Lab 7: Design Vision

Connor Murphy

TA Sina

ECEN 454-504

4/1/2022

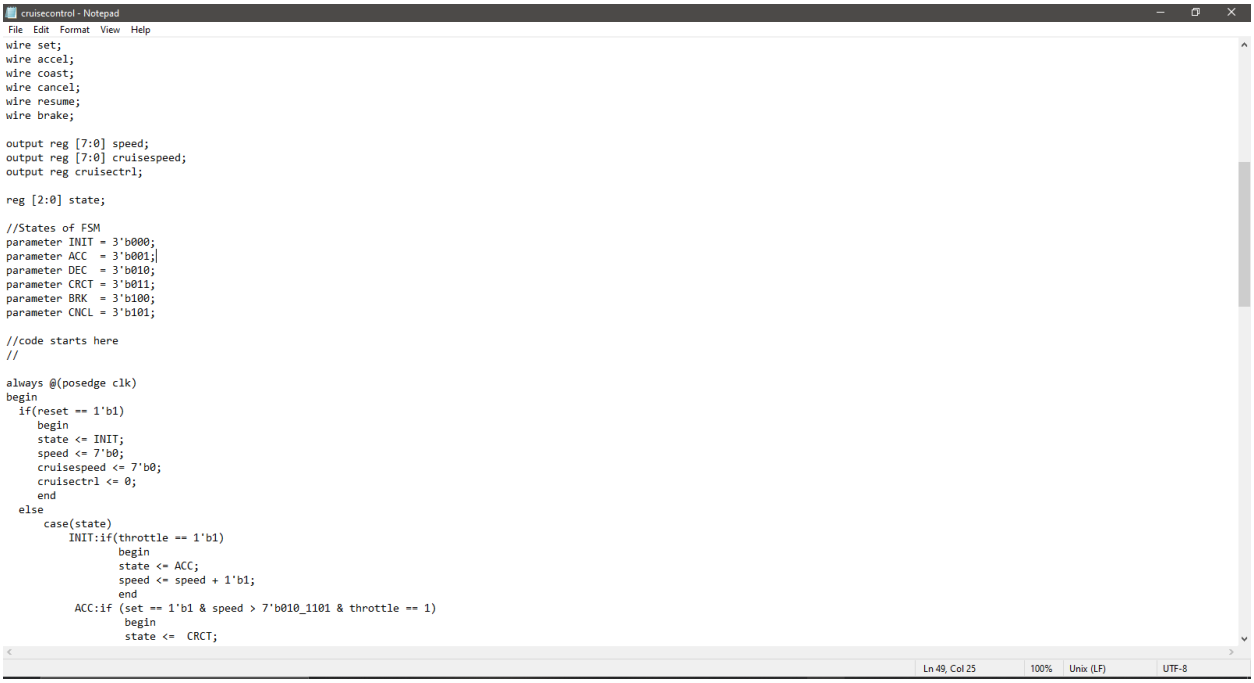
Introduction:

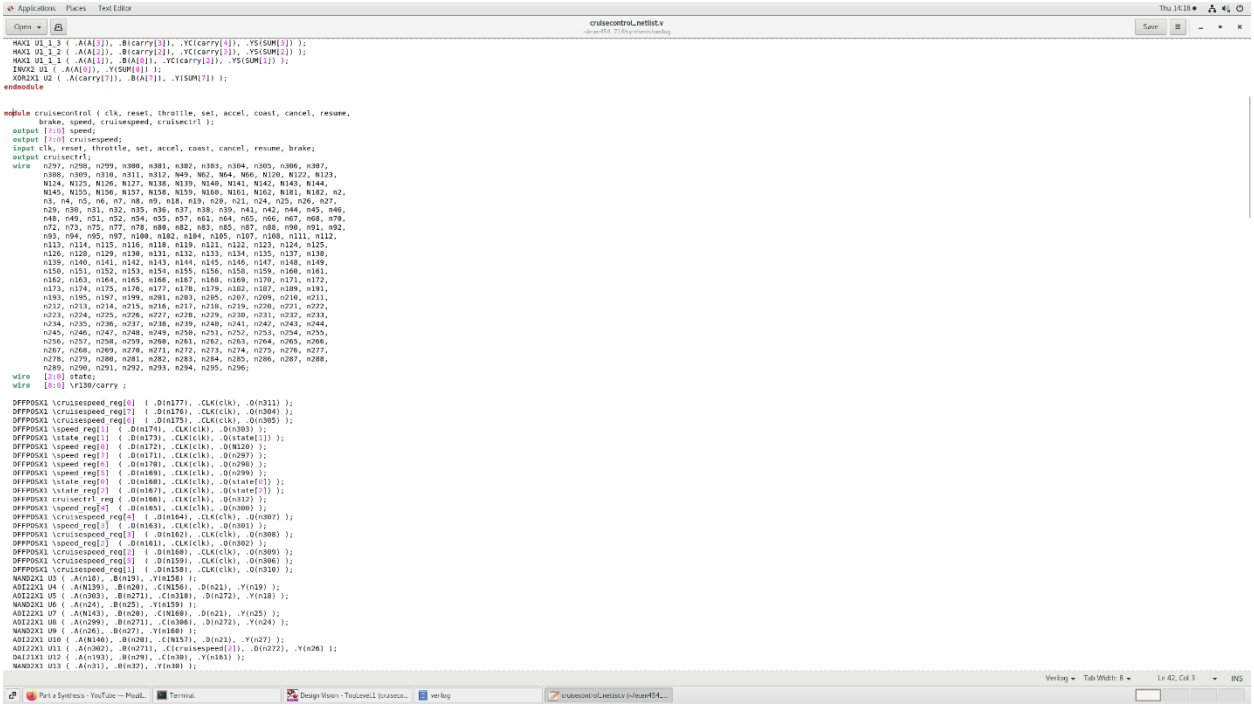
Design Vision is the graphical interface to the Synopsys family of logic synthesis tools. Internally it runs

design compiler. This lab is to familiarize you with the basics of synthesis using Design Vision through a simple example "cruise control design". After synthesis, you will do pre-layout static timing analysis of your synthesized design. You will define constraints for your design and check the timing of all the paths in the design. In the final part of this lab we will use the synthesized Verilog netlist that you will generate for the cruise control logic to "Place and Route" the circuit on a die.

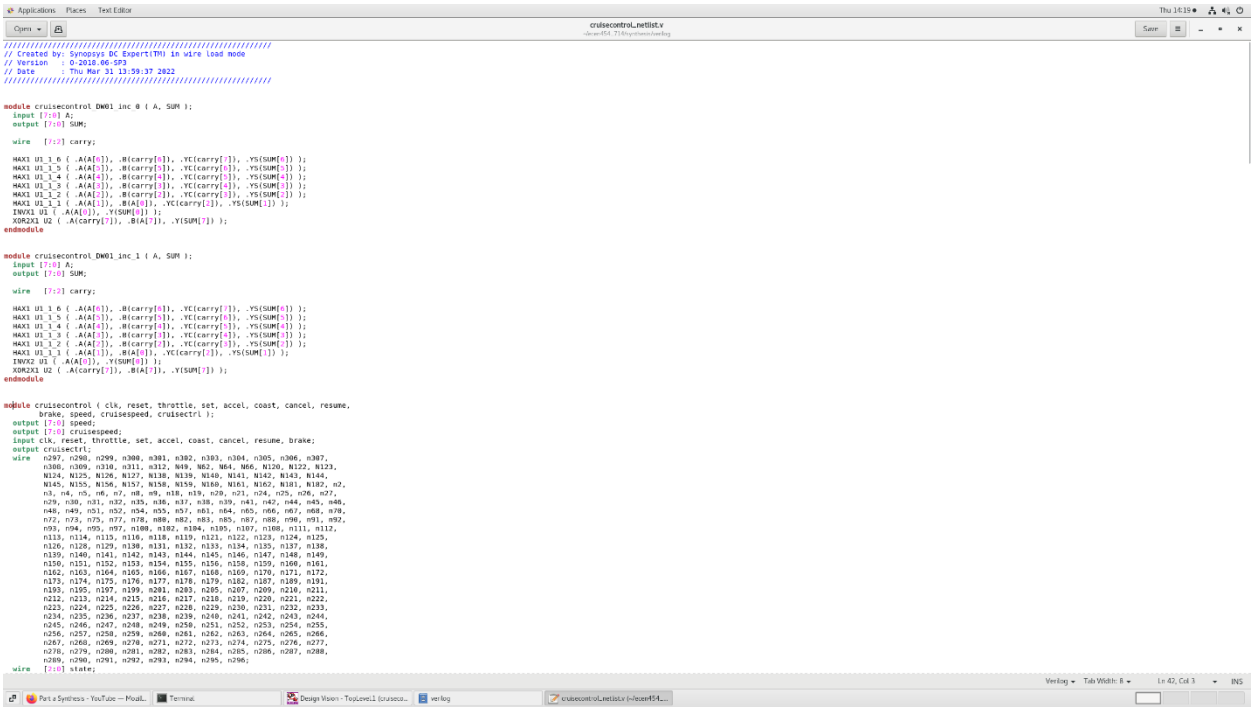
Results:

Part A :





```
HAX1 U1_1_3 ( .A(A[3]), .B(carry[3]), .YC(carry[4]), .YS(SUM[3]) );
HAX1 U1_1_2 ( .A(A[2]), .B(carry[2]), .YC(carry[3]), .YS(SUM[2]) );
HAX1 U1_1_1 ( .A(A[1]), .B(A[0]), .YC(carry[2]), .YS(SUM[1]) );
INVX2 U1 ( .A(A[0]), .Y(SUM[0]) );
XOR2X1 U2 ( .A(carry[7]), .B(A[7]), .Y(SUM[7]) );
endmodule


module cruisecontrol ( clk, reset, throttle, set, accel, coast, cancel, resume,
        brake, speed, cruisespeed, cruisectrl );
  output [7:0] speed;
  output [7:0] cruisespeed;
  input clk, reset, throttle, set, accel, coast, cancel, resume, brake;
  output cruisectrl;
  wire   n297, n298, n299, n300, n301, n302, n303, n304, n305, n306, n307,
         n308, n309, n310, n311, n312, N49, N62, N64, N66, N120, N122, N123,
         N124, N125, N126, N127, N138, N139, N140, N141, N142, N143, N144,
         N145, N155, N156, N157, N158, N159, N160, N161, N162, N181, N182, n2,
         n3, n4, n5, n6, n7, n8, n9, n18, n19, n20, n21, n24, n25, n26, n27,
         n29, n30, n31, n32, n35, n36, n37, n38, n39, n41, n42, n44, n45, n46,
         n48, n49, n51, n52, n54, n55, n57, n61, n64, n65, n66, n67, n68, n70,
         n72, n73, n75, n77, n78, n88, n82, n83, n85, n87, n88, n90, n91, n92,
         n93, n94, n95, n97, n109, n102, n104, n105, n107, n108, n111, n112,
         n113, n114, n115, n116, n118, n119, n121, n122, n123, n124, n125,
         n126, n128, n129, n130, n131, n132, n133, n134, n135, n137, n138,
         n139, n140, n141, n142, n143, n144, n145, n146, n147, n148, n149,
         n150, n151, n152, n153, n154, n155, n156, n158, n159, n160, n161,
         n162, n163, n164, n165, n166, n167, n168, n169, n170, n171, n172,
         n173, n174, n175, n176, n177, n178, n179, n182, n187, n189, n191,
         n193, n195, n197, n199, n201, n203, n205, n207, n209, n210, n211,
         n212, n213, n214, n215, n216, n217, n218, n219, n220, n221, n222,
         n223, n224, n225, n226, n227, n228, n229, n230, n231, n232, n233,
         n234, n235, n236, n237, n238, n239, n240, n241, n242, n243, n244,
         n245, n246, n247, n248, n249, n250, n251, n252, n253, n254, n255,
         n256, n257, n258, n259, n260, n261, n262, n263, n264, n265, n266,
         n267, n268, n269, n270, n271, n272, n273, n274, n275, n276, n277,
         n278, n279, n280, n281, n282, n283, n284, n285, n286, n287, n288,
         n289, n290, n291, n292, n293, n294, n295, n296;
  wire   [2:0] state;
  wire   [8:0] \r130/carry ;

  DFFPOSX1 \cruisespeed_reg[0]  ( .D(n177), .CLK(clk), .Q(n311) );
  DFFPOSX1 \cruisespeed_reg[7]  ( .D(n176), .CLK(clk), .Q(n304) );
  DFFPOSX1 \cruisespeed_reg[6]  ( .D(n175), .CLK(clk), .Q(n305) );
  DFFPOSX1 \speed_reg[1]  ( .D(n174), .CLK(clk), .Q(n303) );
  DFFPOSX1 \state_reg[1]  ( .D(n173), .CLK(clk), .Q(state[1]) );
  DFFPOSX1 \speed_reg[0]  ( .D(n172), .CLK(clk), .Q(N120) );
  DFFPOSX1 \speed_reg[7]  ( .D(n171), .CLK(clk), .Q(n297) );
  DFFPOSX1 \speed_reg[6]  ( .D(n170), .CLK(clk), .Q(n298) );
  DFFPOSX1 \speed_reg[5]  ( .D(n169), .CLK(clk), .Q(n299) );
  DFFPOSX1 \state_reg[0]  ( .D(n168), .CLK(clk), .Q(state[0]) );
  DFFPOSX1 \state_reg[2]  ( .D(n167), .CLK(clk), .Q(state[2]) );
  DFFPOSX1 cruisectrl_reg ( .D(n166), .CLK(clk), .Q(n312) );
  DFFPOSX1 \speed_reg[4]  ( .D(n165), .CLK(clk), .Q(n300) );
  DFFPOSX1 \cruisespeed_reg[4]  ( .D(n164), .CLK(clk), .Q(n307) );
  DFFPOSX1 \speed_reg[3]  ( .D(n163), .CLK(clk), .Q(n301) );
  DFFPOSX1 \cruisespeed_reg[3]  ( .D(n162), .CLK(clk), .Q(n308) );
  DFFPOSX1 \speed_reg[2]  ( .D(n161), .CLK(clk), .Q(n302) );
  DFFPOSX1 \cruisespeed_reg[2]  ( .D(n160), .CLK(clk), .Q(n309) );
  DFFPOSX1 \cruisespeed_reg[5]  ( .D(n159), .CLK(clk), .Q(n306) );
  DFFPOSX1 \cruisespeed_reg[1]  ( .D(n158), .CLK(clk), .Q(n310) );
  NAND2X1 U3 ( .A(n18), .B(n19), .Y(n158) );
  AOI22X1 U4 ( .A(N139), .B(n20), .C(N156), .D(n21), .Y(n19) );
  AOI22X1 U5 ( .A(n303), .B(n271), .C(n310), .D(n272), .Y(n18) );
  NAND2X1 U6 ( .A(n24), .B(n25), .Y(n159) );
  AOI22X1 U7 ( .A(N143), .B(n20), .C(N160), .D(n21), .Y(n25) );
  AOI22X1 U8 ( .A(n299), .B(n271), .C(n306), .D(n272), .Y(n24) );
  NAND2X1 U9 ( .A(n26), .B(n27), .Y(n160) );
  AOI22X1 U10 ( .A(N140), .B(n20), .C(N157), .D(n21), .Y(n27) );
  AOI22X1 U11 ( .A(n302), .B(n271), .C(\r130/carry[3]), .D(n272), .Y(n26) );
  OAI21X1 U12 ( .A(n193), .B(n29), .C(n30), .Y(n161) );
  NAND2X1 U13 ( .A(n31), .B(n32), .Y(n30) );
```



```
////////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version   : O-2018.06-SP3
// Date      : Thu Mar 31 13:59:37 2022
////////////////////////////////////////////////////////////////////


module cruisecontrol_DW01_inc_0 ( A, SUM );
  input [7:0] A;
  output [7:0] SUM;

  wire   [7:2] carry;

  HAX1 U1_1_6 ( .A(A[6]), .B(carry[6]), .YC(carry[7]), .YS(SUM[6]) );
  HAX1 U1_1_5 ( .A(A[5]), .B(carry[5]), .YC(carry[6]), .YS(SUM[5]) );
  HAX1 U1_1_4 ( .A(A[4]), .B(carry[4]), .YC(carry[5]), .YS(SUM[4]) );
  HAX1 U1_1_3 ( .A(A[3]), .B(carry[3]), .YC(carry[4]), .YS(SUM[3]) );
  HAX1 U1_1_2 ( .A(A[2]), .B(carry[2]), .YC(carry[3]), .YS(SUM[2]) );
  HAX1 U1_1_1 ( .A(A[1]), .B(A[0]), .YC(carry[2]), .YS(SUM[1]) );
  INVX1 U1 ( .A(A[0]), .Y(SUM[0]) );
  XOR2X1 U2 ( .A(carry[7]), .B(A[7]), .Y(SUM[7]) );
endmodule


module cruisecontrol_DW01_inc_1 ( A, SUM );
  input [7:0] A;
  output [7:0] SUM;

  wire   [7:2] carry;

  HAX1 U1_1_6 ( .A(A[6]), .B(carry[6]), .YC(carry[7]), .YS(SUM[6]) );
  HAX1 U1_1_5 ( .A(A[5]), .B(carry[5]), .YC(carry[6]), .YS(SUM[5]) );
  HAX1 U1_1_4 ( .A(A[4]), .B(carry[4]), .YC(carry[5]), .YS(SUM[4]) );
  HAX1 U1_1_3 ( .A(A[3]), .B(carry[3]), .YC(carry[4]), .YS(SUM[3]) );
  HAX1 U1_1_2 ( .A(A[2]), .B(carry[2]), .YC(carry[3]), .YS(SUM[2]) );
  HAX1 U1_1_1 ( .A(A[1]), .B(A[0]), .YC(carry[2]), .YS(SUM[1]) );
  INVX2 U1 ( .A(A[0]), .Y(SUM[0]) );
  XOR2X1 U2 ( .A(carry[7]), .B(A[7]), .Y(SUM[7]) );
endmodule


module cruisecontrol ( clk, reset, throttle, set, accel, coast, cancel, resume,
        brake, speed, cruisespeed, cruisectrl );
  output [7:0] speed;
  output [7:0] cruisespeed;
  input clk, reset, throttle, set, accel, coast, cancel, resume, brake;
  output cruisectrl;
  wire   n297, n298, n299, n300, n301, n302, n303, n304, n305, n306, n307,
         n308, n309, n310, n311, n312, N49, N62, N64, N66, N120, N122, N123,
         N124, N125, N126, N127, N138, N139, N140, N141, N142, N143, N144,
         N145, N155, N156, N157, N158, N159, N160, N161, N162, N181, N182, n2,
         n3, n4, n5, n6, n7, n8, n9, n18, n19, n20, n21, n24, n25, n26, n27,
         n29, n30, n31, n32, n35, n36, n37, n38, n39, n41, n42, n44, n45, n46,
         n48, n49, n51, n52, n54, n55, n57, n61, n64, n65, n66, n67, n68, n70,
         n72, n73, n75, n77, n78, n88, n82, n83, n85, n87, n88, n90, n91, n92,
         n93, n94, n95, n97, n109, n102, n104, n105, n107, n108, n111, n112,
         n113, n114, n115, n116, n118, n119, n121, n122, n123, n124, n125,
         n126, n128, n129, n130, n131, n132, n133, n134, n135, n137, n138,
         n139, n140, n141, n142, n143, n144, n145, n146, n147, n148, n149,
         n150, n151, n152, n153, n154, n155, n156, n158, n159, n160, n161,
         n162, n163, n164, n165, n166, n167, n168, n169, n170, n171, n172,
         n173, n174, n175, n176, n177, n178, n179, n182, n187, n189, n191,
         n193, n195, n197, n199, n201, n203, n205, n207, n209, n210, n211,
         n212, n213, n214, n215, n216, n217, n218, n219, n220, n221, n222,
         n223, n224, n225, n226, n227, n228, n229, n230, n231, n232, n233,
         n234, n235, n236, n237, n238, n239, n240, n241, n242, n243, n244,
         n245, n246, n247, n248, n249, n250, n251, n252, n253, n254, n255,
         n256, n257, n258, n259, n260, n261, n262, n263, n264, n265, n266,
         n267, n268, n269, n270, n271, n272, n273, n274, n275, n276, n277,
         n278, n279, n280, n281, n282, n283, n284, n285, n286, n287, n288,
         n289, n290, n291, n292, n293, n294, n295, n296;
  wire   [2:0] state;
```

Register count : 20

Part B:

min_paths - Notepad

File Edit Format View Help

```
****************************************
Report : timing
        -path_type full
        -delay_type min
        -slack_lesser_than 0.00
        -max_paths 3
        -sort_by slack
Design : cruisecontrol
Version: O-2018.06-SP3
Date   : Fri Apr  1 15:33:28 2022
****************************************

No paths with slack less than 0.00.

1
```

max_paths - Notepad

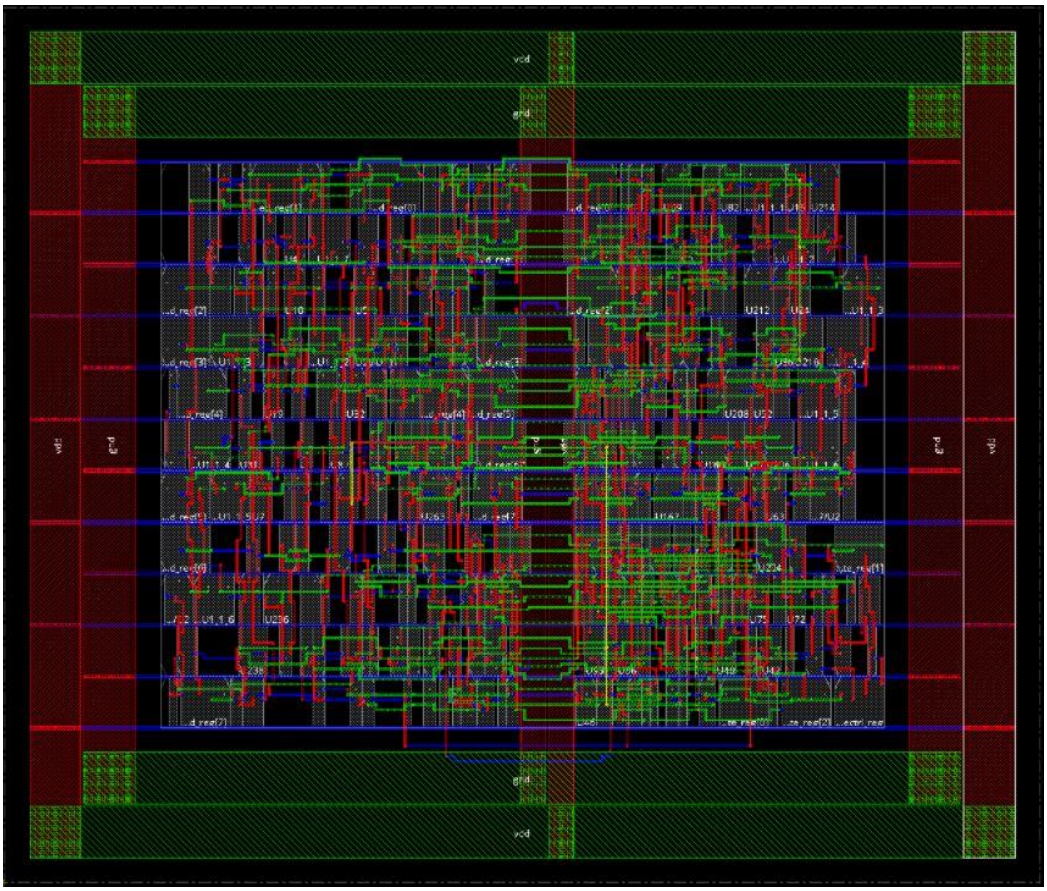
File Edit Format View Help

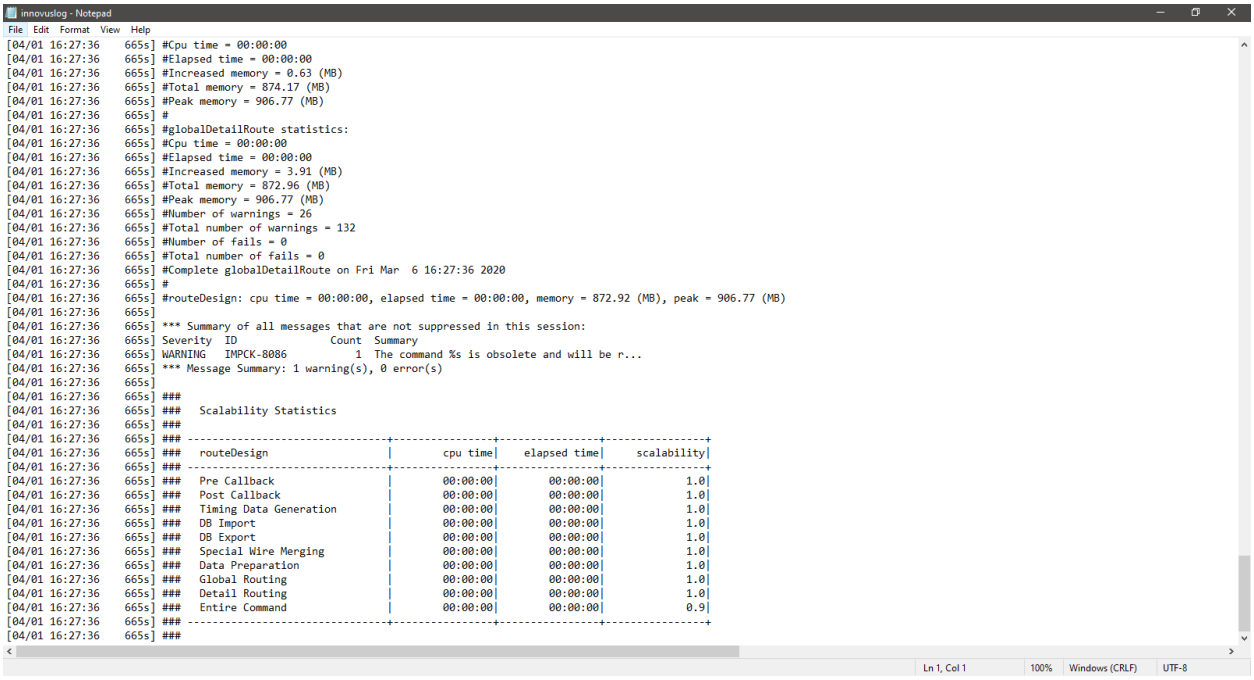
```
****************************************
Report : timing
        -path_type full
        -delay_type max
        -slack_lesser_than 0.00
        -max_paths 3
        -sort_by slack
Design : cruisecontrol
Version: O-2018.06-SP3
Date   : Fri Apr  1 15:33:17 2022
****************************************

No paths with slack less than 0.00.

1
```

Ln 1, Col 1 100% Unix (LF) UTF-8

Ln 1, Col 1 100% Unix (LF) UTF-8

Part C:





#Total wire length = 9374 um.

#Total number of vias = 1436

[04/01 16:27:36 664s] Placement Density:69.78%(10808/15488)

[04/01 16:27:36 664s] Placement Density (including fixed std cells):69.78%(10808/15488)