

MODUL 5

IMPLEMENTASI RANGKAIAN DIGITAL PEMBENTUK DAN PENAMPIL ANGKA DI FPGA MENGGUNAKAN VERILOG-HDL

5.1 Tujuan Praktikum Modul 5

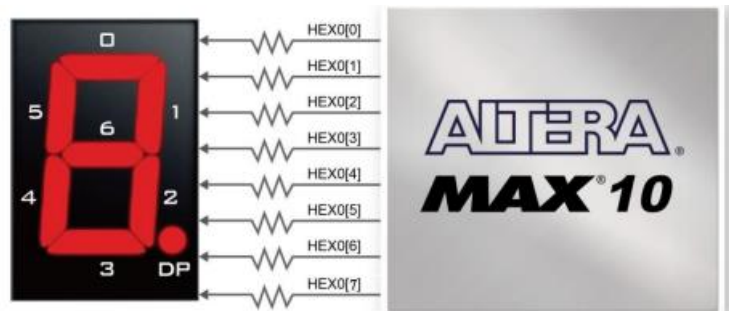
Setelah mempraktekkan Topik ini, mahasiswa diharapkan dapat ahami

1. Mem konsep dasar FPGA dan Verilog-HDL dalam merancang rangkaian digital.
2. Merancang, mensimulasikan, dan mengimplementasikan rangkaian pembentuk angka di FPGA.
3. Memahami teknik penampilan angka menggunakan output dari rangkaian digital.
4. Menguji dan memverifikasi fungsi rangkaian digital yang dirancang menggunakan Verilog-HDL di FPGA.

5.2 Dasar Teori Praktikum Modul 5

5.2.1 7-Segment

Seven segment adalah suatu segmen – segmen yang digunakan untukmenampilkkan angka/bilangan decimal. Seven segment ini terdiri dari 7 batang LEDyang disusun membentuk angka 8 dengan menggunakan huruf a-g yang disebutDOT MATRIKS. Setiap segment ini terdiri dari 1 atau 2 LED (Light Emitting Dioda).



Papan DE10-Lite memiliki enam display 7-segmen untuk menampilkan angka. Gambar 3-17 menunjukkan koneksi dari tujuh segmen (anoda umum) ke pin pada FPGA MAX 10. Segmen dapat dinyalakan atau dimatikan dengan menerapkan level logika rendah atau tinggi dari FPGA, masing-masing.

Setiap segmen dalam sebuah display diindeks dari 0 hingga 6 dan DP (titik desimal), dengan posisi yang sesuai seperti yang diberikan dalam Gambar 3-17. Tabel 3-6 menunjukkan penugasan pin FPGA ke display 7-segmen.

Berikut adalah daftar pin planer untuk 7-segmen:

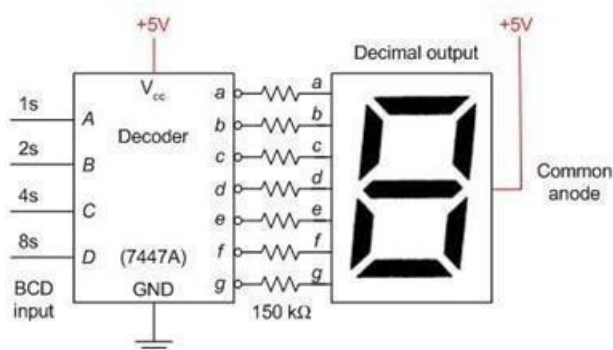
Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTTL
HEX07	PIN_D15	Seven Segment Digit 0[7], DP	3.3-V LVTTTL
HEX10	PIN_C18	Seven Segment Digit 1[0]	3.3-V LVTTTL
HEX11	PIN_D18	Seven Segment Digit 1[1]	3.3-V LVTTTL
HEX12	PIN_E18	Seven Segment Digit 1[2]	3.3-V LVTTTL
HEX13	PIN_B16	Seven Segment Digit 1[3]	3.3-V LVTTTL

Modul Praktikum

HEX14	PIN_A17	Seven Segment Digit 1[4]	3.3-V LVTTTL
HEX15	PIN_A18	Seven Segment Digit 1[5]	3.3-V LVTTTL
HEX16	PIN_B17	Seven Segment Digit 1[6]	3.3-V LVTTTL
HEX17	PIN_A16	Seven Segment Digit 1[7] , DP	3.3-V LVTTTL
HEX20	PIN_B20	Seven Segment Digit 2[0]	3.3-V LVTTTL
HEX21	PIN_A20	Seven Segment Digit 2[1]	3.3-V LVTTTL
HEX22	PIN_B19	Seven Segment Digit 2[2]	3.3-V LVTTTL
HEX23	PIN_A21	Seven Segment Digit 2[3]	3.3-V LVTTTL
HEX24	PIN_B21	Seven Segment Digit 2[4]	3.3-V LVTTTL
HEX25	PIN_C22	Seven Segment Digit 2[5]	3.3-V LVTTTL
HEX26	PIN_B22	Seven Segment Digit 2[6]	3.3-V LVTTTL
HEX27	PIN_A19	Seven Segment Digit 2[7] , DP	3.3-V LVTTTL
HEX30	PIN_F21	Seven Segment Digit 3[0]	3.3-V LVTTTL
HEX31	PIN_E22	Seven Segment Digit 3[1]	3.3-V LVTTTL
HEX32	PIN_E21	Seven Segment Digit 3[2]	3.3-V LVTTTL
HEX33	PIN_C19	Seven Segment Digit 3[3]	3.3-V LVTTTL
HEX34	PIN_C20	Seven Segment Digit 3[4]	3.3-V LVTTTL
HEX35	PIN_D19	Seven Segment Digit 3[5]	3.3-V LVTTTL
HEX36	PIN_E17	Seven Segment Digit 3[6]	3.3-V LVTTTL
HEX37	PIN_D22	Seven Segment Digit 3[7] , DP	3.3-V LVTTTL
HEX40	PIN_F18	Seven Segment Digit 4[0]	3.3-V LVTTTL
HEX41	PIN_E20	Seven Segment Digit 4[1]	3.3-V LVTTTL
HEX42	PIN_E19	Seven Segment Digit 4[2]	3.3-V LVTTTL
HEX43	PIN_J18	Seven Segment Digit 4[3]	3.3-V LVTTTL
HEX44	PIN_H19	Seven Segment Digit 4[4]	3.3-V LVTTTL
HEX45	PIN_F19	Seven Segment Digit 4[5]	3.3-V LVTTTL
HEX46	PIN_F20	Seven Segment Digit 4[6]	3.3-V LVTTTL
HEX47	PIN_F17	Seven Segment Digit 4[7] , DP	3.3-V LVTTTL
HEX50	PIN_J20	Seven Segment Digit 5[0]	3.3-V LVTTTL
HEX51	PIN_K20	Seven Segment Digit 5[1]	3.3-V LVTTTL
HEX52	PIN_L18	Seven Segment Digit 5[2]	3.3-V LVTTTL
HEX53	PIN_N18	Seven Segment Digit 5[3]	3.3-V LVTTTL
HEX54	PIN_M20	Seven Segment Digit 5[4]	3.3-V LVTTTL
HEX55	PIN_N19	Seven Segment Digit 5[5]	3.3-V LVTTTL
HEX56	PIN_N20	Seven Segment Digit 5[6]	3.3-V LVTTTL
HEX57	PIN_L19	Seven Segment Digit 5[7] , DP	3.3-V LVTTTL

5.2.2 BCD to 7-Segment

Decoder BCD to Seven segment adalah decoder yang mengubah nilai binerBCD ke dalam tujuh bit data seven segment untuk ditampilkan nilai desimalnya secara visual Bagan Decoder BCD to seven segment dan tampilan display seven segment.



Modul Praktikum

Adapun data keluaran Decoder BCD to Seven segment ditunjukkan pada table dibawah ini:.

DESIMAL	D	C	B	A	a	b	c	d	e	f	g	7-LED
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0	2
3	0	0	1	1	0	0	0	0	1	1	0	3
4	0	1	0	0	1	0	0	1	1	0	0	4
5	0	1	0	1	0	1	0	0	1	0	0	5
6	0	1	1	0	1	1	0	0	0	0	0	6
7	0	1	1	1	0	0	0	1	1	1	1	7
8	1	0	0	0	0	0	0	0	0	0	0	8
9	1	0	0	1	0	0	0	1	1	0	0	9
10	1	0	1	0	1	1	1	0	0	1	0	10
11	1	0	1	1	1	1	0	0	1	1	0	11
12	1	1	0	0	1	0	1	1	1	0	0	12
13	1	1	0	1	0	1	1	0	1	0	0	13
14	1	1	1	0	1	1	1	0	0	0	0	14
15	1	1	1	1	1	1	1	1	1	1	1	15

Setiap kombinasi nilai DCBA akan menampilkan simbol nilai desimal pada *seven segment*. Jika logika DCBA adalah '0000' maka *seven segment* akan menampilkan angka '0'. Jika nilai DCBA adalah '0001' maka *seven segment* akan menampilkan angka '1'. Dan seterusnya. Selengkapnya *seven segment* akan

Modul Praktikum

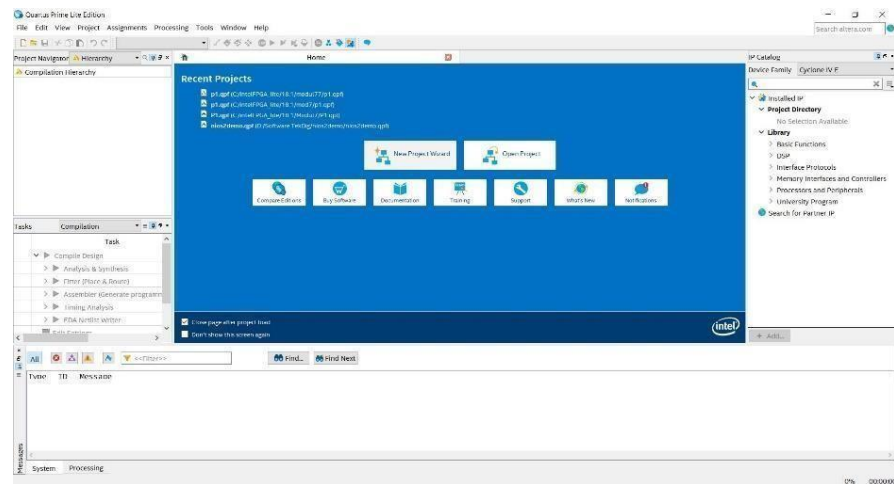
5.3 Lembar Kegiatan Praktikum Modul 5 :

5.3.1 Alat dan Bahan

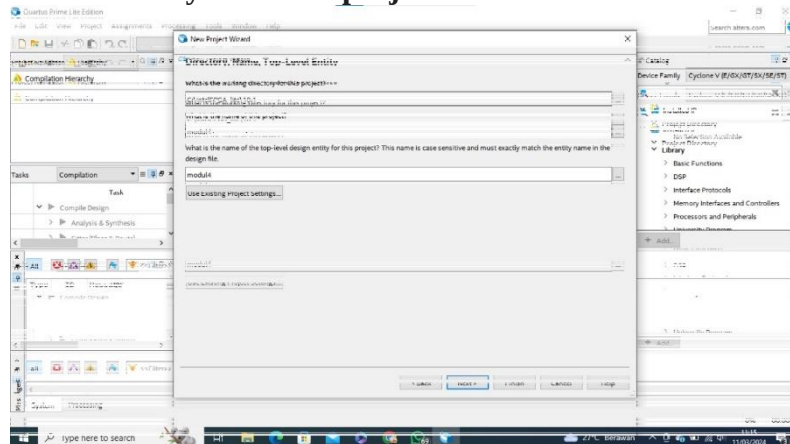
- Laptop yang telah terinstal *software Quartus 18*
- Mouse
- Board FPGA DE10-LITE

5.3.2 Langkah percobaan modul 5 (BCD to 7-Segment)

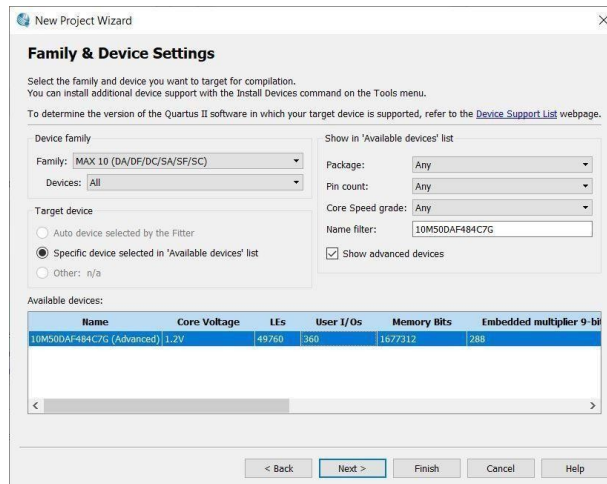
- Hidupkan Laptop
- Buka aplikasi quartus
- Pilih **New Project Wizard** ? next



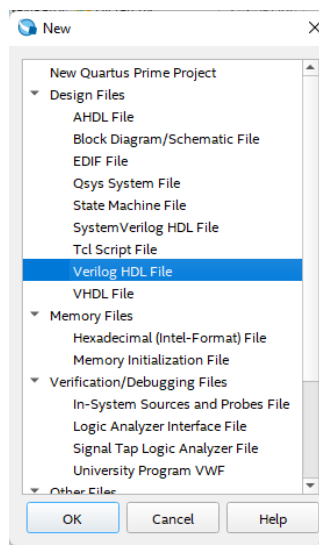
- Menentukan **directory** dan nama **project** next ? next ? next



- Menentukan **device family** dengan **MAX 10(DA/DF/DC/SA/SC)** dan name filter dengan **10M50DAF484C7G** → klik **10M50DAF484C7G** pada **available device** → **next** → **next** → **finish**



- Klik **File > New > Verilog HDL File** lalu tekan **OK**



- Setelah muncul workspace, tuliskan kode Verilog seperti gambar di bawah ini lalu **save**.

```

1 module MODUL5(
2     input [3:0] number,
3     output [6:0] segments
4 );
5
6 assign segments = (number == 4'b0000) ? 7'b1000000 : // 0
7                  (number == 4'b0001) ? 7'b1111001 : // 1
8                  (number == 4'b0010) ? 7'b0100100 : // 2
9                  (number == 4'b0011) ? 7'b0110000 : // 3
10                 (number == 4'b0100) ? 7'b0011001 : // 4
11                 (number == 4'b0101) ? 7'b0010010 : // 5
12                 (number == 4'b0110) ? 7'b0000010 : // 6
13                 (number == 4'b0111) ? 7'b1111000 : // 7
14                 (number == 4'b1000) ? 7'b0000000 : // 8
15                 (number == 4'b1001) ? 7'b0010000 : // 9
16                 7'b1111111; // All segments off for invalid in
17
18 endmodule
19
20 module DigitalNumberDisplay(
21     input clk,
22     input [7:0] data,
23     output reg [6:0] segments,
24     output reg dp
25 );
26
27 reg [3:0] digit0, digit1;
28
29 always @(posedge clk) begin
30     digit0 <= data[3:0];
31     digit1 <= data[7:4];
32 end

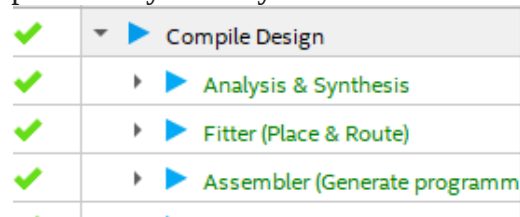
```

```

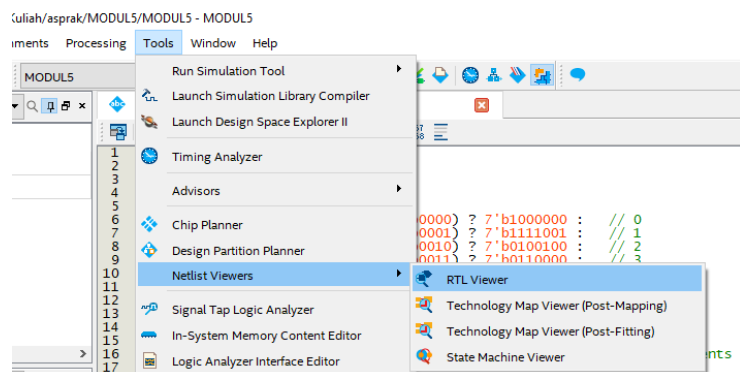
9      (number == 4'b0011) ? 7'b0110000 : // 3
10     (number == 4'b0100) ? 7'b0011001 : // 4
11     (number == 4'b0101) ? 7'b0010010 : // 5
12     (number == 4'b0110) ? 7'b0000010 : // 6
13     (number == 4'b0111) ? 7'b1111000 : // 7
14     (number == 4'b1000) ? 7'b0000000 : // 8
15     (number == 4'b1001) ? 7'b0010000 : // 9
16     7'b1111111; // All segments off for invalid in
17
18 endmodule
19
20 module DigitalNumberDisplay(
21     input clk,
22     input [7:0] data,
23     output reg [6:0] segments,
24     output reg dp
25 );
26
27 reg [3:0] digit0, digit1;
28
29 always @(posedge clk) begin
30     digit0 <= data[3:0];
31     digit1 <= data[7:4];
32 end
33
34 SevenSegmentDecoder seg_decoder0(.number(digit0), .segments(segments));
35 SevenSegmentDecoder seg_decoder1(.number(digit1), .segments(segments));
36
37 endmodule
38

```

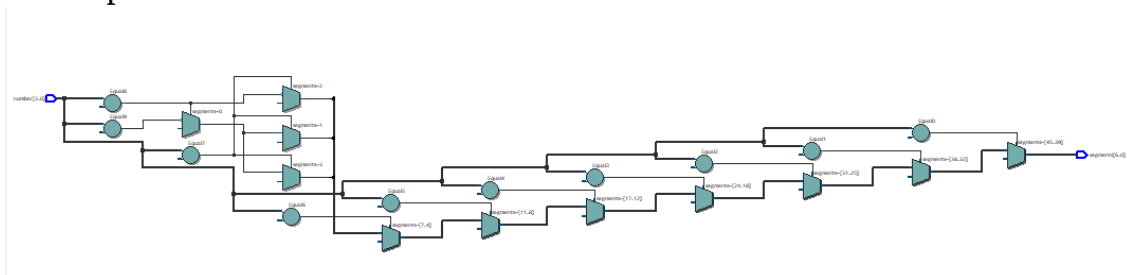
8. Kemudian klik dua kali pada **Analysis & Synthesis**.



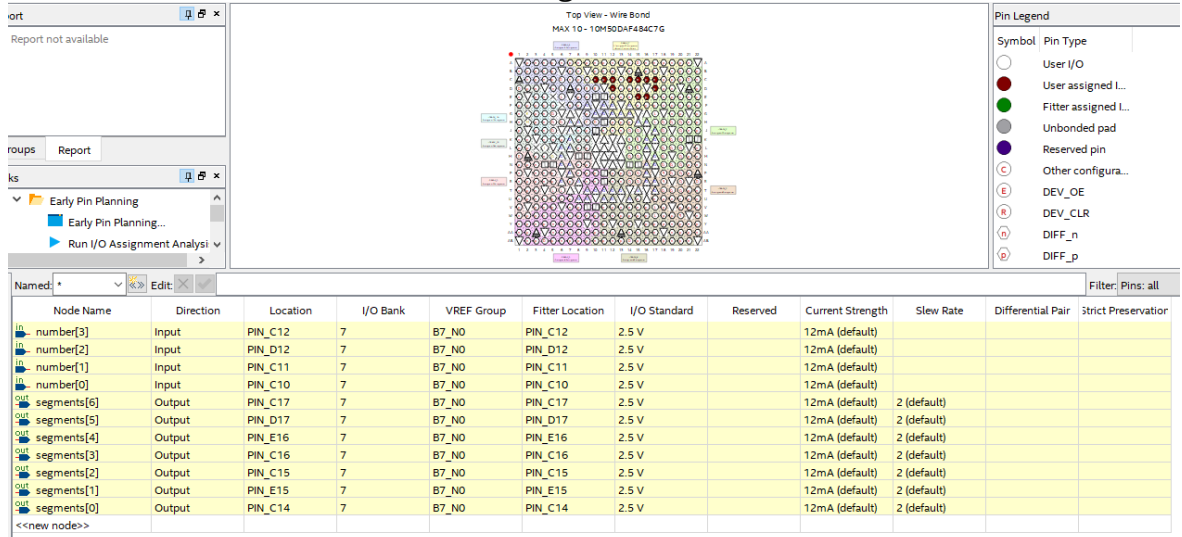
9. Setelah berhasil, klik **Tools** → **Netlist Viewers** → **RTL Viewer**.



10. Tampilan RTL Viewer



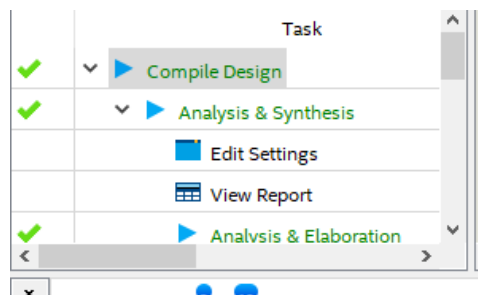
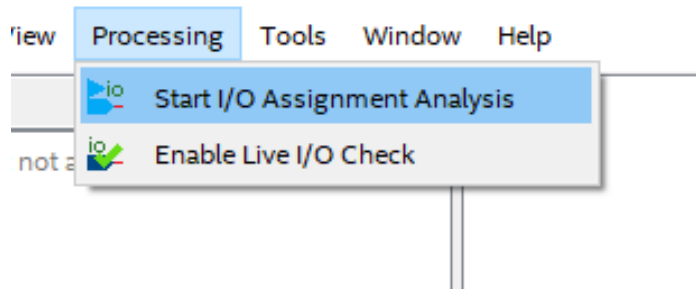
11. Masukan Pin Location sesuai gambar berikut:



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
number[3]	Input	PIN_C12	7	B7_N0	PIN_C12	2.5 V		12mA (default)			
number[2]	Input	PIN_D12	7	B7_N0	PIN_D12	2.5 V		12mA (default)			
number[1]	Input	PIN_C11	7	B7_N0	PIN_C11	2.5 V		12mA (default)			
number[0]	Input	PIN_C10	7	B7_N0	PIN_C10	2.5 V		12mA (default)			
segments[6]	Output	PIN_C17	7	B7_N0	PIN_C17	2.5 V		12mA (default)	2 (default)		
segments[5]	Output	PIN_D17	7	B7_N0	PIN_D17	2.5 V		12mA (default)	2 (default)		
segments[4]	Output	PIN_E16	7	B7_N0	PIN_E16	2.5 V		12mA (default)	2 (default)		
segments[3]	Output	PIN_C16	7	B7_N0	PIN_C16	2.5 V		12mA (default)	2 (default)		
segments[2]	Output	PIN_C15	7	B7_N0	PIN_C15	2.5 V		12mA (default)	2 (default)		
segments[1]	Output	PIN_E15	7	B7_N0	PIN_E15	2.5 V		12mA (default)	2 (default)		
segments[0]	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V		12mA (default)	2 (default)		

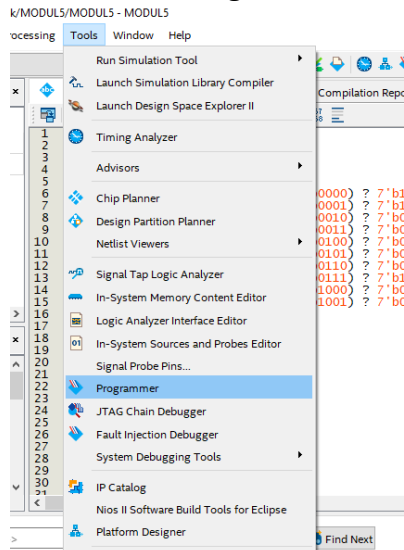
12. Klik Run I/O Assignment, lalu lakukan Compile Design Lagi.

- D:/02. Kuliah/asprak/MODUL5/MODUL5 - MODUL5



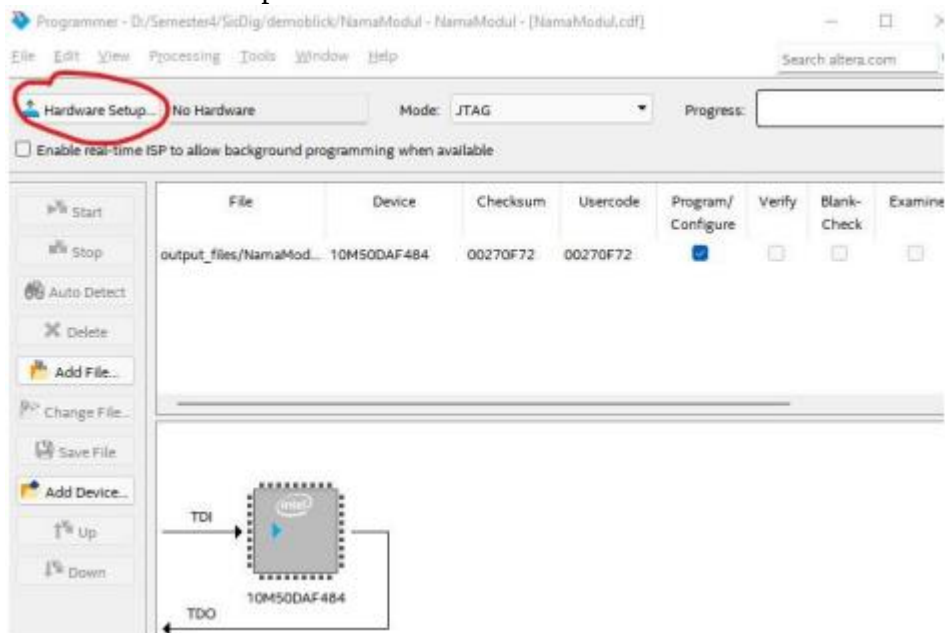
Modul Praktikum

13. Klik Tools > Programmer

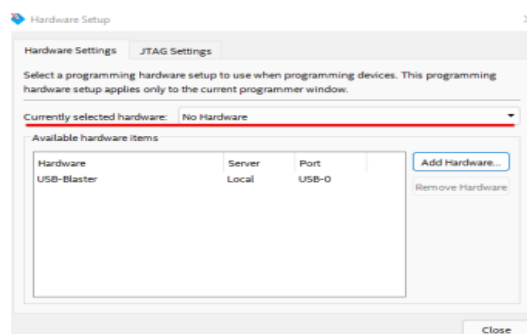


14. Hubungkan FPGA DE10-Lite ke Komputer/Laptop.

15. Klik Hardware Setup

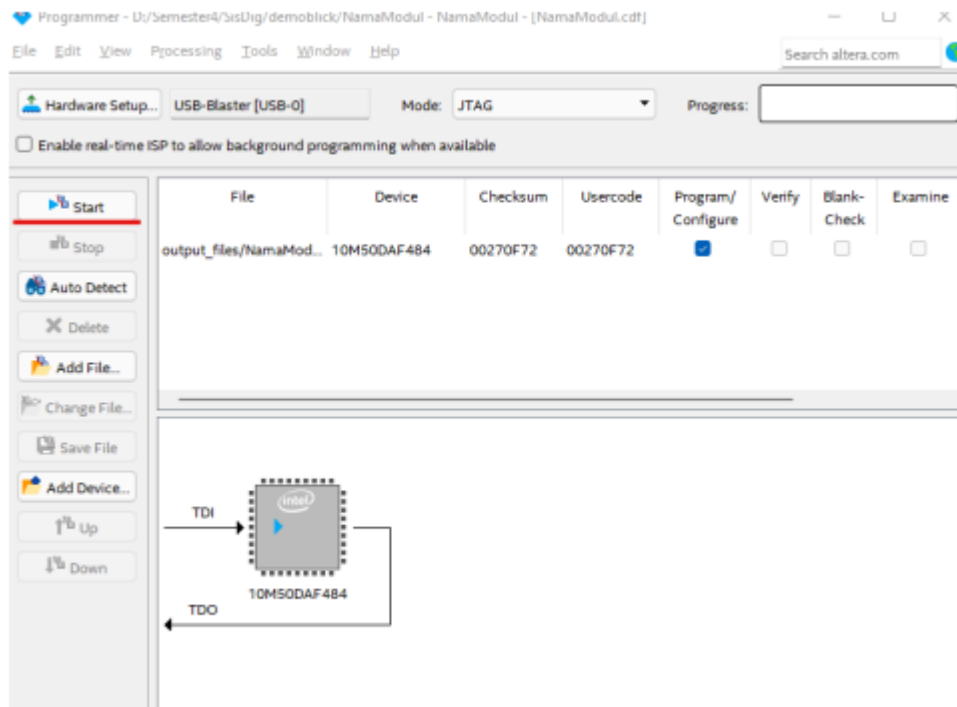


16. Pada Currently Selected Hardware ganti No Hardware Menjadi USB-Blaster. Lalu klik Close



Modul Praktikum

17. Klik Start.



18. Selesai.

