

1 D4 - TEKKOM B

Pengenalan FPGA dan ISE Design Tool Xilinx



Nama	:	Septian Bagus Jumentoro
Kelas	:	1 – D4 Teknik Komputer B
NRP	:	3221600039
Dosen	:	Reni Soelistijorini B.Eng., M.T
Mata Kuliah	:	Praktikum Rangkaian Logika 2
Hari/Tgl. Praktikum	:	Jum`at, 04 Maret 2022



I. Tujuan

1. Dapat membuat *new project* menggunakan software ISE Xilinx.
2. Dapat membuat program rangkaian gerbang logika dasar dan memasukan program VHDL ke dalam *Board Xilinx Spartan-3*

II. Teori

a) FPGA dan VHDL

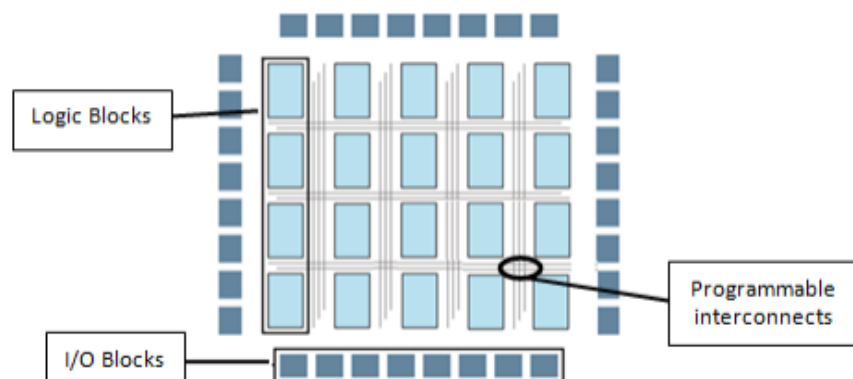
Field Programmable Gate Arrays (FPGA) adalah programmable device besar yang tersusun atas modul-modul logika independen yang dapat dikonfigurasi oleh pemakai yang di hubungkan melalui kanal-kanal routing yang dapat di program.

FPGA mempunyai kelebihan sebagai berikut :

1. Dikonfigurasi oleh *End User*
2. Tidak memerlukan proses Fabrikasi
3. Tersedia solusi yang mendukung chip customixed VLSI

FPGA ini sendiri juga merupakan IC tipe HDL yang mana pemakai dapat mendesain hardware sesuai yang diinginkan di dalam IC FPGA. Perkembangan FPGA saat ini berlangsung begitu cepat, terdapat bermacam-macam keluarga FPGA dengan kebutuhan perancangan dan perangkat perancangan (design tools) yang berbeda. Xilinx merupakan salah satu perusahaan yang memproduksi FPGA disamping perusahaan lain yang juga memproduksi FPGA seperti ACTEL dan PLESSEY Semiconductor. Xilinx sendiri memproduksi beberapa jenis FPGA yaitu VIRTEX, SPARTAN, XC3000, XC4000, dan XC5000.

Secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga element utama yaitu: *Input/Output Blok (IOB)*, *Configurable Logic Block (CLB)*, dan *Programmable Interconnect*. Fungsi logika dan interkoneksi FPGA ditentukan oleh data konfigurasi ke dalam sel memori ini, baik yang dilakukan secara otomatis pada waktu catu daya diberikan maupun dengan membaca konfigurasi data dari eksternal Serial atau Byte Paralel PROM (mode master) atau Data dapat dituliskan pada FPGA dari eksternal device (mode slave dan peripheral).



Gambar 1. 1 Struktur Xilinx FPGA

Proses implementasi pada FPGA :

1. FPGAs diimplementasikan dengan *efficient CAD systems*
2. Design Entry dapat dilakukan dengan cara

- a. Schematic capture program,
 - b. VHDL / Verilog
3. Logic Optimization
 - a. Memodifikasi Boolean Expression untuk mengcopyimalkan penggunaan
 - b. Logic berbasis area dan kecepatan atau kombinasi keduanya
4. Teknologi Mapping
 - a. Transformasi dari Boolean Expression atau VHDL kedalam bentuk circuit pada FPGA logic blocks
 - b. Area optimization – meminimalkan penggunaan block
 - c. Delay Optimixation – meminimalkan jumlah stage pada critical path
5. Placement
 - a. Setelah memetakan rangkaian ke dalam blok logika, selanjutnya algoritma penempatan digunakan untuk meletakkan masing-masing blok ke dalam FPGA array.
 - b. Meminimalkan total panjang interkoneksi yang diperlukan untuk penempatan blok logika
6. Routing Software
 - a. Mengatur FPGA's wire segment dan menentukan programmable switches untuk menghubungkan antar FPGA blocks
 - b. Memastikan 100% koneksi telah terbentuk
 - c. Meminimalkan propagation delay pada *time-critical connections*.
7. Programming unit
 - a. Mengkonfigurasi chip setelah placement dan routing step tuntas
 - b. Keseluruhan proses memakan waktu antara beberapa menit sampai beberapa jam

Secara umum proses kerja FPGA adalah sebagai berikut :

1. **Design Entry** dengan cara schematic, ABEL, VHDL, dan Verilog
2. **Implementation** yaitu Placement dan Routing serta bitsream generation. Juga analyze timing, view layout
3. **Download** meng-upload bitstream ke Hardware (FPGA chip)

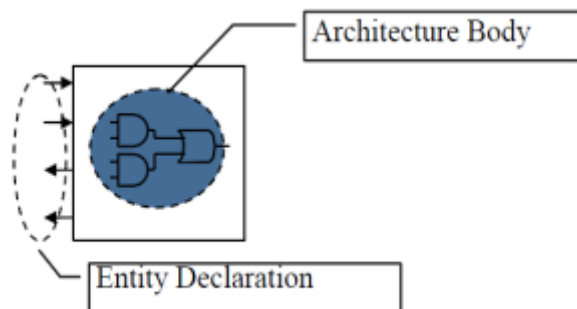
VHDL adalah singkatan dari **V** HSIC (*Very High Speed Integrated Circuits*) **H**ardware **D**escription **L**anguage. Pada pertengahan tahun 1980-an Departemen pertahanan AS dan IEEE mensposori perkembangan bahasa deskripsi perangkat keras dengan tujuan untuk mengembangkan sirkuit terintegrasi dengan kecepatan tinggi. Hal ini telah menjadi salah satu bahasa standart industri yang digunakan untuk menggambarkan sistem digital. Yang banyak digunakan Hardware lainnya bahasa deskripsi Verilog. Keduanya adalah bahasa yang memungkinkan anda untuk menggambarkan dan mensimulasikan sistem digital yang kompleks. Sebuah bahasa HDL ketiga adalah ABEL (Advanced Boolean Equation Language) yang khusus dirancang untuk Programmable Logic Device (PLD).

Meskipun bahasa terlihat sama seperti bahasa pemrograman konvensional terdapat beberapa perbedaan penting. Sebuah bahasa deskripsi hardware secara inhen paralel, yaitu perintah yang sesuai dengan gerbang logika yang dieksekusi (computed) secara paralel begitu masukan tiba. Sebuah program HDL meniru sistem fisik, biasanya digital.

Hal ini memungkinkan penggabungan spesifikasi waktu (delay gate) serta menggambarkan sistem sebagai interkoneksi dari komponen yang berbeda.

Saat ini para desainer menggunakan *hardware description languages (HDLs)* untuk mendesain system. Bahasa yang umum dipakai adalah HDL, VHDL, dan Verilog. Kesemua bahasa membolehkan user untuk mendesain system digital dengan menuliskan behavior dari rangkaian digital. Pemrograman ini juga dapat disimulasikan operasi rangkaian dan men-*syntheize*-kan implementasi secara rangkai nyata dari rangkaian atau membuat aplikasi IC yang spesifik. Pada modul ini, pemrograman yang dipakai adalah pemrograman VHDL. VHDL mirip dengan bahasa pemrograman komputer seperti C/C++. Sebagai contoh, memiliki konstruksi untuk tugas variabel, pernyataan bersyarat, loop, dan fungsi. Dalam bahasa pemrograman komputer, compiler digunakan untuk menerjemahkan kode sumber tingkat tinggi ke kode mesin. Di VHDL sendiri digunakan untuk menerjemahkan kode sumber untuk deskripsi dari sirkuit hardware sebenarnya yang mengimplementasikan kode. Dari uraian ini, yang kita sebut netlist, perangkat digital fisik dapat dibuat secara otomatis. Keakurasian kode juga dapat dites menggunakan simulasi.

VHDL adalah bahasa pemrograman yang digunakan untuk mendeskripsikan *logic circuit* yang dikehendaki. Secara umum struktur dari pemrograman VHDL terdiri atas dua bagian **ENTITY** dan **ARCHITECTURE**.

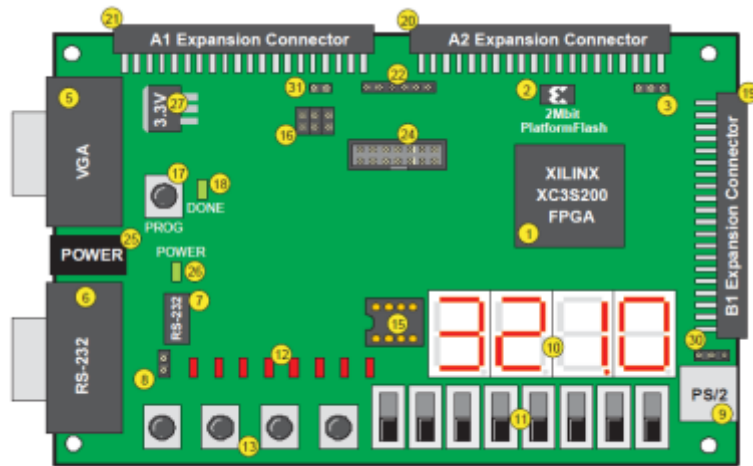


Gambar 1. 2 Struktur Pemrograman VHDL

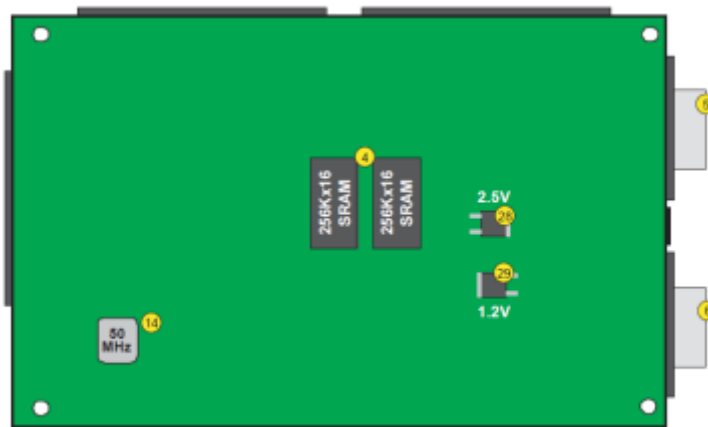
Bagian **ENTITY** menjelaskan spesifikasi pin-pin eksternal yang digunakan dari circuit atau rancangan yang akan dibuat. Bagian **ARCHITECTURE** menjelaskan atau mewakili fungsi sesungguhnya dari circuit atau rangkaian.

b) XILINX SPARTAN 3

Modul yang digunakan untuk mendesain chip adalah modul Xilinx Spartan 3. Fitur-fitur yang disediakan, seperti pada gambar 1.3 dan 1.4.



Gambar 1. 3 Modul Xilinx Spartan 3 tampak atas



Gambar 1. 4 Modul Xilinx Spartan 3 tampak bawah

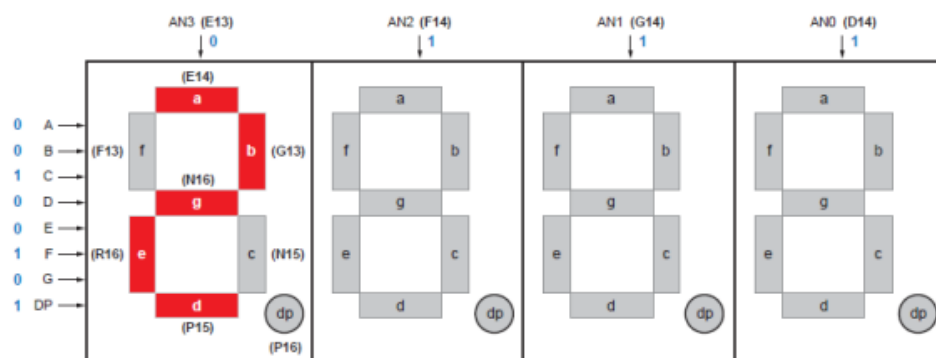
Secara ringkas fitur-fitur yang dimiliki oleh Xilinx Spartan 3 adalah :

1. 200,000-gerbang Xilinx Spartan-3XC3S200 FPGA in a 256-ball thin Ball Grid Array package (XC3S200FT256).
2. 2Mbit Xilinx XCF02S Platform Flash, in-system programmable configure PROM
3. Jumper Option allow FPGA application to read PROM data or FPGA configuration from other source
4. 1M-byte of Fast Asynchronous SRAM.
5. 3-bit, 8 color VGA display port.
6. 9-pin RS-232 Serial Port
7. RS-232 transceiver/level translator
8. Second RS-232 transmit and receive channel available on board test point
9. PS/2-style mouse/keyboard port
10. 4 karakter, seven-segment LED display
11. 8 slide switches
12. 8 LED outputs
13. 4 push button
14. 50 MHz crystal oscillator clock source
15. Socket for an auxiliary crystal oscillator clock source
16. FPGA configuration mode selected via jumper settings
17. Push button switch to force FPGA reconfiguration
18. Indikator LED ketika FPGA sukses dikonfigurasi

19. 40-pin expansion connection port to extend and enhance
20. 40-pin expansion connection port to extend and enhance
21. 40-pin expansion connection port to extend and enhance
22. JTAG port for low-cost download cable,
23. Digilent JTAG download/debugging cable connects to PC parallel port
24. JTAG download/debug port compatible with the xilinx Parallel Cable IV and MultiPRO Desktop Tool
25. AC power adapter input for included international unregulated +5V power Supply
26. Power -on indicator LED
27. On-board 3.3V regulator
28. 2.5V regulator
29. 1.2 regulator

FPGA koneksi ke Seven Segment.

Pada Seven Segment ada 8 jalur kontrol sinyal yang digunakan untuk menghidupkan LED untuk kombinasi angka tertentu. Dan setiap satu seven segment memiliki satu jalur control common anoda sebagai Enable, detail jalur tersebut seperti pada gambar 1.5.



Gambar 1. 5 Koneksi FPGA ke Seven Segment pada modul

Untuk tabel pin FPGA dengan seven segment ditunjukkan pada tabel dibawah ini.

Segment	FPGA Tabel
A	E14
B	G13
C	N15
D	P15
E	R16
F	F13
G	N16
DP	P16

Tabel 1.1 FPGA Koneksi port ke Seven Segment (aktif Low)

Anode Control	AN3	AN2	AN1	AN0
FPGA Pin	E13	F14	G14	D14

Tabel 1.2 sinyal Common Enable Seven Segment (Kontrol Anoda)

FPGA koneksi dengan Switch

Pada board Spartan 3, tombol Switch dilambangkan dengan SW. Misal Switch ke-1 ditulis dengan SW1, dimodul ini switch yang dimiliki sebanyak 8buah (SW7-SW0).

Switch	SW7	SW6	Sw5	SW4	SW3	SW2	SW1	SW0
FPGA pin	K13	K14	J13	J14	H13	H14	G12	F12

Tabel 1.3 Koneksi FPGA dengan SWITCH

Ketika SW pada posisi UP atau posisi ON, pin FPGA terhubung ke VCC/Logic High. Sedangkan jika SW pada posisi Down atau posisi OFF, pin FPGA terhubung ke Ground/Logika Low.

FPGA koneksi dengan Push Button

Pada board Spartan 3 memiliki 4 tombol push button yang ditukiskan pada board dengan BTN (BTN3-BTN0)

Push Button	BTN3	BTN2	BTN1	BTN0
FPGA Pin	L14	L13	M14	M13

Tabel 1.4 Koneksi FPGA dengan Push Button

Ketika menekan push button, pin FPGA terhubung dengan VCC

FPGA koneksi dengan LED

Pada board Spartan 3 memiliki 8 LED yang dituliskan pada board dengan LD (LD7-LD0)

Led	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
FPGA Pin	P11	P12	N12	P13	N14	L12	P14	K12

Tabel 1.5 Koneksi FPGA ke LED

Katoda dari Led terhubung ke Ground dengan resistor 270Ω.

c) Xilinx ISE Design Suite 13.1

Xilinx ISE (integrated Synthesis Environment) adalah sebuah perangkat lunak yang diproduksi oleh Xilinx untuk sintesis dan analisa desai HDL, yang memungkinkan untuk *compile* desain, analisa waktu, RTL diagram, dan menkonfigurasi perangkat target/device dengan programmer.

Xilinx ISE digunakan hanya untuk produk FPGA dari Xilinx. Xilinx ISE digunakan untuk mensitesa rangkaian dan desain sedangkan Model Sim logic Simulator digunakan untuk pengujian system level testing, juga terdapat fitur tambahan termasuk program lain seperti *Embedded Development Kit (EDK)*, *Software Developments Kit (SDK)*, dan ChipScope Pro.



Gambar 1. 6 ISE Project Navigator

III. Alat dan Bahan

Alat dan Bahan untuk praktikum ini adalah :

1. PC beserta software ISE Design Suite 13.1
2. Modul Xilinx Spartan 3
3. Downloader JTAG
4. Power Supply DC 5V

IV. Langkah Percobaan

Rangkailah Modul Xilinx Spartan 3 dengan menghubungkan power Xilinx ke 5VDC. Kemudian pasanglah downloader USB untuk koneksi PC dengan chip pada Board. Posisi jumper pada posisi default (program tidak tersimpan dalam ROM).

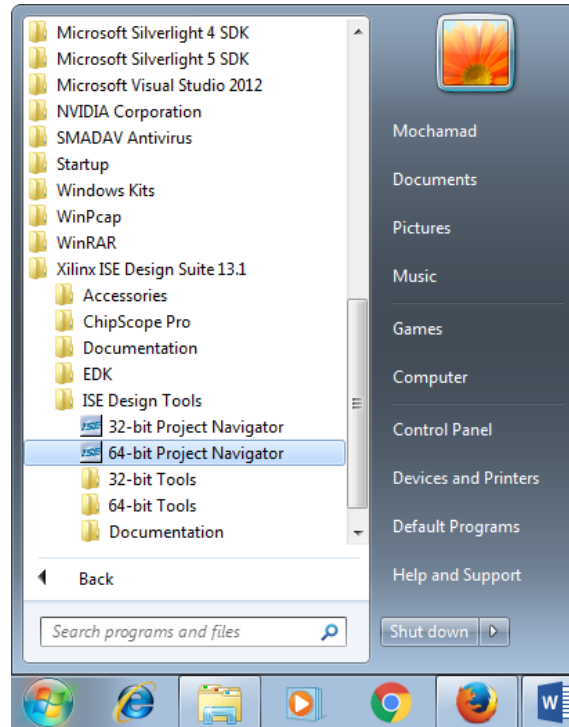


Gambar 1. 7 Modul Xilinx Spartan 3 dengan Downloader JTAG USB

A. Membuat New Project pada ISE Design Suite 13.1

Pada contoh ini, akan dibuat sebuah program yang mengimplementasikan rangkaian AND.

1. Setelah tahap persiapan selesai, buka Start Window di PC Anda -> All Program -> ISE DesignSuit 13.1 kemudian pilih Xilinx ISE Design Tools -> 32/64-bit Project Navigator.



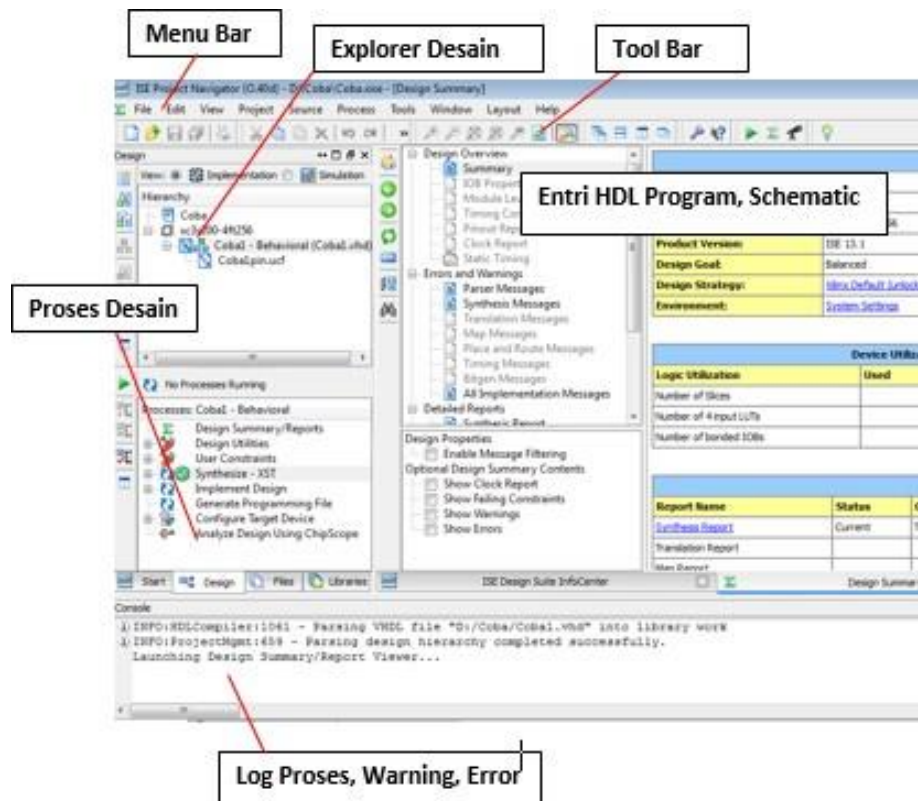
Gambar 1. 8 Start-All Program-Xilinx ISE 13-64bit Project Navigator

2. Setelah dipilih, ISE Project Navigator akan menampilkan Window Project. Pada tampilan ISE Project Navigator tersedia menu-menu seperti :
 - a. Menu Bar
Menu Bar terdiri atas File, Edit, View, Project, Source, Process, Tool, Window, Layout dan Help.
 - b. Tool Bar
Tool Bar adalah komponen-komponen yang disediakan untuk mempercepat kerja user ketika sedang membuat desain. Seperti Tool Save, Open Project, New Project, Cut dll.
 - c. Window Log Proses, Error dan Warning
Adalah window yang digunakan untuk melihat proses synthesise desain "compile" dari program
 - d. Window untuk tempat kita meng-entrykan program atau membuat skematik dari rangkaian.
 - e. Explorer Desain

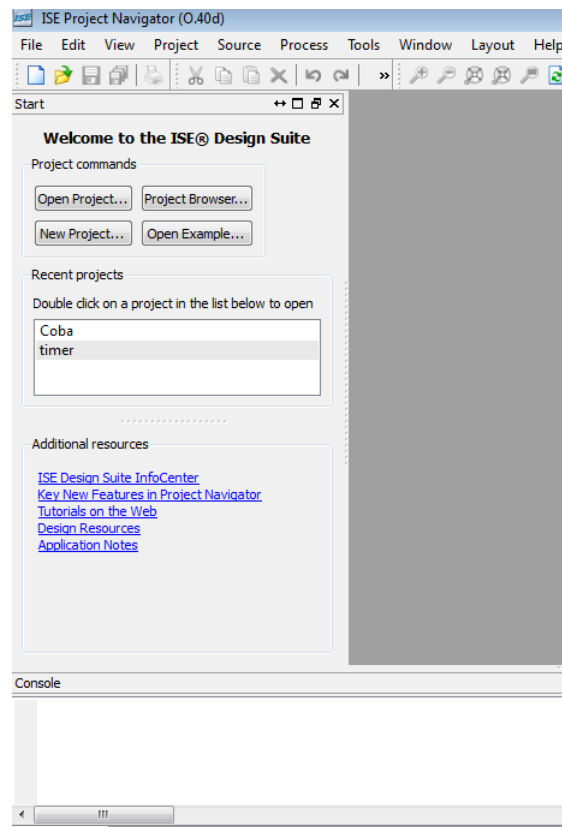
Explorer Desain berisi tentang file-file apa saja yang sedang bekerja/sedang disunakan. Terdiridari file.hdl, .ucf atau file top module.

f. Proses Desain

Adalah window dimana kita men-*synthesize*, mengimplementasikan, men-*generate* suatu file.File tersebut bisa berupa file program .hdl, .ucf atau file skematik.

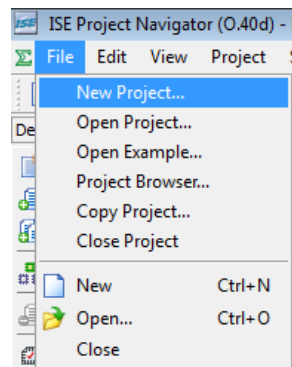


Catatan : Pertama kali membuka biasanya tampilan ISE akan seperti pada gambar 1.9 yang memberitahukan tentang status project yang pernah dibuat sebelumnya (jika sebelumnya pernah membuat sebuah project). Untuk membuat project baru, pilih Menu **File-Close Project** untuk menutup semua file. Sehingga project menjadi kosong.



Gambar 1. 10 Window Project :Close Project

3. Buat project baru dengan klik **File -> new project**



Gambar 1. 11 Membuat Project Baru

- Isikan Nama Project, Lokasi dan Working Direktori anda. Kemudian pilih **HDL** pada Top-Levelsource type. Klik Next.

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: ...

Working Directory: ...

Description:

Select the type of top-level source for the project

Top-level source type:

Gambar 1. 12 Memberi nama project baru

- Pilih **Family= Spartan3, Device=XC3S200, Package = FT256, Speed =-4** (sesuai yang tertera di IC Spartan 3 pada Board). Pilih Preferred Language=VHDL. Kemudian klik Next.

Project Settings

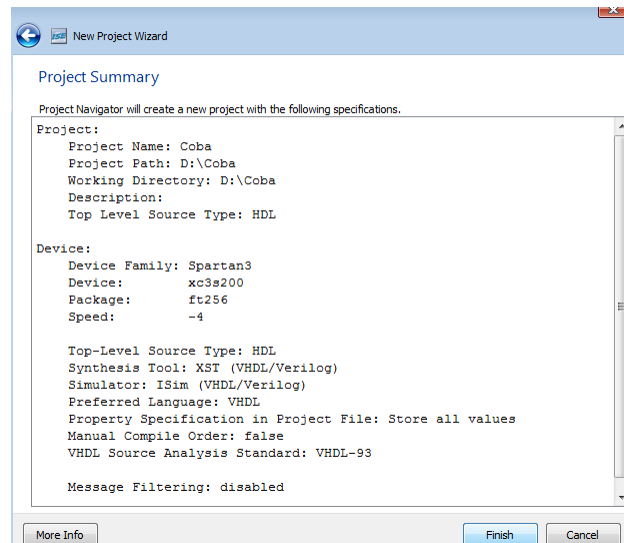
Specify device and project properties.
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3
Device	XC3S200
Package	FT256
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

Gambar 1. 13 Setting Project

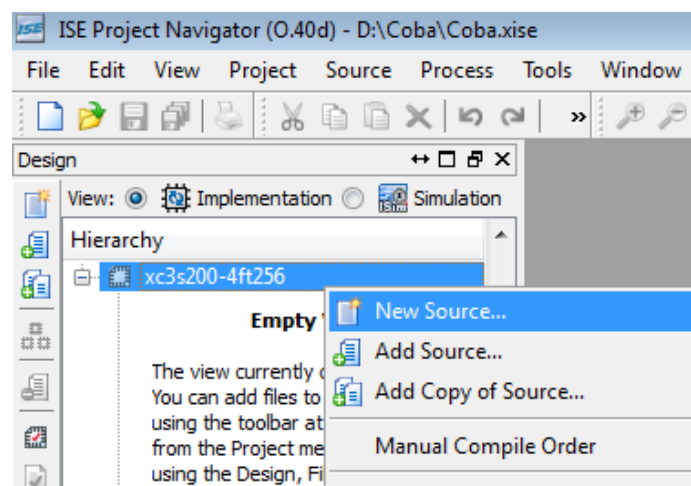
6. Setelah setting project, pilih Finish

Pada bagian ini diberitahukan tentang konfigurasi project yang telah kita buat yang berisi tentang penyimpanan file dan folder file juga tentang model chip IC xilinx yang dipakai. Jika sudah sesuai, kita bisa melanjutkan ke pemrograman dengan memilih **finish**. Jika konfigurasi project belum sesuai dengan apa yang kita inginkan, misalnya salah dalam memilih type IC, kita bisa membatalkan project ini dengan memilih **cancel**.



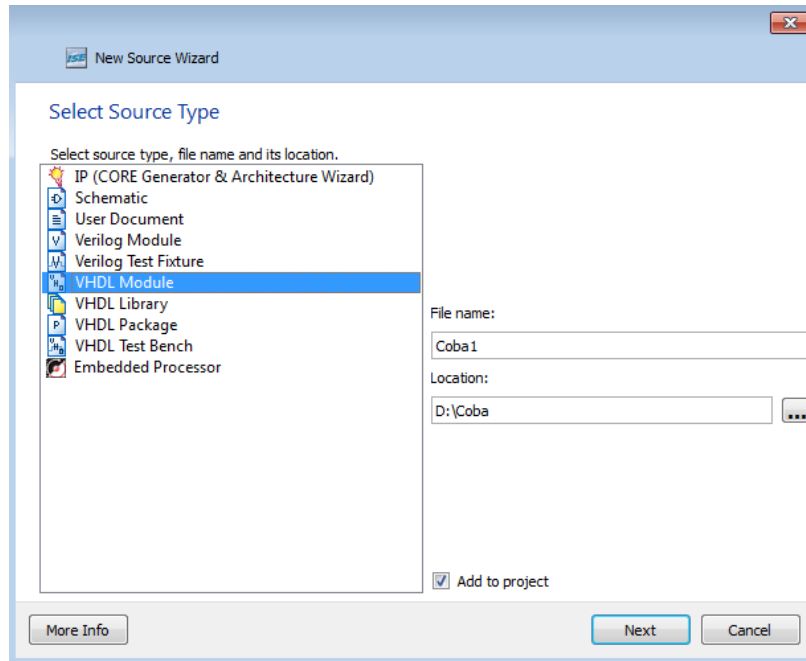
Gambar 1. 14 Selesai membuat project baru

7. Pada kolom hierarchy, klik kanan pada nama project anda kemudian pilih new source untuk membuat file.hdl.



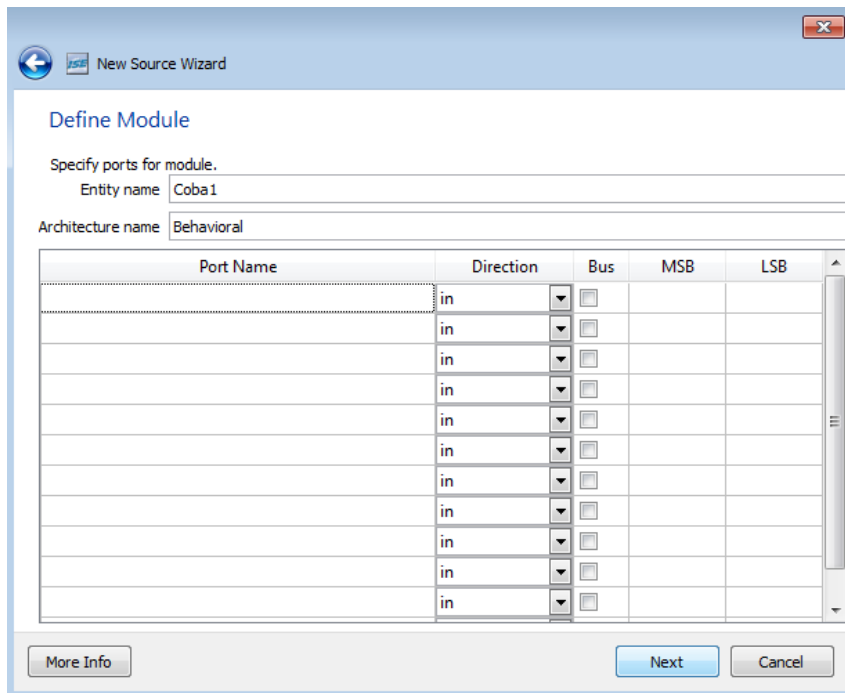
Gambar 1. 15 New Source untuk menambahkan file .hdl

8. Pada New source wizard, pilih VHDL module pada source type, isikan nama file anda, misalkan Coba1. Location adalah lokasi tempat menyimpan file.vhdl tersebut (sudah dalmfolder). Kemudian klik next.



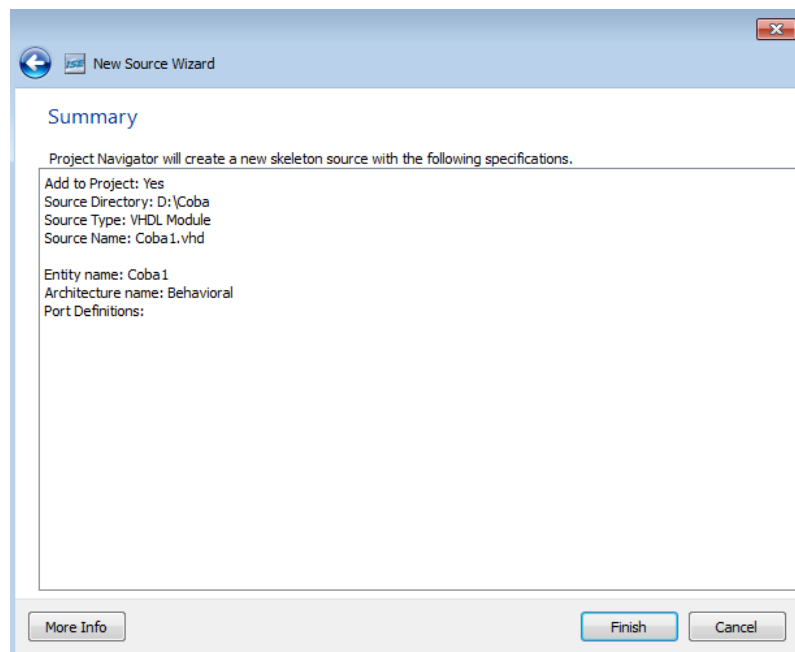
Gambar 1. 16 Membuat file program .vhdl

9. Define module untuk mendefinisikan pin-pin yang digunakan, pada bagian ini langsung dilewati saja. Karena define pin akan kita konfigurasi di akhir. Pilih **next**.



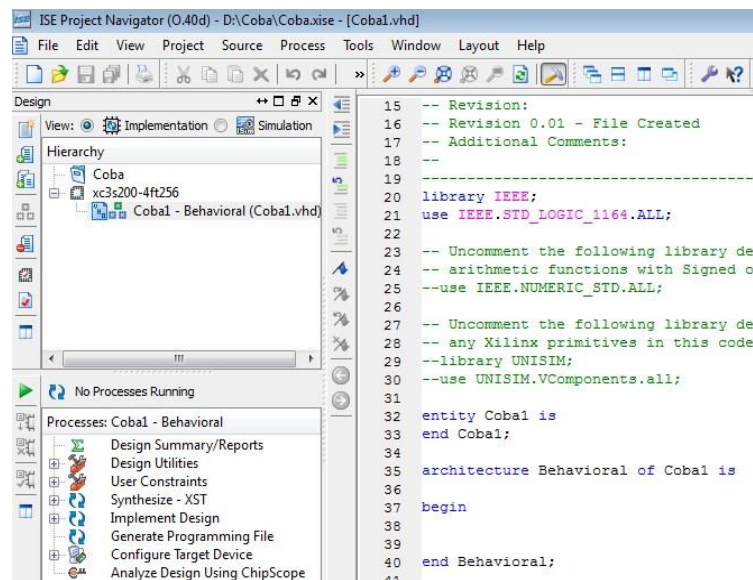
Gambar 1. 17 Define pin

10. Jika konfigurasi sudah selesai, pilih **finish**



Gambar 1. 18 Selesai membuat source wizard

11. Kemudian akan ditampilkan window template coding (Kode program kita akan diletakkandisini).



Gambar 1. 19 Tampilan Template Coding

12. Lengkapi kode program pada gambar 1.17 seperti *script* program 1.1.

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity Coba1 is
port(
    a : in STD_LOGIC;
    b : in STD_LOGIC;

    y : out STD_LOGIC

```

XV

```
);

end Coba1;

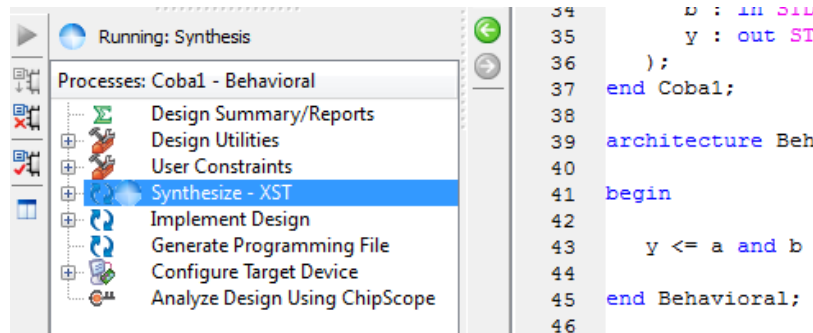
architecture Behavioral of Coba1 is
begin

    y <= a and b

```

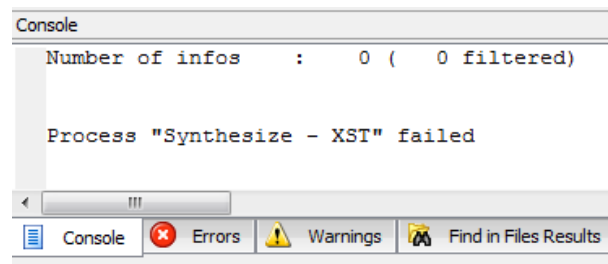
Script 1. 1 Kode program gerbang Logika AND

13. Save project and kemudian **double klik Synthesize-XST** untuk mensintesis program (compileprogram). Jika program anda benar maka akan muncul File Succed jika terdapat kesalahan akan mucul File failed atau Error.



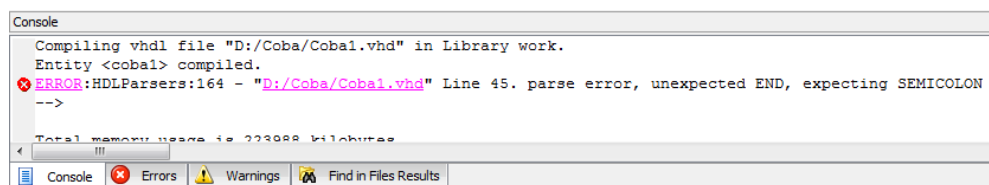
Gambar 1. 20 Proses Synthesize-XST

Ketika selesai mensynthesize program, lihat pada window Log Proses. Jika terdapat Error maka akanmuncul pesan Synthesize –XST seperti pada gambar 1.21.



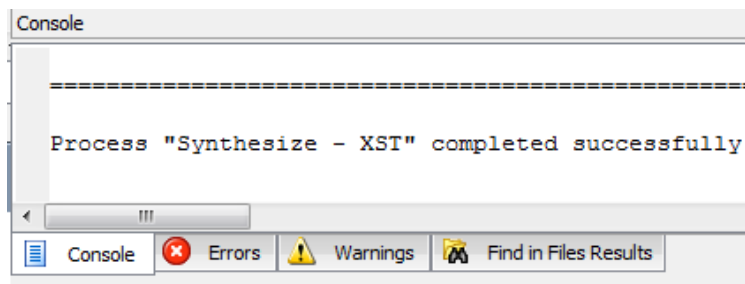
Gambar 1. 21 Proses Synthesize Error

Untuk melihat error, scroll window log process keatas. Kemudian cari bagian keterangan Error. Errorbisa terjadi karena salah penulisan kode program. Dalam keterangan tersebut, juga disebutkan letak baris kode program yang dinyatakan sebagai error. Dari keterangan ini, kita bisa membenarkan program yang salah dengan cepat.



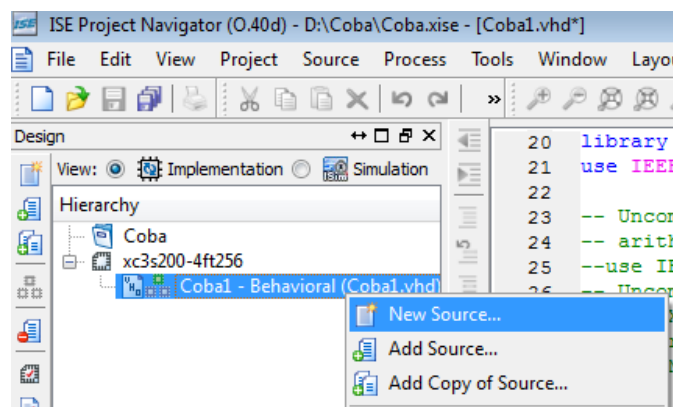
Gambar 1. 22 Pesan Error

Jika program sudah dibenarkan, kemudian lakukan proses Synthesize lagi. Setelh proses synthesizeseselesai, lihat window Log Prosess. Apabila program tidak terdapat kesalahan, maka program akan menampilkan pesan **Synthesize –XST completed successfully**.



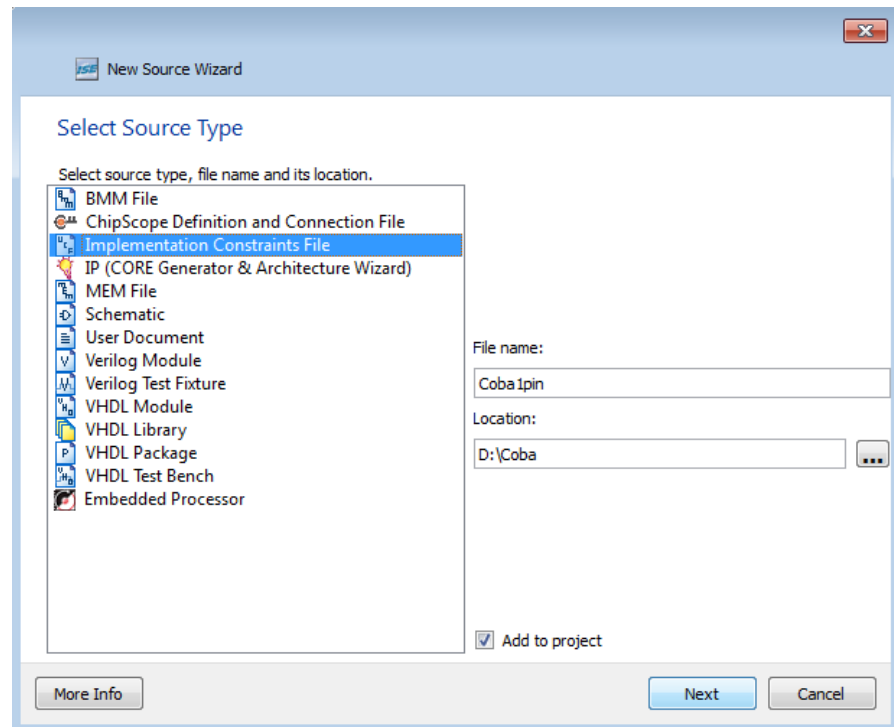
Gambar 1. 23 Synthesize successfully

14. Kemudian klik kanan pada nama file project anda di kolom hierarchy, pilih new source.



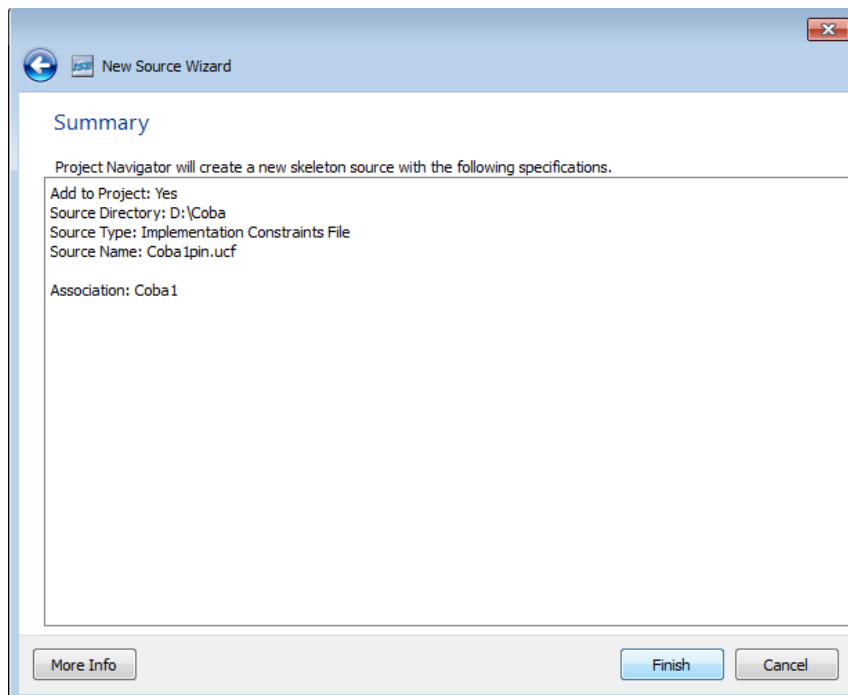
Gambar 1. 24 Menambahkan file baru

15. Pilih Implementasi Constrains File untuk mendefinisikan pin-pin dari program pada Spartan 3. Kemudian isi nama file. (File ini akan berekstensi .ucf)



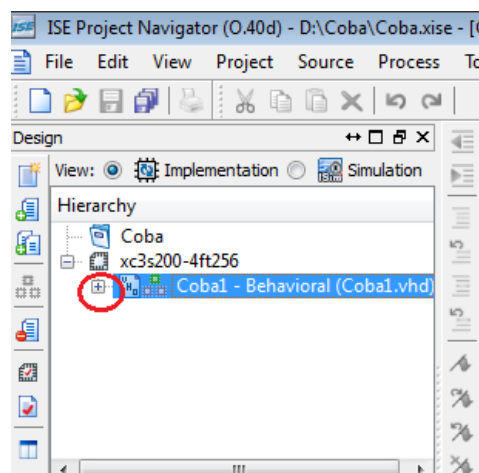
Gambar 1. 25 Membuat source file baru .ucf

16. Setelah selesai, pilih finish



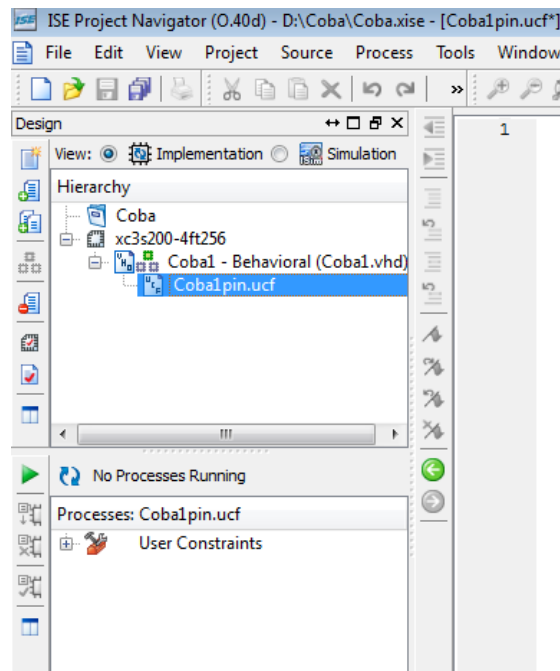
Gambar 1. 26 Selesai membuat project baru .ucf

17. File .ucf yang telah kita buat berada pada sub program .vhd. Untuk membuka klik tandapanah yang berbentuk tanda plus di project.hdl



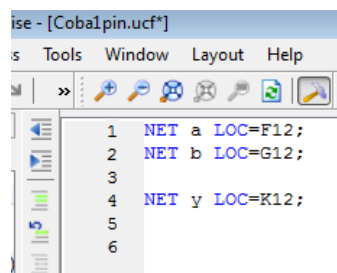
Gambar 1. 27 Letak file .ucf

18. Klik file .ucf yang sudah kita buat, akan muncul template kosong di window sebelah kanan.



Gambar 1. 28 Template untuk mendefinisikan pin

19. Definisikan pin-pin sesuai dengan board Spartan 3. Lihat datasheet pada modul XilinxSpartan3.

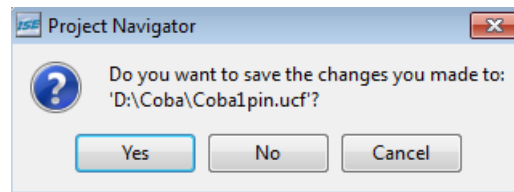


Gambar 1. 29 Penulisan Alamat pin

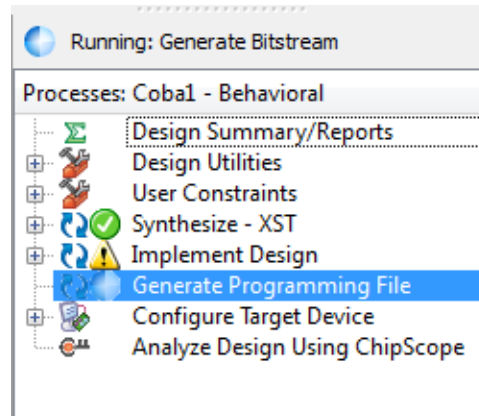
```
NET a LOC=F12;
NET b LOC=G12;
NET y LOC=K12;
```

Script 1. 2 Alamat pin

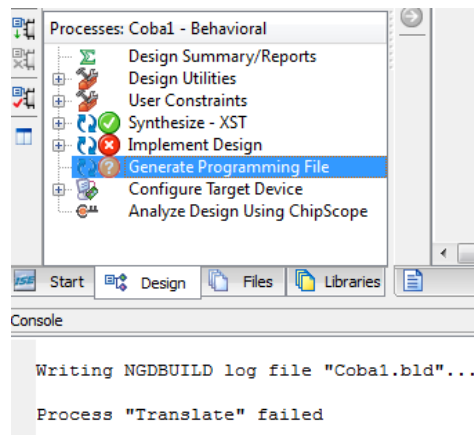
20. Setelah selesai, save project anda. Kemudian klik kembali file project .vhd untuk menampilkan window process desain. Setelah selesai klik **Generate Programming File** pada kolom Processes. Jika centang hijau maka success dan program siap di downloadkan. Jika Adatanda silang merah berarti program/konfigurasi pin ada yang salah.



Gambar 1. 30 Save Project

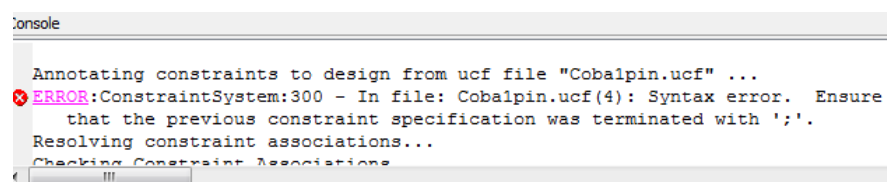


Gambar 1. 31 Process Running Generate Programming File



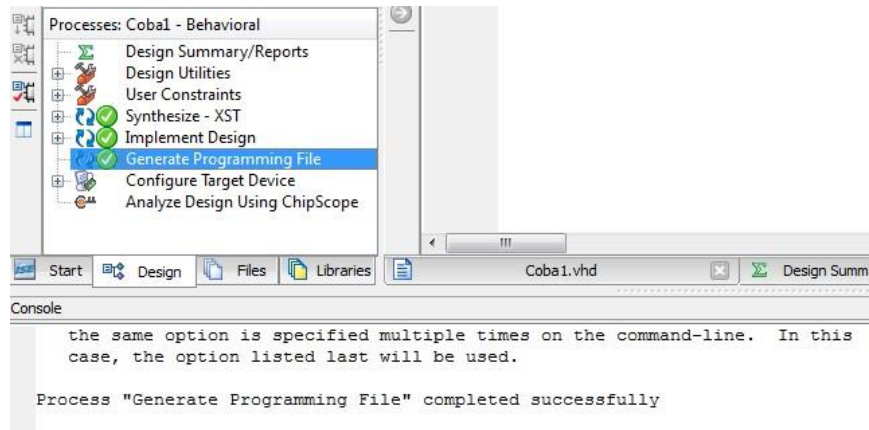
Gambar 1. 32 Program failed

Jika pada process generate file terdapat error, scroll keatas pada window log process. Kemudian lihat error yang terjadi. Error bisa terjadi karena kesalahan sintak dari program.



Gambar 1. 33 Pesan Error

Jika pada process generate file tidak terdapat pesan error, maka program success.



Gambar 1. 34 Process Generate Success

Sampai step ini pembuatan project telah selesai, kemudian file project akan di downloadkan ke IC FPGA Xilinx Spartan 3.

V. Hasil Percobaan

A. Percobaan

Kode program

```
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity test1 is
24 port (
25     a :in STD_LOGIC;
26     b :in STD_LOGIC;
27     y :out STD_LOGIC
28 );
29 end test1;
30
31 architecture Behavioral of test1 is
32 begin
33     y <= a and b;
34 end Behavioral;
35
36
```

Hasil Sintesis program

```
=====
*                               Final Report                               *
=====

Clock Information:
-----
No clock signals found in this design

Asynchronous Control Signals Information:
-----
No asynchronous control signals found in this design

Timing Summary:
-----
Speed Grade: -4

Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 8.957ns

=====

Process "Synthesize - XST" completed successfully
```

Pendefinisian pin - pin

```
1 NET a LOC = F12;
2 NET b LOC = G12;
3 NET y LOC = K12;
```

B. Tugas

Kode program

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity Tugas1 is
24 port(
25     SW0 :in STD_LOGIC;
26     SW1 :in STD_LOGIC;
27     LD0 :out STD_LOGIC;
28     LD1 :out STD_LOGIC;
29     LD2 :out STD_LOGIC;
30     LD3 :out STD_LOGIC;
31     LD4 :out STD_LOGIC);
32 end Tugas1;
33
34 architecture Behavioral of Tugas1 is
35 begin
36
37     LD0 <= SW0 and SW1;
38     LD1 <= SW0 or SW1;
39     LD2 <= not SW0;
40     LD3 <= SW0 nand SW1;
41     LD4 <= SW0 nor SW1;
42
43
44
45 end Behavioral;
46
```

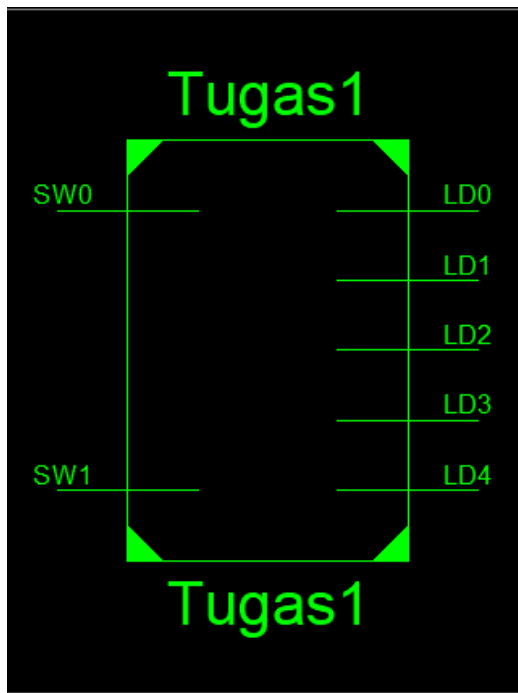

Pendefinisian pin – pin

```
1 NET SW0 LOC=F12;  
2 NET SW1 LOC=G12;  
3  
4  
5 NET LD0 LOC=K12;  
6 NET LD1 LOC=P14;  
7 NET LD2 LOC=L12;  
8 NET LD3 LOC=N14;  
9 NET LD4 LOC=P13;
```

Hasil Sintesis Program

```
Console  
-----  
*                               Final Report                               *  
-----  
  
Clock Information:  
-----  
No clock signals found in this design  
  
Asynchronous Control Signals Information:  
-----  
No asynchronous control signals found in this design  
  
Timing Summary:  
-----  
Speed Grade: -4  
  
Minimum period: No path found  
Minimum input arrival time before clock: No path found  
Maximum output required time after clock: No path found  
Maximum combinational path delay: 9.077ns  
  
-----  
  
Process "Synthesize - XST" completed successfully  
  
< [Progress Bar]
```

Tampilan Schematic Program



VI. Analisa dan Kesimpulan

No. _____

Date: _____

<input checked="" type="checkbox"/>	Analisa
<input type="checkbox"/>	Pada praktikum tersebut telah dilakukan percobaan memprogram
<input type="checkbox"/>	dengan menggunakan Xilinx. Untuk bahasa yang digunakan, algoritmanya
<input type="checkbox"/>	hampir sama dengan bahasa pemrograman lain, hanya berbeda pada syntax
<input type="checkbox"/>	yang digunakan. Seperti contoh syntax $y \leq a \text{ and } b$, yang berarti
<input type="checkbox"/>	bahwa variabel "y" merupakan output yang dilaju karena di tunjuk oleh
<input type="checkbox"/>	syntax " \leq ", lalu variabel a dan b merupakan inputan program yang
<input type="checkbox"/>	telah di deklarasikan dengan syntax Logic gate yaitu "and". Dikarenakan
<input type="checkbox"/>	percobaan tersebut menggunakan board dari Spartan 3, maka pendefinisian
<input type="checkbox"/>	dari pin yang akan digunakan menyesuaikan dengan datasheet Spartan 3.
<input type="checkbox"/>	Dimana pada syntax "Net SW0 LOC = F12" yang berarti variabel
<input type="checkbox"/>	SW0 (switch 0) berada pada "LOC" (lokasi) FPGA Pin "F12".
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	Kesimpulan
<input type="checkbox"/>	Berdasarkan praktikum tersebut dapat disimpulkan bahwa:
<input type="checkbox"/>	→ Bahasa pemrograman yang digunakan pada IDE Xilinx hampir sama dengan
<input type="checkbox"/>	bahasa pemrograman yang lain, hanya berbeda syntax.
<input type="checkbox"/>	→ Ketika terjadi error pada saat mensintesis program, pada bagian
<input type="checkbox"/>	console telah diberitahu letak errornya.
<input type="checkbox"/>	→ Untuk pendefinisian pin dapat melihat datasheet dari board yang
<input type="checkbox"/>	digunakan
<input type="checkbox"/>	
<input type="checkbox"/>	