EE – 271 PROJECT

SCIENTIFIC CALCULATON

Using FPGA DE-10 Lite board

SOUMYA SAHU : 014537666

SARIKA SATISH NATU : 014529138

YAMINI SANTOSH AWASTHI : 014560468

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

CONTENT

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

ABSTRACT:

Calculators are one of the most fundamental products used on an everyday basis by a variety of people. Modern calculators are designed to perform complicated calculations such as complex integrals, permutations etc. before the blink of an eye. Although, these are instruments we regularly use, rarely do we consider about various processing and activities that goes on under the plastic shell. Here, we have designed a 8 digit calculator on a FPGA DE10-Lite (Altera Max 10 DA) development board and a 1602A LCD display with serial interfacing using I2C protocol. This design is written in Verilog HDL language and can execute addition, subtraction, multiplication, division, modulus, power of 2 and factorial. It is designed using the concept of Finite state machine (FSM) which helps in design optimization and an efficient use of the hardware resources.

INTRODUCTION:

A significant part of understanding the functionality and purpose of any kind of design is knowing how it came to be. By creating a simple design for a calculator that can conduct basic arithmetic operations, we gained a better understanding of how data is transferred from a single button input to a computed value on a display. Our basic calculator was no longer as simple as pressing a few buttons and waiting, but instead became a series of translating information through different data types, choosing what outputs go to the display in real time, and understanding the critical time delays necessary for the components to interface with each other smoothly. New challenges arose while designing our datapaths and general logic, constantly diverting and questioning our approach to the problem. Fortunately, as all problems start, so do their solutions.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Problem statement:

The goal of this project was to design a calculator that can executed various calculator functions on up to 4-digit numbers for a maximum 8-digit output. This calculator will be able to perform the given tasks in a standard amount of time, but will be
restricted to whole numbers (i.e. the user cannot use decimal points). Although there are strict timing requirements, especially for the LCD display, we aim to make this calculator as responsive as possible.

Features:

1. It can perform the following operations:
   Custom:
   a. addition
   b. subtraction
   c. multiplication
   d. division

   Scientific:

   a. Modulus
   b. Power of 2
   c. Factorial

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Specifications:

- FPGA DE10-Lite (Altera Max 10 DA) running on 50 Mhz clock cycle.
- 16x2 LCD display
- Arduino Uno has been used to connect the FPGA to the LCD display.
- One reset button (FPGA button 0) to reset the vending machine.

The final product consists of a 1602A LCD display with serial interface, an Arduino Uno board and a De10 lite FPGA board with an altera Max 10DA processor.

The calculator takes in 4-digit inputs from the switches available on the FPGA board that display on the LCD. These inputs are taken as single binary value, which goes through a series of modules that perform the user's desired arithmetic operation. The output would be loaded onto a register that sends its value in real time to the Arduino Uno board where it is converted to the corresponding ASCII value which is then further display on the LCD screen. This is required as the arithmetic logic was coded in binary and the LCD only takes in information and instructions in ASCII. The display would show the final value along with the two operands and the cycle repeats when a new key is pressed. There is also a reset button included for a fresh start. Key 0 of the FPGA board has been used as the RESET button and KEY 1 has been used as KEY FLAG button which needs to be pressed after each input in order for the FSM to change its state.

As the final product is now, we have been able to testbench the modules that involve data intake from the user as well as the main module that handles the arithmetic operations. The input values show on the display along with the final output. The display clears when the clear button was pressed, and the user could continue to input numbers. The Seven segment displays the Mode of the calculator i.e. either custom or scientific.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

WORKING:

A Multiplexer has been used to design the various functionalities of the calculator. Depending on the calculator mode i.e. custom or scientific, the operation to be carried out is determined.

A Finite state machine has been implemented to design the behavior of the calculator in order to carry out the processing from taking in the input from the user to calculating and giving the respective output.

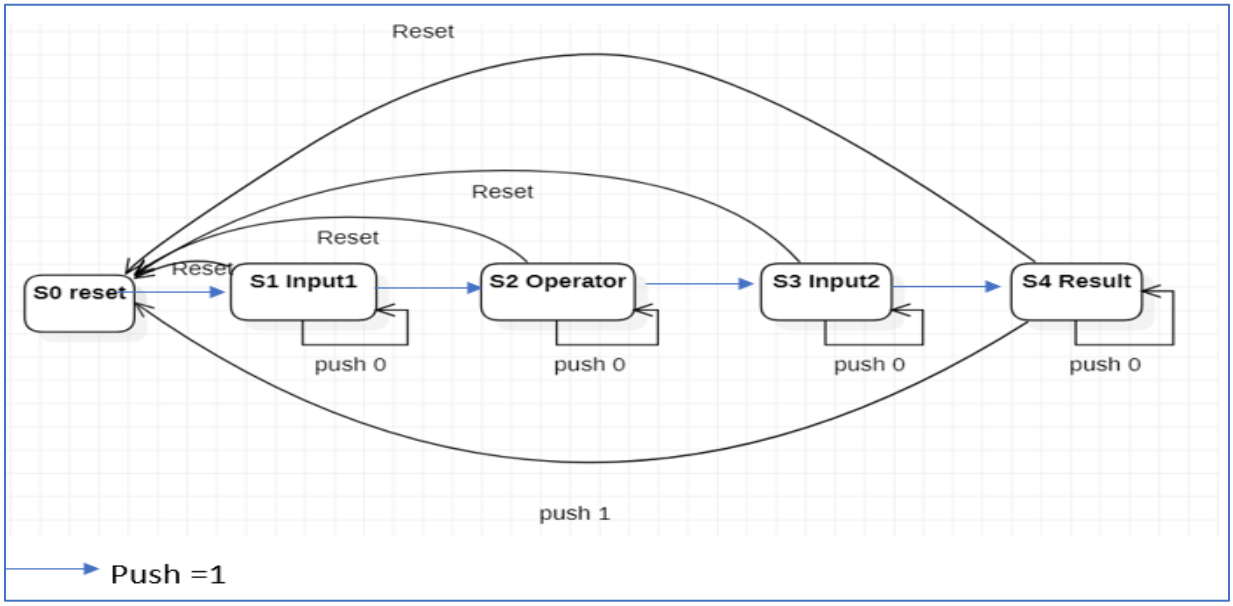


Figure 1: Finite State Machine

The various FSM states are as follows:

S0: This is the first state of the FSM. Here both the operands and the operator are initialized to 0.

Note: By default, the operation of the calculator, in case no operator is selected, is addition.

S1: the first input is assigned to operand1. Correspondingly, it is displayed on the LCD screen.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

S2: The second input is assigned to the operator variable. Here itself, based on the mode of the calculator, either a custom combination or a scientific combination is assigned to the operator.

S3: The third input is assigned to operand2. Correspondingly, it is displayed on the LCD.

S4: The result of the calculation is displayed in this state.

Operating Instructions

The user can simply flip the “ON” switch on the FPGA to get started, enter in the first operand, then select the desired operation in between and then select the second operand. The important thing to remember is that after each entry the Key flag switch needs to be pushed. The user must press the flag key for the display to show the operands and the calculated value and then proceed to select the next operation and input value. Key 0 is used as the is a reset button, so that the display and memory will clear, and the user may begin anew. Figure 1 below shows a simple, annotated picture of what the actual setup looks like.

- FPGA board is connected to the Arduino UNO board which then connects it to the LCD display (Please refer Figure 2).
- switch 0 to 3 are used to provide the 4 bits input
- switch 8 and 9 are used to provide the desired operator
- switch 4 is to change the calculator mode to either scientific or custom, default being custom. The mode of the calculator is visible on the 7-segment display.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

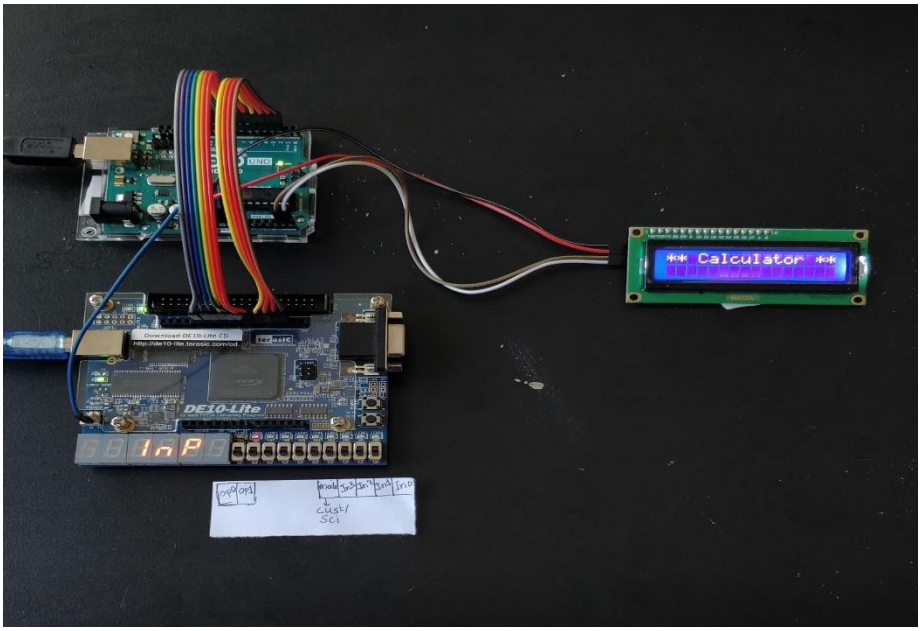


Figure 2: Actual Setup

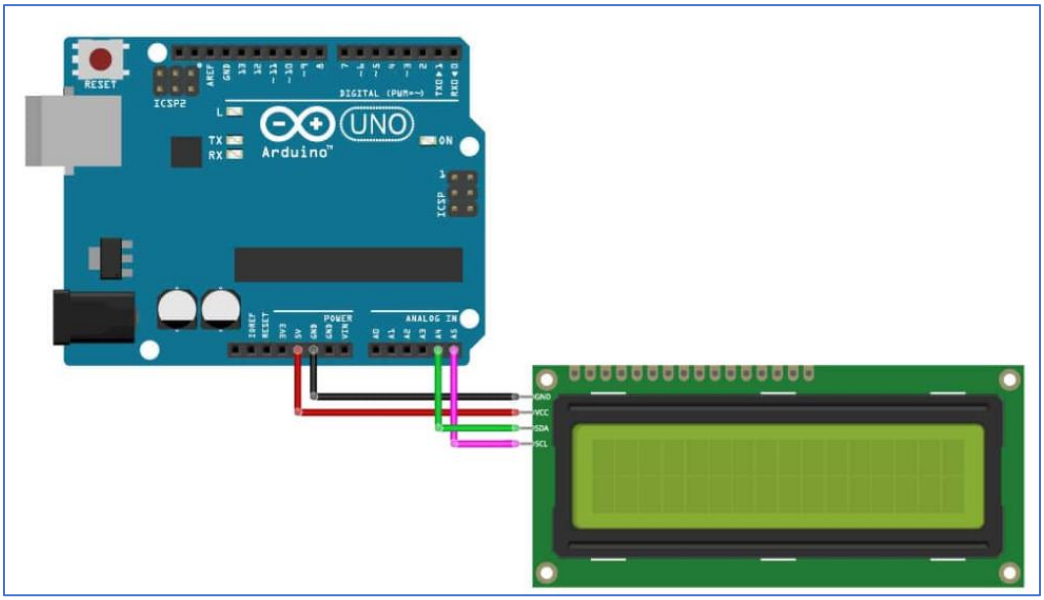


Figure 3: Arduino UNO to LCD Display interface

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Scheme of Operation

Referring to Figure 3 and Figure 4, this is what we have implemented in our final product to embody.

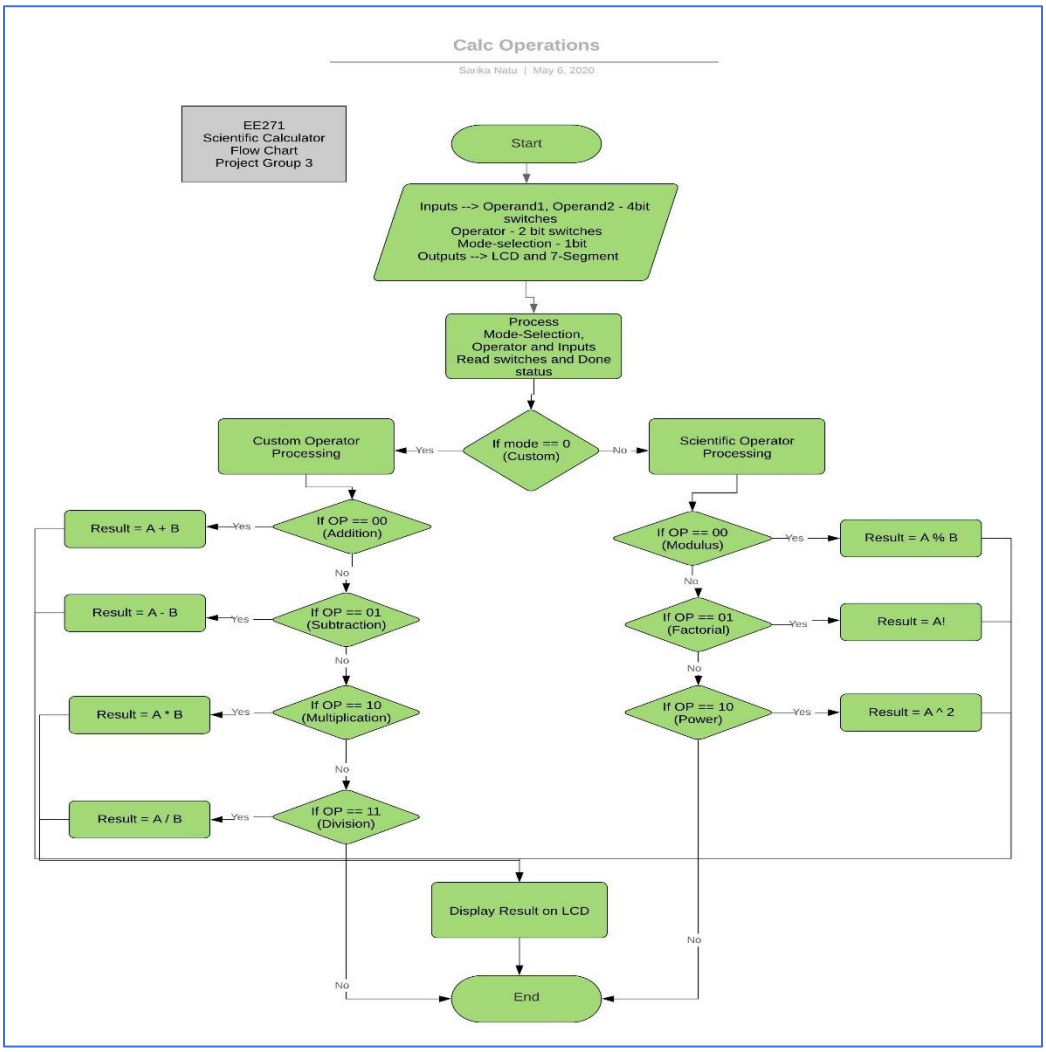


Figure 3: Operation Schematic

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

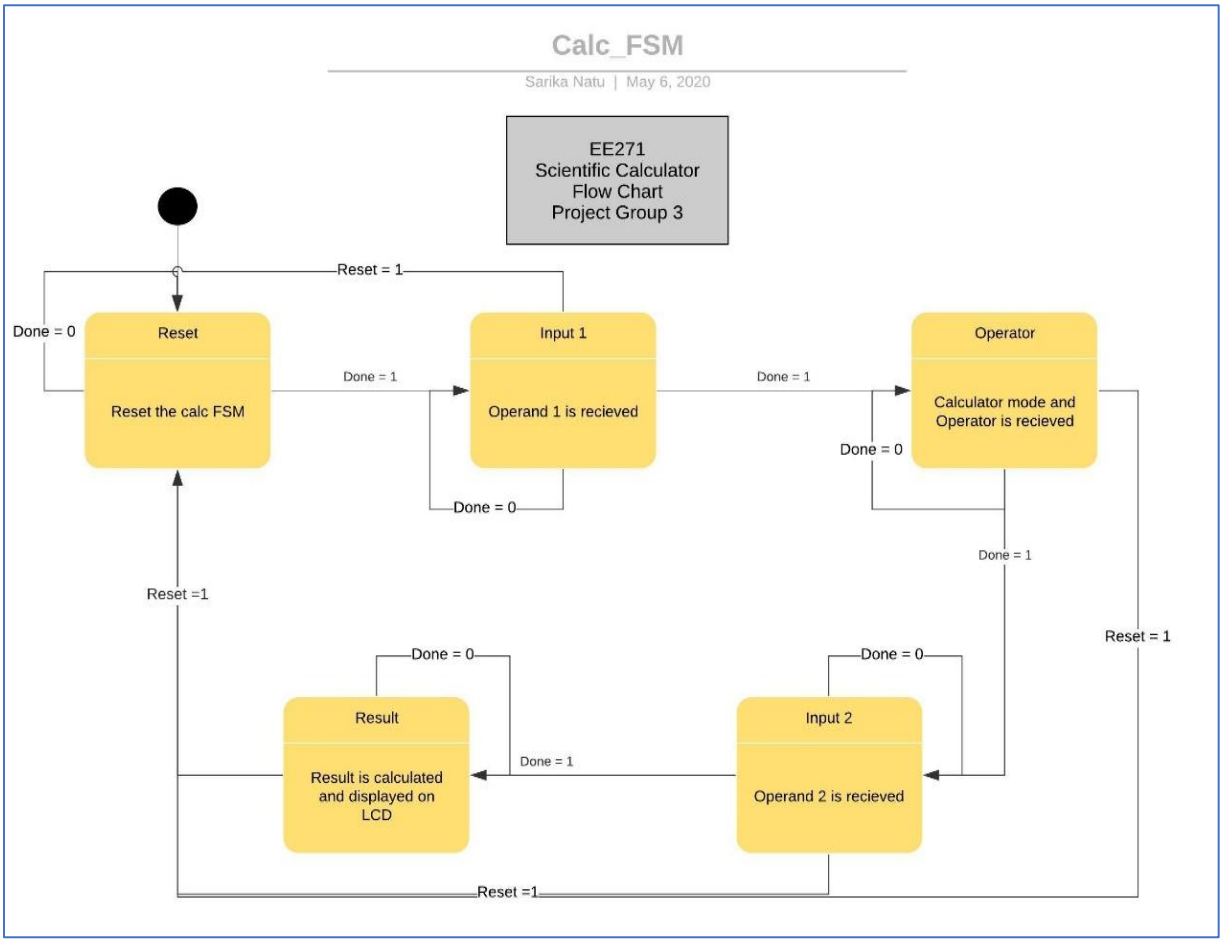


Figure 4: Schematic Flow

DESIGN FILES

The design consists of 4 modules along with a header module:

- switch_calc_top.v : header file
- input_calc.v : FSM for the behavorial flow of the design
- calc_fsm.v : for determining the functionality of the calculator
- clock_divider.v : For bringing down the frequency from 50 Mhz to 1 Hz.
- segment_display.v

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

i)Top_Module

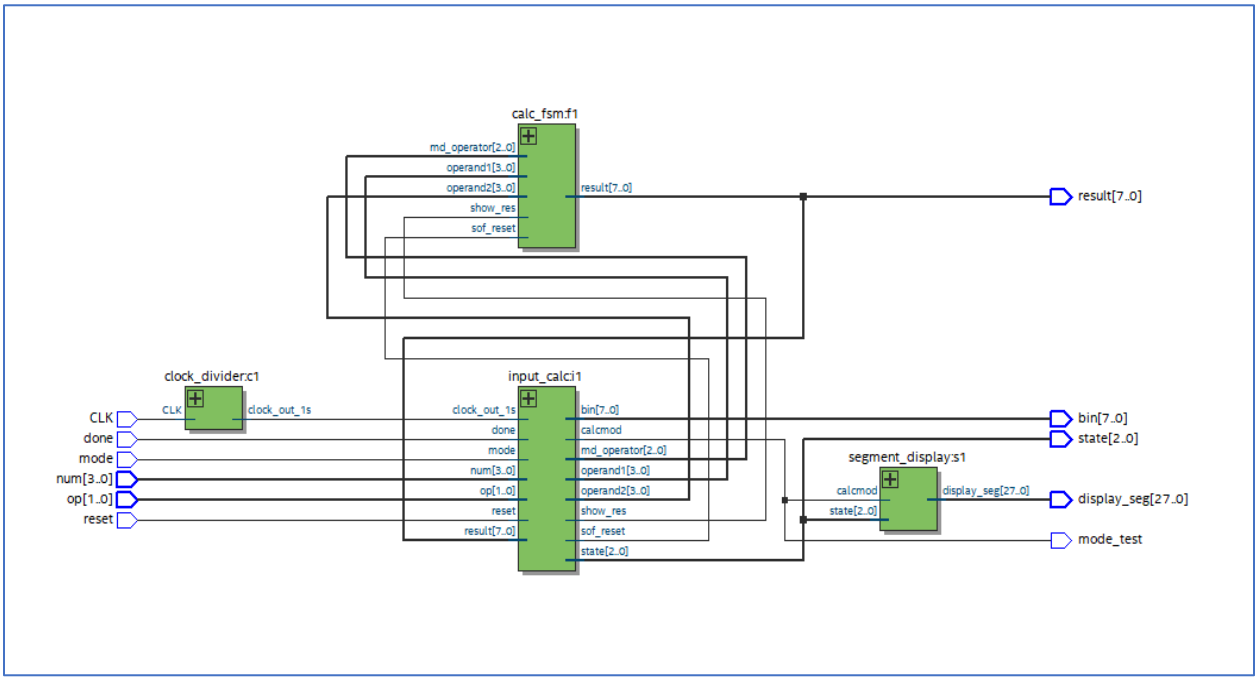


Figure 5: Top Module

ii) Sub Modules:

a. calc_fsm.v

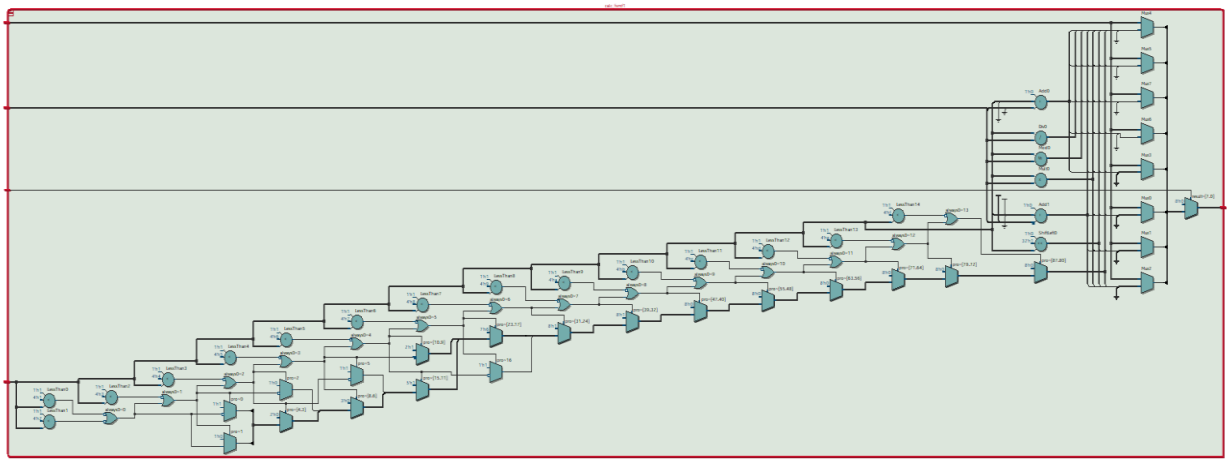


Figure 6: An RTL scematic of operators

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

b. input_calc.v

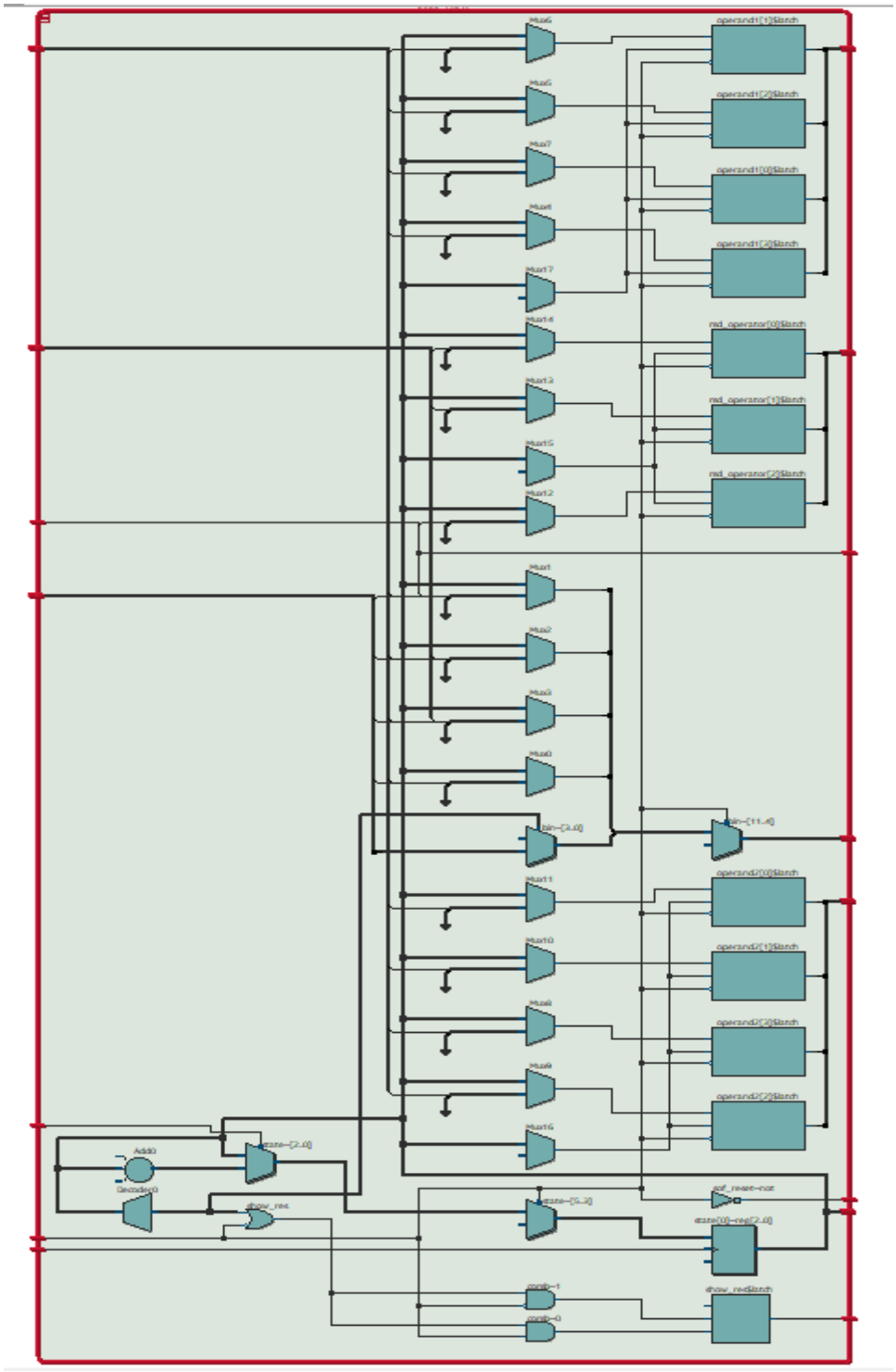


Figure 7: RTL schematic of the state machine

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

c. clock_divider.v

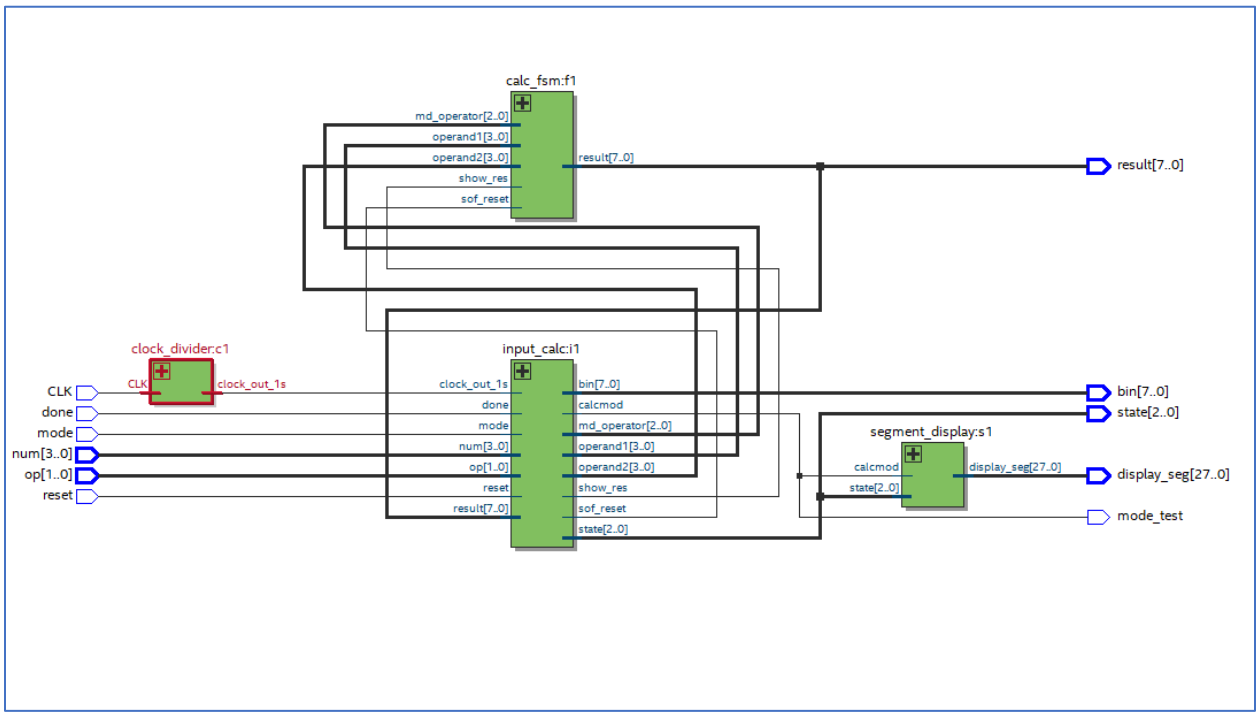


Figure 8: RTL Schematic of the clock_divider

Clock Divider Usage:

The Frequency of the FPGA processor is 50 MHz. Hence, in order to match the human frequency of data input, a clock divider has been used to synchronize the processor frequency with the human frequency.

The concept behind the execution of the clock divider is the implementation of a counter which divides the faster clock, thus bringing it down to the desired frequency. Hence, if we require to bring down 50MHz to 1Hz, the counter will count till 50×10^6, and then increment, thus dividing the parent frequency.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

b. segment_display.v

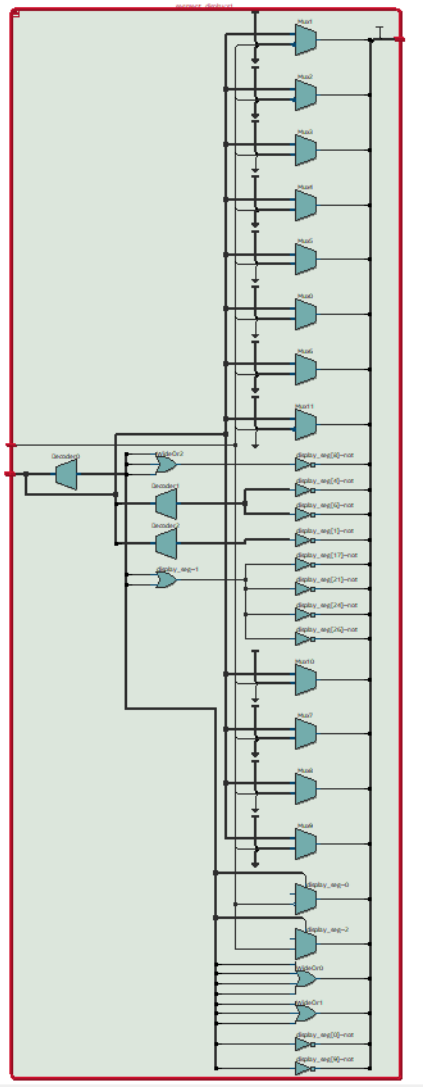


Figure 9. RTL Schematic of 7 segment display

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

REPORTS

FPGA SYNTHESIS REPORT

- The reports shown below are generated by Quartus Prime Lite 18.1 tool which summarized how much amount of the FPGA resources have been used to synthesis this vending machine design.

	Resource	Usage
1	Estimated Total logic elements	261
2		
3	Total combinational functions	261
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	103
2	-- 3 input functions	78
3	-- <=2 input functions	80
5		
6	Logic elements by mode	
1	-- normal mode	188
2	-- arithmetic mode	73
7		
8	Total registers	32
1	-- Dedicated logic registers	32
2	-- I/O registers	0
9		
10	I/O pins	58
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	input_calc:i1|operand1[1]
15	Maximum fan-out	37
16	Total fan-out	966
17	Average fan-out	2.36

Table 2: Analysis and Resource usage report

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Flow Summary

<<Filter>>

Flow Status	Successful - Wed May 06 14:37:20 2020
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	switch_calc
Top-level Entity Name	switch_calc
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	263 / 49,760 (< 1 %)
Total registers	32
Total pins	58 / 360 (16 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Table 3: Resource utilization summary

Post Synthesis Netlist

Post-Synthesis Netlist Statistics for Top Partition

<<Filter>>

	Type	Count
1	boundary_port	58
2	⌄ cycloneiii_ff	32
1	ENA SCLR	2
2	plain	30
3	⌄ cycloneiii_lcell_comb	267
1	⌄ arith	73
1	2 data inputs	31
2	3 data inputs	42
2	⌄ normal	194
1	0 data inputs	7
2	1 data inputs	6
3	2 data inputs	42
4	3 data inputs	36
5	4 data inputs	103
4		
5	Max LUT depth	12.50
6	Average LUT depth	6.00

Table 4: Post Synthesis Netlist

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Timing report:

Slow 1200mV 85C Model Fmax Summary

<<Filter>>

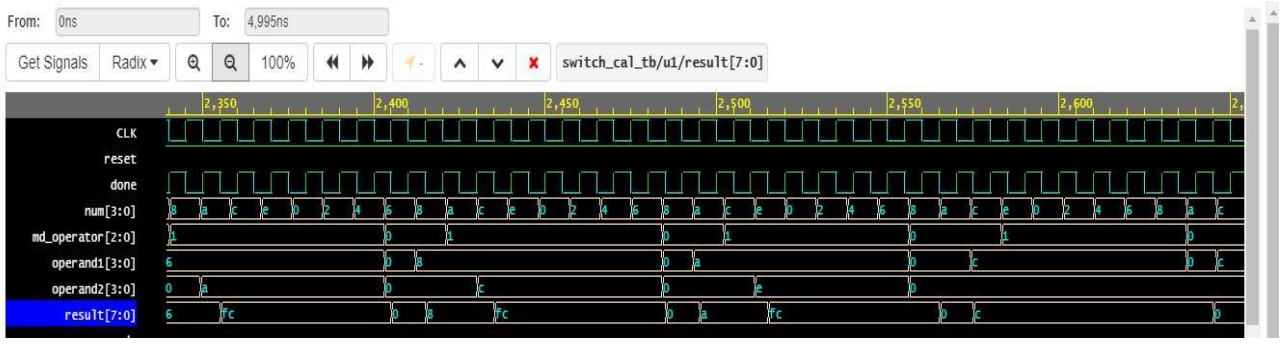
	Fmax	Restricted Fmax	Clock Name	Note
1	244.92 MHz	244.92 MHz	CLK	

Table 5: Clock divider output Frequency

As we can see above, the calculator can work efficiently at a maximum frequency of 244.92 MHZ without any glitches.

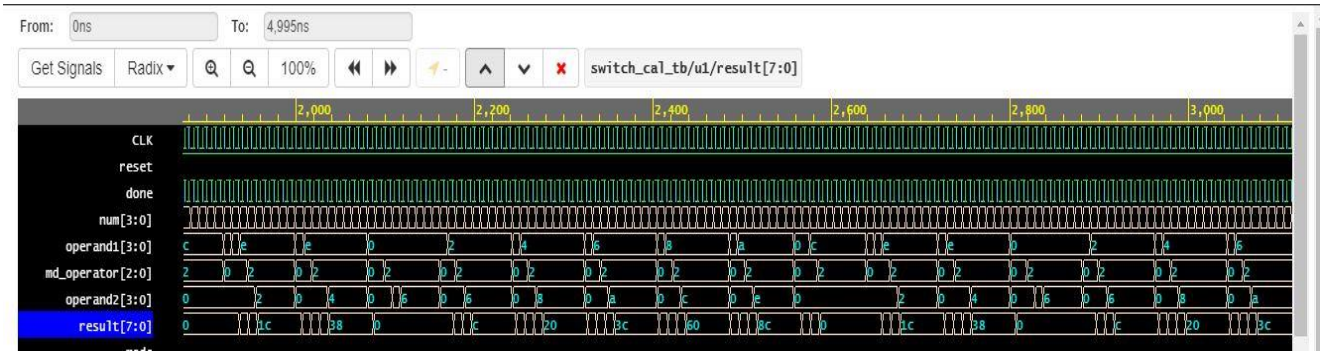
Testbench Simulation

Snap1: Simulation snapshot for addition

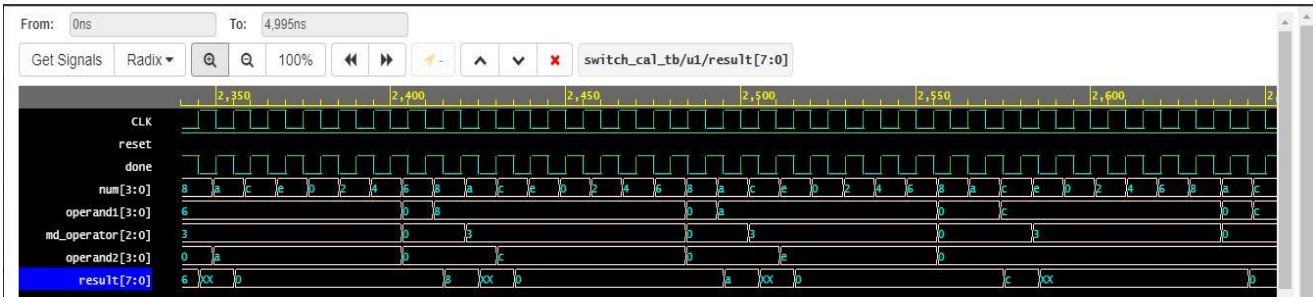


Snap2: Simulation snapshot for subtraction

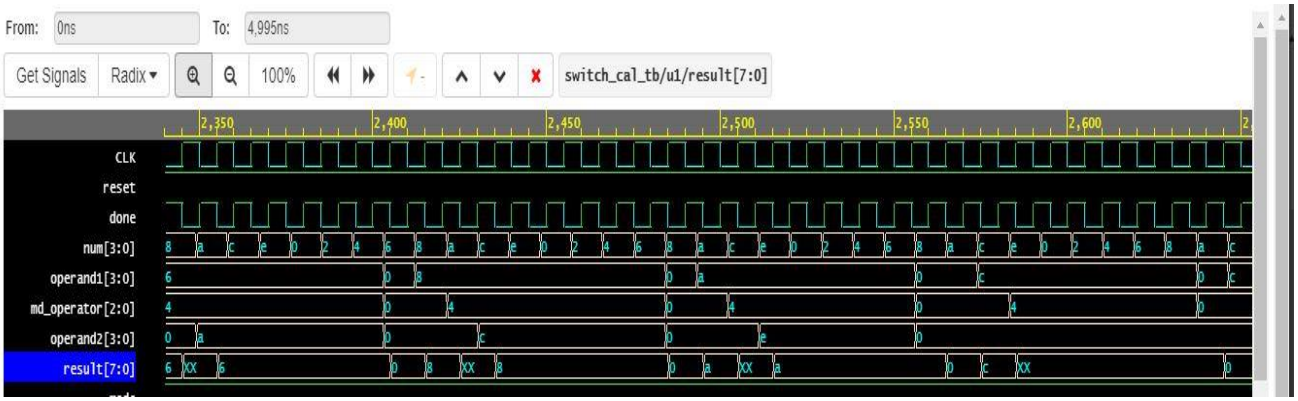
SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468



Snap3: Simulation snapshot for multiplication

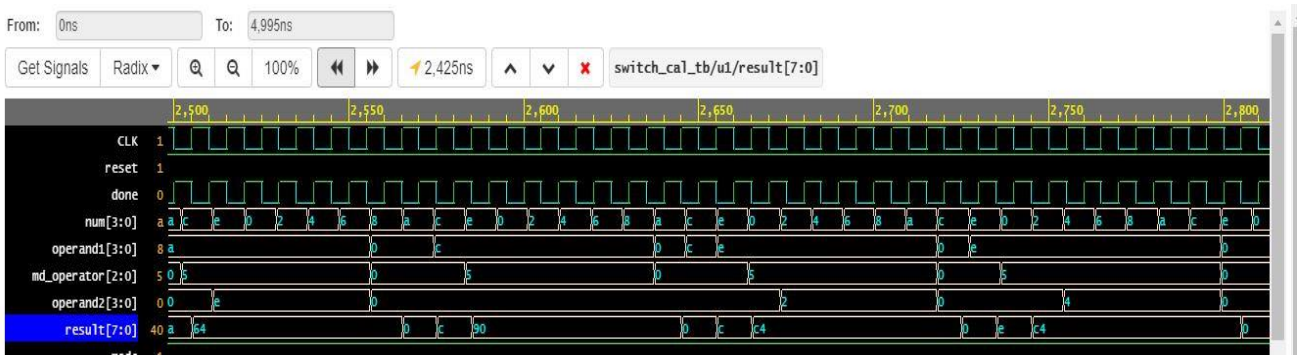


Snap 4: Simulation snapshot for division



Snap5: Simulation snapshot for modulus

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468



Snap6: Snapshot for power of 2

Snap7: Snapshot for factorial

Note: The values have been displayed in their respective hex_codes.

Conclusion

The FPGA based calculator has been designed and implemented using Quartus Prime Lite 18.1. The designed has been tested on the FPGA board DE-10 Lite Max. This calculator enhances the capability of the machine due to its modularity as well as scalability in terms of products due to its reprogrammable feature. The design is user friendly as well as its response time is fast. The future scope of this design is to add more functionality such as calculation of both positive and negative numbers and the ability to store the calculated result for future use.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Contribution

- Architecture designed of product from scratch.
- Defined micro product architecture which includes state machine, state defining and designing of each and every module with precise number of i/o ports and functionality.
- Designed architecture of verification bench to implement and test all features.
- Synthesis and implementation of scientific calculator on FPGA.
- Study and usage of the Arduino UNO board for the purpose of interfacing the LCD display to the FPGA board.
- Designing of the clock divider for the purpose of bringing down the frequency from 50 MHz to 1Hz.
- Efficient usage of the 7-segment display provided on the FPGA board to show the Calculator mode.
- Timing analysis of the full Scientific Calculator using Quartus Prime Lite 18.1 timing analyzer.
  As show below compilation report showed negative slack in setup timing as -2.489 which is now 16.305

	Clock	Slack	End Point TNS
1	CLK	16.305	0.000

- Earlier, the clock report showed the clock frequency (CLK) as 1000Mhz.

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468

Clocks

<<Filter>>

	Clock Name	Type	Period	Frequency
1	CLK	Base	20.000	50.0 MHz

In order to fix that a SDC file was created where in clock was given constraint using the following command, “create_clock -period 20 {get_ports clk}”. This means that clock period is set to 20ns (50Mhz).

Name	Contribution
Soumya Sahu	33.33%
Sarika Satish Natu	33.33%
Yamini Santosh Awasthi	33.33%

SOUMYA SAHU : 014537666
SARIKA SATISH NATU : 014529138
YAMINI SANTOSH AWASTHI : 014560468