All-Digital Time-Domain CNN engine using Bi-directional Memory Delay Lines for Energy Efficient Edge Computing

Aseem Sayal, Shirin Fathima, S.S. Teja Nibhanupudi, Jaydeep P. Kulkarni

Department of Electrical and Computer Engineering

The University of Texas at Austin, TX

THE UNIVERSITY OF TEXAS AT AUSTIN

UT ECE

ELECTRICAL & COMPUTER ENGINEERING

Outline

- Motivation
- Concept of Time-domain MAC
- Proposed CNN Engine Architecture
- Chip Implementation and Measurements
- Comparison with Prior Work
- Summary

Need: Energy Efficient Edge Computing

Applications

- ➢ Image Classification
- ➢ Speech Recognition
- ➢ Object Detection

Energy efficient edge processing required for privacy and minimal data transmission

Convolutional Neural Network (CNN)

- State-of-the-art classification accuracy for image/speech recognition
- Multiple filters to extract specific features
- Memory dominant and compute intensive
- Power hungry MAC operations in convolution

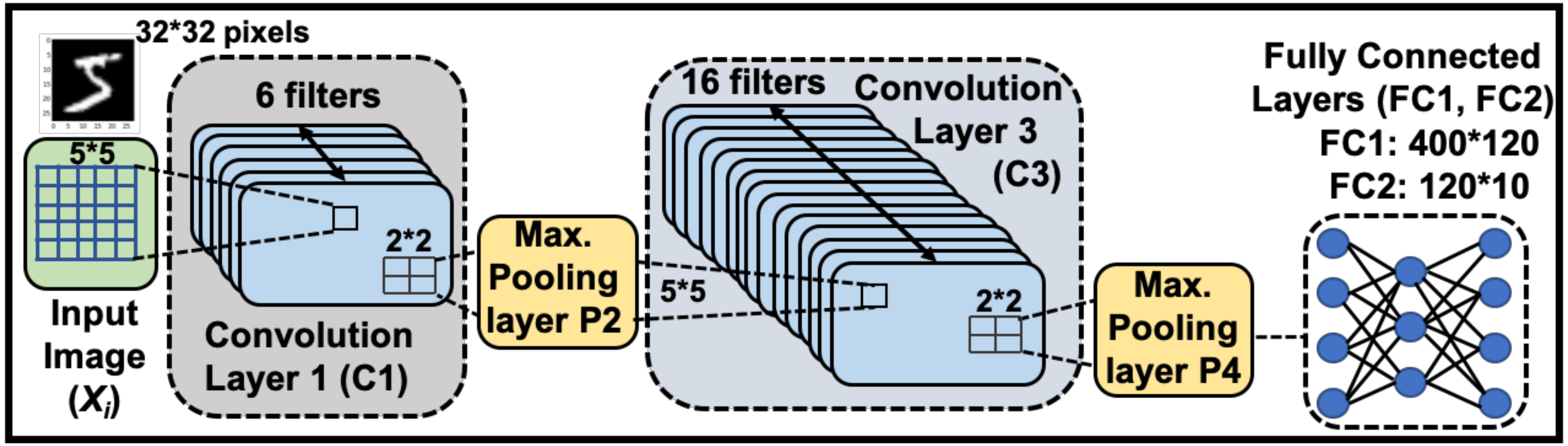
Multiply-Accumulate-Average (MAV)

$$Y = \frac{1}{N}\sum_{i=1}^{N}\left(X_i * w_i\right)$$

N = # dot products per MAC

Output value **Input pixel** **Weight**

LeNet-5 CNN



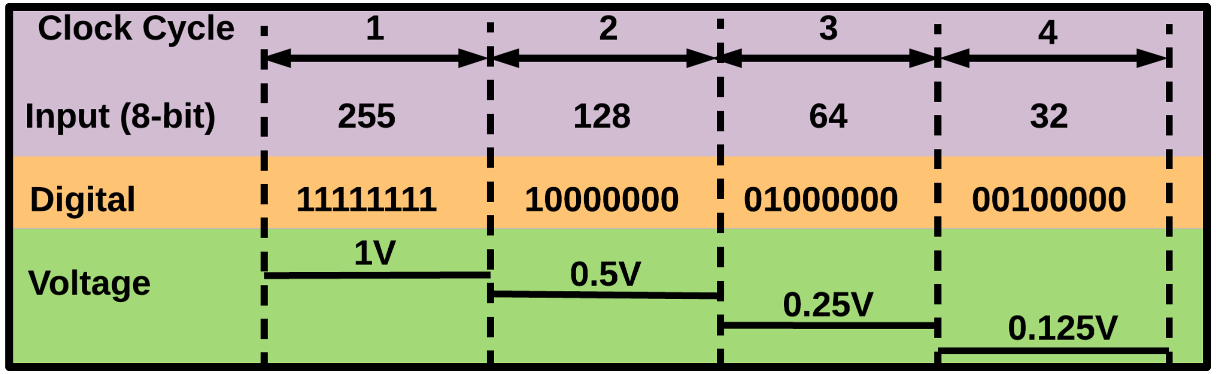
Ref:
LeCun et al., Gradient-based learning applied to document recognition. Proc. Of IEEE, 1998

Why Time Domain Signal Processing?

Clock Cycle	1	2	3	4
Input (8-bit)	255	128	64	32
Digital	11111111	10000000	01000000	00100000

- **Digital domain:** multi-bit digital vector – high C_{DYN} and toggle activity ➔ consequently switching power

Why Time Domain Signal Processing?

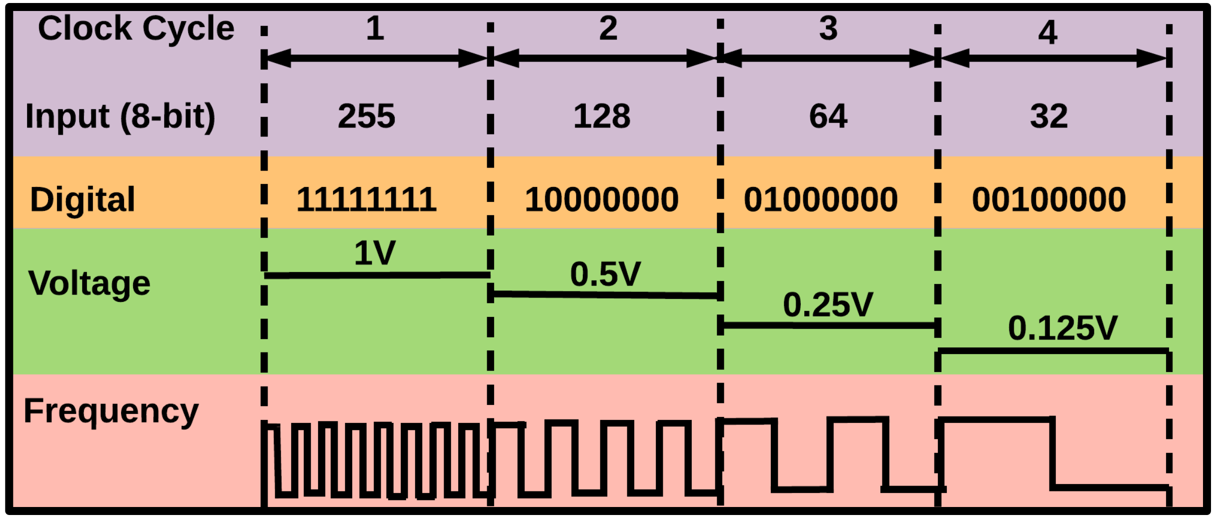


Clock Cycle	1	2	3	4
Input (8-bit)	255	128	64	32
Digital	11111111	10000000	01000000	00100000
Voltage	1V	0.5V	0.25V	0.125V

- ➢ **Digital domain:** multi-bit digital vector – high C_{DYN} and toggle activity ➔ consequently switching power

- ➢ **Analog Voltage domain:** continuously varying voltage signal – limited voltage scalability and need of power and area intensive ADCs/DACs

Why Time Domain Signal Processing?



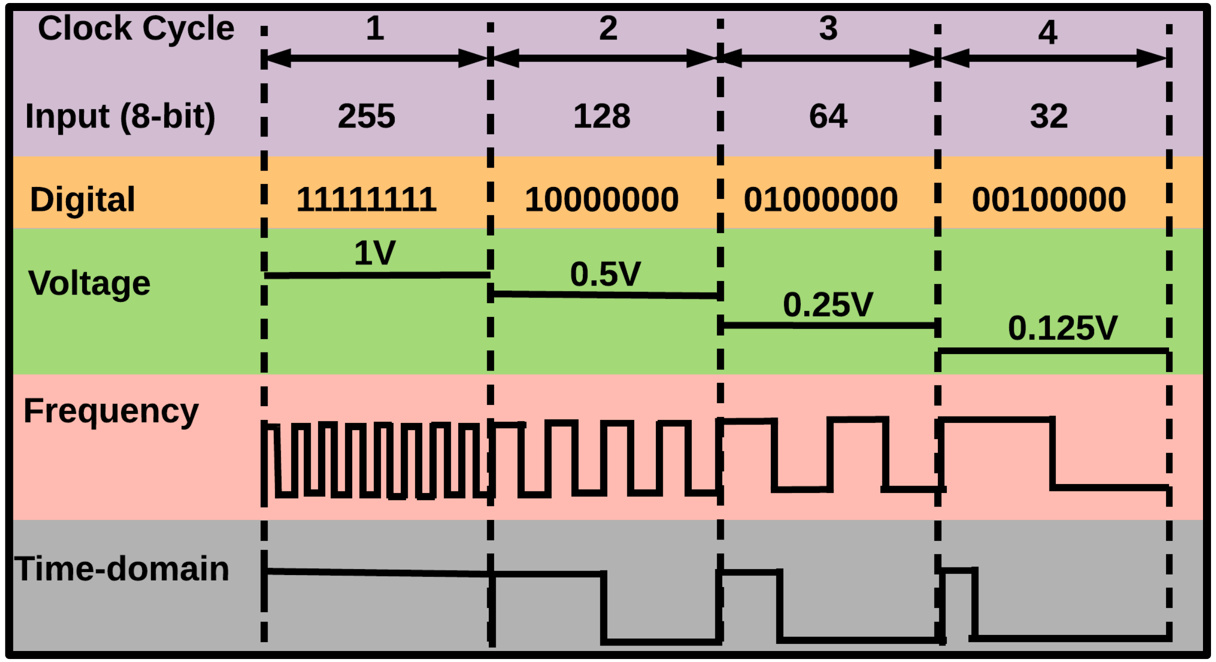
Clock Cycle	1	2	3	4
Input (8-bit)	255	128	64	32
Digital	11111111	10000000	01000000	00100000
Voltage	1V	0.5V	0.25V	0.125V
Frequency				

- **Digital domain:** multi-bit digital vector – high C_{DYN} and toggle activity ➔ consequently switching power

- **Analog Voltage domain:** continuously varying voltage signal – limited voltage scalability and need of power and area intensive ADCs/DACs

- **Frequency domain:** frequency varying signal – need of accurate frequency generators/modulators ➔ limits performance scalability

Why Time Domain Signal Processing?



Clock Cycle	1	2	3	4
Input (8-bit)	255	128	64	32
Digital	11111111	10000000	01000000	00100000
Voltage	1V	0.5V	0.25V	0.125V

Time domain: multi-bit digital bit stream encoded as PWM signal – smaller C_{DYN} and toggle activity, and ultra-low voltage operation resulting in low power, no need of multiple clocks/frequency modulators

- **Digital domain:** multi-bit digital vector – high C_{DYN} and toggle activity ➔ consequently switching power
- **Analog Voltage domain:** continuously varying voltage signal – limited voltage scalability and need of power and area intensive ADCs/DACs
- **Frequency domain:** frequency varying signal – need of accurate frequency generators/modulators ➔ limits performance scalability

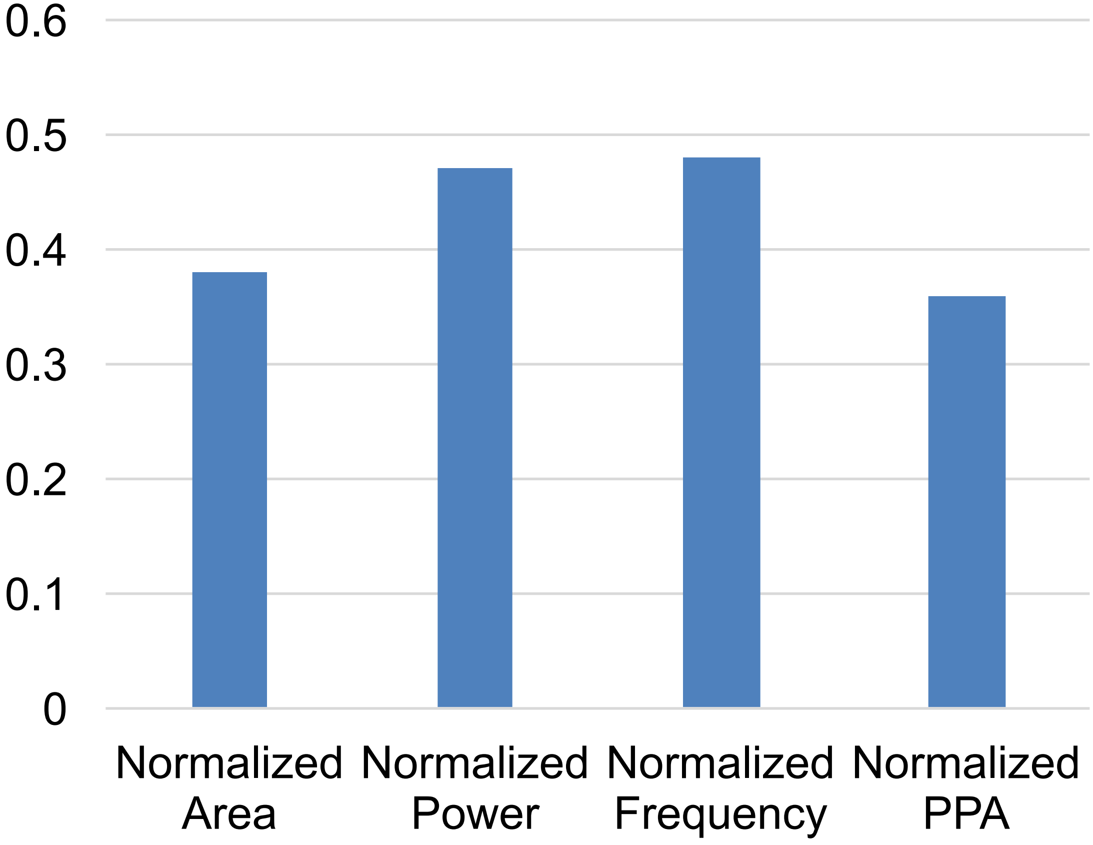
Time Domain Approach *vs.* Digital Domain

Design Metrics	Digital Domain	Proposed Time Domain
Configuration	8b adder + 12b register for partial sum	16-unit MDL + 12b counter for partial sum
Technology Node	40nm	40nm
Area (post-PNR)	381μm^2	144μm^2
F_{MAX} supported	1.67GHz	0.8GHz
Power (post-PNR)	54.06μW	25.23μW
PPA/MAC (μW*μm^2*ns)	12358	4541

PPA – Power Performance Area Product

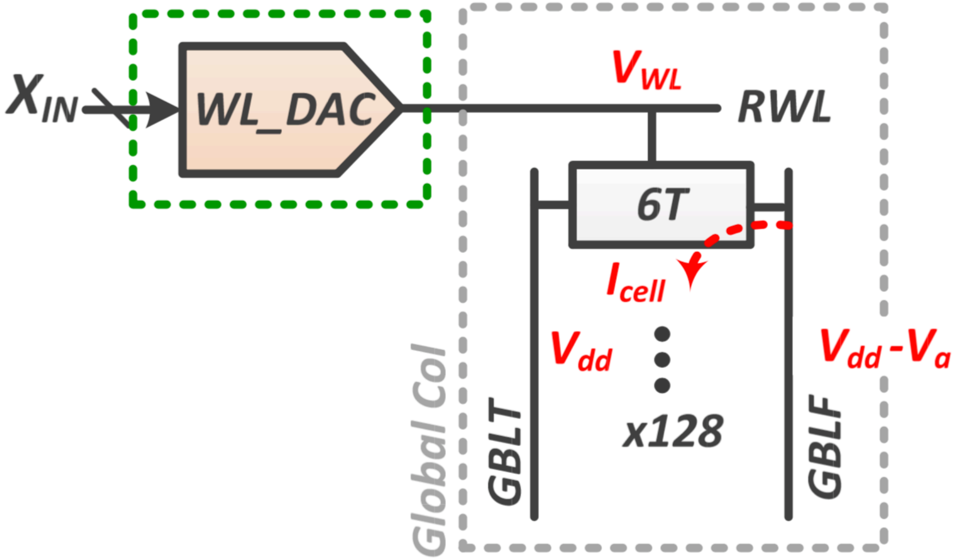
PNR – Place and Route

Time Domain Metrics Normalized *w.r.t.* Digital Domain Metrics



2.72x PPA improvement

Prior Work: Analog Voltage Approaches

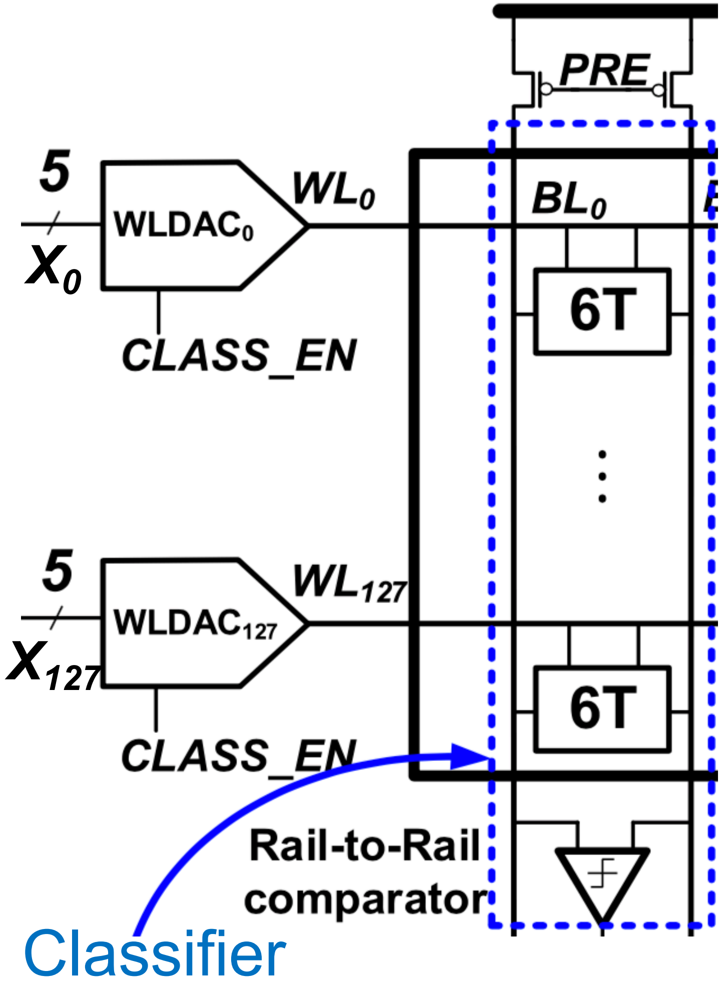


Bit-cell discharge current:

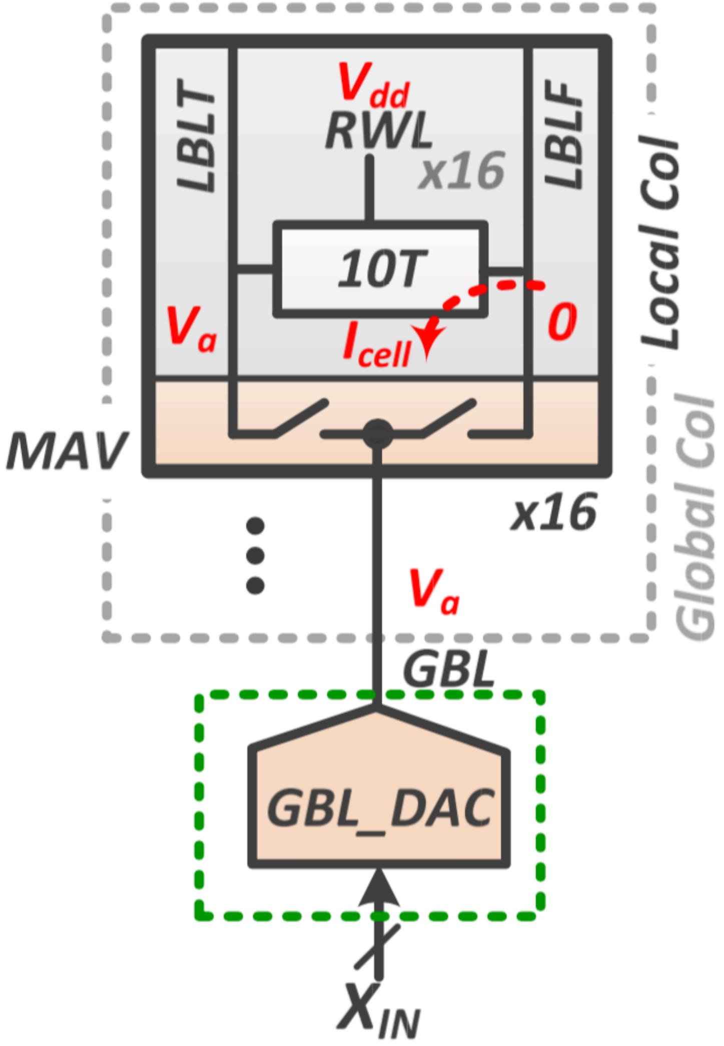
$$I_{cell} \propto (V_{WL} - V_t)$$

$$V_a \propto I_{cell}$$

M. Kang et al., VLSIC'16 (UIUC)

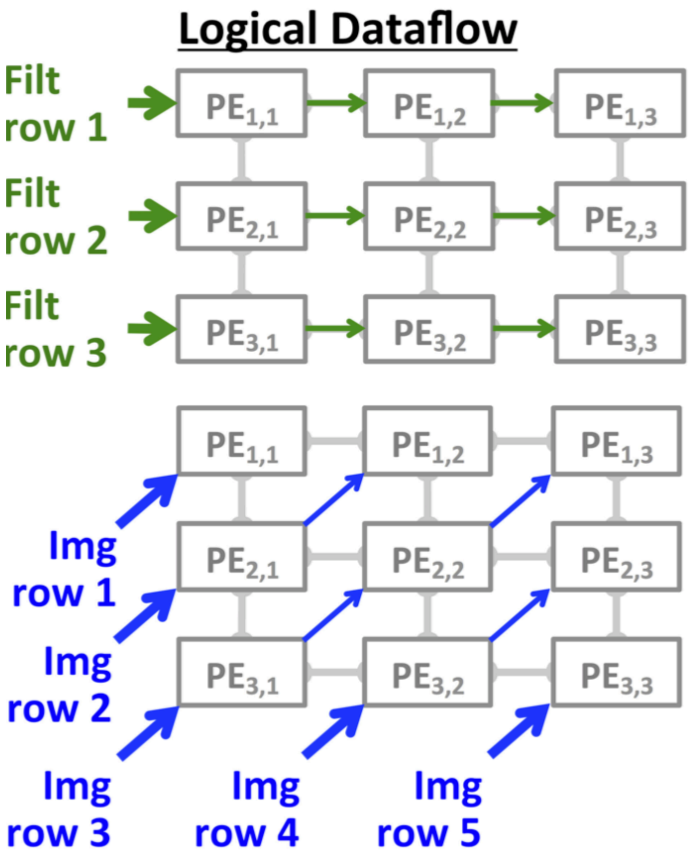


J. Zhang et al., JSSC'17 (Princeton)



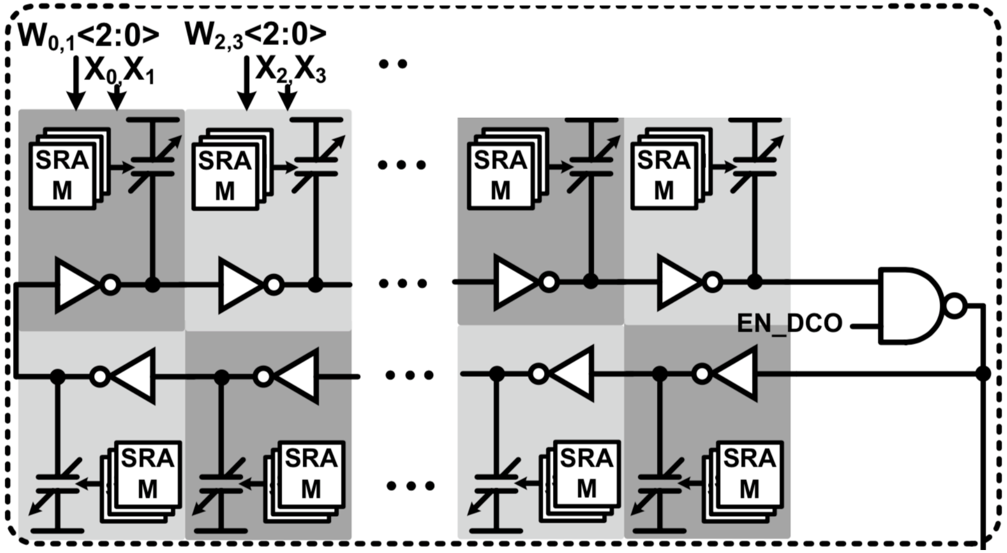
A. Biswas et al., ISSCC'18 (MIT)

Prior Work: Digital, Time and Frequency domain



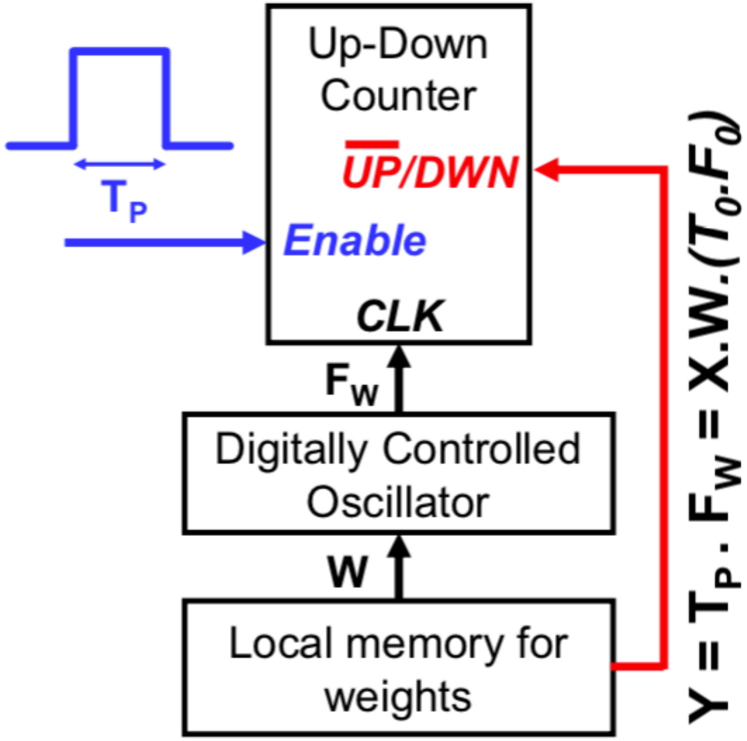
Y. Chen et al., ISSCC'16
(MIT)

$$T_{DCO} = \sum Delay_i \propto \sum X_i \cdot w_i$$



M. Liu et al., CICC'17
(Univ. of Minnesota)

$T_P = X.\ T_0$ $F_W = W.\ F_0$



A. Amravati et al., ISSCC'18
(Georgia Tech)

Outline

- Motivation
- Concept of Time-domain MAC
- Proposed CNN Engine Architecture
- Chip Implementation and Measurements
- Comparison with Prior Work
- Summary

Concept: Time-domain MAC

Image Pixels (X) — **Weights (w)**

$$\begin{array}{|c|c|c|}\hline 123 & 60 & 43 \\ \hline 128 & 255 & 16 \\ \hline 89 & 209 & 76 \\ \hline \end{array} \bullet \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

$$\text{MAC} = 123(1) + 60(0) + 43(1) + 128(0) + 255(1) + 16(1) + 89(0) + 209(0) + 76(1)$$

Concept: Time-domain MAC

Image Pixels (X)

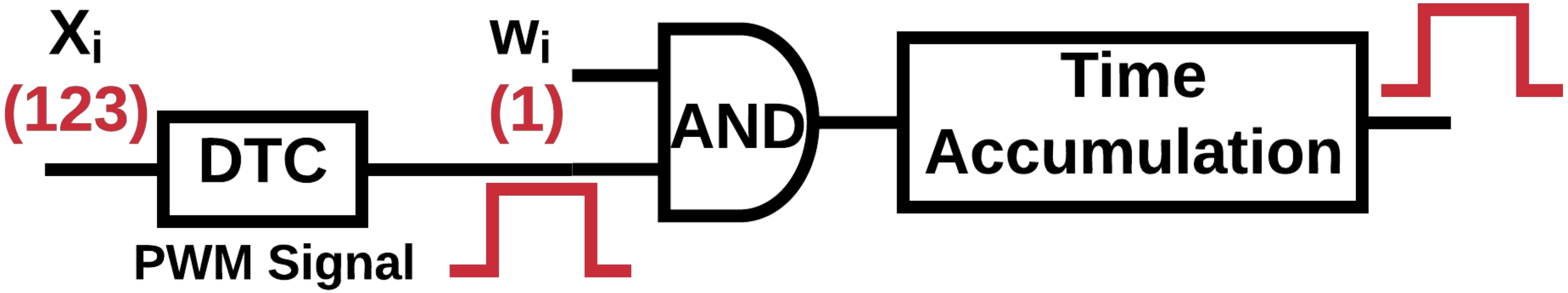
123	60	43
128	255	16
89	209	76

●

Weights (w)

1	0	1
0	1	1
0	0	1

MAC = 123(1) + 60(0) + 43(1) + 128(0) +255(1) + 16(1) + 89(0) + 209(0) + 76(1)



DTC – Digital to Time Converter

Concept: Time-domain MAC

Image Pixels (X)

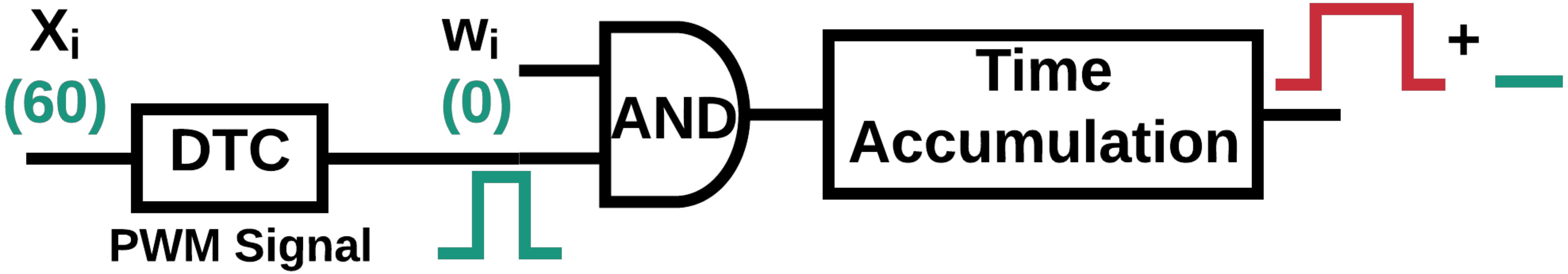
123	60	43
128	255	16
89	209	76

●

Weights (w)

1	0	1
0	1	1
0	0	1

MAC = 123(1) + 60(0) + 43(1) + 128(0) +255(1) + 16(1) + 89(0) + 209(0) + 76(1)



Concept: Time-domain MAC

Image Pixels (X)

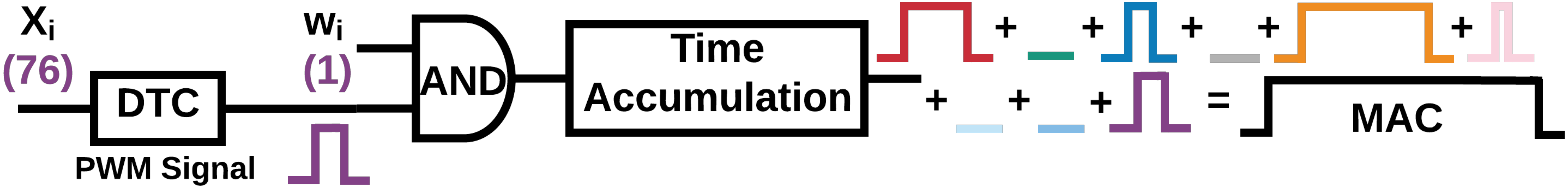
123	60	43
128	255	16
89	209	76

●

Weights (w)

1	0	1
0	1	1
0	0	1

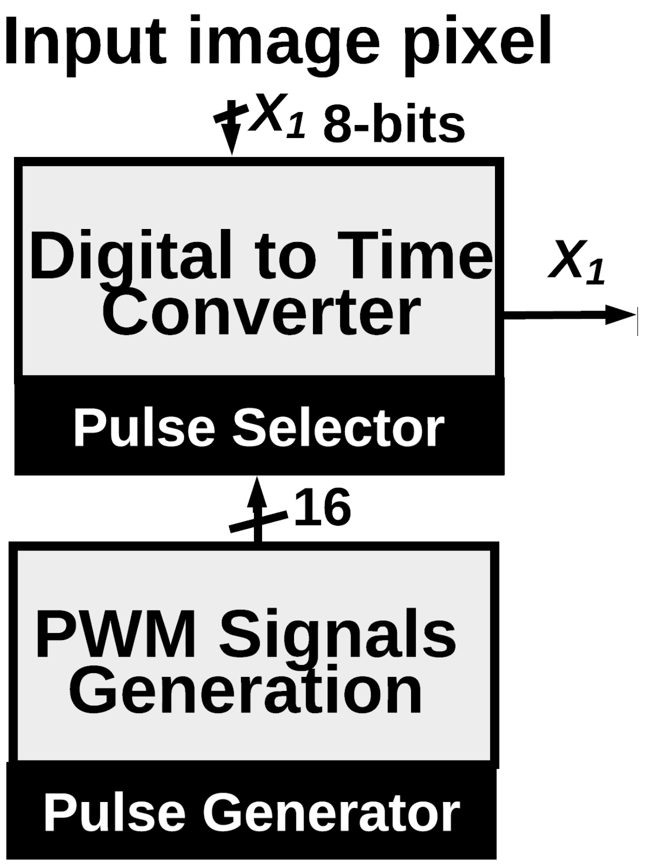
MAC = 123(1) + 60(0) + 43(1) + 128(0) +255(1) + 16(1) + 89(0) + 209(0) + 76(1)

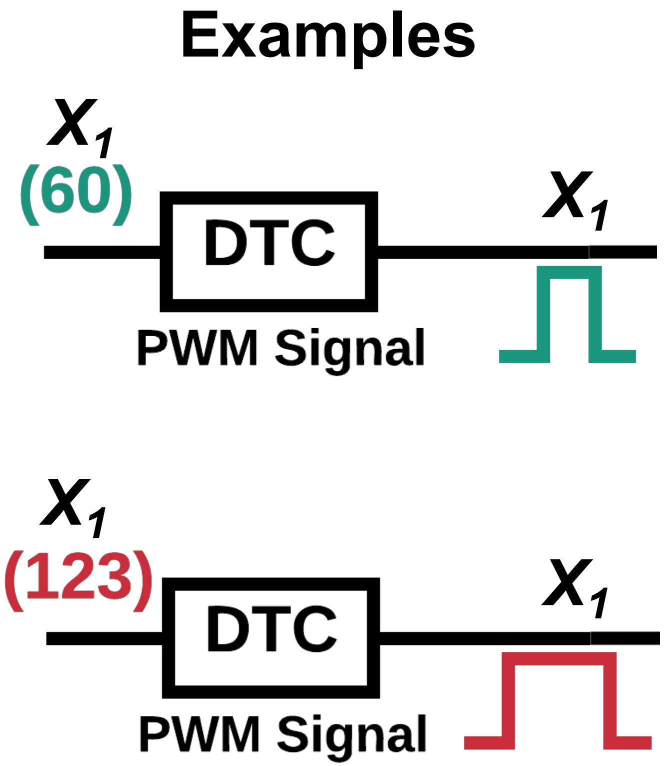


Outline

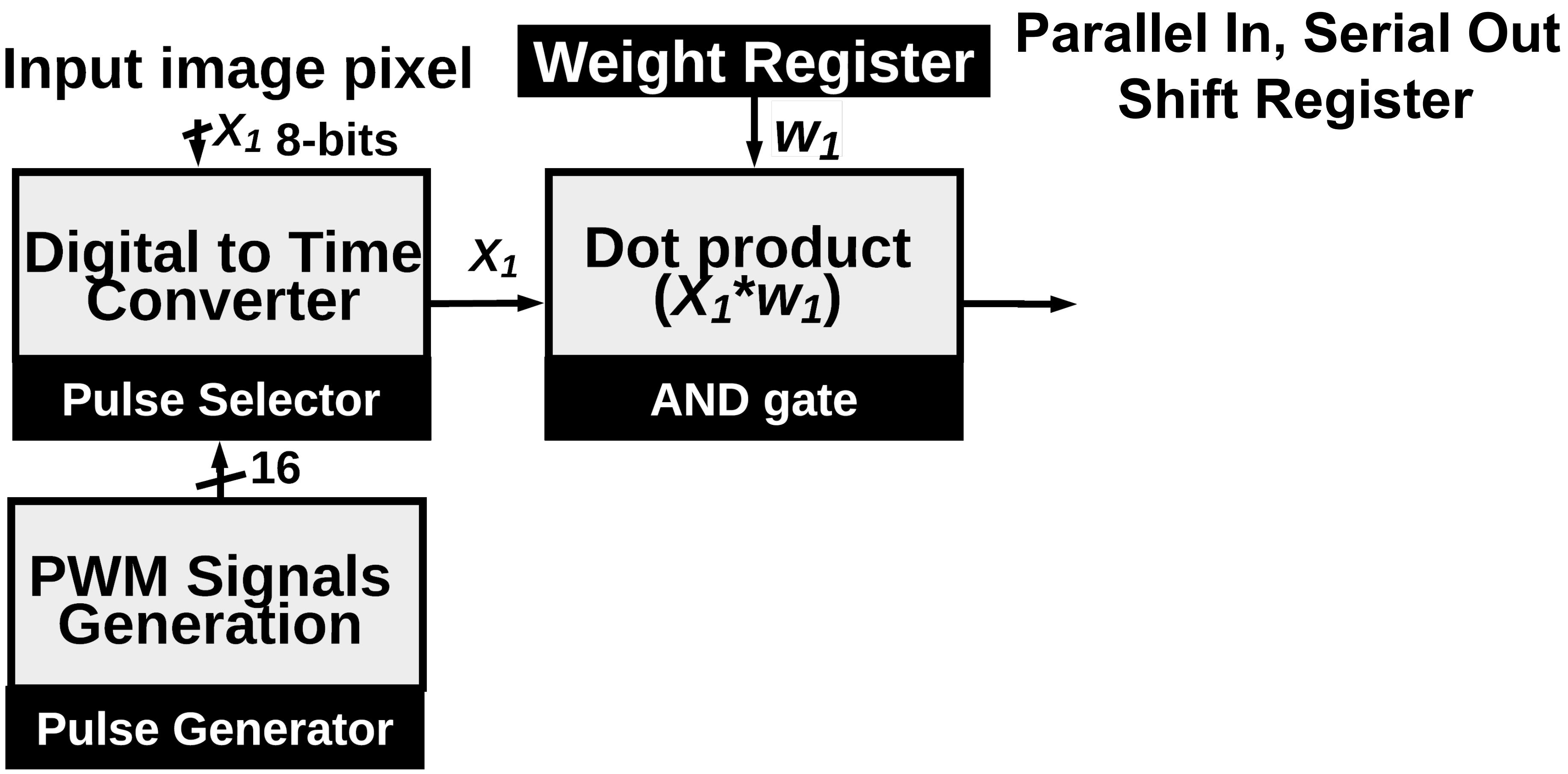
- Motivation
- Concept of Time-domain MAC
- Proposed CNN Engine Architecture
- Chip Implementation and Measurements
- Comparison with Prior Work
- Summary

Architecture of Proposed CNN Engine

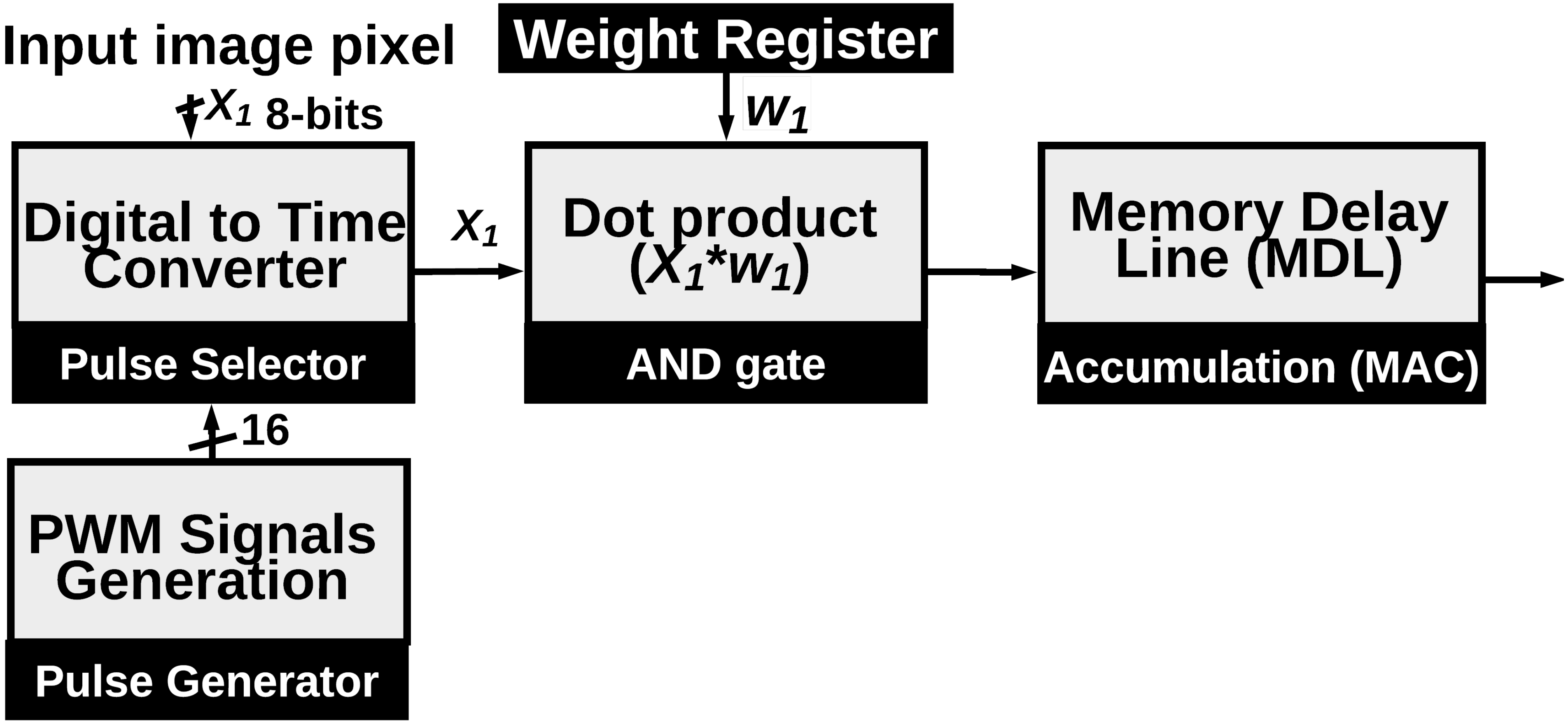




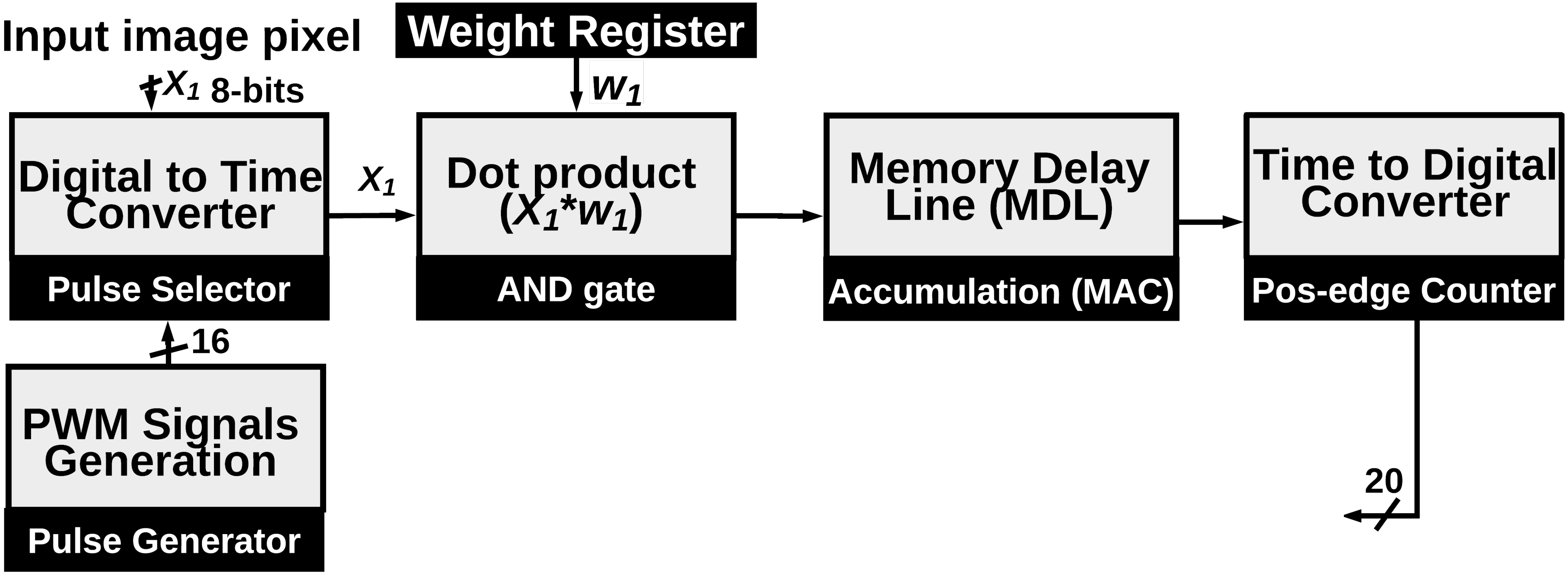
Architecture of Proposed CNN Engine



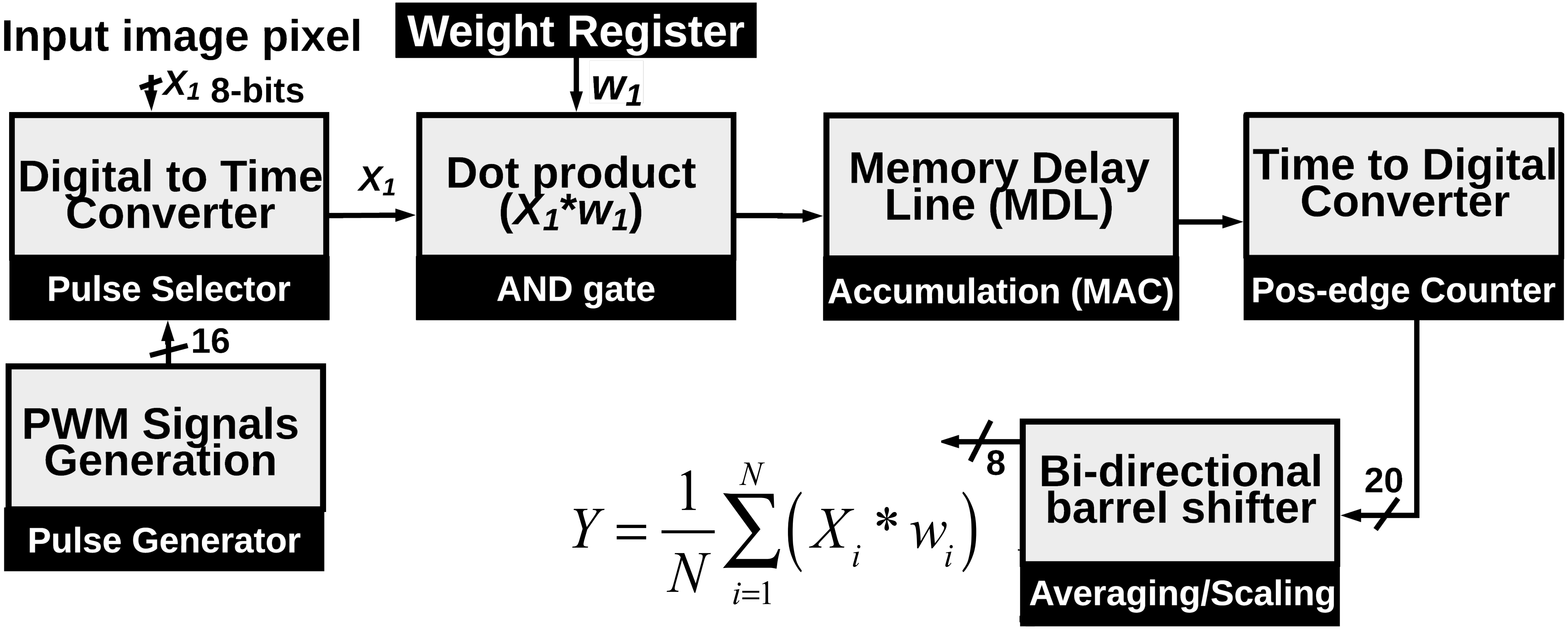
Architecture of Proposed CNN Engine



Architecture of Proposed CNN Engine



Architecture of Proposed CNN Engine

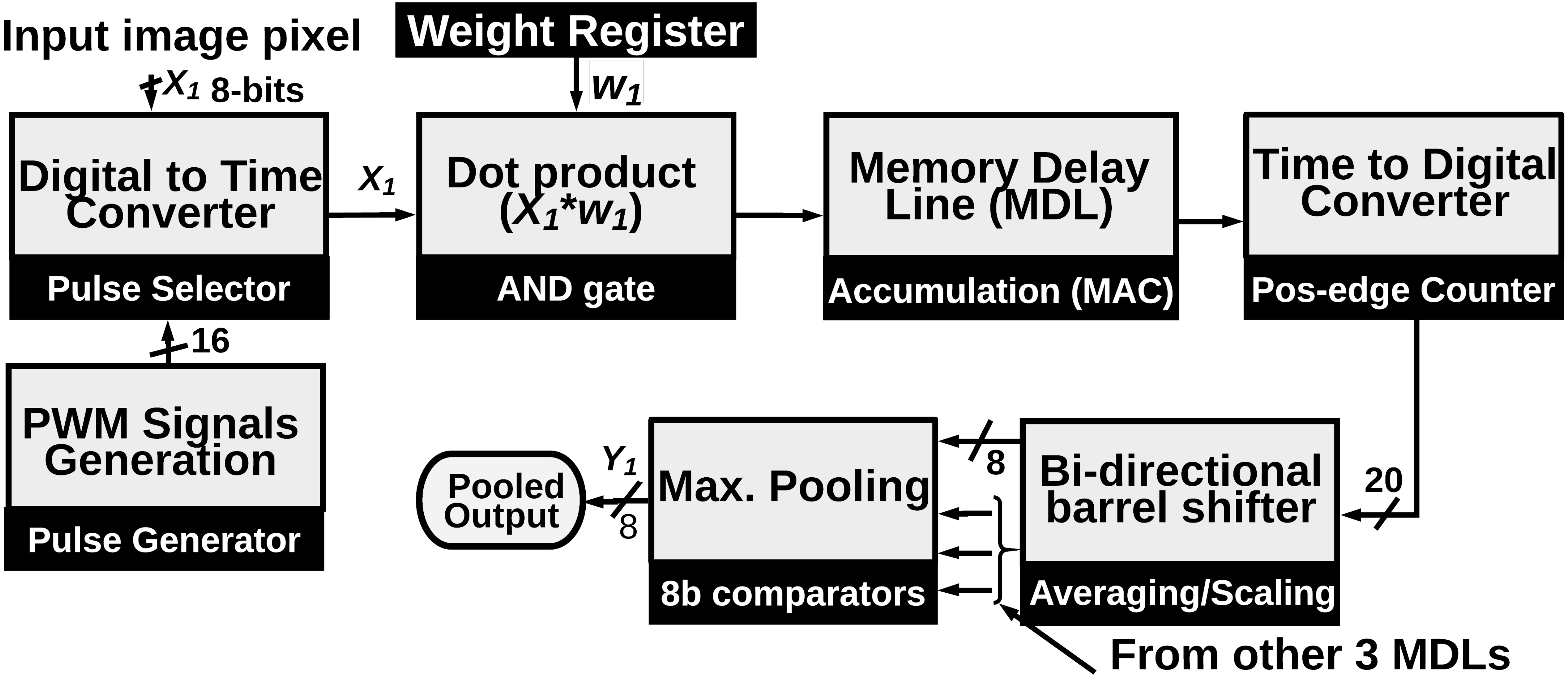


$$Y = \frac{1}{N}\sum_{i=1}^{N}\left(X_i * w_i\right)$$

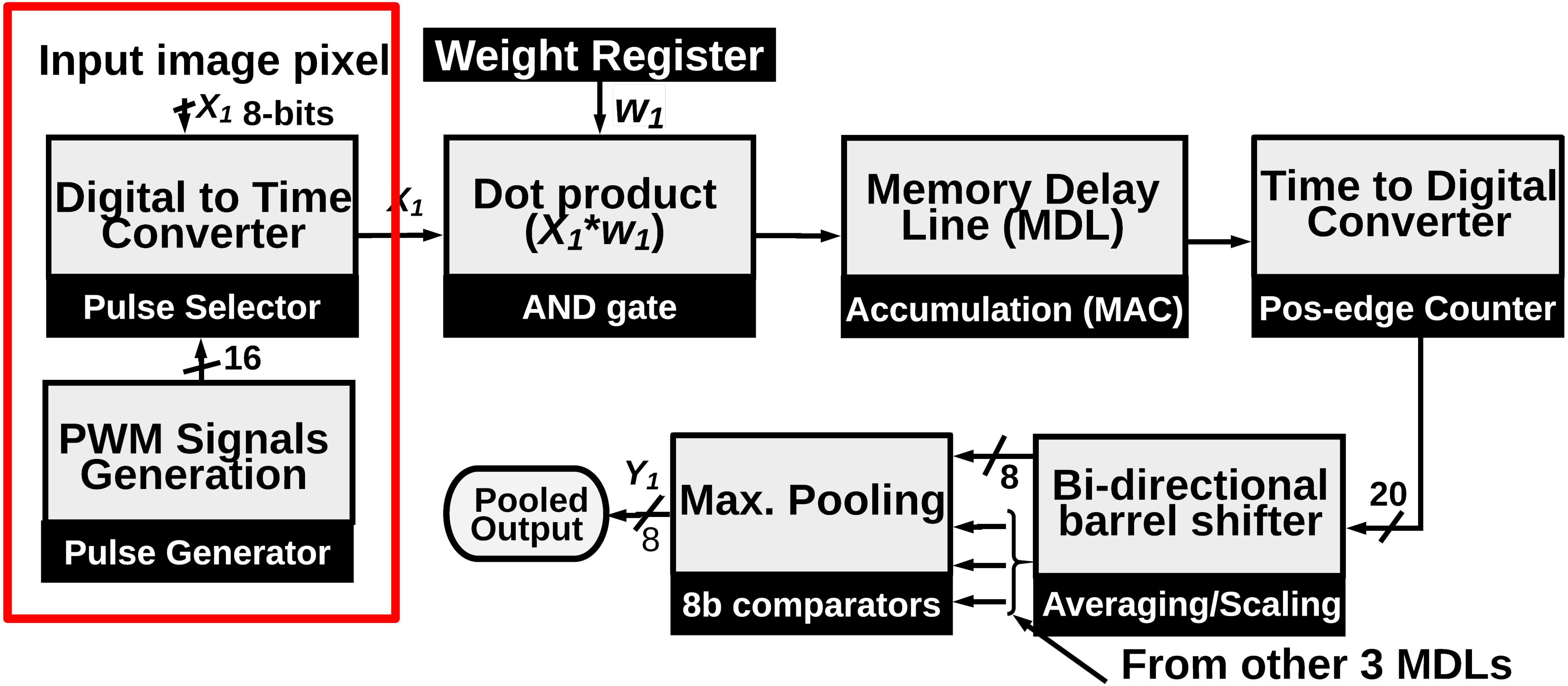
MAV - Multiply Accumulate Average

Implementing MAV operation

Architecture of Proposed CNN Engine

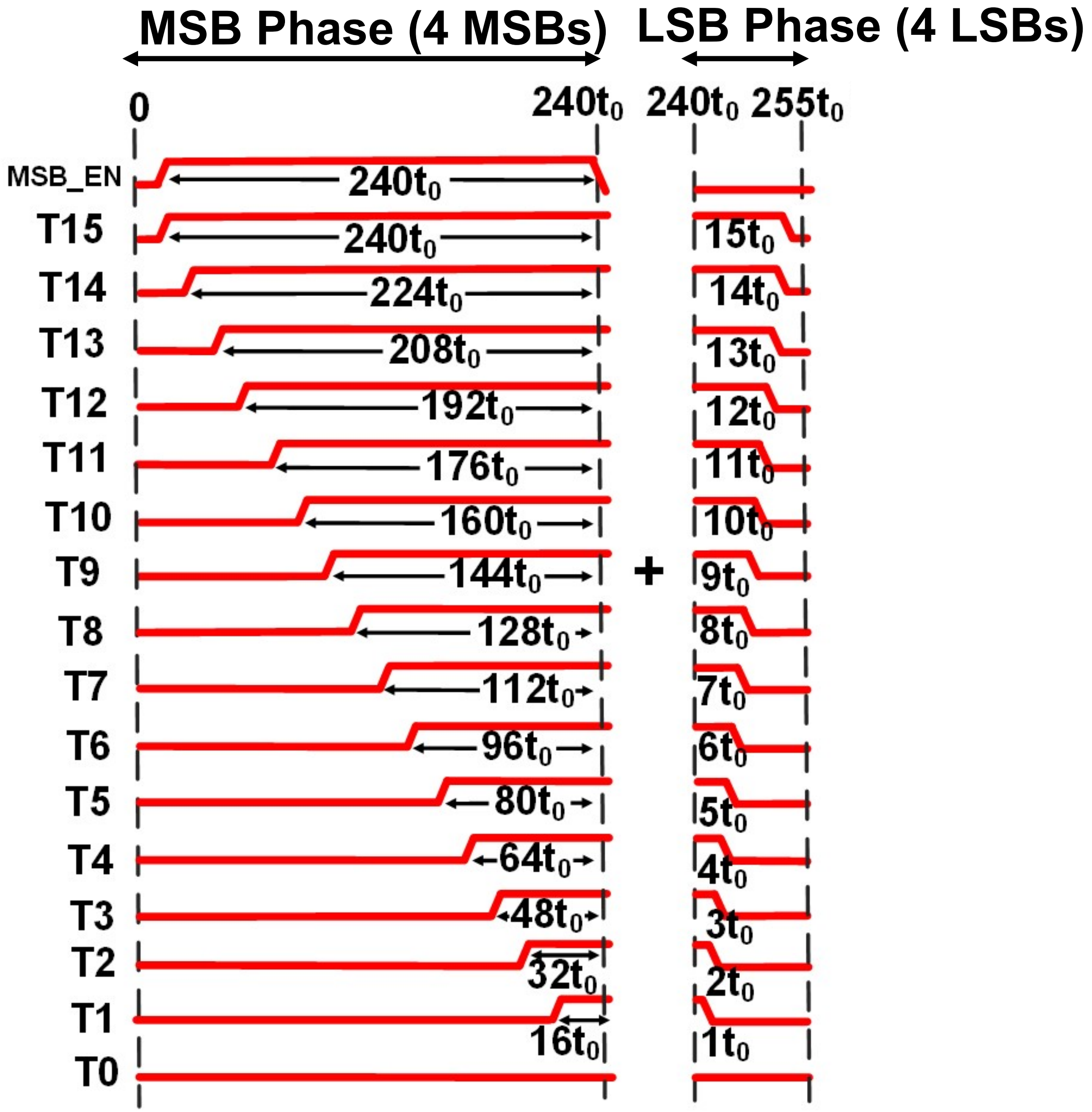


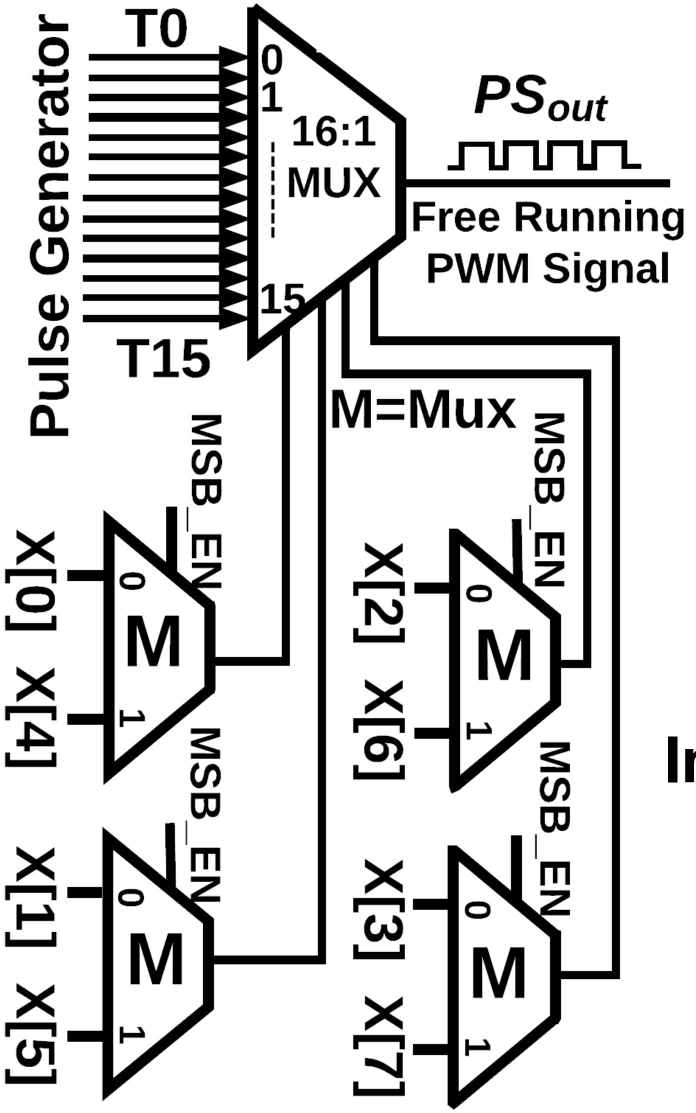
Architecture of Proposed CNN Engine



PWM Pulse Generation & Selection

PWM Signal Generation





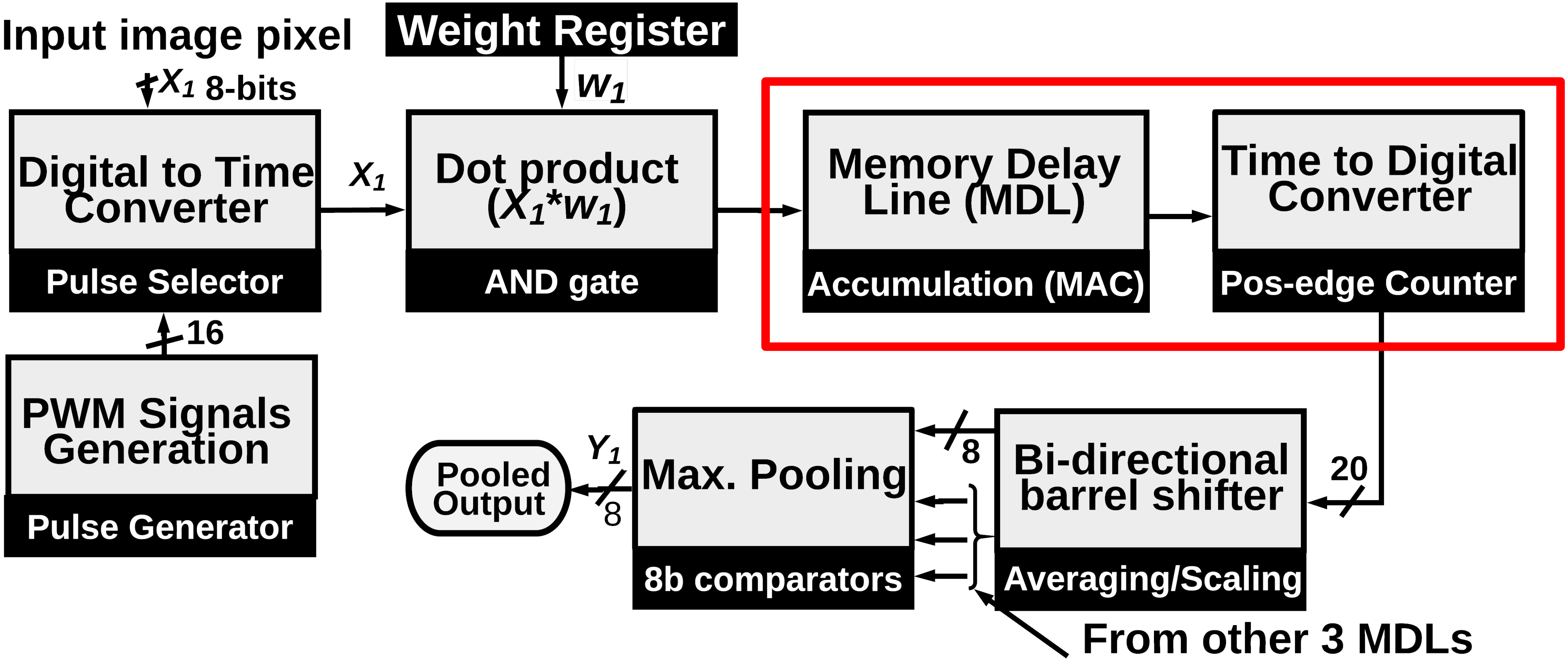
Digital to Time Converter (DTC)

Input: X[7:0]
Output: PS_{out}

PWM Signal selection based on input Image pixel value

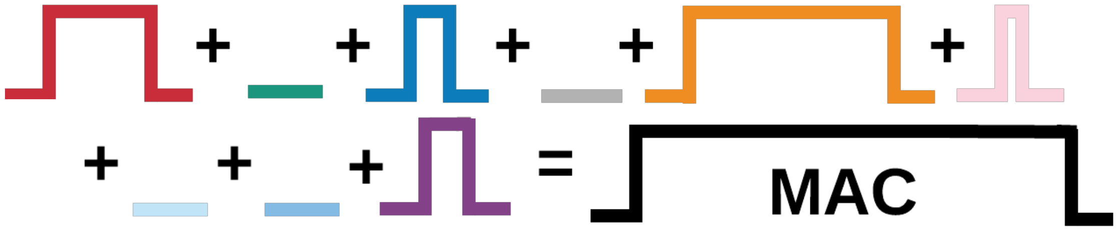
Ref:
A. Biswas et al., ISSCC'18 (MIT)

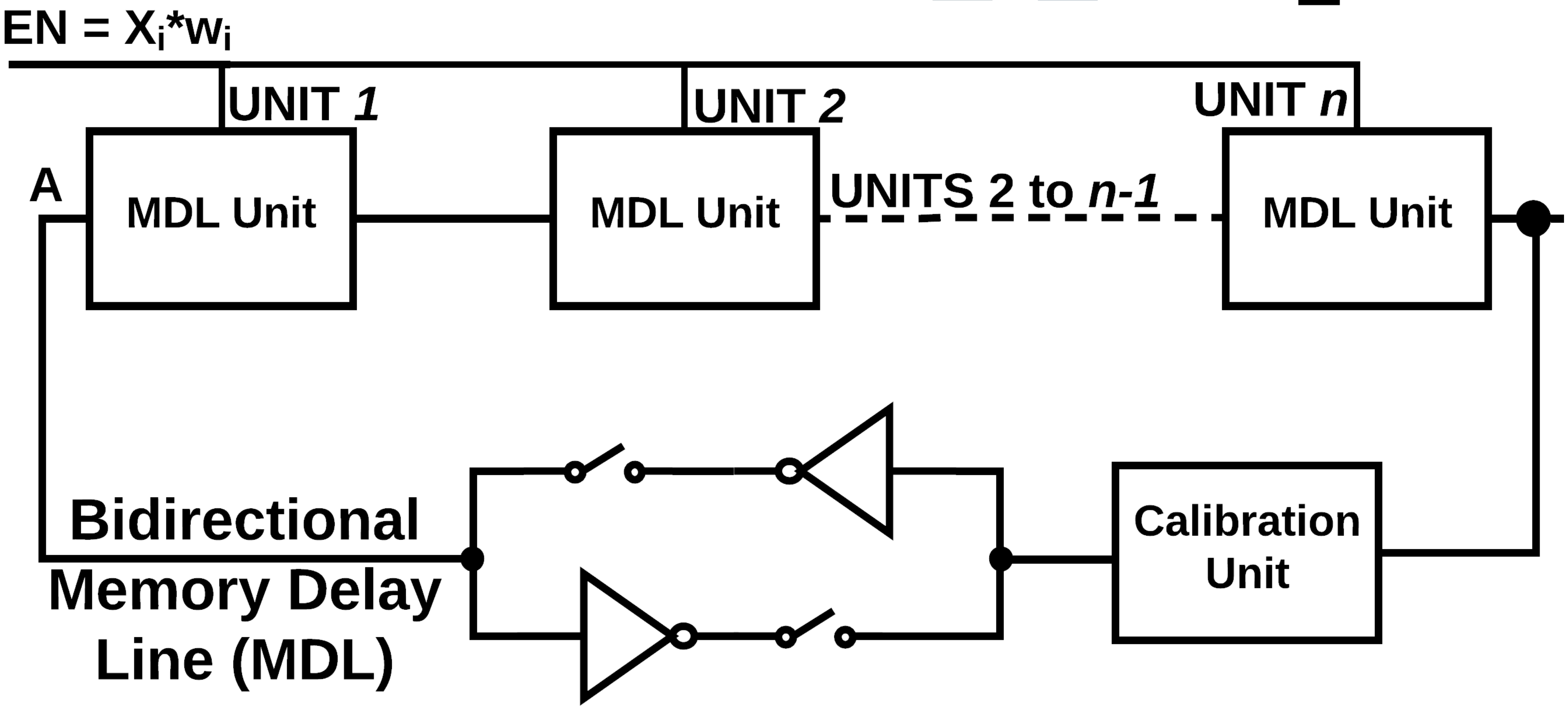
Architecture of Proposed CNN Engine



Bi-directional Memory Delay Line

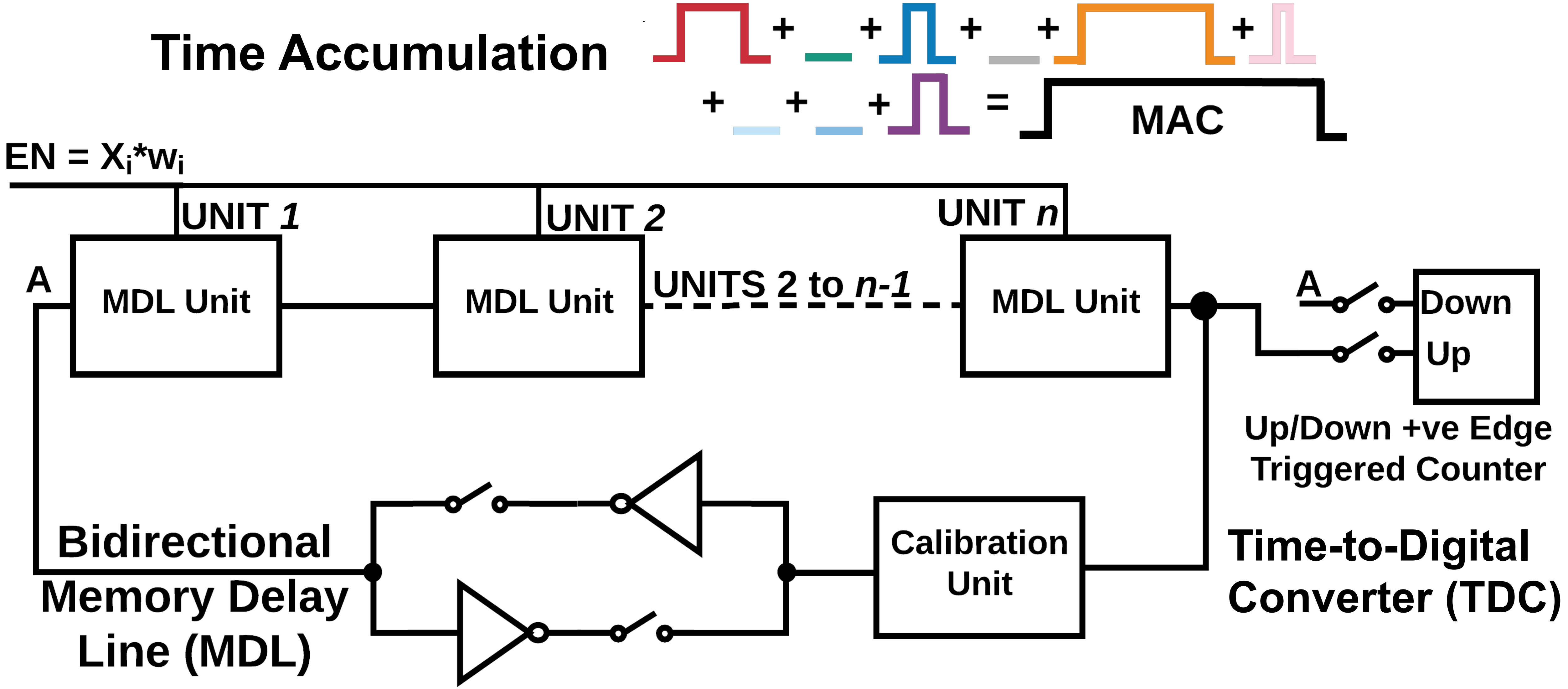
Time Accumulation



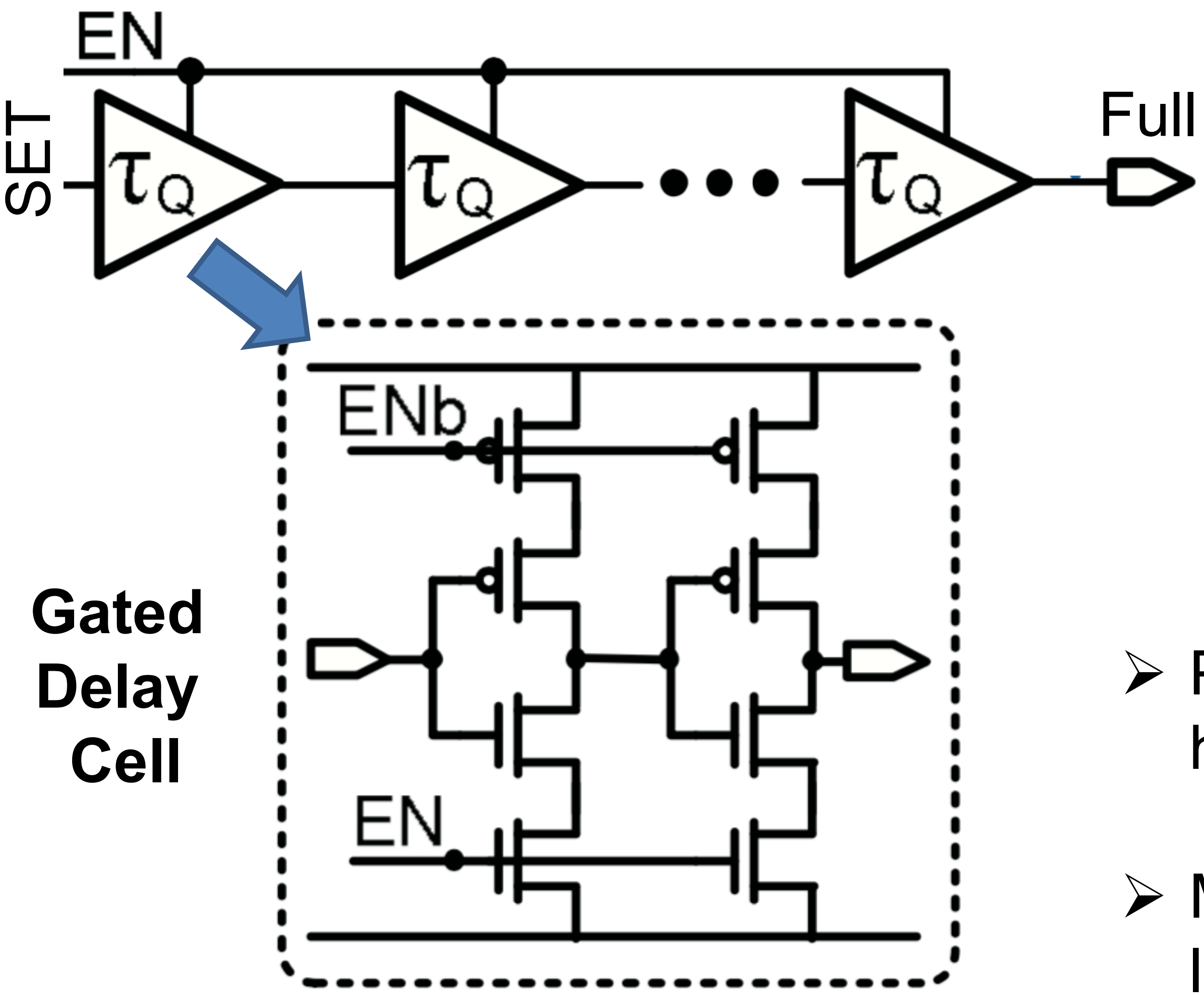


Bidirectional Memory Delay Line (MDL)

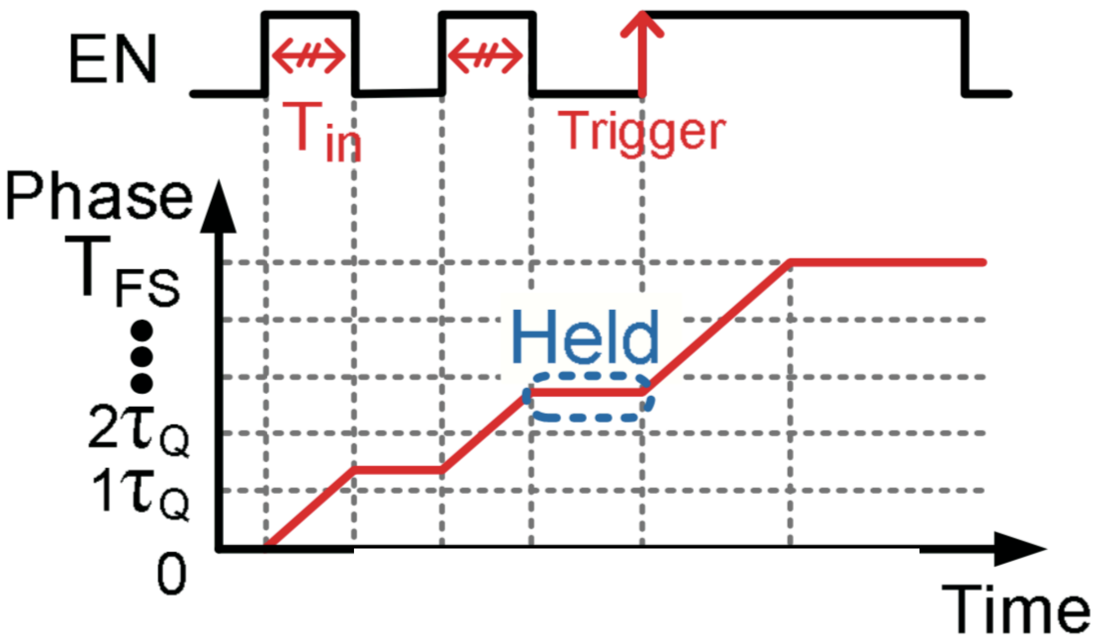
Bi-directional Memory Delay Line



MDL Approach inspired by Time Register

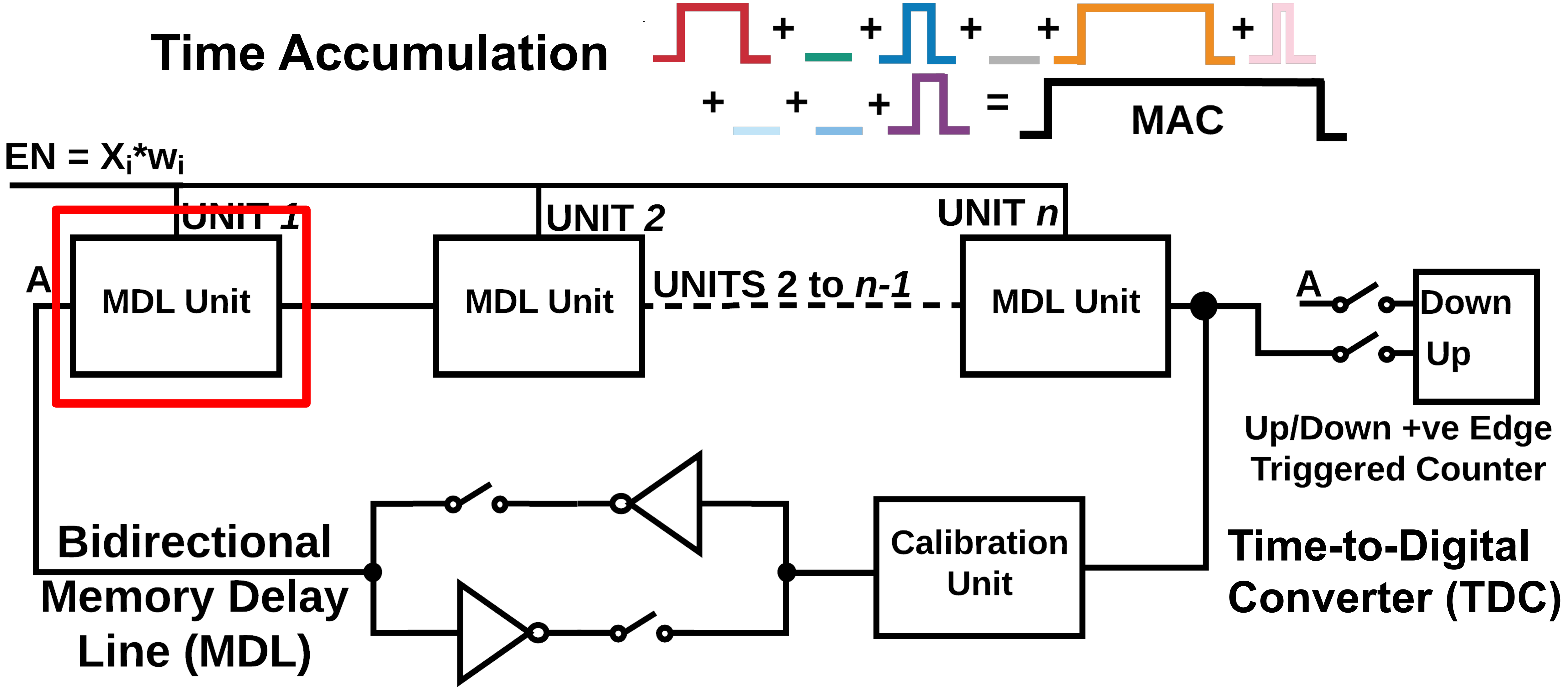


Ref: K. Kim et al., VLSIC'13

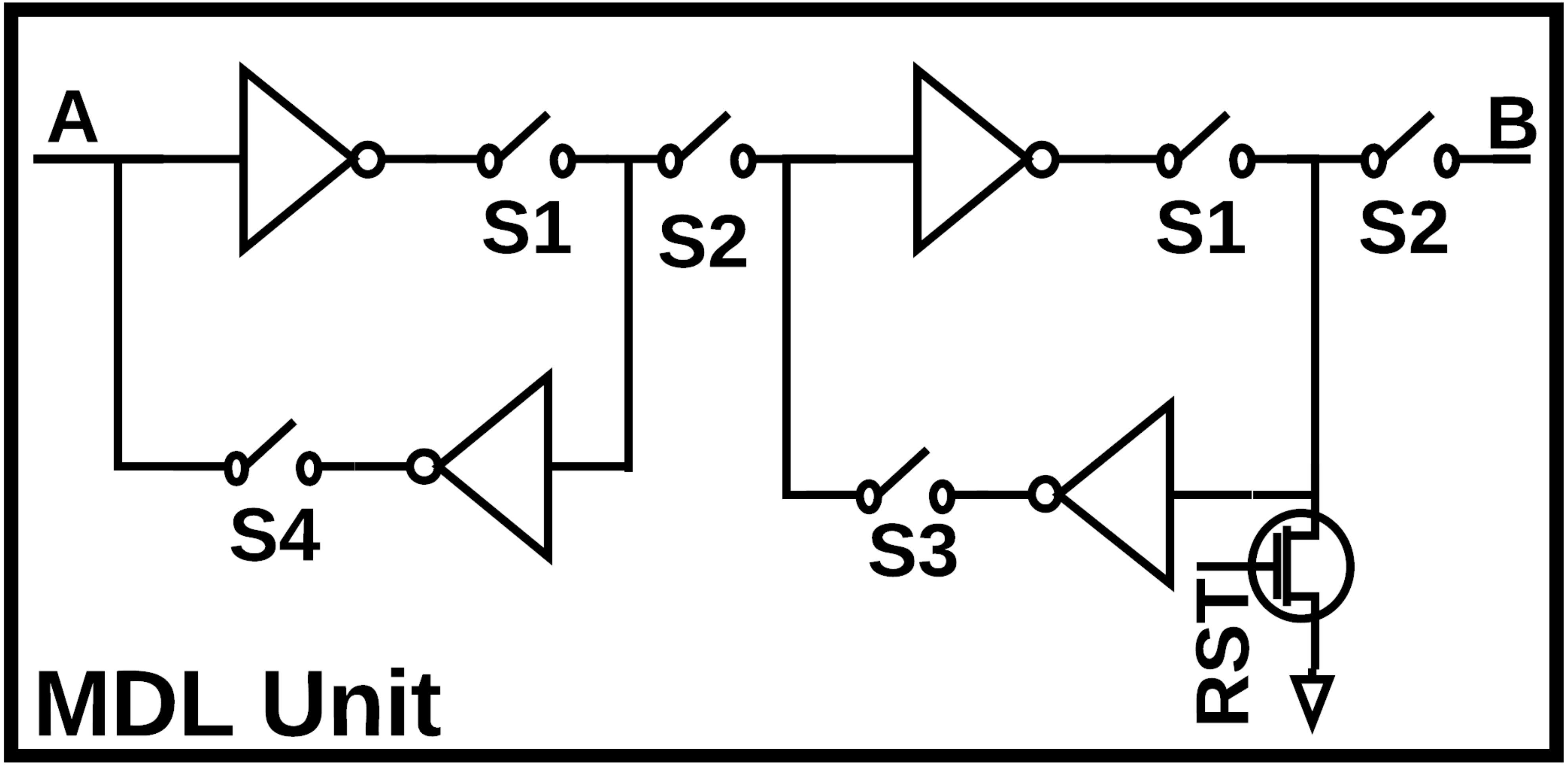


- ➢ Phase increases when EN is high, and is held when the EN is low
- ➢ Metastability risk ➔ tri-stated output when long period of EN disable

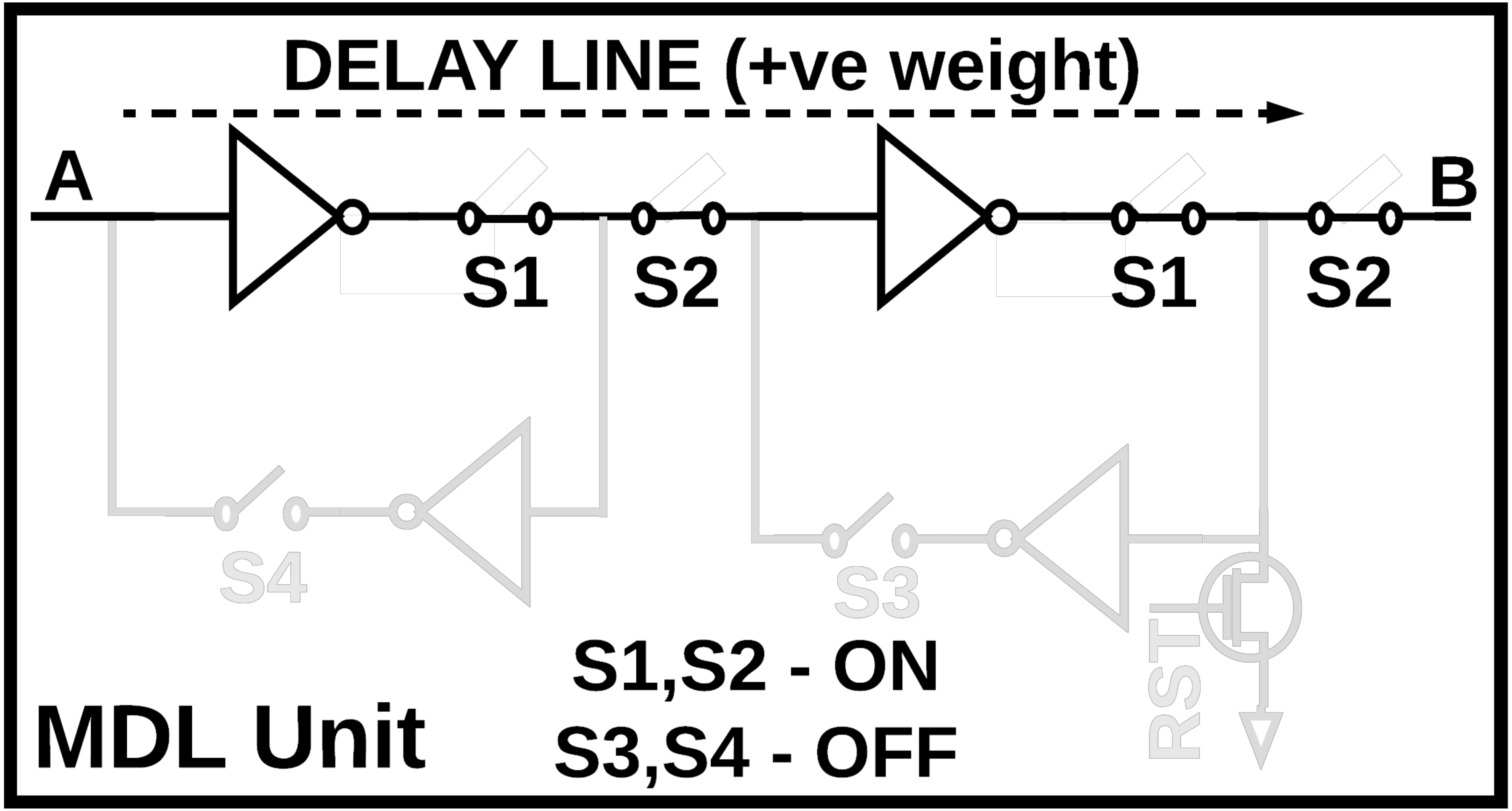
Bi-directional Memory Delay Line



Bi-directional Memory Delay Line Unit



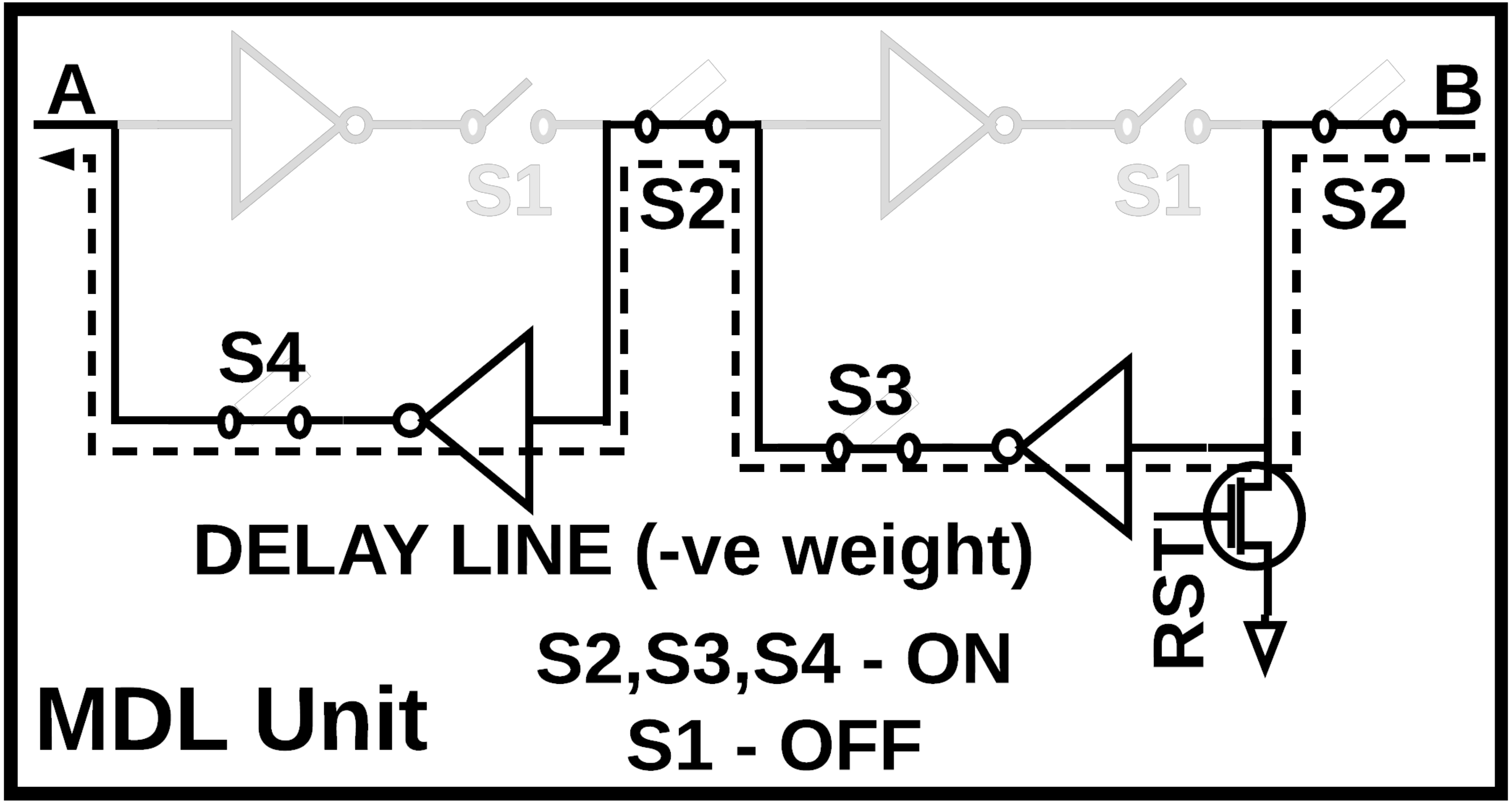
MDL Unit – Delay line (Positive weight)



S1,S2 - ON

S3,S4 - OFF

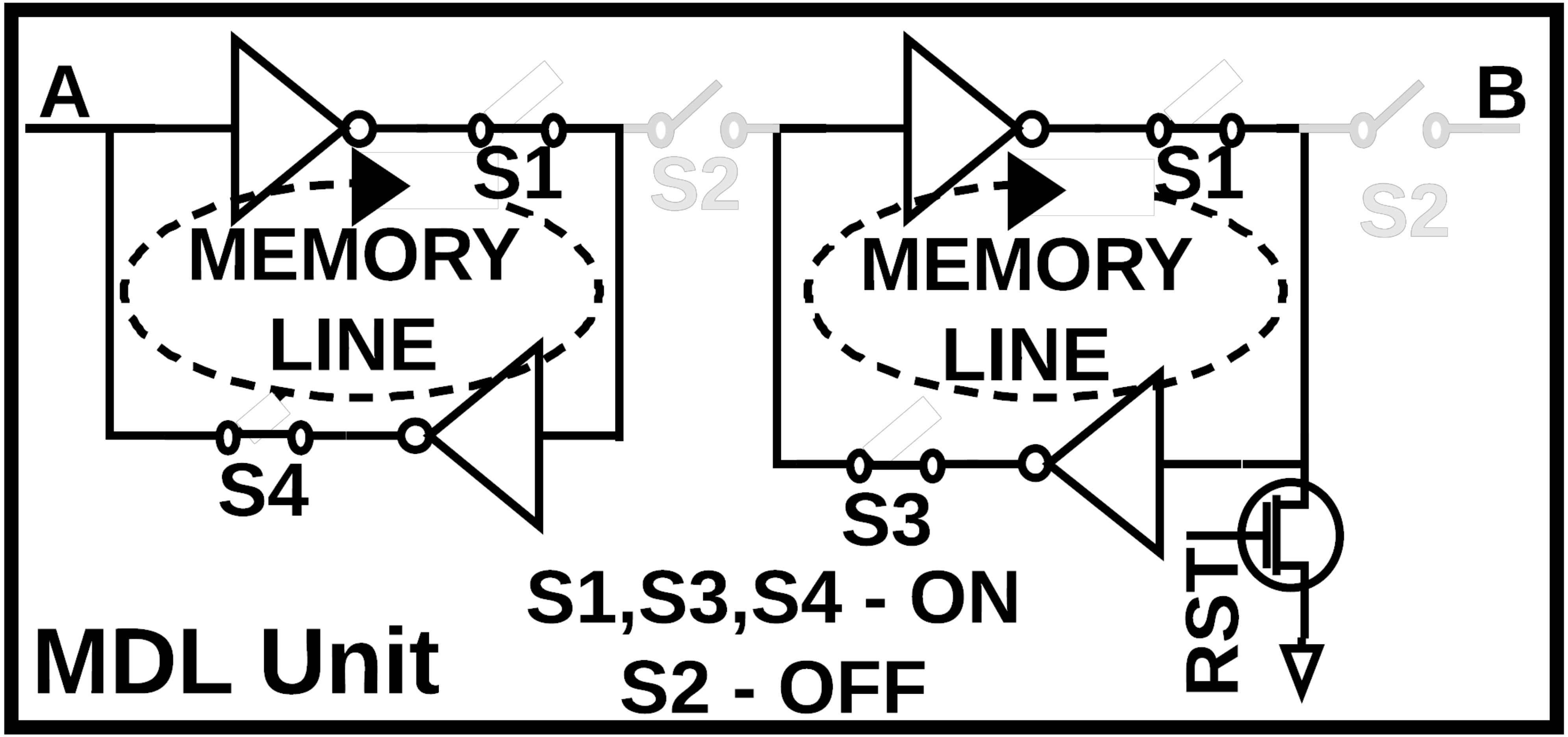
MDL Unit – Delay Line (Negative weight)



S2,S3,S4 - ON

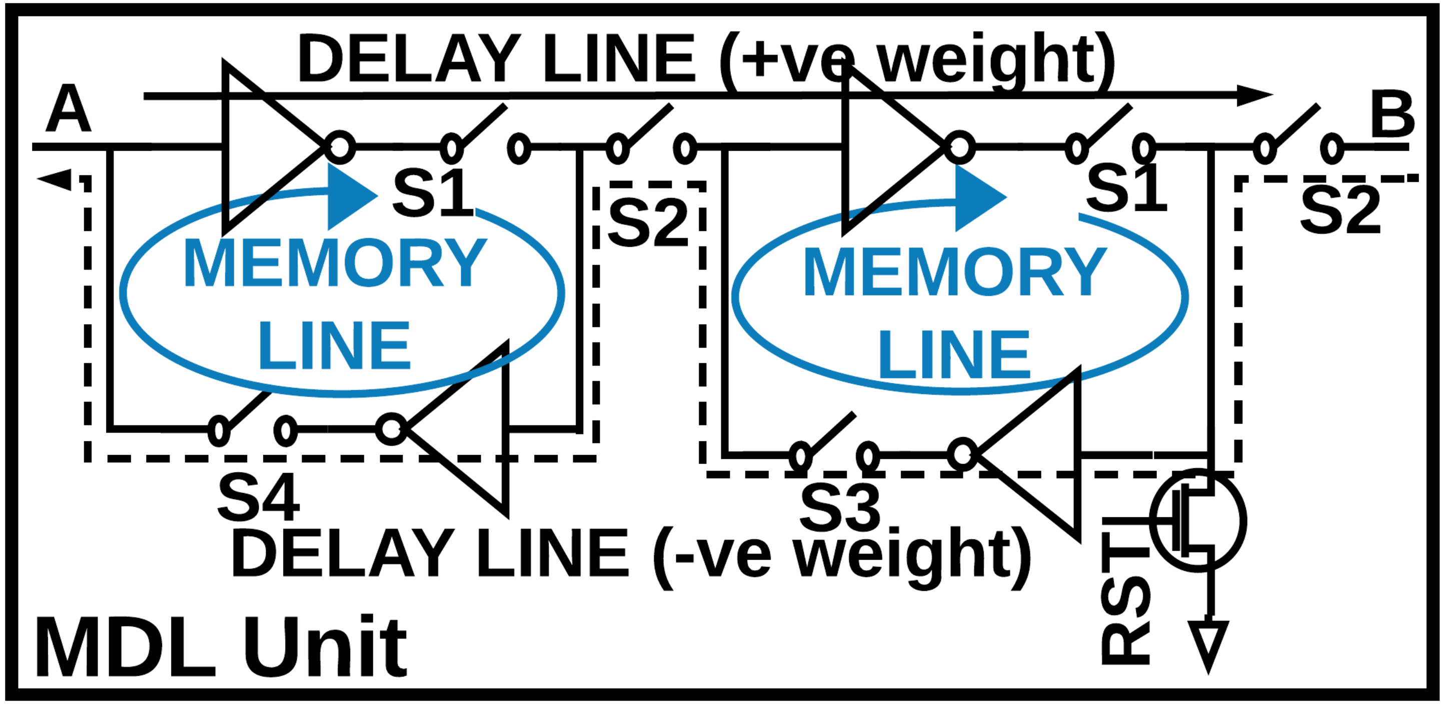
S1 - OFF

MDL Unit – Memory Line



No tristate cells ➔ State of the delay line is retained in memory phase

Bi-directional Memory Delay Line Unit

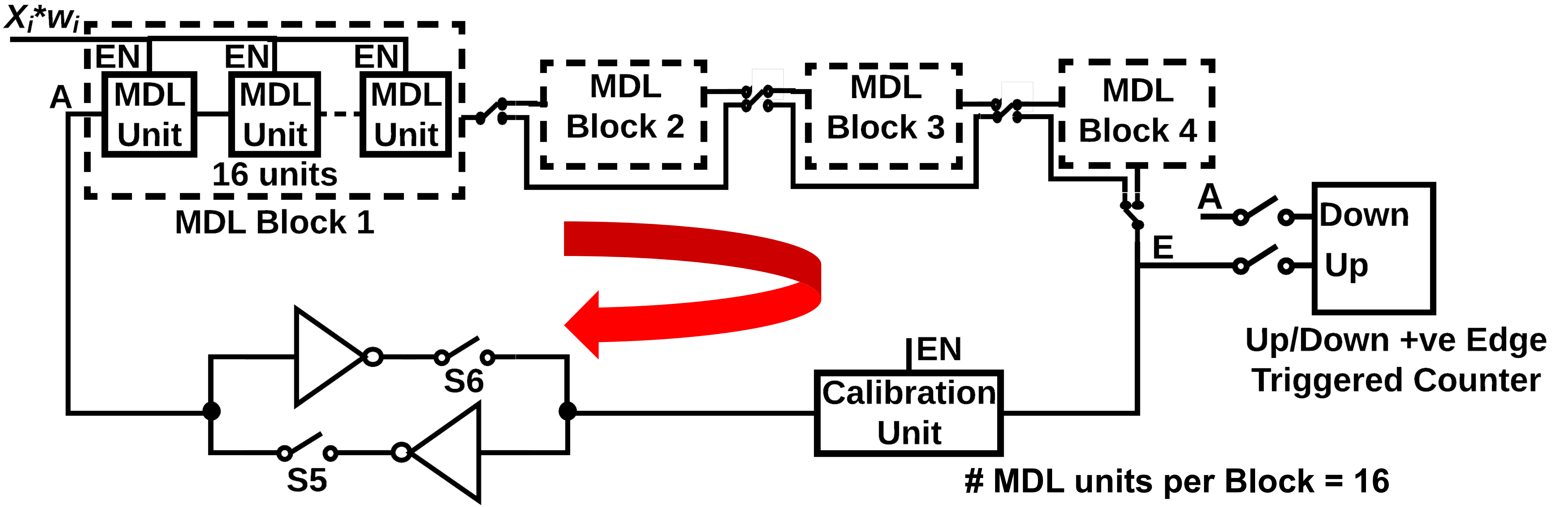


Switch	Value
S1	(EN).(SIGN)
S2	EN
S3	(EN') OR (EN).(SIGN)'
S4	(EN).(SIGN)'

if EN==1,
MDL → Delay Line
if EN==0,
MDL → Memory Line

if weight == +1,
SIGN = 1
if weight == -1,
SIGN = 0

Bi-directional MDL Implementation



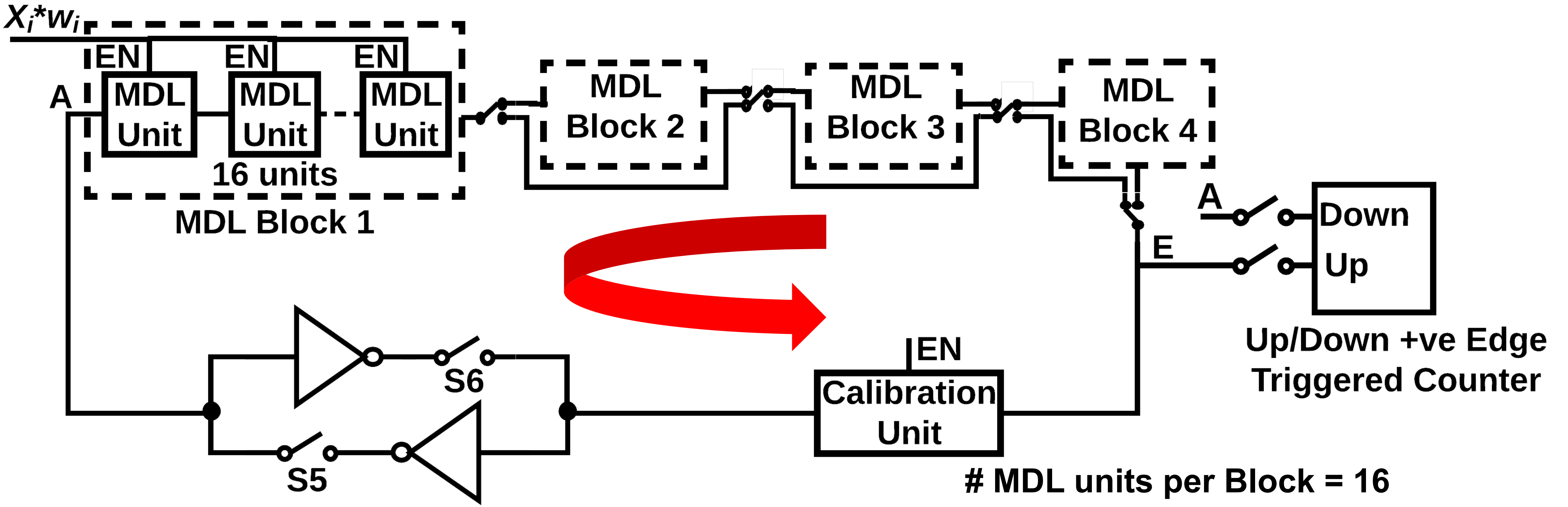
# MDL units per Block = 16
# MDL blocks in MDL = 1 or 2 or 3 or 4

Switch	Value	Weight
S5	SIGN	+ve
S6	(SIGN)'	-ve

➔ Clockwise propagation in MDL

➔ Anti-clockwise propagation in MDL

Bi-directional MDL Implementation



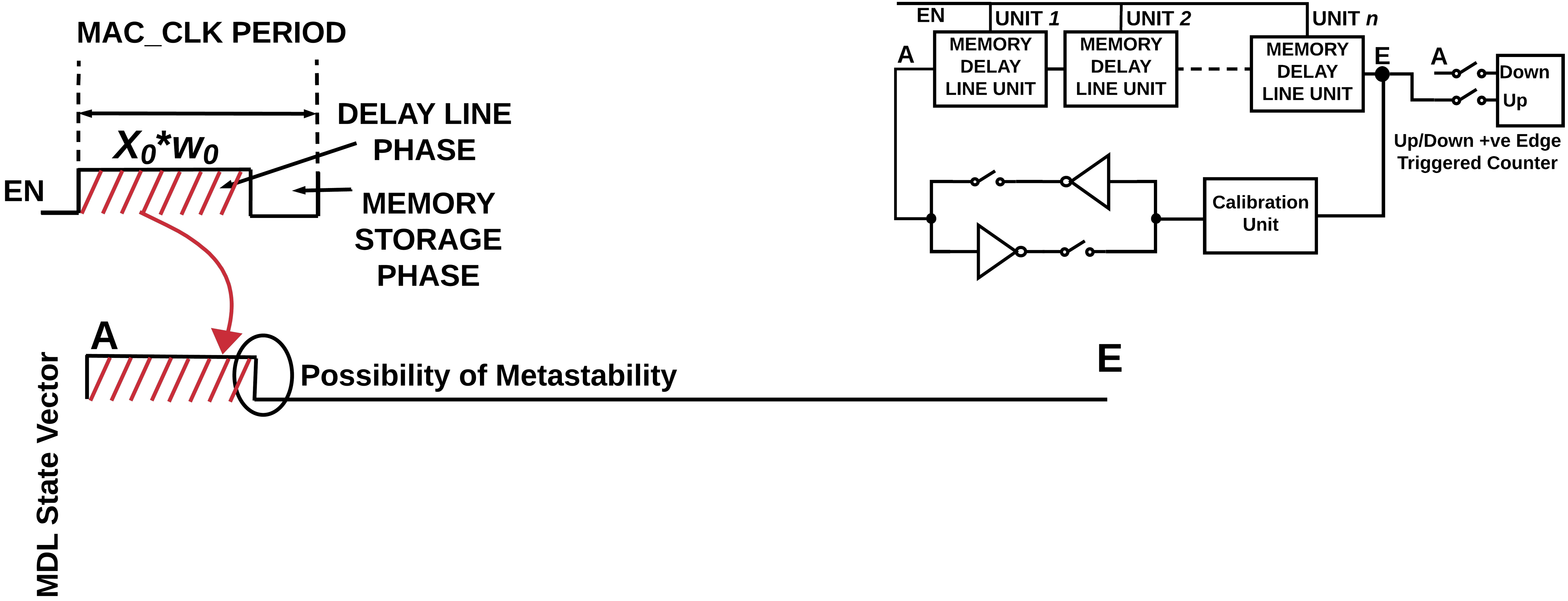
# MDL units per Block = 16
# MDL blocks in MDL = 1 or 2 or 3 or 4

Switch	Value	Weight
S5	SIGN	+ve
S6	(SIGN)'	-ve

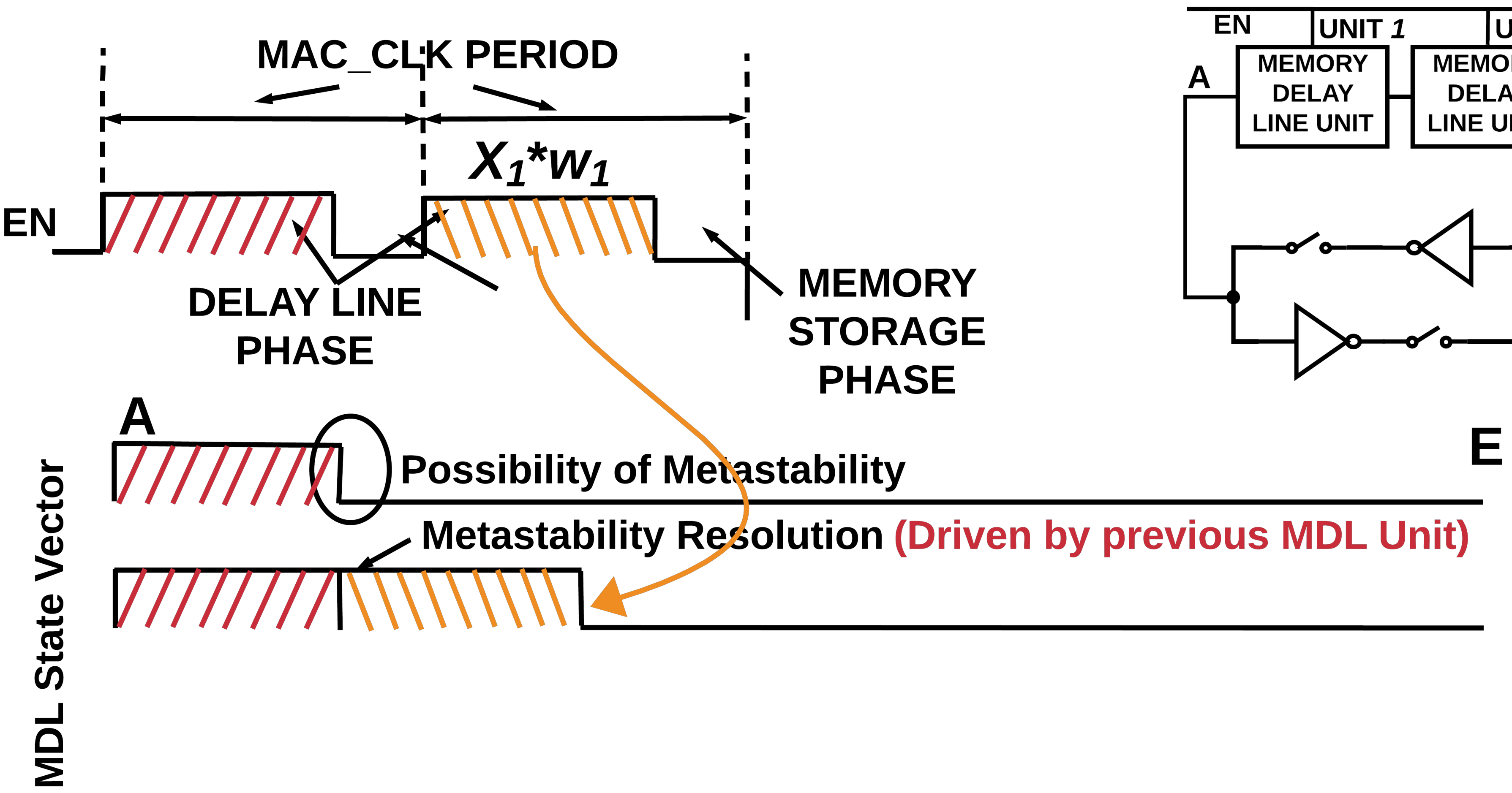
➔ Clockwise propagation in MDL

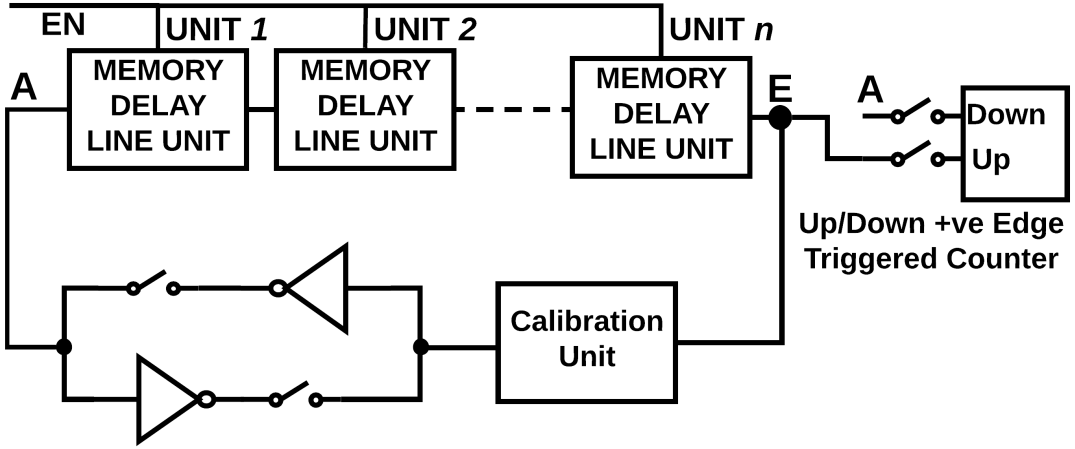
➔ Anti-clockwise propagation in MDL

MDL Timing Diagram

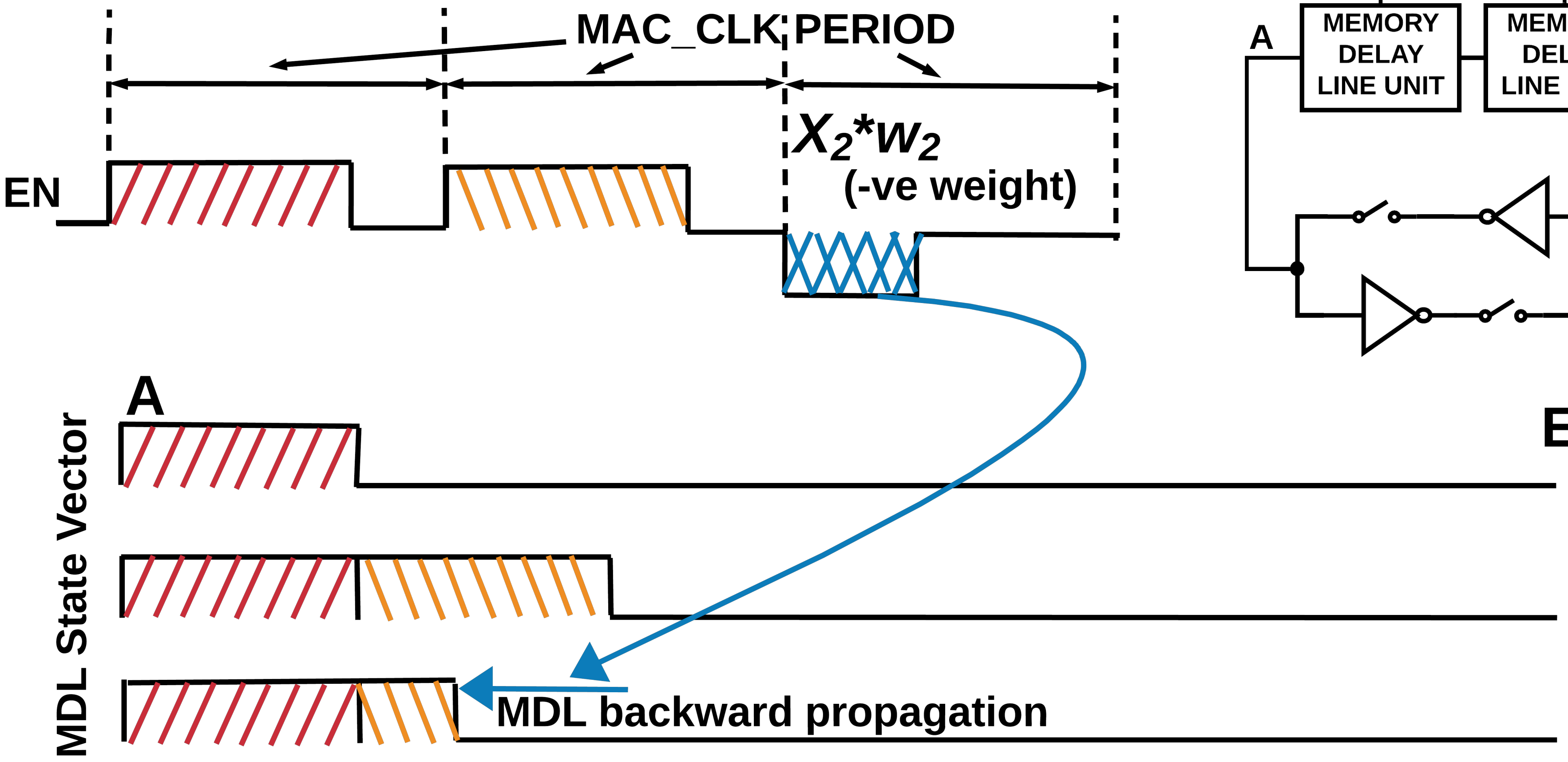


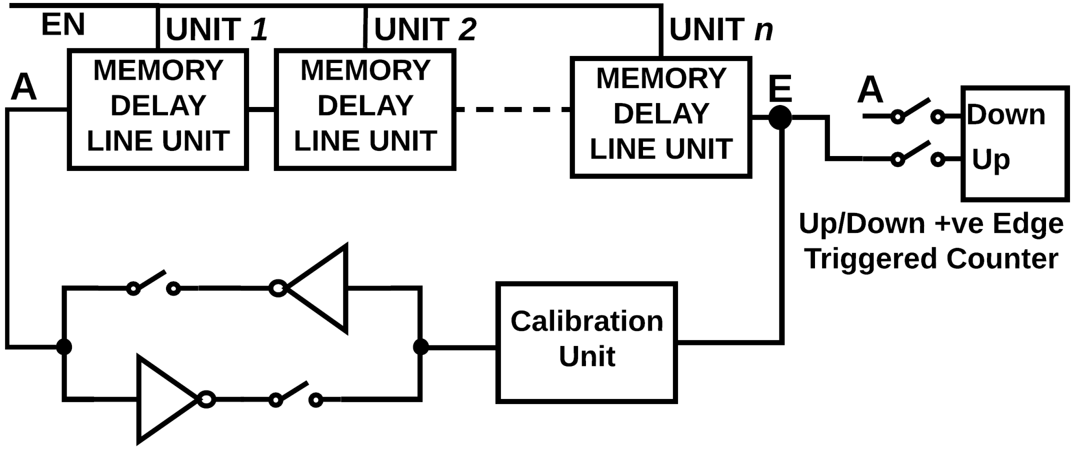
MDL Timing Diagram


MAC_CLK PERIOD
X_1*w_1
EN
DELAY LINE PHASE
MEMORY STORAGE PHASE
A
E
Possibility of Metastability
Metastability Resolution (Driven by previous MDL Unit)
MDL State Vector

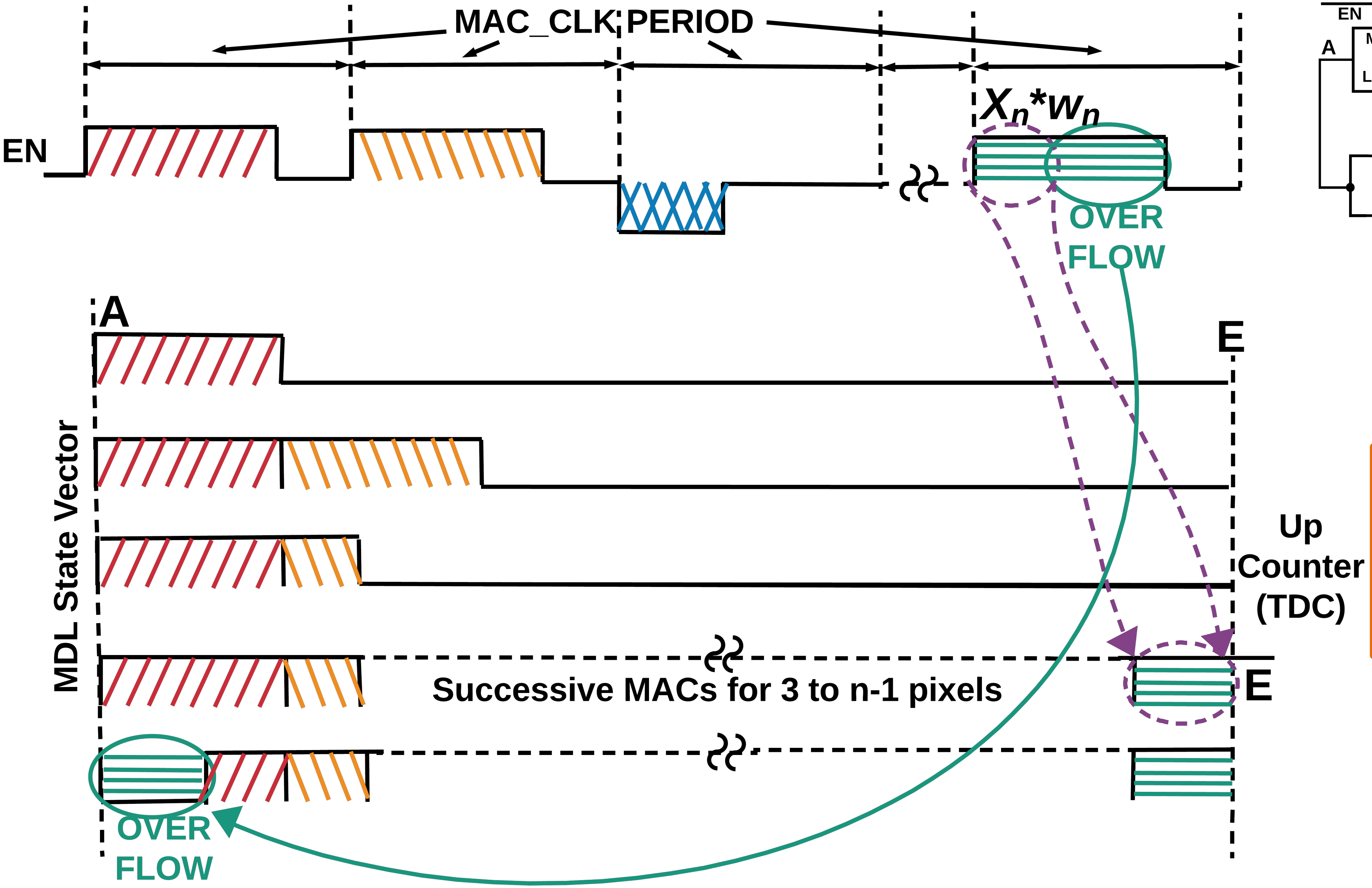


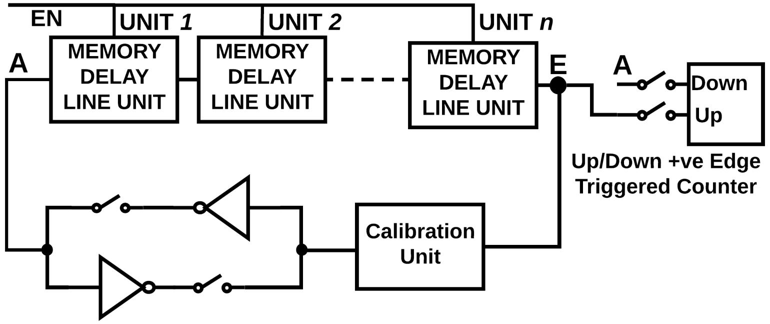
MDL Timing Diagram





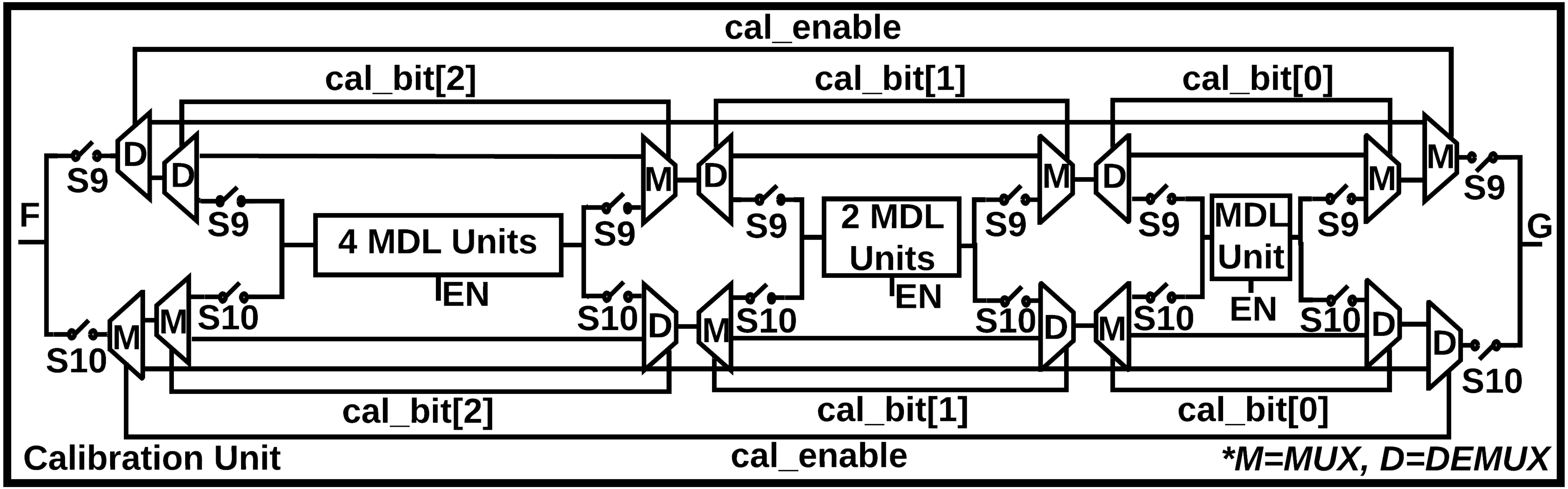
MDL Timing Diagram





Finite MDL length along with up/down counter helps to perform long periods of time accumulation

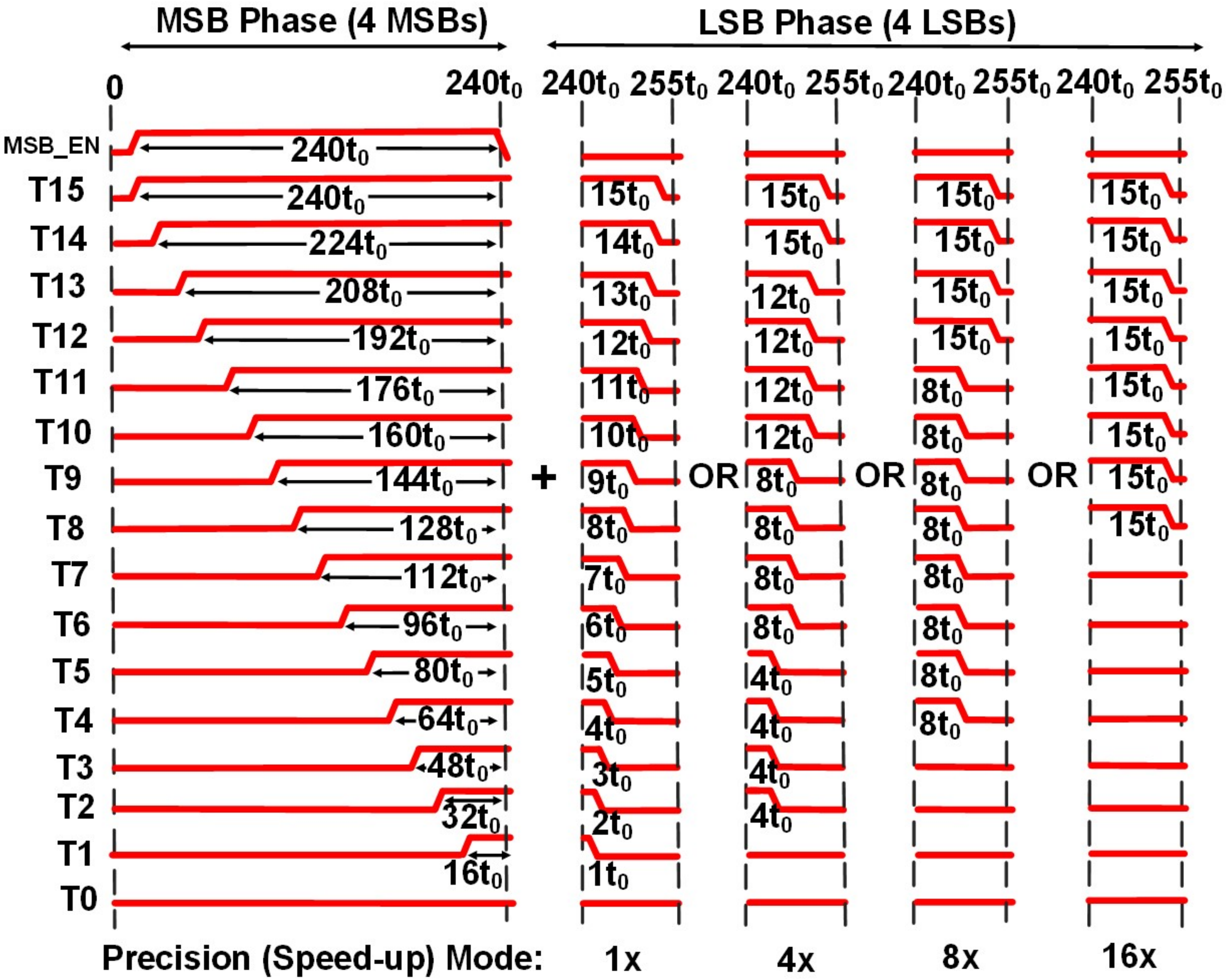
Mitigating Process Variations



Switch S9 value == SIGN, Switch S10 value == (SIGN)'

- Calibration unit is added to all the 4 MDLs for each filter. Due to process variations, delay mismatch among 4 MDLs for each filter are compensated by by tuning *cal_bit[0:2]* and *cal_enable* signals.

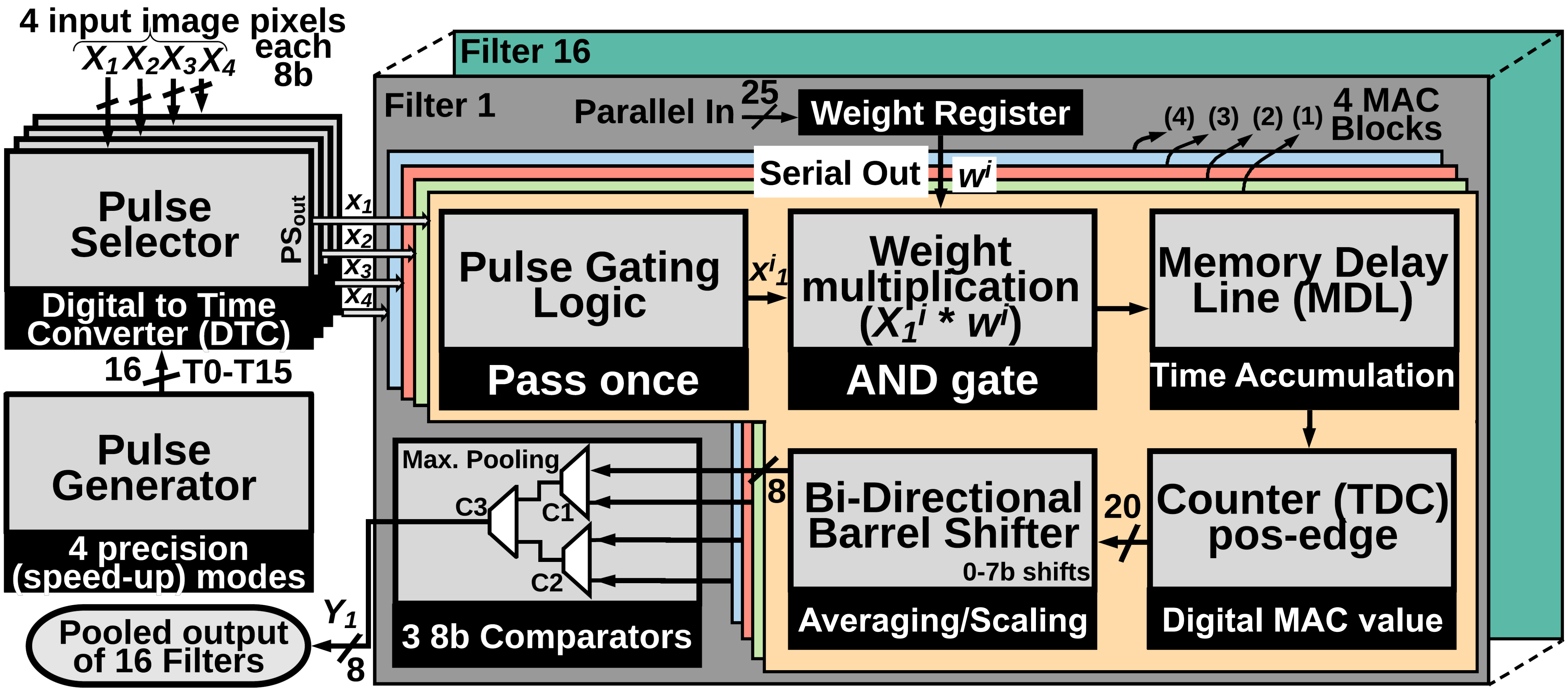
Increasing Throughput: Speedup modes



MSB Phase (4 MSBs)
LSB Phase (4 LSBs)
0
240t0
240t0 255t0 240t0 255t0 240t0 255t0 240t0 255t0
MSB_EN
T15
T14
T13
T12
T11
T10
T9
T8
T7
T6
T5
T4
T3
T2
T1
T0
240t0
240t0
224t0
208t0
192t0
176t0
160t0
144t0
128t0
112t0
96t0
80t0
64t0
48t0
32t0
16t0
+
15t0 14t0 13t0 12t0 11t0 10t0 9t0 8t0 7t0 6t0 5t0 4t0 3t0 2t0 1t0
OR
15t0 15t0 12t0 12t0 12t0 12t0 8t0 8t0 8t0 8t0 4t0 4t0 4t0 4t0
OR
15t0 15t0 15t0 15t0 8t0 8t0 8t0 8t0 8t0 8t0 8t0 8t0
OR
15t0 15t0 15t0 15t0 15t0 15t0 15t0 15t0
Precision (Speed-up) Mode:
1x
4x
8x
16x

Speedup Mode	1x	4x	8x	16x
Input clock period	$2t_o$	$8t_o$	$16t_o$	$32t_o$
MAC clock period	$256t_o$	$256t_o$	$256t_o$	$256t_o$
# input clock cycles/MAC clock cycle	128	32	16	8
Quantization error (input: 0-255)	0	±2	±4	±8

Proposed CNN Engine Implementation



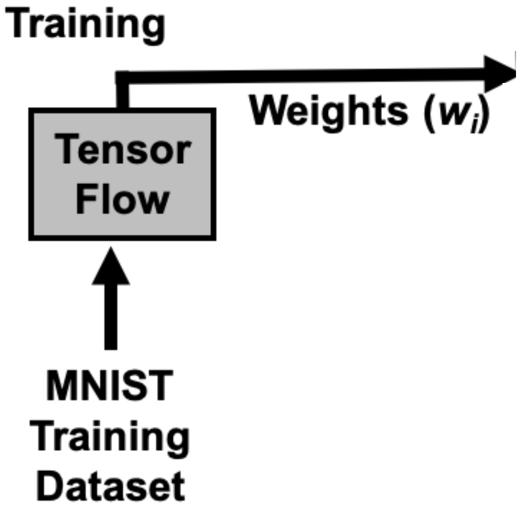
# MAC Blocks per Filter = 4

# Filters = 16

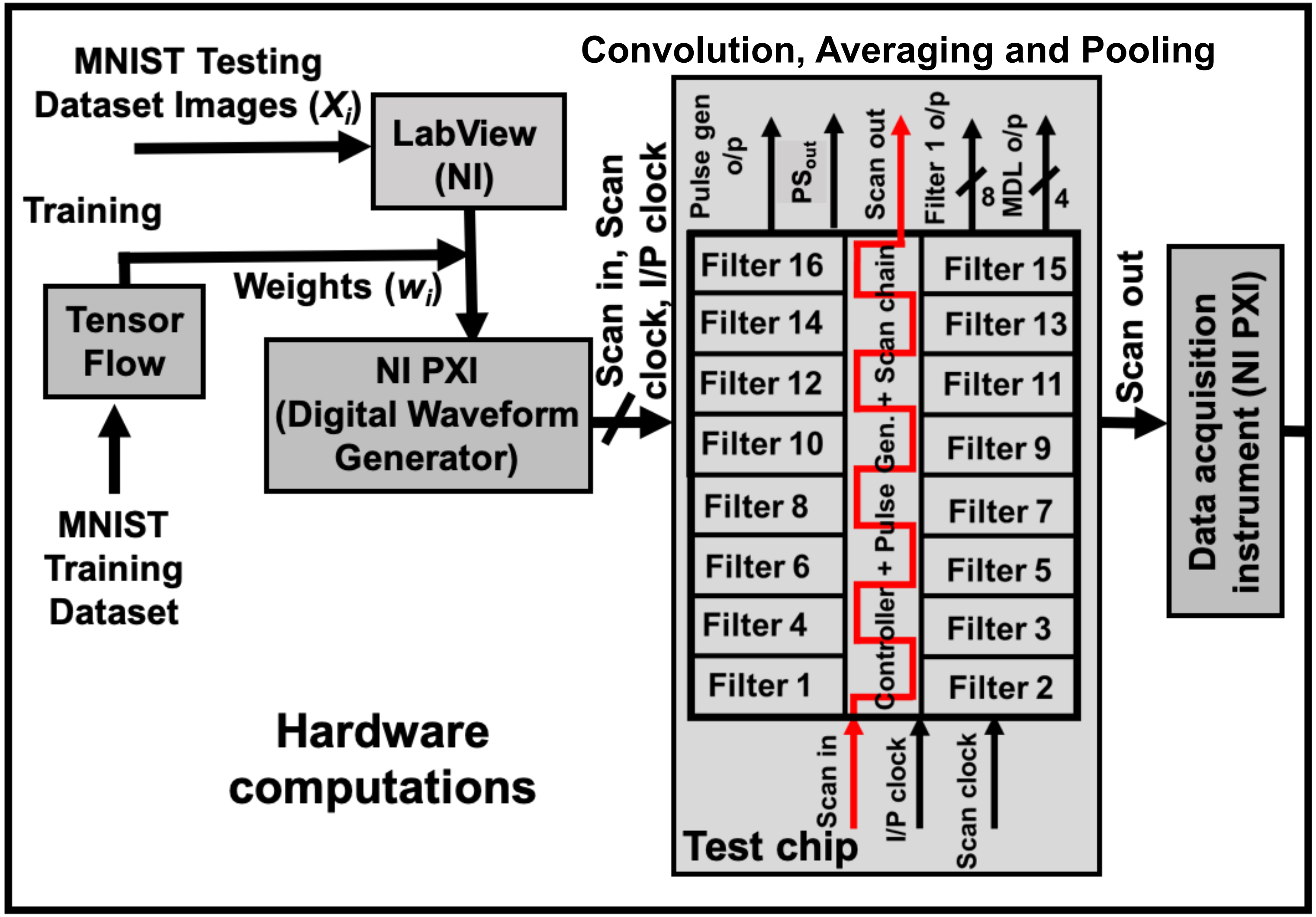
Outline

- Motivation
- Concept of Time-domain MAC
- Proposed CNN Engine Architecture
- Chip Implementation and Measurements
- Comparison with Prior Work
- Summary

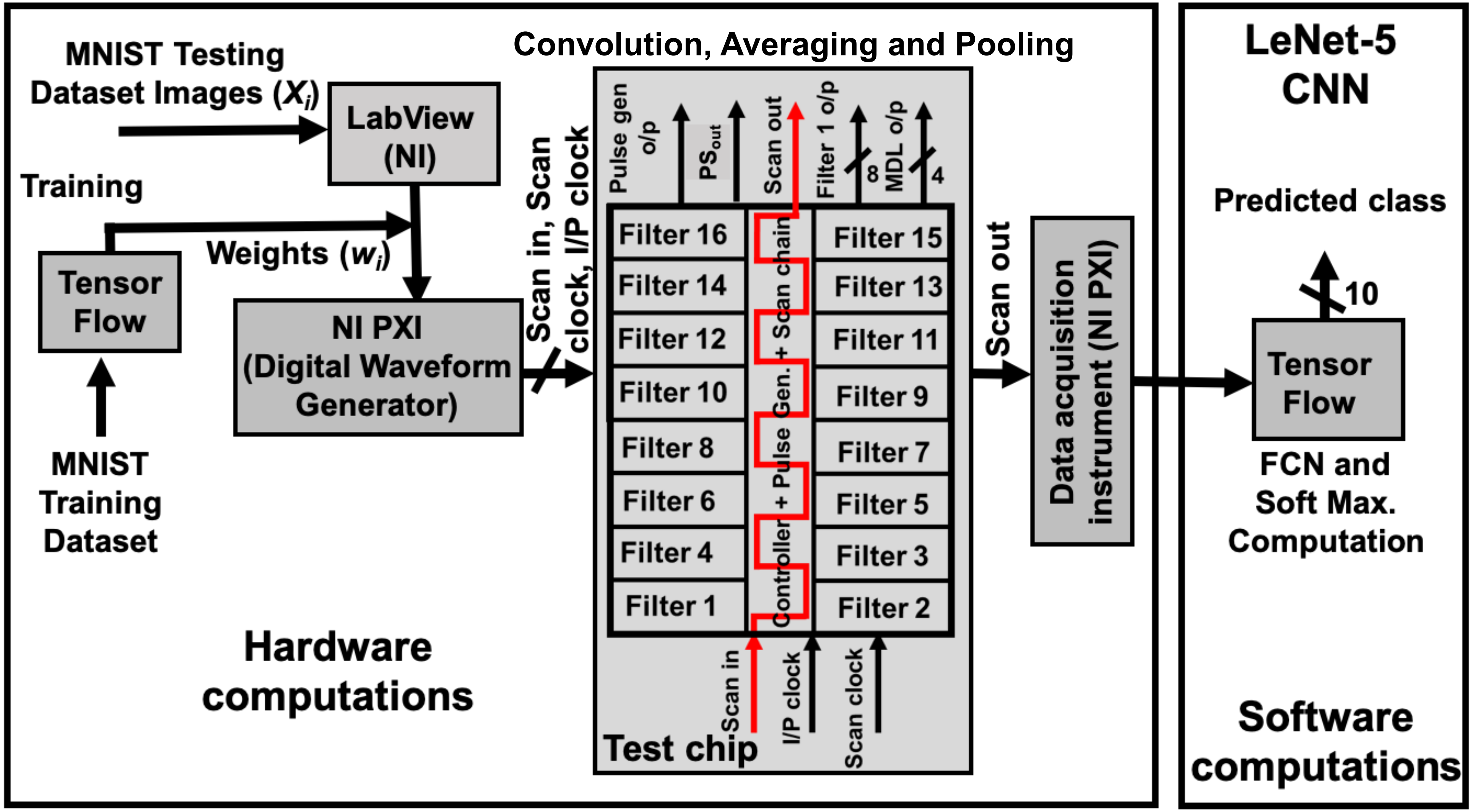
Overall Dataflow Diagram



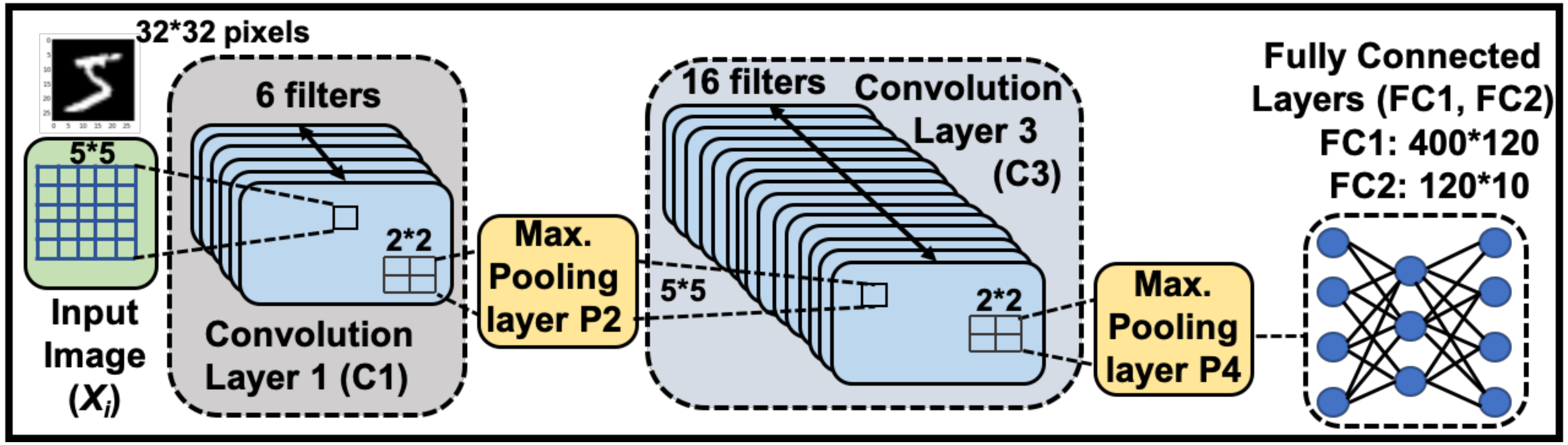
Overall Dataflow Diagram



Overall Dataflow Diagram



LeNet-5 Architecture & Parameters



Ref: LeCun et al., Gradient-based learning applied to document recognition. Proc. Of IEEE, 1998

Parameters for LeNet-5	C1	C3
Filter Size	5*5*1*6	5*5*6*16
Input/Filter bit width	8bits/ 1bit	8bits/ 1bit
Input Size	32*32*1	14*14*6
Output Size	14*14*6	5*5*16
#Filters	6	16
#Operations/convolution*	(25*4*6)*2	(150*4* 16)*2

**Assuming 1 Multiply-Accumulate-Average, and 1 Pooling = 2 operations*

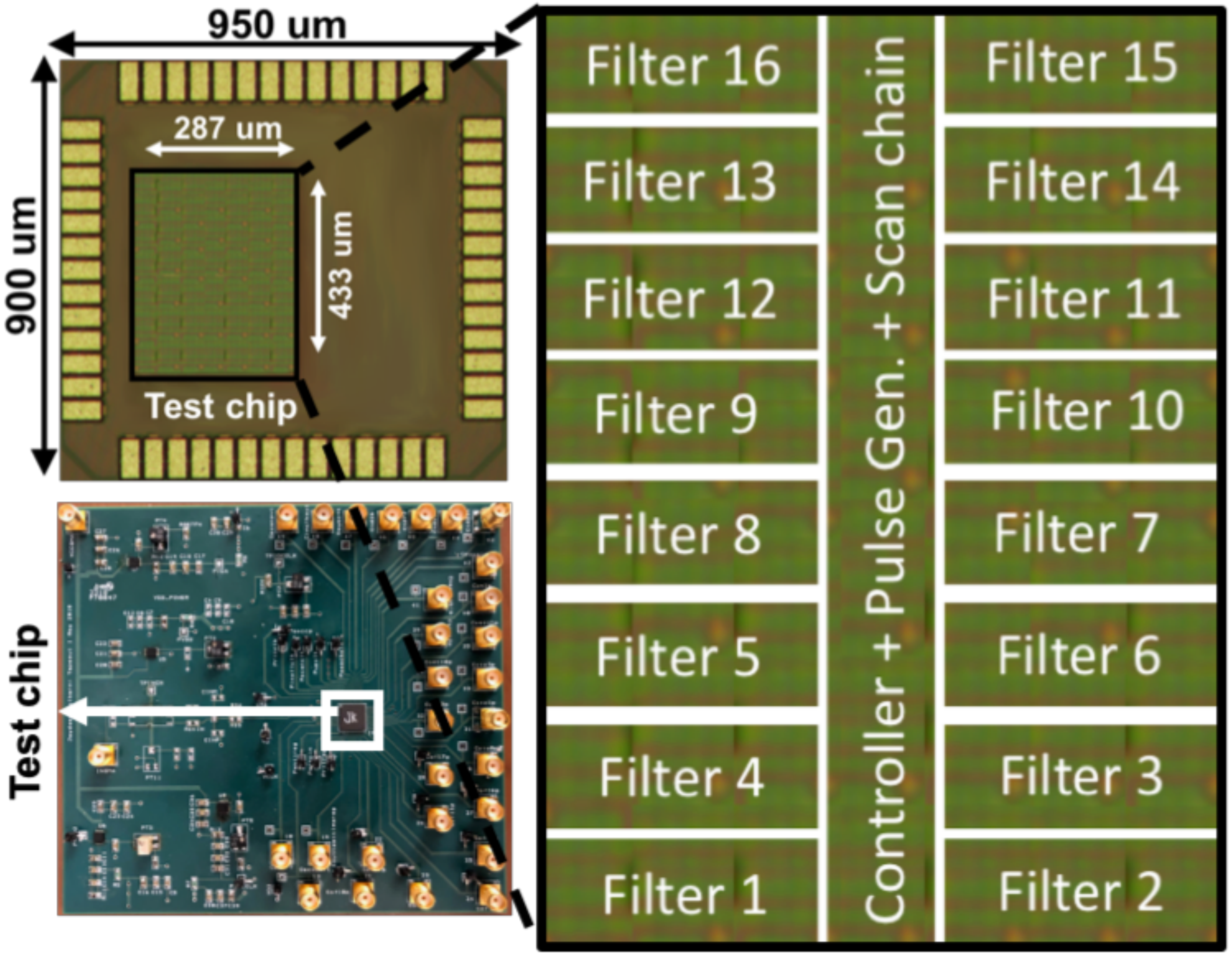
Hardware computations

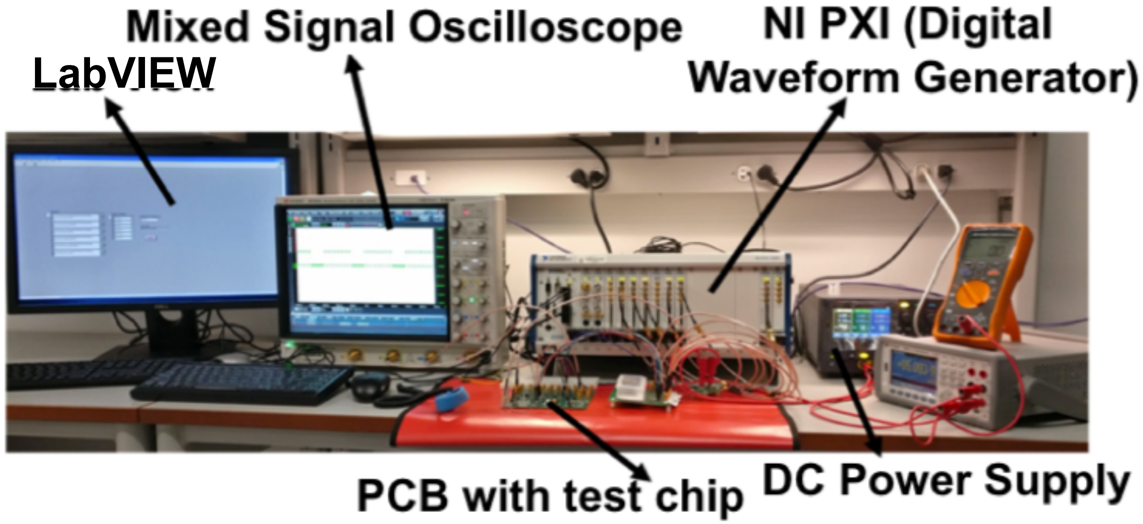
Convolution, Averaging and Pooling (Fixed Point)

Software computations

FCN (16 bit Floating Point)

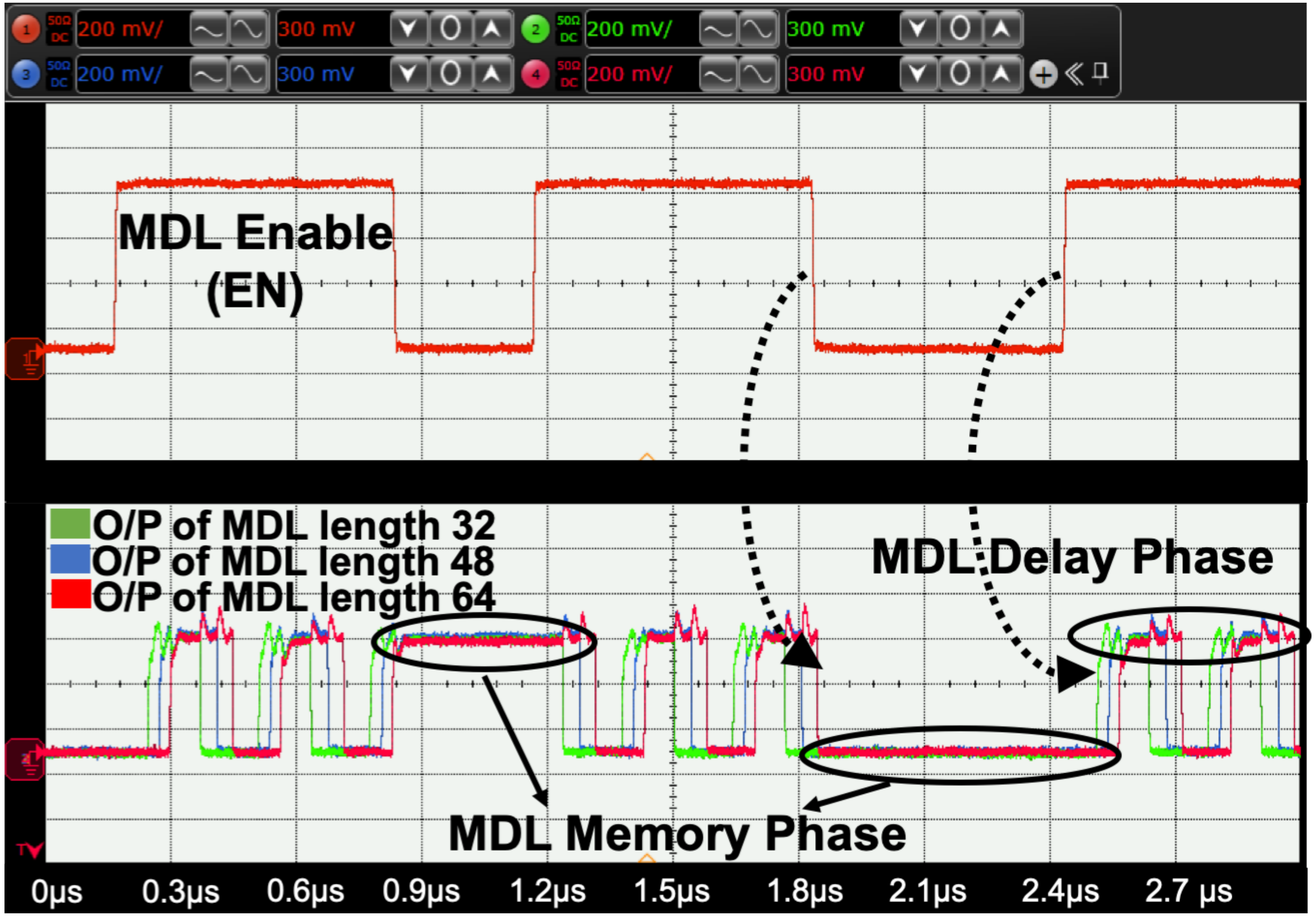
40nm test-chip die micrograph




Mixed Signal Oscilloscope
NI PXI (Digital Waveform Generator)
LabVIEW
PCB with test chip
DC Power Supply

Technology	40nm CMOS
Area	0.124mm^2
# standard cells	~ 40,000
# of filters supported	16
Data type supported	Signed/Unsigned
Voltage Range	0.375V - 1.10 V

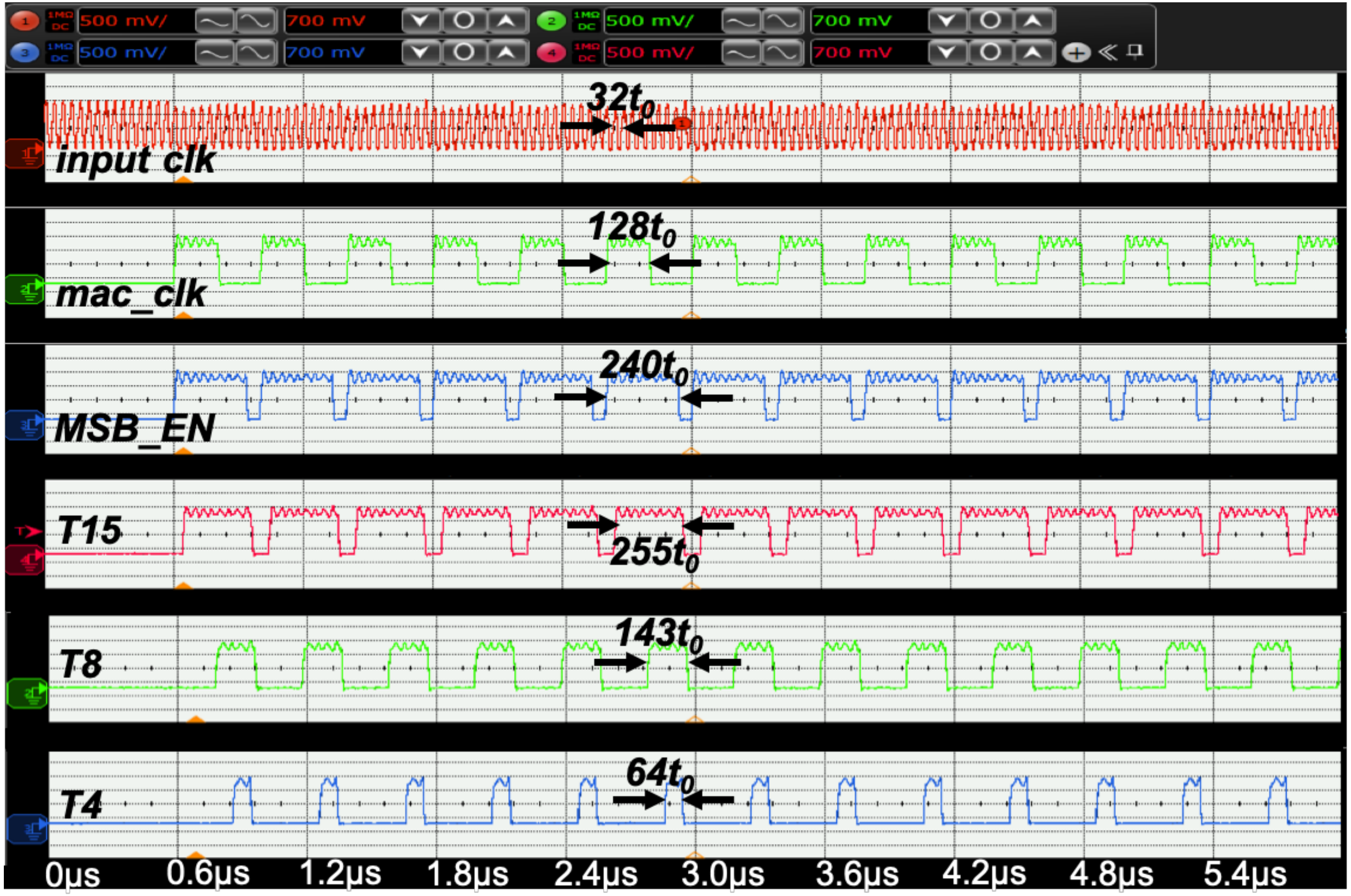
MDL Functionality Demonstration



- **EN == 1 ➔** MDL acts as a delay line
- **EN == 0 ➔** MDL acts as a memory line

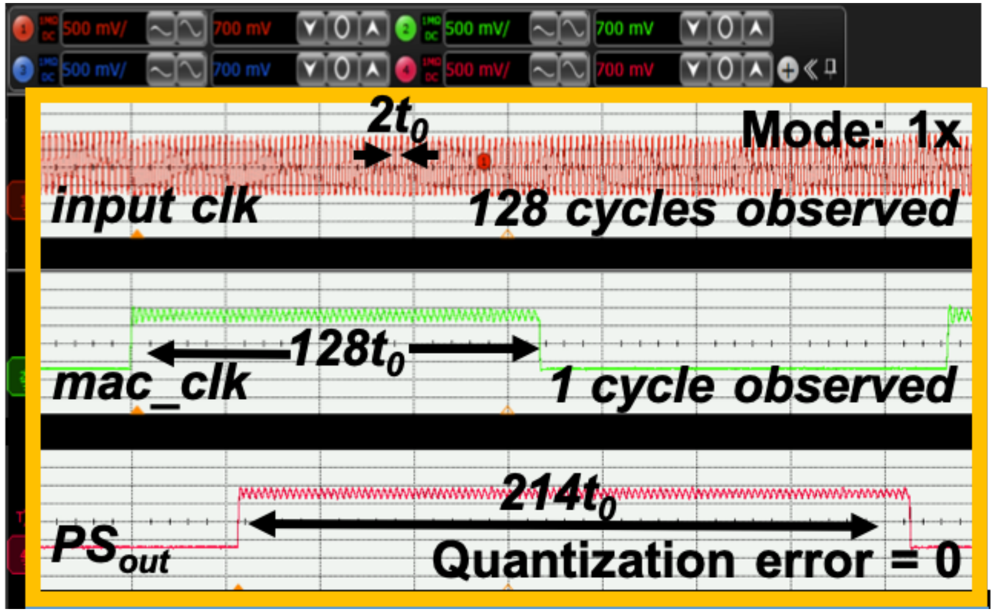
Pulse Generator Functionality Demonstration

Measured waveforms in 16x speedup mode



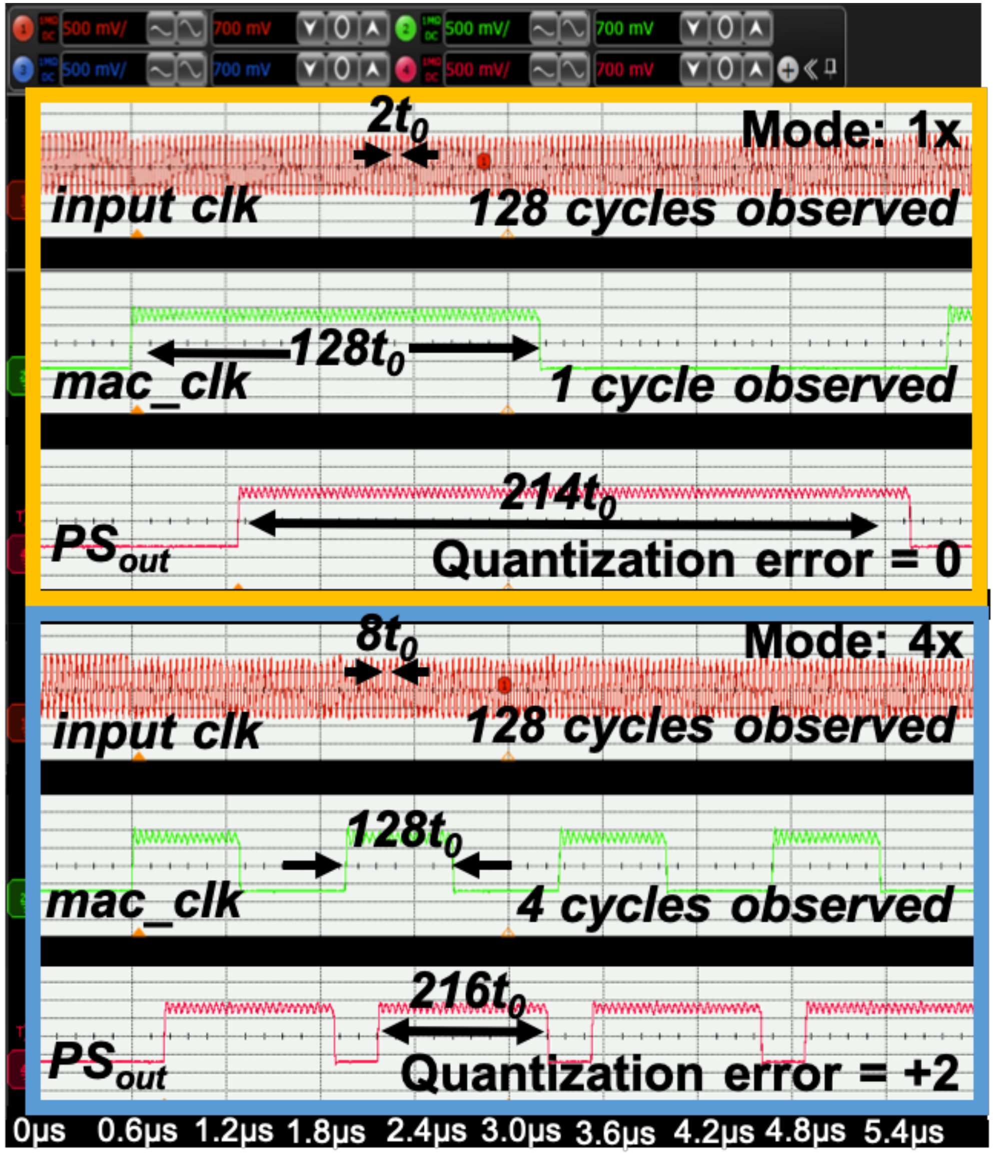
Speedup (1x-16x) Modes Demonstration

Measured waveforms for image pixel value 214



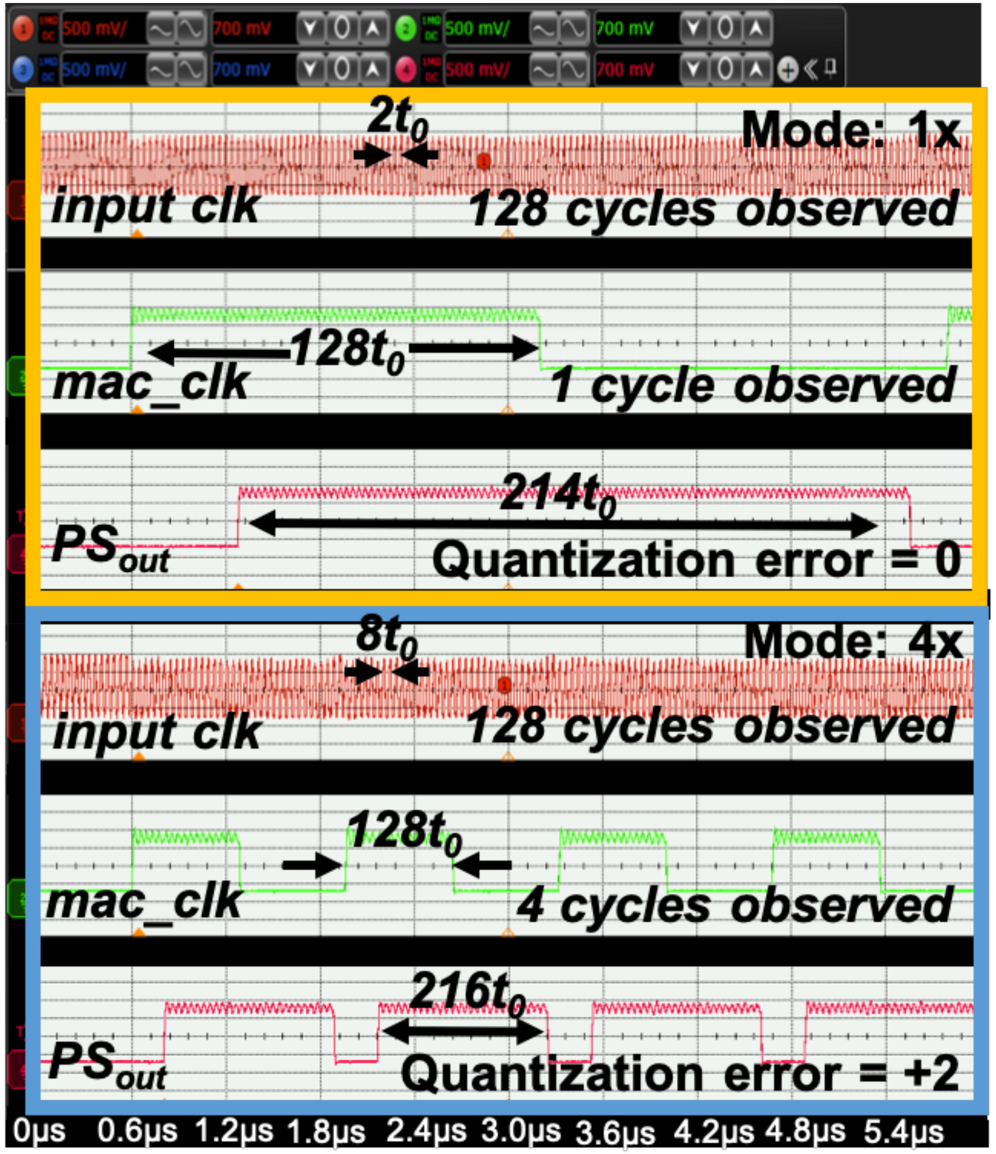
Speedup (1x-16x) Modes Demonstration

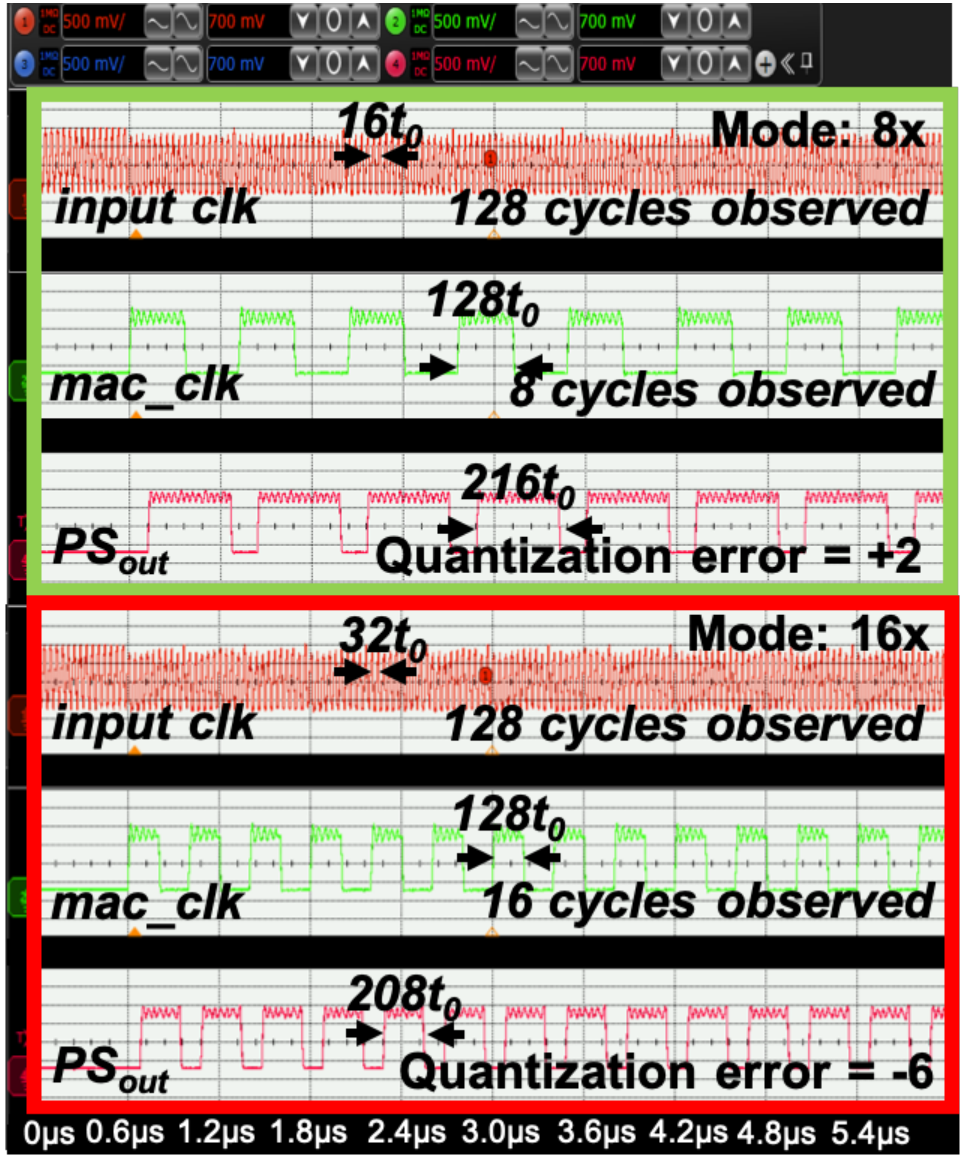
Measured waveforms for image pixel value 214



Speedup (1x-16x) Modes Demonstration

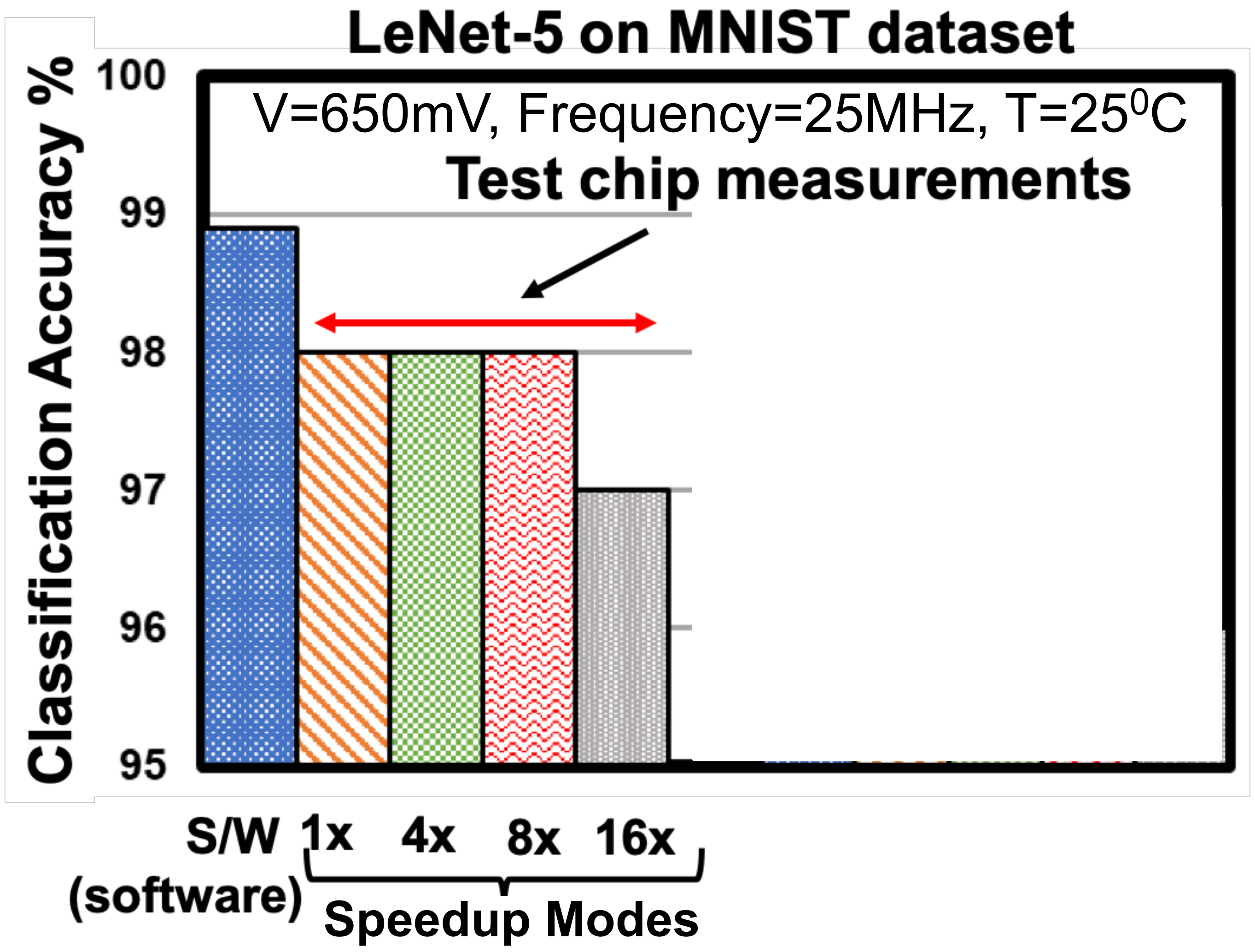
Measured waveforms for image pixel value 214





Measured Accuracy @LeNet-5

Hardware results for 100 test images



Weights: +1/-1

- ➢ **Convolution Layers:** 8-bit fixed point inputs (Hardware/Test-chip)
- ➢ **FCN and Software implementation:** 16-bit floating point inputs and weights

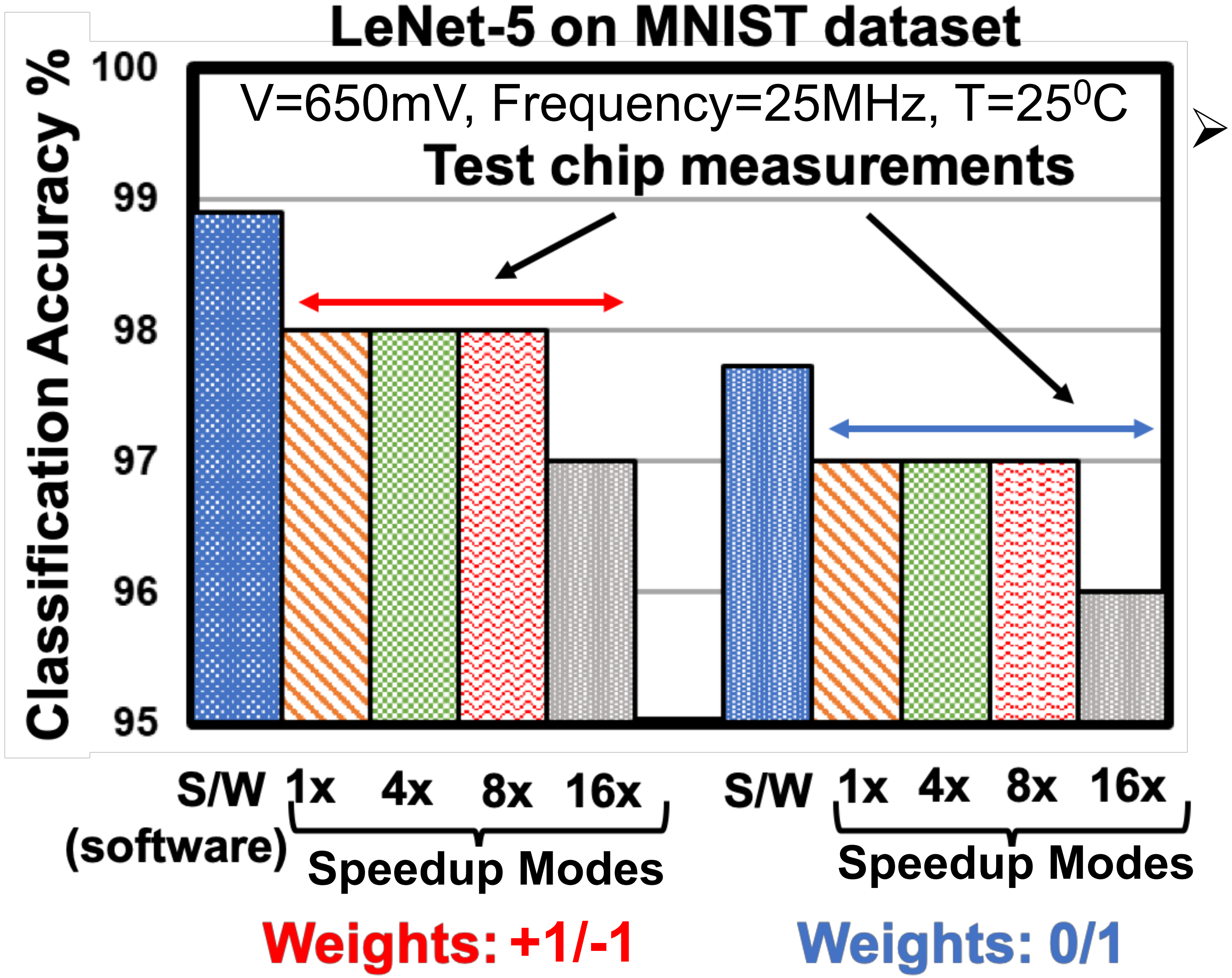
Classification Accuracy

Weights (+1/-1):

- ➢ Software: 98.81%
- ➢ Test-chip:
  - ✓ 98% for speedup modes 1x-8x
  - ✓ 97% for speedup mode 16x

Measured Accuracy @LeNet-5

Hardware results for 100 test images



- ➢ **Convolution Layers:** 8-bit fixed point inputs (Hardware/Test-chip)
- ➢ **FCN and Software implementation:** 16-bit floating point inputs and weights

Classification Accuracy

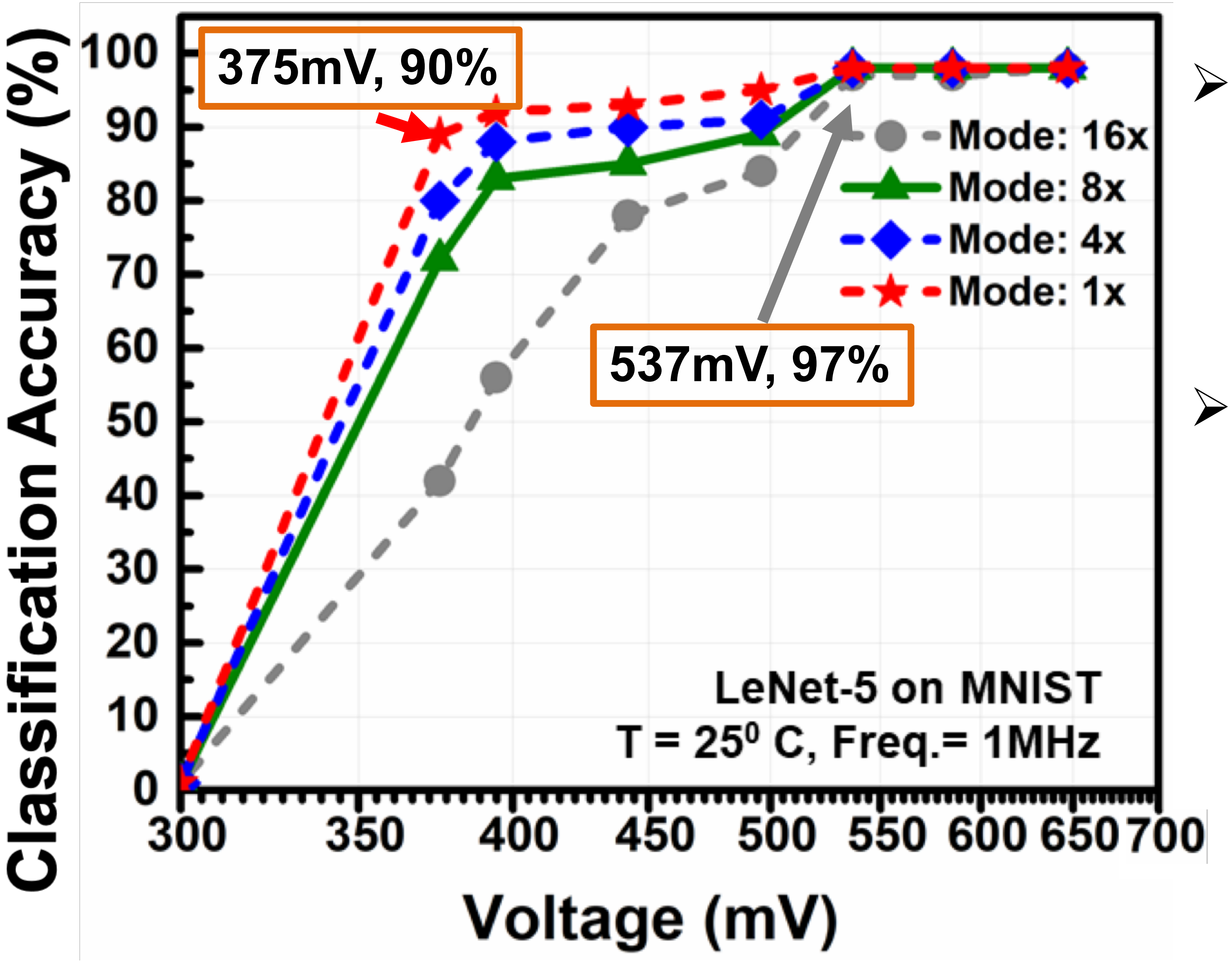
Weights (+1/-1):

- ➢ Software: 98.81%
- ➢ Test-chip:
  - ✓ 98% for speedup modes 1x-8x
  - ✓ 97% for speedup mode 16x

Weights (0/+1):

- ➢ Software: 97.62%
- ➢ Test-chip:
  - ✓ 97% for speedup modes 1x-8x
  - ✓ 96% for speedup mode 16x

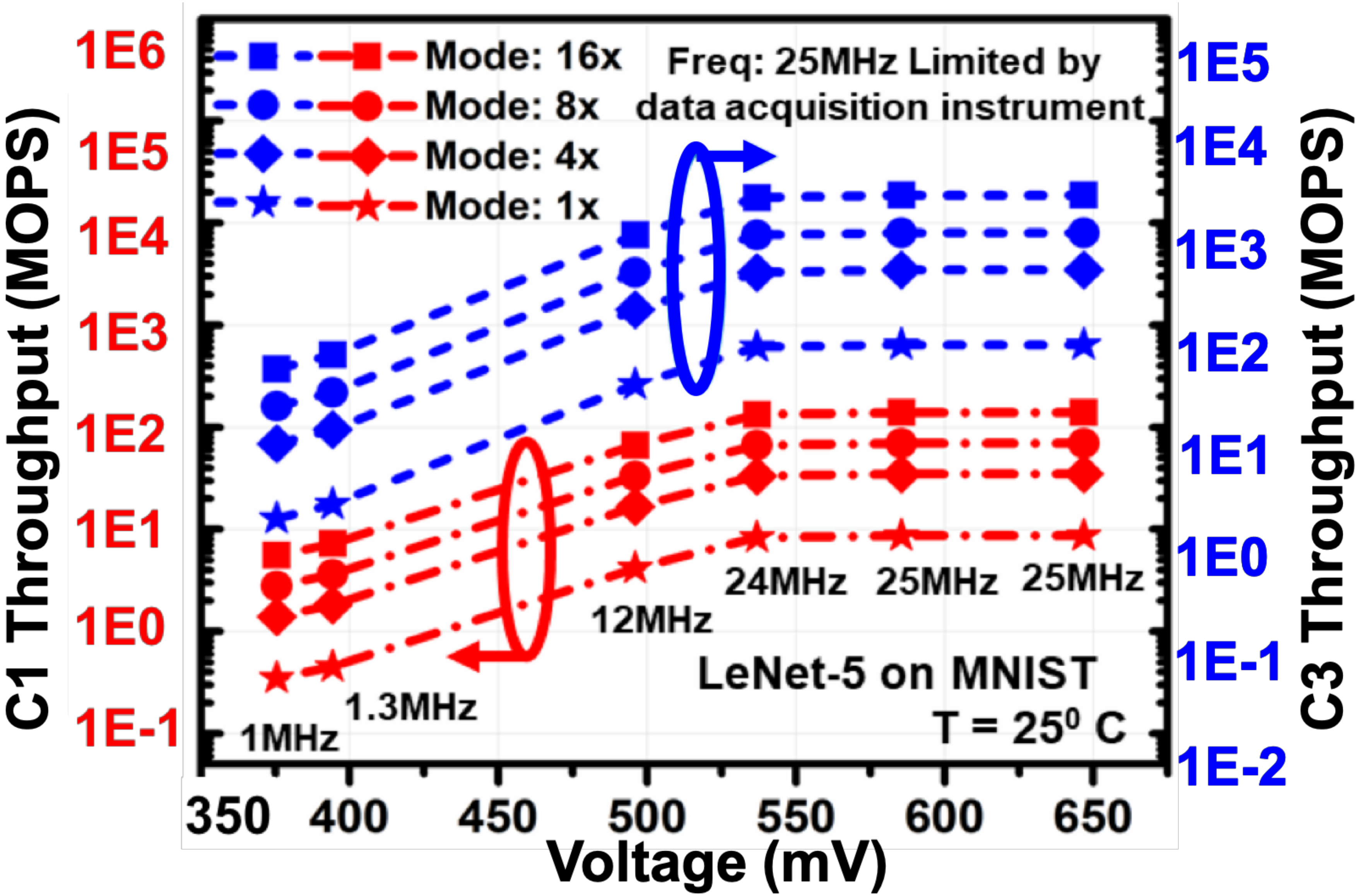
Measured Accuracy *vs.* Voltage



- ➢ Operational down to **375mV** with more than **90%** accuracy in **1x** mode

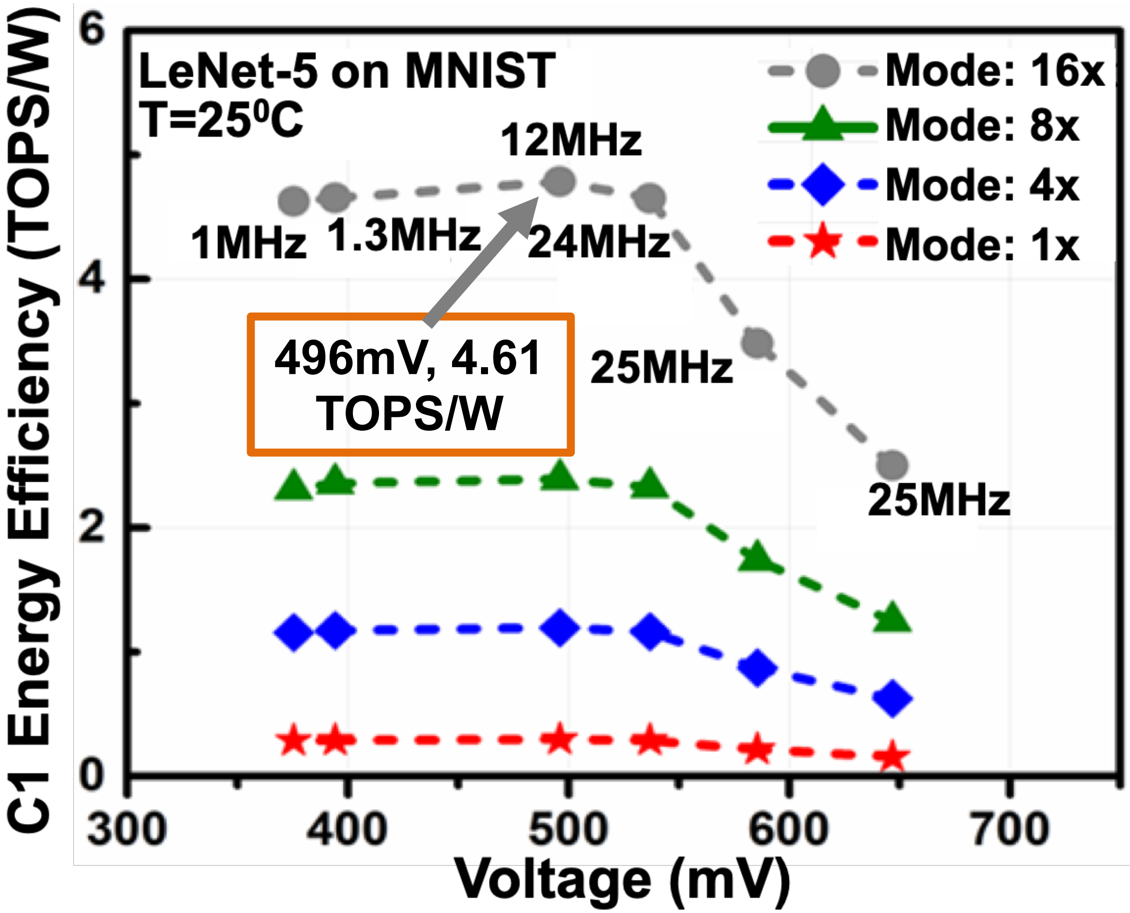
- ➢ Classification accuracy of **97%** observed at voltages down till **537mV** in **16x** mode

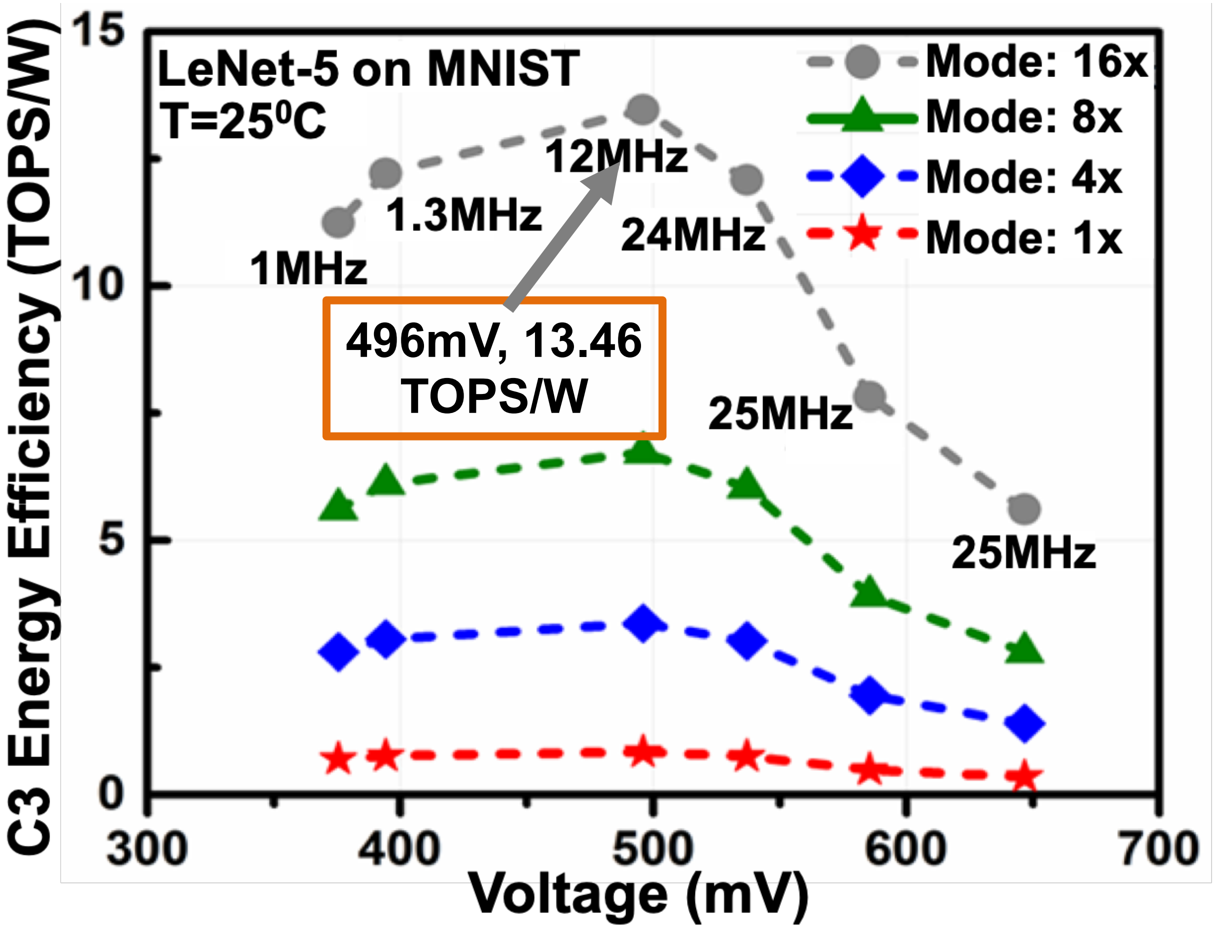
Measured Throughput *vs.* Voltage



- ➢ Throughput increases with voltage and speedup modes
- ➢ Peak Throughput for **C1 = 0.128 GOPS** @585mV
- ➢ Peak Throughput for **C3 = 0.38 GOPS** @585mV

Measured Energy Efficiency *vs.* Voltage

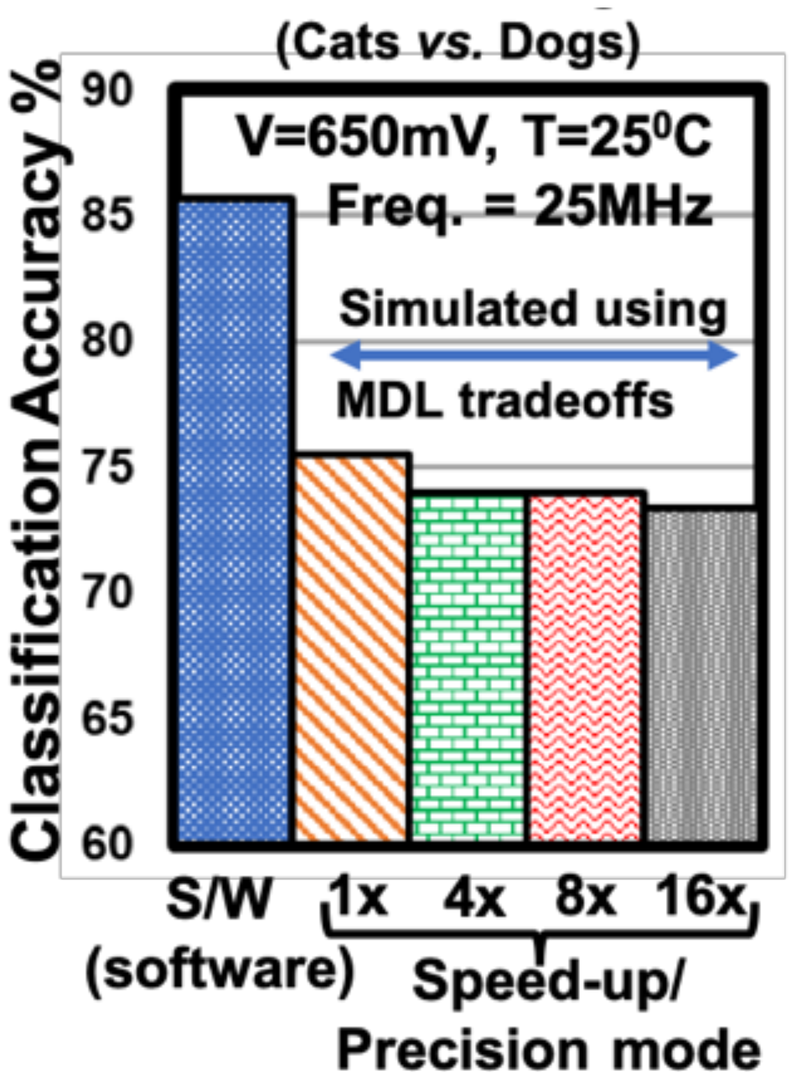




Peak Energy Efficiency for C1 = 4.61 TOPS/W @496mV

Peak Energy Efficiency for C3 = 13.46 TOPS/W @496mV

Simulation Results for AlexNet



- ➢ Dataset: Subset of ImageNet dataset (Classes - Cats and Dogs)
- ➢ Convolution and Pooling – 8 bit fixed point inputs and weights
- ➢ Fully Connected Network Layers – 16 bit floating point inputs and weights
- ➢ 13% accuracy drop observed in simulation
  - ▪ Multiple MDL lines are used for multi-bit weights dot product
  - ▪ Residue from all MDLs ➔ higher accuracy loss

Performance Summary

LeNet-5 Results/Metrics	Convolution Layer – C1				Convolution Layer – C3			
Speedup Mode	1x	4x	8x	16x	1x	4x	8x	16x
Input clock frequency (MHz)	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0
MAC clock frequency (MHz)	0.19	0.75	1.50	3.00	0.19	0.75	1.50	3.00
Convolution Cycle Time (us)	149.3	37.33	18.67	9.33	842.67	210.67	105.33	52.67
Operating Voltage (V)	0.537	0.537	0.537	0.537	0.537	0.537	0.537	0.537
Power (μW)	28.67	28.67	28.67	28.67	30.17	30.17	30.17	30.17
Throughput (GOPS)	0.008	0.032	0.064	0.128	0.023	0.091	0.183	0.365
Energy Efficiency (TOPS/W)	0.29	1.16	2.33	4.65	0.76	3.02	6.04	12.08

- Peak Energy Efficiency: **12.08(4.65) TOPS/W** for C3(C1) layers @537mV
- Peak Throughput: **0.365(0.128) GOPS** for C3(C1) layers @537mV

Outline

- Motivation
- Concept of Time-domain MAC
- Proposed CNN Engine Architecture
- Chip implementation and measurements
- Comparison with Prior Work
- Summary

Comparison with prior approaches

Reference	Tech. (nm)	Circuit Type	Input/ Weight Size	Chip Size (mm^2)	Pooling	Low Vcc Op.	Cap. or ADCs	Accuracy	Throughput (GOPS)	Power (µW)	Energy Efficiency (TOPS/W)
ISSCC’18 [1]	65	Analog	6/1bits	0.067	No	No	Yes	96.0%	10.70	380.7	28.10
ISSCC’18 [2]	65	Analog	8bits	1.44	No	No	Yes	96.0%	-	-	3.125
ISSCC’16 [3]	65	Digital	16bits	16.00	Yes	No	No	98.3%	64	4.51E+4	1.42
VLSI’16 [4]	40	Digital	6/4bits	2.40	No	Yes	No	98.0%	102	3.90E+4	2.60
CICC’17 [5]	65	Time	8/3bits	0.24	No	Yes	Yes	91.0%	0.396	2.05E+4	0.019
ISSCC’18 [6]	55	Time	6/6bits	3.125	No	Yes	No	-	2.152	690	3.12
This work (MDL CNN)	**40**	**Time**	**8bits/ 1bit***	**0.124**	**Yes**	**Yes**	**No**	**97.0%**	**0.365**	**30.17**	**12.08**

**Scalable to multi-bit weights*

References mentioned in paper

Conclusions

- Proposed **Bidirectional Memory Delay Line** for energy efficient time-domain MAC computation

- All-digital compact and **technology scaling friendly** design (no ADCs, DACs, frequency modulators)

- Low power design supporting **near-threshold voltage** operation and 16x performance boost with 4 input encoding modes

- Configurable MDL lengths with **on-chip pooling** and averaging operations

- Demonstrated the proposed time-domain CNN engine in **40nm CMOS** node achieving **12.08 TOPS/W** energy efficiency and **0.365 GOPS** throughput at 537mV

Acknowledgements

- Authors would like to thank TSMC University shuttle program for the test- chip fabrication support.
- Authors would like to thank AMD for the financial support
- Authors also thank Vignesh Radhakrishnan and Jacob Rohan (graduate students in ECE, UT Austin) for helping with the test-chip measurements.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
ISSCC®
UNIVERSITY OF PENNSYLVANIA