VLSI Design ece 5470 Class Project Spring 2012

TOPIC : Designing a floating point unit using IEEE Starndars 754.

by

Divya Desiraju (A01624337)

Tasnuva Tarannum Tithi (A01631527)

Contents:

TOPIC:

Designing a floating point unit using IEEE Starndars 754.

Motivation:

The aim of the project is to design an ALU that is capable of doing following operations:

- Floating point addition
- Subtraction
- Multiplication
- Division
- Over flow
- Underflow
- And rounding.

This floating point unit will be designed using IEEE Standard 754. The base of the project is to design this system in Verilog. We will use Modelsim tool for developing and testing this logic design. Synthesis, placement and routing of this design will be done using the flows learned in ECE-6470 class (Synopsys Design Compiler and Cadence Encounter).

Inroduction to VLSI generations:

In the early days of VLSI, only a few transistors could be placed on a chip, as the scale used was large because of the contemporary technology, and manufacturing yields were low by today's standards. As the degree of integration was small, the design was done easily. The first integrated circuits contained only a few transistors. Called "**small-scale integration**" (**SSI**), digital circuits containing transistors numbering in the tens provided a few logic gates. The next step in the development of integrated circuits, taken in the late 1960s, introduced devices which contained hundreds of transistors on each chip, called "**medium-scale integration**" (**MSI**).Next came the "**large-scale integration**" (**LSI**) in the mid-1970s, with tens of thousands of transistors per chip. The final step in the development process, starting in the 1980s and continuing through the present, was "very large-scale integration" (VLSI). The development started with hundreds of thousands of transistors in the early 1980s, and continues beyond several billion transistors as of 2009.

Moore's law is a rule of thumb in the history of computing hardware. It predicted that number of transistor will double in every 18 months. This prediction has proved to be uncannily accurate, in part because the law is now used in the semiconductor industry to guide long-term planning and to set targets for research and development.

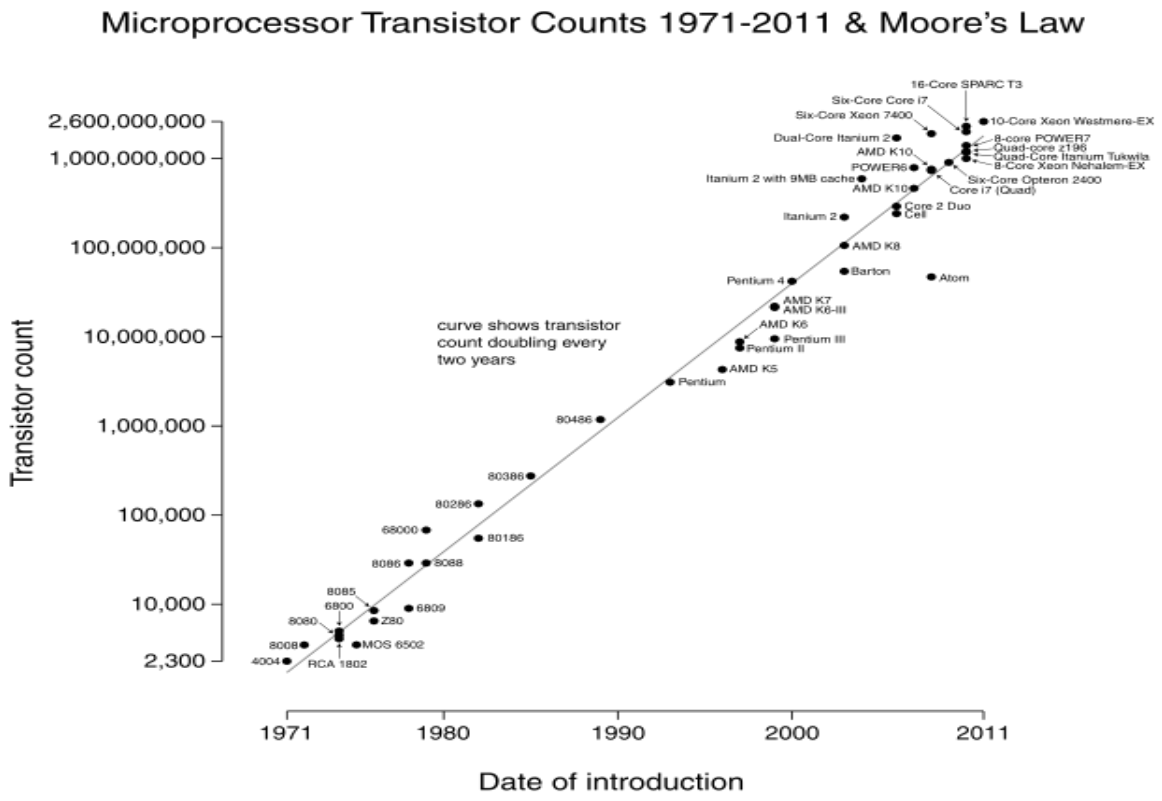

Fig1: Moore's law time vs number of transistors graph

CAD TOOLS:

As the number of transistors increased over the years, manual design is not possible anymore. To design a complex electronic circuit consisting of millions of transistor designers use CAD (Computer aided design) tools. EDA (Electronic Design automation) refers to the use of CAD tools to create complex electronic designs

Companies that specialize in EDA are:

- Cadence ®Design Systems
- Magma ®Design Automation Inc.

- Synopsys®

ADVANTAGES OF CAD TOOLS:

- Ability to evaluate complex conditions where solving one problem creates another one
- Use analytical method to assess the cost of a decision
- Use synthesis methods to provide a solution
- Allows the process of proposing and analyzing the results at the same time

VLSI Design Flow:

Designing a computing unit is a sequential process that consists of several steps.

Steps prior to circuit/physical design are part the Front-end flow. This involves:

- System specification
- Functional design
- Functional verification
- Logic design
- Logic verification

Physical level design is called the Back-end Flow. Physical level design is also called placement and Routing.

This involves:

- Circuit Design
- Circuit Verification
- Physical Design
- Layout Verification

The design flow can be explained with the schematic below:

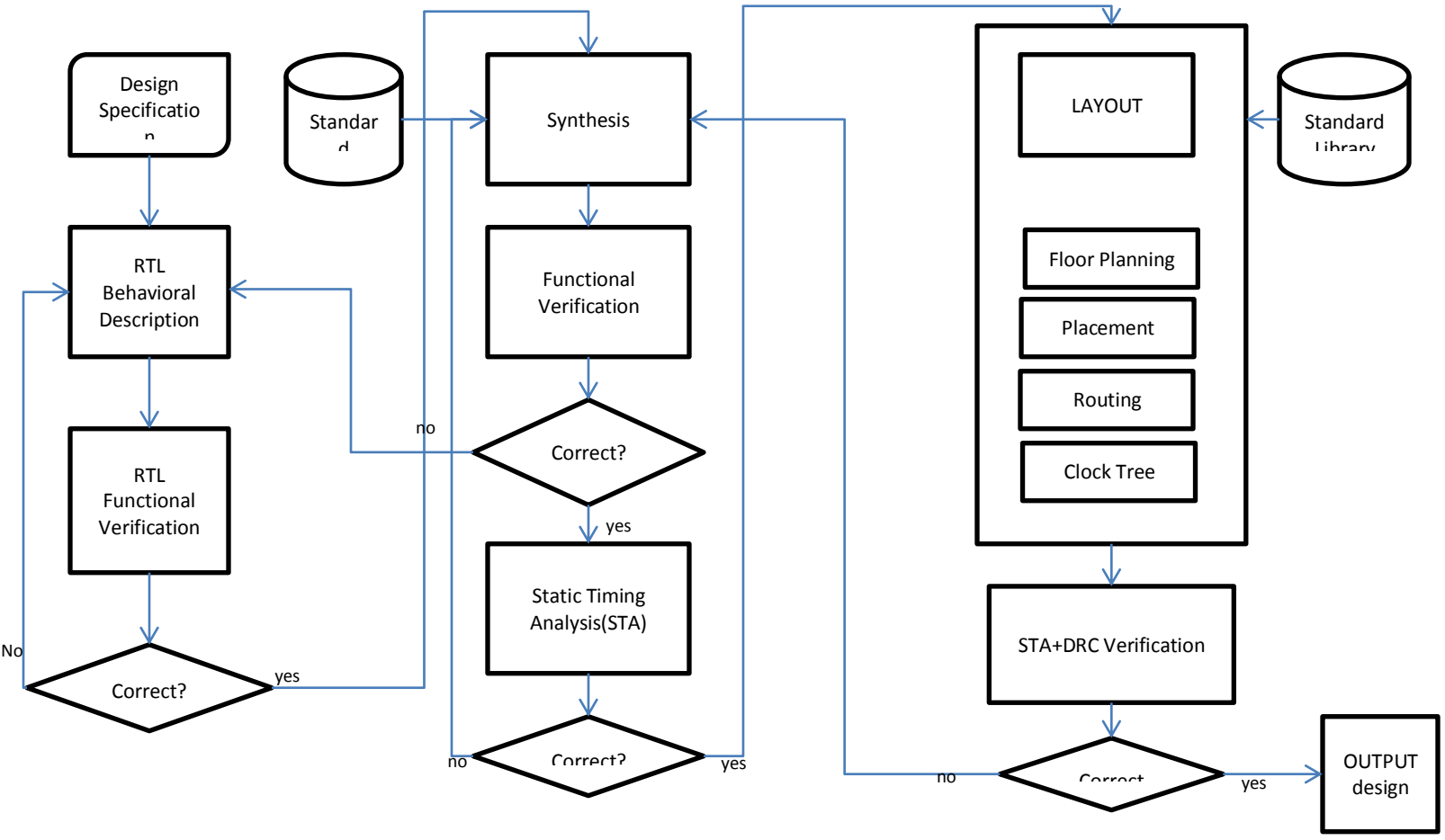

Fig2: VLSI Design Flow

Description of the FLOATING POINT UNIT:

The floating point unit consists of four submodules:

1. Adder
2. Subtractor
3. Multiplier
4. Divider

The operation of this floating point unit is selected through a multiplexer.

Following exceptions are also considered:

1. SNAN
2. QNAN
3. Zero
4. Infinity
5. Over flow
6. Under flow

Behavioral Description:

The behavior of the IC is written in Verilog HDL. Modelsim is used in this project to write the verilog description. All the Verilog description is included in the appendix.

We have following modules in our floating point unit:

Top module : It is the top level module of the design. It has 5 sub modules:

1. mux : the multiplexer

2. fpadder3: The adder. It has following sub modules:
    a) com
    b) comparator
    c) shiftreg
    d) subtractor

3. fpsubtractor: The subtractor. It has following sub modules:
    a) com
    b) comparator
    c) shiftreg
    d) subtractor

4. final_div: The divider. It has 1 submodule " simple_division"

5. fpmultiplier: the multiplier.

A block diagram of the floating point unit helps to visualize how the design is done:

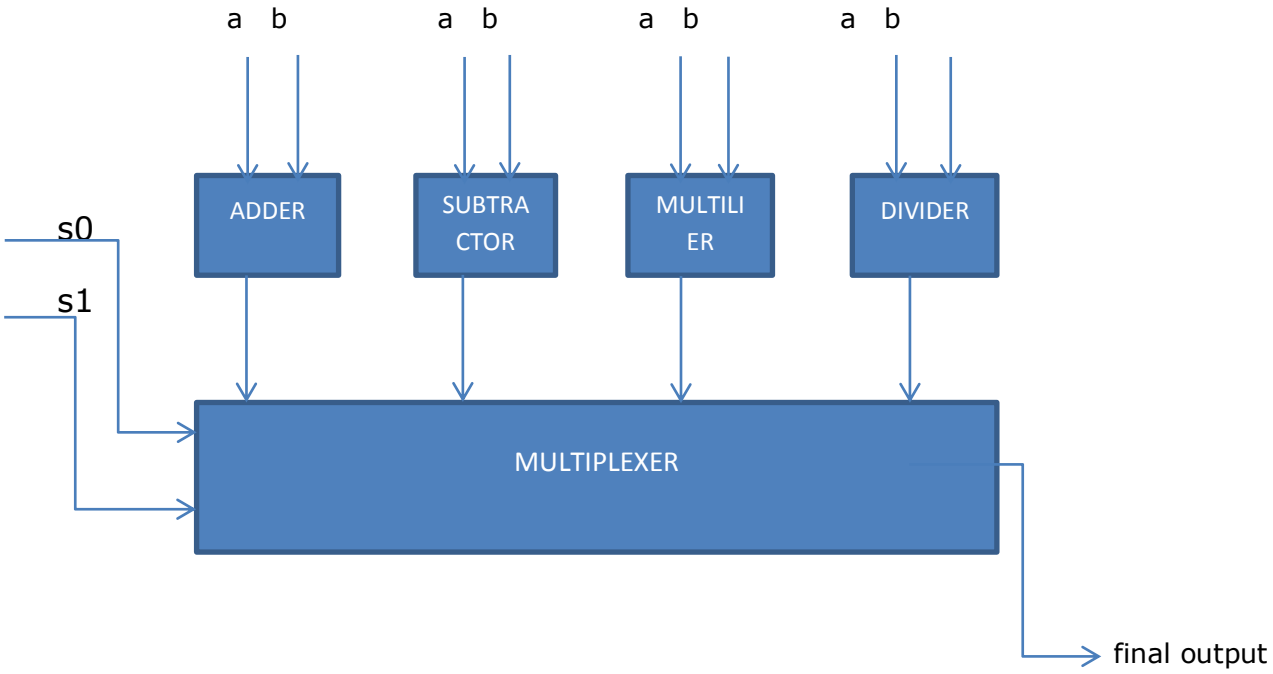

Fig3: Block Diagram of the Floating point unit.

Here we include the simulation results for following inputs:

Module Name	Input1 (a)	Input2 (b)	Output
Adder	44B3EE66	45D24041	45FF3BAD
Subtractor	41400000	C2600000	C2300000
Divider	46500000	49900000	3C38E38E
Multiplier	41233333	42340000	431F0000

Here is the Modelsim simulation for these general cases:

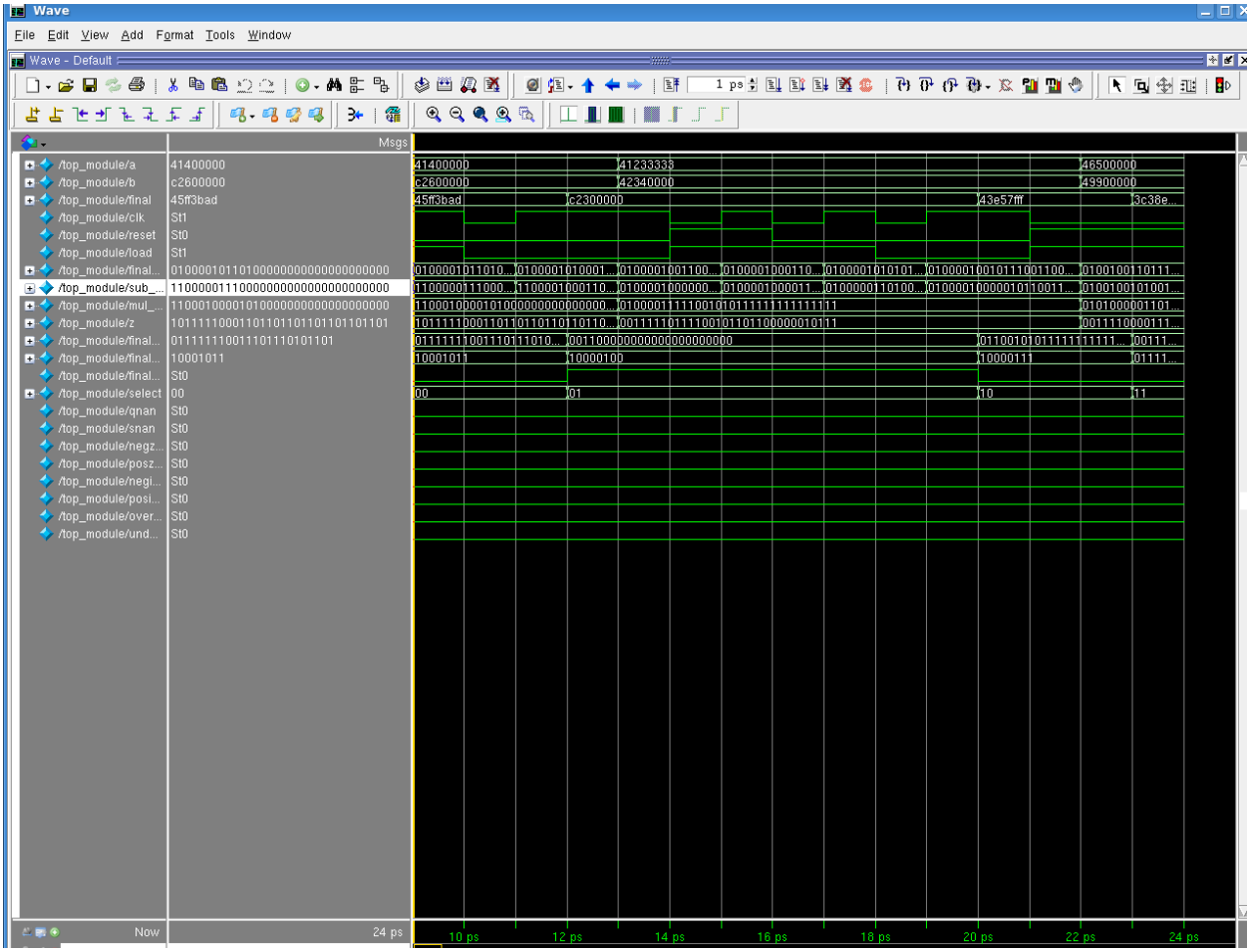

Fig4: simulation results for general cases

Here is the simulation waveform for exceptions:

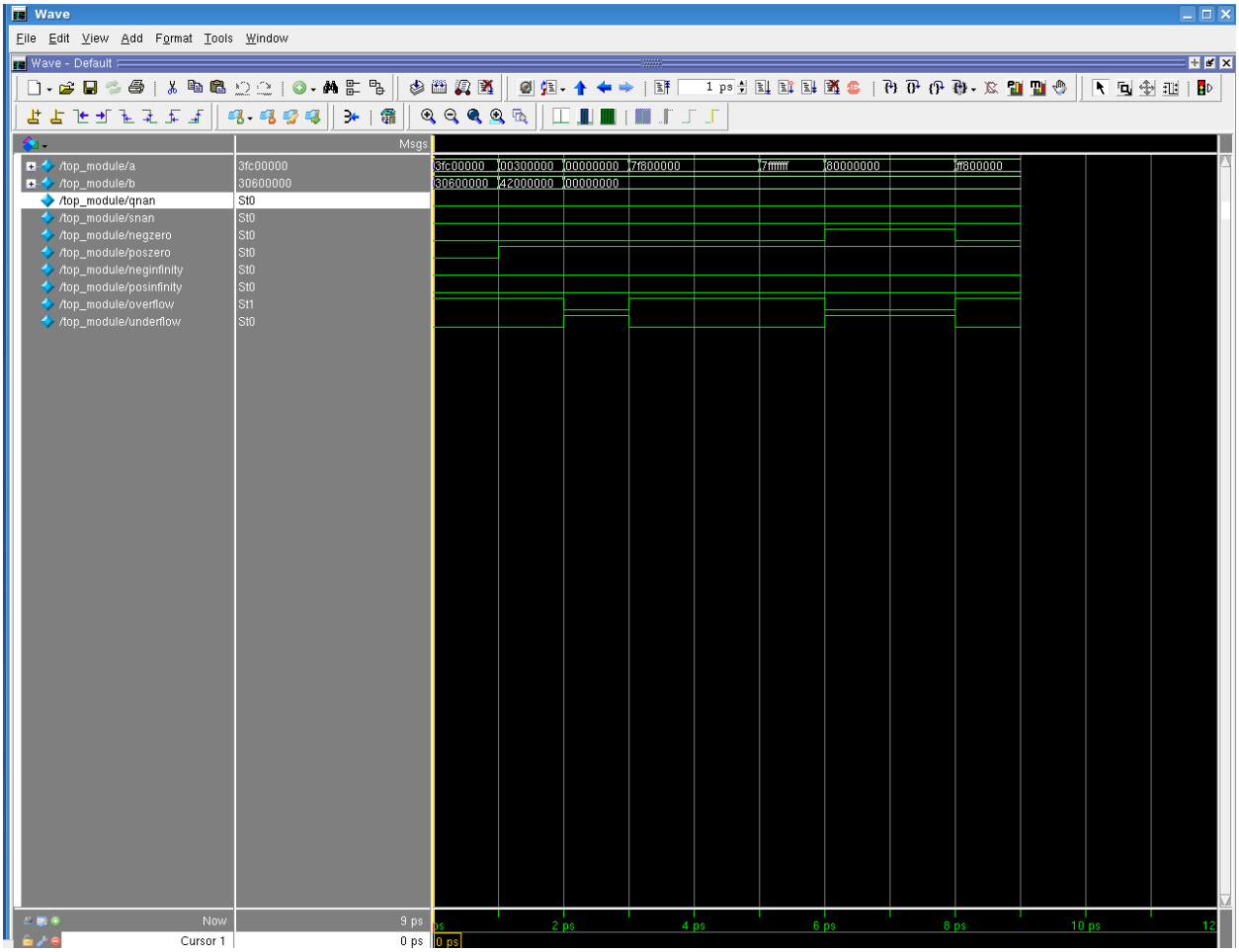

Fig5: simulation for exceptions.

Synthesis:

Synthesis converts the verilog description into design implementation in terms of standard logic gates. The dc_shell tool is used for this purpose.

For the synthesis we use the Verilog descripition of all the modules we have and the dc.tcl and env.tcl file in the working directory. Here is a brief description of the dc.tcl and env.tcl file and the timing, area and power constraint for the design.

- *dc.tcl* **:** This is the script that contains the "TCL commands" for doing the synthesis using dc_shell tool.

- *env.tcl* **:** This file sets up the different path variables that will be used later on to read the design data (e.g., location of design files, gate libraries).

- *Standard cell library* : We specify the standard cell library that is used in the floating point unit inside the dc.tcl. The standard cell library used is the *farady 90nm standard cell library*.

- *Wire load model:* We need a wire load model during synthesis we do not have exact estimates of the wire capacitances as the design is not placed/routed. Wire load models provide the estimates of the capacitance of the interconnects based on the number of fan-outs. We have used the *TSMC64_lowk_conservative* wire load modelfor our synthesis.

- *Timing constraints*: Timing constraint is provided to the tool in the form of clock period. Synthesis is done to ensure that this timing constraint is satisfied. This is specified in the dc.tcl script.

  For our design we have used a clock period of 12 ns which is specified in the env.tcl file.

  We also introduced clock uncertainty of 500 ps which is specified in the env.tcl file.

- *Power constraints:* We set maximum leakage power to zero, and maximum dynamic power to zero. As the synthesis tool attempts to optimize the power according to the constraint, we set it to a minimum value of zero. For dynamic power calculation we have introduced switching activity with a toggle rate of 0.01 and static probability of 0.1 for all inputs.

- *Area constraints:* We set maximum area to zero. As the synthesis tool attempts to optimize the area according to the constraint, we set it to a minimum value of zero.

- *Some additional constraints:* To prevent the feedthrough from input to output and to avoid assign statements we use the following command: *Set_fix_multiple_port_nets –all – buffer_constants [get design].*

The synthesis gives the following outputs in the output directory:

- Area report
- Check design
- Check timing
- Latch report
- Power report
- Threshold voltage
- Timing report
- Violator report
- A Verilog descriptipn of the input gate level mapped netlist (top_module_netlist_synopsys.v)
- A timing constraint file in .sdc format. (top_module.sdc)

Analysis of synthesis reports:

(Here we mention the area, power and timing reports.All the reports are included in the appendix.)

1)Area report:

```
Number of ports:                          107
Number of nets:                           730
Number of cells:                          508
Number of combinational cells:            504
Number of sequential cells:                 0
Number of macros:                           0
Number of buf/inv:                         12
Number of references:                      11

Combinational area:       22757.951883
Noncombinational area:     319.872002
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:          23077.823884
Total area:                 undefined
```

2)Power report:

```
Cell Internal Power  =  16.5197 uW   (52%)
Net Switching Power  =  15.0415 uW   (48%)
```

```
                          ---------
Total Dynamic Power    =   31.5612 uW  (100%)

Cell Leakage Power     =   31.0927 uW
```

3)Timing report:

```
max_delay   12.00

data required time  12.00
data arrival time   -7.17
  -----------------------------------------------------------------------
slack (MET)          4.83
```

Placement:

Placement is done using Soc Encounter. Placement requires :

1.The Configuration(conf) file (top_module.conf): It lists all other input design files along with their location.. Along with this, the file can be used to set some global parameters for the tool. The important entries in this file that must be specified for each design are:

- location of the input gate level netlist
- location of timing constraint (.sdc)file
- location of the timing library
- location of the LEF file
- setting the name of the top level module

2. The timing constraint (top_module.sdc): The timing constraint (.sdc) file is generated during synthesis

3.The netlist (top_module_netlist_synopsys.v):The netlist file generated during synthesis is the input gate level mapped net list to be placed and routed.This is a verilog file.

4.The placement script (place.tcl) works with encounter tool. It lists the basic commands that constitute the back end flow in the encounter tool. The cut sequence we used is HHHVVV.

Generated Files after placement**:** The following files are generated after placement:

1.top_module.placed.enc : This is directory generated after the placement is done where all the files for design import, floorplan, placement, routing etc are saved.

2.placed.def :This file contains specified information about floorplan, placement, routing anf netlist

3.prectsTiming reports: All the timing reports are saved here.

4.preplaceTiming reports: All the preplace timing reports are save here.

The encounter.log file gives following information:

Total CPU time : 1.67 sec

Total real time: 2.0 sec

Total memory usage: 281.5 Mbytes.

The net length list is also found in the encounter.log file.

```
Total net length = 4.741e+04 (2.231e+04 2.510e+04)

Avg net length = 1.458e+01 (sigma = 2.825e+01)
Sqrt of avg square net length = 3.179e+01

Avg connection length = 7.509e+00 (sigma = 4.922e+00)
Sqrt of avg square connection length = 8.978e+00

Net and connection length distribution:

[length   range   ]      #net #connection
[0.00e+00 5.60e+00]:     1154     1420
[5.60e+00 1.12e+01]:     1000     1082
[1.12e+01 1.68e+01]:      479      442
[1.68e+01 2.24e+01]:      255      196
[2.24e+01 2.80e+01]:      104       69
[2.80e+01 3.36e+01]:       55       26
[3.36e+01 3.92e+01]:       23       11
[3.92e+01 4.48e+01]:       23        3
[4.48e+01 5.04e+01]:       17        2
[5.04e+01 5.60e+01]:       13        1
[5.60e+01 6.16e+01]:        8        0
[6.16e+01 6.72e+01]:        5        0
[6.72e+01 7.28e+01]:       10        0
[7.28e+01 7.84e+01]:        9        0
```

```
[7.84e+01 8.40e+01]:        6         0
[8.40e+01 8.96e+01]:        7         0
[8.96e+01 9.52e+01]:        5         0
[9.52e+01 1.01e+02]:        3         0
[1.01e+02 1.06e+02]:        4         0
[1.06e+02 1.12e+02]:        3         0
[1.12e+02 1.18e+02]:        2         0
[1.18e+02 1.23e+02]:        7         0
[1.23e+02 1.29e+02]:        2         0
[1.29e+02 1.34e+02]:        2         0
[1.34e+02 1.40e+02]:        5         0
[1.40e+02 1.46e+02]:        3         0
[1.46e+02 1.51e+02]:        6         0
[1.51e+02 1.57e+02]:        6         0
[1.57e+02 1.62e+02]:        3         0
[1.62e+02 1.68e+02]:        5         0
[1.68e+02 1.74e+02]:        3         0
[1.74e+02 1.79e+02]:        1         0
[1.79e+02 1.85e+02]:        2         0
[1.90e+02 1.96e+02]:        2         0
[1.96e+02 2.02e+02]:        2         0
[2.02e+02 2.07e+02]:        2         0
[2.13e+02 2.18e+02]:        1         0
[2.18e+02 2.24e+02]:        1         0
[2.24e+02 2.30e+02]:        1         0
[2.30e+02 2.35e+02]:        1         0
[2.35e+02 2.41e+02]:        1         0
[2.46e+02 2.52e+02]:        1         0
[2.58e+02 2.63e+02]:        1         0
[2.63e+02 2.69e+02]:        1         0
[2.74e+02 2.80e+02]:        1         0
[2.80e+02 2.86e+02]:        1         0
[2.86e+02 2.91e+02]:        3         0
[2.91e+02 2.97e+02]:        3         0


Total number of long connections = 0
Io: nrCon=0
```

Here is a snapshot of the placement design:

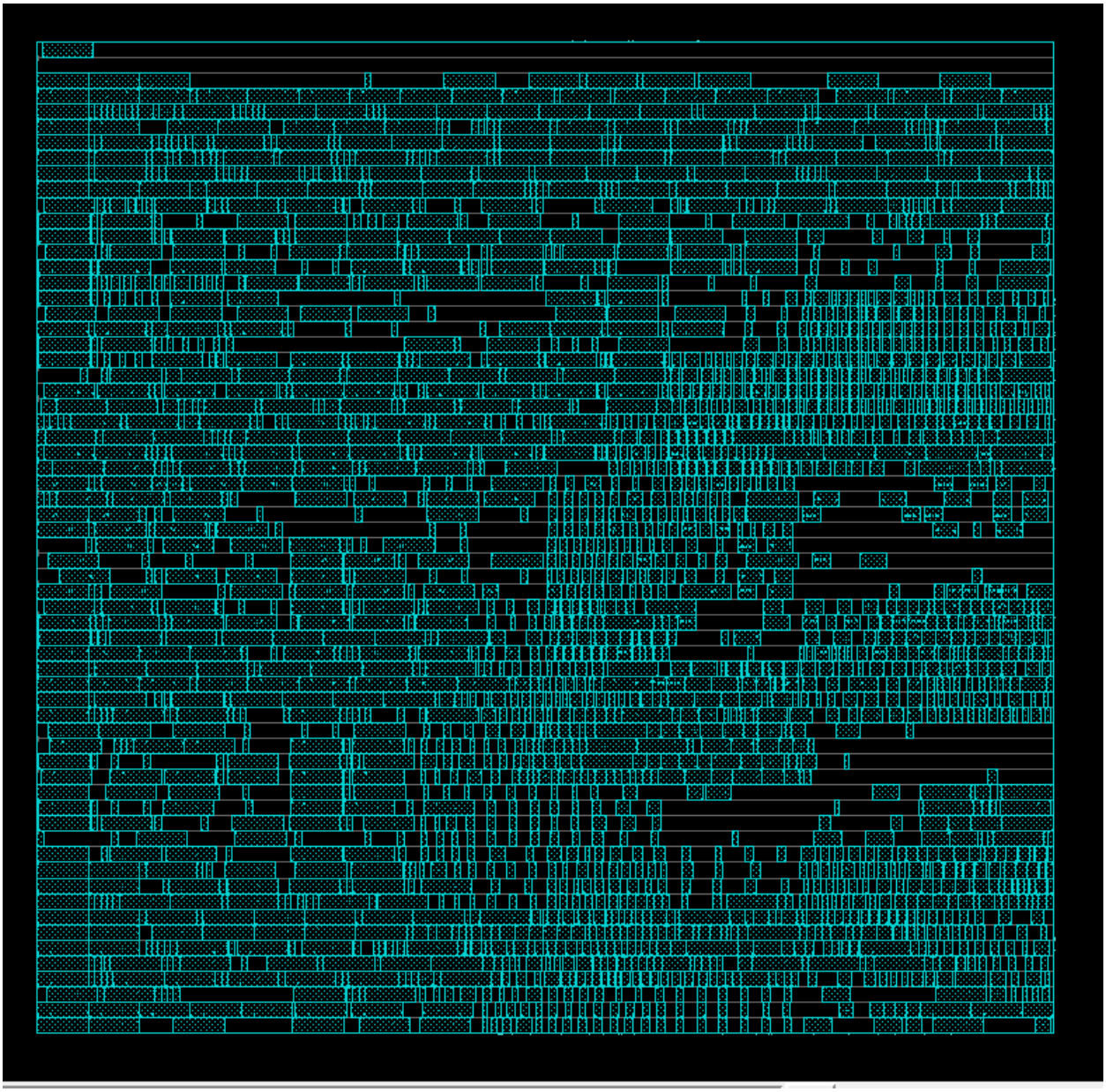

Fig6: Design after Placement

Routing:

Routing is done using Soc Encounter. Routing requires :

The design directory that contain all the input design data needed for placement of the design. The files present in this folder are the configuration file (top_module.conf), the timing constraint file (top_module.sdc) and netlist (top_module _netlist_synopsys.v) file

1. The Output directory of the placement (top_module.placed.enc.dat)
2. The DEF file (top_module.def)
3. Andfially the routing script (route.tcl)

Global routing plans the interconnect by breaking the routing portion of the design into rectangles called global routing cells (gcells) and assigning the signal nets to the gcells.

Detail Routing uses the NanoRoute router to perform routing on the entire design.

Complete detail routing:

```
#Complete Detail Routing.
#Total wire length = 51979 um.
#Total half perimeter of net bounding box = 53654 um.
#Total wire length on LAYER metal1 = 439 um.
#Total wire length on LAYER metal2 = 21969 um.
#Total wire length on LAYER metal3 = 22715 um.
#Total wire length on LAYER metal4 = 6136 um.
#Total wire length on LAYER metal5 = 698 um.
#Total wire length on LAYER metal6 = 23 um.
#Total wire length on LAYER metal7 = 0 um.
#Total wire length on LAYER metal8 = 0 um.
#Total number of vias = 18726
```

Complete global routing:

```
#Complete Global Routing.
#Total wire length = 51764 um.
#Total half perimeter of net bounding box = 53654 um.
#Total wire length on LAYER metal1 = 192 um.
#Total wire length on LAYER metal2 = 23714 um.
#Total wire length on LAYER metal3 = 22875 um.
#Total wire length on LAYER metal4 = 4356 um.
#Total wire length on LAYER metal5 = 614 um.
#Total wire length on LAYER metal6 = 13 um.
```

```
#Total wire length on LAYER metal7 = 0 um.
#Total wire length on LAYER metal8 = 0 um.
#Total number of vias = 12749
```

Here is the snapshot of the routed design:

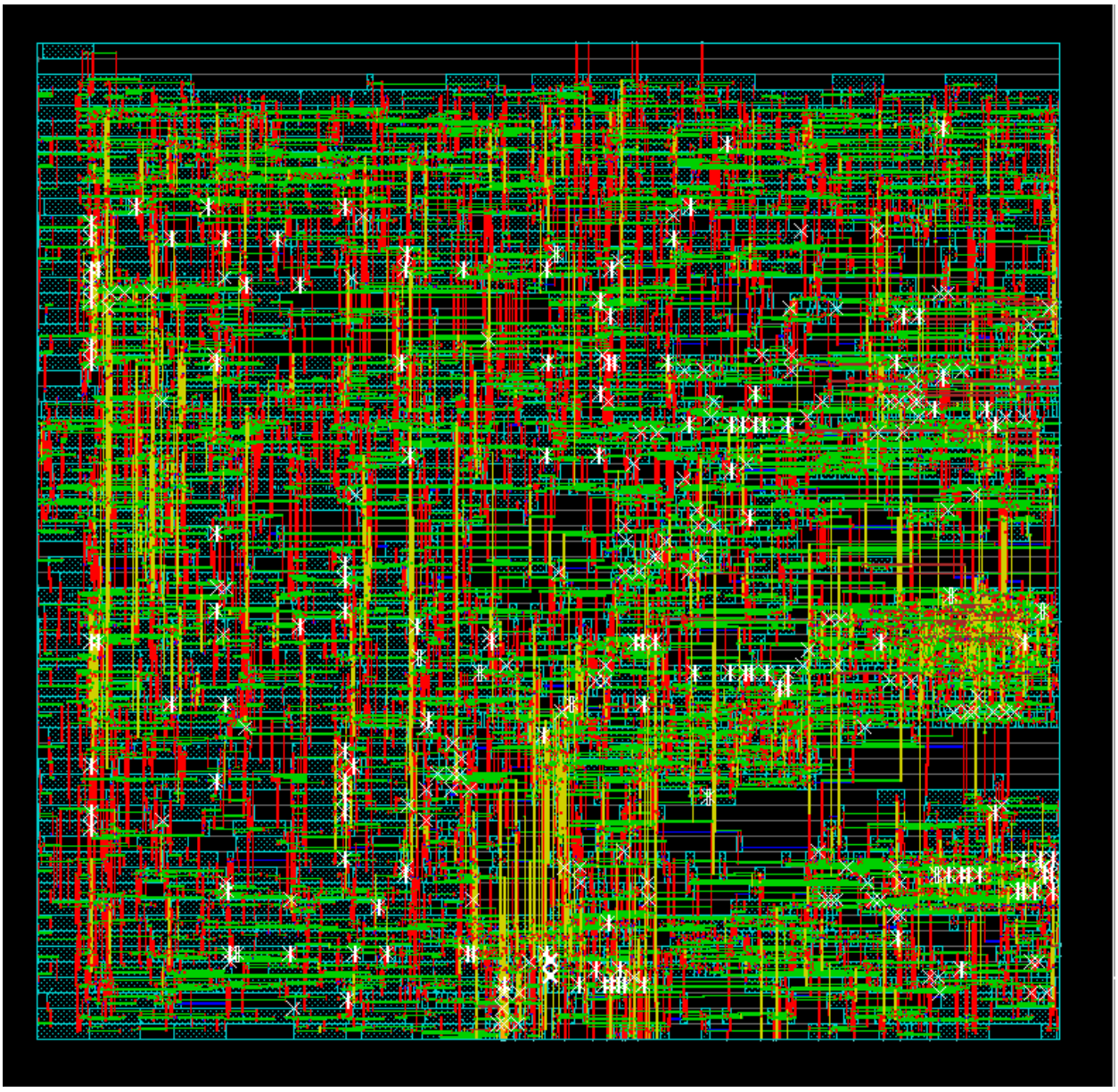

Fig7: Design after Routing.

Reference:

1. Wikipedia
2. ECE 5470/6470 Lab 2: Synthesis using Synopsys Design Compiler.
3. Lab3: Placement Using Cadence Soc Encounter
4. ECE 5470/6470 Lab 4: Routing Using SoC Encounter

Appendix A:

Verilog descriptions used in the floating point unit:

1)com:

```
// this module compares the highest exponent with the first input"a"'s exponent and if they are same we get an output of 1 else0

module com(exp1,e,eeqe);

  input [7:0] e,exp1;

  output eeqe;

  assign eeqe=(exp1==e)?1:0;

  endmodule
```

2)comparator:

```
//compares the two exponent values and assigns highest value to the exp1 and the lowest to exp2

module comparator(exp1,exp2,c,d);

input [7:0] c,d;

output [7:0] exp1,exp2;

assign exp1=(c>d) || (c==d) ? c:d;

assign exp2=(c<d)?c:d;

endmodule
```

3)shiftreg

```
//this module will shift the bits depending on the difference of the exponents
module shiftreg(s6,clk,reset,load,d,s4);
  output reg [23:0] s4;
  input [23:0] s6;
  input [7:0] d;
  input reset,clk,load;
  always @(posedge clk)
  //initially s4 will have a value 0
  if(reset==1) s4<=0;
  //later we give the value of s6 to s4
  else if (load==1) s4<=s6;
  else begin
    case (d)
      //if the difference is zero no shift occurs
8'b00000000:s4<=s4;
      //if the difference is 8'b00000001 we have to shift s4 1bit left the rest is the same
      8'b00000001: s4 <={1'b0,s4[23:1]};
      8'b00000010: s4 <={2'b0,s4[23:2]};
      8'b00000011: s4 <={3'b0,s4[23:3]};
      8'b00000100: s4 <={4'b0,s4[23:4]};
      8'b00000101: s4 <={5'b0,s4[23:5]};
      8'b00000110: s4 <={6'b0,s4[23:6]};
      8'b00000111: s4 <={7'b0,s4[23:7]};
```

```
      8'b00001000: s4 <={8'b0,s4[23:8]};
      8'b00001001: s4 <={9'b0,s4[23:9]};
      8'b00001010: s4 <={10'b0,s4[23:10]};
      8'b00001011: s4 <={11'b0,s4[23:11]};
      8'b00001100: s4 <={12'b0,s4[23:12]};
      8'b00001101: s4 <={13'b0,s4[23:13]};
      8'b00001110: s4 <={14'b0,s4[23:14]};
      8'b00001111: s4 <={15'b0,s4[23:15]};
      8'b00010000: s4 <={16'b0,s4[23:16]};
      8'b00010001: s4 <={17'b0,s4[23:17]};
      8'b00010010: s4 <={18'b0,s4[23:18]};
      8'b00010011: s4 <={19'b0,s4[23:19]};
      8'b00010100: s4 <={20'b0,s4[23:20]};
      8'b00010101: s4 <={21'b0,s4[23:21]};
      8'b00010110: s4 <={22'b0,s4[23:22]};
      8'b00010111: s4 <={23'b0,s4[23]};
      //if the difference is some value greater than 24 we assign this default
case
       default: s4 <=24'b0;
     endcase
     end
     endmodule
```

4)shiftreg_sub

```
module shiftreg_sub(s6,clk,reset,load,d,s4);
  output reg [23:0] s4;
```

```
input [23:0] s6;
input [7:0] d;
input reset,clk,load;
always @(posedge clk)
 //initially s4 will have a value 0
if(reset==1) s4<=0;
 //later we give the value of s6 to s4
else if (load==1) s4<=s6;
else begin
  case (d)
    //if the difference is zero no shift occurs
    8'b00000000:s4<=s4;
    //if the difference is not 8'b00000000 we have to shift s4 d bits left,shifting in 1's
    8'b00000001: s4 <={1'b1,s4[23:1]};
    8'b00000010: s4 <={2'b11,s4[23:2]};
    8'b00000011: s4 <={3'b111,s4[23:3]};
    8'b00000100: s4 <={4'b1111,s4[23:4]};
    8'b00000101: s4 <={5'b11111,s4[23:5]};
    8'b00000110: s4 <={6'b111111,s4[23:6]};
    8'b00000111: s4 <={7'b1111111,s4[23:7]};
    8'b00001000: s4 <={8'b11111111,s4[23:8]};
    8'b00001001: s4 <={9'b111111111,s4[23:9]};
    8'b00001010: s4 <={10'b1111111111,s4[23:10]};
    8'b00001011: s4 <={11'b11111111111,s4[23:11]};
```

```
      8'b00001100: s4 <={12'b111111111111,s4[23:12]};
      8'b00001101: s4 <={13'b1111111111111,s4[23:13]};
      8'b00001110: s4 <={14'b11111111111111,s4[23:14]};
      8'b00001111: s4 <={15'b111111111111111,s4[23:15]};
      8'b00010000: s4 <={16'b1111111111111111,s4[23:16]};
      8'b00010001: s4 <={17'b11111111111111111,s4[23:17]};
      8'b00010010: s4 <={18'b111111111111111111,s4[23:18]};
      8'b00010011: s4 <={19'b1111111111111111111,s4[23:19]};
      8'b00010100: s4 <={20'b11111111111111111111,s4[23:20]};
      8'b00010101: s4 <={21'b111111111111111111111,s4[23:21]};
      8'b00010110: s4 <={22'b1111111111111111111111,s4[23:22]};
      8'b00010111: s4 <={23'b11111111111111111111111,s4[23]};
      default: s4 <=24'b111111111111111111111111;
    endcase
    end
  endmodule
```

5)simple_division:

```
module simple_division(a2,t,p1,p,p2,t1,b_mantissa);
input [24:0] t,p,p1,p2,t1;
input [22:0] b_mantissa;
output a2;
//we call this module 25 times
//if msb bit of t is o we assign 1 to a2 else 0
assign a2=(t[24]==1'b0)? 1:0;
```

```
//if msb bit of t is 0 we assign t tp 0 else p will remani as it is

assign p1 = (t[24]==1'b0)? t:p;

assign p2={p1[23:0],1'b0};

 assign t1=p2-{2'b01,b_mantissa};

 endmodule
```

6)subtractor

//from the comparator module we have the highest module in exp1 and the lowest in exp2 so we take the difference of exp1 and exp2 and we get a positive value

```
module subtractor(d,exp1,exp2);

output [7:0] d;

input [7:0] exp1,exp2;

assign d=exp1-exp2;

endmodule
```

7)fpadder3

```
//adder module

module fpadder3(a,b,finalout,clk,reset,load);

  input clk,reset,load;

  input [31:0] a,b;

  wire [23:0] s1,s2;

  wire msb;

  wire [23:0] sum;

  output [31:0] finalout;

  wire carry;
```

```
 wire [23:0]out;

 wire [7:0] exp1,exp2,exp;

wire [7:0] d;

wire [23:0] s3,s5;

wire [23:0]s4;

assign s1={1'b1,a[22:0]};

assign s2={1'b1,b[22:0]};

assign msb=a[31];

wire aeqe,beqe;

//we get highest exponent from this module

comparator m1(exp1,exp2,a[30:23],b[30:23]);

//we see from  this module if the highest exponent is equal to the first input exponents value

com m2(exp1,a[30:23],aeqe);

//here we subtract the highest exponent from the lowest exponent

subtractor m4(d,exp1,exp2);

//we assign the significand of the input which contains the lowest exponent value to s3

assign s3=(aeqe==1)?s2:s1;

//we assign the significand of the input which contains the highest exponent value to s5

assign s5=(aeqe==1)?s1:s2;

//depending on the difference of the exponents generated we shift the s3 d times where d is the difference of the exponents

shiftreg m5(s3,clk,reset,load,d,s4);

//add the exponents
```

```
 assign {carry,sum}=s5+s4;

 //if carry is generated shift the sum along with the carry one bit right else left shift the sum until it is normalized

 assign out=(carry==1)?{carry,sum[23:1]}:sum;

 //depending on how many bits the sum is left shited  add 1 to the exponent

 assign exp=(carry==1)?(exp1+1):exp1;


 //so from the final sum we get and the exponent we get we can write the finalout

 assign finalout={msb,exp,out[22:0]};

 endmodule
```

8)fpsubtractor

```
module fpsubtractor(a,b,sub_out,clk,reset,load);

  input clk,reset,load;

  input [31:0] a,b;

  wire [23:0] s1,s2;

  wire msb;

  wire [23:0] sum;

  output [31:0] sub_out;

  wire carry;

  wire [23:0]out;

  wire [7:0] exp1,exp2,exp;

 wire[7:0] d;

 wire [23:0] s3,s5;

 wire [23:0]s4,s6;
```

```
 wire sign;

 assign sign=(a[30:0]>b[30:0])?a[31]:b[31];

 //assigning values to the significand including the hidden 1bit

 assign s1={1'b1,a[22:0]};

 assign s2={1'b1,b[22:0]};

 wire aeqe;

  //we get highest exponent from this module

 comparator m1(exp1,exp2,a[30:23],b[30:23]);

  //we see from  this module if the highest exponent is equal to the first input exponents value

 com m2(exp1,a[30:23],aeqe);

 //here we subtract the highest exponent from the lowest exponent

 subtractor m4(d,exp1,exp2);

  //we assign the significand of the input which contains the lowest exponent value to s3

 assign s3=(aeqe==1)?s2:s1;

  //we assign the significand of the input which contains the highest exponent value to s5

 assign s5=(aeqe==1)?s1:s2;

 assign s6=~s3+1;

 //depending on the difference of the exponents generated we shift the 1 in to s3 d times where d is the difference of the exponents

 shiftreg_sub m5(s6,clk,reset,load,d,s4);

 assign {carry,sum}=s5+s4;

  //if no carry is generated and if msb bit is zero and if the inputs are different signs compliment the mantissa
```

```
 assign out=(carry==0)&& (sum[23]==1)? ~sum+1:sum;
 assign exp=(sum[23]==0)?(exp1-1):exp1;
 assign
sub_out=(out[23]==0)?{sign,exp,out[21:0],1'b0}:{sign,exp,out[22:0]};
  endmodule
```

9)fpmultiplier

```
module fpmultiplier(a,b,mul_out);
input [31:0] a,b;
wire [47:0] f;
wire [23:0] s1,s2;
wire [7:0] e1,e2;
wire [7:0] exp,exp1;
wire [23:0] reg1,reg2,reg3;
wire [22:0] final;
wire msb;
xor(msb,a[31],b[31]);
output [31:0] mul_out;
//denormalize the significands
assign s1={1'b1,a[22:0]};
assign s2={1'b1,b[22:0]};
assign e1=a[30:23];
assign e2=b[30:23];
//we add the exponents and subtract 127 for biasing purpose
assign exp=e1+e2-127;
```

```
//multiply the significands
assign f=s1*s2;
assign reg2=f[23:0];
assign reg1=f[47:24];

//if the final matissa has 0 as the 24th bit left shift it and take the remaning 22 bits excluding the most significand bit. this is called normalization
assign reg3=((reg1[23]==0) ? {reg1[22:0],reg2[23]}:reg1);
//depending on how many bits shifted assign the value to the exponent
assign exp1=(reg1[23]==0) ? exp:exp+1;
assign final=reg3[22:0];
//from the above obtained values we can then write mul_out
assign mul_out={msb,exp1,final};
endmodule
```

10)final_div

```
//division module
module final_div(x,b,z);
input [31:0] x,b;
output [31:0] z;
wire[22:0] z_mantissa;
wire [7:0] z_exponent;
wire z_sign;
wire [22:0] x_mantissa,b_mantissa;
wire x_sign,b_sign;
wire [7:0] x_exponent,b_exponent;
```

```
wire [8:0] exp,exp1;

wire [25:0] a2,a1;

wire [24:0] t,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t25;

wire [24:0] p,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,p20,p21,p22,p23,p24,p25,p26,p27,p28,p29,p30,p31,p32,p33,p34,p35,p36,p37,p38,p39,p40,p41,p42,p43,p44,p45,p46,p47,p48,p49,p50;

assign x_mantissa=x[22:0];

assign x_exponent=x[30:23];

assign x_sign=x[31];

assign b_mantissa=b[22:0];

assign b_exponent=b[30:23];

assign b_sign=b[31];

xor(z_sign,x_sign,b_sign);

assign exp={1'b0,x_exponent}-{1'b0,b_exponent}+127;

assign p ={2'b01,x_mantissa};

assign t=p-{2'b01,b_mantissa};

//submodule of division module

// these modules are used instead of loop instruction

simple_division m1(a2[0],t,p1,p,p2,t1,b_mantissa);

simple_division m2(a2[1],t1,p3,p2,p4,t2,b_mantissa);

simple_division m3(a2[2],t2,p5,p4,p6,t3,b_mantissa);

simple_division m4(a2[3],t3,p7,p6,p8,t4,b_mantissa);

simple_division m5(a2[4],t4,p9,p8,p10,t5,b_mantissa);
```

```
simple_division m6(a2[5],t5,p11,p10,p12,t6,b_mantissa);
simple_division m7(a2[6],t6,p13,p12,p14,t7,b_mantissa);
simple_division m8(a2[7],t7,p15,p14,p16,t8,b_mantissa);
simple_division m9(a2[8],t8,p17,p16,p18,t9,b_mantissa);
simple_division m10(a2[9],t9,p19,p18,p20,t10,b_mantissa);
simple_division m11(a2[10],t10,p21,p20,p22,t11,b_mantissa);
simple_division m12(a2[11],t11,p23,p22,p24,t12,b_mantissa);
simple_division m13(a2[12],t12,p25,p24,p26,t13,b_mantissa);
simple_division m14(a2[13],t13,p27,p26,p28,t14,b_mantissa);
simple_division m15(a2[14],t14,p29,p28,p30,t15,b_mantissa);
simple_division m16(a2[15],t15,p31,p30,p32,t16,b_mantissa);
simple_division m17(a2[16],t16,p33,p32,p34,t17,b_mantissa);
simple_division m18(a2[17],t17,p35,p34,p36,t18,b_mantissa);
simple_division m19(a2[18],t18,p37,p36,p38,t19,b_mantissa);
simple_division m20(a2[19],t19,p39,p38,p40,t20,b_mantissa);
simple_division m21(a2[20],t20,p41,p40,p42,t21,b_mantissa);
simple_division m22(a2[21],t21,p43,p42,p44,t22,b_mantissa);
simple_division m23(a2[22],t22,p45,p44,p46,t23,b_mantissa);
simple_division m24(a2[23],t23,p47,p46,p48,t24,b_mantissa);
simple_division m25(a2[24],t24,p49,p48,p50,t25,b_mantissa);
assign a2[25]=(t25[24]==1'b0)? 1:0;
assign a1=a2;
assign z_mantissa=(a2[25]==1)? a1[24:2]:a1[23:1];
assign exp1=(a1[25]==1)? exp-1:exp;
```

```
assign z_exponent=exp1[7:0];
assign z={z_sign,z_exponent,z_mantissa};
endmodule
```

11)mux

```
module mux(finalout,sub_out,mul_out,z,select,final);
  output reg [31:0] final;
  input [31:0] finalout,sub_out,mul_out,z;
  input [1:0] select;
  always @(select)
  case (select)
  //if select is 2'b00 we select finalout output of adder
  0: final=finalout;
  //if select is 2'b01 we select sub_out output of subtractor
  1:final=sub_out;
  //if select is 2'b10 we select mul_out output of multiplier
  2: final=mul_out;
  //if select is 2'b11 we select z output of divider
  3: final=z;
endcase
endmodule
```

12)top_module

```
module
top_module(final,clk,reset,load,qnan,snan,negzero,poszero,neginfinity,posinf
inity,overflow,underflow,a,b);

  input [31:0] a,b;

  output [31:0] final;

  input clk,reset,load;

  wire [31:0] finalout,sub_out,mul_out,z;

  wire [23:0] final_mantissa;

  wire [7:0] final_exponent;

  wire final_sign;

  wire [1:0] select;

  output
qnan,snan,negzero,poszero,neginfinity,posinfinity,overflow,underflow;

  //executing the adder module

  fpadder3 m1(a,b,finalout,clk,reset,load);

  //executing the subtractor module

  fpsubtractor m2(a,b,sub_out,clk,reset,load);

  //executing the multiplier module

  fpmultiplier m3(a,b,mul_out);

  //executing the divider module

  final_div m4(a,b,z);

  //executing the mux module

  mux m5(finalout,sub_out,mul_out,z,select,final);

assign final_mantissa=final[22:0];

assign final_exponent=final[30:23];
```

```
assign final_sign=final[31];

assign poszero=((a[22:0]==23'b0 && a[7:0]==8'b0 && a[31]==1'b0)||(b[22:0]==23'b0 && b[7:0]==8'b0 && b[31]==1'b0)||(final_mantissa==23'b0 && final_exponent==8'b0 && final_sign==1'b0))? 1'b1: 1'b0;

assign posinfinity=((a[31]==1'b0 && a[7:0]==8'b1 && a[22:0]==23'b0)||(b[31]==1'b0 && b[7:0]==8'b1 && b[22:0]==23'b0)||(final_sign==1'b0 && final_exponent==8'b1 && final_mantissa==23'b0)) ?1'b1:1'b0;

assign snan=((a[30:23]==8'b1 && a[22]==1'b0)||(b[30:23]==8'b1 && b[22]==1'b0)||(final_exponent==8'b1 && final_mantissa==1'b0))? 1'b1:1'b0;

assign qnan=((a[30:23]==8'b1 && a[22]==1'b1)||(b[30:23]==8'b1 && b[22]==1'b1)||(final_exponent==8'b1 && final_mantissa==1'b1))? 1'b1:1'b0;

assign negzero=((a[22:0]==23'b0 && a[30:23]==8'b0 && a[31]==1'b1)||(b[22:0]==23'b0 && b[30:23]==8'b0 && b[31]==1'b1)||(final_mantissa==23'b0 && final_exponent==8'b0 && final_sign==1'b1))? 1'b1: 1'b0;

assign neginfinity=(a[31]==1'b1 && a[30:23]==8'b1 && a[22:0]==23'b0)||(a[31]==1'b1 && a[30:23]==8'b1 && a[22:0]==23'b0)||(b[31]==1'b1 && b[30:23]==8'b1 && b[22:0]==23'b0) ? 1'b1:1'b0;

assign overflow=(final_exponent==8'b10000000)?1:0;

assign underflow=(final_exponent==8'b10000010)?1:0;

endmodule
```

Appendix B

1.area report:

```
Warning: Design 'top_module' has '1' unresolved references. For more
detailed information, use the "link" command. (UID-341)
Information: Updating graph... (UID-83)
```

```
****************************************
Report : area
Design : top_module
Version: E-2010.12-SP4
Date   : Fri Apr 20 17:21:54 2012
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    fsd0a_a_generic_core_tt1v25c (File:
/opt/software/cadence/library/fsd0a_90nm_generic_core/timing/fsd0a_a_g
eneric_core_tt1v25c.db)

Number of ports:                          107
Number of nets:                           730
Number of cells:                          508
Number of combinational cells:            504
Number of sequential cells:                 0
Number of macros:                           0
Number of buf/inv:                         12
Number of references:                      11

Combinational area:       22757.951883
Noncombinational area:     319.872002
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:          23077.823884
Total area:                 undefined

Information: This design contains black box (unknown) components.
(RPT-8)
1
```

2.power report:

```
Information: Propagating switching activity (low effort zero delay
simulation). (PWR-6)
Warning: The derived static probability value (0.500000) for the clock
net 'clk' conflicts with the annotated value (0.100000). Using the
annotated value. (PWR-12)
Warning: The derived toggle rate value (0.166667) for the clock net
'clk' conflicts with the annotated value (0.000833). Using the
annotated value. (PWR-12)
Warning: Design has unannotated sequential cell outputs. (PWR-415)
Warning: Design has unannotated black box outputs. (PWR-428)

****************************************
Report : power
        -analysis_effort low
Design : top_module
```

```
Version: E-2010.12-SP4
Date   : Fri Apr 20 17:21:55 2012
****************************************


Library(s) Used:

    fsd0a_a_generic_core_tt1v25c (File:
/opt/software/cadence/library/fsd0a_90nm_generic_core/timing/fsd0a_a_g
eneric_core_tt1v25c.db)


Operating Conditions: TCCOM   Library: fsd0a_a_generic_core_tt1v25c
Wire Load Model Mode: enclosed

Design        Wire Load Model            Library
------------------------------------------------
top_module             enG10K            fsd0a_a_generic_core_tt1v25c
fpadder3               enG5K             fsd0a_a_generic_core_tt1v25c
fpmultiplier           enG10K            fsd0a_a_generic_core_tt1v25c
final_div              enG5K             fsd0a_a_generic_core_tt1v25c
mux                    enG5K             fsd0a_a_generic_core_tt1v25c
comparator             enG5K             fsd0a_a_generic_core_tt1v25c
com                    enG5K             fsd0a_a_generic_core_tt1v25c
subtractor             enG5K             fsd0a_a_generic_core_tt1v25c
shiftreg               enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_24     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_0      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_1      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_2      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_3      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_4      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_5      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_6      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_7      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_8      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_9      enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_10     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_11     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_12     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_13     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_14     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_15     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_16     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_17     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_18     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_19     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_20     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_21     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_22     enG5K             fsd0a_a_generic_core_tt1v25c
simple_division_23     enG5K             fsd0a_a_generic_core_tt1v25c
final_div_DW01_sub_0   enG5K             fsd0a_a_generic_core_tt1v25c
```

```
fpmultiplier_DW01_inc_0
                        enG5K             fsd0a_a_generic_core_tt1v25c
fpadder3_DW01_inc_0     enG5K             fsd0a_a_generic_core_tt1v25c
fpadder3_DW01_add_0     enG5K             fsd0a_a_generic_core_tt1v25c
subtractor_DW01_sub_0   enG5K             fsd0a_a_generic_core_tt1v25c
final_div_DW01_add_0    enG5K             fsd0a_a_generic_core_tt1v25c
fpmultiplier_DW01_add_0
                        enG5K             fsd0a_a_generic_core_tt1v25c
fpmultiplier_DW02_mult_0
                        enG10K            fsd0a_a_generic_core_tt1v25c
fpmultiplier_DW01_add_1
                        enG5K             fsd0a_a_generic_core_tt1v25c


Global Operating Voltage = 1
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =  16.5197 uW   (52%)
  Net Switching Power  =  15.0415 uW   (48%)
                         ---------
Total Dynamic Power    =  31.5612 uW  (100%)

Cell Leakage Power     =  31.0927 uW

1
```

3.timing_report:

```
Warning: Design 'top_module' has '1' unresolved references. For more
detailed information, use the "link" command. (UID-341)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : top_module
Version: E-2010.12-SP4
Date   : Fri Apr 20 17:21:55 2012
****************************************

Operating Conditions: TCCOM   Library: fsd0a_a_generic_core_tt1v25c
Wire Load Model Mode: enclosed
```

```
  Startpoint: m1/m5/s4_reg[23]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: m1/m5/s4_reg[4]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  top_module         enG10K
fsd0a_a_generic_core_tt1v25c
  fpadder3           enG5K
fsd0a_a_generic_core_tt1v25c
  shiftreg           enG5K
fsd0a_a_generic_core_tt1v25c

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  m1/m5/s4_reg[23]/CK (QDFFX1)             0.00       0.00 r
  m1/m5/s4_reg[23]/Q (QDFFX1)              0.21       0.21 r
  m1/m5/U12/O (INVCKXLP)                   0.18       0.39 f
  m1/m5/U40/O (MUX2XLP)                    0.11       0.50 f
  m1/m5/U39/O (OR2XLP)                     0.06       0.56 f
  m1/m5/U101/OB (MXL3XLP)                  0.13       0.69 r
  m1/m5/U196/O (MUX3XLP)                   0.08       0.77 r
  m1/m5/U195/O (AO222XLP)                  0.09       0.86 r
  m1/m5/s4_reg[4]/D (QDFFX1)               0.00       0.86 r
  data arrival time                                   0.86

  clock clk (rise edge)                   12.00      12.00
  clock network delay (ideal)              0.00      12.00
  clock uncertainty                       -0.50      11.50
  m1/m5/s4_reg[4]/CK (QDFFX1)              0.00      11.50 r
  library setup time                      -0.04      11.46
  data required time                                 11.46
  -----------------------------------------------------------
  data required time                                 11.46
  data arrival time                                  -0.86
  -----------------------------------------------------------
  slack (MET)                                        10.60


  Startpoint: b[14] (input port)
  Endpoint: negzero (output port)
  Path Group: default
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
```

```
  top_module         enG10K
fsd0a_a_generic_core_tt1v25c
  fpmultiplier_DW02_mult_0
                     enG10K
fsd0a_a_generic_core_tt1v25c
  fpmultiplier_DW01_add_1
                     enG5K
fsd0a_a_generic_core_tt1v25c
  fpmultiplier       enG10K
fsd0a_a_generic_core_tt1v25c
  mux                enG5K
fsd0a_a_generic_core_tt1v25c

  Point                                                    Incr
Path
  ---------------------------------------------------------------------
------
  input external delay                                     0.00
0.00 r
  b[14] (in)                                               0.00
0.00 r
  m3/b[14] (fpmultiplier)                                  0.00
0.00 r
  m3/mult_17/B[14] (fpmultiplier_DW02_mult_0)              0.00
0.00 r
  m3/mult_17/U528/O (INVCKXLP)                             0.32
0.32 f
  m3/mult_17/U85/O (NR2XLP)                                0.15
0.47 r
  m3/mult_17/U11/O (AN2XLP)                                0.08
0.55 r
  m3/mult_17/S2_2_13/S (FA1X1)                             0.17
0.72 f
  m3/mult_17/S2_3_12/CO (FA1X1)                            0.08
0.80 f
  m3/mult_17/S2_4_12/S (FA1X1)                             0.14
0.94 r
  m3/mult_17/S2_5_11/CO (FA1X1)                            0.07
1.01 r
  m3/mult_17/S2_6_11/S (FA1X1)                             0.17
1.18 f
  m3/mult_17/S2_7_10/CO (FA1X1)                            0.08
1.25 f
  m3/mult_17/S2_8_10/S (FA1X1)                             0.14
1.40 r
  m3/mult_17/S2_9_9/CO (FA1X1)                             0.07
1.46 r
  m3/mult_17/S2_10_9/S (FA1X1)                             0.17
1.63 f
  m3/mult_17/S2_11_8/CO (FA1X1)                            0.08
1.71 f
```

```
  m3/mult_17/S2_12_8/S (FA1X1)                              0.14
1.85 r
  m3/mult_17/S2_13_7/CO (FA1X1)                             0.07
1.92 r
  m3/mult_17/S2_14_7/S (FA1X1)                              0.17
2.08 f
  m3/mult_17/S2_15_6/CO (FA1X1)                             0.08
2.16 f
  m3/mult_17/S2_16_6/S (FA1X1)                              0.14
2.30 r
  m3/mult_17/S2_17_5/CO (FA1X1)                             0.07
2.37 r
  m3/mult_17/S2_18_5/S (FA1X1)                              0.17
2.54 f
  m3/mult_17/S2_19_4/CO (FA1X1)                             0.08
2.62 f
  m3/mult_17/S2_20_4/S (FA1X1)                              0.14
2.76 r
  m3/mult_17/S2_21_3/CO (FA1X1)                             0.07
2.83 r
  m3/mult_17/S2_22_3/S (FA1X1)                              0.17
2.99 f
  m3/mult_17/S4_2/S (FA1X1)                                 0.09
3.09 r
  m3/mult_17/U609/O (AN2XLP)                                0.07
3.16 r
  m3/mult_17/FS_1/B[24] (fpmultiplier_DW01_add_1)           0.00
3.16 r
  m3/mult_17/FS_1/U14/O (ND2XLP)                            0.05
3.21 f
  m3/mult_17/FS_1/U9/O (OAI12XLP)                           0.07
3.27 r
  m3/mult_17/FS_1/U7/O (AOI12XLP)                           0.04
3.31 f
  m3/mult_17/FS_1/U6/O (OA12XLP)                            0.07
3.38 f
  m3/mult_17/FS_1/U10/O (OAI12XLP)                          0.04
3.43 r
  m3/mult_17/FS_1/U40/O (AOI12XLP)                          0.04
3.47 f
  m3/mult_17/FS_1/U39/O (OAI12XLP)                          0.06
3.53 r
  m3/mult_17/FS_1/U38/O (AOI12XLP)                          0.04
3.57 f
  m3/mult_17/FS_1/U37/O (OA12XLP)                           0.08
3.64 f
  m3/mult_17/FS_1/U43/O (OA12XLP)                           0.08
3.72 f
  m3/mult_17/FS_1/U46/O (OA12XLP)                           0.08
3.80 f
  m3/mult_17/FS_1/U49/O (OA12XLP)                           0.08
3.88 f
```

```
  m3/mult_17/FS_1/U52/O (OA12XLP)                          0.08
3.95 f
  m3/mult_17/FS_1/U55/O (OA12XLP)                          0.08
4.03 f
  m3/mult_17/FS_1/U58/O (OA12XLP)                          0.08
4.11 f
  m3/mult_17/FS_1/U61/O (OA12XLP)                          0.08
4.19 f
  m3/mult_17/FS_1/U64/O (OA12XLP)                          0.08
4.26 f
  m3/mult_17/FS_1/U67/O (OA12XLP)                          0.08
4.34 f
  m3/mult_17/FS_1/U70/O (OA12XLP)                          0.08
4.42 f
  m3/mult_17/FS_1/U73/O (OA12XLP)                          0.08
4.50 f
  m3/mult_17/FS_1/U105/O (OAI12XLP)                        0.11
4.61 r
  m3/mult_17/FS_1/U111/O (OR2XLP)                          0.06
4.66 r
  m3/mult_17/FS_1/U110/O (AO22XLP)                         0.07
4.73 r
  m3/mult_17/FS_1/U109/O (XOR2X1)                          0.29
5.02 r
  m3/mult_17/FS_1/SUM[45] (fpmultiplier_DW01_add_1)        0.00
5.02 r
  m3/mult_17/PRODUCT[47] (fpmultiplier_DW02_mult_0)        0.00
5.02 r
  m3/U39/O (INVCKXLP)                                      0.56
5.58 f
  m3/U30/O (AO22XLP)                                       0.17
5.75 f
  m3/mul_out[21] (fpmultiplier)                            0.00
5.75 f
  m5/mul_out[21] (mux)                                     0.00
5.75 f
  m5/U69/O (AOI22XLP)                                      0.06
5.81 r
  m5/U68/O (ND2XLP)                                        0.10
5.91 f
  m5/final[21] (mux)                                       0.00
5.91 f
  U51/O (OR2XLP)                                           0.07
5.98 f
  U50/O (OR2XLP)                                           0.05
6.03 f
  U93/O (OR2XLP)                                           0.05
6.08 f
  U92/O (OR2XLP)                                           0.05
6.13 f
  U91/O (OR2XLP)                                           0.05
6.18 f
```

```
  U90/O (OR2XLP)                                          0.05
6.23 f
  U148/O (OR2XLP)                                         0.05
6.28 f
  U147/O (OR2XLP)                                         0.05
6.33 f
  U146/O (OR2XLP)                                         0.05
6.38 f
  U145/O (OR2XLP)                                         0.05
6.43 f
  U206/O (OR2XLP)                                         0.05
6.48 f
  U205/O (OR2XLP)                                         0.05
6.53 f
  U204/O (OR2XLP)                                         0.05
6.58 f
  U203/O (OR2XLP)                                         0.05
6.63 f
  U292/O (OR2XLP)                                         0.05
6.68 f
  U291/O (OR2XLP)                                         0.05
6.73 f
  U290/O (OR2XLP)                                         0.05
6.78 f
  U289/O (OR2XLP)                                         0.05
6.83 f
  U385/O (OR2XLP)                                         0.05
6.88 f
  U384/O (OR2XLP)                                         0.05
6.93 f
  U383/O (OR2XLP)                                         0.05
6.98 f
  U382/O (OR2XLP)                                         0.05
7.03 f
  U467/O (NR2XLP)                                         0.04
7.08 r
  U496/O (AN2XLP)                                         0.05
7.13 r
  U494/O (OR2XLP)                                         0.04
7.17 r
  negzero (out)                                           0.00
7.17 r
  data arrival time
7.17

  max_delay                                              12.00
12.00
  output external delay                                   0.00
12.00
  data required time
12.00
```

```
  ------------------------------------------------------------------
------
  data required time
12.00
  data arrival time                                                -
7.17
  ------------------------------------------------------------------
------
  slack (MET)
4.83


1
```

4.threshold_voltage:

```
Warning: Design 'top_module' has '1' unresolved references. For more
detailed information, use the "link" command. (UID-341)
****************************************************************
**********
                        Threshold Voltage Group Report

Vth Group                 All                Blackbox           Non-
blackbox
Name                      cells               cells
cells
****************************************************************
**********
undefined            2597 (100.00%)          1 (100.00%)       2596
(100.00%)
****************************************************************
**********

Vth Group                 All                Blackbox           Non-
blackbox
Name                   cell area            cell area            cell
area
****************************************************************
**********
undefined        23077.82 (100.00%)       0.00   (0.00%)   23077.82
(100.00%)
****************************************************************
**********
1
```

5.latch_report:

```
Warning: Design 'top_module' has '1' unresolved references. For more
detailed information, use the "link" command. (UID-341)

****************************************
```

```
Report : register
Design : top_module
Version: E-2010.12-SP4
Date   : Fri Apr 20 17:21:55 2012
****************************************

Attributes:
    b - black box (unknown)
    h - hierarchical
    n - noncombinational
    r - removable
    u - contains unmapped logic

Cell                      Reference       Library              Area
Attributes
--------------------------------------------------------------------------
----------
--------------------------------------------------------------------------
----------
Total 0 cells                                                  0.000000
1
```

6.violator reports:

```
Warning: Design 'top_module' has '1' unresolved references. For more
detailed information, use the "link" command. (UID-341)

****************************************
Report : constraint
        -all_violators
Design : top_module
Version: E-2010.12-SP4
Date   : Fri Apr 20 17:21:55 2012
****************************************


   min_delay/hold ('clk' group)

                             Required        Actual
   Endpoint                Path Delay     Path Delay        Slack
   ---------------------------------------------------------------
   m1/m5/s4_reg[0]/D           0.49           0.18 r        -0.31
(VIOLATED)
   m1/m5/s4_reg[1]/D           0.48           0.18 f        -0.30
(VIOLATED)
   m1/m5/s4_reg[4]/D           0.48           0.19 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[2]/D           0.48           0.19 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[8]/D           0.48           0.19 f        -0.29
(VIOLATED)
```

```
   m1/m5/s4_reg[5]/D            0.48           0.19 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[6]/D            0.48           0.19 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[7]/D            0.48           0.19 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[9]/D            0.48           0.20 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[3]/D            0.48           0.20 f        -0.29
(VIOLATED)
   m1/m5/s4_reg[10]/D           0.48           0.20 f        -0.28
(VIOLATED)
   m1/m5/s4_reg[18]/D           0.48           0.20 f        -0.28
(VIOLATED)
   m1/m5/s4_reg[15]/D           0.48           0.20 f        -0.28
(VIOLATED)
   m1/m5/s4_reg[11]/D           0.48           0.21 f        -0.28
(VIOLATED)
   m1/m5/s4_reg[14]/D           0.48           0.21 f        -0.28
(VIOLATED)
   m1/m5/s4_reg[19]/D           0.48           0.21 f        -0.27
(VIOLATED)
   m1/m5/s4_reg[20]/D           0.48           0.21 f        -0.27
(VIOLATED)
   m1/m5/s4_reg[12]/D           0.48           0.21 f        -0.27
(VIOLATED)
   m1/m5/s4_reg[17]/D           0.48           0.21 f        -0.27
(VIOLATED)
   m1/m5/s4_reg[13]/D           0.48           0.22 f        -0.27
(VIOLATED)
   m1/m5/s4_reg[21]/D           0.48           0.22 f        -0.26
(VIOLATED)
   m1/m5/s4_reg[22]/D           0.48           0.24 f        -0.24
(VIOLATED)
   m1/m5/s4_reg[16]/D           0.48           0.28 f        -0.20
(VIOLATED)
   m1/m5/s4_reg[23]/D           0.48           0.32 f        -0.16
(VIOLATED)


   max_area

                           Required        Actual
   Design                    Area           Area           Slack
   -------------------------------------------------------------------
   top_module                0.00       23077.82       -23077.82
(VIOLATED)


   max_leakage_power

                           Required        Actual
```

Design	Leakage Power	Leakage Power	Slack
top_module	0.00	31092728.00	-31092728.00

(VIOLATED)

max_dynamic_power

Design	Required Dynamic Power	Actual Dynamic Power	Slack
top_module	0.00	0.03	-0.03

(VIOLATED)

1