

VELLORE INSTITUTE OF TECHNOLOGY, CHENNAI

BECE407P – ASIC DESIGN

LAB-5

PHYSICAL DESIGN OF A FULL-ADDER AND ADDING CPU

Name of the Student: Sudharsan S

Roll Number: 21BLC1079

Date of Submission: 04-11-2024

AIM:

To do physical design for full adder and adding CPU.

EDA TOOLS USED:

Cadence® Innovus.

DESIGN PROCEDURE:

- 1) Create a Synopsys design constraint file (.SDC file) by entering the design constraints required for the synthesis.
- 2) Create a (.tcl) file containing all the commands for performing the logic synthesis. Synthesis effort can be medium or high.
- 3) Invoke the C shell and launch the Genus tool by entering the below commands.

```
>csh  
>source home/install/cshrc  
>genus -legacy_ui -f filename.tcl
```

- 4) Execute the commands in the (.tcl file) by entering source addingCPU.tcl command in the command line window.
- 5) After generating the netlist file, copy paste it to a new folder for physical design using innovus.
- 6) Copy the fast.lib and slow.lib files into the folder where (.v) files are located.
- 7) Invoke the C shell and launch the Genus tool by entering the below commands.

```
>csh  
>source home/install/cshrc  
>innovus
```

- 8) Goto File -> Import Design
- 9) Add the netlist file for the design
- 10) Add LEF file from//CAD/FOUNDRY/180NM/dig/lef/all.lef
- 11) Specify the Power Nets as “vdd” and Ground Nets as “vss”

- 12) Customize “MMMC objects” and “Analysis views” by clicking on “Create Analysis Configuration”
- 13) Define • Library Sets • RC Corners • Delay Corners • Constraint Modes • Analysis Views • Setup Analysis Views • Hold Analysis Views
- 14) Go to floorplan. Specify Floorplan -> Set Aspect ratio to 1 and put all ‘Core to Die Boundary’ to 2.5 and click Ok.
- 15) Go to Power -> Power planning -> Add ring -> Nets – VDD VSS (type it). For power we use the top most metal layer (top & bottom choose Metal 5, left & right choose Metal 6). Make all width to 0.7, spacing 0.2 and Offset to 0.5. Click Ok.
- 16) Again go to Power -> Power Planning -> Add Stripes -> Nets – vdd vss(type it) -> Choose Metal 6, Vertical and set width to 0.7 and space to 0.2. Click ‘Set-to-set to 5. Set Start to 1 and Stop to 0 at the bottom(First and last stripe)
- 17) Go to Route -> Special route -> Nets – vdd vss (type it) and press Ok Uncheck all except ‘Follow Pins’ in SRoute
- 18) Go to tools -> Set mode -> Specify analysis mode -> in that enable OCV and CPPR
- 19) Go to tools -> Set mode -> Mode setup -> Click on Placement and enable I/O pins and press Ok
- 20) Now go to Place -> Standard Cell -> Click Ok.
- 21) Go to timing -> report timing, choose pre CTS and do for both setup and hold one by one. A timing report named folder gets generated. Check for values of setup and hold slack whether they are positive or not.
- 22) Write a CTS script file and save it as ‘ccopt.spec’
- 23) In the terminal >source ccopt.spec
- 24) Save Design >saveDesign DBS/cts.enc1
- 25) Go to Place -> Nano route -> Route (both global and detailed routing done in this step). 2. Enable Optimize via and Optimize wire and Click Ok
- 26) Go to Place -> Physical cell -> Add Fillers -> Click on Select -> Add all of the filler cells -> Close -> Ok
- 27) Check ‘Do DRC’ and ‘Fit Gap’
- 28) Go to Timing -> Extract RC -> Check the boxes ‘Save SPF to’ & ‘Save SPEF to’ and press Ok
- 29) Finally, Go to File -> Save -> Netlist -> Ok.
- 30) File -> Save Design -> Enable Innovus and type filename.enc and click Ok

VERILOG PROGRAMS FOR ADDING CPU

ADDING CPU

```
module AddingCPU(input reset,clk, output [5:0]adr_bus, output rd_mem, wr_mem, 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;
wire [1:0] op_code;
Controller cu (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);
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,alu_on_dbus,clk,adr_bus,op_code,data_bus);
endmodule
```

CONTROLLER

```

`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_addr,pc_on_addr,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_addr=1'b0;
    pc_on_addr=1'b0;
    dbus_on_data=1'b0;
    data_on_dbus=1'b0;
    ld_ir=1'b0;
    ld_ac=1'b0;
    ld_pc=11'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_addr = 1'b1;
            pc_on_addr = 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_addr = 1'b1;
                    rd_mem = 1'b1;
                    data_on_dbus = 1'b1;
                    ld_ac = 1'b1;
                end
                2'b01: begin
                    pass = 1'b1;
                    ir_on_addr = 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

```

DATA PATH

```

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 op_code = ir_out[7:6];
endmodule

```

IR

```

module IR(data_in,load,clk,data_out);
input [7:0] data_in;
input clk,load;
output reg [7:0] data_out;
always@(posedge clk)
begin
    if(load)
        begin
            data_out<=data_in;
        end
end
endmodule

```

PC

```

module PC(data_in,load,clk,clr,inc,data_out);
input [5:0] data_in;
input load,clk,clr,inc;
output reg [5:0] data_out;
always@(posedge clk)
begin
    if (clr) data_out<=6'b000_000;
    else if (load) data_out<=data_in;
    else if (inc) data_out<=data_out+1;
end
endmodule

```

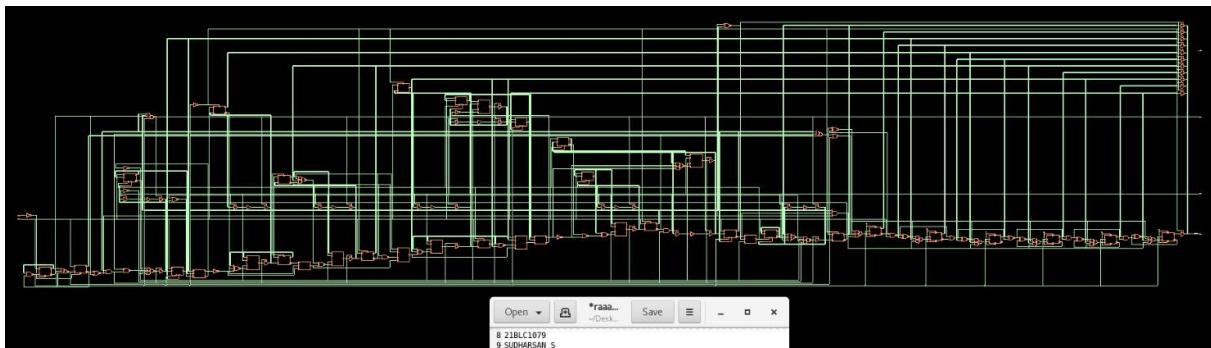
AC

```

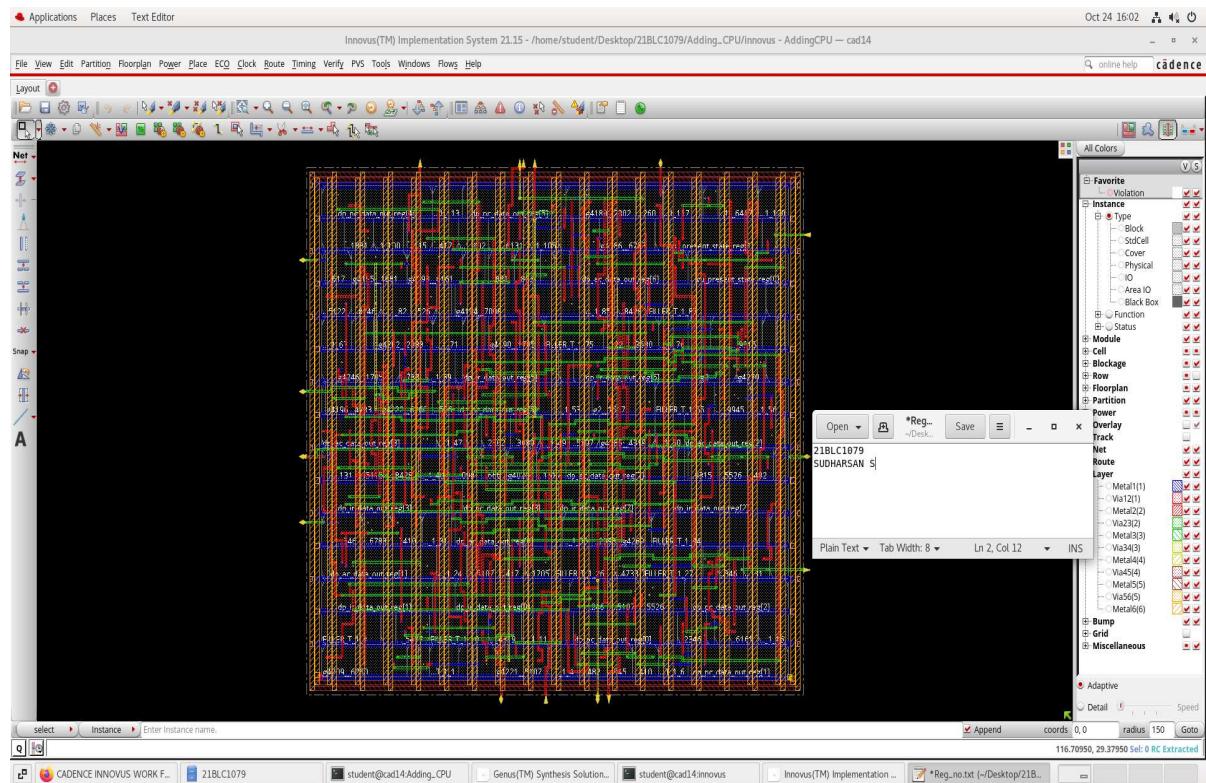
module AC(input [7:0] data_in, input load,clk, output reg[7:0] data_out);
always@(posedge clk)
    if (load) data_out <= data_in;
endmodule

```

GATE LEVEL NETLIST



PHYSICAL DESIGN



OBSERVATIONS

a) ADDINGCPU_NETLIST_CREATED FILE

```

// Generated by Cadence Genus(TM) Synthesis Solution 21.14-s082_1
// Generated on: Oct 24 2024 15:19:32 IST (Oct 24 2024 09:49:32 UTC)

// Verification Directory fv/AddingCPU

module AddingCPU(reset, clk, adr_bus, rd_mem, wr_mem, data_bus);
    input reset, clk;
    output [5:0] adr_bus;
    output rd_mem, wr_mem;
    inout [7:0] data_bus;
    wire reset, clk;
    wire [5:0] adr_bus;
    wire rd_mem, wr_mem;
    wire [7:0] data_bus;
    wire [7:0] dp_dbus;
    wire [7:0] dp_a_side;
    wire [7:0] dp_ir_out;
    wire [1:0] op_code;
    wire [5:0] dp_pc_out;
    wire [1:0] cu_present_state;
    wire UNCONNECTED, UNCONNECTED0, UNCONNECTED1, UNCONNECTED2, dp_n_56,
        dp_n_57, dp_n_58, dp_n_59;
    wire dp_n_60, dp_n_61, dp_n_62, dp_n_63, n_0, n_1, n_2, n_3;
    wire n_5, n_6, n_7, n_8, n_9, n_10, n_11, n_12;
    wire n_13, n_14, n_15, n_16, n_17, n_18, n_19, n_20;
    wire n_21, n_22, n_23, n_24, n_25, n_26, n_27, n_28;
    wire n_29, n_30, n_31, n_32, n_33, n_34, n_35, n_36;
    wire n_37, n_38, n_39, n_40, n_41, n_42, n_43, n_44;
    wire n_45, n_46, n_47, n_48, n_49, n_50, n_51, n_52;
    wire n_53, n_54, n_55, n_56, n_57, n_58, n_59, n_61;
    wire n_63, n_64, n_65, n_66, n_67, n_68, n_69, n_86;
    CLKBUFX2 dp_cdn_loop_breaker(.A (data_bus[7]), .Y (dp_n_56));
    CLKBUFX2 dp_cdn_loop_breaker58(.A (data_bus[6]), .Y (dp_n_57));
    CLKBUFX2 dp_cdn_loop_breaker59(.A (data_bus[5]), .Y (dp_n_58));
    CLKBUFX2 dp_cdn_loop_breaker60(.A (data_bus[4]), .Y (dp_n_59));
    CLKBUFX2 dp_cdn_loop_breaker61(.A (data_bus[3]), .Y (dp_n_60));
    CLKBUFX2 dp_cdn_loop_breaker62(.A (data_bus[2]), .Y (dp_n_61));
    CLKBUFX2 dp_cdn_loop_breaker63(.A (data_bus[1]), .Y (dp_n_62));
    CLKBUFX2 dp_cdn_loop_breaker64(.A (data_bus[0]), .Y (dp_n_63));
    TBUFXL dp_g37_2398(.A (dp_dbus[7]), .OE (wr_mem), .Y (data_bus[7]));
    TBUFXL dp_g32_5107(.A (n_65), .OE (n_38), .Y (dp_dbus[7]));
    TBUFXL dp_g38_6260(.A (dp_dbus[6]), .OE (wr_mem), .Y (data_bus[6]));
    OAI21X1 g1503_4319(.A0 (n_67), .A1 (n_63), .B0 (n_43), .Y (n_65));
    TBUFXL dp_g33_8428(.A (n_64), .OE (n_38), .Y (dp_dbus[6]));
    TBUFXL dp_g39_5526(.A (dp_dbus[5]), .OE (wr_mem), .Y (data_bus[5]));
    OAI21X1 g1504_6783(.A0 (n_67), .A1 (n_61), .B0 (n_42), .Y (n_64));
    XNOR2X1 g1505_3680(.A (dp_a_side[7]), .B (n_59), .Y (n_63));
    TBUFXL dp_g3_1617(.A (n_86), .OE (n_38), .Y (dp_dbus[5]));
    TBUFXL dp_g4_2802(.A (dp_dbus[4]), .OE (wr_mem), .Y (data_bus[4]));
    XNOR2X1 g1507_5122(.A (dp_a_side[6]), .B (n_58), .Y (n_61));
    AND2XL g1509_7098(.A (dp_a_side[6]), .B (n_58), .Y (n_59));
    TBUFXL dp_g34_6131(.A (n_56), .OE (n_38), .Y (dp_dbus[4]));
    ADDFX1 g1510_1881(.A (dp_ir_out[5]), .B (dp_a_side[5]), .CI (n_54),
        .CO (n_58), .S (n_57));
    TBUFXL dp_g40_5115(.A (dp_dbus[3]), .OE (wr_mem), .Y (data_bus[3]));
    AO22XL g1511_7482(.A0 (n_39), .A1 (n_55), .B0 (dp_a_side[4]), .B1
        (wr_mem), .Y (n_56));
    TBUFXL dp_g35_4733(.A (n_53), .OE (n_38), .Y (dp_dbus[3]));
    ADDFX1 g1512_6161(.A (dp_a_side[4]), .B (dp_ir_out[4]), .CI (n_51),
        .CO (n_54), .S (n_55));

```

```

TBUFXL dp_g41_9315(.A (dp_dbus[2]), .OE (wr_mem), .Y (data_bus[2]));
AO22XL g1513_9945(.A0 (n_39), .A1 (n_52), .B0 (dp_a_side[3]), .B1
(wr_mem), .Y (n_53));
TBUFXL dp_g36_2883(.A (n_50), .OE (n_38), .Y (dp_dbus[2]));
ADDFX1 g1514_2346(.A (dp_ir_out[3]), .B (dp_a_side[3]), .CI (n_48),
.CO (n_51), .S (n_52));
TBUFXL dp_g42_1666(.A (dp_dbus[1]), .OE (wr_mem), .Y (data_bus[1]));
AO22XL g1515_7410(.A0 (n_39), .A1 (n_49), .B0 (dp_a_side[2]), .B1
(wr_mem), .Y (n_50));
TBUFXL dp_g7_6417(.A (n_47), .OE (n_38), .Y (dp_dbus[1]));
ADDFX1 g1516_5477(.A (dp_a_side[2]), .B (dp_ir_out[2]), .CI (n_45),
.CO (n_48), .S (n_49));
TBUFXL dp_g43_2398(.A (dp_dbus[0]), .OE (wr_mem), .Y (data_bus[0]));
AO22XL g1517_5107(.A0 (n_46), .A1 (n_39), .B0 (dp_a_side[1]), .B1
(wr_mem), .Y (n_47));
TBUFXL dp_g8_6260(.A (n_44), .OE (n_38), .Y (dp_dbus[0]));
ADDFX1 g1518_4319(.A (dp_ir_out[1]), .B (dp_a_side[1]), .CI (n_40),
.CO (n_45), .S (n_46));
AO22XL g1519_8428(.A0 (n_41), .A1 (n_39), .B0 (dp_a_side[0]), .B1
(wr_mem), .Y (n_44));
TBUFXL dp_g5_5526(.A (dp_n_60), .OE (rd_mem), .Y (dp_dbus[3]));
TBUFXL dp_g48_6783(.A (dp_n_61), .OE (rd_mem), .Y (dp_dbus[2]));
TBUFXL dp_g49_3680(.A (dp_n_62), .OE (rd_mem), .Y (dp_dbus[1]));
TBUFXL dp_g50_1617(.A (dp_n_63), .OE (rd_mem), .Y (dp_dbus[0]));
TBUFXL dp_g45_2802(.A (dp_n_57), .OE (rd_mem), .Y (dp_dbus[6]));
TBUFXL dp_g44_1705(.A (dp_n_56), .OE (rd_mem), .Y (dp_dbus[7]));
TBUFXL dp_g46_5122(.A (dp_n_58), .OE (rd_mem), .Y (dp_dbus[5]));
TBUFXL dp_g47_8246(.A (dp_n_59), .OE (rd_mem), .Y (dp_dbus[4]));
NAND2X1 g1520_7098(.A (dp_a_side[7]), .B (wr_mem), .Y (n_43));
NAND2X1 g1521_6131(.A (dp_a_side[6]), .B (wr_mem), .Y (n_42));
TBUFXL dp_g6_1881(.A (dp_ir_out[0]), .OE (n_37), .Y (adr_bus[0]));
TBUFXL dp_g24_5115(.A (dp_ir_out[3]), .OE (n_37), .Y (adr_bus[3]));
TBUFXL dp_g25_7482(.A (dp_ir_out[2]), .OE (n_37), .Y (adr_bus[2]));
TBUFXL dp_g1_4733(.A (dp_ir_out[5]), .OE (n_37), .Y (adr_bus[5]));
TBUFXL dp_g23_6161(.A (dp_ir_out[4]), .OE (n_37), .Y (adr_bus[4]));
TBUFXL dp_g26_9315(.A (dp_ir_out[1]), .OE (n_37), .Y (adr_bus[1]));
ADDHX1 g1522_9945(.A (dp_a_side[0]), .B (dp_ir_out[0]), .CO (n_40),
.S (n_41));
CLKINVX1 g1524(.A (n_67), .Y (n_39));
NAND2X1 g1525_2883(.A (op_code[1]), .B (n_38), .Y (n_67));
OAI21X1 g1526_2346(.A0 (op_code[1]), .A1 (n_35), .B0 (n_66), .Y
(rd_mem));
TBUFXL dp_g31_1666(.A (dp_pc_out[0]), .OE (n_36), .Y (adr_bus[0]));
TBUFXL dp_g29_7410(.A (dp_pc_out[2]), .OE (n_36), .Y (adr_bus[2]));
TBUFXL dp_g27_6417(.A (dp_pc_out[5]), .OE (n_36), .Y (adr_bus[5]));
TBUFXL dp_g30_5477(.A (dp_pc_out[1]), .OE (n_36), .Y (adr_bus[1]));
TBUFXL dp_g28_2398(.A (dp_pc_out[3]), .OE (n_36), .Y (adr_bus[3]));
TBUFXL dp_g2_5107(.A (dp_pc_out[4]), .OE (n_36), .Y (adr_bus[4]));
NOR2BX1 g1528_6260(.AN (op_code[0]), .B (n_69), .Y (n_38));
INVX1 g1529(.A (n_37), .Y (n_68));
NOR2X1 g1530_4319(.A (op_code[1]), .B (n_69), .Y (n_37));
NAND2X1 g1531_8428(.A (cu_present_state[0]), .B
(cu_present_state[1]), .Y (n_69));
INVX1 g1532(.A (n_36), .Y (n_66));
NAND2BX1 g1533_5526(.AN (op_code[0]), .B (cu_present_state[0]), .Y
(n_35));
NOR2BX1 g1534_6783(.AN (cu_present_state[0]), .B
(cu_present_state[1]), .Y (n_36));
NOR2BX1 g2_3680(.AN (n_38), .B (op_code[1]), .Y (wr_mem));
DFFNSRX1 \dp_pc_data_out_reg[5] (.RN (1'b1), .SN (1'b1), .CKN (n_66),

```

```

.D (n_34), .Q (dp_pc_out[5]), .QN (UNCONNECTED));
DFFNSRX1 \dp_pc_data_out_reg[4] (.RN (1'b1), .SN (1'b1), .CKN (n_66),
.D (n_33), .Q (dp_pc_out[4]), .QN (n_3));
OAI2BB1X1 g1381_1617(.A0N (dp_pc_out[5]), .A1N (n_29), .B0 (n_32),
.Y (n_34));
DFFNSRX1 \dp_pc_data_out_reg[3] (.RN (1'b1), .SN (1'b1), .CKN (n_66),
.D (n_31), .Q (dp_pc_out[3]), .QN (UNCONNECTED0));
OAI2BB1X1 g1383_2802(.A0N (dp_pc_out[4]), .A1N (n_29), .B0 (n_30),
.Y (n_33));
AOI22XL g1384_1705(.A0 (n_27), .A1 (n_0), .B0 (dp_ir_out[5]), .B1
(n_8), .Y (n_32));
OAI2BB1X1 g1385_5122(.A0N (dp_pc_out[3]), .A1N (n_16), .B0 (n_28),
.Y (n_31));
AOI32X1 g1386_8246(.A0 (n_3), .A1 (n_22), .A2 (n_0), .B0
(dp_ir_out[4]), .B1 (n_8), .Y (n_30));
OAI21XL g1387_7098(.A0 (n_22), .A1 (n_14), .B0 (n_15), .Y (n_29));
AOI22XL g1388_6131(.A0 (n_23), .A1 (n_0), .B0 (dp_ir_out[3]), .B1
(n_8), .Y (n_28));
OAI2BB1X1 g1389_1881(.A0N (dp_pc_out[5]), .A1N (n_3), .B0 (n_26), .Y
(n_27));
DFFNSRX1 \dp_pc_data_out_reg[2] (.RN (1'b1), .SN (1'b1), .CKN (n_66),
.D (n_25), .Q (dp_pc_out[2]), .QN (UNCONNECTED1));
NAND3BXL g1391_5115(.AN (dp_pc_out[5]), .B (dp_pc_out[4]), .C
(n_22), .Y (n_26));
OAI2BB1X1 g1392_7482(.A0N (dp_pc_out[2]), .A1N (n_16), .B0 (n_24),
.Y (n_25));
DFFNSRX1 \dp_pc_data_out_reg[1] (.RN (1'b1), .SN (1'b1), .CKN (n_66),
.D (n_21), .Q (dp_pc_out[1]), .QN (UNCONNECTED2));
DFFNSRX1 \dp_pc_data_out_reg[0] (.RN (1'b1), .SN (1'b1), .CKN (n_66),
.D (n_20), .Q (dp_pc_out[0]), .QN (n_2));
AOI22XL g1395_4733(.A0 (n_18), .A1 (n_0), .B0 (dp_ir_out[2]), .B1
(n_8), .Y (n_24));
XNOR2X1 g1396_6161(.A (dp_pc_out[3]), .B (n_17), .Y (n_23));
NOR2BX1 g1397_9315(.AN (dp_pc_out[3]), .B (n_17), .Y (n_22));
OAI2BB1X1 g1398_9945(.A0N (dp_pc_out[1]), .A1N (n_16), .B0 (n_19),
.Y (n_21));
OAI221X1 g1399_2883(.A0 (n_2), .A1 (n_15), .B0 (dp_pc_out[0]), .B1
(n_14), .C0 (n_11), .Y (n_20));
AOI22XL g1400_2346(.A0 (n_7), .A1 (n_0), .B0 (dp_ir_out[1]), .B1
(n_8), .Y (n_19));
SDFFQX1 \dp_ac_data_out_reg[6] (.CK (clk), .D (dp_a_side[6]), .SI
(dp_dbus[6]), .SE (n_10), .Q (dp_a_side[6]));
SDFFQX1 \dp_ac_data_out_reg[5] (.CK (clk), .D (dp_a_side[5]), .SI
(dp_dbus[5]), .SE (n_10), .Q (dp_a_side[5]));
SDFFQX1 \dp_ac_data_out_reg[4] (.CK (clk), .D (dp_a_side[4]), .SI
(dp_dbus[4]), .SE (n_10), .Q (dp_a_side[4]));
SDFFQX1 \dp_ac_data_out_reg[7] (.CK (clk), .D (dp_a_side[7]), .SI
(dp_dbus[7]), .SE (n_10), .Q (dp_a_side[7]));
XNOR2X1 g1405_1666(.A (dp_pc_out[2]), .B (n_6), .Y (n_18));
SDFFQX1 \dp_ac_data_out_reg[2] (.CK (clk), .D (dp_a_side[2]), .SI
(dp_dbus[2]), .SE (n_10), .Q (dp_a_side[2]));
SDFFQX1 \dp_ac_data_out_reg[1] (.CK (clk), .D (dp_a_side[1]), .SI
(dp_dbus[1]), .SE (n_10), .Q (dp_a_side[1]));
SDFFQX1 \dp_ac_data_out_reg[0] (.CK (clk), .D (dp_a_side[0]), .SI
(dp_dbus[0]), .SE (n_10), .Q (dp_a_side[0]));
SDFFQX1 \dp_ac_data_out_reg[3] (.CK (clk), .D (dp_a_side[3]), .SI
(dp_dbus[3]), .SE (n_10), .Q (dp_a_side[3]));
DFFHQX1 \cu_present_state_reg[1] (.CK (clk), .D (n_12), .Q
(cu_present_state[1]));
NAND2BXL g1411_7410(.AN (n_6), .B (dp_pc_out[2]), .Y (n_17));

```

```

INVX1 g1412(.A (n_16), .Y (n_15));
INVX1 g1413(.A (n_0), .Y (n_14));
NOR2XL g1414_6417(.A (clk), .B (n_13), .Y (n_16));
SDFFQX1 \dp_ir_data_out_reg[2] (.CK (clk), .D (dp_dbus[2]), .SI
    (.dp_ir_out[2]), .SE (n_66), .Q (dp_ir_out[2]));
SDFFQX1 \dp_ir_data_out_reg[1] (.CK (clk), .D (dp_dbus[1]), .SI
    (.dp_ir_out[1]), .SE (n_66), .Q (dp_ir_out[1]));
SDFFQX1 \dp_ir_data_out_reg[0] (.CK (clk), .D (dp_dbus[0]), .SI
    (.dp_ir_out[0]), .SE (n_66), .Q (dp_ir_out[0]));
SDFFQX1 \dp_ir_data_out_reg[3] (.CK (clk), .D (dp_dbus[3]), .SI
    (.dp_ir_out[3]), .SE (n_66), .Q (dp_ir_out[3]));
SDFFQX1 \dp_ir_data_out_reg[4] (.CK (clk), .D (dp_dbus[4]), .SI
    (.dp_ir_out[4]), .SE (n_66), .Q (dp_ir_out[4]));
AOI21XL g1421_5477(.A0 (n_66), .A1 (n_5), .B0 (reset), .Y (n_12));
NAND2XL g1422_2398(.A (dp_ir_out[0]), .B (n_8), .Y (n_11));
OAI21XL g1423_5107(.A0 (cu_present_state[0]), .A1
    (cu_present_state[1]), .B0 (n_9), .Y (n_13));
SDFFQX1 \dp_ir_data_out_reg[5] (.CK (clk), .D (dp_dbus[5]), .SI
    (.dp_ir_out[5]), .SE (n_66), .Q (dp_ir_out[5]));
SDFFQX1 \dp_ir_data_out_reg[7] (.CK (clk), .D (dp_dbus[7]), .SI
    (op_code[1]), .SE (n_66), .Q (op_code[1]));
SDFFQX1 \dp_ir_data_out_reg[6] (.CK (clk), .D (dp_dbus[6]), .SI
    (op_code[0]), .SE (n_66), .Q (op_code[0]));
DFFHQX1 \cu_present_state_reg[0] (.CK (clk), .D (n_1), .Q
    (cu_present_state[0]));
OAI21XL g1428_6260(.A0 (op_code[0]), .A1 (n_68), .B0 (n_67), .Y
    (n_10));
INVXL g1429(.A (n_8), .Y (n_9));
XNOR2XL g1430_4319(.A (dp_pc_out[1]), .B (n_2), .Y (n_7));
NOR3BXL g1431_8428(.AN (op_code[1]), .B (op_code[0]), .C (n_69), .Y
    (n_8));
NAND2BX1 g1433_5526(.AN (cu_present_state[0]), .B
    (cu_present_state[1]), .Y (n_5));
NAND2XL g1434_6783(.A (dp_pc_out[1]), .B (dp_pc_out[0]), .Y (n_6));
NOR2BX1 g1535_3680(.AN (n_66), .B (reset), .Y (n_1));
NOR2BX1 g1439_1617(.AN (clk), .B (n_13), .Y (n_0));
AO22XL g2(.A0 (dp_a_side[5]), .A1 (wr_mem), .B0 (n_39), .B1 (n_57),
    .Y (n_86));
endmodule

```

b) ADDING CPU_CONSTRAINT_CREATED FILE

```

# #####
# Created by Genus(TM) Synthesis Solution 21.14-s082_1 on Thu Oct 24 15:19:32 IST 2024
# #####
set sdc_version 2.0
set_units -capacitance 1000ff
set_units -time 1000ps
# 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] -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[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[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[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[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[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[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[1]}]
set_input_delay -clock [get_clocks CLK_100M] -add_delay -max 1.0 [get_ports {data_bus[0]}]
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[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[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[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[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[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[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 {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"

```

c) ADDINGCPU_SDF_CREATED FILE

```

*SPEF "IEEE 1481-1998"
*DESIGN "AddingCPU"
*DATE "Thu Oct 24 15:59:29 2024"
*VENDOR "Cadence Design Systems, Inc."
*PROGRAM "Innovus"
*VERSION "21.15-s110_1"
*DESIGN_FLOW "PIN_CAP NONE" "NAME_SCOPE LOCAL"
*DIVIDER /
*DELIMITER :
*BUS_DELIMITER []
*T_UNIT 1 NS
*C_UNIT 1 PF
*R_UNIT 1 OHM
*L_UNIT 1 HENRY

// MMIC spef file for corner 'default_rc_corner'

*NAME_MAP
*1 reset
*2 clk
*3 adr_bus[5]
*4 adr_bus[4]
*5 adr_bus[3]
*6 adr_bus[2]
*7 adr_bus[1]
*8 adr_bus[0]
*9 rd_mem
*10 wr_mem
*11 data_bus[7]
*12 data_bus[6]
*13 data_bus[5]
*14 data_bus[4]
*15 data_bus[3]
*16 data_bus[2]
*17 data_bus[1]
*18 data_bus[0]
*19 dp_dbus[7]
*20 dp_dbus[6]
*21 dp_dbus[5]
*22 dp_dbus[4]
*23 dp_dbus[3]
*24 dp_dbus[2]
*25 dp_dbus[1]
*26 dp_dbus[0]
*27 dp_a_side[7]
*28 dp_a_side[6]
*29 dp_a_side[5]
*30 dp_a_side[4]
*31 dp_a_side[3]
*32 dp_a_side[2]
*33 dp_a_side[1]
*34 dp_a_side[0]
*35 dp_ir_out[5]
*36 dp_ir_out[4]
*37 dp_ir_out[3]
*38 dp_ir_out[2]
*39 dp_ir_out[1]
*40 dp_ir_out[0]
*41 op_code[1]
*42 op_code[0]
*43 dp_pc_out[5]
*44 dp_pc_out[4]
*45 dp_pc_out[3]
*46 dp_pc_out[2]
*47 dp_pc_out[1]
*48 dp_pc_out[0]
*49 cu_present_state[1]
*50 cu_present_state[0]
*51 UNCONNECTED
*52 UNCONNECTED0
*53 UNCONNECTED1
*54 UNCONNECTED2
*55 dp_n_56
*56 dp_n_57
*57 dp_n_58
*58 dp_n_59
*59 dp_n_60
*60 dp_n_61
*61 dp_n_62
*62 dp_n_63
*63 n_0
*64 n_1
*65 n_2
*66 n_3
*67 n_5
*68 n_6
*69 n_7

*4 O *C 0 0
*5 O *C 0 0
*6 O *C 0 0
*7 O *C 0 0
*8 O *C 0 0
*9 O *C 0 0
*10 O *C 0 0
*11 B *C 0 0
*12 B *C 0 0
*13 B *C 0 0
*14 B *C 0 0
*15 B *C 0 0
*16 B *C 0 0
*17 B *C 0 0
*18 B *C 0 0

*D_NET *1 0.000522807
*CONN
*I *237:B I *C 61.71 40.6 *L 0.0041 *D NOR2BX1
*I *227:B0 I *C 65.01 40.04 *L 0.0041 *D AOI2IX1

*CAP
1 *1:2 0.000218867
2 *1:3 0.000218867
3 *1:4 4.25364e-05
4 *1:5 4.25364e-05

*RES
1 *1:5 *227:B0 6.4
2 *1:4 *1:5 0.0733943
3 *1:3 *1:4 6.4
4 *1:2 *1:3 1.19039
5 *1:1 *1:2 6.4
6 *237:B *1:1 6.4
*END

*D_NET *2 0.00242657
*CONN
*I *238:AN I *C 53.13 45.64 *L 0.00218 *D NOR2BX1
*I *226:A I *C 54.45 46.2 *L 0.00317 *D NOR2XL
*I *222:CK I *C 65.01 45.64 *L 0.00291 *D DFFHQX1
*I *230:CK I *C 59.73 40.6 *L 0.00291 *D DFFHQX1

*CAP
1 *2:2 8.75469e-05
2 *2:3 0.000437734
3 *2:4 0.000700375
4 *2:5 4.25364e-05
5 *2:6 0.000350188
6 *2:7 0.000362828
7 *2:8 4.25364e-05
8 *2:10 0.000362828

*RES
1 *2:10 *230:CK 6.4
2 *2:9 *222:CK 6.4
3 *2:8 *226:A 6.4
4 *2:7 *2:10 0.660548
5 *2:6 *2:9 6.4
6 *2:5 *2:8 0.0733943
7 *2:4 *2:7 6.4
8 *2:4 *2:6 1.98462
9 *2:3 *2:5 6.4
10 *2:3 *2:4 1.90462
11 *2:2 *2:3 0.476154
12 *2:1 *2:2 6.4
13 *238:AN *2:1 6.4
*END

*D_NET *3 9.53973e-05
*CONN
*I *178:Y O *C 6.93 26.6 *L 0.00221 *D TBUFXL
*I *187:Y O *C 7.59 26.6 *L 0.00221 *D TBUFXL

*CAP
1 *178:Y 4.76987e-05
2 *187:Y 4.76987e-05

*RES
1 *178:Y *187:Y 0.238077
*END

*D_NET *4 9.53973e-05
*CONN
*I *179:Y O *C 10.23 36.68 *L 0.00221 *D TBUFXL

```

VERILOG PROGRAM FOR FULL ADDER

```

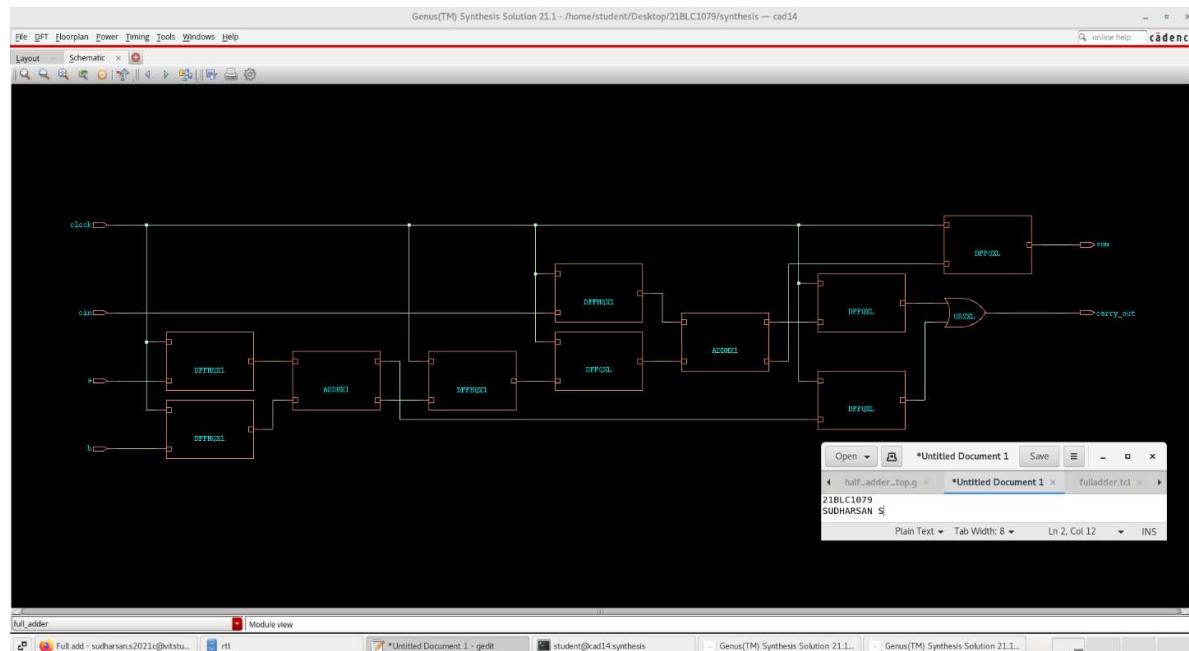
module full_adder (
    input  a,
    input  b,
    input  cin,
    input  clk,
    output reg sum,
    output reg carry_out
);

    always@(posedge clk)
    {carry_out,sum} = a+b+cin;

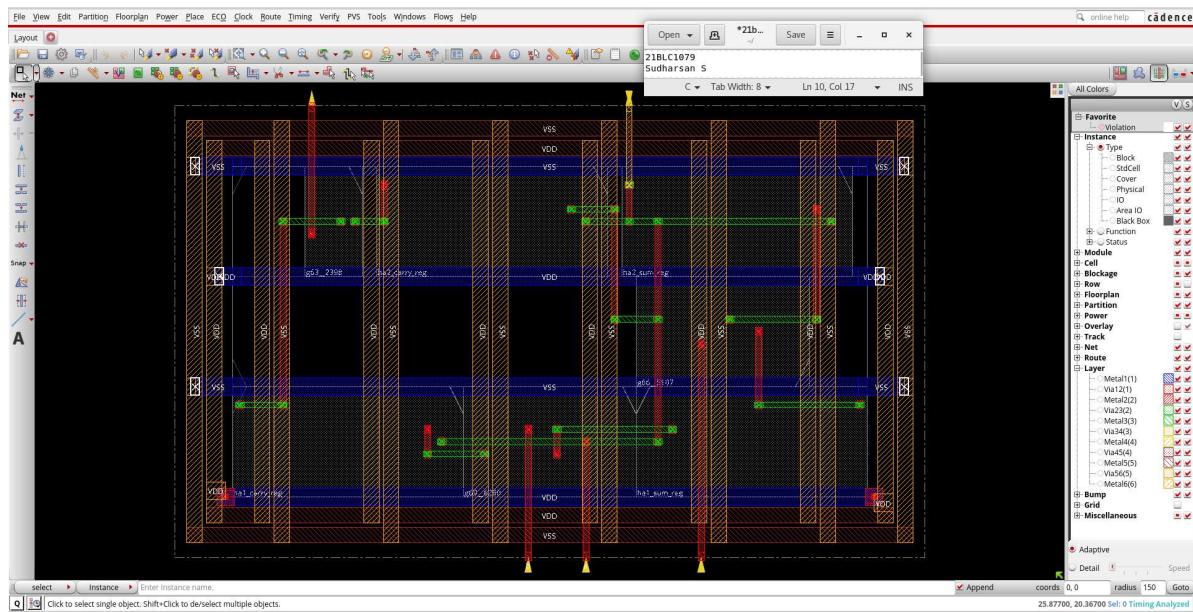
endmodule

```

GATE LEVEL NETLIST



PHYSICAL DESIGN



OBSERVATIONS

a) FULL ADDER_NETLIST_CREATED FILE

```

/*
#####
# Generated by:      Cadence Innovus 21.15-s110_1
# OS:               Linux x86_64(Host ID cad14)
# Generated on:     Thu Oct 24 14:44:16 2024
# Design:           full_adder
# Command:          saveNetlist full_adder.v
#####
*/
// Generated by Cadence Genus(TM) Synthesis Solution 21.14-s082_1
// Generated on: Jul 31 2024 13:21:02 IST (Jul 31 2024 07:51:02 UTC)
// Verification Directory fv/full_adder
module full_adder (
    a,
    b,
    cin,
    clock,
    sum,
    carry_out);
    input a;
    input b;
    input cin;
    input clock;
    output sum;
    output carry_out;

    // Internal wires
    wire c1;
    wire c2;
    wire ha1_d1;
    wire ha1_d3;
    wire ha2_d1;
    wire ha2_d3;
    wire n_0;
    wire n_1;
    wire n_2;
    wire n_3;
    wire s1;

    OR2XL g63_2398 (.A(c2),
                      .B(c1),
                      .Y(carry_out));
    DFFQXL ha2_df4_uut_Q_reg (.CK(clock),
                               .D(n_2),
                               .Q(c2));
    DFFQXL ha2_df3_uut_Q_reg (.CK(clock),
                               .D(n_3),
                               .Q(sum));
    ADDHX1 g66 5107 (.A(ha2_d3),
                      .B(n_0),
                      .C(n_1),
                      .D(n_3),
                      .E(n_2),
                      .F(n_1),
                      .G(n_0),
                      .H(sum),
                      .I(carry_out));

```

```

        .B(ha2_d1),
        .S(n_3),
        .CO(n_2));
DFFQXL ha2_df1_uut_Q_reg (.CK(clock),
    .D(s1),
    .Q(ha2_d1));
DFFQXL ha1_df4_uut_Q_reg (.CK(clock),
    .D(n_0),
    .Q(c1));
DFFHQX1 ha1_df3_uut_Q_reg (.D(n_1),
    .CK(clock),
    .Q(s1));
ADDHX1 g70_6260 (.A(ha1_d1),
    .B(ha1_d3),
    .S(n_1),
    .CO(n_0));
DFFHQX1 ha1_df2_uut_Q_reg (.D(b),
    .CK(clock),
    .Q(ha1_d3));
DFFHQX1 ha2_df2_uut_Q_reg (.D(cin),
    .CK(clock),
    .Q(ha2_d3));
DFFHQX1 ha1_df1_uut_Q_reg (.D(a),
    .CK(clock),
    .Q(ha1_d1));
endmodule

```

b) FULL ADDER _CONSTRAINT CREATED FILE

```

# #####
# Created by Genus(TM) Synthesis Solution 21.14-s082_1 on Mon Oct 21 13:38:15 IST 2024
# #####
set sdc_version 2.0

set_units -capacitance 1000ff
set_units -time 1000ps

# Set the current design
current_design full_adder

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 a]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports b]
set_input_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports cin]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports carry_out]
set_output_delay -clock [get_clocks clk] -add_delay -max 1.0 [get_ports sum]
set_wire_load_mode "enclosed"
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]

```

c) FULL ADDER _GATES CREATED FILE

```
// Generated by Cadence Genus(TM) Synthesis Solution 21.14-s082_1
// Generated on: Jul 31 2024 13:21:02 IST (Jul 31 2024 07:51:02 UTC)

// Verification Directory fv/full_adder

module full_adder(a, b, cin, clock, sum, carry_out);
    input a, b, cin, clock;
    output sum, carry_out;
    wire a, b, cin, clock;
    wire sum, carry_out;
    wire c1, c2, ha1_d1, ha1_d3, ha2_d1, ha2_d3, n_0, n_1;
    wire n_2, n_3, s1;
    OR2XL g63_2398(.A (c2), .B (c1), .Y (carry_out));
    DFFQXL ha2_df4_uut_Q_reg(.CK (clock), .D (n_2), .Q (c2));
    DFFQXL ha2_df3_uut_Q_reg(.CK (clock), .D (n_3), .Q (sum));
    ADDHX1 g66_5107(.A (ha2_d3), .B (ha2_d1), .CO (n_2), .S (n_3));
    DFFQXL ha2_df1_uut_Q_reg(.CK (clock), .D (s1), .Q (ha2_d1));
    DFFQXL ha1_df4_uut_Q_reg(.CK (clock), .D (n_0), .Q (c1));
    DFFHQX1 ha1_df3_uut_Q_reg(.CK (clock), .D (n_1), .Q (s1));
    ADDHX1 g70_6260(.A (ha1_d1), .B (ha1_d3), .CO (n_0), .S (n_1));
    DFFHQX1 ha1_df2_uut_Q_reg(.CK (clock), .D (b), .Q (ha1_d3));
    DFFHQX1 ha2_df2_uut_Q_reg(.CK (clock), .D (cin), .Q (ha2_d3));
    DFFHQX1 ha1_df1_uut_Q_reg(.CK (clock), .D (a), .Q (ha1_d1));
endmodule
```