

**1 D4 - TEKKOM B**



**Pengenalan FPGA dan ISE Design Tool Xilinx**



Nama : Septian Bagus Jumantoro

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

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

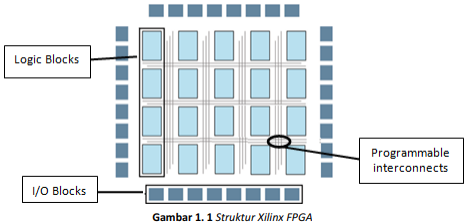
*Field Programmable Gate Arrays (FPGA)* adalah programmable device besar yang tersusun atas modul-modul logika independen yang dapat dikonfigurasi oleh pemakai yang di hubungkan melalui kanal-kanal routing yang dapat di program.

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).



Proses implementasi pada FPGA :

1. FPGAs diimplementasikan dengan *efficient CAD systems*
2. Design Entry dapat dilakukan dengan cara
3. Schematic capture program,
4. VHDL / Verilog

3. Logic Optimization

1. Memodifikasi Boolean Expression untuk mengopyimalkan penggunaan
2. Logic berbasis area dan kecepatan atau kombinasi keduanya

4. Technologi Mapping

1. Transformasi dari Boolean Expression atau VHDL kedalam bentuk circuit pada FPGA logic blocks
2. Area optimization – meminimalkan penggunaan block
3. Delay Optimixation – meminimalkan jumlah stage pada critical path

5. Placement

1. Setelah memetakan rangkaian ke dalam blok logika, selanjutnya algoritma penempatan digunakan untuk meletakan masing-masing blok ke dalam FPGA array.
2. Meminimalkan total panjang interkoneksi yang diperlukan untuk penempatan blok logika

6. Routing Software

1. Mengatur FPGA’s wire segment dan menentukan programmable switches untuk menghubungkan antar FPGA blocks
2. Memastikan 100% koneksi telah terbentuk
3. Meminimalkan propagation delay pada *time-critical connections.*

7. Programming unit

1. Mengkonfigurasi chip setelah placement dan routing step tuntas
2. Keseluruhan proses memakan waktu antara beberapa menit sampai beberapa jam

Secara umum proses kerja FPGA adalah sebagai berikut :

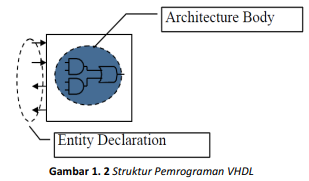
1. **Design Entry** dengan cara schematic, ABEL, VHDL, dan Verilog
2. **Implementation** yaitu Placement dan Routing serta bitsream generation. Juga analyze timing, view layout
3. **Download** meng-upload bitstream ke Hardware (FPGA chip)

**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 sesuia dengan gerbang logica 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, compiller 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.

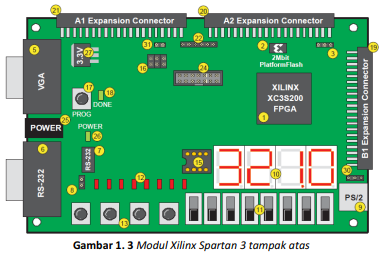
*VHDL* adalah bahasa pemrograman yang digunakan untuk mendeskripsikan *logic circuit* yang dikehendaki. Secara umum struktur dari pemrograman *VHDL* terdiri atas dua bagian **ENTITY** dan **ARCHITECTURE.**



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.

1. **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.



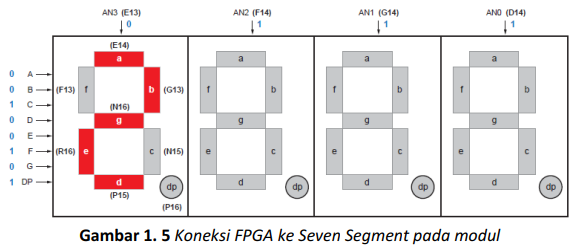


Secara ringkas fitur-fitur yang dimiliki oleh Xilinx Spartan 3 adalah :

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 chanel avaible 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 cryctal 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 Dekstop 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 Sevent 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.



Untuk tabel pin FPGA dengan seven segment ditunjukan 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.

1. **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* desin, 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.



1. **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
5. **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).

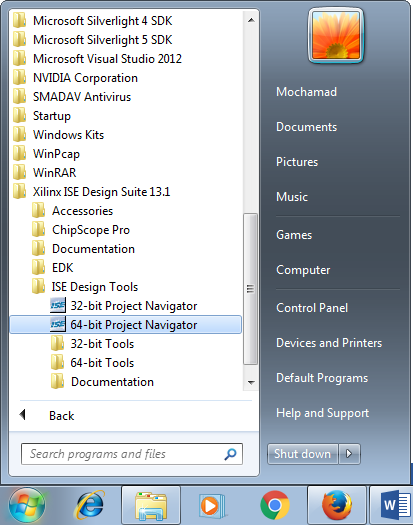


**Gambar 1. 7** *Modul Xilinx Spartan 3 dengan Downloader JTAG USB*

1. **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 Design Suit 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*

* 1. Setelah dipilih, ISE Project Navigator akan menampilkan Window Project. Pada tampilan ISE Project Navigator tersedia menu‐menu seperti :
     1. Menu Bar

Menu Bar terdiri atas File, Edit, View, Project, Source, Process, Tool, Window, Layout dan Help.

* + 1. 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.

* + 1. Window Log Proses, Error dan Warning

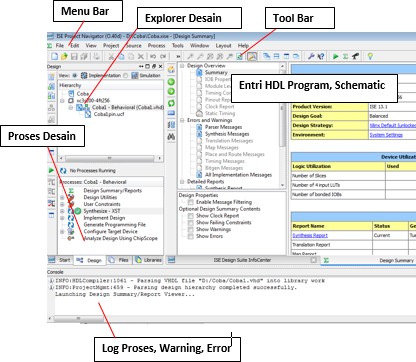
Adalah window yang digunakan untuk melihat proses synthesize desain “compile” dari program

* + 1. Window untuk tempat kita meng‐entrykan program atau membuat skematik dari rangkaian.
    2. Explorer Desain

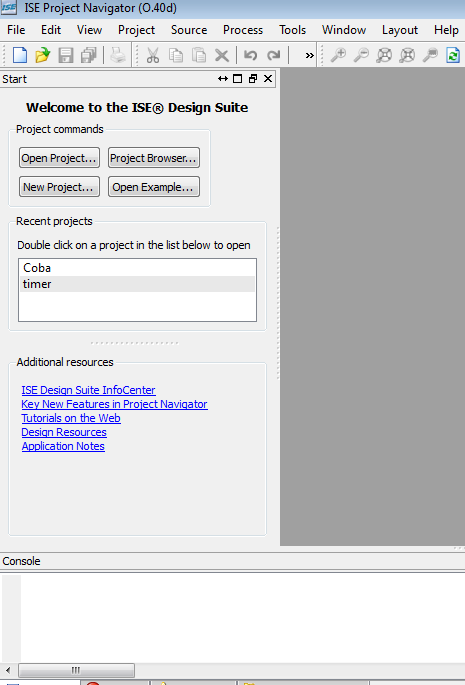
Explorer Desain berisi tentang file‐file apa saja yang sedang bekerja/sedang disunakan. Terdiri dari file.hdl, .ucf atau file top module.

* + 1. Proses Desain

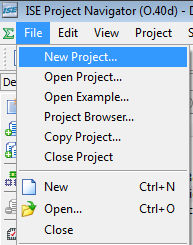
Adalah window dimana kita men‐*synthesize*, mengimplementasikan, men‐*generate* suatu file. File tersebut bisa berupa file program .hdl, .ucf atau file skematik.



*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.*

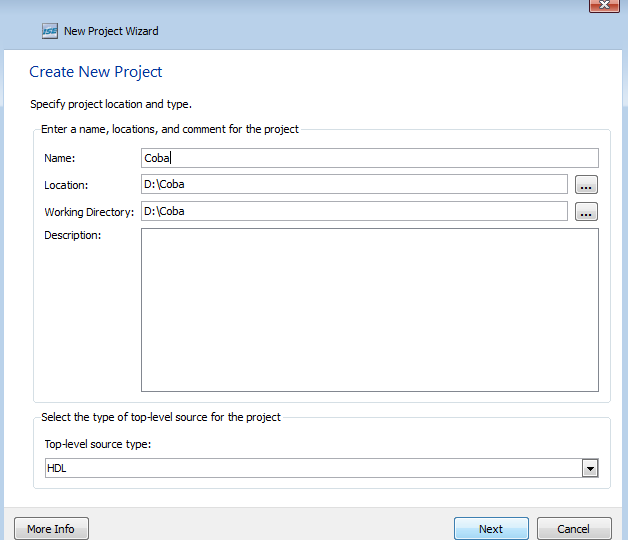


**Gambar 1. 10** *Window Project :Close Project*

* 1. Buat project baru dengan klik ***File ‐> new project***

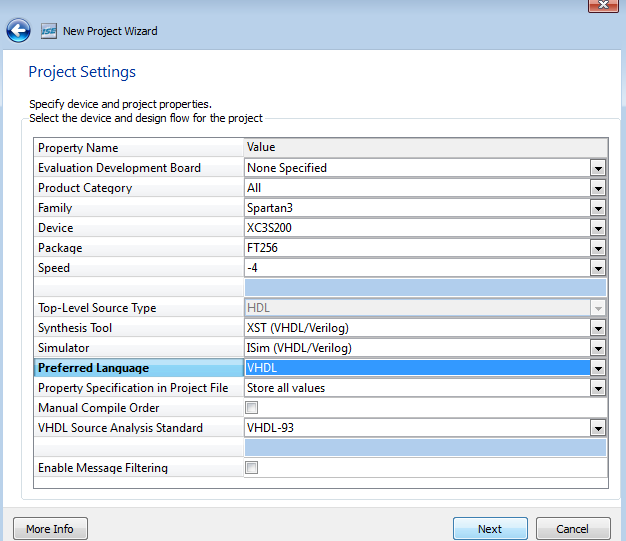
**Gambar 1. 11** *Membuat Project Baru*

* 1. Isikan Nama Project, Lokasi dan Working Direktori anda. Kemudian pilih **HDL** pada Top‐Level source type. Klik Next.



**Gambar 1. 12** *Memberi nama project baru*

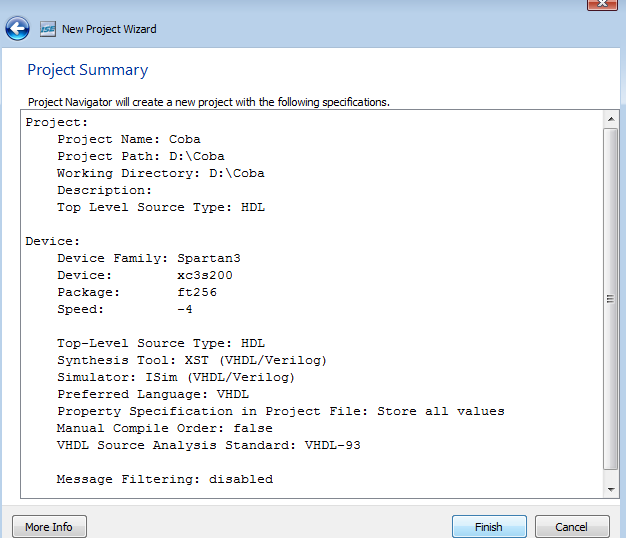
* 1. Pilih **Family= Spartan3, Device=XC3S200, Package = FT256, Speed =‐4** (sesuai yang tertera di IC Spartan 3 pada Board). Pilih Preferred Language=VHDL. Kemudian klik Next.



**Gambar 1. 13** *Setting Project*

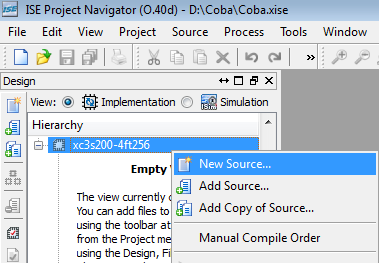
* 1. Setelah setting project, pilih Finish

Pada bagian ini diberitahukan tentang konfxiigiurasi‐konfigurasi project yang telah kita buatyang berisi tentng 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.**

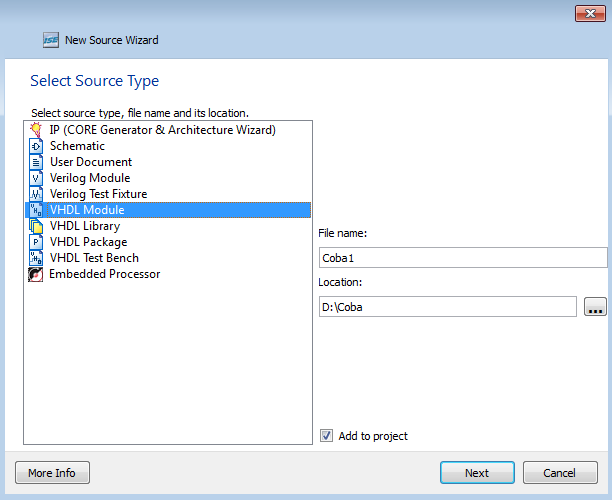


**Gambar 1. 14 *Selesai membuat project baru***

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

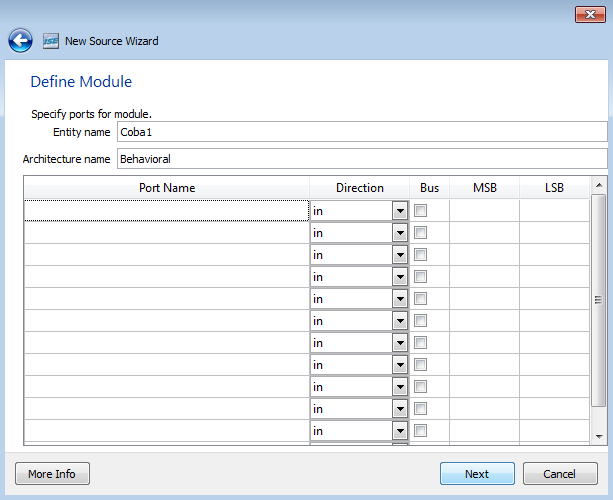


**Gambar 1. 15 *New Source untuk menambahkan file .hdl***

* 1. 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 dalm folder). Kemudian klik next. 

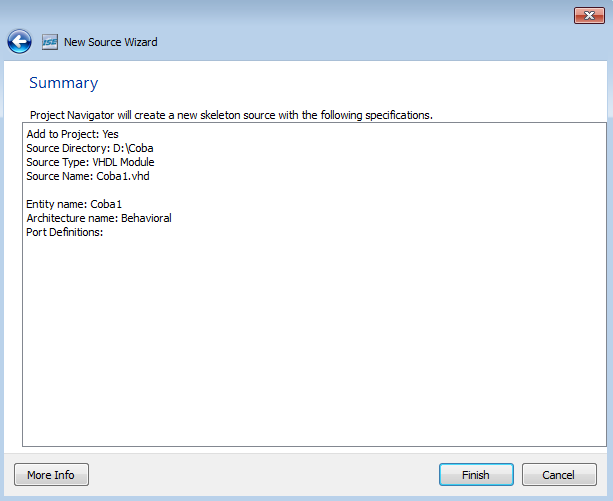
**Gambar 1. 16 *Membuat file program .vhdl***

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



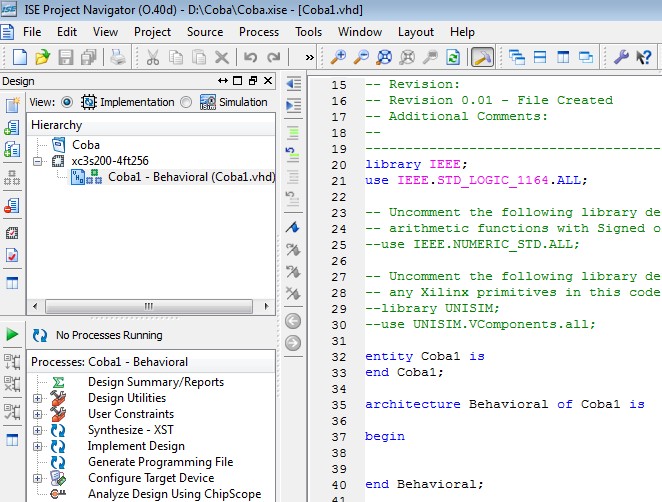
**Gambar 1. 17 *Define pin***

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



**Gambar 1. 18 *Selesai membuat source wizard***

* 1. Kemudian akan ditampilkan window template coding (Kode program kita akan diletakkan disini).



**Gambar 1. 19 *Tampilan Template Coding***

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

**);**

**end** Coba1**;**

**architecture** Behavi or al **of** Coba1 **is begin**

y **<=** a and b

**end** Behavi or al **;**

**library**I EEE**;**

**use**I EEE**.**STD\_ LOGI C\_ 1164**.ALL;**

**entity** Coba1 **is port(**

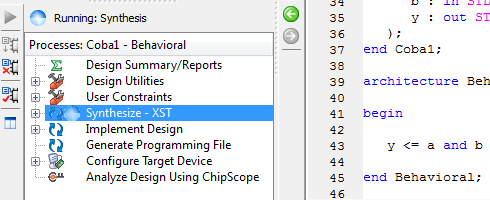
a **:in**STD\_ LOGI C**;** b **:in**STD\_ LOGI C**;**

y **:out**STD\_ LOGI C

xv

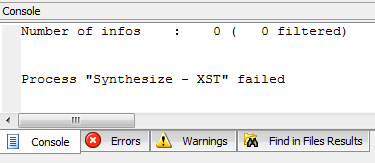
**Script 1. 1 *Kode program gerbang Logika AND***

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



**Gambar 1. 20 *Proses Synthesize‐XST***

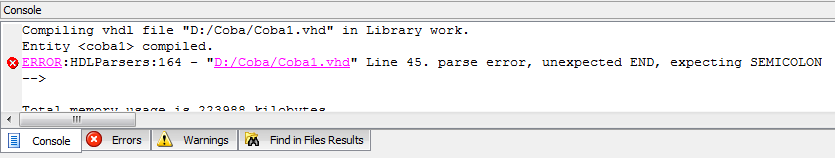
Ketika selesai mensynthesize program, lihat pada window Log Proses. Jika terdapat Error maka akan muncul pesan Synthesize –XST seperti pada gambar 1.21.



**Gambar 1. 21 *Proses Synthesize Error***

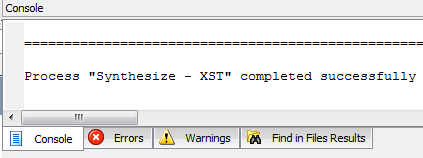
Untuk melihat error, scroll window log process keatas. Kemudian cari bagian keterangan Error. Error bisa terjdi karena salah penulisan kode program. Dalam keterangan tersebut, juga disebutkan letak baris kode program yang dinyatakan sebagai error. Dari keterangan ini, kita bisa membenarkan

program yang salah dengan cepat.



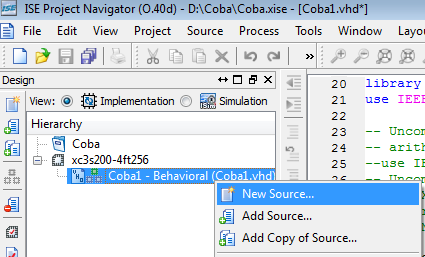
**Gambar 1. 22 *Pesan Error***

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



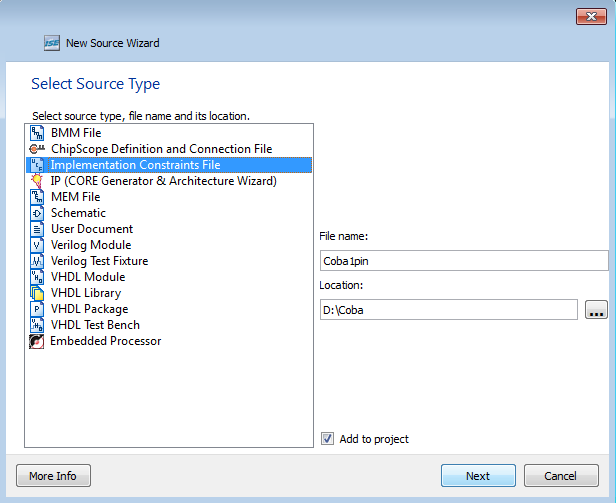
**Gambar 1. 23 *Synthesize successfully***

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



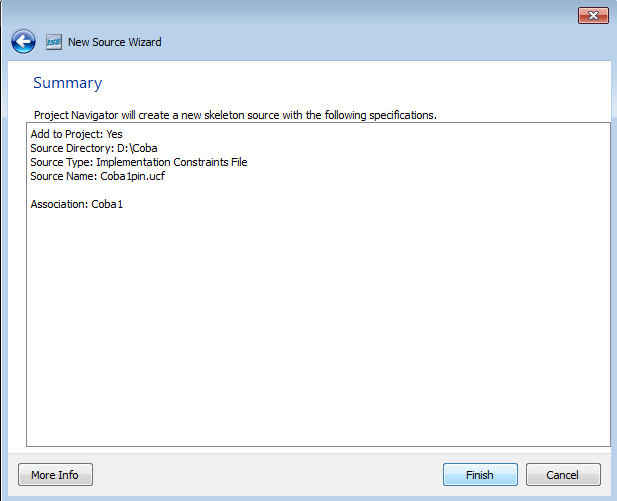
**Gambar 1. 24 *Menambahkan file baru***

* 1. Pilih Implementasi Constrain File untuk mendefinisikan pin‐pin dari program pada Spartan 3. Kemudian isi nama file. (File ini akan berekstensi .ucf)

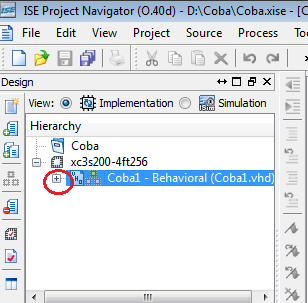


**Gambar 1. 25 *Membuat source file baru .ucf***

* 1. Setelah selesai, pilih **finish**

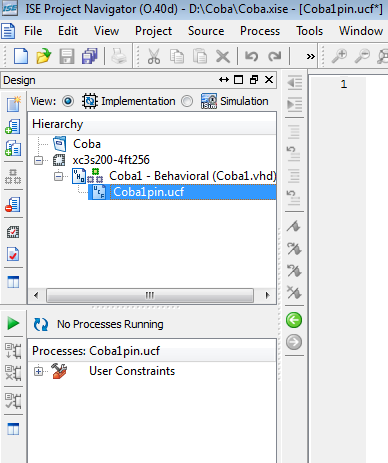


**Gambar 1. 26 *Selesai membuat project baru .ucf***

* 1. File .ucf yang telah kita buat berada pada sub program .vhdl. Untuk membuka klik tanda panah yang berbentuk tanda plus di project.hdl

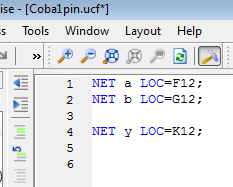
**Gambar 1. 27 *Letak file .ucf***

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



**Gambar 1. 28 *Template untuk mendefiniskan pin***

* 1. Definisikan pin‐pin sesuai dengan board Spartan 3. Lihat datasheet pada modul Xilinx Spartan3.

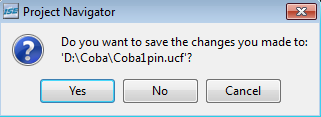


**Gambar 1. 29 *Penulisan Alamat pin***

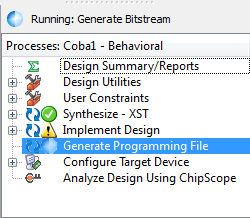
NET a LOC=F12; NET b LOC=G12; NET y LOC=K12;

**Script 1. 2 *Alamat pin***

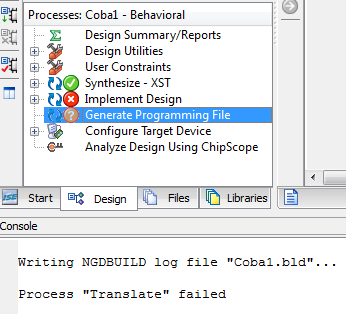
* 1. Setelah selesai, save project anda. Kemudian klik kembali file project .vhdl untuk menampilkan window process desain. Setelah selesai klik **Generate Programming File** pada kolom Processes. Jika centang hijau maka success dan program siap di downloadkan. Jika Ada tanda silang merah berarti program/konfigurasi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



**Gambar 1. 30 *Save Project***

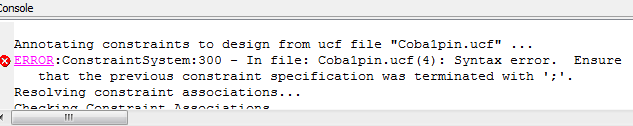


**Gambar 1. 31 Process Running Generate Programming File**

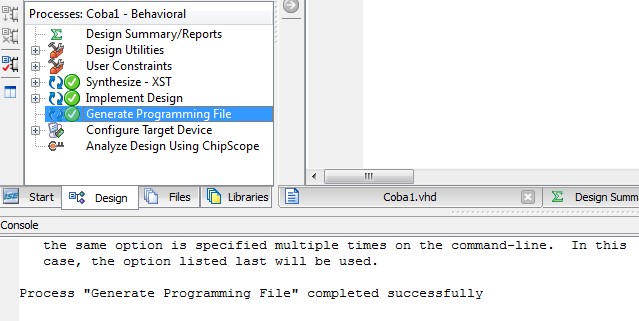


**Gambar 1. 32 *Program failed***

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.



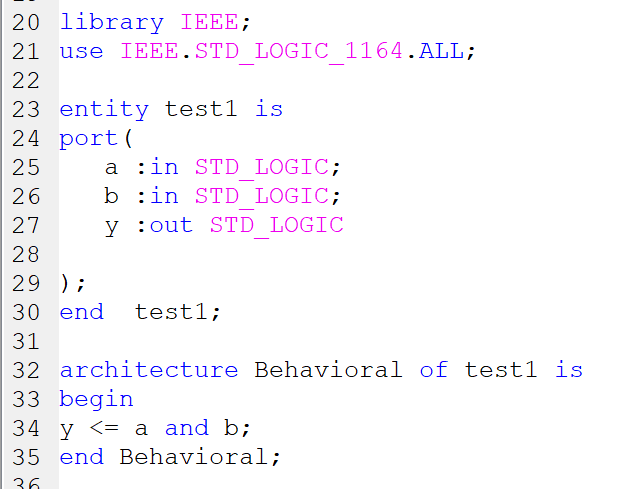
**Gambar 1. 33 Pesan Error**

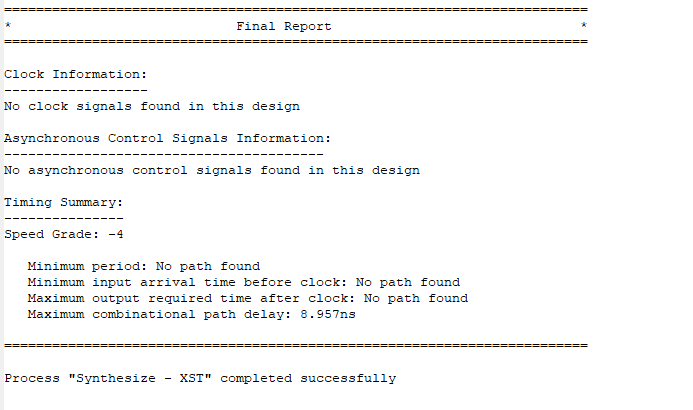
Jika pada process generate file tidak terdapat pesan error, maka program success.

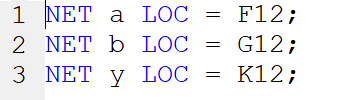
**Gambar 1. 34 *Process Generate Success***

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

1. **Hasil Percobaan**
2. **Percobaan**

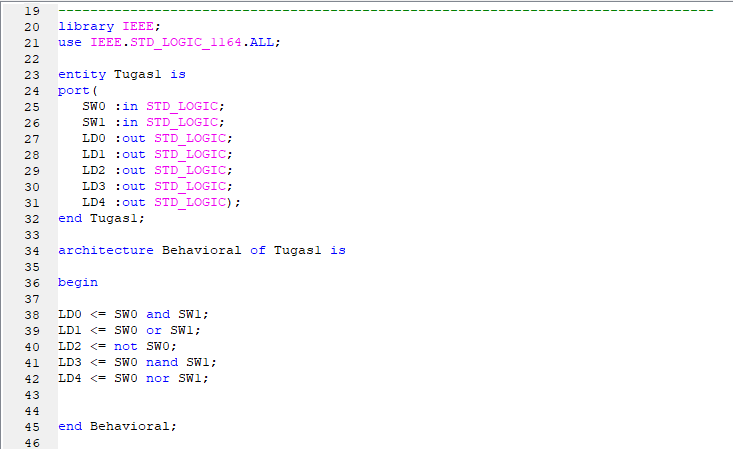
Kode program

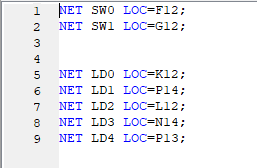
Hasil Sintesis program

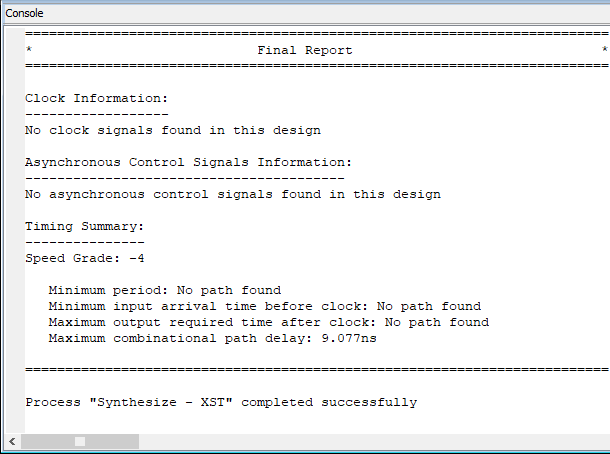
Pendefinisian pin – pin

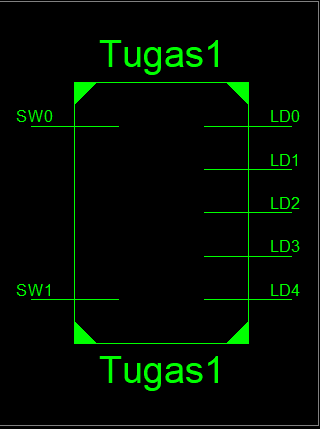
1. **Tugas**

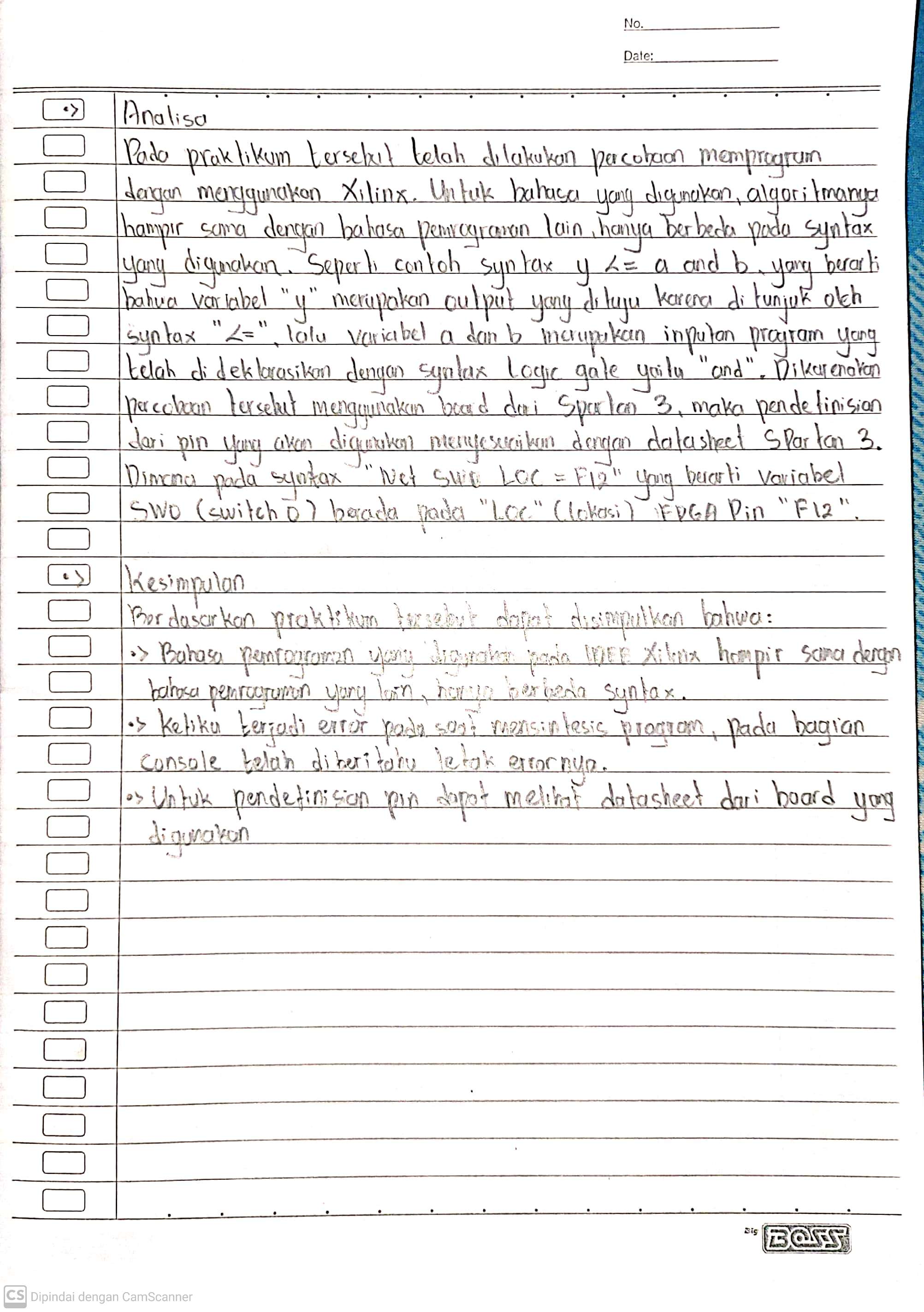
Kode program



****Pendefinisian pin – pin

**** Hasil Sintesis Program

Tampilan Schematic Program

1. **Analisa dan Kesimpulan**