

MODUL 8
IMPLEMENTASI RANGKAIAN DIGITAL PEWAKTU DAN JAM WAKTU NYATA DI
FPGA MENGGUNAKAN VERILOG-HDL

1.1 Tujuan Praktikum Modul

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Praktikan dapat mengetahui dan memahami konsep dasar dari rangkaian digital pewaktu dan jam waktu nyata.
2. Praktikan dapat memahami fungsi dan prinsip kerja pencacah (counter) pada FPGA.
3. Praktikan dapat mengimplementasikan rangkaian digital pewaktu dan jam waktu nyata menggunakan bahasa pemrograman Verilog.

1.2 Dasar Teori Praktikum Modul 1**1.2.1 Real-Time Clock (RTC)**

Real-time Clock (RTC) adalah sebuah komponen dalam sistem komputer atau perangkat digital yang bertugas menjaga waktu secara akurat dan terus-menerus, bahkan saat daya utama sistem dimatikan. RTC menggunakan osilator kristal sebagai sumber waktu dasar yang stabil dan akurat. Fungsi utamanya adalah memberikan informasi waktu dalam format detik, menit, jam, dan komponen waktu lainnya.

Pada DE10-Lite FPGA, Real-time Clock (RTC) dapat diimplementasikan menggunakan rangkaian digital yang memanfaatkan sumber waktu dasar seperti osilator kristal dan modul RTC yang telah tersedia di dalam perangkat FPGA. RTC pada DE10-Lite FPGA berfungsi untuk menyediakan waktu yang akurat dan terus-menerus, bahkan ketika daya FPGA dimatikan atau dipulihkan.

RTC pada DE10-Lite FPGA dapat berfungsi sebagai penyedia waktu real-time yang akurat, memberikan informasi waktu dalam format jam, menit, dan detik. Ini memungkinkan sistem untuk memantau waktu secara terus-menerus dan melakukan tindakan berdasarkan waktu yang tepat.

1.2.2 Rangkaian Sekuensial

Rangkaian sekuensial adalah rangkaian logika yang outputnya dipengaruhi oleh input saat ini dan output sebelumnya. Rangkaian sekuensial dapat dikatakan juga sebagai rangkaian logika yang bekerja berdasarkan urutan waktu.



Gambar 12. 1 Rangkaian Sekuensial

Seperti dilihat dari gambar rangkaian di atas, ciri dari sistem rangkaian logika sekuensial adalah adanya jalur umpan balik (feedback). Adapun contoh dari rangkaian sekuensial yaitu flip-flop, counter, register, dan detektor sekuensial.

1.2.3 RS Flip-Flop

RS flip-flop adalah bentuk dasar dari rangkaian logika sekuensial yang digunakan dalam desain perangkat keras (hardware). Flip-flop ini memiliki dua input, yaitu S (Set) dan R (Reset), serta dua output, yaitu Q dan Q' (invers dari Q). Sinyal clock tidak diperlukan dalam RS flip-flop.

Pada RS flip-flop, input S (Set) digunakan untuk mengatur output Q menjadi logika 1 (set) sedangkan input R (Reset) digunakan untuk mengatur output Q menjadi logika 0 (reset). Kedua input tersebut beroperasi secara saling berlawanan; artinya, ketika salah satu input diaktifkan, input lainnya harus dinonaktifkan untuk mencegah kondisi yang tidak terdefinisi atau konflik logika.

S (Set)	R (Reset)	Q (Output)	Q' (Output Inverted)
0	0	Q	Q'
0	1	0	1
1	0	1	0
1	1	Undefined	Undefined

Berikut adalah beberapa karakteristik utama dari RS flip-flop:

- Saat S = 0 dan R = 0, keadaan flip-flop tetap.
- Saat S = 0 dan R = 1, flip-flop akan direset, membuat output Q menjadi 0.
- Saat S = 1 dan R = 0, flip-flop akan diset, membuat output Q menjadi 1.
- Saat S = 1 dan R = 1, keadaan flip-flop menjadi tidak terdefinisi karena dapat menyebabkan konflik logika.

Modul Praktikum

1.3 Lembar Kegiatan Praktikum Modul 1

1.3.1 Alat dan Bahan

- Software *Quartus II*
- ModelSim*
- Mouse
- Laptop

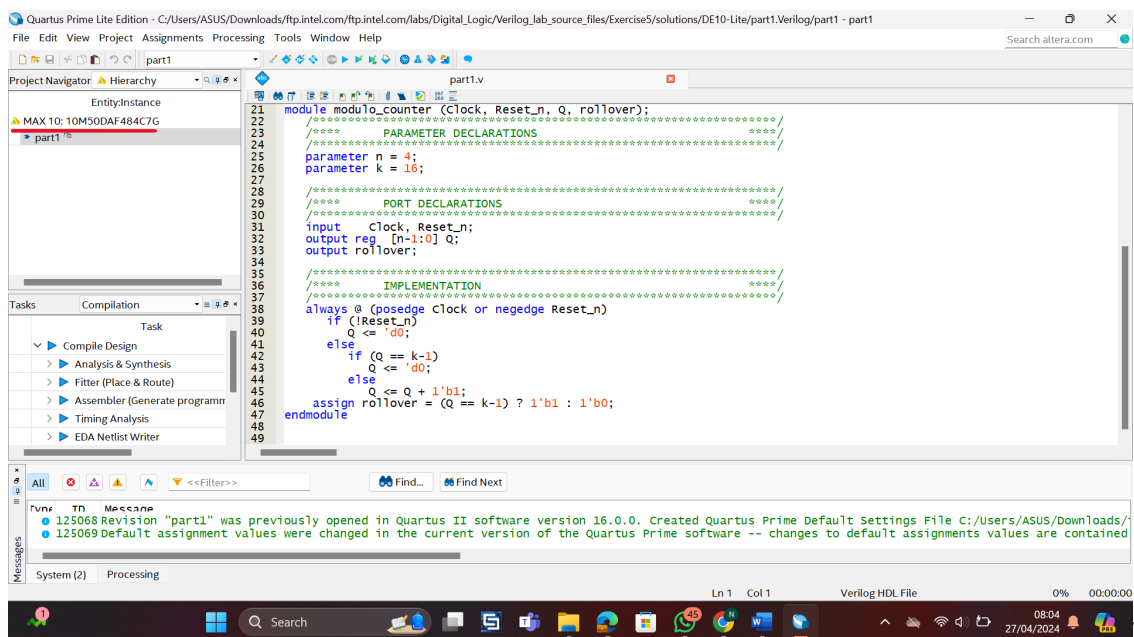
1.3.2 Langkah Praktikum Modul Quartus II

PERCOBAAN I

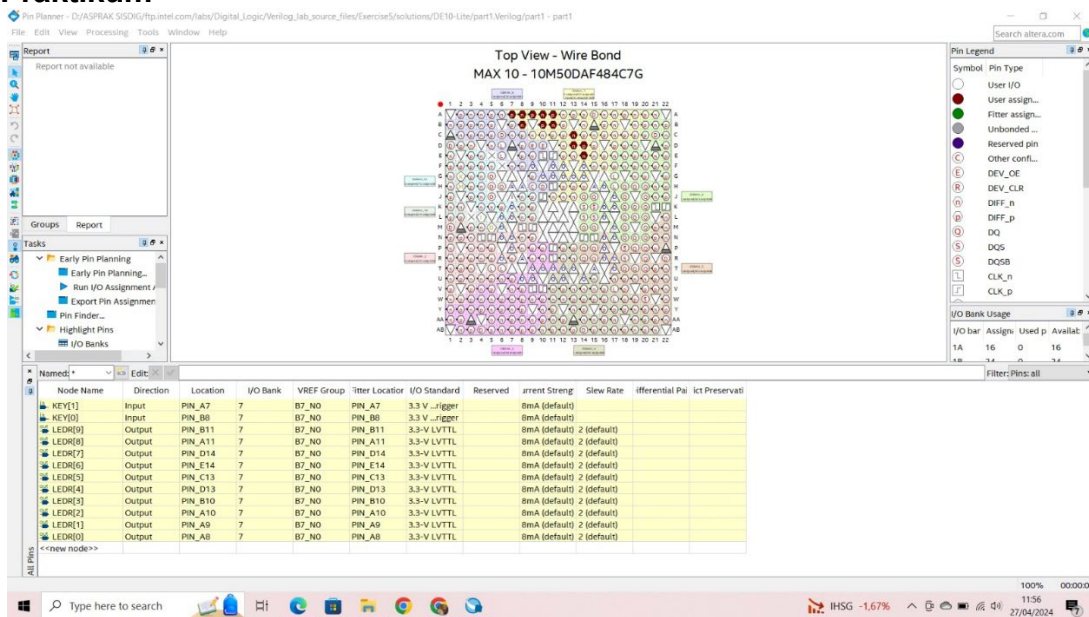
Buat pencacah modulo-k dengan memodifikasi desain pencacah 8-bit agar memuat parameter tambahan. Itu penghitung harus menghitung dari 0 hingga $k-1$. Ketika penghitung mencapai nilai $k-1$, maka nilai penghitung berikutnya seharusnya menjadi 0. Sertakan keluaran dari penghitung yang disebut rollover dan atur keluaran ini ke 1 dalam siklus jam tempat hitungannya nilainya sama dengan $k-1$.

Lakukan langkah-langkah berikut:

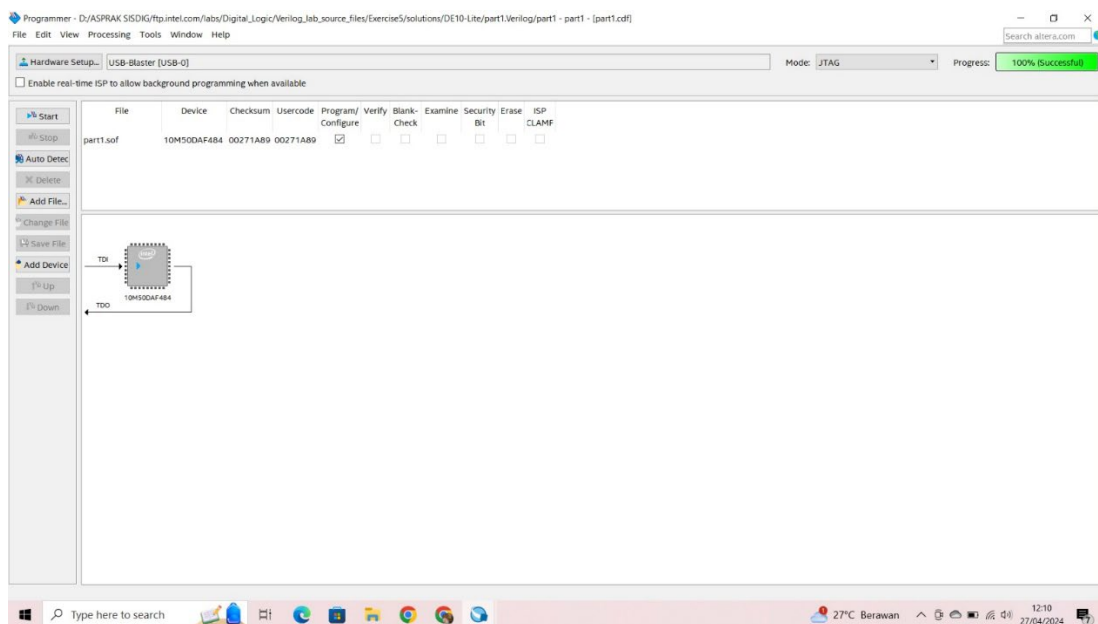
- Buat proyek Quartus® baru yang akan digunakan untuk mengimplementasikan sirkuit yang diinginkan pada board DE-series Anda.
- Tulis file Verilog yang menentukan rangkaian untuk $k = 20$, dan nilai n yang sesuai. Sirkuit Anda harus digunakan tombol tekan KEY0 sebagai reset asinkron dan KEY1 sebagai input jam manual. Isi konter harus ditampilkan pada lampu LEDR merah. Juga menampilkan sinyal rollover pada salah satu lampu LEDR.



Modul Praktikum

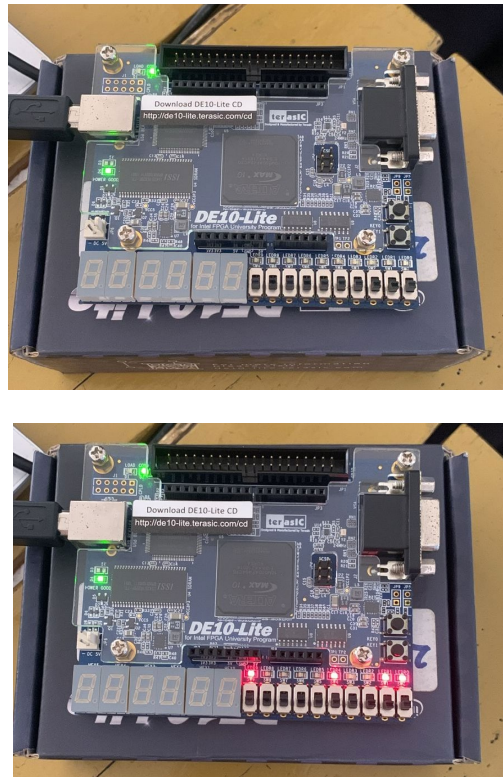


3. Sertakan file Verilog dalam proyek Anda dan kompilasi sirkuitnya.
4. Simulasikan rangkaian yang dirancang untuk memverifikasi fungsinya.



5. Buatlah penetapan pin yang diperlukan untuk mengimplementasikan rangkaian pada papan seri DE Anda, dan kompilasi sirkuit.
6. Pastikan sirkuit Anda berfungsi dengan benar dengan mengamati lampunya

Modul Praktikum



Gambar 1 Hasil dari percobaan 1.

PERCOBAAN II

Menggunakan penghitung modulo dari Bagian I sebagai subsirkuit, terapkan penghitung BCD 3 digit (petunjuk: gunakan beberapa counter, tidak hanya satu). Tampilkan isi penghitung pada tampilan 7-segmen, HEX2-0. Hubungkan semua dari penghitung di sirkuit Anda ke sinyal jam 50 MHz di papan seri DE Anda, dan buat penghitung BCD kenaikan pada interval satu detik. Gunakan sakelar tombol tekan KEY0 untuk mereset penghitung BCD ke 0.

```
1 // 3-digit BCD counter
2 module part2 (CLOCK_50, KEY, HEX2, HEX1, HEX0);
3     // PORT DECLARATIONS
4     //*****
5     //*****
6     input CLOCK_50;
7     input [0:0] KEY;
8     output [0:6] HEX2, HEX1, HEX0;
9
10    //*****
11    //***** WIRE DECLARATIONS *****
12    //*****
13    wire one_second_enable;
14    wire [3:0] one_wires, ten_wires, hundred_wires;
15    wire roll_ones, roll_tens;
16
17    //*****
18    //***** IMPLEMENTATION *****
19    //*****
20    module_counter_0 slow_clock (.clock(CLOCK_50), .reset_n(KEY[0]),
21                                .enable(1'b1), .rollOver(one_second_enable));
22    defparam slow_clock.n = 27;
23    defparam slow_clock.k = 50000000;
24
25    // The three digits
26    module_counter_1 ones (.clock(CLOCK_50), .reset_n(KEY[0]),
27                           .enable(one_second_enable), .rollOver(roll_ones));
28    defparam ones.n = 4;
29    defparam ones.k = 10;
```

Messages

System (2) Processing

125068 Revision "part2" was previously opened in Quartus II software version 16.0.0. Created Quartus Prime Default Settings File C:/Users/ASUS/Downloads/

125069 Default assignment values were changed in the current version of the Quartus Prime software -- changes to default assignments values are contained

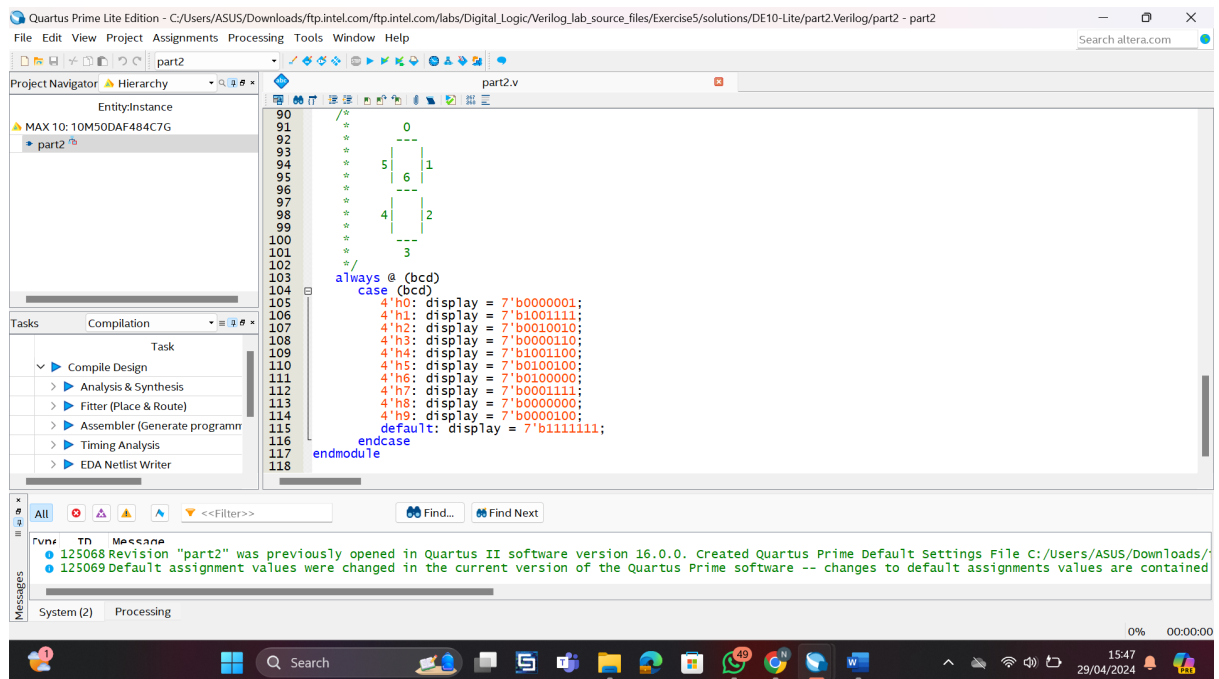
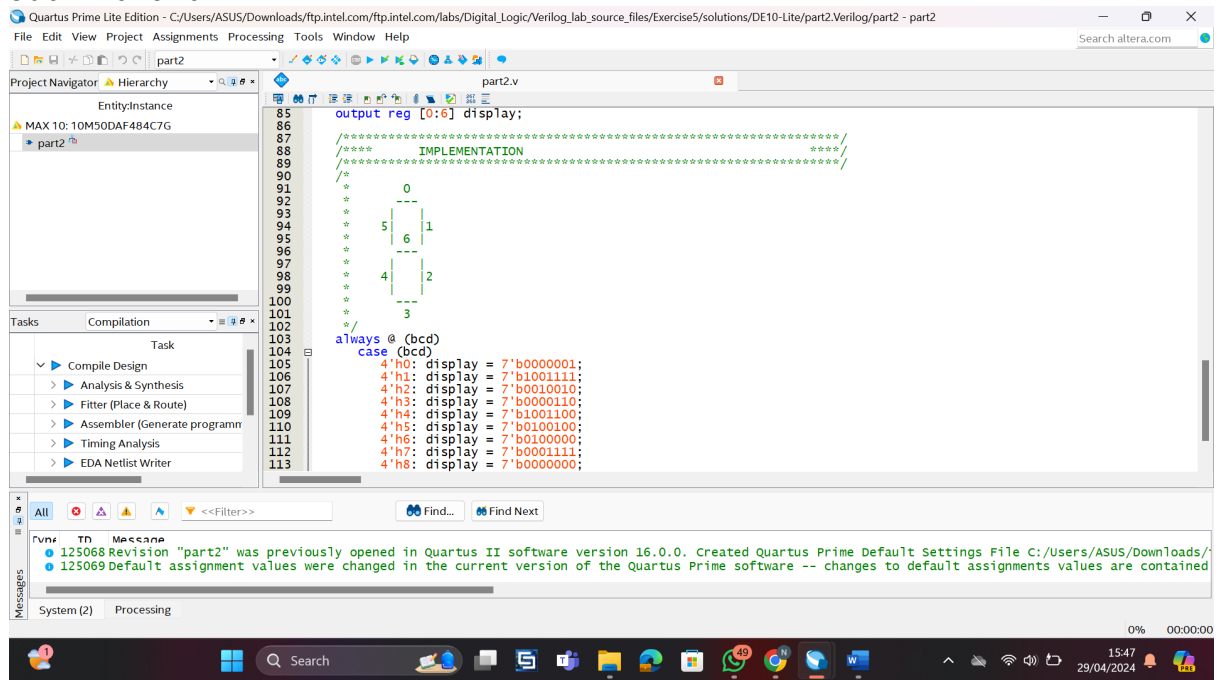
Modul Praktikum

The image displays two screenshots of the Quartus Prime Lite Edition software interface, showing the design and implementation of a digital logic circuit.

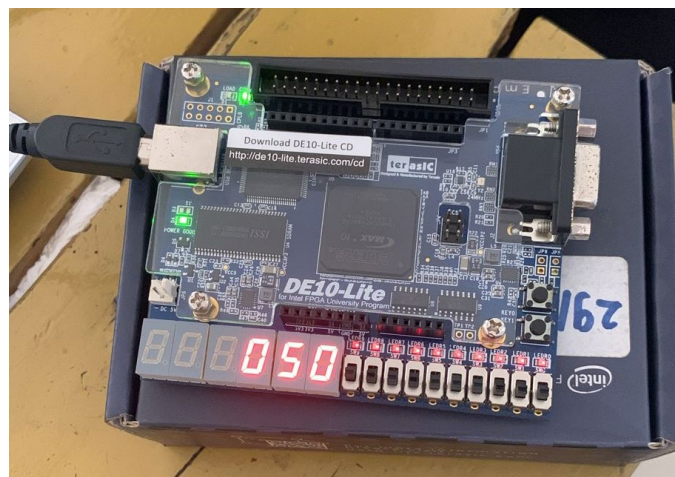
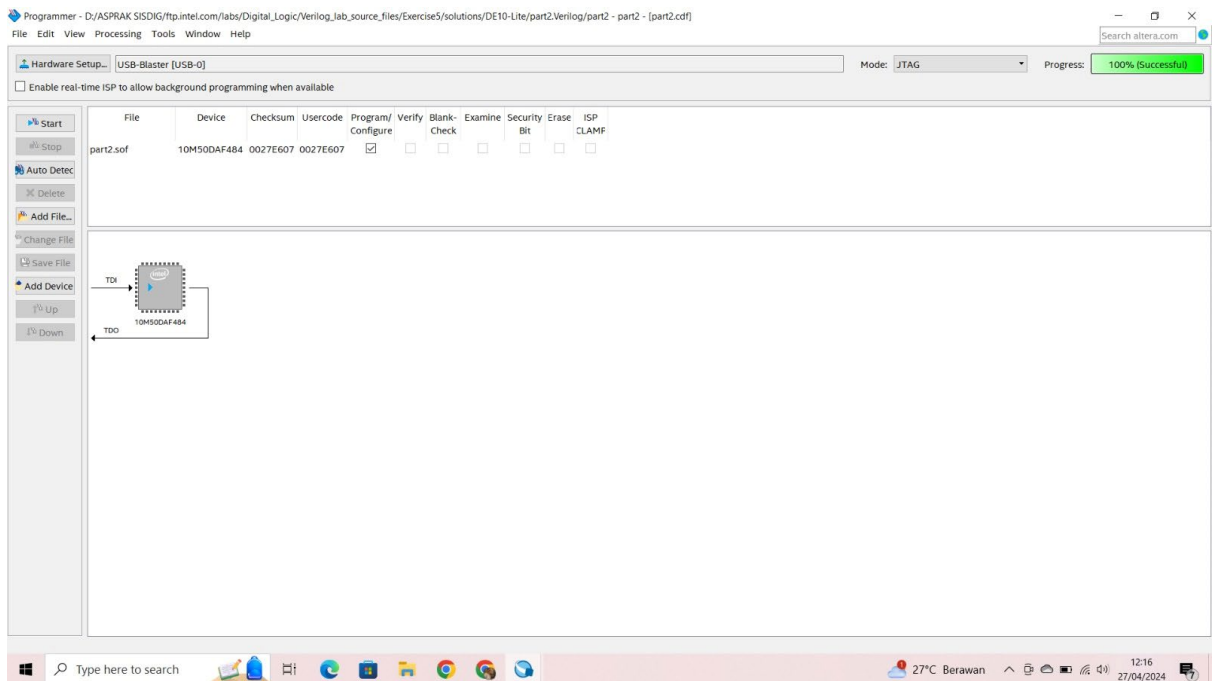
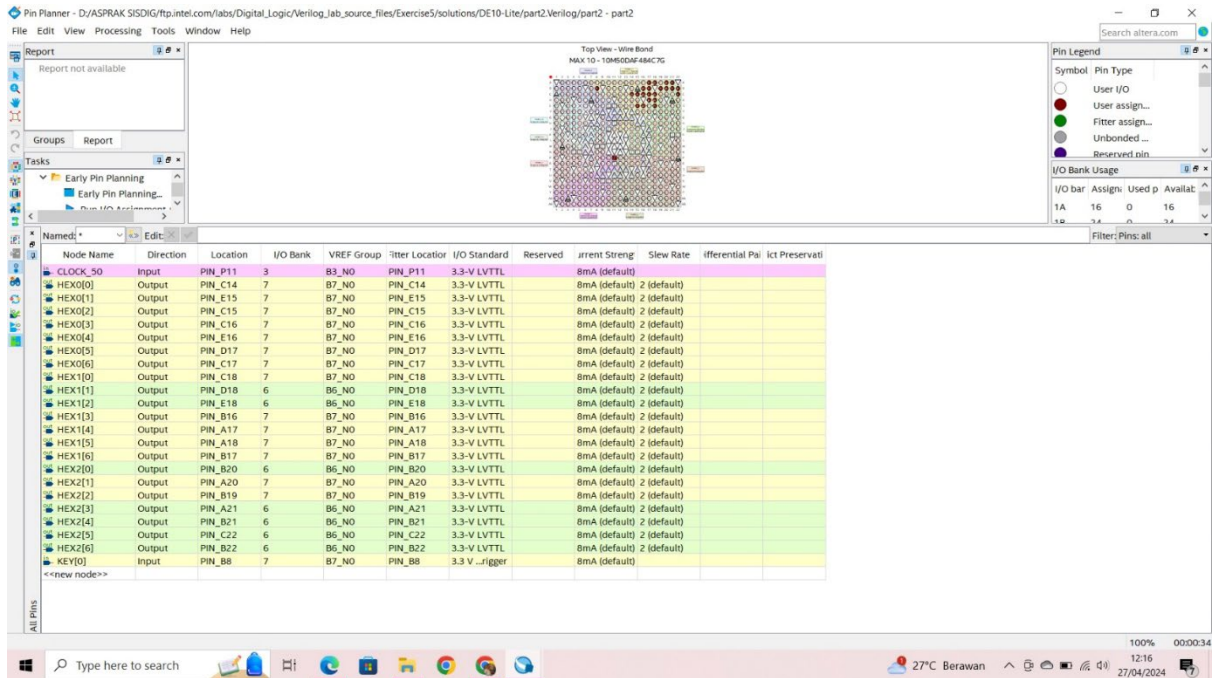
Top Screenshot: The main editor window shows the Verilog code for a module named `part2.v`. The code defines two counters, `ones` and `hundreds`, and a 7-segment display output. The `ones` counter is a modulo-10 counter, and the `hundreds` counter is a modulo-100 counter. The output digits are displayed on 7-segment displays. The code includes parameter declarations for `n` and `k`, and port declarations for `Clock`, `Reset_n`, `enable`, `Q`, and `rollover`.

Bottom Screenshot: The main editor window shows the Verilog code for a module named `part2.v`. The code defines a counter `Q` and a 7-segment display output. The counter `Q` is a modulo-10 counter, and the 7-segment display output is connected to the counter's output. The code includes parameter declarations for `n` and `k`, and port declarations for `Clock`, `Reset_n`, `enable`, `Q`, and `rollover`.

Modul Praktikum



Modul Praktikum

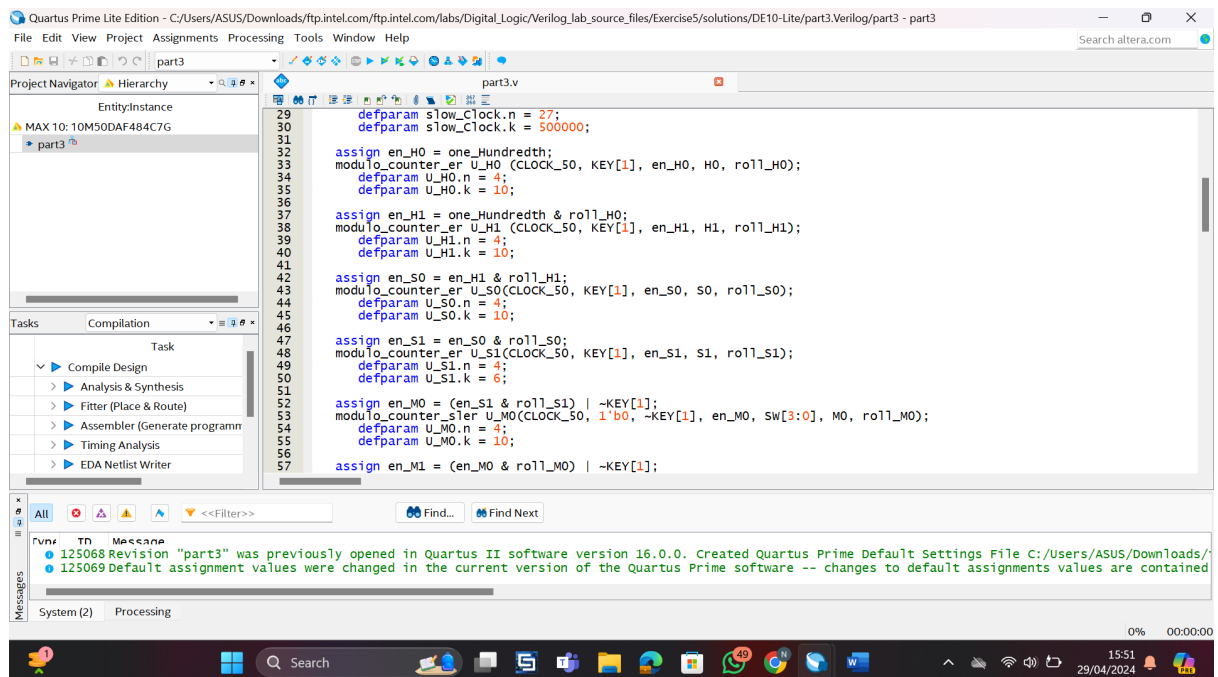
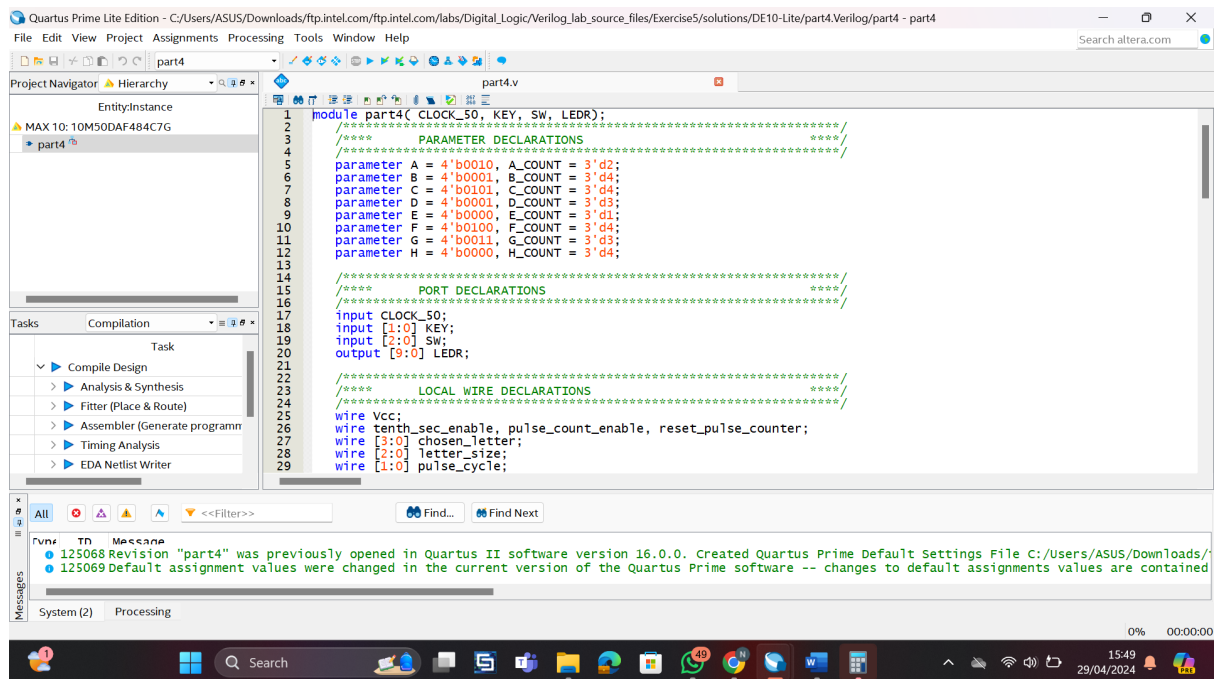


gambar 2 Hasil dari percobaan 2.

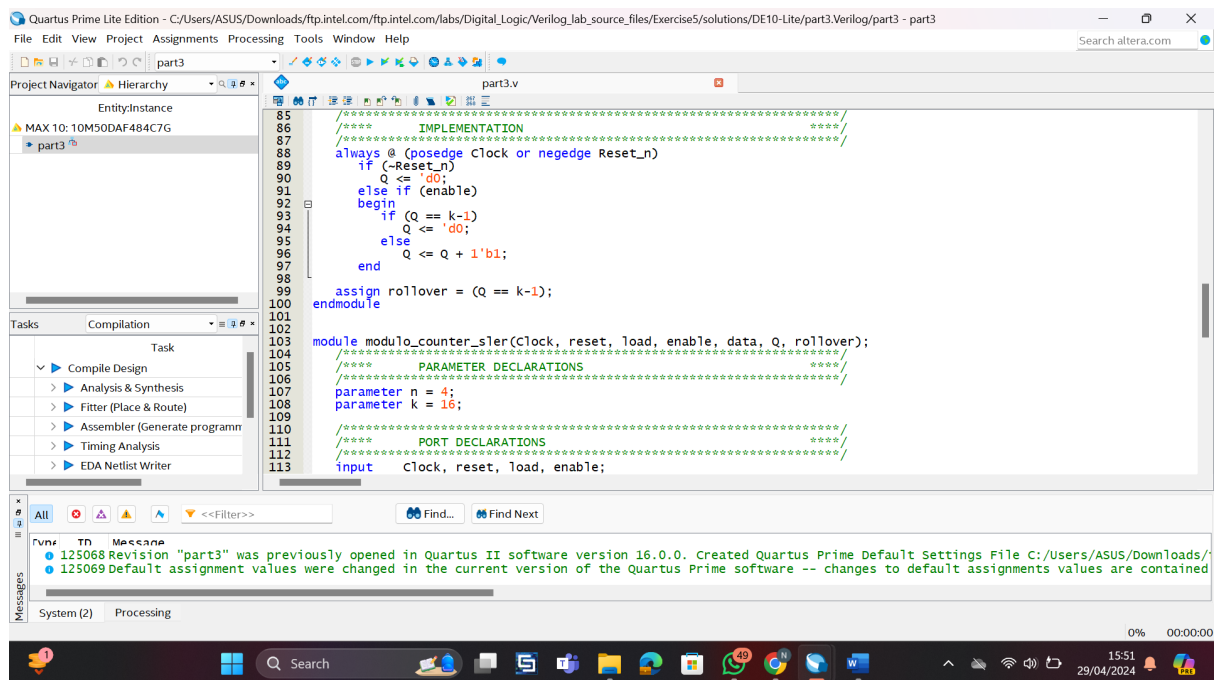
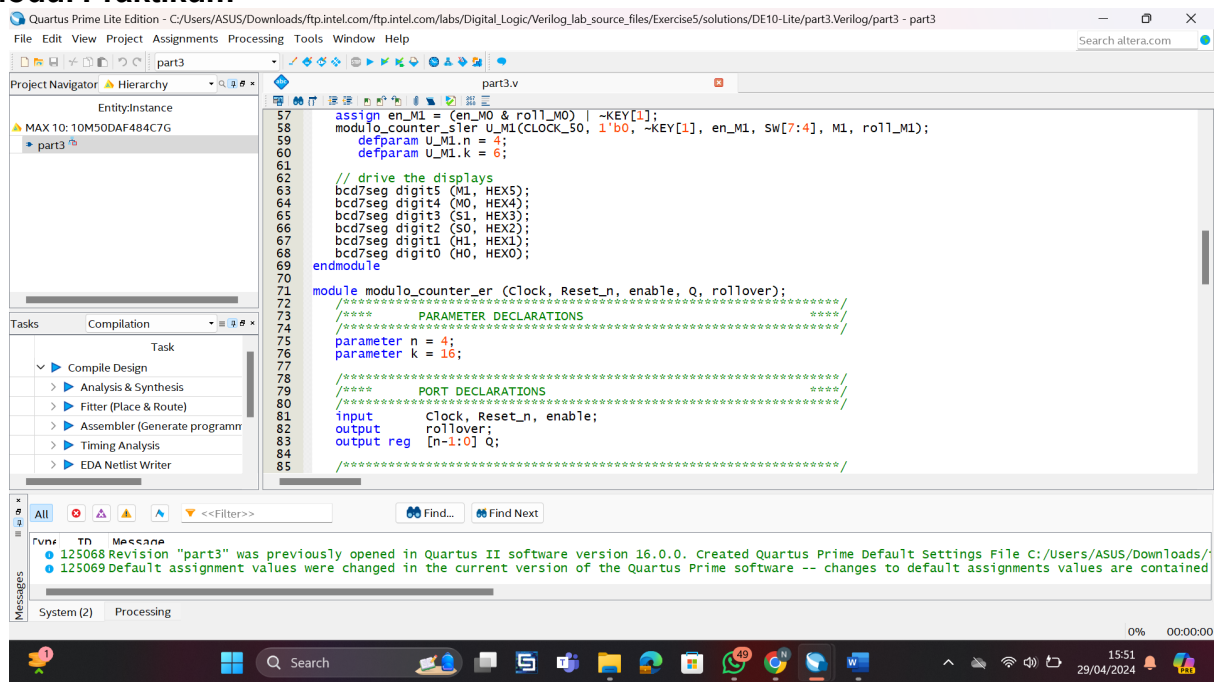
Modul Praktikum

PERCOBAAN III

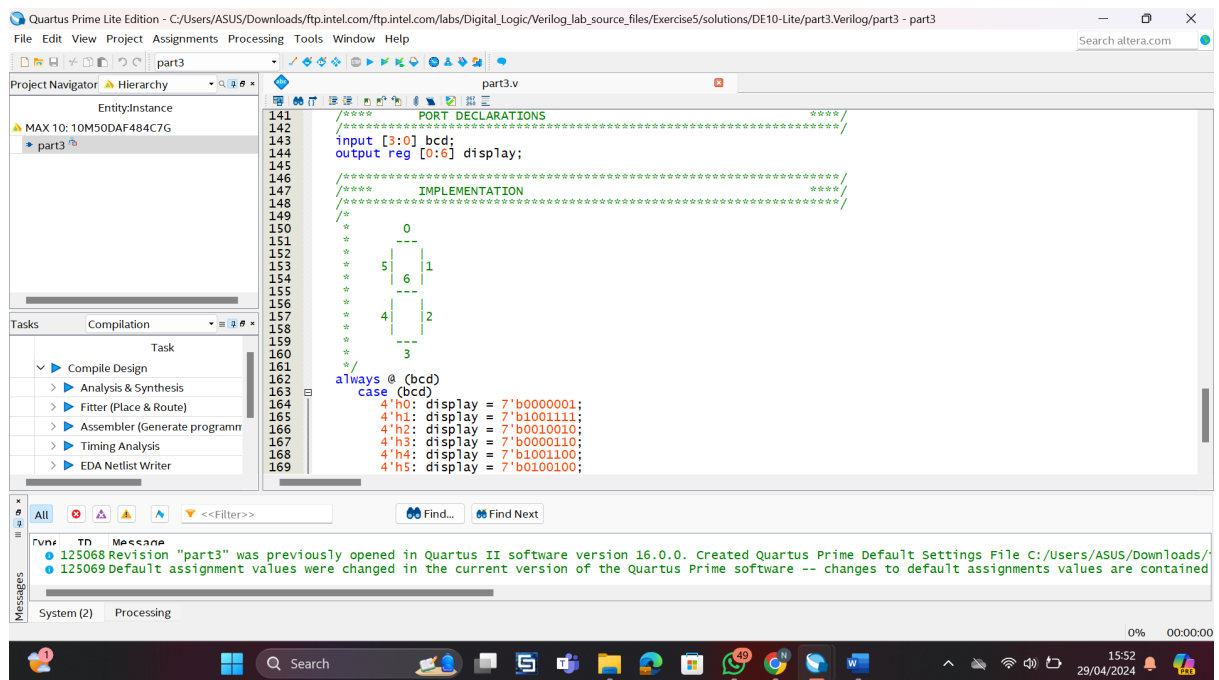
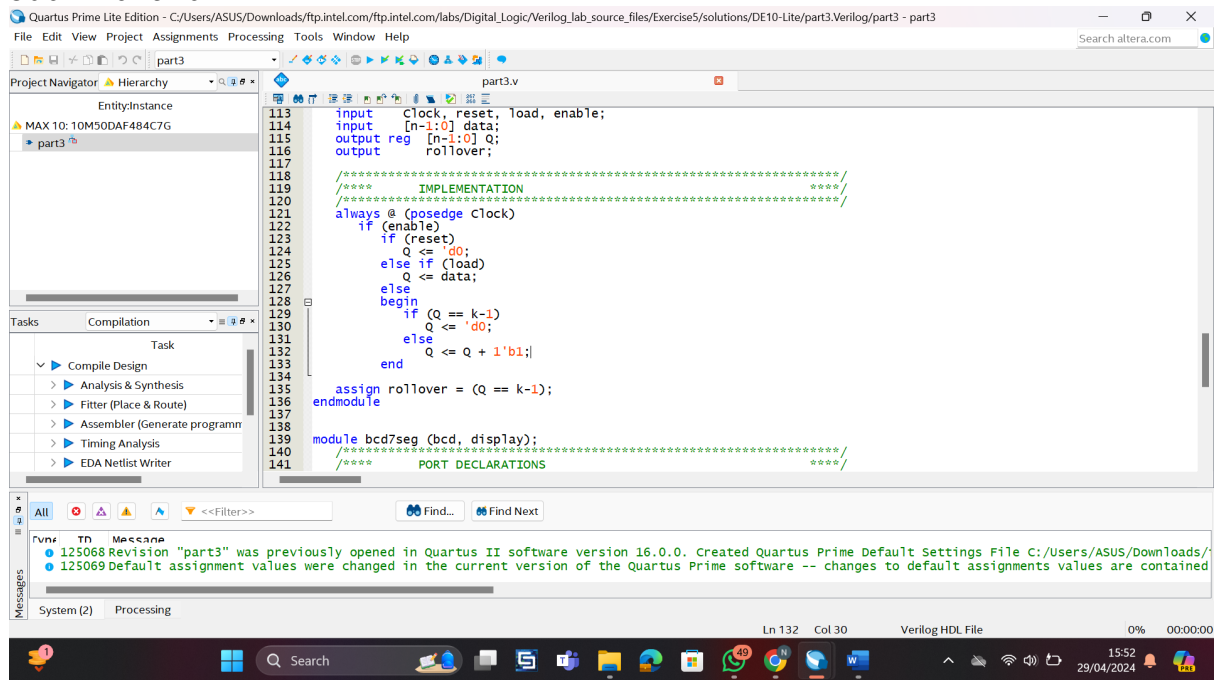
Rancang dan terapkan sirkuit pada papan seri DE Anda yang berfungsi sebagai jam waktu nyata. Ini harus menampilkan menit (dari 0 hingga 59) pada HEX5 – 4, detik (dari 0 hingga 59) pada HEX3 – 2, dan seperseratus detik (dari 0 hingga 99) pada HEX1 – 0. Gunakan sakelar SW7–0 untuk menyetel bagian menit dari waktu yang ditampilkan oleh jam saat KEY1 ditekan. Hentikan jam setiap kali KEY0 ditekan dan lanjutkan jam saat KEY0 dilepaskan.



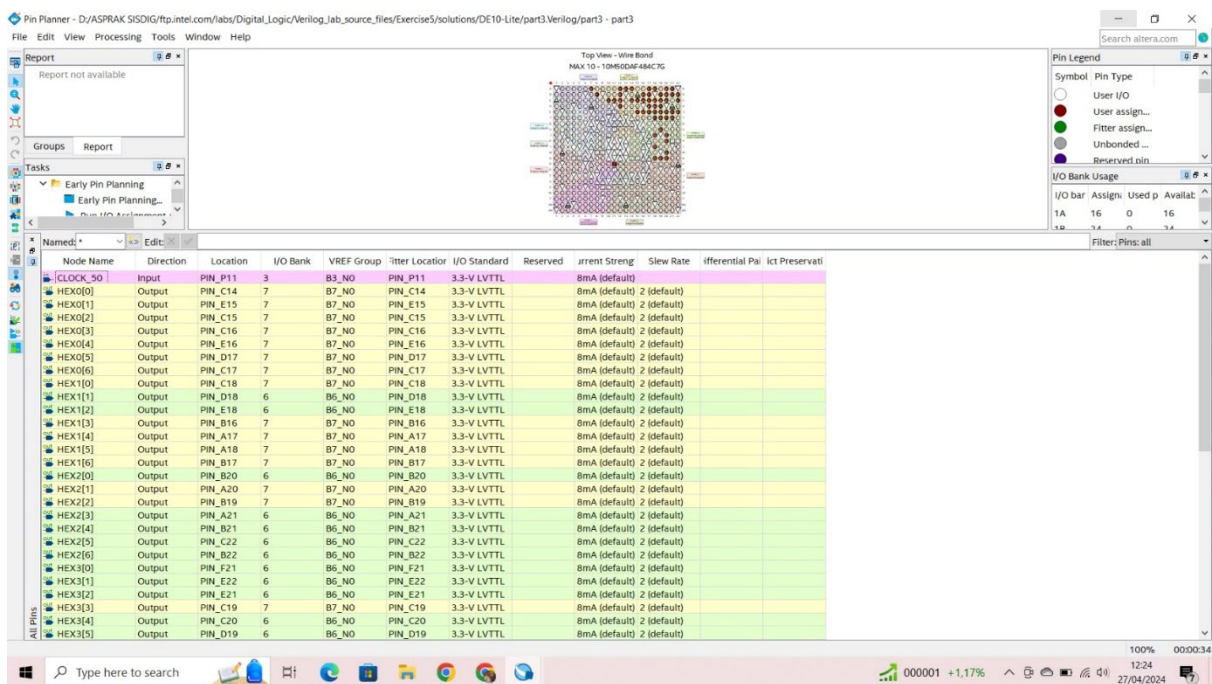
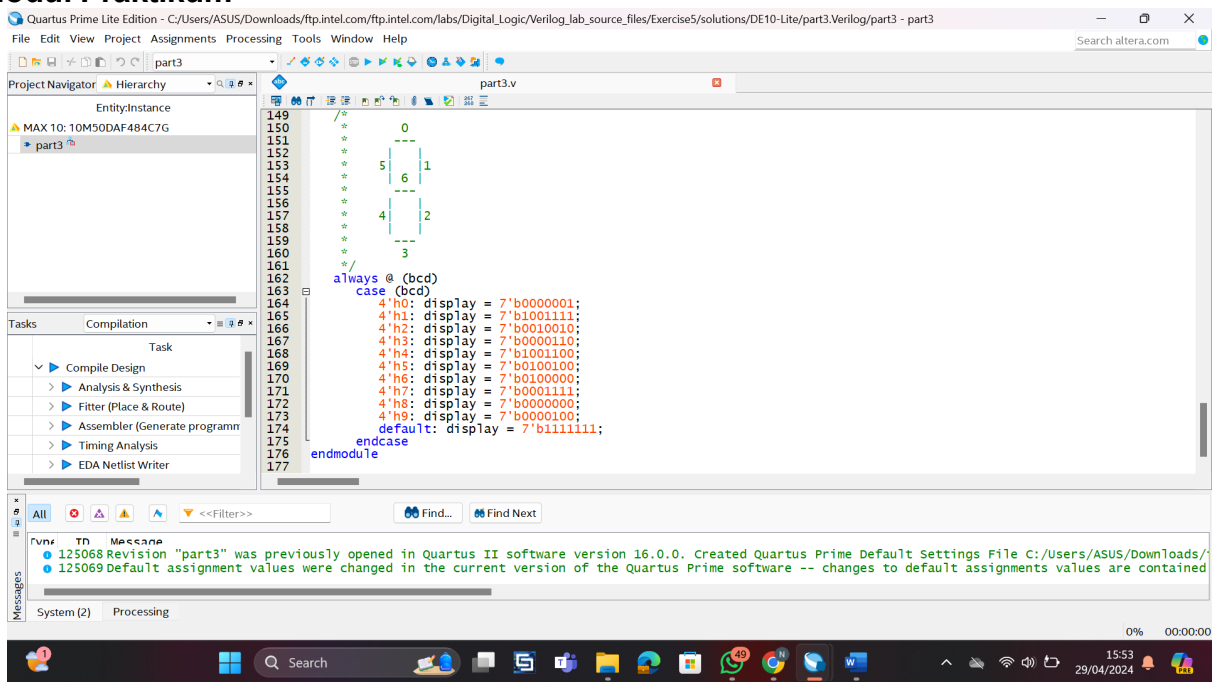
Modul Praktikum



Modul Praktikum



Modul Praktikum



Modul Praktikum

Pin Planner - D:\ASPRAK SISDI\G\tp.intel.com\labs\Digital_Logic\Verilog_lab_source_files\Exercise5\solutions\DE10-Lite\part3.Verilog\part3 - part3

File Edit View Processing Tools Window Help

Report not available

Groups Report

Tasks

Early Pin Planning...

Pin Legend

Symbol Pin Type

User I/O

User assign...

Filter assign...

Unbonded ...

Reserved pin

I/O Bank Usage

I/O bar Assign Used p Availat

1A 16 0 16

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Iter Location	I/O Standard	Reserved	Irrent Streng	Slew Rate	Ifferential Pa	ict Preservati
HEX3[4]	Output	PIN_C20	6	B6_NO	PIN_C20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[5]	Output	PIN_D19	6	B6_NO	PIN_D19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[6]	Output	PIN_E17	6	B6_NO	PIN_E17	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[0]	Output	PIN_F18	6	B6_NO	PIN_F18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[1]	Output	PIN_E20	6	B6_NO	PIN_E20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[2]	Output	PIN_E19	6	B6_NO	PIN_E19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[3]	Output	PIN_J18	6	B6_NO	PIN_J18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[4]	Output	PIN_H19	6	B6_NO	PIN_H19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[5]	Output	PIN_F19	6	B6_NO	PIN_F19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX4[6]	Output	PIN_F20	6	B6_NO	PIN_F20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[0]	Output	PIN_J20	6	B6_NO	PIN_J20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[1]	Output	PIN_K20	6	B6_NO	PIN_K20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[2]	Output	PIN_L18	6	B6_NO	PIN_L18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[3]	Output	PIN_N18	6	B6_NO	PIN_N18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[4]	Output	PIN_M20	6	B6_NO	PIN_M20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[5]	Output	PIN_M19	6	B6_NO	PIN_M19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX5[6]	Output	PIN_N20	6	B6_NO	PIN_N20	3.3-V LVTTTL		8mA (default)	2 (default)		
KEY[1]	Input	PIN_A7	7	B7_NO	PIN_A7	3.3 V -rigger		8mA (default)			
KEY[0]	Input	PIN_B8	7	B7_NO	PIN_B8	3.3 V -rigger		8mA (default)			
SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	3.3-V LVTTTL		8mA (default)			
SW[6]	Input	PIN_A13	7	B7_NO	PIN_A13	3.3-V LVTTTL		8mA (default)			
SW[5]	Input	PIN_B12	7	B7_NO	PIN_B12	3.3-V LVTTTL		8mA (default)			
SW[4]	Input	PIN_A12	7	B7_NO	PIN_A12	3.3-V LVTTTL		8mA (default)			
SW[3]	Input	PIN_C12	7	B7_NO	PIN_C12	3.3-V LVTTTL		8mA (default)			
SW[2]	Input	PIN_D12	7	B7_NO	PIN_D12	3.3-V LVTTTL		8mA (default)			
SW[1]	Input	PIN_C11	7	B7_NO	PIN_C11	3.3-V LVTTTL		8mA (default)			
SW[0]	Input	PIN_C10	7	B7_NO	PIN_C10	3.3-V LVTTTL		8mA (default)			

100% 00:00:34

Type here to search

000001 +1,17%

12:24 27/04/2024

Programmer - D:\ASPRAK SISDI\G\tp.intel.com\labs\Digital_Logic\Verilog_lab_source_files\Exercise5\solutions\DE10-Lite\part3.Verilog\part3 - part3 - [part3.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 Detect

Delete

Add File...

Change File

Save File

Add Device

Up

Down

File	Device	Checksum	Usercode	Program/ Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP
part3.sof	10M50DAF484	0028571A	0028571A	☒	☐	☐	☐	☐	☐

TDI

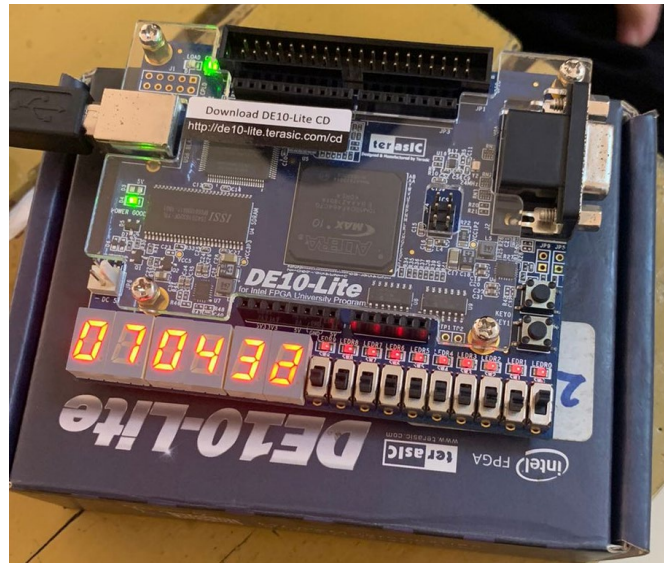
TDO

10M50DAF484

Type here to search

000001 +1,17%

12:23 27/04/2024



gambar 3 Hasil dari percobaan 3.

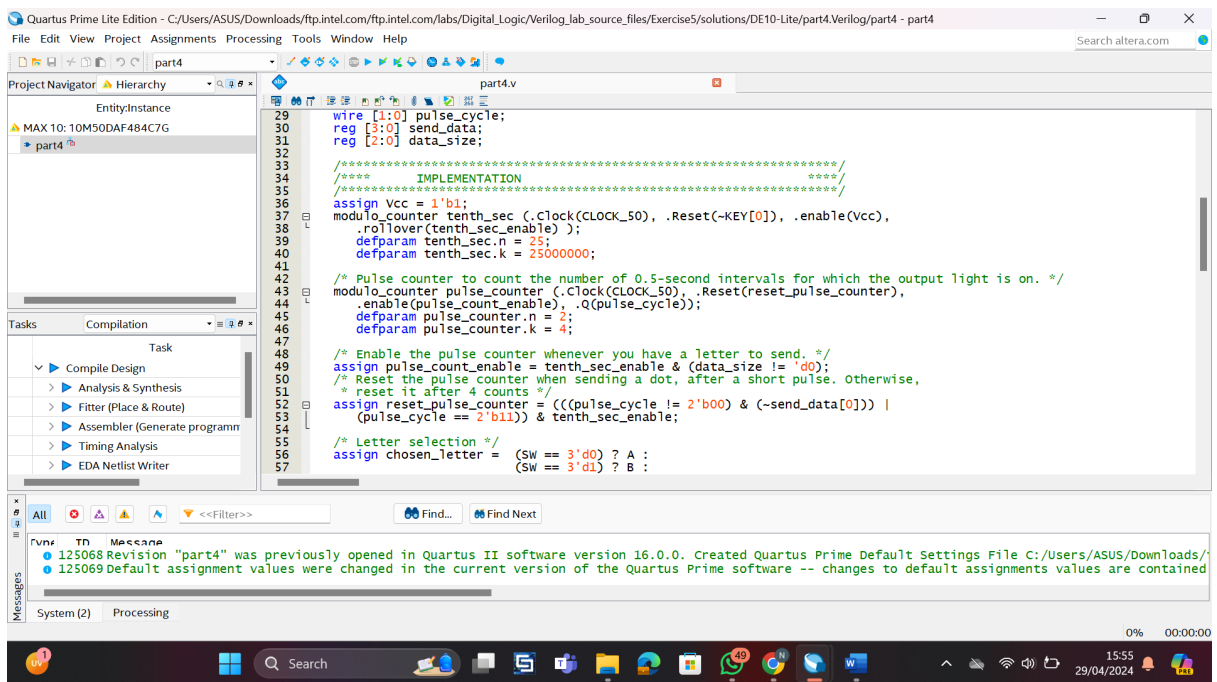
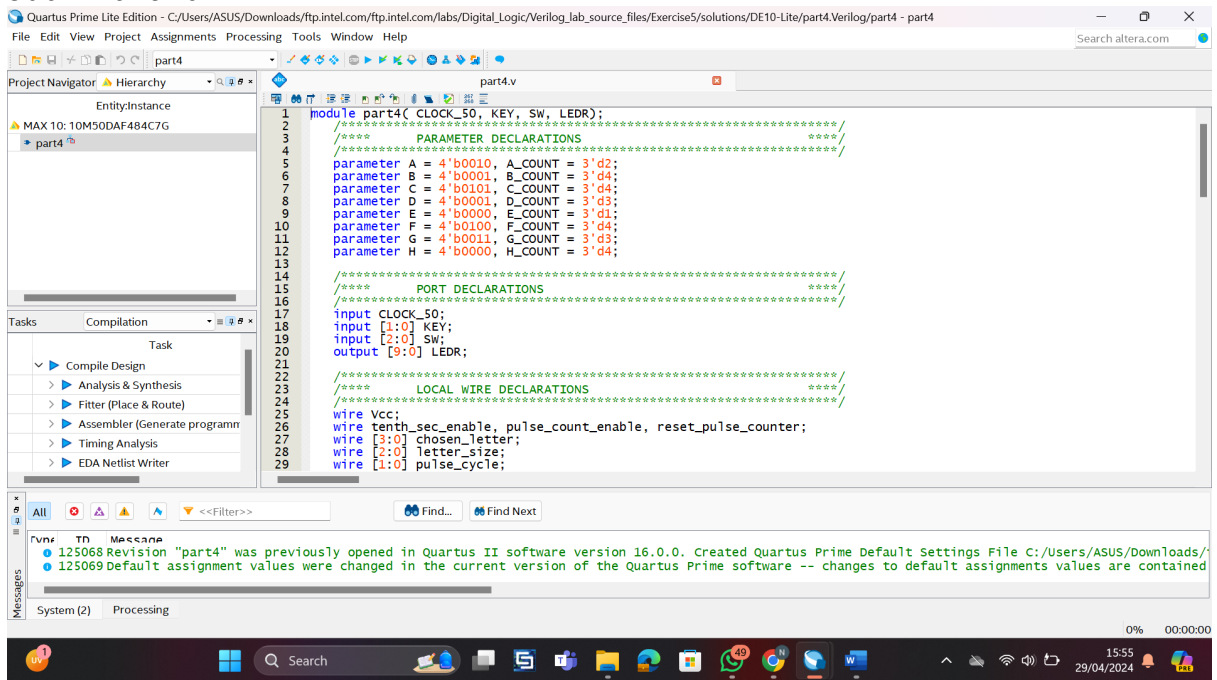
PERCOBAAN IV

Metode awal komunikasi telegraf didasarkan pada kode Morse. Kode ini menggunakan pola pendek dan pulsa panjang untuk mewakili pesan. Setiap huruf direpresentasikan sebagai rangkaian titik (denyut pendek), dan garis (apulsa panjang). Misalnya, delapan huruf pertama alfabet memiliki representasi berikut:

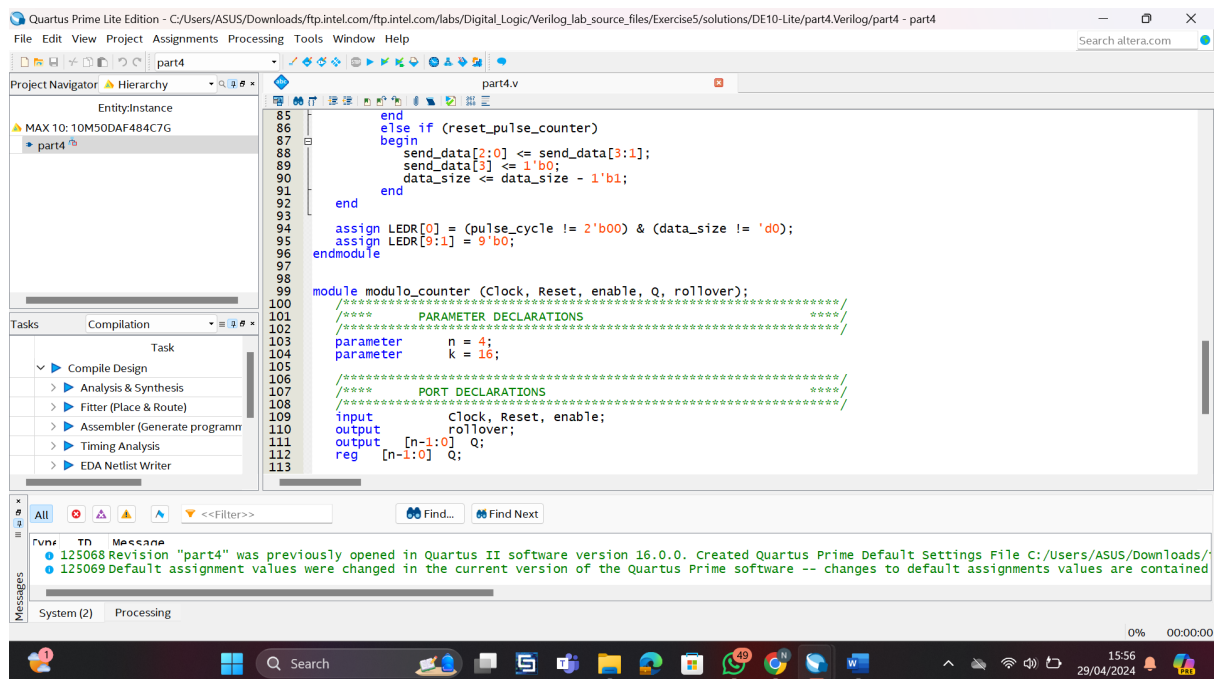
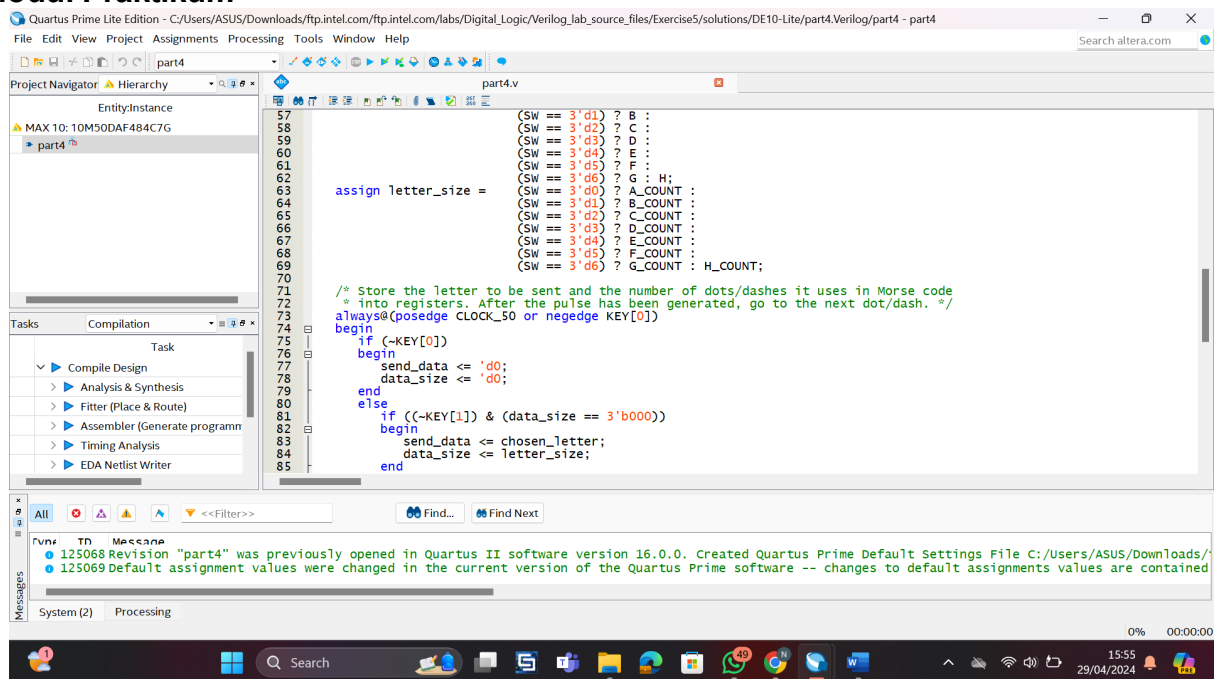
A	• —
B	— • • •
C	— • — •
D	— • •
E	•
F	• • — •
G	— — •
H	• • • •

Rancang dan implementasikan rangkaian yang mengambil masukan salah satu dari delapan huruf pertama alfabet dan menampilkannya Kode morse untuk itu pada LED merah. Sirkuit Anda harus menggunakan sakelar SW2-0 dan tombol tekan KEY1-0 sebagai input. Kapan pengguna menekan KEY1, sirkuit akan menampilkan kode Morse untuk huruf yang ditentukan oleh SW2-0 (000 untuk A, 001 untuk B, dll.), menggunakan pulsa 0,5 detik untuk mewakili titik, dan pulsa 1,5 detik untuk mewakili garis. Tombol Tekan KUNCIO harus berfungsi sebagai reset asinkron. Diagram skema tingkat tinggi dari rangkaian ditunjukkan pada Gambar 2.

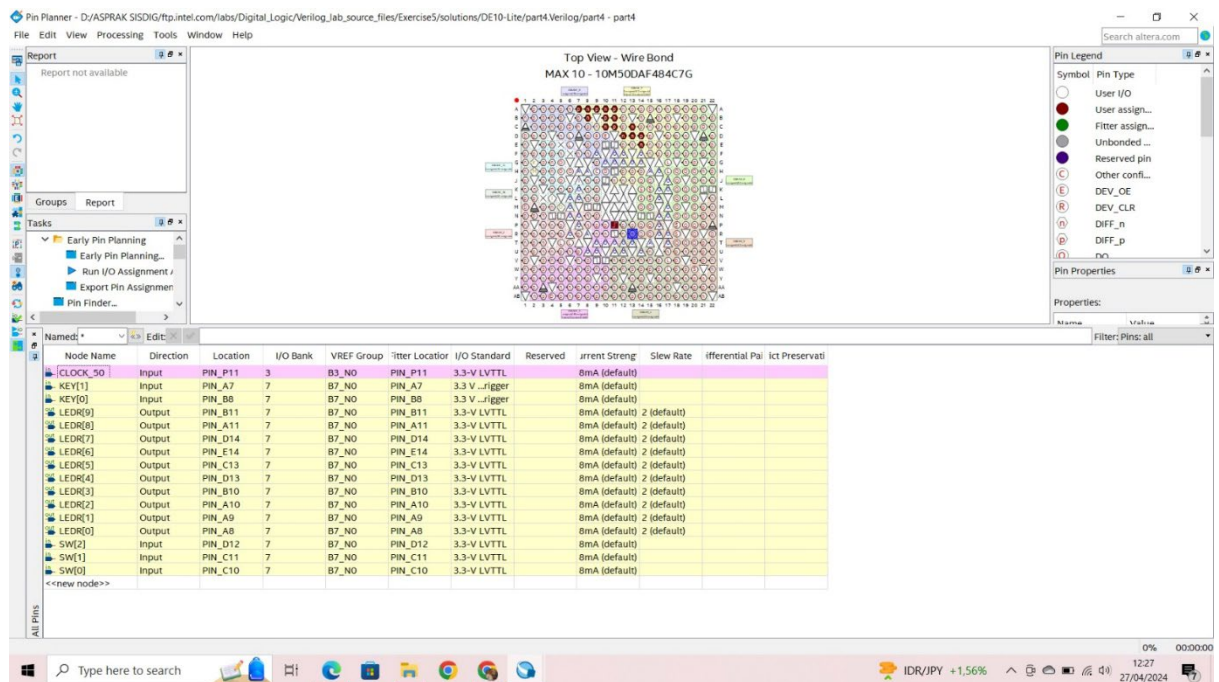
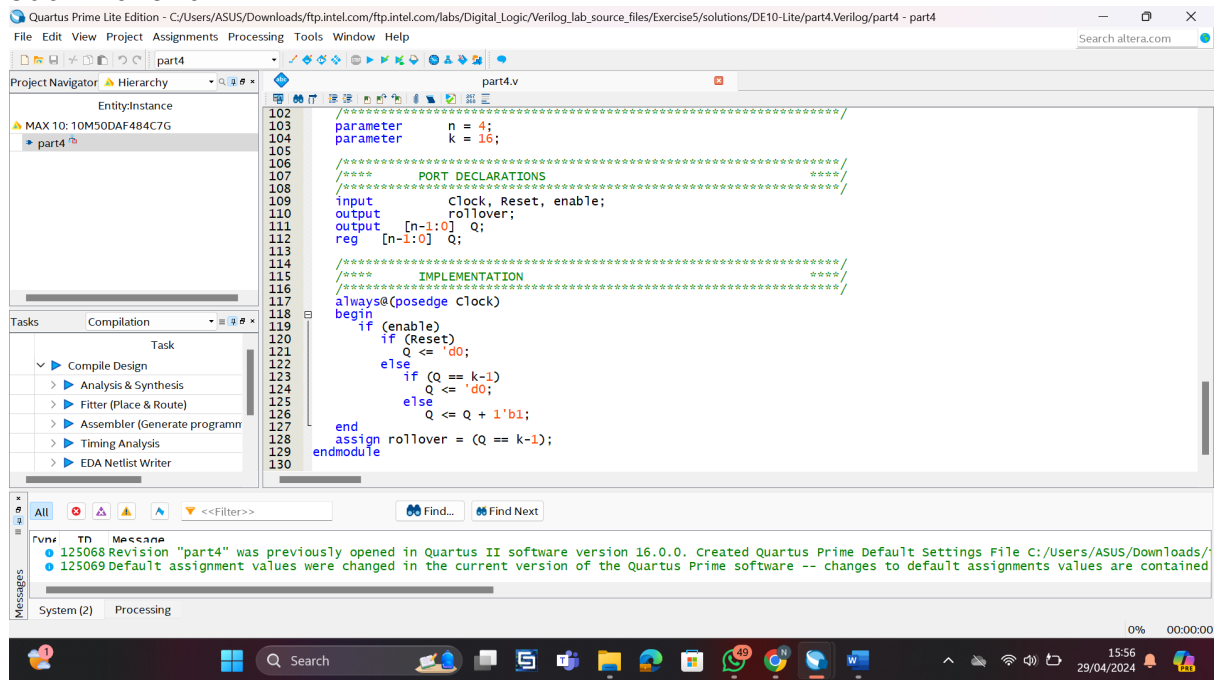
Modul Praktikum



Modul Praktikum

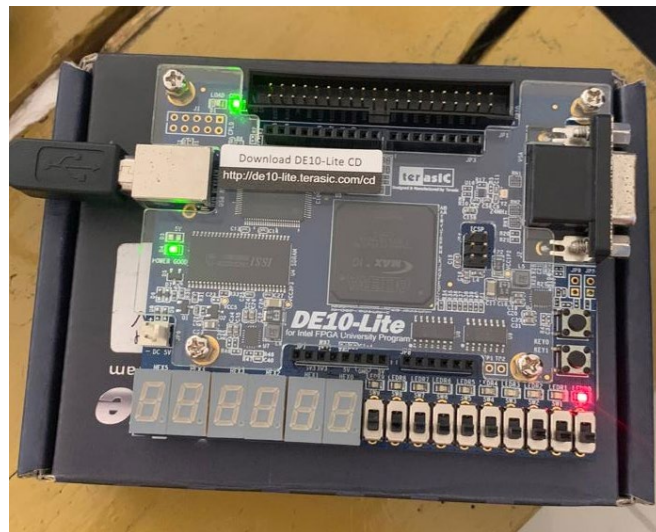


Modul Praktikum



Modul Praktikum

Petunjuk: Gunakan penghitung untuk menghasilkan pulsa 0,5 detik, dan penghitung lain untuk menjaga lampu LEDR0 tetap menyala. 0,5 atau 1,5 detik.



gambar 4 hasil dari percobaan 4

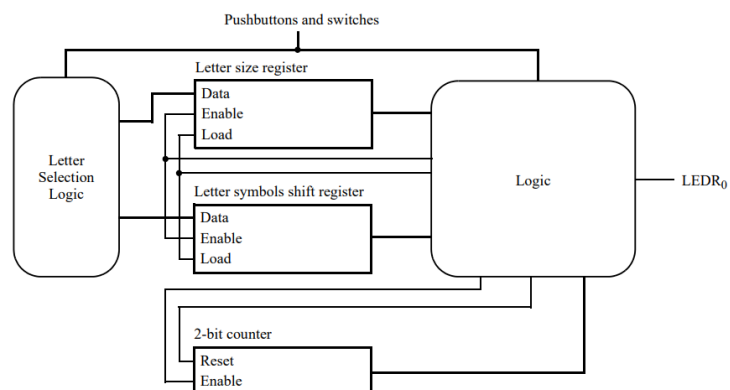


Figure 2: High-level schematic diagram of the circuit for part IV.

SELAMAT MENGERJAKAN