MSISLAB

NEURAL NETWORK PROCESSING UNIT CHIP DESIGNS

2020. 2

MSIS Lab

Chungbuk National University, South Korea

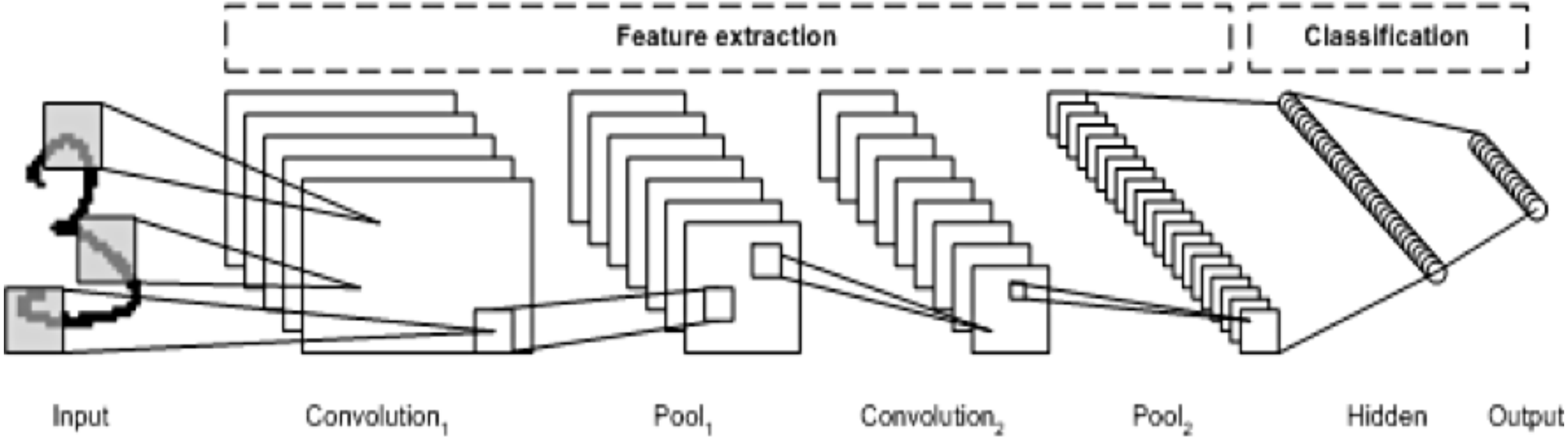
Neural Network Processing SoC Designs

- Current Research
  - CNN (Convolutional Neural Network) Accerelator Chip
  - SNN (Spiking Neural Network) Mixed-Signal Chip
- On-Going & Future Research
  - Reconfigurable CNN Chip
  - Self-Learning CNN chip
  - Federated Learning & Aggregation AI System

Introduction to Convolutional Neural Network (CNN)

❖ CNN Model for Classification of MNIST data set

- Extract and learn features of an image with multiple filters
- Pooling layer to collect and enhance features of the extracted image
- Classify images using neural network



CNN Model Compression with High Accuracy

❖ Comparison of Conventional CNN and Compressed CNN

- Reducing the number of Conv. Layers
- Minimizing the number of required Conv. Filters (Kernels)
- Minmizing the Weight & Data resolution from 32 bits to 8 bits with little sacrifice of accuracy
- Target Classification data set : MNIST Data Set

Original CNN model

Layers	Size
Convolution	32 filters
Pooling	2*2 max pooling
Convolution2	32 filters
Pooling2	2*2 max pooling
Fully connected layer1	10
Fully connected layer2	14

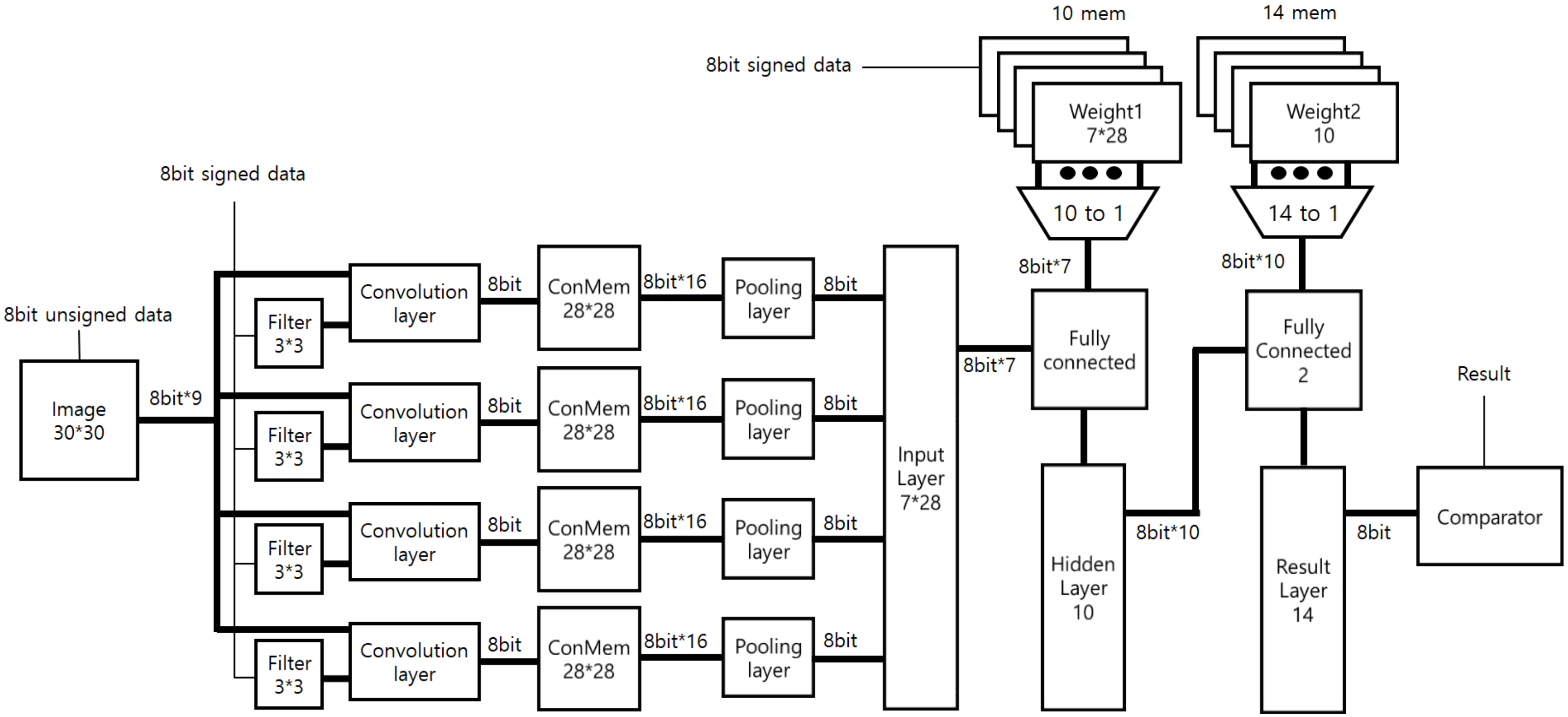
Compressed CNN model

Layers	Size
Convolution	4 filters
Pooling	4*4 max pooling
Fully connected layer1	10
Fully connected layer2	14

CNN Accelerator Chip Design

❖ Classification Engine for MNIST data set

- Minimize the power and size overhead by 100 times
- Maintain the accuracy above 94%

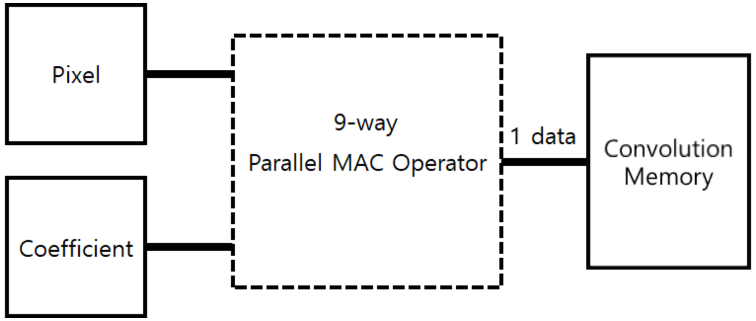


CNN Accelerator Chip Design

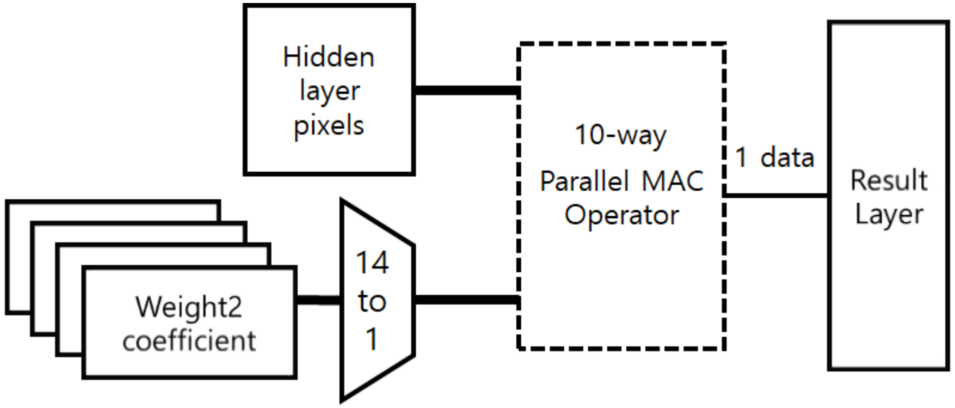
❖ Classification Engine for MNIST data set

- Conv Layer, Maxpooling Layer, FC Layer Architectures

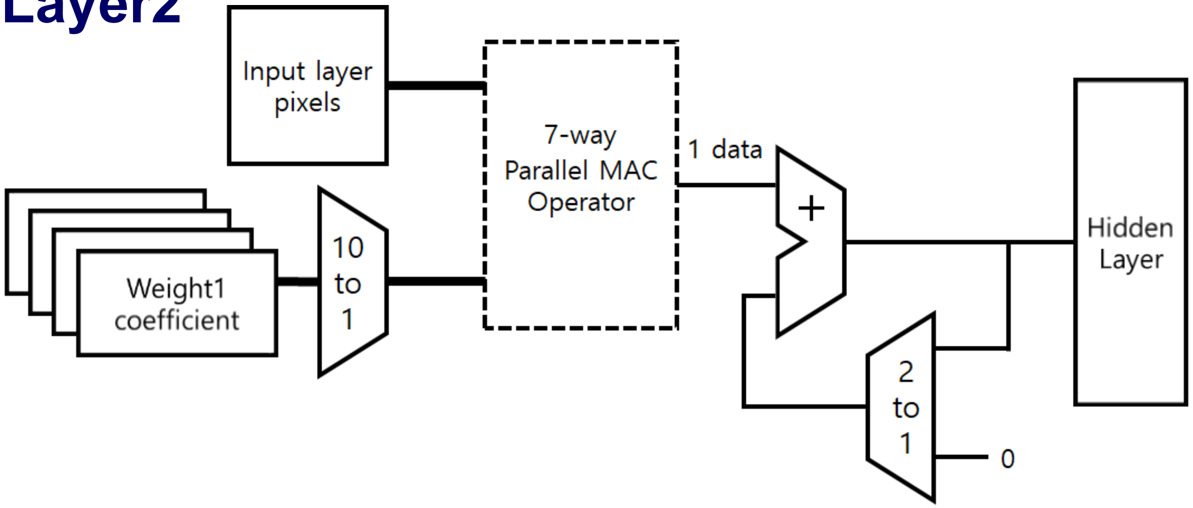
❖ Conv Layer



❖ FC Layer1



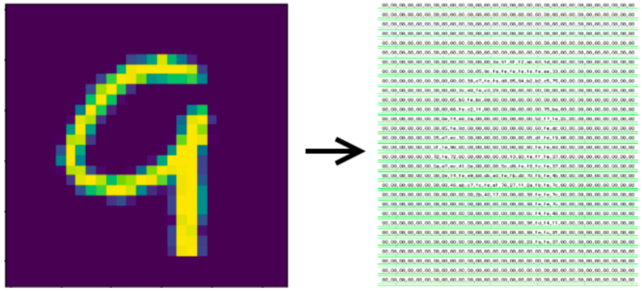
❖ FC Layer2



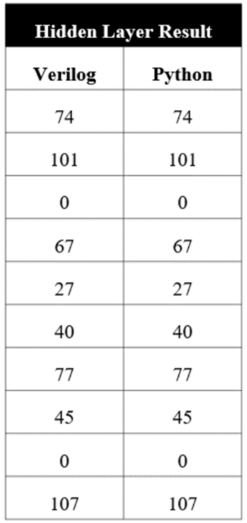
CNN Accelerator Chip Design

❖ Classification Engine for MNIST data set

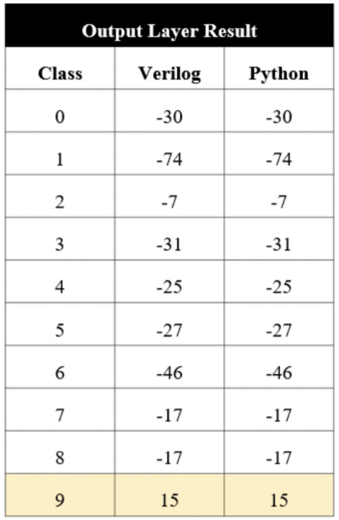
	Convo-lution	Pool-ing	Fully connected1	Fully connected2	Total
Adder	32	0	6	9	47
Multiplier	36	0	7	10	53
Register	16	0	5	11	32



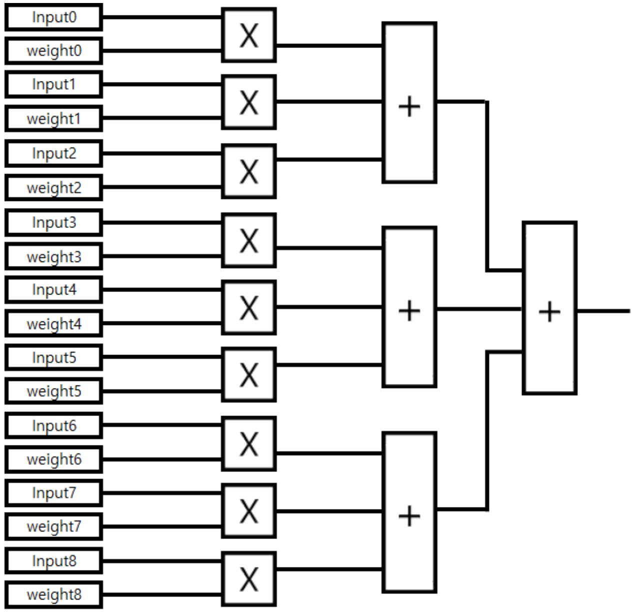
(a)

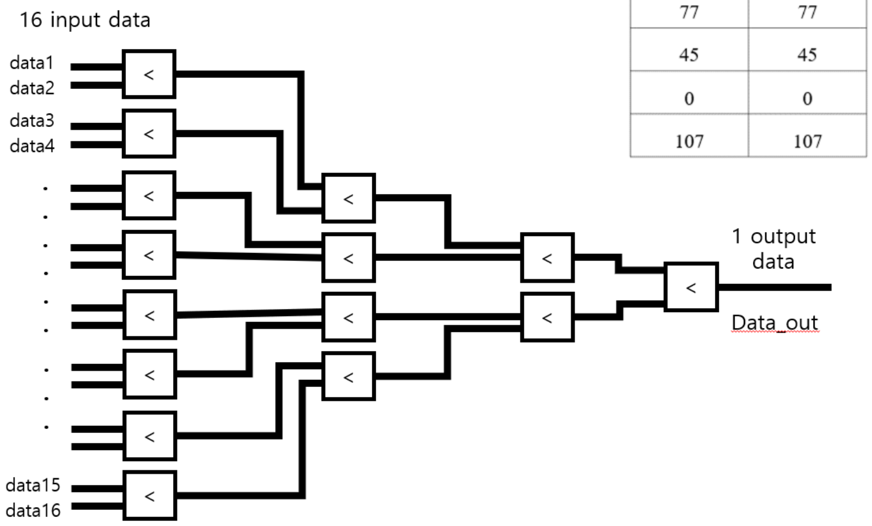


Hidden Layer Result	
Verilog	Python
74	74
101	101
0	0
67	67
27	27
40	40
77	77
45	45
0	0
107	107



Output Layer Result		
Class	Verilog	Python
0	-30	-30
1	-74	-74
2	-7	-7
3	-31	-31
4	-25	-25
5	-27	-27
6	-46	-46
7	-17	-17
8	-17	-17
9	15	15





Chip Size Reduction Result

❖ Size Comparison Between Original CNN vs Compressed CNN Architectures

Uncompressed Original CNN model

	Adder	Multiplier	Memory	Input bit
Convolution	32*8	32*9	32*4	32
Pooling	0	0	32*3	32
Fully connected layer1	223	224	112	32
Fully connected layer2	9	10	11	32
Total	488	522	347 Words x 32 bits = 11,104 bits	

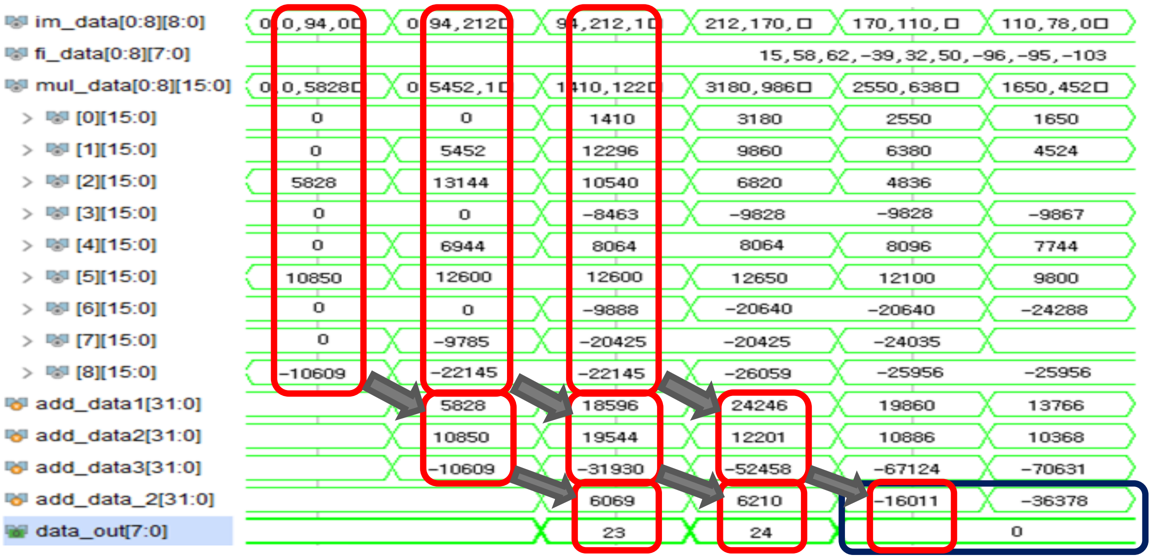
Compressed Optimized CNN model

	Adder	Multiplier	Memory	Input bit
Convolution	4*8	4*9	4*4	8
Pooling	0	0	4*15	8
Fully connected layer1	6	7	5	8
Fully connected layer2	9	10	11	8
Total	47	53	92 Words x 8 bits = 732 bits	
Reduction	90%	89.8%	93.4%	

Simulation Results of CNN Chip Design

(Full Chip Verilog Simulation With MNIST Images)

❖ Operation of Convolution Layer

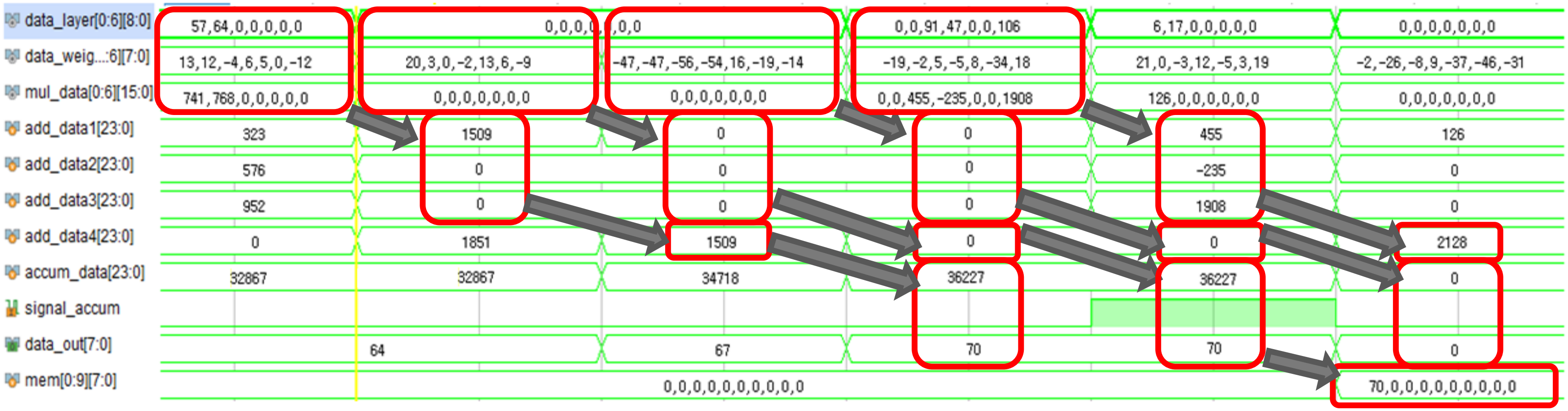


im_data[0:8][8:0] | 0 | 0,94,0 | 0 | 94,212 | 94,212,1 | 212,170,□ | 170,110,□ | 110,78,0□
fi_data[0:8][7:0] | 15,58,62,-39,32,50,-96,-95,-103
mul_data[0:8][15:0] | 0 | 0,5828□ | 0 | 5452,1□ | 1410,122□ | 3180,986□ | 2550,638□ | 1650,452□
[0][15:0] | 0 | 0 | 1410 | 3180 | 2550 | 1650
[1][15:0] | 0 | 5452 | 12296 | 9860 | 6380 | 4524
[2][15:0] | 5828 | 13144 | 10540 | 6820 | 4836
[3][15:0] | 0 | 0 | -8463 | -9828 | -9828 | -9867
[4][15:0] | 0 | 6944 | 8064 | 8064 | 8096 | 7744
[5][15:0] | 10850 | 12600 | 12600 | 12650 | 12100 | 9800
[6][15:0] | 0 | 0 | -9888 | -20640 | -20640 | -24288
[7][15:0] | 0 | -9785 | -20425 | -20425 | -24035
[8][15:0] | -10609 | -22145 | -22145 | -26059 | -25956 | -25956
add_data1[31:0] | 5828 | 18596 | 24246 | 19860 | 13766
add_data2[31:0] | 10850 | 19544 | 12201 | 10886 | 10368
add_data3[31:0] | -10609 | -31930 | -52458 | -67124 | -70631
add_data_2[31:0] | 6069 | 6210 | -16011 | -36378
data_out[7:0] | 23 | 24 | 0

Change 8bit

Relu

❖ Pipelined Operation of Fully-Connected Layer1



data_layer[0:6][8:0] | 57,64,0,0,0,0,0 | 0,0,0,0,0,0,0 | 0,0,91,47,0,0,106 | 6,17,0,0,0,0,0 | 0,0,0,0,0,0,0
data_weig...:6][7:0] | 13,12,-4,6,5,0,-12 | 20,3,0,-2,13,6,-9 | -47,-47,-56,-54,16,-19,-14 | -19,-2,5,-5,8,-34,18 | 21,0,-3,12,-5,3,19 | -2,-26,-8,9,-37,-46,-31
mul_data[0:6][15:0] | 741,768,0,0,0,0,0 | 0,0,0,0,0,0,0 | 0,0,0,0,0,0,0 | 0,0,455,-235,0,0,1908 | 126,0,0,0,0,0,0 | 0,0,0,0,0,0,0
add_data1[23:0] | 323 | 1509 | 0 | 0 | 455 | 126
add_data2[23:0] | 576 | 0 | 0 | 0 | -235 | 0
add_data3[23:0] | 952 | 0 | 0 | 0 | 1908 | 0
add_data4[23:0] | 0 | 1851 | 1509 | 0 | 0 | 2128
accum_data[23:0] | 32867 | 32867 | 34718 | 36227 | 36227 | 0
signal_accum
data_out[7:0] | 64 | 67 | 70 | 70 | 0
mem[0:9][7:0] | 0,0,0,0,0,0,0,0,0,0 | 70,0,0,0,0,0,0,0,0,0

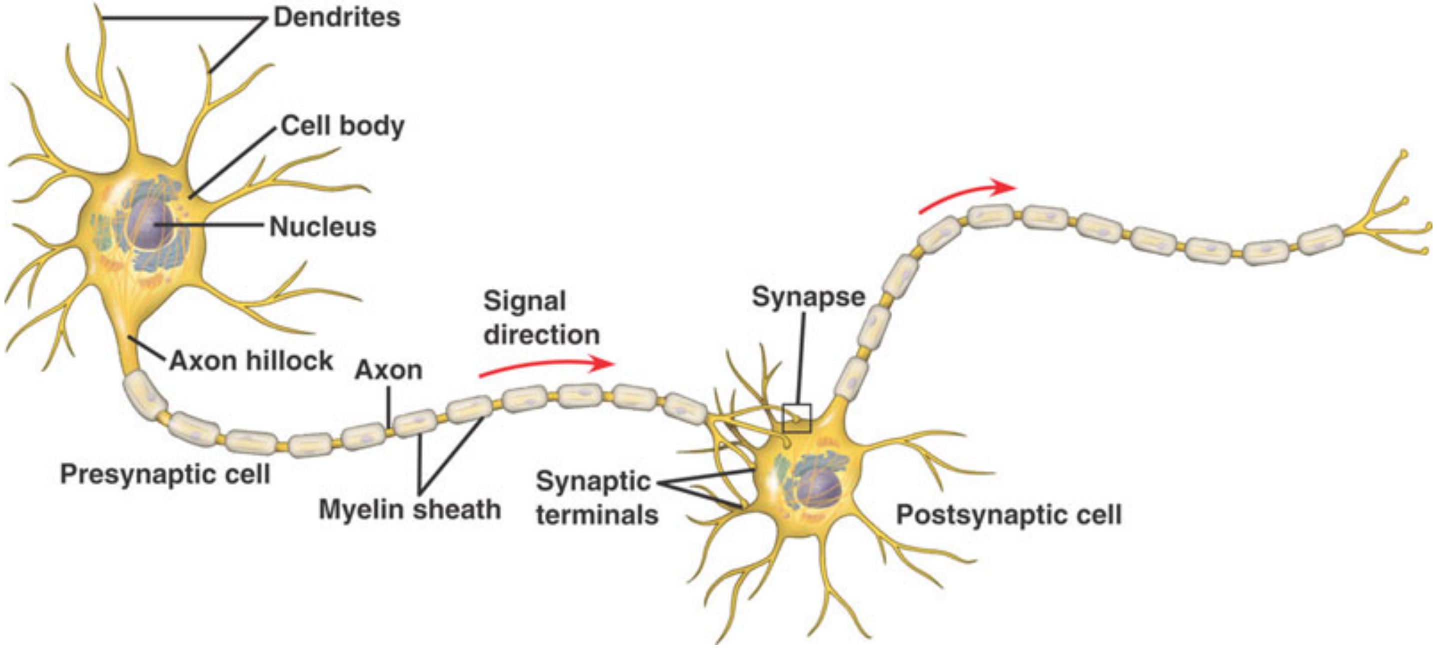
Result of Speed Enhancement

- Cycle time comparison with various architecture

	CPU with Original CNN model	CPU with Compressed CNN model	Accelerator chip with Compressed CNN model (Proposed Chip)	Pipelined Layer CNN Accelerator (Ongoing Design)
Convolution	426496	53312	784+3	196
Pooling	18816	2940	49	49
Fully connected layer1	125430	3910	280+4	140
Fully connected layer2	266	266	14+5	14+5
Comparator	14	14	14	14
Total cycles per frame	571022 clock	60442 clock	1153 clock	196 clock

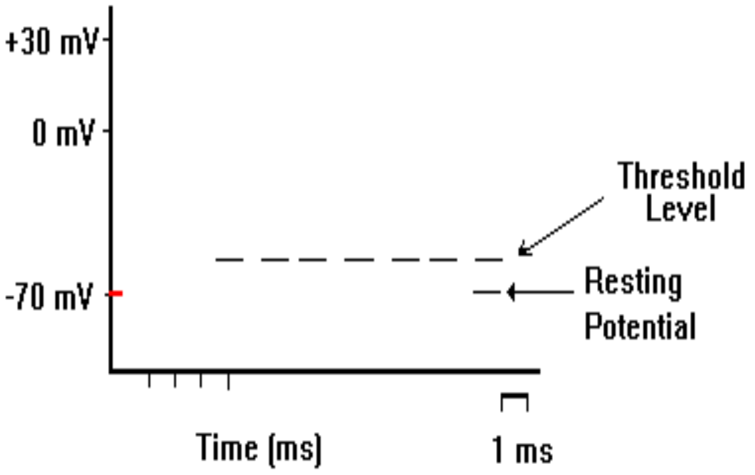
Background of SNN (Spiking Neural Network)

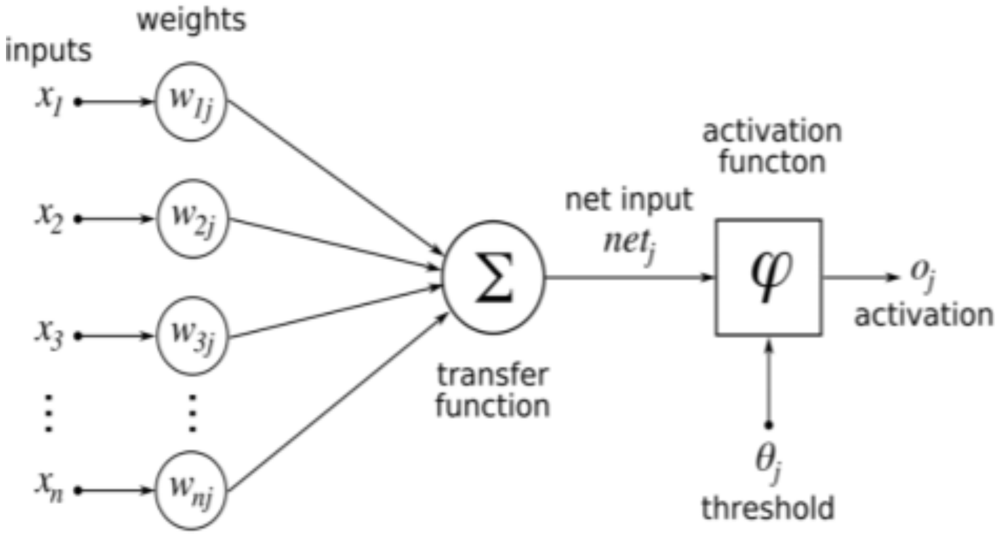
- Animal brain has a massively parallel structure of Neurons interconnected through Synapses
- Synapses
  - Can be implemented with a Storage or Memory with Communication Interface
- Neurons
  - Can be implemented with accumulation and comparison processing circuits



SNN (Spiking Neural Network) Building Block

- Circuit Architecture of SNN's blocks
  - Implementing Synapse and Neuron by Mixed-Signal circuits
- Artificial Synapse Circuit
  - Modulating the input spike rates by a weight value (trained parameter)
- Artificial Neuron Circuit
  - One or more Inputs: Each input can carry a different no. of spikes coming from presynaptic neurons
  - One or more Outputs: Activation function with a threshold

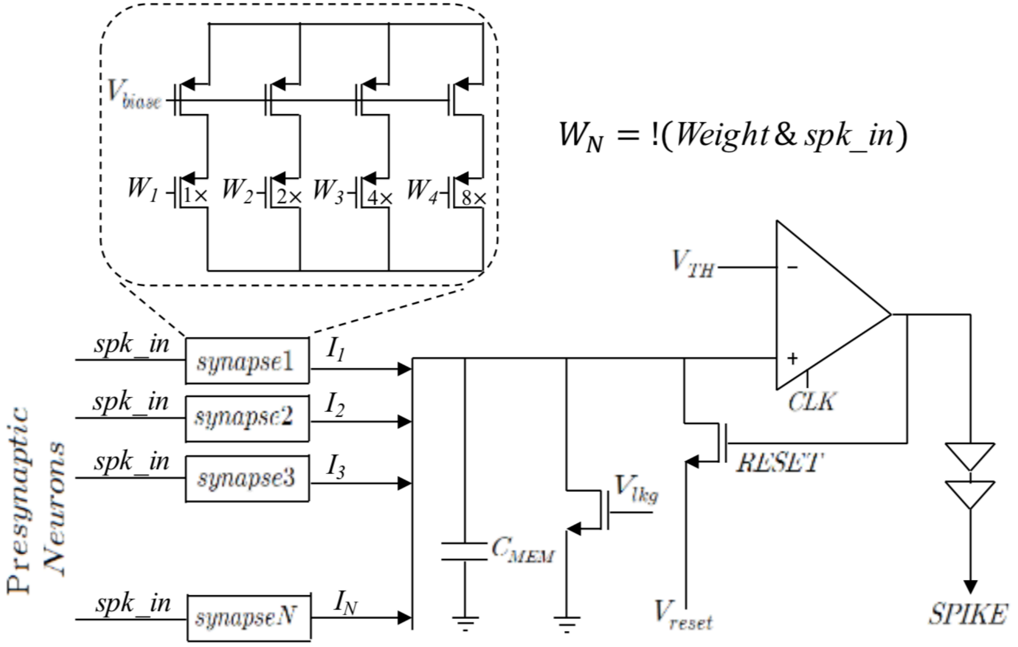


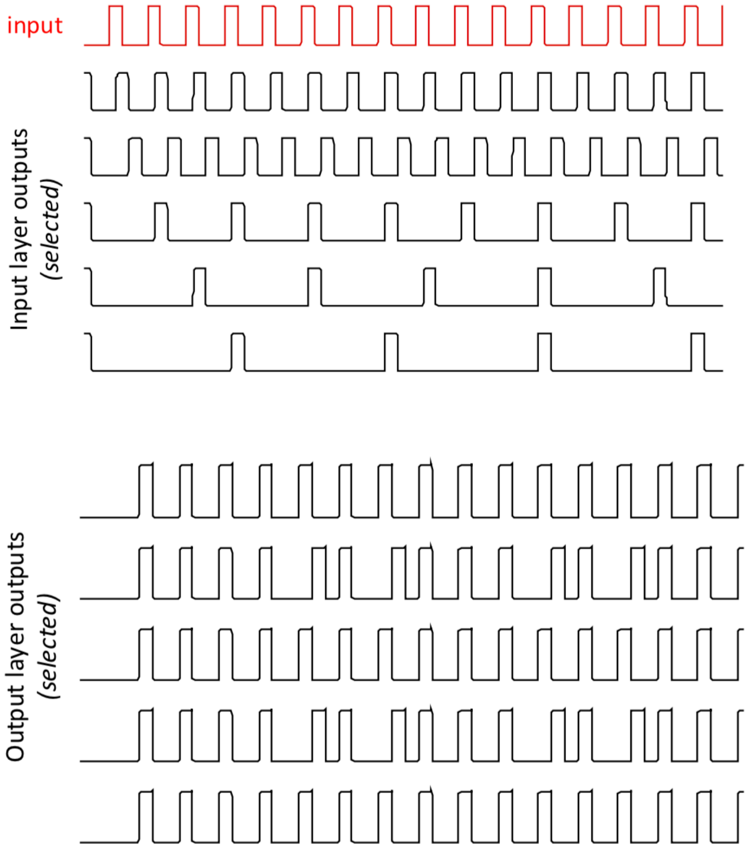


SNN Mixed-Signal Chip Design

❖ **Circuit Design for SNN's building blocks**

- **Mixed-Signal Synapse and Neuron circuit**
- **Minimal number of TRs and components**

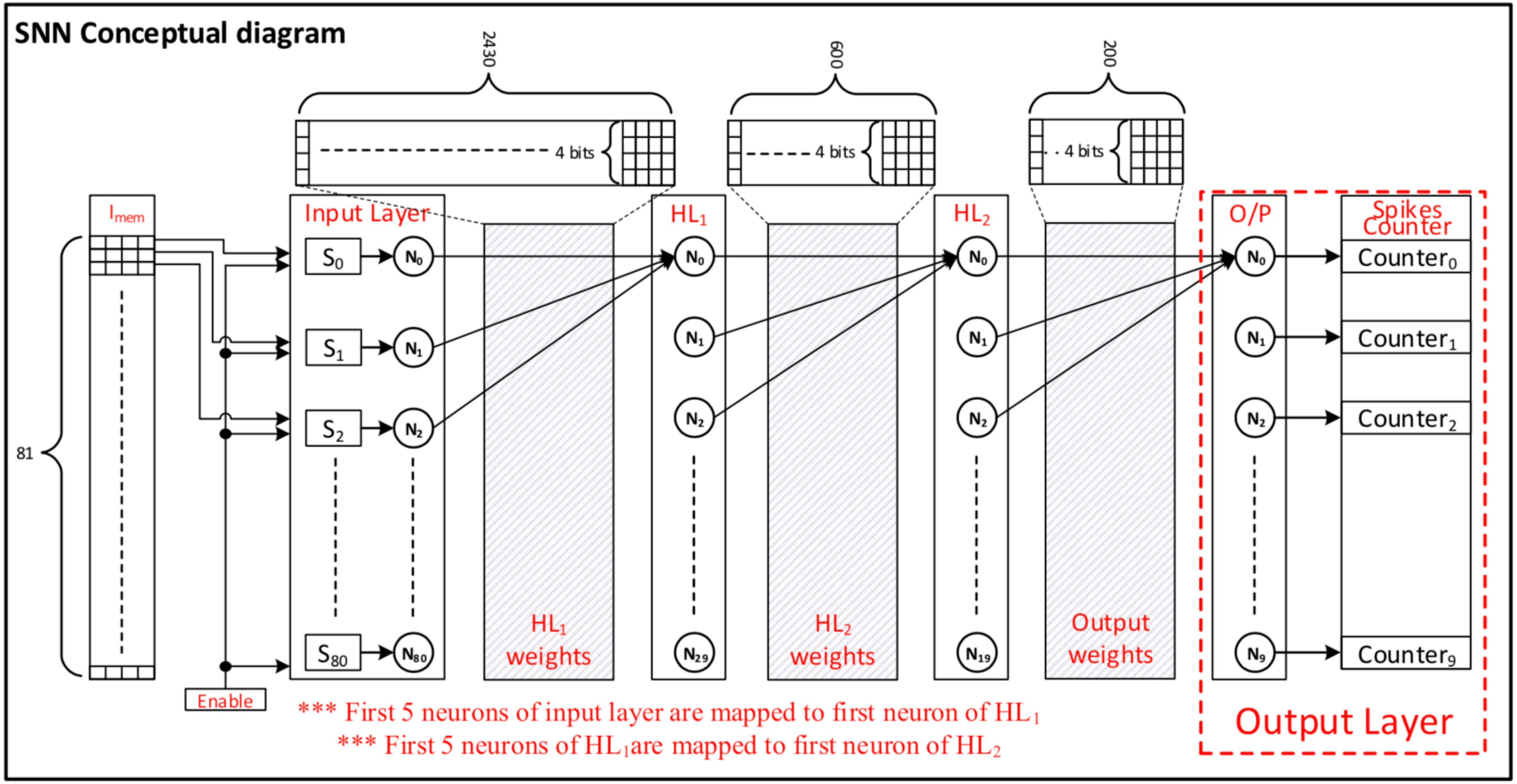




SNN Mixed-Signal Chip Design

❖ **Spiking Neural Network (SNN) Chip for MNIST dataset**

- **Compact Low Power Synapse and Neuron circuit**
- **Minimal number of Synapse and Neuron cells**

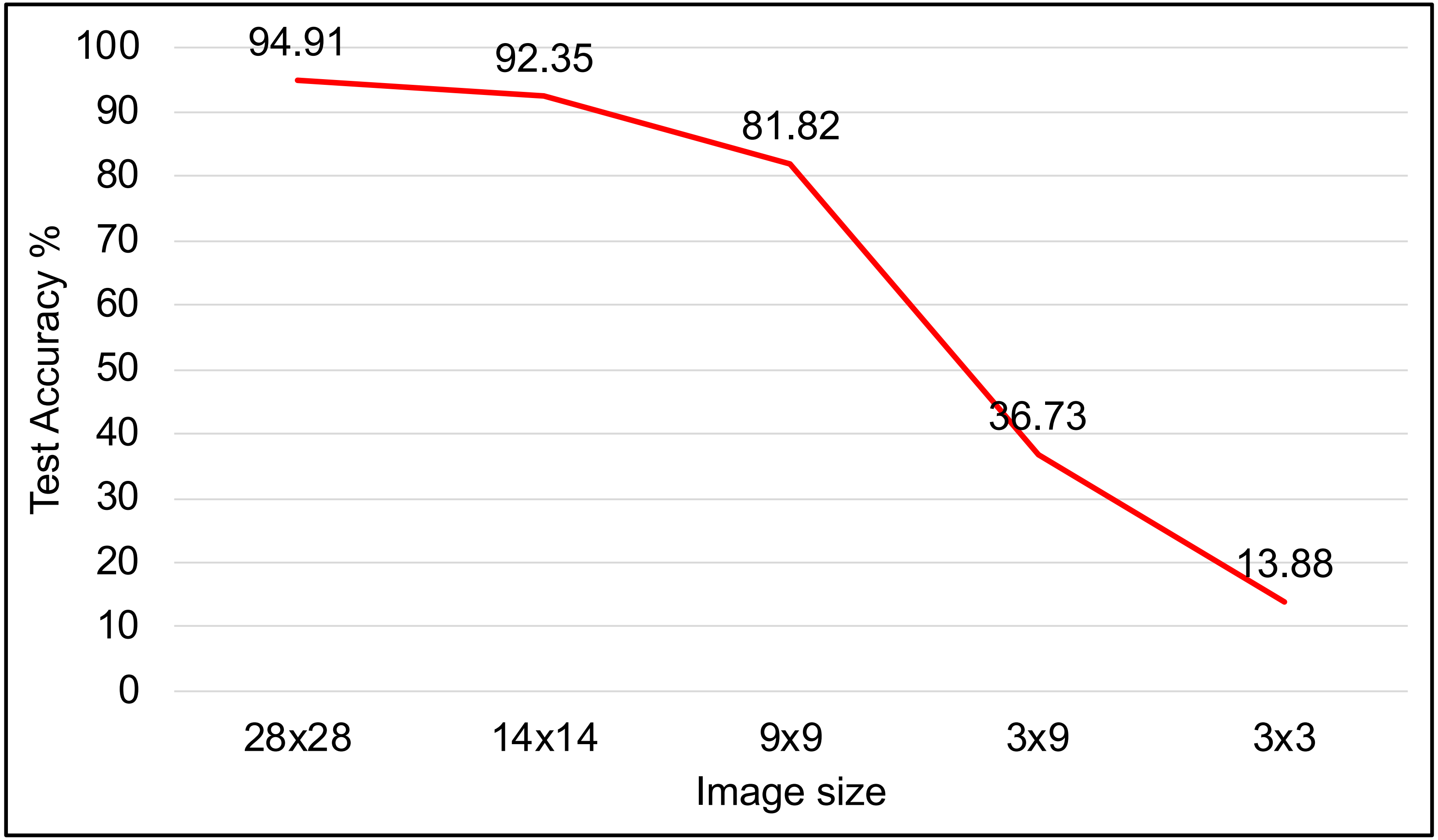


*** First 5 neurons of input layer are mapped to first neuron of HL_1

*** First 5 neurons of HL_1 are mapped to first neuron of HL_2

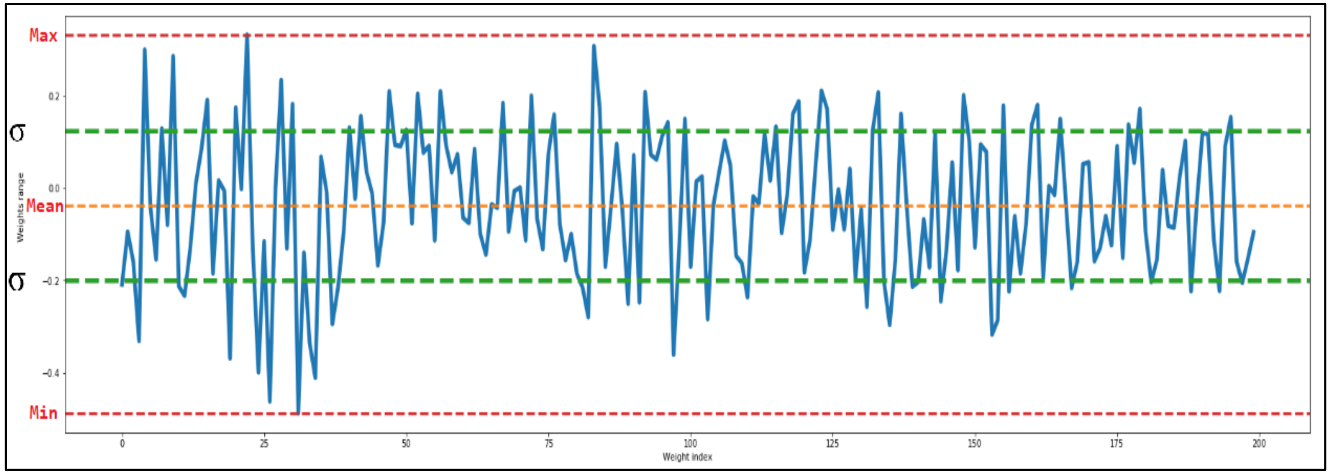
SNN Mixed-Signal Chip Design

- SNN Simulation results to determine the image size for low complexity SNN chip



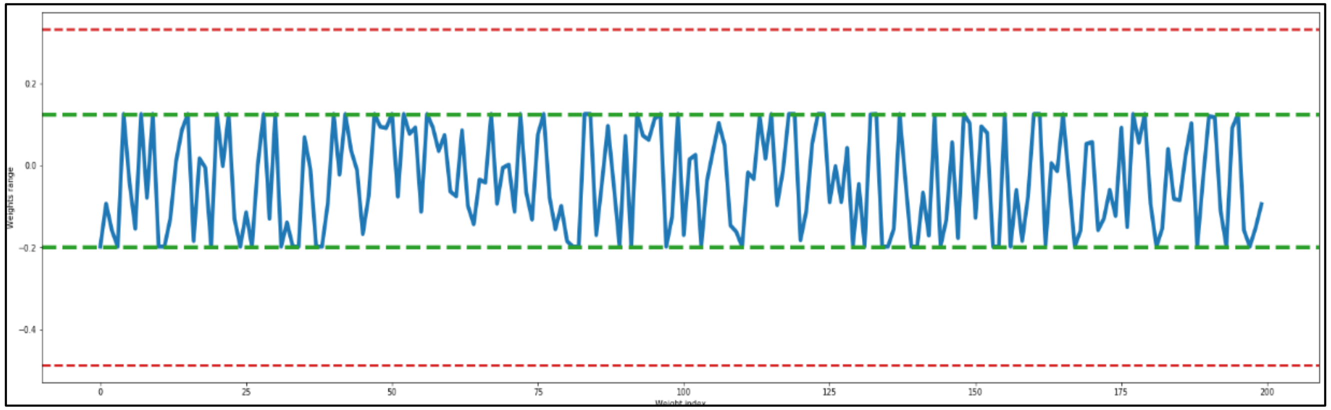
SNN Mixed-Signal Chip Design

- **Weight Conversion: Float to Integer for compact implementation**



Output layer weights before quantization

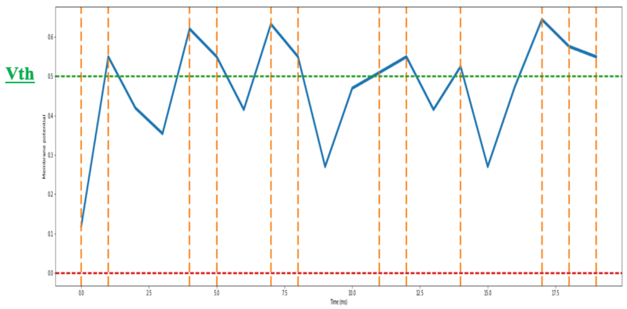
(Max=0.333, Min=-0.488 Mean=-0.037, σ =0.162, mean+σ=0.125 , mean−σ =0.198)



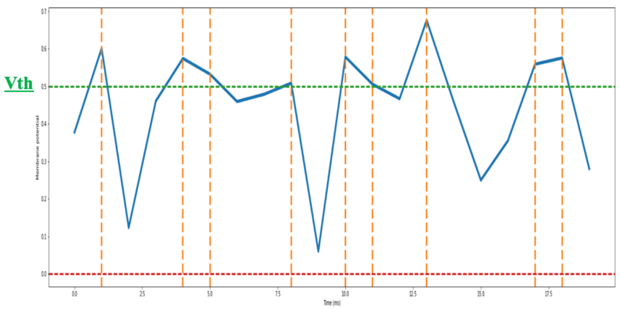
Output layer weights after quantization (quantization range can be easily controlled to achieve minimum error)

SNN Spike Propagation Results

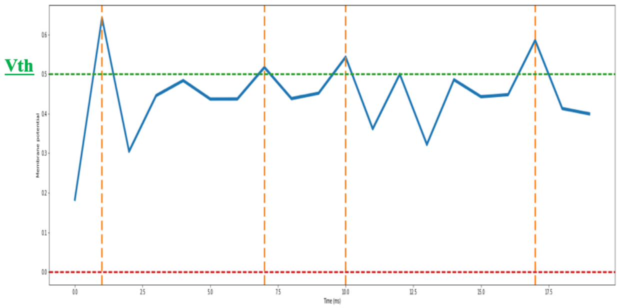
❖ Layer1 Output



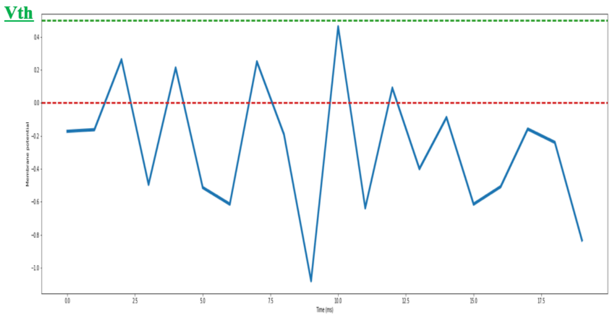
❖ Layer2 Output



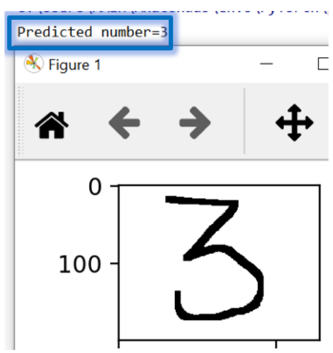
❖ Layer3 Output



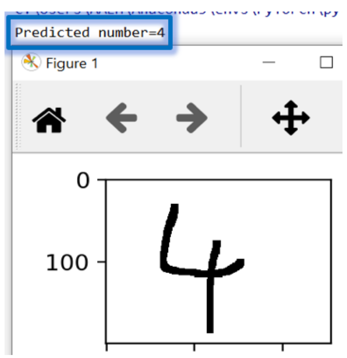
❖ Layer4 Output



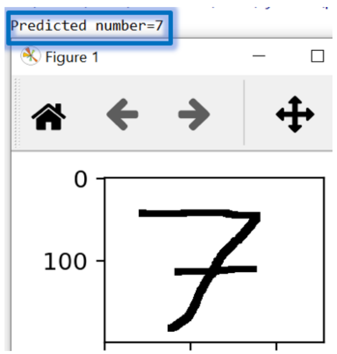
❖ Classification Results



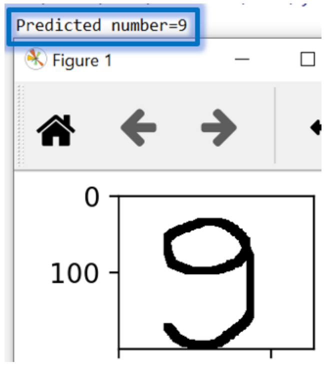
```
Predicted number=3
```



```
Predicted number=4
```



```
Predicted number=7
```



```
Predicted number=9
```