Flash Memory Summit

Analog In-Memory Compute using SST SuperFlash

memBrain™

Mark Reiten

Abstract

Artificial intelligence based on advanced machine learning models has gained tremendous momentum in many industries. Many machine learning optimized digital processing solutions have been introduced in the last 5 years, but none can match the power and performance advantages of analog memory-based computing devices. SST's memBrain cost/performance can be 10 times better and the power can be 100x lower than a comparably performing digital solution. This is accomplished by storing multiple levels (up to 256) per cell to represent a "weight" or "synapse" in a neural model. Multiplication is done through cell operation and addition is done by summing the output lines. Vector Matrix Multiplication is accomplished through design techniques so any existing SuperFlash process can support this new optimized compute paradigm.

Agenda

- Neural Systems and Uses
- Deep Neural Networks: *The Problem*
- Analog vs Digital compute
- What is memBrain
- memBrain^{tm} Analog Inference Designware
- Many Model Types can be supported
- Why it works
- Software Flow

Neural Systems and Uses

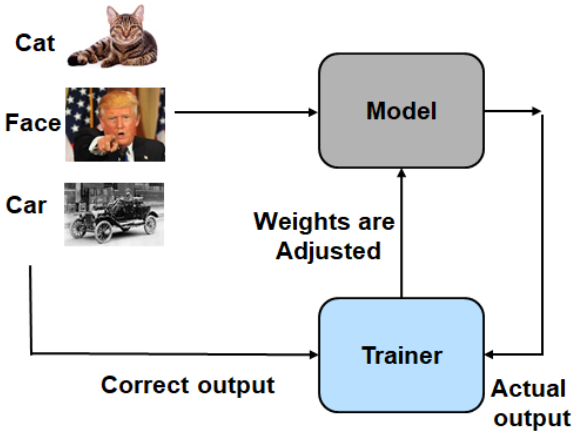
Massive input data sets required

Requires floating point math

***Many** training iterations needed*

GPUS or FPGAs or TPUs or… best

Training



Only simple math needed

*High parallelism needed for **low latency***

Memory with embedded processing is best – ESF1 or ESF3 is best solution!

Inference

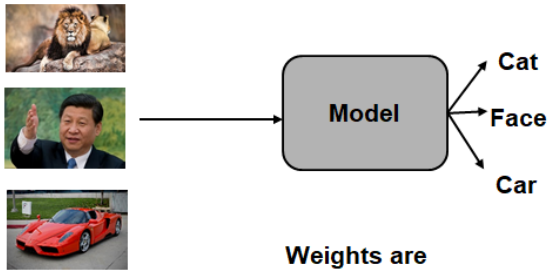


Image Processing

Object Detection and Classification

Speech recognition

Natural Language Processing

Text processing

Search

Similarity/relevance analysis

Recommendations/Ads

Data Mining

Business Pattern extraction

Bioinformatics

Hybrid Applications

Visual scene understanding

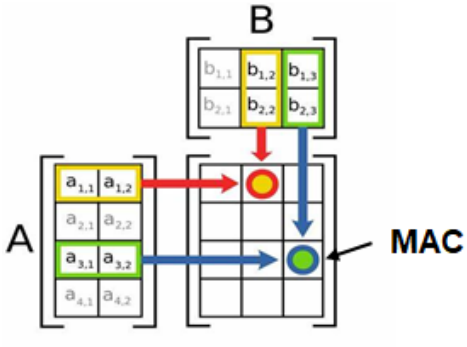
Personal Assistants (speech)

Game playing

Robotics (image + speech)

Deep Neural Networks: *The Problem*

DNNs require vast numbers of Multiply-Accumulate operations (MACs)

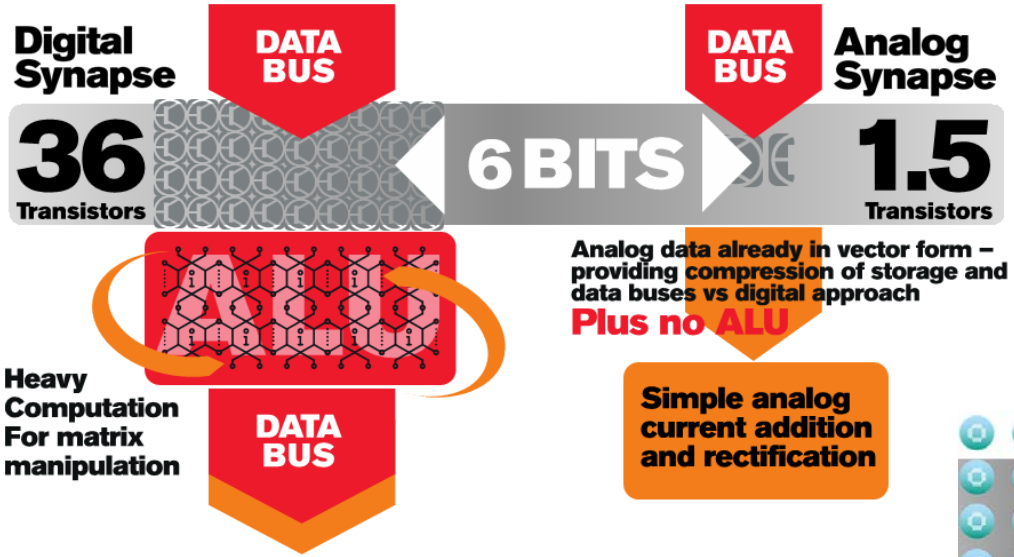


Network	Weights	MACs
Alexnet	61 M	725 M
ResNet 50	23 M	3.5 B
VGG-19	46 M	22 B

Vast numbers of MAC operations favors keeping weights in local storage

Cannot fit these into a stand alone digital edge processor

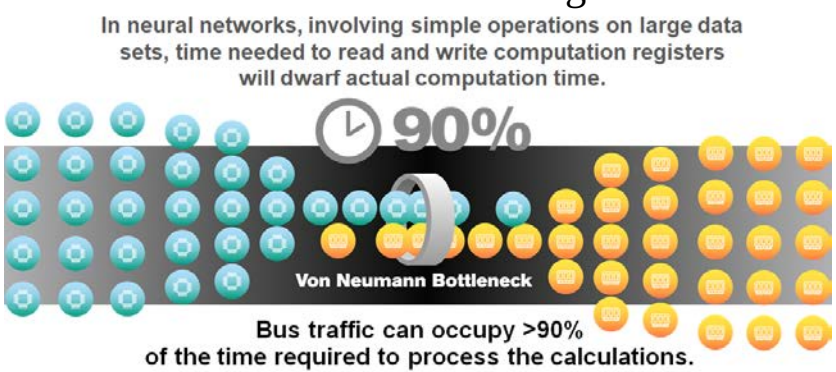
Analog vs Digital Compute



Analog advantages:

Data and compute compression

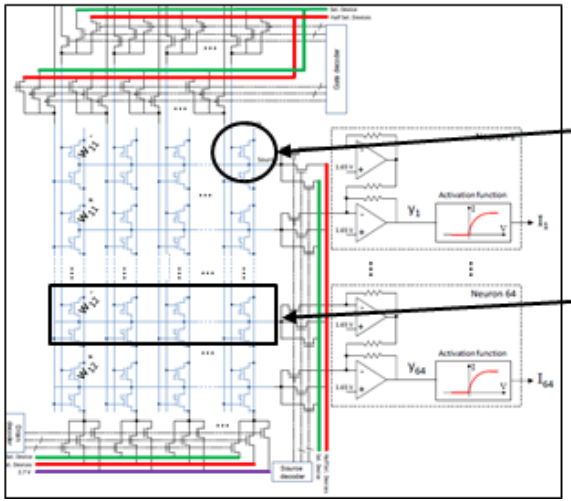
No Data thrashing!



What is memBrain?

memBrain

solves the compute problem by storing the weights in eFlash and using analog cell operation to perform the MAC operations inside the storage array



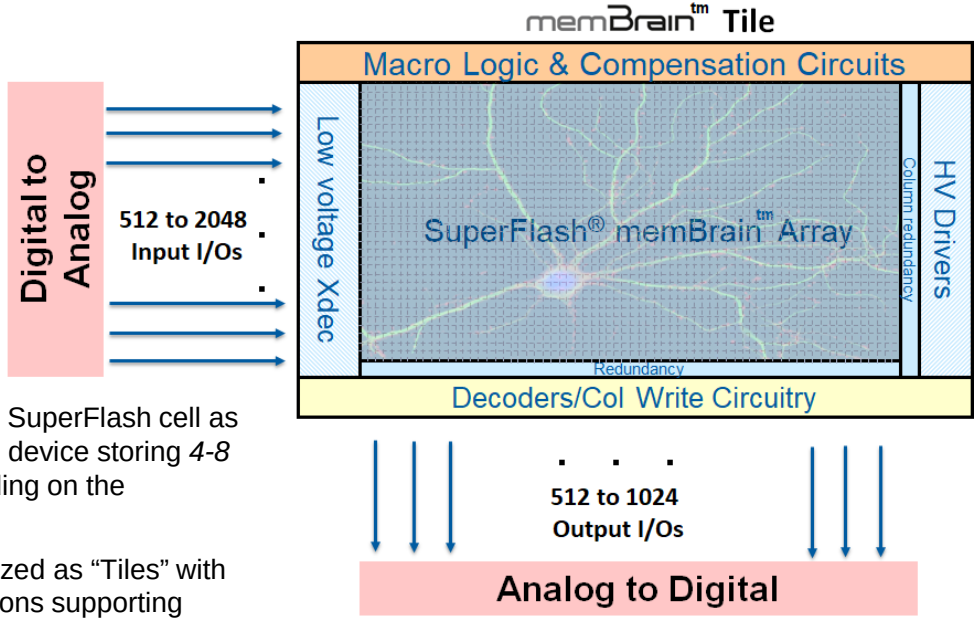
Each cell stores up to 8 bits in 1.5 transistors
Multiply happens through cell operation

Compared to 48 SRAM transistors for 8 bits!

Output line functions as a Neuron
Summation happens along output

Power is ~.3pj per MAC !!

memBrain™ Analog Inference Designware



memBrain uses the SuperFlash cell as a multi-level analog device storing *4-8 bits per cell* depending on the application

memBrain is organized as “Tiles” with wide I/O configurations supporting ***massively parallel*** multiply/accumulate operations

512x512 tile *full frame cycle time 10-30us*

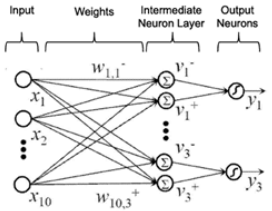
- ***Depends on D-A and A-D power***

Energy is ***0.3pJ per MAC*** with D/A+A/D *@ 30us frame cycle time*

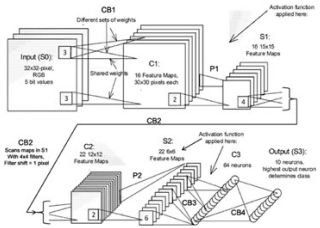
Area with D to A input and A to D output blocks = **.48 mm^2 on 40nm *for 512x512 Tile***

Performance per silicon area and power are orders of magnitude better than optimized digital solutions

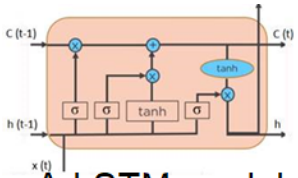
Many Model Types can be supported



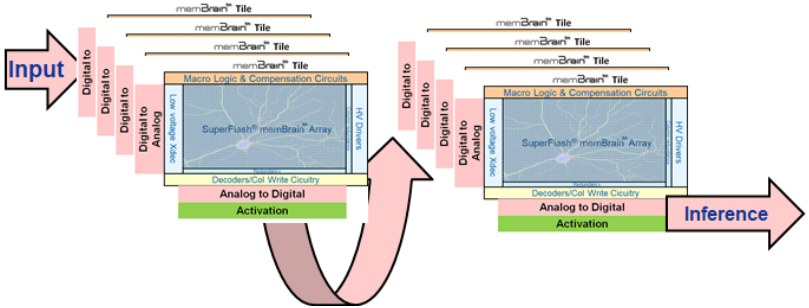
A Muli-layer Perceptron



A Deep CNN



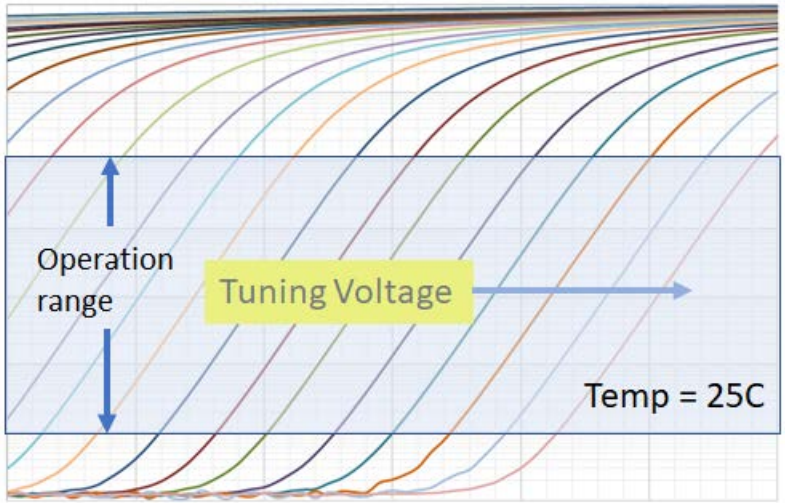
A LSTM module From a recurrent model

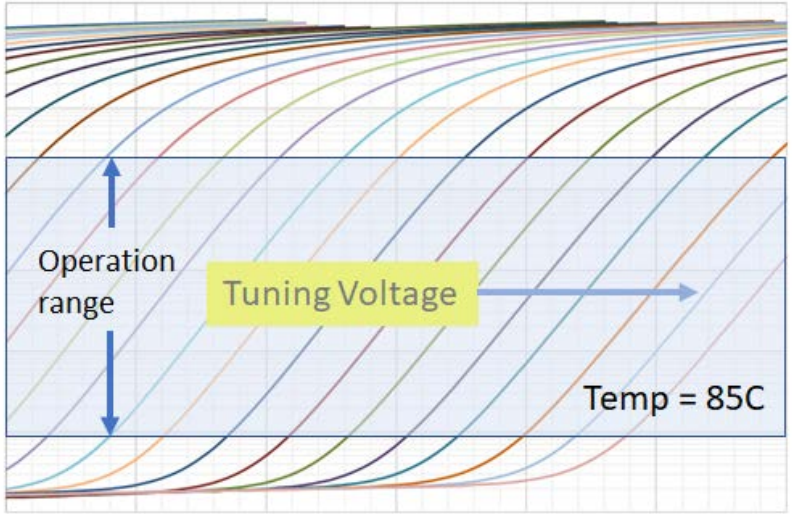


A neural network using
*In-memory analog computing assembled from memBrain "**TILES**"*

Why it Works

ESF3 Cell analog operation varies the floating gate potential with repeatable consistent separation between steps across array and temp





ESF3 Cell data retention is excellent
Noise is low
Noise and temperature can be compensated for

Software Flow

Raw Data	Data cleansing and annotation	Model Training	Model Testing	Load model into memBrain
Data can be images, natural language text, audio or virtually anything	Humans or machines clean and annotate data to describe desired features	Modern solutions use software frameworks and optimized hardware	Check the trained, ***quantized*** model on new data to prove accuracy	Generate a bit map which can be *programmed into the memBrain device*

python™

TensorFlow

Python Script kit and examples for weight quantization and noise injection prior to testing

Programming tool for loading Bitmap into memBrain Tile

Thank You