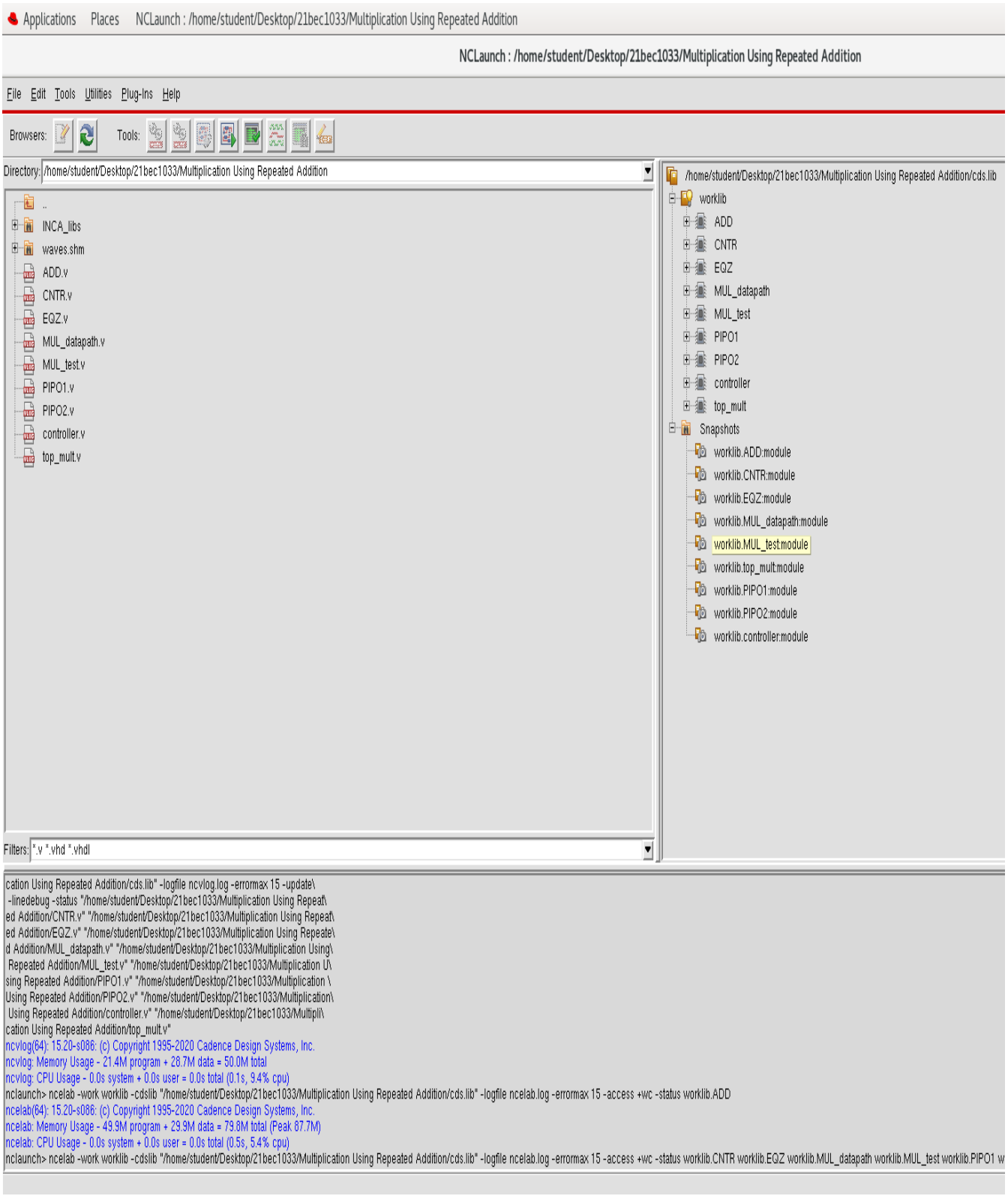
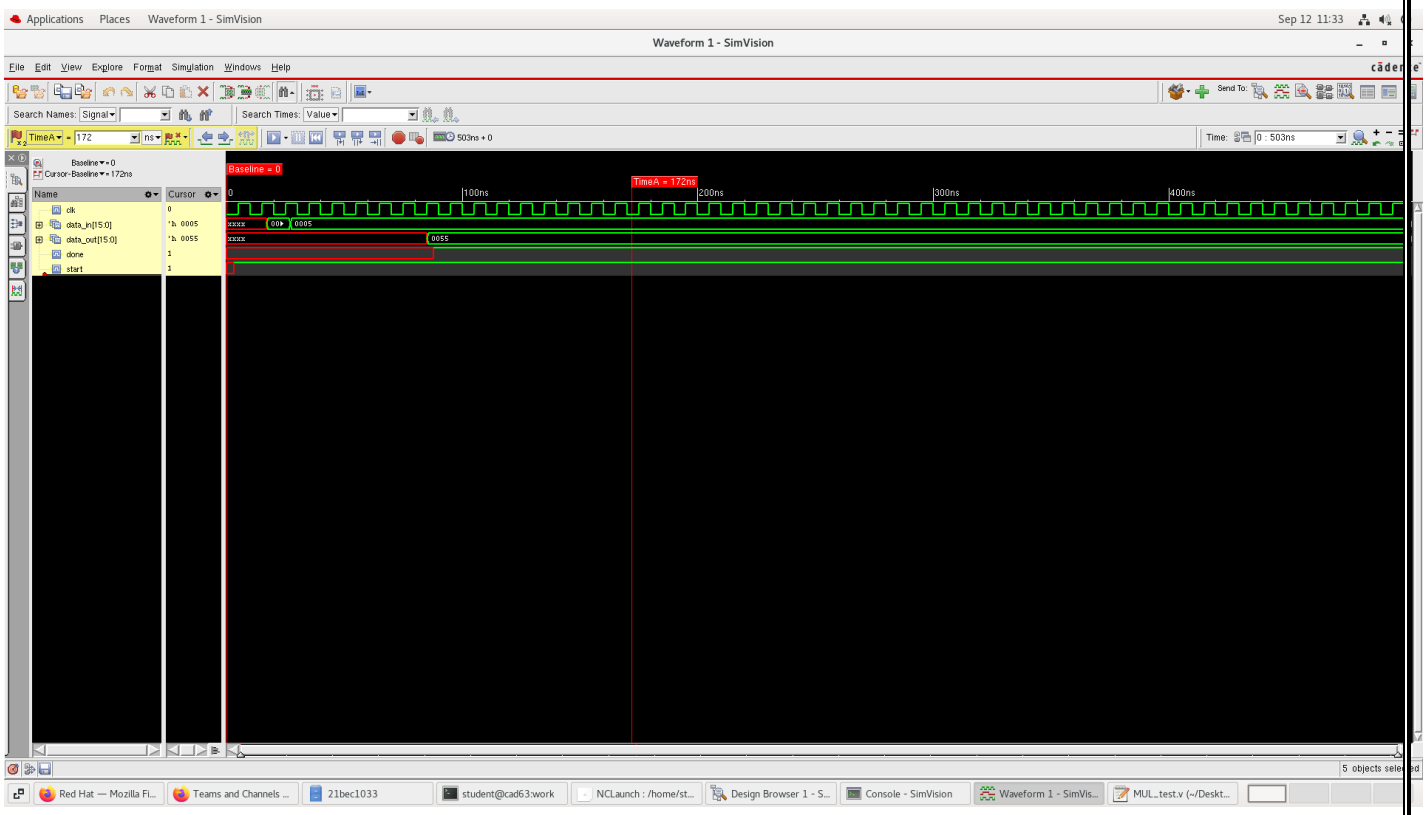
Lab4_1: Design and simulation of a multiplier by repeated addition using Datapath and Control path logic

NCLaunch Compilation and Elaboration:



Lab4_1: Design and simulation of a multiplier by repeated addition using Datapath and Control path logic

Observations/Simulation Waveforms:



Inference: In the above output waveform, the data_in is provided with 5 & 11 values and the data_out is obtained as 55 which is the multiplied result after repeated addition. Output “done” is equal to 1 whenever multiplication result is obtained.

Result: Hence, a multiplier is designed and the simulation and verification of behavior of multiplier by repeated addition using data path and Control path logic is performed for the given stimuli in the test bench.

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Aim: To synthesize a multiplier by repeated addition using Datapath and Control path logic and to get the gate level netlist with timing, area, and power reports.

EDA Tools Used: Cadence Genus

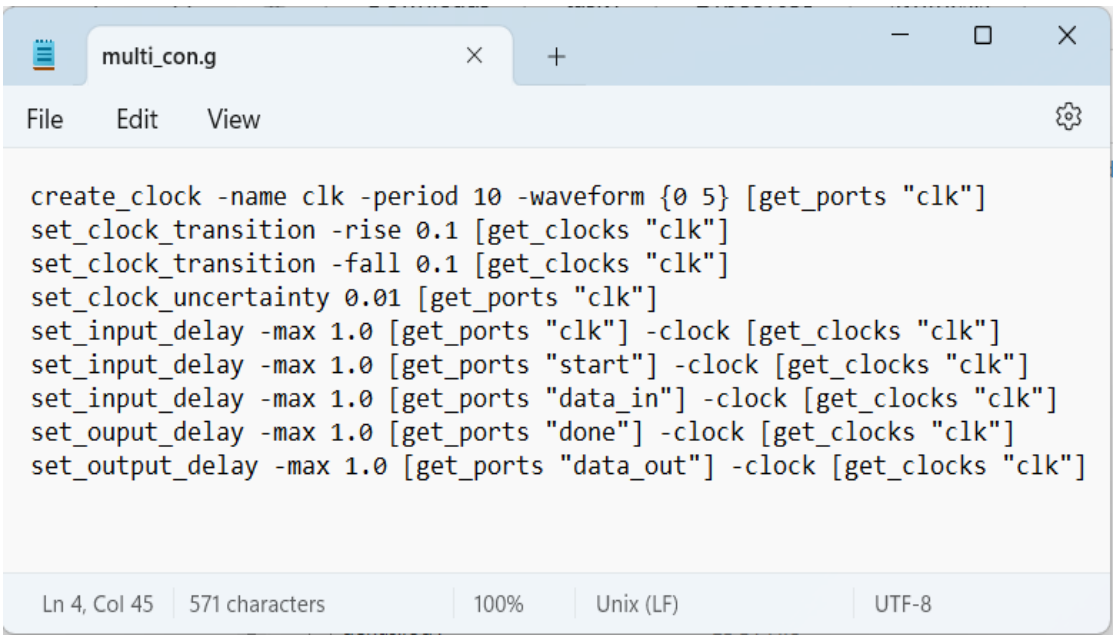
Description:

Datapath is the hardware that performs all the required operations. Control is the hardware that tells the datapath what to do, in terms of switching, operation selection, data movement between ALU components, etc. Simple datapath components include memory (stores the current instruction), PC or program counter (stores the address of current instruction), and ALU (executes current instruction). Control path is the Finite state machine designed by using Moore or Mealy models.

Procedure:

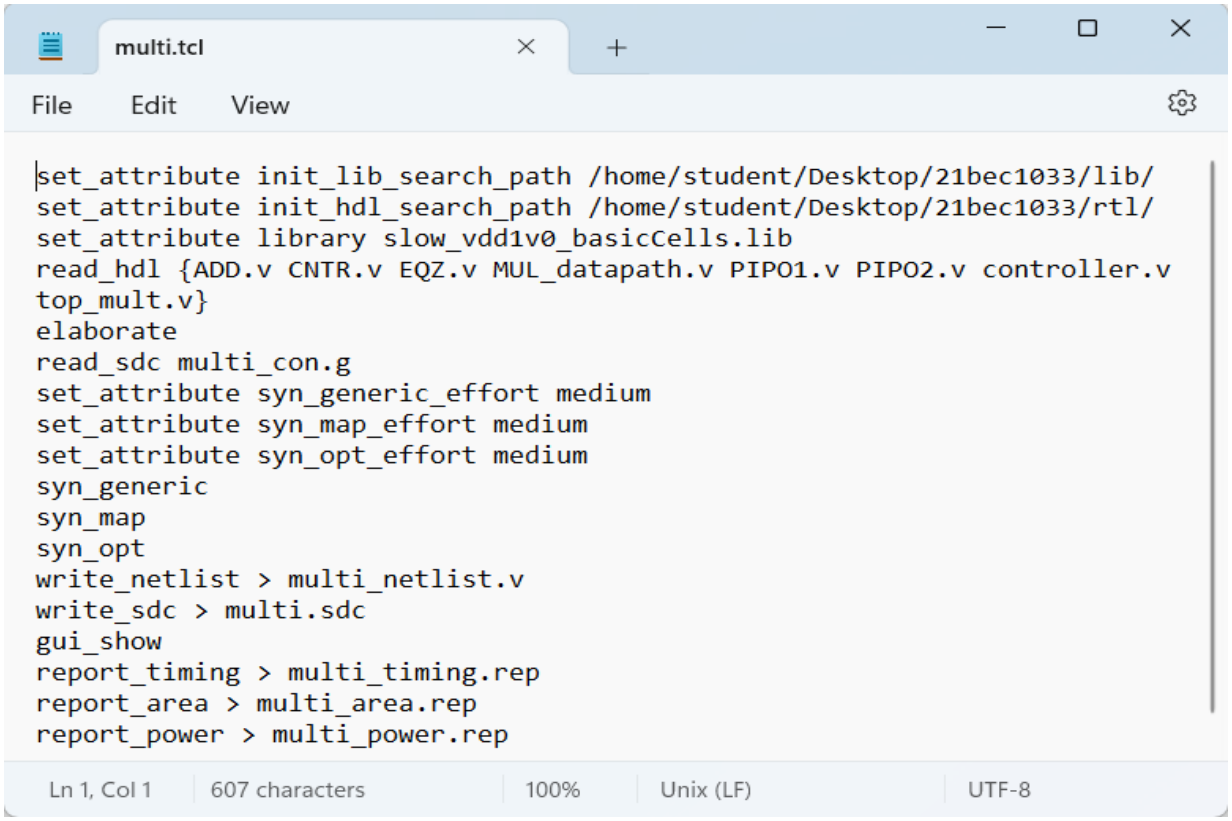
1. Copy the fast.lib and slow.lib files into the folder where (.v) files are located.

2. Create a Synopsys design constraint file (.g file) by entering the design constraints required for the synthesis.



LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

3. Create a (.tcl) file containing all the commands for performing the logic synthesis. Synthesis effort can be medium or high.



4. Invoke the C shell and launch the Genus tool by entering the below commands.
   'csh'
   'source /home/install/cshrc'
   'genus -legacy -ui -f multi.tcl'

5. Execute the commands in the (.tcl file) by entering **source multiplier.tcl** command in the command line window.

6. Check for the area, timing and power reports generated in the respective multiplier folder. Also check the gate level netlist generated in the Genus synthesis solution window.

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Verilog Programs:

//Verilog Program of top-level module containing data path and control path

top_mult.v

File Edit View

```
module top_mult(data_out, done, data_in, clk, start);
output done;
output [15:0] data_out;
input clk, start;
input [15:0]data_in;
wire eqz, LdA, LdB, LdP, clrP, decB;
MUL_datapath uut1 (data_out, eqz, LdA, LdB, LdP, clrP, decB, data_in,clk);
controller uut2(LdA, LdB, LdP, clrP, decB, done, clk, eqz, start);
endmodule
```

Ln 1, Col 1 | 321 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of control path module

controller.v

File Edit View

```
module controller (LdA,LdB,LdP,clrP,decB,done,clk,eqz,start);
input clk,eqz,start;
output reg LdA,LdB,LdP,clrP,decB,done;
reg[2:0] state;
parameter S0 = 3'b000, S1 = 3'b001, S2 = 3'b010, S3 = 3'b011, S4 = 3'b100;
always @ (posedge clk)
begin
case(state)
S0: if(start) state <= S1;
S1: state <= S2;
S2: state <= S3;
S3: #2 if (eqz) state <= S4;
S4: state <= S4;
default: state <= S0;
endcase
end

always @ (state)
begin
case(state)
S0: begin #1 LdA = 0; LdB = 0; LdP = 0; clrP = 0; decB = 0; end
S1: begin #1 LdA = 1; end
S2: begin #1 LdA = 0; LdB = 1; clrP = 1; end
S3: begin #1 LdB = 0; LdP = 1; clrP = 0; decB = 1; end
S4: begin #1 done = 1; LdB = 0; LdP = 0; decB = 0; end
default: begin #1 LdA = 0; LdB = 0; LdP = 0; clrP = 0; decB = 0; end
endcase
end
endmodule
```

Ln 1, Col 1 | 767 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

//Verilog Program of data path module

controller.v | MUL_datapath.v

File Edit View

```
module MUL_datapath(data_out,eqz,LdA,LdB,LdP,clrP,decB,data_in,clk);
input LdA,LdB,LdP,clrP,decB,clk;
input [15:0] data_in;
output eqz;
output reg [15:0] data_out;
wire [15:0] X,Y,Z,Bout;
always @ (eqz)
begin
if(eqz == 1)
data_out <= Y;
else
data_out <= data_out;
end

PIPO1 A (X,data_in,LdA,clk);
PIPO2 P (Y,Z,LdP,clrP,clk);
CNTR B (Bout,data_in,LdB,decB,clk);
ADD AD (Z,X,Y);
EQZ COMP (eqz, Bout);
endmodule
```

Ln 1, Col 1 | 411 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of PIPO1

PIPO1.v

File Edit View

```
module PIPO1(dout,din,ld,clk);
input[15:0] din;
input ld, clk;
output reg [15:0] dout;
always @ (posedge clk)
if(ld) dout <= din;
endmodule
```

Ln 1, Col 1 | 141 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

//Verilog Program of PIPO2

PIPO1.v PIPO2.

File Edit View

```
module PIPO2(dout,din,ld,clr,clk);
input[15:0] din;
input ld, clk,clr;
output reg [15:0] dout;
always @ (posedge clk)
if(clr) dout <= 16'b0;
else if(ld) dout <= din;
endmodule
```

Ln 1, Col 1 | 177 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of Adder

ADD.

File Edit View

```
module ADD(out,in1,in2);
input [15:0] in1,in2;
output reg[15:0] out;
always @ (*)
out = in1 + in2;
endmodule
```

Ln 1, Col 1 | 109 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of Counter

CNTR

File Edit View

```
module CNTR(dout,din,ld,dec,clk);
input [15:0] din;
input ld, dec,clk;
output reg[15:0] dout;
always @ (posedge clk)
if(ld) dout <= din;
else if(dec) dout <= dout - 1;
endmodule
```

Ln 1, Col 1 | 178 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

//Verilog Program of Comparator

EQZ.v

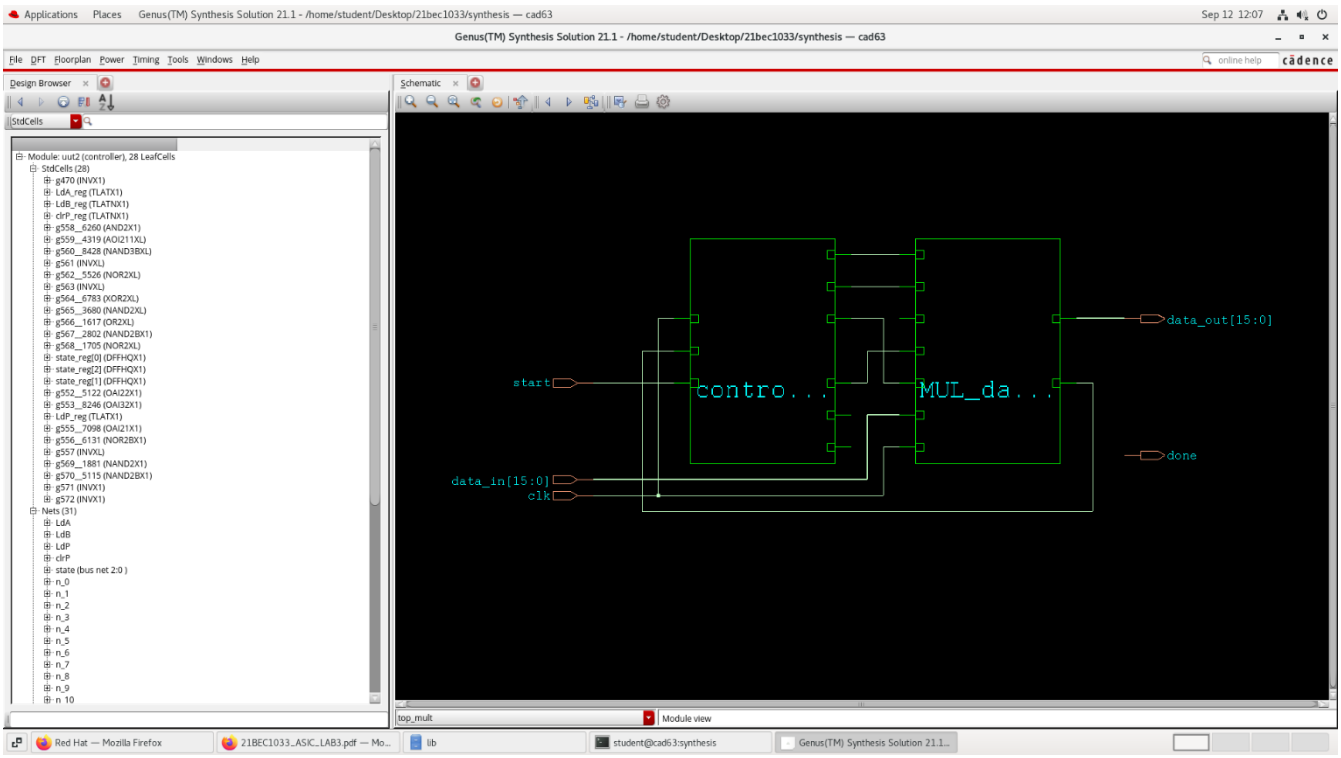
File Edit View

```
module EQZ(eqz,data);
input [15:0] data;
output eqz;
assign eqz = (data == 0);
endmodule
```

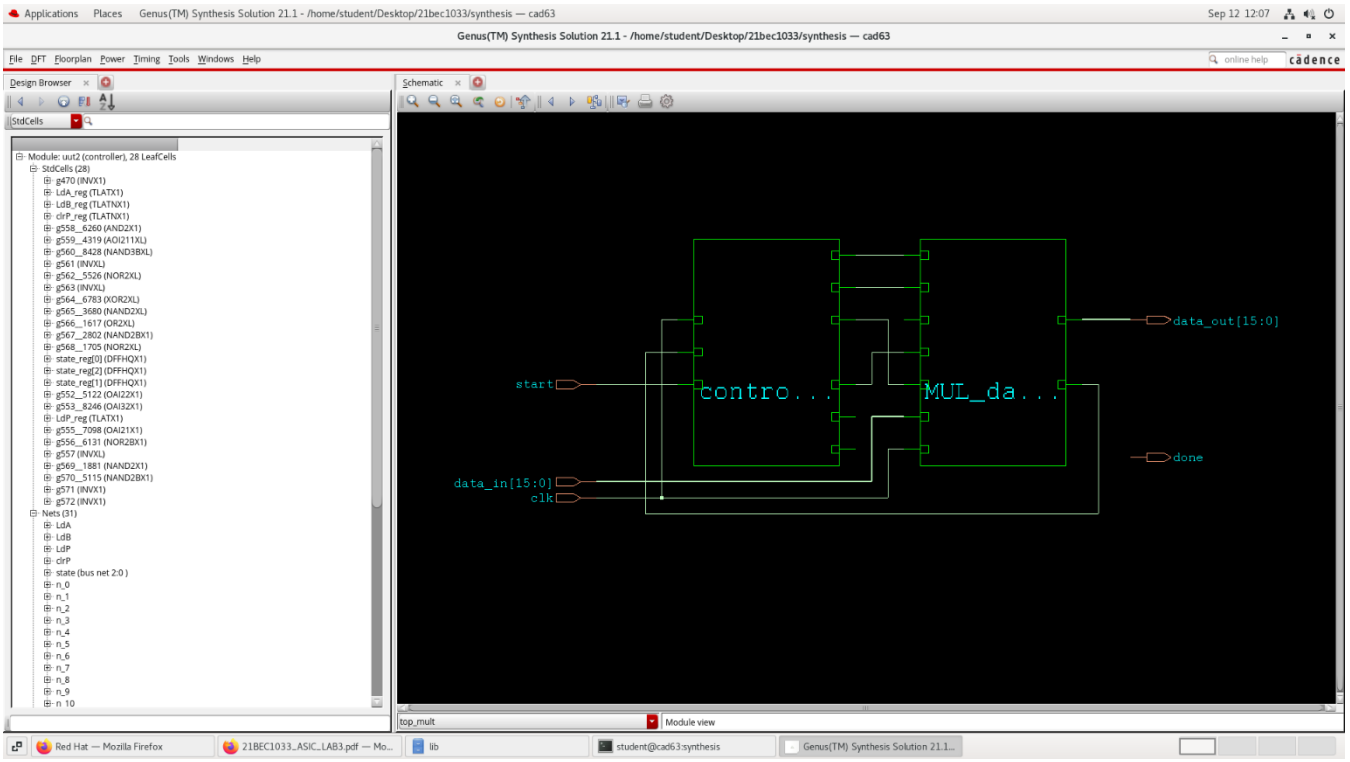
Ln 1, Col 1 | 89 characters | 100% | Unix (LF) | UTF-8

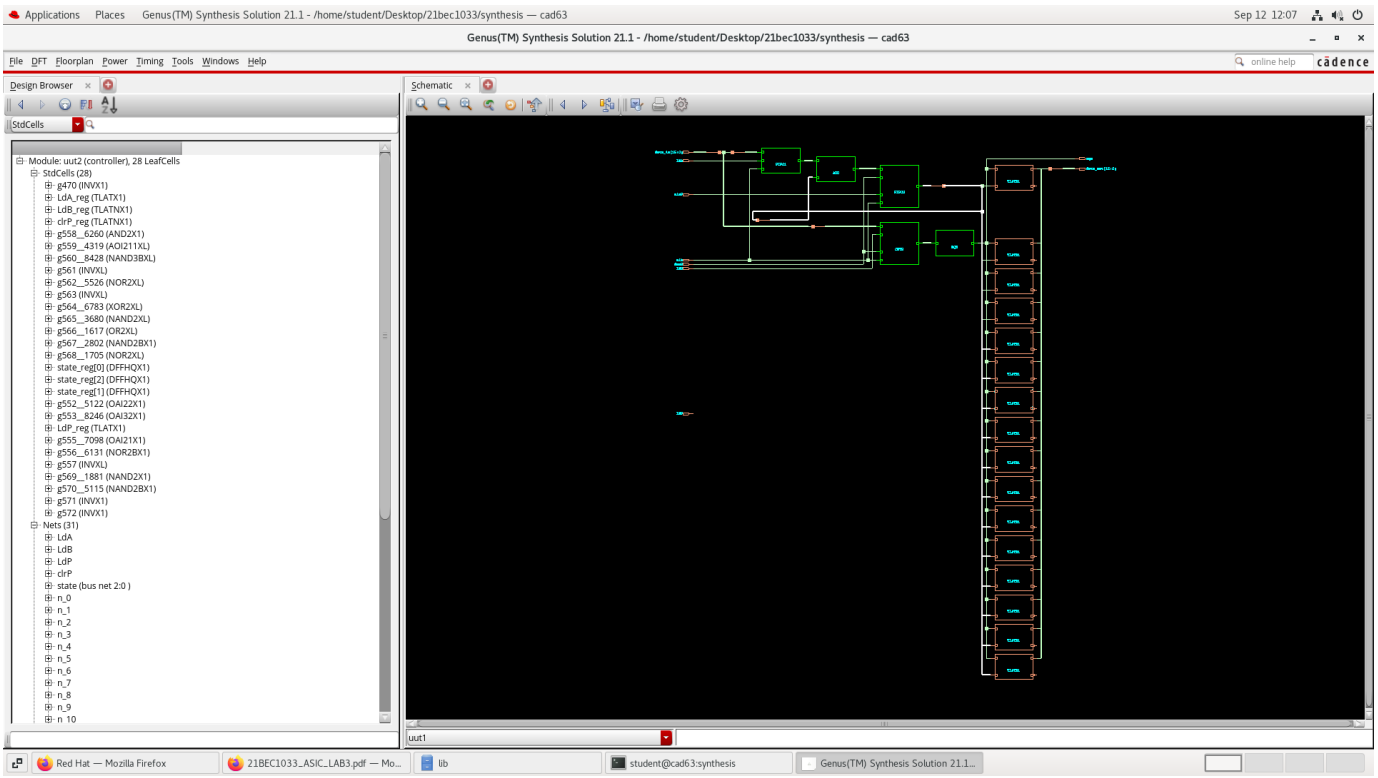
Gate level Netlist:

Synthesis effort: Medium



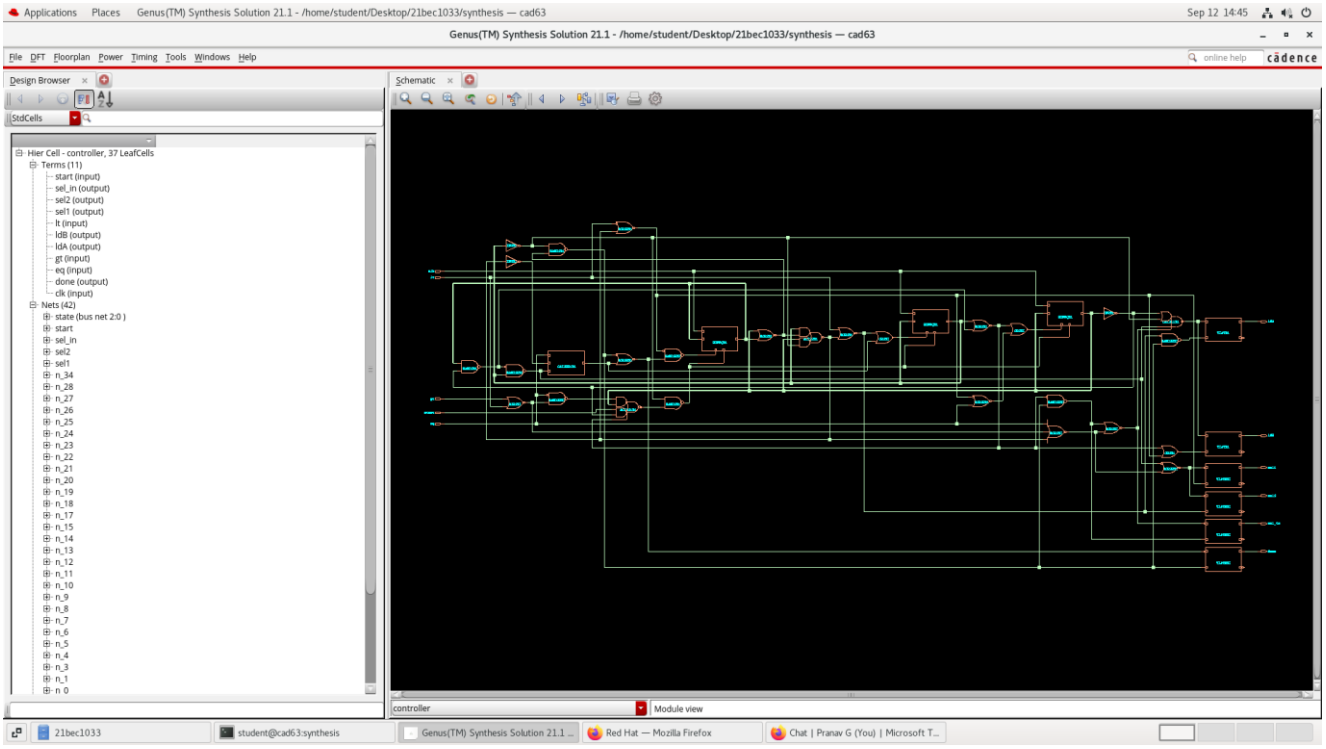
LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic





LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

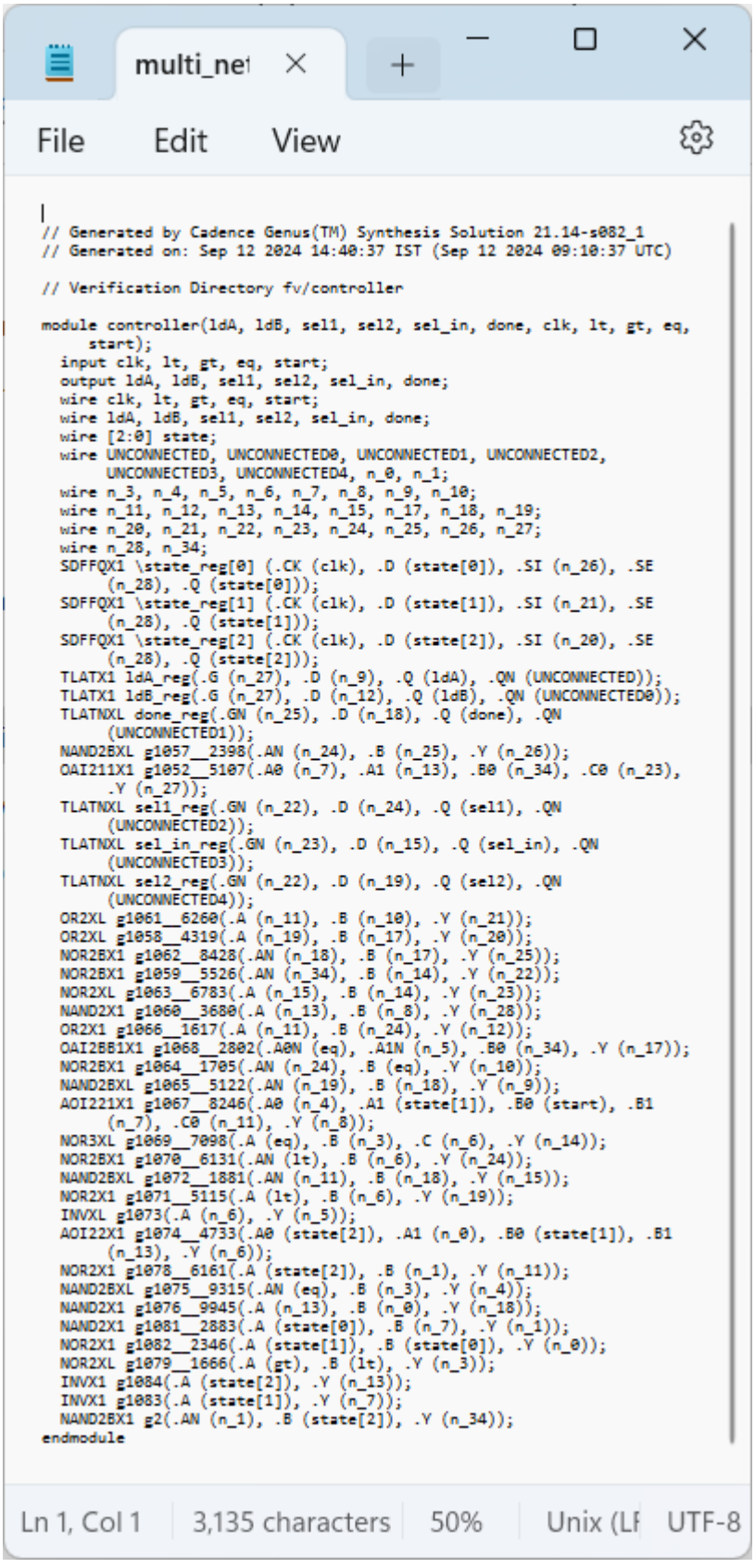
Synthesis effort : High



LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

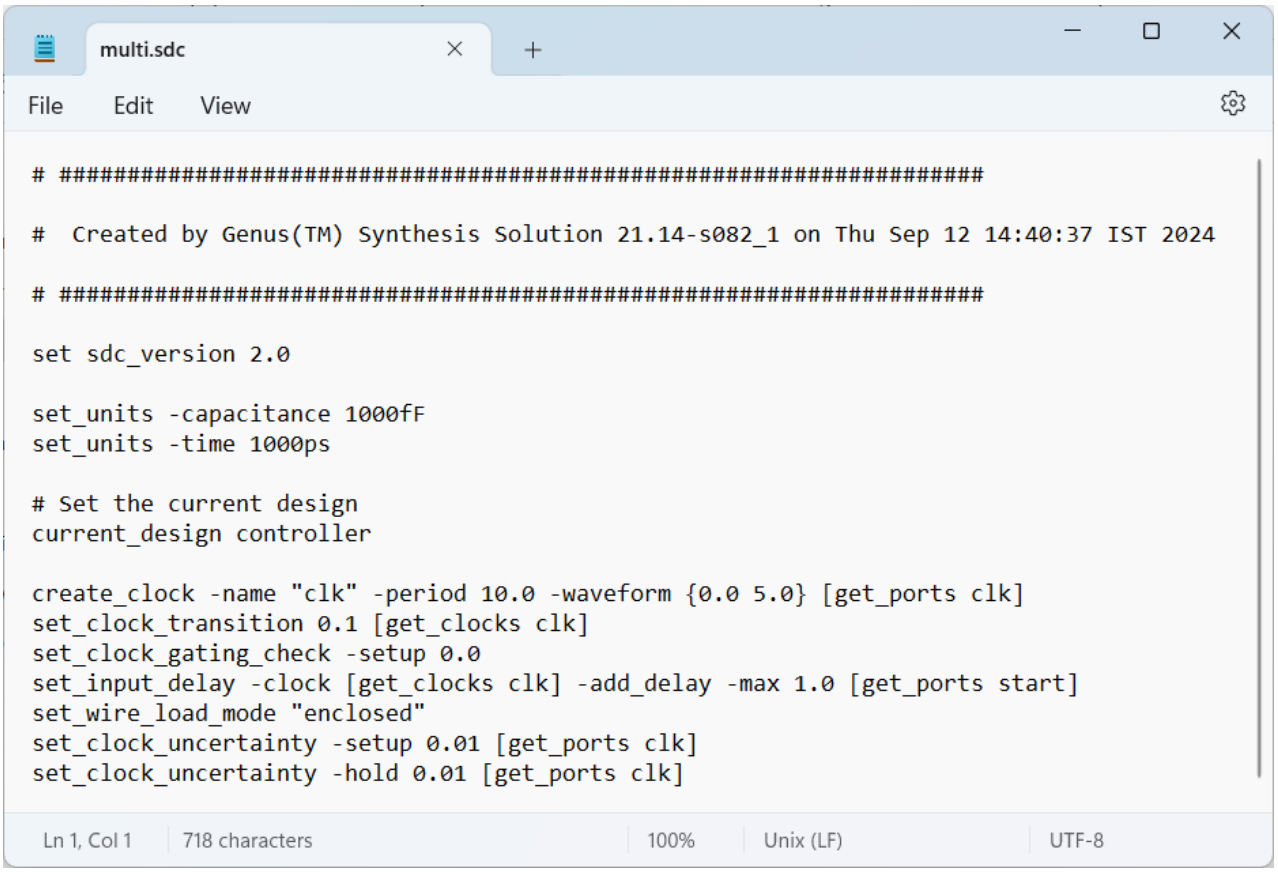
Observations:

a) multiplier_netlist_created file



LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

b) multiplier_constraint_created file



LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

c) multiplier_timing_created file

Synthesis effort: Medium

multi_tim

File Edit View

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  12:05:27 pm
  Module:                 top_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

          Pin                    Type       Fanout Load Slew Delay Arrival
                                                   (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------
(clock clk)                    launch                                    0 R
uut1
  A
    dout_reg[0]/CK                                          100    +0       0 R
    dout_reg[0]/Q              SDFFQX1          3  0.6     24  +225     225 R
  A/dout[0]
  AD/in1[0]
    add_5_11_g383__7482/A                                                +0     225
    add_5_11_g383__7482/Y      AND2XL           2  1.0     49  +124     349 R
    add_5_11_g380__6131/CI                                               +0     349
    add_5_11_g380__6131/CO     ADDFX1           1  0.6     39  +191     540 R
    add_5_11_g379__7098/CI                                               +0     540
    add_5_11_g379__7098/CO     ADDFX1           1  0.6     39  +186     726 R
    add_5_11_g378__8246/CI                                               +0     726
    add_5_11_g378__8246/CO     ADDFX1           1  0.6     39  +186     912 R
    add_5_11_g377__5122/CI                                               +0     912
    add_5_11_g377__5122/CO     ADDFX1           1  0.6     39  +186    1098 R
    add_5_11_g376__1705/CI                                               +0    1098
    add_5_11_g376__1705/CO     ADDFX1           1  0.6     39  +186    1284 R
    add_5_11_g375__2802/CI                                               +0    1284
    add_5_11_g375__2802/CO     ADDFX1           1  0.6     39  +186    1470 R
    add_5_11_g374__1617/CI                                               +0    1470
    add_5_11_g374__1617/CO     ADDFX1           1  0.6     39  +186    1656 R
    add_5_11_g373__3680/CI                                               +0    1656
    add_5_11_g373__3680/CO     ADDFX1           1  0.6     39  +186    1842 R
    add_5_11_g372__6783/CI                                               +0    1842
    add_5_11_g372__6783/CO     ADDFX1           1  0.6     39  +186    2028 R
    add_5_11_g371__5526/CI                                               +0    2028
    add_5_11_g371__5526/CO     ADDFX1           1  0.6     39  +186    2214 R
    add_5_11_g370__8428/CI                                               +0    2214
    add_5_11_g370__8428/CO     ADDFX1           1  0.6     39  +186    2400 R
    add_5_11_g369__4319/CI                                               +0    2400
    add_5_11_g369__4319/CO     ADDFX1           1  0.6     39  +186    2586 R
    add_5_11_g368__6260/CI                                               +0    2586
    add_5_11_g368__6260/CO     ADDFX1           1  0.6     39  +186    2772 R
    add_5_11_g367__5107/CI                                               +0    2772
    add_5_11_g367__5107/CO     ADDFX1           1  0.4     35  +184    2956 R
    add_5_11_g366__2398/B                                                +0    2956
    add_5_11_g366__2398/Y      XNOR2X1          1  0.4     26  +125    3081 R
  AD/out[15]
  P/din[15]
    g80__5477/B1                                                         +0    3081
    g80__5477/Y                AOI22X1          1  0.3    105   +57    3138 F
    g64__5122/B                                                          +0    3138
    g64__5122/Y                NOR2X1           1  0.3     49   +84    3223 R
    dout_reg[15]/D        <<<  DFFHQX1                                   +0    3223
    dout_reg[15]/CK            setup                                   100  +101    3323 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                    capture                                       10000 R
                               uncertainty                              -10   9990 R
-------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    6667ps
Start-point  : uut1/A/dout_reg[0]/CK
End-point    : uut1/P/dout_reg[15]/D
```

Ln 1, Col 1 | 4,446 characters | 50% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Synthesis effort: High

multi_timing.rep

File Edit View

```
╞============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  02:40:37 pm
  Module:                 controller
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

        Pin                 Type       Fanout Load Slew Delay Arrival
                                              (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------
(clock clk)               launch                                    0 R
(multi_con.g_line_6)      ext delay                    +1000     1000 F
start                     in port           1  0.4    0    +0     1000 F
g1067__8246/B0                                           +0     1000
g1067__8246/Y             AOI221X1          1  0.4  101  +110     1110 R
g1060__3680/B                                            +0     1110
g1060__3680/Y             NAND2X1           3  1.2  106  +121     1230 F
state_reg[2]/SE      <<<  SDFFQX1                          +0     1230
state_reg[2]/CK           setup                     100  +233     1464 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)               capture                         10000 R
                          uncertainty                    -10     9990 R
-------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    8526ps
Start-point  : start
End-point    : state_reg[2]/SE
```

Ln 1, Col 1 | 1,740 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

d) multiplier_area_created file

Synthesis effort: Medium

multi_area.rep

File Edit View

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  12:05:27 pm
  Module:                 top_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance    Module     Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------
top_mult                    180    671.688     0.000      671.688 <none> (D)
  uut1   MUL_datapath       152    604.998     0.000      604.998 <none> (D)
    A    PIPO1               16    120.384     0.000      120.384 <none> (D)
    AD   ADD                 18     80.028     0.000       80.028 <none> (D)
    B    CNTR                48    169.632     0.000      169.632 <none> (D)
    COMP EQZ                  5      9.918     0.000        9.918 <none> (D)
    P    PIPO2               49    137.484     0.000      137.484 <none> (D)
  uut2   controller          28     66.690     0.000       66.690 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 1,311 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High

multi_area.rep

File Edit View

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  02:40:37 pm
  Module:                 controller
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

 Instance  Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------
controller              37     91.656     0.000       91.656 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 734 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

e) multiplier_power_created file

Synthesis effort: Medium

multi_power.rep

File Edit View

```
Instance: /top_mult
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
     register     6.03735e-09  9.83431e-06  2.56520e-07  1.00969e-05   76.15%
        latch     2.28749e-09  4.92714e-07  1.81550e-07  6.76552e-07    5.10%
        logic     4.86856e-09  1.31326e-06  3.40639e-07  1.65877e-06   12.51%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
        clock     0.00000e+00  0.00000e+00  8.26200e-07  8.26200e-07    6.23%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
  ------------------------------------------------------------------------
    Subtotal     1.31934e-08  1.16403e-05  1.60491e-06  1.32584e-05   99.99%
  Percentage           0.10%       87.80%       12.10%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Ln 7, Col 76 | 1,202 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High

multi_power.rep

File Edit View

```
Instance: /controller
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
     register     3.54301e-10  1.40330e-06  1.10603e-07  1.51426e-06   40.98%
        latch     5.57475e-10  5.95999e-07  0.00000e+00  5.96557e-07   16.14%
        logic     8.32462e-10  1.04767e-06  4.87399e-07  1.53590e-06   41.56%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
        clock     0.00000e+00  0.00000e+00  4.86000e-08  4.86000e-08    1.32%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
  ------------------------------------------------------------------------
    Subtotal     1.74424e-09  3.04697e-06  6.46602e-07  3.69531e-06  100.00%
  Percentage           0.05%       82.45%       17.50%      100.00% 100.00%
  ------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,204 characters | 100% | Unix (LF) | UTF-8

LAB4_1: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Inference:

Synthesis effort: Medium

A total of 28 leaf instance count is present in the gate level netlist with total area of 671.688, total power of 1.32584e-05 W.

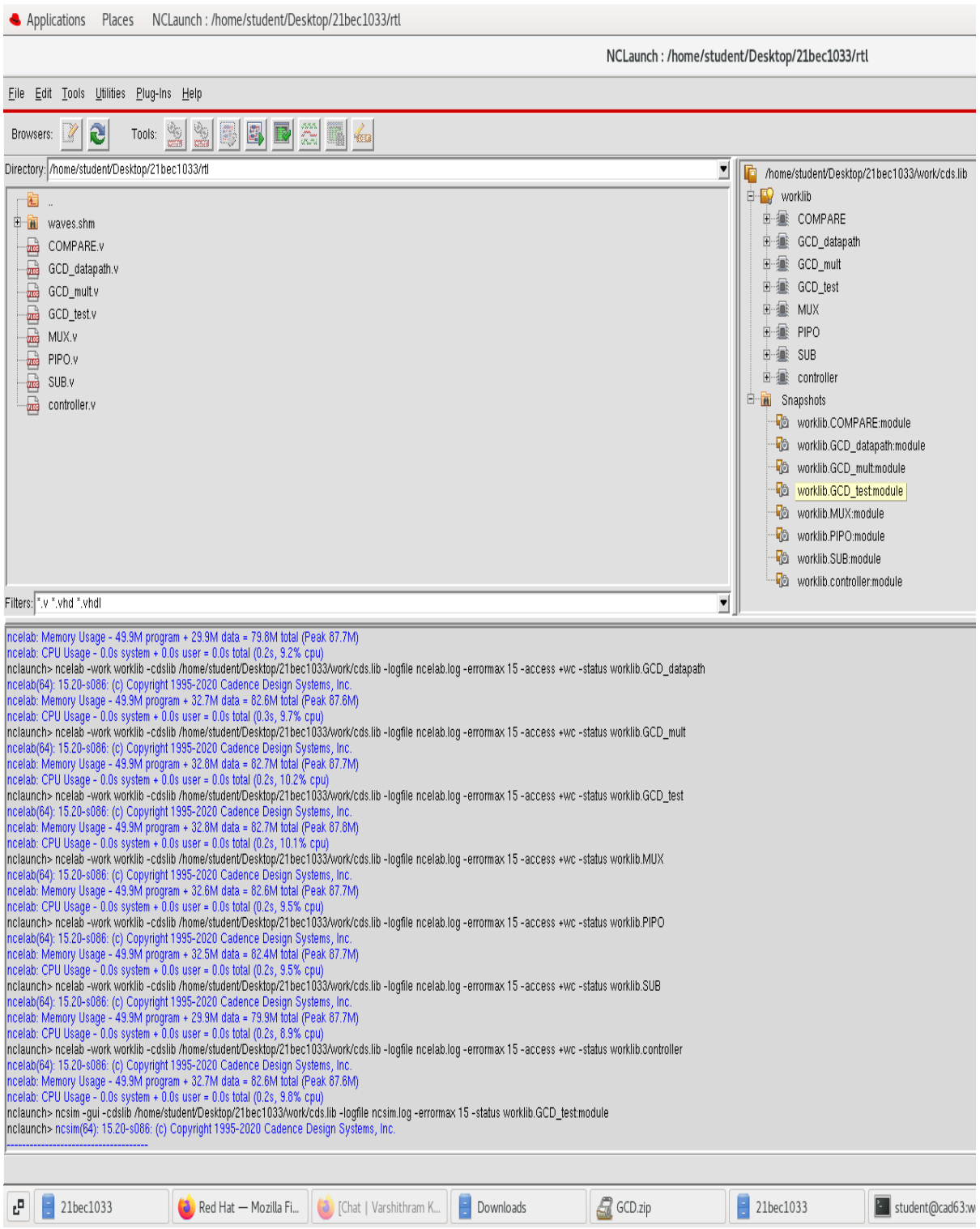
Synthesis effort: High

A total of 37 leaf instance count is present in the gate level netlist with total area of 91.656, total power of 3.69531e-06 W.

Result: Hence a multiplier by repeated addition using Datapath and Control path logic is synthesized and the gate level netlist with timing, area and power report has been generated.

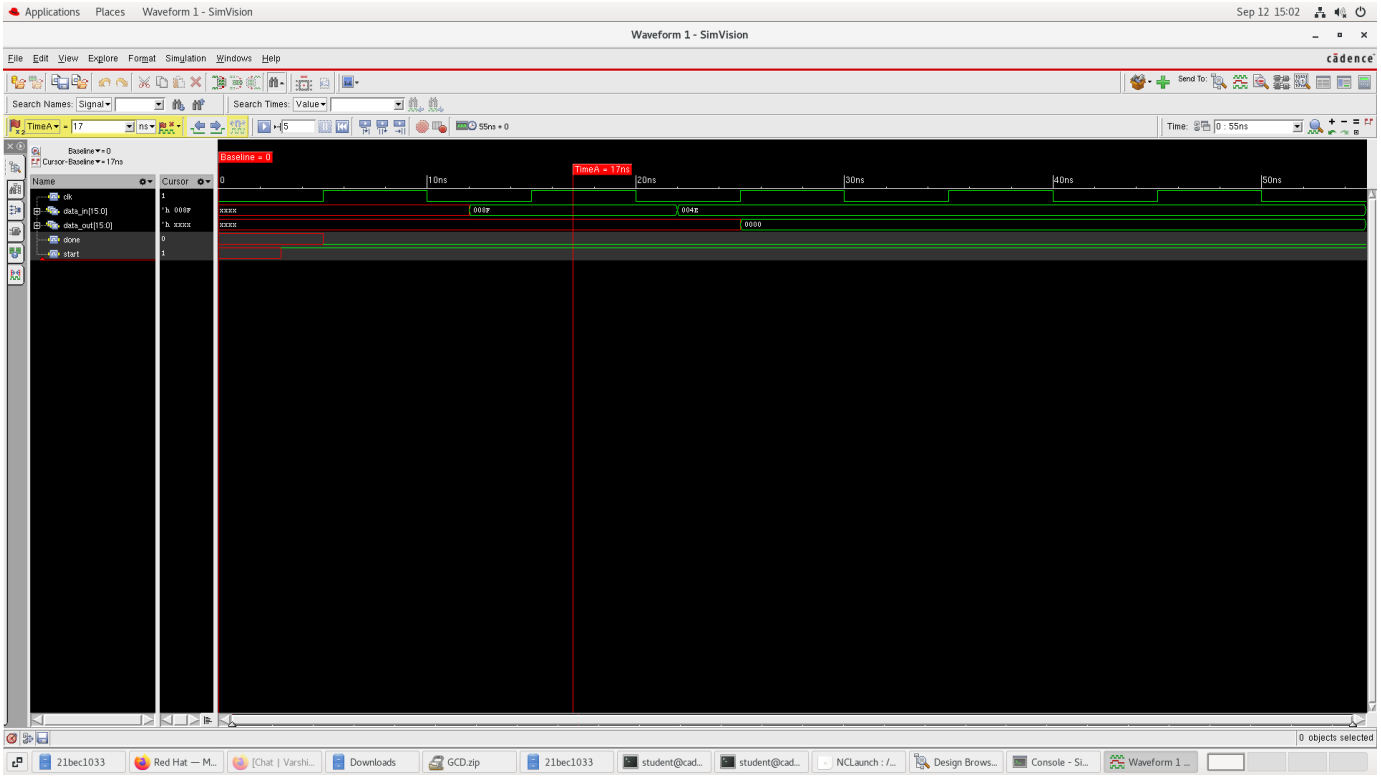
Lab4_2: Design and simulation of a GCD computation using Datapath and Control path logic

NCLaunch Compilation and Elaboration:



Lab4_2: Design and simulation of a GCD computation using Datapath and Control path logic

Observations/Simulation Waveforms:



Inference: In the above output waveform, the data_in is provided with 143 & 78 values and the data_out is obtained as 13 which is the GCD. Output “done” is equal to 1 whenever gcd result is obtained.

Result: Hence, a GCD computer is designed and the simulation and verification of behavior of GCD computer using data path and Control path logic is performed for the given stimuli in the test bench.

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

Aim: To synthesize a GCD computer using Datapath and Control path logic and to get the gate level netlist with timing, area, and power reports.

EDA Tools Used: Cadence Genus

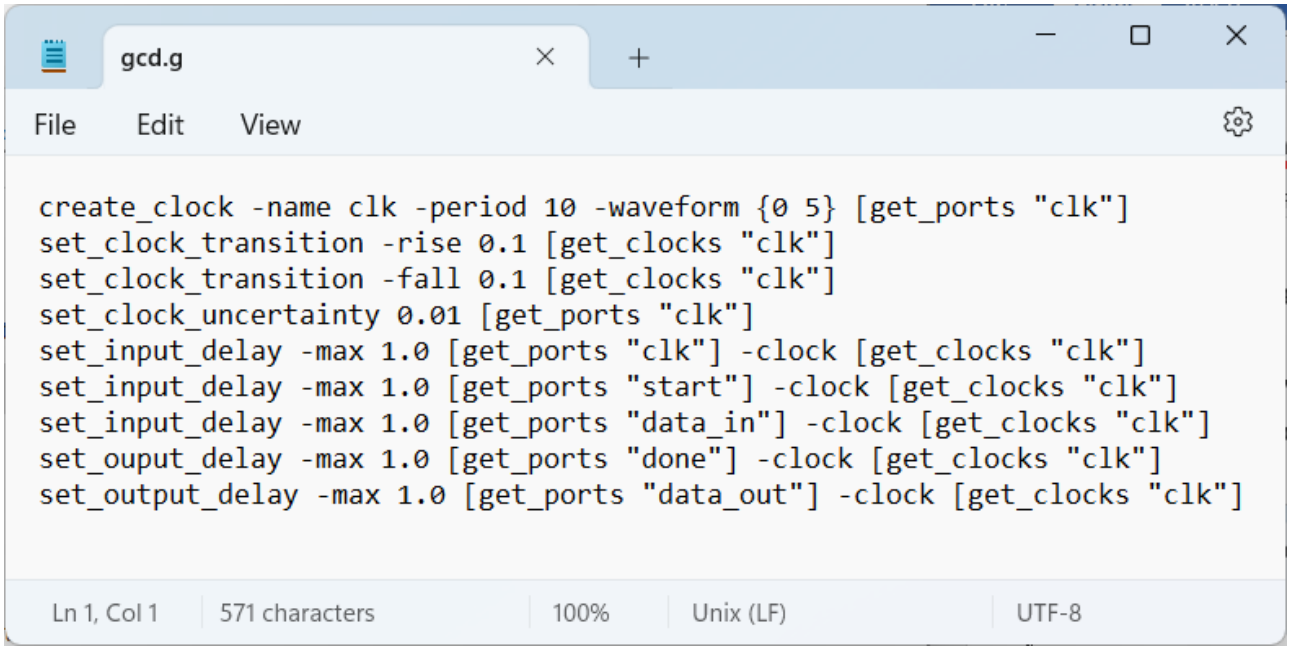
Description:

Datapath is the hardware that performs all the required operations. Control is the hardware that tells the datapath what to do, in terms of switching, operation selection, data movement between ALU components, etc. Simple datapath components include memory (stores the current instruction), PC or program counter (stores the address of current instruction), and ALU (executes current instruction). Control path is the Finite state machine designed by using Moore or Mealy models.

Procedure:

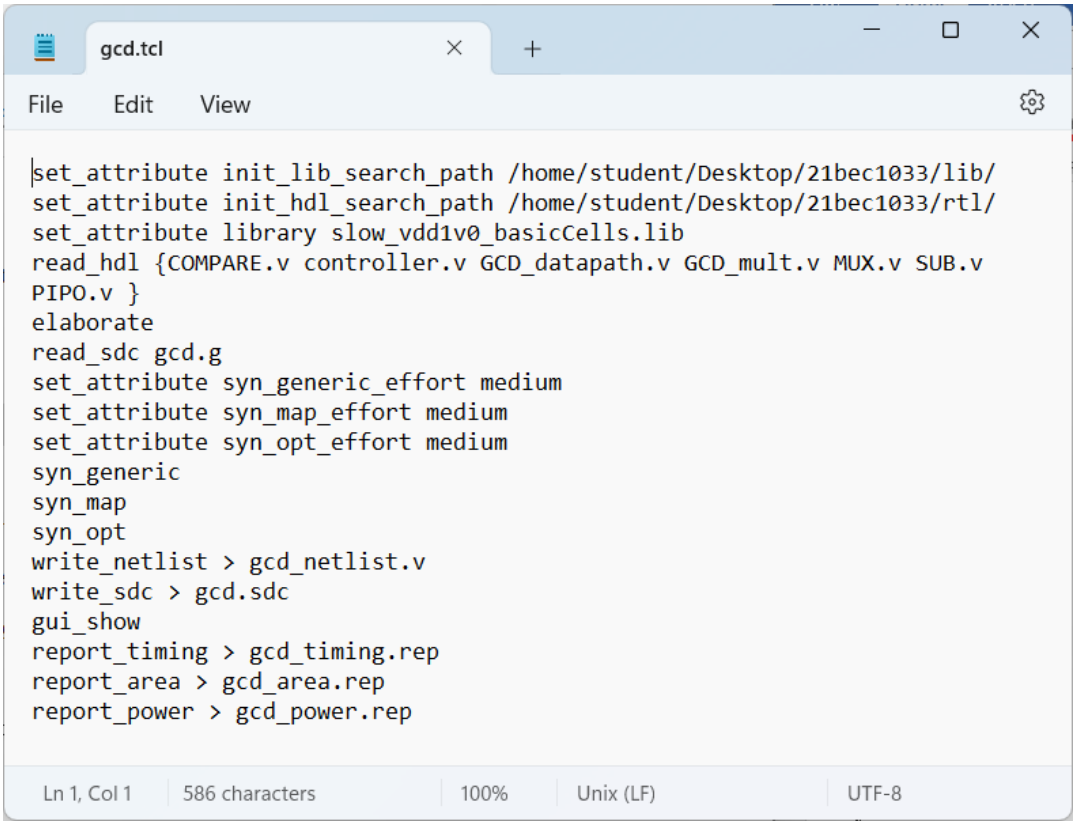
1. Copy the fast.lib and slow.lib files into the folder where (.v) files are located.

2. Create a Synopsys design constraint file (.g file) by entering the design constraints required for the synthesis.



LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

3. Create a (.tcl) file containing all the commands for performing the logic synthesis. Synthesis effort can be medium or high.



4. Invoke the C shell and launch the Genus tool by entering the below commands.
   'csh'
   'source /home/install/cshrc'
   'genus -legacy -ui -f gcd.tcl'

5. Execute the commands in the (.tcl file) by entering **source gcd.tcl** command in the command line window.

6. Check for the area, timing and power reports generated in the respective multiplier folder. Also check the gate level netlist generated in the Genus synthesis solution window.

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

Verilog Programs:

//Verilog Program of top-level module containing data path and control path

gcd.tcl | GCD_mult.v

File Edit View

```
module GCD_mult(data_out, done, data_in, clk, start);
output done;
output [15:0] data_out;
input clk, start;
input [15:0]data_in;
wire gt,lt,eq,ldA,ldB,sel1,sel2,sel_in;
GCD_datapath uut1(data_out,gt,lt,eq,ldA,ldB,sel1,sel2,sel_in,data_in,clk);
controller uut2(ldA, ldB, sel1, sel2, sel_in, done, clk, lt, gt, eq, start);
endmodule
```

Ln 1, Col 1 | 333 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of control path module

controller.v

File Edit View

```
input clk, lt, gt, eq, start;
output reg ldA, ldB, sel1, sel2, sel_in, done;
reg [2:0] state;
parameter S0=3'b000, S1=3'b001, S2=3'b010, S3=3'b011, S4=3'b100, S5=3'b101;

always @(posedge clk)
begin
        case (state)
                S0:     if (start) state <= S1;
                S1:     state <= S2;
                S2:     #2 if (eq) state <= S5;
                        else if (lt) state <= S3;
                        else if (gt) state <= S4;
                S3:     #2 if (eq) state <= S5;
                        else if (lt) state <= S3;
                        else if (gt) state <= S4;
                S4:     #2 if (eq) state <= S5;
                        else if (lt) state <= S3;
                        else if (gt) state <= S4;
                S5:     state <= S5;
                default: state <= S0;
        endcase
end


always @(state)
begin
        case (state)
                S0:     begin sel_in =1;ldA = 1; ldB = 0; done = 0; end
                S1:     begin sel_in = 1; ldA = 0; ldB = 1; end
                S2:     if (eq) done = 1;
                        else if (lt) begin
                                        sel1 = 1; sel2 = 0; sel_in = 0;
                                        #1 ldA = 0; ldB = 1;
                                end
                        else if(gt) begin
                                        sel1 = 0; sel2 = 1; sel_in = 0;
                                        #1 ldA = 1; ldB = 0;
                                end
                S3:     if (eq) done = 1;
                        else if (lt) begin
```

Ln 40, Col 8 | 1,585 characters

module controller (ldA, ldB, sel1,

File Edit View

```
begin
        case (state)
                S0:     begin sel_in =1;ldA = 1; ldB = 0; done = 0; end
                S1:     begin sel_in = 1; ldA = 0; ldB = 1; end
                S2:     if (eq) done = 1;
                        else if (lt) begin
                                        sel1 = 1; sel2 = 0; sel_in = 0;
                                        #1 ldA = 0; ldB = 1;
                                end
                        else if(gt) begin
                                        sel1 = 0; sel2 = 1; sel_in = 0;
                                        #1 ldA = 1; ldB = 0;
                                end
                S3:     if (eq) done = 1;
                        else if (lt) begin
                                        sel1 = 1; sel2 = 0; sel_in = 0;
                                        #1 ldA = 0; ldB = 1;
                                end
                        else if(gt) begin
                                        sel1 = 0; sel2 = 1; sel_in = 0;
                                        #1 ldA = 1; ldB = 0;
                                end
                S4:     if (eq) done = 1;
                        else if (lt) begin
                                        sel1 = 1; sel2 = 0; sel_in = 0;
                                        #1 ldA = 0; ldB = 1;
                                end
                        else if(gt) begin
                                        sel1 = 0; sel2 = 1; sel_in = 0;
                                        #1 ldA = 1; ldB = 0;
                                end
                S5:     begin
                                done = 1; sel1 = 0; sel2 = 0; ldA = 0;
                                ldB = 0;
                        end
                default: begin ldA = 0; ldB = 0; end
        endcase
end
endmodule
```

Ln 68, Col 1 | 1,585 characters | 100% | Windows (CRLF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

//Verilog Program of data path module

GCD_datapath.v

File Edit View

```
module GCD_datapath(data_out,gt,lt,eq,ldA,ldB,sel1,sel2,sel_in,data_in,clk);
input ldA, ldB, sel1,sel2,sel_in,clk;
input [15:0] data_in;
output gt,lt,eq;
output reg [15:0] data_out;
wire [15:0] Aout,Bout,X,Y,Bus,SubOut;

always @ (eq)
begin
if(eq == 1)
data_out <= data_in;
else
data_out <= 0;
end
PIPO A (Aout, Bus, ldA,clk);
PIPO B (Bout, Bus, ldB, clk);
MUX MUX_in1(X,Aout,Bout, sel1);
MUX MUX_in2(Y, Aout,Bout, sel2);
MUX MUX_load(Bus, SubOut,data_in,sel_in);
SUB SB(SubOut,X,Y);
COMPARE COMP(lt,gt,eq,Aout,Bout);
endmodule
```

Ln 1, Col 1 | 528 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of PIPO

PIPO.v

File Edit View

```
module PIPO(data_out,data_in,load,clk);
input [15:0] data_in;
input load,clk;
output reg [15:0] data_out;
always @ (posedge clk)
        if(load) data_out <= data_in;
endmodule
```

Ln 5, Col 23 | 170 characters | 100% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

//Verilog Program of COMPARE

PIPO.v | COMPARE.v

File Edit View

```
module COMPARE(lt,gt,eq,data1,data2);
input [15:0] data1,data2;
output lt,gt,eq;
assign lt = data1 < data2;
assign gt = data1 > data2;
assign eq = data1 == data2;
endmodule
```

Ln 1, Col 1 | 173 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of MUX

MUX.v

File Edit View

```
module MUX(out,in0,in1,sel);
input [15:0] in0, in1;
input sel;
output [15:0] out;
assign out = sel ? in1 : in0;
endmodule
```

Ln 1, Col 1 | 122 characters | 100% | Unix (LF) | UTF-8

//Verilog Program of SUB

MUX.v | SUB.v

File Edit View

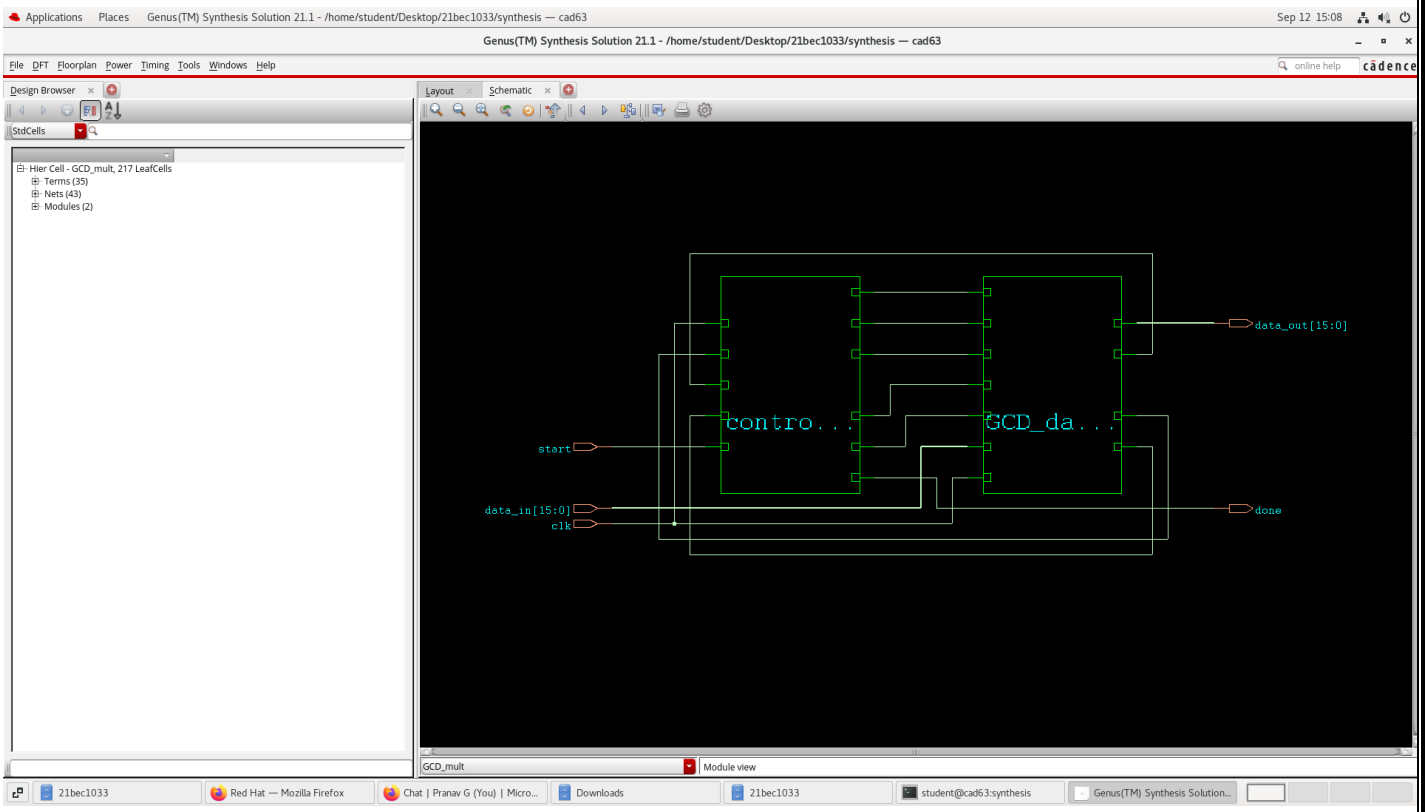
```
module SUB(out,in1,in2);
input [15:0] in1,in2;
output reg [15:0] out;
always @ (*)
        out = in1 - in2;
endmodule
```

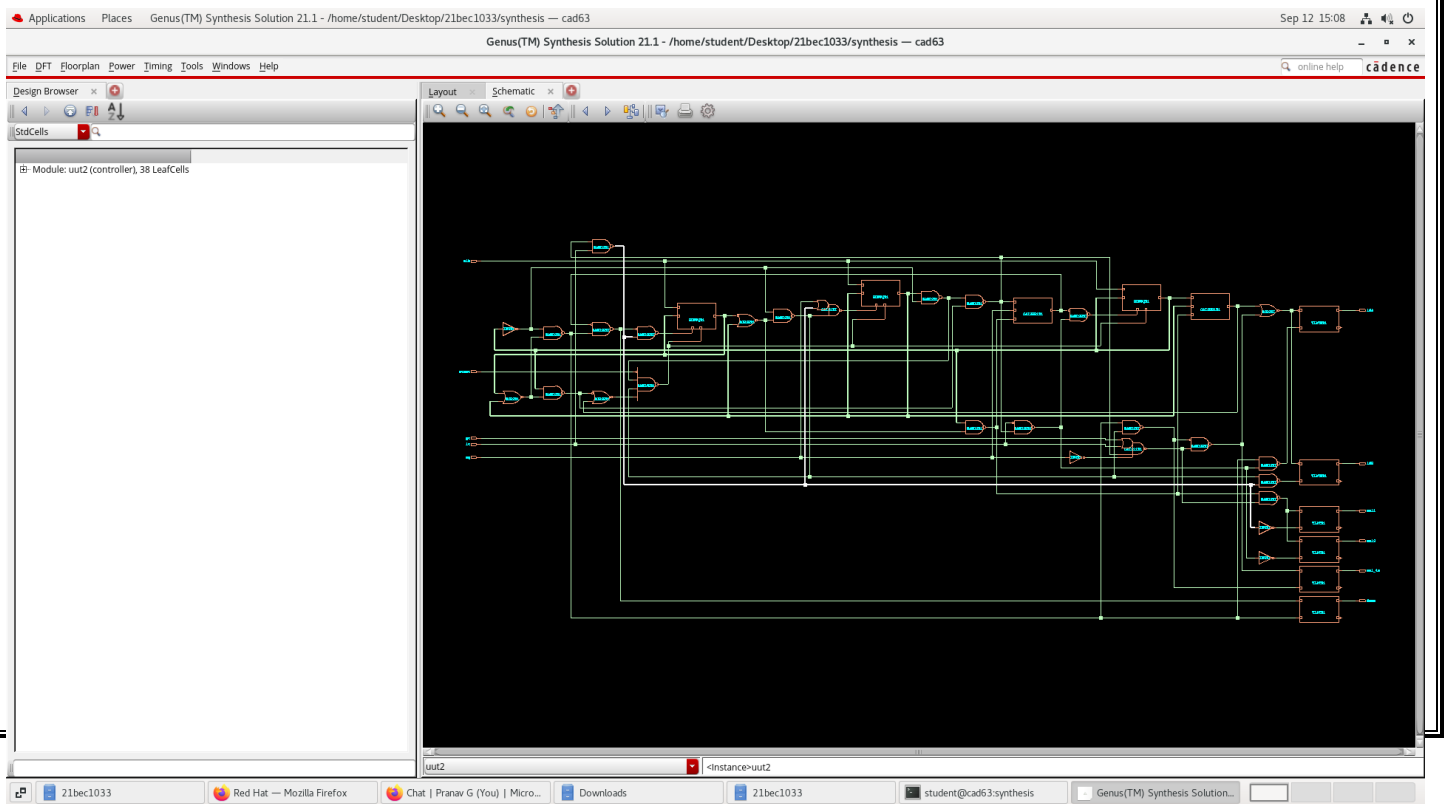
Ln 1, Col 1 | 111 characters | 100% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

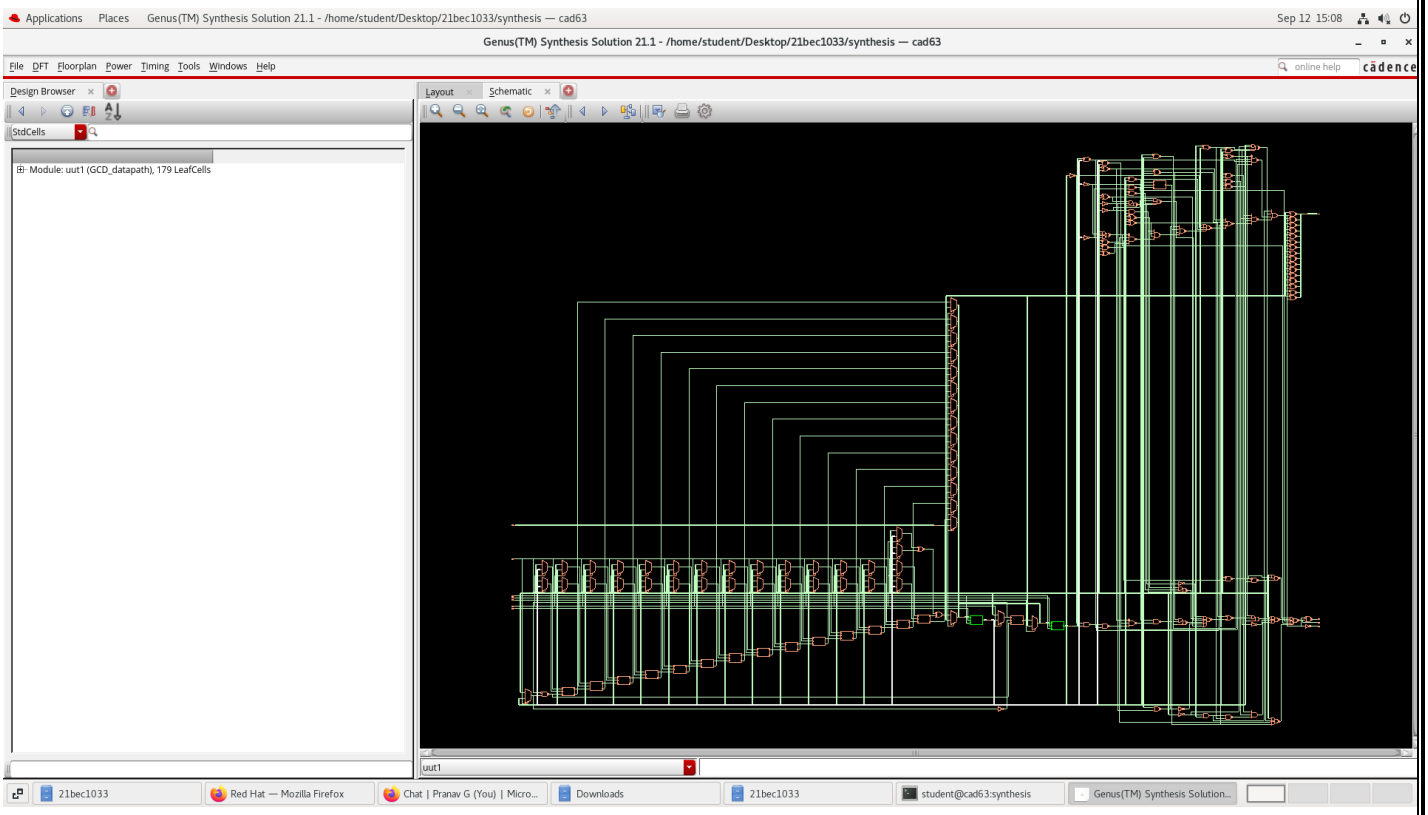
Gate level Netlist:

Synthesis effort: Medium

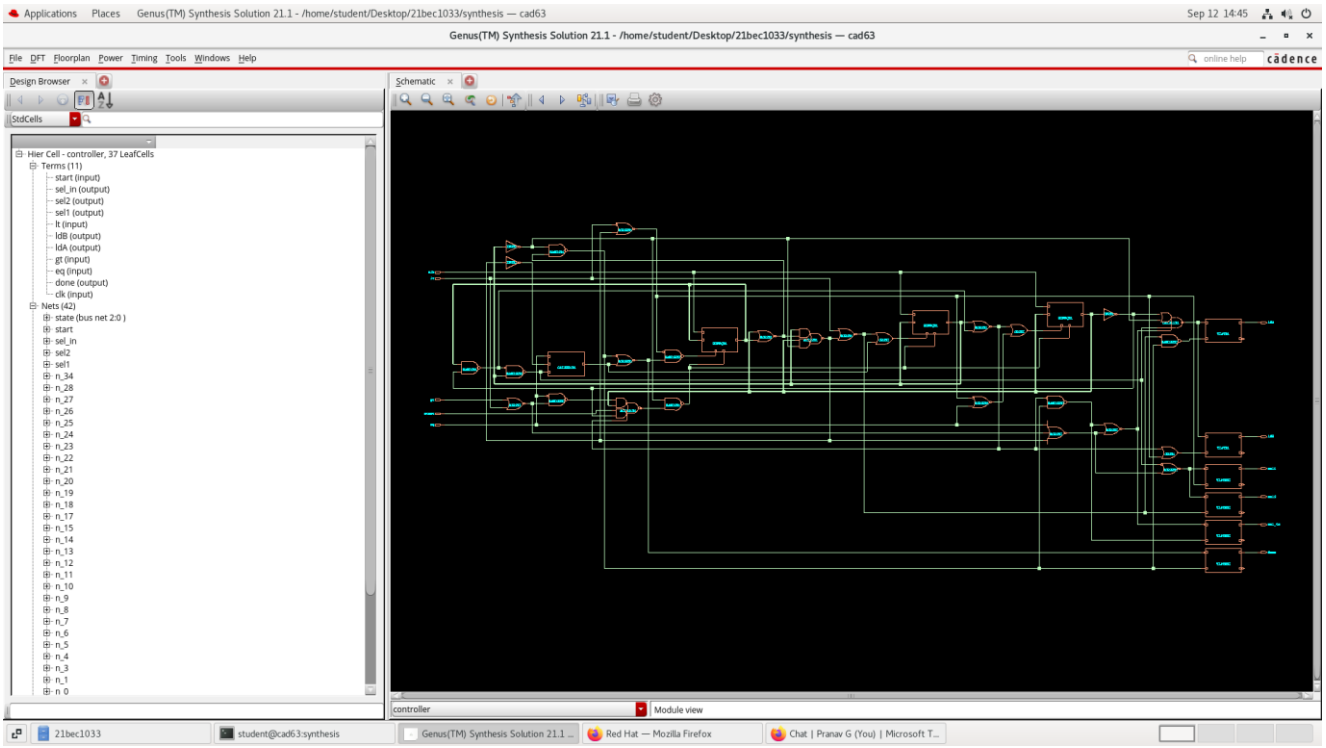




LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic



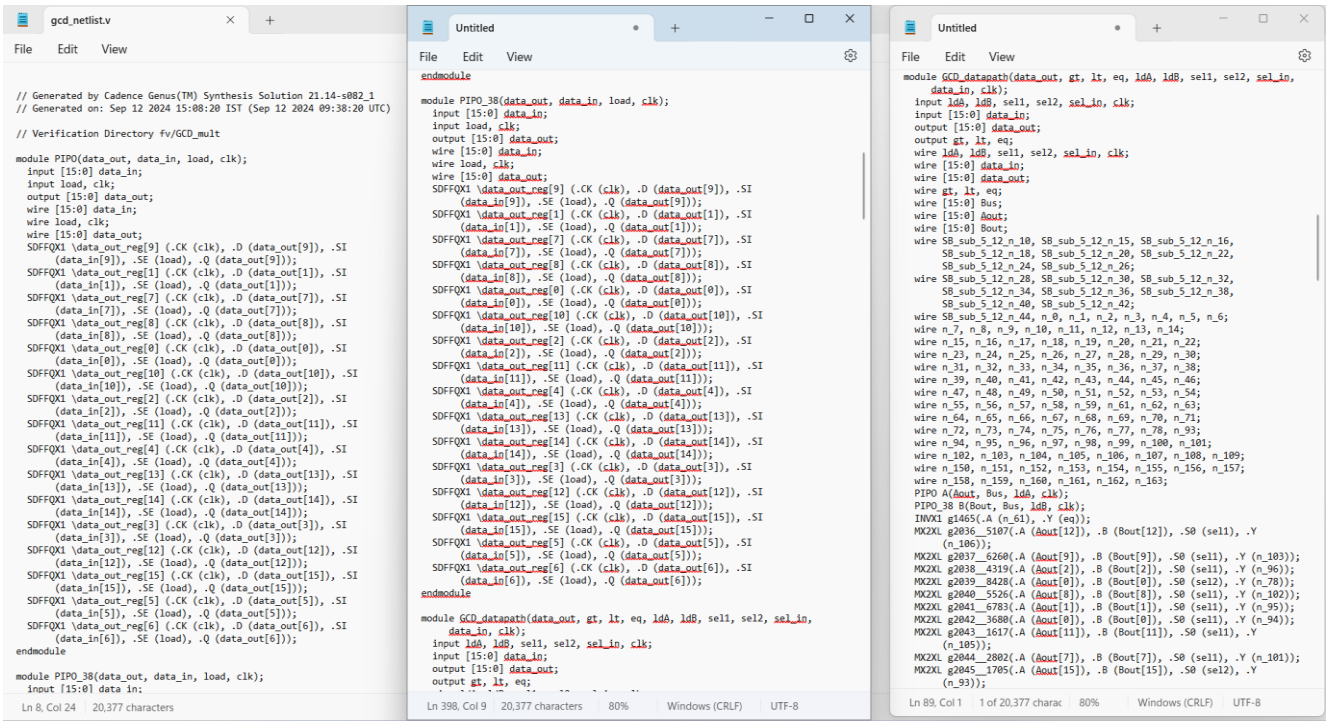
Synthesis effort : High

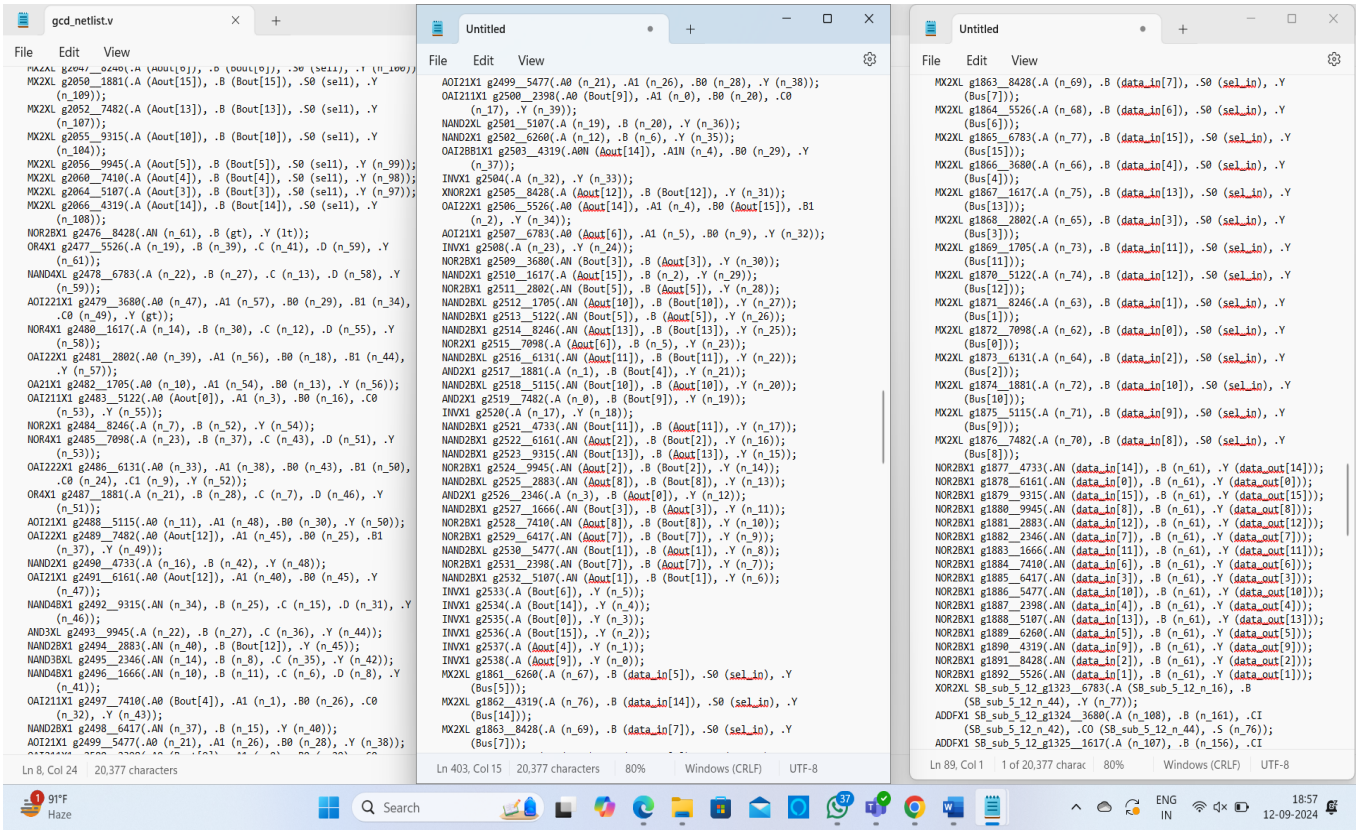


LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

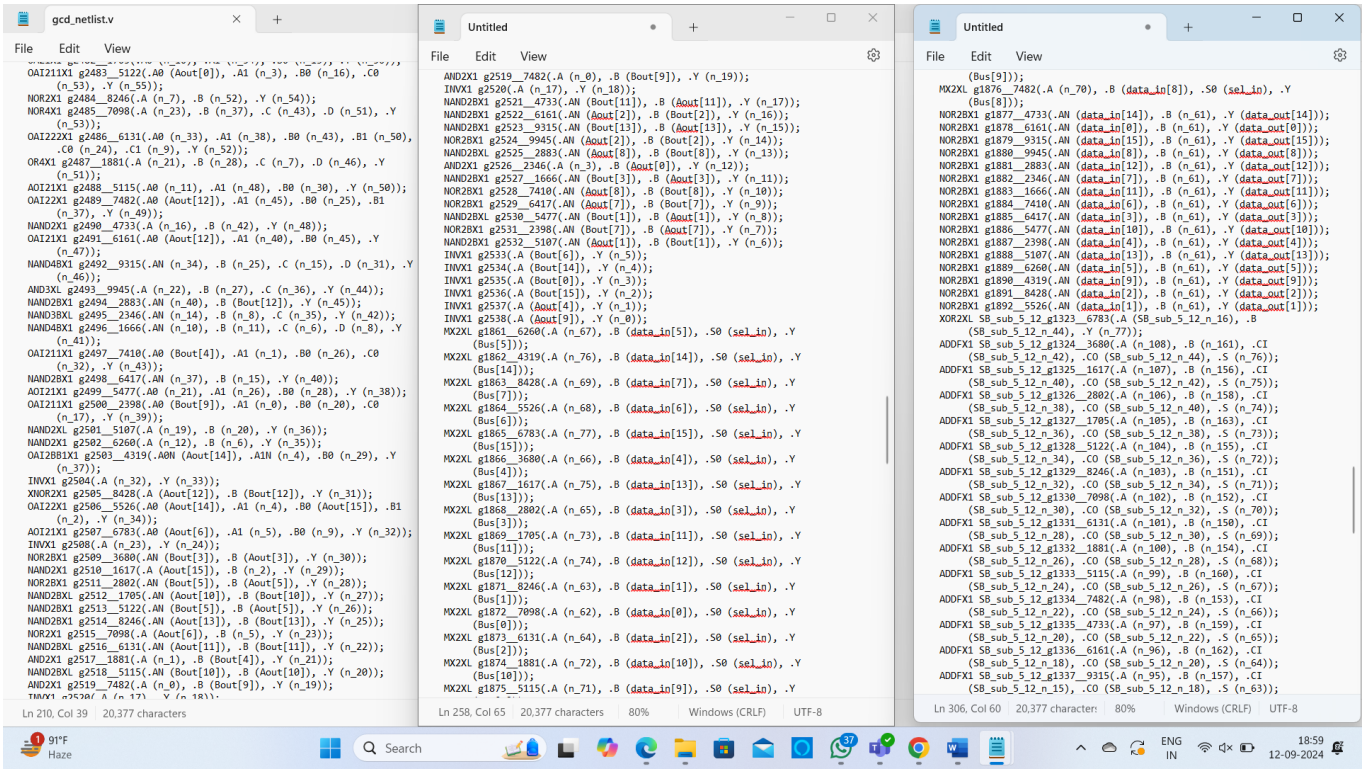
Observations:

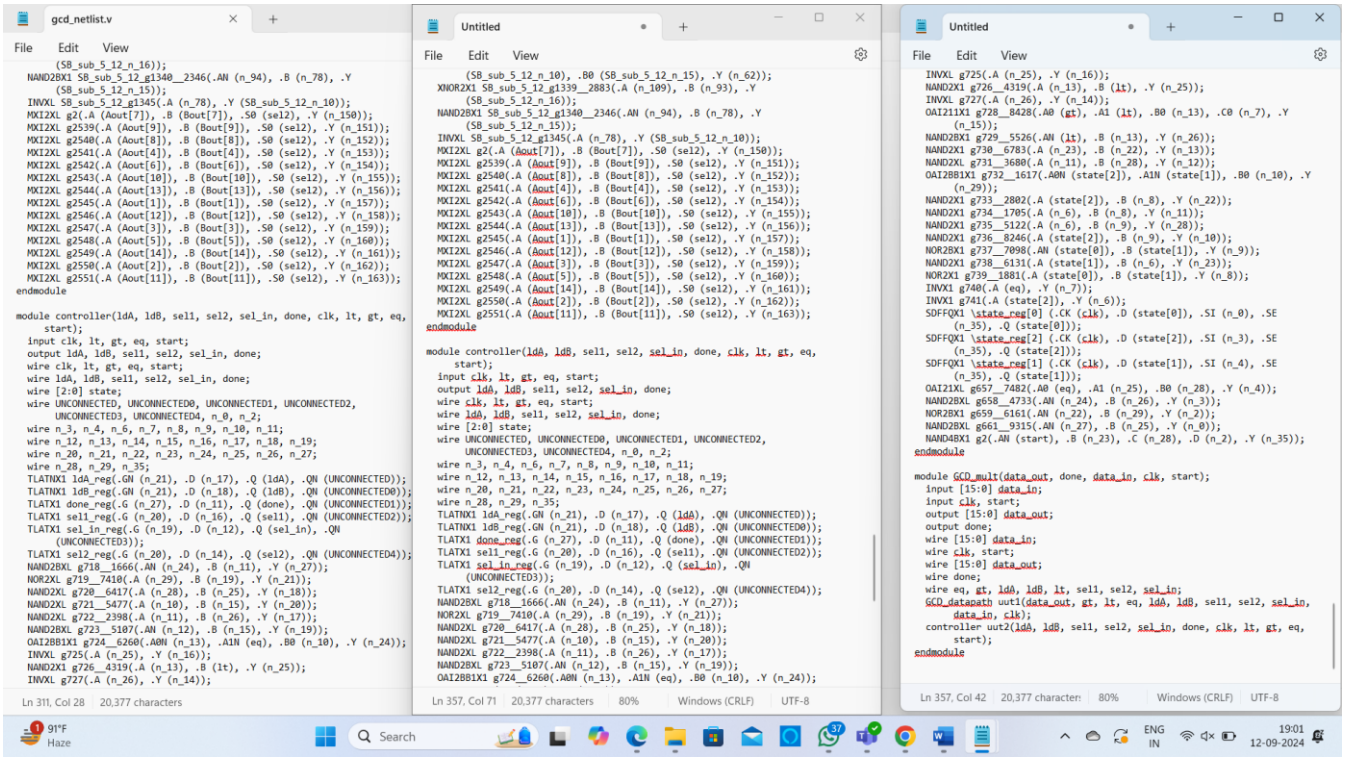
a) GCD computer_netlist_created file





LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic





LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

b) GCD computer_constraint_created file

gcd.sdc

File Edit View

```
# ####################################################################

#  Created by Genus(TM) Synthesis Solution 21.14-s082_1 on Thu Sep 12 15:03:30 IST 2024

# ####################################################################

set sdc_version 2.0

set_units -capacitance 1000fF
set_units -time 1000ps

# Set the current design
current_design GCD_mult

create_clock -name "clk" -period 10.0 -waveform {0.0 5.0} [get_ports clk]
set_clock_transition 0.1 [get_clocks clk]
set_clock_gating_check -setup 0.0
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports start]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[15]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[14]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[13]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[12]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[11]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[10]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[9]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[8]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[7]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[6]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[5]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[4]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[3]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[2]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[1]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[0]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[15]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[14]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[13]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[12]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[11]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[10]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[9]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[8]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[7]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[6]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[5]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[4]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[3]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[2]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[1]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_out[0]}]
set_wire_load_mode "enclosed"
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]
```

Ln 24, Col 83 | 3,480 characters | 60% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

c) GCD computer_timing_created file

Synthesis effort: Medium

gcd_netlist.v

File Edit View

```
#====================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:08:21 pm
  Module:                 GCD_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
#====================================================

            Pin                      Type       Fanout Load Slew Delay Arrival
                                                      (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------
(clock clk)                          launch                                 0 R
uut1
  A
    data_out_reg[0]/CK                                          100    +0       0 R
    data_out_reg[0]/Q                SDFFQX1          5  1.2   37  +232     232 R
  A/data_out[0]
  g2042__3680/A                                                      +0     232
  g2042__3680/Y                      MX2XL            2  0.4   29  +152     384 R
  SB_sub_5_12_g1340__2346/AN                                         +0     384
  SB_sub_5_12_g1340__2346/Y          NAND2BX1         2  1.0   40   +78     461 R
  SB_sub_5_12_g1337__9315/CI                                         +0     461
  SB_sub_5_12_g1337__9315/CO         ADDFX1           1  0.6   39  +186     648 R
  SB_sub_5_12_g1336__6161/CI                                         +0     648
  SB_sub_5_12_g1336__6161/CO         ADDFX1           1  0.6   39  +186     834 R
  SB_sub_5_12_g1335__4733/CI                                         +0     834
  SB_sub_5_12_g1335__4733/CO         ADDFX1           1  0.6   39  +186    1020 R
  SB_sub_5_12_g1334__7482/CI                                         +0    1020
  SB_sub_5_12_g1334__7482/CO         ADDFX1           1  0.6   39  +186    1206 R
  SB_sub_5_12_g1333__5115/CI                                         +0    1206
  SB_sub_5_12_g1333__5115/CO         ADDFX1           1  0.6   39  +186    1392 R
  SB_sub_5_12_g1332__1881/CI                                         +0    1392
  SB_sub_5_12_g1332__1881/CO         ADDFX1           1  0.6   39  +186    1578 R
  SB_sub_5_12_g1331__6131/CI                                         +0    1578
  SB_sub_5_12_g1331__6131/CO         ADDFX1           1  0.6   39  +186    1764 R
  SB_sub_5_12_g1330__7098/CI                                         +0    1764
  SB_sub_5_12_g1330__7098/CO         ADDFX1           1  0.6   39  +186    1950 R
  SB_sub_5_12_g1329__8246/CI                                         +0    1950
  SB_sub_5_12_g1329__8246/CO         ADDFX1           1  0.6   39  +186    2136 R
  SB_sub_5_12_g1328__5122/CI                                         +0    2136
  SB_sub_5_12_g1328__5122/CO         ADDFX1           1  0.6   39  +186    2322 R
  SB_sub_5_12_g1327__1705/CI                                         +0    2322
  SB_sub_5_12_g1327__1705/CO         ADDFX1           1  0.6   39  +186    2508 R
  SB_sub_5_12_g1326__2802/CI                                         +0    2508
  SB_sub_5_12_g1326__2802/CO         ADDFX1           1  0.6   39  +186    2694 R
  SB_sub_5_12_g1325__1617/CI                                         +0    2694
  SB_sub_5_12_g1325__1617/CO         ADDFX1           1  0.6   39  +186    2880 R
  SB_sub_5_12_g1324__3680/CI                                         +0    2880
  SB_sub_5_12_g1324__3680/CO         ADDFX1           1  0.4   35  +184    3063 R
  SB_sub_5_12_g1323__6783/B                                          +0    3063
  SB_sub_5_12_g1323__6783/Y          XOR2XL           1  0.2   22  +121    3184 R
  g1865__6783/A                                                      +0    3184
  g1865__6783/Y                      MX2XL            2  0.4   29  +144    3328 R
  B/data_in[15]
    data_out_reg[15]/SI       <<<    SDFFQX1                            +0    3328
    data_out_reg[15]/CK              setup                         100  +204    3532 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                          capture                              10000 R
                                     uncertainty                    -10   9990 R
-------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    6458ps
Start-point  : uut1/A/data_out_reg[0]/CK
End-point    : uut1/B/data_out_reg[15]/SI
```

Ln 1, Col 1 | 4,532 characters | 60% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

Synthesis effort: High

gcd_timing.rep

File Edit View

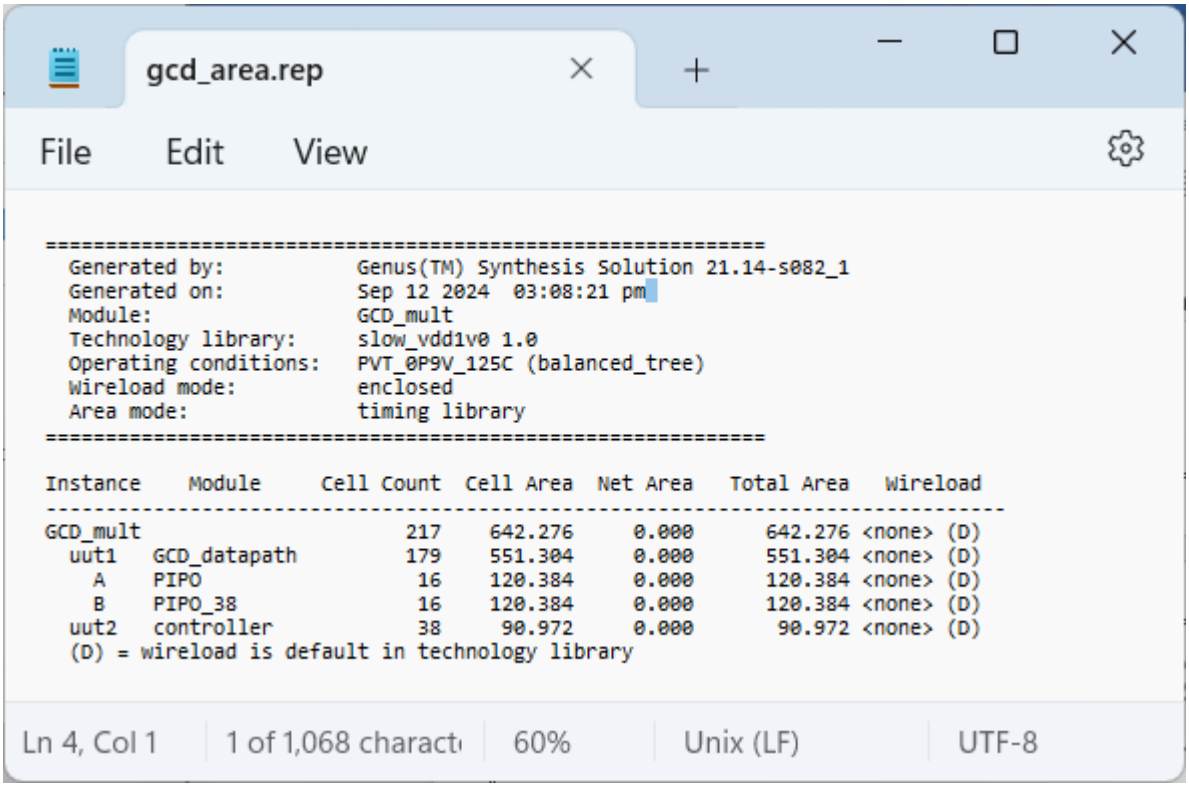
```
#====================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:03:31 pm
  Module:                 GCD_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================

             Pin                      Type       Fanout Load Slew Delay Arrival
                                                       (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------
(clock clk)                         launch                                0 R
uut1_A_data_out_reg[0]/CK                                  100    +0       0 R
uut1_A_data_out_reg[0]/Q            SDFFQX1          5  1.0   32  +230     230 R
g3083__2398/A                                                  +0     230
g3083__2398/Y                       MX2XL            2  0.4   29  +150     379 R
uut1_SB_sub_5_12_g448__1881/AN                                 +0     379
uut1_SB_sub_5_12_g448__1881/Y       NAND2BX1         2  1.0   40   +77     457 R
uut1_SB_sub_5_12_g445__8246/CI                                 +0     457
uut1_SB_sub_5_12_g445__8246/CO      ADDFX1           1  0.6   39  +186     643 R
uut1_SB_sub_5_12_g444__5122/CI                                 +0     643
uut1_SB_sub_5_12_g444__5122/CO      ADDFX1           1  0.6   39  +186     829 R
uut1_SB_sub_5_12_g443__1705/CI                                 +0     829
uut1_SB_sub_5_12_g443__1705/CO      ADDFX1           1  0.6   39  +186    1015 R
uut1_SB_sub_5_12_g442__2802/CI                                 +0    1015
uut1_SB_sub_5_12_g442__2802/CO      ADDFX1           1  0.6   39  +186    1201 R
uut1_SB_sub_5_12_g441__1617/CI                                 +0    1201
uut1_SB_sub_5_12_g441__1617/CO      ADDFX1           1  0.6   39  +186    1387 R
uut1_SB_sub_5_12_g440__3680/CI                                 +0    1387
uut1_SB_sub_5_12_g440__3680/CO      ADDFX1           1  0.6   39  +186    1573 R
uut1_SB_sub_5_12_g439__6783/CI                                 +0    1573
uut1_SB_sub_5_12_g439__6783/CO      ADDFX1           1  0.6   39  +186    1759 R
uut1_SB_sub_5_12_g438__5526/CI                                 +0    1759
uut1_SB_sub_5_12_g438__5526/CO      ADDFX1           1  0.6   39  +186    1945 R
uut1_SB_sub_5_12_g437__8428/CI                                 +0    1945
uut1_SB_sub_5_12_g437__8428/CO      ADDFX1           1  0.6   39  +186    2131 R
uut1_SB_sub_5_12_g436__4319/CI                                 +0    2131
uut1_SB_sub_5_12_g436__4319/CO      ADDFX1           1  0.6   39  +186    2317 R
uut1_SB_sub_5_12_g435__6260/CI                                 +0    2317
uut1_SB_sub_5_12_g435__6260/CO      ADDFX1           1  0.6   39  +186    2503 R
uut1_SB_sub_5_12_g434__5107/CI                                 +0    2503
uut1_SB_sub_5_12_g434__5107/CO      ADDFX1           1  0.6   39  +186    2689 R
uut1_SB_sub_5_12_g433__2398/CI                                 +0    2689
uut1_SB_sub_5_12_g433__2398/CO      ADDFX1           1  0.6   39  +186    2875 R
uut1_SB_sub_5_12_g432__5477/CI                                 +0    2875
uut1_SB_sub_5_12_g432__5477/CO      ADDFX1           1  0.4   35  +184    3059 R
uut1_SB_sub_5_12_g431__6417/B                                  +0    3059
uut1_SB_sub_5_12_g431__6417/Y       XOR2XL           1  0.2   22  +121    3180 R
g1599__4733/A                                                  +0    3180
g1599__4733/Y                       MX2XL            2  0.4   29  +144    3324 R
uut1_A_data_out_reg[15]/SI     <<<  SDFFQX1                      +0    3324
uut1_A_data_out_reg[15]/CK          setup                     100  +204    3527 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                         capture                            10000 R
                                    uncertainty                 -10    9990 R
-------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    6463ps
Start-point  : uut1_A_data_out_reg[0]/CK
End-point    : uut1_A_data_out_reg[15]/SI
```

Ln 1, Col 1 | 4,585 characters | 60% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

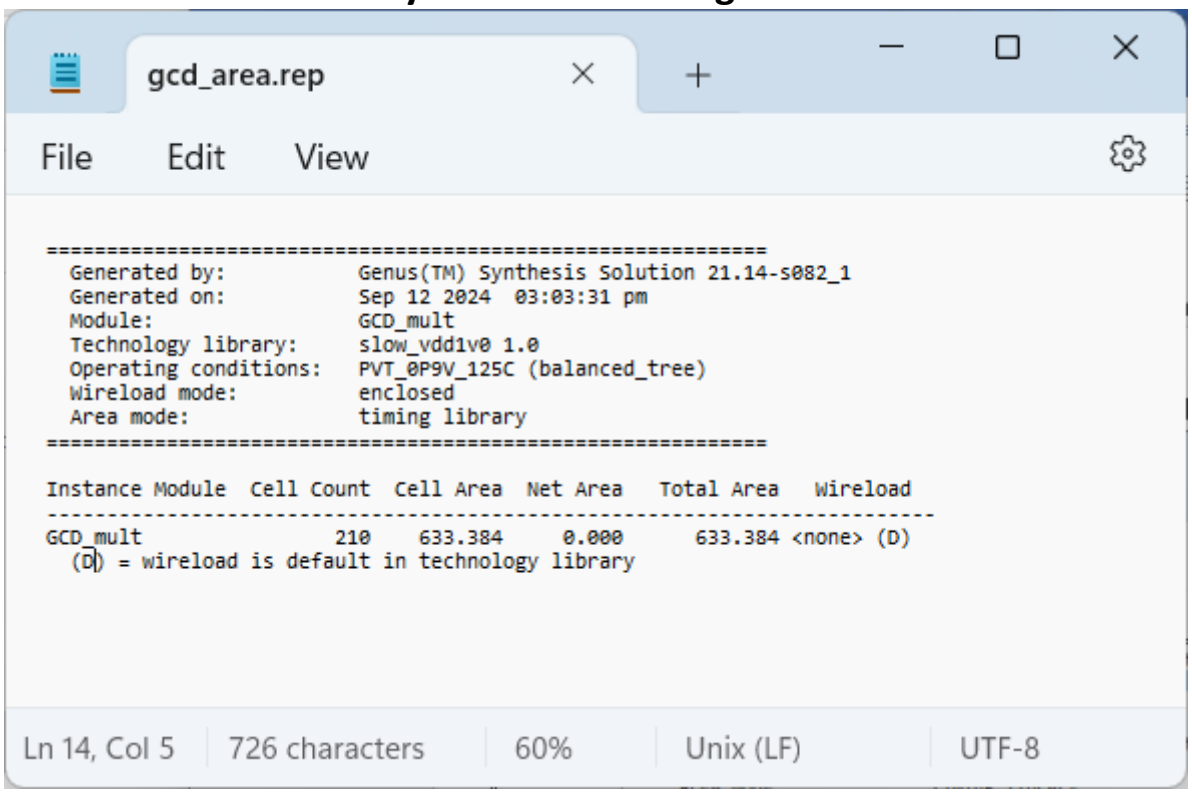
d) GCD computer_area_created file

Synthesis effort: Medium




```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:08:21 pm
  Module:                 GCD_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance    Module     Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------
GCD_mult                     217    642.276     0.000      642.276 <none> (D)
  uut1   GCD_datapath        179    551.304     0.000      551.304 <none> (D)
    A    PIPO                16    120.384     0.000      120.384 <none> (D)
    B    PIPO_38             16    120.384     0.000      120.384 <none> (D)
  uut2   controller           38     90.972     0.000       90.972 <none> (D)
  (D) = wireload is default in technology library
```

Synthesis effort: High




```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:03:31 pm
  Module:                 GCD_mult
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------
GCD_mult               210    633.384     0.000      633.384 <none> (D)
  (D) = wireload is default in technology library
```

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

e) GCD computer_power_created file

Synthesis effort: Medium

gcd_power.rep

File Edit View

```
Instance: /GCD_mult
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ---------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     4.14102e-09  1.44255e-05  9.77810e-07  1.54074e-05  53.95%
        latch     6.80845e-10  5.96203e-07  5.03486e-07  1.10037e-06   3.85%
        logic     6.43104e-09  8.80464e-06  2.67392e-06  1.14850e-05  40.21%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  5.67000e-07  5.67000e-07   1.99%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     1.12529e-08  2.38263e-05  4.72222e-06  2.85598e-05 100.00%
  Percentage           0.04%       83.43%       16.53%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,202 characters | 60% | Unix (LF) | UTF-8

Synthesis effort: High

gcd_power.rep

File Edit View

```
Instance: /GCD_mult
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total    Row%
  ---------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     4.13550e-09  1.35555e-05  6.97469e-07  1.42571e-05  53.58%
        latch     6.81032e-10  7.98210e-07  5.72306e-07  1.37120e-06   5.15%
        logic     6.91522e-09  7.99946e-06  2.40811e-06  1.04145e-05  39.14%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  5.67000e-07  5.67000e-07   2.13%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     1.17318e-08  2.23532e-05  4.24488e-06  2.66098e-05 100.00%
  Percentage           0.04%       84.00%       15.95%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,202 characters | 80% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a GCD computation using Datapath and Control path logic

Inference:

Synthesis effort: Medium

A total of 217 leaf instance count is present in the gate level netlist with total area of 642.276, total power of 2.66098e-05 W.

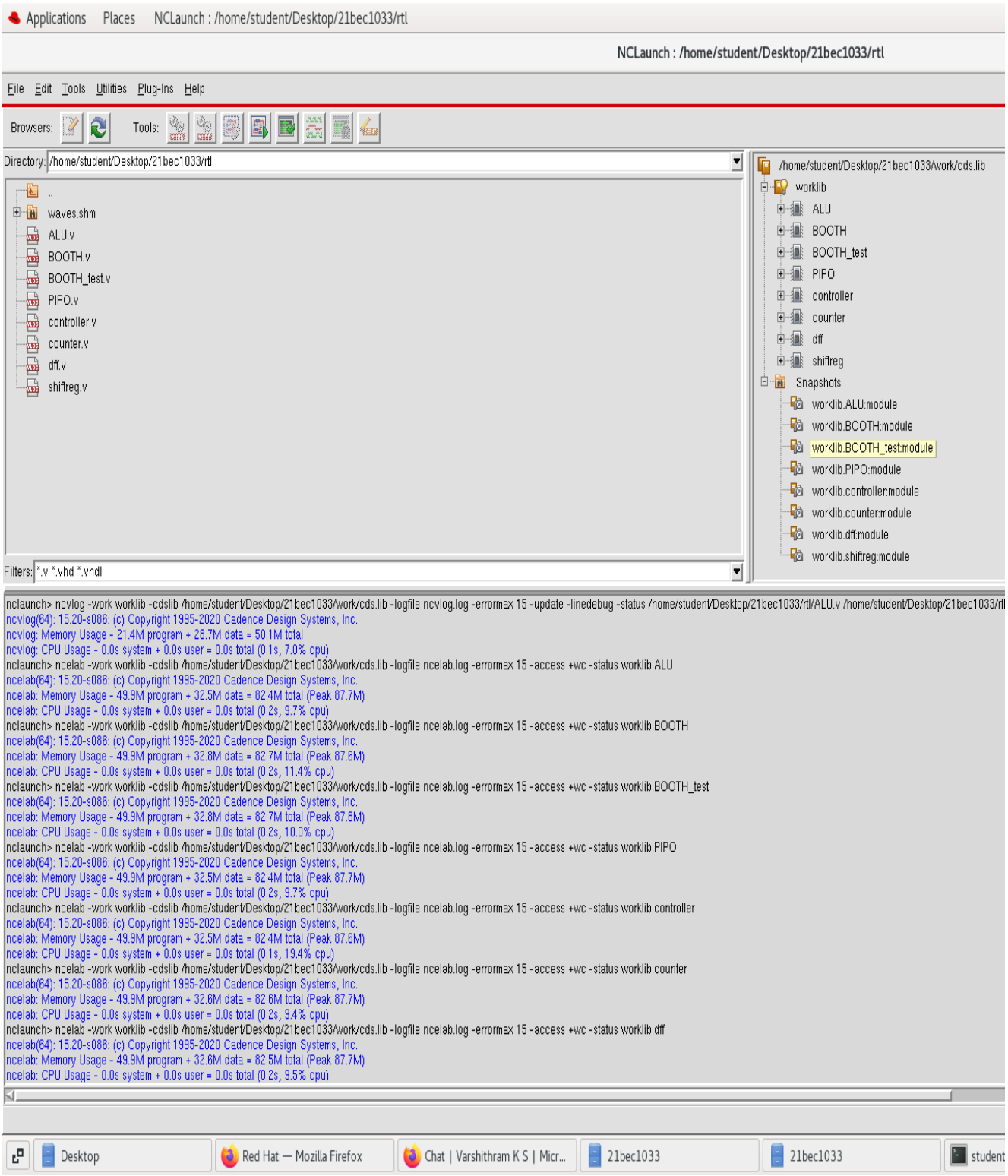
Synthesis effort: High

A total of 210 leaf instance count is present in the gate level netlist with total area of 633.384, total power of 2.85598e-05W.

Result: Hence a gcd computer using Datapath and Control path logic is synthesized and the gate level netlist with timing, area and power report has been generated.

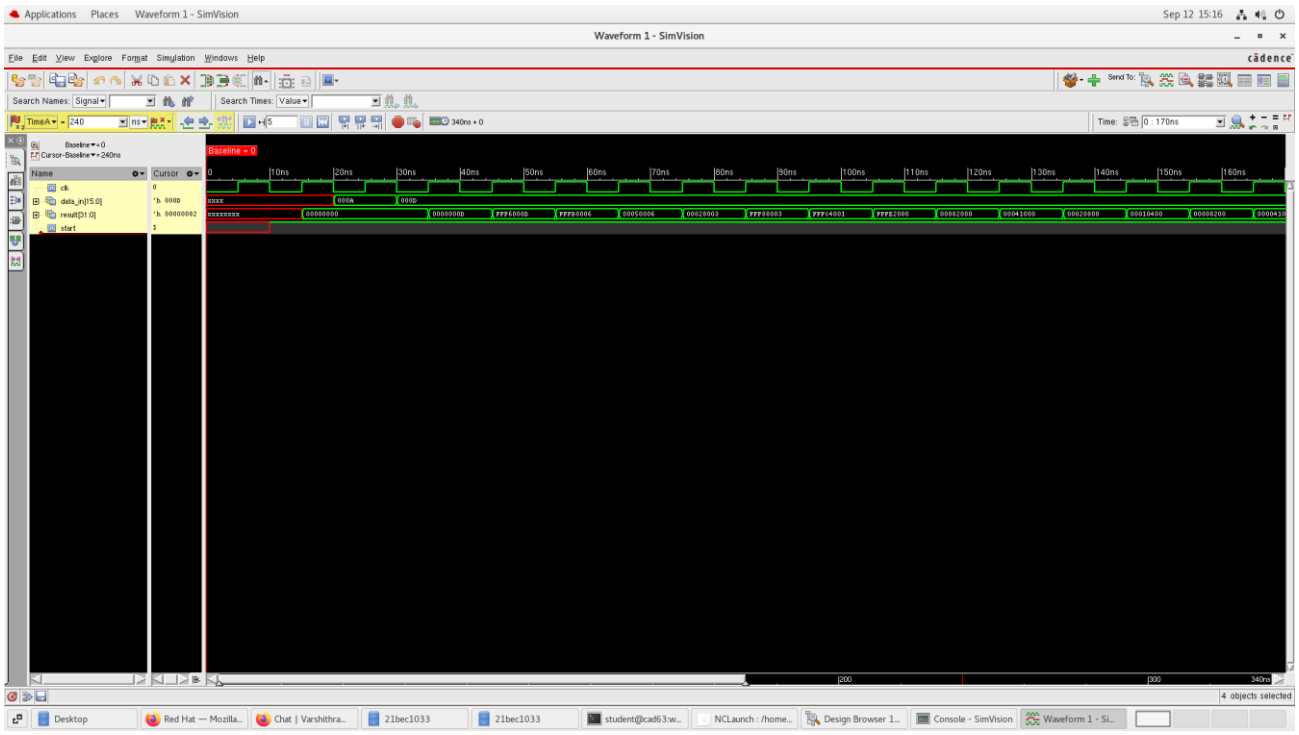
Lab4_3: Design and simulation of a Booth Multiplier using Datapath and Control path logic

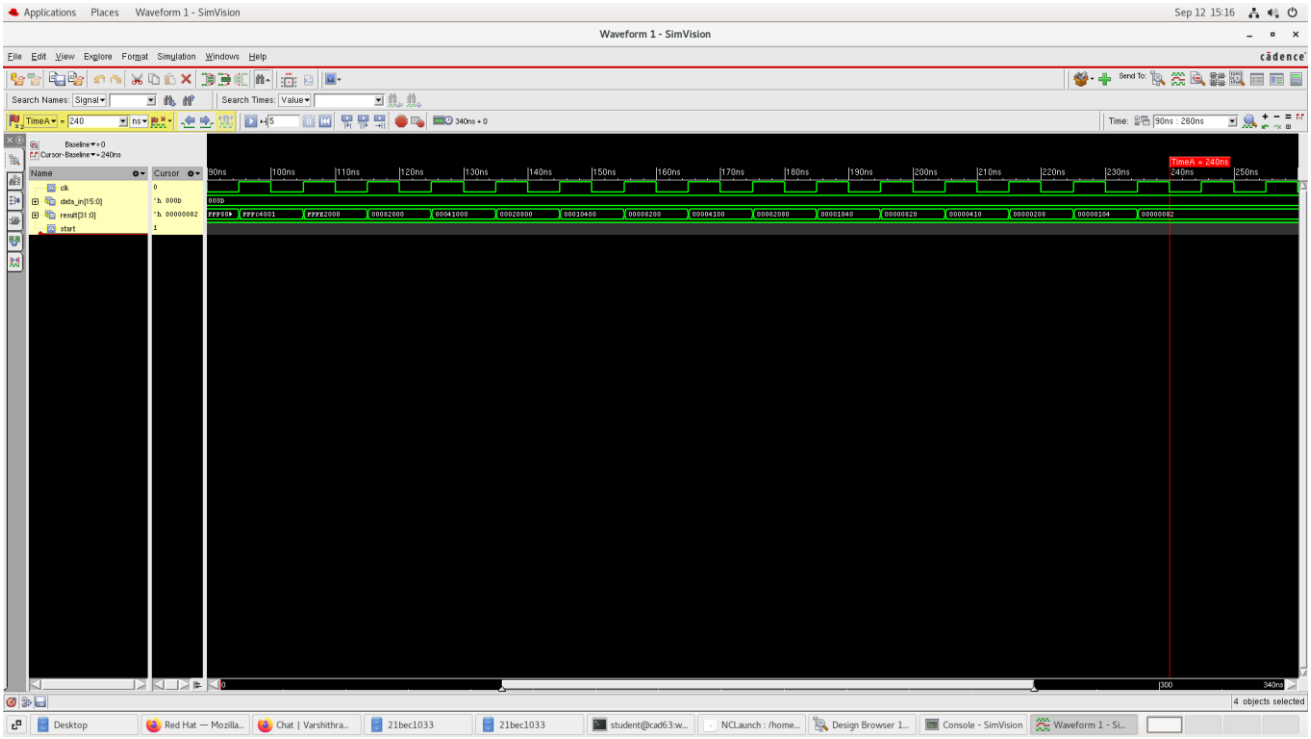
NCLaunch Compilation and Elaboration:



Lab4_3: Design and simulation of a Booth Multiplier using Datapath and Control path logic

Observations/Simulation Waveforms:





Lab4_3: Design and simulation of a Booth Multiplier using Datapath and Control path logic

Inference: In the above output waveform, the data_in is provided with 13 & 10 values and the data_out is obtained as 130 which is the multiplied value. Output “done” is equal to 1 whenever booth multiplier result is obtained.

Result: Hence, a Booth multiplier is designed and the simulation and verification of behavior of booth multiplier using data path and Control path logic is performed for the given stimuli in the test bench.

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

Aim: To synthesize a booth multiplier using Datapath and Control path logic and to get the gate level netlist with timing, area, and power reports.

EDA Tools Used: Cadence Genus

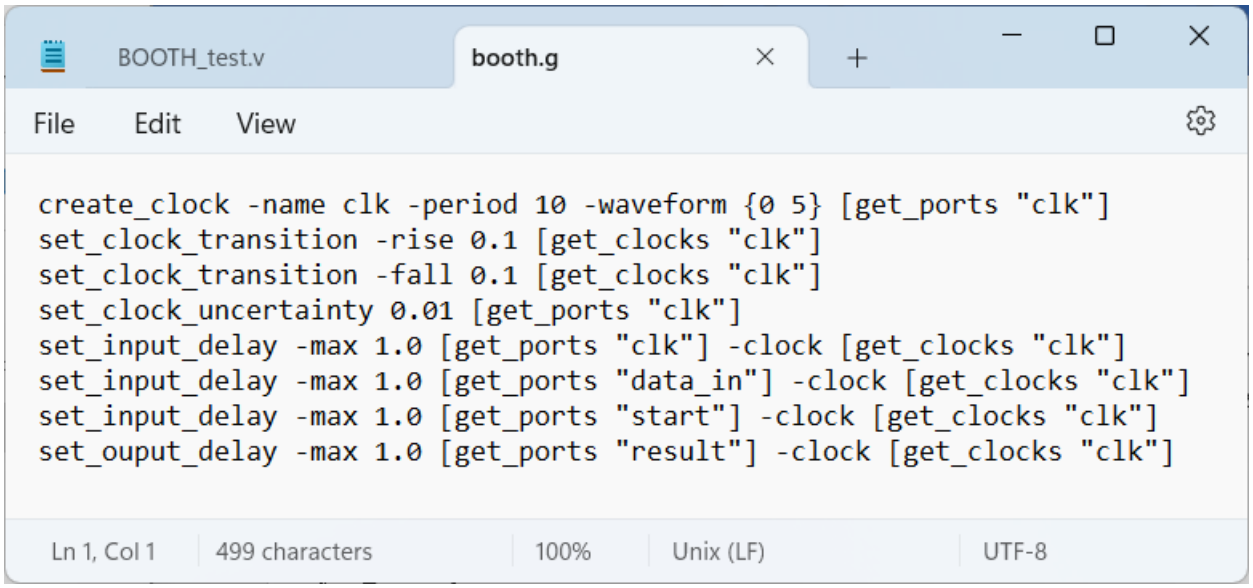
Description:

Datapath is the hardware that performs all the required operations. Control is the hardware that tells the datapath what to do, in terms of switching, operation selection, data movement between ALU components, etc. Simple datapath components include memory (stores the current instruction), PC or program counter (stores the address of current instruction), and ALU (executes current instruction). Control path is the Finite state machine designed by using Moore or Mealy models.

Procedure:

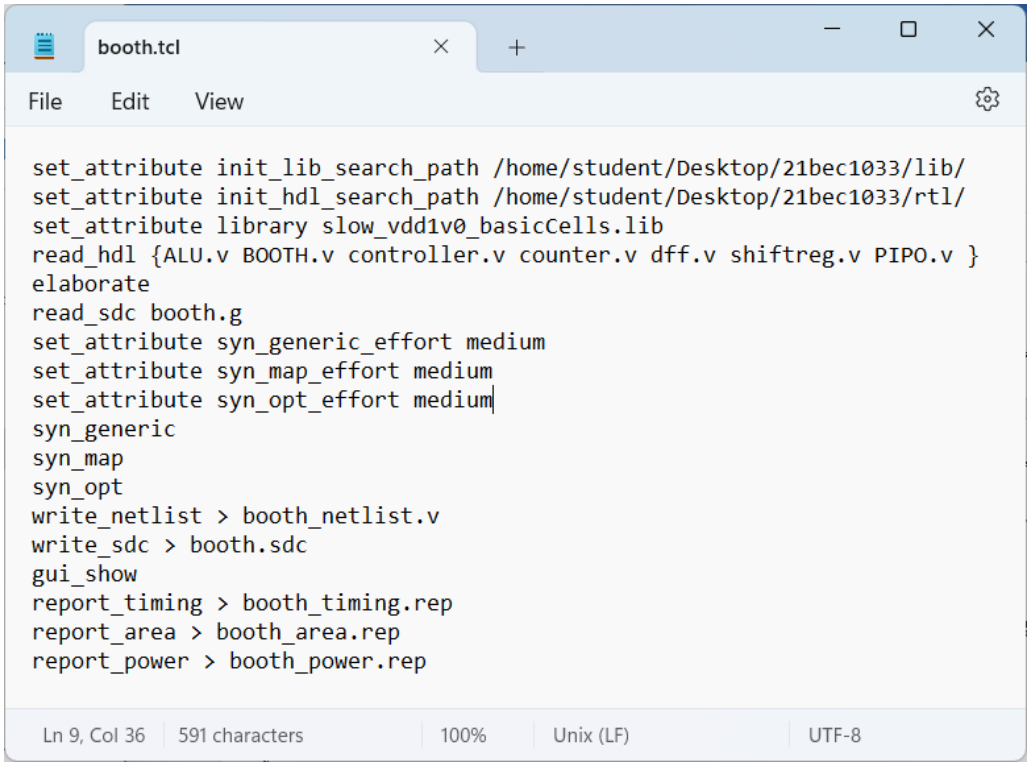
1. Copy the fast.lib and slow.lib files into the folder where (.v) files are located.

2. Create a Synopsys design constraint file (.g file) by entering the design constraints required for the synthesis.



LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

3. Create a (.tcl) file containing all the commands for performing the logic synthesis. Synthesis effort can be medium or high.



4. Invoke the C shell and launch the Genus tool by entering the below commands.
   'csh'
   'source /home/install/cshrc'
   'genus -legacy -ui -f booth.tcl'

5. Execute the commands in the (.tcl file) by entering **source booth.tcl** command in the command line window.

6. Check for the area, timing and power reports generated in the respective multiplier folder. Also check the gate level netlist generated in the Genus synthesis solution window.

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

Verilog Programs:

//Verilog Program of top-level module containing data path and control path

BOOTH.v

File Edit View

```
module BOOTH(data_in,clk,start,result);

wire ldA,clrA,sftA,ldQ,clrQ,sftQ,done;
wire ldM,clrFF,addsub,decr,ldcount;
input signed [15:0]data_in;
input start,clk;
wire qm1,eqz;
output signed [31:0]result;
wire [4:0] countdata;
wire signed [15:0] A,M,Q,Z;
assign eqz = ~| (countdata);
assign result = {A,Q};

shiftreg
Acc_A(.data_out(A),.ld_data(Z),.s_in(A[15]),.clk(clk),.ld(ldA),.rst(clrA),.sft(sftA));

shiftreg
Mlpr_Q(.data_out(Q),.ld_data(data_in),.s_in(A[0]),.clk(clk),.ld(ldQ),.rst(clrQ),.sft(sftQ));

dff d_ff(.data_in(Q[0]),.clk(clk),.rst(clrFF),.data_out(qm1));

PIPO Mlcd_M(.data_out(M),.data_in(data_in),.clk(clk),.ld(ldM));

ALU Adsub(.out(Z),.in1(A),.in2(M),.addsub(addsub));

counter count(.countdata(countdata),.decr(decr),.ldcount(ldcount),.clk(clk));

controller fsm(.ldA(ldA),.clrA(clrA),.sftA(sftA),.ldQ(ldQ),.clrQ(clrQ),.sftQ(sftQ),.ldM(ldM),
               .clrFF(clrFF),.addsub(addsub),.start(start),.decr(decr),.ldcount(ldcount),
               .done(done),.clk(clk),.q0(Q[0]),.qm1(qm1),.eqz(eqz));

endmodule
```

Ln 1, Col 1 | 1,028 characters | 100% | Windows (CRLF) | UTF-8

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

//Verilog Program of control path module

controller.v

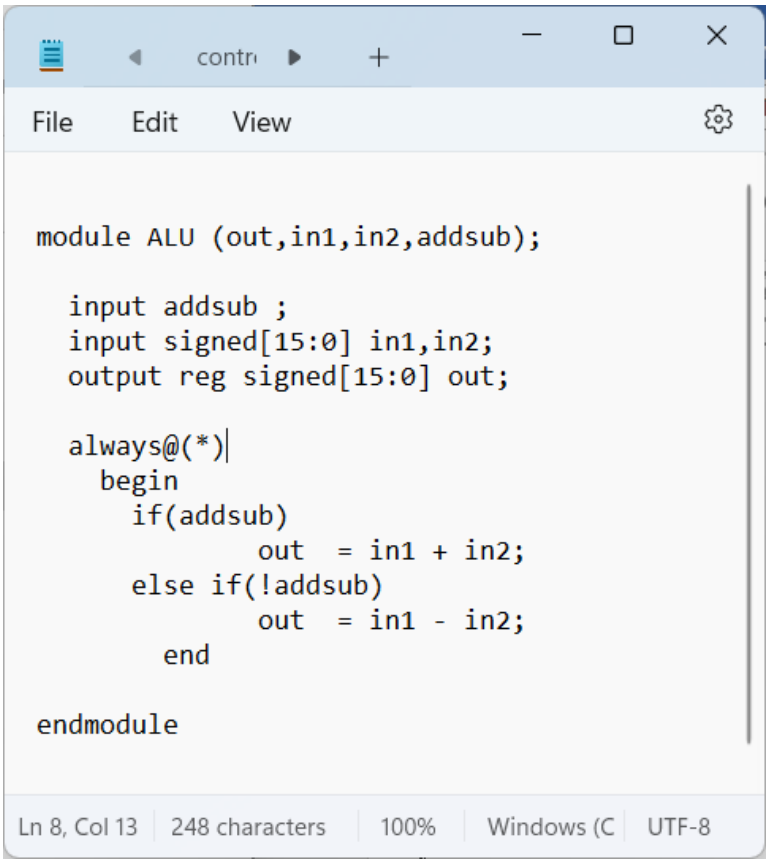
File Edit View

```
module
controller(ldA,clrA,sftA,ldQ,clrQ,sftQ,ldM,clrFF,addsub,start,decr,ldcount,done,clk,q0,qm1,eqz);
input clk,q0,qm1,start,eqz;
output ldA,clrA,sftA,ldQ,clrQ,sftQ,ldM,clrFF,addsub,decr,ldcount,done;
reg [2:0] state;
parameter [2:0] S0 = 3'b000,S1 = 3'b001,S2 = 3'b010,S3 = 3'b011,S4 = 3'b100,S5 = 3'b101,S6 = 3'b110;
always@(posedge clk)
        begin
                case(state)
                        S0 :  state <= start?S1:S0;
                        S1 :  state <= S2;
                        S2 :  begin
                                        #1
                                        if({q0,qm1}==2'b01)
                                                state <=  S3;
                                        else if({q0,qm1}==2'b10)
                                                state <=  S4;
                                        else if(({q0,qm1}==2'b00)||({q0,qm1}==2'b11))
                                                state <=  S5;
                                end
                        S3 :  state <=  S5;
                        S4 :  state <=  S5;
                        S5 :  begin
                                       #1
                                        if(({q0,qm1}==2'b01) && !eqz)
                                                state <=  S3;
                                        else if(({q0,qm1}==2'b10) && !eqz)
                                                state <=  S4;
                                        else if(eqz)
                                                state <= S6;
                                end
                        S6 :  state <= S6;
                default : state <= S0;
                endcase
        end


assign ldA     = ((state == S3)||(state == S4))?1:0;
assign clrA    =  (state == S0)?1:0;
assign sftA    =  (state == S5)?1:0;
assign ldQ     =  (state == S2)?1:0;
assign clrQ    =  (state == S0)?1:0;
assign sftQ    =  (state == S5)?1:0;
assign ldM     =  (state == S1)?1:0;
assign clrFF   =  (state == S0)?1:0;
assign addsub     =  (state == S3)?1:0;

assign decr    =  (state == S5)?1:0;
assign ldcount =  (state == S1)?1:0;
assign done    =  (state == S6)?1:0;

endmodule
```

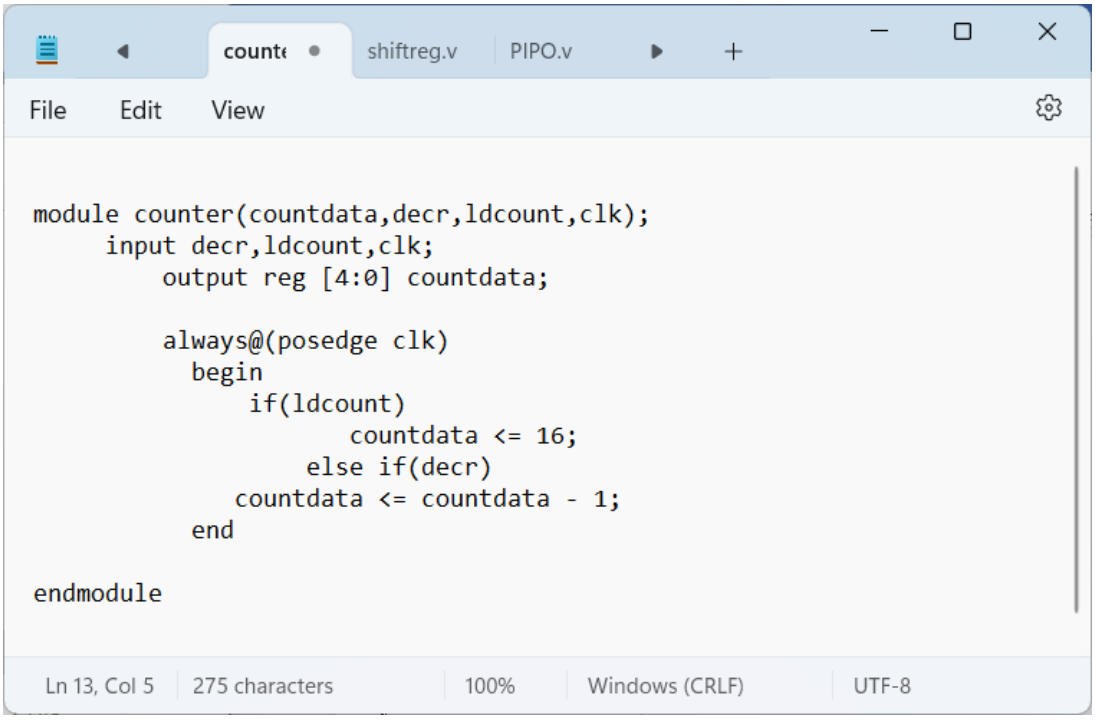
Ln 1, Col 1 | 1,531 characters | 60% | Windows (CRLF) | UTF-8

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

//Verilog Program of ALU



//Verilog Program of counter



LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

//Verilog Program of D-FF

shiftreg.v | PIPO.v | dff.v

File Edit View

```
module dff(data_in,clk,rst,data_out);

input data_in,clk,rst;
output reg data_out;

always@(posedge clk)
   begin
        if(rst)
                   data_out <= 0;
                else
                   data_out <= data_in;
   end

endmodule
```

Ln 1, Col 1 | 208 characters | 100% | Windows (CRLF) | UTF-8

//Verilog Program of PIPO

PIPO.v

File Edit View

```
module PIPO(data_out,data_in,clk,ld);

input signed [15:0] data_in;
input clk,ld;
output reg signed [15:0] data_out;

always@(posedge clk)
  begin
      if(ld)
        data_out <= data_in;
   end
endmodule
```

Ln 1, Col 1 | 209 characters | 100% | Windows (CRLF) | UTF-8

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

//Verilog Program of shiftregister

shiftreg.v

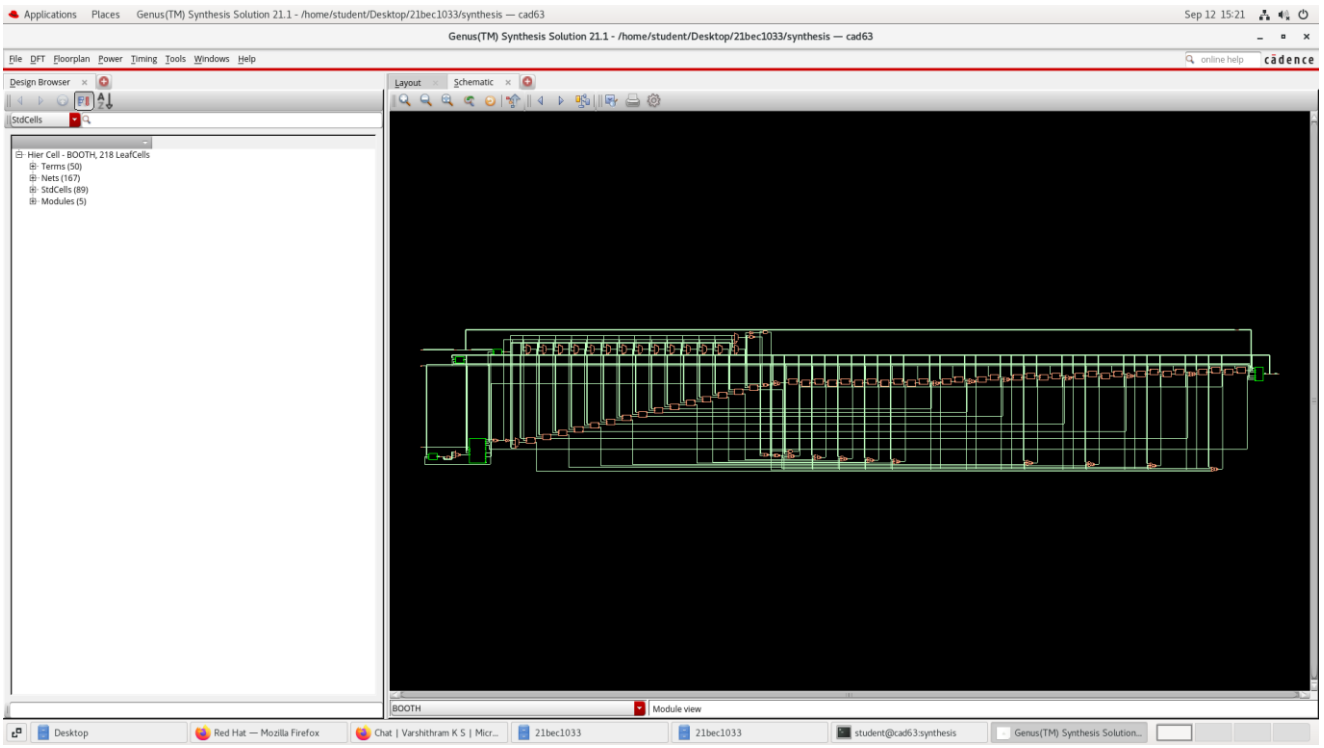
File Edit View

```
module shiftreg(data_out,ld_data,s_in,clk,ld,rst,sft);

input [15:0] ld_data;
input clk,rst,ld,s_in,sft;
output reg [15:0]data_out;

always@(posedge clk)
 begin
    if(rst)
            data_out <= 0;
        else if(ld)
        data_out <= ld_data;
    else if(sft)
        data_out <= {s_in,data_out[15:1]};
 end

 endmodule
```

Ln 1, Col 1 | 319 characters | 90% | Windows (CRLF) | UTF-8

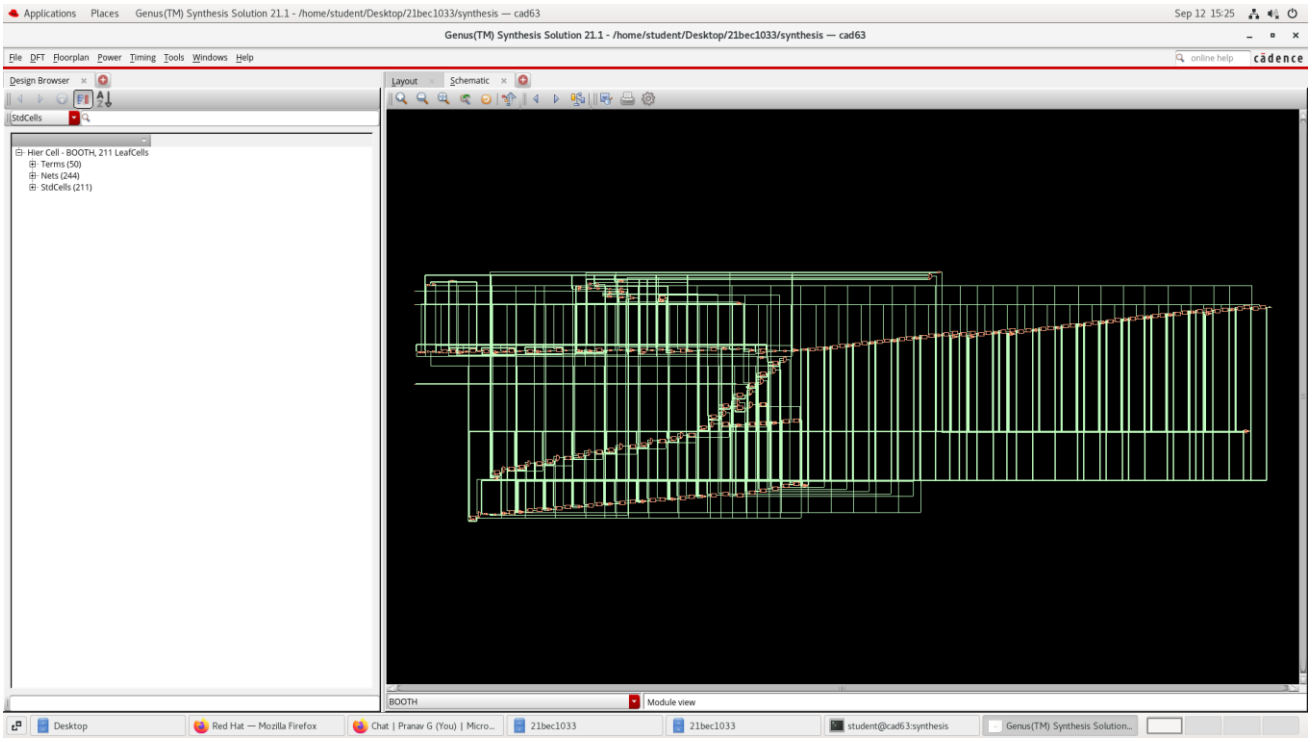
Gate level Netlist:

Synthesis effort: Medium



LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

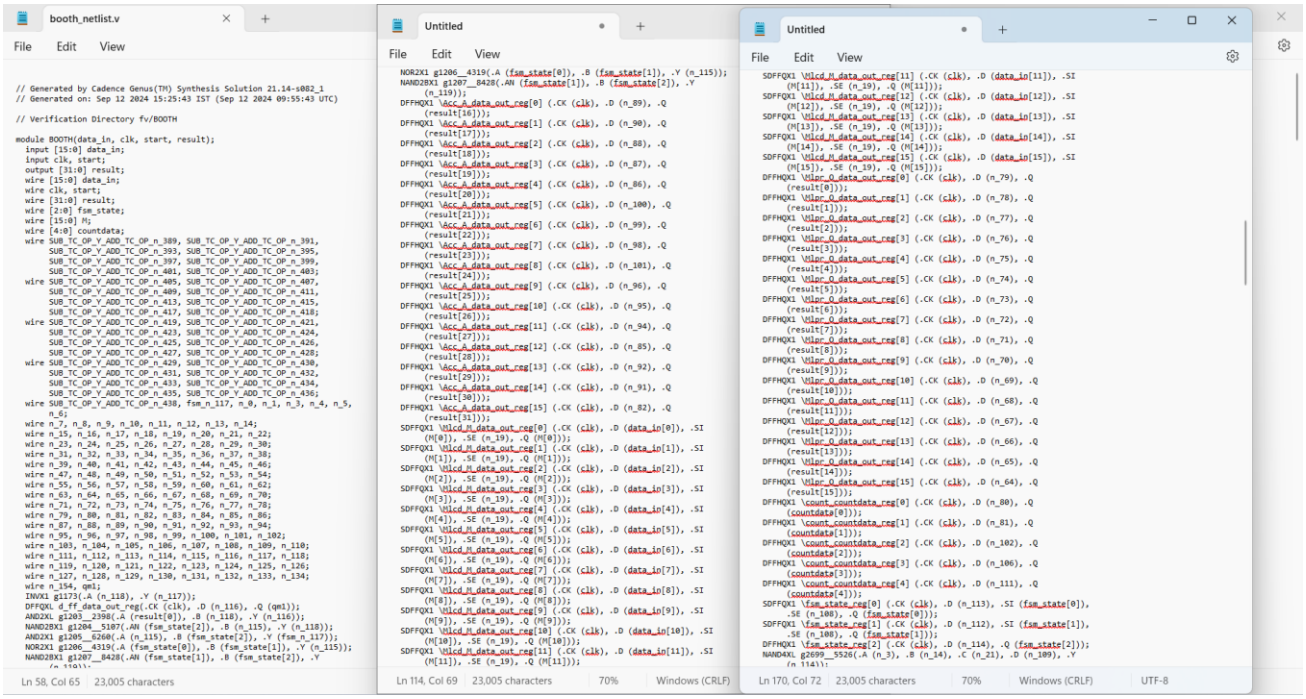
Synthesis effort : High

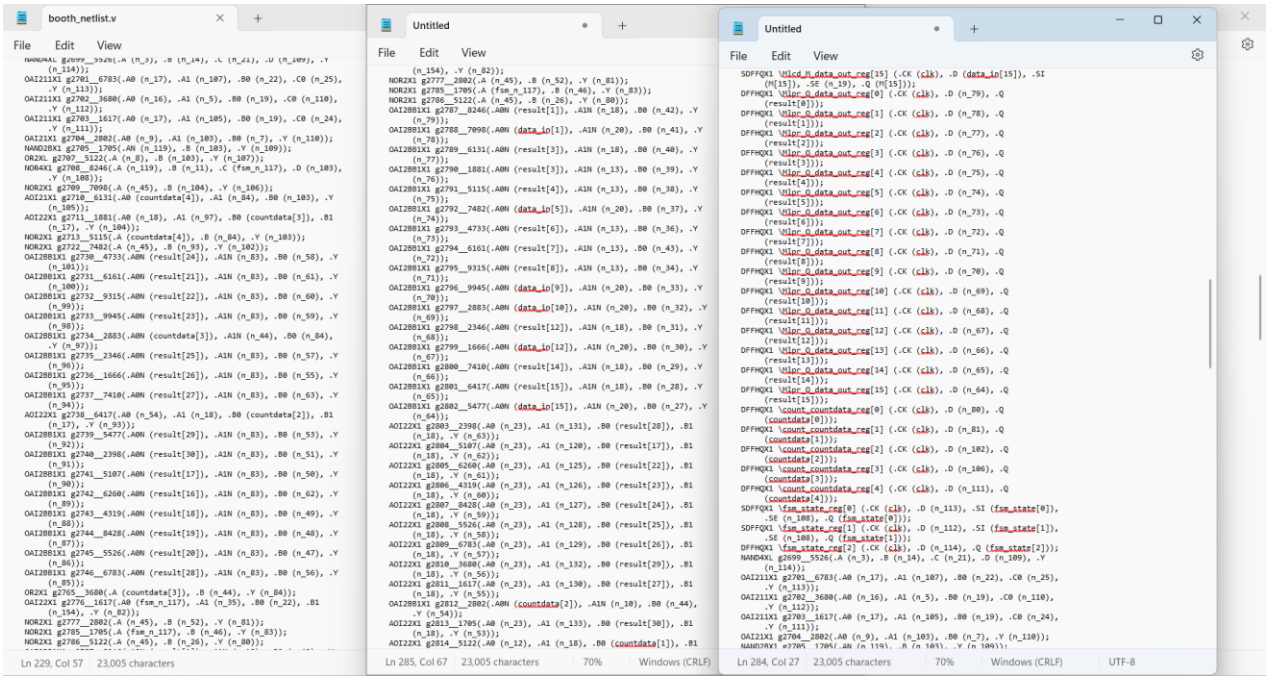


LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

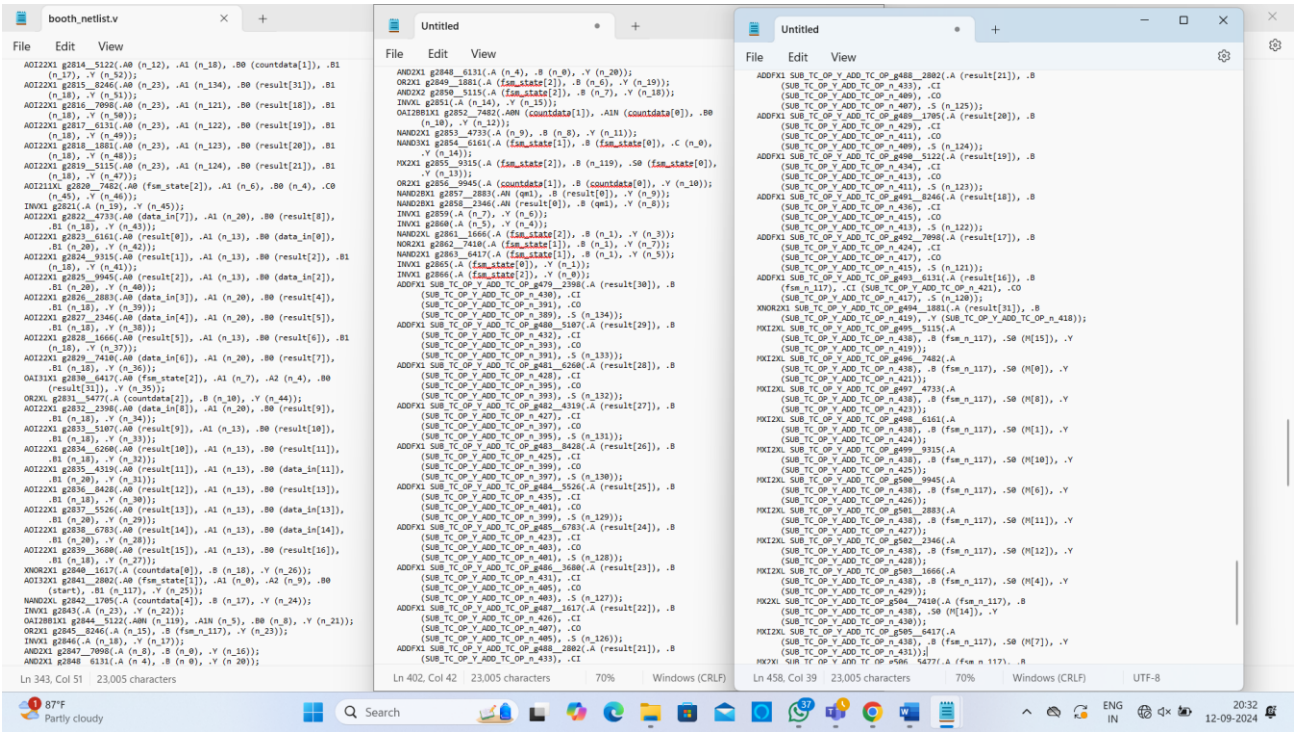
Observations:

a) Booth_multiplier_netlist_created file





LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic



```
booth_netlist.v
File  Edit  View

  MXI2XL SUB_TC_OP_Y_ADD_TC_OP_g503__1666(.A
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .B (fsm_n_117), .S0 (M[4]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_429));
  MX2XL SUB_TC_OP_Y_ADD_TC_OP_g504__7410(.A (fsm_n_117), .B
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .S0 (M[14]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_430));
  MXI2XL SUB_TC_OP_Y_ADD_TC_OP_g505__6417(.A
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .B (fsm_n_117), .S0 (M[7]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_431));
  MX2XL SUB_TC_OP_Y_ADD_TC_OP_g506__5477(.A (fsm_n_117), .B
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .S0 (M[13]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_432));
  MXI2XL SUB_TC_OP_Y_ADD_TC_OP_g507__2398(.A
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .B (fsm_n_117), .S0 (M[5]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_433));
  MX2XL SUB_TC_OP_Y_ADD_TC_OP_g508__5107(.A (fsm_n_117), .B
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .S0 (M[3]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_434));
  MXI2XL SUB_TC_OP_Y_ADD_TC_OP_g509__6260(.A
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .B (fsm_n_117), .S0 (M[9]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_435));
  MX2XL SUB_TC_OP_Y_ADD_TC_OP_g510__4319(.A (fsm_n_117), .B
       (SUB_TC_OP_Y_ADD_TC_OP_n_438), .S0 (M[2]), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_436));
  INVX1 SUB_TC_OP_Y_ADD_TC_OP_g511(.A (fsm_n_117), .Y
       (SUB_TC_OP_Y_ADD_TC_OP_n_438));
  XOR2XL g2(.A (SUB_TC_OP_Y_ADD_TC_OP_n_418), .B
       (SUB_TC_OP_Y_ADD_TC_OP_n_389), .Y (n_154));
endmodule

Ln 455, Col 39  23,005 characters  90%  Unix (LF)  UTF-8
```

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

b) Booth_multiplier_constraint_created file

booth.sdc

File Edit View

```
# ####################################################################

#  Created by Genus(TM) Synthesis Solution 21.14-s082_1 on Thu Sep 12 15:25:43 IST 2024

# ####################################################################

set sdc_version 2.0

set_units -capacitance 1000fF
set_units -time 1000ps

# Set the current design
current_design BOOTH

create_clock -name "clk" -period 10.0 -waveform {0.0 5.0} [get_ports clk]
set_clock_transition 0.1 [get_clocks clk]
set_clock_gating_check -setup 0.0
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[15]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[14]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[13]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[12]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[11]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[10]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[9]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[8]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[7]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[6]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[5]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[4]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[3]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[2]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[1]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports {data_in[0]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports start]
set_wire_load_mode "enclosed"
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]
```

Ln 1, Col 1 2,079 characters

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

c) Booth_multiplier_timing_created file

Synthesis effort: Medium

booth_timing.rep

booth_timing.rep

File Edit View

```
|=============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:21:21 pm
  Module:                 BOOTH
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
=============================================================

               Pin                      Type       Fanout Load Slew Delay Arrival
                                                         (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------------
(clock clk)                           launch                                    0 R
fsm
  state_reg[0]/CK                                                100    +0       0 R
  state_reg[0]/Q                      SDFFQX1           7  1.6   53  +257     257 F
  g574__5526/AN                                                        +0     257
  g574__5526/Y                        NAND3BXL          2  0.8  200  +201     458 F
  g568__5477/A                                                         +0     458
  g568__5477/Y                        NAND2X1           4  1.2   73  +142     600 R
fsm/ldA
g571__1666/AN                                                          +0     600
g571__1666/Y                          NOR2BX1          18  4.4  218  +215     815 R
SUB_TC_OP_Y_ADD_TC_OP_g1628/A                                          +0     815
SUB_TC_OP_Y_ADD_TC_OP_g1628/Y         INVX1            16  3.2  127  +184     998 F
SUB_TC_OP_Y_ADD_TC_OP_g1619__5107/A                                    +0     998
SUB_TC_OP_Y_ADD_TC_OP_g1619__5107/Y   MXI2XL            1  0.6   95  +139    1138 R
SUB_TC_OP_Y_ADD_TC_OP_g1610__9315/CI                                   +0    1138
SUB_TC_OP_Y_ADD_TC_OP_g1610__9315/CO  ADDFX1            1  0.6   39  +215    1352 R
SUB_TC_OP_Y_ADD_TC_OP_g1609__6161/CI                                   +0    1352
SUB_TC_OP_Y_ADD_TC_OP_g1609__6161/CO  ADDFX1            1  0.6   39  +186    1538 R
SUB_TC_OP_Y_ADD_TC_OP_g1608__4733/CI                                   +0    1538
SUB_TC_OP_Y_ADD_TC_OP_g1608__4733/CO  ADDFX1            1  0.6   39  +186    1724 R
SUB_TC_OP_Y_ADD_TC_OP_g1607__7482/CI                                   +0    1724
SUB_TC_OP_Y_ADD_TC_OP_g1607__7482/CO  ADDFX1            1  0.6   39  +186    1910 R
SUB_TC_OP_Y_ADD_TC_OP_g1606__5115/CI                                   +0    1910
SUB_TC_OP_Y_ADD_TC_OP_g1606__5115/CO  ADDFX1            1  0.6   39  +186    2096 R
SUB_TC_OP_Y_ADD_TC_OP_g1605__1881/CI                                   +0    2096
SUB_TC_OP_Y_ADD_TC_OP_g1605__1881/CO  ADDFX1            1  0.6   39  +186    2282 R
SUB_TC_OP_Y_ADD_TC_OP_g1604__6131/CI                                   +0    2282
SUB_TC_OP_Y_ADD_TC_OP_g1604__6131/CO  ADDFX1            1  0.6   39  +186    2468 R
SUB_TC_OP_Y_ADD_TC_OP_g1603__7098/CI                                   +0    2468
SUB_TC_OP_Y_ADD_TC_OP_g1603__7098/CO  ADDFX1            1  0.6   39  +186    2654 R
SUB_TC_OP_Y_ADD_TC_OP_g1602__8246/CI                                   +0    2654
SUB_TC_OP_Y_ADD_TC_OP_g1602__8246/CO  ADDFX1            1  0.6   39  +186    2840 R
SUB_TC_OP_Y_ADD_TC_OP_g1601__5122/CI                                   +0    2840
SUB_TC_OP_Y_ADD_TC_OP_g1601__5122/CO  ADDFX1            1  0.6   39  +186    3026 R
SUB_TC_OP_Y_ADD_TC_OP_g1600__1705/CI                                   +0    3026
SUB_TC_OP_Y_ADD_TC_OP_g1600__1705/CO  ADDFX1            1  0.6   39  +186    3212 R
SUB_TC_OP_Y_ADD_TC_OP_g1599__2802/CI                                   +0    3212
SUB_TC_OP_Y_ADD_TC_OP_g1599__2802/CO  ADDFX1            1  0.6   39  +186    3398 R
SUB_TC_OP_Y_ADD_TC_OP_g1598__1617/CI                                   +0    3398
SUB_TC_OP_Y_ADD_TC_OP_g1598__1617/CO  ADDFX1            1  0.6   39  +186    3584 R
SUB_TC_OP_Y_ADD_TC_OP_g1597__3680/CI                                   +0    3584
SUB_TC_OP_Y_ADD_TC_OP_g1597__3680/CO  ADDFX1            1  0.6   39  +186    3770 R
SUB_TC_OP_Y_ADD_TC_OP_g1596__6783/CI                                   +0    3770
SUB_TC_OP_Y_ADD_TC_OP_g1596__6783/CO  ADDFX1            1  0.4   35  +184    3954 R
SUB_TC_OP_Y_ADD_TC_OP_g1595__5526/B                                    +0    3954
SUB_TC_OP_Y_ADD_TC_OP_g1595__5526/Y   XNOR2X1           1  0.2   22  +123    4077 R
g1094__2398/B1                                                         +0    4077
g1094__2398/Y                         AO22XL            1  0.3   28  +128    4205 R
Acc_A_data_out_reg[15]/D              DFFHQX1                        +0    4205
Acc_A_data_out_reg[15]/CK             setup                   100   +90    4295 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                           capture                               10000 R
                                      uncertainty                 -10    9990 R
-------------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    5695ps
Start-point  : fsm/state_reg[0]/CK
End-point    : Acc_A_data_out_reg[15]/D
```

Ln 1, Col 1 | 5,419 characters

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

Synthesis effort: High

booth_timing.rep

File Edit View

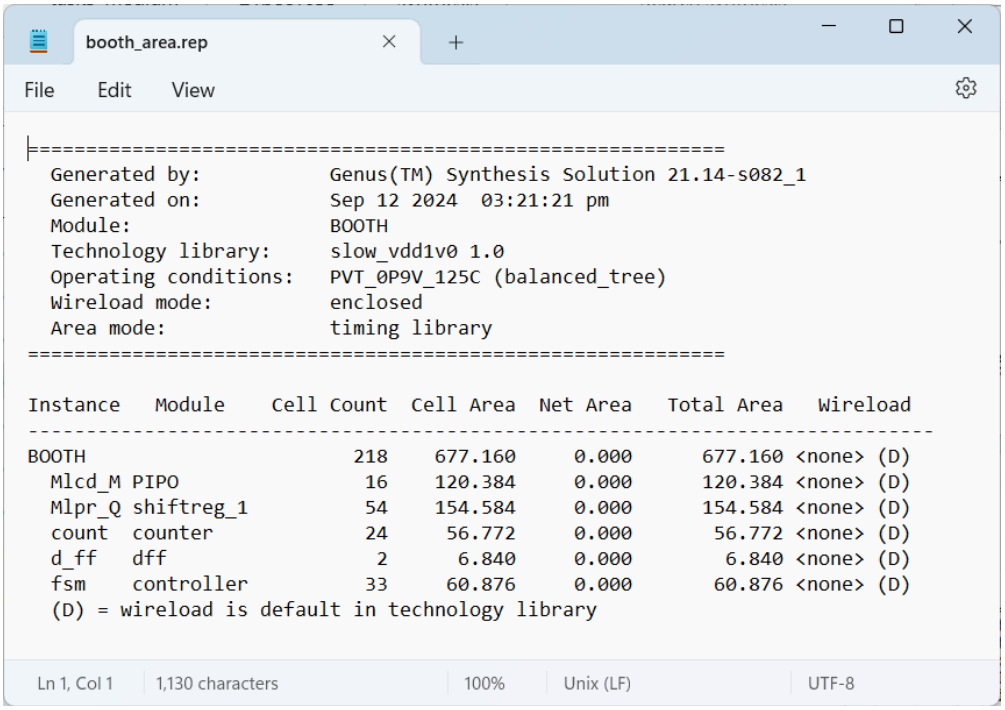
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:25:43 pm
  Module:                 BOOTH
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

             Pin                          Type       Fanout Load Slew Delay Arrival
                                                     (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------------
(clock clk)                             launch                                  0 R
fsm_state_reg[1]/CK                                           100    +0       0 R
fsm_state_reg[1]/Q                      SDFFQX1            7  2.3   70  +267     267 F
g1206__4319/B                                                        +0     267
g1206__4319/Y                           NOR2X1             2  0.6   58   +73     340 R
g1205__6260/A                                                        +0     340
g1205__6260/Y                           AND2X1            22  5.8  135  +197     536 R
SUB_TC_OP_Y_ADD_TC_OP_g511/A                                         +0     536
SUB_TC_OP_Y_ADD_TC_OP_g511/Y            INVX1             16  3.2  112  +134     670 F
SUB_TC_OP_Y_ADD_TC_OP_g496__7482/A                                   +0     670
SUB_TC_OP_Y_ADD_TC_OP_g496__7482/Y      MXI2XL             1  0.6   96  +131     802 R
SUB_TC_OP_Y_ADD_TC_OP_g493__6131/CI                                  +0     802
SUB_TC_OP_Y_ADD_TC_OP_g493__6131/CO     ADDFX1             1  0.6   39  +215    1017 R
SUB_TC_OP_Y_ADD_TC_OP_g492__7098/CI                                  +0    1017
SUB_TC_OP_Y_ADD_TC_OP_g492__7098/CO     ADDFX1             1  0.6   39  +186    1203 R
SUB_TC_OP_Y_ADD_TC_OP_g491__8246/CI                                  +0    1203
SUB_TC_OP_Y_ADD_TC_OP_g491__8246/CO     ADDFX1             1  0.6   39  +186    1389 R
SUB_TC_OP_Y_ADD_TC_OP_g490__5122/CI                                  +0    1389
SUB_TC_OP_Y_ADD_TC_OP_g490__5122/CO     ADDFX1             1  0.6   39  +186    1575 R
SUB_TC_OP_Y_ADD_TC_OP_g489__1705/CI                                  +0    1575
SUB_TC_OP_Y_ADD_TC_OP_g489__1705/CO     ADDFX1             1  0.6   39  +186    1761 R
SUB_TC_OP_Y_ADD_TC_OP_g488__2802/CI                                  +0    1761
SUB_TC_OP_Y_ADD_TC_OP_g488__2802/CO     ADDFX1             1  0.6   39  +186    1947 R
SUB_TC_OP_Y_ADD_TC_OP_g487__1617/CI                                  +0    1947
SUB_TC_OP_Y_ADD_TC_OP_g487__1617/CO     ADDFX1             1  0.6   39  +186    2133 R
SUB_TC_OP_Y_ADD_TC_OP_g486__3680/CI                                  +0    2133
SUB_TC_OP_Y_ADD_TC_OP_g486__3680/CO     ADDFX1             1  0.6   39  +186    2319 R
SUB_TC_OP_Y_ADD_TC_OP_g485__6783/CI                                  +0    2319
SUB_TC_OP_Y_ADD_TC_OP_g485__6783/CO     ADDFX1             1  0.6   39  +186    2505 R
SUB_TC_OP_Y_ADD_TC_OP_g484__5526/CI                                  +0    2505
SUB_TC_OP_Y_ADD_TC_OP_g484__5526/CO     ADDFX1             1  0.6   39  +186    2691 R
SUB_TC_OP_Y_ADD_TC_OP_g483__8428/CI                                  +0    2691
SUB_TC_OP_Y_ADD_TC_OP_g483__8428/CO     ADDFX1             1  0.6   39  +186    2877 R
SUB_TC_OP_Y_ADD_TC_OP_g482__4319/CI                                  +0    2877
SUB_TC_OP_Y_ADD_TC_OP_g482__4319/CO     ADDFX1             1  0.6   39  +186    3063 R
SUB_TC_OP_Y_ADD_TC_OP_g481__6260/CI                                  +0    3063
SUB_TC_OP_Y_ADD_TC_OP_g481__6260/CO     ADDFX1             1  0.6   39  +186    3249 R
SUB_TC_OP_Y_ADD_TC_OP_g480__5107/CI                                  +0    3249
SUB_TC_OP_Y_ADD_TC_OP_g480__5107/CO     ADDFX1             1  0.6   39  +186    3435 R
SUB_TC_OP_Y_ADD_TC_OP_g479__2398/CI                                  +0    3435
SUB_TC_OP_Y_ADD_TC_OP_g479__2398/S      ADDFX1             1  0.4   37  +257    3692 R
g2815__8246/A1                                                       +0    3692
g2815__8246/Y                           AOI22X1            1  0.4  108  +102    3794 F
g2740__2398/B0                                                       +0    3794
g2740__2398/Y                           OAI2BB1X1          1  0.3   40   +80    3874 R
Acc_A_data_out_reg[14]/D           <<<  DFFHQX1                             +0    3874
Acc_A_data_out_reg[14]/CK               setup                       100   +96    3970 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)                             capture                               10000 R
                                        uncertainty                   -10    9990 R
-------------------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    6020ps
Start-point  : fsm_state_reg[1]/CK
End-point    : Acc_A_data_out_reg[14]/D
```

Ln 1, Col 1 | 5,342 characters

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

d) multiplier_area_created file

Synthesis effort: Medium



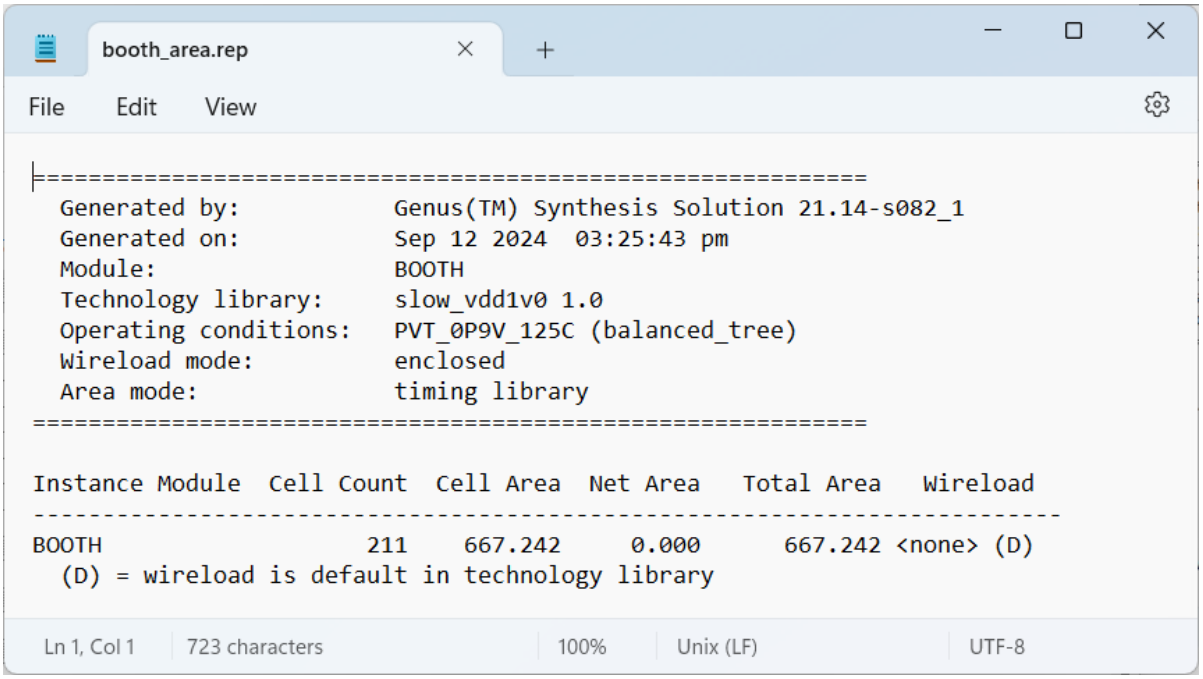
booth_area.rep

File Edit View

```
╞============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:21:21 pm
  Module:                 BOOTH
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance   Module    Cell Count  Cell Area  Net Area   Total Area   Wireload
-----------------------------------------------------------------------------
BOOTH                     218    677.160     0.000      677.160 <none> (D)
  Mlcd_M PIPO              16    120.384     0.000      120.384 <none> (D)
  Mlpr_Q shiftreg_1        54    154.584     0.000      154.584 <none> (D)
  count  counter           24     56.772     0.000       56.772 <none> (D)
  d_ff   dff                2      6.840     0.000        6.840 <none> (D)
  fsm    controller        33     60.876     0.000       60.876 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 1,130 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High



booth_area.rep

File Edit View

```
╞============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:25:43 pm
  Module:                 BOOTH
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-----------------------------------------------------------------------------
BOOTH                   211    667.242     0.000      667.242 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 723 characters | 100% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

e) multiplier_power_created file

Synthesis effort: Medium

booth_power.rep

File Edit View

```
Instance: /BOOTH
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ---------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     6.79047e-09  1.60959e-05  8.38848e-07  1.69416e-05  56.35%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     7.16841e-09  9.07615e-06  3.11648e-06  1.21998e-05  40.58%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  9.23400e-07  9.23400e-07   3.07%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     1.39589e-08  2.51721e-05  4.87873e-06  3.00647e-05 100.00%
  Percentage           0.05%       83.73%       16.23%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,199 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High

booth_area.rep | booth_power.rep

File Edit View

```
Instance: /BOOTH
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ---------------------------------------------------------------------------
       memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     register     6.79055e-09  1.63542e-05  8.64179e-07  1.72251e-05  58.48%
        latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        logic     6.88496e-09  8.53146e-06  2.76721e-06  1.13056e-05  38.38%
         bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        clock     0.00000e+00  0.00000e+00  9.23400e-07  9.23400e-07   3.14%
          pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
           pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     1.36755e-08  2.48856e-05  4.55479e-06  2.94541e-05 100.00%
  Percentage           0.05%       84.49%       15.46%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 14, Col 47 | 1,199 characters | 100% | Unix (LF) | UTF-8

LAB4_2: Logic synthesis of a Booth Multiplier using Datapath and Control path logic

Inference:

Synthesis effort: Medium

A total of 218 leaf instance count is present in the gate level netlist with total area of 677.160, total power of 3.00647e-05W.

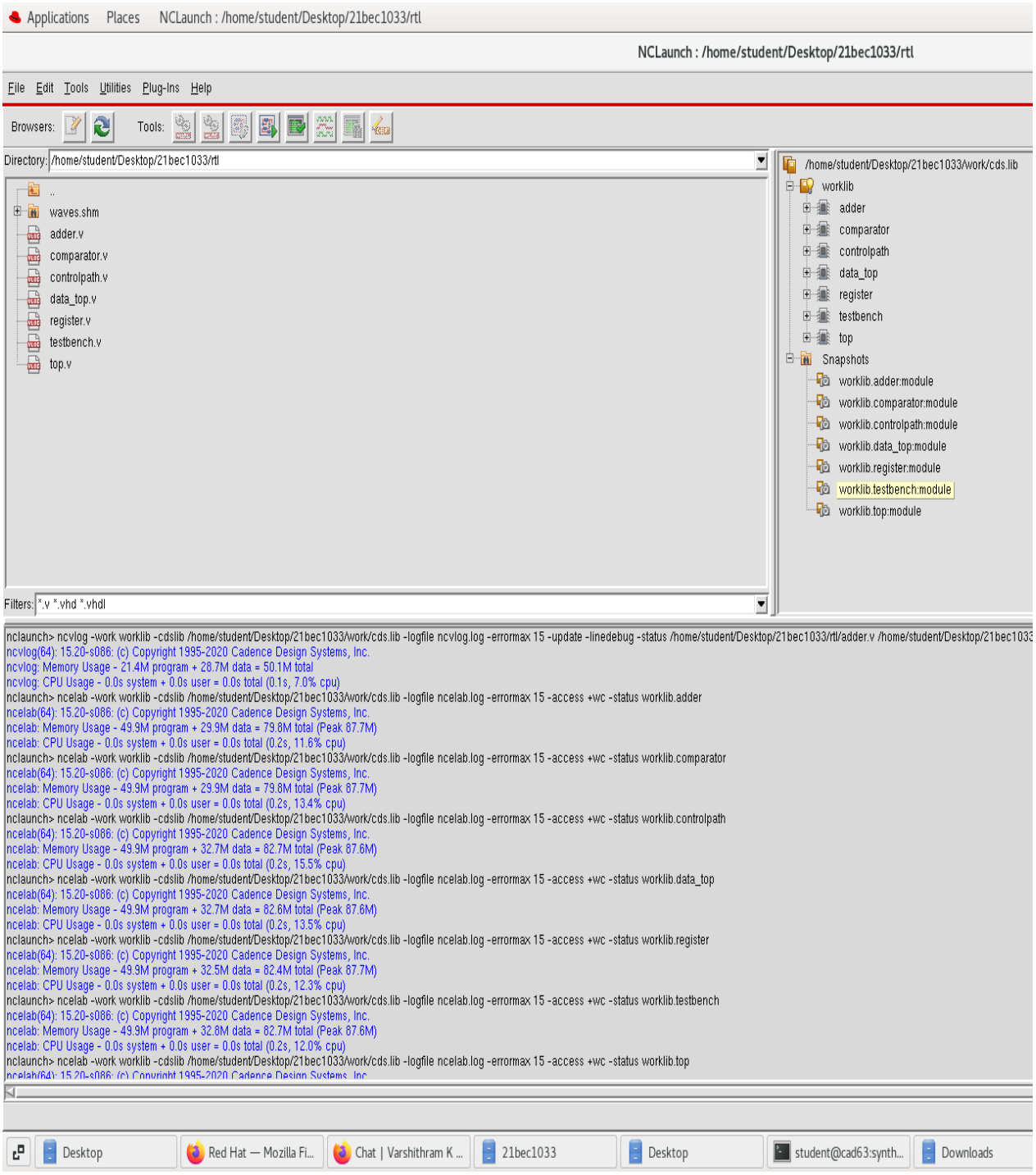
Synthesis effort: High

A total of 211 leaf instance count is present in the gate level netlist with total area of 667.242 , total power of 2.94541e-05W.

Result: Hence a Booth multiplier using Datapath and Control path logic is synthesized and the gate level netlist with timing, area and power report has been generated.

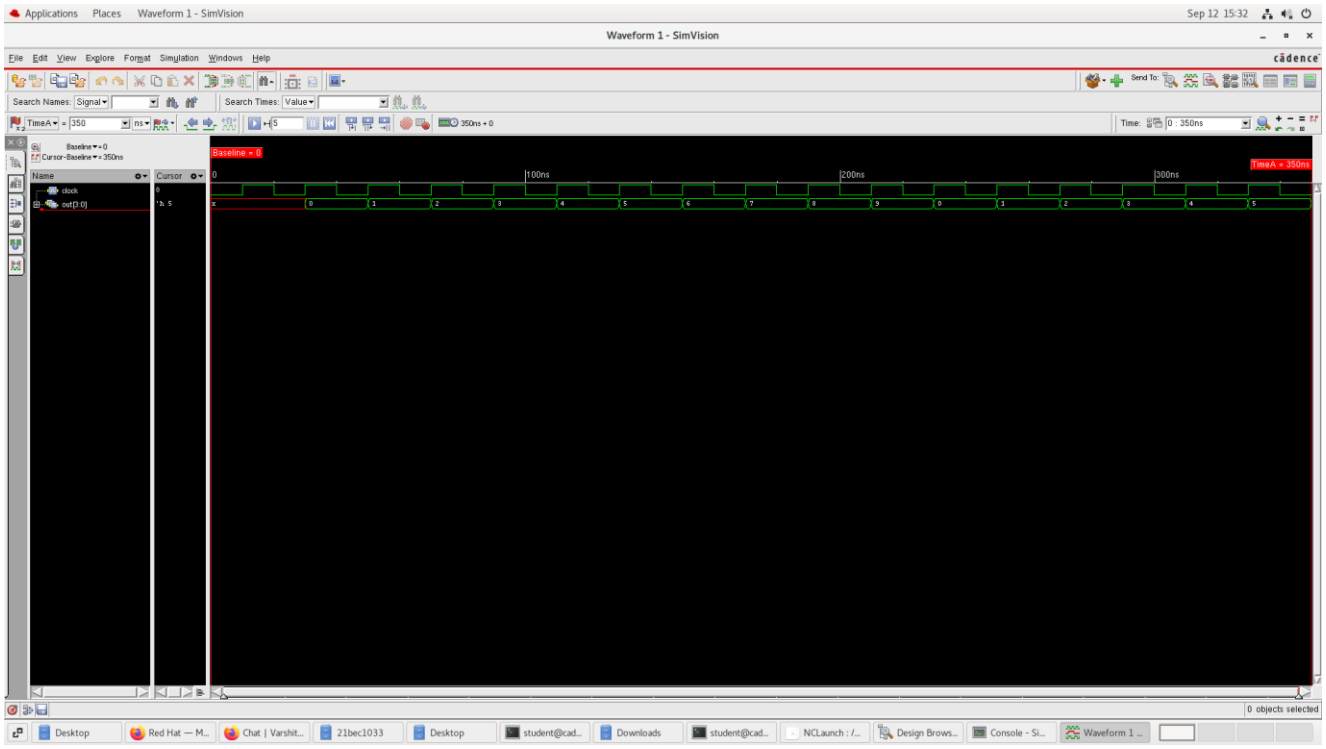
Lab4_4: Design and simulation of a Binary Counter using Datapath and Control path logic

NCLaunch Compilation and Elaboration:



Lab4_4: Design and simulation of a Binary Counter using Datapath and Control path logic

Observations/Simulation Waveforms:



Inference: In the above output waveform, the “out” signal increases by 1 as and when a posedge of the clock occurs. If the out has reached the maximum value then it become zero when it encounters a posedge of the clock.

Result: Hence, a Binary_counter is designed and the simulation and verification of behavior of Binary_counter using data path and Control path logic is performed for the given stimuli in the test bench.

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

Aim: To synthesize a Binary_counter using Datapath and Control path logic and to get the gate level netlist with timing, area, and power reports.

EDA Tools Used: Cadence Genus

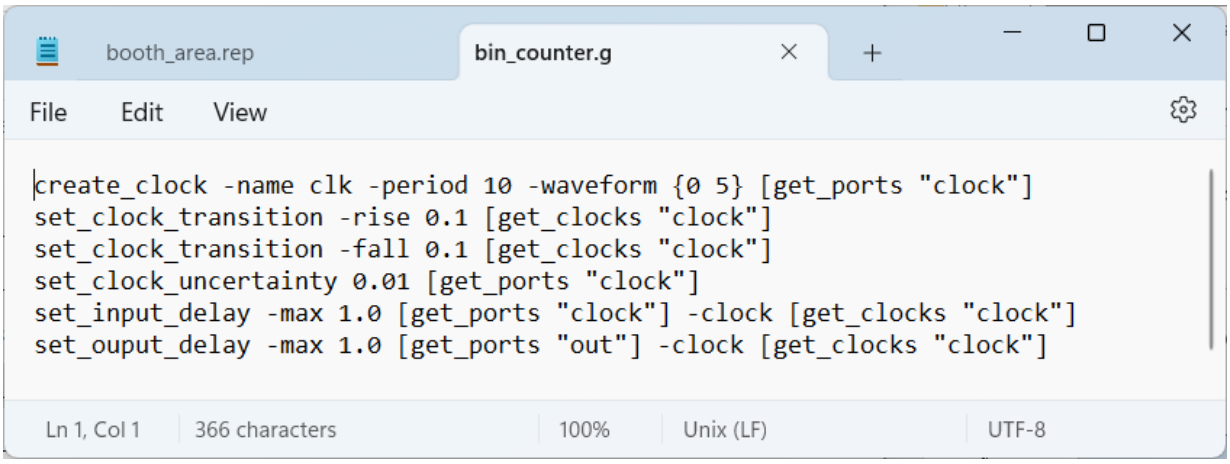
Description:

Datapath is the hardware that performs all the required operations. Control is the hardware that tells the datapath what to do, in terms of switching, operation selection, data movement between ALU components, etc. Simple datapath components include memory (stores the current instruction), PC or program counter (stores the address of current instruction), and ALU (executes current instruction). Control path is the Finite state machine designed by using Moore or Mealy models.

Procedure:

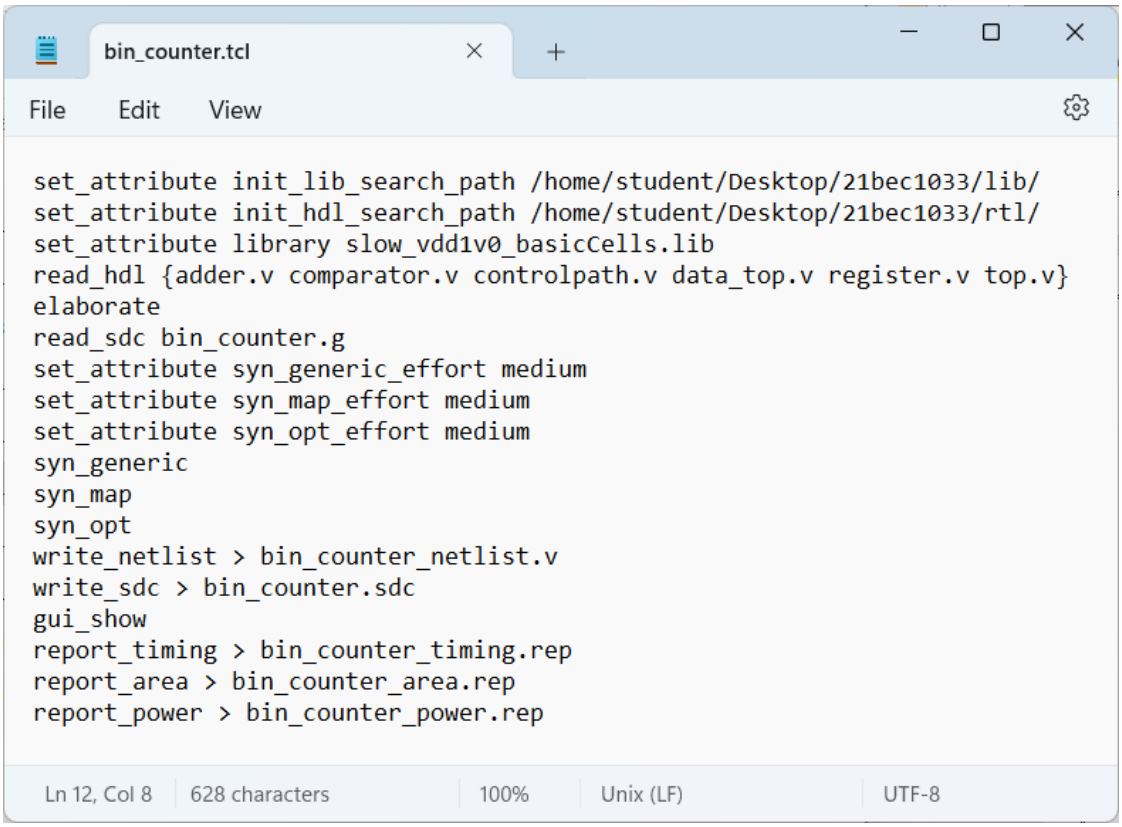
1. Copy the fast.lib and slow.lib files into the folder where (.v) files are located.

2. Create a Synopsys design constraint file (.g file) by entering the design constraints required for the synthesis.



LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

3. Create a (.tcl) file containing all the commands for performing the logic synthesis. Synthesis effort can be medium or high.



4. Invoke the C shell and launch the Genus tool by entering the below commands.
   'csh'
   'source /home/install/cshrc'
   'genus -legacy -ui -f bin_counter.tcl'

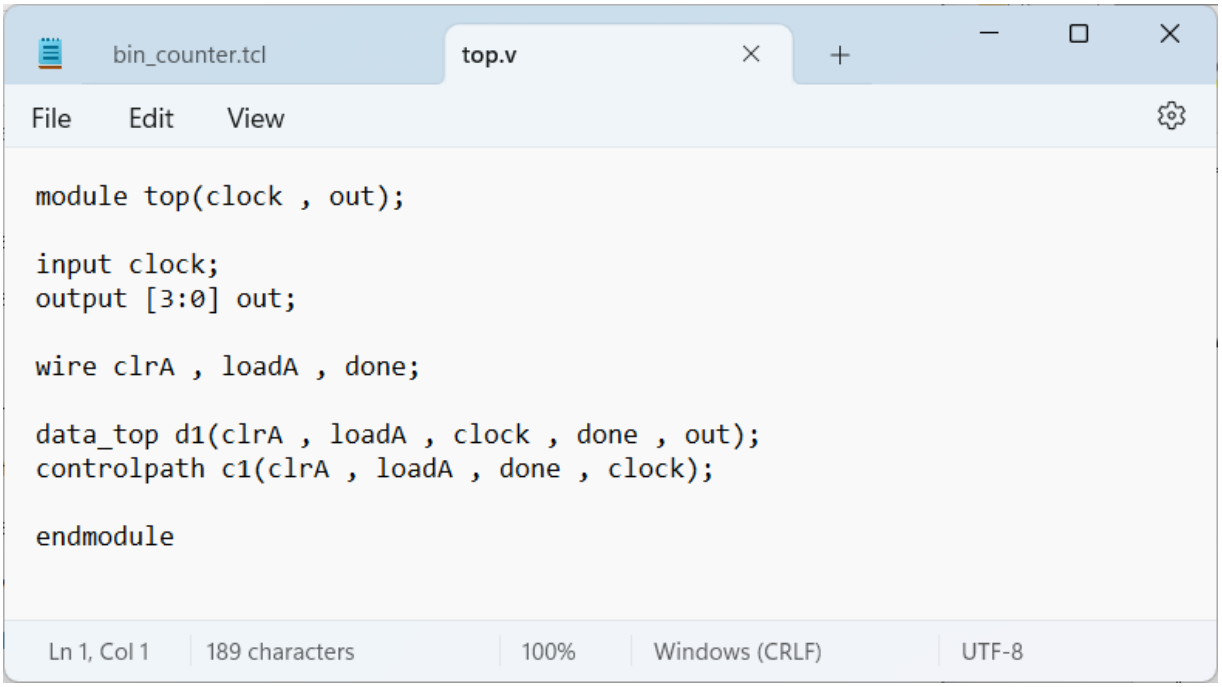
5. Execute the commands in the (.tcl file) by entering **source bin_counter.tcl** command in the command line window.

6. Check for the area, timing and power reports generated in the respective multiplier folder. Also check the gate level netlist generated in the Genus synthesis solution window.

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

Verilog Programs:

//Verilog Program of top-level module containing data path and control path



LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

//Verilog Program of control path module

controlpath.v

File Edit View

```
module controlpath(clrA , loadA , done , clock);

input done , clock;
output reg clrA , loadA;

parameter s0 = 1'b0, s1 = 1'b1;

reg state;

always@(posedge clock)
begin
case(state)
s0: state <= s1;
s1: begin
if(done == 1'b1) state <= s0;
else state <= s1;
end
default: state <= s0;
endcase
end


always@(state)
begin
case(state)
s0: begin clrA = 1'b1; loadA = 1'b0; end
s1: begin clrA = 1'b0; loadA = 1'b1;end
endcase

end
endmodule
```

Ln 13, Col 17 | 439 characters | 100% | Windows (CRLF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

//Verilog Program of datapath

controlpath.v data_top

File Edit View

```
module data_top(clrA , loadA , clock , done , out);

input clrA , clock , loadA;
output done;
output [3:0] out;

wire [3:0] Z;

register r1(Z,out,clrA,loadA,clock);
adder a1(out,Z);
comparator c1(out,done);

endmodule
```

Ln 1, Col 1 219 characters 100% Windows (CRLF) UTF-8

//Verilog Program of adder

adder.v

File Edit View

```
module adder(a,z);

input [3:0] a;
output reg [3:0] z;

always@*
begin
z = a+1;
end
endmodule
```

Ln 1, Col 1 94 characters 100% Windows (CRLF) UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

//Verilog Program of Comparator

comparator.v

File Edit View

```
module comparator(in , done);

input [3:0] in;
output reg done;

always@*
begin
if(in == 8) done = 1'b1;
else done = 1'b0;
end
endmodule
```

Ln 1, Col 1 | 137 characters | 100% | Windows (CRLF) | UTF-8

//Verilog Program of register

comparator.v register.v

File Edit View

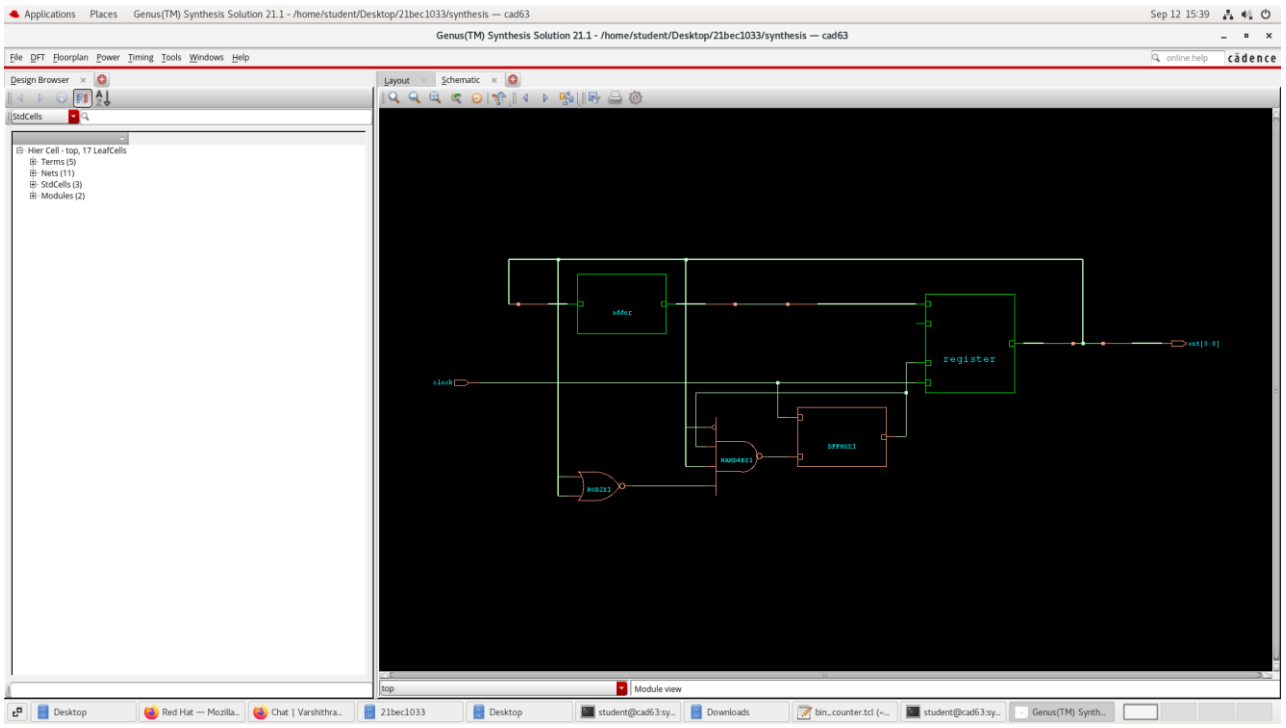
```
module register(in,out,clr,loadA,clock);

input [3:0] in;
output reg [3:0] out;
input clr , clock , loadA;

always@(posedge clock)
begin
if(clr) out <= 4'b0000;
else if (loadA) out <= in;
end
endmodule
```

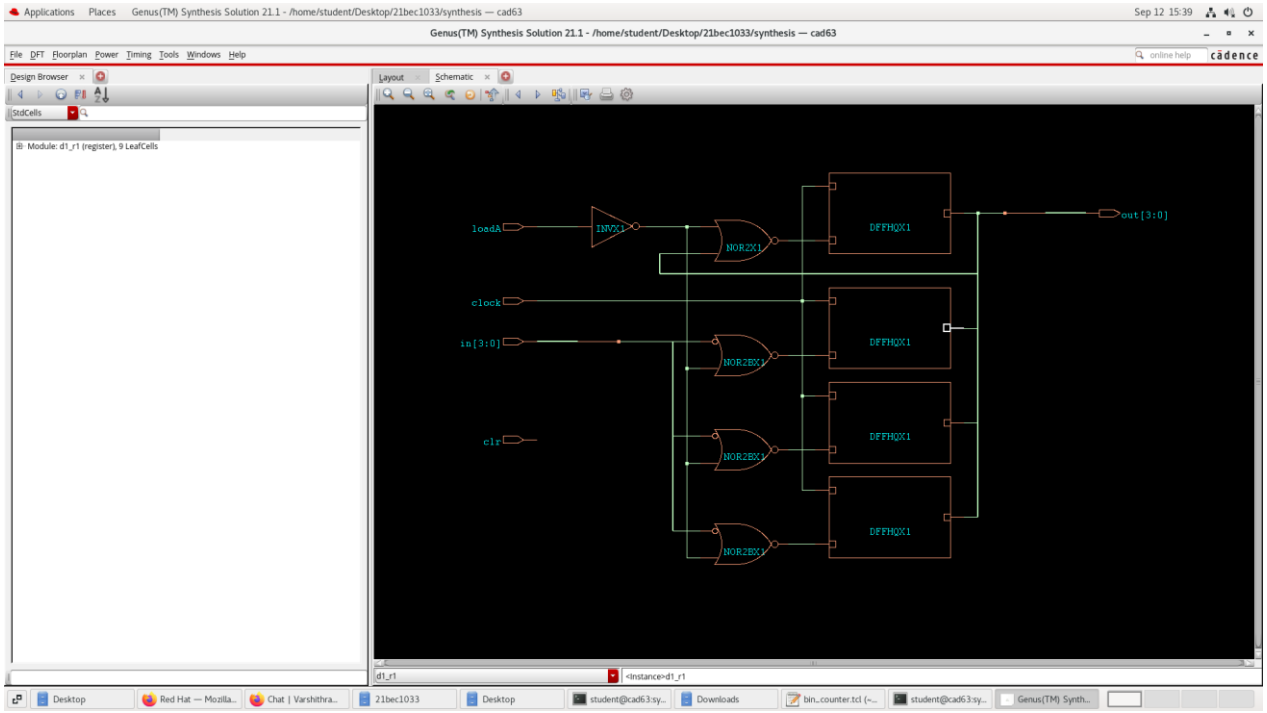
Ln 5, Col 27 | 205 characters | 100% | Windows (CRLF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

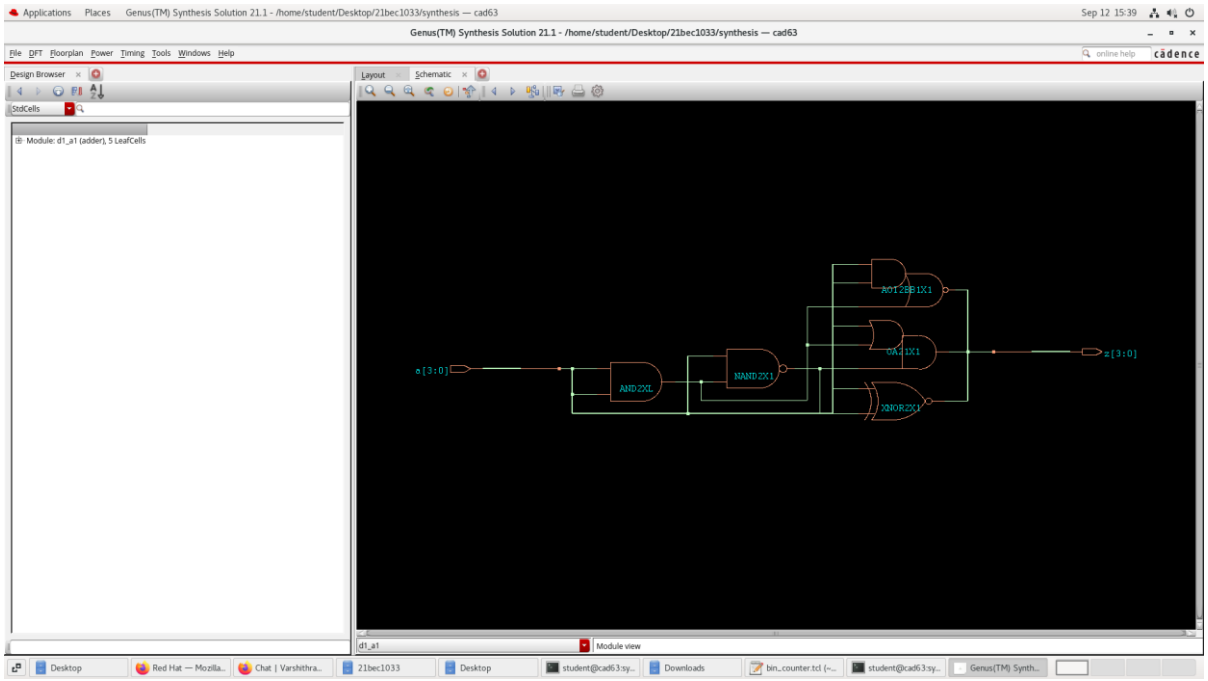
Gate level Netlist:

Synthesis effort: Medium

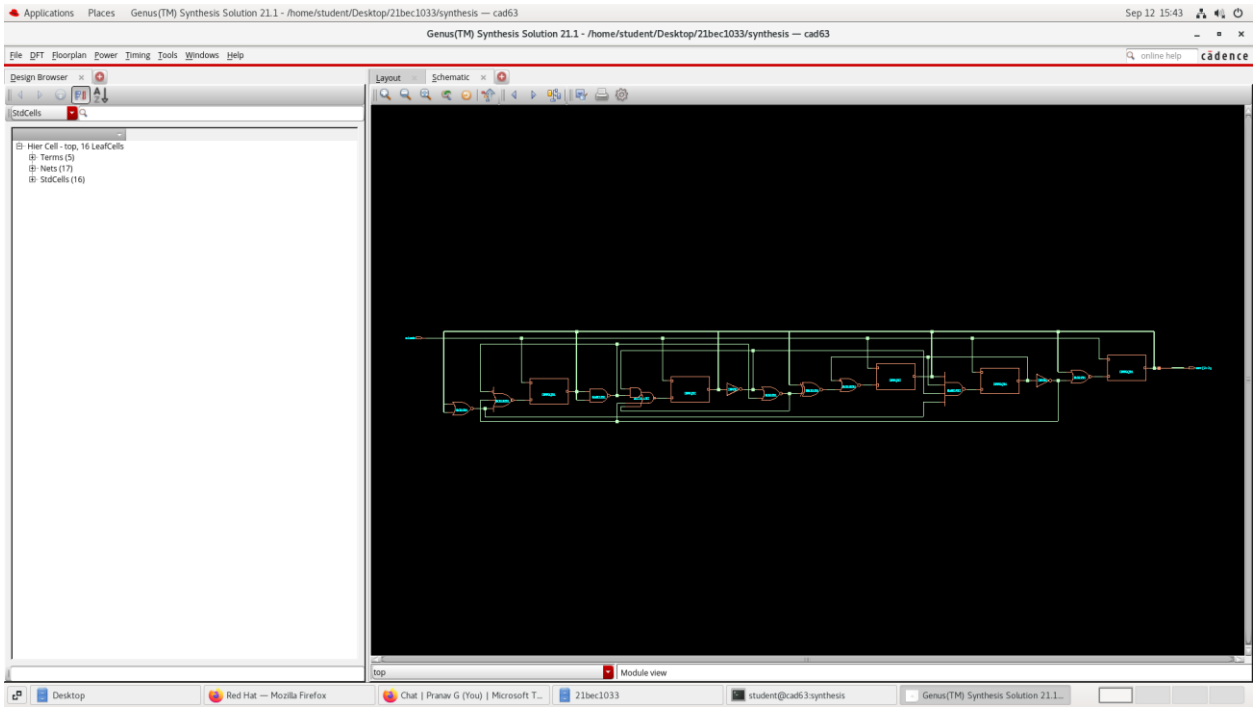




LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic



Synthesis effort : High



LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

Observations:

a) Binary_counter_netlist_created file

bin_counter_netlist.v

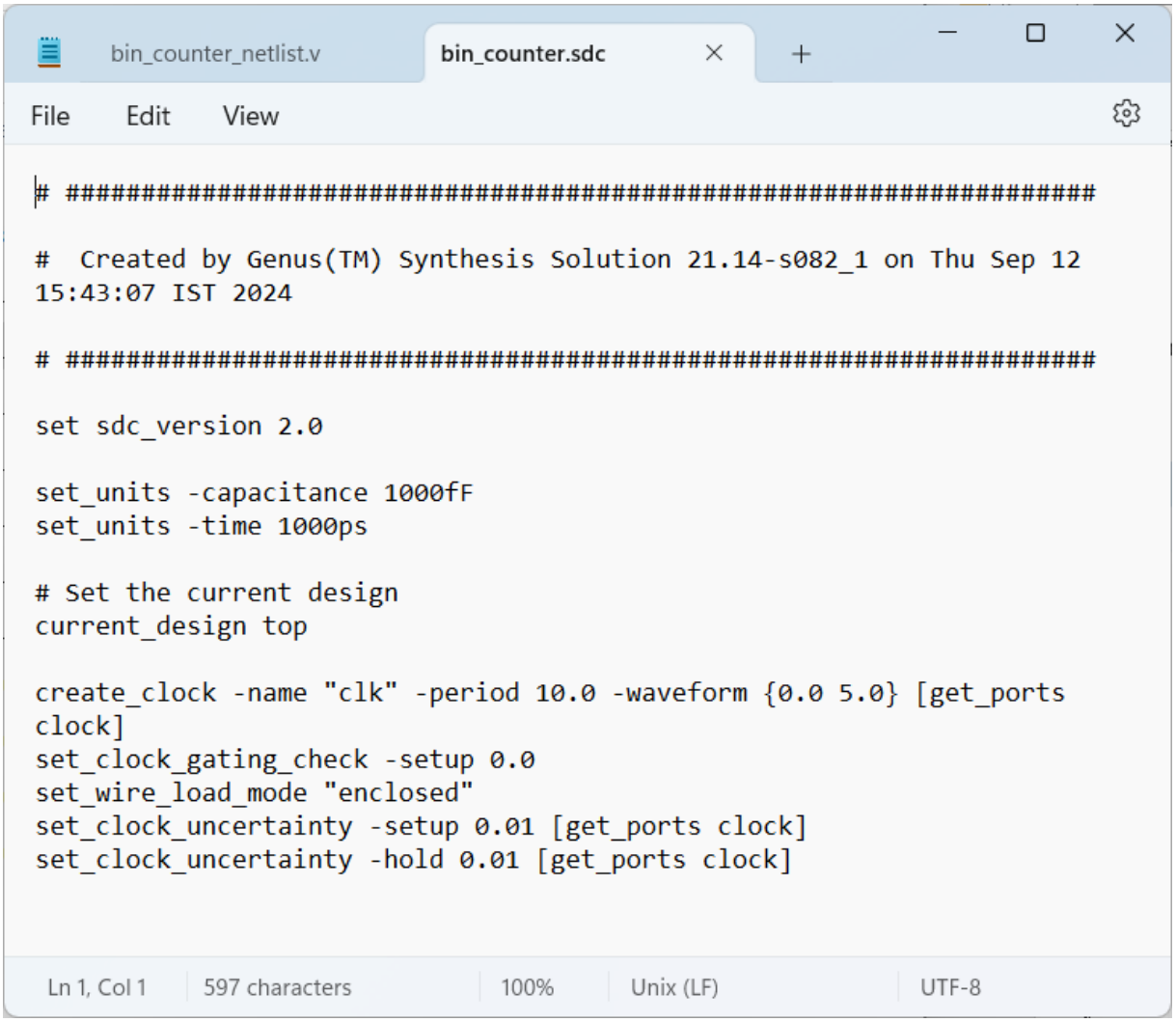
File Edit View

```
// Generated by Cadence Genus(TM) Synthesis Solution 21.14-s082_1
// Generated on: Sep 12 2024 15:43:07 IST (Sep 12 2024 10:13:07 UTC)

// Verification Directory fv/top

module top(clock, out);
  input clock;
  output [3:0] out;
  wire clock;
  wire [3:0] out;
  wire loadA, n_0, n_1, n_2, n_3, n_4, n_5, n_6;
  wire n_7, n_8, n_9, n_10;
  DFFQXL \d1_r1_out_reg[3] (.CK (clock), .D (n_10), .Q (out[3]));
  NOR2BX1 g163__2398(.AN (loadA), .B (n_9), .Y (n_10));
  DFFQXL \d1_r1_out_reg[2] (.CK (clock), .D (n_8), .Q (out[2]));
  XNOR2X1 g165__5107(.A (out[3]), .B (n_7), .Y (n_9));
  AOI211XL g166__6260(.A0 (n_2), .A1 (n_5), .B0 (n_4), .C0 (n_7), .Y
       (n_8));
  DFFHQX1 \d1_r1_out_reg[1] (.CK (clock), .D (n_6), .Q (out[1]));
  DFFHQX1 c1_state_reg(.CK (clock), .D (n_3), .Q (loadA));
  NOR3BX1 g170__4319(.AN (n_5), .B (n_4), .C (n_1), .Y (n_6));
  NAND4XL g169__8428(.A (out[3]), .B (loadA), .C (n_2), .D (n_1), .Y
       (n_3));
  DFFHQX1 \d1_r1_out_reg[0] (.CK (clock), .D (n_0), .Q (out[0]));
  NOR2X1 g172__5526(.A (n_2), .B (n_5), .Y (n_7));
  NOR2X1 g173__6783(.A (out[0]), .B (n_4), .Y (n_0));
  NAND2X1 g175__3680(.A (out[1]), .B (out[0]), .Y (n_5));
  NOR2X1 g174__1617(.A (out[1]), .B (out[0]), .Y (n_1));
  INVX1 g176(.A (out[2]), .Y (n_2));
  INVX1 g177(.A (loadA), .Y (n_4));
endmodule
```

Ln 1, Col 1 | 1,308 characters | 100% | Unix (LF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

b) Binary_counter _constraint_created file



LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

c) Binary_counter_timing_created file

Synthesis effort: Medium

bin_counter_timing.rep

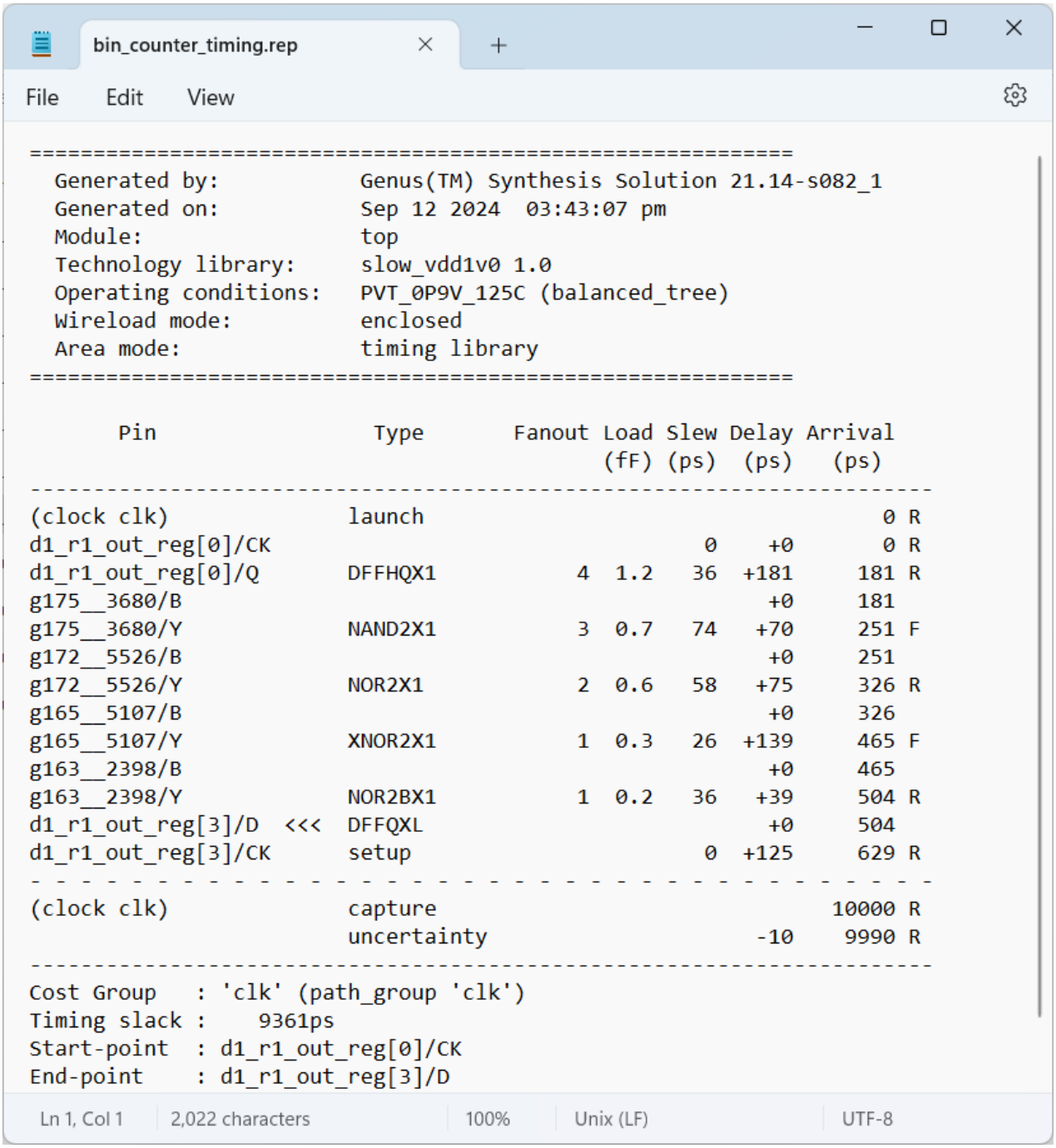
File Edit View

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:39:19 pm
  Module:                 top
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

     Pin                Type       Fanout Load Slew Delay Arrival
                                          (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------
(clock clk)          launch                                    0 R
d1_r1
  out_reg[0]/CK                                    0    +0       0 R
  out_reg[0]/Q       DFFHQX1           5  1.2   36  +181     181 R
d1_r1/out[0]
d1_a1/a[0]
  g86__8428/A                                            +0     181
  g86__8428/Y        AND2XL            3  1.0   49  +130     311 R
  g84__6260/B                                            +0     311
  g84__6260/Y        NAND2X1           2  0.6   70   +74     385 F
  g82__2398/B                                            +0     385
  g82__2398/Y        XNOR2X1           1  0.2   22  +176     561 R
d1_a1/z[3]
d1_r1/in[3]
  g31/AN                                                 +0     561
  g31/Y              NOR2BX1           1  0.3   43   +86     647 R
  out_reg[3]/D  <<<  DFFHQX1                             +0     647
  out_reg[3]/CK      setup                           0  +126     773 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)          capture                                 10000 R
                     uncertainty                        -10   9990 R
-------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    9217ps
Start-point  : d1_r1/out_reg[0]/CK
End-point    : d1_r1/out_reg[3]/D
```

Ln 1, Col 1 | 1,999 characters | 100% | Unix (LF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

Synthesis effort: High



bin_counter_timing.rep

File Edit View

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:43:07 pm
  Module:                 top
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

       Pin                 Type       Fanout Load Slew Delay Arrival
                                             (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------
(clock clk)              launch                                     0 R
d1_r1_out_reg[0]/CK                                    0    +0       0 R
d1_r1_out_reg[0]/Q       DFFHQX1           4  1.2   36  +181     181 R
g175__3680/B                                             +0     181
g175__3680/Y             NAND2X1           3  0.7   74   +70     251 F
g172__5526/B                                             +0     251
g172__5526/Y             NOR2X1            2  0.6   58   +75     326 R
g165__5107/B                                             +0     326
g165__5107/Y             XNOR2X1           1  0.3   26  +139     465 F
g163__2398/B                                             +0     465
g163__2398/Y             NOR2BX1           1  0.2   36   +39     504 R
d1_r1_out_reg[3]/D  <<<  DFFQXL                          +0     504
d1_r1_out_reg[3]/CK      setup                            0  +125     629 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)              capture                                10000 R
                         uncertainty                   -10    9990 R
-------------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :    9361ps
Start-point  : d1_r1_out_reg[0]/CK
End-point    : d1_r1_out_reg[3]/D
```

Ln 1, Col 1 | 2,022 characters | 100% | Unix (LF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

d) Binary_counter_area_created file

Synthesis effort: Medium

bin_counter_timing.rep | bin_counter_area.rep

File Edit View

```
╞============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:39:19 pm
  Module:                 top
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance  Module   Cell Count  Cell Area  Net Area   Total Area   Wireload
------------------------------------------------------------------------------
top                       17     45.486     0.000       45.486 <none> (D)
  d1_a1  adder             5      8.892     0.000        8.892 <none> (D)
  d1_r1  register          9     27.702     0.000       27.702 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 881 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High

bin_counter_area.rep

File Edit View

```
╞============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 12 2024  03:43:07 pm
  Module:                 top
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
------------------------------------------------------------------------------
top                     16     42.408     0.000       42.408 <none> (D)
  (D) = wireload is default in technology library
```

Ln 1, Col 1 | 721 characters | 100% | Unix (LF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

e) Binary_counter_power_created file

Synthesis effort: Medium

bin_counter_power.rep

File Edit View

```
Instance: /top
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ---------------------------------------------------------------------------
      memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register     6.01724e-10  1.07533e-06  2.65163e-08  1.10245e-06  79.10%
       latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic     4.18816e-10  1.73367e-07  3.65876e-08  2.10374e-07  15.09%
        bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock     0.00000e+00  0.00000e+00  8.10000e-08  8.10000e-08   5.81%
         pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     1.02054e-09  1.24870e-06  1.44104e-07  1.39383e-06 100.00%
  Percentage           0.07%       89.59%       10.34%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,197 characters | 100% | Unix (LF) | UTF-8

Synthesis effort: High

bin_counter_area.rep | bin_counter_power.rep

File Edit View

```
Instance: /top
Power Unit: W
PDB Frames: /stim#0/frame#0
  ---------------------------------------------------------------------------
    Category         Leakage     Internal    Switching        Total     Row%
  ---------------------------------------------------------------------------
      memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
    register     5.27962e-10  9.85531e-07  2.27229e-08  1.00878e-06  81.83%
       latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       logic     3.40689e-10  1.05315e-07  3.72653e-08  1.42921e-07  11.59%
        bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       clock     0.00000e+00  0.00000e+00  8.10000e-08  8.10000e-08   6.57%
         pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
          pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  ---------------------------------------------------------------------------
    Subtotal     8.68651e-10  1.09085e-06  1.40988e-07  1.23270e-06  99.99%
  Percentage           0.07%       88.49%       11.44%      100.00% 100.00%
  ---------------------------------------------------------------------------
```

Ln 1, Col 1 | 1,197 characters | 100% | Unix (LF) | UTF-8

LAB4_4: Logic synthesis of a Binary Counter using Datapath and Control path logic

Inference:

Synthesis effort: Medium

A total of 17 leaf instance count is present in the gate level netlist with total area of 45.486, total power of 1.39383e-06 W.

Synthesis effort: High

A total of 16 leaf instance count is present in the gate level netlist with total area of 42.408 , total power of 1.23270e-06 W.

Result: Hence a Binary_counter using Datapath and Control path logic is synthesized and the gate level netlist with timing, area and power report has been generated.