

MODUL 13

IMPLEMENTASI SOFTCORE PROCESSOR NIOS II

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

Modul Praktikum

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

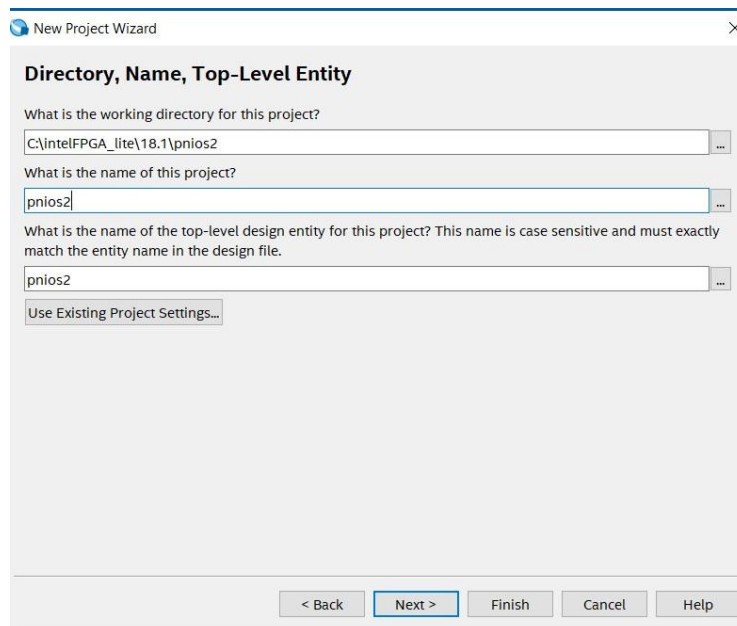
14.3 Lembar Kegiatan Praktikum Modul 14

14.3.1 Alat dan Bahan

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

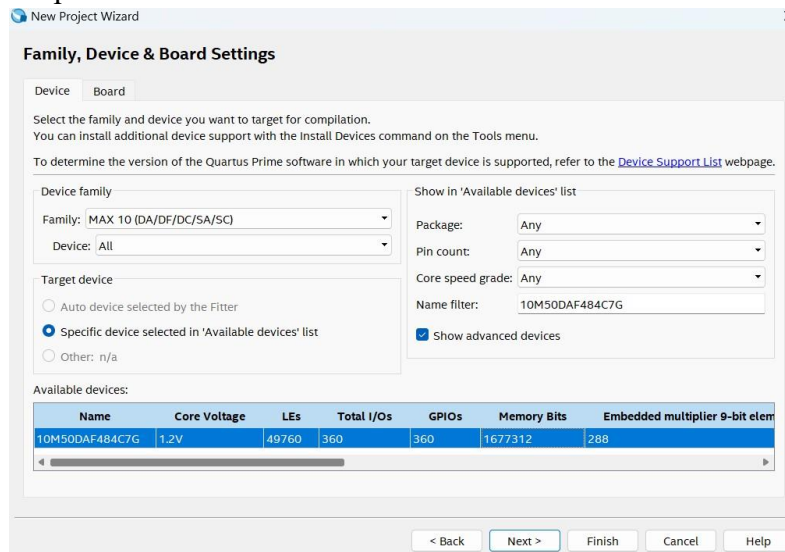
14.3.2 Langkah Praktikum Modul 12

1. Buat Directory baru **\pnios2** dan Nama Project **pnios2**, lalu klik **Next**.



Modul Praktikum

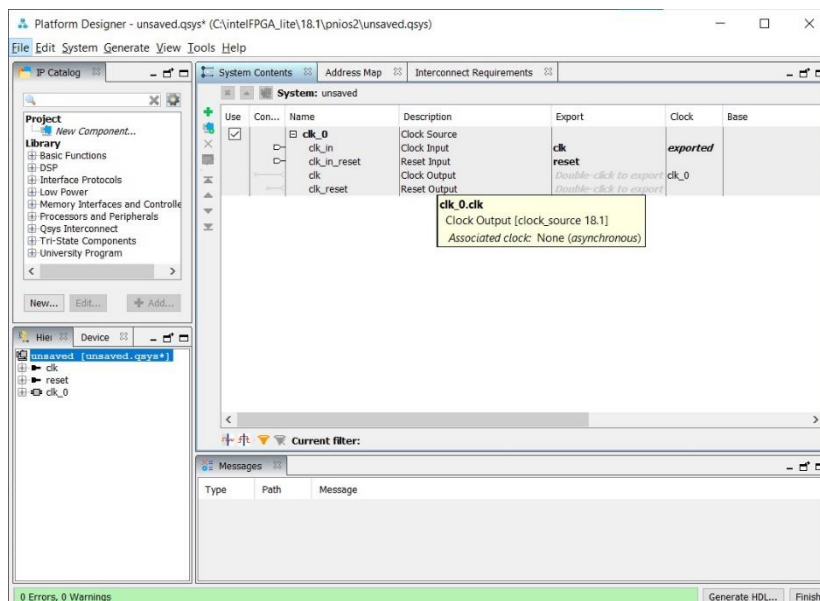
- Lalu ganti **Family** ke **MAX 10 (DA/DF/DC/SA/SF/SC)**, kemudian ketikkan di **Nama Filter 10M50DAF484C7G**, lalu klik Available Device yang tersedia, kemudian klik **Next** sampai ke **Finish**.



- Kemudian buat file baru dengan cara klik **Platform Designer (skematik) qsys (Double Click)**.

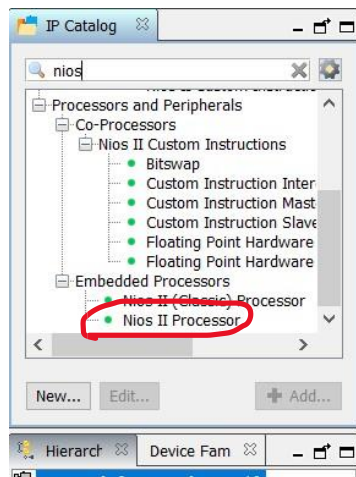


- Setelah muncul workspace (clock (default)), Masukkan komponen.

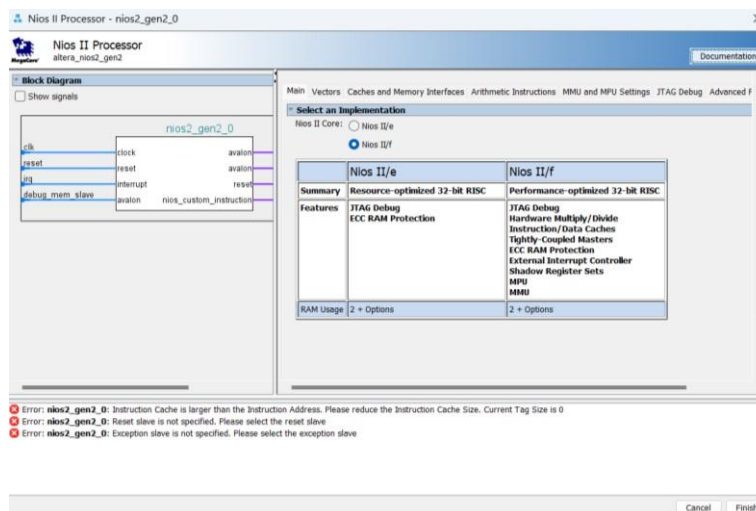


Modul Praktikum

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



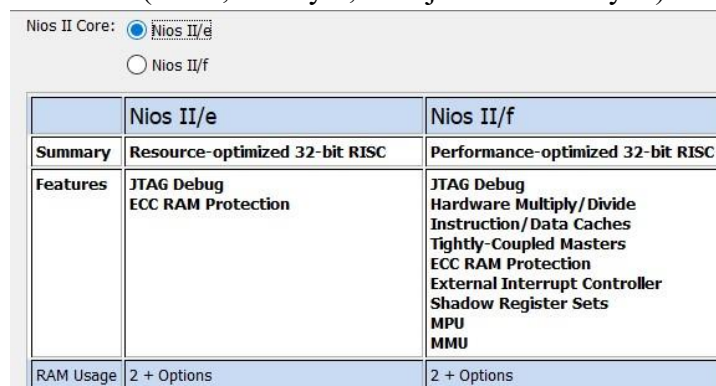
- Window workspace nios ii prosesor akan terbuka.



- Terdapat 2 pilihan Nios II core:

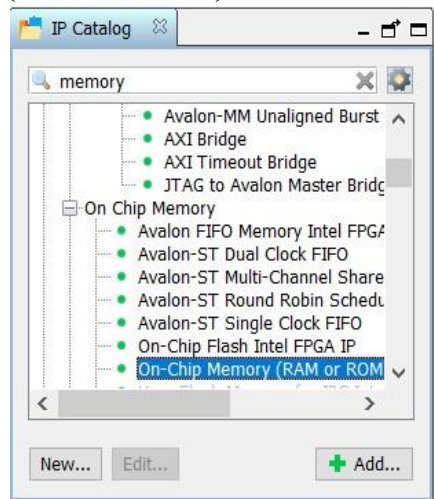
Nios II / e (ekonomis/s, gratis tidak ada lisensi sama sekali, fitur sedikit) praktikum hanya menggunakan JTAG saja

Nios II / f (faster, berbayar, fitur jauh lebih banyak). Pilih **ekonomis** > **finish**.

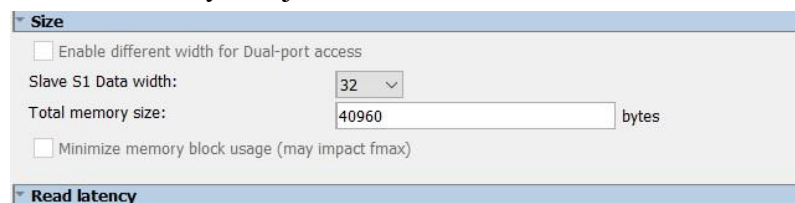


Modul Praktikum

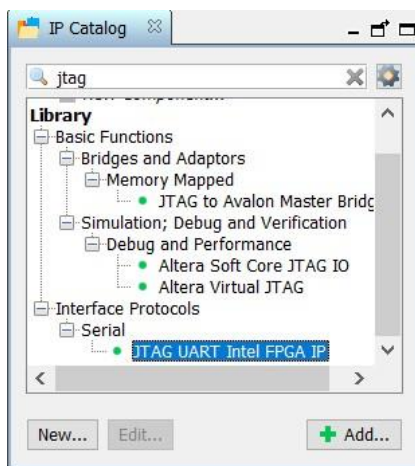
8. Tambah komponen memori, ketik **Memory** > scroll kebawah > pilih **On-Chip Memory (RAM or ROM) intel FPGA IP**. Terbuka workspace on-chip memory (RAM or ROM).



9. Tambah memory menjadi **40960** > **Finish**.

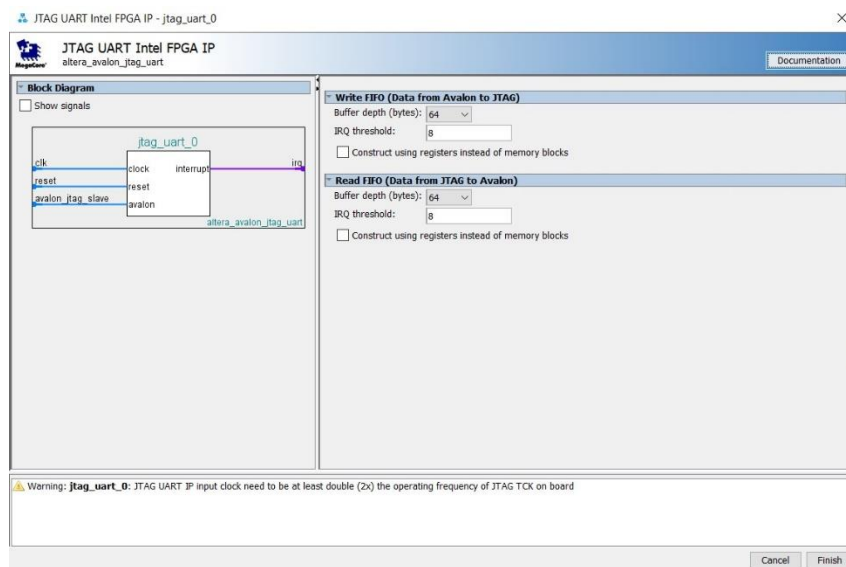


10. Kemudian tambahkan komonen JTAG UART, ketik **JTAG** > scroll kebawah > pilih **JTAG UART Intel FPGA IP**. Terbuka workspace JTAG UART.

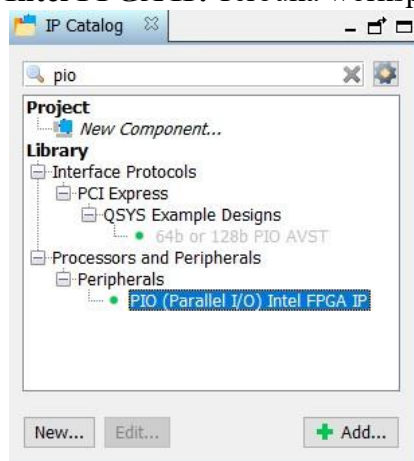


Modul Praktikum

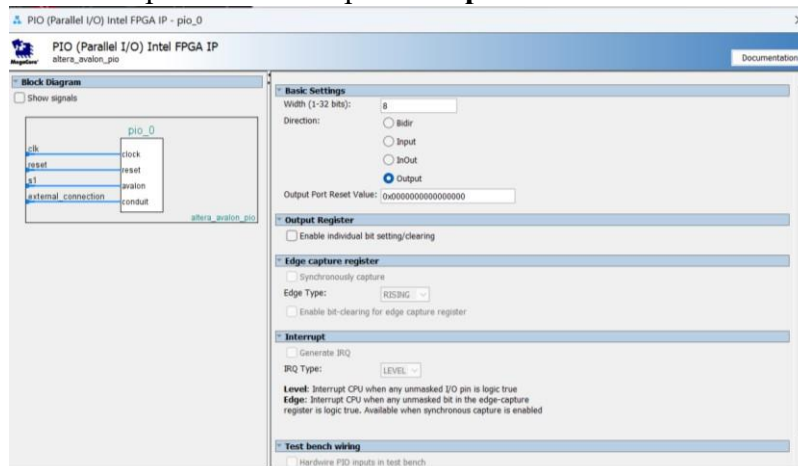
11. Kemudian biarkan sesuai **default** > **finish**.



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

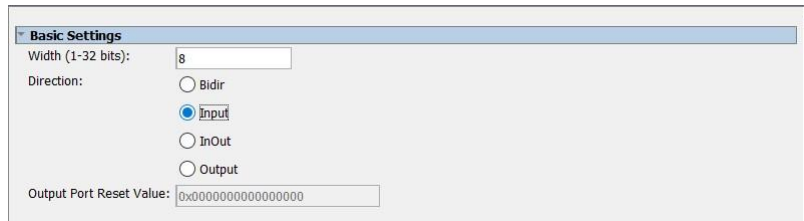


13. Kemudian pada **direction** pilih **Output** > **Finish**.

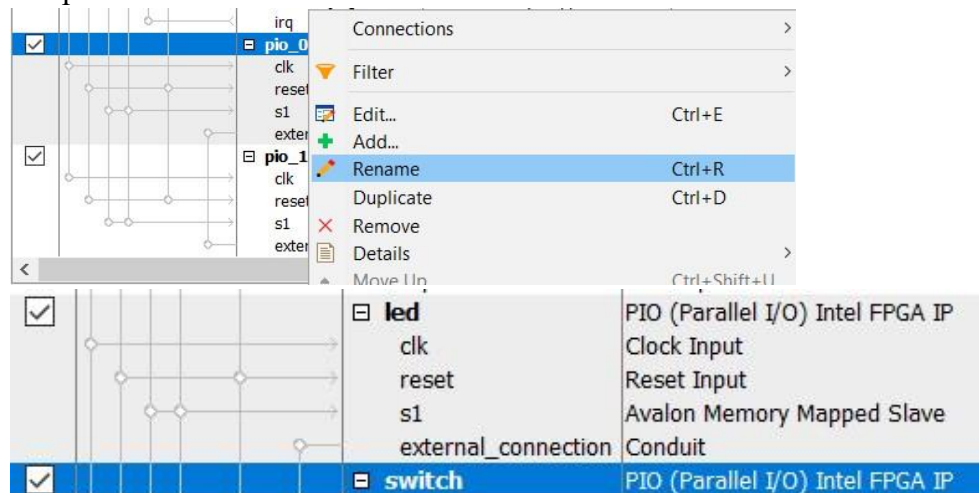


Modul Praktikum

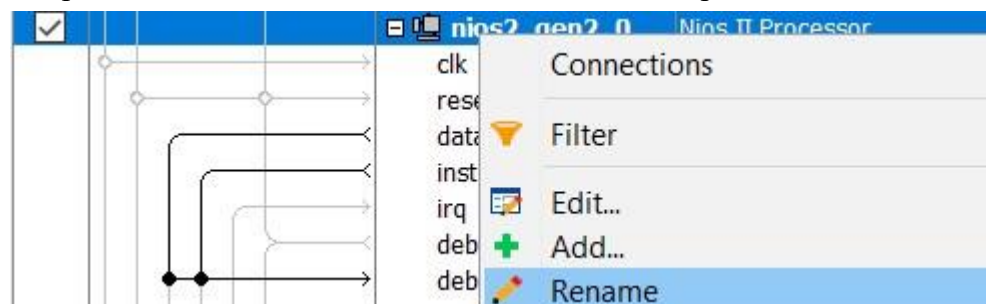
14. Kemudian tambahkan lagi PIO (parallel input/output), pada **direction** pilih **Input** > **Finish**.



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



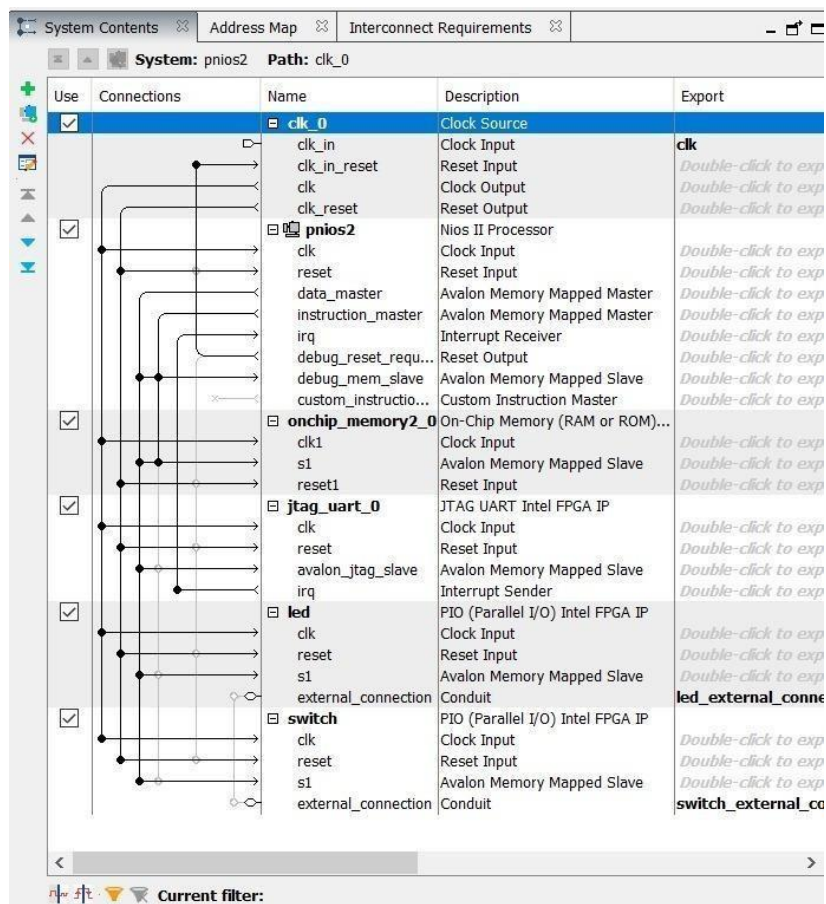
16. Kemudian **rename nios2** menjadi sesuai **nama project pnios2** dengan cara pilih komponen **klik kanan** > **rename**. Pastikan semua komponen sudah masuk dan di rename.



17. Kemudian wiring pada connections dengan klik setiap percabangan sambungan atau simbol "└─"

- 18. **Hubungkan** semua clk yang ada pada nios, onchip, jtag, led, switch
- 19. **Hubungkan** semua clk_reset yang ada pada nios, onchip, jtag, led, switch
- 20. **Hubungkan** clk_in_reset ke satu bagian saja yaitu debug_reset_request
- 21. **Hubungkan** data_master ke semua slave agar semua bisa menerima data
- 22. **Hubungkan** instruction_master hanya terhubung pada s1 onchip memory saja
- 23. **Hubungkan** irq prosesor dengan irq jtag (akan muncul dibagian belakang juga) .

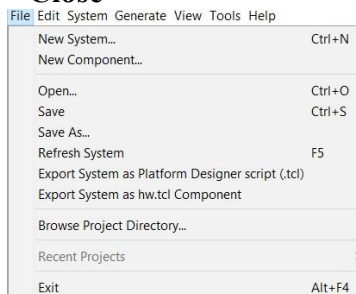
Modul Praktikum



24. Setelah selesai wiring, kemudian pada **led** dan **switch** terdapat **external_connection**, double click pada **Export** maka akan muncul **penamaannya** > enter

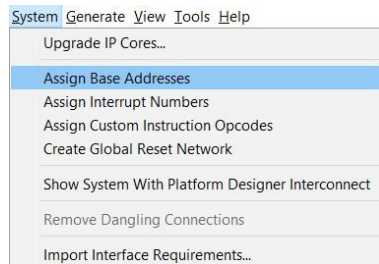
external_connection	Conduit	led external connection	
☒ switch	PIO (Parallel I/O) Intel FPGA IP		
clk	Clock Input	Double-click to export	clk_0
reset	Reset Input	Double-click to export	[clk]
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]
external_connection	Conduit	Double-click to export	

25. Kemudian simpan file, **File > Save** > ubah nama file **pnios 2** (sesuai nama file quartusnya) > **Close**

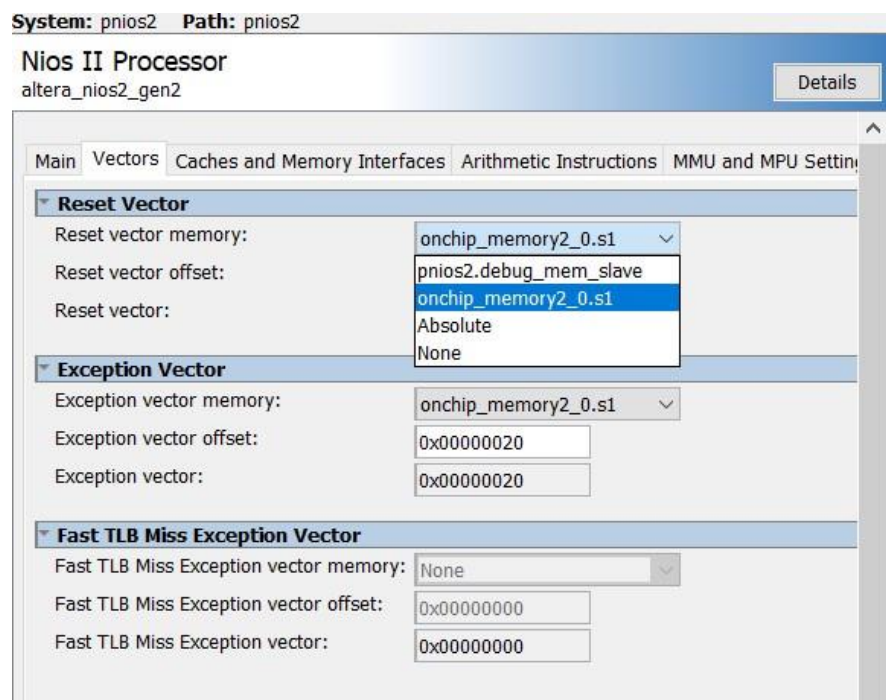


Modul Praktikum

26. Kemudian pilih **System > Assign Base Address**

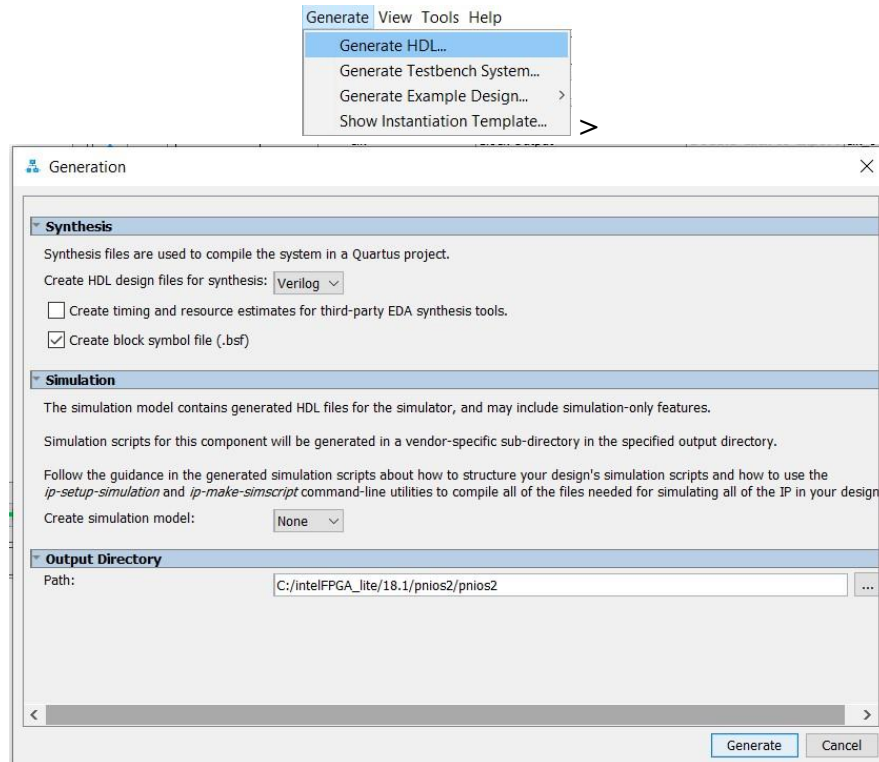


27. Jika masih terdapat error, Double Click pada **pnios2** > pilih **vektor** > pada **Reset Vector Memory** pilih **On Chip Memory** > pada **Exception Vector Memory** pilih **onchip_memory2_0.s1** > **close**, sehingga error tidak ada lagi, simpan file Kembali.

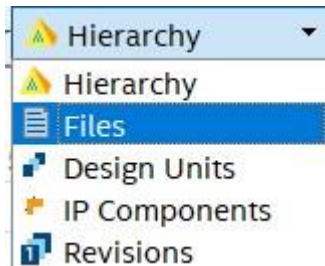


Modul Praktikum

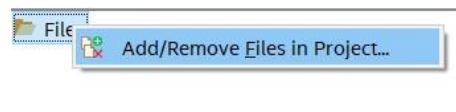
28. Kemudian pilih **Generate hardware > generate HDI > create HDI design file for syntseis menjadi verilog > pilih create block symbol file(.bsf) > generate. tunggu hingga selesai > close > close lagi**



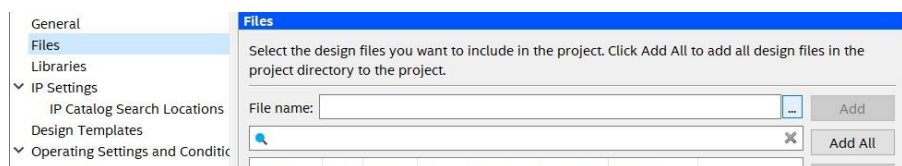
29. Kemudian muncul pemberitahuan berhasil membuat **ip variation in the file > ok** 37. Kemudian pilih **Hierarchy**



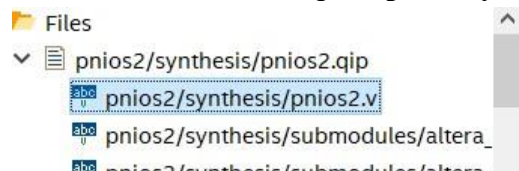
38. Kemudian pilih **Files > klik kanan Files > Add / Remove file**



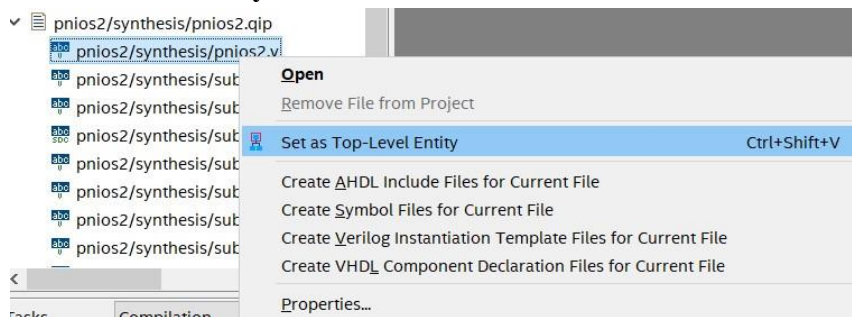
39. Setelah itu pilih **Files > pilih Browse > pilih folder pnios2 > synthesis > pilih file type.qip > Apply > Ok**



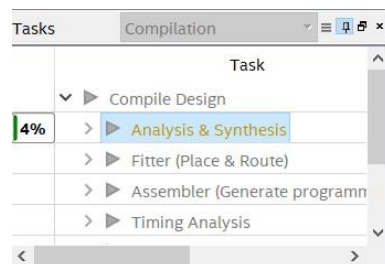
40. Kemudian double click pada **pnios/sythesis/pnios.v** untuk melihat verilognya.



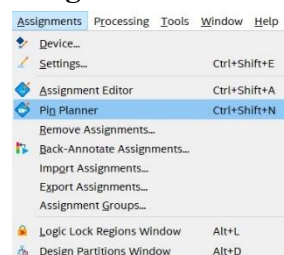
41. Kemudian klik kanan pada **pnios/sythesis/pnios.v** > **Set As Top Level Entity** > **balik ke Hierarchy**



42. Pilih **Analysis & Synthesis**, hingga selesai

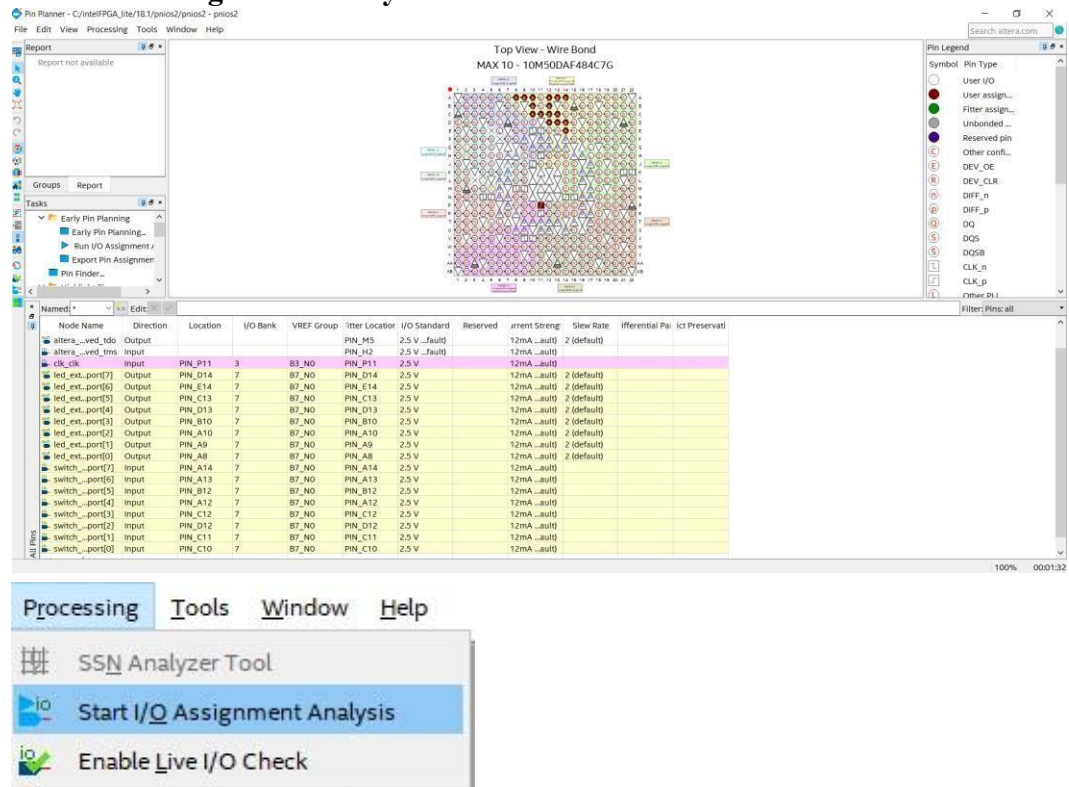


43. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu **Assignments** → **Pin Planner**

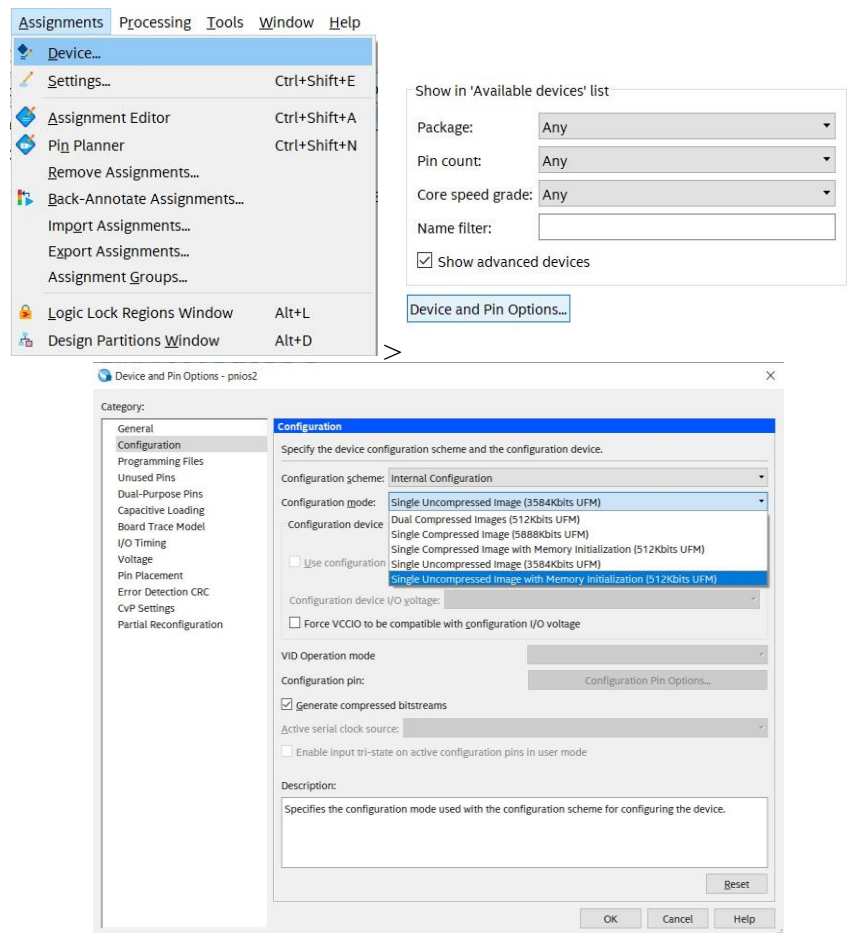


Modul Praktikum

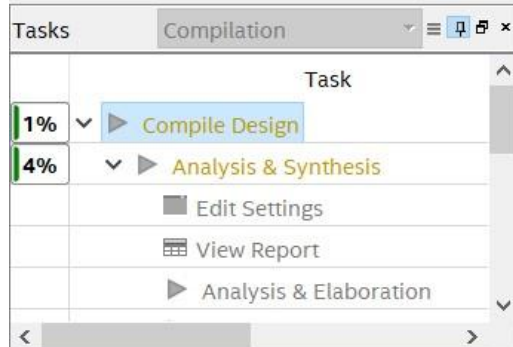
44. Berikan **location** di **Pin Assignment** pada fisik **DE10-LITE** sesuai dengan datasheet
→ **Run I/O Assignment Analysis.**



45. Setelah selesai, pilih **Assignment > Device > Device and pin options > configurations > ubah mode menjadi Single Uncomposed Image With Memory Initialization > Ok > Ok.**



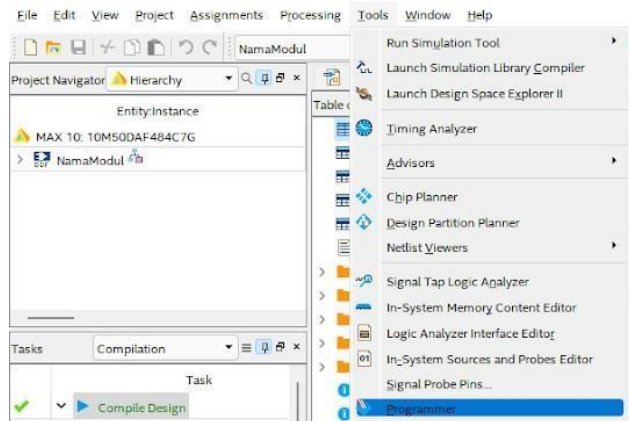
46. Compile Design. Tunggu hingga selesai



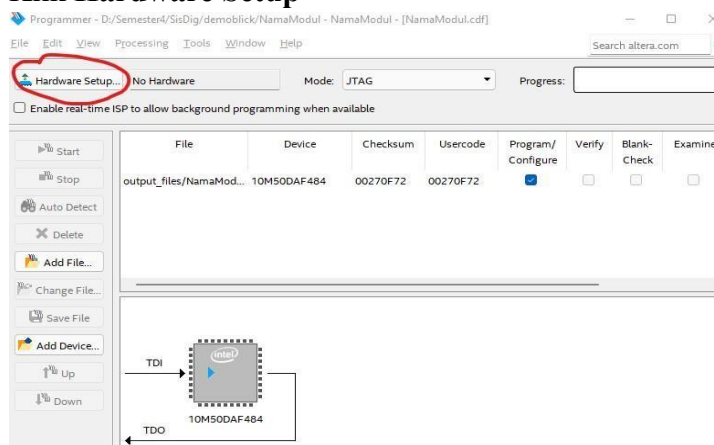
Modul Praktikum

47. Hubungkan FPGA DE10-Lite ke Komputer/Laptop

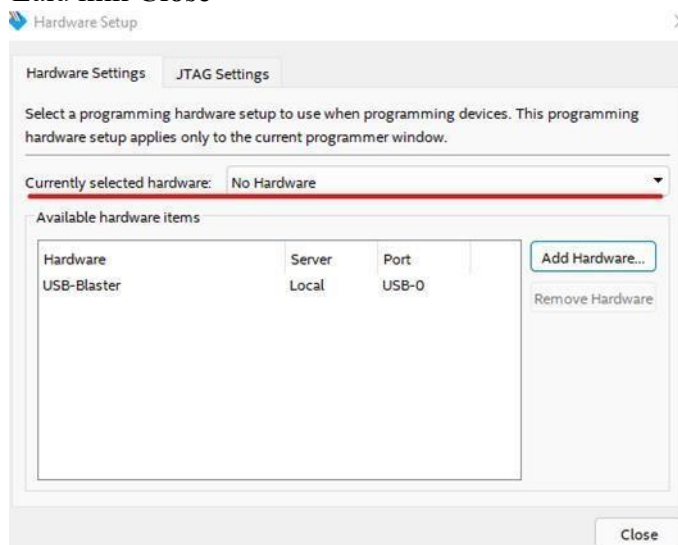
48. Klik Tools > Programmer



49. Klik Hardware Setup

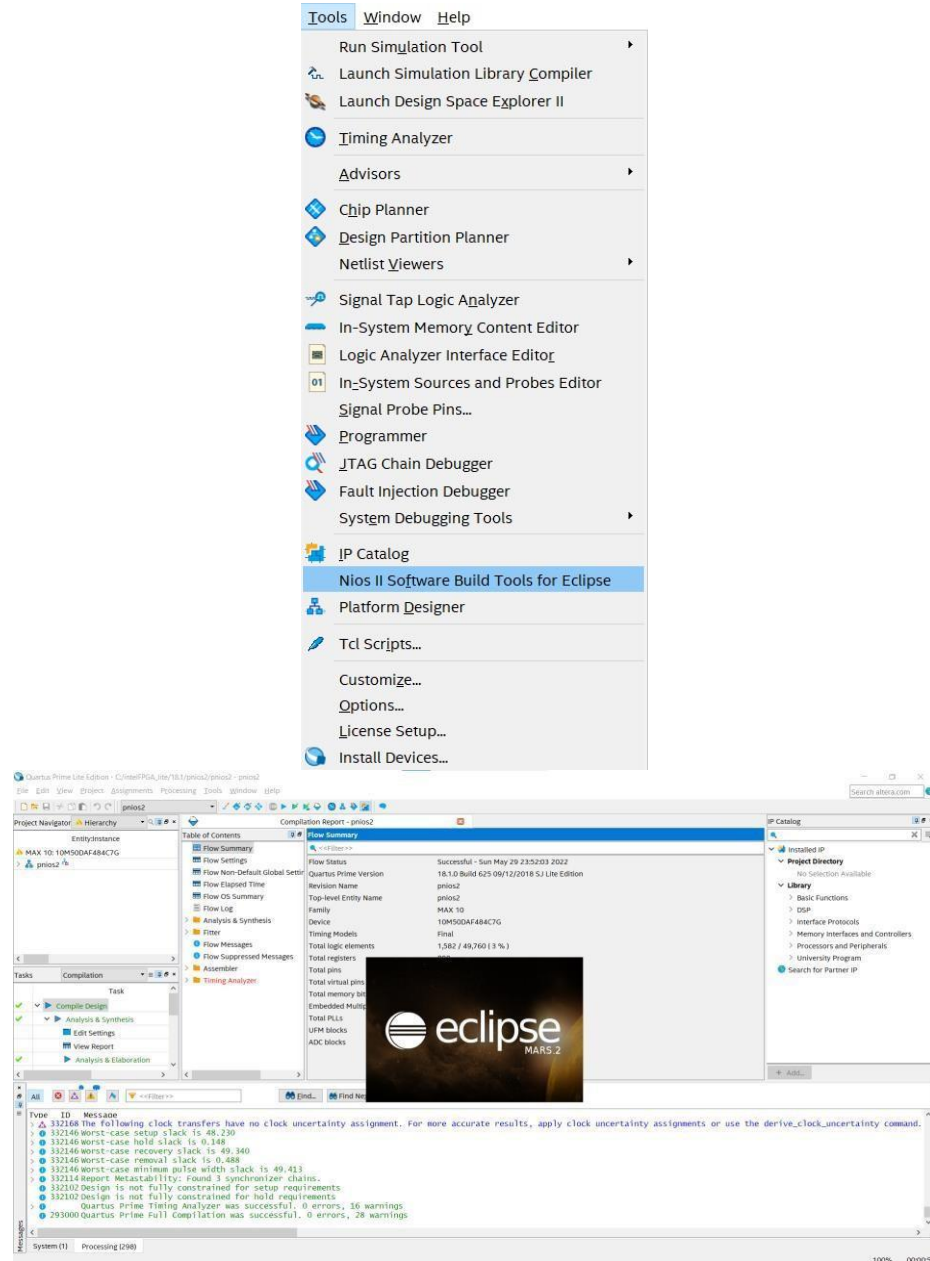


50. Pada Currently Selected Hardware ganti No Hardware Menjadi USB-Blaster. Lalu klik Close



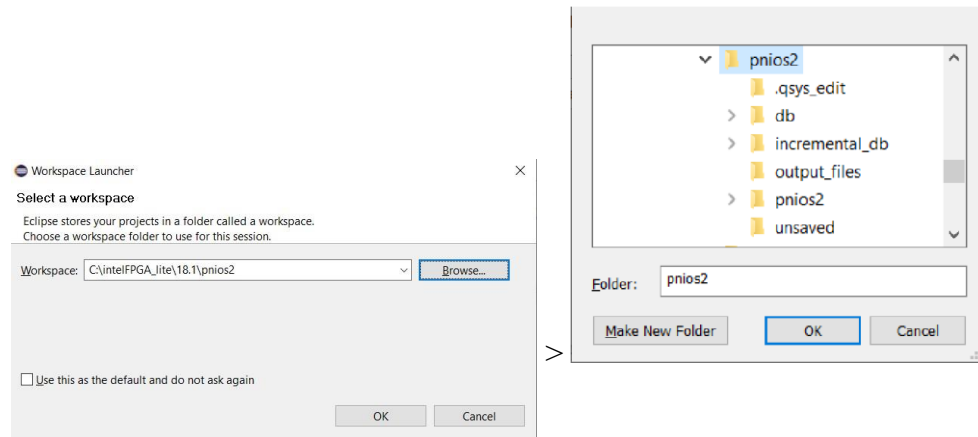
Modul Praktikum

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**

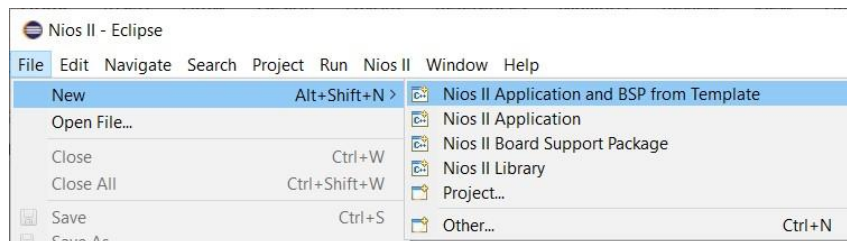


Modul Praktikum

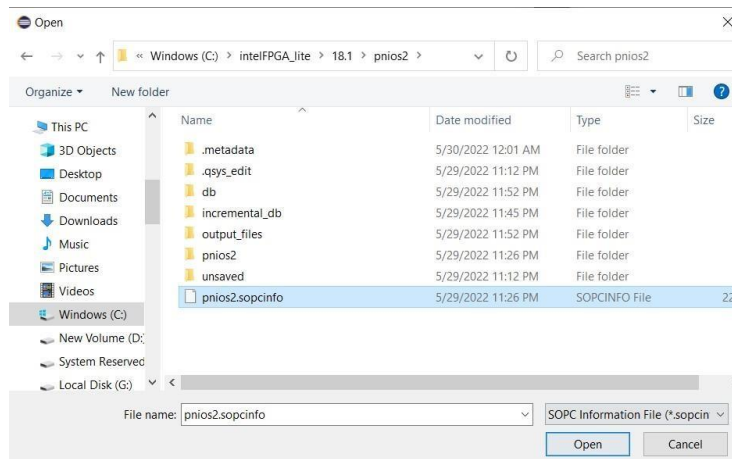
53. Kemudian pilih **Workspace** sesuai project yang telah dibuat > **browse** > **pilih folder pnios2** > **ok**



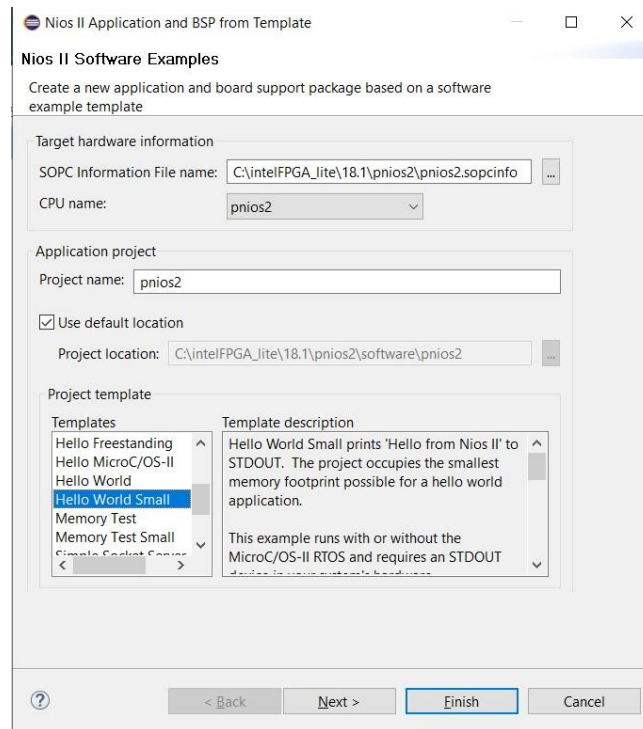
54. Kemudian terbuka workspace pnios ii – eclipse, pilih **file** > **new** > **Nios II application and BSP from template**



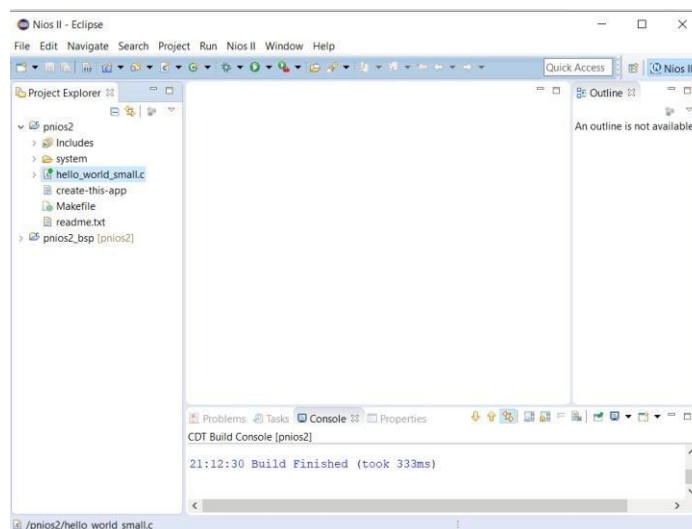
55. Kemudian pilih file **SOPC** > **browse** > **pilih file pnios2 .sopcinfo**



56. Kemudian project name : **pnios2** (sesuaikan nama project quartus) > **pilih template Hello World Small > Finish**

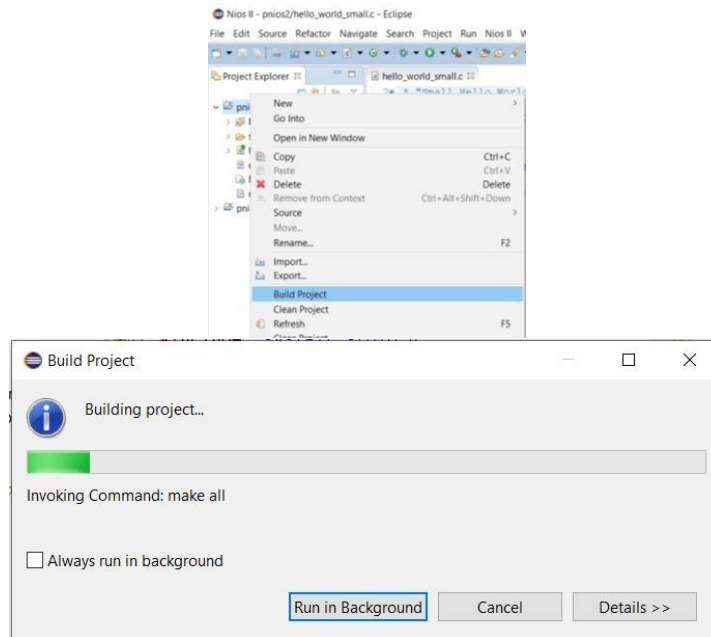


57. Maka akan muncul 2 project, **double click** `hello_world_small.c` pada **folder pnios2** untuk melihat isi file

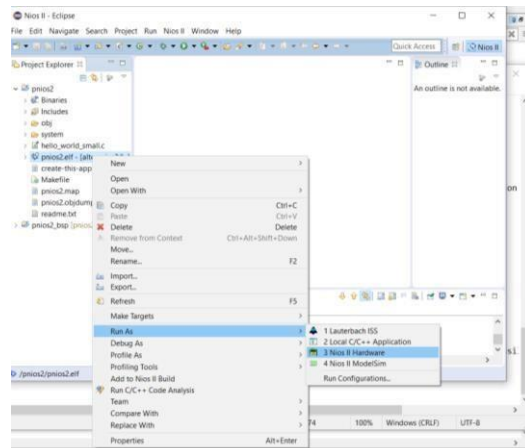


Modul Praktikum

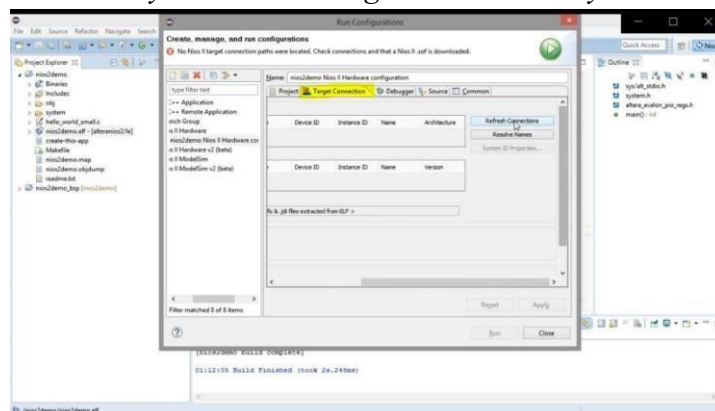
58. Kemudian klik kanan folder **pnios2** > **build project**, tunggu hingga selesai.



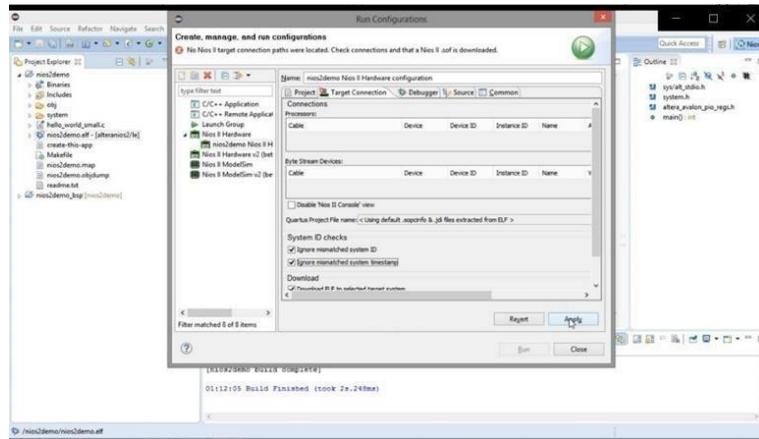
59. Kemudian klik kanan pada file **.elf** > **run.as** > **nios ii hardware** > ketika berhasil maka akan muncul hellon from nios ii



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



61. Klik *Apply* dan *Run*.



62. Berikut adalah hasil nya

