MODUL 11

IMPLEMENTASI RANGKAIAN DIGITAL MESIN KEADAAN BERHINGGA (FINITE STATE MACHINE) DI FPGA MENGGUNAKAN VERILOG-HDL

1. Tujuan Praktikum

Setelah mempraktekan topik ini, praktikan diharapkan dapat :

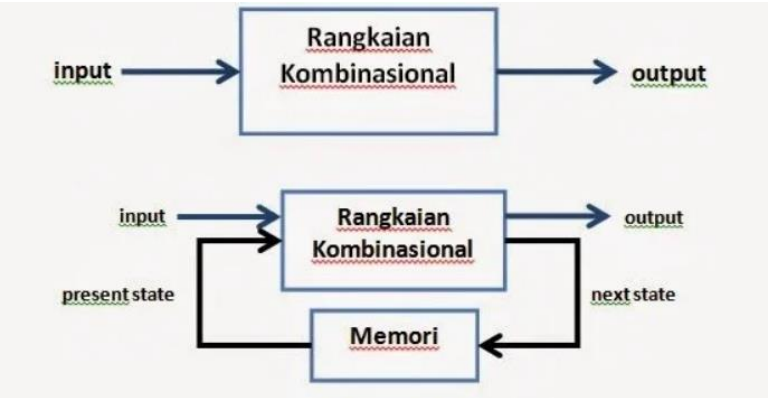
- Praktikkan mampu memahami konsep dasar mesin keadaan berhingga (Finite State Machine - FSM) dalam desain rangkaian digital.
- Praktikkan mampu menggunakan bahasa Verilog untuk mendesain dan mengimplementasikan FSM.
- Praktikkan mampu memahami dan mengimplementasikan kode Verilog pada platform FPGA DE10-Lite.

2. Dasar Teori

2.1 Rangkaian Kombinasional dan Rangkaian Sekuensial

Rangkaian kombinasional terdiri dari gerbang logika yang memiliki output yang selalu tergantung pada kombinasi input yang ada. Rangkaian kombinasional melakukan operasi yang dapat ditentukan secara logika dengan memakai sebuah fungsi boolean.

Rangkaian sekuensial merupakan rangkaian logika yang keadaan outputnya tergantung pada keadaan input-inputnya juga tergantung pada keadaan output sebelumnya. Rangkaian ini juga didefenisikan sebagai rangkaian logika yang outputnya tergantung waktu.



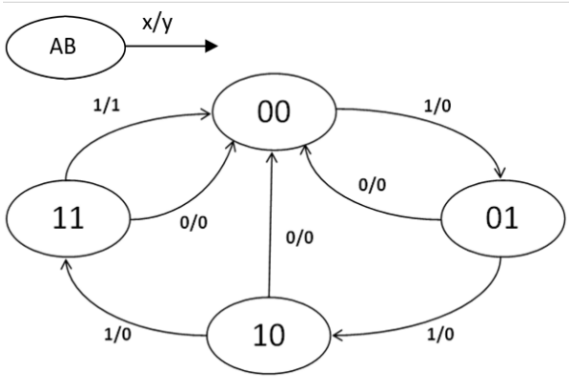
gambar 1 rangakaian kombinasional dan rangkaian sekuensial

2.2 Mesin Keadaan Berhingga (Finite State Machine - FSM)

Mesin Keadaan Berhingga (FSM) adalah model komputasi yang digunakan untuk merancang logika sekuensial. Dalam sebuah FSM, sistem memiliki sejumlah keadaan (states) yang berbeda, dan transisi (transitions) antara keadaan-keadaan ini terjadi berdasarkan input yang diterima.

Dalam konteks praktis, FSM digunakan untuk mendesain dan mengimplementasikan sistem yang memiliki perilaku yang terstruktur berdasarkan kondisi-kondisi tertentu. Misalnya, dalam sebuah mesin penjual otomatis, FSM dapat digunakan untuk mengatur keadaan-keadaan seperti "siap", "menerima koin", "memproses pesanan", dan seterusnya, serta transisi antara keadaan-keadaan ini berdasarkan input seperti koin yang dimasukkan atau tombol yang ditekan Adapun dua contoh dari FSM sebagai berikut :

- FSM Mealy: Output tergantung pada keadaan saat ini dan input. Ini berarti output dapat berubah segera setelah input berubah, tanpa harus menunggu transisi state.
- FSM Moore: Output hanya tergantung pada keadaan saat ini. Output hanya berubah ketika FSM masuk ke state baru.



t			t+1		
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1

gambar 2. FSM ditampilkan sebagai diagram keadaan, dan sebagai tabel keadaan. Legenda di kiri atas menunjukkan variabel keadaan A dan B, serta masukan x dan keluaran y.

3. Lembar Kegiatan Praktikum

3.1 Alat dan Bahan

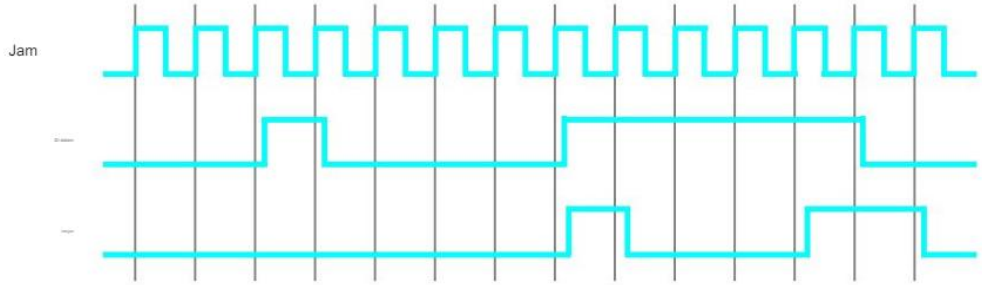
- Software Quartus II
- ModelSim
- Mouse

- Laptop

3.2 Langkah Praktikum Modul 10

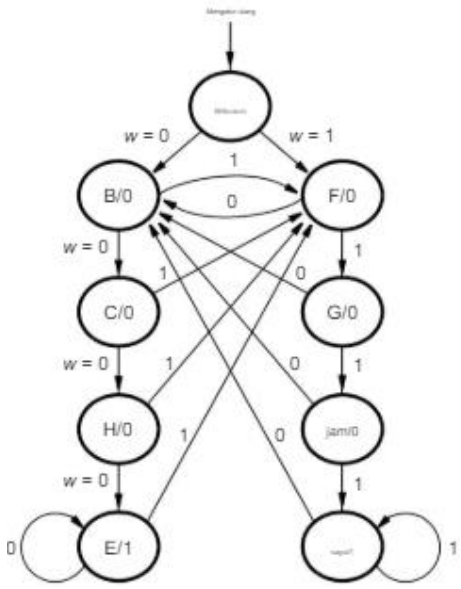
Bagian I

Kami ingin menerapkan mesin keadaan terbatas (FSM) yang mengenali dua urutan spesifik simbol masukan yang diterapkan, yaitu empat angka 1 berturut-turut atau empat angka 0 berturut-turut. Ada masukan w dan keluaran z. Kapanpun w = 1 atau w = 0 untuk empat pulsa clock berturut-turut, nilai z harus 1; jika tidak, z = 0. Urutan yang tumpang tindih diperbolehkan, sehingga jika w = 1 untuk lima pulsa clock berturut- turut, output z akan sama dengan 1 setelah pulsa keempat dan kelima. Gambar 1 mengilustrasikan hubungan yang diperlukan antara w dan z.



Gambar 1: Waktu yang diperlukan untuk keluaran z.

Diagram keadaan untuk FSM ini ditunjukkan pada Gambar 2. Untuk bagian ini Anda harus secara manual mendapatkan rangkaian FSM yang mengimplementasikan diagram keadaan ini, termasuk ekspresi logika yang memberi makan setiap keadaan flip-flop. Untuk



Gambar 2: Diagram keadaan untuk FSM.

Nama	Kode Negara y8y7y6y5y4y3y2y1y0
A	000000001
B	000000010
C	000000100
D	000001000
[illegible]	000010000
F	000100000
G	001000000
H	010000000
[illegible]	100000000

Tabel 1: Kode one-hot untuk FSM.

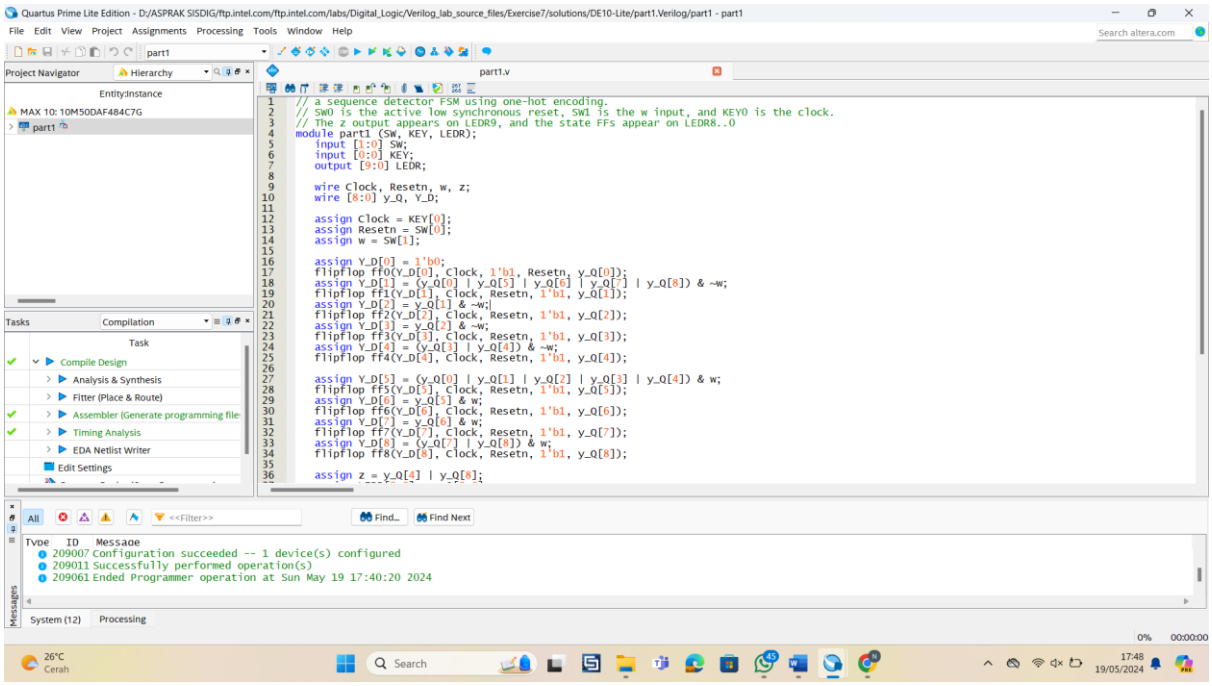
mengimplementasikan FSM gunakan sembilan state flip-flop yang disebut y8,…,y0 dan penetapan status one-hot yang diberikan pada Tabel 1.

1. Buat proyek Quartus® baru untuk sirkuit FSM.
2. Tulis file Verilog yang membuat instance sembilan flip-flop dalam rangkaian dan yang menentukan ekspresi logika yang menggerakkan port input flip-flop. Gunakan hanya pernyataan penetapan sederhana dalam kode Verilog Anda untuk menentukan logika yang memberi makan sandal jepit. Perhatikan bahwa kode one-hot memungkinkan Anda memperoleh ekspresi ini melalui inspeksi.

Gunakan sakelar sakelar SW0 sebagai input reset sinkron aktif-rendah untuk FSM, gunakan SW1 sebagai input w, dan tombol tekan KEY0 sebagai input jam yang diterapkan secara manual. Gunakan LEDR9 lampu merah sebagai keluaran z, dan tetapkan keluaran flip-flop status ke LEDR8 lampu merah ke LEDR0.

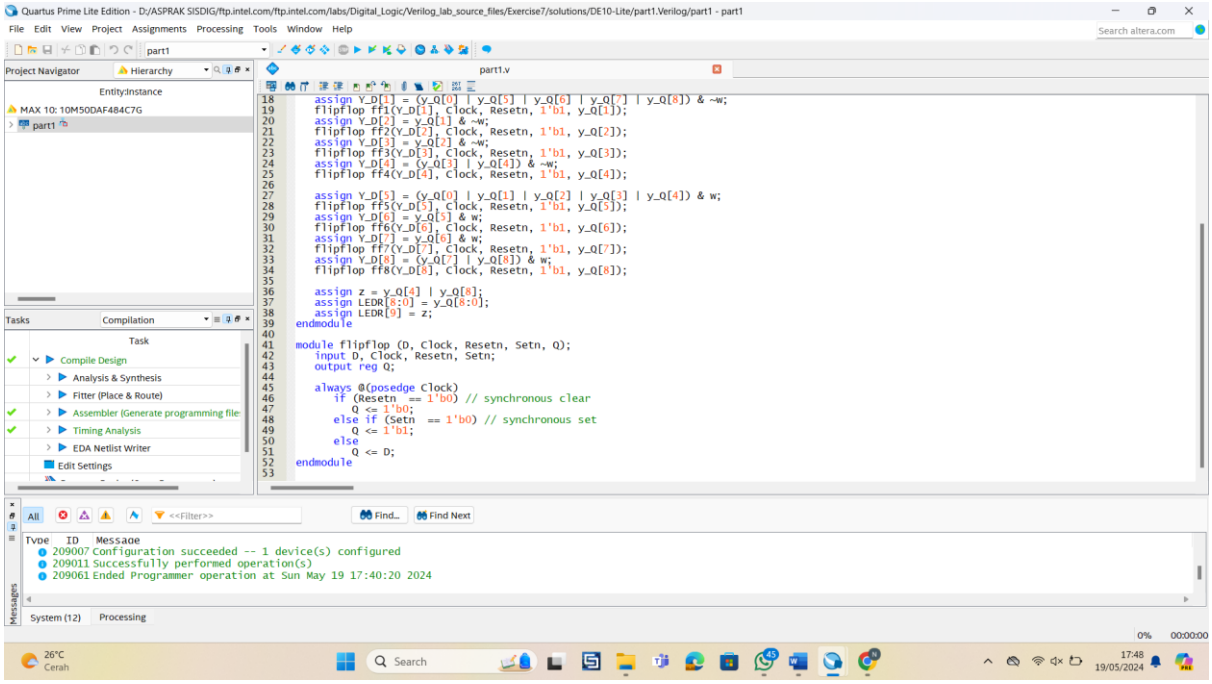
3. Sertakan file Verilog dalam proyek Anda, dan tetapkan pin pada FPGA untuk terhubung ke sakelar dan LED.
4. Simulasikan perilaku sirkuit Anda.
5. Setelah Anda yakin bahwa rangkaian berfungsi dengan baik sebagai hasil simulasi Anda, unduh rangkaiannya ke dalam chip FPGA. Uji fungsionalitas desain Anda dengan menerapkan urutan masukan dan mengamati LED keluaran. Pastikan FSM bertransisi dengan benar antar status seperti yang ditampilkan pada LED merah, dan menghasilkan nilai keluaran yang benar pada LEDR9.
6. Terakhir, pertimbangkan modifikasi kode one-hot yang diberikan pada Tabel 1. Seringkali diinginkan untuk menyetel semua flip-flop output ke nilai 0 dalam keadaan reset.
7. Tabel 2 menunjukkan penetapan status one-hot yang dimodifikasi di mana status reset, A, menggunakan semua angka 0. Hal ini dilakukan dengan membalik variabel keadaan y0. Buat versi modifikasi dari kode Verilog Anda yang mengimplementasikan penugasan negara ini. (Petunjuk: Anda hanya perlu melakukan sedikit perubahan pada ekspresi logika di sirkuit untuk mengimplementasikan tugas negara yang dimodifikasi.)

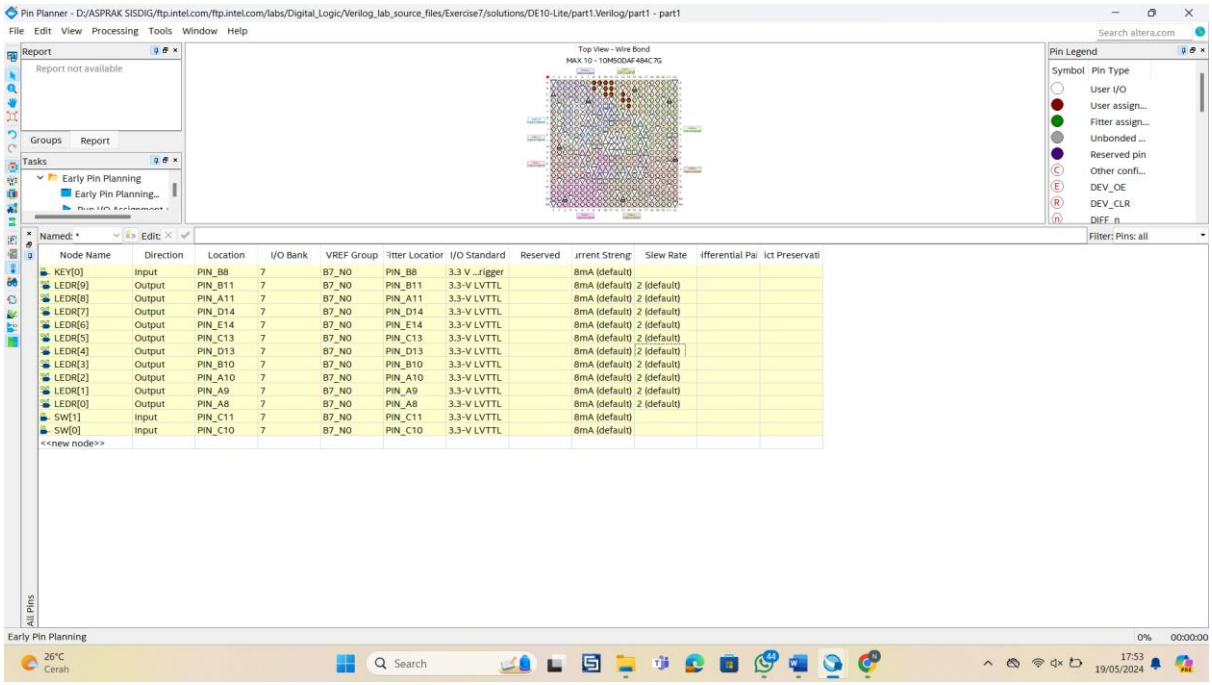
8. Kompilasi sirkuit baru Anda dan ujilah.

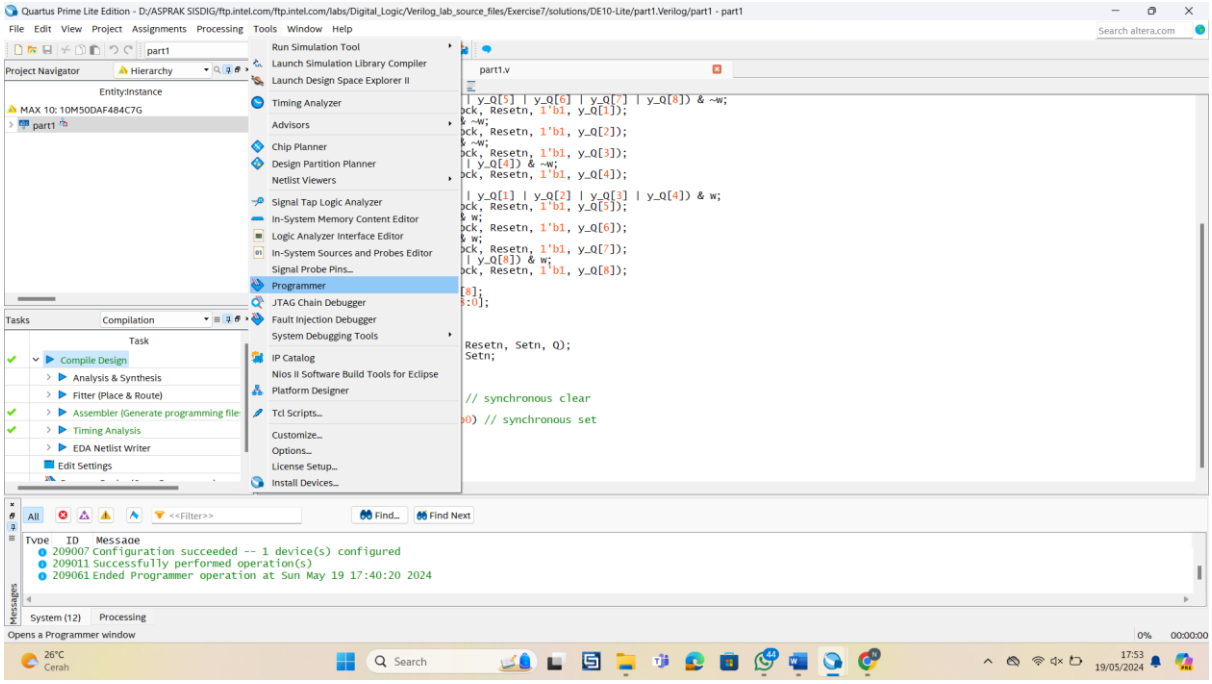


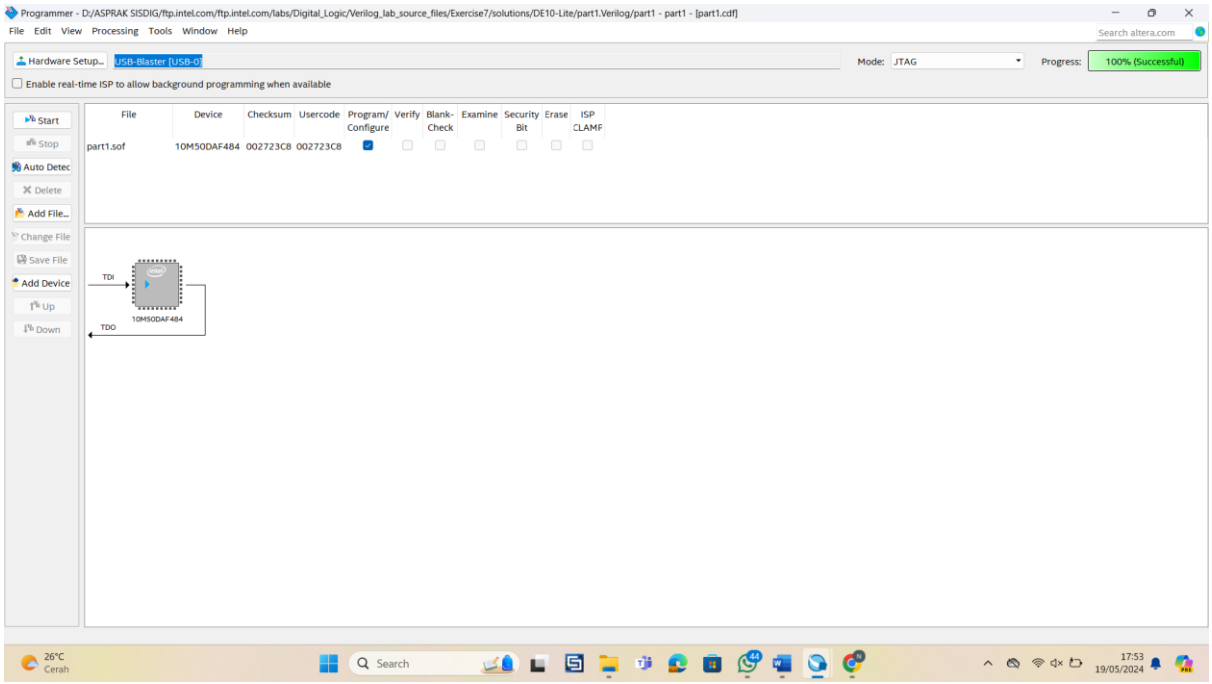
Nama	Kode Negara y8y7y6y5y4y3y2y1y0
A	000000000
B	000000011
C	000000101
D	000001001
DAN	000010001
F	000100001
G	001000001
H	010000001
...	100000001

Tabel 2: Kode one-hot yang dimodifikasi untuk FSM.

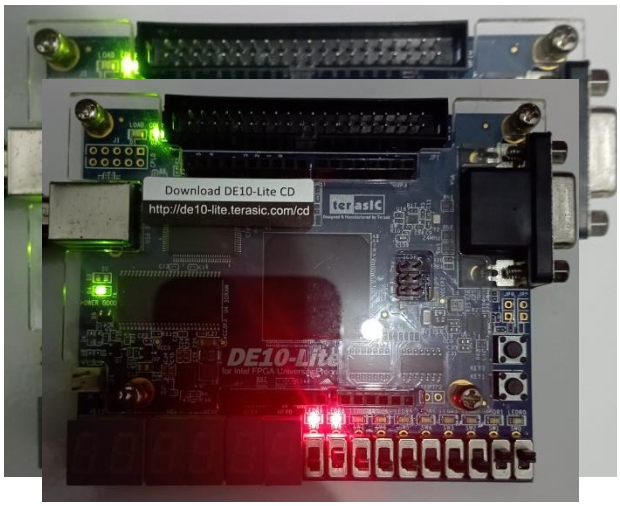

Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part1.Verilog/part1 - part1
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part1
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
part1.v
assign Y_D[1] = (y_Q[0] | y_Q[5] | y_Q[6] | y_Q[7] | y_Q[8]) & ~w;
flipflop ff1(Y_D[1], Clock, Resetn, 1'b1, y_Q[1]);
assign Y_D[2] = y_Q[1] & ~w;
flipflop ff2(Y_D[2], Clock, Resetn, 1'b1, y_Q[2]);
assign Y_D[3] = y_Q[2] & ~w;
flipflop ff3(Y_D[3], Clock, Resetn, 1'b1, y_Q[3]);
assign Y_D[4] = (y_Q[3] | y_Q[4]) & ~w;
flipflop ff4(Y_D[4], Clock, Resetn, 1'b1, y_Q[4]);
assign Y_D[5] = (y_Q[0] | y_Q[1] | y_Q[2] | y_Q[3] | y_Q[4]) & w;
flipflop ff5(Y_D[5], Clock, Resetn, 1'b1, y_Q[5]);
assign Y_D[6] = y_Q[5] & w;
flipflop ff6(Y_D[6], Clock, Resetn, 1'b1, y_Q[6]);
assign Y_D[7] = y_Q[6] & w;
flipflop ff7(Y_D[7], Clock, Resetn, 1'b1, y_Q[7]);
assign Y_D[8] = (y_Q[7] | y_Q[8]) & w;
flipflop ff8(Y_D[8], Clock, Resetn, 1'b1, y_Q[8]);
assign z = y_Q[4] | y_Q[8];
assign LEDR[8:0] = y_Q[8:0];
assign LEDR[9] = z;
endmodule
module flipflop (D, Clock, Resetn, Setn, Q);
input D, Clock, Resetn, Setn;
output reg Q;
always @(posedge Clock)
if (Resetn == 1'b0) // synchronous clear
Q <= 1'b0;
else if (Setn == 1'b0) // synchronous set
Q <= 1'b1;
else
Q <= D;
endmodule
Find...
Find Next
Type ID Message
209007 Configuration succeeded -- 1 device(s) configured
209011 Successfully performed operation(s)
209061 Ended Programmer operation at Sun May 19 17:40:20 2024
System (12)
Processing
0%
00:00:00
26°C
Cerah
Search
17:48
19/05/2024

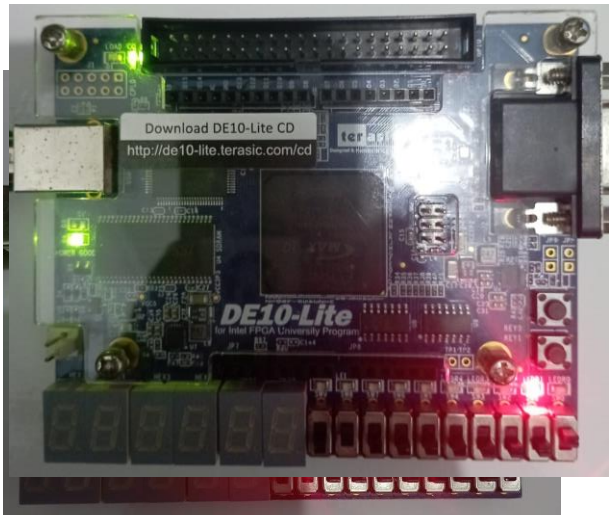

Pin Planner - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part1.Verilog/part1 - part1
File Edit View Processing Tools Window Help
Report
Report not available
Groups
Report
Tasks
Early Pin Planning
Early Pin Planning...
Top View - Wire Bond
MAX 10 - 10M50DAF484C7G
Pin Legend
Symbol Pin Type
User I/O
User assign...
Fitter assign...
Unbonded ...
Reserved pin
Other confi...
DEV_OE
DEV_CLR
Named: *
Edit:
Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard Reserved Current Strength Slew Rate Differential Pair Bus Preservation
KEY[0] Input PIN_B8 7 B7_N0 PIN_B8 3.3 V ...rigger 8mA (default)
LEDR[9] Output PIN_B11 7 B7_N0 PIN_B11 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[8] Output PIN_A11 7 B7_N0 PIN_A11 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[7] Output PIN_D14 7 B7_N0 PIN_D14 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[6] Output PIN_E14 7 B7_N0 PIN_E14 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[5] Output PIN_C13 7 B7_N0 PIN_C13 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[4] Output PIN_D13 7 B7_N0 PIN_D13 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[3] Output PIN_B10 7 B7_N0 PIN_B10 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[2] Output PIN_A10 7 B7_N0 PIN_A10 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[1] Output PIN_A9 7 B7_N0 PIN_A9 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[0] Output PIN_A8 7 B7_N0 PIN_A8 3.3-V LVTTL 8mA (default) 2 (default)
SW[1] Input PIN_C11 7 B7_N0 PIN_C11 3.3-V LVTTL 8mA (default)
SW[0] Input PIN_C10 7 B7_N0 PIN_C10 3.3-V LVTTL 8mA (default)
<<new node>>
All Pins
Early Pin Planning
0%
00:00:00
26°C
Cerah
Search
17:53
19/05/2024

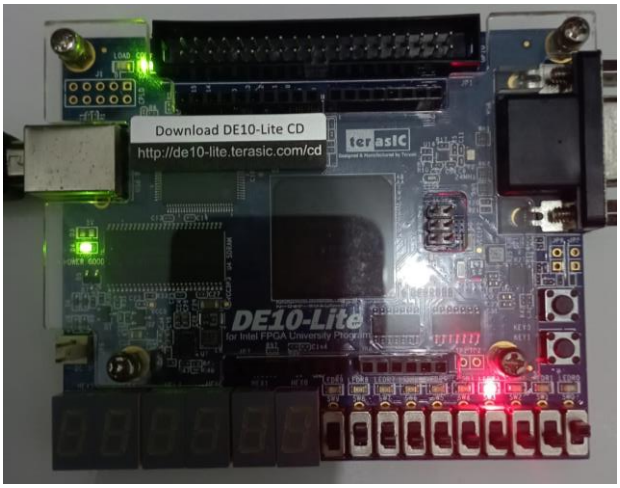

Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part1.Verilog/part1 - part1
File Edit View Project Assignments Processing Tools Window Help
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer II
Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
Signal Tap Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
Signal Probe Pins...
Programmer
JTAG Chain Debugger
Fault Injection Debugger
System Debugging Tools
IP Catalog
Nios II Software Build Tools for Eclipse
Platform Designer
Tcl Scripts...
Customize...
Options...
License Setup...
Install Devices...
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
209007 Configuration succeeded -- 1 device(s) configured
209011 Successfully performed operation(s)
209061 Ended Programmer operation at Sun May 19 17:40:20 2024

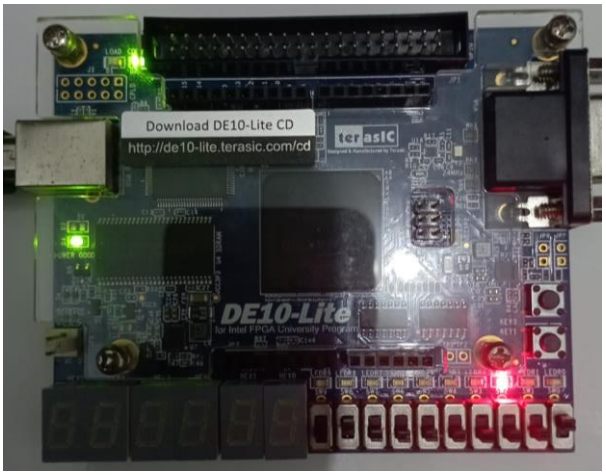

Programmer - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part1.Verilog/part1 - part1 - [part1.cdf]
Hardware Setup...
USB-Blaster [USB-0]
Mode: JTAG
Progress:
100% (Successful)
Enable real-time ISP to allow background programming when available
Start
Stop
Auto Detec
Delete
Add File...
Change File
Save File
Add Device
Up
Down
File
Device
Checksum
Usercode
Program/ Configure
Verify
Blank-Check
Examine
Security Bit
Erase
ISP CLAMP
part1.sof
10M50DAF484
002723C8
002723C8
TDI
TDO
10M50DAF484

Hasil Bagian satu di fpga :









Bagian II

Untuk bagan ini, Anda harus menulis kode Verilog gaya lain untuk FSM pada Gambar 2. Pada versi kode ini, Anda tidak perlu memurunkan ekspresi kagiku yang dibutuhkan untuk setiap state flip-flop secara manual Sebagai gammya, gambarkan tabel state untuk FSM dengan menggunakan pernyataan kasus Verilog dalam blok selalu, dan gunakan blok selalu lainnya untuk menginstansiasi flip-flop state. Anda bisa menggunakan blok selalu ketiga atau pernyataan penugasan sederhana untuk menentukan keluaran Untuk mengimplementasikan FSM. gunakan empat buah sate lip-flop v dan kode-kode biner, seperti yang ditunjukkan pada Tabel 3.

Nama	Kode Negara $y\_y\_y\_y_{3210}$
A	0000
B	0001
C	0010
D	0011
E	0100
F	0101
G	0110
H	0111
I	1000

Tabel 3 Kode biner untuk FSM

1. Membuat proyek baru untuk FSM.

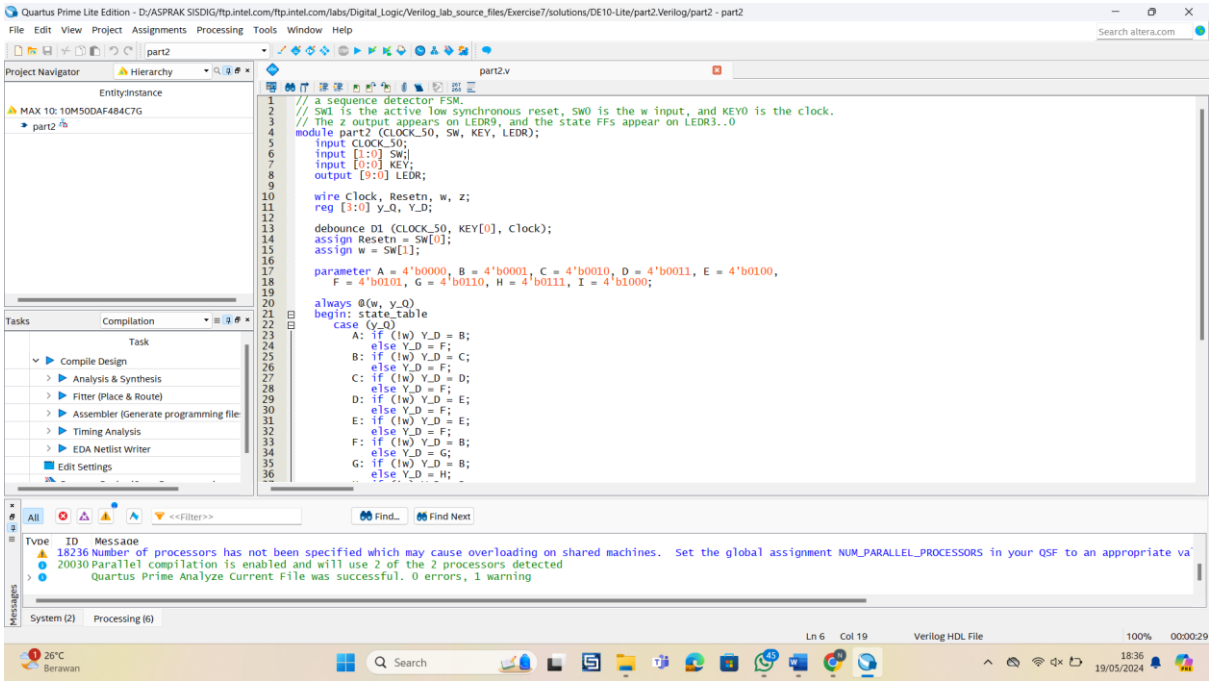
2. Sertakan file Verilog Anda dalam proyek yang menggunakan gaya kode pada Gambar 3. Gunakan sakelar, tombol tekan, dan lampu yang sama dengan yang digunakan pada Bagian 1.

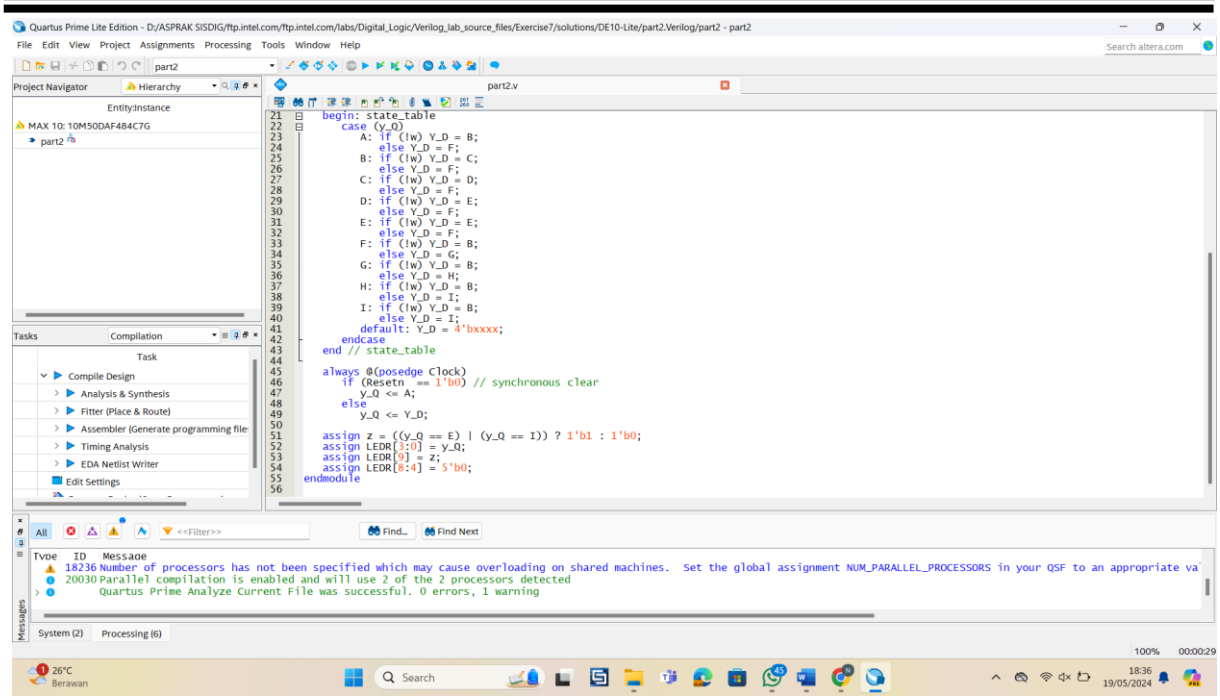
3. Sebelum mengkompilasi kode Anda. Anda harus secara eksplisit memberi tahu alat Synthesis di Quartus bahwa Anda ingin mesin keadaan terbatas dumplementasikan dengan menggunakan pemagasan keadaan yang diuntukan dalam kode Verilog Anda. Jika Anda tidak secara eksplisit memberikan pengaturan ini ke Quartus, Synthesis tool akan secara otomatis menggunakan state assignment yang dipilihnya sendiri, dan akan anengabaikan state code yang ditentukan dalam kode Verilog Anda. Untuk membuat pengaturan ini. pilih Assignments > Settings di Quarties, dan klik pada item Compiler Settings di sisi kiri jendela, lalu lik tombol Advanced Settings (Synthesis), Seperti yang ditunjukkan pada Gambar 4, uhah parameter State Machine Processing ke pengaturan User-Encoded.

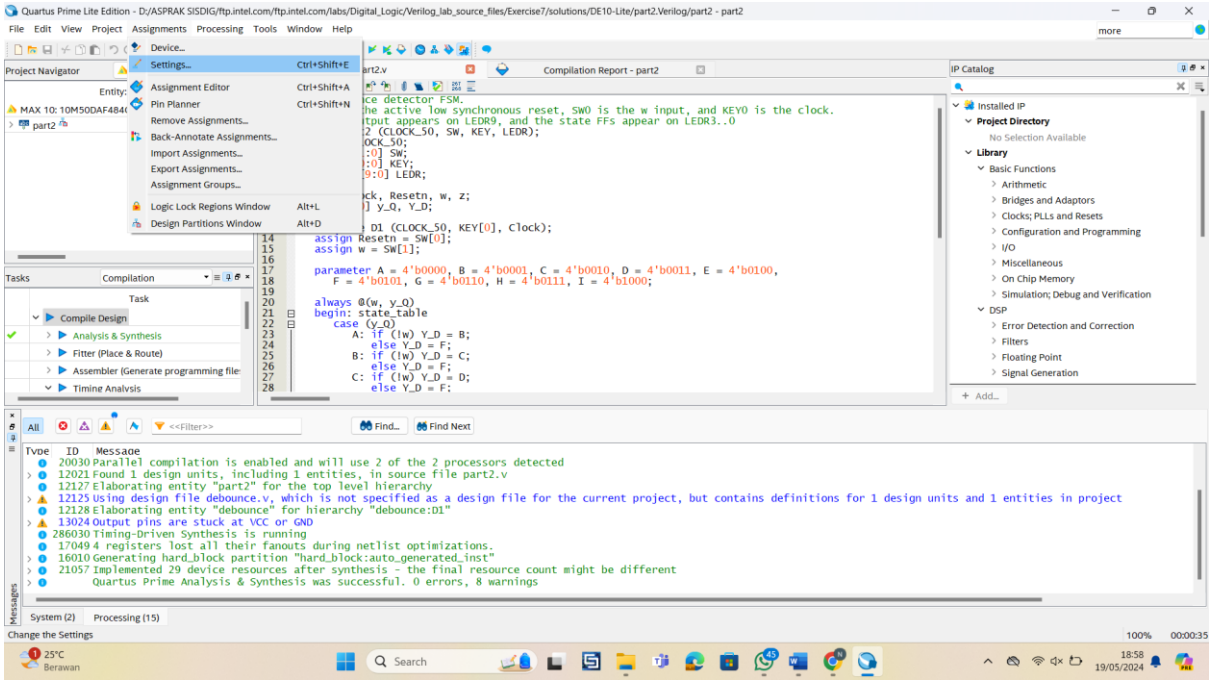
4. Kompilasi proyek Anda. Untuk memeriksa rangkaian yang dihasilkan oleh Quartus, huka alat RTL Viewer. Klik dan kali pada kotak yang ditunjukkan pada rangkaian yang merepresentasikan finite state machine, dan tentukan apakah state diagram yang ditunjukkan sesuai dengan yang ada pada Gumhar 2. Untuk melihat kode-kode state yang digunakan untuk FSM Anda, boka Laporan Kompilasi, pilih bagian Analisis dan Sintesis pada laporan tersebut, dan klik State Machines.

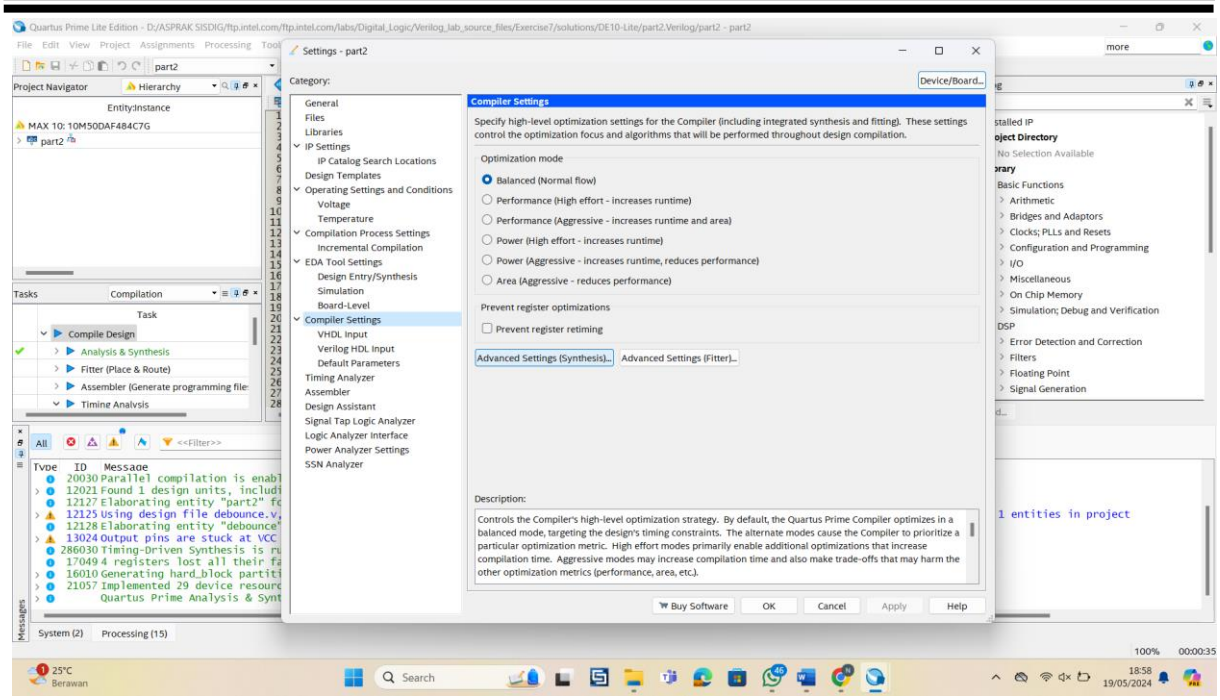
5. Unduh rangkaian ke dalam chip FPGA dan uji fungsinya.

6. Pada langkah 3. Anda menginstruksikan alat Quartas Synthesis untuk menggunakan state assignment yang diberikan pada kode Verilog Anda. Lamak melihat hasil dari menghapus penganuran ini, huka kembali jendela pengaturan Quartus dengan memilih Assignments > Settings, dan klik pada item Compiler Settings di sisi kiri jendela, kemudian klik tombol Advanced Settings (Synthesis). Ibab pengaturan untuk Pemrosesan Mesin Negara dari User-Encoded ke One-Hot Kompilasi ulang rangkaian lalu buka file laporan, pilih bagian Analisis dan Sintesis pada laporan, dan klik State Machines. Bandingkan kode keadaan yang ditampilkan dengan yang diberikan pada Tabel 2, dan diskusikan perbedaan yang Anda amati

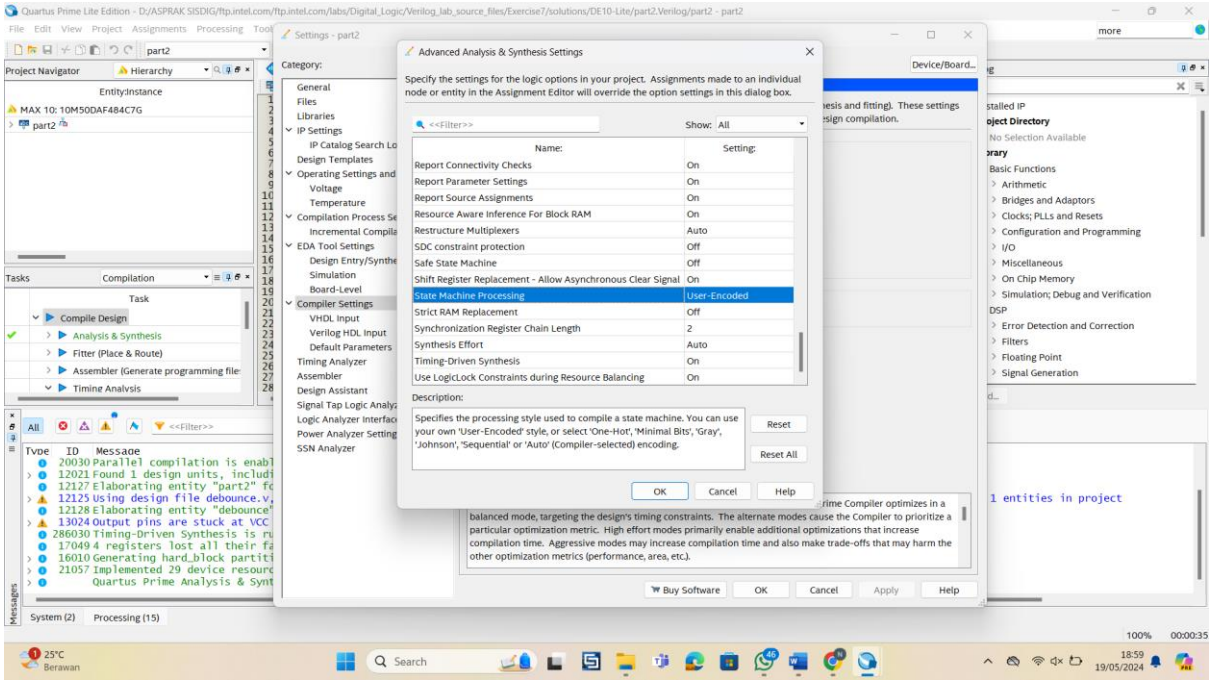


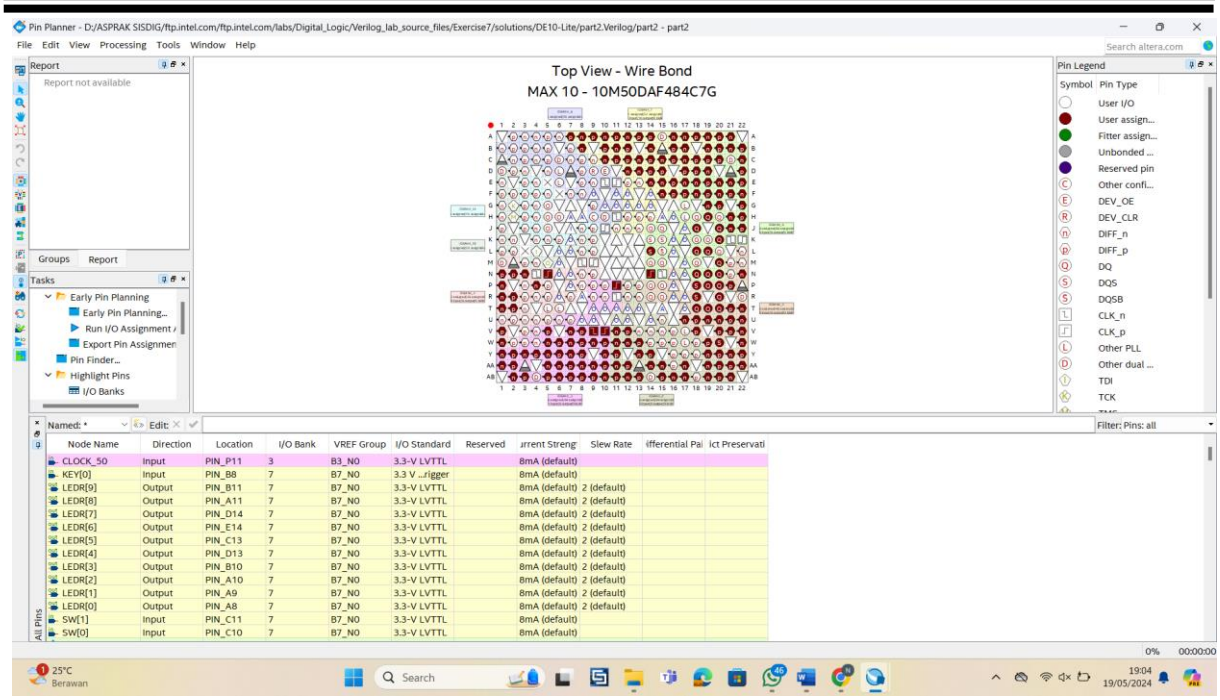

Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part2.Verilog/part2 - part2
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part2
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
part2.v
begin: state_table
case (y_Q)
A: if (!w) Y_D = B;
else Y_D = F;
B: if (!w) Y_D = C;
else Y_D = F;
C: if (!w) Y_D = D;
else Y_D = F;
D: if (!w) Y_D = E;
else Y_D = F;
E: if (!w) Y_D = E;
else Y_D = F;
F: if (!w) Y_D = B;
else Y_D = G;
G: if (!w) Y_D = B;
else Y_D = H;
H: if (!w) Y_D = B;
else Y_D = I;
I: if (!w) Y_D = B;
else Y_D = I;
default: Y_D = 4'bxxxx;
endcase
end // state_table
always @(posedge Clock)
if (Resetn == 1'b0) // synchronous clear
y_Q <= A;
else
y_Q <= Y_D;
assign z = ((y_Q == E) | (y_Q == I)) ? 1'b1 : 1'b0;
assign LEDR[3:0] = y_Q;
assign LEDR[9] = z;
assign LEDR[8:4] = 5'b0;
endmodule
Find... Find Next
Type ID Message
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate va
20030 Parallel compilation is enabled and will use 2 of the 2 processors detected
Quartus Prime Analyze Current File was successful. 0 errors, 1 warning
System (2)
Processing (6)
100%
00:00:29
26°C
Berawan
Search
18:36
19/05/2024

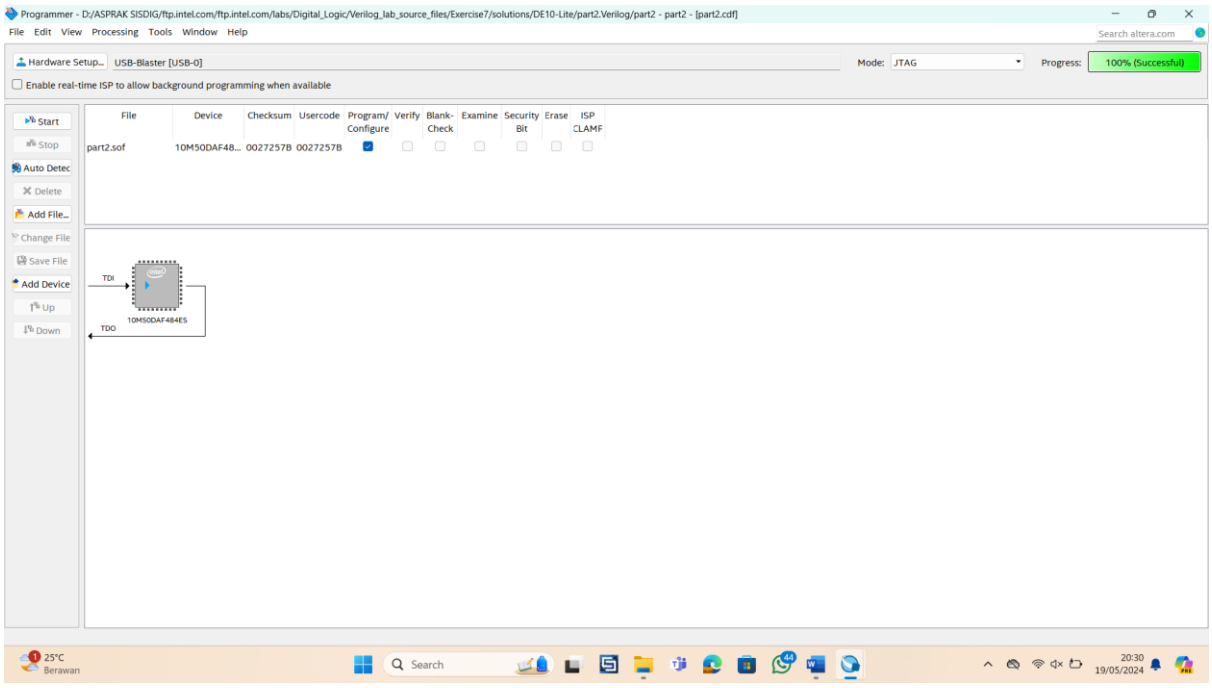

Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part2.Verilog/part2 - part2
File Edit View Project Assignments Processing Tools Window Help
more
Device...
Settings... Ctrl+Shift+E
Assignment Editor Ctrl+Shift+A
Pin Planner Ctrl+Shift+N
Remove Assignments...
Back-Annotate Assignments...
Import Assignments...
Export Assignments...
Assignment Groups...
Logic Lock Regions Window Alt+L
Design Partitions Window Alt+D
Compilation Report - part2
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
Arithmetic
Bridges and Adaptors
Clocks; PLLs and Resets
Configuration and Programming
I/O
Miscellaneous
On Chip Memory
Simulation; Debug and Verification
DSP
Error Detection and Correction
Filters
Floating Point
Signal Generation
Add...
assign Resetn = SW[0];
assign w = SW[1];
parameter A = 4'b0000, B = 4'b0001, C = 4'b0010, D = 4'b0011, E = 4'b0100,
F = 4'b0101, G = 4'b0110, H = 4'b0111, I = 4'b1000;
always @(w, y_Q)
begin: state_table
case (y_Q)
A: if (!w) Y_D = B;
else Y_D = F;
B: if (!w) Y_D = C;
else Y_D = F;
C: if (!w) Y_D = D;
else Y_D = F;
Find... Find Next
Type ID Message
20030 Parallel compilation is enabled and will use 2 of the 2 processors detected
12021 Found 1 design units, including 1 entities, in source file part2.v
12127 Elaborating entity "part2" for the top level hierarchy
12125 Using design file debounce.v, which is not specified as a design file for the current project, but contains definitions for 1 design units and 1 entities in project
12128 Elaborating entity "debounce" for hierarchy "debounce:D1"
13024 Output pins are stuck at VCC or GND
286030 Timing-Driven Synthesis is running
17049 4 registers lost all their fanouts during netlist optimizations.
16010 Generating hard_block partition "hard_block:auto_generated_inst"
21057 Implemented 29 device resources after synthesis - the final resource count might be different
Quartus Prime Analysis & Synthesis was successful. 0 errors, 8 warnings
System (2)
Processing (15)
Change the Settings
100%
00:00:35
25°C
Berawan
Search
18:58
19/05/2024



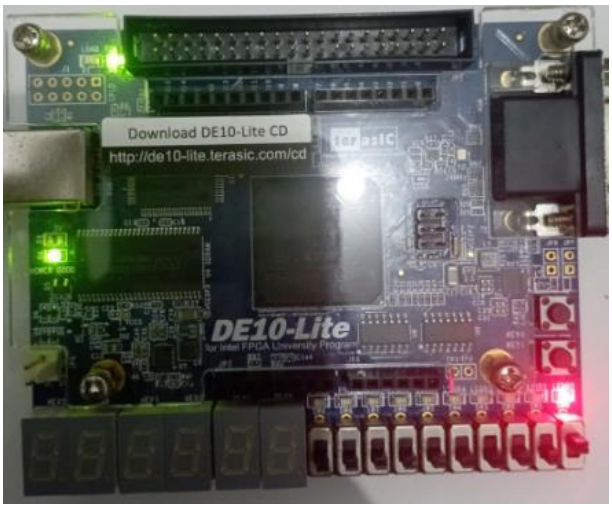
Setelah klik OK lalu APPLY dan OK.

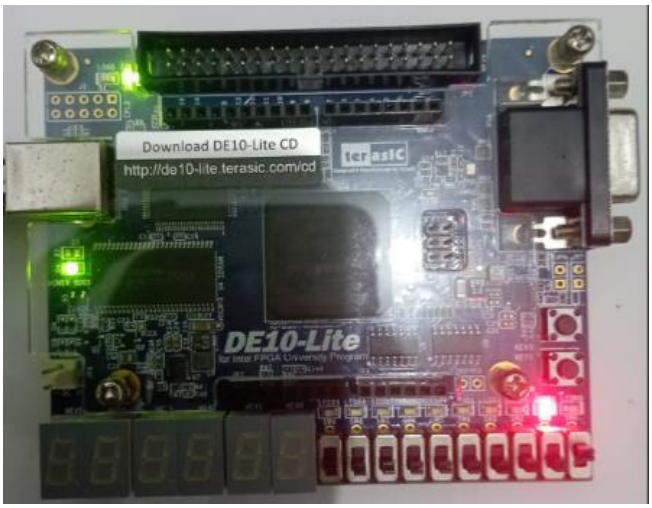


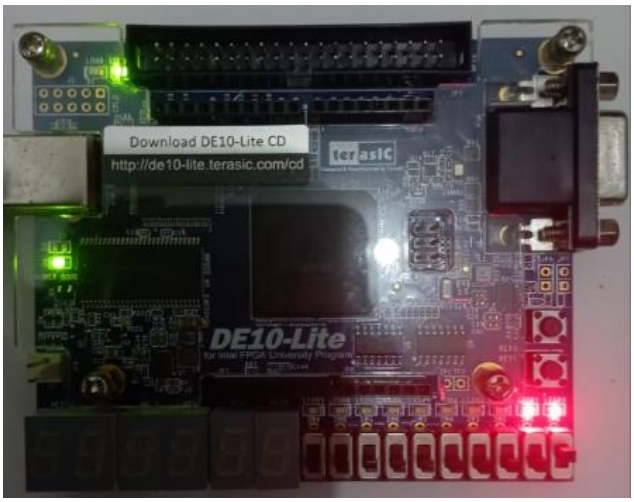




- Hasil compile bagian II di FPGA:



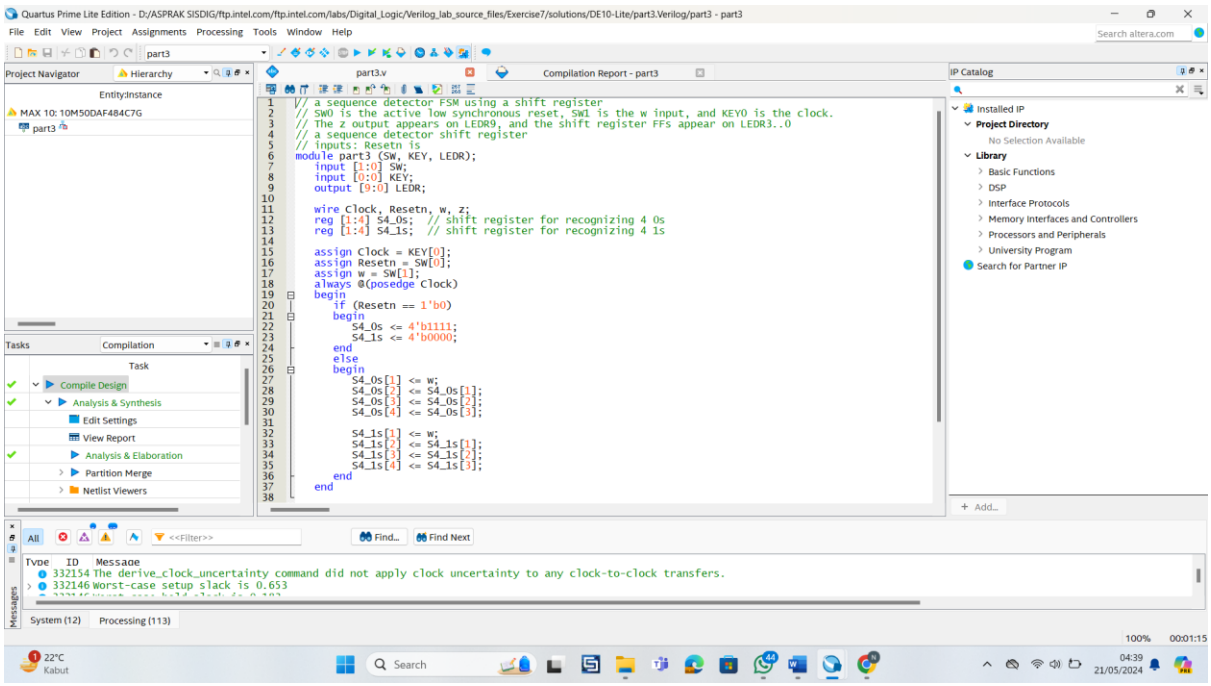


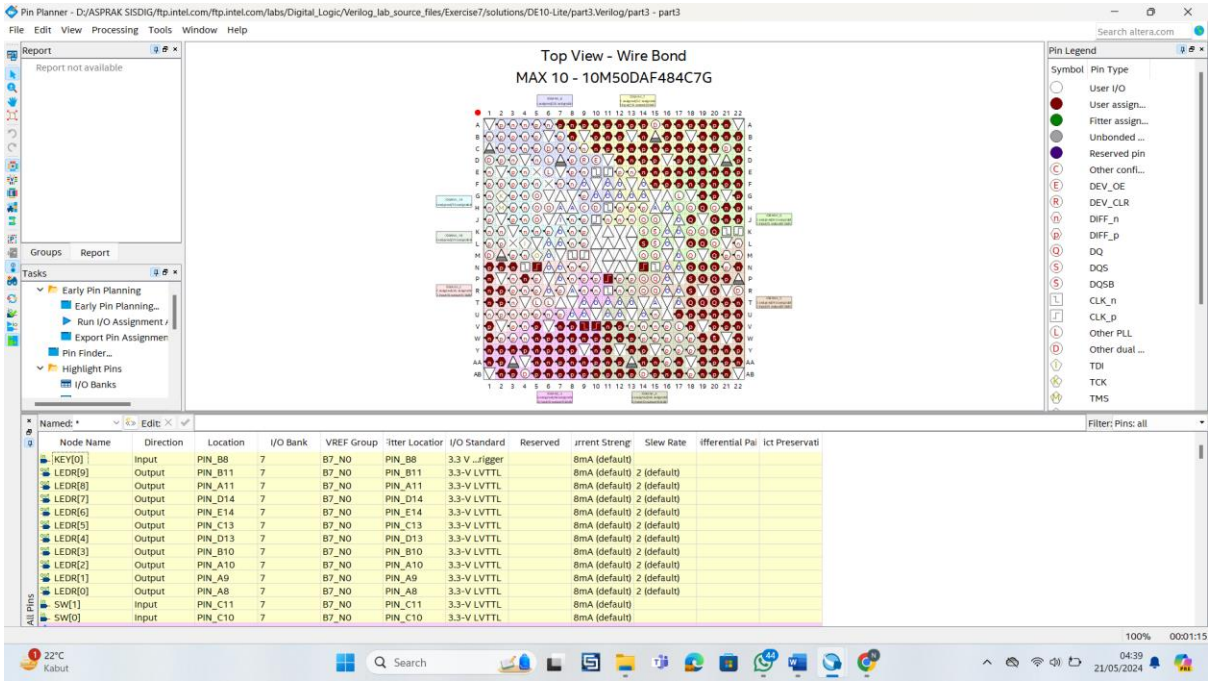


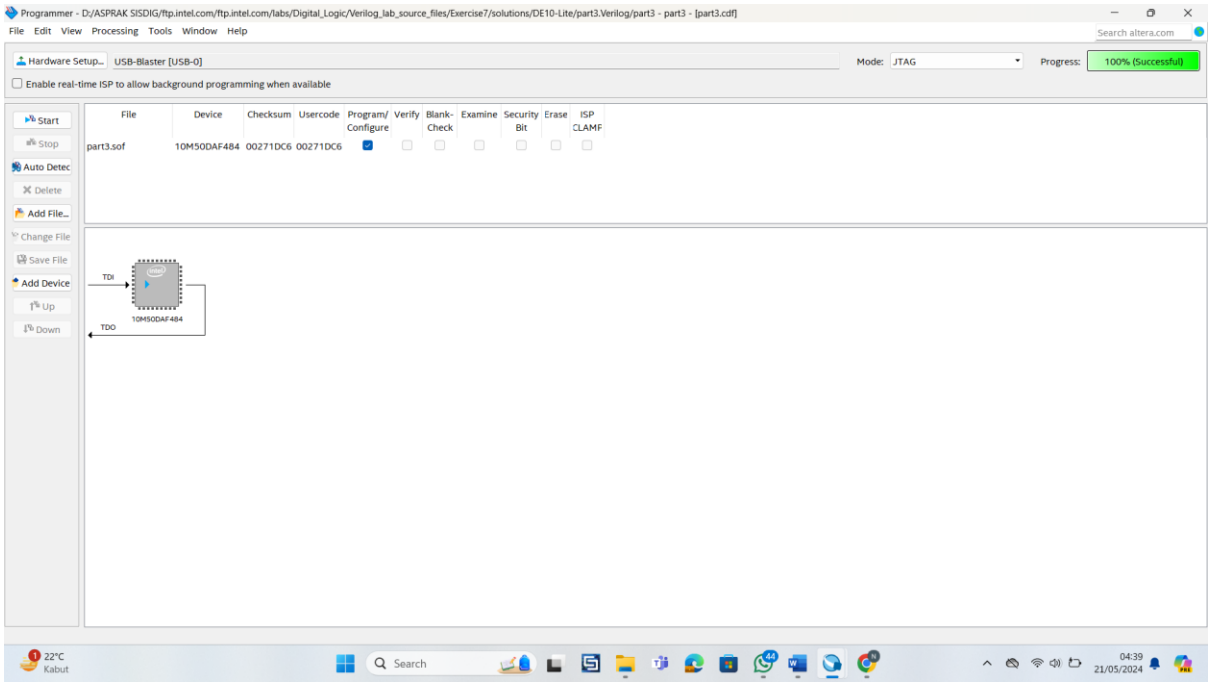
Bagian III

Detektor urutan dapat diimplementasikan secara langsung dengan menggunakan register geser, daripada menggunakan pendekatan yang lebih formal seperti dijelaskan di atas. Buat kode Verilog yang membuat instance dua register geser 4-bit; satu untuk mengenali urutan empat angka 0, dan yang lainnya untuk empat angka 1. Sertakan ekspresi logika yang sesuai dalam desain Anda untuk menghasilkan keluaran z. Buat proyek Quartus untuk desain Anda dan terapkan sirkuit pada papan seri DE Anda. Gunakan sakelar dan LED pada papan dengan cara yang sama seperti yang Anda lakukan pada Bagian I dan II dan amati perilaku register geser dan output z. Jawab pertanyaan berikut:

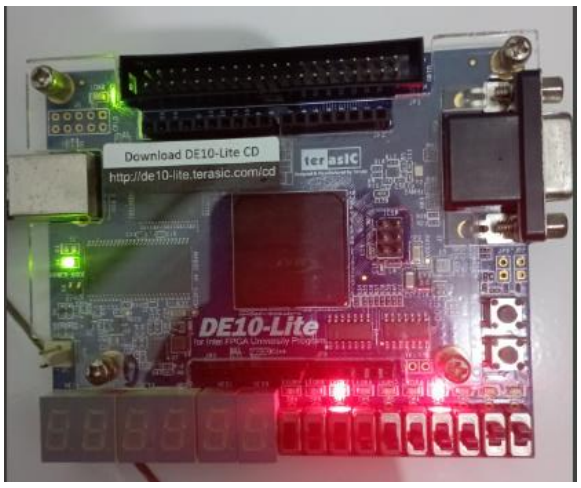
bisakah Anda menggunakan satu register geser 4-bit saja, bukan dua? Jelaskan jawabanmu.

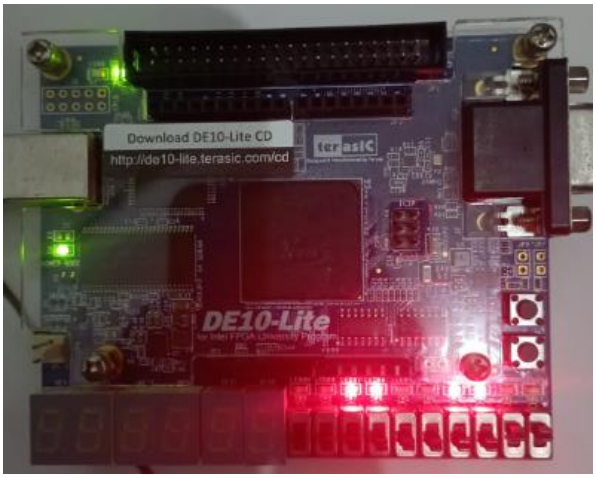


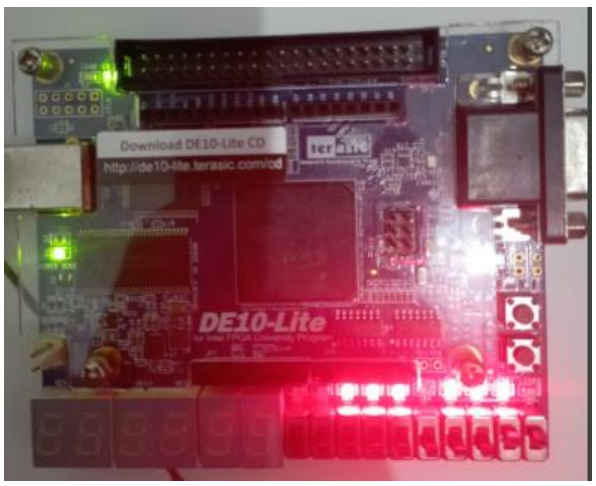


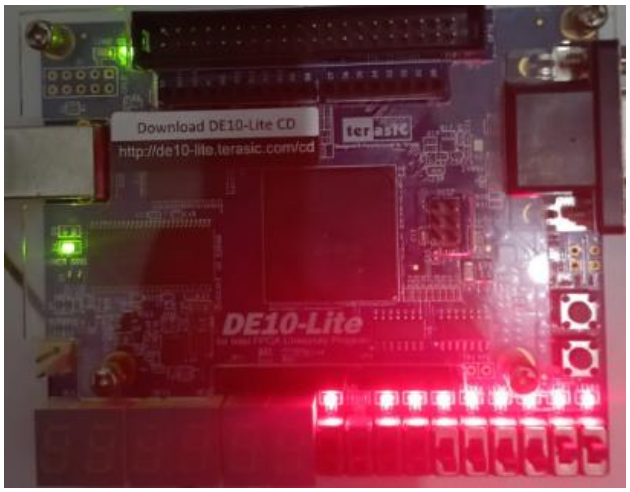


- **Hasil compile bagian III di FPGA :**









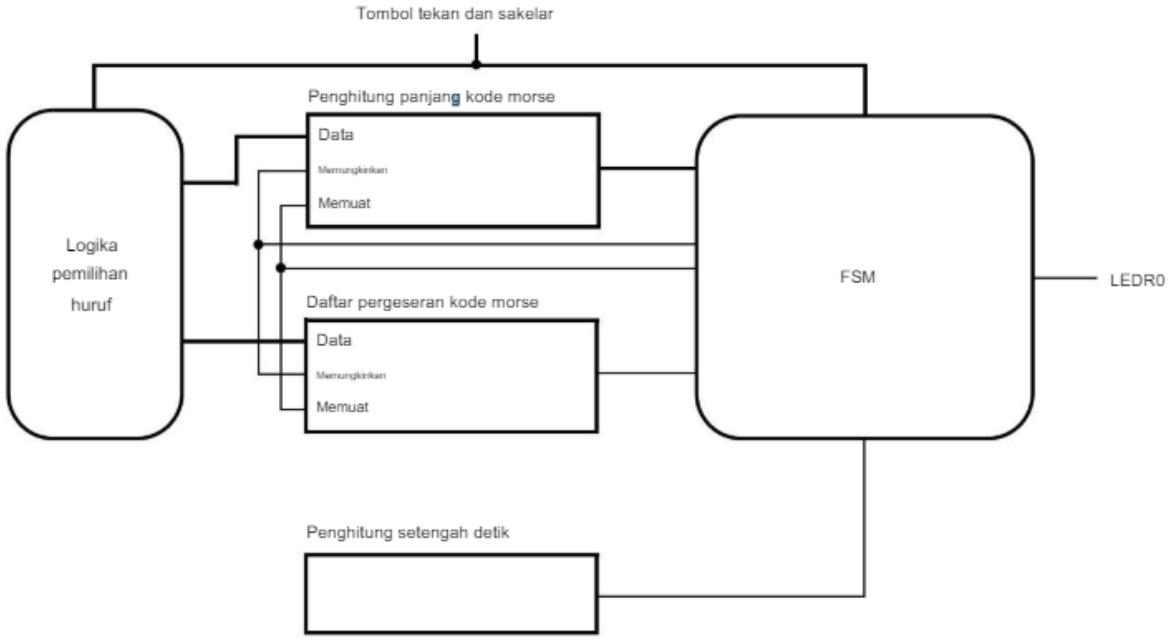
Bagian IV

Pada bagian latihan ini Anda akan mengimplementasikan encoder kode Morse menggunakan FSM. Kode Morse menggunakan pola pulsa pendek dan panjang untuk mewakili sebuah pesan. Setiap huruf direpresentasikan sebagai rangkaian titik (pulsa pendek), dan garis (pulsa panjang). Misalnya, delapan huruf pertama dalam alfabet memiliki representasi berikut:

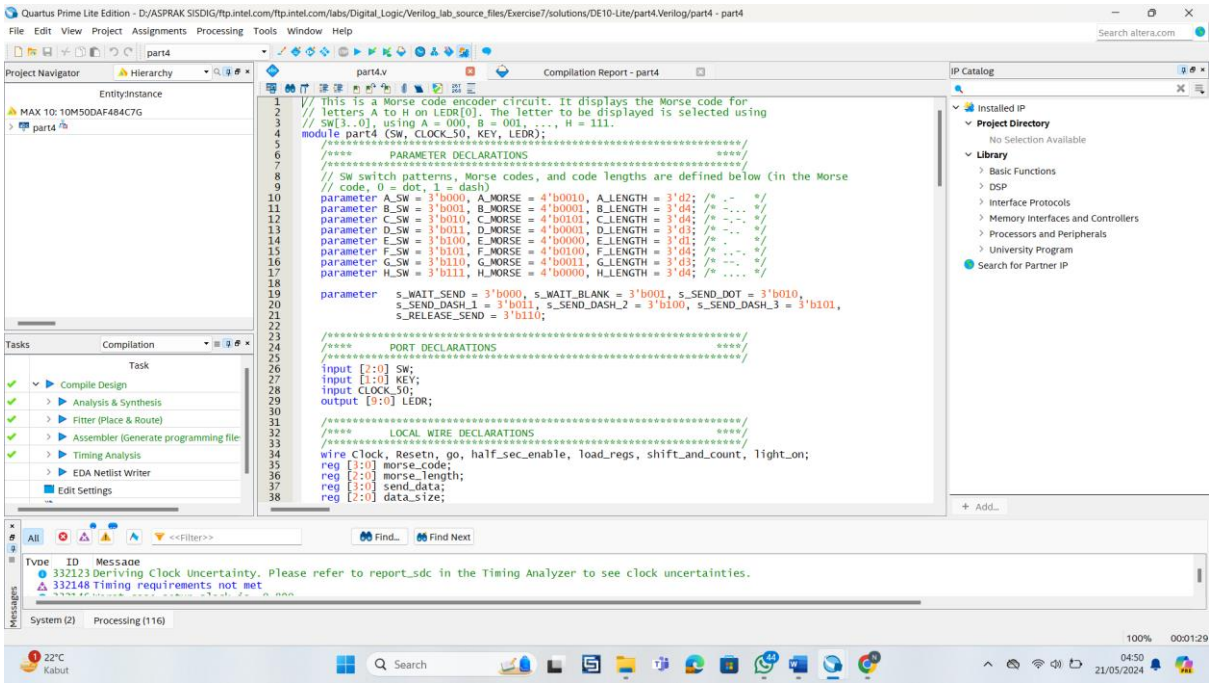
A • -
B - • • •
C - • - •
D - • •
DAN •
F • • - •
G - - •
H • • • •

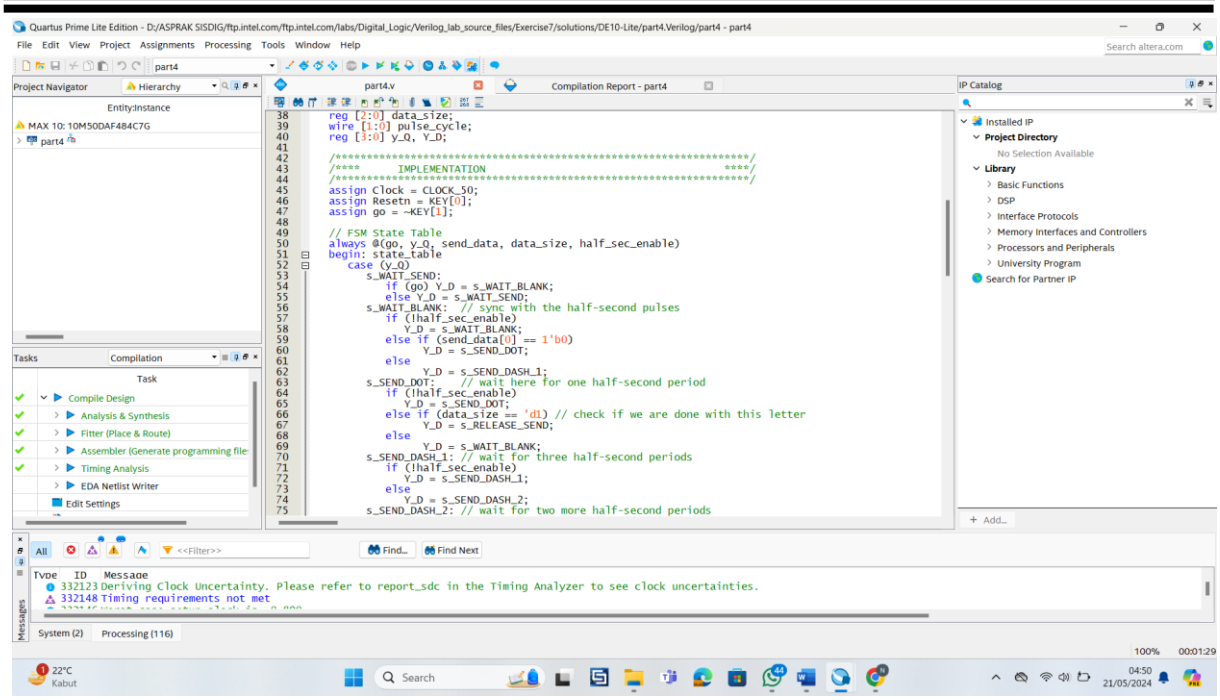
Merancang dan mengimplementasikan rangkaian encoder kode Morse menggunakan FSM. Sirkuit Anda harus mengambil salah satu dari delapan huruf pertama alfabet sebagai masukan dan menampilkan kode Morse pada LED merah. Gunakan sakelar SW2ÿ0 dan tombol tekan KEY1ÿ0 sebagai input. Ketika pengguna menekan KEY1, sirkuit akan menampilkan kode Morse untuk huruf yang ditentukan oleh SW2ÿ0 (000 untuk A, 001 untuk B, dll.), menggunakan pulsa 0,5 detik untuk mewakili titik, dan pulsa 1,5 detik untuk mewakili tanda hubung. Tombol tekan KEY0 harus berfungsi sebagai reset asinkron.

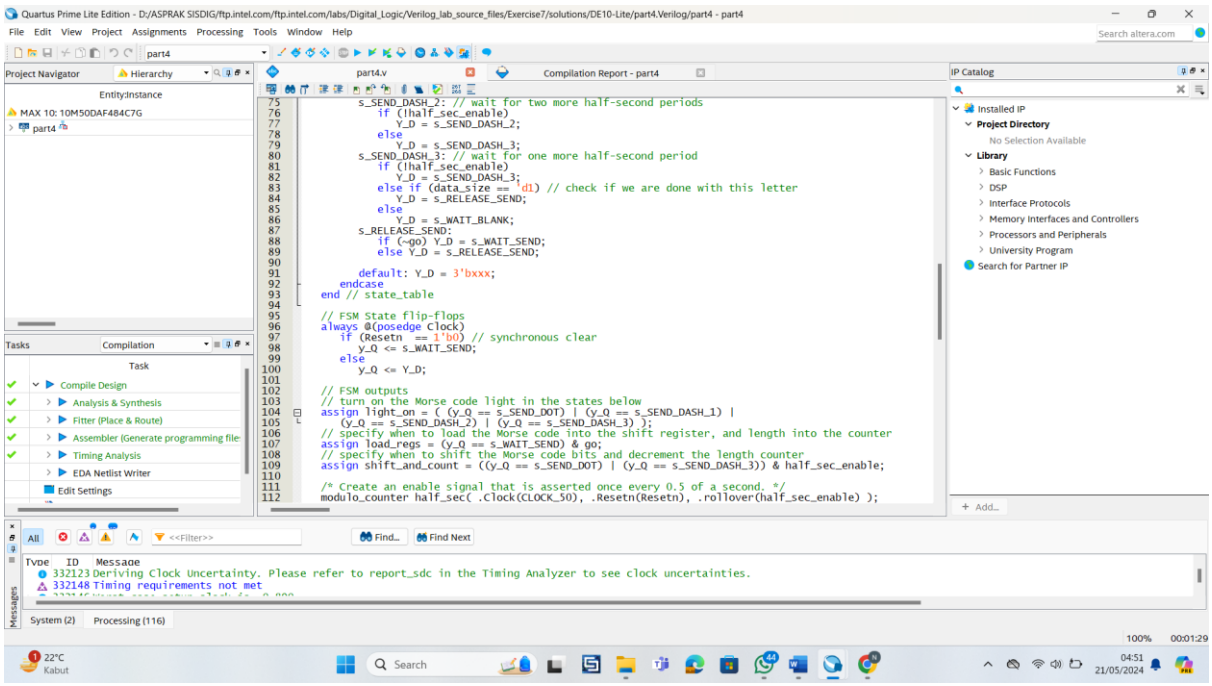
Diagram skema tingkat tinggi dari kemungkinan rangkaian untuk pembuat kode Morse ditunjukkan pada Gambar 5.

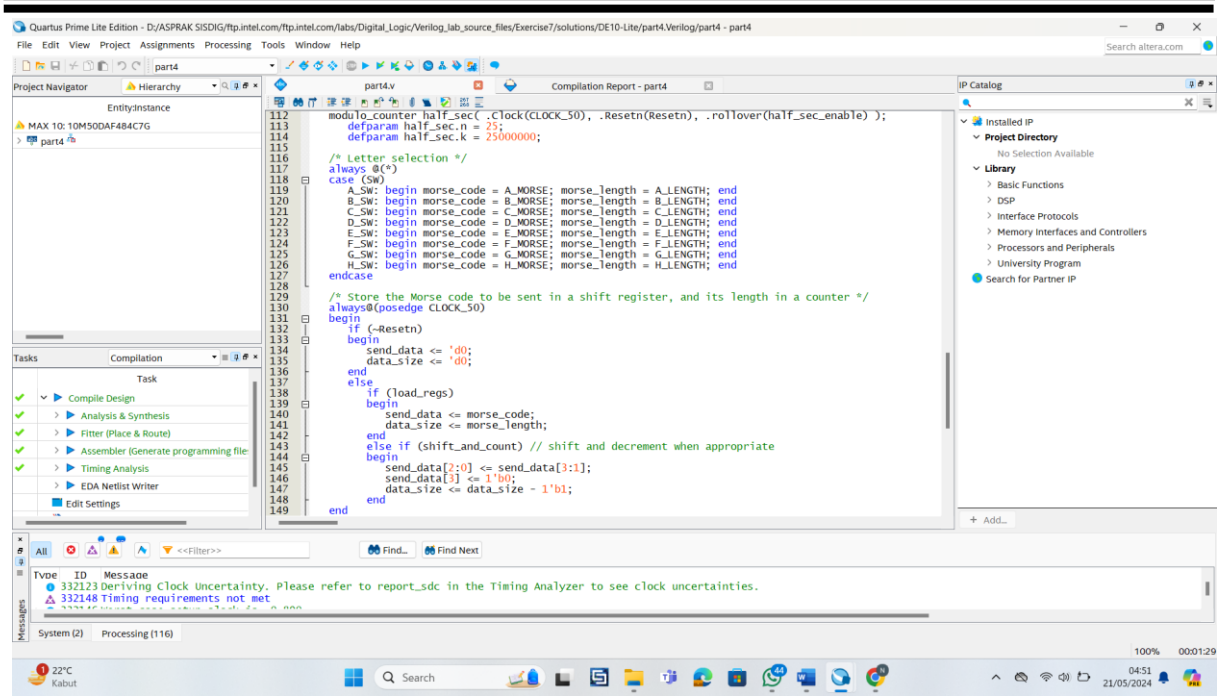


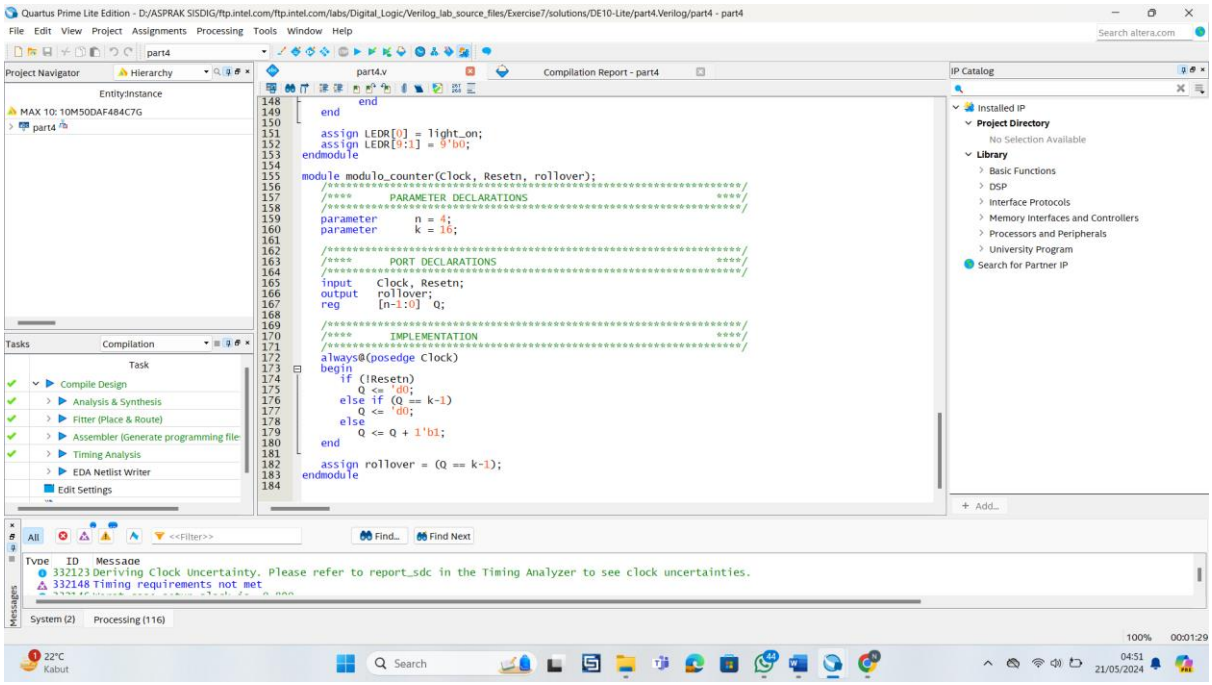
Gambar 5: Diagram skematik tingkat tinggi dari rangkaian untuk Bagian IV.

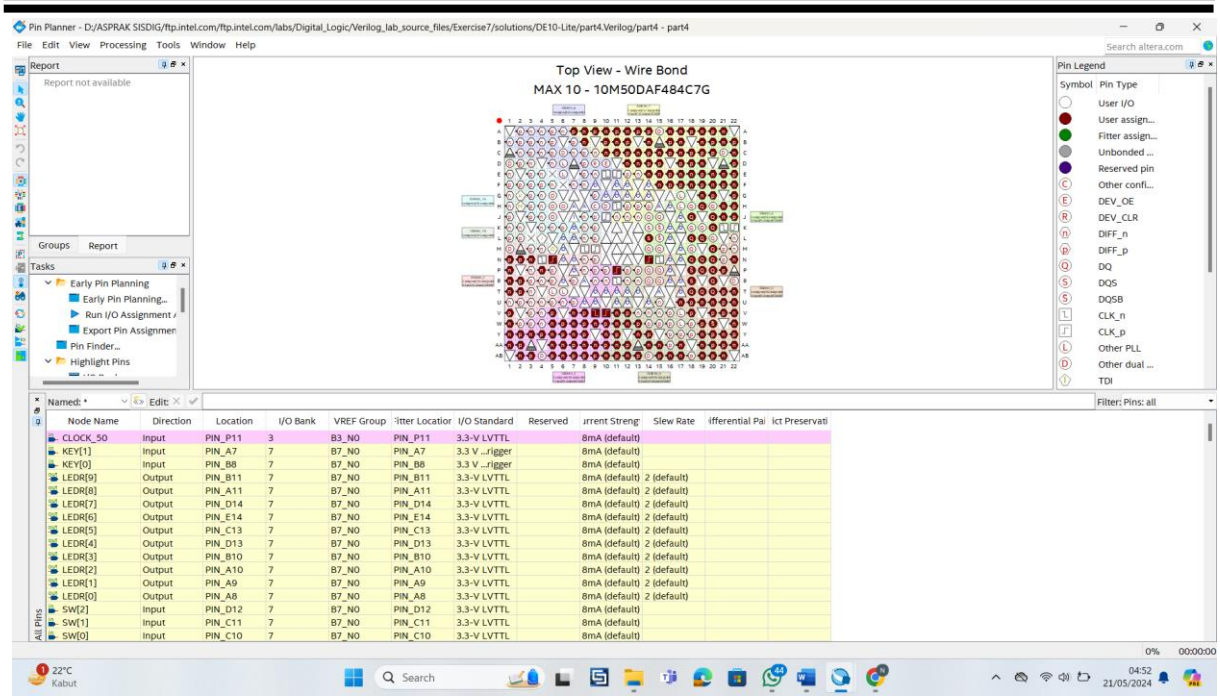


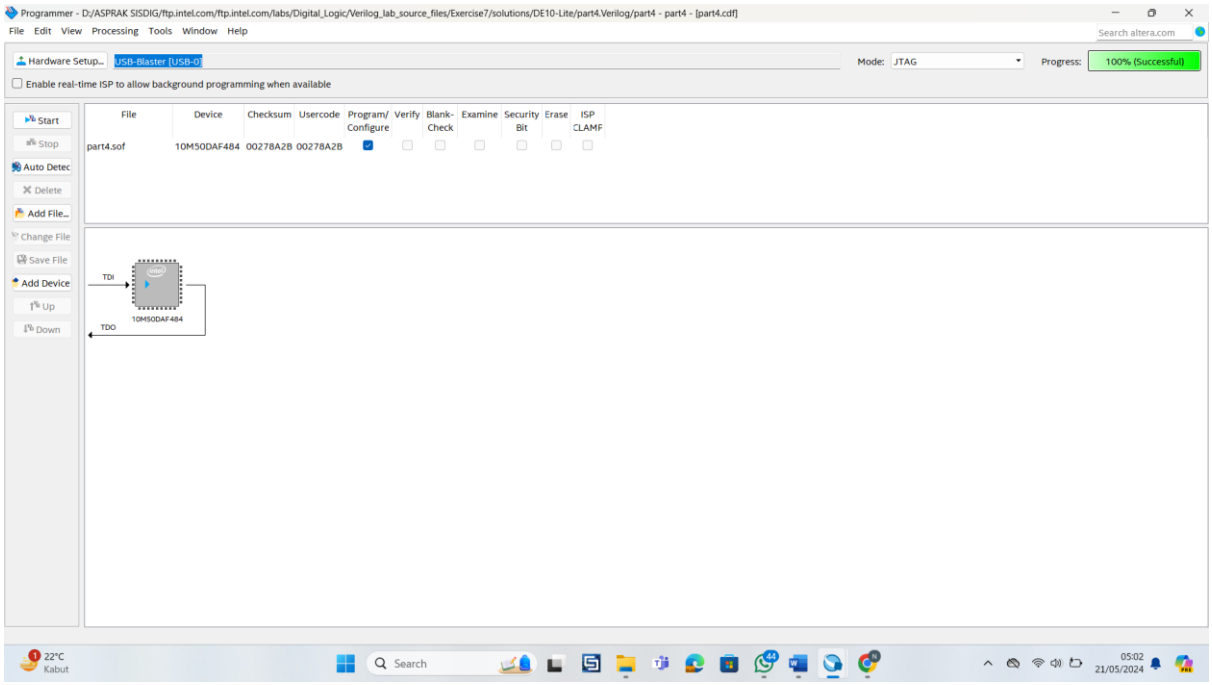

Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part4
Compilation Report - part4
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming file
Timing Analysis
EDA Netlist Writer
Edit Settings
Type ID Message
332123 Deriving Clock Uncertainty. Please refer to report_sdc in the Timing Analyzer to see clock uncertainties.
332148 Timing requirements not met
System (2)
Processing (116)
100%
00:01:29
22°C
Kabut
Search
04:50
21/05/2024


Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part4
part4.v
Compilation Report - part4
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming file
Timing Analysis
EDA Netlist Writer
Edit Settings
Type ID Message
332123 Deriving Clock Uncertainty. Please refer to report_sdc in the Timing Analyzer to see clock uncertainties.
332148 Timing requirements not met
System (2)
Processing (116)
100%
00:01:29
22°C
Kabut
Search
04:51
21/05/2024


Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part4
part4.v
Compilation Report - part4
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming file
Timing Analysis
EDA Netlist Writer
Edit Settings
Find...
Find Next
Type ID Message
332123 Deriving Clock Uncertainty. Please refer to report_sdc in the Timing Analyzer to see clock uncertainties.
332148 Timing requirements not met
System (2)
Processing (116)
100%
00:01:29
22°C
Kabut
Search
04:51
21/05/2024


Quartus Prime Lite Edition - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
MAX 10: 10M50DAF484C7G
part4
part4.v
Compilation Report - part4
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming file
Timing Analysis
EDA Netlist Writer
Edit Settings
Find...
Find Next
Type ID Message
332123 Deriving Clock Uncertainty. Please refer to report_sdc in the Timing Analyzer to see clock uncertainties.
332148 Timing requirements not met
System (2)
Processing (116)
100%
00:01:29
22°C
Kabut
Search
04:51
21/05/2024


Pin Planner - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4
File Edit View Processing Tools Window Help
Top View - Wire Bond
MAX 10 - 10M50DAF484C7G
Pin Legend
Report not available
Tasks
Early Pin Planning
Node Name Direction Location I/O Bank VREF Group I/O Standard Reserved Slew Rate
CLOCK_50 Input PIN_P11 3 B3_N0 PIN_P11 3.3-V LVTTL 8mA (default)
KEY[1] Input PIN_A7 7 B7_N0 PIN_A7 3.3 V ...rigger 8mA (default)
KEY[0] Input PIN_B8 7 B7_N0 PIN_B8 3.3 V ...rigger 8mA (default)
LEDR[9] Output PIN_B11 7 B7_N0 PIN_B11 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[8] Output PIN_A11 7 B7_N0 PIN_A11 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[7] Output PIN_D14 7 B7_N0 PIN_D14 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[6] Output PIN_E14 7 B7_N0 PIN_E14 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[5] Output PIN_C13 7 B7_N0 PIN_C13 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[4] Output PIN_D13 7 B7_N0 PIN_D13 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[3] Output PIN_B10 7 B7_N0 PIN_B10 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[2] Output PIN_A10 7 B7_N0 PIN_A10 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[1] Output PIN_A9 7 B7_N0 PIN_A9 3.3-V LVTTL 8mA (default) 2 (default)
LEDR[0] Output PIN_A8 7 B7_N0 PIN_A8 3.3-V LVTTL 8mA (default) 2 (default)
SW[2] Input PIN_D12 7 B7_N0 PIN_D12 3.3-V LVTTL 8mA (default)
SW[1] Input PIN_C11 7 B7_N0 PIN_C11 3.3-V LVTTL 8mA (default)
SW[0] Input PIN_C10 7 B7_N0 PIN_C10 3.3-V LVTTL 8mA (default)


Programmer - D:/ASPRAK SISDIG/ftp.intel.com/ftp.intel.com/labs/Digital_Logic/Verilog_lab_source_files/Exercise7/solutions/DE10-Lite/part4.Verilog/part4 - part4 - [part4.cdf]
File Edit View Processing Tools Window Help
Hardware Setup... USB-Blaster [USB-0]
Mode: JTAG
Progress: 100% (Successful)
Enable real-time ISP to allow background programming when available
Start
Stop
Auto Detec
Delete
Add File...
Change File
Save File
Add Device
Up
Down
File Device Checksum Usercode Program/ Configure Verify Blank-Check Examine Security Bit Erase ISP CLAMP
part4.sof 10M50DAF484 00278A2B 00278A2B
TDI
TDO
10M50DAF484

- **Hasil compile bagian IV di FPGA :**