INTEL UNNATI INDUSTRIAL TRAINING SUMMER 2023

PROBLEM STATEMENT	**DESIGN & IMPLEMENTATION OF SLOW AND FAST DIVISION ALGORITHM IN COMPUTER ARCHITECTURE**
NAME	**Ms.N.TURIKA**
INSTITUTE NAME	**B.S.ABDUR RAHMAN CRESCENT INSTITUTE OF SCIENCE AND TECHNOLOGY**
TEAM NAME	**FIXER**
TEAM SIZE	**1**
MENTOR NAME	**DR.S.ANUSOOYA**

FAST DIVISION ALGORITHM

In this method, the quotient is predicted to the closest approximation to the actual quotient, and then the calculation starts. The algorithms for fast division can be done through the methods Newton–Raphson, and Goldschmidt.

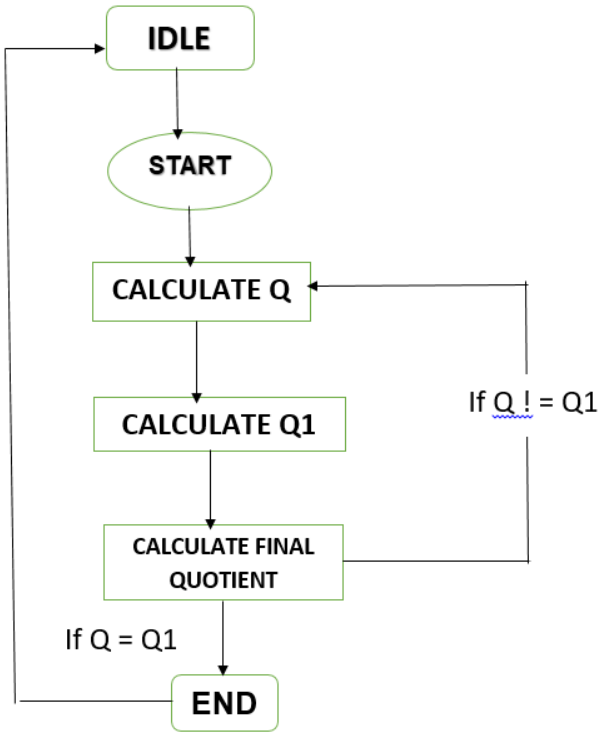
NEWTON-RAPHSON DIVISION ALGORITHM

Newton-Raphson's division algorithm is an iterative method used to approximate the division operation. It is based on finding roots of equations. The algorithm aims to find the reciprocal of the divisor and then multiply it with the dividend to obtain the quotient.

Here are the steps involved in Newton-Raphson's division algorithm:

1. Initialize the reciprocal estimate (Q) to an initial guess, such as 1/divisor.
2. Update the reciprocal estimate using the formula:
   Q1= Q * (2 - divisor * Q).
3. Repeat step 2 until the reciprocal estimate converges to a desired accuracy (i.e., Q = Q1).
4. Calculate the quotient by multiplying the dividend with the final reciprocal estimate (Q1).

MEALY STATE MACHINE



MANUAL CALCULATION

TEST CASE (1)

Dividend = 7 ; Divisor = 2

1) Q = 1/divisor = 0.5

2) Q1 = Q (2 – divisor (Q))
   = 0.5 (2 – 2 (0.5))
   =0.5
3) As Q = Q1

Final quotient = Dividend * Q1
= 7 * 0.5
= 3.5

TEST CASE (2)

Dividend = 13 ; Divisor = 4

1) Q = 1/divisor = 0.25
2) Q1 = Q (2 – divisor (Q))
   = 0.25 (2 – 4 (0.25))
   =0.25
3) As Q = Q1

Final quotient = Dividend * Q1
=13 * 0.25
=3.25

VERILOG CODE

```
module fast_division1 (
  input wire clk,
  input wire start,
  input wire [3:0] dividend,
  input wire [3:0] divisor,
  output wire [3:0] quotient
);

  reg [10:0] Q;
  reg [10:0] Q1;
  reg [10:0] final_quotient;
  reg [2:0] state;

  always @(posedge clk) begin
    case (state)
      0: begin // Idle state, waiting for start signal
        if (start) begin
```

```
        Q <= 0;
        Q1 <= 0;
        final_quotient <= 0;
        state <= 1;
       end
      end
      1: begin // Step 1: Q = (1 << N) / divisor
       Q = ({2'b00, dividend} << 6) / divisor;
       state <= 2;
      end
      2: begin // Step 2: Q1 = Q * (2^N - divisor * Q)
       Q1 = Q * (({2'b00, 2'b00, 4'b0010} << 6) - divisor * Q);
       if (Q == Q1)
        state <= 4; // Proceed to Step 4 if Q = Q1
       else
        state <= 3; // Repeat Step 2 if Q is not equal to Q1
      end
      3: begin // Step 2 (repeat): Q1 = Q * (2^N - divisor * Q)
       Q = Q1;
       Q1 = Q * (({2'b00, 2'b00, 4'b0010} << 6) - divisor * Q);
       state <= 2;
      end
      4: begin // Step 3: Calculate final quotient as dividend * Q1 >> N
       final_quotient = dividend * Q1 >> 6;
       state <= 0; // Return to Idle state
      end
    endcase
  end

  assign quotient = {final_quotient[7:4], final_quotient[3]} + (final_quotient[2] ? 1'b1 :
1'b0);

endmodule
```

TEST BENCH

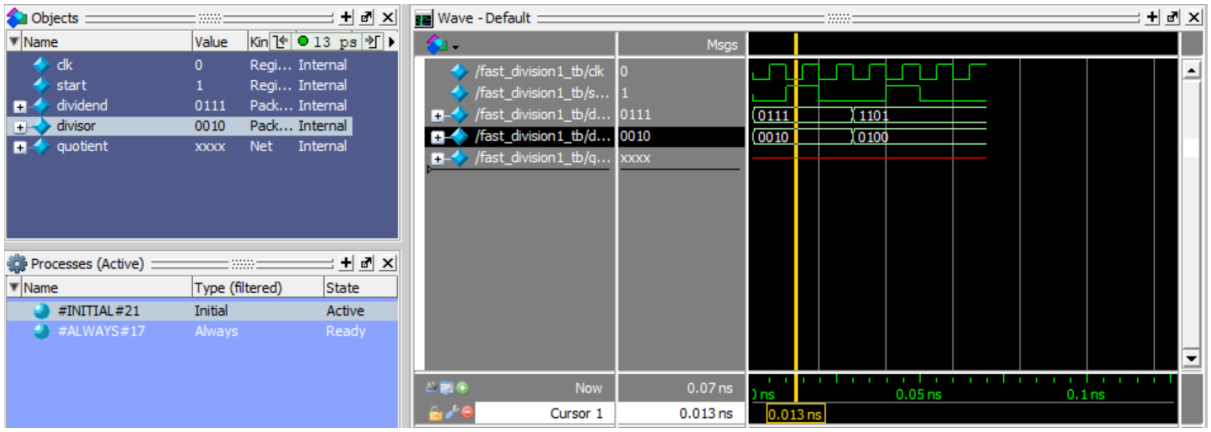
```
module fast_division1_tb;

  reg clk;
  reg start;
  reg [3:0] dividend;
  reg [3:0] divisor;
```

```
  wire [3:0] quotient;

  fast_division1 DUT (
    .clk(clk),
    .start(start),
    .dividend(dividend),
    .divisor(divisor),
    .quotient(quotient)
  );

  always begin
    #5 clk = ~clk; // Toggle the clock every 10 time units
  end

  initial begin
    clk = 0;
    start = 0;
    dividend = 4'b0111;
    divisor = 4'b0010;

    #10;
    start = 1;
    #10;
    start = 0;
    #10;
    $display("Test case 1: dividend = %d, divisor = %d, quotient = %0d.%01d",
dividend, divisor, quotient[3:2], quotient[1]);

    dividend = 4'b1101;
    divisor = 4'b0100;

    #10;
    start = 1;
    #10;
    start = 0;
    #10;
    $display("Test case 2: dividend = %d, divisor = %d, quotient = %0d.%01d",
dividend, divisor, quotient[3:2], quotient[1]);

    #10;
    $finish; // End the simulation
  end
endmodule
```

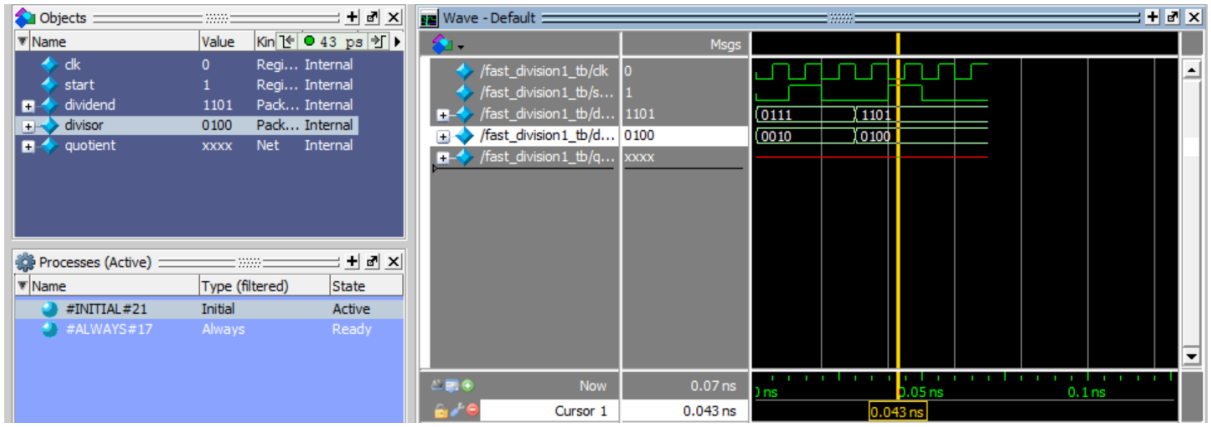
SIMULATION RESULT

```
# .main_pane.objects.interior.cs.body.tree
# run -all
# Test case 1: dividend =  7, divisor =  2, quotient = x.x
# Test case 2: dividend = 13, divisor =  4, quotient = x.x
# ** Note: $finish    : C:/unnati_fastdivision/fast_division1_tb.v(45)
#    Time: 70 ps  Iteration: 0  Instance: /fast_division1_tb
# 1
```

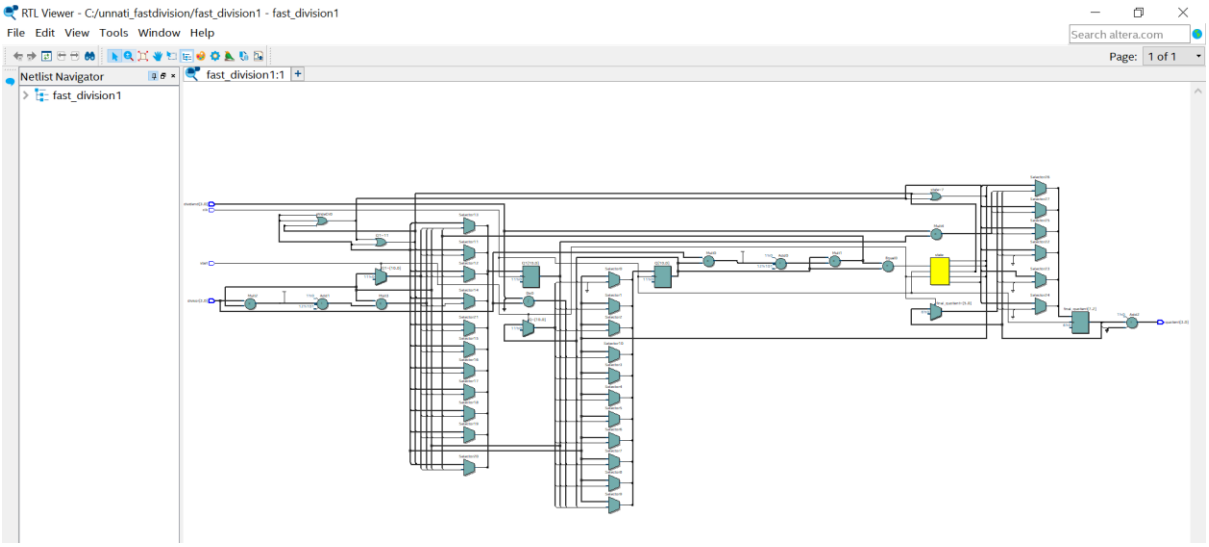
RESULT OF TEST CASE (1)



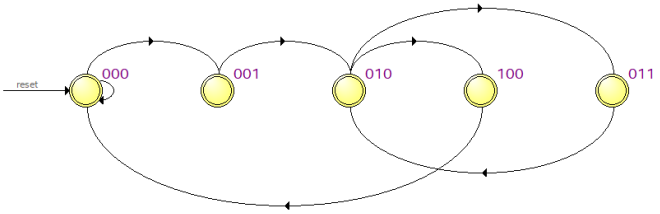
RESULT OF TEST CASE (2)



RTL VIEWER



STATE MACHINE VIEWER

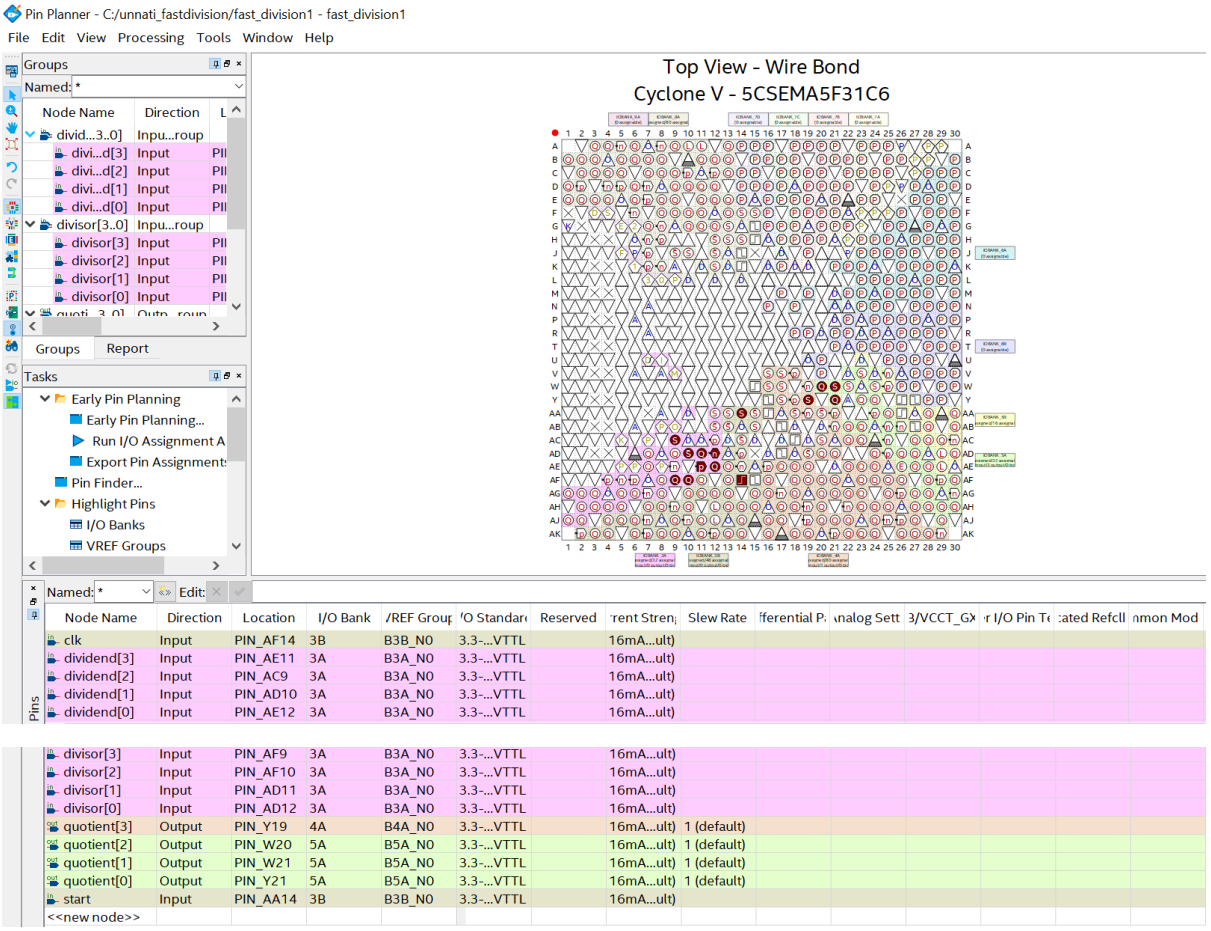


State Table

	Source State	Destination State	Condition
1	000	000	(!start)
2	000	001	(start)
3	001	010	
4	010	011	(!Equal0)
5	010	100	(Equal0)
6	011	010	
7	100	000	

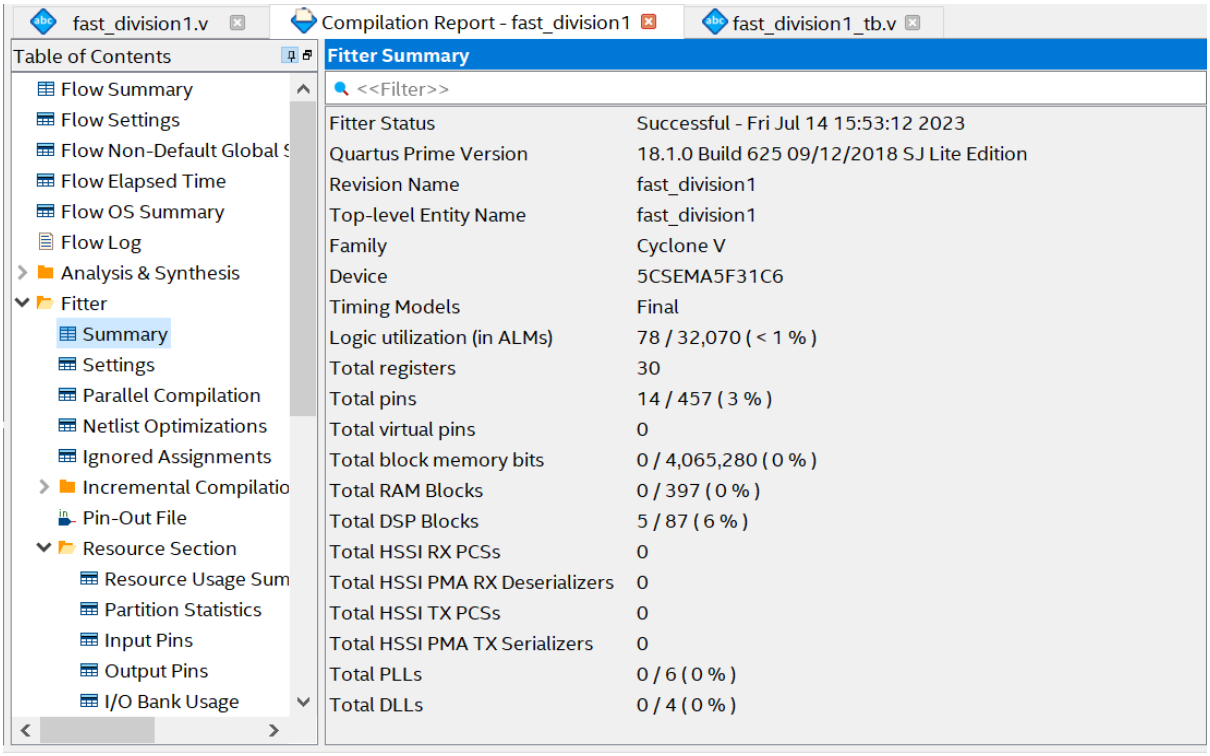
Transitions | Encoding

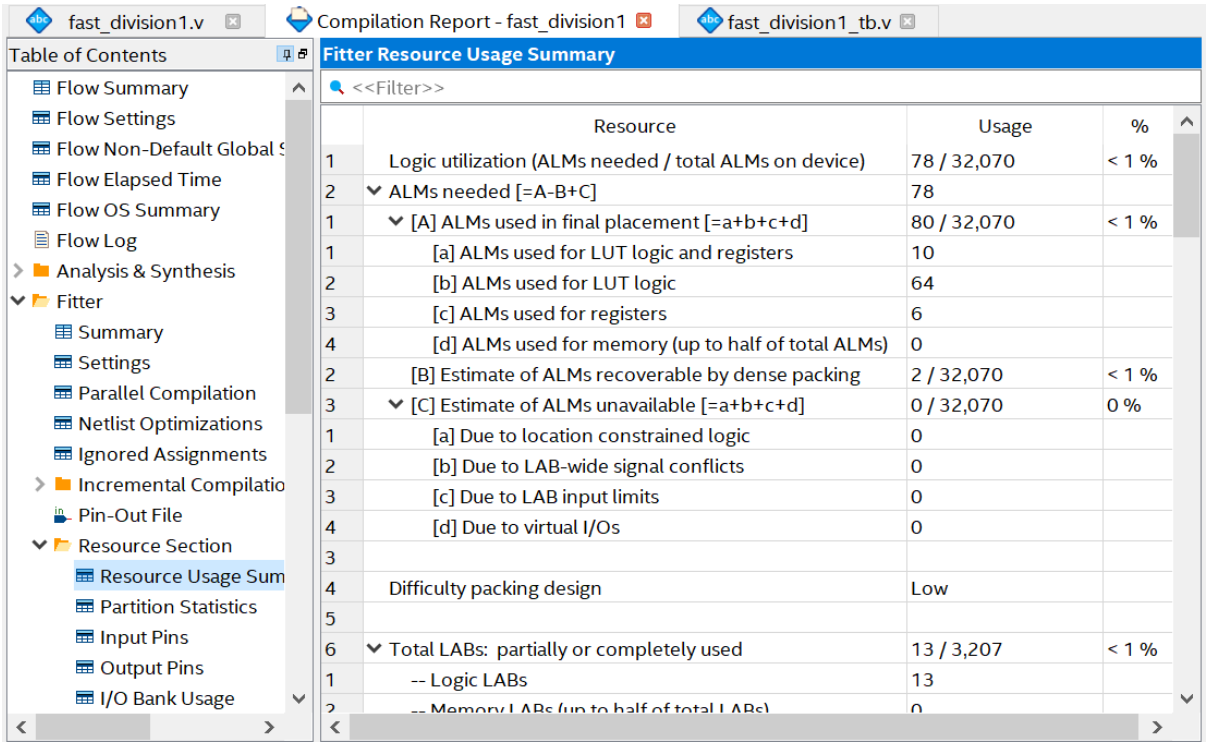
PIN PLANNER

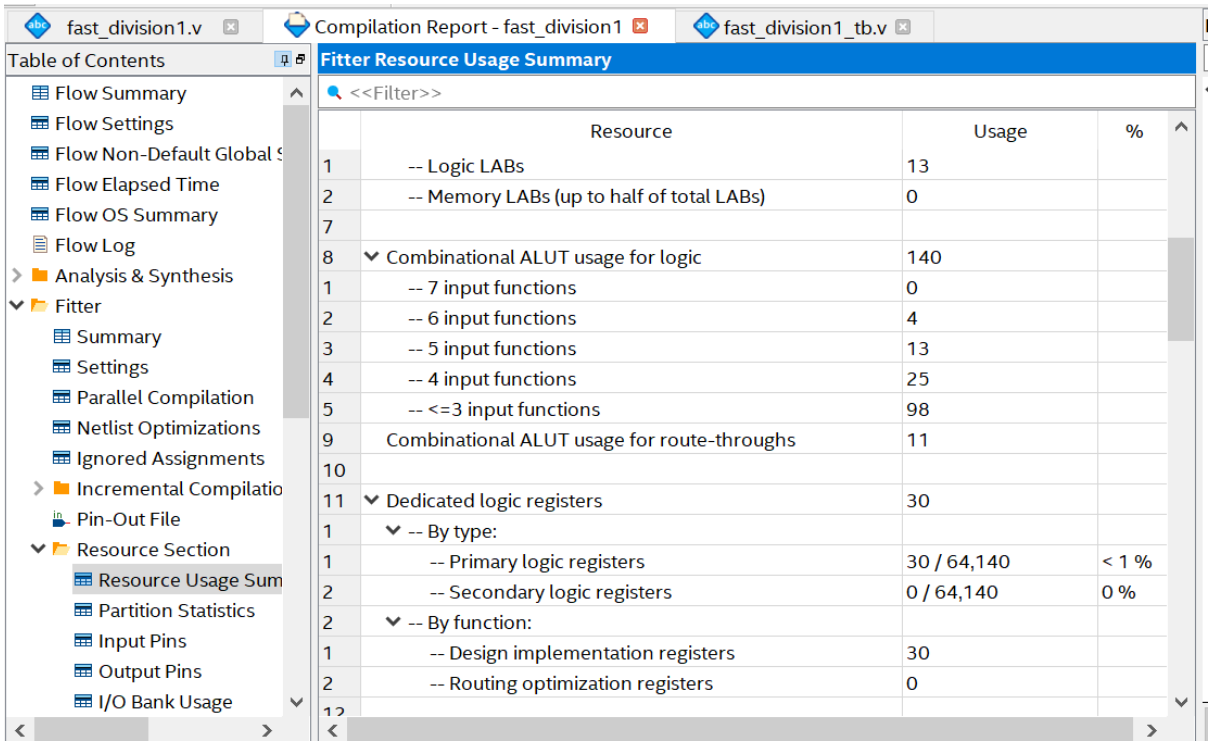


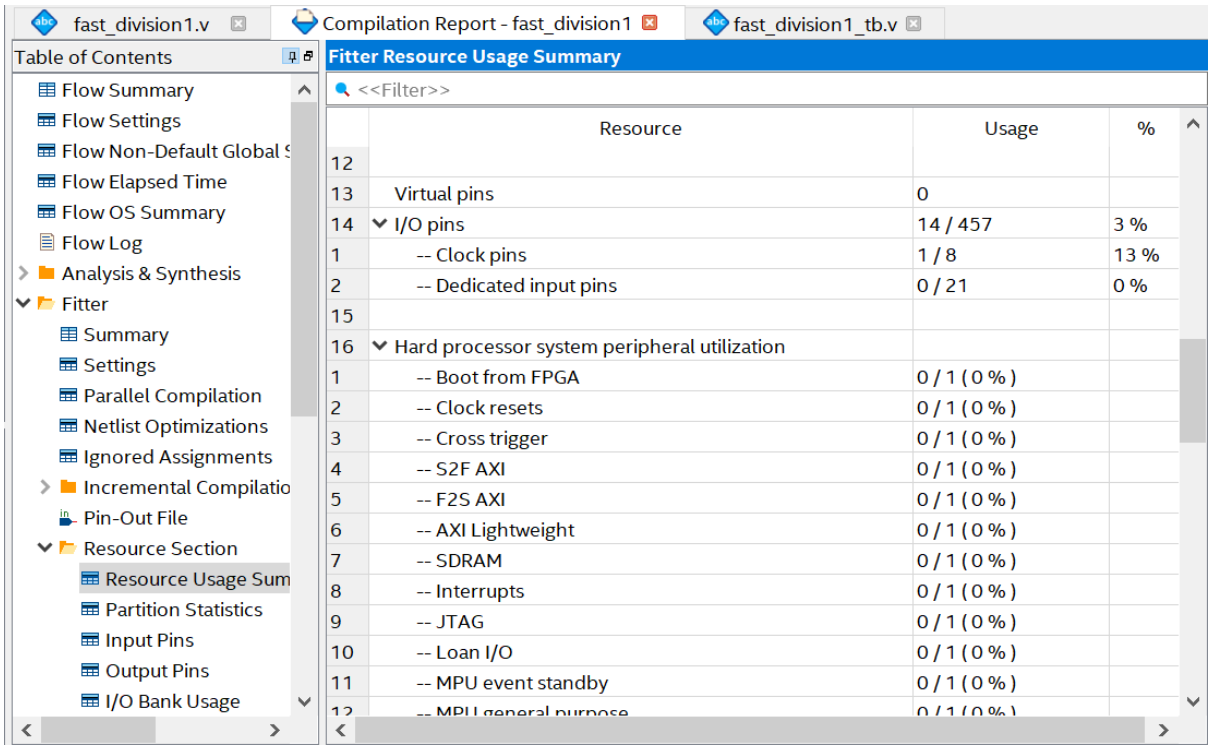
CHIP PLANNER

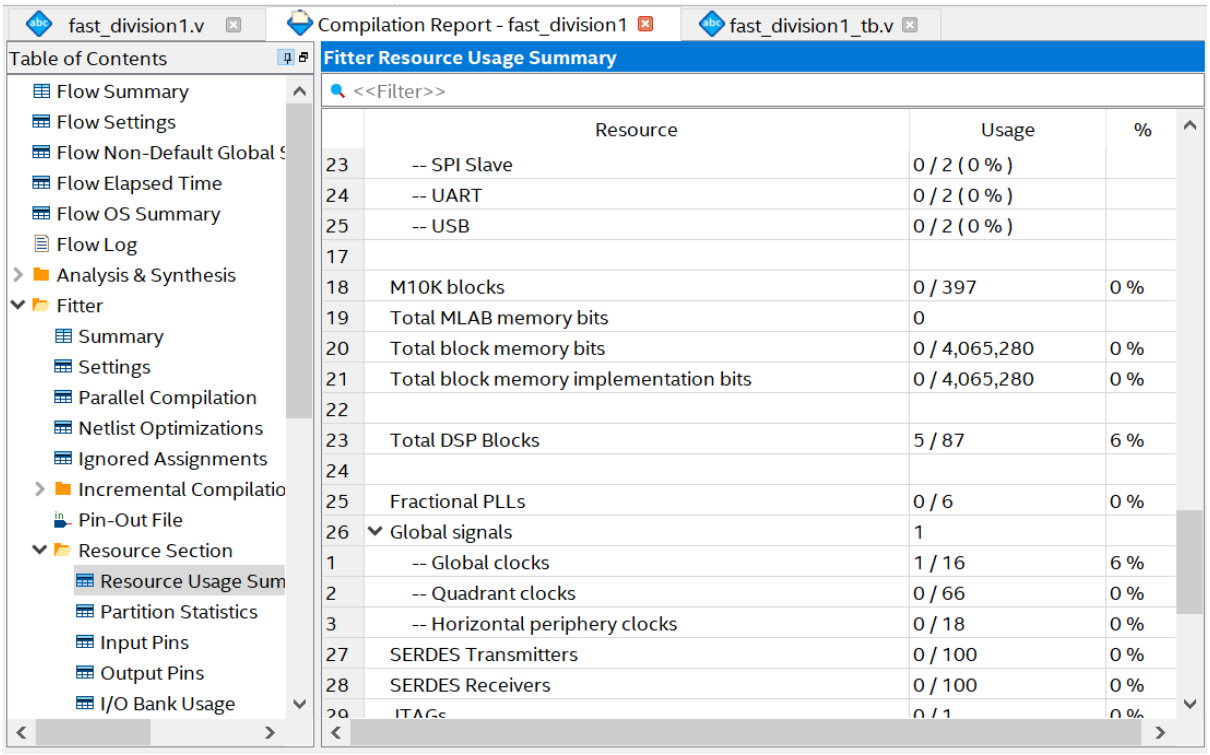
AREA REPORT


fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Summary
Settings
Parallel Compilation
Netlist Optimizations
Ignored Assignments
Incremental Compilatio
Pin-Out File
Resource Section
Resource Usage Sum
Partition Statistics
Input Pins
Output Pins
I/O Bank Usage
Fitter Summary
<<Filter>>
Fitter Status Successful - Fri Jul 14 15:53:12 2023
Quartus Prime Version 18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name fast_division1
Top-level Entity Name fast_division1
Family Cyclone V
Device 5CSEMA5F31C6
Timing Models Final
Logic utilization (in ALMs) 78 / 32,070 (< 1 %)
Total registers 30
Total pins 14 / 457 (3 %)
Total virtual pins 0
Total block memory bits 0 / 4,065,280 (0 %)
Total RAM Blocks 0 / 397 (0 %)
Total DSP Blocks 5 / 87 (6 %)
Total HSSI RX PCSs 0
Total HSSI PMA RX Deserializers 0
Total HSSI TX PCSs 0
Total HSSI PMA TX Serializers 0
Total PLLs 0 / 6 (0 %)
Total DLLs 0 / 4 (0 %)

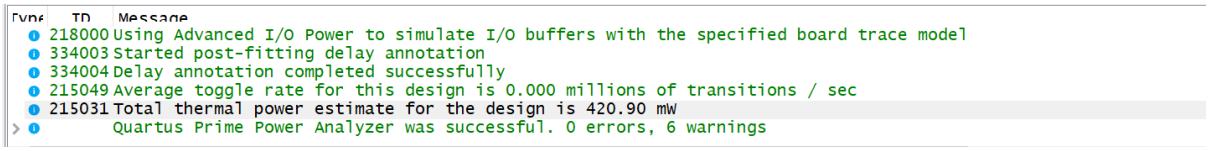

fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Summary
Settings
Parallel Compilation
Netlist Optimizations
Ignored Assignments
Incremental Compilatio
Pin-Out File
Resource Section
Resource Usage Sum
Partition Statistics
Input Pins
Output Pins
I/O Bank Usage
Fitter Resource Usage Summary
<<Filter>>
Resource Usage %
1 Logic utilization (ALMs needed / total ALMs on device) 78 / 32,070 < 1 %
2 ALMs needed [=A-B+C] 78
1 [A] ALMs used in final placement [=a+b+c+d] 80 / 32,070 < 1 %
1 [a] ALMs used for LUT logic and registers 10
2 [b] ALMs used for LUT logic 64
3 [c] ALMs used for registers 6
4 [d] ALMs used for memory (up to half of total ALMs) 0
2 [B] Estimate of ALMs recoverable by dense packing 2 / 32,070 < 1 %
3 [C] Estimate of ALMs unavailable [=a+b+c+d] 0 / 32,070 0 %
1 [a] Due to location constrained logic 0
2 [b] Due to LAB-wide signal conflicts 0
3 [c] Due to LAB input limits 0
4 [d] Due to virtual I/Os 0
3
4 Difficulty packing design Low
5
6 Total LABs: partially or completely used 13 / 3,207 < 1 %
1 -- Logic LABs 13


fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Summary
Settings
Parallel Compilation
Netlist Optimizations
Ignored Assignments
Incremental Compilatio
Pin-Out File
Resource Section
Resource Usage Sum
Partition Statistics
Input Pins
Output Pins
I/O Bank Usage
Fitter Resource Usage Summary
<<Filter>>
Resource
Usage
%
-- Logic LABs 13
-- Memory LABs (up to half of total LABs) 0
Combinational ALUT usage for logic 140
-- 7 input functions 0
-- 6 input functions 4
-- 5 input functions 13
-- 4 input functions 25
-- <=3 input functions 98
Combinational ALUT usage for route-throughs 11
Dedicated logic registers 30
-- By type:
-- Primary logic registers 30 / 64,140 < 1 %
-- Secondary logic registers 0 / 64,140 0 %
-- By function:
-- Design implementation registers 30
-- Routing optimization registers 0

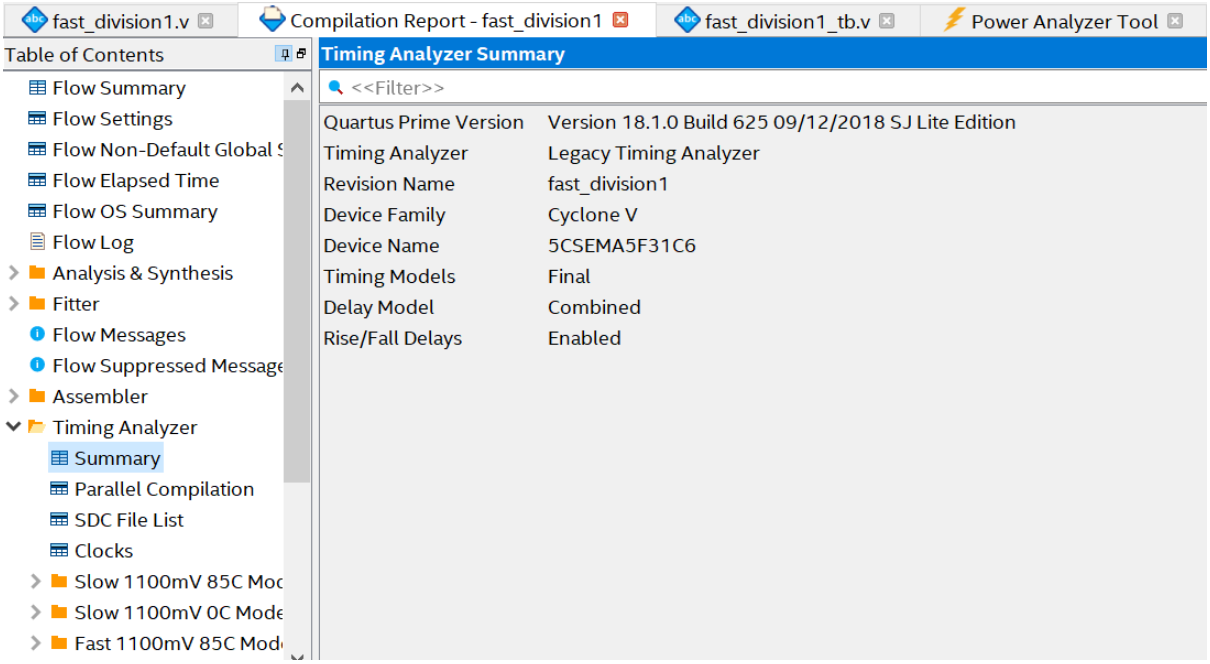

fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Summary
Settings
Parallel Compilation
Netlist Optimizations
Ignored Assignments
Incremental Compilatio
Pin-Out File
Resource Section
Resource Usage Sum
Partition Statistics
Input Pins
Output Pins
I/O Bank Usage
Fitter Resource Usage Summary
<<Filter>>
Resource
Usage
%
Virtual pins 0
I/O pins 14 / 457 3 %
-- Clock pins 1 / 8 13 %
-- Dedicated input pins 0 / 21 0 %
Hard processor system peripheral utilization
-- Boot from FPGA 0 / 1 (0 %)
-- Clock resets 0 / 1 (0 %)
-- Cross trigger 0 / 1 (0 %)
-- S2F AXI 0 / 1 (0 %)
-- F2S AXI 0 / 1 (0 %)
-- AXI Lightweight 0 / 1 (0 %)
-- SDRAM 0 / 1 (0 %)
-- Interrupts 0 / 1 (0 %)
-- JTAG 0 / 1 (0 %)
-- Loan I/O 0 / 1 (0 %)
-- MPU event standby 0 / 1 (0 %)

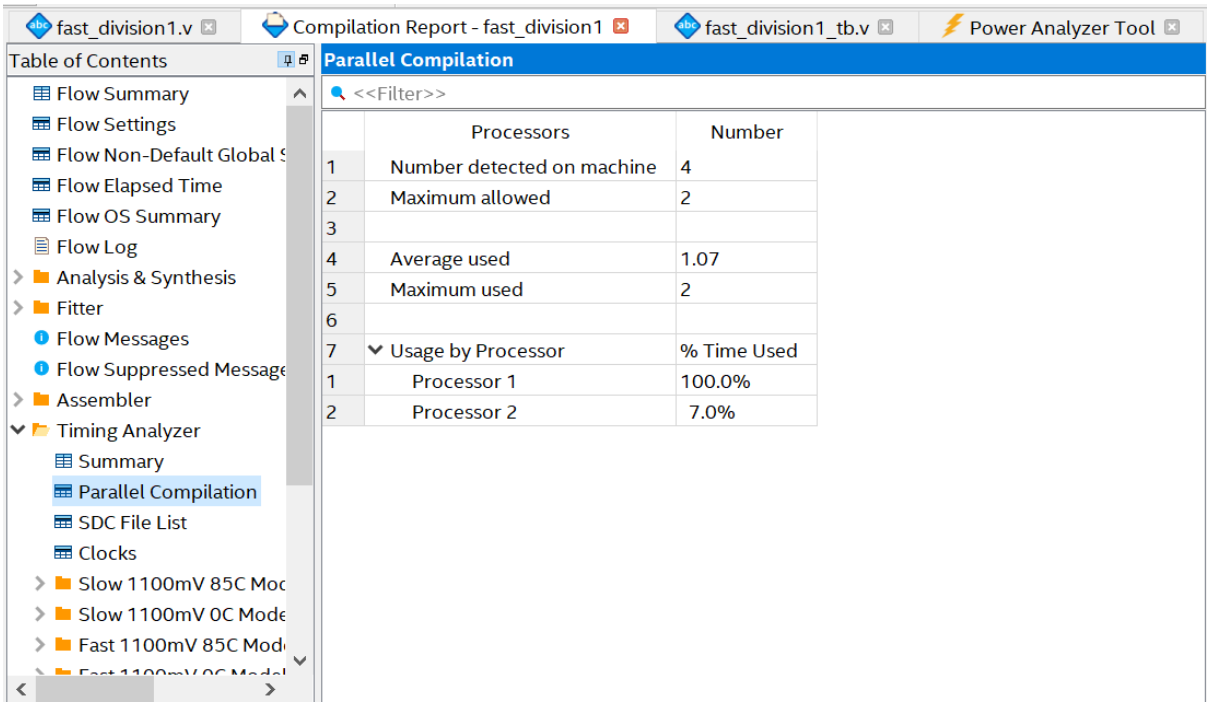


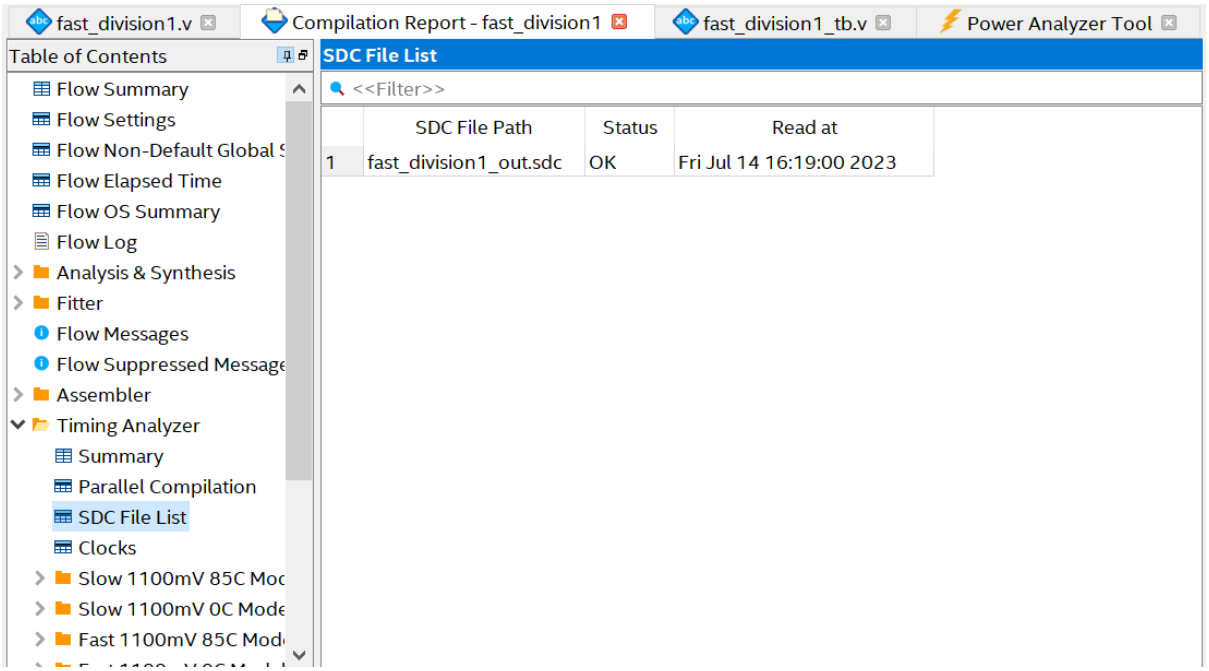
POWER REPORT

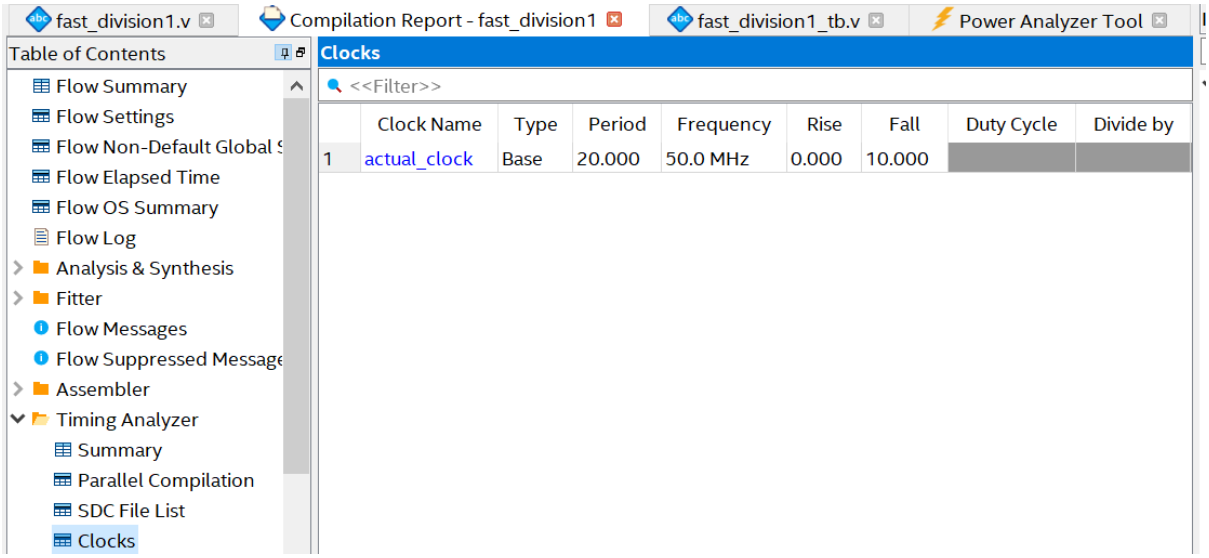


TIMING REPORT

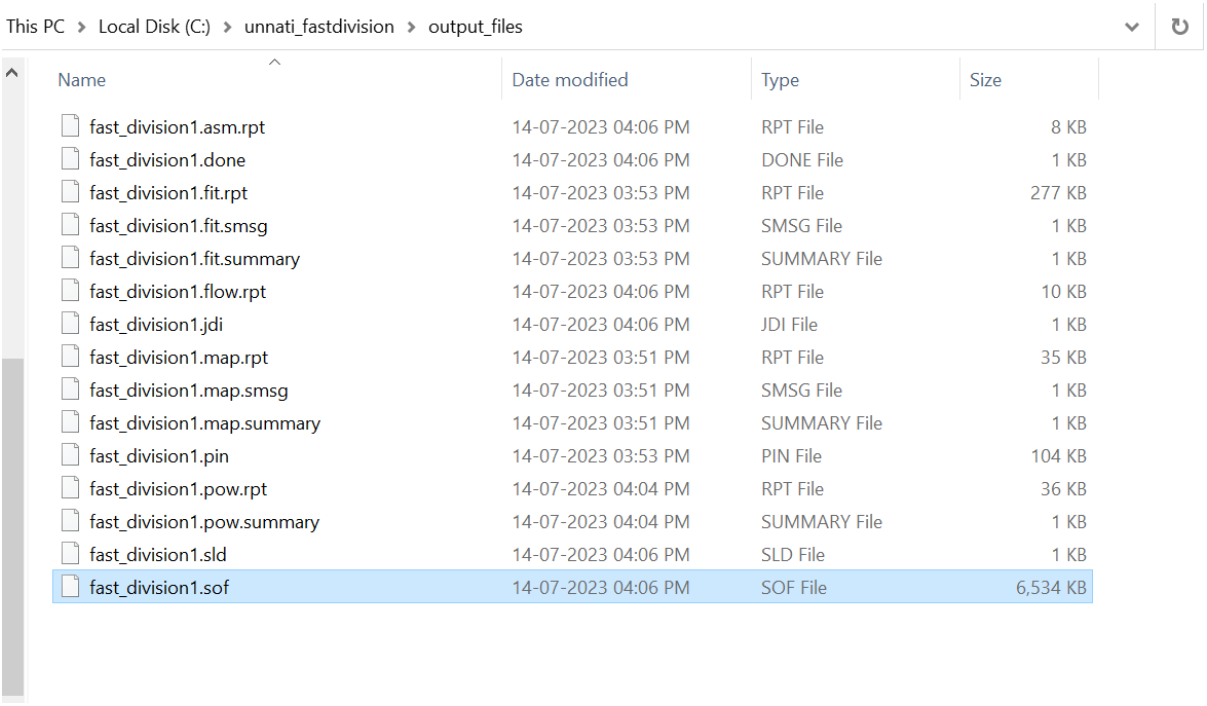



fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Power Analyzer Tool
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Flow Messages
Flow Suppressed Message
Assembler
Timing Analyzer
Summary
Parallel Compilation
SDC File List
Clocks
Slow 1100mV 85C Moc
Slow 1100mV 0C Mode
Fast 1100mV 85C Mod
Parallel Compilation
<<Filter>>
Processors
Number
1 Number detected on machine 4
2 Maximum allowed 2
3
4 Average used 1.07
5 Maximum used 2
6
7 Usage by Processor % Time Used
1 Processor 1 100.0%
2 Processor 2 7.0%


fast_division1.v
Compilation Report - fast_division1
fast_division1_tb.v
Power Analyzer Tool
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global S
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Flow Messages
Flow Suppressed Message
Assembler
Timing Analyzer
Summary
Parallel Compilation
SDC File List
Clocks
Slow 1100mV 85C Moc
Slow 1100mV 0C Mode
Fast 1100mV 85C Mod
SDC File List
<<Filter>>
SDC File Path
Status
Read at
1 fast_division1_out.sdc OK Fri Jul 14 16:19:00 2023

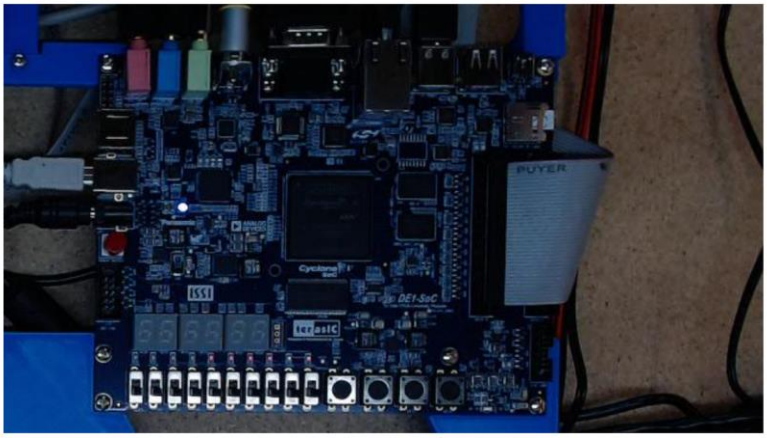


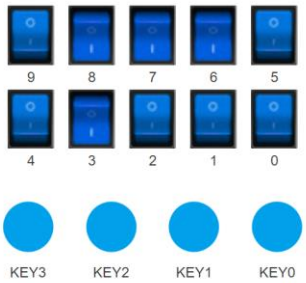
CREATION OF .sof FILE



IMPLEMENTATION ON INTEL FPGA LABORATORY

TEST CASE (1)





TEST CASE (2)