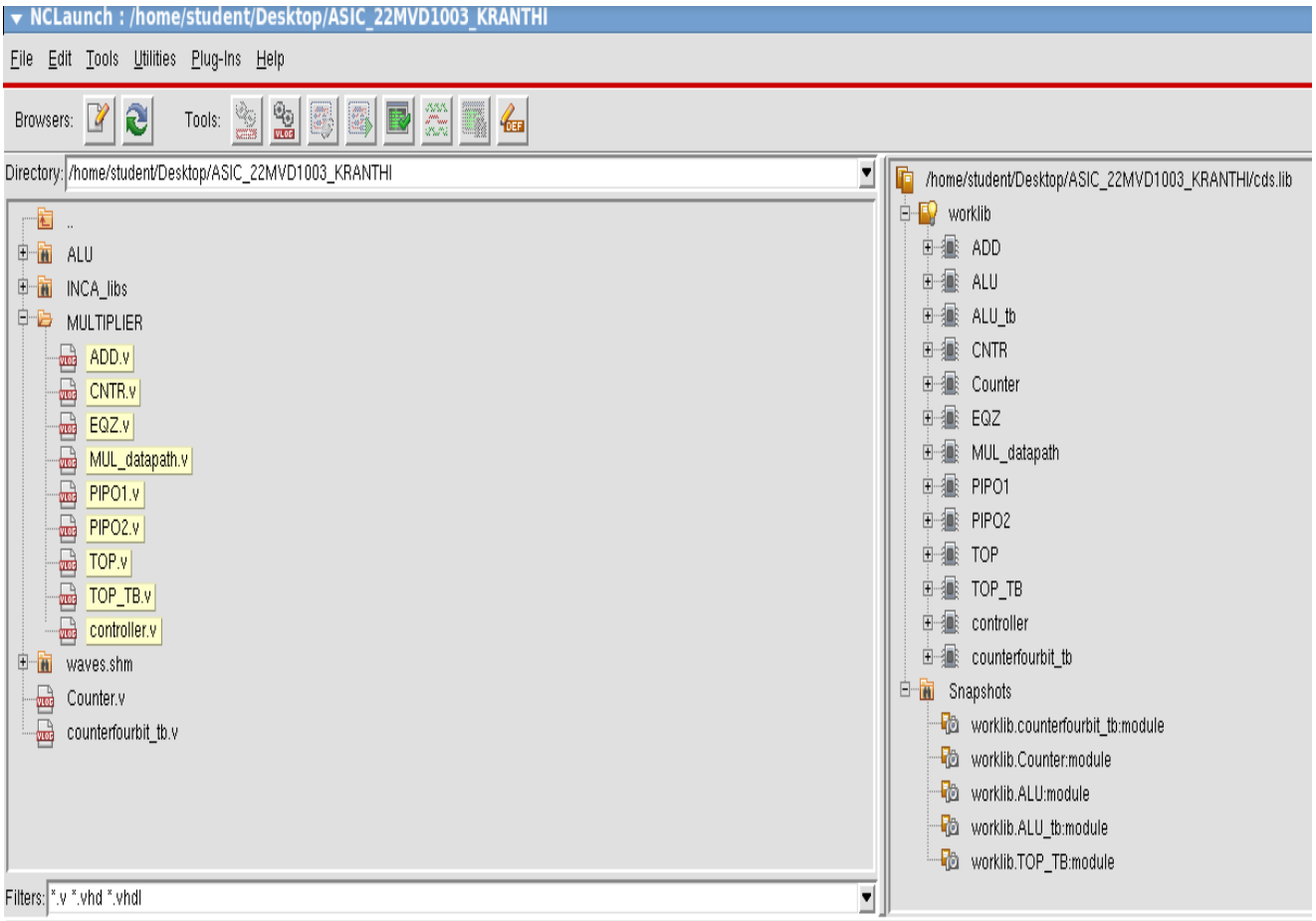
Experiment3: Design and simulation of a multiplier by repeated addition using Datapath and Control path logic

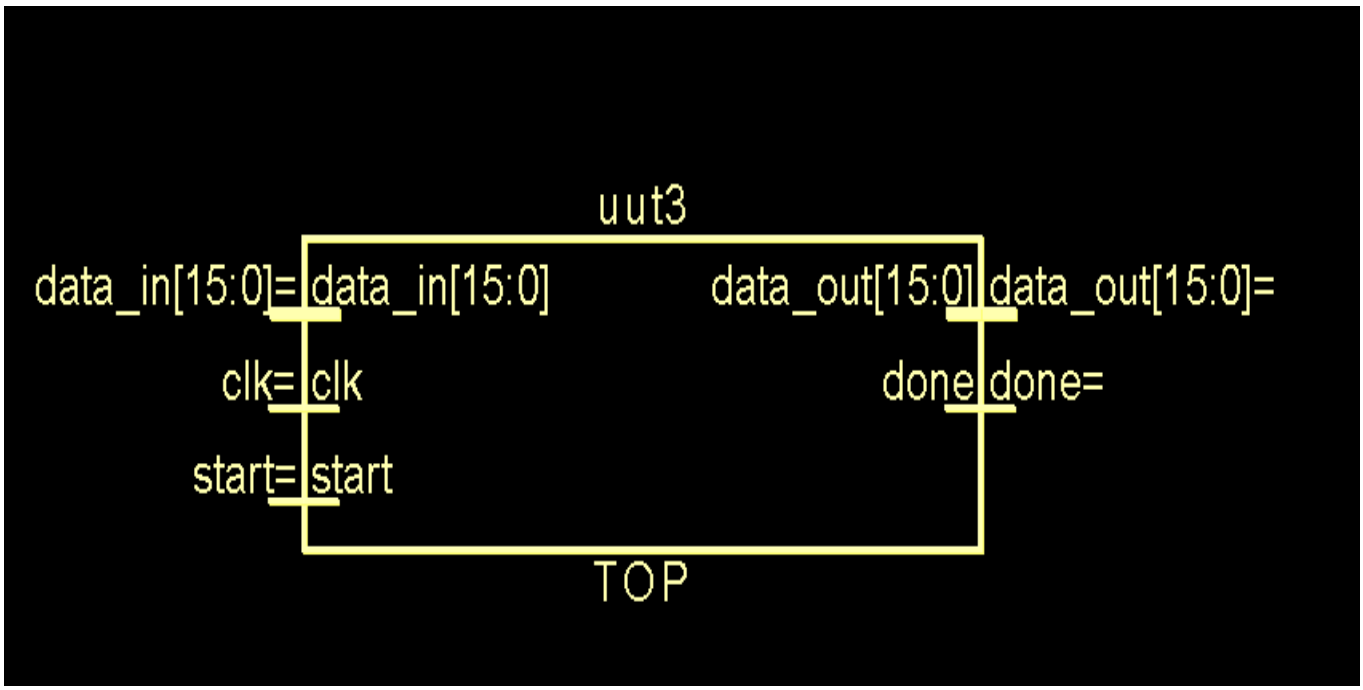
NCLaunch Compilation and Elaboration:

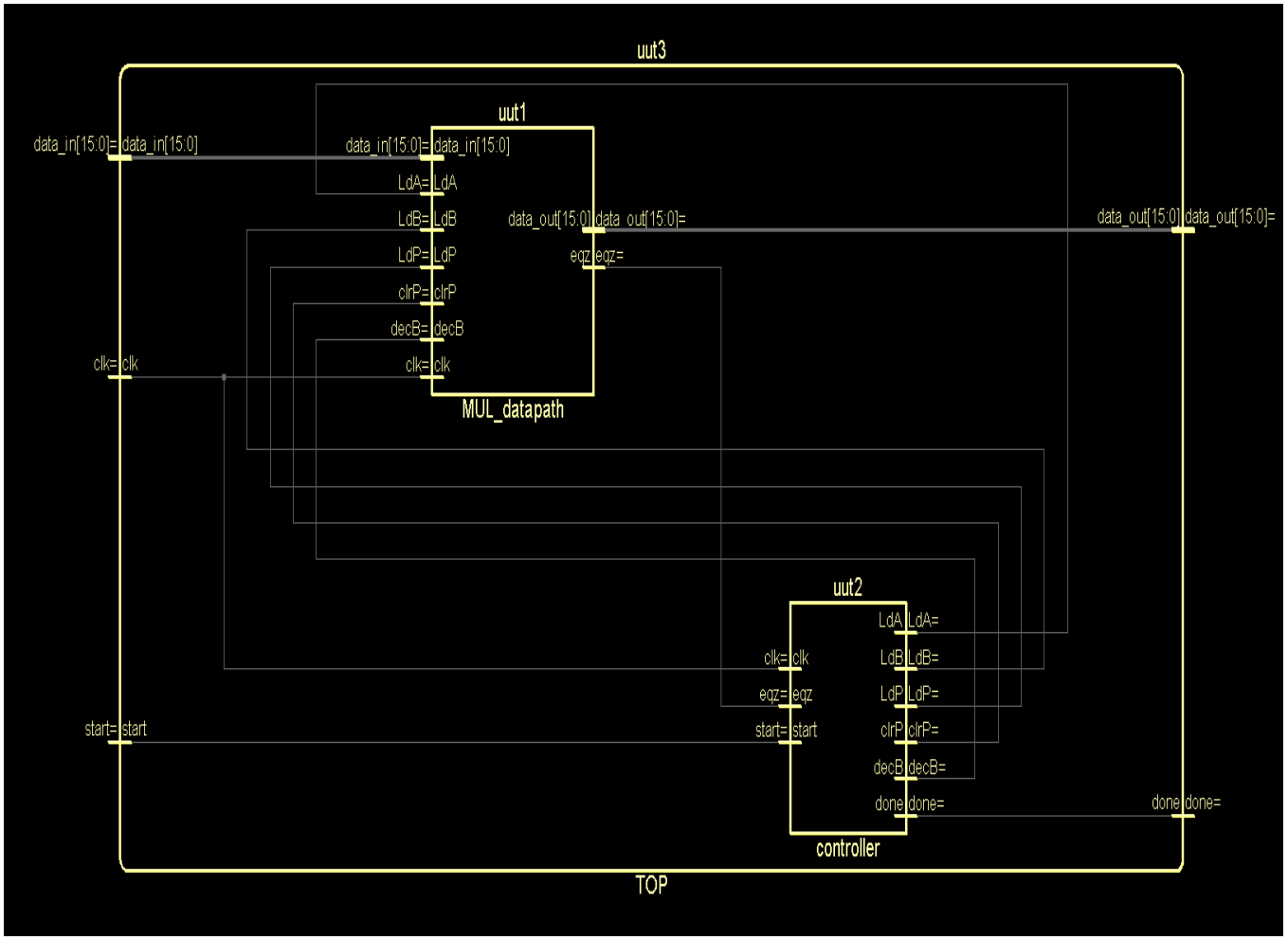



```
simvision(64): 15.20-s045: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
...Regained control from SimVision----------------------------------
ncsim: Memory Usage - 38.6M program + 158.0M data = 196.6M total (200.2M Peak)
ncsim: CPU Usage - 8.1s system + 14.6s user = 22.7s total (335.6s, 6.8% cpu)
simvision(64): 15.20-s045: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
...Regained control from SimVision
----------------------------------------
ncsim: Memory Usage - 38.6M program + 158.3M data = 196.8M total (203.6M Peak)
ncsim: CPU Usage - 3.1s system + 3.2s user = 6.4s total (154.1s, 4.1% cpu)
ncvlog -work worklib -cdslib /home/student/Desktop/ASIC_22MVD1003_KRANTHI/cds.li\
b -logfile ncvlog.log -errormax 15 -update -linedebug -status /home/student/Desk\
top/ASIC_22MVD1003_KRANTHI/MULTIPLIER/ADD.v /home/student/Desktop/ASIC_22MVD1003\
_KRANTHI/MULTIPLIER/CNTR.v /home/student/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLI\
ER/EQZ.v /home/student/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER/MUL_datapath.v \
/home/student/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER/PIPO1.v /home/student/De\
sktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER/PIPO2.v /home/student/Desktop/ASIC_22MVD\
1003_KRANTHI/MULTIPLIER/TOP.v /home/student/Desktop/ASIC_22MVD1003_KRANTHI/MULTI\
PLIER/TOP_TB.v /home/student/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER/controlle\
r.v
ncvlog(64): 15.20-s045: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
ncvlog: Memory Usage - 21.3M program + 27.9M data = 49.2M total
ncvlog: CPU Usage - 0.0s system + 0.0s user = 0.0s total (0.2s, 19.1% cpu)
nclaunch> ncelab -work worklib -cdslib /home/student/Desktop/ASIC_22MVD1003_KRANTHI/cds.lib -logfile ncelab.log -errormax 15 -access +wc -status worklib.TOP_TB
ncelab(64): 15.20-s045: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
ncelab: Memory Usage - 49.8M program + 31.4M data = 81.2M total (Peak 86.1M)
ncelab: CPU Usage - 0.0s system + 0.0s user = 0.0s total (0.4s, 9.8% cpu)
nclaunch> ncsim -gui -cdslib /home/student/Desktop/ASIC_22MVD1003_KRANTHI/cds.lib -logfile ncsim.log -errormax 15 -status worklib.TOP_TB:module
nclaunch> ncsim(64): 15.20-s045: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
----------------------------------------
Relinquished control to SimVision...
ncsim> ncsim> txe(64): 15.20-s045: (c) Copyright 1995-2023 Cadence Design Systems, Inc.
```

Experiment3: Design and simulation of a multiplier by repeated addition using Datapath and Control path logic

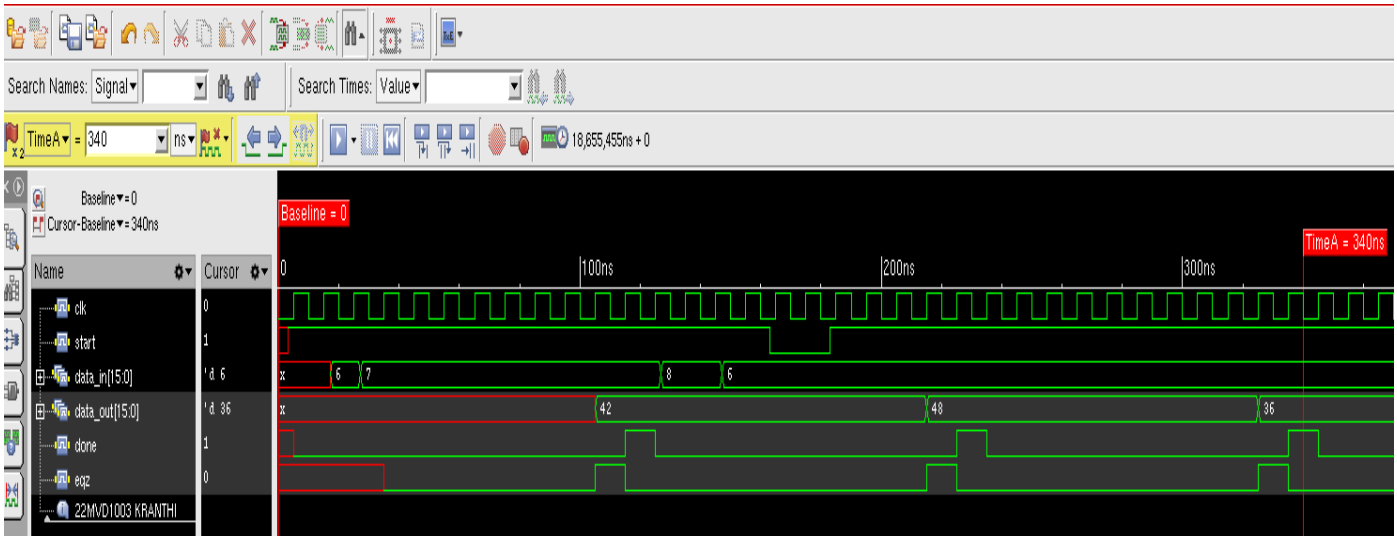
RTL Schematic Window:





Experiment3: 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 6 & 7 values and the data_out is obtained as 42 which is the multiplied result after repeated addition. Then the data_in is changed to 8 & 6 and the corresponding data_out is obtained as 48 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.

Experiment5: 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 (Logic synthesis tool)

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 (.SDC file) by entering the design constraints required for the synthesis.

Multiplier_constraint.sdc

```
create_clock -name CLK_100M -period 10 -waveform {0 5} [get_ports "clk"]

set_clock_transition -rise 0.1 [ get_clocks "CLK_100M" ]

set_clock_transition -fall 0.1 [ get_clocks "CLK_100M" ]

set_clock_uncertainty 0.01 [ get_ports "clk" ]


set_input_delay -max 1.0 [get_ports "clk"] -clock CLK_100M

set_input_delay -max 1.0 [get_ports "start"] -clock CLK_100M

set_input_delay -max 1.0 [get_ports "data_in"] -clock CLK_100M

set_output_delay -max 1.0 [get_ports "done"] -clock CLK_100M

set_output_delay -max 1.0 [get_ports "data_out"] -clock CLK_100M
```

Experiment5: 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.

*multiplier.tcl (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

*multiplier.tcl

```
set_attr library slow.lib
read_hdl { TOP.v ADD.v CNTR.v controller.v EQZ.v MUL_datapath.v PIPO1.v PIPO2.v }

elaborate

read_sdc Multiplier_constraint.sdc
synthesize -to_mapped -effort high
write_hdl > multiplier_netlist.v

write_sdc > multiplier_constraint_created.sdc
write_sdf > multiplier_sdf_created.sdf

gui_show

report_timing > multiplier_timing_created.rep
report_area > multiplier_area_created.rep
report_power > multiplier_power_created.rep
report_gates > multiplier_gates_created.rep
report_qor > multiplier_quality_created.rep
```

4. Invoke the C shell and launch the Genus tool by entering the below commands.

student@cad200:MULTIPLIER

File Edit View Search Terminal Help

```
[student@cad200 MULTIPLIER]$ csh
[student@cad200 MULTIPLIER]$ source /cad/cshrc


Welcome to Cadence Tools Suite

[student@cad200 MULTIPLIER]$ genus -legacy_ui
```

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.

Experiment5: 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

File: /home/student/Desktop/ASIC_22...D1003_KRANTHI/MULTIPLIER/TOP.v

```
module TOP(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
```

//Verilog Program of control path module

File: /home/student/Desktop/ASIC_22...RANTHI/MULTIPLIER/controller.v

```
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:  if (eqz) state <= S4;
S4: if(done) state <= S0;
default: state <= S0;
endcase
end

always@(state)
begin
case(state)
S0:begin LdA=1'b0; LdB=1'b0; LdP=1'b0; clrP=1'b0; decB=1'b0; done=1'b0; end
S1:begin LdA=1'b1; LdB=1'b0; LdP=1'b0; clrP=1'b0; decB=1'b0; done=1'b0; end
S2:begin LdA=1'b0; LdB=1'b1; LdP=1'b0; clrP=1'b1; decB=1'b0; done=1'b0; end
S3:begin LdA=1'b0; LdB=1'b0; LdP=1'b1; clrP=1'b0; decB=1'b1; done=1'b0; end
S4:begin LdA=1'b0; LdB=1'b0; LdP=1'b0; clrP=1'b0; decB=1'b0; done=1'b1; end
default: begin LdA=1'b0; LdB=1'b0; LdP=1'b0; clrP=1'b0; decB=1'b0; done=1'b0; end
endcase
end
endmodule
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

//Verilog Program of data path module

File: /home/student/Desktop/ASIC_22...NTHI/MULTIPLIER/MUL_datapath.v

```
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
```

//Verilog Program of PIPO1

File: /home/student/Desktop/ASIC_22...003_KRANTHI/MULTIPLIER/PIPO1.v

```
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
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

//Verilog Program of PIPO2

File: /home/student/Desktop/ASIC_22...003_KRANTHI/MULTIPLIER/PIPO2.v

```
module PIPO2 (dout, din, ld, clr, clk);
input [15:0] din;
input ld, clr, clk;
output reg [15:0] dout;
always @(posedge clk)
if (clr) dout <= 16'd0;
else if (ld) dout <= din;
endmodule
```

//Verilog Program of Adder

File: /home/student/Desktop/ASIC_22...D1003_KRANTHI/MULTIPLIER/ADD.v

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

//Verilog Program of Counter

File: /home/student/Desktop/ASIC_22...1003_KRANTHI/MULTIPLIER/CNTR.v

```
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
```

//Verilog Program of Comparator

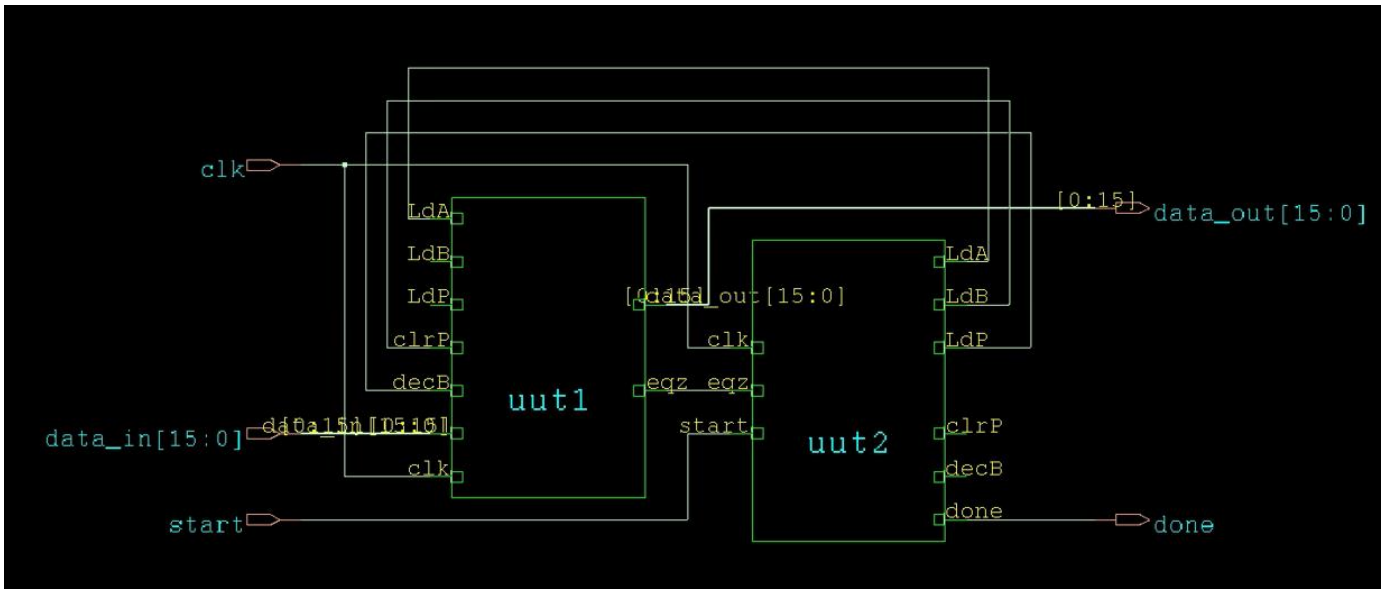
File: /home/student/Desktop/ASIC_22...D1003_KRANTHI/MULTIPLIER/EQZ.v

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

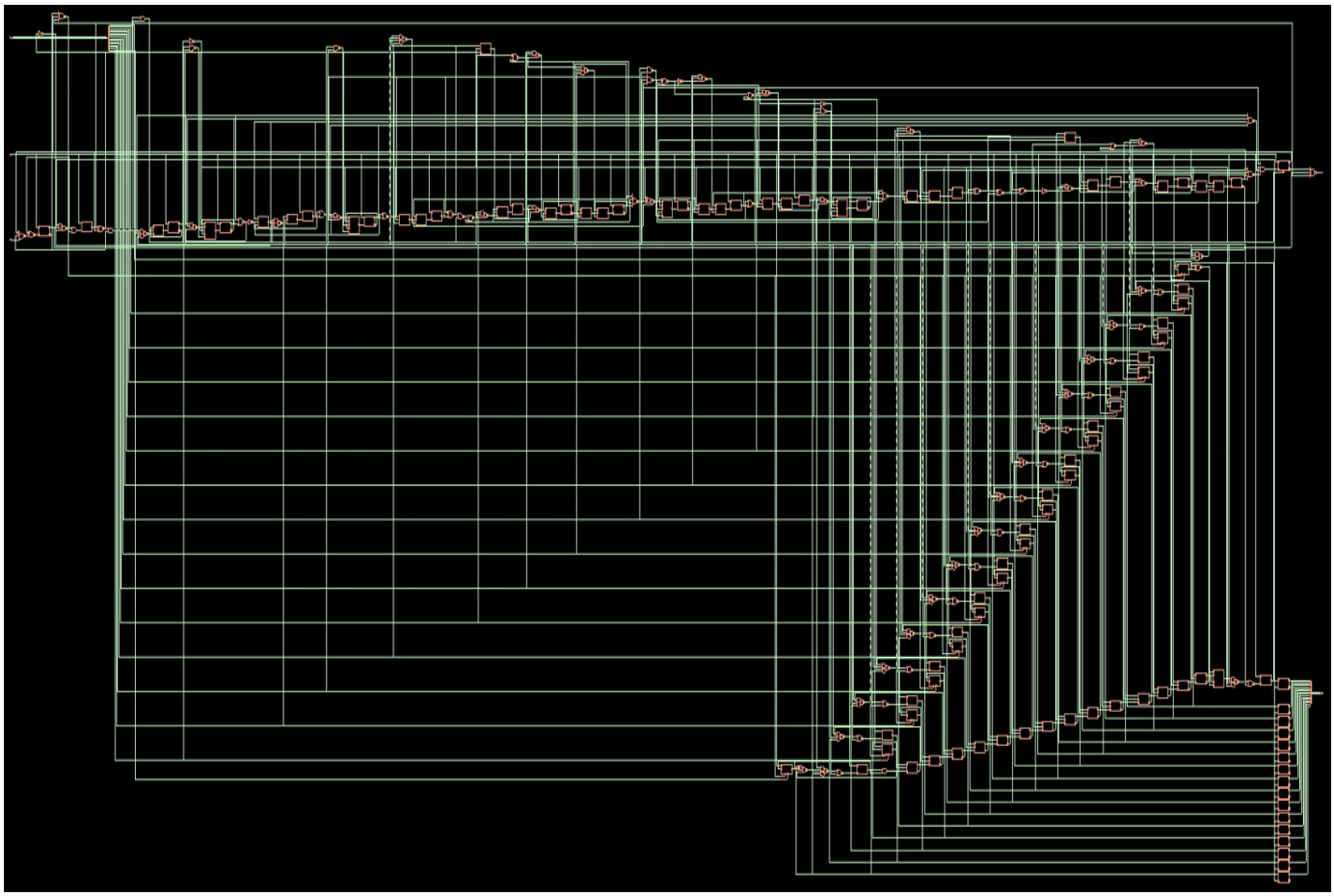
Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Gate level Netlist:

Synthesis effort: Medium



Synthesis effort: High



Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Observations:

a) multiplier_netlist_created file

multiplier_netlist.v

```
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 [15:0] data_out;
  output eqz;
  wire LdA, LdB, LdP, clrP, decB, clk;
  wire [15:0] data_in;
  wire [15:0] data_out;
  wire eqz;
  wire [15:0] X;
  wire [15:0] Y;
  wire [15:0] Z;
  wire [15:0] Bout;
  wire UNCONNECTED, UNCONNECTED0, UNCONNECTED1, UNCONNECTED2,
       UNCONNECTED3, UNCONNECTED4, UNCONNECTED5, UNCONNECTED6;
  wire UNCONNECTED7, UNCONNECTED8, UNCONNECTED9, UNCONNECTED10,
       UNCONNECTED11, UNCONNECTED12, UNCONNECTED13, UNCONNECTED14;
  PIPO1 A(X, data_in, LdA, clk);
  ADD AD(Z, X, Y);
  CNTR B(Bout, data_in, clrP, decB, clk);
  EQZ COMP(eqz, Bout);
  PIPO2 P(Y, Z, decB, clrP, clk);
  TLATX1 \data_out_reg[11] (.G (eqz), .D (Y[11]), .Q (data_out[11]),
       .QN (UNCONNECTED));
  TLATX1 \data_out_reg[14] (.G (eqz), .D (Y[14]), .Q (data_out[14]),
       .QN (UNCONNECTED0));
  TLATX1 \data_out_reg[3] (.G (eqz), .D (Y[3]), .Q (data_out[3]), .QN
       (UNCONNECTED1));
  TLATX1 \data_out_reg[12] (.G (eqz), .D (Y[12]), .Q (data_out[12]),
       .QN (UNCONNECTED2));
  TLATX1 \data_out_reg[15] (.G (eqz), .D (Y[15]), .Q (data_out[15]),
       .QN (UNCONNECTED3));
  TLATX1 \data_out_reg[10] (.G (eqz), .D (Y[10]), .Q (data_out[10]),
       .QN (UNCONNECTED4));
  TLATX1 \data_out_reg[9] (.G (eqz), .D (Y[9]), .Q (data_out[9]), .QN
       (UNCONNECTED5));
  TLATX1 \data_out_reg[8] (.G (eqz), .D (Y[8]), .Q (data_out[8]), .QN
       (UNCONNECTED6));
  TLATX1 \data_out_reg[6] (.G (eqz), .D (Y[6]), .Q (data_out[6]), .QN
       (UNCONNECTED7));
  TLATX1 \data_out_reg[5] (.G (eqz), .D (Y[5]), .Q (data_out[5]), .QN
       (UNCONNECTED8));
  TLATX1 \data_out_reg[4] (.G (eqz), .D (Y[4]), .Q (data_out[4]), .QN
       (UNCONNECTED9));
  TLATX1 \data_out_reg[13] (.G (eqz), .D (Y[13]), .Q (data_out[13]),
       .QN (UNCONNECTED10));
  TLATX1 \data_out_reg[2] (.G (eqz), .D (Y[2]), .Q (data_out[2]), .QN
       (UNCONNECTED11));
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

b) multiplier_constraint_created file

```
Checking for analog nets...
Check completed for analog nets.
Checking for source RTL...
Check completed for source RTL.
Statistics for commands executed by read_sdc:
 "create_clock"             - successful      1 , failed      0 (runtime  0.00)
 "get_clocks"               - successful      2 , failed      0 (runtime  0.00)
 "get_ports"                - successful      7 , failed      0 (runtime  0.00)
 "set_clock_transition"     - successful      2 , failed      0 (runtime  0.00)
 "set_clock_uncertainty"    - successful      1 , failed      0 (runtime  0.00)
 "set_input_delay"          - successful      3 , failed      0 (runtime  0.00)
 "set_output_delay"         - successful      2 , failed      0 (runtime  0.00)
```

multiplier_constraint_created.sdc

```
# Set the current design
current_design TOP

create_clock -name "CLK_100M" -period 10.0 -waveform {0.0 5.0} [get_ports clk]
set_clock_transition 0.1 [get_clocks CLK_100M]
set_clock_gating_check -setup 0.0
set_input_delay -clock [get_clocks CLK_100M] -network_latency_included -add_delay -rise -min 0.0 [get_ports clk]
set_input_delay -clock [get_clocks CLK_100M] -clock_fall -network_latency_included -add_delay -fall -min 0.0 [get_ports clk]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports clk]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports start]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[15]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[14]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[13]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[12]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[11]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[10]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[9]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[8]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[7]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[6]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[5]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[4]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[3]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[2]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[1]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_in[0]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports done]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[15]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[14]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[13]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[12]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[11]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[10]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[9]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[8]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[7]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[6]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[5]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[4]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[3]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[2]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[1]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_out[0]}]
set_wire_load_mode "enclosed"
set_dont_use [get_lib_cells tsmc18/RF1R1WX2]
set_dont_use [get_lib_cells tsmc18/RF2R1WX2]
set_dont_use [get_lib_cells tsmc18/RFRDX1]
set_dont_use [get_lib_cells tsmc18/RFRDX2]
set_dont_use [get_lib_cells tsmc18/RFRDX4]
set_dont_use [get_lib_cells tsmc18/TIEHI]
set_dont_use [get_lib_cells tsmc18/TIELO]
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

c) multiplier_sdf_created file

Synthesis effort: Medium

multiplier_sdf_created.sdf

```
(DELAYFILE
  (SDFVERSION  "OVI 3.0")
  (DESIGN      "TOP")
  (DATE        "Tue Apr 18 16:13:34 IST 2023")
  (VENDOR      "Cadence, Inc.")
  (PROGRAM     "Genus(TM) Synthesis Solution")
  (VERSION     "17.21-s010_1")
  (DIVIDER     .)
  (VOLTAGE     ::1.62)
  (PROCESS     "::1.0")
  (TEMPERATURE ::125.0)
  (TIMESCALE   1ps)
  (CELL
     (CELLTYPE "SDFFHQX1")
     (INSTANCE uut1.A.dout_reg\[9\])
     (DELAY
        (ABSOLUTE
          (PORT CK (::0.0))
          (PORT D (::0.0))
          (PORT SI (::0.0))
          (PORT SE (::0.0))
          (IOPATH CK Q (::374) (::317))
        )
     )
     (TIMINGCHECK
        (SETUPHOLD (negedge D) (posedge CK) (::737) (::-455))
        (SETUPHOLD (posedge D) (posedge CK) (::244) (::-161))
        (SETUPHOLD (negedge SI) (posedge CK) (::702) (::-416))
        (SETUPHOLD (posedge SI) (posedge CK) (::263) (::-179))
        (SETUPHOLD (negedge SE) (posedge CK) (::771) (::-334))
        (SETUPHOLD (posedge SE) (posedge CK) (::777) (::-274))
        (WIDTH (posedge CK) (::177.122))
        (WIDTH (negedge CK) (::337.238))
     )
  )
  (CELL
     (CELLTYPE "SDFFHQX1")
     (INSTANCE uut1.A.dout_reg\[1\])
     (DELAY
        (ABSOLUTE
          (PORT CK (::0.0))
          (PORT D (::0.0))
          (PORT SI (::0.0))
          (PORT SE (::0.0))
          (IOPATH CK Q (::374) (::317))
        )
```

Synthesis effort: High

multiplier_sdf_created.sdf

```
(DELAYFILE
  (SDFVERSION  "OVI 3.0")
  (DESIGN      "TOP")
  (DATE        "Tue Apr 18 15:51:48 IST 2023")
  (VENDOR      "Cadence, Inc.")
  (PROGRAM     "Genus(TM) Synthesis Solution")
  (VERSION     "17.21-s010_1")
  (DIVIDER     .)
  (VOLTAGE     ::1.62)
  (PROCESS     "::1.0")
  (TEMPERATURE ::125.0)
  (TIMESCALE   1ps)
  (CELL
     (CELLTYPE "AND4X2")
     (INSTANCE g344)
     (DELAY
        (ABSOLUTE
          (PORT A (::0.0))
          (PORT B (::0.0))
          (PORT C (::0.0))
          (PORT D (::0.0))
          (IOPATH D Y (::235) (::231))
          (IOPATH A Y (::240) (::188))
          (IOPATH B Y (::241) (::207))
          (IOPATH C Y (::231) (::217))
        )
     )
  )
  (CELL
     (CELLTYPE "NOR3X1")
     (INSTANCE g345)
     (DELAY
        (ABSOLUTE
          (PORT A (::0.0))
          (PORT B (::0.0))
          (PORT C (::0.0))
          (IOPATH A Y (::89) (::52))
          (IOPATH B Y (::148) (::79))
          (IOPATH C Y (::166) (::82))
        )
     )
  )
  (CELL
     (CELLTYPE "NOR4X1")
     (INSTANCE g346)
     (DELAY
        (ABSOLUTE
          (PORT A (::0.0))
          (PORT B (::0.0))
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

d) multiplier_timing_created file

Synthesis effort: Medium

*multiplier_timing_created.rep

Pin	Type	Fanout	Load (fF)	Slew (ps)	Delay (ps)	Arrival (ps)	
(clock CLK_100M)	launch					0	R
uut1							
P							
dout_reg[0]/CK				100		0	R
dout_reg[0]/Q	DFFHQX1	4	14.3	158	+342	342	F
P/dout[0]							
AD/in2[0]							
add_5_11/B[0]							
g383/A					+0	342	
g383/Y	AND2X2	2	9.3	74	+184	527	F
g380/CI					+0	527	
g380/CO	ADDFX1	1	6.2	142	+312	839	F
g379/CI					+0	839	
g379/CO	ADDFX1	1	6.2	142	+330	1169	F
g378/CI					+0	1169	
g378/CO	ADDFX1	1	6.2	142	+330	1499	F
g377/CI					+0	1499	
g377/CO	ADDFX1	1	6.2	142	+330	1829	F
g376/CI					+0	1829	
g376/CO	ADDFX1	1	6.2	142	+330	2159	F
g375/CI					+0	2159	
g375/CO	ADDFX1	1	6.2	142	+330	2489	F
g374/CI					+0	2489	
g374/CO	ADDFX1	1	6.2	142	+330	2819	F
g373/CI					+0	2819	
g373/CO	ADDFX1	1	6.2	142	+330	3149	F
g372/CI					+0	3149	
g372/CO	ADDFX1	1	6.2	142	+330	3479	F
g371/CI					+0	3479	
g371/CO	ADDFX1	1	6.2	142	+330	3809	F
g370/CI					+0	3809	
g370/CO	ADDFX1	1	6.2	142	+330	4139	F
g369/CI					+0	4139	
g369/CO	ADDFX1	1	6.2	142	+330	4469	F
g368/CI					+0	4469	
g368/CO	ADDFX1	1	6.2	142	+330	4799	F
g367/CI					+0	4799	
g367/CO	ADDFX1	2	6.8	146	+334	5133	F
g366/A1N					+0	5133	
g366/Y	OAI2BB2X1	1	4.3	114	+220	5353	F
add_5_11/Z[15]							
AD/out[15]							
P/din[15]							
g80/B0					+0	5353	
g80/Y	AOI22X1	1	4.3	269	+208	5561	R
g64/A					+0	5561	
g64/Y	NOR2X1	1	2.1	330	+63	5625	F
dout_reg[15]/D <<<	DFFHQX1				+0	5625	

Synthesis effort: High

multiplier_timing_created.rep

Area mode: timing library

Pin		Type	Fanout	Load (fF)	Slew (ps)	Delay (ps)	Arrival (ps)	
(clock CLK_100M)		launch					0	R
uut1_P_dout_reg[0]/CK						100	0	R
uut1_P_dout_reg[0]/Q		DFFHQX1	4	13.9	155	+340	340	F
uut1_AD_add_5_11_g383/A						+0	340	
uut1_AD_add_5_11_g383/Y		AND2X2	2	9.3	74	+184	524	F
uut1_AD_add_5_11_g380/CI						+0	524	
uut1_AD_add_5_11_g380/CO		ADDFX1	1	6.2	142	+312	836	F
uut1_AD_add_5_11_g379/CI						+0	836	
uut1_AD_add_5_11_g379/CO		ADDFX1	1	6.2	142	+330	1166	F
uut1_AD_add_5_11_g378/CI						+0	1166	
uut1_AD_add_5_11_g378/CO		ADDFX1	1	6.2	142	+330	1496	F
uut1_AD_add_5_11_g377/CI						+0	1496	
uut1_AD_add_5_11_g377/CO		ADDFX1	1	6.2	142	+330	1826	F
uut1_AD_add_5_11_g376/CI						+0	1826	
uut1_AD_add_5_11_g376/CO		ADDFX1	1	6.2	142	+330	2156	F
uut1_AD_add_5_11_g375/CI						+0	2156	
uut1_AD_add_5_11_g375/CO		ADDFX1	1	6.2	142	+330	2486	F
uut1_AD_add_5_11_g374/CI						+0	2486	
uut1_AD_add_5_11_g374/CO		ADDFX1	1	6.2	142	+330	2816	F
uut1_AD_add_5_11_g373/CI						+0	2816	
uut1_AD_add_5_11_g373/CO		ADDFX1	1	6.2	142	+330	3146	F
uut1_AD_add_5_11_g372/CI						+0	3146	
uut1_AD_add_5_11_g372/CO		ADDFX1	1	6.2	142	+330	3476	F
uut1_AD_add_5_11_g371/CI						+0	3476	
uut1_AD_add_5_11_g371/CO		ADDFX1	1	6.2	142	+330	3806	F
uut1_AD_add_5_11_g370/CI						+0	3806	
uut1_AD_add_5_11_g370/CO		ADDFX1	1	6.2	142	+330	4136	F
uut1_AD_add_5_11_g369/CI						+0	4136	
uut1_AD_add_5_11_g369/CO		ADDFX1	1	6.2	142	+330	4466	F
uut1_AD_add_5_11_g368/CI						+0	4466	
uut1_AD_add_5_11_g368/CO		ADDFX1	1	6.2	142	+330	4796	F
uut1_AD_add_5_11_g367/CI						+0	4796	
uut1_AD_add_5_11_g367/CO		ADDFX1	2	6.8	146	+334	5130	F
uut1_AD_add_5_11_g366/A1N						+0	5130	
uut1_AD_add_5_11_g366/Y		OAI2BB2X1	1	4.7	117	+222	5352	F
g1930/A0						+0	5352	
g1930/Y		AOI22X1	1	4.3	261	+154	5507	R
g1910/A						+0	5507	
g1910/Y		NOR2X1	1	2.1	146	+63	5570	F
uut1_P_dout_reg[15]/D	<<<	DFFHQX1				+0	5570	
uut1_P_dout_reg[15]/CK		setup				100 +335	5905	R

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

e) multiplier_area_created file

Synthesis effort: Medium

multiplier_area_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

multiplier_area_created.rep

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  04:13:35 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

   Instance         Module      Cell Count  Cell Area  Net Area   Total Area  Wireload
-------------------------------------------------------------------------------------------
TOP                                   165   5934.298     0.000     5934.298    <none> (D)
  uut1         MUL_datapath           152   5628.269     0.000     5628.269    <none> (D)
    B          CNTR                    48   1536.797     0.000     1536.797    <none> (D)
    P          PIPO2                   49   1283.990     0.000     1283.990    <none> (D)
    A          PIPO1                   16   1064.448     0.000     1064.448    <none> (D)
    AD         ADD                     18   1057.795     0.000     1057.795    <none> (D)
      add_5_11 add_unsigned_32         18   1057.795     0.000     1057.795    <none> (D)
    COMP       EQZ                      5     99.792     0.000       99.792    <none> (D)
  uut2         controller              13    306.029     0.000      306.029    <none> (D)

 (D) = wireload is default in technology library
```

Synthesis effort: High

multiplier_area_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

multiplier_area_created.rep

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  03:51:48 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area  Wireload
-----------------------------------------------------------------------------
TOP                    199   6180.451     0.000     6180.451    <none> (D)

 (D) = wireload is default in technology library
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

f) multiplier_power_created file

Synthesis effort: Medium

multiplier_power_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

multiplier_power_created.rep

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  04:13:35 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================

                     Leakage    Dynamic     Total
   Instance    Cells Power(nW)  Power(nW)  Power(nW)
--------------------------------------------------------
TOP              165   207.618 820126.048 820333.666
  uut1           152   199.213 745981.474 746180.687
    B             48    54.633 208685.256 208739.889
    AD            18    54.372 128620.841 128675.213
      add_5_11    18    54.372 128620.841 128675.213
    P             49    37.784 218508.493 218546.277
    A             16    34.503 170797.032 170831.535
    COMP           5     5.600   6262.500   6268.100
  uut2            13     8.404  32402.330  32410.735
```

Synthesis effort: High

multiplier_power_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

multiplier_power_created.rep

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  03:51:48 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================

                Leakage    Dynamic     Total
Instance Cells Power(nW)  Power(nW)  Power(nW)
------------------------------------------------
TOP        199   200.103 833564.627 833764.730
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

g) multiplier_gates_created file

Synthesis effort: Medium

multiplier_gates_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - gedit

File Edit View Search Tools Documents Help

multiplier_gates_created.rep

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  04:13:35 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


   Gate    Instances    Area    Library
-----------------------------------------
ADDFX1            14   977.962    tsmc18
AND2X2             1    13.306    tsmc18
AND3X2             1    16.632    tsmc18
AOI21X1            1    13.306    tsmc18
AOI22X1           16   266.112    tsmc18
AOI2BB1XL          1    16.632    tsmc18
DFFHQX1           16   851.558    tsmc18
DFFTRX1            2   113.098    tsmc18
DFFX1              1    56.549    tsmc18
INVX1              2    13.306    tsmc18
NAND2BX1           2    26.611    tsmc18
NAND2X1            2    19.958    tsmc18
NOR2X1            15   149.688    tsmc18
NOR2XL             3    29.938    tsmc18
NOR3X1             1    13.306    tsmc18
NOR3XL             1    13.306    tsmc18
NOR4X1             1    19.958    tsmc18
OAI21XL            1    13.306    tsmc18
OAI2BB1XL         14   232.848    tsmc18
OAI2BB2X1          1    23.285    tsmc18
OAI2BB2XL          2    46.570    tsmc18
OR2X2             14   186.278    tsmc18
OR4X2              4    79.834    tsmc18
SDFFHQX1          32  2128.896    tsmc18
TLATX1            16   585.446    tsmc18
XNOR2X1            1    26.611    tsmc18
-----------------------------------------
total            165  5934.298


     Type      Instances   Area   Area %
-----------------------------------------
sequential            67 3735.547   62.9
inverter               2   13.306    0.2
logic                 96 2185.445   36.8
physical_cells         0    0.000    0.0
-----------------------------------------
total                165 5934.298  100.0
```

Synthesis effort: High

multiplier_gates_created.rep

```
  Area mode:              timing library
============================================================


   Gate     Instances     Area     Library
-----------------------------------------------
ADDFX1              14   977.962     tsmc18
AND2X2               3    39.917     tsmc18
AND4X2               1    19.958     tsmc18
AOI21X1             11   146.362     tsmc18
AOI21XL              2    26.611     tsmc18
AOI22X1             19   316.008     tsmc18
AOI22XL              2    33.264     tsmc18
AOI2BB1X1            1    16.632     tsmc18
AOI2BB1XL            1    16.632     tsmc18
DFFHQX1             24  1277.338     tsmc18
DFFTRX1              1    56.549     tsmc18
DFFX1               10   565.488     tsmc18
INVX1                2    13.306     tsmc18
INVXL                4    26.611     tsmc18
NAND2BX1             2    26.611     tsmc18
NAND2BXL             1    13.306     tsmc18
NAND2X1              1     9.979     tsmc18
NAND2XL              4    39.917     tsmc18
NAND3BX1             1    16.632     tsmc18
NAND3X1              1    13.306     tsmc18
NOR2BX1              4    53.222     tsmc18
NOR2BXL              1    13.306     tsmc18
NOR2X1              19   189.605     tsmc18
NOR2XL               3    29.938     tsmc18
NOR3X1               2    26.611     tsmc18
NOR4X1               4    79.834     tsmc18
OAI21XL             10   133.056     tsmc18
OAI221XL             4    93.139     tsmc18
OAI2BB1X1            4    66.528     tsmc18
OAI2BB1XL            7   116.424     tsmc18
OAI2BB2X1            1    23.285     tsmc18
OR2X2                2    26.611     tsmc18
SDFFHQX1            16  1064.448     tsmc18
TLATX1              16   585.446     tsmc18
XNOR2X1              1    26.611     tsmc18
-----------------------------------------------
total              199  6180.451


     Type       Instances    Area    Area %
-----------------------------------------------
sequential              67 3549.269    57.4
inverter                 6   39.917     0.6
logic                  126 2591.266    41.9
physical_cells           0    0.000     0.0
-----------------------------------------------
total                  199 6180.451   100.0
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

h) multiplier_quality_created file

Synthesis effort: Medium

multiplier_quality_created.rep

```
================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  04:13:35 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
================================================================

Timing
--------

  Clock    Period
-----------------
CLK_100M 10000.0


  Cost      Critical          Violating
  Group   Path Slack  TNS      Paths
-----------------------------------------
CLK_100M      3998.0   0.0           0
default     No paths   0.0
-----------------------------------------
Total                  0.0           0

Instance Count
--------------
Leaf Instance Count              165
Physical Instance count            0
Sequential Instance Count         67
Combinational Instance Count      98
Hierarchical Instance Count        8

Area
----
Cell Area                              5934.298
Physical Cell Area                     0.000
Total Cell Area (Cell+Physical)        5934.298
Net Area                               0.000
Total Area (Cell+Physical+Net)         5934.298

Max Fanout                             51 (clk)
Min Fanout                             1 (done)
Average Fanout                         2.5
Terms to net ratio                     3.3700
Terms to instance ratio                4.0848
Runtime                                16.685785 seconds
Elapsed Runtime                        211 seconds
Genus peak memory usage                711.45
Innovus peak memory usage              no_value
Hostname                               cad200
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Synthesis effort: High

multiplier_quality_created.rep (~/Desktop/ASIC_22MVD1003_KRANTHI/MULTIPLIER) - ged

File Edit View Search Tools Documents Help

multiplier_quality_created.rep

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           Apr 18 2023  03:51:48 pm
  Module:                 TOP
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Timing
--------

  Clock   Period
-----------------
CLK_100M 10000.0


  Cost     Critical         Violating
  Group   Path Slack  TNS     Paths
-------------------------------------
CLK_100M      4084.8   0.0         0
default     No paths   0.0
-------------------------------------
Total                  0.0         0

Instance Count
--------------
Leaf Instance Count             199
Physical Instance count           0
Sequential Instance Count        67
Combinational Instance Count    132
Hierarchical Instance Count       0

Area
----
Cell Area                         6180.451
Physical Cell Area                0.000
Total Cell Area (Cell+Physical)   6180.451
Net Area                          0.000
Total Area (Cell+Physical+Net)    6180.451

Max Fanout                        51 (clk)
Min Fanout                        1 (uut1_B_n_154)
Average Fanout                    2.3
Terms to net ratio                3.2438
Terms to instance ratio           3.9447
Runtime                           12.548062999999999 seconds
Elapsed Runtime                   53 seconds
Genus peak memory usage           701.72
Innovus peak memory usage         no_value
Hostname                          cad200
```

Experiment5: Logic synthesis of a multiplier by repeated addition using Datapath and Control path logic

Inference:

Synthesis effort: Medium

A total of 165 leaf instance count (98 Combinational and 67 Sequential) is present in the gate level netlist with total area of 5934.298, total power of 820333.666nW.

Synthesis effort: High

A total of 199 leaf instance count (132 Combinational and 67 Sequential) is present in the gate level netlist with total area of 6180.451, total power of 833764.730nW.

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.