ECE 598NSG/498NSU
Deep Learning in Hardware
Fall 2020

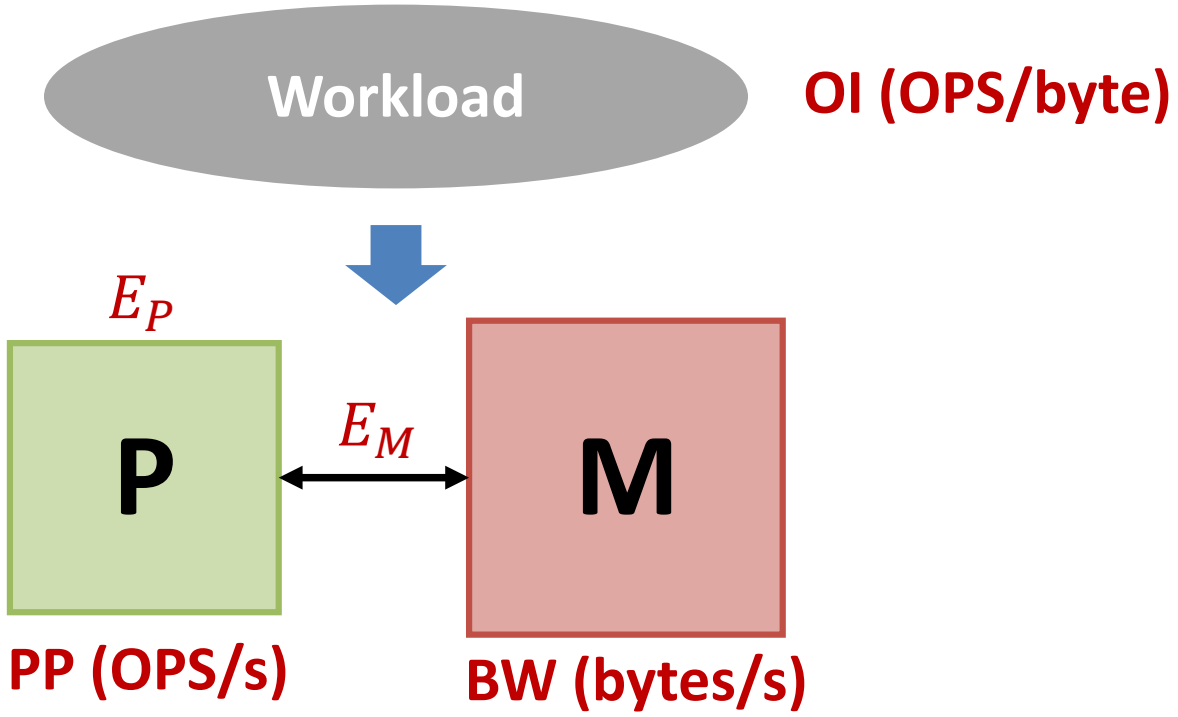
Energy Models

Naresh Shanbhag
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

http://shanbhag.ece.uiuc.edu

- the goal of this lecture is to understand the where energy is dissipated in hardware, i.e., the source of energy-per-MAC and energy-per-access

DNN Architecture



- Employed for hardware benchmarking
- Energy/OP (E_P); Peak OP/s (PP); Bytes read/s (BW); Energy/byte (E_M)

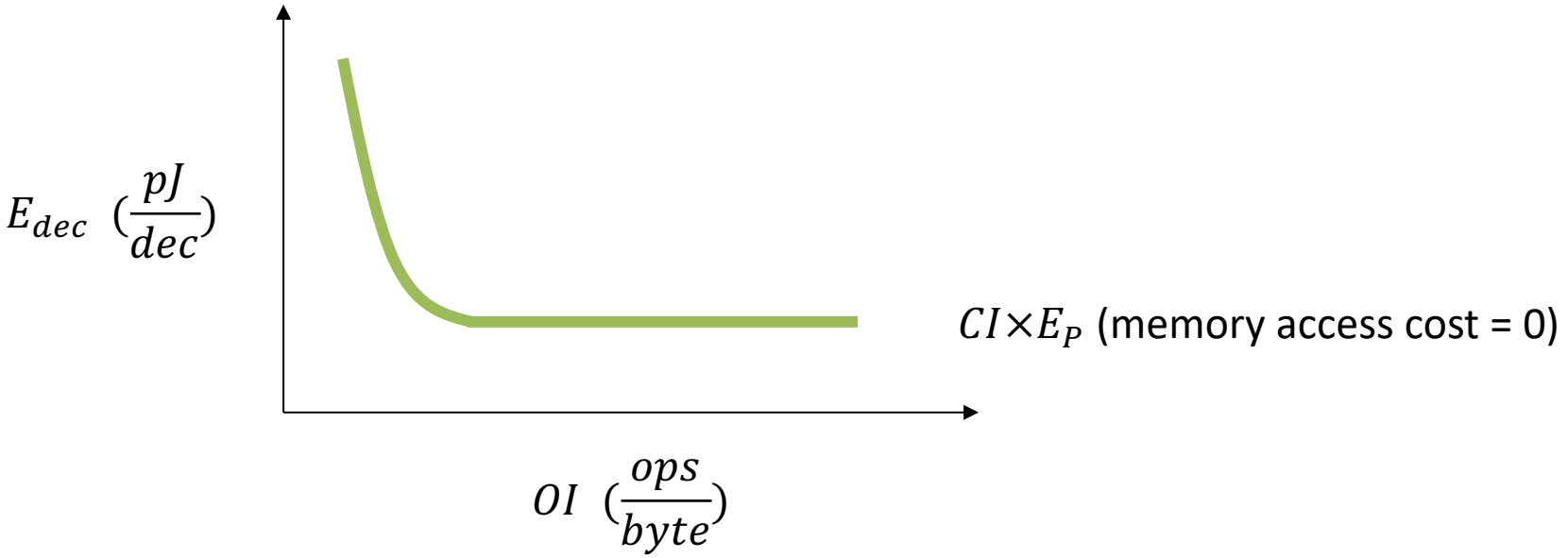
System Energy Efficiency – Floorline Plot

(Shanbhag, Dbouk, Sakr, ECE 498NSU/ECE 598NSG 2019)

- Decision energy (pJ/dec):

$$E_{dec} = CI \times E_P + M_{dec} \times E_M = CI \times E_P + \frac{CI}{OI} \times E_M = CI \left(E_P + \frac{E_M}{OI} \right) = CI \times E_P \left(1 + \frac{E_R}{OI} \right)$$

Floorline plot

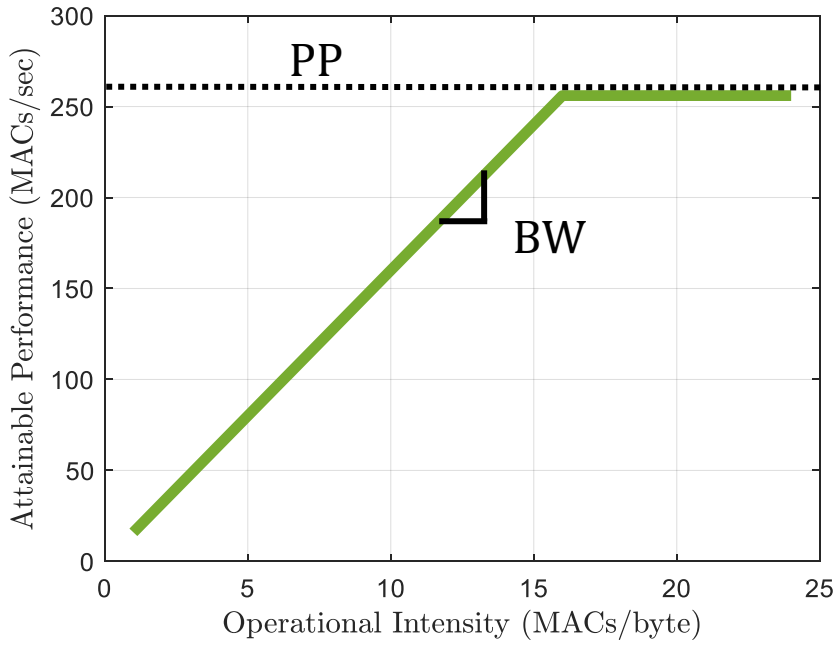


Energy Models for Computation (E_P)

Modeling $E_P, 1/PP, \frac{E_P}{PP} = EDP$

Roofline

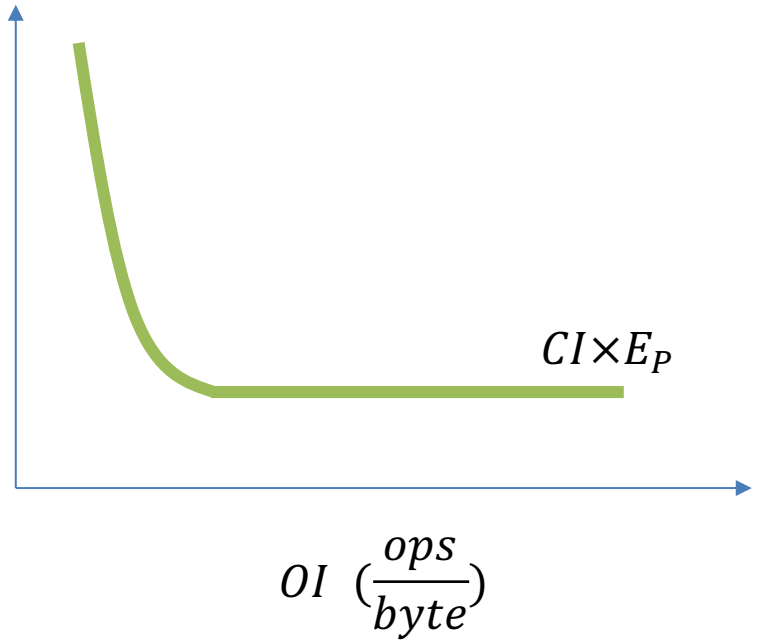
$$AP = \min(PP, OI \times BW)$$



Floorline

E_{dec} $(\frac{pJ}{dec})$

$$CI \times E_P \left(1 + \frac{E_M/E_P}{OI}\right)$$



OI $(\frac{ops}{byte})$

- PP and E_P are dependent circuit parameters

Energy & Power Consumption in Digital Circuits

- Energy is consumed in digital circuits whenever it is powered up, $V_{dd} > 0$.
- Two sources:
  - ***dynamic energy*** E_{dyn} caused by signal transition at any circuit node
  - standby or ***leakage energy*** E_{lkg} cause by off state currents in MOS switches
- The total energy consumption:

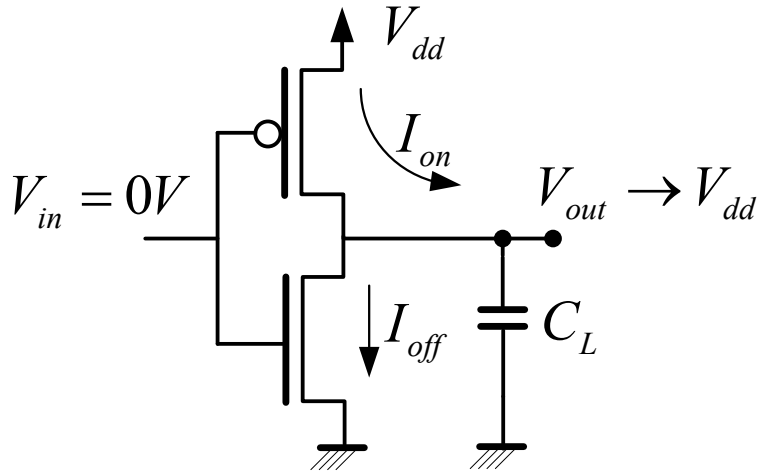
$$E_{tot} = E_{dyn} + E_{lkg}$$

- The total power consumption is given by

$$P = \frac{E_{tot}}{T_p}$$

T_p: delay of the circuit, typically the clock period T_{clk}

Gate Level Dynamic Energy Consumption (E_{dyn})



- If a node i with capacitance C_L makes a $0 \rightarrow 1$ (or $1 \rightarrow 0$) transition, then

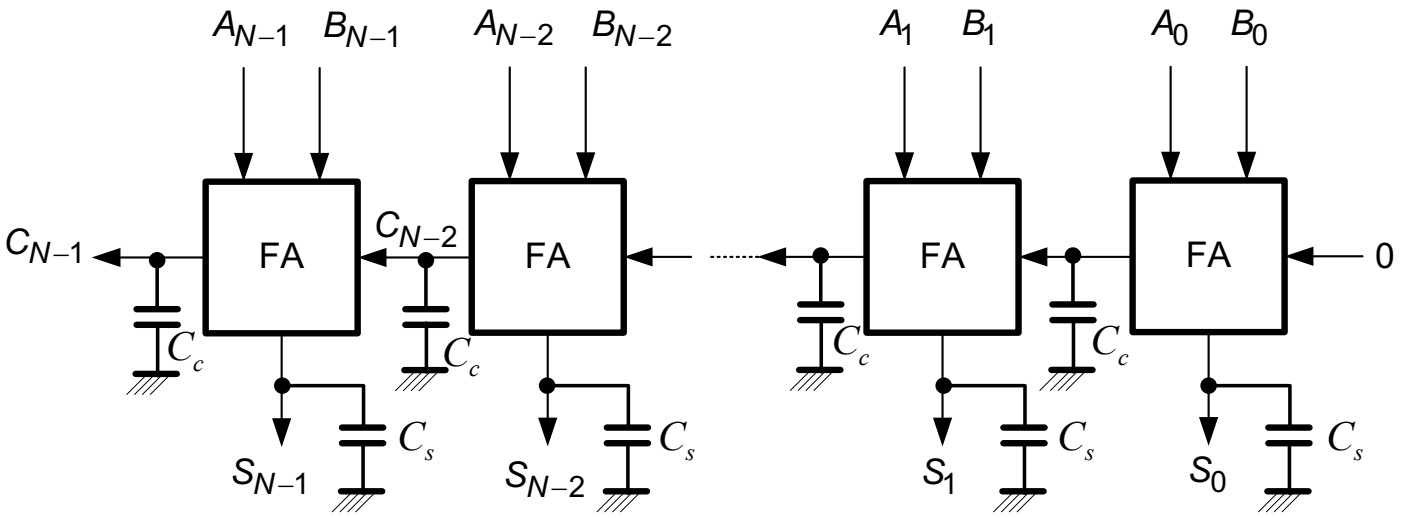
$$E_{dyn} = 0.5 C_L V_{dd}^2$$

- In general,

$$E_{dyn} = \alpha C_L V_{dd}^2$$

where α is the **activity factor** = 0.3-0.5

Dynamic Energy of a Ripple Carry Adder (RCA)



$$E_{dyn,RCA} = \left[\sum_{i=1}^{N_g} \alpha_i C_i\right] V_{dd}^2 \approx N_g \alpha C_g V_{dd}^2$$

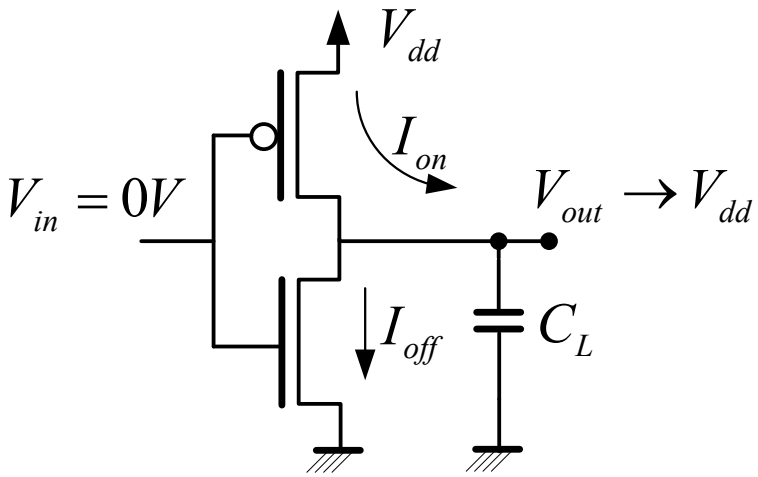
$E_{dyn,RCA}$: dynamic energy per clock period

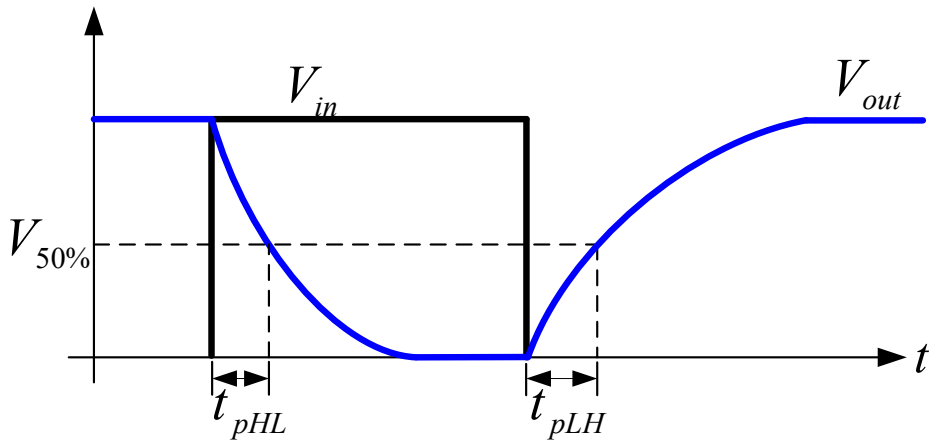
α_i: activity factor at i^{th} node; C_i: capacitance at the i^{th} node

N_g: number of nodes in the circuit; C_g average node capacitance

V_{dd}: supply voltage

Gate Delay



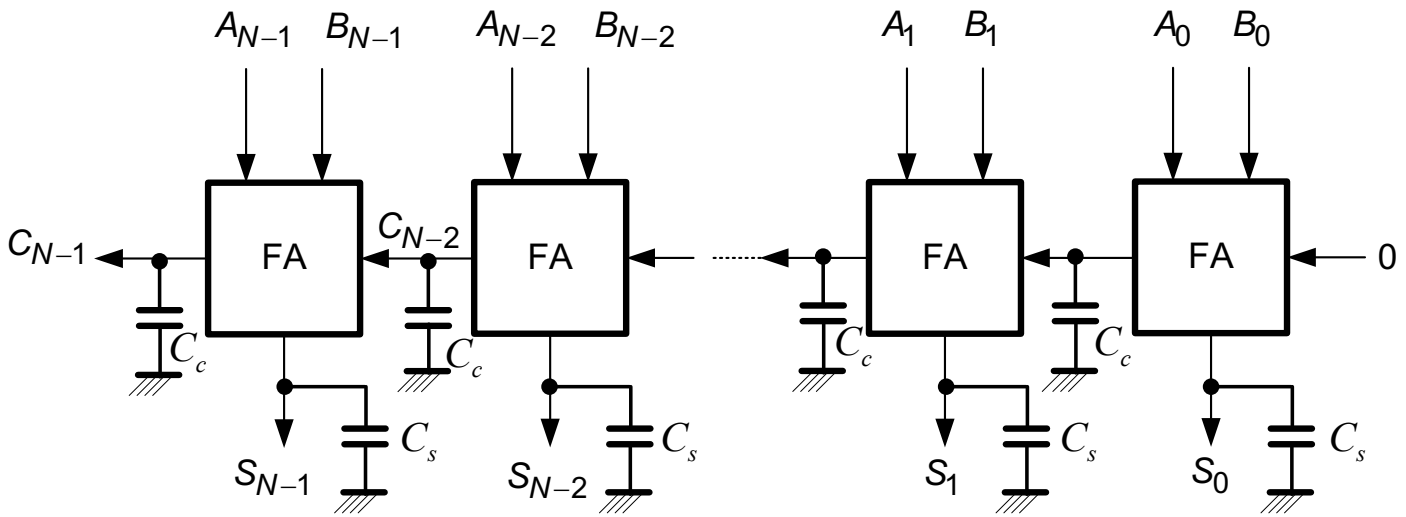


- Delay in occurs due to the finite time needed to charge/discharge capacitances.
- Delay (50% input-to-50% output) of a gate is defined as:

$$T_p = \beta \frac{C_L V_{dd}}{I_{on}} = \beta \frac{C_L V_{dd}}{KW C_{ox} v_{sat} (V_{dd} - V_t)}$$

where C_L: output capacitance. I_{on} is the ON-current.

RCA Delay



- RCA delay

$$T_{RCA,N} = NT_{FA} = \beta \frac{NC_cV_{dd}}{I_{on(sup\ or\ sub)}} = \beta \frac{C_{cp}V_{dd}}{I_{on(sup\ or\ sub)}}$$

- $C_{cp} = NC_c$ is the capacitance of the RCA's critical path

MAC Energy and Delay Models

- Energy model

$$E_{tot} = E_P = \left[\sum_{i=1}^{N_o} \alpha_i C_i\right] V_{dd}^2 + N_g V_{dd} I_S \left(\frac{W}{L}\right) e^{\frac{V_{dd}-V_t}{nV_T}} \left(1 - e^{-\frac{V_{dd}}{V_T}}\right) T_p]$$

- Delay model

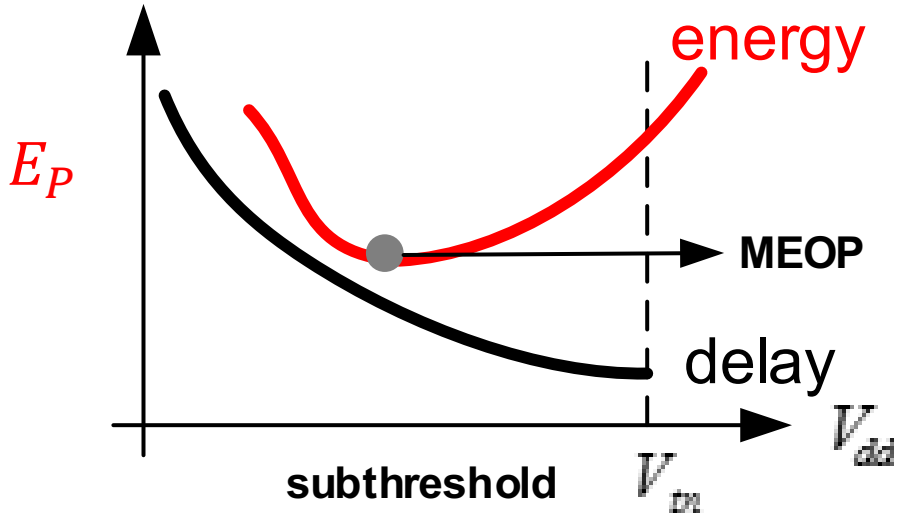
$$T_p = \beta \frac{C_{cp} V_{dd}}{I_{on}}$$

  - I_{on} can be in superthreshold or subthreshold

- How to minimize E_P?

Energy-Delay Trade-off

[Abdallah, Shanbhag, IEEE Embedded Systems Letters, Dec. 2010]



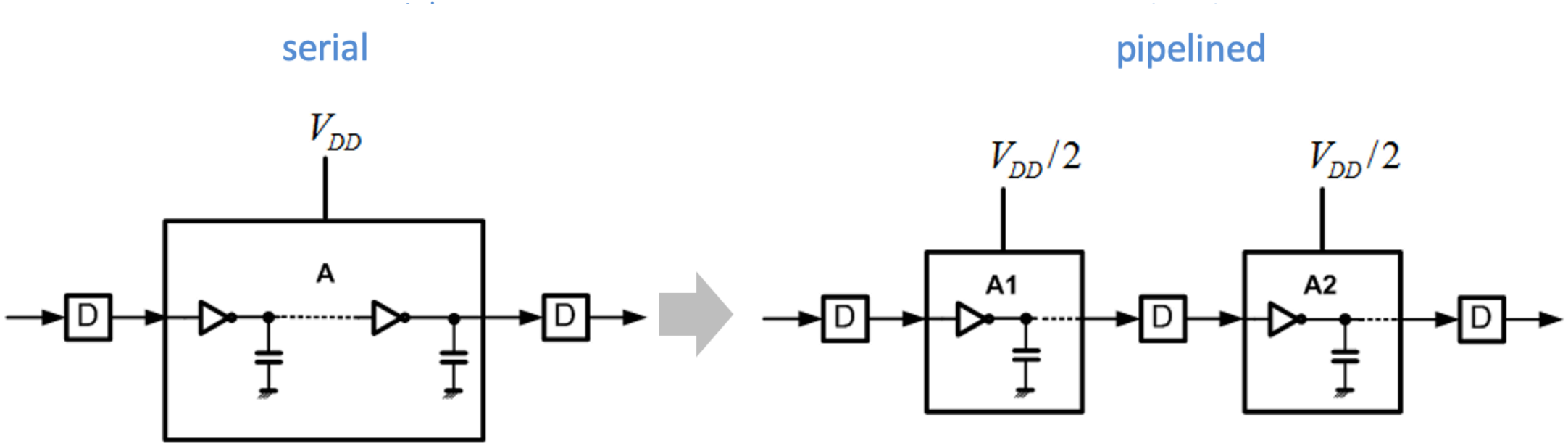
- E_P achieves a ***minimum at*** $\rightarrow$ **minimum energy operating point** (MEOP) (V_{dd}^*, f^*, E_P^*)
- MEOP occurs in subthreshold $(V_{dd} < V_t\ (200mV - 400mV))$

Methods to Reduce E_P

- reduce $V_{dd} \rightarrow$ reduce f_{clk} or pipelining/parallel processing (high-speed architectural methods) to accommodate increased circuit delay
- reduce C_L
- power and clock gating → shut down blocks not being used

- operate in subthreshold region → drastic choice but effective in energy reduction

Pipelining for Low-Power



- basic idea: trade-off throughput increase from pipelining with power via supply voltage V_{dd} reduction

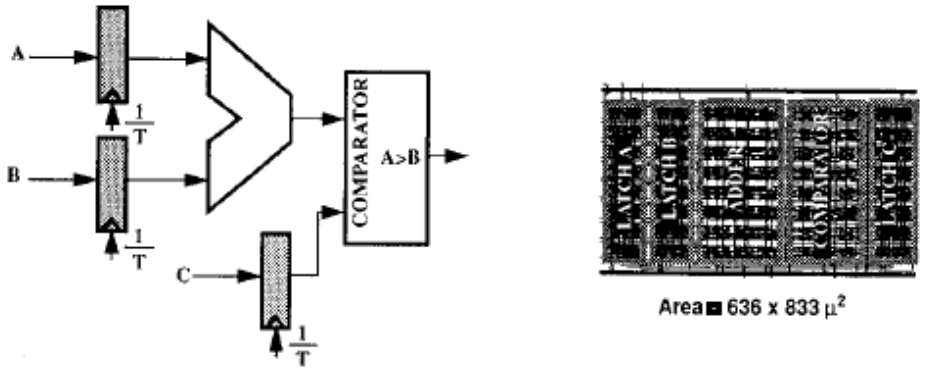


Fig. 7. A simple data path with corresponding layout.

$$P_{ref} = C_{ref} V_{ref}^2 f_{ref}$$

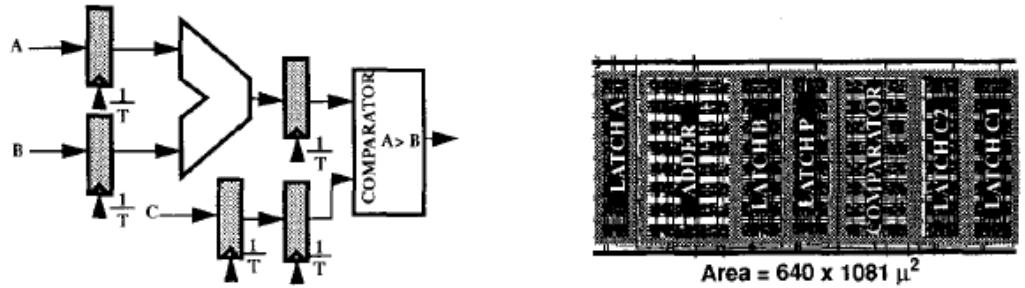
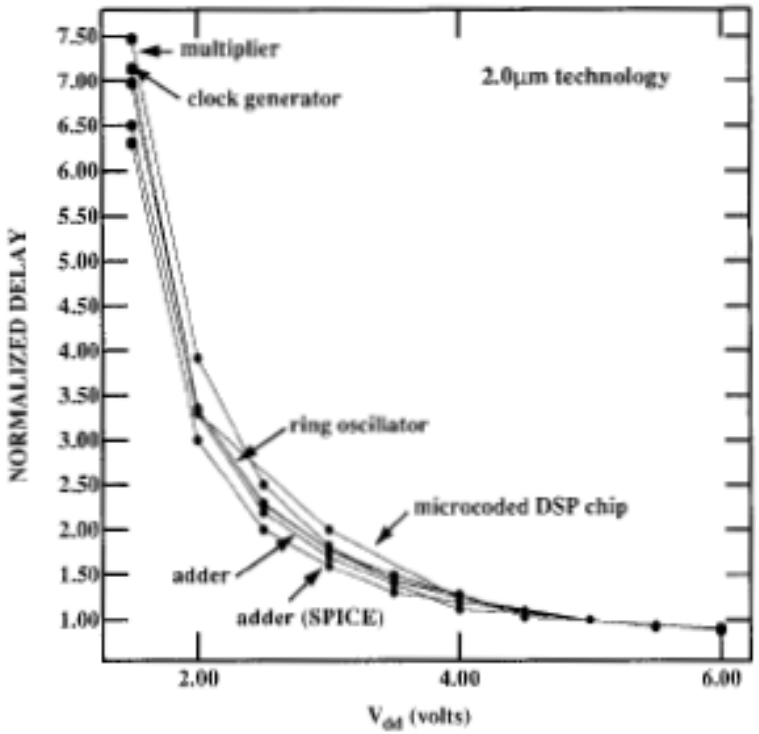


Fig. 9. Pipelined implementation of the simple data path.



delay doubles when supply voltage is reduced from $5V$ to $2.9V$

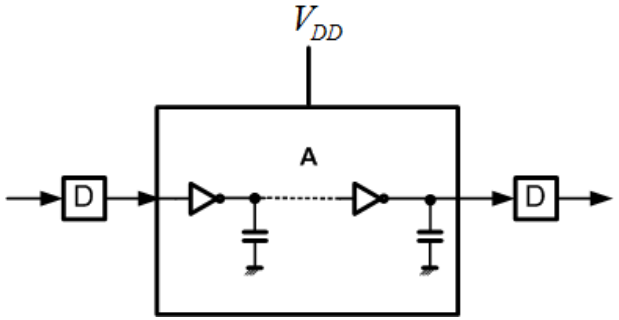
$$P_{pipe} = C_{pipe} V_{pipe}^2 f_{pipe}$$

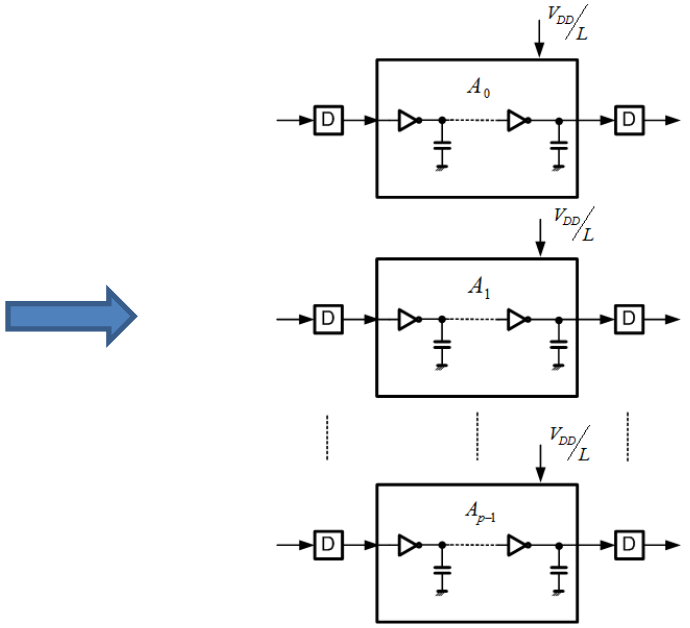
$$= (1.15 C_{ref})(0.58 V_{ref})^2 f_{ref} = 0.39 P_{ref} \rightarrow 2.5X \text{ reduction}$$

- Other pipelining benefit: energy reduction due to glitches and hazard reduction.

- analysis ignores leakage power, and already reduced (~1V) nominal supply voltages in modern process nodes
- in practice, this idealized scenario possible only for small values of M
- as M increases and V_{dd} reduces
  - exponential increase in delay → near/subthreshold operation
  - energy and delay overhead due to pipelining registers

Parallel Processing as a Low-Power Technique





$$T_{cp} = \frac{C_L L}{k_n V_{DD}} = T_{block} = LT_s$$

$$f_{block} = \frac{f_s}{L} = P_{D,serial}\frac{P}{L^3}$$

$$P_{D,par} = PC_L\left(\frac{V_{DD}}{L}\right)^2 f_{block}$$

$$\frac{V_{DD}}{L}$$

- P-parallel blocks can operate at $\frac{V_{dd}}{L}$ supply. Therefore, the dynamic power is given by
- If $P{=}L \rightarrow L^2$ power reduction feasible

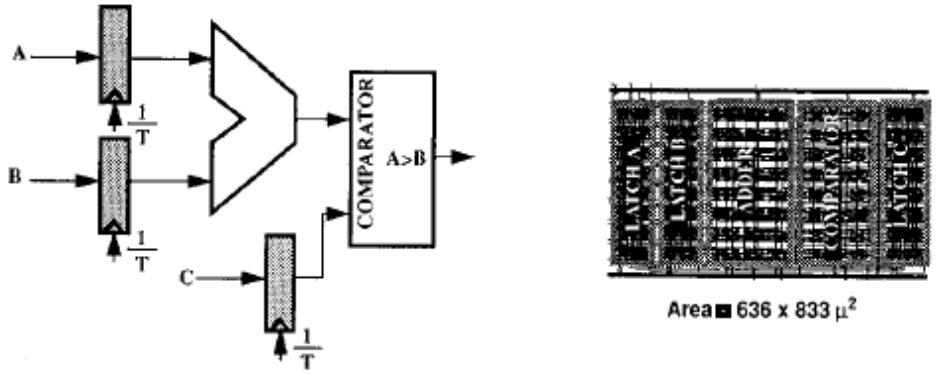


Fig. 7. A simple data path with corresponding layout.

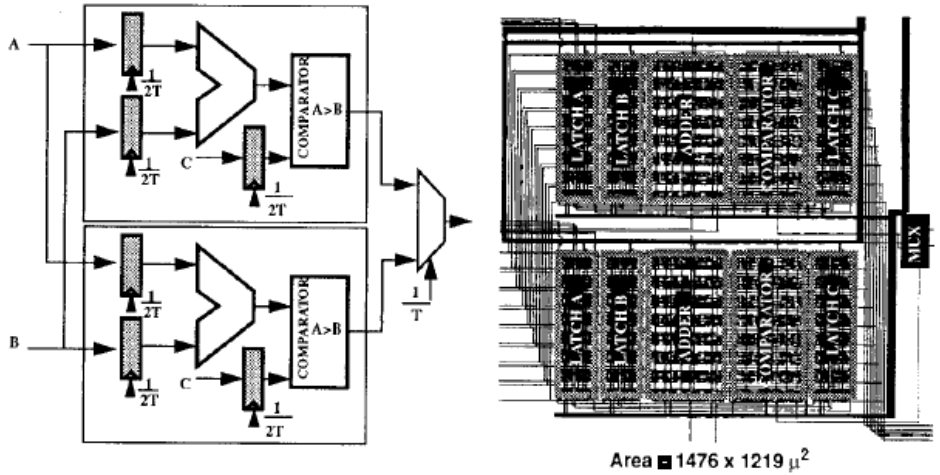


Fig. 8. Parallel implementation of the simple data path.

$$P_{ref} = C_{ref} V_{ref}^2 f_{ref}$$

Delay doubles when supply voltage is reduced from $5V$ to $2.9V$

$$P_{par} = C_{par} V_{par}^2 f_{par}$$

$$= (2.15C_{ref})(0.58V_{ref})^2 \frac{f_{ref}}{2} = 0.36P_{ref} \rightarrow 2.8X \text{ reduction}$$

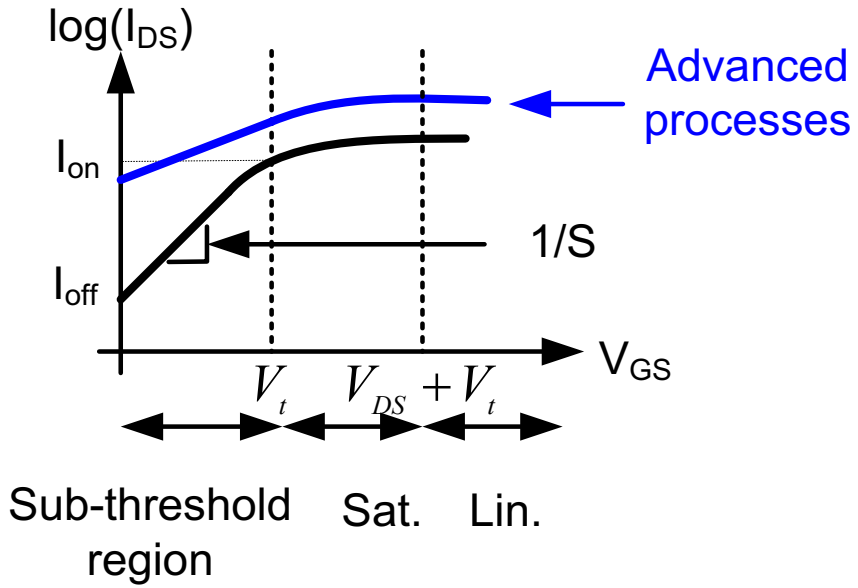
Methods to Reduce E_P

- reduce V_{dd} → reduce f_{clk} or pipelining/parallel processing (high-speed architectural methods) to accommodate increased circuit delay
- reduce C_L
- power and clock gating → shut down blocks not being used

- operate in subthreshold region → drastic choice but effective in energy reduction

Subthreshold Computing

(2000-10)

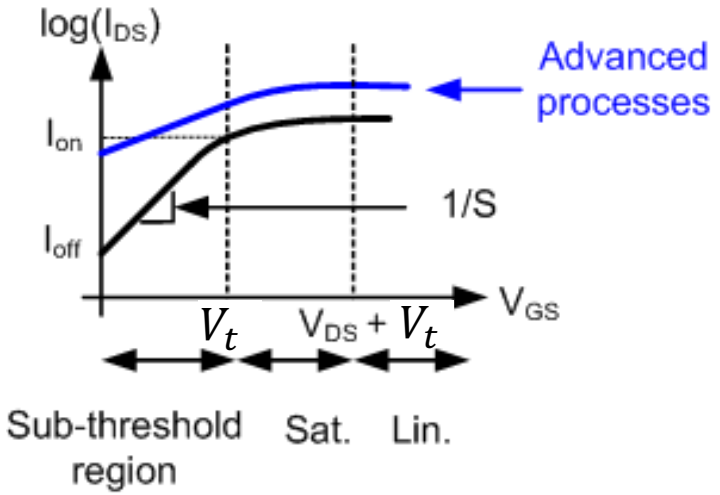


$$I_D = I_S e^{\frac{(V_{GS}-V_t)}{nV_{TH}}} \left(1 - e^{\frac{-V_{DS}}{V_{TH}}}\right)(1 + \lambda V_{DS})$$

$V_{TH} = \frac{kT}{q}$: thermal voltage (26mV @ 25C)

- compute using OFF transistor currents
- idea is to reduce V_{dd} below transistor threshold voltage V_t
- subthreshold designs – logic, memory, analog, interconnect
- key issue – very slow and very unreliable (sensitive to PVT variations) -> hot area for IoT start-ups

Subthreshold Operation ($V_{dd} < V_t$)



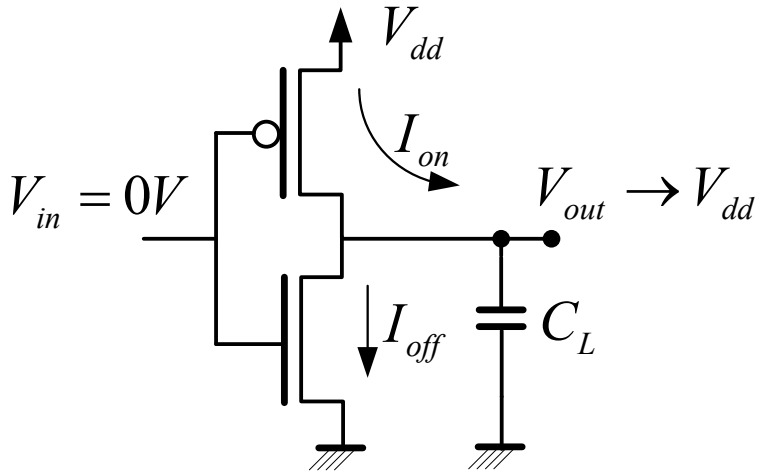
$$I_{D(sub)}(V_{GS}, V_{DS}) = I_S \left(\frac{W}{L}\right) e^{\frac{V_{gs}-V_t}{nV_T}} \left(1 - e^{-\frac{V_{ds}}{V_T}}\right)$$

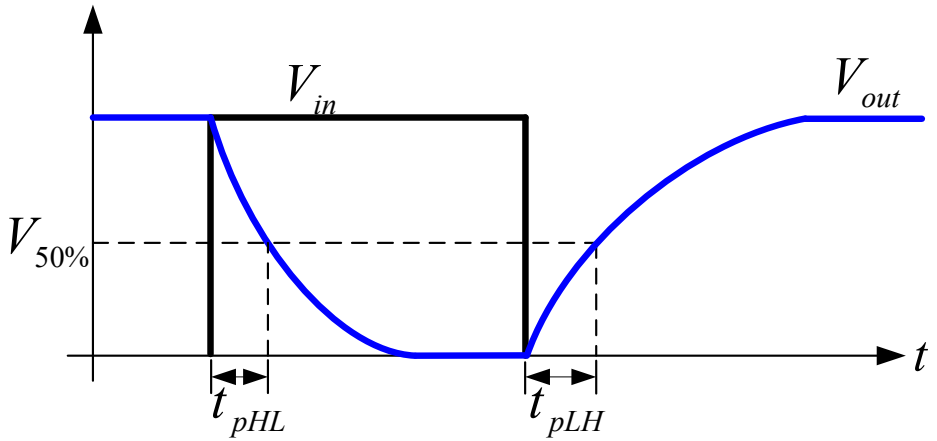
$$I_{on(sub)} = I_{D(sub)}(V_{dd}, V_{dd}) \approx I_S \left(\frac{W}{L}\right) e^{\frac{V_{dd}-V_t}{nV_T}}$$

$$I_{off} = I_{D(sub)}(0, V_{dd}) \approx I_S \left(\frac{W}{L}\right) e^{\frac{-V_t}{nV_T}}$$

- When $V_{dd} < V_t$: subthreshold region. Useful computing can occur in the subthreshold
- MOS in OFF state ($V_{GS} < V_t$) has a:
  - **sub-ON state** ($0 < V_{GS} = V_{dd} < V_t$)
  - **sub-OFF state** ($V_{GS} = 0$)

Gate Delay in Subthreshold





- Delay (50% input-to-50% output):

$$T_p = \beta \frac{C_L V_{dd}}{I_{on}} = \beta \frac{C_L V_{dd}}{I_S \left(\frac{W}{L}\right) e^{\frac{V_{dd}-V_t}{nV_T}}}$$

- Delay increases rapidly as V_{dd} reduces and is *extremely sensitive to PVT variations* due to the exponential

IEEE EMBEDDED SYSTEMS LETTERS, VOL. 2, NO. 4, DECEMBER 2010

Minimum-Energy Operation Via Error Resiliency

Rami A. Abdallah and Naresh R. Shanbhag

- Total energy

$$E_{tot} = N_g C_g V_{dd}^2 (\alpha + \beta L e^{-\frac{V_{dd}}{nV_T}})$$

- N_g: # of gates/blocks in the architecture
- C_g: average load capacitance per block
- α: average activity;
- L: number of gates/blocks in the critical path
- Constants: $V_T = 26\text{mV}$ (thermal voltage); β and n constants

- dynamic energy E_{dyn} ***reduces*** as V_{dd} ***decreases***

$$E_{dyn} = \alpha N_g C_L V_{dd}^2$$

- leakage energy E_{lkg} ***increases*** as V_{dd} ***decreases***

$$E_{lkg} = N_g V_{dd} I_S \left(\frac{W}{L}\right) e^{\frac{-V_t}{nV_T}} \left(1 - e^{-\frac{V_{dd}}{V_T}}\right) T_p$$

- delay T_p ***increases exponentially*** as V_{dd} ***decreases***

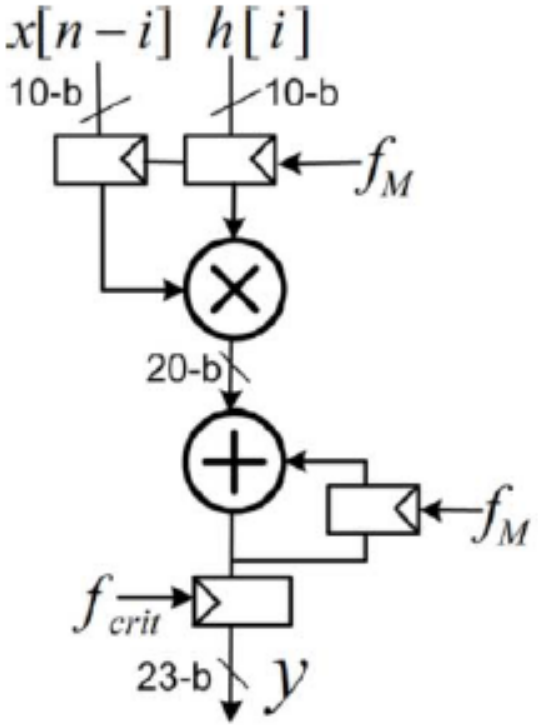
$$T_p = \beta \frac{C_{cp} V_{dd}}{I_S \left(\frac{W}{L}\right) e^{\frac{V_{dd} - V_t}{nV_T}} \left(1 - e^{-\frac{V_{dd}}{V_T}}\right)}$$

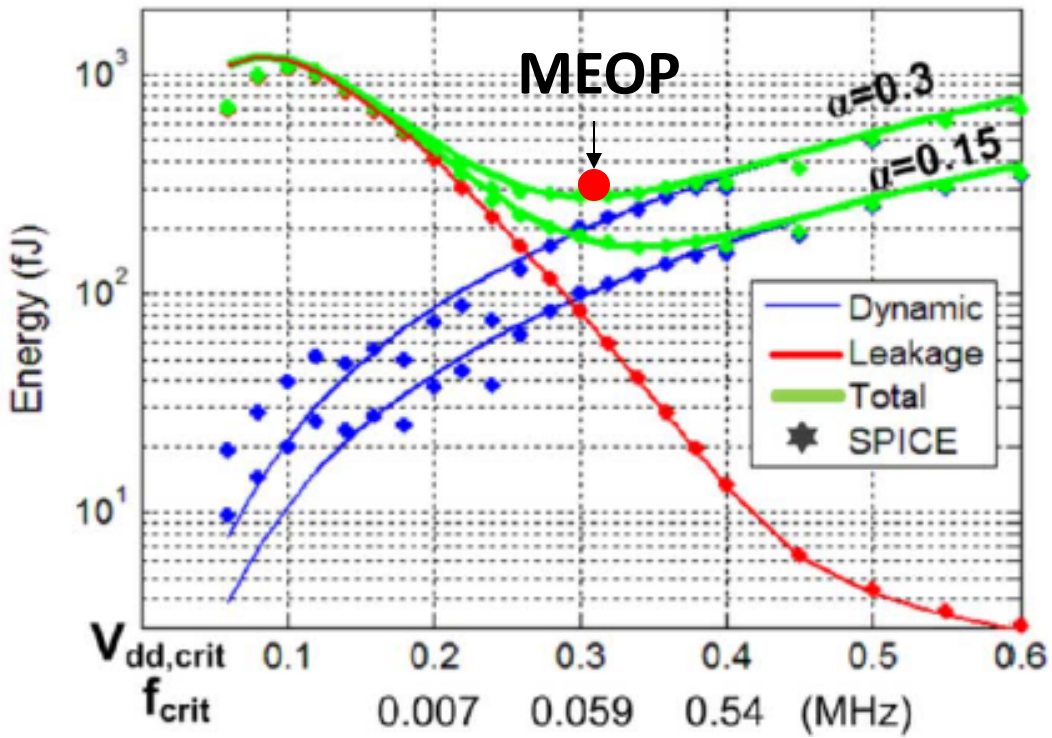
- Total energy

$$E_{tot} = N_g C_g V_{dd}^2 (\alpha + \beta L e^{-\frac{V_{dd}}{nV_T}})$$

- first term reduces quadratically with V_{dd}
- second term increases exponentially as V_{dd} reduces
- E_{tot} reduces initially with V_{dd} and then increases → MEOP
- the MEOP 3-tuple (V_{dd}^*, f^*, E^*) can be calculated numerically

MEOP for a MAC

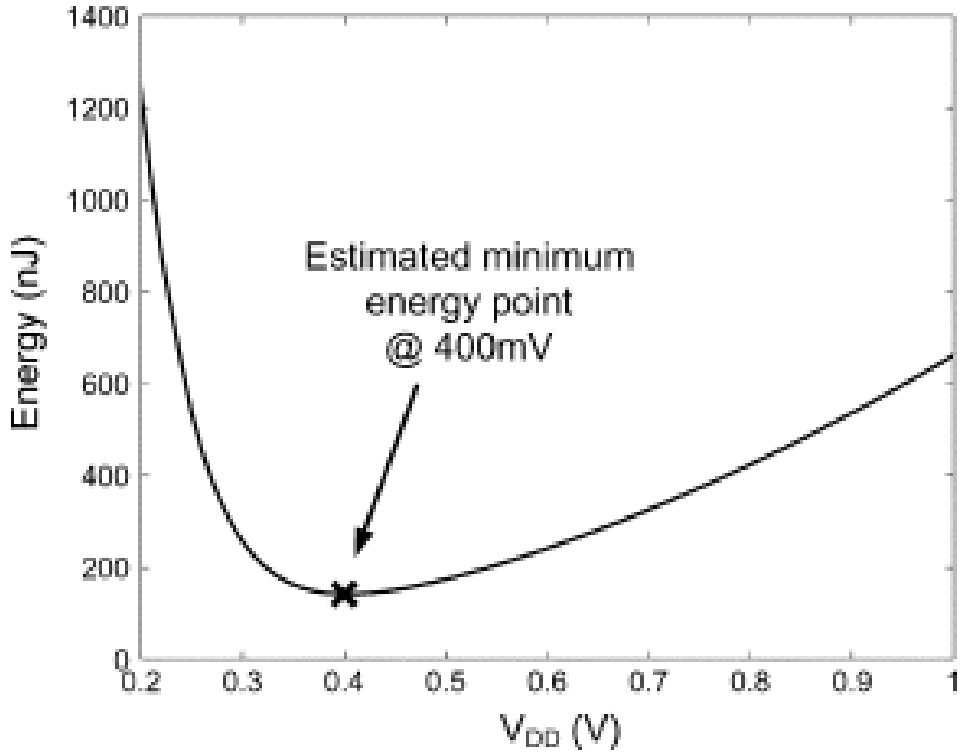


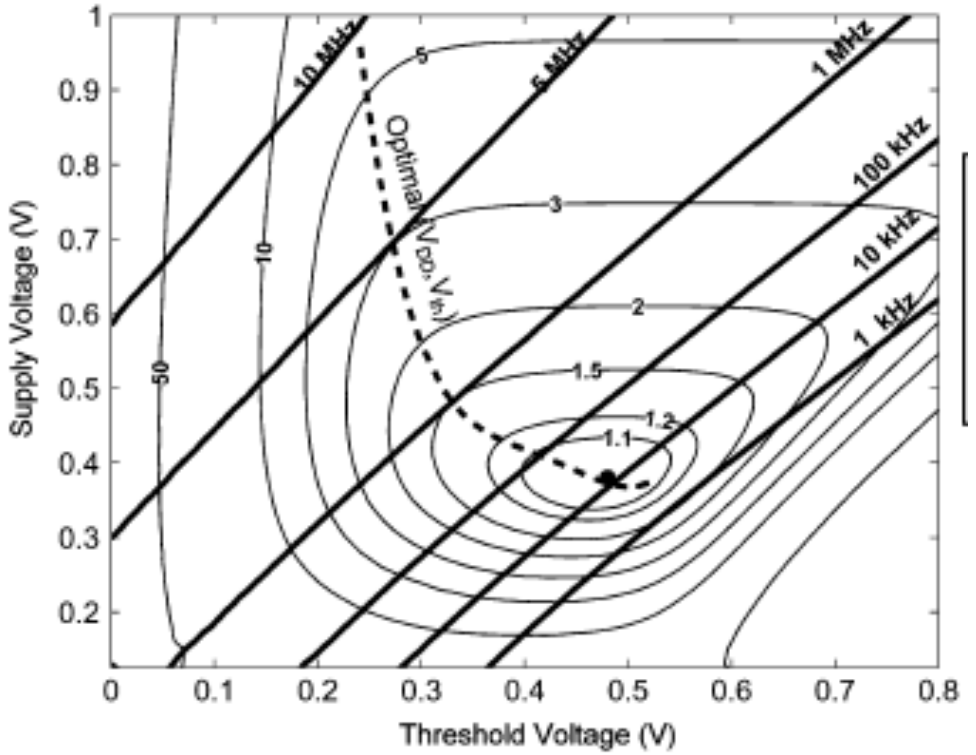


- $10b \times 10b$ MAC; 130nm CMOS;
- MEOP = $(V_{dd}^* = 330mV, f^* = 1.2MHz, E^* = 300fJ)$ for $\alpha = 0.3$

MEOP for FFT

[Wang, Chandrakasan, IEEE J. of Solid-State Circuits, Jan. 2005]





- 16b, 1024-pt FFT processor, 180nm CMOS
- MEOP = $(V_{dd}^* = 400mV, f^* = 10kHz, E^* = 1\ (normalized)$

Derivation of Energy Expression

- let the dynamic energy be given by

$$E_{dyn} = \alpha N_g C_g V_{dd}^2$$

where C_g is the average load capacitance per gate

- the leakage energy is given by

$$E_{lkg} = N_g I_{off} V_{dd} T_p$$

where $I_{off} = I_{D(sub)}(V_{GS} = 0, V_{DS} = V_{dd})$

- set $T_p = T_{cp}$; i.e., the clock period is equal to the critical path delay, and $C_{cp} = LC_g$, where L is the number of logic stages in the critical path

[Abdallah-ESL10]

- the delay is given by

$$T_p = \beta \frac{LC_g V_{dd}}{I_{on(sub)}}$$

where $I_{on(sub)} = I_{D(sub)}(V_{dd}, V_{dd})$, and β is a fitting parameter needed to account for finite rise and fall times.

- thus,

$$E_{lkg} = \beta L N_g V_{dd}^2 C_g \left(\frac{I_{off}}{I_{on}}\right)$$

where $\frac{I_{off}}{I_{on(sub)}} = e^{-\frac{V_{dd}}{nV_T}}$ in the subthreshold region

- adding E_{dyn} and E_{lkg}, we get

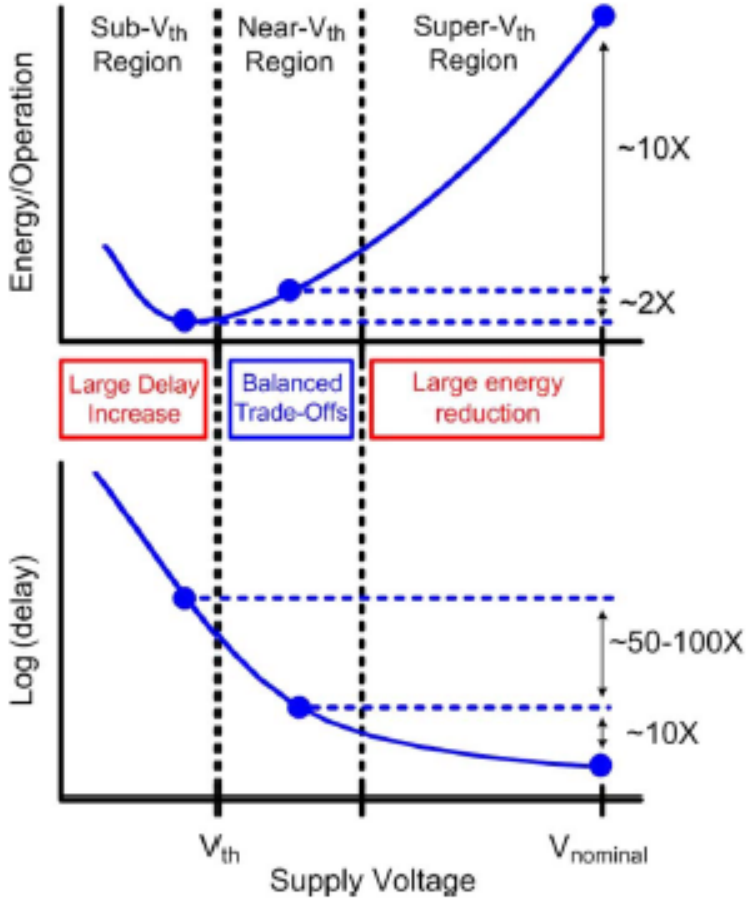
$$E_{tot} = N_g C_g V_{dd}^2 (\alpha + \beta L e^{-\frac{V_{dd}}{nV_T}})$$

Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits

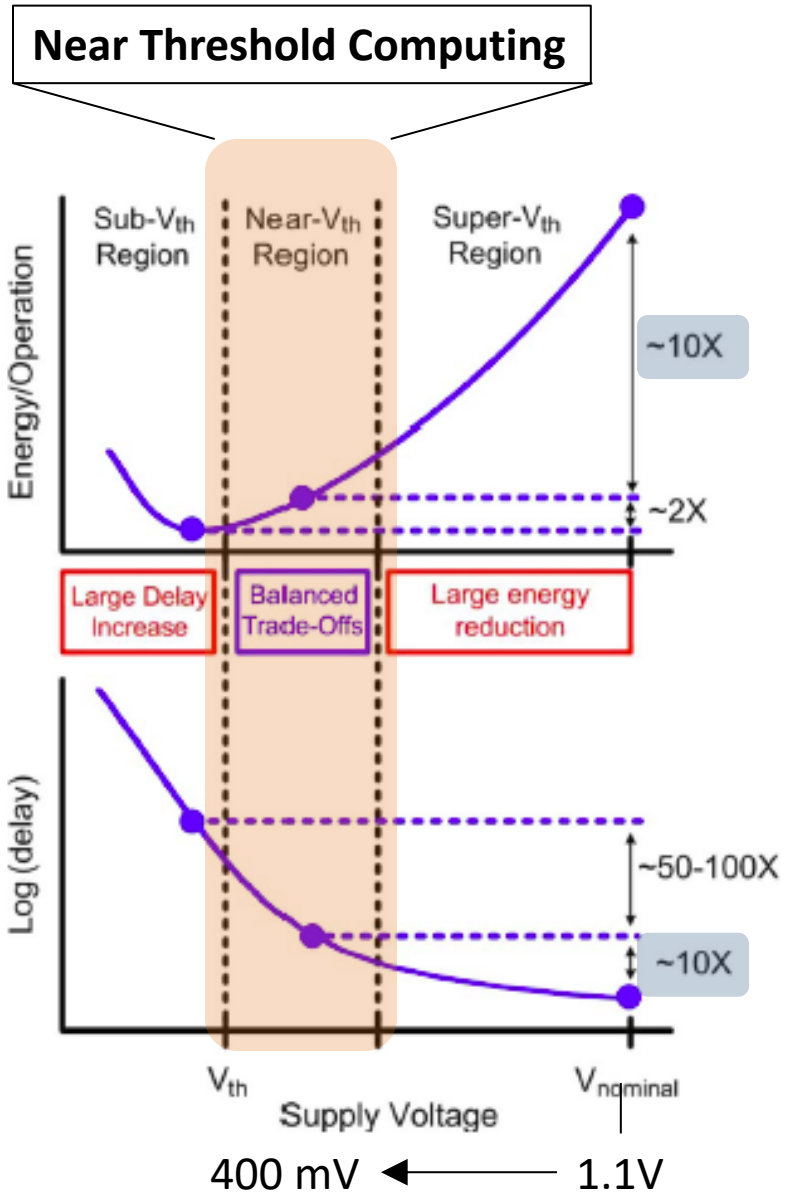
Future computer systems promise to achieve an energy reduction of 100 or more times with memory design, device structure, device fabrication techniques, and clocking, all optimized for low-voltage operation.

By Ronald G. Dreslinski, Michael Wieckowski, David Blaauw, *Senior Member IEEE*, Dennis Sylvester, *Senior Member IEEE*, and Trevor Mudge, *Fellow IEEE*

Vol. 98, No. 2, February 2010 | Proceedings of the IEEE



- delay penalty from NTV to SubT is huge (100X)
- most gains energy savings obtained in going from nominal→ NTV
- but….NTV (and SubT) suffer from increased sensitivity to variations → logic errors due to timing violations can occur



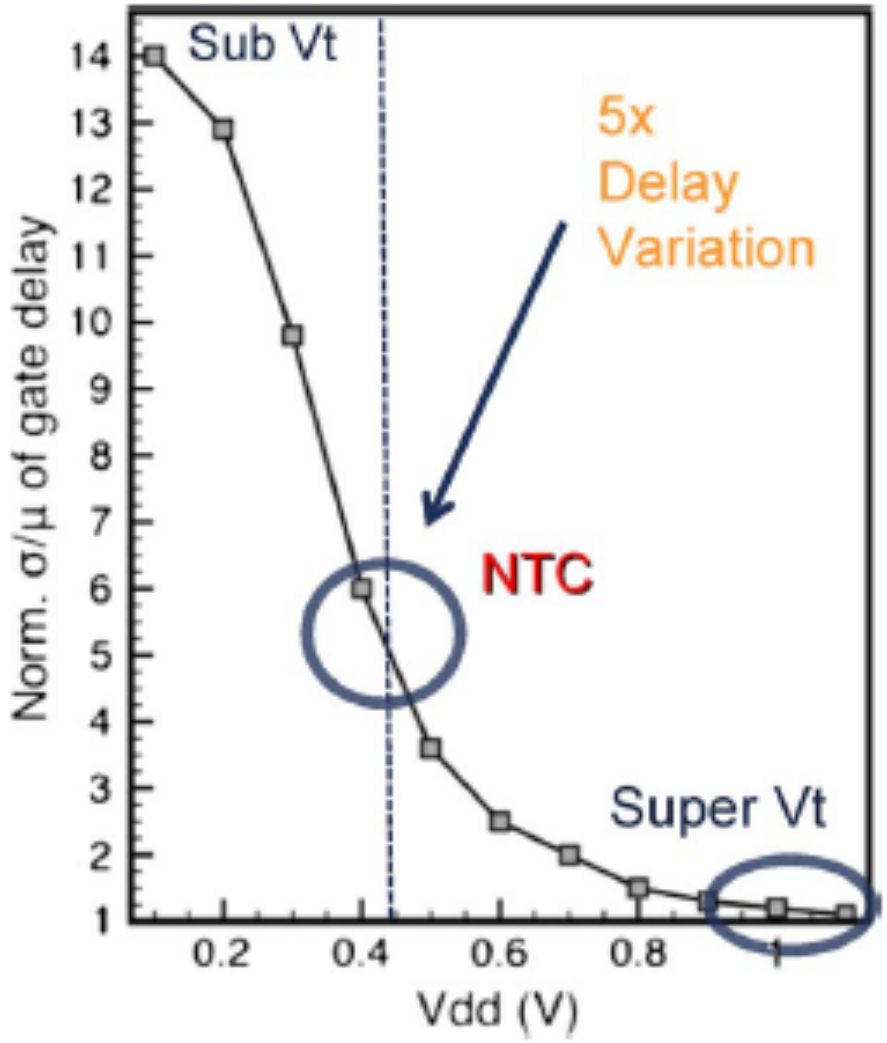
energy efficiency

5X increase in delay **variation**
10^5 X increase in memory **failure** rate
Increased logic **failures**

throughput loss

[Dreslinski-ProcIEEE10]

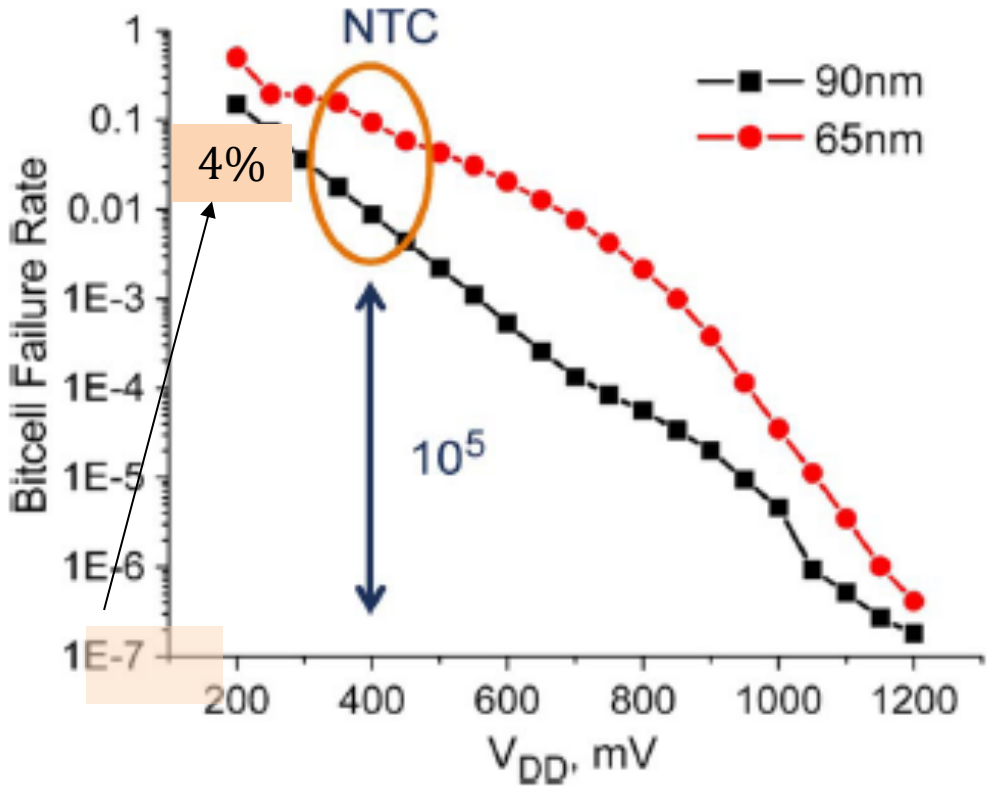
Barriers to NTV Operation



- **Performance loss**: 10X
- **Performance variation**
  - Defined as $\frac{\sigma}{\mu}$ of T_p
  - 1.3X (nominal) → 5X (NTV)
  - 2X (ripple) × 2X (temp)
  - Total: $5{\times}2{\times}2 = 20X$ variation in delay

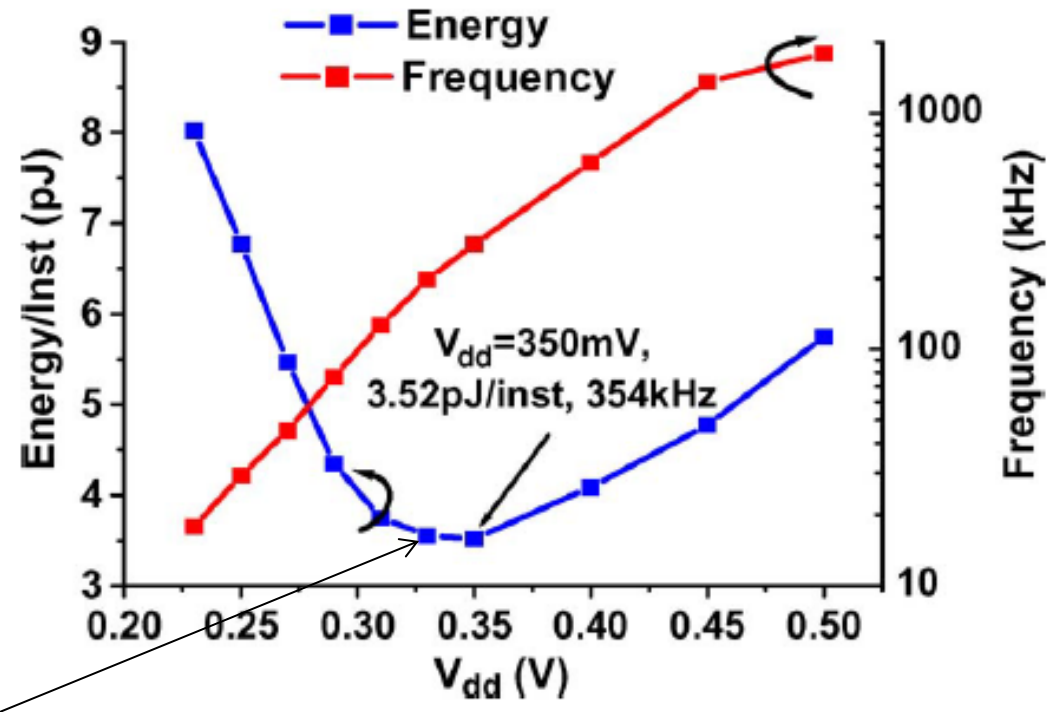
Barriers to NTV Operation

SRAM (memory) bit-cell failure rates



- increased functional failures
- reduced I_D due to increase in V_t (timing failure)
- skewed $I_{Dp}\ vs.\ I_{Dn}$ (write failure)
- local V_t mismatch (read upset)

NTV & Subthreshold Processors



minimum energy operating point (MEOP)

Subliminal Processor

- impact on frequency much more dramatic than energy
- nominal → NTV (0.5V): 7X energy & 11X frequency reductions

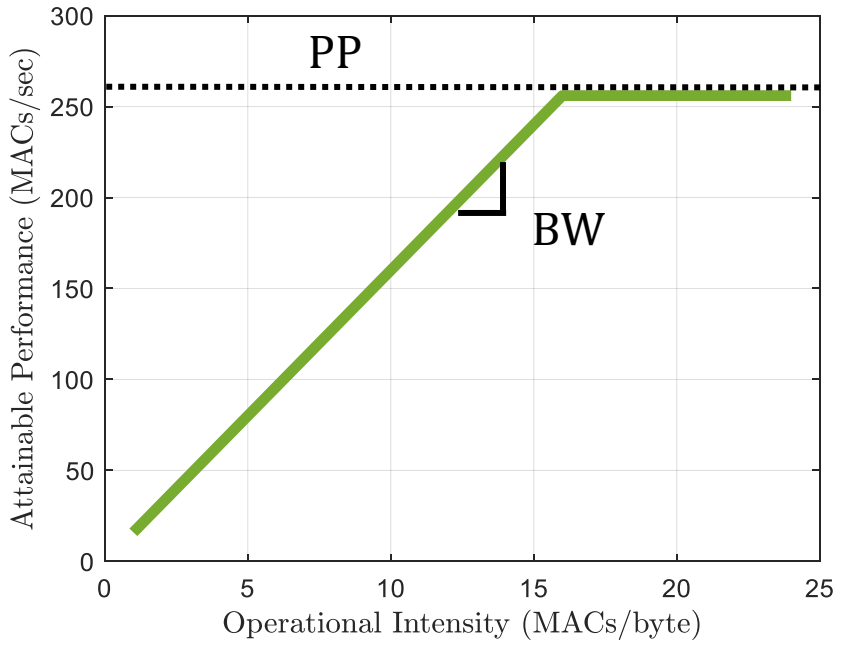
- can these trade-off be captured analytically?
- need relationship between energy, delay, and V_{dd}

- can these trade-off be captured analytically?
- need relationship between energy, delay, and V_{dd}

Energy Models of Memory Access (E_M)

Reducing E_M (memory read energy)

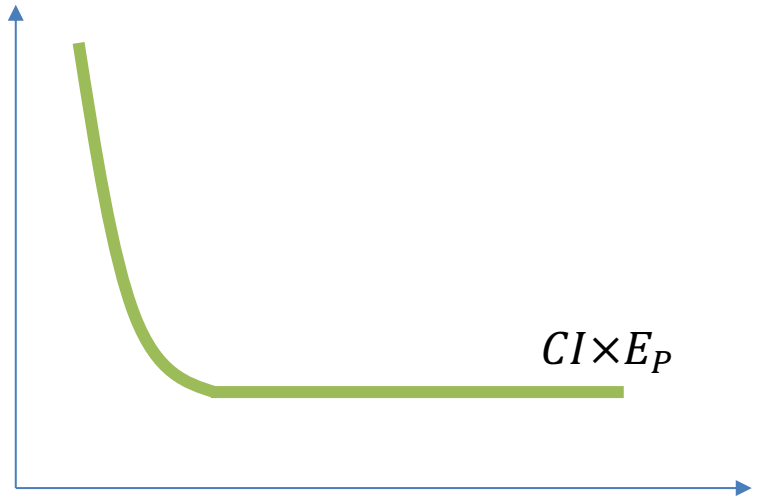
Roofline



Floorline

E_{dec} $(\frac{pJ}{dec})$

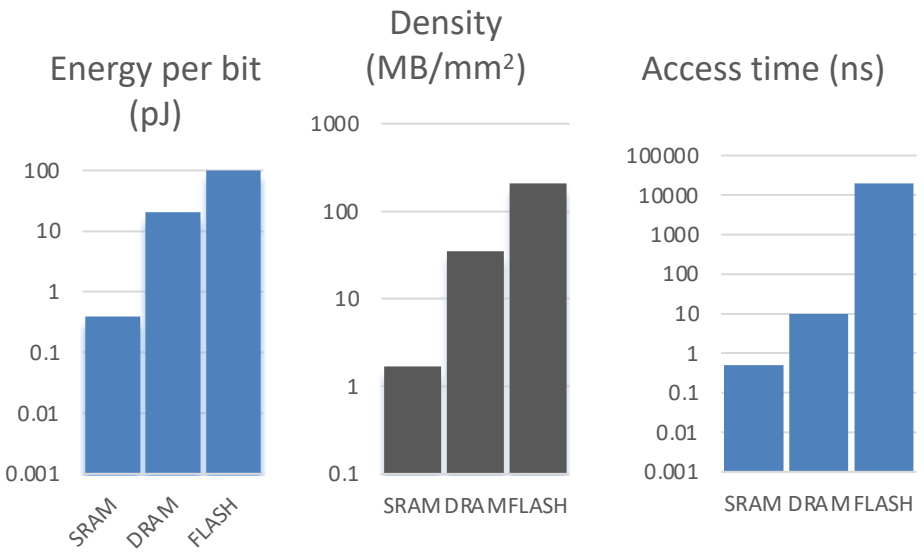
$CI \times E_P \left(1 + \frac{E_M/E_P}{OI}\right)$



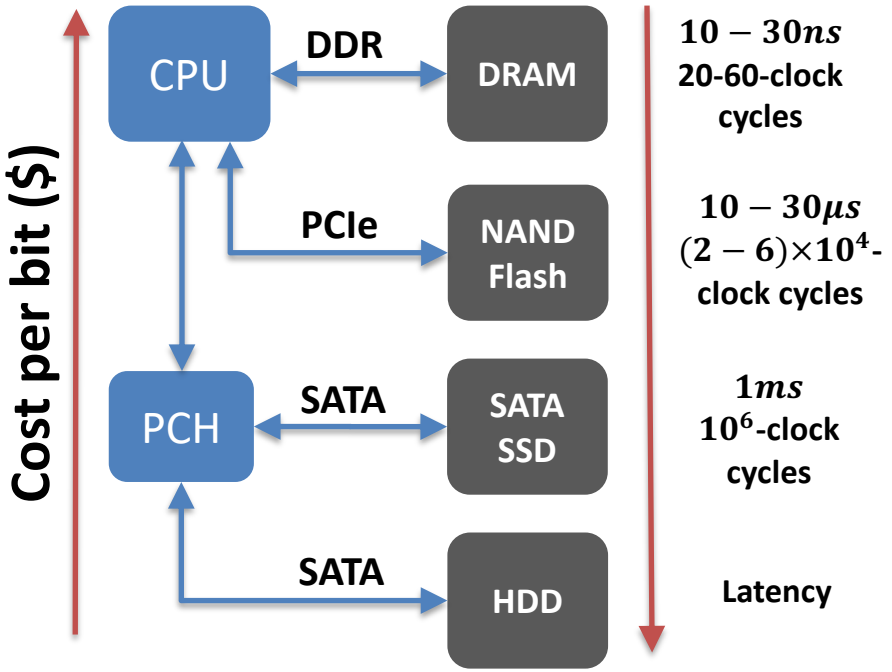
OI $(\frac{ops}{byte})$

- E_M includes array energy + memory-interface (bus) energy

Energy & Delay Cost of Data Access

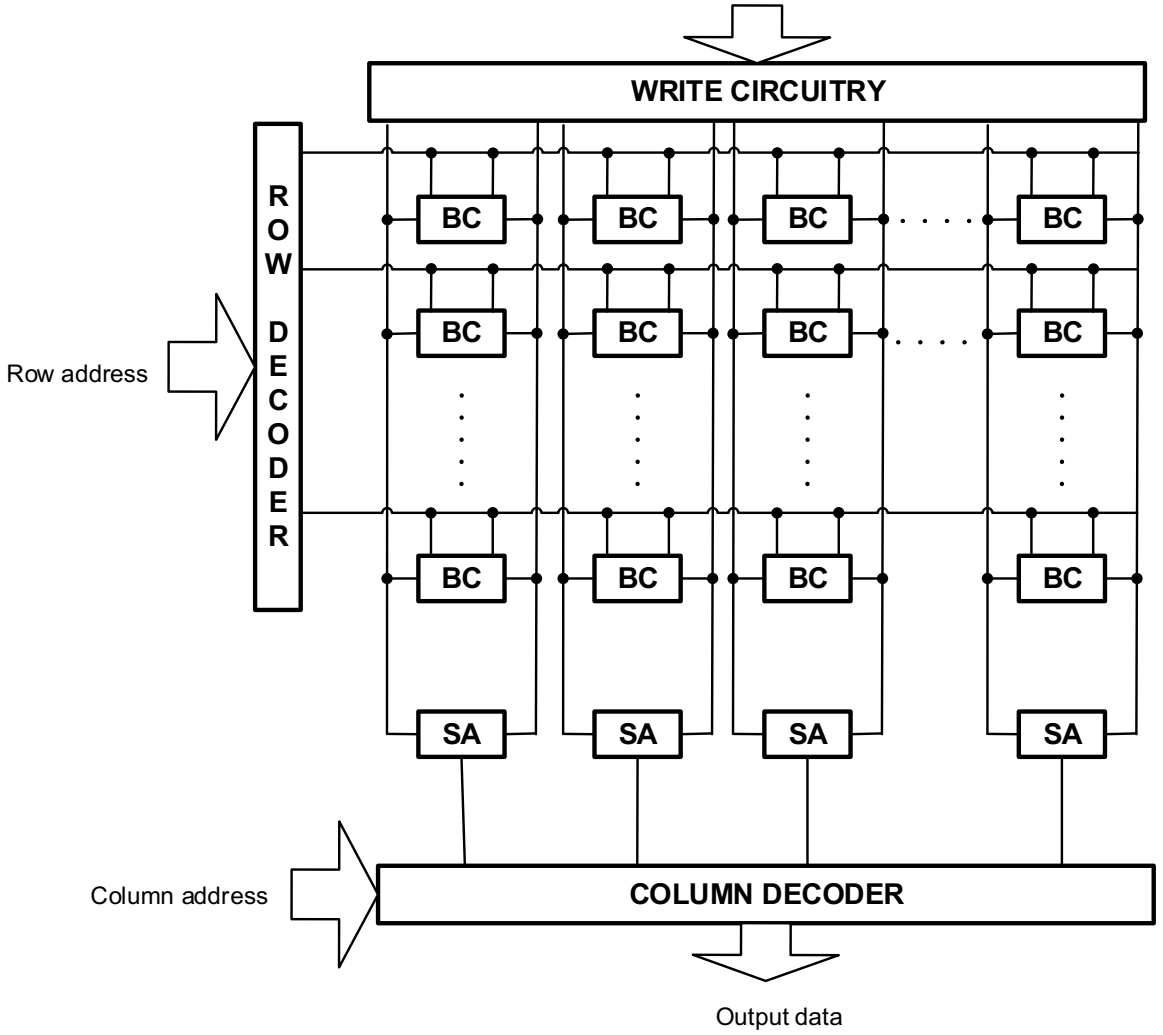


[Yang, J. Joshua, Dmitri B. Strukov, and Duncan R. Stewart. "Memristive devices for computing." *Nature nanotechnology* 8.1 (2013): 13-24.]



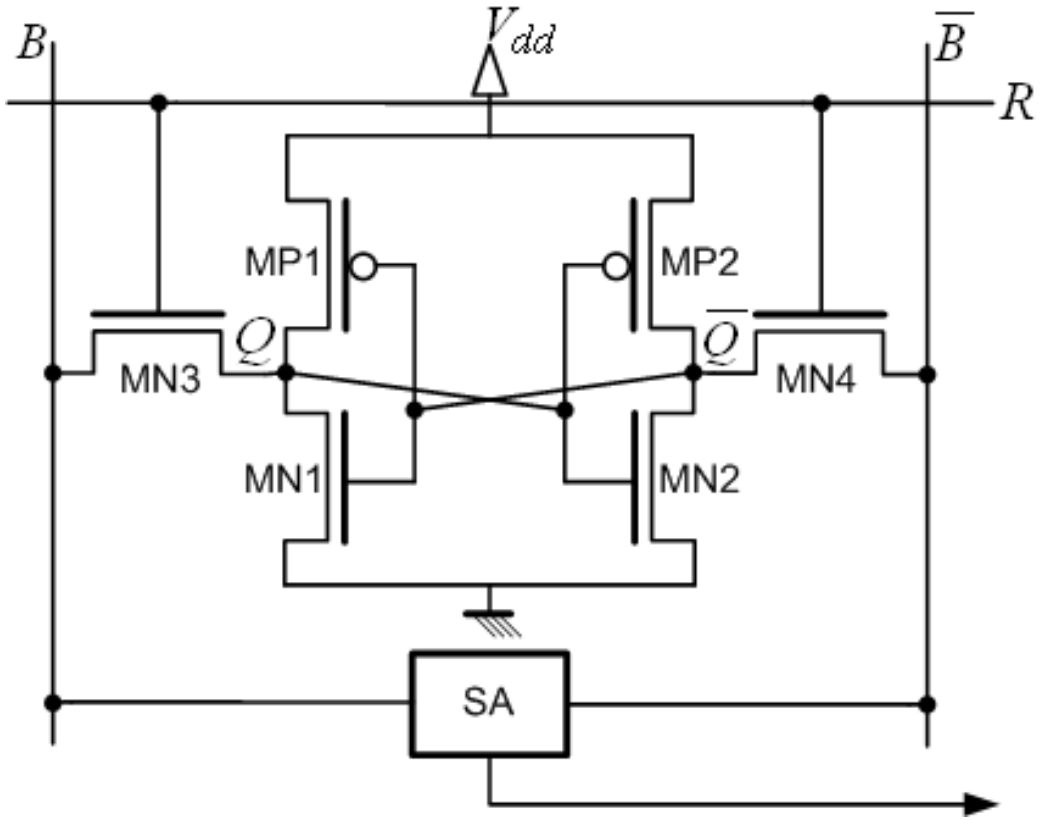
- large data volumes & large data access energy cost
- processor-memory bandwidth limitations

Static Random Access Memory (SRAM)



- used when large storage capacity is needed near processor
- slower than registers but energy efficient/bit
- column muxing (4:1, 8:1, 16:1) reduces bandwidth further

SRAM Bit-Cell (BC)



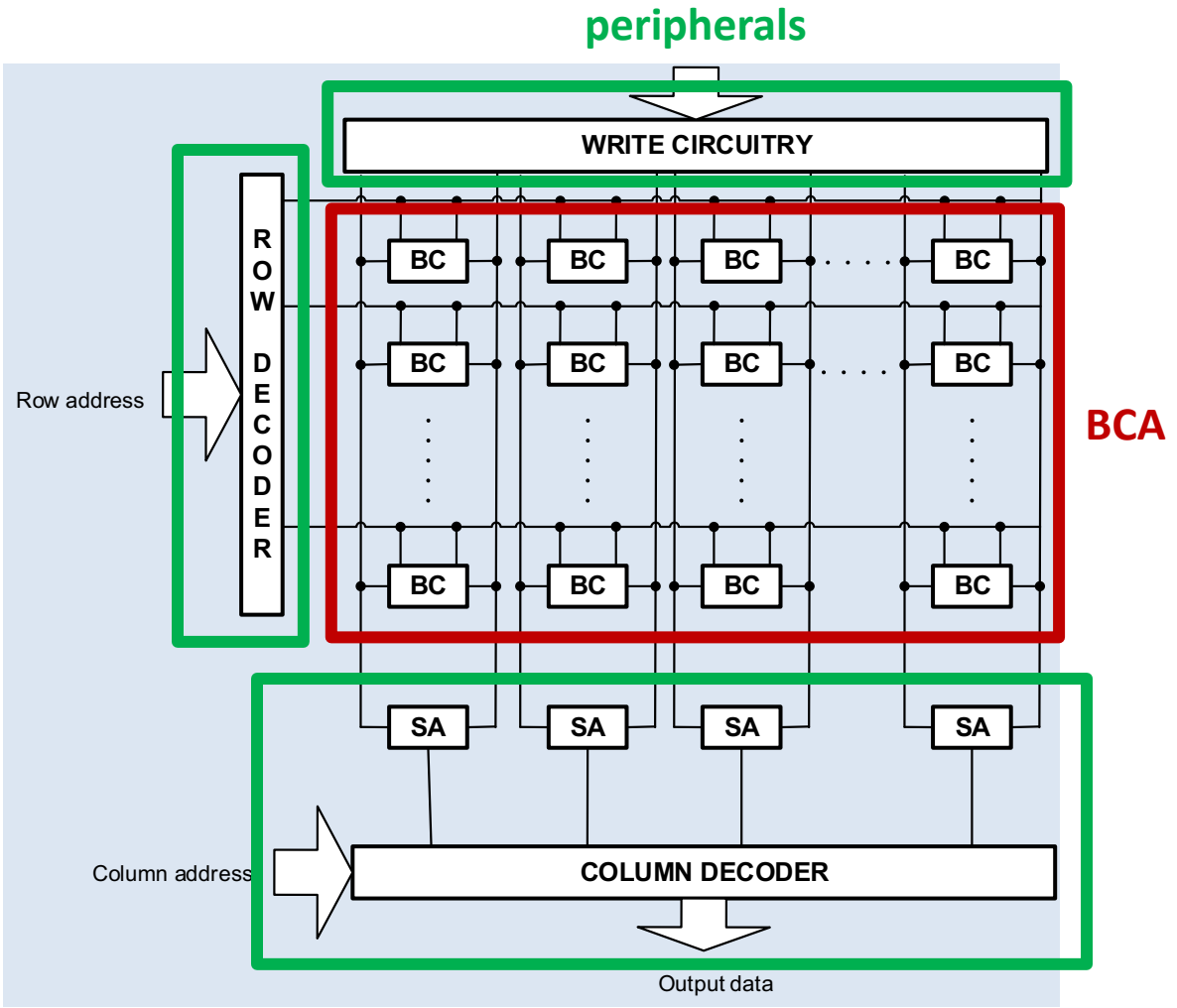
- cross-coupled inverter with access switches (MN3,MN4)
- small swings on bit-lines (B and $\bar{B}$)
- sense-amplifier (SA) resolves voltage drop

Energy Consumption in SRAM

- The total energy consumption:

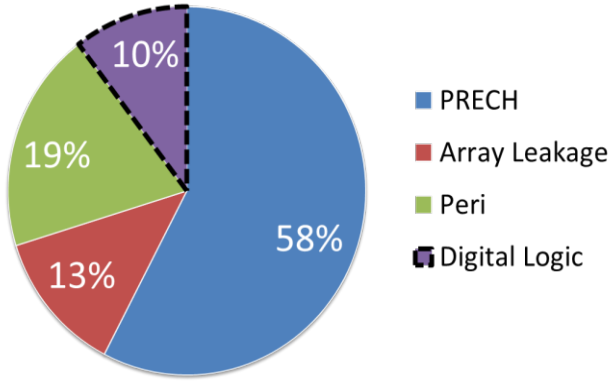
$$E_M = E_{BCA} + E_{peri} + E_{bus}$$
$$E_{BCA} = E_{PRE} + E_{LKG}$$

- E_{BCA}: bitcell array (BCA) energy
- E_{PRE}: BCA precharge energy
- E_{LKG}: BCA leakage energy
- E_{peri}: peripheral circuitry energy
- E_{bus}: bus energy (treated separately)

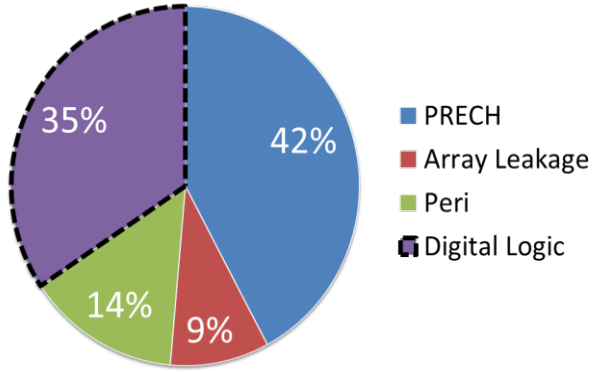


Where is Energy Consumed in SRAM?

SRAM Read Energy
+
Sum of Absolute Difference*



SRAM Read Energy
+
Cross Correlation*



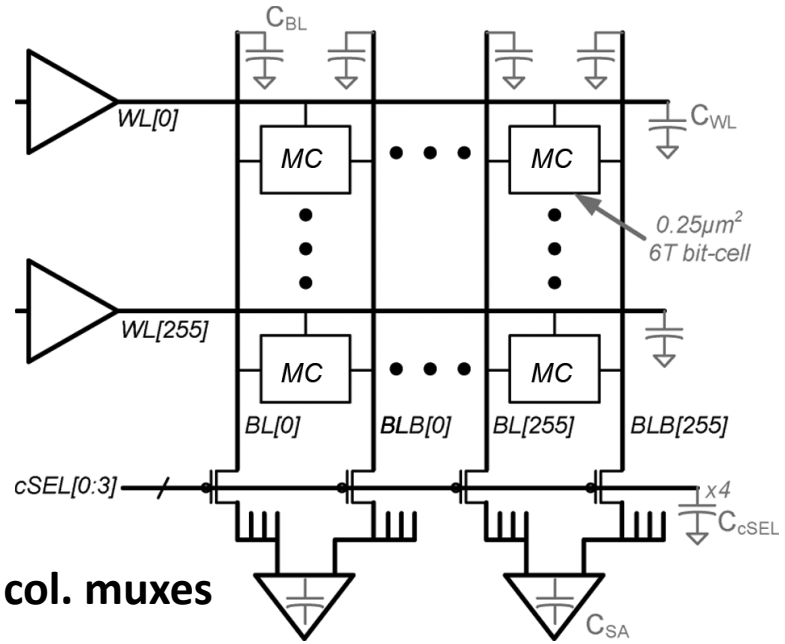
***Post-layout simulation in 65nm CMOS**

- E_{BCA} (E_{PRE}) dominates over E_{peri} in active mode (READ/WRITE)
- E_{LKG} dominates in standby, i.e., when memory is not being used but powered-up

SRAM Energy Model

[Verma, TVLSI, 2010]

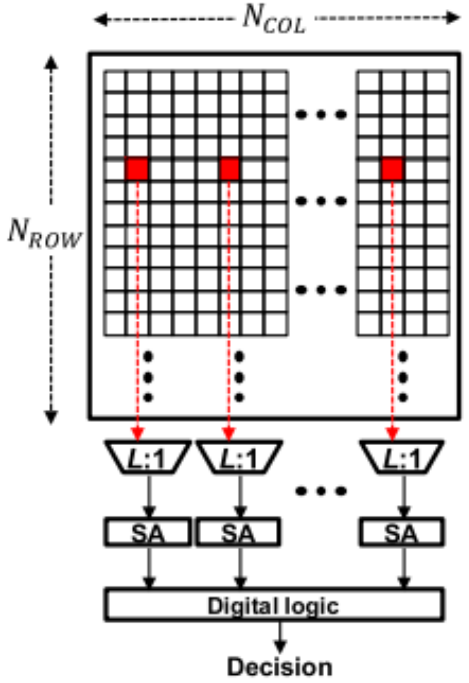
$N \times M$ SRAM array



L: 1 col. muxes

[Kang, TCAS'2020]

$N_{ROW} \times N_{COL}$ SRAM array



$$E_M = E_{BCA} + E_{peri}$$

$$E_{BCA} = E_{PRE} + E_{LKG}$$

$E_{BCA}\ \alpha\ NM$ (BCA size)

- $E_{READ} = (C_{WL} + C_{cSEL})V_{dd}^2 + MNC_{BLC}V_{dd}\Delta V_{BL}(= E_{PRE}) + \left(\frac{M}{L}\right)C_{SA}V_{dd}^2$
- $E_{WRITE} = (C_{WL} + C_{cSEL})V_{dd}^2 + [\left(\frac{M}{L}\right)NC_{BLC}V_{dd}^2 + \frac{M(L-1)}{L}NC_{BLC}V_{dd}\Delta V_{BL}](= E_{PRE})$
- $E_{LKG} = NMI_{LKG-BC}V_{dd}T_{ACC}$

Example

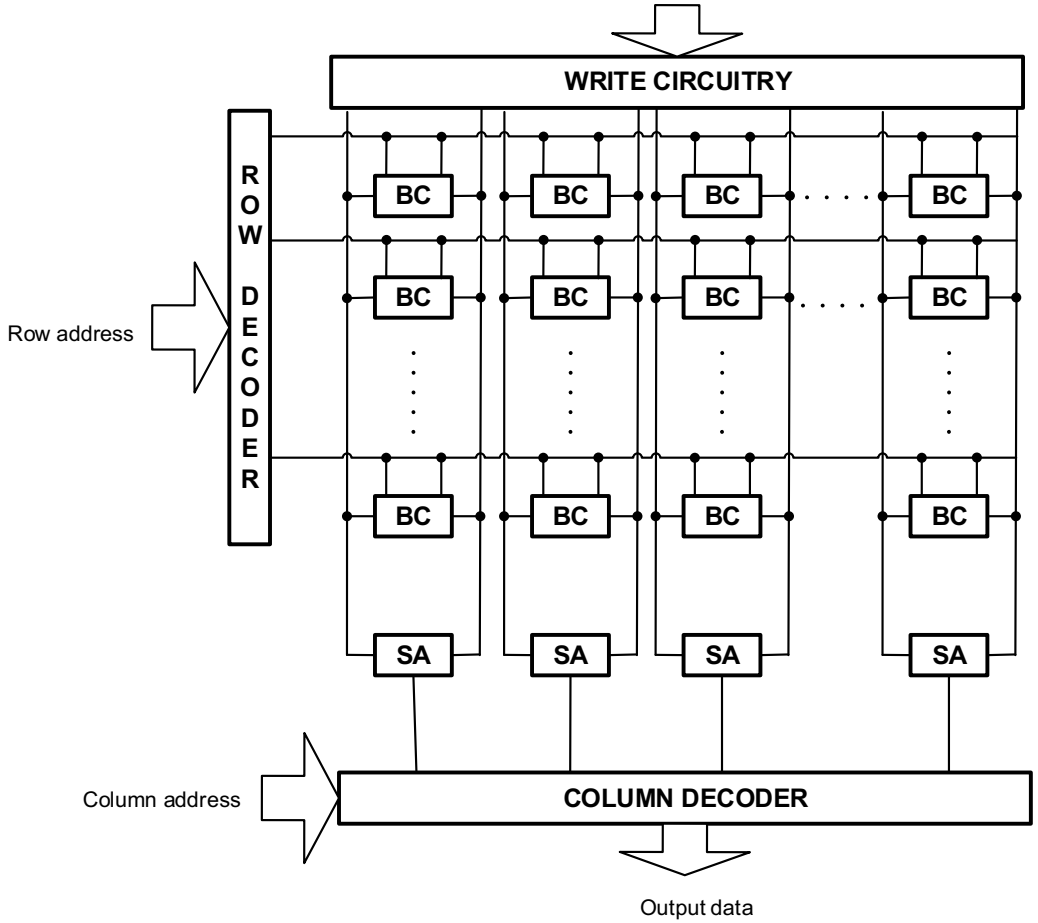
- 65nm CMOS process; $V_{dd} = 1V; V_t = 0.4V$
- $N = 512; M = 256; \Delta V_{BL} = 500mV;$
- $C_{BL} = 512C_{BLC} = 300fF;$

- $E_{READ} = (C_{WL} + C_{cSEL})V_{dd}^2 + MNC_{BLC}V_{dd}\Delta V_{BL}(= E_{PRE}) + \left(\frac{M}{L}\right)C_{SA}V_{dd}^2$
- $E_{PRE} = (256)\times 300\times 10^{-15}\times 1\times 0.4 = 30pJ$

- $E_P(inv) = 0.5\times 0.1\times 10^{-15}\times (1)^2 = 0.05fJ$
- $E_P(8b\ MAC) \approx 64\times 20\times E_P(inv) = 64fJ \rightarrow E_R = \frac{E_M}{E_P} \approx 500\times$

Delay Model

- delay is a complex function
- $T_p = T_{ROW} + T_{BL} + T_{COL}$
- T_{ROW}: delay of row decoder and row drivers
- T_{BL}: bitline discharge time
- T_{COL}: delay of sense amplifiers and column decoder
- $T_{BL} = \frac{C_{BL}\Delta V_{BL}}{k(V_{dd}-V_t)^2}$



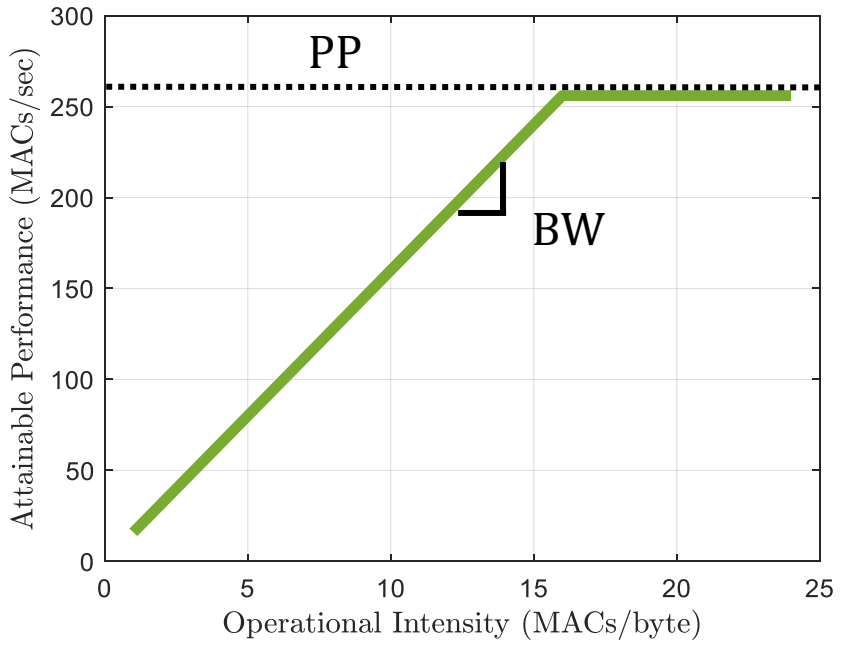
Methods to Reduce E_M

- BCs are already made compact for density reasons → C_L is small
- active mode methods: reduce core (BCA) V_{dd} and ΔV_{BL} → including subthreshold operation
- standby mode methods: power gating, low standby supply voltages, shut down blocks not being used

Energy Models of Bus (E_{bus}) (processor-memory interface)

Reducing E_M & $1/BW$

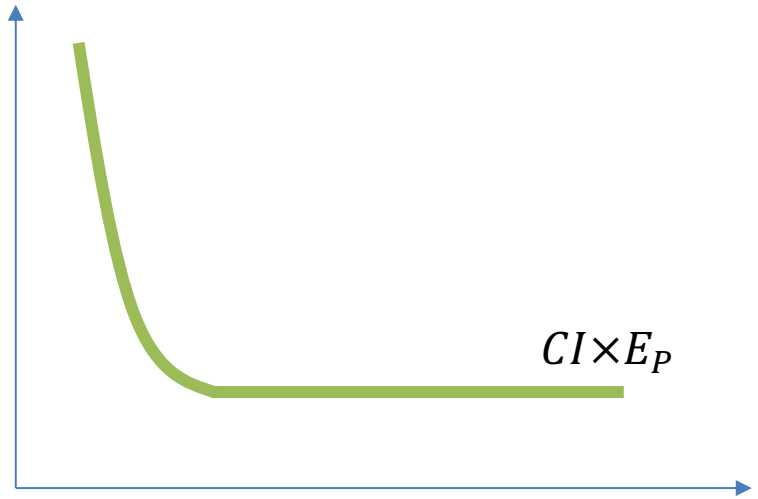
Roofline



Floorline

E_{dec} $(\frac{pJ}{dec})$

$CI \times E_P \left(1 + \frac{E_M/E_P}{OI}\right)$



OI $(\frac{ops}{byte})$

- E_M includes array energy + memory-interface energy
- BW is the memory-interface bandwidth

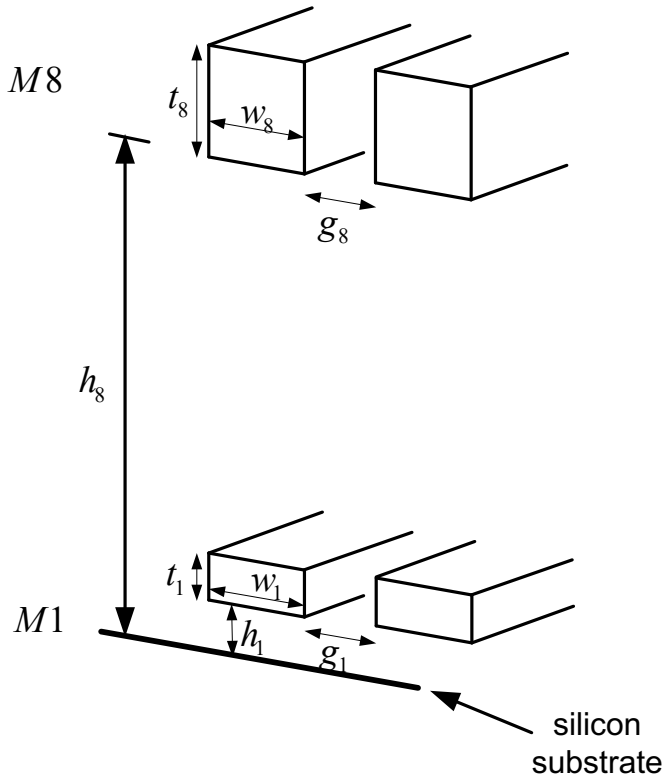
- On-chip interconnect:
  - *local* interconnect for gate-to-gate or within compact circuit macros) level
  - *global* interconnect (busses, clock distribution networks)
- Off-chip interconnect → Chip I/O or serial links
- focus on on-chip interconnect

- global interconnects do not scale with technology node → chip sizes are steady (more functions are packed into the die)

- need to model delay and energy of interconnect

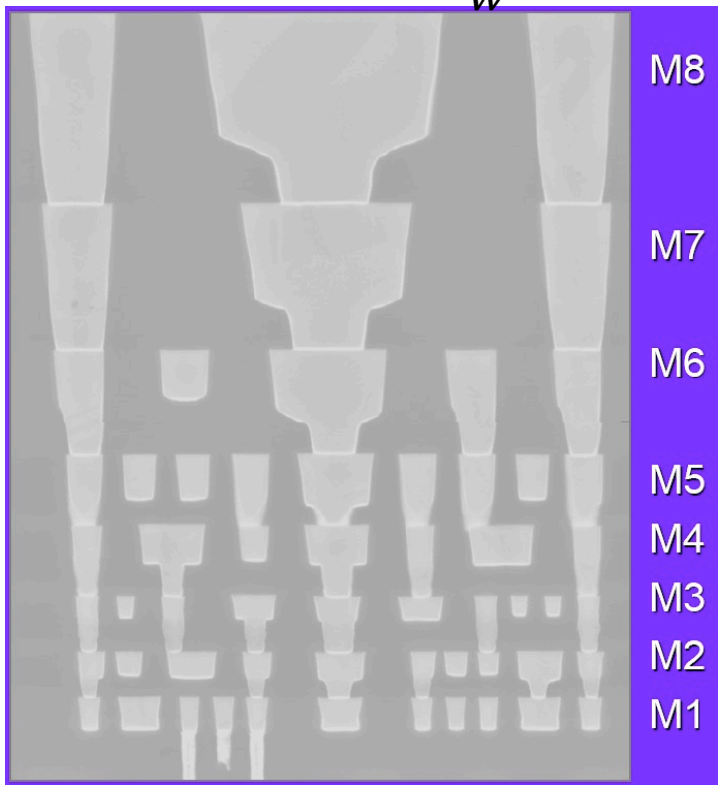
- Overall goal of system design should be to minimize data transfers over long interconnects.

Interconnect Stack



Intel's 65nm node

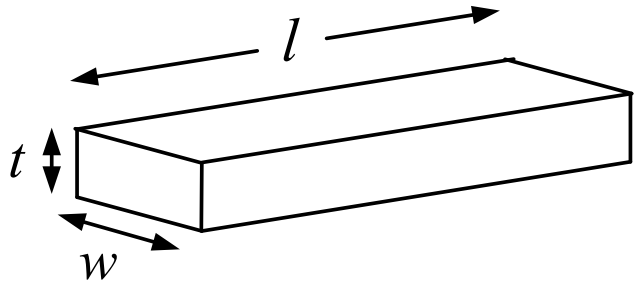
8 metal layers; $\frac{t}{w} = 2$



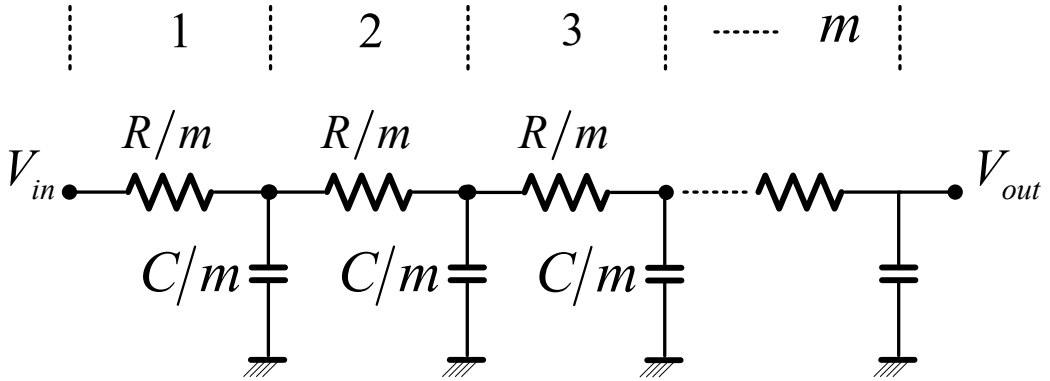
[P. Bai, IEDM 2004]

- Metal (Al/Cu) stack employed as interconnect
- Lower layers (wide & thin), upper layers (thicker)
  - Lower layers used for local connections. Upper layers for global connections

Isolated Wire Delay & Energy Model



$R = rl \qquad C = cl$



- simple RC model for an isolated interconnect
- delay is proportional to length squared (l^2):

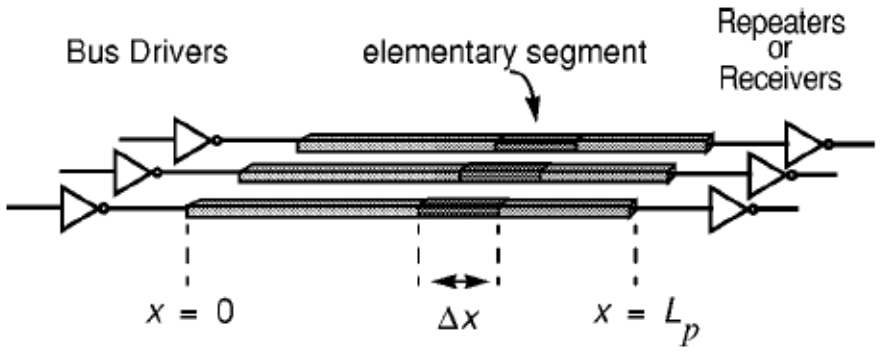
$$T_p = 0.38\tau; \qquad \tau = 0.5RC = 0.5rcl^2$$

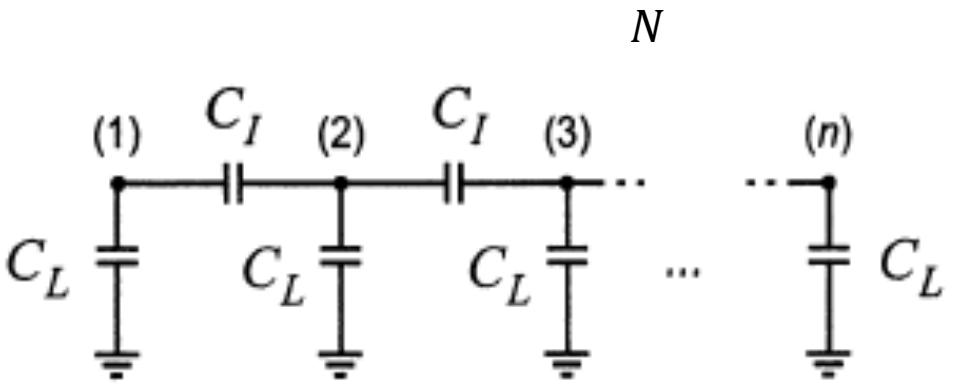
- energy model of an isolated interconnect

$$E = CV_{dd}^2$$

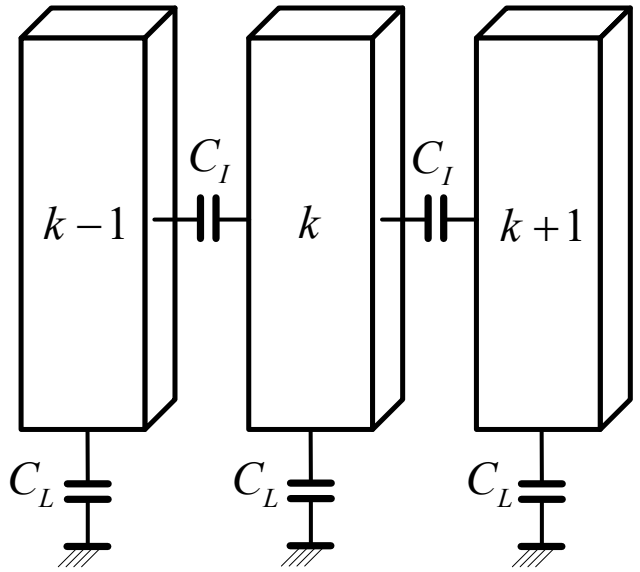
Bus Delay & Energy Models

[Sotiriadis-ASPDAC01]





- multiple wires/lines running in parallel
- capacitance to ground (C_L) and interwire capacitance (C_I)
- C_I makes the delay of a bus data-dependent



Wire # and transitions			Delay (x T_{p0})
k-1	*k*	*k*+1	
↑	↑	↑	1
-	↓	↓	1+λ
↑	↑	↓	1+2λ
-	↑	↓	1+3λ
↓	↑	↓	1+4λ

Max delay case: ↑↓↑ and ↓↑↓

Min delay case: ↑↑↑ and ↓↓↓

- $T_{p0} = 0.38\tau$ = delay of an isolated wire.
- $T_k = (1 + p\lambda)T_{p0}$: delay of the k^{th} wire (T_k) depends upon the data transitions on the $(k-1)^{th}$ and $(k+1)^{th}$ wire (Miller effect)
- $\lambda = \frac{C_I}{C_L}$ is the **coupling ratio** which varies from 3 to 6 in 180nm CMOS.

Bus Delay Model

$$\boldsymbol{T}(\boldsymbol{\Delta}) = \begin{cases} (1+\lambda)\Delta_1^2 - \lambda\Delta_1\Delta_2, \ldots\ldots k = 1 \\ (1+2\lambda)\Delta_k^2 - \lambda\Delta_k(\Delta_{k-1} + \Delta_{k+1}), \ldots . 1 < k < N \\ (1+\lambda)\Delta_N^2 - \lambda\Delta_N\Delta_{N-1}, \ldots\ldots k = N \end{cases}$$

transitions on the k^{th} wire in the bus

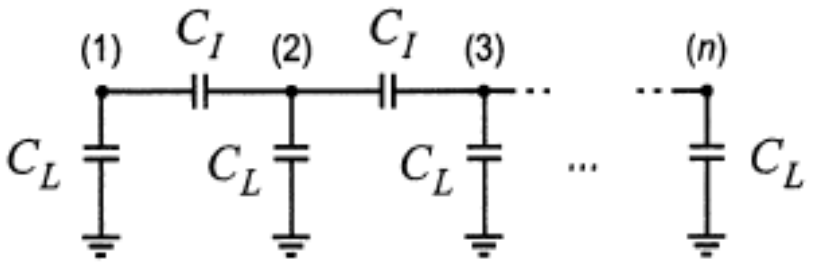
$$\Delta_k = 1\ (0 \to 1)$$

$$\Delta_k = -1\ (1 \to 0)$$

$$\Delta_k = 0\ (1 \to 1 \text{ or } 0 \to 0).$$

Bus Energy Model

[Sotiriadis-TCAS03]



$$\bar{C} = \begin{bmatrix} 1+\lambda & -\lambda & 0 & \cdots & 0 & 0 \\ -\lambda & 1+2\lambda & -\lambda & \cdots & 0 & 0 \\ 0 & -\lambda & 1+2\lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdot & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1+2\lambda & -\lambda \\ 0 & 0 & 0 & \cdots & -\lambda & 1+\lambda \end{bmatrix} \cdot C_L$$

- for input transition from $\boldsymbol{u}^i \rightarrow \boldsymbol{u}^f$:

$$E = \frac{C_L V_{dd}^2}{2} \left(\boldsymbol{u}^f - \boldsymbol{u}^i\right)^T \boldsymbol{C} \left(\boldsymbol{u}^f - \boldsymbol{u}^i\right) = \frac{C_L V_{dd}^2}{2} \boldsymbol{\Delta}^T \boldsymbol{C} \boldsymbol{\Delta}$$

Example – N=2

TABLE I
ENERGY DRAWN FROM V_{dd} BY A TWO-LINE BUS MODELED AS IN FIG. 4

Trans. Energy / $C_L \cdot V_{dd}^2$		$[V_1^f, V_2^f]$			
		00	01	10	11
$[V_1^i, V_2^i]$	00	0	$1+\lambda$	$1+\lambda$	2
	01	0	0	$1+2\lambda$	1
	10	0	$1+2\lambda$	0	1
	11	0	λ	λ	0

- bus energy consumption is data dependent much like logic
- possibility of using coding techniques to reduce bus energy consumption

Summary

- energy and delay models of on-chip computation, storage, and interconnect help us evaluate energy efficiency and throughput of architectural alternatives

References

- [Ho-IEEEProc07] R. Ho, K. Mai, and M. Horowitz, "The future of wires", *IEEE Proceedings*, vol. 89, no. 4, pp. 490-504, April 2001.
- [Sotiriadis-ASPDAC01] P. P. Sotiriadis and A. P. Chandrakasan, "Reducing bus delay in submicron technology using coding," ASP-DAC 2001.
- [Sotiriadis-TCAS03] P. P. Sotiriadis and A. P. Chandrakasan, "Bus energy reduction by transition pattern coding using a detailed deep submicrometer bus model," IEEE Transactions on Circuits and Systems – I, vol. 50, no. 10, pp. 1280-1295, October 2003.
- [Itoh01] K. Itoh, *VLSI Memory Chip Design*, Springer-Verlag, 2001.
- [Rabaey-03] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits – A Design Perspective*, 2nd Edition, Prentice Hall, 2003.
- N. Verma, "Analysis Towards Minimization of Total SRAM Energy Over Active and Idle Operating Modes," *IEEE Transactions on VLSI*, 2010.
- Y.-H. Chen et al., "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid-State Circuits, Jan. 2017.

Course Web Page

https://courses.grainger.illinois.edu/ece598nsg/fa2020/
https://courses.grainger.illinois.edu/ece498nsu/fa2020/

http://shanbhag.ece.uiuc.edu