MODUL 7
SIMULASI RANGKAIAN MUX DAN DEMUX

7.1 Tujuan Praktikum Modul 7

Setelah mempraktekkan Topik ini, mahasiswa diharapkan dapat :

1. Dapat mengetahui dan memahami konsep dasar dari rangkaian *multiplexer* dan *demultiplexer.*
2. Dapat membuat rangkaian *multiplexer* dan *demultiplexer* dengan menggunakan bahasa VHDL atau Verilog.

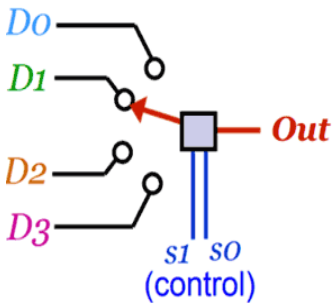
7.2 Dasar Teori Praktikum Modul 7

7.2.1 Multiplexer

Multiplexer (MUX) adalah perangkat yang memungkinkan informasi digital dari beberapa input dapat dialihkan kedalam satu jalur output untuk menuju tujuan bersama. Dengan kata lain, rangkaian ini mengacu pada memilih satu output dari banyak input yang tersedia.

Untuk memahami *multiplexer* adalah dengan melihat *single pole multi- positioned* seperti gambar dibawah. Disini terdapat *switch* dengan banyak *input* (D0, D1, D2, dan D3) tetapi hanya memiliki satu pin *output* (*out*). Tombol kontrol digunakan untuk memilih satu dari empat data yang tersedia dan data ini akan tercermin di sisi *output*. Dengan ini, pengguna dapat memilih sinyal yang diperlukan diantara banyak sinyal yang tersedia.

Gambar 7.1 Input dan output multiplexer



Ini adalah contoh sederhana dari *multiplexer* mekanik. Tetapi, dalam sirkuit elektronik yang melibatkan perpindahan kecepatan tinggi dan transfer data mengharuskan agar dapat memilih input yang dibutuhkan dengan sangat cepat menggunakan sirkuit digital. Sinyal kontrol (S1 dan S0) melakukan hal yang sama, yaitu memilih satu *input* dari banyak yang tersedia berdasarkan dari banyak sinyal yang diberikan.

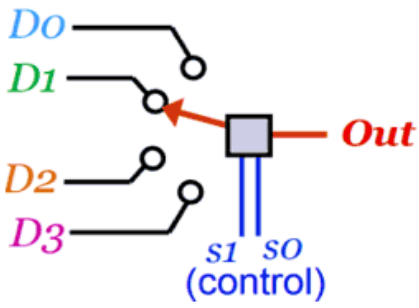
Jadi, ada tiga persyaratan minimum pada *multiplexer* apapun adalah sebagai berikut:

- **Pin *Input***: ini adalah sinyal yang tersedia yang harus dipilih. Sinyal-sinyal ini dapat berupa sinyal digital atau analog.
- **Pin *Output*:** pin yang menyediakan keluaran sinyal dari pin *input* yang dipilih. *Multiplexer* akan selalu memiliki satu pin *input*.
- **Pin kontrol:** digunakan untuk memilih sinyal dari pin input. Jumlah kontrol pin pada multiplexer tergantung pada jumlah pin input. Misalnya *multiplexer* 4-*input* akan memiliki 2 pin sinyal.

Tabel kebenaran di bawah menggambarkan status pin kontrol (S0 dan S1) dari gambar diatas.

Gambar 7.2 Status pin

S1	S0	Out
0	0	D0
0	1	D1
1	0	D2
1	1	D3

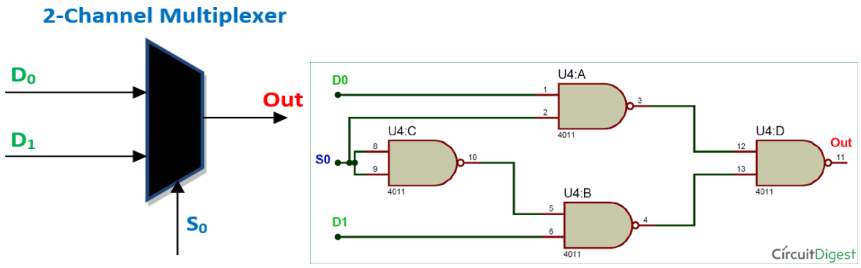


7.2.2 Tipe Multiplexer

7.2.2.1 2-Channel Multiplexer

Pada 2-*channel Multiplexer* akan memiliki 2 *input* dan satu *output*. Juga hanya memiliki satu pin kontrol untuk memilih antara 2 pin *input* yang tersedia.

Gambar 7.3 Rangkaian 2-channel multiplexer



Ketika pada S0 ditetapkan logika 0 maka input D0 akan tercermin pada pin output dan jika S0 tetap logika 1 maka input D1 yang akan tercermin pada pin output. Dapat ditujukan pada tabel kebenaran dibawah ini:

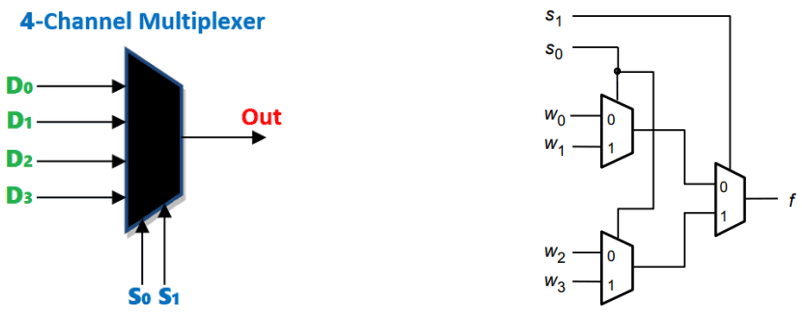
Tabel 7.1 Rangkaian 2-channel multiplexer

S0	D0	D1	Out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

7.2.2.2 4-Channel Multiplexer

Pada 4-Channel multiplexer akan memiliki 4 pin input dan 1 pin output dengan 2 pin kontrol. Dua pin kontrol ini akan membentuk 4 sinyal logika yang berbeda dan untuk setiap sinyal satu input tertentu akan dipilih. Multiplexer ini merupakan kombinasi dari 3 buah 2-channel multiplexer.

Gambar 7.4 Rangkaian 4-channel multiplexer



Pin kontrol dari dua MUX pertama dihubungkan bersama untuk membentuk pin kontrol pertama (S0) dan kemudian pin kontrol MUX ke tiga digunakan sebagai pembentuk pin kontrol kedua (S1). Dengan demikian, mendapatkan *multiplexer* dengan 4 *input* (W0, W1, W2, W3) dan hanya satu *output* (f). Tabel kebenaran untuk *multiplexer* 4:1 ditunjukan dibawah ini.

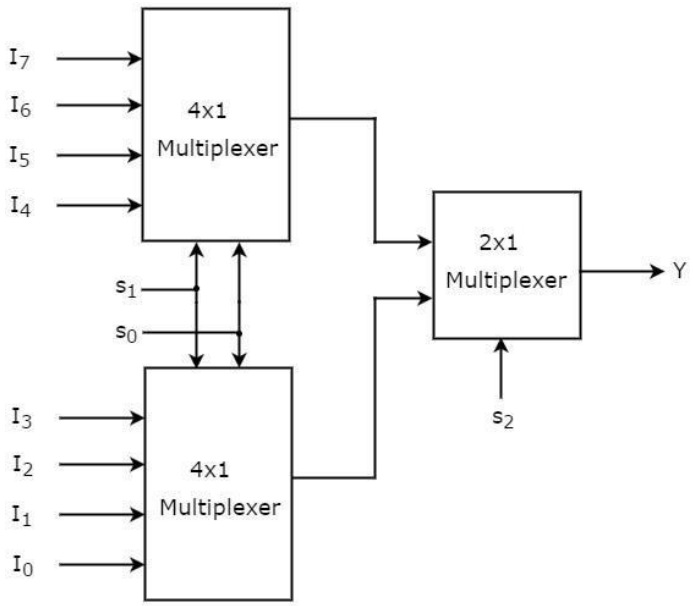
Tabel 7.2 Tabel kebenaran 4-channel multiplexer

S0	S0	Out (f)
0	0	W_0
0	1	W_1
1	0	W_2
1	1	W_3

7.2.2.3 8-Channel Multiplexer

8-Channel Multiplexer merupakan kombinasi dari 2 buah 4-channel multiplexer dan 1 buah 2-channel multiplexer. Sehingga diperoleh 8 input, 3 pin kontrol, dan 1 output.

Gambar 7.5 Rangkaian 8-channel multiplexer



Tabel kebenaran dari 8-Channel Multiplexer seperti yang ditunjukan dibawah ini:

Tabel 7.3 Tabel kebenaran 8-channel multiplexer

Selection Inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

7.2.3 Implementasi Multiplexer

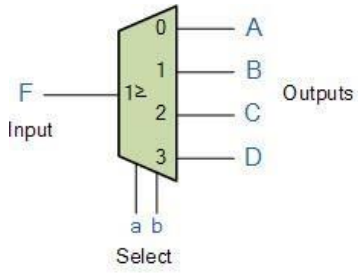
Digunakan untuk transmisi jaringan jarak jauh baik yang menggunakan kabel maupun yang menggunakan media udara seperti wireless atau radio.Sebagai contoh satu helai optic Surabaya–Jakarta bisa dipakai untuk menyalurkan ribuan percakapan pada telepon. Dan juga digunakan untuk remot TV (atau sejenisnya) dan kalkulator yang mempunyai beberapa inputan ke dalam satu outputan.

7.2.4 Demultiplexer

Demultiplexer (DEMUX) adalah rangkaian logika kombinasional yang dirancang untuk mengalihkan satu jalur input ke salah satu dari beberapa jalur output. Dengan kata lain, demultiplexer merupakan kebalikan dari *multiplexer*.

Pada *demultiplexer*, masukkan data dapat terdiri dari beberapa bit. Keluarannya terdiri dari beberapa jalur, masing-masing jalur terdiri dari satu bit atau lebih. Bit untuk masukan pada selektor (kontrol) tergantung banyaknya jalur.

Gambar 7.6 Rangkaian demultiplexer

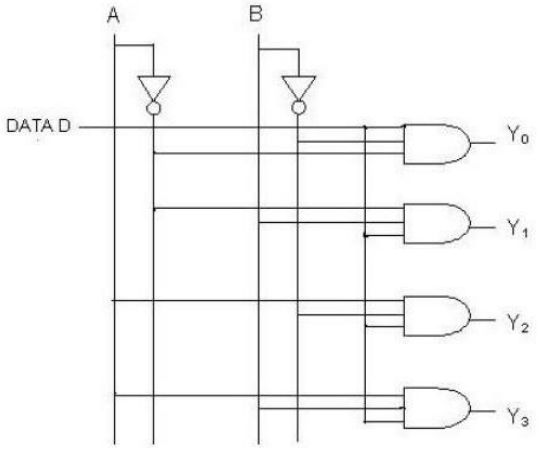


7.2.5 Tipe Demultiplexer

7.2.5.1 1 To 4 Demultiplexer

1 *to* 4 *Channel Demultiplexer* terdiri dari satu *input*, empat *output*, dan dua pin kontrol untuk membuat pilihan. Diagram dibawah menunjukkan rangkaian 1 *to* 4 *Channel Demultiplexer*.

Gambar 7. 7 Rangkaian 1 to 4 demultiplexer



Data D adalah bit *input* dengan 2 pin kontrol yaitu A dan B. Bit *input* D ditransmisikan ke empat bit *output* yaitu Y0, Y1, Y2, Y3. Ketika AB adalah 0, gerbang AND kedua paling atas diaktifkan sedangkan gerbang AND lainnya dinonaktifkan. Dengan demikian, hanya satu data yang dikirimkan pada Y1. Jika D rendah, maka Y1 rendah dan jika D tinggi, Y1 tinggi. Nilai Y1 tergantung pada nilai D.

Jika input kontrol berubah menjadi AB =10, semua gerbang dinonaktifkan kecuali gerbang ketiga dari atas. Kemudian D ditransmisikan ke output Y2. Berikut adalah tabel kebenaran untuk 1 to 4 Channel Demultiplexer.

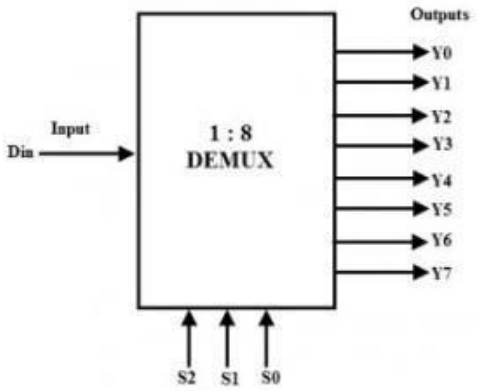
Tabel 7.4 Tabel kebenaran 1 to 4 demultiplexer

Input	Select Lines	Output Lines
I	S_1 S_0	D_0 D_1 D_2 D_3
I	0 0	1 0 0 0
I	0 1	0 1 0 0
I	1 0	0 0 1 0
I	1 1	0 0 0 1

2.2.5.2 1 To 8 Demultiplexer

1 to 8 Demultiplexer terdiri dari satu input, 8 output, dan 3 pin kontrol.

Gambar 7.8 Rangkaian 1 to 8 demultiplexer



Dibawah ini adalah tabel kebenaran untuk 1-to-8 Demultiplexer. Jika S0S1S2 = 000, maka output yang aktif adalah Y0 dan seterusnya.

Tabel 7.5 Tabel kebenaran 1 to 8 demultiplexer

Data Input	Select Inputs			Outputs							
D	S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

7.2.6 Implementasi Demultiplexer

Demultiplexer Biasanya digunakan pada televisi karena cara televisi adalah menerima sinyal data yang kemudian akan dipisahkan berdasarkan chanel yang ada cara ini disebut dengan teknik demultiplexing atau lebih jelasnya adalah proses penerimaan data dan kemudian akan dipisahkan sesuai dengan channel yang ada.

7.3 Lembar Kegiatan Praktikum Modul 7 :

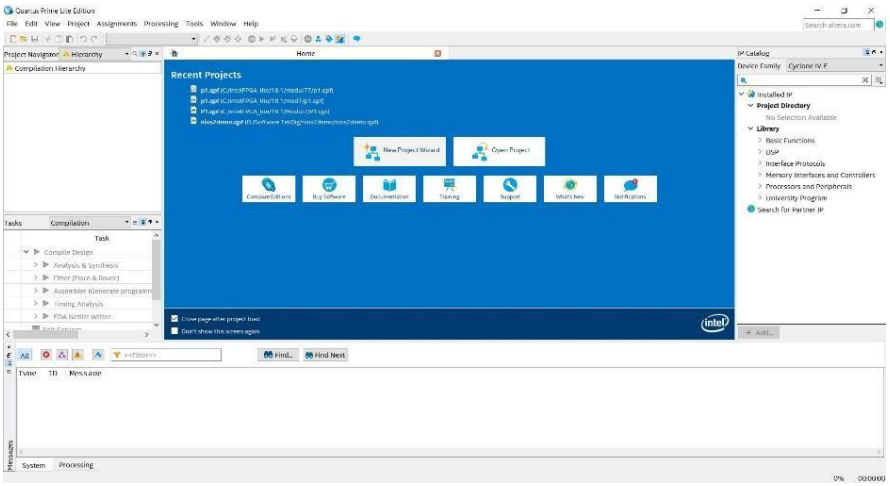
7.3.1 Alat dan Bahan

a. Laptop yang telah terinstal *software Quartus 18*

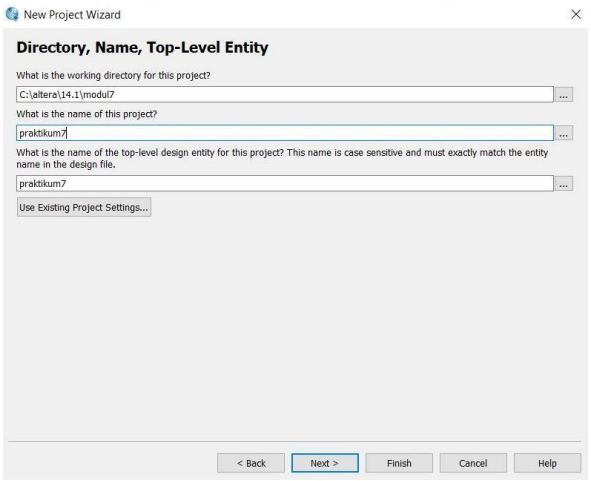
b. Mouse

7.3.2 Langkah percobaan modul 7 (4-channel multiplexer)

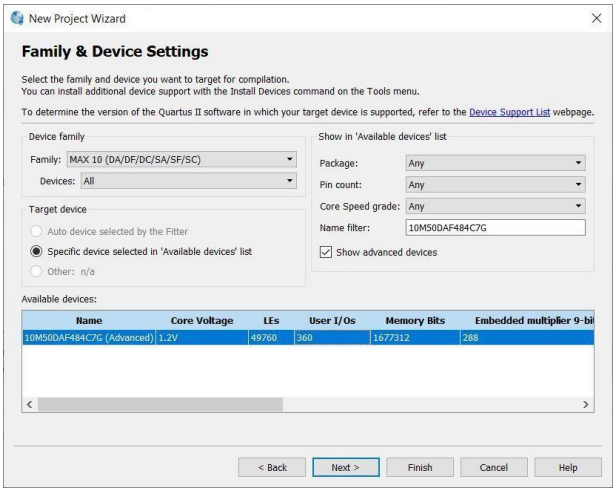
1. Hidupkan Laptop
2. Buka aplikasi quartus
3. Pilih **New Project Wizard → next**



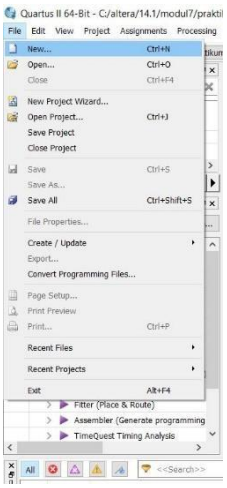
4. Menentukan **directory** dan nama **project** **next → next → next**

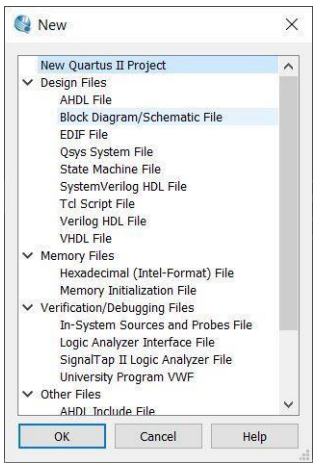


5. Menentukan **device family** dengan **MAX 10(DA/DF/DC/SA/SC)** dan name filter dengan **10M50DAF484C7G → klik 10M50DAF484C7G** pada **available device → next → next → finish**

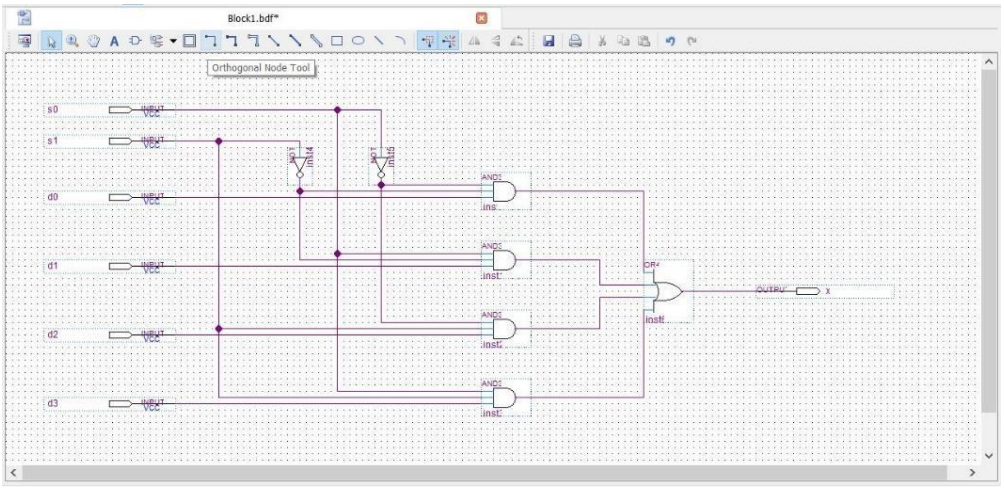


6. Pilih menu **file → new →** pilih **Block Diagram/Schematic File → ok**



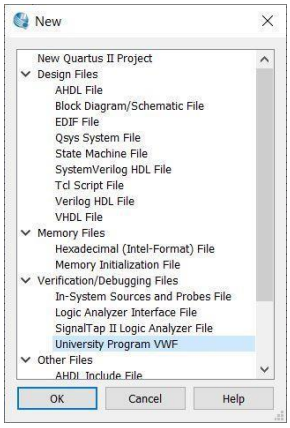


7. Membuat rangkaian multiplexer, klik **Symbol Tool → klik tanda panah → primitives → logic → pilih gerbang logika yang ada**. Dan buat inputan serta outputan dari rangkaian tersebut, klik **Pin Tool → pilih input serta output sesuaikan dengan rangkaian tersebut**, serta rename pin input dan output dan wiring sesuai dengan rangkaian.

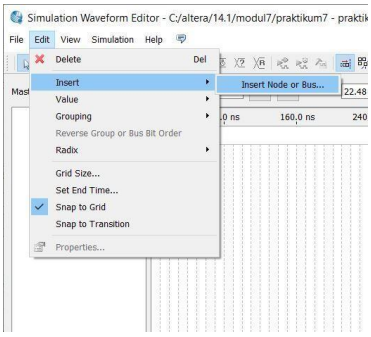


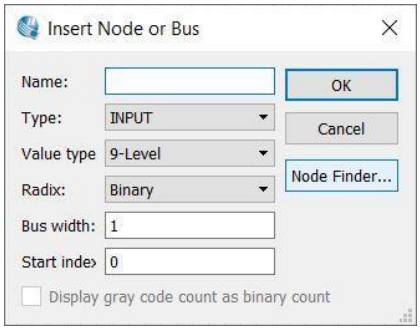
8. Setelah itu, klik **Compile design**. Tunggu hingga berhasil

9. Pilih menu **file → new →** pilih **Univertity Program VWF →** ok

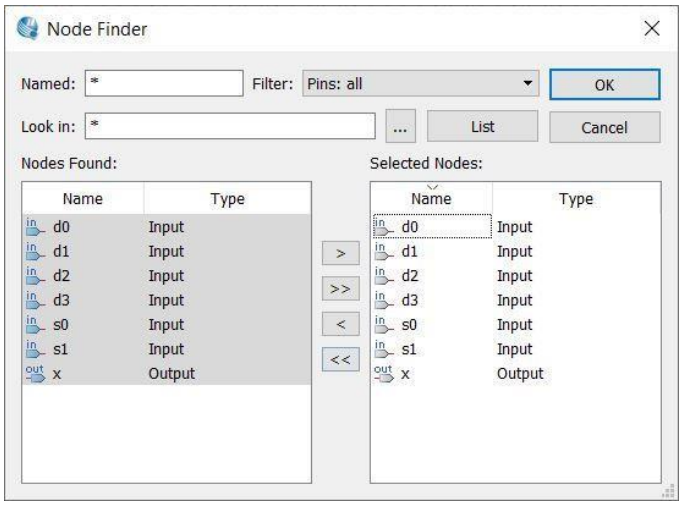


10. Pilih menu **Edit → Insert → Insert Node or Bus → Pilih Node Finder**





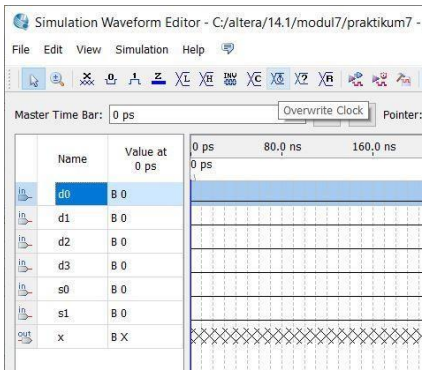
11. Klik **List** → klik **“>>”** → **ok** → **ok**

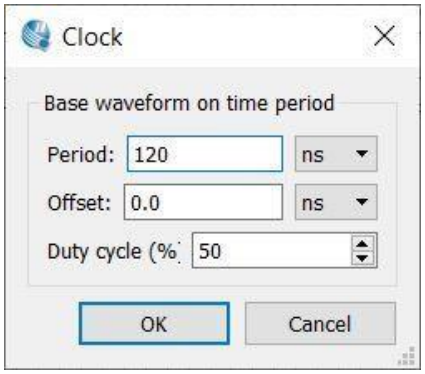


12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :

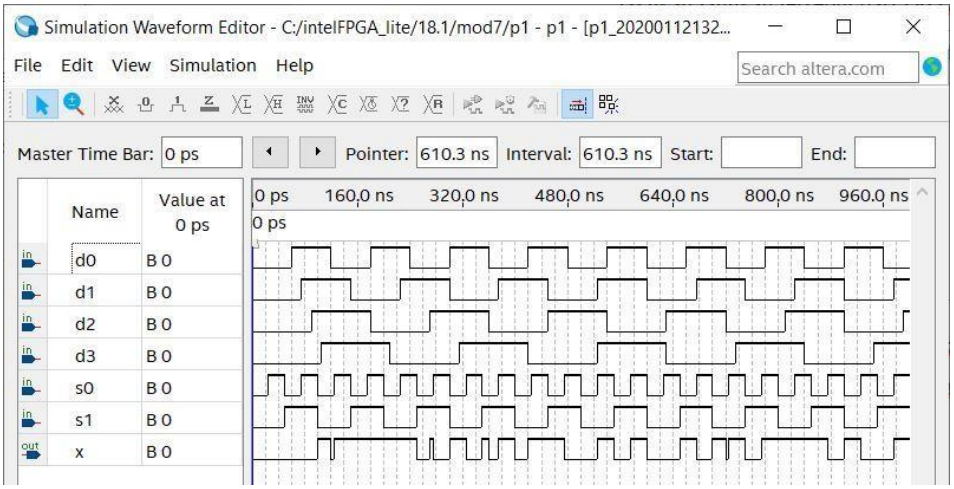
- Masukan D0 : clock periode 120ms dan duty cycle 50%
- Masukan D1 : clock periode 150ms dan duty cycle 50%
- Masukan D2 : clock periode 180ms dan duty cycle 50%
- Masukan D3 : clock periode 210ms dan duty cycle 50%
- Masukan S0 : clock periode 50ms dan duty cycle 50%
- Masukan S1 : clock periode 100ms dan duty cycle 50%

Dengan **cara klik kiri pada D0** → klik menu **Edit** → **Value** → **Overwrite Clock** → **masukan clock periode sesuai dengan masukan tersebut.** lakukan langkah tersebut dari D0 sampai S1.

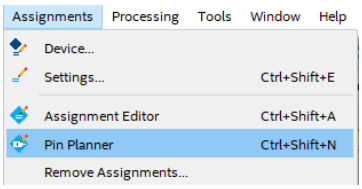




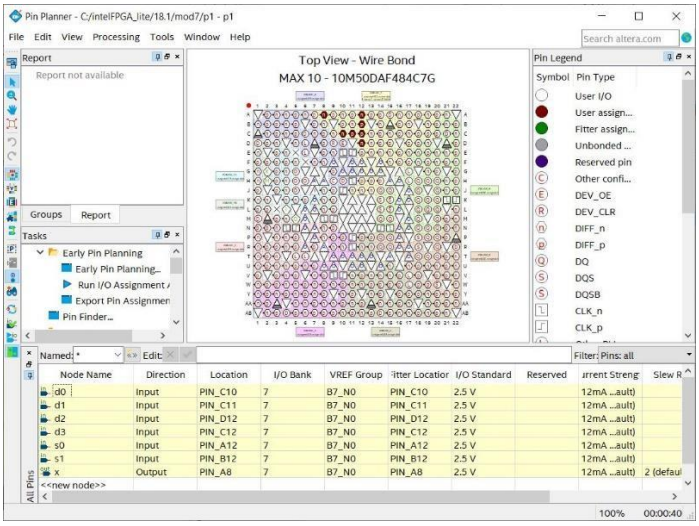
13. Lakukan simulasi dengan cara klik pada menu **Simulation → Run Timing Simulation →** tunggu hingga selesai.



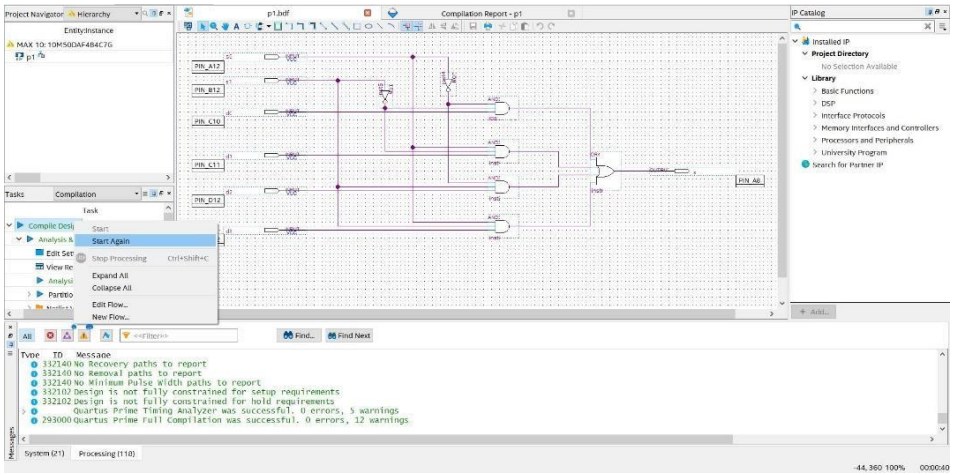
14. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu **assignments → pin planner**



15. Berikan **location** pin **assignment** pada fisik **DE10-LITE** sesuai dengan datasheet → **Run I/O assignment analysis.**

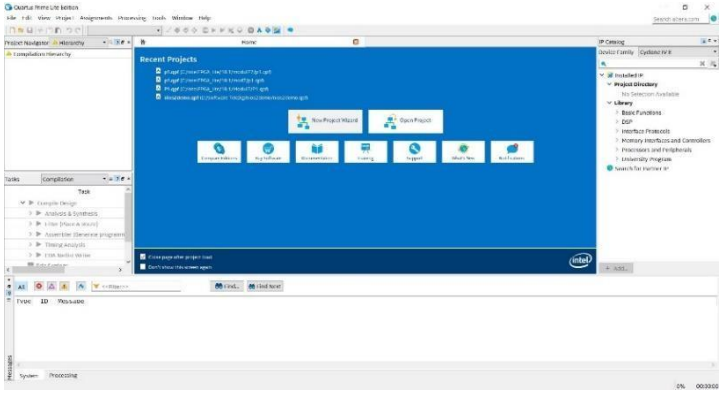


16. **Compile design**. Tunggu hingga selesai

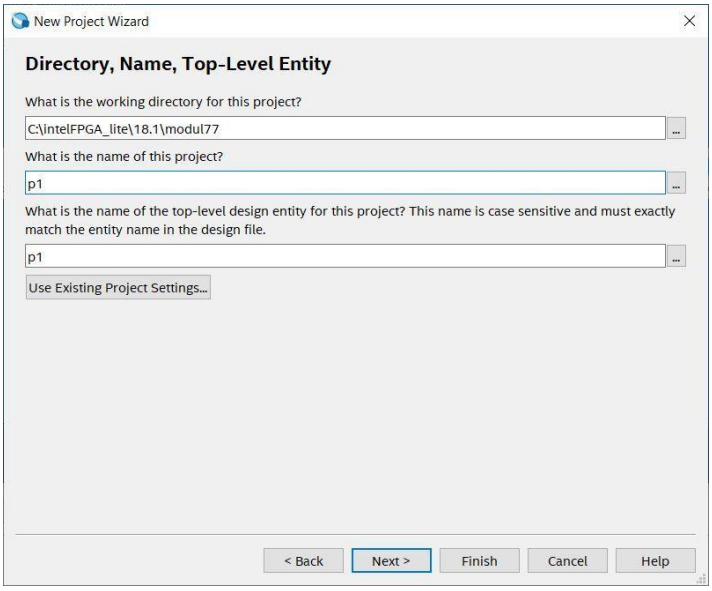


7.3.3 Langkah percobaan modul 7 (1 to 4 demultiplexer)

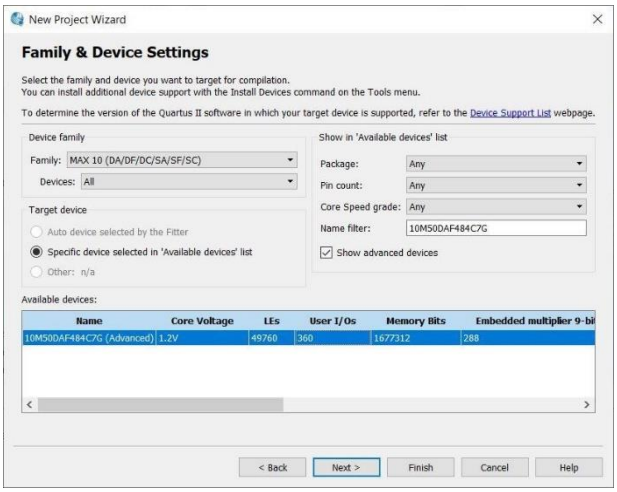
1. Hidupkan Laptop
2. Buka aplikasi quartus
3. Pilih **New Project Wizard → next**



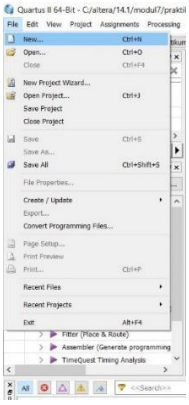
4. Menentukan **directory** dan nama **project → next → next → next**

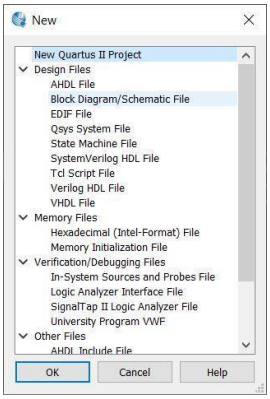


5. Menentukan **device family** dengan **MAX 10 (DA/DF/DC/SA/SC)** dan name filter dengan **10M50DAF484C7G → klik 10M50DAF484C7G** pada **available device → next → next → finish**

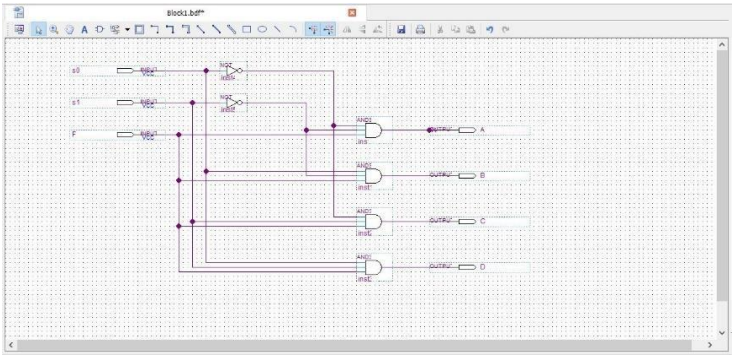


6. Pilih menu **file → new →** pilih **Block Diagram/Schematic File → ok**



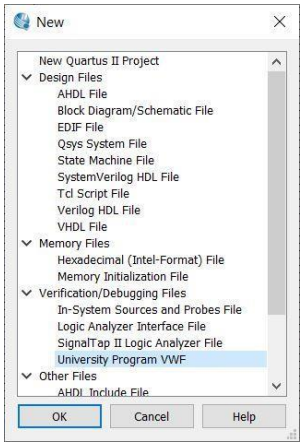


7. Membuat rangkaian multiplexer, klik **Symbol Tool → klik tanda panah → primitives → logic → pilih gerbang logika yang ada**. Dan buat inputan serta outputan dari rangkaian tersebut, klik **Pin Tool → pilih input serta output sesuaikan dengan rangkaian tersebut**, serta rename pin input dan output dan wiring sesuai dengan rangkaian.

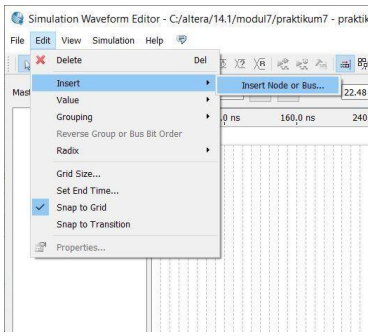


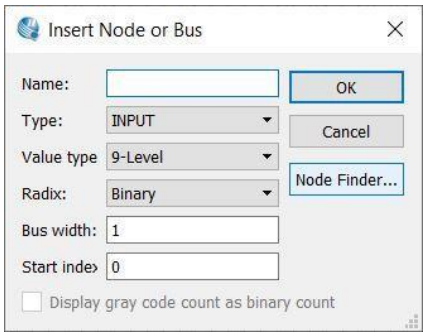
8. Setelah itu, klik **Compile design**. Tunggu hingga berhasil

9. Pilih menu **file → new →** pilih **Univertity Program VWF→ ok**

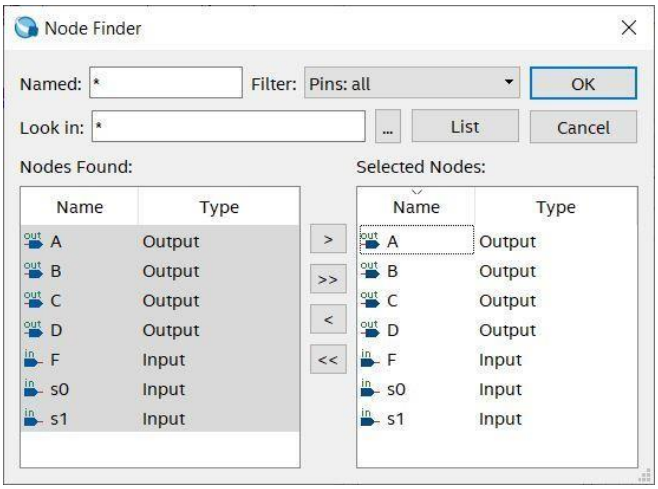


10. Pilih menu **Edit → Insert → Insert Node or Bus →** Pilih **Node Finder**



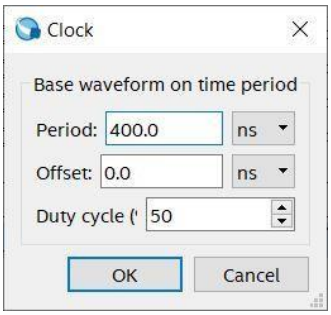


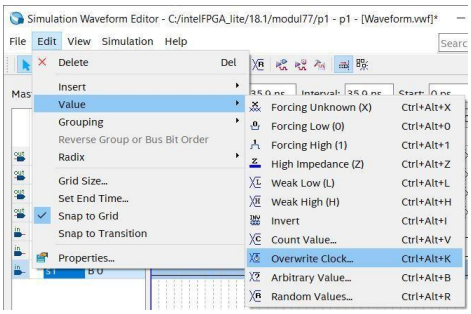
11. Lalu klik **List →** klik ">>" **→ ok → ok**



12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :

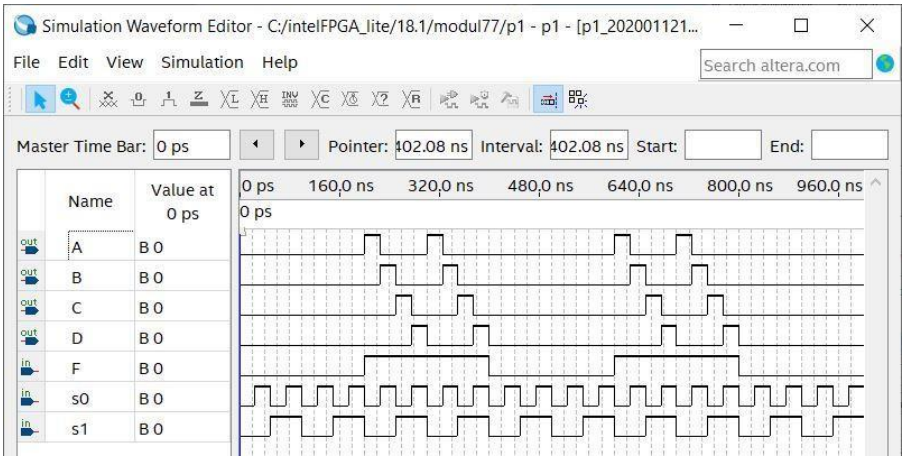
- Masukan S0 : clock periode 50ms dan duty cycle 50%
- Masukan S1 : clock periode 100ms dan duty cycle 50%
- Masukan F : clock periode 400ms dan duty cycle 50%



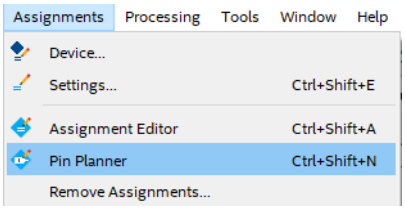


Dengan cara **klik kiri pada D0 →** klik menu **Edit → Value → Overwrite Clock → masukan clok periode sesuai dengan masukan tersebut.** lakukan langkah tersebut dari D0 sampai S1.

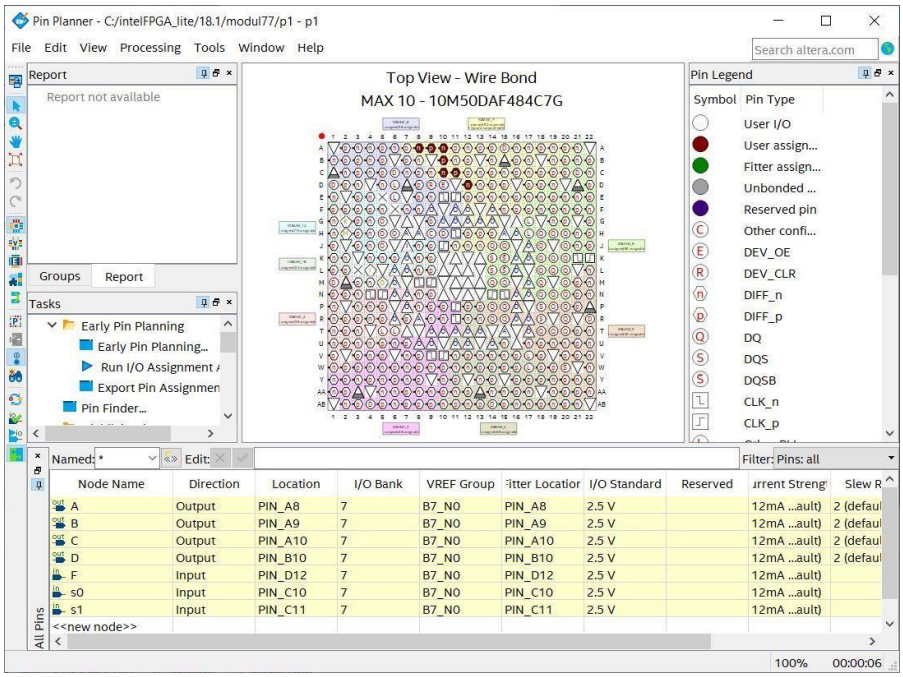
13. Lakukan simulasi dengan cara klik pada menu **Simulation → Run Timing Simulation →** tunggu hingga selesai.



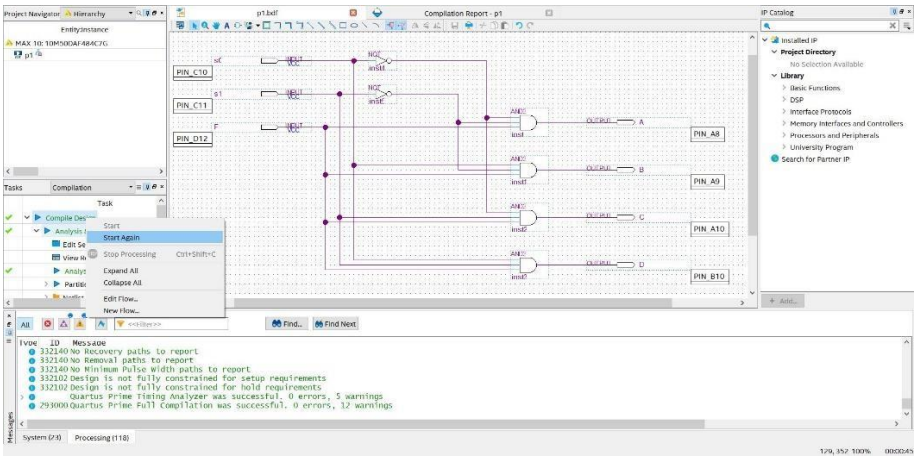
14. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu **assignments → pin planner**



15. Berikan **location** pin **assignment** pada fisik **DE10-LITE** sesuai dengan datasheet → **Run I/O assignment analysis.**

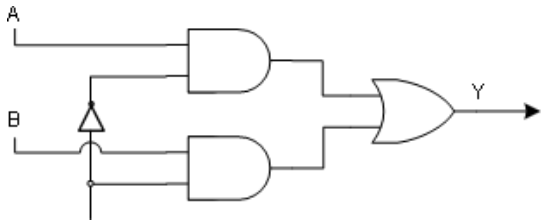


16. **Compile design**. Tunggu hingga selesai



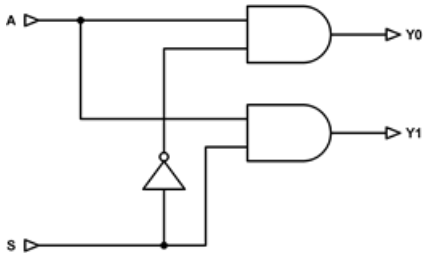
7.4 Soal Jurnal

1. Buatlah rangkaian berikut (multiplexer) pada buku jurnal kemudian analisa input serta outputnya.



A	***B***	***C***	***Y***
0	0	0	…
0	0	1	…
0	1	0	…
0	1	1	…
1	0	0	…
1	0	1	…
1	1	0	…
1	1	1	…

2. Buatlah rangkaian berikut (demultiplexer) pada buku jurnal kemudian analisa input serta outputnya.



A	***S***	***Y 0***	***Y1***
0	0	…	…
0	1	…	…
1	0	…	…
1	1	…	…

3. Tuliskan apa yang telah dilakukan pada praktikum modul 7 menggunakan Bahasa kalian sendiri!