Cover Page

CSCI 2272 - Computer Organization and Lab

Project Name: The Best Vending Machine Ever

Student 1 Name and BC Email: Kieran Roth rothki@bc.edu

Student 2 Name and BC Email: Lindsay Penkrat

Name of the TA: Brian Wang

Timings of the Lab slot: Wednesday 4-6PM

Final Project

The Best Vending Machine Ever!

Project Description:

Our Vending Machine is built through strictly Verilog Code. It utilizes the Seven Segment Display that has been modified to become a Two Digit Seven Segment. This machine works by having three different products; Gum costs 5 cents, Chips cost 25 cents, and Soda costs 30 cents. The user will flip the switch assigned to the specific product they desire. Then they will flip the assigned switch of the coin they wish to enter (Nickel, Dime, Quarter.) The Board displays "G" when gum is selected, "S" when Soda is selected, and "C" when Chips are selected. The machine also has a reset button which turns all LEDS and resets the display back to 0. The user must manually flip the item switch down though.

For Soda, as seen in the video there are multiple combinations of coins which allow the product to be dispensed. For all items, there is an "Error" light which displays when the user inputs too little money, which only appears for Soda and Chips (not gum because it's the lowest value). The "More Money" light illuminates when you put in too much money to purchase the good. This acts as a signifier for change. Soda is an odd amount, so it requires change to be displayed. Finally, if the cost has been reached, the green Dispense light turns on.

Relevance to Course:

Our Vending Machine uses the Seven Segment code we did much earlier in lab. But, we edited this file to become a two-digit Seven Segment display. We also utilized the methods of assigning LEDs presented in the Traffic Light Lab.

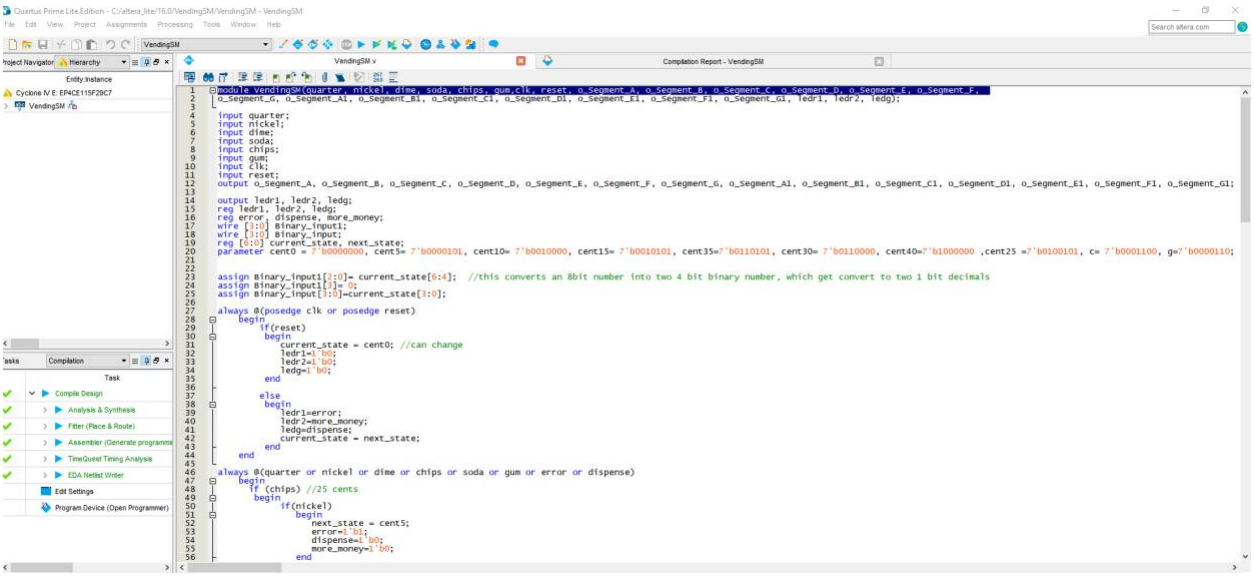
At first, we thought we would need online help for the 2 digit seven segment, but our questions were fully answered in Project Review Sessions by Professor Biswas.

Project Roles:

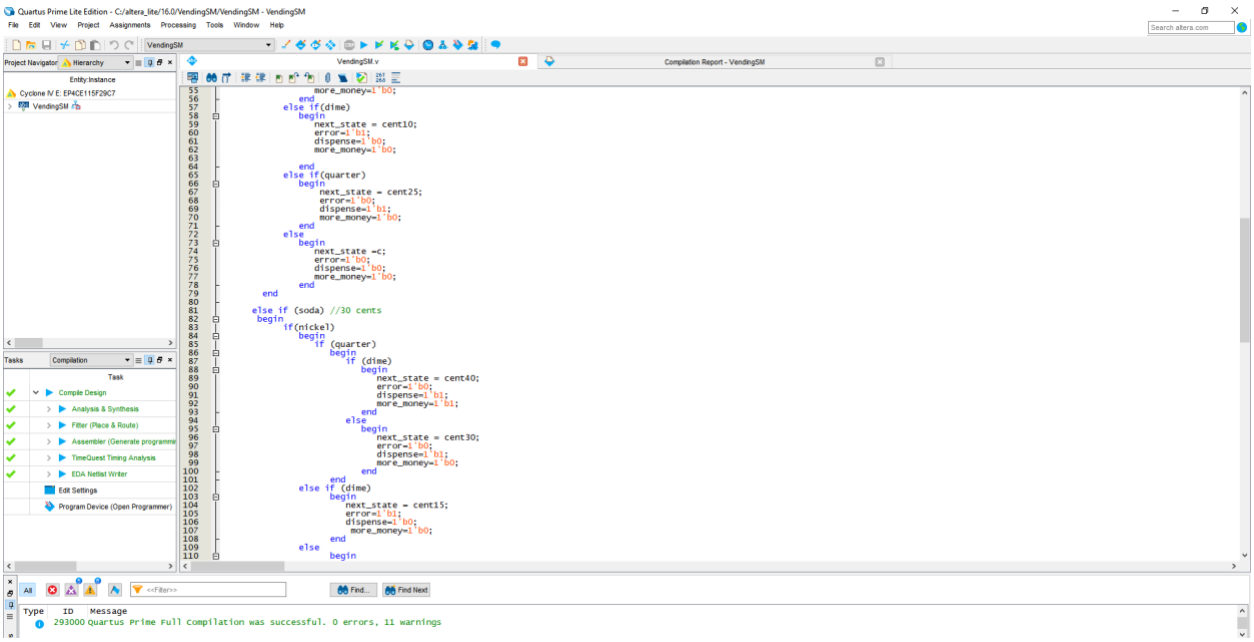
We both contributed completely equally to this project. We both took turns coding specifically in Verilog. At first, we tried the schematic design but we both did not understand that well enough.

Reflection:

We learned so much about the structure of Verilog and just how these boards operate. We tried for so long to utilize the buttons, but that was not something we were able to accomplish. But we were able to move over to the switches, which was much easier. We wanted to use the buttons to have an increased count of the coin you input, but we would have had to include a "debouncer" which was far too complicated for this course. We thought this would be super fun project, so once we started and ran into issues we did not want to give up.

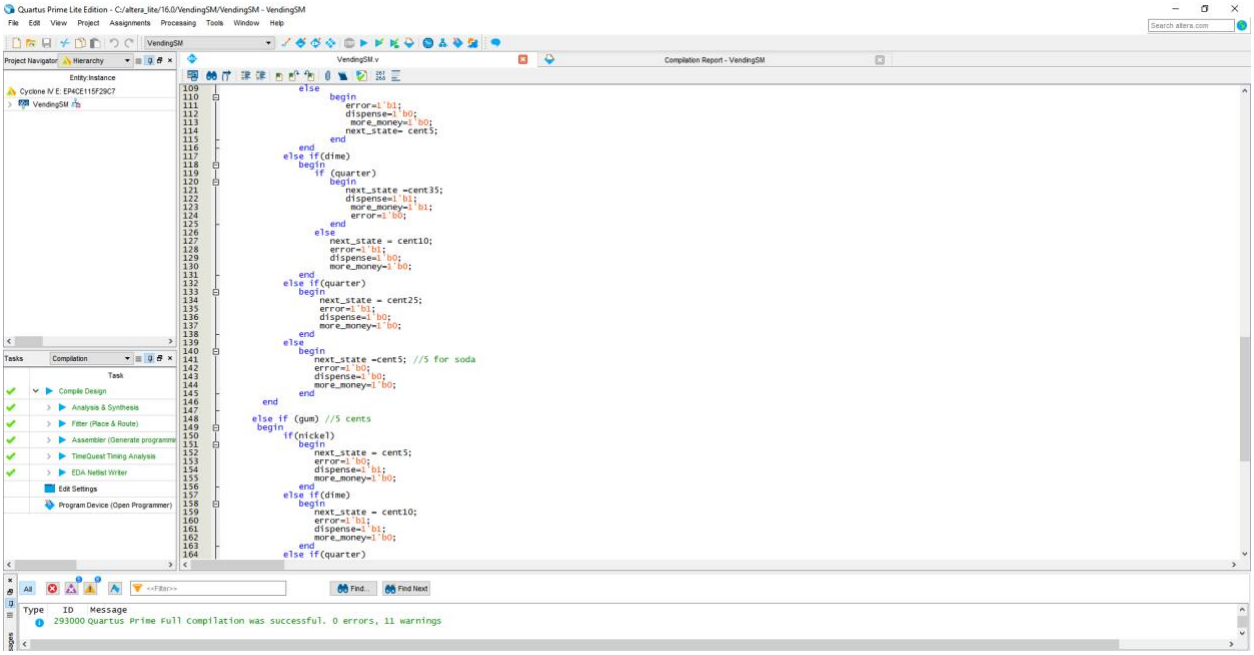


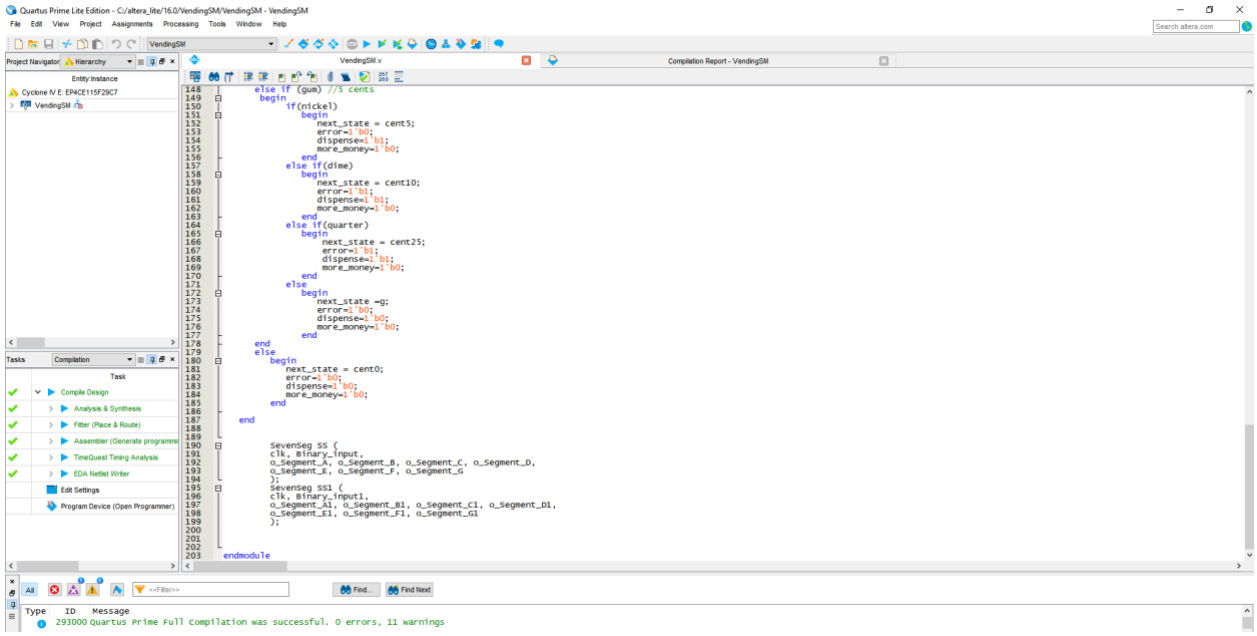
```
module VendingSM(quarter, nickel, dime, soda, chips, gum,clk, reset, o_Segment_A, o_Segment_B, o_Segment_C, o_Segment_D, o_Segment_E, o_Segment_F,
o_Segment_G, o_Segment_A1, o_Segment_B1, o_Segment_C1, o_Segment_D1, o_Segment_E1, o_Segment_F1, o_Segment_G1, ledr1, ledr2, ledg);

input quarter;
input nickel;
input dime;
input soda;
input chips;
input gum;
input clk;
input reset;
output o_Segment_A, o_Segment_B, o_Segment_C, o_Segment_D, o_Segment_E, o_Segment_F, o_Segment_G, o_Segment_A1, o_Segment_B1, o_Segment_C1, o_Segment_D1, o_Segment_E1, o_Segment_F1, o_Segment_G1;

output ledr1, ledr2, ledg;
reg ledr1, ledr2, ledg;
reg error, dispense, more_money;
wire [3:0] Binary_input1;
wire [3:0] Binary_input;
reg [6:0] current_state, next_state;
parameter cent0 = 7'b0000000, cent5= 7'b0000101, cent10= 7'b0010000, cent15= 7'b0010101, cent35=7'b0110101, cent30= 7'b0110000, cent40=7'b1000000 ,cent25 =7'b0100101, c= 7'b0001100, g=7'b0000110;


assign Binary_input1[2:0]= current_state[6:4];  //this converts an 8bit number into two 4 bit binary number, which get convert to two 1 bit decimals
assign Binary_input1[3]= 0;
assign Binary_input[3:0]=current_state[3:0];

always @(posedge clk or posedge reset)
    begin
        if(reset)
        begin
            current_state = cent0; //can change
            ledr1=1'b0;
            ledr2=1'b0;
            ledg=1'b0;
        end

        else
        begin
            ledr1=error;
            ledr2=more_money;
            ledg=dispense;
            current_state = next_state;
        end
    end

always @(quarter or nickel or dime or chips or soda or gum or error or dispense)
    begin
      if (chips) //25 cents
       begin
          if(nickel)
             begin
                next_state = cent5;
                error=1'b1;
                dispense=1'b0;
                more_money=1'b0;
             end
```

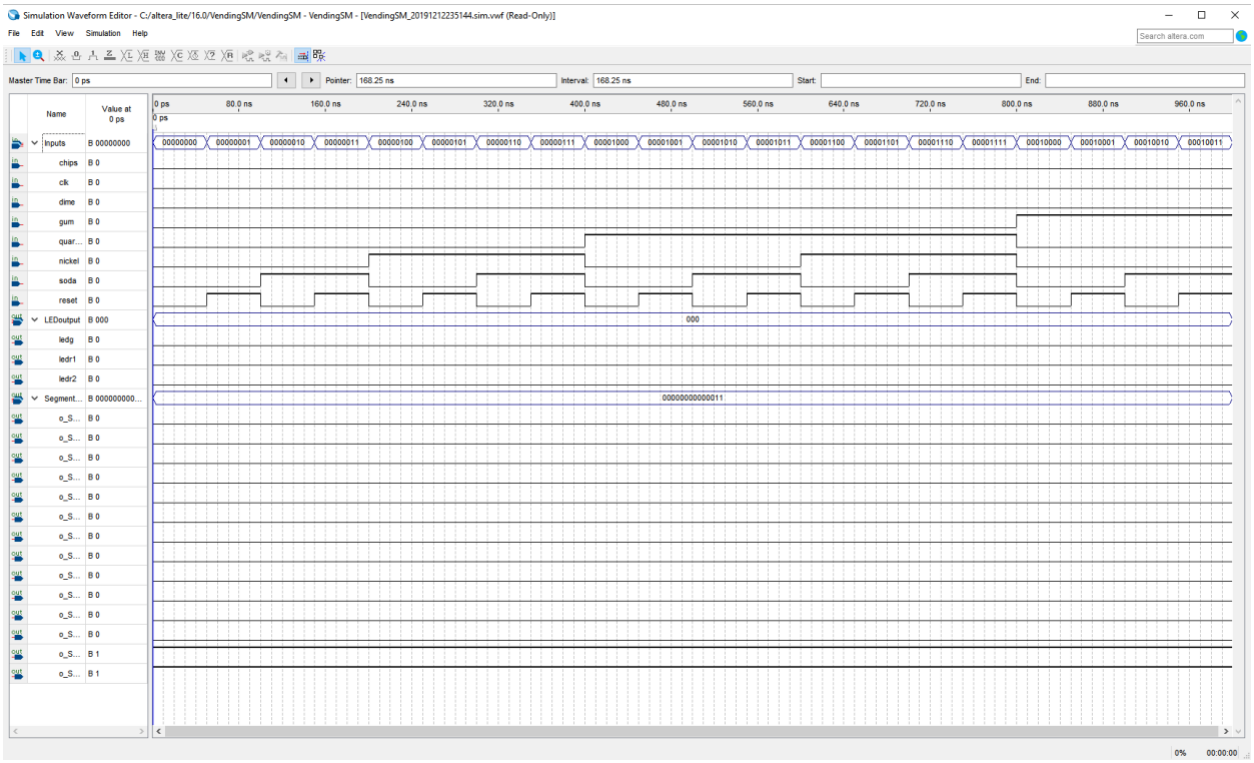


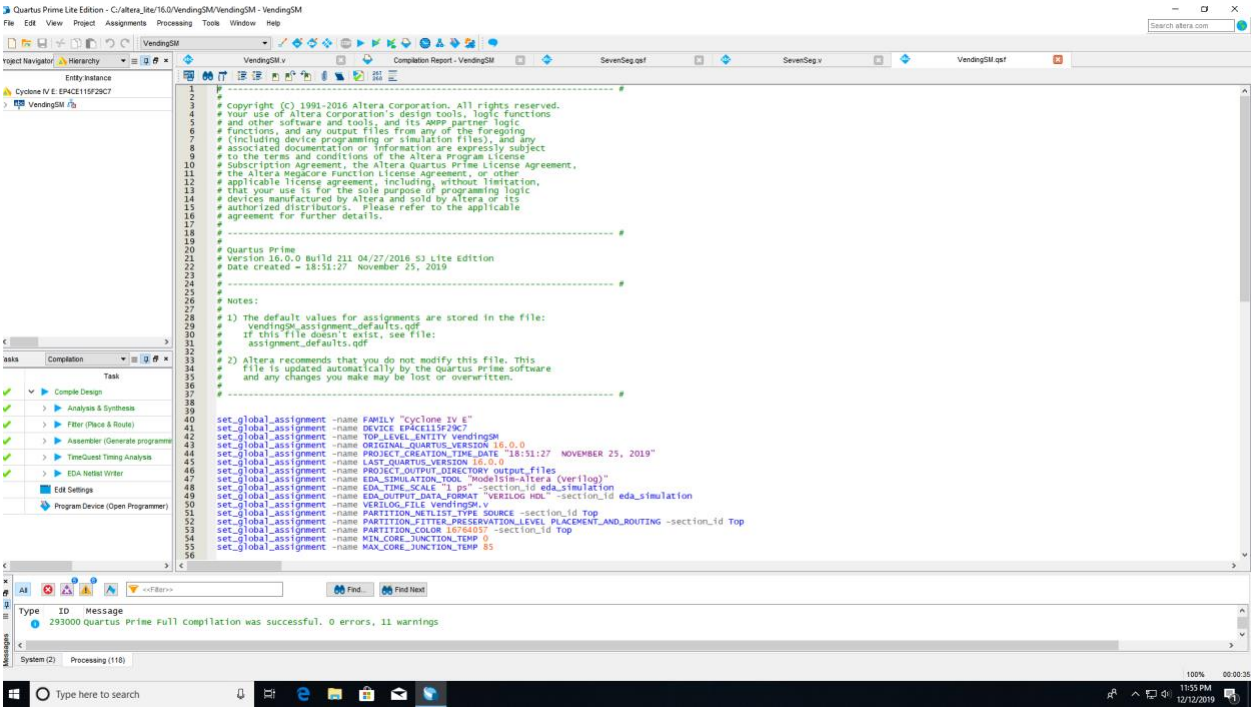
```
                more_money=1'b0;
             end
          else if(dime)
             begin
                next_state = cent10;
                error=1'b1;
                dispense=1'b0;
                more_money=1'b0;

             end
          else if(quarter)
             begin
                next_state = cent25;
                 error=1'b0;
                dispense=1'b1;
                more_money=1'b0;
             end
          else
             begin
                next_state =c;
                error=1'b0;
                dispense=1'b0;
                more_money=1'b0;
             end
       end

    else if (soda) //30 cents
     begin
          if(nickel)
             begin
                if (quarter)
                   begin
                      if (dime)
                      begin
                         next_state = cent40;
                         error=1'b0;
                         dispense=1'b1;
                         more_money=1'b1;
                      end
                   else
                      begin
                         next_state = cent30;
                         error=1'b0;
                         dispense=1'b1;
                         more_money=1'b0;
                      end
                end
             else if (dime)
                begin
                   next_state = cent15;
                   error=1'b1;
                   dispense=1'b0;
                    more_money=1'b0;
                end
             else
                begin
```

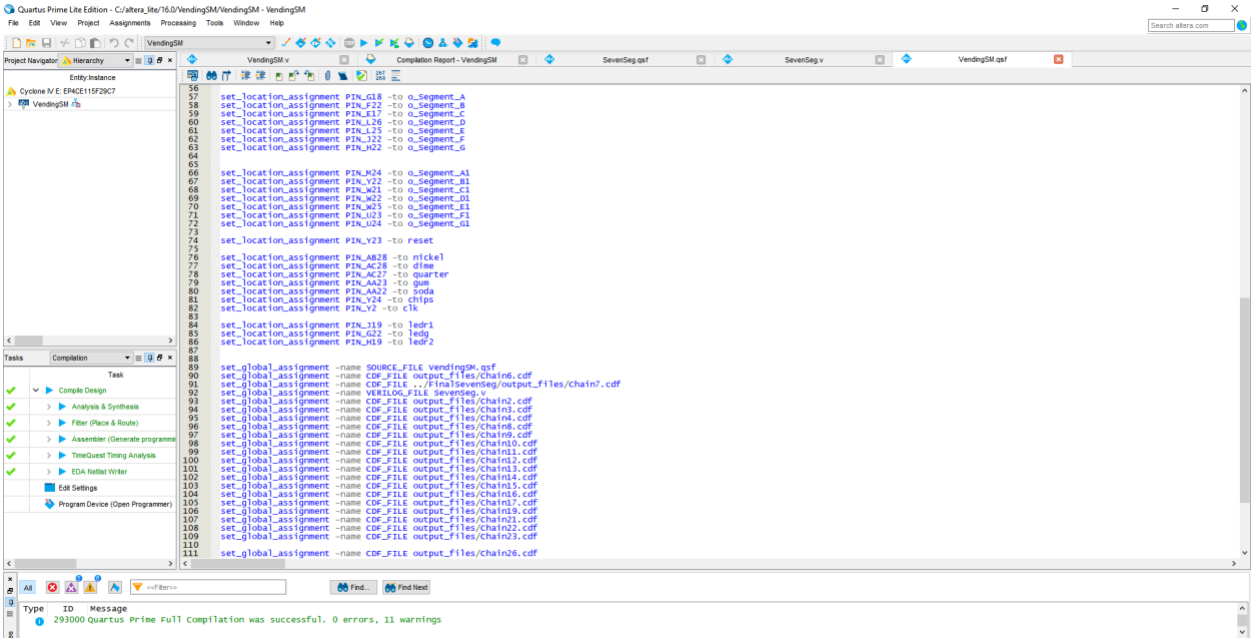
```
Type    ID    Message
  293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings
```

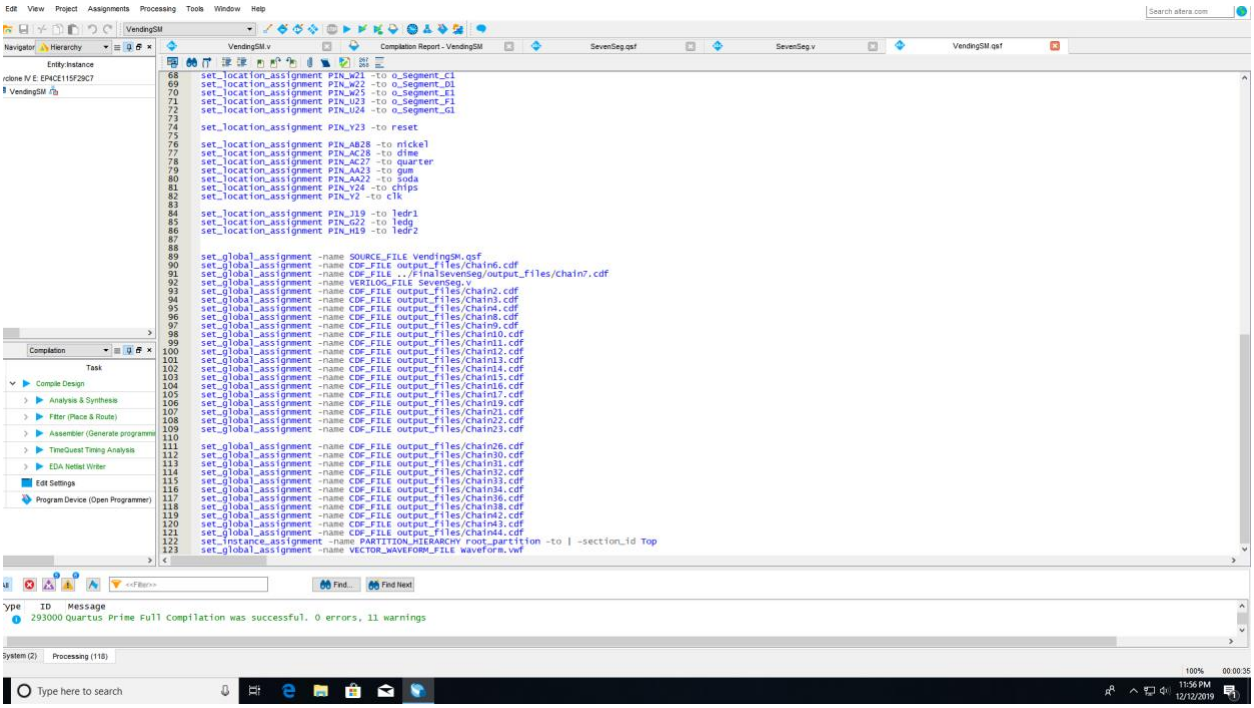

Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
VendingSM
Project Navigator Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
VendingSM.v
Compilation Report - VendingSM
109 else
110 begin
111 error=1'b1;
112 dispense=1'b0;
113 more_money=1'b0;
114 next_state= cent5;
115 end
116 end
117 else if(dime)
118 begin
119 if (quarter)
120 begin
121 next_state =cent35;
122 dispense=1'b1;
123 more_money=1'b1;
124 error=1'b0;
125 end
126 else
127 next_state = cent10;
128 error=1'b1;
129 dispense=1'b0;
130 more_money=1'b0;
131 end
132 else if(quarter)
133 begin
134 next_state = cent25;
135 error=1'b1;
136 dispense=1'b0;
137 more_money=1'b0;
138 end
139 else
140 begin
141 next_state =cent5; //5 for soda
142 error=1'b0;
143 dispense=1'b0;
144 more_money=1'b0;
145 end
146 end
147
148 else if (gum) //5 cents
149 begin
150 if(nickel)
151 begin
152 next_state = cent5;
153 error=1'b0;
154 dispense=1'b1;
155 more_money=1'b0;
156 end
157 else if(dime)
158 begin
159 next_state = cent10;
160 error=1'b1;
161 dispense=1'b1;
162 more_money=1'b0;
163 end
164 else if(quarter)
All Find... Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings

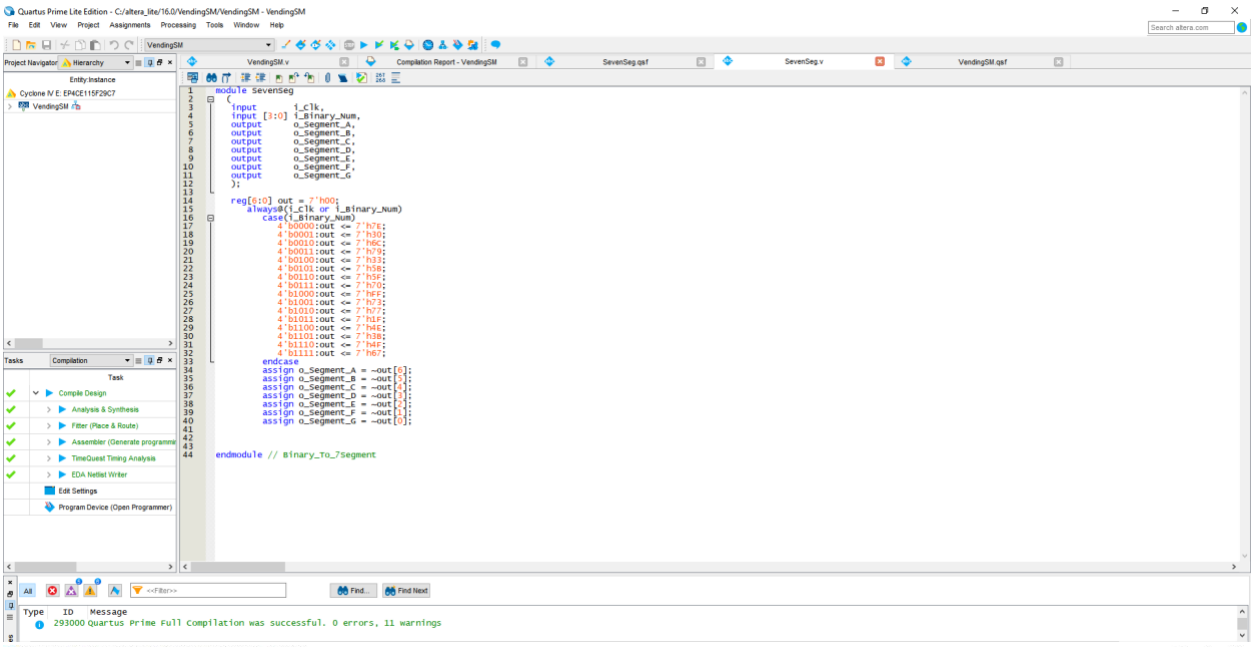

Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
VendingSM
Project Navigator Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
VendingSM.v
Compilation Report - VendingSM
148 else if (gum) //5 cents
149 begin
150 if(nickel)
151 begin
152 next_state = cent5;
153 error=1'b0;
154 dispense=1'b1;
155 more_money=1'b0;
156 end
157 else if(dime)
158 begin
159 next_state = cent10;
160 error=1'b1;
161 dispense=1'b1;
162 more_money=1'b0;
163 end
164 else if(quarter)
165 begin
166 next_state = cent25;
167 error=1'b1;
168 dispense=1'b1;
169 more_money=1'b0;
170 end
171 else
172 begin
173 next_state =0;
174 error=1'b0;
175 dispense=1'b0;
176 more_money=1'b0;
177 end
178 end
179 else
180 begin
181 next_state = cent0;
182 error=1'b0;
183 dispense=1'b0;
184 more_money=1'b0;
185 end
186
187 end
188
189
190 SevenSeg S5 (
191 clk, Binary_input,
192 o_Segment_A, o_Segment_B, o_Segment_C, o_Segment_D,
193 o_Segment_E, o_Segment_F, o_Segment_G
194);
195 SevenSeg S51 (
196 clk, Binary_input1,
197 o_Segment_A1, o_Segment_B1, o_Segment_C1, o_Segment_D1,
198 o_Segment_E1, o_Segment_F1, o_Segment_G1
199);
200
201
202
203 endmodule
All Find... Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings

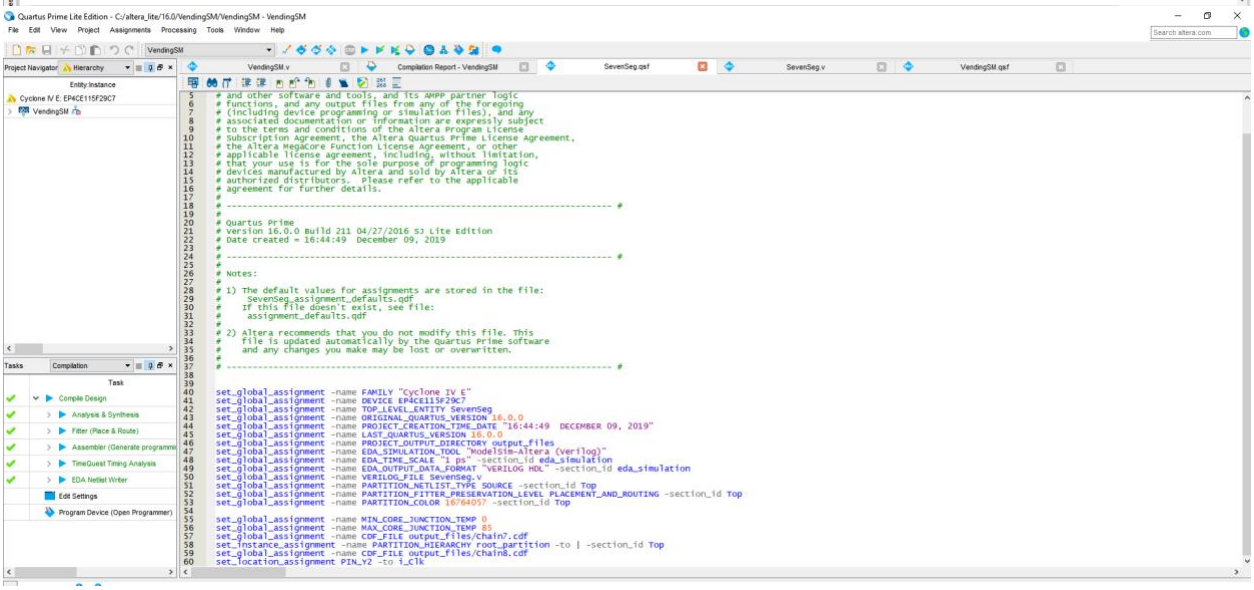

Simulation Waveform Editor - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM - [VendingSM_20191212235144.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 168.25 ns
Interval: 168.25 ns
Start:
End:
Name
Value at 0 ps
Inputs
chips
clk
dime
gum
quar...
nickel
soda
reset
LEDoutput
ledg
ledr1
ledr2
Segment...
o_S...
0 ps
80.0 ns
160.0 ns
240.0 ns
320.0 ns
400.0 ns
480.0 ns
560.0 ns
640.0 ns
720.0 ns
800.0 ns
880.0 ns
960.0 ns
000
0000000000011
0%
00:00:00


Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
VendingSM
Project Navigator
Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
Compilation Report - VendingSM
SevenSeg.qsf
SevenSeg.v
VendingSM.qsf
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Copyright (C) 1991-2016 Altera Corporation. All rights reserved.
Your use of Altera Corporation's design tools, logic functions
and other software and tools, and its AMPP partner logic
functions, and any output files from any of the foregoing
(including device programming or simulation files), and any
associated documentation or information are expressly subject
to the terms and conditions of the Altera Program License
Subscription Agreement, the Altera Quartus Prime License Agreement,
the Altera MegaCore Function License Agreement, or other
applicable license agreement, including, without limitation,
that your use is for the sole purpose of programming logic
devices manufactured by Altera and sold by Altera or its
authorized distributors. Please refer to the applicable
agreement for further details.
Quartus Prime
Version 16.0.0 Build 211 04/27/2016 SJ Lite Edition
Date created = 18:51:27 November 25, 2019
Notes:
1) The default values for assignments are stored in the file:
VendingSM_assignment_defaults.qdf
If this file doesn't exist, see file:
assignment_defaults.qdf
2) Altera recommends that you do not modify this file. This
file is updated automatically by the Quartus Prime software
and any changes you make may be lost or overwritten.
set_global_assignment -name FAMILY "Cyclone IV E"
set_global_assignment -name DEVICE EP4CE115F29C7
set_global_assignment -name TOP_LEVEL_ENTITY VendingSM
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 16.0.0
set_global_assignment -name PROJECT_CREATION_TIME_DATE "18:51:27 NOVEMBER 25, 2019"
set_global_assignment -name LAST_QUARTUS_VERSION 16.0.0
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim-Altera (Verilog)"
set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id eda_simulation
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL" -section_id eda_simulation
set_global_assignment -name VERILOG_FILE VendingSM.v
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -section_id Top
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
Find...
Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings
System (2)
Processing (118)
100%
00:00:35
Type here to search
11:55 PM
12/12/2019


Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
VendingSM.v
Compilation Report - VendingSM
SevenSeg.qsf
SevenSeg.v
VendingSM.qsf
set_location_assignment PIN_G18 -to o_Segment_A
set_location_assignment PIN_F22 -to o_Segment_B
set_location_assignment PIN_E17 -to o_Segment_C
set_location_assignment PIN_L26 -to o_Segment_D
set_location_assignment PIN_L25 -to o_Segment_E
set_location_assignment PIN_J22 -to o_Segment_F
set_location_assignment PIN_H22 -to o_Segment_G
set_location_assignment PIN_M24 -to o_Segment_A1
set_location_assignment PIN_Y22 -to o_Segment_B1
set_location_assignment PIN_W21 -to o_Segment_C1
set_location_assignment PIN_W22 -to o_Segment_D1
set_location_assignment PIN_W25 -to o_Segment_E1
set_location_assignment PIN_U23 -to o_Segment_F1
set_location_assignment PIN_U24 -to o_Segment_G1
set_location_assignment PIN_Y23 -to reset
set_location_assignment PIN_AB28 -to nickel
set_location_assignment PIN_AC28 -to dime
set_location_assignment PIN_AC27 -to quarter
set_location_assignment PIN_AA23 -to gum
set_location_assignment PIN_AA22 -to soda
set_location_assignment PIN_Y24 -to chips
set_location_assignment PIN_Y2 -to clk
set_location_assignment PIN_J19 -to ledr1
set_location_assignment PIN_G22 -to ledg
set_location_assignment PIN_H19 -to ledr2
set_global_assignment -name SOURCE_FILE VendingSM.qsf
set_global_assignment -name CDF_FILE output_files/Chain6.cdf
set_global_assignment -name CDF_FILE ../FinalSevenSeg/output_files/Chain7.cdf
set_global_assignment -name VERILOG_FILE SevenSeg.v
set_global_assignment -name CDF_FILE output_files/Chain2.cdf
set_global_assignment -name CDF_FILE output_files/Chain3.cdf
set_global_assignment -name CDF_FILE output_files/Chain4.cdf
set_global_assignment -name CDF_FILE output_files/Chain8.cdf
set_global_assignment -name CDF_FILE output_files/Chain9.cdf
set_global_assignment -name CDF_FILE output_files/Chain10.cdf
set_global_assignment -name CDF_FILE output_files/Chain11.cdf
set_global_assignment -name CDF_FILE output_files/Chain12.cdf
set_global_assignment -name CDF_FILE output_files/Chain13.cdf
set_global_assignment -name CDF_FILE output_files/Chain14.cdf
set_global_assignment -name CDF_FILE output_files/Chain15.cdf
set_global_assignment -name CDF_FILE output_files/Chain16.cdf
set_global_assignment -name CDF_FILE output_files/Chain17.cdf
set_global_assignment -name CDF_FILE output_files/Chain19.cdf
set_global_assignment -name CDF_FILE output_files/Chain21.cdf
set_global_assignment -name CDF_FILE output_files/Chain22.cdf
set_global_assignment -name CDF_FILE output_files/Chain23.cdf
set_global_assignment -name CDF_FILE output_files/Chain26.cdf
Project Navigator Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Find... Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings


Edit View Project Assignments Processing Tools Window Help
VendingSM.v
Compilation Report - VendingSM
SevenSeg.qsf
SevenSeg.v
VendingSM.qsf
set_location_assignment PIN_W21 -to o_Segment_C1
set_location_assignment PIN_W22 -to o_Segment_D1
set_location_assignment PIN_W25 -to o_Segment_E1
set_location_assignment PIN_U23 -to o_Segment_F1
set_location_assignment PIN_U24 -to o_Segment_G1
set_location_assignment PIN_Y23 -to reset
set_location_assignment PIN_AB28 -to nickel
set_location_assignment PIN_AC28 -to dime
set_location_assignment PIN_AC27 -to quarter
set_location_assignment PIN_AA23 -to gum
set_location_assignment PIN_AA22 -to soda
set_location_assignment PIN_Y24 -to chips
set_location_assignment PIN_Y2 -to clk
set_location_assignment PIN_J19 -to ledr1
set_location_assignment PIN_G22 -to ledg
set_location_assignment PIN_H19 -to ledr2
set_global_assignment -name SOURCE_FILE VendingSM.qsf
set_global_assignment -name CDF_FILE output_files/Chain6.cdf
set_global_assignment -name CDF_FILE ../FinalSevenSeg/output_files/Chain7.cdf
set_global_assignment -name VERILOG_FILE SevenSeg.v
set_global_assignment -name CDF_FILE output_files/Chain2.cdf
set_global_assignment -name CDF_FILE output_files/Chain3.cdf
set_global_assignment -name CDF_FILE output_files/Chain4.cdf
set_global_assignment -name CDF_FILE output_files/Chain8.cdf
set_global_assignment -name CDF_FILE output_files/Chain9.cdf
set_global_assignment -name CDF_FILE output_files/Chain10.cdf
set_global_assignment -name CDF_FILE output_files/Chain11.cdf
set_global_assignment -name CDF_FILE output_files/Chain12.cdf
set_global_assignment -name CDF_FILE output_files/Chain13.cdf
set_global_assignment -name CDF_FILE output_files/Chain14.cdf
set_global_assignment -name CDF_FILE output_files/Chain15.cdf
set_global_assignment -name CDF_FILE output_files/Chain16.cdf
set_global_assignment -name CDF_FILE output_files/Chain17.cdf
set_global_assignment -name CDF_FILE output_files/Chain19.cdf
set_global_assignment -name CDF_FILE output_files/Chain21.cdf
set_global_assignment -name CDF_FILE output_files/Chain22.cdf
set_global_assignment -name CDF_FILE output_files/Chain23.cdf
set_global_assignment -name CDF_FILE output_files/Chain26.cdf
set_global_assignment -name CDF_FILE output_files/Chain30.cdf
set_global_assignment -name CDF_FILE output_files/Chain31.cdf
set_global_assignment -name CDF_FILE output_files/Chain32.cdf
set_global_assignment -name CDF_FILE output_files/Chain33.cdf
set_global_assignment -name CDF_FILE output_files/Chain34.cdf
set_global_assignment -name CDF_FILE output_files/Chain36.cdf
set_global_assignment -name CDF_FILE output_files/Chain38.cdf
set_global_assignment -name CDF_FILE output_files/Chain42.cdf
set_global_assignment -name CDF_FILE output_files/Chain43.cdf
set_global_assignment -name CDF_FILE output_files/Chain44.cdf
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
set_global_assignment -name VECTOR_WAVEFORM_FILE Waveform.vwf
Find... Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings
Processing (118)
Type here to search
11:56 PM
12/12/2019


Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
VendingSM
Project Navigator Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
VendingSM.v
Compilation Report - VendingSM
SevenSeg.qsf
SevenSeg.v
VendingSM.qsf
module SevenSeg
(
input i_Clk,
input [3:0] i_Binary_Num,
output o_Segment_A,
output o_Segment_B,
output o_Segment_C,
output o_Segment_D,
output o_Segment_E,
output o_Segment_F,
output o_Segment_G
);
reg[6:0] out = 7'h00;
always@(i_Clk or i_Binary_Num)
case(i_Binary_Num)
4'b0000:out <= 7'h7E;
4'b0001:out <= 7'h30;
4'b0010:out <= 7'h6C;
4'b0011:out <= 7'h79;
4'b0100:out <= 7'h33;
4'b0101:out <= 7'h5B;
4'b0110:out <= 7'h5F;
4'b0111:out <= 7'h70;
4'b1000:out <= 7'h7F;
4'b1001:out <= 7'h73;
4'b1010:out <= 7'h77;
4'b1011:out <= 7'h1F;
4'b1100:out <= 7'h4E;
4'b1101:out <= 7'h3D;
4'b1110:out <= 7'h4F;
4'b1111:out <= 7'h67;
endcase
assign o_Segment_A = ~out[6];
assign o_Segment_B = ~out[5];
assign o_Segment_C = ~out[4];
assign o_Segment_D = ~out[3];
assign o_Segment_E = ~out[2];
assign o_Segment_F = ~out[1];
assign o_Segment_G = ~out[0];
endmodule // Binary_To_7Segment
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
Find...
Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 11 warnings


Quartus Prime Lite Edition - C:/altera_lite/16.0/VendingSM/VendingSM - VendingSM
File Edit View Project Assignments Processing Tools Window Help
VendingSM
Project Navigator Hierarchy
Entity:Instance
Cyclone IV E: EP4CE115F29C7
VendingSM
VendingSM.v
Compilation Report - VendingSM
SevenSeg.qsf
SevenSeg.v
VendingSM.qsf
and other software and tools, and its AMPP partner logic
functions, and any output files from any of the foregoing
(including device programming or simulation files), and any
associated documentation or information are expressly subject
to the terms and conditions of the Altera Program License
Subscription Agreement, the Altera Quartus Prime License Agreement,
the Altera MegaCore Function License Agreement, or other
applicable license agreement, including, without limitation,
that your use is for the sole purpose of programming logic
devices manufactured by Altera and sold by Altera or its
authorized distributors. Please refer to the applicable
agreement for further details.
Quartus Prime
Version 16.0.0 Build 211 04/27/2016 SJ Lite Edition
Date created = 16:44:49 December 09, 2019
Notes:
1) The default values for assignments are stored in the file:
SevenSeg_assignment_defaults.qdf
If this file doesn't exist, see file:
assignment_defaults.qdf
2) Altera recommends that you do not modify this file. This
file is updated automatically by the Quartus Prime software
and any changes you make may be lost or overwritten.
set_global_assignment -name FAMILY "Cyclone IV E"
set_global_assignment -name DEVICE EP4CE115F29C7
set_global_assignment -name TOP_LEVEL_ENTITY SevenSeg
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 16.0.0
set_global_assignment -name PROJECT_CREATION_TIME_DATE "16:44:49 DECEMBER 09, 2019"
set_global_assignment -name LAST_QUARTUS_VERSION 16.0.0
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim-Altera (Verilog)"
set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id eda_simulation
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL" -section_id eda_simulation
set_global_assignment -name VERILOG_FILE SevenSeg.v
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -section_id Top
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
set_global_assignment -name CDF_FILE output_files/chain7.cdf
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
set_global_assignment -name CDF_FILE output_files/chain8.cdf
set_location_assignment PIN_Y2 -to i_Clk
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
TimeQuest Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)

Project Files:

- Zip the entire Quartus Project folder and upload them separately.

In summary, you have to submit three things:

- This Report
- The Video (or link to the video in the report itself)
- Entire Project folder

Sample Project Topics

Following are some potential topics for your project, and you are encouraged to either pick one or come up with another topic.

The project must be implemented on the board.

1. Design and Implement 8 Bit Multiplier or 8 Bit Divider using Schematic Design entry in Quartus.

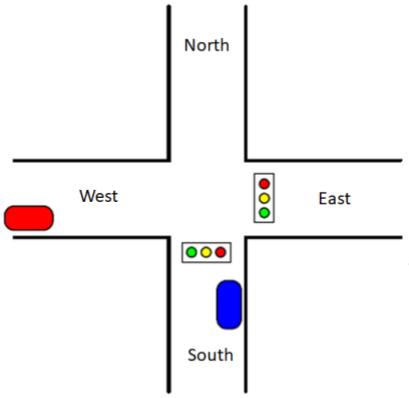
2. Design and Implement a counter (8 Bit) which can have a different initial value than its default value and can have a step size which is determined by the user. It should have atleast two of the following functionalities:
    i. Reset
    ii. Count down
    iii. Stopwatch
    iv. Alarm Clock (the LEDs can present the alarm)

3. Design and Implement a vending machine controller, which can have multiple commodities to dispense and should return the balance money if the user inputs excess.
    a. It should also have a reset button. What if the user wants to cancel or he made a wrong choice

4. Extend the ALU connections that you designed in Lab 7 and now add instruction memory and Program Counter to the design. You can embed the instructions within the instruction memory.

5. Implement a tic-tac-toe game using board switches. Design the game in verilog or schematic and use the switches of the board as inputs from two players.

6. Design and implement a Traffic Light Controller
   Here is a layout for the intersection: There are two lights (one to control the traffic between the north and south, one to control the traffic between west and east). Each light has yellow, red and green colors. We also need to integrate Pedestrian crossing signals in the traffic light controller.



7. Design and implement an elevator controller.