ICHIP ROUND -1

PROBLEM STATEMENT

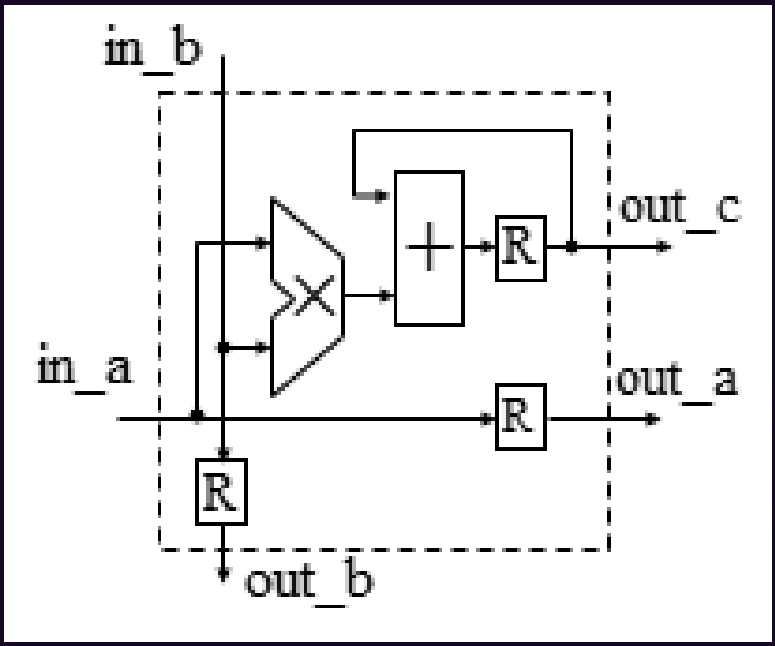
In the heart of Silicon Valley, where innovation ignites the sky and dreams are forged in lines of code, an esteemed alum from your college Jatin, stands at the helm of a burgeoning startup poised to revolutionize the world of AI acceleration. With a background in hardware design and a passion for pushing the boundaries of technology, Jatin embarks on a journey that will shape the future of AI engineering.

As the founder and CEO of his startup, Jatin is driven by a relentless determination to push the limits of what's possible in AI acceleration. He understands that to stay ahead in the competitive tech landscape, he must leverage the talent and expertise of the next generation of engineers - the students from his alma mater.

With this vision in mind, Jatin extends an invitation to the students, offering them a unique opportunity to collaborate on a groundbreaking project: the design of a cutting-edge AI accelerator in Verilog. Knowing that their contributions could have a real impact on the trajectory of his startup, the students eagerly accept the challenge, eager to prove themselves in the fast-paced world of tech entrepreneurship.

TASK:

The task at hand is to design and optimize a **Processing Element (PE)** that will serve as a fundamental building block for the AI accelerator envisioned by Jatin.



The PE consists of the following components:

- **Floating Point Multiplier:** This component is dedicated to performing multiplication operations on 32-bit floating-point numbers.
- **Floating Point Adder:** Responsible for executing addition and subtraction operations on 32-bit floating-point numbers.
- **Register:** The PE includes a register for storing intermediate results and operands during arithmetic operations.

EXAMPLE :

Suppose we have two arrays of 32-bit floating-point numbers:
Array 1=[1.5,-2.75]
Array 2=[-3.0,2.0]

Now to calculate floating-point binary representation of these arrays:
1.5 = (-1)^(0) * 2^(127 - 127) * (1.5)
0_01111111_10000000000000000000000
-2.75 = (-1)^(1) * 2^(128 - 127) * (1.375)
1_10000000_01100000000000000000000
-3.0 = (-1)^(1) * 2^(128 - 127) * (1.5)
1_10000000_10000000000000000000000
2.0 = (-1)^(0) * 2(128 - 127) * (1.0)
0_10000000_00000000000000000000000

The floating-point binary representation of these arrays will be:
Array 1=[0_01111111_10000000000000000000000,
1_10000000_01100000000000000000000]
Array 2=[1_10000000_10000000000000000000000,
0_10000000_00000000000000000000000]

Let's say the processing element is initialized with a previous output value of 10.0.

Now, we'll compute the output of the processing element by taking the product of corresponding elements from Array 1 and Array 2, and then adding it to the previous output for each cycle.

First cycle:

1.5 × -3.0 = -4.5
= (-1)^(0) * 2^(127 - 127) * (1.5) * (-1)^(1) * 2^(128 - 127) * (1.5)
= (-1)^(1) * 2^(128 - 127) * (2.25)
= (-1)^(1) * 2^(129 - 127) * (1.125)
= 1_10000001_00100000000000000000000

Add to the previous output: 10.0 + -4.5 = 5.5
= (-1)^(0) * 2^(130-127) * (1.25) + (-1)^(1) * 2^(129 - 127) * (1.125)
= 2^(129 - 127) * ((-1)^(0) * 2^(128 - 127) * (1.25) + (-1)^(1) * (1.125))
= 2^(129 - 127) * (2.5 - 1.125)
= (-1)^(0) * 2^(129 - 127) * (1.375)
= 0_10000001_01100000000000000000000

Second cycle :

-2.75 × 2.0 = -5.5
= (-1)^(1) * 2^(128 - 127) * (1.375) * (-1)^(0) * 2(128 - 127) * (1.0)
= (-1)^(1) * 2^(129 - 127) * (1.375)
= 1_10000001_01100000000000000000000

Add to the previous output: 5.5 + -5.5 = 0.0
= (-1)^(0) * 2^(129 - 127) * (1.375) + (-1)^(1) * 2^(129 - 127) * (1.375)
= 2^(129 - 127) * ((-1)^(0) * (1.375) + (-1)^(1) * (1.375))
= 2^(129 - 127) * (0)
= 2^(0 - 127) * (1.0)
= 0_00000000_00000000000000000000000








DESIGN SPECIFICATIONS :

- Inputs and outputs are of 32 bits each following the IEEE 754 binary 32 standard.
- Both floating-point adder and multiplier are combinational circuits.
- Describe the design as partitioned into modules (e.g., adder and multiplier as separate modules) to promote reuse and readability.

SUBMISSION :

- Submit the Verilog files of the main design module and the test bench.
- Submit a pdf file explaining your approach (no need of decorating it).