64-BIT FLOATING POINT Arithmetic Unit with Instruction & Data Memory USING VERILOG

INTRODUCTION

In modern scientific research and engineering applications, floating-point computations are essential for handling large dynamic ranges with high precision.
This project focuses on designing a 64-bit Floating Point Arithmetic Unit based on the IEEE-754 standard that can perform:

- Addition
- Subtraction
- Multiplication
- Division

The design includes:

- Instruction Memory
- Data Memory
- Program Counter (PC)
- Arithmetic Unit (using gate-level modeling)

DESIGN OVERFLOW

- **Instruction Memory:** 16-bit word length; stores operation code and data memory address. Operation selected using the last 2 LSB bits
- **Data Memory:** 128-bit word length; stores two 64-bit IEEE-754 floating-point operands (Operand A and Operand B).
- **Program Counter (PC):** 8-bit counter; provides sequential instruction addresses with synchronous reset and automatic increment after execution.
- **Arithmetic Unit (ALU):**Performs 64-bit floating-point addition, subtraction, multiplication, and division using gate-level modeling as per IEEE-754 standard.

INSTRUCTION FORMAT, DATA MEMORY & PROGRAM COUNTER

INSTRUCTION FORMAT (16-BIT):

- Bits 15-3: 13-bit Data Memory Address
- Bit 2: for Reading/Writing data at memory location
- Bits 1-0: Operation Code (00 Add, 01 Sub, 10 Mul, 11 Div)

DATA MEMORY (128-BIT WORD):

- First 64 bits → Operand A
- Last 64 bits → Operand B
- IEEE-754 double-precision format

PROGRAM COUNTER (PC):

- 8-bit register
- Increments after each instruction
- Synchronous reset to 00000000

GATE LEVEL ARITHMETIC UNIT

ADDITION & SUBTRACTION:

- Extract sign, exponent, and mantissa.
- Align exponents before operation.
- Perform addition/subtraction on mantissas.
- Normalize result and handle rounding.
- Reconstruct final IEEE-754 format.

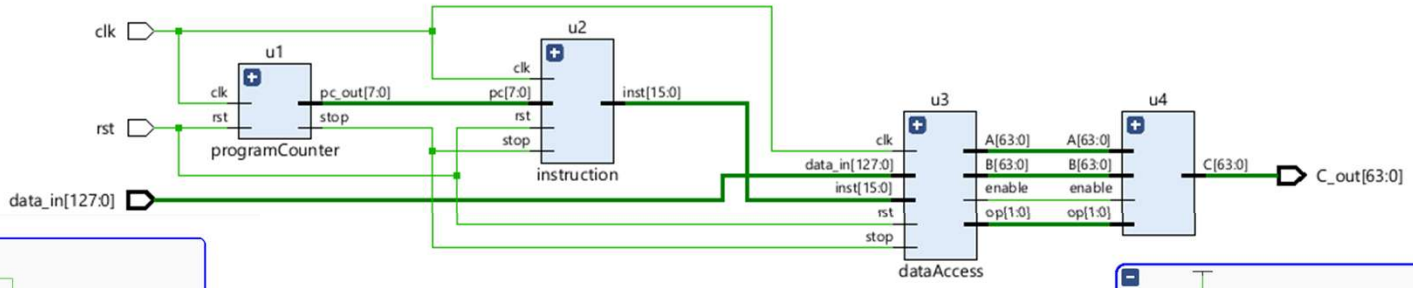
MULTIPLICATION:

- Add exponents (bias adjustment).
- Multiply mantissas.
- Normalize and round result.

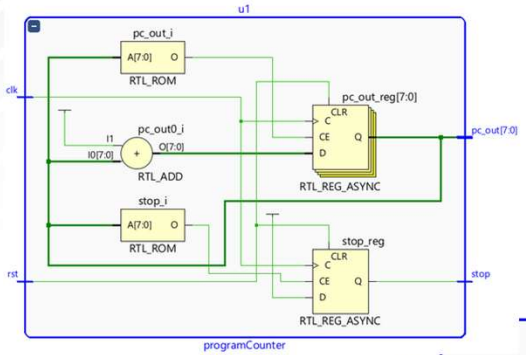
DIVISION:

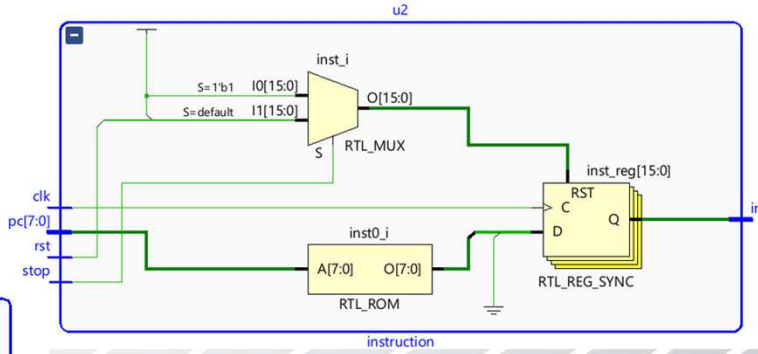
- Subtract exponents.
- Divide mantissas.
- Normalize and round result

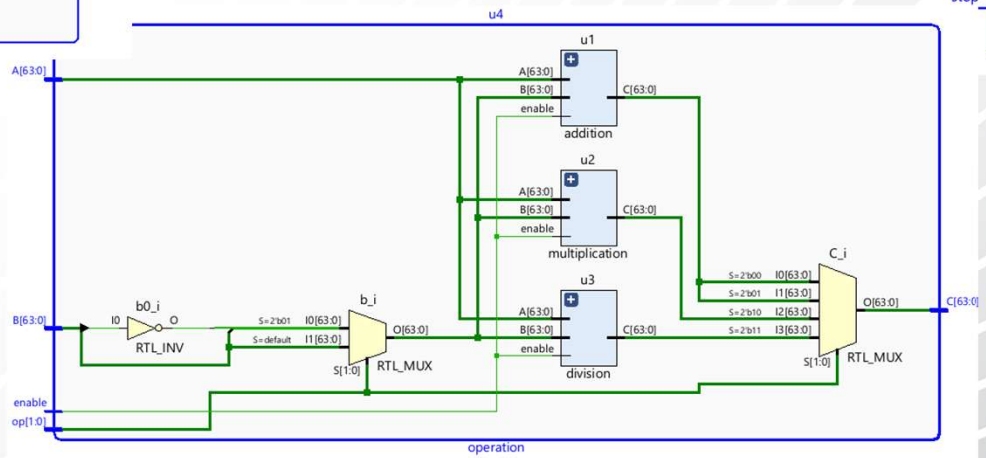
BEHAVIORAL MODULES



Top_module







TESTBENCH

1) c3FF0000000000000 4000000000000000 // 1.0,2.0
2) cBFF0000000000000 3FF0000000000000 // -1.0, 1.0
3) c3FF0000000000000 0000000000000000 // 1.0, 0.0
4) c0000000000000000 3FF0000000000000 // 0.0, 1.0
5) 7FF0000000000000 3FF0000000000000 // Inf, 1.0 write
6) 3FF0000000000000 7FF0000000000000 // 1.0, Inf
7) FFF0000000000000 3FF0000000000000 // -Inf, 1.0 write
8) 7FF8000000000000 3FF0000000000000 // NaN, 1.0
9) 3FF0000000000000 7FF8000000000000 // 1.0, NaN write
10) 0010000000000000 3FF0000000000000 // Denormal, 1.0 write
11) 3FF0000000000000 0010000000000000 // 1.0, Denormal
12) 3FF0000000000000 3FF0000000000000 // 1.0, 1.0
13) 3FE0000000000000 3FF0000000000000 // 0.5, 1.0
14) 3FF0000000000000 3FE0000000000000 // 1.0, 0.5
15) 3FF0000000000000 BFF0000000000000 // 1.0, (-1.0)
16) 4008000000000000 3FD0000000000000 // 3.0, 0.25
17) 3FF0000000000000 0000000000000001 // 1.0, very small
18) 3FF0000000000000 7FEFFFFFFFFFFFF // 1.0, max double
19) 3FF0000000000000 000FFFFFFFFFFFF // 1.0, largest subnormal
20) 3FF0000000000000 0000000000000000 // 1.0, 0.0
21) 400000000000000BFE0000000000000 //2, -0.5
22) 40520147AE147AE14046072B020C49BA //72.02,44.056
23) 405B6AE147AE147B4037000000000000 //109.67,23
24) 410C1B9A39581062C11BF199D495182B //230259.278, -457830.4576
25) C0808B999999999AC0B1D83851EB851F //-529.45, -4568.22

Dataset

```
# run 1000ns
Time     rst pc  inst    stop    en  A                B                C                C_out            r/w
10ns     0   0   xxxx    0   0   0000000000000000     0000000000000000     0010000000000000     0010000000000000     x
20ns     0   1   0002    0   0   xxxxxxxxxxxxxxxx     xxxxxxxxxxxxxxxx     0000000000000000     0000000000000000     0
60ns     0   2   0008    0   0   3ff0000000000000     4000000000000000     4000000000000000     4000000000000000     0
100ns    0   3   0012    0   0   bff0000000000000     3ff0000000000000     3ca0000000000000     3ca0000000000000     0
140ns    0   4   0019    0   0   3ff0000000000000     0000000000000000     3ff0000000000000     3ff0000000000000     0
180ns    0   5   0025    0   0   0000000000000000     3ff0000000000000     bff0000000000000     bff0000000000000     1
220ns    0   6   002a    0   0   0000000000000000     3ff0000000000000     0000000000000000     0000000000000000     0
260ns    0   7   0036    0   0   3ff0000000000000     7ff0000000000000     3ff0000000000000     3ff0000000000000     1
300ns    0   8   0038    0   0   3ff0000000000000     7ff0000000000000     0000000000000000     0000000000000000     0
340ns    0   9   0049    0   0   7ff8000000000000     3ff0000000000000     7ff8000000000000     7ff8000000000000     0
380ns    0   10  004d    0   0   0010000000000000     3ff0000000000000     bff0000000000000     bff0000000000000     1
420ns    0   11  0054    0   0   0010000000000000     3ff0000000000000     0000000000000000     0000000000000000     1
460ns    0   12  0058    0   0   0010000000000000     3ff0000000000000     0000000000000000     0000000000000000     0
500ns    0   13  0061    0   0   3ff0000000000000     3ff0000000000000     4000000000000000     4000000000000000     0
540ns    0   14  006a    0   0   3fe0000000000000     3ff0000000000000     bfe0000000000000     bfe0000000000000     0
580ns    0   15  0071    0   0   3ff0000000000000     3fe0000000000000     3fe0000000000000     3fe0000000000000     0
620ns    0   16  007a    0   0   3ff0000000000000     bff0000000000000     4000000000000000     4000000000000000     0
660ns    0   17  0082    0   0   4008000000000000     3fd0000000000000     3fe8000000000000     3fe8000000000000     0
700ns    0   18  008b    0   0   3ff0000000000000     0000000000000001     0000000000000000     0000000000000000     0
740ns    0   19  0090    0   0   3ff0000000000000     7fefffffffffffff     7ff0000000000000     7ff0000000000000     0
780ns    0   20  009a    0   0   3ff0000000000000     000fffffffffffff     3ff0000000000000     3ff0000000000000     0
820ns    0   21  00a3    0   0   3ff0000000000000     0000000000000000     0000000000000000     0000000000000000     0
860ns    0   22  00a8    0   0   4000000000000000     bfe0000000000000     c010000000000000     c010000000000000     0
900ns    0   23  00b2    0   0   40520147ae147ae1     4046072b020c49ba     405d04dd2f1a9fbe     405d04dd2f1a9fbe     0
940ns    0   24  00bb    0   0   405b6ae147ae147b     4037000000000000     40a3b4d1eb851eb8     40a3b4d1eb851eb8     0
980ns    0   25  00c1    0   0   410c1b9a39581062     c11bf199d495182b     bfe0180c97a0ac8c     bfe0180c97a0ac8c     0
xsim: Time (s): cpu = 00:00:05 ; elapsed = 00:00:06 . Memory (MB): peak = 3242.250 ; gain = 0.000
INFO: [USF-XSim-96] XSim completed. Design snapshot 'top_module_tb_behav' loaded.
```

output table of testbench

THANK YOU