Lab-5: Logic synthesis of an adding CPU

Aim: To synthesize an adding CPU and to get the gate level netlist with timing, area, and power reports.

EDA Tools Used: Cadence Genus (Logic synthesis tool)

Description: An Adding CPU reads Load, Add, Store, and Jump instructions from its memory, and depending on the instruction it reads, loads data, performs addition, stores data into memory, or jumps to another memory location. The Adding CPU has a main register called AC (accumulator). The Load instruction directs the machine to load the addressed data from the memory into AC. The Store instruction causes contents of AC to be written into the addressed location in the memory. The operand of the Add instruction is immd (immediate). This instruction adds immd to the present contents of AC and puts the result back into AC. The Jump instruction loads the 6-bit address into the program counter of our machine, causing the next instruction to be fetched from this address.

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.

Lab-5: Logic synthesis of an adding CPU

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
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 "reset"] -clock CLK_100M

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

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


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

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

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


```

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

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
set_attr library slow.lib
read_hdl { AddingCPU.v DataPath.v Controller.v IR.v PC.v AC.v ALU.v }

elaborate

read_sdc addingCPU.sdc
synthesize -to_mapped -effort medium
write_hdl > addingCPU_netlist.v

write_sdc > addingCPU_constraint_created.sdc
write_sdf > addingCPU_sdf_created.sdf

gui_show

report_timing > addingCPU_timing_created.rep
report_area > addingCPU_area_created.rep
report_power > addingCPU_power_created.rep
report_gates > addingCPU_gates_created.rep
report_qor > addingCPU_quality_created.rep

```

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

Lab-5: Logic synthesis of an adding CPU

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

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

Verilog Programs:

//Verilog Program of top module of adding CPU

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
1 module AddingCPU(input reset,clk, output [5:0]adr_bus,output rd_mem, wr_mem, inout [7:0]data_bus);
2
3 wire ir_on_adr, pc_on_adr,dbus_on_data,data_on_dbus,ld_ir,ld_ac,ld_pc,inc_pc,clr_pc,pass,add,alu_on_dbus;
4 wire [1:0]op_code;
5
6 Controller cu (reset,clk,op_code,rd_mem,wr_mem,ir_on_adr,pc_on_adr,dbus_on_data,data_on_dbus,ld_ir,
7 ld_ac,ld_pc,inc_pc,clr_pc,pass,add,alu_on_dbus);
8
9 DataPath dp (ir_on_adr,pc_on_adr,dbus_on_data,data_on_dbus,ld_ir,ld_ac,ld_pc,inc_pc,clr_pc,pass,add,
10 alu_on_dbus,clk,adr_bus,op_code,data_bus);
11
12 endmodule
```

Lab-5: Logic synthesis of an adding CPU

//Verilog Program of the Controller

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
`define Reset 2'b00
`define Fetch 2'b01
`define Decode 2'b10
`define Execute 2'b11
module Controller ( input reset, clk, input [1:0] op_code,
                    output reg rd_mem, wr_mem, ir_on_adr, pc_on_adr, dbus_on_data, data_on_dbus, ld_ir, ld_ac, ld_pc, inc_pc, clr_pc, pass, add, alu_on_dbus);

reg [1:0] present_state, next_state;
always @( posedge clk )
  if( reset ) present_state <= `Reset;
  else        present_state <= next_state;

always @( present_state or reset ) begin : Combinational
 rd_mem=1'b0; wr_mem=1'b0; ir_on_adr=1'b0; pc_on_adr=1'b0;
 dbus_on_data=1'b0; data_on_dbus=1'b0; ld_ir=1'b0;
 ld_ac=1'b0; ld_pc=1'b0; inc_pc=1'b0; clr_pc=1'b0;
 pass=1'b0; add=1'b0; alu_on_dbus=1'b0;

 case ( present_state )

   `Reset : begin next_state = reset ? `Reset : `Fetch; clr_pc = 1'b1;
            end // End `Reset
   `Fetch : begin next_state = `Decode; pc_on_adr = 1'b1; rd_mem = 1'b1; data_on_dbus = 1'b1; ld_ir = 1'b1; inc_pc = 1'b1;
            end // End `Fetch
   `Decode : next_state = `Execute; // End `Decode
   `Execute: begin next_state = `Fetch;

    case( op_code )

         2'b00: begin ir_on_adr = 1'b1; rd_mem = 1'b1; data_on_dbus = 1'b1; ld_ac = 1'b1;
                end
         2'b01: begin pass = 1'b1; ir_on_adr = 1'b1; alu_on_dbus = 1'b1; dbus_on_data = 1'b1; wr_mem = 1'b1;
                end
         2'b10: ld_pc = 1'b1;
         2'b11: begin add = 1'b1; alu_on_dbus = 1'b1; ld_ac = 1'b1;
                end
     endcase
     end // End `Execute
    default : next_state = `Reset;
  endcase
 end
endmodule
```

//Verilog Program of DataPath

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
module DataPath ( input ir_on_adr, pc_on_adr, dbus_on_data,
                        data_on_dbus, ld_ir, ld_ac, ld_pc,
                        inc_pc, clr_pc, pass, add, alu_on_dbus,clk,
                  output [5:0] adr_bus,
                  output [1:0] op_code,
                  inout [7:0] data_bus );
wire [7:0] dbus, ir_out, a_side, alu_out;
wire [5:0] pc_out;
IR ir ( dbus, ld_ir, clk, ir_out );
PC pc ( ir_out[5:0], ld_pc, inc_pc, clr_pc, clk, pc_out );
AC ac ( dbus, ld_ac, clk, a_side );
ALU alu ( a_side, {2'b00,ir_out[5:0]}, pass, add, alu_out );

assign adr_bus = ir_on_adr ? ir_out[5:0] : 6'bzz_zzzz;
assign adr_bus = pc_on_adr ? pc_out : 6'bzz_zzzz;
assign dbus = alu_on_dbus ? alu_out : 8'bzzzz_zzzz;
assign data_bus = dbus_on_data ? dbus : 8'bzzzz_zzzz;
assign dbus = data_on_dbus ? data_bus : 8'bzzzz_zzzz;
assign op_code = ir_out[7:6];
endmodule
```

Lab-5: Logic synthesis of an adding CPU

//Verilog Program of IR

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
1 module IR(data_in,load,clk,data_out);
2 input [7:0] data_in;
3 input clk,load;
4 output reg [7:0]data_out;
5 always@(posedge clk)
6 begin
7 if(load)
8 begin
9 data_out<=data_in;
10 end
11 endmodule
```

//Verilog Program of PC

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
1 module PC(data_in,load,clk,clr,inc,data_out);
2 input [5:0] data_in;
3 input load,clk,clr,inc;
4 output reg [5:0]data_out;
5 always@(posedge clk)
6 begin
7 if(clr) data_out<=6'b000_000;
8 else if (load) data_out<=data_in;
9 else if (inc) data_out<=data_out +1;
10 end
11 endmodule
```

//Verilog Program of AC

File Edit View Search Tools Documents Help

AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

```
1 module AC(input [7:0] data_in, input load,clk,output reg[7:0] data_out);
2 always@(posedge clk)
3 if(load) data_out <= data_in;
4 endmodule
```

Lab-5: Logic synthesis of an adding CPU

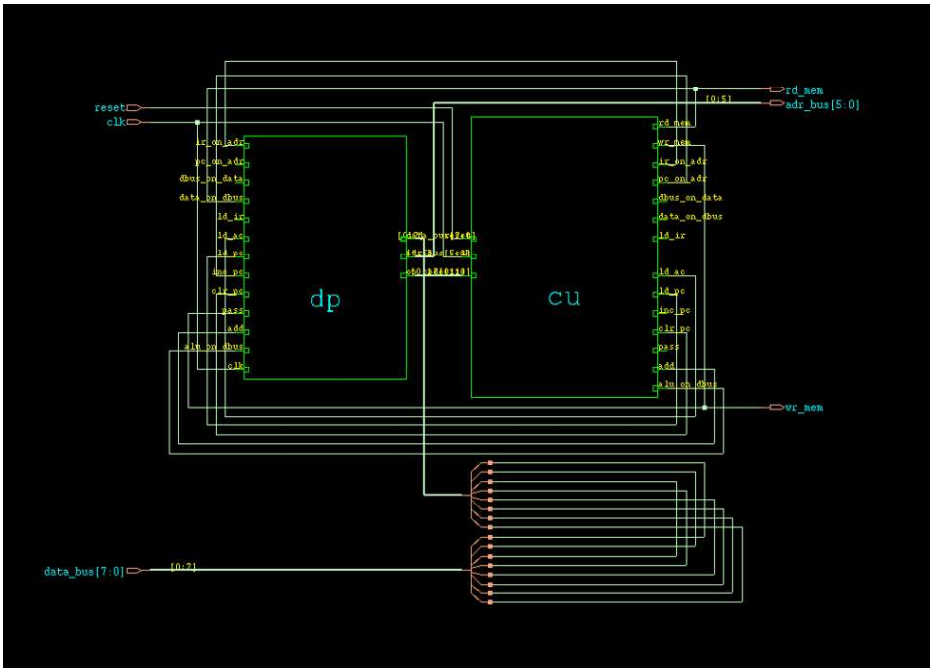
//Verilog Program of the ALU

File Edit View Search Tools Documents Help

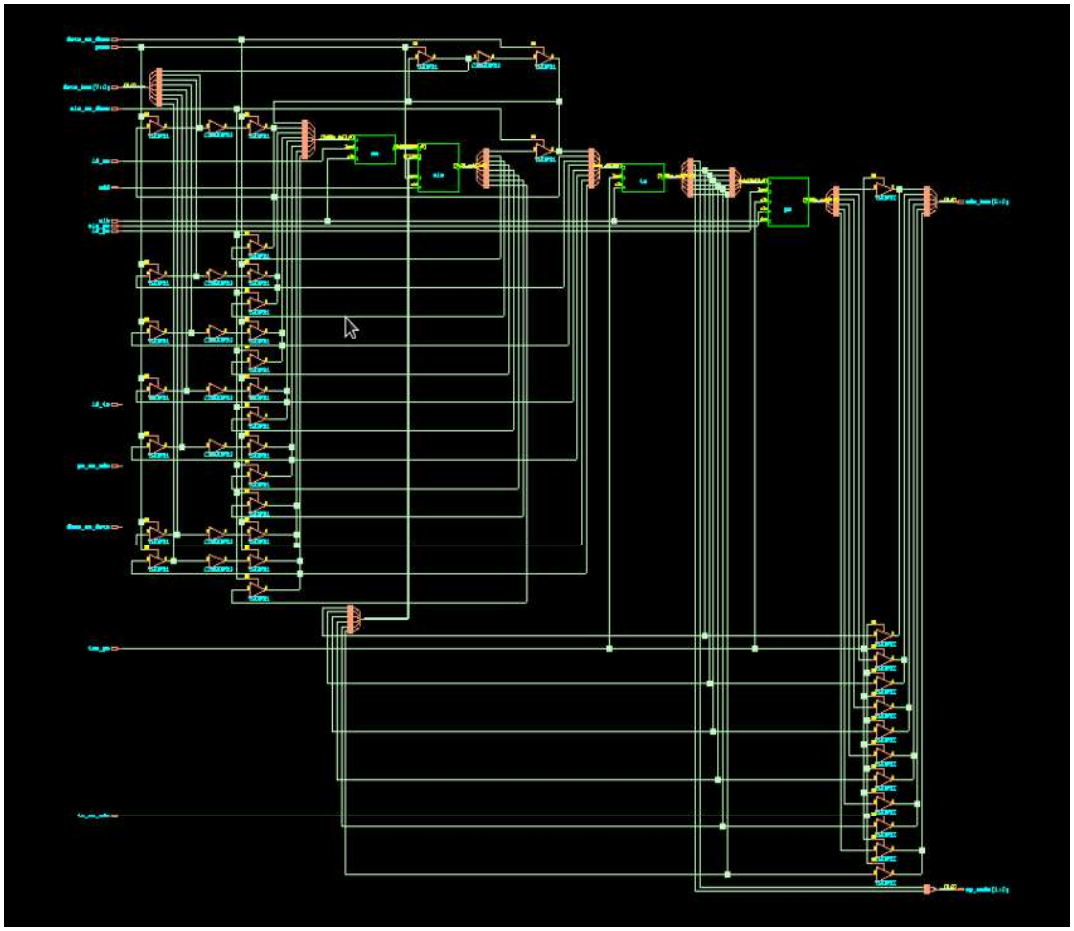
AC.v | addingCPU.sdc | addingCPU.tcl | AddingCPU.v | ALU.v | Controller.v | DataPath.v | IR.v | PC.v

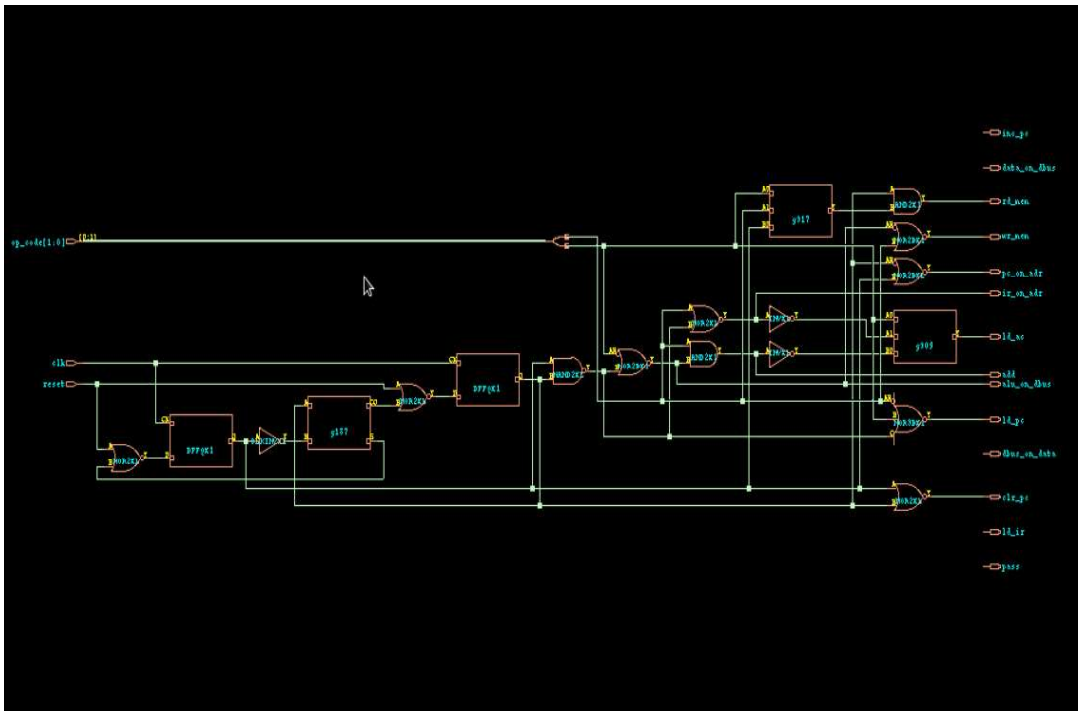
```
1 module ALU (input [7:0]a,b,input pass,add, output reg [7:0]alu_out);
2 always@(a or b or pass or add)
3 if(pass)alu_out = a;
4 else if (add) alu_out = a+b;
5 else alu_out =0;
6 endmodule
```

Gate level Netlist:



Lab-5: Logic synthesis of an adding CPU





Lab-5: Logic synthesis of an adding CPU

Observations:

Synthesis effort: Medium

a) addingCPU_netlist_created file

addingCPU_netlist.v

```
// Generated by Cadence Genus(TM) Synthesis Solution 17.21-s010_1
// Generated on: May  4 2023 12:35:49 IST (May  4 2023 07:05:49 UTC)

// Verification Directory fv/AddingCPU

module Controller(reset, clk, op_code, rd_mem, wr_mem, ir_on_adr,
     pc_on_adr, dbus_on_data, data_on_dbus, ld_ir, ld_ac, ld_pc,
     inc_pc, clr_pc, pass, add, alu_on_dbus);
  input reset, clk;
  input [1:0] op_code;
  output rd_mem, wr_mem, ir_on_adr, pc_on_adr, dbus_on_data,
       data_on_dbus, ld_ir, ld_ac, ld_pc, inc_pc, clr_pc, pass, add,
       alu_on_dbus;
  wire reset, clk;
  wire [1:0] op_code;
  wire rd_mem, wr_mem, ir_on_adr, pc_on_adr, dbus_on_data,
       data_on_dbus, ld_ir, ld_ac, ld_pc, inc_pc, clr_pc, pass, add,
       alu_on_dbus;
  wire [1:0] present_state;
  wire n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7;
  wire n_8;
  OAI21X1 g309(.A0 (op_code[0]), .A1 (n_7), .B0 (n_8), .Y (ld_ac));
  AND2X1 g310(.A (present_state[0]), .B (n_6), .Y (rd_mem));
  NOR2BX1 g311(.AN (alu_on_dbus), .B (op_code[1]), .Y (wr_mem));
  INVXL g312(.A (add), .Y (n_8));
  AND2X1 g314(.A (op_code[1]), .B (alu_on_dbus), .Y (add));
  NOR2BX1 g315(.AN (op_code[0]), .B (n_5), .Y (alu_on_dbus));
  INVXL g316(.A (ir_on_adr), .Y (n_7));
  OAI21X1 g317(.A0 (op_code[0]), .A1 (op_code[1]), .B0
       (present_state[1]), .Y (n_6));
  NOR2XL g318(.A (op_code[1]), .B (n_5), .Y (ir_on_adr));
  NAND2XL g319(.A (present_state[1]), .B (present_state[0]), .Y (n_5));
  NOR2BXL g320(.AN (present_state[0]), .B (present_state[1]), .Y
       (pc_on_adr));
  NOR2XL g321(.A (present_state[1]), .B (present_state[0]), .Y
       (clr_pc));
  NOR3BX1 g2(.AN (op_code[1]), .B (op_code[0]), .C (n_5), .Y (ld_pc));
  DFFQX1 \present_state_reg[0] (.CK (clk), .D (n_4), .Q
       (present_state[0]));
  DFFQX1 \present_state_reg[1] (.CK (clk), .D (n_3), .Q
       (present_state[1]));
  NOR2XL g185(.A (reset), .B (n_1), .Y (n_4));
  NOR2XL g186(.A (reset), .B (n_2), .Y (n_3));
  ADDHXL g187(.A (present_state[0]), .B (n_0), .CO (n_1), .S (n_2));
  CLKINVX1 g188(.A (present_state[1]), .Y (n_0));
endmodule
```

addingCPU_netlist.v

```
module DataPath(ir_on_adr, pc_on_adr, dbus_on_data, data_on_dbus,
     ld_ir, ld_ac, ld_pc, inc_pc, clr_pc, pass, add, alu_on_dbus, clk,
     adr_bus, op_code, data_bus);
  input ir_on_adr, pc_on_adr, dbus_on_data, data_on_dbus, ld_ir, ld_ac,
       ld_pc, inc_pc, clr_pc, pass, add, alu_on_dbus, clk;
  output [5:0] adr_bus;
  output [1:0] op_code;
  inout [7:0] data_bus;
  wire ir_on_adr, pc_on_adr, dbus_on_data, data_on_dbus, ld_ir, ld_ac,
       ld_pc, inc_pc, clr_pc, pass, add, alu_on_dbus, clk;
  wire [5:0] adr_bus;
  wire [1:0] op_code;
  wire [7:0] data_bus;
  wire [7:0] dbus;
  wire [7:0] a_side;
  wire [7:0] ir_out;
  wire [7:0] alu_out;
  wire [5:0] pc_out;
  wire UNCONNECTED_HIER_Z, UNCONNECTED_HIER_Z0, n_1, n_2, n_3, n_4,
       n_5, n_6;
  wire n_7, n_8;
  AC ac(dbus, ld_ac, clk, a_side);
  ALU alu(a_side, {UNCONNECTED_HIER_Z0, UNCONNECTED_HIER_Z,
       ir_out[5:0]}, pass, add, alu_out);
  IR ir(dbus, inc_pc, clk, {op_code, ir_out[5:0]});
  PC pc(ir_out[5:0], ld_pc, inc_pc, clr_pc, clk, pc_out);
  CLKBUFX2 cdn_loop_breaker(.A (data_bus[7]), .Y (n_8));
  CLKBUFX2 cdn_loop_breaker51(.A (data_bus[6]), .Y (n_7));
  CLKBUFX2 cdn_loop_breaker52(.A (data_bus[5]), .Y (n_6));
  CLKBUFX2 cdn_loop_breaker53(.A (data_bus[4]), .Y (n_5));
  CLKBUFX2 cdn_loop_breaker54(.A (data_bus[3]), .Y (n_4));
  CLKBUFX2 cdn_loop_breaker55(.A (data_bus[2]), .Y (n_3));
  CLKBUFX2 cdn_loop_breaker56(.A (data_bus[1]), .Y (n_2));
  CLKBUFX2 cdn_loop_breaker57(.A (data_bus[0]), .Y (n_1));
  TBUFXL g1(.A (ir_out[5]), .OE (ir_on_adr), .Y (adr_bus[5]));
  TBUFXL g23(.A (ir_out[4]), .OE (ir_on_adr), .Y (adr_bus[4]));
  TBUFXL g25(.A (ir_out[2]), .OE (ir_on_adr), .Y (adr_bus[2]));
  TBUFXL g26(.A (ir_out[1]), .OE (ir_on_adr), .Y (adr_bus[1]));
  TBUFXL g24(.A (ir_out[3]), .OE (ir_on_adr), .Y (adr_bus[3]));
  TBUFXL g6(.A (ir_out[0]), .OE (ir_on_adr), .Y (adr_bus[0]));
  TBUFXL g27(.A (pc_out[5]), .OE (inc_pc), .Y (adr_bus[5]));
  TBUFXL g2(.A (pc_out[4]), .OE (inc_pc), .Y (adr_bus[4]));
  TBUFXL g29(.A (pc_out[2]), .OE (inc_pc), .Y (adr_bus[2]));
  TBUFXL g30(.A (pc_out[1]), .OE (inc_pc), .Y (adr_bus[1]));
  TBUFXL g28(.A (pc_out[3]), .OE (inc_pc), .Y (adr_bus[3]));
  TBUFXL g31(.A (pc_out[0]), .OE (inc_pc), .Y (adr_bus[0]));
  TBUFX1 g39(.A (dbus[5]), .OE (pass), .Y (data_bus[5]));
  TBUFX1 g38(.A (dbus[6]), .OE (pass), .Y (data_bus[6]));
  TBUFX1 g37(.A (dbus[7]), .OE (pass), .Y (data_bus[7]));
  TBUFX1 g41(.A (dbus[2]), .OE (pass), .Y (data_bus[2]));
  TBUFX1 g43(.A (dbus[0]), .OE (pass), .Y (data_bus[0]));
  TBUFX1 g42(.A (dbus[1]), .OE (pass), .Y (data_bus[1]));
  TBUFX1 g4(.A (dbus[4]), .OE (pass), .Y (data_bus[4]));
  TBUFX1 g40(.A (dbus[3]), .OE (pass), .Y (data_bus[3]));
  TBUFX1 g7(.A (alu_out[1]), .OE (alu_on_dbus), .Y (dbus[1]));
  TBUFX1 g5(.A (n_4), .OE (data_on_dbus), .Y (dbus[3]));
  TBUFX1 g46(.A (n_6), .OE (data_on_dbus), .Y (dbus[5]));
  TBUFX1 g48(.A (n_3), .OE (data_on_dbus), .Y (dbus[2]));
  TBUFX1 g36(.A (alu_out[2]), .OE (alu_on_dbus), .Y (dbus[2]));
  TBUFX1 g3(.A (alu_out[5]), .OE (alu_on_dbus), .Y (dbus[5]));
  TBUFX1 g49(.A (n_2), .OE (data_on_dbus), .Y (dbus[1]));
  TBUFX1 g44(.A (n_8), .OE (data_on_dbus), .Y (dbus[7]));
  TBUFX1 g45(.A (n_7), .OE (data_on_dbus), .Y (dbus[6]));
```

Lab-5: Logic synthesis of an adding CPU

b) addingCPU _Constraint_created file

addingCPU_constraint_created.sdc

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

#  Created by Genus(TM) Synthesis Solution 17.21-s010_1 on Thu May 04 12:35:49 IST 2023

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

set sdc_version 2.0

set_units -capacitance 1000.0fF
set_units -time 1000.0ps

# Set the current design
current_design AddingCPU

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 reset]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[7]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[6]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[5]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[4]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[3]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[2]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[1]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[0]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[7]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[6]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[5]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[4]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[3]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[2]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[1]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[0]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[5]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[4]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[3]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[2]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[1]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {adr_bus[0]}]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports rd_mem]
set_output_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports wr_mem]
set_wire_load_mode "enclosed"
set_dont_use [get_lib_cells slow/HOLDX1]
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]
```

c) addingCPU _sdf_created file

addingCPU_sdf_created.sdf

```
(DELAYFILE
  (SDFVERSION  "OVI 3.0")
  (DESIGN      "AddingCPU")
  (DATE        "Thu May 04 12:35:49 IST 2023")
  (VENDOR      "Cadence, Inc.")
  (PROGRAM     "Genus(TM) Synthesis Solution")
  (VERSION     "17.21-s010_1")
  (DIVIDER     .)
  (VOLTAGE     ::0.9)
  (PROCESS     "::1.0")
  (TEMPERATURE ::125.0)
  (TIMESCALE   1ps)
  (CELL
     (CELLTYPE "OAI21X1")
     (INSTANCE cu.g309)
     (DELAY
        (ABSOLUTE
           (PORT A0 (::0.0))
           (PORT A1 (::0.0))
           (PORT B0 (::0.0))
           (IOPATH A0 Y (::402) (::336))
           (IOPATH B0 Y (::213) (::321))
           (IOPATH A1 Y (::419) (::327))
        )
     )
  )
  (CELL
     (CELLTYPE "AND2X1")
     (INSTANCE cu.g310)
     (DELAY
        (ABSOLUTE
           (PORT A (::0.0))
           (PORT B (::0.0))
           (IOPATH A Y (::295) (::236))
           (IOPATH B Y (::287) (::221))
        )
     )
  )
  (CELL
     (CELLTYPE "NOR2BX1")
     (INSTANCE cu.g311)
     (DELAY
        (ABSOLUTE
           (PORT AN (::0.0))
           (PORT B (::0.0))
           (IOPATH B Y (::548) (::257))
           (IOPATH AN Y (::638) (::351))
        )
     )
  )
  (CELL
     (CELLTYPE "INVXL")
     (INSTANCE cu.g312)
     (DELAY
        (ABSOLUTE
           (PORT A (::0.0))
           (IOPATH A Y (::78) (::68))
        )
     )
  )
  (CELL
     (CELLTYPE "AND2X1")
```

Lab-5: Logic synthesis of an adding CPU

d) addingCPU _timing_created file

addingCPU_timing_created.rep

```
================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           May 04 2023  12:35:50 pm
  Module:                 AddingCPU
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
================================================================

             Pin                        Type       Fanout Load Slew Delay Arrival
                                                      (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------------------------------
(clock CLK_100M)                    launch                                    0 R
cu
  present_state_reg[0]/CK                                        100             0 R
  present_state_reg[0]/Q            DFFQX1             5 10.7  108  +330     330 R
  g319/B                                                            +0     330
  g319/Y                            NAND2XL            3  7.4  199  +168     498 F
  g315/B                                                            +0     498
  g315/Y                            NOR2BX1           10 30.6  505  +410     908 R
  g311/AN                                                           +0     908
  g311/Y                            NOR2BX1           17 45.8  742  +638    1546 R
cu/wr_mem
dp/pass
  alu/pass
    g1080/A                                                         +0    1546
    g1080/Y                         INVX1              2  4.5  191   +96    1642 F
    g1074/B                                                         +0    1642
    g1074/Y                         NAND2XL            6 15.5  253  +240    1881 R
    g1070/A                                                         +0    1881
    g1070/Y                         INVX1              3  4.8   97   +77    1958 F
    g1048/A0                                                        +0    1958
    g1048/Y                         AO22X1             1  3.9   74  +204    2162 F
  alu/alu_out[3]
  g35/A                                                             +0    2162
  g35/Y                             TBUFX1             3 10.1   84  +147    2309 F
  g40/A                                                             +0    2309
  g40/Y                             TBUFX1             2  3.1   43  +108    2417 F
dp/data_bus[3]
data_bus[3]                  <<<  interconnect                      43    +0    2417 F
                                  out port                               +0    2417 F
(addingCPU.sdc_line_16_11_1)      ext delay                           +1000    3417 F
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock CLK_100M)                  capture                                  10000 R
-------------------------------------------------------------------------------------
Cost Group   : 'CLK_100M' (path_group 'CLK_100M')
Timing slack :    6583ps
Start-point  : cu/present_state_reg[0]/CK
End-point    : data_bus[3]
```

e) addingCPU _area_created file

addingCPU_area_created.rep

```
================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           May 04 2023  12:35:50 pm
  Module:                 AddingCPU
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
================================================================

 Instance   Module     Cell Count  Cell Area  Net Area   Total Area  Wireload
-----------------------------------------------------------------------------------
AddingCPU                   152   1224.664     0.000     1224.664    <none> (D)
  dp       DataPath         133   1120.969     0.000     1120.969    <none> (D)
    pc     PC                30    215.716     0.000      215.716    <none> (D)
    alu    ALU               43    214.960     0.000      214.960    <none> (D)
    ir     IR                 8    163.490     0.000      163.490    <none> (D)
    ac     AC                 8    163.490     0.000      163.490    <none> (D)
  cu       Controller        19    103.695     0.000      103.695    <none> (D)

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

Lab-5: Logic synthesis of an adding CPU

f) addingCPU _power_created file

addingCPU_power_created.rep

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           May 04 2023  12:35:50 pm
  Module:                 AddingCPU
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================

                  Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------
AddingCPU   152  6669.930 77398.874 84068.804
  dp        133  6137.071 65606.163 71743.234
    pc       30  1167.830  9818.709 10986.539
    ac        8  1090.048 18564.195 19654.243
    ir        8  1090.048 20632.354 21722.402
    alu      43   887.626  5009.390  5897.015
  cu         19   532.858  8834.186  9367.045
```

g) addingCPU _gates_created file

addingCPU_gates_created.rep

```
====================================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           May 04 2023  12:35:50 pm
  Module:                 AddingCPU
  Technology library:     slow
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
====================================================================


   Gate    Instances    Area   Library
-----------------------------------------
ADDFX1             2   39.359    slow
ADDHXL             1   12.110    slow
AND2X1             2    9.083    slow
AO22X1            10   75.698    slow
AOI21X1            7   31.790    slow
AOI21XL            1    4.541    slow
AOI2BB1X1          1    6.055    slow
CLKBUFX2           8   36.331    slow
CLKINVX1           2    4.541    slow
CLKXOR2X1          1    8.326    slow
DFFQX1             8  127.159    slow
INVX1              2    4.541    slow
INVXL              6   13.624    slow
MXI2XL             1    6.055    slow
NAND2BX1           1    4.541    slow
NAND2BXL           4   18.166    slow
NAND2XL           12   36.331    slow
NOR2BX1            2    9.083    slow
NOR2BXL            2    9.083    slow
NOR2XL            12   36.331    slow
NOR3BX1            1    6.055    slow
OAI21X1            5   22.707    slow
OAI21XL            5   22.707    slow
OAI31X1            1    6.055    slow
OAI32X1            3   20.436    slow
SDFFQX1           16  326.981    slow
TBUFX1            24  217.987    slow
TBUFXL            12  108.994    slow
-----------------------------------------
total            152  1224.664


     Type      Instances   Area   Area %
-----------------------------------------
sequential            24  454.140   37.1
inverter              10   22.707    1.9
buffer                 8   36.331    3.0
tristate              36  326.981   26.7
logic                 74  384.505   31.4
physical_cells         0    0.000    0.0
-----------------------------------------
total                152 1224.664  100.0
```

Lab-5: Logic synthesis of an adding CPU

h) addingCPU _quality_created file

addingCPU_quality_created.rep

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.21-s010_1
  Generated on:           May 04 2023  12:35:50 pm
  Module:                 AddingCPU
  Technology library:     slow
  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      6582.9   0.0          0
default     No paths   0.0
-------------------------------------
Total                  0.0          0

Instance Count
--------------
Leaf Instance Count             152
Physical Instance count           0
Sequential Instance Count        24
Combinational Instance Count    128
Hierarchical Instance Count       6

Area
----
Cell Area                         1224.664
Physical Cell Area                0.000
Total Cell Area (Cell+Physical)   1224.664
Net Area                          0.000
Total Area (Cell+Physical+Net)    1224.664

Max Fanout                        20 (pc_on_adr)
Min Fanout                        0 (dp/alu/b[7])
Average Fanout                    2.8
Terms to net ratio                3.6345
Terms to instance ratio           3.4671
Runtime                           15.096715 seconds
Elapsed Runtime                   197 seconds
Genus peak memory usage           751.47
Innovus peak memory usage         no_value
Hostname                          cad20
```

Inference: A total of 152 leaf instance count is present in the gate level netlist with total area of 1224.664, total power of 84068.804nW.

Result: Hence an addingCPU is synthesized and the gate level netlist with timing, area and power report has been generated.