An Approximate and Iterative Posit Multiplier

Group 33	
Saketh Jooluri	**IMT2022528**
Sai Swaraj Pachipala	**IMT2022581**
Likithkumar Rachaputi	**MT2024528**

Introduction:

Posit number system is an emerging number system which aims to be a competitor to the existing IEEE floating-point number system.

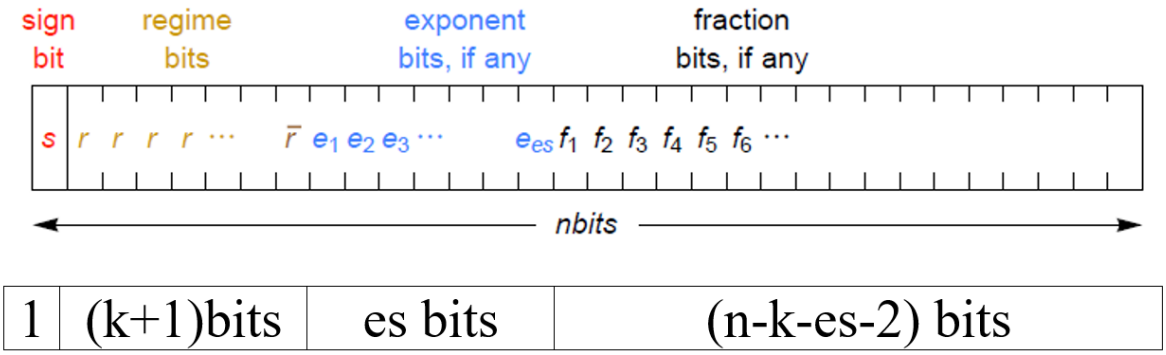


Figure 1. Generic posit format for an n-bit number

$$value = (-1)^s * (useed)^k * (2)^{exp} * (1 + frac)$$

$$useed = (2^{es})^k$$

Methodology:

1) First, we extracted the sign, regime, exponent, and mantissa bits from the given 32-bit numbers. This step is referred to as *decoding*.
2) The decoding process was performed for both numbers that we need to multiply.
3) Using the sign, regime, and exponent of the two numbers, we calculated the sign, regime, and exponent of the result based on the reference paper.
4) Using the mantissas of the two numbers, we computed the mantissa of the result using an iterative approach, as described in the reference paper.
5) Finally, using the result's sign, regime, and exponent (from step 3) and the mantissa (from step 4), we reconstructed the final result. This step is called *encoding*.

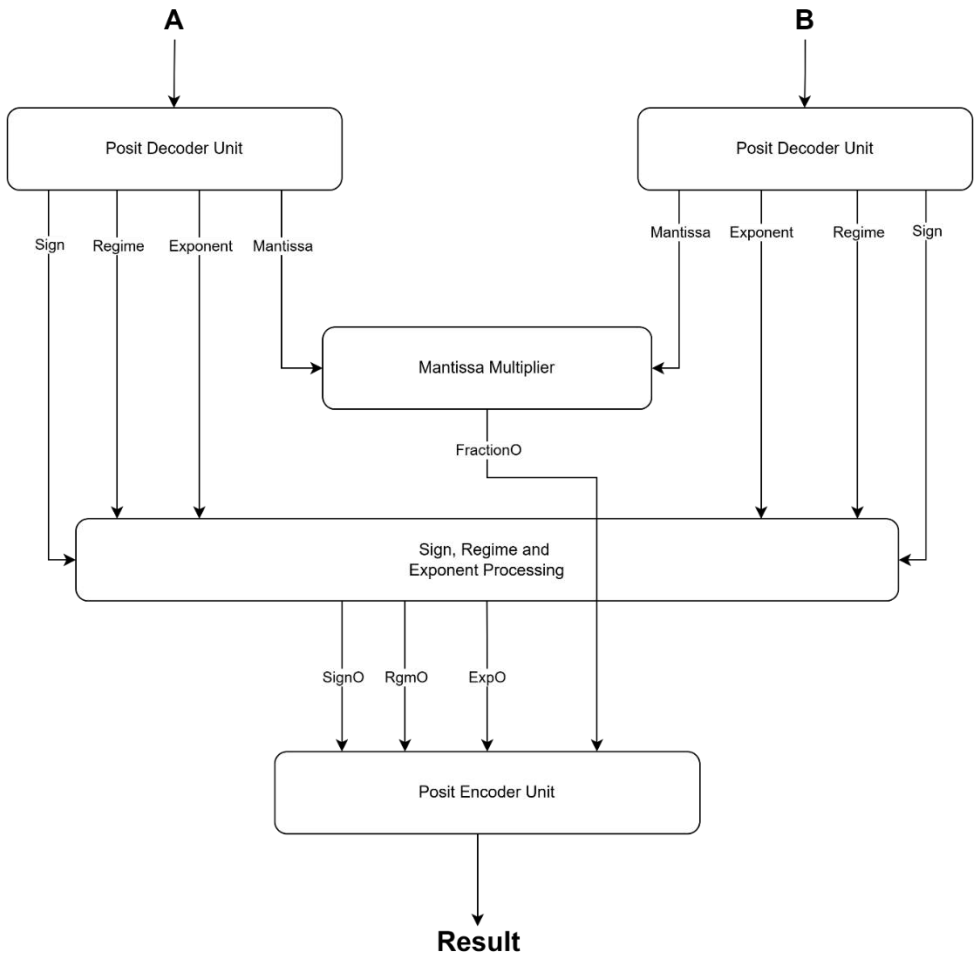


Figure 2: Datapath of the proposed multiplier circuit

Once we implemented the design in Verilog, we successfully simulated and verified the results. However, during synthesis, we encountered the following issues:

1. **Non-synthesizable loop variables**: To resolve this, we replaced the loop variable with its maximum value and used a conditional `if` statement within the loop to eliminate the error.
2. **Unsupported `break` statement in Verilog**: Since the `break` statement is not supported in Verilog, we replaced it with `i = -1;` which causes the loop to exit as intended.

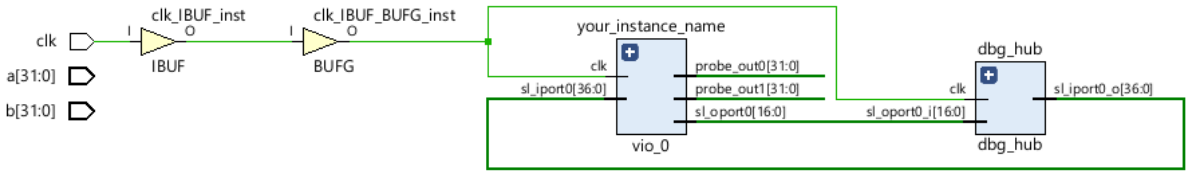
Simulation (2 errors)

sim_1 (2 errors)

[VRFC 10-2989] 'break' is not declared [posit_multiplier_top.v:158]

[XSIM 43-3322] Static elaboration of top level Verilog design unit(s) in library work failed.

Figure 3: Error screenshot during synthesis in Vivado



Results:

a) Max Clock Frequency Achieved:

Name	Waveform	Period (ns)	Frequency (MHz)
clk	{0.000 2.250}	4.500	222.222

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.130 ns	Worst Hold Slack (WHS):	0.075 ns	Worst Pulse Width Slack (WPWS):	1.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	2239	Total Number of Endpoints:	2223	Total Number of Endpoints:	1198

All user specified timing constraints are met.

$$Max\ Freq = \frac{1}{Tclk\ -\ WNS} = \frac{1}{(4.5\ -\ 0.13)ns} = 228.23MHz$$

b) Resource Utilization:

Hierarchy

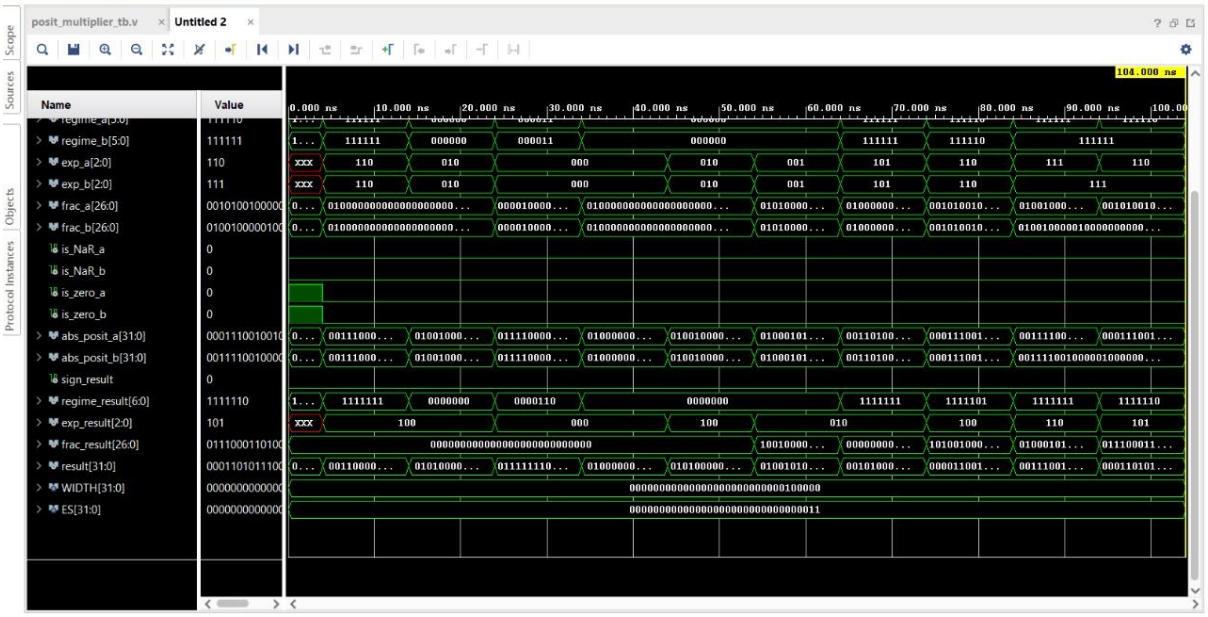
Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	Bonded IOB (106)	BUFGCTRL (32)	BSCANE2 (4)
posit_multiplier	582	1148	306	558	24	1	2	1
dbg_hub (dbg_hub)	449	741	219	425	24	0	1	1
your_instance_name (vio_0)	133	407	98	133	0	0	0	0

Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	Bonded IOB (106)	BUFGCTRL (32)	BSCANE2 (4)
posit_multiplier	2.80%	2.76%	3.74%	2.68%	0.25%	0.94%	6.25%	25.00%
dbg_hub (dbg_hub)	2.16%	1.78%	2.70%	2.04%	0.25%	0.00%	3.13%	25.00%
your_instance_name (vio_0)	0.64%	0.98%	1.18%	0.64%	0.00%	0.00%	0.00%	0.00%

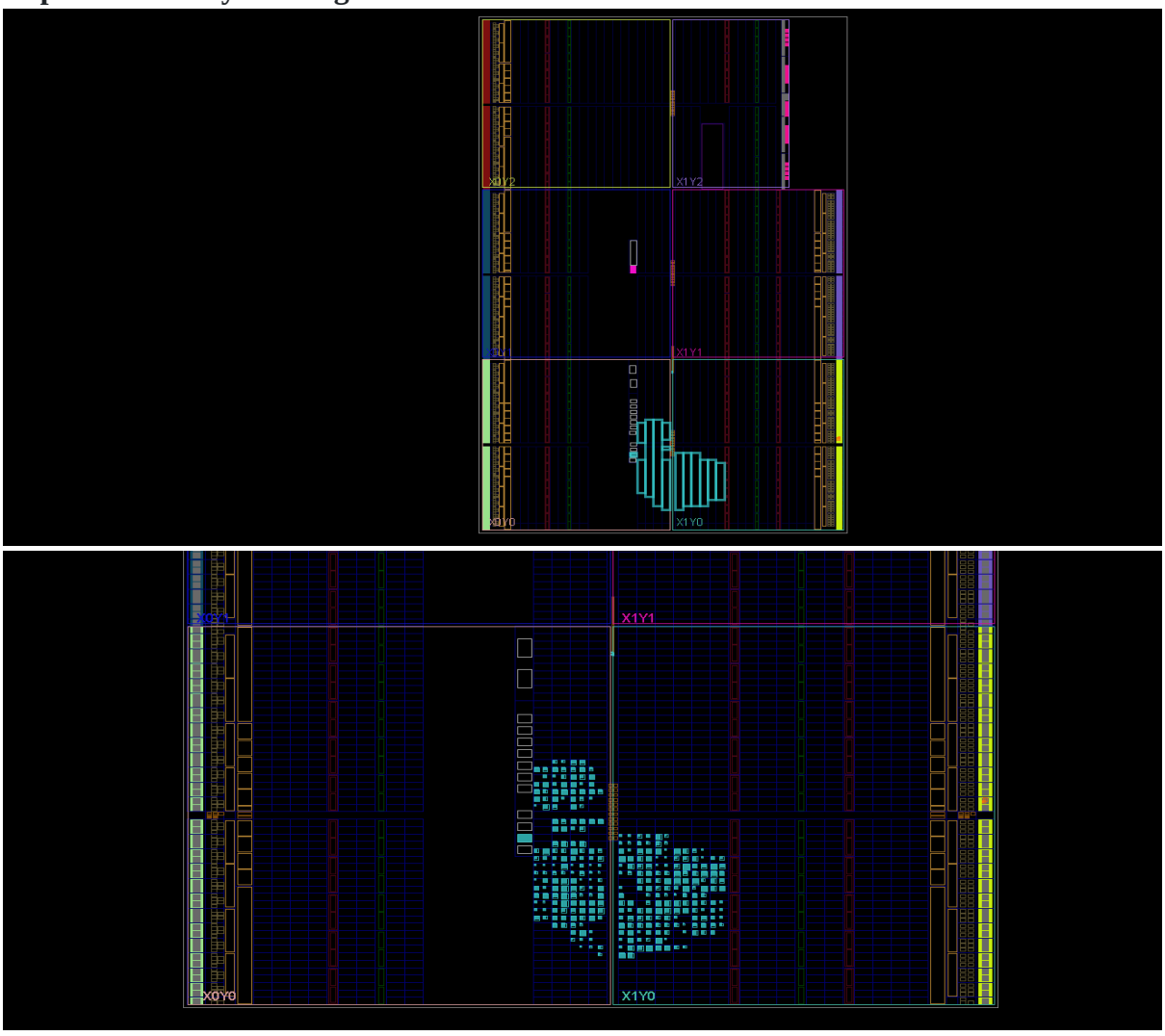
c) Latency of Computation:

Behavioural and Post Implementation simulation Result



No latency

d) Implemented Layout Diagram:



e) Max throughput achievable:

$$Max\ throughput = \frac{1}{Tclk * Max\ Utilization\ percentage}$$

$$Max\ throughput = \frac{1}{4.5ns * 3.8} * 100$$

$$Max\ throughput = 5.84 * 10^9 Hz$$

Maximum achievable throughput = 5.84GHz

f) Error Calculation

Error calculation is performed using a Python program

```
PS C:\Users\racha\OneDrive - iiit-b\Documents\VSCode\Python> & C:/Users/racha/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/racha/OneD
rive - iiit-b/Documents/VSCode/Python/posit.py"
Enter space-separated n and es: 32 3
Enter the number A: 00011100100100000000000000000000
sign_a = 0
Rgm_a =  -2
exp_a =  6
fraction_a =  1.28125
Posit number_a =  0.001251220703125

Enter space-separated n and es: 32 3
Enter the number B: 00111100100000100000000000000000
sign_b = 0
Rgm_b =  -1
exp_b =  7
fraction_b =  1.126953125
Posit number_b =  0.5634765625

SignO =  0
RgmO =  -2
ExpO =  5

Multiplication Result O:
0 0 0 1 1 0 1 0 1 1 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0

Posit_Result_1  0.000705033540725708
Posit_Result_2  0.000867009162902832
Error =  -0.229741725493511 43
PS C:\Users\racha\OneDrive - iiit-b\Documents\VSCode\Python>
```

- [GitHub Link]