

# FIXED POINT CALCULATOR

Oakland University  
**School of Engineering and Computer Science**

Course: ECE 5736  
Summer 2022

Kushagra Gupta  
Roman Hryntsiv

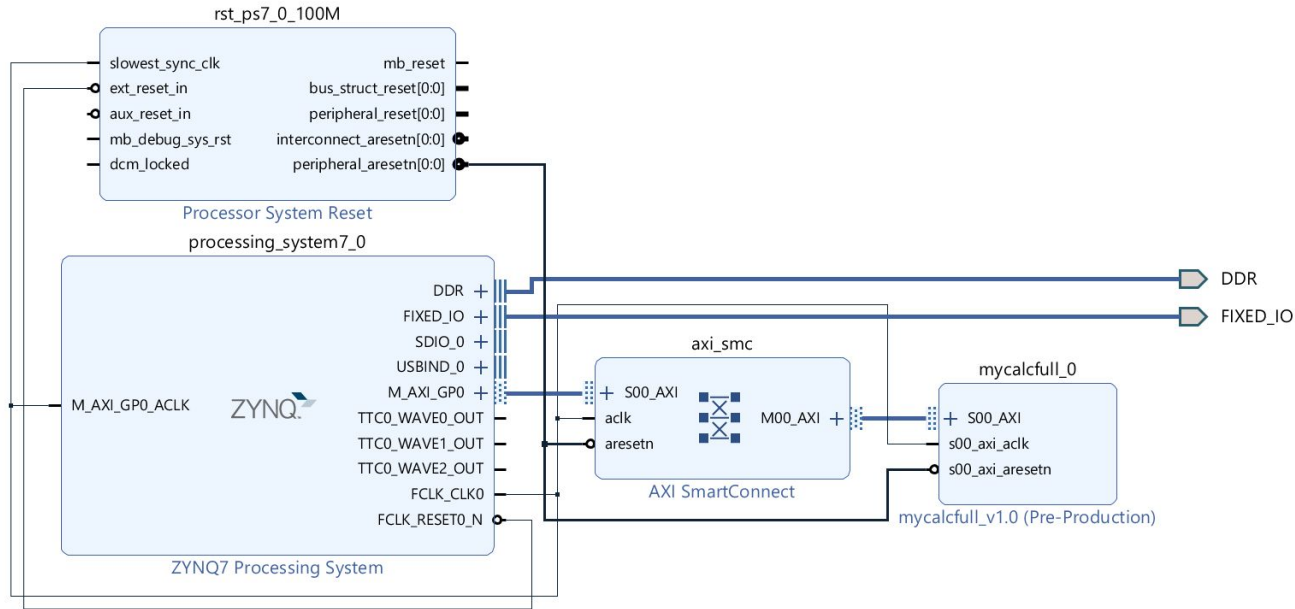
# AGENDA

- Project Overview
- Block Diagram
- Components
- Test Bench
- Experimental Setup
- Demo

# PROJECT OVERVIEW

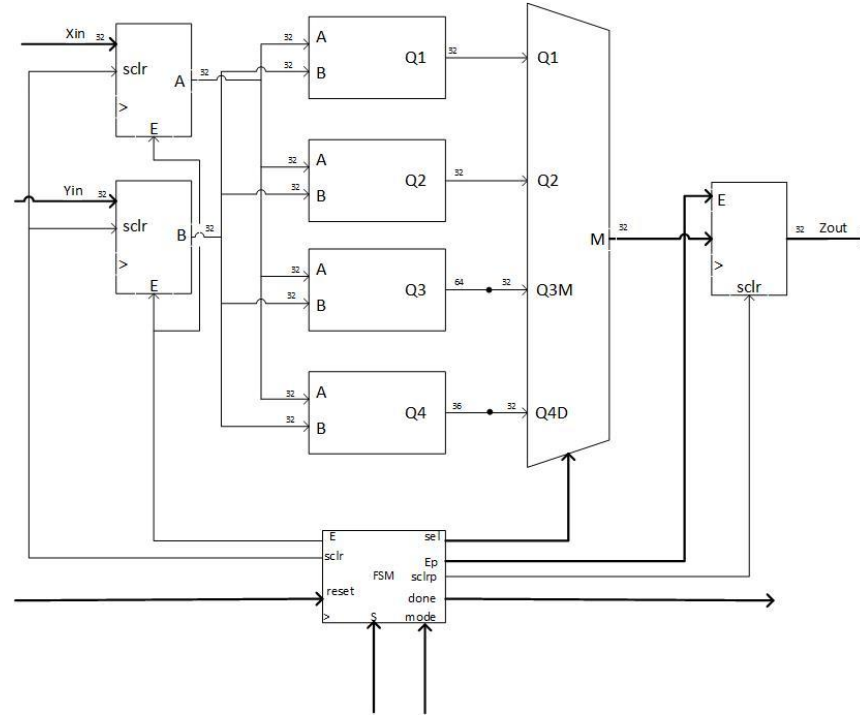
- Fixed-Point calculator using format [32 24]
- Operations Implemented - Addition, Subtraction, Multiplication, Division
- AXI4-Full peripheral for PS-PL communication
- Software design to read/write i/o data to text files on SD card

# BLOCK DIAGRAM



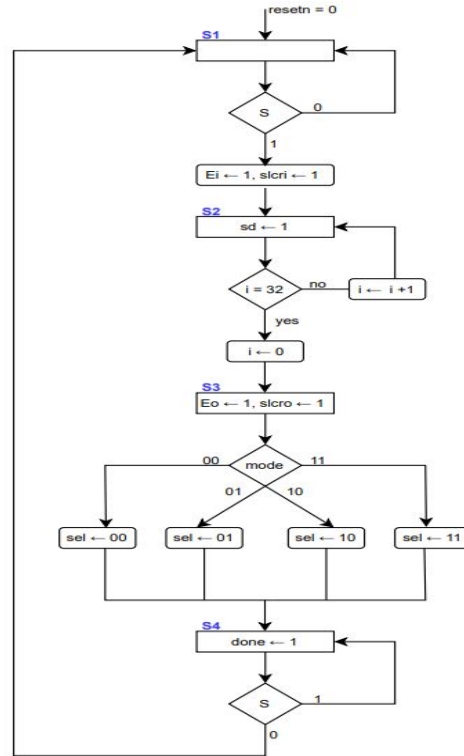
# COMPONENTS

- Calculator



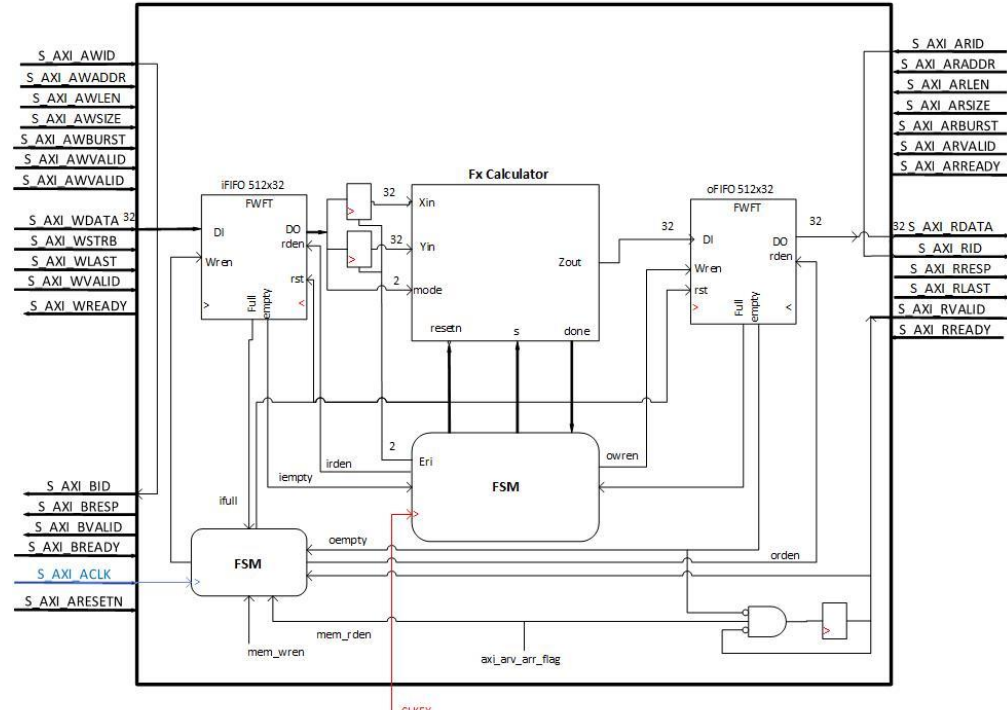
# COMPONENTS

- Calculator FSM



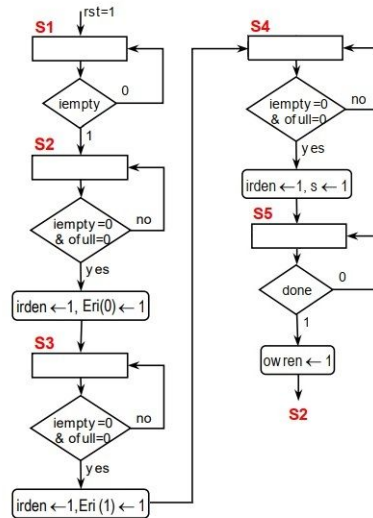
# COMPONENTS

- AXI4 -Full Peripheral



# COMPONENTS

- AXI4 -Full FSM



FSM @ CLKFX

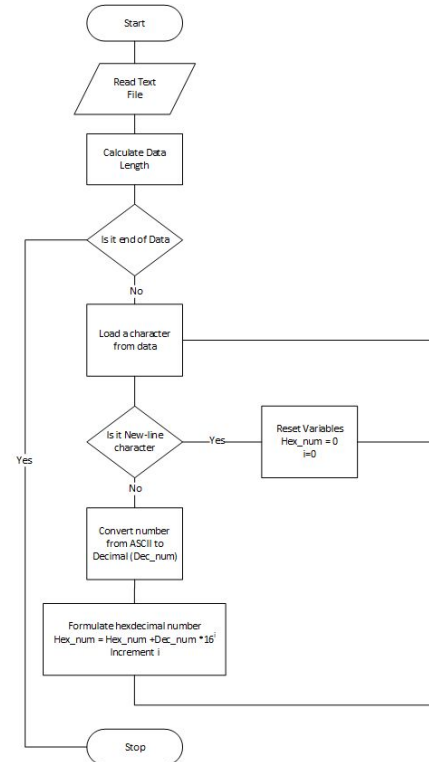


# COMPONENTS

- Software
- Main functionalities implemented in software are
  - Reading multiple text files
  - Extracting numbers from input data, converting to hex number and writing to AXI bus
  - Reading Data from AXI bus
  - Convert hex numbers to strings with fixed point representation
  - Writing Data to text files

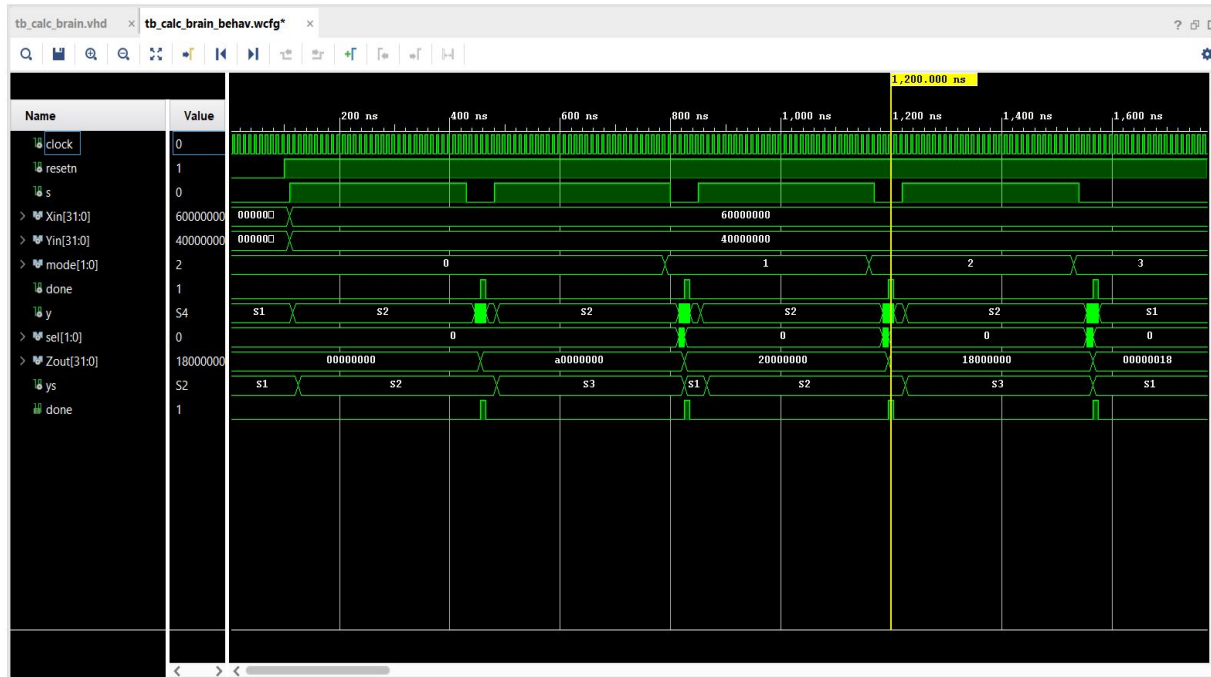
# COMPONENTS

- Software
  - Algorithm to formulate hex numbers from input text string



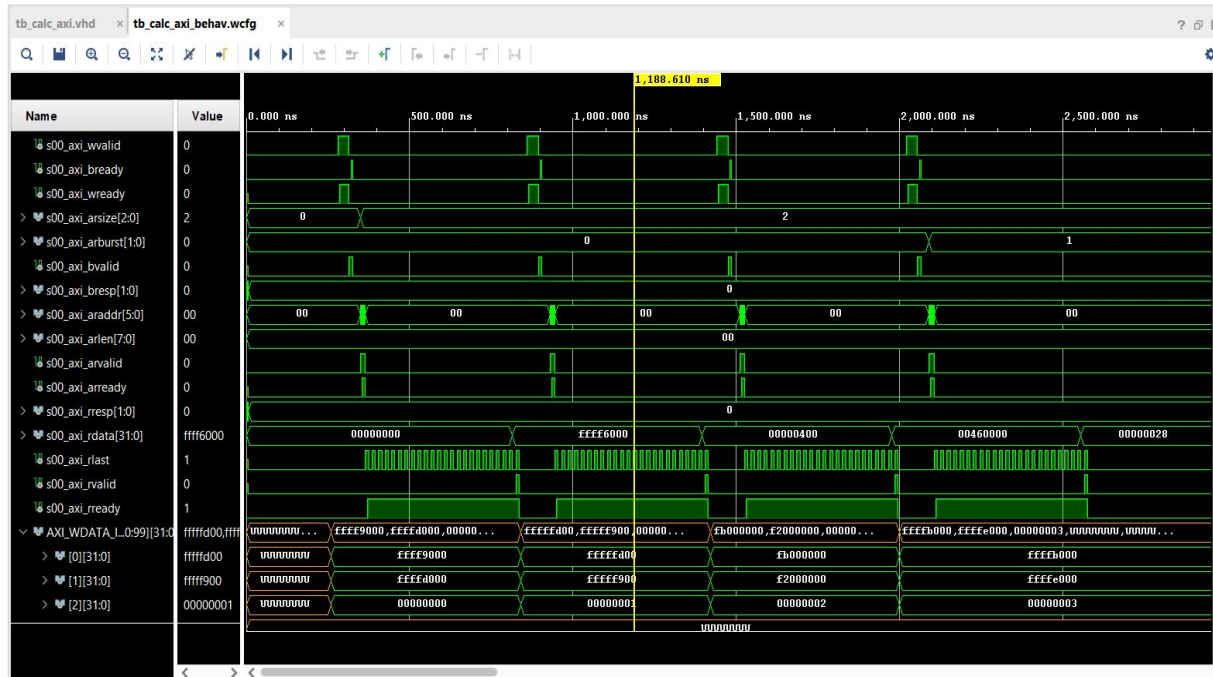
# TEST BENCH

- Calculator Test Bench



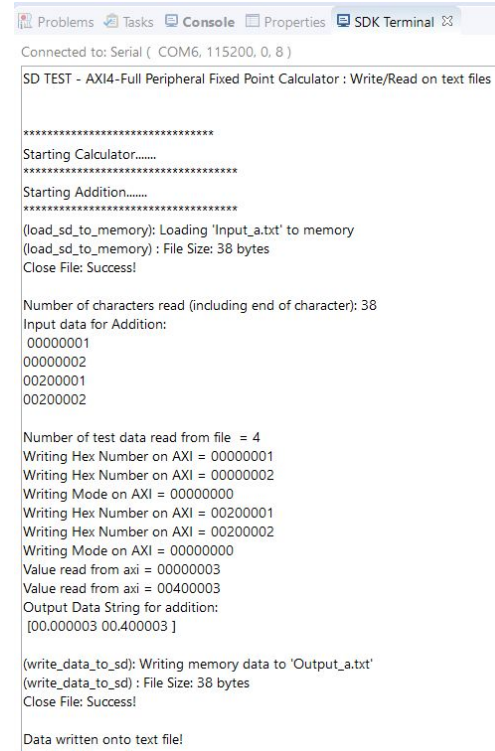
# TEST BENCH

- AXI4-Full Peripheral Test Bench



# EXPERIMENTAL SETUP

- Simple Calculations were used to test complete project with SDK
- Data for intermediate steps is printed on terminal
- Initial Testing for SDK and read/write to SD card performed using Lab3 (cordic with axi-full)



```
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( COM6, 115200, 0, 8 )

SD TEST - AXI4-Full Peripheral Fixed Point Calculator : Write/Read on text files

*****
Starting Calculator.....
*****
Starting Addition.....
*****
(load_sd_to_memory): Loading 'Input_a.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

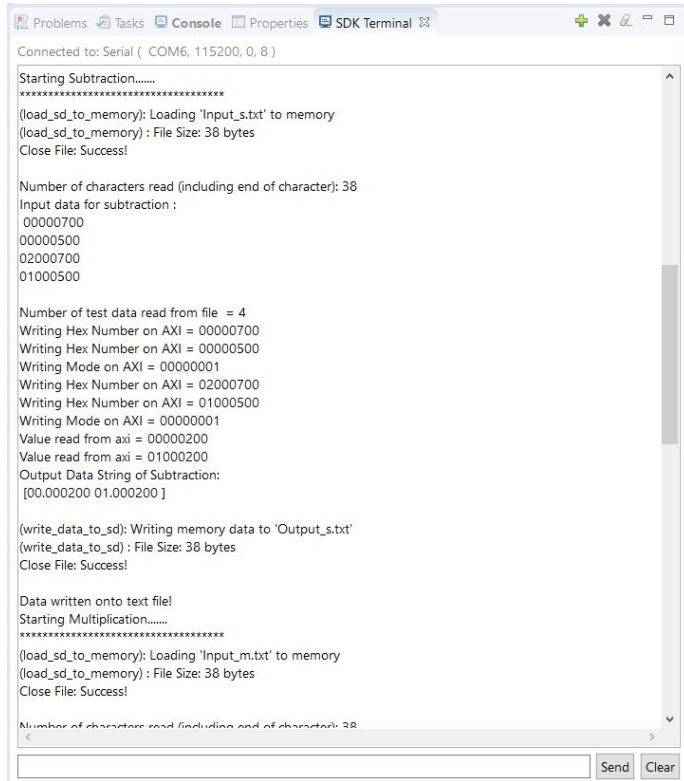
Number of characters read (including end of character): 38
Input data for Addition:
00000001
00000002
00200001
00200002

Number of test data read from file = 4
Writing Hex Number on AXI = 00000001
Writing Hex Number on AXI = 00000002
Writing Mode on AXI = 00000000
Writing Hex Number on AXI = 00200001
Writing Hex Number on AXI = 00200002
Writing Mode on AXI = 00000000
Value read from axi = 00000003
Value read from axi = 00400003
Output Data String for addition:
[00.000003 00.400003 ]

(write_data_to_sd): Writing memory data to 'Output_a.txt'
(write_data_to_sd) : File Size: 38 bytes
Close File: Success!

Data written onto text file!
```

# EXPERIMENTAL SETUP



```
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( COM6, 115200, 0, 8 )

Starting Subtraction.....
*****
(load_sd_to_memory): Loading 'Input_s.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

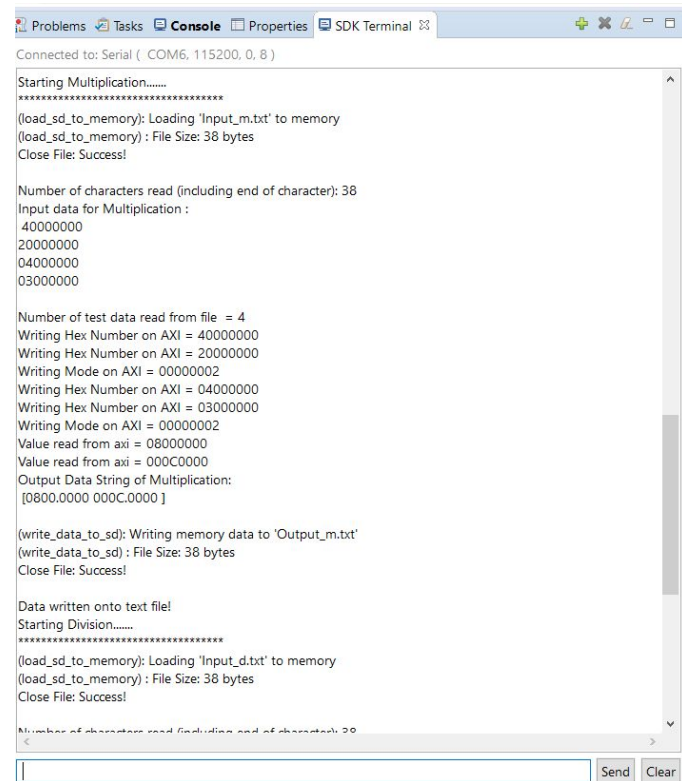
Number of characters read (including end of character): 38
Input data for subtraction :
00000700
00000500
02000700
01000500

Number of test data read from file = 4
Writing Hex Number on AXI = 00000700
Writing Hex Number on AXI = 00000500
Writing Mode on AXI = 00000001
Writing Hex Number on AXI = 02000700
Writing Hex Number on AXI = 01000500
Writing Mode on AXI = 00000001
Value read from axi = 00000200
Value read from axi = 01000200
Output Data String of Subtraction:
[00.000200 01.000200 ]

(write_data_to_sd): Writing memory data to 'Output_s.txt'
(write_data_to_sd) : File Size: 38 bytes
Close File: Success!

Data written onto text file!
Starting Multiplication.....
*****
(load_sd_to_memory): Loading 'Input_m.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

Number of characters read (including end of character): 38
```



```
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( COM6, 115200, 0, 8 )

Starting Multiplication.....
*****
(load_sd_to_memory): Loading 'Input_m.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

Number of characters read (including end of character): 38
Input data for Multiplication :
40000000
20000000
04000000
03000000

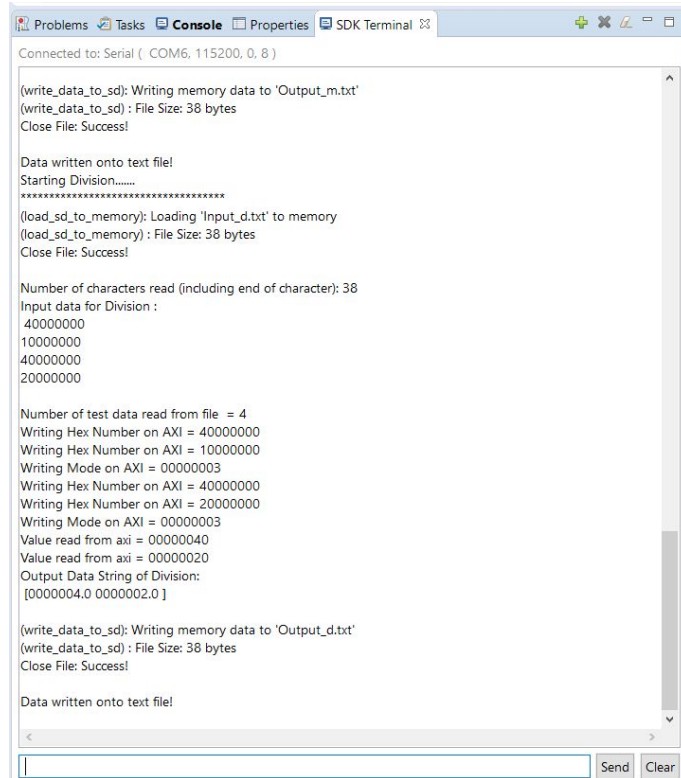
Number of test data read from file = 4
Writing Hex Number on AXI = 40000000
Writing Hex Number on AXI = 20000000
Writing Mode on AXI = 00000002
Writing Hex Number on AXI = 04000000
Writing Hex Number on AXI = 03000000
Writing Mode on AXI = 00000002
Value read from axi = 08000000
Value read from axi = 000C0000
Output Data String of Multiplication:
[0800.0000 000C.0000 ]

(write_data_to_sd): Writing memory data to 'Output_m.txt'
(write_data_to_sd) : File Size: 38 bytes
Close File: Success!

Data written onto text file!
Starting Division.....
*****
(load_sd_to_memory): Loading 'Input_d.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

Number of characters read (including end of character): 38
```

# EXPERIMENTAL SETUP



The screenshot shows an IDE window with a tab labeled 'SDK Terminal'. The terminal content is as follows:

```
Connected to: Serial ( COM6, 115200, 0, 8 )

(write_data_to_sd): Writing memory data to 'Output_m.txt'
(write_data_to_sd) : File Size: 38 bytes
Close File: Success!

Data written onto text file!
Starting Division.....
*****

(load_sd_to_memory): Loading 'Input_d.txt' to memory
(load_sd_to_memory) : File Size: 38 bytes
Close File: Success!

Number of characters read (including end of character): 38
Input data for Division :
40000000
10000000
40000000
20000000

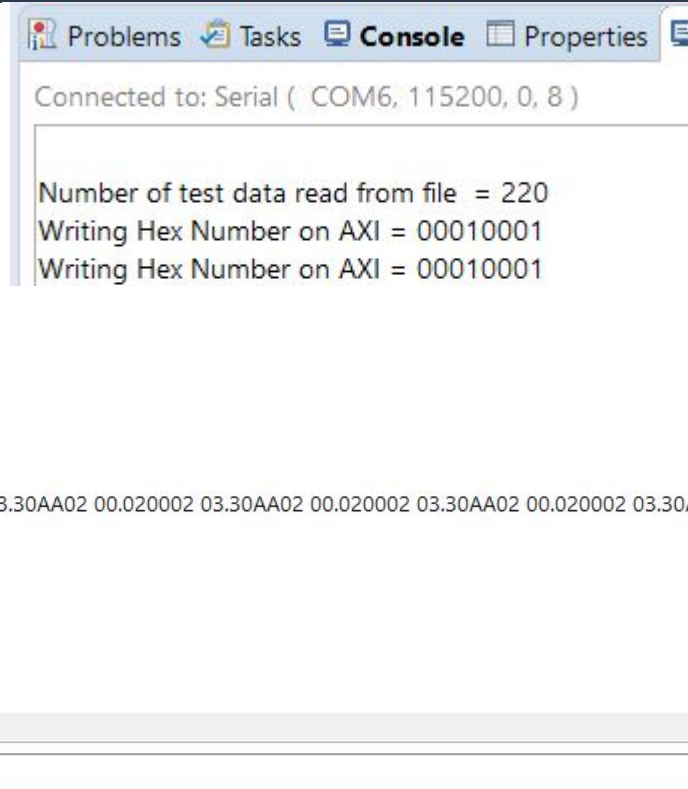
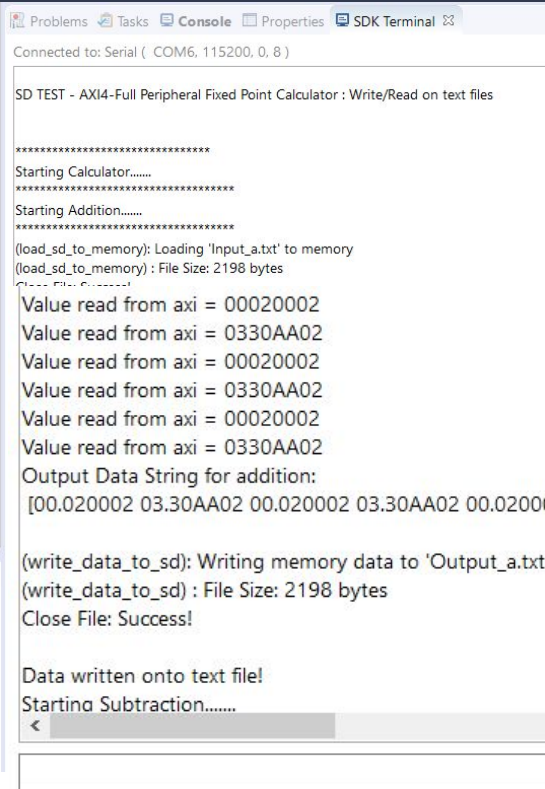
Number of test data read from file = 4
Writing Hex Number on AXI = 40000000
Writing Hex Number on AXI = 10000000
Writing Mode on AXI = 00000003
Writing Hex Number on AXI = 40000000
Writing Hex Number on AXI = 20000000
Writing Mode on AXI = 00000003
Value read from axi = 00000040
Value read from axi = 00000020
Output Data String of Division:
[0000004.0 0000002.0 ]

(write_data_to_sd): Writing memory data to 'Output_d.txt'
(write_data_to_sd) : File Size: 38 bytes
Close File: Success!

Data written onto text file!
```

At the bottom of the terminal window, there is an input field with a cursor and two buttons labeled 'Send' and 'Clear'.

# EXPERIMENTAL SETUP





DEMO