SAN JOSÉ STATE UNIVERSITY
Charles W. Davidson College of Engineering
DEPARTMENT OF ELECTRICAL ENGINEERING
EE 271 – Advanced Digital System Design and Synthesis

Fall 2016 Final Project Report

Implementing a 64-bit Signed Binary Multiplier & Divider Circuit

Name	MIT VASANT MORABIA	SJSU ID	011469653
Email	mit.morabia@sjsu.edu	Phone	669-258-7346

Executive Summary (50 to 80 words)

Designing and implementation of digital circuits needs some very crucial decisions. Area and delay are always a trade-off. If we try to improve speed by reducing the clock frequency, the area might increase which also increases the power consumption. However, according to the design requirement, if low power consumption and smaller area is desired, speed will have to be compromised. Multipliers and dividers are the most important modules of mainly all the systems. This project designs a 32 bit signed multiplier and divider using a 32 bit RCA adder for additions and subtractions.

I. General Project Information

Table I.1: List of EDA Tools Used

EDA Tool Name	Company	You Used it for
VCS	Synopsys	Simulation & test
ModelSim	MentorGraphics	Simulation & test
Design Vision	Synopsys	Synthesis & optimization
NC Verilog	Cadence	Post Synthesis Simulation

Table I.2: List of Libraries Used

Library file name	**Used with** (EDA tool name)	**The library is at** (directories on eecad systems)
WCCOM	Synopsys Design Compiler	/apps/toshiba/sjsu/synopsys/tc20c/tc240c.db_WCCOM25
BCCOM	Synopsys Design Compiler	/apps/toshiba/sjsu/synopsys/tc20c/tc240c.db_BCCOM25
NOMIN	Synopsys Design Compiler	/apps/toshiba/sjsu/synopsys/tc20c/tc240c.db_NOMIN25

Table I.3: List of Verilog Modules (both design and test modules)

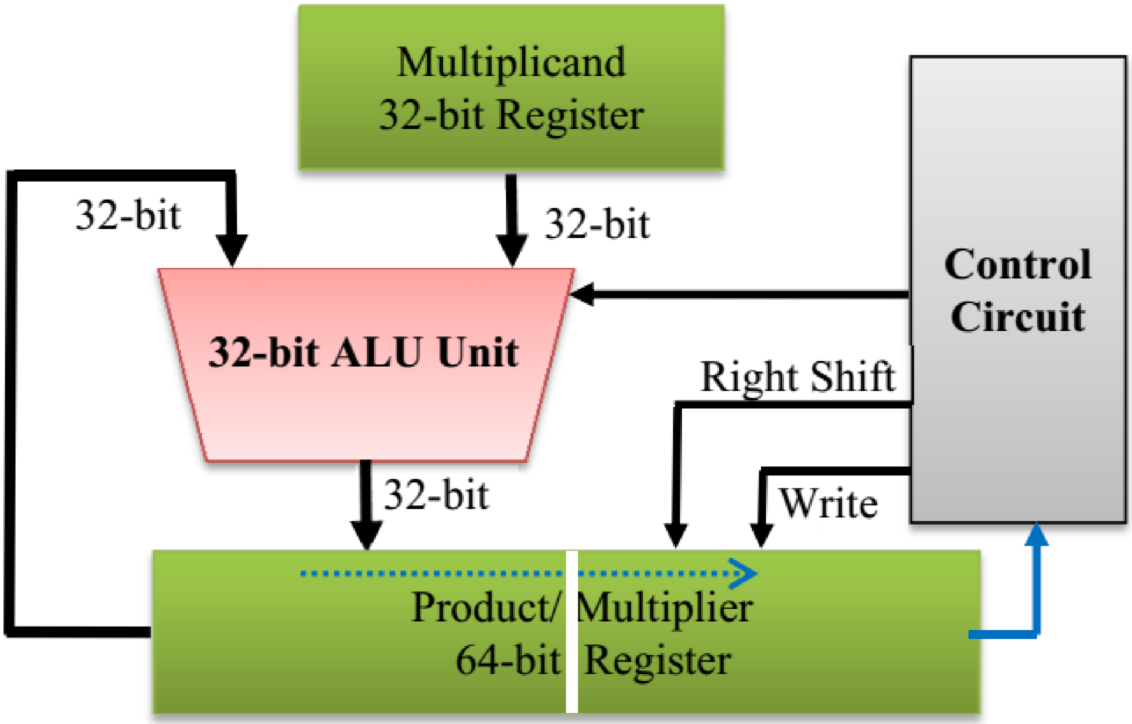
A) Multiplication

Module Name	Ports	Short Description
MULTIPLIER	OPERA1, OPERA2, START, MUORDI, CLOCK, RESULT, VALID.	32 BIT SEQUENTIAL MULTIPLIER.
ADD_RCA_1	SUM, COUT, A, B, CIN	32 BIT RIPPLE CARRY ADDER
ADD_RCA_16_1	SUM, COUT, A, B, CIN	16 BIT RIPPLE CARRY ADDER
ADD_RCA_4_1	SUM, COUT, A, B, CIN	4 BIT RIPPLE CARRY ADDER
ADD_FULL_1	SUM, COUT, A, B, CIN	FULL ADDER
ADD_HALF_1	SUM, COUT, A, B	HALF ADDER
TESTBENCH_MULTIPLY	OPERA1, OPERA2, START, MUORDI, CLOCK, RESULT, VALID.	TESTBENCH MODULE FOR MULTIPLIER

B) Division

Module Name	Ports	Short Description
DIVISION	OPERA1, OPERA2, START, MUORDI, CLOCK, RESULT, VALID.	32 BIT SEQUENTIAL DIVIDER
ADD_RCA_1	SUM, COUT, A, B, CIN	32 BIT RIPPLE CARRY ADDER
ADD_RCA_16_1	SUM, COUT, A, B, CIN	16 BIT RIPPLE CARRY ADDER
ADD_RCA_4_1	SUM, COUT, A, B, CIN	4 BIT RIPPLE CARRY ADDER
ADD_FULL_1	SUM, COUT, A, B, CIN	FULL ADDER
ADD_HALF_1	SUM, COUT, A, B	HALF ADDER
TESTBENCH_DIVISION	OPERA1, OPERA2, START, MUORDI, CLOCK, RESULT, VALID.	TESTBENCH MODULE FOR DIVISION

II. Implementation Overview



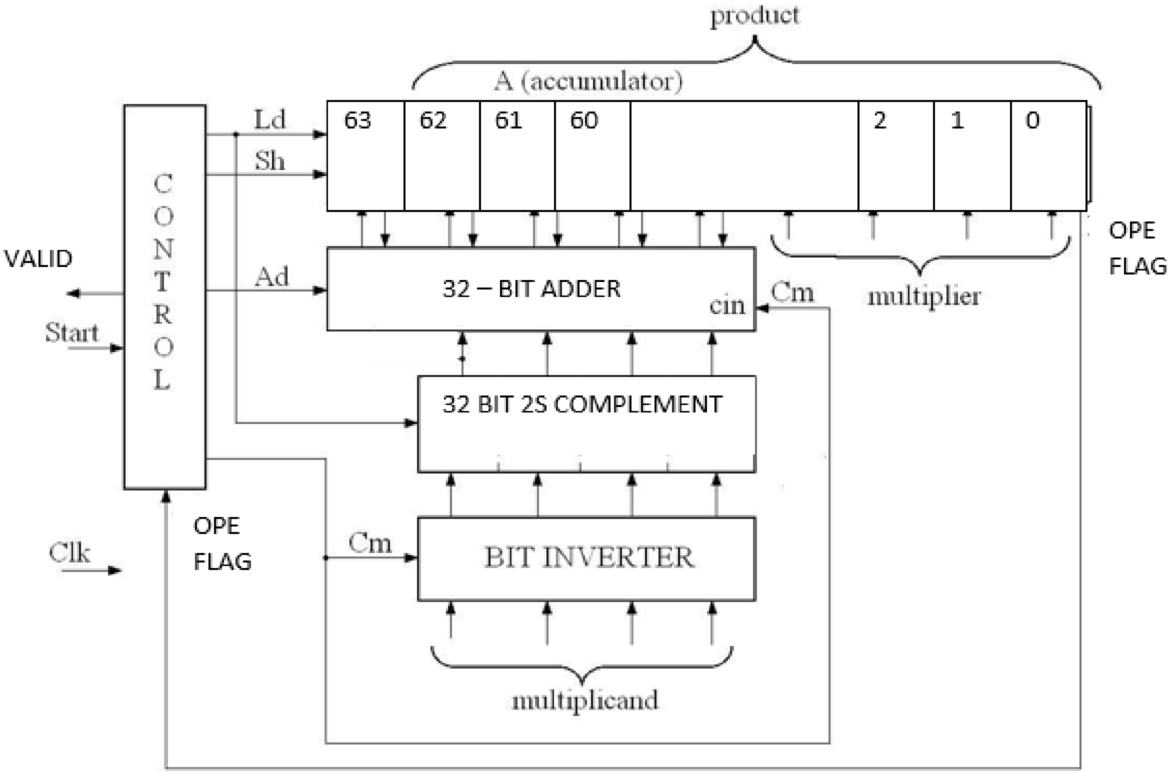
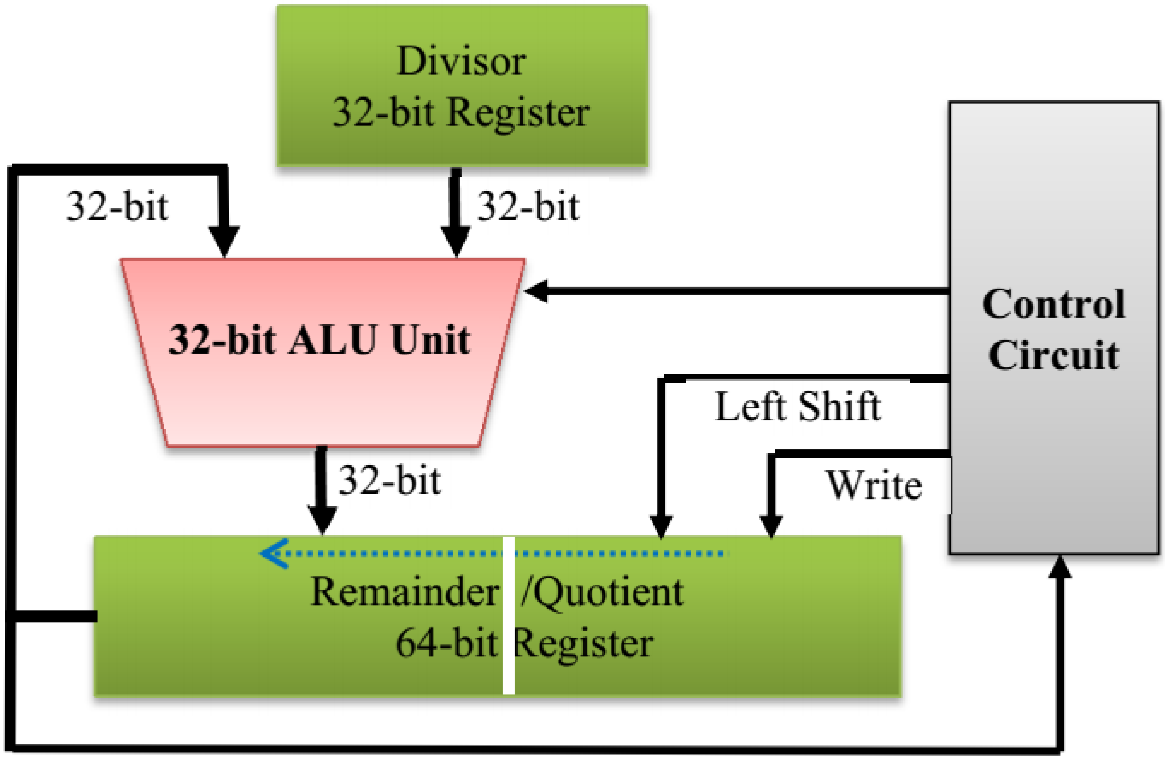


Figure II.1: Block Diagram of 32-bit ALU unit

Discuss the structure and characteristics of the ALU unit (less than 100 words)

The ALU shown in the above diagram is a 32 bit signed multiplier and division. The product of Multiplicand (Opera 1) and Multiplier/ Dividend (Opera2) is stored in the Product (Result) of the circuit. After the operation is complete, the value in Result changes from Opera2 to the product of Opera1 and Opera2. The bit inversion and 2s complementing of inputs depends on the operation as well as its original sign. The control logic depends on the the input start and generates output valid to state whether the result is ready. The OPE flag in Control block decides the shifting and addition or subtraction of the signal with the result register. Also the muordi flag decides the operation (Multiply or Divide).



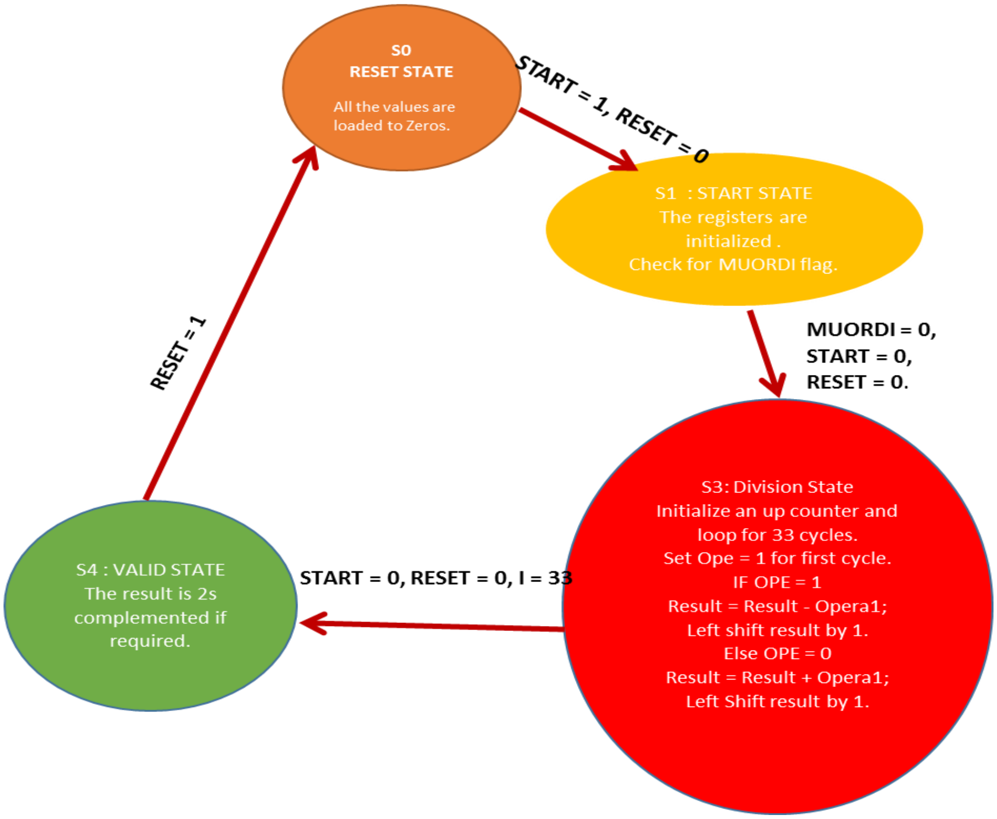

S0
RESET STATE
All the values are loaded to Zeros.
START = 1, RESET = 0
S1 : START STATE
The registers are initialized .
Check for MUORDI flag.
MUORDI = 0,
START = 0,
RESET = 0.
S3: Division State
Initialize an up counter and loop for 33 cycles.
Set Ope = 1 for first cycle.
IF OPE = 1
Result = Result - Opera1;
Left shift result by 1.
Else OPE = 0
Result = Result + Opera1;
Left Shift result by 1.
START = 0, RESET = 0, I = 33
S4 : VALID STATE
The result is 2s complemented if required.
RESET = 1

Figure II.2: State Transition Diagram of Sequential Multiplier

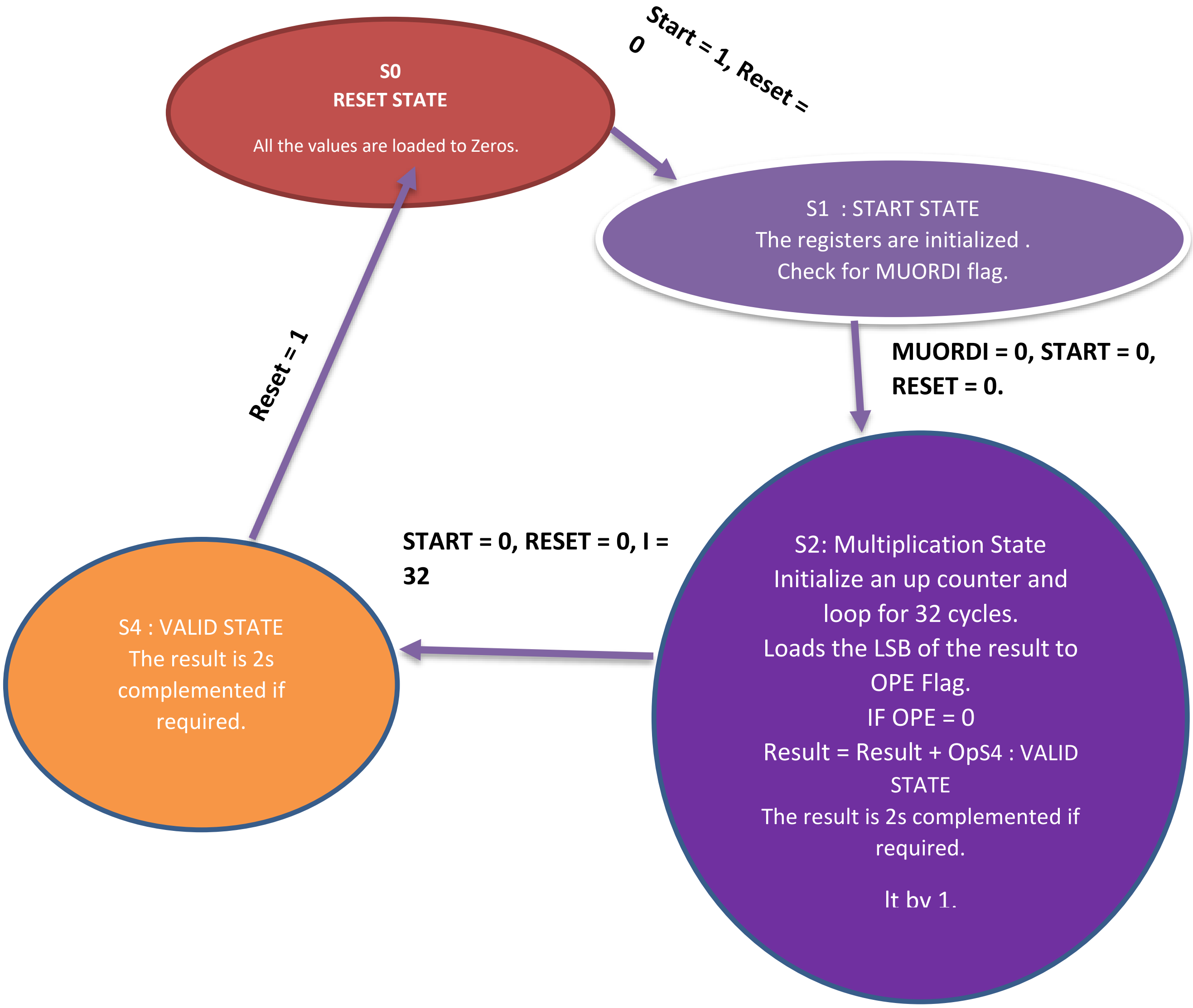

S0
RESET STATE
All the values are loaded to Zeros.
Start = 1, Reset = 0
S1 : START STATE
The registers are initialized .
Check for MUORDI flag.
MUORDI = 0, START = 0, RESET = 0.
Reset = 1
START = 0, RESET = 0, I = 32
S2: Multiplication State
Initialize an up counter and loop for 32 cycles.
Loads the LSB of the result to OPE Flag.
IF OPE = 0
Result = Result + OpS4 : VALID STATE
The result is 2s complemented if required.
It bv 1.
S4 : VALID STATE
The result is 2s complemented if required.

Figure II.3: State Transition Diagram of Sequential Divider

Discuss the timing characteristics of the multiplier and divider (less than 100 words)

Multiplier:
32 bit signed multiplier/ divider designed is to find the multiplication/ division result of two inputs given.
State S0 is the **Reset state** used to reset the result to Zeros i.e.– Reset State.
State S1 is the **Start state** used to initialize registers based on the inputs given. State S1 takes one cycle to be executed and depends on Start Input, which when transits from low to high and back to high for one clock, initializes in the next clock cycle.
State S2 : is for multiplication process. It takes 32 clock cycles to carry out the operation.
State S3: is for Division process. It takes 33 clock cycles to carry out the operation.
State S4: Valid State : The result is 2s complemented depending on the inputs

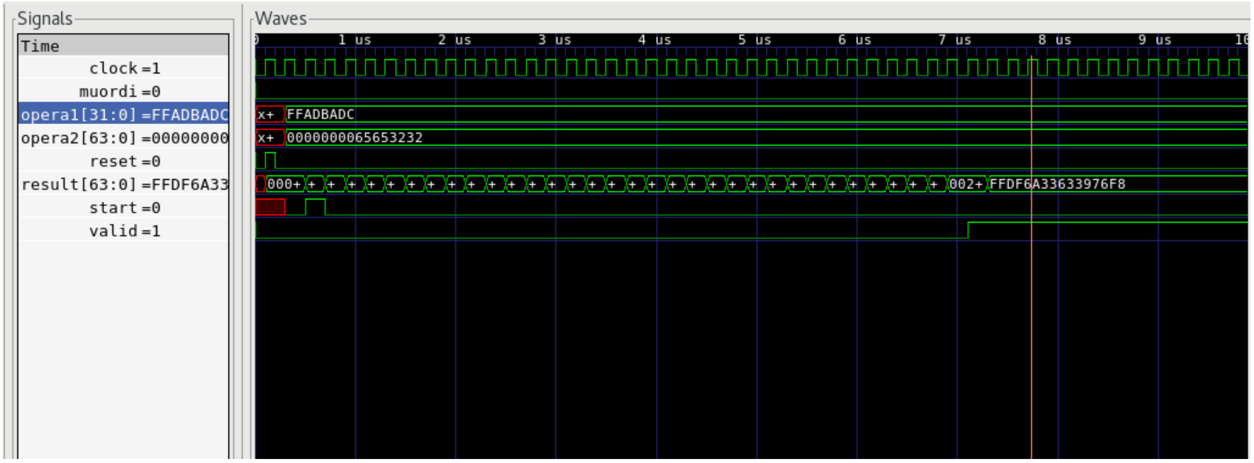
III. RTL-Level (Pre-synthesis) Simulations/Tests

***Short descriptions of testbench(es)** that were used to test your modules during the project implementation process (less than 50 words)*

*For each table in this section, you need to show simulation waveforms that can demonstrate major coverage in testing of your implementation and to support your expected results listed in the tables. Try to justify yourself if what parts of the tests and the changes from one value to other value that are important and should be included here in the report. **You should make some notes on the waveforms to show and explain your results and analyses.***

Table III.1 – Four Selected Test Data for Multiplier Circuit

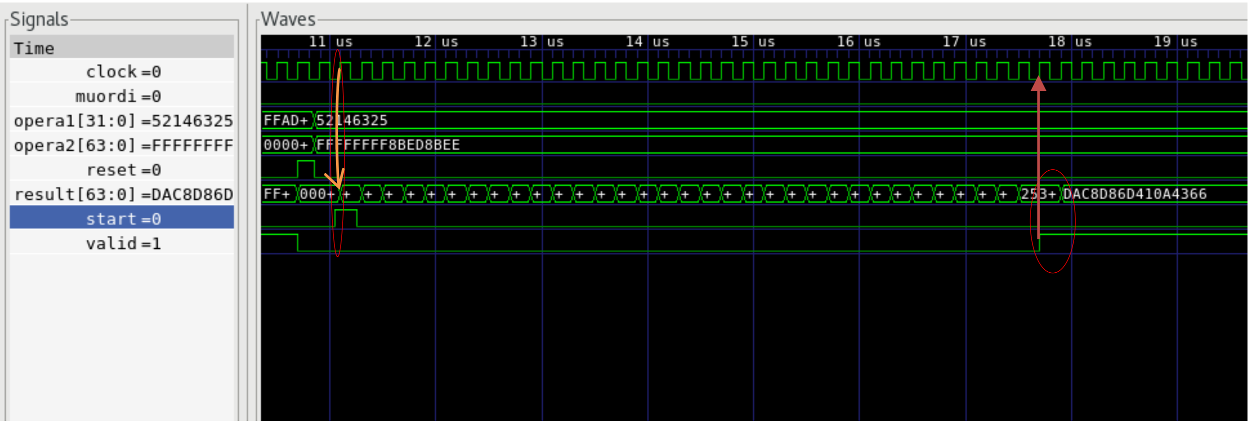
Test Case	`opera1` (hex)	`opra2` (hex)	`result (hex)`
1	-32'h00524524	64'h00000000_65653232	64'h FFDF6A33633976F8
2	32'h52146325	-64'h00000000_74127412	64'h DAC8D86D_410A4366
3	-32h7FFF_FFFF	-64'h00000000_65413654	64'h32A09B29_9ABEC9AC
4	32'h12123456	64'h00000000_21542369	64'h025A4938_28D73946



Time
clock =1
muordi =0
opera1[31:0] =FFADBADC
opra2[63:0] =00000000
reset =0
result[63:0] =FFDF6A33
start =0
valid =1

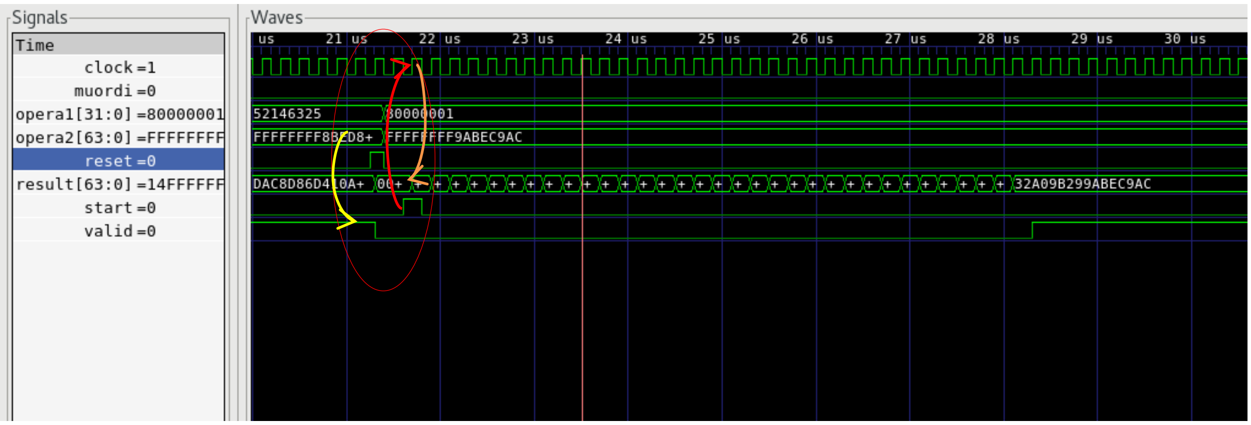
Figure III.1a: RTL simulation waveform that contains test case #1

As shown in Figure III. 1b and 1c, the start signal triggers the entire process. As the start signal goes high, valid signal goes low at the next positive edge of the clock, the result is loaded with the values in opera2. After 32 cycles, the valid goes high, which then converts the result in its 2s complement value if required.



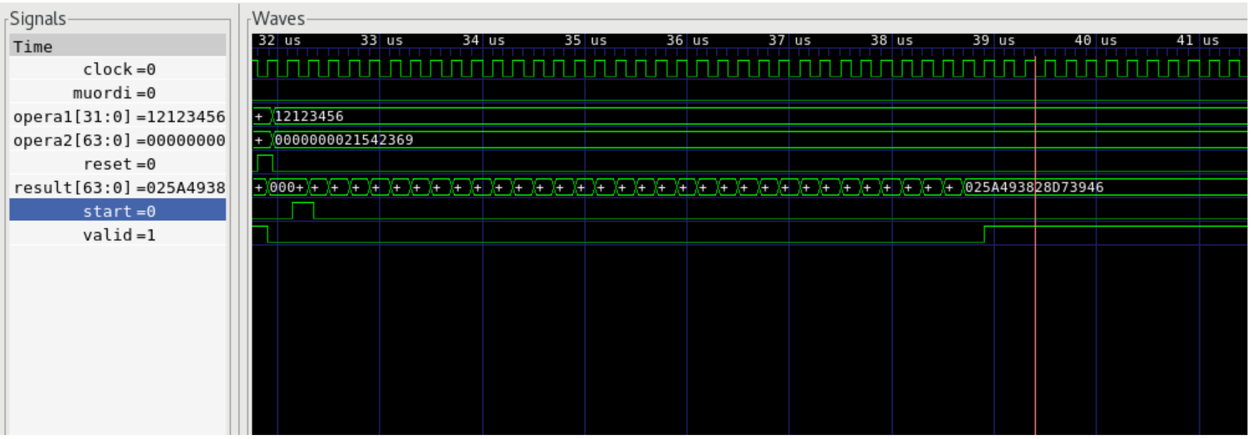
```
Signals
Time
clock =0
muordi =0
opera1[31:0] =52146325
opera2[63:0] =FFFFFFFF
reset =0
result[63:0] =DAC8D86D
start =0
valid =1
```

Figure III.1b: RTL simulation waveform that contains test case #2



```
Signals
Time
clock =1
muordi =0
opera1[31:0] =80000001
opera2[63:0] =FFFFFFFF
reset =0
result[63:0] =14FFFFFF
start =0
valid =0
```

Figure III.1c: RTL simulation waveform that contains test case #3

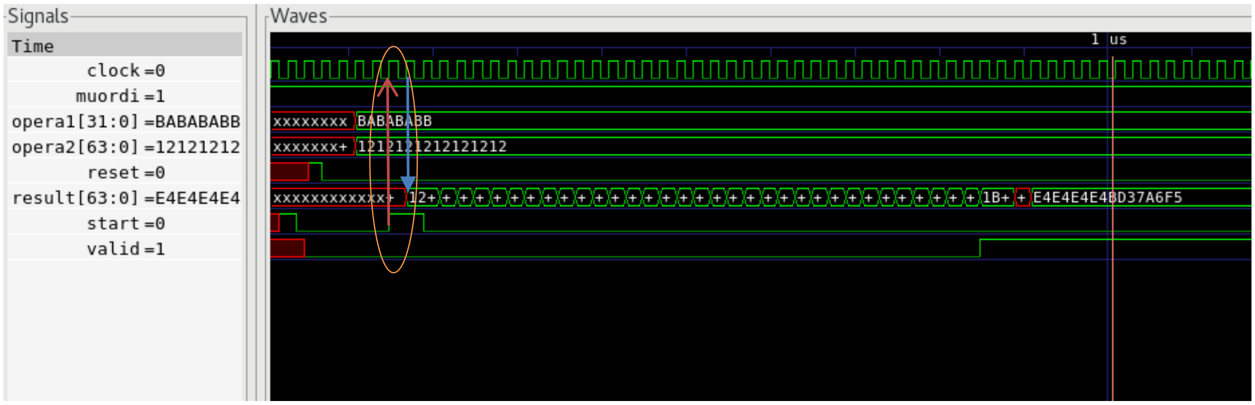


```
Signals
Time
clock =0
muordi =0
opera1[31:0] =12123456
opera2[63:0] =00000000
reset =0
result[63:0] =025A4938
start =0
valid =1
```

Figure III.1d: RTL simulation waveform that contains test case #4

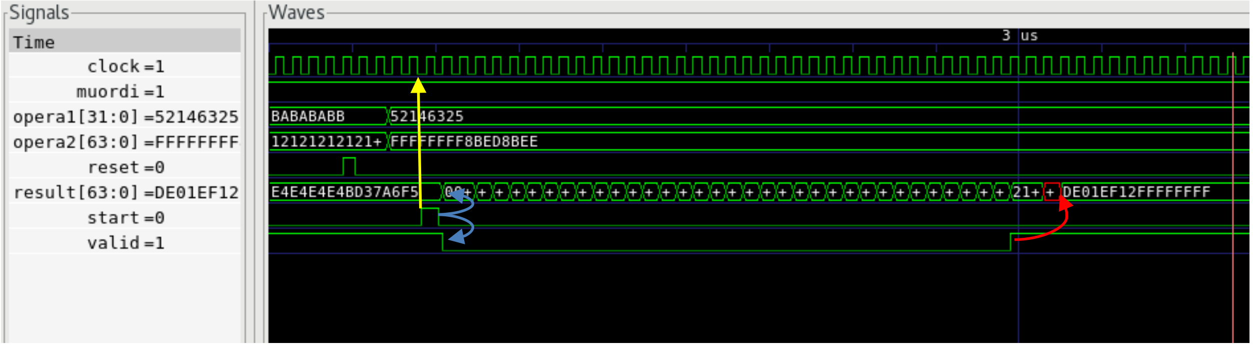
Table III.2 – Four Selected Test Data for Divider Circuit

Test Case	**opera1** (hex)	**opra2** (hex)	**result (hex)**
1	**-32'h45454545**	**64'h12121212_12121212**	**64'hE4E4E4E4_BD37A655**
2	**32'h52146325**	**-64'h00000000_74127412**	**64'hDE01EF12_FFFFFFFF**
3	**-32'h7fffffff**	**-64'h00000000_65413654**	**64'hE5413655_00000000**
4	**32'h524524**	**64'h00000000_65653232**	**64'h002A1EE6_0000013B**



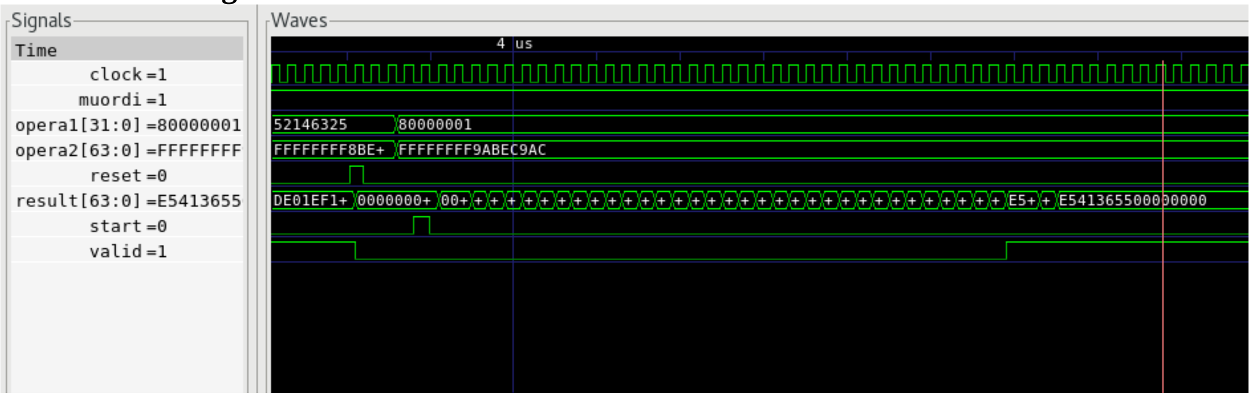
Signals
Time
clock =0
muordi =1
opera1[31:0] =BABABABB
opera2[63:0] =12121212
reset =0
result[63:0] =E4E4E4E4
start =0
valid =1

Figure III.2a: RTL simulation waveform that contains test case #1



Signals
Time
clock =1
muordi =1
opera1[31:0] =52146325
opera2[63:0] =FFFFFFFF
reset =0
result[63:0] =DE01EF12
start =0
valid =1

Figure III.2b: RTL simulation waveform that contains test case #2



Signals
Time
clock =1
muordi =1
opera1[31:0] =80000001
opera2[63:0] =FFFFFFFF
reset =0
result[63:0] =E5413655
start =0
valid =1

Figure III.2c: RTL simulation waveform that contains test case #3

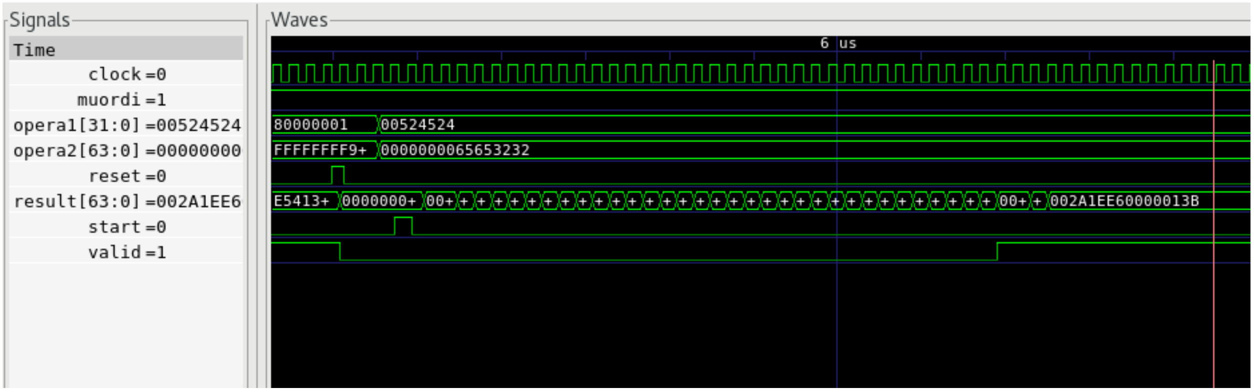


Figure III.2d: RTL simulation waveform that contains test case #4

As shown in figure III. 2a and 2b, when the start signal goes high, the next clock cycle, if the clock is still high, will load the values of opera2 in result, and carry out the operation. After 33 cycles, the valid signal goes high, which goes into the valid state, and if it needs to perform 2s complement, it will take 2 more clocks before displaying the output, however, if the result need not be 2s complemented, as shown in figure 2c and 2d, the value will maintain the value and display the value after the valid signal is high, till the next reset or start input.

IV. Synthesis and Optimizations

In each table below, show 6 selected trials that you have gone through in synthesizing and optimizing your implementations. ***You may try tens of trials, but you need to select 6 to show in the tables****. Select the trials that can represent your synthesizing and optimizing efforts so that you can conclude that the last trial (trial #6) is the one you finally used as the most optimized circuit.* ***Save the data from your (many) trials and use them to plot the "Area vs. Delay" curves in figures IV, which need many more data points.***

Table IV.1: Synthesis Constraints and Results for Multiplier – WCC library used

Trial #	**Your design constraint settings** (such as area, clock, delay, etc.)	**Results after synthesis** (such as area, time slack, power, etc.)
1	Area =2000, Clock =15	Area =3733.5, Time Slack =-4.94: violated, Data Arrival time =19.33, Power=2.7383
2	Area =4000, Clock =20	Area =3645.5, Time Slack =-0.01: violated, Data Arrival time =19.44, Power=1.9719
3	Area =2500, Clock =30	Area =3054.5, Time Slack =0.44, Data Arrival time =28.85, Power=1.3102
4	Area =3000, Clock =21	Area =3572, Time Slack =0.02, Data Arrival time =20.42, Power=1.908
5	Area =3500, Clock =22	Area =3458, Time Slack =0, Data Arrival time =21.44, Power=1.8465
6	Area =3000, Clock =25	Area =3157, Time Slack =0, Data Arrival time =24.4, Power=1.5992

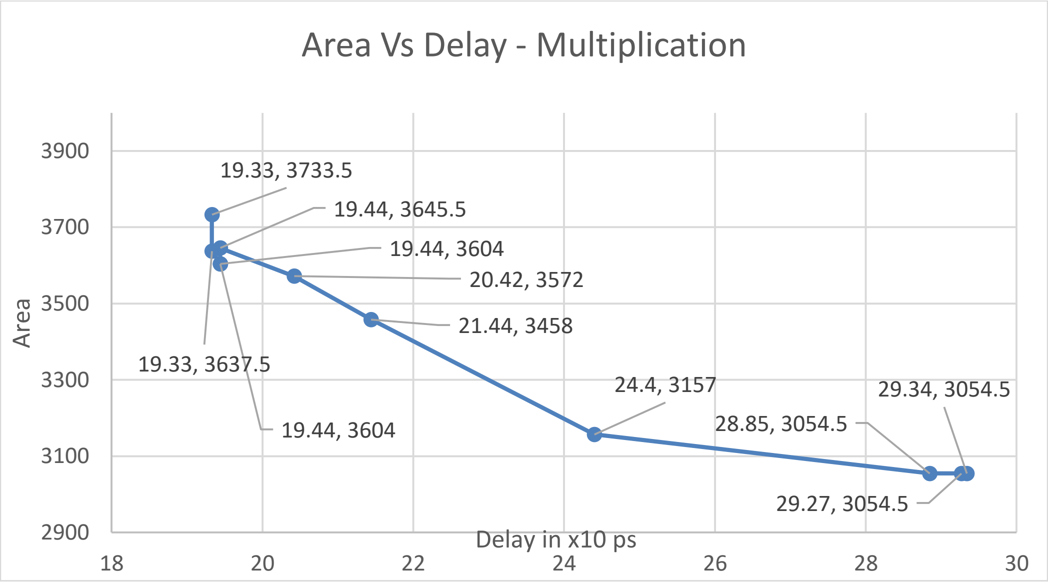


Figure IV.1: Area vs. Delay Curve of Multiplier

Table IV.2: Synthesis Constraints and Results for Divider

Trial #	**Your design constraint settings** (such as area, clock, delay, etc.)	**Results after synthesis** (such as area, time slack, power, etc.)
1	Area =0, Clock =20	Area =5387, Time Slack =-0.24: violated, Data Arrival time =19.68, Power=2.9153
2	Area =0, Clock =25	Area =4402.5, Time Slack =0, Data Arrival time =24.29, Power=2.1286
3	Area =3000, Clock =15	Area =5369, Time Slack =-5.41: violated, Data Arrival time =19.85, Power=3.877
4	Area =3000, Clock =40	Area =4199, Time Slack =9.81, Data Arrival time =29.47, Power=1.431
5	Area =4000, Clock =22	Area =4964.5, Time Slack =0, Data Arrival time =21.44, Power=2.4895
6	Area =3000, Clock =25	Area =4295, Time Slack =0, Data Arrival time =24.4, Power=2.1713

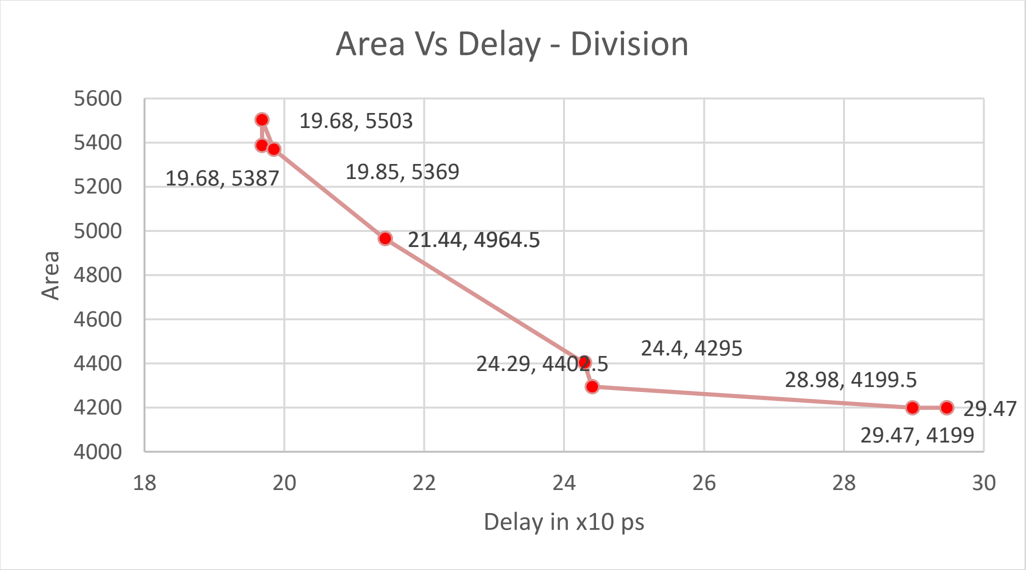


Figure IV.2: Area vs. Delay Curve of Divider

The timing of the circuit increases if we try to decrease the area i.e. Area is inversely proportional to time (data arrival time), and also power consumed is directly proportional to Area.
In the cases tried, an attempt to decrease the clock time, increased the area and power, and hence by taking a proper tradeoff and not affecting the other constraints much, the values mentioned in the table are the best optimized values for this circuit.

Data : **Area =3157, Time Slack =0, Data Arrival time =24.4, Power=1.5992 for Multiplier**

Data **: Area =4295, Time Slack =0, Data Arrival time =24.4, Power=2.1713** ***for Divider.***

However, both the circuits combined, since they share a lot of hardware, would be a more practical circuit. ***The Area required for a combined circuit after performing synthesis with the constraints used for the above mentioned cases, is, 5069 and data arrival time for 2.7789,*** *for a combined circuit, which may sound feasible even if the need of both the circuits at a time is not high.*

V. Gate-Level (Post-synthesis) Dynamic Simulations/Tests

For each table in this section, you need to show simulation waveforms that can demonstrate major coverage in testing of your implementation and to support your measured performance shown in the tables. The test data in these tables should be the same as test data in Section III tables, however for post-synthesis simulation, you need to select data points and adjust your timing scale of the displayed waveforms such that you are able to show the fluctuation of the immediate data through the circuits based on the values of the input operands and the correctness of the logic functions. ***You should make some notes on the waveforms to show and explain the timing delays that result in data fluctuations as you observed from the simulation result****.*

Table V.1 – Four Selected Test Data for Multiplier Circuit

Test Case	`opera1` (hex)	`opra2` (hex)	**`Time Delay (in time unit)`**
1	**-32'h00524524**	**64'h00000000_65653232**	**60 x 10 ps = 600 ps**
2	**32'h52146325**	**-64'h00000000_74127412**	**60 x 10 ps = 600 ps**
3	**-32h7FFF_FFFF**	**-64'h00000000_65413654**	**60 x 10 ps = 600 ps**
4	**32'h12123456**	**64'h00000000_21542369**	**60 x 10 ps = 600 ps**

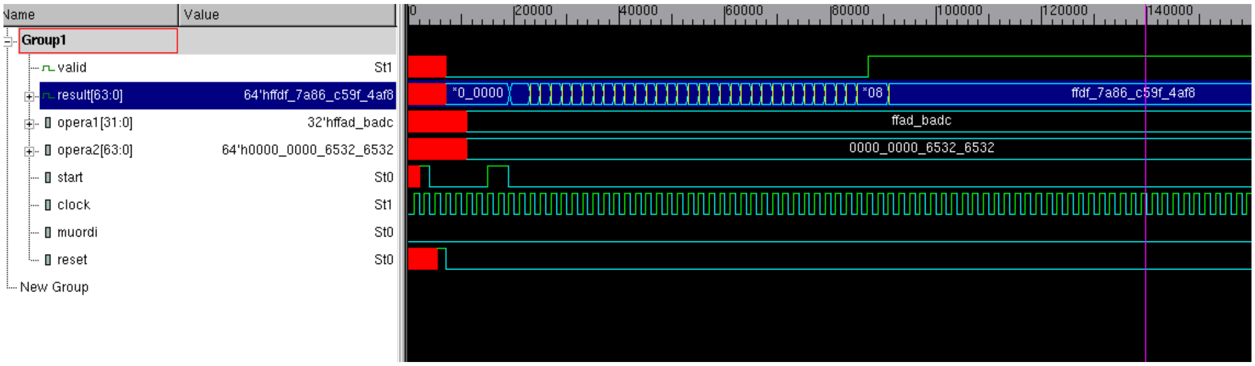


Figure V.1a: Gate-level simulation waveform that contains test case #1

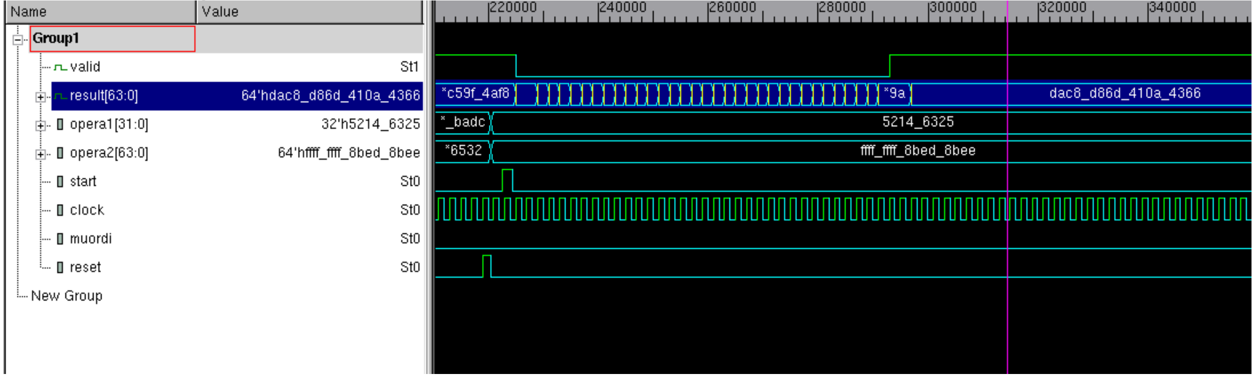
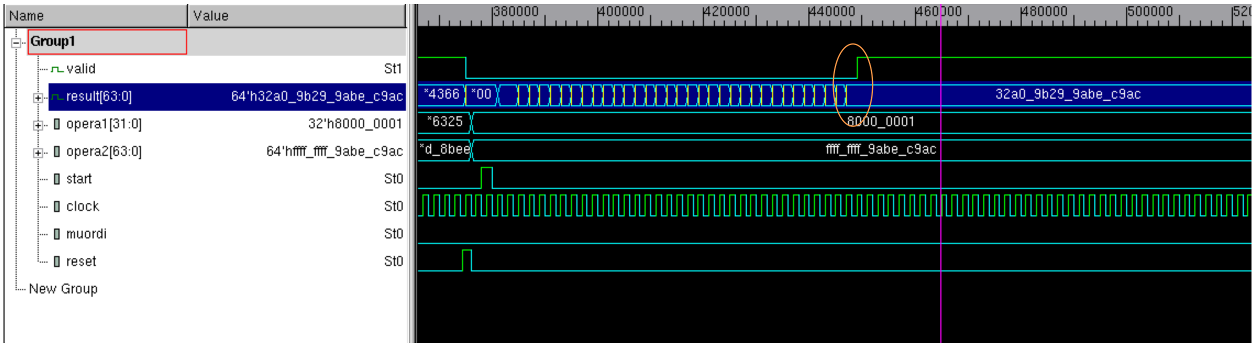
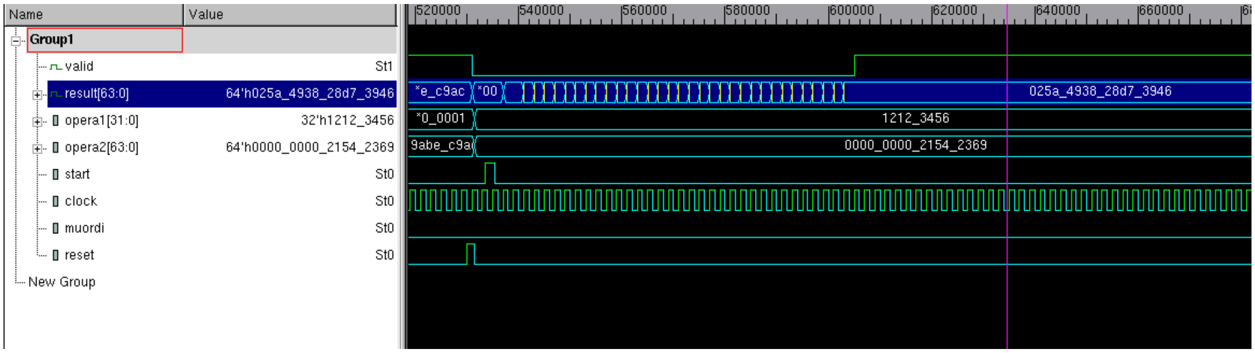


Figure V.1b: Gate-level simulation waveform that contains test case #2



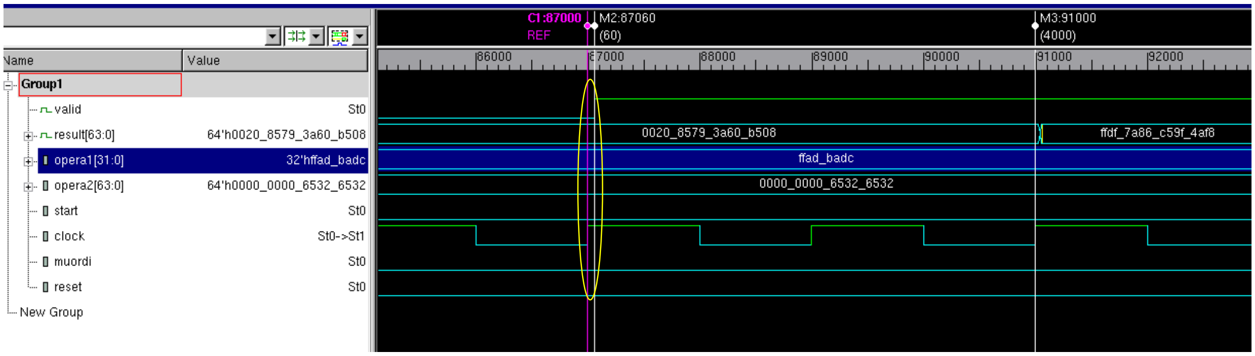
Name Value
Group1
valid St1
result[63:0] 64'h32a0_9b29_9abe_c9ac
opera1[31:0] 32'h8000_0001
opera2[63:0] 64'hffff_ffff_9abe_c9ac
start St0
clock St0
muordi St0
reset St0
New Group

Figure V.1c: Gate-level simulation waveform that contains test case #3



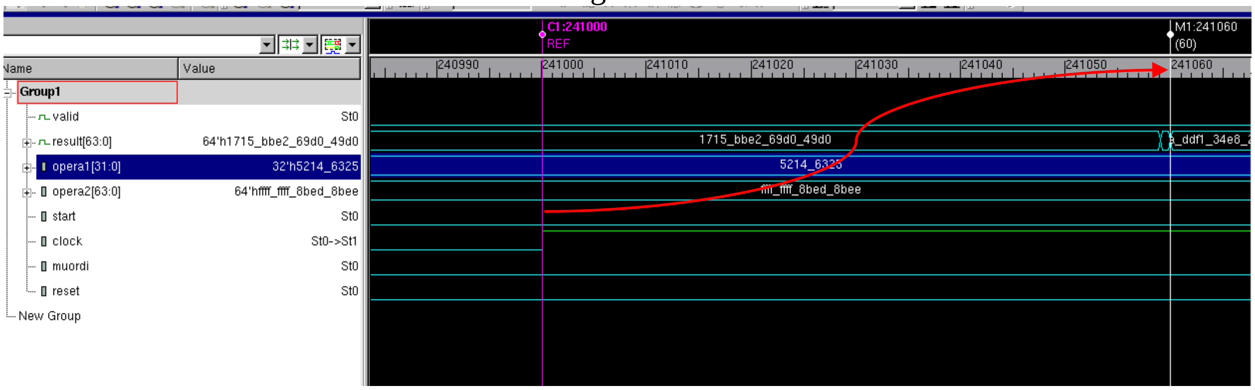
Name Value
Group1
valid St1
result[63:0] 64'h025a_4938_28d7_3946
opera1[31:0] 32'h1212_3456
opera2[63:0] 64'h0000_0000_2154_2369
start St0
clock St0
muordi St0
reset St0
New Group

Figure V.1d: Gate-level simulation waveform that contains test case #4



Name Value
Group1
valid St0
result[63:0] 64'h0020_8579_3a60_b508
opera1[31:0] 32'hffad_badc
opera2[63:0] 64'h0000_0000_6532_6532
start St0
clock St0->St1
muordi St0
reset St0
New Group

Fig V. 1e.



Name Value
Group1
valid St0
result[63:0] 64'h1715_bbe2_69d0_49d0
opera1[31:0] 32'h5214_6325
opera2[63:0] 64'hffff_ffff_8bed_8bee
start St0
clock St0->St1
muordi St0
reset St0
New Group

Fig. V. 1f

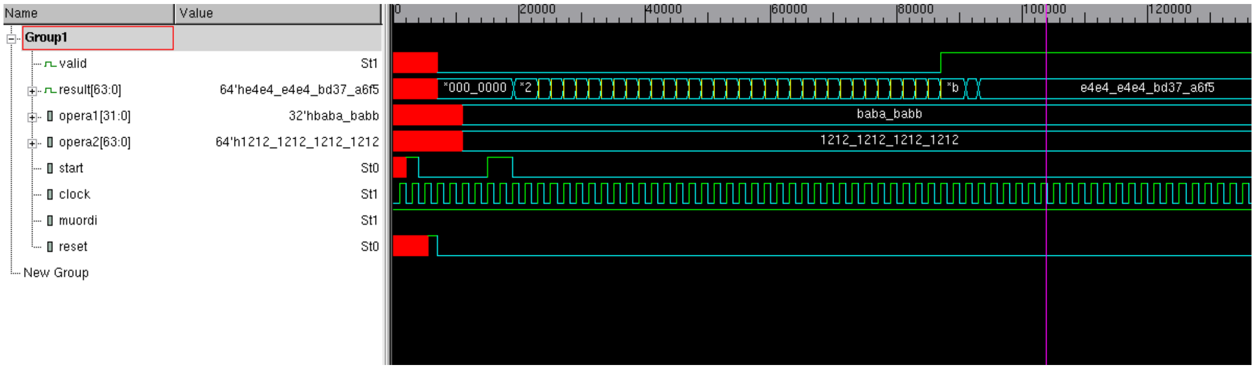
Delay Waveforms :

As shown in delay waveforms Fig V 1e and 1f, the delay between the positive edge trigger of the clock and the change of the result value is 60x10ps.

Also the valid signal goes high after the positive edge of the clock, due to which, it waits for one more positive edge of the clock, only after which it starts 2s complementing the result, which results in a total delay of extra 3 clock cycles.

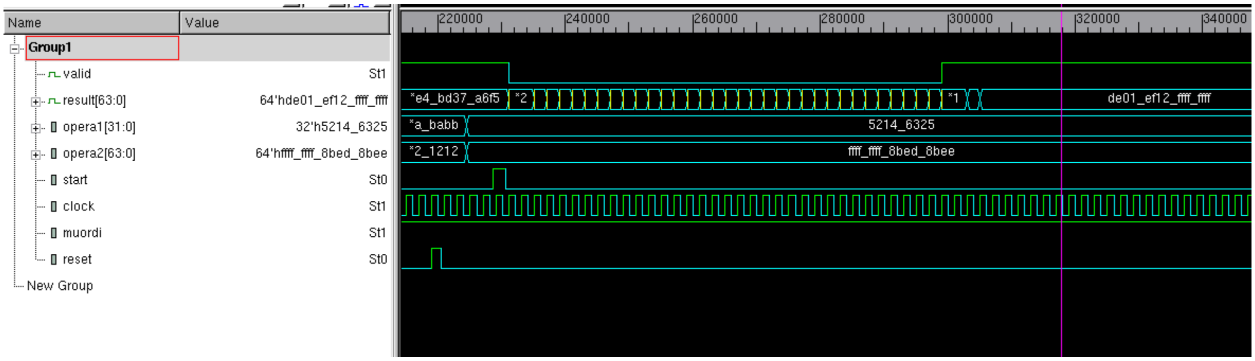
Table V.2 – Four Selected Test Data for Divider Circuit

Test Case	opera1 (hex)	opra2 (hex)	Time Delay (in time unit)
1	-32’h45454545	64’h12121212_12121212	60 x 10 ps = 600 ps
2	32’h52146325	-64’h00000000_74127412	60 x 10 ps = 600 ps
3	-32’h7fffffff	-64’h00000000_65413654	60 x 10 ps = 600 ps
4	32’h524524	64’h00000000_65653232	60 x 10 ps = 600 ps



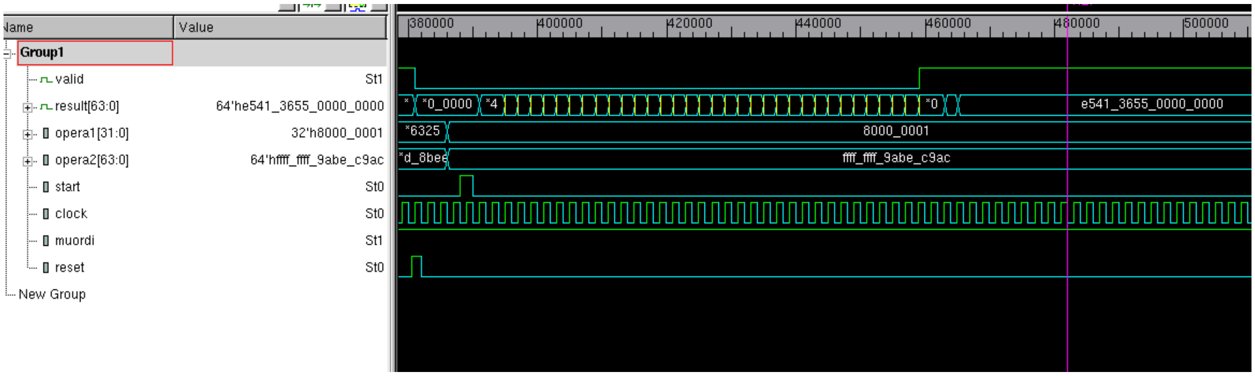
Name	Value
Group1	
valid	St1
result[63:0]	64'he4e4_e4e4_bd37_a6f5
opera1[31:0]	32'hbaba_babb
opera2[63:0]	64'h1212_1212_1212_1212
start	St0
clock	St1
muordi	St1
reset	St0
New Group	

Figure V.2a: Gate-level simulation waveform that contains test case #1



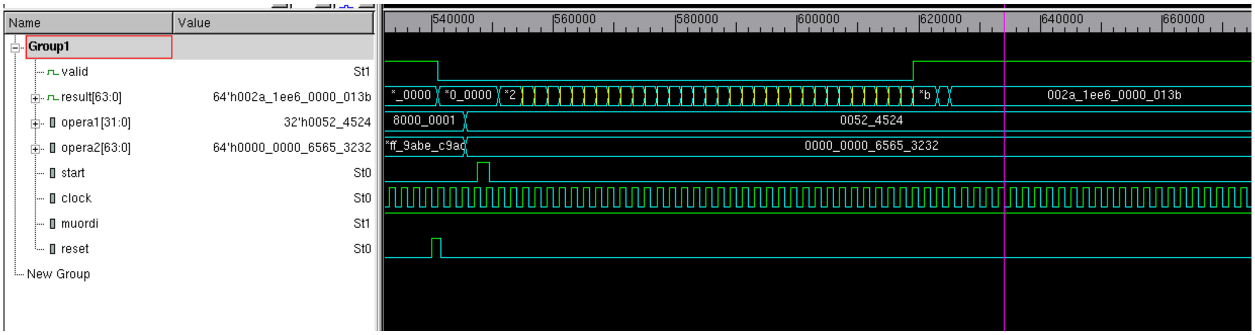
Name	Value
Group1	
valid	St1
result[63:0]	64'hde01_ef12_ffff_ffff
opera1[31:0]	32'h5214_6325
opera2[63:0]	64'hffff_ffff_8bed_8bee
start	St0
clock	St1
muordi	St1
reset	St0
New Group	

Figure V.2b: Gate-level simulation waveform that contains test case #2



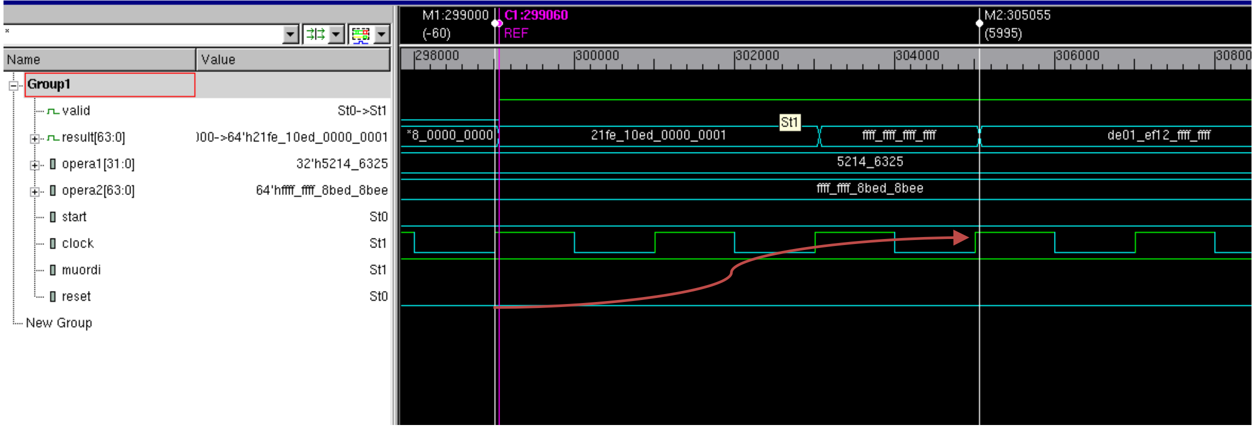
Name	Value
Group1	
valid	St1
result[63:0]	64'he541_3655_0000_0000
opera1[31:0]	32'h8000_0001
opera2[63:0]	64'hffff_ffff_9abe_c9ac
start	St0
clock	St0
muordi	St1
reset	St0
New Group	

Figure V.2c: Gate-level simulation waveform that contains test case #3



Name	Value
Group1	
valid	St1
result[63:0]	64'h002a_1ee6_0000_013b
opera1[31:0]	32'h0052_4524
opera2[63:0]	64'h0000_0000_6565_3232
start	St0
clock	St0
muordi	St1
reset	St0
New Group	

Figure V.2d: Gate-level simulation waveform that contains test case #4



Name	Value
Group1	
valid	St0->St1
result[63:0]	)00->64'h21fe_10ed_0000_0001
opera1[31:0]	32'h5214_6325
opera2[63:0]	64'hffff_ffff_8bed_8bee
start	St0
clock	St1
muordi	St1
reset	St0
New Group	

Multiplication : The multiplication of opera1 and opera2 always give a correct result for all the cases. It takes 34 clock cycles(1 for start, 32 clocks for the operation and 1 clock cycle extra for 2s complementing the result.) with an extra time delay of 600 ps.
Division : The division would take 35 clocks (1 for start, 33 clocks for operation and 1 clock extra for 2s complementing with a extra delay of 600 ps. The relevant graphs are attached. The delay between the clock risig edge and the valid set to 1 and the change of result to 2s complement takes 3 clock cycles.

VI. Conclusion

The multiplication of opera1 and opera2 always give a correct result for all the cases .However, for division process, the correctness of the output depends on the inputs provided. For both positive or negative inputs, the result would match the actual output; however, there are exceptions for different inputs. For example, in division, if opera 1 (divisor) is positive and opera2 (dividend) is negative, the result would be always negative, which would not cause any error. However, if opera1 is negative and opera2 is positive, the quotient would be negative and remainder would be positive which is not considered while 2s complementing the output. Also, if we have a quotient or a remainder larger than 32 bits, we wont be able to store the actual output received.

Eg: opera2 = 64’h545454545454 and opera1 = 32’h1; the result should be Quotient 545454545454 which is not a case which could be stored.

Appendix A

A.1 Contents from EDA Tool Configurations and Setup Files

```
set link_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
set target_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
set symbol_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.workview.sdb}
set synthetic_library {dw_foundation.sldb standard.sldb}
set_min_library     /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25     -
min_version /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_BCCOM25
```

A.2 Commands and/or Scripts Used for Simulation and Synthesis

```
set link_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
set target_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
set symbol_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.workview.sdb}
set synthetic_library {dw_foundation.sldb standard.sldb}
set_min_library     /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25     -
min_version /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_BCCOM25

read_verilog {multiplier.v}
current_design multiplier
link
create_clock clock -name clock -period 21.00
set_propagated_clock clock
set_clock_uncertainty .25 clock


set_max_area 3000
set_fix_hold clock
compile -map_effort high
report_cell
report_net
update_timing
report_timing -max_paths 10
report_area
report_power

report_timing >> m12_report_timing.txt

report_area >> m12_report_area.txt
report_power >> m12_report_power.txt


write -hierarchy -format verilog -output m12_netlist.v
quit
```

```
Compile :
vcs +v2k testbench_multiplier.v

Simulation:
./simv | tee sim_report.txt
```

Pre-synthesis Waveform:

```
Gtkwave multiplier.vcd &
```

Synthesis:

```
Dc_shell -xg -f synthesis.script | tee synreport.txt
```

Post Synthesis:

```
Ncverilog -y /apps/Toshiba/sjsu/Verilog/tc240c +libext+.tsbvlibp
+access+r multiplier_testbench.v multiplier.v

./simv |tee post_syn.txt
```

Appendix B
Completed Verilog Source Codes and Testbenches

(Use Courier New 10 font)

MULTIPLICATION DESIGN:

```
// Code your design here
`timescale 1ns/10ps
module multiplier (result, valid, opera1, opera2, muordi, clock, reset,
start);
//define inputs
input opera1,
opera2,
muordi,
reset,
start;
input clock;
wre clock;
// define output
output [63:0] result;
output valid;
// input size and type
wire [31:0] opera1;
wire [63:0] opera2;
reg ope_flag;

// output size and type
reg [63:0] result;
reg valid; //valid_dummy;

//define extra wires and registers required
wire c_out, cout;
wire [63:0] res_store;
wire [63:0] st_op2;
wire [31:0] z1, op1_not_wire, op2_not_wire, op1_store_not, op2_store_not;
wire [63:0] res_store_not, res_not_wire;
reg [63:0] res_not;
wire c_inop1, c_inop2, c_inres;
reg [31:0] op1_not, op2_not;
reg [63:0] result1;
reg sign1,sign2, sign, idle;
reg [31:0] op1_store, op2_store;
wire [31:0] op1_store_aswire;
integer i;
// For states
reg [2:0] cst, nst;
parameter S0 = 3'b000, //all state
          S1 = 3'b001,
          S2 = 3'b010,
          S3 = 3'b011,
          S4 = 3'b100,
     S5 = 3'b101;

//assign start = 1'b0;
```

```
assign op1_not_wire = op1_not;
assign op2_not_wire = op2_not;
assign c_in =1'b0;
assign st_op2 = result;
assign op1_store_aswire = op1_store_not;
assign res_not_wire = res_not;


initial begin
valid = 0;
end


// Multiplier instantiation
Add_rca M1 (res_store[63:32], , op1_store_aswire[31:0], st_op2[63:32], c_in);



always@(posedge clock or posedge reset)
begin
    if (reset)
    begin
        cst = S0;
    end
    else
    begin
        if (start)
        begin
               cst = S1;
        end
        else
        begin
            cst = nst;
        end
    end
end

always @(posedge clock)
case (cst)
S0 :    // reset state
        if (reset)
        begin
        valid = 0;
        //valid_dummy= 0;
        res_not = 64'h0;
//         result1 = 64'h0;
result = 64'h0;
i = 40;
        nst = cst;
        end

S1:       // start state
        if (muordi == 1'b0)
        begin
            sign = sign1^sign2;
            result [31:0] = op2_store_not;
            //result [31] = 1'b0;
            result [63:32] = 32'h0;
            ope_flag = 1'b0;
            valid = 1'b0;
            //valid_dummy = 1'b0;
            res_not = 64'h0;
            i = 0;
            nst = S2;
        end
        else
        begin
            /// Division logic
            nst = S3;
        end


S2:     // Multiplication State

    if (i<32)
    begin
        nst = cst;
        i =i+1;
        if (i>=0)
        begin
            ope_flag = result[0];
            if (ope_flag == 1'b1)
            begin
                result [63:32] = res_store[63:32];
                result = result>>1;
            end
            else
            begin
                result = result>>1;
            end
        end
    end
    else if (i==32)
    begin
        nst = S4;
        i =i+1;
        valid = 1'b1;
        nst = S4; /// valid state
        if (sign)
        begin
            res_not = (~result);
        end
        else
        begin
            res_not = result;
        end

    end

S3 : if (muordi==1)         /// division state
begin
end
```

```
S4 : if (valid)
begin
// valid state
result = res_store_not;
nst = S5; //idle state
idle = 1;
end

S5 : if (idle)
begin
end // idle state



default: nst = S0;
endcase

// 2ss complememt

/// instantiate for 2s complement
6

assign z1 = 32'h0;
notif1 A1(c_inop1, c_in,sign1);
bufif0 A2(c_inop1, c_in,sign1);
notif1 A3(c_inop2, c_in,sign2);
bufif0 A4(c_inop2, c_in,sign2);
notif1 A5(c_inres, c_in,sign);
bufif0 A6(c_inres, c_in,sign);

always@(opera1)
begin
sign1 = opera1[31];
if (sign1)
begin
op1_not = (~opera1);
end
else
begin
op1_not = opera1;
end
end

always @(opera2)
begin
sign2 = opera2[63];
if (sign2)
begin
op2_not = (~opera2[31:0]);
e
else
begin
op2_not = opera2[31:0];
end
end

endmodule
```

```
module Add_rca(sum, c_out, a, b, c_in);
output [31:0] sum;
output c_out;
input [31:0] a, b;
input c_in;
wire c_in8, c_out;
Add_rca_16 Z1 (sum[15:0], c_in8, a[15:0], b[15:0], c_in);
Add_rca_16 Z2 (sum[31:16], c_out, a[31:16], b[31:16], c_in8);
endmodule


module Add_rca_16 (sum, c_out, a, b, c_in);
output [15:0] sum;
output c_out;
input [15:0] a, b;
input c_in;
wire c_in4, c_in8, c_in12, c_out;
Add_rca_4 M1 (sum[3:0], c_in4, a[3:0], b[3:0], c_in);
Add_rca_4 M2 (sum[7:4], c_in8, a[7:4], b[7:4], c_in4);
Add_rca_4 M3 (sum[11:8], c_in12, a[11:8], b[11:8], c_in8);
Add_rca_4 M4 (sum[15:12], c_out, a[15:12], b[15:12], c_in12);
endmodule

module Add_rca_4 (sum, c_out, a, b, c_in);
output [3: 0] sum;
output c_out;
input [3: 0] a, b;
input c_in;
wire c_in2, c_in3, c_in4;
Add_full M1 (sum[0], c_in2, a[0], b[0], c_in);
Add_full M2 (sum[1], c_in3, a[1], b[1], c_in2);
Add_full M3 (sum[2], c_in4, a[2], b[2], c_in3);
Add_full M4 (sum[3], c_out, a[3], b[3], c_in4);
endmodule

module Add_full (sum, c_out, a, b, c_in);
output sum, c_out;
input a, b, c_in;
wire w1, w2, w3;
Add_half M1 (w1, w2, a, b);
Add_half M2 (sum, w3, w1, c_in);
or M3 (c_out, w2, w3);
endmodule

module Add_half (sum, c_out, a, b);
output sum, c_out;
input a, b;
xor M1 (sum, a, b);
and M2 (c_out, a, b);
endmodule
```

********TESTBENCH *******

```
`timescale 1 ns/10 ps

`include "multiplier.v"
module multiplier_tb();
reg [31:0] opera1;
reg [63:0] opera2;
reg start, clock, muordi, reset;
wire valid;
wire [63:0] result;

 multiplier M1 (result, valid, opera1, opera2, muordi, clock, reset, start);


initial begin
$monitor ($time,,"opera1=%h, opera2=%h, start=%h, clock=%b, muordi=%b,
reset=%b, valid=%b, result=%b", opera1, opera2, start, clock, muordi, reset,
valid, result);
end


initial begin

// clock=0;
muordi = 0;
#20 start = 1;
#20 start =0;
#15 reset = 1;
 #15 reset = 0;
 #40 opera1 = -32'h524524;
 opera2 = 64'h65326532;
 #40 start = 1;
 #40 start =0;
 #2000 reset = 1;
 #15 reset = 0;
 opera1 = 32'h52146325;
 opera2 = -64'h74127412;
 #20 start = 1;
 #20 start =0;
 #1500 reset = 1;
 #15 reset = 0;
 opera1 = -32'h7fffffff;
 opera2 = -64'h65413654;
 #20 start = 1;
 #20 start =0;
 #1500 reset = 1;
 #15 reset = 0;
 opera1 = 32'h12123456;
 opera2 = 64'h21542369;
 #20 start = 1;
 #20 start =0;
 #2000 $finish;

end





initial begin
//$display ("time\t opera1 opera2 start result valid");
//$monitor ("%g \t %h %h %h %h %h", $time, opera1, opera2, start, result,
valid);
$dumpfile("multiplier.vcd");

$dumpvars(0, multiplier_tb);

end





//clock assigned

initial begin

clock =0;

forever begin

#10 clock=~clock;

end

end

endmodule
```

```
DIVISION DESIGN :
// Code your design here

`timescale 1ns/10ps
module division(result, valid, opera1, opera2, muordi, clock, reset, start);
  input [31:0] opera1;
  input [63:0] opera2;
  output [63:0] result;
output valid;
  input start, muordi, reset;
input clock;
  wire [31:0] opera1;
  wire [63:0] opera2;
  reg [63:0] result;
  reg[63:0] res_store, //  used in 32nd cycle to store the values of
result ... extra reg.. should try and delete
op1_sub;     // used for storing subtract value
  wire start, muordi, c_in, reset, cout1, cout, c_inop1s;
  reg [31:0] op1_store, op2_store;   // stores the 2 complements of opera1
and 2
  reg sign, sign1, sign2, ope_flag;  // flags for storing 2s complement
  reg valid;
wire clock;   // output specs
  integer i;  //  for counter
  wire [31:0] res_sub, res_add;  // store the subtracted value and addition
value
  wire [63:0] st_op2;  // used since we cant use a reg in instantiation
  reg res_msb, idle;  // flag to check the result to be negative or positive
for algorithm

// values for 2s complement

wire [31:0] z1, op1_not_wire, op1_not_sub_wire, op1_store_not,
op1_sub_store_not;
wire [63:0] res_store_not, res_not_wire, op2_not_wire, op2_store_not;
reg [63:0] res_not;
wire c_inop1, c_inop2, c_inres;
reg [31:0] op1_not, op1_not_sub;
reg [63:0] op2_not;
assign op1_not_wire = op1_not;
assign op2_not_wire = op2_not;
assign op1_not_sub_wire = op1_not_sub;
assign res_not_wire = res_not;
assign z1 = 32'h0;
/////////////// reset logic
  assign st_op2 = result;  //store the changed value of result for next
addition/ subtraction
  assign c_in = 0;   // assign cin

//////////////////////////////////////////
reg [2:0] cst, nst;
parameter S0 = 3'b000, //all state
          S1 = 3'b001,
          S2 = 3'b010,
          S3 = 3'b011,
          S4 = 3'b100,
       S5 = 3'b101;


always@(posedge clock or posedge reset)
begin
      if (reset)
      begin
            cst <= S0;
      end
      else
      begin
            if (start)
            begin
                  cst <= S1;
            end
            else
            begin
                  cst <= nst;
            end
      end
end

always @(posedge clock)
case (cst)
S0 :    // reset state
            if (reset)
            begin
            valid = 0;
            //valid_dummy= 0;
            //sign = 0;
            //sign1=0;
            //sign2 = 0;
            //op1_store =32'h0;
            //op2_store = 64'h0;
            //op1_not = 32'h0;
            //op2_not = 64'h0;
            //op1_not_sub =32'h0;
            res_not = 64'h0;
            result = 64'h0;
            i = 35;
            nst = cst;
            end

S1:     // start state
            if (muordi == 1'b0)
            begin
                  sign = sign1^sign2;
                  result [31:0] = op2_store_not;
                  //result [31] = 1'b0;
                  result [63:32] = 32'h0;
                  ope_flag = 1'b0;
                  valid = 1'b0;
                  //valid_dummy = 1'b0;
                  res_not = 64'h0;
                  i = 0;
                  nst = S2;
            end
            else
            begin
                  /// Division logic
if (muordi==1'b1)
                  begin
                        sign = sign1^sign2;

                        ope_flag = 1'b1;
                        result = op2_store_not;
i = 0;
                        //valid_dummy = 1'b0;
                        valid = 1'b0;
                        nst = S3;
                  end
            end

S2: begin
/// Multiplication State COMMENTED FOR DIVISION STATE

      /*if (i<32)
      begin
            nst = cst;
            i =i+1;
            if (i>=0)
            begin
                  ope_flag = result[0];
                  if (ope_flag == 1'b1)
                  begin
                        result [63:32] = res_store[63:32];
                        result = result>>1;
                  end
                  else
                  begin
                        result = result>>1;
                  end
            end
      end
      else if (i==32)
      begin
            //valid = 1'b1;
            nst = S4;
            i =i+1;
            valid = 1'b1;
            nst = S4; /// valid state
            if (sign)
            begin
                  res_not = (~result);
            end
            else
            begin
                  res_not = result;
            end

      end*/
```

```
end



//////////////////////////////

S3 : if (muordi==1)           /// division state
begin
if (muordi == 1)
        begin
          if (i<32)
            begin
              nst = cst;
i = i+1;
              if (ope_flag == 1'b1)
                begin
                  result [63:32] = res_sub[31:0];
                  //res_store = result;
                  res_msb = result[63];
                  if (res_msb == 1'b0)
                    begin
                      result[63:0] = result[63:0] <<1;
                      result[0] = 1'b1;
                      ope_flag = 1'b1;
                    end
                  else if (res_msb == 1'b1)
                    begin
                      result = result<<1;
                      result[0] = 1'b0;
                      ope_flag = 1'b0;
                    end
                end
              else if(ope_flag == 1'b0)
                begin
                  result [63:32] = res_add[31:0];
                  res_msb = result[63];
                  if (res_msb == 1'b0)
                    begin
                      result = result<<1;
                      result[0] = 1'b1;
                      ope_flag = 1'b1;
                    end
                  else if (res_msb == 1'b1)
                    begin
                      result = result<<1;
                      result[0] = 1'b0;
                      ope_flag = 1'b0;
                    end
                end
            end

          else if (i==32)
            begin
              i = i+1;
      nst = S4;
            valid = 1'b1;
```

```
              if (ope_flag == 1'b1)
                begin
                  result [63:32] = res_sub[31:0];
                  res_store = result[31:0];
                  res_msb = result[63];
                        if     (res_msb == 1'b0)
                    begin
                      result[31:0] = res_store <<1;
                      result[0] = 1'b1;
                      ope_flag = 1'b1;
                      result[63:32] = res_sub[31:0];

                    end
                  else if (res_msb == 1'b1)
                    begin
                      result[31:0] = res_store <<1;
                      result[0] = 1'b0;
                      ope_flag = 1'b0;
                      result[63:32] = res_sub[31:0];
                    end
                end
              else if(ope_flag == 1'b0)
                begin
                  result [63:32] = res_add[31:0];   //checking the code .
uncomment once done
                  res_store = result[31:0];
                  res_msb = result[63];
                  if (res_msb == 1'b0)
                    begin
                      result[31:0] = res_store <<1;
                      result[0] = 1'b1;
                      ope_flag = 1'b1;
                      //result [63:32] = res_add[31:0];
                    end
                  else if (res_msb == 1'b1)
                    begin
                      result[31:0] = res_store <<1;
                      result[0] = 1'b0;   //CHANGED THE VALUE TO 1 FROM 0
MENTIONED IN THE CODE
                      ope_flag = 1'b0;
                            //result [63:32] = res_add[31:0];
                    end
                end
            end
        end

end
///////////////////////////////

S4 : if (valid)
begin
// valid state
valid = 1'b1;
result = res_store_not;
nst = S5; //idle state
idle = 1;
end
```

```
S5 : if (idle)
begin
end // idle state



default: nst = S0;
endcase






  //// cases end
///// Division Instantiation
///Subtract instantiation
  Add_rca_1 D1 (res_sub[31:0], , op1_sub_store_not, st_op2[63:32],c_in) ;

////add instantiation
  Add_rca_1 D2 (res_add[31:0], ,op1_store_not, st_op2[63:32],c_in);


///////2s complement


/// instantiate for 2s complement
Add_rca_1 C1 (op1_store_not, , op1_not_wire, z1, c_inop1);

Add_rca_1 C2 (op2_store_not[31:0],cout1, op2_not_wire[31:0], z1, c_inop2);
Add_rca_1 C3 (op2_store_not[63:32], , op2_not_wire[63:32], z1, cout1);

Add_rca_1 C4 (res_store_not[31:0], cout , res_not_wire[31:0], z1, c_inres);
Add_rca_1 C5 (res_store_not[63:32], , res_not_wire[63:32], z1, cout);

/// for sub opera1
Add_rca_1 C6 (op1_sub_store_not, , op1_not_sub_wire, z1, c_inop1s);



notif1 A1(c_inop1, c_in,sign1);
bufif0 A2(c_inop1, c_in,sign1);
notif1 A3(c_inop2, c_in,sign2);
bufif0 A4(c_inop2, c_in,sign2);
notif1 A5(c_inres, c_in,sign);
bufif0 A6(c_inres, c_in,sign);

//for sub value of opera1
notif0 A7(c_inop1s, c_in,sign1);
bufif1 A8(c_inop1s, c_in,sign1);

/// for opera1 - 2s complement if negative
always@(opera1)
begin
sign1 = opera1[31];
if(sign1)
```

```
begin
op1_not = (~opera1);
op1_not_sub = opera1;
end
else
begin
op1_not = opera1;
op1_not_sub = (~opera1);
end
end



/////////////// FOR result - 2s complement if negative /// 2s complement the
ouput if negative. based on valid bit
/*
always@(posedge valid_dummy)
begin
sign = sign1 ^ sign2;
valid = 1'b1;
end
*/

/// for opera 2 - 2s complement if negative
always @(opera2)
begin
sign2 = opera2[63];

if (sign2)
begin
op2_not = (~opera2);
end
else
begin
op2_not = opera2;
end
end

endmodule


module Add_rca_1 (sum, c_out, a, b, c_in);
output [31:0] sum;
output c_out;
input [31:0] a, b;
input c_in;
wire c_in8, c_out;
Add_rca_16_1 Z1 (sum[15:0], c_in8, a[15:0], b[15:0], c_in);
Add_rca_16_1 Z2 (sum[31:16], c_out, a[31:16], b[31:16], c_in8);
endmodule


module Add_rca_16_1 (sum, c_out, a, b, c_in);
output [15:0] sum;
output c_out;
input [15:0] a, b;
input c_in;
```

```
wire c_in4, c_in8, c_in12, c_out;
Add_rca_4_1 M1 (sum[3:0], c_in4, a[3:0], b[3:0], c_in);
Add_rca_4_1 M2 (sum[7:4], c_in8, a[7:4], b[7:4], c_in4);
Add_rca_4_1 M3 (sum[11:8], c_in12, a[11:8], b[11:8], c_in8);
Add_rca_4_1 M4 (sum[15:12], c_out, a[15:12], b[15:12], c_in12);
endmodule

module Add_rca_4_1 (sum, c_out, a, b, c_in);
output [3: 0] sum;
output c_out;
input [3: 0] a, b;
input c_in;
wire c_in2, c_in3, c_in4;
Add_full_1 M1 (sum[0], c_in2, a[0], b[0], c_in);
Add_full_1 M2 (sum[1], c_in3, a[1], b[1], c_in2);
Add_full_1 M3 (sum[2], c_in4, a[2], b[2], c_in3);
Add_full_1 M4 (sum[3], c_out, a[3], b[3], c_in4);
endmodule

module Add_full_1 (sum, c_out, a, b, c_in);
output sum, c_out;
input a, b, c_in;
wire w1, w2, w3;
Add_half_1 M1 (w1, w2, a, b);
Add_half_1 M2 (sum, w3, w1, c_in);
or M3 (c_out, w2, w3);
endmodule

module Add_half_1 (sum, c_out, a, b);
output sum, c_out;
input a, b;
xor M1 (sum, a, b);
and M2 (c_out, a, b);
endmodule
```

TESTBENCH: DIVISION

```
`timescale 1 ns/10 ps
`include "division.v"
module testbench_division();
reg [31:0] opera1;
reg [63:0] opera2;
reg start, clock, muordi, reset;
wire valid;
wire [63:0] result;
division A1 (result, valid, opera1, opera2, muordi, clock, reset, start);

initial begin
$monitor ($time,,"opera1=%h, opera2=%h, start=%h, muordi=%b, reset=%b,
valid=%b, result=%h", opera1, opera2, start, muordi, reset, valid, result);
end


initial begin

// clock=0;
muordi = 1;
#20 start = 1;
#20 start =0;
#15 reset = 1;
 #15 reset = 0;
#40 opera1 = -32'h2;
opera2 = 64'h18;
 #40 start = 1;
 #40 start =0;
 #2000 reset = 1;
 #15 reset = 0;
#40 opera1 = 32'h52146325;
opera2 = -64'h74127412;
#40 start = 1;
 #20 start =0;
 #1500 reset = 1;
 #15 reset = 0;
 #40 opera1 = -32'h7fffffff;
opera2 = -64'h65413654;
#20 start = 1;
 #20 start =0;
 #1500 reset = 1;
 #15 reset = 0;
 #40 opera1 = 32'h524524;
opera2 = 64'h65653232;
#20 start = 1;
 #20 start =0;
 #2000 $finish;

end
```

```
initial begin
$dumpfile("division.vcd");
$dumpvars(0, testbench_division);

end


//clock assigned

initial begin
clock =0;
forever begin
#10 clock=~clock;
end
end
endmodule
```

Appendix C
Reports and Circuits from EDA Tools

(Use Courier New 10 font)

C.1 Contents of Selected Reports from RTL (Pre-synthesis) Simulations (VCS or NCVERILOG)

```
Multiplication
```

```
Chronologic VCS simulator copyright 1991-2014
Contains Synopsys proprietary information.
Compiler version I-2014.03-2; Runtime version I-2014.03-2;  Dec 12 05:24
2016
                   0 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=x,
muordi=0, reset=x, valid=0, result=xxxxxxxxxxxxxxxx
                  20 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=1,
muordi=0, reset=x, valid=0, result=xxxxxxxxxxxxxxxx
                  30 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=1,
muordi=0, reset=x, valid=0, result=00000000xxxxxxxx
                  40 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=0, reset=x, valid=0, result=00000000xxxxxxxx
                  50 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=0, reset=x, valid=0, result=00000000Xxxxxxxx
                  55 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=0, reset=1, valid=0, result=00000000Xxxxxxxx
                  70 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=0, reset=0, valid=0, result=00000000Xxxxxxxx
                  90 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=0, reset=0, valid=0, result=00000000Xxxxxxxx
                 110 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000000000xxxxxxx
                 130 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000000000Xxxxxxx
                 150 opera1=ffadbadc, opera2=0000000065326532, start=1,
muordi=0, reset=0, valid=0, result=0000000065326532
                 190 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0000000032993299
                 210 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00292292194c994c
                 230 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=001491490ca64ca6
                 250 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000a48a486532653
                 270 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=002e46e443299329
                 290 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=004046042194c994
                 310 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0020230210ca64ca
                 330 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0010118108653265
```

```
                 350 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00312b5284329932
                 370 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=001895a942194c99
                 390 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00356d66a10ca64c
                 410 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=001ab6b350865326
                 430 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000d5b59a8432993
                 450 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=002fd03ed42194c9
                 470 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00410ab16a10ca64
                 490 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00208558b5086532
                 510 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=001042ac5a843299
                 530 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=003143e82d42194c
                 550 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0018a1f416a10ca6
                 570 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000c50fa0b508653
                 590 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=002f4b0f05a84329
                 610 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0040c81982d42194
                 630 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0020640cc16a10ca
                 650 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=0010320660b50865
                 670 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00313b95305a8432
                 690 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00189dca982d4219
                 710 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=003571774c16a10c
                 730 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=001ab8bba60b5086
                 750 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=000d5c5dd305a843
                 770 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=002fd0c0e982d421
                 790 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=00410af274c16a10
                 810 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=0, result=002085793a60b508
                 830 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=1, result=002085793a60b508
                 850 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=0, valid=1, result=ffdf7a86c59f4af8
                2190 opera1=ffadbadc, opera2=0000000065326532, start=0,
muordi=0, reset=1, valid=0, result=0000000000000000
```

```
                2205 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0000000000000000
                2225 opera1=52146325, opera2=ffffffff8bed8bee, start=1,
muordi=0, reset=0, valid=0, result=0000000000000000
                2230 opera1=52146325, opera2=ffffffff8bed8bee, start=1,
muordi=0, reset=0, valid=0, result=0000000074127412
                2245 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0000000074127412
                2250 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=000000003a093a09
                2270 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=290a31929d049d04
                2290 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=148518c94e824e82
                2310 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0a428c64a7412741
                2330 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=2e2b77c4d3a093a0
                2350 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=1715bbe269d049d0
                2370 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0b8addf134e824e8
                2390 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=05c56ef89a741274
                2410 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=02e2b77c4d3a093a
                2430 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=01715bbe269d049d
                2450 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=29c2df71934e824e
                2470 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=14e16fb8c9a74127
                2490 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=337ae96ee4d3a093
                2510 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=42c7a649f269d049
                2530 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=4a6e04b77934e824
                2550 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=2537025bbc9a7412
                2570 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=129b812dde4d3a09
                2590 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=3257f2296f269d04
                2610 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=192bf914b7934e82
                2630 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0c95fc8a5bc9a741
                2650 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=2f552fd7ade4d3a0
                2670 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=17aa97ebd6f269d0
                2690 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=0bd54bf5eb7934e8
```

```
                2710 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=05eaa5faf5bc9a74
                2730 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=02f552fd7ade4d3a
                2750 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=017aa97ebd6f269d
                2770 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=29c78651deb7934e
                2790 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=14e3c328ef5bc9a7
                2810 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=337c1326f7ade4d3
                2830 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=42c83b25fbd6f269
                2850 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=4a6e4f257deb7934
                2870 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=0, result=25372792bef5bc9a
                2890 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=1, result=25372792bef5bc9a
                2910 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=0, valid=1, result=dac8d86d410a4366
                3745 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=1, valid=1, result=dac8d86d410a4366
                3750 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=0, reset=1, valid=0, result=0000000000000000
                3760 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=0000000000000000
                3780 opera1=80000001, opera2=ffffffff9abec9ac, start=1,
muordi=0, reset=0, valid=0, result=0000000000000000
                3790 opera1=80000001, opera2=ffffffff9abec9ac, start=1,
muordi=0, reset=0, valid=0, result=0000000065413654
                3800 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=0000000065413654
                3810 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=0000000032a09b2a
                3830 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=0000000019504d95
                3850 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=3fffffff8ca826ca
                3870 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=1fffffffc6541365
                3890 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=4fffffff632a09b2
                3910 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=27ffffffb19504d9
                3930 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=53ffffff58ca826c
                3950 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=29ffffffac654136
                3970 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=14ffffffd632a09b
                3990 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=4a7fffff6b19504d
```

```
                4010 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=653fffff358ca826
                4030 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=329fffff9ac65413
                4050 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=594fffff4d632a09
                4070 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=6ca7ffff26b19504
                4090 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=3653ffff9358ca82
                4110 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=1b29ffffc9ac6541
                4130 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=4d94ffff64d632a0
                4150 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=26ca7fffb26b1950
                4170 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=13653fffd9358ca8
                4190 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=09b29fffec9ac654
                4210 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=04d94ffff64d632a
                4230 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=026ca7fffb26b195
                4250 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=413653ff7d9358ca
                4270 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=209b29ffbec9ac65
                4290 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=504d94ff5f64d632
                4310 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=2826ca7fafb26b19
                4330 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=5413653f57d9358c
                4350 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=2a09b29fabec9ac6
                4370 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=1504d94fd5f64d63
                4390 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=4a826ca76afb26b1
                4410 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=65413653357d9358
                4430 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=0, result=32a09b299abec9ac
                4450 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=0, valid=1, result=32a09b299abec9ac
                5300 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=1, valid=1, result=32a09b299abec9ac
                5310 opera1=80000001, opera2=fffffff9abec9ac, start=0,
muordi=0, reset=1, valid=0, result=0000000000000000
                5315 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0000000000000000
                5335 opera1=12123456, opera2=0000000021542369, start=1,
muordi=0, reset=0, valid=0, result=0000000000000000
```

```
                5350 opera1=12123456, opera2=0000000021542369, start=1,
muordi=0, reset=0, valid=0, result=0000000021542369
                5355 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0000000021542369
                5370 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=09091a2b10aa11b4
                5390 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=04848d15885508da
                5410 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0242468ac42a846d
                5430 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0a2a3d7062154236
                5450 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=05151eb8310aa11b
                5470 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0b93a9871885508d
                5490 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0ed2eeee8c42a846
                5510 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0769777746215423
                5530 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0cbdd5e6a310aa11
                5550 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0f68051e51885508
                5570 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=07b4028f28c42a84
                5590 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=03da014794621542
                5610 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=01ed00a3ca310aa1
                5630 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=09ff9a7ce5188550
                5650 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=04ffcd3e728c42a8
                5670 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=027fe69f39462154
                5690 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=013ff34f9ca310aa
                5710 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=009ff9a7ce518855
                5730 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=095916fee728c42a
                5750 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=04ac8b7f73946215
                5770 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0b5f5feab9ca310a
                5790 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=05afaff55ce51885
                5810 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0be0f225ae728c42
                5830 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=05f07912d7394621
                5850 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0c0156b46b9ca310
```

```
                5870 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=0600ab5a35ce5188
                5890 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=030055ad1ae728c4
                5910 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=01802ad68d739462
                5930 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=00c0156b46b9ca31
                5950 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=096924e0a35ce518
                5970 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=04b4927051ae728c
                5990 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=0, result=025a493828d73946
                6010 opera1=12123456, opera2=0000000021542369, start=0,
muordi=0, reset=0, valid=1, result=025a493828d73946
$finish called from file "testbench_multiplier.v", line 59.
$finish at simulation time               735500
           V C S   S i m u l a t i o n   R e p o r t
Time: 7355000 ps
CPU Time:      0.170 seconds;       Data structure size:   0.0Mb
Mon Dec 12 05:24:23 2016
```

```
Division
```

```
Chronologic VCS simulator copyright 1991-2014
Contains Synopsys proprietary information.
Compiler version I-2014.03-2; Runtime version I-2014.03-2;  Dec 12 05:19
2016
                   0 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=x,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                  20 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=1,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                  40 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                  50 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=x, valid=0, result=xxxxxxxxxxxxxxxx
                  55 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=1, valid=0, result=xxxxxxxxxxxxxxxx
                  70 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=0, valid=0, result=xxxxxxxxxxxxxxxx
                 110 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=xxxxxxxxxxxxxxxx
                 150 opera1=bababab, opera2=1212121212121212, start=1,
muordi=1, reset=0, valid=0, result=xxxxxxxxxxxxxxxx
                 170 opera1=bababab, opera2=1212121212121212, start=1,
muordi=1, reset=0, valid=0, result=1212121212121212
                 190 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=1212121212121212
                 210 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=9999999a24242424
                 230 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=bdbdbdbe48484848
                 250 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=0606060690909091
                 270 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=8181818321212122
                 290 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=8d8d8d9042424244
                 310 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=a5a5a5aa84848488
                 330 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=d5d5d5df09090910
                 350 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=3636364812121221
                 370 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=e1e1e20624242442
                 390 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=4e4e4e9648484885
                 410 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=121212a29090910b
                 430 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=99999abb21212216
                 450 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=bdbdc0004242442c
                 470 opera1=bababab, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=06060a8a84848859
```

```
                 490 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=81818a8b090910b2
                 510 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=8d8d9fa012122164
                 530 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=a5a5c9ca242442c8
                 550 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=d5d61e1e48488590
                 570 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=3636c6c690910b21
                 590 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=e1e3030321221642
                 610 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=4e50909042442c85
                 630 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=121696968488590b
                 650 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=99a2a2a30910b216
                 670 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=bdcfcfd01221642c
                 690 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=062a2a2a2442c859
                 710 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=81c9c9ca488590b2
                 730 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=8e1e1e1e910b2164
                 750 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=a6c6c6c7221642c8
                 770 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=d8181818442c8590
                 790 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=3abababa88590b21
                 810 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=eaeaeaeb10b21642
                 830 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=6060606021642c85
                 850 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=1, result=1b1b1b1b42c8590b
                 890 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=1, result=xxxxxxxxxxxxxxxx
                 910 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=1, result=e4e4e4e4bd37a6f5
                2190 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=1, valid=1, result=e4e4e4e4bd37a6f5
                2205 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=1, result=e4e4e4e4bd37a6f5
                2245 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=1, result=e4e4e4e4bd37a6f5
                2285 opera1=52146325, opera2=ffffffff8bed8bee, start=1,
muordi=1, reset=0, valid=1, result=e4e4e4e4bd37a6f5
                2305 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=1, result=e4e4e4e4bd37a6f5
                2310 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=0000000074127412
```

```
                2350 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739b6e824e824
                2370 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739b7d049d048
                2390 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739b9a093a090
                2410 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739bd41274120
                2430 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739c4824e8240
                2450 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739d3049d0480
                2470 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd739f0093a0900
                2490 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd73a2a12741200
                2510 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd73a9e24e82400
                2530 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd73b8649d04800
                2550 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd73d5693a09000
                2570 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd740f727412000
                2590 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd748384e824000
                2610 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd756ba9d048000
                2630 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd773bf3a090000
                2650 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd7adc874120000
                2670 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd821dae8240000
                2690 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bd909ffd0480000
                2710 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bdada49a0900000
                2730 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bde7add41200000
                2750 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5be5bc0482400000
                2770 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5bf43e5304800000
                2790 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5c1142f009000000
                2810 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5c4b4c2a12000000
                2830 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5cbf5e9e24000000
                2850 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5da7838648000000
                2870 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=5f77cd5690000000
```

```
                2890 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=631860f720000000
                2910 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=6a59883840000000
                2930 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=78dbd6ba80000000
                2950 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=95e073bf00000000
                2970 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=cfe9adc800000000
                2990 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=1, result=21fe10ed00000001
                3030 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=1, result=xxxxxxxxxxxxxxxx
                3050 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=1, result=de01ef12ffffffff
                3805 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=1, valid=1, result=de01ef12ffffffff
                3810 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=1, valid=0, result=0000000000000000
                3820 opera1=52146325, opera2=ffffffff8bed8bee, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                3860 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                3880 opera1=80000001, opera2=ffffffff9abec9ac, start=1,
muordi=1, reset=0, valid=0, result=0000000000000000
                3900 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                3910 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000065413654
                3950 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00000002ca826ca8
                3970 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000000039504d950
                3990 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000000052a09b2a0
                4010 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000854136540
                4030 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000ea826ca80
                4050 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000001b504d9500
                4070 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00000034a09b2a00
                4090 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000006741365400
                4110 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000000cc826ca800
                4130 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000019704d95000
                4150 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000032c09b2a000
                4170 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000065613654000
```

```
                4190 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00000caa26ca8000
                4210 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000019524d950000
                4230 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000032a29b2a0000
                4250 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000654336540000
                4270 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000ca846ca80000
                4290 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00019506d9500000
                4310 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00032a0bb2a00000
                4330 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0006541565400000
                4350 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=000ca828ca800000
                4370 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0019504f95000000
                4390 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0032a09d2a000000
                4410 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0065413854000000
                4430 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=00ca826ea8000000
                4450 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=019504db50000000
                4470 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=032a09b4a0000000
                4490 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0654136740000000
                4510 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0ca826cc80000000
                4530 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=19504d9700000000
                4550 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=32a09b2c00000000
                4570 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=6541365600000000
                4590 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=1, result=e541365500000000
                4630 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=1, result=0000000000000000
                4650 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=1, result=e541365500000000
                5400 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=1, valid=1, result=e541365500000000
                5410 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=1, valid=0, result=0000000000000000
                5415 opera1=80000001, opera2=ffffffff9abec9ac, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                5455 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
```

```
                5475 opera1=00524524, opera2=0000000065653232, start=1,
muordi=1, reset=0, valid=0, result=0000000000000000
                5495 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                5510 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0000000065653232
                5550 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75b8caca6464
                5570 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75b99594c8c8
                5590 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75bb2b299190
                5610 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75be56532320
                5630 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75c4aca64640
                5650 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75d1594c8c80
                5670 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b75eab2991900
                5690 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b761d65323200
                5710 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b7682ca646400
                5730 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b774d94c8c800
                5750 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b78e329919000
                5770 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b7c0e53232000
                5790 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b8264a6464000
                5810 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5b8f114c8c8000
                5830 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5ba86a99190000
                5850 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5bdb1d32320000
                5870 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5c408264640000
                5890 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5d0b4cc8c80000
                5910 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff5ea0e191900000
                5930 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff61cc0b23200000
                5950 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff68225e46400000
                5970 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff74cf048c800000
                5990 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ff8e285119000000
                6010 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ffc0daea32000000
```

```
                6030 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0026401c64000001
                6050 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ffa7f5f0c8000002
                6070 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=fff4762990000004
                6090 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=008d769b20000009
                6110 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=007662ee40000013
                6130 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=00483b9480000027
                6150 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ffebece10000004e
                6170 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=007c640a0000009d
                6190 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=002a1ee60000013b
                6230 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=0000000000000000
                6250 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=002a1ee60000013b
$finish called from file "testbench_division.v", line 49.
$finish at simulation time               749500
           V C S   S i m u l a t i o n   R e p o r t
Time: 7495000 ps
CPU Time:      0.210 seconds;       Data structure size:   0.0Mb
Mon Dec 12 05:19:04 2016
```

C.2 Contents of Selected Reports from Netlist (Post-synthesis) Simulations (VCS or NCVERILOG)

MULTIPLICATION

```
Chronologic VCS simulator copyright 1991-2014
Contains Synopsys proprietary information.
Compiler version I-2014.03-2; Runtime version I-2014.03-2;  Dec
12 05:43 2016
                   0 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=x, muordi=0, reset=x, valid=0, result=xxxxxxxxxxxxxxxx
                  20 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=1, muordi=0, reset=x, valid=0, result=xxxxxxxxxxxxxxxx
                  30 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=1, muordi=0, reset=x, valid=0, result=00000000xxxxxxxx
                  40 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=0, muordi=0, reset=x, valid=0, result=00000000xxxxxxxx
                  50 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=0, muordi=0, reset=x, valid=0, result=00000000Xxxxxxxx
                  55 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx,
start=0, muordi=0, reset=1, valid=0, result=00000000Xxxxxxxx
```

```
                  70 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, 
start=0, muordi=0, reset=0, valid=0, result=00000000Xxxxxxxx
                  90 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, 
start=0, muordi=0, reset=0, valid=0, result=00000000Xxxxxxxx
                 110 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=000000000xxxxxxx
                 130 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=000000000Xxxxxxx
                 150 opera1=ffadbadc, opera2=0000000065326532, 
start=1, muordi=0, reset=0, valid=0, result=0000000065326532
                 190 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=0000000032993299
                 210 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=00292292194c994c
                 230 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=001491490ca64ca6
                 250 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=000a48a486532653
                 270 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=002e46e443299329
                 290 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=004046042194c994
                 310 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=0020230210ca64ca
                 330 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=0010118108653265
                 350 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=00312b5284329932
                 370 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=001895a942194c99
                 390 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=00356d66a10ca64c
                 410 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=001ab6b350865326
                 430 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=000d5b59a8432993
                 450 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=002fd03ed42194c9
                 470 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=00410ab16a10ca64
                 490 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=00208558b5086532
                 510 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=001042ac5a843299
                 530 opera1=ffadbadc, opera2=0000000065326532, 
start=0, muordi=0, reset=0, valid=0, result=003143e82d42194c
```

```
                 550 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=0018a1f416a10ca6
                 570 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=000c50fa0b508653
                 590 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=002f4b0f05a84329
                 610 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=0040c81982d42194
                 630 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=0020640cc16a10ca
                 650 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=0010320660b50865
                 670 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=00313b95305a8432
                 690 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=00189dca982d4219
                 710 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=003571774c16a10c
                 730 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=001ab8bba60b5086
                 750 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=000d5c5dd305a843
                 770 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=002fd0c0e982d421
                 790 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=00410af274c16a10
                 810 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=0, result=002085793a60b508
                 830 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=1, result=002085793a60b508
                 850 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=0, valid=1, result=ffdf7a86c59f4af8
                2190 opera1=ffadbadc, opera2=0000000065326532,
start=0, muordi=0, reset=1, valid=0, result=0000000000000000
                2205 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0000000000000000
                2225 opera1=52146325, opera2=ffffffff8bed8bee,
start=1, muordi=0, reset=0, valid=0, result=0000000000000000
                2230 opera1=52146325, opera2=ffffffff8bed8bee,
start=1, muordi=0, reset=0, valid=0, result=0000000074127412
                2245 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0000000074127412
                2250 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=000000003a093a09
                2270 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=290a31929d049d04
```

```
                2290 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=148518c94e824e82
                2310 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0a428c64a7412741
                2330 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=2e2b77c4d3a093a0
                2350 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=1715bbe269d049d0
                2370 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0b8addf134e824e8
                2390 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=05c56ef89a741274
                2410 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=02e2b77c4d3a093a
                2430 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=01715bbe269d049d
                2450 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=29c2df71934e824e
                2470 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=14e16fb8c9a74127
                2490 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=337ae96ee4d3a093
                2510 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=42c7a649f269d049
                2530 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=4a6e04b77934e824
                2550 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=2537025bbc9a7412
                2570 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=129b812dde4d3a09
                2590 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=3257f2296f269d04
                2610 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=192bf914b7934e82
                2630 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0c95fc8a5bc9a741
                2650 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=2f552fd7ade4d3a0
                2670 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=17aa97ebd6f269d0
                2690 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=0bd54bf5eb7934e8
                2710 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=05eaa5faf5bc9a74
                2730 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=02f552fd7ade4d3a
```

```
                2750 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=017aa97ebd6f269d
                2770 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=29c78651deb7934e
                2790 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=14e3c328ef5bc9a7
                2810 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=337c1326f7ade4d3
                2830 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=42c83b25fbd6f269
                2850 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=4a6e4f257deb7934
                2870 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=0, result=25372792bef5bc9a
                2890 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=1, result=25372792bef5bc9a
                2910 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=0, valid=1, result=dac8d86d410a4366
                3745 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=1, valid=1, result=dac8d86d410a4366
                3750 opera1=52146325, opera2=ffffffff8bed8bee,
start=0, muordi=0, reset=1, valid=0, result=0000000000000000
                3760 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=0000000000000000
                3780 opera1=80000001, opera2=fffffff9abec9ac,
start=1, muordi=0, reset=0, valid=0, result=0000000000000000
                3790 opera1=80000001, opera2=fffffff9abec9ac,
start=1, muordi=0, reset=0, valid=0, result=0000000065413654
                3800 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=0000000065413654
                3810 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=0000000032a09b2a
                3830 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=0000000019504d95
                3850 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=3ffffff8ca826ca
                3870 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=1ffffffc6541365
                3890 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=4ffffff632a09b2
                3910 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=27ffffffb19504d9
                3930 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=53ffffff58ca826c
                3950 opera1=80000001, opera2=fffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=29ffffffac654136
```

```
                3970 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=14ffffffd632a09b
                3990 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=4a7fffff6b19504d
                4010 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=653fffff358ca826
                4030 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=329fffff9ac65413
                4050 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=594fffff4d632a09
                4070 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=6ca7ffff26b19504
                4090 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=3653ffff9358ca82
                4110 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=1b29ffffc9ac6541
                4130 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=4d94ffff64d632a0
                4150 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=26ca7fffb26b1950
                4170 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=13653fffd9358ca8
                4190 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=09b29fffec9ac654
                4210 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=04d94ffff64d632a
                4230 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=026ca7fffb26b195
                4250 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=413653ff7d9358ca
                4270 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=209b29ffbec9ac65
                4290 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=504d94ff5f64d632
                4310 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=2826ca7fafb26b19
                4330 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=5413653f57d9358c
                4350 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=2a09b29fabec9ac6
                4370 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=1504d94fd5f64d63
                4390 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=4a826ca76afb26b1
                4410 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=65413653357d9358
```

```
                4430 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=0, result=32a09b299abec9ac
                4450 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=0, valid=1, result=32a09b299abec9ac
                5300 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=1, valid=1, result=32a09b299abec9ac
                5310 opera1=80000001, opera2=ffffffff9abec9ac,
start=0, muordi=0, reset=1, valid=0, result=0000000000000000
                5315 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0000000000000000
                5335 opera1=12123456, opera2=0000000021542369,
start=1, muordi=0, reset=0, valid=0, result=0000000000000000
                5350 opera1=12123456, opera2=0000000021542369,
start=1, muordi=0, reset=0, valid=0, result=0000000021542369
                5355 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0000000021542369
                5370 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=09091a2b10aa11b4
                5390 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=04848d15885508da
                5410 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0242468ac42a846d
                5430 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0a2a3d7062154236
                5450 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=05151eb8310aa11b
                5470 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0b93a9871885508d
                5490 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0ed2eeee8c42a846
                5510 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0769777746215423
                5530 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0cbdd5e6a310aa11
                5550 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0f68051e51885508
                5570 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=07b4028f28c42a84
                5590 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=03da014794621542
                5610 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=01ed00a3ca310aa1
                5630 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=09ff9a7ce5188550
                5650 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=04ffcd3e728c42a8
```

```
                5670 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=027fe69f39462154
                5690 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=013ff34f9ca310aa
                5710 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=009ff9a7ce518855
                5730 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=095916fee728c42a
                5750 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=04ac8b7f73946215
                5770 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0b5f5feab9ca310a
                5790 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=05afaff55ce51885
                5810 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0be0f225ae728c42
                5830 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=05f07912d7394621
                5850 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0c0156b46b9ca310
                5870 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=0600ab5a35ce5188
                5890 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=030055ad1ae728c4
                5910 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=01802ad68d739462
                5930 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=00c0156b46b9ca31
                5950 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=096924e0a35ce518
                5970 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=04b4927051ae728c
                5990 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=0, result=025a493828d73946
                6010 opera1=12123456, opera2=0000000021542369,
start=0, muordi=0, reset=0, valid=1, result=025a493828d73946
$finish called from file "testbench_multiplier.v", line 59.
$finish at simulation time              735500
           V C S   S i m u l a t i o n   R e p o r t
Time: 7355000 ps
CPU Time:      0.180 seconds;       Data structure size:   0.0Mb
Mon Dec 12 05:43:25 2016
```

DIVISION

```
ncverilog: 14.10-p001: (c) Copyright 1995-2014 Cadence Design Systems, Inc.
Recompiling... reason: file './testbench_division.v' is newer than expected.
      expected: Mon Dec 12 02:40:10 2016
      actual:   Mon Dec 12 03:11:31 2016
file: testbench_division.v
      module worklib.testbench_division:v
            errors: 0, warnings: 0
            Caching library 'tc240c' ....... Done
            Caching library 'worklib' ....... Done
      Elaborating the design hierarchy:
  Add_rca_1_0 D1 ( .sum(res_sub), .a(op1_sub_store_not), .b(result[63:32]),
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10347|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,1182): c_out

  Add_rca_1_7 D2
( .sum(res_add), .a(op1_store_not), .b(result[63:32]), .c_in(
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10349|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,2377): c_out

  Add_rca_1_6 C1 ( .sum(op1_sto     re_not), .a({1'b0, n819, n821, n822,
n823, n824,
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10351|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,3575): c_out

  Add_rca_1_4 C3 ( .sum(op2_store_not[63:32]), .a({1'b0,
op2_not_wire[62:32]}),
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10363|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,6222): c_out

  Add_rca_1_2 C5 ( .sum(res_store_not[63:32]), .a(res_not_wire[63:32]), .b({
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10373|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,8854): c_out

  Add_rca_1_1 C6 ( .sum(op1_sub_store_not), .a({1'b1, n419, n421, n422, n423,
               |
ncelab: *W,CUVWSP (./d5_netlist.v,10377|15): 1 output port was not connected:
ncelab: (./d5_netlist.v,10055): c_out

  CFD1XL \nst_reg[2]  ( .D(n608), .CP(clock), .QN(n851) );
                    |
ncelab: *W,CUVWSP (./d5_netlist.v,10388|20): 1 output port was not connected:
ncelab: (/apps/toshiba/sjsu/verilog/tc240c/CFD1XL.tsbvlibp,7): Q

  CFD1XL \nst_reg[1]  ( .D(n574), .CP(clock), .QN(n852) );
                    |
ncelab: *W,CUVWSP (./d5_netlist.v,10389|20): 1 output port was not connected:
ncelab: (/apps/toshiba/sjsu/verilog/tc240c/CFD1XL.tsbvlibp,7): Q

      Building instance overlay tables: .................... Done
      Generating native compiled code:
            worklib.testbench_division:v <0x7b5d4a61>
                  streams:  10, words: 12025
      Building instance specific data structures.
      Loading native compiled code:     .................... Done
      Design hierarchy summary:
                               Instances  Unique
            Modules:                3785     924
            UDPs:                    170       4
            Primitives:             8941      11
            Timing outputs:         2798      35
            Registers:               175      15
            Scalar wires:           2991       -
            Expanded wires:          288      14
            Vectored wires:            6       -
            Initial blocks:            4       4
            Pseudo assignments:       18      18
            Timing checks:          1029     351
            Simulation timescale:   10ps
      Writing initial simulation snapshot: worklib.testbench_division:v
Loading snapshot worklib.testbench_division:v .................... Done
ncsim> source /apps/cadence/INCISIV141/tools/inca/files/ncsimrc
ncsim> run
                     0 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=x,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                    20 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=1,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                    40 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=x, valid=x, result=xxxxxxxxxxxxxxxx
                    55 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=1, valid=x, result=xxxxxxxxxxxxxxxx
                    70 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=0, valid=x, result=xxxxxxxxxxxxxxxx
                    70 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=0, valid=x, result=Xxxxxxxxxxxxxxxx
                    71 opera1=xxxxxxxx, opera2=xxxxxxxxxxxxxxxx, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                   110 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                   150 opera1=babababb, opera2=1212121212121212, start=1,
muordi=1, reset=0, valid=0, result=0000000000000000
                   190 opera1=babababb, opera2=1212121212121212, start=0,
muordi=1, reset=0, valid=0, result=0000000000000000
                   191 opera1=babababb, opera2=1212121212121212, start=0,

****Content Deleted ****
                  6071 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=fff4762990000004
                  6090 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=7ff4762990000004
                  6091 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0084760900000000
                  6091 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=008d769b20000009
                  6111 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0004628a00000001
```

```
                 6111 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=007662ee40000013
                 6131 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=0040228400000003
                 6131 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=00483b9480000027
                 6150 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=80483b9480000027
                 6151 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=8048288000000006
                 6151 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=ffebece10000004e
                 6170 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=7febece10000004e
                 6171 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=006864000000000c
                 6171 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=0, result=007c640a0000009d
                 6191 opera1=00524524, opera2=0000000065653232, start=0,
jmuordi=1, reset=0, valid=0, result=0028040200000019
                 6191 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=002a1ee60000013b
                 6231 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=0000000000000000
                 6251 opera1=00524524, opera2=0000000065653232, start=0,
muordi=1, reset=0, valid=1, result=002a1ee60000013b
Simulation complete via $finish(1) at time 7495 NS + 0
./testbench_division.v:49  #2000 $finish;
ncsim> exit
```

C.3 Contents of Selected Reports from Synthesis (Design Compiler)

```
Initializing...
set link_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25
set target_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25}
/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25
set symbol_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.workview.sdb}
/apps/toshiba/sjsu/synopsys/tc240c/tc240c.workview.sdb
set synthetic_library {dw_foundation.sldb standard.sldb}
dw_foundation.sldb standard.sldb
set_min_library /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25 -
min_version /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_BCCOM25
Loading db file '/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25'
Loading db file '/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_BCCOM25'
Warning: Function '=' leaked 1 allocations for 16 bytes. (EQN-21)
1
read_verilog {division.v}
Loading db file '/apps/synopsys/SYNTH/libraries/syn/gtech.db'
Loading db file '/apps/synopsys/SYNTH/libraries/syn/standard.sldb'
  Loading link library 'tc240c'
  Loading link library 'gtech'
Loading verilog file
'/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'
Detecting input file type automatically (-rtl or -netlist).
Running DC verilog reader
Reading with Presto HDL Compiler (equivalent to -rtl option).

tc240c (library)/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25

1
create_clock clock -name clock -period 25.00
1
set_propagated_clock clock
Information: set_input_delay values are added to the propagated clock skew.
(TIM-113)
1
set_clock_uncertainty .25 clock
1
set_max_area 3000
1
set_fix_hold clock
1
compile -map_effort high
Warning: The following synthetic libraries should be added to
      the list of link libraries:
      'dw_foundation.sldb'. (UISN-26)
Information: Checking out the license 'DesignWare'. (SEC-104)
Information: Evaluating DesignWare library utilization. (UISN-27)
```

```
Statistics for case statements in always block at line 71 in file
      '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'
================================================
|           Line           |  full/ parallel  |
================================================
|            72            |    auto/auto     |
================================================

Inferred memory devices in process
      in routine division line 52 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
============================================================================
=====
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS |
ST |
============================================================================
=====
|       cst_reg       | Flip-flop |   3   |  Y  | N  | Y  | N  | N  | N  |
N  |
============================================================================
=====

Inferred memory devices in process
      in routine division line 71 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
============================================================================
=====
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS |
ST |
============================================================================
=====
|     res_not_reg     | Flip-flop |  64   |  Y  | N  | N  | N  | N  | N  |
N  |
|        i_reg        | Flip-flop |  32   |  N  | N  | N  | N  | N  | N  |
N  |
|       nst_reg       | Flip-flop |   3   |  N  | N  | N  | N  | N  | N  |
N  |
|     result_reg      | Flip-flop |  64   |  N  | N  | N  | N  | N  | N  |
N  |
|    res_store_reg    | Flip-flop |  32   |  N  | N  | N  | N  | N  | N  |
N  |
|    ope_flag_reg     | Flip-flop |   1   |  N  | N  | N  | N  | N  | N  |
N  |
|      sign_reg       | Flip-flop |   1   |  N  | N  | N  | N  | N  | N  |
N  |
|      valid_reg      | Flip-flop |   1   |  N  | N  | N  | N  | N  | N  |
N  |
============================================================================
=====

Inferred tri-state devices in process
      in routine division line 321 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
```

```
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A1       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 323 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A3       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 325 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A5       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 329 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A7       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 322 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A2       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 324 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A4       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 330 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
```

```
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A8       | Tri-State Buffer |   1   | N  |
=================================================

Inferred tri-state devices in process
      in routine division line 326 in file
            '/home/011469653@SJSUAD.SJSU.EDU/division/final/division.v'.
=================================================
| Register Name |       Type       | Width | MB |
=================================================
|      A6       | Tri-State Buffer |   1   | N  |
=================================================
```

REPORT – MULTIPLICATION

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : multiplier
Version: C-2009.06-SP5
Date   : Sat Dec 10 20:03:53 2016
****************************************

Operating Conditions: WCCOM25   Library: tc240c
Wire Load Model Mode: top

  Startpoint: res_not_reg[0]
              (rising edge-triggered flip-flop clocked by clock)
  Endpoint: result_reg[63]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -------------------------------------------------------------------
  clock clock (rise edge)                  0.00       0.00
  clock network delay (propagated)         0.00       0.00
  res_not_reg[0]/CP (CFD1QX4)              0.00       0.00 r
  res_not_reg[0]/Q (CFD1QX4)               0.42       0.42 f
  C3/a[0] (Add_rca_2)                      0.00       0.42 f
  C3/Z1/a[0] (Add_rca_16_4)                0.00       0.42 f
  C3/Z1/M1/a[0] (Add_rca_4_16)             0.00       0.42 f
  C3/Z1/M1/M1/a (Add_full_64)              0.00       0.42 f
  C3/Z1/M1/M1/M1/a (Add_half_128)          0.00       0.42 f
  C3/Z1/M1/M1/M1/U1/Z (CIVX3)              0.06       0.48 r
  C3/Z1/M1/M1/M1/U5/Z (CND2IX2)            0.08       0.56 f
  C3/Z1/M1/M1/M1/U6/Z (CND2X2)             0.07       0.63 r
  C3/Z1/M1/M1/M1/sum (Add_half_128)        0.00       0.63 r
  C3/Z1/M1/M1/M2/a (Add_half_127)          0.00       0.63 r
  C3/Z1/M1/M1/M2/U1/Z (CNR2IX2)            0.21       0.84 r
  C3/Z1/M1/M1/M2/c_out (Add_half_127)      0.00       0.84 r
  C3/Z1/M1/M1/U1/Z (COR2X1)                0.16       1.00 r
  C3/Z1/M1/M1/c_out (Add_full_64)          0.00       1.00 r
  C3/Z1/M1/M2/c_in (Add_full_63)           0.00       1.00 r
  C3/Z1/M1/M2/M2/b (Add_half_125)          0.00       1.00 r
  C3/Z1/M1/M2/M2/U2/Z (CAN2X1)             0.18       1.18 r
  C3/Z1/M1/M2/M2/c_out (Add_half_125)      0.00       1.18 r
  C3/Z1/M1/M2/U1/Z (CIVX1)                 0.08       1.27 f
  C3/Z1/M1/M2/U3/Z (CND2X2)                0.07       1.34 r
  C3/Z1/M1/M2/c_out (Add_full_63)          0.00       1.34 r
  C3/Z1/M1/M3/c_in (Add_full_62)           0.00       1.34 r
  C3/Z1/M1/M3/M2/b (Add_half_123)          0.00       1.34 r
  C3/Z1/M1/M3/M2/U4/Z (CIVX2)              0.07       1.41 f
  C3/Z1/M1/M3/M2/U5/Z (CNR2IX2)            0.10       1.51 r
  C3/Z1/M1/M3/M2/c_out (Add_half_123)      0.00       1.51 r
  C3/Z1/M1/M3/U1/Z (CIVX1)                 0.10       1.60 f
  C3/Z1/M1/M3/U2/Z (CND2X2)                 0.07       1.67 r
  C3/Z1/M1/M3/c_out (Add_full_62)           0.00       1.67 r
  C3/Z1/M1/M4/c_in (Add_full_61)            0.00       1.67 r
  C3/Z1/M1/M4/M2/b (Add_half_121)           0.00       1.67 r
  C3/Z1/M1/M4/M2/U3/Z (CIVX2)               0.07       1.74 f
  C3/Z1/M1/M4/M2/U4/Z (CNR2IX2)             0.10       1.84 r
  C3/Z1/M1/M4/M2/c_out (Add_half_121)       0.00       1.84 r
  C3/Z1/M1/M4/U1/Z (CIVX1)                  0.10       1.94 f
  C3/Z1/M1/M4/U3/Z (CND2X2)                 0.07       2.01 r
  C3/Z1/M1/M4/c_out (Add_full_61)           0.00       2.01 r
  C3/Z1/M1/c_out (Add_rca_4_16)             0.00       2.01 r
  C3/Z1/M2/c_in (Add_rca_4_15)              0.00       2.01 r
  C3/Z1/M2/M1/c_in (Add_full_60)            0.00       2.01 r
  C3/Z1/M2/M1/M2/b (Add_half_119)           0.00       2.01 r
  C3/Z1/M2/M1/M2/U4/Z (CIVX2)               0.07       2.08 f
  C3/Z1/M2/M1/M2/U5/Z (CNR2IX2)             0.10       2.18 r
  C3/Z1/M2/M1/M2/c_out (Add_half_119)       0.00       2.18 r
  C3/Z1/M2/M1/U1/Z (CIVX1)                  0.10       2.27 f
  C3/Z1/M2/M1/U2/Z (CND2X2)                 0.07       2.34 r
  C3/Z1/M2/M1/c_out (Add_full_60)           0.00       2.34 r
  C3/Z1/M2/M2/c_in (Add_full_59)            0.00       2.34 r
  C3/Z1/M2/M2/M2/b (Add_half_117)           0.00       2.34 r
  C3/Z1/M2/M2/M2/U3/Z (CIVX2)               0.07       2.41 f
  C3/Z1/M2/M2/M2/U4/Z (CNR2IX2)             0.10       2.51 r
  C3/Z1/M2/M2/M2/c_out (Add_half_117)       0.00       2.51 r
  C3/Z1/M2/M2/U1/Z (CIVX1)                  0.10       2.61 f
  C3/Z1/M2/M2/U3/Z (CND2X2)                 0.07       2.67 r
  C3/Z1/M2/M2/c_out (Add_full_59)           0.00       2.67 r
  C3/Z1/M2/M3/c_in (Add_full_58)            0.00       2.67 r
  C3/Z1/M2/M3/M2/b (Add_half_115)           0.00       2.67 r
  C3/Z1/M2/M3/M2/U3/Z (CIVX2)               0.07       2.75 f
  C3/Z1/M2/M3/M2/U4/Z (CNR2IX2)             0.10       2.85 r
  C3/Z1/M2/M3/M2/c_out (Add_half_115)       0.00       2.85 r
  C3/Z1/M2/M3/U1/Z (CIVX1)                  0.10       2.94 f
  C3/Z1/M2/M3/U3/Z (CND2X2)                 0.07       3.01 r
  C3/Z1/M2/M3/c_out (Add_full_58)           0.00       3.01 r
  C3/Z1/M2/M4/c_in (Add_full_57)            0.00       3.01 r
  C3/Z1/M2/M4/M2/b (Add_half_113)           0.00       3.01 r
  C3/Z1/M2/M4/M2/U4/Z (CIVX2)               0.07       3.08 f
  C3/Z1/M2/M4/M2/U5/Z (CNR2IX2)             0.10       3.18 r
  C3/Z1/M2/M4/M2/c_out (Add_half_113)       0.00       3.18 r
  C3/Z1/M2/M4/U1/Z (CIVX1)                  0.10       3.28 f
  C3/Z1/M2/M4/U3/Z (CND2X2)                 0.07       3.34 r
  C3/Z1/M2/M4/c_out (Add_full_57)           0.00       3.34 r
  C3/Z1/M2/c_out (Add_rca_4_15)             0.00       3.34 r
  C3/Z1/M3/c_in (Add_rca_4_14)              0.00       3.34 r
  C3/Z1/M3/M1/c_in (Add_full_56)            0.00       3.34 r
  C3/Z1/M3/M1/M2/b (Add_half_111)           0.00       3.34 r
  C3/Z1/M3/M1/M2/U3/Z (CIVX2)               0.07       3.42 f
  C3/Z1/M3/M1/M2/U4/Z (CNR2IX2)             0.10       3.51 r
  C3/Z1/M3/M1/M2/c_out (Add_half_111)       0.00       3.51 r
  C3/Z1/M3/M1/U2/Z (CIVX1)                  0.10       3.61 f
  C3/Z1/M3/M1/U1/Z (CND2X2)                 0.07       3.68 r
  C3/Z1/M3/M1/c_out (Add_full_56)           0.00       3.68 r
  C3/Z1/M3/M2/c_in (Add_full_55)            0.00       3.68 r
  C3/Z1/M3/M2/M2/b (Add_half_109)           0.00       3.68 r
  C3/Z1/M3/M2/M2/U3/Z (CIVX2)               0.07       3.75 f
  C3/Z1/M3/M2/M2/U4/Z (CNR2IX2)            0.10       3.85 r
  C3/Z1/M3/M2/M2/c_out (Add_half_109)      0.00       3.85 r
  C3/Z1/M3/M2/U1/Z (CIVX1)                 0.10       3.95 f
  C3/Z1/M3/M2/U3/Z (CND2X2)                0.07       4.01 r
  C3/Z1/M3/M2/c_out (Add_full_55)          0.00       4.01 r
  C3/Z1/M3/M3/c_in (Add_full_54)           0.00       4.01 r
  C3/Z1/M3/M3/M2/b (Add_half_107)          0.00       4.01 r
  C3/Z1/M3/M3/M2/U3/Z (CIVX2)              0.07       4.08 f
  C3/Z1/M3/M3/M2/U4/Z (CNR2IX2)            0.10       4.18 r
  C3/Z1/M3/M3/M2/c_out (Add_half_107)      0.00       4.18 r
  C3/Z1/M3/M3/U2/Z (CIVX1)                 0.10       4.28 f
  C3/Z1/M3/M3/U1/Z (CND2X2)                0.06       4.34 r
  C3/Z1/M3/M3/c_out (Add_full_54)          0.00       4.34 r
  C3/Z1/M3/M4/c_in (Add_full_53)           0.00       4.34 r
  C3/Z1/M3/M4/M2/b (Add_half_105)          0.00       4.34 r
  C3/Z1/M3/M4/M2/U3/Z (CAN2X1)             0.18       4.52 r
  C3/Z1/M3/M4/M2/c_out (Add_half_105)      0.00       4.52 r
  C3/Z1/M3/M4/U2/Z (CIVX1)                 0.08       4.61 f
  C3/Z1/M3/M4/U1/Z (CND2X2)                0.06       4.67 r
  C3/Z1/M3/M4/c_out (Add_full_53)          0.00       4.67 r
  C3/Z1/M3/c_out (Add_rca_4_14)            0.00       4.67 r
  C3/Z1/M4/c_in (Add_rca_4_13)             0.00       4.67 r
  C3/Z1/M4/M1/c_in (Add_full_52)           0.00       4.67 r
  C3/Z1/M4/M1/M2/b (Add_half_103)          0.00       4.67 r
  C3/Z1/M4/M1/M2/U3/Z (CAN2X1)             0.18       4.85 r
  C3/Z1/M4/M1/M2/c_out (Add_half_103)      0.00       4.85 r
  C3/Z1/M4/M1/U1/Z (CIVX1)                 0.08       4.94 f
  C3/Z1/M4/M1/U3/Z (CND2X2)                0.07       5.00 r
  C3/Z1/M4/M1/c_out (Add_full_52)          0.00       5.00 r
  C3/Z1/M4/M2/c_in (Add_full_51)           0.00       5.00 r
  C3/Z1/M4/M2/M2/b (Add_half_101)          0.00       5.00 r
  C3/Z1/M4/M2/M2/U4/Z (CIVX2)              0.07       5.08 f
  C3/Z1/M4/M2/M2/U5/Z (CNR2IX2)            0.10       5.18 r
  C3/Z1/M4/M2/M2/c_out (Add_half_101)      0.00       5.18 r
  C3/Z1/M4/M2/U2/Z (CIVX1)                 0.10       5.27 f
  C3/Z1/M4/M2/U1/Z (CND2X2)                0.06       5.33 r
  C3/Z1/M4/M2/c_out (Add_full_51)          0.00       5.33 r
  C3/Z1/M4/M3/c_in (Add_full_50)           0.00       5.33 r
  C3/Z1/M4/M3/M2/b (Add_half_99)           0.00       5.33 r
  C3/Z1/M4/M3/M2/U3/Z (CAN2X1)             0.18       5.51 r
  C3/Z1/M4/M3/M2/c_out (Add_half_99)       0.00       5.51 r
  C3/Z1/M4/M3/U1/Z (CIVX1)                 0.08       5.60 f
  C3/Z1/M4/M3/U3/Z (CND2X2)                0.07       5.67 r
  C3/Z1/M4/M3/c_out (Add_full_50)          0.00       5.67 r
  C3/Z1/M4/M4/c_in (Add_full_49)           0.00       5.67 r
  C3/Z1/M4/M4/M2/b (Add_half_97)           0.00       5.67 r
  C3/Z1/M4/M4/M2/U3/Z (CIVX2)              0.07       5.74 f
  C3/Z1/M4/M4/M2/U4/Z (CNR2IX2)            0.10       5.84 r
  C3/Z1/M4/M4/M2/c_out (Add_half_97)       0.00       5.84 r
  C3/Z1/M4/M4/U2/Z (CIVX1)                 0.10       5.93 f
  C3/Z1/M4/M4/U1/Z (CND2X2)                0.07       6.00 r
  C3/Z1/M4/M4/c_out (Add_full_49)          0.00       6.00 r
  C3/Z1/M4/c_out (Add_rca_4_13)            0.00       6.00 r
  C3/Z1/c_out (Add_rca_16_4)               0.00       6.00 r
  C3/Z2/c_in (Add_rca_16_3)                0.00       6.00 r
  C3/Z2/M1/c_in (Add_rca_4_12)             0.00       6.00 r
  C3/Z2/M1/M1/c_in (Add_full_48)           0.00       6.00 r
  C3/Z2/M1/M1/M2/b (Add_half_95)           0.00       6.00 r
  C3/Z2/M1/M1/M2/U3/Z (CIVX2)              0.07       6.07 f
  C3/Z2/M1/M1/M2/U4/Z (CNR2IX2)            0.10       6.17 r
  C3/Z2/M1/M1/M2/c_out (Add_half_95)       0.00       6.17 r
  C3/Z2/M1/M1/U1/Z (CIVX1)                 0.10       6.27 f
  C3/Z2/M1/M1/U3/Z (CND2X2)                0.07       6.34 r
  C3/Z2/M1/M1/c_out (Add_full_48)          0.00       6.34 r
  C3/Z2/M1/M2/c_in (Add_full_47)           0.00       6.34 r
  C3/Z2/M1/M2/M2/b (Add_half_93)           0.00       6.34 r
  C3/Z2/M1/M2/M2/U4/Z (CIVX2)              0.07       6.41 f
  C3/Z2/M1/M2/M2/U5/Z (CNR2IX2)            0.10       6.51 r
  C3/Z2/M1/M2/M2/c_out (Add_half_93)       0.00       6.51 r
  C3/Z2/M1/M2/U2/Z (CIVX1)                 0.10       6.60 f
  C3/Z2/M1/M2/U1/Z (CND2X2)                0.06       6.67 r
  C3/Z2/M1/M2/c_out (Add_full_47)          0.00       6.67 r
  C3/Z2/M1/M3/c_in (Add_full_46)           0.00       6.67 r
  C3/Z2/M1/M3/M2/b (Add_half_91)           0.00       6.67 r
  C3/Z2/M1/M3/M2/U3/Z (CAN2X1)             0.18       6.85 r
  C3/Z2/M1/M3/M2/c_out (Add_half_91)       0.00       6.85 r
  C3/Z2/M1/M3/U1/Z (CIVX1)                 0.08       6.93 f
  C3/Z2/M1/M3/U3/Z (CND2X2)                0.07       7.00 r
  C3/Z2/M1/M3/c_out (Add_full_46)          0.00       7.00 r
  C3/Z2/M1/M4/c_in (Add_full_45)           0.00       7.00 r
  C3/Z2/M1/M4/M2/b (Add_half_89)           0.00       7.00 r
  C3/Z2/M1/M4/M2/U3/Z (CIVX2)              0.07       7.07 f
  C3/Z2/M1/M4/M2/U4/Z (CNR2IX2)            0.10       7.17 r
  C3/Z2/M1/M4/M2/c_out (Add_half_89)       0.00       7.17 r
  C3/Z2/M1/M4/U2/Z (CIVX1)                 0.10       7.27 f
  C3/Z2/M1/M4/U1/Z (CND2X2)                0.06       7.33 r
  C3/Z2/M1/M4/c_out (Add_full_45)          0.00       7.33 r
  C3/Z2/M1/c_out (Add_rca_4_12)            0.00       7.33 r
  C3/Z2/M2/c_in (Add_rca_4_11)             0.00       7.33 r
  C3/Z2/M2/M1/c_in (Add_full_44)           0.00       7.33 r
  C3/Z2/M2/M1/M2/b (Add_half_87)           0.00       7.33 r
  C3/Z2/M2/M1/M2/U3/Z (CAN2X1)             0.18       7.51 r
  C3/Z2/M2/M1/M2/c_out (Add_half_87)       0.00       7.51 r
  C3/Z2/M2/M1/U1/Z (CIVX1)                 0.08       7.59 f
  C3/Z2/M2/M1/U3/Z (CND2X2)                0.07       7.66 r
  C3/Z2/M2/M1/c_out (Add_full_44)          0.00       7.66 r
  C3/Z2/M2/M2/c_in (Add_full_43)           0.00       7.66 r
  C3/Z2/M2/M2/M2/b (Add_half_85)           0.00       7.66 r
  C3/Z2/M2/M2/M2/U3/Z (CIVX2)              0.07       7.73 f
  C3/Z2/M2/M2/M2/U4/Z (CNR2IX2)            0.10       7.83 r
  C3/Z2/M2/M2/M2/c_out (Add_half_85)       0.00       7.83 r
  C3/Z2/M2/M2/U2/Z (CIVX1)                 0.10       7.93 f
  C3/Z2/M2/M2/U1/Z (CND2X2)                0.06       7.99 r
  C3/Z2/M2/M2/c_out (Add_full_43)          0.00       7.99 r
  C3/Z2/M2/M3/c_in (Add_full_42)           0.00       7.99 r
  C3/Z2/M2/M3/M2/b (Add_half_83)           0.00       7.99 r
  C3/Z2/M2/M3/M2/U3/Z (CAN2X1)             0.18       8.17 r
  C3/Z2/M2/M3/M2/c_out (Add_half_83)       0.00       8.17 r
  C3/Z2/M2/M3/U2/Z (CIVX1)                 0.08       8.26 f
  C3/Z2/M2/M3/U1/Z (CND2X2)                0.06       8.32 r
  C3/Z2/M2/M3/c_out (Add_full_42)          0.00       8.32 r
  C3/Z2/M2/M4/c_in (Add_full_41)           0.00       8.32 r
  C3/Z2/M2/M4/M2/b (Add_half_81)           0.00       8.32 r
  C3/Z2/M2/M4/M2/U3/Z (CAN2X1)             0.18       8.50 r
```

```
C3/Z2/M2/M4/M2/c_out (Add_half_81)       0.00        8.50 r
C3/Z2/M2/M4/U2/Z (CIVX1)                 0.08        8.59 f
C3/Z2/M2/M4/U1/Z (CND2X2)                0.06        8.65 r
C3/Z2/M2/M4/c_out (Add_full_41)          0.00        8.65 r
C3/Z2/M2/c_out (Add_rca_4_11)            0.00        8.65 r
C3/Z2/M3/c_in (Add_rca_4_10)             0.00        8.65 r
C3/Z2/M3/M1/c_in (Add_full_40)           0.00        8.65 r
C3/Z2/M3/M1/M2/b (Add_half_79)           0.00        8.65 r
C3/Z2/M3/M1/M2/U3/Z (CAN2X1)             0.18        8.83 r
C3/Z2/M3/M1/M2/c_out (Add_half_79)       0.00        8.83 r
C3/Z2/M3/M1/U1/Z (CIVX1)                 0.08        8.91 f
C3/Z2/M3/M1/U3/Z (CND2X2)                0.07        8.98 r
C3/Z2/M3/M1/c_out (Add_full_40)          0.00        8.98 r
C3/Z2/M3/M2/c_in (Add_full_39)           0.00        8.98 r
C3/Z2/M3/M2/M2/b (Add_half_77)           0.00        8.98 r
C3/Z2/M3/M2/M2/U4/Z (CIVX2)              0.07        9.05 f
C3/Z2/M3/M2/M2/U5/Z (CNR2IX2)            0.10        9.15 r
C3/Z2/M3/M2/M2/c_out (Add_half_77)       0.00        9.15 r
C3/Z2/M3/M2/U1/Z (CIVX1)                 0.10        9.25 f
C3/Z2/M3/M2/U3/Z (CND2X2)                0.07        9.32 r
C3/Z2/M3/M2/c_out (Add_full_39)          0.00        9.32 r
C3/Z2/M3/M3/c_in (Add_full_38)           0.00        9.32 r
C3/Z2/M3/M3/M2/b (Add_half_75)           0.00        9.32 r
C3/Z2/M3/M3/M2/U3/Z (CIVX2)              0.07        9.39 f
C3/Z2/M3/M3/M2/U4/Z (CNR2IX2)            0.10        9.49 r
C3/Z2/M3/M3/M2/c_out (Add_half_75)       0.00        9.49 r
C3/Z2/M3/M3/U1/Z (CIVX1)                 0.10        9.58 f
C3/Z2/M3/M3/U3/Z (CND2X2)                0.07        9.65 r
C3/Z2/M3/M3/c_out (Add_full_38)          0.00        9.65 r
C3/Z2/M3/M4/c_in (Add_full_37)           0.00        9.65 r
C3/Z2/M3/M4/M2/b (Add_half_73)           0.00        9.65 r
C3/Z2/M3/M4/M2/U4/Z (CIVX2)              0.07        9.72 f
C3/Z2/M3/M4/M2/U5/Z (CNR2IX2)            0.10        9.82 r
C3/Z2/M3/M4/M2/c_out (Add_half_73)       0.00        9.82 r
C3/Z2/M3/M4/U1/Z (CIVX1)                 0.10        9.92 f
C3/Z2/M3/M4/U3/Z (CND2X2)                0.07        9.99 r
C3/Z2/M3/M4/c_out (Add_full_37)          0.00        9.99 r
C3/Z2/M3/c_out (Add_rca_4_10)            0.00        9.99 r
C3/Z2/M4/c_in (Add_rca_4_9)              0.00        9.99 r
C3/Z2/M4/M1/c_in (Add_full_36)           0.00        9.99 r
C3/Z2/M4/M1/M2/b (Add_half_71)           0.00        9.99 r
C3/Z2/M4/M1/M2/U4/Z (CIVX2)              0.07       10.06 f
C3/Z2/M4/M1/M2/U5/Z (CNR2IX2)            0.10       10.16 r
C3/Z2/M4/M1/M2/c_out (Add_half_71)       0.00       10.16 r
C3/Z2/M4/M1/U1/Z (CIVX1)                 0.10       10.25 f
C3/Z2/M4/M1/U3/Z (CND2X2)                0.07       10.32 r
C3/Z2/M4/M1/c_out (Add_full_36)          0.00       10.32 r
C3/Z2/M4/M2/c_in (Add_full_35)           0.00       10.32 r
C3/Z2/M4/M2/M2/b (Add_half_69)           0.00       10.32 r
C3/Z2/M4/M2/M2/U3/Z (CIVX2)              0.07       10.39 f
C3/Z2/M4/M2/M2/U4/Z (CNR2IX2)            0.10       10.49 r
C3/Z2/M4/M2/M2/c_out (Add_half_69)       0.00       10.49 r
C3/Z2/M4/M2/U2/Z (CIVX1)                 0.10       10.59 f
C3/Z2/M4/M2/U1/Z (CND2X2)                0.06       10.65 r
C3/Z2/M4/M2/c_out (Add_full_35)          0.00       10.65 r
C3/Z2/M4/M3/c_in (Add_full_34)           0.00       10.65 r
C3/Z2/M4/M3/M2/b (Add_half_67)           0.00       10.65 r
```

```
  C3/Z2/M4/M3/M2/U3/Z (CAN2X1)              0.18       10.83 r
  C3/Z2/M4/M3/M2/c_out (Add_half_67)        0.00       10.83 r
  C3/Z2/M4/M3/U1/Z (CIVX1)                  0.08       10.92 f
  C3/Z2/M4/M3/U3/Z (CND2X2)                 0.07       10.98 r
  C3/Z2/M4/M3/c_out (Add_full_34)           0.00       10.98 r
  C3/Z2/M4/M4/c_in (Add_full_33)            0.00       10.98 r
  C3/Z2/M4/M4/M2/b (Add_half_65)            0.00       10.98 r
  C3/Z2/M4/M4/M2/U3/Z (CIVX2)               0.07       11.06 f
  C3/Z2/M4/M4/M2/U4/Z (CNR2IX2)             0.10       11.15 r
  C3/Z2/M4/M4/M2/c_out (Add_half_65)        0.00       11.15 r
  C3/Z2/M4/M4/U2/Z (CIVX1)                  0.10       11.25 f
  C3/Z2/M4/M4/U1/Z (CND2X2)                 0.07       11.32 r
  C3/Z2/M4/M4/c_out (Add_full_33)           0.00       11.32 r
  C3/Z2/M4/c_out (Add_rca_4_9)              0.00       11.32 r
  C3/Z2/c_out (Add_rca_16_3)                0.00       11.32 r
  C3/c_out (Add_rca_2)                      0.00       11.32 r
  C4/c_in (Add_rca_1)                       0.00       11.32 r
  C4/Z1/c_in (Add_rca_16_2)                 0.00       11.32 r
  C4/Z1/M1/c_in (Add_rca_4_8)               0.00       11.32 r
  C4/Z1/M1/M1/c_in (Add_full_32)            0.00       11.32 r
  C4/Z1/M1/M1/M2/b (Add_half_63)            0.00       11.32 r
  C4/Z1/M1/M1/M2/U4/Z (CND2IX2)             0.09       11.41 f
  C4/Z1/M1/M1/M2/U5/Z (CIVX2)               0.06       11.48 r
  C4/Z1/M1/M1/M2/c_out (Add_half_63)        0.00       11.48 r
  C4/Z1/M1/M1/U1/Z (CIVX2)                  0.06       11.54 f
  C4/Z1/M1/M1/U2/Z (CND2X2)                 0.06       11.60 r
  C4/Z1/M1/M1/c_out (Add_full_32)           0.00       11.60 r
  C4/Z1/M1/M2/c_in (Add_full_31)            0.00       11.60 r
  C4/Z1/M1/M2/M2/b (Add_half_61)            0.00       11.60 r
  C4/Z1/M1/M2/M2/U2/Z (CAN2X1)              0.18       11.78 r
  C4/Z1/M1/M2/M2/c_out (Add_half_61)        0.00       11.78 r
  C4/Z1/M1/M2/U2/Z (CIVX1)                  0.08       11.87 f
  C4/Z1/M1/M2/U1/Z (CND2X2)                 0.06       11.93 r
  C4/Z1/M1/M2/c_out (Add_full_31)           0.00       11.93 r
  C4/Z1/M1/M3/c_in (Add_full_30)            0.00       11.93 r
  C4/Z1/M1/M3/M2/b (Add_half_59)            0.00       11.93 r
  C4/Z1/M1/M3/M2/U3/Z (CAN2X1)              0.18       12.11 r
  C4/Z1/M1/M3/M2/c_out (Add_half_59)        0.00       12.11 r
  C4/Z1/M1/M3/U2/Z (CIVX1)                  0.08       12.19 f
  C4/Z1/M1/M3/U1/Z (CND2X2)                 0.06       12.26 r
  C4/Z1/M1/M3/c_out (Add_full_30)           0.00       12.26 r
  C4/Z1/M1/M4/c_in (Add_full_29)            0.00       12.26 r
  C4/Z1/M1/M4/M2/b (Add_half_57)            0.00       12.26 r
  C4/Z1/M1/M4/M2/U3/Z (CAN2X1)              0.18       12.44 r
  C4/Z1/M1/M4/M2/c_out (Add_half_57)        0.00       12.44 r
  C4/Z1/M1/M4/U2/Z (CIVX1)                  0.08       12.52 f
  C4/Z1/M1/M4/U1/Z (CND2X2)                 0.06       12.59 r
  C4/Z1/M1/M4/c_out (Add_full_29)           0.00       12.59 r
  C4/Z1/M1/c_out (Add_rca_4_8)              0.00       12.59 r
  C4/Z1/M2/c_in (Add_rca_4_7)               0.00       12.59 r
  C4/Z1/M2/M1/c_in (Add_full_28)            0.00       12.59 r
  C4/Z1/M2/M1/M2/b (Add_half_55)            0.00       12.59 r
  C4/Z1/M2/M1/M2/U3/Z (CAN2X1)              0.18       12.77 r
  C4/Z1/M2/M1/M2/c_out (Add_half_55)        0.00       12.77 r
  C4/Z1/M2/M1/U1/Z (CIVX1)                  0.08       12.85 f
  C4/Z1/M2/M1/U3/Z (CND2X2)                 0.08       12.93 r
  C4/Z1/M2/M1/c_out (Add_full_28)           0.00       12.93 r
  C4/Z1/M2/M2/c_in (Add_full_27)           0.00      12.93 r
  C4/Z1/M2/M2/M2/b (Add_half_53)           0.00      12.93 r
  C4/Z1/M2/M2/M2/U4/Z (CND2X2)             0.09      13.02 f
  C4/Z1/M2/M2/M2/U5/Z (CIVX2)              0.06      13.08 r
  C4/Z1/M2/M2/M2/c_out (Add_half_53)       0.00      13.08 r
  C4/Z1/M2/M2/U1/Z (CIVX2)                 0.06      13.14 f
  C4/Z1/M2/M2/U3/Z (CND2X2)                0.07      13.22 r
  C4/Z1/M2/M2/c_out (Add_full_27)          0.00      13.22 r
  C4/Z1/M2/M3/c_in (Add_full_26)           0.00      13.22 r
  C4/Z1/M2/M3/M2/b (Add_half_51)           0.00      13.22 r
  C4/Z1/M2/M3/M2/U4/Z (CND2X2)             0.09      13.31 f
  C4/Z1/M2/M3/M2/U5/Z (CIVX2)              0.06      13.37 r
  C4/Z1/M2/M3/M2/c_out (Add_half_51)       0.00      13.37 r
  C4/Z1/M2/M3/U1/Z (CIVX2)                 0.06      13.43 f
  C4/Z1/M2/M3/U3/Z (CND2X2)                0.07      13.51 r
  C4/Z1/M2/M3/c_out (Add_full_26)          0.00      13.51 r
  C4/Z1/M2/M4/c_in (Add_full_25)           0.00      13.51 r
  C4/Z1/M2/M4/M2/b (Add_half_49)           0.00      13.51 r
  C4/Z1/M2/M4/M2/U4/Z (CND2X2)             0.09      13.60 f
  C4/Z1/M2/M4/M2/U5/Z (CIVX2)              0.06      13.66 r
  C4/Z1/M2/M4/M2/c_out (Add_half_49)       0.00      13.66 r
  C4/Z1/M2/M4/U1/Z (CIVX2)                 0.06      13.72 f
  C4/Z1/M2/M4/U3/Z (CND2X2)                0.07      13.80 r
  C4/Z1/M2/M4/c_out (Add_full_25)          0.00      13.80 r
  C4/Z1/M2/c_out (Add_rca_4_7)             0.00      13.80 r
  C4/Z1/M3/c_in (Add_rca_4_6)              0.00      13.80 r
  C4/Z1/M3/M1/c_in (Add_full_24)           0.00      13.80 r
  C4/Z1/M3/M1/M2/b (Add_half_47)           0.00      13.80 r
  C4/Z1/M3/M1/M2/U4/Z (CND2X2)             0.09      13.89 f
  C4/Z1/M3/M1/M2/U5/Z (CIVX2)              0.06      13.95 r
  C4/Z1/M3/M1/M2/c_out (Add_half_47)       0.00      13.95 r
  C4/Z1/M3/M1/U1/Z (CIVX2)                 0.06      14.01 f
  C4/Z1/M3/M1/U3/Z (CND2X2)                0.06      14.07 r
  C4/Z1/M3/M1/c_out (Add_full_24)          0.00      14.07 r
  C4/Z1/M3/M2/c_in (Add_full_23)           0.00      14.07 r
  C4/Z1/M3/M2/M2/b (Add_half_45)           0.00      14.07 r
  C4/Z1/M3/M2/M2/U2/Z (CAN2X1)             0.18      14.25 r
  C4/Z1/M3/M2/M2/c_out (Add_half_45)       0.00      14.25 r
  C4/Z1/M3/M2/U2/Z (CIVX1)                 0.08      14.34 f
  C4/Z1/M3/M2/U1/Z (CND2X2)                0.06      14.40 r
  C4/Z1/M3/M2/c_out (Add_full_23)          0.00      14.40 r
  C4/Z1/M3/M3/c_in (Add_full_22)           0.00      14.40 r
  C4/Z1/M3/M3/M2/b (Add_half_43)           0.00      14.40 r
  C4/Z1/M3/M3/M2/U3/Z (CAN2X1)             0.18      14.58 r
  C4/Z1/M3/M3/M2/c_out (Add_half_43)       0.00      14.58 r
  C4/Z1/M3/M3/U1/Z (CIVX1)                 0.08      14.67 f
  C4/Z1/M3/M3/U3/Z (CND2X2)                0.08      14.74 r
  C4/Z1/M3/M3/c_out (Add_full_22)          0.00      14.74 r
  C4/Z1/M3/M4/c_in (Add_full_21)           0.00      14.74 r
  C4/Z1/M3/M4/M2/b (Add_half_41)           0.00      14.74 r
  C4/Z1/M3/M4/M2/U4/Z (CND2X2)             0.09      14.83 f
  C4/Z1/M3/M4/M2/U5/Z (CIVX2)              0.06      14.89 r
  C4/Z1/M3/M4/M2/c_out (Add_half_41)       0.00      14.89 r
  C4/Z1/M3/M4/U1/Z (CIVX2)                 0.06      14.96 f
  C4/Z1/M3/M4/U3/Z (CND2X2)                0.07      15.03 r
  C4/Z1/M3/M4/c_out (Add_full_21)          0.00      15.03 r
  C4/Z1/M3/c_out (Add_rca_4_6)             0.00      15.03 r
  C4/Z1/M4/c_in (Add_rca_4_5)              0.00      15.03 r
  C4/Z1/M4/M1/c_in (Add_full_20)           0.00      15.03 r
  C4/Z1/M4/M1/M2/b (Add_half_39)           0.00      15.03 r
  C4/Z1/M4/M1/M2/U4/Z (CND2X2)             0.09      15.12 f
  C4/Z1/M4/M1/M2/U5/Z (CIVX2)              0.06      15.18 r
  C4/Z1/M4/M1/M2/c_out (Add_half_39)       0.00      15.18 r
  C4/Z1/M4/M1/U1/Z (CIVX2)                 0.06      15.25 f
  C4/Z1/M4/M1/U3/Z (CND2X2)                0.07      15.32 r
  C4/Z1/M4/M1/c_out (Add_full_20)          0.00      15.32 r
  C4/Z1/M4/M2/c_in (Add_full_19)           0.00      15.32 r
  C4/Z1/M4/M2/M2/b (Add_half_37)           0.00      15.32 r
  C4/Z1/M4/M2/M2/U4/Z (CND2X2)             0.09      15.41 f
  C4/Z1/M4/M2/M2/U5/Z (CIVX2)              0.06      15.47 r
  C4/Z1/M4/M2/M2/c_out (Add_half_37)       0.00      15.47 r
  C4/Z1/M4/M2/U1/Z (CIVX2)                 0.06      15.54 f
  C4/Z1/M4/M2/U3/Z (CND2X2)                0.07      15.61 r
  C4/Z1/M4/M2/c_out (Add_full_19)          0.00      15.61 r
  C4/Z1/M4/M3/c_in (Add_full_18)           0.00      15.61 r
  C4/Z1/M4/M3/M2/b (Add_half_35)           0.00      15.61 r
  C4/Z1/M4/M3/M2/U4/Z (CND2X2)             0.09      15.70 f
  C4/Z1/M4/M3/M2/U5/Z (CIVX2)              0.06      15.76 r
  C4/Z1/M4/M3/M2/c_out (Add_half_35)       0.00      15.76 r
  C4/Z1/M4/M3/U1/Z (CIVX2)                 0.06      15.83 f
  C4/Z1/M4/M3/U3/Z (CND2X2)                0.06      15.89 r
  C4/Z1/M4/M3/c_out (Add_full_18)          0.00      15.89 r
  C4/Z1/M4/M4/c_in (Add_full_17)           0.00      15.89 r
  C4/Z1/M4/M4/M2/b (Add_half_33)           0.00      15.89 r
  C4/Z1/M4/M4/M2/U2/Z (CAN2X1)             0.18      16.07 r
  C4/Z1/M4/M4/M2/c_out (Add_half_33)       0.00      16.07 r
  C4/Z1/M4/M4/U2/Z (CIVX1)                 0.08      16.16 f
  C4/Z1/M4/M4/U1/Z (CND2X2)                0.06      16.22 r
  C4/Z1/M4/M4/c_out (Add_full_17)          0.00      16.22 r
  C4/Z1/M4/c_out (Add_rca_4_5)             0.00      16.22 r
  C4/Z1/c_out (Add_rca_16_2)               0.00      16.22 r
  C4/Z2/c_in (Add_rca_16_1)                0.00      16.22 r
  C4/Z2/M1/c_in (Add_rca_4_4)              0.00      16.22 r
  C4/Z2/M1/M1/c_in (Add_full_16)           0.00      16.22 r
  C4/Z2/M1/M1/M2/b (Add_half_31)           0.00      16.22 r
  C4/Z2/M1/M1/M2/U3/Z (CAN2X1)             0.18      16.40 r
  C4/Z2/M1/M1/M2/c_out (Add_half_31)       0.00      16.40 r
  C4/Z2/M1/M1/U2/Z (CIVX1)                 0.08      16.48 f
  C4/Z2/M1/M1/U1/Z (CND2X2)                0.06      16.55 r
  C4/Z2/M1/M1/c_out (Add_full_16)          0.00      16.55 r
  C4/Z2/M1/M2/c_in (Add_full_15)           0.00      16.55 r
  C4/Z2/M1/M2/M2/b (Add_half_29)           0.00      16.55 r
  C4/Z2/M1/M2/M2/U3/Z (CAN2X1)             0.18      16.73 r
  C4/Z2/M1/M2/M2/c_out (Add_half_29)       0.00      16.73 r
  C4/Z2/M1/M2/U1/Z (CIVX1)                 0.08      16.81 f
  C4/Z2/M1/M2/U3/Z (CND2X2)                0.08      16.89 r
  C4/Z2/M1/M2/c_out (Add_full_15)          0.00      16.89 r
  C4/Z2/M1/M3/c_in (Add_full_14)           0.00      16.89 r
  C4/Z2/M1/M3/M2/b (Add_half_27)           0.00      16.89 r
  C4/Z2/M1/M3/M2/U4/Z (CND2X2)             0.09      16.98 f
  C4/Z2/M1/M3/M2/U5/Z (CIVX2)              0.06      17.04 r
  C4/Z2/M1/M3/M2/c_out (Add_half_27)       0.00      17.04 r
  C4/Z2/M1/M3/U1/Z (CIVX2)                 0.06      17.10 f
  C4/Z2/M1/M3/U3/Z (CND2X2)                0.06      17.16 r
  C4/Z2/M1/M3/c_out (Add_full_14)          0.00      17.16 r
  C4/Z2/M1/M4/c_in (Add_full_13)           0.00      17.16 r
  C4/Z2/M1/M4/M2/b (Add_half_25)           0.00      17.16 r
  C4/Z2/M1/M4/M2/U2/Z (CAN2X1)             0.18      17.35 r
  C4/Z2/M1/M4/M2/c_out (Add_half_25)       0.00      17.35 r
  C4/Z2/M1/M4/U1/Z (CIVX1)                 0.08      17.43 f
  C4/Z2/M1/M4/U3/Z (CND2X2)                0.08      17.51 r
  C4/Z2/M1/M4/c_out (Add_full_13)          0.00      17.51 r
  C4/Z2/M1/c_out (Add_rca_4_4)             0.00      17.51 r
  C4/Z2/M2/c_in (Add_rca_4_3)              0.00      17.51 r
  C4/Z2/M2/M1/c_in (Add_full_12)           0.00      17.51 r
  C4/Z2/M2/M1/M2/b (Add_half_23)           0.00      17.51 r
  C4/Z2/M2/M1/M2/U4/Z (CND2X2)             0.09      17.60 f
  C4/Z2/M2/M1/M2/U5/Z (CIVX2)              0.06      17.66 r
  C4/Z2/M2/M1/M2/c_out (Add_half_23)       0.00      17.66 r
  C4/Z2/M2/M1/U1/Z (CIVX2)                 0.06      17.72 f
  C4/Z2/M2/M1/U3/Z (CND2X2)                0.06      17.79 r
  C4/Z2/M2/M1/c_out (Add_full_12)          0.00      17.79 r
  C4/Z2/M2/M2/c_in (Add_full_11)           0.00      17.79 r
  C4/Z2/M2/M2/M2/b (Add_half_21)           0.00      17.79 r
  C4/Z2/M2/M2/M2/U1/Z (CAN2X1)             0.18      17.97 r
  C4/Z2/M2/M2/M2/c_out (Add_half_21)       0.00      17.97 r
  C4/Z2/M2/M2/U2/Z (CIVX1)                 0.08      18.06 f
  C4/Z2/M2/M2/U1/Z (CND2X2)                0.06      18.12 r
  C4/Z2/M2/M2/c_out (Add_full_11)          0.00      18.12 r
  C4/Z2/M2/M3/c_in (Add_full_10)           0.00      18.12 r
  C4/Z2/M2/M3/M2/b (Add_half_19)           0.00      18.12 r
  C4/Z2/M2/M3/M2/U3/Z (CAN2X1)             0.18      18.30 r
  C4/Z2/M2/M3/M2/c_out (Add_half_19)       0.00      18.30 r
  C4/Z2/M2/M3/U2/Z (CIVX1)                 0.08      18.38 f
  C4/Z2/M2/M3/U1/Z (CND2X2)                0.06      18.45 r
  C4/Z2/M2/M3/c_out (Add_full_10)          0.00      18.45 r
  C4/Z2/M2/M4/c_in (Add_full_9)            0.00      18.45 r
  C4/Z2/M2/M4/M2/b (Add_half_17)           0.00      18.45 r
  C4/Z2/M2/M4/M2/U3/Z (CAN2X1)             0.18      18.63 r
  C4/Z2/M2/M4/M2/c_out (Add_half_17)       0.00      18.63 r
  C4/Z2/M2/M4/U1/Z (CIVX1)                 0.08      18.71 f
  C4/Z2/M2/M4/U3/Z (CND2X2)                0.08      18.79 r
  C4/Z2/M2/M4/c_out (Add_full_9)           0.00      18.79 r
  C4/Z2/M2/c_out (Add_rca_4_3)             0.00      18.79 r
  C4/Z2/M3/c_in (Add_rca_4_2)              0.00      18.79 r
  C4/Z2/M3/M1/c_in (Add_full_8)            0.00      18.79 r
  C4/Z2/M3/M1/M2/b (Add_half_15)           0.00      18.79 r
  C4/Z2/M3/M1/M2/U4/Z (CND2X2)             0.09      18.88 f
  C4/Z2/M3/M1/M2/U5/Z (CIVX2)              0.06      18.94 r
  C4/Z2/M3/M1/M2/c_out (Add_half_15)       0.00      18.94 r
  C4/Z2/M3/M1/U1/Z (CIVX2)                 0.06      19.00 f
  C4/Z2/M3/M1/U3/Z (CND2X2)                0.07      19.08 r
  C4/Z2/M3/M1/c_out (Add_full_8)           0.00      19.08 r
  C4/Z2/M3/M2/c_in (Add_full_7)            0.00      19.08 r
  C4/Z2/M3/M2/M2/b (Add_half_13)           0.00      19.08 r
  C4/Z2/M3/M2/M2/U4/Z (CND2X2)             0.09      19.17 f
  C4/Z2/M3/M2/M2/U5/Z (CIVX2)              0.06      19.23 r
  C4/Z2/M3/M2/M2/c_out (Add_half_13)       0.00      19.23 r
  C4/Z2/M3/M2/U2/Z (CIVX2)                 0.06      19.29 f
  C4/Z2/M3/M2/U1/Z (CND2X2)                0.06      19.35 r
  C4/Z2/M3/M2/c_out (Add_full_7)           0.00      19.35 r
  C4/Z2/M3/M3/c_in (Add_full_6)             0.00      19.35 r
  C4/Z2/M3/M3/M2/b (Add_half_11)            0.00      19.35 r
  C4/Z2/M3/M3/M2/U1/Z (CAN2X1)              0.18      19.54 r
  C4/Z2/M3/M3/M2/c_out (Add_half_11)        0.00      19.54 r
  C4/Z2/M3/M3/U2/Z (CIVX1)                  0.08      19.62 f
  C4/Z2/M3/M3/U1/Z (CND2X2)                 0.06      19.68 r
  C4/Z2/M3/M3/c_out (Add_full_6)            0.00      19.68 r
  C4/Z2/M3/M4/c_in (Add_full_5)             0.00      19.68 r
  C4/Z2/M3/M4/M2/b (Add_half_9)             0.00      19.68 r
  C4/Z2/M3/M4/M2/U3/Z (CAN2X1)              0.18      19.86 r
  C4/Z2/M3/M4/M2/c_out (Add_half_9)         0.00      19.86 r
  C4/Z2/M3/M4/U1/Z (CIVX1)                  0.08      19.95 f
  C4/Z2/M3/M4/U3/Z (CND2X2)                 0.07      20.02 r
  C4/Z2/M3/M4/c_out (Add_full_5)            0.00      20.02 r
  C4/Z2/M3/c_out (Add_rca_4_2)              0.00      20.02 r
  C4/Z2/M4/c_in (Add_rca_4_1)               0.00      20.02 r
  C4/Z2/M4/M1/c_in (Add_full_4)             0.00      20.02 r
  C4/Z2/M4/M1/M2/b (Add_half_7)             0.00      20.02 r
  C4/Z2/M4/M1/M2/U1/Z (CAN2X1)              0.18      20.20 r
  C4/Z2/M4/M1/M2/c_out (Add_half_7)         0.00      20.20 r
  C4/Z2/M4/M1/U1/Z (CIVX1)                  0.08      20.28 f
  C4/Z2/M4/M1/U3/Z (CND2X2)                 0.07      20.35 r
  C4/Z2/M4/M1/c_out (Add_full_4)            0.00      20.35 r
  C4/Z2/M4/M2/c_in (Add_full_3)             0.00      20.35 r
  C4/Z2/M4/M2/M2/b (Add_half_5)             0.00      20.35 r
  C4/Z2/M4/M2/M2/U1/Z (CAN2X1)              0.18      20.53 r
  C4/Z2/M4/M2/M2/c_out (Add_half_5)         0.00      20.53 r
  C4/Z2/M4/M2/U1/Z (CIVX1)                  0.08      20.62 f
  C4/Z2/M4/M2/U3/Z (CND2X2)                 0.07      20.69 r
  C4/Z2/M4/M2/c_out (Add_full_3)            0.00      20.69 r
  C4/Z2/M4/M3/c_in (Add_full_2)             0.00      20.69 r
  C4/Z2/M4/M3/M2/b (Add_half_3)             0.00      20.69 r
  C4/Z2/M4/M3/M2/U2/Z (CAN2X1)              0.18      20.86 r
  C4/Z2/M4/M3/M2/c_out (Add_half_3)         0.00      20.86 r
  C4/Z2/M4/M3/U1/Z (COR2X2)                 0.19      21.05 r
  C4/Z2/M4/M3/c_out (Add_full_2)            0.00      21.05 r
  C4/Z2/M4/M4/c_in (Add_full_1)             0.00      21.05 r
  C4/Z2/M4/M4/M2/b (Add_half_1)             0.00      21.05 r
  C4/Z2/M4/M4/M2/U1/Z (CENX2)               0.20      21.26 r
  C4/Z2/M4/M4/M2/sum (Add_half_1)           0.00      21.26 r
  C4/Z2/M4/M4/sum (Add_full_1)              0.00      21.26 r
  C4/Z2/M4/sum[3] (Add_rca_4_1)             0.00      21.26 r
  C4/Z2/sum[15] (Add_rca_16_1)              0.00      21.26 r
  C4/sum[31] (Add_rca_1)                    0.00      21.26 r
  U538/Z (CND2IX2)                          0.11      21.37 f
  U923/Z (CND2IX1)                          0.07      21.44 r
  result_reg[63]/D (CFD3QX2)                0.00      21.44 r
  data arrival time                                   21.44

  clock clock (rise edge)                  22.00      22.00
  clock network delay (propagated)          0.00      22.00
  clock uncertainty                        -0.25      21.75
  result_reg[63]/CP (CFD3QX2)               0.00      21.75 r
  library setup time                       -0.31      21.44
  data required time                                  21.44
  -----------------------------------------------------------
  data required time                                  21.44
  data arrival time                                 -21.44
  ------------------------------------------------------------
  slack (MET)                                         0.00



****************************************
Report : area
Design : multiplier
Version: C-2009.06-SP5
Date   : Sat Dec 10 20:03:53 2016
****************************************

Library(s) Used:

    tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25)

Number of ports:              165
Number of nets:               973
Number of cells:              708
Number of references:          53

Combinational area:       2833.500000
Noncombinational area:     624.500000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          3458.000000
Total area:                 undefined
1

****************************************
Report : power
        -analysis_effort low
Design : multiplier
Version: C-2009.06-SP5
Date   : Sat Dec 10 20:03:53 2016
****************************************


Library(s) Used:

    tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25)


Operating Conditions: WCCOM25   Library: tc240c
Wire Load Model Mode: top


Global Operating Voltage = 2.3
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = Unitless
  Cell Internal Power  =   1.6279 mW   (88%)
  Net Switching Power  = 218.6020 uW   (12%)
                         ---------
Total Dynamic Power    =   1.8465 mW  (100%)

Cell Leakage Power     =   0.0000

1
```

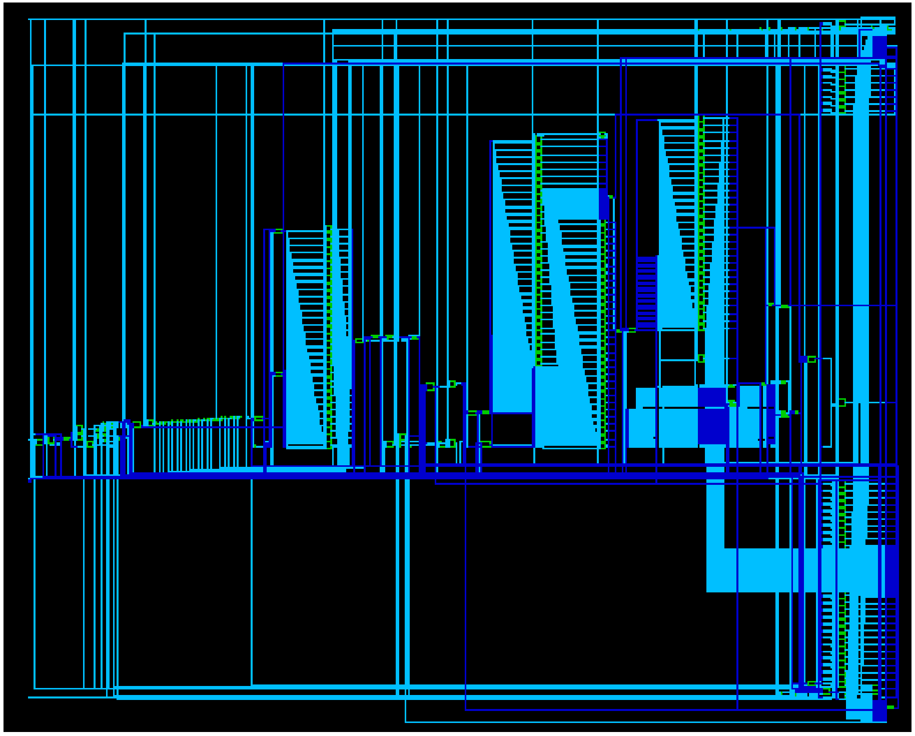
REPORT - DIVISION

```
  Beginning Pass 1 Mapping
  ------------------------
  Processing 'Add_half_1_0'
  Processing 'Add_full_1_0'
  Processing 'Add_rca_4_1_0'
  Processing 'Add_rca_16_1_0'
  Processing 'Add_rca_1_0'
  Processing 'division'

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : division
Version: C-2009.06-SP5
Date   : Sat Dec 10 17:41:18 2016
****************************************

Operating Conditions: WCCOM25   Library: tc240c
Wire Load Model Mode: top

  Startpoint: sign_reg (rising edge-triggered flip-flop clocked by clock)
  Endpoint: result_reg[63]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                                   Incr       Path
  --------------------------------------------------------------------------
  clock clock (rise edge)                                 0.00       0.00
  clock network delay (propagated)                        0.00       0.00
  sign_reg/CP (CFD1X1)                                    0.00       0.00 r
  sign_reg/Q (CFD1X1)                                     0.44       0.44 r
  A5/Z (CTSX2)                                            0.27       0.71 f
  C4/c_in (Add_rca_1_3)                                   0.00       0.71 f
  C4/Z1/c_in (Add_rca_16_1_6)                             0.00       0.71 f
  C4/Z1/M1/c_in (Add_rca_4_1_24)                          0.00       0.71 f
  C4/Z1/M1/M1/c_in (Add_full_1_96)                        0.00       0.71 f
  C4/Z1/M1/M1/M2/b (Add_half_1_191)                       0.00       0.71 f
  C4/Z1/M1/M1/M2/U3/Z (CIVX2)                             0.05       0.76 r
  C4/Z1/M1/M1/M2/U2/Z (CNR2IX1)                           0.06       0.83 f
  C4/Z1/M1/M1/M2/c_out (Add_half_1_191)                   0.00       0.83 f
  C4/Z1/M1/M1/U2/Z (CND2IX1)                              0.19       1.01 f
  C4/Z1/M1/M1/c_out (Add_full_1_96)                       0.00       1.01 f
  C4/Z1/M1/M2/c_in (Add_full_1_95)                        0.00       1.01 f
  C4/Z1/M1/M2/M2/b (Add_half_1_189)                       0.00       1.01 f
  C4/Z1/M1/M2/M2/U2/Z (CAN2X1)                            0.16       1.17 f
  C4/Z1/M1/M2/M2/c_out (Add_half_1_189)                   0.00       1.17 f
  C4/Z1/M1/M2/U2/Z (CND2IX1)                              0.18       1.36 f
  C4/Z1/M1/M2/c_out (Add_full_1_95)                       0.00       1.36 f
  C4/Z1/M1/M3/c_in (Add_full_1_94)                        0.00       1.36 f
  C4/Z1/M1/M3/M2/b (Add_half_1_187)                       0.00       1.36 f
  C4/Z1/M1/M3/M2/U2/Z (CAN2X1)                            0.16       1.52 f
  C4/Z1/M1/M3/M2/c_out (Add_half_1_187)                   0.00       1.52 f
  C4/Z1/M1/M3/U2/Z (CND2IX1)                              0.19       1.70 f
  C4/Z1/M1/M3/c_out (Add_full_1_94)                       0.00       1.70 f
  C4/Z1/M1/M4/c_in (Add_full_1_93)                        0.00       1.70 f
  C4/Z1/M1/M4/M2/b (Add_half_1_185)                       0.00       1.70 f
  C4/Z1/M1/M4/M2/U2/Z (CAN2X1)                            0.18       1.88 f
  C4/Z1/M1/M4/M2/c_out (Add_half_1_185)                   0.00       1.88 f
  C4/Z1/M1/M4/U2/Z (CND2IX1)                              0.18       2.06 f
  C4/Z1/M1/M4/c_out (Add_full_1_93)                       0.00       2.06 f
  C4/Z1/M1/c_out (Add_rca_4_1_24)                         0.00       2.06 f
  C4/Z1/M2/c_in (Add_rca_4_1_23)                          0.00       2.06 f
  C4/Z1/M2/M1/c_in (Add_full_1_92)                        0.00       2.06 f
  C4/Z1/M2/M1/M2/b (Add_half_1_183)                       0.00       2.06 f
  C4/Z1/M2/M1/M2/U1/Z (CAN2X1)                            0.16       2.22 f
  C4/Z1/M2/M1/M2/c_out (Add_half_1_183)                   0.00       2.22 f
  C4/Z1/M2/M1/U1/Z (COR2X1)                               0.24       2.46 f
  C4/Z1/M2/M1/c_out (Add_full_1_92)                       0.00       2.46 f
  C4/Z1/M2/M2/c_in (Add_full_1_91)                        0.00       2.46 f
  C4/Z1/M2/M2/M2/b (Add_half_1_181)                       0.00       2.46 f
  C4/Z1/M2/M2/M2/U2/Z (CAN2X1)                            0.16       2.63 f
  C4/Z1/M2/M2/M2/c_out (Add_half_1_181)                   0.00       2.63 f
  C4/Z1/M2/M2/U2/Z (CND2IX1)                              0.18       2.81 f
  C4/Z1/M2/M2/c_out (Add_full_1_91)                       0.00       2.81 f
  C4/Z1/M2/M3/c_in (Add_full_1_90)                        0.00       2.81 f
  C4/Z1/M2/M3/M2/b (Add_half_1_179)                       0.00       2.81 f
  C4/Z1/M2/M3/M2/U1/Z (CAN2X1)                            0.16       2.97 f
  C4/Z1/M2/M3/M2/c_out (Add_half_1_179)                   0.00       2.97 f
  C4/Z1/M2/M3/U2/Z (CND2IX1)                              0.18       3.15 f
  C4/Z1/M2/M3/c_out (Add_full_1_90)                       0.00       3.15 f
  C4/Z1/M2/M4/c_in (Add_full_1_89)                        0.00       3.15 f
  C4/Z1/M2/M4/M2/b (Add_half_1_177)                       0.00       3.15 f
  C4/Z1/M2/M4/M2/U2/Z (CAN2X1)                            0.16       3.31 f
  C4/Z1/M2/M4/M2/c_out (Add_half_1_177)                   0.00       3.31 f
  C4/Z1/M2/M4/U2/Z (CND2IX1)                              0.19       3.50 f
  C4/Z1/M2/M4/c_out (Add_full_1_89)                       0.00       3.50 f
  C4/Z1/M2/c_out (Add_rca_4_1_23)                         0.00       3.50 f
  C4/Z1/M3/c_in (Add_rca_4_1_22)                          0.00       3.50 f
  C4/Z1/M3/M1/c_in (Add_full_1_88)                        0.00       3.50 f
  C4/Z1/M3/M1/M2/b (Add_half_1_175)                       0.00       3.50 f
  C4/Z1/M3/M1/M2/U2/Z (CAN2X1)                            0.17       3.67 f
  C4/Z1/M3/M1/M2/c_out (Add_half_1_175)                   0.00       3.67 f
  C4/Z1/M3/M1/U1/Z (COR2X1)                               0.24       3.91 f
  C4/Z1/M3/M1/c_out (Add_full_1_88)                       0.00       3.91 f
  C4/Z1/M3/M2/c_in (Add_full_1_87)                        0.00       3.91 f
  C4/Z1/M3/M2/M2/b (Add_half_1_173)                       0.00       3.91 f
  C4/Z1/M3/M2/M2/U1/Z (CAN2X1)                            0.16       4.08 f
  C4/Z1/M3/M2/M2/c_out (Add_half_1_173)                   0.00       4.08 f
  C4/Z1/M3/M2/U2/Z (CND2IX1)                              0.19       4.26 f
  C4/Z1/M3/M2/c_out (Add_full_1_87)                       0.00       4.26 f
  C4/Z1/M3/M3/c_in (Add_full_1_86)                        0.00       4.26 f
  C4/Z1/M3/M3/M2/b (Add_half_1_171)                       0.00       4.26 f
  C4/Z1/M3/M3/M2/U1/Z (CAN2X1)                            0.18       4.44 f
  C4/Z1/M3/M3/M2/c_out (Add_half_1_171)                   0.00       4.44 f
  C4/Z1/M3/M3/U2/Z (CND2IX1)                              0.19       4.62 f
  C4/Z1/M3/M3/c_out (Add_full_1_86)                       0.00       4.62 f
  C4/Z1/M3/M4/c_in (Add_full_1_85)                        0.00       4.62 f
  C4/Z1/M3/M4/M2/b (Add_half_1_169)                       0.00       4.62 f
  C4/Z1/M3/M4/M2/U1/Z (CAN2X1)                            0.18       4.80 f
  C4/Z1/M3/M4/M2/c_out (Add_half_1_169)                   0.00       4.80 f
  C4/Z1/M3/M4/U2/Z (CND2IX1)                              0.18       4.98 f
  C4/Z1/M3/M4/c_out (Add_full_1_85)                       0.00       4.98 f
  C4/Z1/M3/c_out (Add_rca_4_1_22)                         0.00       4.98 f
  C4/Z1/M4/c_in (Add_rca_4_1_21)                          0.00       4.98 f
  C4/Z1/M4/M1/c_in (Add_full_1_84)                        0.00       4.98 f
  C4/Z1/M4/M1/M2/b (Add_half_1_167)                       0.00       4.98 f
  C4/Z1/M4/M1/M2/U1/Z (CAN2X1)                            0.16       5.14 f
  C4/Z1/M4/M1/M2/c_out (Add_half_1_167)                   0.00       5.14 f
  C4/Z1/M4/M1/U1/Z (COR2X1)                               0.24       5.38 f
  C4/Z1/M4/M1/c_out (Add_full_1_84)                       0.00       5.38 f
  C4/Z1/M4/M2/c_in (Add_full_1_83)                        0.00       5.38 f
  C4/Z1/M4/M2/M2/b (Add_half_1_165)                       0.00       5.38 f
  C4/Z1/M4/M2/M2/U2/Z (CAN2X1)                            0.17       5.56 f
  C4/Z1/M4/M2/M2/c_out (Add_half_1_165)                   0.00       5.56 f
  C4/Z1/M4/M2/U1/Z (COR2X1)                               0.24       5.80 f
  C4/Z1/M4/M2/c_out (Add_full_1_83)                       0.00       5.80 f
  C4/Z1/M4/M3/c_in (Add_full_1_82)                        0.00       5.80 f
  C4/Z1/M4/M3/M2/b (Add_half_1_163)                       0.00       5.80 f
  C4/Z1/M4/M3/M2/U1/Z (CAN2X1)                            0.16       5.96 f
  C4/Z1/M4/M3/M2/c_out (Add_half_1_163)                   0.00       5.96 f
  C4/Z1/M4/M3/U1/Z (COR2X1)                               0.24       6.19 f
  C4/Z1/M4/M3/c_out (Add_full_1_82)                       0.00       6.19 f
  C4/Z1/M4/M4/c_in (Add_full_1_81)                        0.00       6.19 f
  C4/Z1/M4/M4/M2/b (Add_half_1_161)                       0.00       6.19 f
  C4/Z1/M4/M4/M2/U3/Z (CAN2X1)                            0.16       6.35 f
  C4/Z1/M4/M4/M2/c_out (Add_half_1_161)                   0.00       6.35 f
  C4/Z1/M4/M4/U1/Z (COR2X1)                               0.24       6.60 f
  C4/Z1/M4/M4/c_out (Add_full_1_81)                       0.00       6.60 f
  C4/Z1/M4/c_out (Add_rca_4_1_21)                         0.00       6.60 f
  C4/Z1/c_out (Add_rca_16_1_6)                            0.00       6.60 f
  C4/Z2/c_in (Add_rca_16_1_5)                             0.00       6.60 f
  C4/Z2/M1/c_in (Add_rca_4_1_20)                          0.00       6.60 f
  C4/Z2/M1/M1/c_in (Add_full_1_80)                        0.00       6.60 f
  C4/Z2/M1/M1/M2/b (Add_half_1_159)                       0.00       6.60 f
  C4/Z2/M1/M1/M2/U1/Z (CAN2X1)                            0.18       6.77 f
  C4/Z2/M1/M1/M2/c_out (Add_half_1_159)                   0.00       6.77 f
  C4/Z2/M1/M1/U2/Z (CND2IX1)                              0.18       6.96 f
  C4/Z2/M1/M1/c_out (Add_full_1_80)                       0.00       6.96 f
  C4/Z2/M1/M2/c_in (Add_full_1_79)                        0.00       6.96 f
  C4/Z2/M1/M2/M2/b (Add_half_1_157)                       0.00       6.96 f
  C4/Z2/M1/M2/M2/U1/Z (CAN2X1)                            0.16       7.12 f
  C4/Z2/M1/M2/M2/c_out (Add_half_1_157)                   0.00       7.12 f
  C4/Z2/M1/M2/U2/Z (CND2IX1)                              0.19       7.30 f
  C4/Z2/M1/M2/c_out (Add_full_1_79)                       0.00       7.30 f
  C4/Z2/M1/M3/c_in (Add_full_1_78)                        0.00       7.30 f
  C4/Z2/M1/M3/M2/b (Add_half_1_155)                       0.00       7.30 f
  C4/Z2/M1/M3/M2/U1/Z (CAN2X1)                            0.18       7.48 f
  C4/Z2/M1/M3/M2/c_out (Add_half_1_155)                   0.00       7.48 f
  C4/Z2/M1/M3/U2/Z (CND2IX1)                              0.18       7.66 f
  C4/Z2/M1/M3/c_out (Add_full_1_78)                       0.00       7.66 f
  C4/Z2/M1/M4/c_in (Add_full_1_77)                        0.00       7.66 f
  C4/Z2/M1/M4/M2/b (Add_half_1_153)                       0.00       7.66 f
  C4/Z2/M1/M4/M2/U1/Z (CAN2X1)                            0.16       7.82 f
  C4/Z2/M1/M4/M2/c_out (Add_half_1_153)                   0.00       7.82 f
  C4/Z2/M1/M4/U2/Z (CND2IX1)                              0.19       8.01 f
  C4/Z2/M1/M4/c_out (Add_full_1_77)                       0.00       8.01 f
  C4/Z2/M1/c_out (Add_rca_4_1_20)                         0.00       8.01 f
  C4/Z2/M2/c_in (Add_rca_4_1_19)                          0.00       8.01 f
  C4/Z2/M2/M1/c_in (Add_full_1_76)                        0.00       8.01 f
  C4/Z2/M2/M1/M2/b (Add_half_1_151)                       0.00       8.01 f
  C4/Z2/M2/M1/M2/U1/Z (CAN2X1)                            0.18       8.18 f
  C4/Z2/M2/M1/M2/c_out (Add_half_1_151)                   0.00       8.18 f
  C4/Z2/M2/M1/U2/Z (CND2IX1)                              0.18       8.37 f
  C4/Z2/M2/M1/c_out (Add_full_1_76)                       0.00       8.37 f
  C4/Z2/M2/M2/c_in (Add_full_1_75)                        0.00       8.37 f
  C4/Z2/M2/M2/M2/b (Add_half_1_149)                       0.00       8.37 f
  C4/Z2/M2/M2/M2/U1/Z (CAN2X1)                            0.16       8.53 f
  C4/Z2/M2/M2/M2/c_out (Add_half_1_149)                   0.00       8.53 f
  C4/Z2/M2/M2/U1/Z (COR2X1)                               0.23       8.76 f
  C4/Z2/M2/M2/c_out (Add_full_1_75)                       0.00       8.76 f
  C4/Z2/M2/M3/c_in (Add_full_1_74)                        0.00       8.76 f
  C4/Z2/M2/M3/M2/b (Add_half_1_147)                       0.00       8.76 f
  C4/Z2/M2/M3/M2/U2/Z (CAN2X1)                            0.16       8.91 f
  C4/Z2/M2/M3/M2/c_out (Add_half_1_147)                   0.00       8.91 f
  C4/Z2/M2/M3/U1/Z (COR2X1)                               0.24       9.16 f
  C4/Z2/M2/M3/c_out (Add_full_1_74)                       0.00       9.16 f
  C4/Z2/M2/M4/c_in (Add_full_1_73)                        0.00       9.16 f
  C4/Z2/M2/M4/M2/b (Add_half_1_145)                       0.00       9.16 f
  C4/Z2/M2/M4/M2/U1/Z (CAN2X1)                            0.17       9.33 f
  C4/Z2/M2/M4/M2/c_out (Add_half_1_145)                   0.00       9.33 f
  C4/Z2/M2/M4/U1/Z (COR2X1)                               0.24       9.57 f
  C4/Z2/M2/M4/c_out (Add_full_1_73)                       0.00       9.57 f
  C4/Z2/M2/c_out (Add_rca_4_1_19)                         0.00       9.57 f
  C4/Z2/M3/c_in (Add_rca_4_1_18)                          0.00       9.57 f
  C4/Z2/M3/M1/c_in (Add_full_1_72)                        0.00       9.57 f
  C4/Z2/M3/M1/M2/b (Add_half_1_143)                       0.00       9.57 f
  C4/Z2/M3/M1/M2/U1/Z (CAN2X1)                            0.16       9.73 f
  C4/Z2/M3/M1/M2/c_out (Add_half_1_143)                   0.00       9.73 f
  C4/Z2/M3/M1/U1/Z (COR2X1)                               0.24       9.97 f
  C4/Z2/M3/M1/c_out (Add_full_1_72)                       0.00       9.97 f
  C4/Z2/M3/M2/c_in (Add_full_1_71)                        0.00       9.97 f
  C4/Z2/M3/M2/M2/b (Add_half_1_141)                       0.00       9.97 f
  C4/Z2/M3/M2/M2/U1/Z (CAN2X1)                            0.16      10.13 f
  C4/Z2/M3/M2/M2/c_out (Add_half_1_141)                   0.00      10.13 f
  C4/Z2/M3/M2/U2/Z (CND2IX1)                              0.18      10.32 f
  C4/Z2/M3/M2/c_out (Add_full_1_71)                       0.00      10.32 f
  C4/Z2/M3/M3/c_in (Add_full_1_70)                        0.00      10.32 f
  C4/Z2/M3/M3/M2/b (Add_half_1_139)                       0.00      10.32 f
  C4/Z2/M3/M3/M2/U1/Z (CAN2X1)                            0.16      10.48 f
  C4/Z2/M3/M3/M2/c_out (Add_half_1_139)                   0.00      10.48 f
  C4/Z2/M3/M3/U1/Z (COR2X1)                               0.24      10.72 f
  C4/Z2/M3/M3/c_out (Add_full_1_70)                       0.00      10.72 f
  C4/Z2/M3/M4/c_in (Add_full_1_69)                        0.00      10.72 f
  C4/Z2/M3/M4/M2/b (Add_half_1_137)                       0.00      10.72 f
  C4/Z2/M3/M4/M2/U1/Z (CAN2X1)                            0.17      10.89 f
  C4/Z2/M3/M4/M2/c_out (Add_half_1_137)                   0.00      10.89 f
  C4/Z2/M3/M4/U1/Z (COR2X1)                               0.24      11.13 f
  C4/Z2/M3/M4/c_out (Add_full_1_69)                       0.00      11.13 f
  C4/Z2/M3/c_out (Add_rca_4_1_18)                         0.00      11.13 f
  C4/Z2/M4/c_in (Add_rca_4_1_17)                          0.00      11.13 f
  C4/Z2/M4/M1/c_in (Add_full_1_68)                        0.00      11.13 f
  C4/Z2/M4/M1/M2/b (Add_half_1_135)                       0.00      11.13 f
  C4/Z2/M4/M1/M2/U1/Z (CAN2X1)                            0.16      11.29 f
  C4/Z2/M4/M1/M2/c_out (Add_half_1_135)                   0.00      11.29 f
  C4/Z2/M4/M1/U1/Z (COR2X1)                               0.24      11.54 f
  C4/Z2/M4/M1/c_out (Add_full_1_68)                       0.00      11.54 f
  C4/Z2/M4/M2/c_in (Add_full_1_67)                        0.00      11.54 f
  C4/Z2/M4/M2/M2/b (Add_half_1_133)                       0.00      11.54 f
  C4/Z2/M4/M2/M2/U1/Z (CAN2X1)                            0.18      11.71 f
  C4/Z2/M4/M2/M2/c_out (Add_half_1_133)                   0.00      11.71 f
  C4/Z2/M4/M2/U2/Z (CND2IX1)                              0.18      11.89 f
  C4/Z2/M4/M2/c_out (Add_full_1_67)                       0.00      11.89 f
  C4/Z2/M4/M3/c_in (Add_full_1_66)                        0.00      11.89 f
  C4/Z2/M4/M3/M2/b (Add_half_1_131)                       0.00      11.89 f
  C4/Z2/M4/M3/M2/U2/Z (CAN2X1)                            0.16      12.05 f
  C4/Z2/M4/M3/M2/c_out (Add_half_1_131)                   0.00      12.05 f
  C4/Z2/M4/M3/U2/Z (CND2IX1)                              0.18      12.23 f
  C4/Z2/M4/M3/c_out (Add_full_1_66)                       0.00      12.23 f
  C4/Z2/M4/M4/c_in (Add_full_1_65)                        0.00      12.23 f
  C4/Z2/M4/M4/M2/b (Add_half_1_129)                       0.00      12.23 f
  C4/Z2/M4/M4/M2/U2/Z (CAN2X1)                            0.16      12.39 f
  C4/Z2/M4/M4/M2/c_out (Add_half_1_129)                   0.00      12.39 f
  C4/Z2/M4/M4/U1/Z (COR2X1)                               0.24      12.63 f
  C4/Z2/M4/M4/c_out (Add_full_1_65)                       0.00      12.63 f
  C4/Z2/M4/c_out (Add_rca_4_1_17)                         0.00      12.63 f
  C4/Z2/c_out (Add_rca_16_1_5)                            0.00      12.63 f
  C4/c_out (Add_rca_1_3)                                  0.00      12.63 f
  C5/c_in (Add_rca_1_2)                                   0.00      12.63 f
  C5/Z1/c_in (Add_rca_16_1_4)                             0.00      12.63 f
  C5/Z1/M1/c_in (Add_rca_4_1_16)                          0.00      12.63 f
  C5/Z1/M1/M1/c_in (Add_full_1_64)                        0.00      12.63 f
  C5/Z1/M1/M1/M2/b (Add_half_1_127)                       0.00      12.63 f
  C5/Z1/M1/M1/M2/U1/Z (CAN2X1)                            0.17      12.80 f
  C5/Z1/M1/M1/M2/c_out (Add_half_1_127)                   0.00      12.80 f
  C5/Z1/M1/M1/U1/Z (COR2X1)                               0.24      13.05 f
  C5/Z1/M1/M1/c_out (Add_full_1_64)                       0.00      13.05 f
  C5/Z1/M1/M2/c_in (Add_full_1_63)                        0.00      13.05 f
  C5/Z1/M1/M2/M2/b (Add_half_1_125)                       0.00      13.05 f
  C5/Z1/M1/M2/M2/U1/Z (CAN2X1)                            0.18      13.22 f
  C5/Z1/M1/M2/M2/c_out (Add_half_1_125)                   0.00      13.22 f
  C5/Z1/M1/M2/U2/Z (CND2IX1)                              0.17      13.39 f
  C5/Z1/M1/M2/c_out (Add_full_1_63)                       0.00      13.39 f
  C5/Z1/M1/M3/c_in (Add_full_1_62)                        0.00      13.39 f
  C5/Z1/M1/M3/M2/b (Add_half_1_123)                       0.00      13.39 f
  C5/Z1/M1/M3/M2/U2/Z (CAN2X1)                            0.16      13.55 f
  C5/Z1/M1/M3/M2/c_out (Add_half_1_123)                   0.00      13.55 f
  C5/Z1/M1/M3/U1/Z (COR2X1)                               0.24      13.79 f
  C5/Z1/M1/M3/c_out (Add_full_1_62)                       0.00      13.79 f
  C5/Z1/M1/M4/c_in (Add_full_1_61)                        0.00      13.79 f
  C5/Z1/M1/M4/M2/b (Add_half_1_121)                       0.00      13.79 f
  C5/Z1/M1/M4/M2/U1/Z (CAN2X1)                            0.16      13.95 f
  C5/Z1/M1/M4/M2/c_out (Add_half_1_121)                   0.00      13.95 f
  C5/Z1/M1/M4/U2/Z (CND2IX1)                              0.17      14.12 f
  C5/Z1/M1/M4/c_out (Add_full_1_61)                       0.00      14.12 f
  C5/Z1/M1/c_out (Add_rca_4_1_16)                         0.00      14.12 f
  C5/Z1/M2/c_in (Add_rca_4_1_15)                          0.00      14.12 f
  C5/Z1/M2/M1/c_in (Add_full_1_60)                        0.00      14.12 f
  C5/Z1/M2/M1/M2/b (Add_half_1_119)                       0.00      14.12 f
  C5/Z1/M2/M1/M2/U2/Z (CAN2X1)                             0.16      14.28 f
  C5/Z1/M2/M1/M2/c_out (Add_half_1_119)                    0.00      14.28 f
  C5/Z1/M2/M1/U1/Z (COR2X1)                                0.24      14.52 f
  C5/Z1/M2/M1/c_out (Add_full_1_60)                        0.00      14.52 f
  C5/Z1/M2/M2/c_in (Add_full_1_59)                         0.00      14.52 f
  C5/Z1/M2/M2/M2/b (Add_half_1_117)                        0.00      14.52 f
  C5/Z1/M2/M2/M2/U1/Z (CAN2X1)                             0.18      14.70 f
  C5/Z1/M2/M2/M2/c_out (Add_half_1_117)                    0.00      14.70 f
  C5/Z1/M2/M2/U2/Z (CND2IX1)                               0.19      14.89 f
  C5/Z1/M2/M2/c_out (Add_full_1_59)                        0.00      14.89 f
  C5/Z1/M2/M3/c_in (Add_full_1_58)                         0.00      14.89 f
  C5/Z1/M2/M3/M2/b (Add_half_1_115)                        0.00      14.89 f
  C5/Z1/M2/M3/M2/U1/Z (CAN2X1)                             0.18      15.06 f
  C5/Z1/M2/M3/M2/c_out (Add_half_1_115)                    0.00      15.06 f
  C5/Z1/M2/M3/U2/Z (CND2IX1)                               0.17      15.23 f
  C5/Z1/M2/M3/c_out (Add_full_1_58)                        0.00      15.23 f
  C5/Z1/M2/M4/c_in (Add_full_1_57)                         0.00      15.23 f
  C5/Z1/M2/M4/M2/b (Add_half_1_113)                        0.00      15.23 f
  C5/Z1/M2/M4/M2/U2/Z (CAN2X1)                             0.16      15.39 f
  C5/Z1/M2/M4/M2/c_out (Add_half_1_113)                    0.00      15.39 f
  C5/Z1/M2/M4/U1/Z (COR2X1)                                0.24      15.63 f
  C5/Z1/M2/M4/c_out (Add_full_1_57)                        0.00      15.63 f
  C5/Z1/M2/c_out (Add_rca_4_1_15)                          0.00      15.63 f
  C5/Z1/M3/c_in (Add_rca_4_1_14)                           0.00      15.63 f
  C5/Z1/M3/M1/c_in (Add_full_1_56)                         0.00      15.63 f
  C5/Z1/M3/M1/M2/b (Add_half_1_111)                        0.00      15.63 f
  C5/Z1/M3/M1/M2/U1/Z (CAN2X1)                             0.16      15.79 f
  C5/Z1/M3/M1/M2/c_out (Add_half_1_111)                    0.00      15.79 f
  C5/Z1/M3/M1/U1/Z (COR2X1)                                0.24      16.03 f
  C5/Z1/M3/M1/c_out (Add_full_1_56)                        0.00      16.03 f
  C5/Z1/M3/M2/c_in (Add_full_1_55)                         0.00      16.03 f
  C5/Z1/M3/M2/M2/b (Add_half_1_109)                        0.00      16.03 f
  C5/Z1/M3/M2/M2/U1/Z (CAN2X1)                             0.17      16.21 f
  C5/Z1/M3/M2/M2/c_out (Add_half_1_109)                    0.00      16.21 f
  C5/Z1/M3/M2/U1/Z (COR2X1)                                0.24      16.45 f
  C5/Z1/M3/M2/c_out (Add_full_1_55)                        0.00      16.45 f
  C5/Z1/M3/M3/c_in (Add_full_1_54)                         0.00      16.45 f
  C5/Z1/M3/M3/M2/b (Add_half_1_107)                        0.00      16.45 f
  C5/Z1/M3/M3/M2/U1/Z (CAN2X1)                             0.18      16.63 f
  C5/Z1/M3/M3/M2/c_out (Add_half_1_107)                    0.00      16.63 f
  C5/Z1/M3/M3/U2/Z (CND2IX1)                               0.17      16.80 f
  C5/Z1/M3/M3/c_out (Add_full_1_54)                        0.00      16.80 f
  C5/Z1/M3/M4/c_in (Add_full_1_53)                         0.00      16.80 f
  C5/Z1/M3/M4/M2/b (Add_half_1_105)                        0.00      16.80 f
  C5/Z1/M3/M4/M2/U2/Z (CAN2X1)                             0.16      16.95 f
  C5/Z1/M3/M4/M2/c_out (Add_half_1_105)                    0.00      16.95 f
  C5/Z1/M3/M4/U1/Z (COR2X1)                                0.24      17.19 f
  C5/Z1/M3/M4/c_out (Add_full_1_53)                        0.00      17.19 f
  C5/Z1/M3/c_out (Add_rca_4_1_14)                          0.00      17.19 f
  C5/Z1/M4/c_in (Add_rca_4_1_13)                           0.00      17.19 f
  C5/Z1/M4/M1/c_in (Add_full_1_52)                         0.00      17.19 f
  C5/Z1/M4/M1/M2/b (Add_half_1_103)                        0.00      17.19 f
  C5/Z1/M4/M1/M2/U1/Z (CAN2X1)                             0.16      17.35 f
  C5/Z1/M4/M1/M2/c_out (Add_half_1_103)                    0.00      17.35 f
  C5/Z1/M4/M1/U1/Z (COR2X1)                                0.24      17.59 f
  C5/Z1/M4/M1/c_out (Add_full_1_52)                        0.00      17.59 f
  C5/Z1/M4/M2/c_in (Add_full_1_51)                         0.00      17.59 f
  C5/Z1/M4/M2/M2/b (Add_half_1_101)                       0.00      17.59 f
  C5/Z1/M4/M2/M2/U1/Z (CAN2X1)                            0.16      17.76 f
  C5/Z1/M4/M2/M2/c_out (Add_half_1_101)                   0.00      17.76 f
  C5/Z1/M4/M2/U2/Z (CND2IX1)                              0.19      17.94 f
  C5/Z1/M4/M2/c_out (Add_full_1_51)                       0.00      17.94 f
  C5/Z1/M4/M3/c_in (Add_full_1_50)                        0.00      17.94 f
  C5/Z1/M4/M3/M2/b (Add_half_1_99)                        0.00      17.94 f
  C5/Z1/M4/M3/M2/U2/Z (CAN2X1)                            0.18      18.12 f
  C5/Z1/M4/M3/M2/c_out (Add_half_1_99)                    0.00      18.12 f
  C5/Z1/M4/M3/U2/Z (CND2IX1)                              0.19      18.30 f
  C5/Z1/M4/M3/c_out (Add_full_1_50)                       0.00      18.30 f
  C5/Z1/M4/M4/c_in (Add_full_1_49)                        0.00      18.30 f
  C5/Z1/M4/M4/M2/b (Add_half_1_97)                        0.00      18.30 f
  C5/Z1/M4/M4/M2/U1/Z (CAN2X1)                            0.18      18.48 f
  C5/Z1/M4/M4/M2/c_out (Add_half_1_97)                    0.00      18.48 f
  C5/Z1/M4/M4/U2/Z (CND2IX1)                              0.19      18.66 f
  C5/Z1/M4/M4/c_out (Add_full_1_49)                       0.00      18.66 f
  C5/Z1/M4/c_out (Add_rca_4_1_13)                         0.00      18.66 f
  C5/Z1/c_out (Add_rca_16_1_4)                            0.00      18.66 f
  C5/Z2/c_in (Add_rca_16_1_3)                             0.00      18.66 f
  C5/Z2/M1/c_in (Add_rca_4_1_12)                          0.00      18.66 f
  C5/Z2/M1/M1/c_in (Add_full_1_48)                        0.00      18.66 f
  C5/Z2/M1/M1/M2/b (Add_half_1_95)                        0.00      18.66 f
  C5/Z2/M1/M1/M2/U1/Z (CAN2X1)                            0.17      18.84 f
  C5/Z2/M1/M1/M2/c_out (Add_half_1_95)                    0.00      18.84 f
  C5/Z2/M1/M1/U1/Z (COR2X1)                               0.23      19.07 f
  C5/Z2/M1/M1/c_out (Add_full_1_48)                       0.00      19.07 f
  C5/Z2/M1/M2/c_in (Add_full_1_47)                        0.00      19.07 f
  C5/Z2/M1/M2/M2/b (Add_half_1_93)                        0.00      19.07 f
  C5/Z2/M1/M2/M2/U2/Z (CAN2X1)                            0.16      19.23 f
  C5/Z2/M1/M2/M2/c_out (Add_half_1_93)                    0.00      19.23 f
  C5/Z2/M1/M2/U2/Z (CND2IX1)                              0.18      19.41 f
  C5/Z2/M1/M2/c_out (Add_full_1_47)                       0.00      19.41 f
  C5/Z2/M1/M3/c_in (Add_full_1_46)                        0.00      19.41 f
  C5/Z2/M1/M3/M2/b (Add_half_1_91)                        0.00      19.41 f
  C5/Z2/M1/M3/M2/U2/Z (CAN2X1)                            0.16      19.57 f
  C5/Z2/M1/M3/M2/c_out (Add_half_1_91)                    0.00      19.57 f
  C5/Z2/M1/M3/U2/Z (CND2IX1)                              0.19      19.75 f
  C5/Z2/M1/M3/c_out (Add_full_1_46)                       0.00      19.75 f
  C5/Z2/M1/M4/c_in (Add_full_1_45)                        0.00      19.75 f
  C5/Z2/M1/M4/M2/b (Add_half_1_89)                        0.00      19.75 f
  C5/Z2/M1/M4/M2/U1/Z (CAN2X1)                            0.18      19.93 f
  C5/Z2/M1/M4/M2/c_out (Add_half_1_89)                    0.00      19.93 f
  C5/Z2/M1/M4/U2/Z (CND2IX1)                              0.18      20.11 f
  C5/Z2/M1/M4/c_out (Add_full_1_45)                       0.00      20.11 f
  C5/Z2/M1/c_out (Add_rca_4_1_12)                         0.00      20.11 f
  C5/Z2/M2/c_in (Add_rca_4_1_11)                          0.00      20.11 f
  C5/Z2/M2/M1/c_in (Add_full_1_44)                        0.00      20.11 f
  C5/Z2/M2/M1/M2/b (Add_half_1_87)                        0.00      20.11 f
  C5/Z2/M2/M1/M2/U1/Z (CAN2X1)                            0.16      20.27 f
  C5/Z2/M2/M1/M2/c_out (Add_half_1_87)                    0.00      20.27 f
  C5/Z2/M2/M1/U2/Z (CND2IX1)                              0.19      20.46 f
  C5/Z2/M2/M1/c_out (Add_full_1_44)                       0.00      20.46 f
  C5/Z2/M2/M2/c_in (Add_full_1_43)                        0.00      20.46 f
  C5/Z2/M2/M2/M2/b (Add_half_1_85)                        0.00      20.46 f
  C5/Z2/M2/M2/M2/U1/Z (CAN2X1)                            0.18      20.63 f
  C5/Z2/M2/M2/M2/c_out (Add_half_1_85)                    0.00      20.63 f
  C5/Z2/M2/M2/U2/Z (CND2IX1)                                  0.19      20.82 f
  C5/Z2/M2/M2/c_out (Add_full_1_43)                           0.00      20.82 f
  C5/Z2/M2/M3/c_in (Add_full_1_42)                            0.00      20.82 f
  C5/Z2/M2/M3/M2/b (Add_half_1_83)                            0.00      20.82 f
  C5/Z2/M2/M3/M2/U2/Z (CAN2X1)                                0.18      21.00 f
  C5/Z2/M2/M3/M2/c_out (Add_half_1_83)                        0.00      21.00 f
  C5/Z2/M2/M3/U2/Z (CND2IX1)                                  0.18      21.17 f
  C5/Z2/M2/M3/c_out (Add_full_1_42)                           0.00      21.17 f
  C5/Z2/M2/M4/c_in (Add_full_1_41)                            0.00      21.17 f
  C5/Z2/M2/M4/M2/b (Add_half_1_81)                            0.00      21.17 f
  C5/Z2/M2/M4/M2/U3/Z (CAN2X1)                                0.16      21.33 f
  C5/Z2/M2/M4/M2/c_out (Add_half_1_81)                        0.00      21.33 f
  C5/Z2/M2/M4/U2/Z (CND2IX1)                                  0.17      21.51 f
  C5/Z2/M2/M4/c_out (Add_full_1_41)                           0.00      21.51 f
  C5/Z2/M2/c_out (Add_rca_4_1_11)                             0.00      21.51 f
  C5/Z2/M3/c_in (Add_rca_4_1_10)                              0.00      21.51 f
  C5/Z2/M3/M1/c_in (Add_full_1_40)                            0.00      21.51 f
  C5/Z2/M3/M1/M2/b (Add_half_1_79)                            0.00      21.51 f
  C5/Z2/M3/M1/M2/U4/Z (CAN2X1)                                0.16      21.66 f
  C5/Z2/M3/M1/M2/c_out (Add_half_1_79)                        0.00      21.66 f
  C5/Z2/M3/M1/U2/Z (CND2IX1)                                  0.19      21.85 f
  C5/Z2/M3/M1/c_out (Add_full_1_40)                           0.00      21.85 f
  C5/Z2/M3/M2/c_in (Add_full_1_39)                            0.00      21.85 f
  C5/Z2/M3/M2/M2/b (Add_half_1_77)                            0.00      21.85 f
  C5/Z2/M3/M2/M2/U2/Z (CAN2X1)                                0.18      22.03 f
  C5/Z2/M3/M2/M2/c_out (Add_half_1_77)                        0.00      22.03 f
  C5/Z2/M3/M2/U2/Z (CND2IX1)                                  0.19      22.21 f
  C5/Z2/M3/M2/c_out (Add_full_1_39)                           0.00      22.21 f
  C5/Z2/M3/M3/c_in (Add_full_1_38)                            0.00      22.21 f
  C5/Z2/M3/M3/M2/b (Add_half_1_75)                            0.00      22.21 f
  C5/Z2/M3/M3/M2/U2/Z (CAN2X1)                                0.18      22.39 f
  C5/Z2/M3/M3/M2/c_out (Add_half_1_75)                        0.00      22.39 f
  C5/Z2/M3/M3/U2/Z (CND2IX1)                                  0.18      22.57 f
  C5/Z2/M3/M3/c_out (Add_full_1_38)                           0.00      22.57 f
  C5/Z2/M3/M4/c_in (Add_full_1_37)                            0.00      22.57 f
  C5/Z2/M3/M4/M2/b (Add_half_1_73)                            0.00      22.57 f
  C5/Z2/M3/M4/M2/U2/Z (CAN2X1)                                0.16      22.73 f
  C5/Z2/M3/M4/M2/c_out (Add_half_1_73)                        0.00      22.73 f
  C5/Z2/M3/M4/U2/Z (CND2IX1)                                  0.17      22.90 f
  C5/Z2/M3/M4/c_out (Add_full_1_37)                           0.00      22.90 f
  C5/Z2/M3/c_out (Add_rca_4_1_10)                             0.00      22.90 f
  C5/Z2/M4/c_in (Add_rca_4_1_9)                               0.00      22.90 f
  C5/Z2/M4/M1/c_in (Add_full_1_36)                            0.00      22.90 f
  C5/Z2/M4/M1/M2/b (Add_half_1_71)                            0.00      22.90 f
  C5/Z2/M4/M1/M2/U3/Z (CAN2X1)                                0.16      23.06 f
  C5/Z2/M4/M1/M2/c_out (Add_half_1_71)                        0.00      23.06 f
  C5/Z2/M4/M1/U2/Z (CND2IX1)                                  0.18      23.25 f
  C5/Z2/M4/M1/c_out (Add_full_1_36)                           0.00      23.25 f
  C5/Z2/M4/M2/c_in (Add_full_1_35)                            0.00      23.25 f
  C5/Z2/M4/M2/M2/b (Add_half_1_69)                            0.00      23.25 f
  C5/Z2/M4/M2/M2/U2/Z (CAN2X1)                                0.16      23.41 f
  C5/Z2/M4/M2/M2/c_out (Add_half_1_69)                        0.00      23.41 f
  C5/Z2/M4/M2/U2/Z (CND2IX1)                                  0.18      23.59 f
  C5/Z2/M4/M2/c_out (Add_full_1_35)                           0.00      23.59 f
  C5/Z2/M4/M3/c_in (Add_full_1_34)                            0.00      23.59 f
  C5/Z2/M4/M3/M2/b (Add_half_1_67)                            0.00      23.59 f
  C5/Z2/M4/M3/M2/U2/Z (CAN2X1)                                0.16      23.75 f
  C5/Z2/M4/M3/M2/c_out (Add_half_1_67)                    0.00      23.75 f
  C5/Z2/M4/M3/U1/Z (CND2IX1)                              0.19      23.94 f
  C5/Z2/M4/M3/c_out (Add_full_1_34)                       0.00      23.94 f
  C5/Z2/M4/M4/c_in (Add_full_1_33)                        0.00      23.94 f
  C5/Z2/M4/M4/M2/b (Add_half_1_65)                        0.00      23.94 f
  C5/Z2/M4/M4/M2/U2/Z (CND2X1)                            0.08      24.02 r
  C5/Z2/M4/M4/M2/U5/Z (CND2X1)                            0.11      24.13 f
  C5/Z2/M4/M4/M2/sum (Add_half_1_65)                      0.00      24.13 f
  C5/Z2/M4/M4/sum (Add_full_1_33)                         0.00      24.13 f
  C5/Z2/M4/sum[3] (Add_rca_4_1_9)                         0.00      24.13 f
  C5/Z2/sum[15] (Add_rca_16_1_3)                          0.00      24.13 f
  C5/sum[31] (Add_rca_1_2)                                0.00      24.13 f
  U739/Z (CND2IX1)                                        0.07      24.20 r
  U759/Z (CND2X1)                                         0.09      24.29 f
  result_reg[63]/D (CFD1QXL)                              0.00      24.29 f
  data arrival time                                                 24.29

  clock clock (rise edge)                                25.00      25.00
  clock network delay (propagated)                        0.00      25.00
  clock uncertainty                                      -0.25      24.75
  result_reg[63]/CP (CFD1QXL)                             0.00      24.75 r
  library setup time                                     -0.46      24.29
  data required time                                                24.29
  -------------------------------------------------------------------------
  data required time                                                24.29
  data arrival time                                                -24.29
  -------------------------------------------------------------------------
  slack (MET)                                                        0.00


****************************************
Report : area
Design : division
Version: C-2009.06-SP5
Date   : Sat Dec 10 17:41:18 2016
****************************************

Library(s) Used:

    tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25)

Number of ports:              165
Number of nets:              1319
Number of cells:              928
Number of references:          58

Combinational area:       3774.500000
Noncombinational area:     628.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          4402.500000
Total area:                 undefined
1


****************************************
Report : power
```

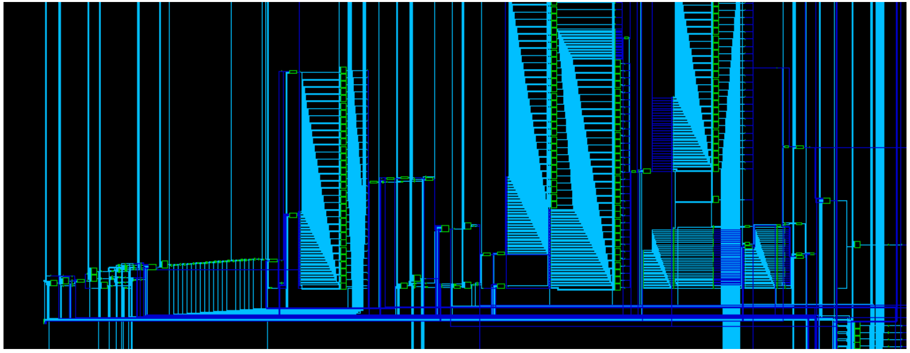
```
        -analysis_effort low
Design : division
Version: C-2009.06-SP5
Date   : Sat Dec 10 17:41:18 2016
****************************************


Library(s) Used:

    tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db_WCCOM25)


Operating Conditions: WCCOM25   Library: tc240c
Wire Load Model Mode: top


Global Operating Voltage = 2.3
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = Unitless


  Cell Internal Power  =   1.7617 mW   (83%)
  Net Switching Power  = 366.9811 uW   (17%)
                         ---------
Total Dynamic Power    =   2.1286 mW  (100%)

Cell Leakage Power     =   0.0000

1
```

C.4 Selected Screenshot Circuits from Synthesis (Design Compiler)

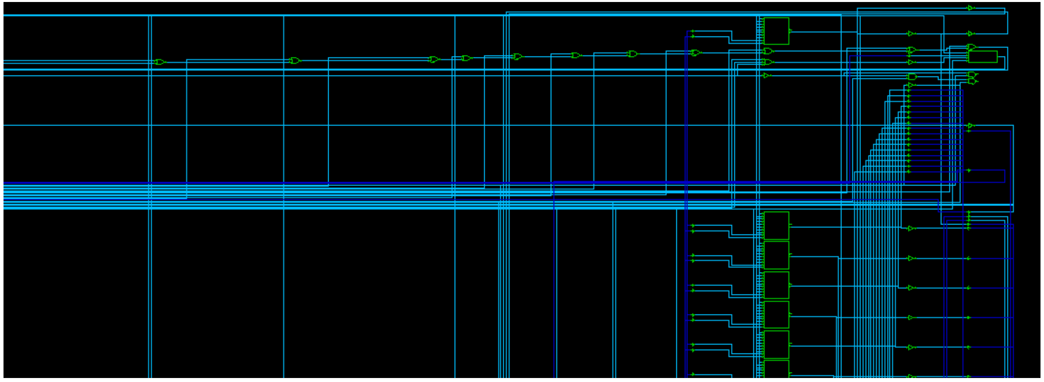
Circuit for Multiplication



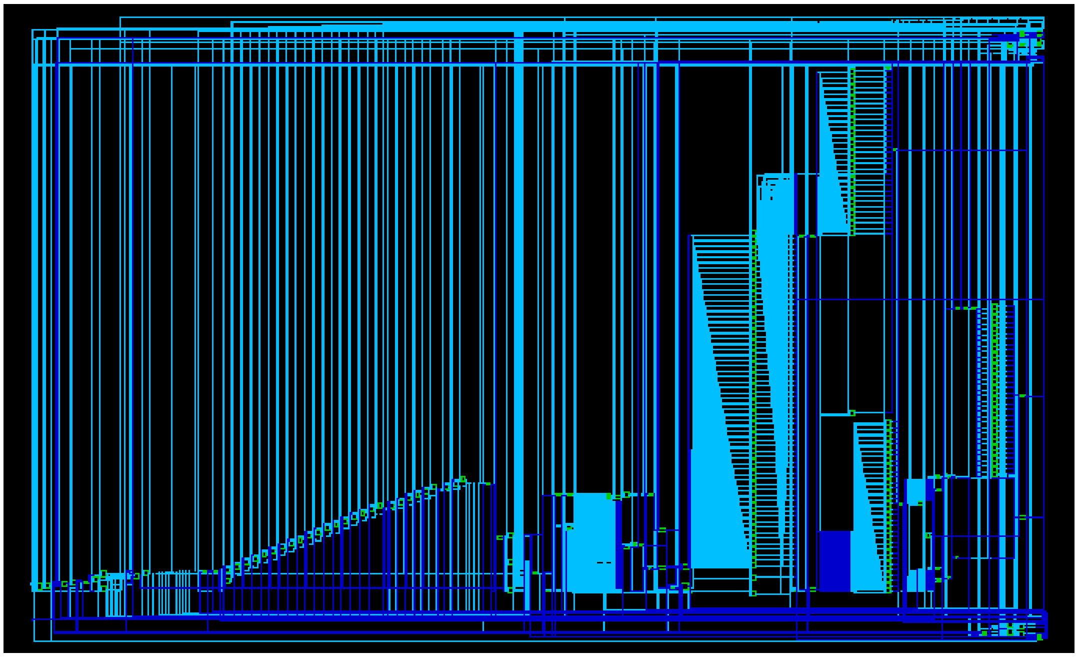
Input Section - Multiplication



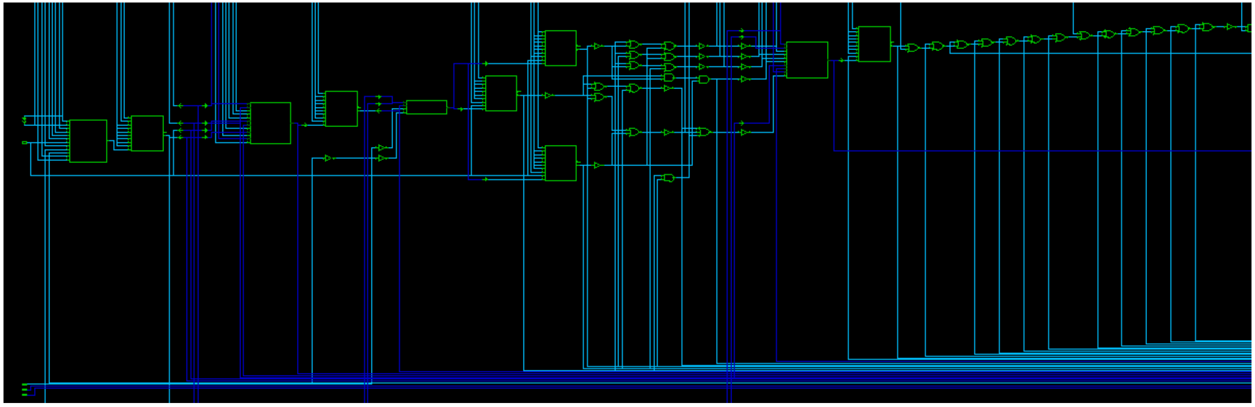
Output Section - Multiplication



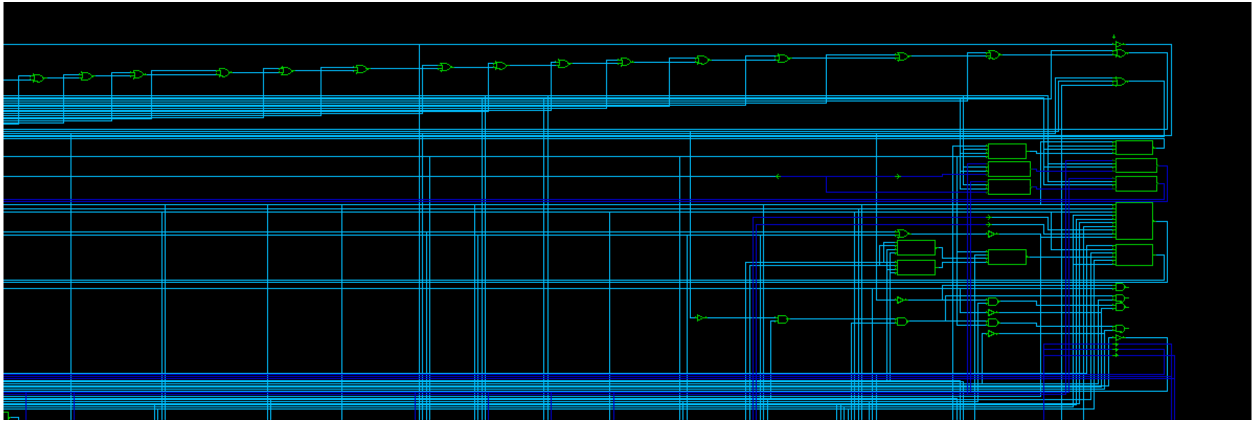
Circuit for Division



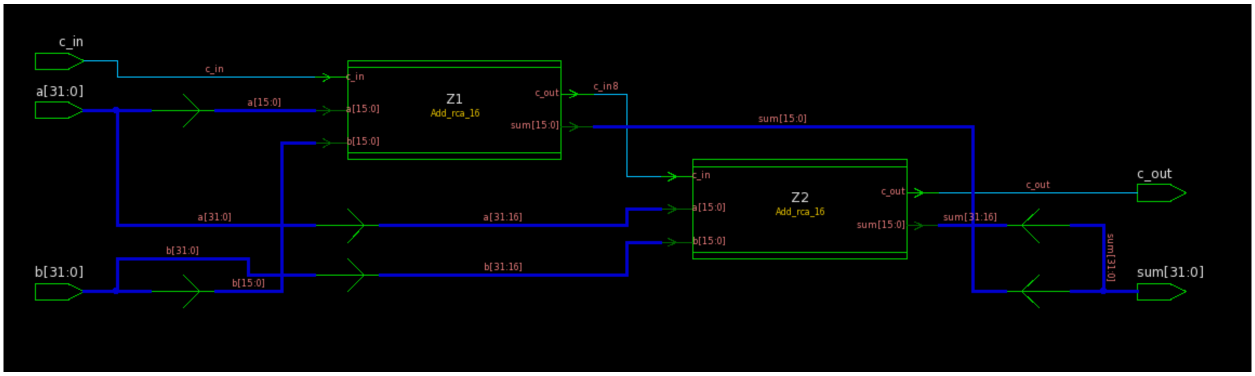
Input Section



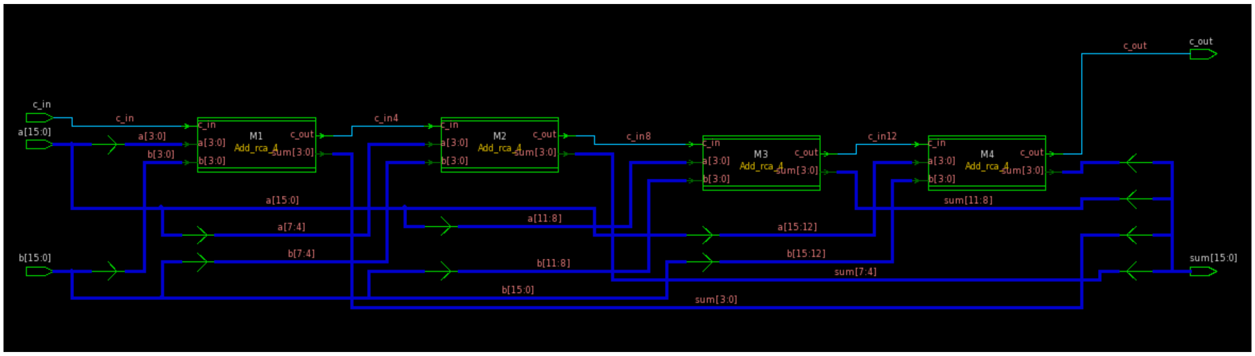
Output Section



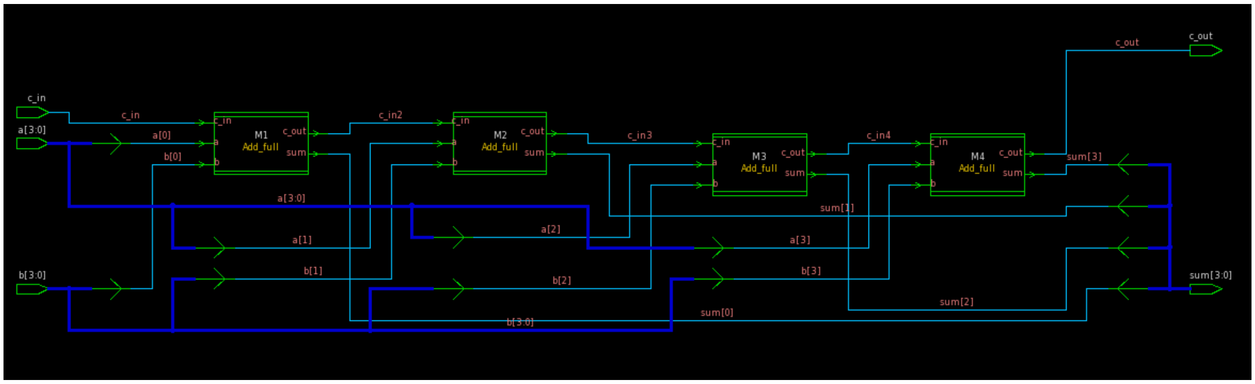
32 Bit Adder



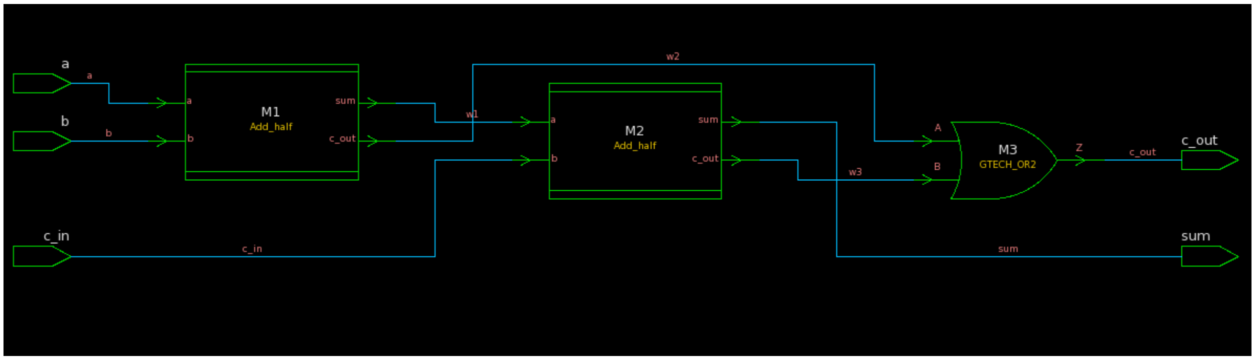
16 Bit RCA Adder



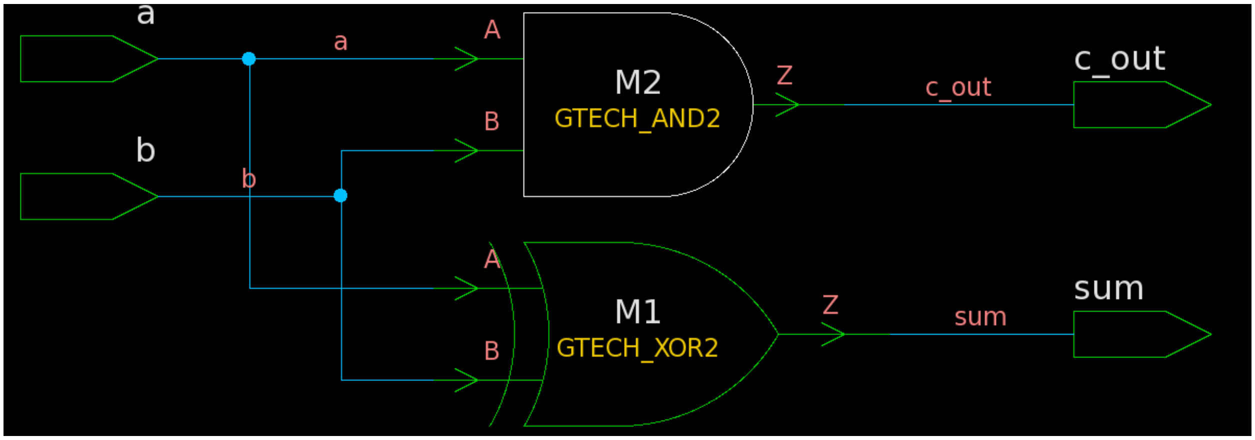
4 Bit RCA Adder



Full Adder



Half Adder


a
a
A
M2
GTECH_AND2
Z
c_out
c_out
B
b
b
A
M1
GTECH_XOR2
Z
sum
sum
B