

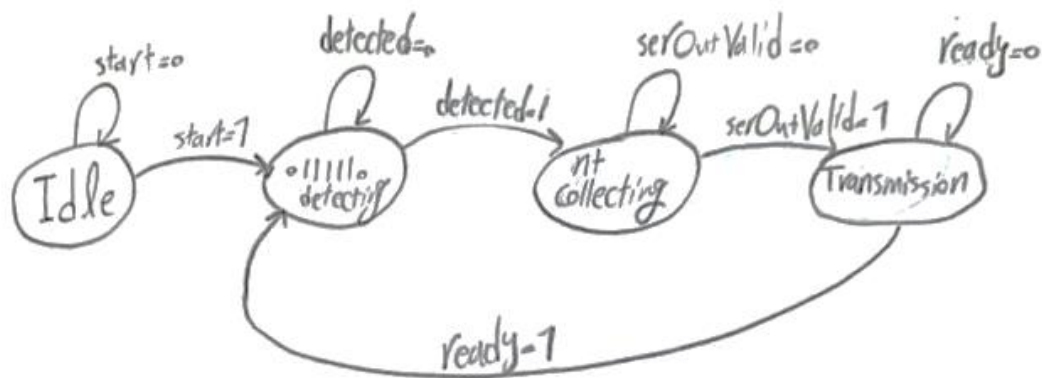
Name : Mostafa Kermaninia

SID : 810101575

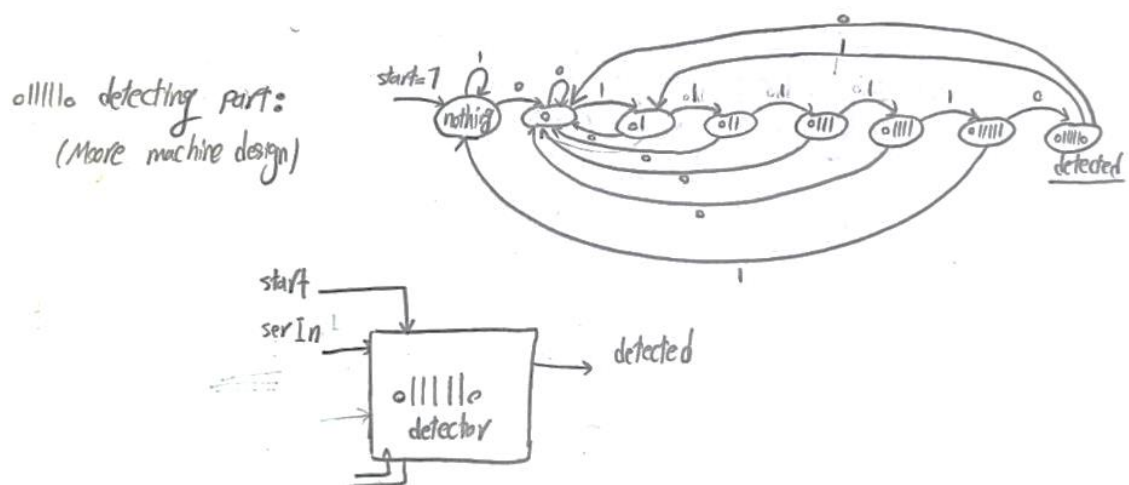
Course name : Introduction to Digital System Design

Course number : 4021810136701

در ابتدا تصویر کلی از کارهایی که قرار است بکنیم می کشم.

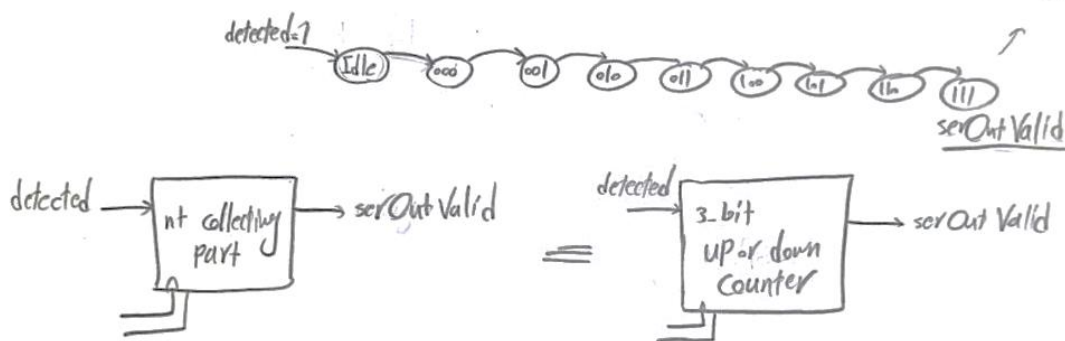


حالا بخش های آنرا هم تا حد کمی بیان می کنیم (بعدا تغییرات بسیاری اعمال میشود، این تصویری است که در اولین نگاه کشیدم)

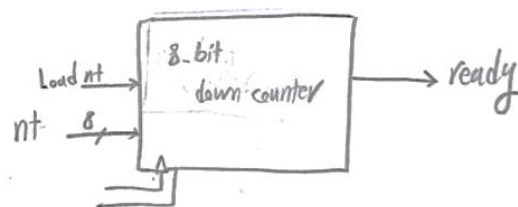


nt collecting part :

We could build it by a down counter/...

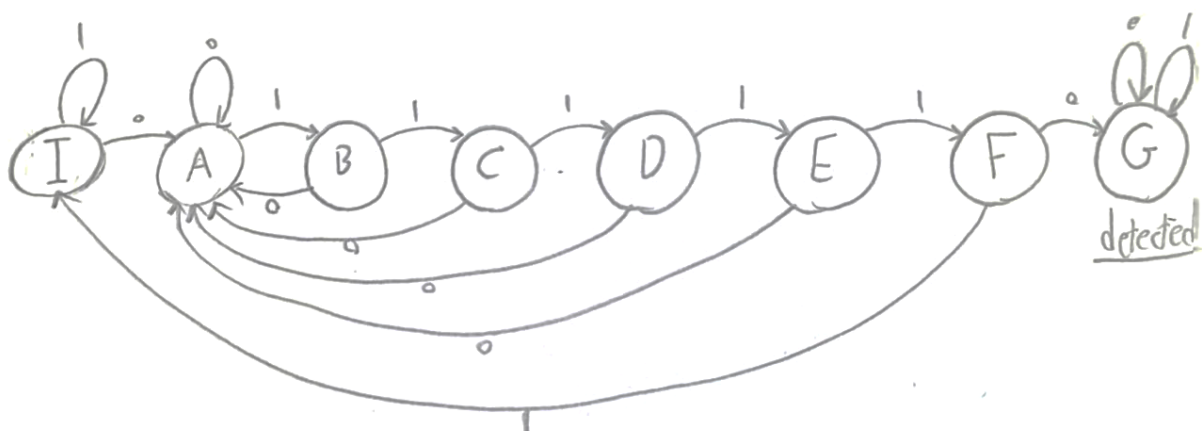


Transmission part :



Part a

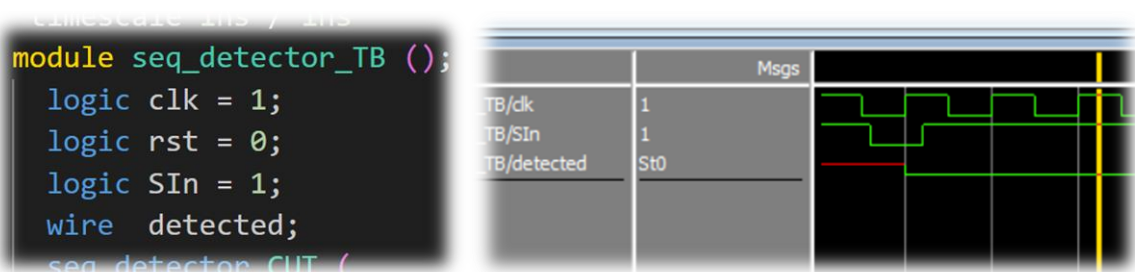
در این بخش در جهت اینکه بخش **sequence detector** را کاملاً جدا از سایر بخش‌ها طراحی کنیم، آنرا بصورت زیر تغییر می‌دهیم (در **state G** دیگر فرایند یافت توالی ادامه پیدا نکرده چون قرار است به سراغ بقیه بخش‌های مسئله برویم در آنجا، تا وقتی که دوباره در جایی دیگر **State** ا فراخوانی شود)



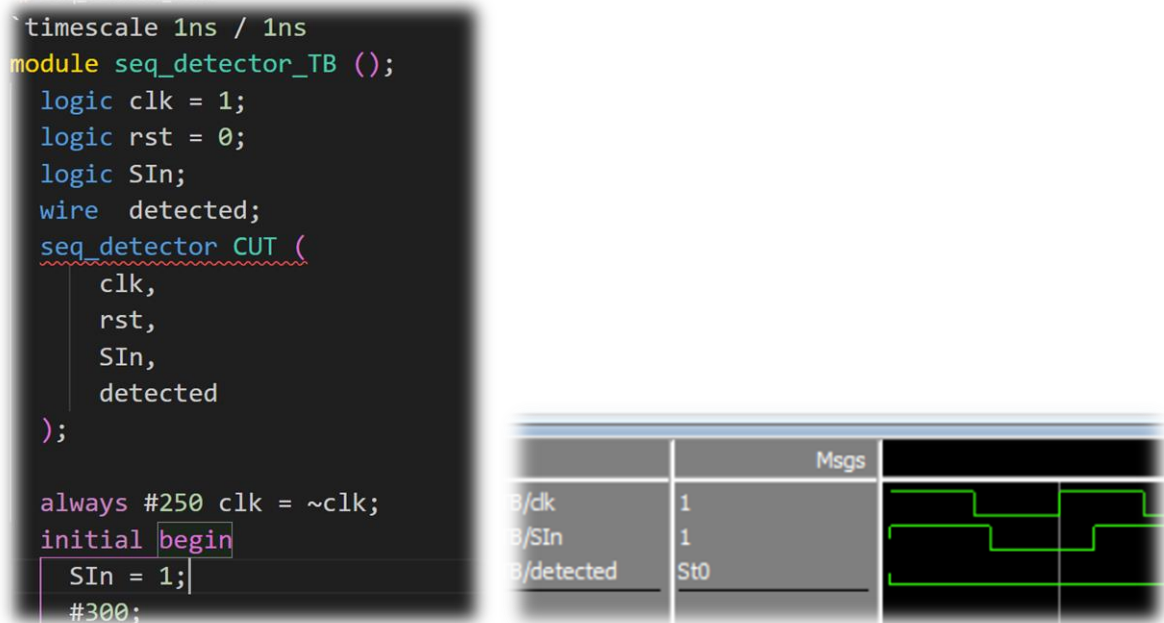
I = 000 , A = 001 , B = 010 , C = 011 , D = 100 , E = 101 , F = 110 , G = 111

i.

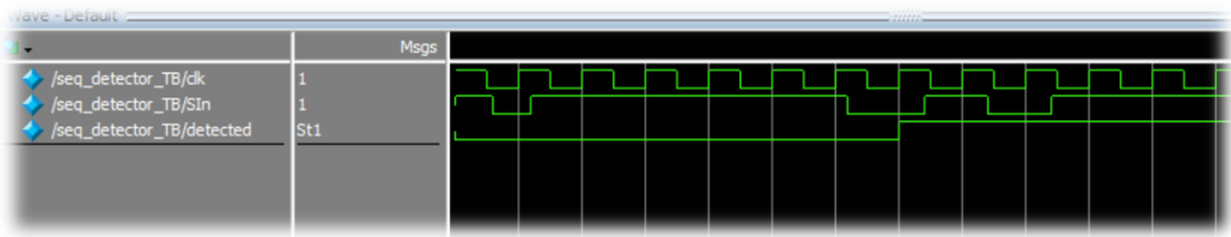
نکته ی جالب اینکه وقتی مقدار اولیه ی Sin را در حین تعریف کردن مایپول تعریف میکنم، گویا لحظه ی اول که $clk = 1$ است بعنوان اولین **posedge** ای که **clk** میزند دیده نمیشود و مقداری در خروجی نمیروود تا وقتی که اولین بار واقعا **posedge** بطور کامل دیده شود



اما در حالتی که در **initial begin** مقدار اولیه می دهیم، آن مقدار Z ابتدایی دیده نشده و یعنی همان **initial value** داده شده به **clk** که مساوی 1 است، بعنوان **posedge** تشخیص داده میشود:



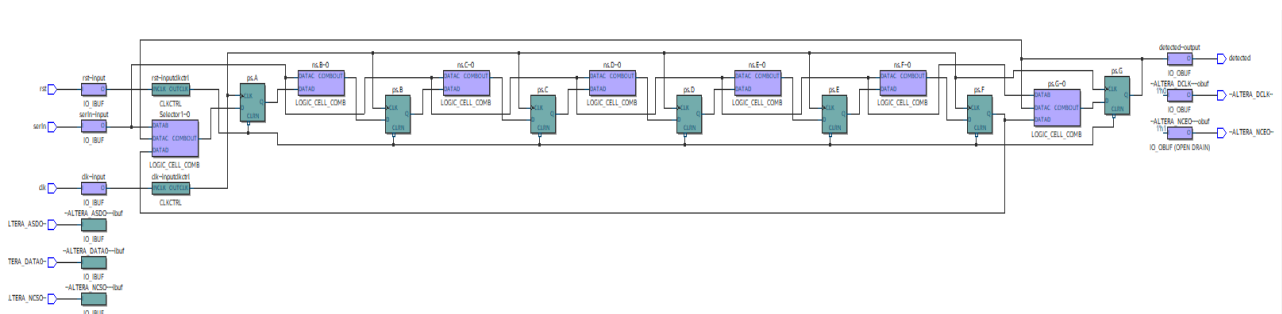
حالا در کل ما با روش دوم رفتیم و **Waveform** همانطور که انتظار داریم ابتدا یکبار **0111110** را تشخیص داده و سپس انقدر در همان **state** اخر می ماند تا وقتی در ادامه در ساخت این ماشین، این بخش را با بخش های دیگر ترکیب کرده و بخشی اضافه کنیم که بعد از **downcount** کردن بیاید و این بخش را به **state I** برگرداند:

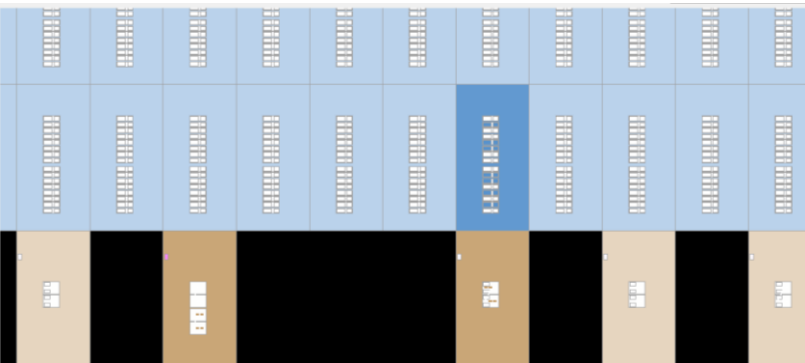
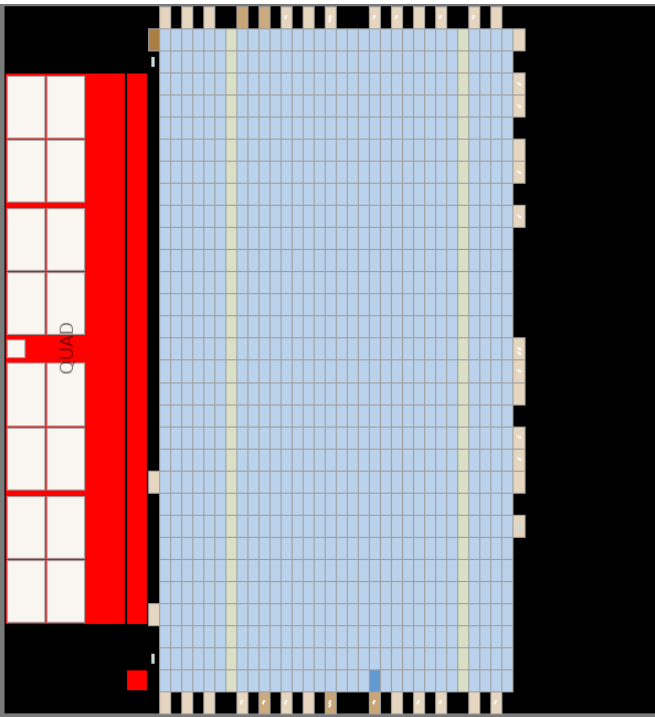


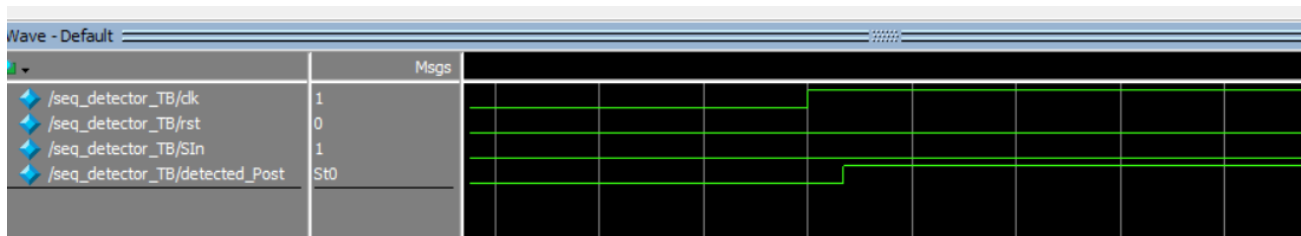
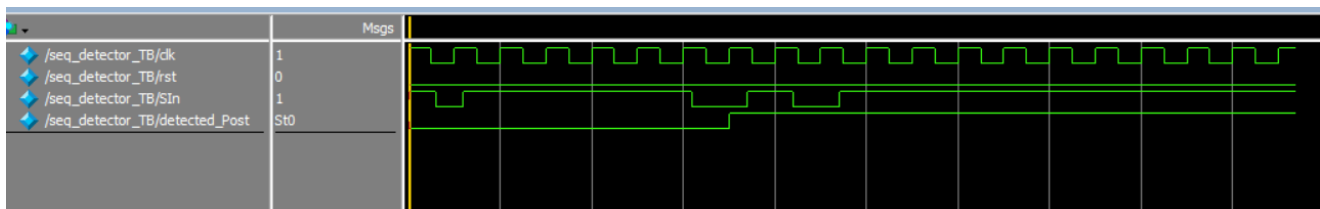
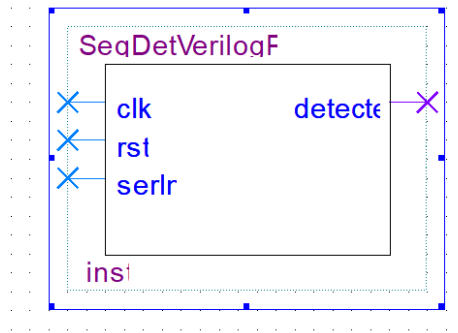
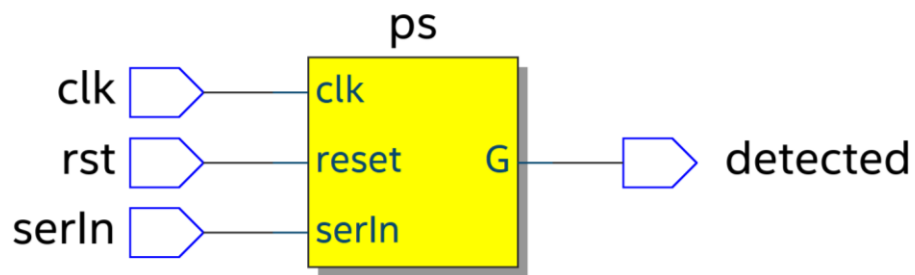
ii.

اطلاعات ساختاری مدار post_synthesis :

<<Filter>>	
Flow Status	Successful - Tue Jan 02 15:52:47 2024
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	SeqDetVerilogPP
Top-level Entity Name	SeqDetVerilogPP
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	7 / 14,400 (< 1 %)
Total registers	7
Total pins	4 / 81 (5 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)



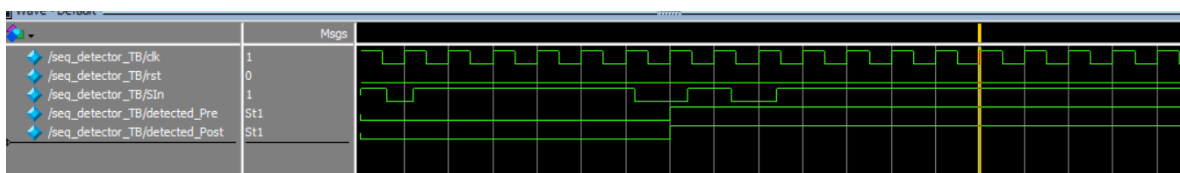




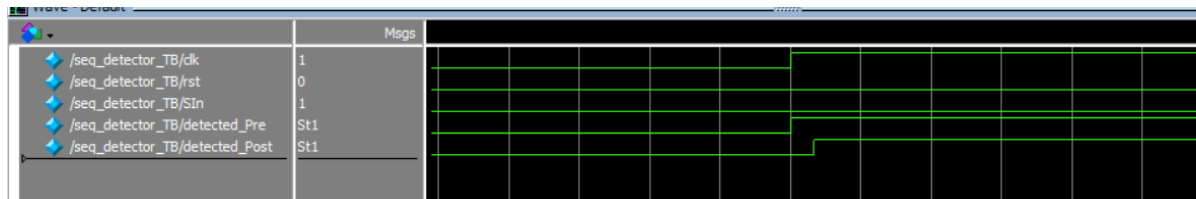
iii.

حالا waveform را برای pre و post مقایسه میکنیم:

دورنمای آنها یکسان است:



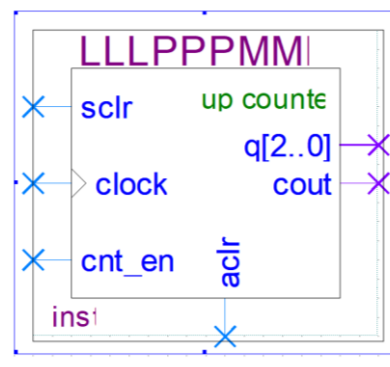
اما از نزدیک که می نگریم، بخاطر افزودن دیلی توسط quartus به حالت postsynthesis ، شکل موج ها اندکی دیلی دارند:



Part b

تا جاییکه متوجه شدم برای این بخش ترجیح بر این بود که بخش رجیستر را با وریلاگ نوشته و برای بخش counter مستقیما از سمبل های LPM استفاده کنیم. یعنی همچنین چیزی برای counter مدنظر بود:

ویژگی ها: count_enable, synch and asynch reset, clk, carry out, just up count



```

37 // synopsys translate_off
38 `timescale 1 ps / 1 ps
39 // synopsys translate_on
40 module LLLPPPM (
41     aclr,
42     clock,
43     cnt_en,
44     sclr,
45     cout,
46     q);
47
48     input  aclr;
49     input  clock;
50     input  cnt_en;
51     input  sclr;
52     output cout;
53     output [2:0] q;
54
55     wire sub_wire0;
56     wire [2:0] sub_wire1;
57     wire cout = sub_wire0;
58     wire [2:0] q = sub_wire1[2:0];
59
60     lpm_counter LPM_COUNTER_component (
61         .aclr (aclr),
62         .clock (clock),
63         .cnt_en (cnt_en),
64         .sclr (sclr),
65         .cout (sub_wire0),
66         .q (sub_wire1),
67         .aload (1'b0),
68         .aset (1'b0),
69         .cin (1'b1),
70         .clk_en (1'b1),
71         .data ({3{1'b0}}),
72         .eq (),
73         .sload (1'b0),
74         .sset (1'b0),
75         .updown (1'b1));
76
77     defparam
78         LPM_COUNTER_component.lpm_direction = "UP",
79         LPM_COUNTER_component.lpm_port_updown = "POF",
80         LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
81         LPM_COUNTER_component.lpm_width = 3;
82

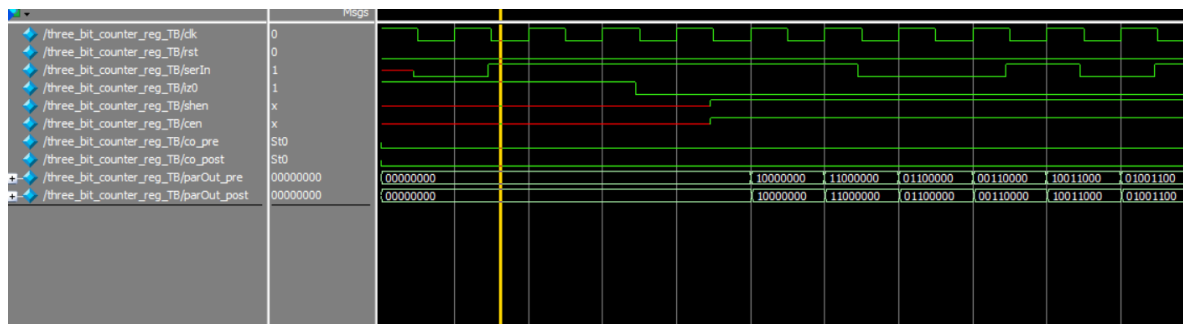
```

اما میتوان کد وریلاگ آنرا هم همینطوری دو بخشی نوشت و اینجوری بنظرم راحتتر بود پس از ابتدا همه را وریلاگ زدم.

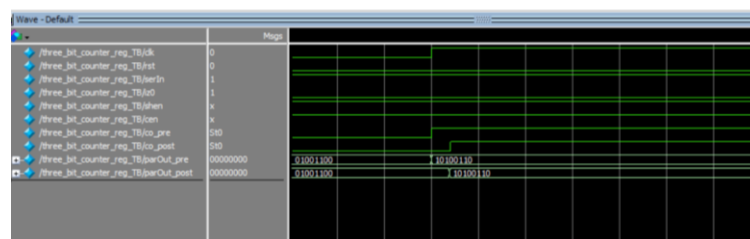
i.

ii.

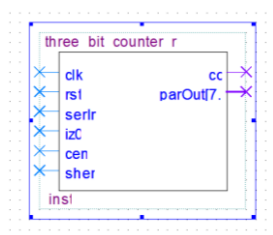
مقایسه ی pre و post در modelsim به صورت زیر است:



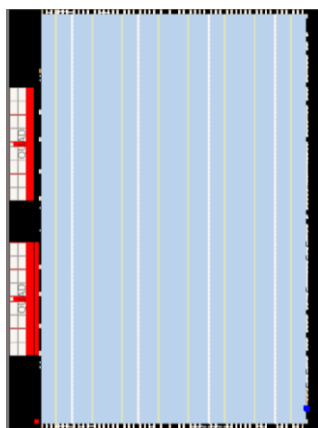
که دقیقاً مدار post با مقداری دیلی نسبت به pre کار خود را میکند:



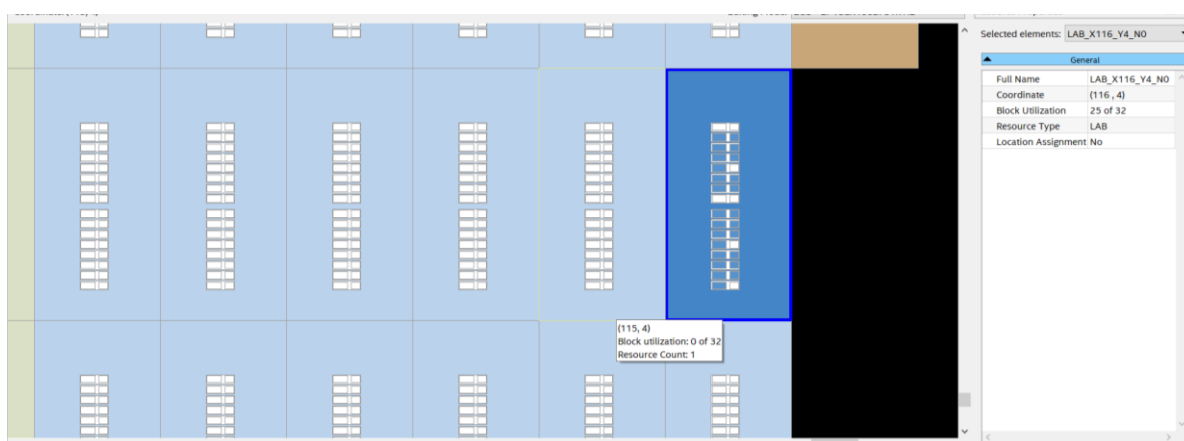
و سمبل آن:



و floor plan آن:



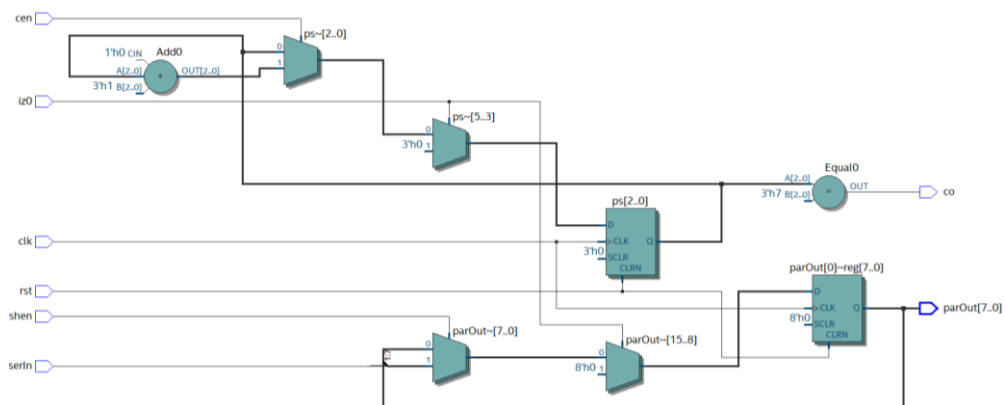
که در این FPGA بخشی که استفاده شده است این است:



و اطلاعات ساختاری آن:

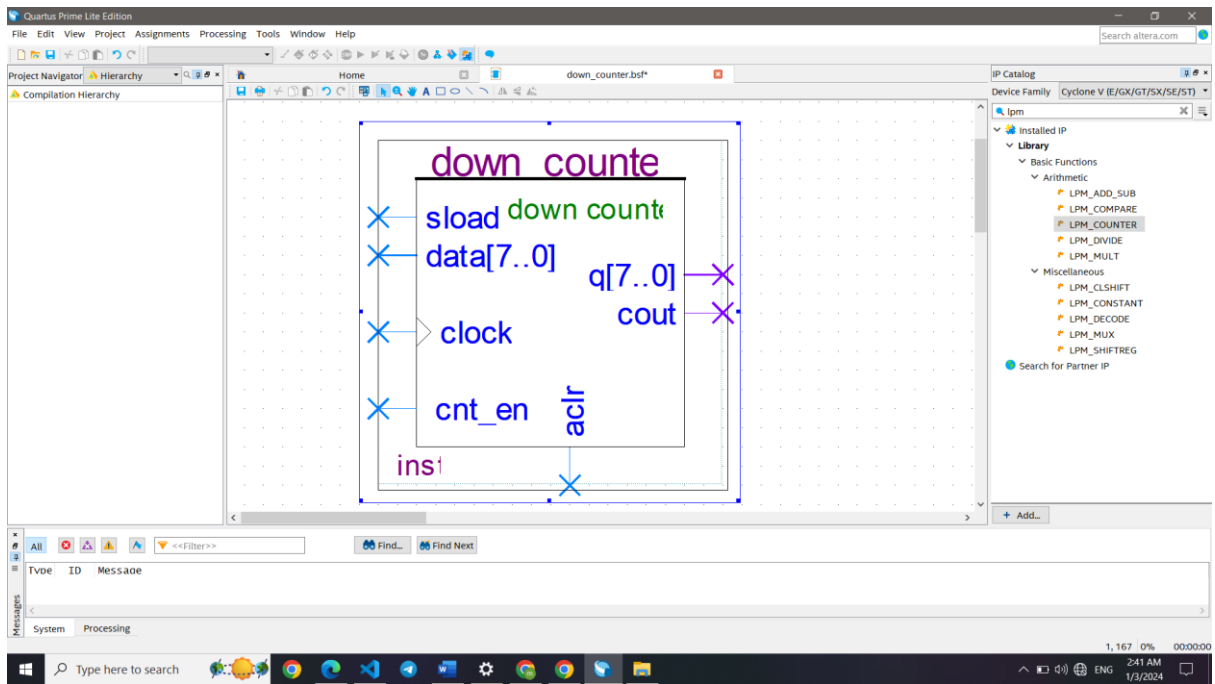
Flow Summary	
Flow Status	Successful - Wed Jan 03 00:29:18 2024
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	three_bit_counter_reg
Top-level Entity Name	three_bit_counter_reg
Family	Cyclone IV GX
Device	EP4CGX150DF317AD
Timing Models	Final
Total logic elements	14 / 149,760 (< 1 %)
Total registers	11
Total pins	15 / 508 (3 %)
Total virtual pins	0
Total memory bits	0 / 6,635,520 (0 %)
Embedded Multiplier 9-bit elements	0 / 720 (0 %)
Total GXB Receiver Channel PCS	0 / 8 (0 %)
Total GXB Receiver Channel PMA	0 / 8 (0 %)
Total GXB Transmitter Channel PCS	0 / 8 (0 %)
Total GXB Transmitter Channel PMA	0 / 8 (0 %)
Total PLLs	0 / 8 (0 %)

و : RTL view



Part c

اگر با lpm بنویسیم:

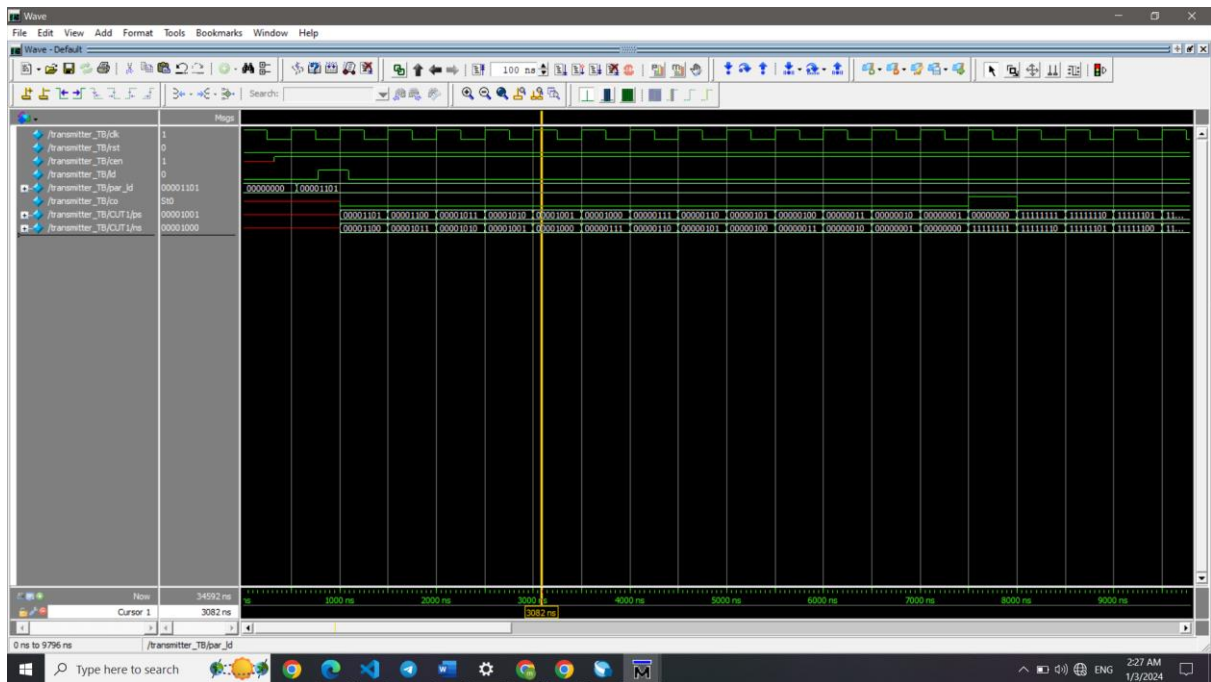


```

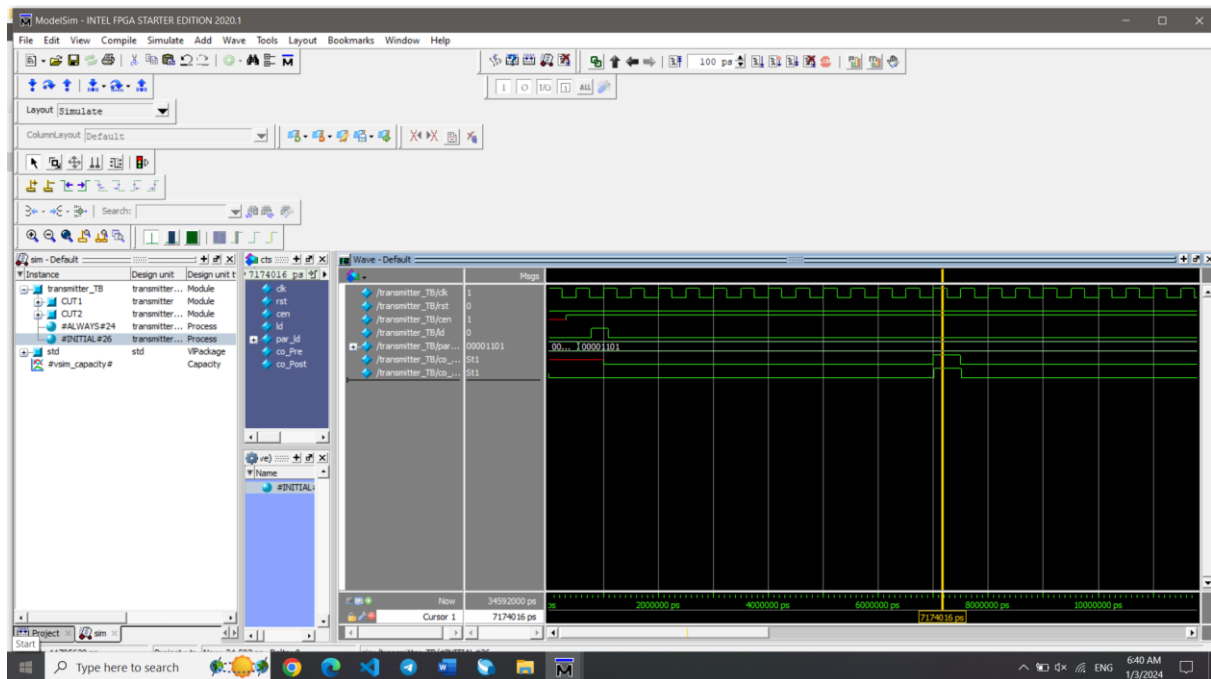
D: > uni > Semester 3 > Digital Systems > CAs > CA5 > transmitter > V down_counter.v
37 // synopsys translate_off
38 `timescale 1 ps / 1 ps
39 // synopsys translate_on
40 module down_counter (
41     aclr,
42     clock,
43     cnt_en,
44     data,
45     sload,
46     cout,
47     q);
48
49 input  aclr;
50 input  clock;
51 input  cnt_en;
52 input  [7:0] data;
53 input  sload;
54 output cout;
55 output [7:0] q;
56
57 wire sub_wire0;
58 wire [7:0] sub_wire1;
59 wire cout = sub_wire0;
60 wire [7:0] q = sub_wire1[7:0];
61
62 lpm_counter LPM_COUNTER_component (
63     .aclr (aclr),
64     .clock (clock),
65     .cnt_en (cnt_en),
66     .data (data),
67     .sload (sload),
68     .cout (sub_wire0),
69     .q (sub_wire1),
70     .aload (1'b0),
71     .aset (1'b0),
72     .cin (1'b1),
73     .clk_en (1'b1),
74     .eq (),
75     .sclr (1'b0),
76     .sset (1'b0),
77     .updown (1'b1));
78
79 defparam
80     LPM_COUNTER_component.lpm_direction = "DOWN",
81     LPM_COUNTER_component.lpm_port_updown = "POF",
82     LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
83     LPM_COUNTER_component.lpm_width = 8;
84
85 endmodule

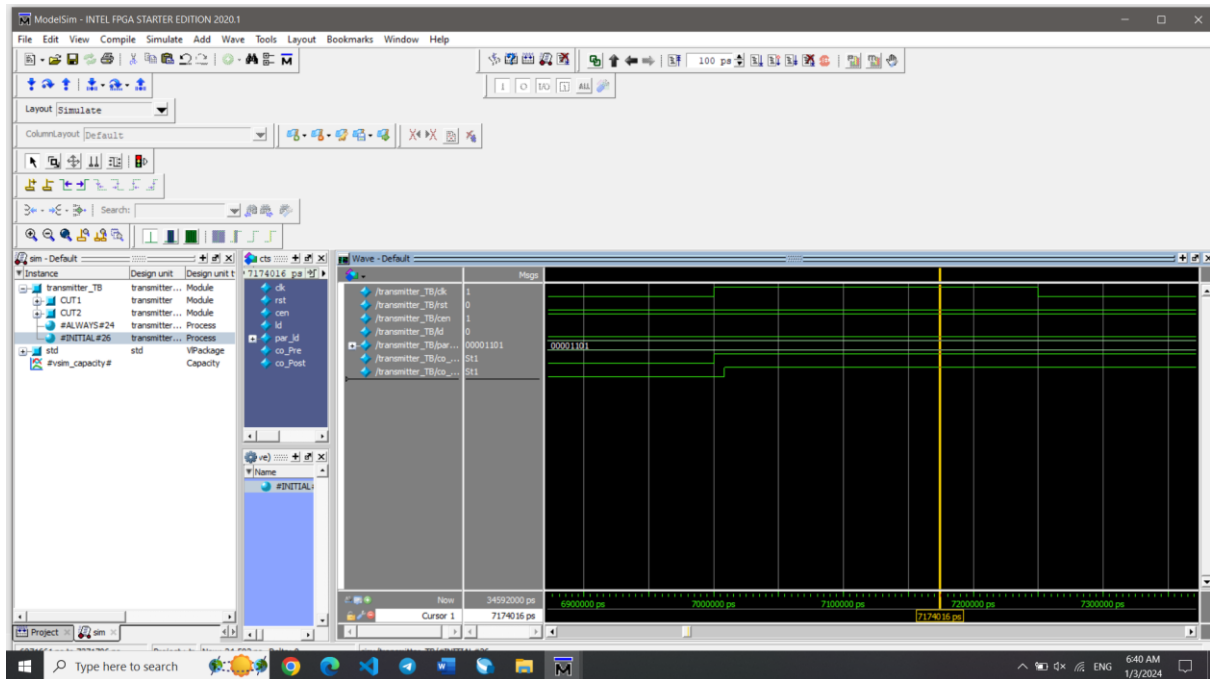
```

سیمولیت کردن Pre

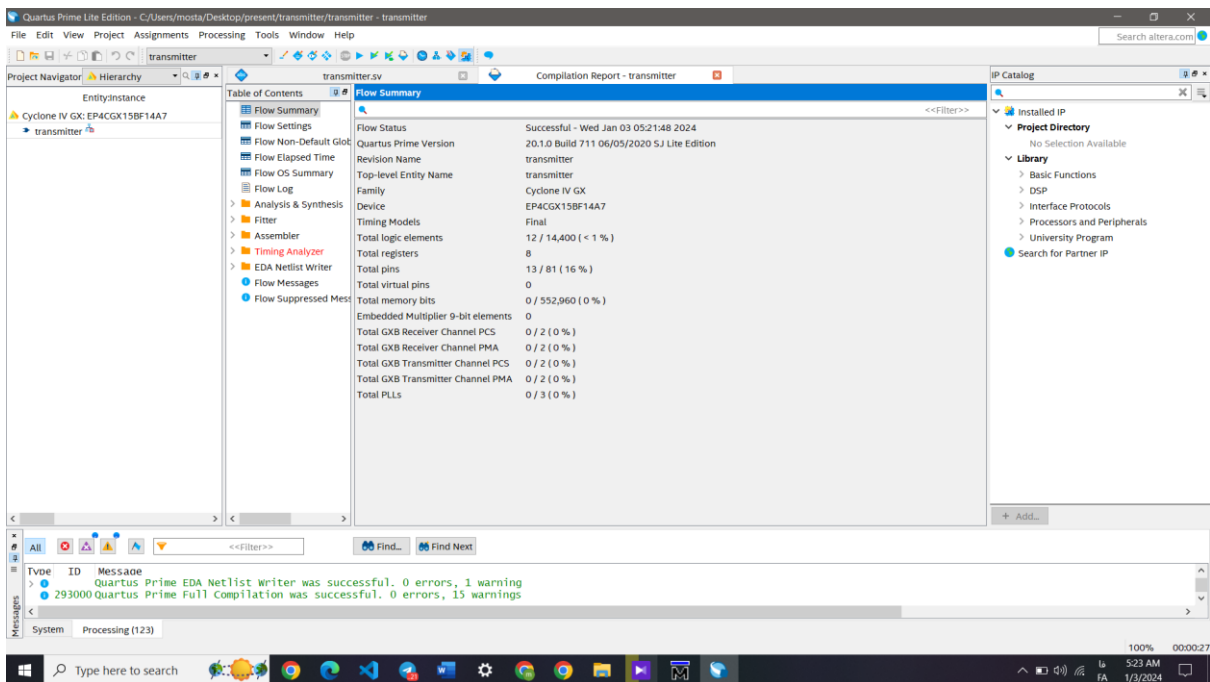


مقایسه ی pre و post:





حالا اگر با وریلاگ بنیم:



Quartus Prime Lite Edition - C:/Users/mosta/Desktop/present/transmitter/transmitter - transmitter

File Edit View Project Assignments Processing Tools Window Help

transmitter

transmitter.v

Compilation Report - transmitter

IP Catalog

Entity/Instance

Cyclone IV GX: EP4CGX15BF14A7

transmitter

Table of Contents

- Assembler
- Timing Analyzer
 - Summary
 - Parallel Compilation
 - Clocks
 - Slow 1200mV 125C Model
 - Slow 1200mV -40C Model
 - Fmax Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary
 - Removal Summary
 - Minimum Pulse Width Summary
 - Worst-Case Timing Paths
 - Metastability Summary
 - Fast 1200mV -40C Model
 - Multicorner Timing Analysis Summary
 - Advanced I/O Timing
 - Clock Transfers
 - Report TCSS
 - Report RSKM
 - Unconstrained Paths
 - Messages
 - EDA Netlist Writer
 - Flow Messages
 - Flow Suppressed Messages

Multicorner Timing Analysis Summary

	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	Worst-case Slack	-1.028	0.254	N/A	N/A	-3.000
1	clk	-1.028	0.254	N/A	N/A	-3.000
2	Design-wide TNS	-6.39	0.0	0.0	0.0	-17.856
1	clk	-6.390	0.000	N/A	N/A	-17.856

IP Catalog

Installed IP

No Selection Available

Project Directory

- Basic Functions
- DSP
- Interface Protocols
- Processors and Peripherals
- University Program

Search for Partner IP

Messages

Type ID Message

- Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 15 warnings

System Processing (123)

100% 00:00:27

Type here to search

Quartus Prime Lite Edition - C:/Users/mosta/Desktop/present/transmitter/transmitter - transmitter

File Edit View Project Assignments Processing Tools Window Help

transmitter

transmitter.v

Compilation Report - transmitter

transmitter.bsf

IP Catalog

Entity/Instance

Cyclone IV GX: EP4CGX15BF14A7

transmitter

transmitter

clk

rst

ld

cen

par ld7.0

ins!

IP Catalog

Installed IP

No Selection Available

Project Directory

- Basic Functions
- DSP
- Interface Protocols
- Processors and Peripherals
- University Program

Search for Partner IP

Messages

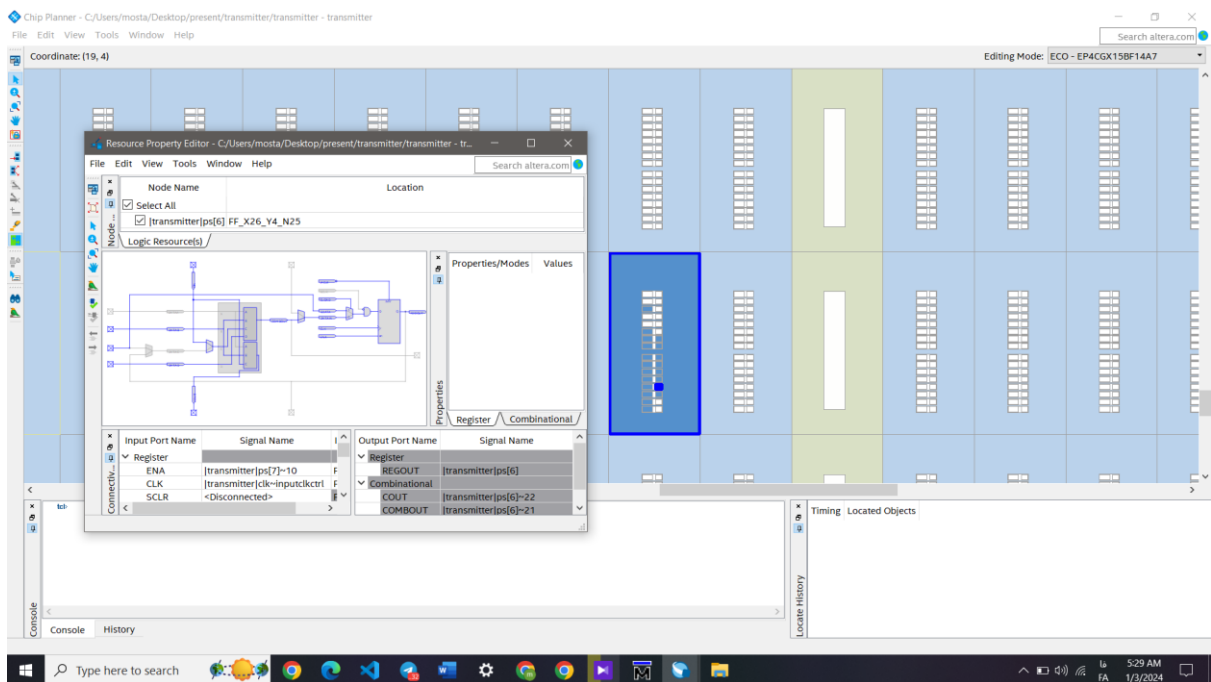
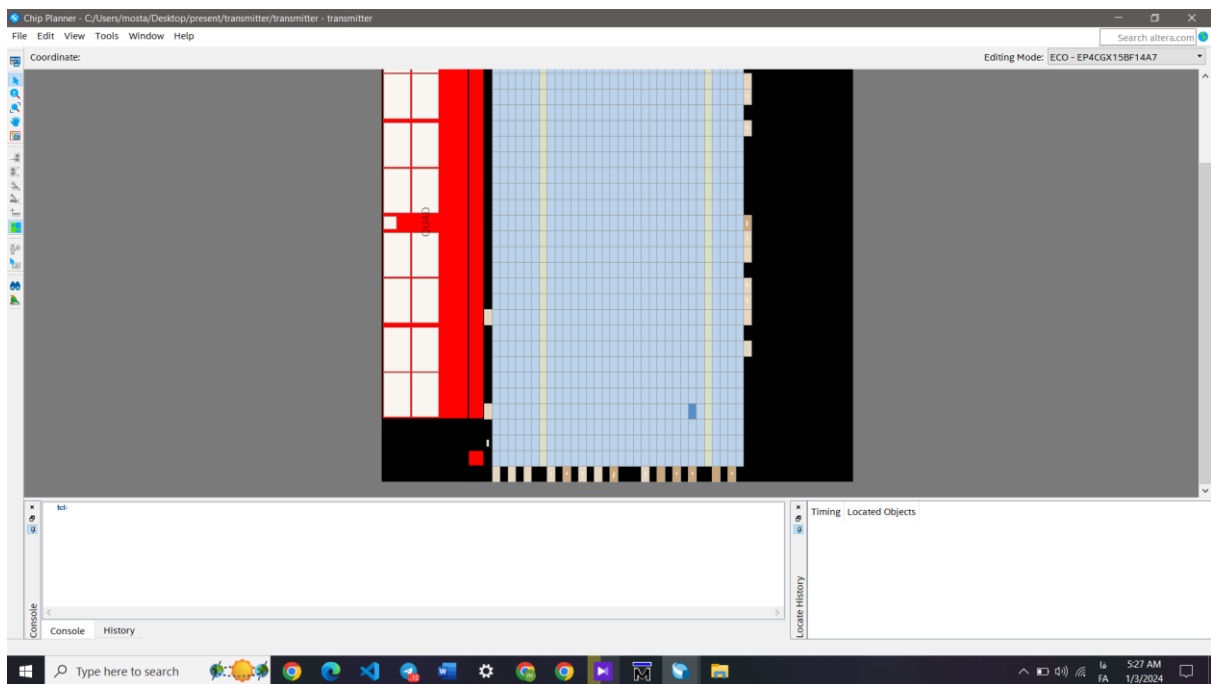
Type ID Message

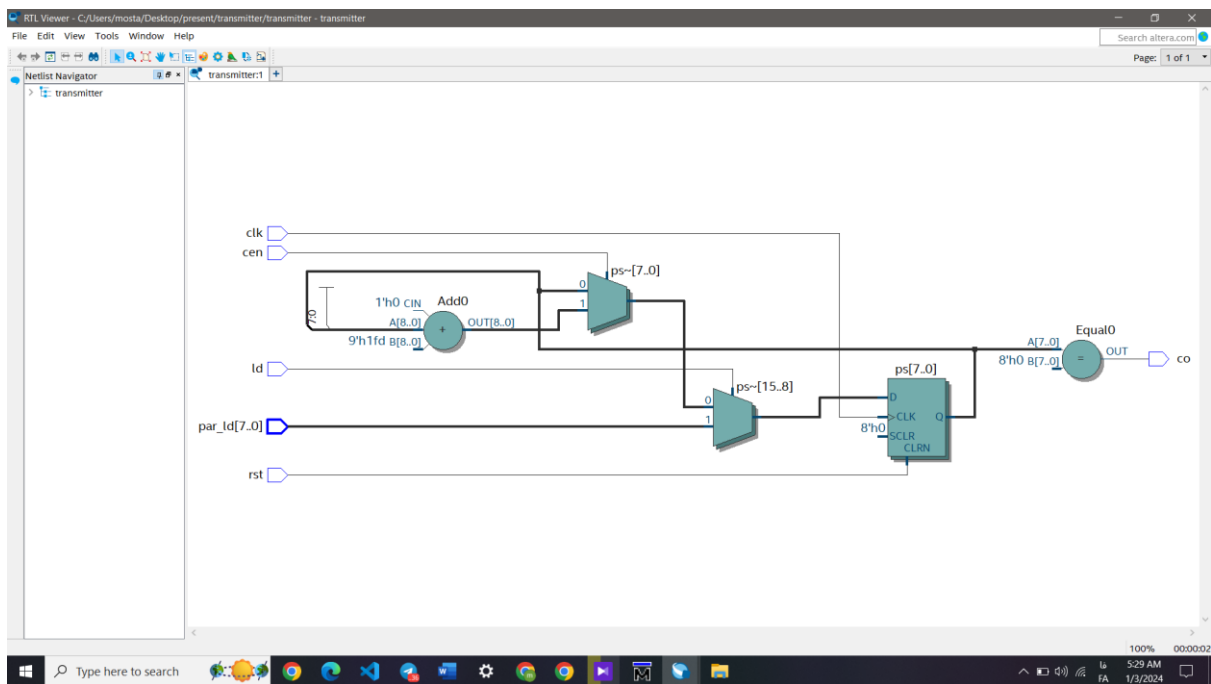
- Command: quartus_map --read_settings_files=on --write_settings_files=off transmitter -c transmitter --generate_symbol=C:/Users/mosta/Desktop/present/transmitter/transmitter.v
- Quartus Prime Create Symbol File was successful. 0 errors, 0 warnings

System Processing (4)

386,417 100% 00:00:01

Type here to search





Part d

برای این بخش ابتدا بخش seq detector را با دادن تغییرات بصورت کنترلر در آوردیم. شکل نهایی کار اینگونه است:

State Machine Viewer - C:/Users/mosta/Desktop/present/seqDet_cont/seq_cont - seq_cont

File Edit View Tools Window Help

State Machine: seq_conttps

	Source State	Destination State	Condition
1	A	A	(tserin)
2	A	B	(serin)
3	B	C	(serin)
4	B	A	(tserin)
5	C	D	(serin)
6	C	A	(tserin)
7	D	E	(serin)
8	D	A	(tserin)
9	E	A	(tserin)
10	E	F	(serin)
11	F	I	(serin)
12	F	G	(tserin)
13	G	Id_state	(co1)
14	G	G	(tco1)
15	I	I	(serin)
16	I	A	(tserin)
17	Id_state	tr_state	
18	tr_state	I	(co2)
19	tr_state	tr_state	(tco2)

State Table

Transitions / Encoding /

Start

100% 00:00:01

Type here to search

State Machine Viewer - C:/Users/mosta/Desktop/present/seqDet_cont/seq_cont - seq_cont

File Edit View Tools Window Help

State Machine: seq_conttps

```
graph LR; I((I)) -- "(tserin)" --> A((A)); A -- "(serin)" --> B((B)); B -- "(serin)" --> C((C)); C -- "(serin)" --> D((D)); D -- "(serin)" --> E((E)); E -- "(serin)" --> F((F)); F -- "(serin)" --> I((I)); F -- "(tserin)" --> G((G)); G -- "(tco1)" --> Id_state((Id_state)); G -- "(tco1)" --> G((G)); Id_state -- "" --> tr_state((tr_state)); tr_state -- "(co2)" --> I((I)); tr_state -- "(tco2)" --> tr_state((tr_state)); A -- "(tserin)" --> A((A)); B -- "(tserin)" --> A((A)); C -- "(tserin)" --> A((A)); D -- "(tserin)" --> A((A)); E -- "(tserin)" --> A((A));
```

	Source State	Destination State	Condition
1	A	A	(tserin)
2	A	B	(serin)

State Table

Transitions / Encoding /

Start

100% 00:00:01

Type here to search

