EE6361 Advanced Topics in VLSI

Project Presentation

Akshay Sethia | EE15B072
Alfred Festus Davidson | EE15B073

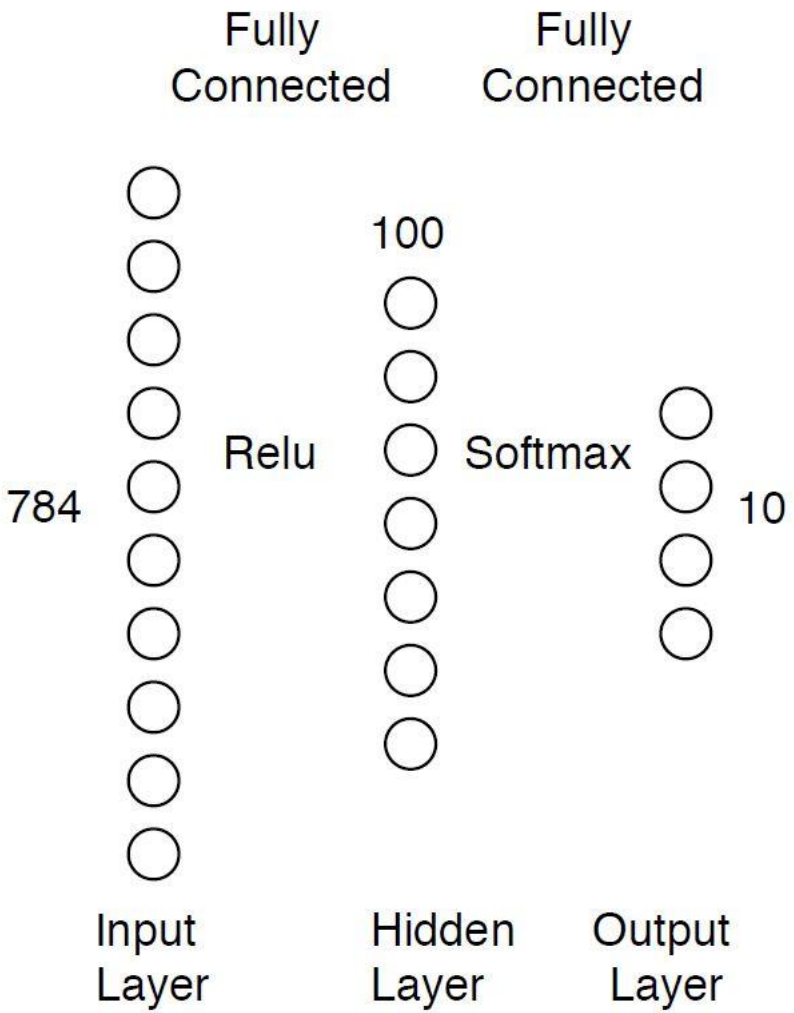
Introduction

- One of the main bottlenecks of machine learning algorithms is that they require access to large amounts of memory
- By modifying the word line drivers and bit line voltages, there have been circuit implementations that use bitline voltage/current for multiplication and charge based accumulation for summation.

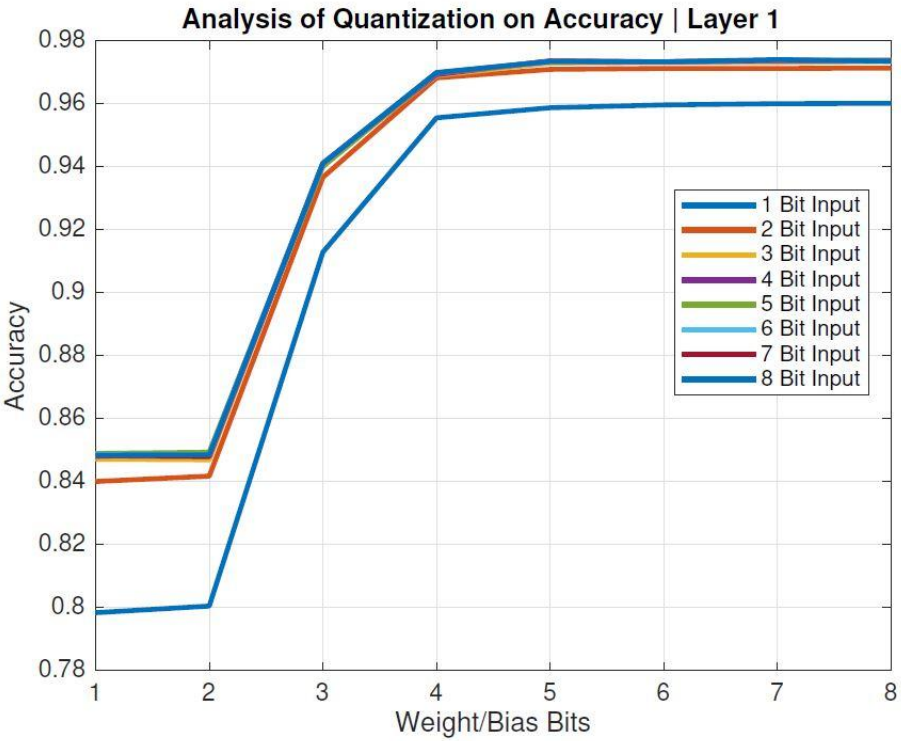
Introduction

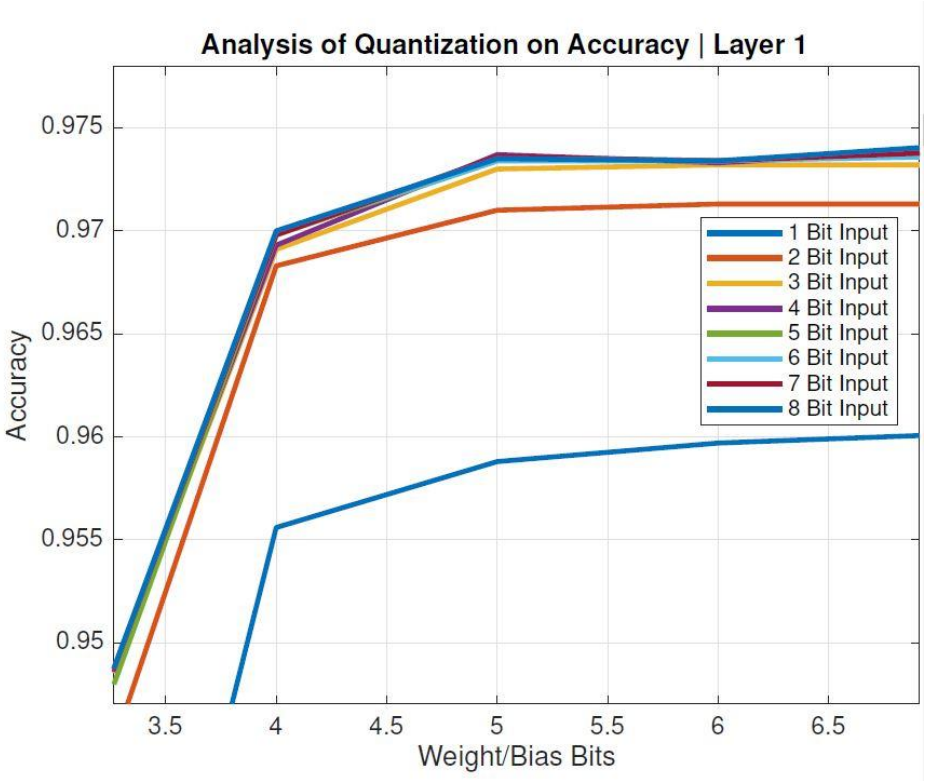
- However, they do not scale well for deep neural networks (DNN), simply because charge based accumulation will take longer time when the number of nodes are large.
- We propose a completely current based MAC unit that can be implemented alongside a standard 6T SRAM array that is easily scalable for large DNN.

Evaluation Network | MNIST

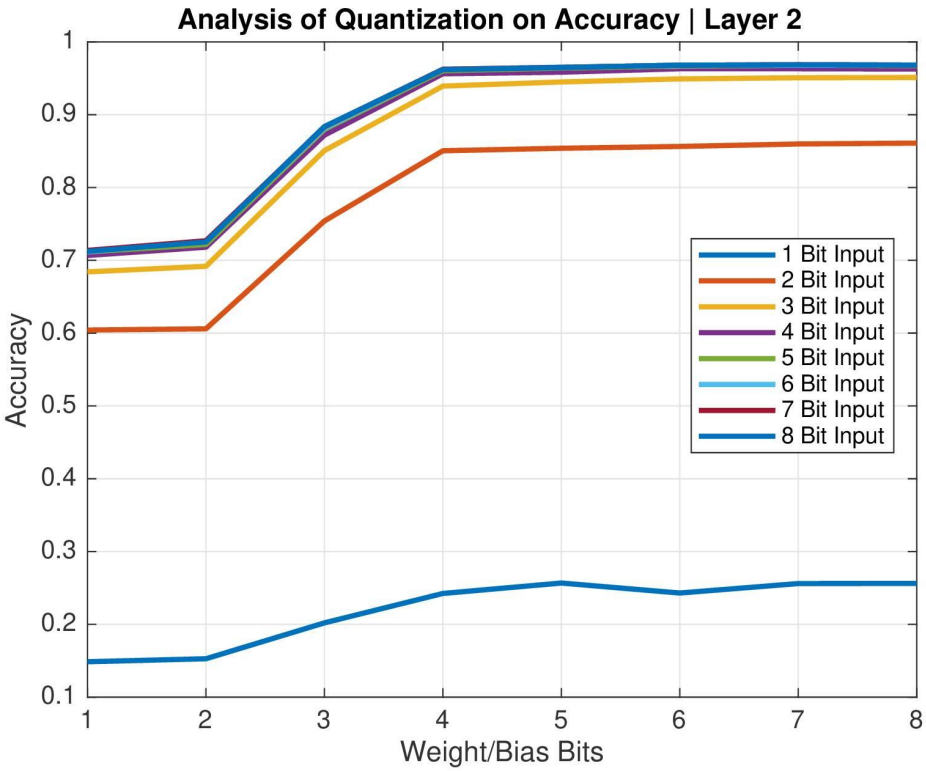


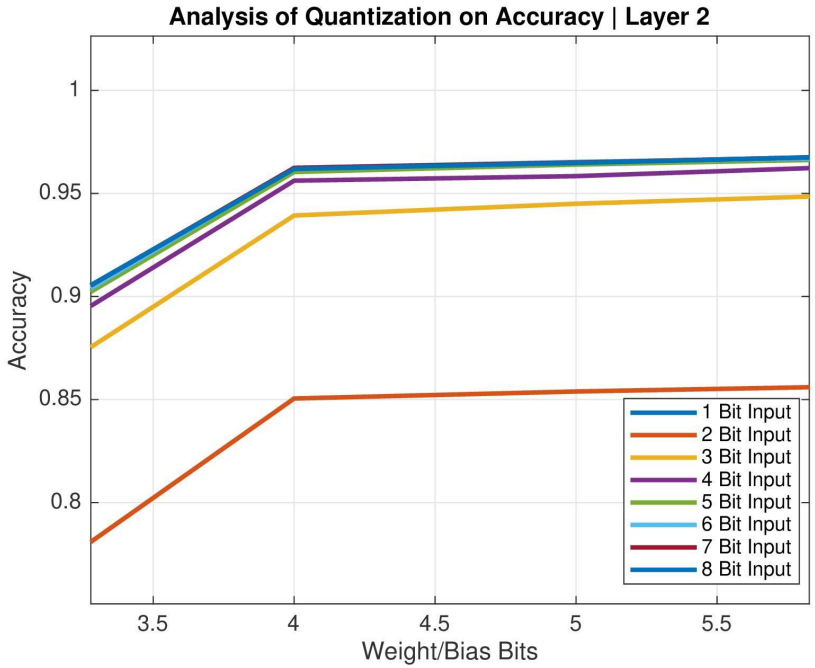
Effects of Quantization | Layer 1





Effects of Quantization | Layer 2





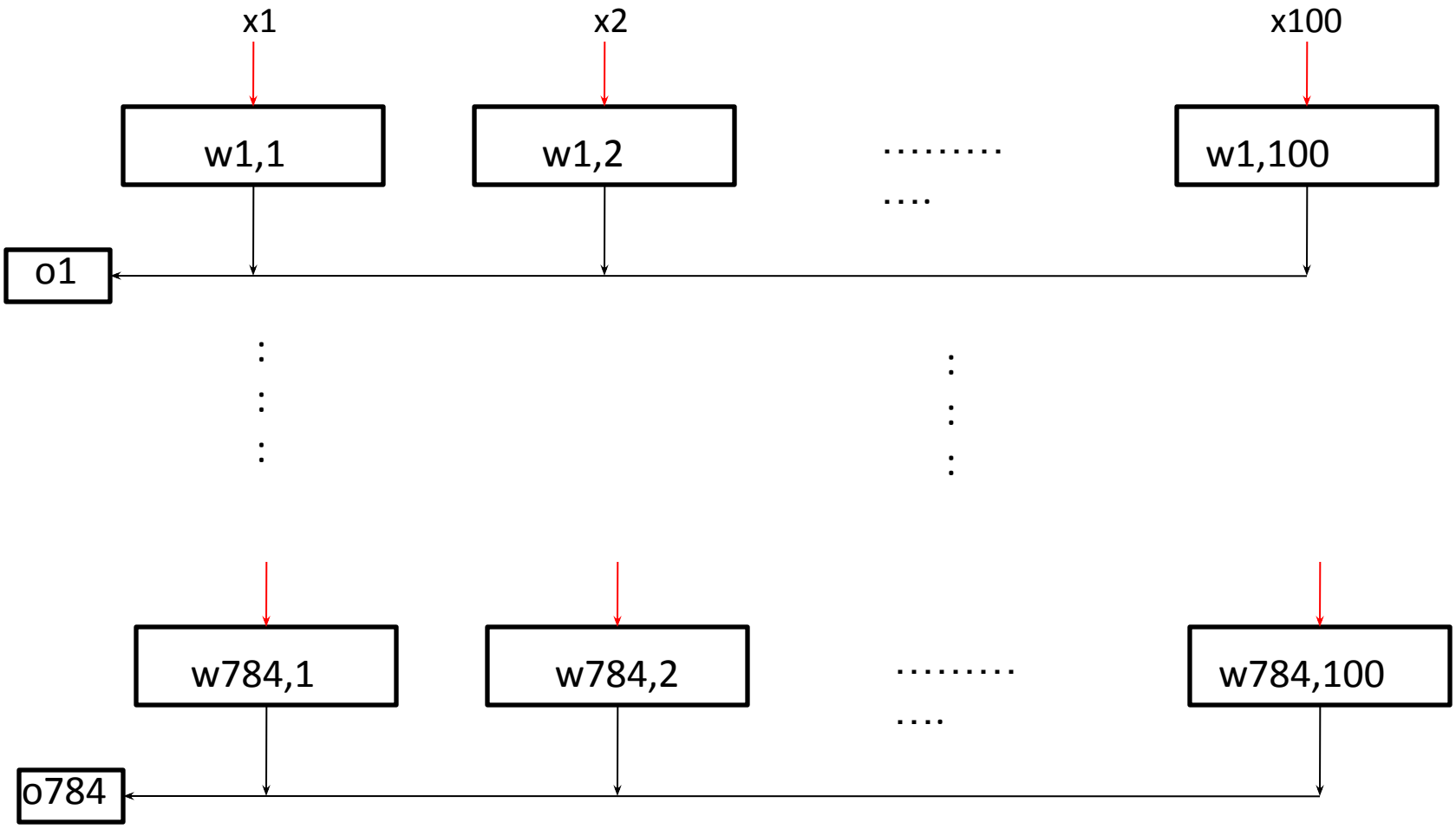
Effects of Quantization

We decide to quantize the first layer input to 2 bits and the first layer weights/biases to 4 bits.

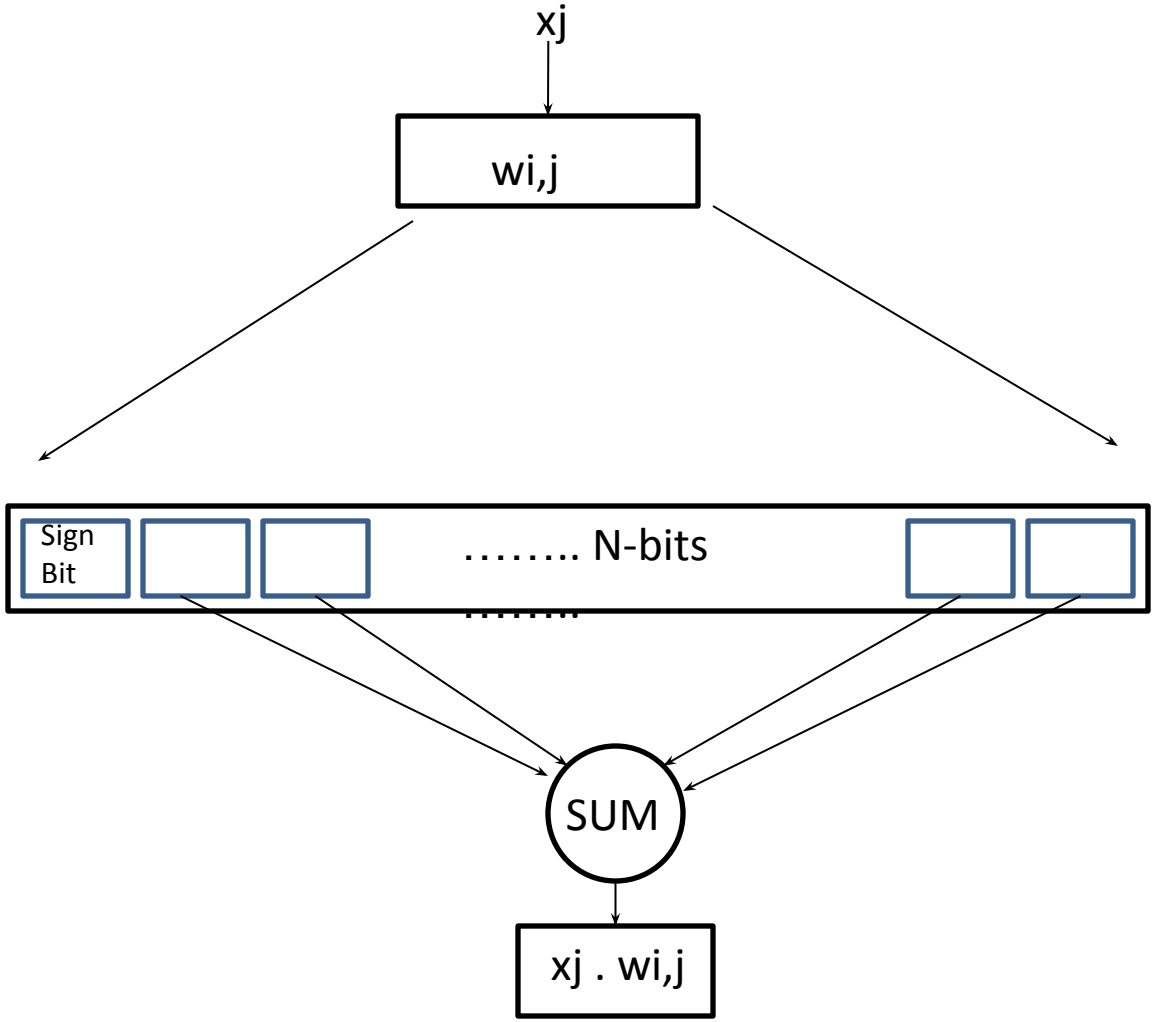
We decide to quantize the second layer input to 4 bits and the second layer weights/biases to 4 bits.

We note that the inputs to the first and second layer are non-negative, while the weights are negative. The way the quantization is done also affects the implementation. We use a mid-rise type quantization for the weights, allowing for differential operation. Hence, the binary value W, represents $15 - 2 \times W$.

Strategy To Handle Multi-Bit Weights:



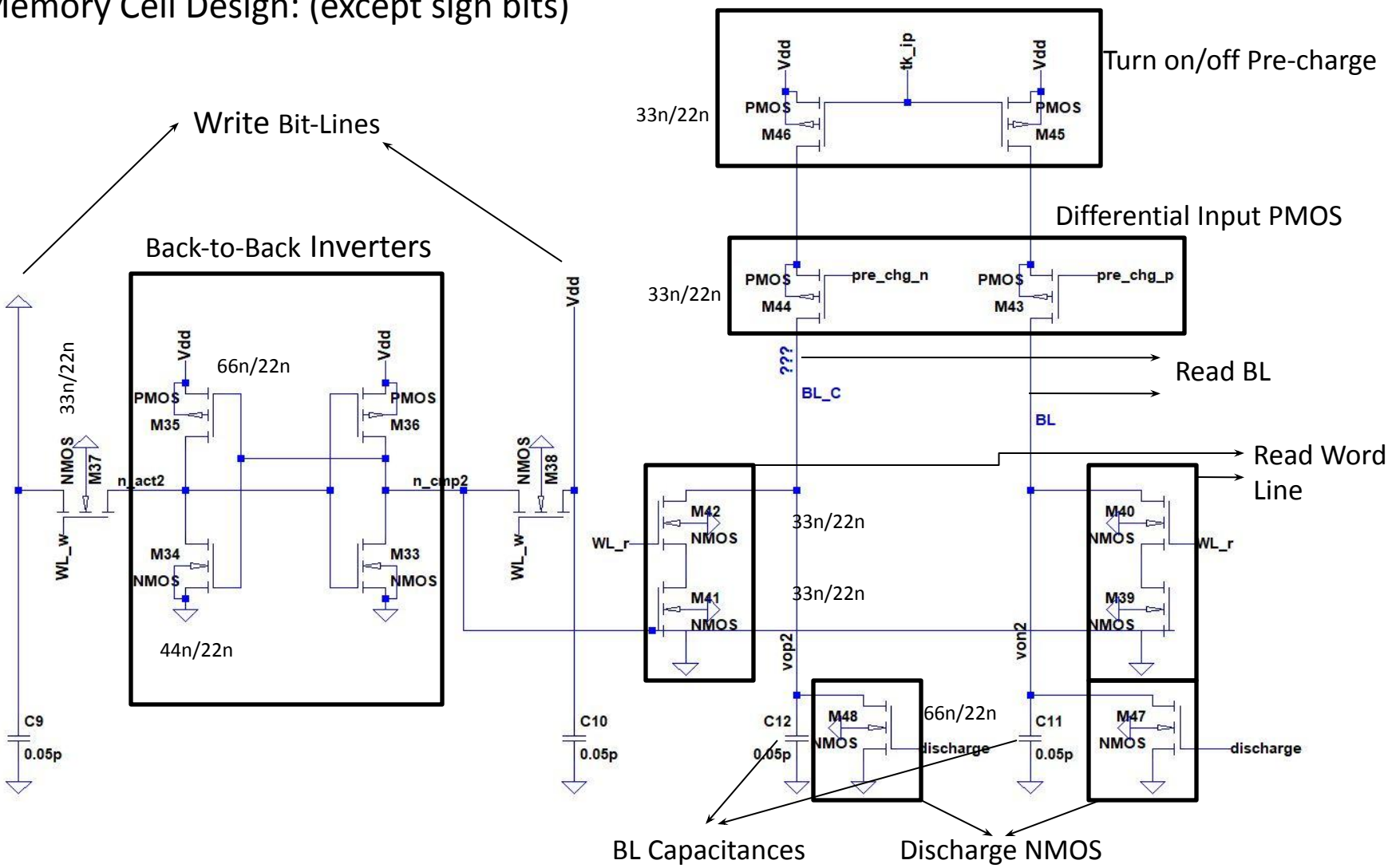
Strategy To Handle Multi-Bit Weights:



How IMC Can Be Performed?

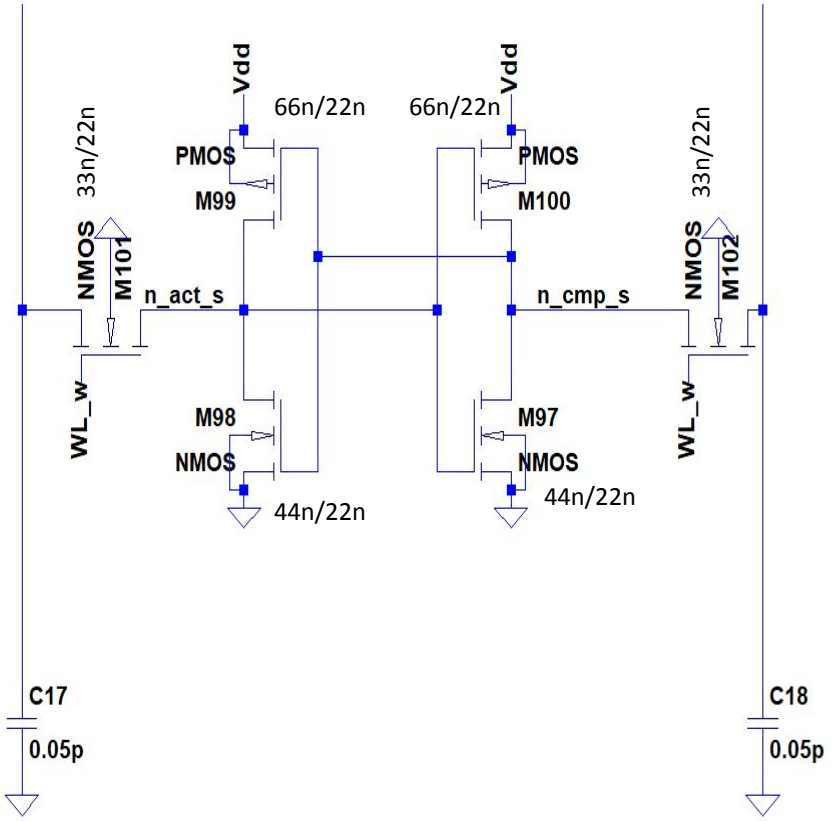
(As Proposed previously)

Memory Cell Design: (except sign bits)



How IMC Can Be Performed?

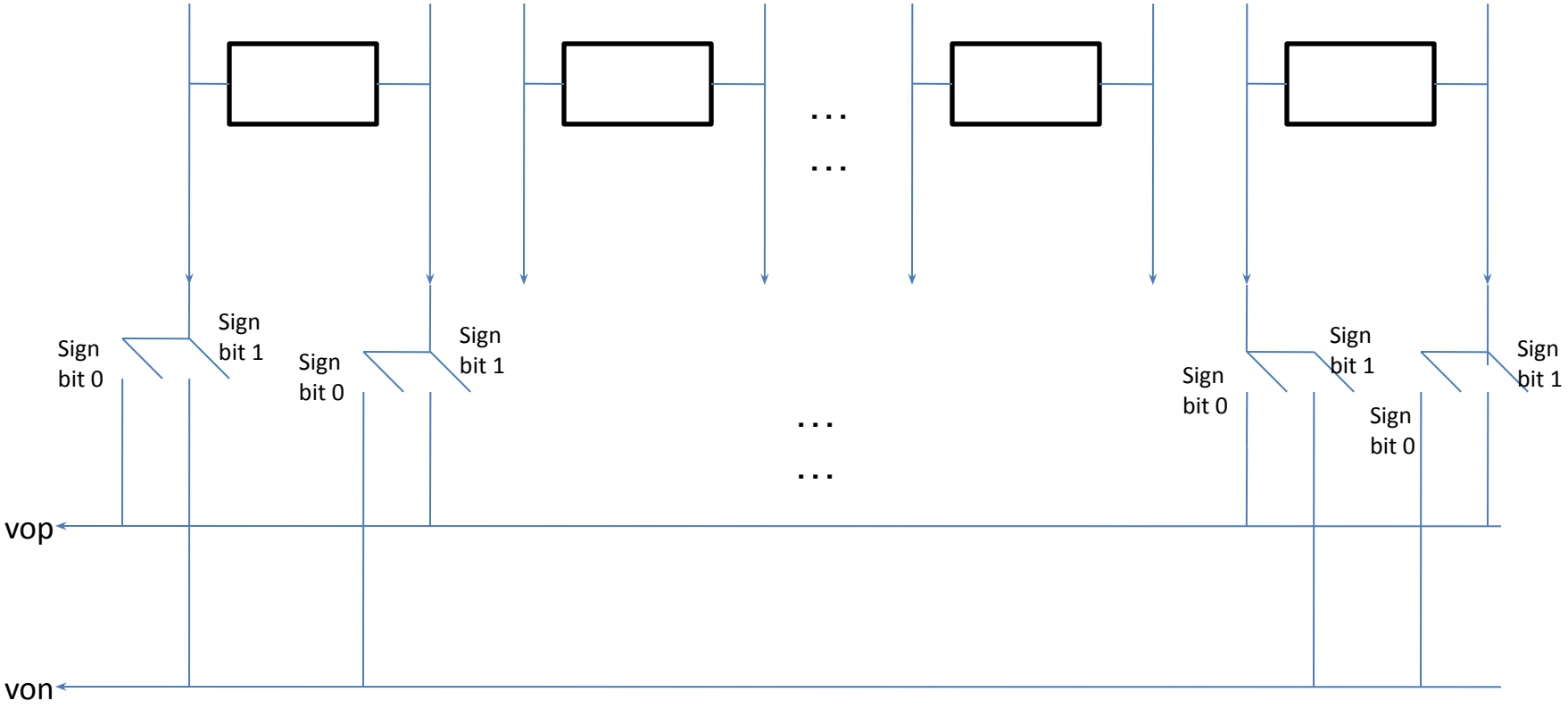
Memory Cell Design:
(To store the sign bit of the weight)



How IMC Can Be Performed?

- The previous slide shows the memory cell for each bit of the weight(except the sign bit).
- Each input (xj) is multiplied to the bits of the weight(wij) and then added in weighted manner to get the differential analog output(wij . xj).
- For weighted addition, the outputs for each bits are accumulated using a switched capacitor circuit between the bit-lines of each bit.
- After getting the multiplication result, the sign bit of the weight is used to decide the way output should be connected to the Global Bitline for accumulation.
- In the end, the bitlines are discharged by making ‘discharge’ signal as high.

How IMC Can Be Performed?



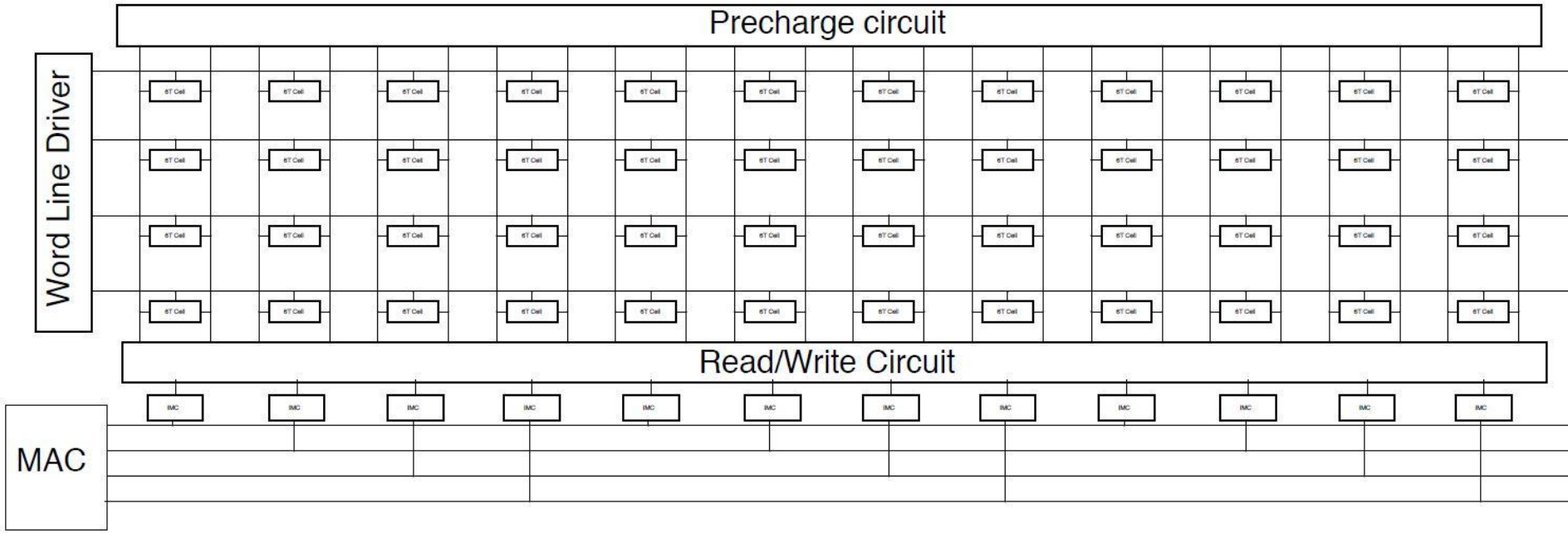
Issues with the Charge Sharing Based IMC Design:

- The time period of charge distribution increases directly with the Bit Line Capacitance (i.e. with the number of cell per bitline) : $T \propto C\_BL$
- The error in the output was increased due to the parasitics of pass-gates.
- To reduce the charge redistribution period, increase the number of pass-gates which means less layout friendly.
- The cell used in this case is an asymmetric non-conventional 10T cell.
- Since, in the given scenario, the charge is distributed across 784 capacitors, a high sensitivity ADC(~4mV resolution) is needed.
- It is quite susceptible to variations in Bit Line and hence, would increase the error.

Solution

- Adding up Currents instead of Charge Sharing for accumulation.
- Using 6T cell to reduce the area consumption.

Circuit Implementation



Multibit MAC Operation

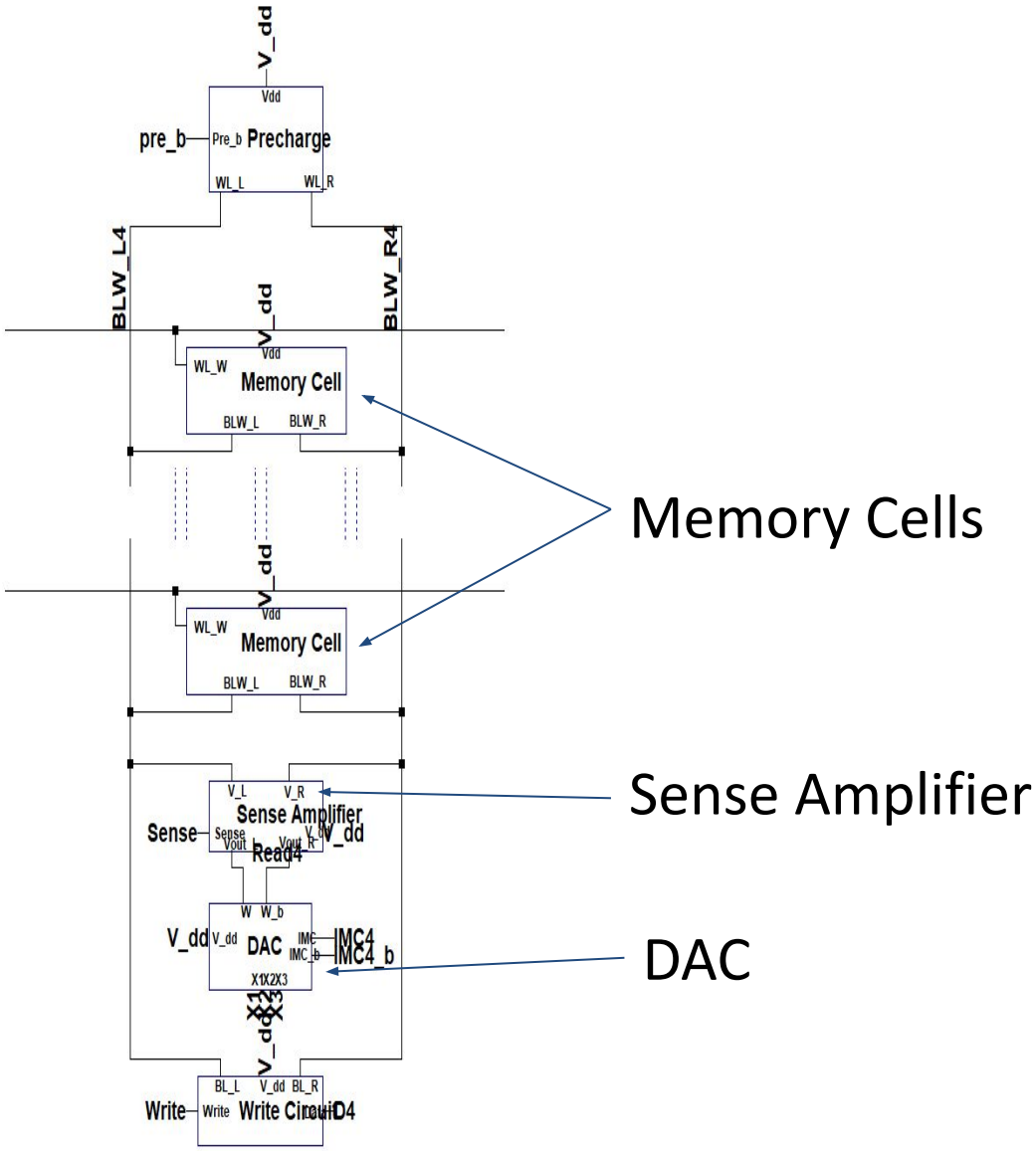
- We need to calculate $\sum_i X_i W_i$ for the MAC operation. X_i is implemented as its normal unsigned binary number, which ranges from $0-3$ (2 bit quantization).

  W_i is decomposed into its mid-rise quantization representation. Hence $W_{3i}W_{2i}W_{1i}W_{0i}$ represents $15-16\times W_{3i}-8\times W_{2i}-4\times W_{1i}-2\times W_{0i}$, where W_{ji} can be either 0 or 1.

  Since we have 4 bit weights, the MAC will implement 4 differential outputs $V_k=\sum_i X_i(1-2\times W_{ki})$.

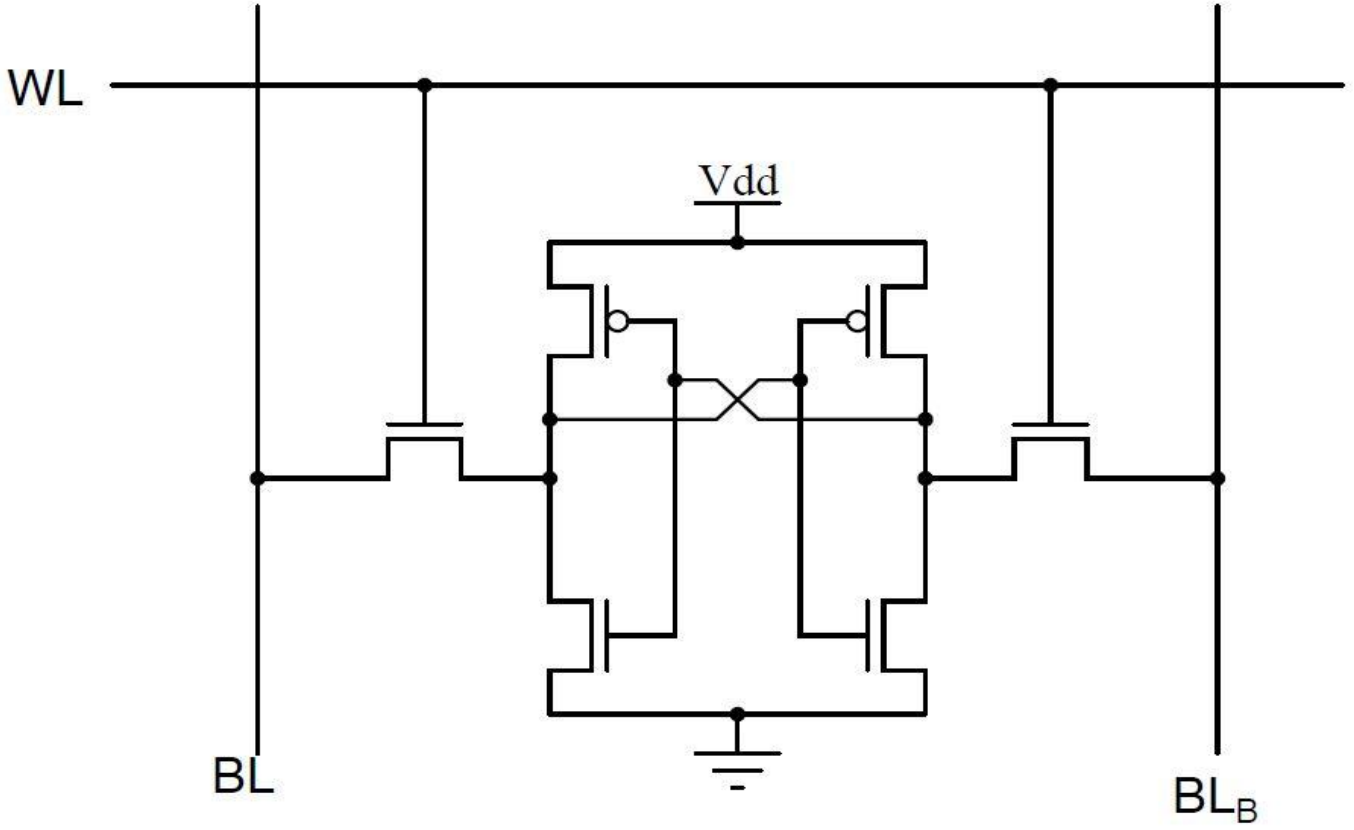
  The final output of the MAC operation is obtained by performing the digital operation of $\sum_{k=0}^{3} V_k 2^k$. This evaluates to $\sum_i X_i[(1-2\times W_{0i})+2\times(1-2\times W_{1i})+4\times(1-2\times W_{2i})+8\times(1-2\times W_{3i})]$, which simplifies to$\sum_i X_i[15-16\times W_{3i}-8\times W_{2i}-4\times W_{1i}-2\times W_{0i}]=\sum_i X_i W_i$, which is the required output.

Memory Array Structure



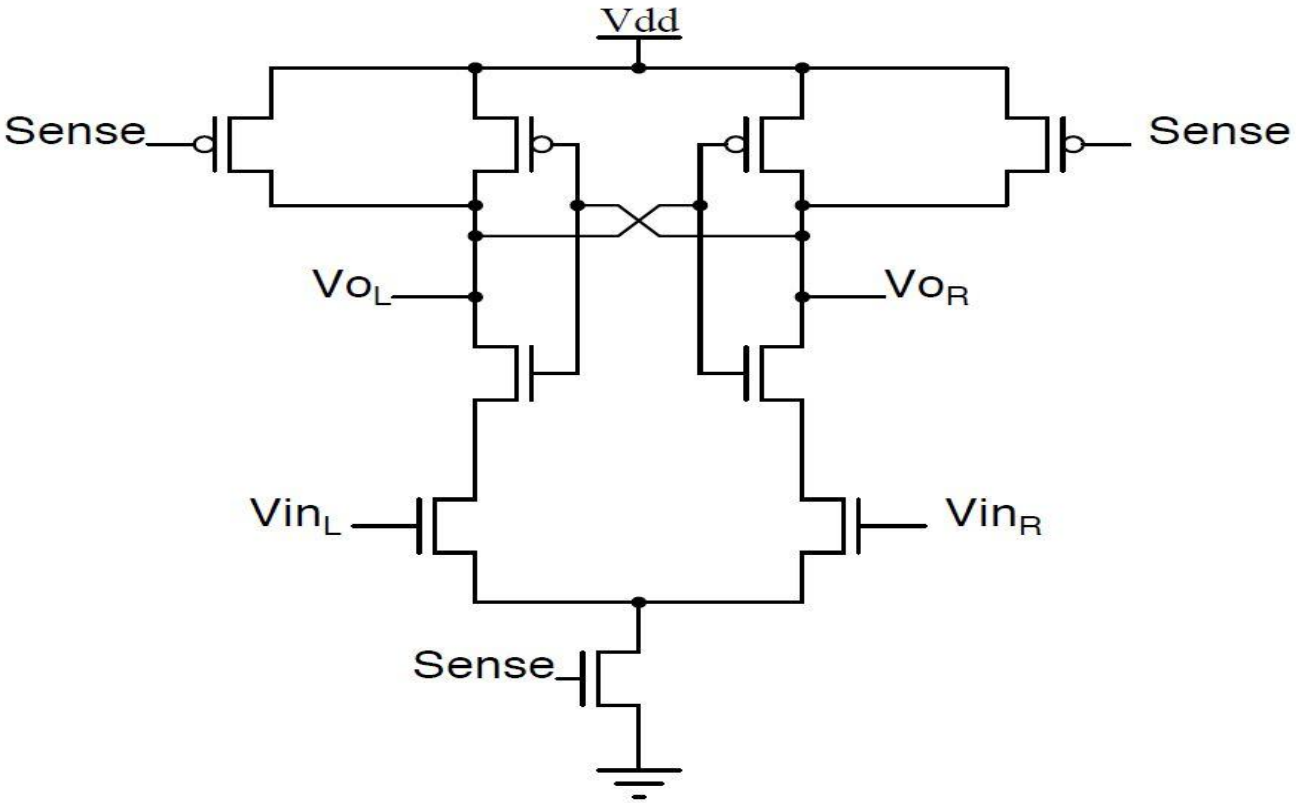
Memory Cell

Standard 6T Memory Cell:



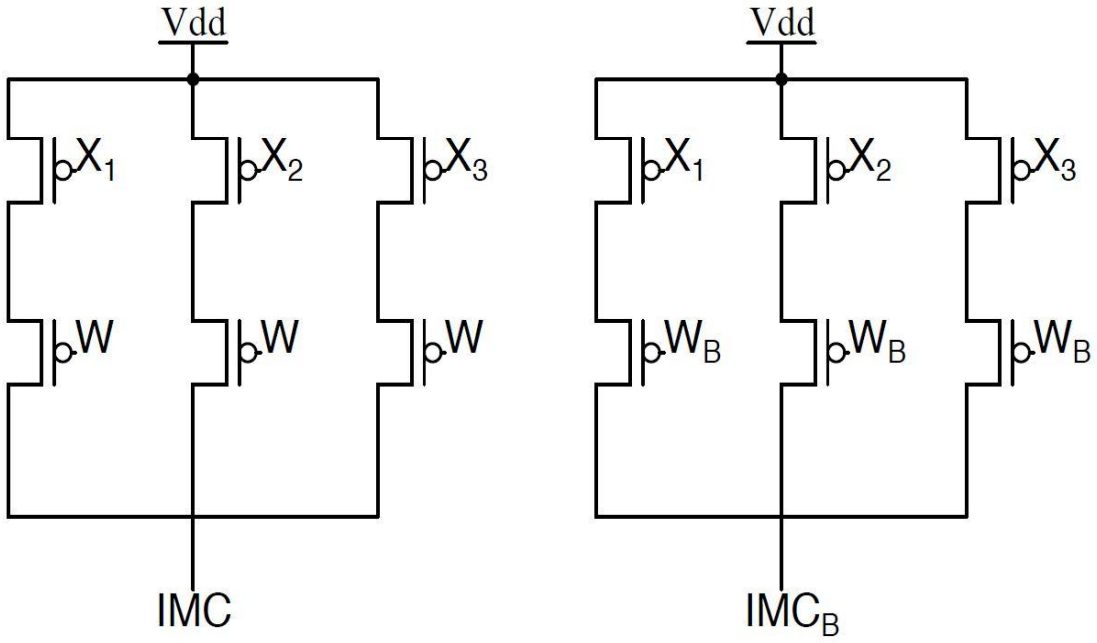
Peripherals

Sense Amplifier: A strong ARM latch based sense amplifier is used for reading from the memory array



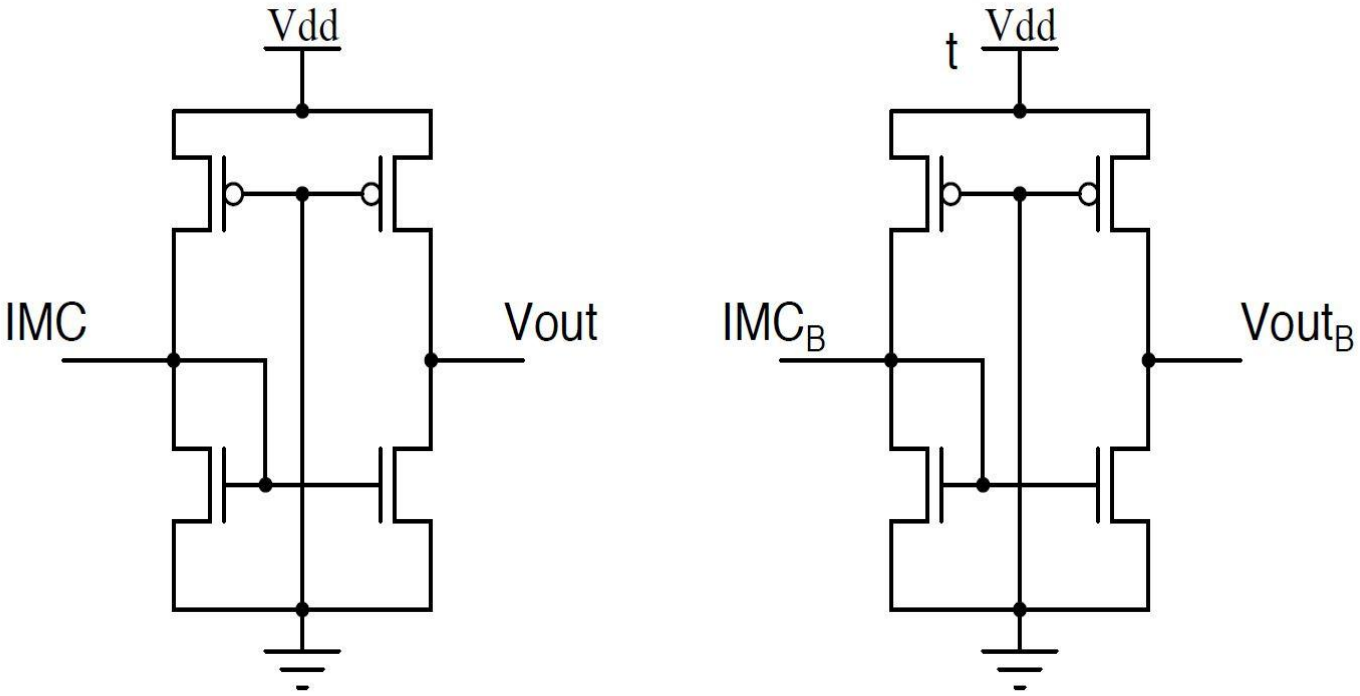
Peripherals

Multiply operation : Converts code into current



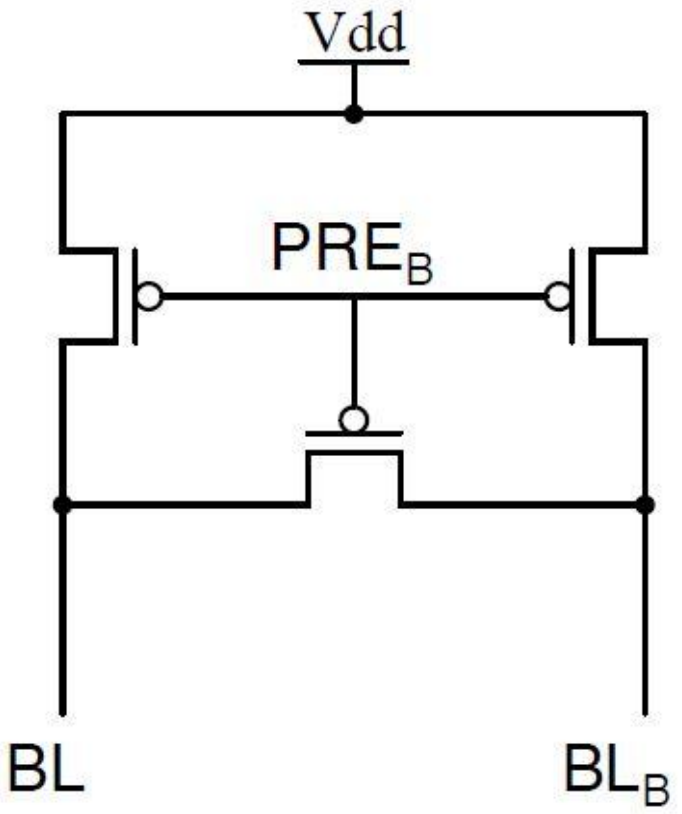
Peripherals

Accumulation Circuit: Converts current to voltage as input to the ADC



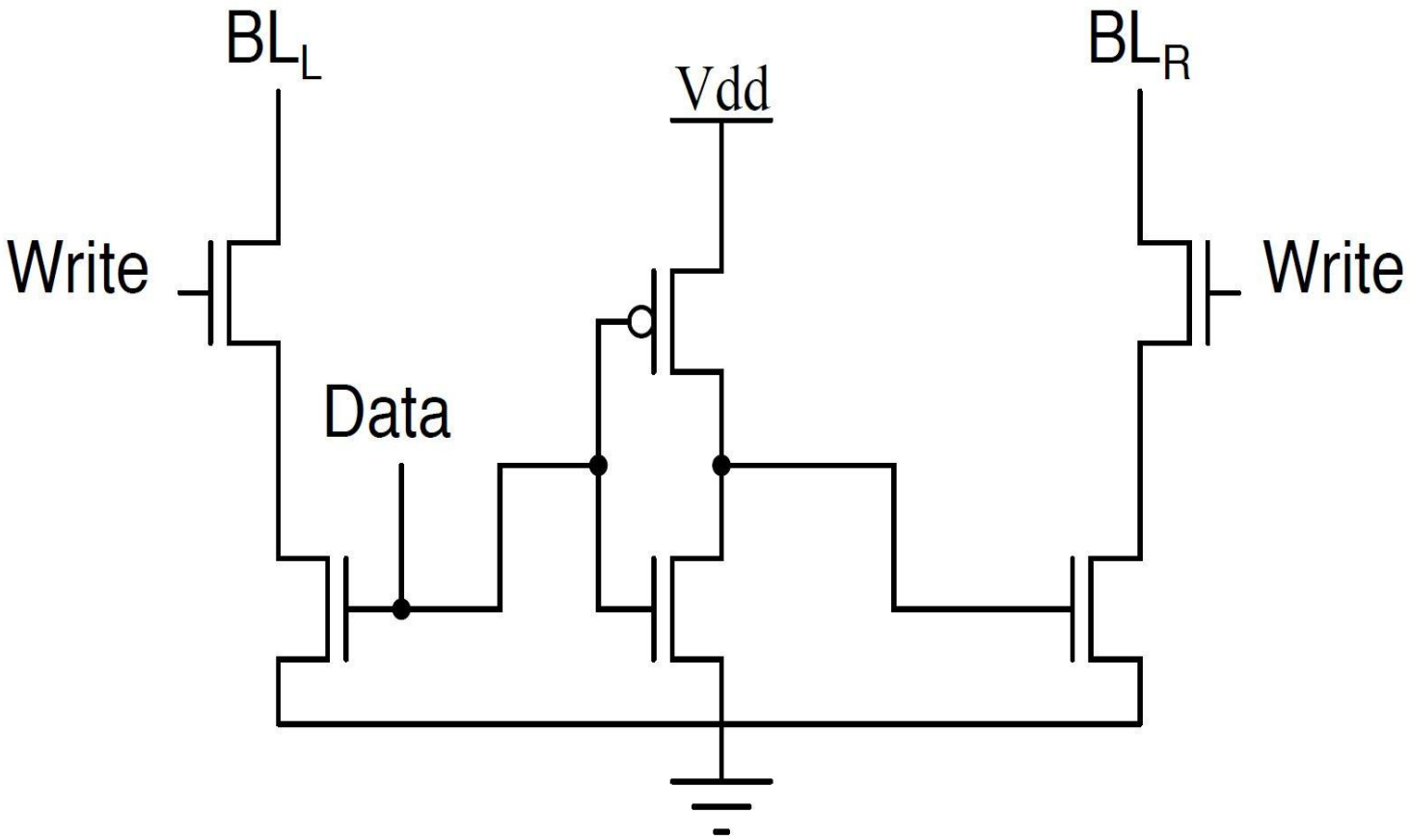
Peripherals

Pre-Charge Circuit: A standard circuit topology is used for preconditioning and equalization.



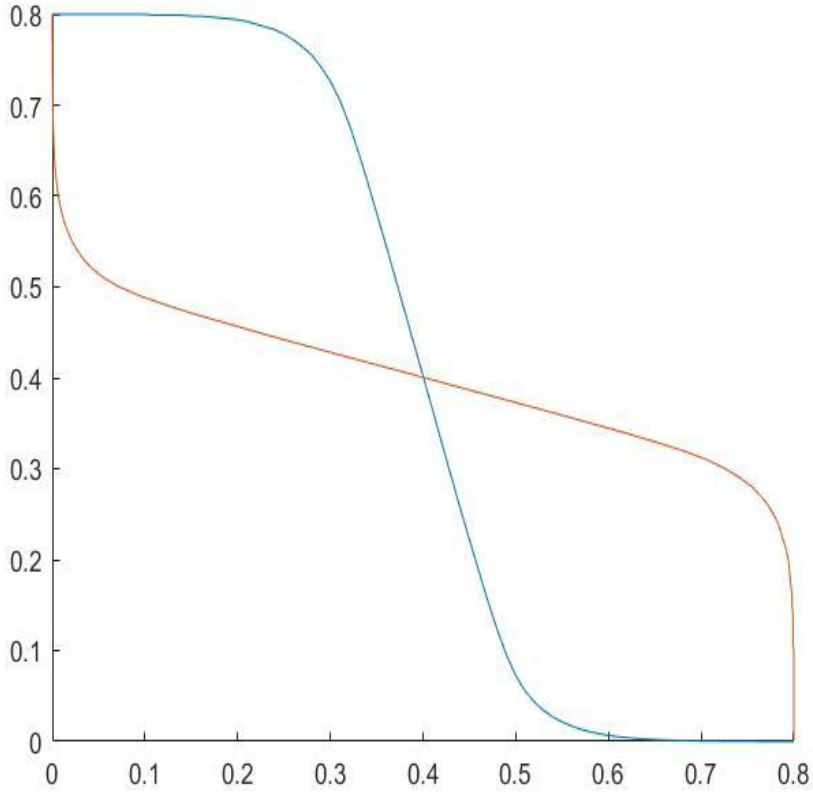
Peripherals

Write Circuit: A standard write driver is used for writing the data onto the bit lines



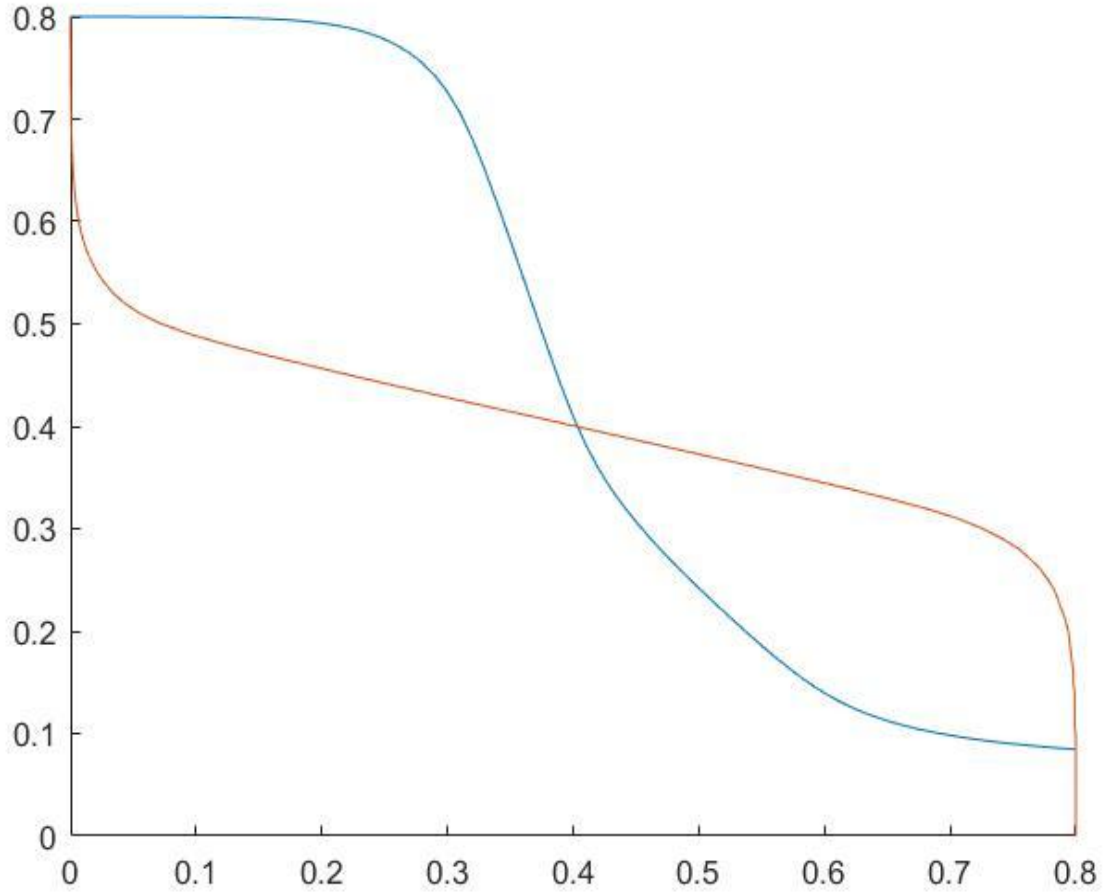
Results | Cell Characteristics

Hold Margin:



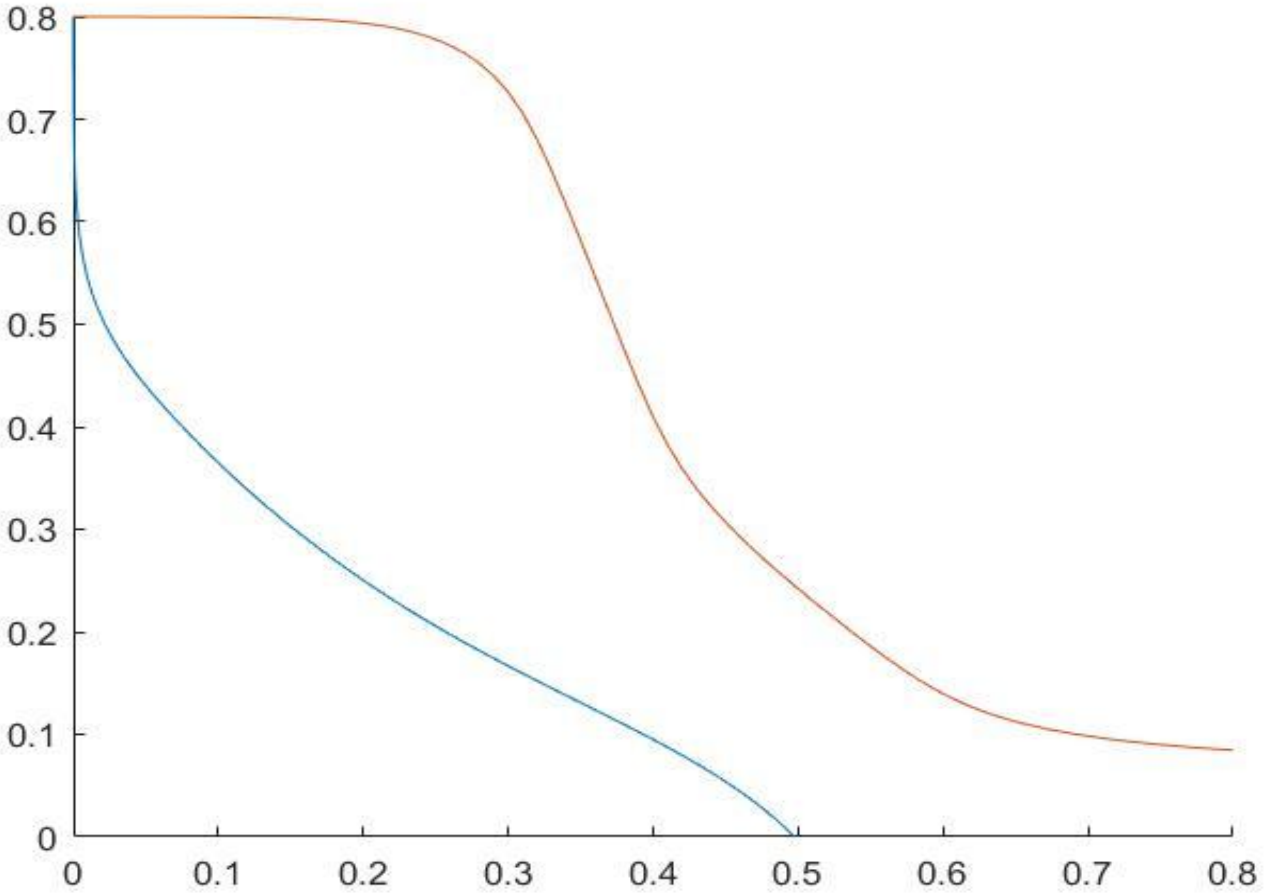
Results | Cell Characteristics

Read Margin:



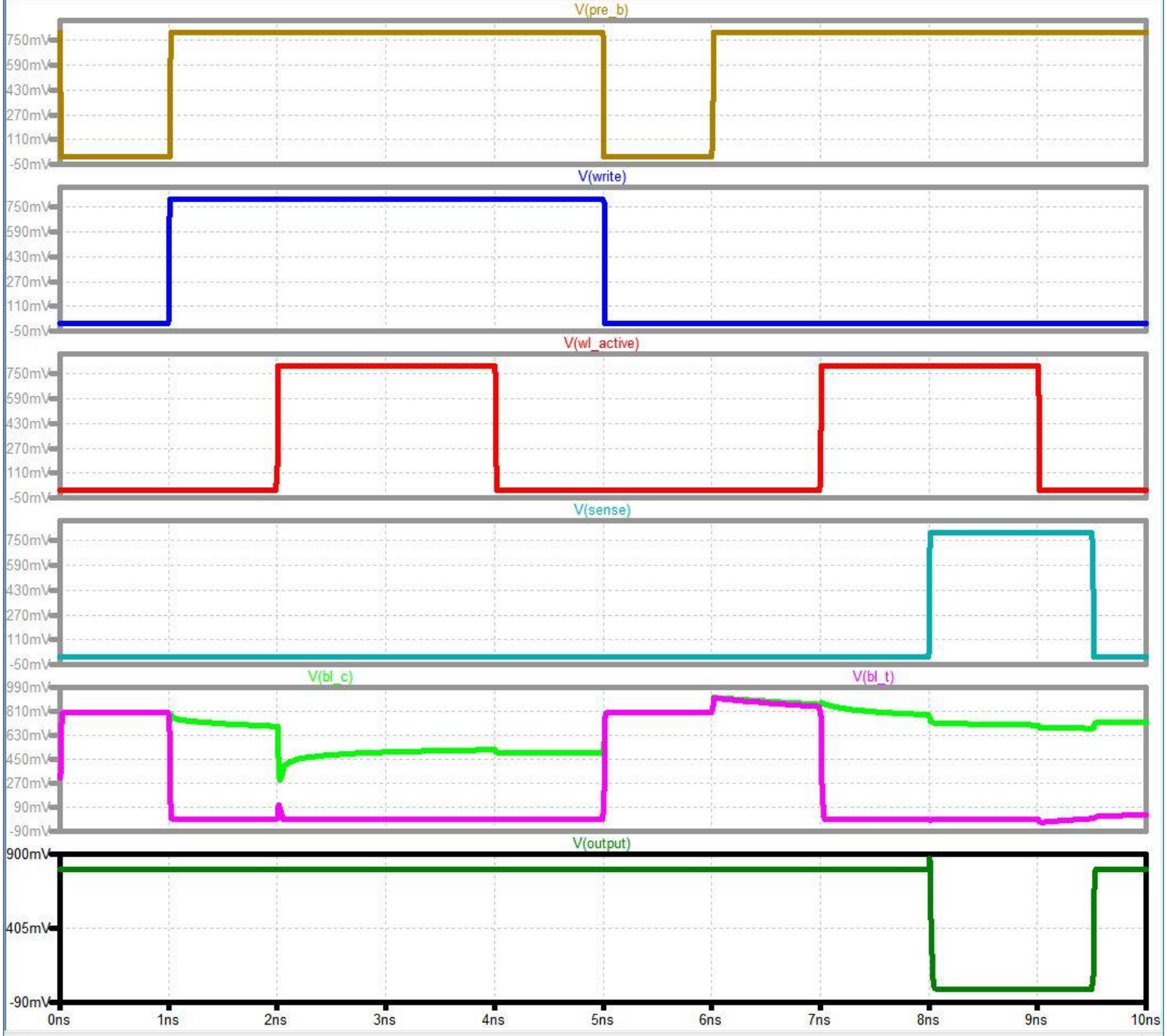
Results | Cell Characteristics

Write Margin:



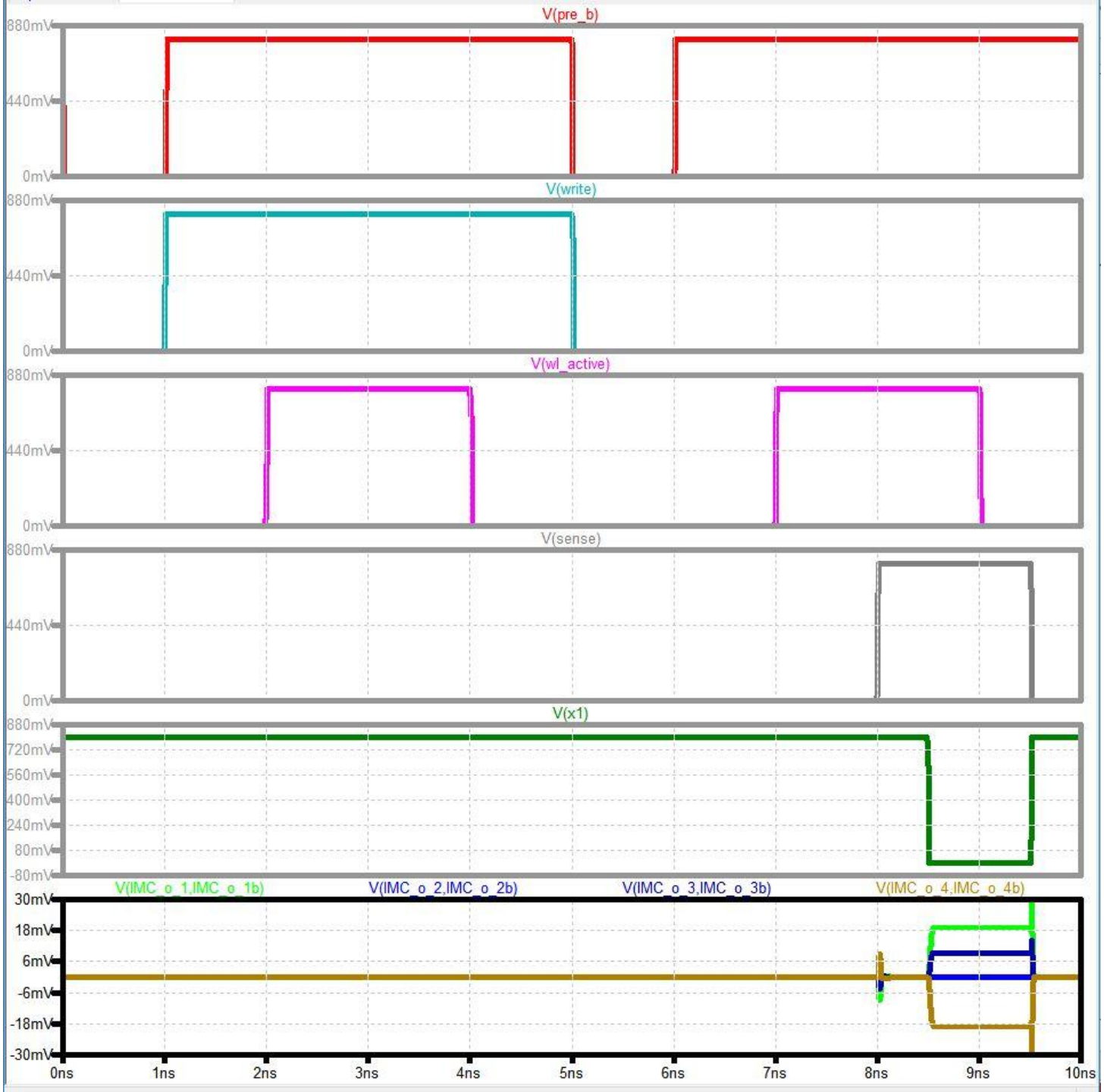
Circuit Operation | Read & Write

A target of 5ns read/write time was chosen (200MHz). The clocking scheme for read and write is illustrated, where 0 is written to and then read from the same cell.

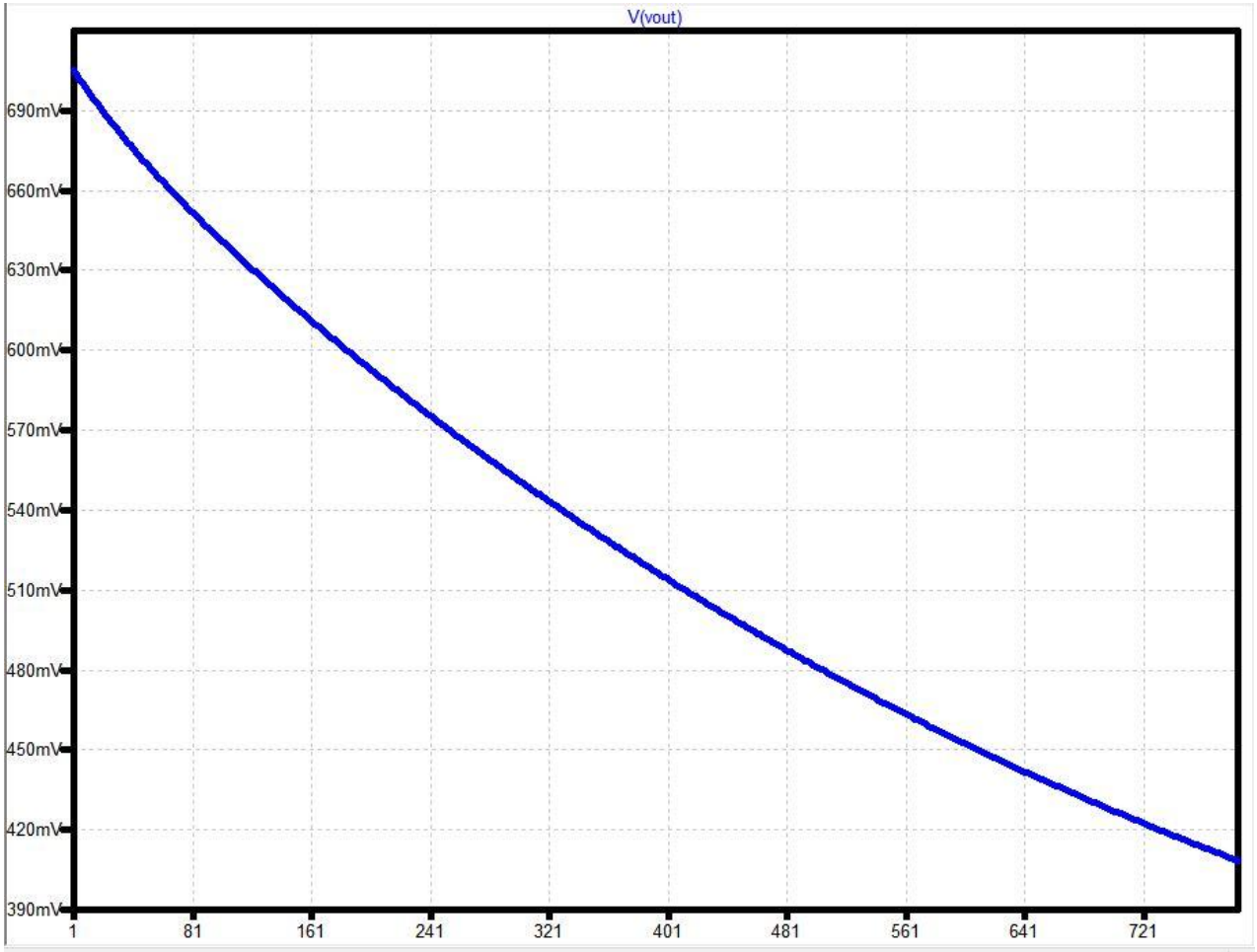

V(pre_b)
750mV
590mV
430mV
270mV
110mV
-50mV
V(write)
750mV
590mV
430mV
270mV
110mV
-50mV
V(wl_active)
750mV
590mV
430mV
270mV
110mV
-50mV
V(sense)
750mV
590mV
430mV
270mV
110mV
-50mV
V(bl_c)
V(bl_t)
990mV
810mV
630mV
450mV
270mV
90mV
-90mV
V(output)
900mV
405mV
-90mV
0ns
1ns
2ns
3ns
4ns
5ns
6ns
7ns
8ns
9ns
10ns

Circuit Operation | IMC

A target of 5ns per output (of 786 MAC operations) was chosen (156:8GMACs). The clocking scheme for a single write and IMC is illustrated below.


V(pre_b)
880mV
440mV
0mV
V(write)
880mV
440mV
0mV
V(wl_active)
880mV
440mV
0mV
V(sense)
880mV
440mV
0mV
V(x1)
880mV
720mV
560mV
400mV
240mV
80mV
-80mV
V(IMC_o_1,IMC_o_1b)
V(IMC_o_2,IMC_o_2b)
V(IMC_o_3,IMC_o_3b)
V(IMC_o_4,IMC_o_4b)
30mV
18mV
6mV
-6mV
-18mV
-30mV
0ns
1ns
2ns
3ns
4ns
5ns
6ns
7ns
8ns
9ns
10ns

IMC Characterization



Performance Estimation

Metric	Value	Assumptions/Notes
Cell Type	Standard $6T$ SRAM Cell	
Energy	$1.34pJ$ per Image	Excluding energy for SRAM reads + ADC/Digital Logic
Latency	$5.5\mu s$ per Image	$550ns$ if ideal ADC/Digital Logic present
Accuracy	95.87%	Only Layer 1 is computed using IMC

Variability Analysis

We note that the MAC operation depends on works on encoding X as $\bar{X}_1 + \bar{X}_2 + \bar{X}_3$, where X_1, X_2, X_3 are 1 or 0. One way to combat variations is to recompute the same MAC operation with a different encoding for each X, and take the most common output.

We further note that to reduce the power consumption in the current accumulation, we have used large lengths in the mosfets. This will reduce short channel effects and variations due to RDF.

Implementation Considerations

We note that the biases can also be stored in the memory cell, and a dummy input $X = 1$ can be given to that column, and so the current accumulation operation will take care of adding the bias also.

We also note that for the second stage, the input is scaled due to the digital code to current conversion and then back to voltage. Hence, an appropriate full scale quantization range must be chosen to get the required accuracy. We have chosen $100mV$ full scale range.

Further, since the output of the IMC is differential, the relu activation can be easily implemented in hardware using a mux that detects if the output is positive or negative.

Area Considerations

The MAC operation required many pmos current sources which supply a small current and a large nmos current sink that sinks a large current. Hence, we have used large length for the pmos and large width for the nmos. This area can be amortized over 100 rows (pmos) and 784 columns (nmos). However, if the technology allows for low V_T and high V_T devices, they can be used and area can be saved. Other than that, no further area or complexity is overhead is added to the 6T SRAM array, as the IMC only works on the output of the array.

ADC Considerations

We note that we are assuming a voltage input ADC. However, there are ADC architectures that internally convert the voltage to a current and then process it to get the output. Since we are computing MAC using current accumulation, we can directly feed the current output into such ADC, saving power and area.

Conclusion

- A scalable IMC circuit was designed and simulated
- A model for the IMC was simulated and the accuracy of the computation was evaluated
- Differential signalling was used to combat effects of noise
- Various aspects including implementation, variability and multi-bit handling issues were discussed

Thank You