

MODUL 2

PENGENALAN IC DIGITAL KELUARGA TTL DAN CMOS

2.1 Tujuan Praktikum Modul 2 :

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Praktikan dapat mengenal dan mengetahui Gerbang Logika, IC jenis TTL dan CMOS
2. dapat membuat IC bermuatan gerbang logika menggunakan software Quartus II

2.2 Dasar Teori Pendukung Modul 2

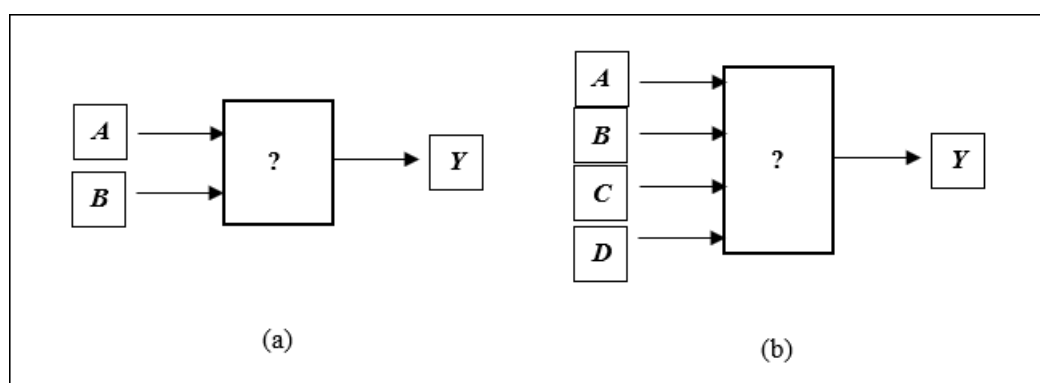
2.2.1 Gerbang Logika

Gerbang logika atau dalam Bahasa Inggris disebut dengan *Logic Gate* adalah dasar pembentuk Sistem Elektronika Digital yang berfungsi untuk mengubah satu atau beberapa Input (masukan) menjadi sebuah sinyal Output (keluaran) Logis. Gerbang Logika beroperasi berdasarkan system bilangan biner yaitu bilangan yang hanya memiliki 2 kode symbol yakni 0 dan 1 dengan menggunakan Teori Aljabar Boolean.

2.2.2 Tabel Kebenaran

Tabel Kebenaran terdiri dari urutan kemungkinan logika input dan logika output. Tabel kebenaran digunakan untuk menunjukkan bagaimana logika output pada rangkaian logika dipengaruhi oleh logika input pada rangkaian logika.

Gambar 2. 1 Bagan Sistem; (a) 2 masukan/input, (b) 4 masukan/input



Gambar diatas menunjukkan contoh bagan sistem dari rangkaian logika dimana pada gambar (a) menunjukkan 2 Inputan dan gambar (b) menunjukkan 4 Inputan. Kita dapat membuat contoh tabel kebenaran untuk masing-masing bagan rangkaian yang terdapat pada gambar (a) dan (b) dengan melihat pada tabel dibawah ini :

Tabel 2. 1 Contoh table kebenaran (a) 2 masukan (A dan B) dan 1 keluaran

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

Tabel 2. 2 Contoh tabel kebenaran gambar (b) 4 masukan (A,B,C) dan 1 keluaran (Y)

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Dari kedua tabel diatas diperlihatkan contoh tabel kebenaran untuk 2 masukan dan 4 masukan, kedua tabel diatas memperlihatkan keluaran Y dimana keluaran tersebut dipengaruhi oleh masukan dari A dan B yang mana protokol atau aturan yang menentukan keluaran tersebut berada pada gerbang logika yang digunakan.

Jenis-Jenis Gerbang Logika Dasar dan Simbolnya

Terdapat 7 jenis gerbang logika dasar yang membentuk sebuah sistem Elektronika digital, yaitu:

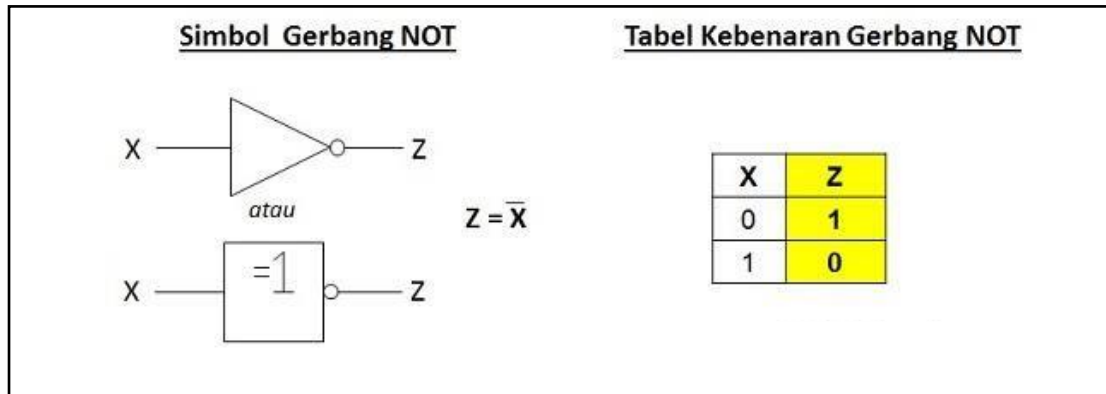
1. Gerbang NOT
2. Gerbang AND
3. Gerbang OR
4. Gerbang NAND
5. Gerbang NOR
6. Gerbang X-OR (Exclusive OR)
7. Gerbang X-NOR (Exclusive NOR)

Penjelasan Gerbang Logika dan Simbolnya :

1. Gerbang NOT

Gerbang NOT hanya memerlukan sebuah Masukan (Input) untuk menghasilkan hanya 1 Keluaran (Output). Gerbang NOT disebut juga dengan Inverter (Pembalik) karena menghasilkan Keluaran (Output) yang berlawanan (kebalikan) dengan Masukan atau Inputnya. Berarti jika kita ingin mendapatkan Keluaran (Output) dengan nilai Logika 0 maka Input atau Masukannya harus bernilai Logika 1. Gerbang NOT biasanya dilambangkan dengan simbol minus (“-“) di atas Variabel Inputnya.

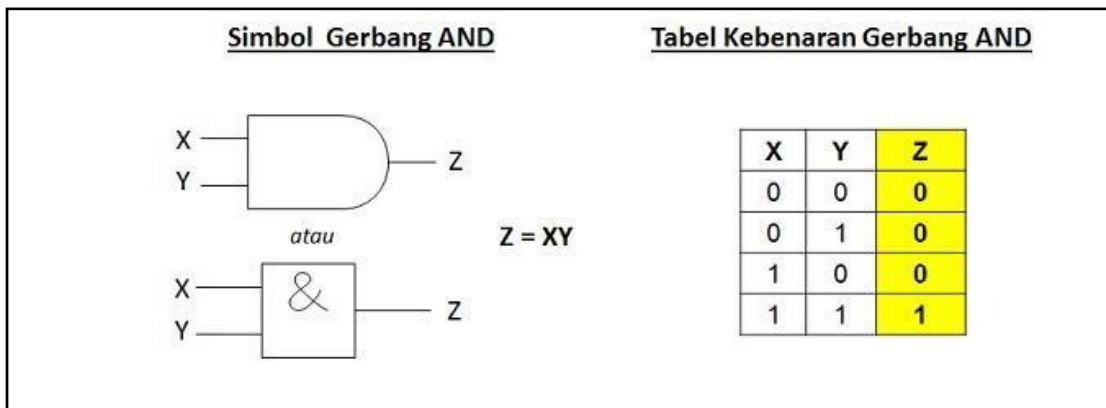
Gambar 2. 2 Simbol dan Tabel Kebenaran Gerbang NOT



2. Gerbang AND

Gerbang AND memerlukan 2 atau lebih Masukan (Input) untuk menghasilkan hanya 1 Keluaran (Output). Gerbang AND akan menghasilkan Keluaran (Output) Logika 1 jika semua masukan (Input) bernilai Logika 1 dan akan menghasilkan Keluaran (Output) Logika 0 jika salah satu dari masukan (Input) bernilai Logika 0. Simbol yang menandakan Operasi Gerbang Logika AND adalah tanda titik (“.”) atau tidak memakai tanda sama sekali. Contohnya : $Z = X.Y$ atau $Z = XY$.

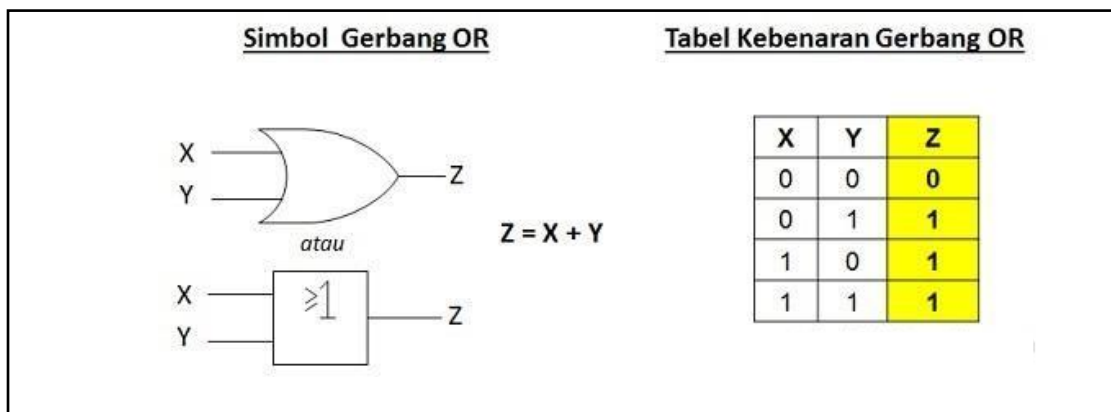
Gambar 2. 3 Simbol dan Tabel Kebenaran Gerbang AND



3. Gerbang OR

Gerbang OR memerlukan 2 atau lebih Masukan (Input) untuk menghasilkan hanya 1 Keluaran (Output). Gerbang OR akan menghasilkan Keluaran (Output) 1 jika salah satu dari Masukan (Input) bernilai Logika 1 dan jika ingin menghasilkan Keluaran (Output) Logika 0, maka semua Masukan (Input) harus bernilai Logika 0. Simbol yang menandakan Operasi Logika OR adalah tanda Plus (“+”). Contohnya : $Z = X + Y$.

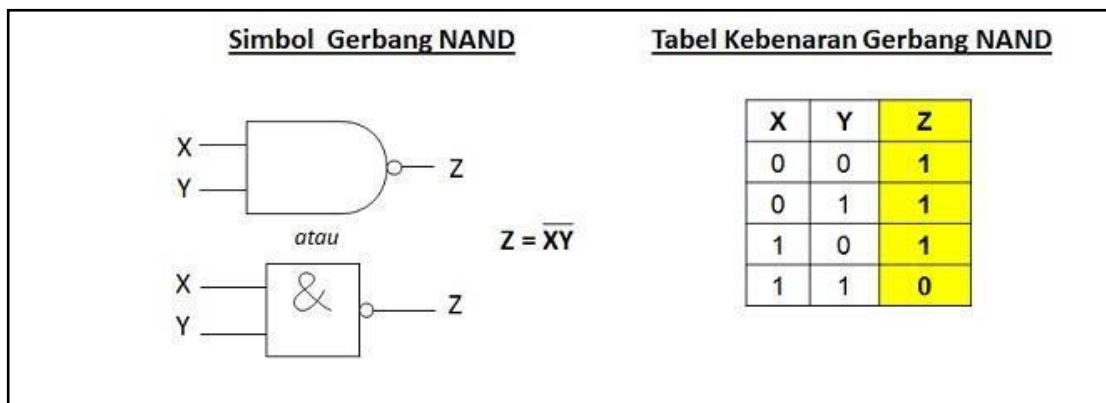
Gambar 2. 4 Simbol dan Tabel Kebenaran Gerbang OR



4. Gerbang NAND

Arti NAND adalah NOT AND atau BUKAN AND, Gerbang NAND merupakan kombinasi dari Gerbang AND dan Gerbang NOT yang menghasilkan kebalikan dari Keluaran (Output) Gerbang AND. Gerbang NAND akan menghasilkan Keluaran Logika 0 apabila semua Masukan (Input) pada Logika 1 dan jika terdapat sebuah Input yang bernilai Logika 0 maka akan menghasilkan Keluaran (Output) Logika 1.

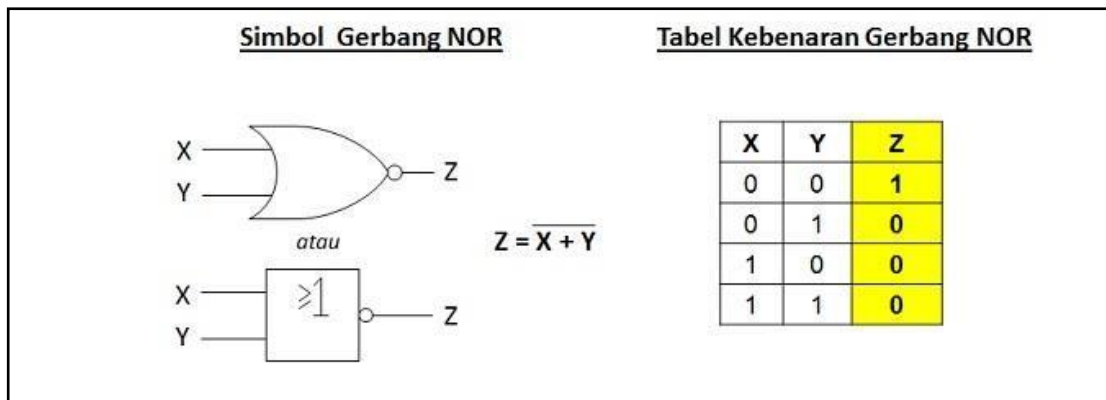
Gambar 2. 5 Simbol dan Tabel Kebenaran Gerbang Nand



5. Gerbang NOR

Arti NOR adalah NOT OR atau BUKAN OR, Gerbang NOR merupakan kombinasi dari Gerbang OR dan Gerbang NOT yang menghasilkan kebalikan dari Keluaran (Output) Gerbang OR. Gerbang NOR akan menghasilkan Keluaran Logika 0 jika salah satu dari Masukan (Input) bernilai Logika 1 dan jika ingin mendapatkan Keluaran Logika 1, maka semua Masukan (Input) harus bernilai Logika 0.

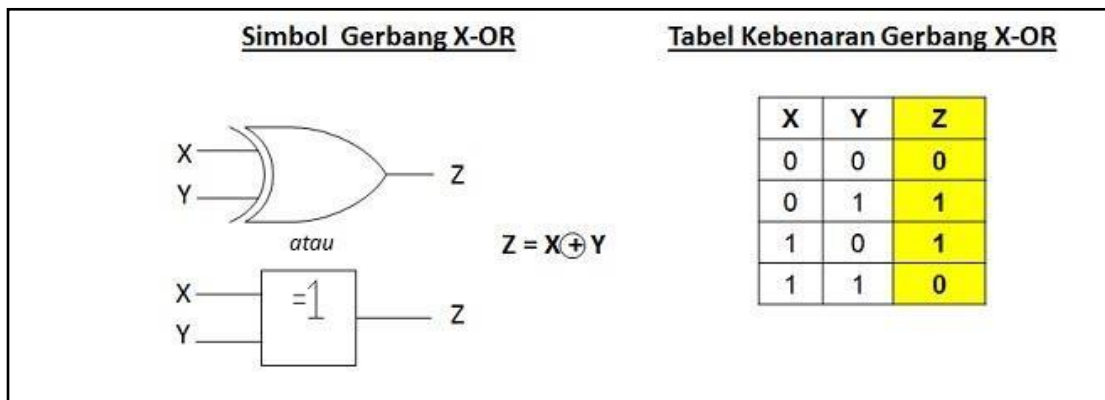
Gambar 2. 6 Simbol dan Tabel Kebenaran Gerbang NOR



6. Gerbang X-OR (Exclusive OR)

X-OR adalah singkatan dari Exclusive OR yang terdiri dari 2 Masukan (Input) dan 1 Keluaran (Output) Logika. Gerbang X-OR akan menghasilkan Keluaran (Output) Logika 1 jika semua Masukan-masukannya (Input) mempunyai nilai Logika yang berbeda. Jika nilai Logika Inputnya sama, maka akan memberikan hasil Keluaran Logika 0.

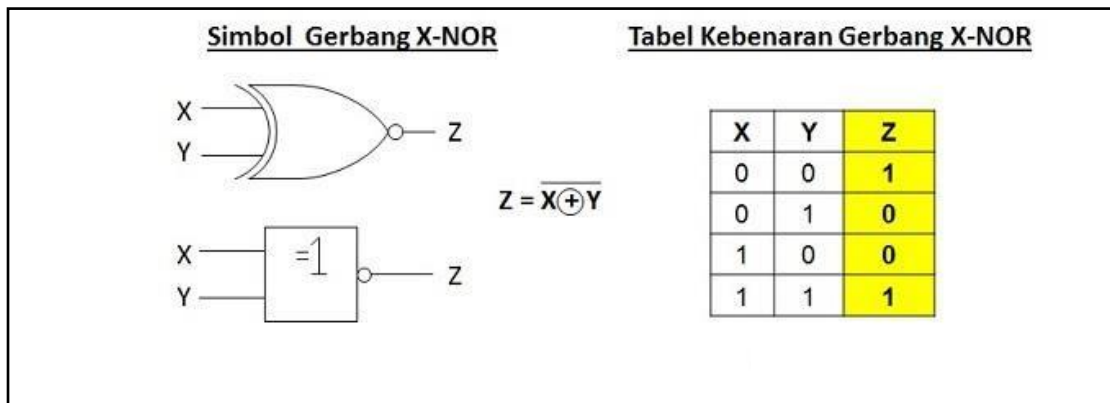
Gambar 2. 7 Simbol dan Tabel Kebenaran Gerbang X-OR



7. Gerbang X-NOR (Exclusive NOR)

Seperti Gerbang X-OR, Gerbang X-NOR juga terdiri dari 2 Masukan (Input) dan 1 Keluaran (Output). X-NOR adalah singkatan dari Exclusive NOR dan merupakan kombinasi dari Gerbang X-OR dan Gerbang NOT. Gerbang X-NOR akan menghasilkan Keluaran (Output) Logika 1 jika semua Masukan atau Inputnya bernilai Logika yang sama dan akan menghasilkan Keluaran (Output) Logika 0 jika semua Masukan atau Inputnya bernilai Logika yang berbeda. Hal ini merupakan kebalikan dari Gerbang X-OR (Exclusive OR).

Gambar 2. 8 Simbol dan Tabel Kebenaran Gerbang X-NOR



2.2.3 IC (Integrated Circuit)

Integrated Circuit atau disingkat dengan IC adalah Komponen Elektronika Aktif yang terdiri dari gabungan ratusan, ribuan bahkan jutaan Transistor, Dioda, Resistor dan Kapasitor yang diintegrasikan menjadi suatu Rangkaian Elektronika dalam sebuah kemasan kecil. Bahan utama yang membentuk sebuah Integrated Circuit (IC) adalah Bahan Semikonduktor. Silicon merupakan bahan semikonduktor yang paling sering digunakan dalam Teknologi Fabrikasi Integrated Circuit (IC). Dalam bahasa Indonesia, Integrated Circuit atau IC ini sering diterjemahkan menjadi Sirkuit Terpadu.

Terdapat banyak sekali jenis IC, namun yang akan dibahas pada praktikum kali ini adalah jenis-jenis IC yang memuat gerbang logika. Jenis IC yang memuat gerbang logika terbagi menjadi 2 jenis, yaitu :

- IC TTL (*Integrated Circuit Transistor-Transistor Logic*)
- IC CMOS (*Integrated Circuit Complementary Metal Oxide Semiconductor*)

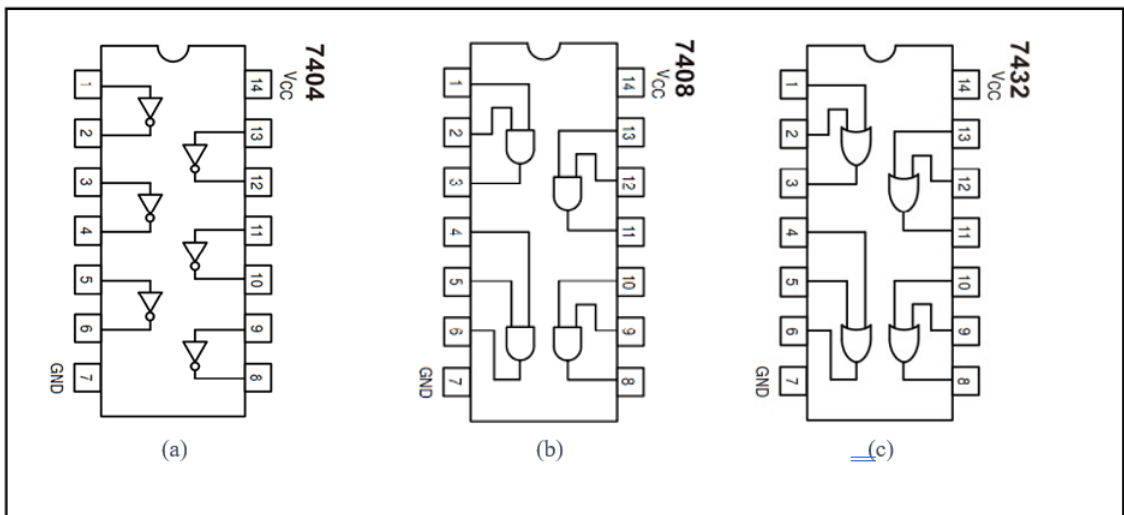
Penjelasan Singkat IC TTL & IC CMOS

a. IC TTL

IC-TTL dibangun dengan menggunakan transistor sebagai komponen utamanya dan fungsinya dipergunakan untuk berbagai variasi Logic, sehingga dinamakan Transistor-Transistor Logic. Dalam satu kemasan IC terdapat beberapa macam gate (gerbang) yang dapat melakukan berbagai macam fungsi logic seperti AND, NAND, OR, NOR, XOR serta beberapa fungsi logic lainnya seperti Decoder, Encoder, Multiflexer dan Memory sehingga pin (kaki) IC jumlahnya banyak dan bervariasi ada yang 8, 14, 16, 24 dan 40.

- Sumber tegangan 4,75 – 5,25 V
- Ditandai dengan kode 73 (seri 74XX, 741XX, 742XX)
- Dapat diaplikasikan sebagai saklar ON/OFF

Gambar 2.9 (a) IC TTL 7404, (b) IC TTL 7408, (c) IC TTL 7432



Tabel 2. 3 Daftar IC TTL gerbang logika

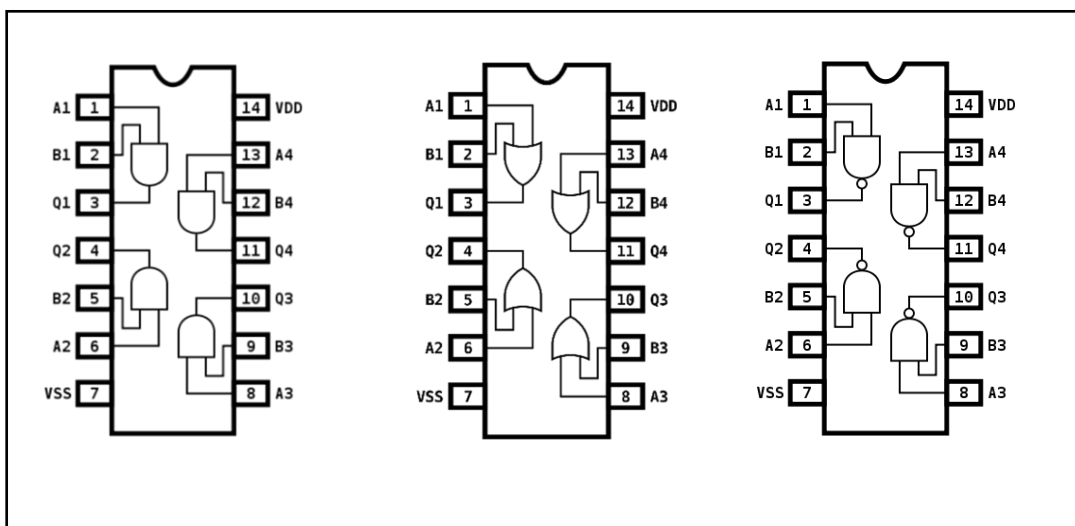
Tipe IC	Gerbang Logika yang dimuat
7400	Quad 2-input NAND Gate
7401	Quad 2-input NAND Gate (Open Collector)
7402	Quad 2-input NOR Gate
7403	Quad 2-input NAND Gate
7404	Hex Inverter
7405	Hex Inverter (Open Collector)
7406	Hex Inverter Buffer/Driver (Open Collector)
7407	Hex Buffer/Driver (Open Collector)
7408	Quad 2-input AND Gate
7409	Quad 2-input AND Gate (Open Collector)
7410	Triple 3-input NAND Gate
7411	Triple 3-input AND Gate
7414	Hex Inverter Schmitt Trigger
7420	Dual 4-input NAND Gate
7421	Dual 4-input AND Gate
7427	Triple 3-input NOR Gate
7430	8-input NAND Gate
7432	Quad 2-input OR Gate
7437	Quad 2-input NAND Buffer
7438	Quad 2-input NAND Buffer (Open Collector)
7486	Quad Exclusive-OR Gate
74132	Quad 2-input NAND Schmitt Trigger
74136	Quad Exclusive-OR Gate (Open Collector)
74260	Dual 5-input NOR Gate
74266	Quad Exclusive-NOR Gate
74365	Hex Buffer with Logical OR Tri-State
74368	Hex Inverter Tri-State

b. IC CMOS

Selain TTL, jenis IC digital lainnya adalah C-MOS (Complementary with MOSFET) yang berisi rangkaian yang merupakan gabungan dari beberapa komponen MOSFET untuk membentuk gate-gate dengan fungsi logic seperti halnya IC-TTL. Dalam satu kemasan IC C-MOS dapat berisi beberapa macam gate(gerbang) yang dapat melakukan berbagai macam fungsi logic seperti AND,NAND,OR,NOR,XOR serta beberapa fungsi logic lainnya seperti Decoders, Encoders, Multiplexer dan Memory.

- Sumber tegangan 3 – 18 V
- Merupakan gabungan dari komponen MOSFET untuk membuat gerbang dengan fungsi logika
- Ditandai dengan kode 40 (seri 40XX)

Gambar 2. 10 (a) IC CMOS 4081, (b) IC CMOS 4071, (c) IC CMOS 4011



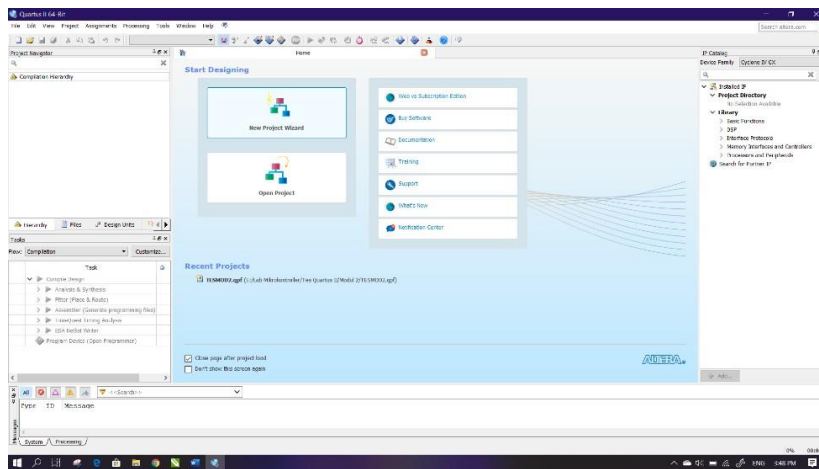
2.3 Lembar Kegiatan Praktikum Modul 2 :

2.3.1 Alat dan Bahan

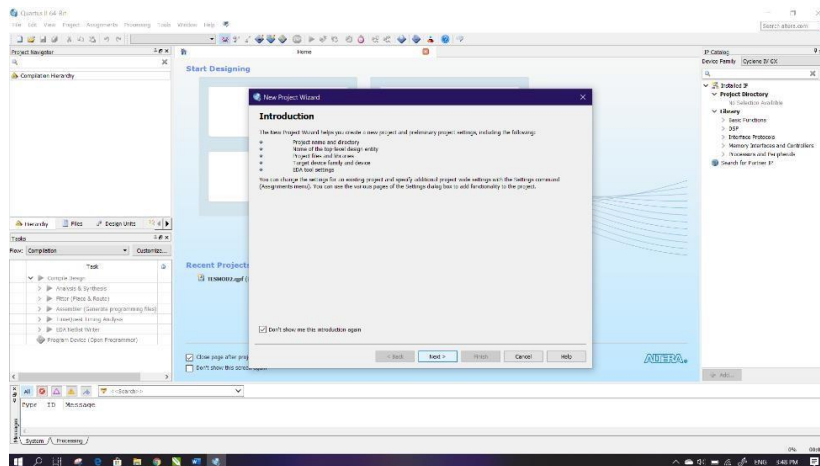
1. Software Quartus II
2. Laptop
3. Mouse

2.3.2 Langkah Praktikum

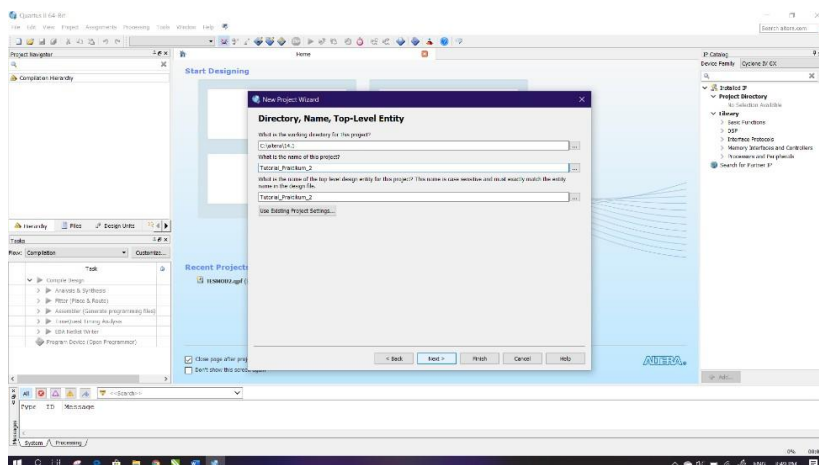
1. Buka software Quartus II di laptop dan klik New Project Wizard



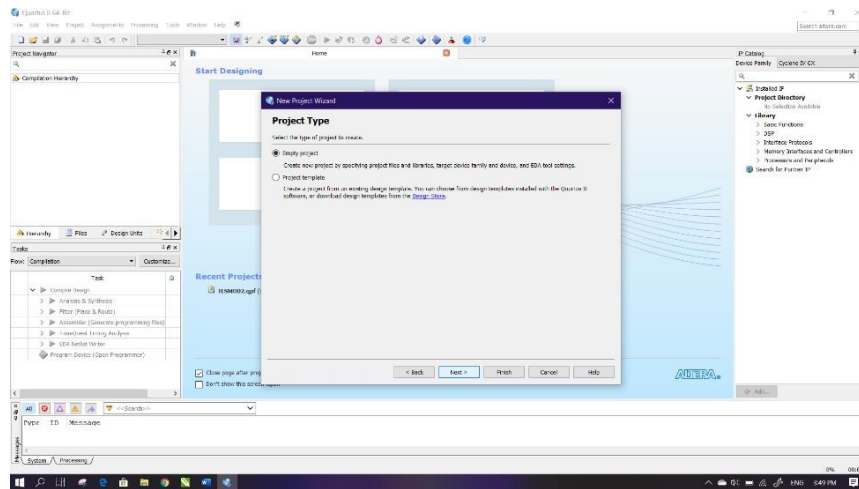
2. Klik Next



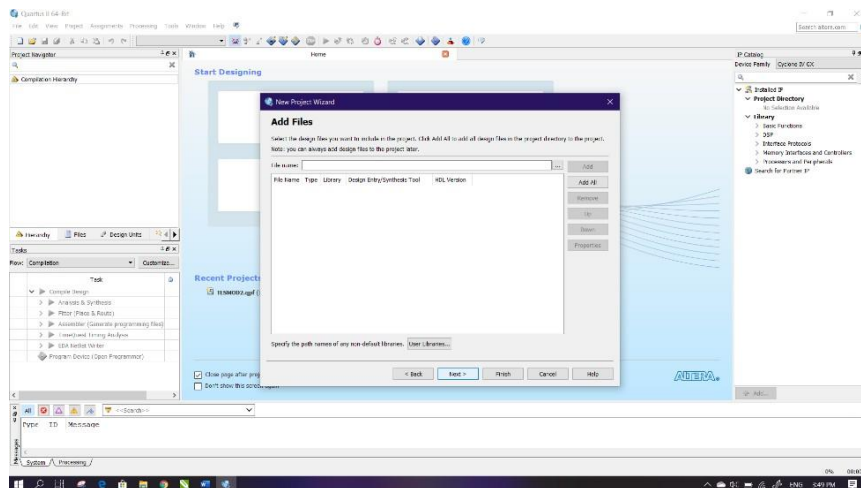
3. Kemudian tentukan Directory Project dan Nama Project, lalu klik Next



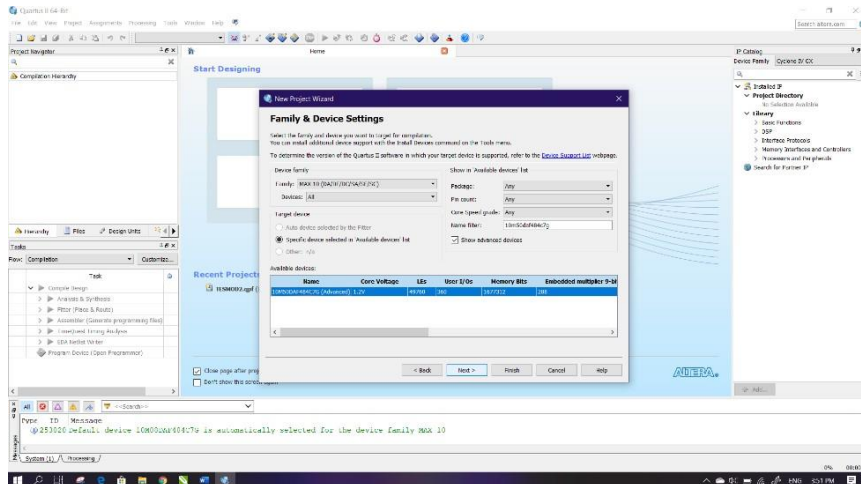
4. Pilih **Empty Project**, kemudian klik **Next** lagi



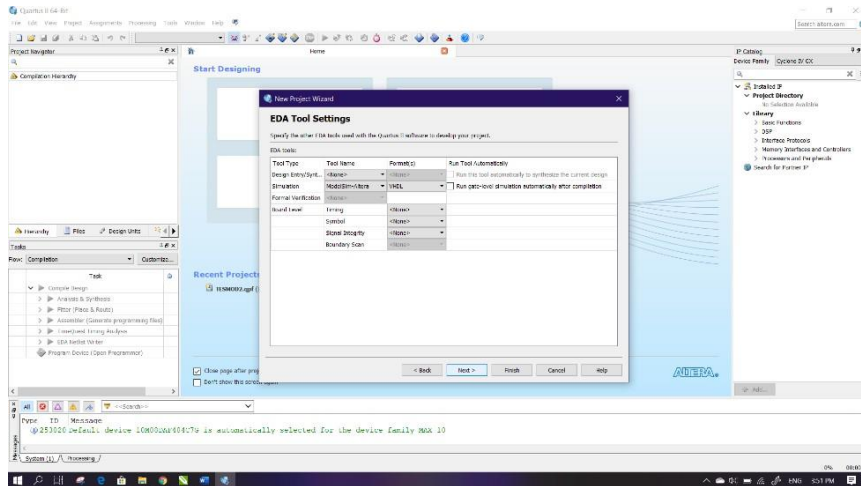
5. Setelah itu, klik **Next** lagi



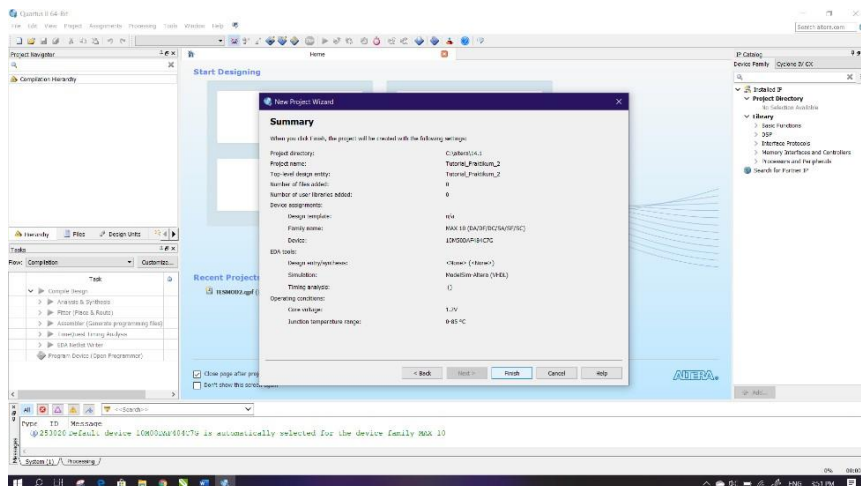
6. Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik **Available Device** yang tersedia, kemudian klik **Next**



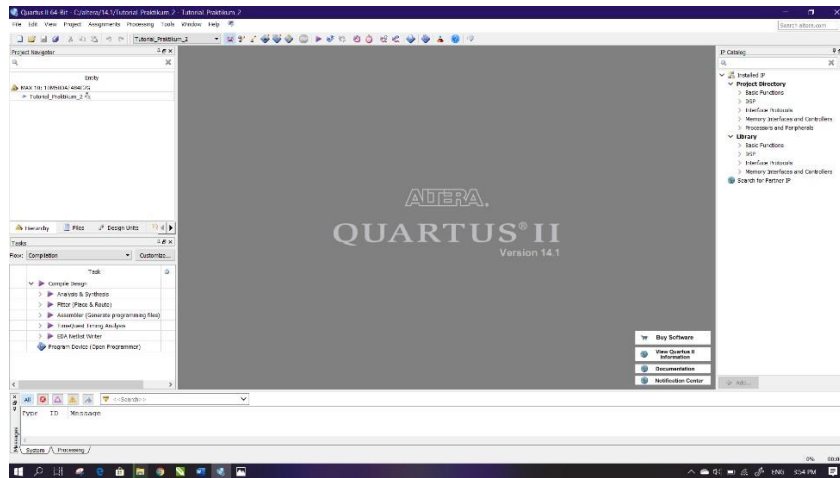
7. Kemudian klik Next



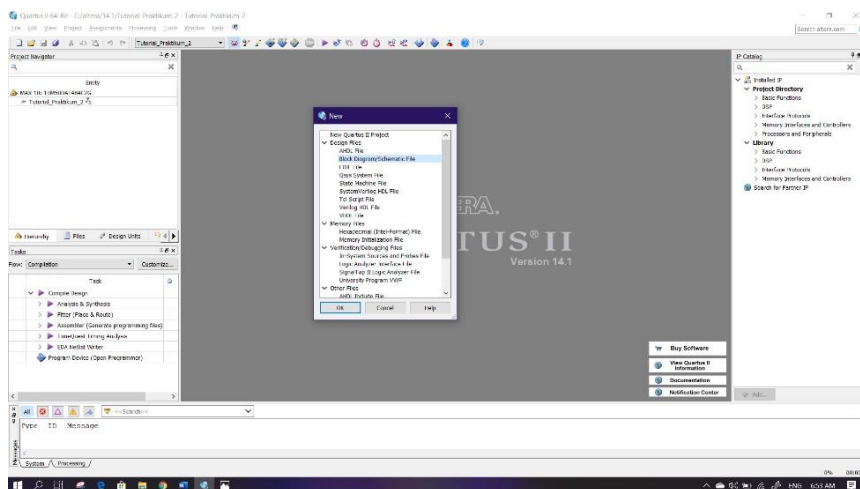
8. Lalu klik Finish



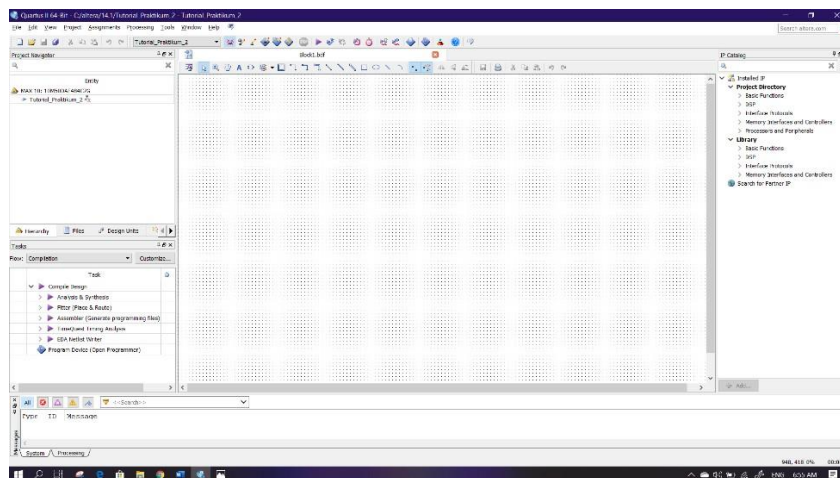
9. Setelah itu akan muncul tampilan awal dari Project Quartus II seperti gambar dibawah ini



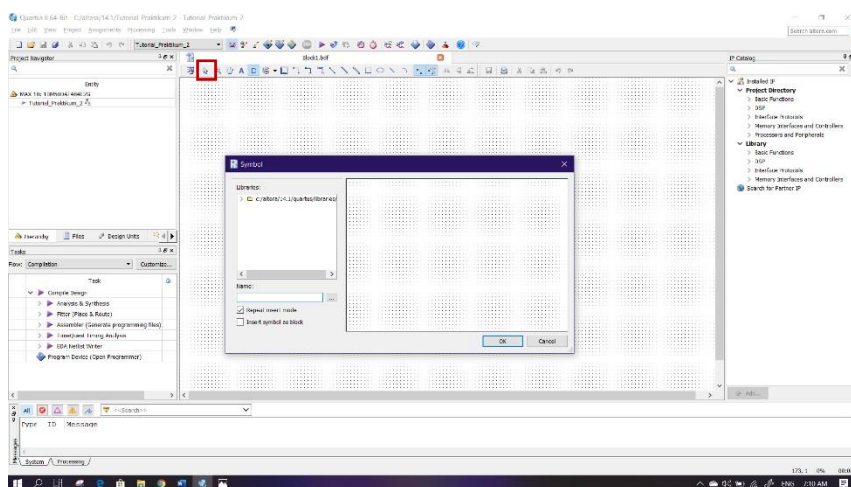
10. Kemudian buat file baru dengan cara klik **File** → **New**, lalu pilih **Block Diagram/Schematic File**, kemudian klik **OK**



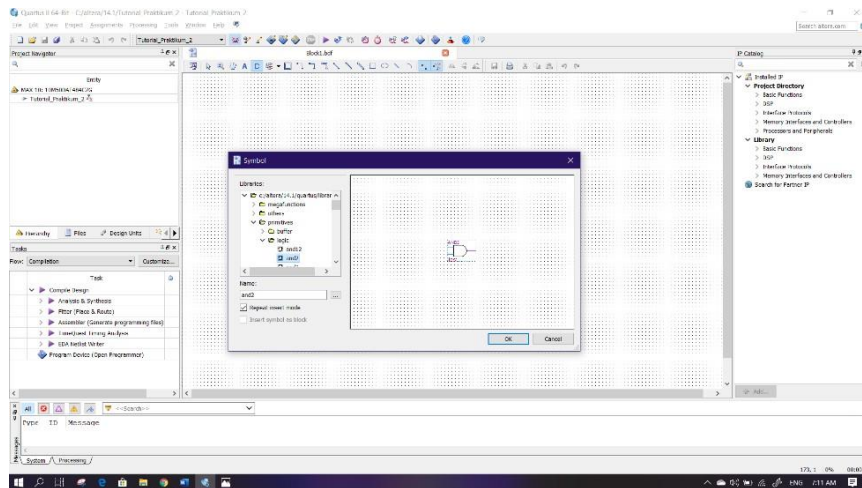
11. Akan muncul workspace seperti gambar dibawah ini



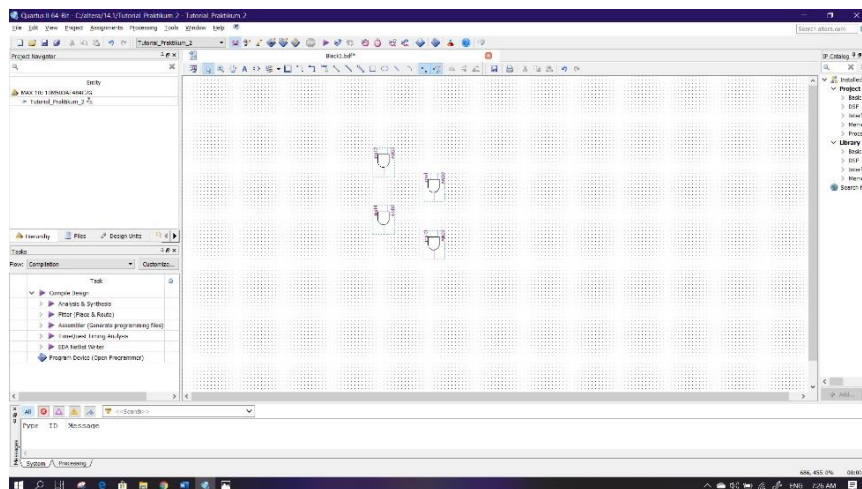
12. Lalu kita masuk pada pembuatan IC, sebelumnya tentukan IC jenis apa yang akan dibuat. Pada langkah praktikum kali ini IC yang akan dibuat adalah IC TTL 7408. Klik tool **Symbol Tool** seperti gambar dibawah (di mark merah)



13. Kemudian pada Libraries, klik folder **c:/altera/14.1/quartus/libraries/ → primitives → logic**, lalu pilih gerbang logika dengan nama **and2** dan klik **OK**

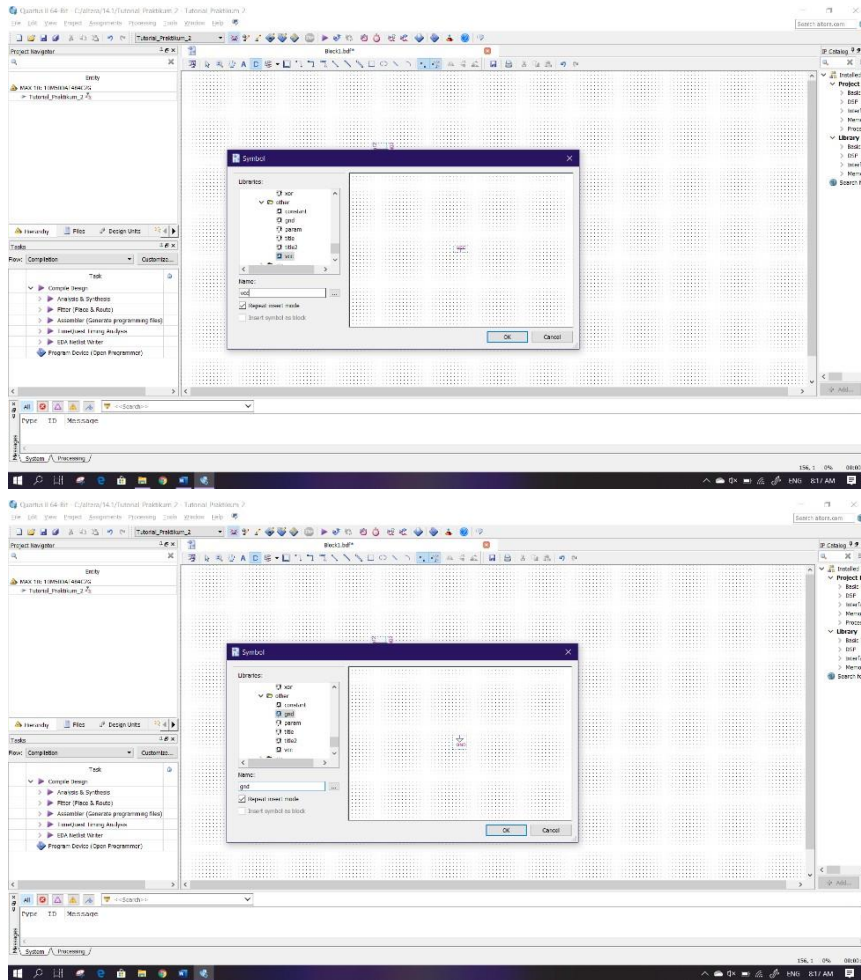


14. Lalu atur posisi dan jumlah gerbang logika tersebut seperti gambar dibawah ini, gunakan fungsi rotate untuk memutar posisi gerbang logika dengan cara klik **kanan** komponen (gerbang logika) → **Rotate by Degrees** → **Rotate Left 270**. Lalu untuk melepas komponen yang masih ada pada cursor tekan **ESC** pada keyboard

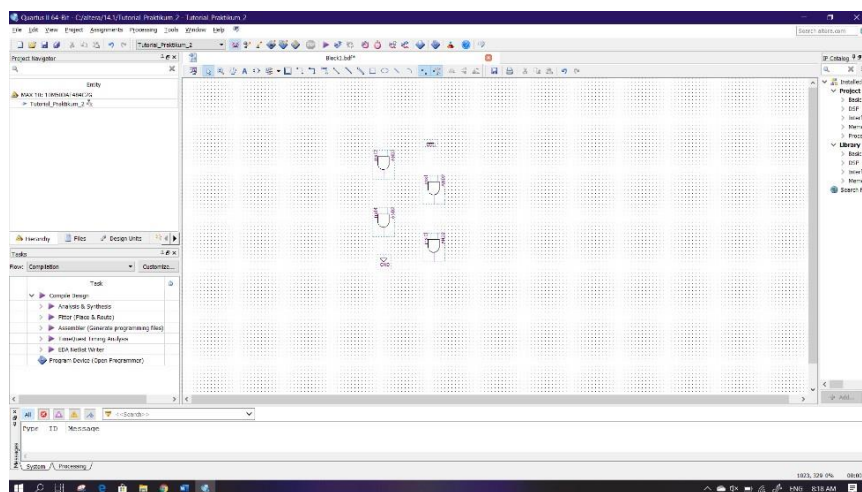


15. Tambahkan VCC dan GND (Ground) dengan cara klik tool **Symbol Tool**, lalu ketik pada **Name : vcc** dan letakkan pada workspace. Lakukan hal yang sama pada Ground klik tool **Symbol Tool**, lalu ketik pada **Name : gnd** kemudian letakkan juga pada workspace.

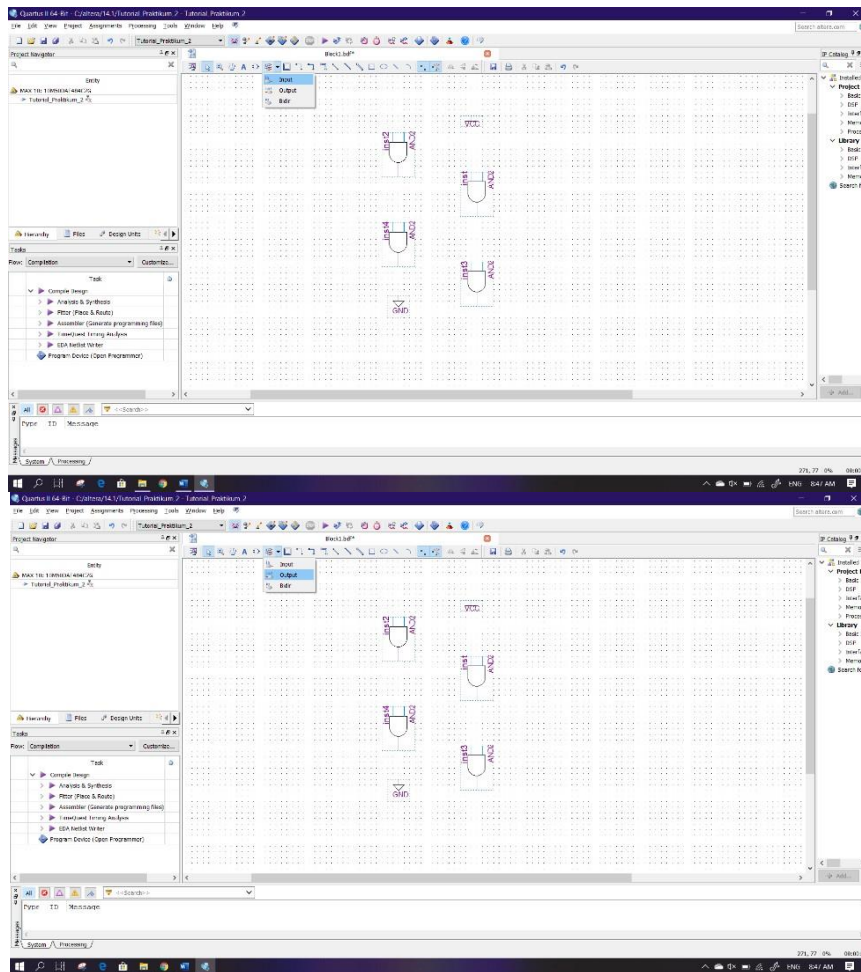
Modul Praktikum



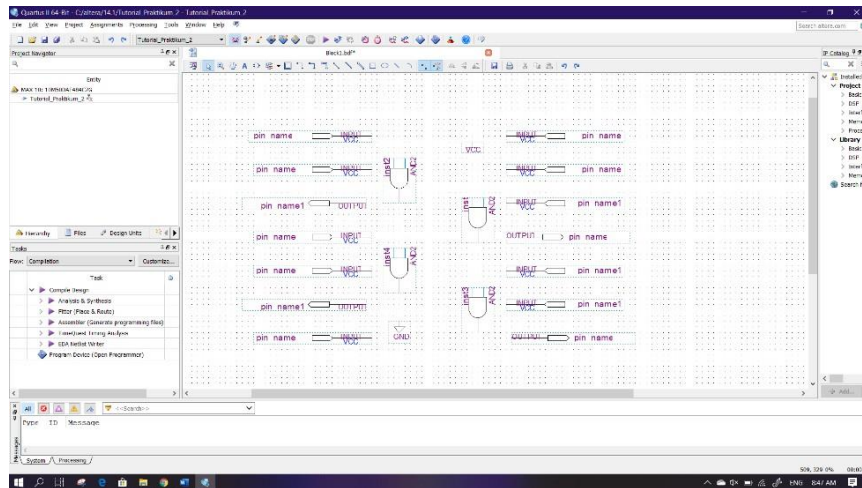
16. Sehingga komponen yang ada pada workspace seperti pada gambar dibawah ini



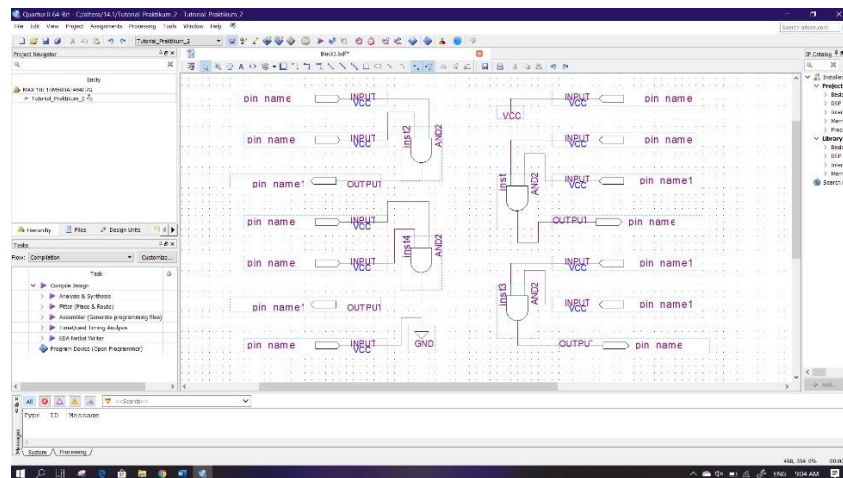
17. Kemudian buat pin input dan output dengan cara klik tool **Pin Tool** lalu pilih **Input** dan juga **Output**.



18. Posisikan pin input dan pin output seperti pada gambar dibawah ini, gunakan fungsi flip vertical/horizontal untuk membuat komponen berbalik posisi dengan cara **klik kanan** komponen lalu klik **Flip Vertical** atau bisa juga **Flip Horizontal**

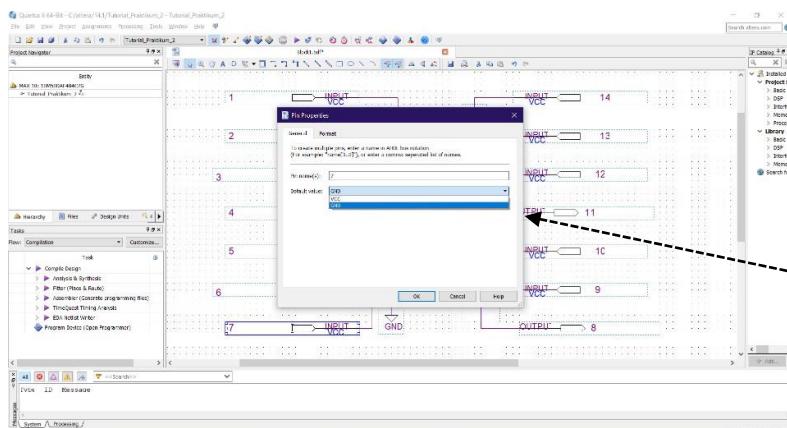


19. Kemudian lakukan wiring dengan cara klik tool **Orthogonal Node Tool** atau bisa juga secara langsung dengan klik dan tahan ujung input/output lalu sambungkan ke input gerbang logika, vcc, ground. Jika terdapat bulatan pada wiring, itu artinya wiring belum tersambung dengan benar, hapus wiring tersebut kemudian lakukan wiring ulang.

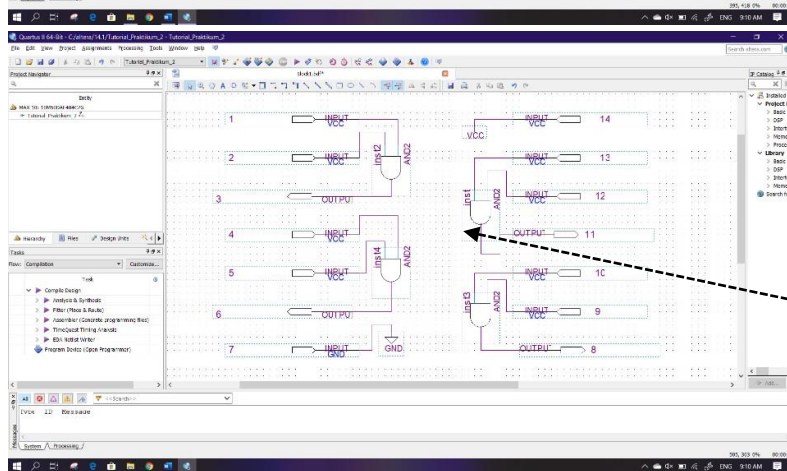


20. Ubah nama pin (pin name) dengan nomor sesuai dengan nomor kaki IC yang ada pada IC tersebut. Caranya **double click** pada tulisan pin name atau bisa juga dengan **klik kanan** tulisan pin name lalu klik **Properties**. Sebelum itu ganti **pin value** yang tersambung pada Ground menjadi **GND**

Modul Praktikum

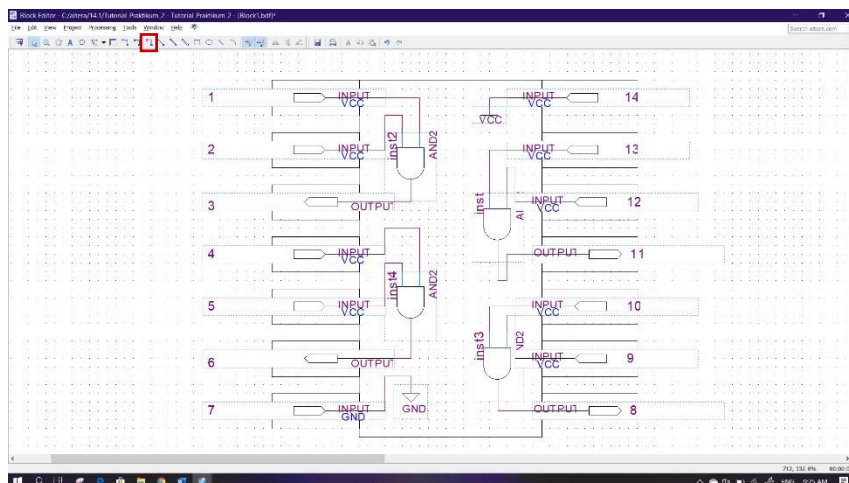


*Mengganti pin
value Ground*



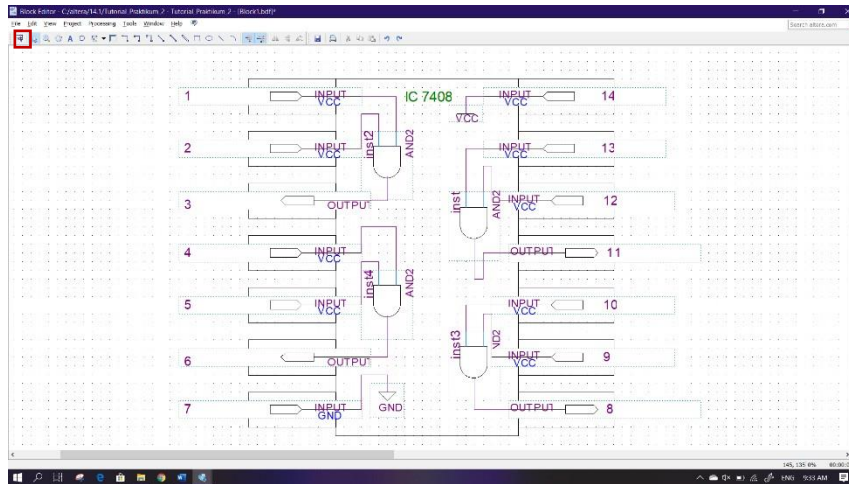
Hasil Wiring

21. Kemudian berikan bingkai atau frame dengan menggunakan tool **Rectangle Tool** lalu rapikan sesuai dengan gambar dibawah

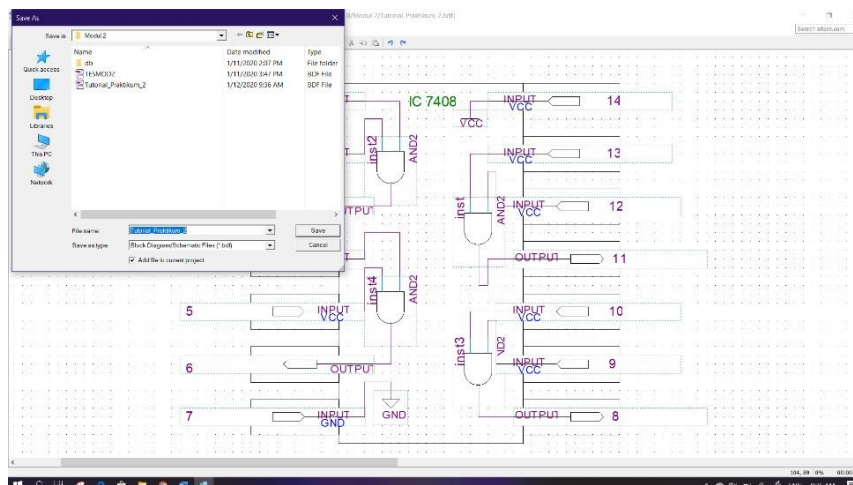


22. Berikan label atau nama dengan menggunakan **Text Tool**

Modul Praktikum

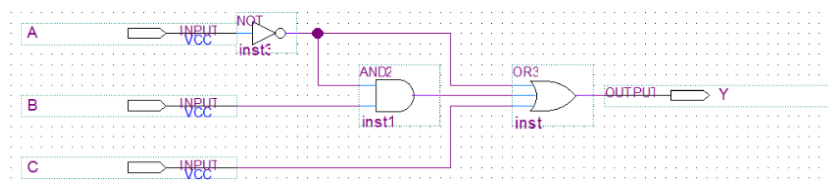


23. Terakhir Save dengan cara menekan **Ctrl+S** atau bias dengan cara klik **File** → **Save**. Lalu tentukan directory folder untuk menyimpan file tersebut



2.4 Soal Jurnal

1. Buatlah IC CMOS 4071 dengan menggunakan blok diagram menggunakan software Quartus II
2. Tuliskan tabel kebenaran dari gerbang logika berikut !



3. Tuliskan apa yang telah dilakukan pada praktikum modul 2 menggunakan Bahasa kalian sendiri!

MODUL 3

SIMULASI DIGITAL MENGGUNAKAN MODELSIM

3.1 Tujuan Praktikum Modul 3

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Praktikan dapat memahami tentang Modelsim
2. Praktikan memahami dan mengerti cara kerja Modelsim
3. Praktikan memahami proses-proses penggunaan Modelsim

3.2 Dasar Teori Praktikum Modul 3

3.2.1 ModelSim

ModelSim adalah *software* simulator multi Bahasa HDL yang dibuat oleh *Mentor Graphics*. ModelSim digunakan untuk simulasi Bahasa deskripsi perangkat keras seperti VHDL, Verilog, dan SystemC. ModelSim dapat digunakan secara *independent* atau bersamaan dengan Intel Quartus Prime, Xilinx, ISE, atau Xilinx Vivado. Simulasi dilakukan menggunakan *Interface* grafis (GUI) atau secara otomatis menggunakan skrip.

Peran ModelSim pada praktikum kali ini untuk mensimulasikan rangkaian yang telah dirangkai pada *Software Quartus Altera* dan mengetahui fungsi dan hasil dari nilai digital pada gerbang logika.

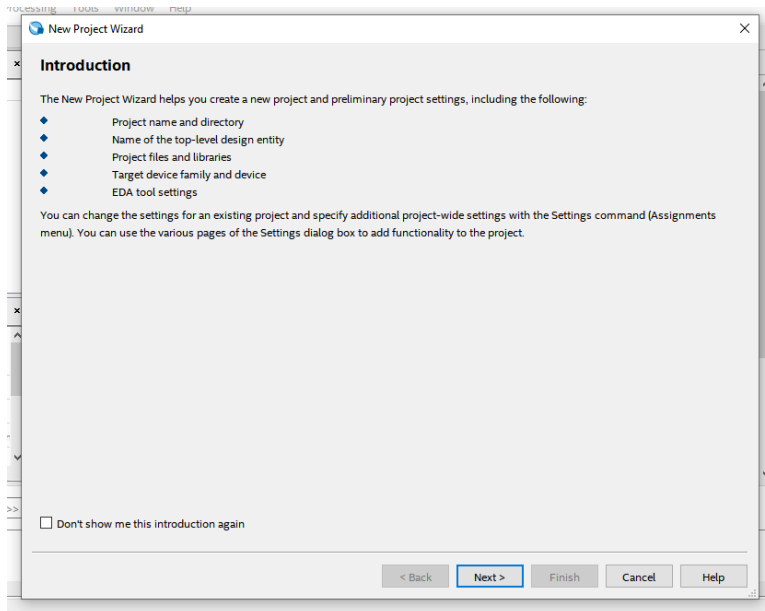
3.3 Lembar Kegiatan Praktikum Modul 3

3.3.1 Alat dan Bahan

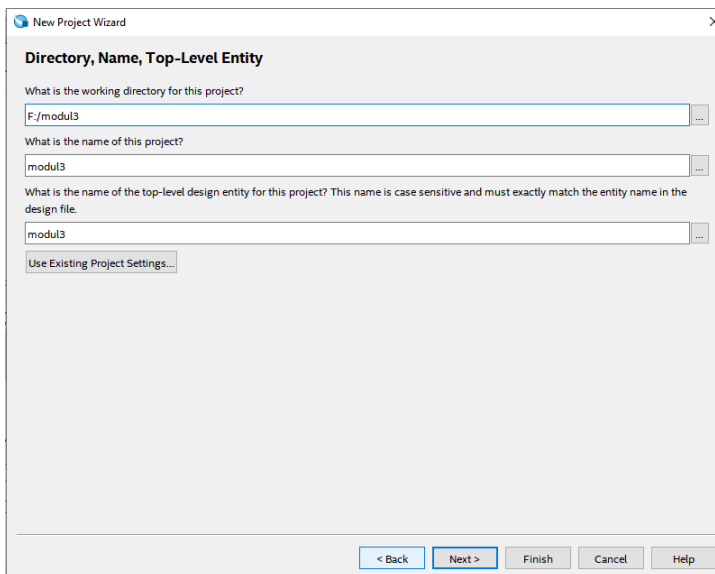
- a. Laptop yang telah ter-*install software Quartus Altera* dan *ModelSim*
- b. Mouse

3.3.2 Langkah Praktikum Modul 3

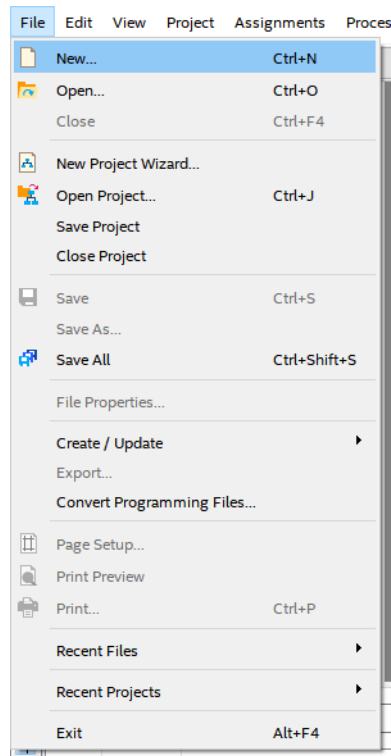
1. Buka *Software Quartus* lalu klik *Create a new project*
2. Klik *Next*



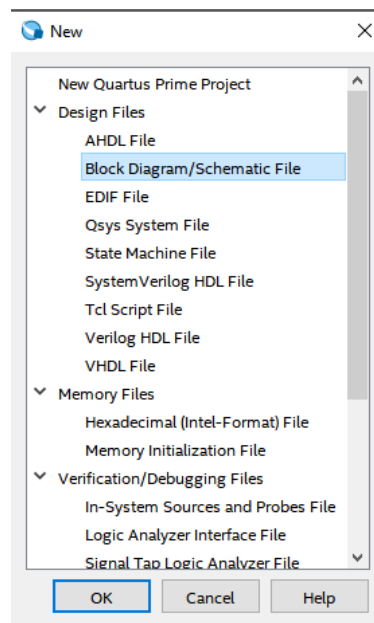
3. Lalu, buat nama *project* yang ingin



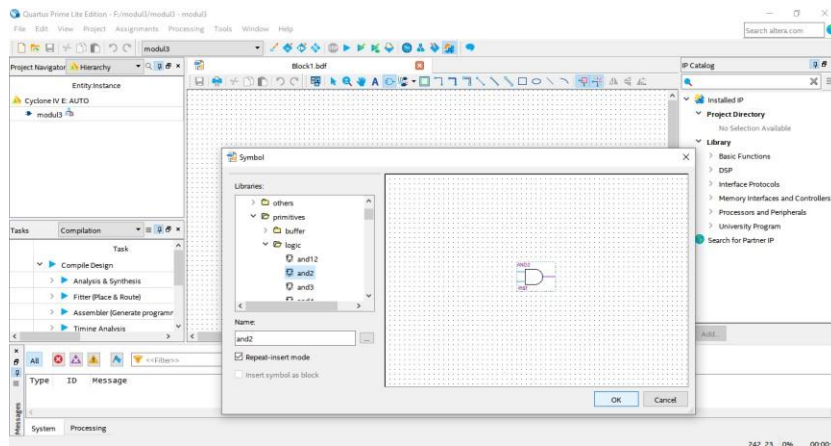
4. Lalu buat *file* baru dengan klik *File* → *New* (Ctrl+N)



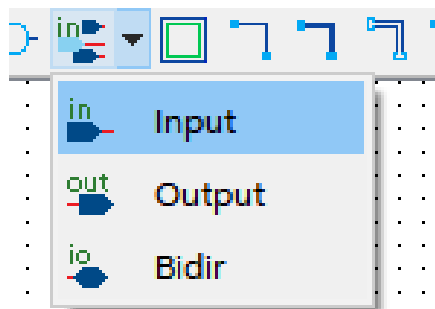
5. Pilih Blok Diagram/Schematic File → Klik OK



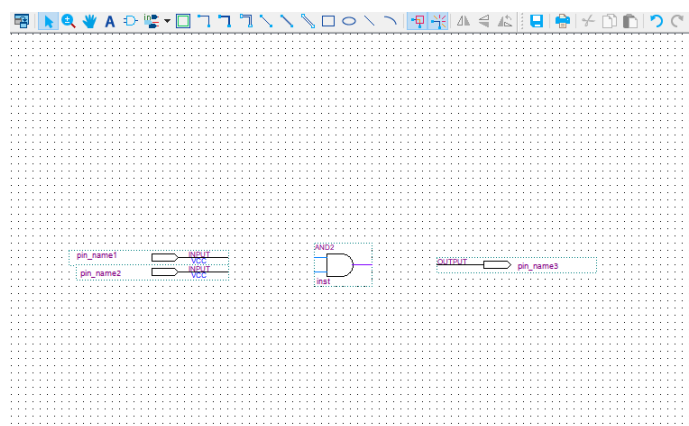
- Buat blok gerbang logika AND klik tools tombol → buka folder `c:/altera/13.0/quartus/libraries/` → buka folder primitives → buka folder logic → pilih and2 → klik OK → klik next



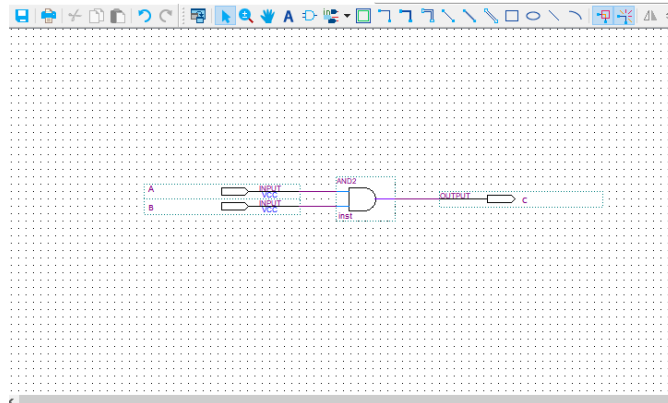
- Pilih pin tools untuk menambahkan 2 input dan 1 output



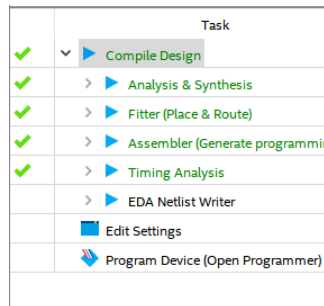
- Pilih diagonal node tool untuk membuat wire antara input, output dan gerbang logika agar terhubung.



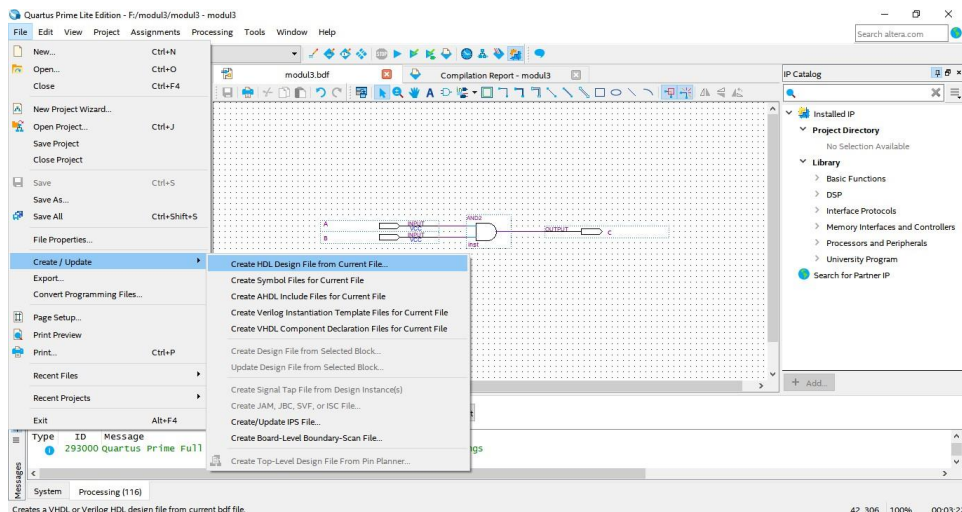
9. Beri nama pada input dan output dengan cara klik 2 kali pada bagian nama



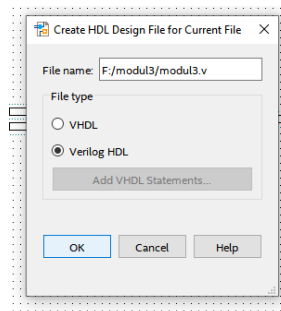
10. Compile hasil rangkaian dengan cara pilih compile design pada bagian task >> klik kanan → start >> yes → tunggu compiling hingga sukses



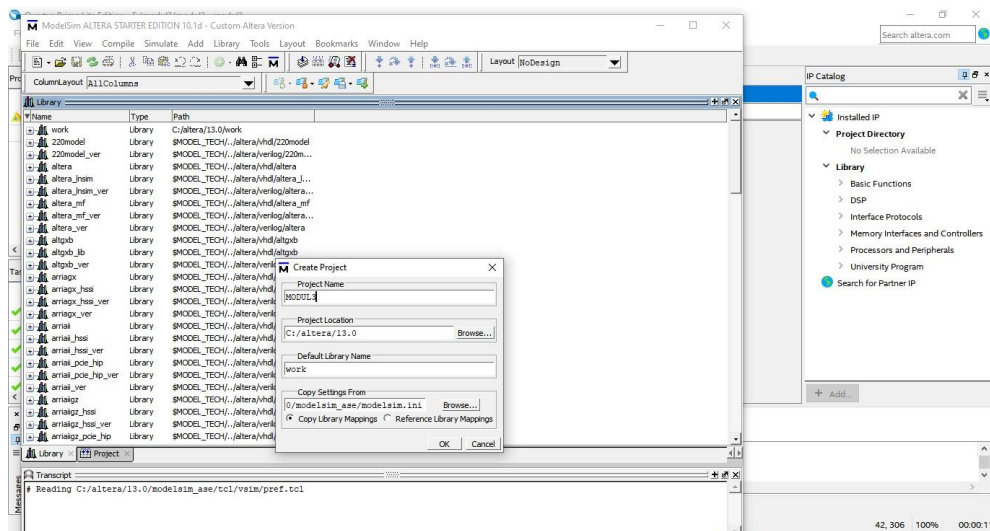
11. Pilih file → *Create/Update* → Create HDL file from Current File



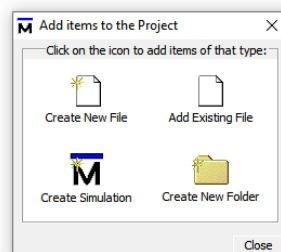
12. Pilih Verilog HDL → Oke



13. Buka Software ModelSim pilih file → New → Project → Create project → oke

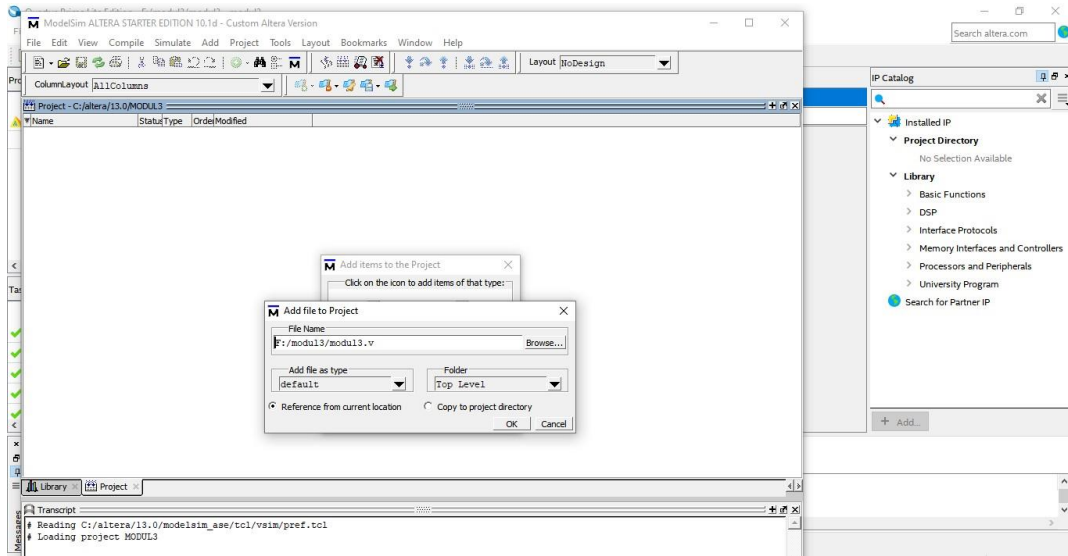


14. Pilih Add existing file

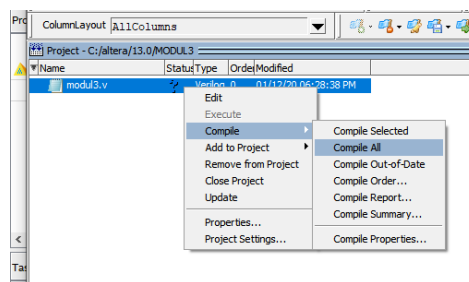


15. Pilih project yang sebelumnya telah di simpan → oke

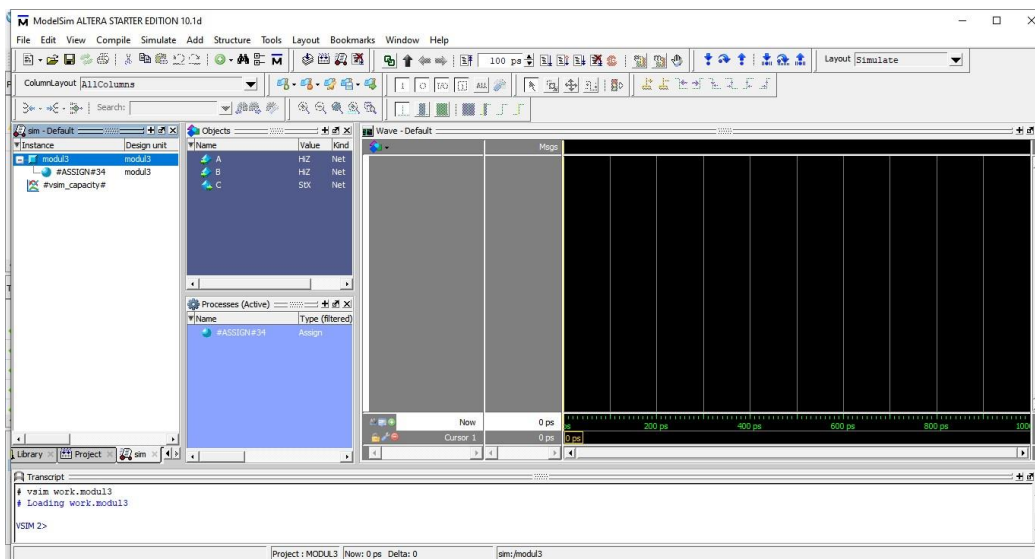
Modul Praktikum



16. Klik kanan pada Project → Compile → Compile All

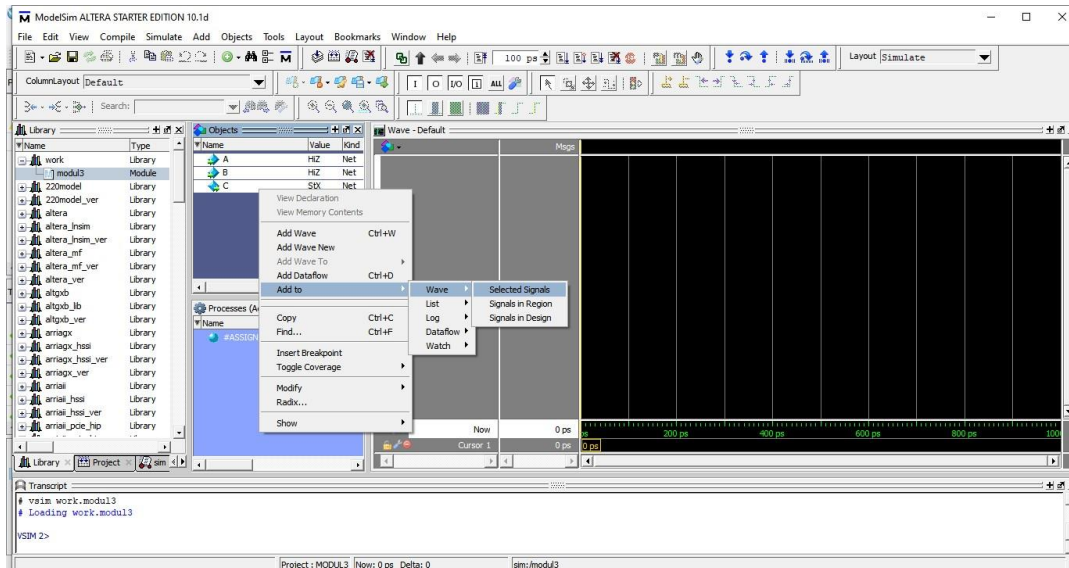


17. Pilih Library → pilih nama file yang telah dibuat → klik dua kali

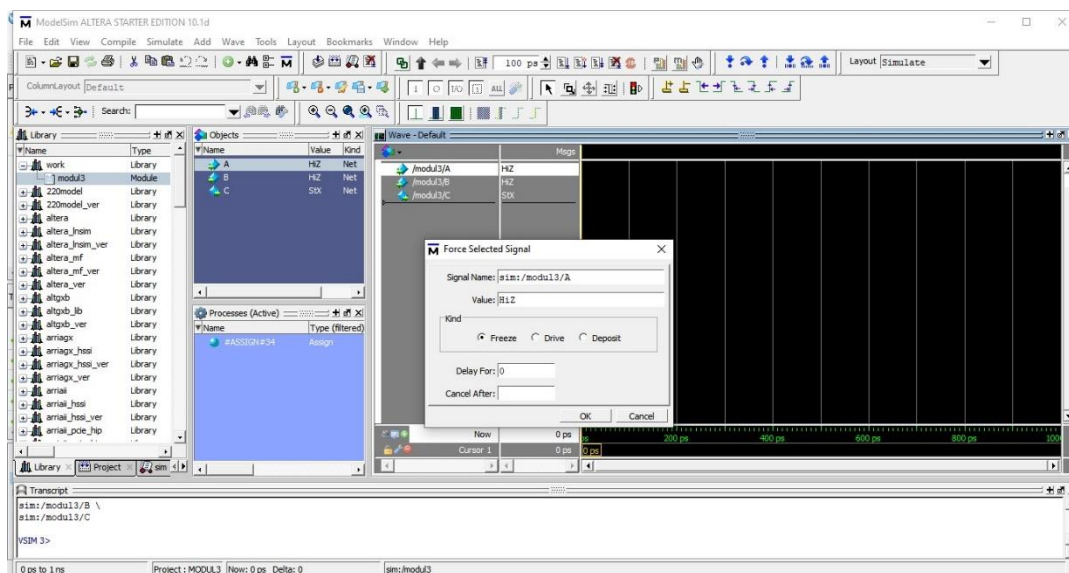


18. Blok semua → Klik kanan → Add to → wave → Selected Signal

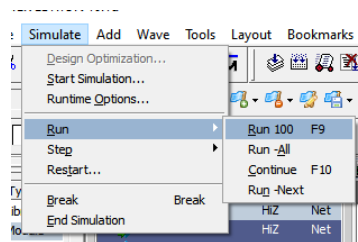
Modul Praktikum



19. Blok salah satu object → Klik kanan → Force → Ubah Value menjadi 1 atau 0 lalu ulangi pada object kedua.

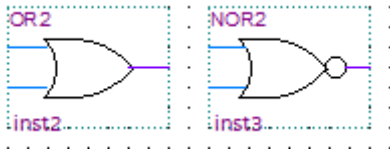


20. Klik Simulate → Run → Run 100 atau F9



3.4 JURNAL

1. Analisa table kebenaran “NAND” dan apa perbedaanya dengan “AND” pada praktikum kali ini berdasarkan ModelSIM!
2. Analisa table kebenaran 2 gerbang logika dibawah ini dengan ModelSIM!



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

MODUL 4

SIMULASI DIGITAL MENGGUNAKAN QUARTUS II (PERANCANGAN BERBASIS SKEMATIK)

4.1 Tujuan Praktikum Modul 4

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Praktikan dapat Mengenal dan Mengetahui Quartus II
2. Praktikan dapat merangkai Gerbang Logika menggunakan *software* Quartus II
3. Praktikan dapat Mensimulasikan Rangkaian digital ke Quartus II

4.2 Dasar Teori Praktikum Modul 4

4.2.1 Quartus Altera II

Quartus merupakan sebuah *software* yang digunakan untuk membuat simulasi rangkaian logika secara digital dengan memanfaatkan bahasa Pemrograman yaitu VHDL ataupun Verilog, dan dapat juga digunakan dengan pembuatan gerbang logika secara visual melalui diagram skematik.

Software yang keluaran dari Altera ini, dapat melakukan *analysis* dan *synthesis* untuk desain, HDL, *compling* desain, analisis diagram pewaktuan, pengetesan reaksi desain kepada beberapa stimulus yang berbeda, dan lain-lain.

Pada praktikum kali ini kita akan membuat sebuah rangkaian gerbang logika dengan menggunakan aplikasi Quartus Altera II pada blok diagram. Perlu diketahui terlebih dahulu *tools* yang ada pada blok diagram, yaitu sebagai berikut.

Tabel 4. 1 Bagian Tools Blok Diagram Quartus Altera II

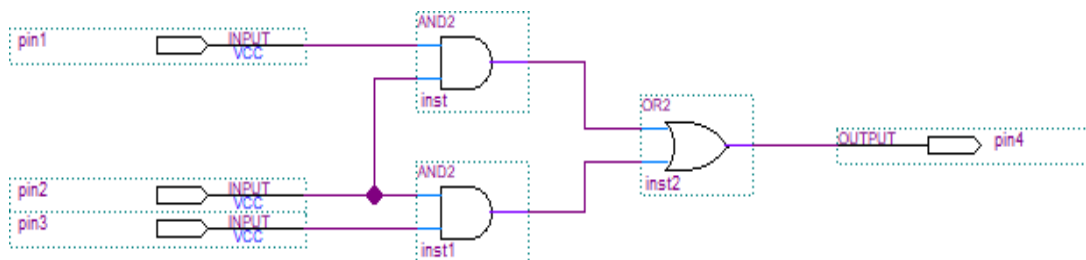
1		<ul style="list-style-type: none"> • Detach Window, untuk memisahkan layar workspace dari layar utama • Selection Tool, untuk menyeleksi suatu objek • Zoom Tool, untuk memperbesar atau memperkecil suatu objek • Hand Tool, untuk menggeser/menggerakkan area objek • Text Tool, untuk membuat text atau tulisan pada lembar kerja • Symbol Tool, untuk menambahkan objek atau symbol berupa gerbang logika, IC, dll • Pin Tool, untuk menambahkan input, output dan bidir pada rangkaian
2		<ul style="list-style-type: none"> • Orthogonal Node, Kabel satu jalur dengan bentuk tegak lurus • Orthogonal Bus, kabel memiliki banyak jalur dalam satu kabel dengan bentuk tegak lurus • Orthogonal Conduit, seperti kabel telepon dengan bentuk tegak lurus • Diagonal Node, kabel satu jalur dengan bentuk diagonal

4.2.2 ModelSim

ModelSim adalah *software* simulator multi Bahasa HDL yang dibuat oleh *Mentor Graphics*. ModelSim digunakan untuk simulasi Bahasa deskripsi perangkat keras seperti VHDL, Verilog, dan SystemC. ModelSim dapat digunakan secara *independent* atau bersamaan dengan Intel Quartus Prime, Xilinx ISE, atau Xilinx Vivado. Simulasi dilakukan menggunakan *interface* grafis (GUI) atau secara otomatis menggunakan skrip.

Dalam peran ModelSim pada praktikum ini untuk menjalankan blok diagram yang sudah dibuat untuk mengetahui fungsi dan hasil dari nilai digital pada gerbang logika.

Gambar 4. 1 Contoh Gerbang Logika



4.3 Lembar Kegiatan Praktikum Modul 4

4.3.1 Alat dan Bahan

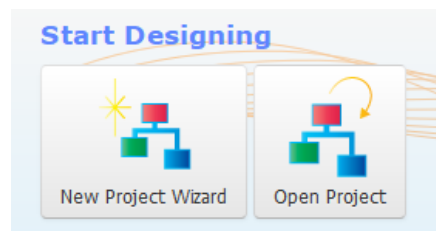
- Laptop yang telah terinstal *software* Quartus Altera II
- Mouse

4.3.2 Langkah Praktikum Modul 4

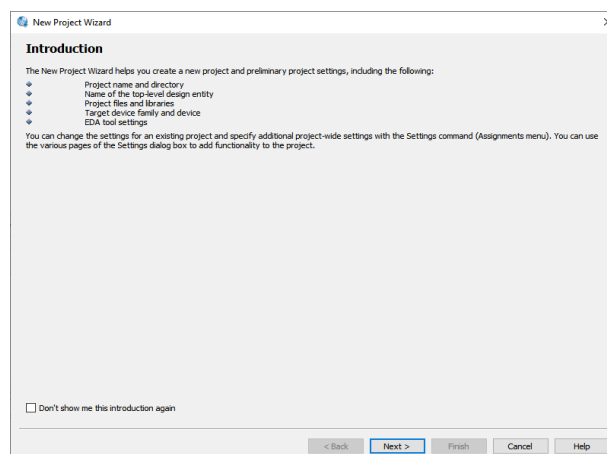
- Buka aplikasi Quartus II 14.1



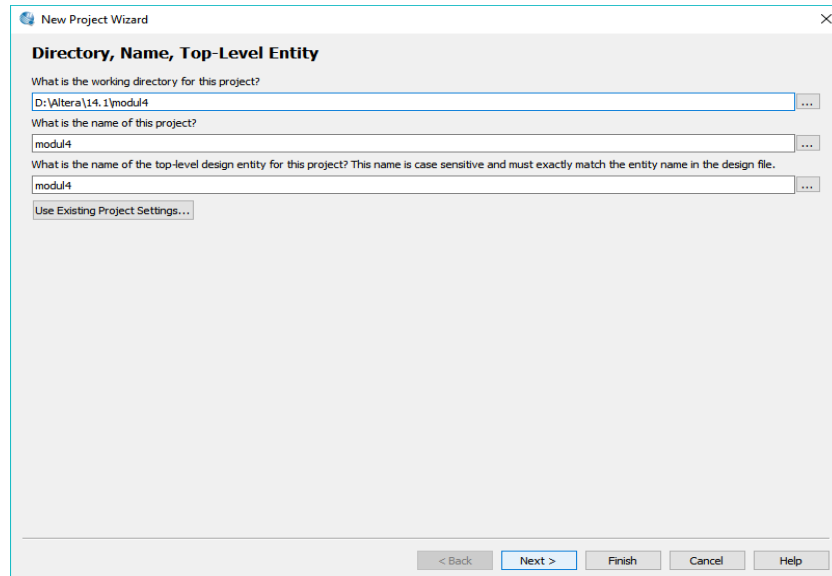
- Buatlah projek baru dengan cara mengklik pada pilihan *New Project Wizard*



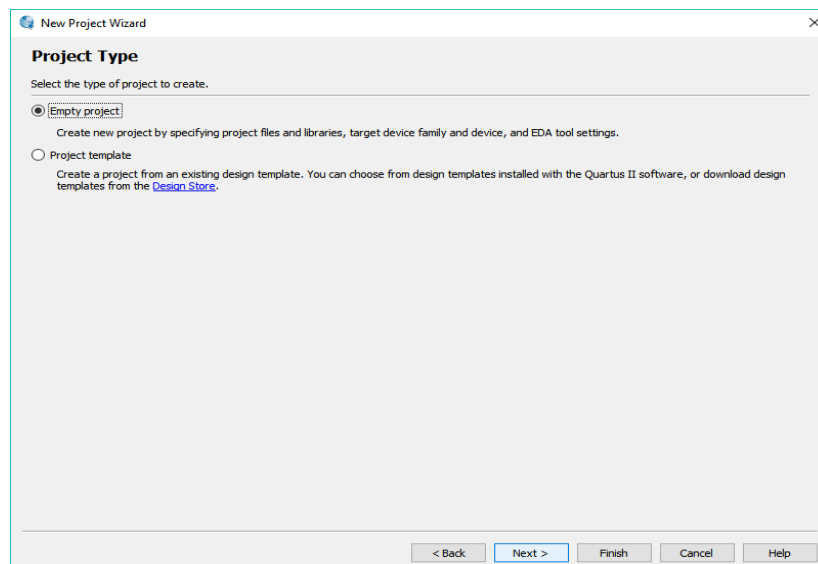
- Klik *Next*



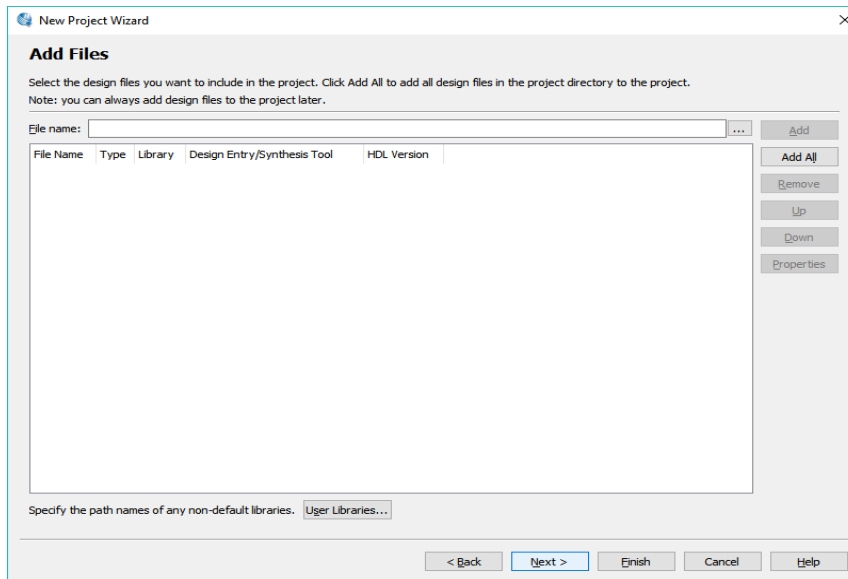
4. Tulis **namamodul** pada bagian *directory* agar *file* proyek dapat terkumpul pada 1 folder utama. Dengan nama yang sama pada *directory*, tulis juga pada bagian nama *project* dan nama *top-level design entity* (*top-level design entity* bersifat *case sensitive* dan harus persis dengan nama *entity* pada *file* desain)



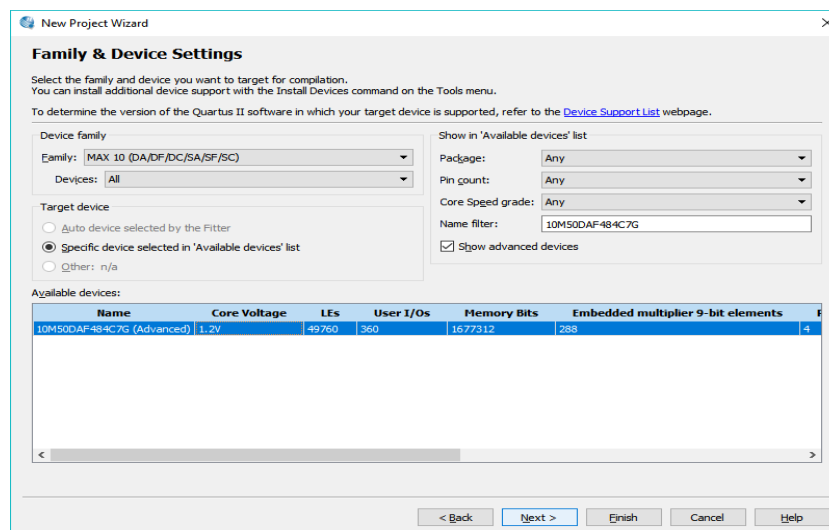
5. Pilih empty project → klik next



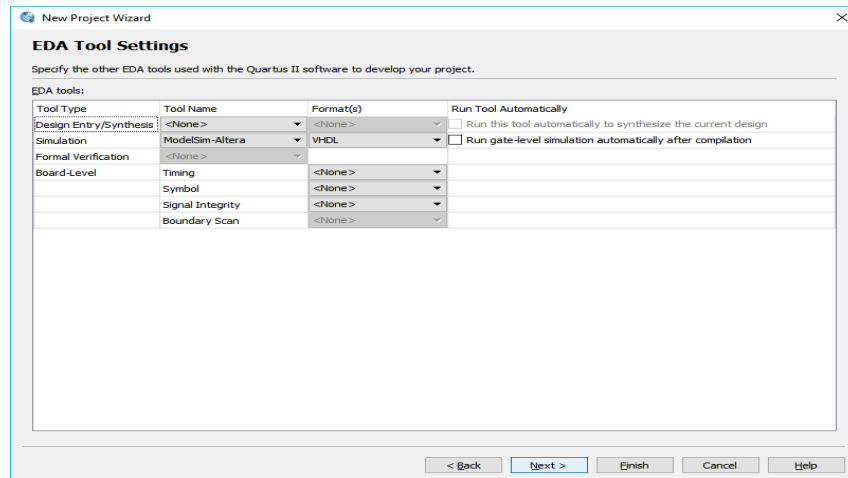
6. Klik *next*



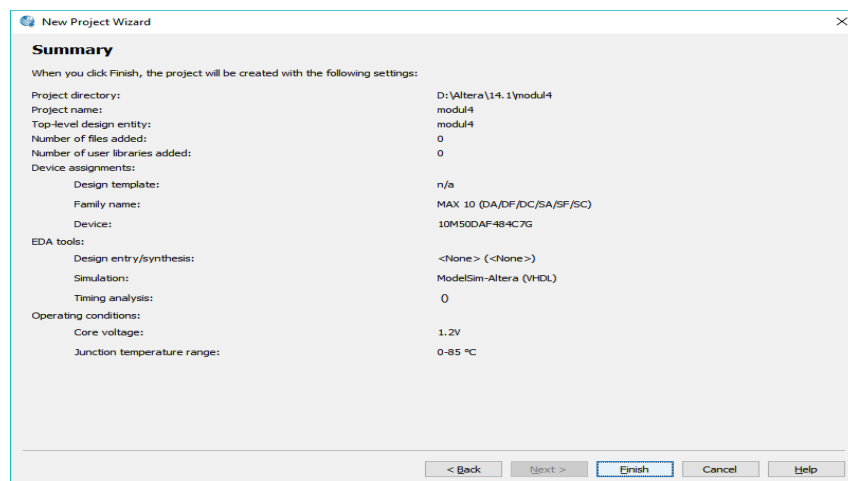
7. Klik *Family* pilih opsi **Max 10** ketik Nama filter sesuai yang ada di FPGA “**10M50DAF484C7G**”, lalu klik *next*



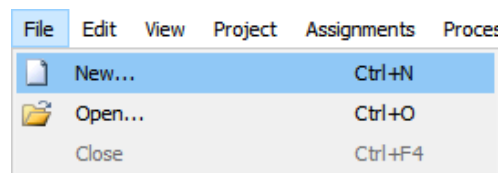
8. Klik *next*



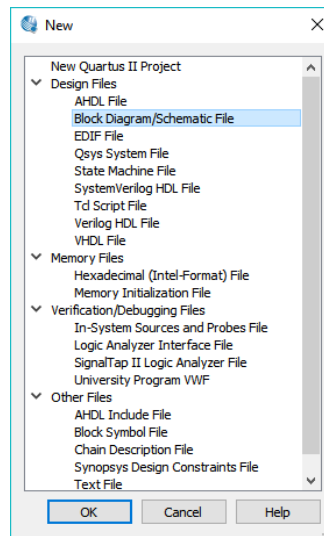
9. Lalu Finish



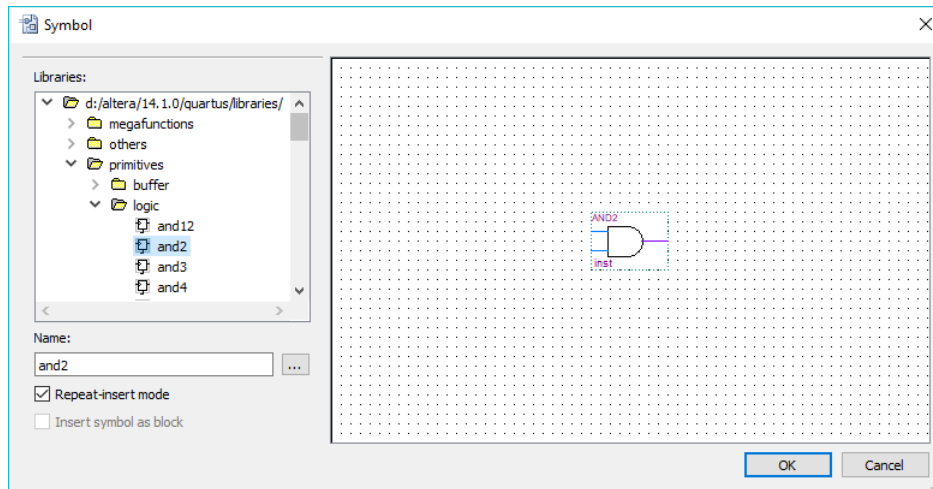
10. Buat *file* baru dengan klik *File*, lalu *New*. Dapat juga menggunakan shortcut **Ctrl+N**.



11. Pilih Block Diagram/Schematic File → Klik OK

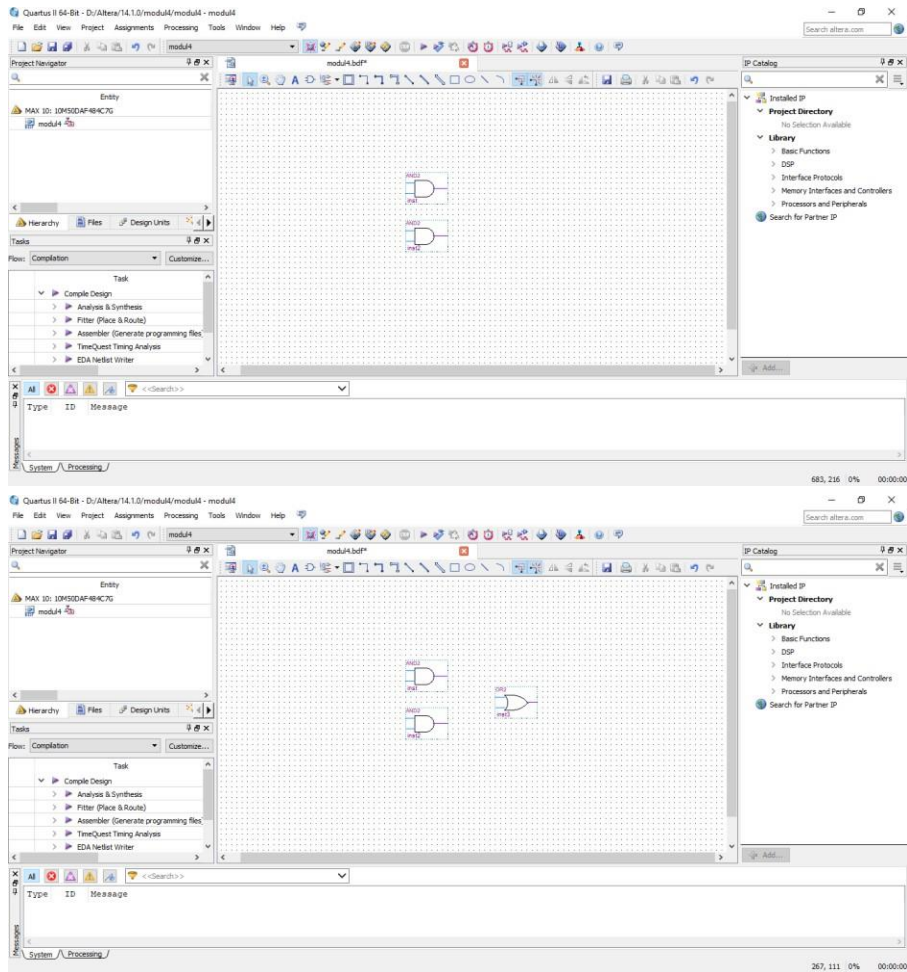


12. Jika workspace-nya sudah ditampilkan, buatlah rangkaian berikut seperti yang sudah di contohkan pada gambar menggunakan tool, untuk mencari gerbang logikanya, Klik Symbol Tool → buka folder Libraries → buka folder Primitives → buka folder Logic → lalu carilah gerbang logikanya. Lalu Ok

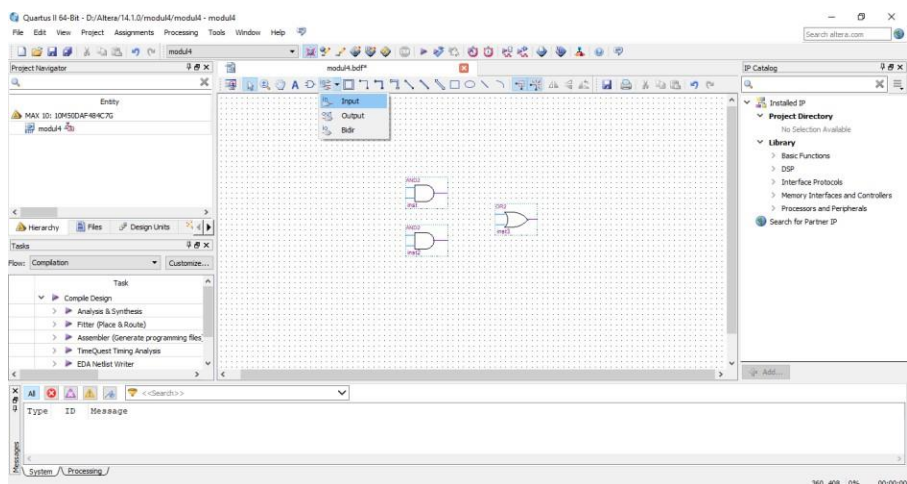


13. Lakukan seterusnya hingga total gerbang ada 3

Modul Praktikum

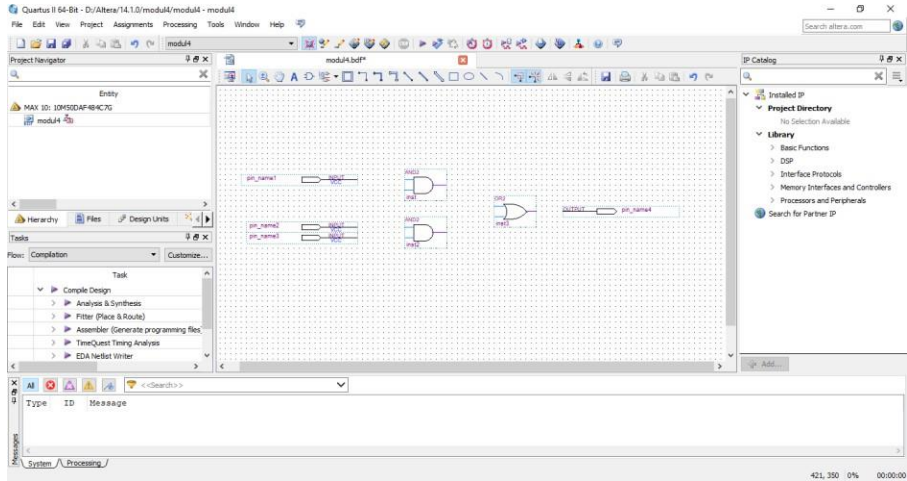


14. Setelah itu, Untuk mencari pin **Input** dan **Output**-nya, pilih **Pin Tool** → klik yang tanda panah kebawah → lalu pilih yang mau di tampilkan.

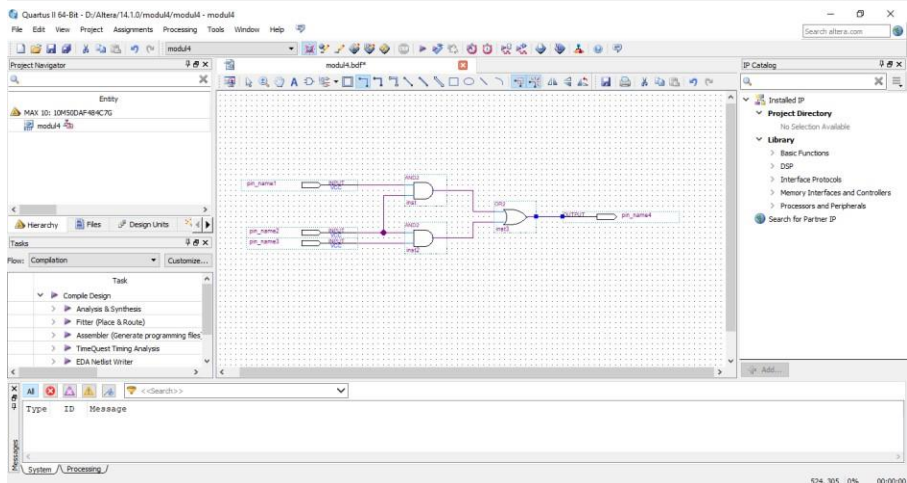
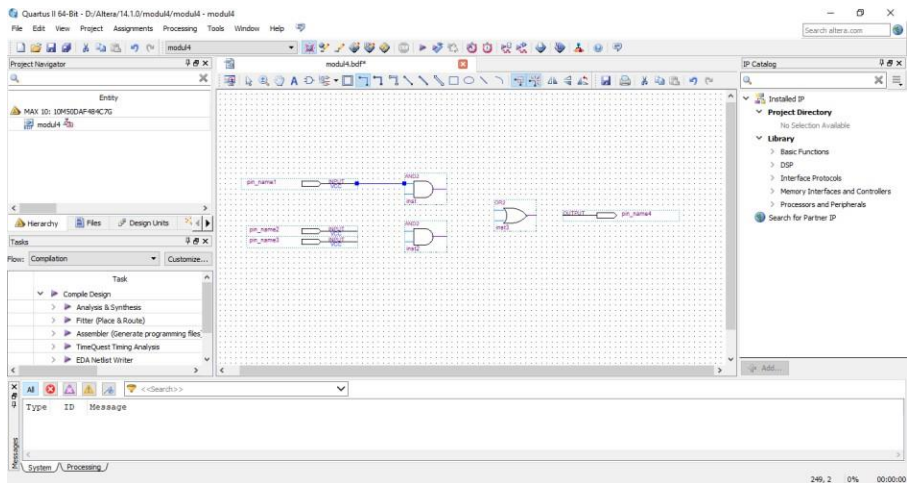


15. Lakukan seterusnya hingga ada 3 input dan 1 output

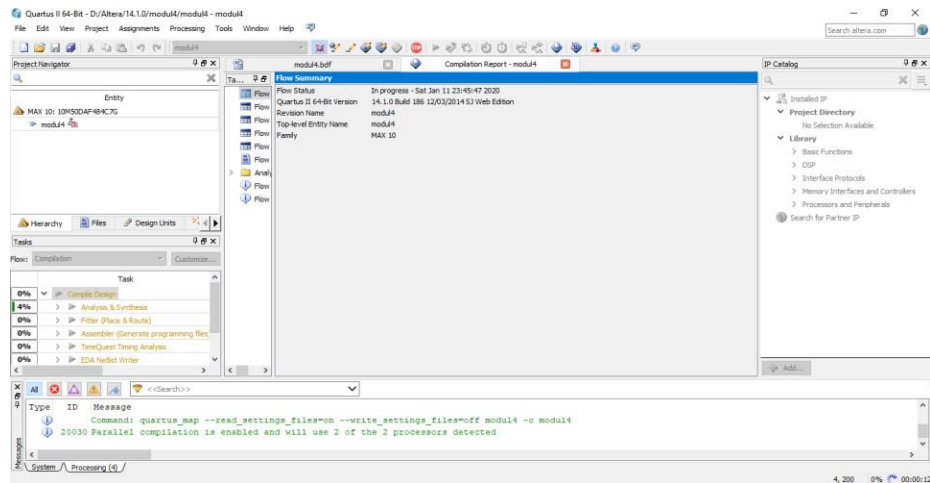
Modul Praktikum



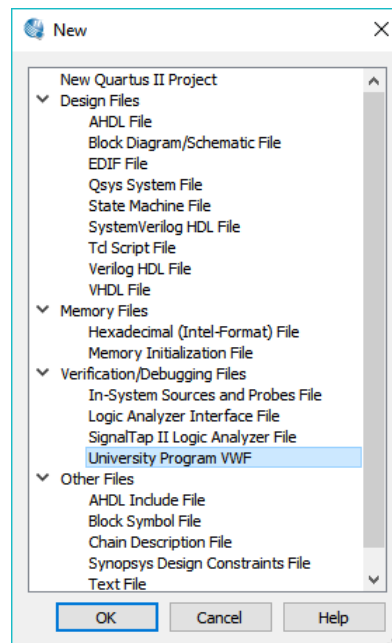
16. Lalu lakukan *wiring* antar kaki pin-nya



17. Jika sudah merangkainya, klik **file** → save as (utamakan Namanya sama dengan file project yang sudah kalian buat), klik Processing → klik Start Compilation → tunggu hingga 100% Complete

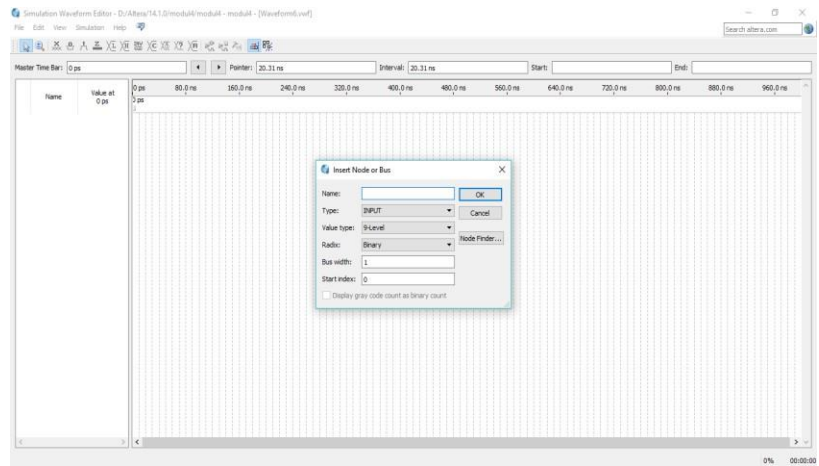


18. Klik **File** → klik **New** → klik **University Program VWF** → klik OK

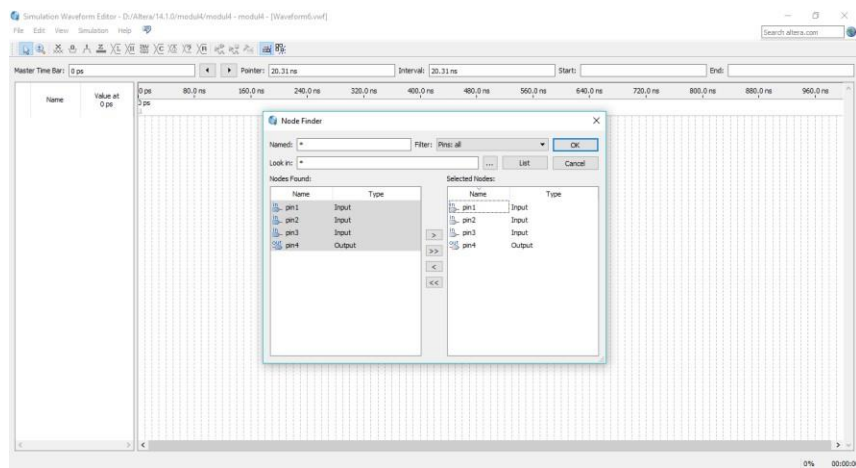


19. Jika sudah ditampilkan, klik **Edit** → klik **Insert** → klik **Insert Node or Bus** → klik **Node Finder**

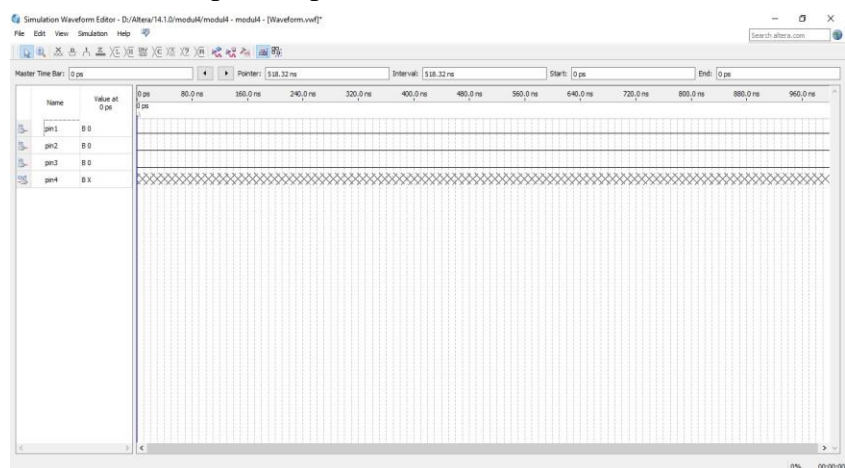
Modul Praktikum



20. Klik *list* → klik simbol “>>” lalu OK → klik OK

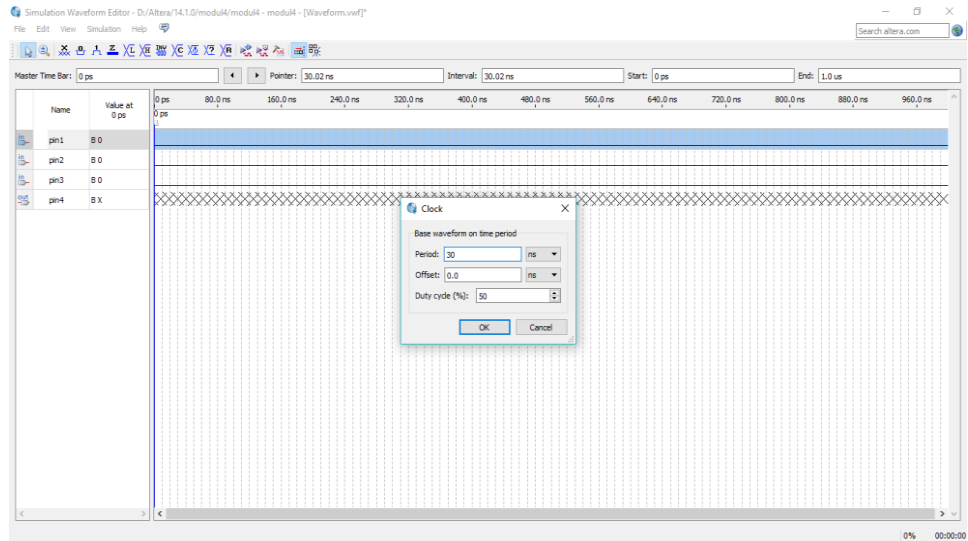


21. Lalu akan muncul tampilan seperti berikut

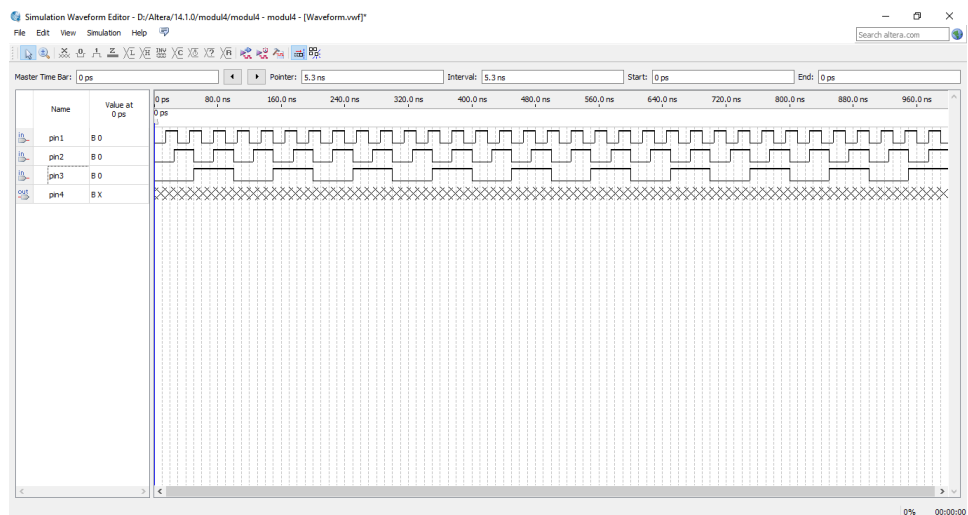


22. Klik pin1 atau input1 → klik *edit* → klik *value* → pilih *overwrite clock*. Ganti periodenya menjadi 30 (optional), lalu OK → lakukan seterusnya kecuali pin 4 atau pin output

Modul Praktikum



23. Jika sudah akan menjadi seperti gambar berikut



24. Klik **Simulation** → klik **Run Time Simulation** → Save sesuai dengan nama project kalian. Lalu pin4 akan menampilkan hasil output-nya seperti gambar berikut

MODUL 5

SIMULASI RANGKAIAN DIGITAL SEDERHANA

5.1 Tujuan Kegiatan Praktikum 5

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Praktikan mampu membuat rangkaian digital sederhana pada software Quartus.
2. Praktikan mampu memahami FPGA DE10-Lite.
3. Praktikan mampu mensimulasikan Rangkaian Digital Sederhana

5.2 Dasar Teori Kegiatan Praktikum 5

5.2.1 FPGA (Field Programmable Gate Array)

FPGA atau Field Programmable Gate Array, merupakan integrated circuit (IC) yang dapat diprogram sesuai dengan kebutuhan user. FPGA dapat diprogram dengan menggunakan bahasa pemrograman HDL (Hardware Description Language) atau menggunakan diagram skematik.

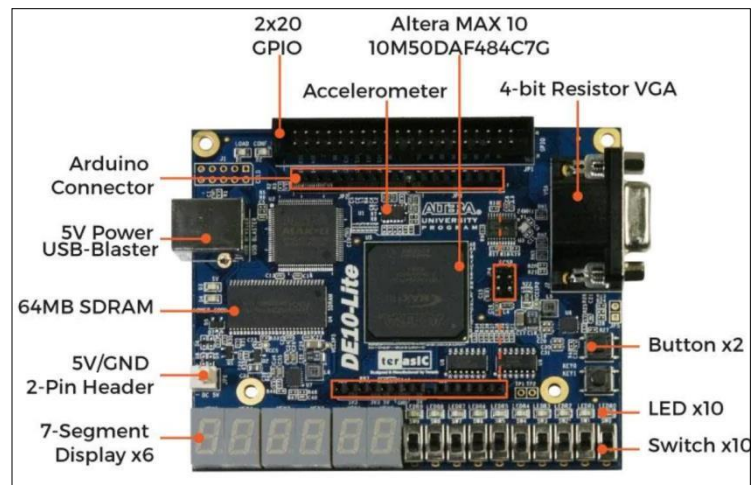
Gambar 5. 1 FPGA XILINK Spartan



5.2.2 Board FPGA DE10-Lite

Board FPGA DE10-Lite merupakan salah satu development board untuk FPGA. Board FPGA ini menggunakan family MAX 10 dengan nama device 10M50DAF484C7G. Berikut adalah spesifikasi dari DE10-Lite.

Gambar 5. 2 FPGA Board DE10-Lite



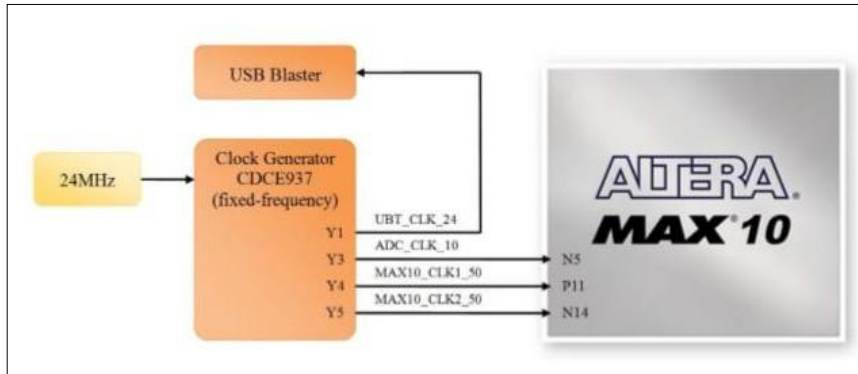
- a. MAX 10 10M50DAF484C7G Device
- b. 50.000 logic elements max.
- c. Integrated dual ADCs, each ADC supports 1 dedicated analog input and 8 dual function pins
- d. 1,638 Kbits M9K Memory
- e. 2 Push Button Debounced
- f. USB Type B port
- g. 5,888 Kbits user flash memory 32MB SDRAM
- h. 144 18×18 Multiplier
- i. 2x20 GPIO Header
- j. Arduino Uno R3 Connector, including six ADC channels
- k. 10 LEDs
- l. 5V DC input from USB or external power connector
- m. 64MB SDRAM, x16 bits data bus
- n. 10 Slide Switches
- o. Six 7-Segments

Untuk menggunakan *Board* FPGA DE10-Lite, perlu diketahui pin-pin yang ada agar dapat dimanfaatkan dengan baik.

a. Clock Inputs

Gambar 5.3 menunjukkan frekuensi standar pada seluruh clock yang ada pada FPGA MAX 10. Clock generator berfungsi untuk mengalirkan sinyal clock dengan jitter rendah. Kedua clock 50MHz digunakan sebagai sumber clock untuk logika pengguna dan clock 24MHz dihubungkan ke USB Blaster.

Gambar 5. 3 Clock Circuit untuk board FPGA



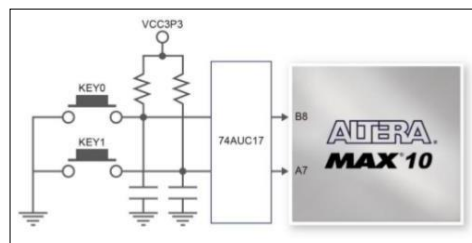
Tabel 5. 1 Pin assignment untuk clock Inputs

Signal Name	FPGA Pin No.	Description	I/O Standard
ADC_CLK_10	PIN_N5	10 MHz clock input for ADC (Bank 3B)	3.3-V LVTTTL
MAX10_CLK1_50	PIN_P11	50 MHz clock input(Bank 3B)	3.3-V LVTTTL
MAX10_CLK2_50	PIN_N14	50 MHz clock input(Bank 3B)	3.3-V LVTTTL

b. Push-buttons

Board ini memiliki 2 push-buttons yang memungkinkan pengguna untuk terhubung dengan perangkat FPGA MAX 10. Dengan menggunakan sirkuit Schmitt Trigger, pengguna dapat meningkatkan ketebalan sinyal terhadap noise, khususnya sinyal lemah.

Gambar 5. 4 Hubungan antara push-button dengan FPGA MAX 10



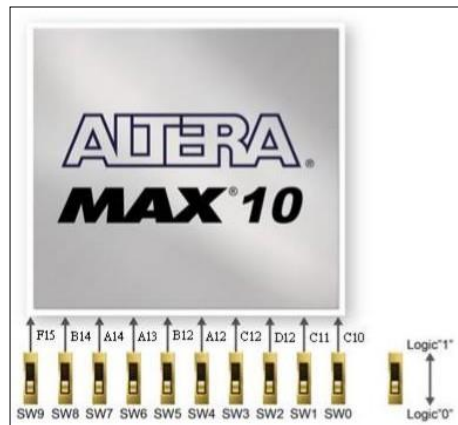
Tabel 5.2 Pin assignment untuk push-buttons

Signal Name	FPGA Pin No.	Description	I/O Standard
KEY0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER"
KEY1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER"

c. Slide Switches

Terdapat 10 slide switches pada board FPGA. Switch ini berfungsi untuk mengirimkan data input ke dalam sirkuit FPGA. Setiap switch sudah terhubung dengan masing-masing pin yang dapat memberikan masukan berupa nilai 0 (*low*) ketika berada di posisi bawah dan bernilai 1 (*high*) ketika berada di posisi atas.

Gambar 5. 5 Koneksi antara slide switches dan FPGA MAX 10



Tabel 5.3 Pin assignment untuk slide switches

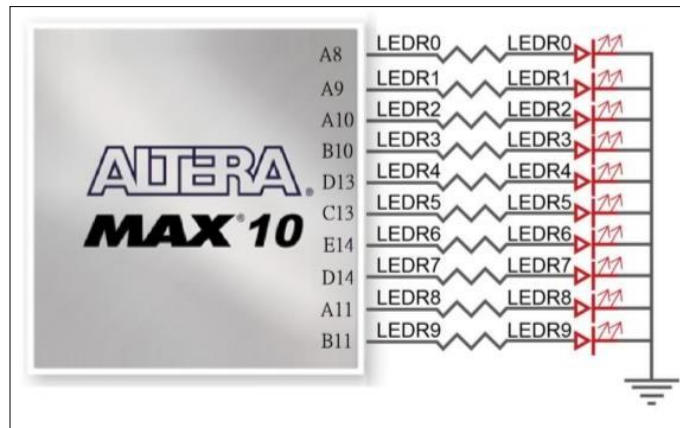
Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTTL

d. LEDs

Selain 10 switch, ada juga 10 LED berwarna merah yang terhubung ke FPGA. Tiap LED sudah terhubung secara langsung dengan masing-masing pin.

Ketika logika 1 (high), maka LED akan menyala dan ketika logika 0 (low), maka LED akan mati.

Gambar 5. 6 Koneksi antara LED dan FPGA MAX 10



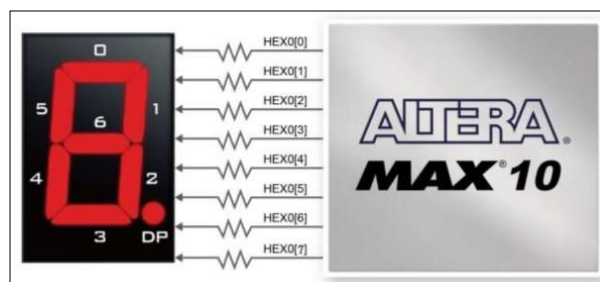
Tabel 5. 4 Pin assignment untuk LED

Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTTL

e. 7-segment Displays

Fitur lain pada board FPGA MAX 10 adalah 7-segment. Pada board ini terdapat 6 buah 7-segment (common anode) yang setiap pin sudah terhubung langsung dengan FPGA. 7-segment dapat dinyalakan dengan mengatur logika 1 (*high*) dan 0 (*low*).

Gambar 5. 7 Koneksi antara 7-segment dengan FPGA



Tabel 5. 5 Pin assignment 7-segment

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

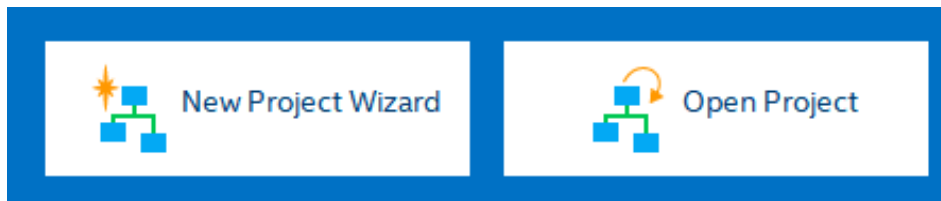
5.3 Lembar Kegiatan Praktikum Modul 5

5.3.1 Alat dan Bahan

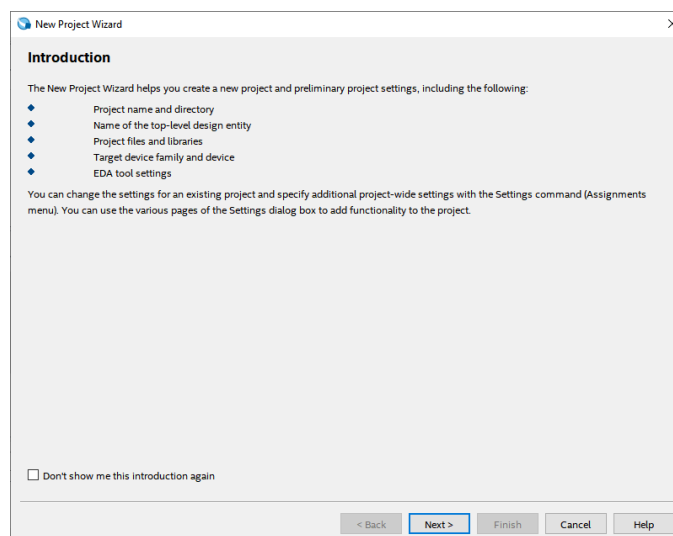
- Laptop yang sudah terinstall Quartus
- Mouse

5.3.2 Langkah Praktikum Modul 5

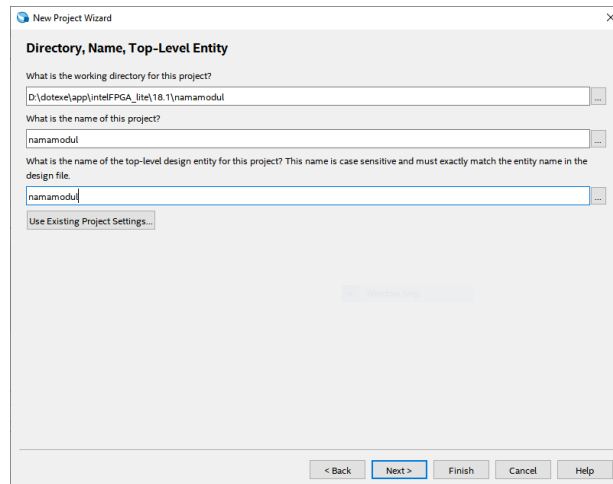
- Buka aplikasi Quartus.
- Buatlah projek baru dengan cara mengklik pada pilihan **New Project Wizard**.



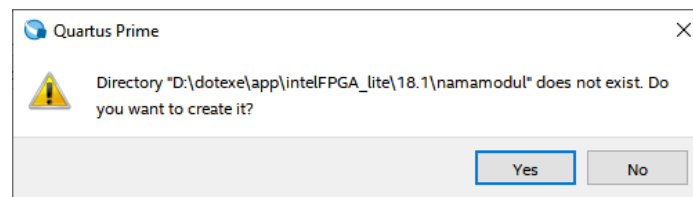
- Klik **Next**.



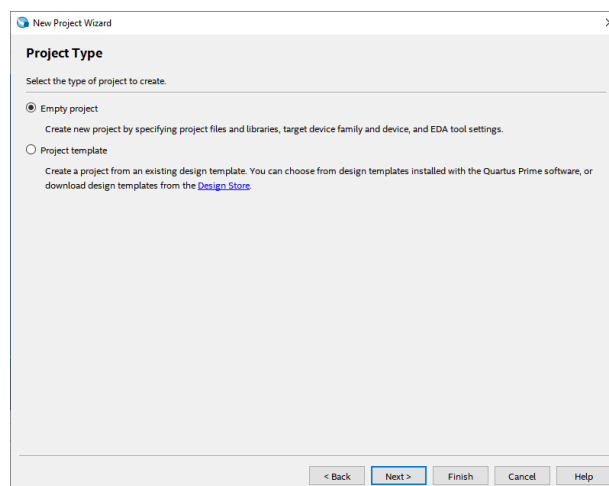
4. Tulis namamodul pada bagian directory agar file proyek dapat terkumpul pada 1 folder utama. Dengan nama yang sama pada directory, tulis juga pada bagian nama project dan nama *top-level design entity* (*top-level design entity* bersifat *case ensitive* dan harus persis dengan nama *entity* pada file desain).



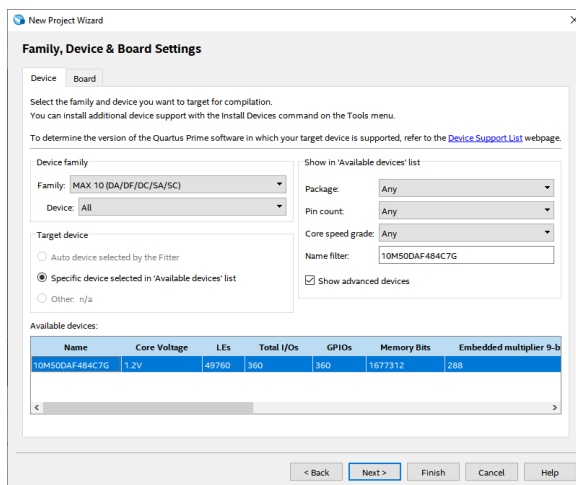
5. Klik **Yes** agar *directory* baru dapat dibuat.



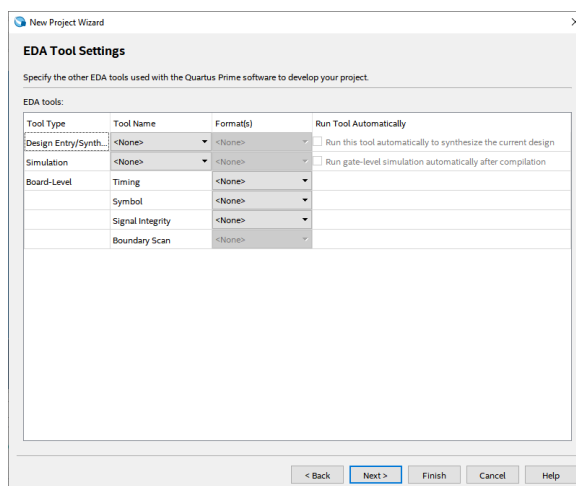
6. Pilih bagian **Empty Project** lalu klik Next.



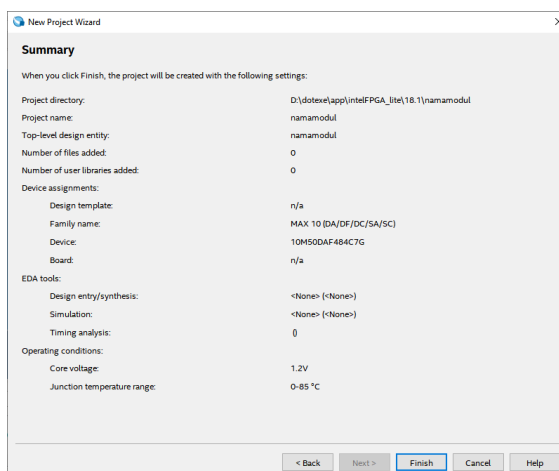
7. Pada bagian *family*, pilih MAX 10. Tuliskan 10M50DAF484C7G pada Name filter kemudian klik *Available devices* yang ada. Klik **Next**.



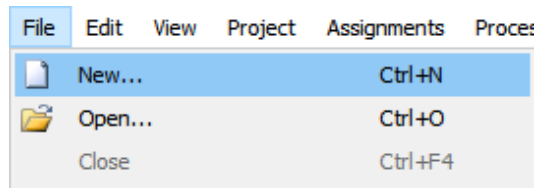
8. Klik **Next**.



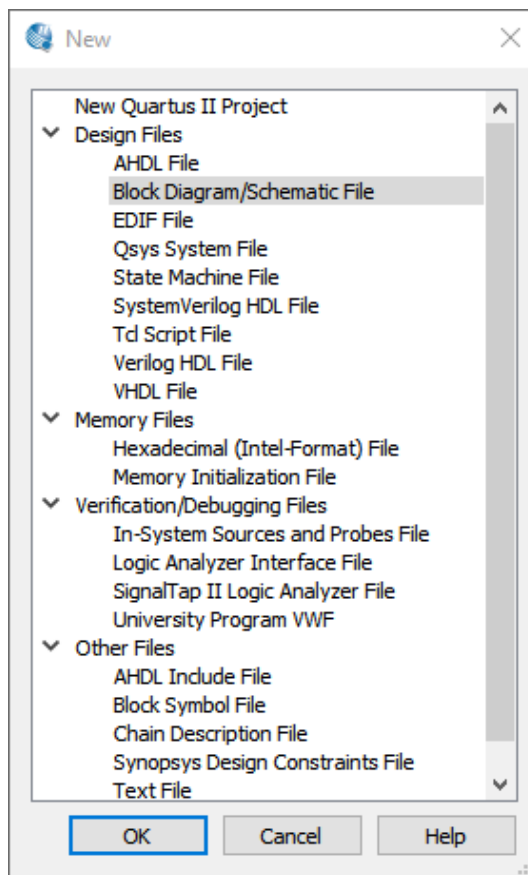
9. Klik **Finish**.



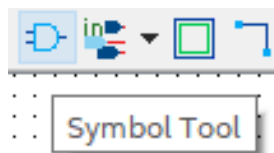
10. Buat file baru dengan klik **File**, lalu **New**. Dapat juga menggunakan *shortcut* Ctrl+N.



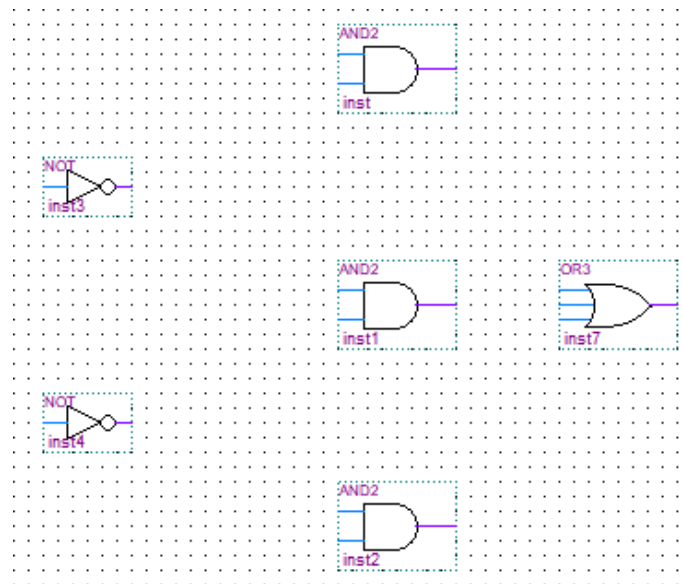
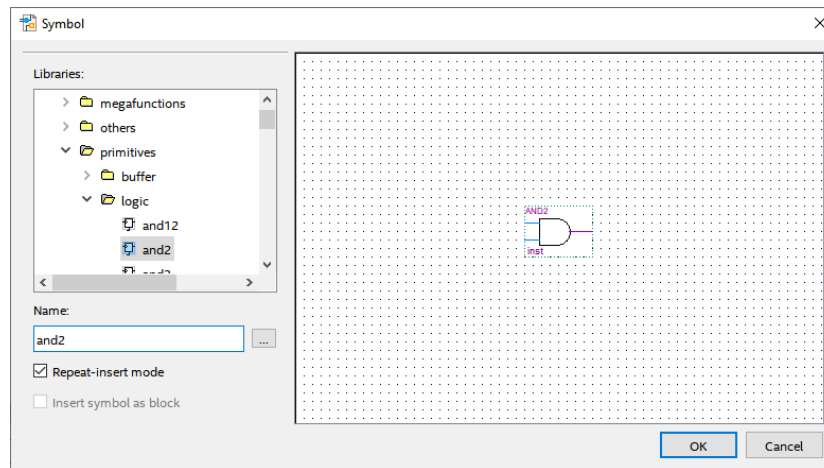
11. Pilih **Block Diagram/Schematic File**, kemudian klik **OK**.



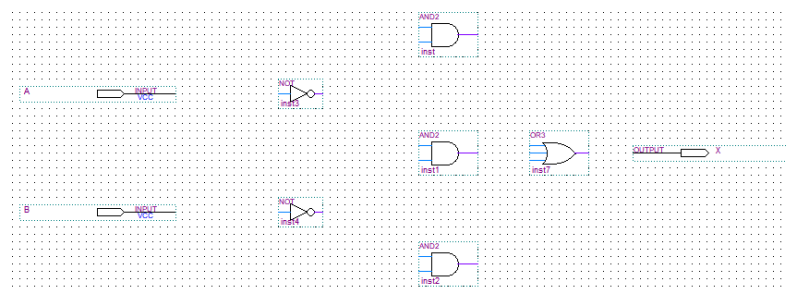
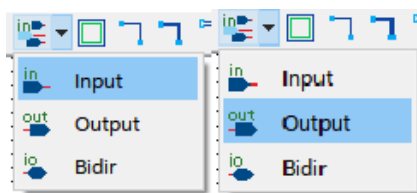
12. Klik icon gerbang logika.



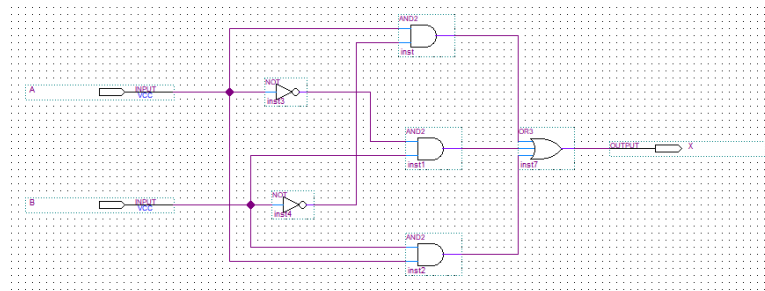
13. Cari gerbang logika NOT, AND2, dan OR3, lalu rangkai seperti gambar.



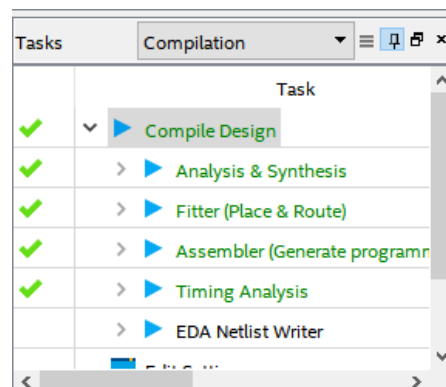
14. Pilih Pin Tool untuk Input dan Output kemudian rangkai seperti gambar.



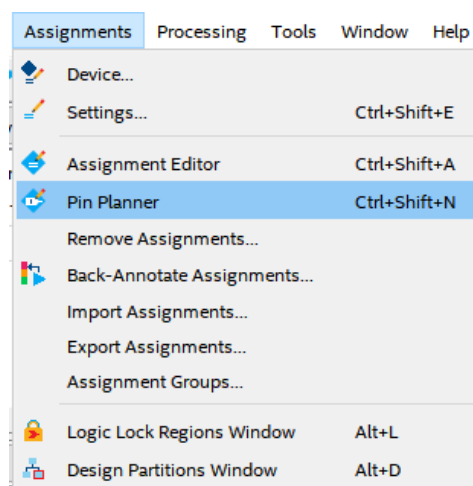
15. Lakukan *wiring* dengan menghubungkan satu titik dengan titik yang lain.



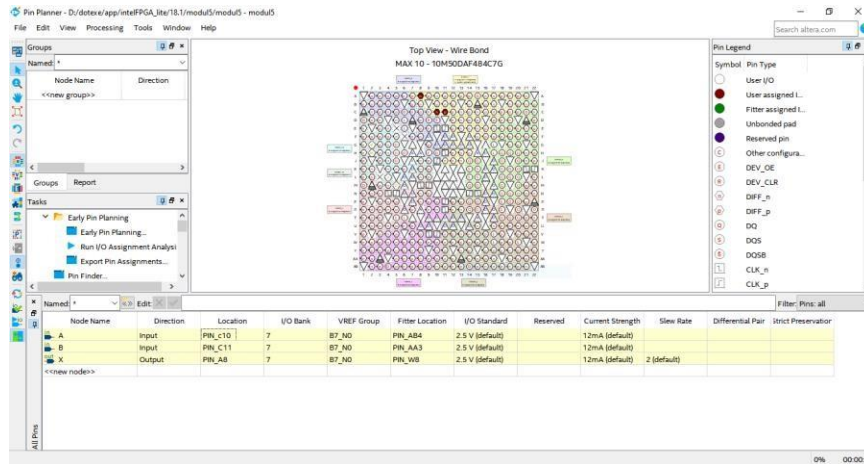
16. *Double-click* pada bagian **Compile Design** lalu tunggu hingga muncul centang hijau yang menandakan *compile* berhasil dilakukan.



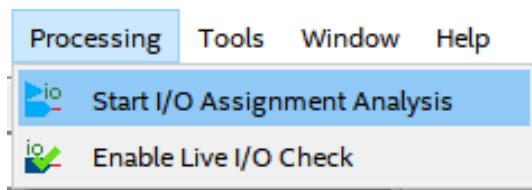
17. Pada menu, pilih Assignment, kemudian klik Pin Planner.



18. Sesuaikan *pin assignment* seperti pada gambar.



19. Pada menu, pilih **Processing**, kemudian klik **Start I/O Assignment Analysis**.

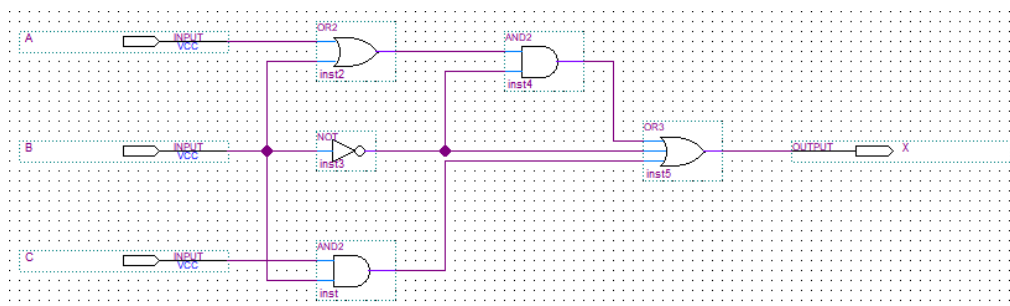


20. Tunggu hingga hasil *compile* selesai, kemudian lakukan **Compile Design** lagi.

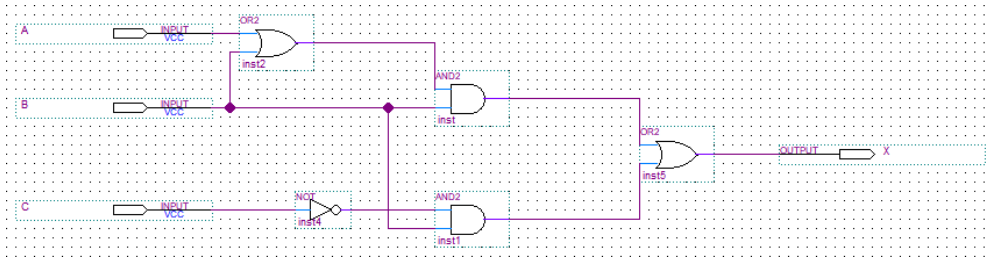
5.4 Soal Jurnal

1. Buatlah rangkaian berikut pada Quartus!

a. Rangkaian 1



b. Rangkaian 2



2. Jawablah pertanyaan berikut pada buku praktikum!
 - a. Sebutkan dan jelaskan minimal 5 pin yang ada pada Board FPGA DE10- Lite!
 - b. Apakah input A dan input B pada rangkaian praktikum dapat diganti dengan menggunakan push-button? Jelaskan!
3. Tuliskan apa yang telah dilakukan pada praktikum modul 5 menggunakan Bahasa kalian sendiri!

MODUL 6

SIMULASI RANGKAIAN PENJUMLAHAN DAN PENGURANGAN 4-BIT

6.1 Tujuan Praktikum Modul 6

Setelah mempraktekan topik ini, praktikan diharapkan dapat :

1. Dapat menggunakan Quartus 18 Prime Lite
2. Dapat memahami tipe data pada VHDL dan Verilog HDL
3. Dapat memahami konsep Adder dan mampu mengimplementasikannya dalam VHDL dan Verilog HDL

6.2 Dasar Teori Modul 6

6.2.1 Quartus 18 Prime Lite

Quartus 18 Prime Lite merupakan software untuk perancangan elemen logika untuk FPGA keluaran Altera. Dengan Quartus 18 Prime Lite, pengembang dapat melakukan analysis and synthesis untuk desain HDL, compiling desain, analisis diagram pewaktuan, pengetesan reaksi desain kepada beberapa stimulus yang berbeda, dan lain-lain. Quartus 18 Prime Lite dapat digunakan dengan Bahasa pemrograman VHDL atau Verilog HDL, dan dapat juga digunakan dengan pembuatan gerbang logika secara visual melalui diagram skematik.

6.2.2 Pin Plener

6.2.2.1 Pin Plener Switch

Tabel 6.1 Pin plener switch

Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTTL

6.2.2.2 Pin Plener LED

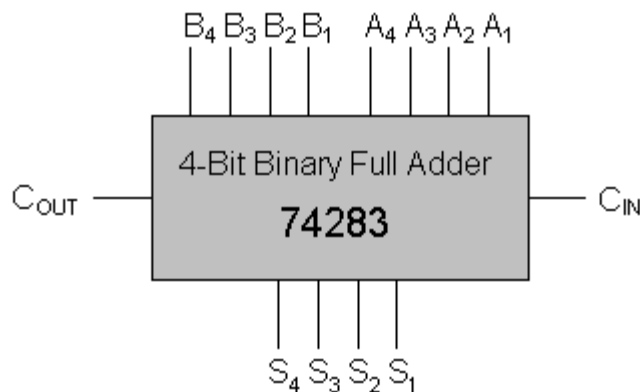
Tabel 6.2 Pin plener LED

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR0	PIN_A8	LED [0]	3.3-V LVTTTL
LEDR1	PIN_A9	LED [1]	3.3-V LVTTTL
LEDR2	PIN_A10	LED [2]	3.3-V LVTTTL
LEDR3	PIN_B10	LED [3]	3.3-V LVTTTL
LEDR4	PIN_D13	LED [4]	3.3-V LVTTTL
LEDR5	PIN_C13	LED [5]	3.3-V LVTTTL
LEDR6	PIN_E14	LED [6]	3.3-V LVTTTL
LEDR7	PIN_D14	LED [7]	3.3-V LVTTTL
LEDR8	PIN_A11	LED [8]	3.3-V LVTTTL
LEDR9	PIN_B11	LED [9]	3.3-V LVTTTL

6.2.3 Adder

Penjumlahan atau Adder adalah komponen elektronika digital yang dipakai untuk menjumlahkan dua buah angka dalam sistem bilangan biner. Dalam komputer dan mikroprosesor, Adder biasanya berada di bagian ALU (Arithmetic Logic Unit). Sistem bilangan yang dipakai dalam proses penjumlahan, selain bilangan biner, juga 2's complement untuk bilangan negatif, bilangan BCD (binary-coded decimal), dan excess-3. Jika sistem bilangan yang dipakai adalah 2's complement, maka proses operasi penjumlahan dan operasi pengurangan akan sangat mudah dilakukan.

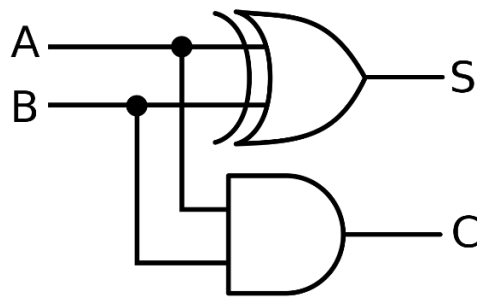
Gambar 6.1 IC 74283 (4bit adder)



6.2.3.1 Half Adder

Half Adder adalah rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan dari dua buah bilangan biner, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki dua input dan dua buah output, salah satu outputnya dipakai sebagai tempat nilai pindahan (carry) dan yang lain sebagai hasil dari penjumlahan (sum).

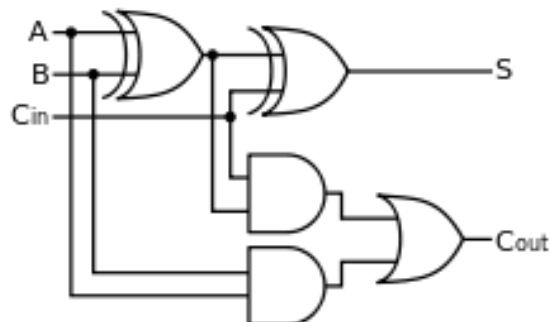
Gambar 6.2 Gerbang logika rangkaian half adder



6.2.3.2 Full Adder

Full Adder adalah rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan sepenuhnya dari dua buah bilangan biner, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki tiga input dan dua buah output, salah satu input merupakan nilai dari pindahan penjumlahan (carry in). Kemudian sama seperti pada half adder salah satu outputnya dipakai sebagai tempat nilai pindahan (carry out) dan yang lain sebagai hasil dari penjumlahan (sum).

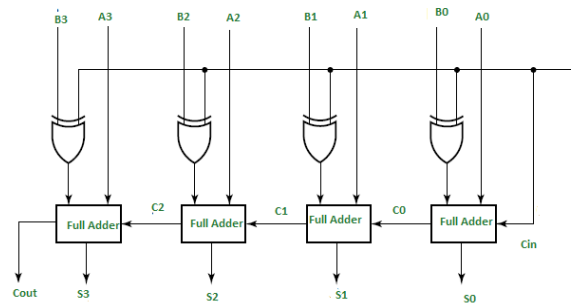
Gambar 6.3 Gerbang logika rangkaian full adder



6.2.4 Binary Adder-Subtractor

Binary Adder-Subtractor adalah salah satu yang mampu menambah dan mengurangi bilangan biner dalam satu sirkuit itu sendiri. Operasi yang dilakukan tergantung pada nilai biner yang dimiliki oleh sinyal kontrol. Ini adalah salah satu komponen dari ALU (Unit Logika Aritmatika). Sirkuit ini membutuhkan pengetahuan prasyarat Gerbang Exor, Penambahan dan Pengurangan Biner, Penambah Lengkap. Mari kita pertimbangkan dua angka biner 4-bit A dan B sebagai input ke Sirkuit Digital untuk operasi dengan digit.

Gambar 6.4 Rangkaian 4 bit adder/subtractor



Jika nilai K (garis Kontrol) adalah 1, keluaran th dari B0 (exor) $K = B0'$ (Komplemen B0). Dengan demikian operasi akan menjadi $A + (B0')$. Sekarang pengurangan komplemen 2 untuk dua angka A dan B diberikan oleh $A + B'$. Ini menunjukkan bahwa ketika $K = 1$, operasi yang dilakukan pada empat angka bit adalah pengurangan.

Demikian pula jika Nilai $K = 0$, B0 (exor) $K = B0$. Operasi adalah $A + B$ yang merupakan penambahan biner sederhana. Ini menunjukkan bahwa Ketika $K = 0$, operasi yang dilakukan pada empat angka bit adalah tambahan.

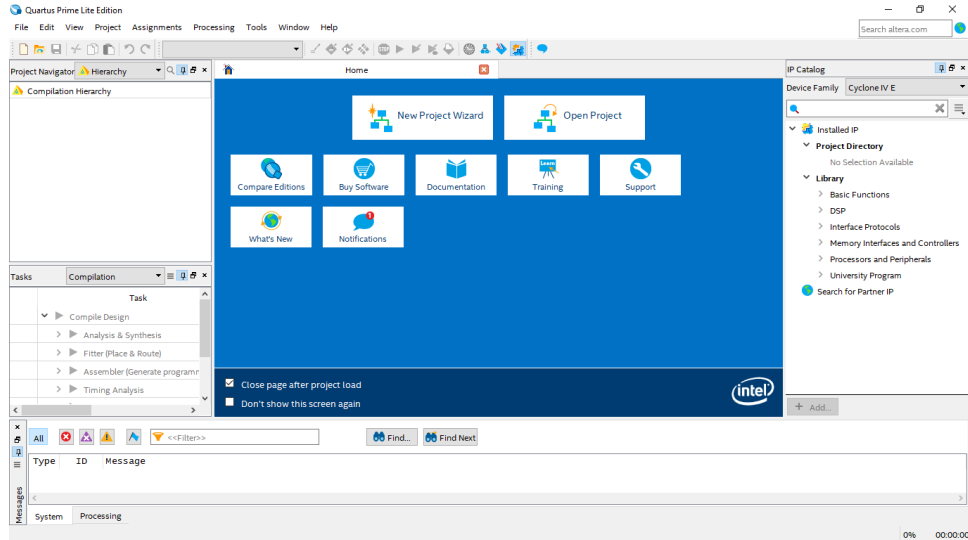
6.3 Lembar Kegiatan Praktikum Modul 6

6.3.1 Alat dan bahan

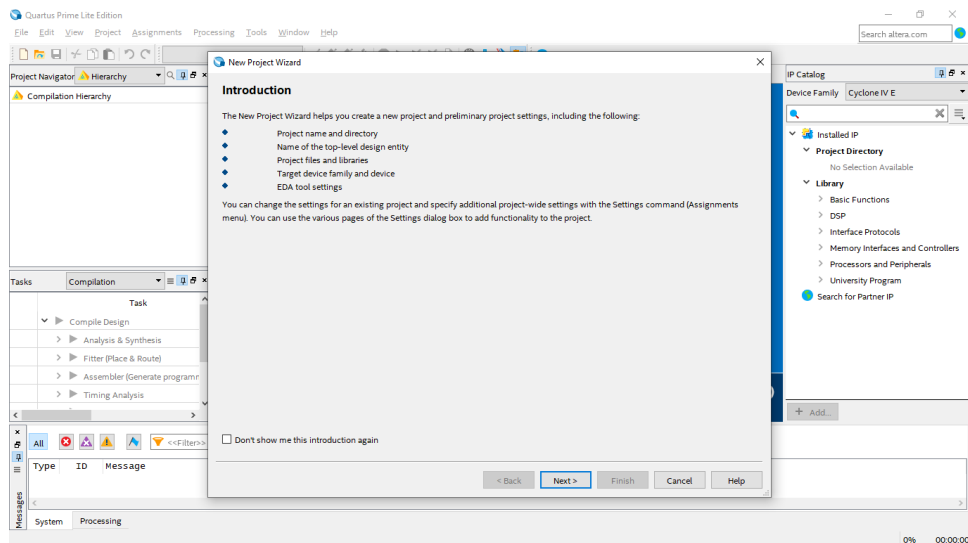
1. Laptop
2. *Software Quartus 18 Prime Lite*

6.3.2 Langkah Praktikum Modul 6

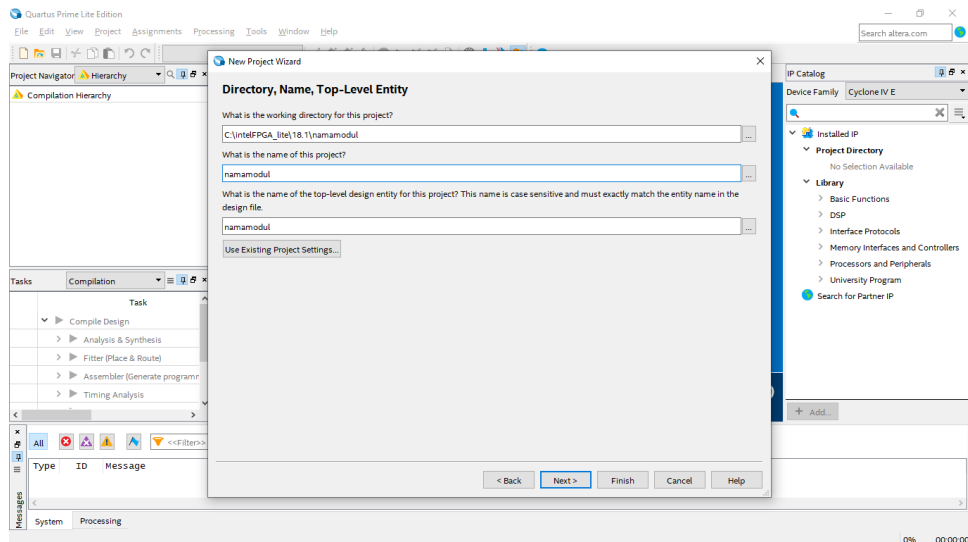
1. Buka software Quartus II di laptop dan klik New Project Wizard



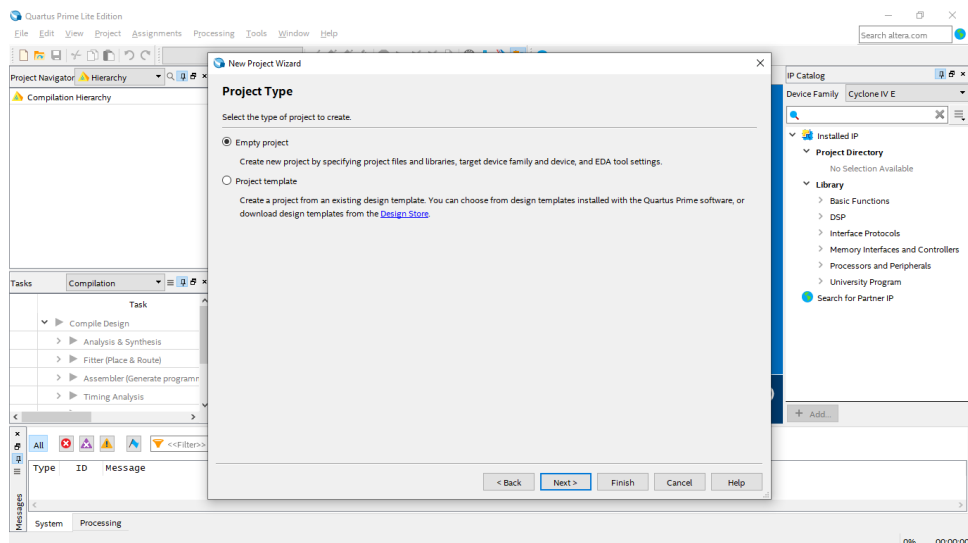
2. Klik Next



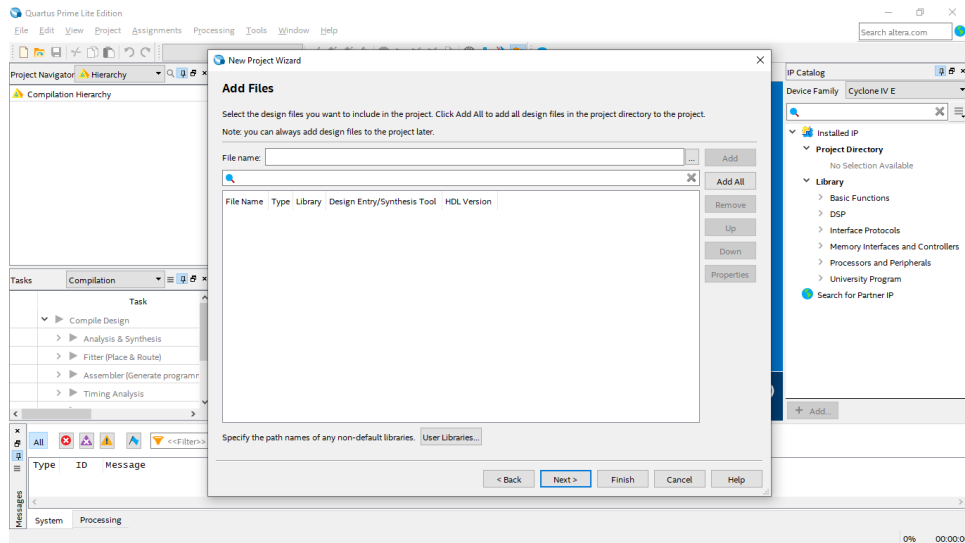
3. Tulis namamodul pada bagian **directory** agar **file project** dapat terkumpul pada 1 **folder** utama. Dengan nama yang sama pada **directory**, tulis juga pada bagian nama **project** dan nama **top-level design entity** (**top-level design entity** bersifat **case sensitive** dan harus persis dengan nama **entity** pada **file desain**), lalu klik **Next**



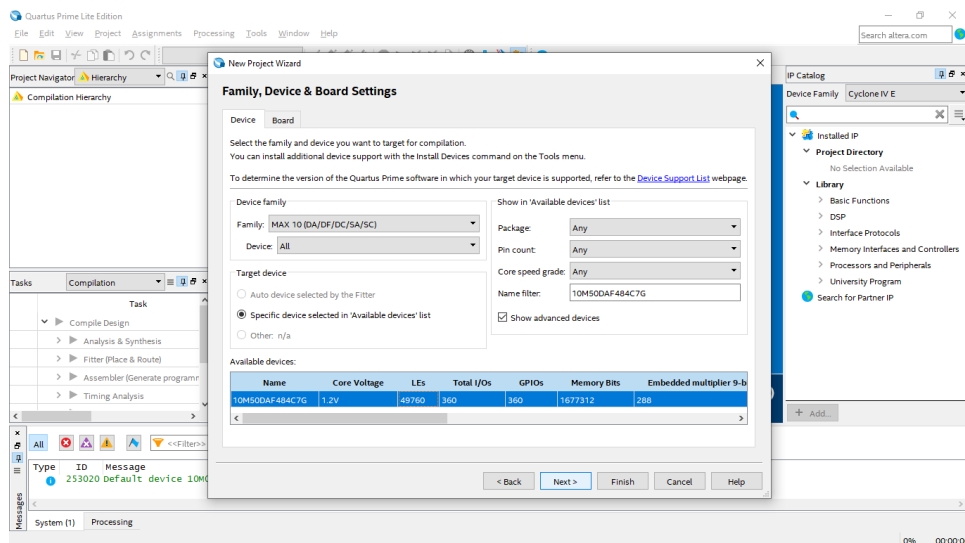
4. Pilih **Empty Project**, kemudian klik **Next** lagi



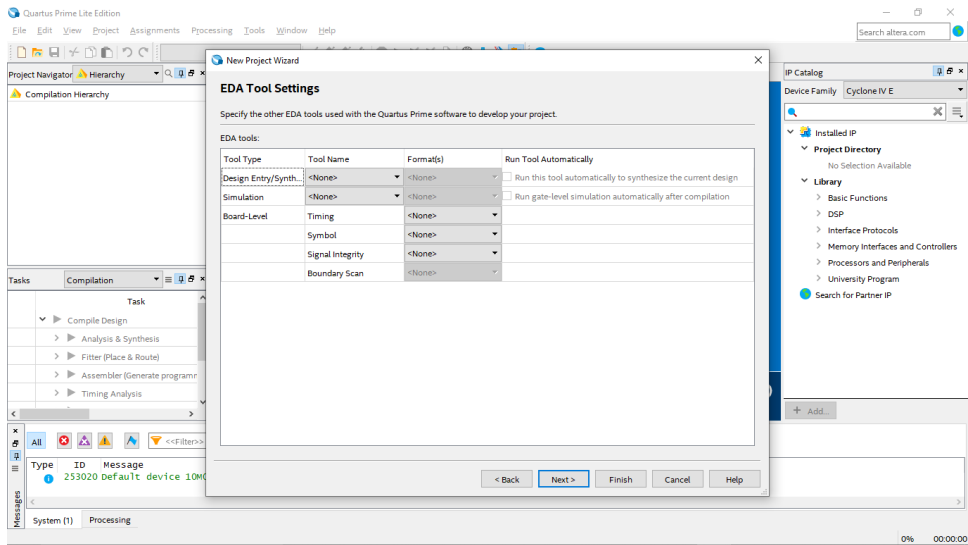
5. Setelah itu, klik Next lagi



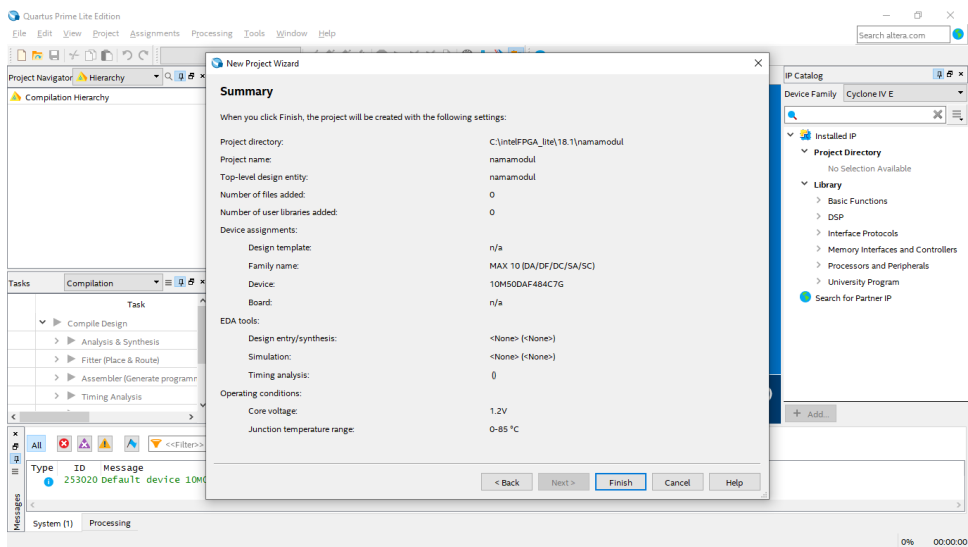
6. Lalu ganti Family ke MAX 10 (DA/DF/DC/SA/SF/SC), kemudian ketikkan di Nama Filter 10M50DAF484C7G, lalu klik Available Device yang tersedia, kemudian klik Next



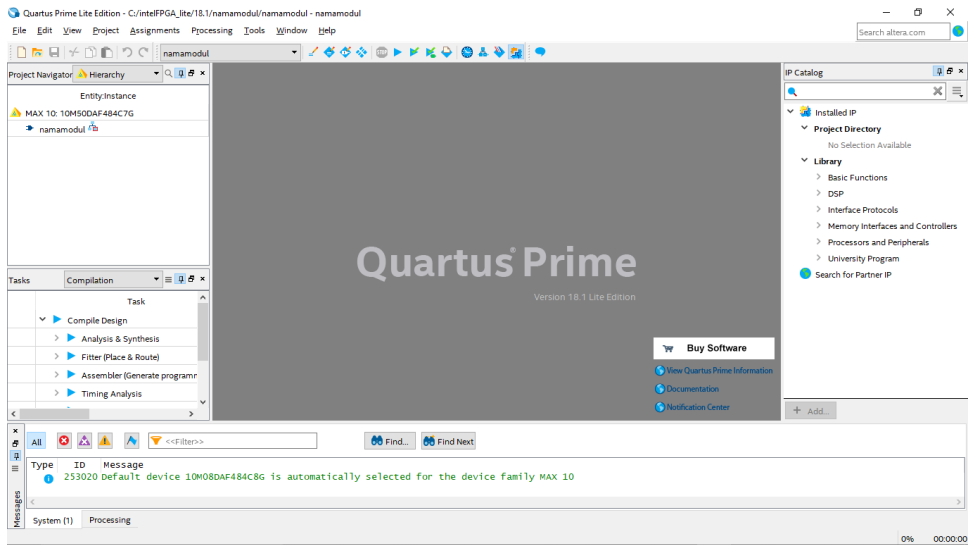
7. Kemudian klik **Next**



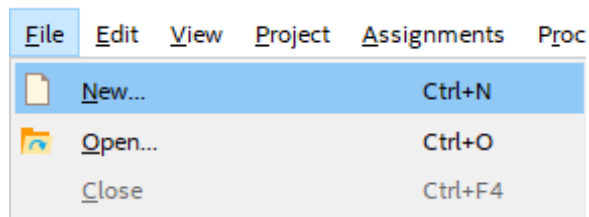
8. Lalu klik **Finish**



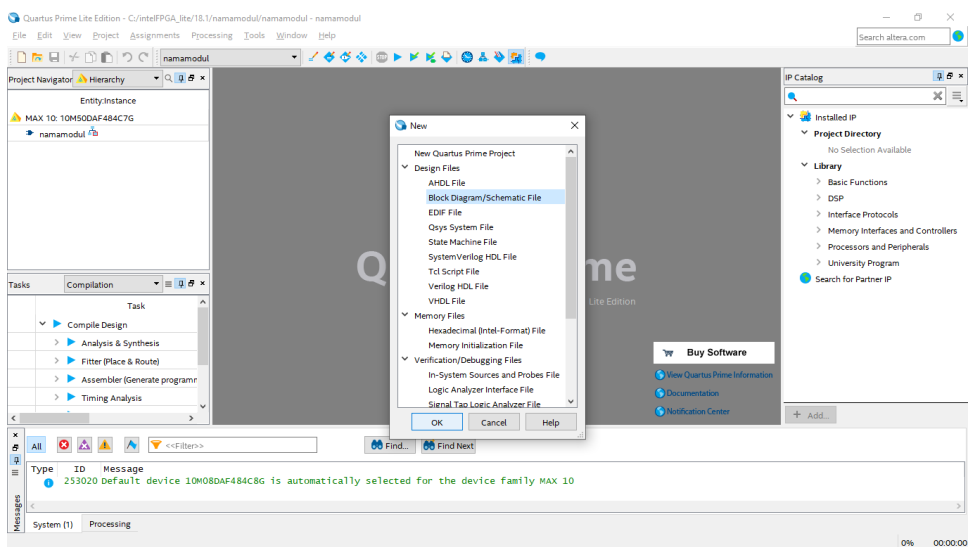
9. Setelah itu akan muncul tampilan awal dari **Project Quartus II** seperti gambar dibawah ini.



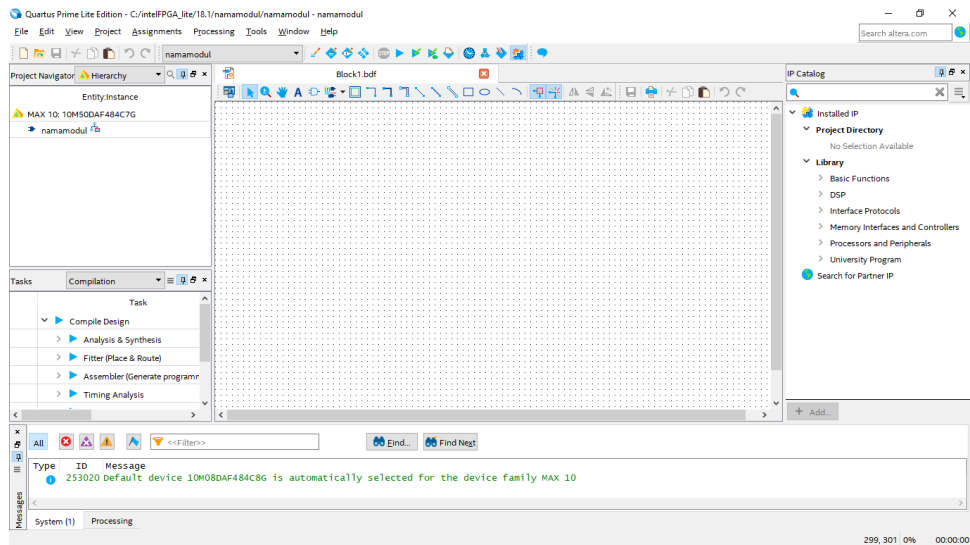
10. Kemudian buat file baru dengan cara klik **File** → **New**.



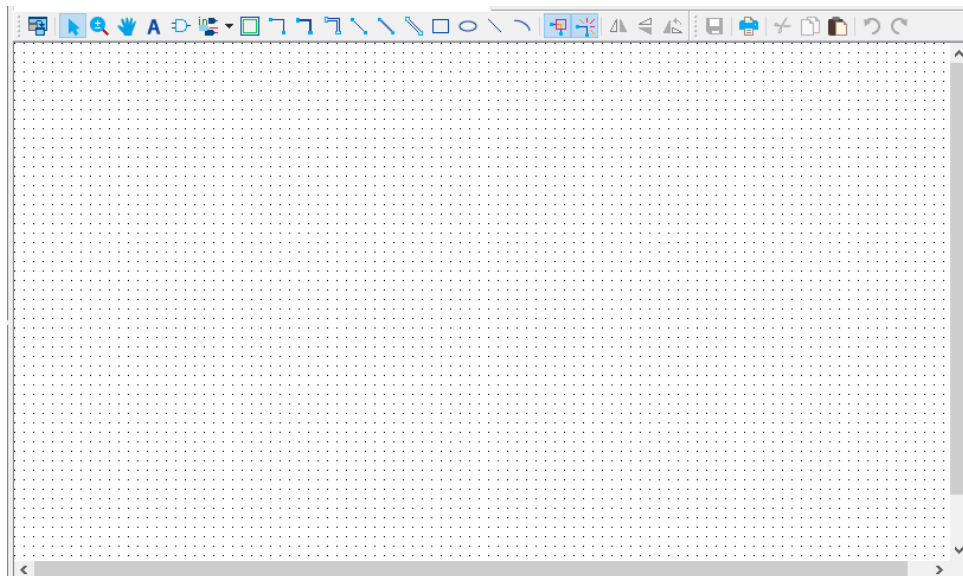
11. Lalu pilih **Block Diagram/Schematic File**, kemudian klik **OK**



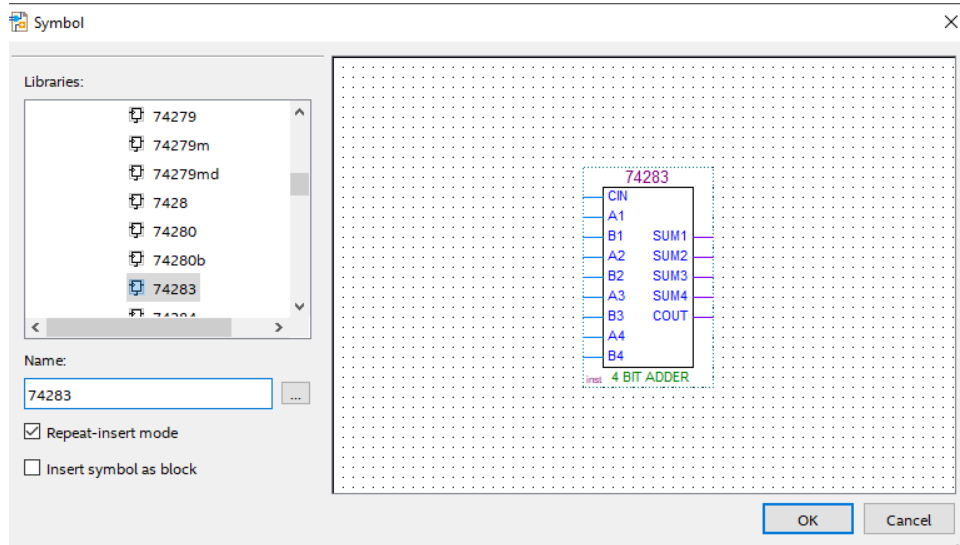
12. Akan muncul **workspace** seperti gambar dibawah ini



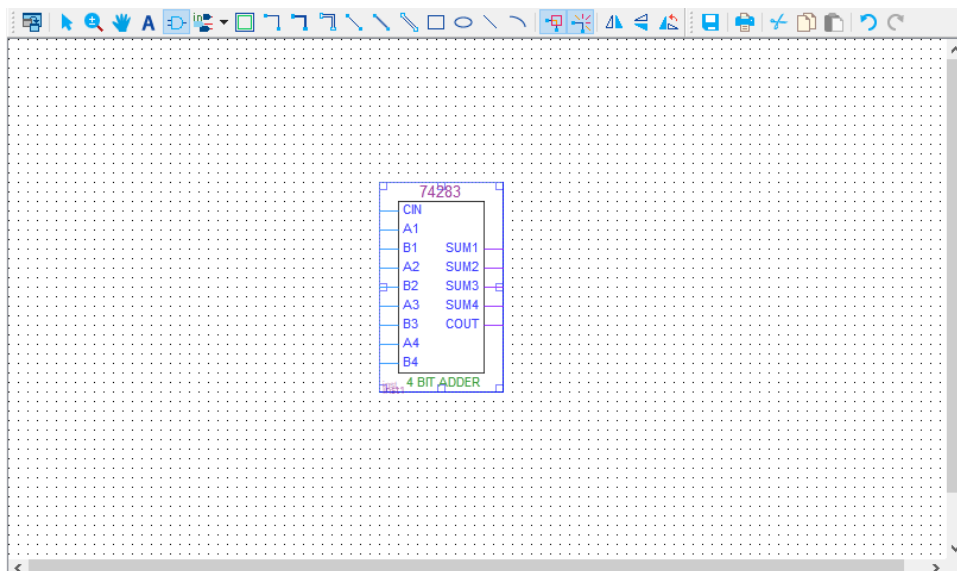
13. Pada langkah praktikum kali ini menggunakan IC TTL 74283. Klik tool **Symbol Tool** seperti gambar dibawah (di mark merah)



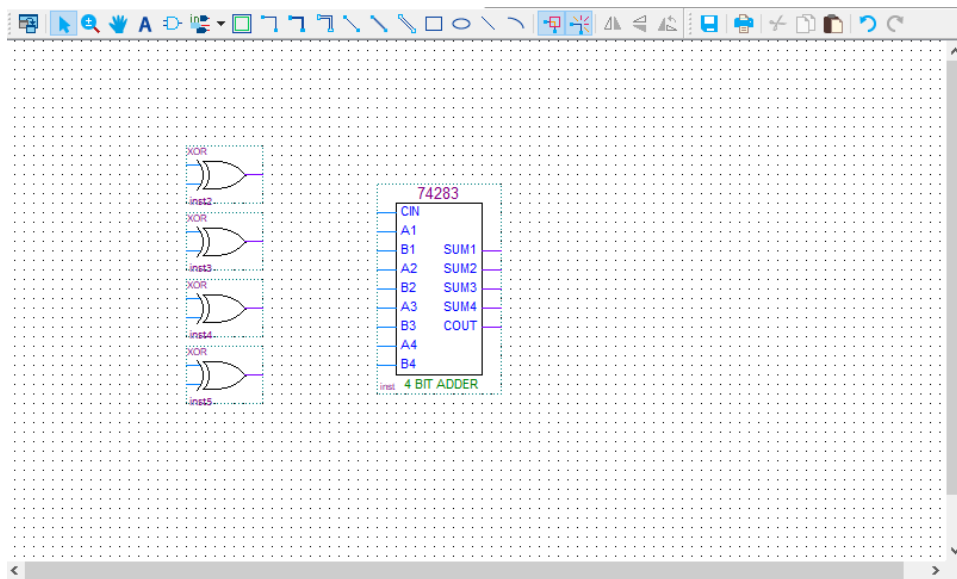
14. Kemudian pada **Libraries**, lalu pada kolom **name** cari IC 74283 → klik **OK**



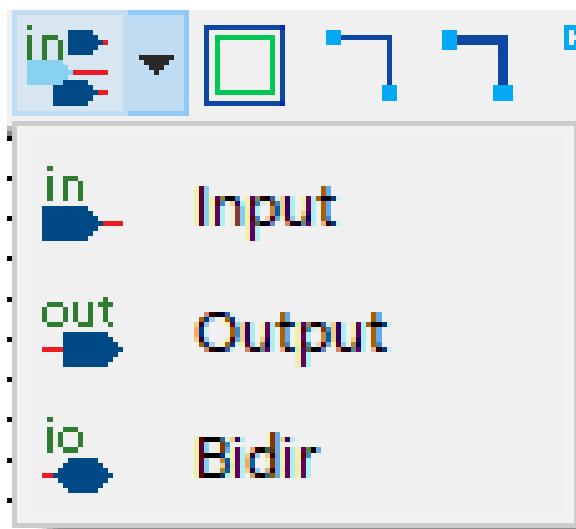
15. Lalu atur posisi tersebut seperti gambar dibawah ini. Lalu untuk melepas komponen yang masih ada pada cursor tekan **ESC** pada **keyboard**



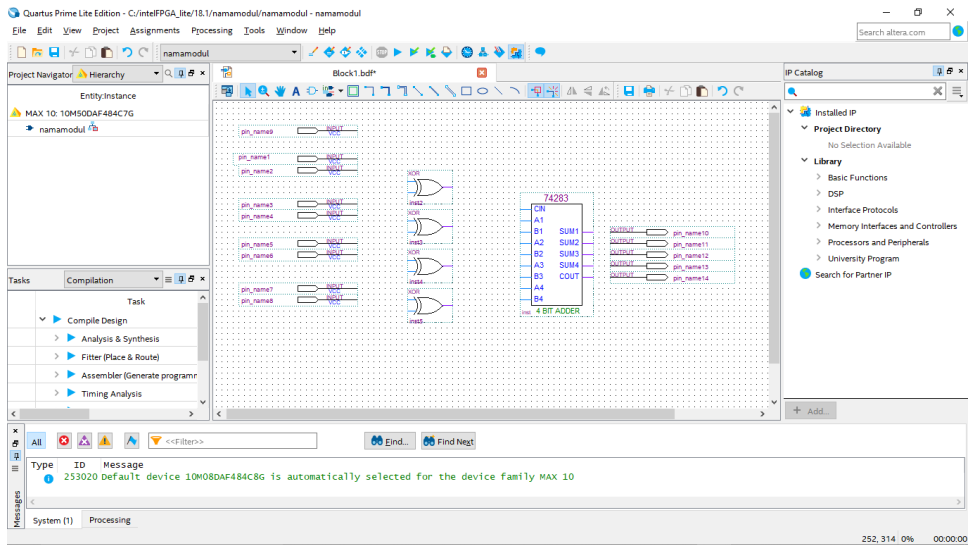
16. Tambahkan gerbang logika **xor** dengan cara klik **tool Symbol Tool**, lalu ketik pada kolom **Name : xor** dan letakkan pada **workspace**.



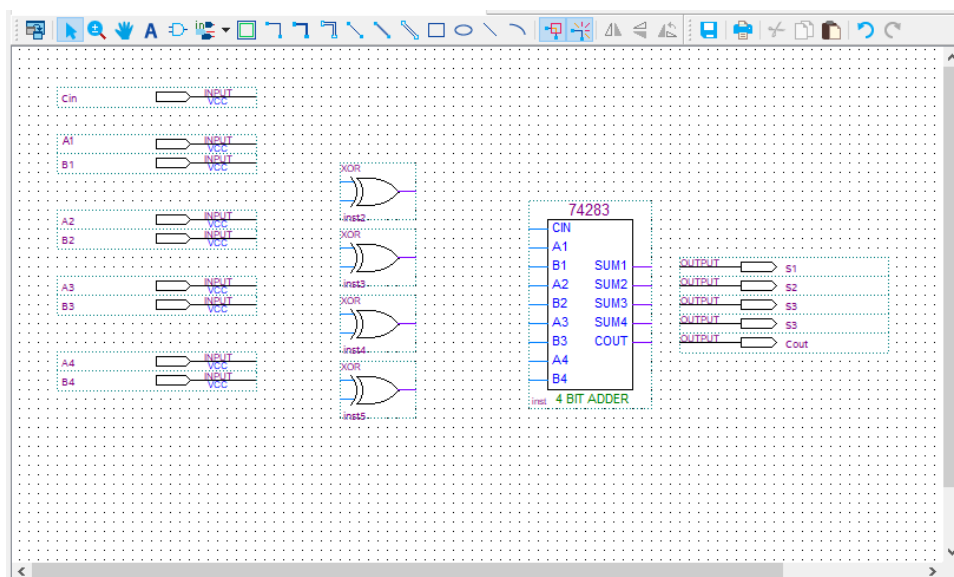
17. Kemudian buat pin **input** dan **output** dengan cara klik **tool Pin Tool** lalu pilih **Input** dan juga **Output**.



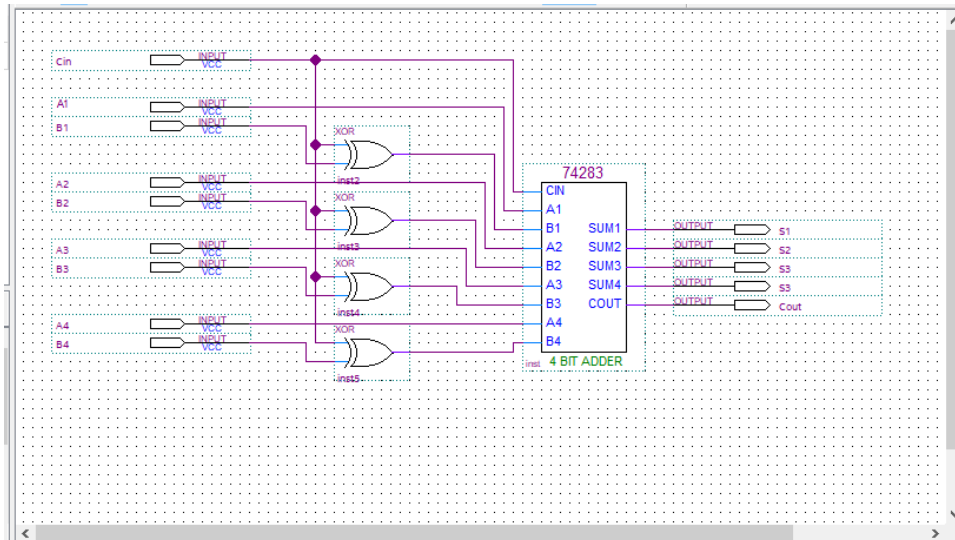
18. Posisikan pin **input** dan pin **output** seperti pada gambar dibawah ini



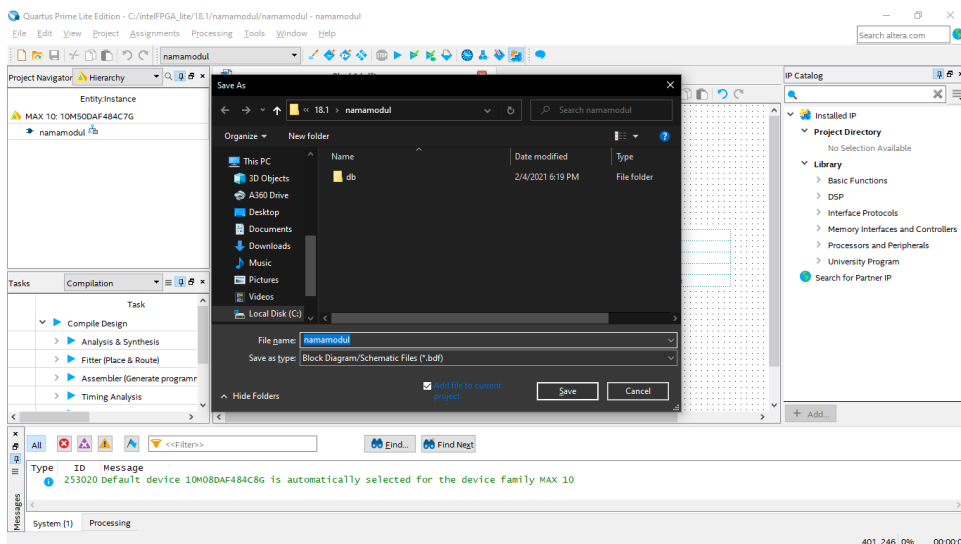
19. Ubah nama pin (**pin name**) dengan nomor sesuai dengan nomor kaki IC yang ada pada IC tersebut. Caranya **double click** pada tulisan pin name dan ubah nama sesuai gambar dibawah ini.



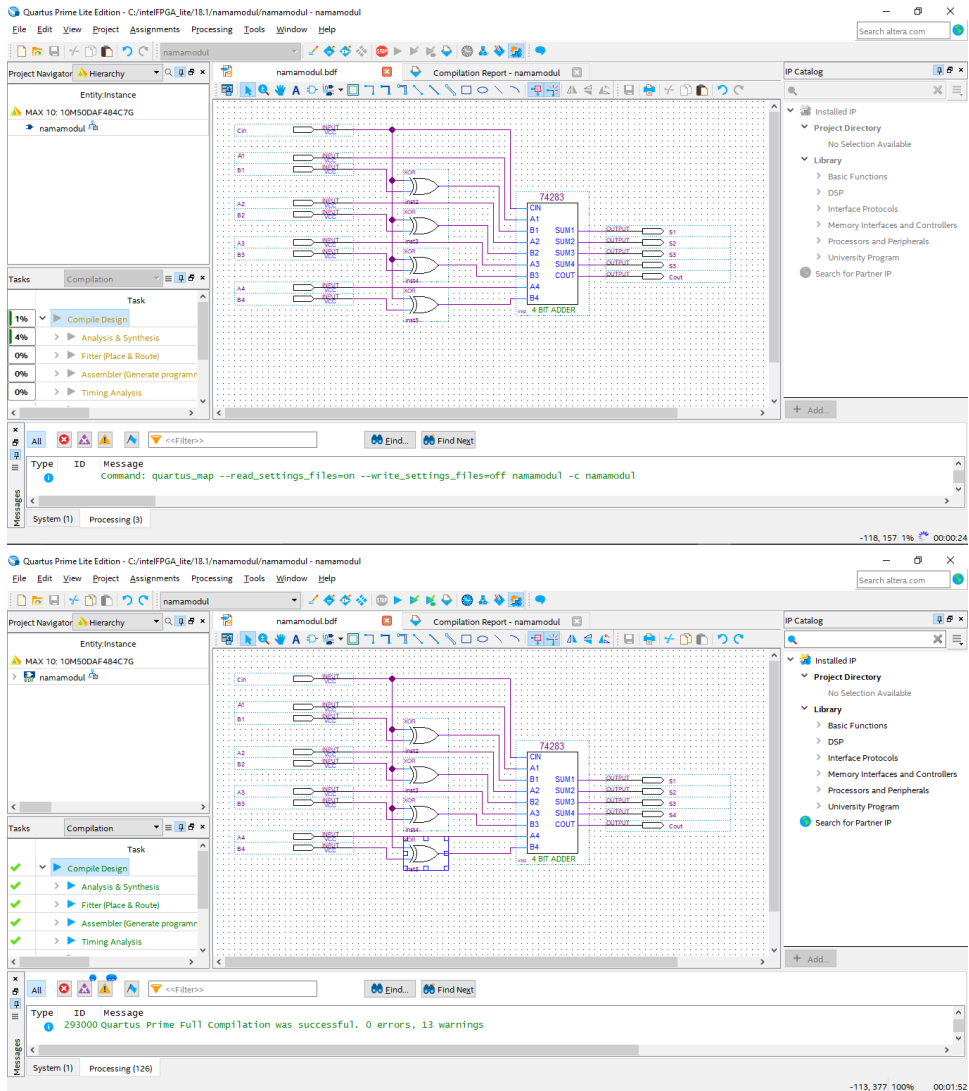
20. Kemudian lakukan **wiring** dengan cara klik **tool Orthogonal Node Tool** atau bisa juga secara langsung dengan klik dan tahan ujung **input/output** lalu sambungkan ke **input** dan **output IC**. Jika terdapat bulatan pada **wiring**, itu artinya **wiring** belum tersambung dengan benar, hapus **wiring** tersebut kemudian lakukan **wiring** ulang.



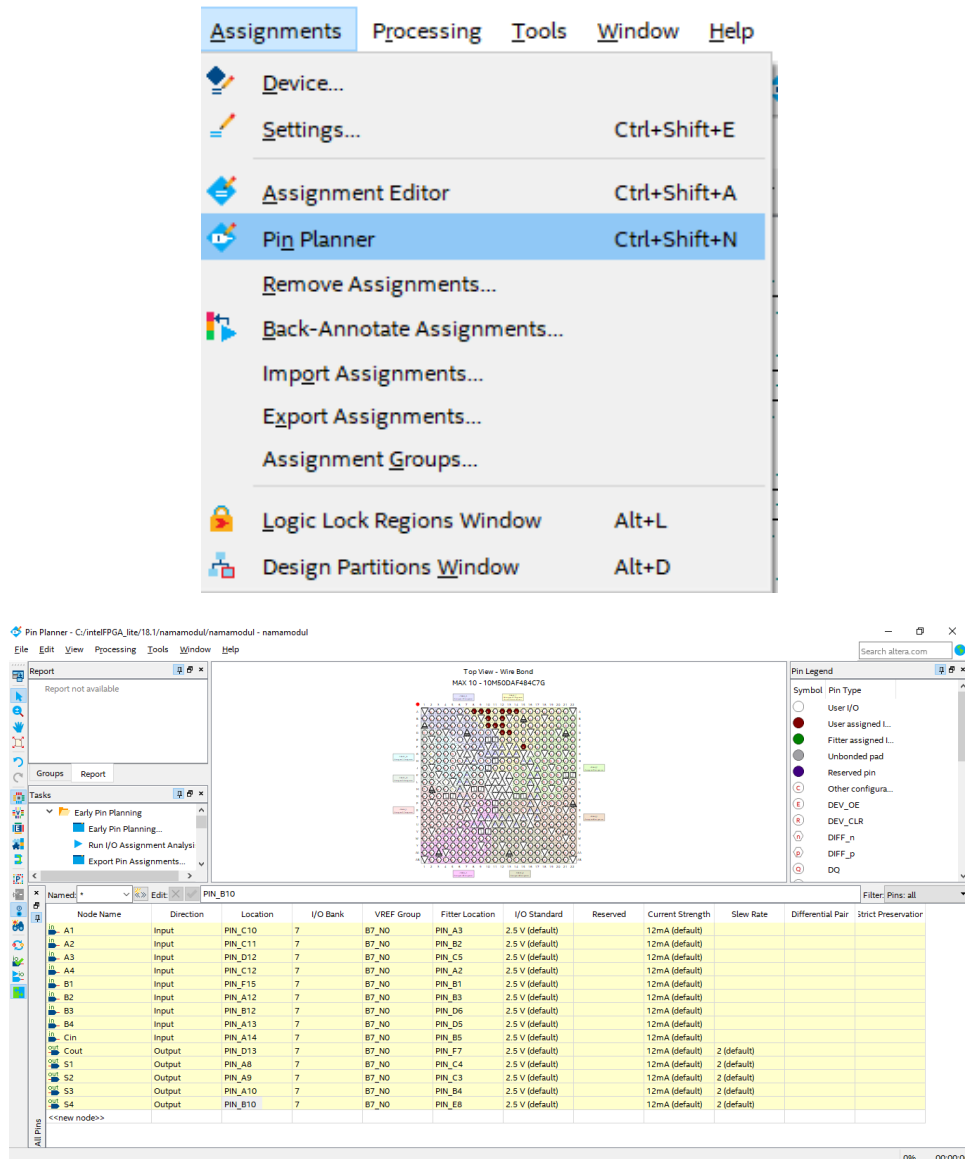
21. Terakhir **Save** dengan cara menekan **Ctrl+S** atau bisa dengan cara klik **File** → **Save**. Lalu tentukan **directory folder** untuk menyimpan **file** tersebut.



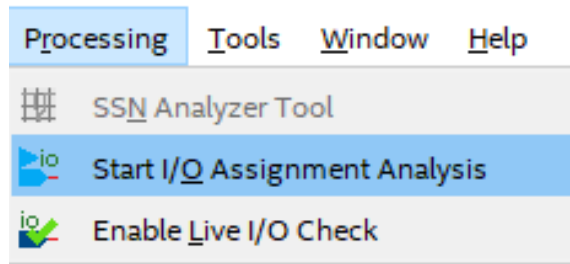
22. Lalu klik kanan pada **Compile Design** → klik **start** → tunggu hingga **success**

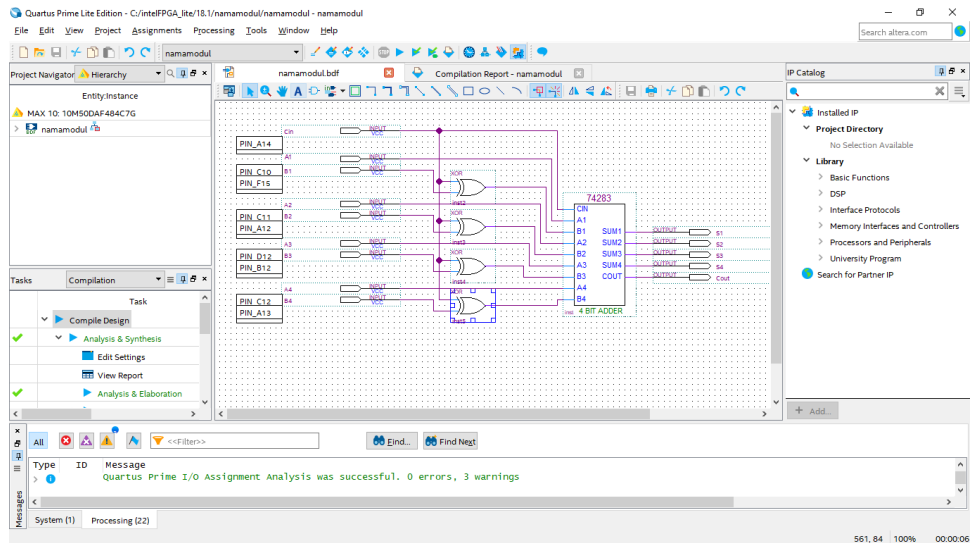


23. Masuk ke menu **Assignment** → **Pin planner** → berikan location pin **Assignment** pada fisik **DE10-lite** sesuai dengan datasheet.



24. Lalu **compile** lagi **Compile Design** → tunggu hingga **success**





6.4 Soal Jurnal

1. Apa yang dimaksud Adder dan Adder-Subtractor?
2. Buatlah rangkaian gerbang logika half adder dengan menggunakan blok diagram menggunakan software Quartus II
3. Tuliskan apa yang telah dilakukan pada praktikum modul 6 menggunakan Bahasa kalian sendiri!

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.

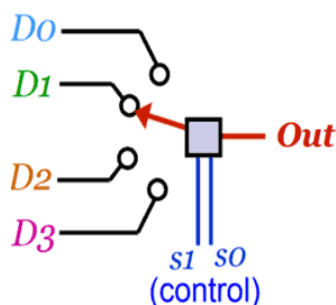
7.2 Dasar Teori Praktikum Modul 7

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



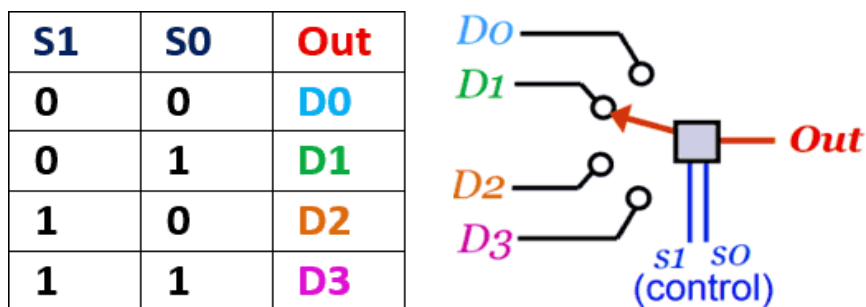
Ini adalah contoh sederhana dari *multiplexer* mekanik. Tetapi, dalam sirkuit elektronik yang melibatkan perpindahan kecepatan tinggi dan transfer data mengharuskan agar dapat memilih input yang dibutuhkan dengan sangat cepat menggunakan sirkuit digital. Sinyal kontrol (S1 dan S0) melakukan hal yang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

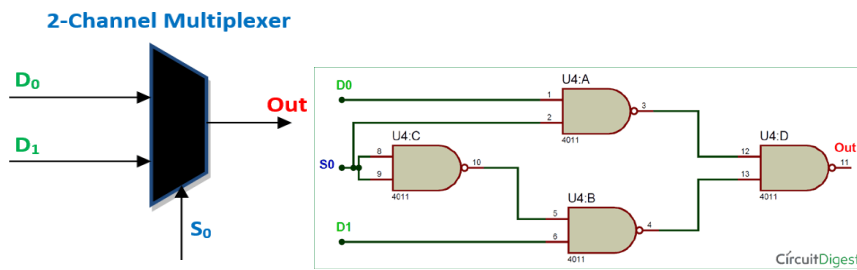


7.2.2 Tipe Multiplexer

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 S_0 ditetapkan logika 0 maka input D_0 akan tercermin pada pin output dan jika S_0 tetap logika 1 maka input D_1 yang akan tercermin pada pin output. Dapat ditunjukkan pada tabel kebenaran dibawah ini:

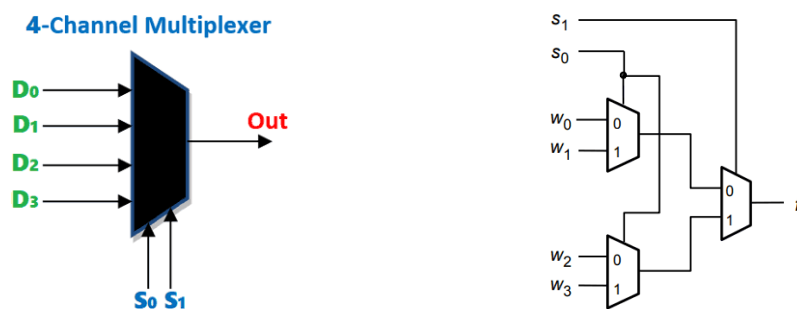
Tabel 7.1 Rangkaian 2-channel multiplexer

S_0	D_0	D_1	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 (S_0) dan kemudian pin kontrol MUX ke tiga digunakan sebagai pembentuk pin kontrol kedua (S_1). Dengan demikian, mendapatkan *multiplexer* dengan 4 *input* (W_0, W_1, W_2, W_3) dan hanya satu *output* (f). Tabel kebenaran untuk *multiplexer* 4:1 ditunjukkan dibawah ini.

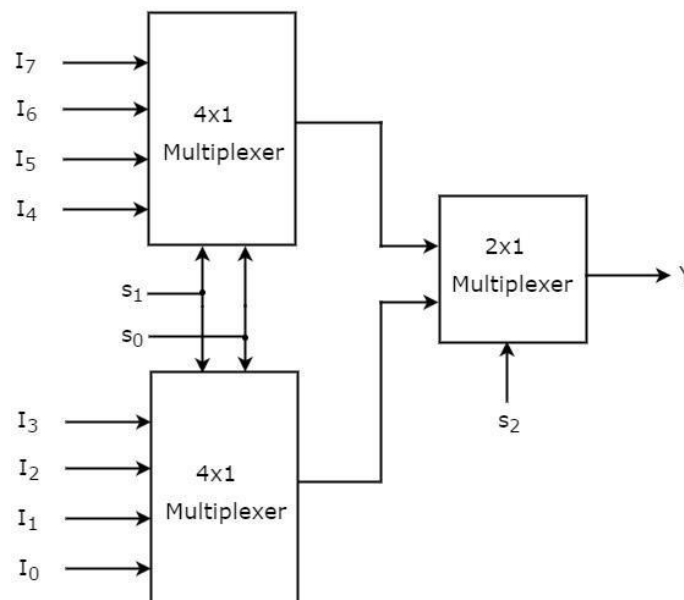
Tabel 7.2 Tabel kebenaran 4-channel multiplexer

S_0	S_1	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 ditunjukkan 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

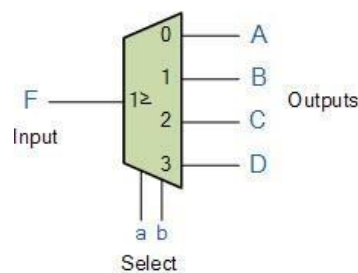
Digunakan untuk transmisi jaringan jarak jauh baik yang menggunakan kabel maupun yang menggunakan media udara seperti wireless atau radio. Sebagai contoh satu helai optic Surabaya–Jakarta bisa dipakai untuk menyalurkan ribuan percakapan pada telepon. Dan juga digunakan untuk remot TV (atau sejenisnya)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

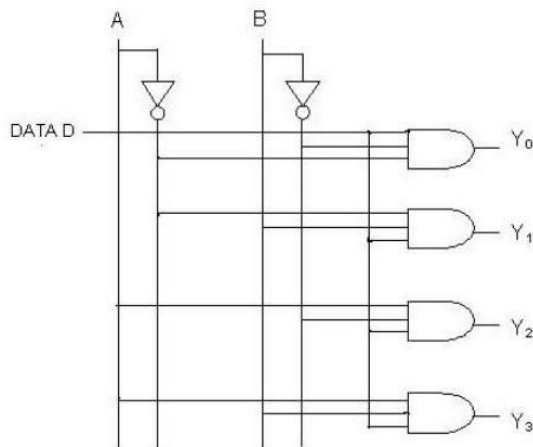


7.2.5 Tipe Demultiplexer

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.

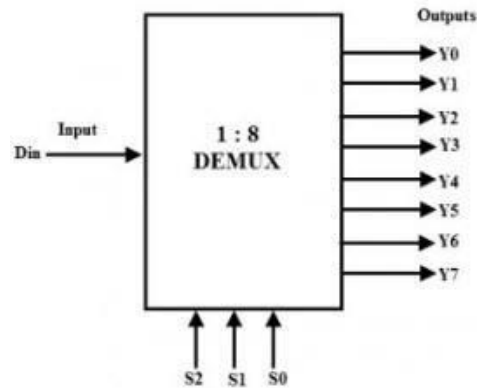
Tabel 7.4 Tabel kebenaran 1 to 4 demultiplexer

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 $S_0S_1S_2 = 000$, maka output yang aktif adalah Y_0 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.

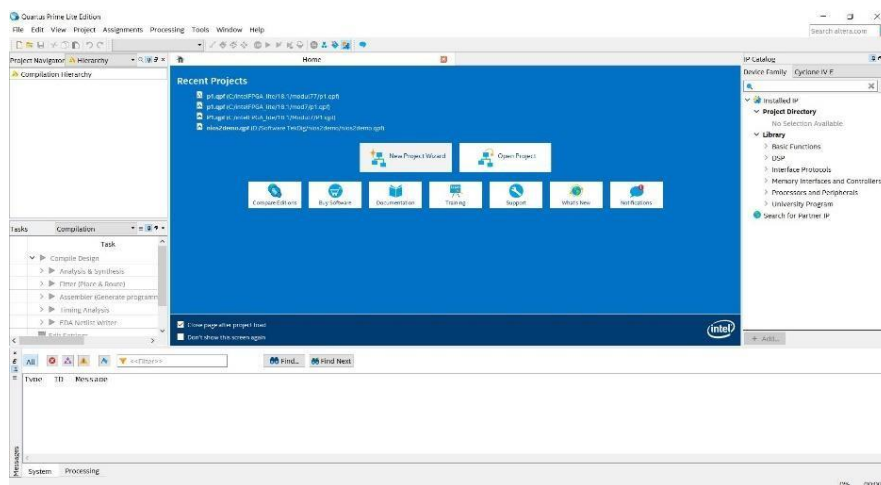
7.3 Lembar Kegiatan Praktikum Modul 7 :

7.3.1 Alat dan Bahan

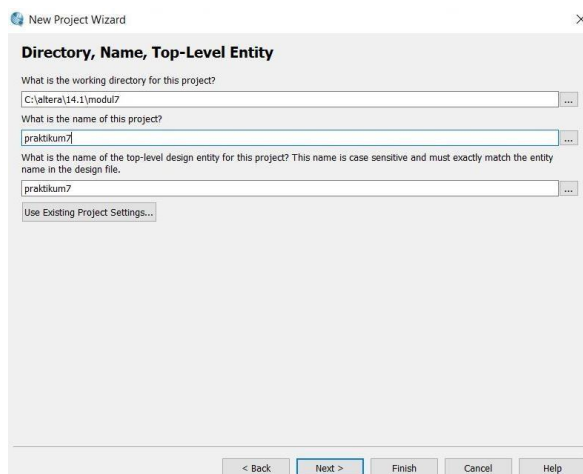
- Laptop yang telah terinstal *software Quartus 18*
- Mouse

7.3.2 Langkah percobaan modul 7 (4-channel multiplexer)

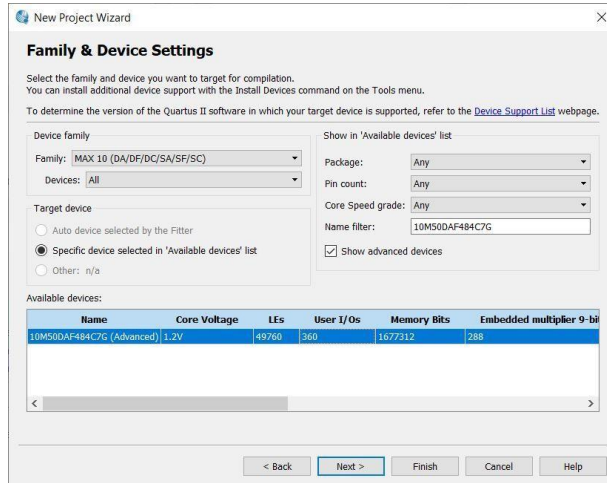
- Hidupkan Laptop
- Buka aplikasi quartus
- Pilih **New Project Wizard** → next



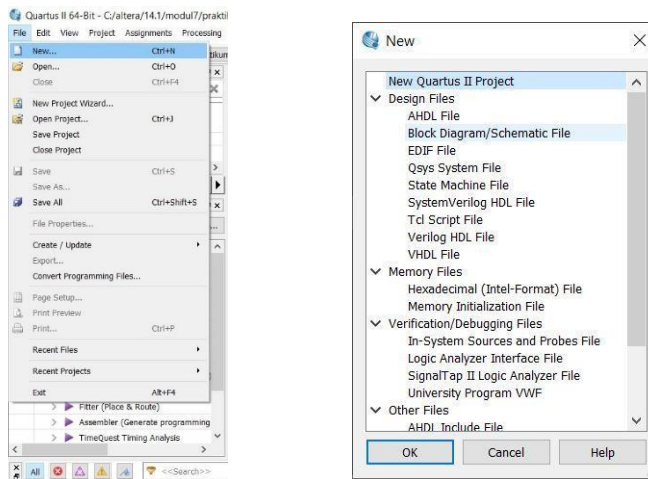
- Menentukan **directory** dan nama **project** next → next → next



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**

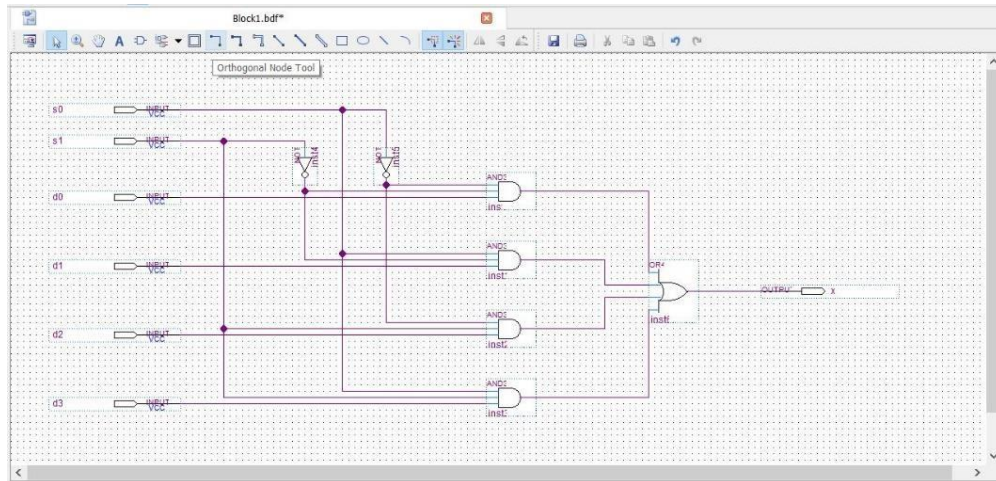


6. Pilih menu **file** → **new** → pilih **Block Diagram/Schematic File** → **ok**

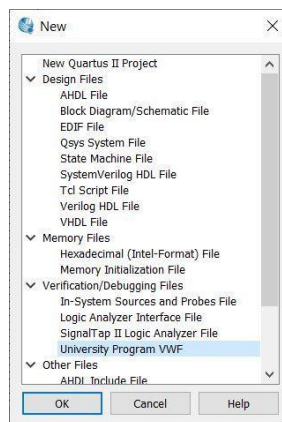


Modul Praktikum

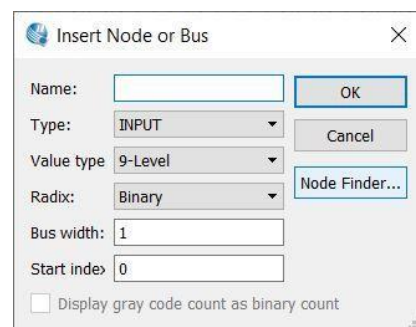
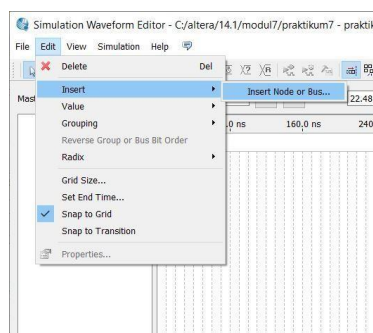
7. Membuat rangkaian multiplexer, klik **Symbol Tool** → klik **tanda panah** → **primitives** → **logic** → **pilih gerbang logika yang ada**. Dan buat inputan serta outputan dari rangkaian tersebut, klik **Pin Tool** → **pilih input serta output sesuaikan dengan rangkaian tersebut**, serta rename pin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



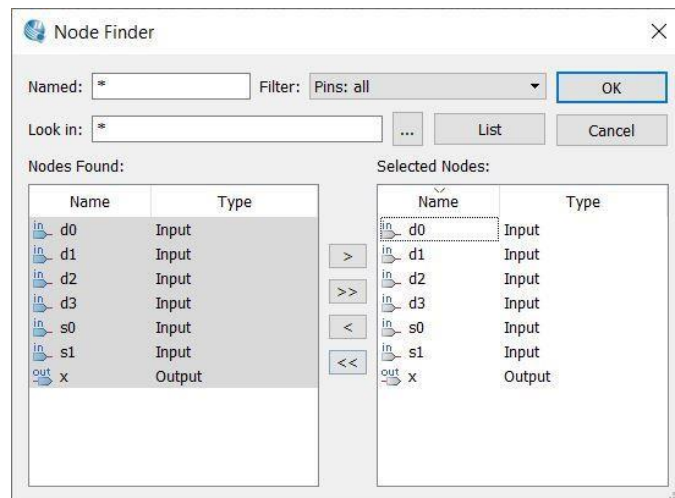
8. Setelah itu, klik **Compile design**. Tunggu hingga berhasil
9. Pilih menu **file** → **new** → pilih **Univerty Program VWF** → ok



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



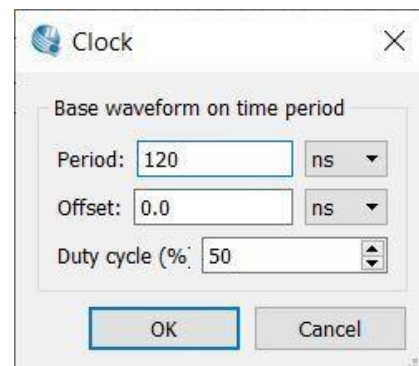
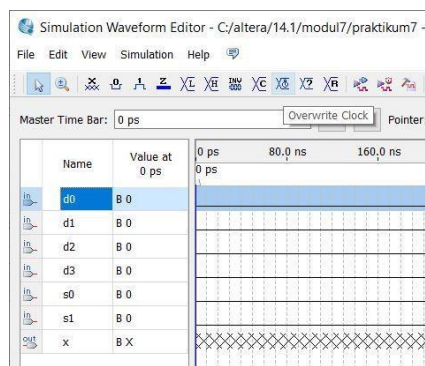
11. Klik **List** → klik “>>” → **ok** → **ok**



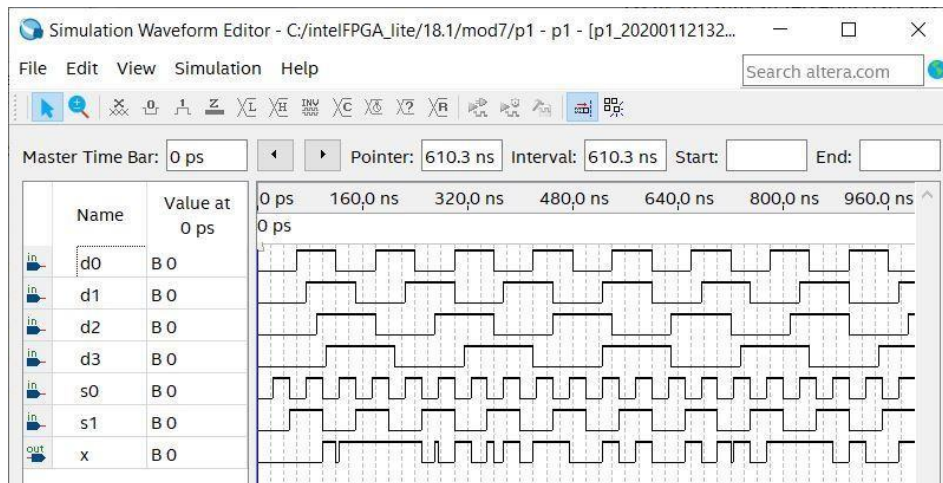
12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :

- 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%

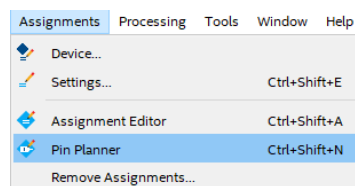
Dengan **cara klik kiri pada D0** → klik menu **Edit** → **Value** → **Overwrite Clock** → **masukan clock periode sesuai dengan masukan tersebut**. lakukan langkah tersebut dari D0 sampai S1.



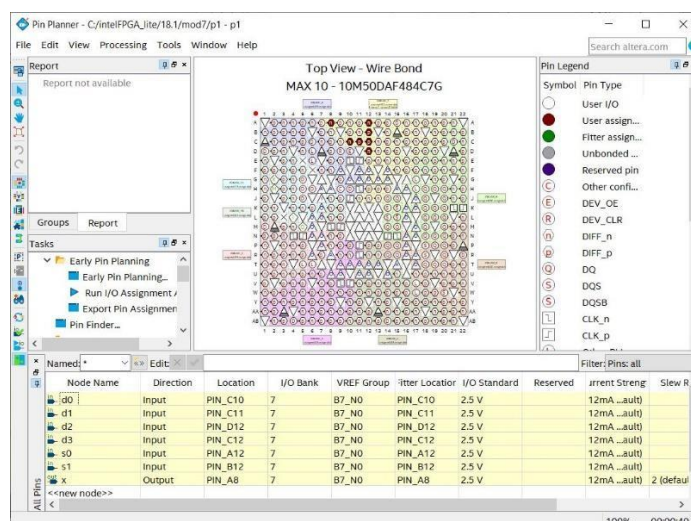
13. Lakukan simulasi dengan cara klik pada menu **Simulation** → **Run Timing Simulation** → tunggu hingga selesai.



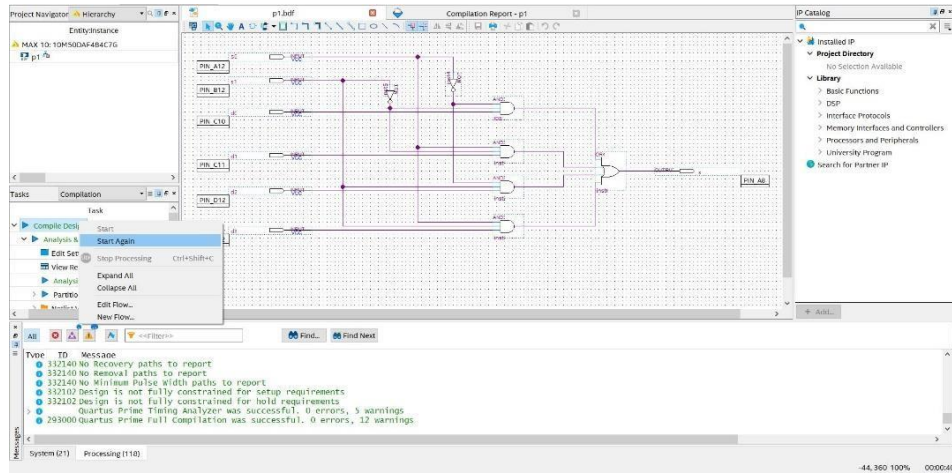
14. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu **assignments** → **pin planner**



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

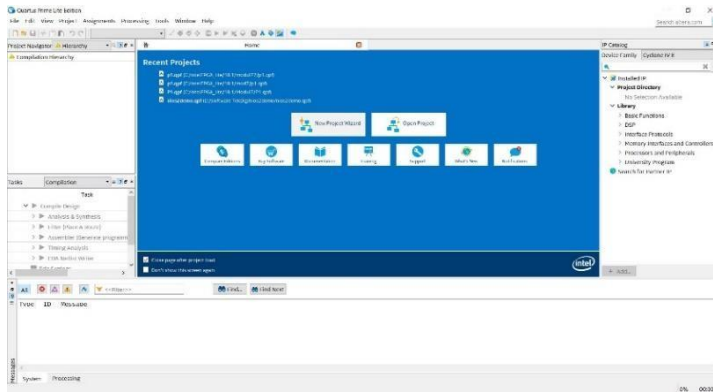


16. Compile design. Tunggu hingga selesai

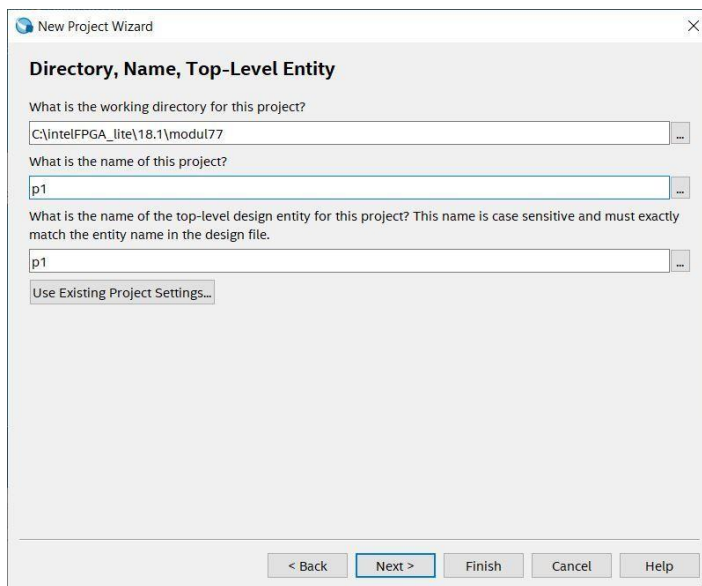


7.3.3 Langkah percobaan modul 7 (1 to 4 demultiplexer)

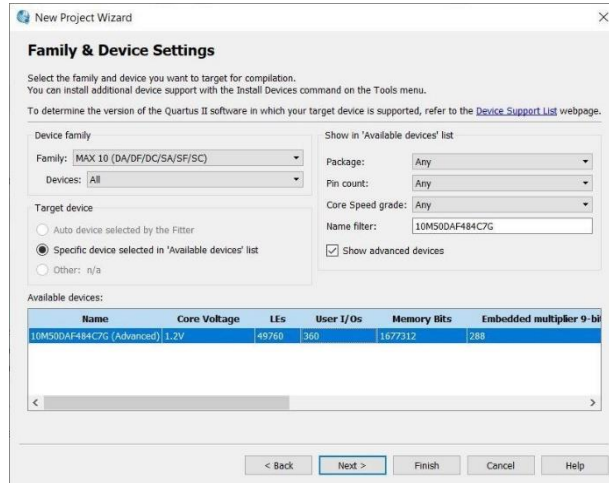
1. Hidupkan Laptop
2. Buka aplikasi quartus
3. Pilih **New Project Wizard** → next



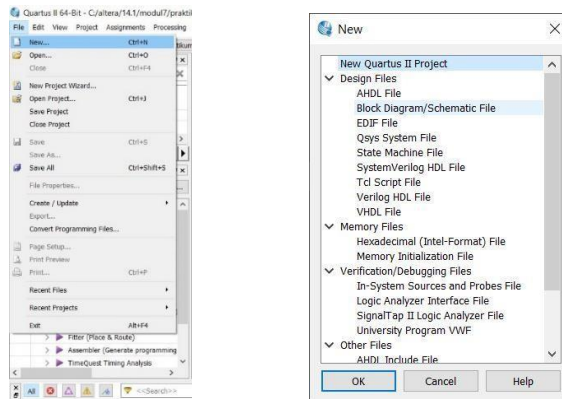
4. Menentukan **directory** dan nama **project** → next → next → next



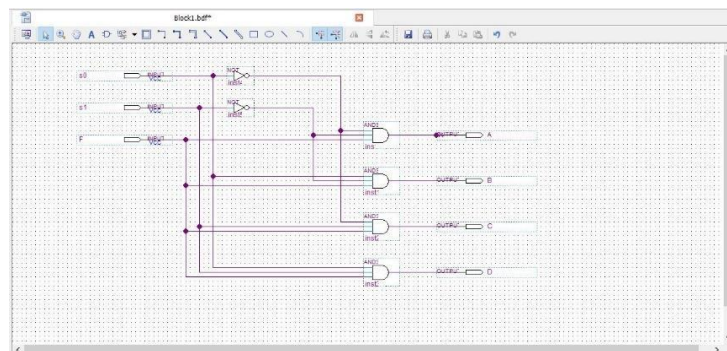
5. Menentukan **device family** dengan **MAX 10 (DA/DF/DC/SA/SC)** dan name filter dengan **10M50DAF484C7G** → klik **10M50DAF484C7G** pada **available device** → **next** → **next** → **finish**



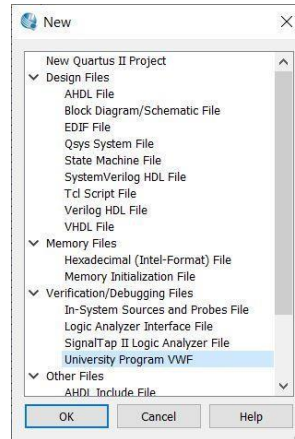
6. Pilih menu **file** → **new** → pilih **Block Diagram/Schematic File** → **ok**



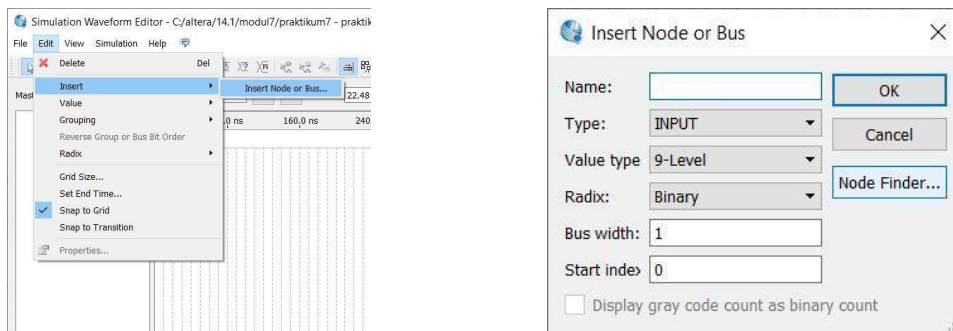
7. Membuat rangkaian multiplexer, klik **Symbol Tool** → klik **tanda panah** → **primitives** → **logic** → pilih gerbang logika yang ada. Dan buat inputan serta outputan dari rangkaian tersebut, klik **Pin Tool** → pilih **input serta output** sesuaikan dengan rangkaian tersebut, serta rename pin input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



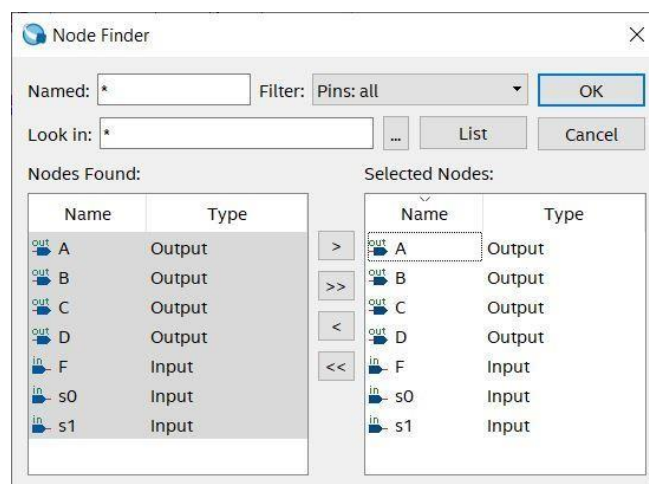
8. Setelah itu, klik **Compile design**. Tunggu hingga berhasil
9. Pilih menu **file** → **new** → pilih **Univerty Program VWF** → **ok**



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

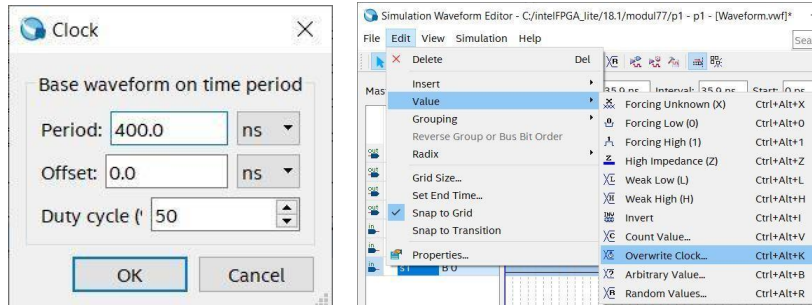


11. Lalu klik **List** → klik ">>" → **ok** → **ok**



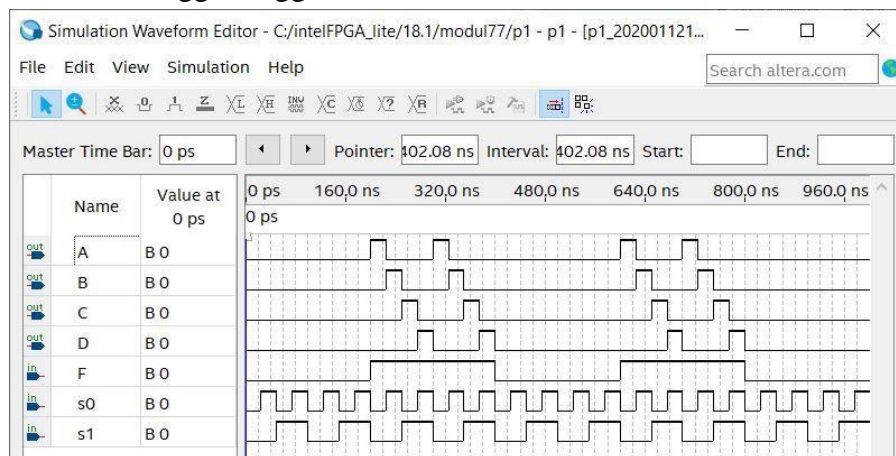
12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :

- 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%

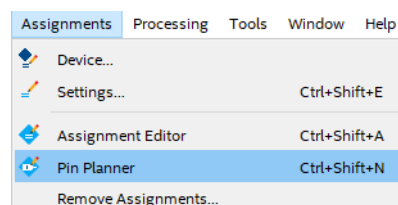


Dengan cara **klik kiri pada D0** → klik menu **Edit** → **Value** → **Overwrite Clock** → **masukan clock periode sesuai dengan masukan tersebut**. lakukan langkah tersebut dari D0 sampai S1.

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.**

The screenshot shows the Pin Planner interface for the MAX 10 - 10M50DAF484C7G device. The top view wire bond diagram shows the physical layout of the device with pins assigned to various nodes. The Pin Legend on the right lists various pin types and their symbols. The table below shows the pin assignment details.

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate
A	Output	PIN_A8	7	B7_NO	PIN_A8	2.5 V		12mA ..ault	2 (default)
B	Output	PIN_A9	7	B7_NO	PIN_A9	2.5 V		12mA ..ault	2 (default)
C	Output	PIN_A10	7	B7_NO	PIN_A10	2.5 V		12mA ..ault	2 (default)
D	Output	PIN_B10	7	B7_NO	PIN_B10	2.5 V		12mA ..ault	2 (default)
F	Input	PIN_D12	7	B7_NO	PIN_D12	2.5 V		12mA ..ault	
s0	Input	PIN_C10	7	B7_NO	PIN_C10	2.5 V		12mA ..ault	
s1	Input	PIN_C11	7	B7_NO	PIN_C11	2.5 V		12mA ..ault	

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

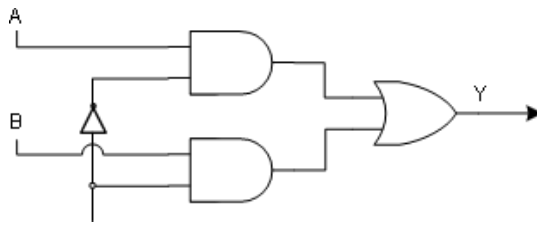
The screenshot shows the Quartus Prime software interface during the compilation process. The Project Navigator on the left shows the project hierarchy. The main window displays a logic diagram with various components and connections. The Messages window at the bottom shows the compilation results, indicating that the compilation was successful with 0 errors and 12 warnings.

Messages:

- 332140 No recovery paths to report
- 332140 No removal paths to report
- 332140 No minimum pulse width paths to report
- 332102 design is not fully constrained for setup requirements
- Quartus Prime Fitting Analyzer was successful. 0 errors, 5 warnings
- 293900 Quartus Prime Full Compilation was successful. 0 errors, 12 warnings

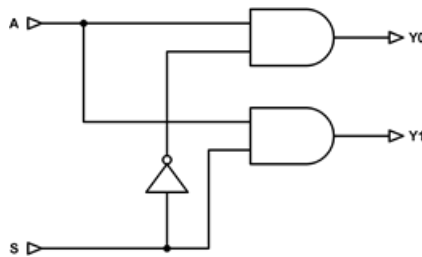
7.4 Soal Jurnal

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!

MODUL 8

SIMULASI RANGKAIAN ENCODER DAN DECODER

8.1 Tujuan Praktikum Modul 8

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Dapat mengetahui dan memahami konsep dasar dari rangkaian *encoder* dan *decoder*
2. Dapat membuat rangkaian *encoder* dan *decoder* dengan menggunakan rangkaian skematik pada software Quartus

8.2 Dasar Teori Praktikum Modul 8

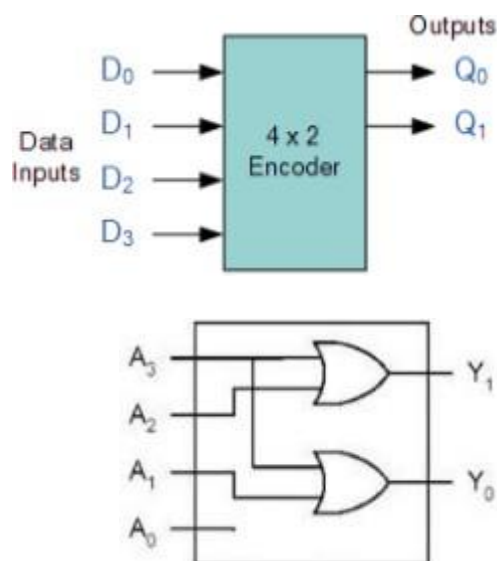
8.2.1 Encoder

a. Definisi

Encoder adalah sebuah rangkaian yang dapat berfungsi mengkonversi suatu data kedalam bentuk data baru. Encoder dibutuhkan untuk mengkonversi data dari suatu sistem bilangan ke sistem bilangan lainnya. Prinsip kerja Encoder adalah mengkonversi suatu data agar data tersebut dapat diterima oleh *receiver* dalam keadaan utuh. Dimana pada bagian penerima terdapat decoder yang dapat mengambil data yang telah dikonversi oleh Encoder. Jadi, cara kerja *Encoder* adalah kebalikan dari *Decoder*.

Contoh *Encoder* :

Gambar 8. 1 Encoder



Tabel 8.1 Tabel kebenaran encoder

Inputs				Outputs	
D ₃	D ₂	D ₁	D ₀	Q ₁	Q ₀
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

8.2.2 Decoder

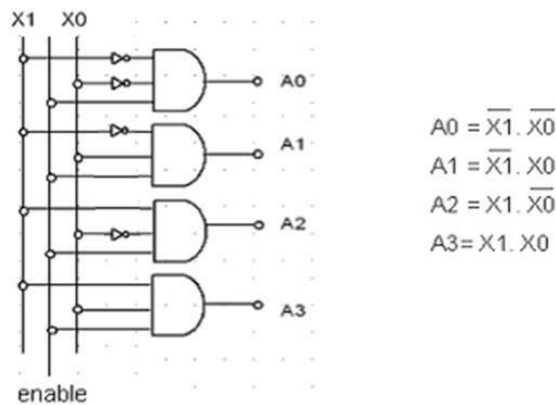
a. Definisi

Decoder adalah alat yang di gunakan untuk dapat mengembalikan proses *encoding* sehingga dapat terlihat atau menerima informasi aslinya. Pengertian *Decoder* juga dapat di artikan sebagai rangkaian logika yang di tugaskan untuk menerima input biner dan mengaktifkan salah satu *outputnya* sesuai dengan urutan biner tersebut.

b. Konsep Dasar

Fungsi *Decoder* adalah untuk memudahkan dalam menyalakan *seven segmen*. *Output* dari *decoder* maksimum adalah 2^n . Jadi dapat dibentuk n -to- 2^n decoder. Beberapa rangkaian *decoder* yang sering dijumpai saat ini adalah *decoder* jenis 3×8 (*3 bit input dan 8 output line*), *decoder* jenis 4×16 , *decoder* jenis *BCD to Decimal* (*4 bit input dan 10 output line*) dan *decoder* jenis *BCD to 7 segmen* (*4 bit input dan 8 output line*). Khusus untuk pengertian *decoder* jenis *BCD to seven segmen* mempunyai prinsip kerja yang berbeda dengan *decoder* lainnya, di mana kombinasi setiap inputnya dapat mengaktifkan beberapa *output linenya*. Salah satu jenis IC *decoder* yang umum di pakai adalah 74138, karena IC ini mempunyai 3 *input* biner dan 8 *output* line, di mana nilai *output* adalah 1 untuk salah satu dari ke 8 jenis kombinasi inputnya. Jika diperhatikan, pengertian *decoder* sangat mirip dengan demultiplexer dengan pengecualian yaitu *decoder* tidak mempunyai data *input*. Sehingga *input* hanya di gunakan sebagai data *control*. *Decoder* dapat di bentuk dari susunan gerbang logika dasar atau menggunakan IC yang banya dijual di pasaran, seperti *decoder* 74LS48, 74LS154, 74LS138, 74LS155, dan sebagainya.

Gambar 8. 2 Decoder



Tabel 8. 2 Tabel kebenaran decoder

X ₁	X ₀	A ₀	A ₁	A ₂	A ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Decoder 2 to 4 bekerja dikendalikan keluarannya oleh variasi X0 dan X1. Keluaran dari a0, a1, a2, dan a3 akan aktif secara bergantian tergantung dari kondisi X0 dan X1. Pada kondisi masukan X0 = 0 dan X1 = 0, maka gerbang AND pertama mendapatkan masukan “**high**”, maka kondisi keluaran A0 = 1 sementara keluaran yang lain “0”. Apabila X0 = 0 dan X1 = 1, maka hanya gerbang AND ke dua akan “1” sementara yang lain “0” dan seterusnya.

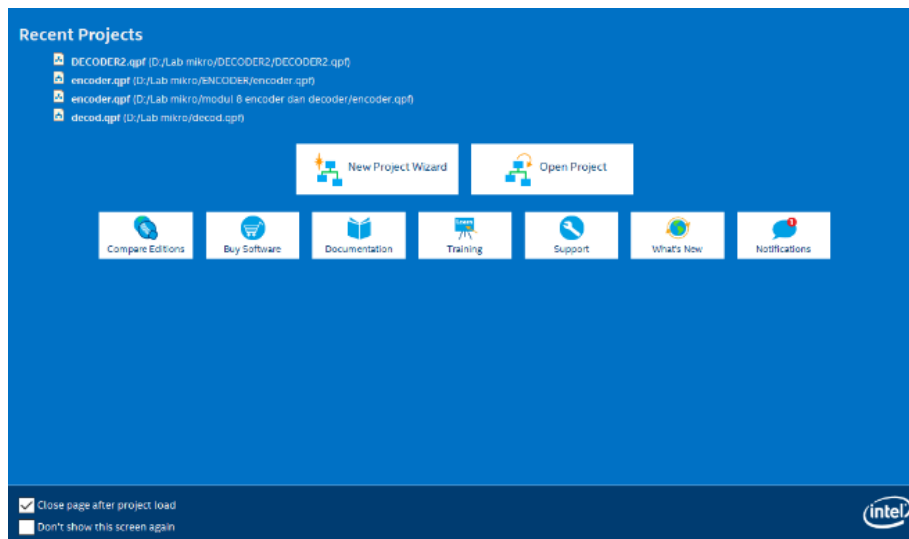
8.3 Lembar Kegiatan Praktikum Modul 8

8.3.1 Alat dan Bahan

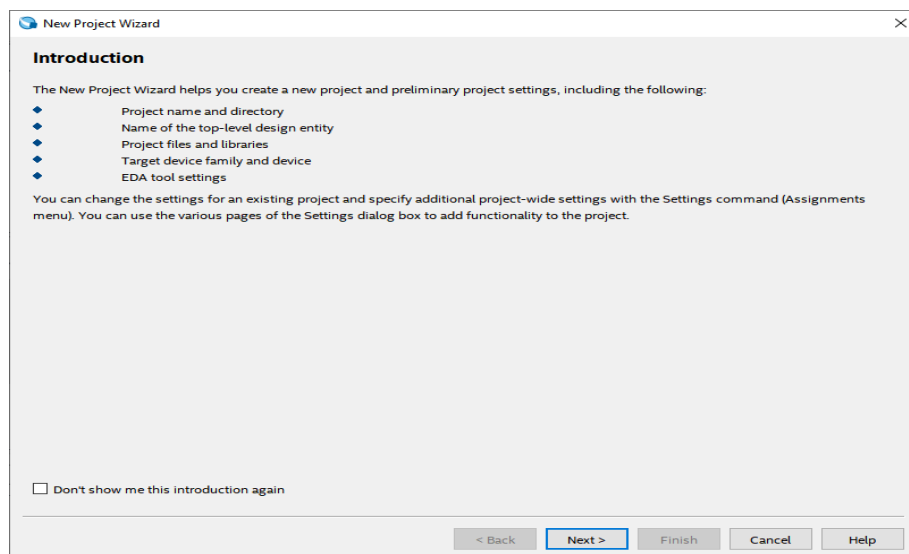
- a. Laptop yang sudah terinstall Software Quartus Prime Lite
- b. Mouse

8.3.2 Langkah Praktikum Modul 8 (Rangkaian *Decoder*)

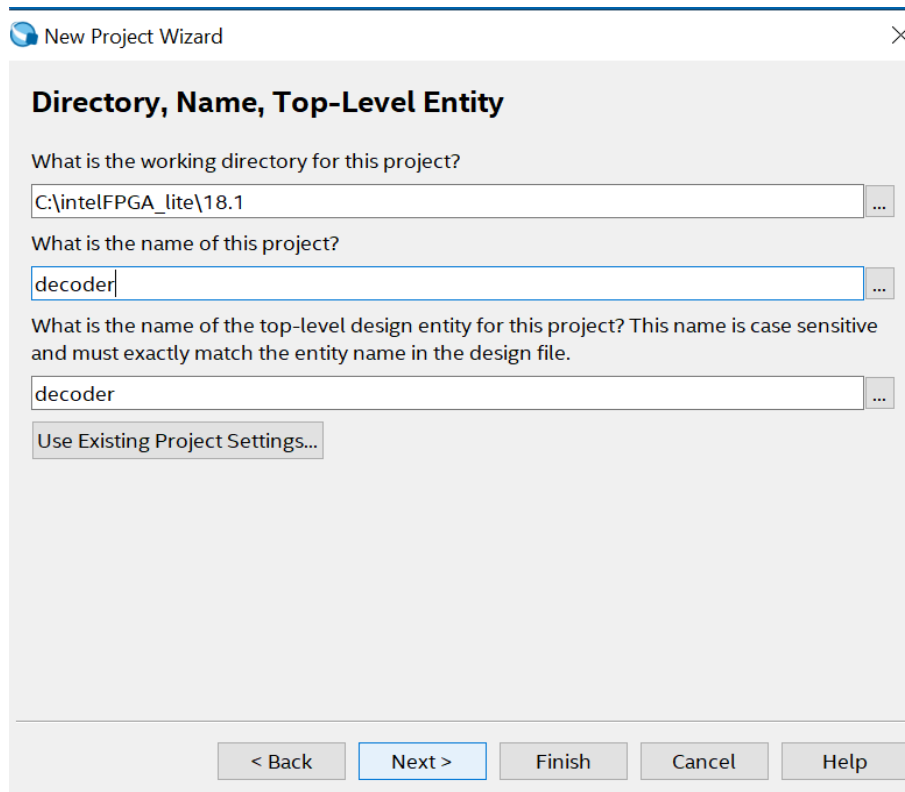
1. Buka software Quartus lite di laptop dan klik **New Project Wizard**



2. Klik **Next**

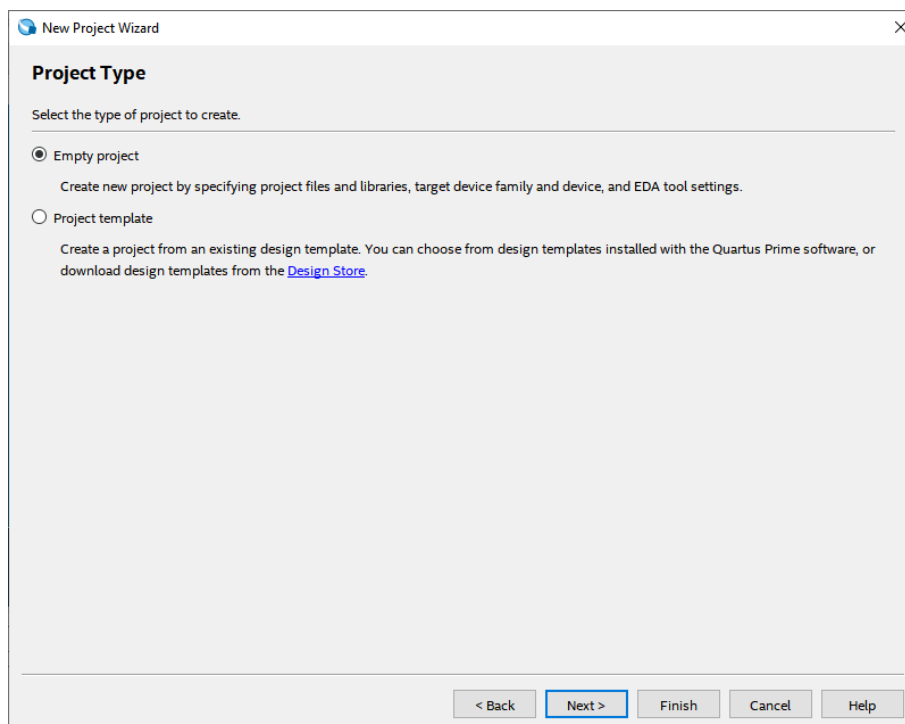


3. Kemudian tentukan Directory Project dan Nama Project, lalu klik **Next**



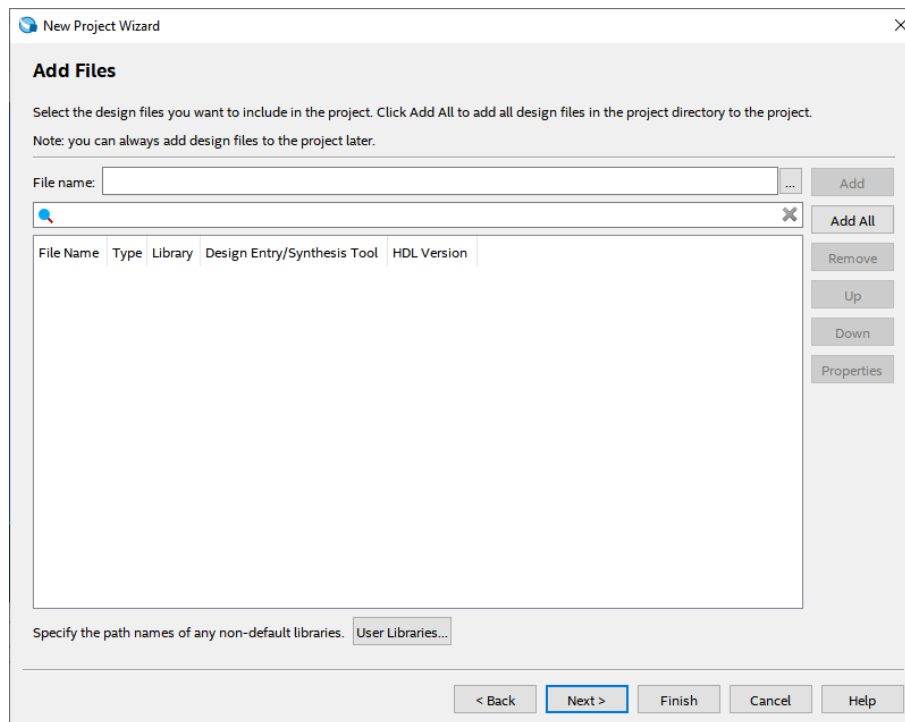
The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button. The main heading is 'Directory, Name, Top-Level Entity'. Below this, there are three text input fields with '...' buttons to their right. The first field is labeled 'What is the working directory for this project?' and contains the text 'C:\intelFPGA_lite\18.1'. The second field is labeled 'What is the name of this project?' and contains the text 'decoder'. The third field is labeled 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' and contains the text 'decoder'. Below these fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog box, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

4. Pilih **Empty Project**, kemudian klik **Next** lagi

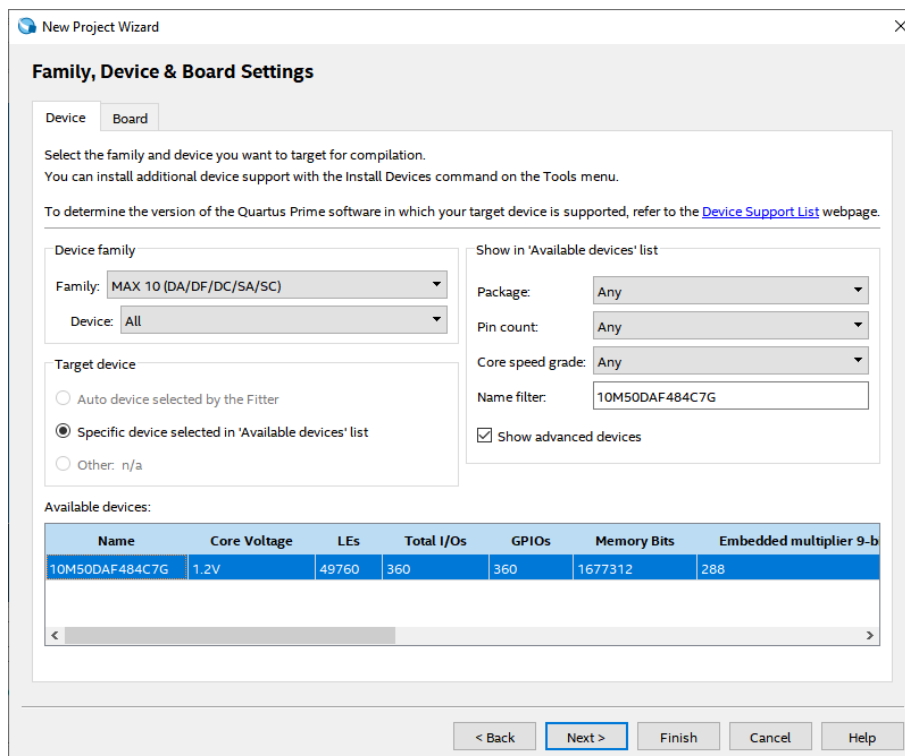


The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button. The main heading is 'Project Type'. Below this, there is a text label 'Select the type of project to create.' followed by two radio button options. The first option is 'Empty project', which is selected (indicated by a filled radio button). Below it is the text 'Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.' The second option is 'Project template', which is not selected (indicated by an empty radio button). Below it is the text 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog box, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

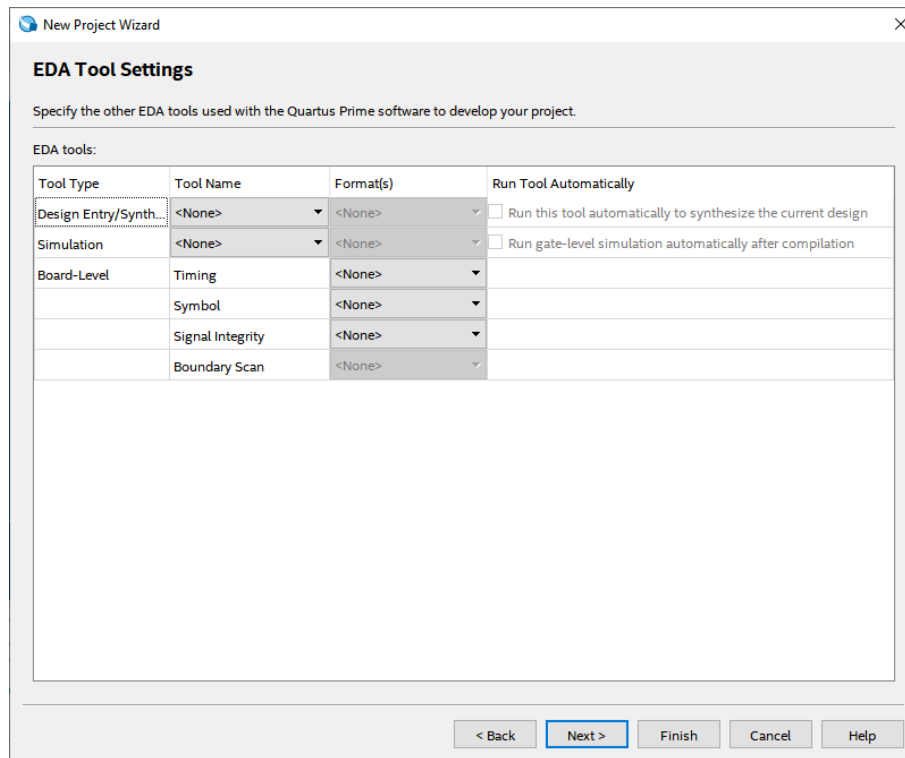
5. Setelah itu, klik **Next** lagi



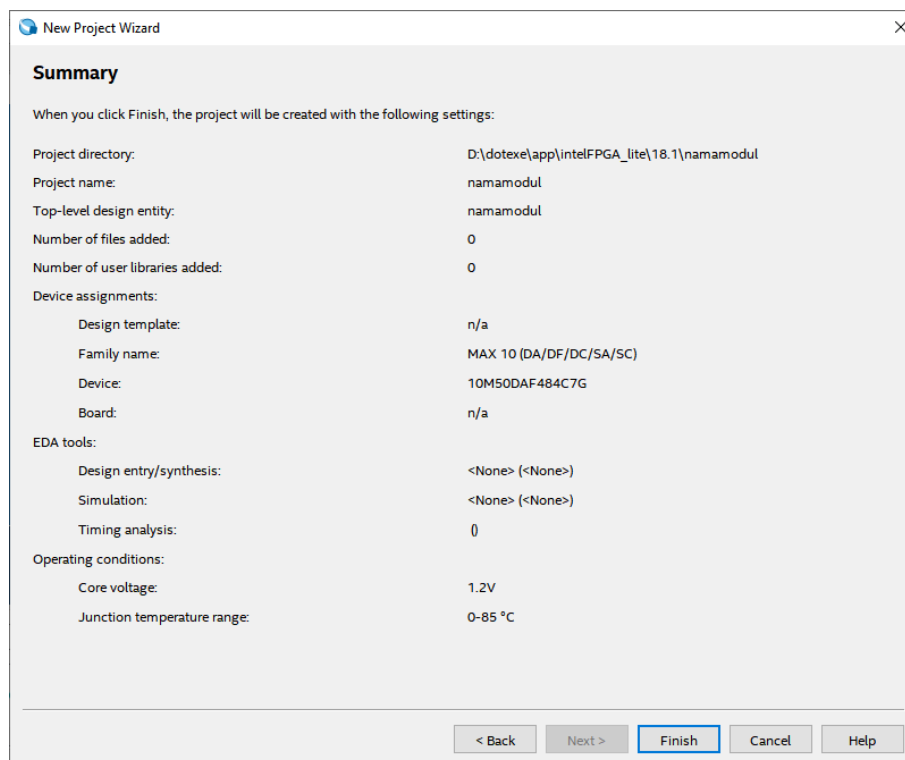
6. Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next**



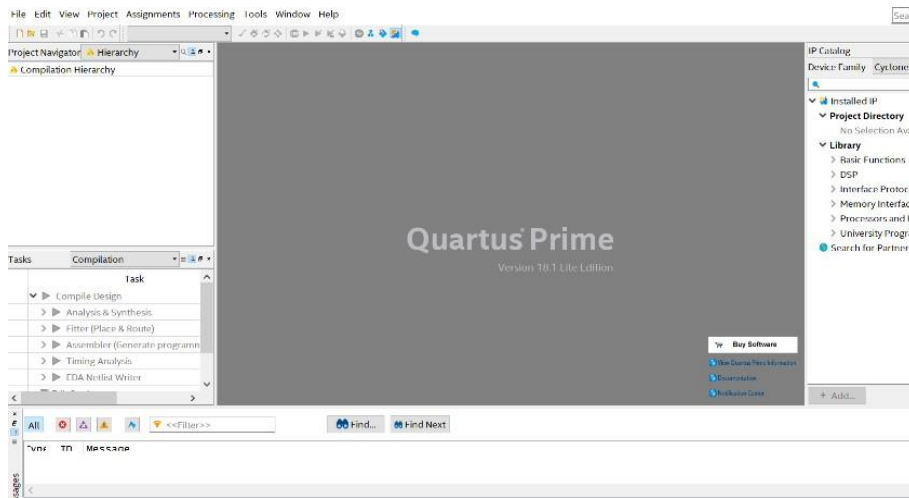
7. Kemudian klik **Next**



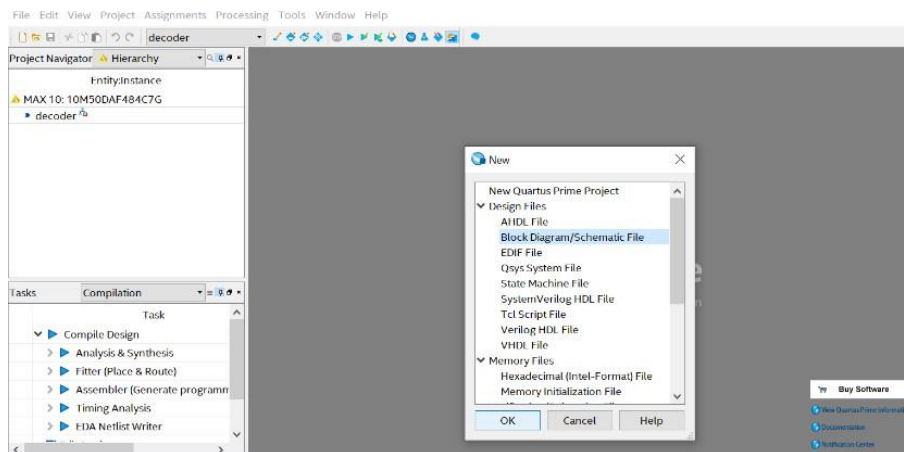
8. Lalu klik **Finish**



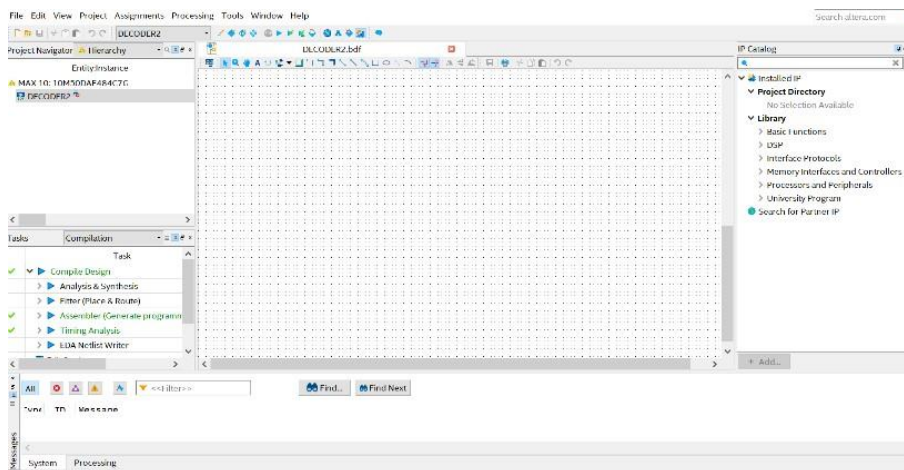
9. Setelah itu akan muncul tampilan awal dari Project Quartus Prime seperti gambar dibawah ini



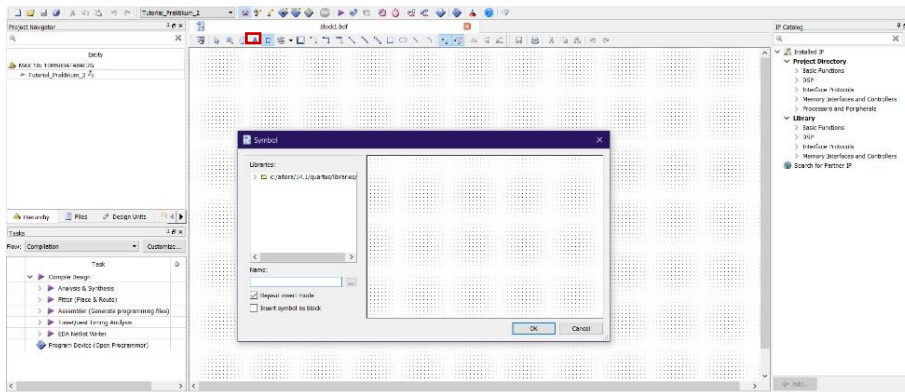
10. Kemudian buat file baru dengan cara klik **File** → **New**, lalu pilih **Block Diagram/Schematic File**, kemudian klik **OK**



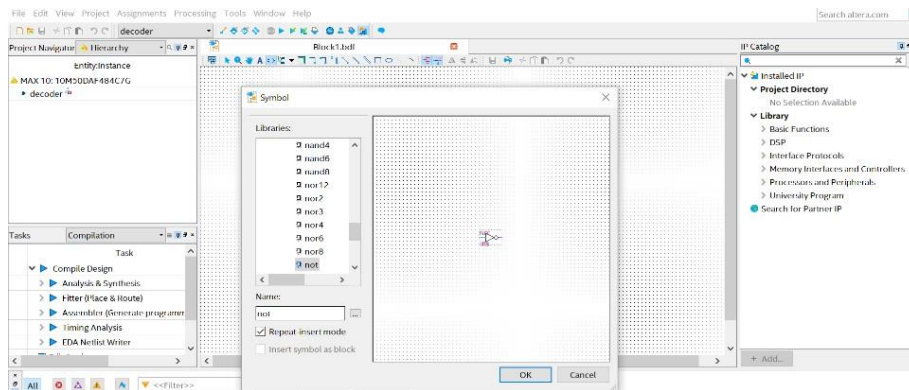
11. Akan muncul workspace seperti gambar dibawah ini



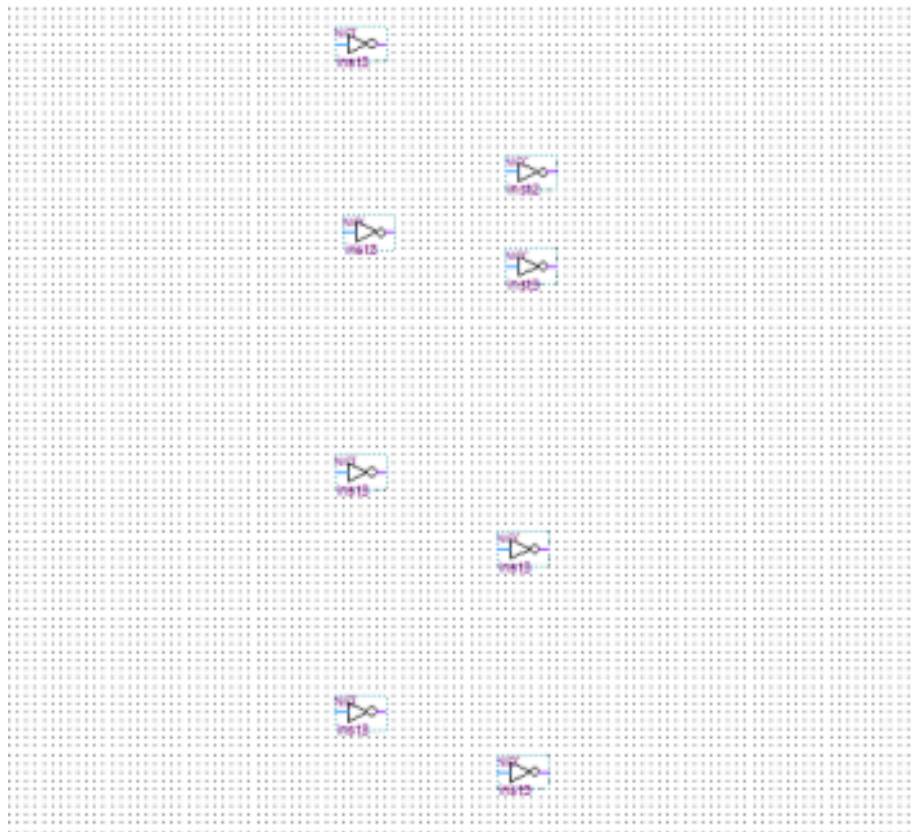
12. Lalu kita masuk pada pembuatan rangkaian *decoder*. Klik tool **Symbol Tool** seperti gambar dibawah (di mark merah)



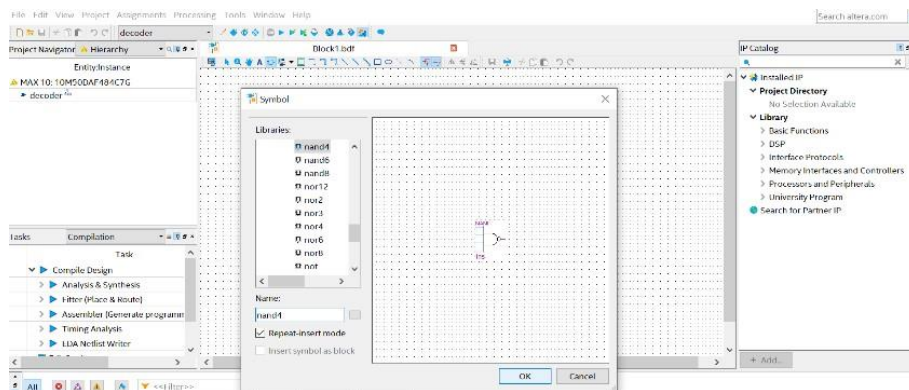
13. Kemudian pada Libraries, klik folder **c:/altera/14.1/quartus/libraries/** → **primitives** → **logic**, lalu pilih gerbang logika dengan nama **not** dan kli **OK**



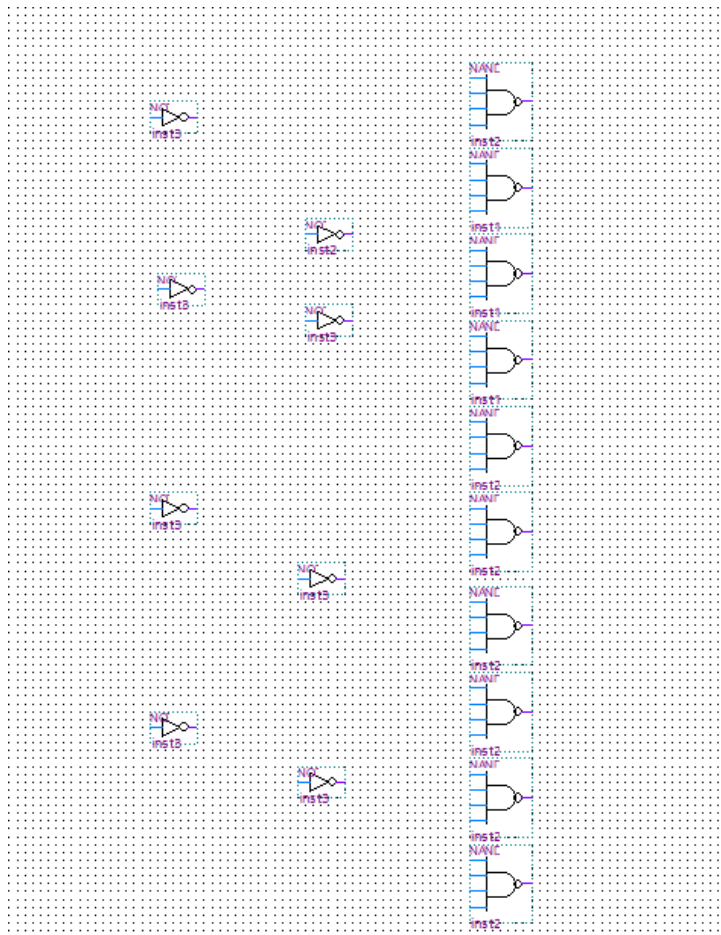
14. Lalu susun gerbang logika **not** sebanyak 8 gerbang logika



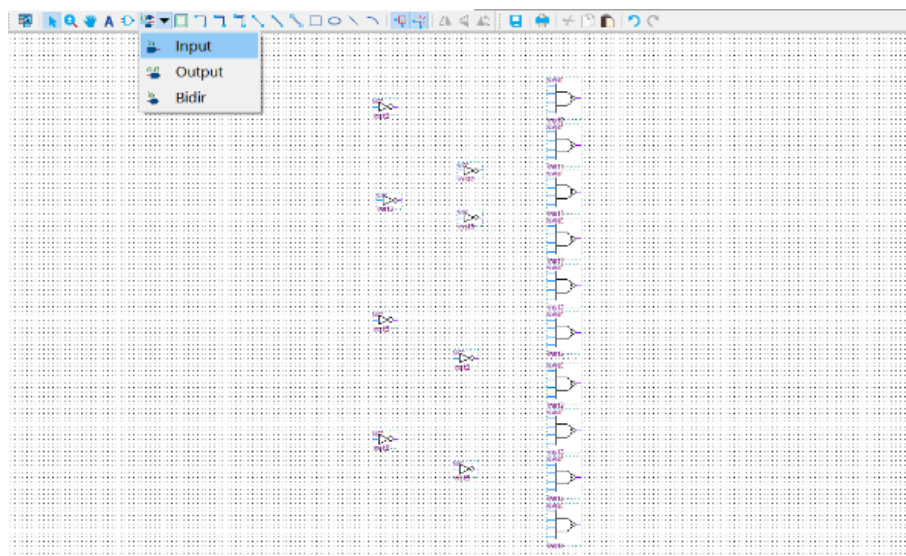
15. Kemudian pada libraries, klik folder **c:/altera/14.1/quartus/libraries/** → **primitives** → **logic**, lalu pilih gerbang logika dengan nama **nand4** dan klik **OK**

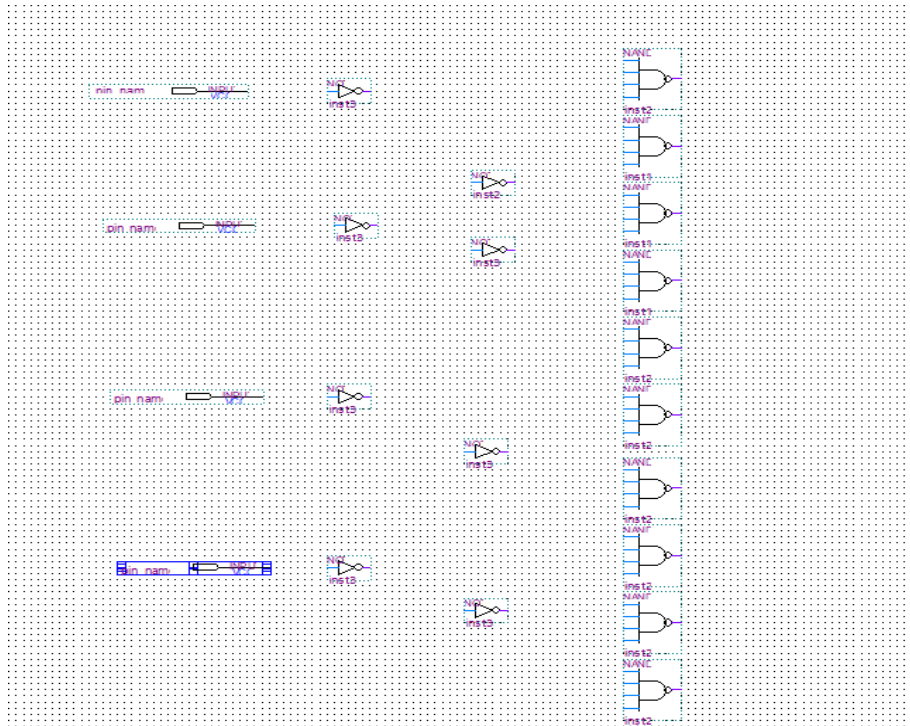


16. Lalu susun gerbang logika **nand4** sebanyak 10 gerbang logika

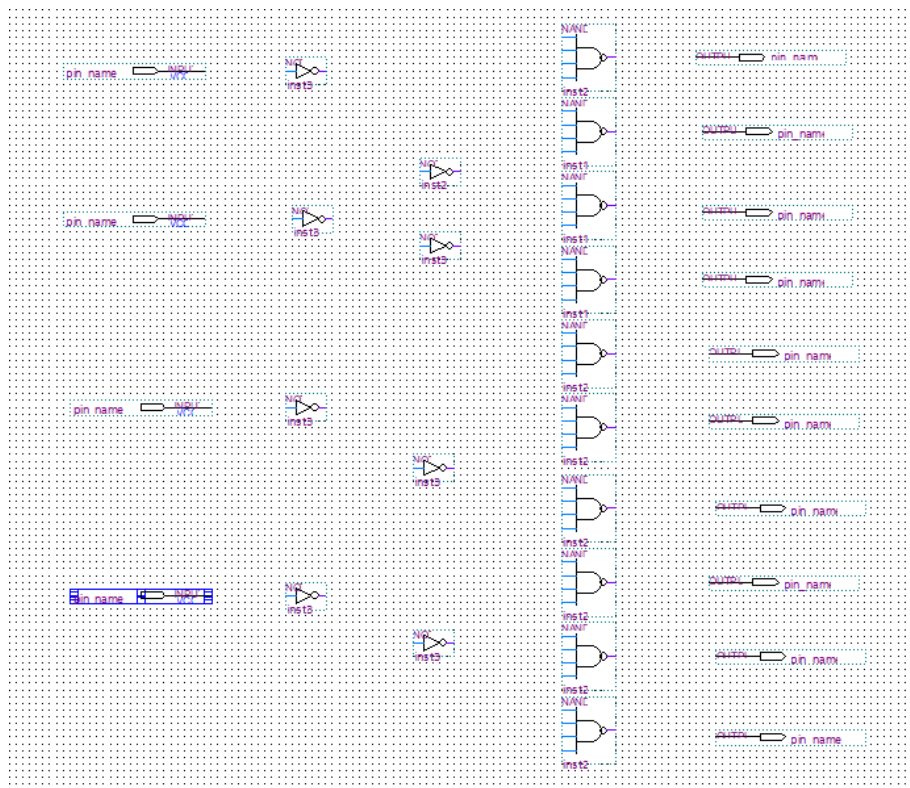


17. Klik tools **Pin Tools** → **input**, letakkan pin input pada kiri gerbang logika **not**

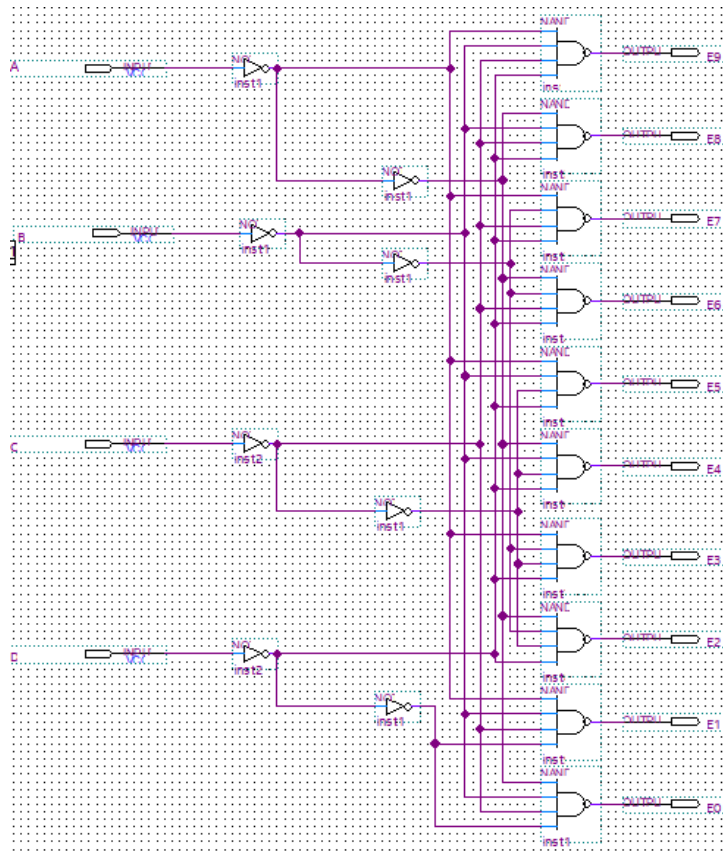




18. Klik tools **Pin Tools** → **output**, letakkan pin input pada kanan gerbang logika **nand4**.



19. Lakukan wiring seperti gambar dibawah ini



20. Ubah nama pin (pin name) sesuai dengan yang tersedia pada gambar. Caranya **double click** pada tulisan pin name atau bias juga dengan **klik kanan** tulisan pin name lalu klik **Propeties**

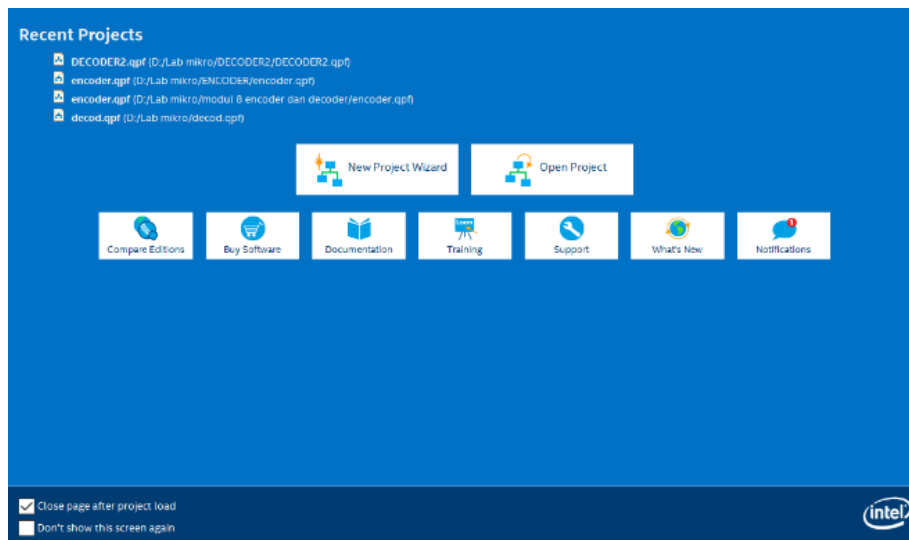
21. Lalu klik **Assignment** → **Pin Planner**. Setelah selesai lalu **Run I/O Assigmen**

Node Name	Direction	Location	I/O Bank	/REF Group	Iter Locatic	/O Standar	Reserved	rent Stren	Slew Rate	fferential P.	it Preserve
A	Input	PIN_C10	7	B7_NO	PIN_C10	2.5 V		12mA...ult)			
B	Input	PIN_C11	7	B7_NO	PIN_C11	2.5 V		12mA...ult)			
C	Input	PIN_D12	7	B7_NO	PIN_D12	2.5 V		12mA...ult)			
D	Input	PIN_C12	7	B7_NO	PIN_C12	2.5 V		12mA...ult)			
E0	Output	PIN_A8	7	B7_NO	PIN_A8	2.5 V		12mA...ult) 2 (default)			
E1	Output	PIN_A9	7	B7_NO	PIN_A9	2.5 V		12mA...ult) 2 (default)			
E2	Output	PIN_A10	7	B7_NO	PIN_A10	2.5 V		12mA...ult) 2 (default)			
E3	Output	PIN_B10	7	B7_NO	PIN_B10	2.5 V		12mA...ult) 2 (default)			
E4	Output	PIN_D13	7	B7_NO	PIN_D13	2.5 V		12mA...ult) 2 (default)			
E5	Output	PIN_C13	7	B7_NO	PIN_C13	2.5 V		12mA...ult) 2 (default)			
E6	Output	PIN_E14	7	B7_NO	PIN_E14	2.5 V		12mA...ult) 2 (default)			
E7	Output	PIN_D14	7	B7_NO	PIN_D14	2.5 V		12mA...ult) 2 (default)			
E8	Output	PIN_A11	7	B7_NO	PIN_A11	2.5 V		12mA...ult) 2 (default)			
E9	Output	PIN_B11	7	B7_NO	PIN_B11	2.5 V		12mA...ult) 2 (default)			

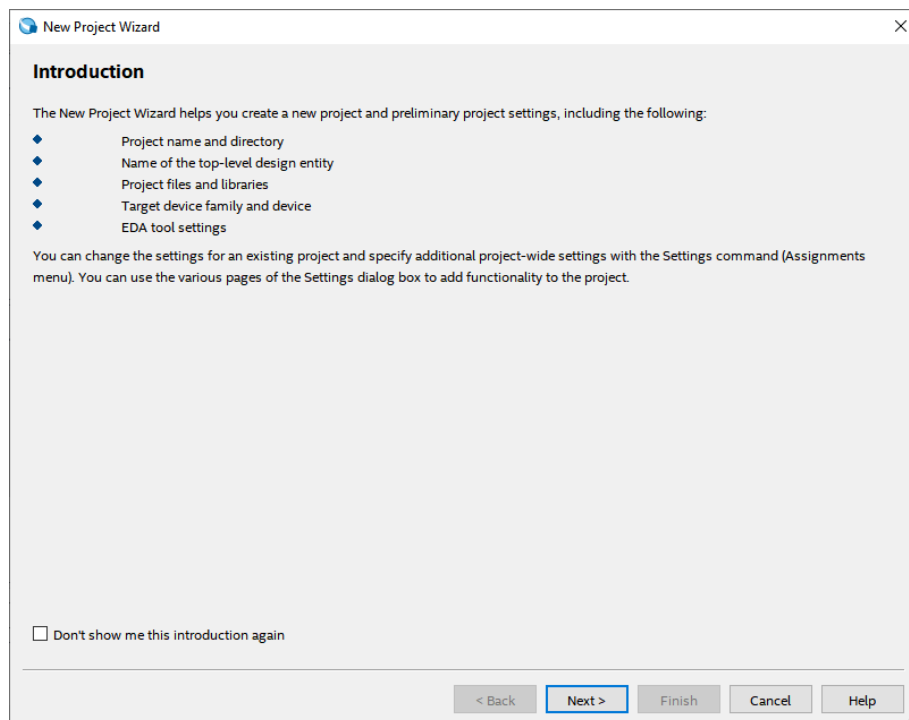
22. Lakukan **Compile Design**, setelah selesai proses **Compile Design** maka langkah praktikum modul 8 Rangkaian Decoder telah selesai

8.3.3 Langkah Praktikum Modul 8 (Rangkaian *Encoder*)

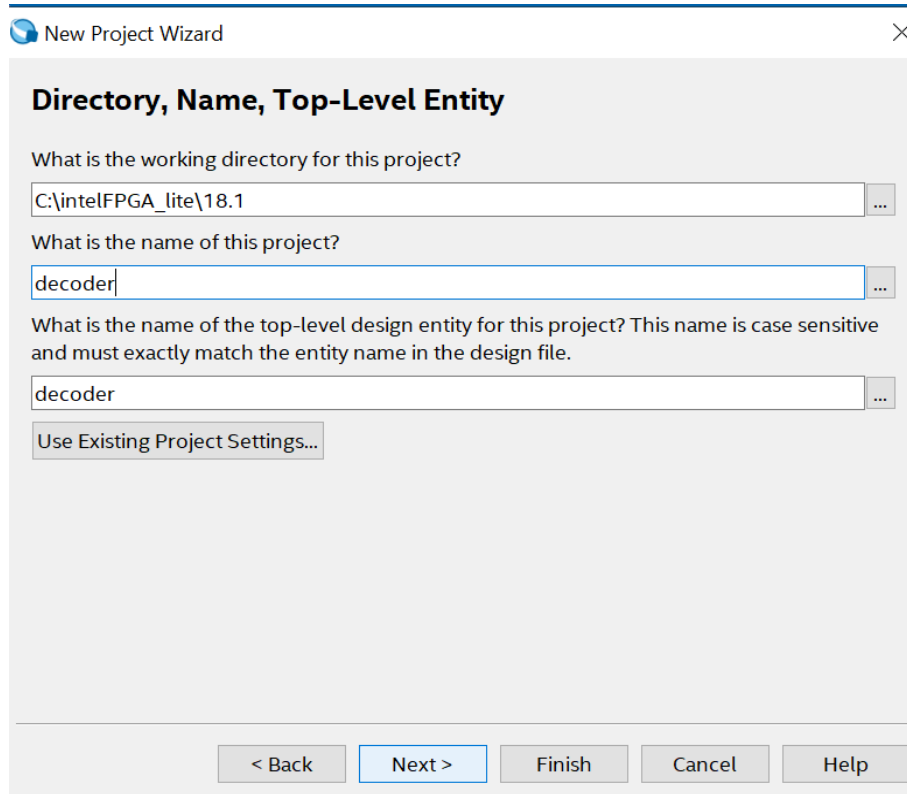
1. Buka software Quartus lite di laptop dan klik **New Project Wizard**



2. **Klik Next**

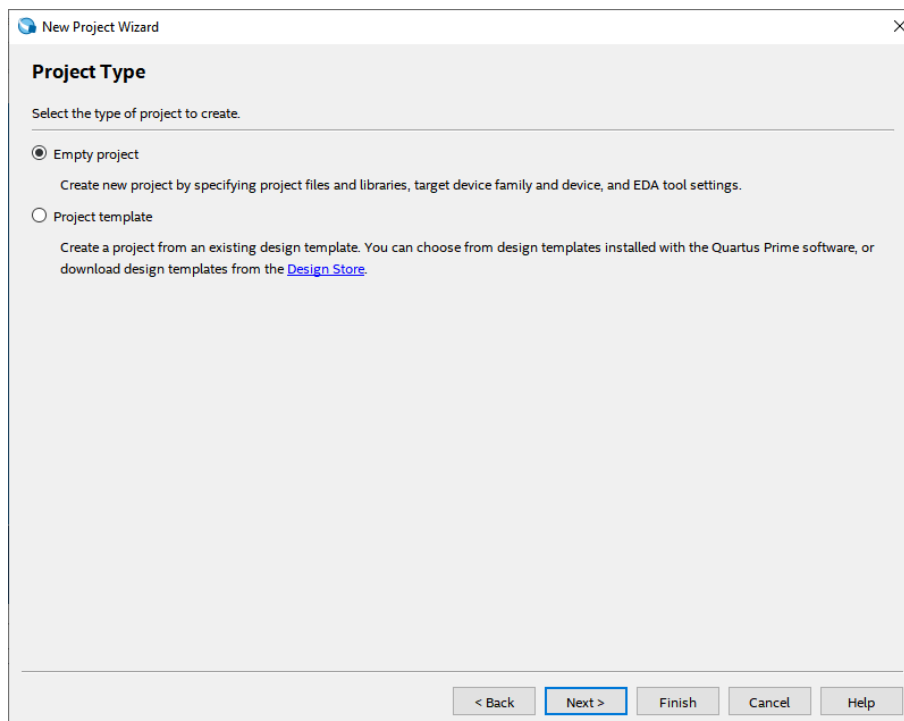


3. Kemudian tentukan Directory Project dan Nama Project, lalu klik **Next**



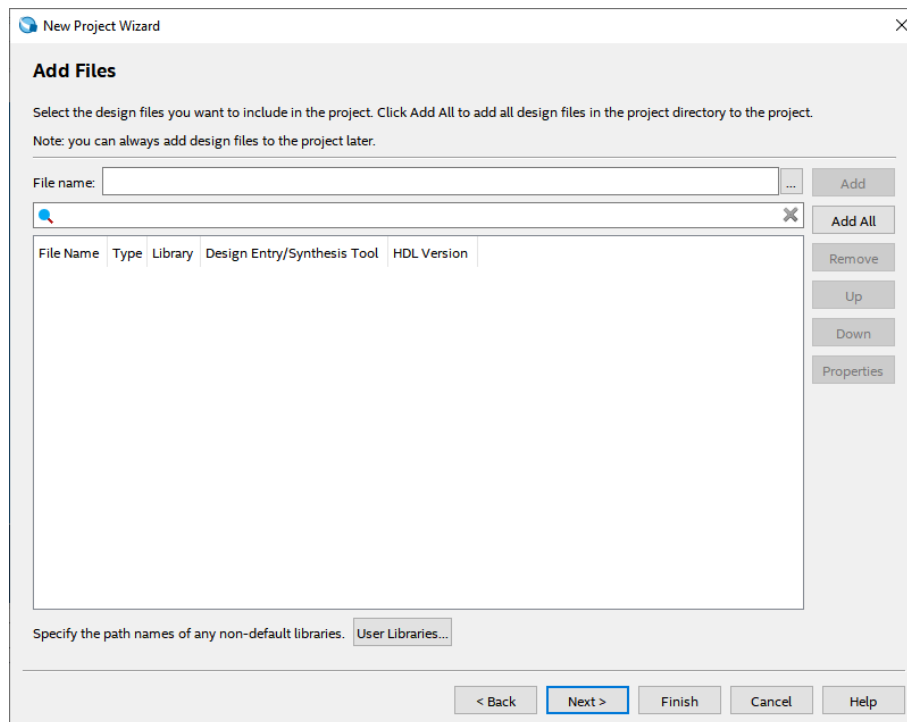
The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button. The main heading is 'Directory, Name, Top-Level Entity'. Below this, there are three text input fields with '...' buttons to their right. The first field is labeled 'What is the working directory for this project?' and contains the text 'C:\intelFPGA_lite\18.1'. The second field is labeled 'What is the name of this project?' and contains the text 'decoder'. The third field is labeled 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' and contains the text 'decoder'. Below these fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

4. Pilih **Empty Project**, kemudian klik **Next** lagi

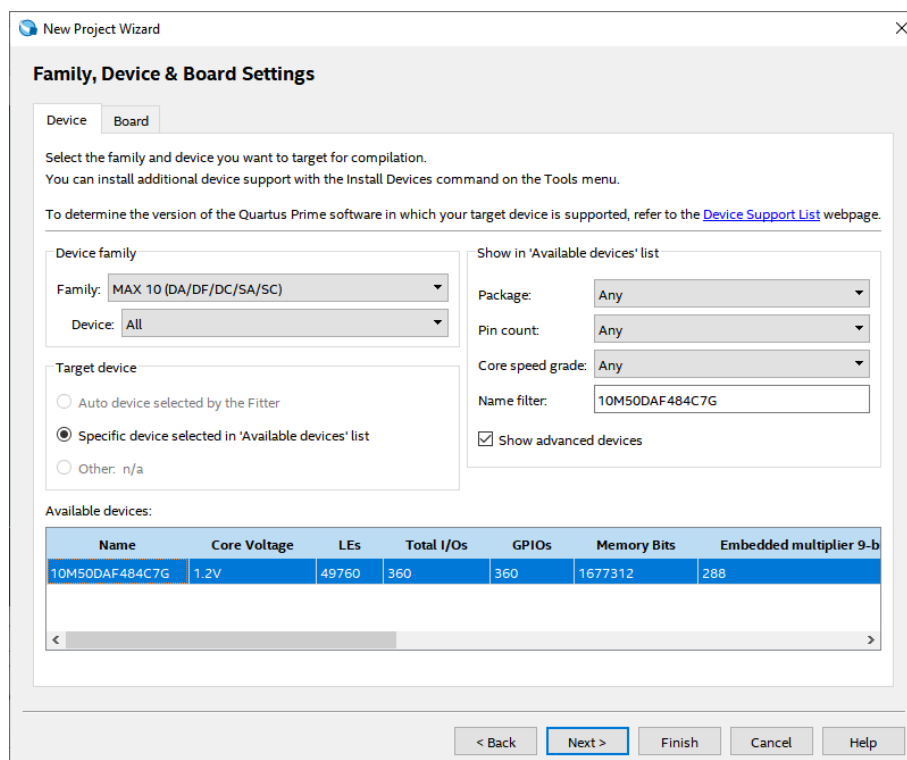


The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button. The main heading is 'Project Type'. Below this, there is a text label 'Select the type of project to create.' followed by two radio button options. The first option is 'Empty project', which is selected (indicated by a filled radio button). Below it is the text 'Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.' The second option is 'Project template', which is not selected (indicated by an empty radio button). Below it is the text 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

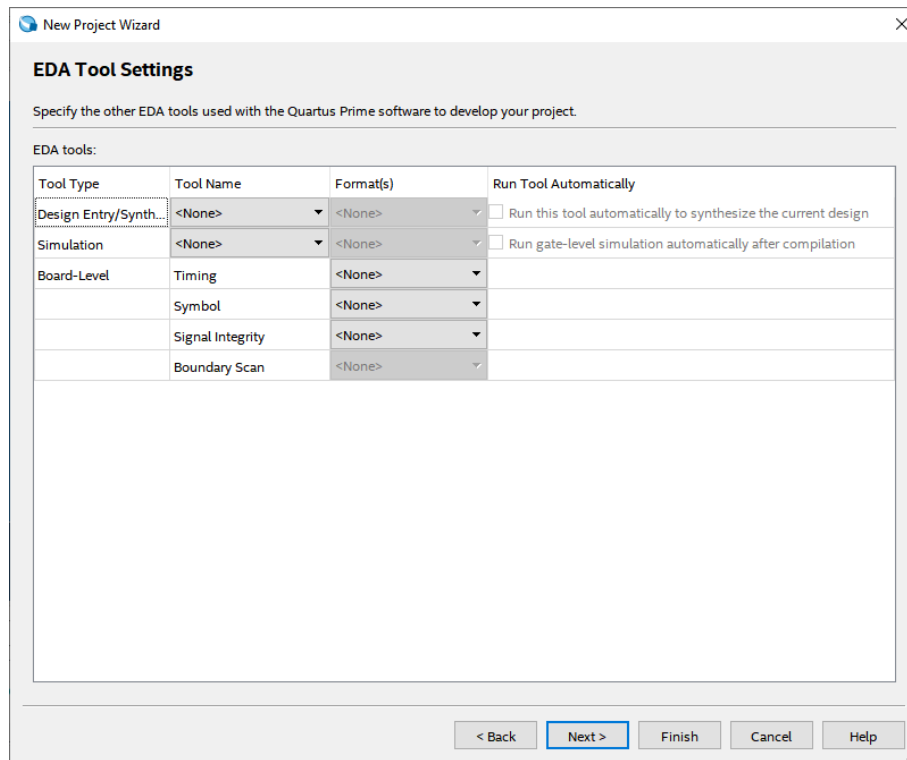
5. Setelah itu, klik **Next** lagi



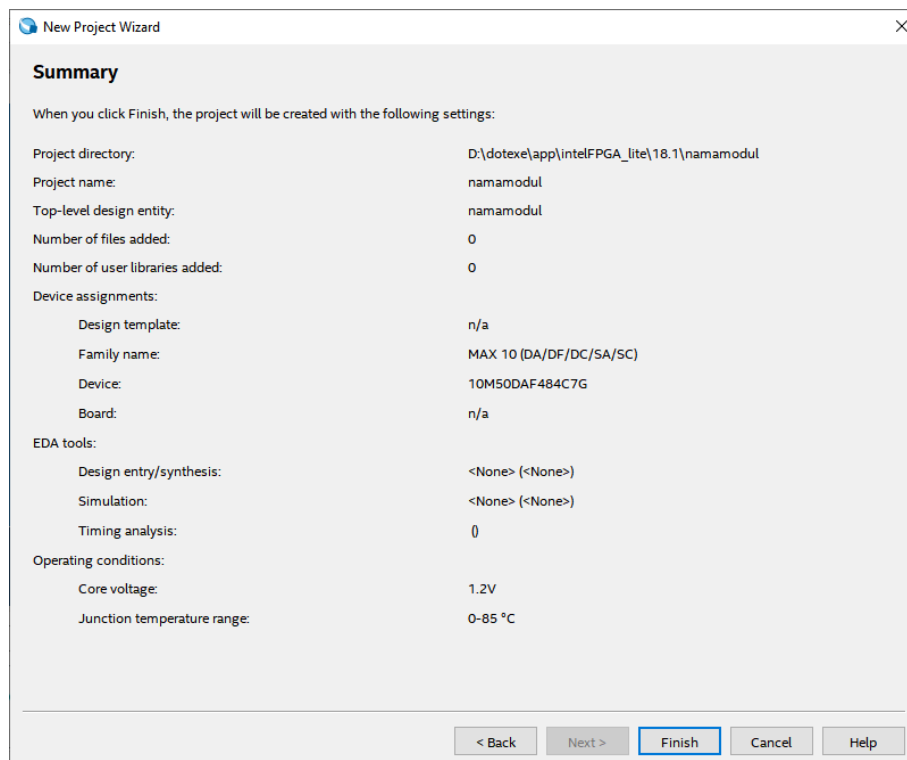
6. Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next**



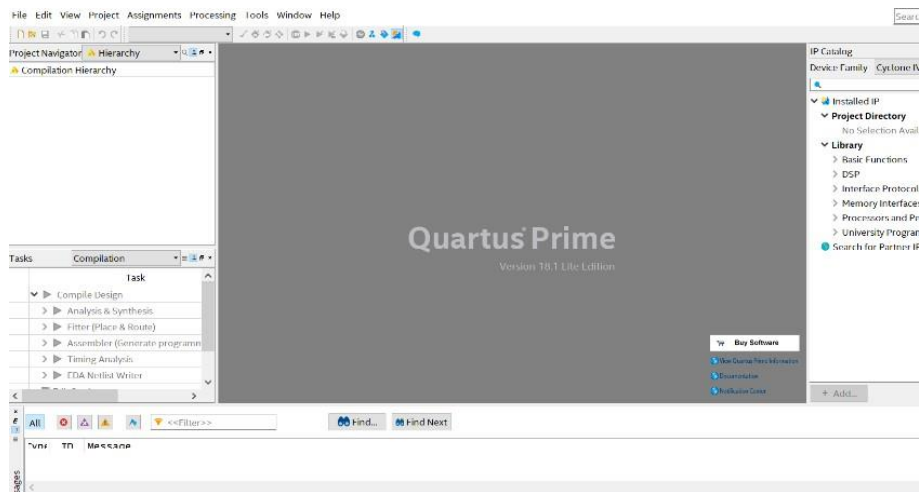
7. Kemudian klik **Next**



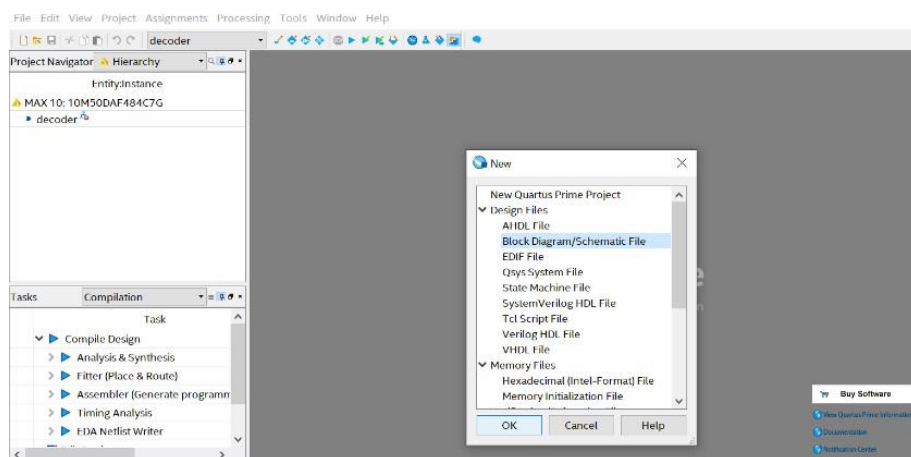
8. Lalu klik **Finish**



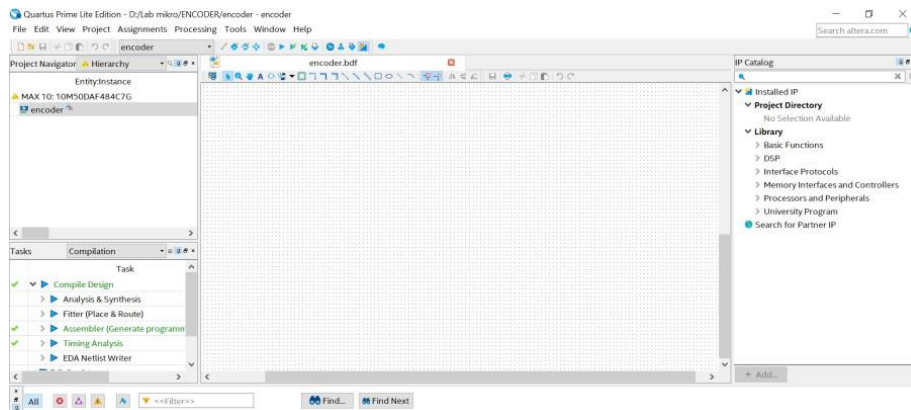
9. Setelah itu akan muncul tampilan awal dari Project Quartus Prime seperti gambar dibawah ini



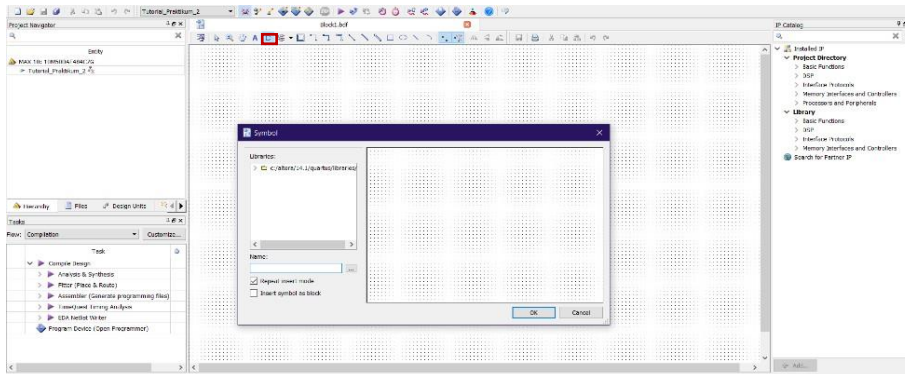
10. Kemudian buat file baru dengan cara klik **File** → **New**, lalu pilih **Block Diagram/Schematic File**, kemudian klik **OK**



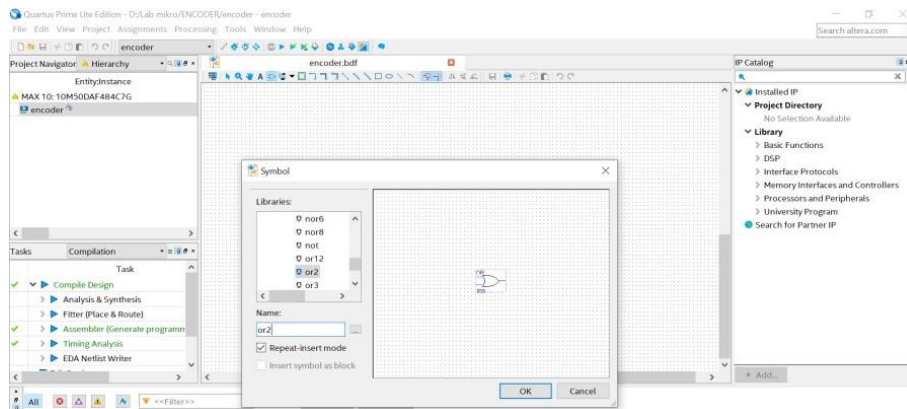
11. Akan muncul workspace seperti gambar dibawah ini



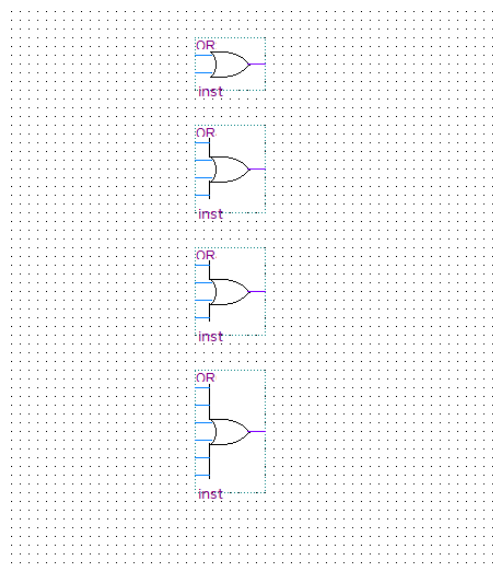
12. Lalu kita masuk pada pembuatan rangkaian *decoder*. Klik tool **Symbol Tool** seperti gambar dibawah (di mark merah)



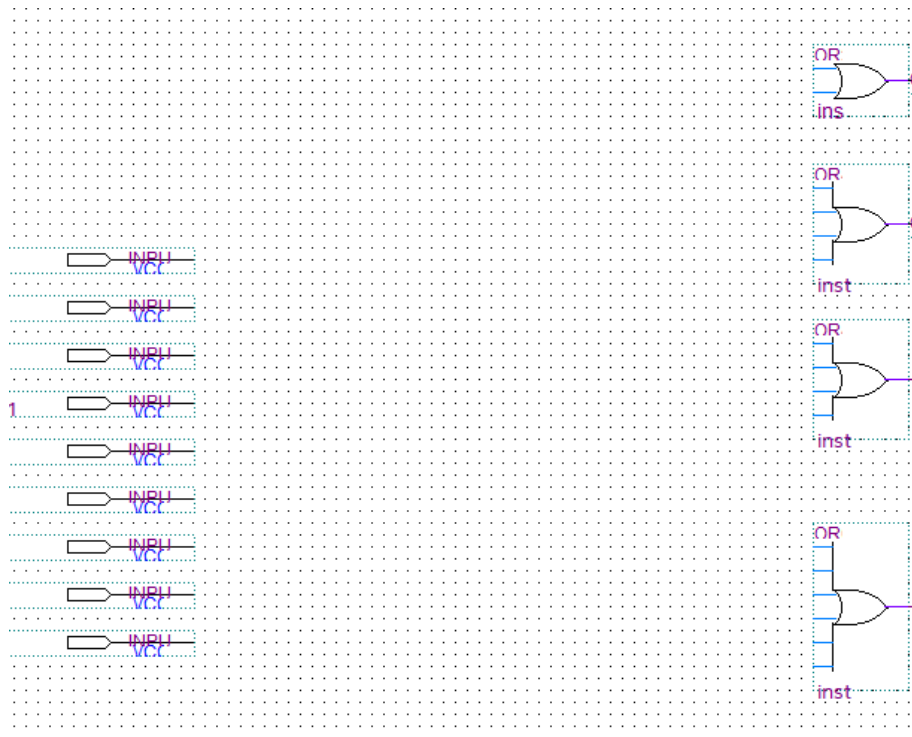
13. Kemudian pada Libraries, klik folder **c:/altera/14.1/quartus/libraries/→ primitives → logic**, lalu pilih gerbang logika dengan nama **OR2** dan klik **OK**. Lakukan langkah yang sama untuk menambahkan gerbang logika **OR4** dan **OR6**.



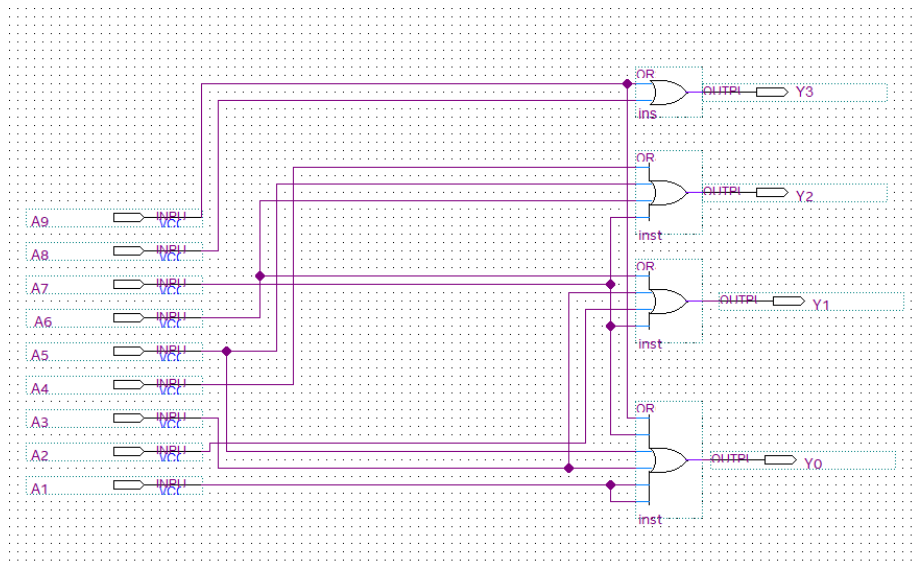
14. Lalu susun gerbang logika **OR2**, **OR4**, dan **OR6** seperti gambar dibawah.



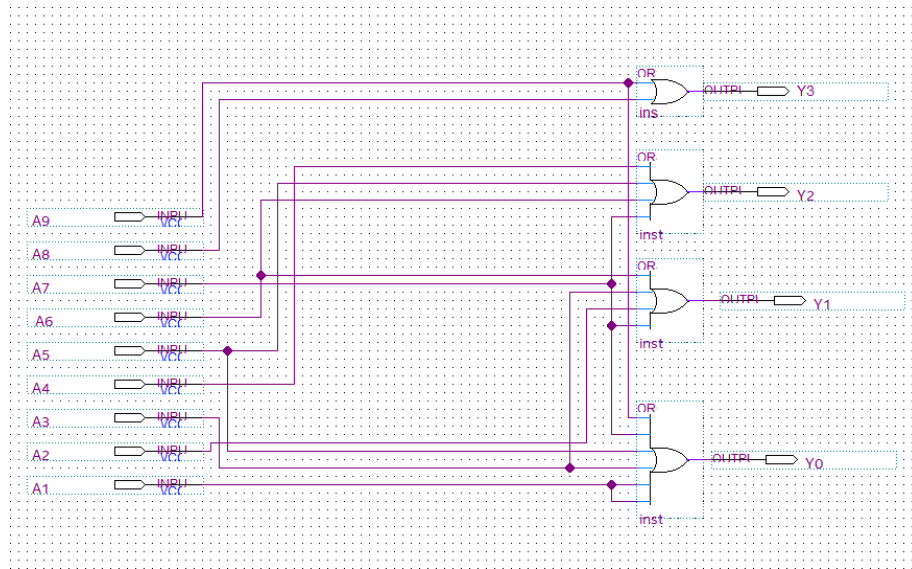
15. Klik tools **Pin Tools** → **input**, letakkan pin input pada sebelah kiri.



16. Lakukan wiring seperti gambar dibawah ini



17. Ubah nama pin (pin name) sesuai gambar. Caranya **double click** pada tulisan pin name atau bias juga dengan **klik kanan** tulisan pin name lalu klik **Propeties**



18. Lalu klik **Assignment** → **Pin Planner**. Setelah selesai lalu **Run I/O Assignment**.

The screenshot shows the Pin Planner software interface. The main window displays a 'Top View - Wire Bond' of a MAX 10-10M50DAF484C7G chip. On the left, there are panels for 'Report', 'Groups', and 'Tasks'. The 'Tasks' panel shows 'Early Pin Planning' and 'Run I/O Assignment' as active tasks. Below the chip view, there is a table with the following columns: Node Name, Direction, Location, I/O Bank, /REF Group, I/O Location, I/O Standard, Reserved, Current Strength, Slew Rate, Differential Pair, and Preserve.

Node Name	Direction	Location	I/O Bank	/REF Group	I/O Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Preserve
A1	Input	PIN_C10	7	B7_NO	PIN_C10	2.5V		12mA...ult			
A2	Input	PIN_C11	7	B7_NO	PIN_C11	2.5V		12mA...ult			
A3	Input	PIN_D12	7	B7_NO	PIN_D12	2.5V		12mA...ult			
A4	Input	PIN_C12	7	B7_NO	PIN_C12	2.5V		12mA...ult			
A5	Input	PIN_A12	7	B7_NO	PIN_A12	2.5V		12mA...ult			
A6	Input	PIN_B12	7	B7_NO	PIN_B12	2.5V		12mA...ult			
A7	Input	PIN_A13	7	B7_NO	PIN_A13	2.5V		12mA...ult			
A8	Input	PIN_A14	7	B7_NO	PIN_A14	2.5V		12mA...ult			
A9	Input	PIN_B14	7	B7_NO	PIN_B14	2.5V		12mA...ult			
Y0	Output	PIN_A8	7	B7_NO	PIN_A8	2.5V		12mA...ult	2 (default)		
Y1	Output	PIN_A9	7	B7_NO	PIN_A9	2.5V		12mA...ult	2 (default)		
Y2	Output	PIN_A10	7	B7_NO	PIN_A10	2.5V		12mA...ult	2 (default)		
Y3	Output	PIN_B10	7	B7_NO	PIN_B10	2.5V		12mA...ult	2 (default)		

19. Lakukan **Compile Design**, setelah selesai proses **Compile Design** maka langkah praktikum modul 8 Rangkaian Encoder telah selesai

8.4 Soal Jurnal

1. Lengkapilah Tabel pengujian dari rangkaian *encoder* maupun rangkaian *decoder*

a. Tabel Pengujian *encoder*

A1	A2	A3	A4	A5	A6	A7	A8	A9	Y0	Y1	Y2	Y3
0	0	0	0	0	0	0	0	0				
1	0	0	0	0	0	0	0	0				
0	1	0	0	0	0	0	0	0				
0	0	1	0	0	0	0	0	0				
0	0	0	1	0	0	0	0	0				
0	0	0	0	1	0	0	0	0				
0	0	0	0	0	1	0	0	0				
0	0	0	0	0	0	1	0	0				
0	0	0	0	0	0	0	1	0				
0	0	0	0	0	0	0	0	1				

b. Tabel Pengujian *decoder*

Input				Output Keluaran									
A	B	C	D	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9
0	0	0	0										
1	0	0	0										
0	1	0	0										
1	1	0	0										
0	0	1	0										
1	0	1	0										
0	1	1	0										
1	1	1	0										
0	0	0	1										
1	0	0	1										
0	1	0	1										

2. Jelaskan pengertian *encoder* dan *decoder* menggunakan Bahasa kalian sendiri setelah melakukan praktikum modul 8!

3. Tuliskan apa yang telah dilakukan pada praktikum modul 8 menggunakan Bahasa kalian sendiri!

MODUL 9

SIMULASI RANGKAIAN BCD-TO-7 SEGMENT

9.1 Tujuan Praktikum Modul 9

Setelah mempraktekkan topik ini, praktikan diharapkan dapat:

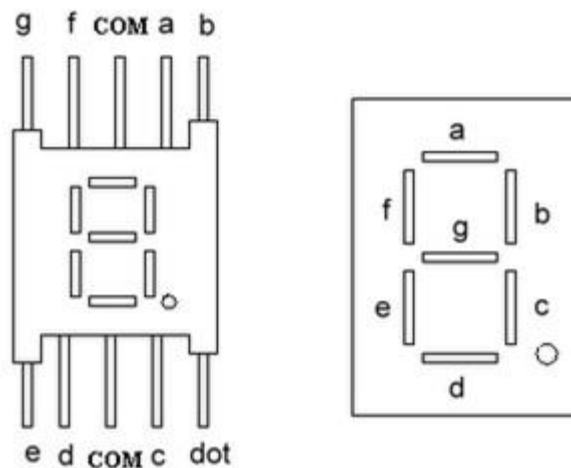
1. Praktikan dapat mengenal dan mengetahui *BCD to 7 segment*
2. Praktikan dapat membuat rangkaian *BCD to 7 segment*
3. Praktikan dapat mensimulasikan rangkaian *BCD to 7 segment*

9.2 Dasar Teori Praktikum Modul 9

9.2.1 7-Segment

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

Gambar 9.1 Seven segment display



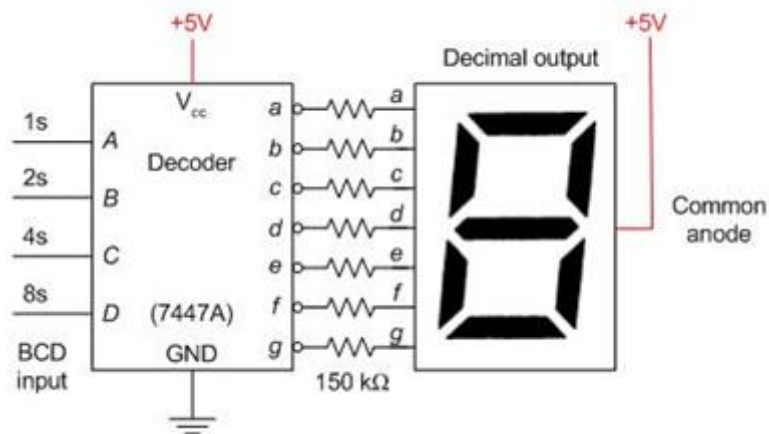
Seven-Segment Display

Seven segment dapat menampilkan angka – angka decimal dan beberapa karakter tertentu melalui kombinasi aktif atau tidaknya LED penyusunan dalam seven segment. Untuk mempermudah pengguna seven segment, umumnya digunakan sebuah decoder atau sebuah seven segment driver yang akan mengatur aktif atau tidaknya led-led dalam seven segment sesuai dengan inputan biner yang diberikan.

9.2.2 BCD to 7-Segment

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

Gambar 9.2 Decoder BCD to 7 Segment



Tabel Kebenaran

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

Tabel 9.1 BCD to 7 Segment Display Decoder

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	
1	0	0	0	1	1	0	0	1	1	1	1	
2	0	0	1	0	0	0	1	0	0	1	0	
3	0	0	1	1	0	0	0	0	1	1	0	
4	0	1	0	0	1	0	0	1	1	0	0	
5	0	1	0	1	0	1	0	0	1	0	0	
6	0	1	1	0	1	1	0	0	0	0	0	
7	0	1	1	1	0	0	0	1	1	1	1	
8	1	0	0	0	0	0	0	0	0	0	0	
9	1	0	0	1	0	0	0	1	1	0	0	
10	1	0	1	0	1	1	1	0	0	1	0	
11	1	0	1	1	1	1	0	0	1	1	0	
12	1	1	0	0	1	0	1	1	1	0	0	
13	1	1	0	1	0	1	1	0	1	0	0	
14	1	1	1	0	1	1	1	0	0	0	0	
15	1	1	1	1	1	1	1	1	1	1	1	

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

menampilkan visual nilai. IC TTL *Decoder to seven segment* dimuat pada IC TTL 7447 dan 7448.

9.3 Lembar Kegiatan Praktikum Modul 9

9.3.1 Alat dan Bahan

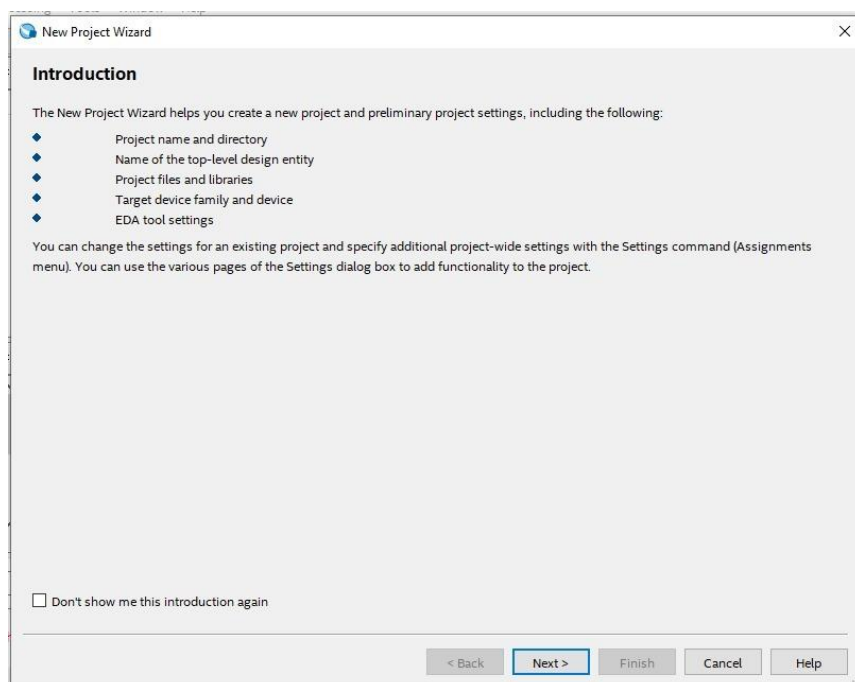
1. Software Quartus 18.1
2. Laptop
3. Mouse

9.3.2 Langkah Praktikum Modul 9

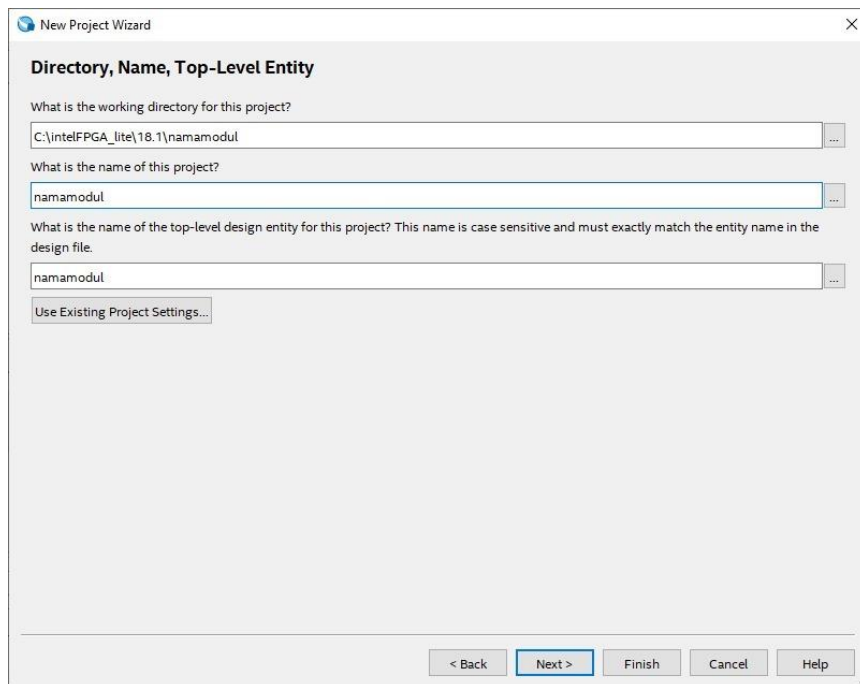
1. Klik **New Project Wizard**



2. Klik **Next**

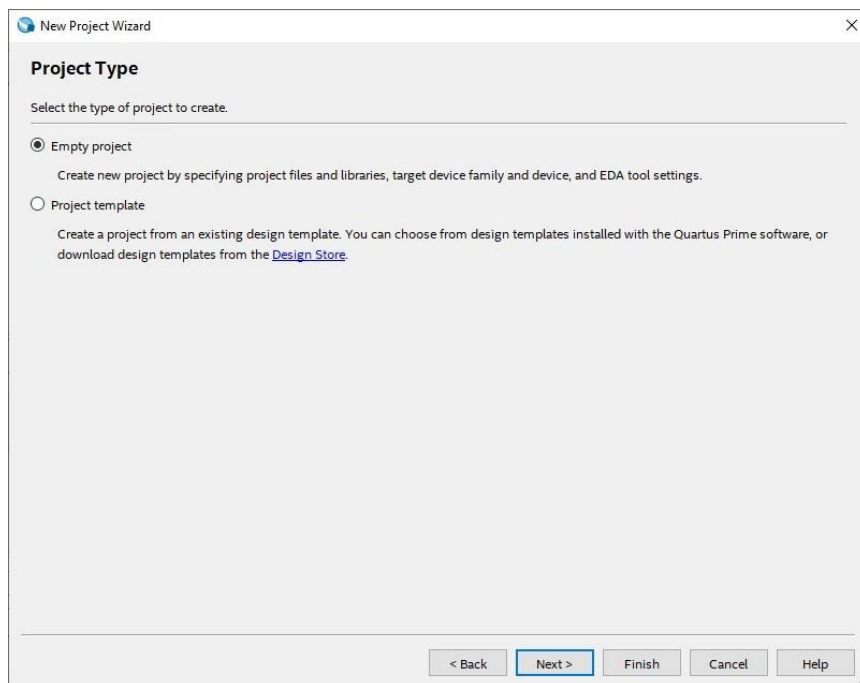


3. Kemudian tentukan Directory Project dan Nama Project, lalu klik **Next**



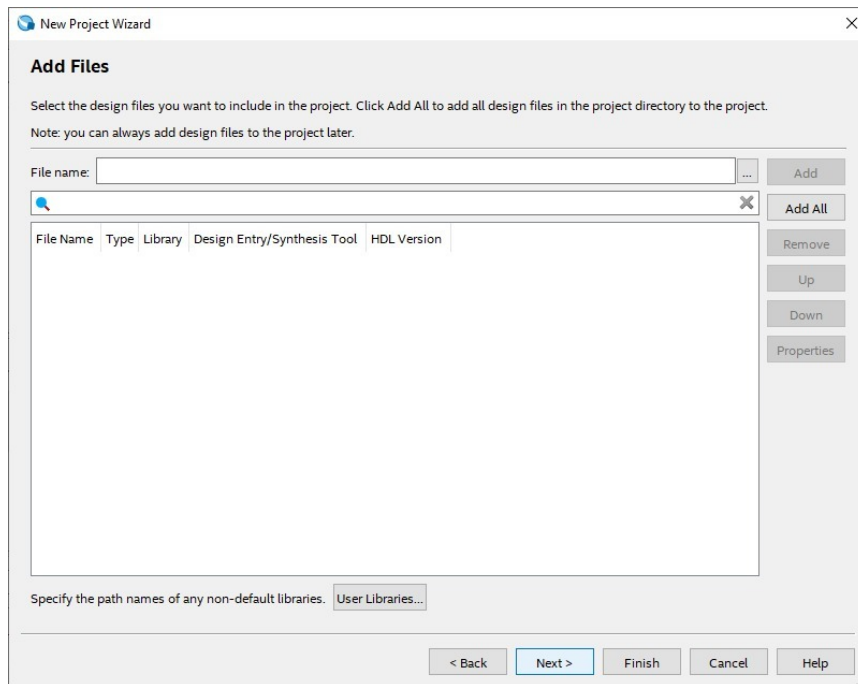
The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard'. The main heading is 'Directory, Name, Top-Level Entity'. There are three text input fields, each with a browse button ('...') to its right. The first field is labeled 'What is the working directory for this project?' and contains the path 'C:\intelFPGA_lite\18.1\namamodul'. The second field is labeled 'What is the name of this project?' and contains 'namamodul'. The third field is labeled 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' and contains 'namamodul'. Below the fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

4. Pilih **Empty Project**, kemudian klik **Next** lagi

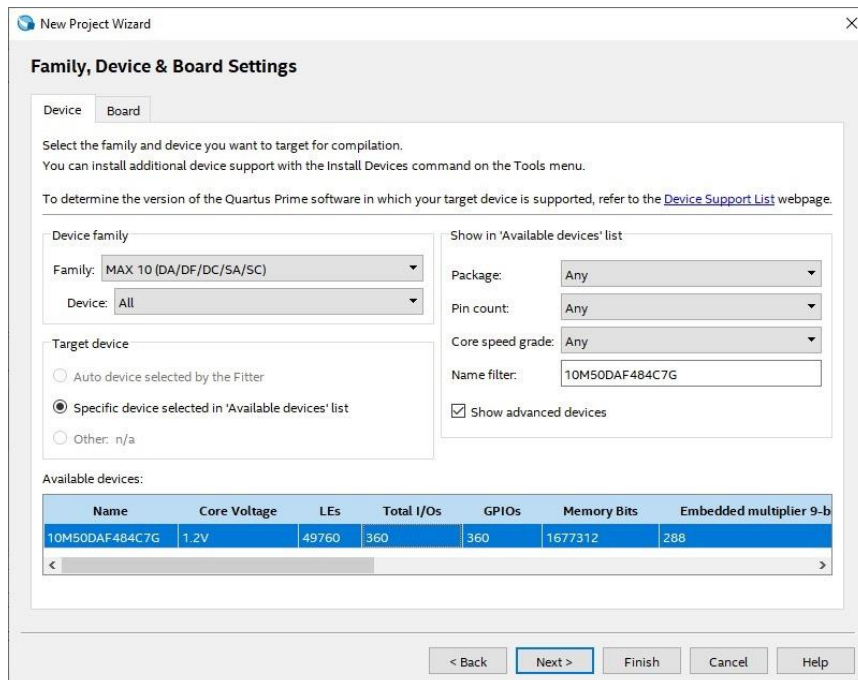


The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard'. The main heading is 'Project Type'. Below the heading is the instruction 'Select the type of project to create.' There are two radio button options. The first option is 'Empty project', which is selected (indicated by a filled radio button). Below it is the text 'Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.' The second option is 'Project template', which is not selected (indicated by an empty radio button). Below it is the text 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

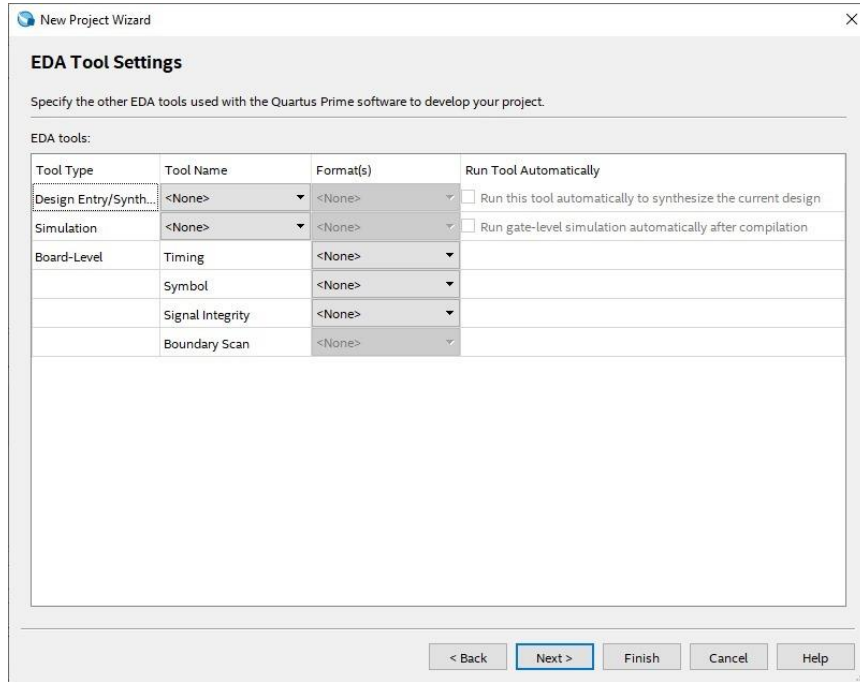
5. Setelah itu klik **Next** lagi



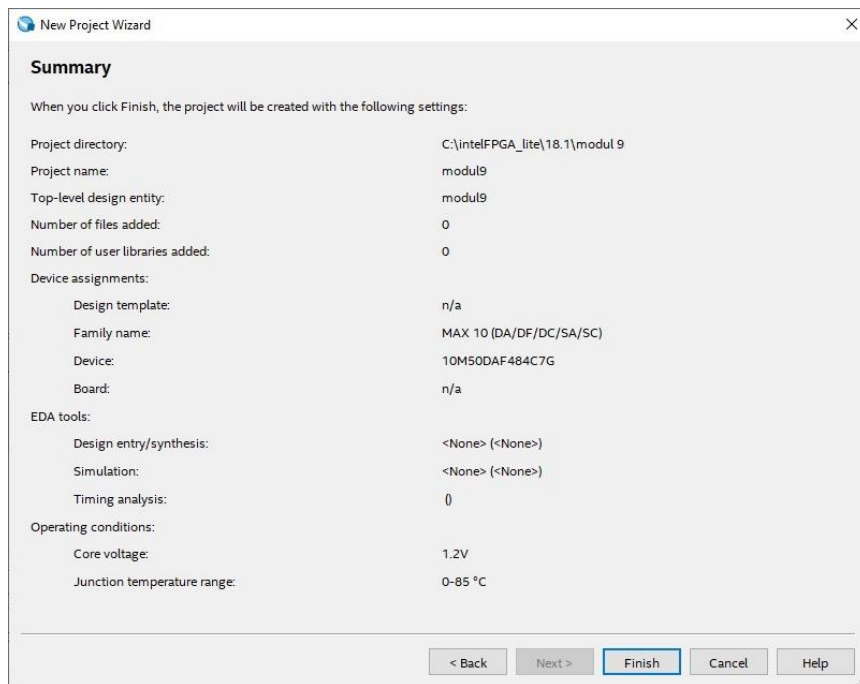
6. Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter** **10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next**



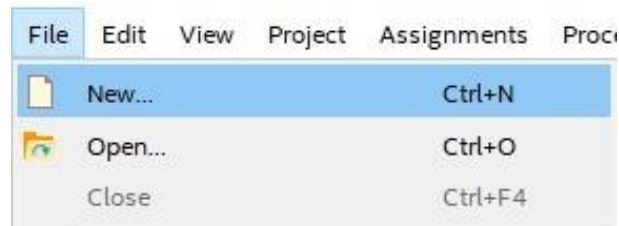
7. Kemudian klik **Next**



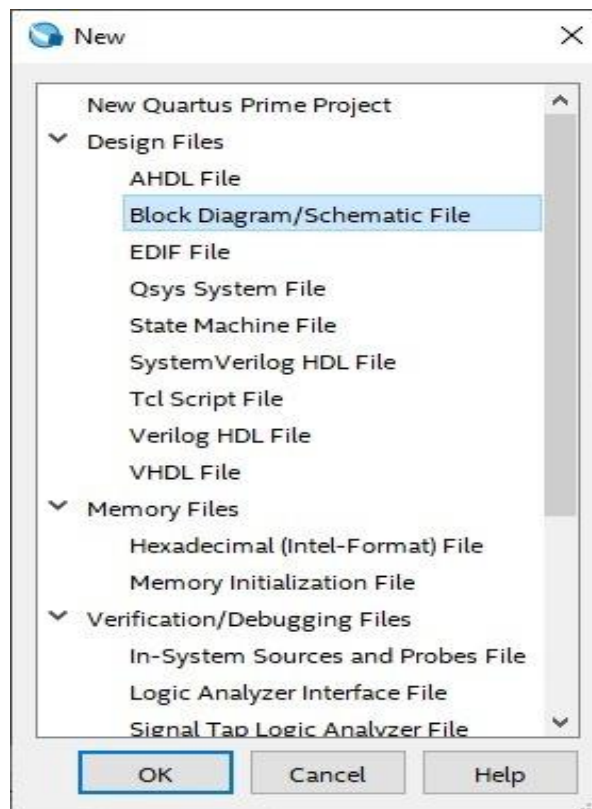
8. Lalu klik **Finish**



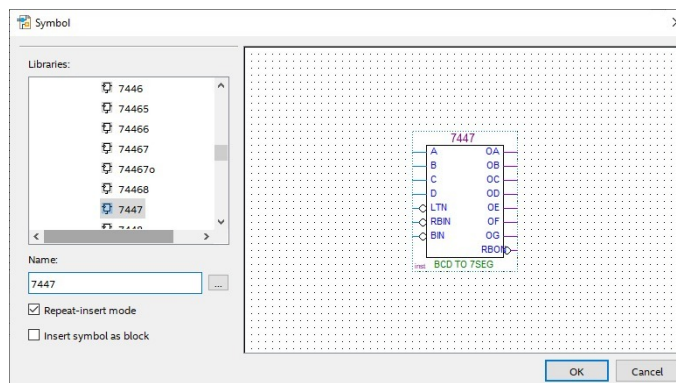
9. Buat file baru dengan klik File, lalu **New**. Dapat juga menggunakan shortcut Ctrl+N.



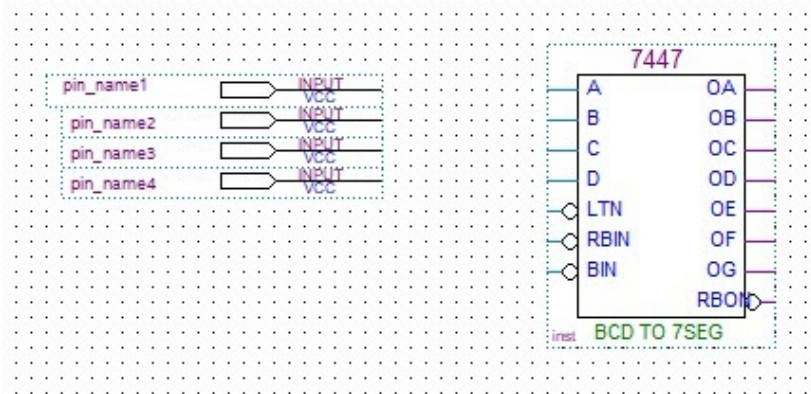
10. Pilih **Block Diagram/Schematic File**, kemudian klik OK.



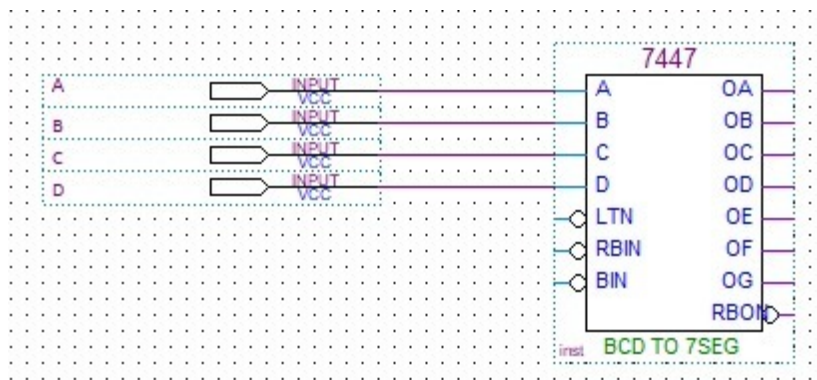
11. Cari komponen BCD to 7 Segment dengan cari "7447" pada **Symbol Tool**



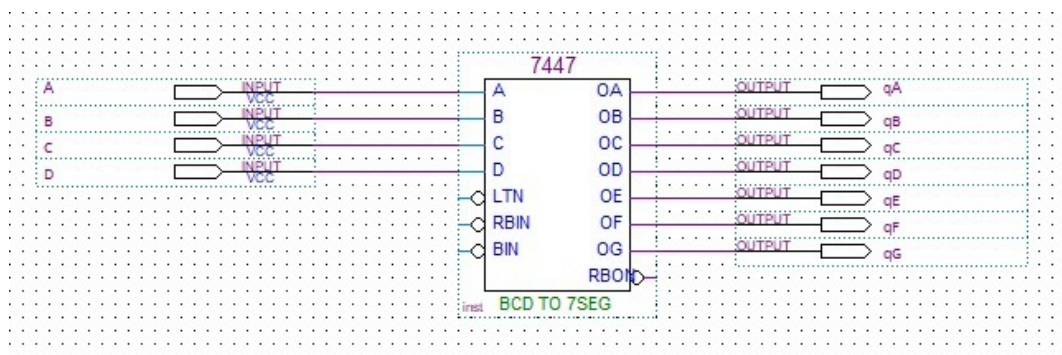
12. Masukkan 4 pin input pada project board



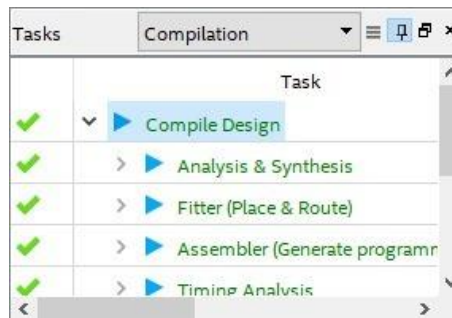
13. Sambungkan pin input pada komponen BCD to 7 Segment pada pin A, B, C, D



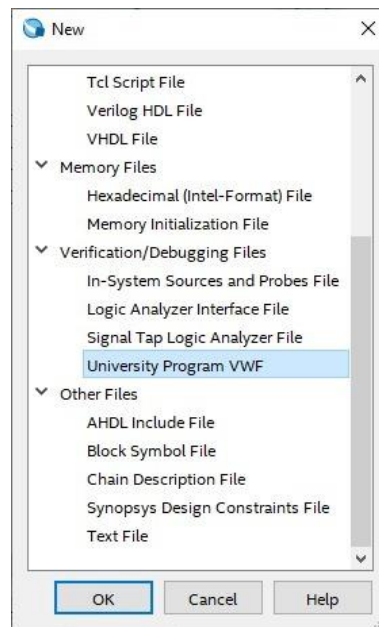
14. Pasang 7 pin output pada komponen BCD to 7 Segment



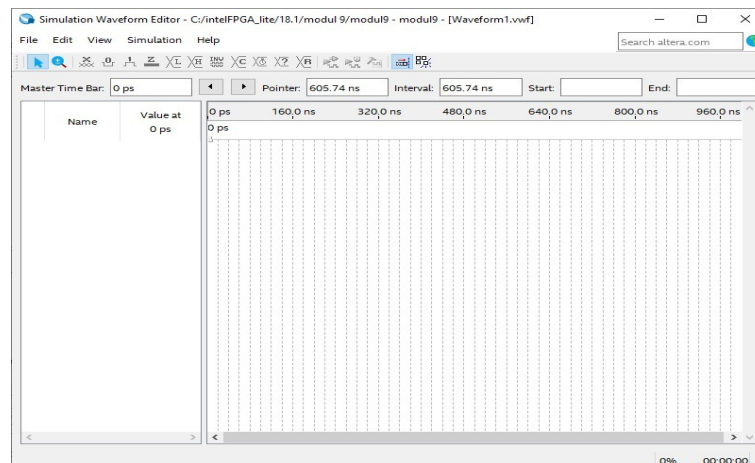
15. Compile hasil rangkaian dengan cara pilih compile design pada bagian task >> klik kanan → start >> yes → tunggu compiling hingga sukses



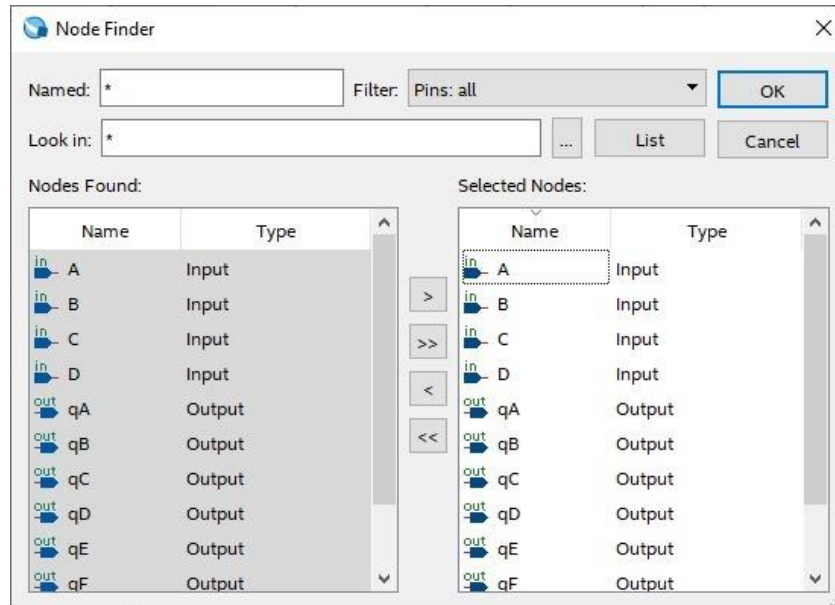
16. Lalu masuklah ke **University Program VWF** dengan CTRL+N



17. Lalu akan muncul tampilan seperti ini



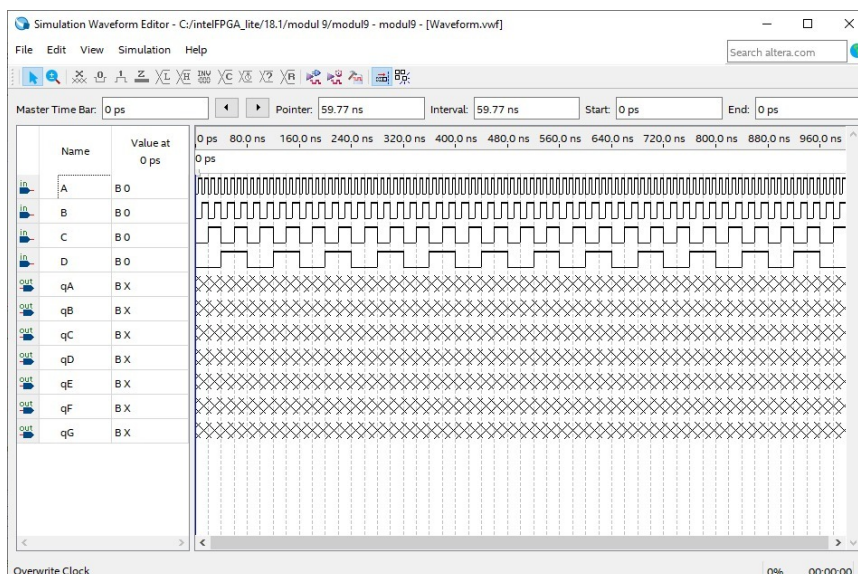
18. Lalu buka **Edit** → **Insert** → **Insert Node Bus** → **Node Finder** → **List** lalu klik tanda ('>>')



19. Masukan A : clock periode 10ns dan duty cycle 50%
Masukan B : clock periode 20ns dan duty cycle 50%
Masukan C : clock periode 40ns dan duty cycle 50%
Masukan D : clock periode 80ns dan duty cycle 50%



20. Akan muncul hasil seperti ini

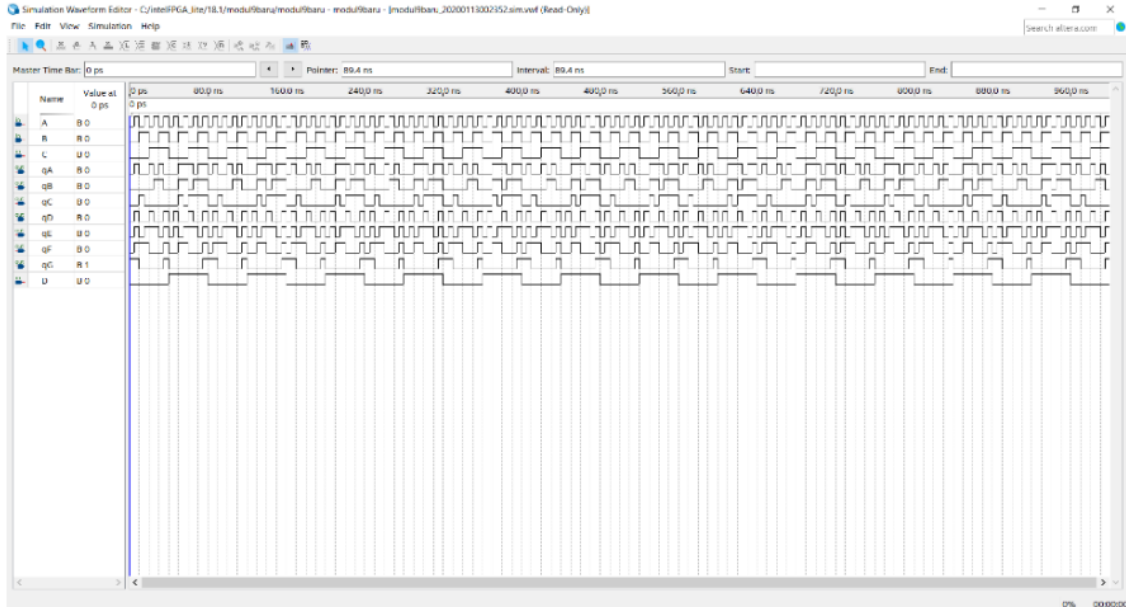


Modul Praktikum

21. Lalu buka **Simulation** → **Run Timing Simulation**

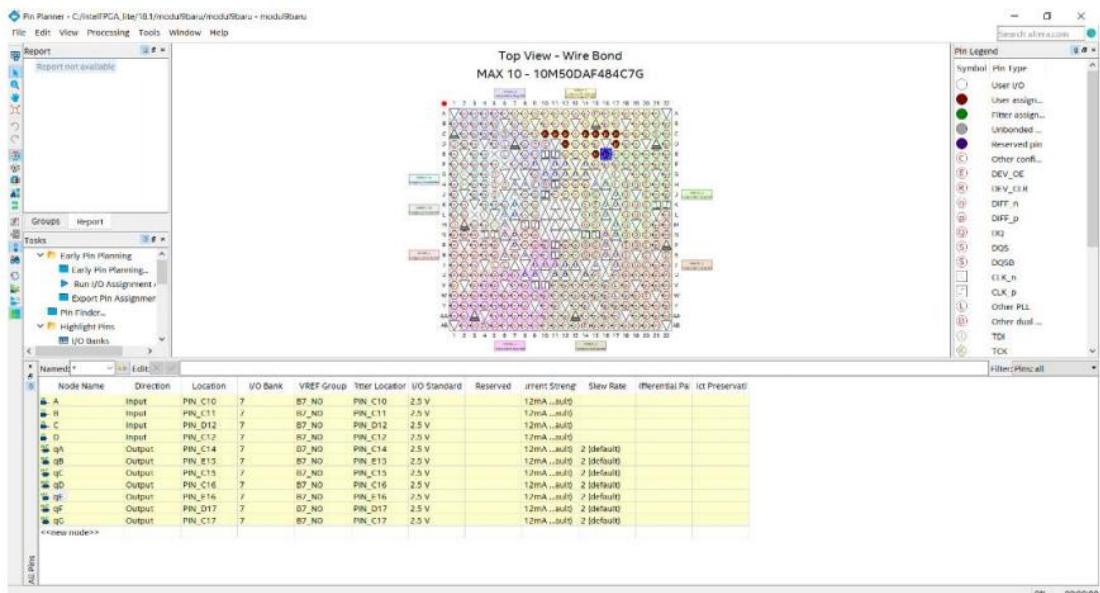
22. Tunggu progress flow simulation selesai

23. Akan muncul seperti ini



24. Klik pin planner

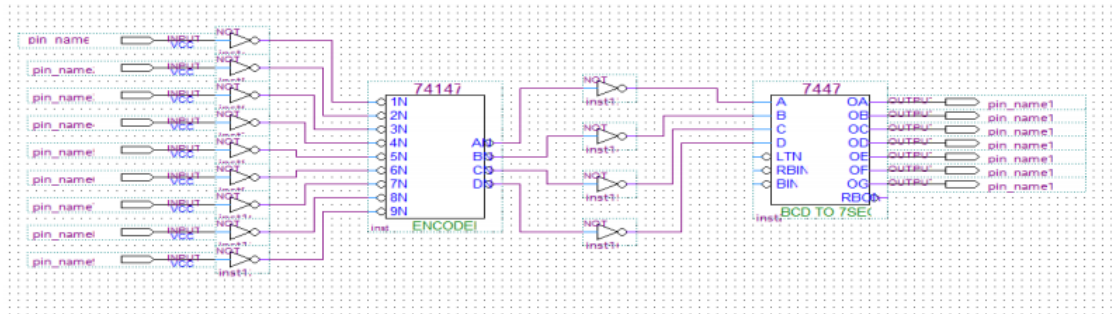
25. Lalu masukan pin seperti ini



26. Lakukan **Compile Design**, setelah selesai proses **Compile Design** maka langkah praktikum modul 8 Rangkaian Decoder telah selesai

9.4 Soal Jurnal

1. Apa yang dimaksud dengan BCD to 7 Segment?
2. Buatlah rangkaian BCD to 7 Segment seperti ini pada software Quartus!



3. Isilah keluaran 7 Segment pada table berikut!

A	B	C	D	Keluaran 7 segment
0	1	0	1	
1	1	0	1	
1	1	1	1	
0	0	0	1	
0	0	1	1	
1	0	0	1	
0	1	1	0	
1	0	0	0	
1	0	1	1	

4. Tuliskan apa yang telah dilakukan pada praktikum modul 9 menggunakan Bahasa kalian sendiri!

MODUL 10
SIMULASI *COUNTER* DAN *REGISTER*

10.1 Tujuan Praktikum Modul 10 :

Setelah mempraktekkan topic ini, praktikan diharapkan dapat :

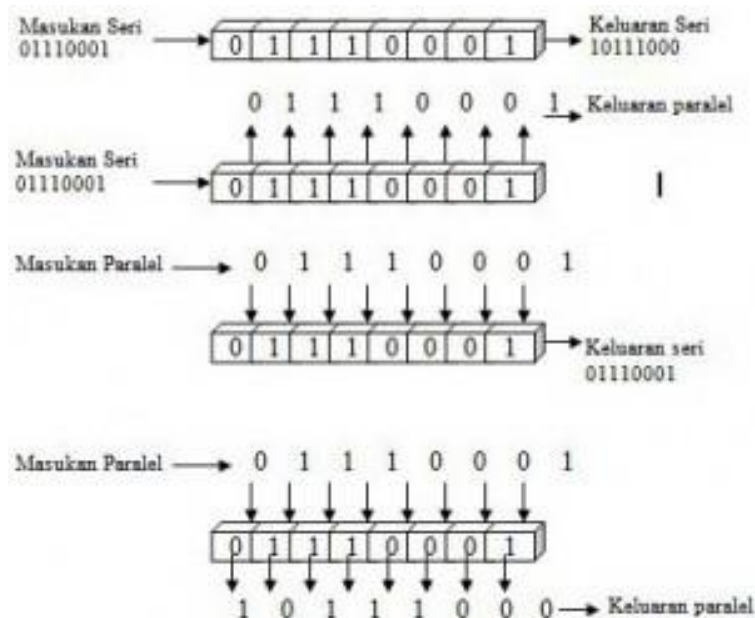
1. Mengetahui dan memahami konsep dasar dari rangkaian *counter* dan *register* serta dapat membedakan jenis-jenis pada counter dan register.
2. Dapat membuat rangkaian *counter* dan *register* pada quartus prime lite.

10.2 Dasar Teori Praktikum Modul 10

10.2.1 Register

Register merupakan rangkaian untuk menyimpan data per bit. Register tersusun dari rangkaian flip-flop yang digunakan untuk menyimpan data sementara sebelum data diolah lebih lanjut, register juga digunakan untuk pergerakan/transmisi data pada operasi computer. Salah satu implementasi register adalah shift register atau register penggeser. Rangkaian shift register berfungsi untuk menyimpan data sementara dan untuk pergeseran data ke kiri atau ke kanan. Shift register juga terdapat beberapa macam yaitu PIPO, SISO, SIPO, PISO.

Gambar 10.1 Pergeseran Data Pada Register Geser

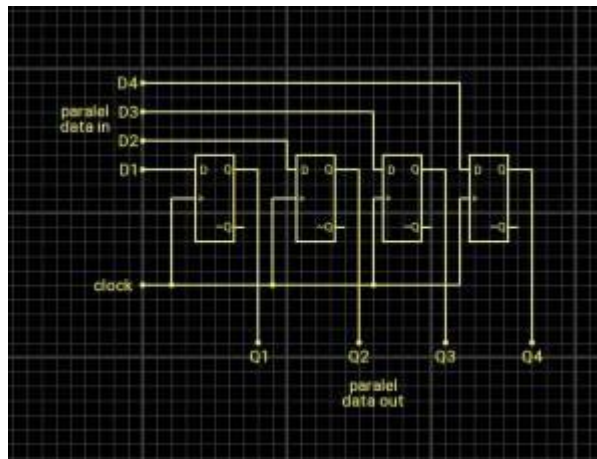


Macam-macam tipe Shift Register :

1. Register Parallel In Parallel Out (PIPO)

Register parallel in parallel out (PIPO) merupakan register geser yang input dan outputnya parallel, register geser PIPO akan mengubah format nilai dari data yang digeser dengan format data tetap parallel. Contoh : IC TTL 74LS174

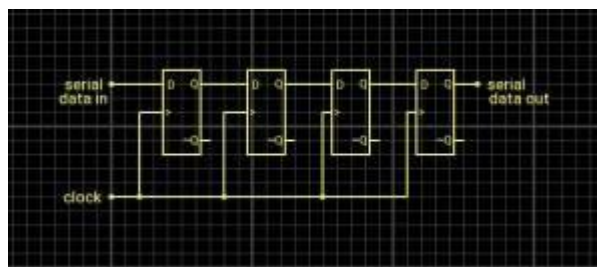
Gambar 10.2 Register Parallel In Parallel Out (PIPO)



2. Register Serial In Serial Out (SISO)

Register serial in serial out (SISO) merupakan register yang input dan outputnya seri. Register SISO tidak mengubah format data, yang berubah adalah nilai dari data tersebut. Contoh : IC TTL 74LS91

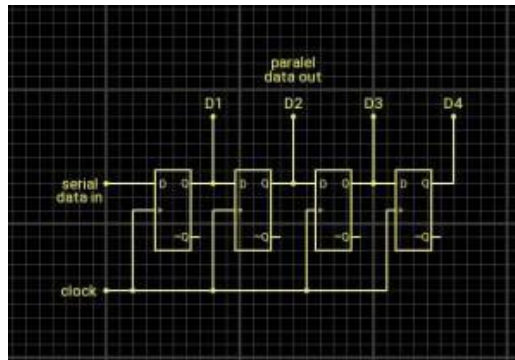
Gambar 10.3 Register Serial In Serial Out (SISO)



3. Register Serial In Parallel Out (SIPO)

Register serial in parallel out (SIPO) merupakan register geser yang inputnya seri dan output parallel. Register ini akan menggeser data secara seri dan mengeluarkannya dalam format parallel tanpa mengubah nilai data tersebut. Contoh : IC TTL 74LS164

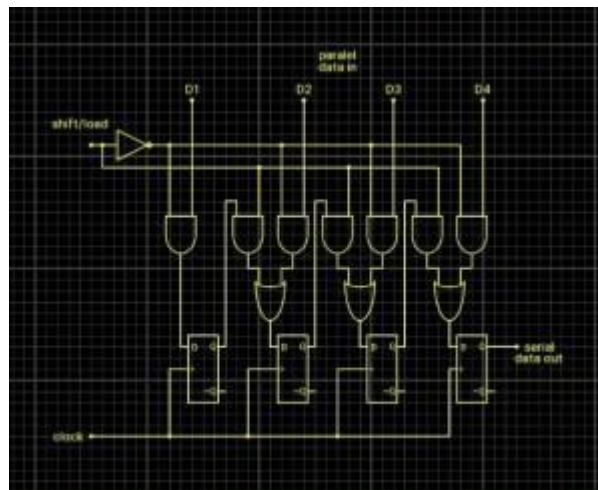
Gambar 10.4 Register Serial In Parallel Out (SIPO)



4. Register Parallel In Serial Out (PISO)

Register parallel in serial out (PISO) merupakan register geser yang inputnya parallel dan output seri. Register ini hanya mengubah format data parallel menjadi output serial tanpa mengubah nilai dari data tersebut.

Gambar 10.5 Register Parallel In Serial Out (PISO)



10.2.2 Counter

Counter berfungsi sebagai pencacah bit, digunakan untuk menghitung banyaknya pulsa yang dimasukkan pada suatu rangkaian digital.

Jenis-jenis counter :

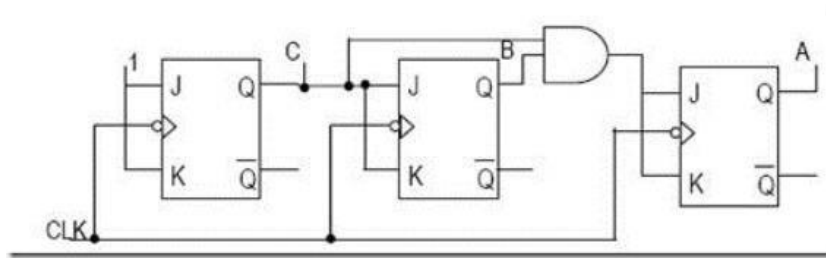
1. Counter Up

Counter up dapat menghitung secara berurutan dari bilangan terkecil sampai bilangan terbesar atau bisa juga disebut sebagai penghitung maju. Up Counter dibagi menjadi 2 :

a. Synchronous Up

Synchronous up merupakan penghitung maju dimana setiap flip-flopnya menerima input secara bersamaan karena counter sinkron dirangkai secara parallel.

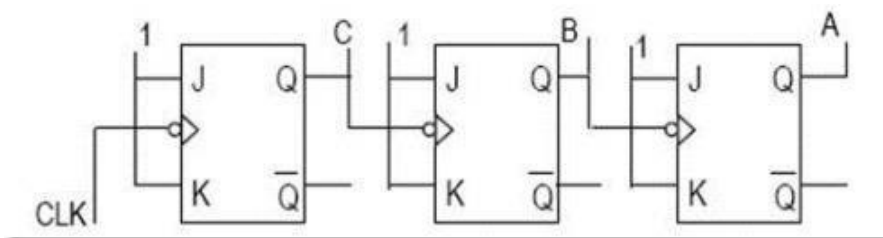
Gambar 10.6 Synchronous Up Counter



b. Asynchronous Up

Asynchronous up merupakan penghitung maju dimana flip-flop dirangkai secara seri sehingga flip-flopnya menerima clock dari sumber yang berbeda, keluaran tiap flip-flop digunakan sebagai clock untuk flip-flop berikutnya secara berurutan.

Gambar 10.7 Asynchronous Up Counter



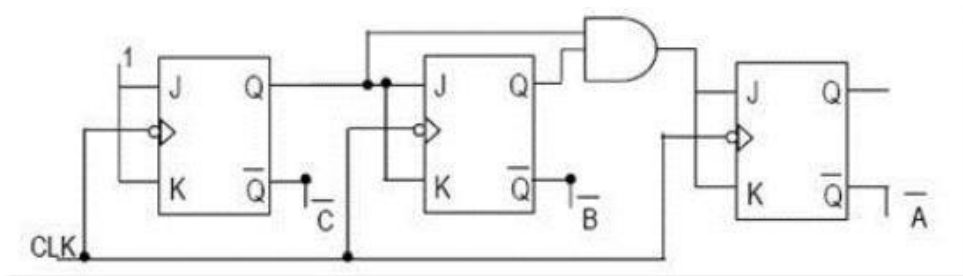
2. Counter Down

Counter down dapat menghitung dari nilai yang sudah ditentukan ke nilai terkecil atau bisa disebut sebagai penghitung mundur. Down Counter dibagi menjadi 2 :

1. Synchronous Down

Synchronous down merupakan penghitung mundur dimana setiap flip-flopnya menerima input secara bersamaan karena counter sinkron dirangkai secara parallel.

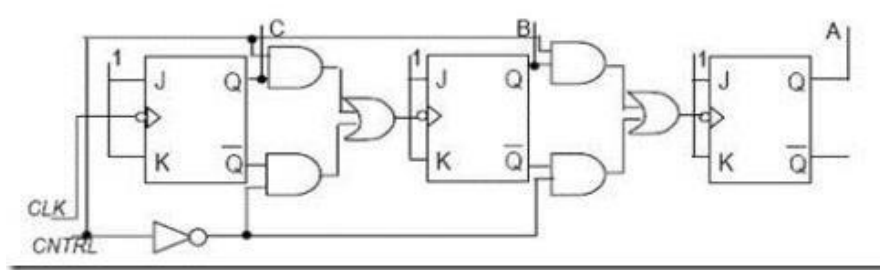
Gambar 10.8 Synchronous Down Counter



2. Asynchronous Down

Asynchronous down merupakan penghitung mundur dimana flip-flop dirangkai secara seri sehingga flip-flopnya menerima clock dari sumber yang berbeda, keluaran tiap flip-flop digunakan sebagai clock untuk flip-flop berikutnya secara berurutan.

Gambar 10.9 Asynchronous Down Counter



3. Counter Up dan Down

Counter up dan down merupakan rangkaian yang menggunakan perhitungan maju dan mundur.

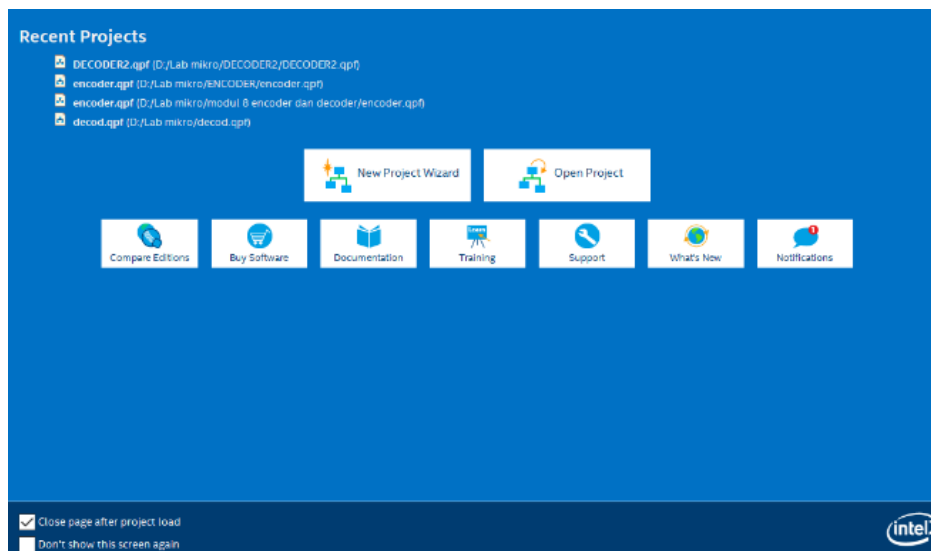
10.3 Lembar Kegiatan Praktikum Modul 10 :

10.3.1 Alat dan Bahan

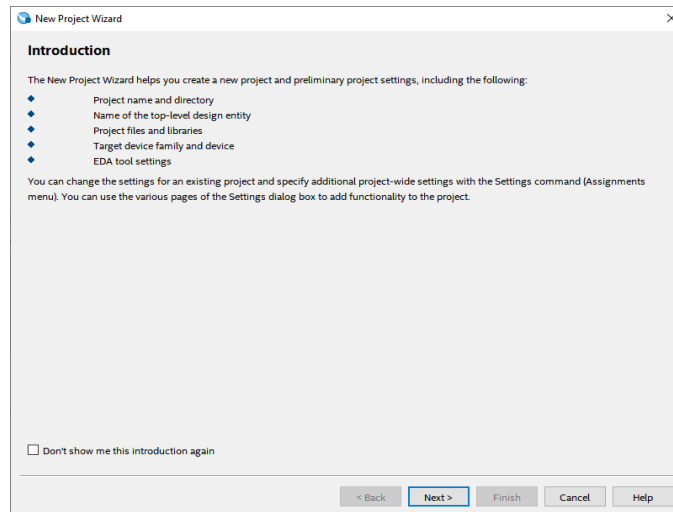
1. Mouse
2. Laptop
3. Quartus Prime Lite

10.3.2 Langkah Praktikum Modul 10

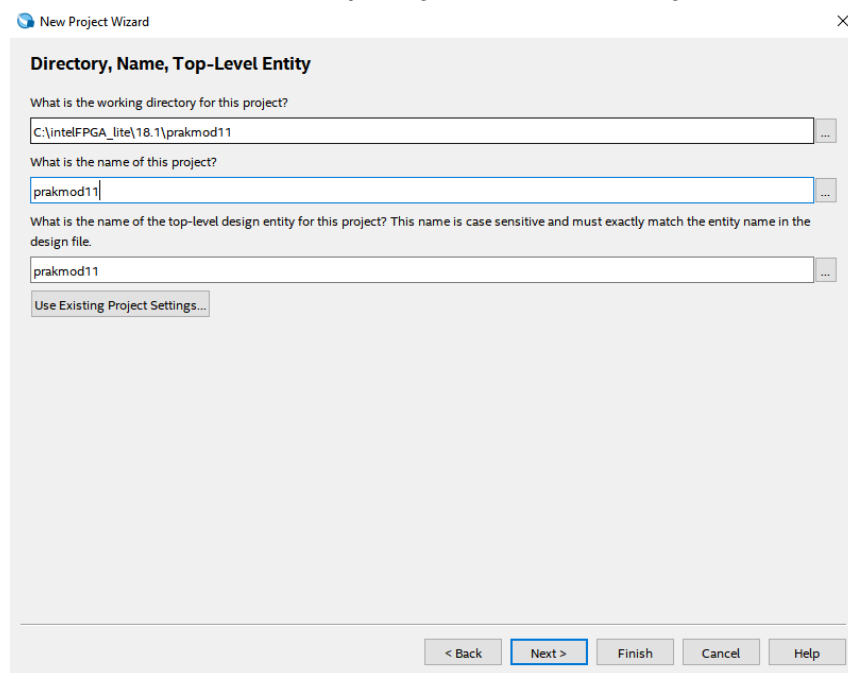
1. Buka software quartus prime lite, lalu klik **New Project Wizard**.



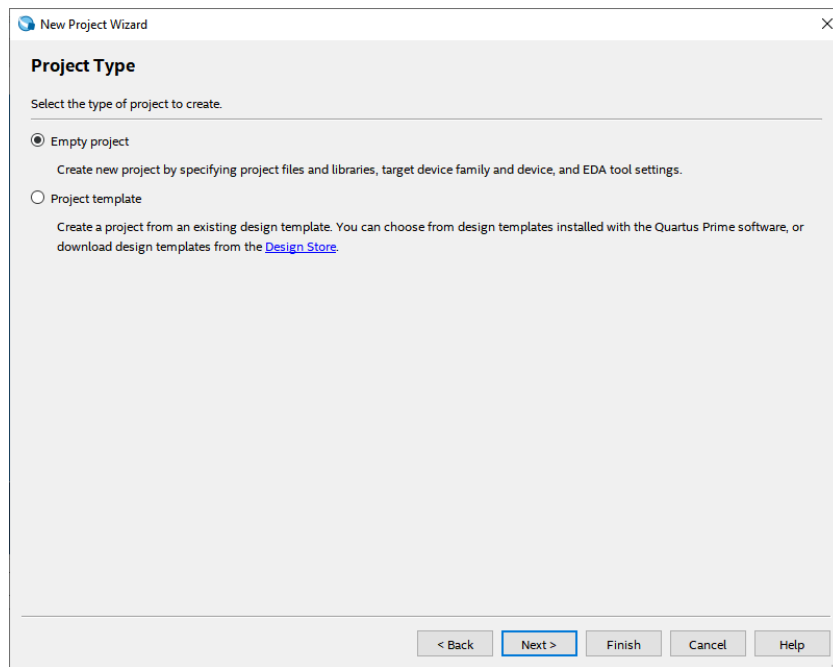
2. Klik Next



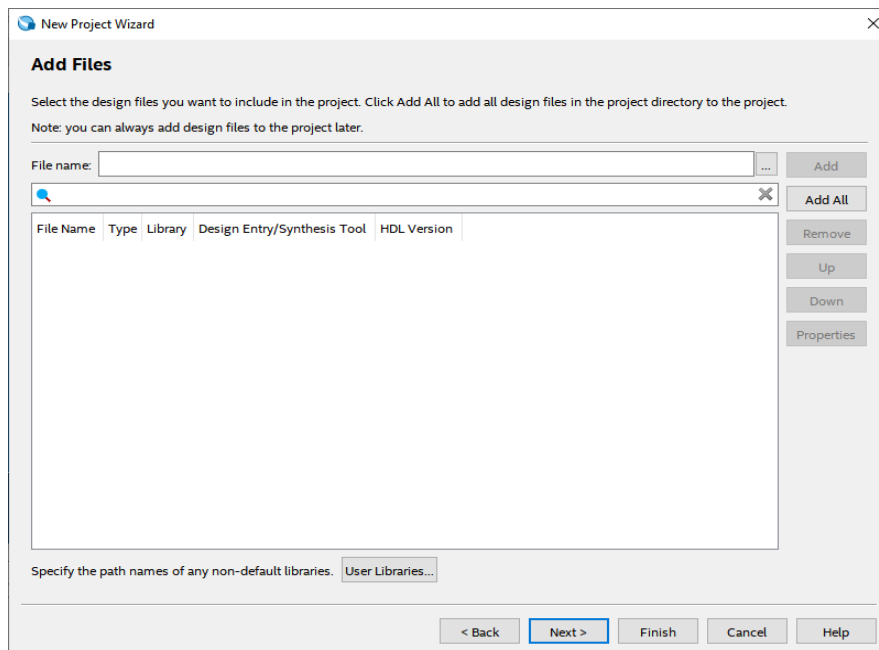
3. Kemudian tentukan **Directory Project** dan **Nama Project**, lalu klik **Next**.



4. Pilih **Empty Project**, kemudian klik **Next** lagi



5. Setelah itu, klik **Next** lagi



- Ganti family ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketik nama filter **10M50DAF484C7G**, lalu klik available device yang tersedia, terakhir klik **Next**.

Family, Device & Board Settings

Device | Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.
To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family:
Family: MAX 10 (DA/DF/DC/SA/SF/SC)
Device: All

Target device:
 Auto device selected by the Fitter
 Specific device selected in 'Available devices' list
 Other: n/a

Show in 'Available devices' list:
Package: Any
Pin count: Any
Core speed grade: Any
Name filter: 10M50DAF484C7G
 Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-b
10M50DAF484C7G	1.2V	49760	360	360	1677312	288

< Back | **Next >** | Finish | Cancel | Help

- Klik **Next**

EDA Tool Settings

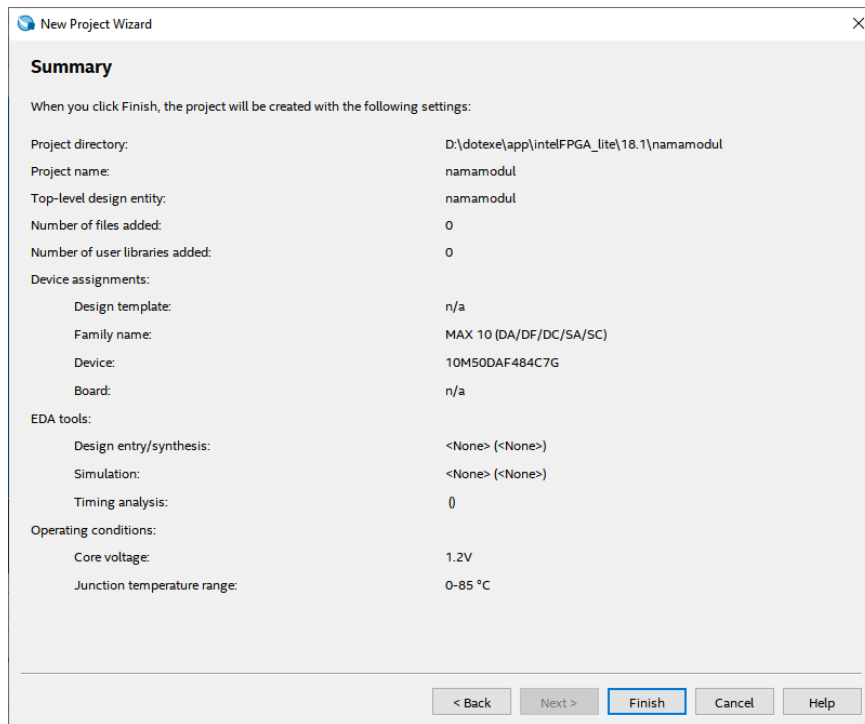
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

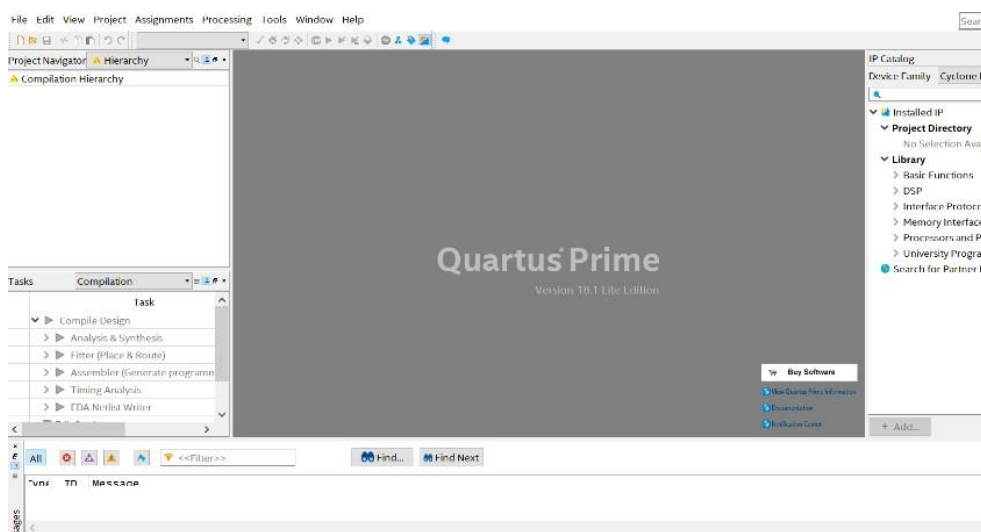
Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synth...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	<None>	<None>	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back | **Next >** | Finish | Cancel | Help

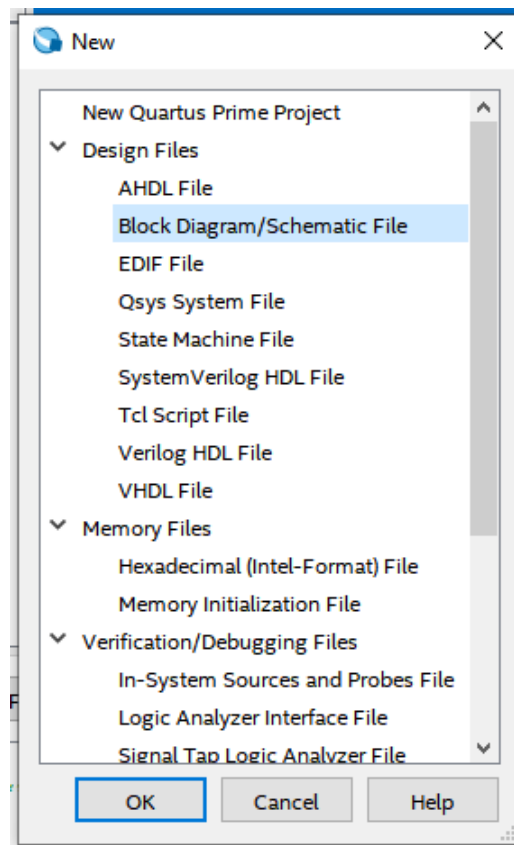
- Setelah itu klik **Finish**.



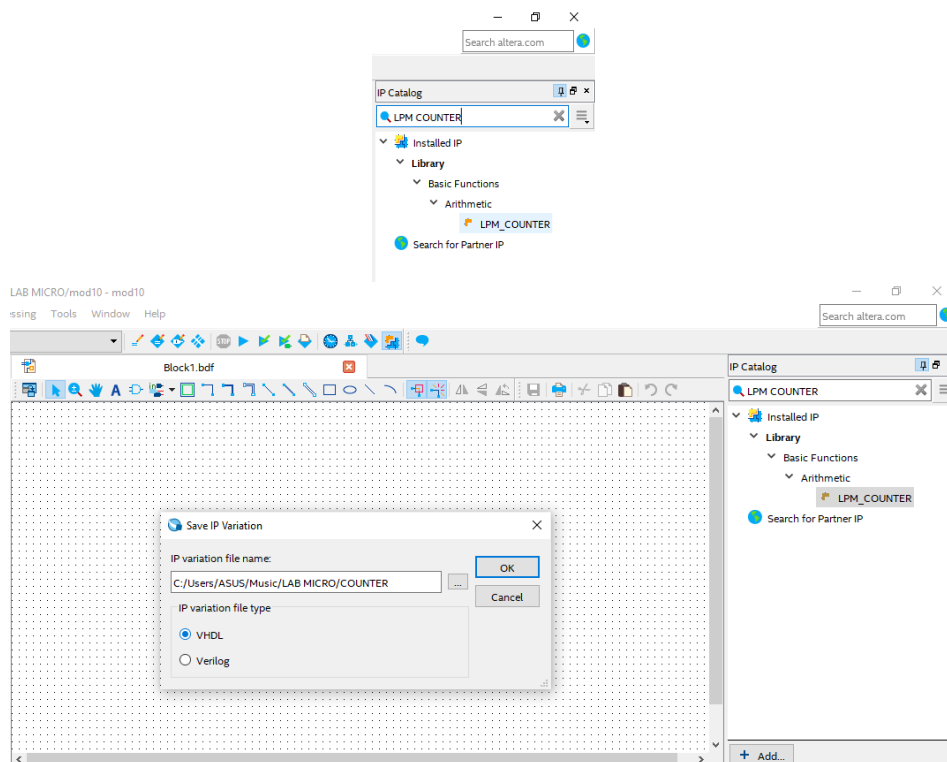
9. Akan muncul tampilan seperti gambar di dibawah ini.



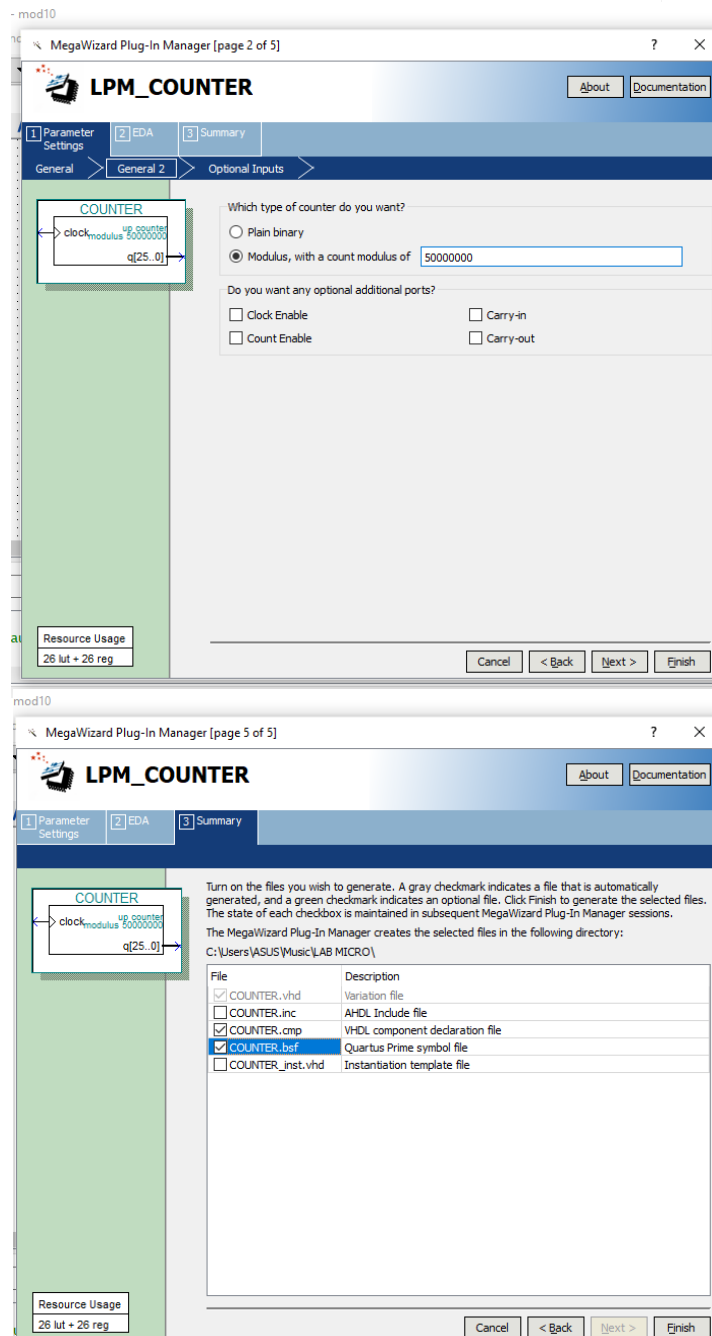
10. Kemudian buat file baru dengan cara klik **File**, kemudian pilih **Block Diagram/Schematic**, lalu klik **OK**



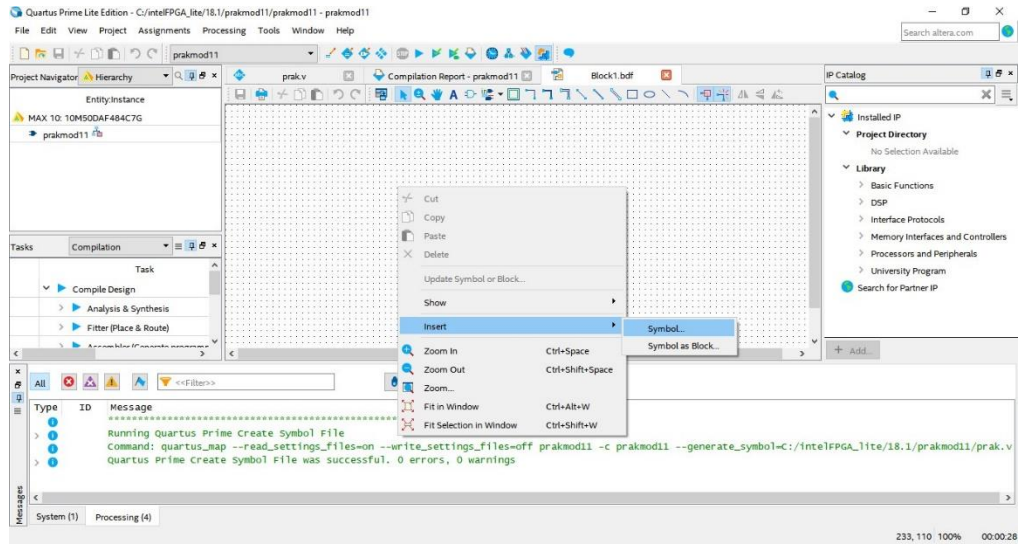
11. Setelah workspace muncul, tulislah **LPM COUNTER** pada searching di sebelah kanan, lalu klik **LPM_COUNTER**, lalu akan muncul save IP variation, pilih tempat penyimpanan, kemudian klik **VHDL**, lalu **OK**.



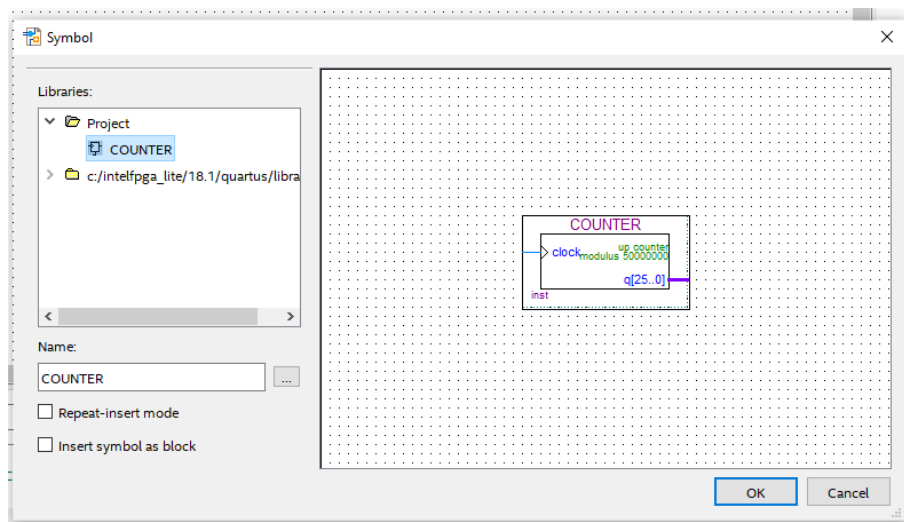
12. Akan muncul gambar seperti di bawah ini, ubah bits menjadi **26**, moduls menjadi **50000000**, dan centang **.bsf**. Ikuti seluruh gambar di bawah.



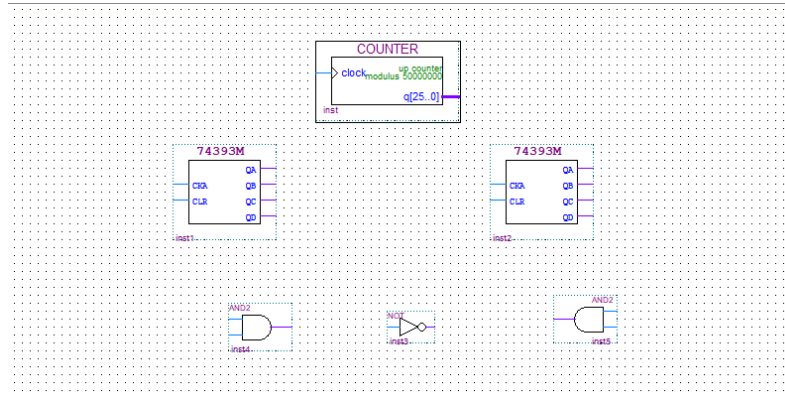
13. klik kanan pada Workspace, kemudian klik **Insert** lalu **Symbol**.



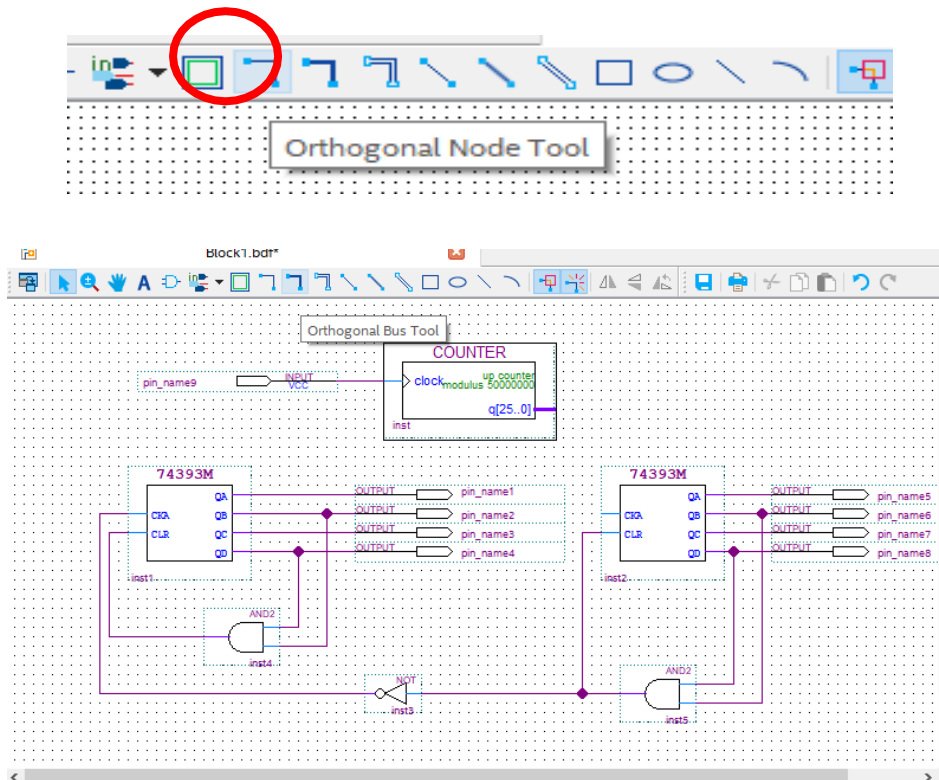
14. Klik **Project**, klik counter yang telah dibuat, lalu klik **OK**.

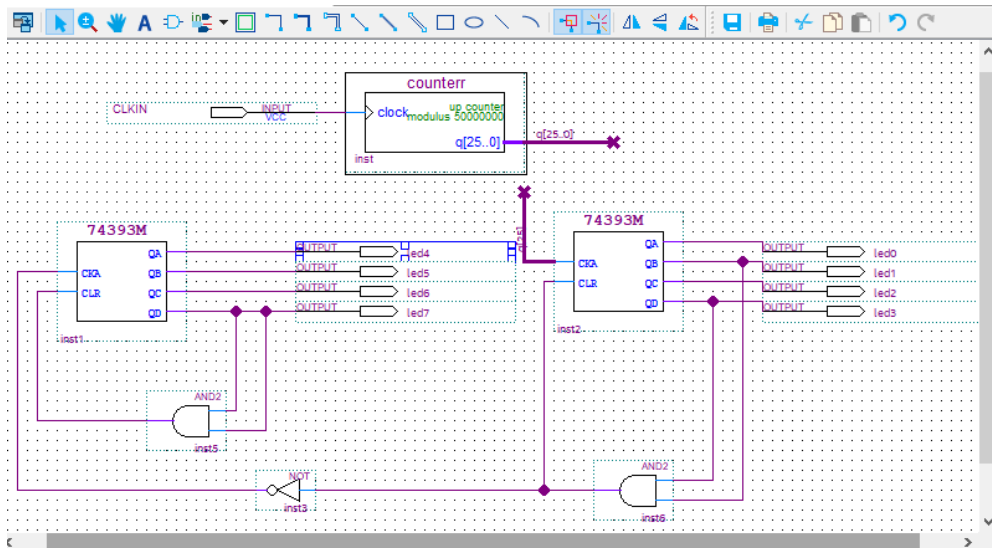


15. Masukkan komponen-komponen (74393M, AND, NOT) dengan cara klik kanan lalu **Insert, Symbol** dan tulis nama komponen yang dicari.

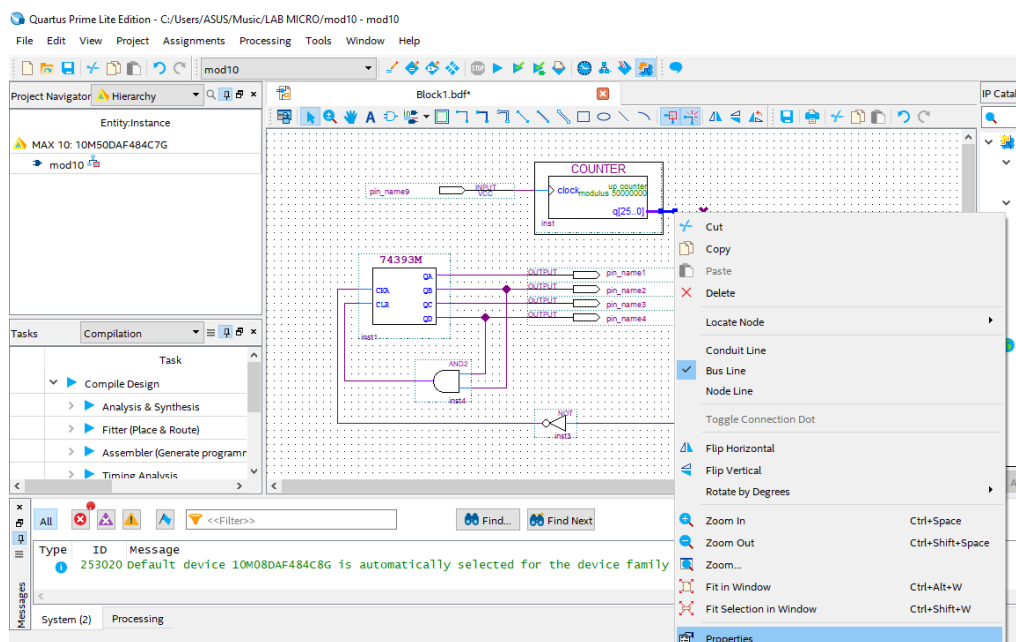


16. Sambungkan komponen-komponen dengan **Orthogonal Node Tool**, dan gunakan **Orthogonal Bus Tool** untuk garis yang tebal, pasang juga output dan input seperti gambar di bawah.

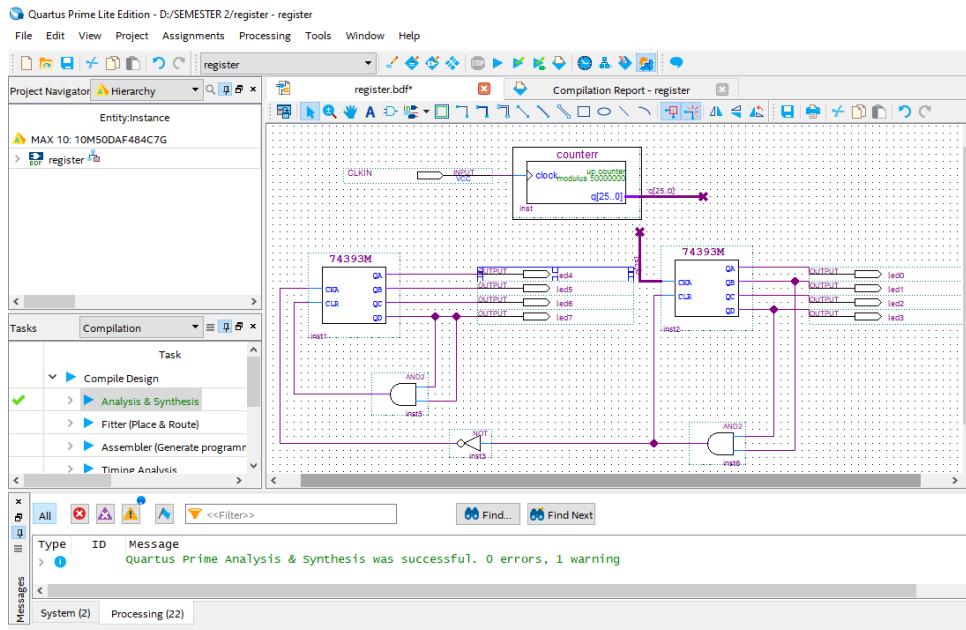




17. Ubah orthogonal bus dengan cara klik kanan lalu **Properties**. Untuk orthogonal bus pada counter tuliskan “**q[25..0]**”, sedangkan pada 74393M tuliskan “**q[25]**”.



18. Ganti nama output menjadi LED0-LED7 dan input menjadi CLKIN. Lalu klik Analysis & Synthesis. Setelah berhasil tanpa error maka praktikum modul 10 telah selesai.



10.4 Soal Jurnal

1. Jelaskan apa yang dimaksud dengan register dan counter! Menurut pemahaman kalian sendiri.
2. Jelaskan perbedaan asynvchronous dan synchronous counter!
3. Jelaskan perbedaan PIPO, SISO, SIPO, PISO dan contoh-contohnya!
4. Tuliskan apa yang telah dilakukan pada praktikum modul 10 menggunakan Bahasa kalian sendiri!

MODUL 11
SIMULASI RANGKAIAN *ENCODER* DAN *DECODER*
(PERANCANGAN BERBASIS *CODE*)

11.1 Tujuan Praktikum Modul 11 :

Setelah mempraktekkan topik ini, praktikan diharapkan dapat:

1. Dapat mengetahui dan memahami konsep dasar dari rangkaian *decoder* dan *encoder*.
2. Dapat mensimulasikan rangkaian *decoder* dan *encoder* dengan perancangan berbasis *code* pada *software Quartus Prime Lite*.

11.2 Dasar Teori Praktikum Modul 11:

11.2.1 *Encoder*

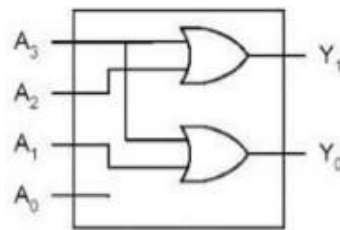
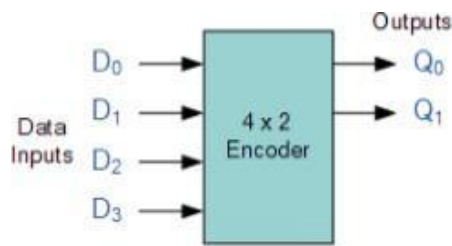
Encoder adalah sebuah rangkaian yang dapat berfungsi mengkonversi suatu data kedalam bentuk data baru. *Encoder* dibutuhkan untuk mengkonversi data dari suatu sistem bilangan ke sistem bilangan lainnya. Prinsip kerja *Encoder* adalah mengkonversi suatu data agar data tersebut dapat diterima oleh *receiver* dalam keadaan utuh. Dimana pada bagian penerima terdapat *decoder* yang dapat mengambil data yang telah dikonversi oleh *Encoder*. Jadi, cara kerja *Encoder* adalah kebalikan dari *Decoder*.

Encoder terdiri dari beberapa input line, hanya salah satu dari input-input tersebut diaktifkan pada waktu tertentu, yang selanjutnya akan menghasilkan kode output N-bit. Rangkaian *encoder* merupakan aplikasi dari gerbang OR. Sebuah *priority encoder* adalah rangkaian *encoder* yang mempunyai fungsi prioritas, operasi dari rangkaian *priority encoder* yaitu jika ada dua lebih input bernilai “1” pada saat yang sama, maka input yang mempunyai prioritas tertinggi yang akan diambil.

Dalam teori digital banyak ditemukan istilah *encoder* seperti “*Decimal to BCD Encoder*” yang berarti rangkaian digital yang berfungsi untuk mengkodekan *line input* dengan jumlah *line input* desimal (0-9) menjadi kode bilangan biner 4 bit BCD (*Binary Coded Decimal*). Atau “*8 line to 3 line encoder*” yang berarti rangkaian *encoder* dengan input 8 line dan *output 3 line* (3 bit BCD).

Contoh *Encoder*:

Gambar 11. 1 Encoder



Tabel 11. 1 Tabel kebenaran encoder

Inputs				Outputs	
D ₃	D ₂	D ₁	D ₀	Q ₁	Q ₀
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

11.2.2 Decoder

a. Definisi

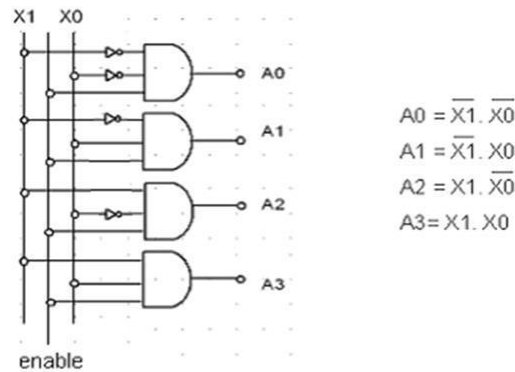
Decoder adalah alat yang digunakan untuk dapat mengembalikan proses *encoding* sehingga dapat terlihat atau menerima informasi aslinya. Pengertian *Decoder* juga dapat diartikan sebagai rangkaian logika yang ditugaskan untuk menerima *input* biner dan mengaktifkan salah satu *outputnya* sesuai dengan urutan biner tersebut.

b. Konsep Dasar

Fungsi *Decoder* adalah untuk memudahkan dalam menyalakan *seven segmen*. *Output* dari *decoder* maksimum adalah 2^n . Jadi dapat dibentuk n -to- 2^n *decoder*. Beberapa rangkaian *decoder* yang sering dijumpai saat ini adalah *decoder* jenis 3×8 (3 bit *input* dan 8 *output line*), *decoder* jenis 4×16 , *decoder* jenis *BCD to Decimal* (4 bit *input* dan 10 *output line*) dan *decoder* jenis *BCD to 7 segmen* (4 bit *input* dan 8 *output line*). Khusus untuk pengertian *decoder* jenis *BCD to seven segmen* mempunyai prinsip kerja yang berbeda dengan *decoder* lainnya, di mana kombinasi setiap *inputnya* dapat

mengaktifkan beberapa *output linenya*. Salah satu jenis IC *decoder* yang umum di pakai adalah 74138, karena IC ini mempunyai 3 *input* biner dan 8 *output* line, di mana nilai *output* adalah 1 untuk salah satu dari ke 8 jenis kombinasi inputnya. Decoder dapat dibentuk dari susunan gerbang logika dasar atau menggunakan IC.

Gambar 11. 2 Decoder



Tabel 11. 2 Tabel kebenaran decoder

X ₁	X ₀	A ₀	A ₁	A ₂	A ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Decoder 2 to 4 bekerja dikendalikan keluarannya oleh variasi X0 dan X1. Keluaran dari a0, a1, a2 dan a3 akan aktif secara bergantian tergantung dari kondisi X0 dan X1. Pada kondisi masukan X0 = 0 dan X1 = 0, maka gerbang AND pertama mendapatkan masukan “**high**”, maka kondisi keluaran A0 = 1 sementara keluaran yang lain “0”. Apabila X0 = 0 dan X1 = 1, maka hanya gerbang AND ke dua akan “1” sementara yang lain “0” dan seterusnya.

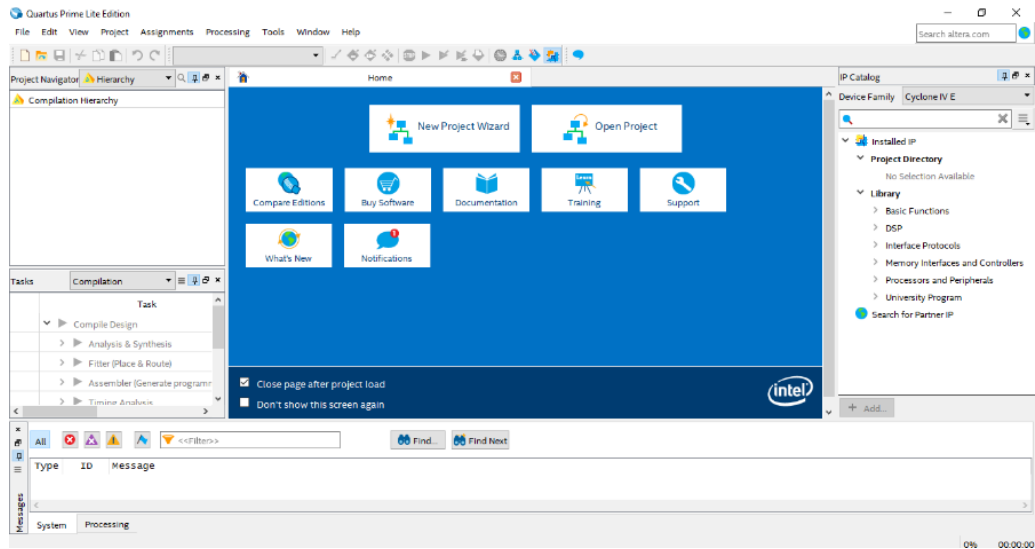
11.3 Lembar Kegiatan Praktikum Modul 11

11.3.1 Alat dan Bahan

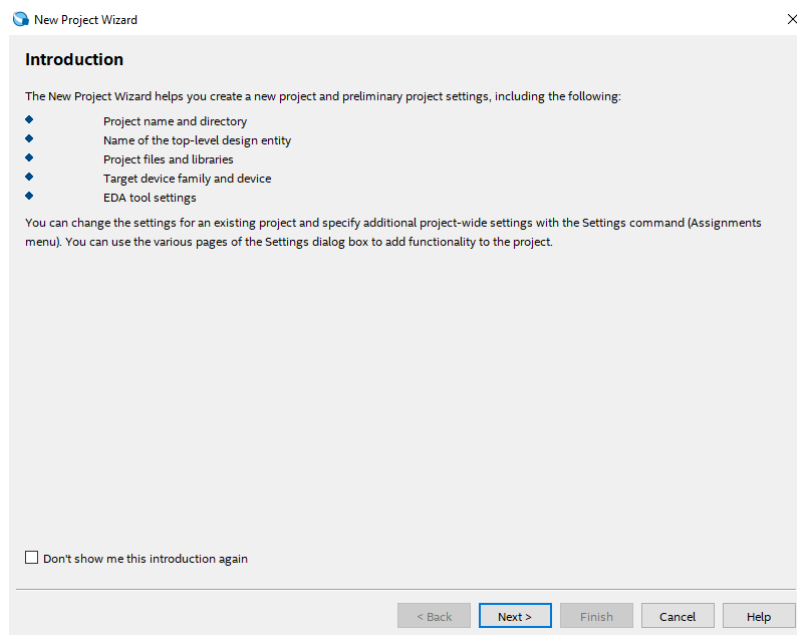
1. Laptop
2. Mouse
3. *Software Quartus Prime Lite*

11.3.2 Langkah Praktikum Modul 11 (Rangkaian *Encoder*)

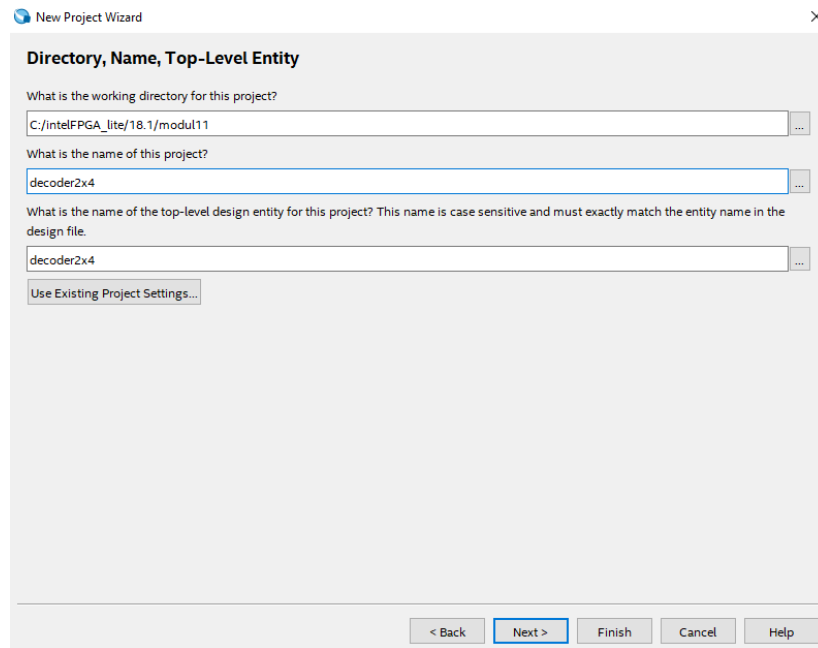
1. Buka software *Quartus lite* di laptop dan klik *New Project Wizard*.



2. Klik *Next*.

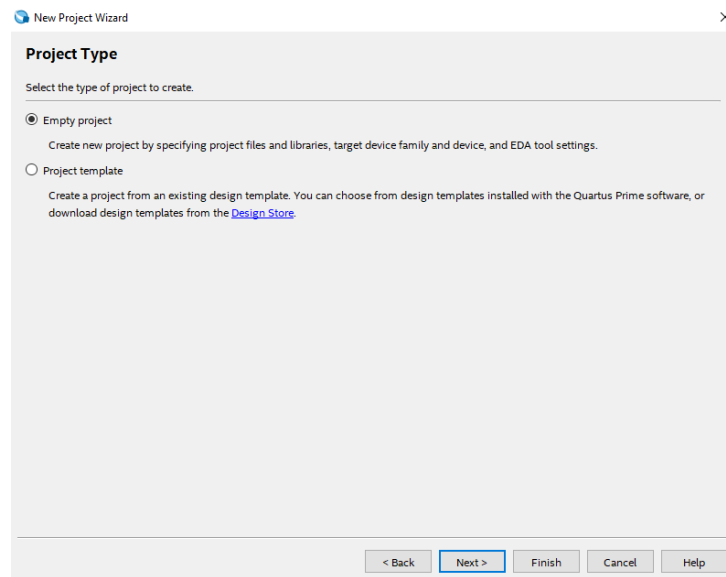


3. Kemudian tentukan *Directory Project* dan Nama *Project*, lalu klik *Next*.



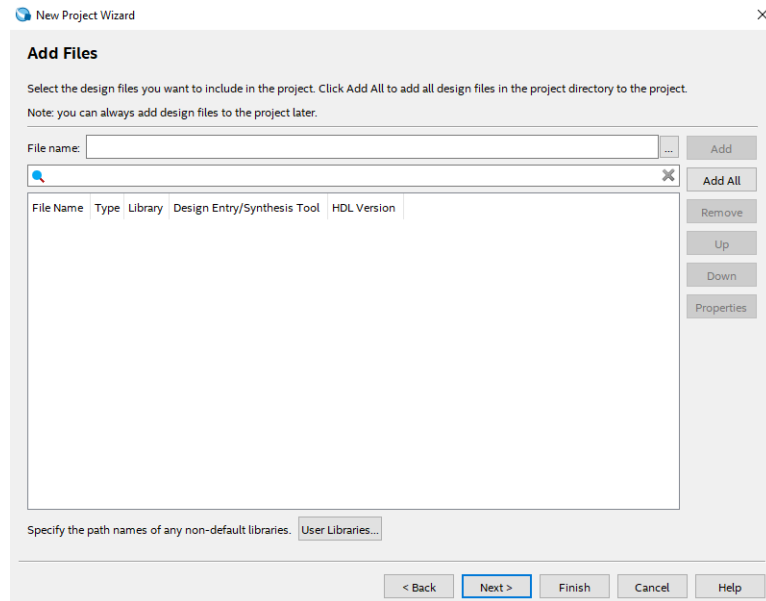
The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button 'X'. The main heading is 'Directory, Name, Top-Level Entity'. It contains three text input fields: 'What is the working directory for this project?' with the value 'C:/intelFPGA_lite/18.1/modul11', 'What is the name of this project?' with the value 'decoder2x4', and 'What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.' with the value 'decoder2x4'. Below these fields is a button labeled 'Use Existing Project Settings...'. At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

4. Pilih *Empty Project*, kemudian klik *Next* lagi.

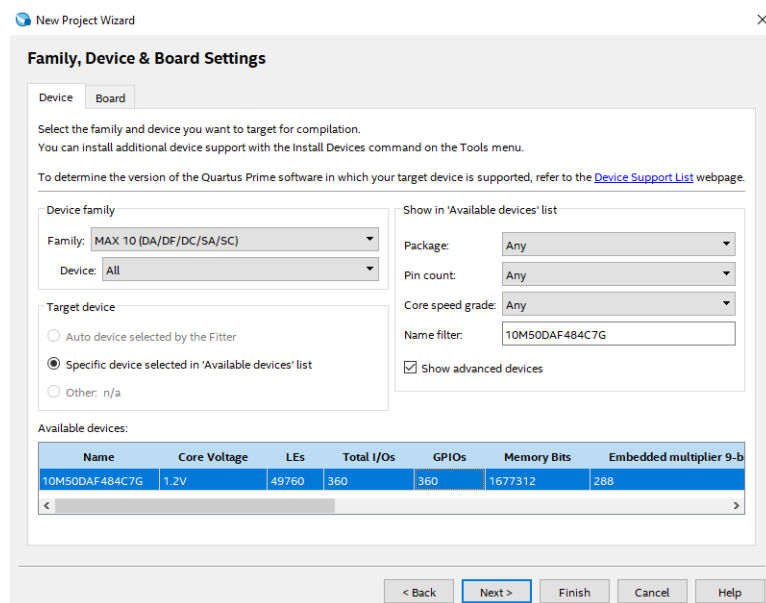


The screenshot shows the 'New Project Wizard' dialog box with the title 'New Project Wizard' and a close button 'X'. The main heading is 'Project Type'. It contains the text 'Select the type of project to create.' followed by two radio button options: 'Empty project' (which is selected) and 'Project template'. Below the 'Project template' option is a small text block: 'Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).' At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

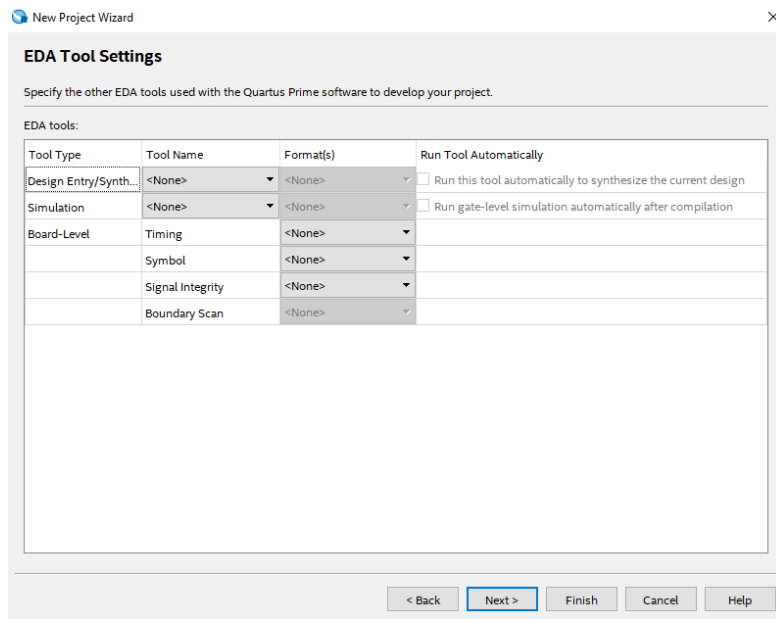
5. Setelah itu, klik *Next* lagi.



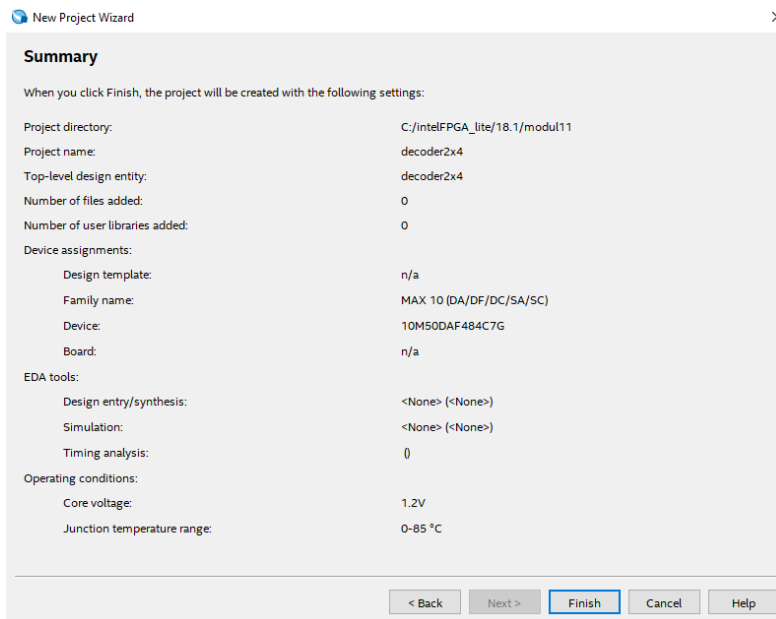
6. Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik *Available Device* yang tersedia, kemudian klik *Next*.



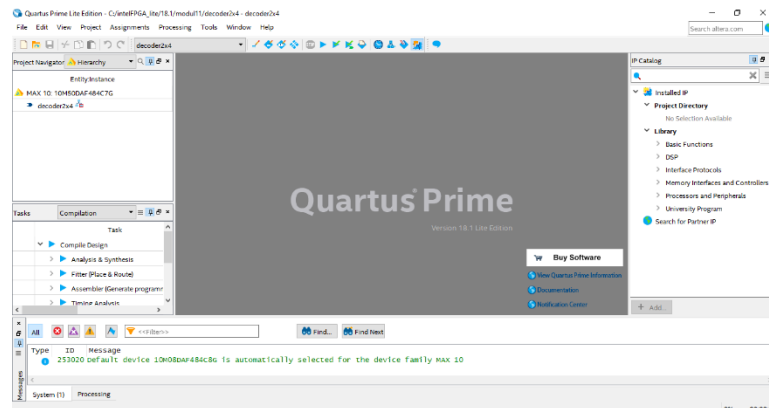
7. Kemudian klik *Next*.



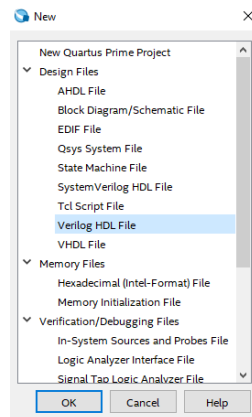
8. Lalu klik *Finish*.



9. Setelah itu akan muncul tampilan awal dari *Project Quartus Prime* seperti gambar di bawah ini.



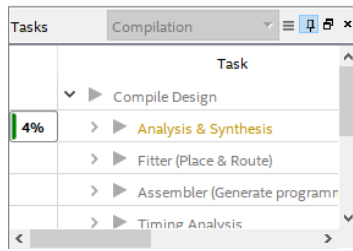
10. Kemudian buat file baru dengan cara klik **File** → **New**, lalu pilih **Verilog HDL File**, kemudian klik **OK**.



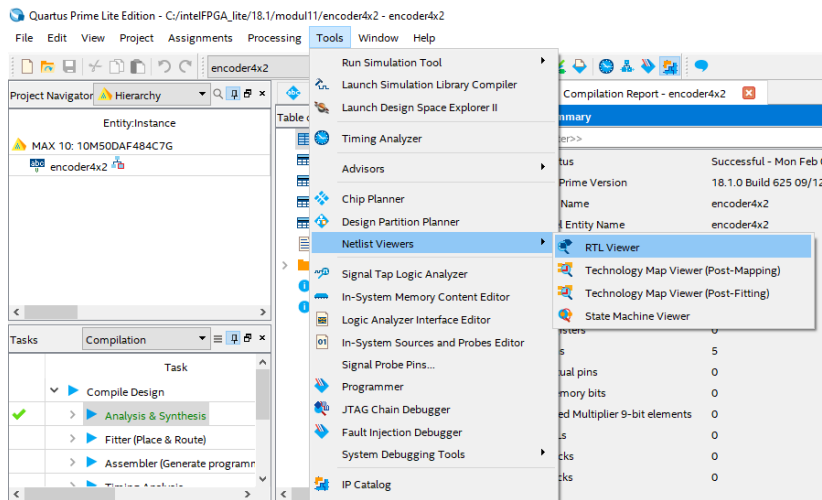
11. Setelah muncul workspace, tuliskan code Verilog seperti gambar dibawah ini, kemudian **Save**.

```
encoder4x2.v
1  module encoder4x2 (Y3, Y2, Y1, A0, A1);
2
3  input  Y1, Y2, Y3;
4  output A0, A1;
5
6  wire  w1, w2;
7
8  or(A1, w1, Y3, Y2);
9  or(A0, w2, Y3, Y1);
10
11 endmodule
12
```

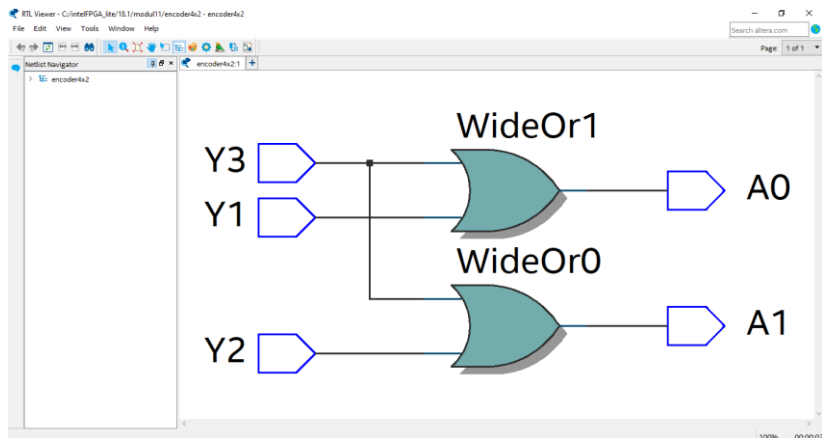
12. Kemudian klik dua kali pada **Analysis & Synthesis**.



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

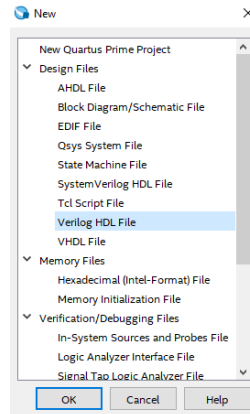


14. Selesai.



11.3.3 Langkah Praktikum Modul 11 (Rangkaian *Decoder*)

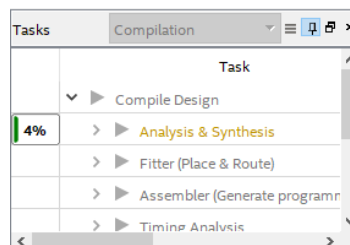
1. Kita masuk pada rangkaian *decoder* dengan membuat file baru. Klik **File** → **New**, lalu pilih **Verilog HDL File**, kemudian klik **OK**.



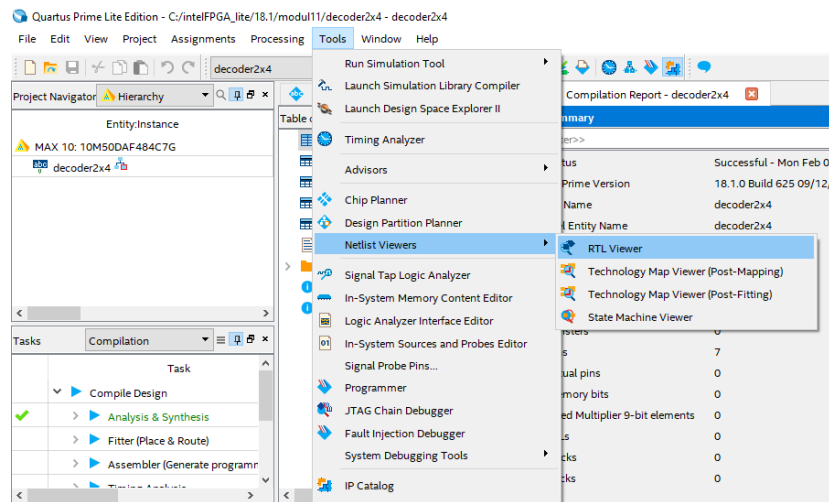
2. Setelah muncul workspace, tulislah code Verilog seperti gambar dibawah ini, kemudian jangan lupa untuk di **Save**.

```
1 module decoder2x4(a,b,en);
2   input [1:0] b;
3   input en;
4   output [3:0] a;
5   wire [1:0] bb;
6   not (bb[1], b[1]); (bb[0], b[0]);
7   and (a[0],en,bb[1],bb[0]), (a[1],en,b[1],bb[0]), (a[2],en,bb[1],b[0]), (a[3],en,b[1],b[0]);
8 endmodule
```

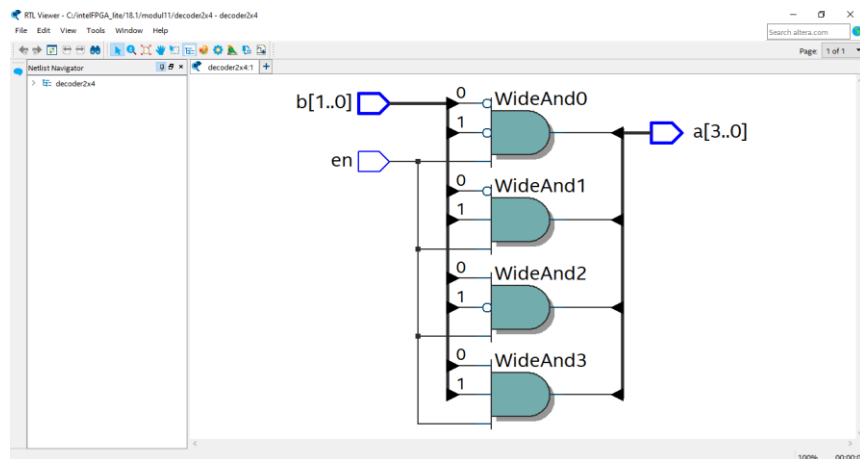
3. Kemudian, klik dua kali pada **Analysis & Synthesis**.



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



5. Selesai.



11.4 Soal Jurnal

1. Jelaskan prinsip kerja *Encoder* dan *Decoder*!
2. Sebutkan jenis-jenis *Encoder* dan *Decoder*!
3. Tuliskan apa yang telah dilakukan pada praktikum modul 11 menggunakan Bahasa kalian sendiri!

MODUL 12
SIMULASI COUNTER DAN REGISTER
(PERANCANGAN BERBASIS CODE)

12.1 Tujuan Pratikum Modul 12 :

Dengan mengikuti praktikum ini kita dapat :

1. Dapat mengetahui konsep dasar dari simulasi counter dan register (perancangan berbasis code)
2. Dapat mensimulasikan counter dan register dengan perancangan berbasis code

12.2 Dasar Teori Pratikum Modul 12

12.2.1 Counter

Counter atau pencacah merupakan register yang menunjukkan urutan tertentu yang perubahannya mengikuti pulse-pulse input . Sesuai dengan namanya counter berfungsi sebagai pencacah bit.

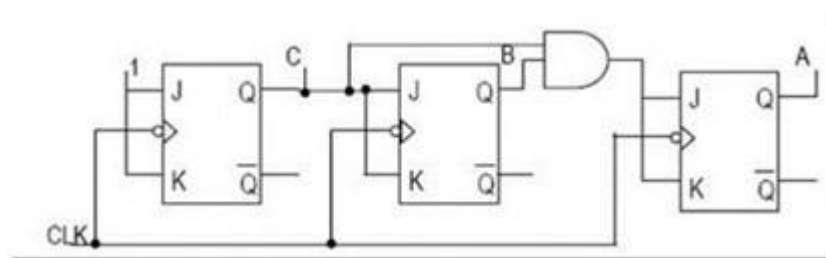
Jenis-jenis Counter :

1. Counter Up

Counter up adalah rangkaian yang berfungsi sebagai penghitung maju. Counter up dibagi menjadi 2 :

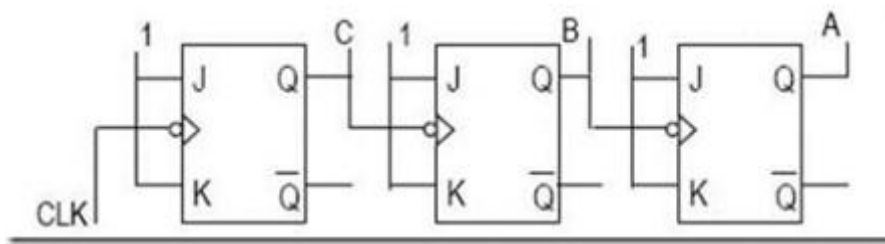
- a. Synchronous Up Counter adalah rangkaian penghitung maju yang pemicuannya terjadi serempak (dipicu oleh satu sumber clock) dengan susunan flipflopnya parallel.

Gambar 12.1 Synchronous Up Counter



- b. Asynchronous Up Counter adalah rangkaian perhitungan maju yang pemicuannya oleh sumber clock lain dengan susunan flipflopnya seri.

Gambar 12.2 Asynchronous Up Counter

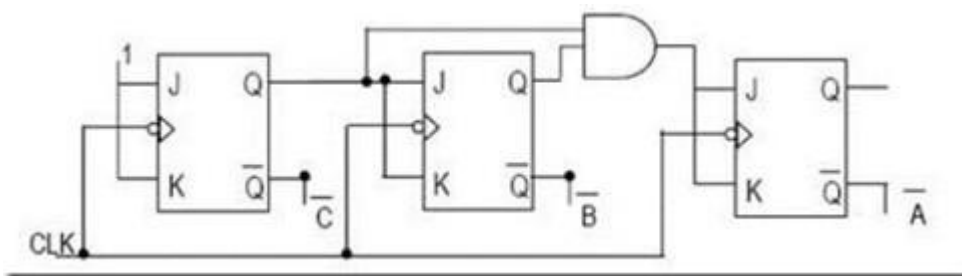


2. Counter Down

Counter down adalah rangkaian yang berfungsi sebagai penghitung mundur. Counter down dibagi menjadi 2 :

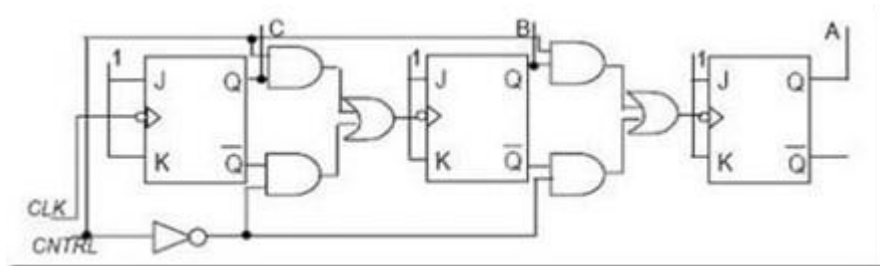
- a. Synchronous Down Counter adalah rangkaian penghitung mundur yang pemicuannya terjadi serempak (dipicu oleh satu sumber clock).

Gambar 12.3 Synchronous Down Counter



- b. Asynchronous Down Counter adalah rangkaian perhitungan maju yang pemicuannya oleh sumber clock lain.

Gambar 12.4 Asynchronous Down Counter



3. Counter Up dan Down

Counter up dan down adalah rangkaian yang menggabungkan perhitungan maju dan mundur.

12.2.2 Register

Register adalah rangkaian logika yang digunakan untuk menyimpan data per bit. Register tersusun dari satu atau beberapa flip-flop. Register digunakan untuk menyimpan grup bit sementara, register juga digunakan untuk pergerakan data pada operasi komputer.

Shift Register

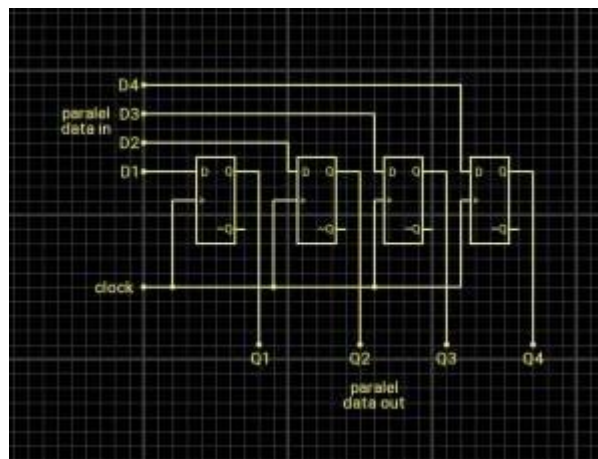
Salah satu simulasi dari register adalah shift register atau register penggeser. Rangkaian ini digunakan sebagai memori sementara dan untuk pergeseran data ke kiri atau ke kanan. Register geser dapat juga digunakan untuk mengubah format data seri ke paralel atau dari paralel ke seri.

Macam-macam tipe Shift Register :

1. Register Parallel In Parallel Out (PIPO)

Shift Register Parallel In Parallel Out (PIPO) merupakan register geser yang input dan outputnya adalah paralel. Contoh: IC TTL 74LS174.

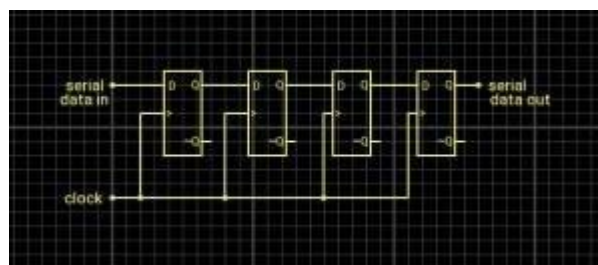
Gambar 12.5 Register Parallel In Parallel Out (PIPO)



2. Register Serial In Serial Out (SISO)

Shift register Serial In Serial Out (SISO) merupakan register geser yang menggunakan input dan output seri, SISO memiliki 2 pergeseran shift right dan shift left. Dikatakan shift right ketika data bit awalnya berupa LSB dan data bit terakhirnya MSB, dan sebaliknya untuk shift left. Contoh: IC TTL 74LS91.

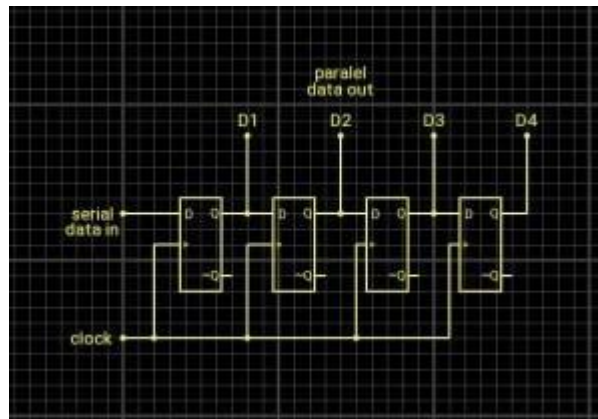
Gambar 12.6 Register Serial In Serial Out (SISO)



3. Register Serial In Parallel Out (SIPO)

Shift Register Serial In Parallel Out (SIPO) merupakan register geser dengan input seri dan output paralel. Contoh: IC TTL 74LS164.

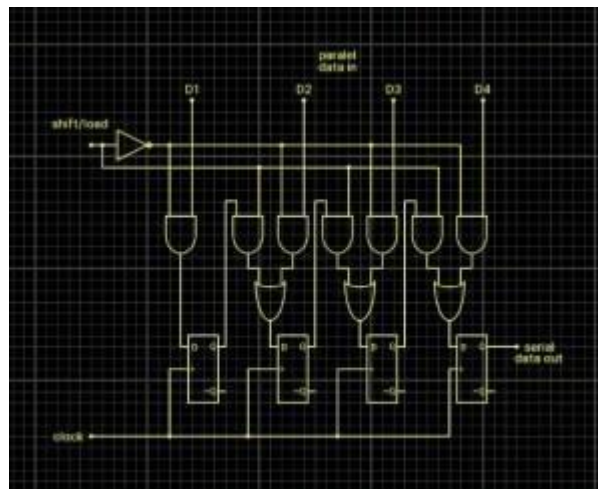
Gambar 12.7 Register Serial In Parallel Out (SIPO)



4. Register Parallel In Serial Out (PISO)

Shift register Parallel In Serial Out (PISO) merupakan register geser dengan input parallel dan output seri. Contoh: IC TTL 74LS165.

Gambar 12.8 Register Parallel In Serial Out (PISO)



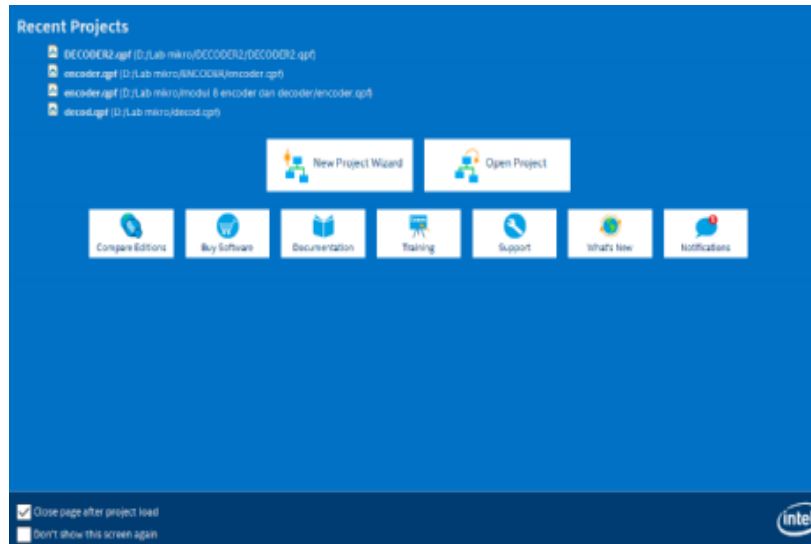
12.3 Langkah Praktikum Modul 12

12.3.1 Alat dan Bahan

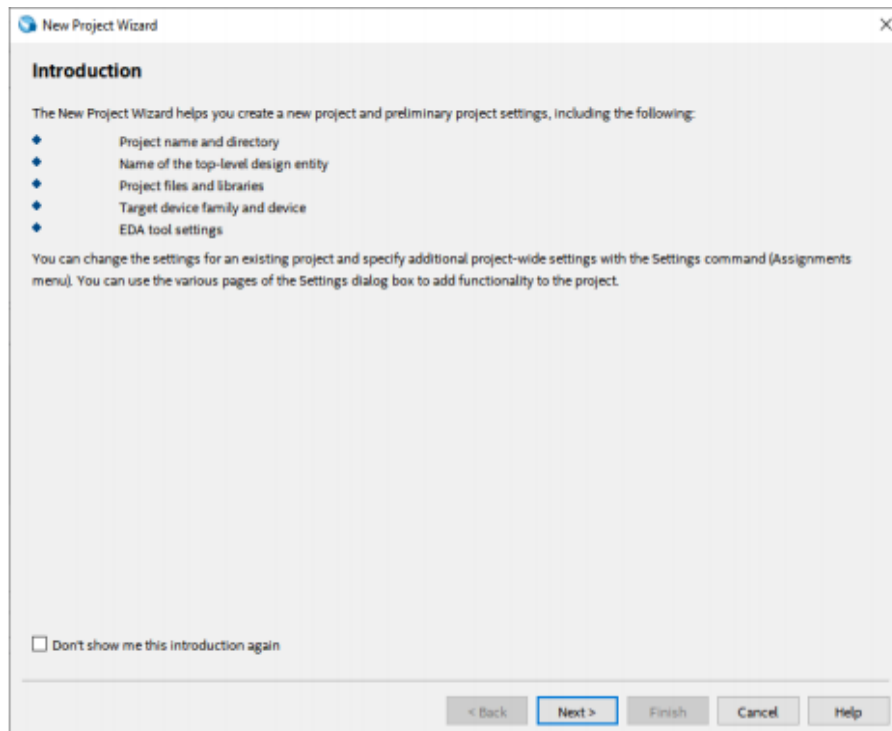
1. Laptop yang sudah terinstal Quartus Prime Lite
2. Mouse

12.3.2 Langkah Praktikum

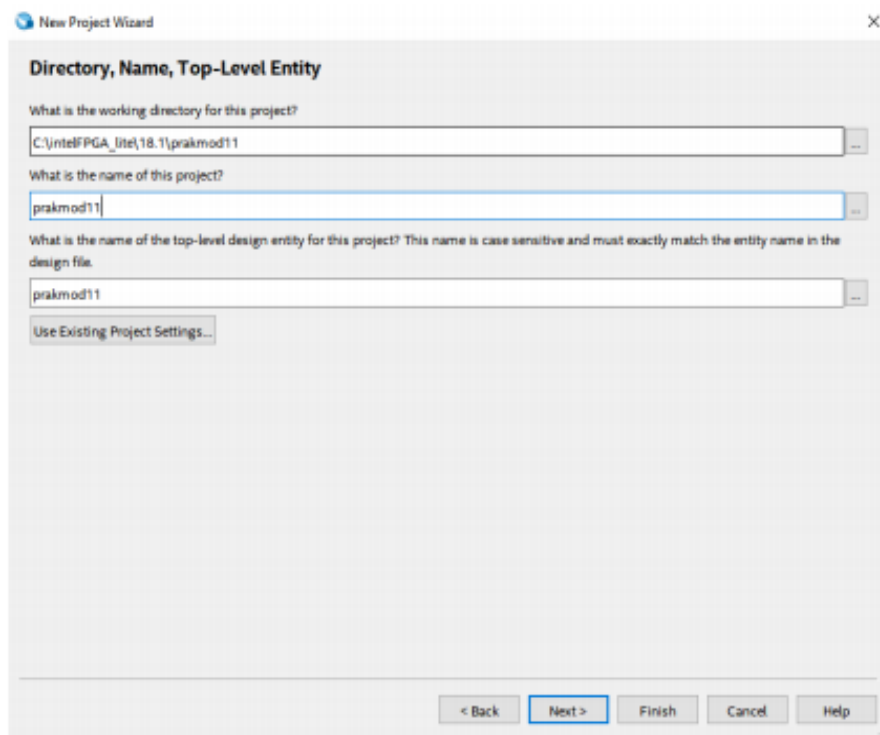
1. Buka software Quartus lite di laptop dan klik **New Project Wizard**



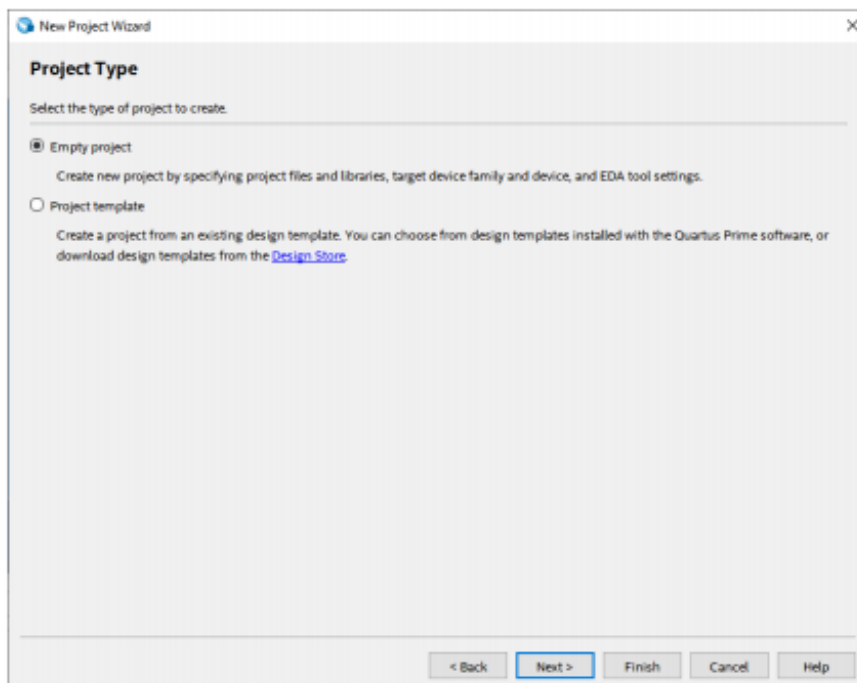
2. Klik **Next**



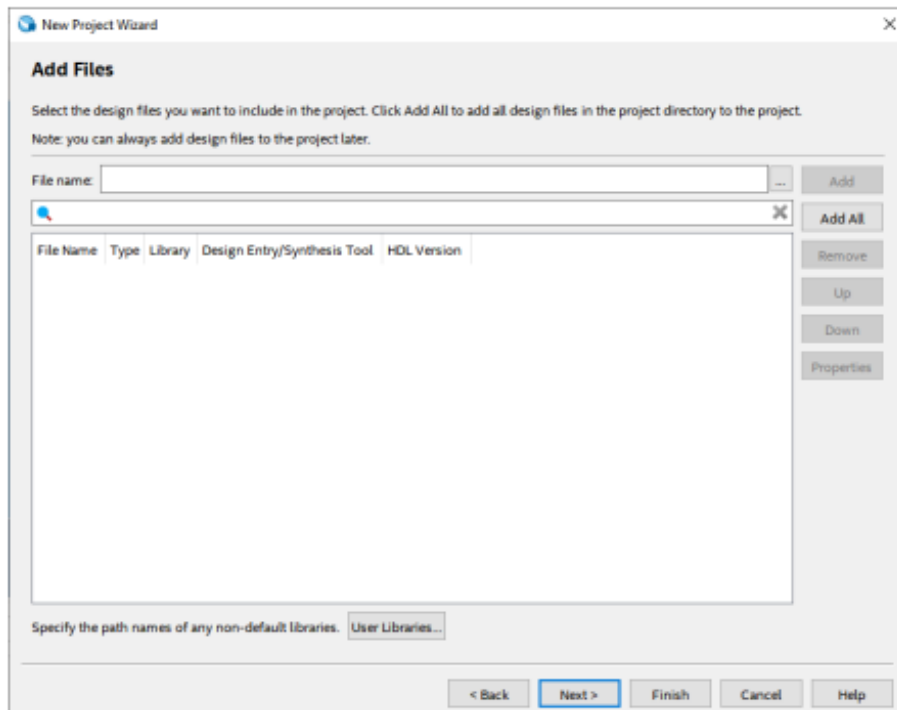
3. Kemudian tentukan Directory Project dan Nama Project, lalu klik **Next**



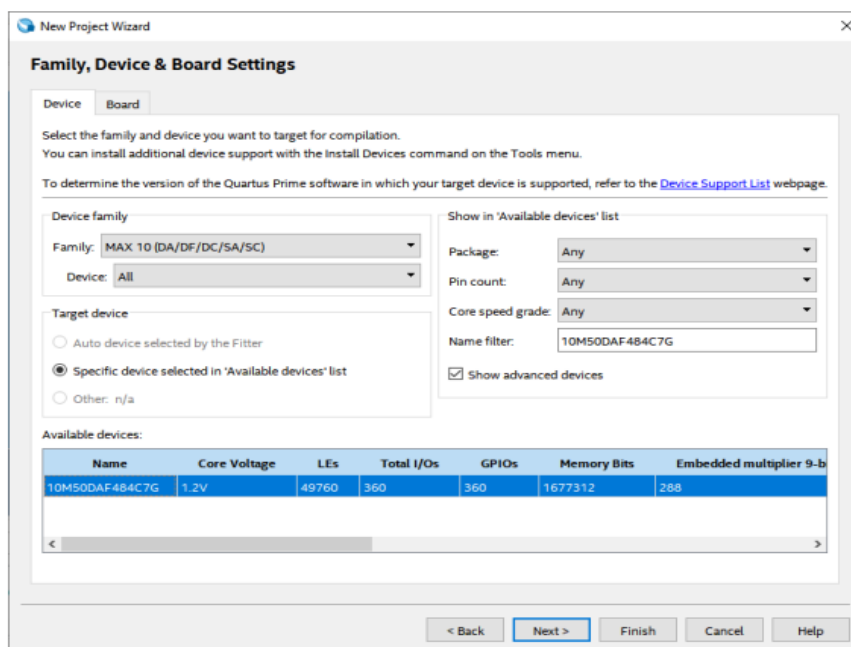
4. Pilih **Empty Project**, kemudian klik **Next** lagi



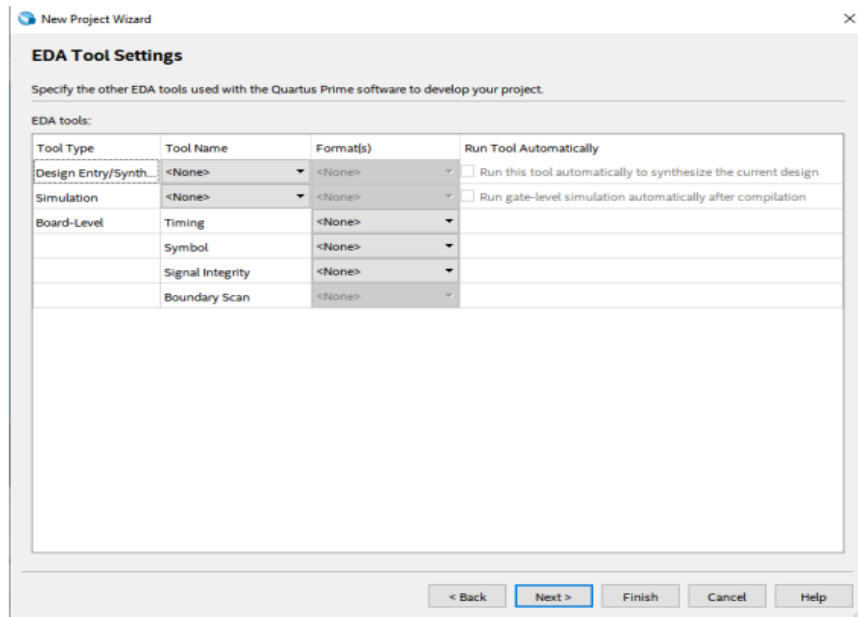
- Setelah itu, klik **Next** lagi



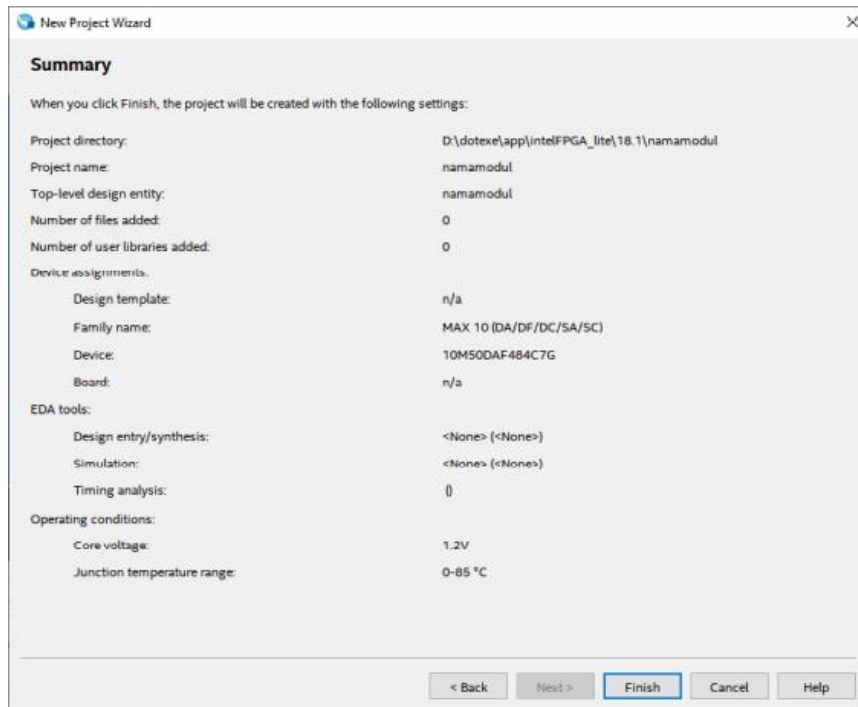
- Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next**



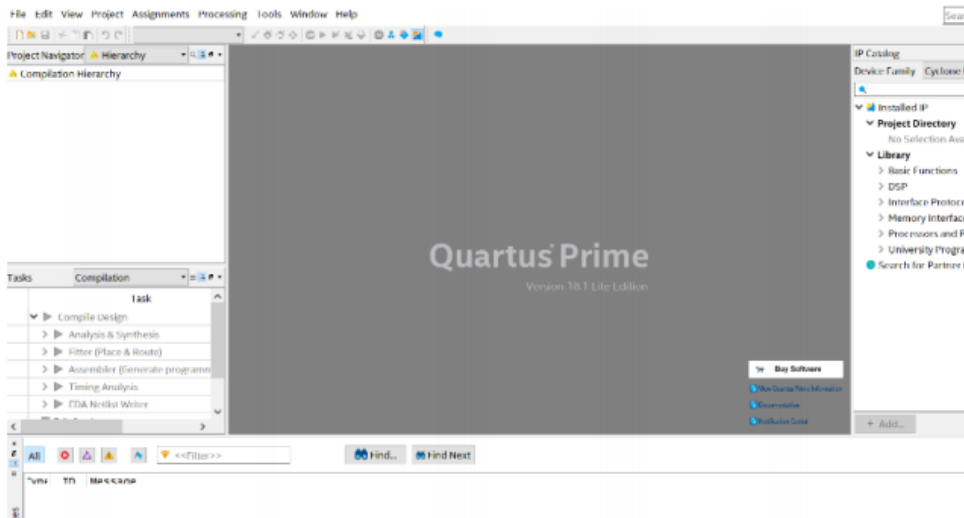
7. Kemudian klik **Next**



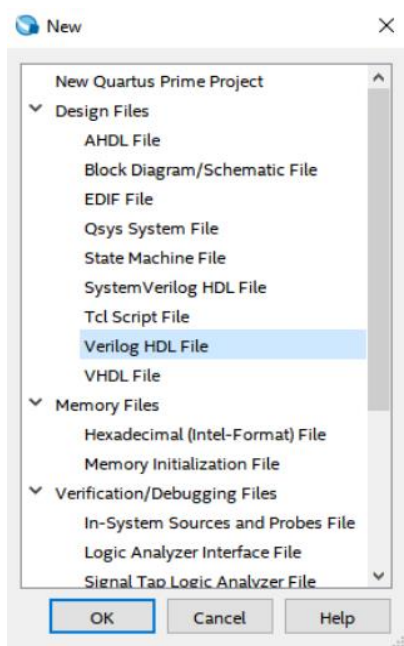
8. Lalu klik **Finish**



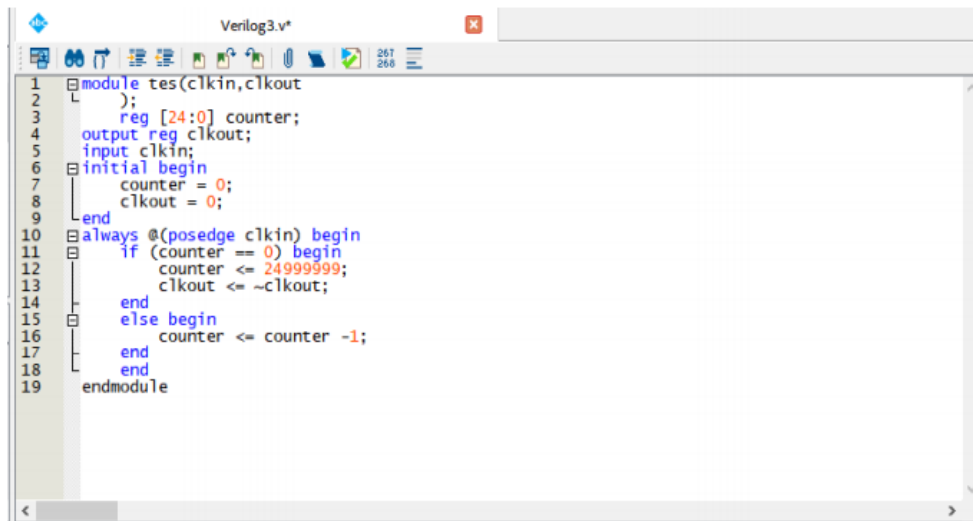
9. Setelah itu akan muncul tampilan awal dari Project Quartus Prime seperti gambar dibawah ini



10. Kemudian buat file baru dengan cara **klik File → New**, lalu pilih **Verilog HDL File**, kemudian klik **OK**

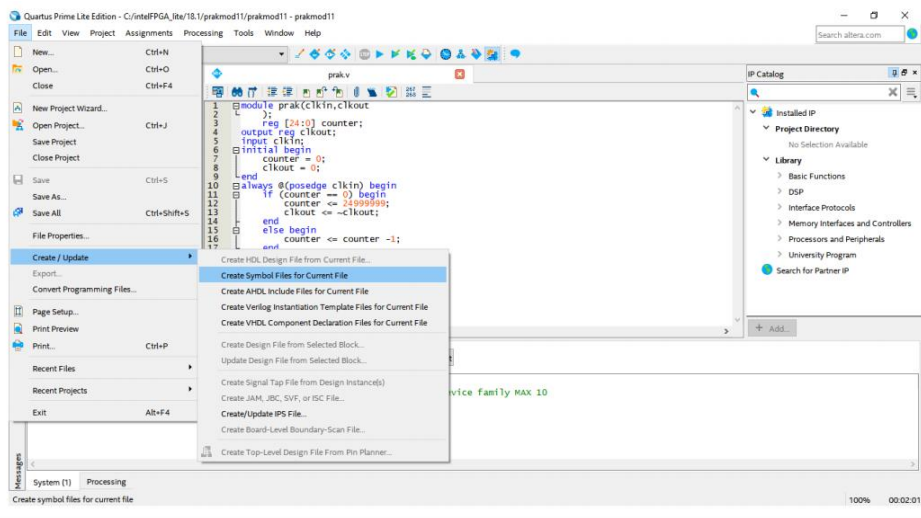


11. Setelah muncul workspace, tuliskan code Verilog seperti gambar dibawah, kemudian **Save** (Pastikan Nama module dan nama file Verilog berbeda dengan nama project).

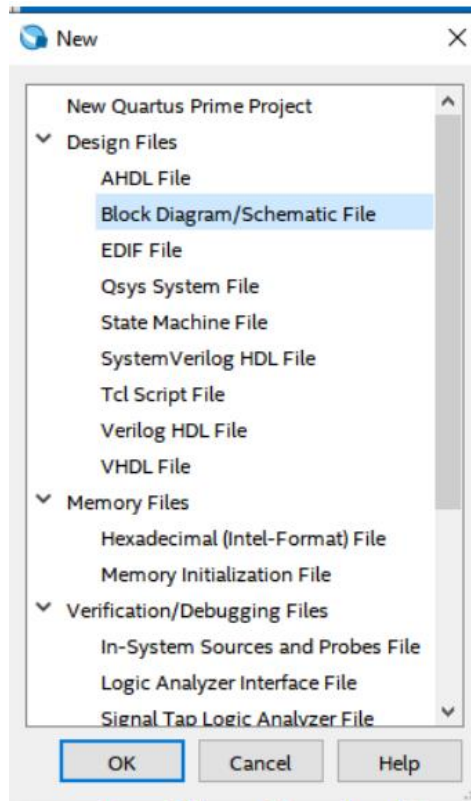


```
1 module tes(clkin,clkout
2 );
3     reg [24:0] counter;
4     output reg clkout;
5     input clkin;
6     initial begin
7         counter = 0;
8         clkout = 0;
9     end
10    always @(posedge clkin) begin
11        if (counter == 0) begin
12            counter <= 24999999;
13            clkout <= ~clkout;
14        end
15        else begin
16            counter <= counter -1;
17        end
18    end
19 endmodule
```

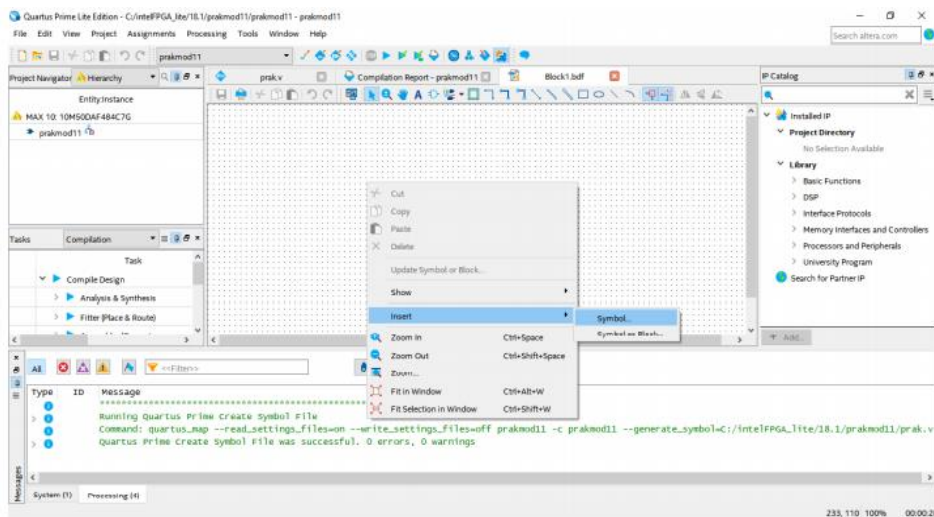
12. Kemudian **klik File**, kemudian **Create/Update** pilih **Create Symbol Files For Current File**.



13. Buat file baru, dengan **klik File** → **New**, pilih **Block Diagram/Schematic File**, lalu **klik OK**.

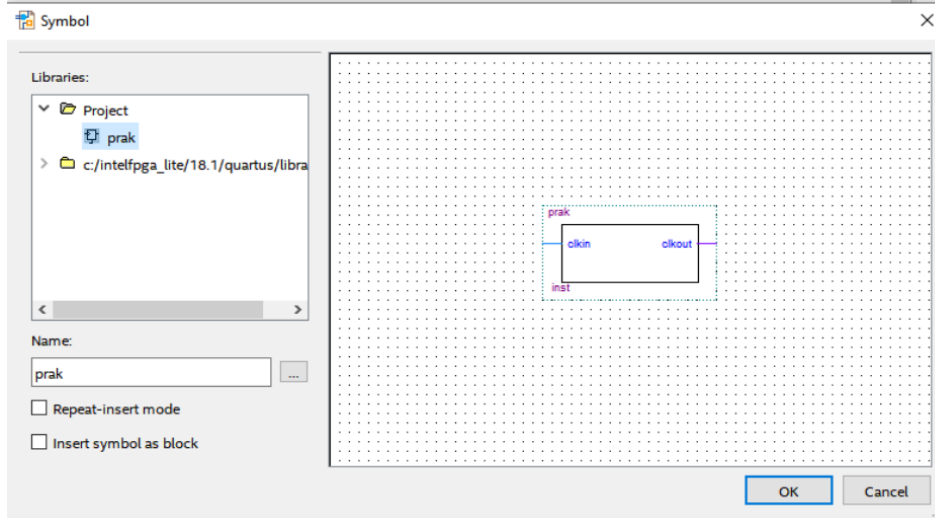


14. Setelah muncul workspace, masukkan symbol yang telah dibuat. Klik kanan pada workspace, kemudian klik **Insert** → **Symbol**.

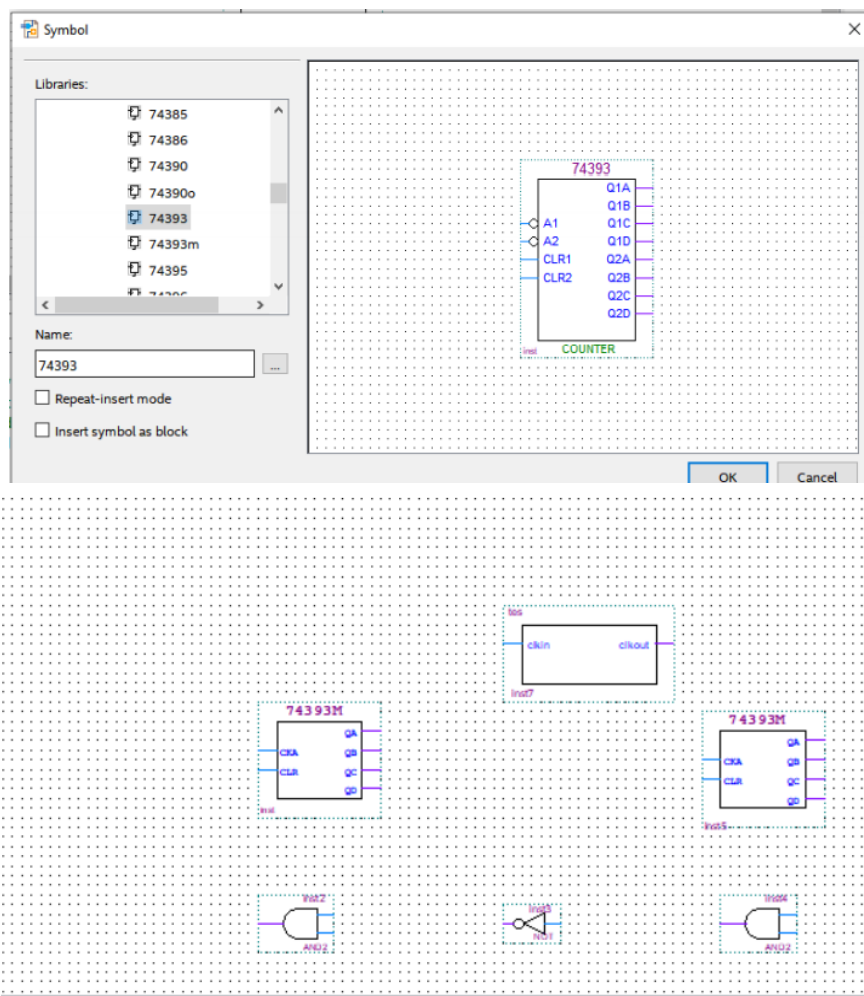


15. Kemudian klik Project, masukkan block yang sudah dibuat lalu klik **OK**.

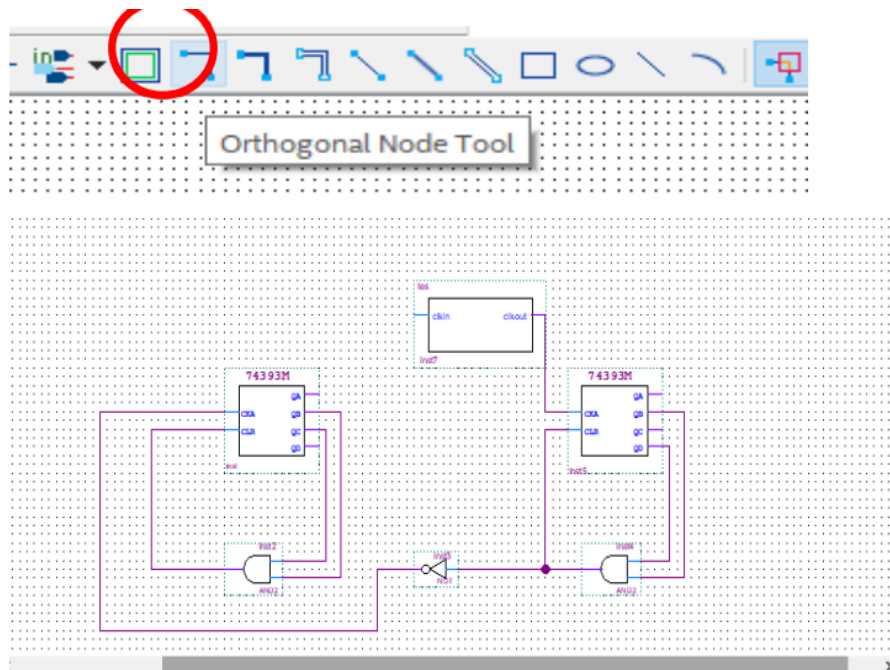
Modul Praktikum



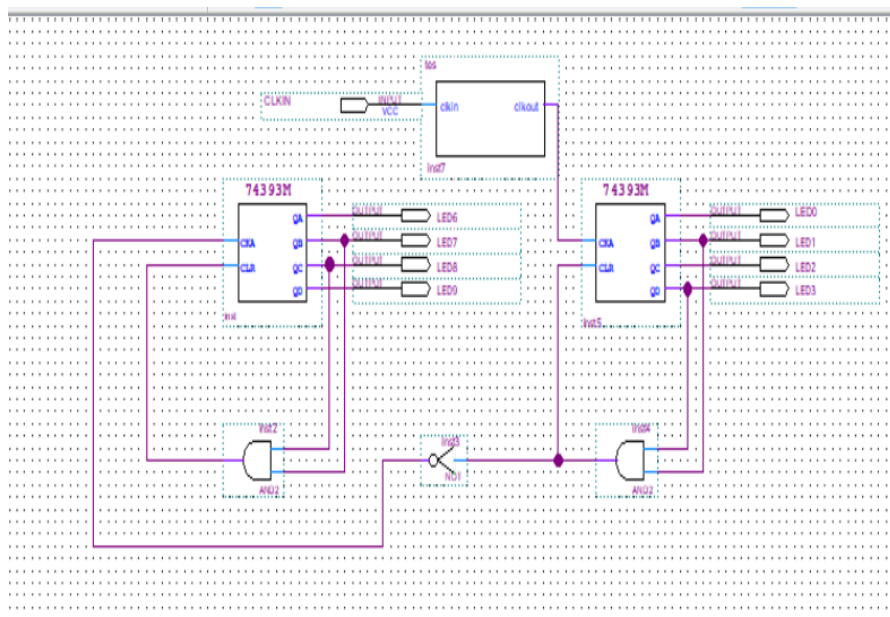
16. Masukkan komponen-komponen lain, dengan langkah yang sama seperti langkah praktikum, cari nama komponen dalam pencarian komponen.



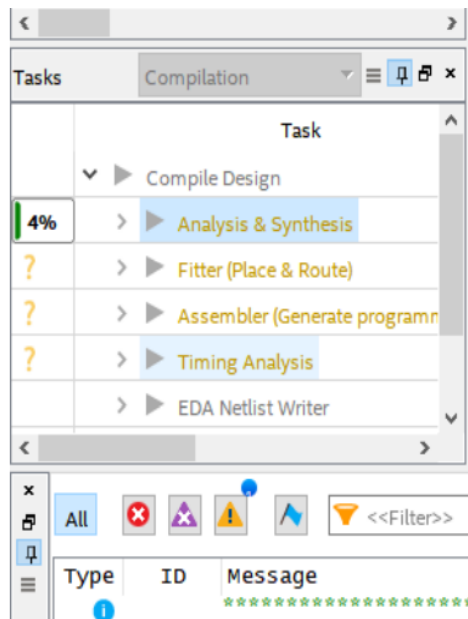
17. Kemudian sambungkan rangkaian dengan **Orthogonal Node Tool**.



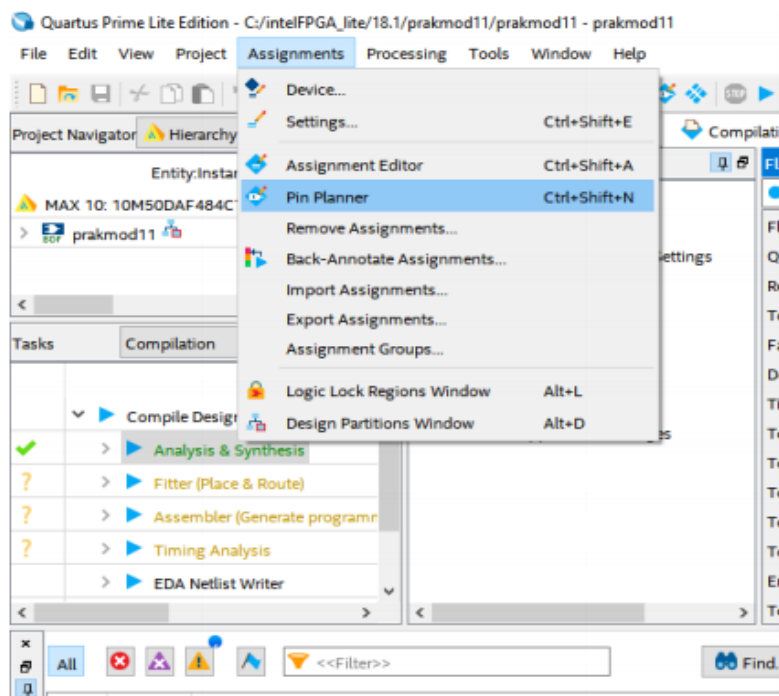
18. Berikan pin input dan output pada rangkaian



19. Klik dua kali pada **Analysis & Synthesis**



20. Klik Assignment → Pin Planner

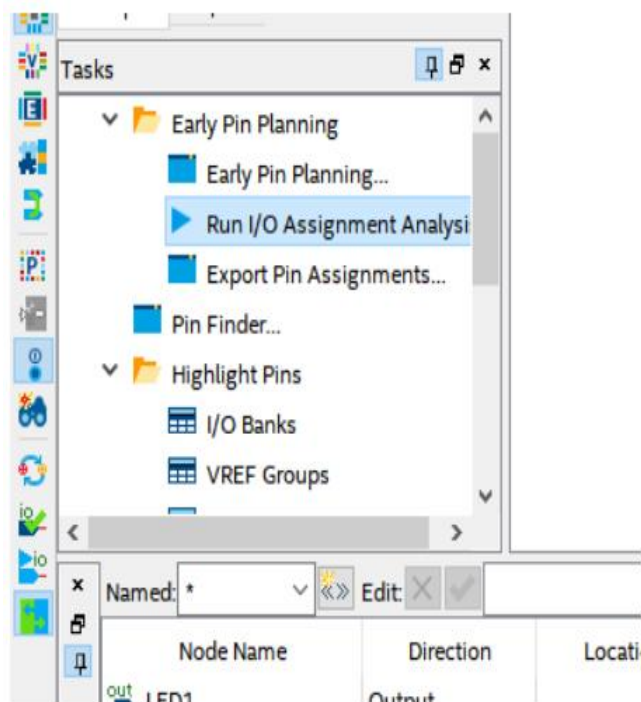


21. Masukkan pin sesuai gambar dibawah

Modul Praktikum

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reser
In CLKIN	Input	PIN_P11	3	B3_NO	2.5 V (default)	
out LED0	Output	PIN_A8	7	B7_NO	2.5 V (default)	
out LED1	Output	PIN_A9	7	B7_NO	2.5 V (default)	
out LED2	Output	PIN_A10	7	B7_NO	2.5 V (default)	
out LED3	Output	PIN_B10	7	B7_NO	2.5 V (default)	
out LED6	Output	PIN_E14	7	B7_NO	2.5 V (default)	
out LED7	Output	PIN_D14	7	B7_NO	2.5 V (default)	
out LED8	Output	PIN_A11	7	B7_NO	2.5 V (default)	
out LED9	Output	PIN_B11	7	B7_NO	2.5 V (default)	
<<new node>>						

22. Klik **Run I/O Assignment**



23. Lalu klik dua kali **Compile Design**