

Modul 1

Arsitektur Zynq

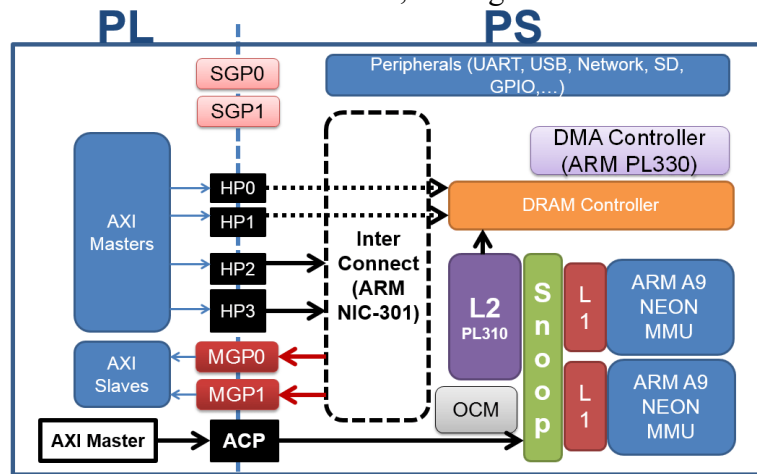
1. Tujuan

- Memahami arsitektur Zynq.
- Memahami software Vivado 2016.4 dan membuat project baru.

2. Materi

a. Arsitektur Zynq

Zynq merupakan FPGA SoC. Block diagram arsitektur Zynq ditunjukkan pada gambar 1. Pada Zynq terdapat Processing System (PS) dan Programmable Logic (PL). Pada PS terdapat processor dual core ARM Cortex-A9, sedangkan PL sendiri merupakan FPGA.



Gambar 1. Arsitektur Zynq

Pada PL, kita dapat membuat design menggunakan Verilog ataupun menggunakan IP core yang telah disediakan oleh Xilinx. Design pada PL dapat berkomunikasi dengan PS melalui beberapa interface seperti SGP, MGP, HP, dan ACP. Design yang dibuat pada PL harus compatible dengan standard AXI bus agar dapat berkomunikasi dengan PS.

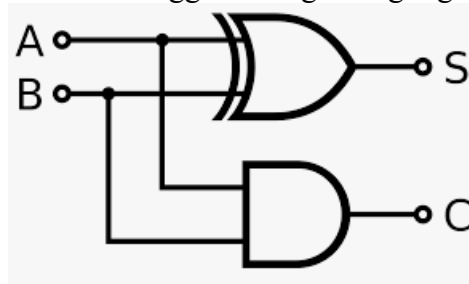
b. Half Adder

Merupakan elemen penjumlahan yang paling kecil yang memiliki tabel kebenaran pada tabel 1.

input		output	
a	b	sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabel 1. Tabel kebenaran half adder

Rangkaian half adder dapat dibuat menggunakan gerbang logika seperti pada gambar 2.



Gambar 2. Rangkaian half adder

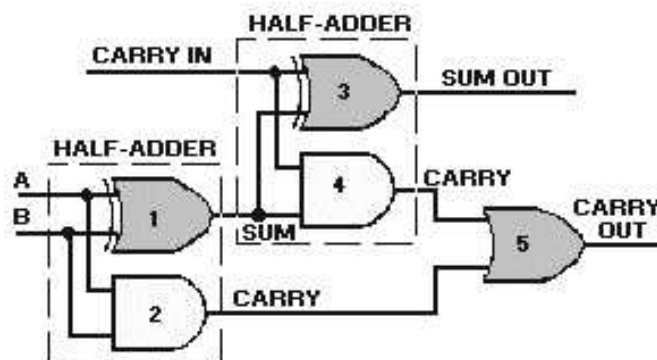
c. Full Adder

Merupakan elemen penjumlahan yang dibuat dari half adder yang memiliki tabel kebenaran seperti pada tabel 2.

input			output	
a	b	c	sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabel 2. Tabel kebenaran full adder

Rangkaian full adder dapat dibuat dari komponen half adder dan sebuah gerbang OR seperti pada gambar 3.



Gambar 3. Rangkaian full adder

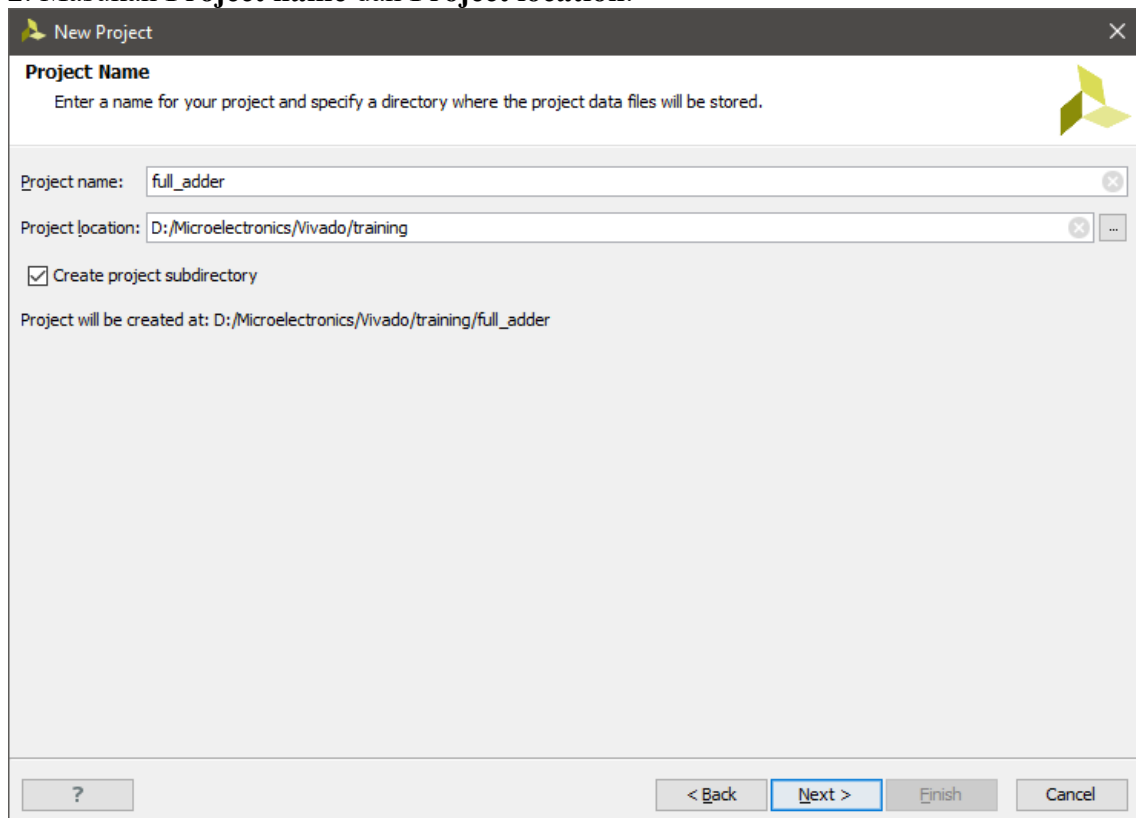
3. Tutorial

Pada tutorial ini, kita akan membuat dan mensimulasikan full adder pada Vivado, kemudian men-download file bitstream ke PL. Pada tutorial ini kita hanya akan memakai PL, sedangkan untuk PS akan digunakan pada tutorial berikutnya.

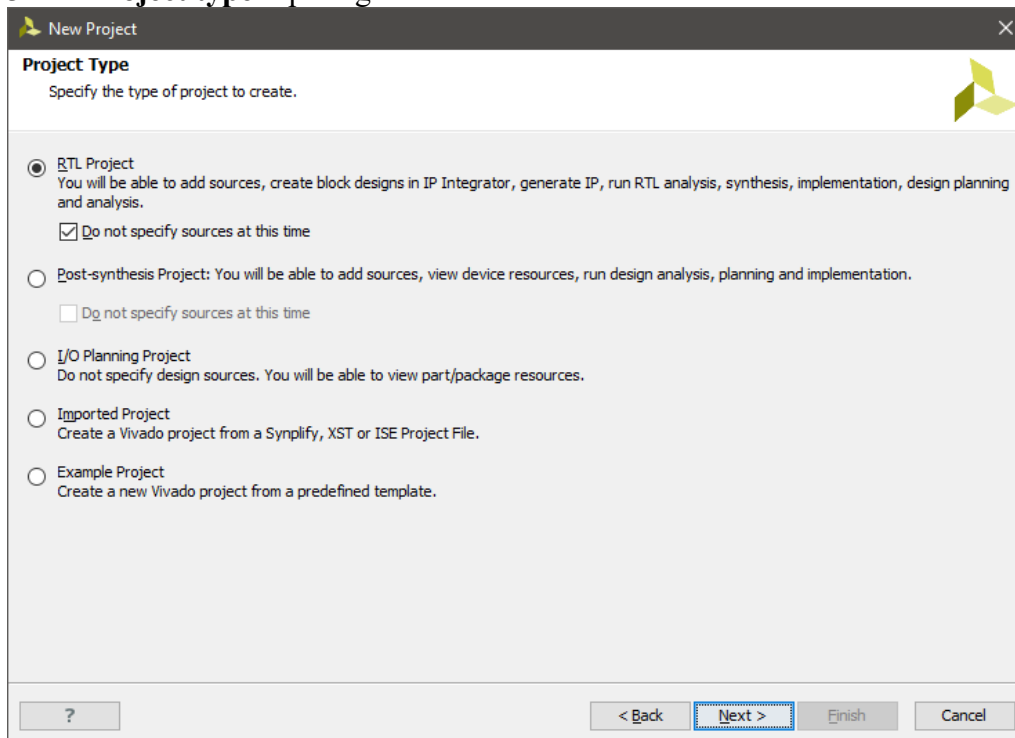
a. Membuat project baru

1. Masuk ke menu **File** → **New Project** → **Next**.

2. Masukkan **Project name** dan **Project location**.



3. Set **Project type** seperti gambar berikut.



New Project

Project Type
Specify the type of project to create.

☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
☒ Do not specify sources at this time

☐ **Post-synthesis Project:** You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time

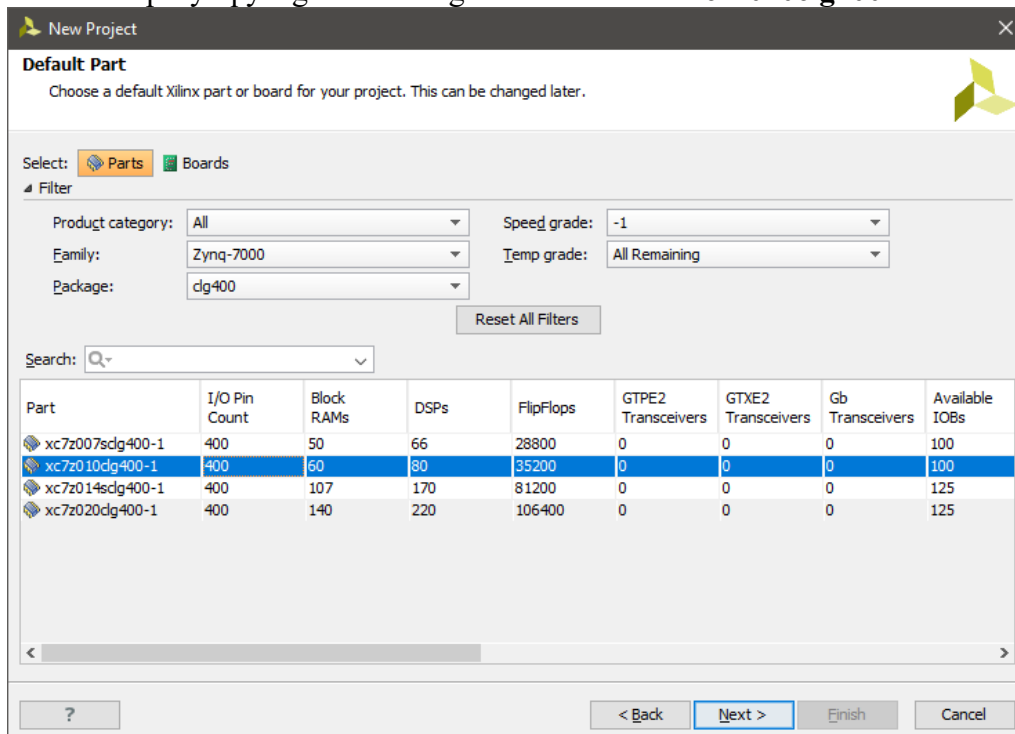
☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**
Create a Vivado project from a Synplify, XST or ISE Project File.

☐ **Example Project**
Create a new Vivado project from a predefined template.

? < Back Next > Finish Cancel

4. Pilih chip Zynq yang sesuai dengan board ZYBO: **xc7z010clg400-1**



New Project

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

Select: **Parts** Boards

Filter

Product category: All Speed grade: -1
Family: Zynq-7000 Temp grade: All Remaining
Package: clg400

Reset All Filters

Search: Q

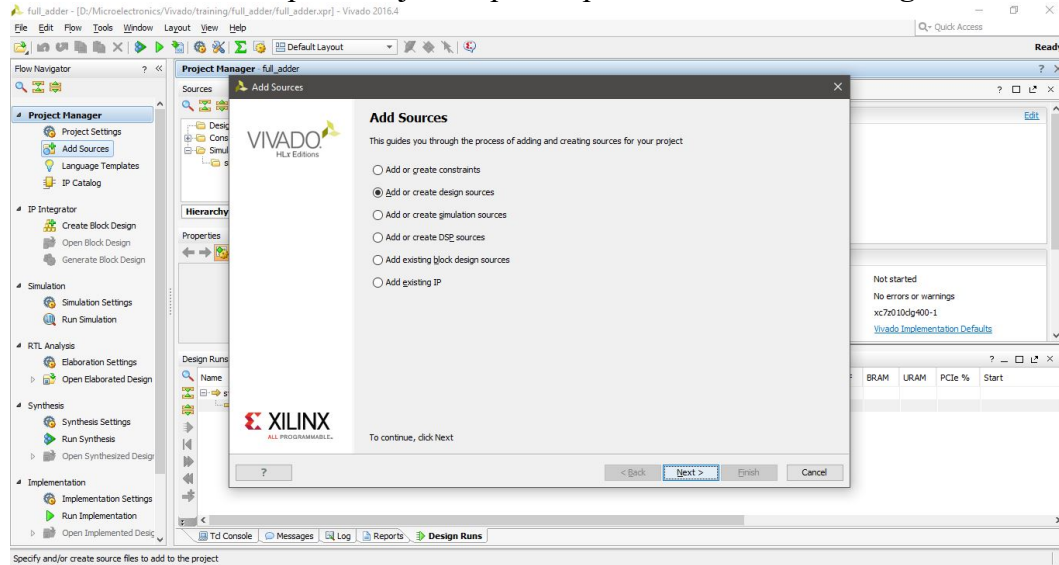
Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	GTXE2 Transceivers	Gb Transceivers	Available IOBs
xc7z007sclg400-1	400	50	66	28800	0	0	0	100
xc7z010clg400-1	400	60	80	35200	0	0	0	100
xc7z014sclg400-1	400	107	170	81200	0	0	0	125
xc7z020dclg400-1	400	140	220	106400	0	0	0	125

? < Back Next > Finish Cancel

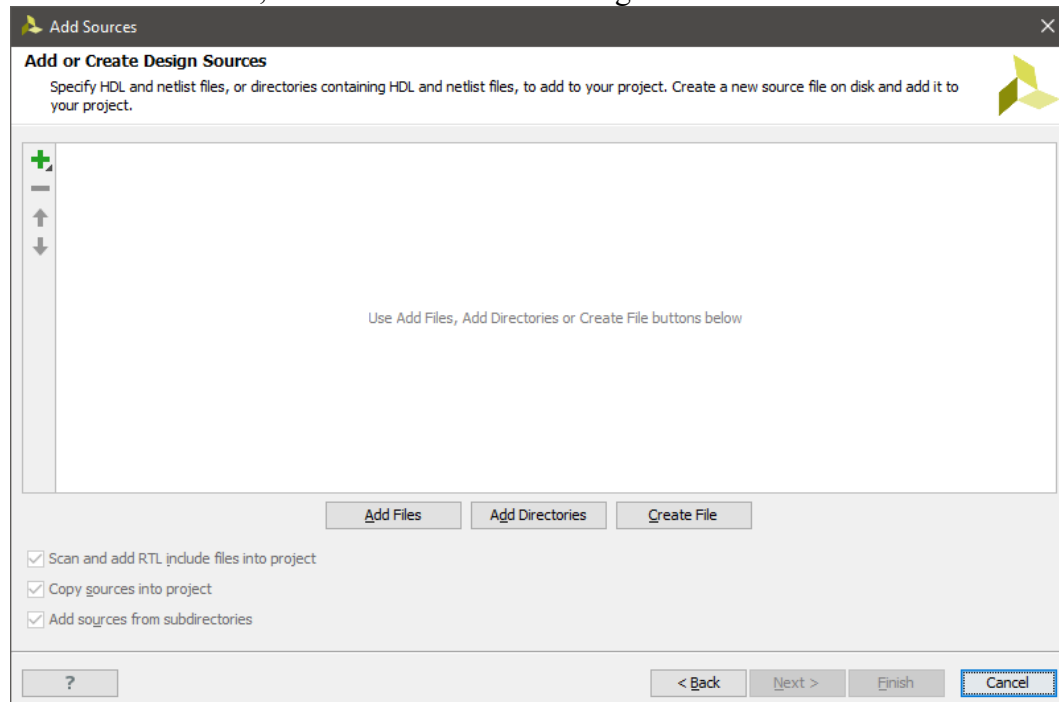
5. Klik **Finish** untuk membuat project baru.

b. Membuat source code Verilog

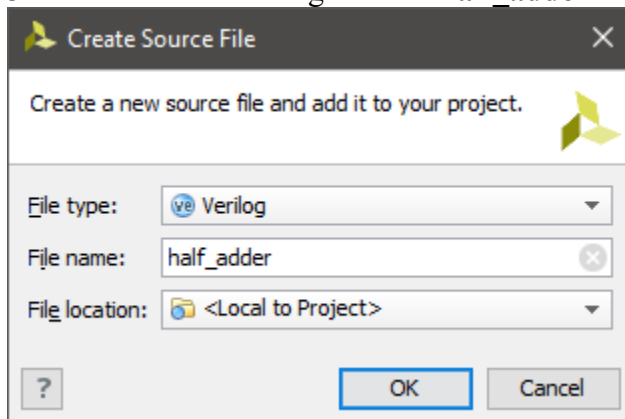
1. Klik Add Sources pada Project Explorer, pilih Add or create design sources.



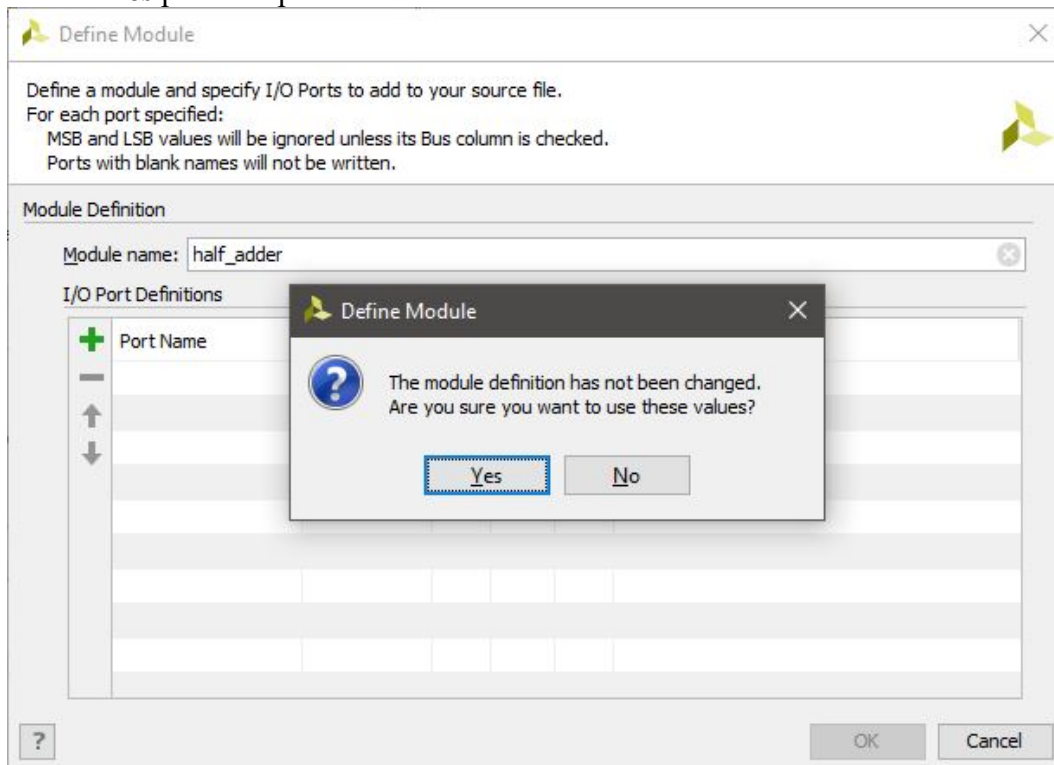
2. Klik Create File, untuk membuat file Verilog baru.



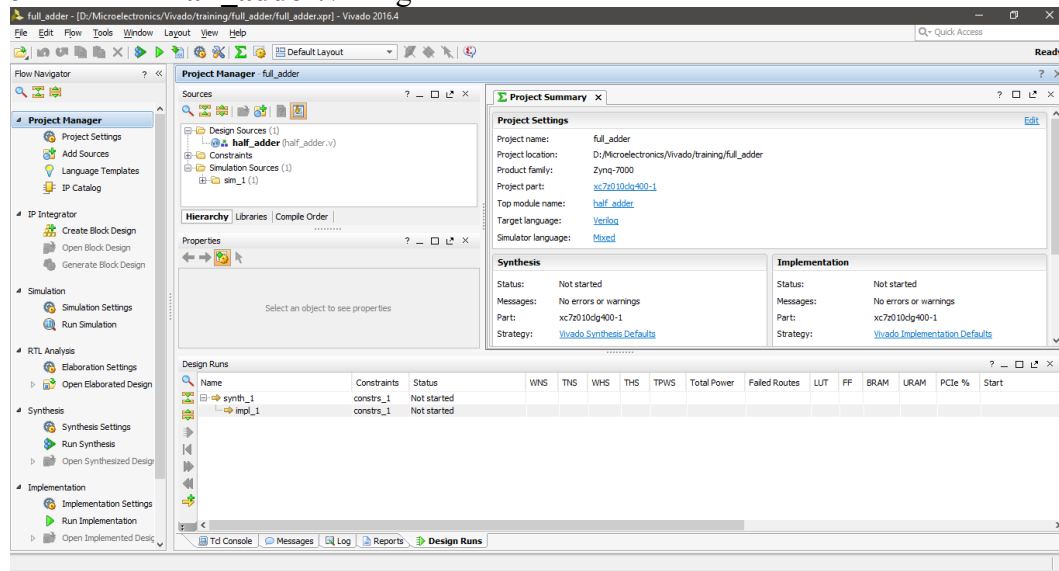
3. Buat modul baru dengan nama **half_adder**.



4. Klik **Yes** pada tampilan berikut.



5. Buka file **half_adder.v** dengan cara double klik.



c. Membuat modul half adder dan full adder

1. Masukkan kode Verilog ini untuk **half_adder.v**.

```
`timescale 1ns / 1ps

module half_adder
(
    input wire a,
    input wire b,
    output wire sum,
    output wire carry
);

    assign sum = a ^ b;
    assign carry = a & b;

endmodule
```

2. Buat source baru dengan nama **full_adder.v**, kemudian buat kode berikut ini.

```
`timescale 1ns / 1ps

module full_adder
(
    input wire a,
    input wire b,
    input wire c,
    output wire sum,
    output wire carry
);

    wire sum_0, carry_0, carry_1;

    half_adder half_adder_0
    (
        .a(a),
        .b(b),
```

```

        .sum(sum_0),
        .carry(carry_0)
    );

    half_adder half_adder_1
    (
        .a(sum_0),
        .b(c),
        .sum(sum),
        .carry(carry_1)
    );

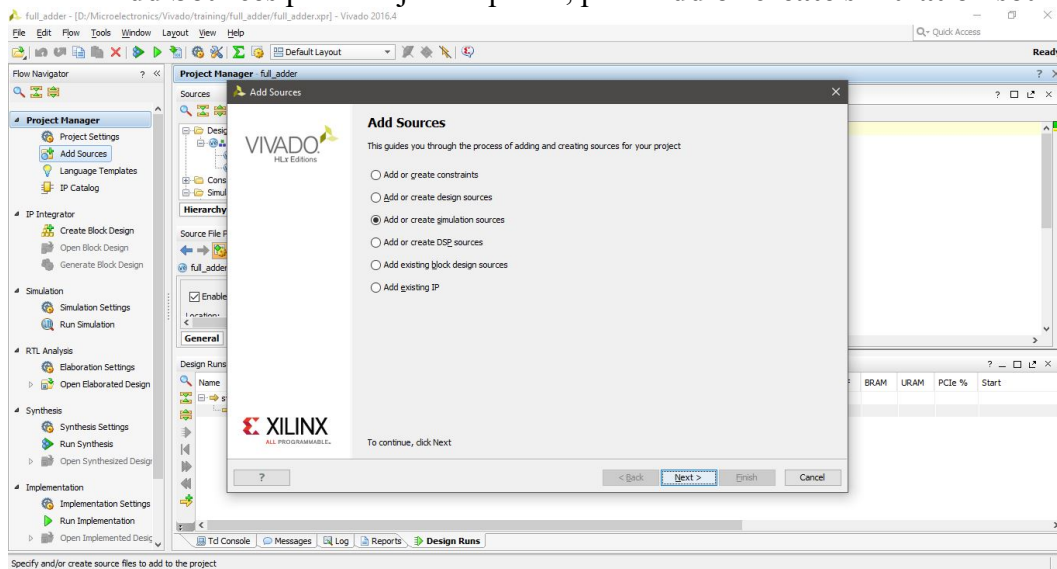
    assign carry = carry_0 | carry_1;

endmodule

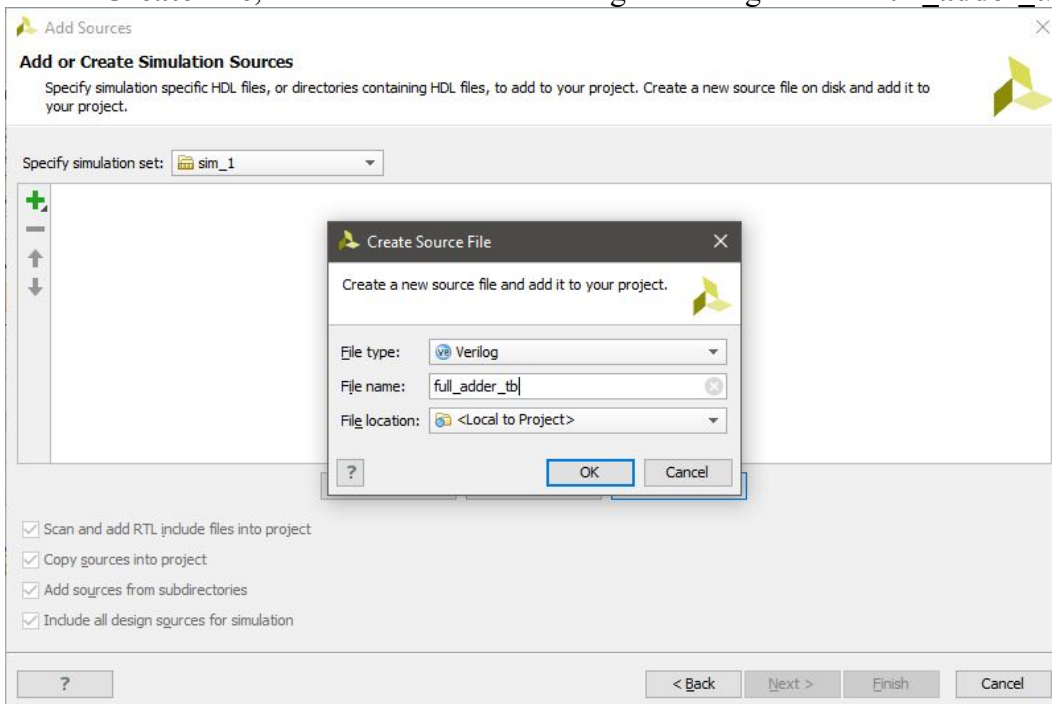
```

d. Membuat file testbench

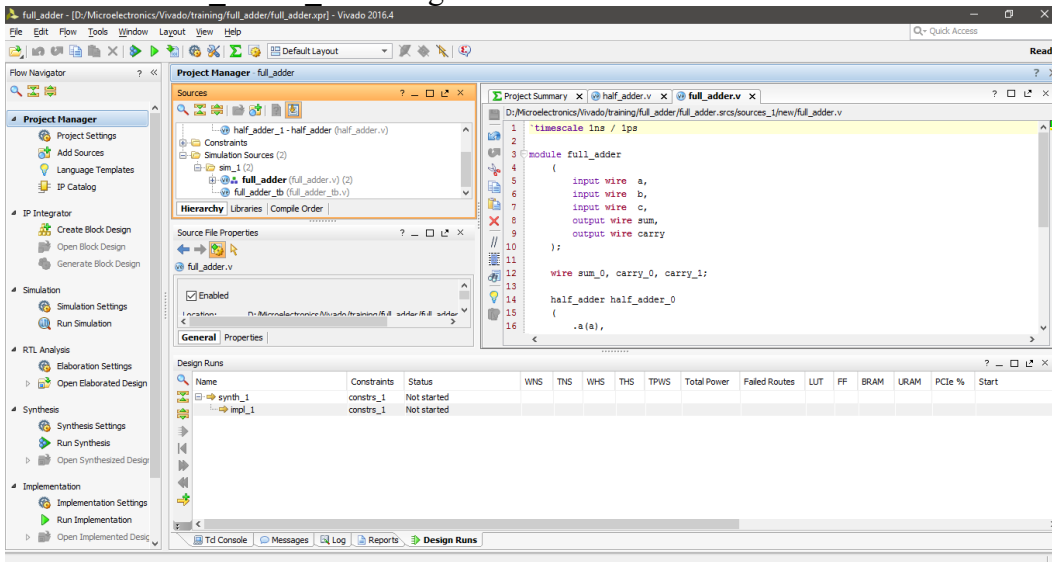
1. Klik Add Sources pada Project Explorer, pilih Add or create simulation sources.



2. Klik **Create File**, untuk membuat file Verilog baru dengan nama **full_adder_tb.v**.



3. Buka file **full_adder_tb.v** dengan cara double klik.



4. Masukkan kode Verilog ini untuk **full_adder_tb.v**.

```
`timescale 1ns / 1ps

module full_adder_tb();
    localparam T = 10;

    reg a;
    reg b;
    reg c;
    wire sum;
```

```

wire carry;

full_adder full_adder_0
(
    .a(a),
    .b(b),
    .c(c),
    .sum(sum),
    .carry(carry)
);

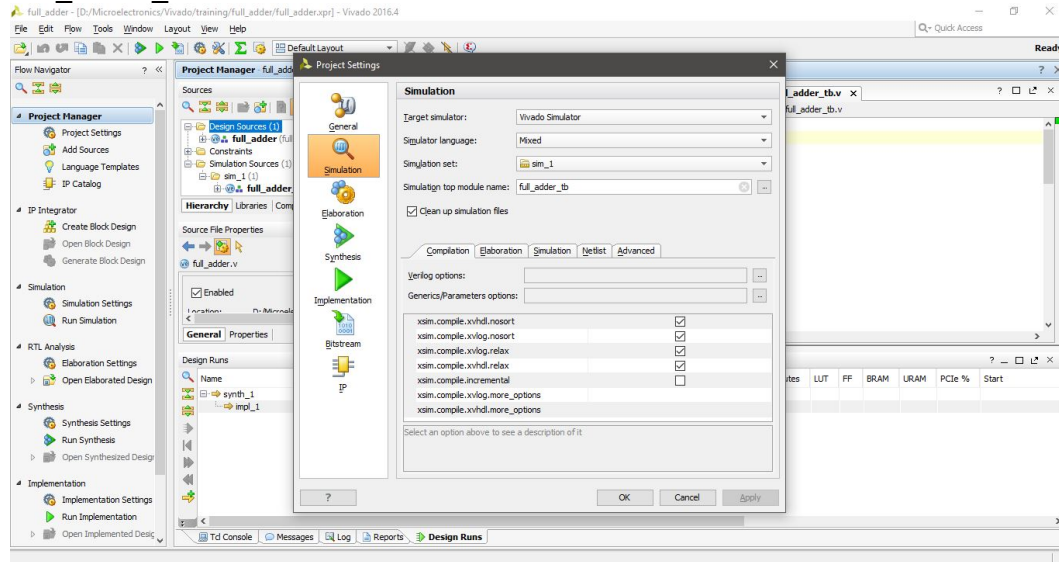
initial
begin
    a = 0;
    b = 0;
    c = 0;
    #T;
    a = 0;
    b = 0;
    c = 1;
    #T;
    a = 0;
    b = 1;
    c = 0;
    #T;
    a = 0;
    b = 1;
    c = 1;
    #T;
    a = 1;
    b = 0;
    c = 0;
    #T;
    a = 1;
    b = 0;
    c = 1;
    #T;
    a = 1;
    b = 1;
    c = 0;
    #T;
    a = 1;
    b = 1;
    c = 1;
    #T;
end

endmodule

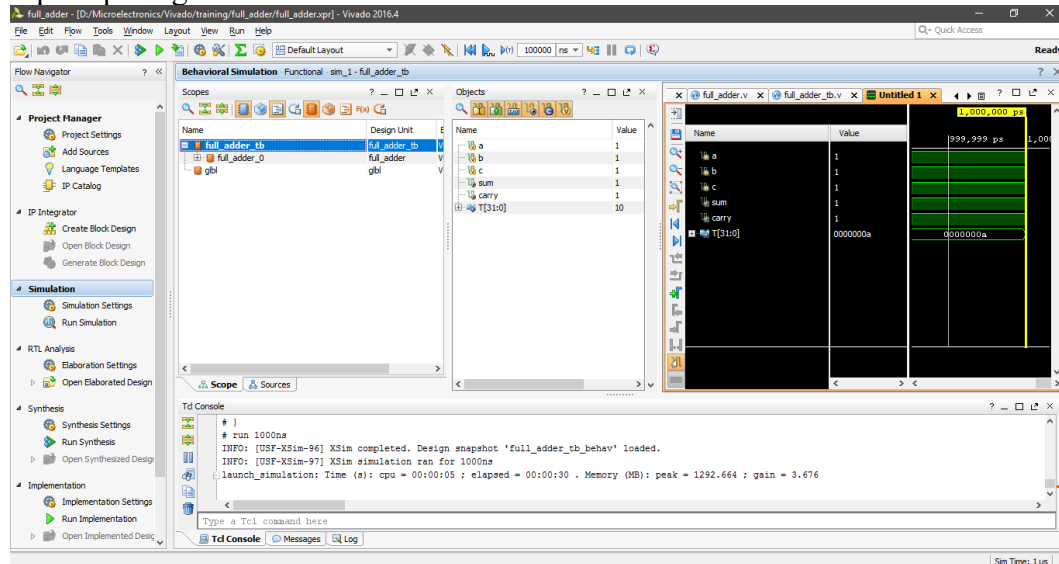
```

e. **Simulasi design pada Vivado**

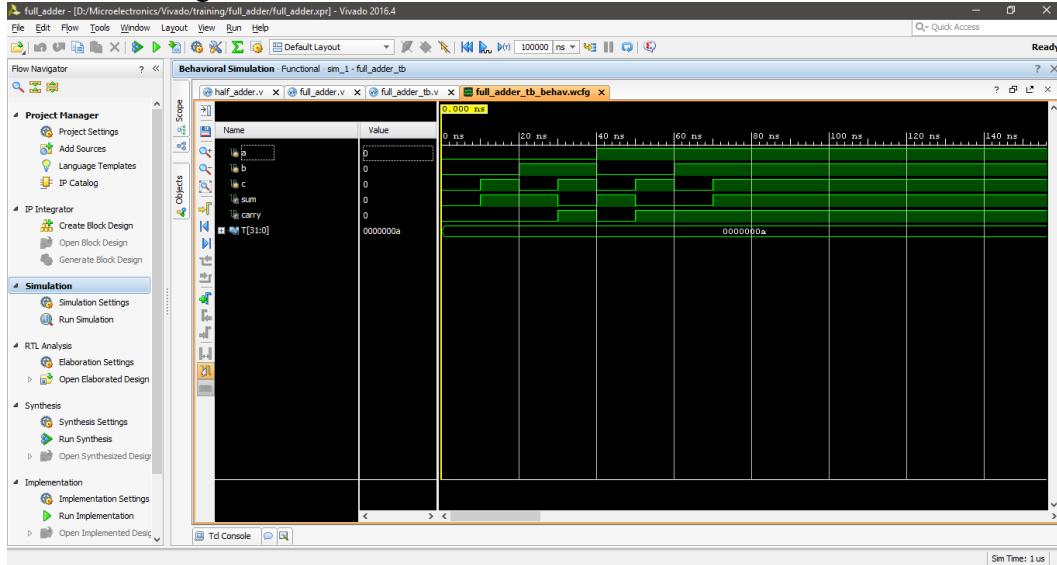
1. Pada Project Explorer, klik **Simulation Settings**. Pastikan top module nya adalah **full_adder_tb**.



2. Run simulation dengan cara klik, **Run Simulation** pada Project Explorer. Hasil simulasi seperti pada gambar berikut.

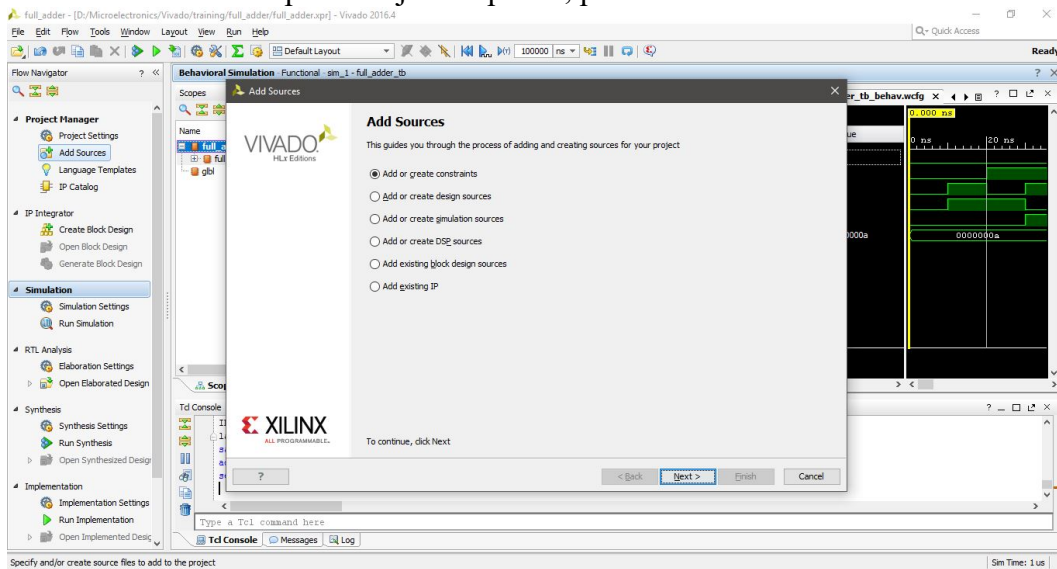


3. Save waveform dengan cara **File → Save Waveform Configuration As**. Hasil simulasi full adder sebagai berikut.

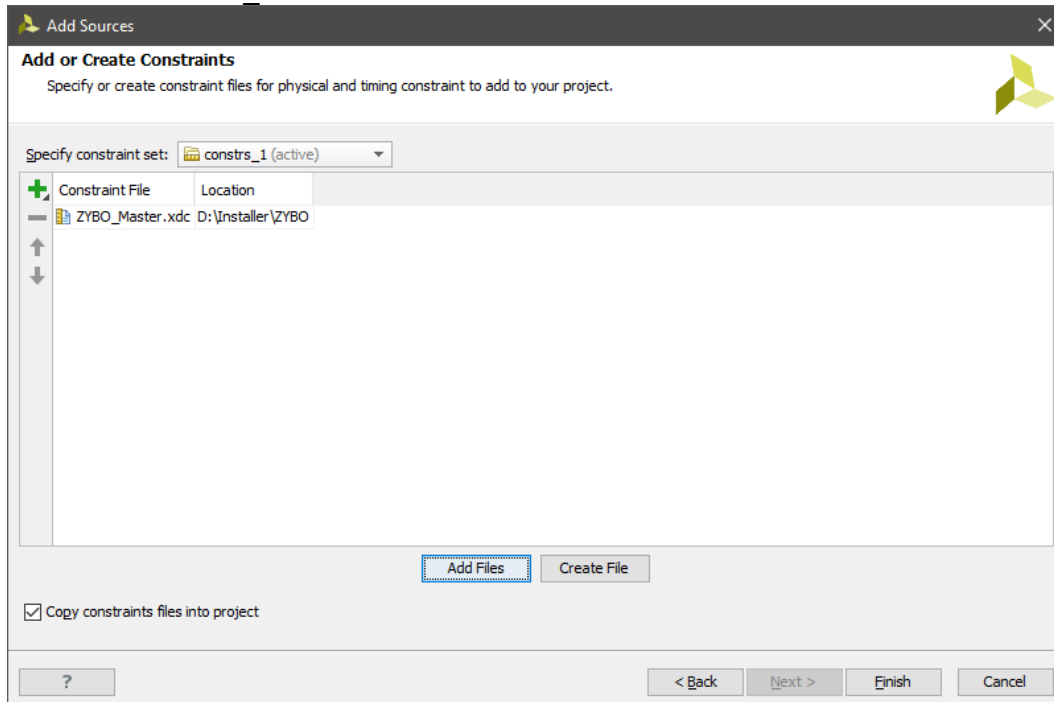


f. Mapping pin ke I/O

1. Klik **Add Sources** pada Project Explorer, pilih **Add or create constraints**.



2. Add file **ZYBO_master.xdc**.



3. Hubungkan pin switches ke input full adder, a, b, dan c.

```
##Switches
##IO_L19N_T3_VREF_35
set_property PACKAGE_PIN G15 [get_ports {a}]
set_property IOSTANDARD LVCMOS33 [get_ports {a}]

##IO_L24P_T3_34
set_property PACKAGE_PIN P15 [get_ports {b}]
set_property IOSTANDARD LVCMOS33 [get_ports {b}]

##IO_L4N_T0_34
set_property PACKAGE_PIN W13 [get_ports {c}]
set_property IOSTANDARD LVCMOS33 [get_ports {c}]
```

4. Hubungkan pin LED ke output full adder, sum dan carry.

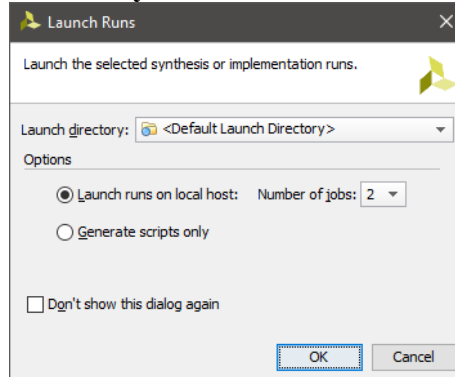
```
##LEDs
##IO_L23P_T3_35
set_property PACKAGE_PIN M14 [get_ports {sum}]
set_property IOSTANDARD LVCMOS33 [get_ports {sum}]

#IO_L23N_T3_35
set_property PACKAGE_PIN M15 [get_ports {carry}]
set_property IOSTANDARD LVCMOS33 [get_ports {carry}]
```

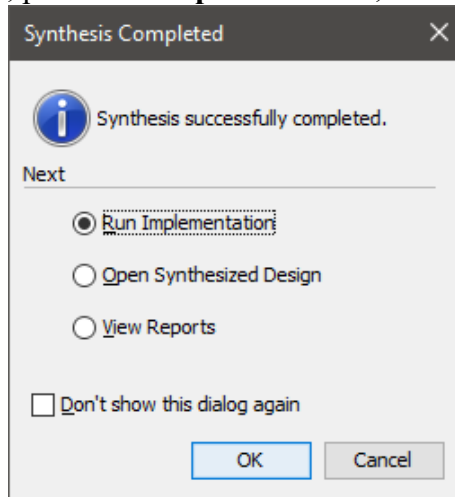
5. Save file **ZYBO_master.xdc** tersebut.

g. Synthesis design pada Vivado

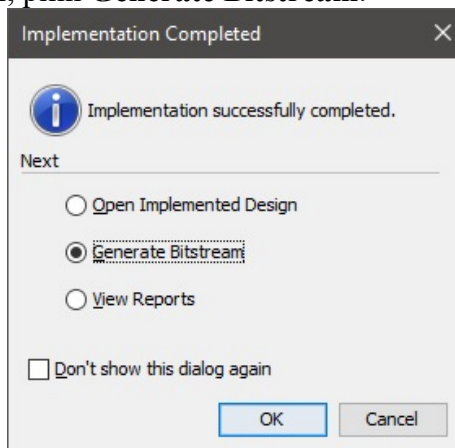
1. Pada Project Explorer, klik **Run Synthesis**, kemudian klik **Yes**.



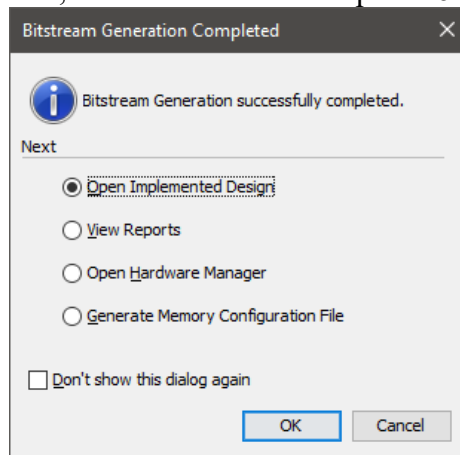
2. Setelah synthesis selesai, pilih **Run Implementation**, kemudian klik **Yes**.



3. Setelah synthesis selesai, pilih **Generate Bitstream**.

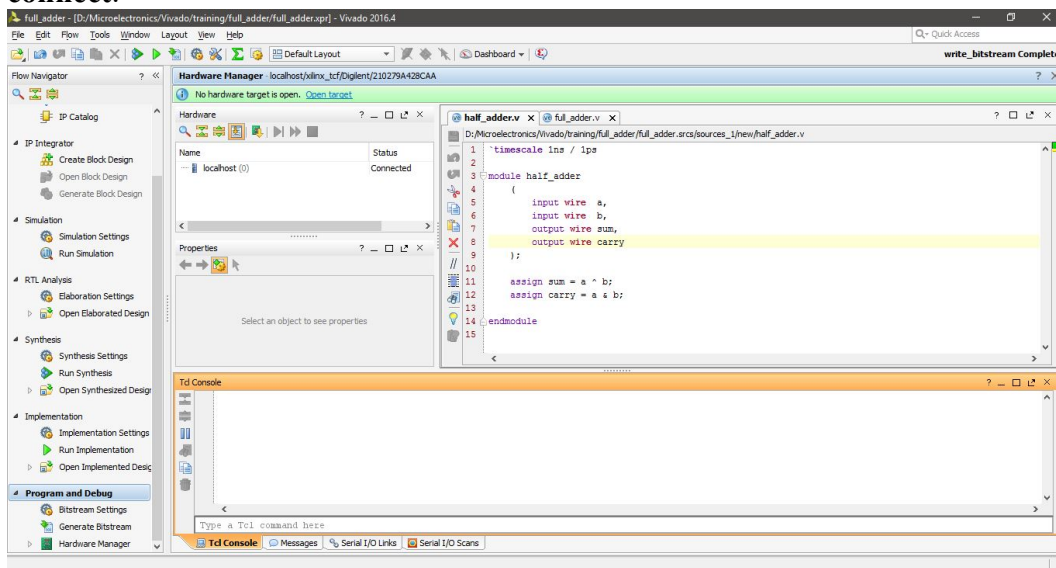


4. Setelah bitstream digenerate, window berikut ini dapat di **close**.

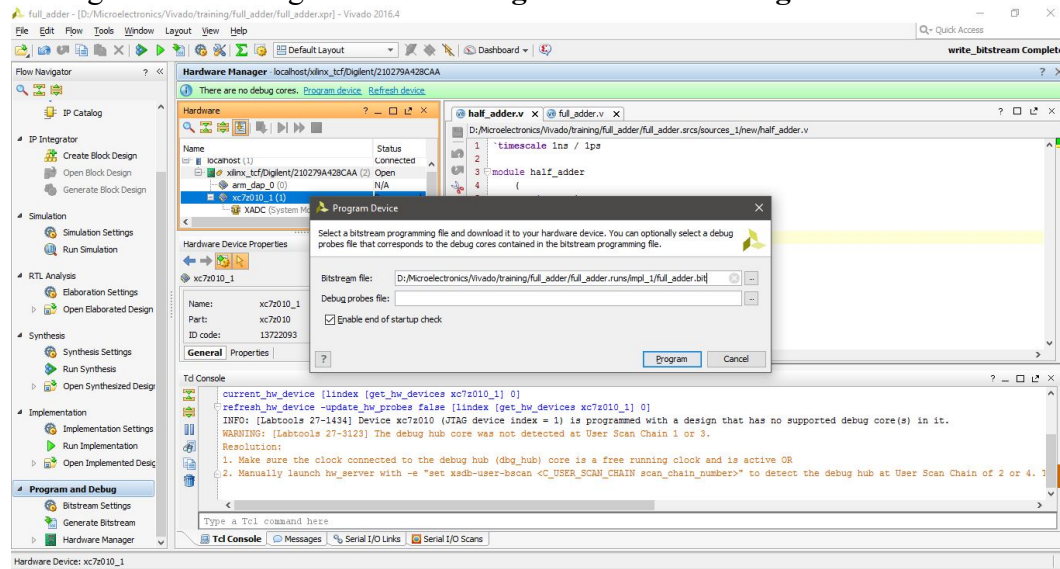


h. Upload ke board ZYBO

1. Pada Project Explorer, **Program and Debug**, pilih **Hardware Manager**, kemudian hubungkan board ZYBO ke development PC, dan nyalakan. Klik **Open target** → **Auto connect**.



2. Program device dengan cara klik **Program device** → **Program**.



3. Hasil akhir dapat dilihat pada board dengan mengubah switch dan output pada LED.