

### MODUL 13 IMPLEMENTASI SOFTCORE PROCESSOR NIOS II (IPCORE)

#### 14.1 Tujuan Praktikum Modul 13 :

Setelah mempraktekkan topik ini, praktikan diharapkan dapat:

1. Dapat mengetahui konsep dasar dari Prosesor NIOS II dan eclipse
2. Dapat mengimplementasikan NIOS II menggunakan DE10-Lite FPGA Board

#### 14.2 Dasar Teori Praktikum Modul 13:

##### 14.2.1 NIOS II

*Nios II* adalah arsitektur prosesor 32 bit tertanam yang dirancang khusus untuk keluarga Altera dari rangkaian terintegrasi field-programmable gate array (FPGA). Nios II menggabungkan banyak peningkatan dibandingkan arsitektur Nios asli, membuatnya lebih cocok untuk berbagai aplikasi komputasi tertanam, dari pemrosesan sinyal digital (DSP) hingga kontrol sistem.

##### 14.2.2 Arsitektur Nios II

Arsitektur Nios II adalah arsitektur *soft-core* RISC yang diimplementasikan seluruhnya dalam logika yang dapat diprogram dan blok memori Altera FPGAs. Sifat *soft-core* dari prosesor Nios II memungkinkan perancang sistem menentukan dan menghasilkan inti Nios II kustom, yang dirancang untuk persyaratan aplikasi spesifiknya.

##### 14.2.3 Eclipse

Eclipse adalah *integrated development environment* (IDE) yang digunakan dalam pemrograman komputer, dan merupakan IDE Java yang paling banyak digunakan, berisi ruang kerja dasar dan sistem plug-in yang dapat dikembangkan untuk menyesuaikan lingkungan. Eclipse ditulis sebagian besar di Java dan penggunaan utamanya adalah untuk mengembangkan aplikasi Java, tetapi juga dapat digunakan untuk mengembangkan aplikasi dalam bahasa pemrograman lain melalui *plug-in*, termasuk Ada, ABAP, C, C ++, C #, Clojure, COBOL, D , Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (termasuk kerangka kerja Rubyon Rails), Rust, Scala, dan Skema. Ini juga dapat digunakan untuk mengembangkan dokumen dengan LaTeX (melalui plug-in TeXlipse) dan paket untuk perangkat lunak Mathematica. Lingkungan pengembangan termasuk alat pengembangan Java Eclipse (JDT) untuk Java dan Scala, Eclipse CDT untuk C /C ++, dan Eclipse PDT untuk PHP.

##### 14.2.1 Arsitektur Eclipse

Eclipse menggunakan *plug-in* untuk menyediakan semua fungsionalitas di dalam dan di atas sistem *run-time*. Sistem *run-time* didasarkan pada Equinox, sebuah implementasi dari spesifikasi kerangka kerja OSGi. Selain memungkinkan *Platform* Eclipse diperluas menggunakan bahasa pemrograman lain, seperti C dan Python, kerangka *plug-in* memungkinkan *Platform* Eclipse untuk bekerja dengan bahasa penyusunan huruf seperti LaTeX dan aplikasi jaringan seperti telnet dan sistem manajemen basis data . Arsitektur *plug-in* mendukung penulisan ekstensi apa pun yang diinginkan ke lingkungan, seperti untuk manajemen konfigurasi. Dukungan Java dan CVS disediakan di Eclipse SDK, dengan dukungan untuk sistem kontrol versi lain yang disediakan oleh *plug-in* pihak ketiga. **Eclipse** mengimplementasikan elemen kontrol grafis Java *toolkit* yang disebut *Standard Widget Toolkit* (SWT), sedangkan sebagian besar aplikasi Java menggunakan *Java Window Abstract Toolkit* (AWT) atau Swing standar Java. Antarmuka pengguna Eclipse juga menggunakan lapisan antarmuka

## Modul Praktikum

pengguna grafis perantara yang disebut JFace, yang menyederhanakan konstruksi aplikasi berdasarkan SWT.

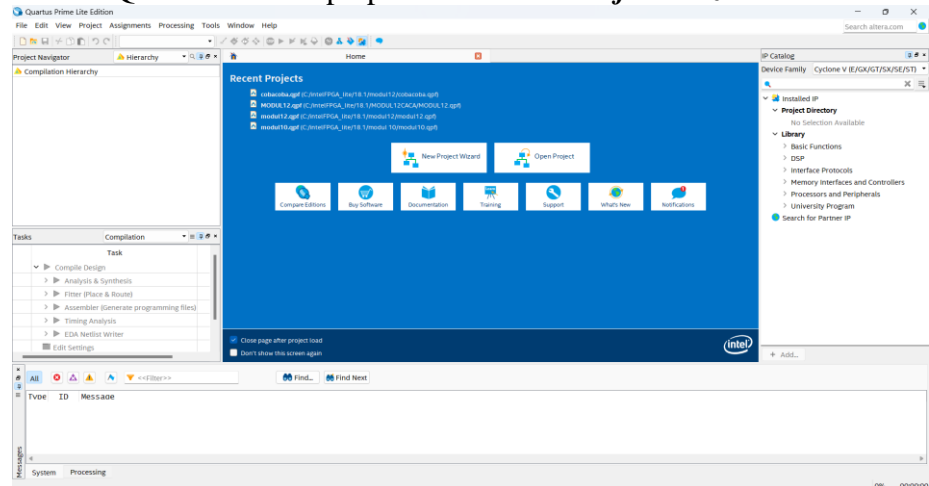
### 14.3 Lembar Kegiatan Praktikum Modul 14

#### 14.3.1 Alat dan Bahan

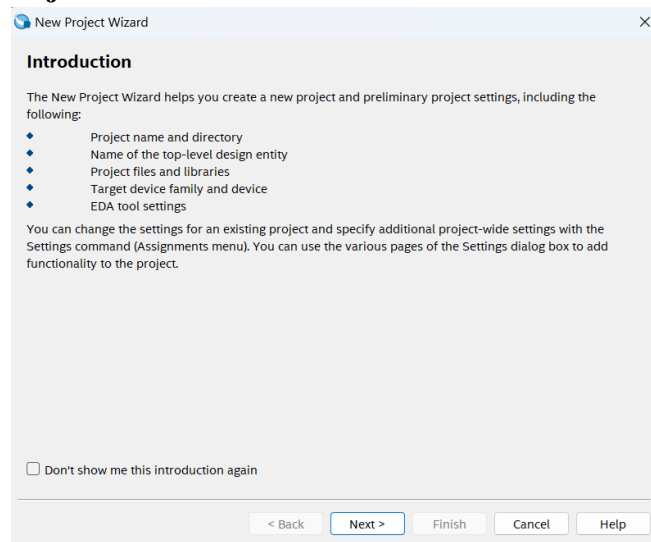
1. Laptop yang sudah terinstal *software* Quartus 18.1 dan NIOS II
2. *terasiC* DE10-Lite FPGA board

#### 14.3.2 Langkah Praktikum Modul 12

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

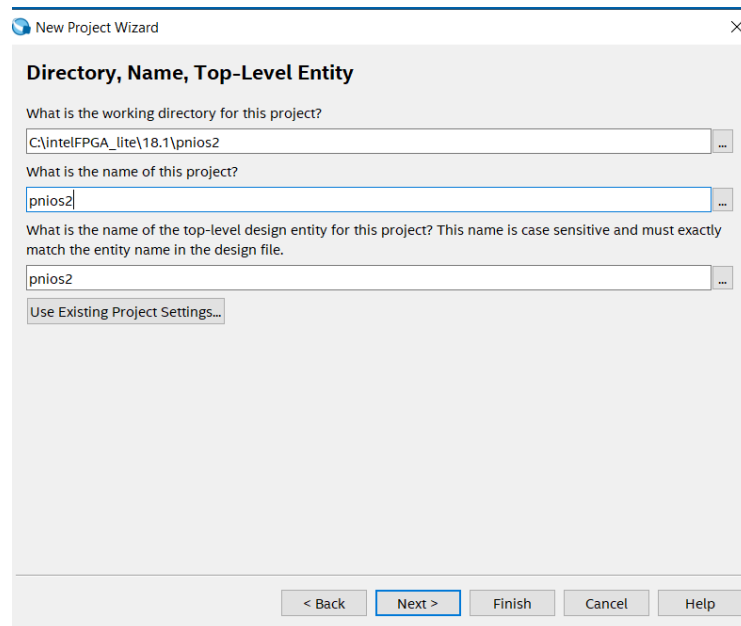


2. Klik *New Project Wizard* > *Next*.

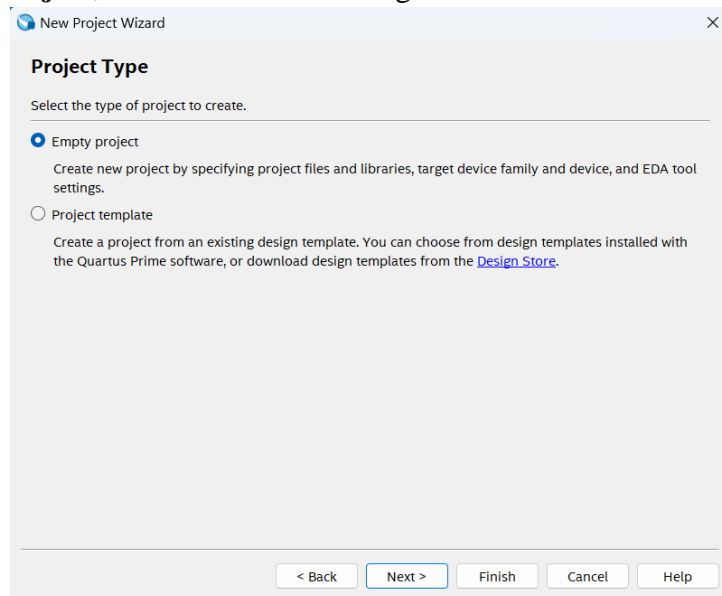


## Modul Praktikum

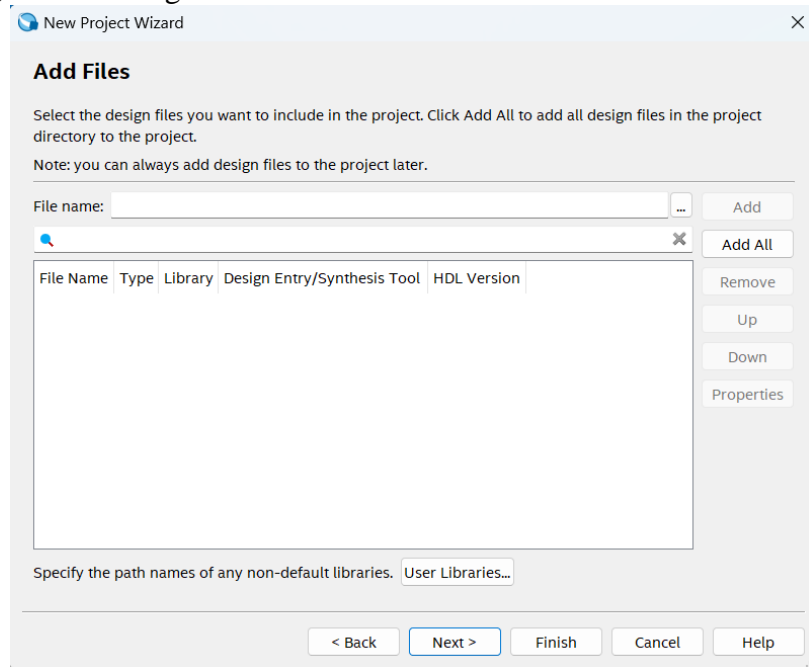
3. Kemudian tentukan buat Directory baru `\pnios2` dan Nama Project **pnios2**, lalu klik **Next**.



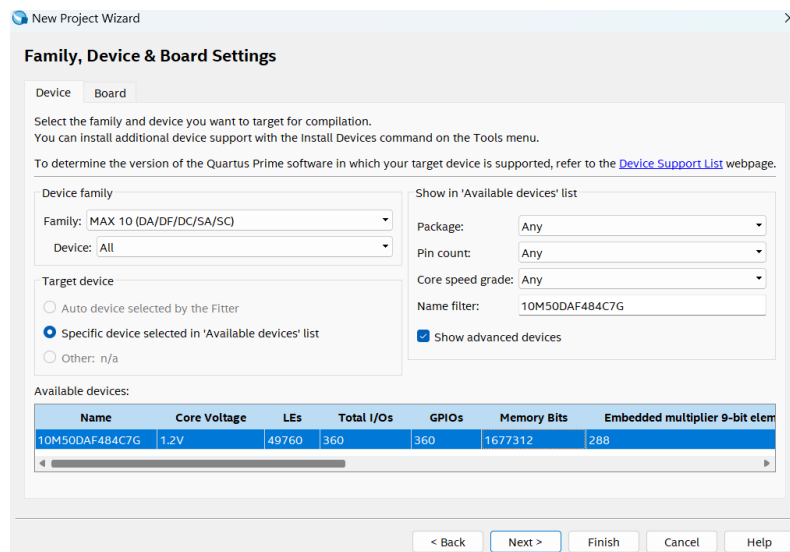
4. Pilih **Empty Project**, kemudian klik **Next** lagi.



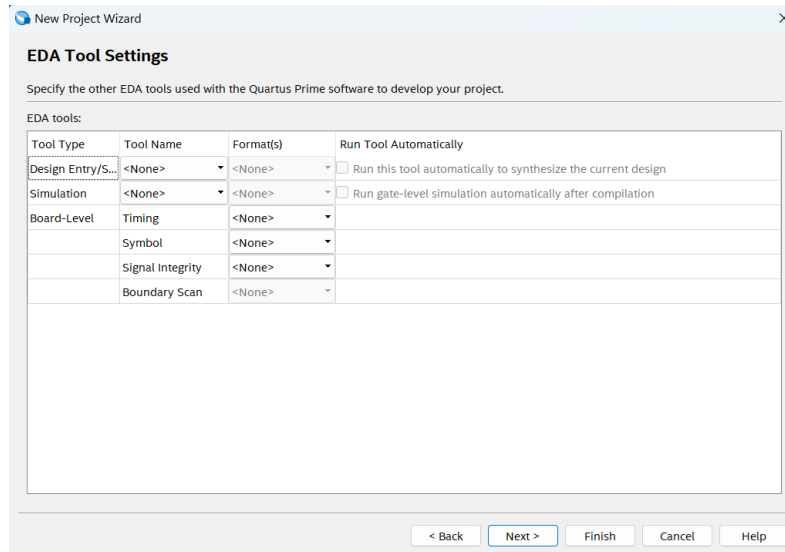
- Setelah itu, klik **Next** lagi.



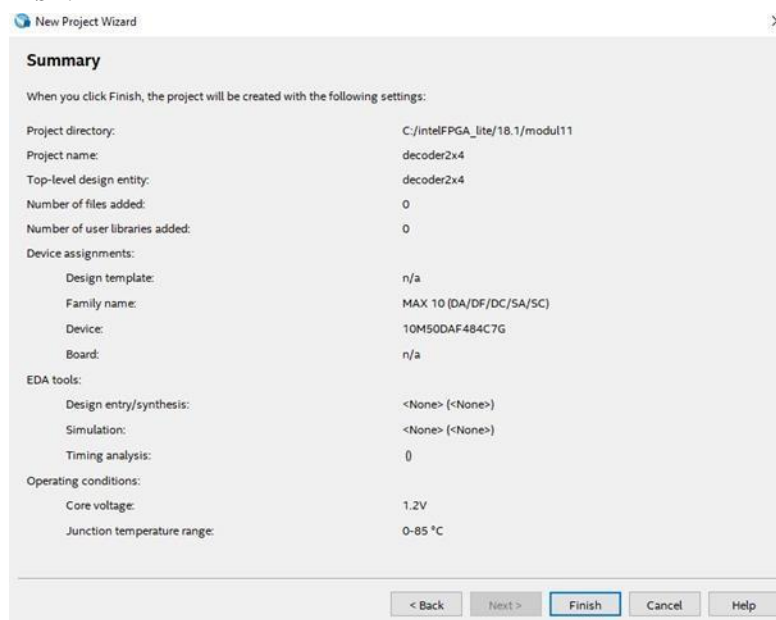
- Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next**.



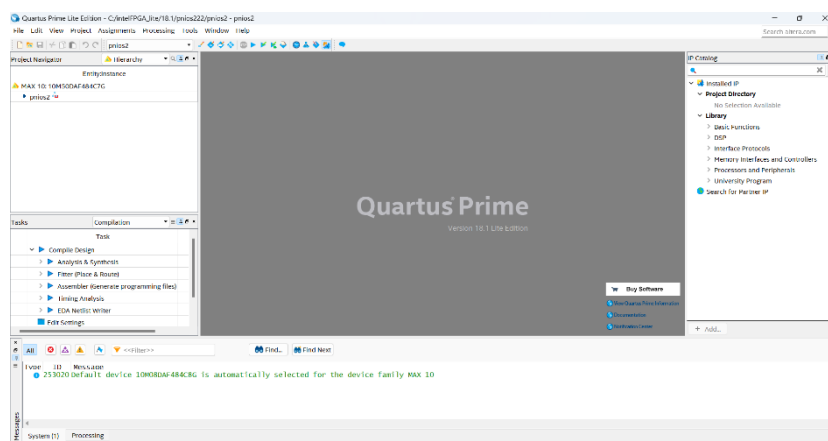
## 7. Kemudian klik **Next**.



## 8. Lalu klik **Finish**.

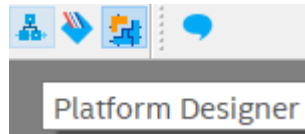


## 9. Setelah itu akan muncul tampilan awal dari Project Quartus Prime seperti gambar di bawah ini.

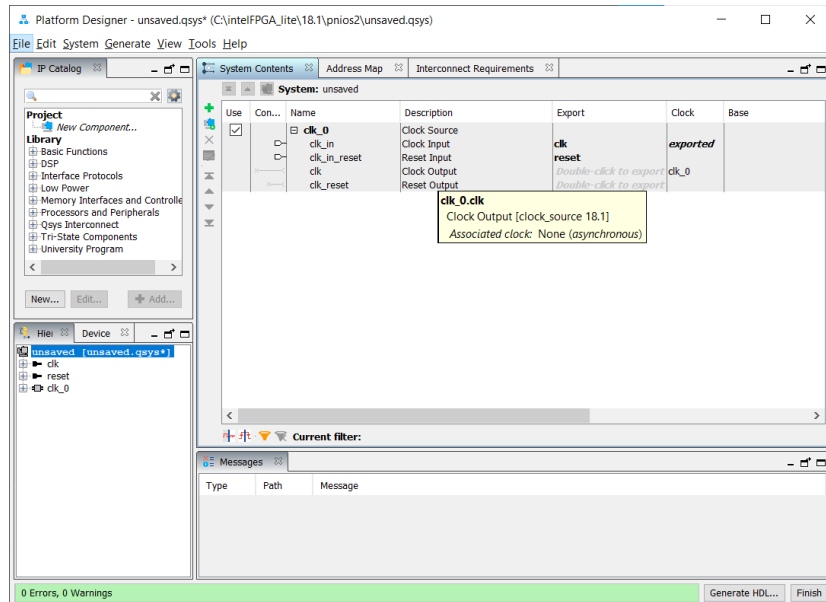


## 10. Kemudian buat file baru dengan cara klik **platform designer (skematik) qsys (doble**

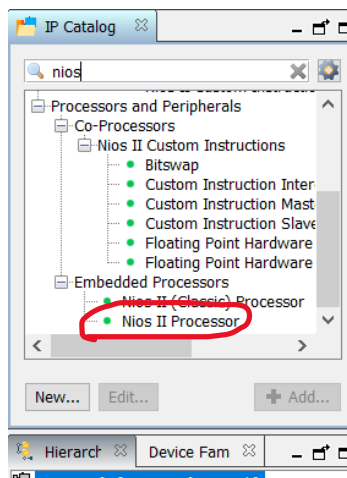
click).



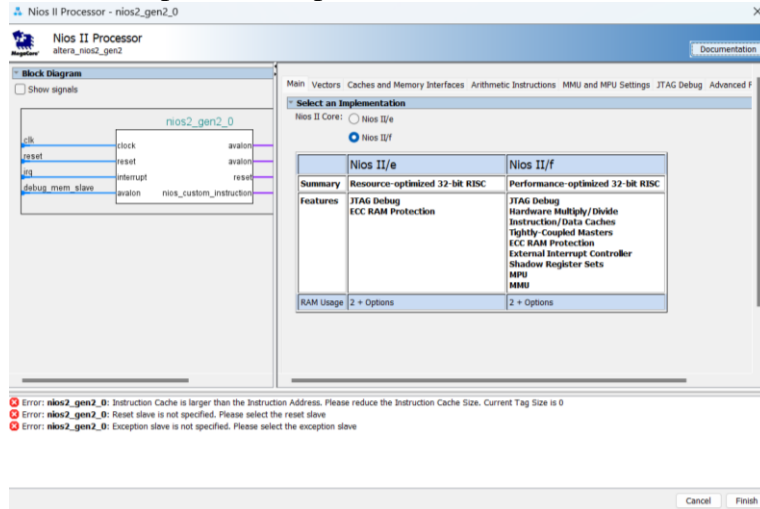
11. Setelah muncul workspace (clock (default)), Masukkan komponen.



12. Tambah komponen nios ii, ketik nios > scrool kebawah > double klik pada niosprosesor. Secara otomatis maka modul akan ditambahkan pada workspace.



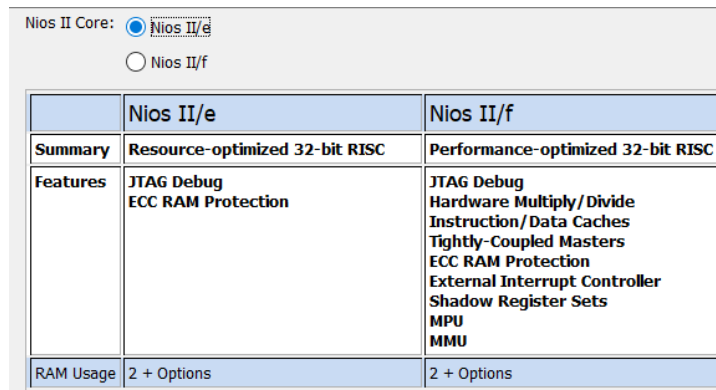
13. Kemudian terbuka workspace nios ii prosesor.



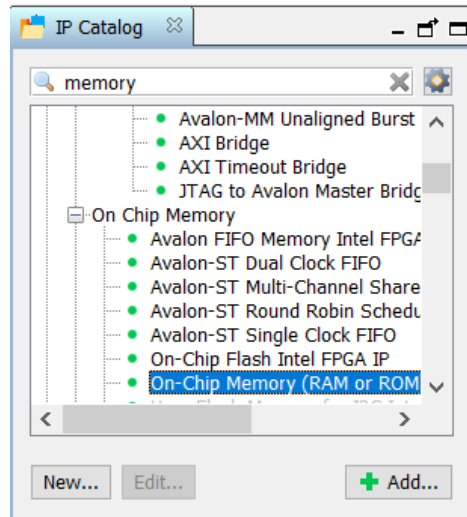
14. Terdapat 2 pilihan nios ii core

1. nios ii / e (ekonomis/s, gratis tidak ada lisensi sama sekali, fitur sedikit) praktikum hanya menggunakan JTAG saja
2. nios ii / f (faster, berbayar, fitur jauh lebih banyak).

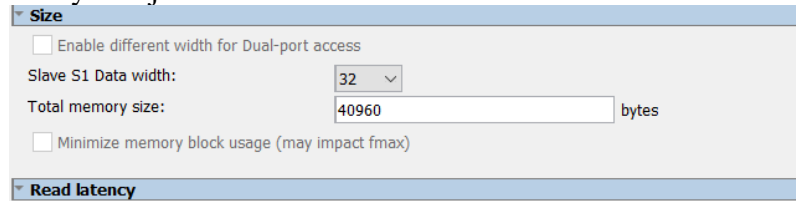
Pilih **ekonomis > finish**



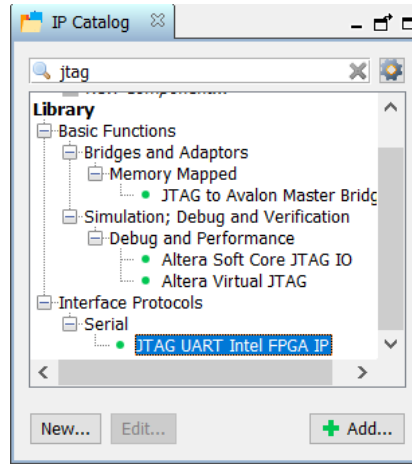
15. Tambah komponen memori, ketik **memory > scroll kebawah > pilih on-chip memory (RAM or ROM) intel FPGA Ip**. Terbuka workspace on-chip memory (RAM or ROM).



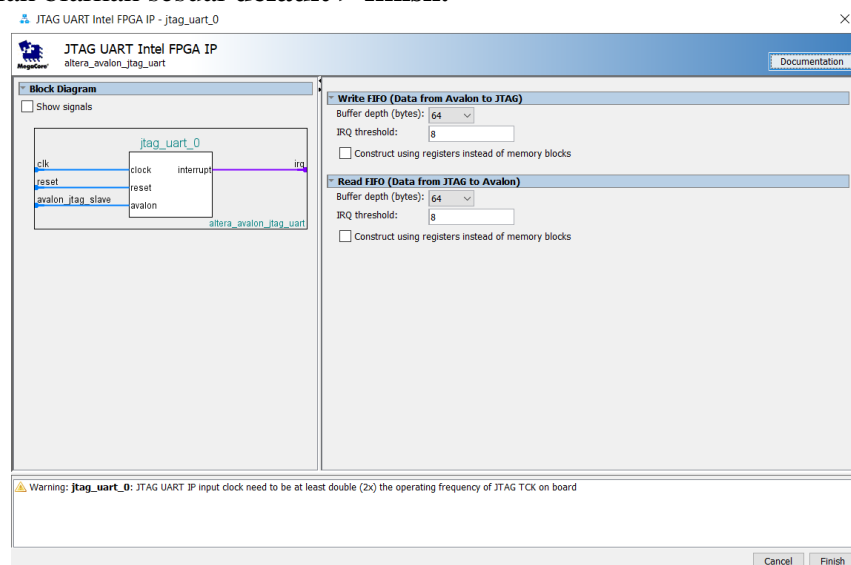
### 16. Tambah memory menjadi **40960** > **Finish**.



### 17. Kemudian tambahkan komponen JTAG UART, ketik **jtag** > **scroll kebawah** > **pilih JTAG UART Intel FPGA IP**. Terbuka workspace JTAG UART.



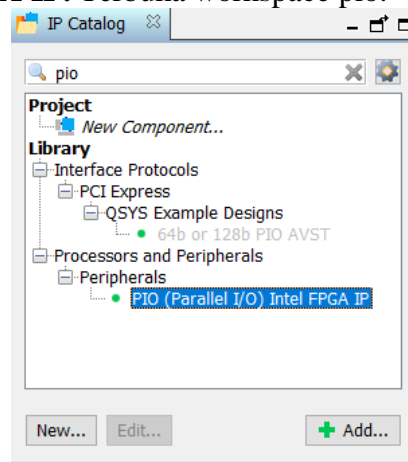
### 18. Kemudian biarkan sesuai **default** > **finish**.



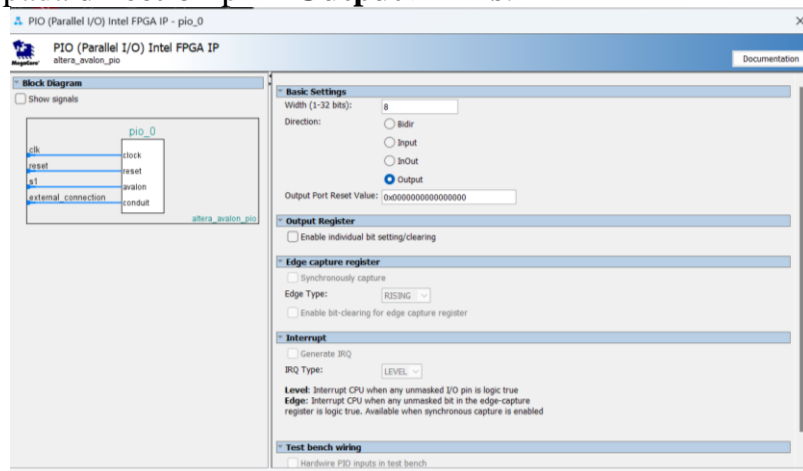


## Modul Praktikum

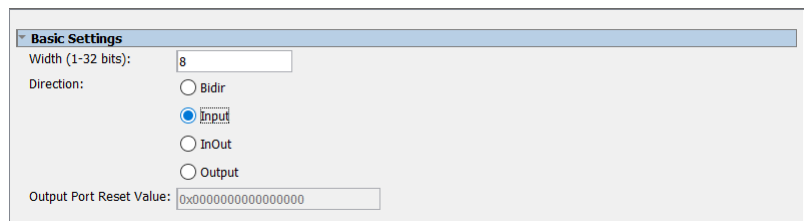
19. Kemudian tambahkan komponen pio (paralel input/output), ketik **pio** > pilih **PIO (Parallel I/O) Intel FPGA IP**. Terbuka workspace pio.



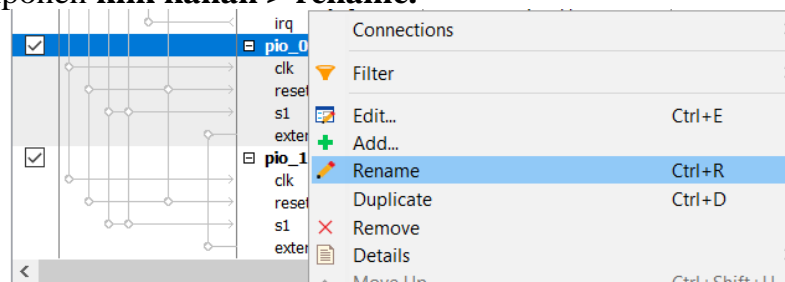
20. Kemudian pada **direction** pilih **Output** > **finis**.



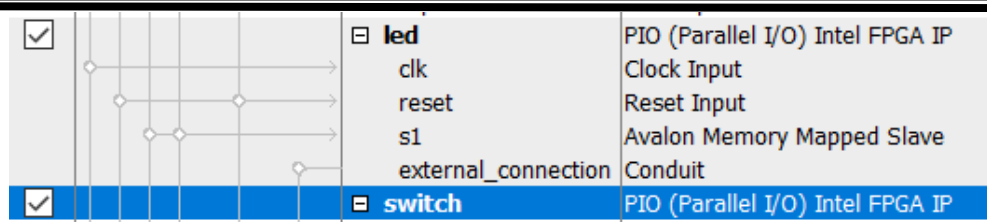
21. Kemudian tambahkan lagi pio (paralel input/output), pada **direction** pilih **Input** > **finis**



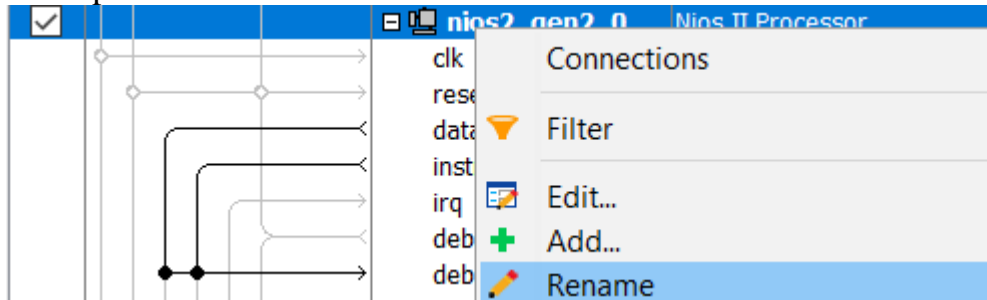
22. Setelah itu, **rename pio 0** menjadi **led** dan **pio 1** menjadi **switch** dengan cara pilih komponen **klik kanan** > **rename**.



## Modul Praktikum



23. Kemudian **rename nios2** menjadi sesuai **nama project pnios2** dengan cara pilih komponen **klik kanan > rename**

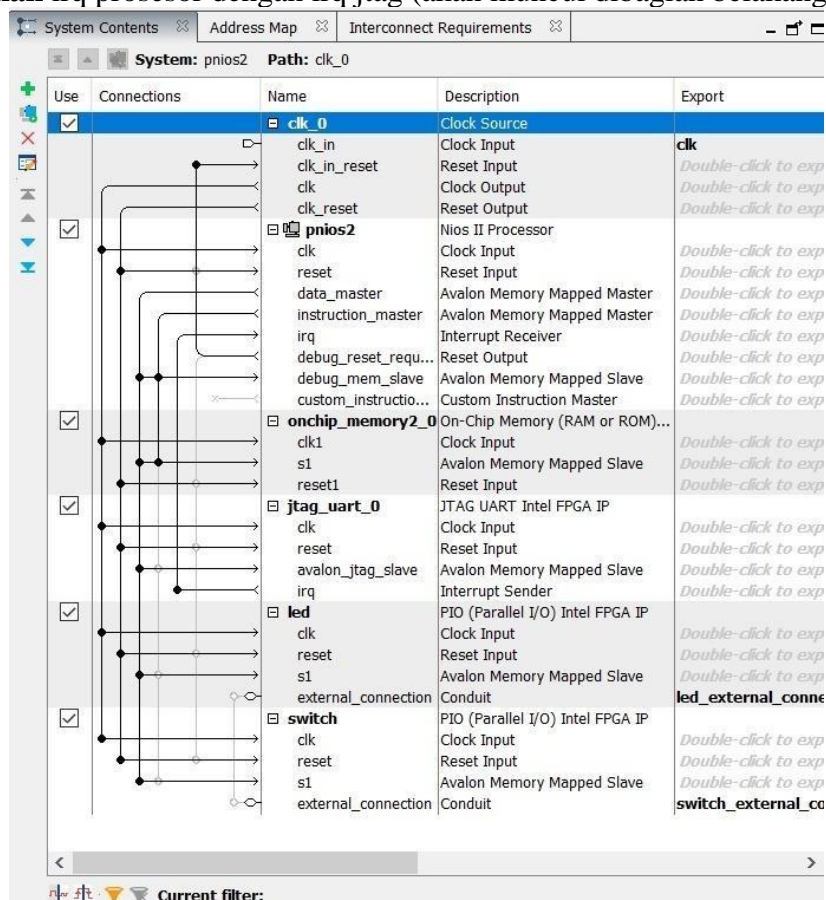


Pastikan semua komponen sudah masuk dan di rename

24. Kemudian wiring pada connections dengan klik setiap percabangan

sambungan atau simbol " | " " "

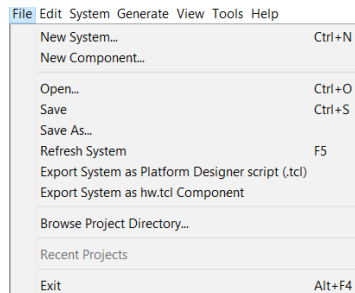
25. **Hubungkan** semua clk yang ada pada nios, onchip, jtag, led, switch
26. **Hubungkan** semua clk\_reset yang ada pada nios, onchip, jtag, led, switch
27. **Hubungkan** clk\_in\_reset ke satu bagian saja yaitu debug\_reset\_request
28. **Hubungkan** data\_master ke semua slave agar semua bisa menerima data
29. **Hubungkan** instruction\_master hanya terhubung pada s1 onchip memory saja
30. **Hubungkan** irq prosesor dengan irq jtag (akan muncul dibagian belakang juga)



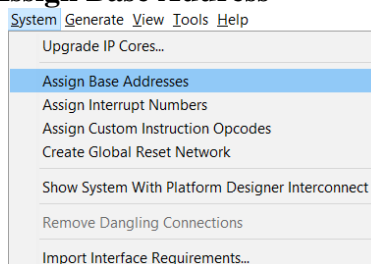
31. Setelah selesai wiring, kemudian pada **led** dan **switch** terdapat **external\_connection**, **double klik** pada **export** maka akan muncul penamaannya > enter

external_connection	Conduit	led external connection	
<input type="checkbox"/> <b>switch</b>	PIO (Parallel I/O) Intel FPGA IP		
clk	Clock Input	<i>Double-click to export</i>	<b>clk_0</b>
reset	Reset Input	<i>Double-click to export</i>	[clk]
s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]
external_connection	Conduit	<i>Double-click to export</i>	

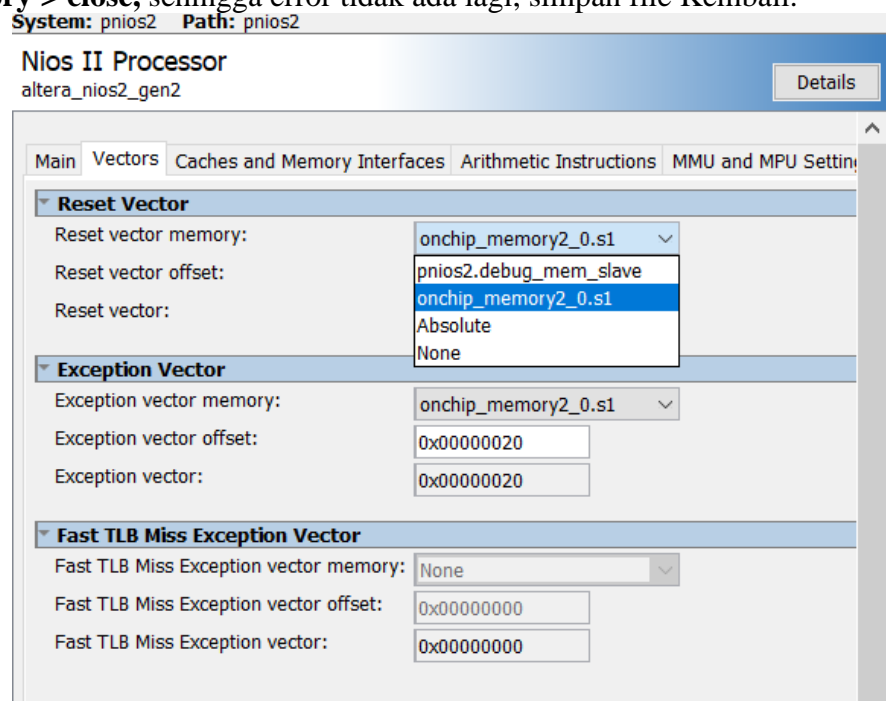
32. Kemudian simpan file, **file > save > ubah nama file pnios 2** (sesuai nama file quartusnya) > **close**



33. Kemudian pilih **system > Assign Base Address**

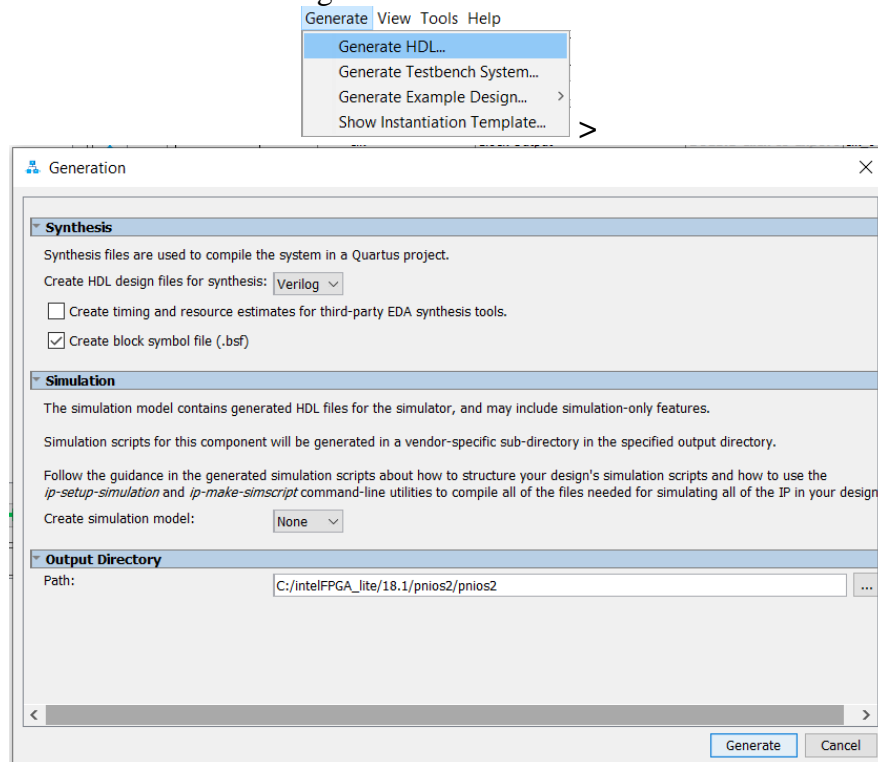


34. Jika masih terdapat error, **double klik** pada **pnios2 > pilih vektor > pada reset vector memory pilih on chip memory > pada exception vector memory pilih pn chip memory > close**, sehingga error tidak ada lagi, simpan file Kembali.

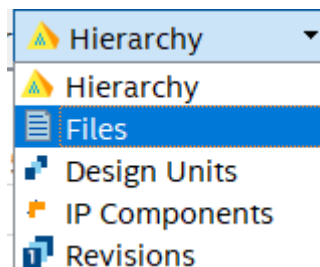


## Modul Praktikum

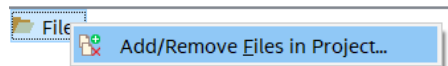
35. Kemudian pilih **generate hardware > generate HDL > create HDL design file for syntseis menjadi verilog > pilih create block symbol file(.bsf) > generate. tunggu hingga selesai > close > close lagi**



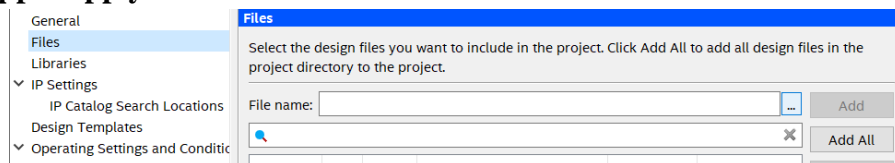
36. Kemudian muncul pemberitahuan berhasil membuat **ip variation in the file > ok**  
37. Kemudian pilih **Hirarcy**



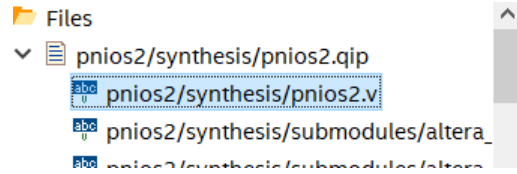
38. Kemudian **pilih files > klik kanan files > add / remove file**



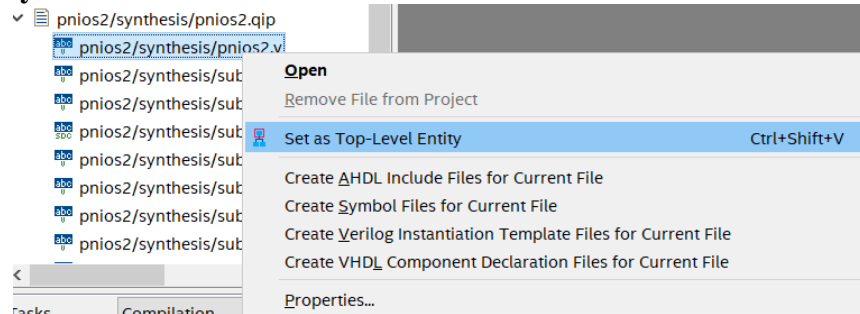
39. Setelah itu pilih **files > pilih browse > pilih folder pnios2 > synthesis > pilih file type.qip > apply > ok**



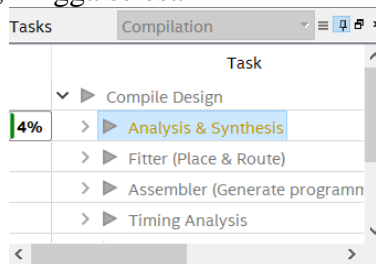
40. Kemudian double klik pada pnios/synthesis/pnios.v untuk melihat verilognya.



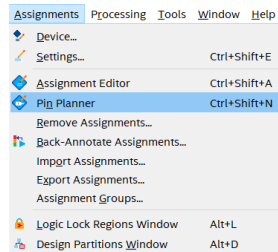
41. Kemudian klik kanan pada pnios/synthesis/pnios.v > set as top level entity > balik ke hirarcy



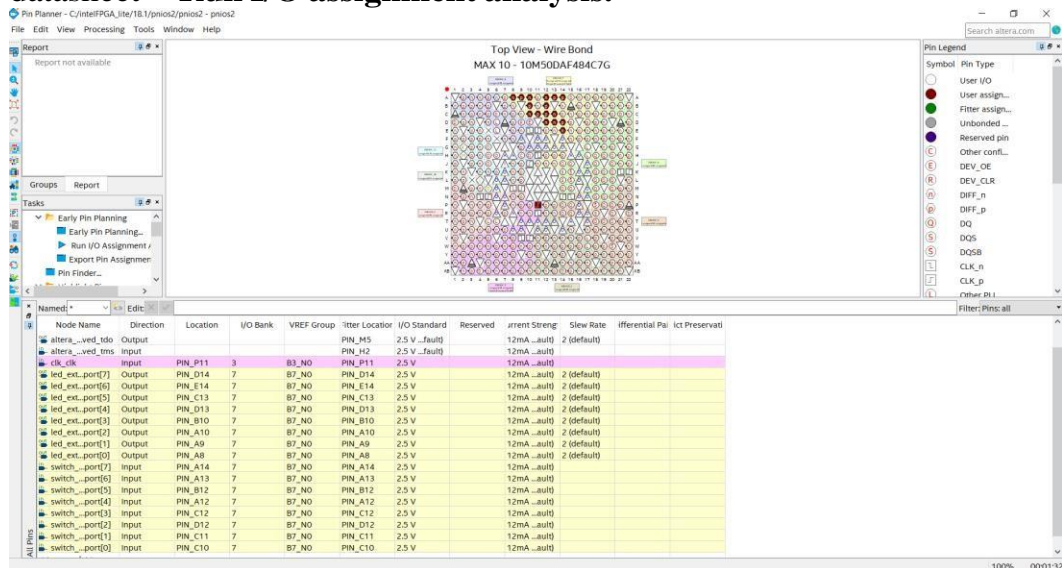
42. Pilih analysis & synthesis, hingga selesai

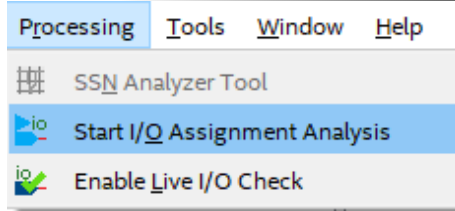


43. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu assignments → pin planner

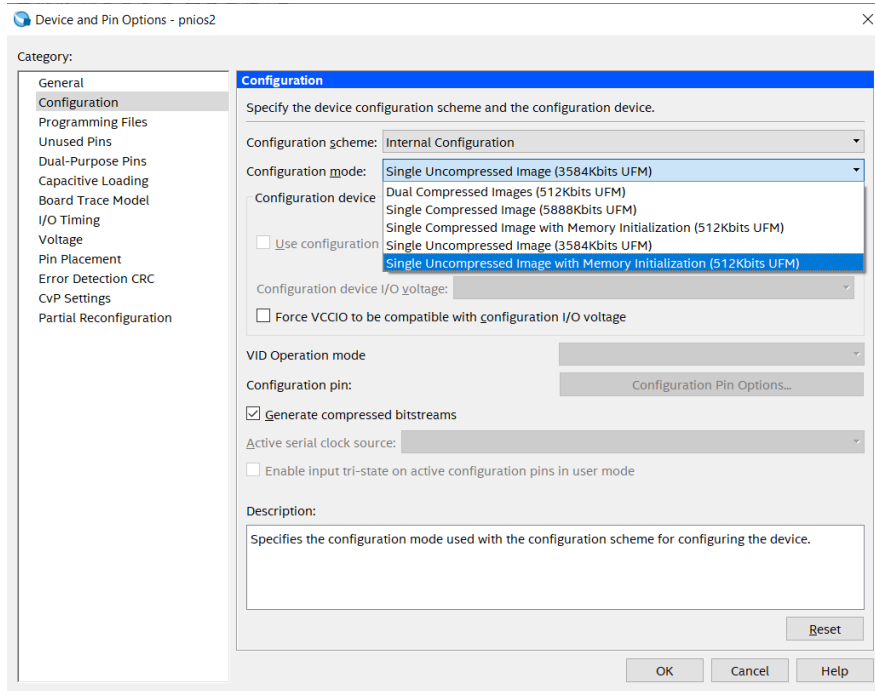
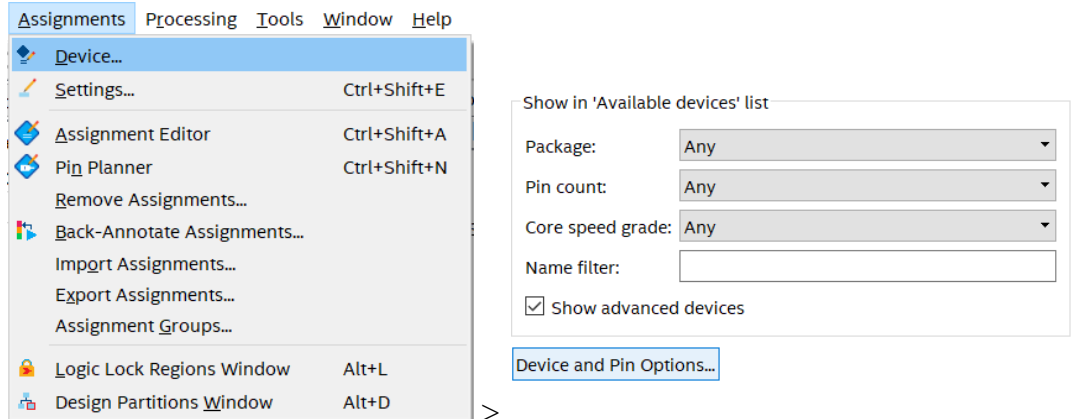


44. Berikan location pin assignment pada fisik DE10-LITE sesuai dengan datasheet → Run I/O assignment analysis.

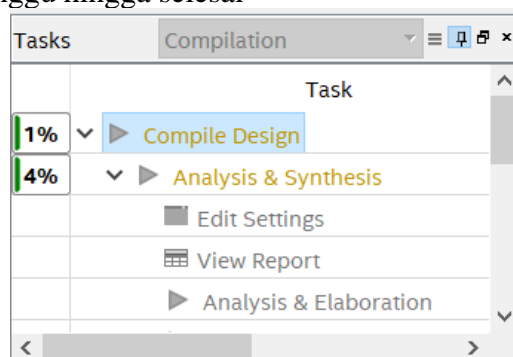




45. Setelah selesai, pilih **Assignment > Device > Device and pin options > configurations > ubah mode menjadi single uncomposed image with memory initialization > ok > ok.**



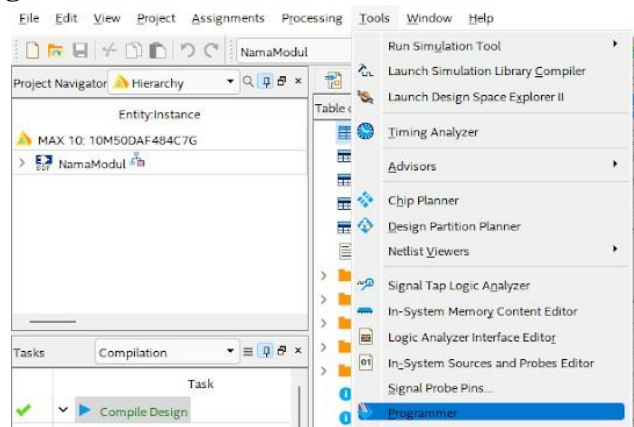
46. **Compile design.** Tunggu hingga selesai



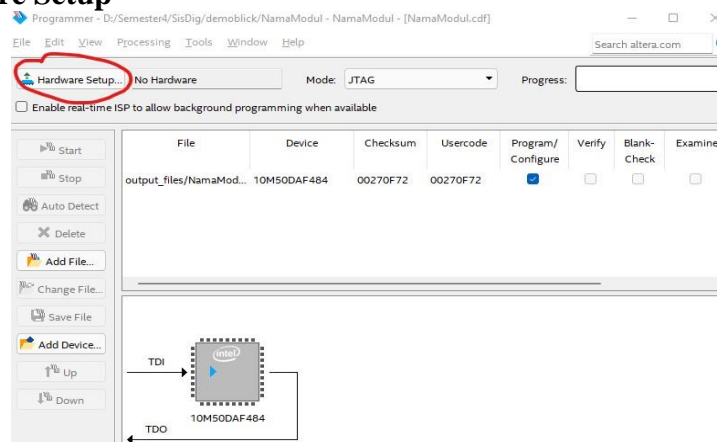
## Modul Praktikum

47. Hubungkan FPGA DE10-Lite ke Komputer/Laptop

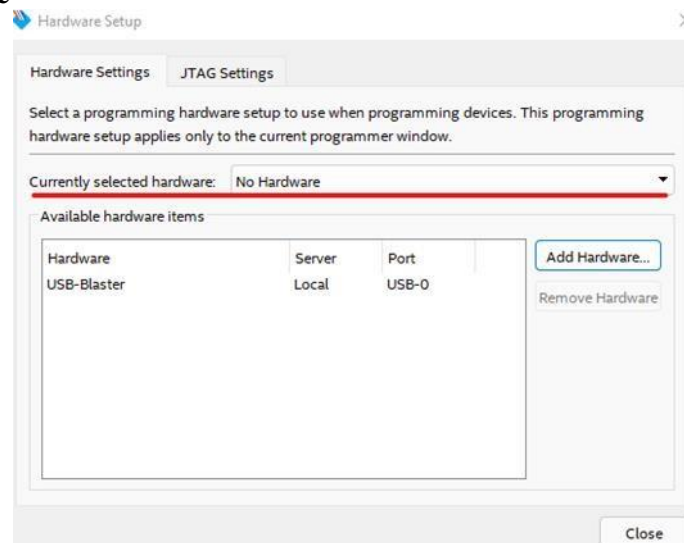
48. Klik **Tools > Programmer**



49. Klik **Hardware Setup**



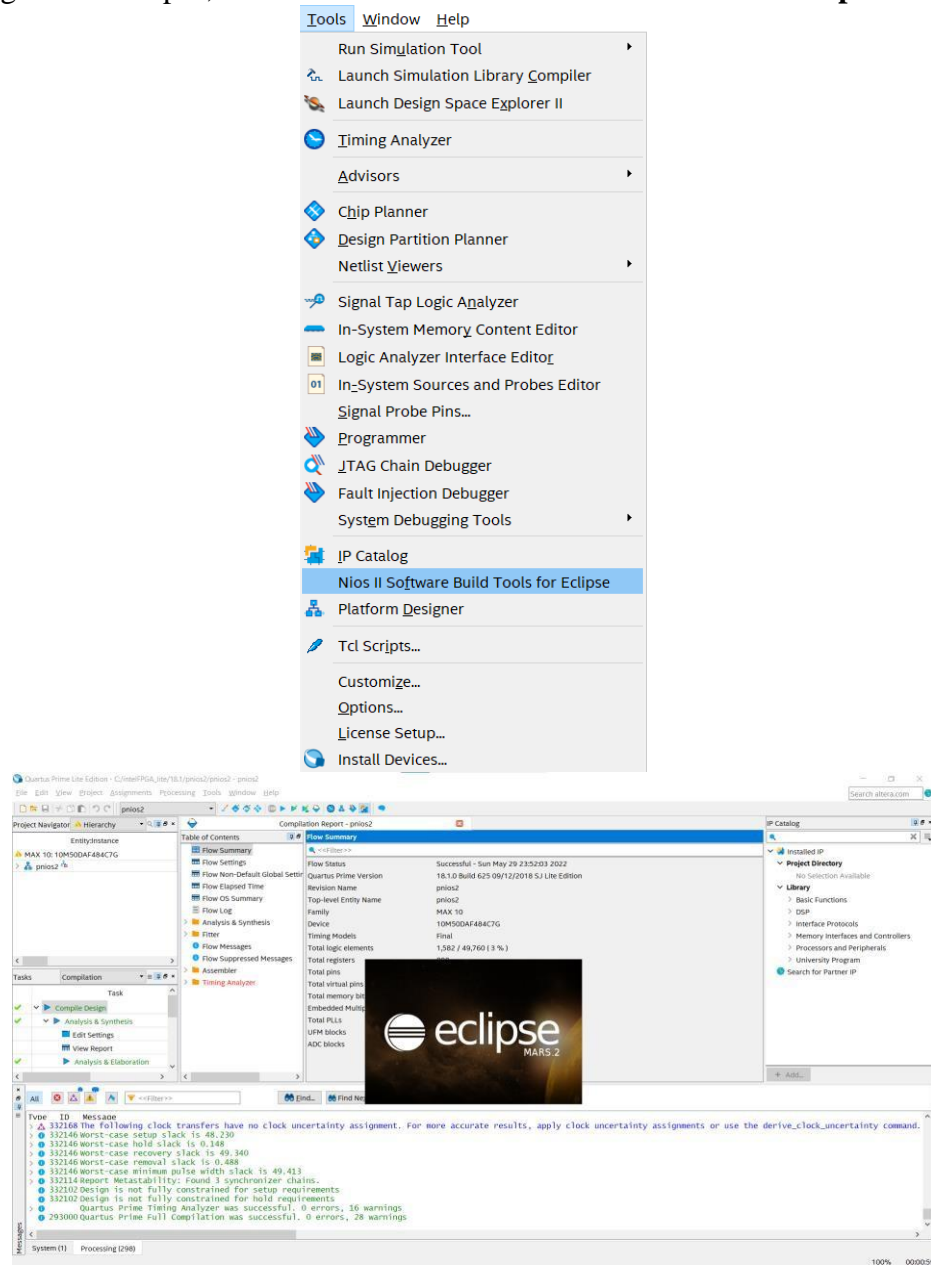
50. Pada **Currently Selected Hardware** ganti **No Hardware** Menjadi **USB-Blaster**.  
Lalu klik **Close**



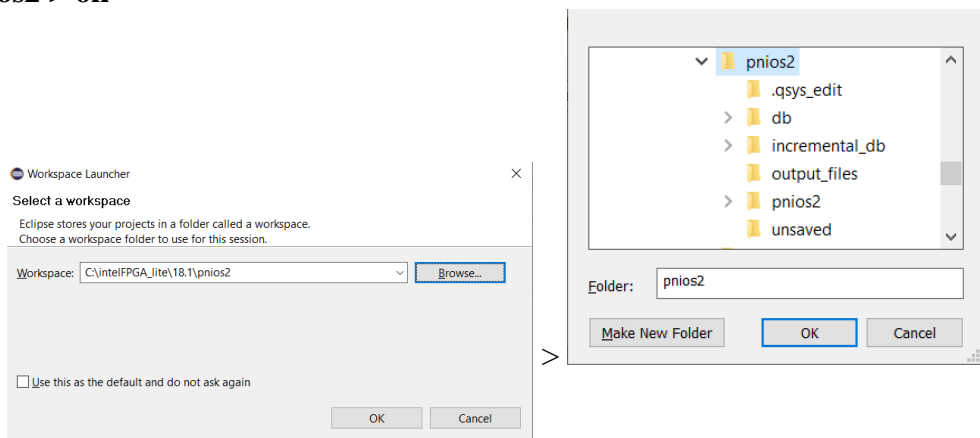
51. Kemudian centang program/configure, Klik **Start**, tunggu sampai success



52. Setelah selesai tahap generating hardware, kemudian tahap creating software menggunakan eclipse, klik **tools > nios ii software build tools for eclipse**

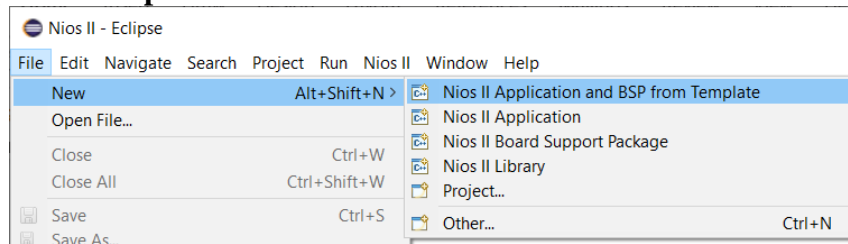


53. Kemudian pilih **workspace** sesuai project yang telah **dibuat > browse > pilih folder pnios2 > ok**

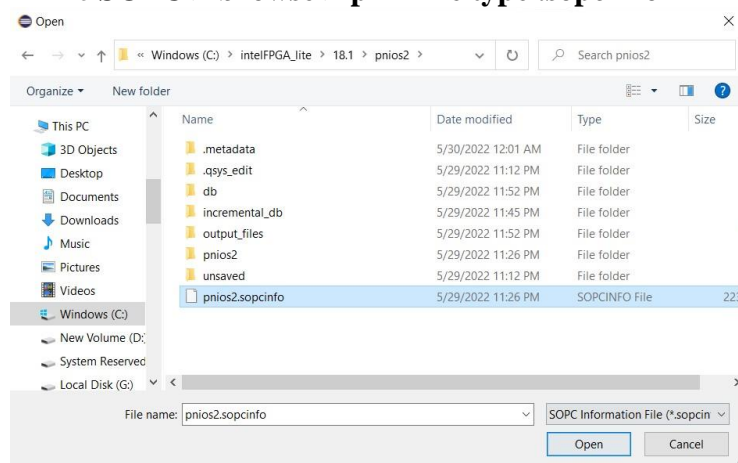




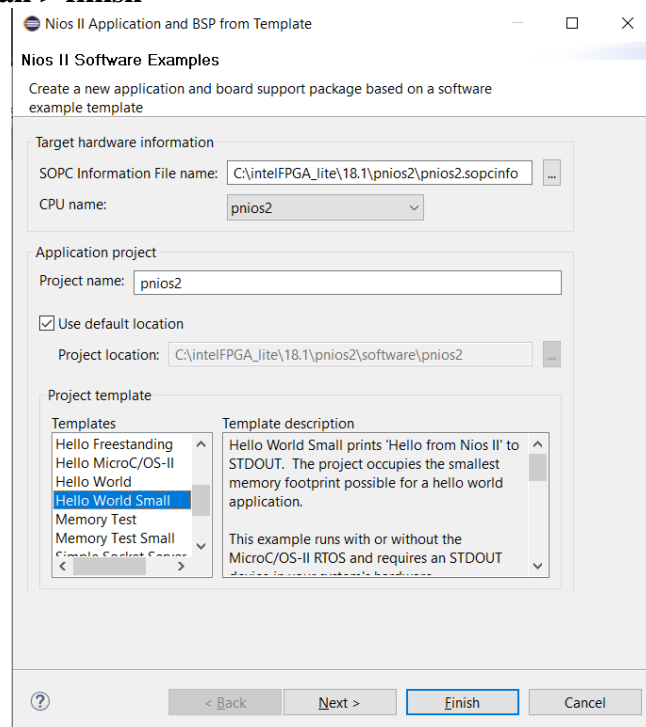
54. Kemudian terbuka workspace pnios ii – eclipse, pilih **file > new > nios ii application and BSp from template**



55. Kemudian pilih file **SOPC > browse > pilih file type .sopcinfo**

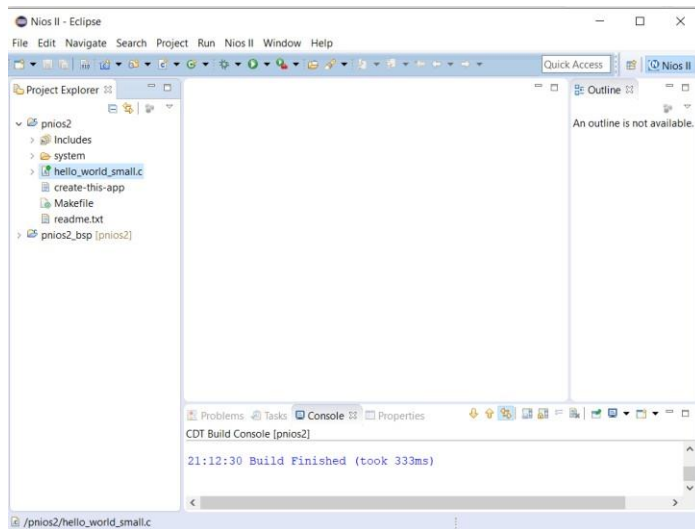


56. Kemudian project name : **pnios2** (sesuaikan nama project quartus) > pilih template **hello world small > finish**

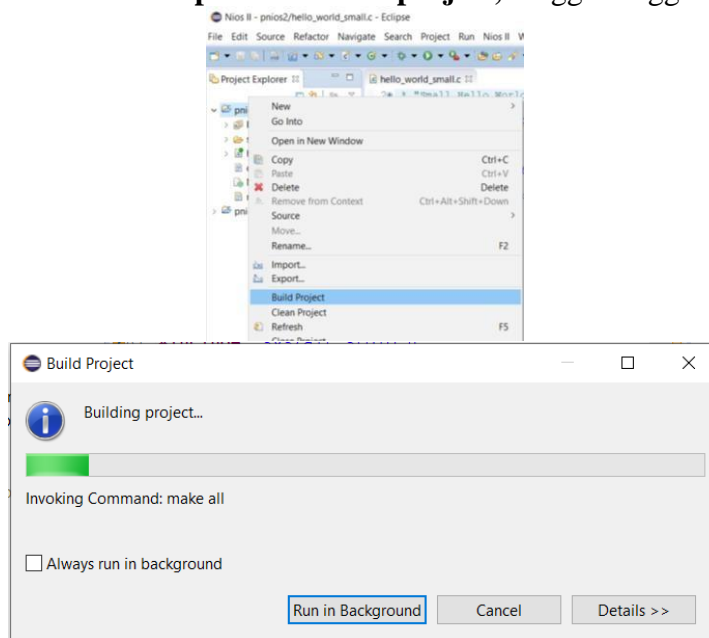


## Modul Praktikum

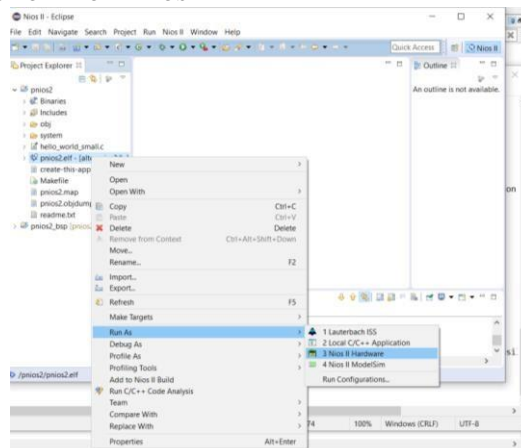
57. Maka akan muncul 2 project, **double klik** hello world small.c pada **folder pnios2** untuk melihat isi file



58. Kemudian klik kanan **folder pnios2 > build project**, tunggu hingga selesai.

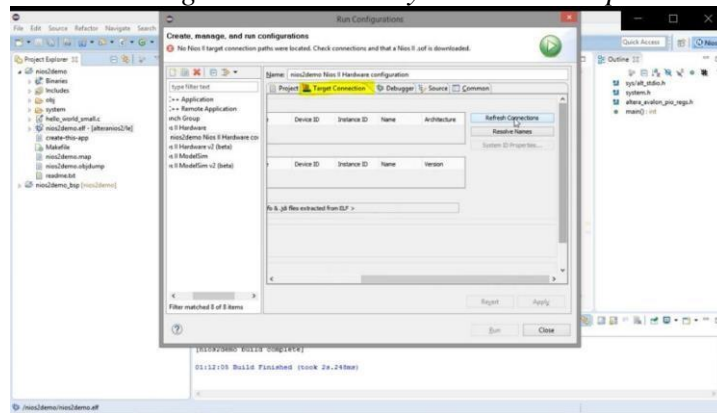


59. Kemudian klik kanan pada **file .elf > run.as > nios ii hardware** > ketika berhasil maka akan muncul hellon from nios ii



## Modul Praktikum

60. pilih **Target Connection**, lalu **Refresh connection**, selanjutnya centang “**ignore mismatch system ID**” dan “**ignore mismatch system timedtamp**”



61. Klik **Run**

