

Design of VLSI Integrated Circuits

A (very) deep dive into processors...

Olivier Sentieys

IRISA/INRIA – Cairn team

University of Rennes 1

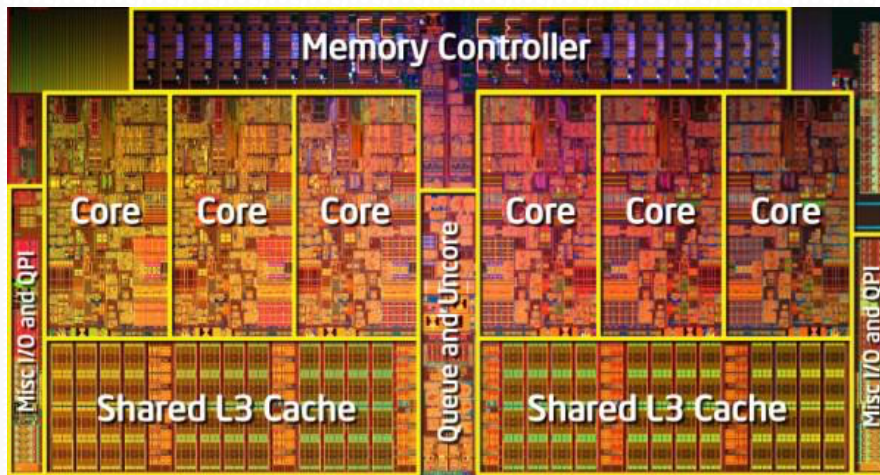
olivier.sentieys@inria.fr



http://people.rennes.inria.fr/Olivier.Sentieys/?page_id=95

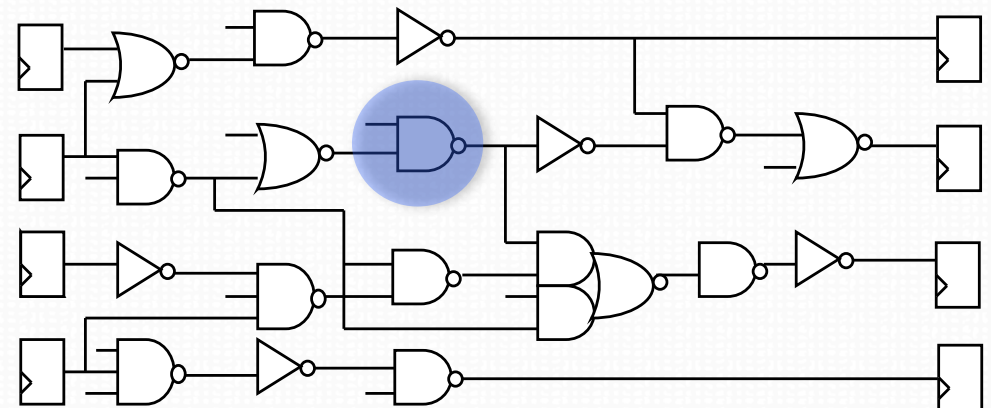
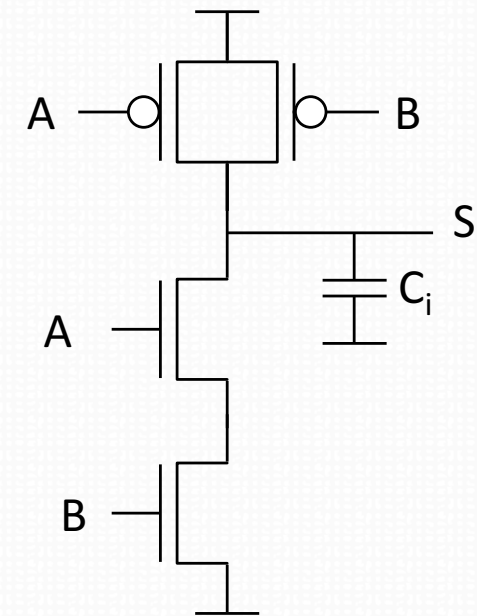
VLSI Design

- Chips, logic gates and transistors



Intel's Xeon Chip

```
#pragma hls_design top
void my_design (int *a, int *o) {
  process static int i,j;
  begin for(i=1;i<=n-1; i++)
    if ( for(j=1;j<=n-1; j++)
      A   a[i][j] = (a[i-1][j]+a[i][j]+a[i][j-1])/3.0;
      S ...
    end }
end process;
```



Key Questions

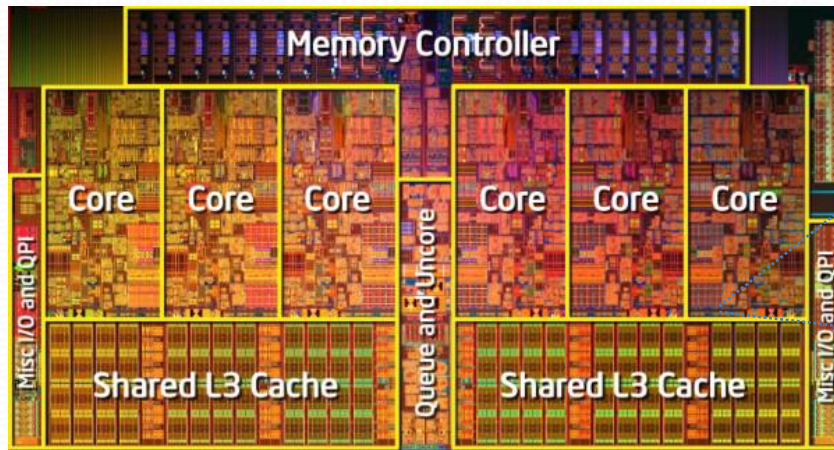
- A deep dive into processors... *(I hope not too deep)*
- What is **CMOS**? How basic logic **gates**, registers and memory are designed?
- How to calculate the delay and the maximal **frequency**?
- How much **power** does my processor consume?
- What can advanced semiconductor **technology** bring?
- Are (homogeneous) multicores the right solution for performance or energy efficiency?

Outline

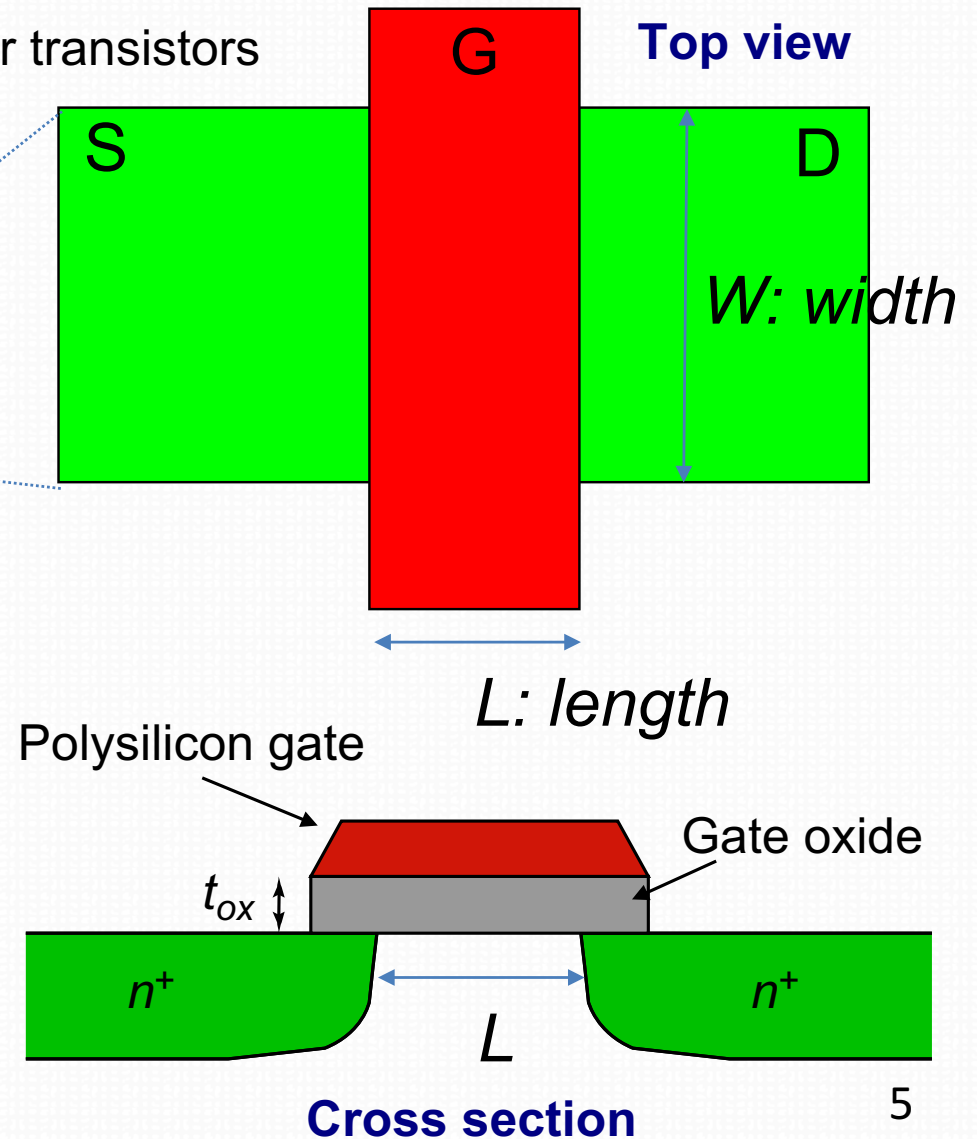
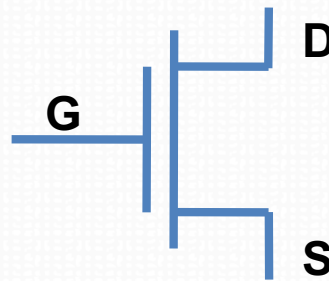
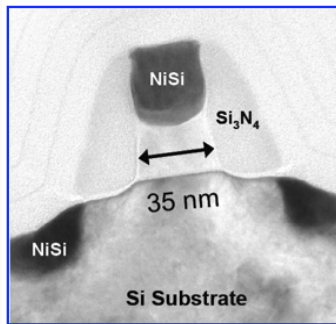
- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

Fundamental Building Block: MOSFET Transistor

Now several billions of transistors



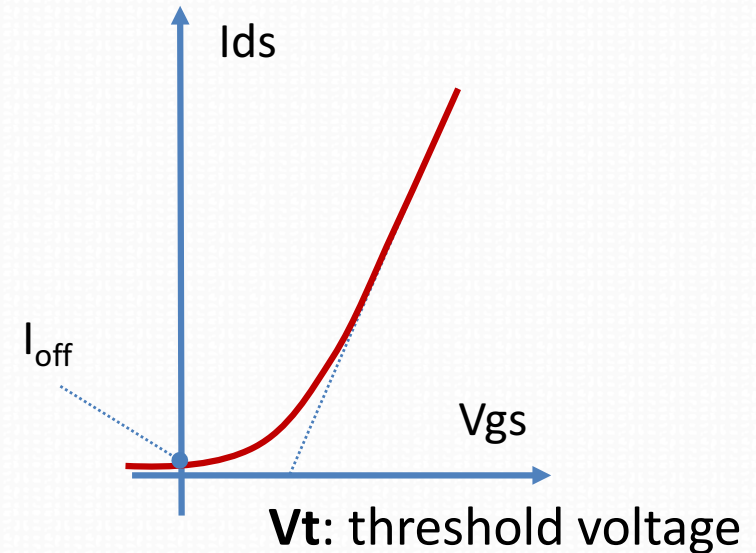
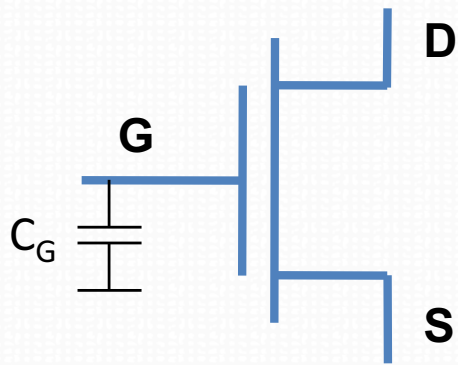
Intel's Xeon Chip



MOSFET: Metal Oxide Silicium Field Effect

The Basic Element: Transistor

- Transistor as a switch



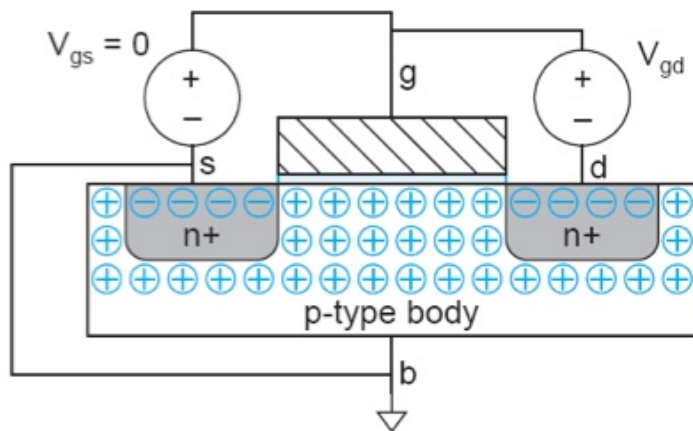
- $V_{gs} > V_t$: NMOS on
 - Resistance R_{DS}
 - $V_{gs} < V_t$: NMOS off
 - Leakage I_{off}
- Gate: capacitance C_G
 - Switch: resistance R_{DS}

The Basic Element: Transistor

- Cutoff or sub-threshold mode:

$$V_{GS} < V_t$$

$$R_{DS} \approx +\infty$$

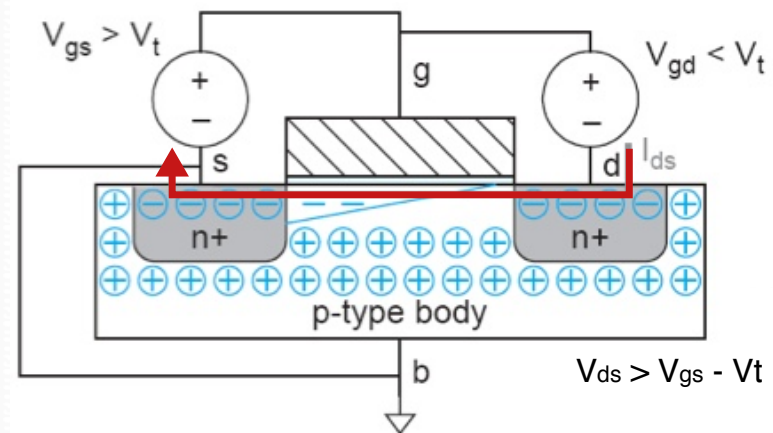


Cutoff: $V_{gs} = 0V$, V_{ds} can be $0V$ or V_{dd}
No Channel, $I_{ds} = 0$

- Saturation mode:

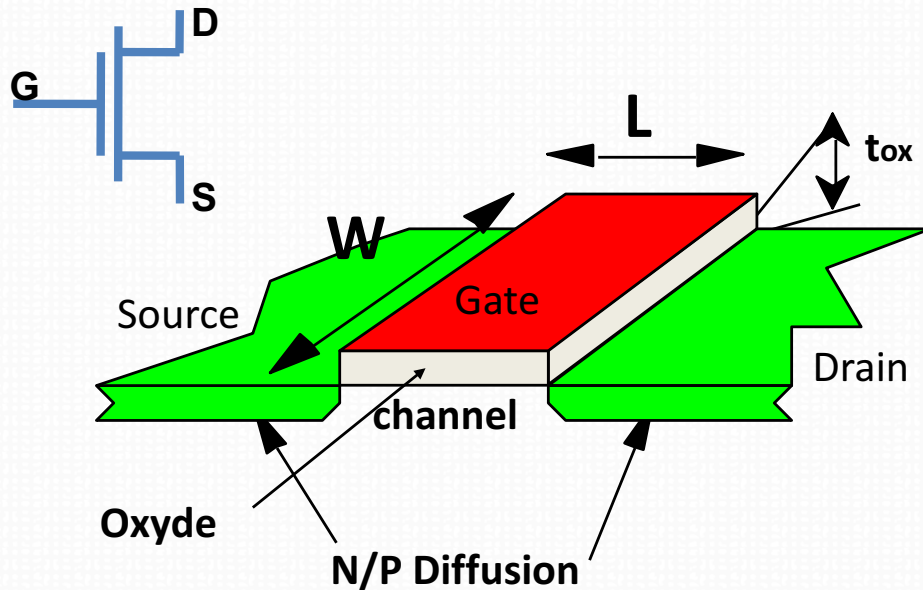
$$V_{GS} > V_t \text{ and } V_{DS} > V_{GS} - V_t$$

- A channel is created which allows current to flow between the drain and the source



Saturation: Channel Pinched Off,
 I_{ds} independent of V_{ds}

MOS Transistor Models

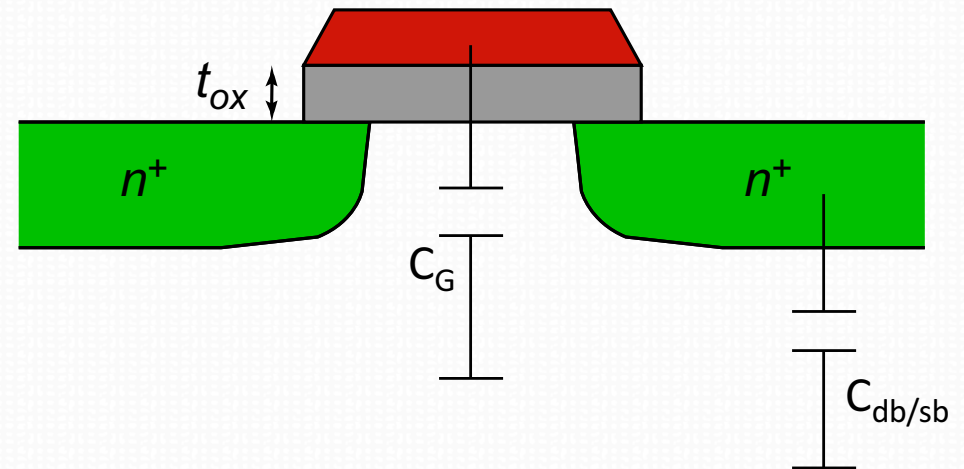
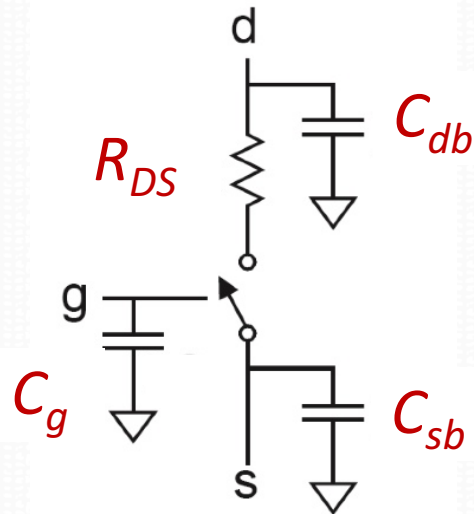
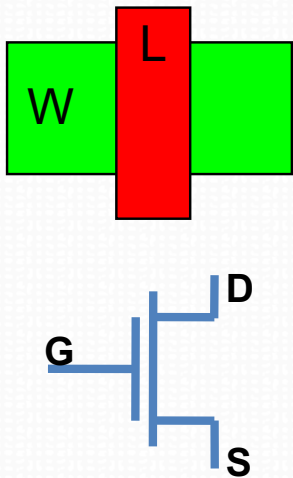


- W : gate width
- L : gate length
- tox : oxyde width (# $L/10$)
- $K = \mu \cdot \epsilon \cdot W / (tox \cdot L) = \mu \cdot Cox \cdot W / L = k W / L$
- Cox : gate oxide capacitance per unit area
- μ : charge-carrier effective mobility
 NMOS (electrons) $\mu_N = 500 \text{ cm}^2/\text{V-sec}$ # $2 \mu_P$
 PMOS (holes) $\mu_P = 270 \text{ cm}^2/\text{V-sec}$
- ϵ : oxyde permittivity # $4 \epsilon_0 = 3.5 \cdot 10^{-13} \text{ F/cm}$

$$I_{ds} = \begin{cases} 0 & \text{off} & V_{gs} - V_{th} < 0 \\ K \left[(V_{gs} - V_{th})V_{ds} - \frac{V_{ds}^2}{2} \right] & \text{linear} & 0 < V_{ds} < V_{gs} - V_{th} \\ \frac{K}{2} (V_{gs} - V_{th})^2 & \text{saturated} & 0 < V_{gs} - V_{th} < V_{ds} \end{cases}$$

- K defines transistor speed, $K \propto W/L$, $K_{NMOS} \sim 2 \cdot K_{PMOS}$
- Temperature increases $\rightarrow \mu$ decreases

NMOS Parasitic Elements

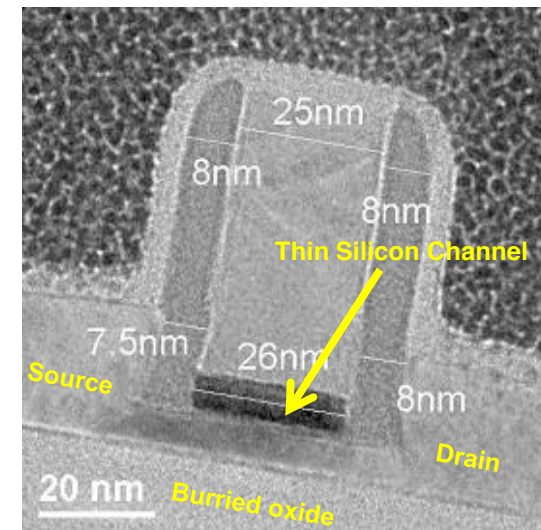
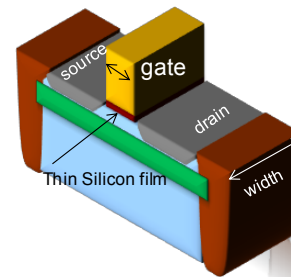
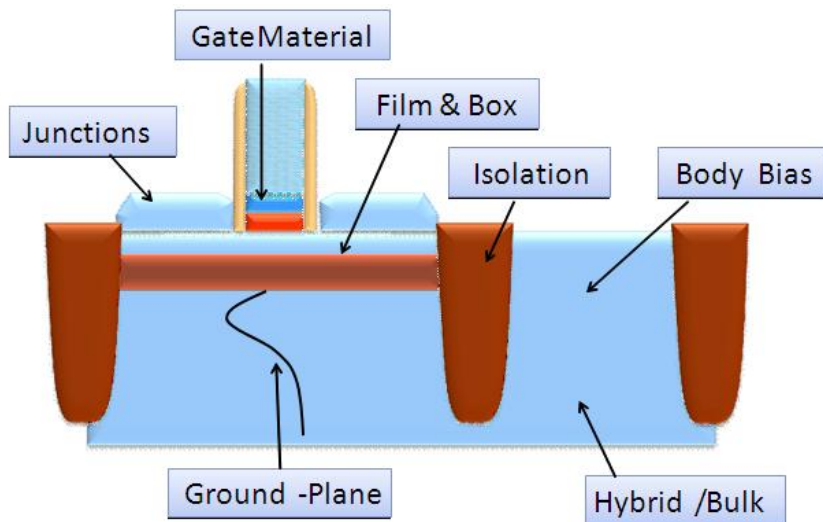
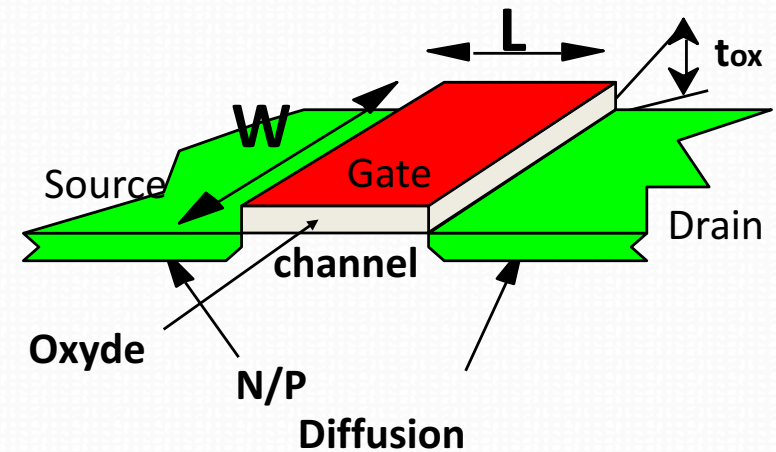


- Drain-Source Resistance: $R_{DS} = \frac{L}{W} \frac{1}{k(V_{dd} - V_t)}$
- Gate Capacitance: $C_g = \frac{\epsilon W \cdot L}{t_{ox}} = W \cdot L \cdot C_{ox}$
- Drain/Source-Bulk Cap.: $C_{db} = C_{sb} \approx W \cdot L \cdot C_j$

$$\text{Delay} \propto R_{DS} \cdot C_g \propto \frac{L^2}{V_{dd} - V_t}$$

Transistors

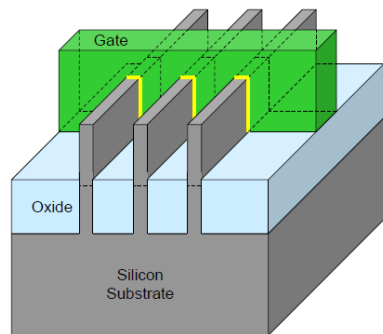
- Bulk CMOS
- Ultra Thin Body (FD) – SOI
 - Total dielectric isolation
 - Lower S/D capacitances & leakages
 - Latch-up immunity
 - Improved VT variation
 - Promoted by STMicroelectronics



Transistors

- Intel FinFET: transistors go 3D

22 nm Tri-Gate Transistor

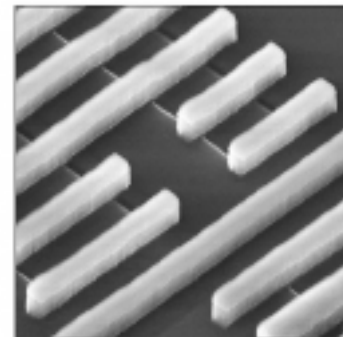


Tri-Gate transistors can have multiple fins connected together to increase total drive strength for higher performance

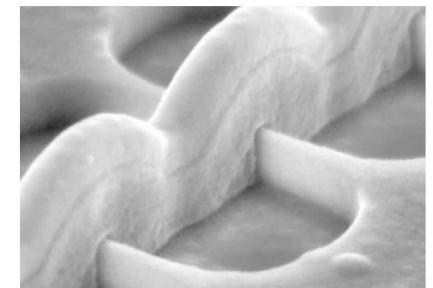
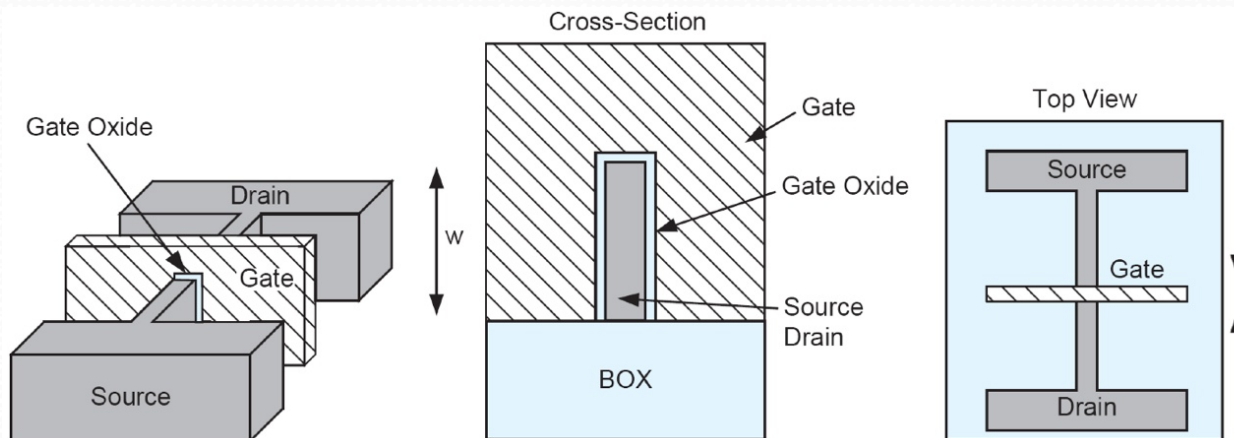
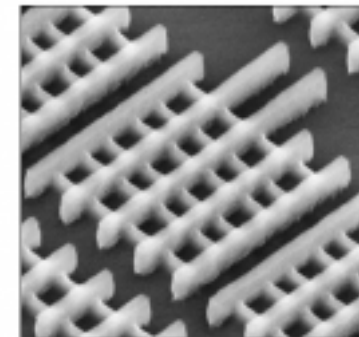


7

32 nm Planar Transistors



22 nm Tri-Gate Transistors



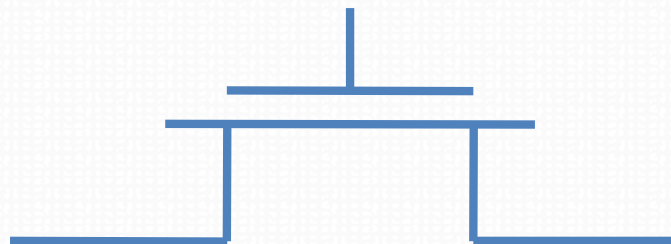
NMOS/PMOS Transistors

- NMOS
 - A '0' is well transmitted
 - A degraded '1' is transmitted ($V_{dd} - V_{tn}$)

- $V_{gs} < V_{tn}$



- $V_{gs} > V_{tn}$

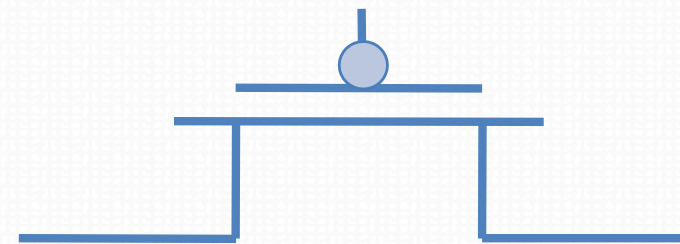


- PMOS
 - A '1' is well transmitted
 - A degraded '0' is transmitted ($V_{ss} + |V_{tp}|$)

- $V_{gs} < |V_{tp}|$



- $V_{gs} > |V_{tp}|$

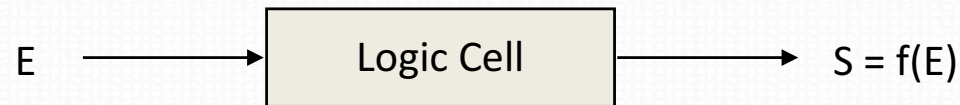


Outline

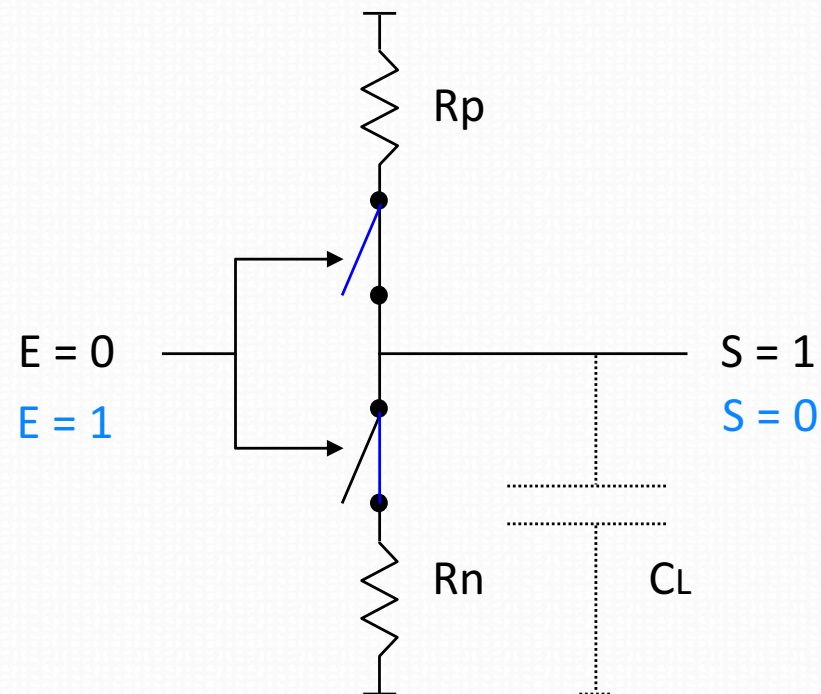
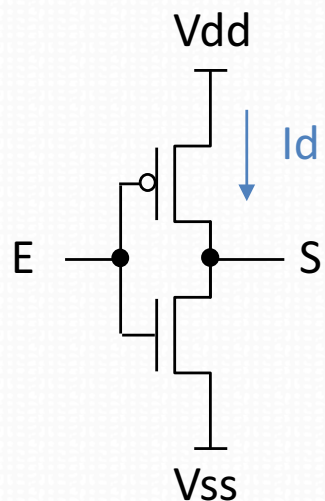
- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

Combinatorial Logic Cells

- Complementary Logic (CMOS)
 - CMOS Static Logic

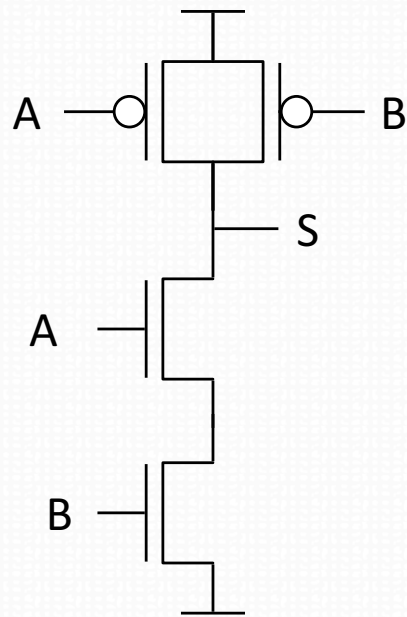


CMOS Inverter

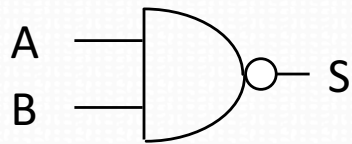


NAND and NOR

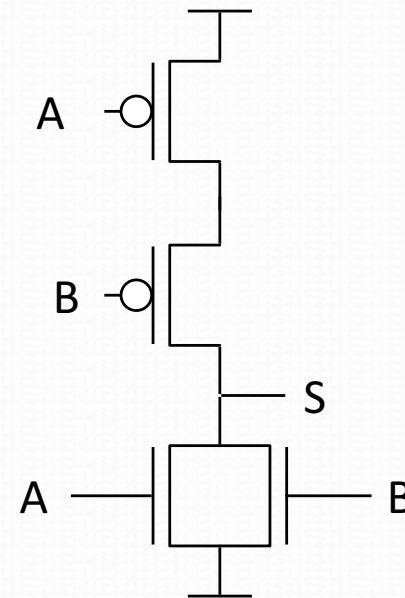
NAND



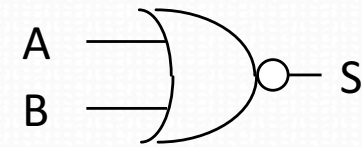
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



NOR



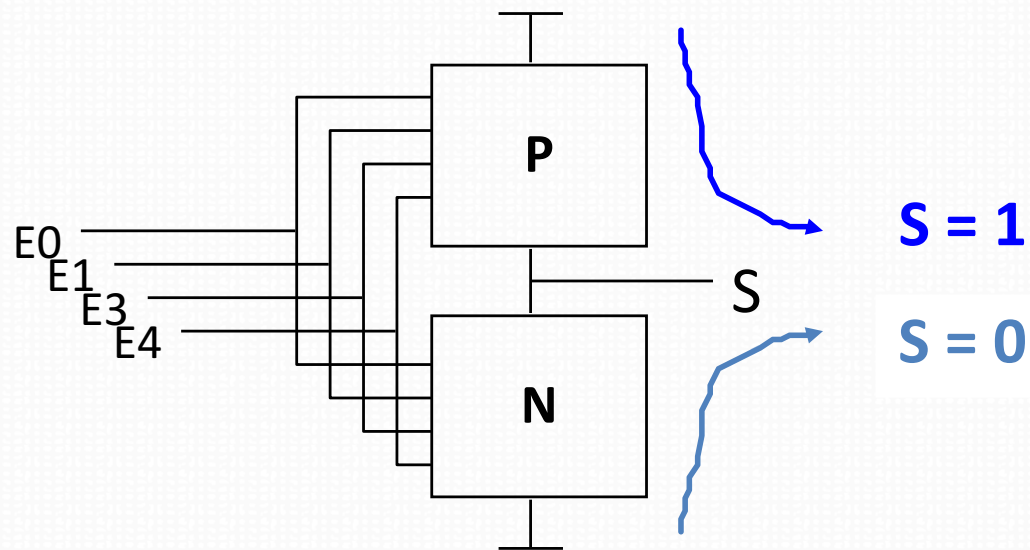
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



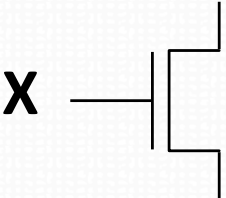
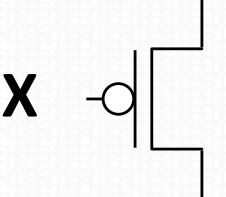
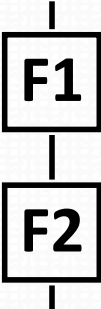
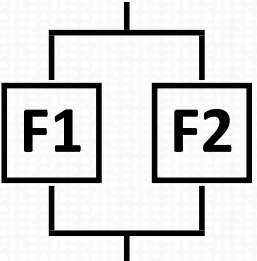
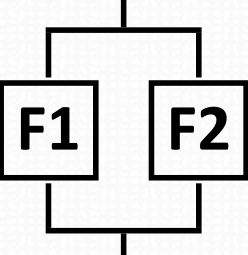
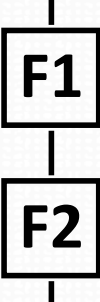
Complex gates

- One CMOS stage can generate any sum-of-product or product-of-sum:

$$S = f(E_1, E_2, \dots, E_N) = \overline{\text{SUM} [\text{PROD}]} = \overline{\text{PROD} [\text{SUM}]}$$



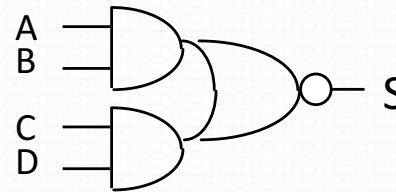
General rules for constructing $F(X)$

F	N network	P network
X		
F1.F2		
F1+F2		

Static Logic

- Examples
 - Direct application of the design rules

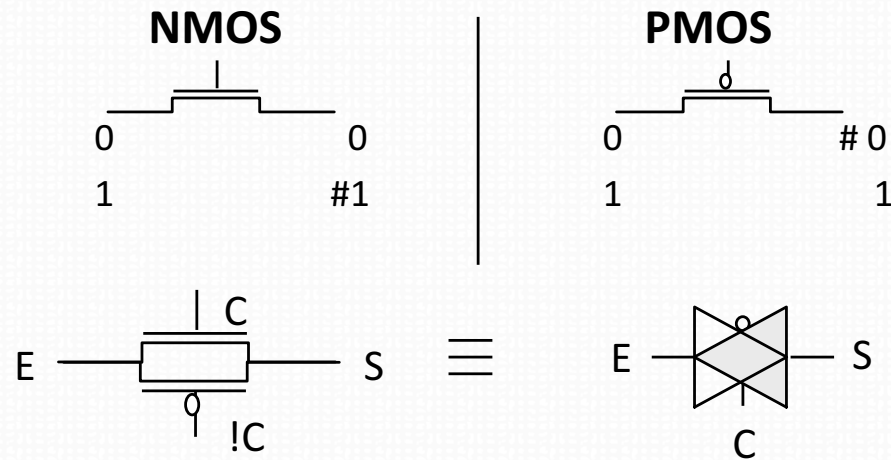
- Example: $S = \overline{A.B + C.D}$
 - AOI (And-Or-Invert) gate



- Multiple-Stage Complex Functions
 - Optimisation of the logic equation
 - Trade-off between speed and area
 - $S3 = A.B.C.D$
 - $S4 = !A.B+A.!B$ (XOR)

Pass-Transistor Logic

- Switch or Transmission Gate



- Example: 2-input multiplexer



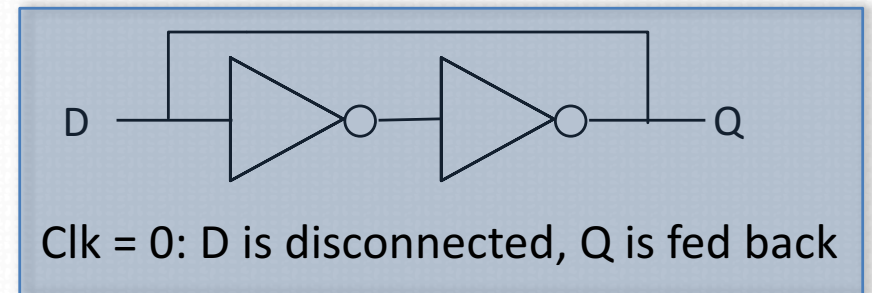
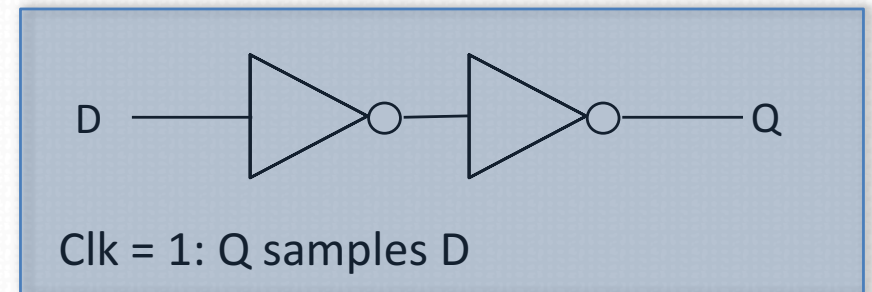
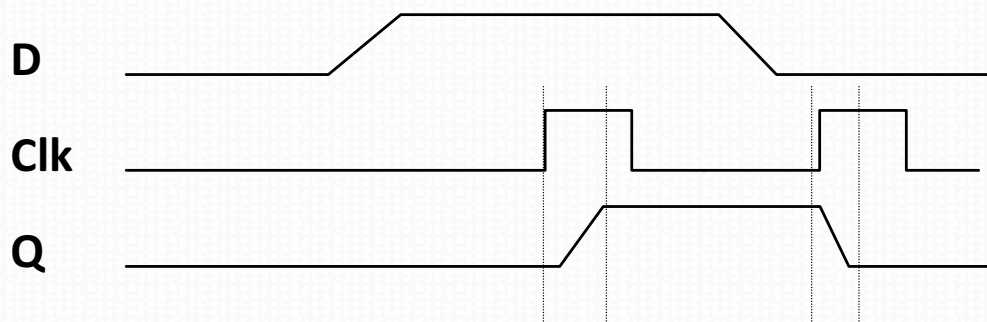
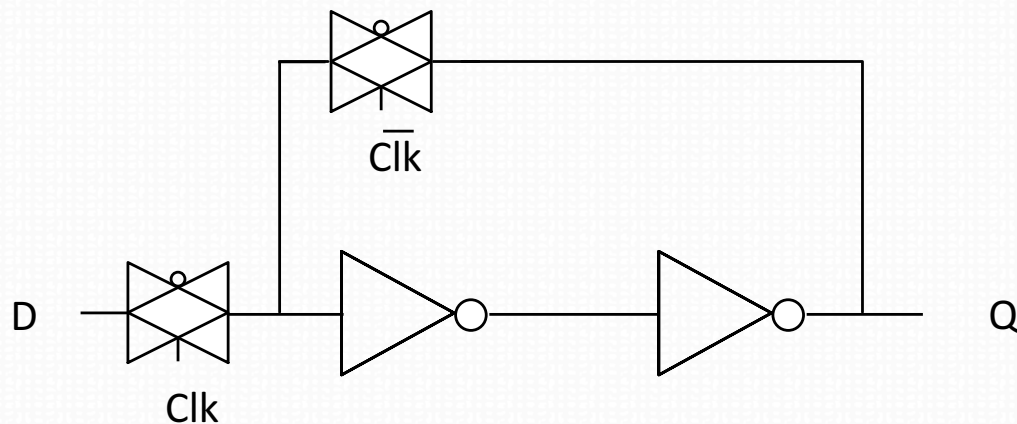
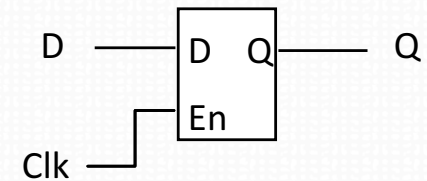
- Example: XOR

Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- **Memory Cells**
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

Elementary Memory Cells

- Static Memory Basic Cell: Latch



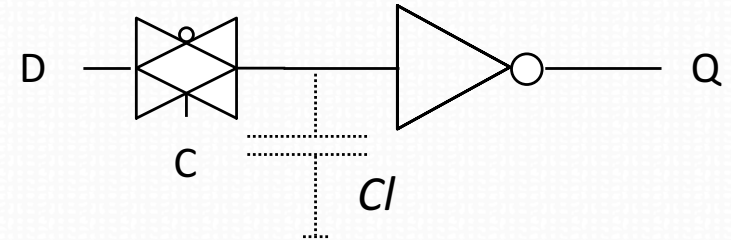
Elementary Memory Cells

- Dynamic Memory Cell

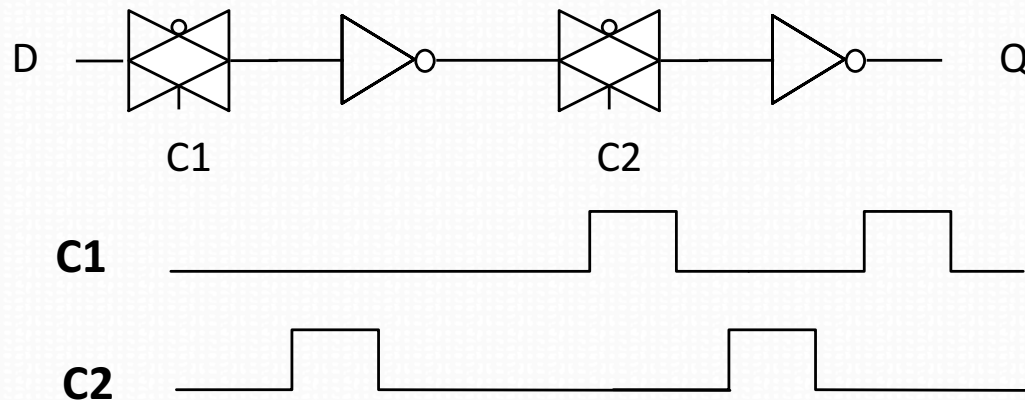
- MOS Capacitor: $C_l = f(\text{area})$

- State ('1') (voltage level) is stored for few ms

- Leakage current
 - Need for refreshing state



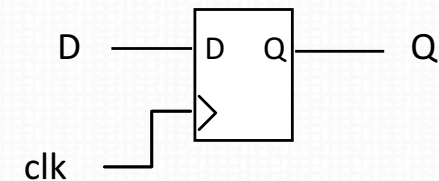
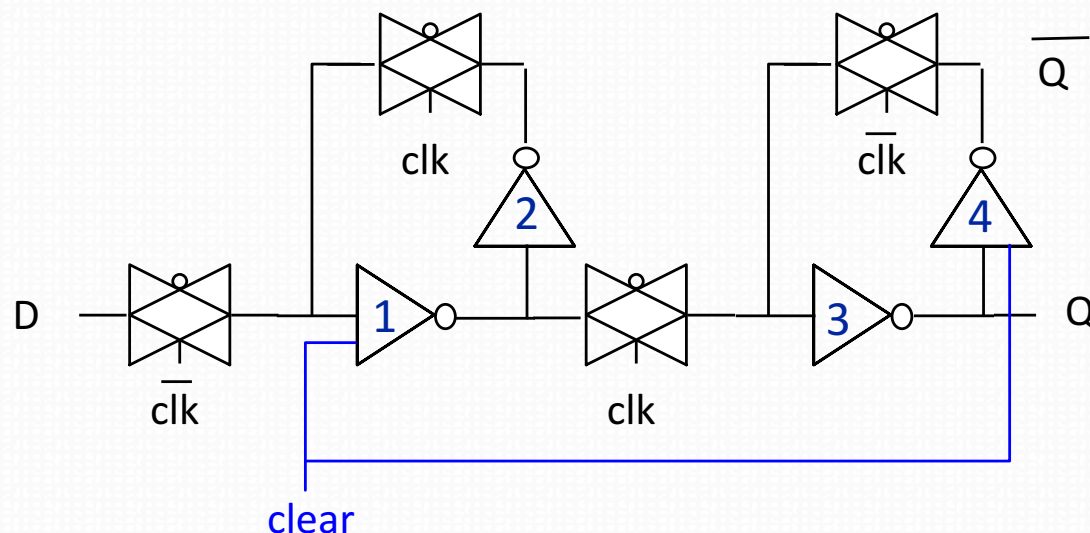
- Ex. Shift Register



Sequential Logic Circuits

- D Flip-Flop (edge-triggered)

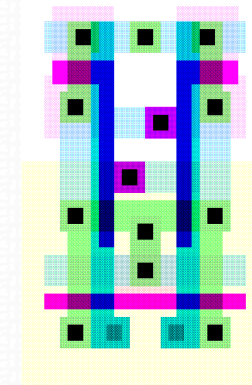
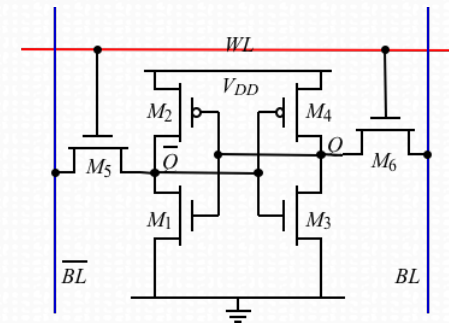
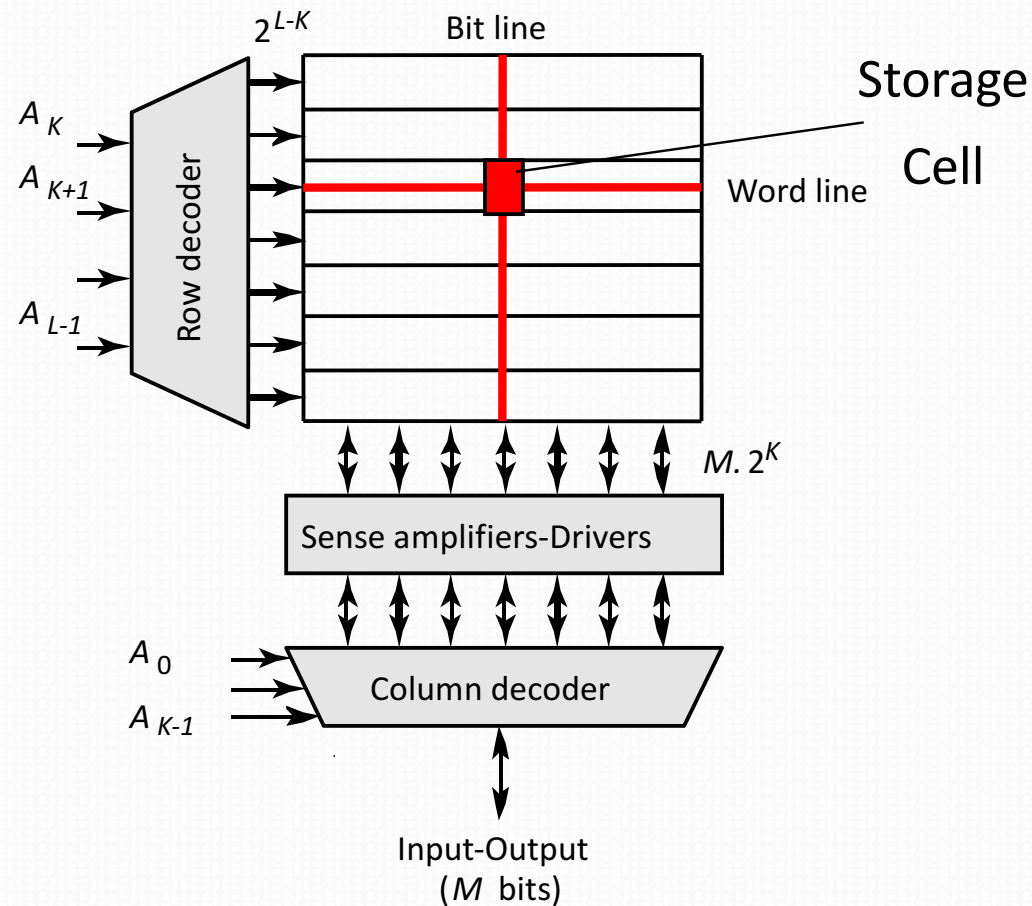
- Two latches in series



- D is sampled in inverter (1) when $\text{clk} = 0$
- Latch (1) and (2) keeps D value when $\text{clk} = 1$ until !D is transferred to second latch (3) and (4)
- Asynchronous clear signal: replace inv. (1) and (4) by NAND

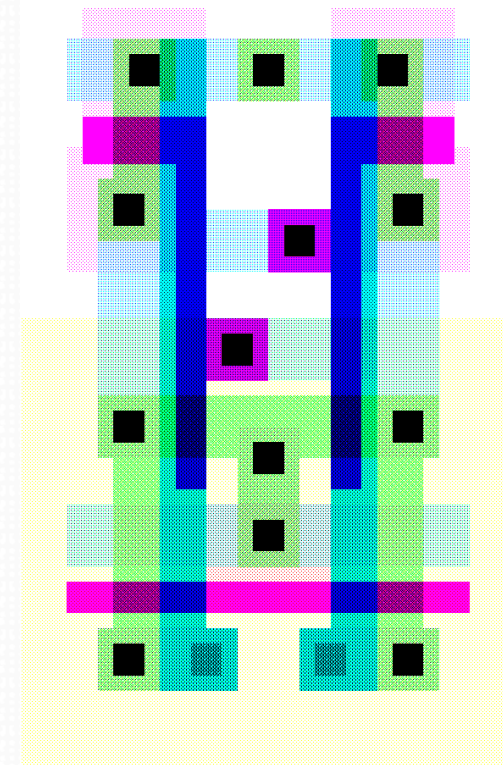
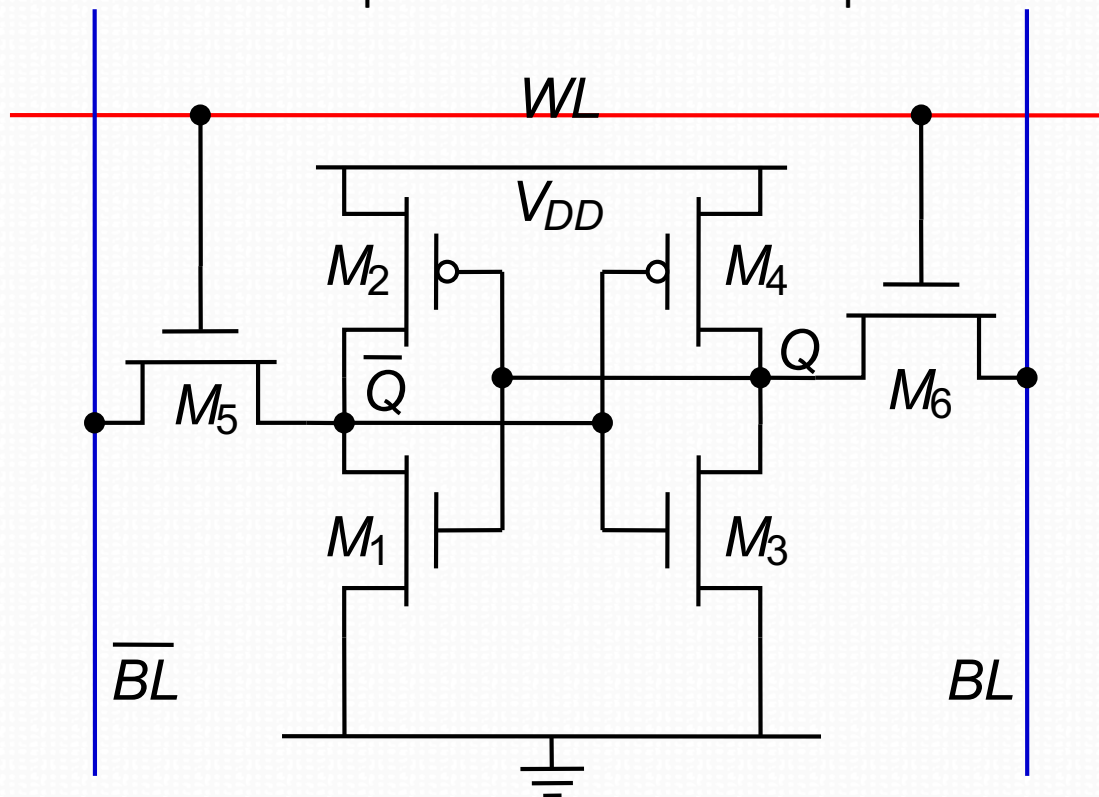
Memory

- L2 Cache contains 4 Millions SRAM cells
 - Row/column of 2000 cells



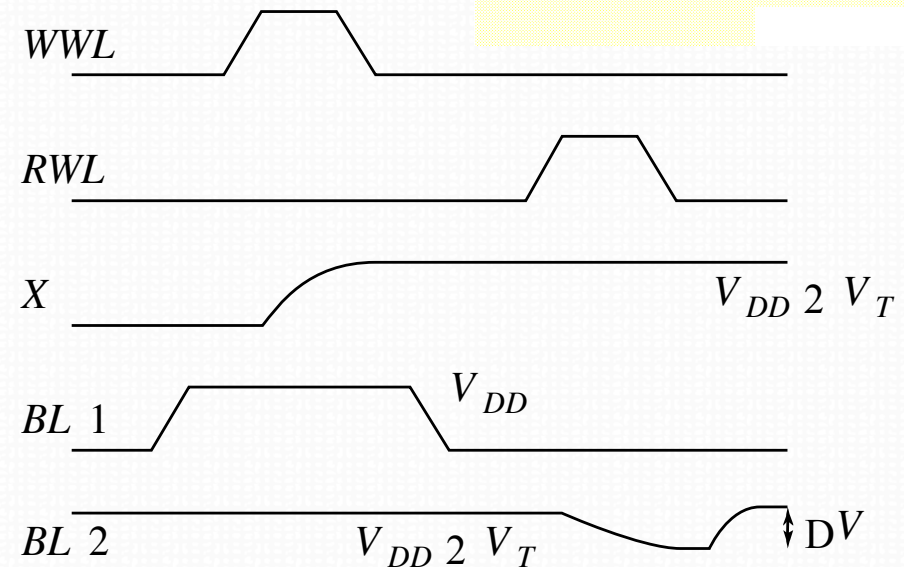
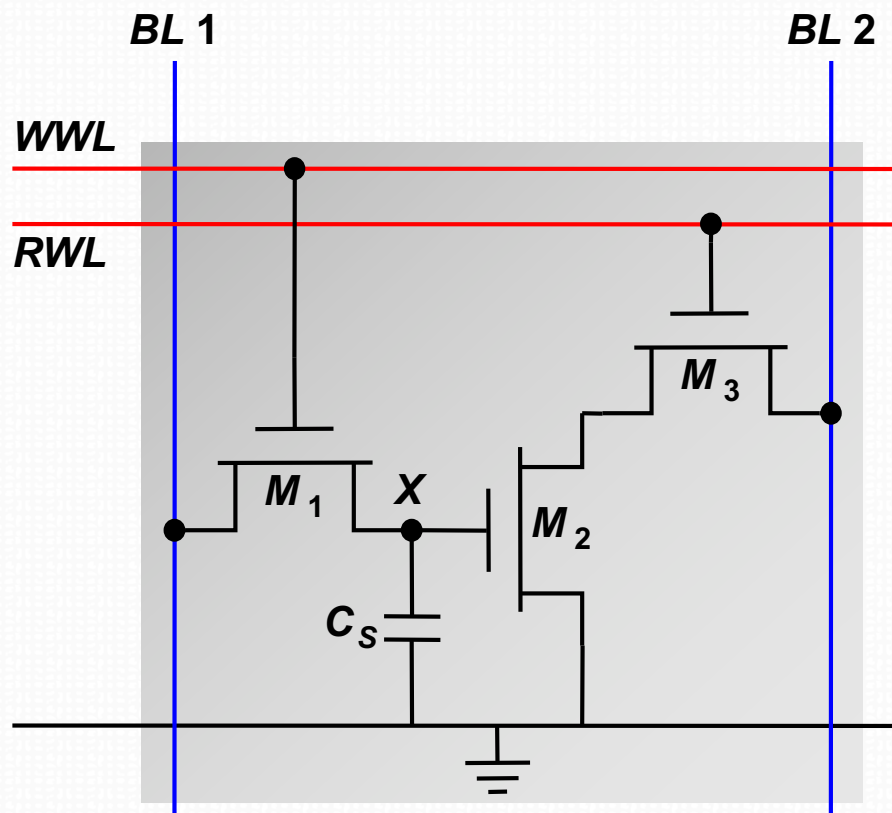
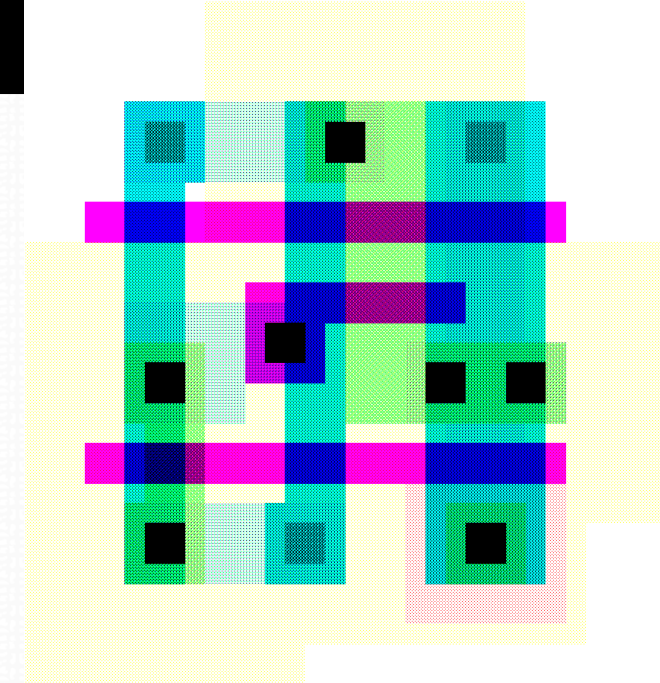
6-Transistor CMOS SRAM Cell

- Latch where WL replaces clock
- Dual-rail bit-lines required to increase noise margin during R/W
- WL selection: $WL[i] = 1$
- Write 0: $BL=0$ et $!BL=1 \Leftrightarrow$ Reset of Latch
- Read: BL et $!BL$ pre-charged to 1, WL selection $\rightarrow BL=Q$ and $!BL=!Q$
 - Sense amplifiers will act as a comparator to increase speed of Latch value to output



3-Transistor DRAM

- 2 lines WL and BL: read and write
- No amplification

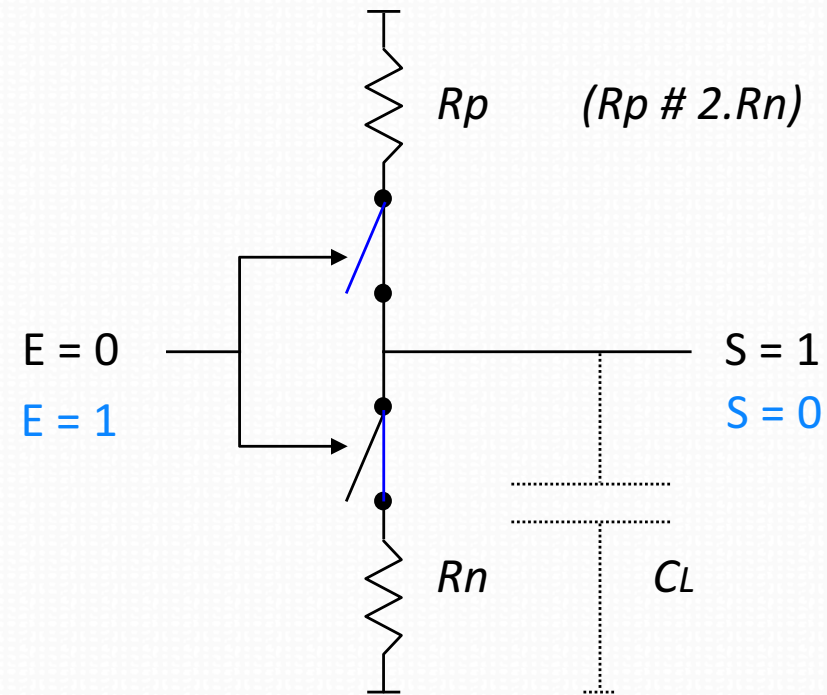
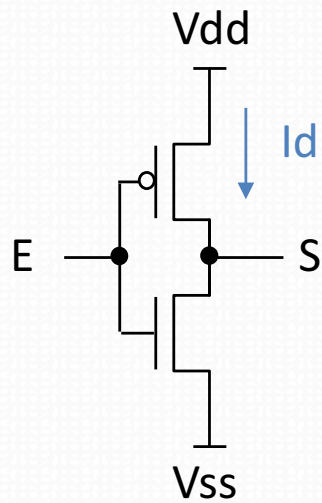


Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- **Delay**
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

Simplified Delay Model

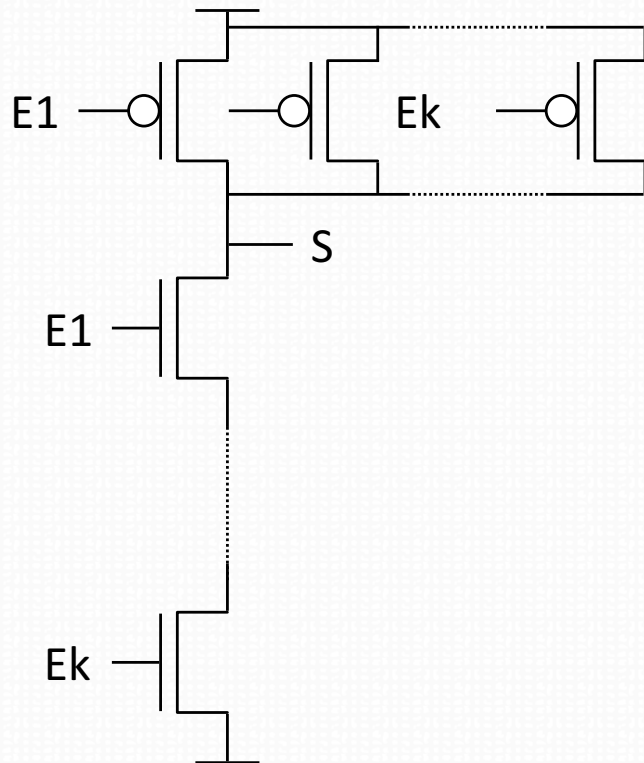
CMOS Inverter



- Rising Output: $t_{plh} = R_p \cdot C_L$
- Falling Output: $t_{phl} = R_n \cdot C_L$

Delay of Complex Gates

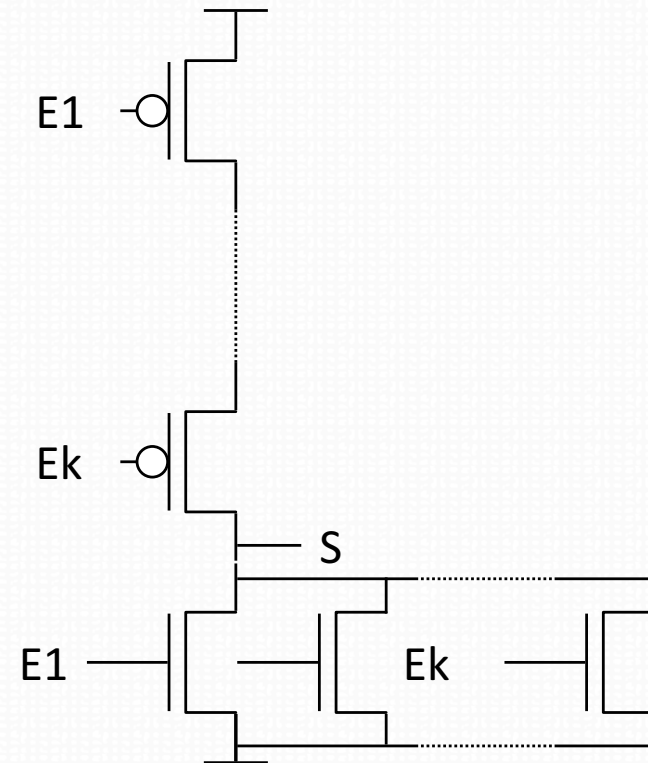
k-input NAND



$$t_{plh} =$$

$$t_{phl} =$$

k-input NOR

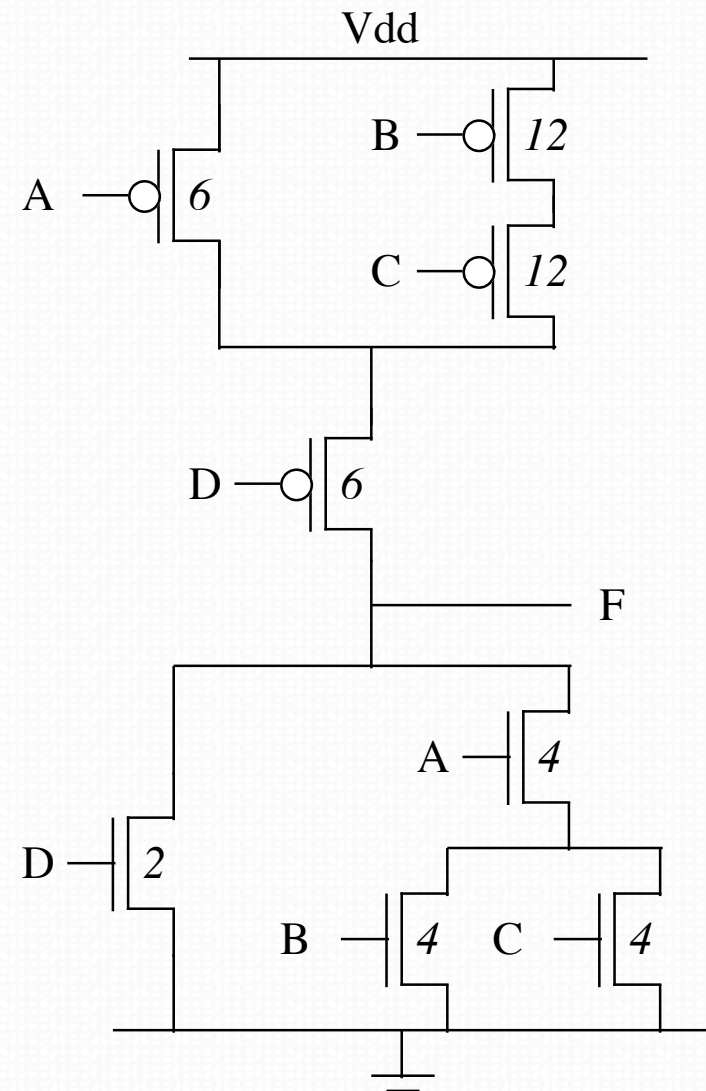


$$t_{plh} =$$

$$t_{phl} =$$

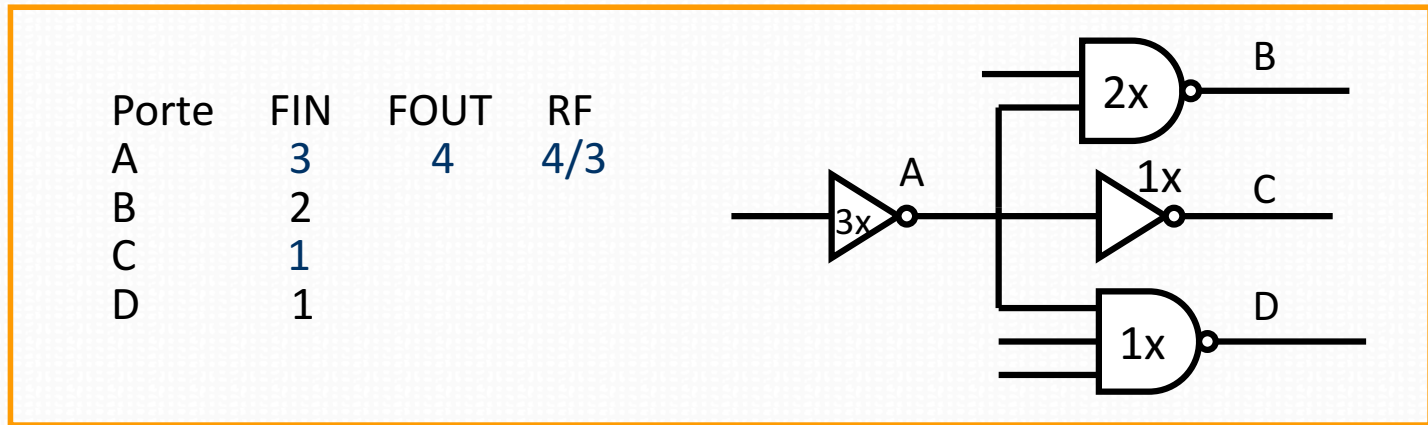
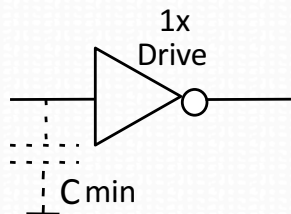
Transistor Sizing

- Complex function
 - $F =$
 - $T_{plh} =$
 - $T_{phl} =$
 - Indicate critical path
 - *Which input values give the best/worst case delay?*



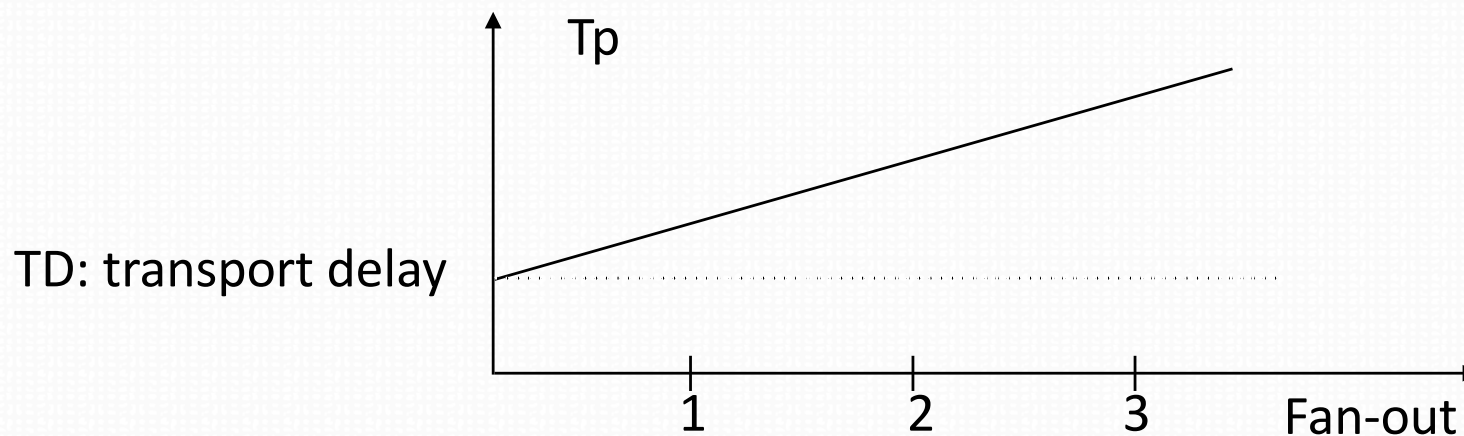
Logic-Level Delay Model

- Fan-In (or Drive): relative to size of transistors
 - Basic inverter is 1x
- Fan-Out: ratio between load capacitance and drive
- Relative Fan-Out (RF): ratio between fan-out and next-stage fan-in



Logic-Level Delay Model

- $T_p = \text{transport delay} + \text{inertial delay} = TD + ID$
- $ID = RF \cdot UD$
- Equivalent to $T_p = R_{DS} [C_{int} + C_{ext}]$



$$T_p = TD + RF \cdot UD$$

UD: unit delay

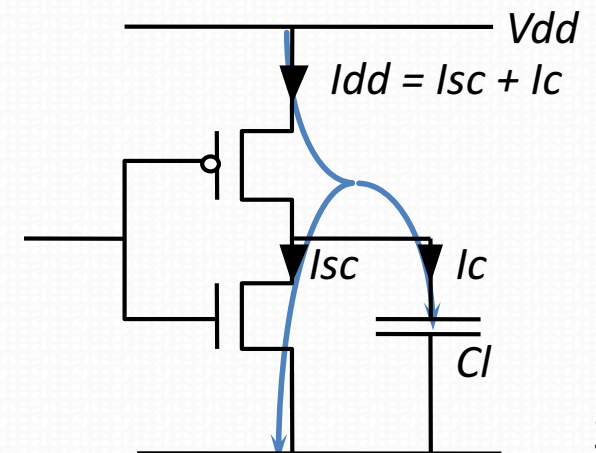
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- **Power Consumption**
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

Power Consumption

- Dynamic power: P_{dyn}
 - Charge and discharge of node capacitance
- Short-circuit power: P_{sc}
 - Short circuit path in logic cells ($V_{dd} \rightarrow V_{ss}$) during commutation
 - Strongly depends on rising time and on V_{th} (NMOS/PMOS)
- Static power: P_s
 - Sub-threshold leakage current (\sim OFF)
 - Source/Drain-Bulk junction leakage

$$P = P_{dyn} + P_{sc} + P_s$$



Dynamic power

- Energy per transition = $C_L V_{dd}^2$
- Power = Energy per transition x rate of transition

$$P_c = C_L V_{dd}^2 f_{0 \rightarrow 1}$$

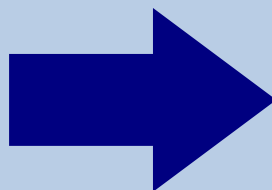
$$P_c = C_L V_{dd}^2 f \text{Prob}_{0 \rightarrow 1}$$

$$P_c = \alpha C_L V_{dd}^2 f$$

$$P_c = \alpha \cdot f \cdot C_L \cdot V_{dd}^2$$

α : activity, C_L : total load capacitance, f : frequency

Power



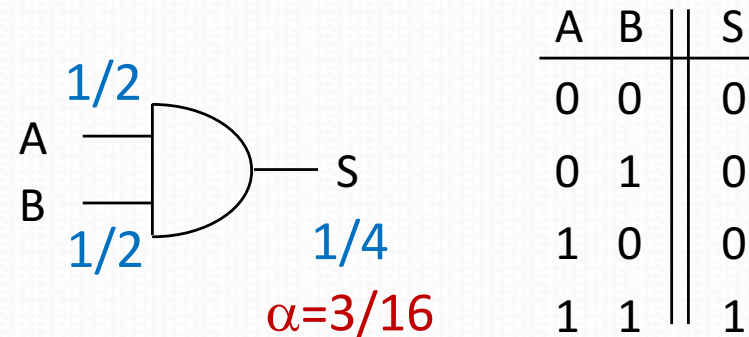
**Data dependant
Activity dependant**

Activity

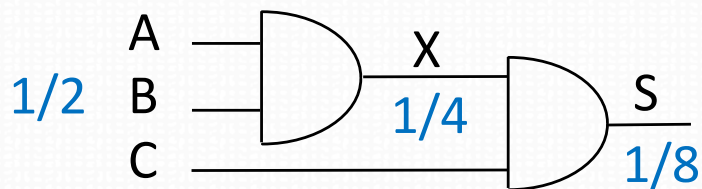
- Activity α_i is the **probability** to have a $0 \rightarrow 1$ transitions at the output of a gate
- Example: AND gate

$$- P_S = P(S=1) = P_A P_B$$

$$- \alpha_i = P_S(1 - P_S)$$

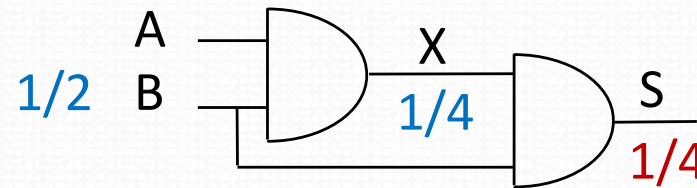
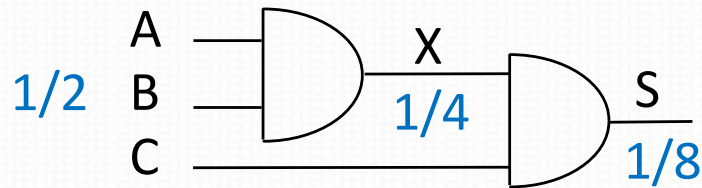


- Activity propagation

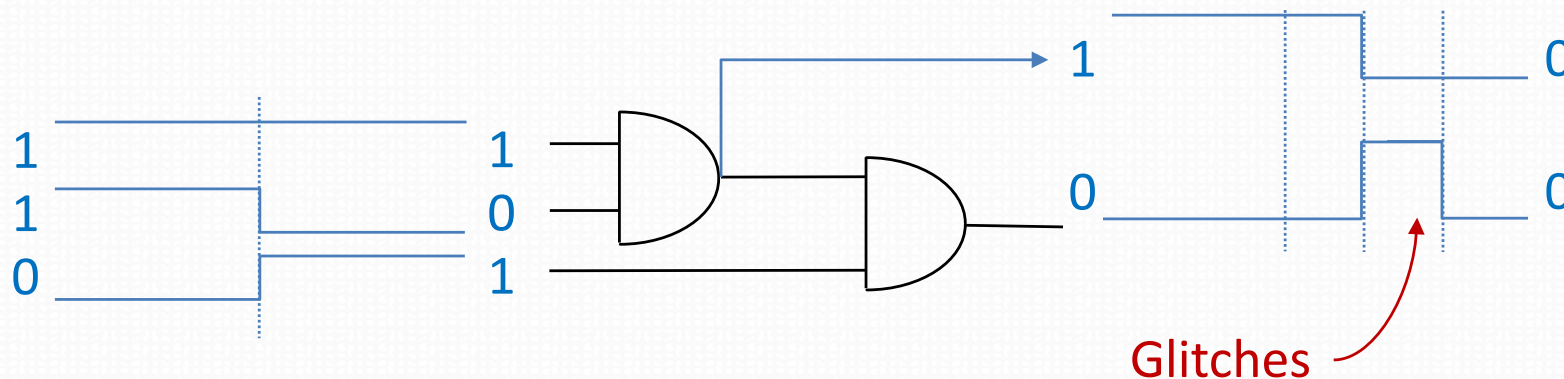


Propagating Activity is not So Simple

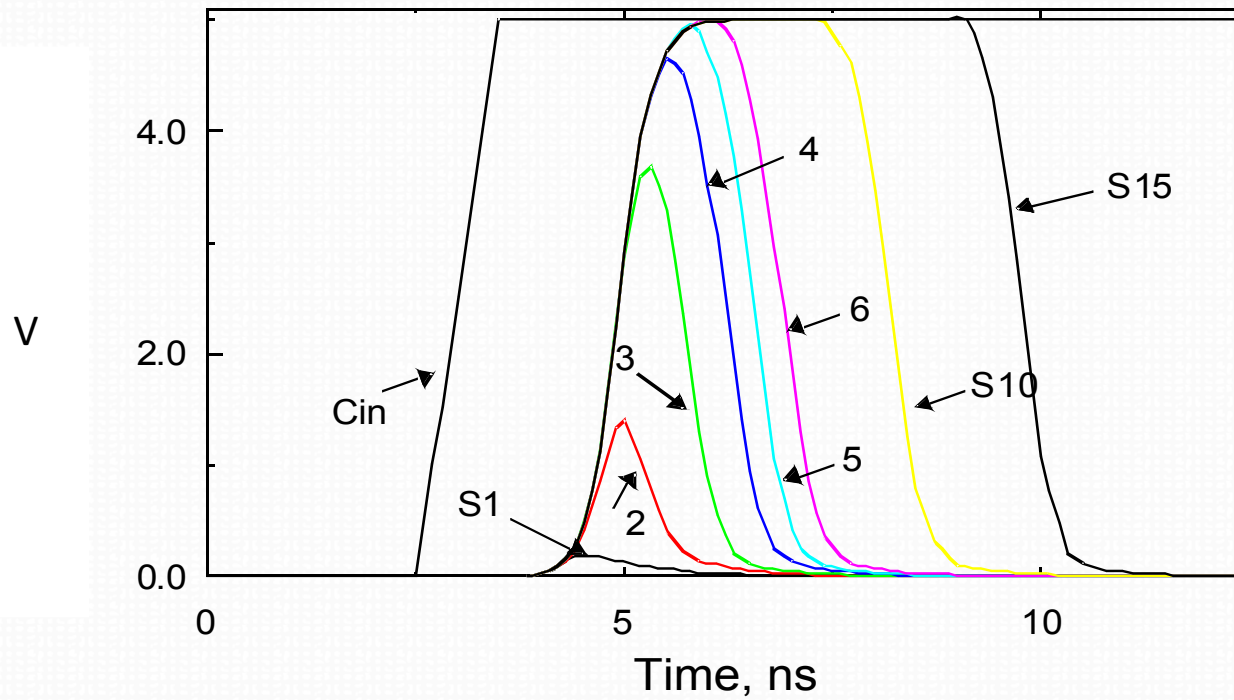
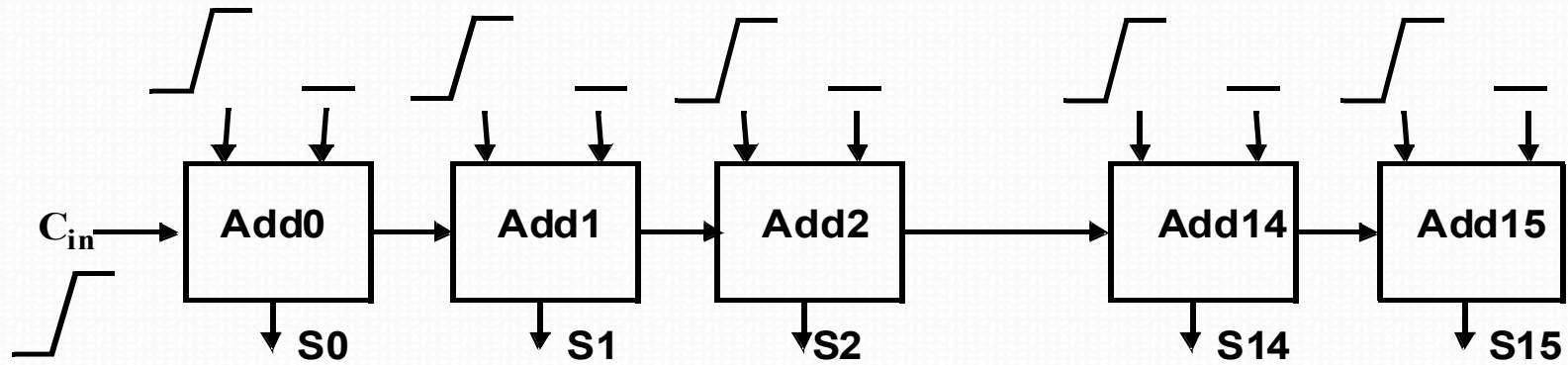
- Conditional probabilities



- Glitches: gate delay
 - Significant in arithmetic

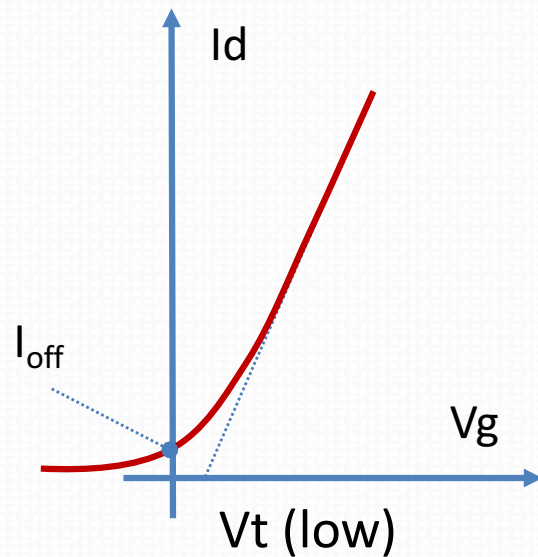


Example: Adder

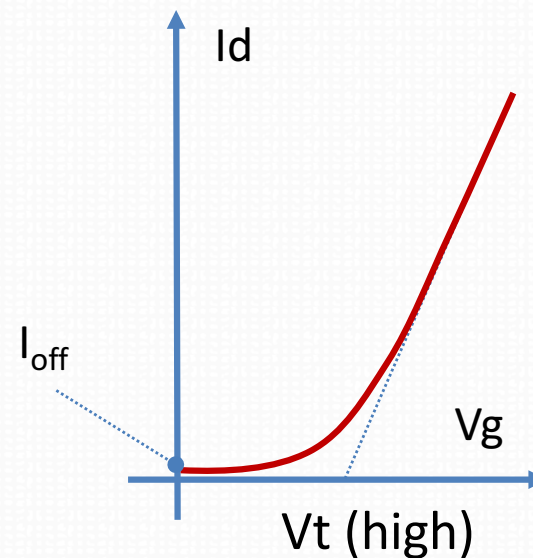


Static Power: Leakage

- High performance



- Low leakage

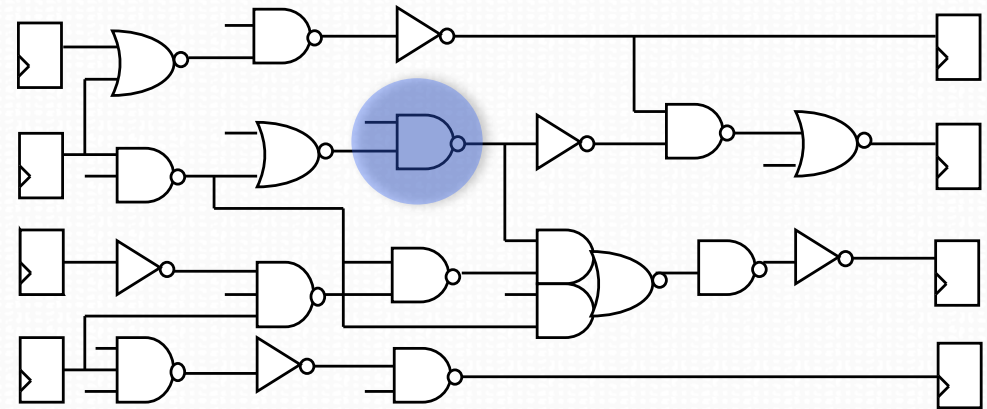
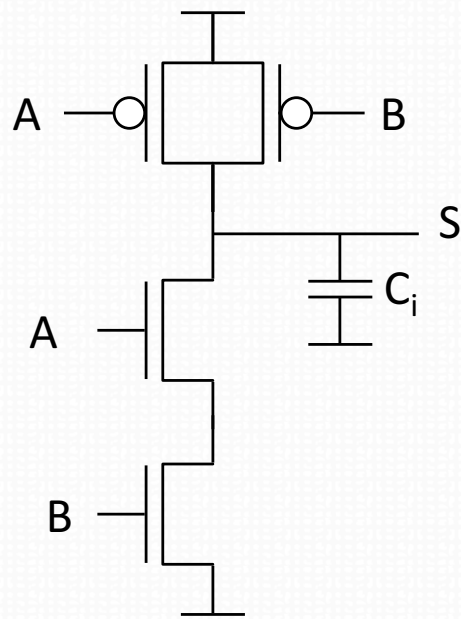


$$P_{stat_i} = N \cdot I_{off} \cdot V_{dd}$$

I_{off} : Sub-threshold Leakage Current

- Exponential in inverse of V_t
- Exponential in temperature
- ~Linear in device count

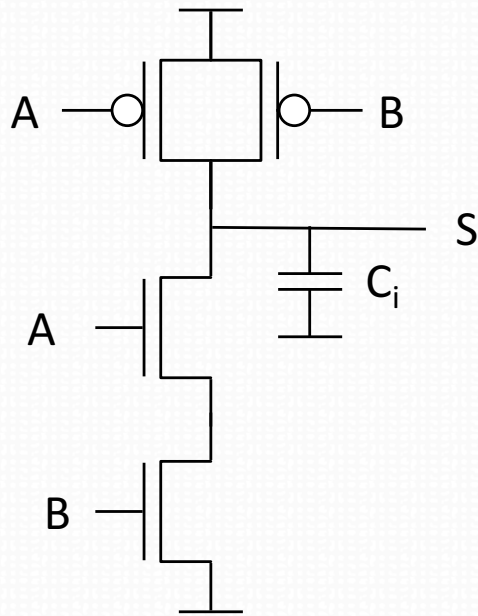
Sum-up: Power at Gate Level



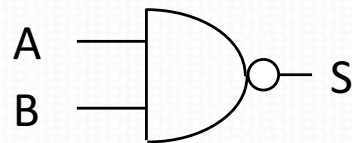
$$P_i = \alpha_i \cdot f_i \cdot C_i \cdot V_{dd}^2 + I_{leak_i} \cdot V_{dd}$$

$$P = \sum_i [\alpha_i \cdot f_i \cdot C_i \cdot V_{dd}^2 + I_{leak_i} \cdot V_{dd}]$$

Power vs. Performance



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



- Delay of a gate

$$\text{Delay} \propto R_{DS} \cdot C_i$$

$$\propto \frac{\text{Relative FanOut}}{V_{dd} - V_t}$$

- Dynamic power

$$P_{\text{dyn}_i} = \alpha_i \cdot f_{\text{clk}} \cdot C_i \cdot V_{dd}^2$$

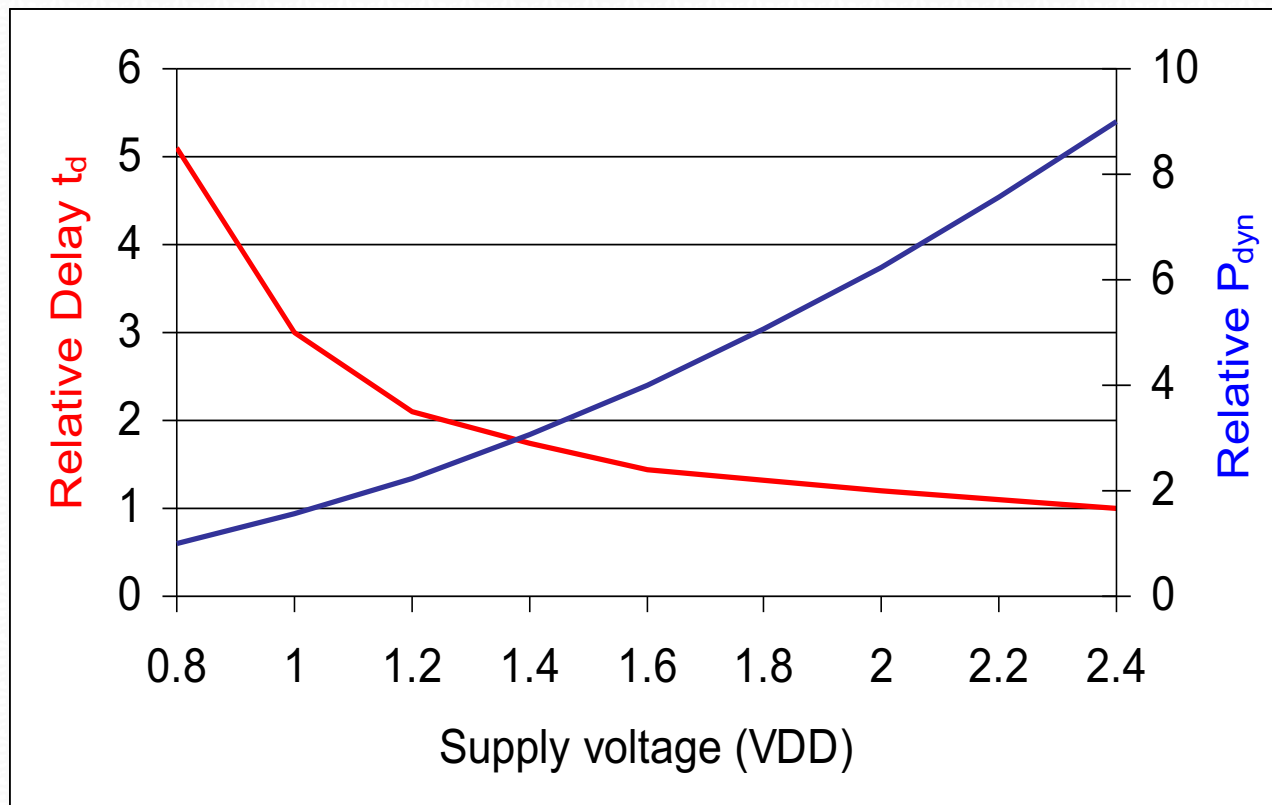
- Leakage power

$$P_{\text{stat}_i} = N \cdot I_{\text{off}} \cdot V_{dd}$$

Dynamic Power vs. Performance

- Decreasing Vdd reduces power **but** increases delay

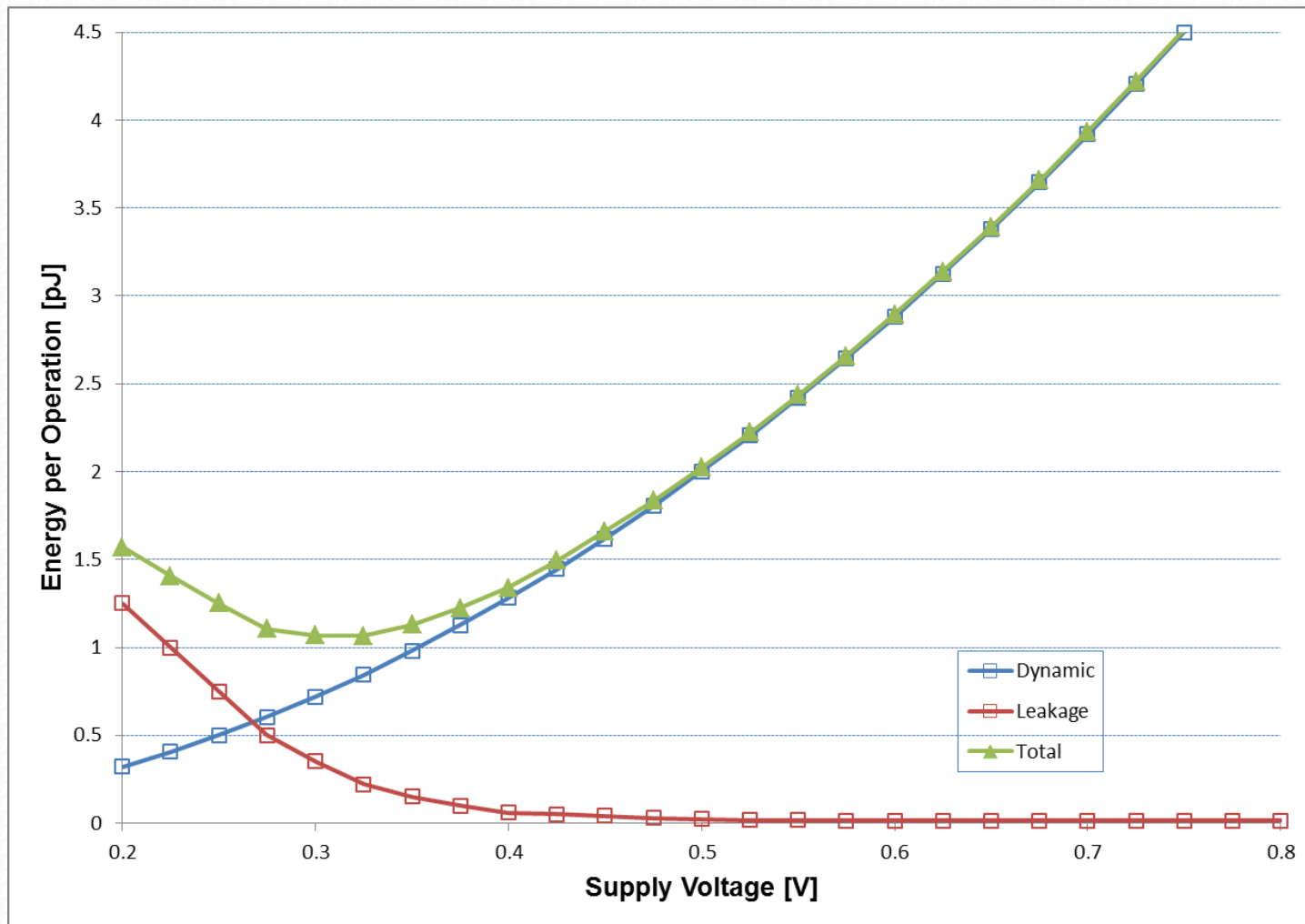
$$P_{\text{dyn}_i} = \alpha_i \cdot f_{\text{clk}} \cdot C_i \cdot V_{\text{dd}}^2$$



$$\text{Delay} \propto \frac{1}{V_{\text{dd}} - V_t}$$

Minimum Energy per Operation

- Putting all together



Conclusion: Power

$$P = \alpha f C_L V_{DD}^2 + V_{DD} I_{peak} (P_{0 \rightarrow 1} + P_{1 \rightarrow 0}) + V_{DD} I_{leak}$$

Dynamic power
($\approx 40\text{-}70\%$ today
and decreasing
relatively)

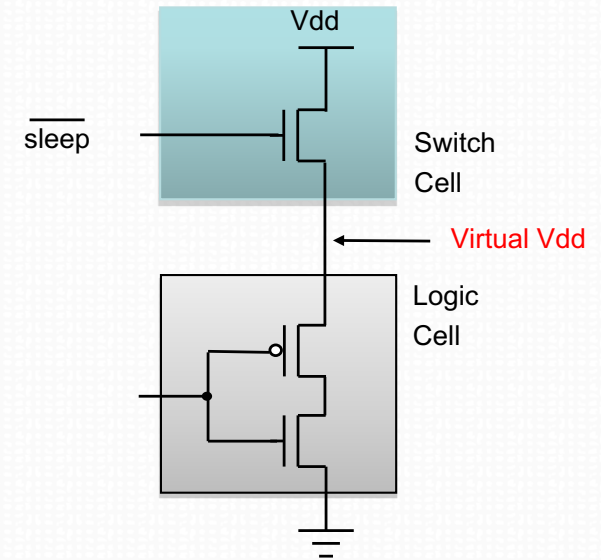
Short-circuit power
($\approx 10\%$ today and
decreasing
absolutely)

Leakage power
($\approx 20\text{-}50\%$ today
and increasing)

$$P = \frac{\text{energy}}{\text{operation}} \times \text{rate} + \text{static power}$$

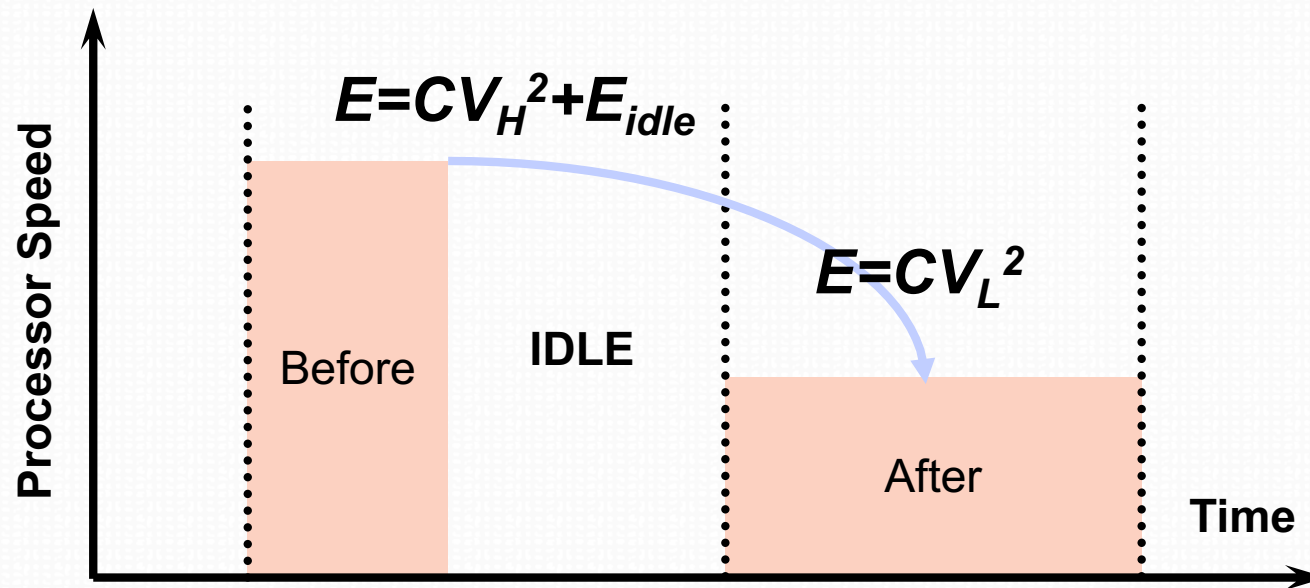
Reducing Power

- Power gating, multi-Vt
- Clock gating
- Vdd scaling
 - Parallel, pipeline
- Activity reduction
 - Pre-computation, correlation, encoding
- Glitch Power Reduction



Dynamic Power Management

- Dynamic Voltage and Frequency Scaling (DVFS)
- Reduce speed (clock freq.) and Vdd depending on processor activity

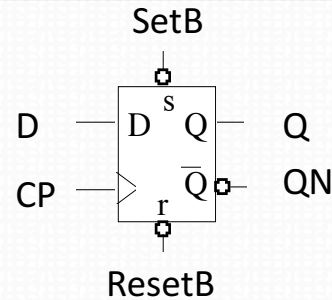


Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

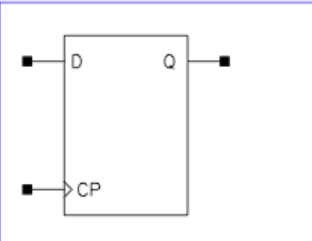
Timing Parameters

- D Flip-Flop
 - Setup Time: T_{setup}
 - Hold Time: T_{hold}
 - Propagation Time: T_p
 - on Clock and Reset



FD1QLL
FD1QLLP
FD1QLLX4

Function: Function = D Flip-Flop with 1 Phase Positive Edge Triggered Clock, Q Output Only



Truth Table

IQ	Q
IQ	IQ

Truth Table

D	CP	IQ	IQ
D	/	-	D
-	-	IQ	IQ

Physical Dimensions

Property	FD1QLL	FD1QLLP	FD1QLLX4
Area(um2)	28.241	28.241	30.258

Capacitance

picoFarads

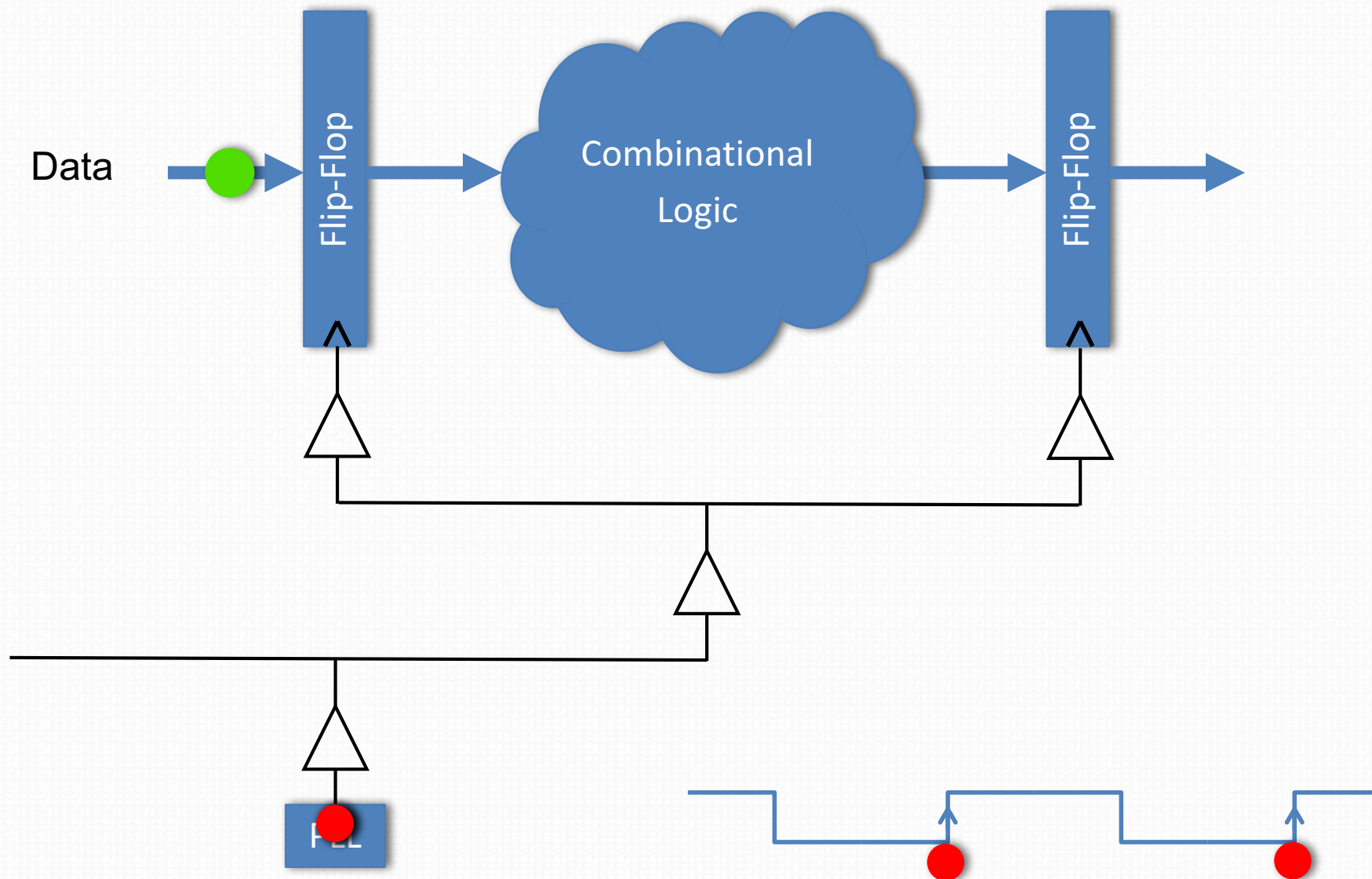
Cell	Property	Best 1.32V -40C	Worst 1.08V 125C	Nominal 1.2V 25C
FD1QLL	CP Input Cap.	0.0032	0.0028	0.0030
FD1QLL	Q Max Load	0.160	0.160	0.160
FD1QLL	D Input Cap.	0.0023	0.0020	0.0021
FD1QLLP	Q Max Load	0.320	0.320	0.320
FD1QLLP	D Input Cap.	0.0022	0.0019	0.0021
FD1QLLP	CP Input Cap.	0.0032	0.0027	0.0029
FD1QLLX4	CP Input Cap.	0.0032	0.0027	0.0029
FD1QLLX4	Q Max Load	0.640	0.640	0.640
FD1QLLX4	D Input Cap.	0.0022	0.0019	0.0020

Propagation Delay

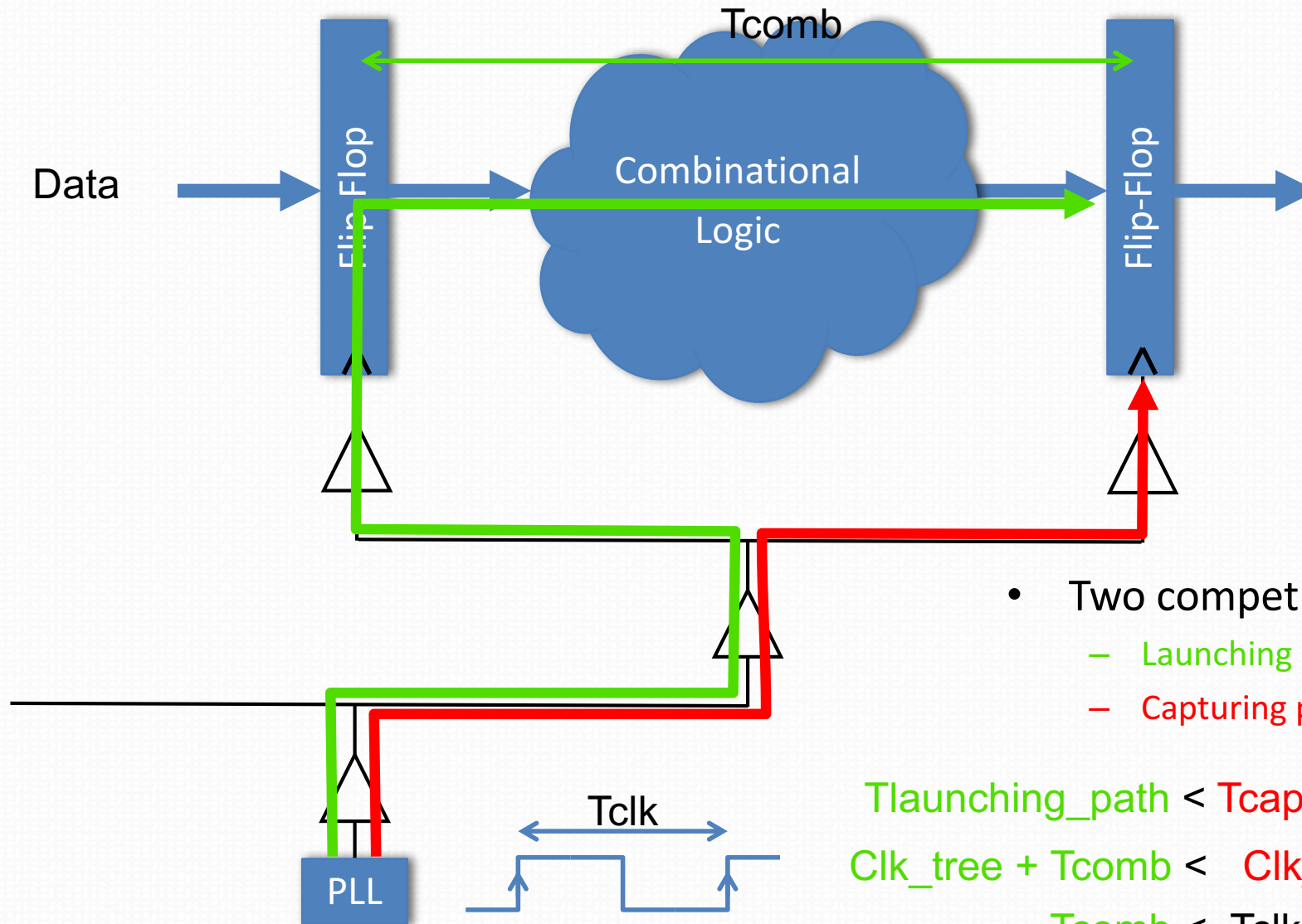
nanoSeconds, as a function of C (load in pF) and Tr (input transition time in nS)

Cell	Path	Event	Best 1.32V -40C	Worst 1.08V 125C	Nominal 1.2V 25C
FD1QLL	CP-Q	CP_Q (fall)	0.082 + 0.119*Tr + 1.221°C	0.195 + 0.179*Tr + 2.777°C	0.125 + 0.148*Tr + 1.731°C
FD1QLL	CP-Q	CP_Q (rise)	0.075 + 0.118*Tr + 1.672°C	0.178 + 0.180*Tr + 3.473°C	0.113 + 0.148*Tr + 2.408°C
FD1QLLP	CP-Q	CP_Q (fall)	0.087 + 0.121*Tr + 0.644°C	0.205 + 0.182*Tr + 1.428°C	0.133 + 0.150*Tr + 0.903°C
FD1QLLP	CP-Q	CP_Q (rise)	0.079 + 0.120*Tr + 0.836°C	0.189 + 0.182*Tr + 1.727°C	0.120 + 0.150*Tr + 1.198°C
FD1QLLX4	CP-Q	CP_Q (fall)	0.111 + 0.122*Tr + 0.342°C	0.267 + 0.183*Tr + 0.760°C	0.173 + 0.152*Tr + 0.482°C
FD1QLLX4	CP-Q	CP_Q (rise)	0.093 + 0.121*Tr + 0.425°C	0.224 + 0.184*Tr + 0.891°C	0.141 + 0.151*Tr + 0.612°C

Synchronous Circuits



Synchronous Circuits



- Two competing paths
 - Launching path
 - Capturing path

$$T_{\text{launching_path}} < T_{\text{capturing_path}} + T_{\text{clk}}$$

$$Clk_tree + T_{\text{comb}} < Clk_tree + T_{\text{clk}}$$

$$T_{\text{comb}} < T_{\text{clk}} \quad (\text{no clk skew}) \quad 50$$

Critical Path

- All circuits have a **maximal frequency**, which is given by finding its **critical path**
 - Data must be stable when sampled by the clock
- T_{cp} : critical path delay of the logic

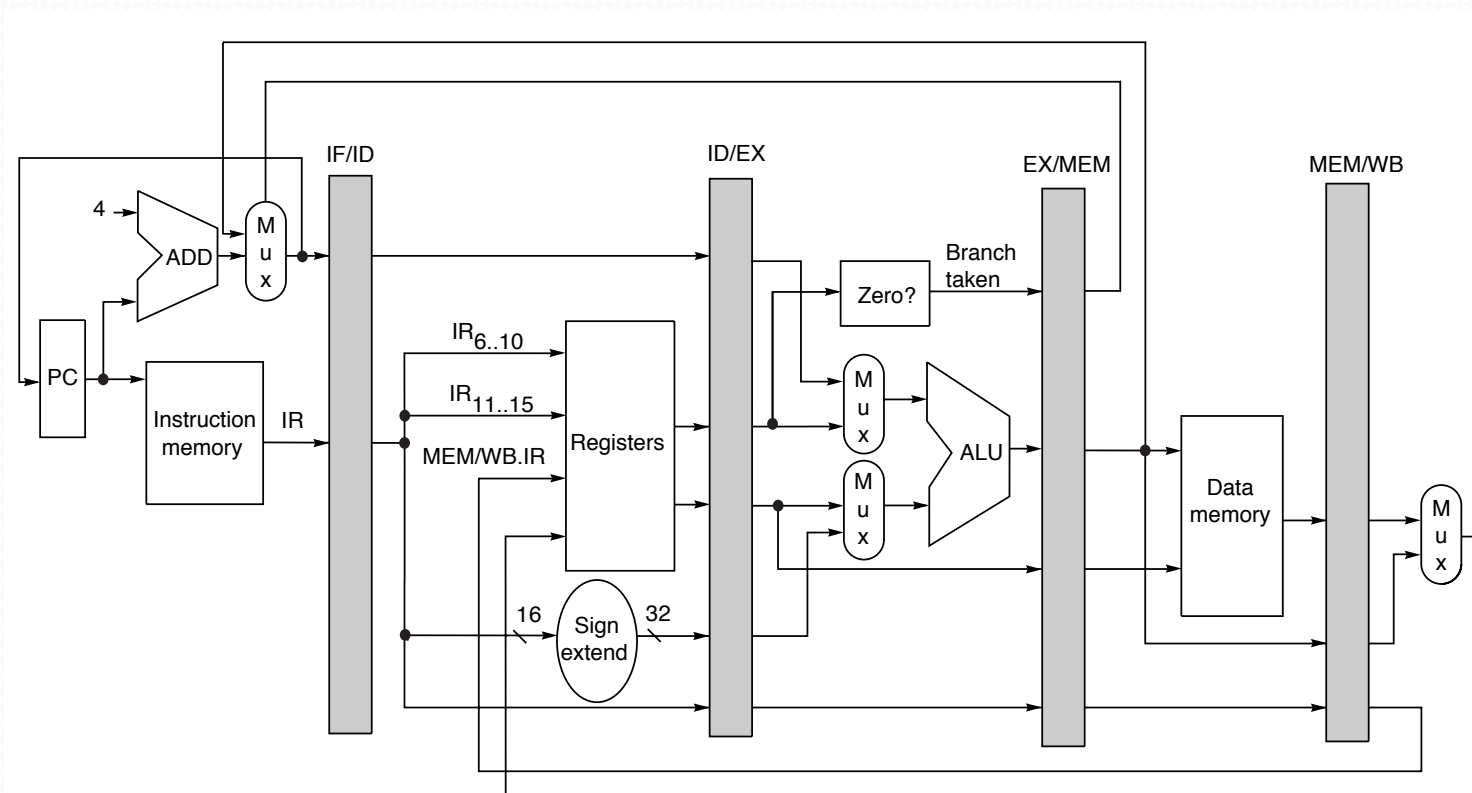
$$T_{cp} = \underset{\forall i}{MAX}(D_i), \text{ with } D_i \text{ Delay of path } i$$

- Maximal Frequency

$$F_{clk_{max}} = \frac{1}{T_{cp} + T_p + T_{setup}}$$

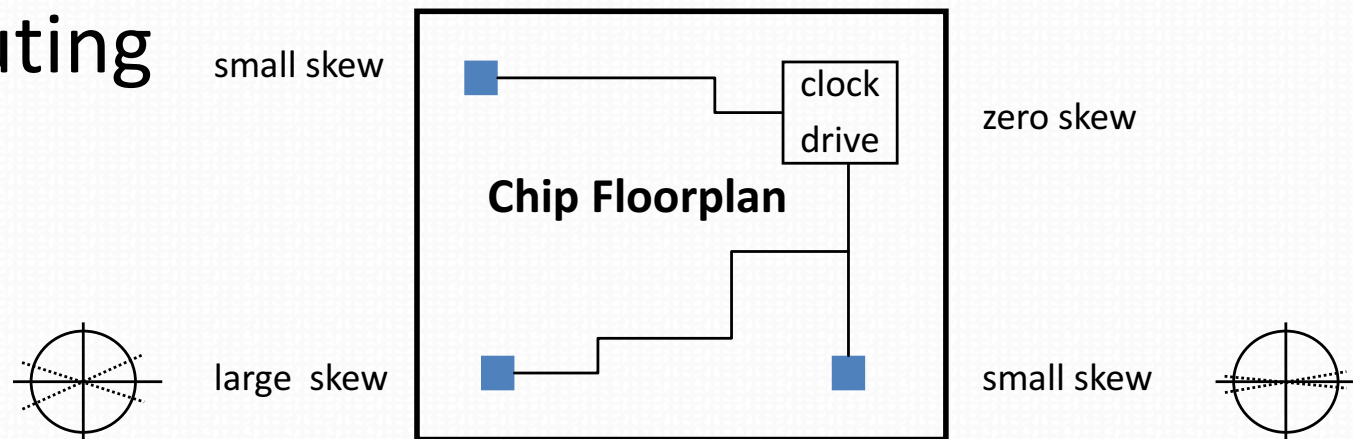
Critical Path in Processor Pipelines

- A typical (yet simple) processor pipeline



Clock Skew

- Every FF receives the clock edge at a different time
- Clock routing



- Light Speed: $300\mu\text{m}/\text{ps}$
- Diagonal : 30 mm (21mm side)
- 100 ps
- 1 clock cycle @ 10GHz
- 5-10 clock cycles @ 1-2GHz

Clock Skew: problems

- Skew δ can be negative or positive

- Reduction of maximal frequency

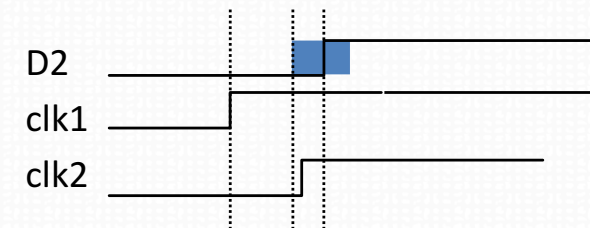
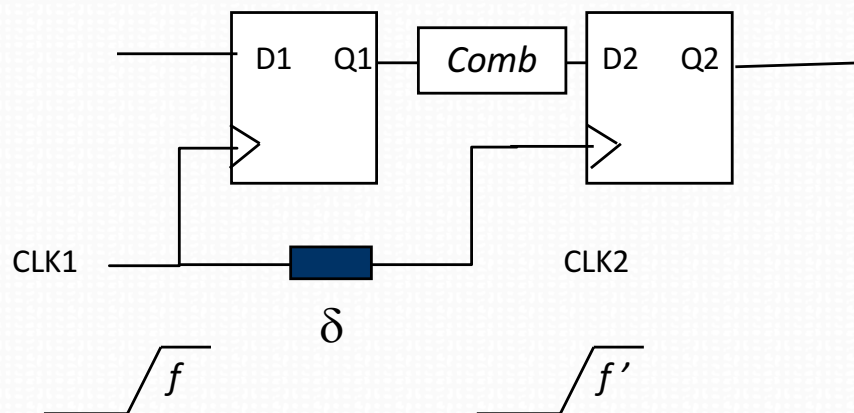
$$F_{e_{\max}} = \frac{1}{T_{cc} + T_p + T_{\text{setup}} + \delta}$$

- Maximal skew for circuit operation

- Worst case is when receiving edge arrives late

- Edge f' of CLK2 should not violate hold time of D2

- Race between data and clock



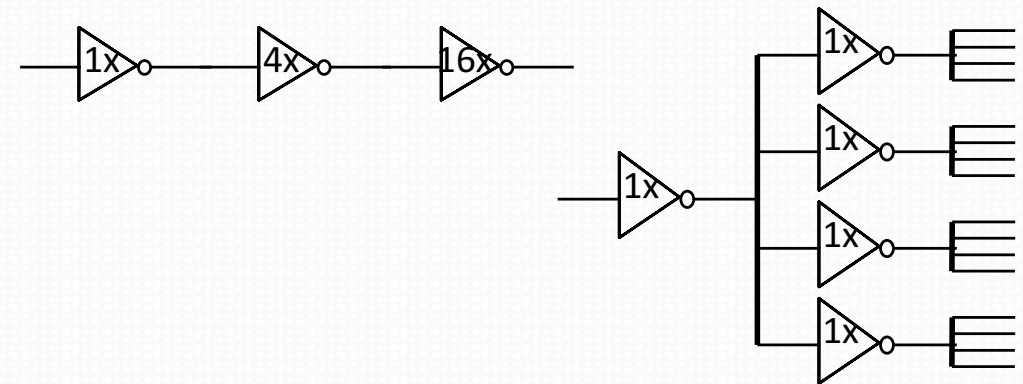
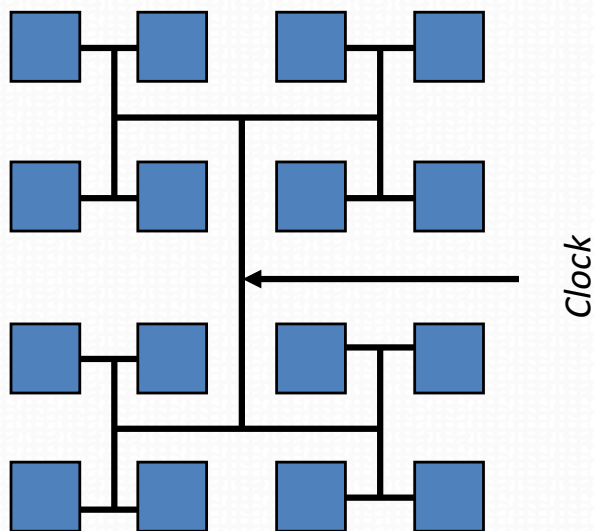
$$t_{p1} + \text{MIN}(t_{\text{comb}}) > \delta + t_{\text{hold}}$$

$$\delta < t_{p1} + \text{MIN}(t_{\text{comb}}) - t_{\text{hold}}$$

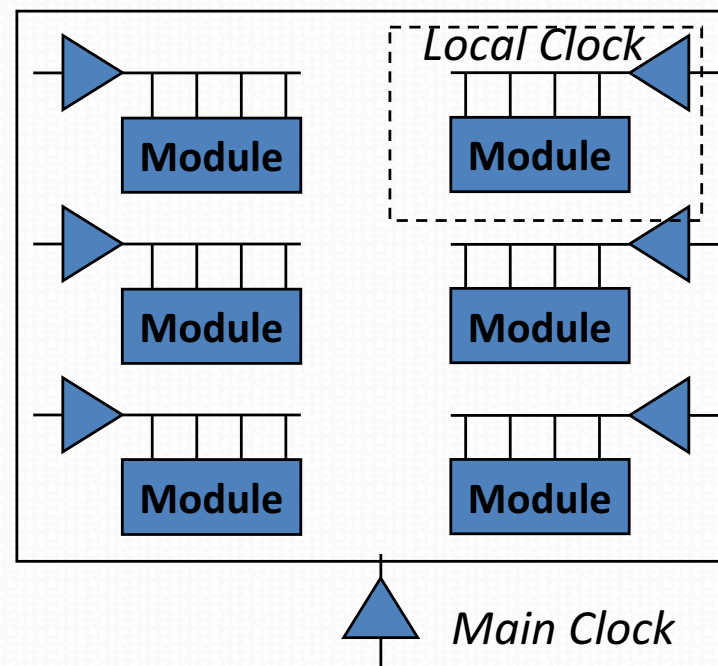
Clock Distribution

- Geometric buffering
- Tree-based

H-tree: constant skew in each block with equivalent number of flip-flops



Buffering: local reduction of skew

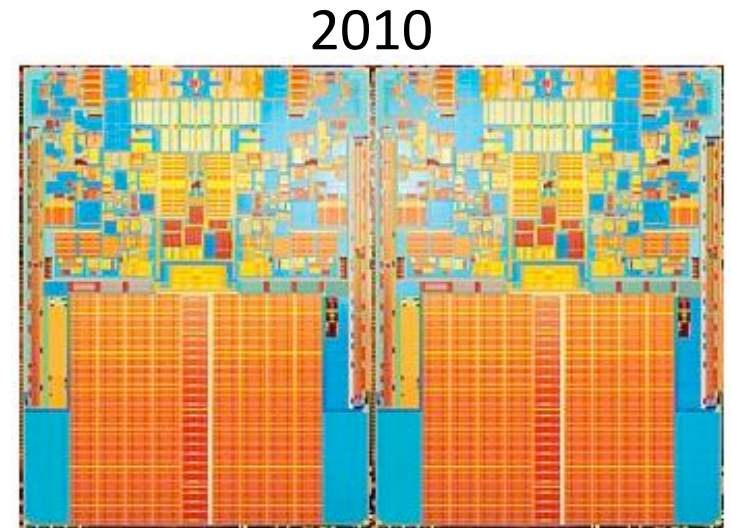
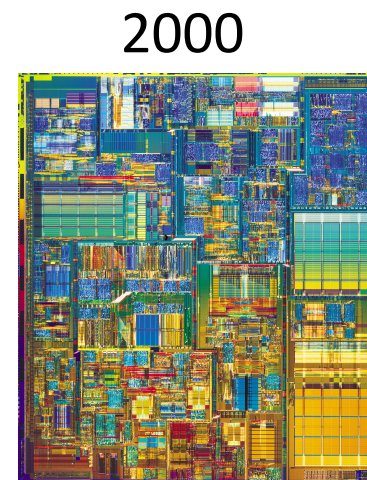
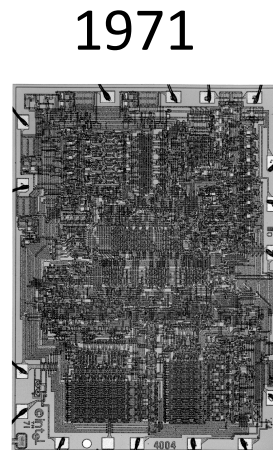
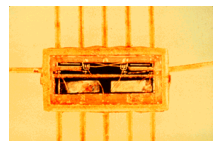


Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- Multicore: power and utilization walls

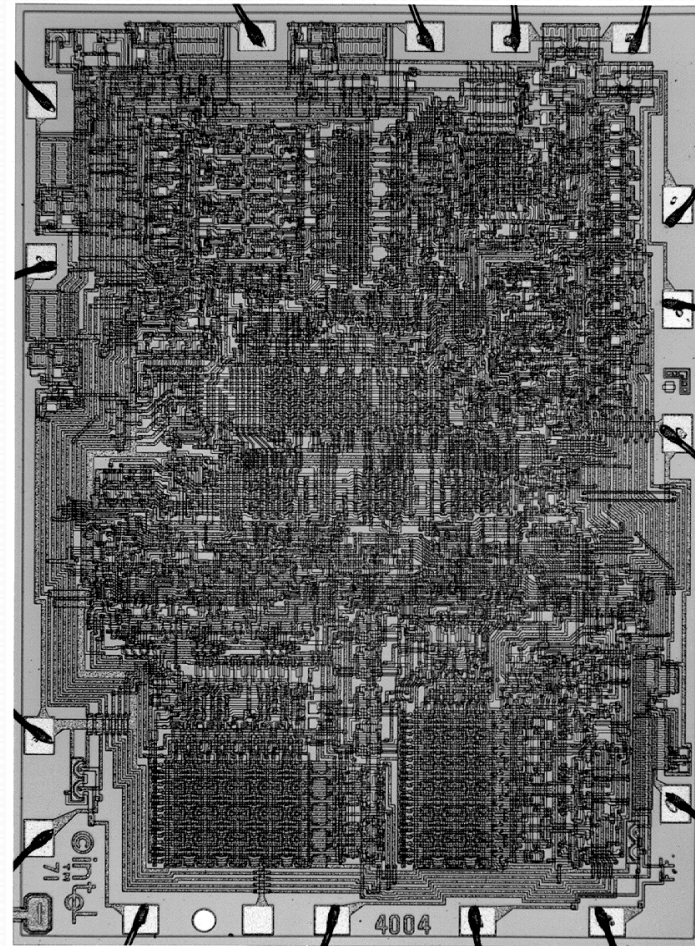
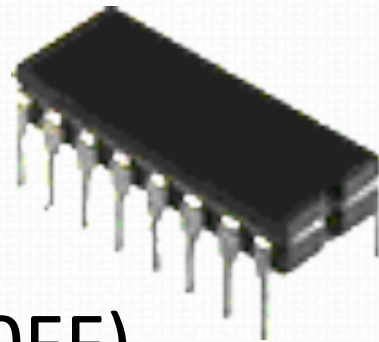
Technology Evolution and Scaling

- 10 μm – 1971
- 6 μm – 1974
- 3 μm – 1977
- 1.5 μm – 1982
- 1 μm – 1985
- 800 nm – 1989
- 600 nm – 1994
- 350 nm – 1995
- 250 nm – 1997
- 180 nm – 1999
- 130 nm – 2001
- 90 nm – 2004
- 65 nm – 2006
- 45 nm – 2008
- 32 nm – 2010
- 22 nm – 2012
- 14 nm – 2014
- 10 nm – 2017
- 7 nm – ~2018
- 5 nm – ~2020
- and then?



The First Microprocessor

- Intel 4004
- 1971
- 400 kHz
- 4 bits
- 200 US\$ (1200FF)
- 0,06 MOPS
- 10 microns
- 2300 transistors
- 640 addressable bytes



intel.

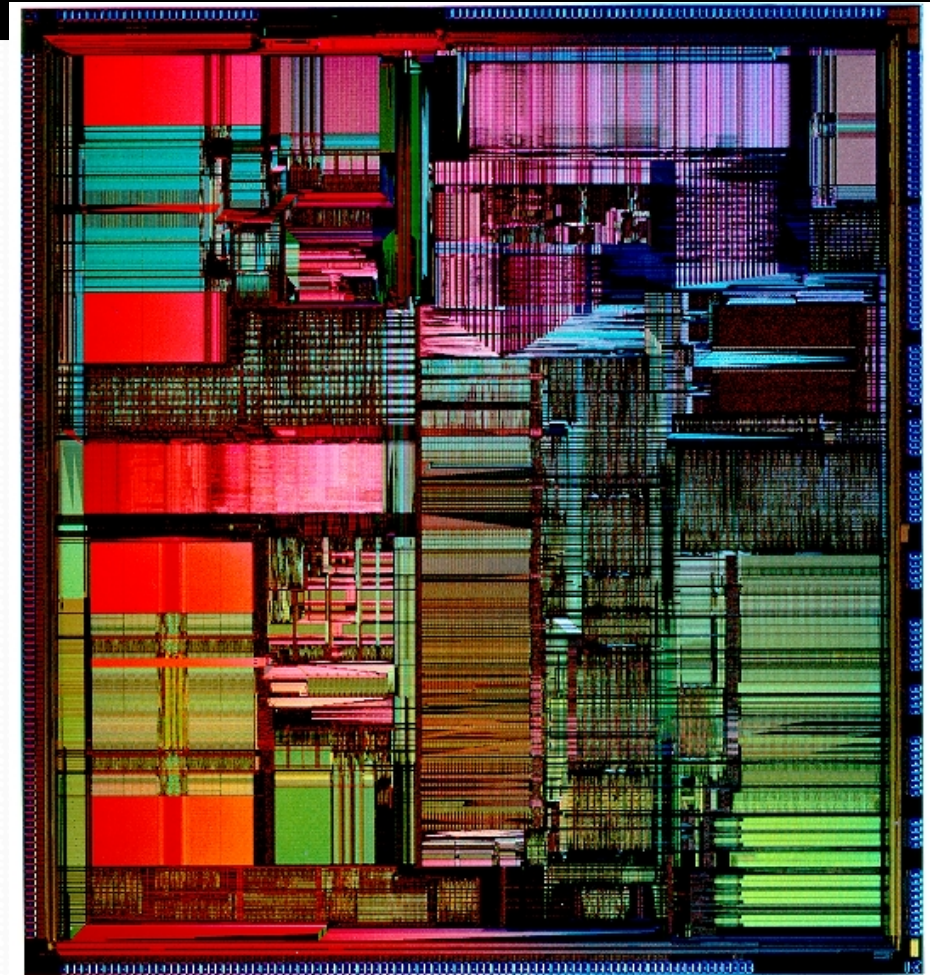
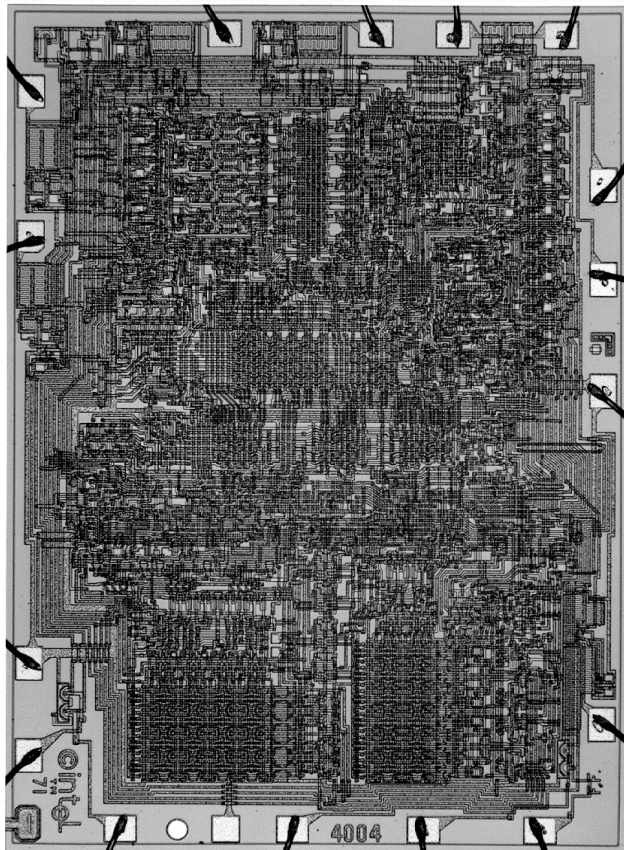
Microprocessor Gallery

INTEL 4004 (1971)

4-bit data

2300 transistors, 10 microns

0,06 MOPS, 108 kHz



INTEL Pentium II (1996)

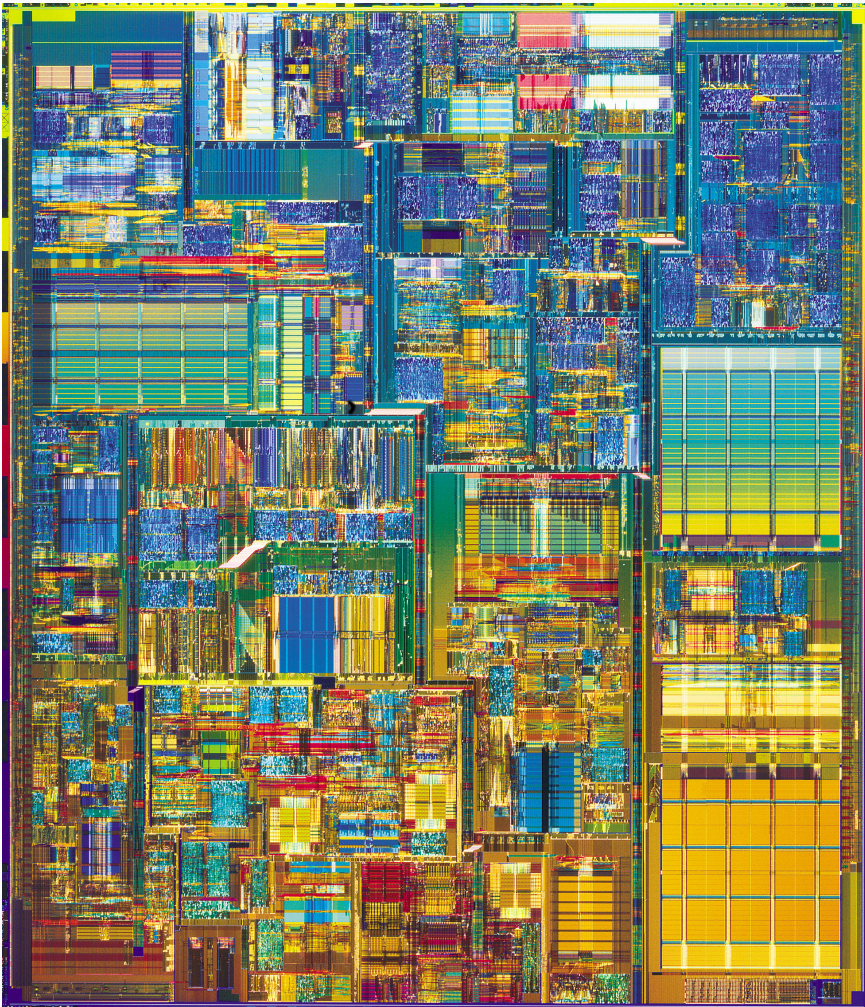
32-bit data

5.5M transistors, 0.35μ , 2 cm^2

200 MHz, 200 MOPS, 3.3V, 35W

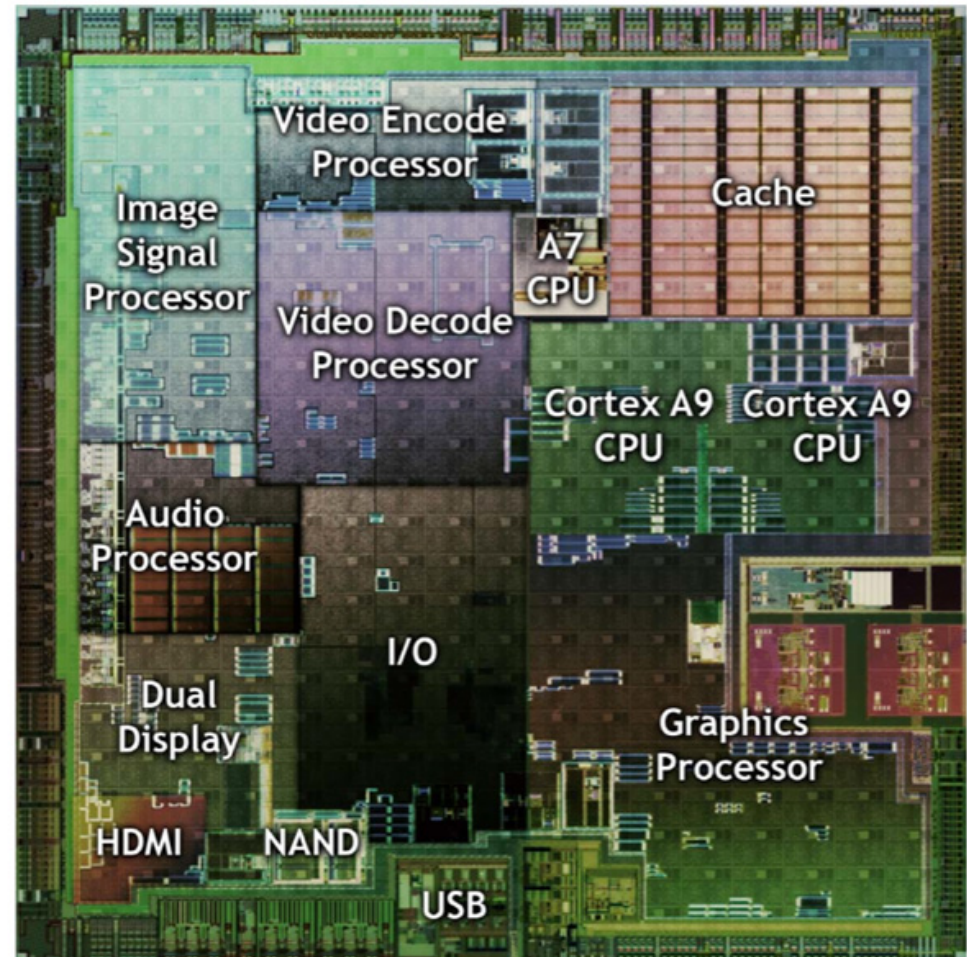
Microprocessor Gallery

2000 : Intel® Pentium® 4 Processor
42M Tr, 0.18um, 1.5GHz – 3.6GHz



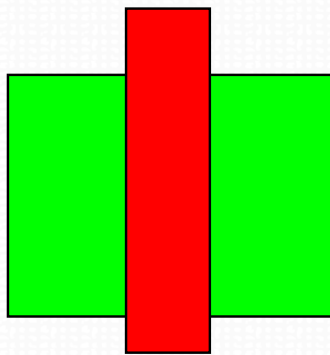
2010: NVIDIA Tegra 2 SoC

260M Tr, 40nm, 49mm², 2 Cortex A9 1Ghz,
300 Mhz (rest of the chip)

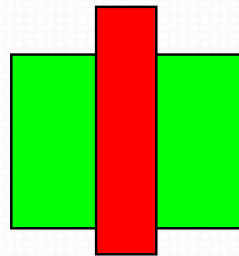


Technology Scaling

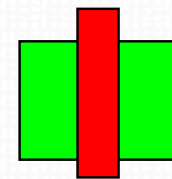
- Scaling factor: s
- Between two successive generations: $s \approx 0.7$



250 nm



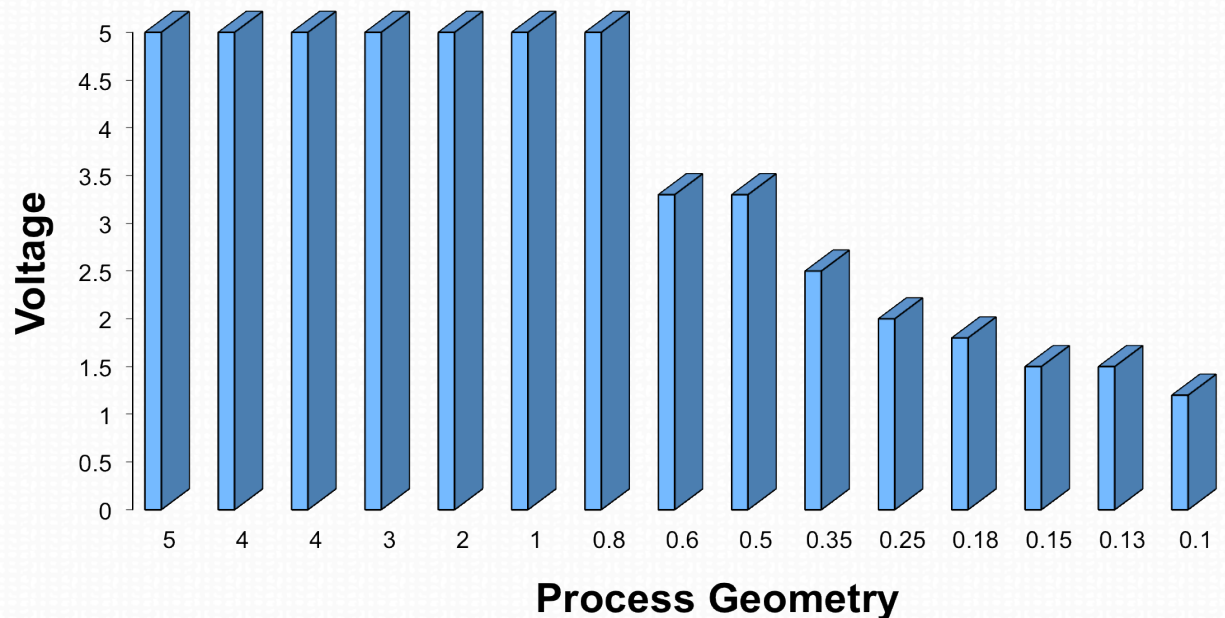
180 nm



130 nm

Technology Scaling

- Device dimensions W, L, t_{ox} : s
- Transistor density: s^2
- Speed (before power wall...)
 - V_{dd}, V_t : s
 - delay: s
 - frequency: $1/s$



Technology Scaling

- **Energy**
 - $E = C.V_{dd}^2$
 - Capacitances $C=W.L.C_{ox}$: s
 - Energy: s^3
- **Power** is decreased by 50%
 - $P = f.C.V_{dd}^2$
 - Power: s^2
 - Activity is supposed constant
- But this is for a constant transistor count!
 - But...Transistor density ($\#Tr/cm^2$): s^2
 - **Power Density**
 - And power supply current increases a lot
 - 100W at 1v equals to...?

Technology (Dennard's) Scaling

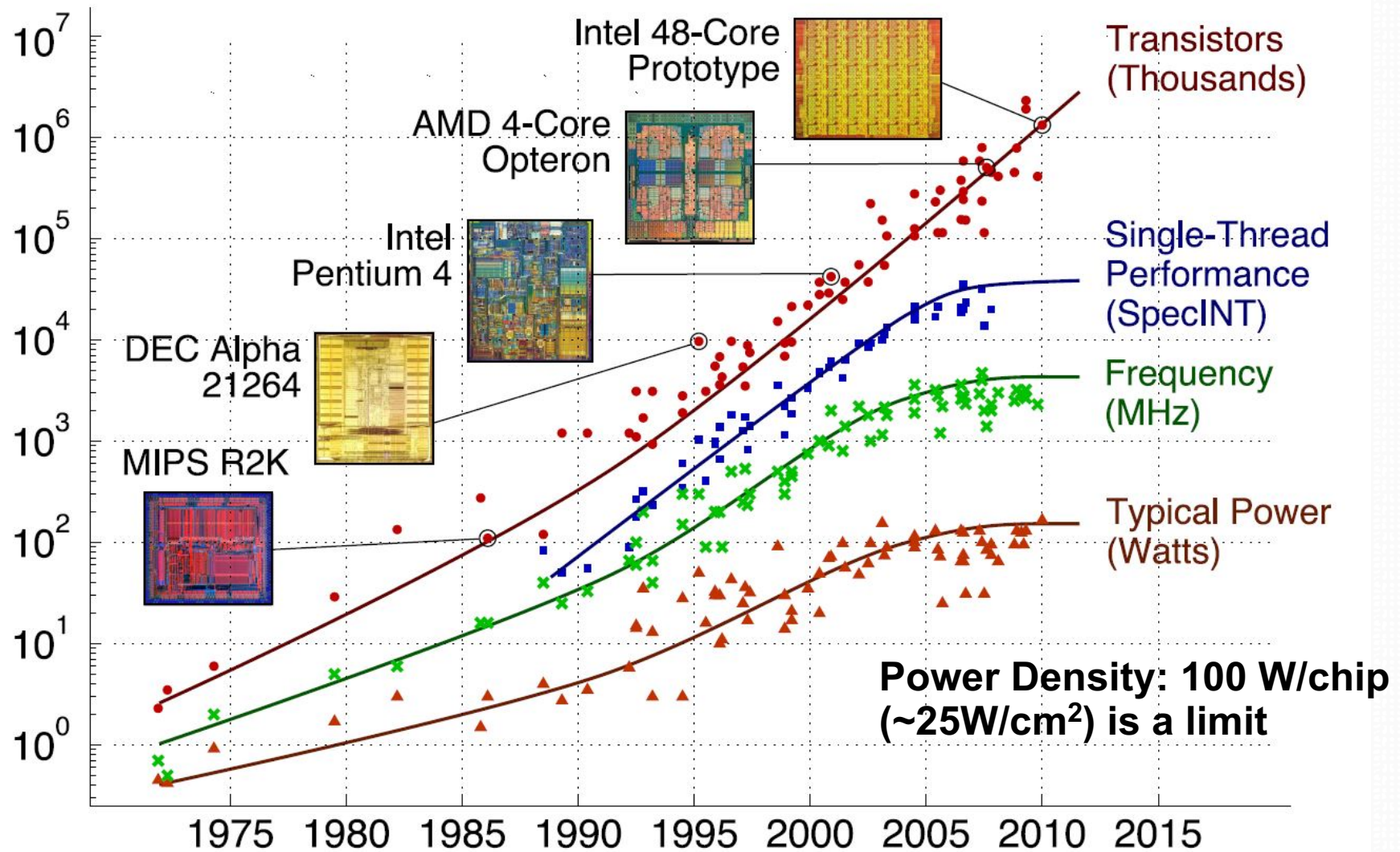
- Scaling factor: s
- Between two successive generations: $s \approx 0.7$

Device dimensions : W, L, t_{ox} , junction depth	s
Transistor area (W.L)	s^2
Capacitance per unit area : C_{ox}	$1/s$
Capacitances : $C=WLC_{ox}$	s
V_{dd} , V_t	s
Gate delay	s
Power/gate	s^2
Power.delay product	s^3
Power density	1

Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Technology Scaling (Moore's Law revisited)
- **Multicore: power and utilization walls**

And then came the "Power Wall"

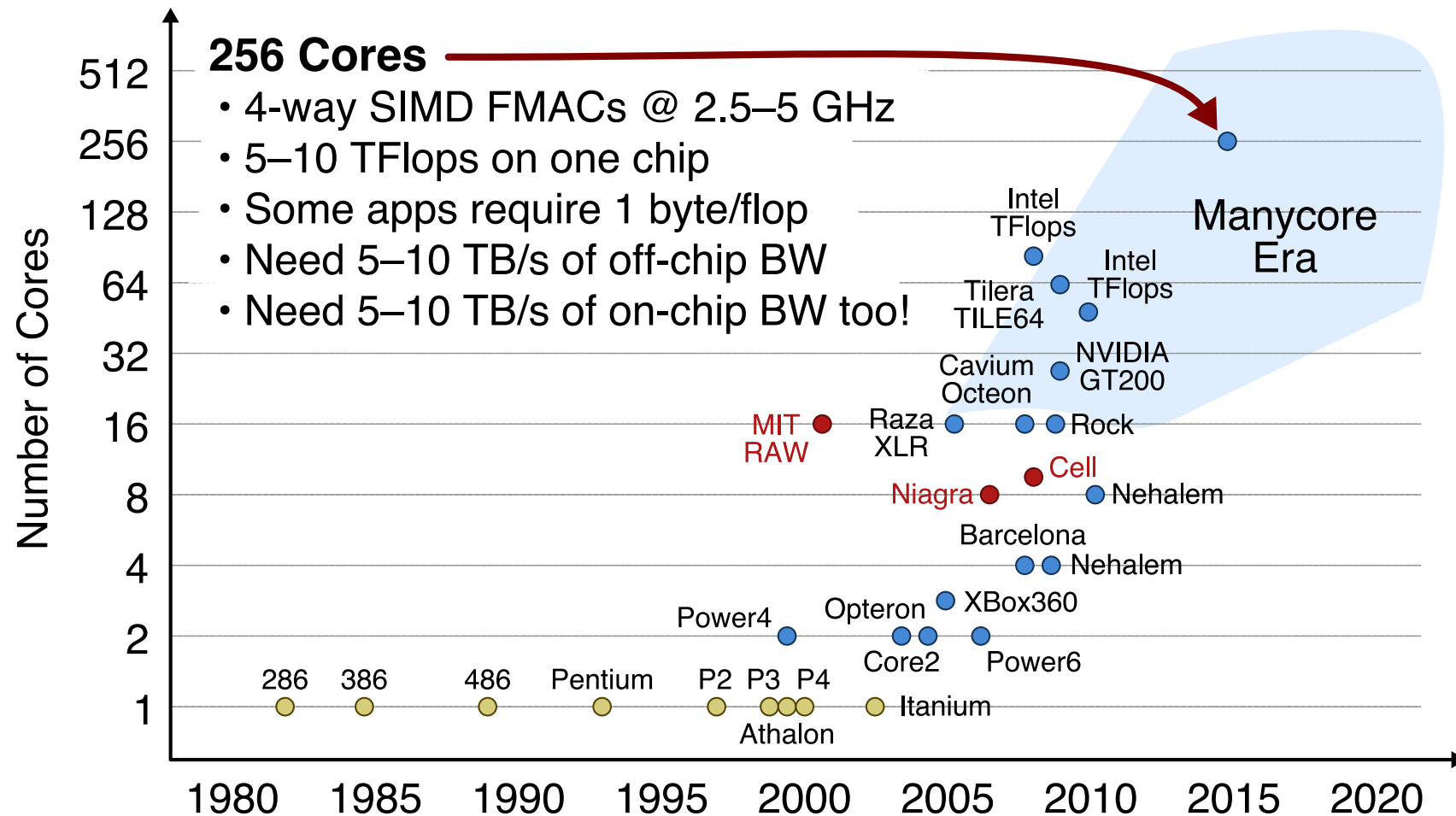


Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond

Source: C. Batten, Cornell

and the “Multicore Era”

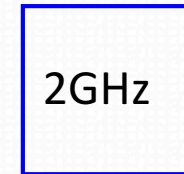
- Increasing performance by increasing # of cores



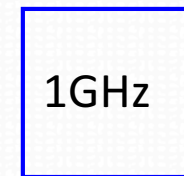
–Source: C. Batten, Cornell

Moving to multicore

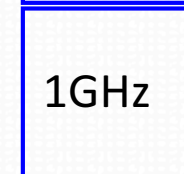
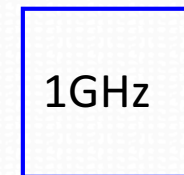
- 1 core@2GHz@1.2V@1W
- 1 core@1GHz@0.8V@0.25W
- 2 cores@1GHz@0.8V@0.5W
- But... twice area (and not so simple)
- **Advanced technology nodes?**



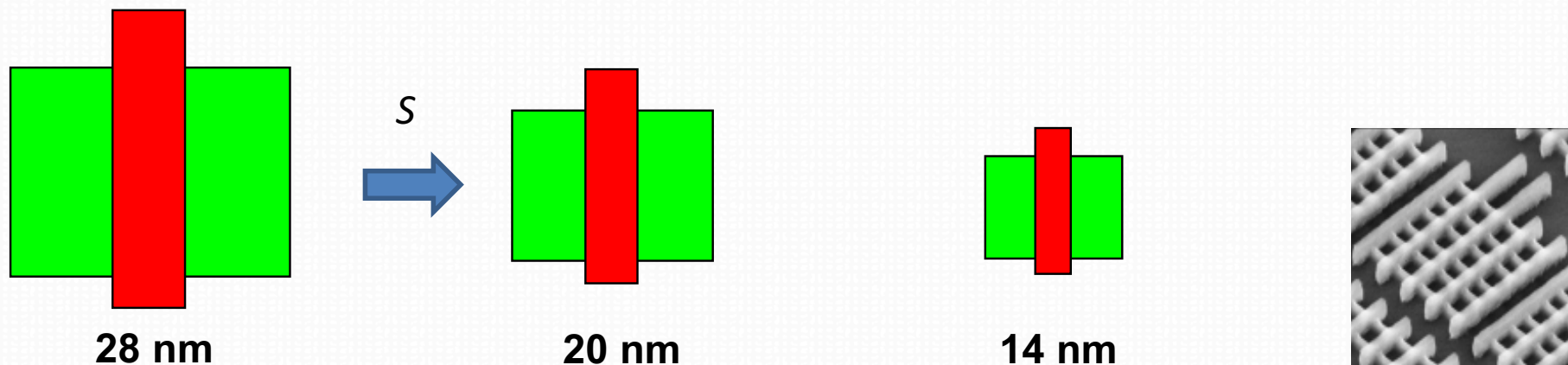
1W
1.2V



0.22W
0.8V

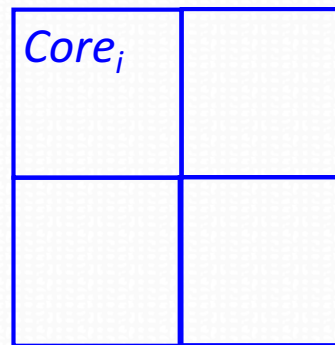


Technology Scaling

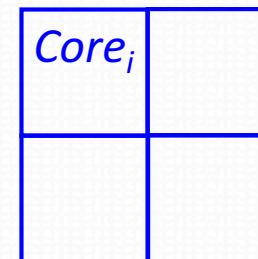


Classical (Dennard's) scaling

Device count	S^2
Device frequency	S
Capacitance, V_{dd}	$1/S$
Device power	$1/S^2$
Utilization	1



100W@ f



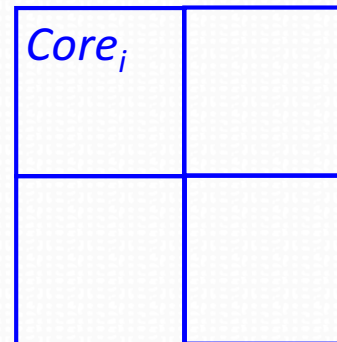
50W@ $1.4f$

End of Dennard's Scaling

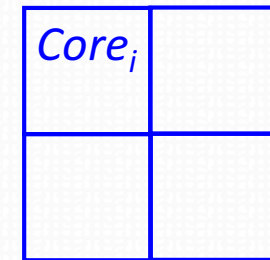
- Energy efficiency **is not** scaling along with integration capacity

Leakage limited scaling

Device count	S^2
Device frequency	S
Device power (cap)	$1/S$
Device power (V_{dd})	~ 1
Utilization	$1/S^2$

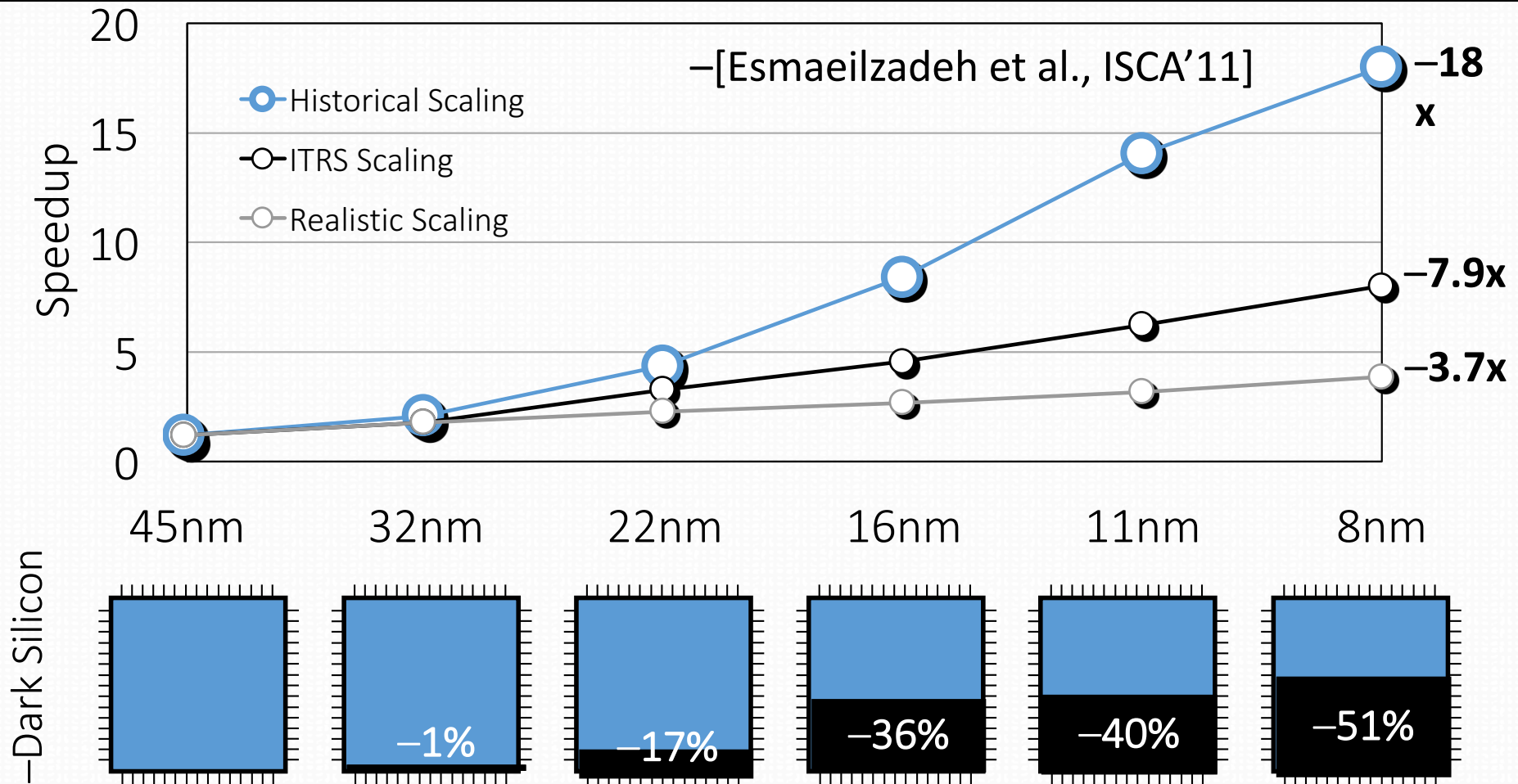


$100W@f$

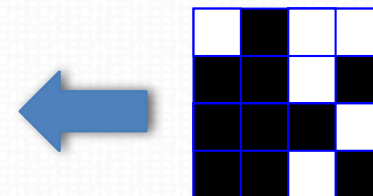


$100W@1.4.f$
(w/o) leakage

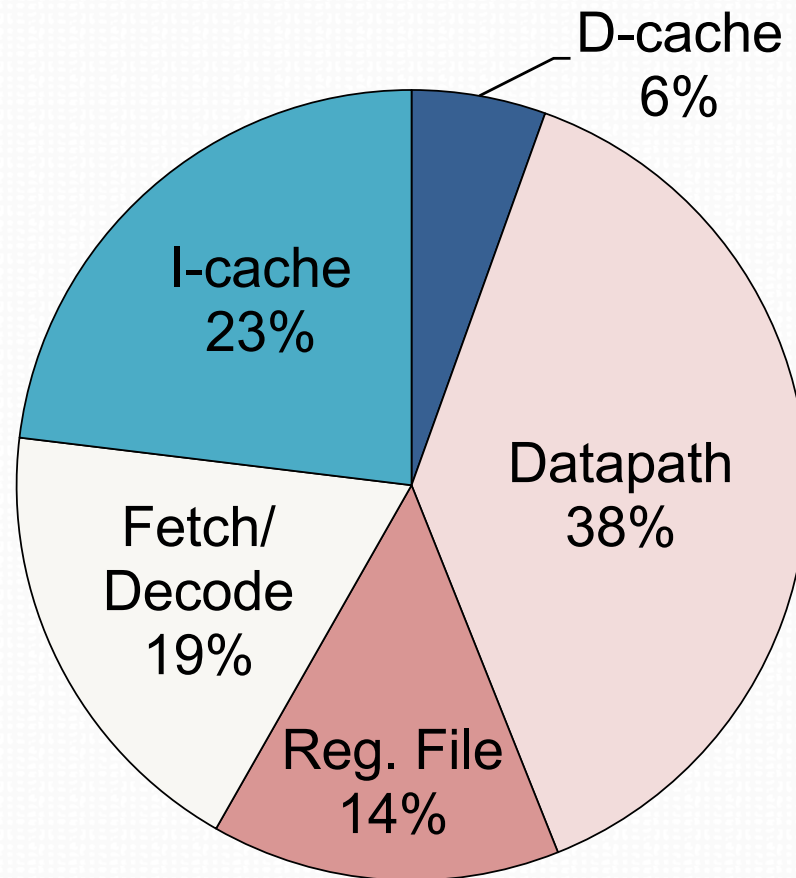
Multicore and Dark Silicon



- Replace dark cores with specialized cores (10-100x more energy efficient)



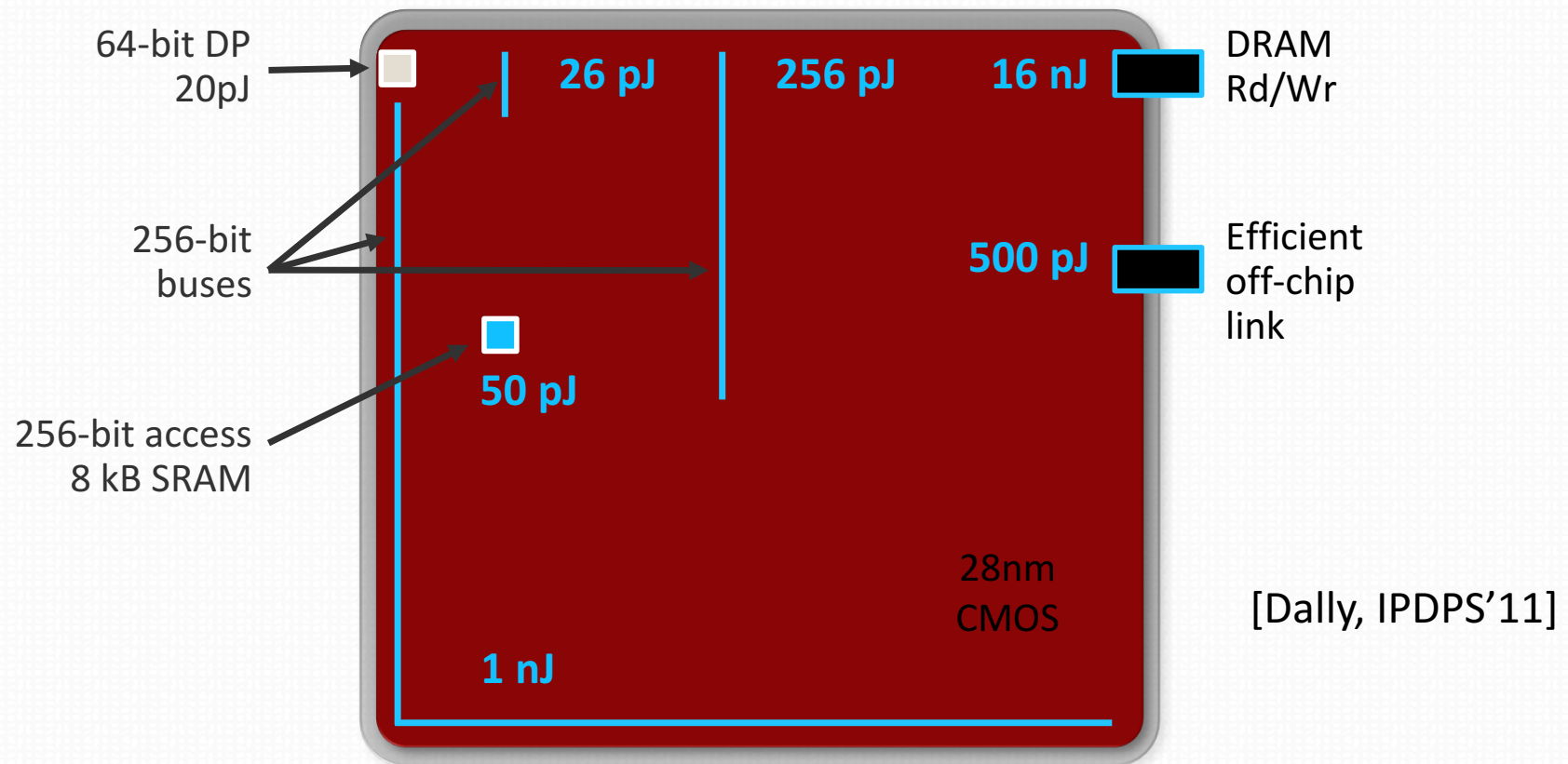
Energy Cost in a Processor



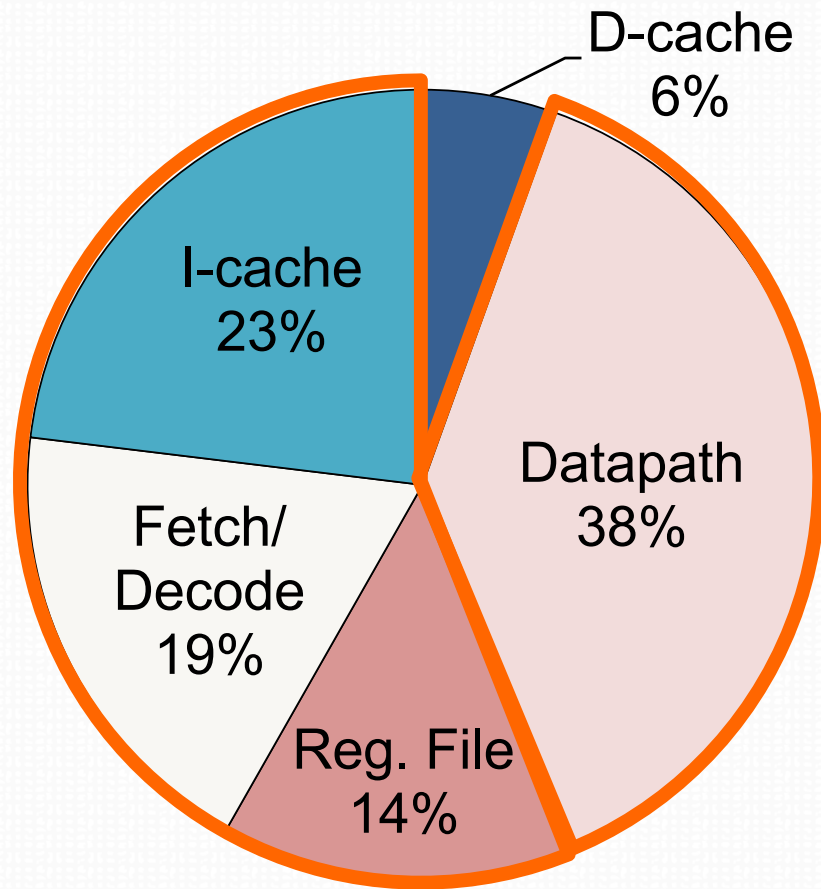
MIPS processor
91 pJ/instr.

Energy Cost in a Processor

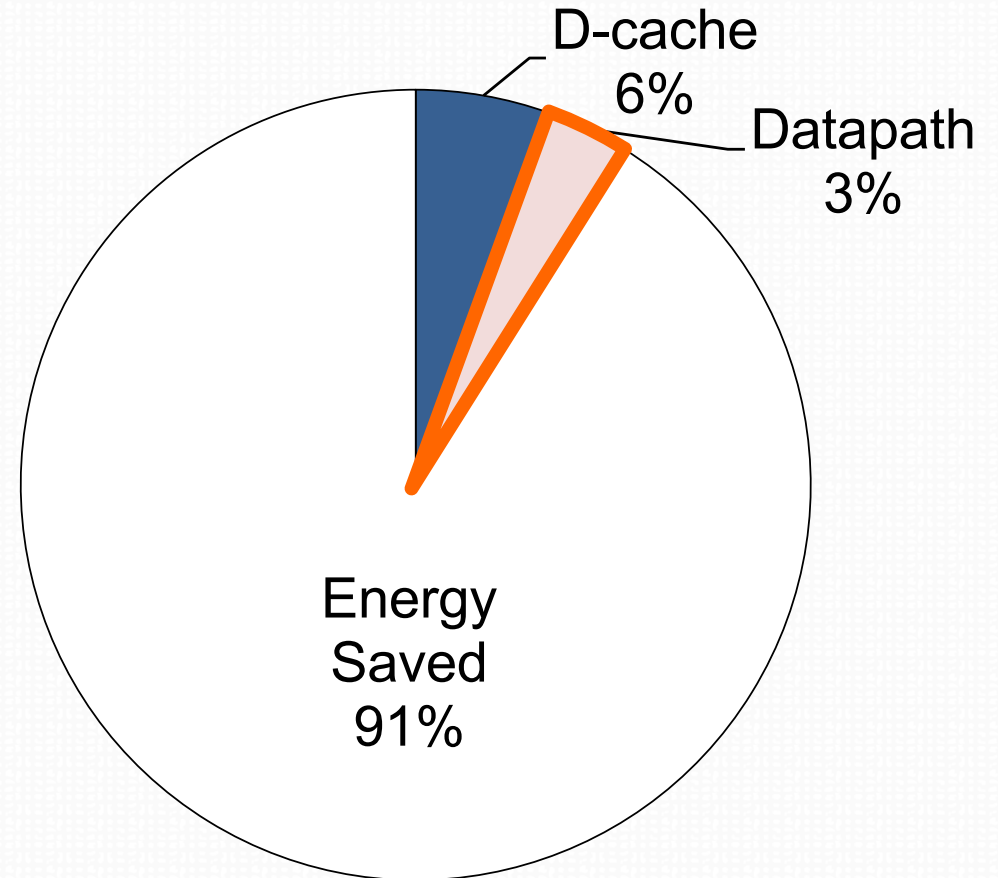
- Fetching operands costs more than computing



Energy Savings in Specialized HW



MIPS baseline
91 pJ/instr.



Specialized core
8 pJ/instr.

An example: Bitcoin Mining



Type	Model	Mhash/s	Mhash/J	Power (W)
GPP	Intel Xeon X5355 (dual)	22.76	0.09	120
GPP	ARMCortex-A9	0.57	1.14	1.5
GPP	Intel Core i7 3930k	66.6	0.51	130
GPU	AMD 7970x3	2050	2.41	850
GPU	Nvidia GTX460	158	0.66	240
ASIC	AntMiner S1	180.000	500	360
ASIC	AntMiner S5	1.155.000	1957	590
FPGA	Bitcoin Dominator X5000	100	14.7	6.8
FPGA	Butterflylabs Mini Rig	25.200	20.16	1250



BITCOIN MINER

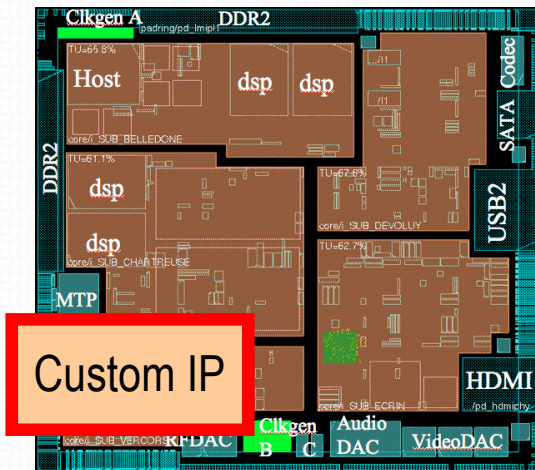
Time has Come for Specialization

- Microsoft Unveils Catapult to Accelerate Bing!
 - One FPGA per blade
 - 6×8 2-D torus topology
 - High-end Stratix V FPGAs
- Running Bing Kernels for feature extraction and machine learning
- Increase **ranking throughput by 95%** at comparable latency to software-only
- Increase power consumption by 10%
- Increase total cost of ownership by less than 30%



Towards Heterogeneous Multicores

- Embedded and High-Performance Computing



Embedded heterogeneous multicore



Heterogeneous platforms



FPGA accelerators for HPC

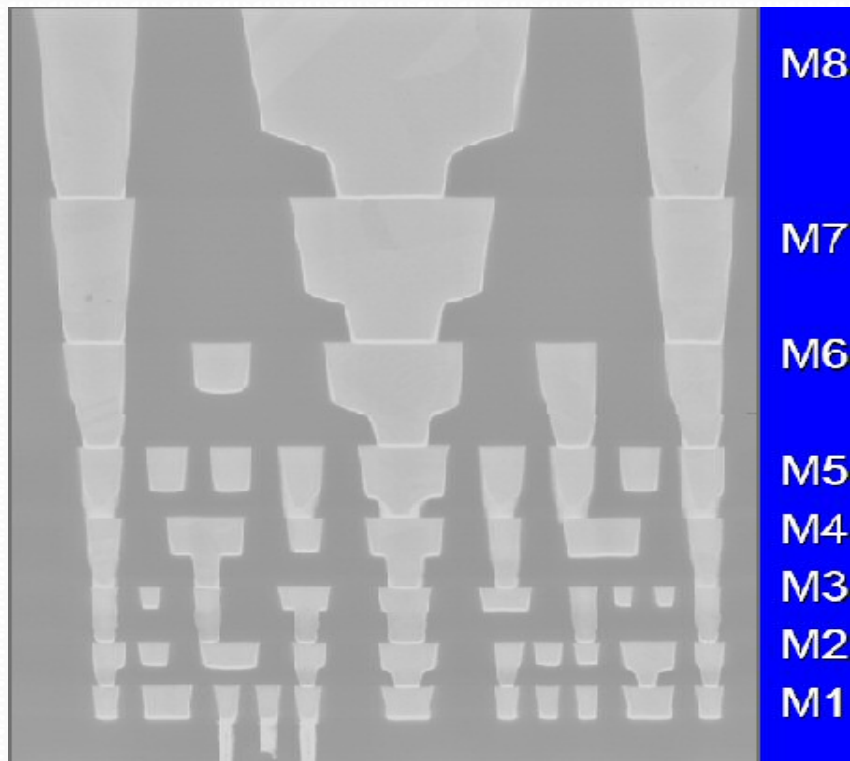
- C to hardware **high-level synthesis** boosts hardware designer productivity

Conclusions

- A **not too** deep dive into processors?
- Transistors, logic gates, registers and memory
- Delay and maximal frequency
- **Power** is data dependent and dominated by data transfers
- Energy efficiency **is no more** scaling along with integration density
- Efficiency of hardware **specialization**
- Dark Silicon is an opportunity
 - **Heterogeneous** manycore architectures
 - Bring a new demand for genuinely **high level synthesis tools** and (JIT) **compilers** that map programs to accelerators

On-Chip Interconnect?

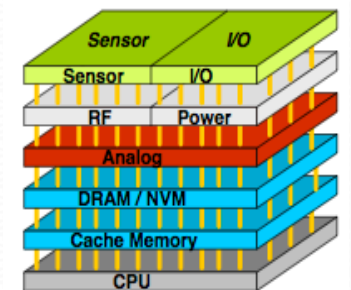
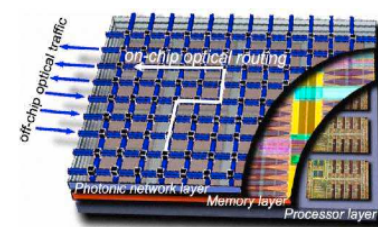
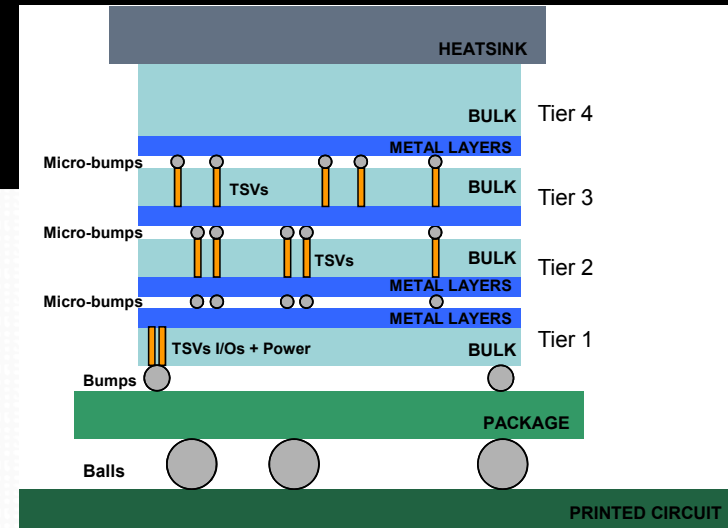
- Gate delay decreases but... wire delay increases
- Crossing chip in 5-10 clock cycles
- Also affected by noise...



- Metal layers to reduce wire delay
- Repeaters
- Towards **network-on-chip**

Chips go 3D!

- 3D Integrated Circuits
 - Stack Multiple Dies
- Wire Length Reduction
 - Replace long, high capacitance wires by Through Silicon Vias (TSVs)
 - Low latency, low energy, high bandwidth
- Heterogeneous Integration
 - Image Sensors, Sensor Network Nodes
 - **Processor + Memory**



3D Heterogeneous Multicores

- 3D Optical Manycore Project

