

# **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**

**MENTOR NAME**

**DR.S.ANUSOOYA**

# DESIGN & INPLEMENTATION OF SLOW AND FAST DIVISION ALGORITHM IN COMPUTER ARCHITECTURE

## ABOUT DIVISION ALGORITHM

A division algorithm is a method or procedure used to perform division operations on numerical values. Division is an arithmetic operation that involves dividing one value (the dividend) by another value (the divisor) to obtain a quotient and a remainder.

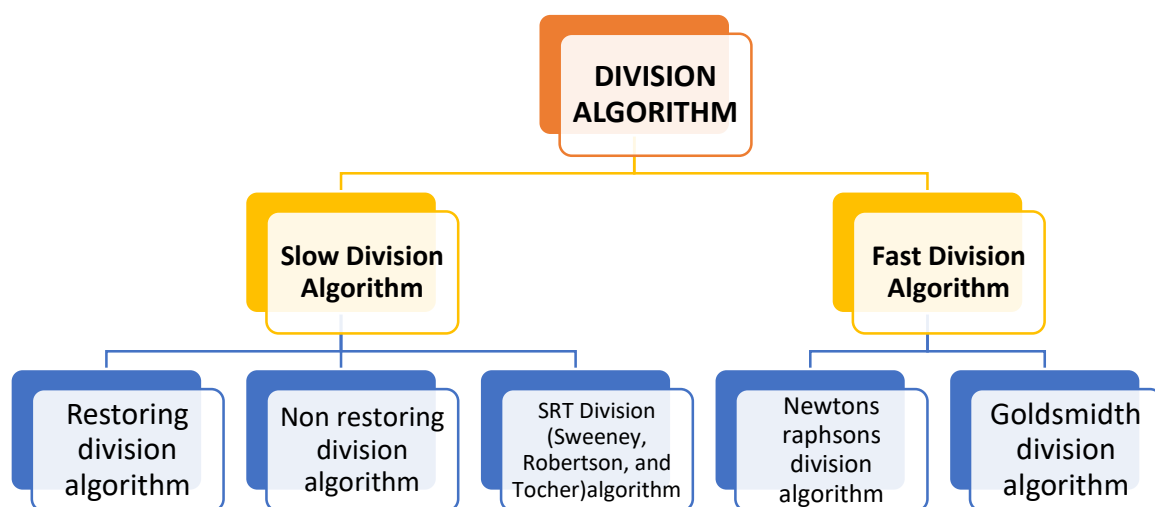
## DIVISION ALGORITHM IN COMPUTER ARCHITECTURE

Division algorithms are essential in computer architecture because they provide a way to perform division operations using the available hardware resources and arithmetic operations supported by the computer system.

The design of division algorithms involves considerations such as accuracy, efficiency, speed, and resource utilization. Different division algorithms have varying trade-offs in terms of these factors.

Division algorithm in computer architecture uses registers for storing the numbers and calculations.

## CLASSIFICATION OF DIVISION ALGORITHM



## SLOW DIVISION ALGORITHM

Slow division algorithms produce one digit of the final quotient per iteration. Examples of slow division include restoring, non-restoring, and SRT division.

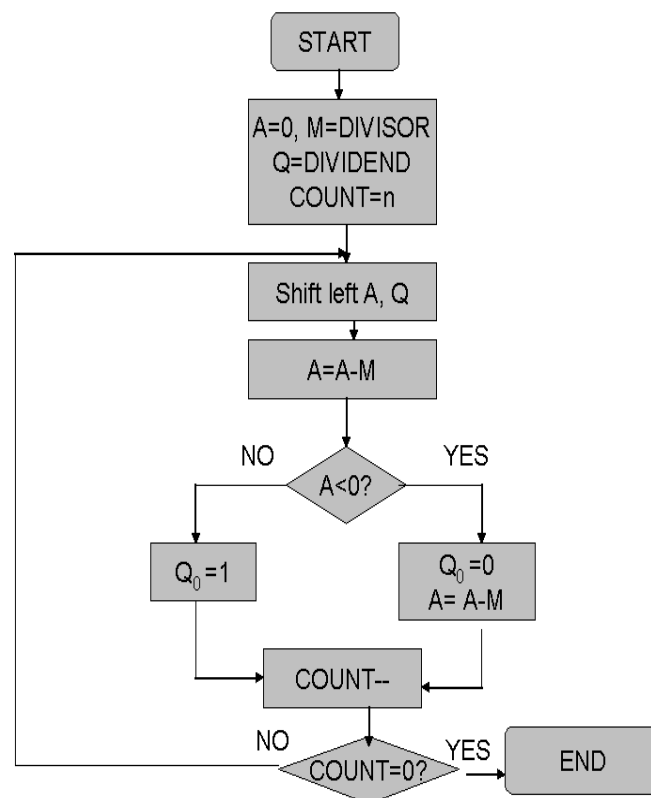
## RESTORING DIVISION ALGORITHM

This algorithm is based on repeated subtraction and shifting to find the quotient and remainder. It involves restoring the dividend in each iteration and comparing it with the divisor. Due to this process, it has been named as Restoring Division Algorithm.

This algorithm is performed for an unsigned integer. Below are the letters and its operations applied in this algorithm.

- Register A contains remainder which will be restored after each iteration and initially it is kept as zero.
- Register Q contains quotient, where n-bit dividend is stored.
- M stores n-bit divisor.

## FLOWCHART



## MANUAL CALCULATION

### TEST CASE (1)

Dividend (Nr)= 7 (place in Q) and Divisor(Dr) = 2 (place in M),count n=4

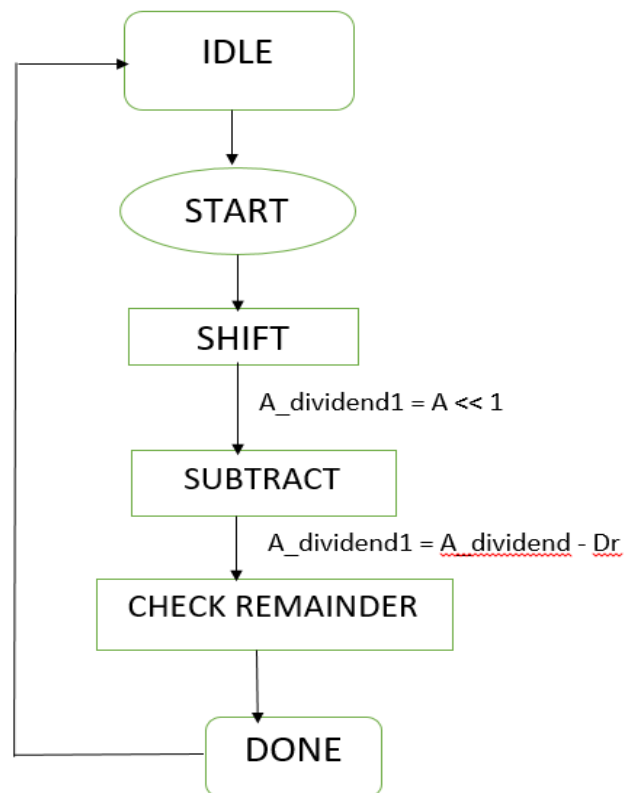
A	Q	OPERATION	COUNT
00000	0111	Initialize	4
00000	111_	Shift left A,Q	
11110	111_	A=A-M	
00000	1110	Q[0]<-0,restore A	
00001	110_	Shift left A,Q	3
11111	110_	A=A-M	
00001	1100	Q[0]<-0,restore A	
00011	100_	Shift left A,Q	2
00001	100_	A=A-M	
00001	1001	Q[0]<-1	
00011	001_	Shift left A,Q	1
00001	001_	A=A-M	
00001	0011	Q[0]<-1	
Final result : Quotient=0011 , Remainder=0001			0

### TEST CASE (2)

Dividend (Nr)= 14 (place in Q) and Divisor(Dr) = 5 (place in M),count n=4

A	Q	OPERATION	COUNT
00000	1110	Initialize	4
00001	110_	Shift left A,Q	
11100	110_	A=A-M	
00001	1100	Q[0]<-0,restore A	
00011	100_	Shift left A,Q	3
11110	100_	A=A-M	
00011	1000	Q[0]<-0,restore A	
00111	000_	Shift left A,Q	2
00010	000_	A=A-M	
00010	0001	Q[0]<-1	
00100	001_	Shift left A,Q	1
11111	001_	A=A-M	
00100	0010	Q[0]<-0,restore A	
Final result : Quotient=0010 , Remainder=0100			0

## MEALY STATE MACHINE



## VERILOG CODE LOGIC

### Input & Output Assumptions

Restoring Division algorithm for 4-bit operands in an FSM (Mealy State Machine) format with completely synthesizable Verilog Code.

### Inputs

clk, reset, Nr, Dr (2 operands to be multiplied), start (This pulse indicates start of computation)

### Outputs

Q (Quotient), R (Remainder) and done (This pulse indicates the availability of final answer)

1. In IDLE state, we wait for the start pulse. Upon its arrival, move to START state.

2. In START state, we follow the same algorithm and perform the computations using verilog logic as long as counter < 3. (Incrementing counter is used)
3. Upon counter completion, we send out done pulse and goto IDLE state.

## VERILOG CODE

```
module slow_division1(clk,reset,start,Nr,Dr,done,Q,R);
```

```
input clk;  
input reset;  
input start;  
input [3:0]Nr,Dr;  
output [3:0]Q,R;  
output done;
```

```
reg [7:0] A,next_A,A_dividend,A_dividend1;  
reg next_state, pres_state;  
reg [1:0] count,next_count;  
reg done, next_done;
```

```
parameter IDLE = 1'b0;  
parameter START = 1'b1;
```

```
assign R = A[7:4];  
assign Q = A[3:0];
```

```
always @ (posedge clk or negedge reset)  
begin  
if(!reset)  
begin  
A      <= 8'd0;  
done   <= 1'b0;  
pres_state <= 1'b0;  
count  <= 2'd0;  
end  
else
```

```

begin
  A      <= next_A;
  done   <= next_done;
  pres_state <= next_state;
  count  <= next_count;
end
end

```

```

always @ (*)
begin
  case(pres_state)
  IDLE:
  begin
    next_count = 2'b0;
    next_done = 1'b0;
    if(start)
    begin
      next_state = START;
      next_A     = {4'd0,Nr};
    end
  end
  else
  begin
    next_state = pres_state;
    next_A     = 8'd0;
  end
end
end

```

```

START:
begin
  next_count = count + 1'b1;
  A_dividend = A << 1;
  A_dividend1 = {A_dividend[7:4]-Dr,A_dividend[3:0]};
  next_A      = A_dividend1[7] ? {A_dividend[7:4],A_dividend[3:1],1'b0} :
                {A_dividend1[7:4],A_dividend[3:1],1'b1};
  next_done = (&count) ? 1'b1 : 1'b0;
  next_state = (&count) ? IDLE : pres_state;
end
endcase

```

```
end  
endmodule
```

## **TEST BENCH CODE**

```
module slow_division1_tb;  
  
    reg clk,reset,start;  
    reg [3:0]Nr,Dr;  
    wire [3:0]Q,R;  
    wire done;  
  
    always #5 clk = ~clk;  
  
    slow_division1 inst (clk,reset,start,Nr,Dr,done,Q,R  
    );  
  
    initial  
    $monitor($time,"Nr=%d, Dr=%d, done=%d, Q=%d, R=%d  
    ",Nr,Dr,done,Q,R);  
  
    initial  
    begin  
        Nr=7;Dr=2;clk=1'b1;reset=1'b0;start=1'b0;  
        #10 reset = 1'b1;  
        #10 start = 1'b1;  
        #10 start = 1'b0;  
        @done  
        #10 Nr=14;Dr=5;start = 1'b1;  
        #10 start = 1'b0;  
    end  
endmodule
```



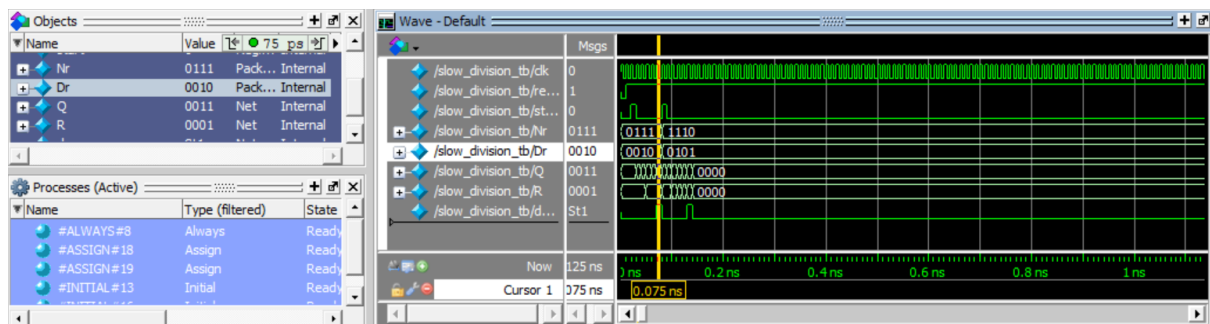
## SIMULATION RESULT

```

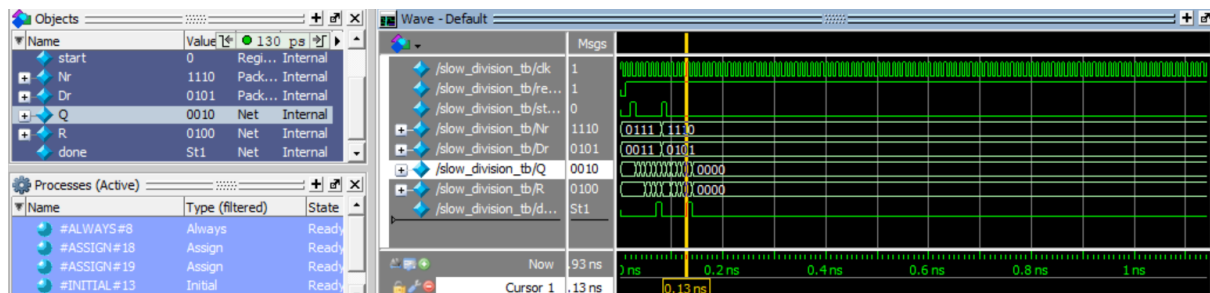
0Nr= 7, Dr= 2, done=0, Q= 0, R= 0
30Nr= 7, Dr= 2, done=0, Q= 7, R= 0
40Nr= 7, Dr= 2, done=0, Q=14, R= 0
50Nr= 7, Dr= 2, done=0, Q=12, R= 1
60Nr= 7, Dr= 2, done=0, Q= 9, R= 1
70Nr= 7, Dr= 2, done=1, Q= 3, R= 1
80Nr=14, Dr= 5, done=0, Q= 0, R= 0
90Nr=14, Dr= 5, done=0, Q=14, R= 0
100Nr=14, Dr= 5, done=0, Q=12, R= 1
110Nr=14, Dr= 5, done=0, Q= 8, R= 3
120Nr=14, Dr= 5, done=0, Q= 1, R= 2
130Nr=14, Dr= 5, done=1, Q= 2, R= 4
140Nr=14, Dr= 5, done=0, Q= 0, R= 0

```

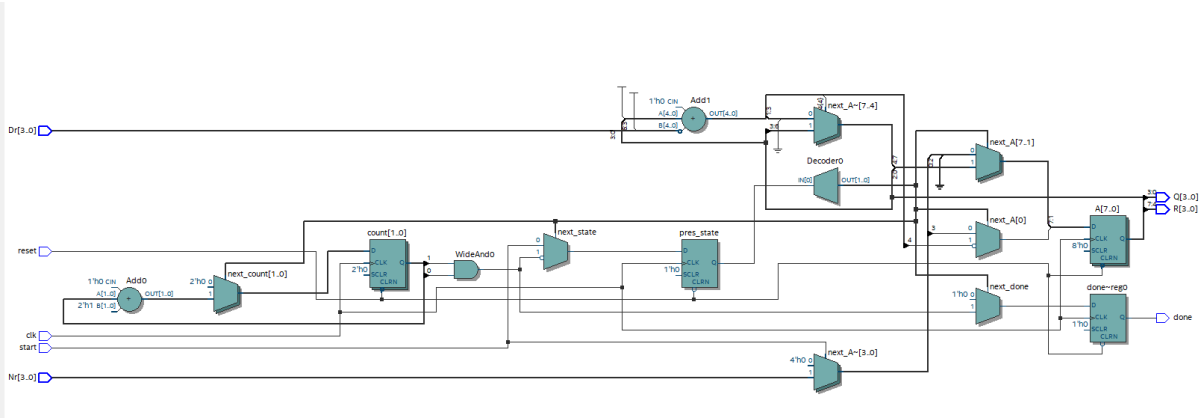
## RESULT OF TEST CASE (1)



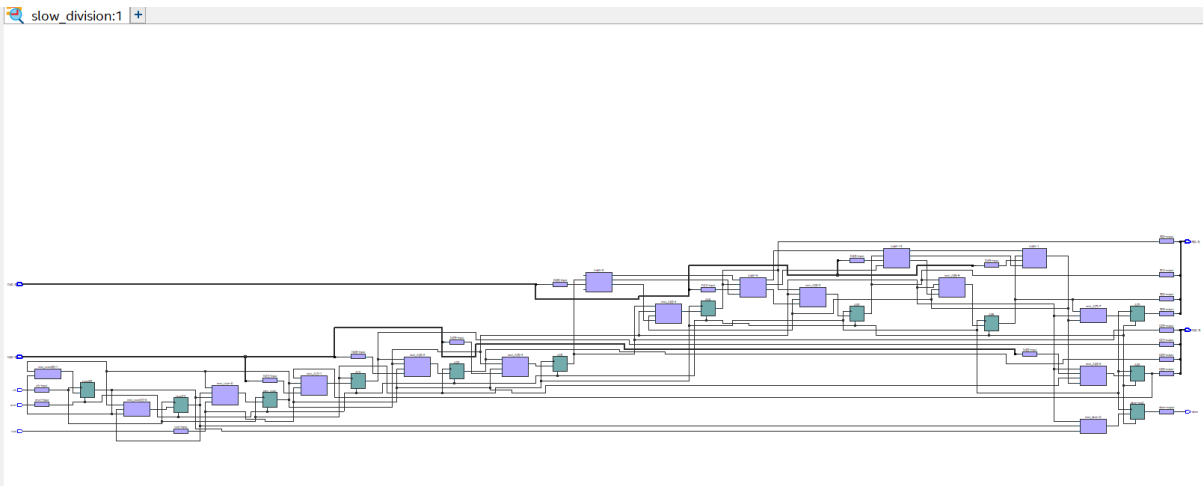
## RESULT OF TEST CASE (2)



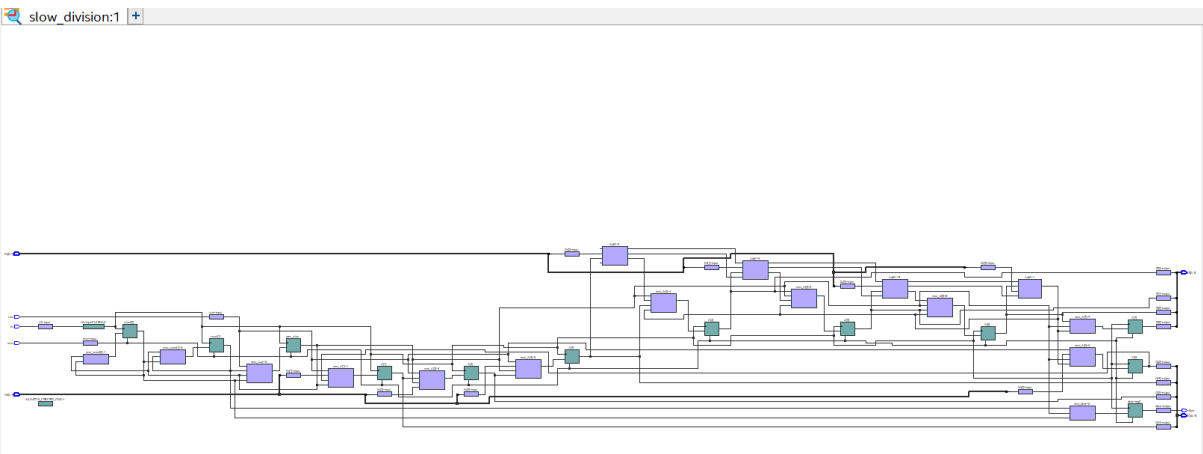
# RTL VIEWER



# TECHNOLOGY MAP VIEWER (POST MAPPING)



# TECHNOLOGY MAP VIEWER (POST FITTING)



# PIN PLANNER

Pin Planner - C:/slow\_div/slow\_division1 - slow\_division1  
File Edit View Processing Tools Window Help

Report

Report not available

Groups

Report

Tasks

Early Pin Planning

Early Pin Planning

Run I/O Assignm

Export Pin Assign

Pin Finder...

Highlight Pins

I/O Banks

VREF Groups

Top View - Wire Bond

Cyclone V - 5CSEMA5F31C6

Node Name	Direction	Location	I/O Bank	/REF Group	ttter Locatic	'O Standar	Reserved	rent Stren	Slew Rate	fferential P	analog Sett	3/VCCT_Gx	r I/O Pin T
Dr[3]	Input	PIN_AF9	3A	B3A_NO	PIN_AF9	3.3-...VTTL		16mA...ult)					
Dr[2]	Input	PIN_AF10	3A	B3A_NO	PIN_AF10	3.3-...VTTL		16mA...ult)					
Dr[1]	Input	PIN_AD11	3A	B3A_NO	PIN_AD11	3.3-...VTTL		16mA...ult)					
Dr[0]	Input	PIN_AD12	3A	B3A_NO	PIN_AD12	3.3-...VTTL		16mA...ult)					
Nr[3]	Input	PIN_AE11	3A	B3A_NO	PIN_AE11	3.3-...VTTL		16mA...ult)					
Nr[2]	Input	PIN_AC9	3A	B3A_NO	PIN_AC9	3.3-...VTTL		16mA...ult)					
Nr[1]	Input	PIN_AD10	3A	B3A_NO	PIN_AD10	3.3-...VTTL		16mA...ult)					
Nr[0]	Input	PIN_AE12	3A	B3A_NO	PIN_AE12	3.3-...VTTL		16mA...ult)					
Q[3]	Output	PIN_Y19	4A	B4A_NO	PIN_Y19	3.3-...VTTL		16mA...ult)	1 (default)				
Q[2]	Output	PIN_W20	5A	B5A_NO	PIN_W20	3.3-...VTTL		16mA...ult)	1 (default)				
Q[1]	Output	PIN_W21	5A	B5A_NO	PIN_W21	3.3-...VTTL		16mA...ult)	1 (default)				
Q[0]	Output	PIN_Y21	5A	B5A_NO	PIN_Y21	3.3-...VTTL		16mA...ult)	1 (default)				
R[3]	Output	PIN_V17	4A	B4A_NO	PIN_V17	3.3-...VTTL		16mA...ult)	1 (default)				
R[2]	Output	PIN_V18	4A	B4A_NO	PIN_V18	3.3-...VTTL		16mA...ult)	1 (default)				
R[1]	Output	PIN_W17	4A	B4A_NO	PIN_W17	3.3-...VTTL		16mA...ult)	1 (default)				
R[0]	Output	PIN_W19	4A	B4A_NO	PIN_W19	3.3-...VTTL		16mA...ult)	1 (default)				
clk	Input	PIN_AF14	3B	B3B_NO	PIN_AF14	3.3-...VTTL		16mA...ult)					
done	Output	PIN_V16	4A	B4A_NO	PIN_V16	3.3-...VTTL		16mA...ult)	1 (default)				
reset	Input	PIN_AA14	3B	B3B_NO	PIN_AA14	3.3-...VTTL		16mA...ult)					
start	Input	PIN_AA15	3B	B3B_NO	PIN_AA15	3.3-...VTTL		16mA...ult)					

# CHIP PLANNER

Chip Planner - C:/slow\_div/slow\_division1 - slow\_division1  
File Edit View Tools Window Help

Report

Report not available

Tasks

Generate Clr

Toggle Back

Report Resou

Report Comp

Mark Selectic

Core Reports

Report Hig

Report Roi

Rapid Recom

Show Cha

Show Cha

Show Cha

Clock Report

Coordinate:

Editing Mode: ECO - 5CSEMA5F31C6

Layers Settings

Basic

Background

None

Block Utilization

Design Partition Planner

Logic Lock Regions

User-assigned Logic Lock Re...

Fitter-placed Logic Lock Regi...

Clock Regions

Overlay Objects

Connection Lines

Layers Settings

Color Legend

Device Properties

Selected elements:

General

Device

5CSEMA5F31C6

Family

Cyclone V

Device Extents

90 x 82

ALMs

32070

Total I/Os

457

GPIOs

457

GXB Channel PMA

0

GXB Channel PCS

0

# AREA REPORT

slow\_division1.v x slow\_division1\_tb.v x Compilation Report - slow\_division1 x

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 Compilation
  - Pin-Out File
  - Resource Section
  - I/O Rules Section
  - Device Options
  - Operating Settings and
  - Estimated Delay Added
  - Messages

**Fitter Summary**

<<Filter>>

Fitter Status	Successful - Mon Jul 10 19:32:51 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	slow_division1
Top-level Entity Name	slow_division1
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	9 / 32,070 (< 1 %)
Total registers	14
Total pins	20 / 457 (4 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total RAM Blocks	0 / 397 (0 %)
Total DSP Blocks	0 / 87 (0 %)
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 %)

v\_division1.v x slow\_division1\_tb.v x Compilation Report - slow\_division1 x Power Analyzer Tool x

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Glob
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
  - Summary
  - Settings
  - Parallel Compilation
  - Netlist Optimization:
  - Ignored Assignment
  - Incremental Compil
  - Pin-Out File
  - Resource Section
    - Resource Usage S
    - Partition Statistics

**Fitter Resource Usage Summary**

<<Filter>>

	Resource	Usage	%
1	Logic utilization (ALMs needed / total ALMs on device)	9 / 32,070	< 1 %
2	ALMs needed [=A-B+C]	9	
1	[A] ALMs used in final placement [=a+b+c+d]	9 / 32,070	< 1 %
1	[a] ALMs used for LUT logic and registers	6	
2	[b] ALMs used for LUT logic	3	
3	[c] ALMs used for registers	0	
4	[d] ALMs used for memory (up to half of total ALMs)	0	
2	[B] Estimate of ALMs recoverable by dense packing	0 / 32,070	0 %
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	2 / 3,207	< 1 %

Flow Non-Default Glob	1	-- Logic LABs	2		
Flow Elapsed Time	2	-- Memory LABs (up to half of total LABs)	0		
Flow OS Summary	7				
Flow Log	8	▼ Combinational ALUT usage for logic	17		
Analysis & Synthesis	1	-- 7 input functions	0		
Fitter	2	-- 6 input functions	0		
Summary	3	-- 5 input functions	0		
Settings	4	-- 4 input functions	8		
Parallel Compilation	5	-- <=3 input functions	9		
Netlist Optimization	9	Combinational ALUT usage for route-throughs	0		
Ignored Assignment	10				
Incremental Compil	11	▼ Dedicated logic registers	14		
Pin-Out File	1	▼ -- By type:			
Resource Section	1	-- Primary logic registers	12 / 64,140	< 1 %	
Resource Usage S	2	-- Secondary logic registers	2 / 64,140	< 1 %	
Partition Statistics					
Flow Non-Default Glob	2	▼ -- By function:			
Flow Elapsed Time	1	-- Design implementation registers	12		
Flow OS Summary	2	-- Routing optimization registers	2		
Flow Log	12				
Analysis & Synthesis	13	Virtual pins	0		
Fitter	14	▼ I/O pins	20 / 457	4 %	
Summary	1	-- Clock pins	0 / 8	0 %	
Settings	2	-- Dedicated input pins	0 / 21	0 %	
Parallel Compilation	15				
Flow Non-Default Glob	18	M10K blocks	0 / 397	0 %	
Flow Elapsed Time	19	Total MLAB memory bits	0		
Flow OS Summary	20	Total block memory bits	0 / 4,065,280	0 %	
Flow Log	21	Total block memory implementation bits	0 / 4,065,280	0 %	
Analysis & Synthesis	22				
Fitter	23	Total DSP Blocks	0 / 87	0 %	
Summary	24				

## POWER REPORT

Time	ID	Message
334003		Started post-fitting delay annotation
334004		Delay annotation completed successfully
215049		Average toggle rate for this design is 0.000 millions of transitions / sec
215031		Total thermal power estimate for the design is 421.03 mw
		Quartus Prime Power Analyzer was successful. 0 errors, 6 warnings

# TIMING REPORT

slow\_division1.v x slow\_division1\_tb.v x Compilation Report - slow\_division1 x

Table of Contents

- Fitter
  - Flow Messages
  - Flow Suppressed Message
- Assembler
- Timing Analyzer
  - Summary
  - Parallel Compilation
  - SDC File List
  - Clocks
  - Slow 1100mV 85C Model
    - Fmax Summary
    - Timing Closure Reco
    - Setup Summary
    - Hold Summary
    - Recovery Summary
    - Removal Summary
    - Minimum Pulse Width
    - Metastability Summa
  - Slow 1100mV 0C Model
  - Fast 1100mV 85C Model
  - Fast 1100mV 0C Model

Timing Analyzer Summary

<<Filter>>

Quartus Prime Version	Version 18.1.0 Build 625 09/12/2018 SJ Lite Edition
Timing Analyzer	Legacy Timing Analyzer
Revision Name	slow_division1
Device Family	Cyclone V
Device Name	5CSEMA5F31C6
Timing Models	Final
Delay Model	Combined
Rise/Fall Delays	Enabled

slow\_division1.v x slow\_division1\_tb.v x Compilation Report - slow\_division1 x

Table of Contents

- Fitter
  - Flow Messages
  - Flow Suppressed Message
- Assembler
- Timing Analyzer
  - Summary
  - Parallel Compilation
  - SDC File List
  - Clocks
  - Slow 1100mV 85C Model
    - Fmax Summary
    - Timing Closure Reco
    - Setup Summary
    - Hold Summary
    - Recovery Summary
    - Removal Summary
    - Minimum Pulse Width
    - Metastability Summa
  - Slow 1100mV 0C Model
  - Fast 1100mV 85C Model
  - Fast 1100mV 0C Model

Parallel Compilation

<<Filter>>

	Processors	Number
1	Number detected on machine	4
2	Maximum allowed	2
3		
4	Average used	1.01
5	Maximum used	2
6		
7	Usage by Processor	% Time Used
1	Processor 1	100.0%
2	Processor 2	0.8%

slow\_division1.v slow\_division1\_tb.v Compilation Report - slow\_division1

Table of Contents SDC File List

> Fitter

- Flow Messages
- Flow Suppressed Message

> Assembler

▼ Timing Analyzer

- Summary
- Parallel Compilation
- SDC File List
- Clocks

▼ Slow 1100mV 85C Moc

	SDC File Path	Status	Read at
1	slow_division1.out.sdc	OK	Mon Jul 10 22:33:30 2023

slow\_division1.v slow\_division1\_tb.v Compilation Report - slow\_division1

Table of Contents Clocks

> Fitter

- Flow Messages
- Flow Suppressed Message

> Assembler

▼ Timing Analyzer

- Summary
- Parallel Compilation
- SDC File List
- Clocks

▼ Slow 1100mV 85C Moc

	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle	Divide by
1	actual_clock	Base	20.000	50.0 MHz	0.000	10.000		

## CREATION OF .sof FILE

This PC > Local Disk (C:) > slow\_div > output\_files

Name	Date modified	Type	Size
slow_division1.asm.rpt	10-07-2023 10:10 AM	RPT File	8 KB
slow_division1.done	10-07-2023 10:10 AM	DONE File	1 KB
slow_division1.fit.rpt	10-07-2023 09:36 AM	RPT File	262 KB
slow_division1.fit.smsg	10-07-2023 09:36 AM	SMSG File	1 KB
slow_division1.fit.summary	10-07-2023 09:36 AM	SUMMARY File	1 KB
slow_division1.flow.rpt	10-07-2023 10:10 AM	RPT File	10 KB
slow_division1.jdi	10-07-2023 10:10 AM	JDI File	1 KB
slow_division1.map.rpt	10-07-2023 09:35 AM	RPT File	26 KB
slow_division1.map.smsg	10-07-2023 09:35 AM	SMSG File	1 KB
slow_division1.map.summary	10-07-2023 09:35 AM	SUMMARY File	1 KB
slow_division1.pin	10-07-2023 09:36 AM	PIN File	104 KB
slow_division1.pow.rpt	10-07-2023 09:54 AM	RPT File	34 KB
slow_division1.pow.summary	10-07-2023 09:54 AM	SUMMARY File	1 KB
slow_division1.sld	10-07-2023 10:10 AM	SLD File	1 KB
slow_division1.sof	10-07-2023 10:10 AM	SOF File	6,534 KB