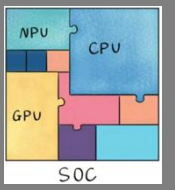


Topic 5

SoC Design Space II

Xinfei Guo
xinfei.guo@sjtu.edu.cn

November 6th, 2024

T5 learning goals

- Chip design space
  - Key metrics
  - Timing and Area
  - Power
  - Reliability

POWER

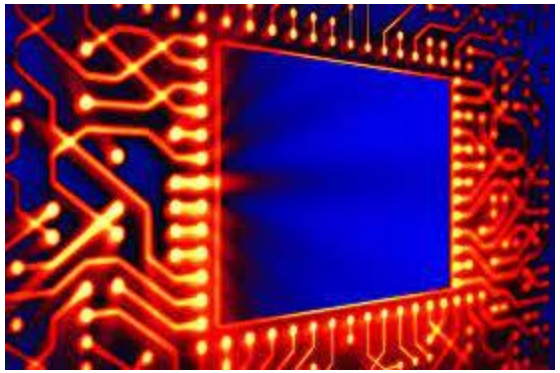
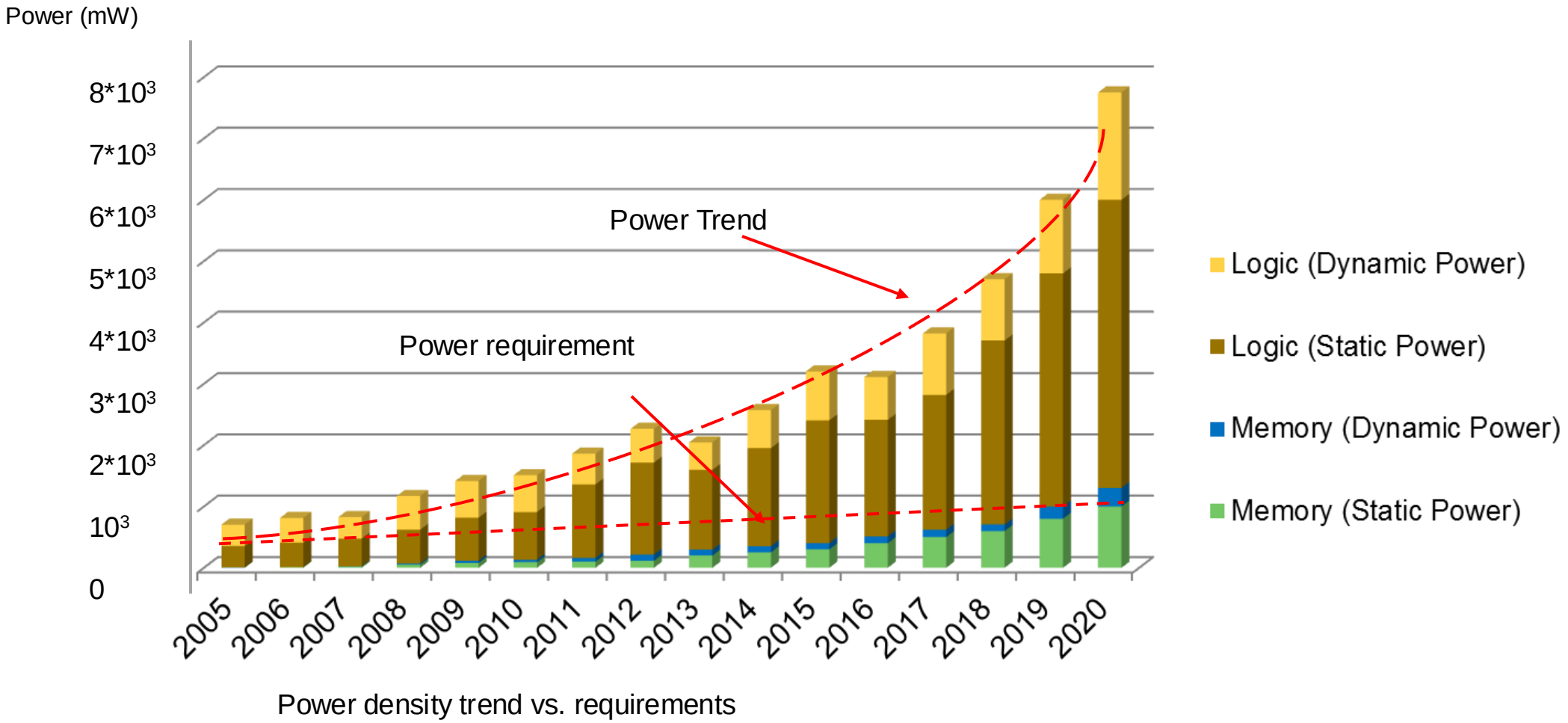


Figure: https://semiengineering.com/new-approaches-for-dealing-with-thermal-problems/

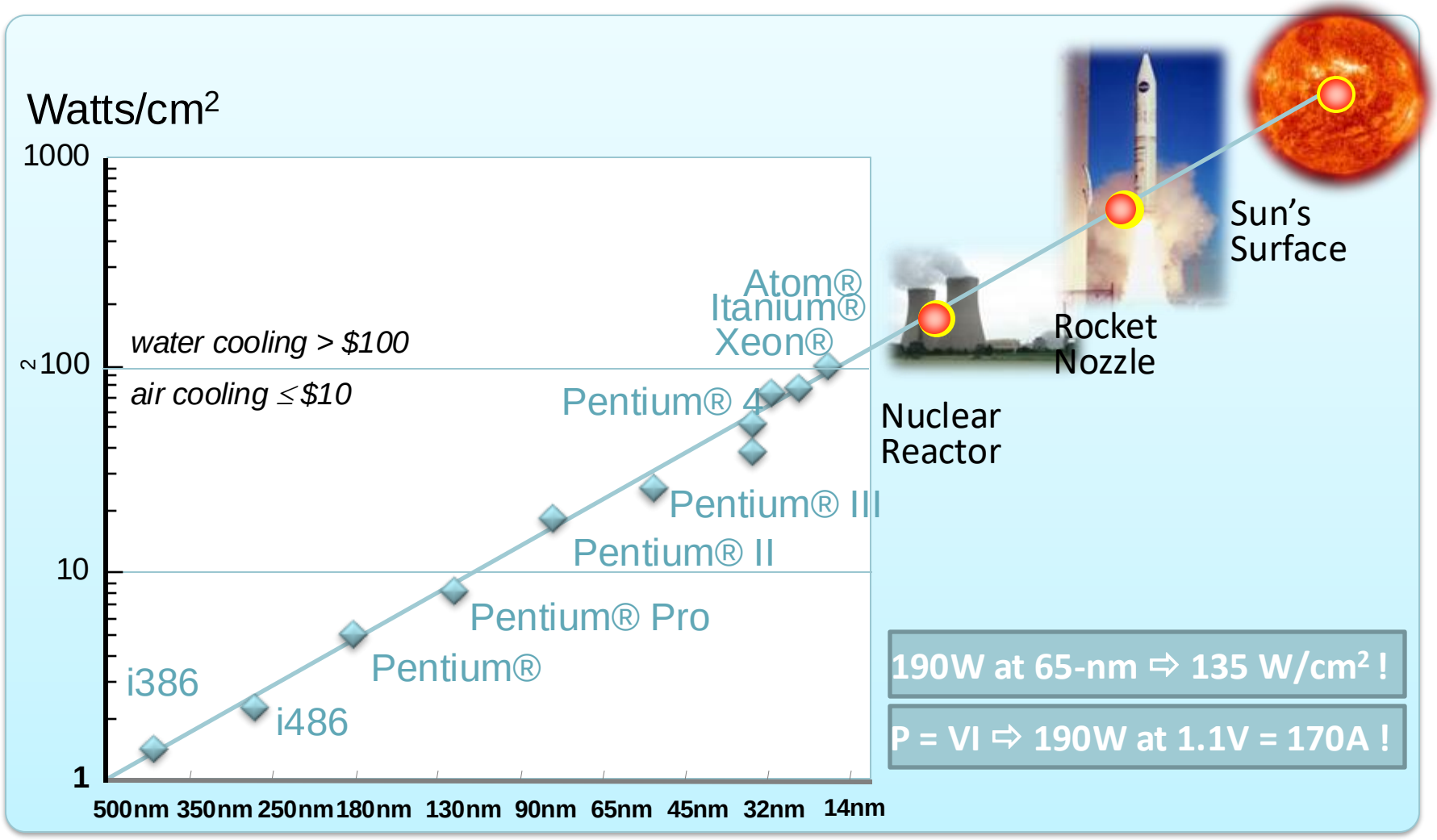
Power Consumption of Integrated Circuits



Power density trend vs. requirements

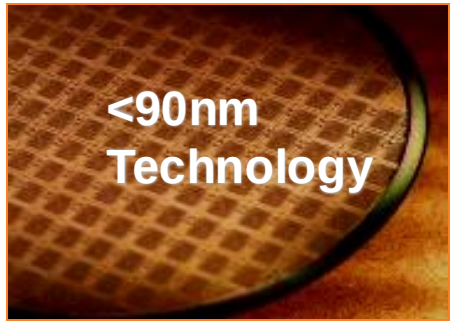
http://www.eetimes.com/document.asp?doc_id=1278448

Power Consumption Trends



Source: Synopsys

Power Affected Problems

	Low Power	Power Efficiency	Reliability
			
Application	• Wireless • Handheld • Embedded systems	• Microprocessors • Graphics/multimedia • Networking/telecom	• All design<90nm
Concern	• Battery life • Leakage power • Dynamic power	• Thermal management • Packaging, cooling cost	• Leakage power • IR-drop • Electromigration

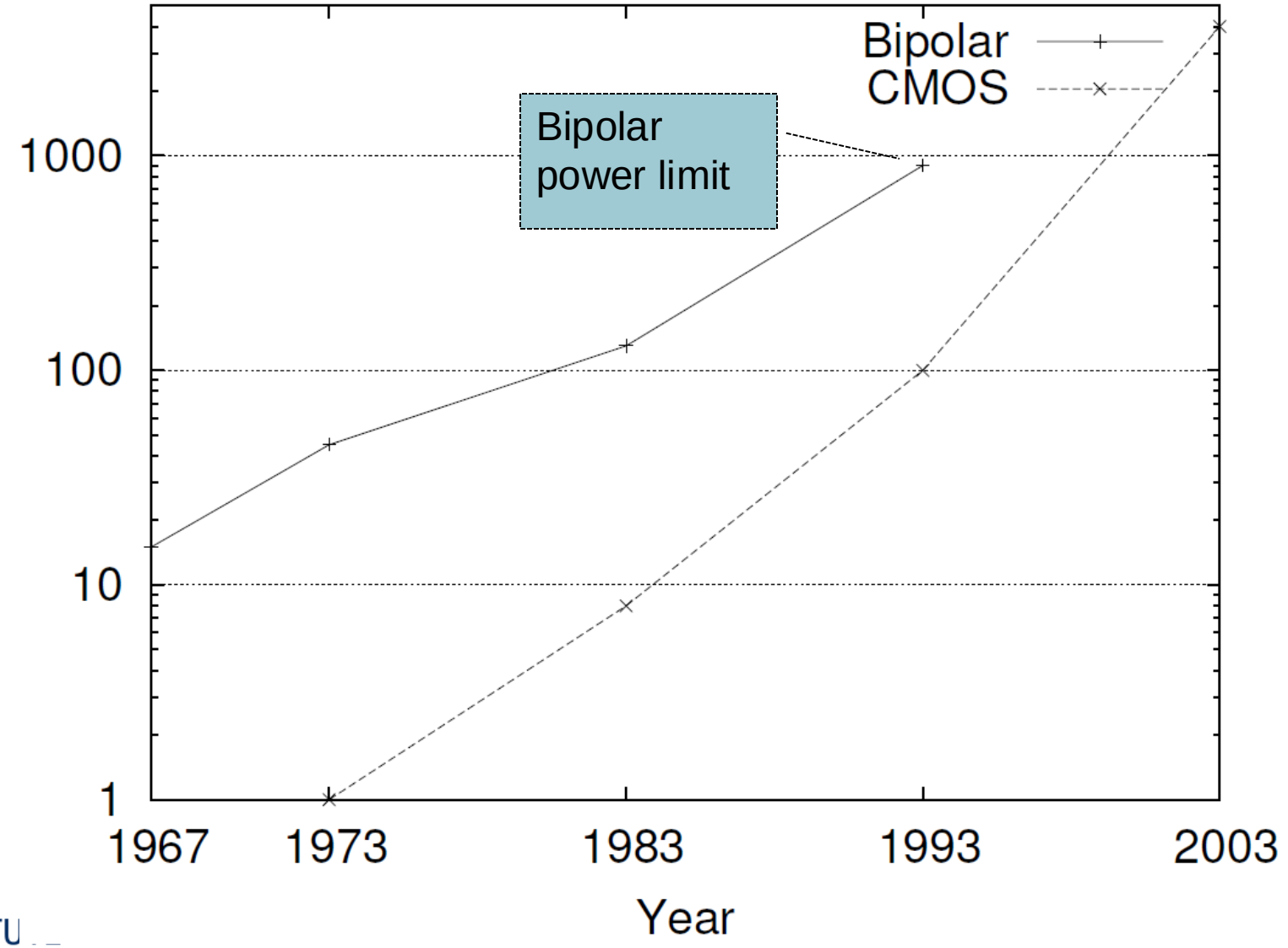
Source: Synopsys

Power consumption

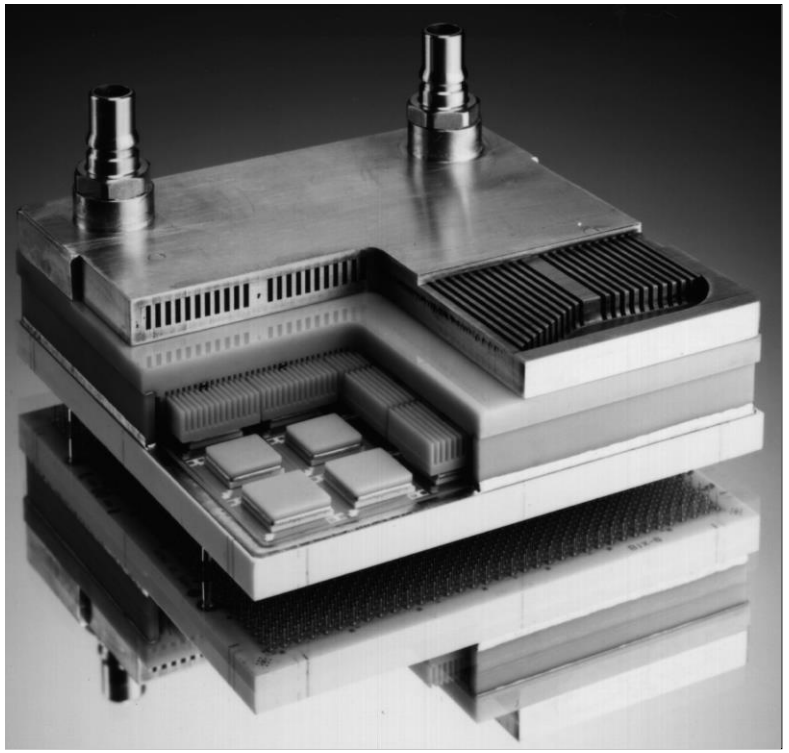
- power consumption: becoming key design issue
- increased power: largely due to higher frequency operation, higher overall capacitance, and larger size
- power = thermal

Type	Power/Die	Source/Environment
Cooled high power	70.0 W	Plug - in, chilled
High power	10.0 – 50.0 W	Plug - in, fan
Low power	0.1 – 2.0 W	Rechargeable battery
Very low power	1.0 – 100.0 mW	AA batteries
Extremely low power	1.0 – 100.0 μW	Button battery

Bipolar and CMOS clock frequency



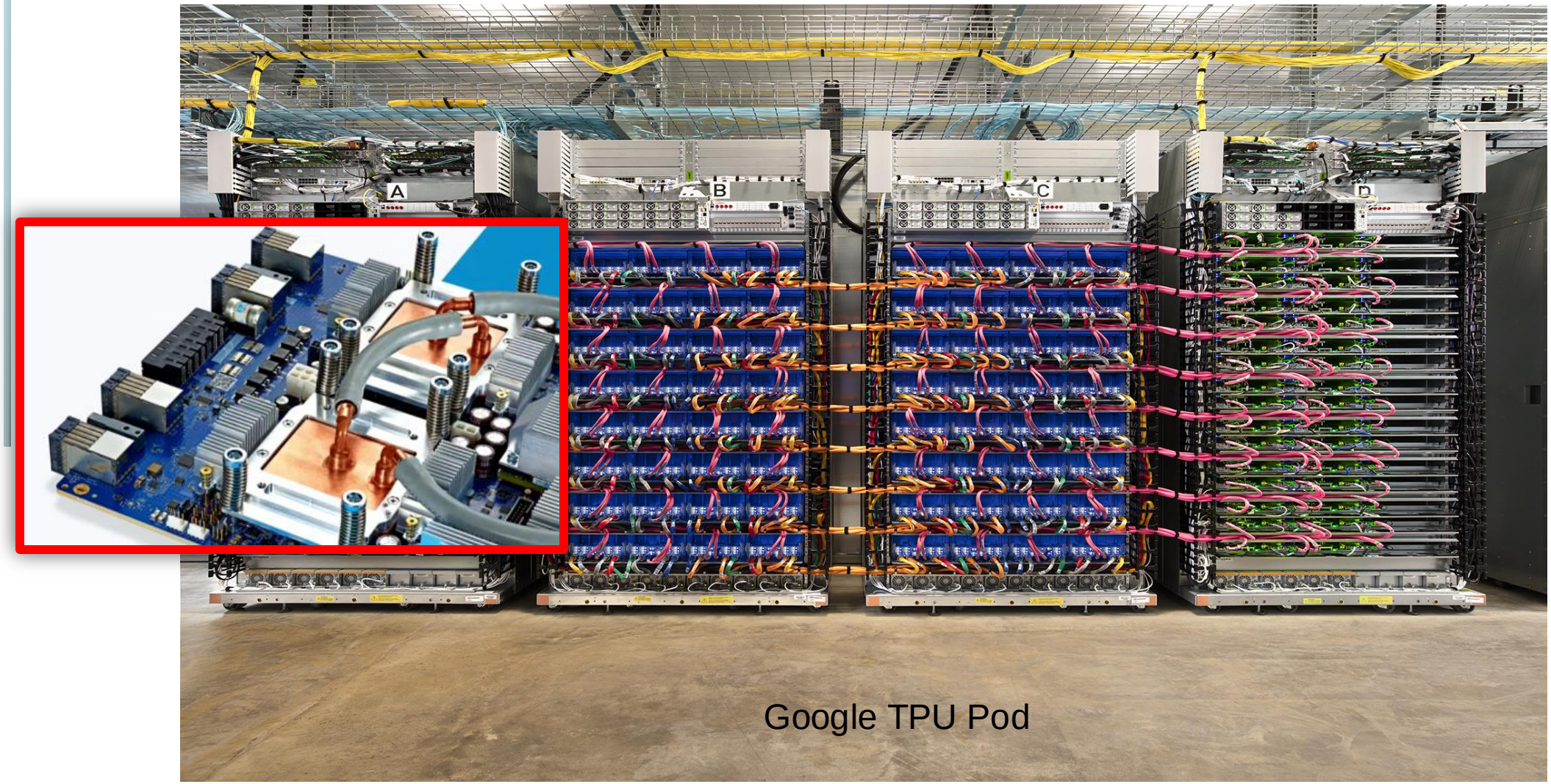
Bipolar cooling technology (ca ’91)



Hitachi M880: 500 MHz; one processor/module, 40 die sealed in helium then cooled by a water jacket.
Power consumed: about 800 watts per module.

F. Kobayashi, et al . “Hardware technology for Hitachi M-880.” Proceedings Electronic Components and Tech Conf., 1991.

Liquid Cooling



Google TPU Pod

Source: Google

Battery energy and usage

- Power dissipation (not performance) is the critical issue for SoC applications such as portable ones running on batteries.

Type	Energy Capacity (mAh)	Duty Cycle/Lifetime	At Power
Rechargeable	10,000	50 h (10–20% duty)	400 mW–4 W
2 × AA	4000	0.5 year (10–20% duty)	1–10 mW
Button	40	5 years (always on)	1 μW

Battery Capacity and Duty Cycle

Battery Types Grouped by Application (1)

Primary batteries

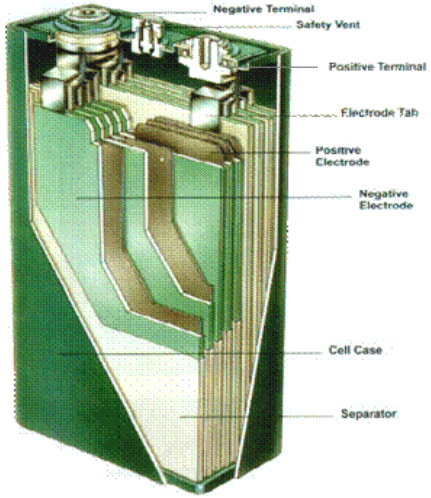
- General consumer electronics (portable audio equipment, toys, etc.)
  - Alkaline (Duracell, Energizer, etc.) – Standard AAA, AA, C, D, 9V cells
  - Zinc-Carbon (old technology, but cheap) – Standard AAA, AA, C, D, 9V cells
- Film cameras and flash units
  - Alkaline
  - Lithium
- Wristwatches
  - Silver – “Button” batteries
- Hearing aids
  - Zinc-Air – “Button” or “coin” batteries
- Smoke detectors
  - Mercury
  - Lithium
- CMOS memory backup
  - Lithium
- Medical implants (pacemakers, etc.)
  - Mercury – Used in implants before 1972
  - Zinc-Air – Used in many modern implants
  - Lithium-SVO (silver vanadium oxide) – used in implantable defibrillators, where they can supply microamps for years and occasional amp-level pulses.

source: synopsys

Battery Types Grouped by Application (2)

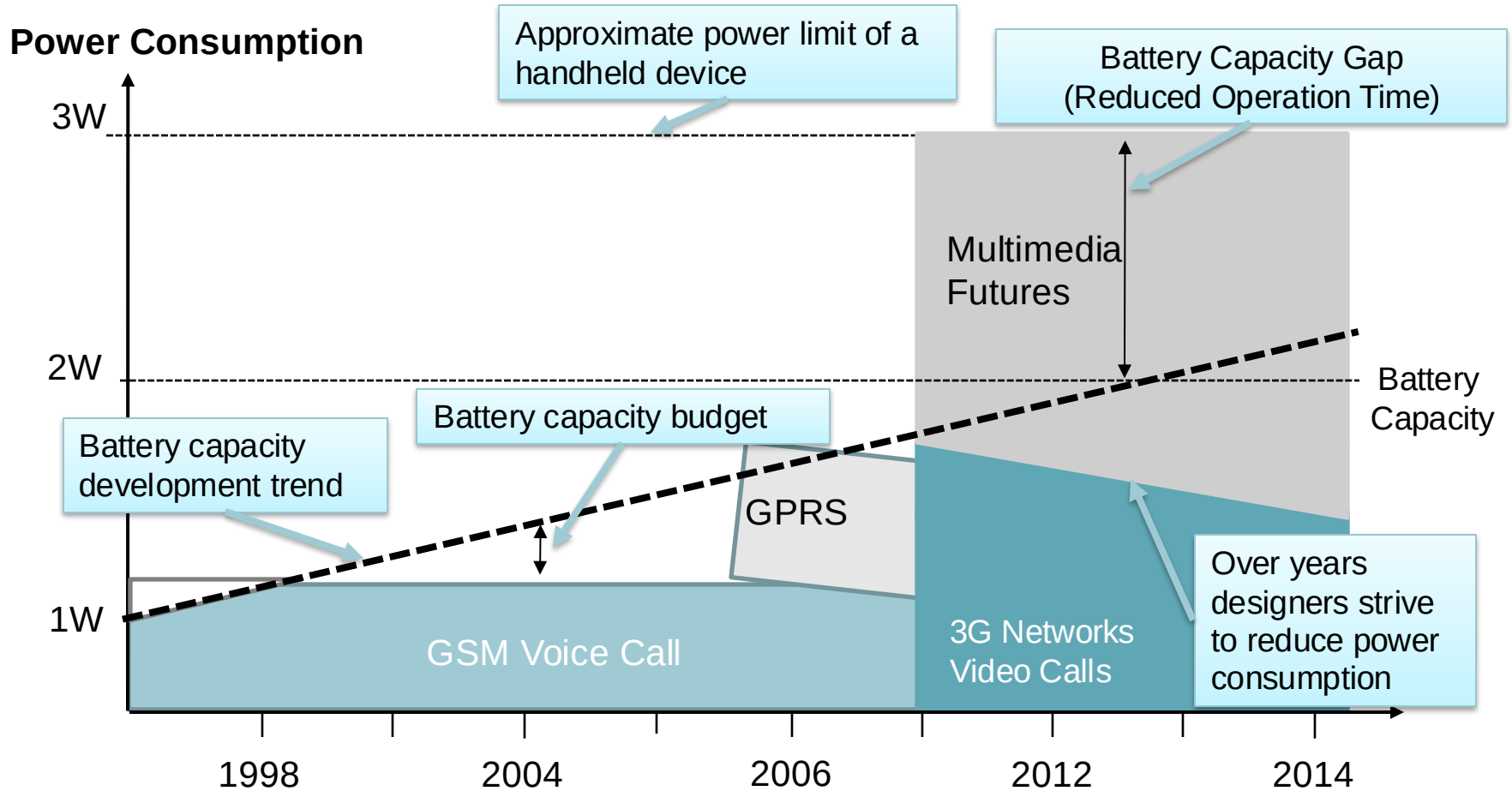
Secondary batteries

- General consumer electronics (portable audio equipment, toys, etc.)
  - Nickel-Cadmium (Nicad) – Available in standard AAA, AA, C, D, 9V cells
  - Nickel-Metal Hydride (NiMH) – Available in standard AAA, AA, C, D, 9V cells
- Cell phones
  - Nickel-Metal Hydride (NiMH)
  - Lithium Ion
- Laptops
  - Nickel-Metal Hydride (NiMH)
  - Lithium Ion
- Palmtops
  - Lithium Ion
- Handheld video recorders
  - Lead acid (older models)
  - Lithium Ion
- Gasoline automobiles
  - Lead acid
- Electric/hybrid automobiles
  - Lead acid (General Motors EV1 electric car)
  - Nickel-Metal Hydride (NiMH) (newer EV1; Honda Insight hybrid gas/electric car)



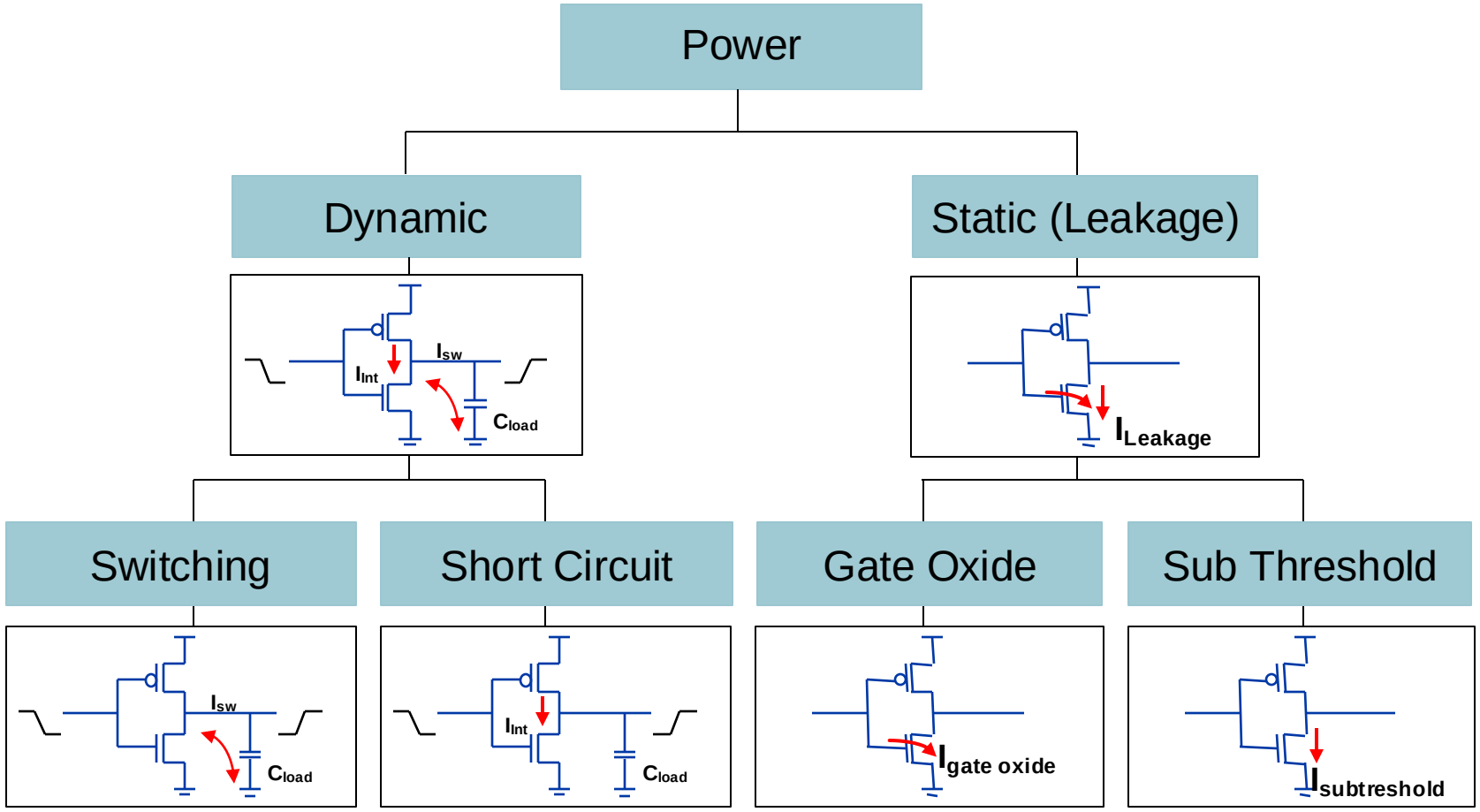
source: synopsys

Power Consumption & Battery Capacity Trends



source: synopsys

Sources of power



Source: Synopsys

Recall: CMOS Power Equation

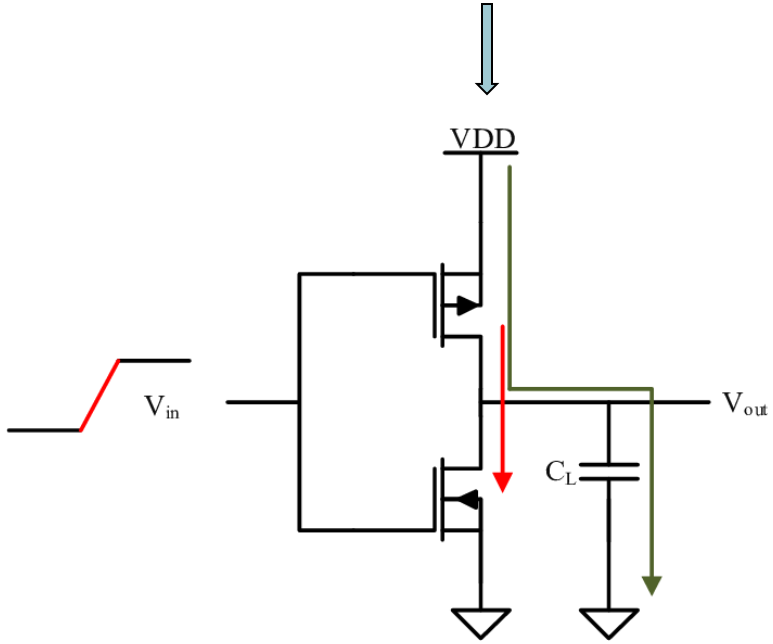
1/2 typically, but not always

$$P = \alpha . f . C_L . V_{DD}^2 + f . I_{peak} . V_{DD} + V_{DD} . I_{leakage}$$

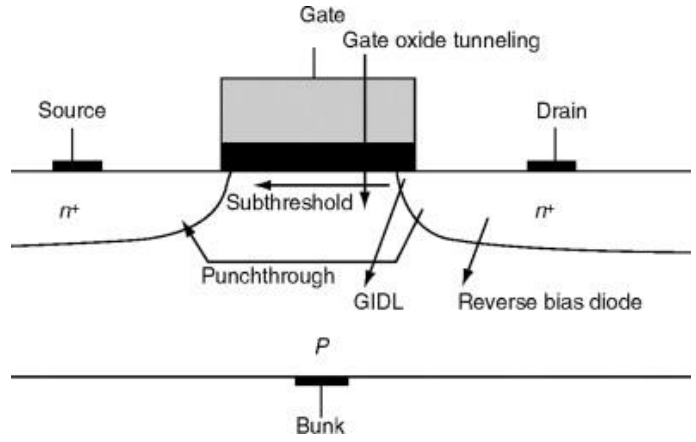
Dynamic power

Short-circuit power

Leakage power



$$I_{sub} = \mu_0 \cdot C_{ox} \cdot \frac{W}{L} \cdot V^2 \cdot e^{1.8} \cdot e^{\frac{(V_{gs} - V_T)}{nV}}$$



Recall: Dynamic Power

Capacitance:
Function of fan-out (gates), wire length, transistor sizes

Supply Voltage:
Has been dropping with successive generations

$$P_{dyn} = C_L V_{DD}^2 \alpha f$$

Activity factor:
How often, on average, do wires switch?

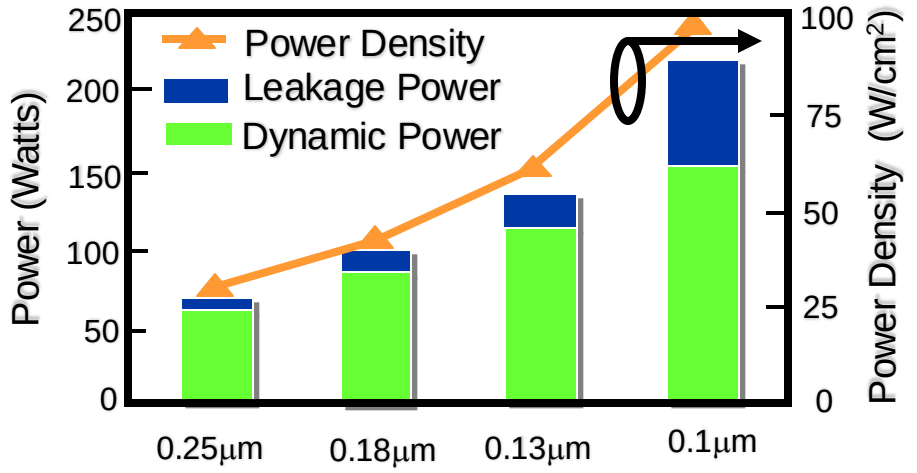
Clock frequency:
Increasing…

Power and Feature Size

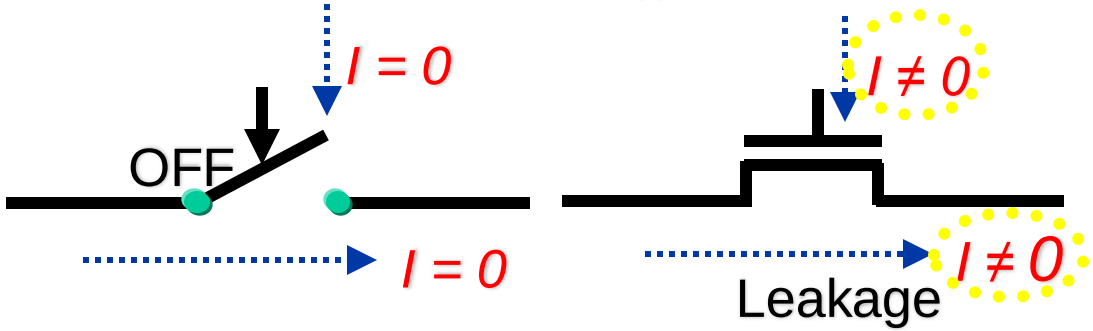
- A feature size decrease results in lower device size.
- Smaller device sizes will reduce the capacitance.
- Low capacitance decreases the dynamic power consumption and gate delays (both).
- As device size decreases, the electric field applied becomes destructively large.
- To increase the device reliability, we need to reduce the supply voltage, V.
- Low Voltage will effectively reduce the dynamic power consumption but results in an increase in gate delays.
- Gate delays increase can be avoided by reducing, Vth
- On the other hand, reducing Vth will increase the leakage current and, therefore leakage power consumption.

Scaling Challenge: Leakage Power

- Leakage issues:
  - Power dissipation
  - Robustness of dynamic logic circuits



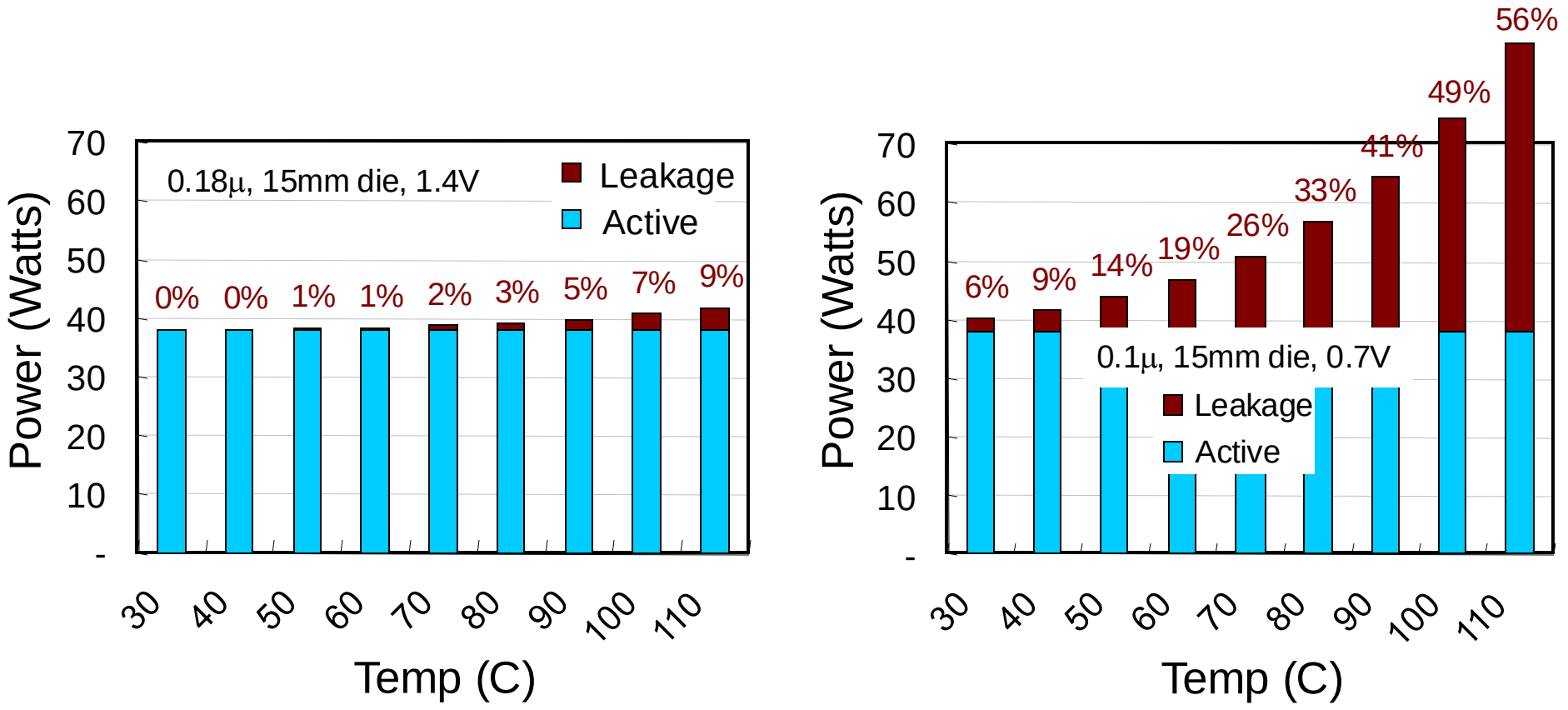
Source: Intel



Need for low power design, leakage reduction, and leakage tolerant design

source: synopsys

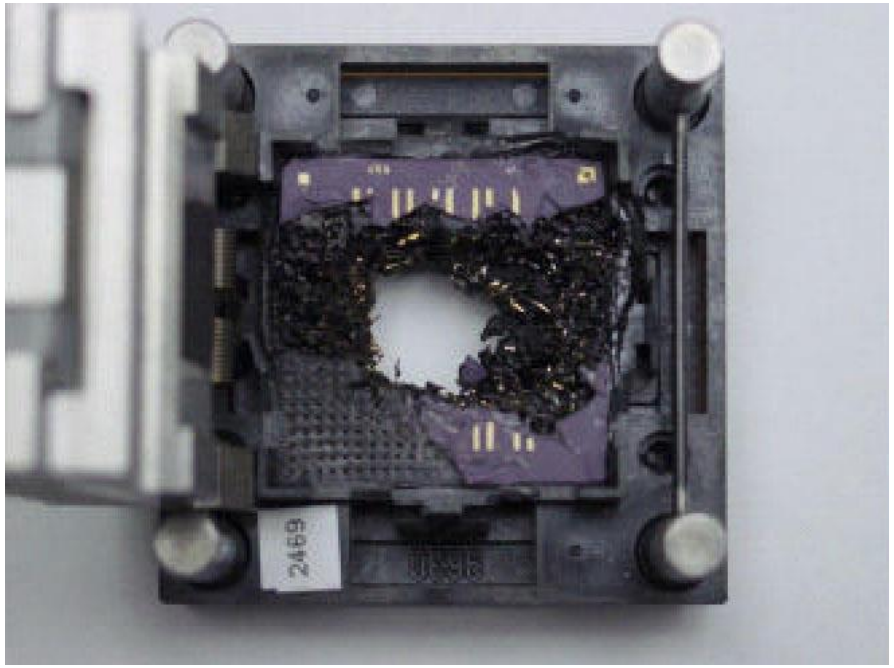
Leakage Power vs. Temp.

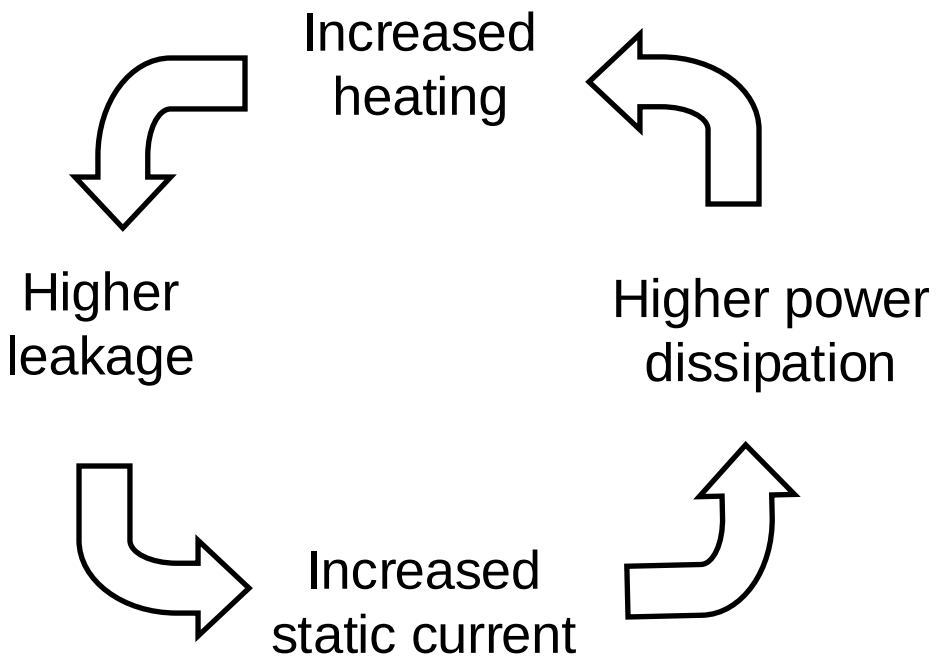


- Leakage power is more problematic in active mode for high performance microprocessors and SoCs

source: synopsys

Thermal Runaway





- Destructive positive feedback mechanism
- Leakage increases exponentially with temperature
- May destroy the test socket → thermal sensors required

source: synopsys

Power and frequency

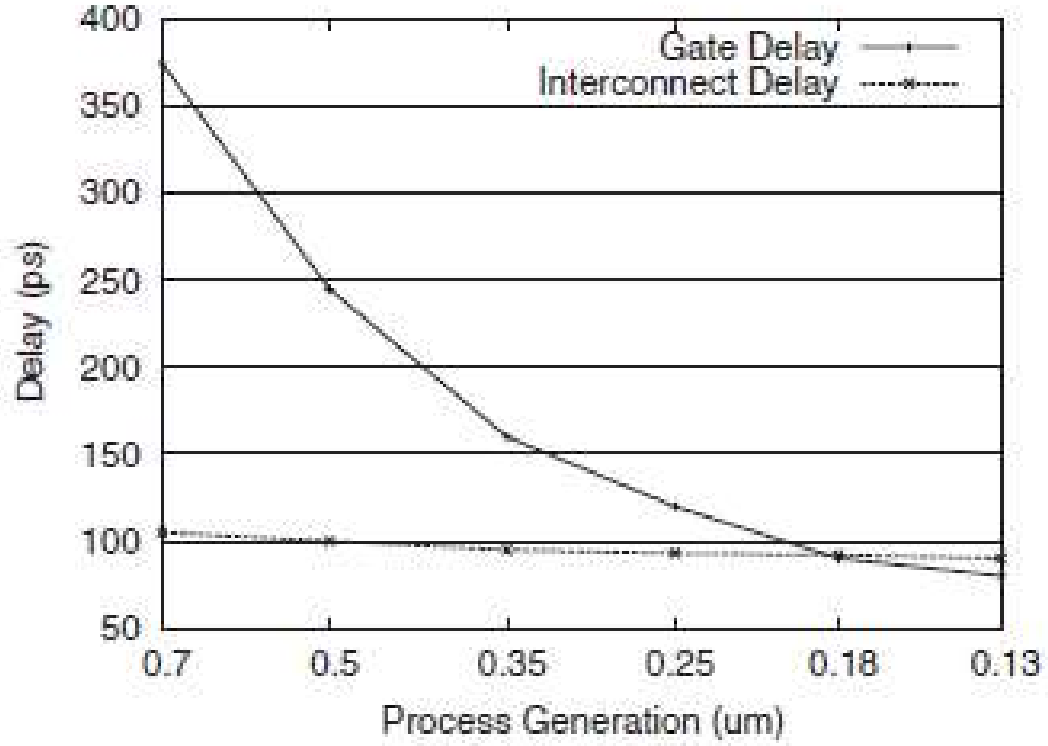
- I = C dV/dt ….smaller C enables higher dV/dt (frequency)
- but $I = (V - V_{th})^2/V$ and I also directly determines max. frequency.
- for V_{th} = 0.6v , halving V also halves the frequency. (E.g. if V goes from 3 to 1.5v then freq is ½)
- so halving the voltage (V_{DD} or the signal V) halves the frequency BUT reduces the ***dynamic power*** by 1/8 … ($CV^2f/2$)
- so

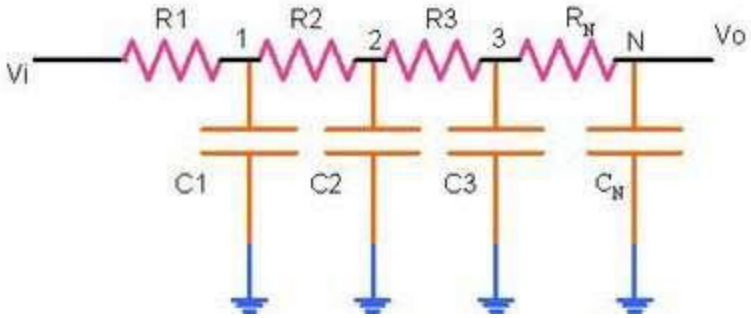
$$\frac{freq_1}{freq_2} = \sqrt[3]{\frac{P_2}{P_1}}$$

- We can optimize an existing design for frequency and modify that design to operate at a lower voltage.

Interconnect Power

- Although gate delay scales with technology generation, wire delays do not scale at the same rate
- SoC does have a lot of wires!





Area–Time–Power Tradeoff

- Workstation Processor: Designs are high-clock based AC power sources. (excluding Tabs/Laptops)
- Cache occupies large die area and consumes power.
- CPU designs are complex (superscalar, multi-core, etc). Need ample power.
- SoC Processors: Generally simpler in control. May be complex in execution facilities (DSP). Area is a factor as well as the design time and power.

	Processor on a Chip	SOC
Area used by storage	80% cache	50% ROM/RAM
Clock frequency	3.5 GHz	0.5 GHz
Power	≥50 W	≤10 W
Memory	≥1-GB DRAM	Mostly on-die

Push for the low-power SoC

- While leading companies were searching for other sources of low power, SoC designers had only one way to meet the market demands
  - Find new ways to reduce IC power and by its application design super-low power ICs, simultaneously providing high functionality.

Dynamic Power Reduction

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

Dynamic Power Consumption Data Dependent

- Switching activity, $P_{0\rightarrow1}$ has two components
  - A static component – function of the logic topology
  - A dynamic component – function of the timing behavior (glitching)

2-input NOR Gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Static transition probability

$P_{0\rightarrow1} = P_{out=0} \times P_{out=1} = P_0 \times (1-P_0)$

With input signal probabilities

$P_{A=1} = 1/2$

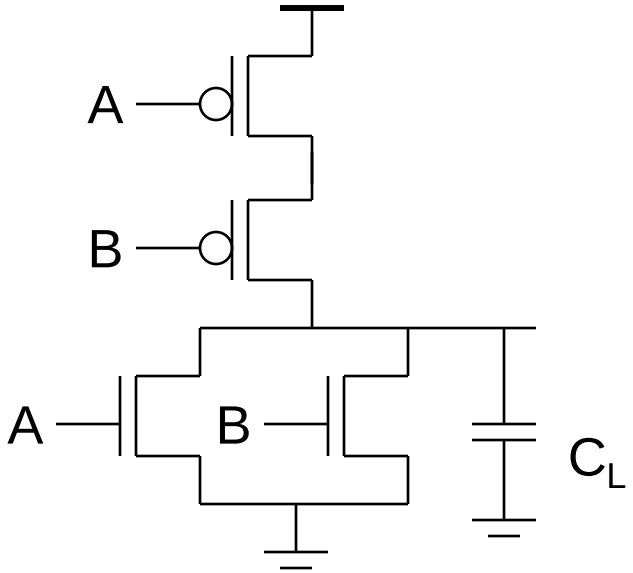
$P_{B=1} = 1/2$

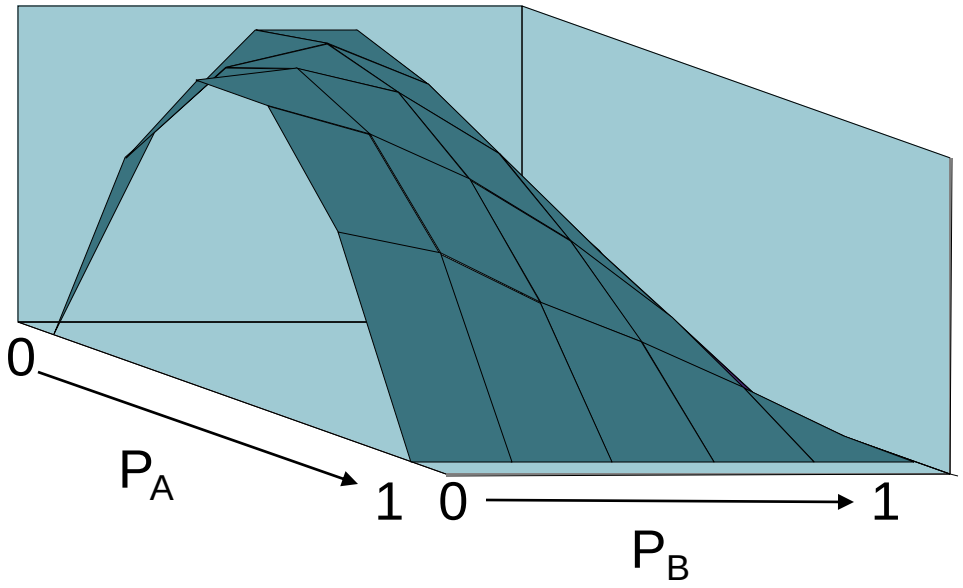
NOR static transition probability

$= 3/4 \times 1/4 = 3/16$

NOR Gate Transition Probabilities

- PA and PB are the probabilities the inputs A and B of which are one



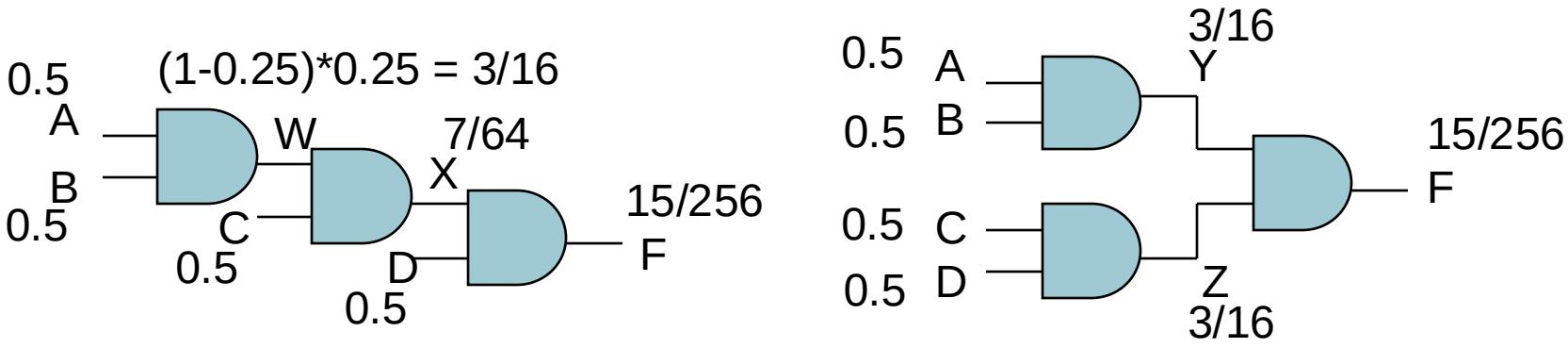


$$P_{0\rightarrow1} = P_0 \times P_1 = (1-(1-P_A)(1-P_B))\ (1-P_A)(1-P_B)$$

Logic Restructuring

- Logic restructuring: changing the topology of a logic network to reduce transitions

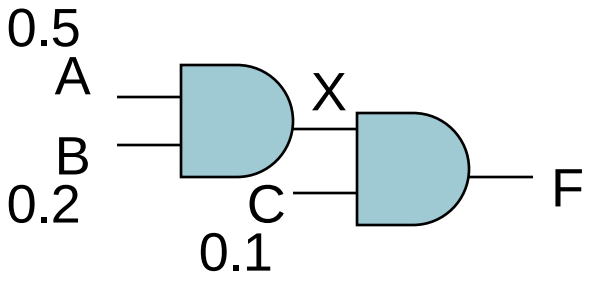
  AND: $P_{0\rightarrow1} = P_0 \times P_1 = (1 - P_A P_B) \times P_A P_B$

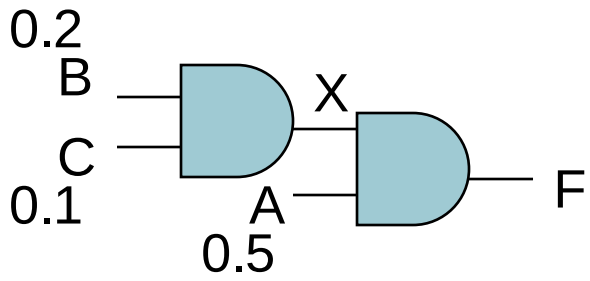


- Chain implementation has a lower overall switching activity than the tree implementation for random inputs

  Ignores glitching effects

Input Ordering

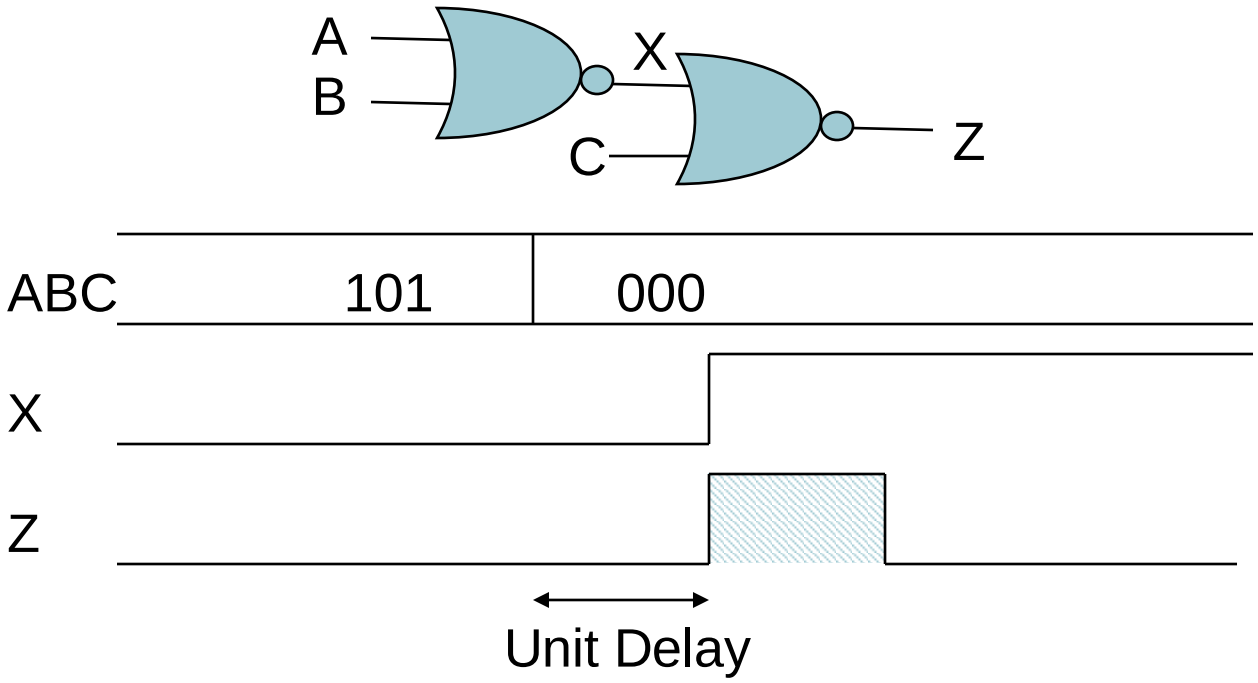




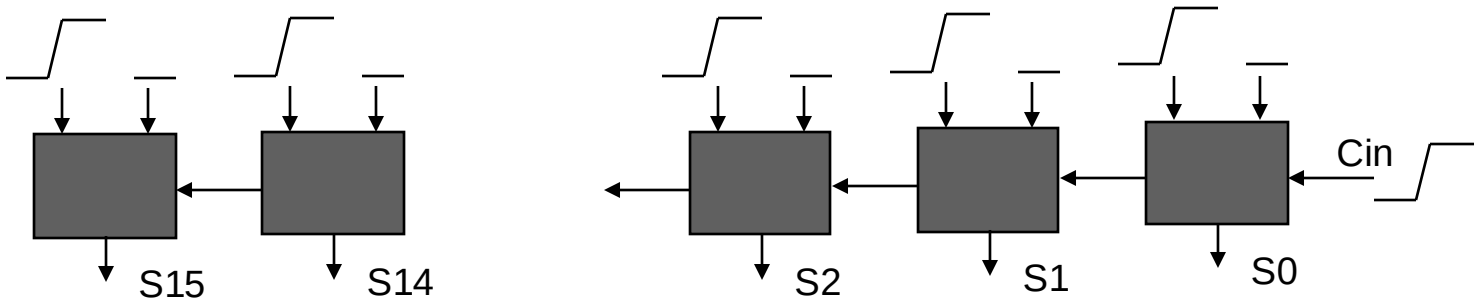
- Beneficial to postpone the introduction of signals with a high transition rate (signals with signal probability close to 0.5)

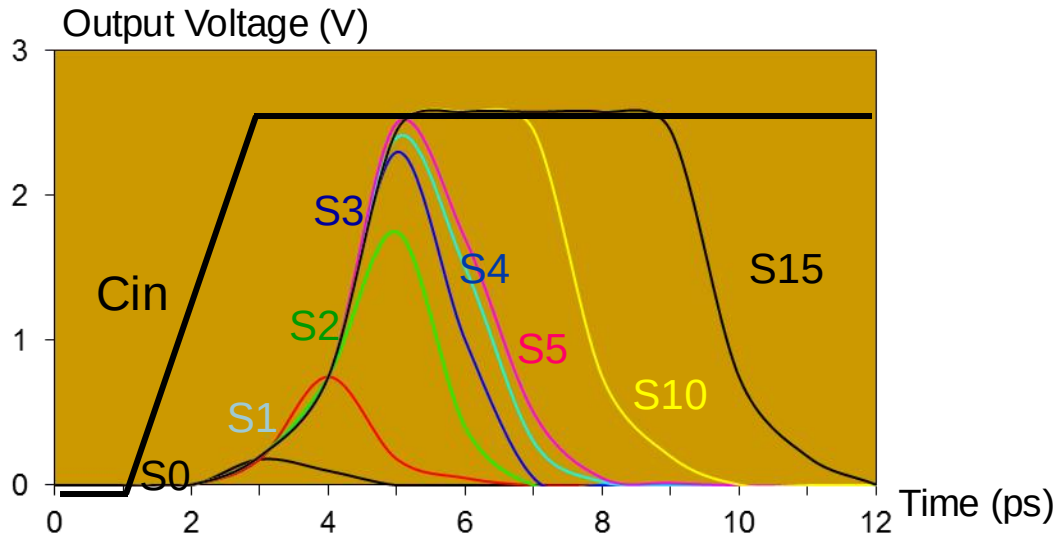
Glitching in Static CMOS Networks

- Glitch: node exhibits multiple transitions in a single cycle before settling to the correct logic value



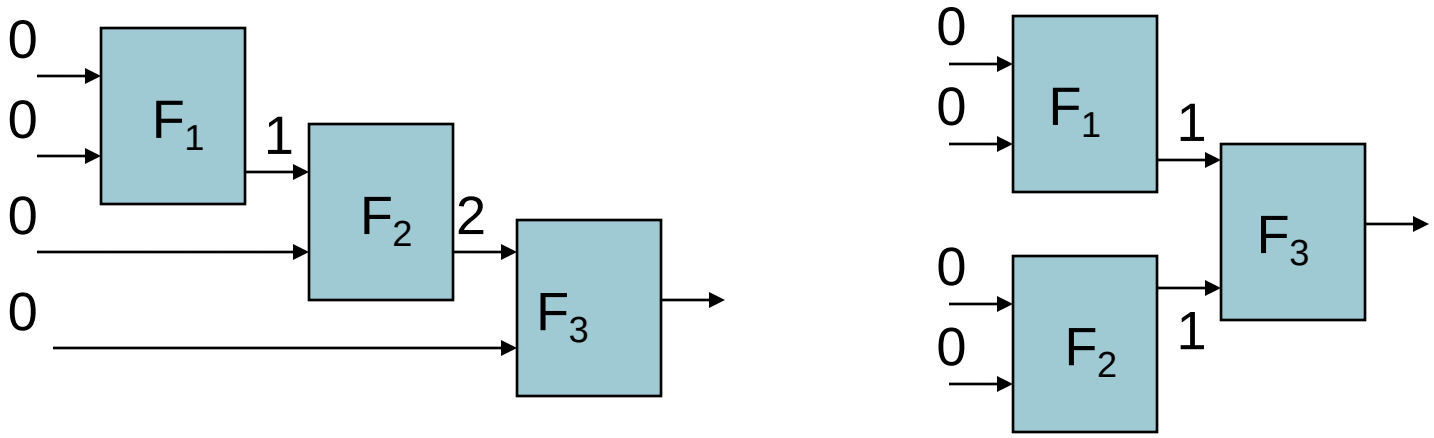
Glitching in an Carry Ripple Adder





Balanced Delay Paths to Reduce Glitching

- Glitching is due to a mismatch in the path lengths in the logic network; if all input signals of a gate change simultaneously, no glitching occurs



The lengths of timing paths are equalized through logic

Dynamic Power Reduction

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

VDD Reduction

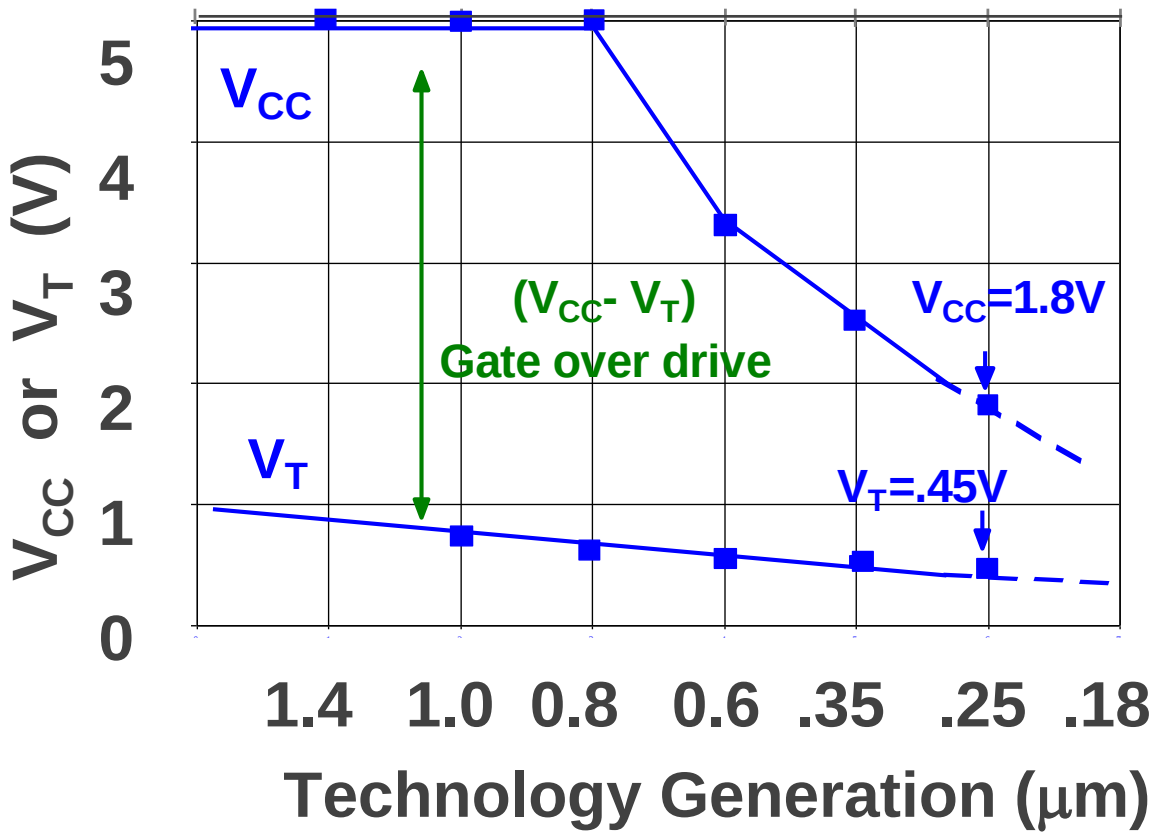
$$\text{Power} = CV_{DD}^2$$

$$\text{Delay} = \frac{KV_{DD}}{(V_{DD} - V_T)\alpha}$$

- Reducing V_{DD} supply voltage reduce the power consumption
- No effect on area
- Power consumption is a quadratic function of voltage
- Decrease in supply voltage increases the overall delay

Vdd vs. Vt Scaling

- Recently: constant e-field scaling, aka voltage scaling
- V_{CC} ~ 1V
- V_{CC} & modest V_T scaling
- Loss in gate overdrive (V_{CC}-V_T)



Voltage scaling is good for controlling IC's active power, but it requires aggressive V_T scaling for high performance

source: synopsys

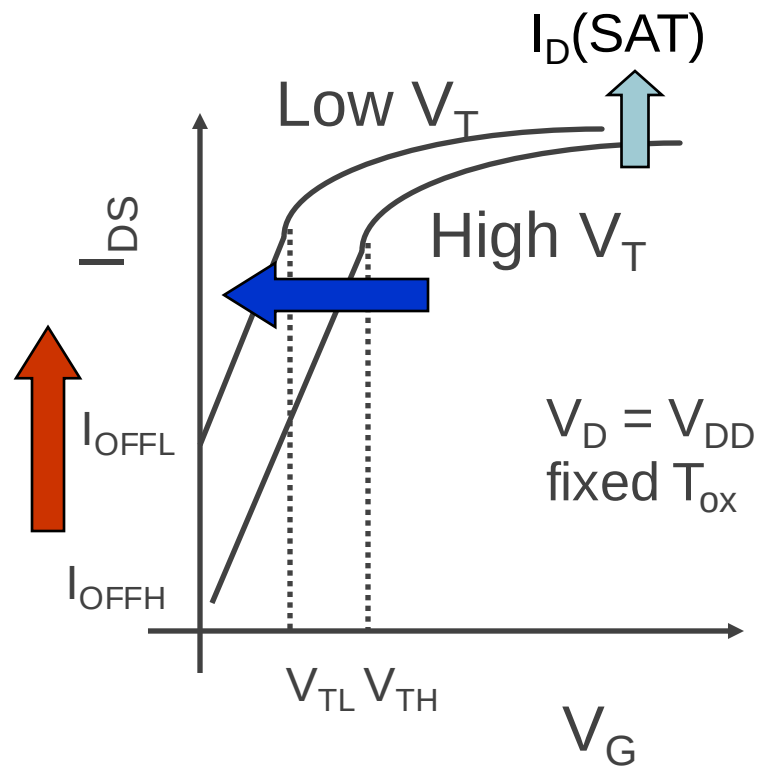
Barriers to Voltage Scaling

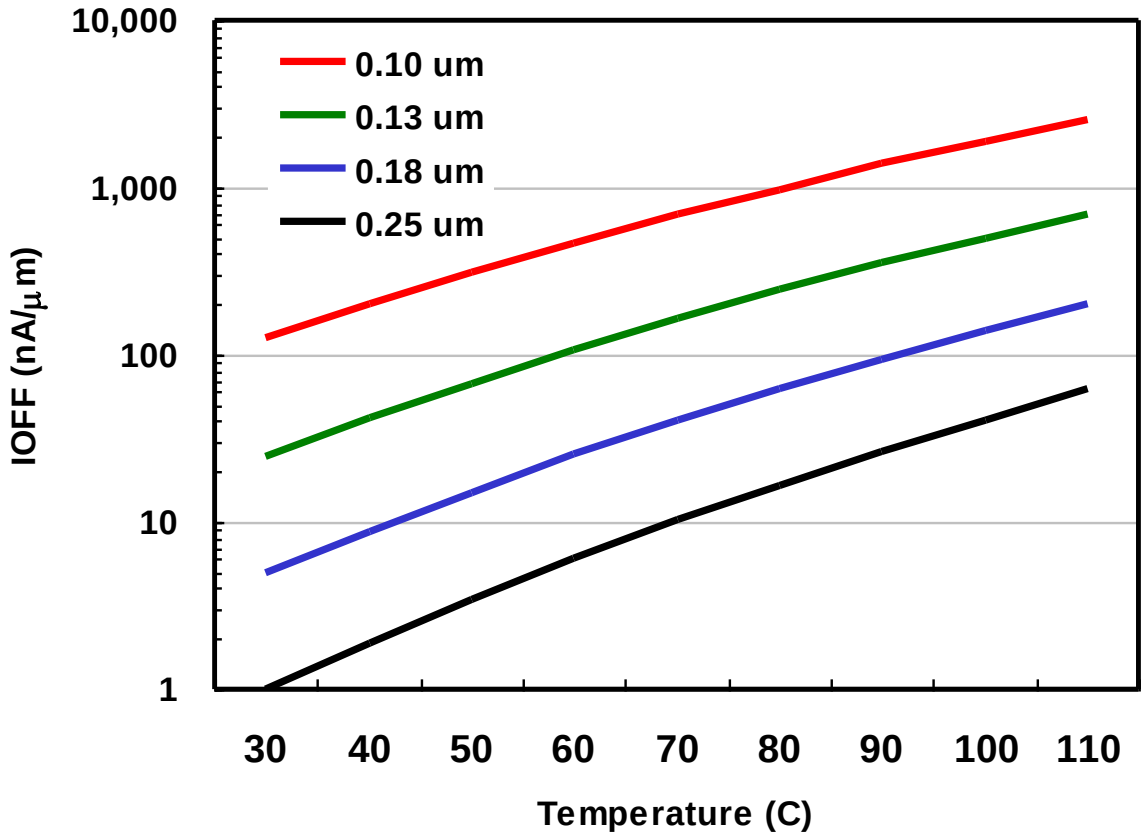
Voltage Scaling = Constant Electric Field Scaling

- Voltage scaling is good for IC's active power, but degrades gate over drive. Requires V_T scaling.
- Leakage power
- Special circuit functionality, noise
- Soft error
- Parameter variation

V_T Scaling: V_T and I_{OFF} Trade-off

As V_T decreases, leakage increases
Leakage is a barrier to voltage

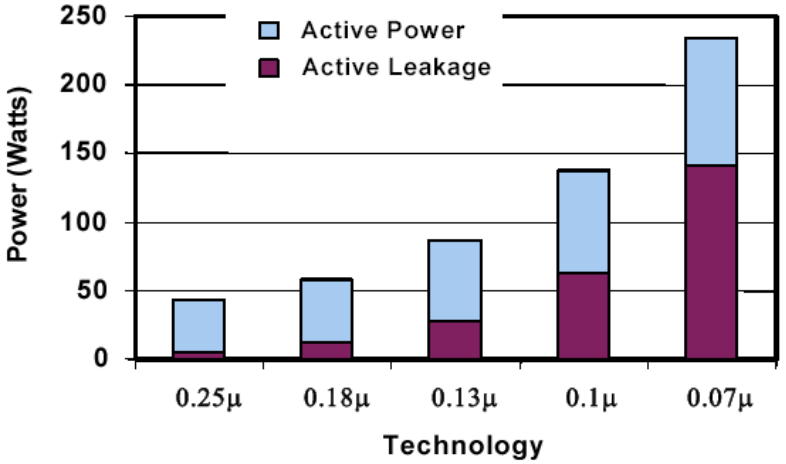


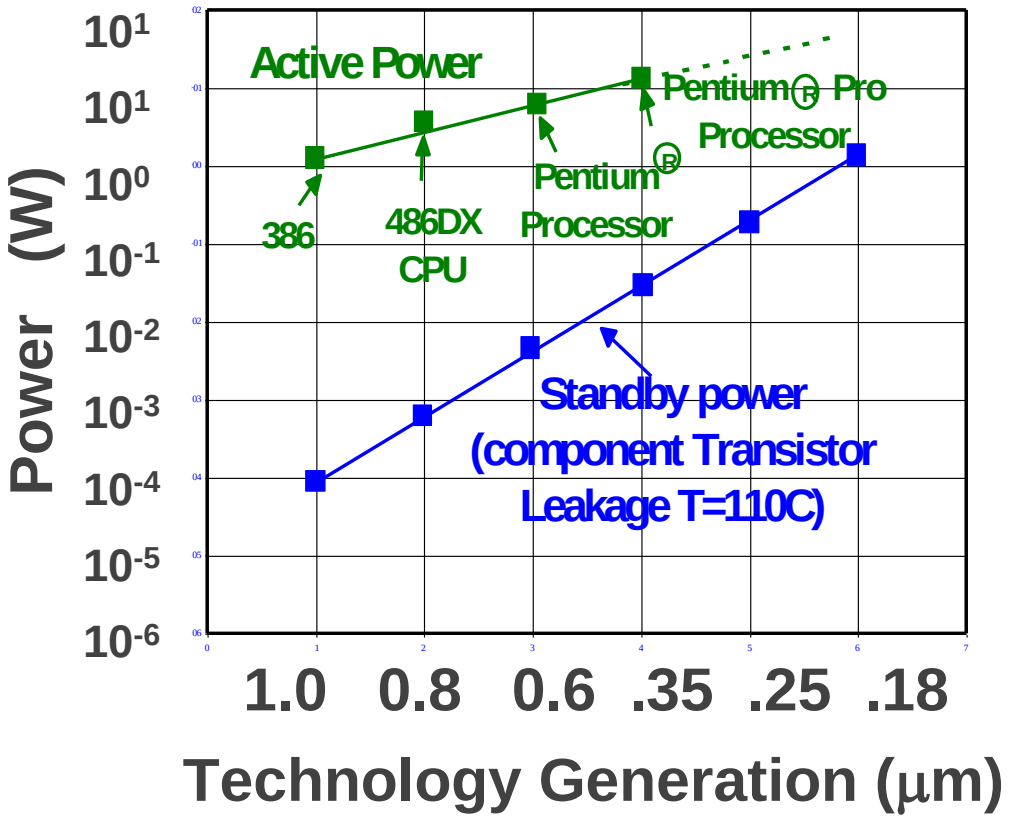


Projected Leakage Trends

source: synopsys

Why Excessive Leakage an Issue?

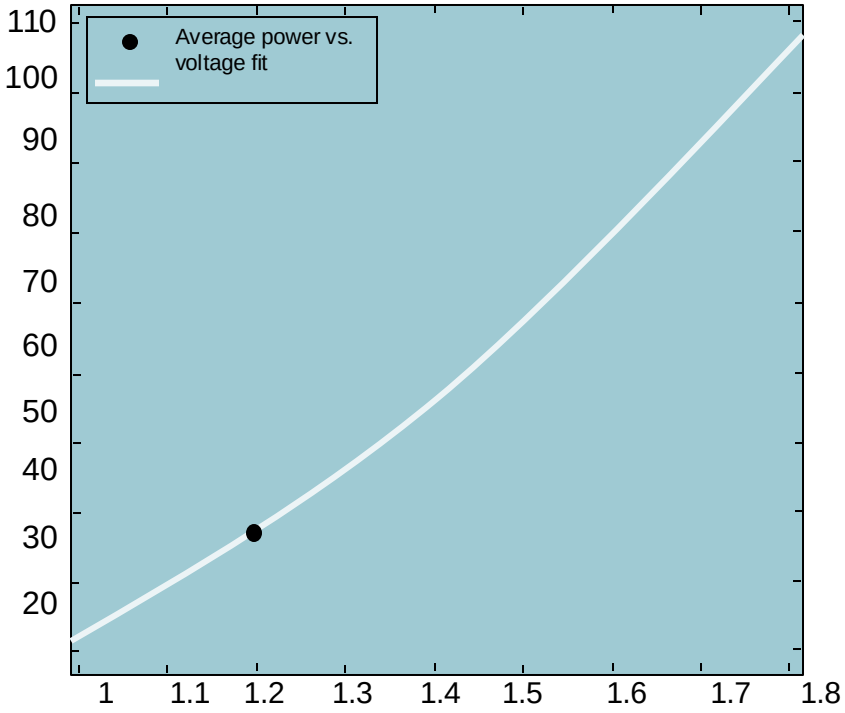




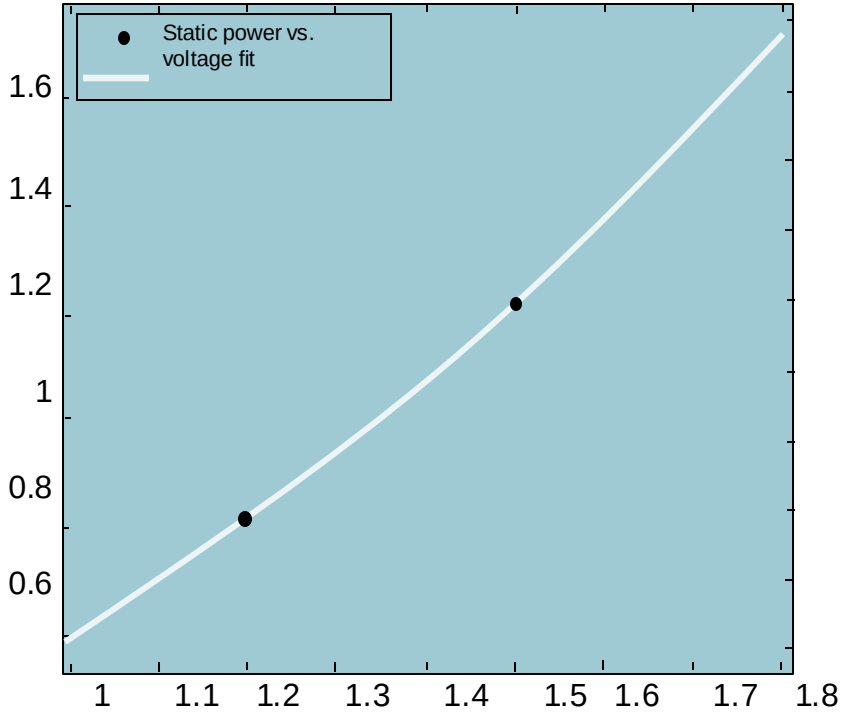
- Leakage component becomes significant % of total power
- Aggressive leakage control techniques are required!

source: synopsys

Power Versus Voltage

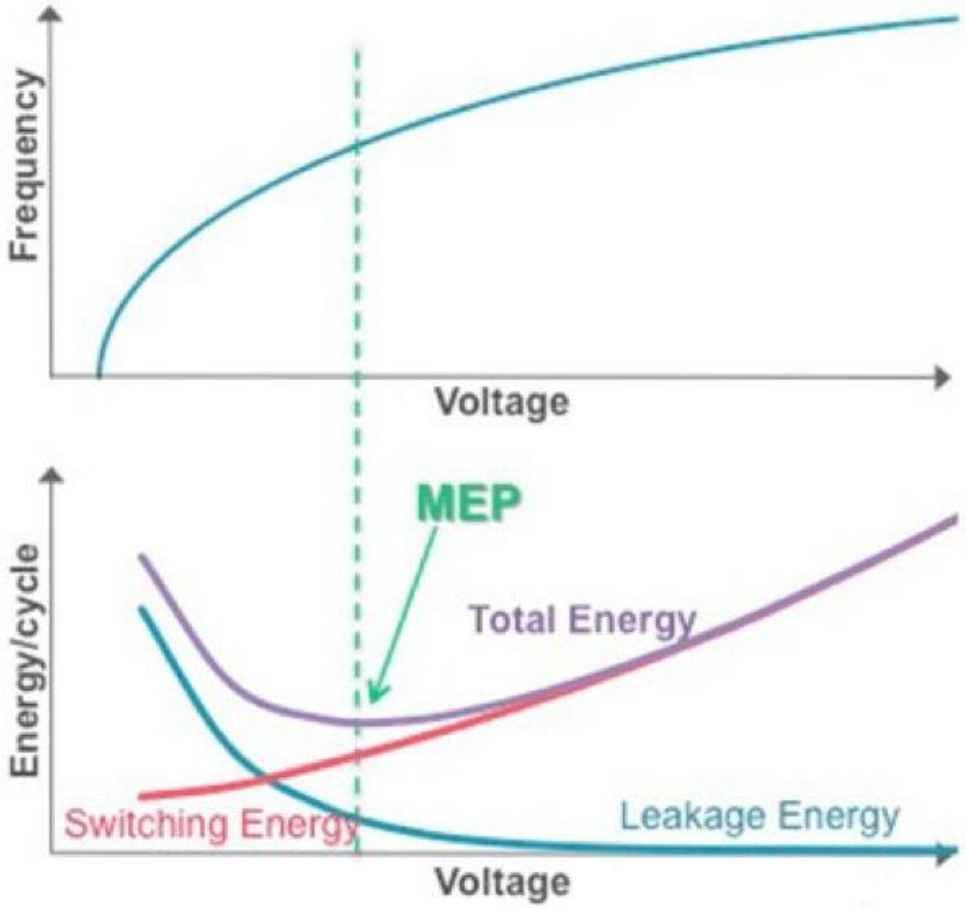


Dynamic Power vs. Voltage



Static Power vs. Voltage

Near-threshold Computing



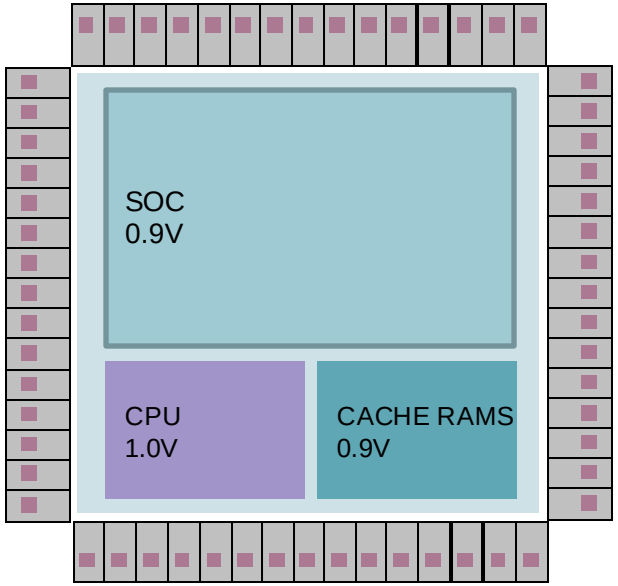
- As the voltage is dropped toward the transistor threshold voltage (Vt), switching energy decreases but at the same time leakage energy increases.
- The optimum operating point is usually slightly above Vt and is called the near-threshold operating point or minimum energy point.

source: https://semiengineering.com/near-threshold-computing-2/

Dynamic Power Reduction

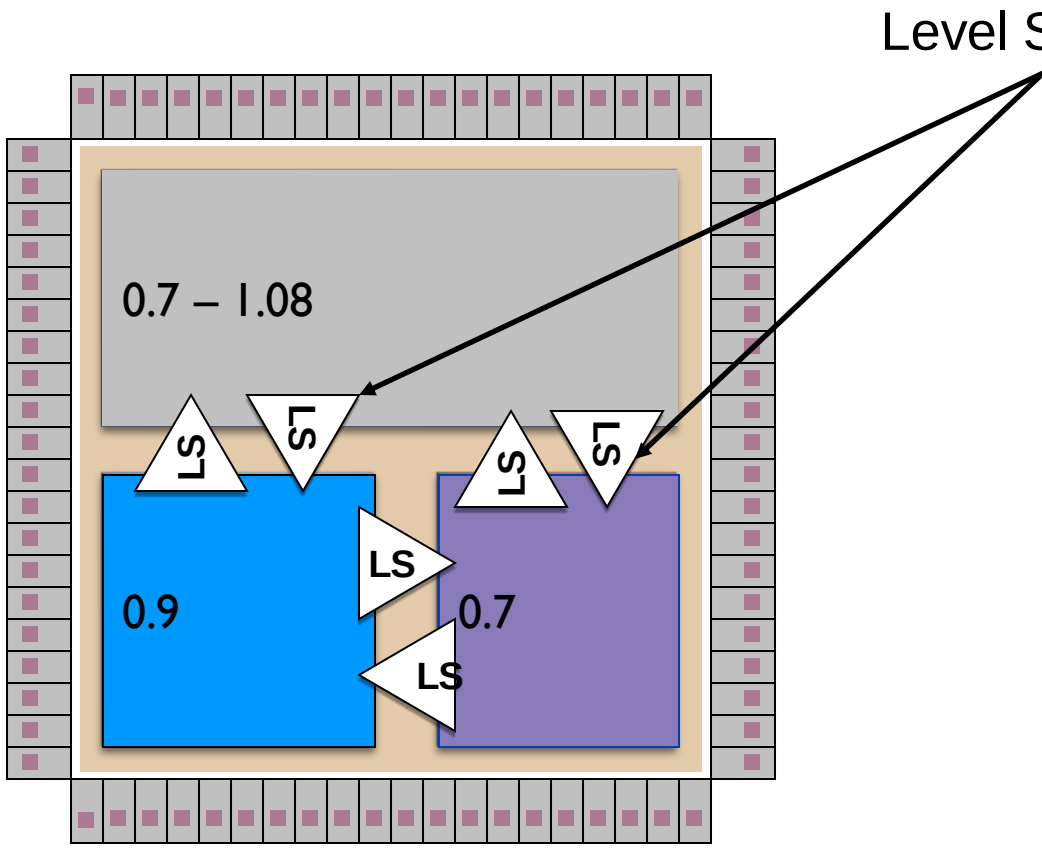
	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Sizing Reduced V_{dd} **Multi-V_{dd}**	Clock Gating	DFS, DVS (Dynamic Freq., Voltage Scaling)
Leakage	Stack effects Multi-V_T	Sleep Transistors Power Gating Variable V_T +Input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

Multi-Voltage Design Techniques

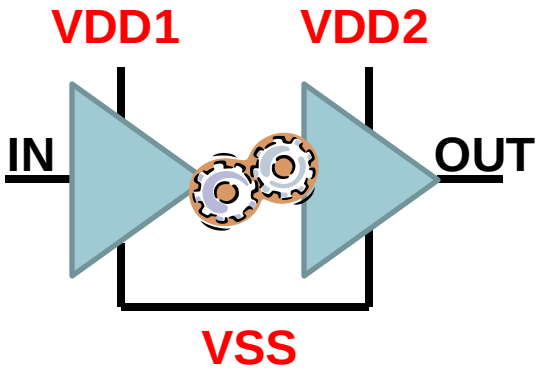


- Cache RAMS are run at the highest voltage because they are on the critical timing path.
- The CPU is run at a high voltage because its performance determines system performance. But it can be run at a slightly lower voltage than the cache and still have the overall CPU subsystem performance determined by the cache speed.
- The rest of the chip can run at a lower voltage still without impacting overall system performance.

Special Cell Requirements: Level Shifters



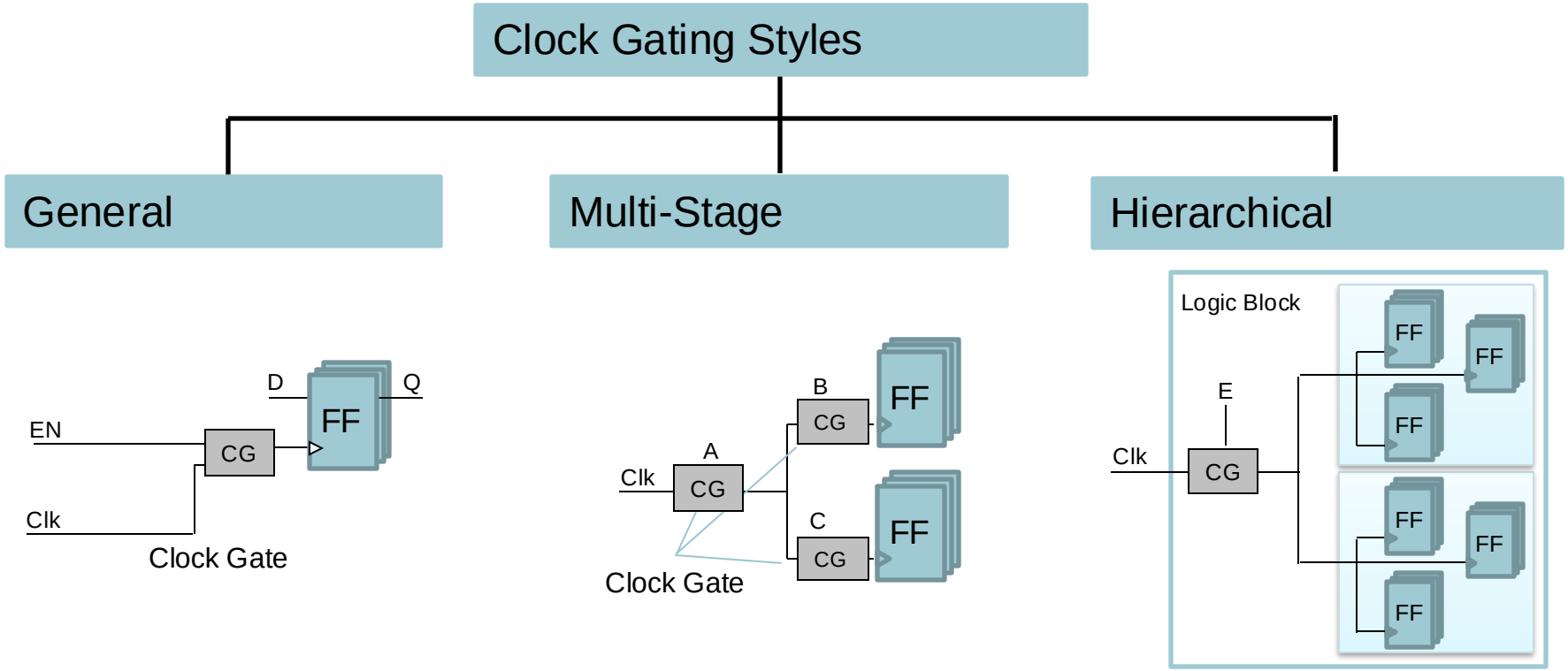
Level Shifter
logic model



Dynamic Power Reduction

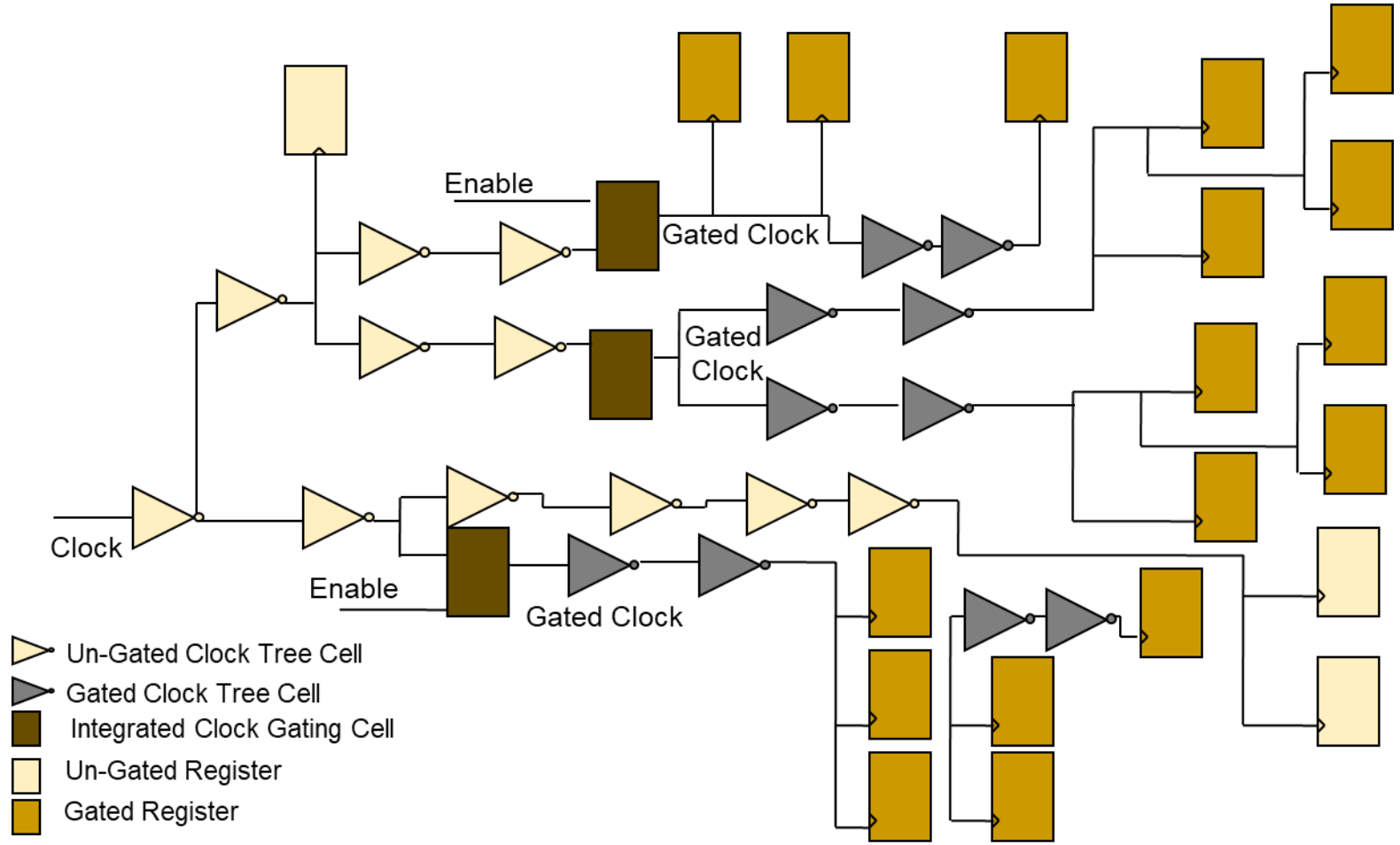
	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Sizing Reduced V_{dd} Multi-V_{dd}	**Clock Gating**	DFS, DVS (Dynamic Freq., Voltage Scaling)
Leakage	Stack effects Multi-V_T	Sleep Transistors Power Gating Variable V_T +Input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

Clock Gating Styles



A significant fraction of the dynamic power in a chip is in the distribution network of the clock.

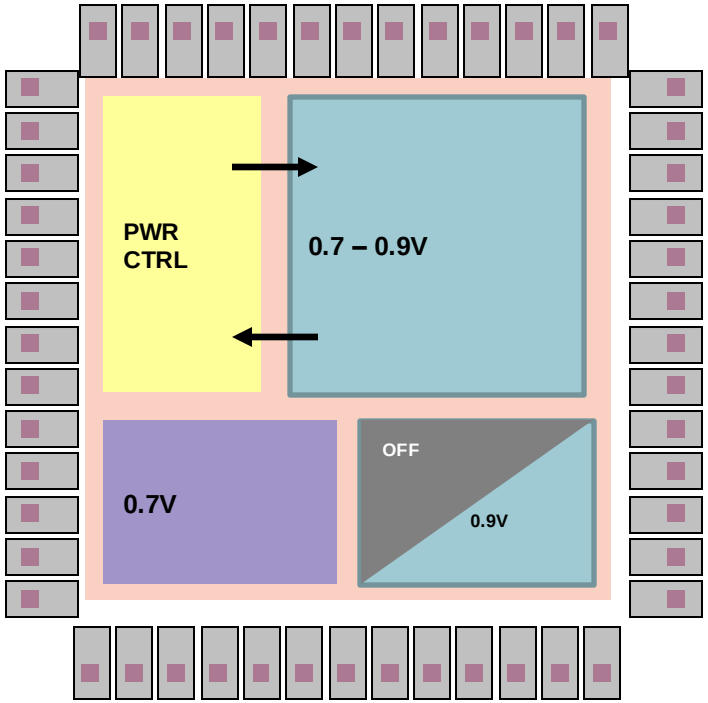
Clock Gating Example



Dynamic Power Reduction

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

Dynamic Voltage/Frequency Scaling (DVFS)



Dynamic Voltage/Frequency Scaling (DVFS)

- Dynamic voltage and frequency scaling
  - Adjusts performance and energy consumption levels while the device is active
  - Key is to meet users performance needs while saving energy
  - Reduces processor frequency and voltage to obtain quadratic energy savings

DVFS

- Workload of a task , $W_{task,}$ is defined as the total number of CPU clock cycles required to finish the task

$$W_{task} \overset{\Delta}{=} \sum_{i=1}^{n} CPI_i$$

n : total number of instructions in a task
CPI : clock cycles per instructions

- The task execution time, T_{task}, is a function of CPU frequency, f_{CPU}

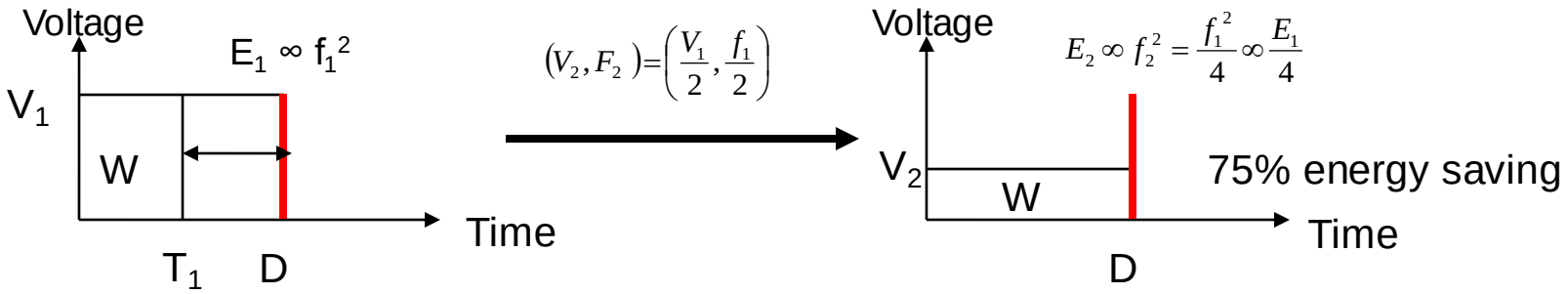
$$T_{task} = \frac{W_{task}}{f_{CPU}}$$

- To save CPU energy using DVFS for a given deadline, D, choose a $f_{CPU,}$ at which T_{task} can be closest to D

$$T_{task} = D, \qquad f_{CPU} = \frac{W_{task}}{D}$$

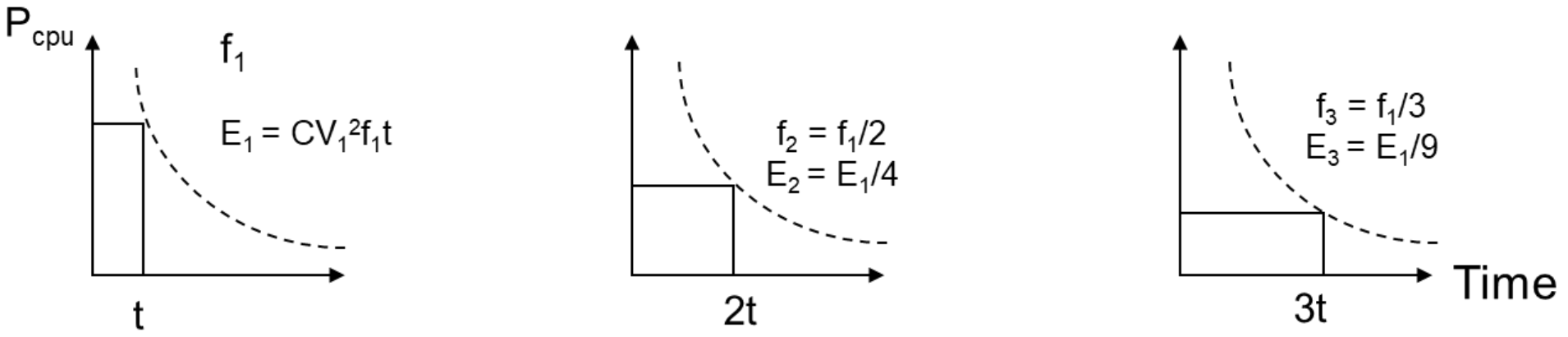
DVFS by Example

- Suppose that a task with workload W should be finished by deadline D

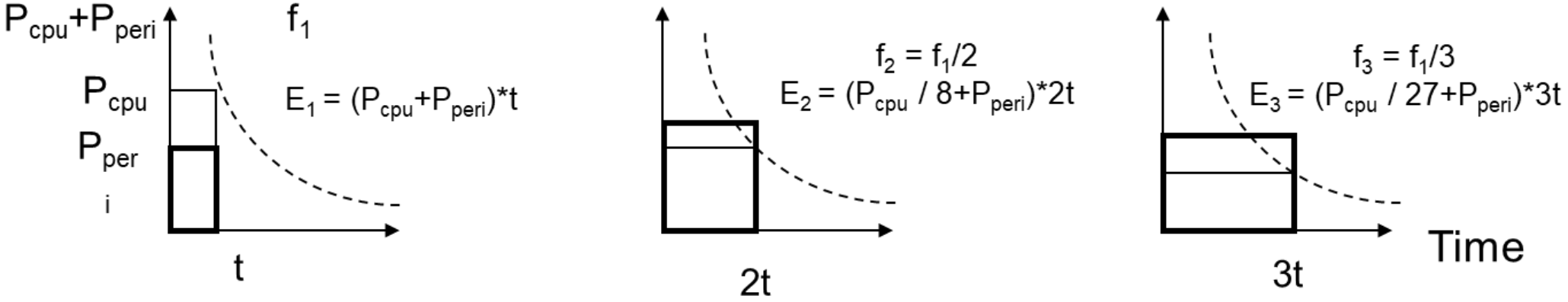


- DVFS is an affective way of reducing the CPU energy consumption by providing computation power

Total System Energy Variation in DVFS



For CPU, lower frequency gives lower energy consumption

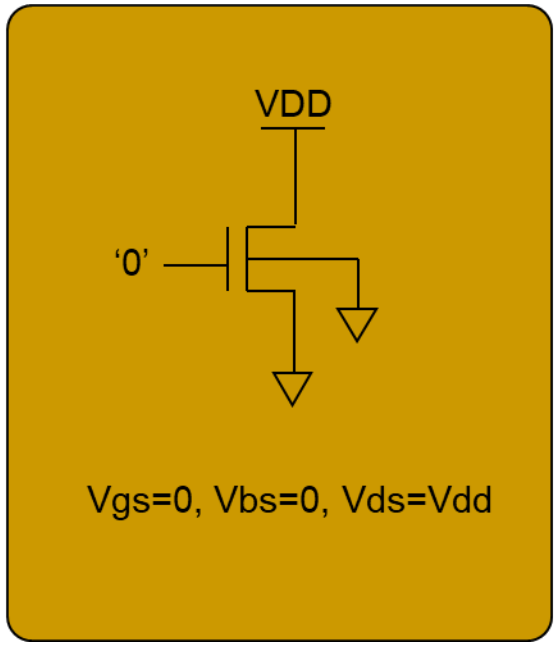


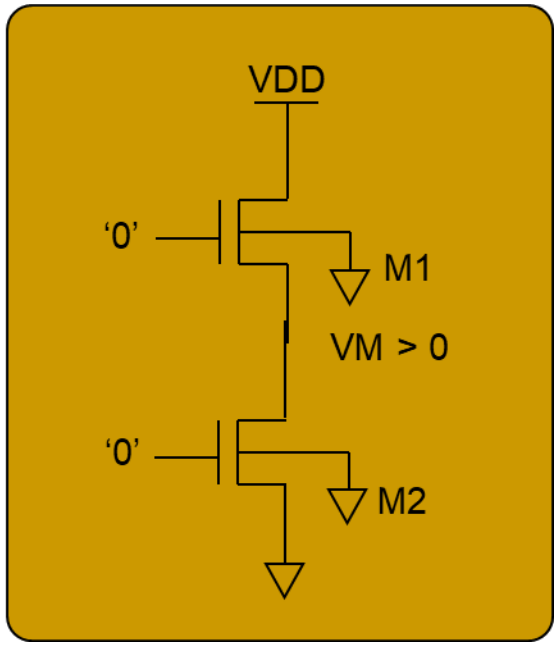
For a system, lower frequency does not give lower energy consumption

Static Power Reduction

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

Leakage Control Stacking

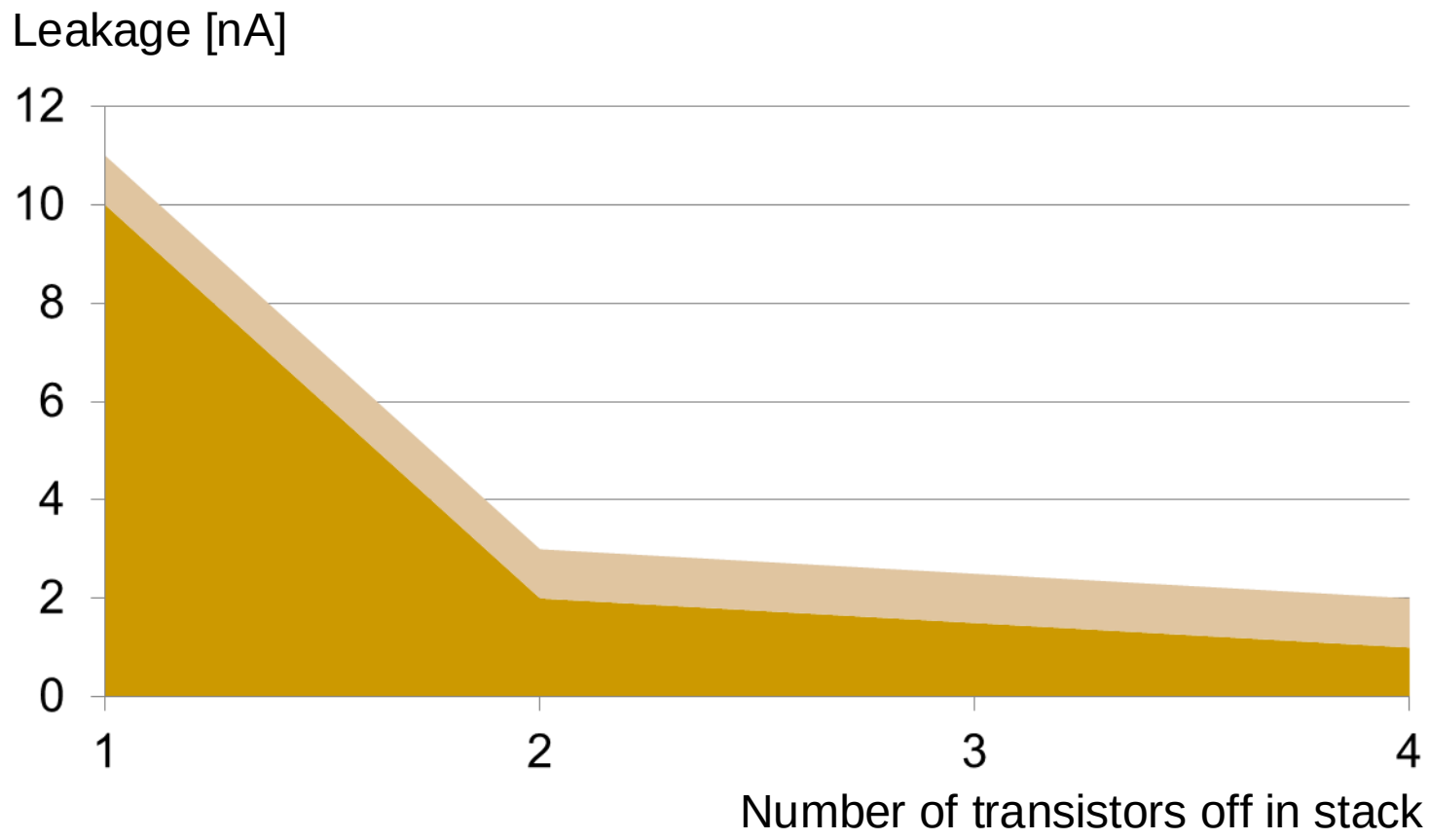




Negative V_{gs},
Negative V_{bs}-more
Body effect,
Reduced V_{ds}-less DIBL
2-T stack has lower
Subthreshold leakage

For M1
$V_{gs} = - V_M < 0$, $V_{bs} = -V_M < 0$,
$V_{ds} = Vdd – V_m < Vdd$
For M2
$V_{gs} = 0$, $V_{bs} = 0$
$V_{ds} = V_M < Vdd$

Leakage Versus Transistors

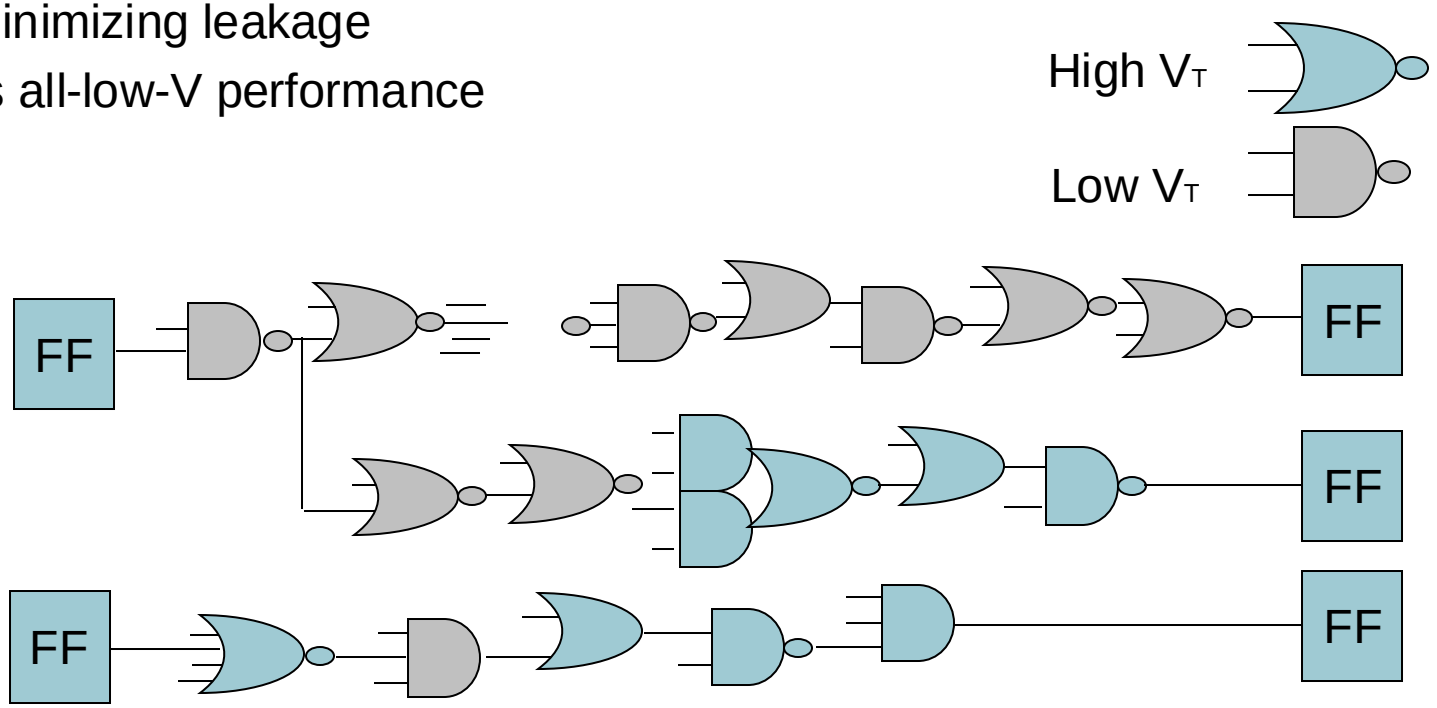


Static Power Reduction

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

Using Multiple Thresholds

- Cell-by-cell VT assignment (not block level)
- Allows minimizing leakage
- Achieves all-low-V performance



- No need for level shifting
- Only two thresholds are needed per block
  - Using more than two yields small improvements

Static Power Reduction

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

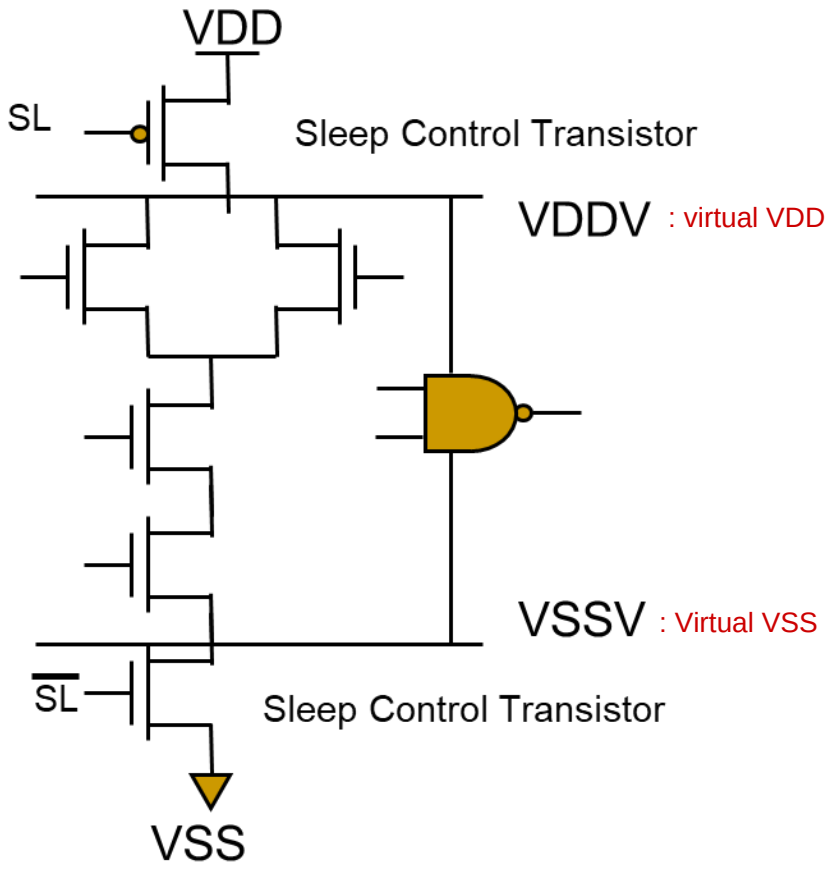
Leakage is happened when the module is IDLE or Non-active

so use sleep transistor to turn off that non using transistors

Multi-Threshold CMOS (MTCMOS)

- In active mode: SL=0, MP and MN are “on”
  - VDDV and VSSV almost function as VDD and VSS.
- In standby mode: SL=1, MP and MN are “off” and leakage is suppressed.

per gate

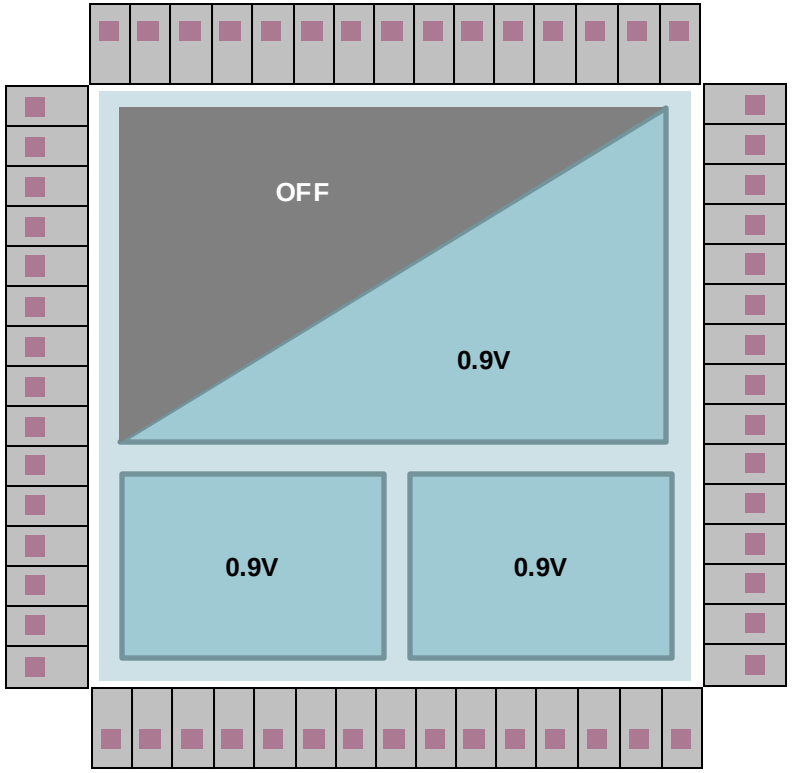


Sleep transistors should not produce area overhead

Static Power Reduction

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

Power Gating Bases

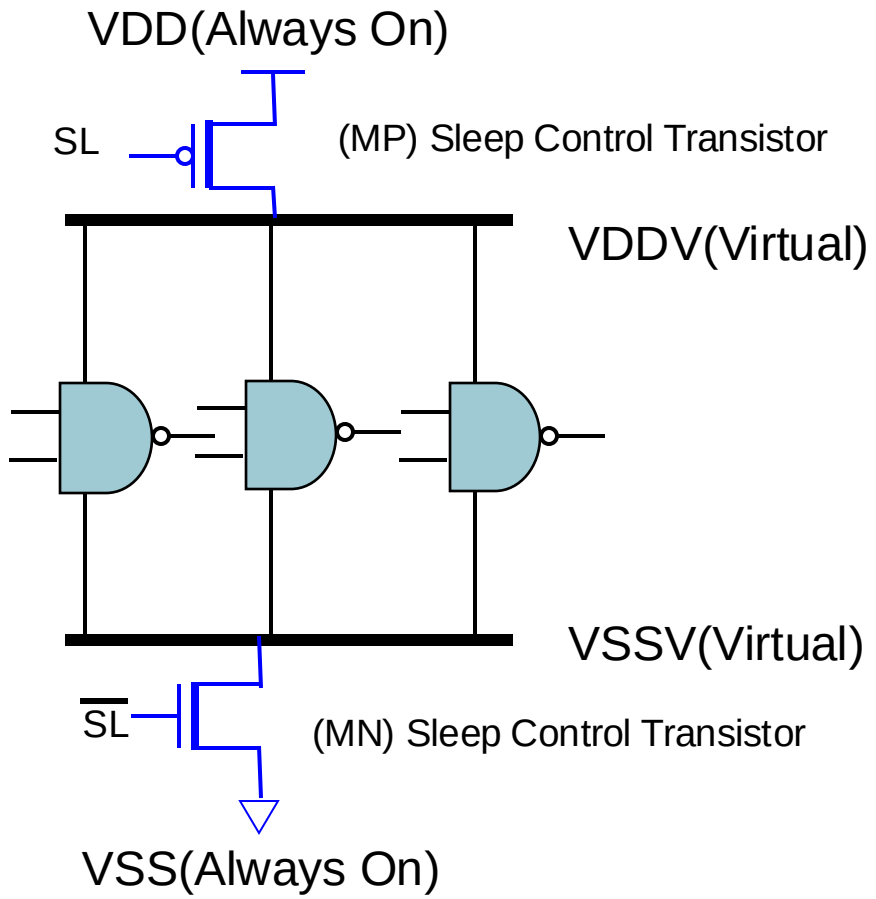


Power gating

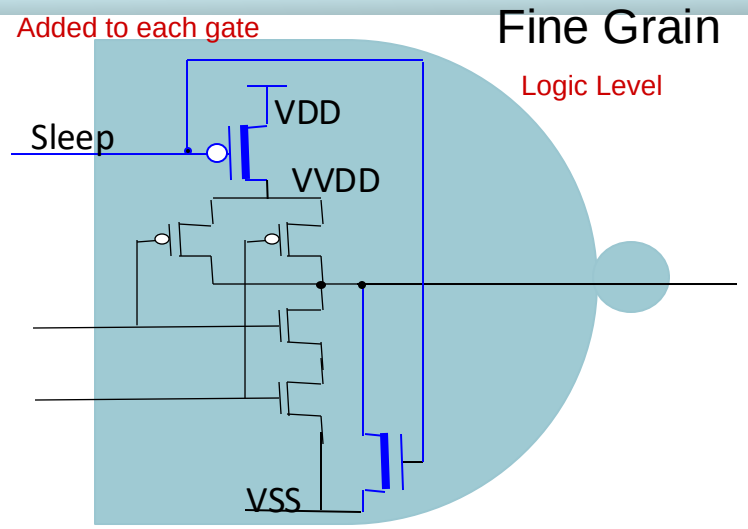
Power Gating Details

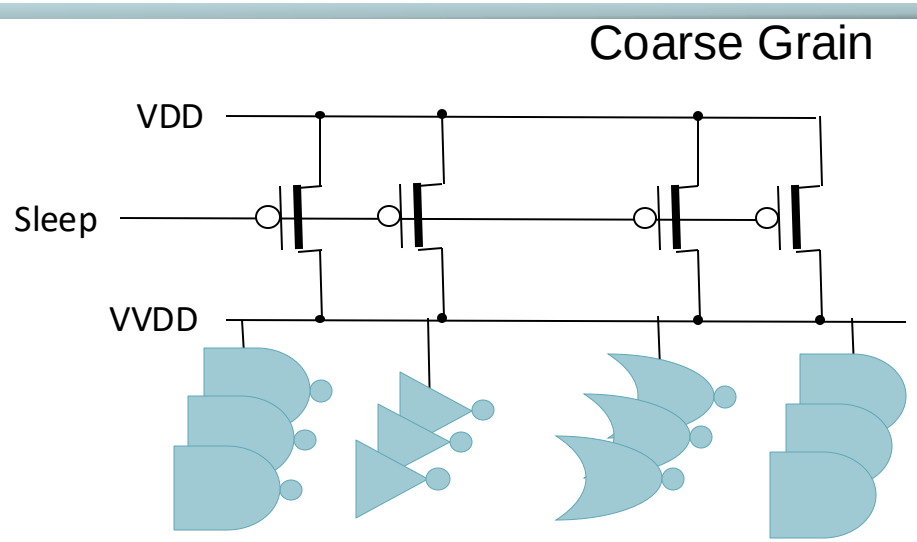
- In active mode: SL=0, MP and MN are “on”
  - VDDV and VSSV almost function as VDD and VSS.
- In standby mode: SL=1, MP and MN are “off” and leakage is suppressed.

power gating controls more gates at once



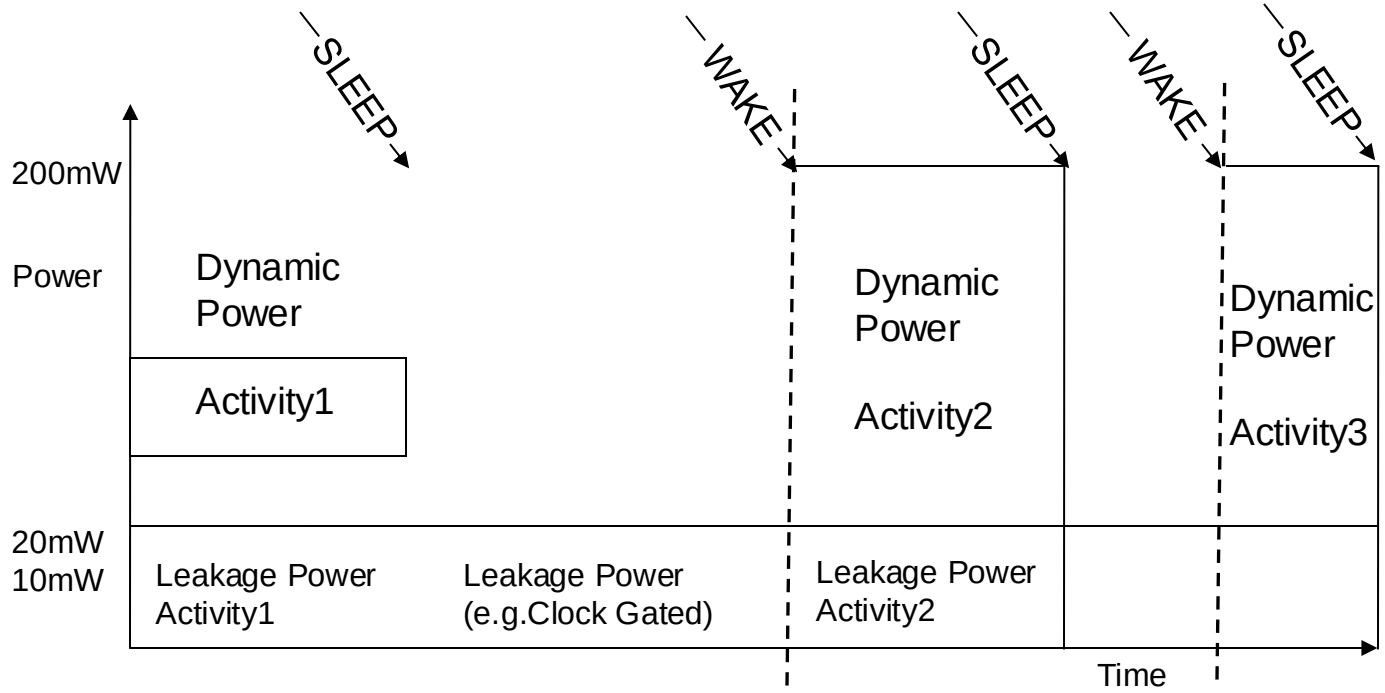
Power Switching – Fine Grain vs. Coarse Grain





Parameter	Fine Grain	Coarse Grain
Reduce leakage	10x	50x
Wakeup time (transition time from idle to active)	Fast	Slow
Wakeup power	Small	Large
Library requirements	New cell library	New footer or header cell
Sensitive to PVT variation	Large	Small
Can be implemented by usual physical synthesis	Yes	No (Cannot be synthesize, need customization)
IR-drop variations	Large	Small
Chip Area	Large	Small

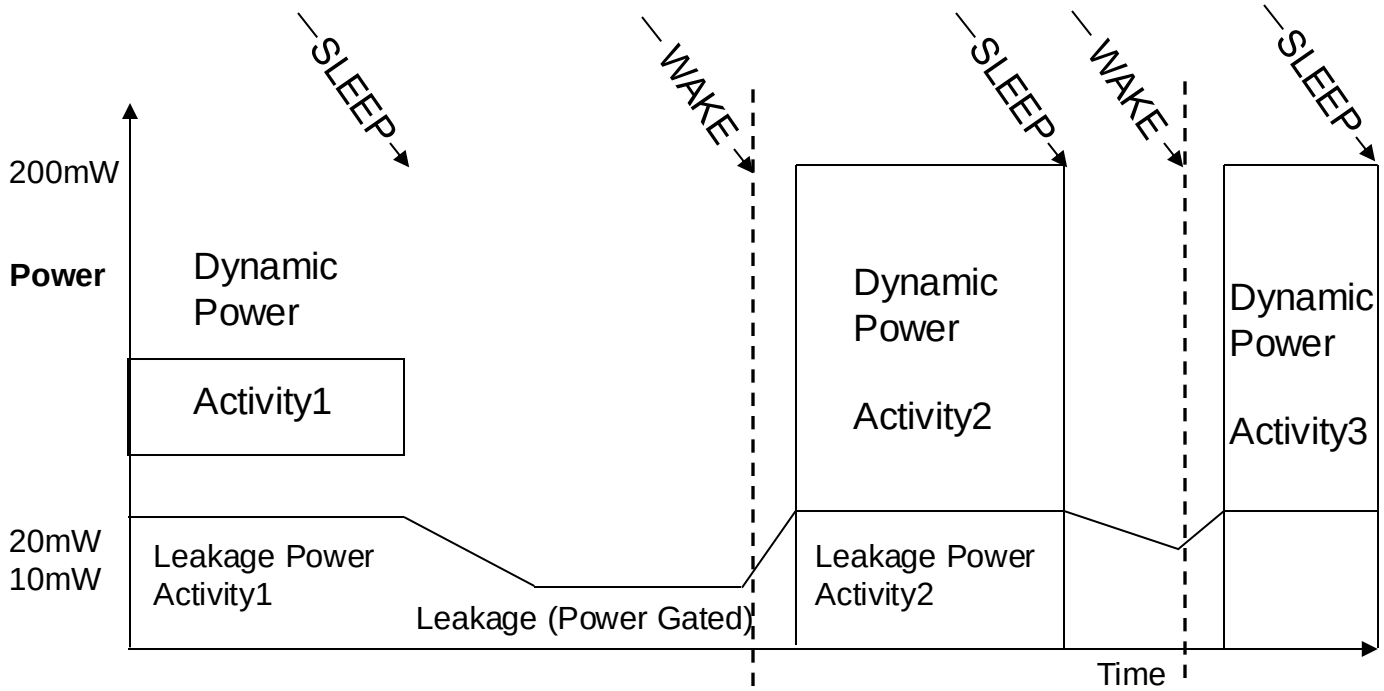
Dynamic and Leakage Power Profiles



An example activity profile for a sub-system using clock gating to reduce power.

- SLEEP events initiate entry to the low power mode
- WAKE events initiate return to active mode

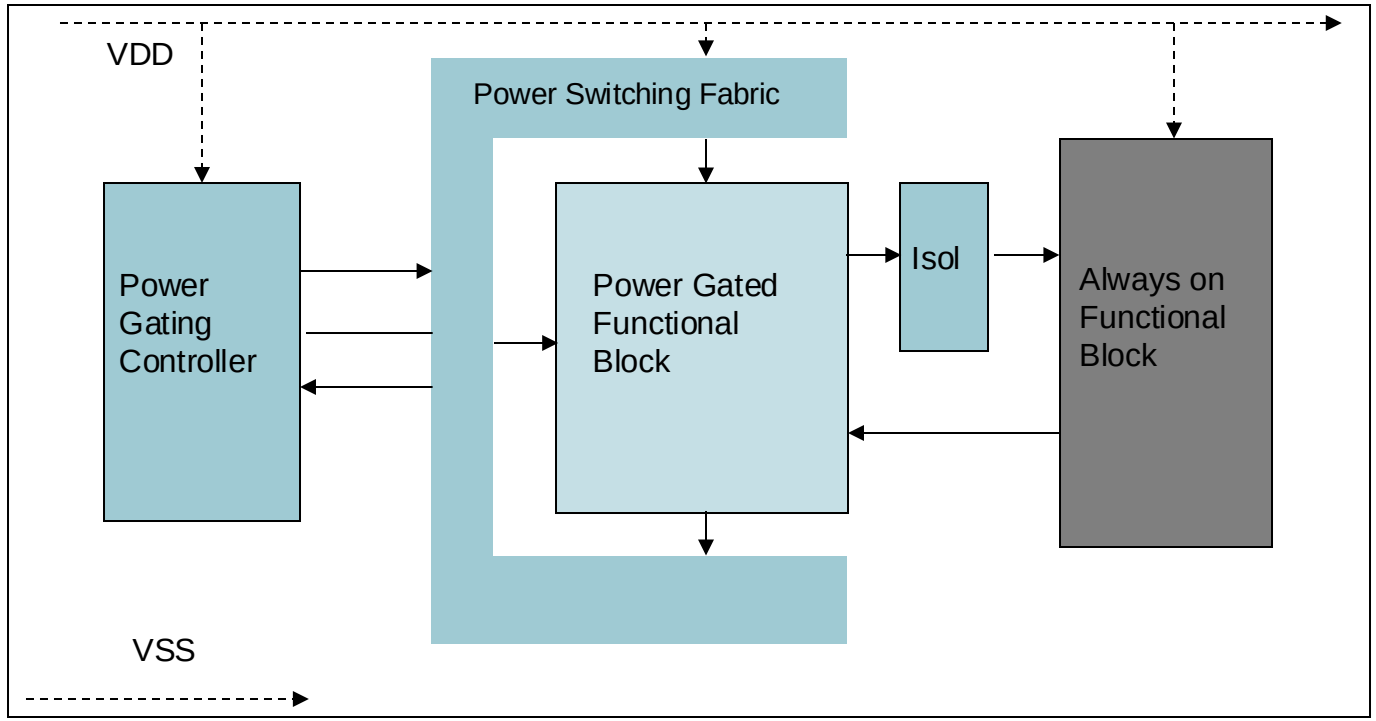
Realistic Profile with Power Gating



- Figure shows Realistic Profile with Power Gating. The leakage power savings are not perfect and instantaneous; the full leakage power savings take some time to reach target levels. This is due partly to the hotter thermal profile of the preceding activity and partly to the non-ideal nature of the power-gating technology. Therefore the achievable savings are compromised to some extent.

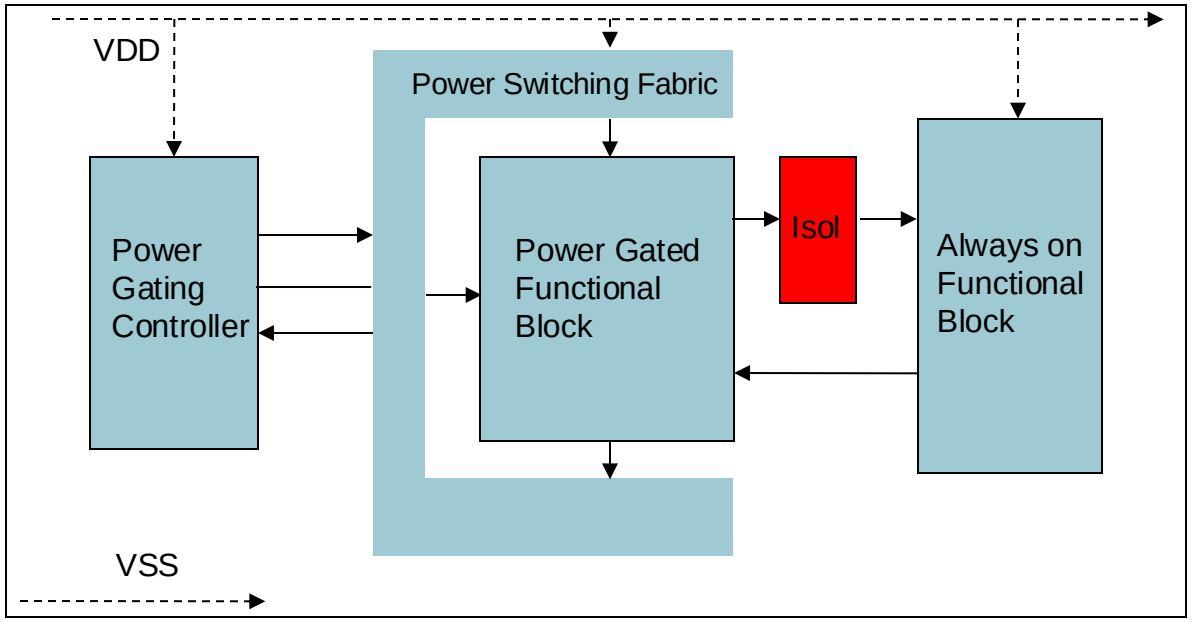
Block Diagram of an SoC with Power Gating (1)

- A simplified view of an SoC that uses internal power gating is shown. Unlike a block that is always powered on, the power-gated block receives its power through a power-switching network. This network switches either V_{DD} or V_{SS} to the power gated block. In this example, V_{DD} is switched; V_{SS} is provided directly to the entire chip. The switching fabric typically consists of a large number of CMOS switches distributed around or within the power gated block.



Block Diagram of an SoC with Power Gating (2)

- One challenge for power gating designs is that the outputs of the power gated block **may ramp off very slowly**. The result could be that these outputs spend a significant amount of time at threshold voltage, causing large crowbar currents in the always powered on block. To prevent these crowbar currents, isolation cells (the “Isol” block in the figure) are placed between the outputs of the power gated block and the inputs of the always on block. These isolation cells are designed so that they do not experience crowbar current when one of the inputs is at threshold, as long as the control input is off. The power gating controller provides this isolation control signal.

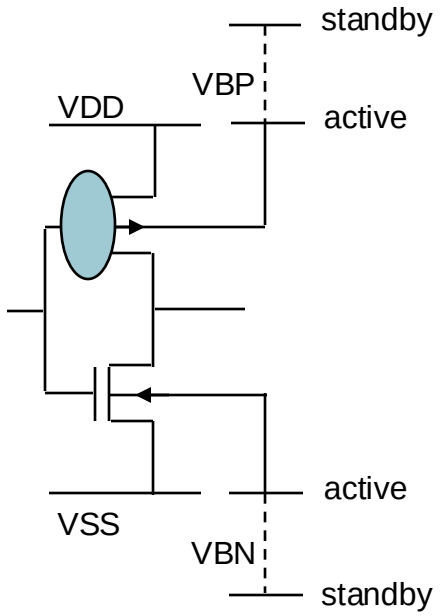


Static Power Reduction

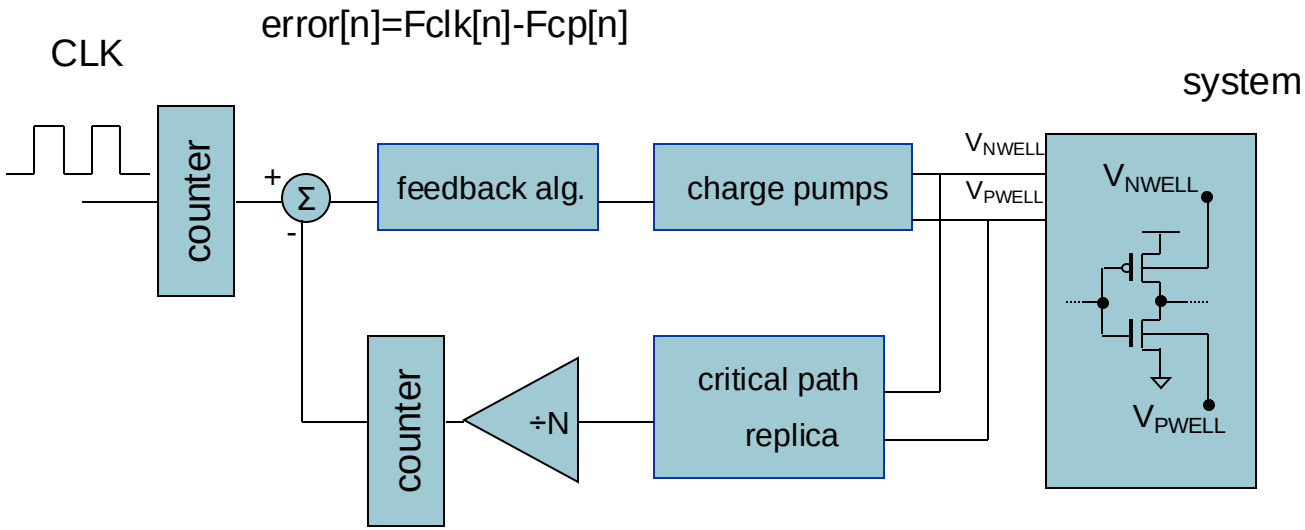
	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Sizing Reduced V_{dd} Multi-V_{dd}	Clock Gating	DFS, DVS (Dynamic Freq., Voltage Scaling)
Leakage	Stack effects Multi-V_T	Sleep Transistors Power Gating **Variable V_T** +Input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

Variable Threshold CMOS (VTCMOS)

- Variable Threshold CMOS (from T. Kuroda, ISSCC, 1996)
- A body biasing design technique
  - In active mode:
    - Zero or slightly forward body bias for high speed
  - In standby mode:
    - Deep reverse body bias for low leakage
  - Triple well technology required
- VTCMOS requires modification to cell libraries to separate back-gate bias lines from V_{dd} and V_{ss} lines



Dynamic Vth Scaling (DVTS)



- DVTS scheme uses body biasing to adaptively change the V_{th} based on the performance demand
- The lowest V_{th} is delivered via ZBB, if the highest performance is required
- When performance demand is low, clock frequency is lowered and V_{th} is raised via RBB to reduce the run-time leakage power dissipation

Static Power Reduction

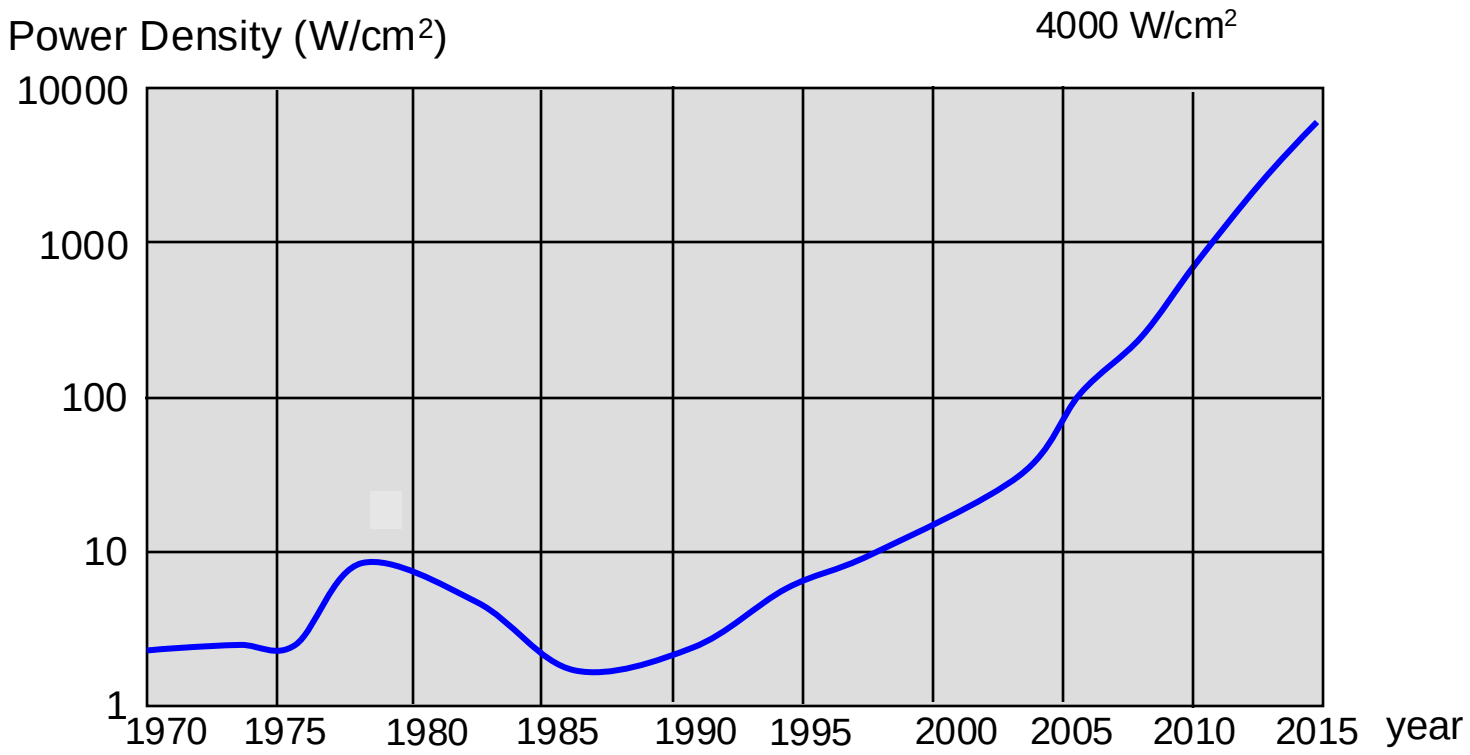
	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Sizing Reduced V_{dd} Multi-V_{dd}	Clock Gating	DFS, DVS (Dynamic Freq., Voltage Scaling)
Leakage	Stack effects Multi-V_T	Sleep Transistors Power Gating Variable V_T **+Input Control**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

Forcing Input Vectors into Lowest Leakage State

- Individual cells are forced into lowest leakage state during stand-by.
- For example, for a CMOS inverter the lowest leakage state corresponds to INPUT=1
  - The advantages of this approach:
    - No process change is required
    - No special cells are required
  - The disadvantages of this approach:
    - Dynamic power consumption by additional control cells
    - Area overhead of additional logic
    - Extent of leakage reduction is unproved.
- This approach could be more effective with usage of dual threshold voltage (VT) and dual oxide thickness (Tox) technologies. The high-VT reduces sub-threshold leakage, the thick oxide reduces gate leakage.

Summary: Power is important!

- Power Density increases twice every year!
- Energy and power are two different concepts!
- Low power can be enabled by multiple design techniques!



RELIABILITY

Figure: https://www.pioneer-engineering.com/resources/availability-vs-reliability-part-1

Reliability

- Reliability is a measure of functionality over time. If a chip completely stops working, it is deemed unreliable. But in many cases performance degrades over a period of years, whether that's due to electromigration, layer upon layer of software patches, memory bit failures, or a host of other issues that can crop up. Reliability in these cases may be more subjective than definitive.
- Reliability is also known as Dependability and Fault-Tolerance

source: https://semiengineering.com/are-chips-getting-more-reliable/

Reliability

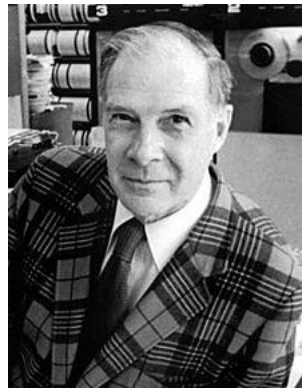
- Reliability is related to die area, clock frequency, and power.
- Die area increases the amount of circuitry and the probability of a fault.
- It also allows the use of error correction and detection techniques.
- Higher clock frequencies increase electrical noise and noise sensitivity.
- Faster circuits are smaller and more susceptible to radiation.

Fault-Tolerance: Definition/Design

- *Failure* is a deviation from a design specification.
- *Fault* is an error that manifests itself as an incorrect result.
  - *Physical fault* is a failure caused by environment: aging, radiation, temperature, etc. The probability of physical faults increases with time.
  - *Design fault* is a failure caused by a bad design. Design faults occur early in the lifetime of a design.
- Faults do not necessarily produce incorrect program execution.
- Can be masked by detection/correction logic, e.g. hamming code or ecc codes

The Hamming SEC Code

- SEC: single error correction
- Turing Award in 1968
- “if a machine can find out that there is an error, why can’t it locate where it is and change the setting of the relay from one to zero or zero to one?” – Richard Hamming (Bell Lab) in 1947



NOTE: This materials was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

Source: https://history.computer.org/pioneers/hamming.html

The Hamming SEC Code

- Hamming distance
  - Number of bits that are different between two bit patterns
  - Example: 011011 and 001111 -> 2
- Minimum distance = 2, one bit error provides single bit error detection
  - E.g. parity code - the number of 1 s in a word is counted;
  - the word has odd parity if the number of 1 s is odd and even otherwise.
  - That is, **the parity of the N+1 bit word should always be even (or odd, by predetermined convention).**

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

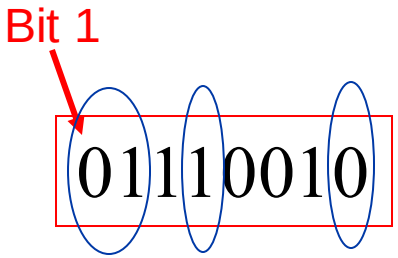
Example (1-bit parity scheme)

- Calculate the parity of a byte with the value 31_{ten} and show the pattern stored to memory. Assume the parity bit is on the right. Suppose the most significant bit was inverted in memory, and then you read it back. Did you detect the error? What happens if the two most significant bits are inverted?
  - $31_{ten} = 00011111_{two}$
  - $00011111\mathbf{1}_{two}$ Parity bit, N+1
  - Inverted $\mathbf{1}0011111\mathbf{1}_{two}$ but we expect **even** parity, thus we get an error.
  - What if the first two bits are inverted?
    - **11**0111111 -> No errors…

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

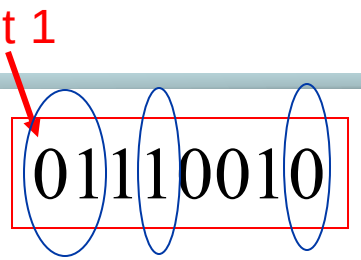
Encoding SEC (ECC)

- ECC: Hamming Error Correction Code
- Use extra bits (e.g. 4 bits)
- To calculate Hamming code:
  - Number bits from 1 on the left
  - All bit positions that are a power 2 are parity bits
  - Each parity bit checks certain data bits



NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

Encoding SEC



Bit 1 (0001_{two}) checks bits (1,3,5,7,9,11,...), which are bits where rightmost bit of address is 1 (0001_{two}, 0011_{two}, 0101_{two}, 0111_{two}, 1001_{two}, 1011_{two},...).

Bit 2 (0010_{two}) checks bits (2,3,6,7,10,11,14,15,...), which are the bits where the second bit to the right in the address is 1.

Bit 4 (0100_{two}) checks bits (4–7, 12–15, 20–23,...), which are the bits where the third bit to the right in the address is 1.

Bit 8 (1000_{two}) checks bits (8–15, 24–31, 40–47,...), which are the bits where the fourth bit to the right in the address is 1.

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

Encoding SEC

each data bit is covered by two or more parity bits

Bit position		1	2	3	4	5	6	7	8	9	10	11	12
Encoded date bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
Parity bit coverate	p1	X		X		X		X		X		X	
	p2		X	X			X	X			X	X	
	p4				X	X	X	X					X
	p8								X	X	X	X	X

What can be done with this?

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

Decoding SEC

- Value of parity bits indicates which bits are in error
  - Use numbering from encoding procedure
  - E.g.
    - Parity bits = 0000 indicates no error
    - Parity bits = 1010 indicates bit 10 was flipped

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

How many bits do we need?

- Support that we have a 16-bit message, m=16
- Assume k is the number of correction bits for error correction

$$2^k \geq m + k + 1,$$

$$2^k \geq 16 + k + 1; \text{ therefore, } k = 5$$

k_5	bits 16–21.
k_4	bits 8–15.
k_3	bits 4–7, 12–15, and 20–21.
k_2	bits 2–3, 6–7, 10–11, 14–15, and 18–19.
k_1	bits 1, 3, 5, 7, 9 . . . , 19, 21.

ECC Example

- Assume one byte data value is 10011010_{two}. First show the Hamming ECC code for that byte, and then invert 10^{th} bit and show that the ECC code finds and corrects the single bit error.

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

ECC Example (cont'd)

- Step 1:

Leaving spaces for the parity bits, the 12 bit pattern (8+4) is__ 1_ 0 0 1_ 1 0 1 0

- Step 2:
  - Position 1

Position 1 checks bits 1,3,5,7,9, and 11, which we highlight:__ **1**_ **0** 0 **1**_ **1** 0 **1** 0. To make the group even parity, we should set bit 1 to 0.

  - Position 2

Even

Position 2 checks bits 2,3,6,7,10,11, which is 0_ **1**_ 0 **0** **1**_ 1 **0** **1** 0 or ~~odd~~ parity, so we set position 2 to a 1.

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

ECC Example (cont'd)

- Position 4 & 8

  Position 4 checks bits 4,5,6,7,12, which is 0 1 1_ **0 0 1**_ 1 0 1**0** so we set it to a 1.
  Position 8 checks bits 8,9,10,11,12, which is 0 1 1 1 0 0 1_ **1 0 1 0**, so we set it to a 0.

- Final code: 011100101010
- Inverting bit 10: 011100101**1**10
  - Parity bit 1 is 0 (**0**11**1**0**0**1**0**1**1**10 is four 1s, so even parity; this group is OK).
  - Parity bit 2 is 1 (0**11**10**0**1011**1**0 is five 1s, so odd parity; there is an error somewhere).
  - Parity bit 4 is 0 (011**100**1011**1**0 is two 1s, so even parity; this group is OK).

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

ECC Example (cont'd)

- Parity bit 8 is 1 (0111001**01110** is three 1s, so odd parity; there is an error somewhere).
- Observation: bit 2 & 8 (0101) are incorrect -> 10^{th} bit must be wrong
- Correction: inverting bit 10, done!

NOTE: This material was taken from Summer 2021 *VE370 Intro. to Computer Organization*.

Highly reliable designs

- Fault-tolerant designs/Highly reliable designs involve simpler Hardware:
  - *Error Detection:* The use of parity, residue, and other codes are essential to reliable system configurations.
  - *Action Retry:* Once a fault is detected, the action can be retried to overcome transient errors.
  - *Error Correction:* Since most of the system is storage and memory, an ECC can be effective in overcoming storage faults.
  - *Reconfiguration:* Once a fault is detected, it may be possible to reconfigure parts of the system so that the failing subsystem is isolated from further computation.

Redundancy: carefully applied

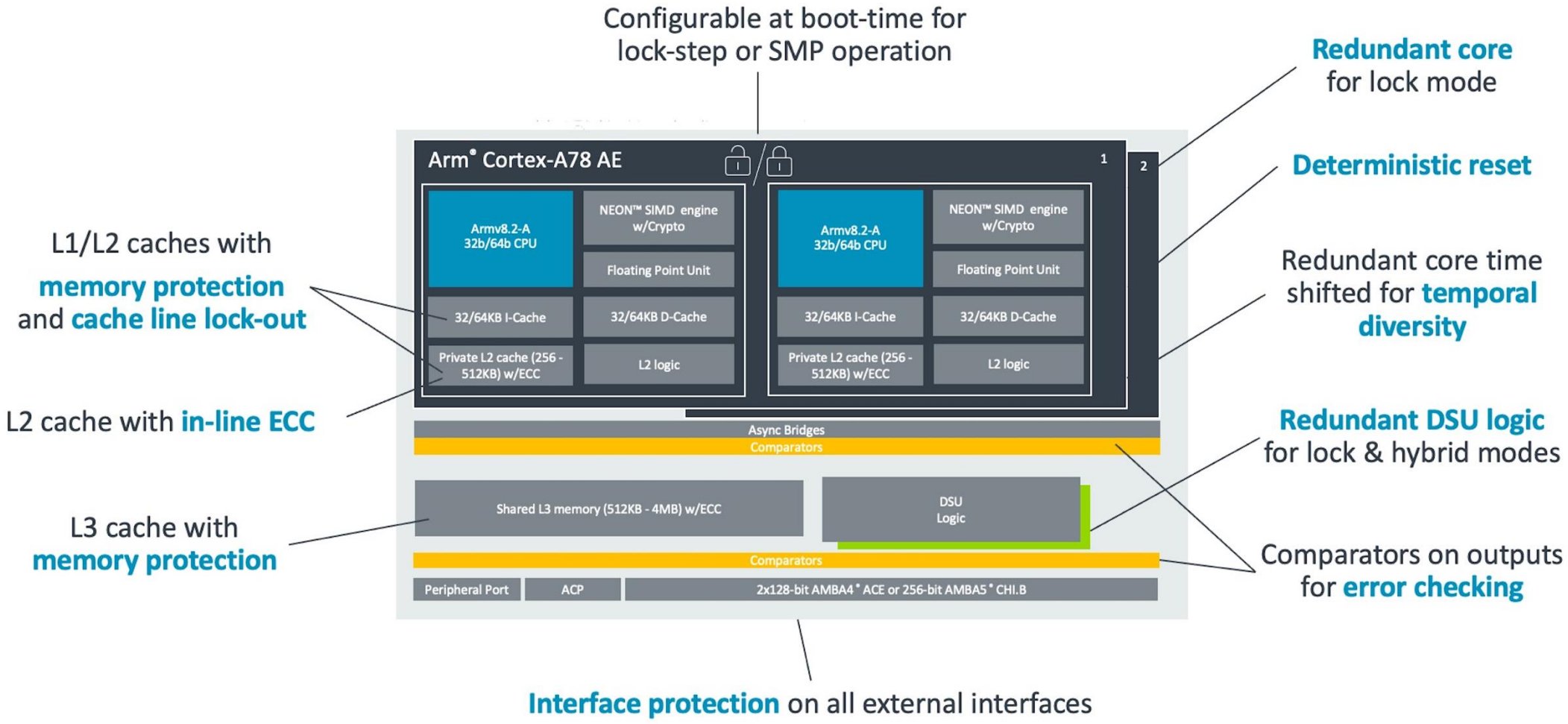
- $P(t) = e^{-t/\tau}$
  - derived in the same way as the yield equation
- TMR (triple modular redundancy) system
  - additional reliability over a time much less than τ, the expected failure time for a single module
- Additional hardware
  - makes the occurrence of multiple module failures more probable

RAID: Redundant Arrays of (Inexpensive) Disks

- Data is stored across multiple disks
- Files are “striped” across multiple disks
- Redundancy yields high data availability
- Availability: service still provided to user, even if some components failed

Image: http://marketexplorereports.blogspot.com/2018/01/global-redundant-array-of-independent.html

Redundancy can also help security



source: https://community.arm.com/arm-community-blogs/b/embedded-blog/posts/arm-cortex-a78ae-on-the-road-to-an-autonomous-future

Main Issues of Physical Reliability

- Electromigration
- Self heating
- Hot problems
- Oxide failure
- Bipolar transistor degradation
- Package/chip power dissipation
- Latchup
- Electro-Static Discharge (ESD) protection

Accelerated life testing is used to simulate aging

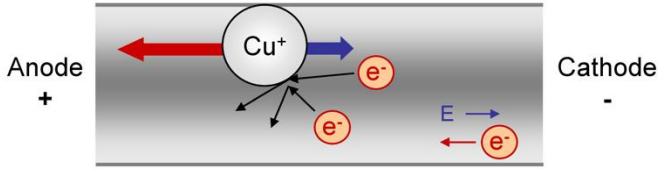
Electromigration

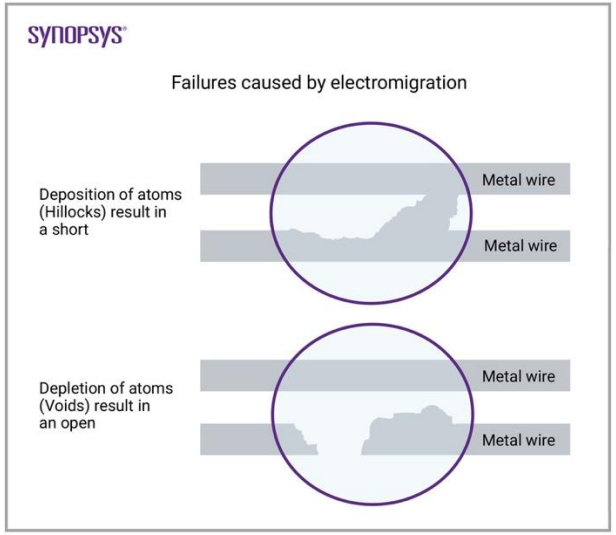
- "Electron wind" causes movement of metal atoms along wires
- Excessive electromigration leads to open circuits
- Most significant for unidirectional current
  - Depends on current density J_{dc} (current / area)
  - Exponential dependence on temperature
  - Dependence of average time of non-failure operation of IC elements on temperature is characterized by dependence of Black:

$$t_{av}(T) = AJ^{-n}\exp(E/kT)$$

A is parameter depending on technology (microstructure); J is the current density; n is the level pointer of density of the current, accepting values =1.0 ÷ 2.0, at changing density of a current within the limits of J = (0.2÷2.0)10^6 Am/cm^2; E is the energy of activation (for semi-conductor IC with aluminum interconnections accepting value E=0.5÷0.6 eV); k- is the Boltzmann constant; T -is the temperature of IC element in ^{0}K.

  - Typical limits: J_{dc} < 1 – 2 mA / mm^2





Images: Wikipedia, Synopsys

Self-heating

- Current through wire resistance generates heat
  - The oxide surrounding wires is a thermal insulator
  - Heat tends to build up in wires
  - Hotter wires are more resistive, slower
- Self-heating limits current densities for reliability
  - Typical limits: $J_r < 15$ mA / mm^2 $J_r = \sqrt{\frac{\int_0^T I(t)^2 dt}{T}}$

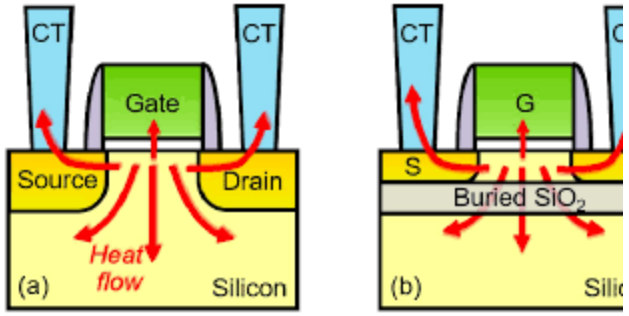
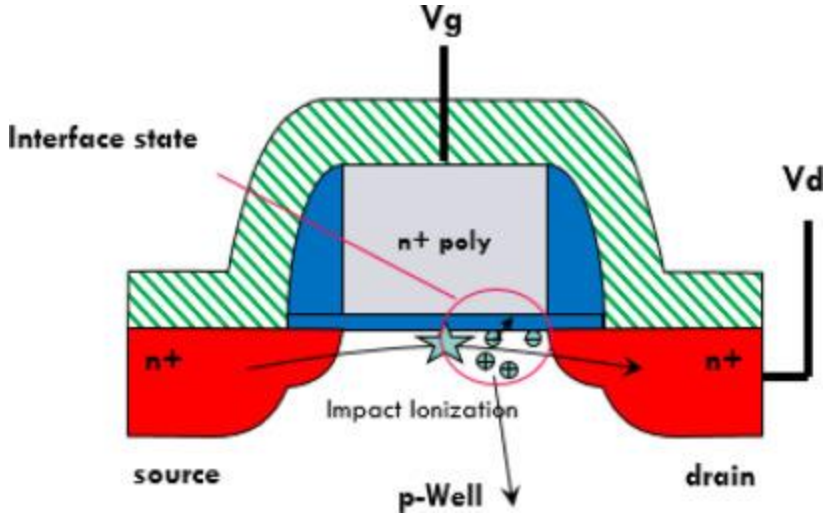


Image: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8691399

Hot Carriers

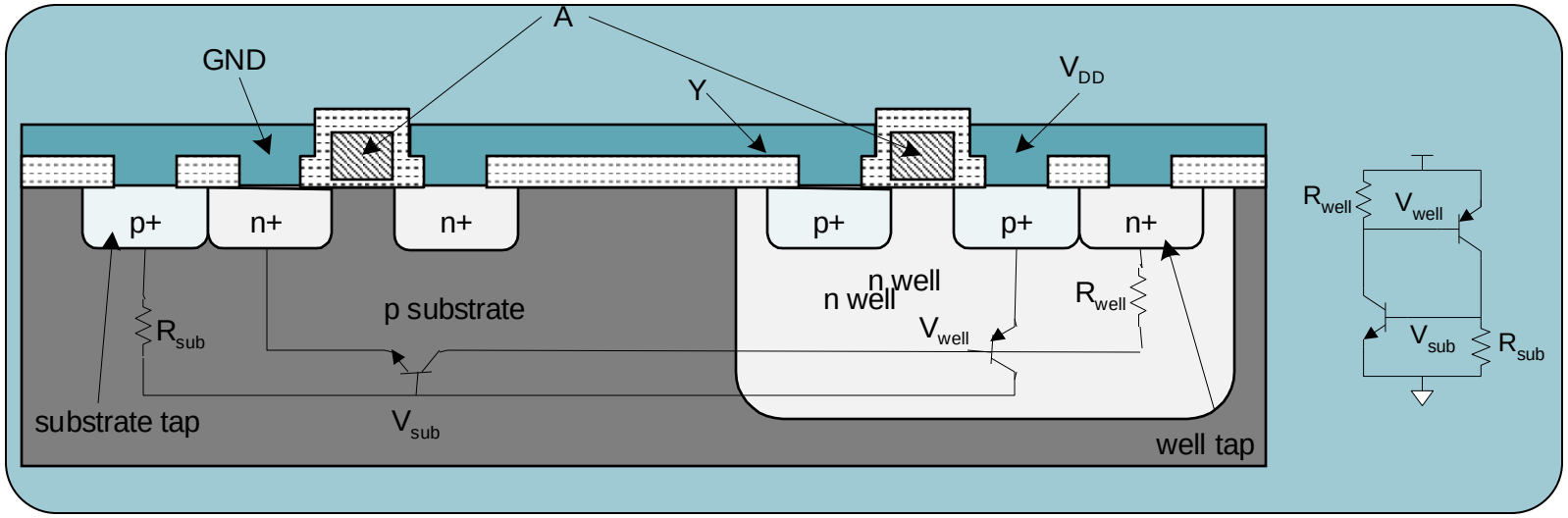
- Electric fields across channel impart high energies to some carriers
  - These “hot” carriers may be blasted into the gate oxide where they become trapped
  - Accumulation of charge in oxide causes shift in Vt over time
  - Eventually Vt shifts too far for devices to operate correctly
- VDD is chosen to achieve reasonable product lifetime
  - Worst problems for inverters and NORs with slow input risetime and long propagation delays



source: sciencedirect.com/topics/engineering/hot-carrier-injection

Latchup

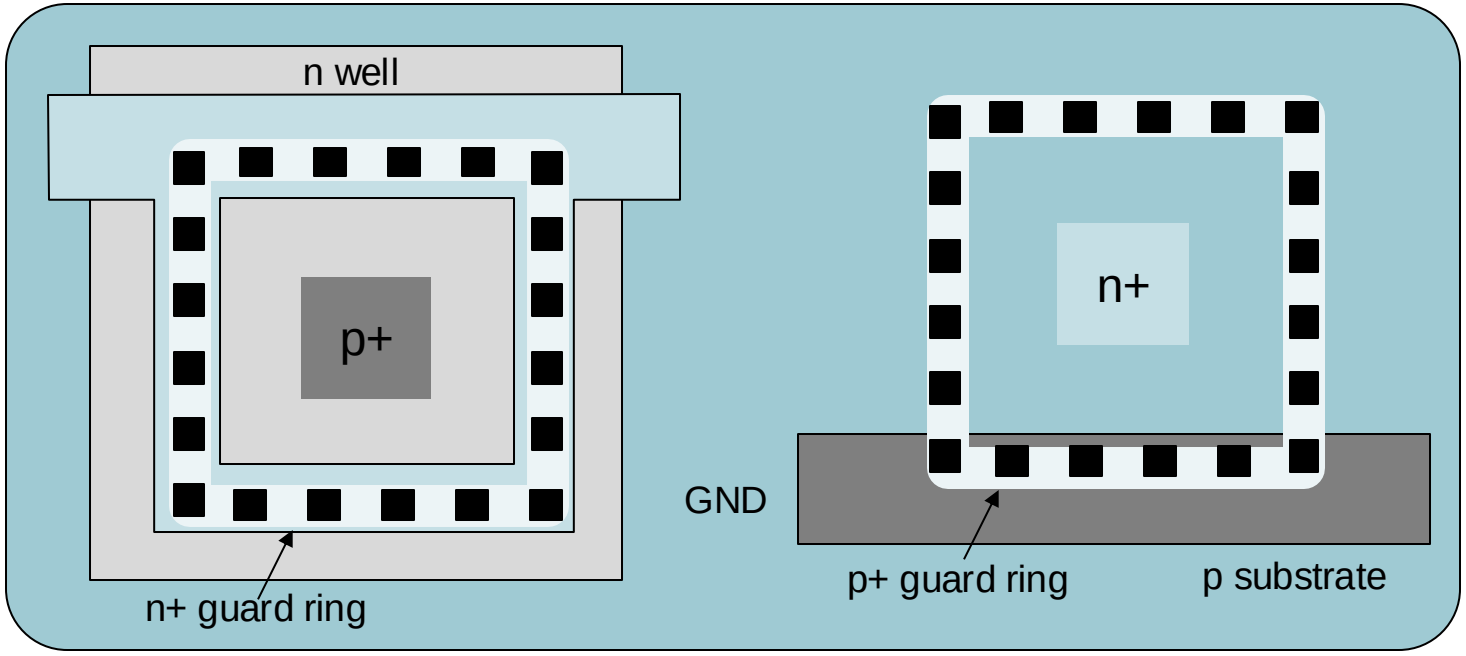
- Latchup: positive feedback leading to VDD – GND short
  - Major problem for 1970's CMOS processes before it was well understood
- Avoid by minimizing resistance of body to GND / VDD
  - Use plenty of substrate and well taps



source: Synopsys

Guard Rings

- Latchup risk is the greatest when diffusion-to-substrate diodes become forward-biased
- Surround sensitive region with guard ring to collect injected charge



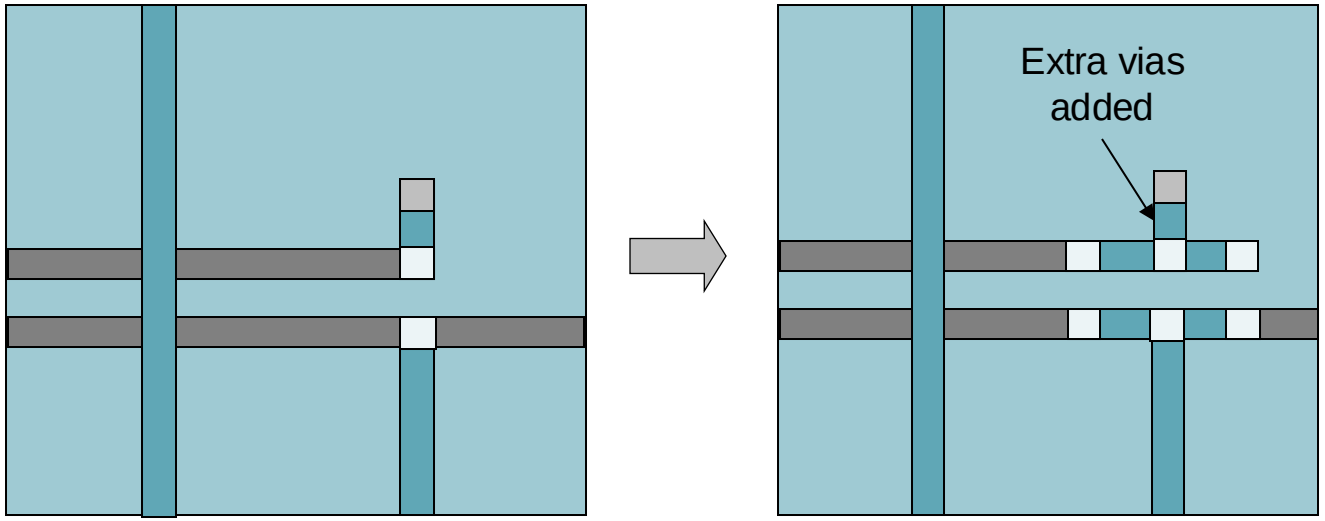
source: Synopsys

Overvoltage

- High voltages can damage transistors
  - Electrostatic discharge
  - Oxide arcing
  - Punchthrough
  - Time-dependent dielectric breakdown (TDDB)
    - Accumulated wear from tunneling currents
- Requires low VDD for thin oxides and short channels
- Use ESD protection structures where chip meets real world

Via Resistance and Reliability

- Replacing one contact with multiple contacts can improve yield and timing (series R reduction)
- Inserts multiple contacts without rerouting



source: Synopsys

INTERCONNECT RELIABILITY

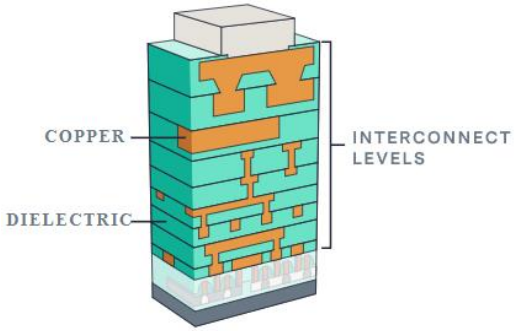
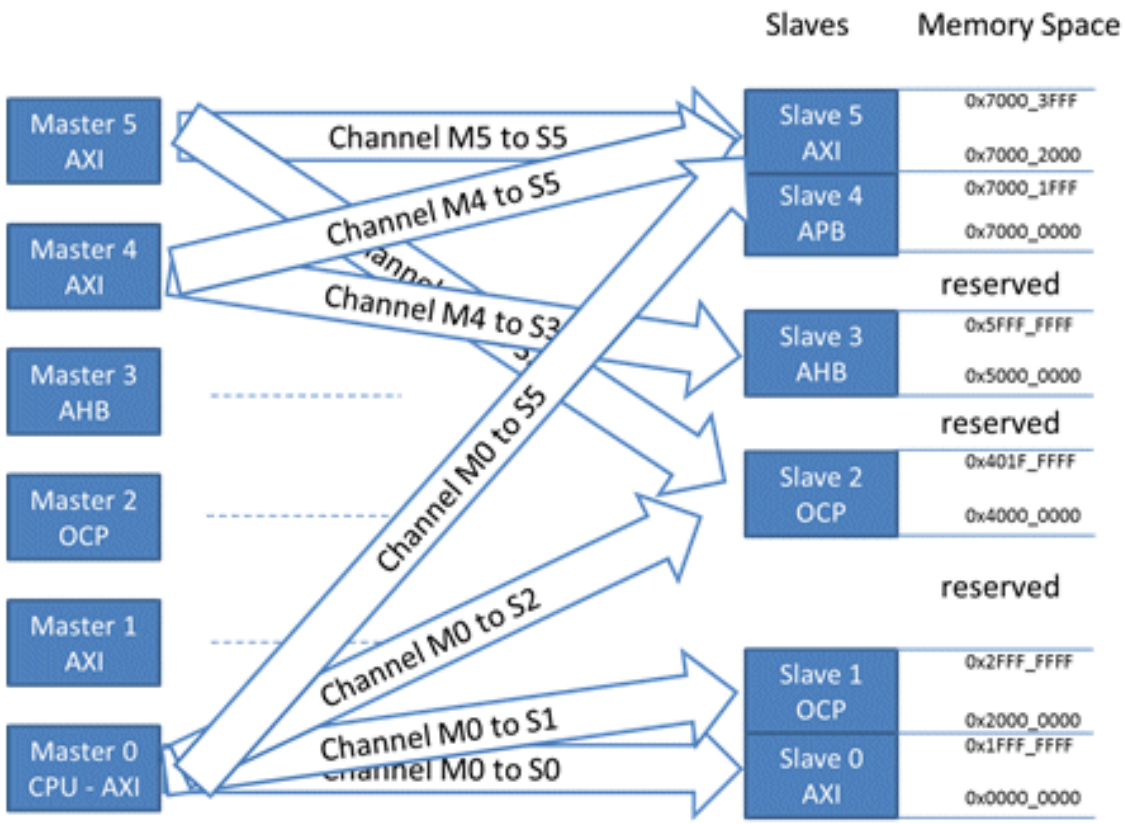


Figure: https://www.lamresearch.com/products/our-solutions/interconnect-solutions/

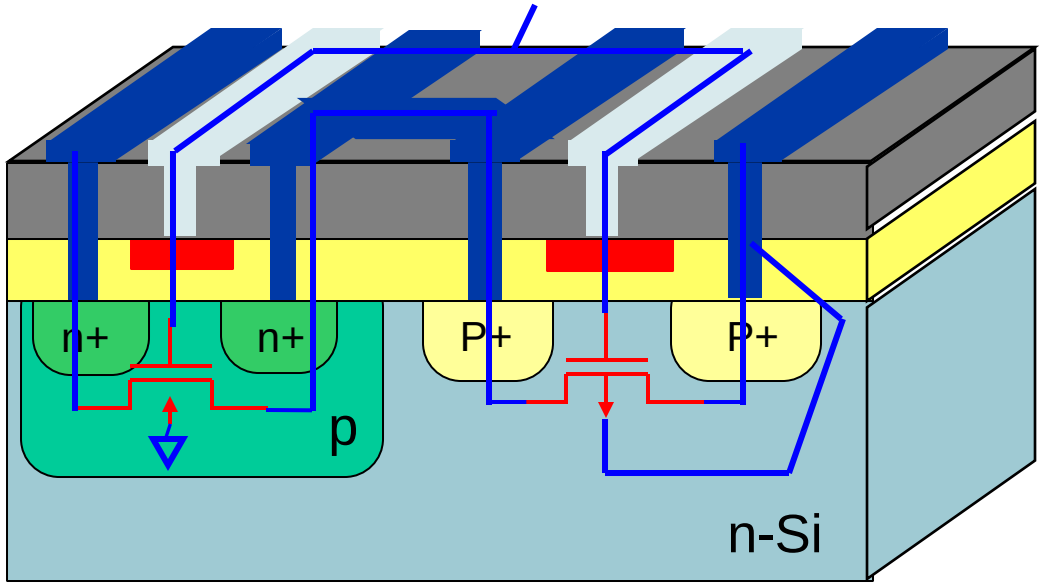
SoC is full of interconnect!



source: https://www.design-reuse.com/articles/31993/soc-interconnect-verification-challenge.html

A CMOS Circuit

- Typical cross-section of a CMOS circuit



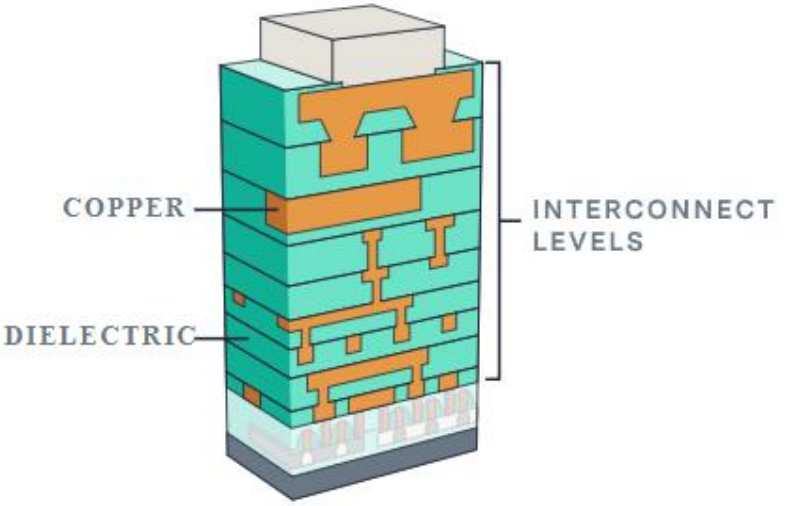
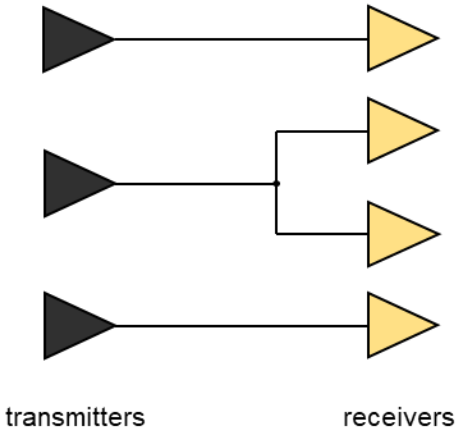
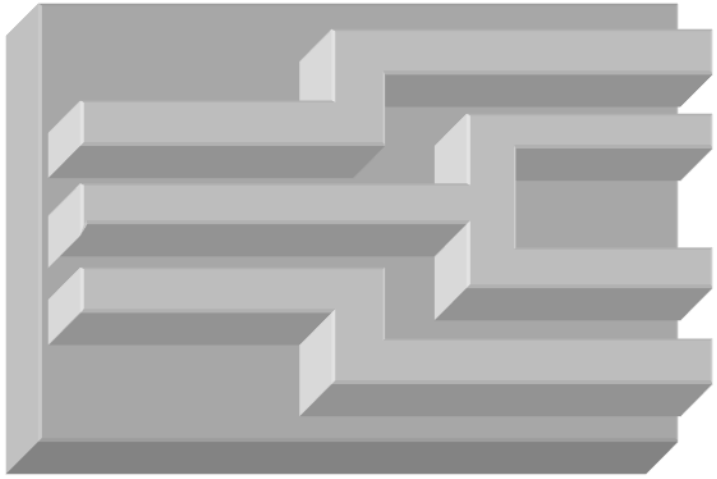


Figure: https://www.lamresearch.com/products/our-solutions/interconnect-solutions/

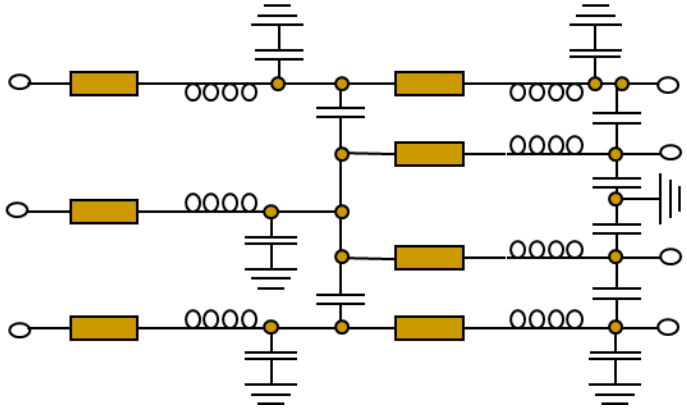
Interconnects



schematics



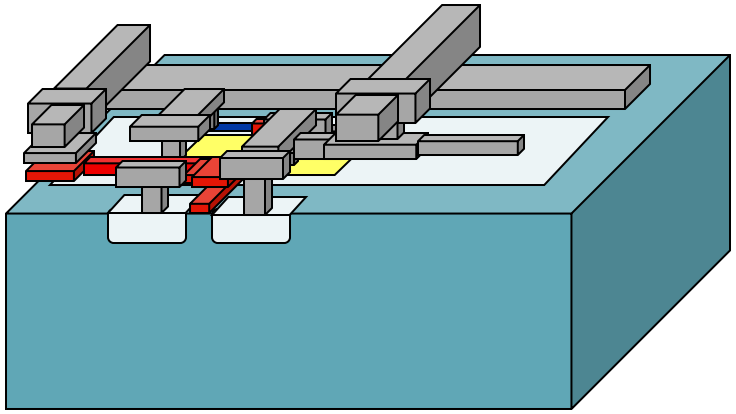
layout



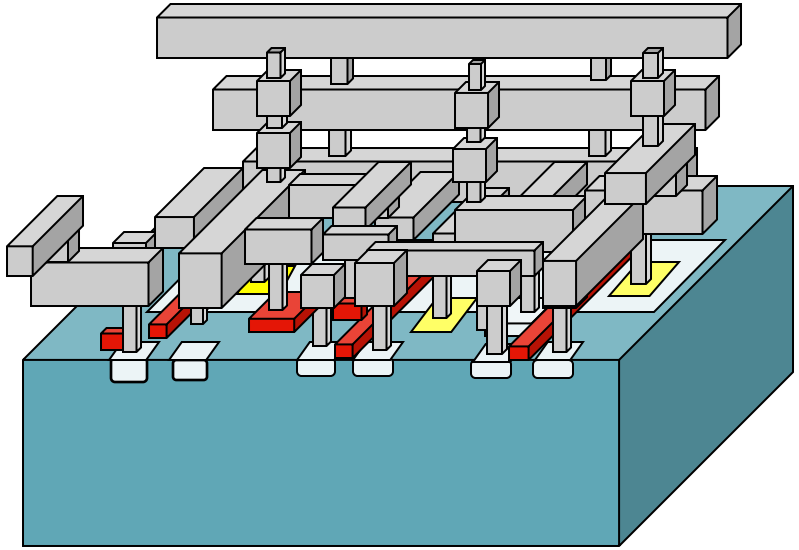
model example

source: Synopsys

Interconnects



0.7µm 2 metal layers
Main delay: gate delay
No signal integrity problems



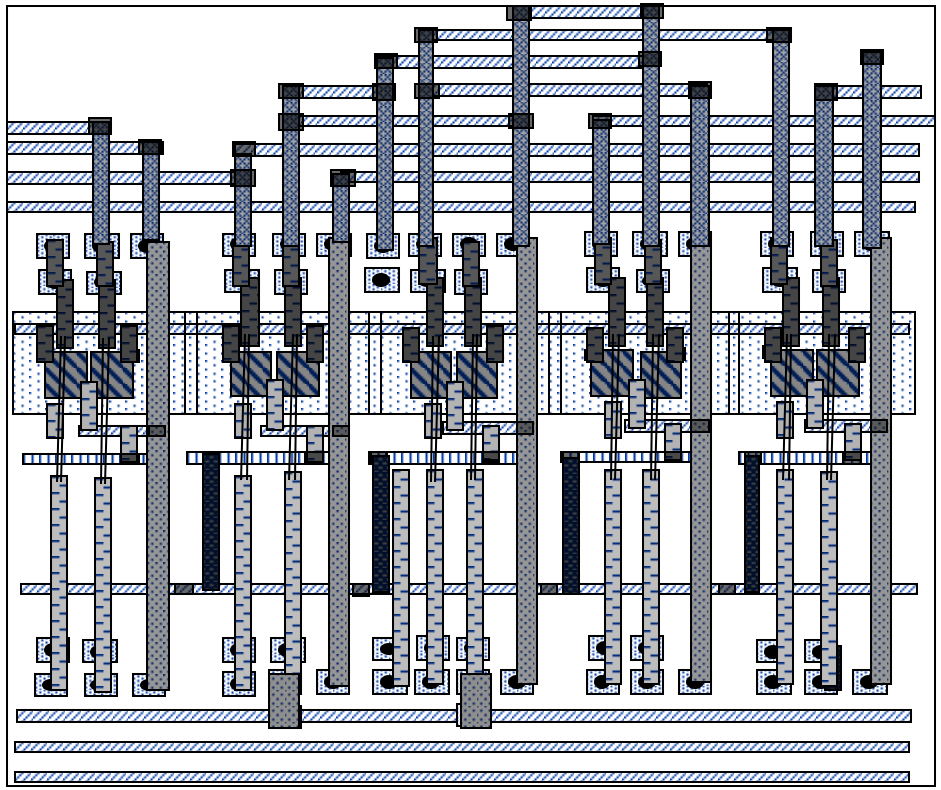
32nm 10 metal layers
Main delay: interconnects
Many signal integrity problems

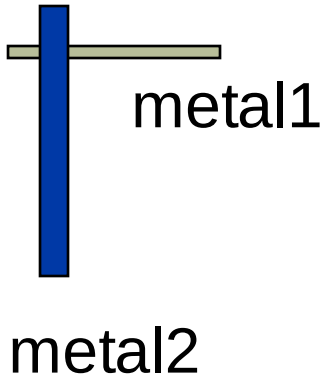
source: Synopsys

Traditional CMOS Technology

2 interconnect layers

1980

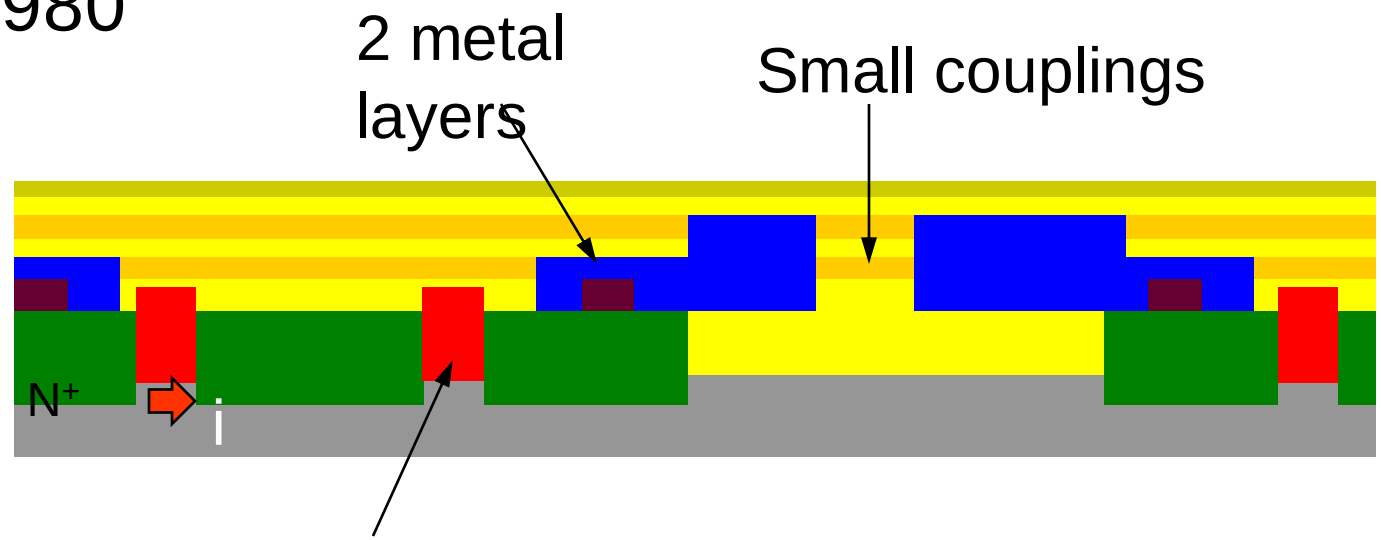




source: Synopsys

Traditional CMOS Technology

1980



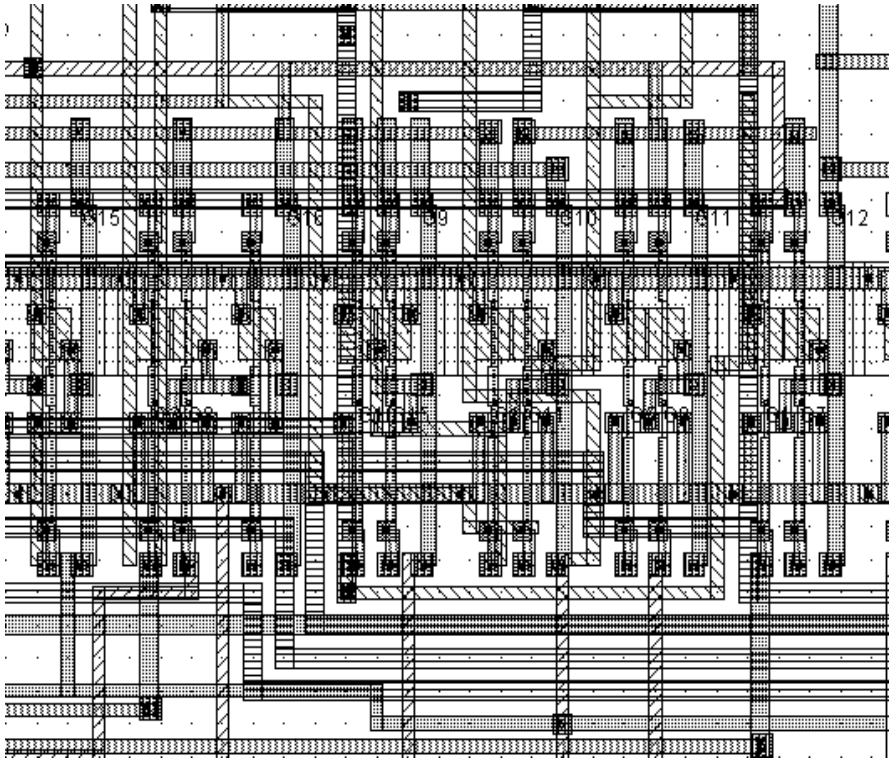
- Slow internal switching
- Interconnect = capacitance

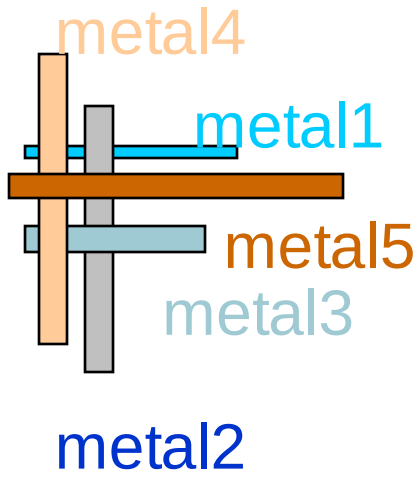
source: Synopsys

Deep Submicron Technology

10 interconnect layers

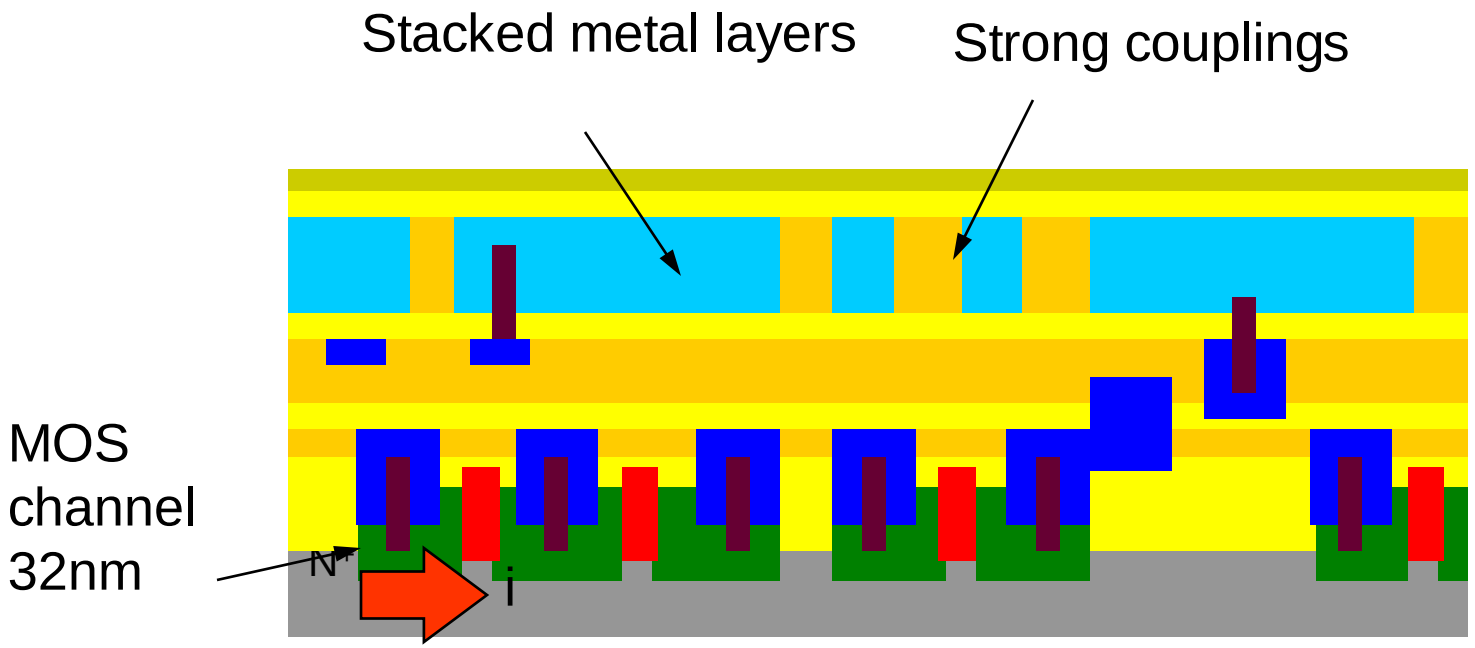
2011





source: Synopsys

Deep Submicron Technology



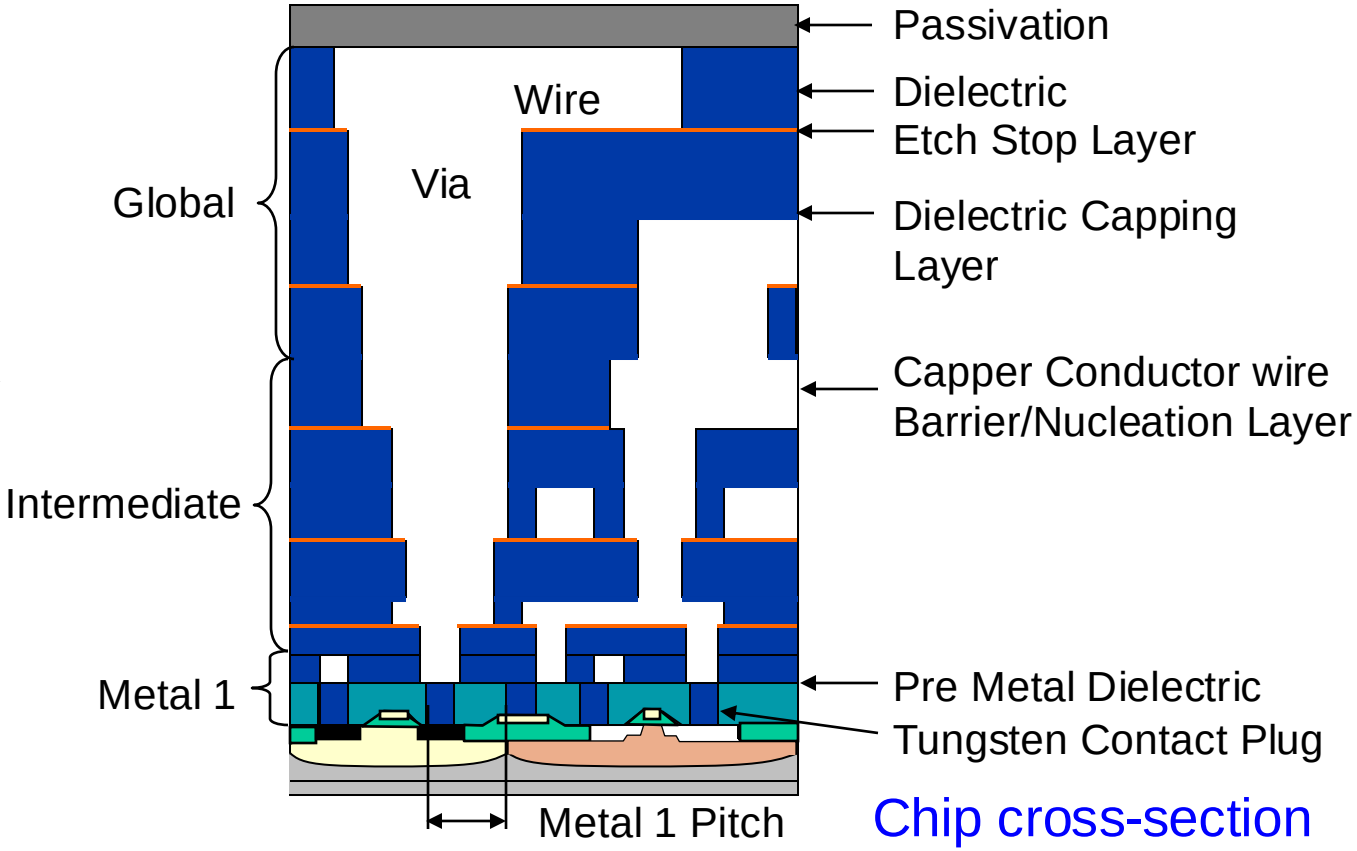
Fast internal switching
Interconnect = capacitance + resistance

source: Synopsys

Interconnects

- Local
- Global

Buses
Vdd and Gnd planes
System Clock

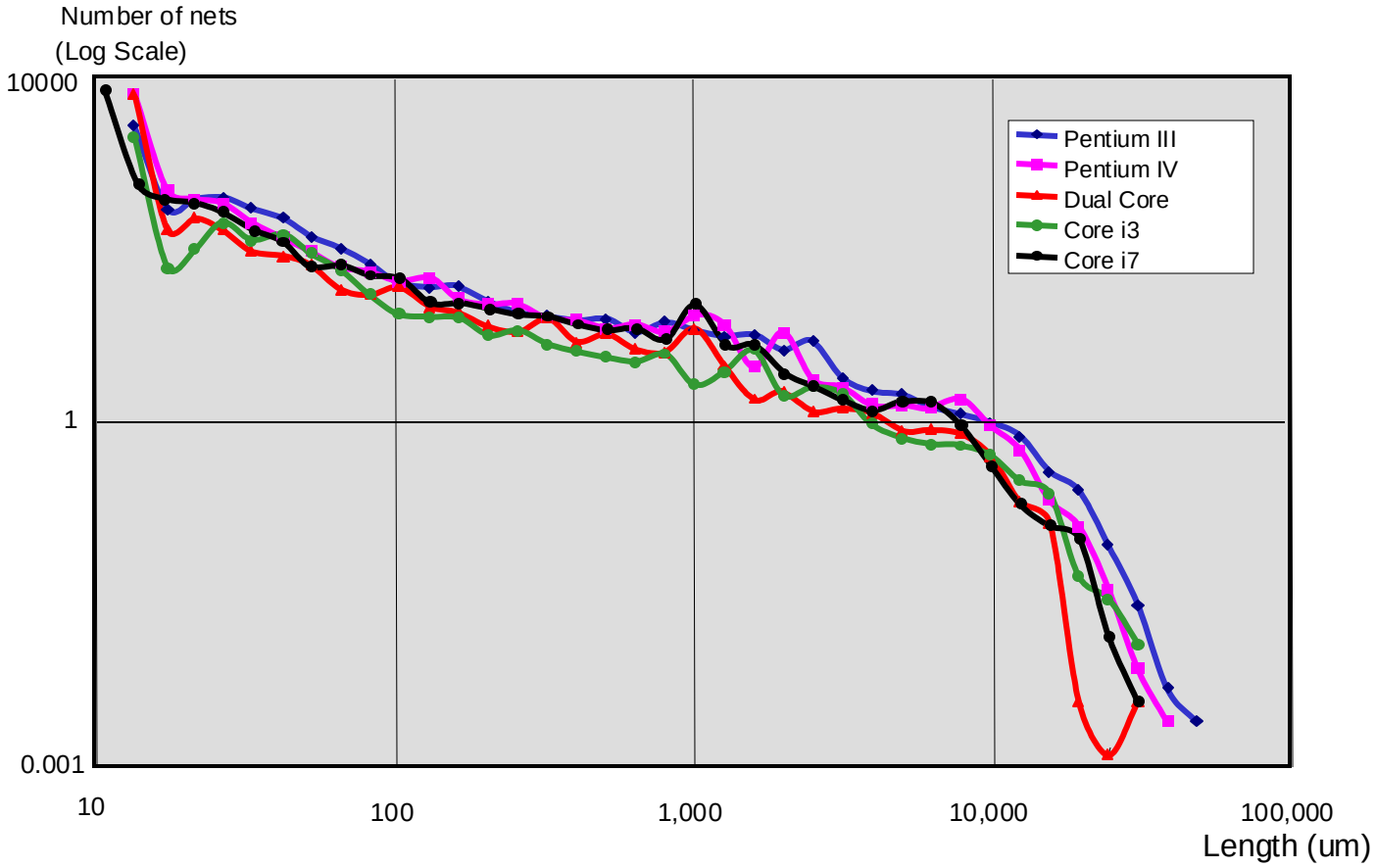


Chip cross-section

source: Synopsys

Interconnect Length Distribution

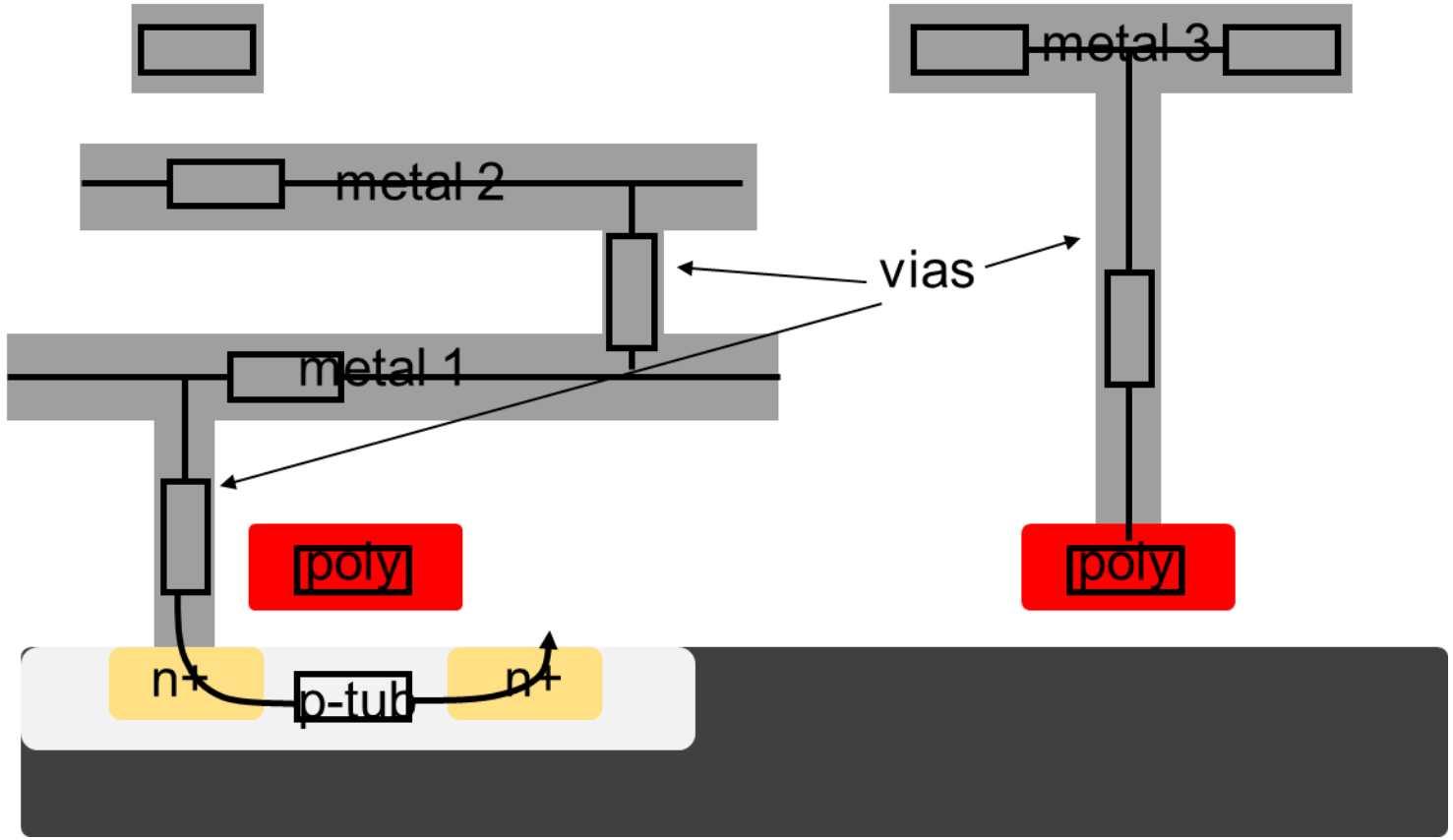
Interconnect distribution does not change significantly



source: Synopsys

MOSFET Resistance

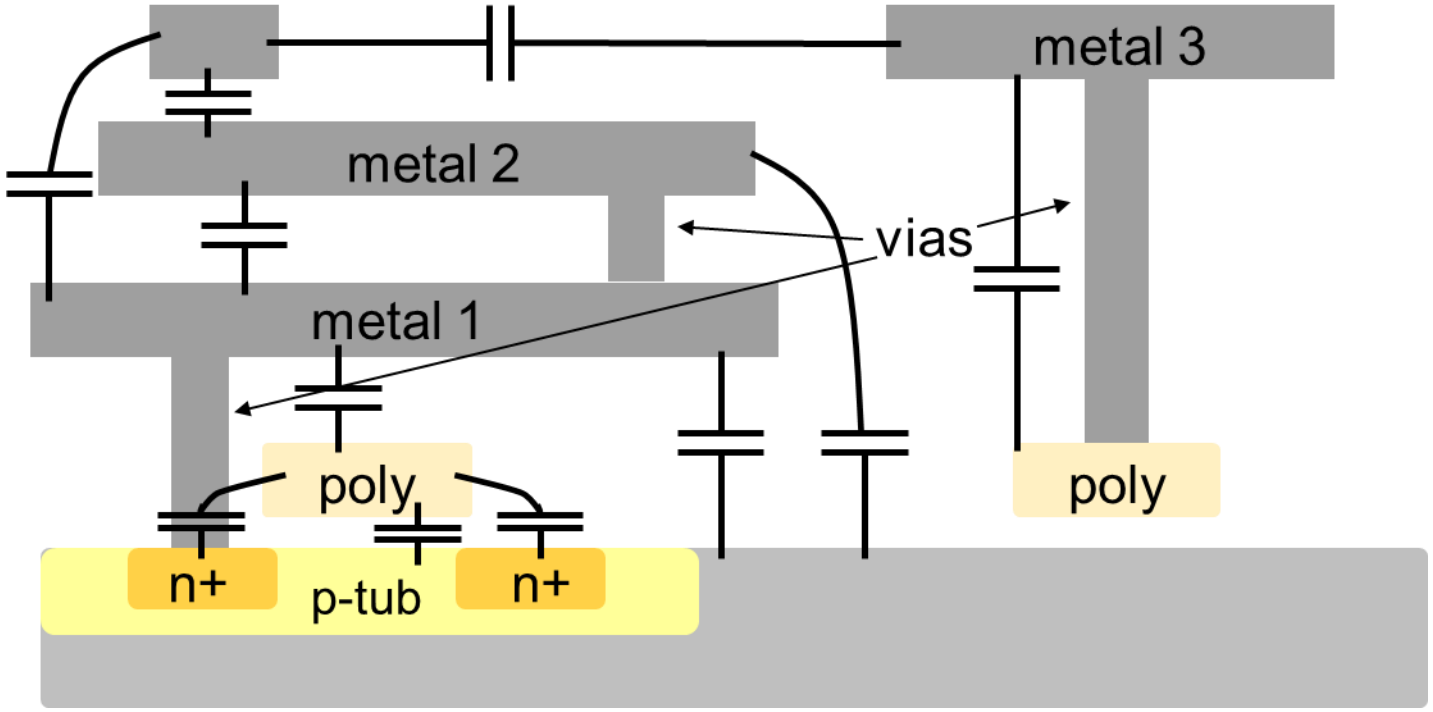
Metal Layers and Resistance



source: Synopsys

MOSFET Capacitance

- Metal layers and capacitance



source: Synopsys

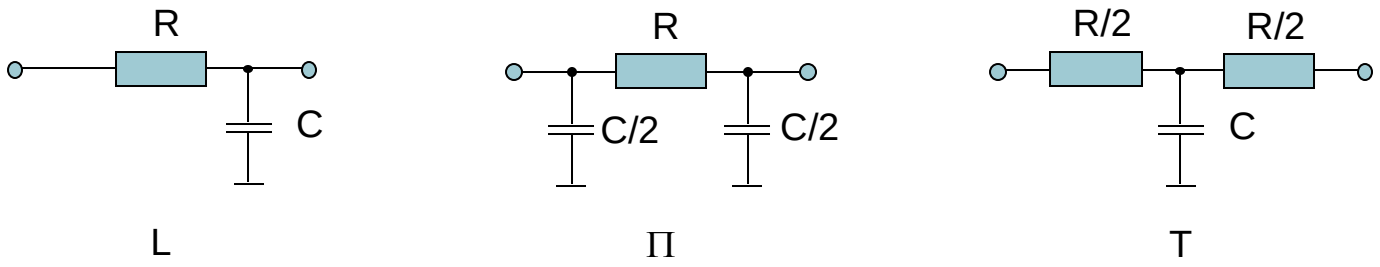
Interconnect Modeling

Wires are a distributed RC circuit

Wires have r=resistance/mm and c=capacitance/mm

(resistance and capacitance per unit length)

Use a simple L, Π or T model assuming C=cL and R=rL



- 3-segment Π-model is accurate to 3% in simulation
- L-model needs 100 segment for same accuracy

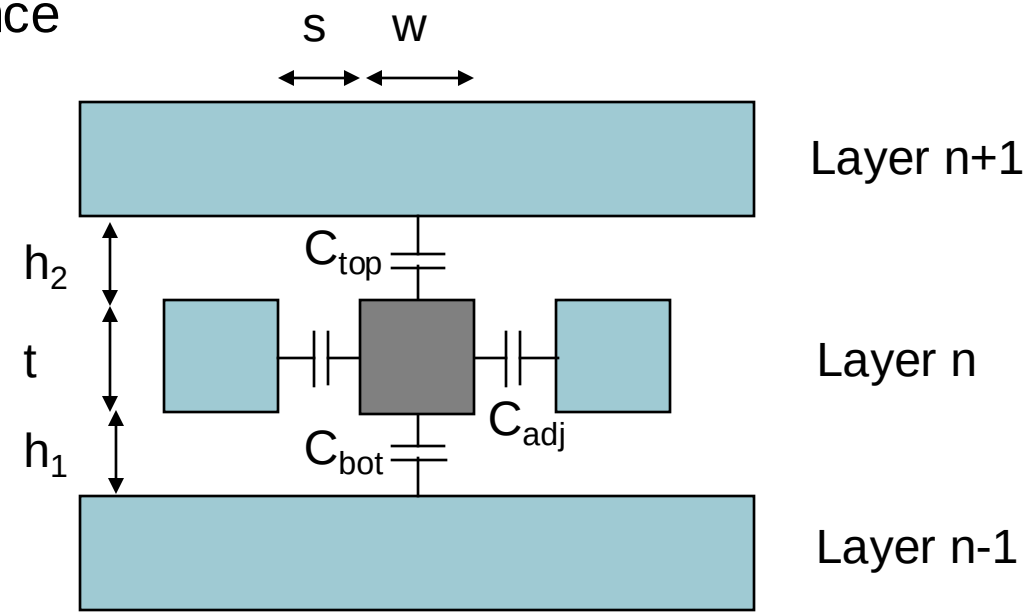
source: Synopsys

New Interconnect Materials

- Aluminum $\rightarrow$ Copper
  - Lower resistivity
  - Higher immunity to electromigration
- $SiO_2 \rightarrow$ Low-k dielectrics

Crosstalk

Wire capacitance

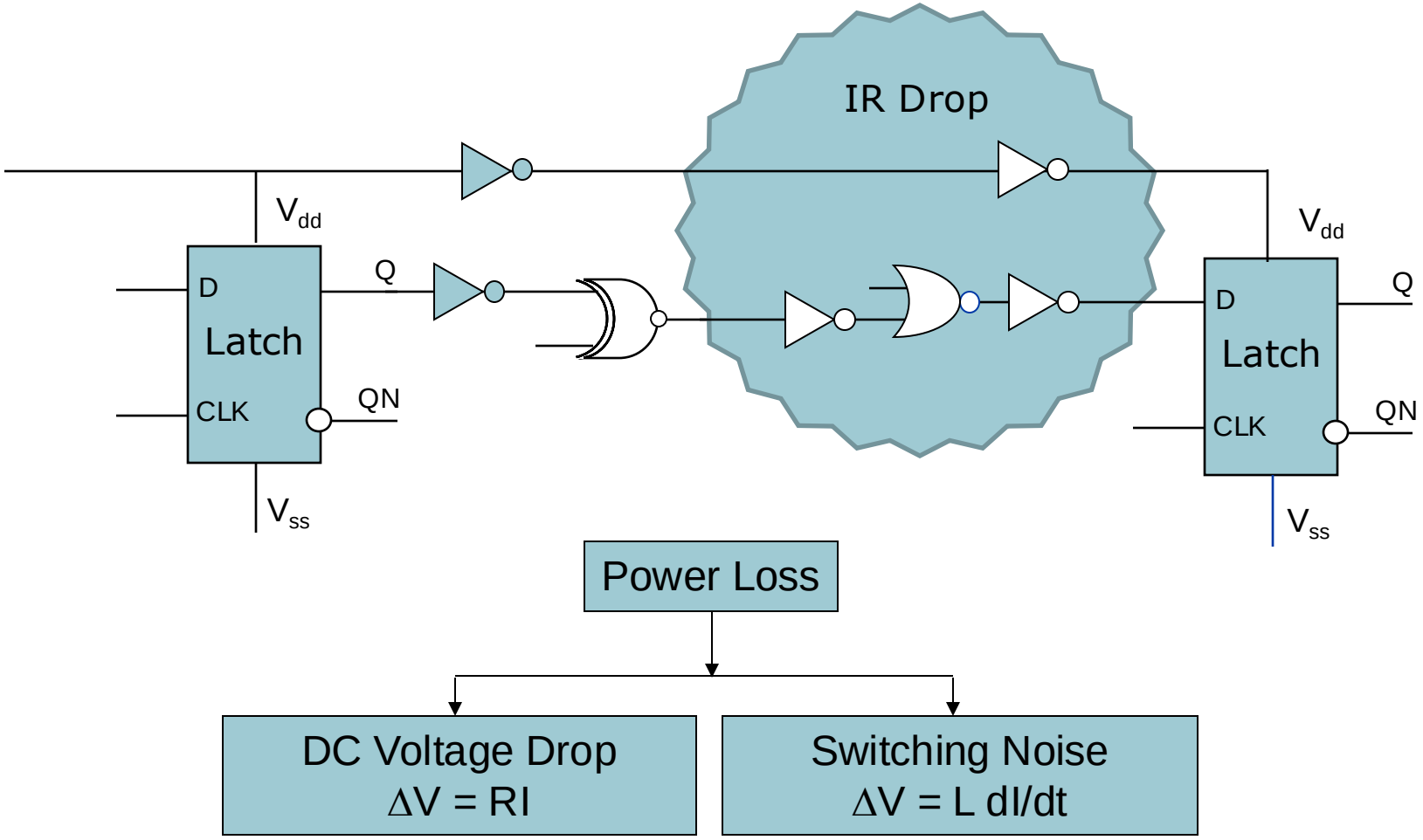


- Wire has capacitance per unit length
  - With neighbors
  - With layers above and below

$$C_{total} = C_{top} + C_{bot} + 2C_{adj}$$

source: Synopsys

IR-Drop



source: Synopsys

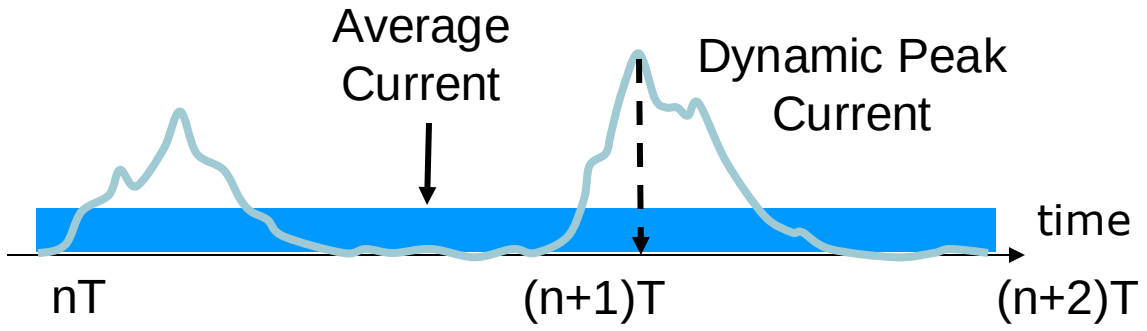
Static and Dynamic IR Analysis

Static IR-Drop

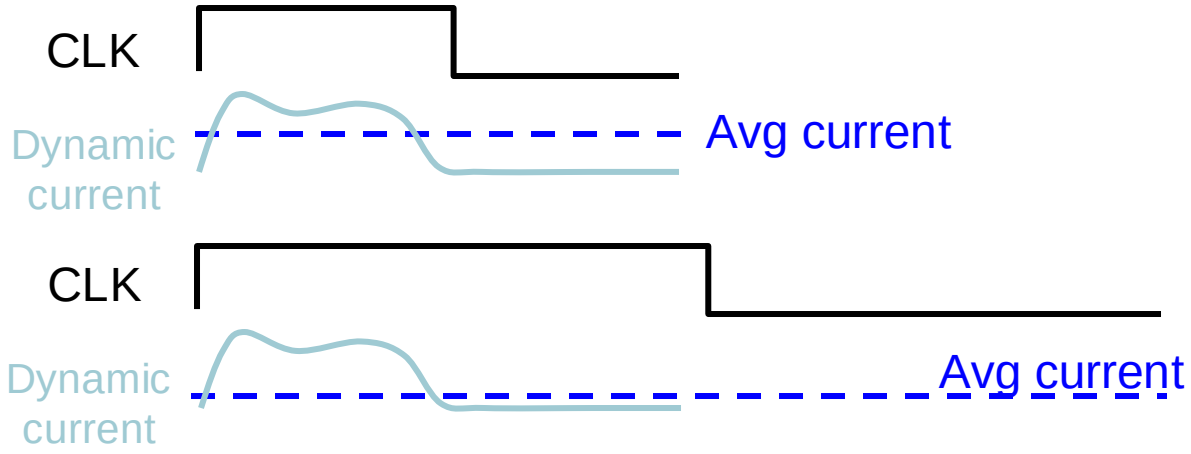
$V_{static} = I_{avg} R$

Dynamic IR-Drop

$V(t) = I(t) R + L \, dI(t)/dt$

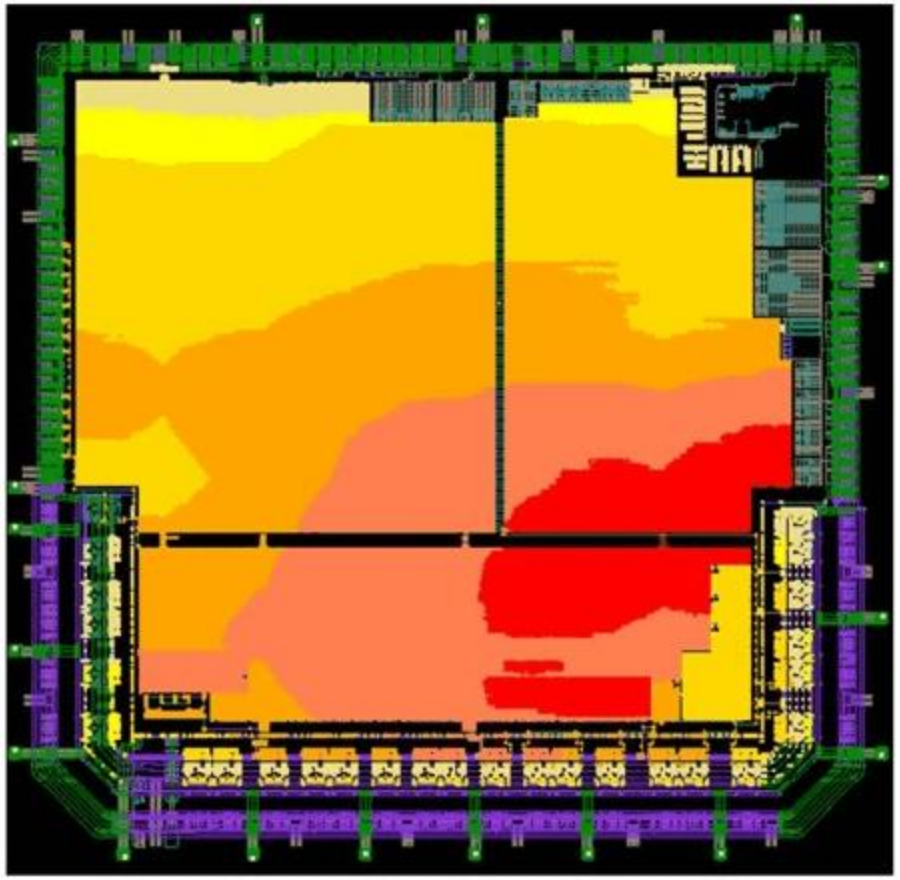


Influence of different frequencies on static and dynamic IR-Drop



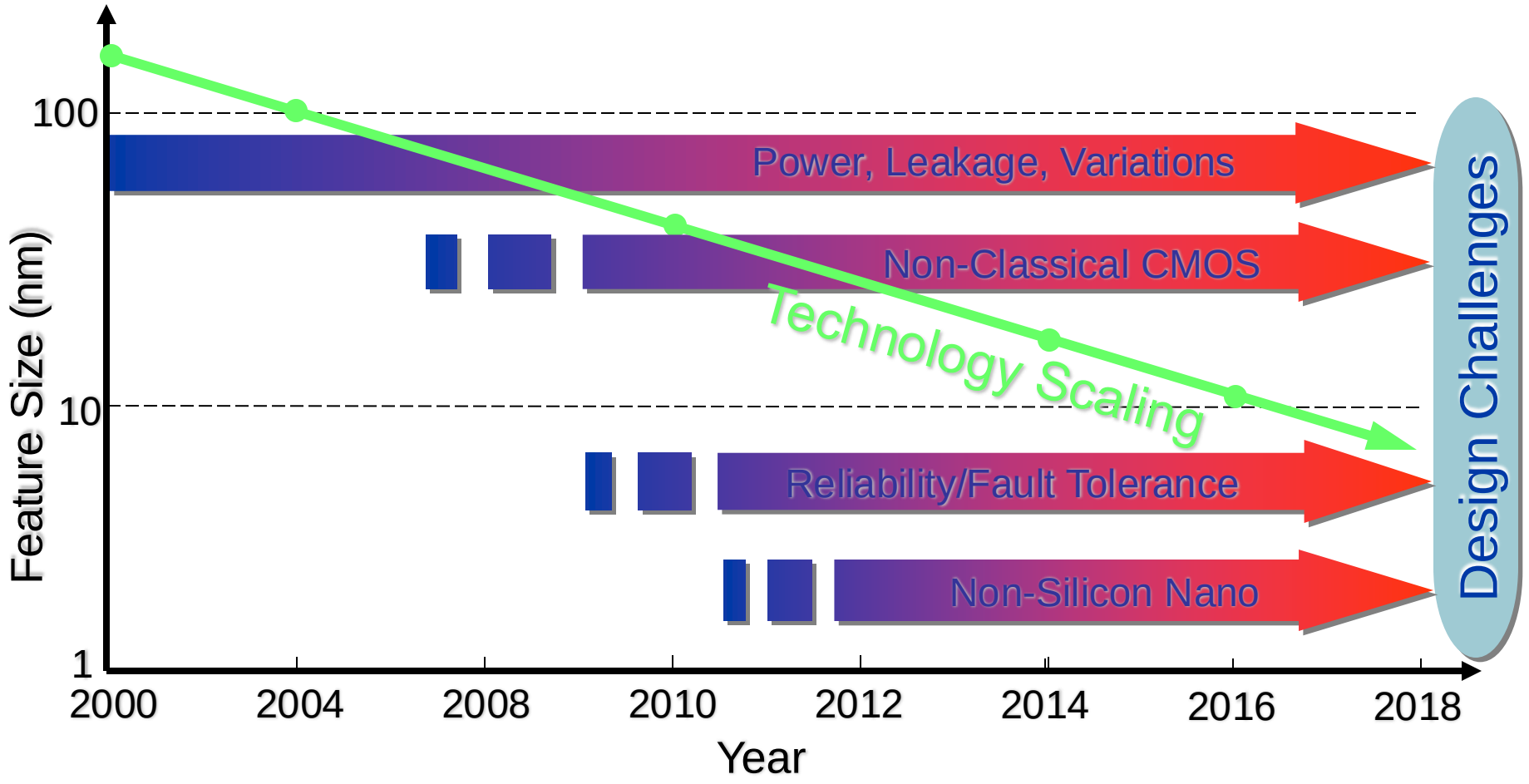
source: Synopsys

Animated Voltage Drop Map



source: Synopsys

Summary: Design Challenges



Summary: SoC Design Space

- best optimise: time, area, power
- area: die floorplanning, rbe model
- power: cooling + battery implications
- reliability: computational integrity, redundancy

Where are we Heading?

- Overview of SoC External Interfaces

Action Items

- Project is upcoming
- Reading Materials
  - Ch. 2

Acknowledgement

Slides in this topic are inspired in part by material developed and copyright by:

- Dr. Wayne Luk (Imperial College)
- Dr. Gul N. Khan (Ryerson University)
- Synopsys Courseware