or TA when you have completed each part of A, B and C.

[IN REPORT] When you have completed everything from A to C, do it completely again. As you have already designed the full source code successfully, you can just copy and paste into the new source code file. In the report, you shall include all the steps with images, such as in Experiment 1 (Figures 2~28), as well as the coding and test results (with images as well). This provide training for you to write the manual for your designed system in your future research.

[QUESTION] Design the code of a count-down timer for 2 hours (e.g. for an examination) with the system clock of 100MHz. When the start-button is pressed, the timer starts. When the 2-hour period is completed, counter stops and an LED is turned on. Include the VHDL source code in the report, and you do not need to include the I/O planning.

4. Lab Report

Submit the report of Lab 6 in **PDF** to the folder **Digital Systems Design Lab/Report Submission/Lab 6** on Blackboard by the deadline below:

- **23:59, May 17, 2024**

Each day of late submission will result in 10% deduction in the report raw marks.

Remember to sort and put the FPGA development board in your cabinet after lab.