3155 Olsen Drive
San Jose, CA 95117

etched

Take-Home Multiplier Assignment

Warning - this task is very hard. It is not required for a position at Etched, but solving this task will show world-class talent in solving very hard problems. It may be solved in many ways - through clever math symmetrics, through very efficient algorithms, or by reasoning it out on paper.

The problem

Transformers are made up of many matmuls, of which one key piece is their **multiplier**. This takes in two numbers and computes their product. Etched's next-gen product will support MxFP4, a format for binary floating point numbers with four bits. The sixteen bit strings it supports are shown on the right. For the purposes of this problem, we may ignore signed zero, and may ignore the scaling factors[1].

Value	Binary
0	0000
0.5	0001
1	0010
1.5	0011
2	0100
3	0101
4	0110
6	0111
0	1000
-0.5	1001
-1	1010
-1.5	1011
-2	1100
-3	1101
-4	1110
-6	1111

Your job is to design the best possible FP4 multiplier. It will take in two FP4 numbers, and output **four times their product** as a **nine-bit signed integer** using two's complement (we do it this way as it makes the "add" portion of the multiply-add operations faster). Some examples:

- 0.5 (represented in FP4 binary as 0001) times 0.5 equals 0.25. Four times that is , so your circuit should output the binary string **000000001**.
- -3 (represented in FP4 binary as 0001) times 1.5 equals -4.5. Four times that is -18, so your circuit should output the binary string **110111000**.
- We will call this format (nine-bit two's complement signed integers where the LSB represents one-fourth) **QI9**. A list of QI9[2] numbers may be seen in the appendix.

[1] In the real multiple-accumulate circuit, scaling factors are added after a 32-input adder. It takes up very little space.
[2] QI9 is a simplification of what Etched uses for our accumulator datatypes. Improving this format can make the adders even smaller, but we will ignore that for this problem.

3155 Olsen Drive
San Jose, CA 95117

etched

Input Remapping

There is one other technique at your disposal - **input remapping**[3]. By default the meanings of the four-bit bitstrings are shown above. But if it makes the multiplier smaller, we could make 1110 represent -6 and 1111 represent -4, or any combination of the above!

You may remap the inputs however you like, as long as the results are still correct. You must remap both of the inputs to the multiplier in the same way[4]. The remapped inputs must also have four bits[5]. You may not remap the output.

The Grading

Build the multiplier with the **lowest number of gates**. The legal gates are XOR, AND, OR (each with two inputs) and NOT (which has one input)[6]. Each of these counts as exactly one gate[7]

The constant-zero and constant-one gates are free. Any remapping of the inputs is free.

You may use any resources you like online, including AI assistants and search engines. Online resources will probably not be helpful.

A full-credit solution will have the best possible number of gates, and a compelling reason for why you cannot do it in any fewer.

Testing your Solution

To test your solution, you can use this Python notebook. Enter your input remapping into the INPUT_REMAP section, and your multiplier in write_your_multiplier_here, to see whether it gives the correct results.

Etched Take-Home Multiplier Assignment

Please submit any questions to your recruiter.

[3] It is not possible to build an optimal solution without this. Etched does remapping in our hardware today in many places - not just for inputs, but also for intermediate accumulator results and outputs.
[4] Relaxing this restriction and having separate mappings for the two inputs wouldn't help us anyway.
[5] Relaxing this restriction can help us, but must be done very carefully and with regard to the electrical properties of the circuit as having more bits for the remapped inputs requires additional flip-flops in our circuit, which draw more power. For the purposes of this problem, we can ignore it.
[6] Sohu uses hundreds of different gates, most of which have far more than two inputs/outputs. This further helps with the circuit but we will ignore it in this problem for simplicity.
[7] In real life the sizes are a little different, but these are all very simple gates anyway we can ignore this.

3155 Olsen Drive
San Jose, CA 95117

etched

Appendix: QI9 Datatype Reference (Selected Values)

Value	Value * 4	Binary
-36	-144	101110000
-24	-96	110100000
-18	-72	110111000
-16	-64	111000000
-12	-48	111010000
-9	-36	111011100
-8	-32	111100000
-6	-24	111101000
-4.5	-18	111101110
-4	-16	111110000
-3	-12	111110100
-2.25	-9	111110111
-2	-8	111111000
-1.5	-6	111111010
-1	-4	111111100
-0.75	-3	111111101
-0.5	-2	111111110
-0.25	-1	111111111
0	0	000000000
0.25	1	000000001
0.5	2	000000010
0.75	3	000000011
1	4	000000100

3155 Olsen Drive
San Jose, CA 95117

etched

1.5	6	000000110
2	8	000001000
2.25	9	000001001
3	12	000001100
4	16	000010000
4.5	18	000010010
6	24	000011000
8	32	000100000
9	36	000100100
12	48	000110000
16	64	001000000
18	72	001001000
24	96	001100000
36	144	010010000