THIẾT KẾ HỆ THỐNG SỐ VỚI HDL
BÀI THỰC HÀNH 5 ,6, 7

GVHD: Tạ Trí Đức

Sinh viên thực hiện: Phạm Quốc Tiến – 22521472

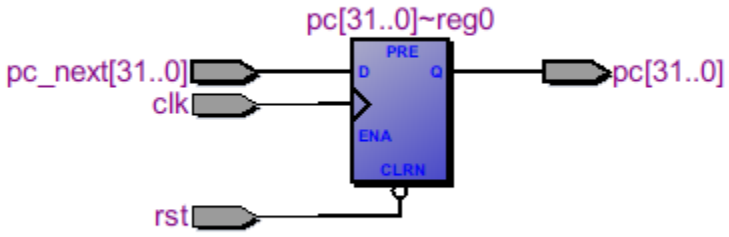
1) Thiết kế khối MIPS

a. Khối PC

Code:

```
1   module register
2   (
3       input clk,
4       input rst,
5       input [31:0] pc_next,
6       output reg [31:0] pc
7   );
8
9   always @(posedge clk or negedge rst)
10      if(~rst)
11          pc <= {32{1'b0}};
12      else
13          pc <= pc_next;
14  endmodule
```

RTL:

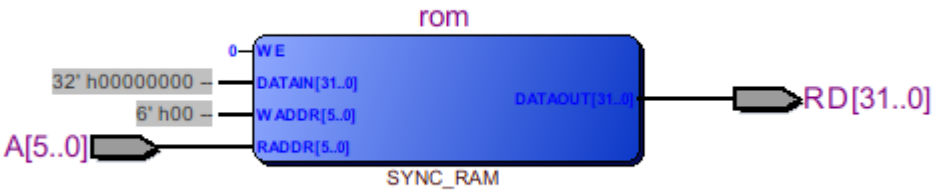


b. Khối I-MEM

Code:

```
1   module rom
2   (
3       input [5:0] A,
4       output [31:0] RD
5   );
6       reg [31:0] rom [63:0];
7       assign RD = rom[A];
8       initial begin
9           $readmemh("program.hex", rom);
10      end
11  endmodule
12
```

RTL:

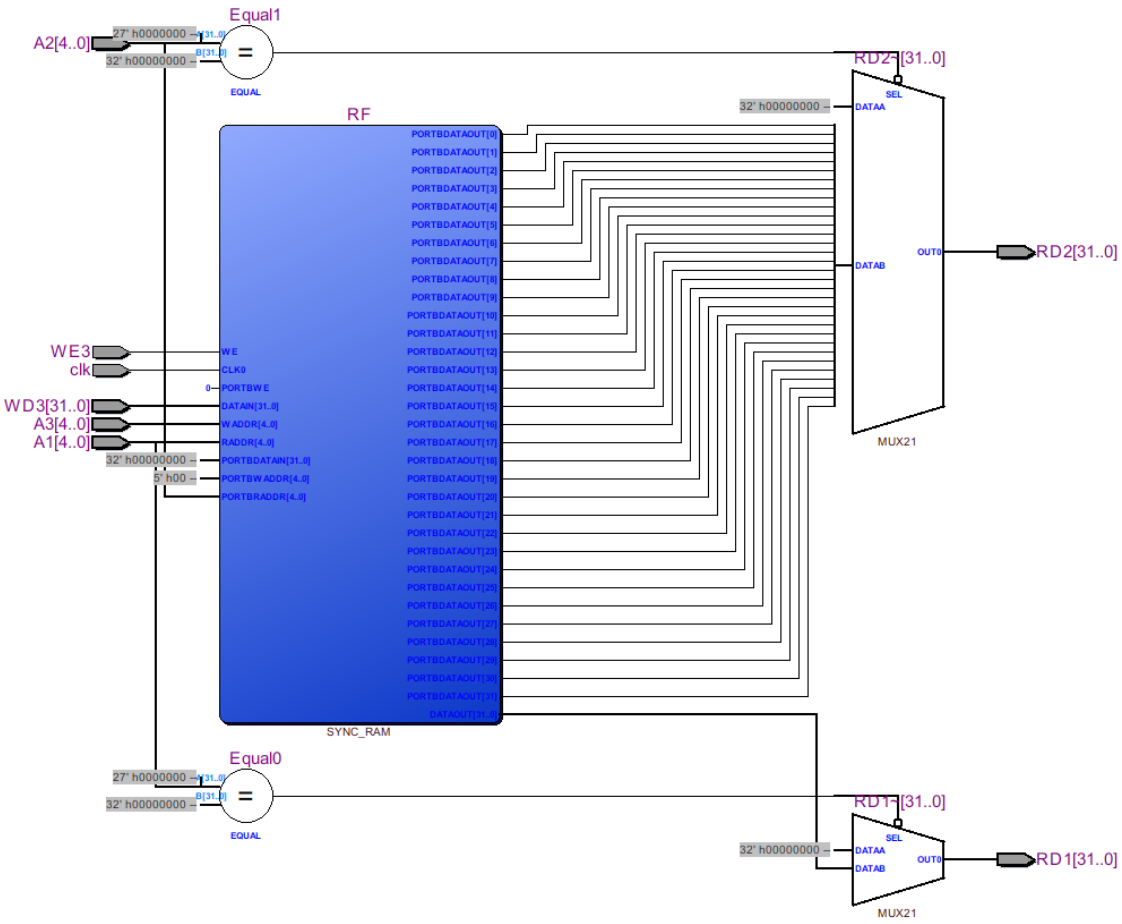


c. Khối Register File

Code:

```
1  module register_file
2  (
3      input clk,
4      input [4:0] A1,
5      input [4:0] A2,
6      input [4:0] A3,
7      input [31:0] WD3,
8      input WE3,
9      output [31:0] RD1,
10     output [31:0] RD2
11 );
12     reg [31:0] RF [31:0];
13     assign RD1 = (A1 != 0) ? RF[A1] : 32'b0;
14     assign RD2 = (A2 != 0) ? RF[A2] : 32'b0;
15     always @ (posedge clk)
16        if(WE3) RF[A3] <= WD3;
17 endmodule
18
```

RTL:

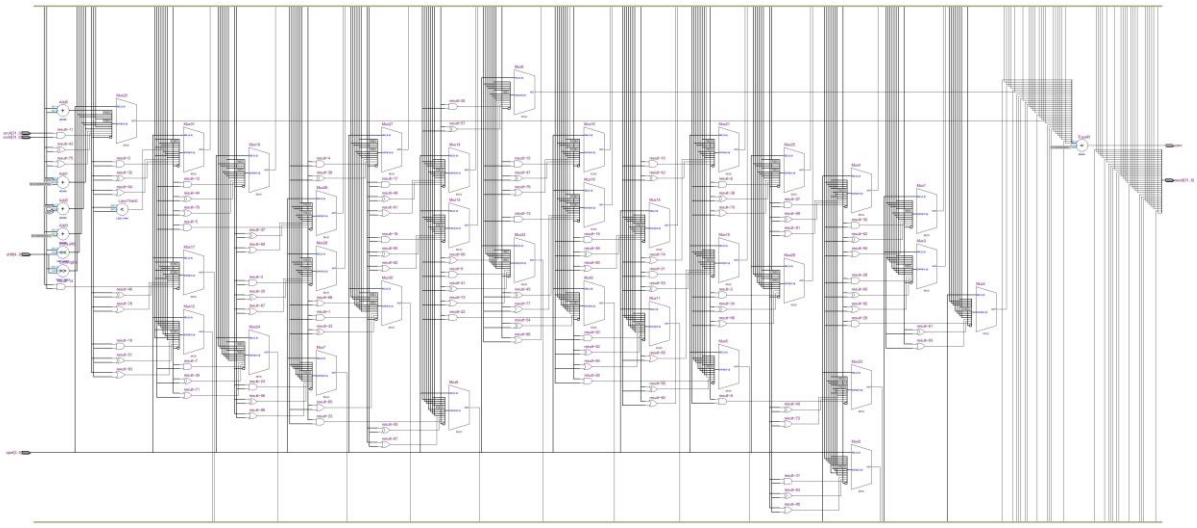


d. Khối ALU

Code:

```
1  module alu
2  (
3      input [31:0] srcA,
4      input [31:0] srcB,
5      input [3:0] oper,
6      input [4:0] shift,
7      output zero,
8      output reg [31:0] result
9  );
10 always @ (*) begin
11     case (oper)
12         default : result = srcA + srcB;
13         4'b0000 : result = ~srcA;
14         4'b0001 : result = srcA & srcB;
15         4'b0010 : result = srcA ^ srcB;
16         4'b0011 : result = srcA | srcB;
17         4'b0100 : result = srcA - 1;
18         4'b0101 : result = srcA + srcB;
19         4'b0110 : result = srcA - srcB;
20         4'b0111 : result = srcA + 1;
21         4'b1000 : result = (srcA < srcB) ? 1 : 0;
22         4'b1010 : result = srcB << shift;
23         4'b1011 : result = srcB >> shift;
24         4'b1100 : result = (srcB << 16);
25     endcase
26 end
27     assign zero = (result == 0);
28 endmodule
```

RTL:



e. Khối Control

Code:

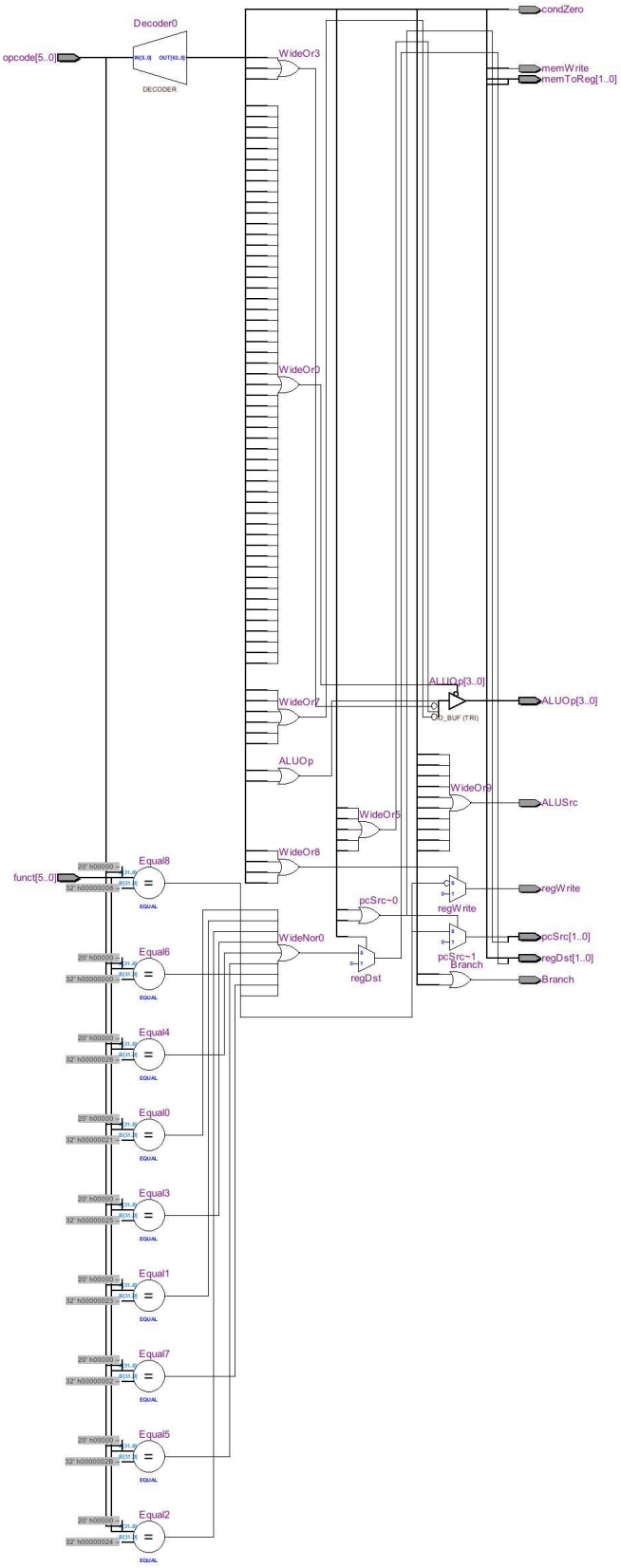
```
module control_unit
(
    input [5:0] opcode,
    input [5:0] funct,
    output reg condZero,
    output reg Branch,
    output reg regWrite,
    output reg [1:0] regDst,
    output reg ALUSrc,
    output reg [3:0] ALUOp,
    output reg [1:0] pcSrc,
    output reg memWrite,
    output reg [1:0] memToReg
);

always @(*) begin
    regDst <= 2'b0;
    regWrite <= 1;
    Branch <= 0;
    condZero <= 0;
    ALUSrc <= 0;
    memWrite <= 0;
    memToReg <= 2'b0;
    pcSrc <= 2'b0;
    case({opcode, funct})
        33: begin
            ALUOp <= 4'b0101;
            regDst <= 2'b01;
        end
        35: begin
            ALUOp <= 4'b0110;
            regDst <= 2'b01;
        end
```

```
34          36: begin
35              ALUOp <= 4'b0001;
36              regDst <= 2'b01;
37          end
38          37: begin
39              ALUOp <= 4'b0011;
40              regDst <= 2'b01;
41          end
42          38: begin
43              ALUOp <= 4'b0010;
44              regDst <= 2'b01;
45          end
46          43: begin
47              ALUOp <= 4'b1000;
48              regDst <= 2'b01;
49          end
50          0: begin
51              ALUOp <= 4'b1010;
52              regDst <= 2'b01;
53          end
54          2: begin
55              ALUOp <= 4'b1011;
56              regDst <= 2'b01;
57          end
58          8: begin
59              ALUOp <= 4'b0110;
60              regDst <= 2'b01;
61              regWrite <= 0;
62              pcSrc <= 2'b10;
63          end
64          default: ALUOp <= 4'bZ;
65      endcase
66      case(opcode)
```

```
67          9: begin
68              ALUOp <= 4'b0101;
69              ALUSrc <= 1;
70          end
71          12: begin
72              ALUOp <= 4'b0001;
73              ALUSrc <= 1;
74          end
75          13: begin
76              ALUOp <= 4'b0011;
77              ALUSrc <= 1;
78          end
79          11: begin
80              ALUOp <= 4'b1000;
81              ALUSrc <= 1;
82          end
83          15: begin
84              ALUOp <= 4'b1100;
85              ALUSrc <= 1;
86          end
87          4: begin
88              ALUOp <= 4'b0110;
89              regWrite <= 0;
90              Branch <= 1;
91              condZero <= 1;
92              ALUSrc <= 1;
93          end
94          5: begin
95              ALUOp <= 4'b0110;
96              regWrite <= 0;
97              Branch <= 1;
98              ALUSrc <= 1;
99          end
```

```
100         35: begin
101             ALUOp <= 4'b0101;
102             ALUSrc <= 1;
103             memToReg <= 2'b01;
104         end
105         43: begin
106             ALUOp <= 4'b0101;
107             regWrite <= 0;
108             ALUSrc <= 1;
109             memWrite <= 1;
110         end
111         2: begin
112             ALUOp <= 4'b0110;
113             regWrite <= 0;
114             pcSrc <= 2'b01;
115         end
116         3: begin
117             ALUOp <= 4'b0110;
118             regDst <= 2'b10;
119             ALUSrc <= 1;
120             memToReg <= 2'b10;
121             pcSrc <= 2'b01;
122         end
123         default: ALUOp <= 4'bZ;
124     endcase
125 end
126 endmodule
```

RTL:


condZero
Decoder0
opcode[5..0]
DECODER
WideOr3
memWrite
memToReg[1..0]
WideOr0
WideOr7
ALUOp[3..0]
ALUOp[3..0]
TRI
ALUOp
WideOr6
ALUSrc
WideOr5
WideOr8
Equal8
funct[5..0]
regWrite
pcSrc~0
regWrite
WideNor0
pcSrc[1..0]
regDst[1..0]
pcSrc~1
Branch
regDst
Branch
Equal6
Equal4
Equal0
Equal3
Equal1
Equal7
Equal5
Equal2
EQUAL

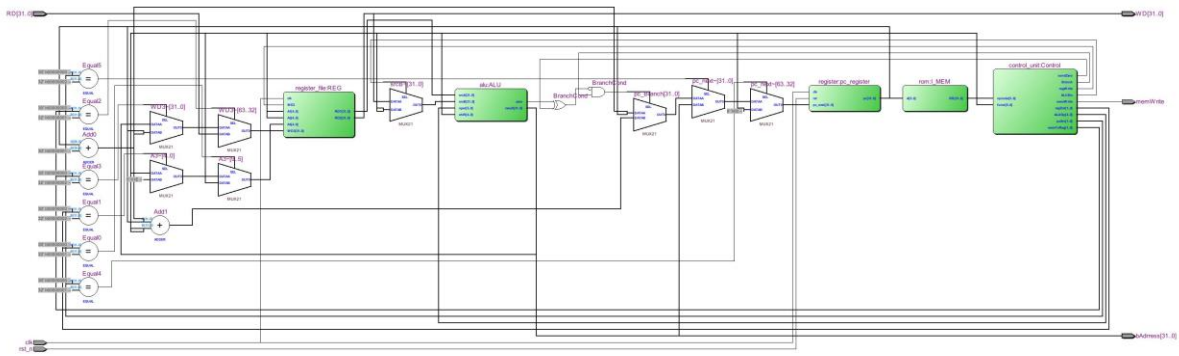
f. Khối MIPS

Code:

```
module mips
(
    input clk,
    input rst_n,
    input [31:0] RD,
    output [31:0] WD,
    output [31:0] bAdrress,
    output memWrite
);
    wire [31:0] pc_next, pc;
    wire [5:0] imAddr;
    wire [31:0] instruction;
    wire [5:0] opcode;
    wire [4:0] rs;
    wire [4:0] rt;
    wire [4:0] rd;
    wire [4:0] shamt;
    wire [5:0] funct;
    wire [4:0] A3;
    wire [31:0] WD3;
    wire [31:0] RD1;
    wire [31:0] RD2;
    wire [31:0] srcB;
    wire ALU_Zero;
    wire [31:0] ALU_Result;
    //wire [31:0] RD;
    wire [31:0] Sign_extend;
    wire [31:0] pcNext;
    wire [31:0] pcBranch;
    wire BranchCond;
    wire [31:0] pc_Branch;

    wire [1:0] regDst;
```

```
34      wire regWrite;
35      wire ALUSrc;
36      wire [3:0] ALUOp;
37      //wire memWrite;
38      wire [1:0] memToReg;
39      wire Branch;
40      wire condZero;
41      wire [1:0] pcSrc;
42
43      assign imAddr = pc >> 2;
44      assign opcode = instruction[31:26];
45      assign rs = instruction[25:21];
46      assign rt = instruction[20:16];
47      assign rd = instruction[15:11];
48      assign shamt = instruction[10:6];
49      assign funct = instruction[5:0];
50      assign A3 = (regDst == 1) ? rd : ((regDst == 2) ? 5'b1111 : rt);
51      assign WD3 = (memToReg == 1) ? RD : ((memToReg == 2) ? pcNext : ALU_Result);
52      assign srcB = (ALUSrc) ? Sign_extend : RD2;
53      assign Sign_extend = {{16 {instruction[15]}}, instruction[15:0]};
54      assign pcNext = pc + 4;
55      assign pcBranch = pcNext + Sign_extend;
56      assign BranchCond = Branch & (ALU_Zero == condZero);
57      assign pc_Branch = BranchCond ? pcBranch : pcNext;
58      assign pc_next = (pcSrc == 1) ? instruction[25:0] : ((pcSrc == 2) ? ALU_Result : pc_Branch);
59
60      assign bAdrress = ALU_Result;
61      assign WD = RD2;
62      register pc_register(clk, rst_n, pc_next, pc);
63      rom I_MEM(imAddr, instruction);
64      register_file REG(clk, rs, rt, A3, WD3, regWrite, RD1, RD2);
65      alu ALU(RD1, srcB, ALUOp, shamt, ALU_Zero, ALU_Result);
66      //ram D_MEM(clk, ALU_Result, memWrite, RD2, RD);
67      control_unit Control(opcode, funct, condZero, Branch, regWrite, regDst, ALUSrc, ALUOp, pcSrc,
68      memWrite, memToReg);
69
70
71
72  endmodule
```

RTL:



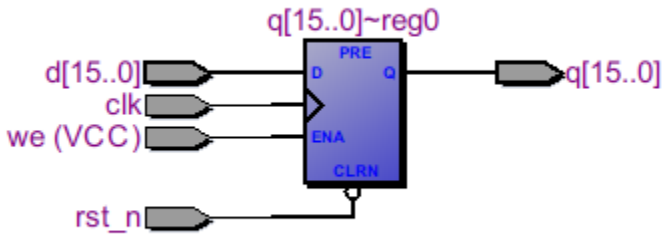
2) Thiết kế khối GPIO

a. Khối Register

Code:

```
1  module register_we
2  (
3      input clk,
4      input rst_n,
5      input we,
6      input [15:0] d,
7      output reg [15:0] q
8  );
9  always @ (posedge clk or negedge rst_n)
10     if(~rst_n)
11         q <= { 16 { 1'b0 } };
12     else
13         if(we) q <= d;
14 endmodule
```

RTL:

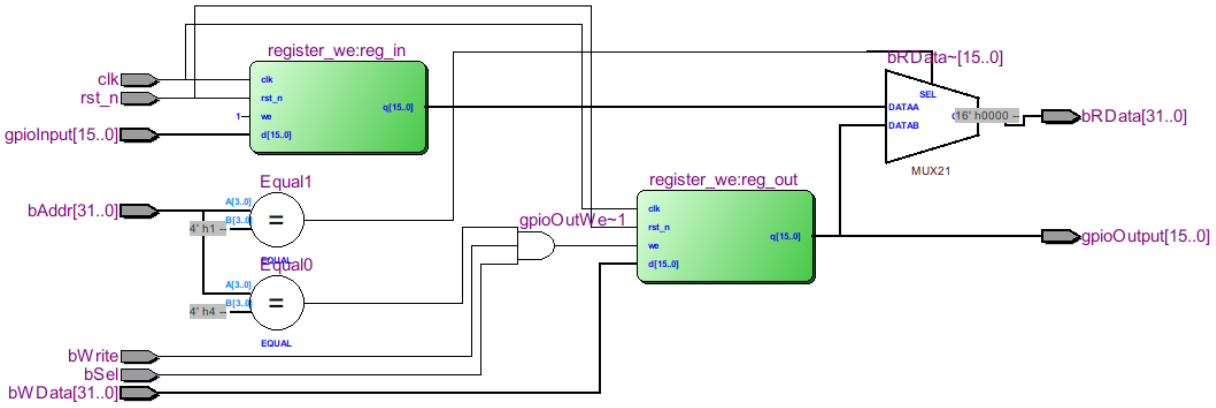


b. Khối GPIO

Code:

```
1  module gpio
2  (
3      input bWrite,
4      input bSel,
5      input clk,
6      input rst_n,
7      input [15:0] gpioInput,
8      input [31:0] bAddr,
9      input [31:0] bWData,
10     output reg [31:0] bRData,
11     output [15:0] gpioOutput
12 );
13     wire [16 - 1:0] gpioIn;
14     wire gpioOutWe;
15     wire [16 - 1:0] gpioOut;
16     assign gpioOut = bWData [16 - 1:0];
17     assign gpioOutWe = bSel & bWrite & (bAddr[3:0] == 4'h4);
18     register_we reg_in (clk, rst_n, 1, gpioInput, gpioIn);
19     register_we reg_out (clk, rst_n, gpioOutWe, gpioOut, gpioOutput);
20     always @ (*)
21        case(bAddr[3:0])
22           default : bRData = { { 16{1'b0}}, gpioIn };
23           4'h0 : bRData = { { 16{1'b0}}, gpioIn };
24           4'h1 : bRData = { { 16{1'b0}}, gpioOutput };
25        endcase
26 endmodule
```

RTL:

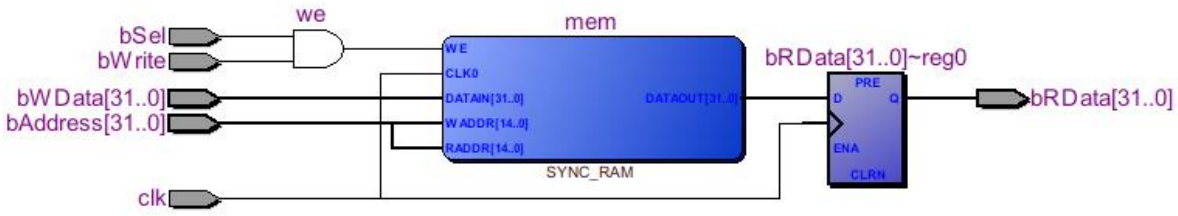


3) Thiết kế khối Memory IO

Code:

```
1   module IO_Mem
2   (
3       input clk,
4       input [31:0] bAddress,
5       input bSel,
6       input bWrite,
7       input [31:0] bWData,
8       output reg [31:0] bRData
9   );
10
11      reg [31:0] mem [16384:0];
12      wire we;
13      assign we = bWrite & bSel;
14      always @(posedge clk) begin
15          if(we) begin
16              mem[bAddress] <= bWData;
17          end
18          bRData <= mem[bAddress];
19      end
20      endmodule
```

RTL:



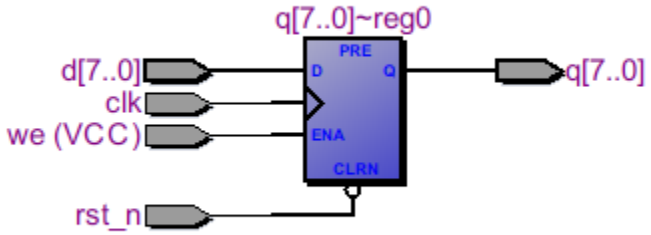
4) Thiết kế khối PWM

a. Khối Register

Code:

```
1   module register_we_0
2   (
3       input clk,
4       input rst_n,
5       input we,
6       input [7:0] d,
7       output reg [7:0] q
8   );
9   always @ (posedge clk or negedge rst_n)
10      if(~rst_n)
11          q <= { 8 { 1'b0 } };
12      else
13          if(we) q <= d;
14  endmodule
```

RTL:

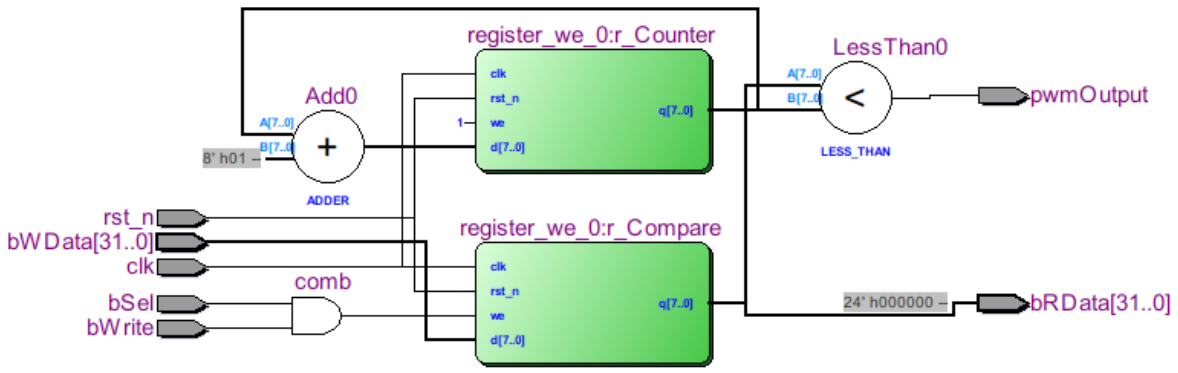


b. Khối PWM

Code:

```
1  module pwm
2  (
3      input bSel,
4      input bWrite,
5      input clk,
6      input rst_n,
7      input [31:0] bWData,
8      output [31:0] bRData,
9      output pwmOutput
10 );
11 wire [7:0] comp;
12 wire [7:0] count;
13 wire [7:0] countNext = count + 1;
14 assign bRData = { { 24 { 1'b0 }}, comp };
15 wire compWe = bSel & bWrite;
16 assign pwmOutput = (count > comp);
17 register_we_0 r_Compare(clk, rst_n, compWe, bWData[7:0], comp);
18 register_we_0 r_Counter (clk, rst_n, 1, countNext, count);
19 endmodule
```

RTL:



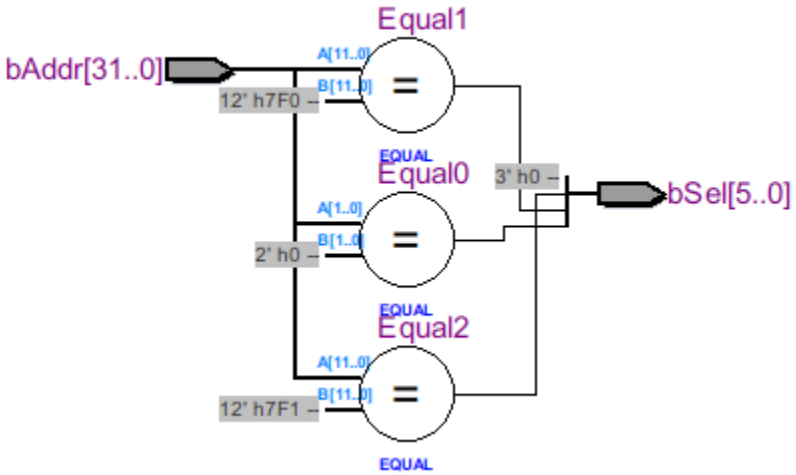
5) Thiết kế BUS

a. Khối Decoder

Code:

```
1  module bus_decoder
2  (
3      input [31:0] bAddr,
4      output [5:0] bSel
5  );
6      assign bSel[0] = ( bAddr [ 15:14 ] == 2'b00);
7      assign bSel[1] = ( bAddr [ 15:4 ] == 12'h7f0);
8      assign bSel[2] = ( bAddr [ 15:4 ] == 12'h7f1);
9      assign bSel[3] = 1'b0;
10     assign bSel[4] = 1'b0;
11     assign bSel[5] = 1'b0;
12 endmodule
```

RTL:

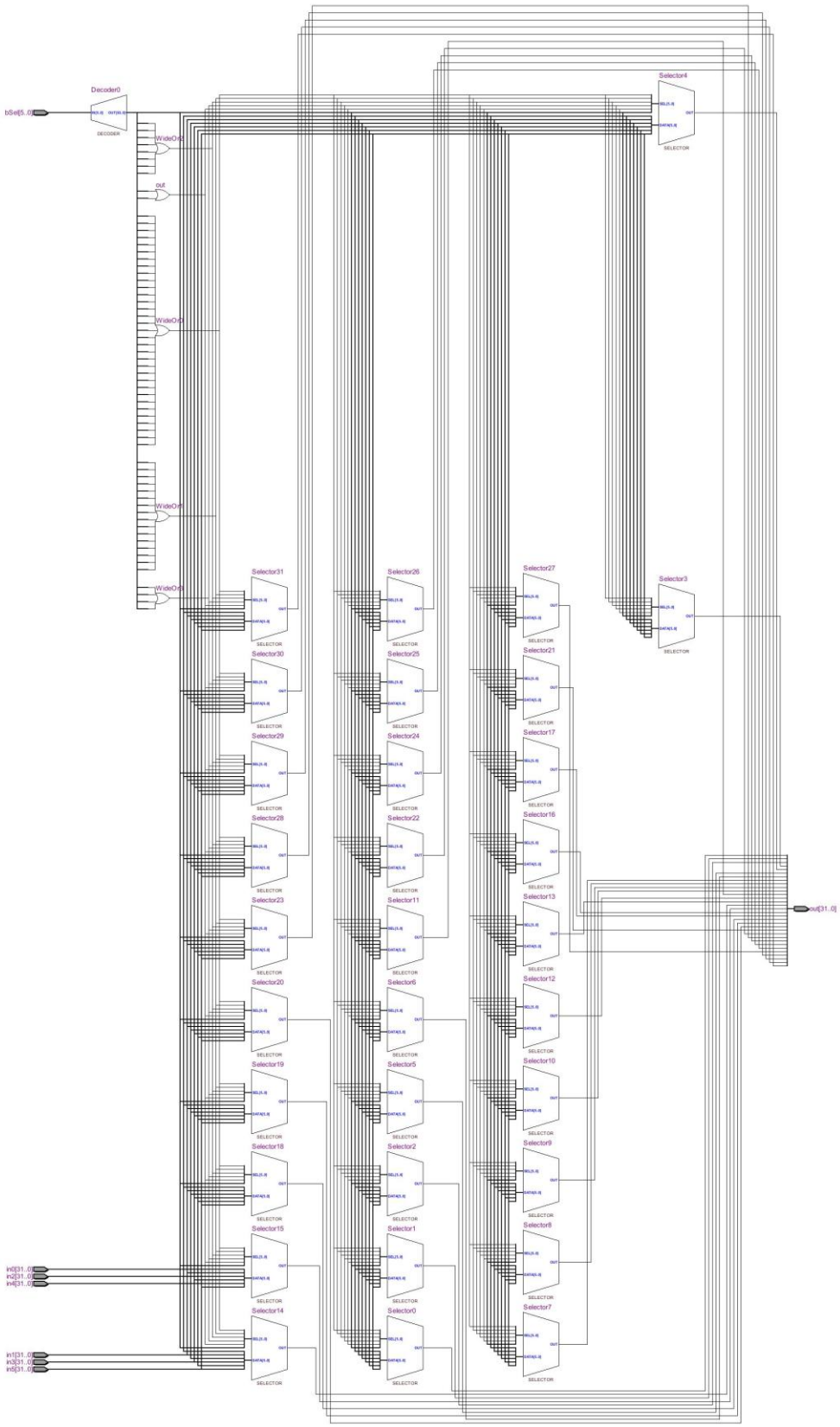


b. Khối Mux

Code:

```
1   module bus_mux
2   (
3       input [5:0] bSel,
4       input [31:0] in0,
5       input [31:0] in1,
6       input [31:0] in2,
7       input [31:0] in3,
8       input [31:0] in4,
9       input [31:0] in5,
10      output reg [31:0] out
11  );
12  always @*
13      casez (bSel)
14          default : out = in0;
15          6'b?????1 : out = in0;
16          6'b????10 : out = in1;
17          6'b???100 : out = in2;
18          6'b??1000 : out = in3;
19          6'b?10000 : out = in4;
20          6'b100000 : out = in5;
21      endcase
22  endmodule
```

RTL:

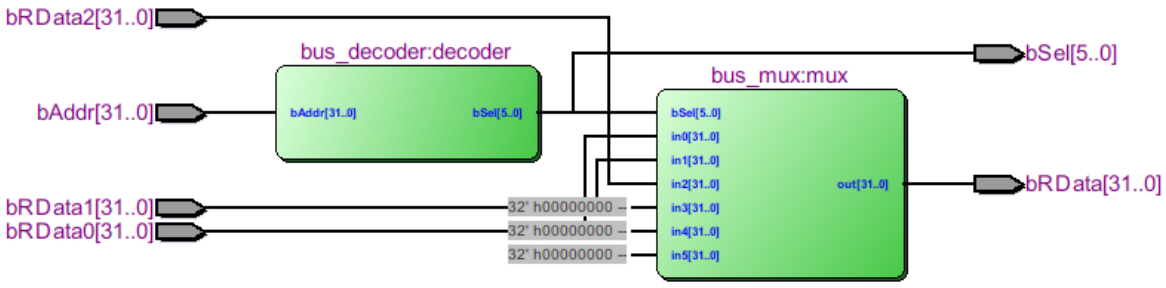

Decoder0
bSel[5..0]
DECODER
WideOr0
out
WideOr1
WideOr2
WideOr3
Selector4
Selector3
Selector31
Selector30
Selector29
Selector28
Selector23
Selector20
Selector19
Selector18
Selector15
Selector14
Selector26
Selector25
Selector24
Selector22
Selector11
Selector6
Selector5
Selector2
Selector1
Selector0
Selector27
Selector21
Selector17
Selector16
Selector13
Selector12
Selector10
Selector9
Selector8
Selector7
SELECTOR
in0[31..0]
in2[31..0]
in4[31..0]
in1[31..0]
in3[31..0]
in5[31..0]
out[31..0]

c. BUS

Code:

```
1   module bus
2   (
3       input [31:0] bAddr,
4       input [31:0] bRData0,
5       input [31:0] bRData1,
6       input [31:0] bRData2,
7       output [31:0] bRData,
8       output [5:0] bSel
9   );
10  bus_decoder decoder
11  (
12      .bAddr ( bAddr ),
13      .bSel ( bSel )
14  );
15  bus_mux mux
16  (
17      .bSel ( bSel ),
18      .out ( bRData ),
19      .in0 ( bRData0 ),
20      .in1 ( bRData1 ),
21      .in2 ( bRData2 ),
22      .in3 ( 32'b0 ),
23      .in4 ( 32'b0 ),
24      .in5 ( 32'b0 )
25  );
26  endmodule
```

RTL:

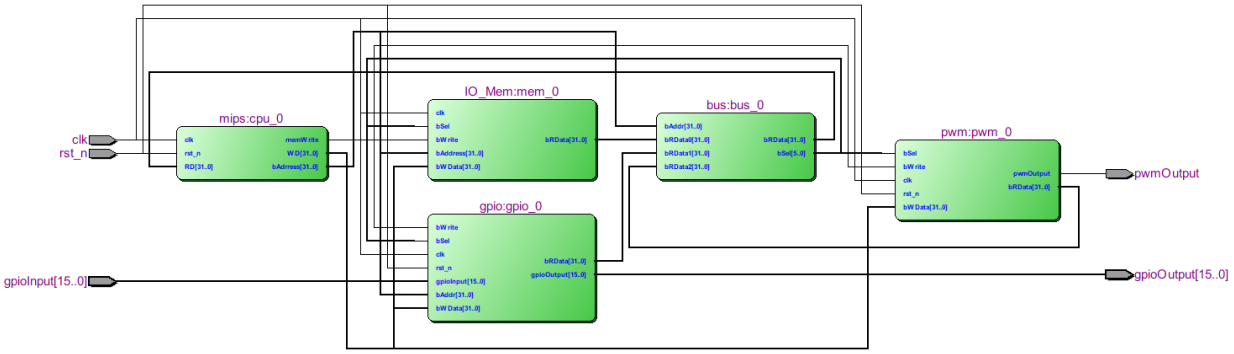


6) Thiết kế SOC

Code:

```
module SOC
(
    input clk,
    input rst_n,
    input [15:0] gpioInput,
    output [15:0] gpioOutput,
    output pwmOutput
);
    wire [31:0] bAddress;
    wire [31:0] bRData0;
    wire [31:0] bRData1;
    wire [31:0] bRData2;
    wire [31:0] bRData;
    wire [31:0] bWData;
    wire [5:0] bSel;
    wire bWrite;

    mips cpu_0(clk, rst_n, bRData, bWData, bAddress, bWrite);
    bus bus_0(bAddress, bRData0, bRData1, bRData2, bRData, bSel);
    IO_Mem mem_0(clk, bAddress, bSel[0], bWrite, bWData, bRData0);
    gpio gpio_0(bWrite, bSel[1], clk, rst_n, gpioInput, bAddress, bWData, bRData1, gpioOutput);
    pwm pwm_0(bSel[2], bWrite, clk, rst_n, bWData, bRData2, pwmOutput);

endmodule //2 chu kỳ xong 1 lệnh
```

RTL:



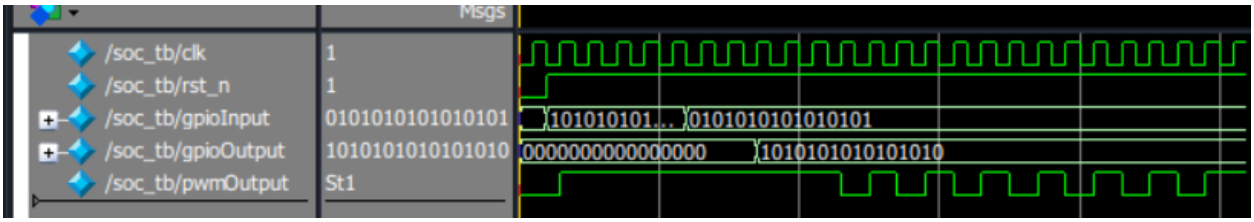
Code Assembly:

```
addiu $t0, $zero, 100
addiu $t1, $zero, 0x00000004
sw $t0, ($t1)
lw $t2, ($t1)
addiu $t3, $zero, 0x00007f00
lw $t4, ($t3)
addiu $t5, $zero, 0x00007f04
sw $t4, ($t5)
addiu $t6, $zero, 0x00007f15
addiu $t7, $zero, 11
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
addiu $t7, $t7, 2
sw $t7, ($t6)
```

Testbench:

```
1   module soc_tb();
2
3      reg clk;
4      reg rst_n;
5      reg [15:0] gpioInput;
6      wire [15:0] gpioOutput;
7      wire pwmOutput;
8
9      SOC blabla(clk, rst_n, gpioInput, gpioOutput, pwmOutput);
10
11     initial begin
12        clk = 0;
13        rst_n = 1;
14        gpioInput = 16'b0;
15        forever #5 clk = ~clk;
16     end
17
18     initial begin
19          rst_n = 0;
20          #10 rst_n = 1;
21          gpioInput = 16'hAAAA;
22
23          #50;
24
25          gpioInput = 16'h5555;
26
27          #200;
28
29          $stop;
30      end
31  endmodule
```

Sóng mô phỏng:



7) Report

a. Resource

Analysis & Synthesis Resource Usage Summary

	Resource	Usage
1	Estimated Total logic elements	853
2		
3	Total combinational functions	589
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	493
2	-- 3 input functions	54
3	-- <=2 input functions	42
5		
6	Logic elements by mode	
1	-- normal mode	557
2	-- arithmetic mode	32
7		
8	Total registers	311
1	-- Dedicated logic registers	311
2	-- I/O registers	0
9		
10	I/O pins	35
11	Total memory bits	262160
12	Embedded Multiplier 9-bit elements	0
13	Maximum fan-out node	clk
14	Maximum fan-out	391
15	Total fan-out	4178
16	Average fan-out	4.12

b. Timing

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	62.48 MHz	62.48 MHz	clock	

Slow Model Setup: 'clock'

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	3.995	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.016	16.025
2	4.036	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~292	clock	clock	20.000	-0.014	15.986
3	4.048	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~7	clock	clock	20.000	-0.002	15.986
4	4.052	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~135	clock	clock	20.000	-0.002	15.982
5	4.058	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~356	clock	clock	20.000	-0.022	15.956
6	4.060	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~260	clock	clock	20.000	-0.022	15.954
7	4.072	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~462	clock	clock	20.000	-0.009	15.955
8	4.072	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~398	clock	clock	20.000	-0.009	15.955
9	4.073	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~1	clock	clock	20.000	0.002	15.965
10	4.073	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.017	15.946
11	4.076	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~129	clock	clock	20.000	0.002	15.962
12	4.077	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~366	clock	clock	20.000	-0.003	15.956
13	4.079	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~110	clock	clock	20.000	-0.003	15.954
14	4.113	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.016	15.907
15	4.114	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~292	clock	clock	20.000	-0.015	15.907
16	4.122	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.016	15.898
17	4.126	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~7	clock	clock	20.000	-0.003	15.907
18	4.130	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~135	clock	clock	20.000	-0.003	15.903
19	4.133	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~97	clock	clock	20.000	-0.009	15.894
20	4.136	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~356	clock	clock	20.000	-0.023	15.877
21	4.137	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~225	clock	clock	20.000	-0.009	15.890
22	4.138	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~260	clock	clock	20.000	-0.023	15.875
23	4.150	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~462	clock	clock	20.000	-0.010	15.876
24	4.150	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~398	clock	clock	20.000	-0.010	15.876
25	4.151	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~1	clock	clock	20.000	0.001	15.886
26	4.154	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~292	clock	clock	20.000	-0.014	15.868
27	4.154	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~129	clock	clock	20.000	0.001	15.883
28	4.155	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~366	clock	clock	20.000	-0.004	15.877
29	4.157	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~481	clock	clock	20.000	-0.006	15.873
30	4.157	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~110	clock	clock	20.000	-0.004	15.875
31	4.163	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~292	clock	clock	20.000	-0.014	15.859
32	4.166	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~7	clock	clock	20.000	-0.002	15.868
33	4.167	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.016	15.853
	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
34	4.170	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~135	clock	clock	20.000	-0.002	15.864
35	4.175	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~7	clock	clock	20.000	-0.002	15.859
36	4.176	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~356	clock	clock	20.000	-0.022	15.838
37	4.178	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~132	clock	clock	20.000	0.002	15.860
38	4.178	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~260	clock	clock	20.000	-0.022	15.836
39	4.179	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~135	clock	clock	20.000	-0.002	15.855
40	4.185	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~356	clock	clock	20.000	-0.022	15.829
41	4.187	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~260	clock	clock	20.000	-0.022	15.827
42	4.190	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~462	clock	clock	20.000	-0.009	15.837
43	4.190	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~398	clock	clock	20.000	-0.009	15.837
44	4.191	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~1	clock	clock	20.000	0.002	15.847
45	4.194	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~129	clock	clock	20.000	0.002	15.844
46	4.195	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~366	clock	clock	20.000	-0.003	15.838
47	4.196	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~136	clock	clock	20.000	0.002	15.842
48	4.196	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~8	clock	clock	20.000	0.002	15.842
49	4.197	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~110	clock	clock	20.000	-0.003	15.836
50	4.199	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~462	clock	clock	20.000	-0.009	15.828
51	4.199	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~398	clock	clock	20.000	-0.009	15.828
52	4.200	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~1	clock	clock	20.000	0.002	15.838
53	4.203	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~129	clock	clock	20.000	0.002	15.835
54	4.204	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~366	clock	clock	20.000	-0.003	15.829
55	4.206	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~110	clock	clock	20.000	-0.003	15.827
56	4.208	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~292	clock	clock	20.000	-0.014	15.814
57	4.211	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~97	clock	clock	20.000	-0.010	15.815
58	4.215	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~225	clock	clock	20.000	-0.010	15.811
59	4.220	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~7	clock	clock	20.000	-0.002	15.814
60	4.224	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~135	clock	clock	20.000	-0.002	15.810
61	4.225	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~3	clock	clock	20.000	0.002	15.813
62	4.230	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~356	clock	clock	20.000	-0.022	15.784
63	4.232	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~260	clock	clock	20.000	-0.022	15.782
64	4.235	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~481	clock	clock	20.000	-0.007	15.794
65	4.244	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~462	clock	clock	20.000	-0.009	15.783
66	4.244	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~398	clock	clock	20.000	-0.009	15.783

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
67	4.245	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~1	clock	clock	20.000	0.002	15.793
68	4.247	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~103	clock	clock	20.000	-0.014	15.775
69	4.248	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~129	clock	clock	20.000	0.002	15.790
70	4.249	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~366	clock	clock	20.000	-0.003	15.784
71	4.251	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~110	clock	clock	20.000	-0.003	15.782
72	4.251	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~97	clock	clock	20.000	-0.009	15.776
73	4.255	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~270	clock	clock	20.000	-0.022	15.759
74	4.255	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~225	clock	clock	20.000	-0.009	15.772
75	4.256	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~417	clock	clock	20.000	-0.013	15.767
76	4.256	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~132	clock	clock	20.000	0.001	15.781
77	4.259	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~140	clock	clock	20.000	0.002	15.779
78	4.260	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~97	clock	clock	20.000	-0.009	15.767
79	4.264	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~225	clock	clock	20.000	-0.009	15.763
80	4.265	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~268	clock	clock	20.000	-0.022	15.749
81	4.267	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~263	clock	clock	20.000	-0.008	15.761
82	4.267	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~364	clock	clock	20.000	-0.022	15.747
83	4.268	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~359	clock	clock	20.000	-0.008	15.760
84	4.272	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~67	clock	clock	20.000	-0.027	15.737
85	4.274	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~136	clock	clock	20.000	0.001	15.763
86	4.274	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~8	clock	clock	20.000	0.001	15.763
87	4.275	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~483	clock	clock	20.000	-0.027	15.734
88	4.275	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~481	clock	clock	20.000	-0.006	15.755
89	4.284	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~481	clock	clock	20.000	-0.006	15.746
90	4.286	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~46	clock	clock	20.000	-0.013	15.737
91	4.296	mips:cpu_0|register:pc_register|pc[7]	mips:cpu_0|register_file:REG|RF~132	clock	clock	20.000	0.002	15.742
92	4.303	pwm:pwm_0|register_we_0:r_Counter|q[1]	mips:cpu_0|register_file:REG|RF~3	clock	clock	20.000	0.001	15.734
93	4.305	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~289	clock	clock	20.000	-0.014	15.717
94	4.305	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~33	clock	clock	20.000	-0.014	15.717
95	4.305	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~97	clock	clock	20.000	-0.009	15.722
96	4.305	mips:cpu_0|register:pc_register|pc[5]	mips:cpu_0|register_file:REG|RF~132	clock	clock	20.000	0.002	15.733
97	4.306	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~238	clock	clock	20.000	-0.009	15.721
98	4.309	mips:cpu_0|register:pc_register|pc[4]	mips:cpu_0|register_file:REG|RF~225	clock	clock	20.000	-0.009	15.718
99	4.312	pwm:pwm_0|register_we_0:r_Counter|q[0]	mips:cpu_0|register_file:REG|RF~494	clock	clock	20.000	-0.016	15.708
100	4.313	mips:cpu_0|register:pc_register|pc[6]	mips:cpu_0|register_file:REG|RF~167	clock	clock	20.000	-0.019	15.704

c. Power

PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Fri Jun 13 18:00:54 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Bai5_6_7
Top-level Entity Name	SOC
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	183.22 mW
Core Dynamic Thermal Power Dissipation	61.36 mW
Core Static Thermal Power Dissipation	80.17 mW
I/O Thermal Power Dissipation	41.69 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data