Energy Autonomous Wireless Systems (EAWS)

Prof. Catherine Dehollain
EPFL-IEL RFIC
Prof. Anja Skrivernik
EPFL-TCL LEMA
Prof. Franco Maloberti
Guest Lecturer, Univ. of Pavia
Prof. Andreas Burg
EPFL-IEL TCL

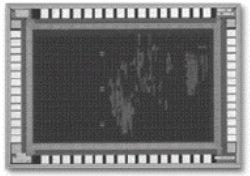
LESSON 2 – Digital Low Power VLSI and System Design

Prof. Andreas Burg

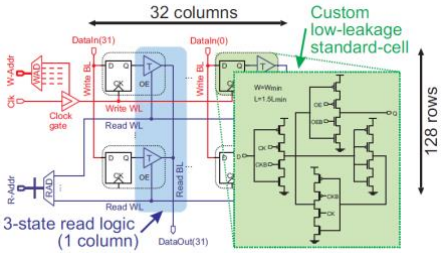
EPFL-STI-IEL-TCL

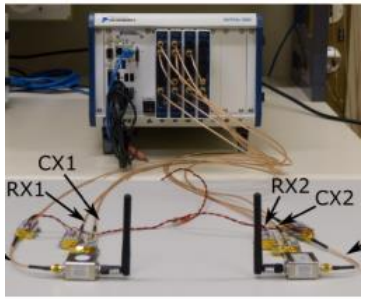
Andreas Burg

- 1994-2000 – Diploma in Electrical Engineering at *ETH Zurich*
- 2000-2006 – PhD. Studies in Electrical Engineering, Integrated Systems Laboratory & Communication Theory Group, ETH Zurich
- 2007 – Co-Founder Celestrius AG
- 2007- 2008 – Director for VLSI at Celestrius AG & PostDoc at ETH Zurich
- 2009-2010 – SNF Assistant Professor at ETH Zurich, Signal Processing Circuits and Systems (SPCaS) group
- 2011 - present –Professor at EPFL, Telecommunications Circuits Laboratory (TCL)
  - **Low-power digital VLSI signal processing**
  - **Low-power embedded systems**
  - **Low-power embedded memories**
  - **Reliability issues in nanometer CMOS**
  - **Signal processing for wireless communication**
  - **Prototyping and demonstration of communication systems**



Turbo decoder
55% memory





Course contents

Objectives :

- To provide a basic introduction to power consumption in digital VLSI circuits

- To give an overview of the key techniques for designing low power digital VLSI circuits

- To raise awareness and introduce potential solutions for the difficulties associated with ultra-low-power design in advanced technology nodes

- Provide an example of a low-power system design (ECG monitoring)

- Get acquianted with the anatomy of a basic low-power embedded system based on commercial off the shelf (COTS) components

- Lear about the various system components and modes to be able to make design decisions when using or selecting an embedded platform

- Collect some initial experience in using an embedded low-power system and its low-power modes

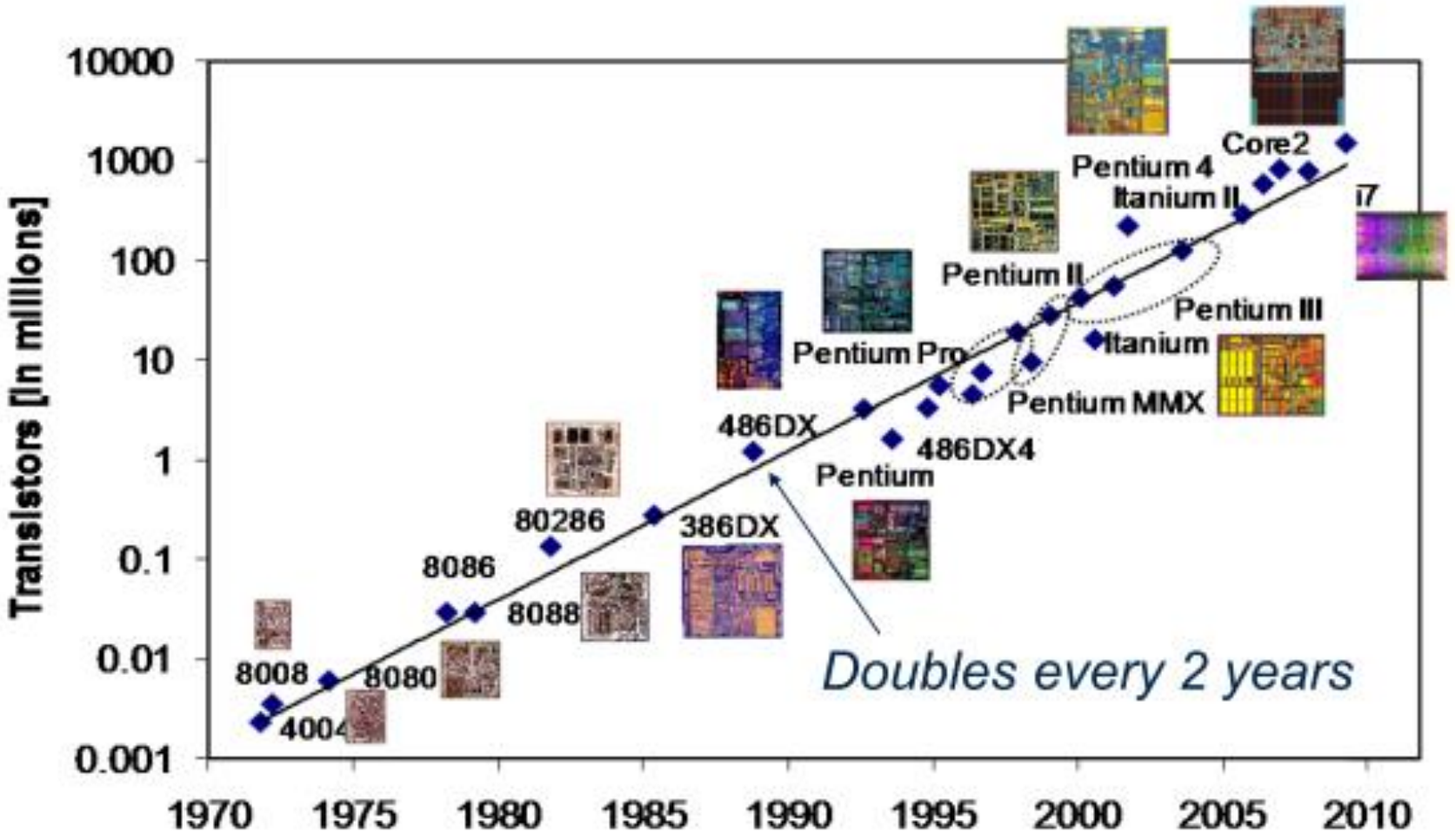
LESSON 2a – Low Power Digital VLSI Design

Prof. Andreas Burg

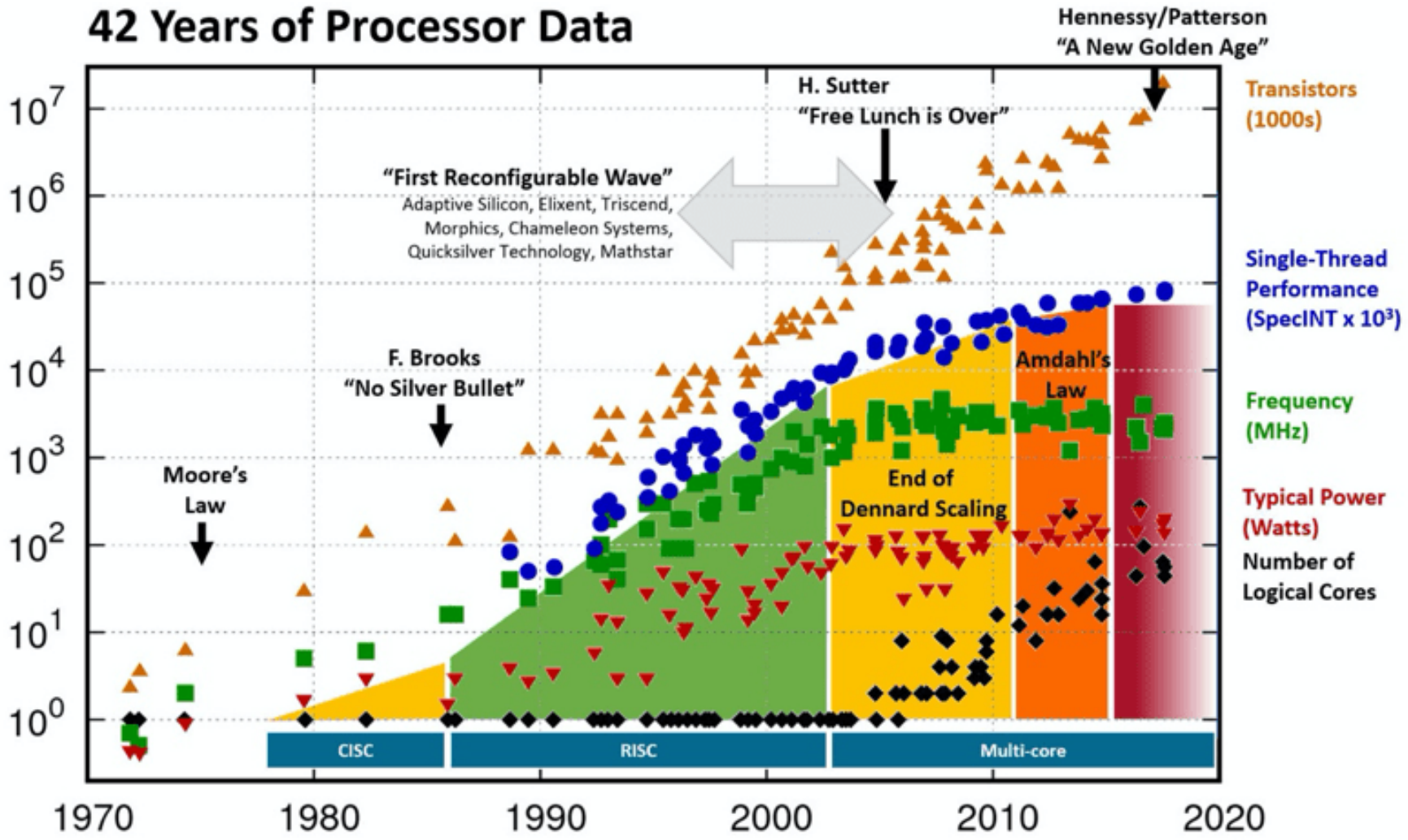
TOC: Low Power Digital VLSI Design

- CMOS Basics
- Active Power Reduction on Register Transfer Level
- Voltage Scaling and Sub-VT Design
- Leakage Power Reduction
- Low Power Memories
- Variation Aware Design
- Low Power and Variation Aware Design with Approximate Computing

Moore's Law

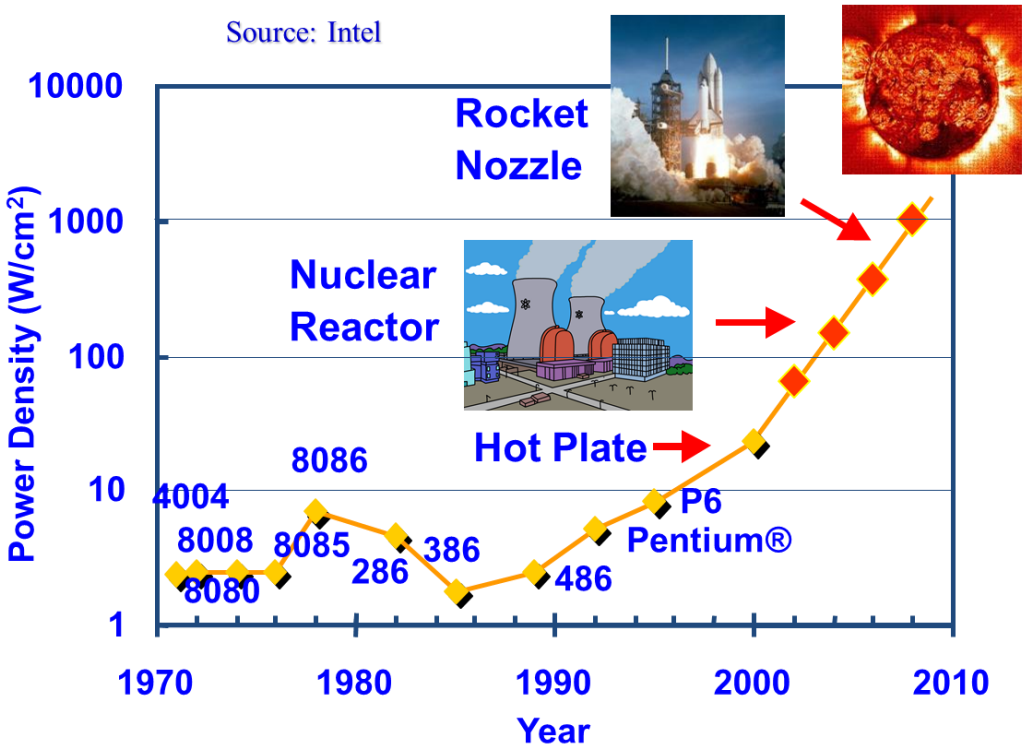

Transistors [In millions]
10000
1000
100
10
1
0.1
0.01
0.001
1970
1975
1980
1985
1990
1995
2000
2005
2010
8008
4004
8080
8086
8088
80286
386DX
486DX
Pentium
486DX4
Pentium Pro
Pentium II
Pentium MMX
Pentium III
Itanium
Itanium II
Pentium 4
Core2
i7
Doubles every 2 years

Frequency Trends



Hennessy and Patterson, Turing Lecture 2018, overlaid over "42 Years of Processors Data"
https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/;"First Wave" added by Les Wilson, Frank Schirrmeister
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Thermal Issues



Package cost
- **4-5W for cheap packages**
- **100 W/cm² for air cooling**
- **7.5kW/rack**

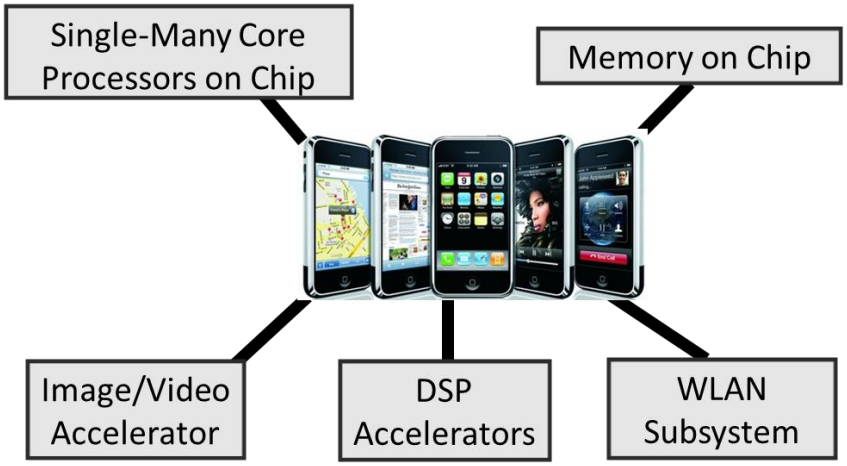
Power delivery
- **> 1000 pins for power delivery on a 100W processor**

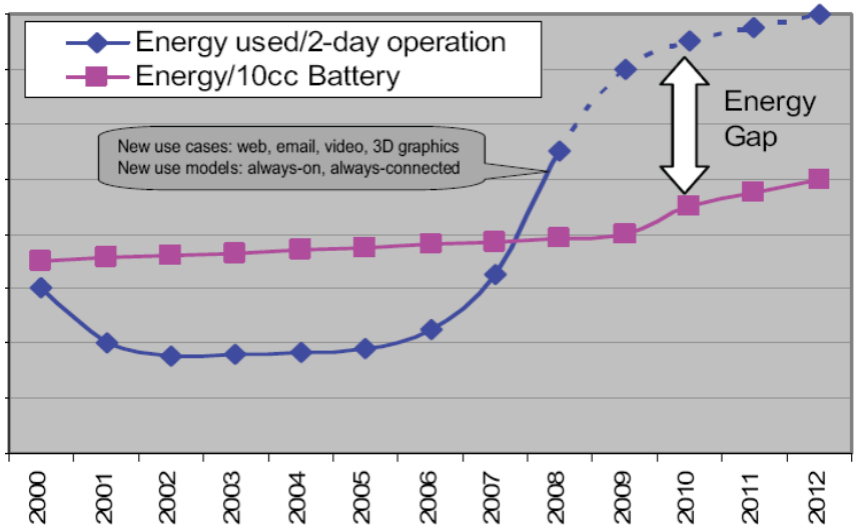
Performance penalty
- **25 -> 100 deg. C : 30%**

- **Thermal Design Power:** upper limit on power consumption
  - **Microprocessors for servers: ~30-100 W/cm²**
  - **Mobile devices: ~3W total (handheld)**

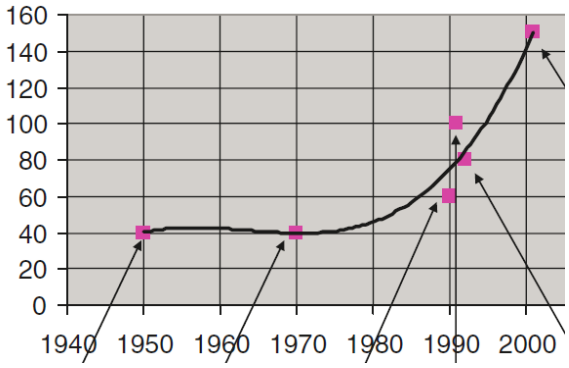
Power Consumption Bottleneck

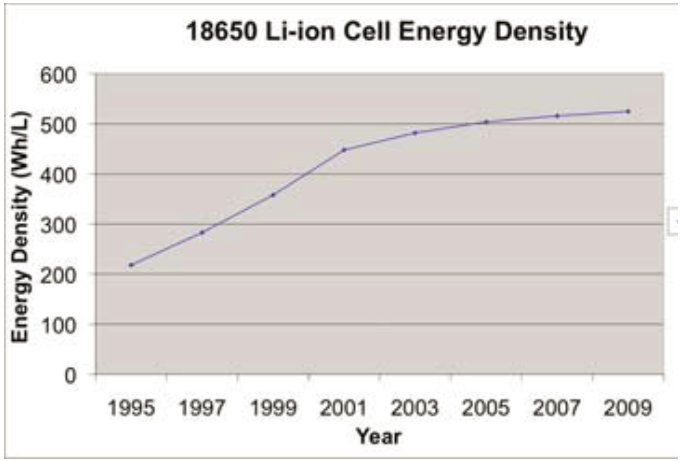
- **Mobile devices: energy-efficiency**





- **Battery capacity grows only very slowly**
  - **Boost in the 1990s due to Mobile Phone introduction**
  - **Capacity growth stalled since 2000 at the limit of Li-ion**
  - **Only 3%-7% annual improvement**

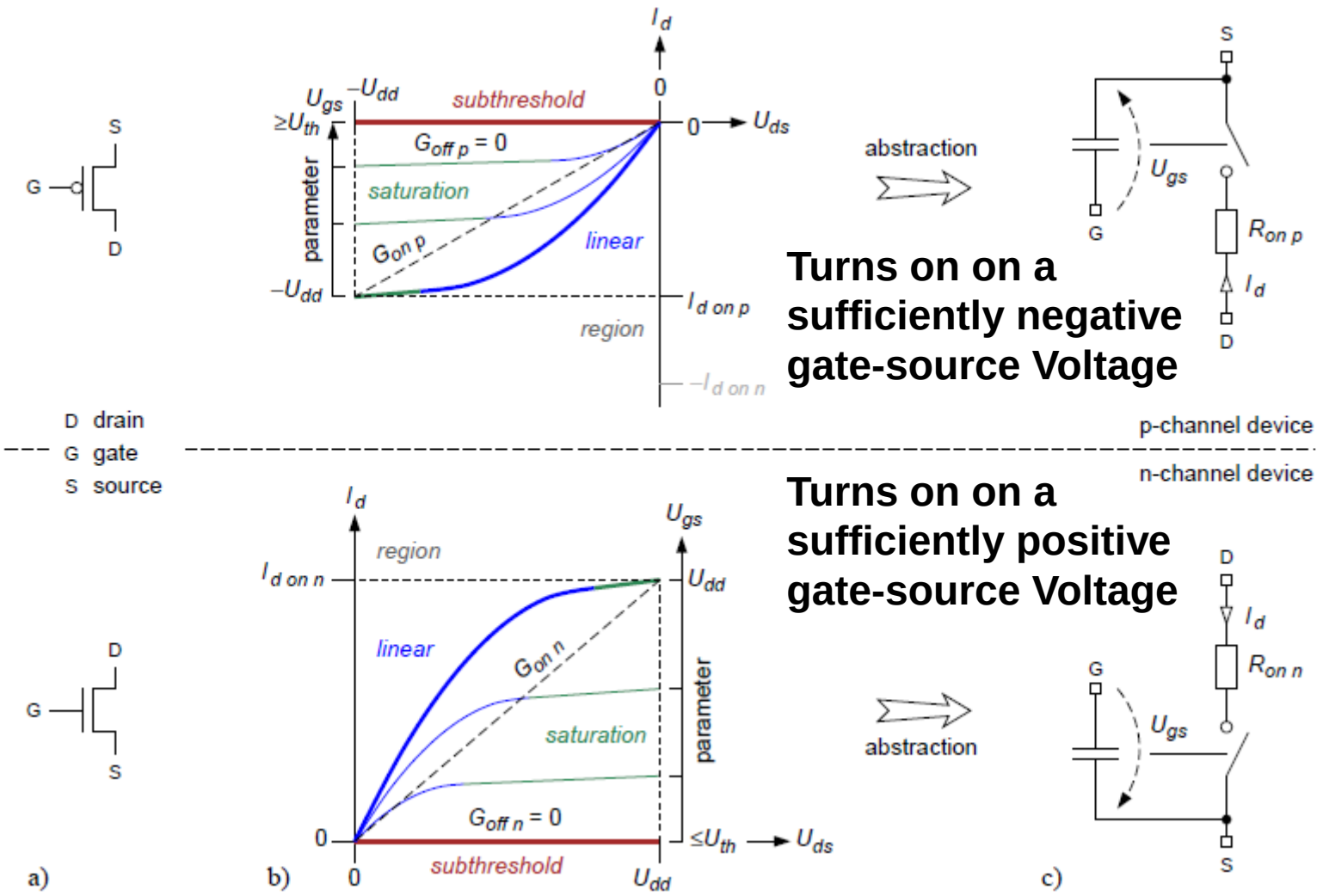




Low-Power Design Overview

	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Reduced V_{dd} Sizing Multi-V_{dd}	Clock Gating	DFS, DVS (Dynamic Freq, Voltage Scaling)
Leakage	+ Multi-V_T	Sleep Transistors Multi-V_{dd} Variable V_T	+ Variable V_T

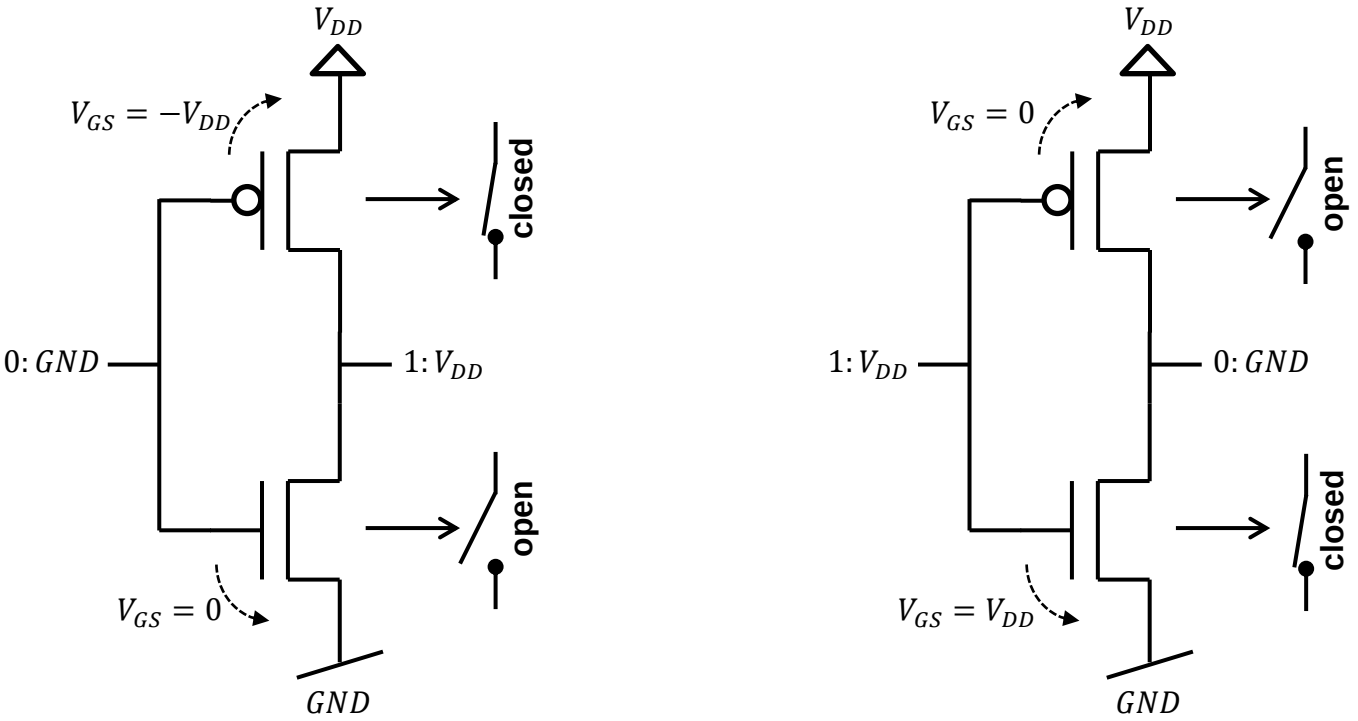
CMOS Basics


S
G
D
I_d
U_{gs}
$-U_{dd}$
subthreshold
$\geq U_{th}$
$G_{off\,p} = 0$
saturation
$G_{on\,p}$
linear
parameter
$-U_{dd}$
0
U_{ds}
$I_{d\,on\,p}$
region
$-I_{d\,on\,n}$
abstraction
U_{gs}
$R_{on\,p}$
I_d
D drain
G gate
S source
p-channel device
n-channel device
Turns on on a sufficiently negative gate-source Voltage
Turns on on a sufficiently positive gate-source Voltage
$I_{d\,on\,n}$
U_{dd}
$G_{on\,n}$
$G_{off\,n} = 0$
$\leq U_{th}$
a)
b)
c)

Basic Inverter in Steady State (ideal)

- PMOS performs pull-up to V_{DD}
- NMOS performs pull-down to GND

- **Complementary gate:** output connected to either V_{DD} or GND
  - Static (steady state)



  - Ideally, no current path from V_{DD} to GND
  - Ideally, no static power consumption

CMOS Transistors: Operating Regimes (NMOS)

- **Three different operating regions**
  - **Sub-threshold region: almost OFF**

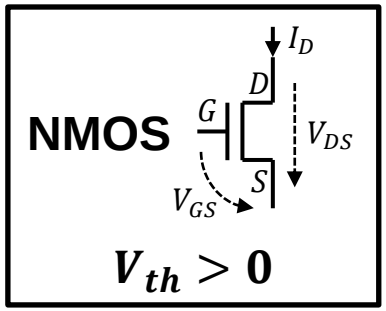
$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}} \quad \text{if} \quad V_{GS} < V_{th}$$

  - **Linear region: resistive behavior**

$$I_{DS} = \beta\left((V_{GS}-V_{th})V_{DS} - \frac{V_{DS}^2}{2}\right) \quad \text{if} \quad \begin{array}{c} V_{GS} > V_{th} \\ V_{DS} < V_{GS}-V_{th} \end{array}$$

  - **Saturation region: voltage controlled current source**

$$I_{DS} = \frac{\beta}{2}(V_{GS}-V_{th})^{\alpha} \quad \text{if} \quad \begin{array}{c} V_{GS} > V_{th} \\ V_{DS} > V_{GS}-V_{th} \end{array}$$

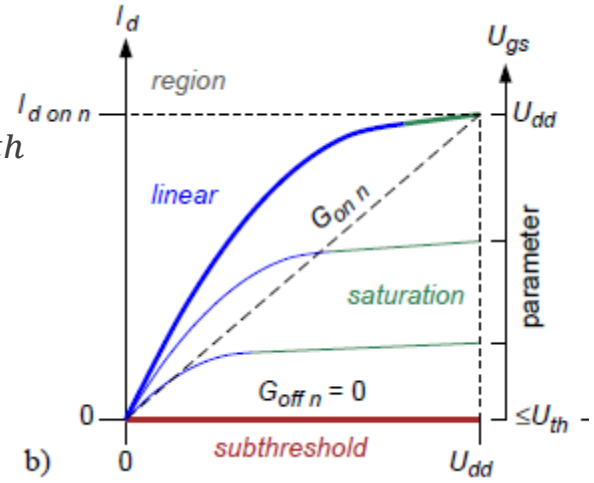


$v_t = kT/q$

β: gain factor

- Velocity saturation in new technologies:

L [nm]	2000	1000	500	250	130	65 …
$\alpha \approx$	1.6 … 1.65	1.45 … 1.6	1.3 … 1.5	1.1 … 1.4	1.0 … 1.25	1.0 … 1.1



CMOS Transistors: Operating Regimes (PMOS)

- **Three different operating regions**
  - **Sub-threshold region: almost OFF**

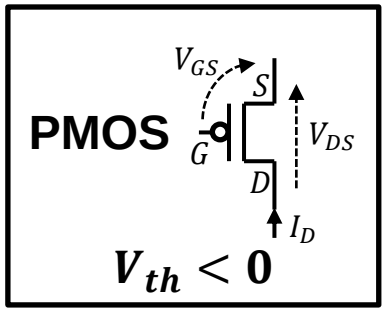
$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}} \quad \text{if} \quad V_{GS} > V_{th}$$

  - **Linear region: resistive behavior**

$$I_{DS} = \beta\left((V_{GS}-V_{th})V_{DS} - \frac{V_{DS}^2}{2}\right) \quad \text{if} \quad \begin{array}{c} V_{GS} < V_{th} \\ V_{DS} > V_{GS} - V_{th} \end{array}$$

  - **Saturation region: voltage controlled current source**

$$I_{DS} = \frac{\beta}{2}(V_{GS}-V_{th})^{\alpha} \quad \text{if} \quad \begin{array}{c} V_{GS} < V_{th} \\ V_{DS} < V_{GS} - V_{th} \end{array}$$

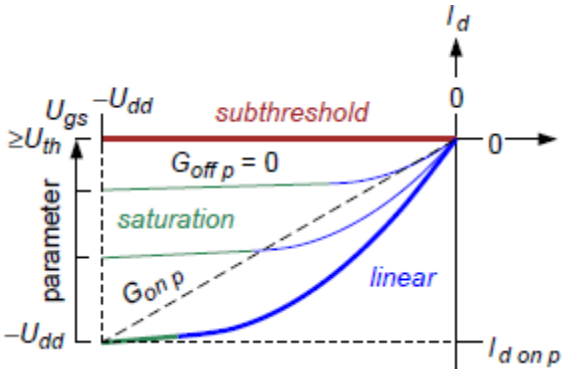


$v_t = kT/q$

β: gain factor

- Velocity saturation in new technologies:

L [nm]	2000	1000	500	250	130	65 …
$\alpha \approx$	1.6 … 1.65	1.45 … 1.6	1.3 … 1.5	1.1 … 1.4	1.0 … 1.25	1.0 … 1.1



CMOS Transistors: Parameters

The gain factor β is a function of **process parameters** and **layout geomerty**

$$\beta = \frac{\mu \varepsilon_{OX}}{t_{OX}} \frac{W}{L}$$

where

t_{OX}	gate dielectric thikness
ε_{OX}	gate dielectric permitivity
μ	effective carrier mobility in inversion layer
W	channel (gate) width
L	channel (gate) length

Design parameters

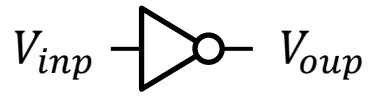
- **Designer sets the drive-strength by controlling width and length of the transistor**

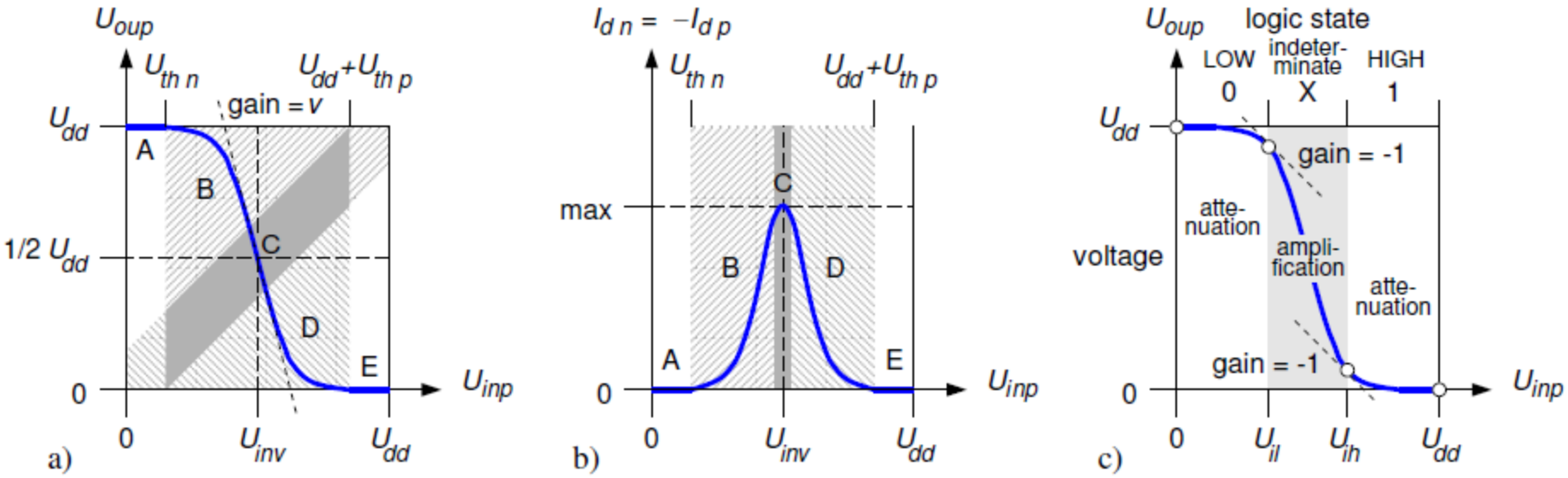
CMOS Inverter: In-Out Transfer Characteristic (Static)

Inverter as non-linear amplifier with a large, but finite gain in the transition region

range	applies when	n-channel ▼	p-channel ▲
A	$0 \le U_{inp} \le U_{th\,n}$	subthreshold	linear
B	$U_{th\,n} < U_{inp} < U_{inv}$	saturation	linear
C	$U_{inp} \approx U_{inv}$	saturation	saturation
D	$U_{inv} < U_{inp} < U_{dd} + U_{th\,p}$	linear	saturation
E	$U_{dd} + U_{th\,p} \le U_{inp} \le U_{dd}$	linear	subthreshold

Dominant during transition region

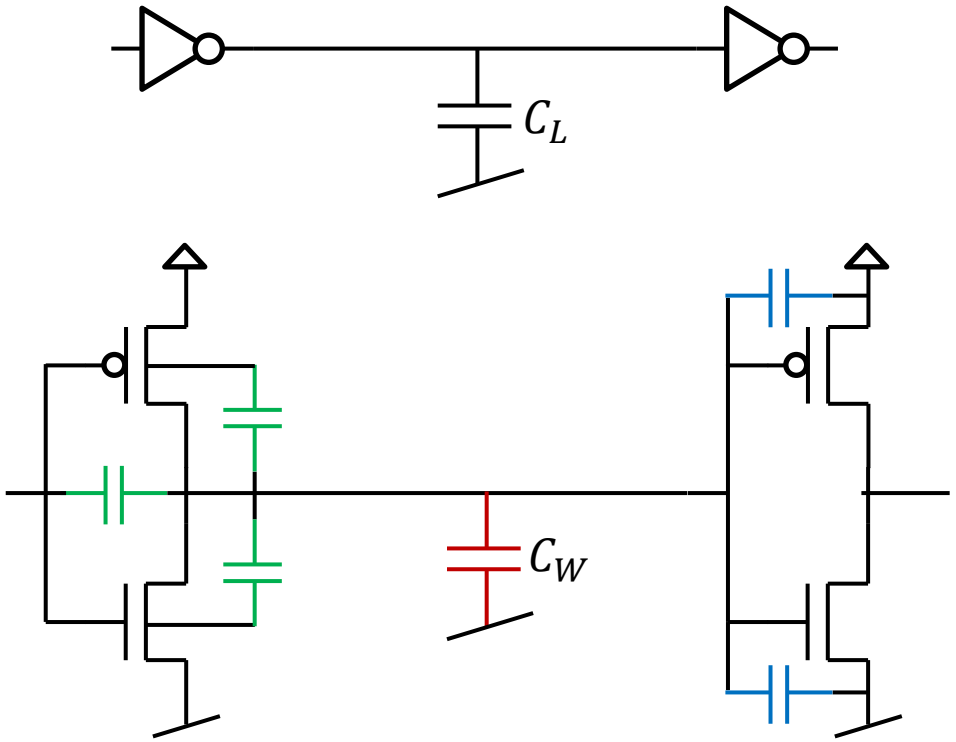




(a) Transfer characteristic (b) Crossover current (c) Logic states

- **Cross-over currents lead to power consumption during transients**

CMOS Gates With Capacitive Load

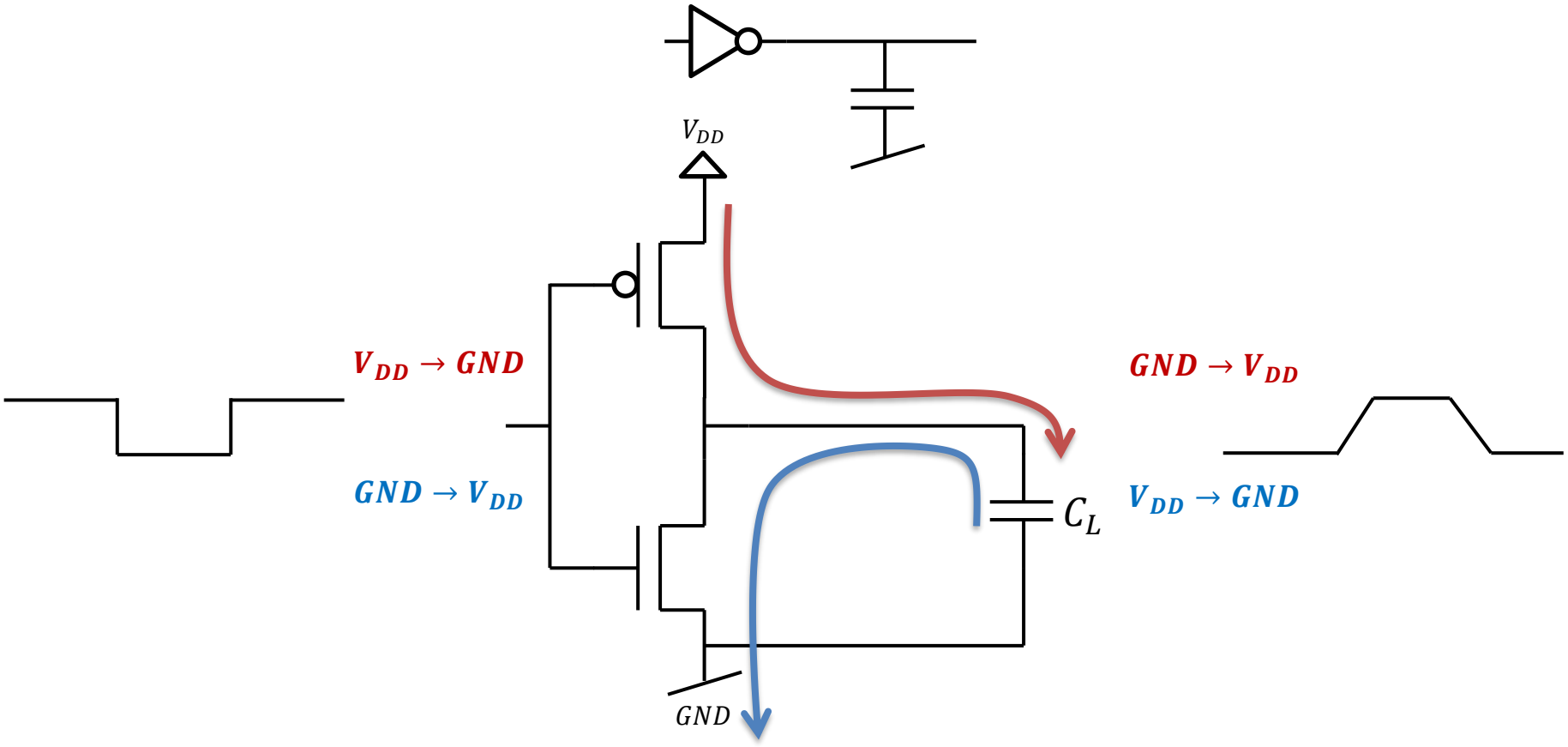


Wider transistors increase the gain factor (drive) but also increase the load (capacitance)

- **Various capacitances are merged into a single load capacitor C_L**
  - **Intrainsic MOS transistor capacitors (driver)**
  - Extrinsig **(fanout)** MOS transistor capacitances
  - **Interconnect capacitance**

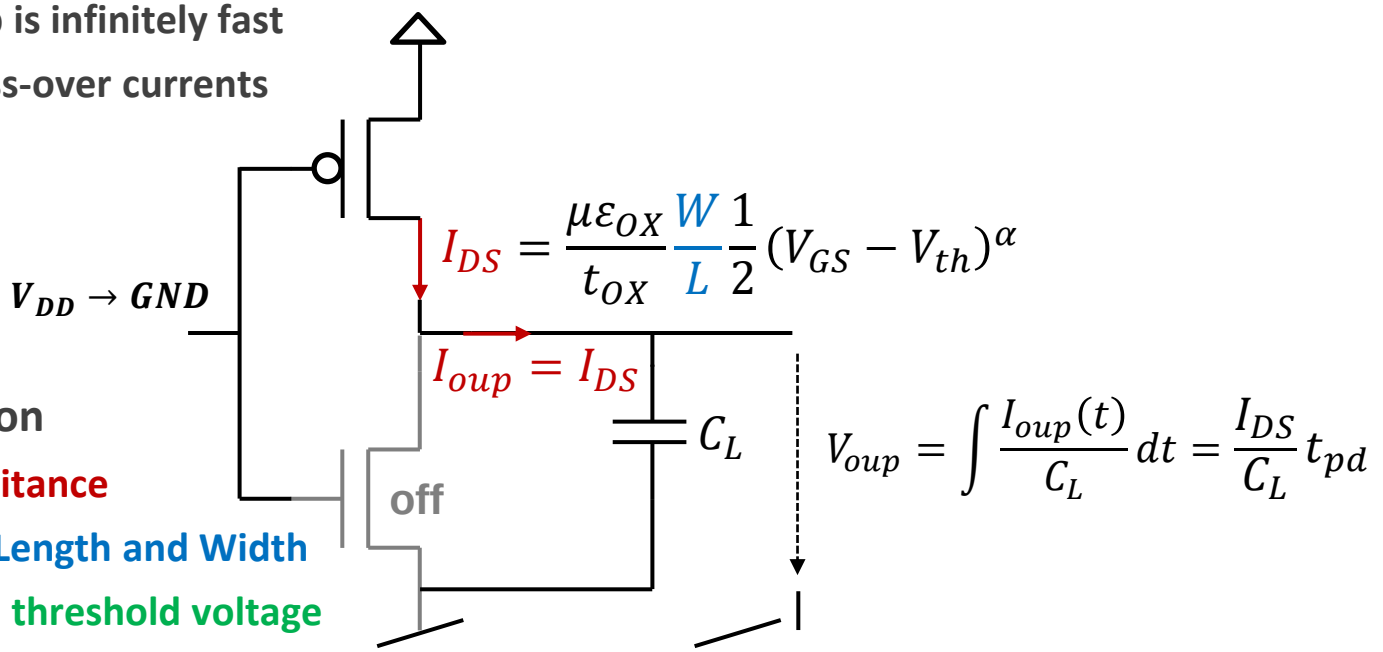
Dynamic Behavior of an Inverter with Capacitive Load

- Switching implies **charging** and **discharging** of the load capacitance which
  - Requires times and
  - Consumes energy: charges (current) flow from V_{DD} to GND

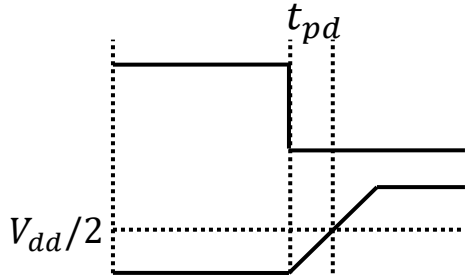


Dynamic Behavior of an Inverter with Capacitive Load (Delay)

- **Assumptions for a simple delay model for $V_{DD} > V_{TH}$ (nominal voltage)**
  - Current-carrying MOSFET operates in saturation throughout the transition
  - Input ramp is infinitely fast
  - Ignore cross-over currents



- **Delay depends on**
  - **Load capacitance**
  - **Transistor Length and Width**
  - **Supply and threshold voltage (overdrive)**

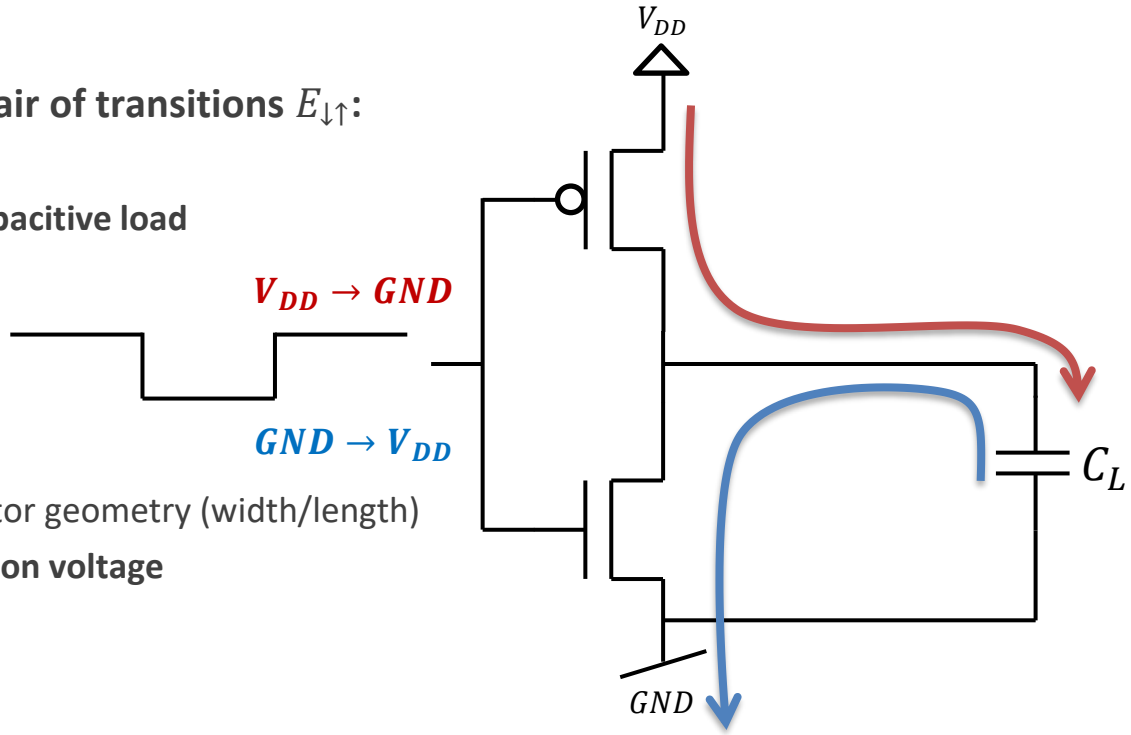


Sakuri α-power law for delay

$$t_{pd} = \frac{t_{OX}}{\mu\varepsilon_{OX}} \frac{L}{W} C_L \frac{V_{DD}}{(V_{DD} - V_{th})^{\alpha}}$$

Dynamic Behavior of an Inverter with Capacitive Load (Energy)

- **Energy consumed during one pair of transitions $E_{\downarrow\uparrow}$:**
  - **Cross-over currents**
  - **Charge pumped onto the capacitive load (dominant):**



- $E_{\downarrow\uparrow} = (C_L V_{dd}) V_{dd} = C_L V_{dd}^2$
  - independent of transistor geometry (width/length)
  - **quadratic dependency on voltage**

- Energy/transition
- $E_t = C_L V_{dd}^2 / 2$

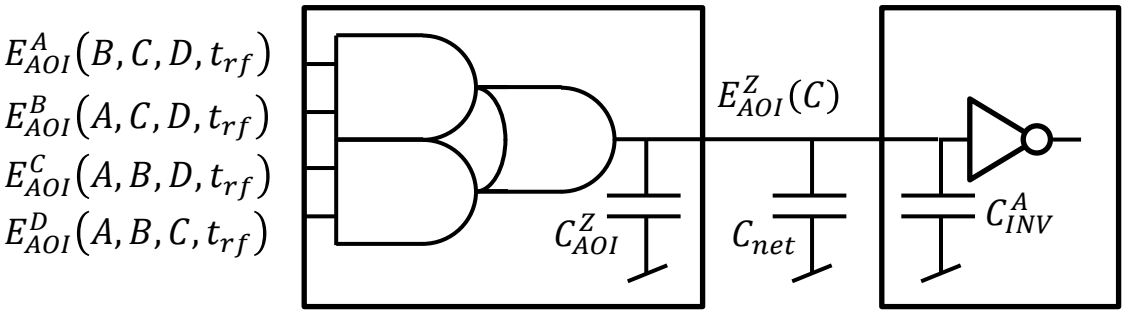
- Power consumption = Energy/transition * transition/cycle (β) * frequency (f_{clk})
- $P = \frac{C_L V_{dd}^2}{2} \beta f_{clk}$

Active Power Reduction on Register Transfer Level

Gate-Level Power Modeling

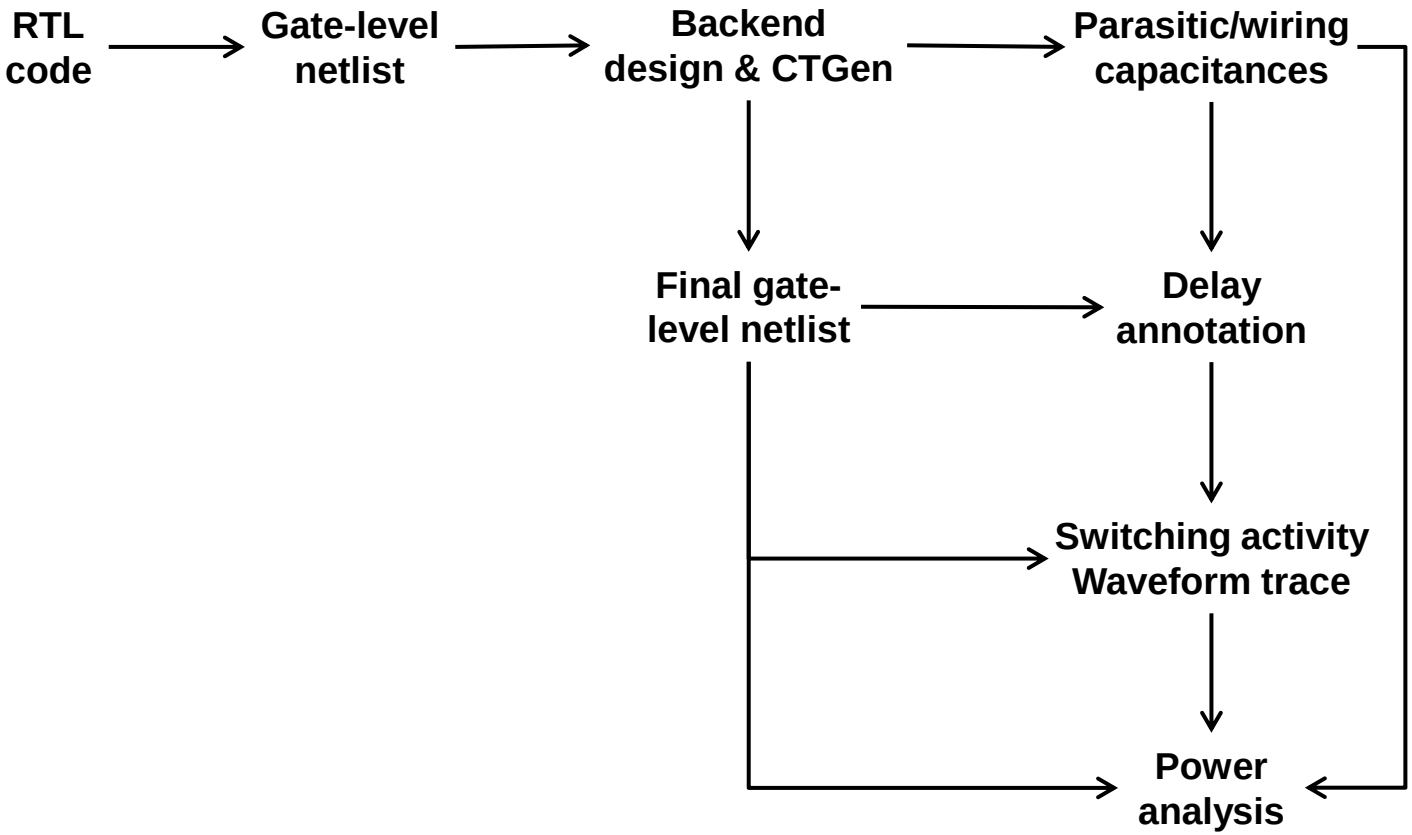
- **Power consumption is divided into**
  - **Net switching power**
  - **Internal power**
    - **Internal power depends on actual input values**
    - **Power is consumed even if output does not change**

- **Library files: internal energy characterization for each cell at given supply voltage**
  - **Internal energy (cross-current, switching) per change in each input and output (as functions of input slope t_{rf} and output load C)**
  - **Contribution to capacitance of the connected net (input/output load)**



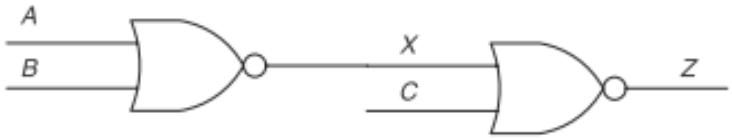
$$C = C^Z_{AOI} + C_{net} + C^A_{INV}$$

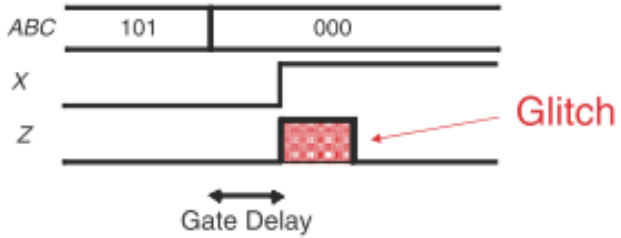
Gate-Level Power Analysis Flow



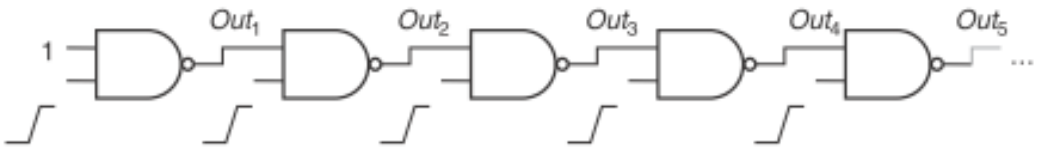
The need for dynamic power analysis

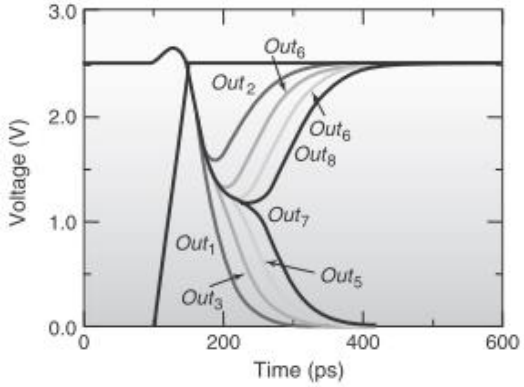
- Static analysis (timing and especially power) does not account for transient effects
- Glitches (dynamic hazards): single input change causes multiple changes in the output
  - Output: static analysis is correct
  - Static prediction of switching activity leads to wrong energy/power estimates
  - Dynamic simulation based power analysis provides reasonably correct results



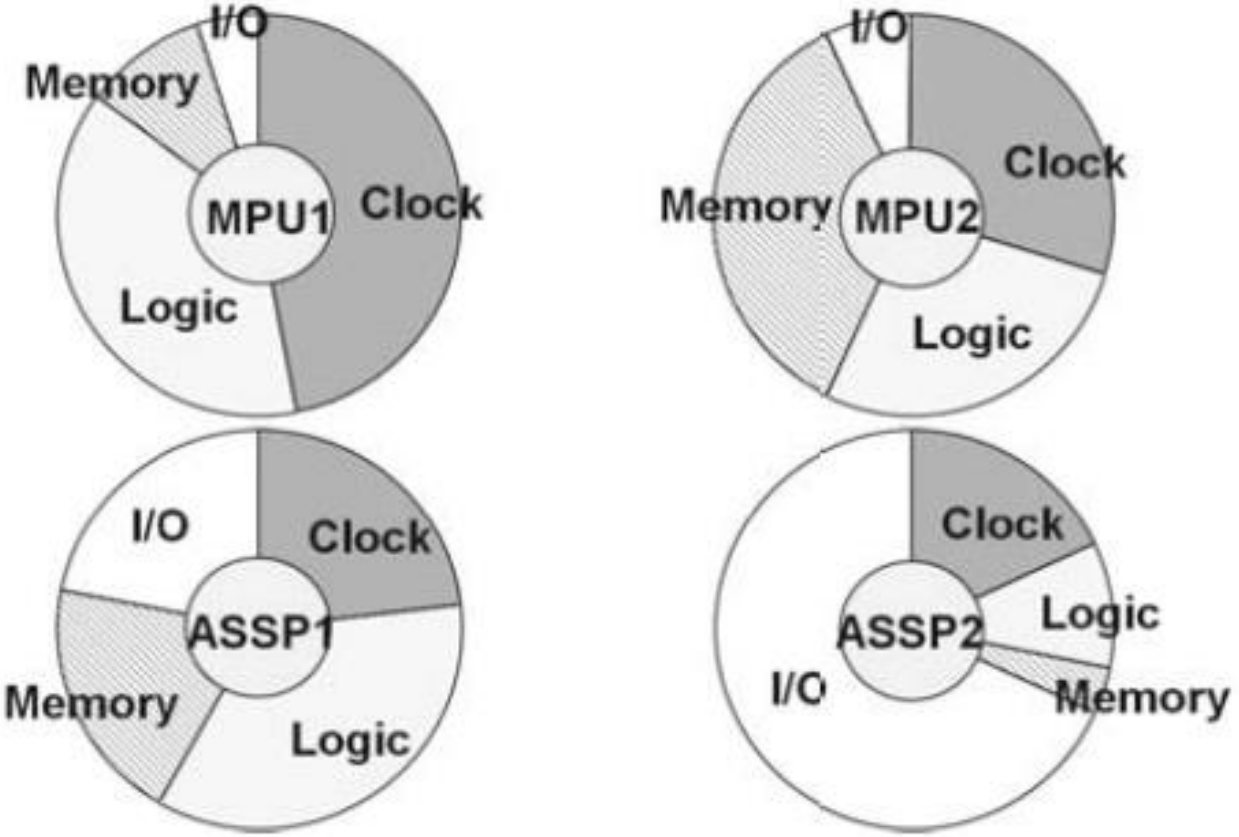


- Transient effects
  - Partial transitions: not captured by power analysis





- The clock is a major source of power consumption in many synchronous designs

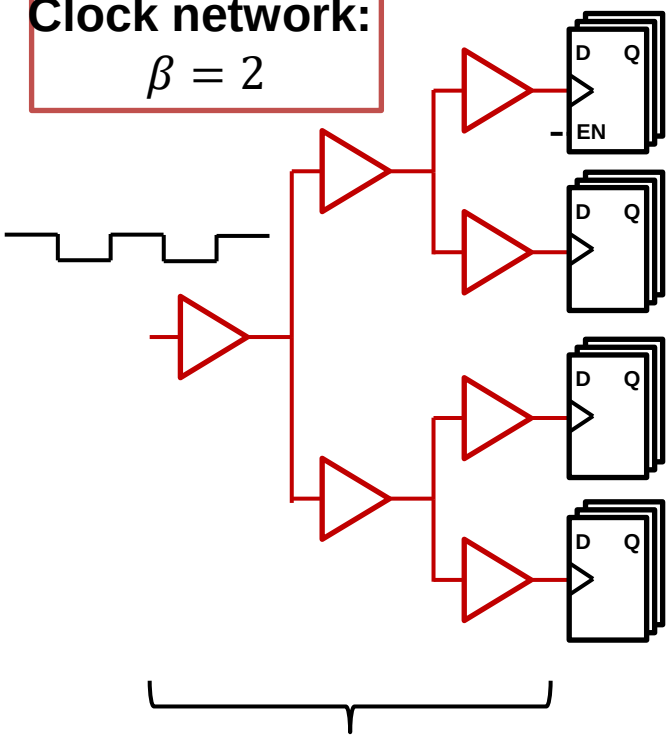


J. Rabaey: Power figures from sever microprocessors and DSPs

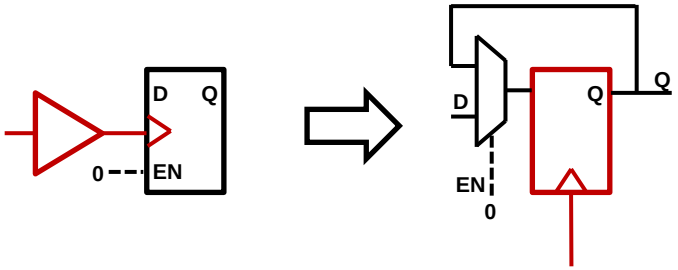
RTL Power Reduction: Clocking

- The clock is a major source of power consumption in many synchronous designs
  - Clock distribution network (clock tree)
  - Intrinsic power of sequential elements (even when data input is constant)

Clock network:
$\beta = 2$



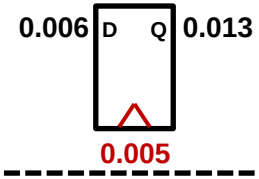
Clock tree: distribute clock signal with minimum skew to all sequential elements



Clock input still toggles even when no new data is latched (FF disabled)

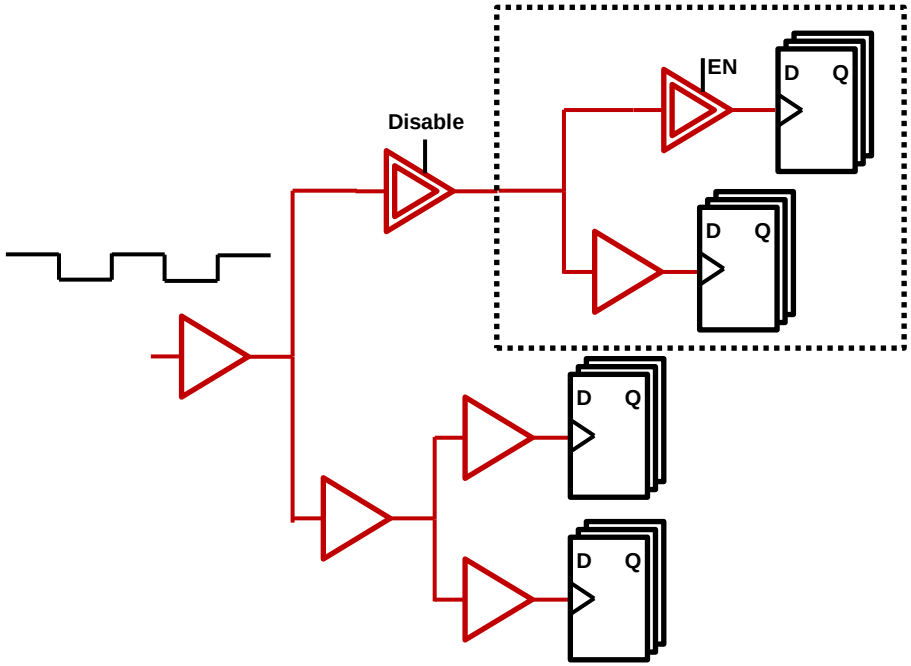
Example for DFF in 90nm CMOS

Cell internal power in [pJ/transition]

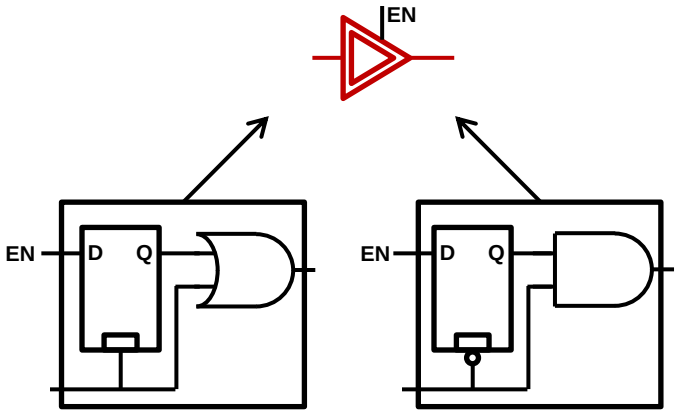


Clock input still toggles even when no new data is latched (FF disabled) causing significant power consumption

- Clock gating: reduce power consumption by disabling the clock for
  - Inactive parts of the design (coarse grained)
  - Disabling FFs without consuming internal power (fine-grained)



Need special clock-gating cells to protect against glitches in the Enable signal

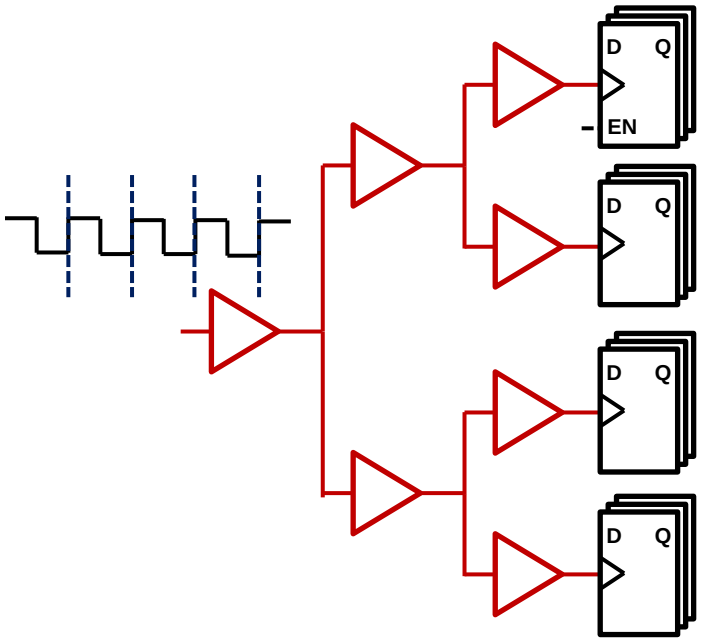


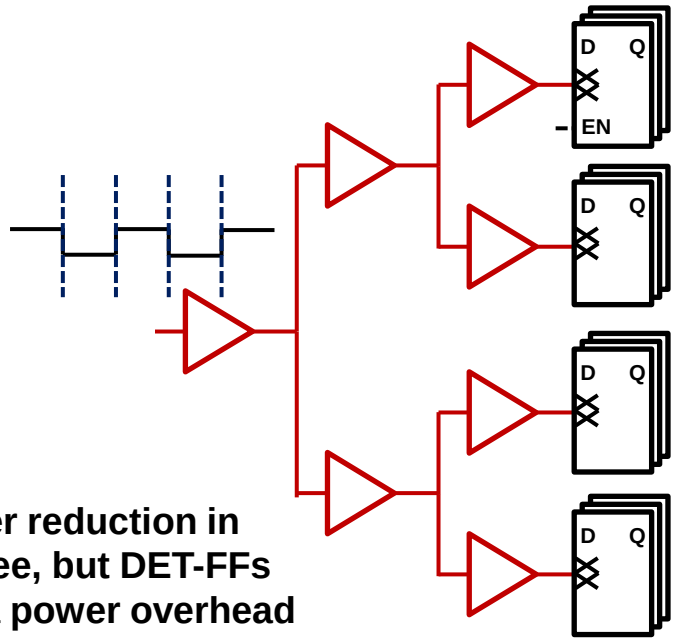
Power consumption can be on the order of 2-3 FFs: consider overhead!!

RTL Power Reduction: Clocking

- Double-data rate design
  - Clock network has the highest activity factor ($\beta = 2$)
  - Two transitions per clock period with only one transition triggering a state change

- Replace FFs with double-edge triggered FFs
  - Clock frequency can be cut in ½ for same number of operations



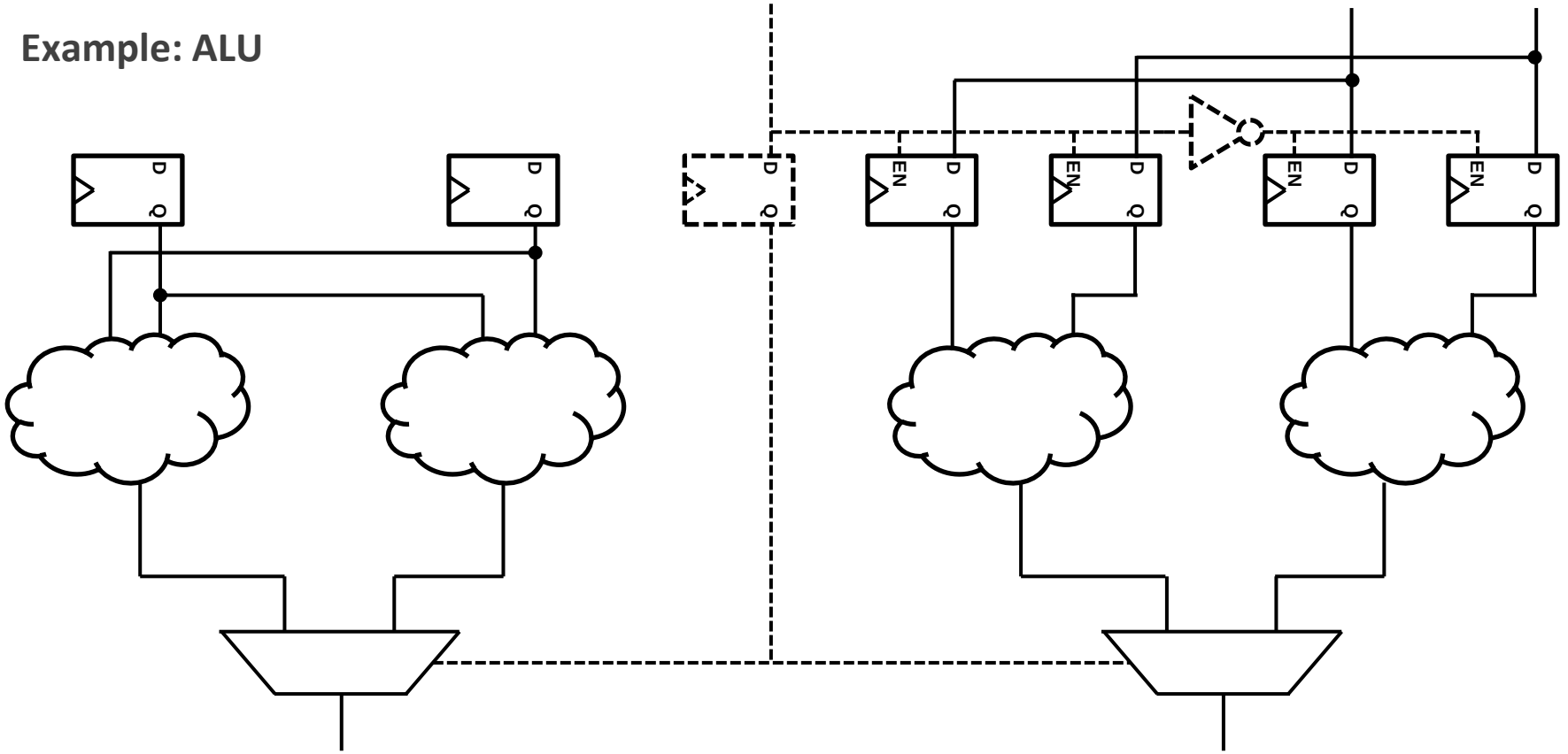


50% power reduction in the clock tree, but DET-FFs can involve a power overhead

RTL Power Reduction: Logic

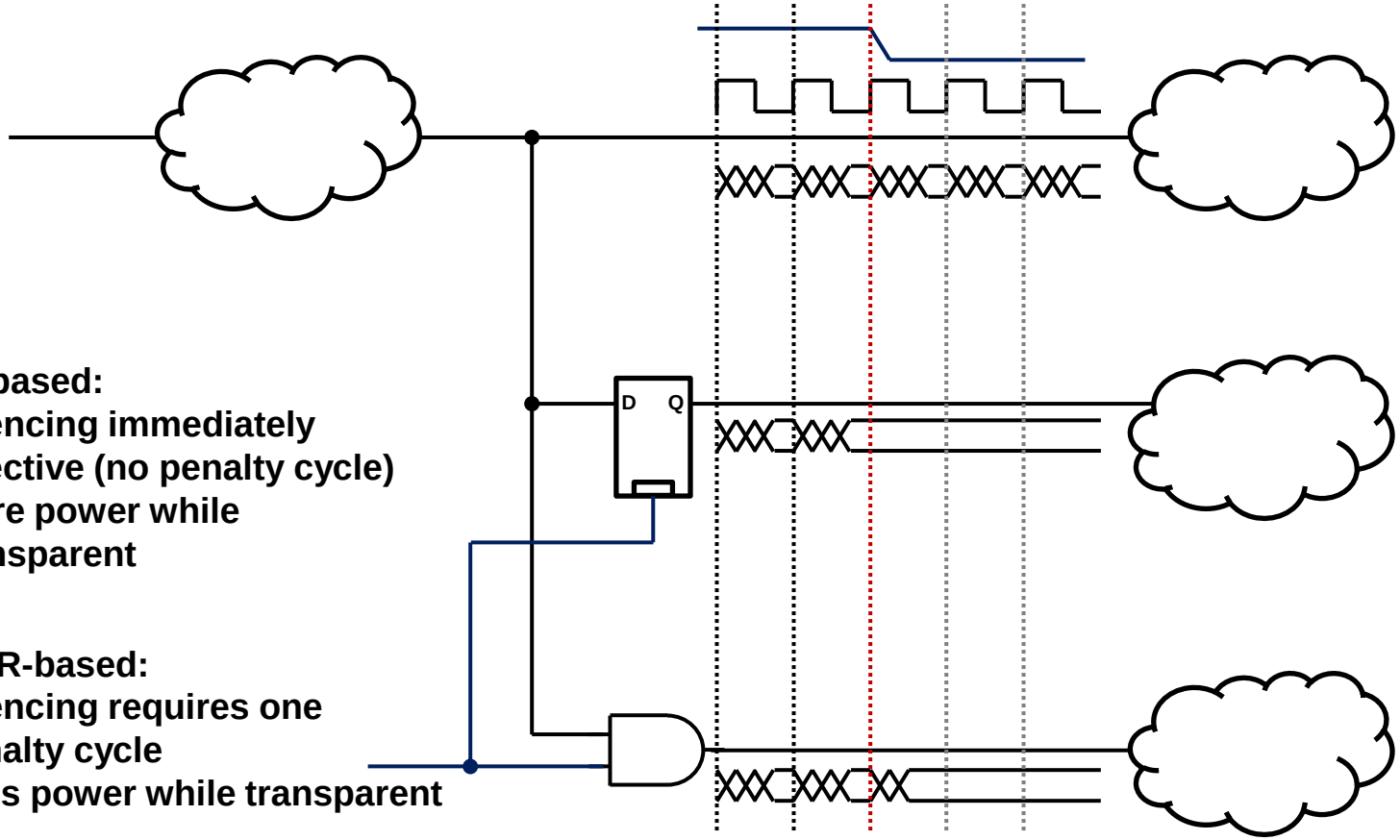
- **Operand isolation**
  - Logic consumes power whenever the input changes
  - Computation of unused results consumed unnecessary power

- **Example: ALU**



RTL Power Reduction: Logic

- **Silencing:** avoid activity in unused logic
  - **Unused logic is not always immediately preceded by registers**
  - **Avoid changes to the input of unused parts of the logic**



Latch-based:
- **Silencing immediately effective (no penalty cycle)**
- **More power while transparent**

AND/OR-based:
- **Silencing requires one penalty cycle**
- **Less power while transparent**

Voltage Scaling and Sub-VT Design

Supply voltage down-scaling (V_{dd} ↓) leads to…

Quadratic improvement in energy

$$E \propto V_{\mathrm{dd}}^2$$

$$f \propto V_{dd}$$

Linear reduction in speed

Cubic reduction in power

$$P \propto f \cdot V_{\mathrm{dd}}^2 = V_{\mathrm{dd}}^3$$

- **Inverter Delay as proxy for critical path delay (max. freq.)**

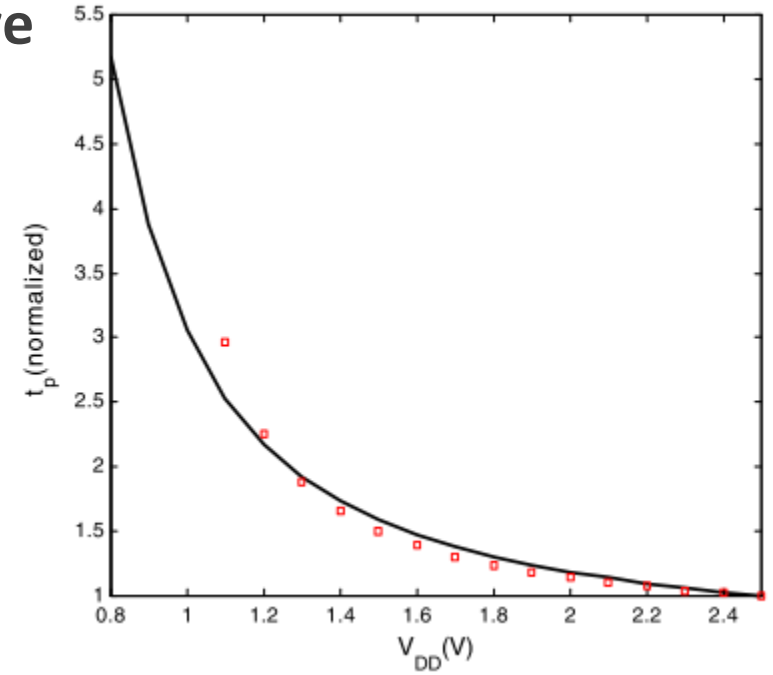
$$t_{pd} \sim \frac{V_{DD} C_L}{\frac{W}{L} \mu C_{ox} (V_{DD} - V_T)^2}$$

- **Delay as a function of the overdrive (supply voltage above V_T)**
  - Impact on maximum frequency

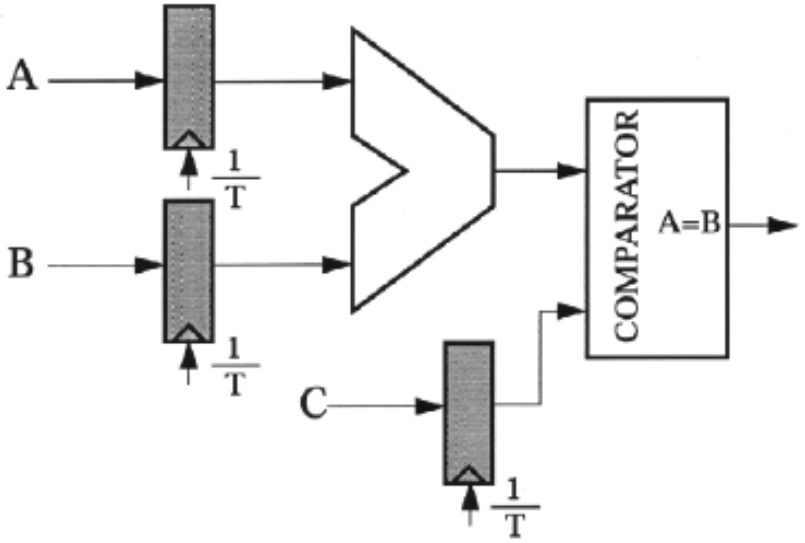
$V_{DD} \uparrow \rightarrow f_{max} \uparrow$

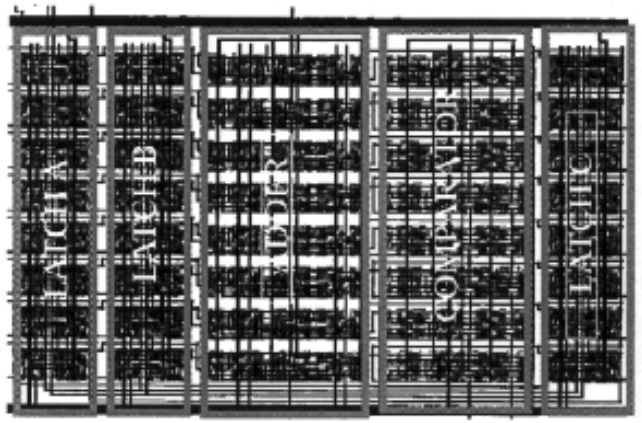
$V_{DD} \downarrow \rightarrow f_{max} \downarrow$

$$f_{max} \sim \frac{1}{t_{pd}}$$



Dealing with Performance Loss is Easy

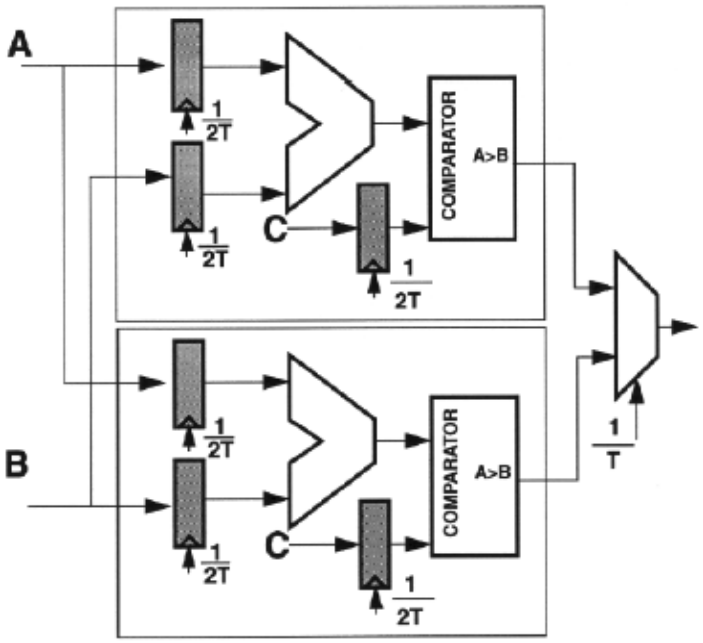


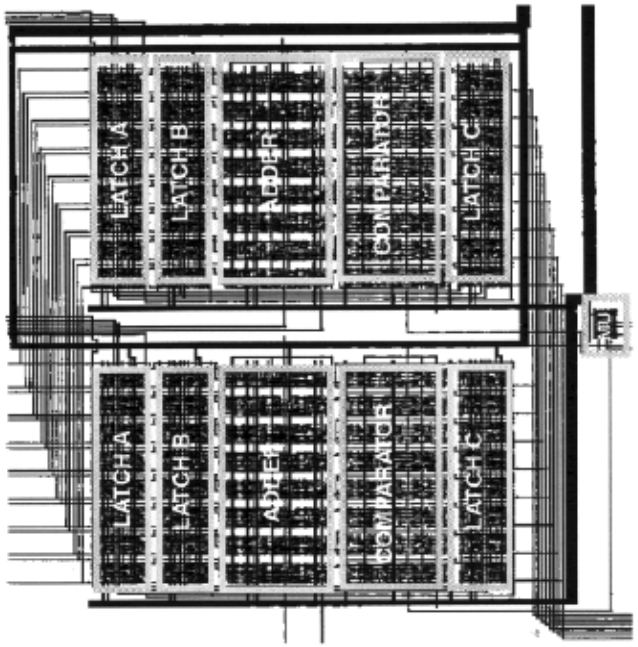


Area = 636 x 833 u^2

- Critical path delay: $T_{adder} + T_{comparator} = 25$ ns
- Frequency: $f_{ref} = 40$ MHz
- Total switched capacitance = C_{ref}
- $V_{dd} = V_{ref} = 5V$
- Power for reference datapath = $P_{ref} = C_{ref} V_{ref}^2 f_{ref}$

Dealing with Performance Loss is Easy: Parallel Datapaths



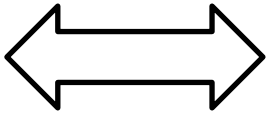


Area = 1476 x 1219 μ^2

- The clock rate can be reduced by x2 with the same throughput: $f_{par} = f_{ref}/2 = 20$ MHz
- Total switched capacitance = $C_{par} = 2.15C_{ref}$
- $V_{par} = V_{ref}/1.7$
- $P_{par} = (2.15C_{ref})(V_{ref}/1.7)^2(f_{ref}/2) = 0.36P_{ref}$

Unfortunately, Energy Proportionality is Hard to Achieve

- Energy proportionality: $Energy \sim Work$

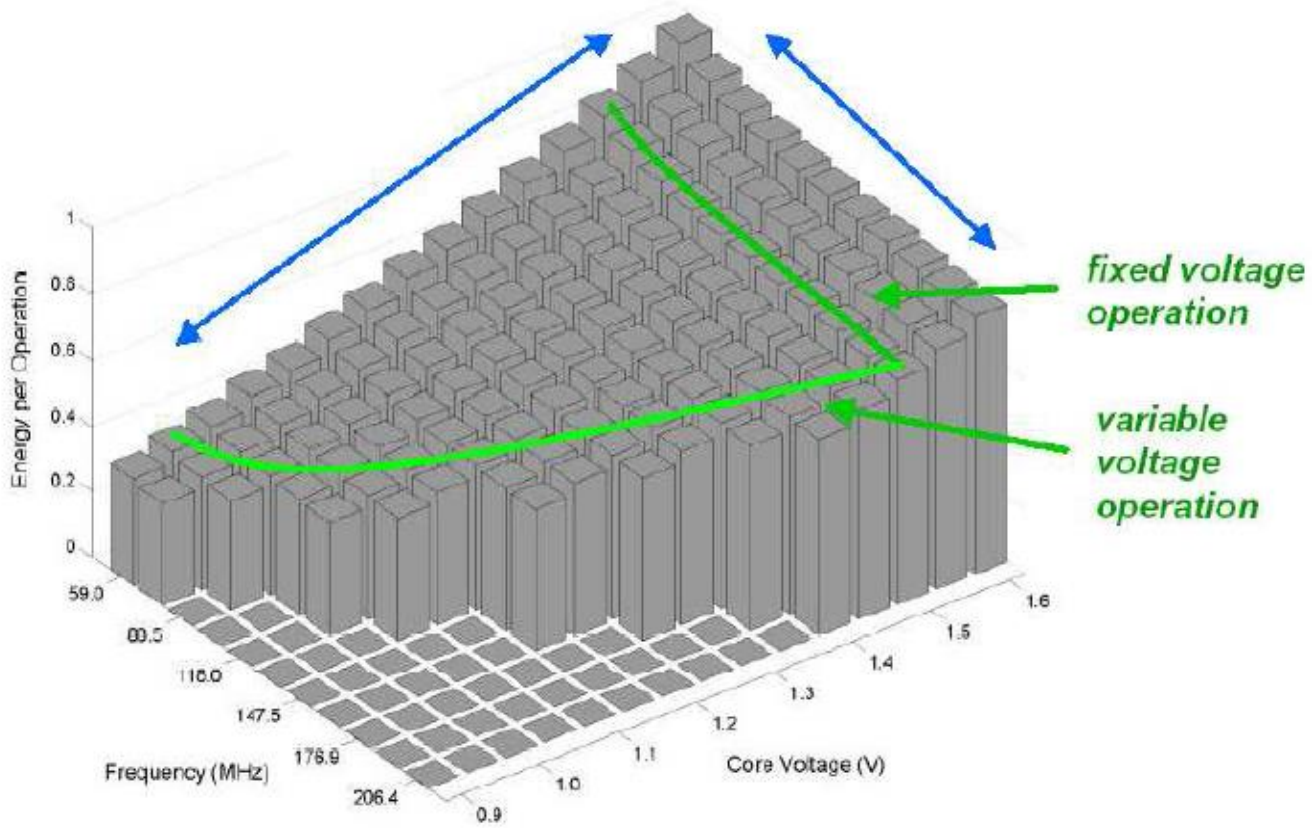


- Rarely achieved over a large range…

- Beyond energy proportionality: $Energy \sim {Work}/{Time}$
- Doing things fast is more difficult than doing things slow
- Even more difficult to achieve!!

BUT WHY?

Example: StrongARM SA-1100 processor



Energy per operation as a function of voltage and clock rate

Observation: better energy efficiency for higher frequency at constant voltage.

Leakage Power and Leakage Power Reduction as Main Overhead in ULP Systems

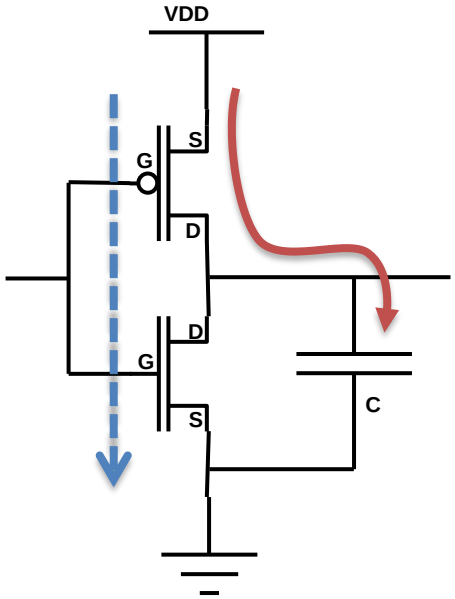
Impact of Static Currents (Overhead)

$$E = \int_0^{T_{clk}} \left(\frac{C_L V_{dd}^2}{2} \beta \frac{1}{T_{clk}} + V_{dd} I_{static} \right) dt$$

Total Energy Consumption

Static Energy Consumption

Dynamic Energy Consumption



$$\int_0^{T_{clk}} V_{dd} I_{static} dt$$

$$\propto V_{dd} I_{static} T_{clk}$$

$$\frac{1}{T_{clk}} \propto V_{dd}$$

$$\int_0^{T_{clk}} \frac{C_L V_{dd}^2}{2} \beta \frac{1}{T_{clk}} dt$$

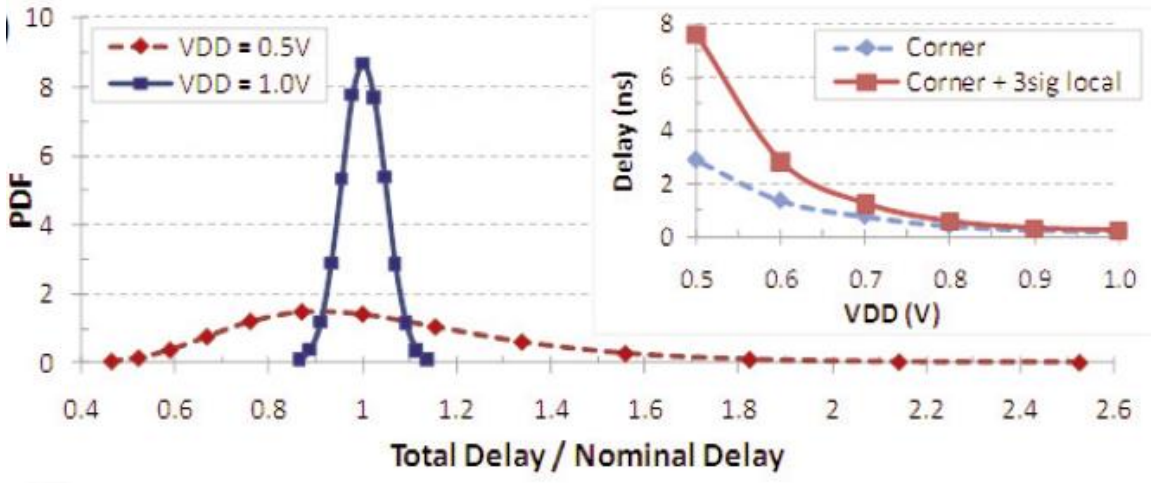
$$\propto \frac{C_L V_{dd}^2}{2} \beta$$

Ultra-Low-Power Design: Sub-Threshold Operation

- Scaling supply voltage below the threshold voltage
  - Transistors operate in the sub-threshold regime
- Impact on gate delay

$$t_{pd} = \frac{C_L V_{DD}}{I_0 e^{\frac{V_{DD} - V_{th}}{v_t n}}}$$

  - Reducing supply voltage rapidly (exponentially) increases the delay
  - Sensitivity: a small variation in V_{th} or in V_{dd} leads to a large change in delay (see later -> variability)

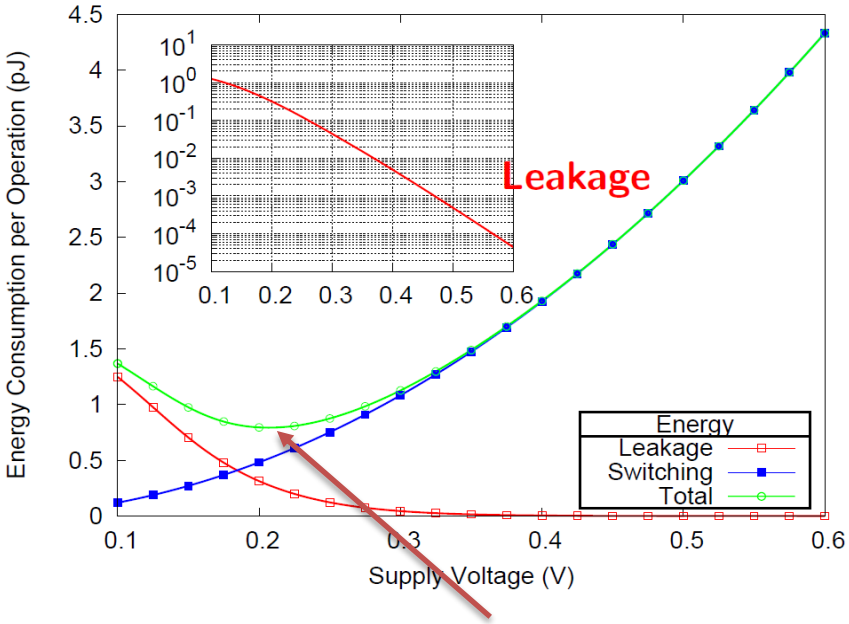


Gammie, G.; Ickes, N.; Sinangil, M.E.; Rithe, R.; Gu, J.; Wang, A; Mair, H.; Datla, S.; Bing Rong; Honnavara-Prasad, S.; Ho, L.; Baldwin, G.; Buss, D.; Chandrakasan, AP.; Uming Ko, "A 28nm 0.6V low-power DSP for mobile applications," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International* , vol., no., pp.132,134, 20-24 Feb. 2011

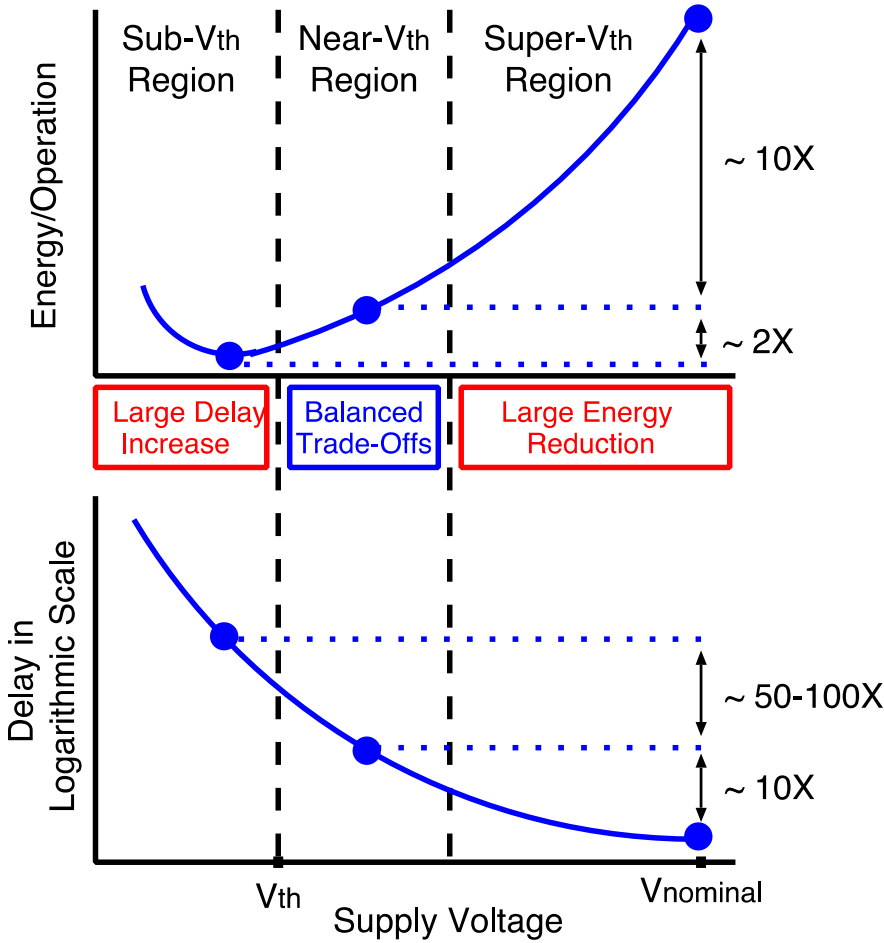
Ultra-Low-Power Design: Sub-Threshold Operation

- **Near/below VT operation:**
  - **Exponential delay/leakage increase**
  - **Minimum energy voltage: balance between leakage and active power consumption**

J. Rodrigues, PATMOS 2011, Keynote

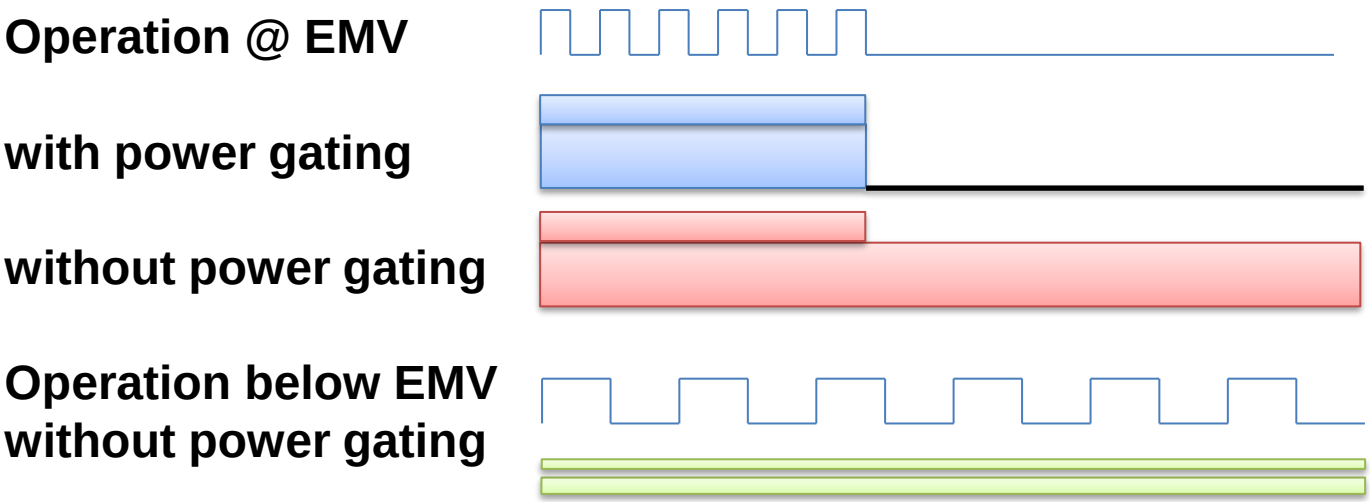


Relatively flat around EMV



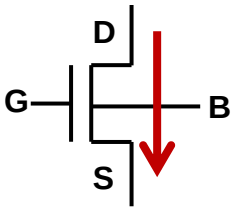
Ultra-Low-Power Design: Sub-Threshold Operation

- Real-time embedded system requirements
  - Handle a given workload with lowest power consumption
- Optimum solution
  - Operation at the energy minimum voltage with power gating during idle periods to avoid leakage
  - But, power gating is only effective when idle periods are long and memories can often not be power gated and are the major source of leakage

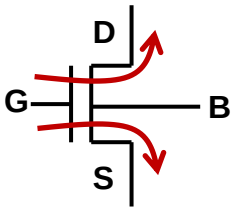


Leakage Power: the Hidden Evil

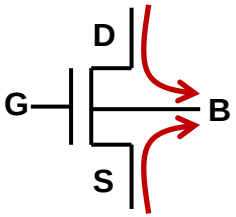
- Transistors leak currents even when in off-state
- Sources for leakage
  - Sub-threshold leakage
    - Dominant component in most circuits

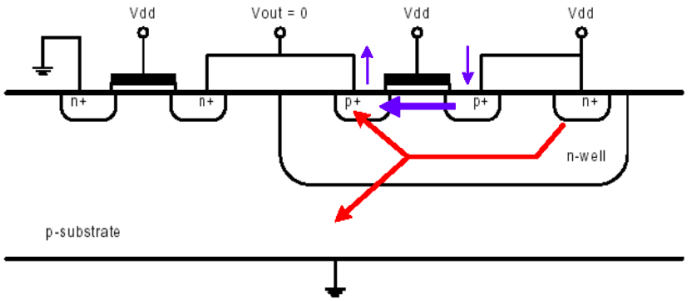


  - Gate tunneling
    - Generally low, even in modern technologies due to high-k gate dielectrics
    - Decreases very rapidly with decreasing V_{dd}



  - Junction current
    - Generally low
    - Decreases very rapidly with decreasing V_{dd}



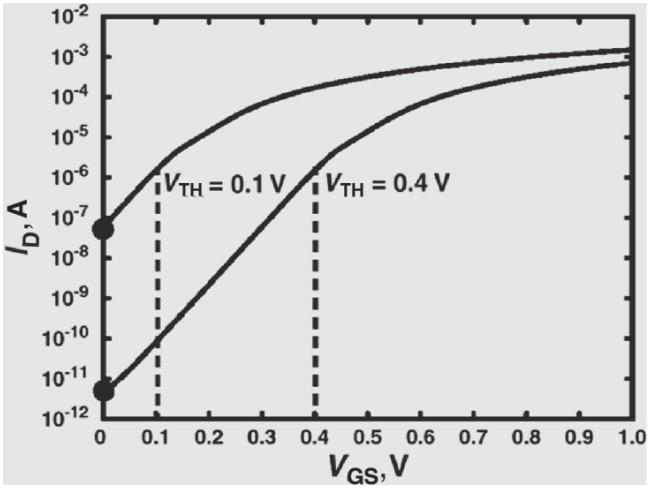


$$I_{subthreshold} = I_O \cdot e^{\frac{q(V_{gs}-V_T)}{\alpha kT}}$$

$$I_{reverse} = A \cdot J_S \left(e^{\frac{qVbias}{kT}} - 1 \right)$$

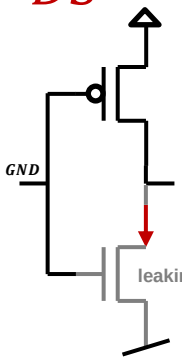
Leakage Power

- **Long channel deices (>130nm):** $I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}}$
  - I_{DS} **mostly independent from Drain-Source Voltage**
  - **Leakage current depends strongly on** $V_{GS} - V_{th}$
    - **Decreasing threshold voltage increases leakage**



- **Impact of technology scaling on sub-threshold leakage (<130nm)**
  - **Drain-Induced Barrier Lowering (DIBL):** V_{DS} **modulates threshold voltage**
  - I_{DS} **becomes a function of** V_{DS}

- $I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}+\lambda_{DS}V_{DS}}{v_t n}}$

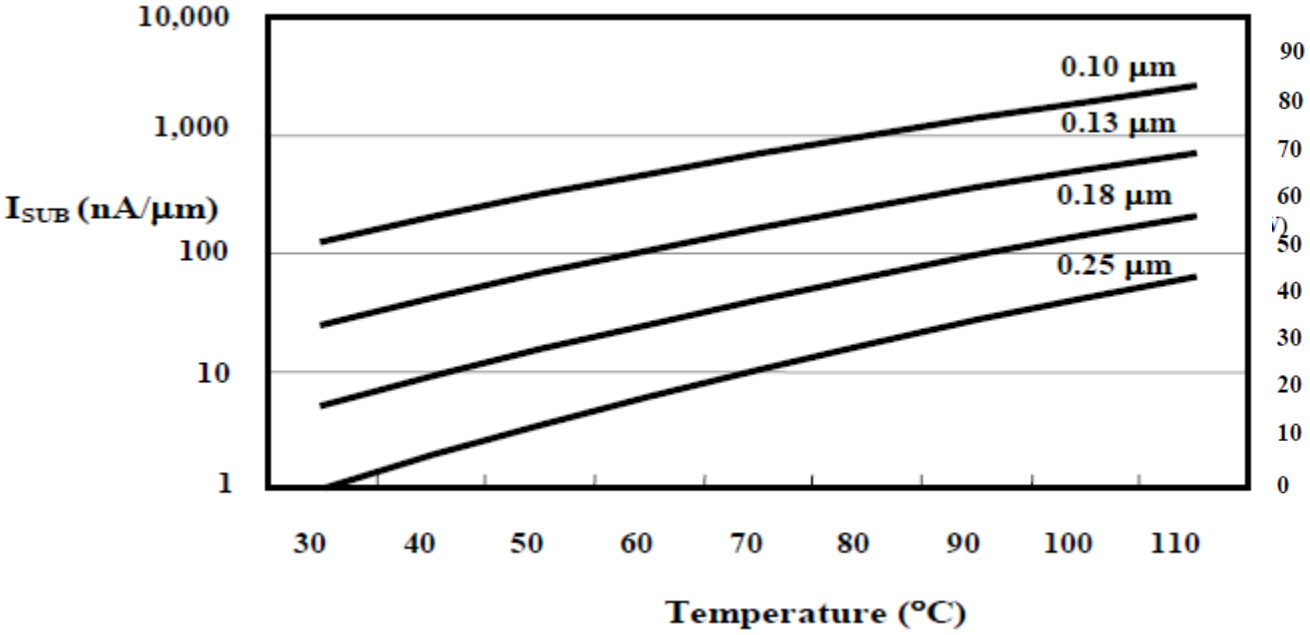


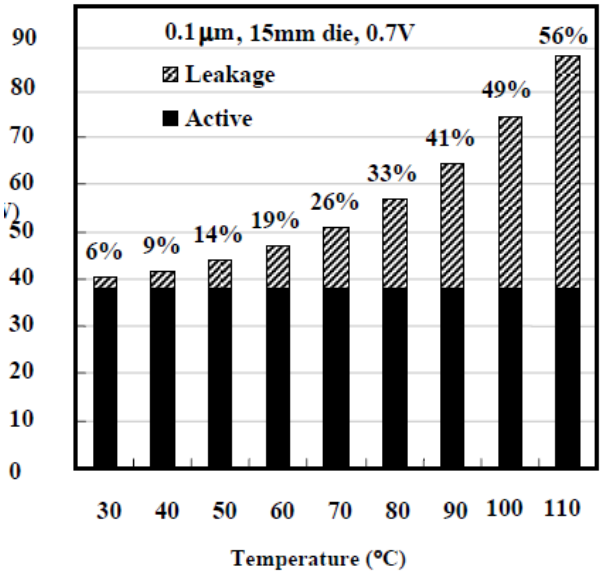
$I_{leak} = I_0 e^{\frac{-V_{th}+\lambda_{DS}V_{DD}}{v_t n}}$ ← **Voltage scaling reduces leakage**

Drain current depends exponentially on thermal voltage $v_t = kT/q$

$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}}$$

- **Exponential I_{DS} increase with temperature**



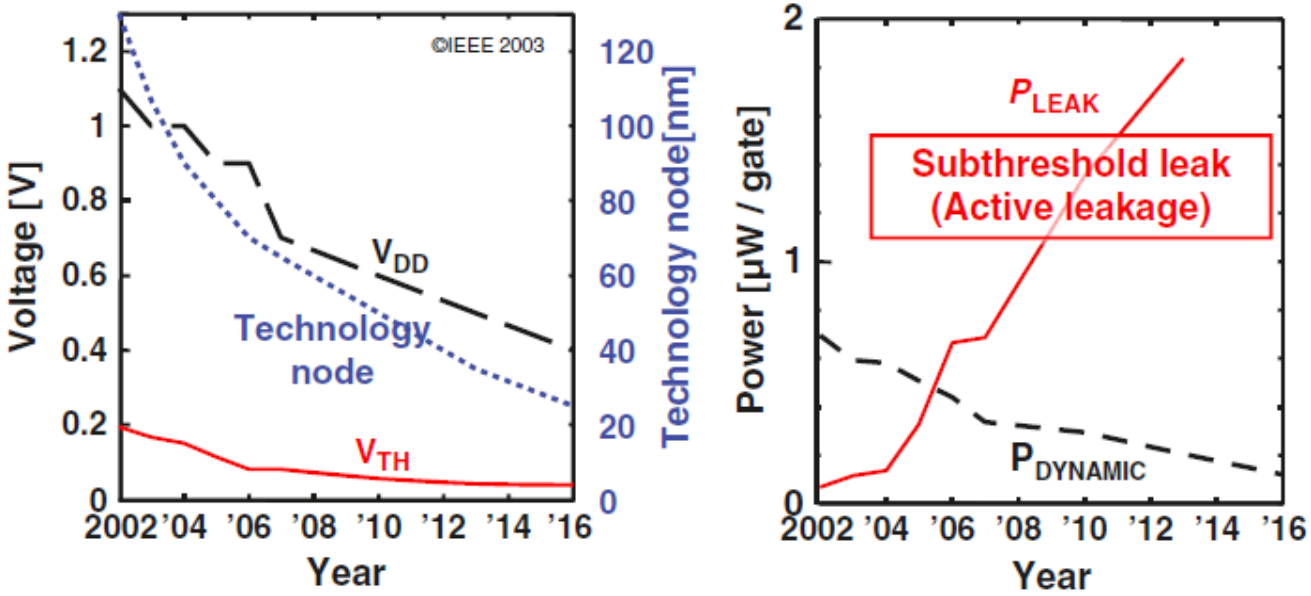


Example: 0.7V, 100nm process, 15mm2 die

Vivek De, Intel

Technology Trends in Power Consumption

- **Constant-filed scaling:** for given circuit at constant frequency, scaling
  - **reduces dynamic (active) power consumption**
  - **Increases leakage power dramatically (due to Vth reduction)**
- **Vth scaling limited => Vdd scaling limited => impact of technology scaling decreases**

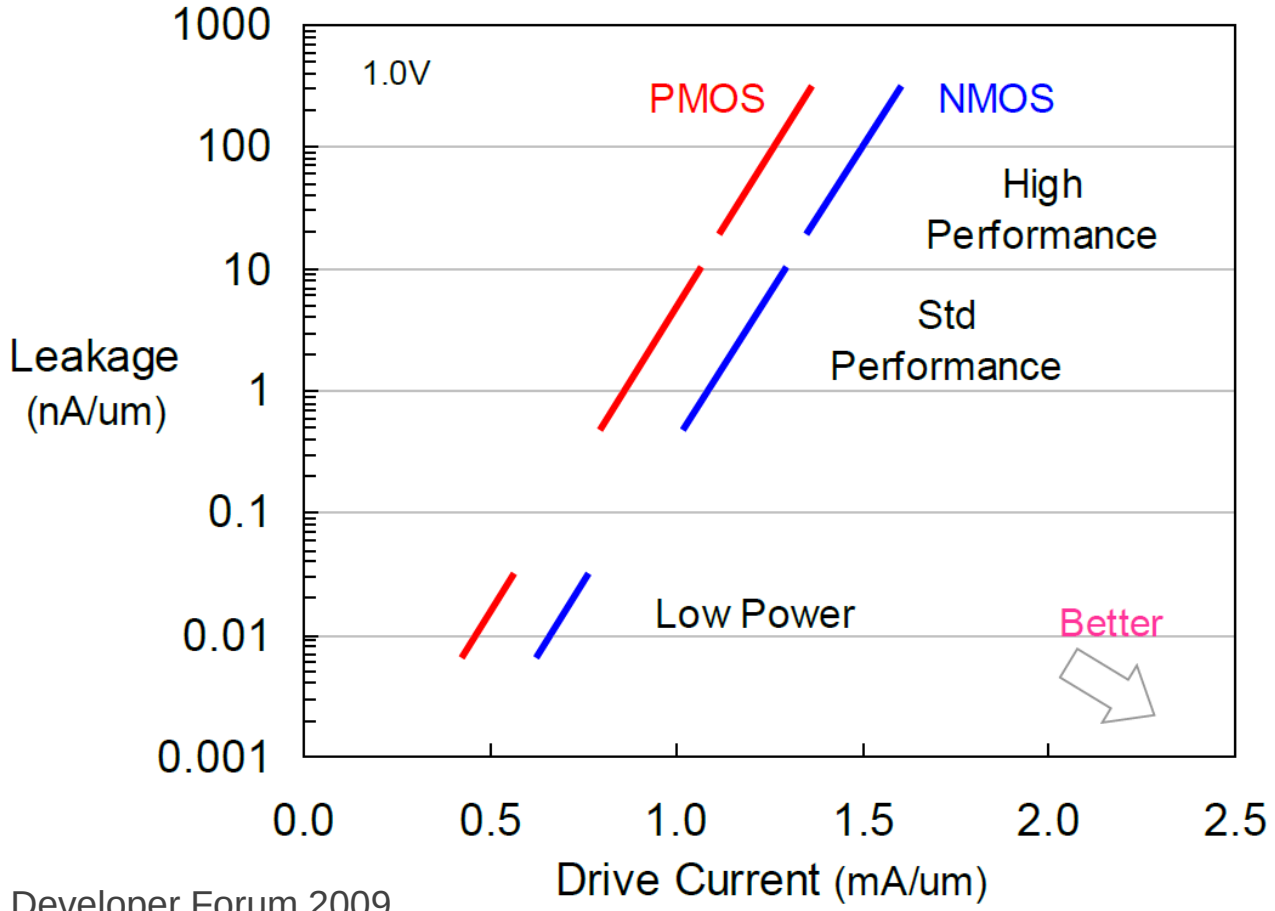


[Ref: T. Sakurai, ISSCC'03]

- **Leakage: important role compared to low active power and long standby**

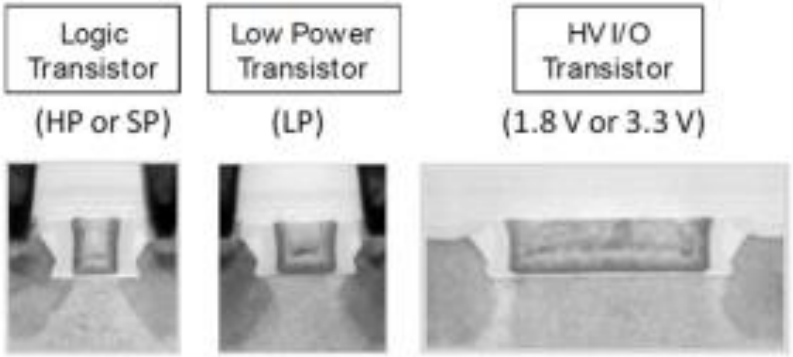
Modern technologies offer different device flavors

- **Devices with different threshold voltages => can often be combined on same die/wafer**
- **Different process flavors (can typically not be mixed on same wafer)**



M. Bohr, Intel Developer Forum 2009

Modern technologies offer different device flavors



Transistor Type	Logic (option for HP or SP)		Low Power	HV I/O (option for 1.8 or 3.3 V)	
	HP	SP	LP	1.8V	3.3V
EOT(nm)	0.95	0.95	0.95	~ 4	~ 7
Vdd (V)	.75/ 1	.75/ 1	0.75/1.2	1.5 /1.8	1.5 /3.3
Pitch(nm)	112.5	112.5	126	min. 338	min. 675
Lgate (nm)	30	34	46	>140	>320
NMOS Idsat (mA/um)	1.53 @ 1 V	1.12 @ 1 V	0.71 @ 1 V	0.68 1.8 V	0.7 3.3 V
PMOS Idsat (mA/um)	1.23 @ 1V	0.87 @ 1 V	0.55 @ 1 V	0.59 1.8 V	.34 @3.3 V
Ioff (nA/um)	100	1	0.03	0.1	<0.01

- **Sometimes IO transistors are an interesting option: low-leakage, high-VT but large distance to core transistors in the layout required**

Leakage Power Reduction

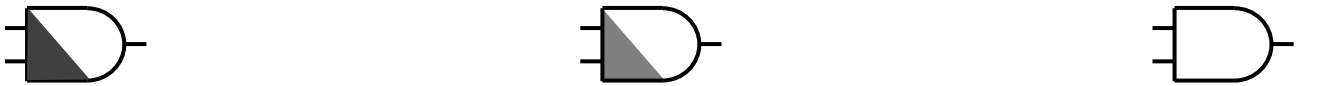
Threshold Voltage Selection

- **Modern process technologies support devices with different threshold voltages**
  - **Typically three flavors: low-VT, standard-VT, high-VT**
  - **Often all three flavors can be mixed in the same design**

- **VT-selection:** tradeoff between speed and leakage

$$t_{pd} = \frac{t_{OX}}{\mu \varepsilon_{OX}} \frac{L}{W} C_L \frac{V_{DD}}{(V_{DD} - V_{th})^{\alpha}} \qquad I_{leak} = I_0 e^{\frac{-V_{th} + \lambda_{DS} V_{DS}}{v_t n}}$$

- **Example**: 55nm process



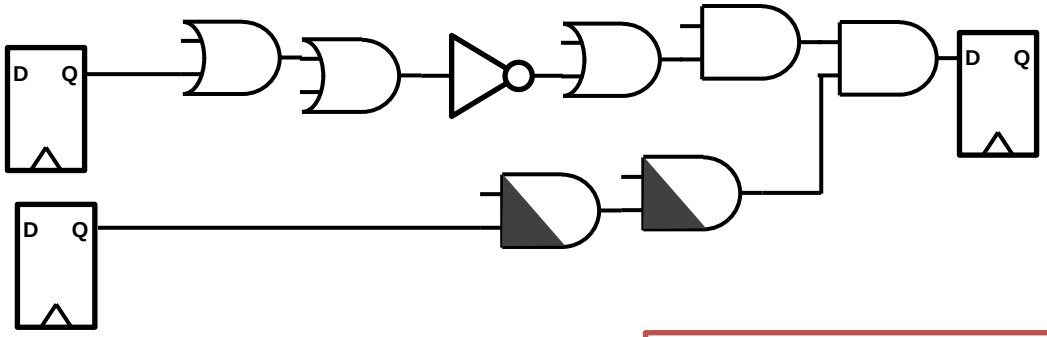
	HVT	SVT	LVT
Delay	**20ps**	**16ps**	**14ps**
Leakage	**30nW**	**60nW**	**200nW**

  - **Small increase in speed comes with a significant leakage penalty**

Multi-VT Design

- **Design tradeoff when choosing a VT flavor:**
  - Less leakage (high-VT) increases delay and vice versa
  - **Threshold voltage types can often be mixed**

- **Multi-VT design**



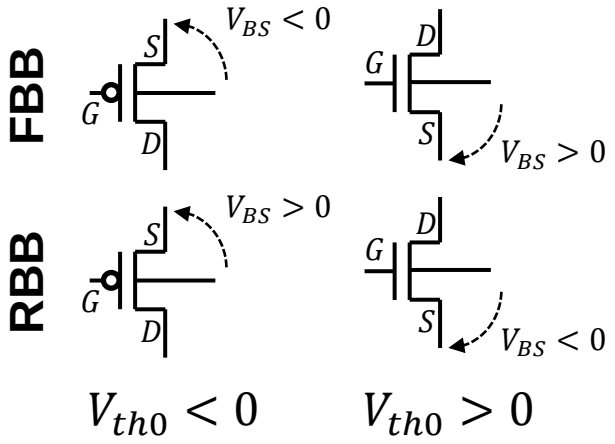
  - **Use low-VT cells only on critical paths**
  - **High-VT cells are used in all other paths**

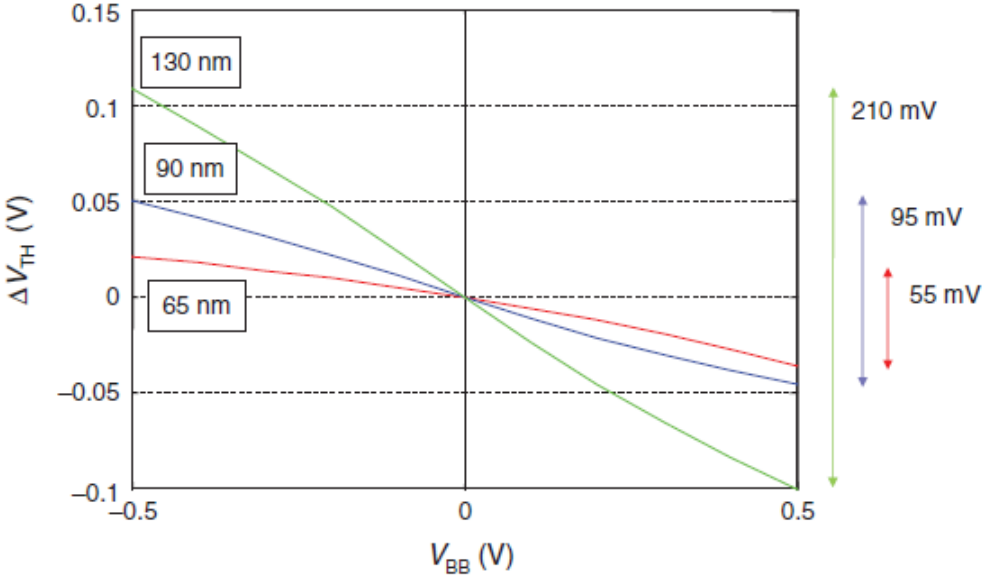
Caveat: can be very problematic for near-VT or sub-VT design: path delays scale very differently

- **Methodology:**
  - **Either done by replacing non-critical cells in the backend OR already during synthesis by providing multiple libraries (HVT/SVT *and* LVT)**

Body Bias Modulates Threshold Voltage

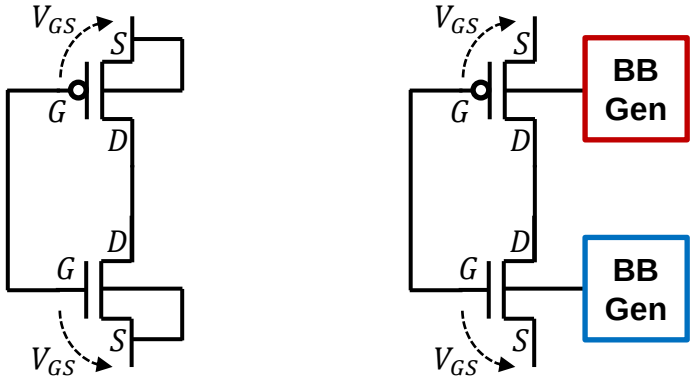
- Body of the transistor is often connected to the source (no body bias)
- Introducing a body bias modulates threshold voltage
  - Forward Body Bias (FBB): increases threshold voltage
  - Reverse Body Bias (RBB): reduces threshold voltage
- $V_{th} = V_{th0} - \lambda_{BS} V_{BS}$
- BULK CMOS:
  - Effect of body bias decreases for technologies below 100nm
  - FBB is limited to ~300mV to avoid operating junction diodes in forward direction

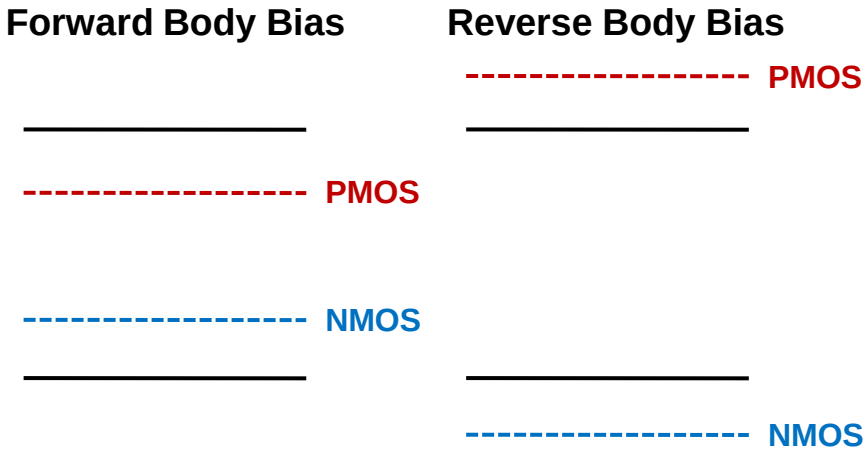




Body Bias for Leakage Reduction

- **Application to logic**

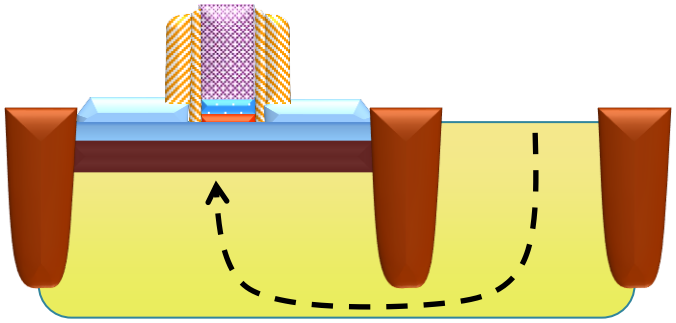




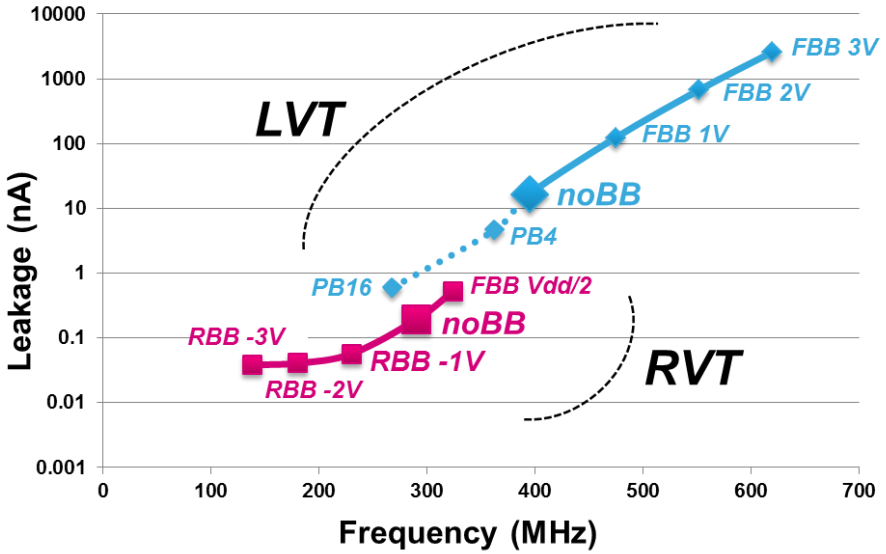
  - FBB improves speed, but also increases leakage
  - RBB reduces leakage, but also increases delay
- **Dynamic body bias:**
  - Adjust BB (threshold voltage) dynamically for sleep-mode and active-mode
  - Generation of bias voltages involves overhead
    - Voltage regulators (usually only very small currents)
    - Generation of negative or >VDD voltages (switching regulator)

Body Bias in FD-SOI Technologies

- Thin un-doped channel
- Thin buried oxide layer (BOX) isolates channel from the body (back-gate)
- Body contact allows to control the back-gate



- **Advantages:**
  - Un-doped channel avoids V_{th} variation due to RDF compared to Bulk-CMOS
  - Threshold voltage modulation through back-gate
  - Large back-gate voltage range and high V_{th} sensitivity: ~60 mV/V



Leakage in Transistor Stacks

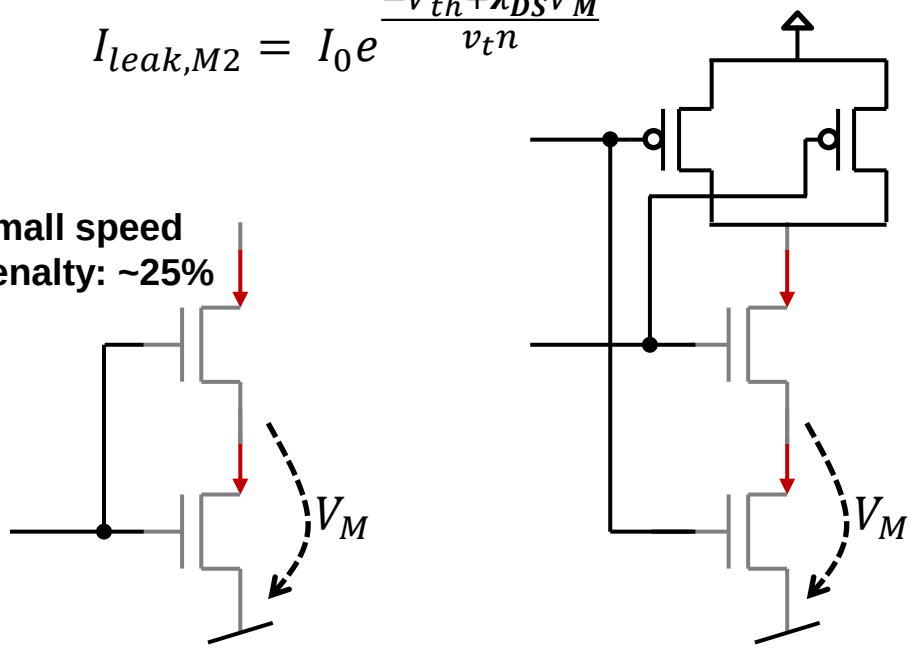
- Stacking occurs
  - In many logic gates (> 1 input)
  - When introduced intentionally for leakage reduction

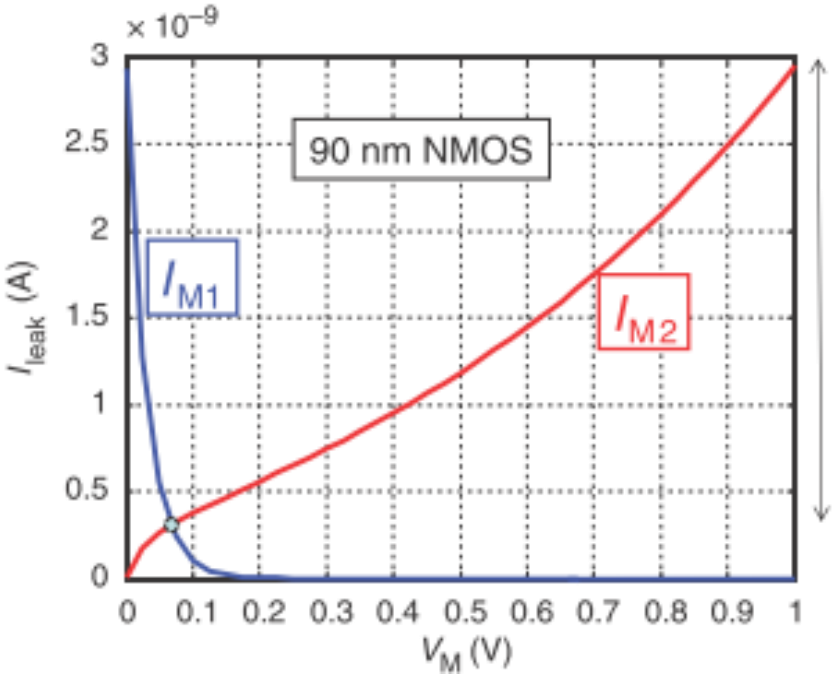
$$I_{leak,M1} = I_0 e^{\frac{-V_M - V_{th} + \lambda_{DS}(V_{dd} - V_M)}{v_t n}}$$

$$I_{leak,M2} = I_0 e^{\frac{-V_{th} + \lambda_{DS} V_M}{v_t n}}$$

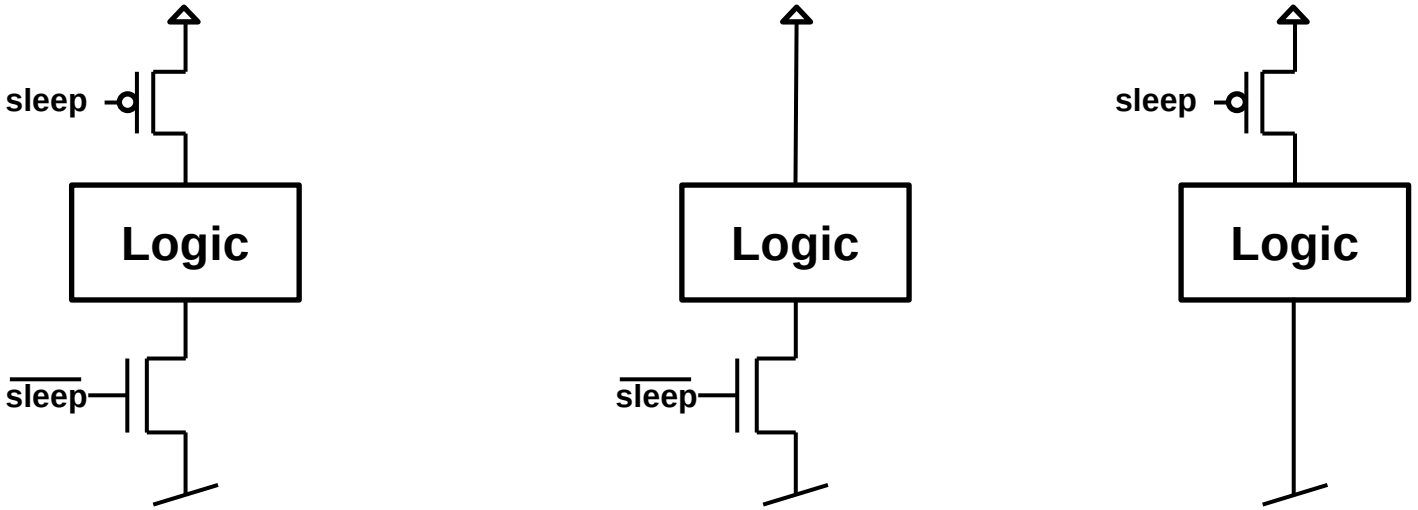
Leakage Reduction	
2 NMOS	9
3 NMOS	17
4 NMOS	24
2 PMOS	8
3 PMOS	12
4 PMOS	16

Small speed penalty: ~25%



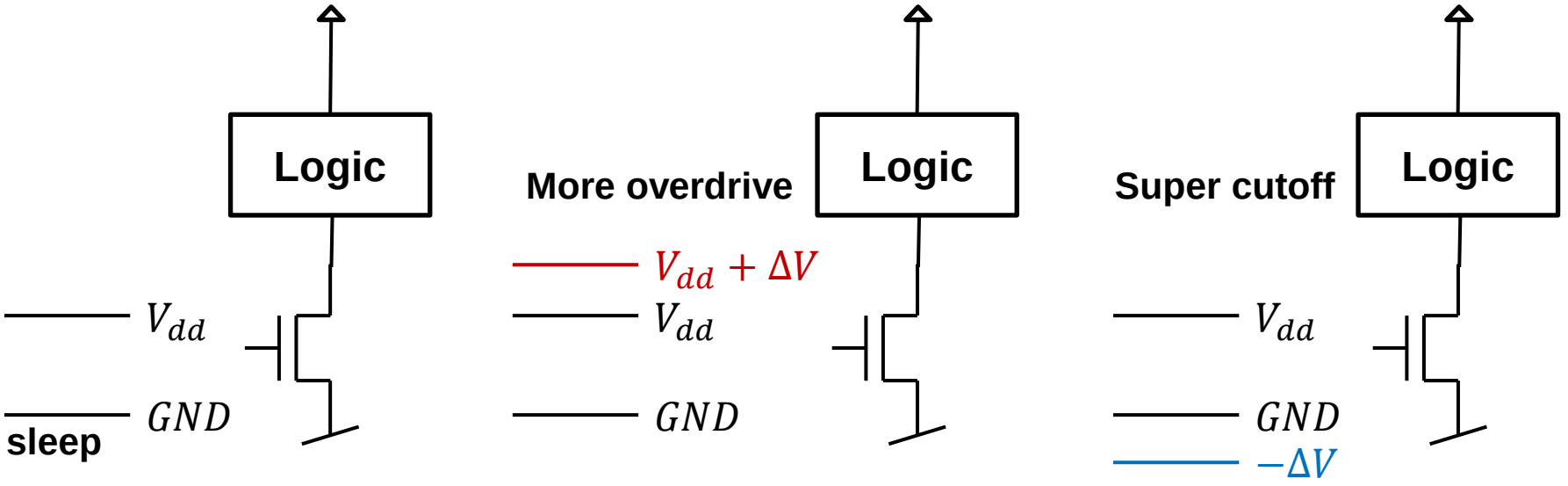


- Avoid leakage almost completely when individual design units are not used:
  - Disconnect entire modules from the supply with headers and/or footers)



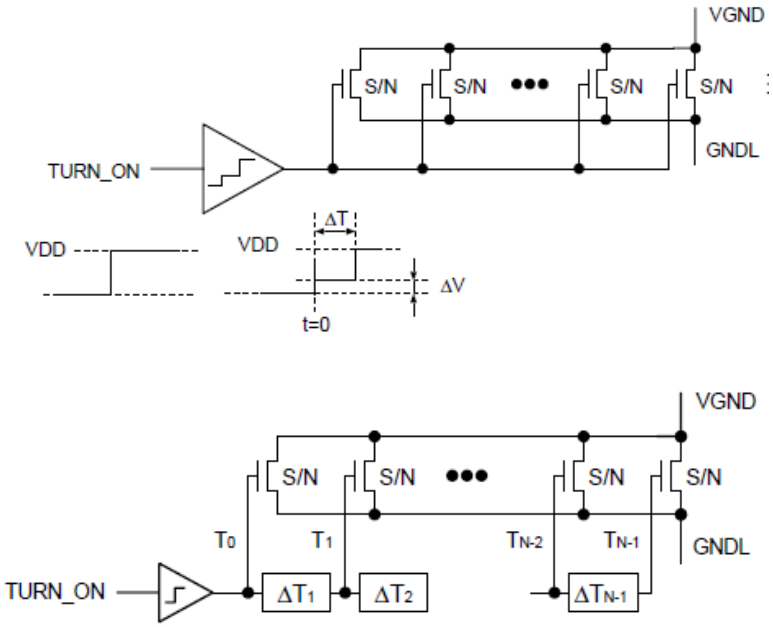
- Objectives with conflicting requirements
  - Sleep mode: large off-resistance to avoid leakage (stacking)
    - PMOS preferred over NMOS and HVT over LVT, header+footer
  - Active mode: minimize on-resistance to reduce negative impact on timing
    - Sleep transistors require large area
    - NMOS preferred over PMOS, LVT over HVT, footer-only

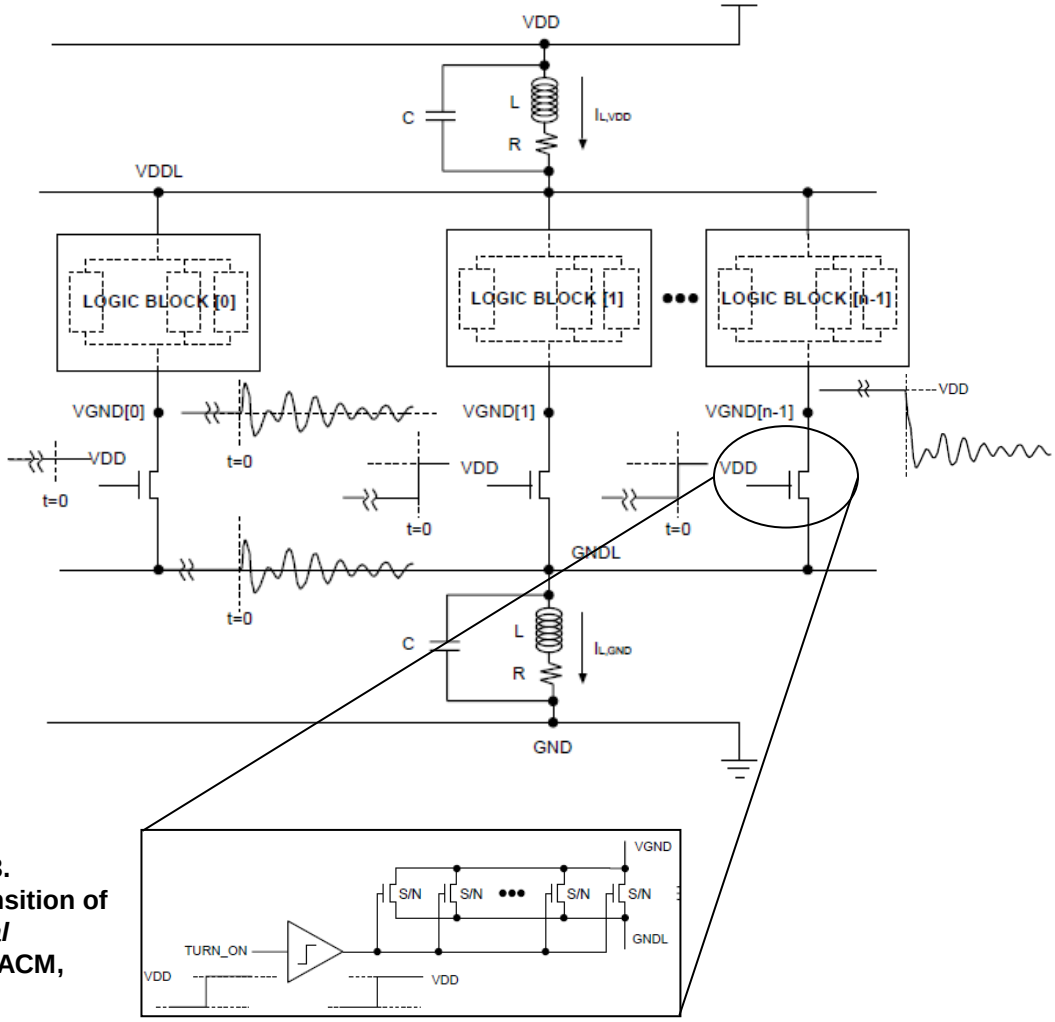
- **Improving performance of power switches**
  - Change gate-source voltage



Power Mode Transition

- **Rapid re-activation of a power gated block can cause large spikes on the supply network of the entire circuit**
- **Popular solutions:**



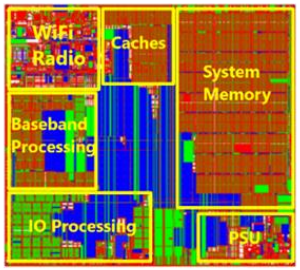
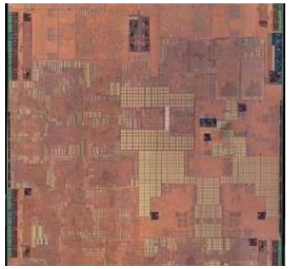
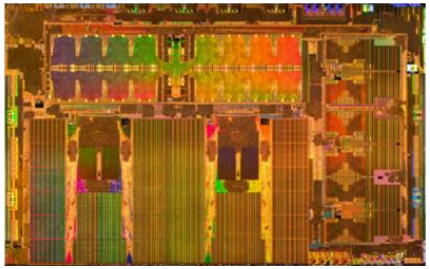
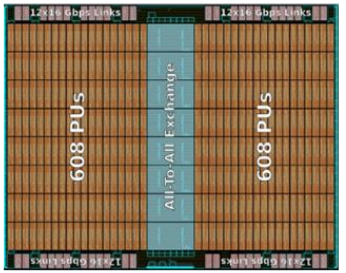


Suhwan Kim, Stephen V. Kosonocky, and Daniel R. Knebel. 2003. Understanding and minimizing ground bounce during mode transition of power gating structures. In *Proceedings of the 2003 international symposium on Low power electronics and design* (ISLPED '03). ACM, New York, NY, USA, 22-25. DOI=10.1145/871506.871515 http://doi.acm.org/10.1145/871506.871515

Low Power Memories

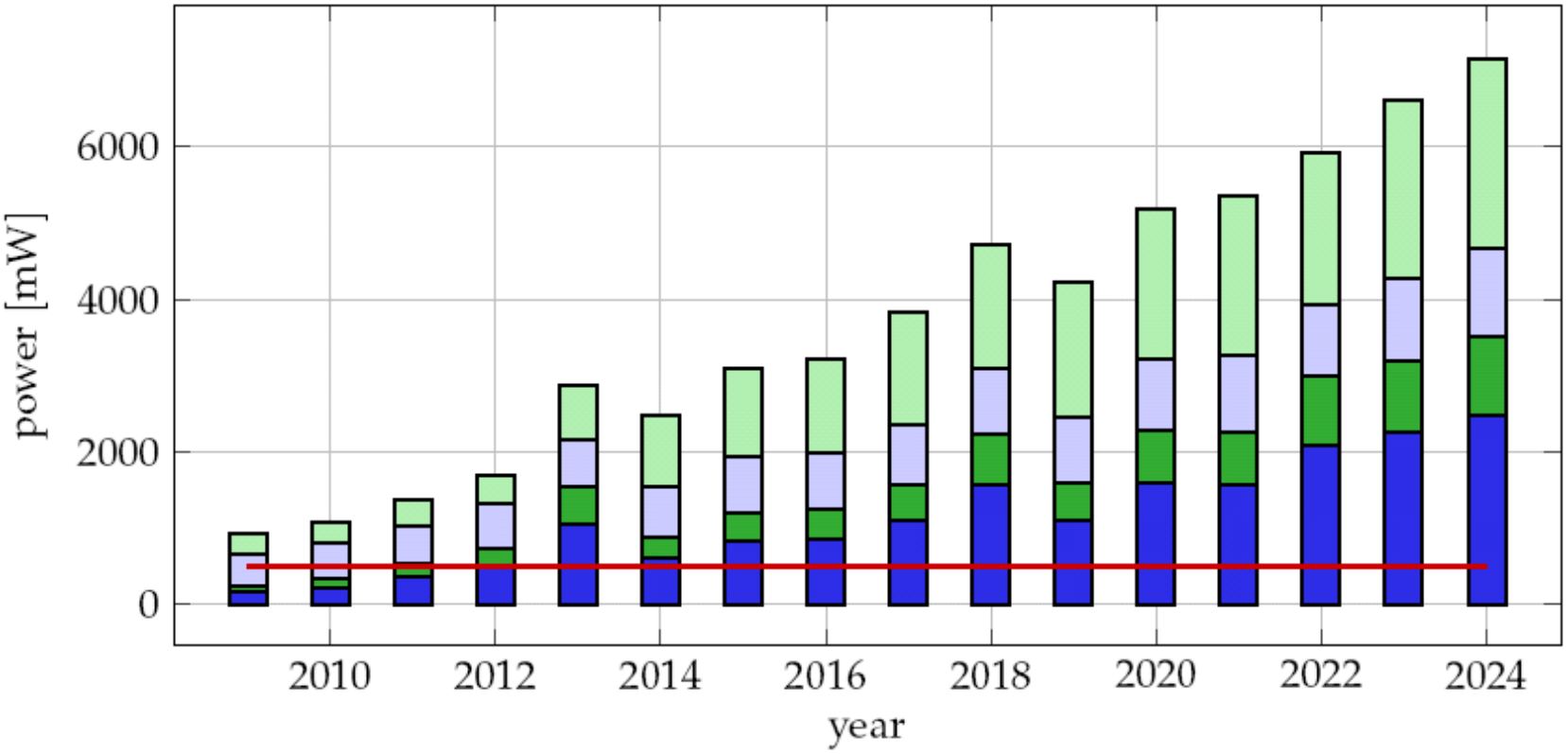
Memories are the limiting factor for cost and energy

- On-chip memories have a poor area density and often dominate chip area and cost in many computing systems
- Memory often accounts for >50% of the active power and for 100% of the power during sleep/standby periods in low-power systems

	IoT & MCU	Mobile	Automotive	ML/AI & Server
				
	MediaTek MT3620, 40nm	Apple A11, 10nm	Tesla FSD, 14nm	Graphcore IPU, 16nm
SRAM Area [%]	35%	31%	36%	75%

ITRS Roadmap Prediction on the Role of Memory

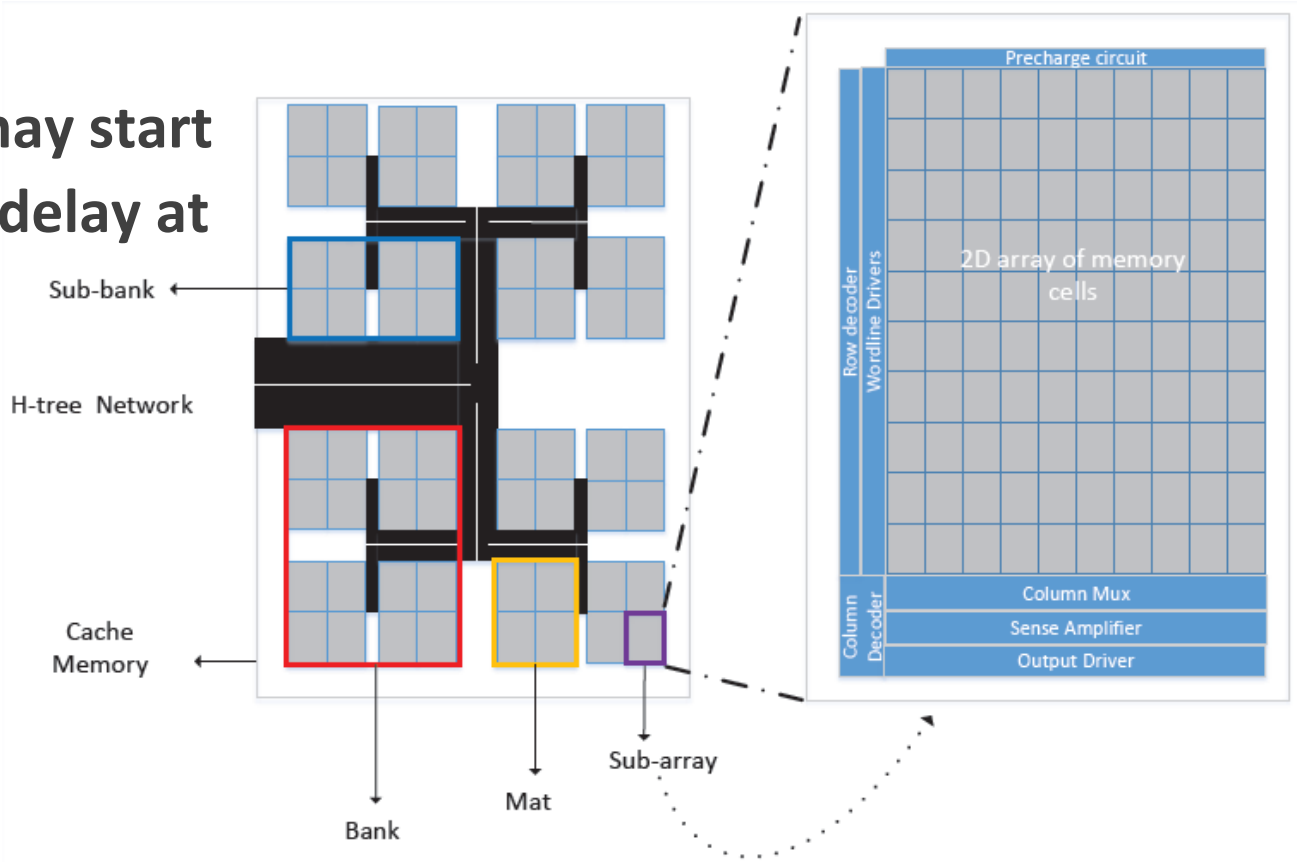
- Amount of memory (percentage) increases
- Leakage becomes increasingly relevant in modern technologies



Split large arrays into smaller sub-units (sub-banks and mats)

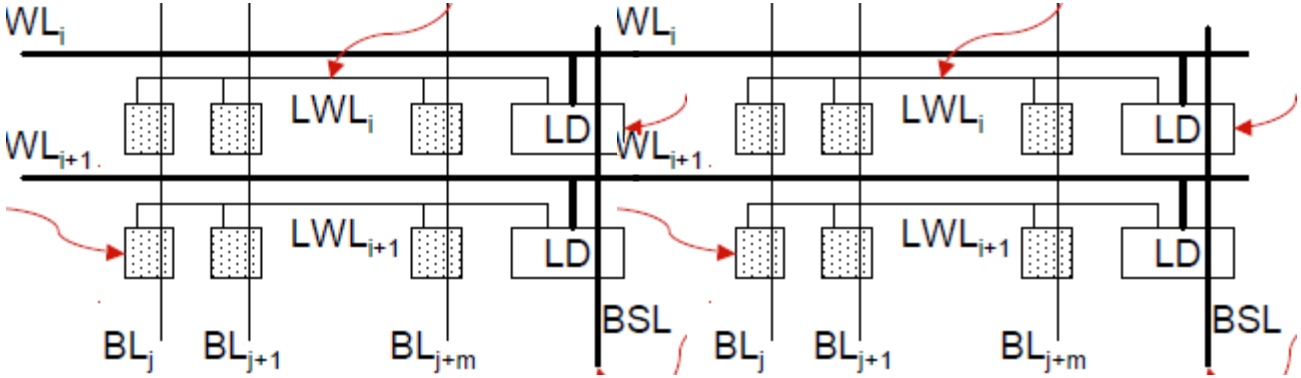
- Sorter word- and bit-lines (faster and less power)
- Can limit activation to only a relevant sub-array/mat

- Bank-decoding may start to dominate the delay at some point (2-8 banks)

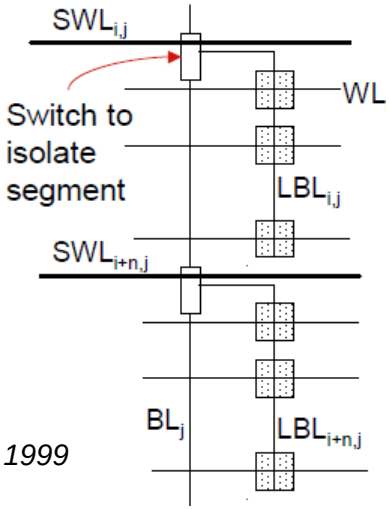


SRAM: Segmented (hierarchical) word-/bit-lines

- **Hierarchical word-line**
  - **Reduced capacitance on the word-line**
  - **Slower, due to delay of the local decoder**



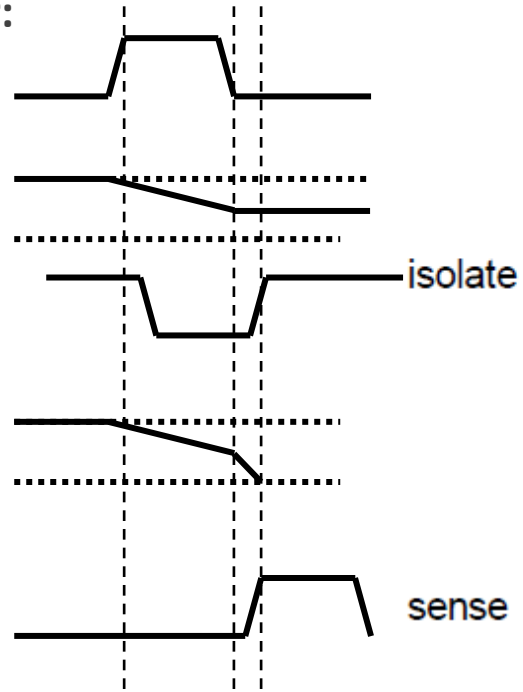
- **Segmented bit-line**
  - **Reduces bit-line capacitance (mostly from access transistor junction capacitance)**
    - **Enables smaller sense amplifiers**
    - **Better speed and lower power**

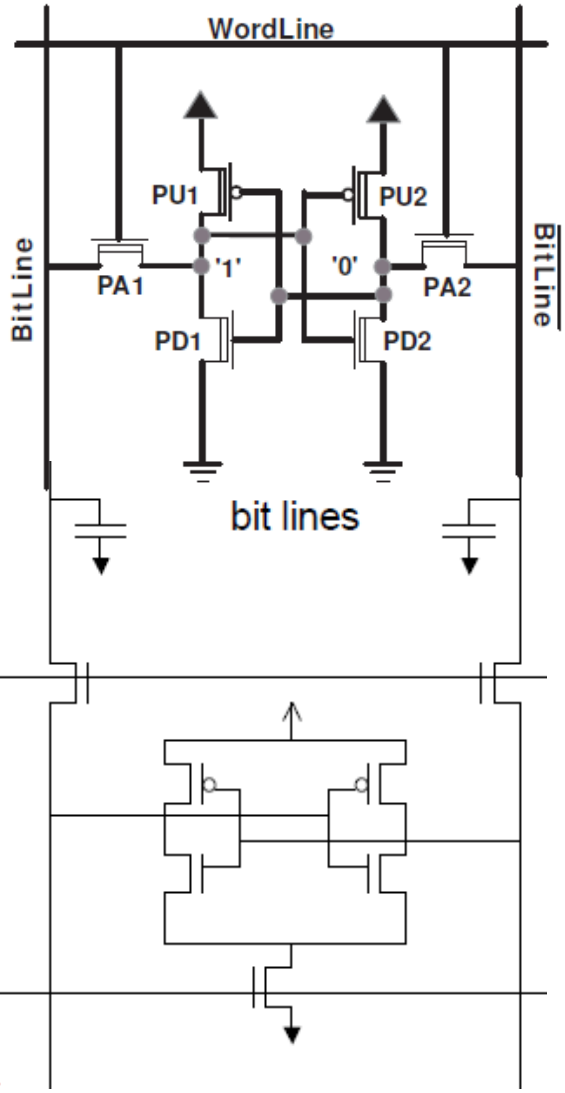


Margala, M., "Low-power SRAM circuit design," *Memory Technology, Design and Testing, 1999. Records of the 1999 IEEE International Workshop on* , vol., no., pp.115,122, 1999

SRAM: Pulsed Wordline & Bitline isolation / Clocked SA

- **Normal operation:**
  - Precharge of both bitlines to VDD
  - Word-line opens access transistor
  - One bit-line starts to discharge
  - Sufficient voltage difference: sense amplifier pulls one bit line up and one all the way down
- **Sub-optimal in terms of power: unnecessary discharging of one bit-line**
- **Pulsed scheme:**
  - Decouple bit-line from SA before activating feedback
  - Avoids complete BL discharge
  - BUT: isolate/sense timing becomes critical (sensitivity to variations)



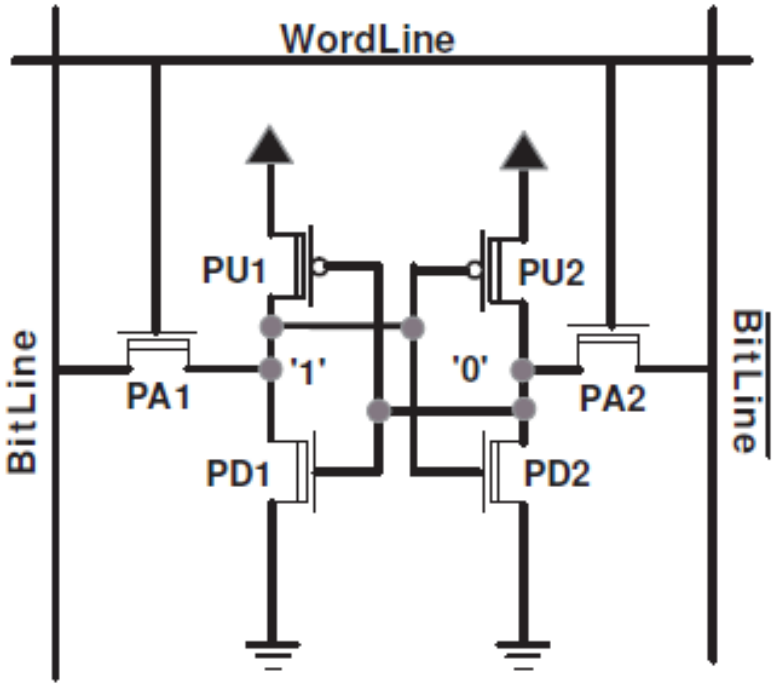


Standard 6T SRAM becomes unreliable at scaled voltages due to process variation

- **28nm technology: 6T SRAM functional down to 0.7V**

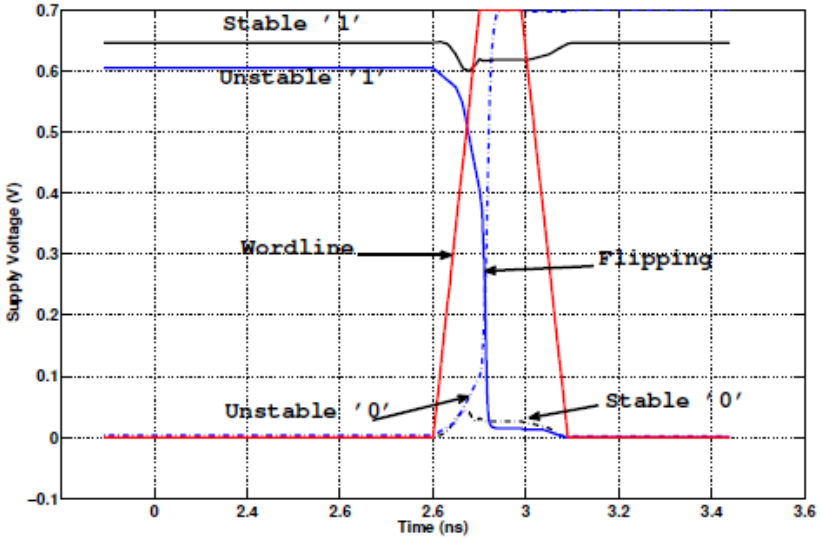
6T SRAM cells rely on rationed circuits

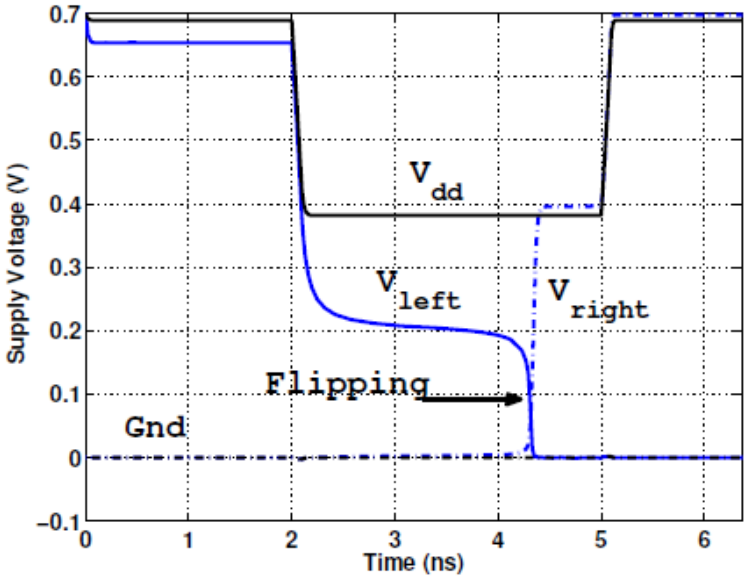
- **Mismatch between access and pull-up/down transistors during cell access**

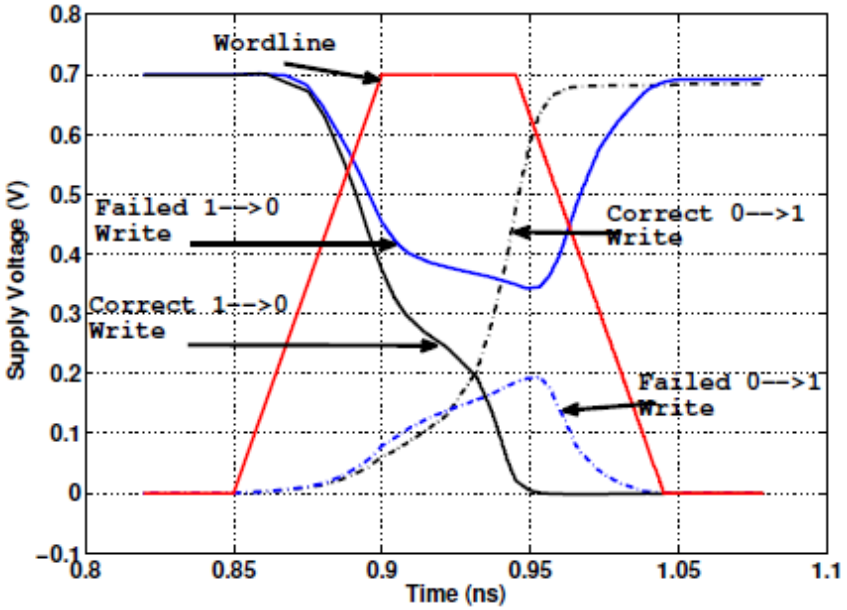


SRAM failure mechanisms at low voltages

- Write-failures: inability to write
- Access-failures: inability to read
- Read-failures: pre-charged bit-line flips cell content during read
- Hold-failure: content flips without access

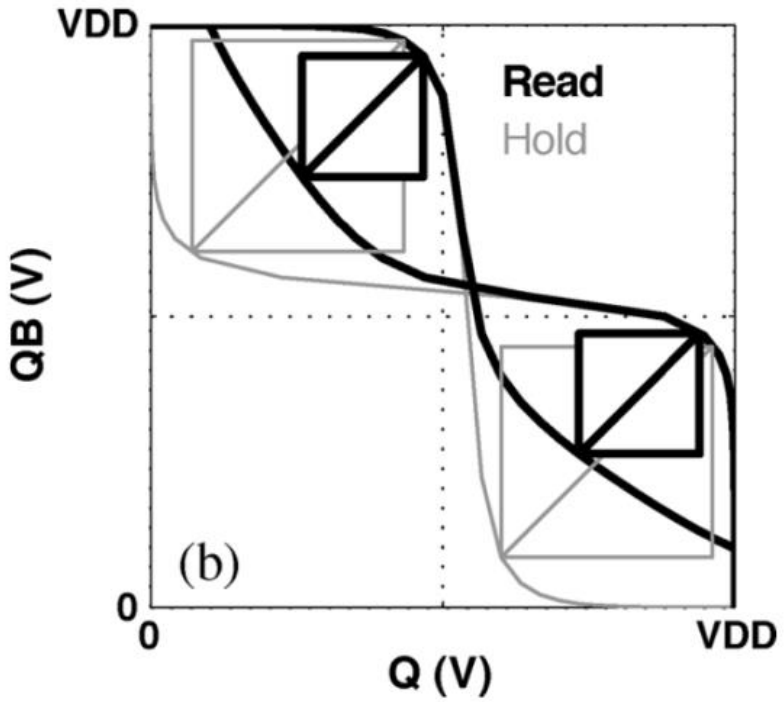


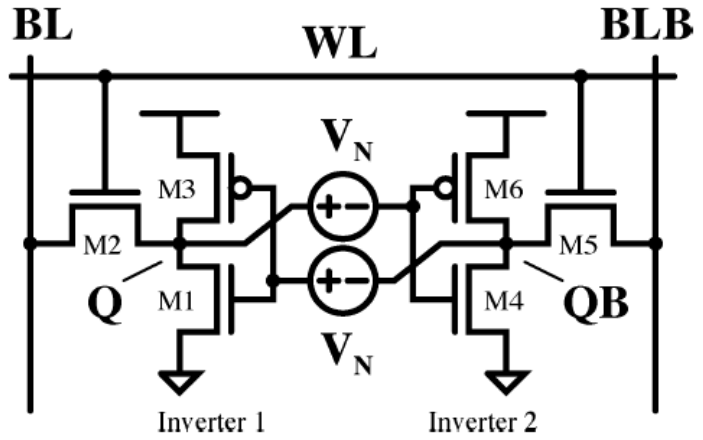


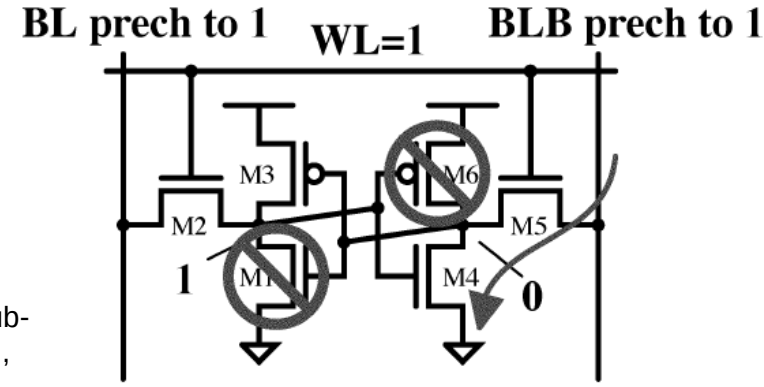


SRAm Stability Analysis at low Voltages: Static Noise Margin

- **Static noise margin (SNM): maximum amount of voltage noise that can be introduced at the output f the back-coupled inverter pair while maintaining stability**
  - **Draw transfer characteristics of the two inverters**
  - **Diagonal of the largest rectangle that can be embedded between the two curves**





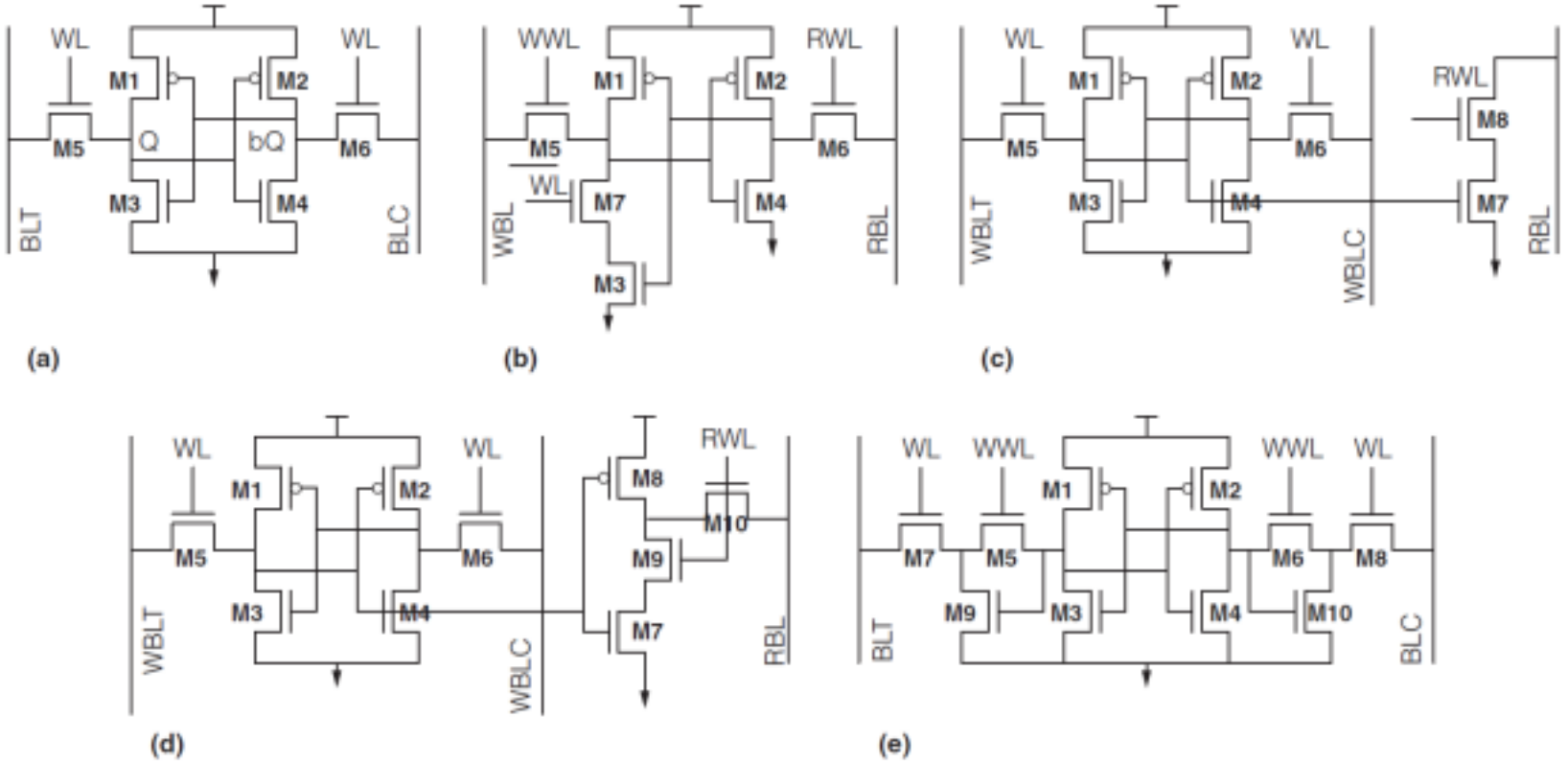


Calhoun, B.H.; Chandrakasan, A.P., "Static noise margin variation for sub-threshold SRAM in 65-nm CMOS," *Solid-State Circuits, IEEE Journal of* , vol.41, no.7, pp.1673,1679, July 2006

Alternative SRAM cells for low voltage operation

Objectives:

- **Isolate the read-path from the RBL to avoid read-failures**
- **Remove or weaken feedback during write operation**

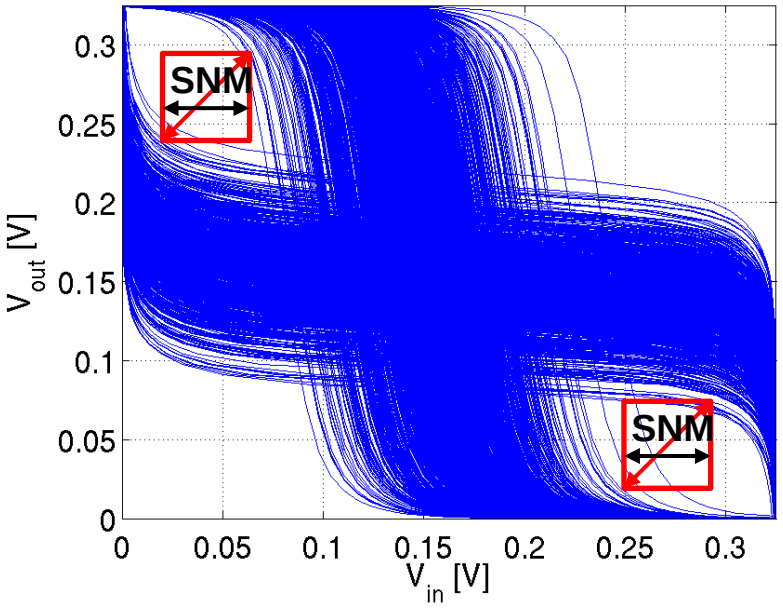


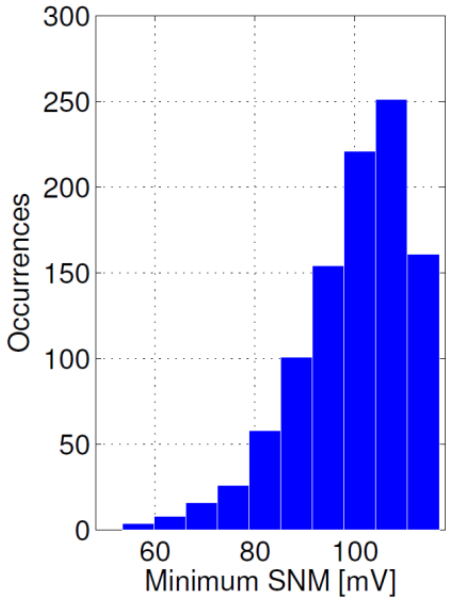
- **Generally: considerable area penalty**

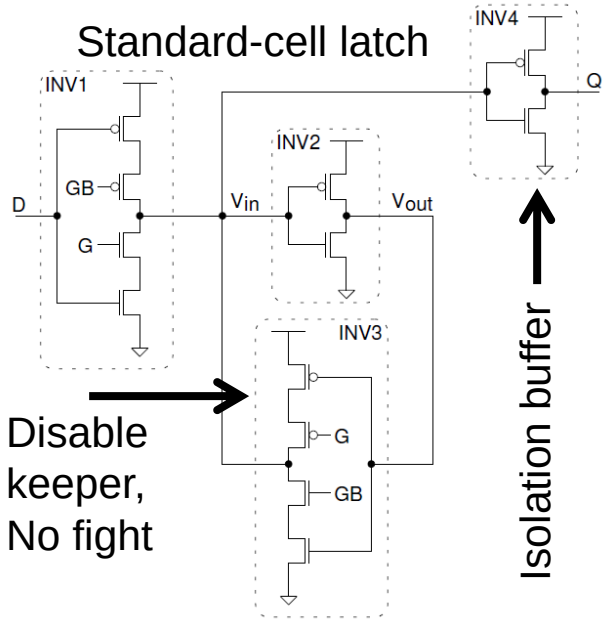
Latches: the perfect low-voltage SRAM cell

Most reliability issues of 6T SRAM are avoided by latch

- **Write failures: unusually strong keeper ↔ feedback disabled**
- **Read failures: degradation of SNM ↔ isolated output**
- **Hold failures: SRAM bitcell = latch in non-transparent phase with still good SNM at VDD=300mV**







Meinerzhagen *et al*., JETCAS'11; [1] Calhoun *et al*., JSSC'05; [2] Calhoun *et al*., JSSC'07

Standard-Cell based Memories (SCMs)

Assemble small memories from standard-cells

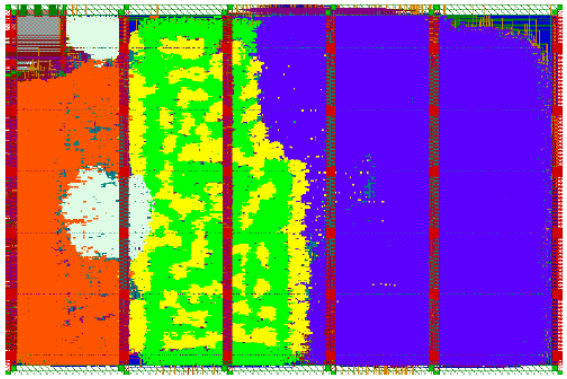
- **Robust against voltage scaling**
- **Inherently low power**

Advantages beyond robustness and power

- Generic description in any HDL
- Any desired size is possible
- Modifications at design time
- Portability (unless custom cells)
- Fine-granular organizations
- **Automatic or guided placement**
- Merge with logic (where appropriate)
- Avoid power routing

Main drawback

- **Area (if storage capacity > 1kb)**



Meinerzhagen *et al*., MWSCAS'10
Roth, Meinerzhagen *et al*., A-SSCC'10

Teman et al.,
ASP-DAC 2015

SCMs best practice

Write Logic

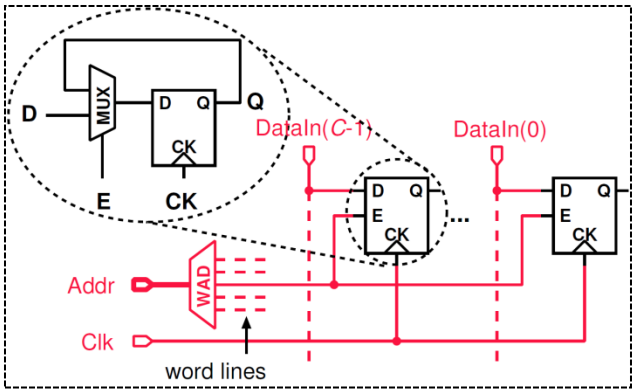
- *Clock-gates (b):* smaller and less power than *enable flip-flops (a)*

Read Logic

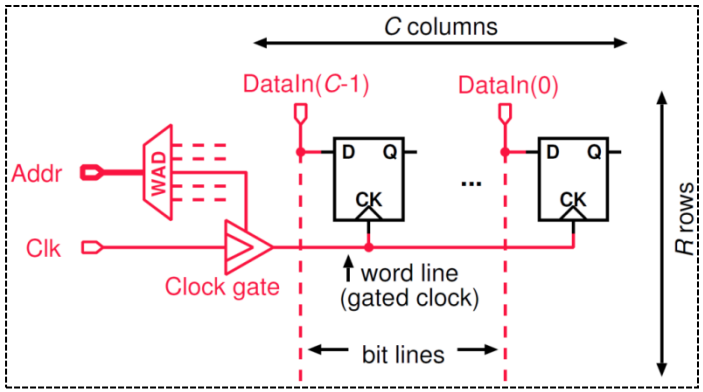
- Above-VT
  - ***Multiplexers (c)*: smaller, faster, and less power than tri-state buffers**
- Sub-VT
  - ***Tri-state buffers (d)*: less leakage (energy) than multiplexers**

Array of Storage Cells

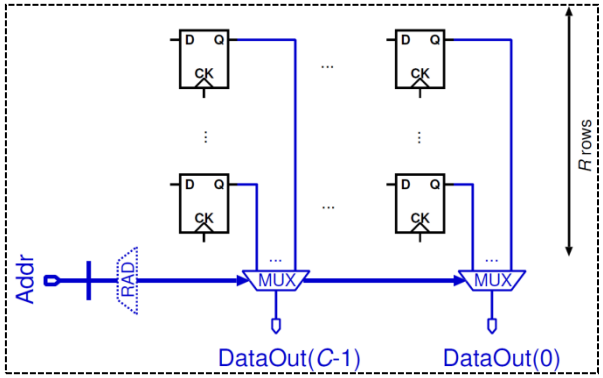
- *Latch* arrays smaller than *flip-flop* arrays, but longer write-address setup time



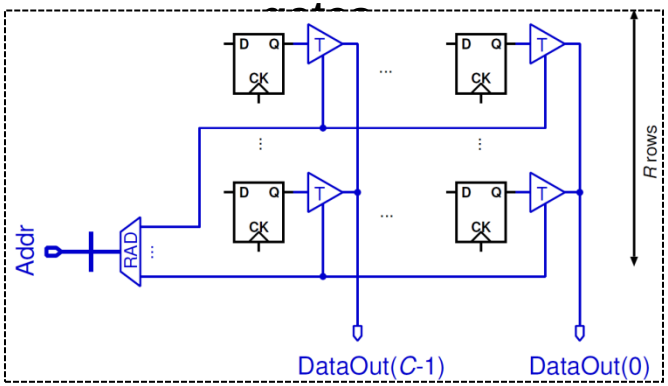
(a) Enable flip-flops



(b) Clock



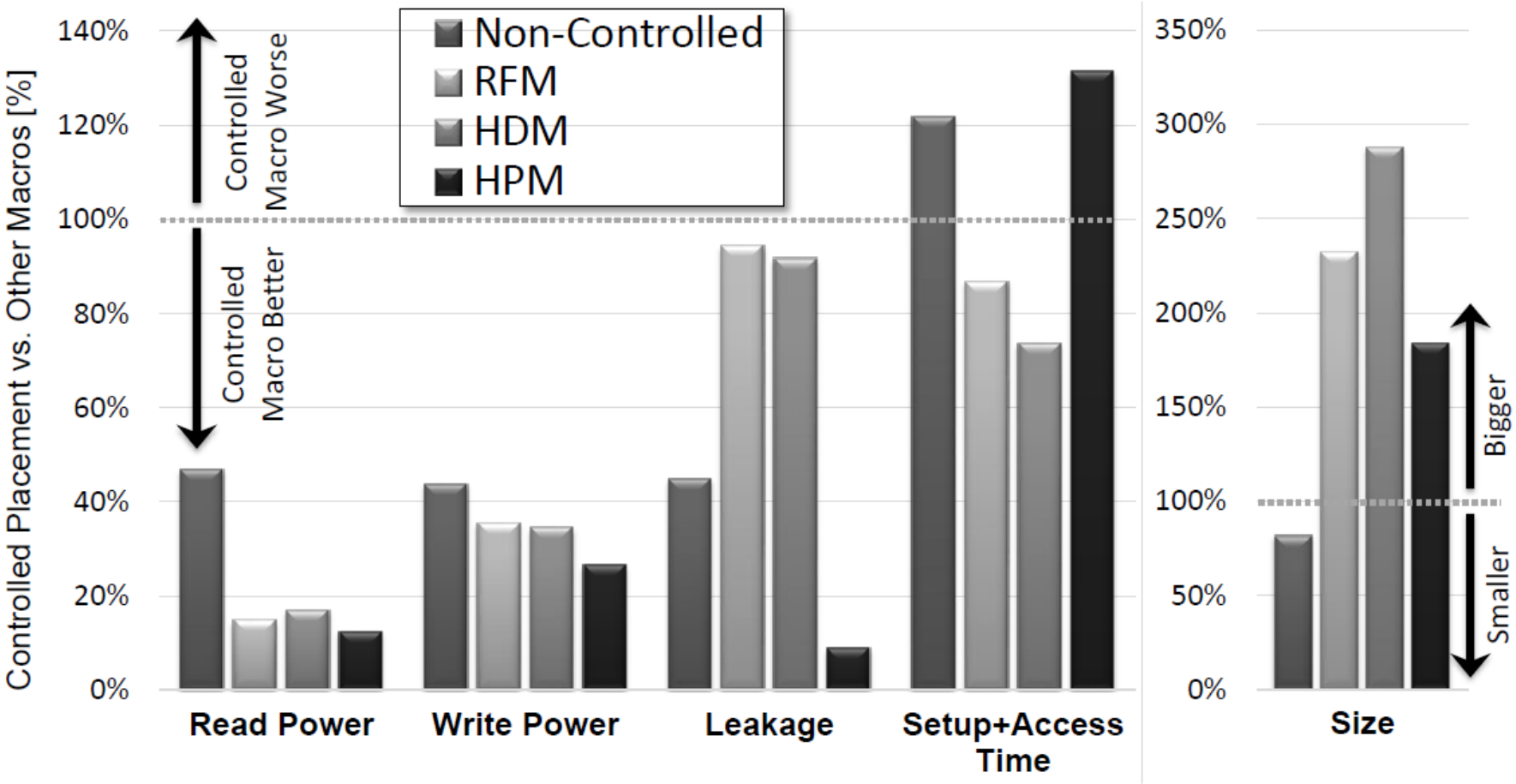
(c) Multiplexers



(d) Tri-state buffers

SCMs Outperform 6T SRAM Macros und nominal Voltages

- Lower read- and write-energy
- Lower leakage power
- Sometimes even better access speed

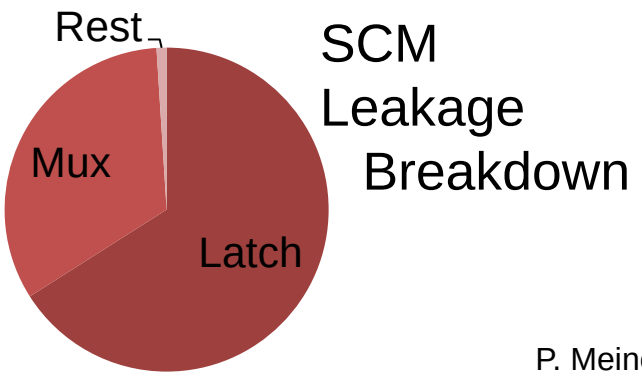


Sub-V_T SCM: Insights and Leakage Breakdown

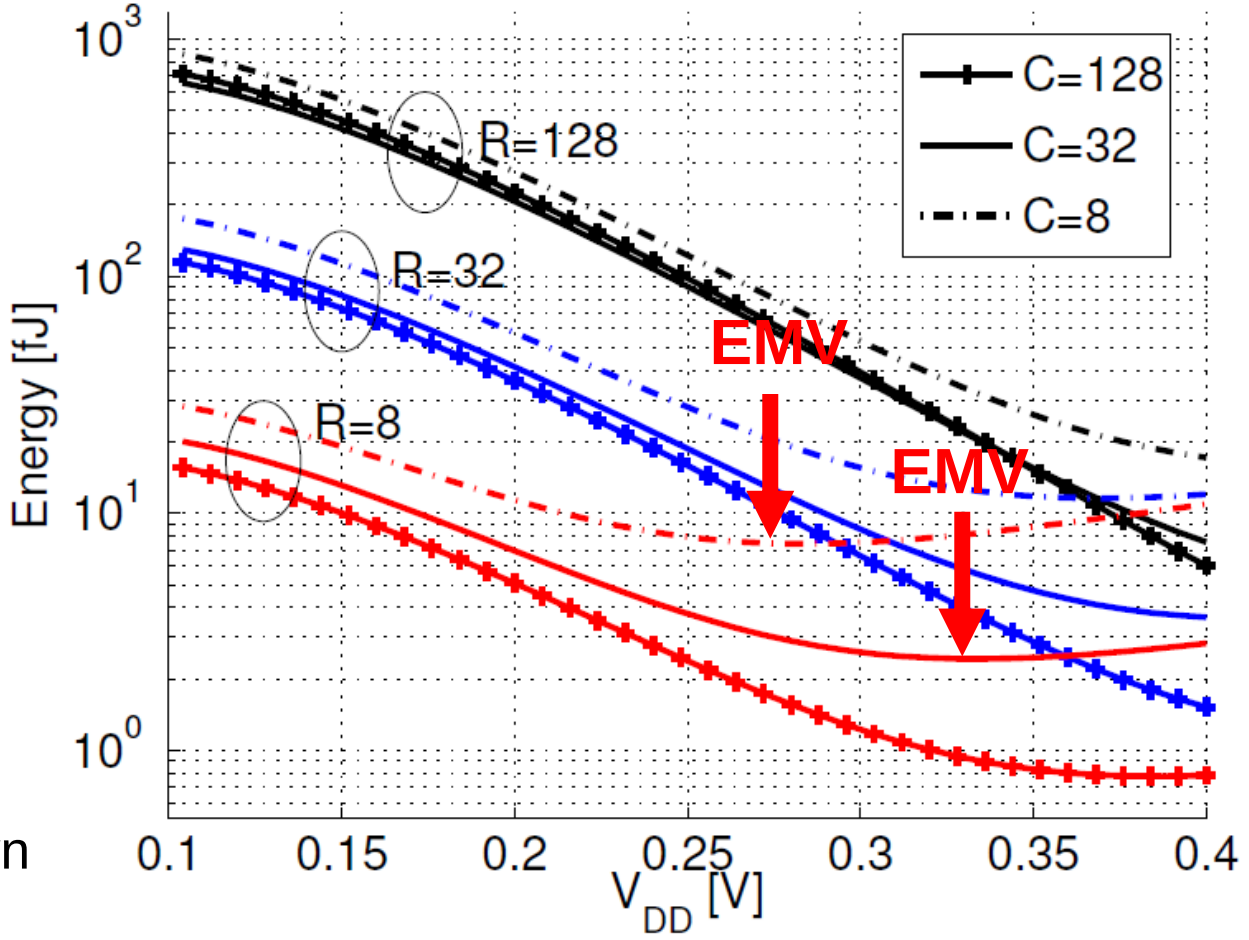
Large memory arrays: little switching activity

- **Total energy is dominated by leakage**
- **Active energy negligible, except for smallest SCMs**
- **Only smallest SCMs reach EMV in sub-V_T domain**

→ Minimize leakage!



SCM Leakage Breakdown



P. Meinerzhagen *et al.*, JETCAS'11

Custom Cell: Low-Leakage Latch with Tri-State Output

Best practice for low leakage

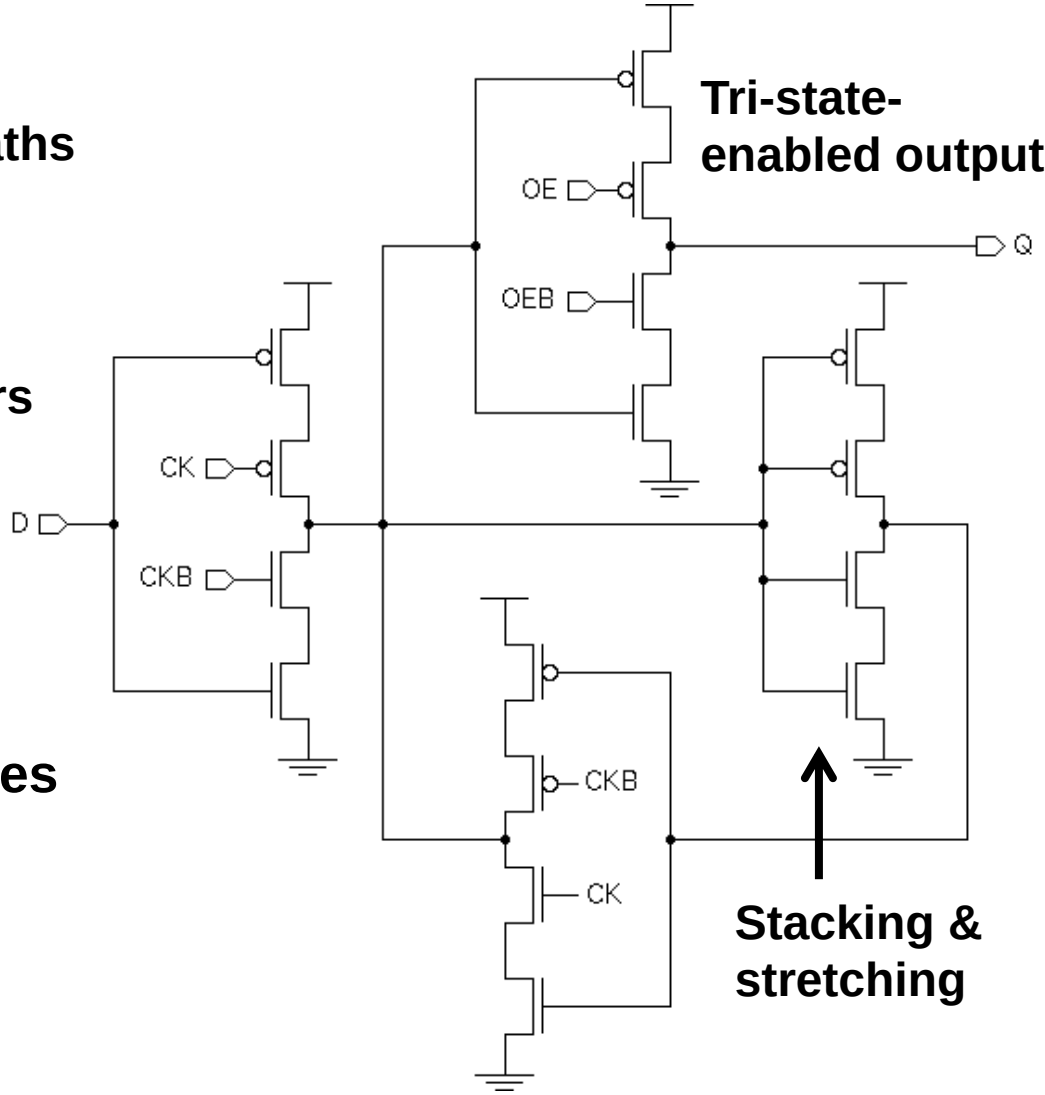
1. **Lowest number of V_{DD}-ground paths**
2. **Highest resistance on each such path**

- **Tri-state buffers**
- **Stacking & stretching for inverters**

Stacking factor: max 2

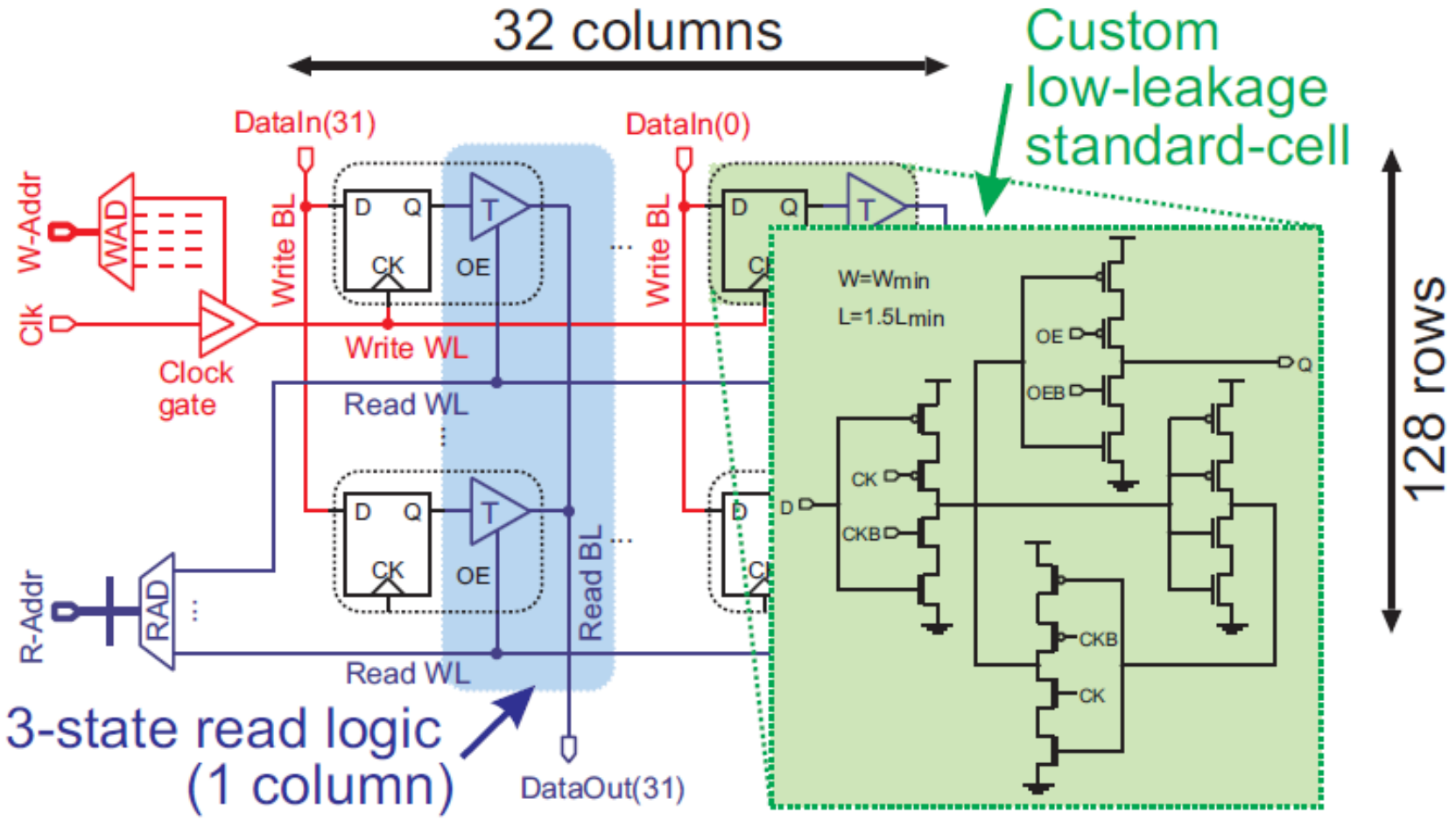
Channel length stretching: $2L_{min}$

Convert output buffer to tri-state buffer to avoid static CMOS muxes



Test Chip: Architecture of Low-Leakage 4kb SCM

- **Write logic uses clock-gates**
- **3-state buffers used for read operation are integrated in low-leakage latch**



Meinerzhagen *et al.*, ESSCIRC 2012

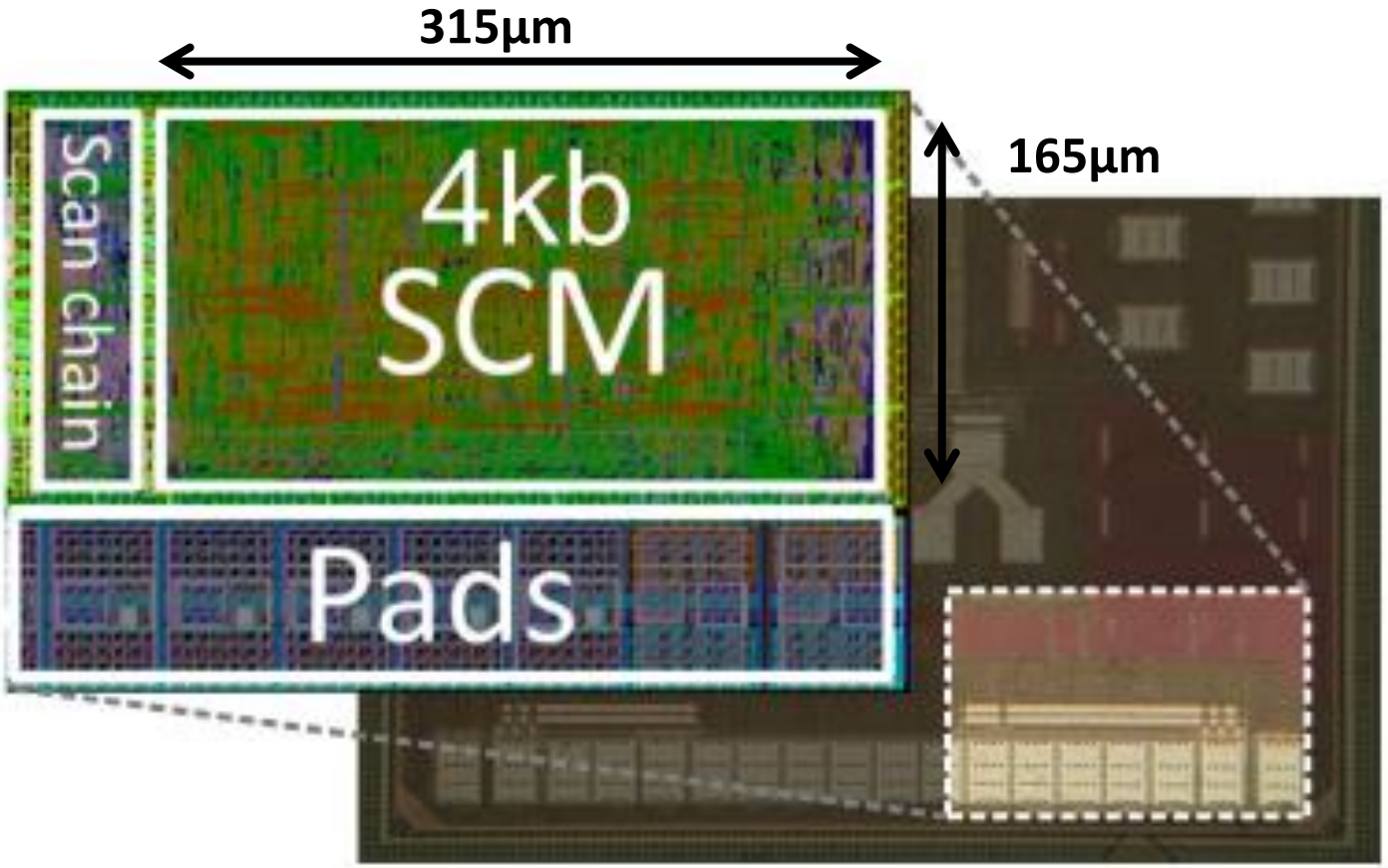
Chip microphotograph and zoomed-in layout picture

Area cost of 12.7 µm² per bit (including peripherals)

Scan-chain test interface

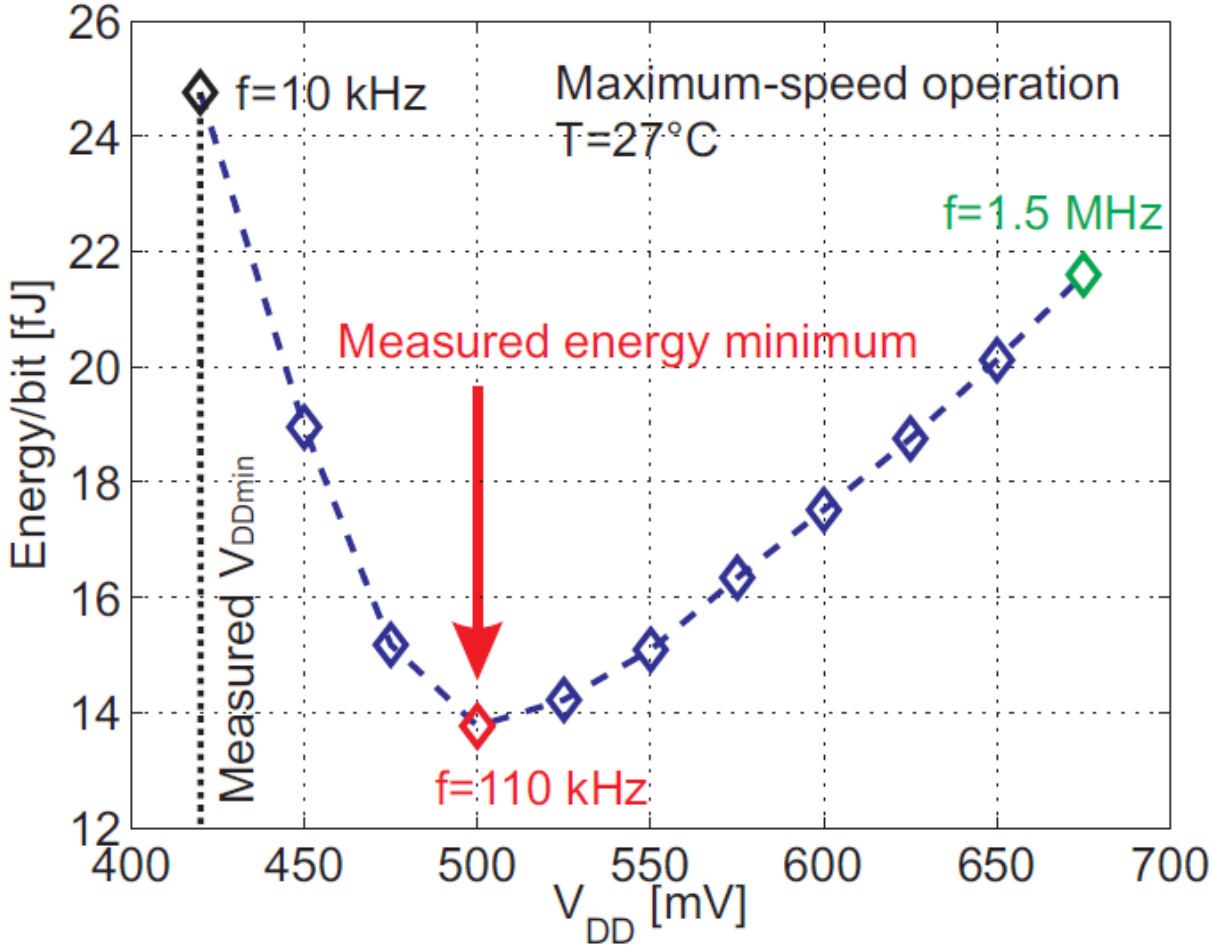
Functionality verification:
W/R random and checker-board patterns

Oven to control temperature:
27 or 37°C



Measured energy per bit-access performed at maximum speed

Measured energy minimum is 14fJ/bit at 500mV, 110kHz



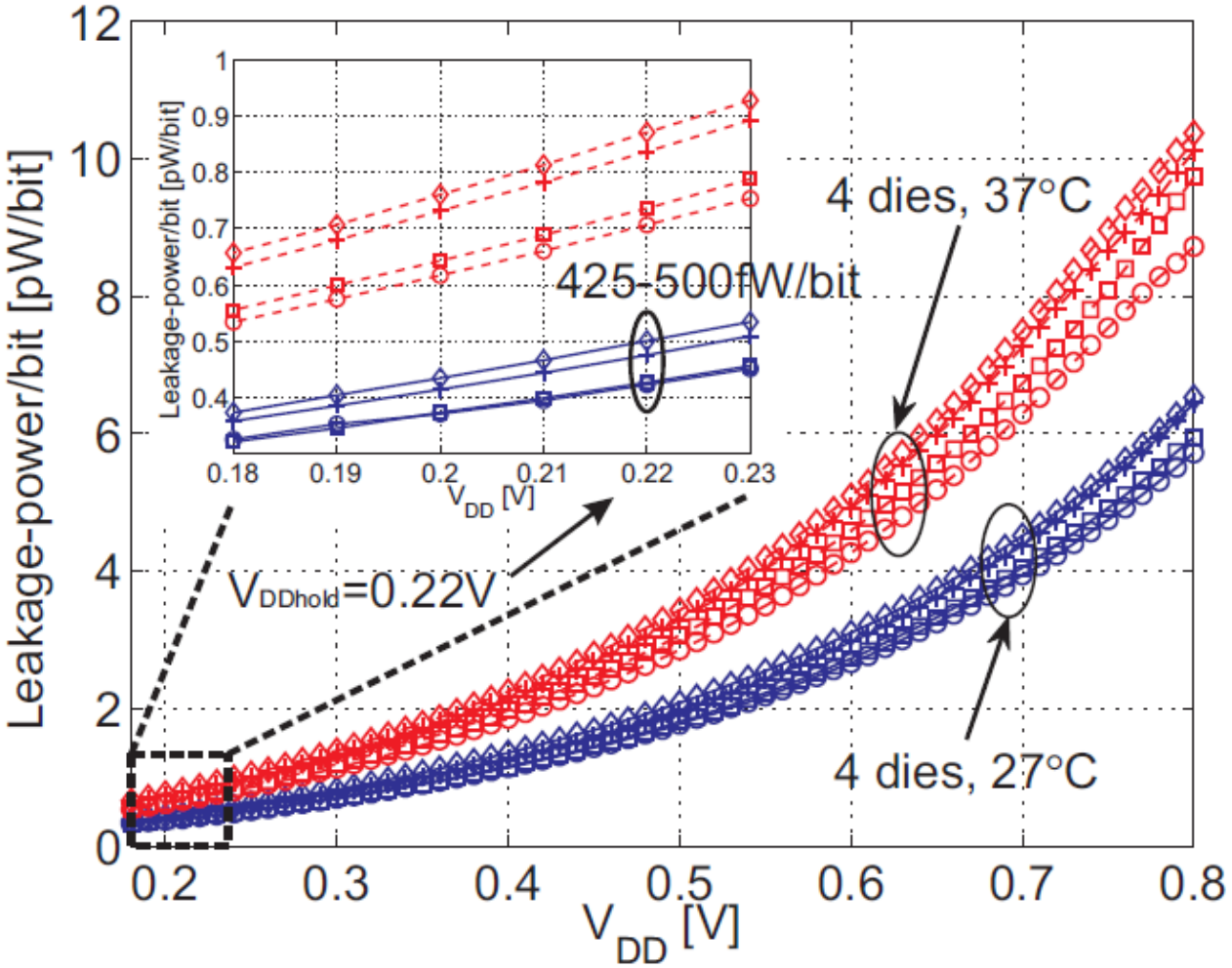
Silicon Measurements: Leakage Power is 500fW/bit

At VDDhold=220mV, data is correctly held with a leakage power of 425-500fW per bit (best and worst out of 4 measured dies)

At 37°C (typical for biomedical implants)

- VDDmin=400mV (instead of 420mV at 27°C)
- Maximum operating frequency doubles
- But: higher leakage power

- Low retention voltage is key for low power



Comparison with Prior-Art Sub-V_T Memories

Benefits of designing 1 custom standard cell

- **Leakage power reduced by 50% (at no area increase) w.r.t. commercial standard cell latch [Meinerzhagen et al., JETCAS'11]**

Considered work: Full macro, measured, 65nm node

	[1]	[2]	[3]	[4]	This work
V_{DDmin} [mV]	380	250	700	350	**420**
V_{DDhold} [mV]	230	250	500	250	**220**
$E_{tot/bit}$ [fJ/bit]	54 (0.4V)	86 (0.4V)	-	55	**14 (0.5V)**
$P_{leak/bit}$ [pW/bit]	7.6 (0.3V)	6.1	6.0, 1.0[a]	-	**0.5**

[a] Leakage-power of bitcell only

[1] MIT: Calhoun and Chandrakasan, JSSC 2007; [2] MIT: Sinangil, Verma, and Chandrakasan, JSSC 2009; [3] Intel CRL: Wang et al., JSSC 2008; [4] STM: Clerc et al., ESSCIRC 2012

- **Lowest leakage-power/bit ever reported in 65nm CMOS**
- **Lowest active energy/bit-access ever reported in 65nm CMOS**
- **Reduce leakage in array and periphery!**

ReRAM-based Non-Volatile Flip-Flops

Goal:

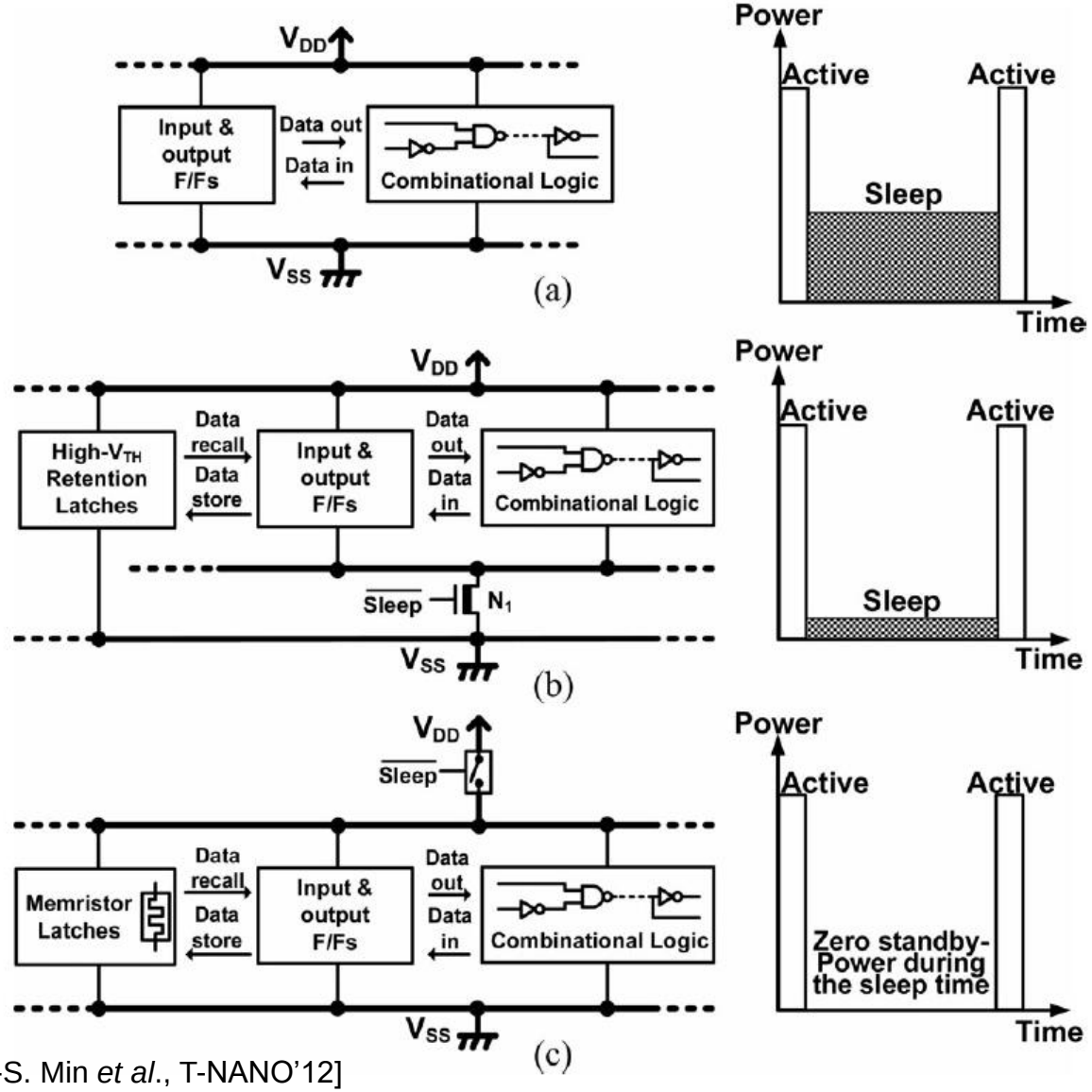
- **Instant-on computers**
- **Low-, or zero-leakage sleep states**

Conventional approach:

- **Power gating**
- **Low-leakage (high-VT) state retention latches on separate supply**

Future trend: non-volatile FFs based on new materials

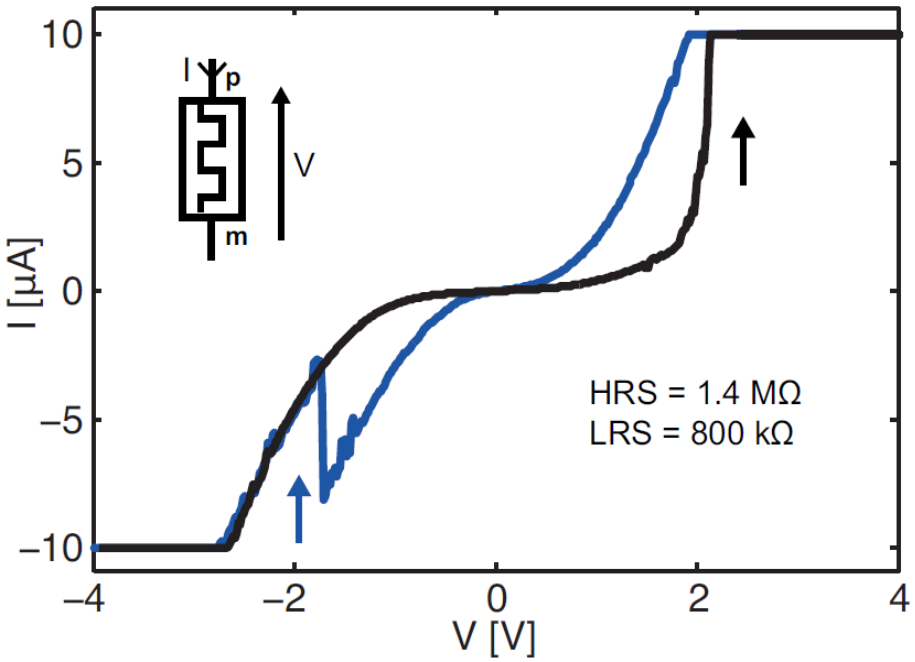
- **Exploit ReRAM technology**
- **Integrate it on top of CMOS chips**



[C.-M. Jung, K.-S. Min *et al.*, T-NANO'12]

Resistive memories / memristors: based on new materials that change and keep their resistance even when power is off

- **Based on a new material that can be deposited on top of standard CMOS process**



Read

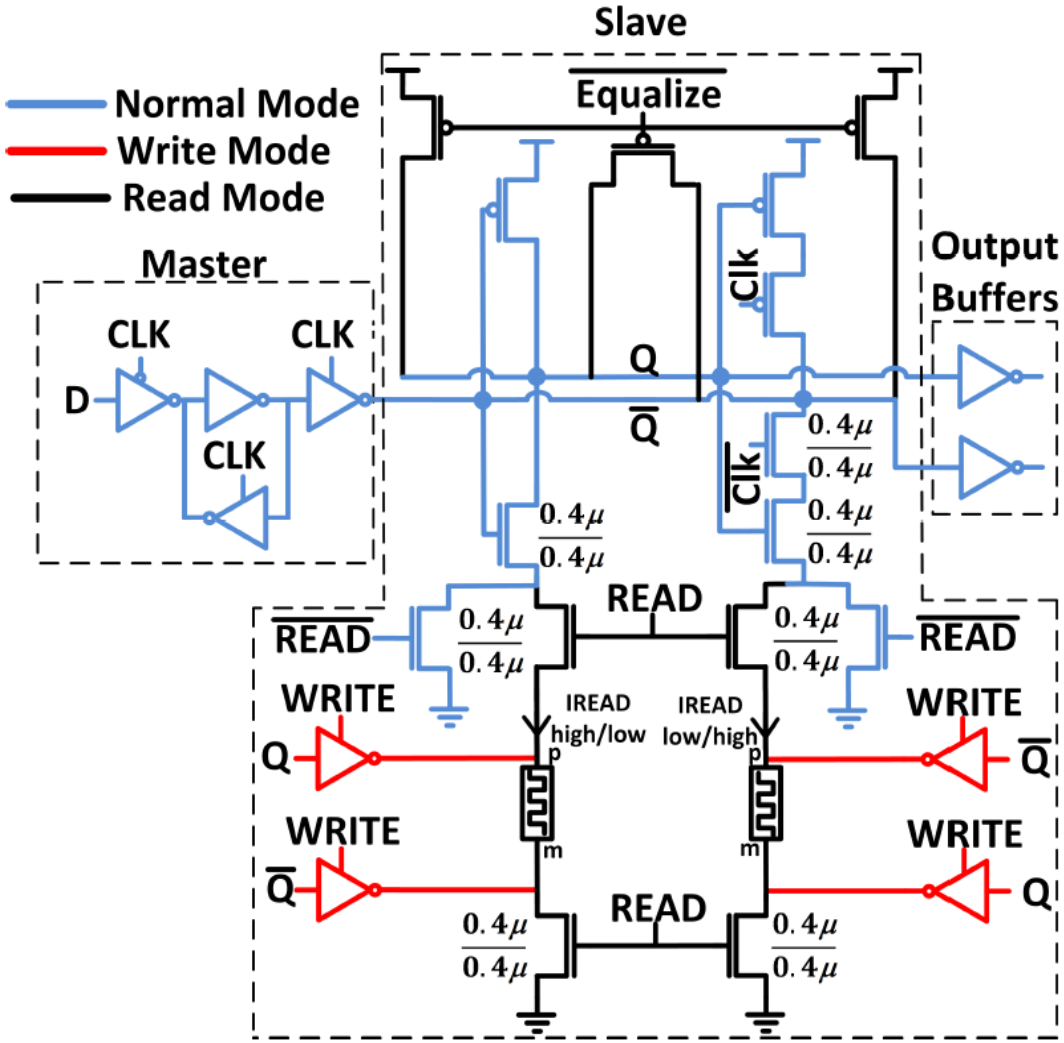
- **Based on resistance value**
- **Low current/power/voltage**

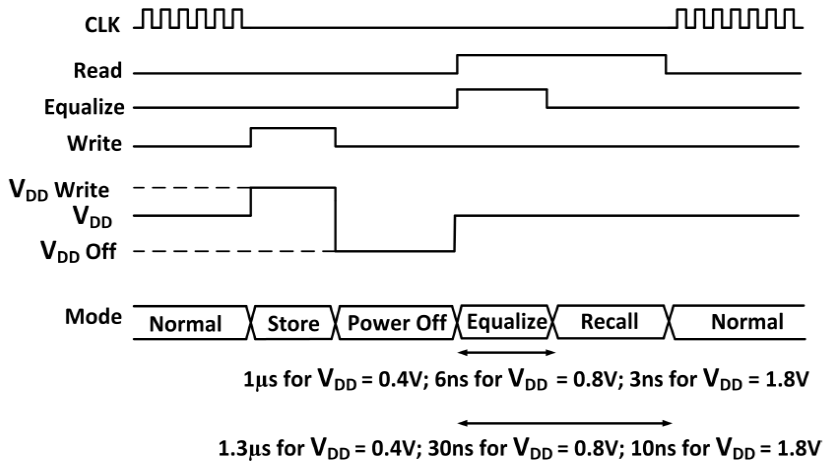
Write:

- **Switching of OxRAM resistor**
- **10μA current compliance**
- **$\pm$2V pulse for programming**
- **Needs relatively high power/current & voltages**

Fig. 1. Al/TiO_2/Al ReRAM stack switching under 10 µA current compliance.

Design for Nominal Voltage Operation (180nm, 1.8-2V)





Normal operation

- Conventional slave-latch

Transition to sleep

- Write input to ReRAM
- Requires high nominal voltage of 1.8-2V

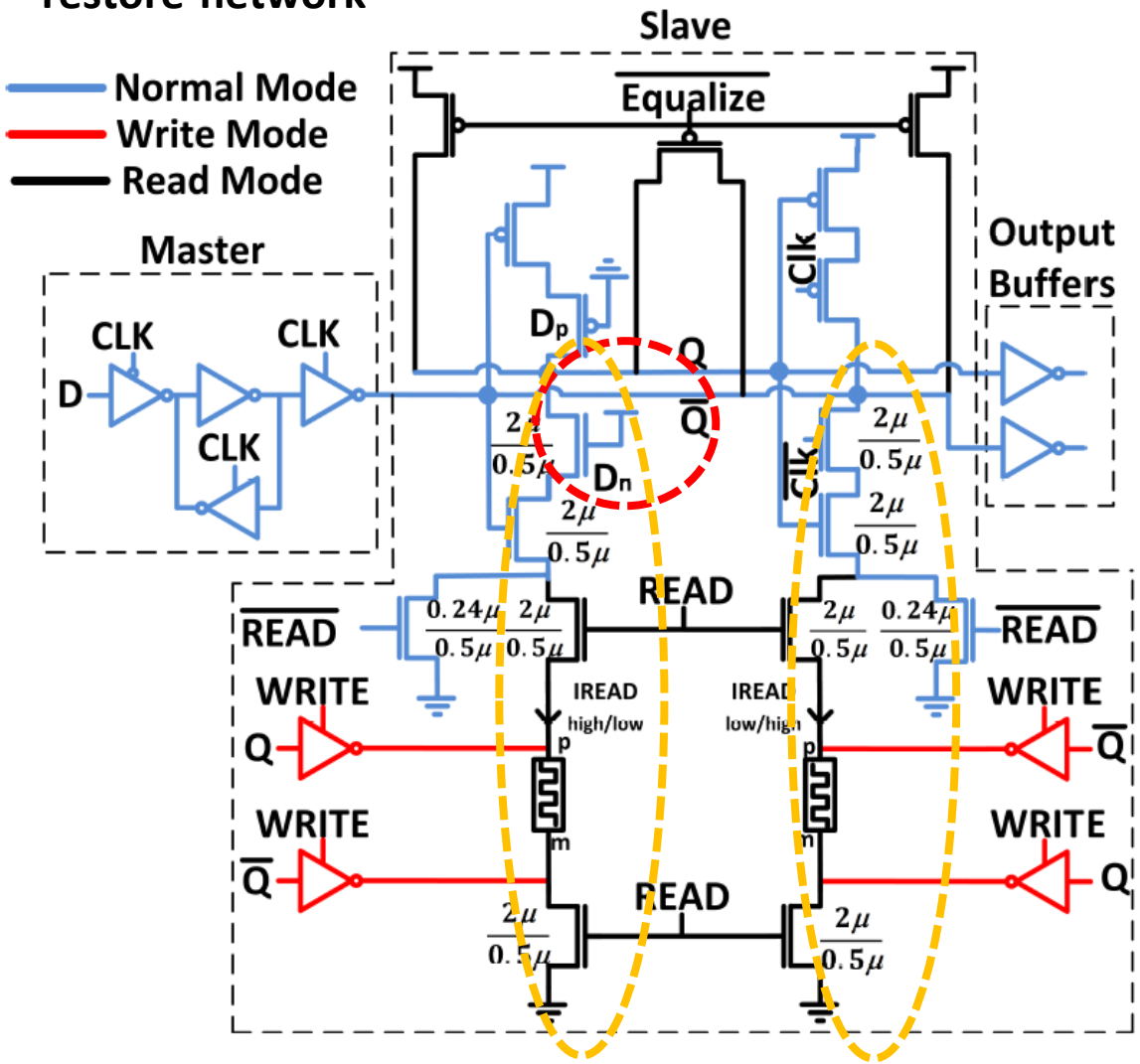
Wakeup

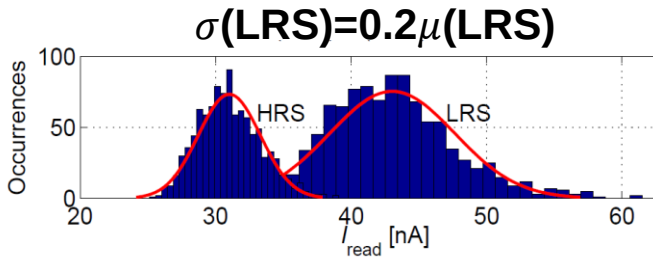
- Restore state from ReRAM into regular latch
- Pre-charge / evaluate

I. Kazi, P. Meinerzhagen, P.-E. Gaillardon *et al.*, EPFL

Design for Low-Voltage or Sub-V_T Operation

Modified circuit for better matching of differential restore-network





Normal operation

- Conventional slave-latch at scaled (low) VDD

Transition to sleep:

- Write input to ReRAM
- Requires separate high-VDD voltage of 1.8-2V -> **expensive in terms of power**

Wakeup

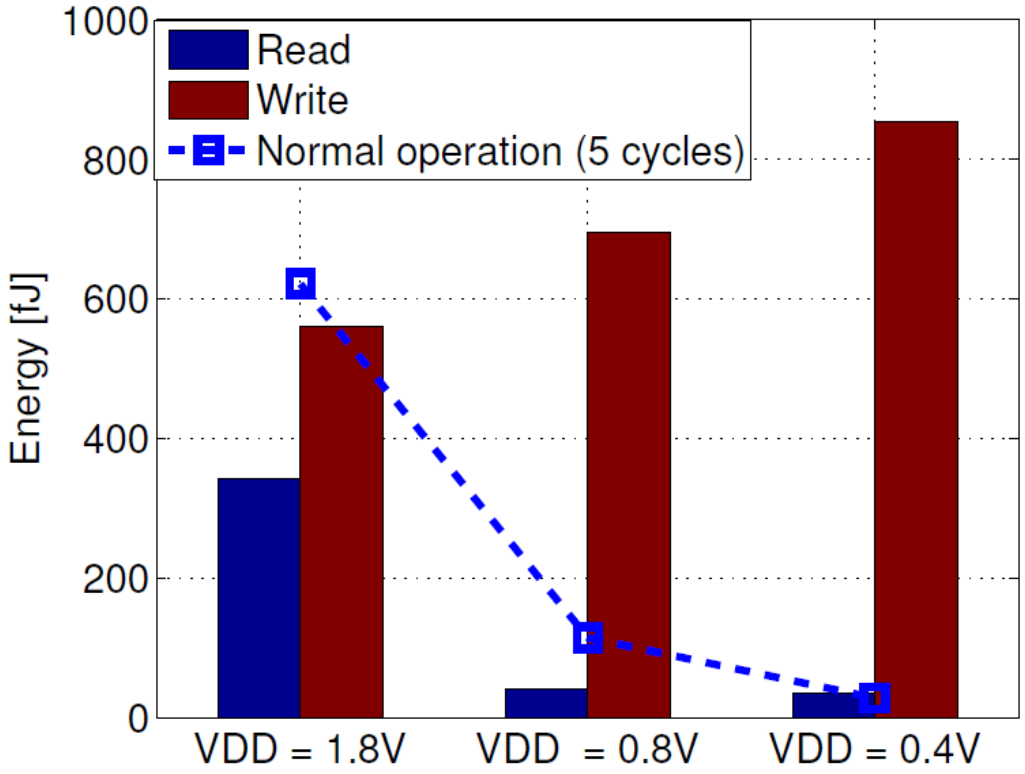
- Restore state from ReRAM into regular latch
- Pre-charge / evaluate

I. Kazi, P. Meinerzhagen, P.-E. Gaillardon *et al.*, EPFL

Energy Characterization: Near-V_T Minimum-Energy Point

Transition to and from sleep mode (write/read) incur significant overhead, especially when operating at low voltages in active mode

- **Longer read pulse if entering sub-V_T regime → energy savings saturate**
- **Energy cost to rise V_{DD} to 1.8-2V for write increase**



Need for high voltages/current for writing offsets advantage of ReRAM over low-leakage FFs without power gating

- **Minimum total read+write energy is 735fJ, found at 0.8V**
- **1.47s break-even compared to ULV ultra-low leakage latch [P. Meinerzhagen et al., ESSCIRC'12]**
- **Effective only for long sleep periods**

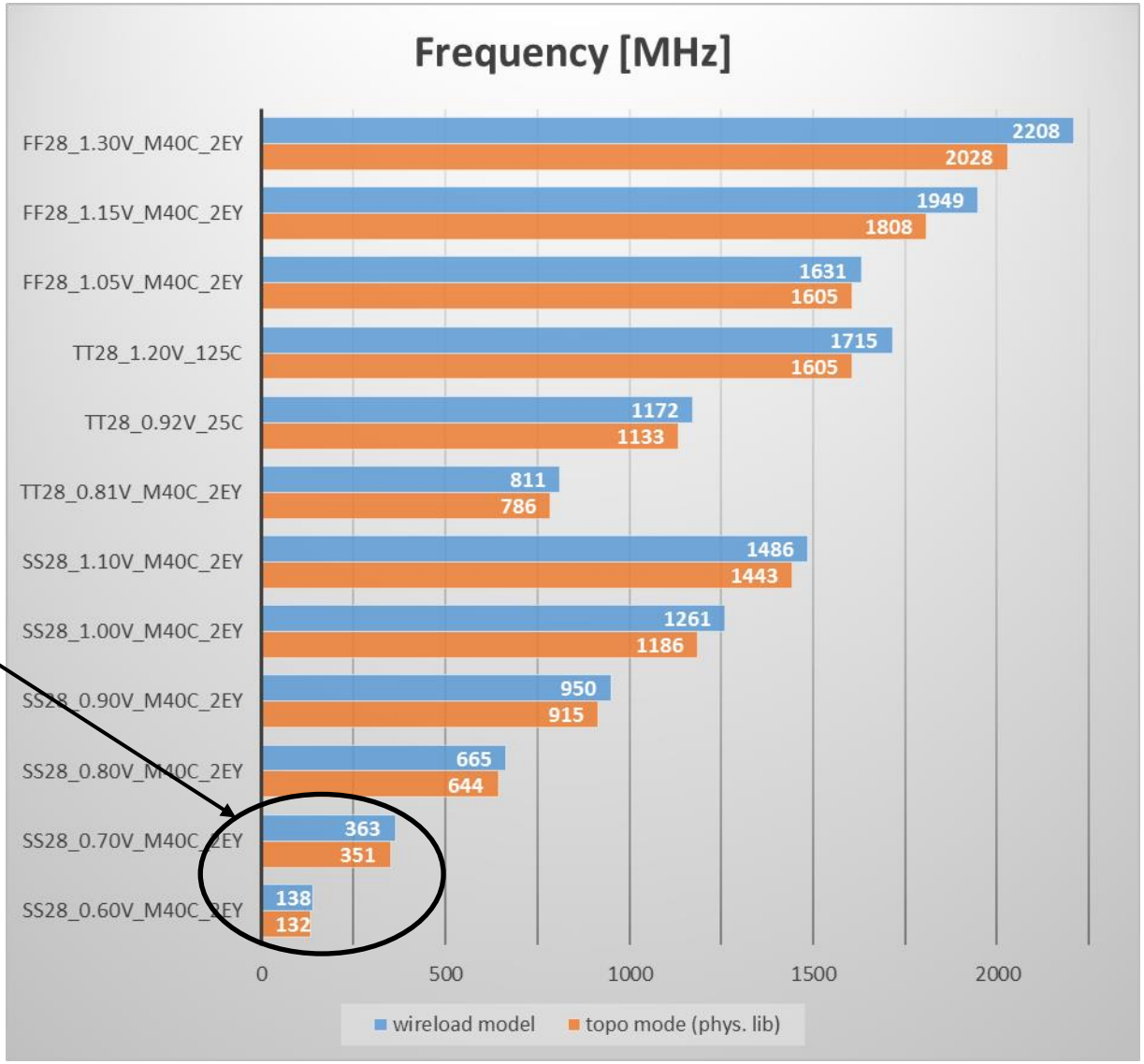
I. Kazi, P. Meinerzhagen, P.-E. Gaillardon *et al.*, EPFL

Variation Aware Design

Sensitivity at Different Operating Conditions: Voltage Scaling Introduces Uncertainties

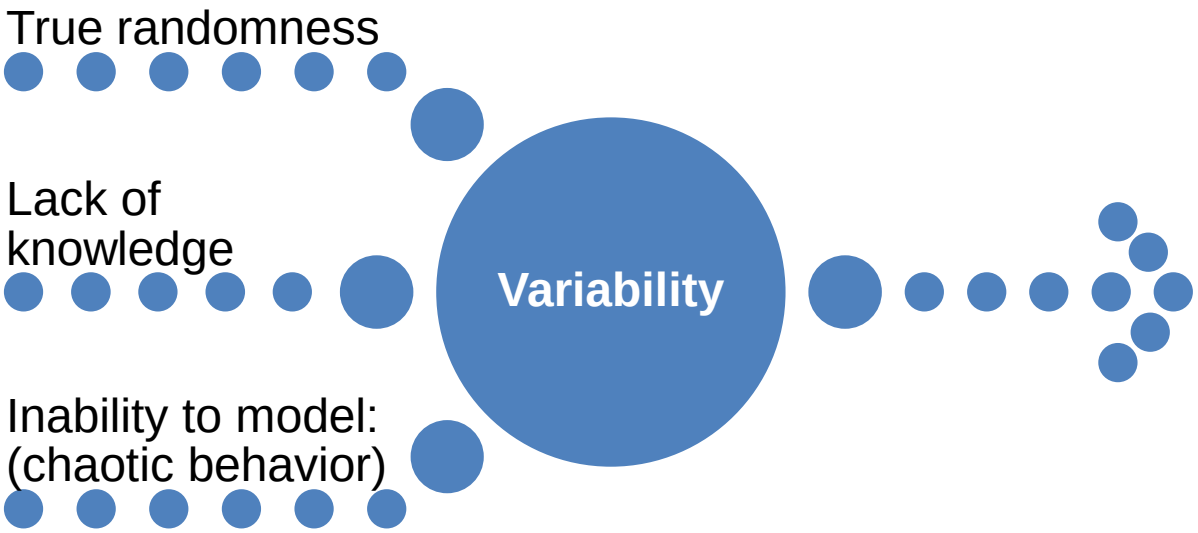
OpenRISC Processor "Cappucino" core in 28nm FDSOI

Timing is extremely sensitive to voltage



What is Variability?

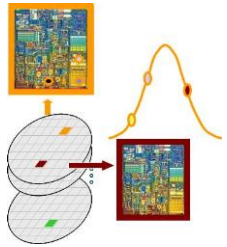
- Variability summarizes three different problems



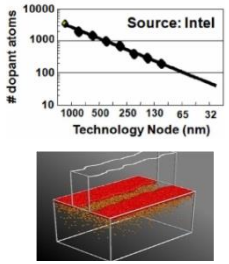
- Device parameters, operating conditions and circuit behavior are considered to be random variables
- Impact of variations is not an ergodic process
  - Time average is not the same as average over individual realizations (ensemble average) of the random process

Sources of Variability: Overview

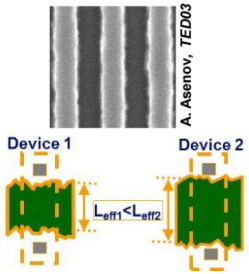
- Static components …



Process variations

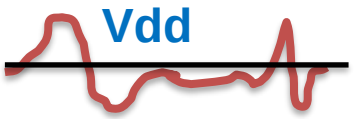


Random dopant fluctuation

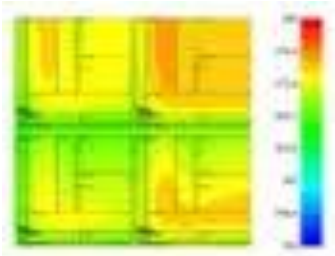


Line-edge roughness

- Dynamic/runtime factors …



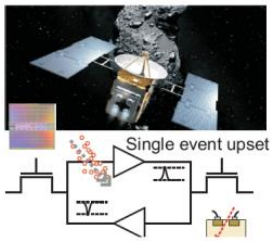
Voltage variation



Thermal

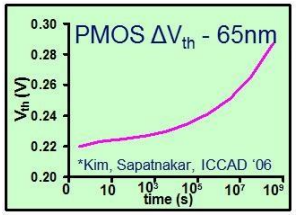
1010010
0100100
0100100

Data dependencies



Single event upsets

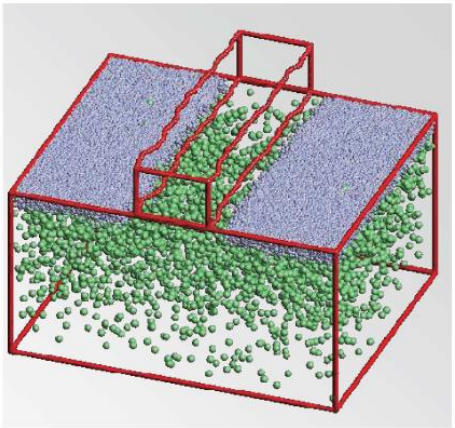
- Wearout/aging …



NBTI

Manufacturing Variations: RDF (Bulk Process)

- **Discrete number of dopants in the channel depletion region**
  - **Implantation is a random process that leads to statistical fluctuation of the number of dopants N in a given volume (channel)**
  - **Variance follows Poisson distribution: $\sigma_N = \sqrt{N}$**
  - **Example: $W = L = 90\text{nm}$, $D = 350\text{Å}$, $N_a = 10^{18}\text{cm}^{-3}$**



Miyamura, M., et al.

- $N = W \cdot L \cdot D = 284 \rightarrow \sigma_N = 17$
- **Number of dopants determines threshold voltage**
  - **Threshold voltage variation is Gaussian with variance**

$$\sigma_{V_{th}} = \sqrt[4]{2q^3 \varepsilon_{Si} N_a \phi_B} \frac{T_{Ox}}{\varepsilon_{Ox}} \frac{1}{\sqrt{3WL}}$$

Impact of RDF decreases with increasing transistor size (WL)
- **Upsizing helps**
- **Large impact on min. size SRAM**

Mizuno, Tomohisa, J. Okumtura, and Akira Toriumi. "Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFET's." Electron Devices, IEEE Transactions on 41.11 (1994): 2216-2221.

Miyamura, M., et al. "SRAM critical yield evaluation based on comprehensive physical/statistical modeling, considering anomalous non-Gaussian intrinsic transistor fluctuations." VLSI Technology, 2007 IEEE Symposium on. IEEE, 2007.

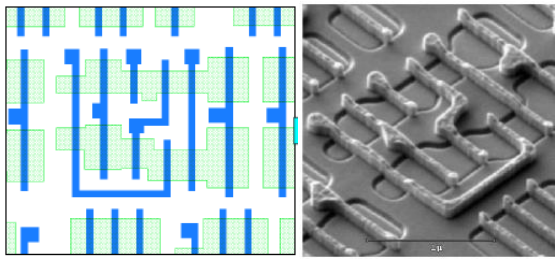
Manufacturing Variations: LER and Proximitry Effects

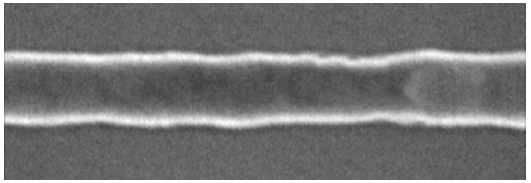
Optical lithography: feature size far below wavelength of the light

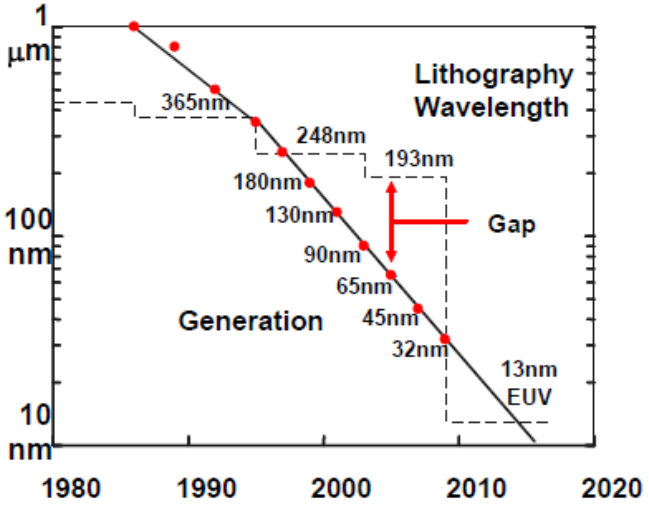
- Sub-wavelength lithography with optical proximity correction (OPC)
- Systematic variation of dimensions (gate and interconnect)
- Hard to predict, but deterministic

Line-edge roughness (LER)

- Caused by un-isotropic edging
- Generally random but impact is small



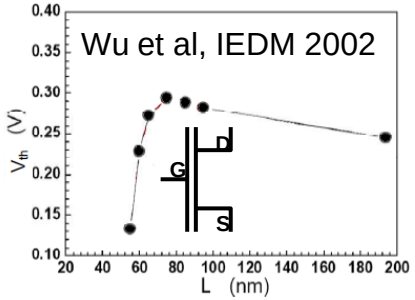




Mack CA, Conley W; Special section guest editorial: line-edge roughness. J. Micro/Nanolith. MEMS MOEMS.

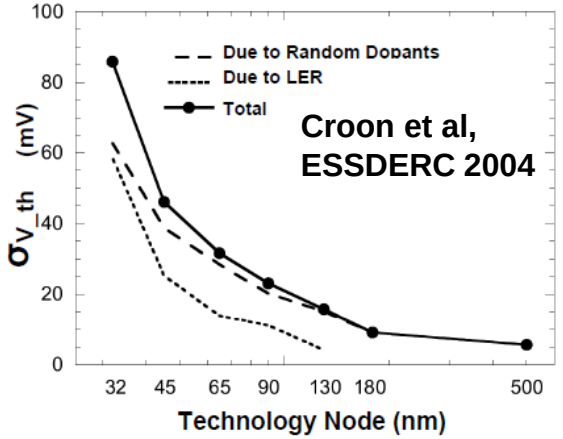
Impact of channel-length variation

- Threshold voltage through drain induced barrier lowering (DIBL)
- Directly on drain current through channel length



$$I_D \propto 1/L \qquad V_{th} \propto V_{th0} - (\zeta + \eta V_{DS})e^{-L/\lambda}$$

Line-edge roughness becomes relevant beyond 45-nm nodes



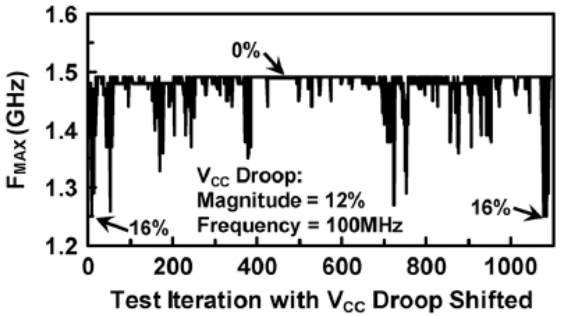
J. Tschanz, K. Bowman, and V. De, \Variation-tolerant circuits: circuit solutions and techniques," in DAC '05: Proceedings of the 42nd annual conference on Design automation, 2005, pp. 762{763.

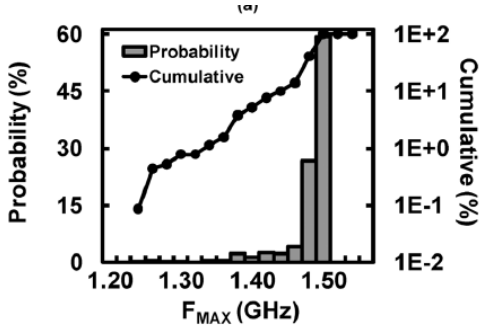
Dynamic/Lifetime Variations: Voltage & Temperature

Voltage

- Inject droops at different times
- Measure impact on max. freq.

Impact on frequency depends on location in time of a voltage droop



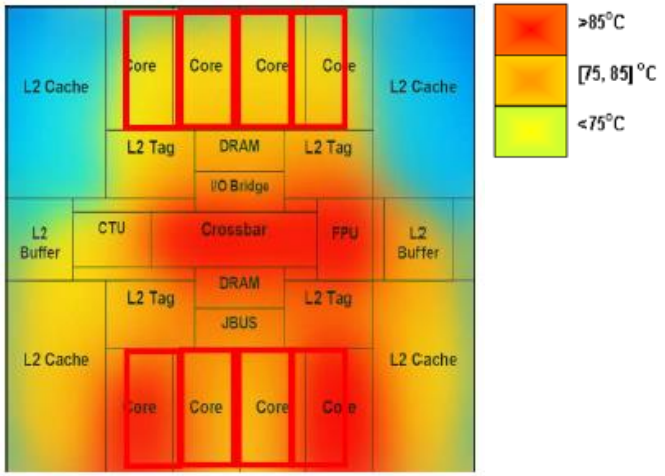


Bowman, Keith A., et al. "All-digital circuit-level dynamic variation monitor for silicon debug and adaptive clock control." *Circuits and Systems I: Regular Papers, IEEE Transactions on* 58.9 (2011): 2017-2025.

Temperature

- Temperature variations within the die cause variations in device mobility and threshold voltage as well as wire resistivity.
- Time constants: ~100ms [Coskun et al., 2010]

Coskun, Ayse Kivilcim, et al. "Energy-efficient variable-flow liquid cooling in 3D stacked architectures." Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010. IEEE, 2010.

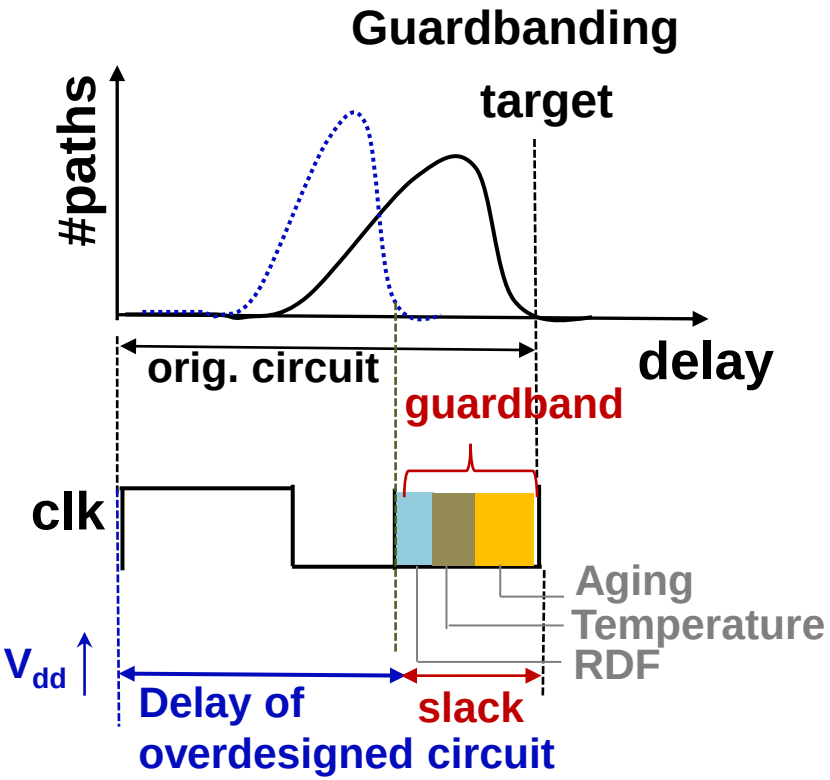


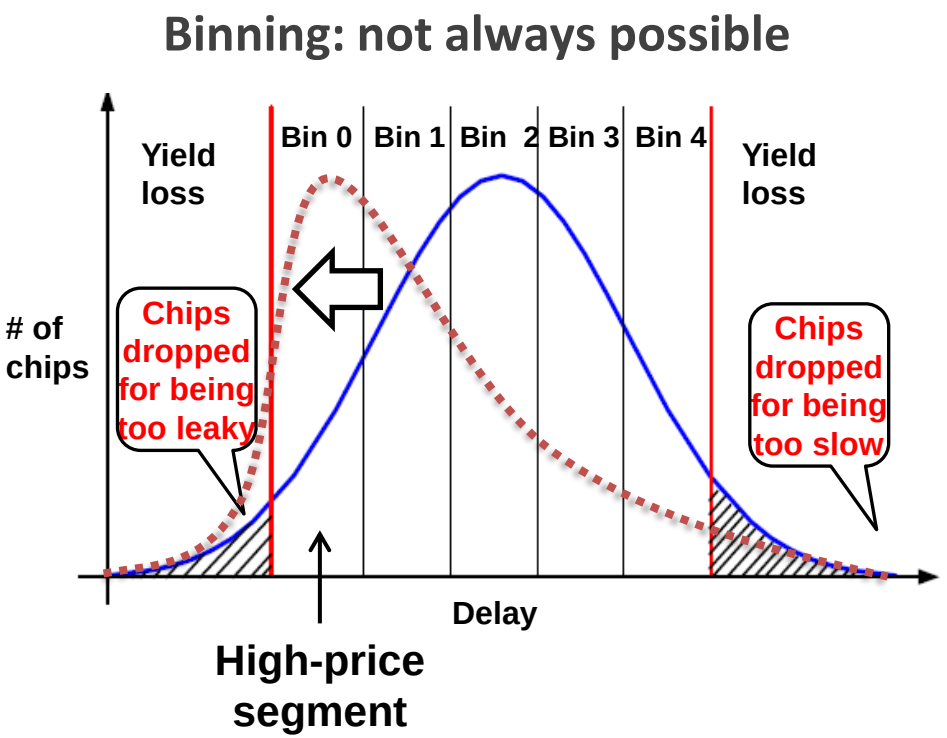
D. Atienza, ECRTS 2011, Keynote.

Variability: Cross Layer Overview

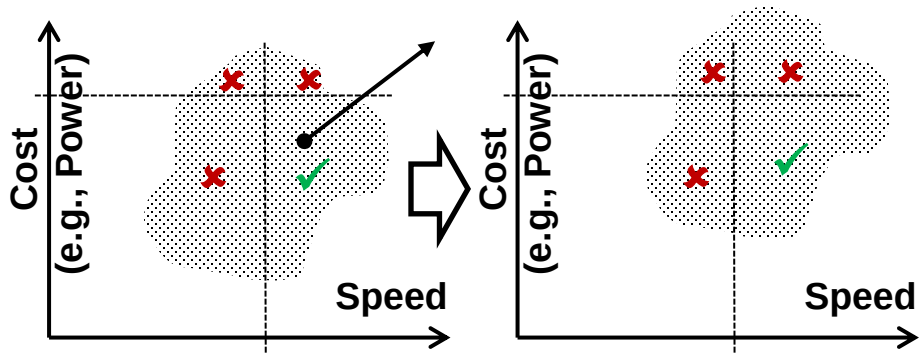
Abstraction level	Variability is or affects:
System	• Quality of Service
Algorithm	• SNR, PSNR, BER, PER
SoC	• Flexibility, Features, GOps
RTL	• Timing, Energy, Area
Compact Model	• Electrical (VDD, Body Bias)
Technology	• Geometry, Chemical

Conventional Approach: Global Yield Optimization

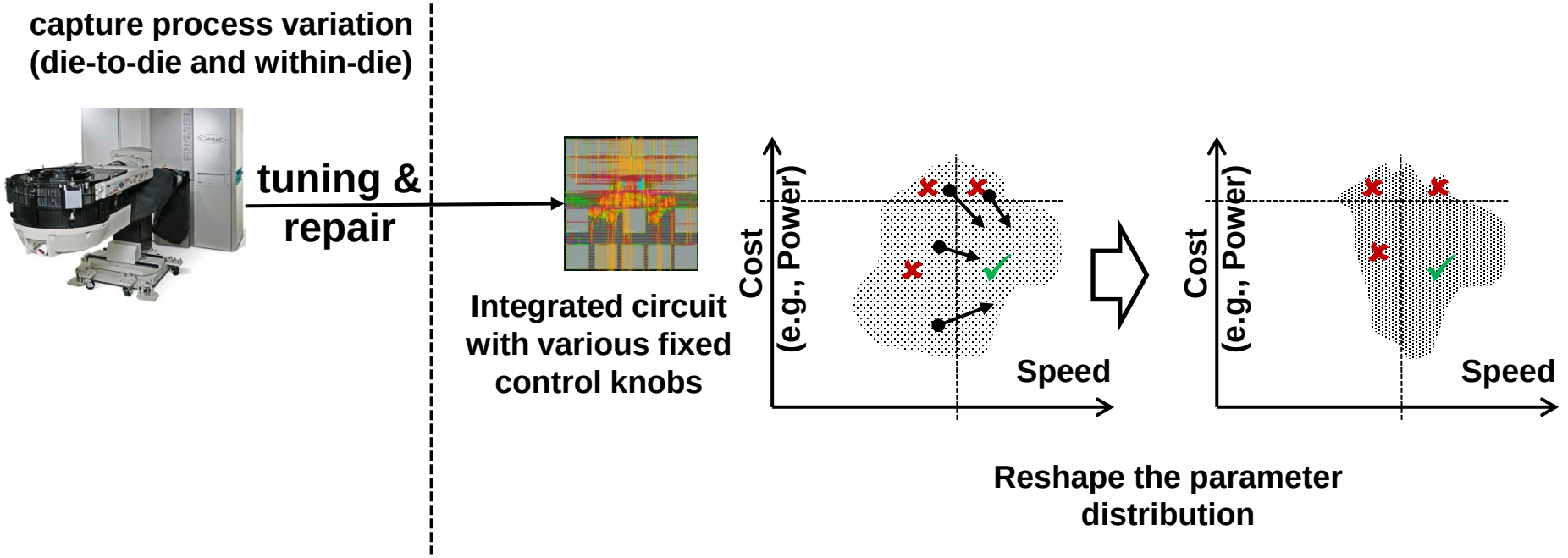




- Global shift affects the entire population
- Good dies suffer unnecessary penalties
- Multi-dimensional parameter space: global opt. can even introduce penalties



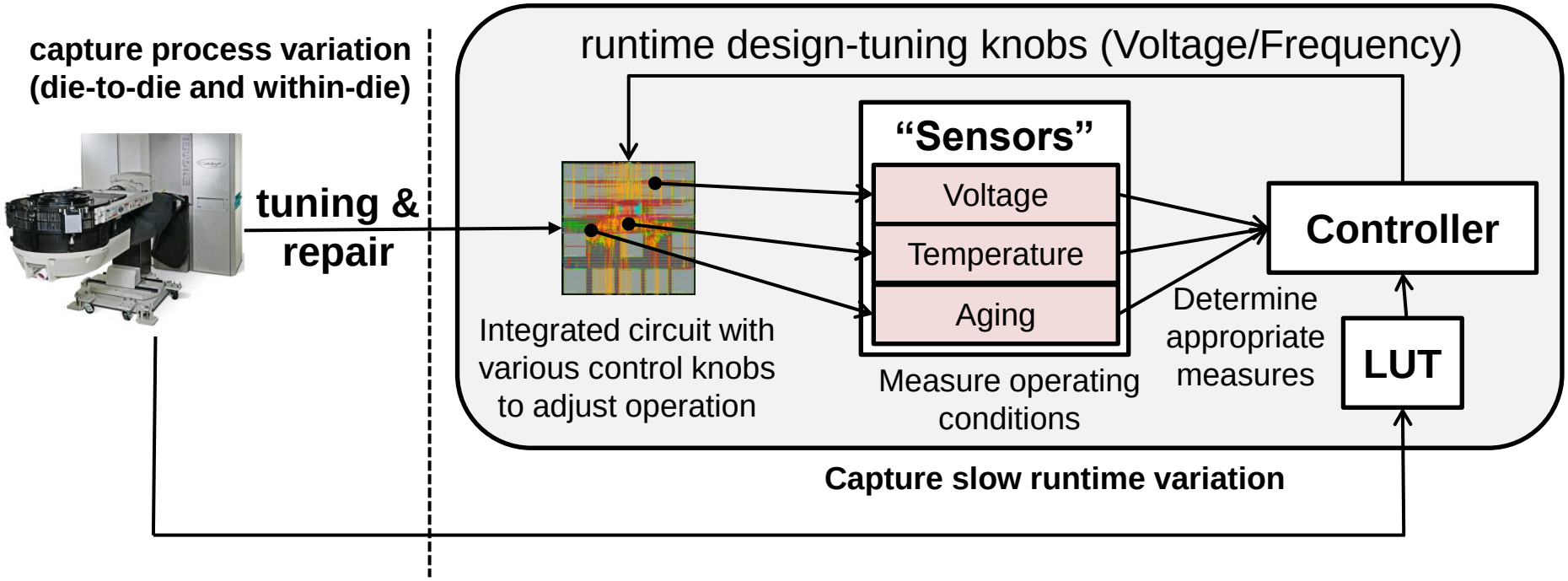
Adaptive Tuning: Basic Principle



Objective: adjust design-knobs for "safe" operation at minimum "cost" for each chip

- Production test identifies process variations and sets design parameters (fuses) or decides on binning
- Can not track variations over time

Adaptive Tuning: Basic Principle

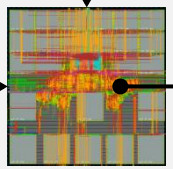


- **Objective: adjust design-knobs for "safe" operation at minimum "cost" for each chip at any point in time**
  - **Initial testing provides**
    - **a baseline calibration for process variations**
    - **tuning parameters as a function of sensor readings => long and complex testing required to characterize many operating points (interpolation difficult)**

Adaptive Tuning: Basic Principle

tuning & repair

runtime design-tuning knobs



“Sensors”

Controller

Integrated circuit with various control knobs to adjust operation

Check for the impact of variability

Determine appropriate measures

capture process variation (die-to-die and within-die)

capture runtime variation

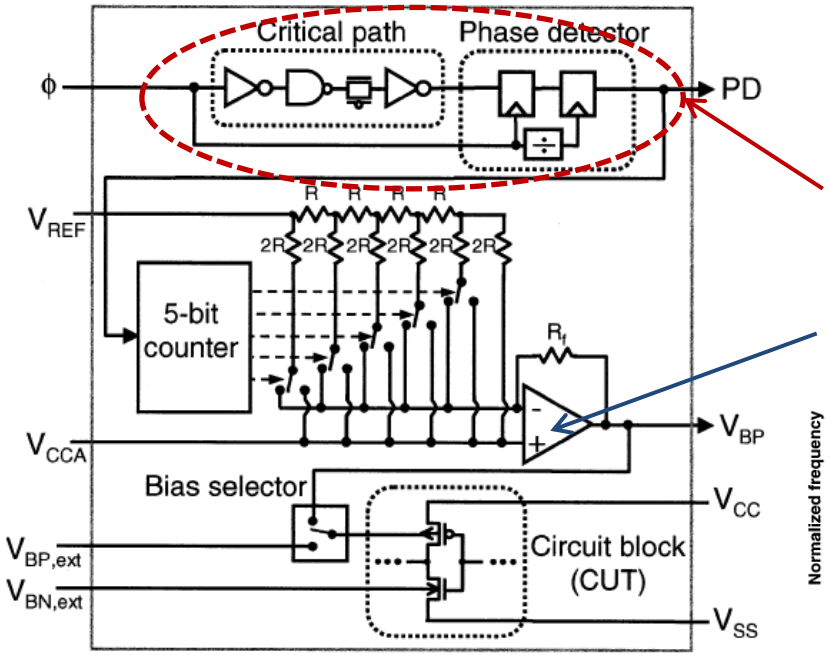
- **Objective: adjust design-knobs for “safe” operation at minimum “cost” for each chip at any point in time**
  - **On-chip sensors check directly on the operating margin and detect errors**

Electrical Knobs: Adaptive Body Bias

Maximize clock frequency under total power constraint

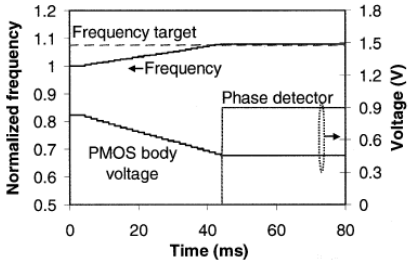
- V_{th} variation determines leakage/freq. operating point
- Adjust V_{th}: forward/reverse body bias (FBB/RBB)

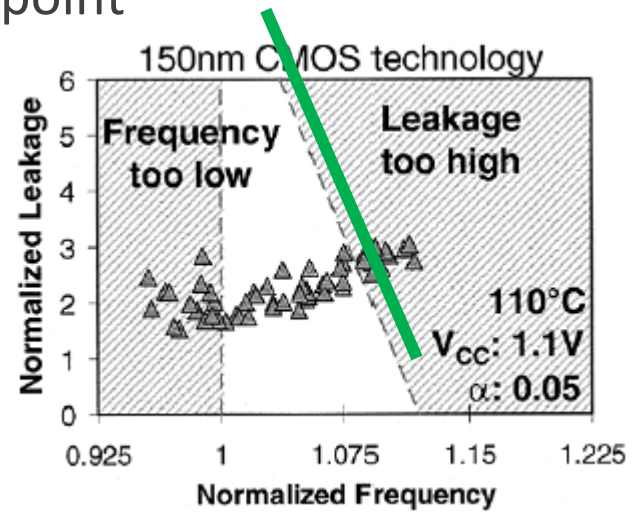
RBB: leakage ↓, speed ↓ FBB: leakage ↑, speed ↑

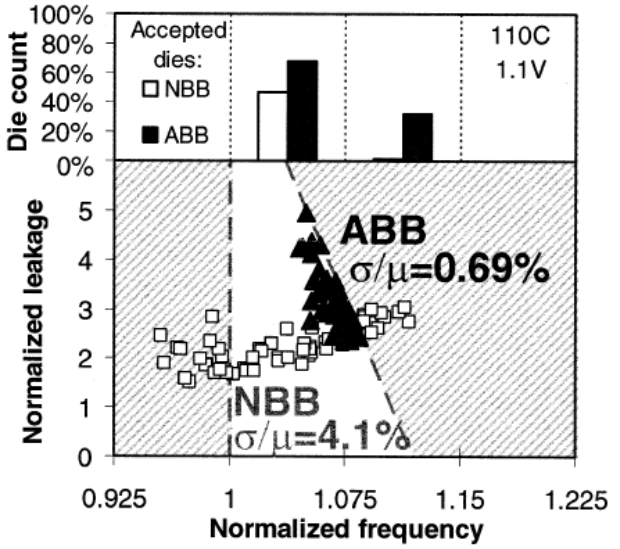


Sensor: replica critical path with phase detector

Control knob: Variable BB generator





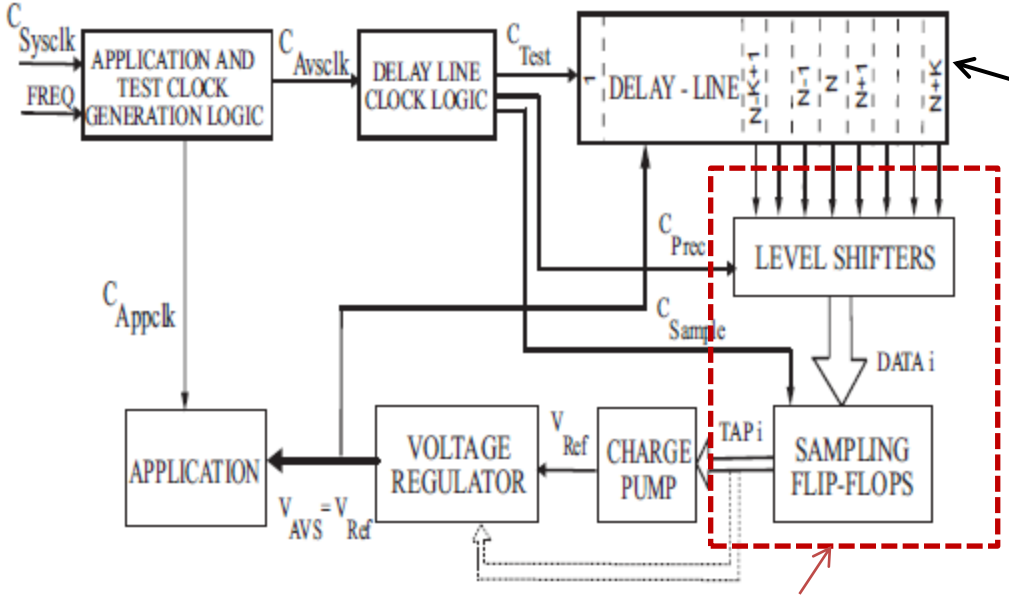


Tschanz, James W., et al. "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage." Solid-State Circuits, IEEE Journal of 37.11 (2002): 1396-1402.

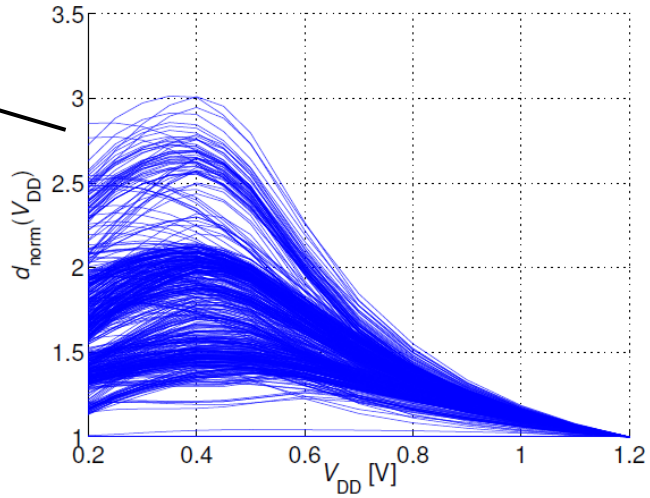
Delay lines capture delay increase due to global PVT variations

- Single instance *cannot* capture intra-die variations, local supply drops, crosstalk

Resolution: 1 FO2 inverter delay



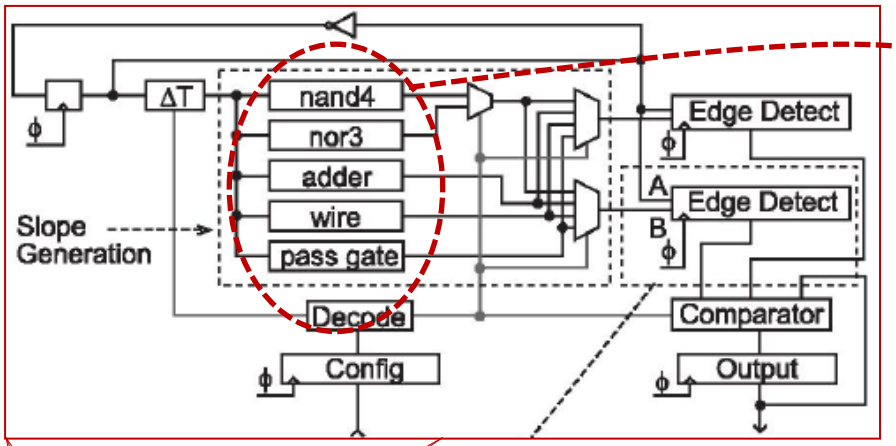
Time-to-digital-converter to measure delay as input to closed-loop voltage control

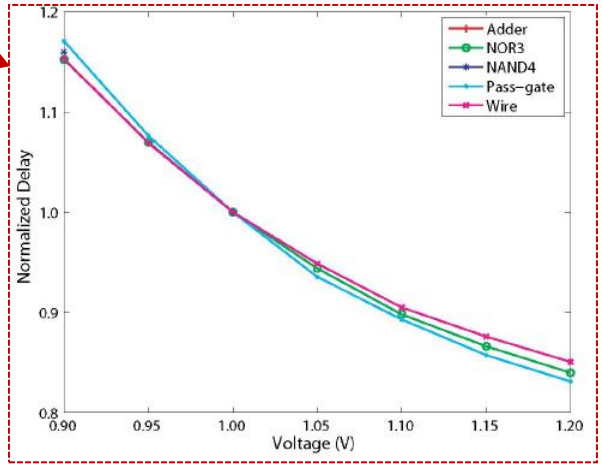


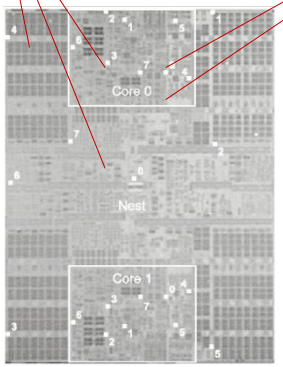
Delay variation of complex gates is significantly different from that of an inverter, especially at low voltages

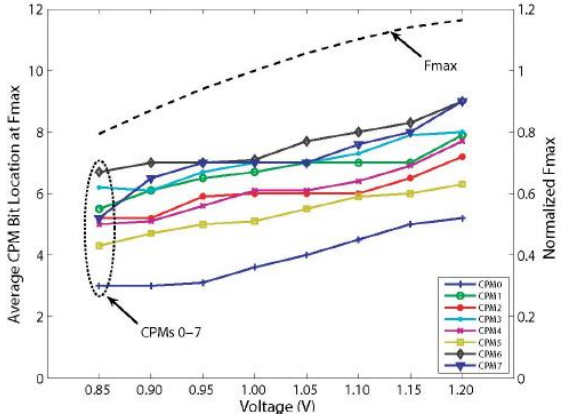
Dhar, Sandeep, Dragan Maksimović, and Bruno Kranzen. "Closed-loop adaptive voltage scaling controller for standard-cell ASICs." ***Proceedings of the 2002 international symposium on Low power electronics and design*****. ACM, 2002.**

- ***Distributed across the chip*** **to better capture local variations**









Calibration during test

Multiple delay-paths cover different types of delay variation

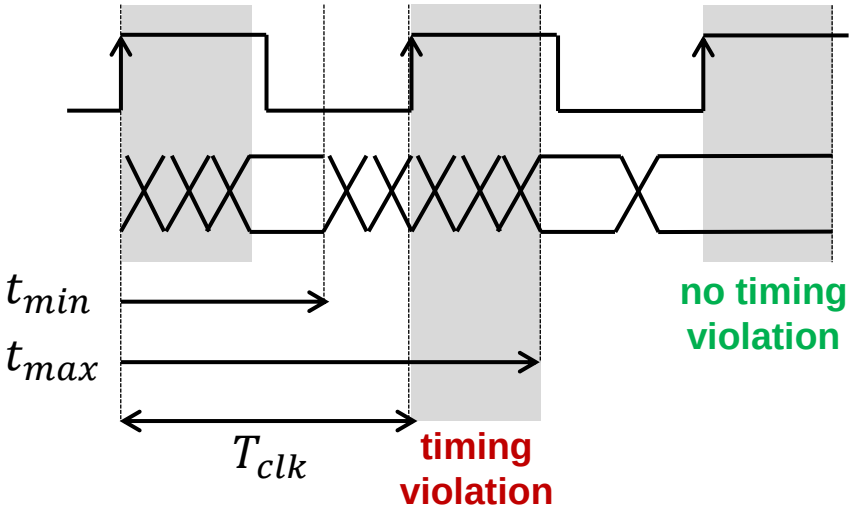
- **Different gates (4-NAND, 3-NOR, ADD) and "wire-dominated" paths**

Runtime frequency setting

Drake, Alan, et al. "A distributed critical-path timing monitor for a 65nm high-performance microprocessor." *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*. IEEE, 2007.

Basic idea: detect timing violations directly at each path endpoint to capture also local variations, cross-talk, and data dependent delay variations

- **Detect any change of data after the clock edge (late-arrivals)**



t_{max}: worst case path delay

t_{min}: contamination delay

$$t_{CLK} < t_{max}$$

$$t_W = t_{max} - t_{CLK}$$

Tradeoff between t_W and t_{min}

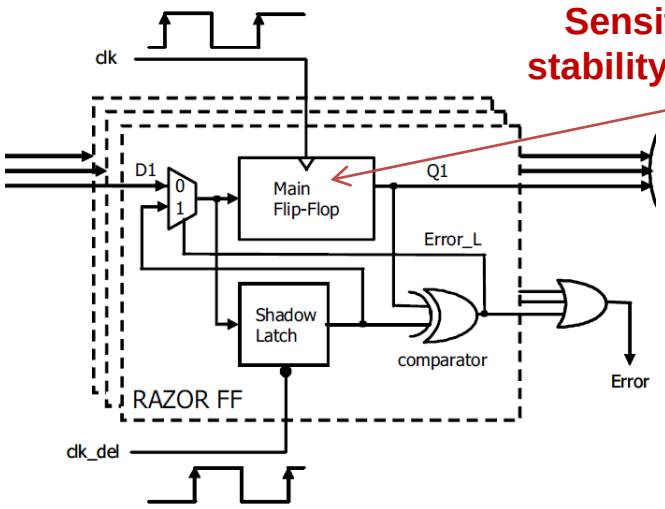
$$t_{min} > t_W + t_{hold}$$

- t_W ↑: more margin for variability, but more extra buffers to guarantee t_{min}

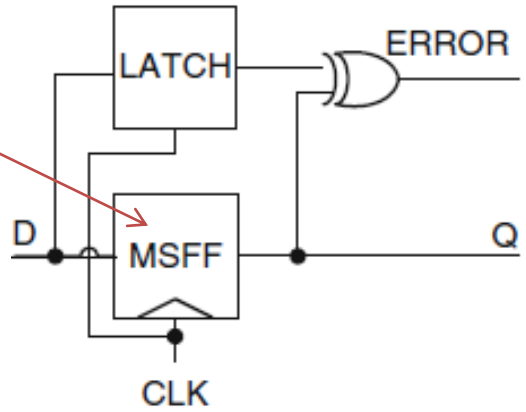
- **Data is captured on the positive clock edge**
- **Monitor the input of the sequential circuit and report any transition in the timing violation window => ERROR**
- **Early arrivals are not allowed within the timing violation window t_W (can not be distinguished from late transitions)**

Examples: 1st generation

- Data stored with an edge-triggered FlipFlop
- A shadow latch remains open during the timing violation window
- Error signal generation by comparing data and shadow-latch output



Sensitive to meta-stability on late inputs

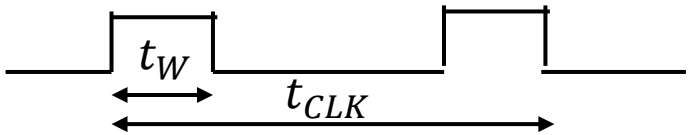


Razor

- Correct data can be restored after an error from the shadow-latch

Double sampling error detection sequential (DS-EDS)

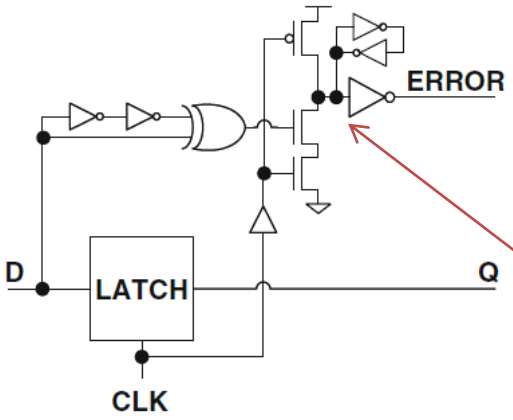
- Single pulsed clock signal simplifies clock distribution



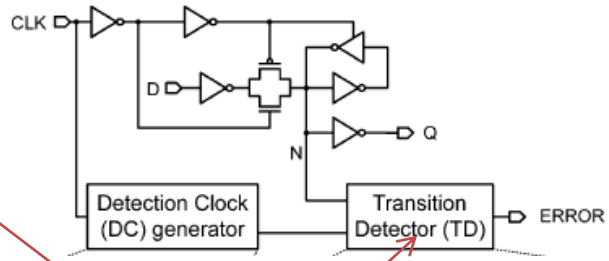
D. Ernst et al., Razor: A low-power pipeline based on circuit-level timing speculation, in Proceedings of the IEEE/ACM International Symposium Microarchitecture (MICRO-36), 2003

Example: 2nd generation resolves metastability issue on the data

- **Data is captured by a pulsed latch (used as FlipFlop): closes in a safe place**
- **Pulse duration $t_{pulse} = t_W < t_{min}$ defines timing violation window**

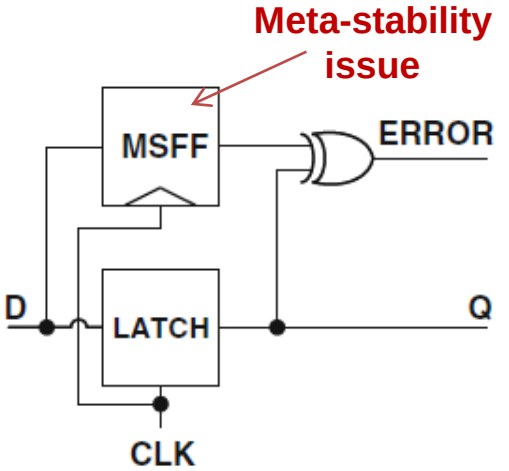


Transition detector with time borrowing (TDTB)

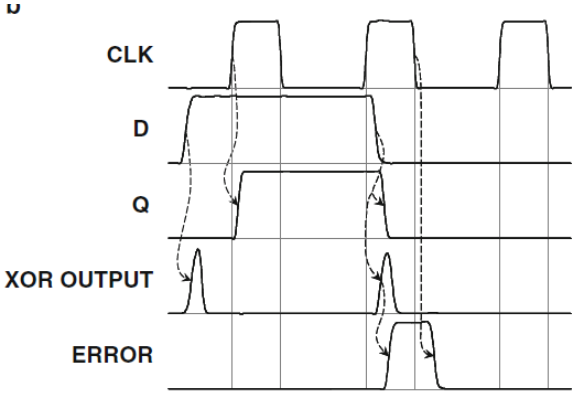


RAZOR II

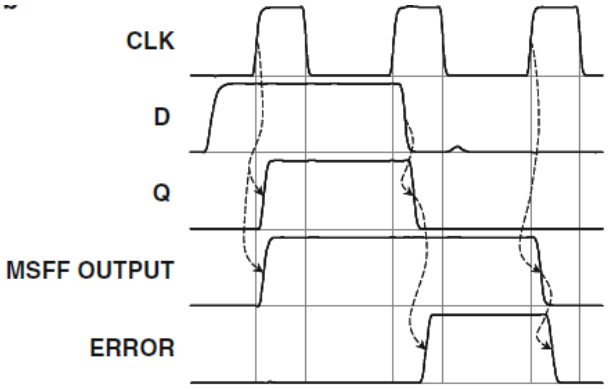
Dynamic transition detector: sensitive to variations



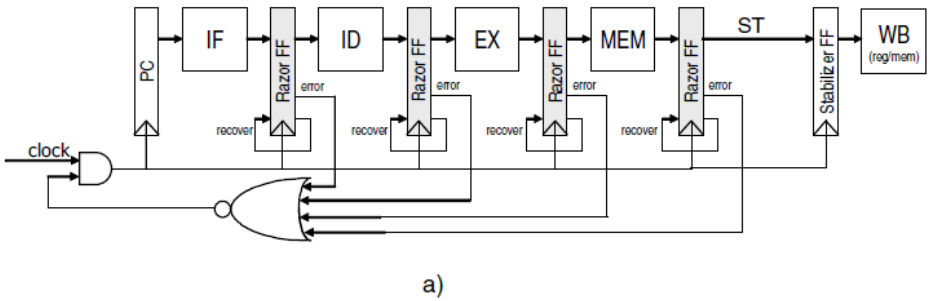
Double sampling with time borrowing (DSTB)



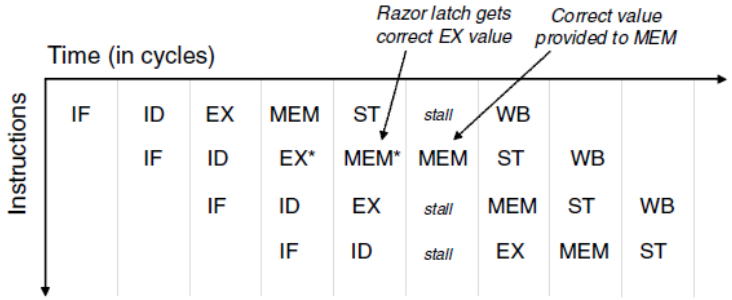
Design risk compared to RAZOR or DS-EDS: Duty-cycle control of the clock is always required (can not be operated on a regular clock)

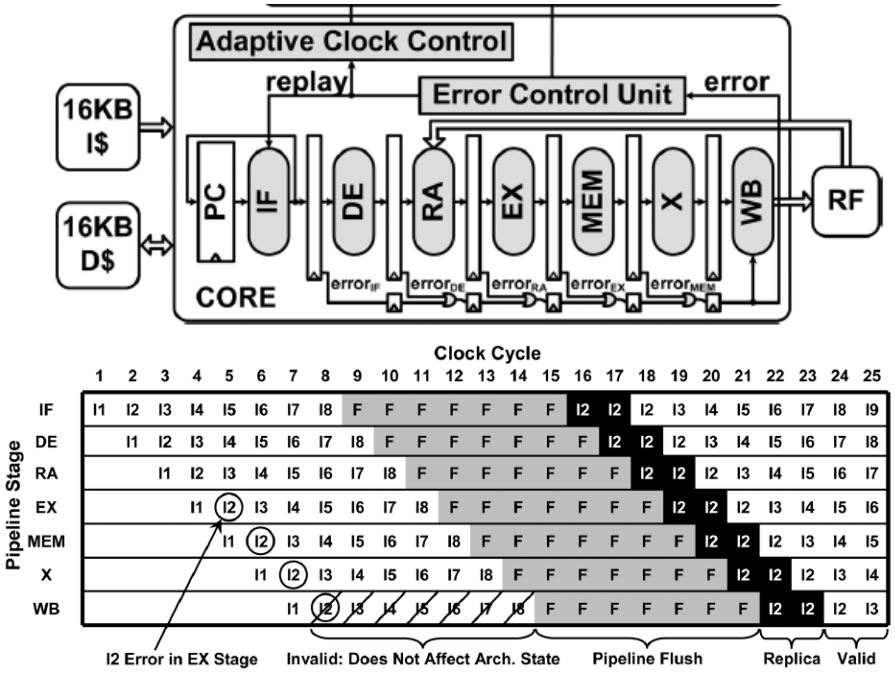


Architecture Level Knobs: Correcting Timing Violations



a)





Timing violations trigger a pipeline stall

- Correct data available from Razor latch
- Re-evaluation/recovery in stall cycle
- ✗ Immediate full-chip clock-gating feedback is difficult in high-frequency designs (latency)

D. Ernst et al., Razor: A low-power pipeline based on circuit-level timing speculation, in Proceedings of the IEEE/ACM International Symposium Microarchitecture (MICRO-36), 2003

Erroneous instructions are re-issued into the pipeline, avoiding the need for clock gating

- Timing errors flush the pipeline
- Failing instruction is re-issued N times to avoid repeated failure during replay
- ✗ Overhead due to pipeline flush

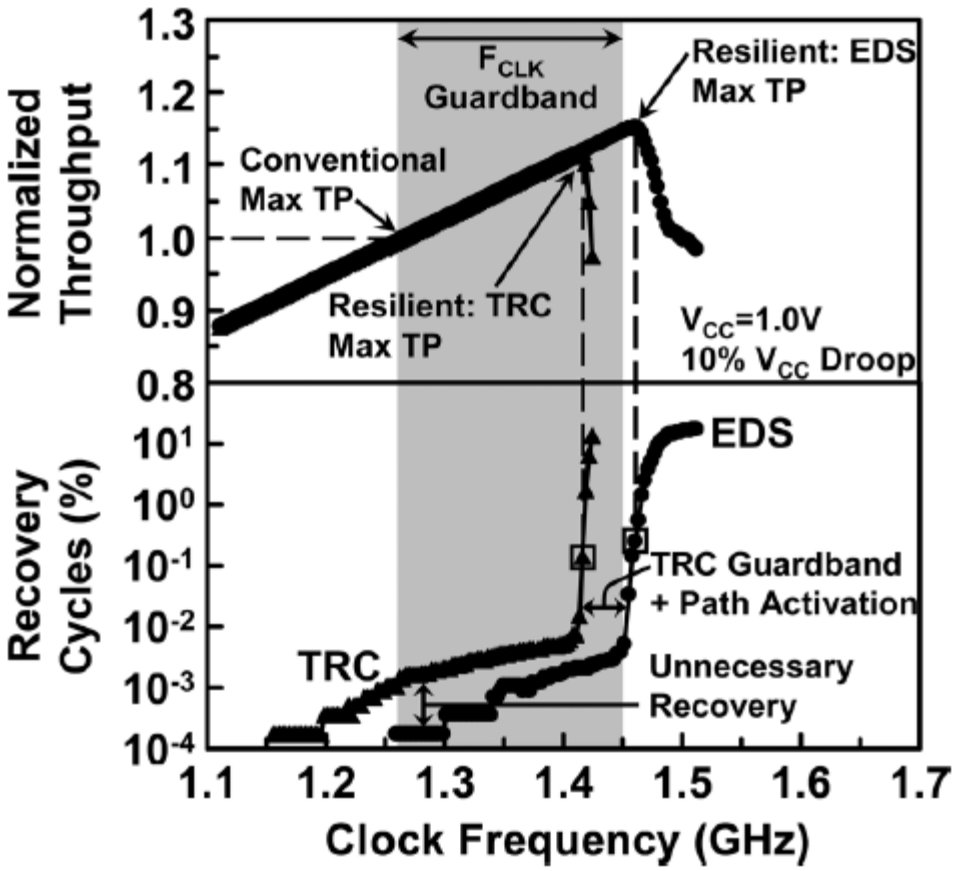
Bowman, Keith A., et al. "A 45 nm resilient microprocessor core for dynamic variation tolerance." *Solid-State Circuits, IEEE Journal of* 46.1 (2011): 194-208.

Architecture Level Knobs: EDS vs. TRC based Detection

Compare Error Detection Sequentials vs. Timing Replica Circuits (TRC)

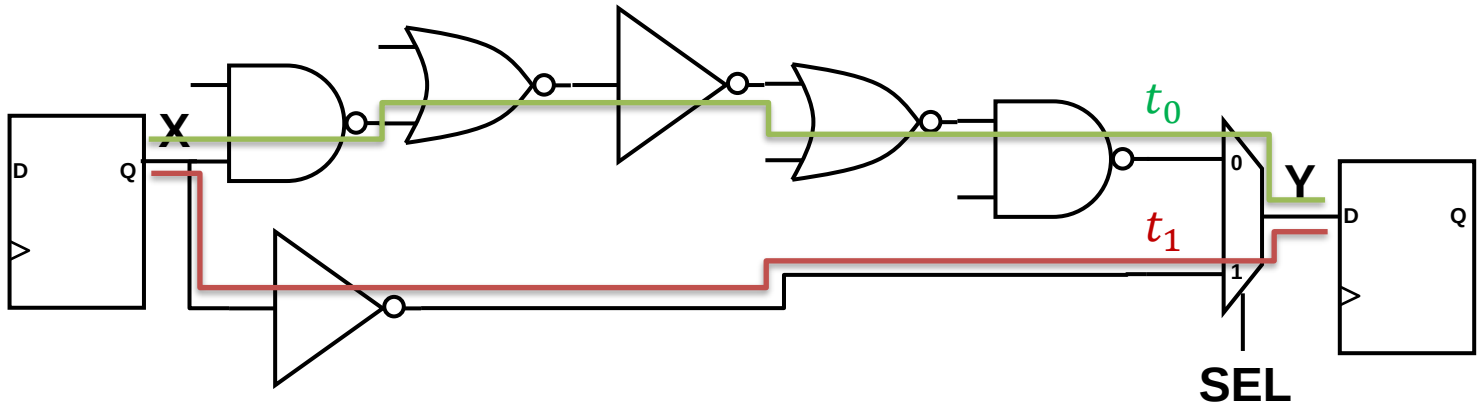
Example: Microprocessor [Intel]

- **EDS is more complicated to implement**
- **EDS involves more overhead than TRCs since sequential elements are more complex**
- **EDS reduces the number of replay events and provides tighter margin**
  - **Captures data dependent delays**
  - **Does not require guard-bands to cover local variations**



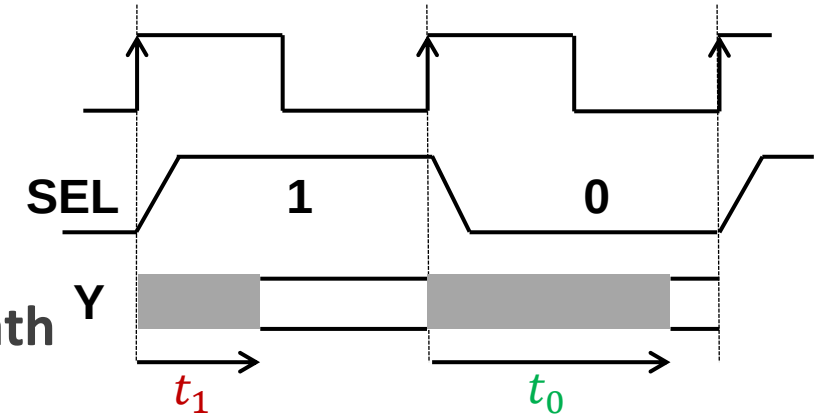
Bowman, Keith A., et al. "A 45 nm resilient microprocessor core for dynamic variation tolerance." *Solid-State Circuits, IEEE Journal of* 46.1 (2011): 194-208.

Architecture Level Knobs: Critical Path Excitation



Setup- and Hold constraints ensure that the system is in a steady-state in the data call window of the sequential elements

- Worst-case assumption based on a transition triggered by the critical path



However, the critical path is not always excited and arrival times depend on data

- Critical path may be blocked/interrupted
- Signals may remain constant without any glitches

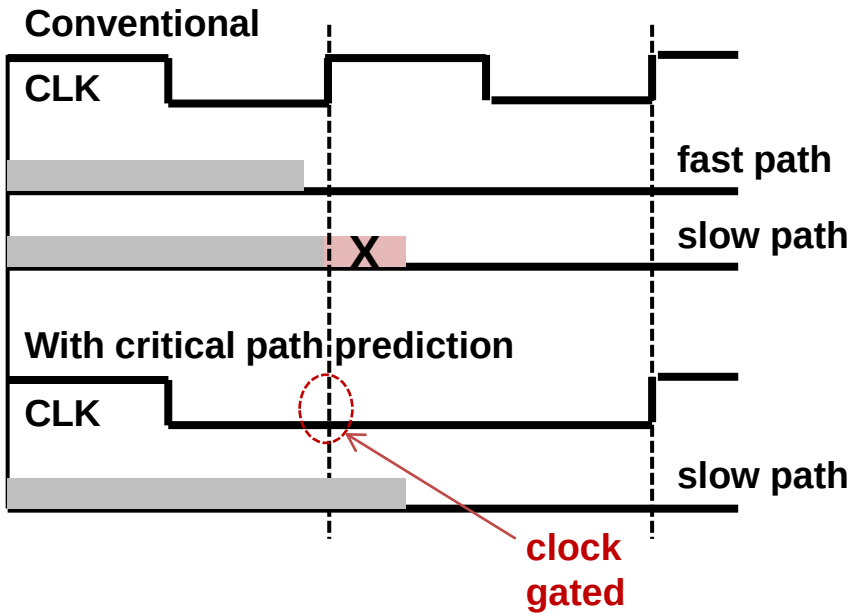
Architecture Level Knobs: Exploiting Data Dependent Delays without Error Detection Sequentials

Assumption: critical path is triggered only rarely

- Most clock cycles would remain without timing errors even under frequency-over-scaling (clock period shorter than the critical path from STA)

Idea: adjust clock frequency to a typical (rather than worst-case) delay and use multi-cycle operation when this delay is exceeded

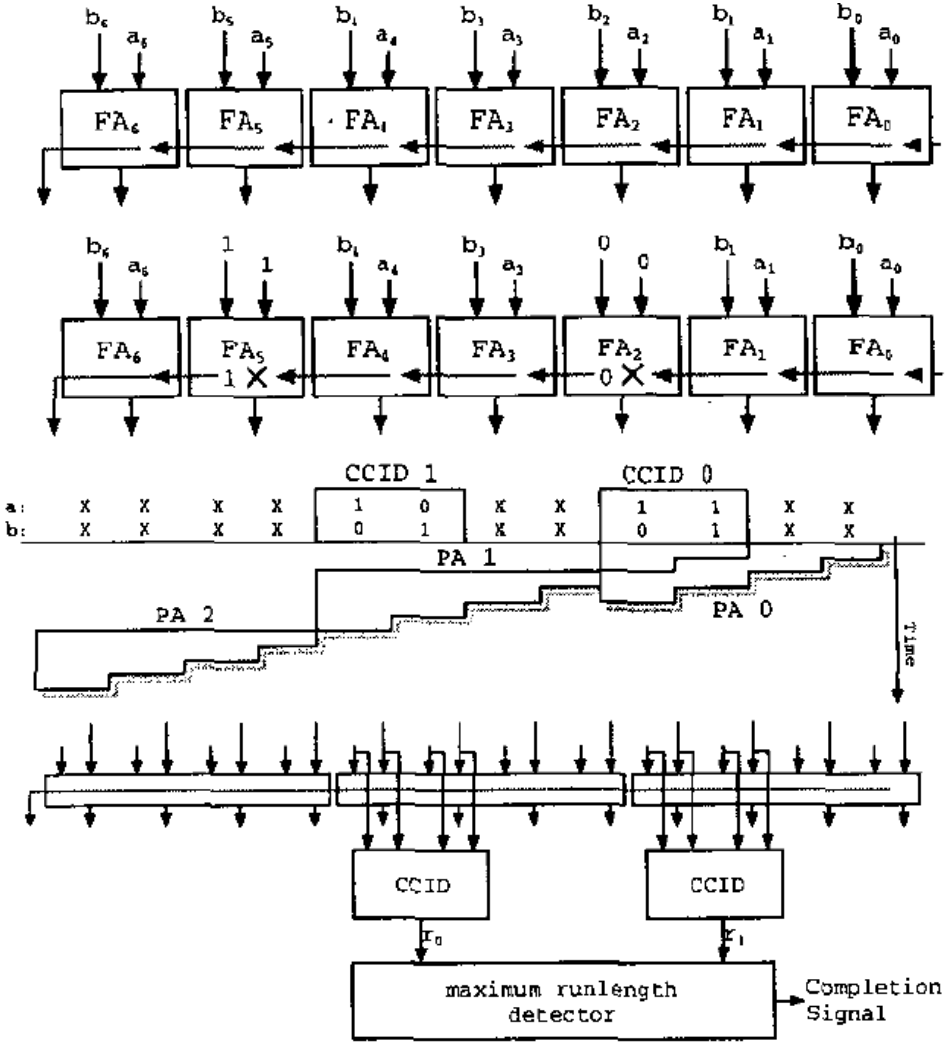
- Delay *Prediction (pessimistic)*
  - Classify operations into long and short latency
  - Detect/predict inputs that *may* trigger worst case delays and are thus susceptible to failure
- Allow Extra Cycle
  - Long latency operations are given 2 cycles

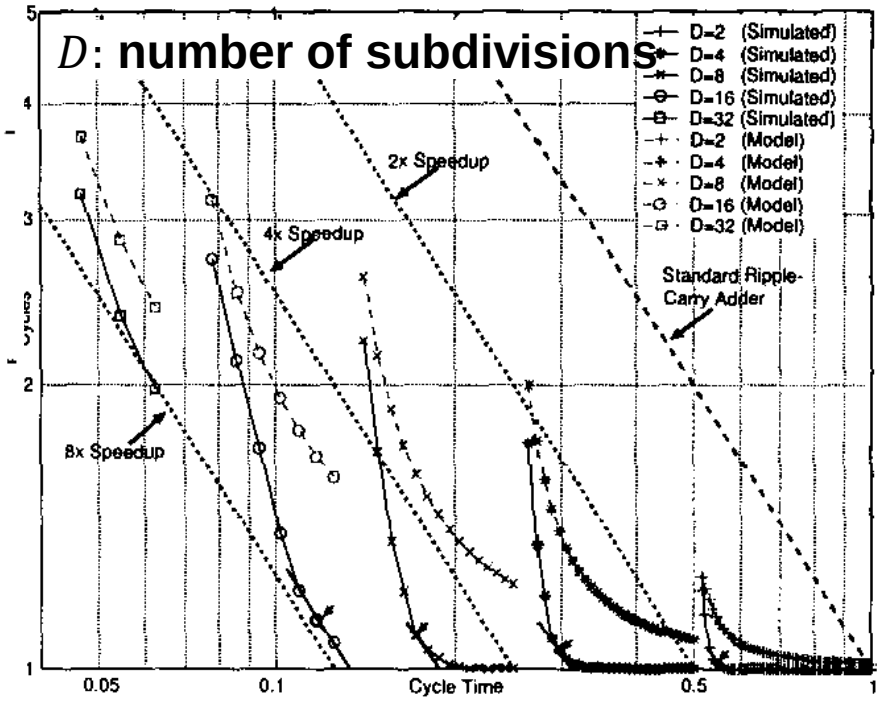


Power reduction: voltage-scaling of a conservative design at constant frequency leads to occasional timing violations, that can be avoided through multi-cycle operation

Yield improvement: Fast/nominal dies use single-cycle operation, while slow dies enable occasional two/multi-cycle operation

Ripple Carry Adder with Carry Chain Interrupt Detection





More complex prediction logic leads to

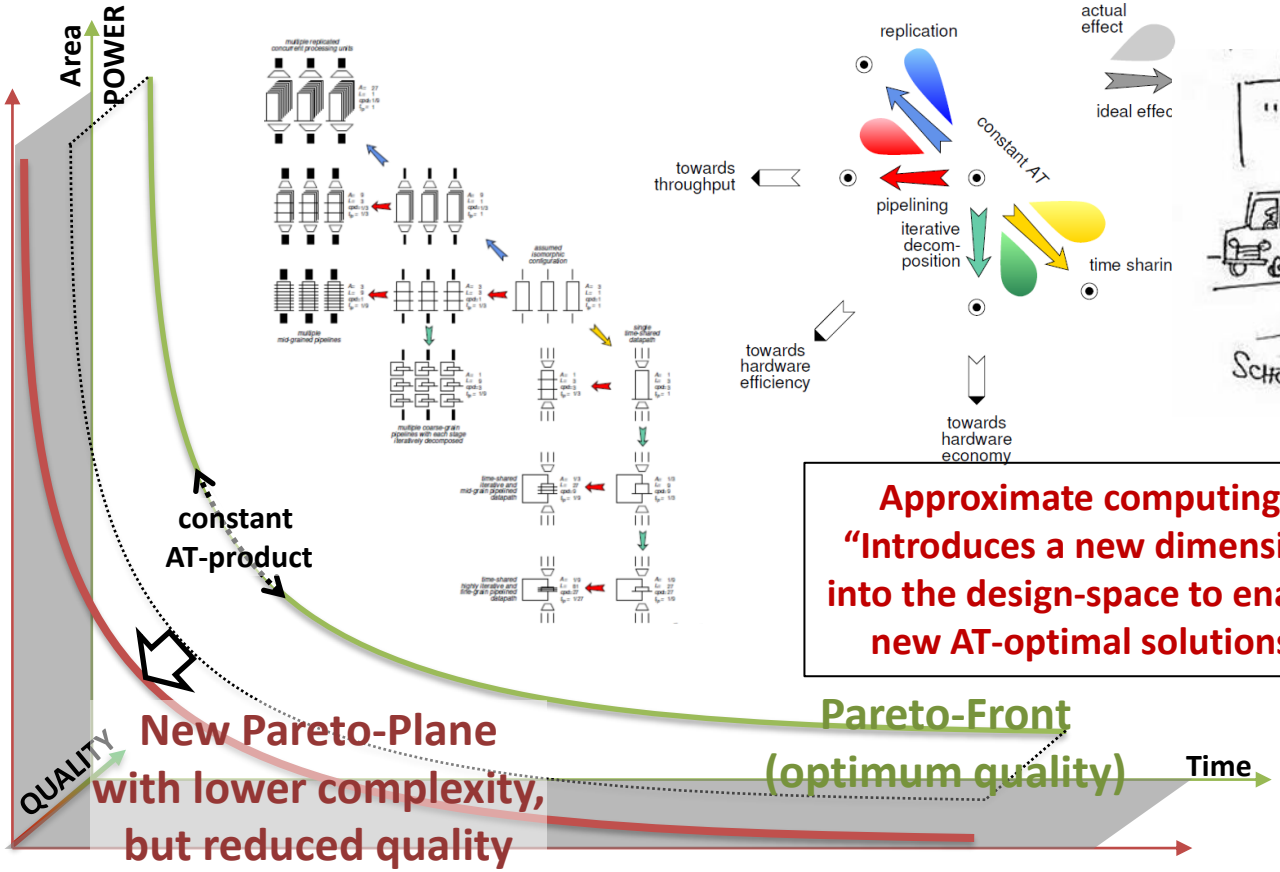
- Fewer multi-cycle operations (fewer false long-path predictions)
- More fine-grained partitioning into multiple cycles
- Longer delay for the prediction logic

} higher throughput

Burg, Andreas, et al. "Variable delay ripple carry adder with carry chain interrupt detection." ***Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*****. Vol. 5. IEEE, 2003.**

Low Power and Variation Aware Design with Approximate Computing

No Free Lunch Theorem Requires New Dimension in the Design Space



Approximate computing:
"Introduces a new dimension into the design-space to enable new AT-optimal solutions"

What is “Approximate Computing”?

- **Approximation:** A well known art ... in many aspects of our daily life

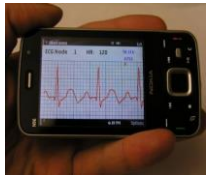



Communications

‘Noisy’ real world Inputs

Multimedia

Perceptual Limitations

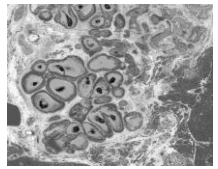


Data Mining

No single ‘Golden’ Result

Web search

Statistical Computations

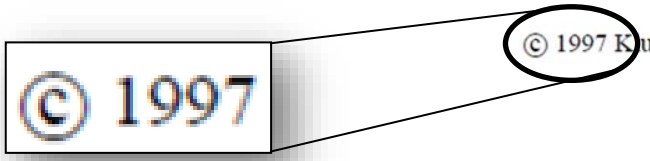


Pattern Recognition

Iterative Self healing

- **Many applications are inherently tolerant against inaccuracies or distortions**

Approximate Computing: an Ancient Art



Journal of VLSI Signal Processing 15, 177–200 (1997)

Approximate Signal Processing

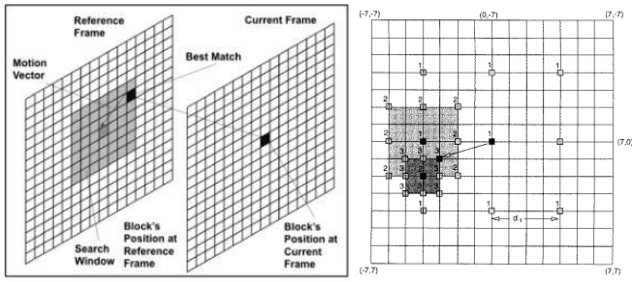
S. HAMID NAWAB
ECE Department, Boston University, 44 Cummington St., Boston, MA 02215

ALAN V. OPPENHEIM AND ANANTHA P. CHANDRAKASAN
EECS Department, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139

JOSEPH M. WINOGRAD
ECE Department, Boston University, 44 Cummington St., Boston, MA 02215

JEFFREY T. LUDWIG
EECS Department, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139

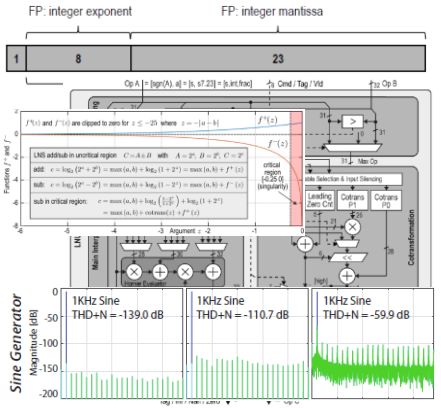
VLSI-Signal Processing: Many Beautiful Examples for "Approximate Computing"



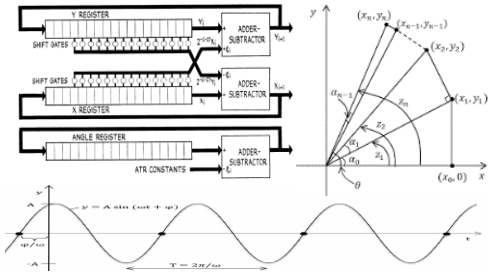
Approximation of algorithms (e.g., Motion estimation)

Cross-correlation | Sum-of-Absolute Difference | Census (Binary)

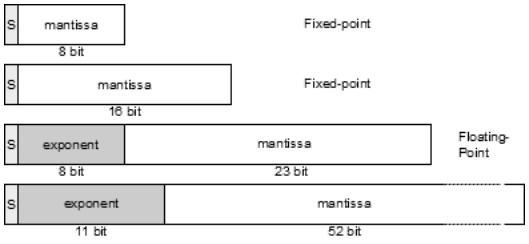
Reduced complexity cost functions (e.g., SAD or Census in 3D vision)



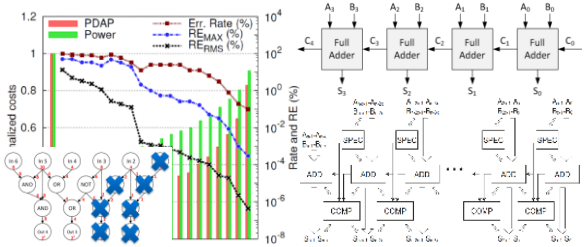
Number representations (e.g., Floating Point or Log Number Systems)



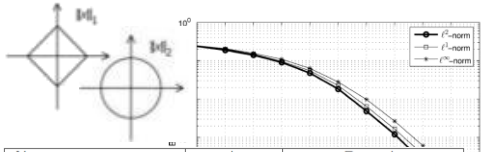
Recursive computation of transcendental functions (e.g., CORDIC for DDS)



Finite word-length optimization



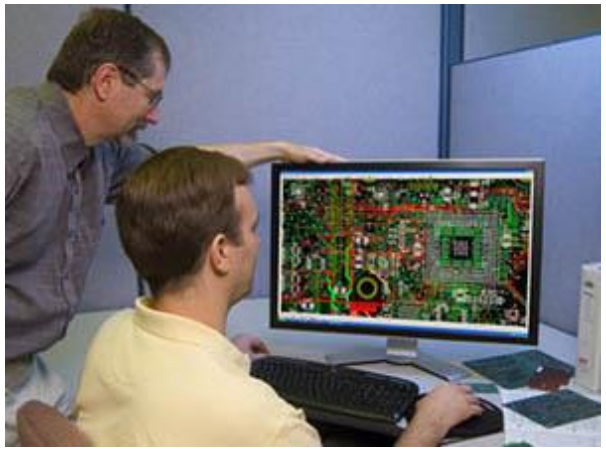
Approximate arithmetic operations, derived from by gate-level considerations (e.g., pruned or speculative adders or multipliers)



Name	aka	Formula
lesser	$\ell^{-\infty}$-norm	$l = \min(\lvert a\rvert, \lvert b\rvert)$
sum	ℓ^1-norm	$s = \lvert a\rvert + \lvert b\rvert$
magnitude (reference)	ℓ^2-norm	$m = \sqrt{a^2 + b^2}$
greater	ℓ^∞-norm	$g = \max(\lvert a\rvert, \lvert b\rvert)$
Approximation 1		$m \approx \frac{3}{8}s + \frac{5}{8}g$
Approximation 2		$m \approx \max(g, \frac{7}{8}g + \frac{1}{2}l)$

Algebraic approximations of complex operations

… and many more!!



At Design Time

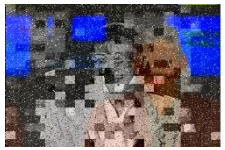
- Incorporated “intentionally by design”
- Modifications of algorithms or arithmetic operations
- 100% predictable

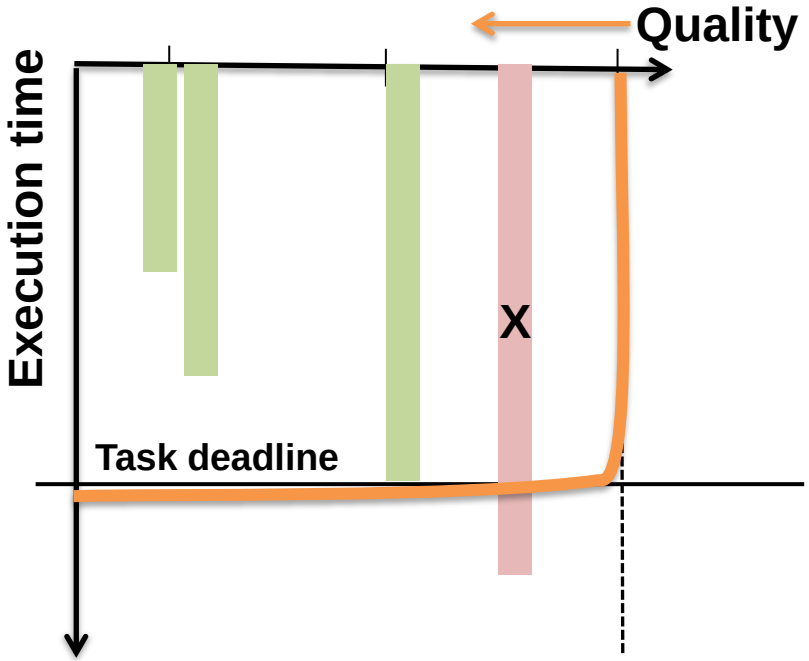
New opportunities

Approximation to the Manufacturing

- Deviating from the requirement of 100% reliable operation
- Tolerate faults in memories or logic
- Difficult to predict at design time

Real-time Systems Need Tuning Beyond the Architectural Level





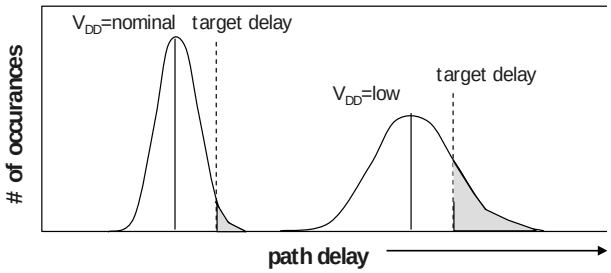
Frequency-scaling and RAZOR techniques fail at some point: sudden loss in Quality-of-Service or complete crash

Nominal voltage
- ✓ Very fast circuits: sufficient timing margin to handle reliability issues
- × Poor energy efficiency

Reliable near- and sub-VT operation
- × Overhead to meet real-time constraints even without variation
- × Large impact of variations on timing **requires large margin to meet real-time constraints**

Conventional paradigm: 100% reliable/accurate operation, always meeting real-time constraints

Real-time applications impose processing deadlines
- × **Insufficient number of cycles for task completion**
- × **Insufficient time in a clock period for necessary logic**



Voltage scaling
- ✓ Improves energy efficiency
- × Reduces speed
- × Increases variability
- × Reduces reliability

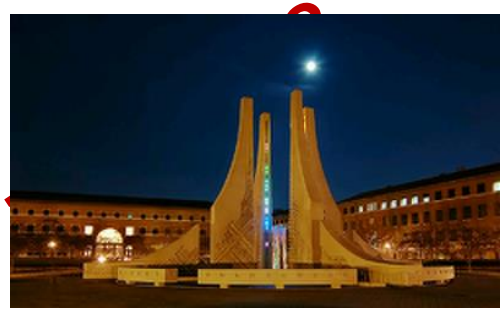
Impact of Error Mitigation

- No hardware protection against errors: erroneous behavior

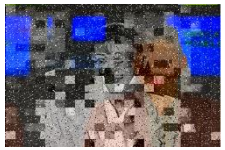
- HW error recovery or frequency scaling: incomplete or corrupted results (missed deadline)

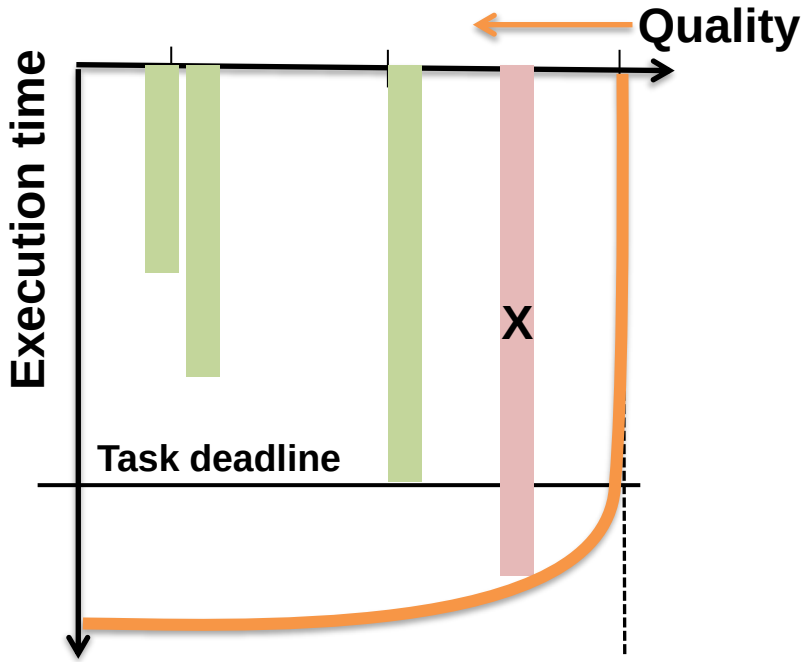
Objective: graceful performance degradation

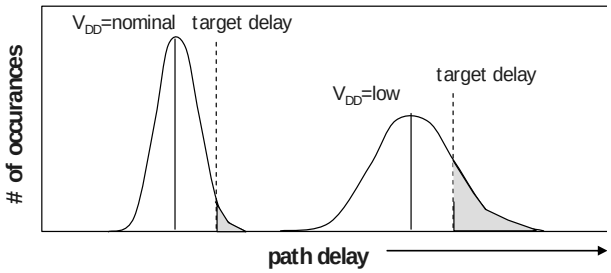
Approximations +



Real-time Systems Need Tuning Beyond the Architectural Level







Consideration of the application level provides additional scalability: graceful performance degradation

Nominal voltage
- ✓ Very fast circuits: sufficient timing margin to handle reliability issues
- ✗ Poor energy efficiency

Reliable near- and sub-VT operation
- ✗ Overhead to meet real-time constraints even without variation
- ✗ Large impact of variations on timing **requires large margin to meet real-time constraints**

- ✓ Approximate computing
- ✓ Scalable algorithms
- ✓ Stochastic computing
- ✓ Application/algorithm-level fault tolerance

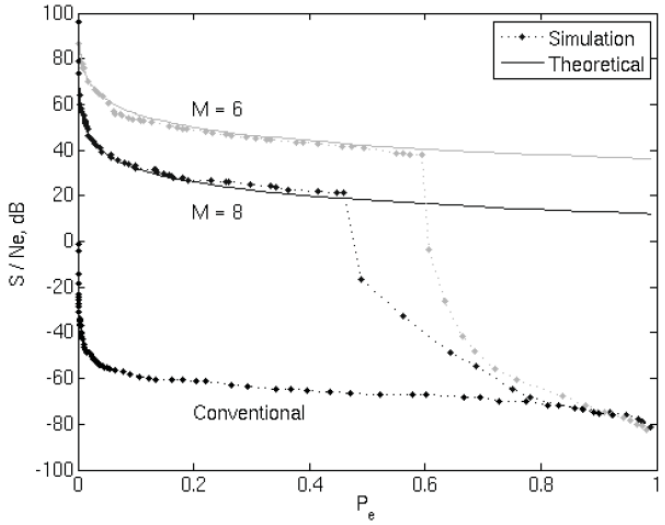
Conventional paradigm: 100% reliable/accurate operation, always meeting real-time constraints

New paradigm: Allow for graceful performance degradation

Why is Assuring Graceful Performance Degradation so Difficult?

Problem: small errors from a hardware perspective do not necessarily translate into small errors from an algorithm/application perspective

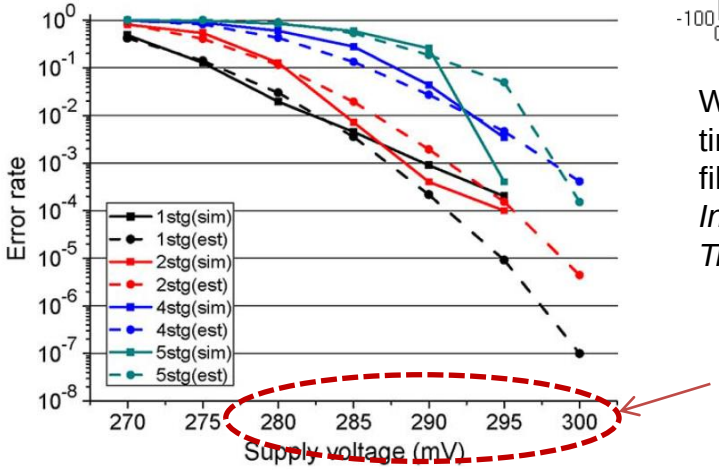
- Datapath: impact of bit-errors depends on
  - Frequency (probability) of error
  - Weight of the affected bit (LSB or MSB)
- Logic/control: impact on data is highly non-linear and can not be captured in mathematical terms



Whatmough, Paul N., et al. "Circuit-level timing error tolerance for low-power dsp filters and transforms." *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 21.6 (2013): 989-999.

Example: multiplier 16x16 bit

Jeon, Dongsuk, et al. "Design Methodology for Voltage-Overscaled Ultra-Low-Power Systems." *IEEE Transactions on Circuits and Systems II: Express Briefs* 59.12 (2012): 952-956.



Error reate increases rapidly over a range of just a few mV

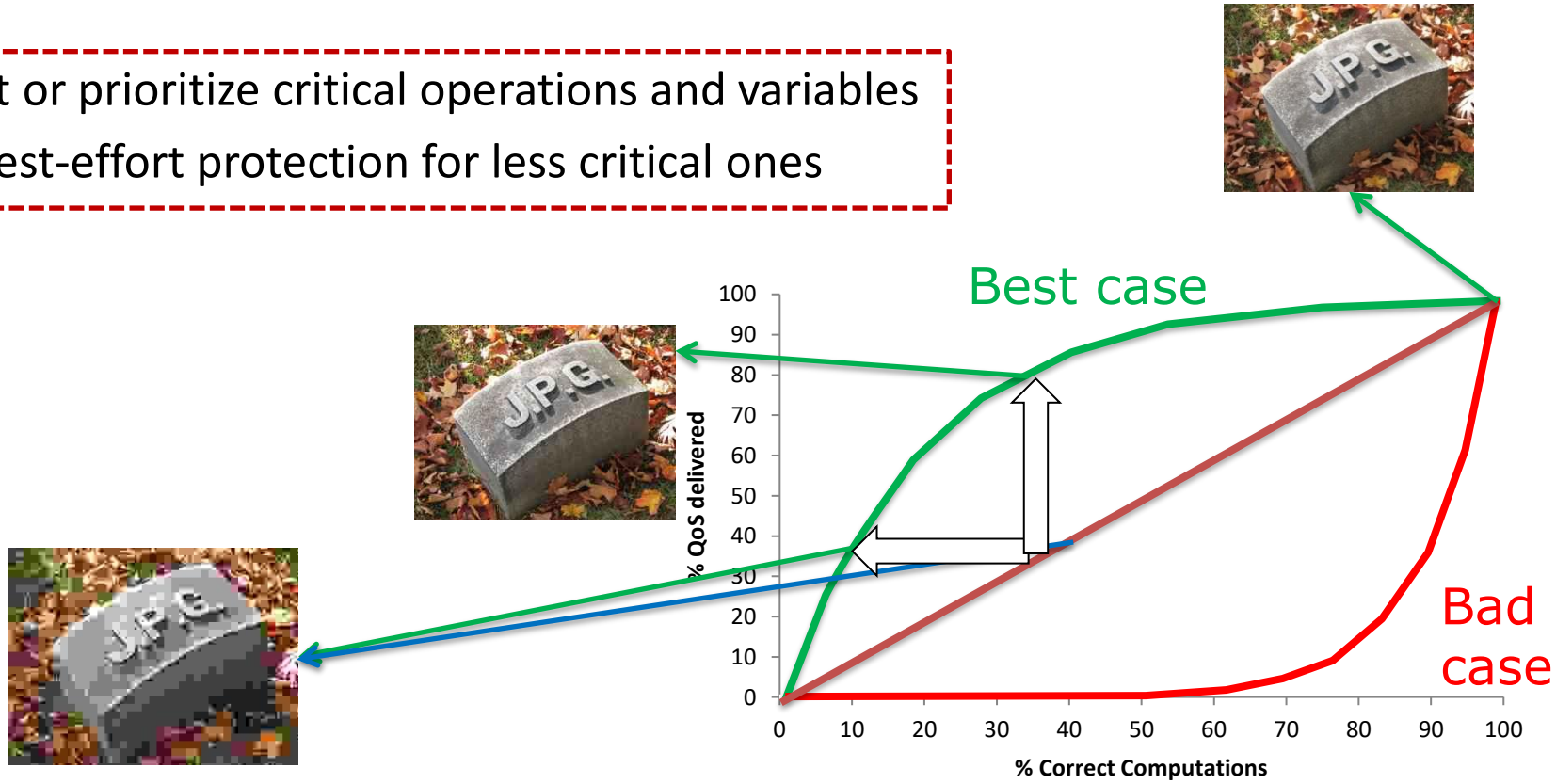
Algorithm Knobs: Significance Driven Design

All computations required for a valid result ✘

Variables/computations contribute equally to QoS

Computations do not contribute equally to the QoS ✓

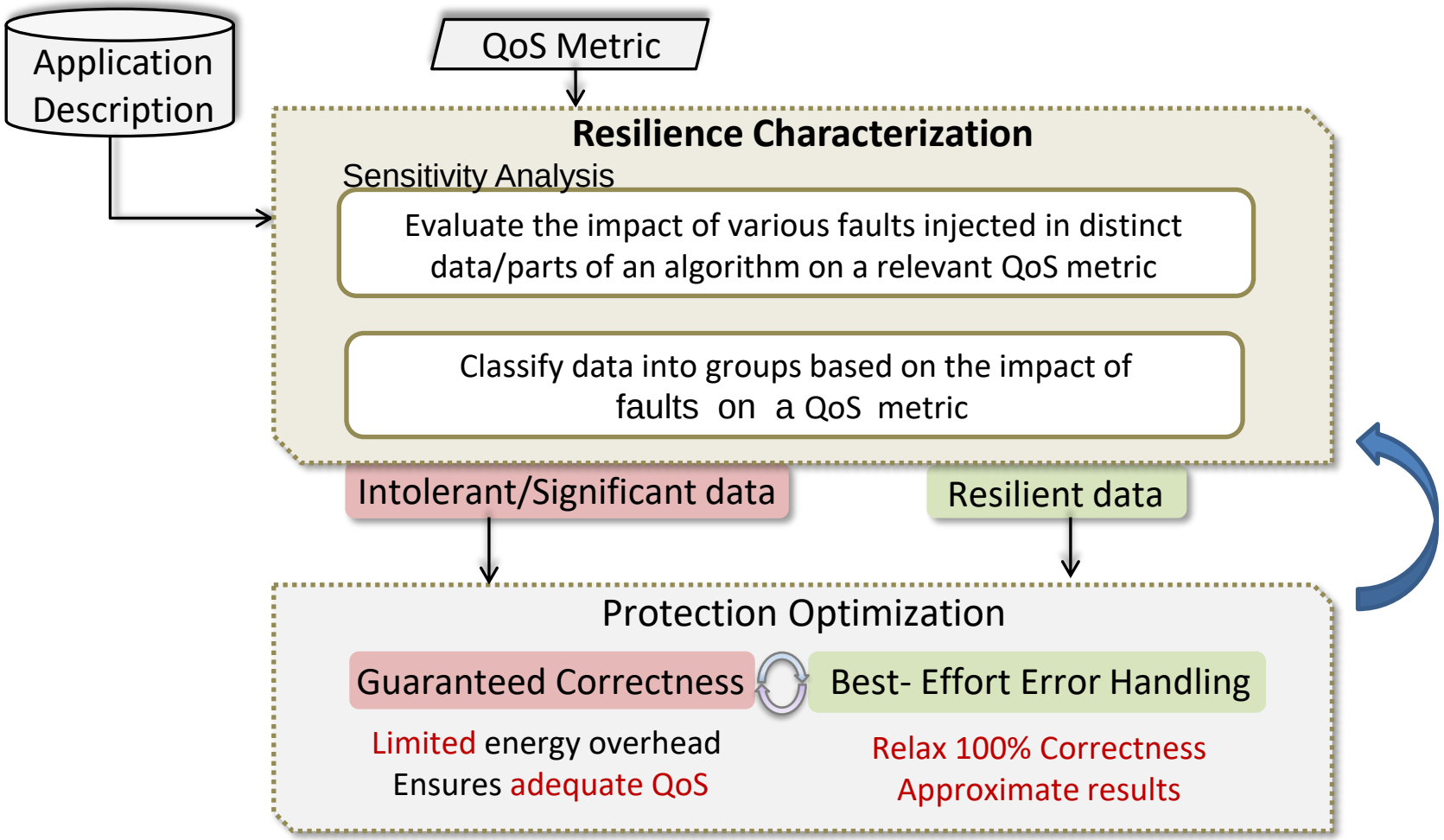
- Protect or prioritize critical operations and variables
- Only best-effort protection for less critical ones



Algorithm Knobs: Significance Driven Design

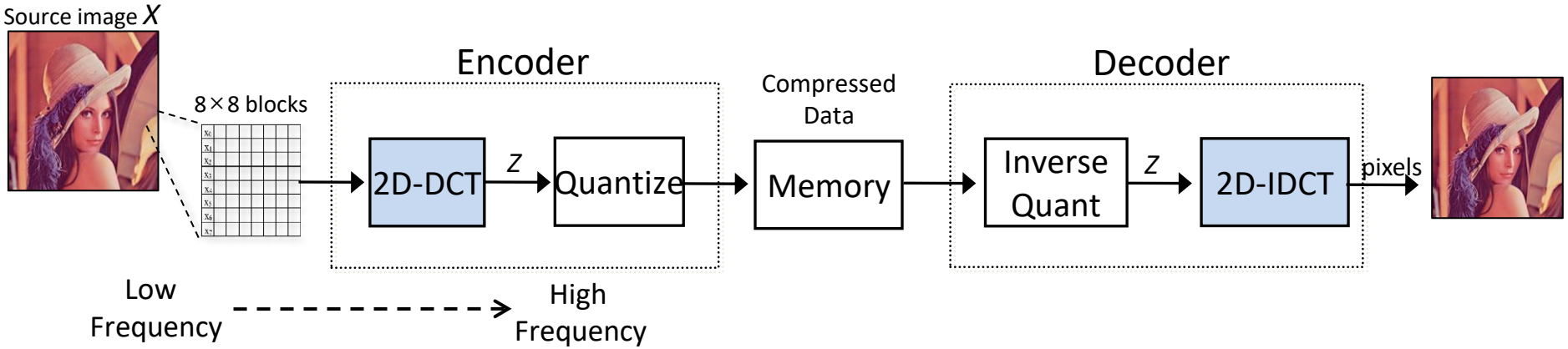
Circuit Level Pruning	Algorithm Level Pruning
• Fine grained / gate-level	• Coarse grained / operation-level
• Significance: defined on a bit-by-bit basis, designed to reflect impact on numeric precision	• Significance: defined directly in terms of numeric precision and error
• Pruning: on a gate-by-gate basis	• Pruning: individual instructions or variables
• Starting point: netlist of arithmetic operations	• Starting point: algorithm and RTL architecture
• Quality-impact-analysis: by simulations on gate-level or with operator overloading	• Quality-impact-analysis: by simulations OR *analytically*
• Area/timing analysis: very accurate	• Area/timing analysis: RTL implementation and estimation

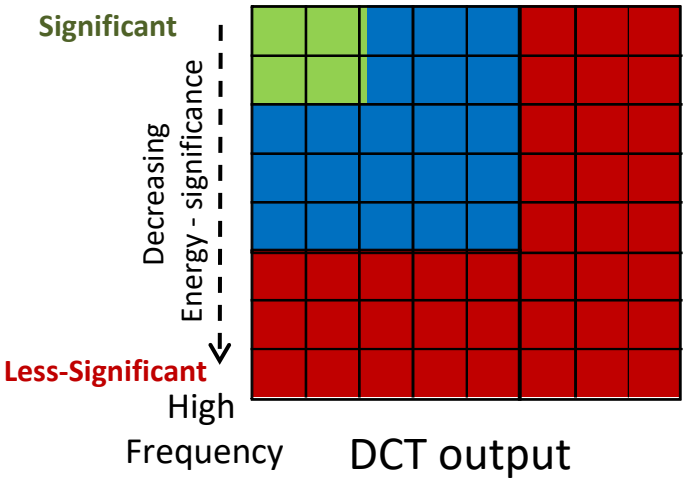
Algorithm Knobs: Significance Driven Design (Analysis)



Minimize the protection overhead while ensuring adequate QoS under any fault

Algorithm Knobs: Significance Driven Design Example (JPEG)



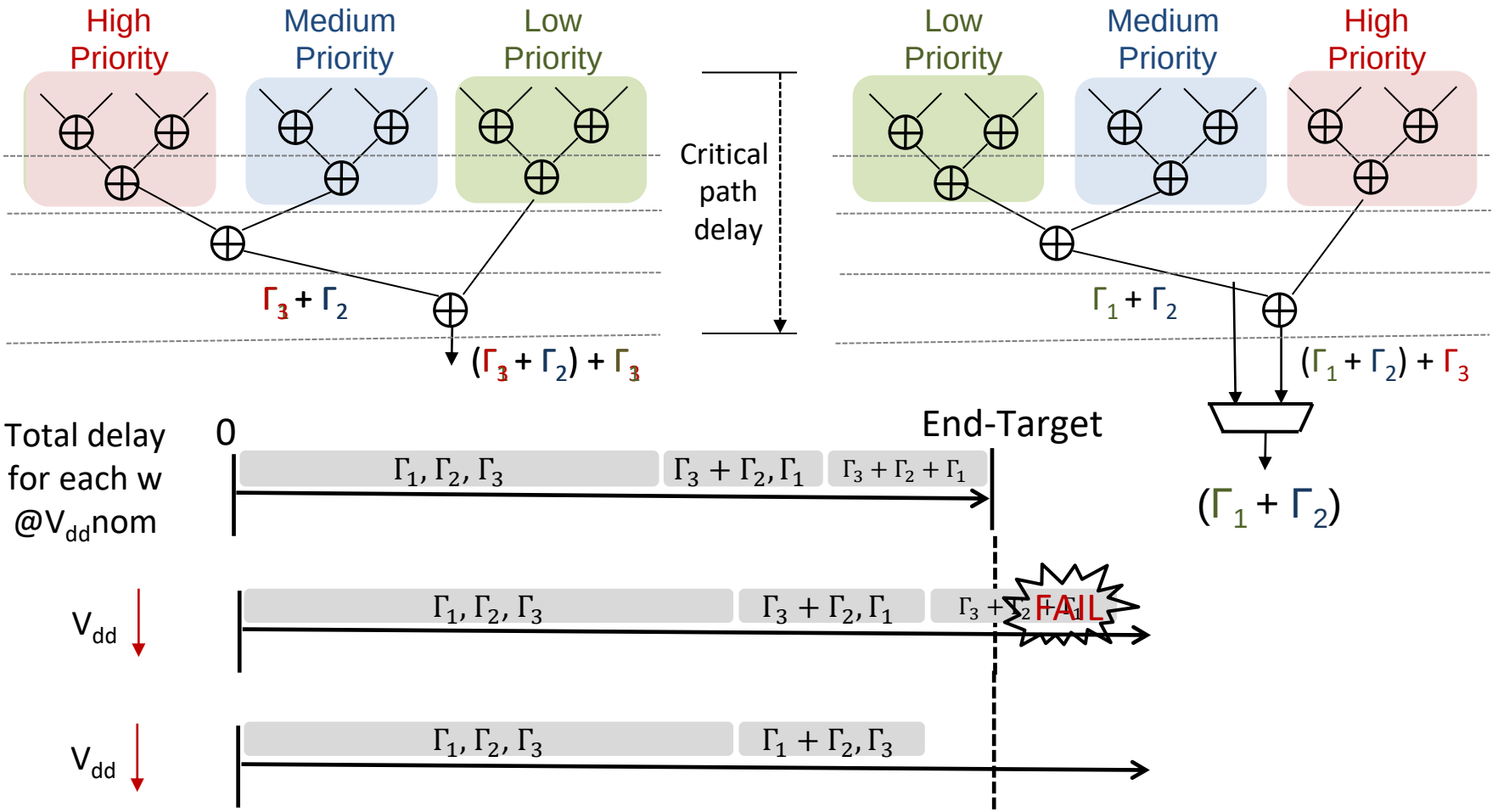


JPEG decompression:

- DCT coefficients have different "significance": carry different energy with different distributions
- Less significant coefficients are close to zero with small variance: can be approximated by zero
- IDCT:

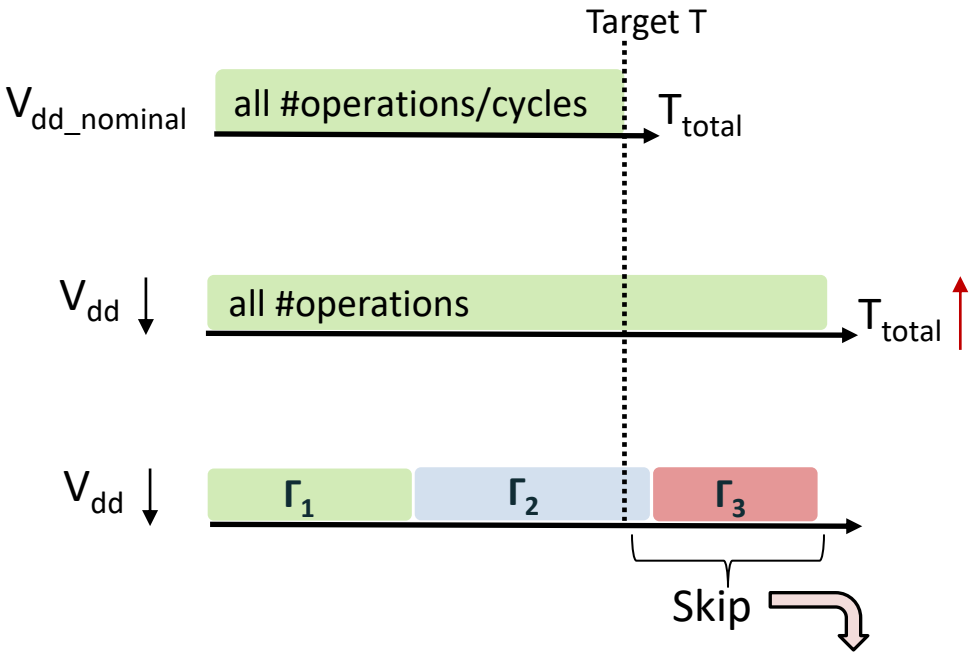
$$I_{x,y} = \underbrace{\begin{bmatrix}\cdot & \cdot\end{bmatrix}\begin{bmatrix}z_0\\ z_1\end{bmatrix}}_{\Gamma_1} + \underbrace{\begin{bmatrix}\cdot & \cdot & \cdot\end{bmatrix}\begin{bmatrix}z_2\\ z_3\\ z_4\end{bmatrix}}_{\Gamma_2} + \underbrace{\begin{bmatrix}\cdot & \cdot & \cdot\end{bmatrix}\begin{bmatrix}z_5\\ z_6\\ z_7\end{bmatrix}}_{\Gamma_3}$$

Algorithm Knobs: Significance Driven Design Example (JPEG)



- Cluster code into groups based on their significance
  - Execute and store only the required data based on the preferred Power, Quality, Performance and hardware capability under variations/scaled-voltages

```
If mode Q3 {
    Compute all terms within Γ1, Γ2 ,Γ3
    … }
            …
else if mode Q1 {
    Compute only the significant Γ1
… }
```



T_{total}=#cycles * $d_{clk}(V_{dd})$

Conventional

Protect all operations by d_{clk} ↑ giving all of them more time to finish

Proposed

Execute correctly by d_{clk} ↑ only the sig. ops and what can be completed within the target time from the rest ops

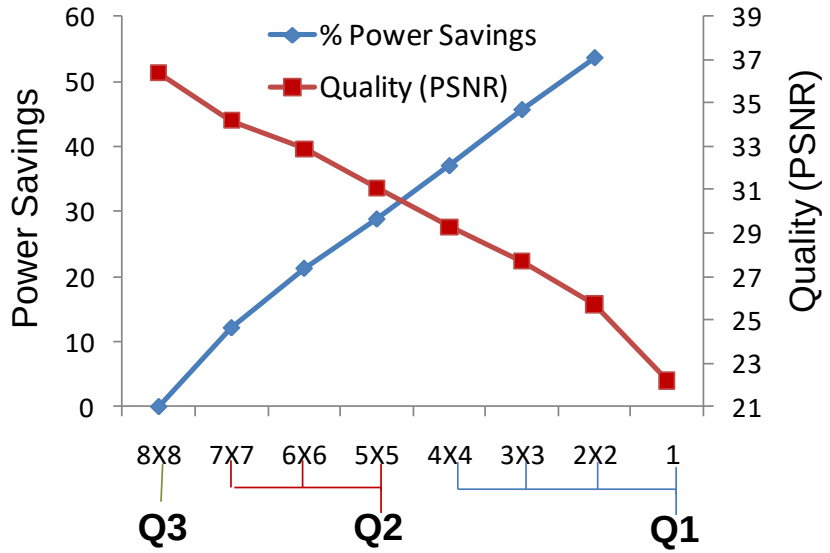
Minor quality loss with no throughput penalty

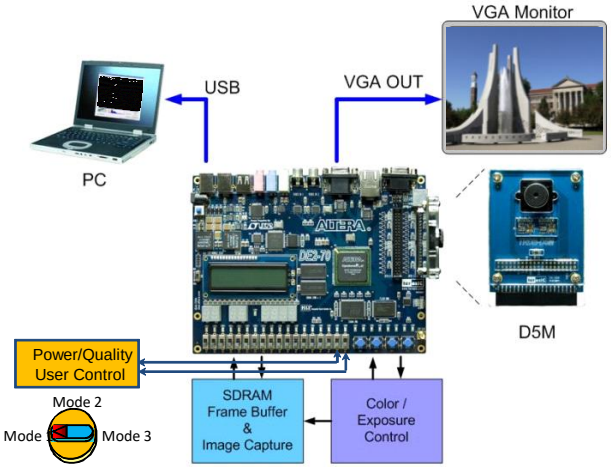
Energy and Quality Scalable Camera System-On-Chip

Modes	Time (sec)	Time (cycles)
Q3(8x8)	5.56336	278168063
Q2(5x5)	4.19744	209872136
Q1(1)	2.32733	116366577

Karakonstantis, et al., DAC'11, Altera Innovate '10

→ Only significant data executed and stored





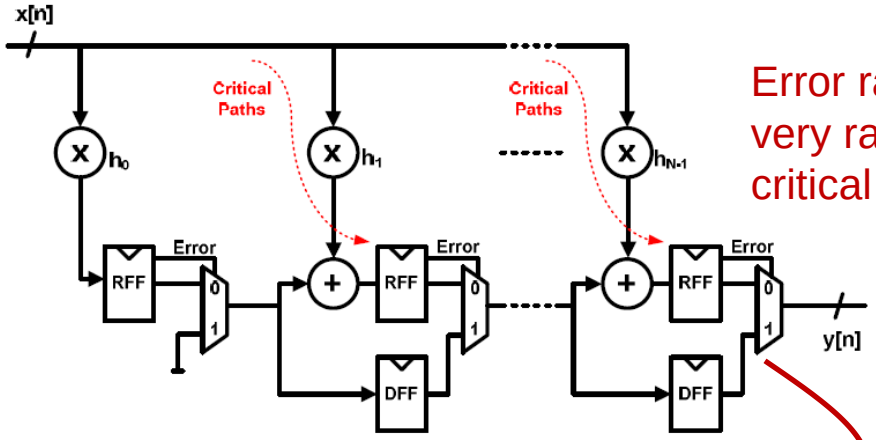
Q3

Q2

Q1

Algorithm Knobs: Detect and Approximate

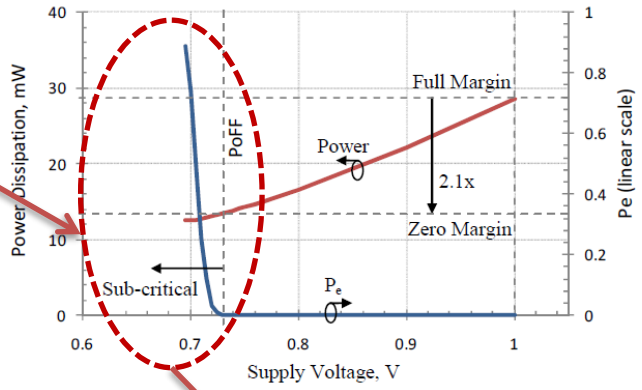
Whatmough, Paul N., et al. "A robust FIR filter with in situ error detection." Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, 2010.

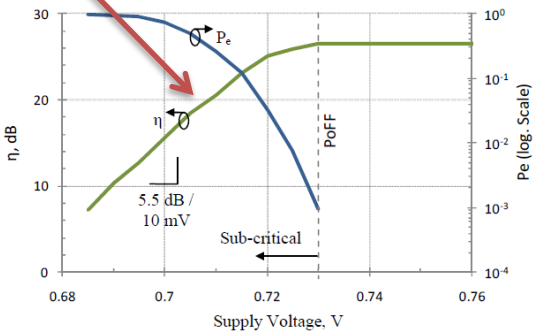


$$y[n] = \sum_{k=0}^{N-1} \begin{cases} h_k x[n-k] & e_k = 0 \\ 0 & e_k = 1 \end{cases}$$

Error rate increases very rapidly for sub-critical timing

Approximate error correction (in-situ)

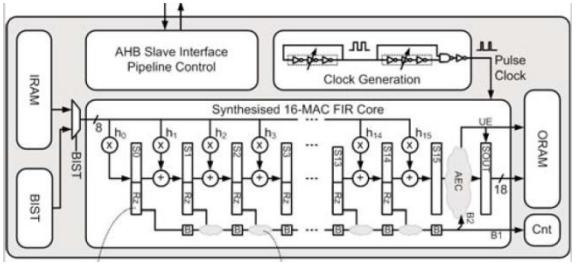




- Signal quality degrades rapidly for sub-critical timing

Possible Solution: time-borrowing

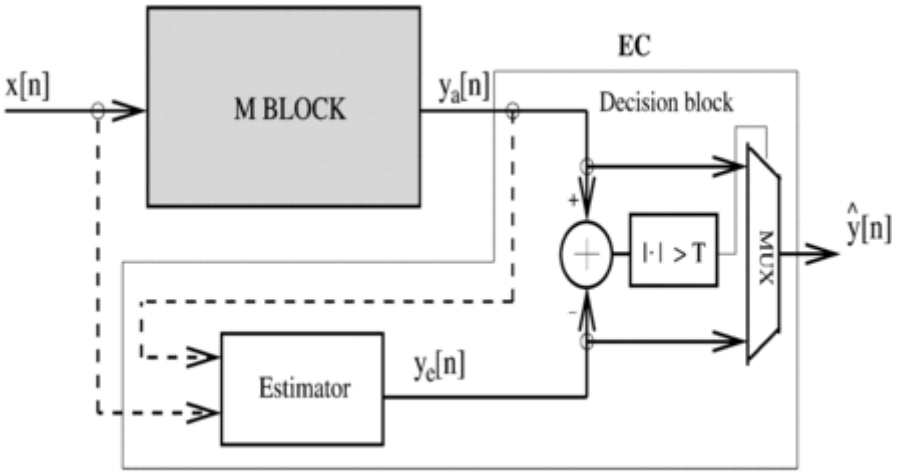
- Consider only timing violations that do not get resolved in the next pipeline stage
- **Improved approximation**



Whatmough, Paul N., Shidhartha Das, and David M. Bull. "A low-power 1GHz razor FIR accelerator with time-borrow tracking pipeline and approximate error correction in 65nm CMOS." *ISSCC, 2013*

Error detection and correction on algorithmic level

- Predict the output of a DSP function (estimation theory)
- Use prediction to perform error detection (compare predicted and computed value)
- Error correction: replace erroneous sample with prediction

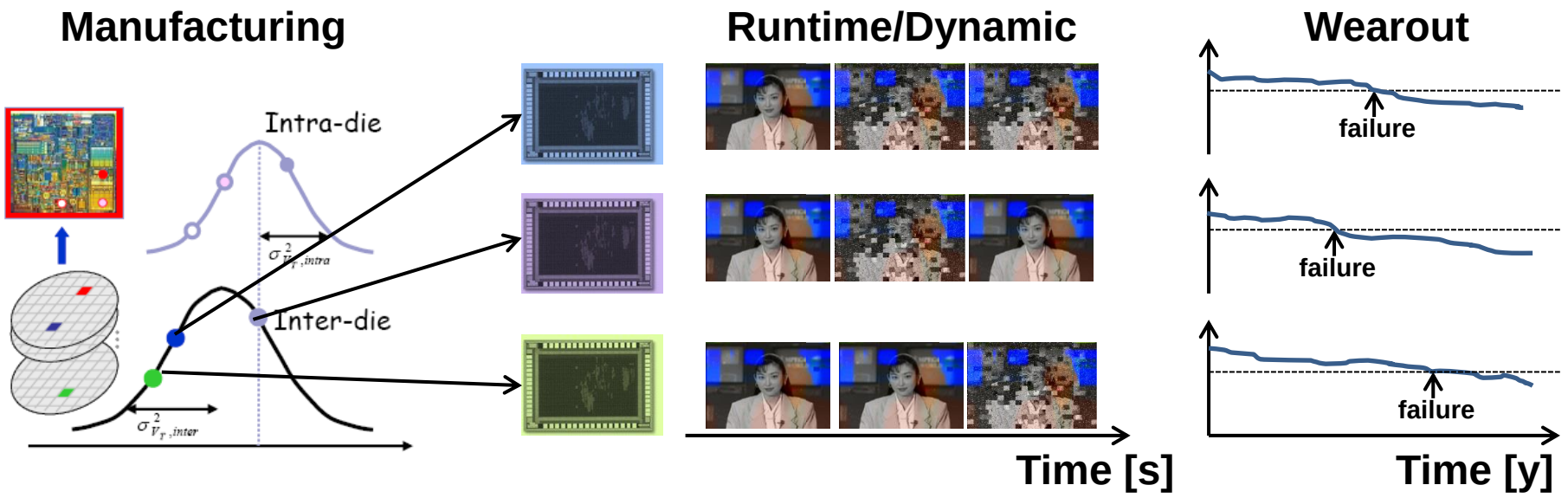


(b)

Byonghyo Shim; Sridhara, S.R.; Shanbhag, N.R., "Reliable low-power digital signal processing via reduced precision redundancy," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.12, no.5, pp.497,510, May 2004 doi: 10.1109/TVLSI.2004.826201

Shanbhag, Naresh R. "Reliable and efficient system-on-chip design." *Computer* 37.3 (2004): 42-50.

Time Scales for Variability



Die to die and within die variations
- Each die is an individual realization of a random process
- Parameters are fixed after manufacturing

Behavior of each circuit *mostly deterministic* and on *short time scale*
- "Randomness" due to random data and model uncertainty
- Averaging only meaningful with true random input

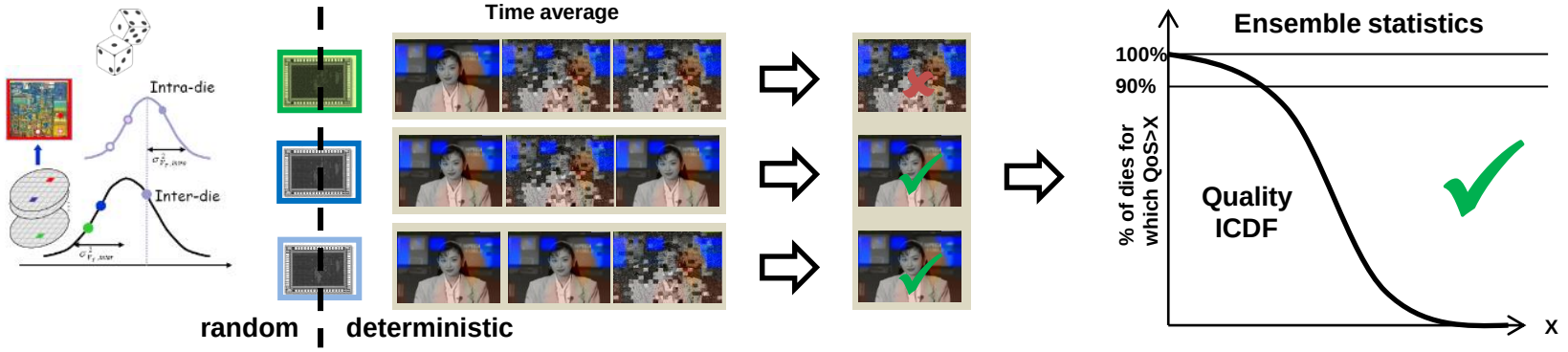
Aging is slow process
- Parameters change on a long time scale
- Long-term average is meaningless

- ***Non-ergodic behavior*** **renders analysis of circuits under variations difficult: averaging requires great care**

Unreliable Silicon: Tradeoff Between Quality and Yield

Average quality across chips is not meaningful

- Need to consider quality individually for each chip:



- Production test: keep only chips that achieve a sufficient minimum quality

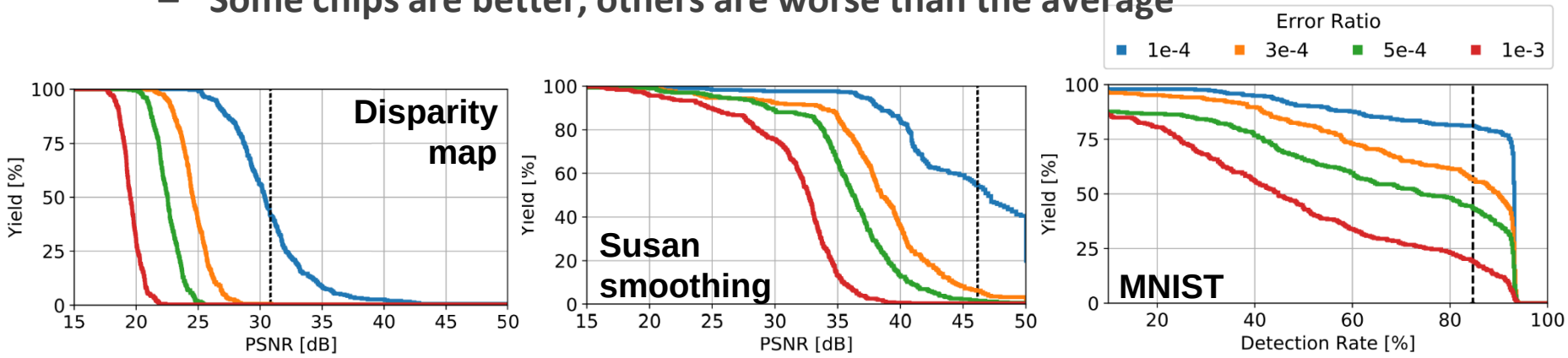
- Inverse CDF (ICDF) of the quality defines a parametric yield
  - Operational meaning: yield for a given minimum quality

Simulation Based Quality-Yield Analysis Reveals Interesting Tradeoffs

Quality-yield analysis (quality CDFs) shows performance across a population of chips

Different chips may have very different average (over input data) quality!!

- **Quality-yield depends strongly on the application**
- **Long tails indicate the presence of significant outliers**
  - **Some chips are better, others are worse than the average**

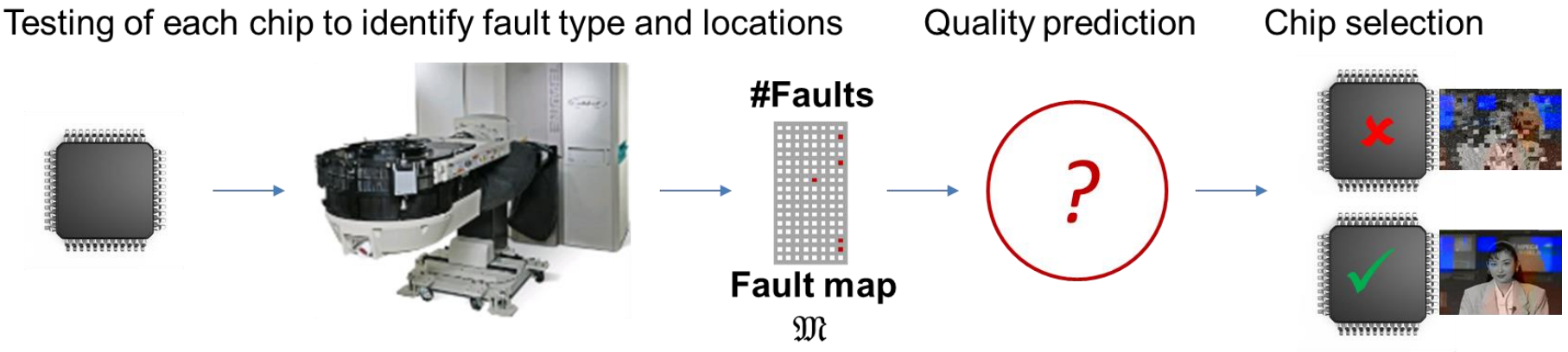


Marco Widmer, Andrea Bonetti, and Andreas Burg. 2019. FPGA-Based Emulation of Embedded DRAMs for Statistical Error Resilience Evaluation of Approximate Computing Systems. In Proceedings of the 56th Annual Design Automation Conference 2019 (DAC '19).

Parametric Test: Straightforward, But Hopeless

Challenge: Different chips provide very different (but fixed) quality levels

- **Straightforward solution: select only well-performing ("good") chips during test**



- **How to predict quality from measured error patterns?**
  - **On-the-spot quality measurement: too time consuming**
  - **Closed-form expression for output quality: not available**
  - **Look-up tables: number of possible patterns prohibitive**

None of these ideas is feasible in mass production