SAND2020-9224PE

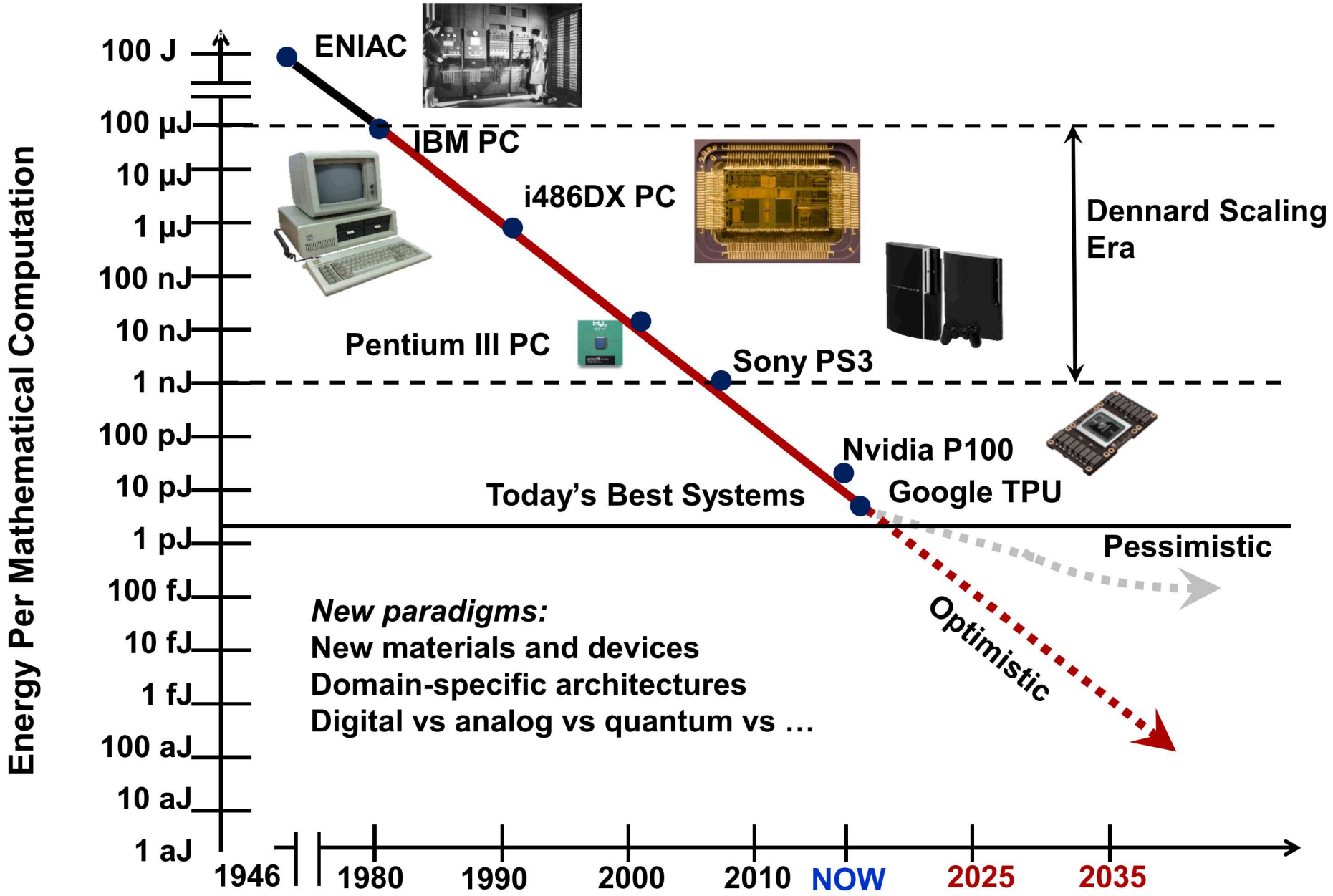
Energy-efficient stateful logic with magnetic domain walls

T. Patrick Xiao[1], Christopher H. Bennett[1], Xuan Hu[2], Ben Feinberg[1], Robin Jacobs-Gedrim[1], Sapan Agarwal[1], John Brunhaver[3], Joseph S. Friedman[2], Jean Anne C. Incorvia[4], and Matthew J. Marinella[1]

[1]Sandia National Laboratories | [2]University of Texas at Dallas | [3]Arizona State University | [4]University of Texas at Austin

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Evolution of computing machinery



Inefficiencies in current systems

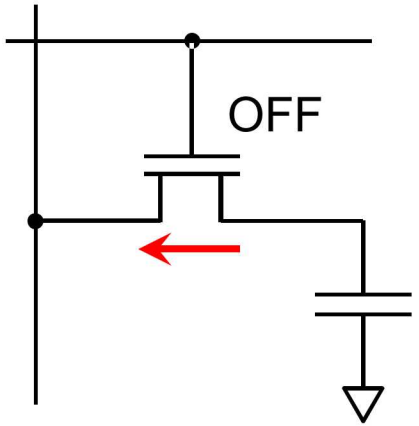
Communication cost

Memory access is more expensive than computation (per bit), by multiple orders of magnitude!

↓

(1) *Redesign the architecture to perform computation close to or inside memory*

Static leakage power



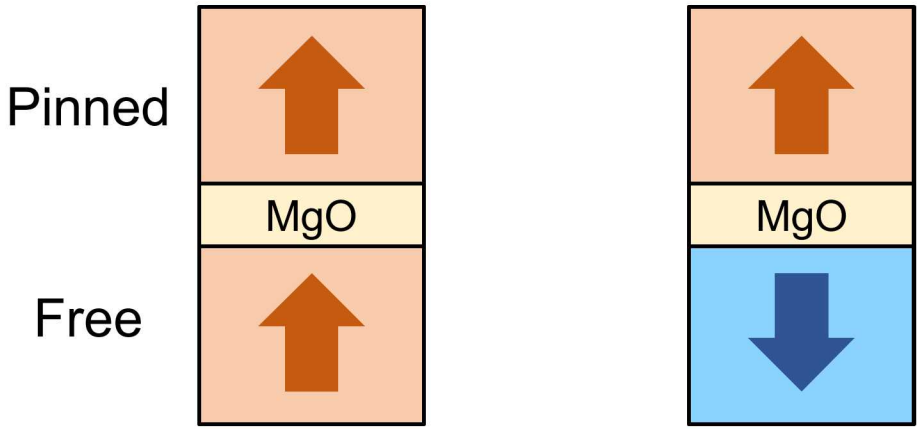
Leakage in idle devices accounts for at least 20% of system power,[1] and prevents scaling CMOS to low voltages

↓

(2) *Perform computation using non-volatile memory devices*

[1]F. Tachibana, et al. IEEE Journal of Solid-State Circuits 49, 118-26 (2014).

Magnetic memory (MRAM): based on the magnetic tunnel junction (MTJ)



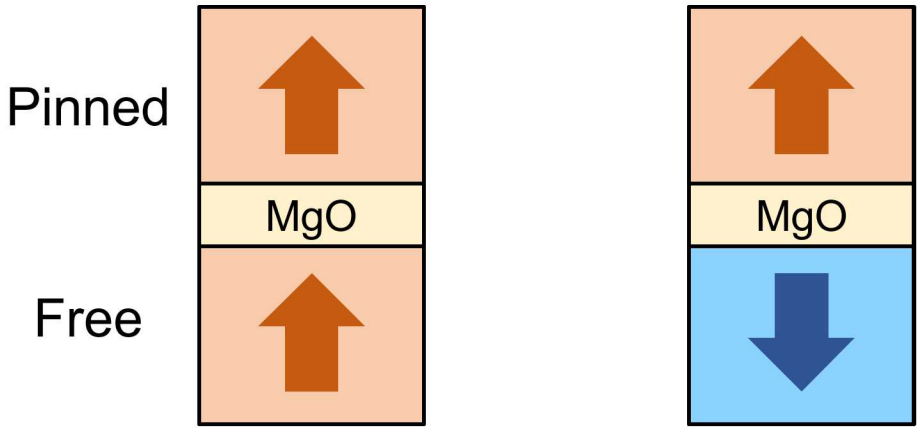
Parallel state (P): low resistance

Anti-parallel state (AP): high resistance

Written by:

- Ampere's law
- Spin transfer torque (STT)
- Spin orbit torque (SOT)

Magnetic memory (MRAM): based on the magnetic tunnel junction (MTJ)



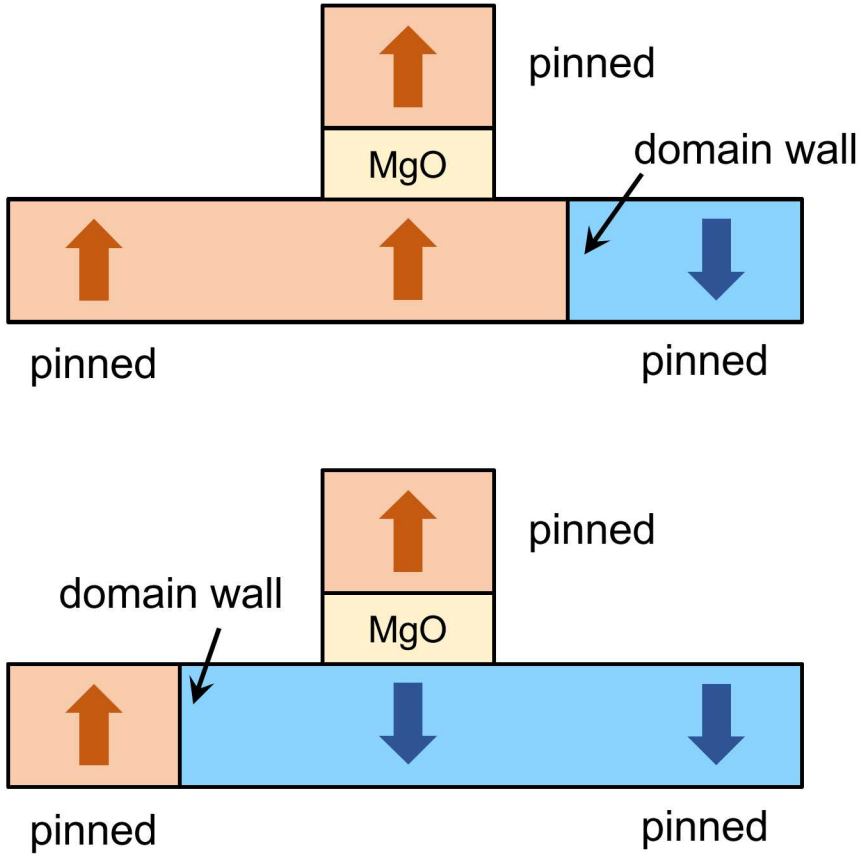
Parallel state (P): low resistance

Anti-parallel state (AP): high resistance

Written by:

- Ampere's law
- Spin transfer torque (STT)
- Spin orbit torque (SOT)

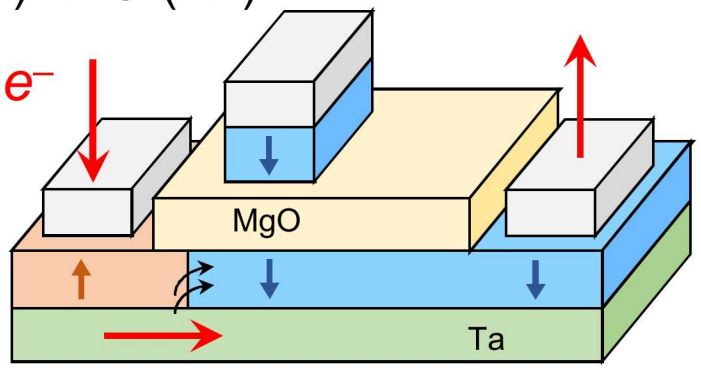
Domain wall MTJ device



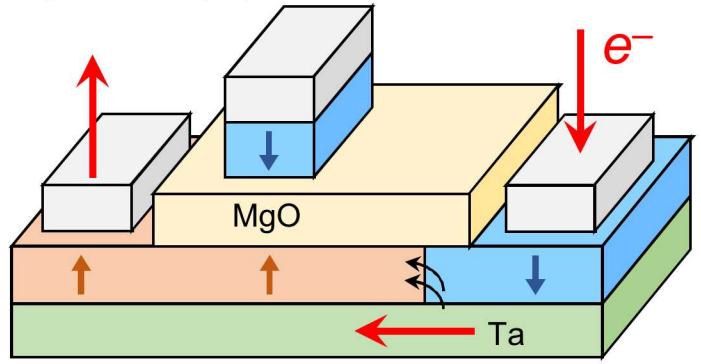
Domain wall memory: read/write

Write:

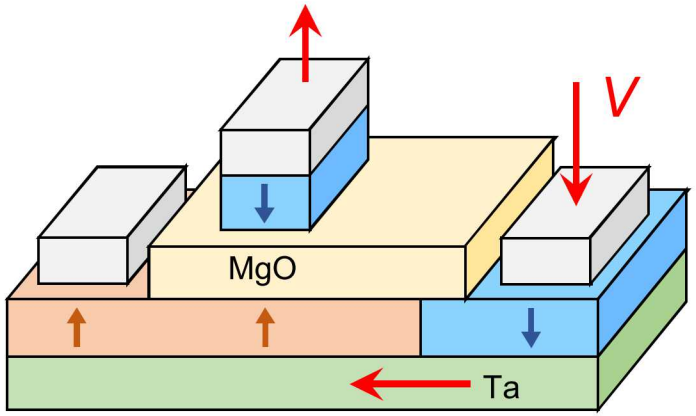
‘1’ (P) → ‘0’ (AP)



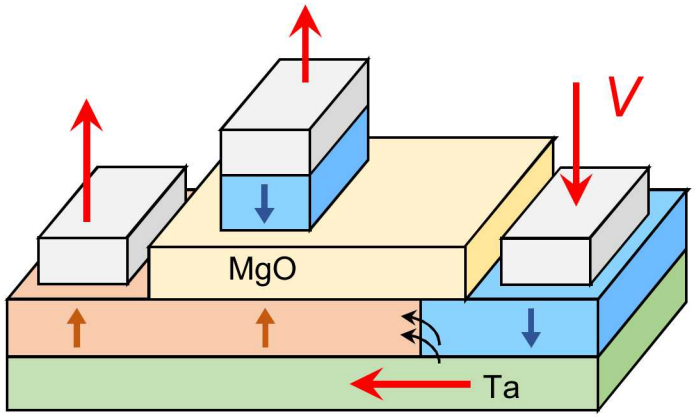
‘0’ (AP) → ‘1’ (P)



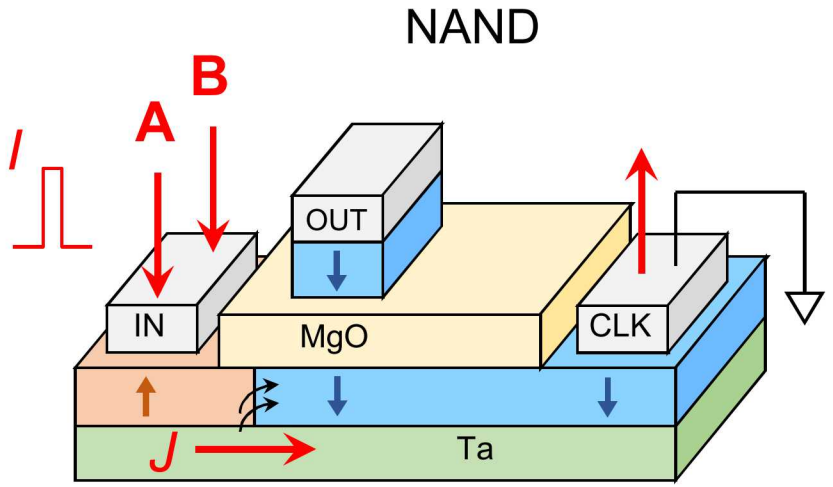
Read:



Read & reset:

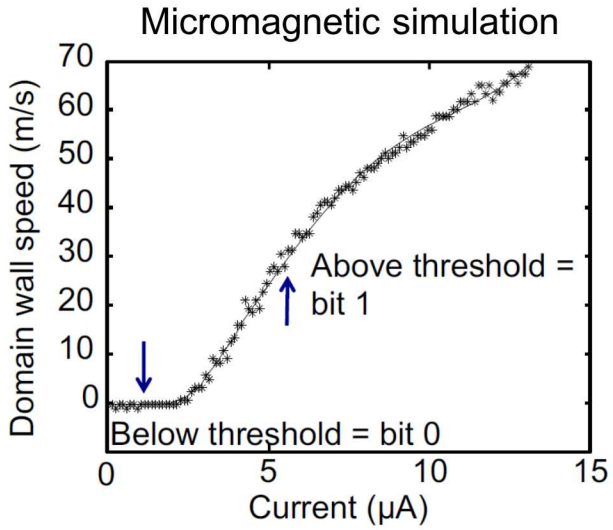


A single domain wall device implements any elementary logic gate



Current in : Current out → *Cascadable*

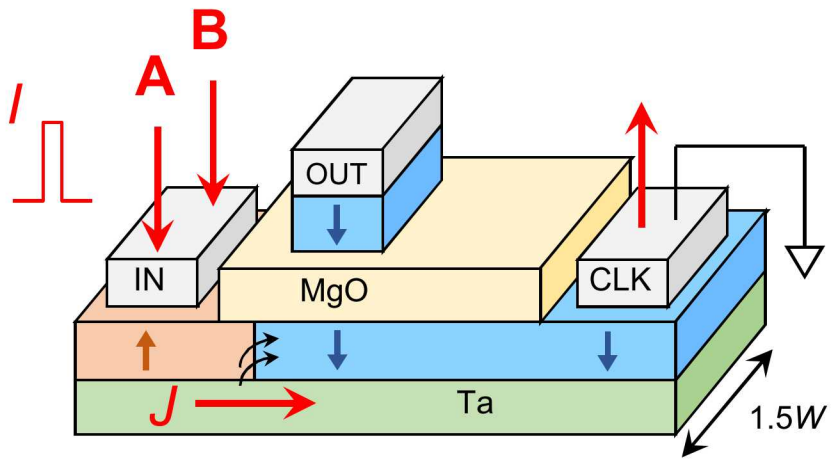
A	B	DW state
0	0	P (1)
1	0	P (1)
0	1	P (1)
1	1	AP (0)



J. A. Currivan et al, *IEEE Magnetics Letters* 2012

A single domain wall device implements any elementary logic gate

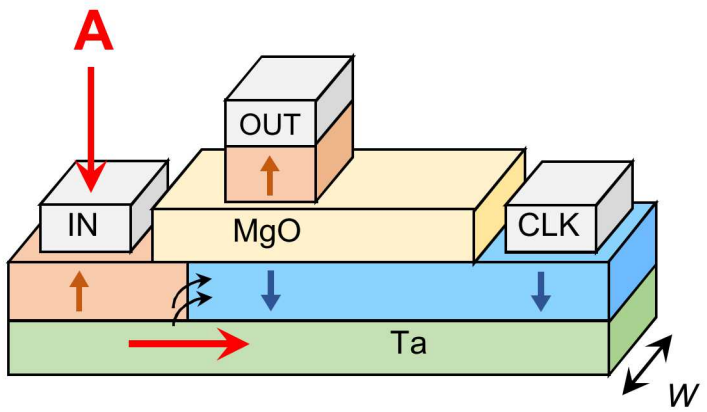
NAND



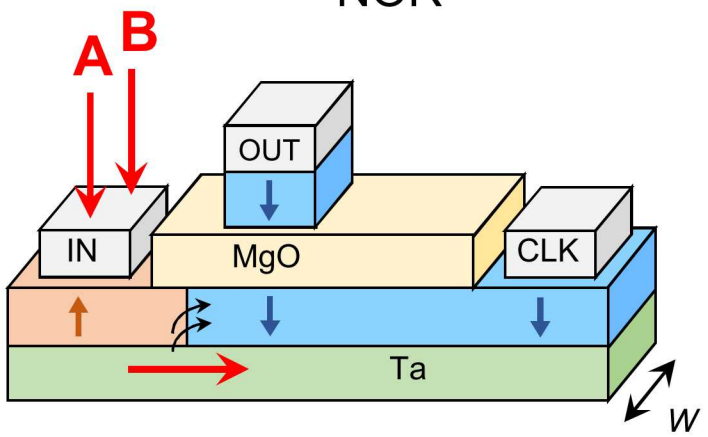
Current in : Current out → *Cascadable*

A	B	DW state
0	0	P (1)
1	0	P (1)
0	1	P (1)
1	1	AP (0)

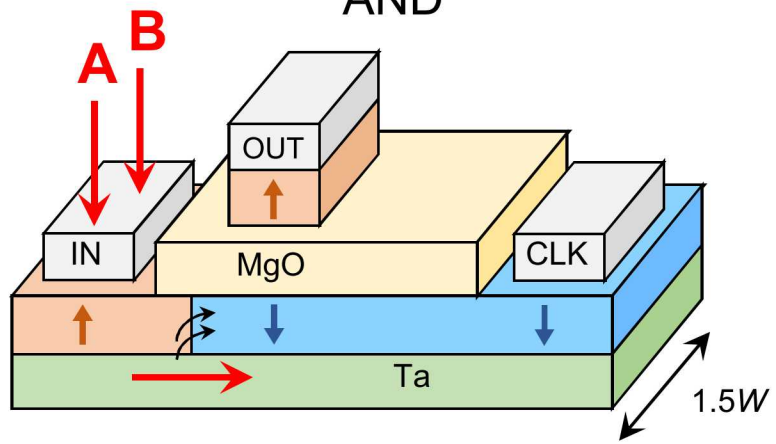
Buffer



NOR



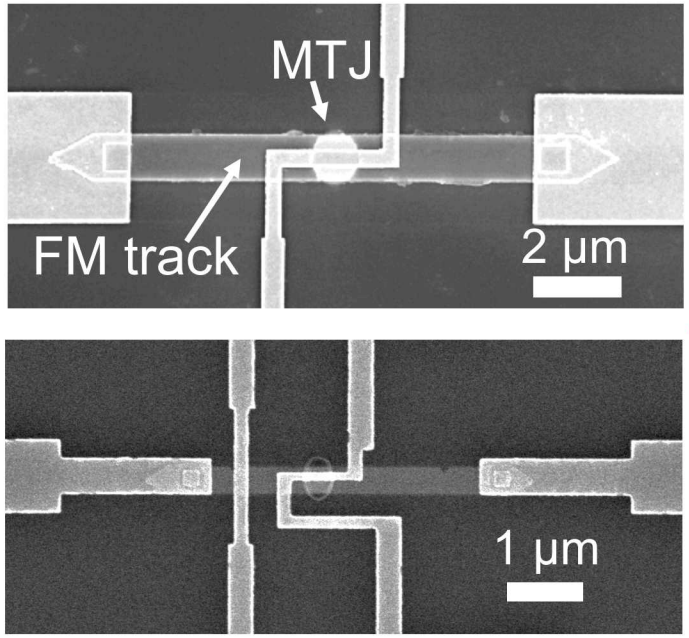
AND



J. A. Currivan et al, *IEEE Magnetics Letters* 2012

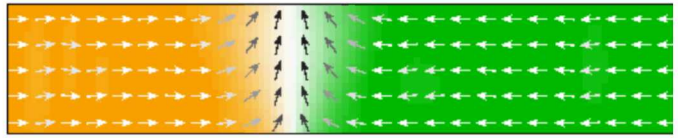
Experimental demonstration and modeling

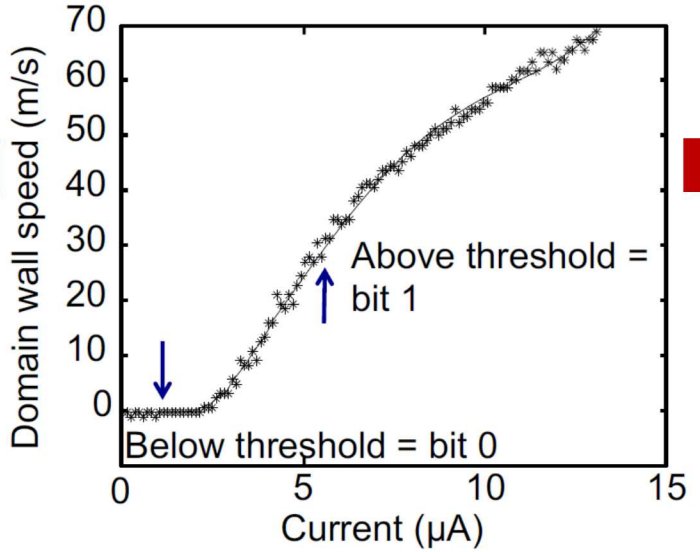
Electrical control of perpendicular anisotropy domain wall MTJs



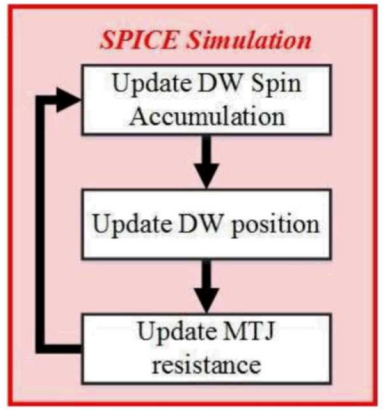
Courtesy of Jean Anne Incorvia

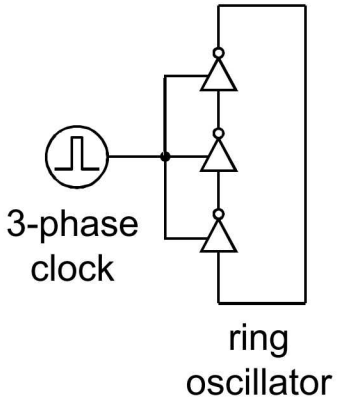
Micromagnetics simulation

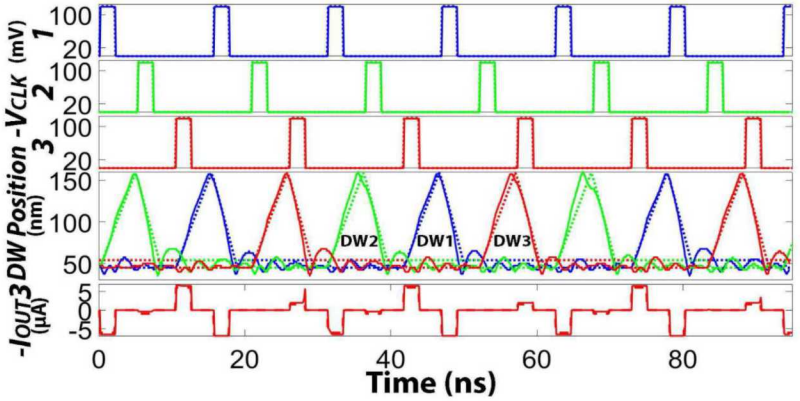




SPICE model



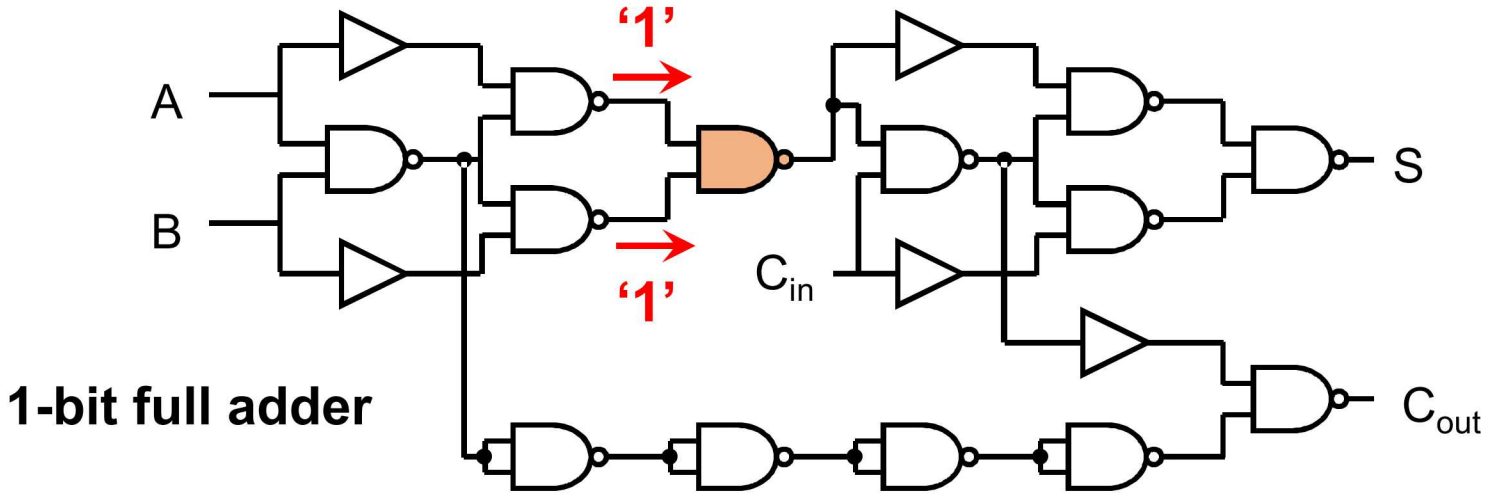




J. A. Currivan-Incoriva et al, *IEEE Magnetics Letters* 2012
J. A. Currivan-Incorvia, et al. *Nature Comm* 2016

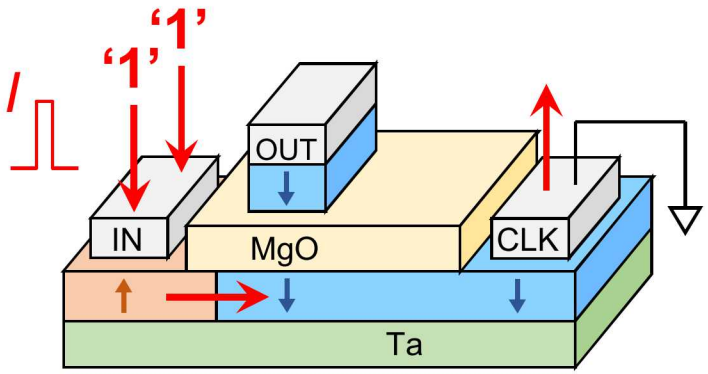
X. Hu et al, *IEEE Trans Elec Dev* 2019

Cascaded logic using clocked domain wall devices



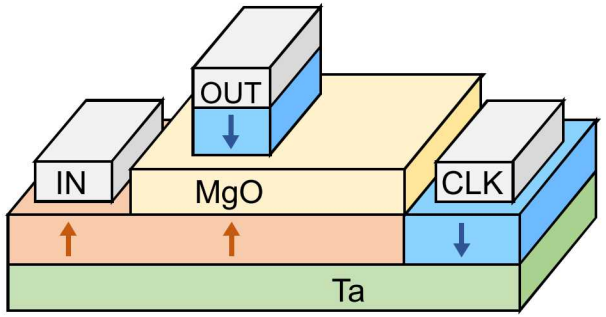
1-bit full adder

Clock phase 1:
receive input & compute
State: '1' (P) → '0' (AP)



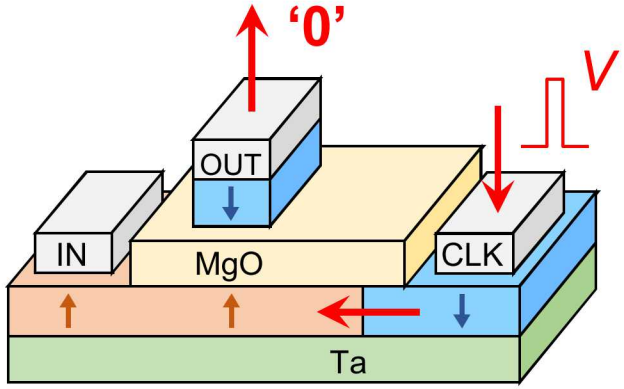
DW NAND gate

Clock phase 2:
standby
State: '0' (AP)



DW NAND gate

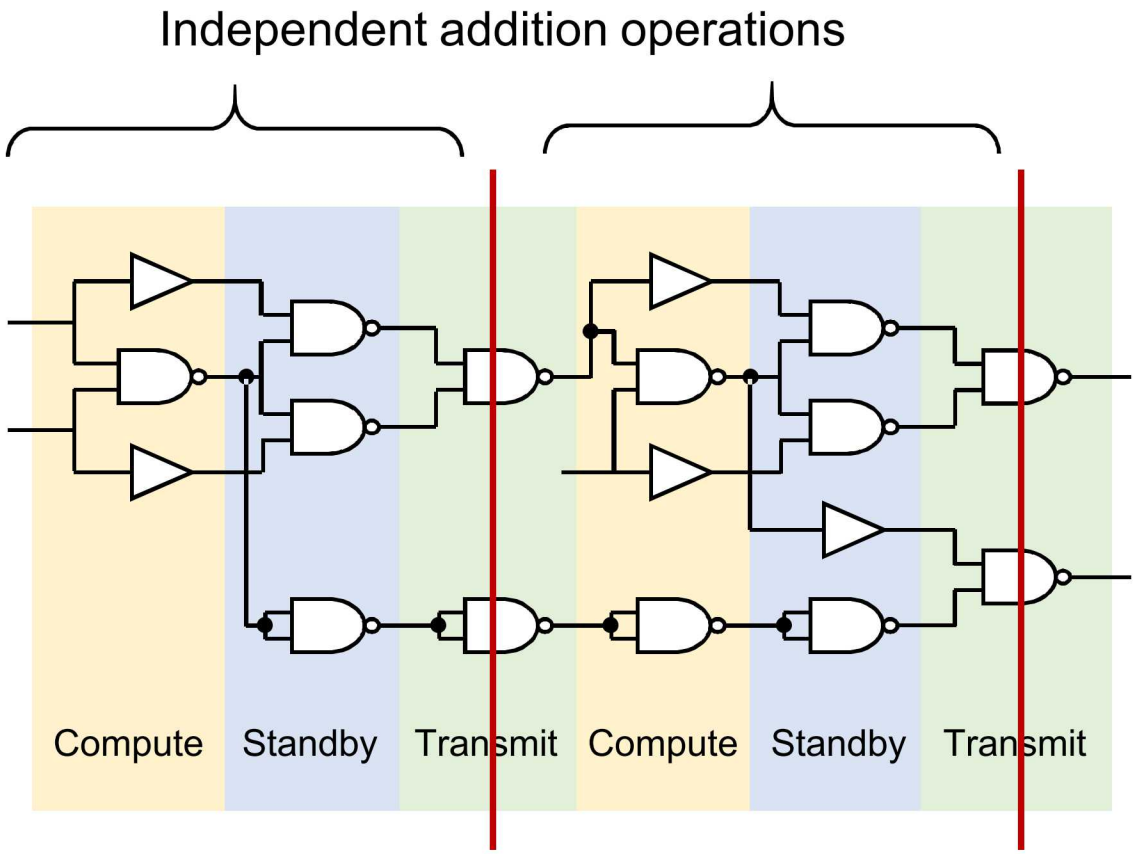
Clock phase 3:
transmit & reset
State: '0' (AP) → '1' (P)



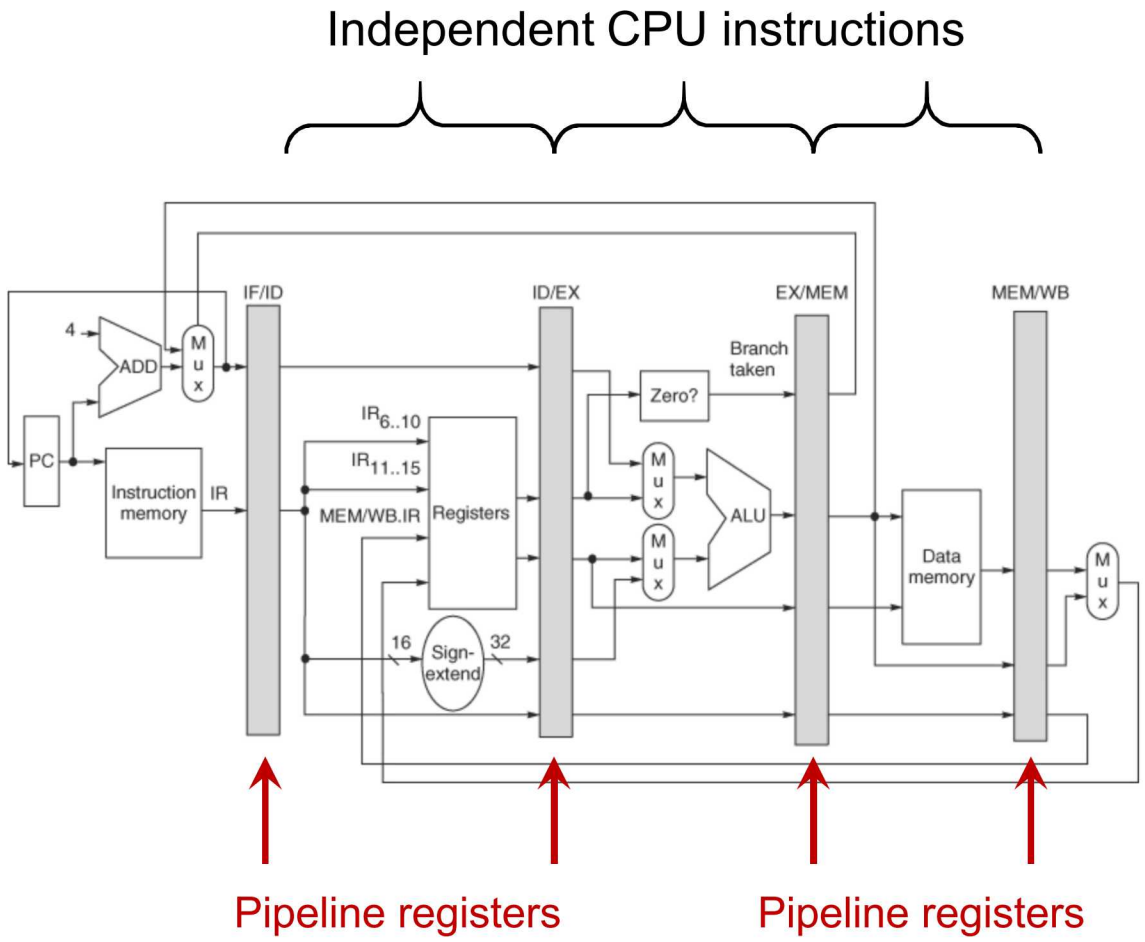
DW NAND gate

Domain wall devices serve double duty as processing elements and non-volatile data buffers

Micro-pipelined domain wall logic

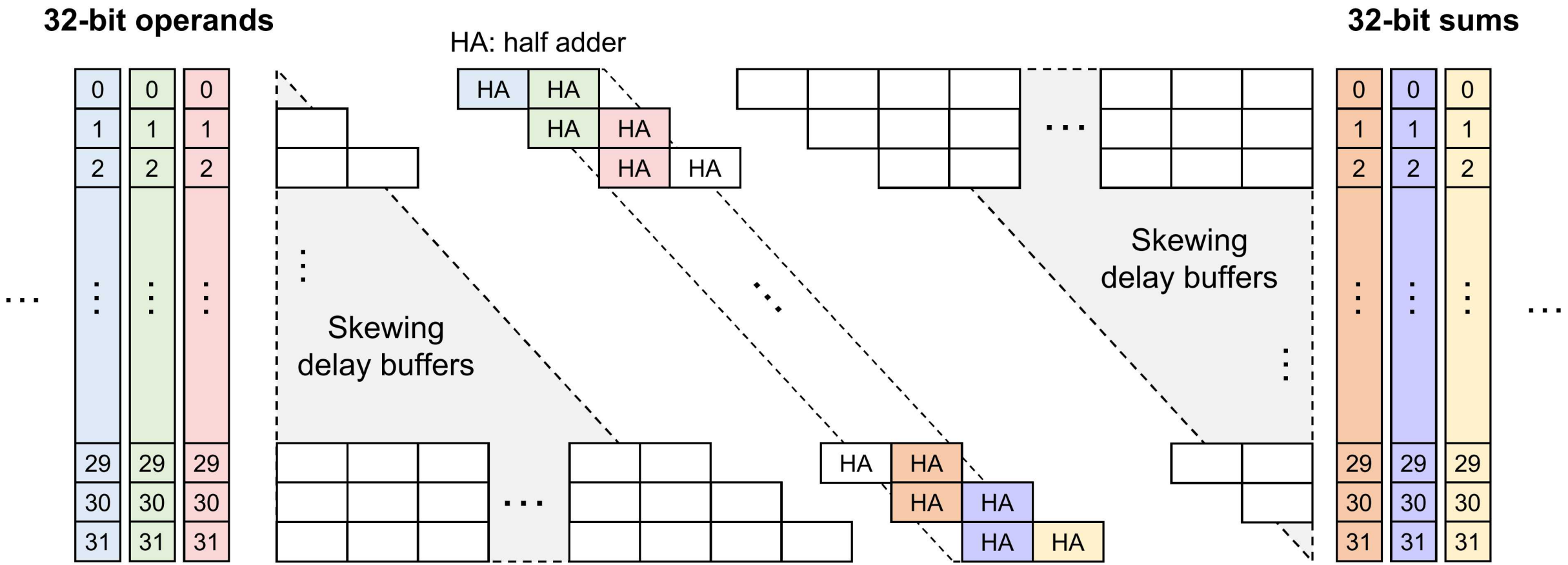


Pipelined CMOS processor



MIPS architecture, from Hennessy & Patterson, *Computer Architecture*, 2011

Bit-level micro-pipelining enables high-throughput arithmetic operations

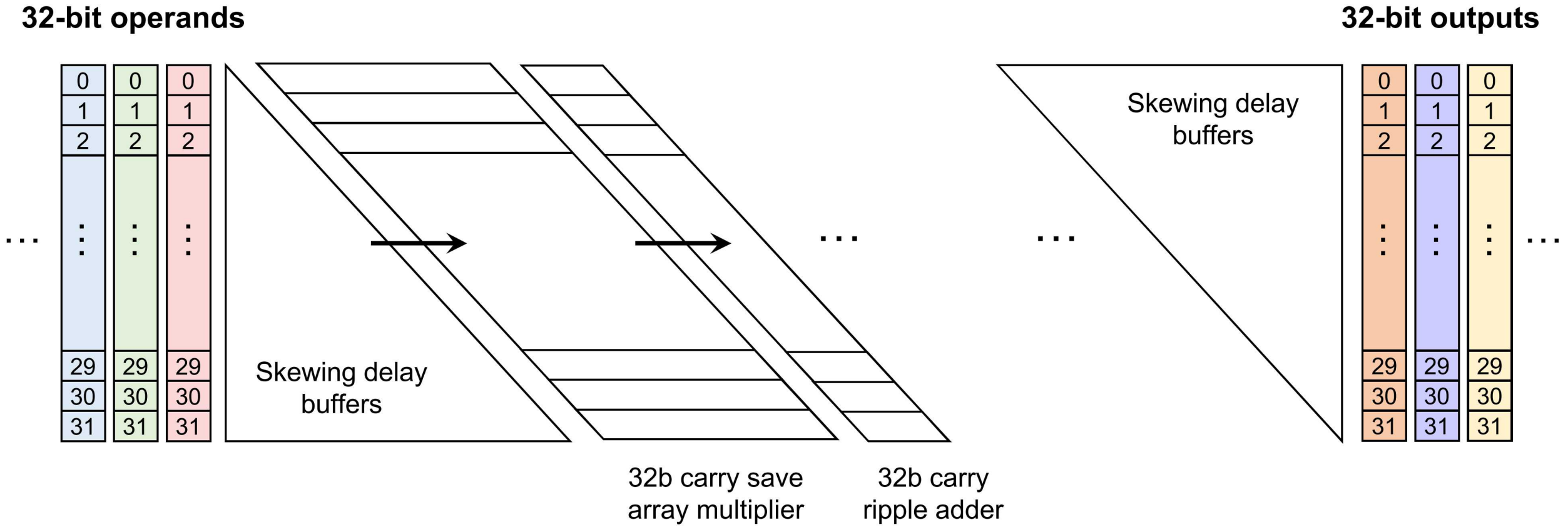


32-bit ripple carry adder

Delay per add: 32 cycles

Throughput: 1 sum / cycle = 1 sum / 3 gate delays

Bit-level micro-pipelining enables high-throughput arithmetic operations

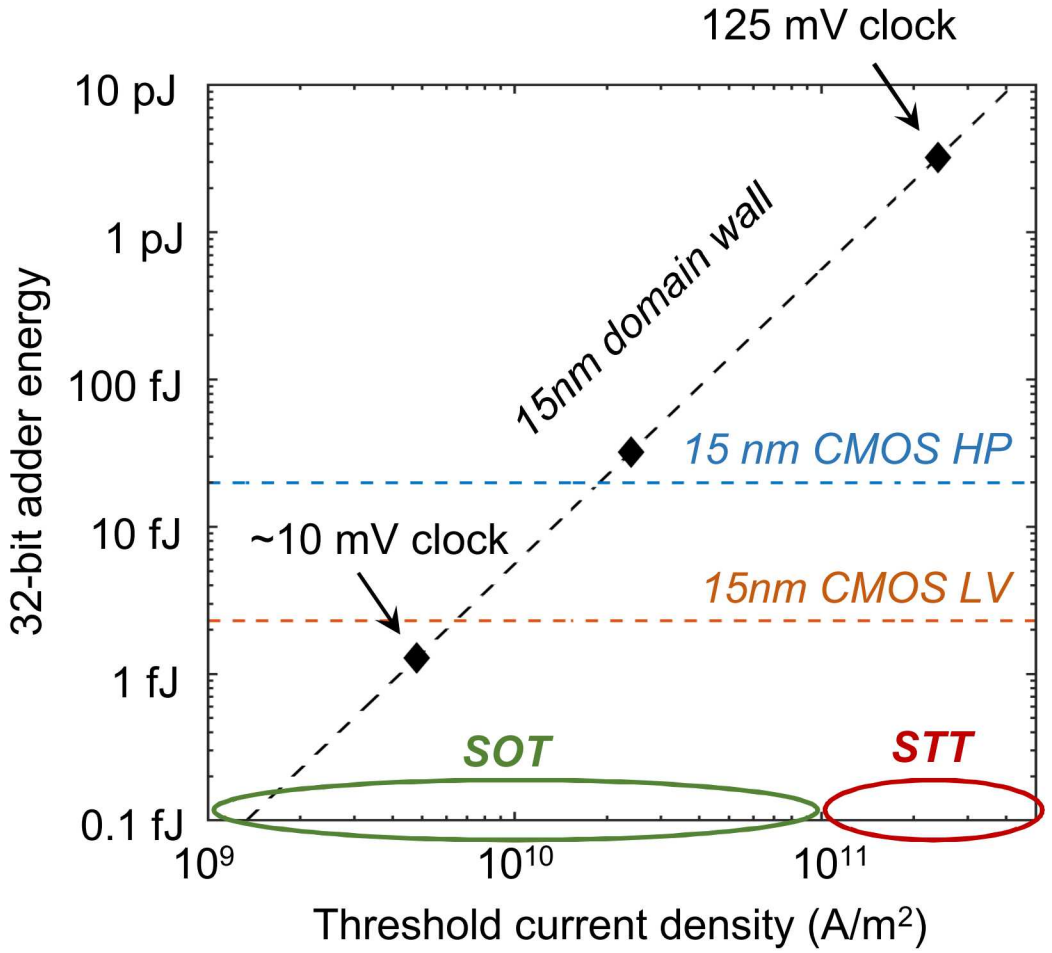


Chain of arithmetic operations (e.g. matrix multiplication)

Delay per operation: *X* cycles

Throughput: 1 output / 3 gate delays

Domain wall logic dynamic energy consumption can surpass scaled CMOS



32-bit adder performance comparison

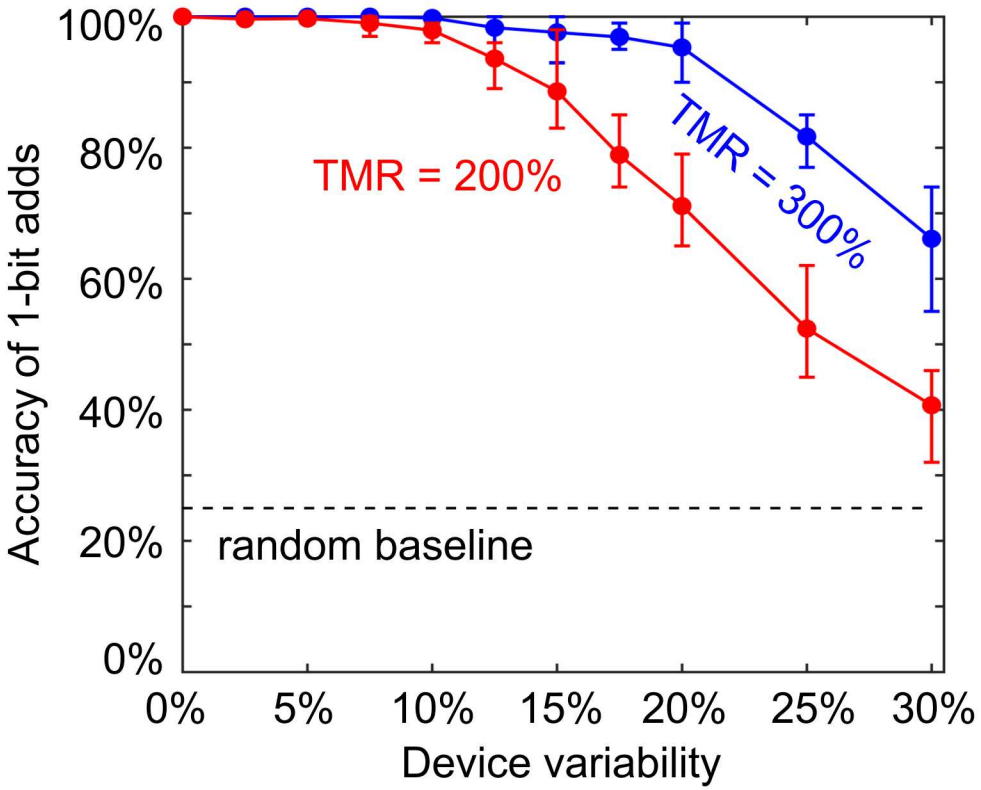
	CMOS HP	CMOS LV	DW-MTJ v_{DW} ~ 15 m/s
Latency	0.43 ns	4.2 ns	495 ns
Throughput [10^9 Op/s]	2.4	0.24	0.07

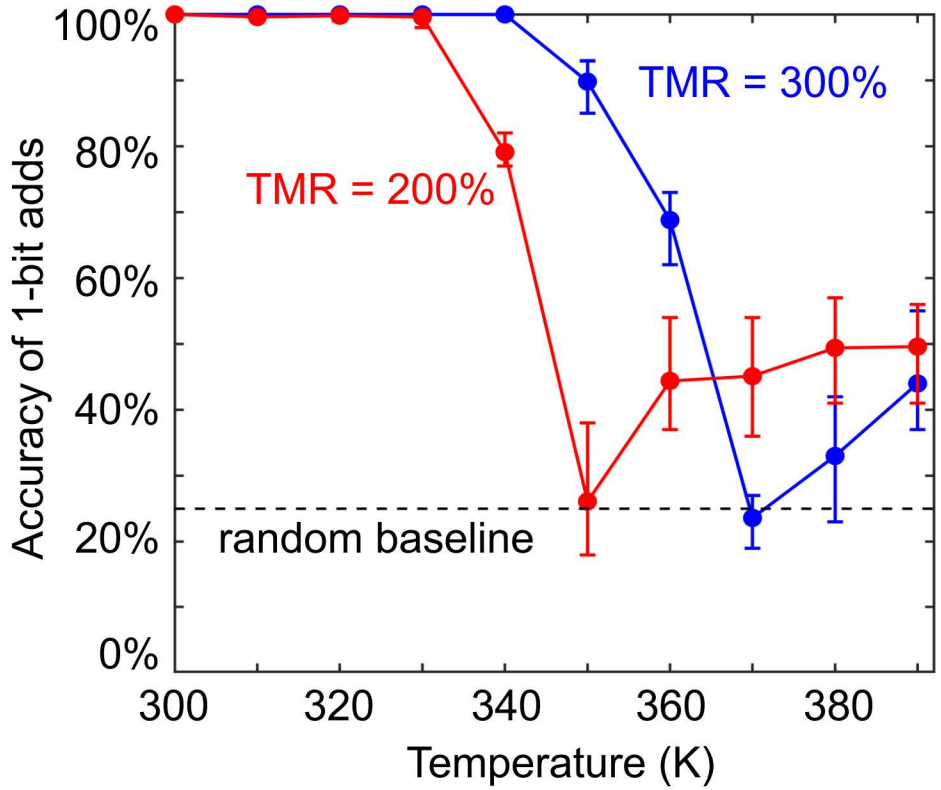
The more complex and regular the computation, the greater the benefit in throughput from micro-pipelining

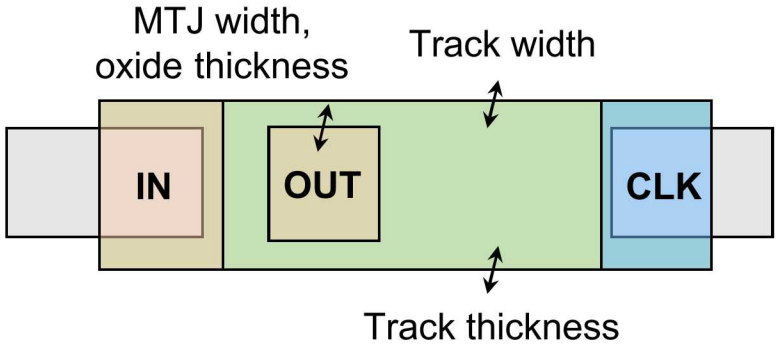
CMOS: D. Nikonov, I. Young, IEEE JXCDC 2015

T. Xiao et al, IEEE JXCDC 2019

Robust to process variations and high temperature



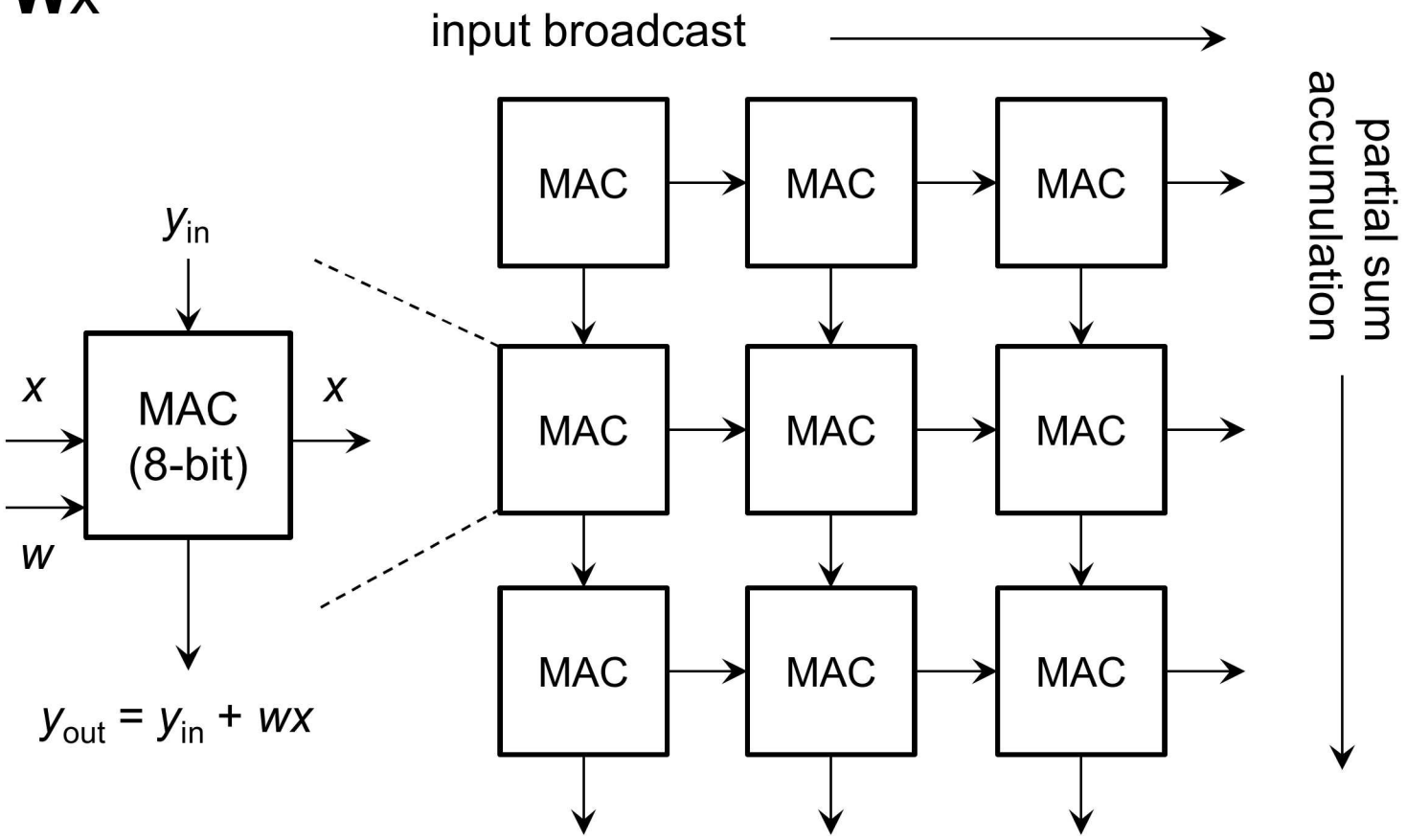




Higher TMR → Higher current On/Off ratio
→ Higher robustness to noise/variations

Systolic matrix multiplication

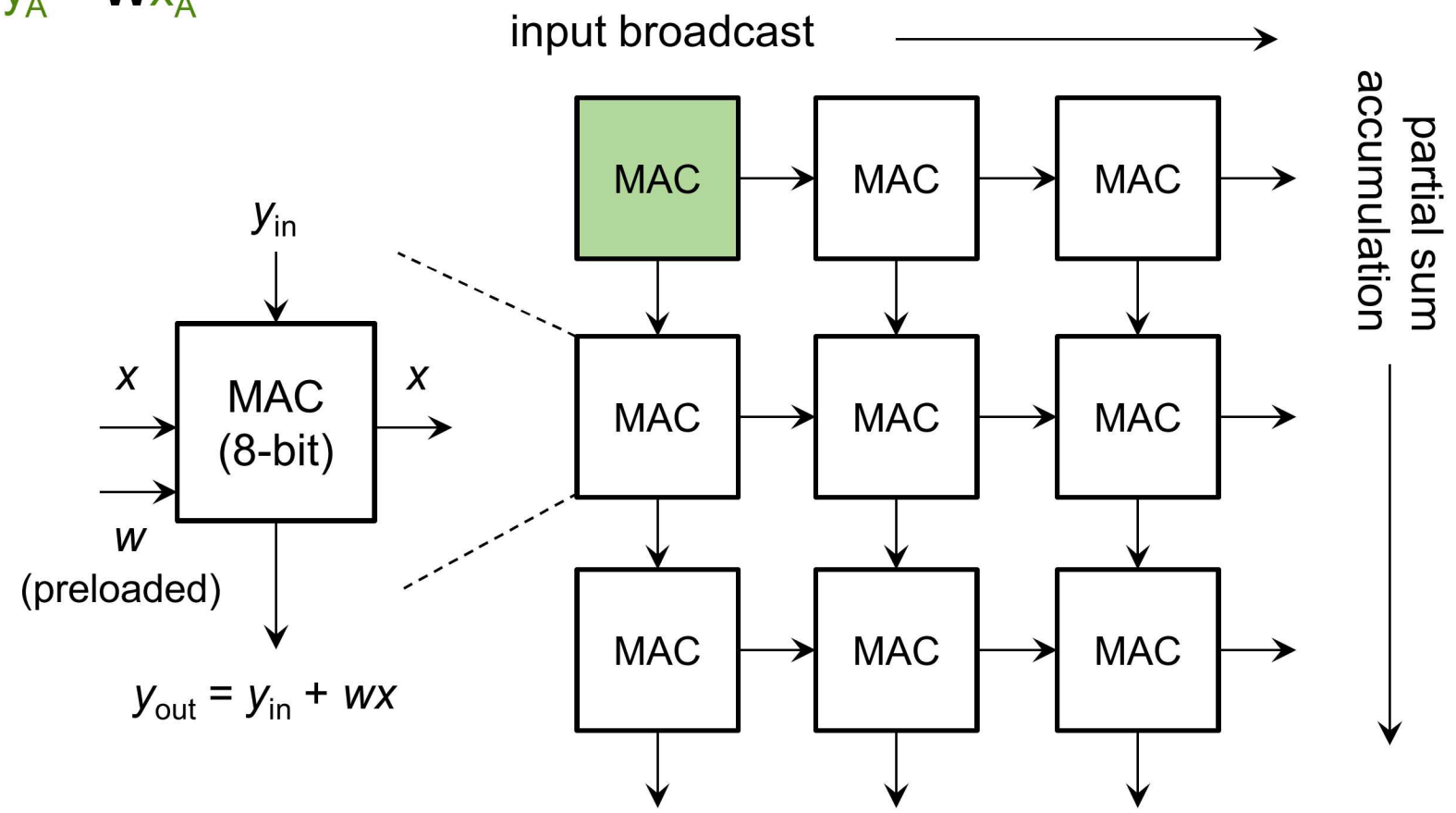
$y = \mathbf{W}x$



N. Jouppi et al, arXiv [cs.AR], 2017

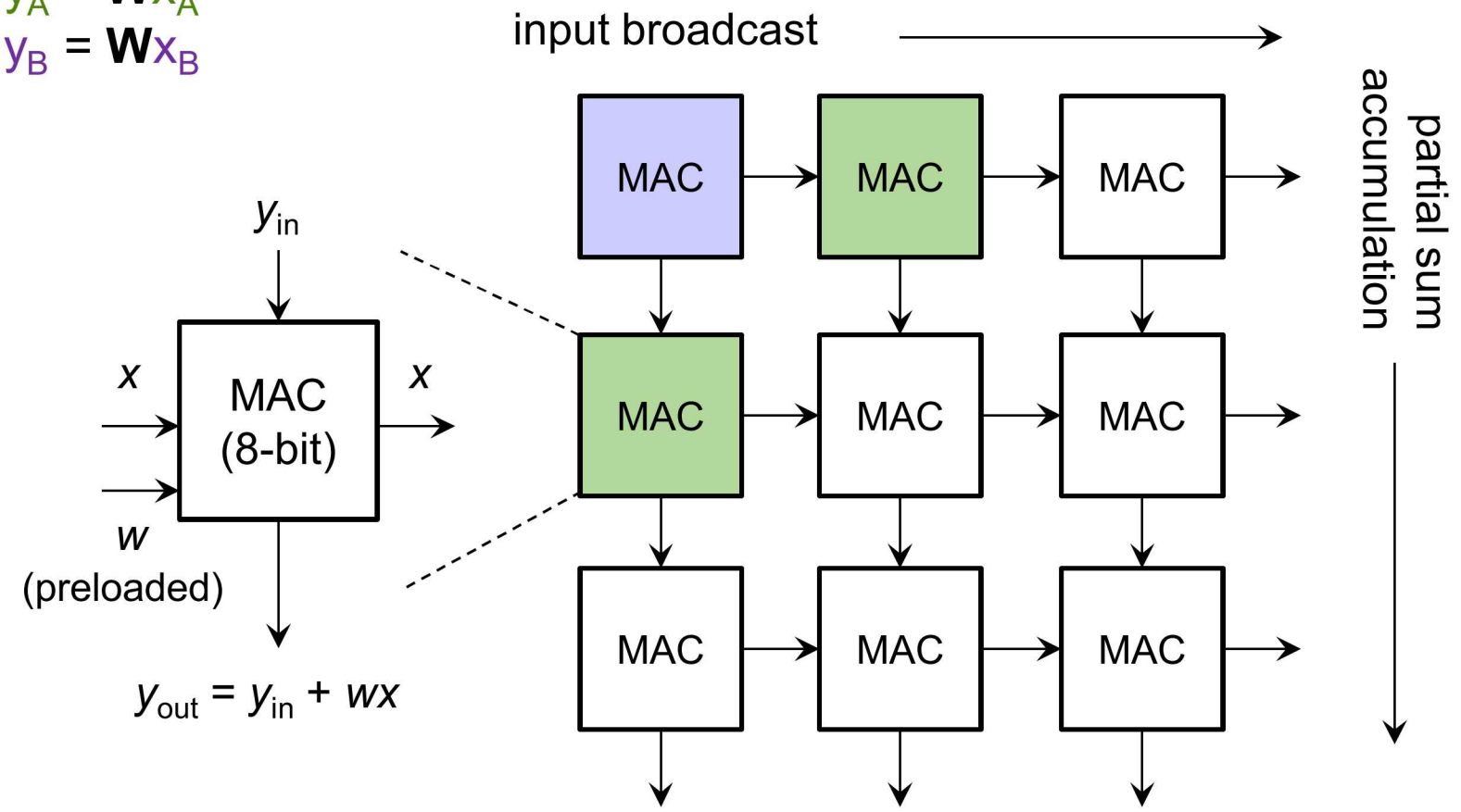
Systolic matrix multiplication

$y_A = \mathbf{W}x_A$



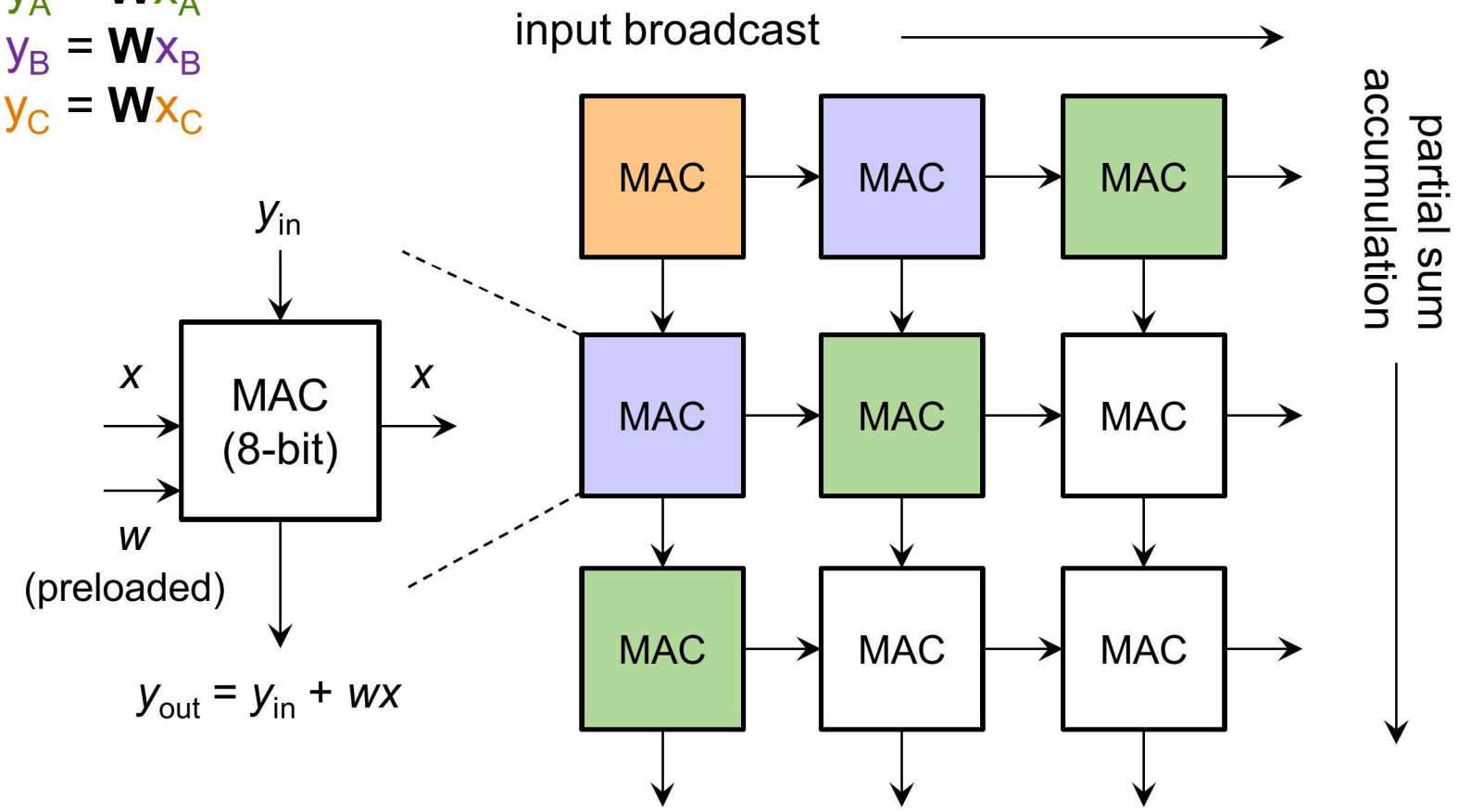
Systolic matrix multiplication

$y_A = \mathbf{W}x_A$
$y_B = \mathbf{W}x_B$



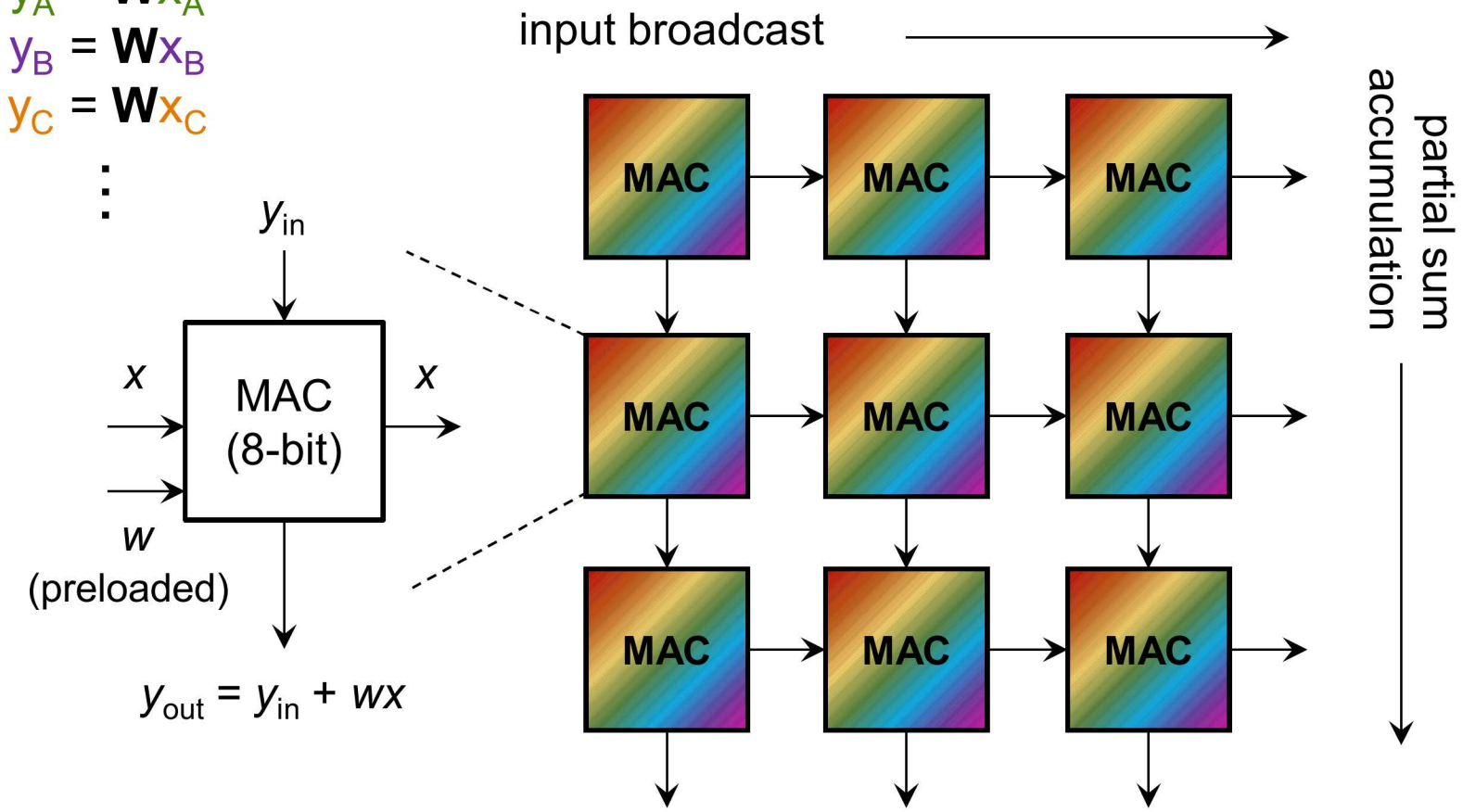
Systolic matrix multiplication

$y_A = \mathbf{W}x_A$
$y_B = \mathbf{W}x_B$
$y_C = \mathbf{W}x_C$

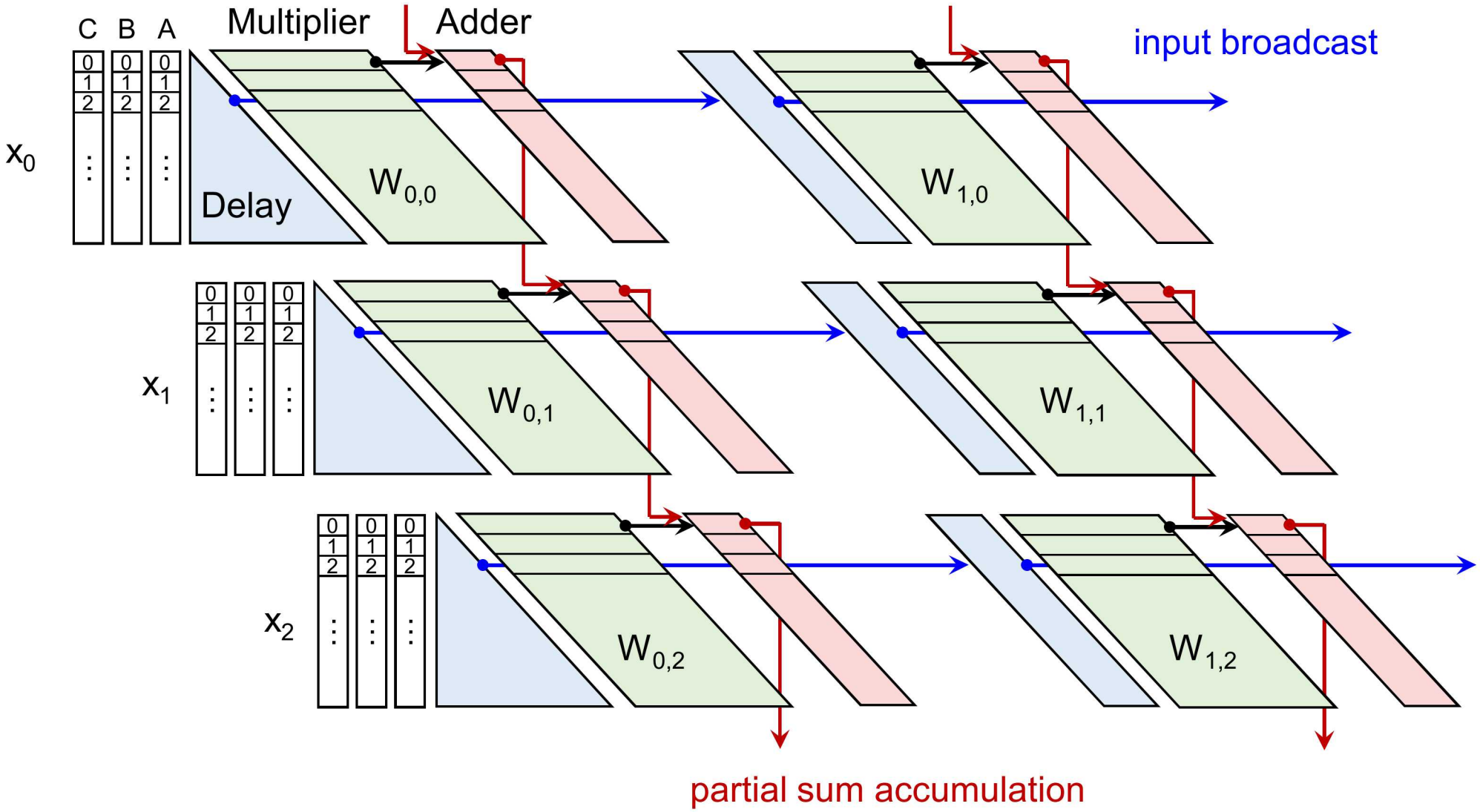


Micro-pipelined systolic matrix multiplication

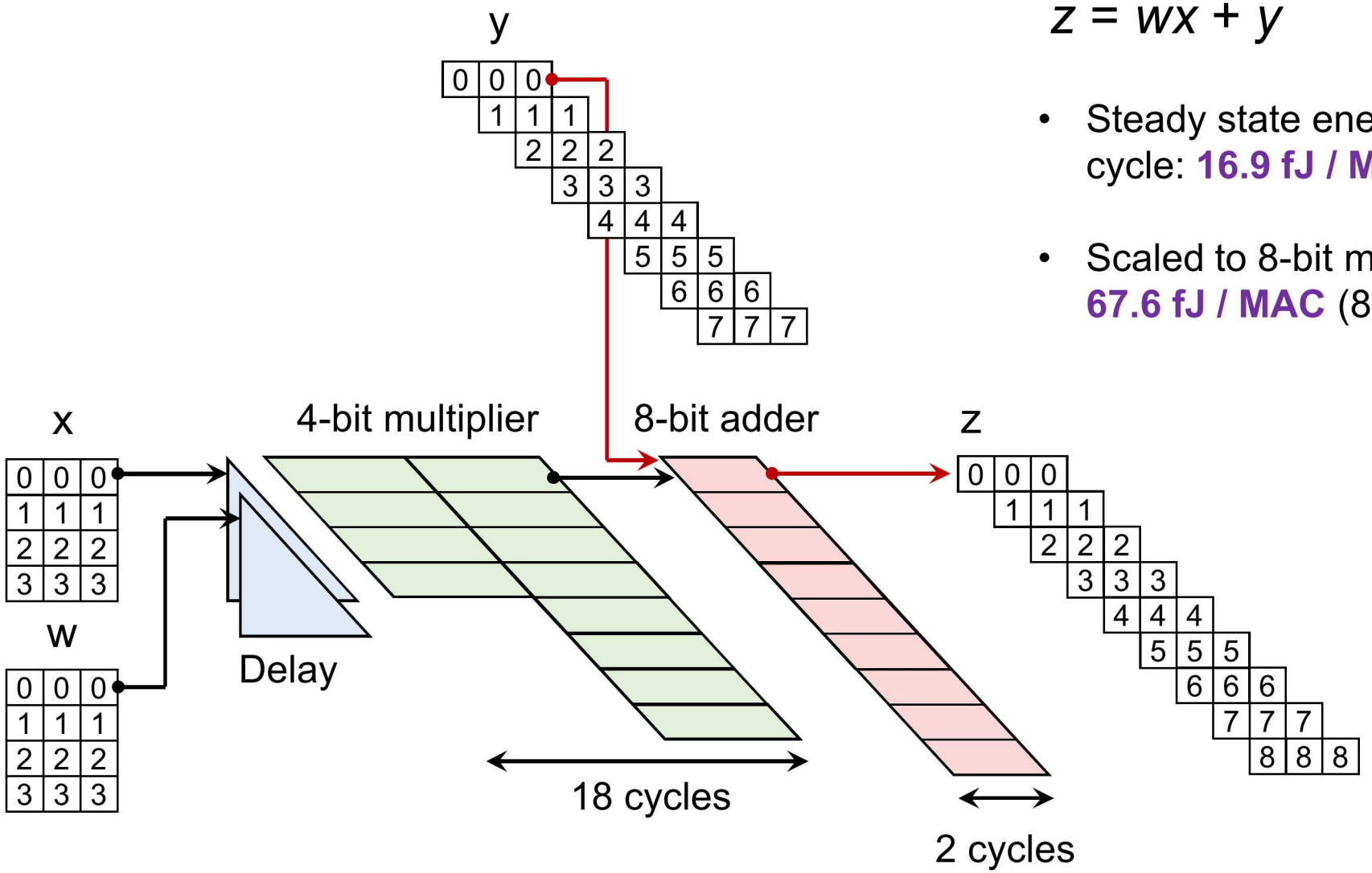
$y_A = \mathbf{W}x_A$
$y_B = \mathbf{W}x_B$
$y_C = \mathbf{W}x_C$
⋮



Systolic matrix multiplication in a domain wall processor



Domain-wall multiply-accumulate (MAC) unit



$z = wx + y$

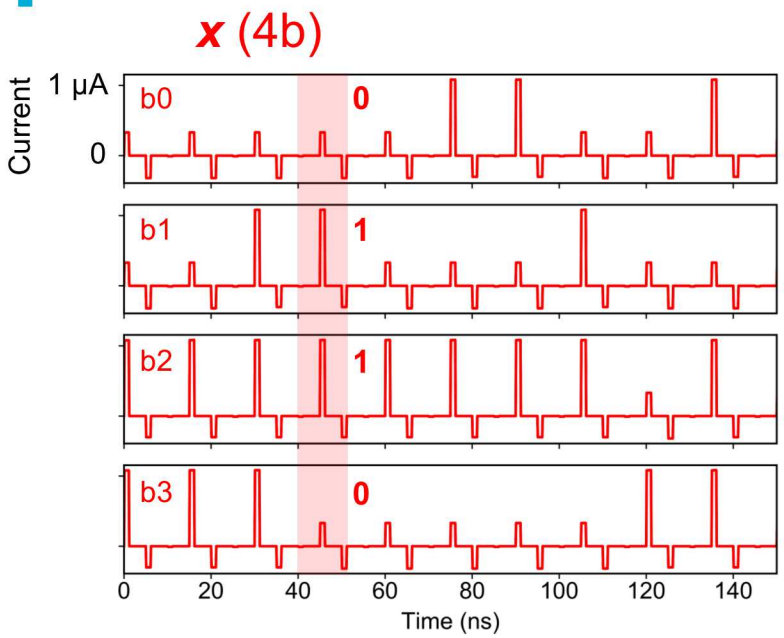
- Steady state energy consumption per cycle: **16.9 fJ / MAC** (4-bit)
- Scaled to 8-bit multiplier, 32-bit adder: **67.6 fJ / MAC** (8-bit)

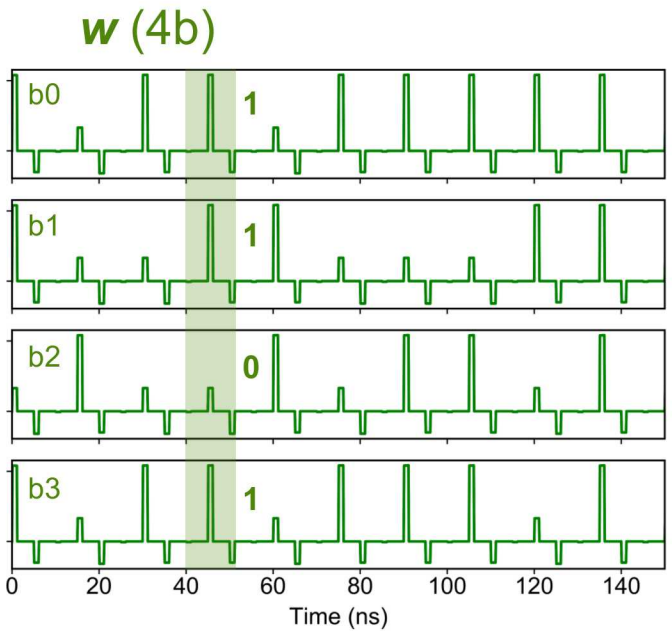
SOT MTJ
$J_{th} = 2.4 \times 10^{10}$ A/m^2
$V_{CLK} = 12.5$ mV
T = 15 ns
L = 120 nm

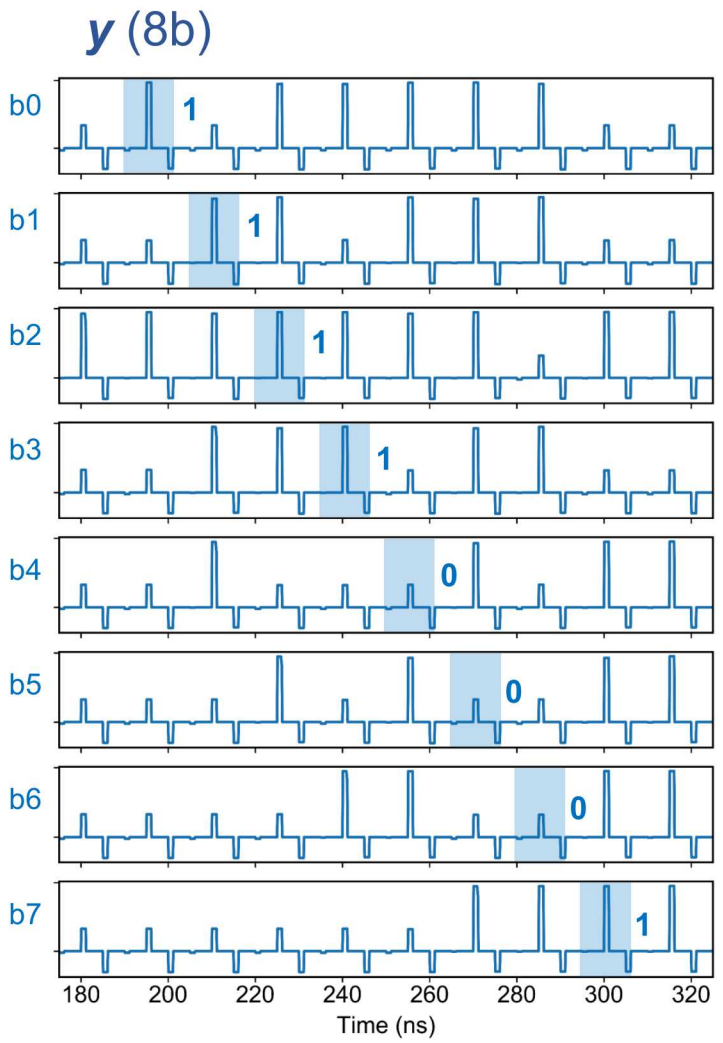
Domain-wall multiply-accumulate (MAC) unit

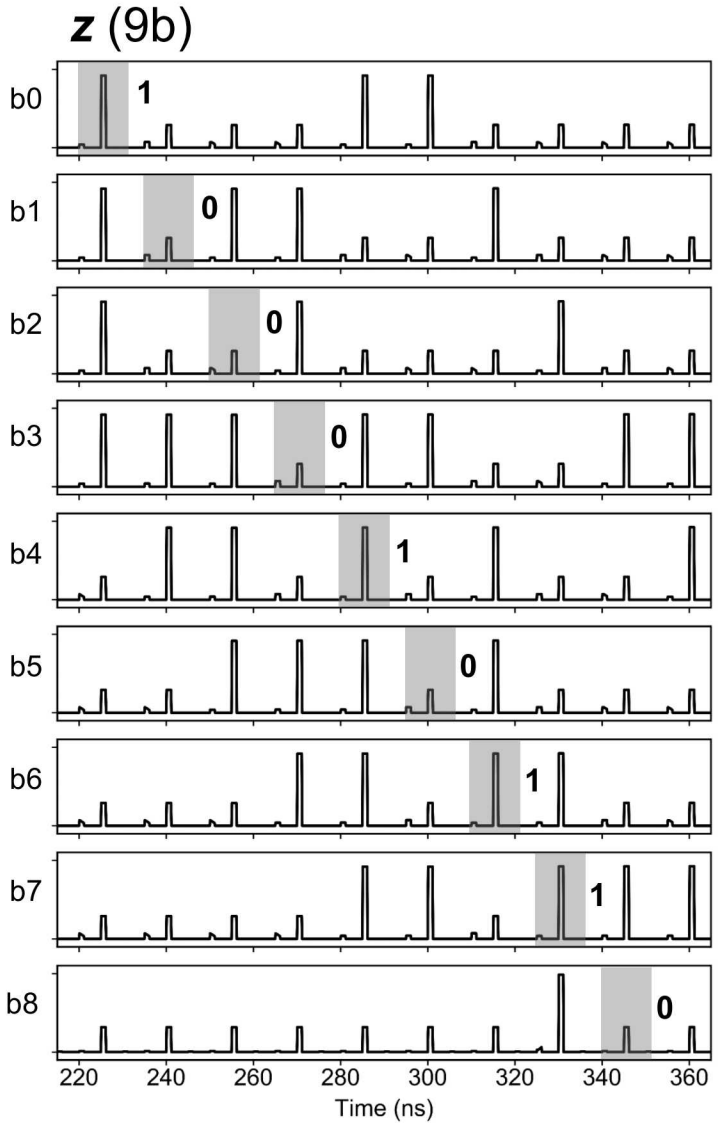
SOT MTJ
J_{th} = 2.4 × 10^{10} A/m^2
V_{CLK} = 12.5 mV
T = 15 ns
L = 120 nm

$z = wx + y$

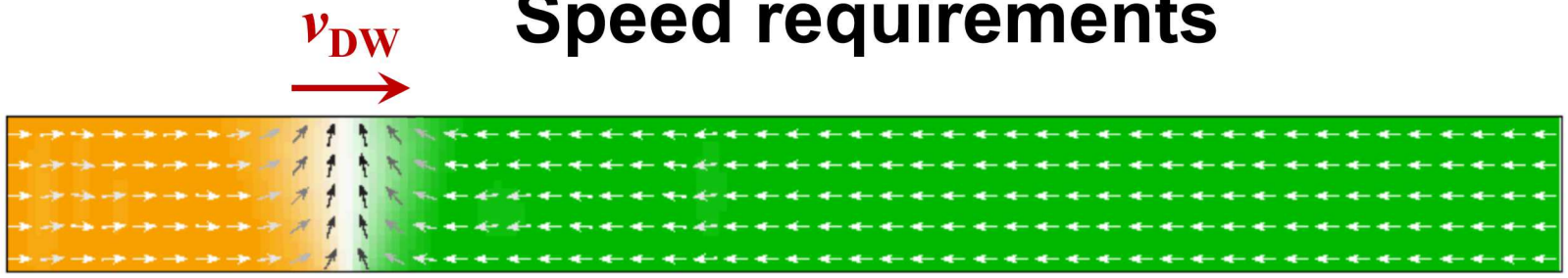








Speed requirements



The throughput target of any compute application can be directly translated to a material / device requirement:

Google TPUv1 (28nm)

256 × 256 systolic CMOS 8-bit MAC array at 700 MHz

Peak throughput = 2×256×256×700 MHz = **92 TeraOps/s**

Domain wall processor

256 × 256 micro-pipelined DW array

Peak throughput = 2×256×256 / (3×gate delay)

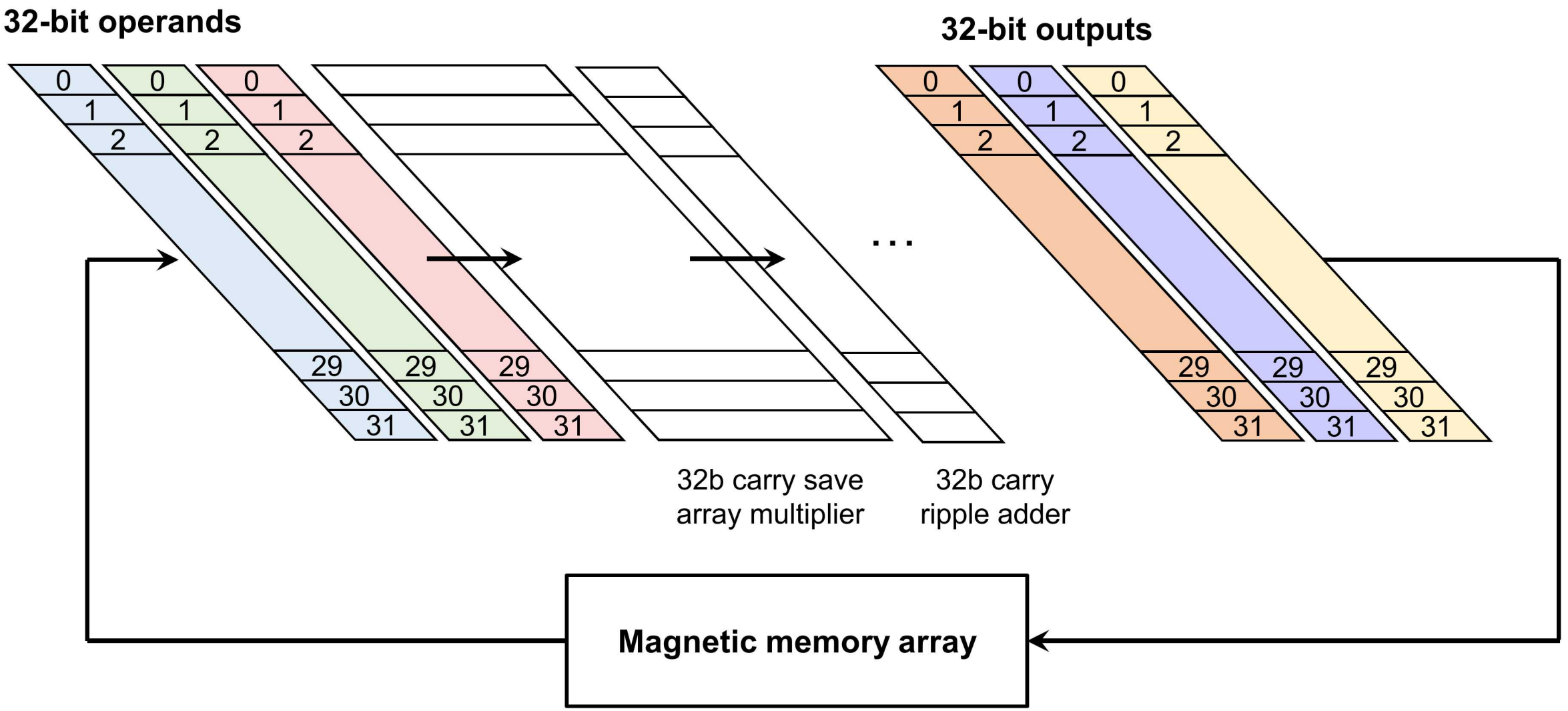
Assume $L_{\text{eff,track}}$ = 75 nm
To match CMOS peak throughput, need

DW velocity v_{DW} > **158 m/s**

N. Jouppi et al, arXiv [cs.AR], 2017

S. Velez et al, Nature Comm 2019

For better efficiency, co-design domain wall processing with memory structure



Conclusions

- Domain walls implement a form of stateful logic that enables **micro-pipelining**
  - Well suited to large, regular computations (e.g. matrix multiplication)
  - Unlike CMOS, throughput is independent of complexity of operation and depends directly on device properties
  - Non-volatile: mitigate leakage
  - Logic can be closely co-integrated with digital memory, or can be organized as analog compute-in-memory arrays

Conclusions

- Domain walls implement a form of stateful logic that enables **micro-pipelining**
  - Well suited to large, regular computations (e.g. matrix multiplication)
  - Unlike CMOS, throughput is independent of complexity of operation and depends directly on device properties
  - Non-volatile: mitigate leakage
  - Logic can be closely co-integrated with digital memory, or can be organized as analog compute-in-memory arrays

Thank you!

+ others

Dr. Chris Bennett
Sandia National Labs

Dr. Ben Feinberg
Sandia National Labs

Xuan Hu
UT Dallas

Prof. Joseph Friedman
UT Dallas

Prof. Jean Anne Incorvia
UT Austin

Dr. Matt Marinella
Sandia National Labs