Design of VLSI Integrated Circuits

A (very) deep dive into computing machines…

Olivier Sentieys
Inria
Univ. Rennes, Irisa
olivier.sentieys@inria.fr

http://people.rennes.inria.fr/Olivier.Sentieys

Cairn Team at a Glance

Inria Rennes

Enssat Lannion

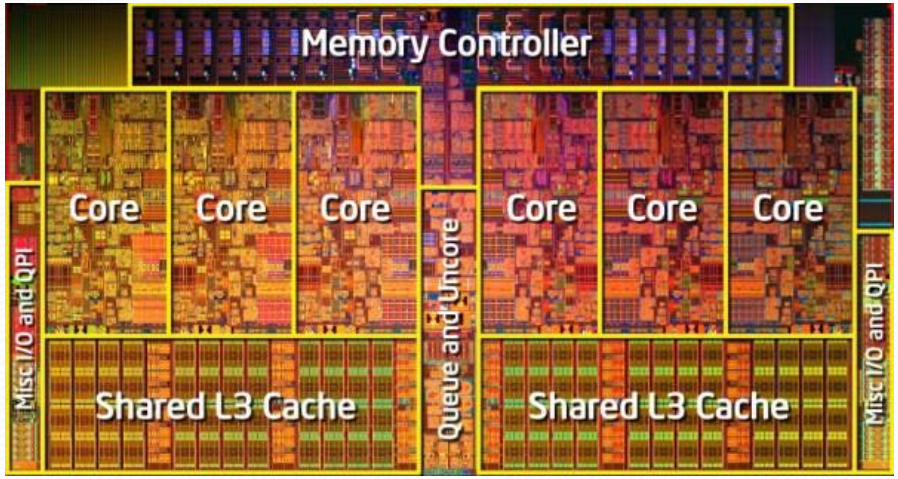
- *Energy-Efficient Computing Architectures*

- IRISA/INRIA
- ~35 people, Rennes and Lannion campuses
- from INRIA, Univ. Rennes 1, ENS Rennes
- Electrical Engineering & Computer Science

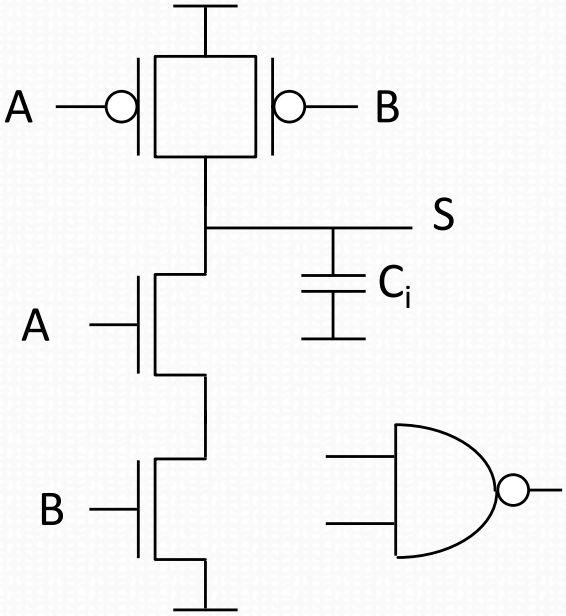
- Domain-specific computing architectures
- Design tools and compilers

VLSI Integrated Circuit Design

- Chips, logic gates and transistors

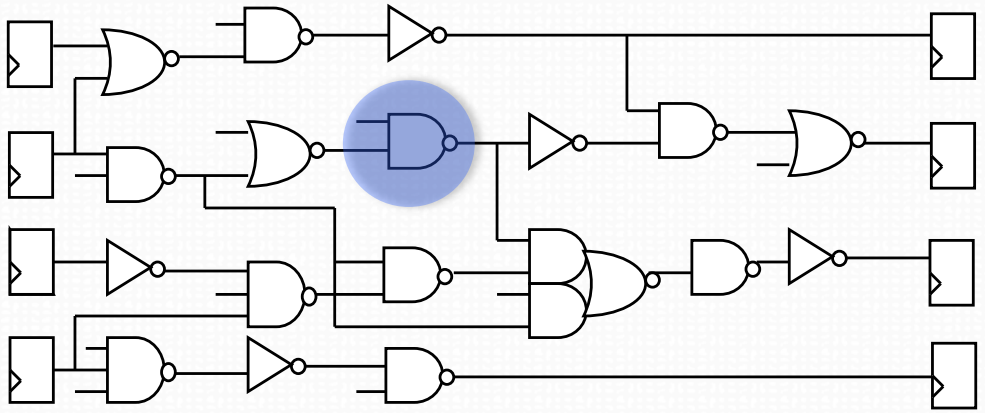


Intel's Xeon Chip




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

```
  end if;
end process;
```



Key Questions

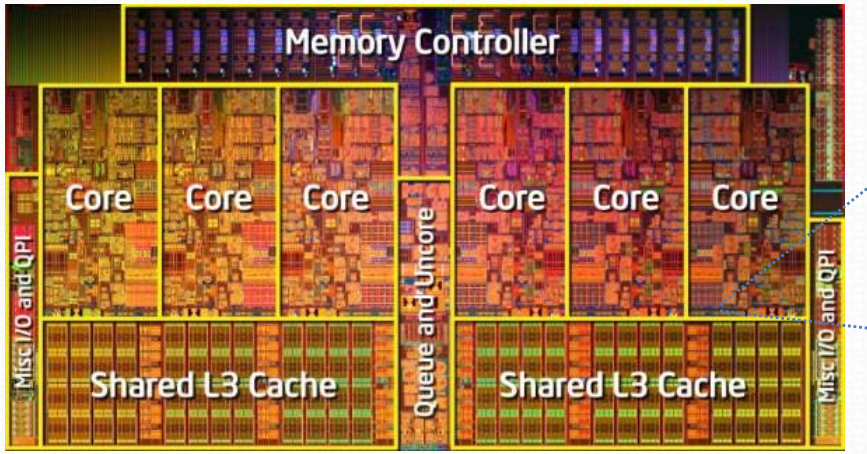
- A deep dive into processors… *(I hope not too deep)*
- What is CMOS? How basic logic gates, registers and memory are designed?
- How to calculate the delay and the maximal frequency?
- How much power does my processor consume?
- What can advanced semiconductor technology bring?
- Are (homogeneous)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

Outline

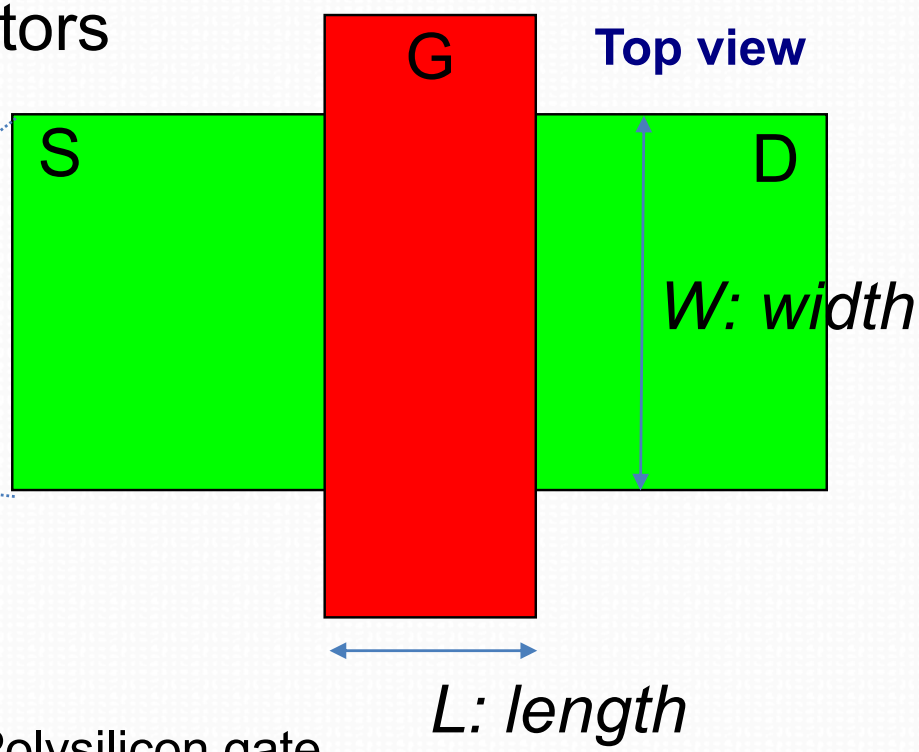
- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

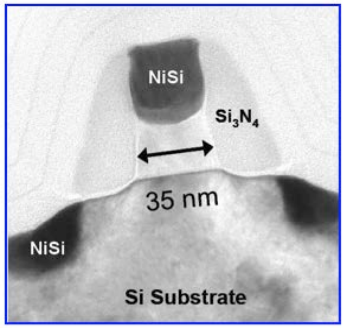
Fundamental Building Block: MOSFET Transistor

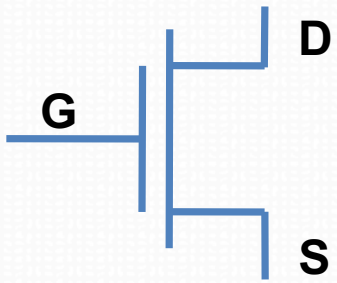
Now several billions or transistors

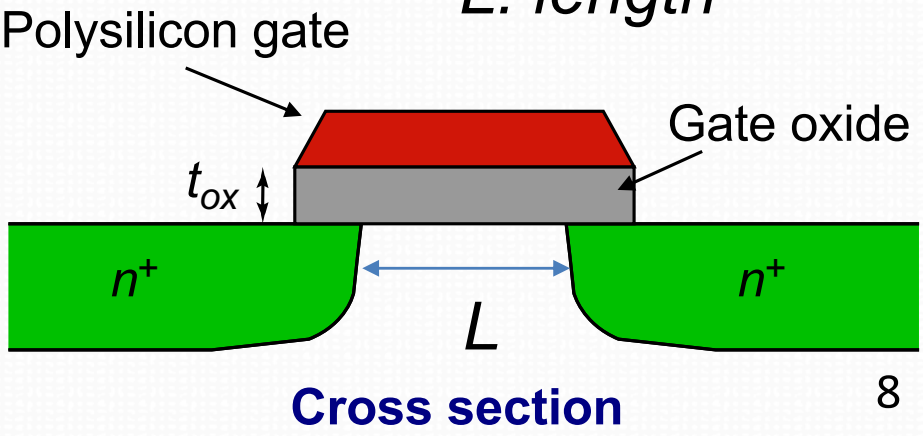


Intel's Xeon Chip





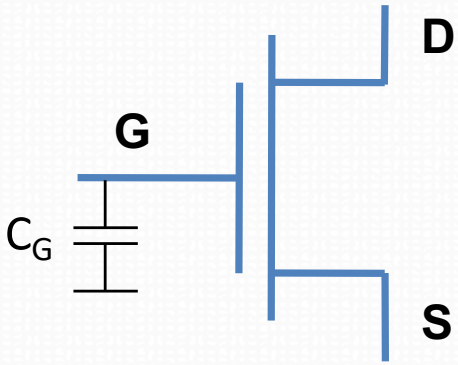


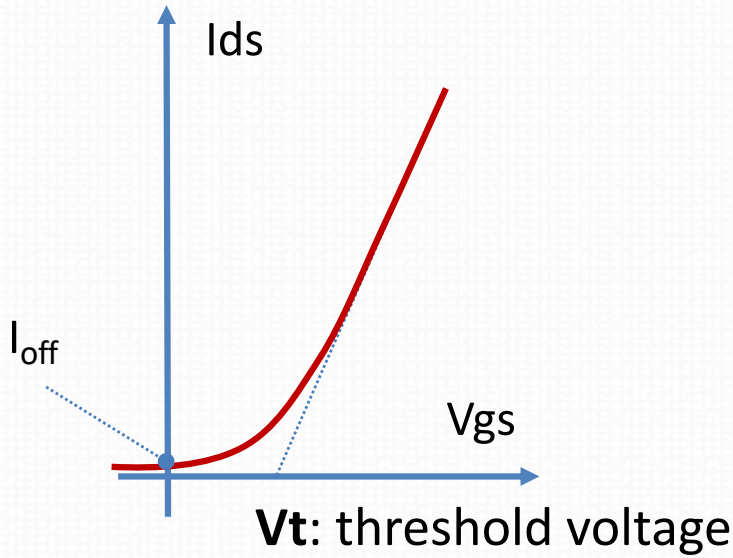


MOSFET: Metal Oxide Silicium Field Effect

The Basic Element: Transistor

- Transistor as a switch



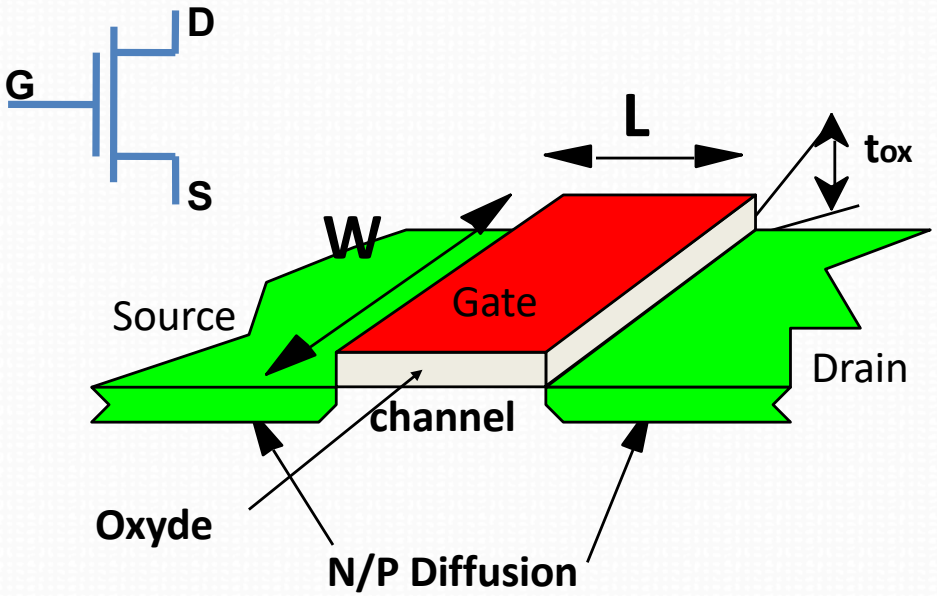


- Vgs > Vt: NMOS on
  - Resistance R_{DS}

- Vgs < Vt: NMOS off
  - Leakage I_{off}

- Gate: capacitance C_G
- Switch: resistance R_{DS}

MOS Transistor Models



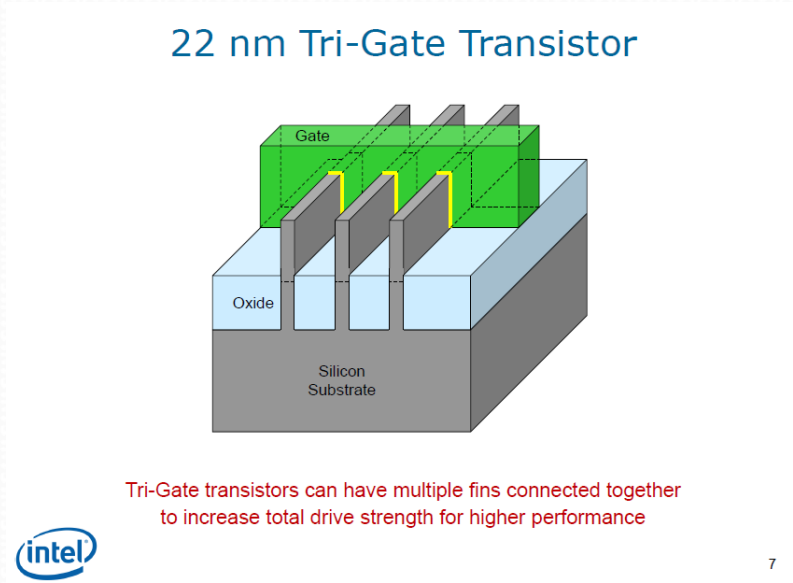
- ***W***: gate width
- ***L***: gate length
- t_{ox}: oxyde width (#L/10)
- $K = \mu.\varepsilon.W/(t_{ox}.L) = k\ W/L$
- μ: charge-carrier effective mobility
  NMOS (electrons) μ_N = 500 cm^2/V-sec # 2 µP
  PMOS (holes) μ_P = 270 cm^2/V-sec
- ε: oxyde permittivity # 4 ε_0 = 3.5 10^{-13} F/cm

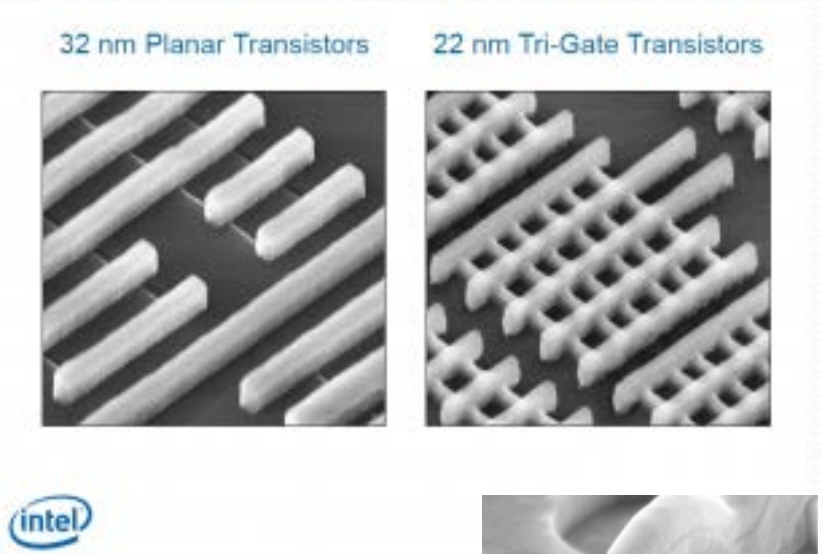
$$I_{ds} = \begin{cases} 0 & \text{off} & Vgs - Vth < 0 \\ K\left[(Vgs - Vth)Vds - \frac{Vds^2}{2}\right] & \text{linear} & 0 < Vds < Vgs - Vth \\ \frac{K}{2}(Vgs - Vth)^2 & \text{saturated} & 0 < Vgs - Vth < Vds \end{cases}$$

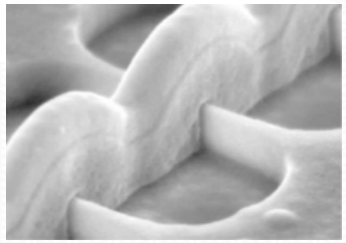
- *K* defines transistor speed, $K \propto W/L$, $K_{NMOS} \sim 2.K_{PMOS}$
- Temperature increases → μ decreases

Transistors Nowadays

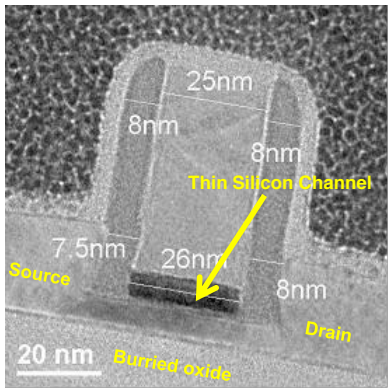
- Intel FinFET: transistors go 3D







- Fully Depleted SOI[1]
  - Low-power



[1]Silicon on Insulator

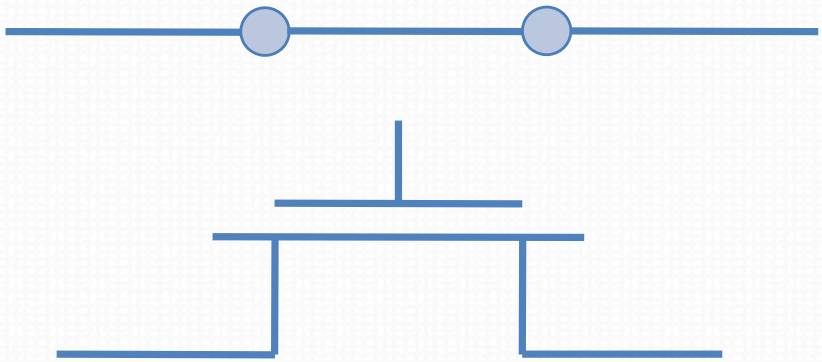
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

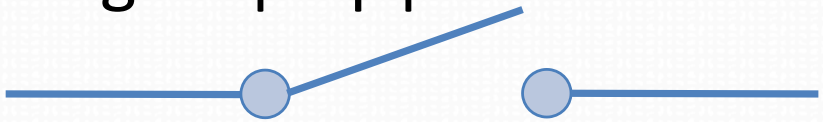
NMOS/PMOS Transistors

- NMOS
  - A ‘0’ is well transmitted
  - A degraded ‘1’ is transmitted (Vdd-Vtn)
- Vgs < Vtn

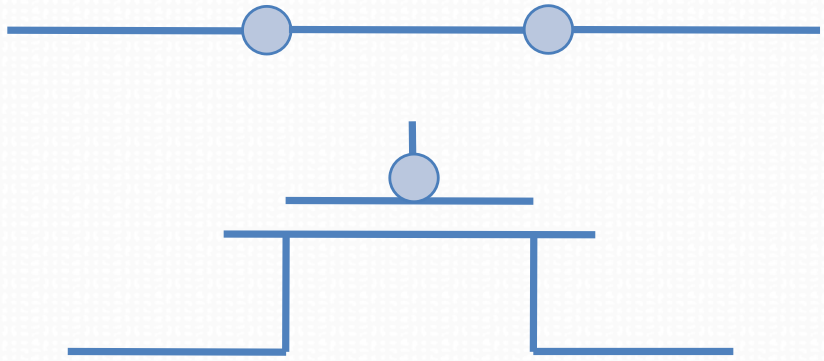
- Vgs > Vtn



- PMOS
  - A ‘1’ is well transmitted
  - A degraded ‘0’ is transmitted (Vss+|Vtp|)
- Vgs < |Vtp|

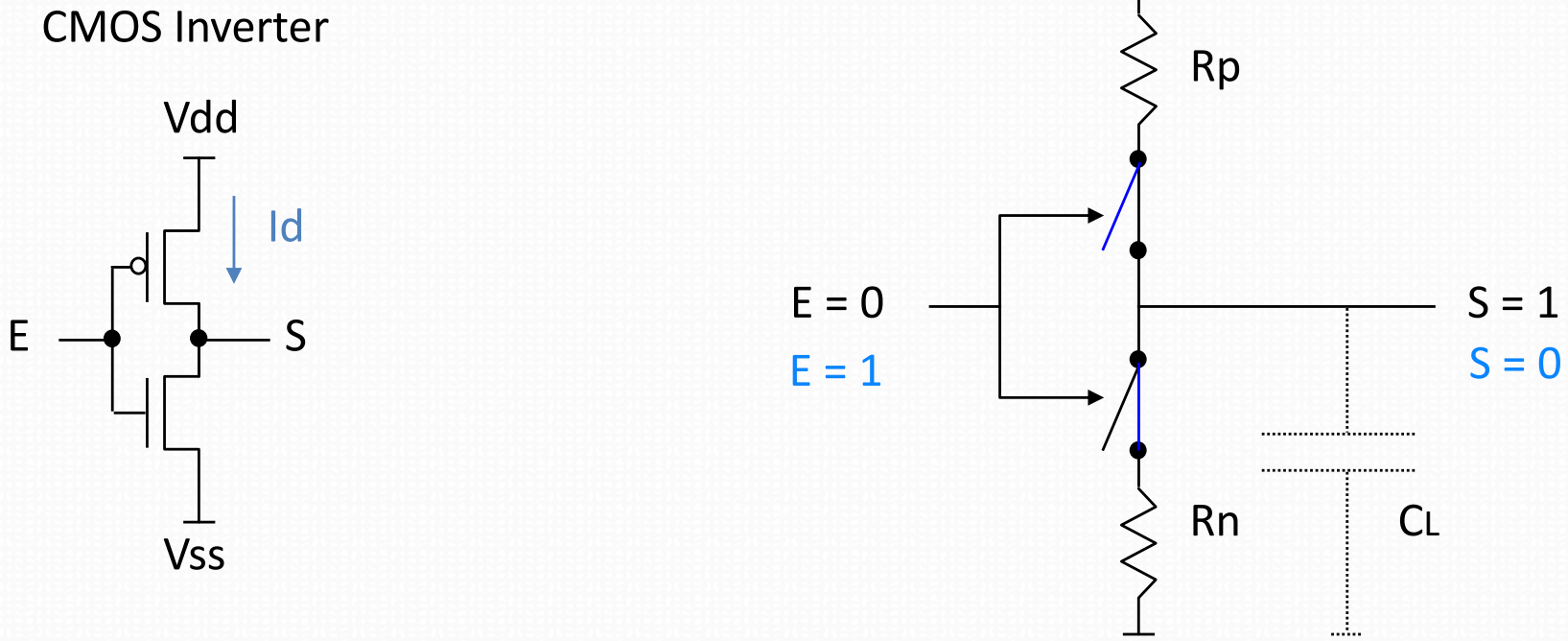


- Vgs > |Vtp|



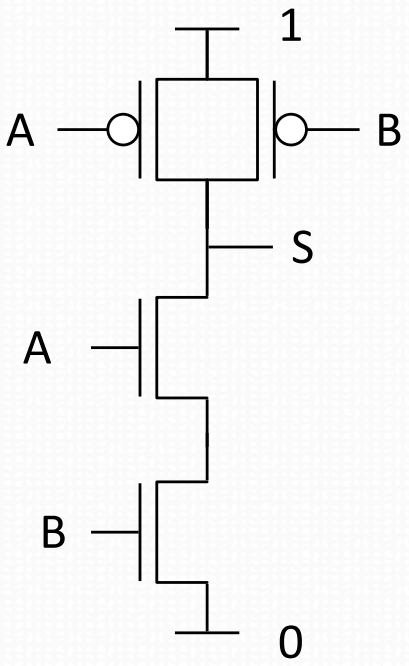
Combinatorial Logic Cells

- Complementary Logic (CMOS)

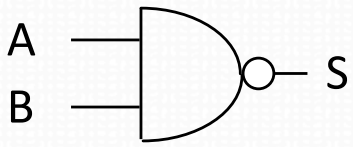


NAND and NOR

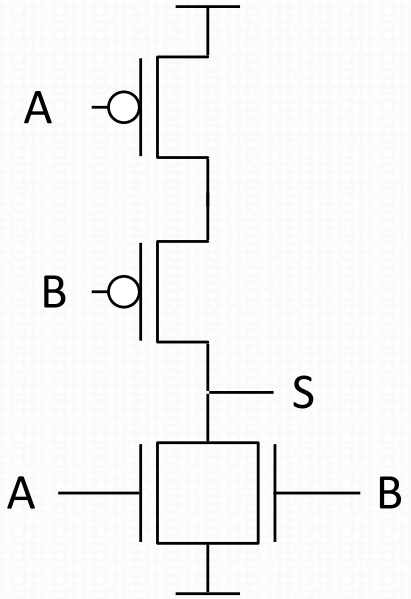
NAND



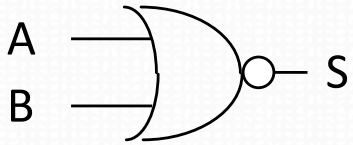
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



NOR

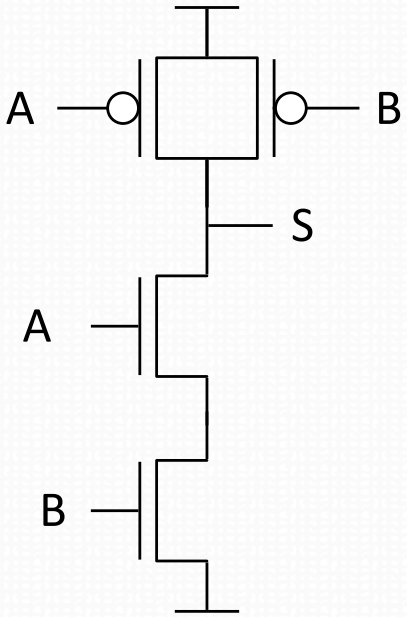


A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

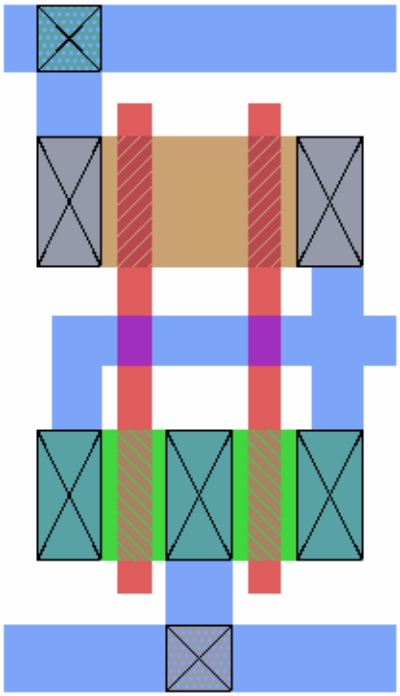


Layout Design

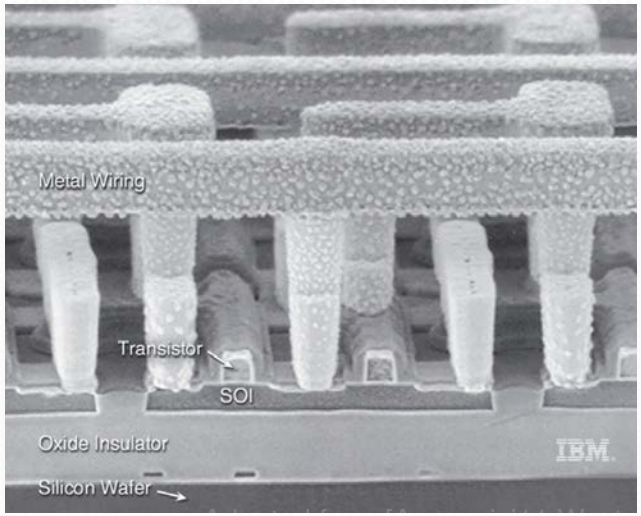
Transistor Schematic



Layout



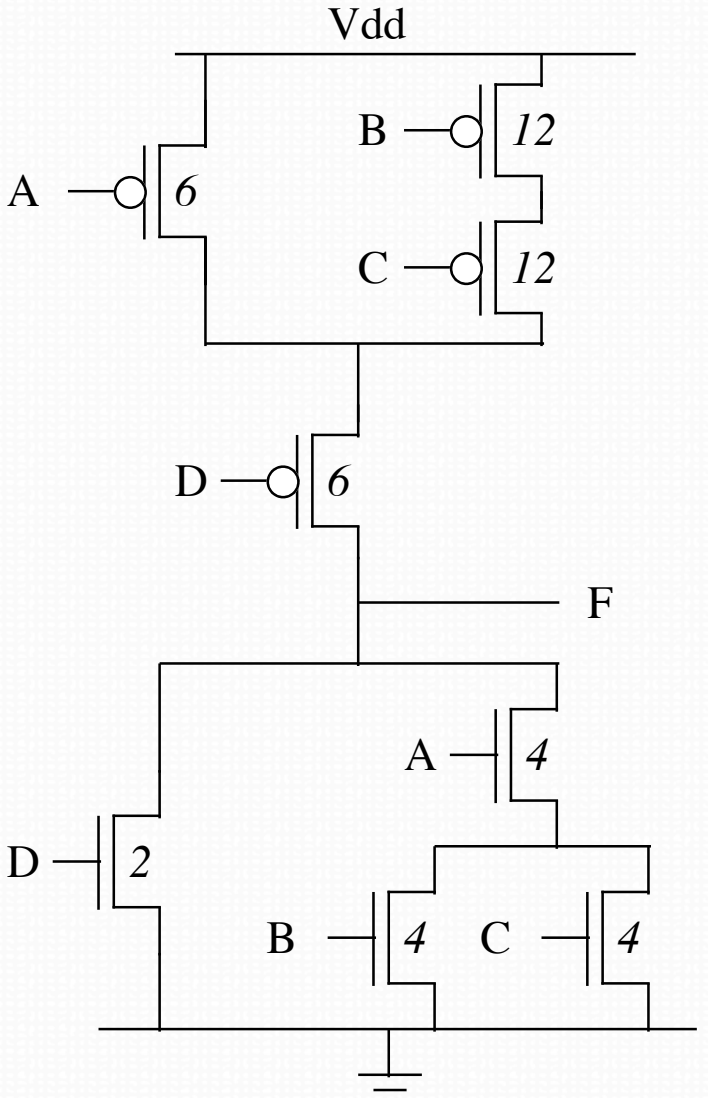
Silicon



Complex Gates

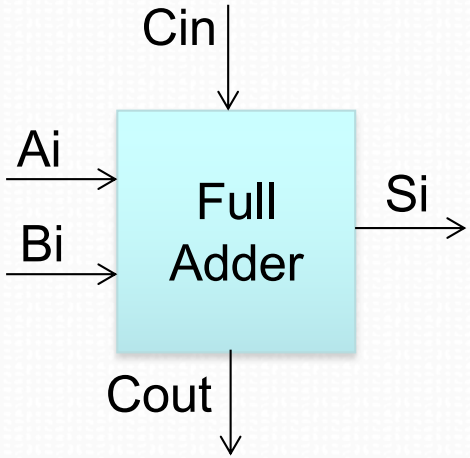
- $F = \overline{A.(B+C)+D}$

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

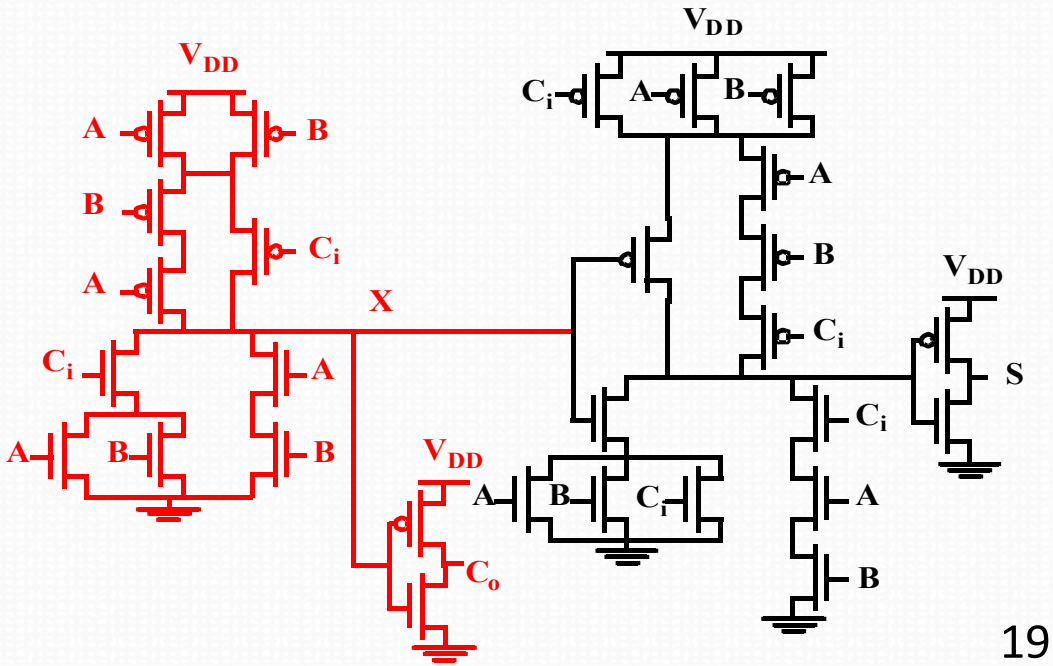


Complex Gates: Full Adder

- Full Adder

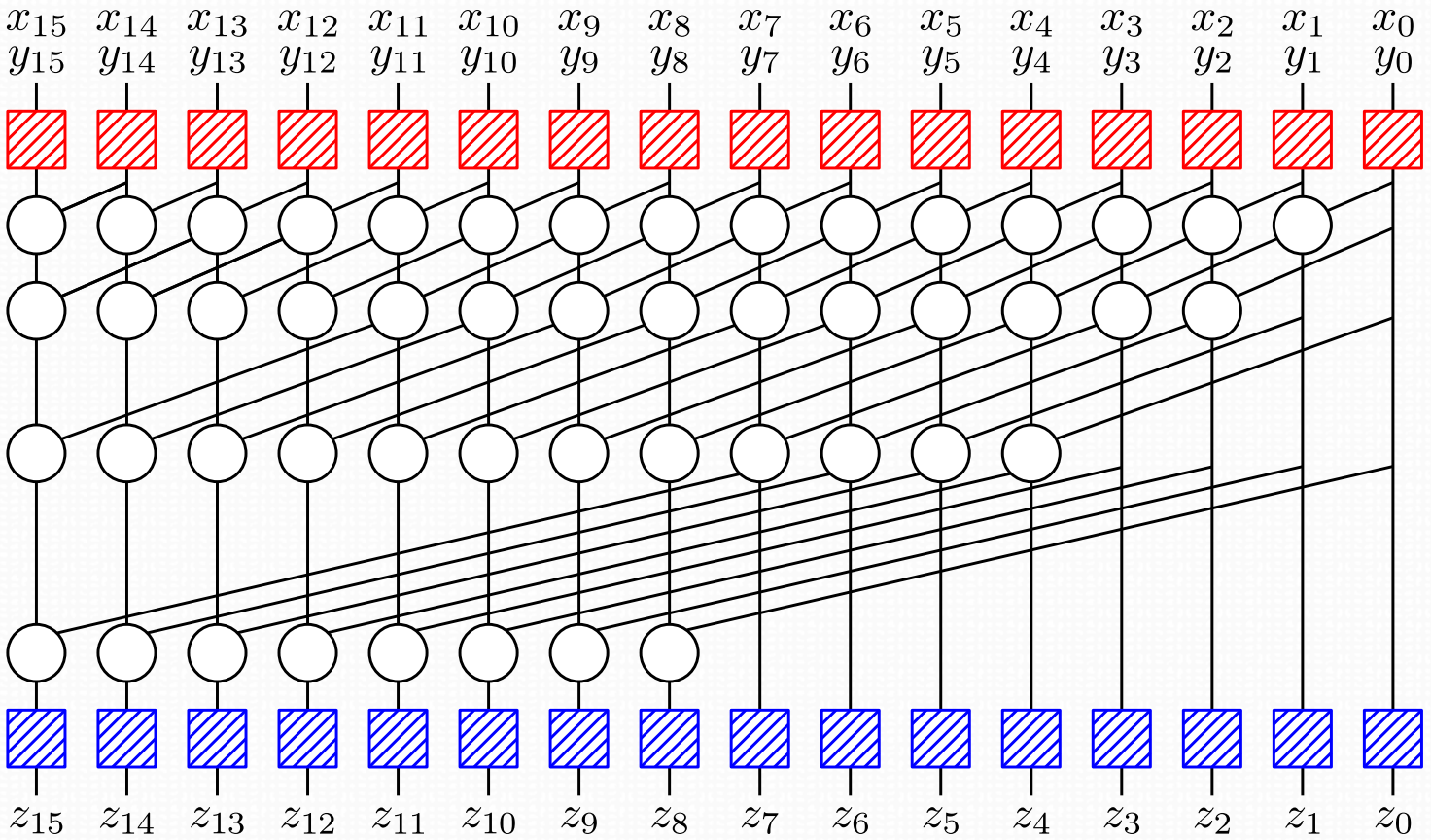


Ai	Bi	Ci	Co	Si
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



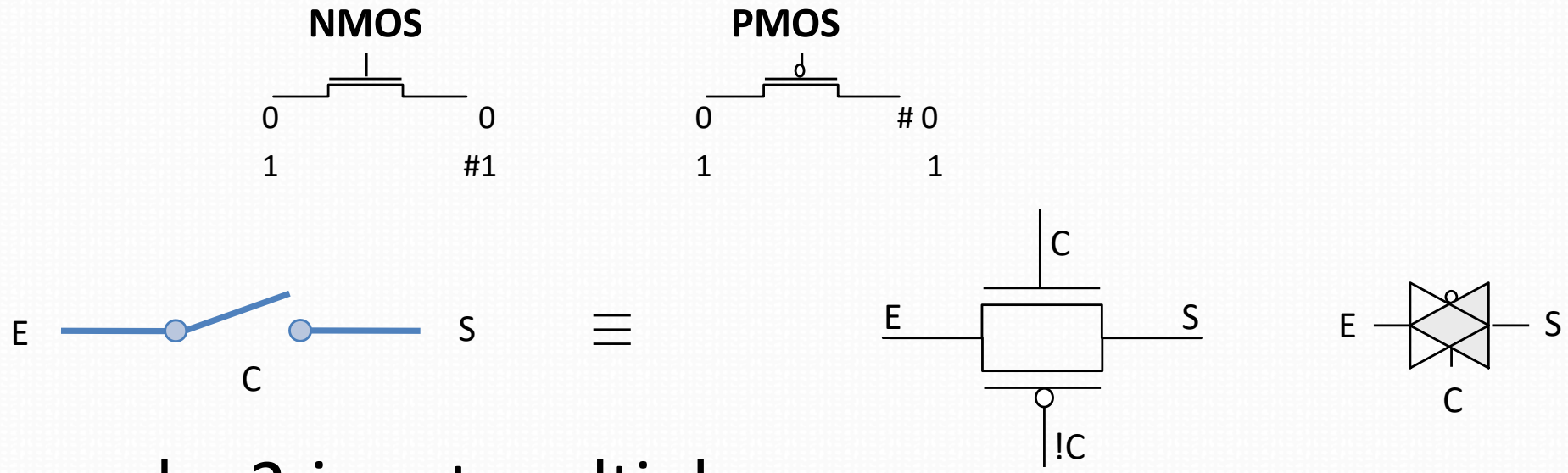
Complex Functions

- 16-bit Adder (integer)



Pass-Transistor Logic

- Switch or Transmission Gate



- Example: 2-input multiplexer

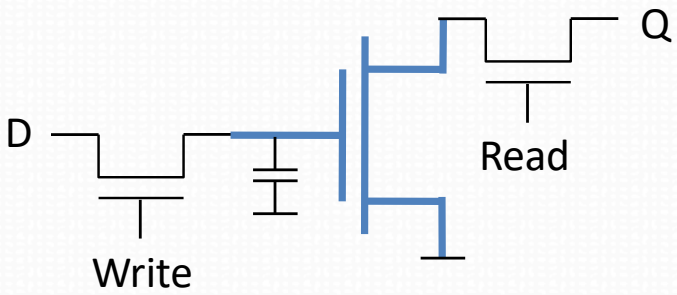
- Example: XOR

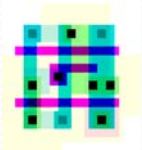
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

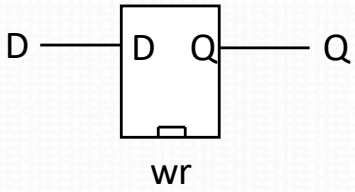
Storing Values

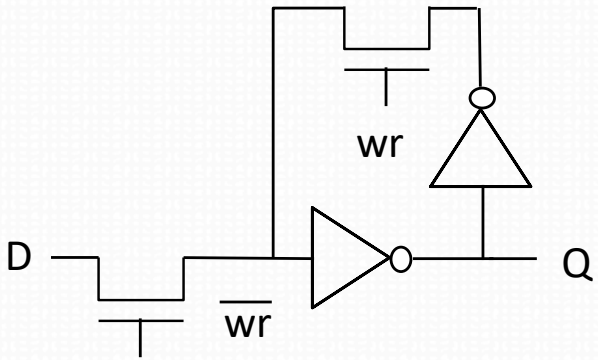
Capacitor (dynamic) (DRAM)

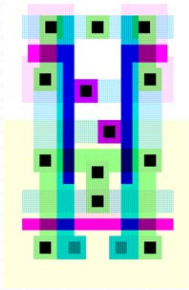




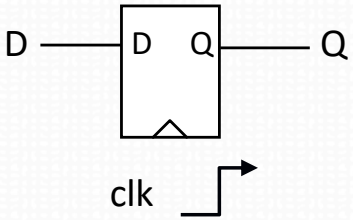
Latch (static) (SRAM)



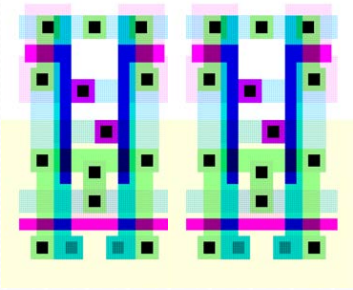




Flip-Flop (static) (Register)

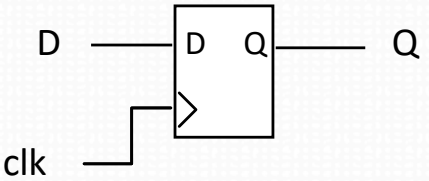


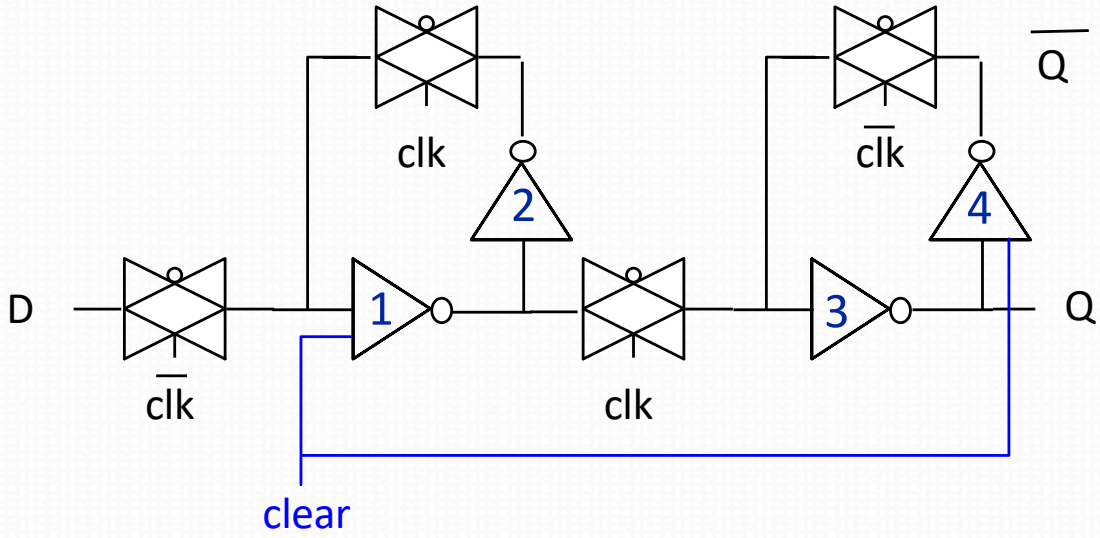
- Setup Time: Tsetup
- Hold Time: Thold
- Propagation Time: Tp



D Flip-Flop (edge-triggered)

- Two latches in series

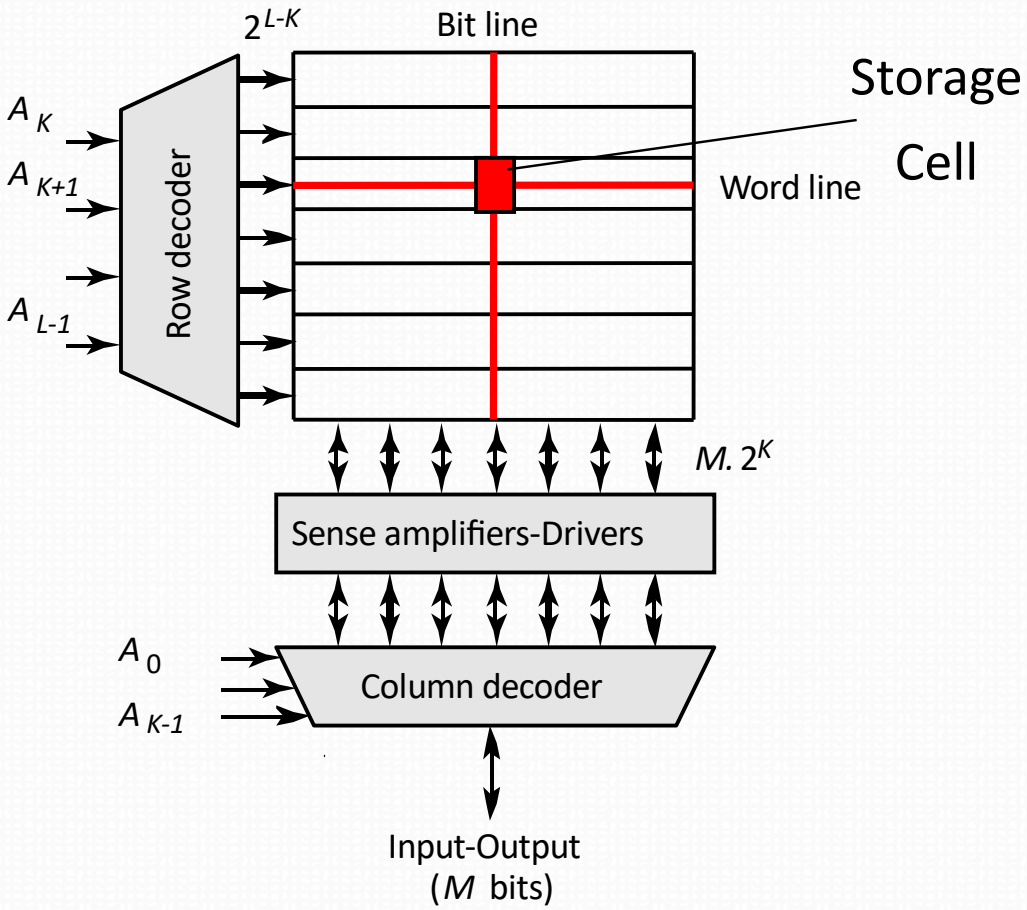


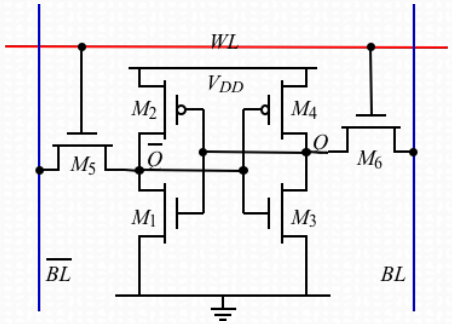


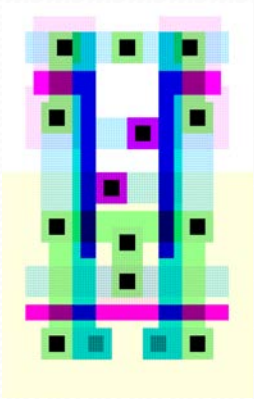
  - D is sampled in inverter (1) when clk = 0
  - Latch (1) and (2) keeps D value when clk = 1 until !D is transferred to second latch (3) and (4)
  - Asynchronous clear signal: replace inv. (1) and (4) by NAND

Memory

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



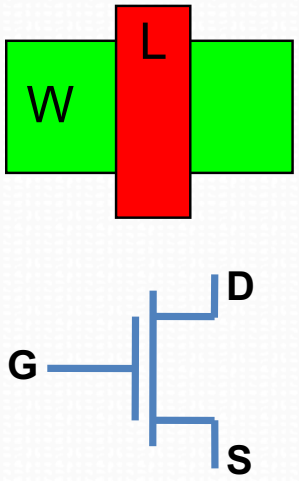


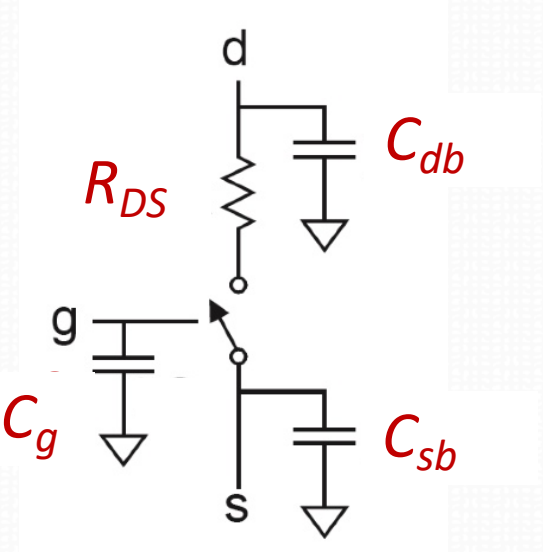


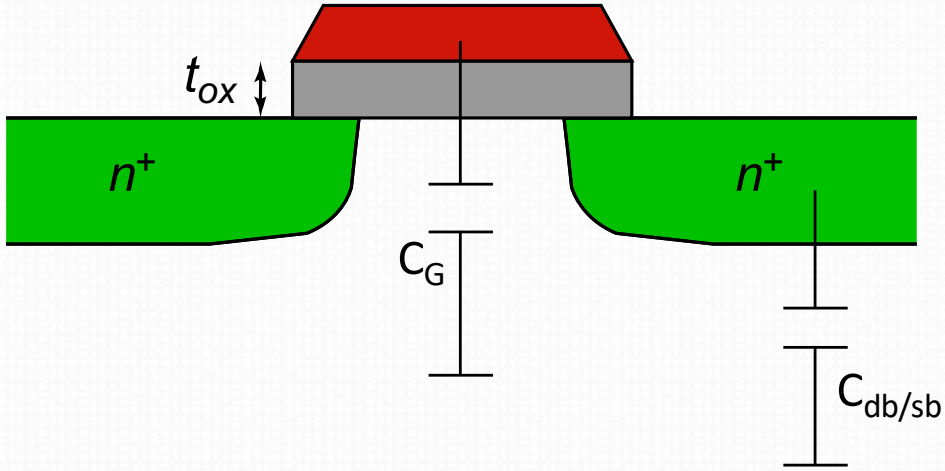
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

Delay: Parasitic Elements



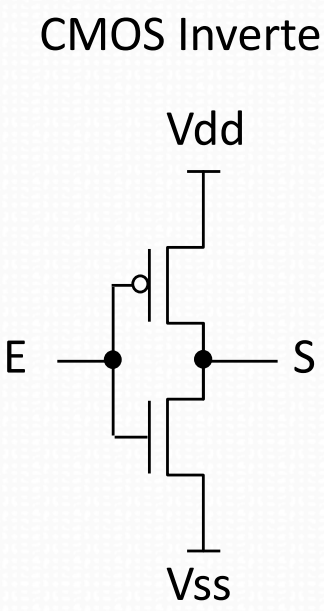


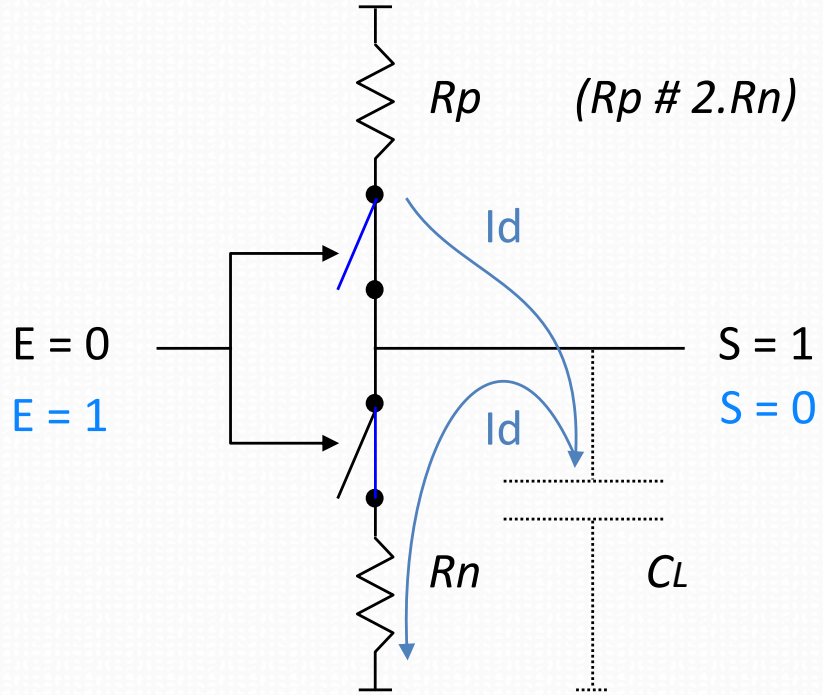


- Drain-Source Resistance: $R_{DS} = \frac{L}{W}\frac{1}{k(V_{dd}-V_t)}$
- Gate Capacitance: $C_g = \frac{\epsilon W.L}{t_{ox}} = W.L.C_{ox}$

$$\text{Delay} \propto R_{DS}.C_g \propto \frac{L^2}{Vdd-Vt}$$

Simplified Delay Model

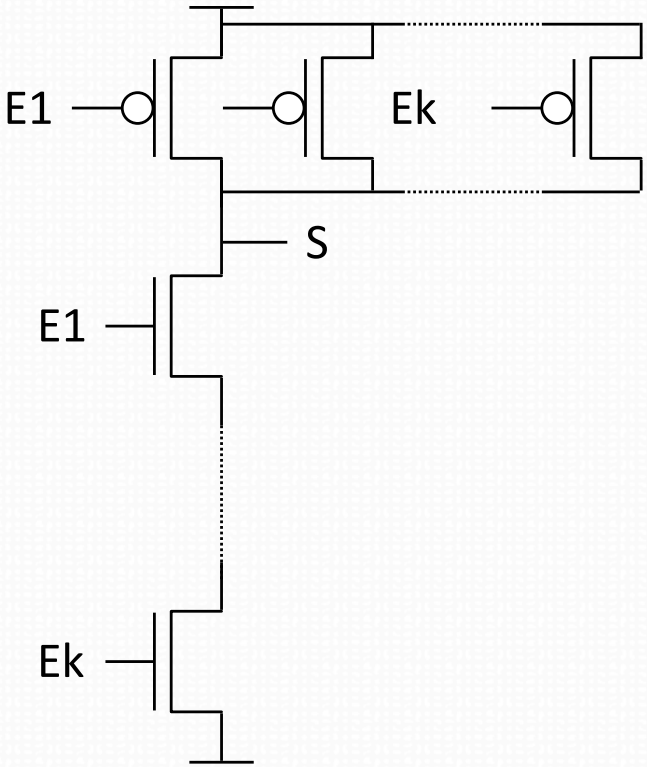




- Rising Output: $t_{plh} = Rp.C_L$
- Falling Output: $t_{phl} = Rn.C_L$

Delay of Complex Gates

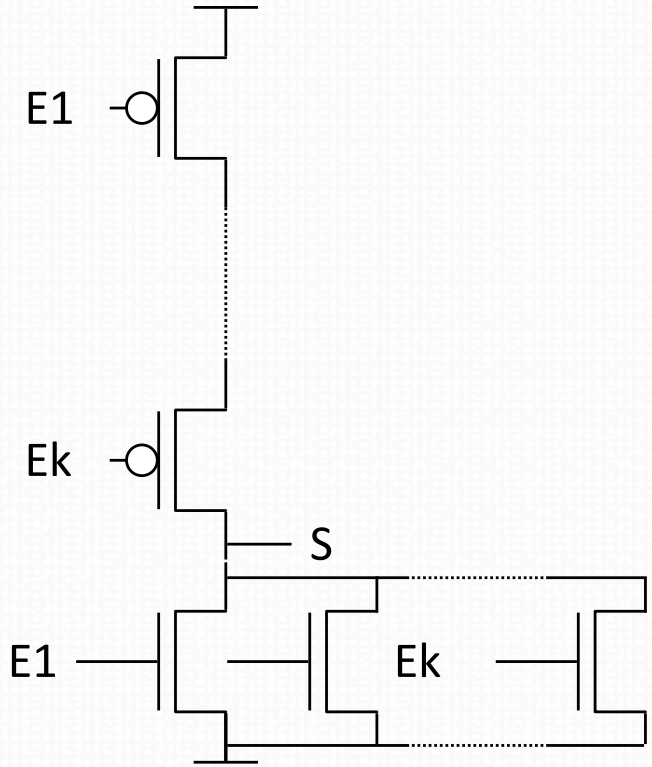
k-input NAND



$t_{plh} = Rp.C_L$

$t_{phl} = k.Rn.C_L$

k-input NOR



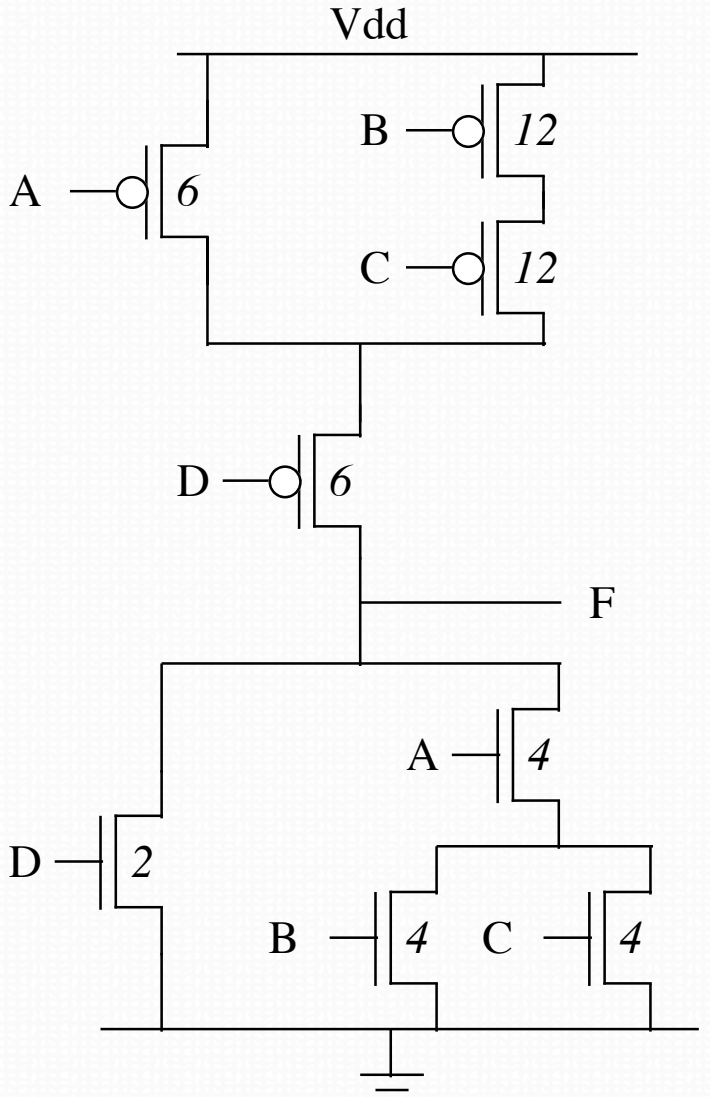
$t_{plh} = k.Rp.C_L$

$t_{phl} = Rn.C_L$

Transistor Sizing

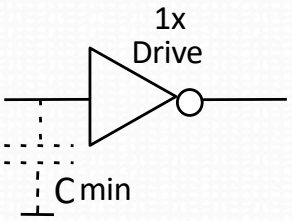
- $F = \overline{A.(B + C) + D}$

- The art of transistor sizing
  - Equilibrate delay for $0 \rightarrow 1$ and $1 \rightarrow 0$ output transitions
  - Minimize cell area

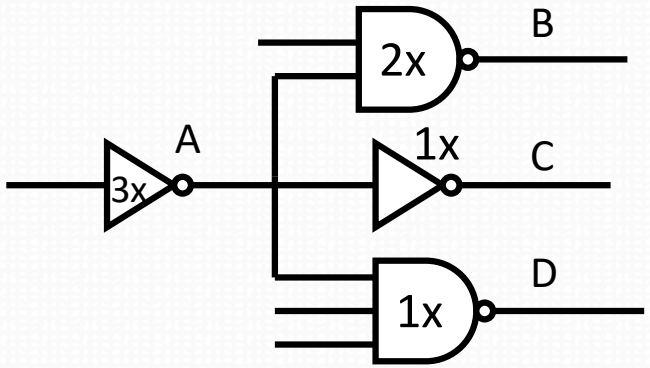


Logic-Level Delay Model

- 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

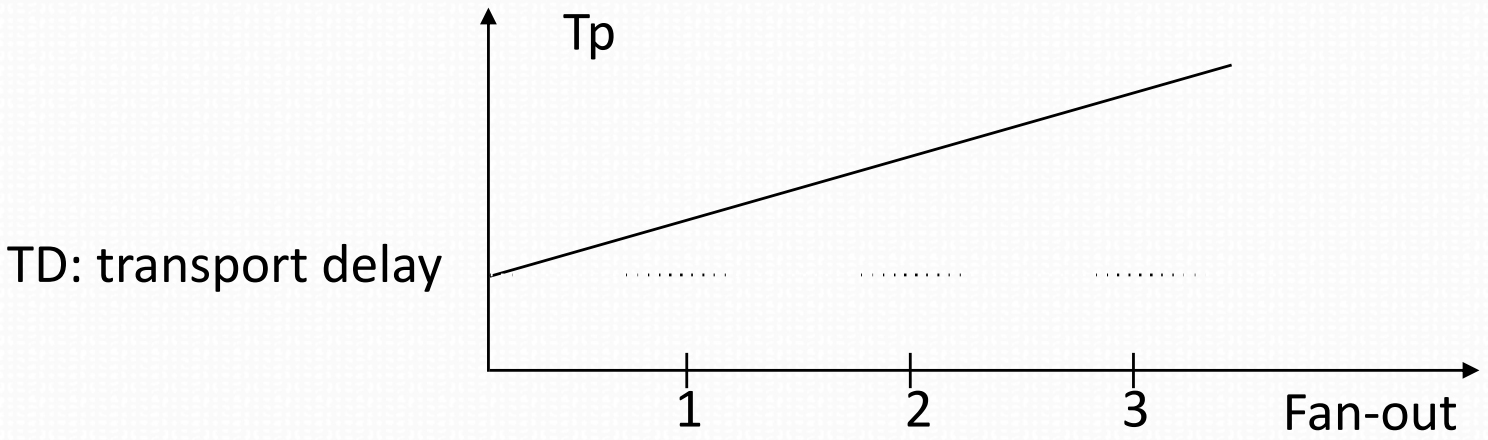


Porte	FIN	FOUT	RF
A	3	4	4/3
B	2		
C	1		
D	1		



Logic-Level Delay Model

- Tp = transport delay + inertial delay = TD + ID
- ID = RF.UD



Tp = TD + RF.UD

UD: unit delay

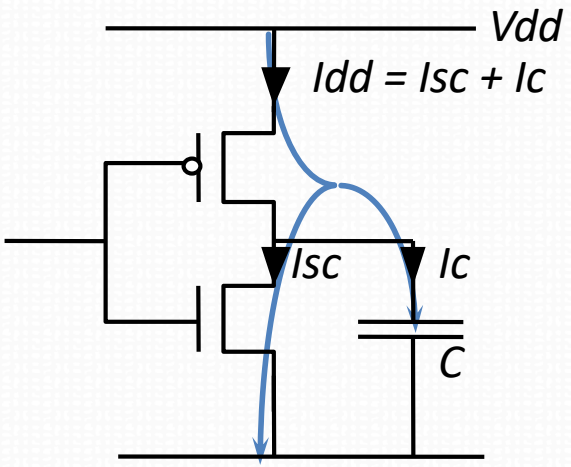
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

Power and Energy Consumption

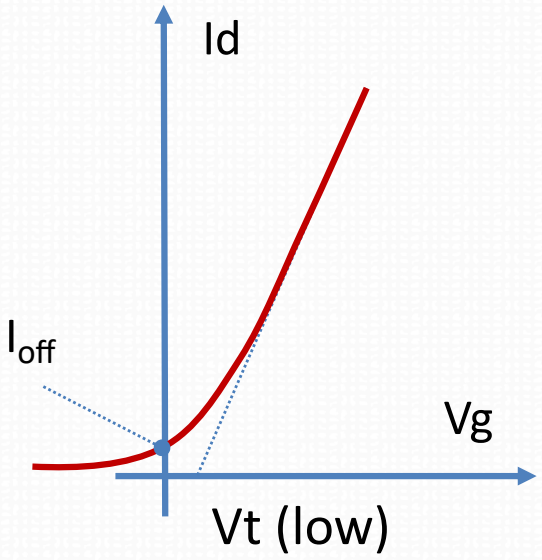
- Dynamic power
  - Charge and discharge of node capacitance
- Energy = $\mathbf{C.Vdd^2}$
- Power

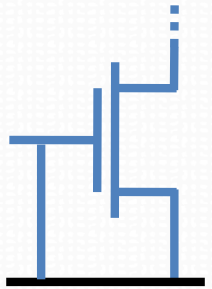
$$\mathbf{P_{dyn_i} = C.Vdd^2.f.Prob_{0\rightarrow 1}}$$



- Static power: *Ps*
  - Sub-threshold and junction leakage current

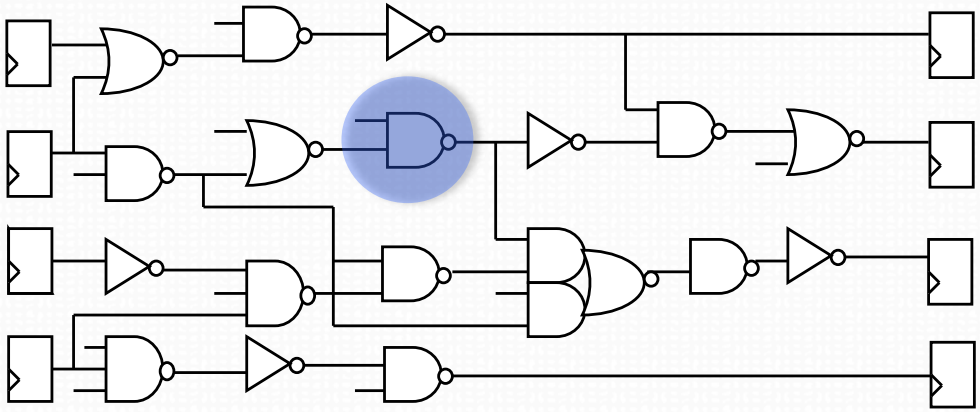
$$\mathbf{P_{stat_i} = N.I_{off}.Vdd}$$





Power at Higher Level

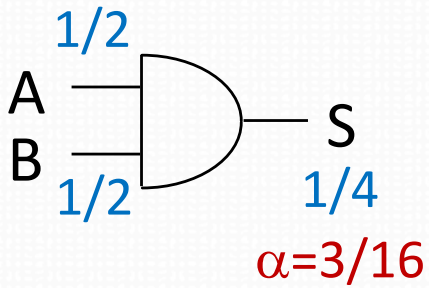
- Propagating activity



$$\mathbf{P} = \sum_{\mathbf{i}} \left[\alpha_{\mathbf{i}} . \mathbf{f_i} . \mathbf{C_i} . \mathbf{Vdd^2} + \mathbf{I_{leak_i}} . \mathbf{Vdd} \right]$$

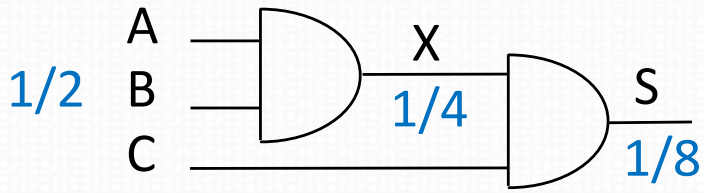
Activity

- Activity α_i is the probability to have a *0→1* transitions at the output of a gate
- Example: AND gate
  - $P_S = P(S=1) = P_A P_B$
  - $\alpha_i = P_S(1- P_S)$



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

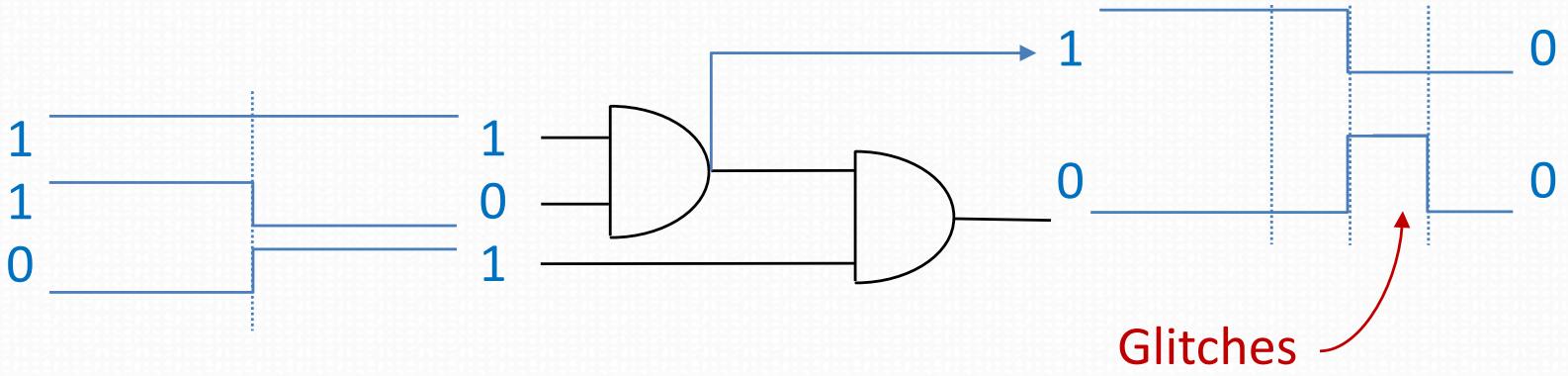
- Activity propagation



Propagating Activity is not So Simple

- Conditional probabilities

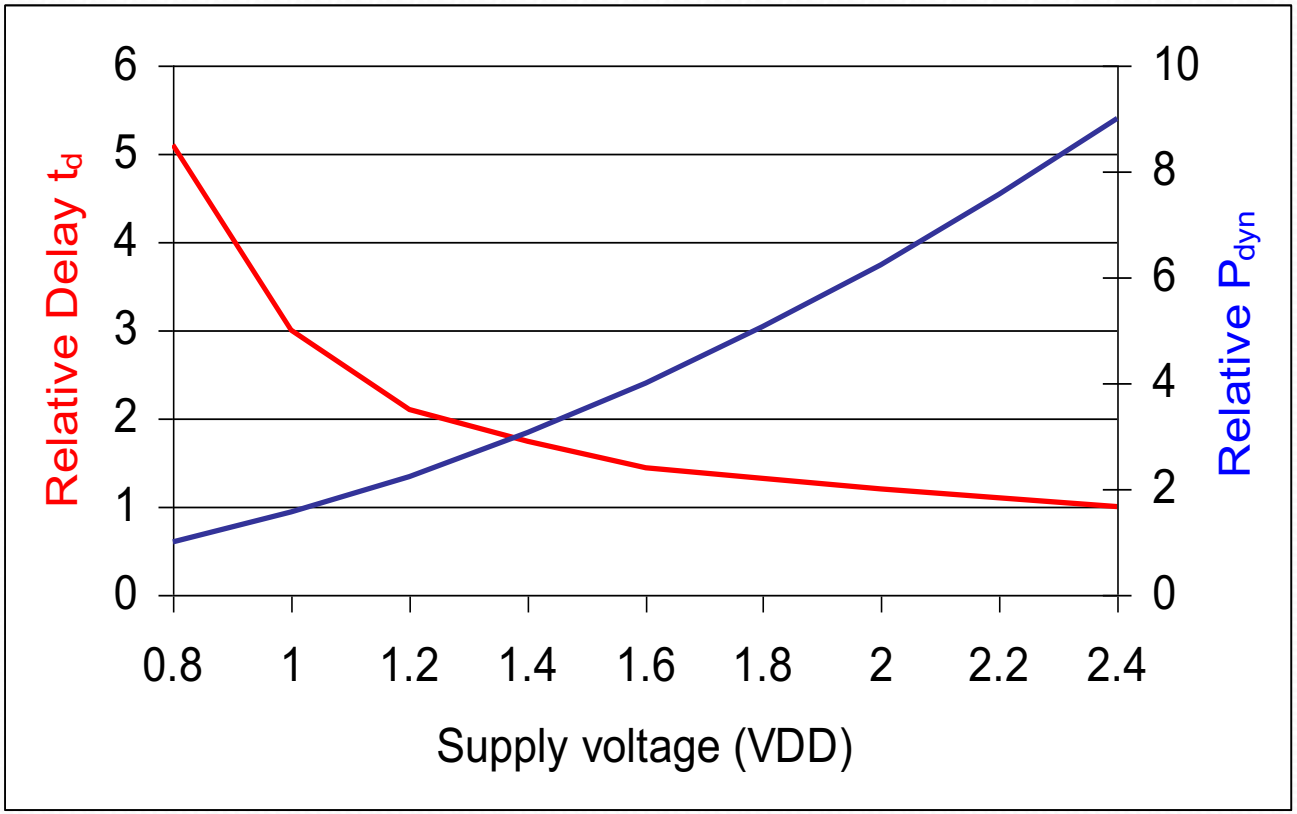
- Glitches: gate delay
  - Significant in arithmetic



Dynamic Power vs. Performance

- Decreasing Vdd reduces power but increases delay

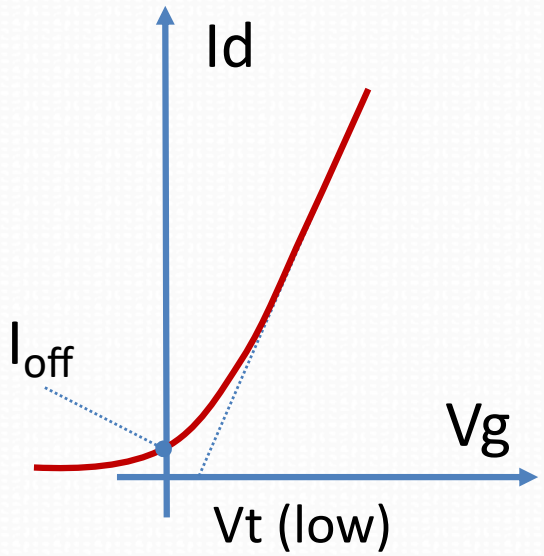
$$\mathbf{P_{dyn_i} = \alpha_i . f_{clk} . C_i . Vdd^2}$$



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

Leakage vs. performance

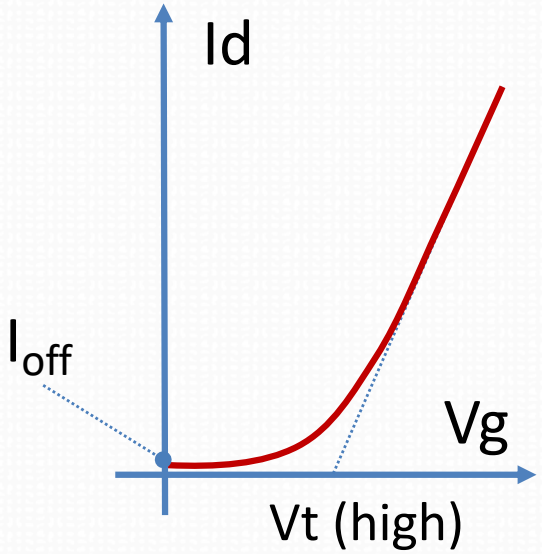
- High performance



$$\mathbf{P_{stat_i} = N.I_{off}.Vdd}$$

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

- Low leakage

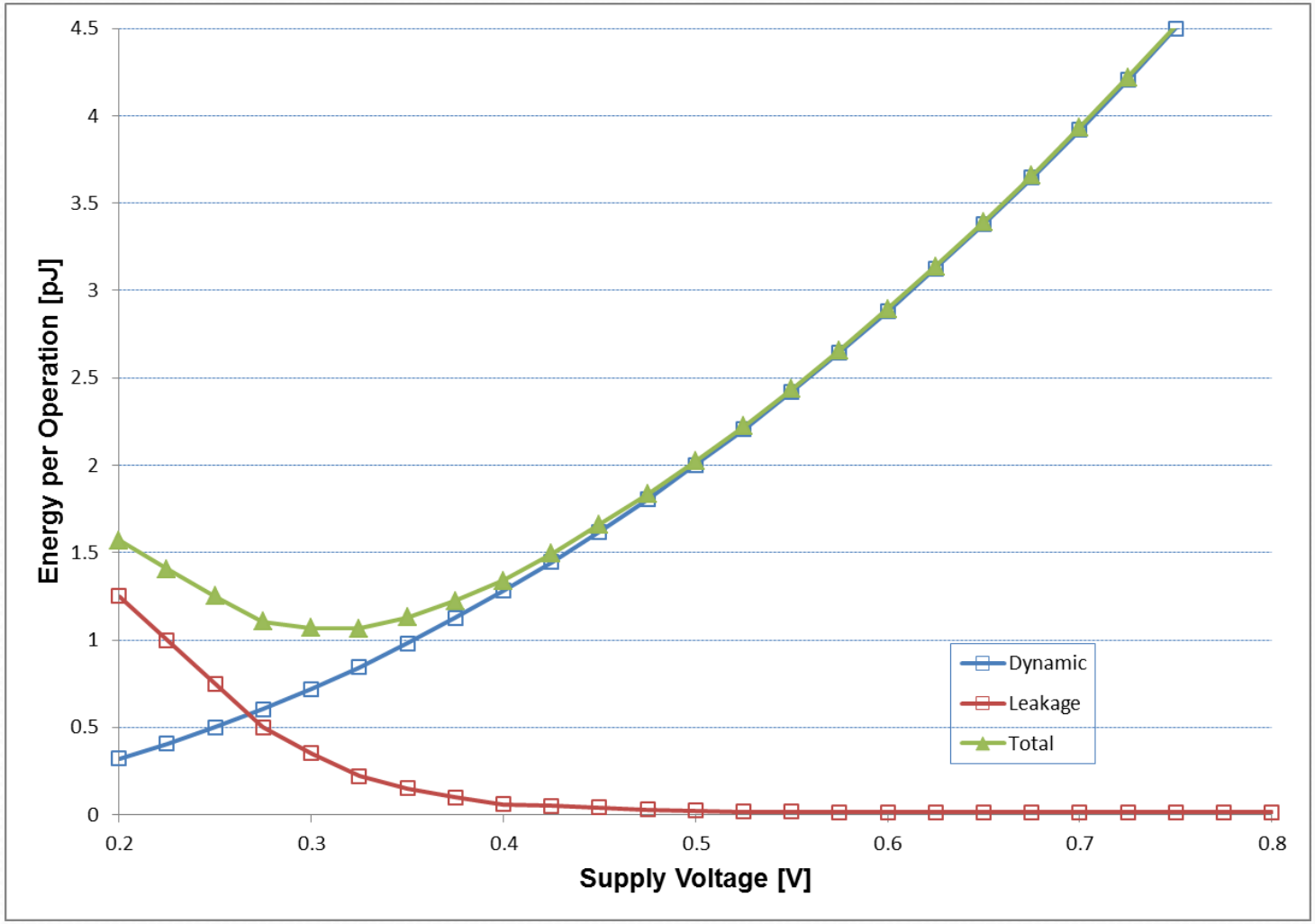


Ioff:

- Exponential in inverse of Vt
- Exponential in temperature
- Linear in device count

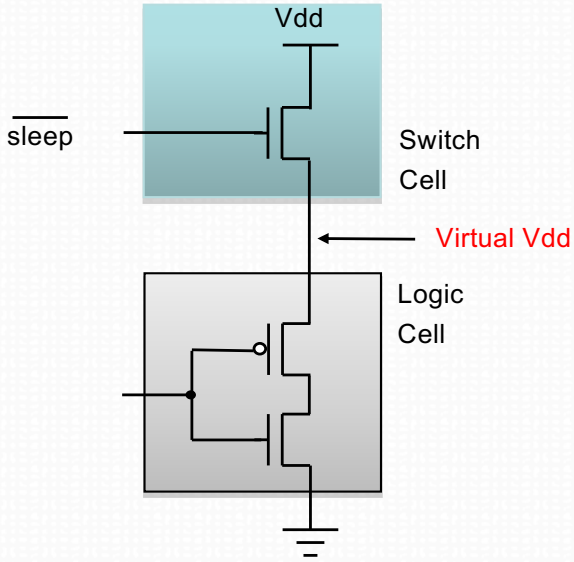
Minimum Energy per Operation

- Putting all together



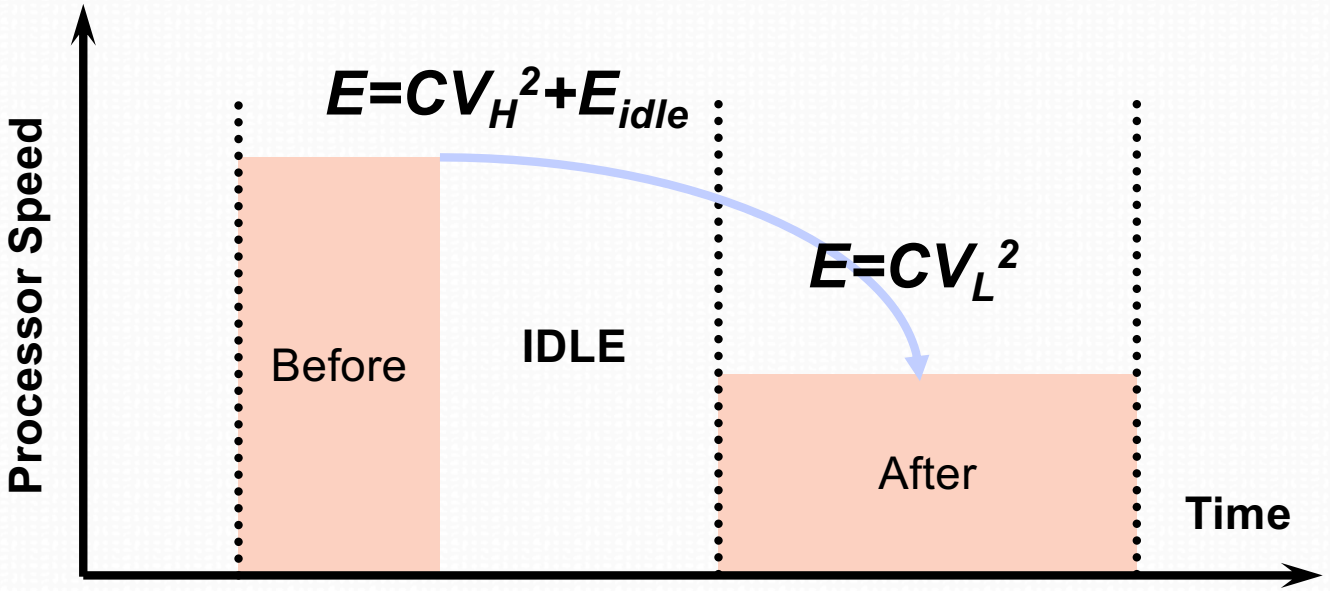
Reducing Power

- 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



Conclusion: Power in CMOS

$$\mathbf{P} = \sum_{\mathbf{i}} \left[\alpha_{\mathbf{i}} . \mathbf{f_i} . \mathbf{C_i} . \mathbf{Vdd^2} + \mathbf{I_{leak_i}} . \mathbf{Vdd} \right]$$

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

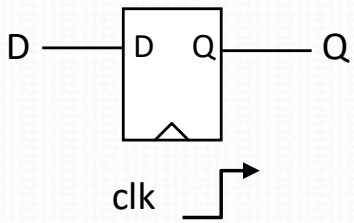
- Dynamic power
  - 40-70% today
  - Decreasing relatively
  - DVFS becomes more and more difficult
- Leakage power
  - 20-50 % today
  - Increasing rapidly
    - number of transistors
    - Vdd/Vt scaling
  - Critical for memory

Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

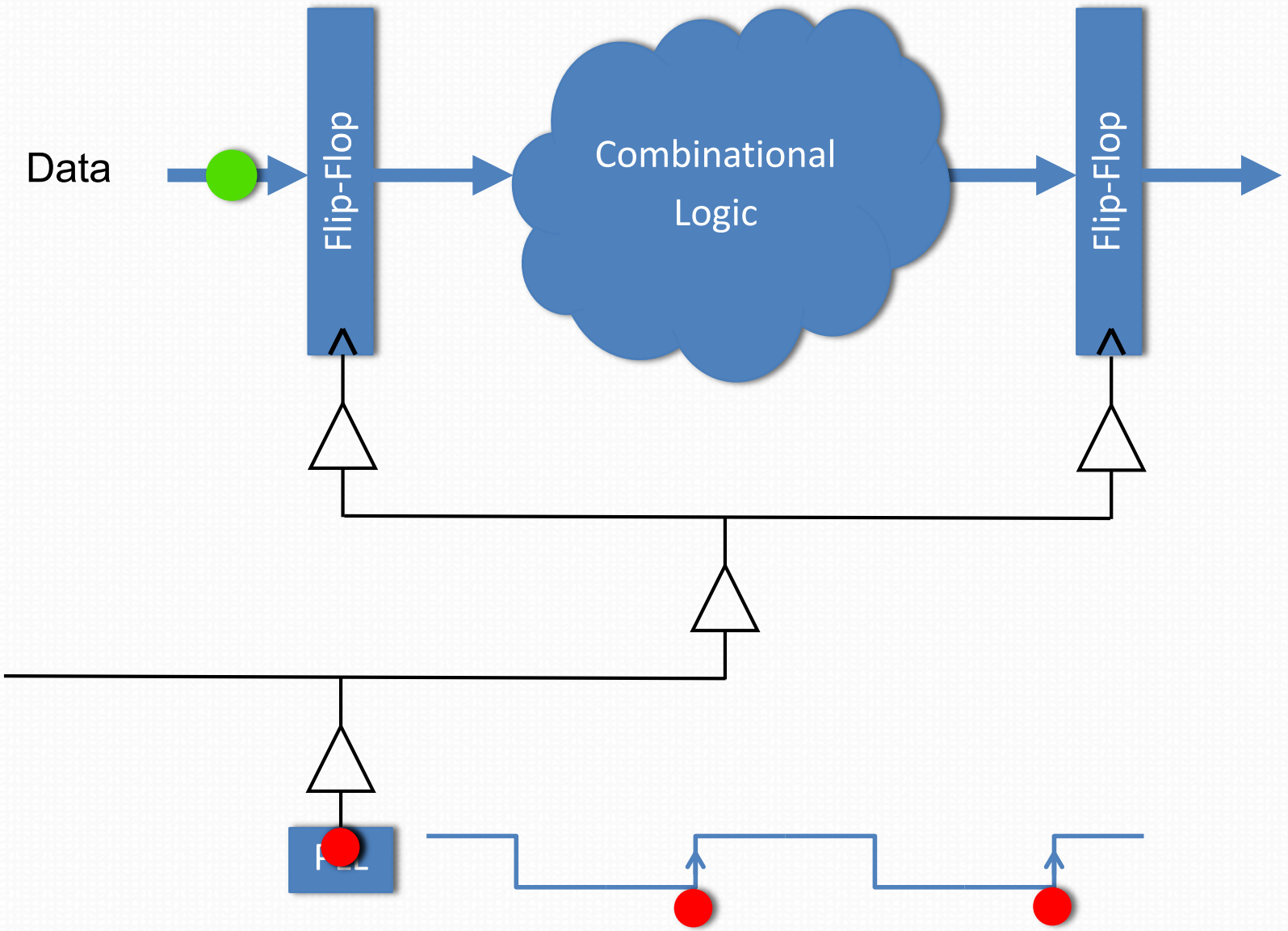
Timing Parameters

- D Flip-Flop

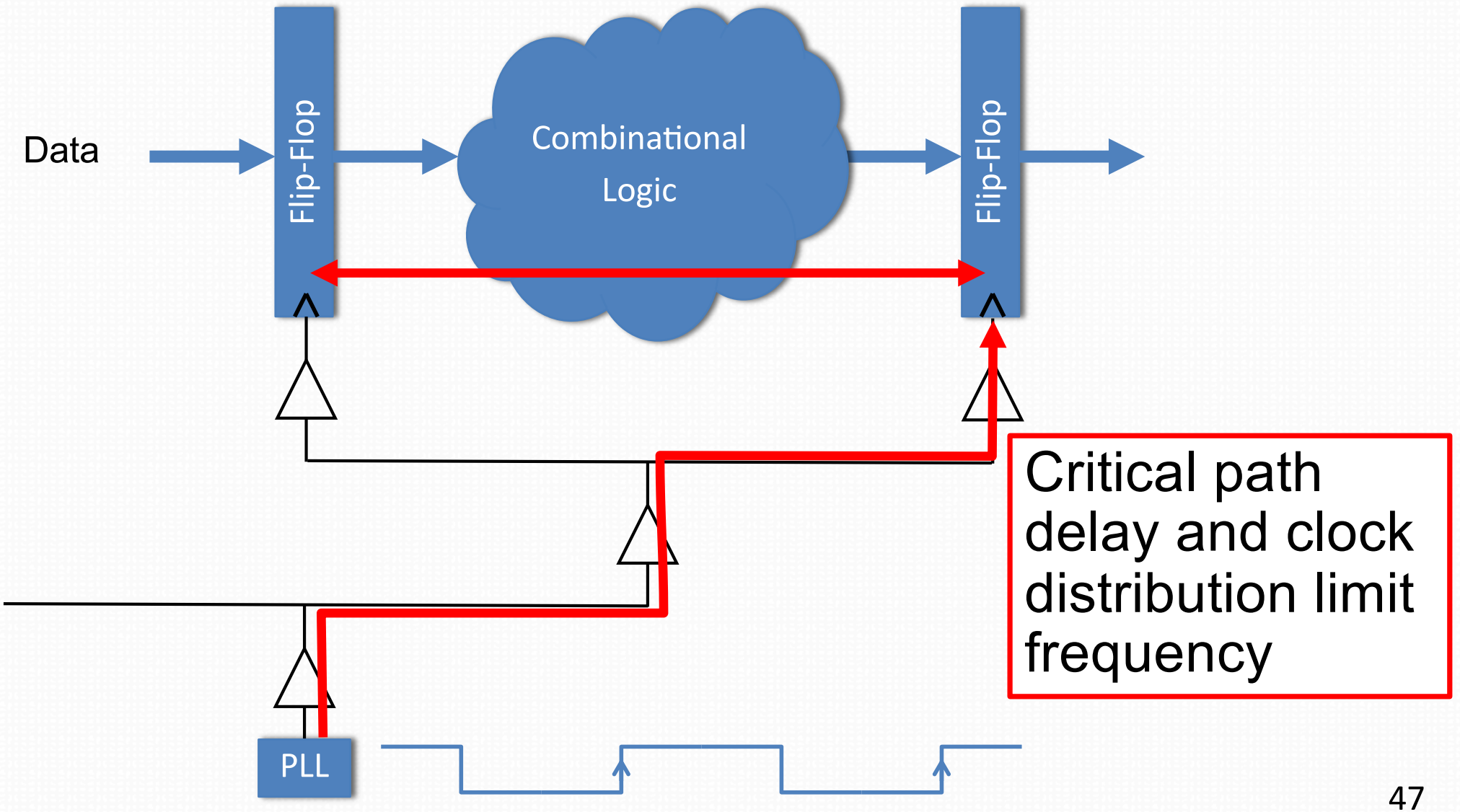


  - Setup Time: Tsetup
    - amount of time the data at the synchronous input (D) must be stable before the active edge of clock
  - Hold Time: Thold
    - amount of time the data at the synchronous input (D) must be stable after the active edge of clock
  - Propagation Time: Tp
    - amount of time for the data at the output (Q) to change after the active edge of clock

Synchronous Circuits



Synchronous Circuits



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
- Tcp: critical path delay of the logic

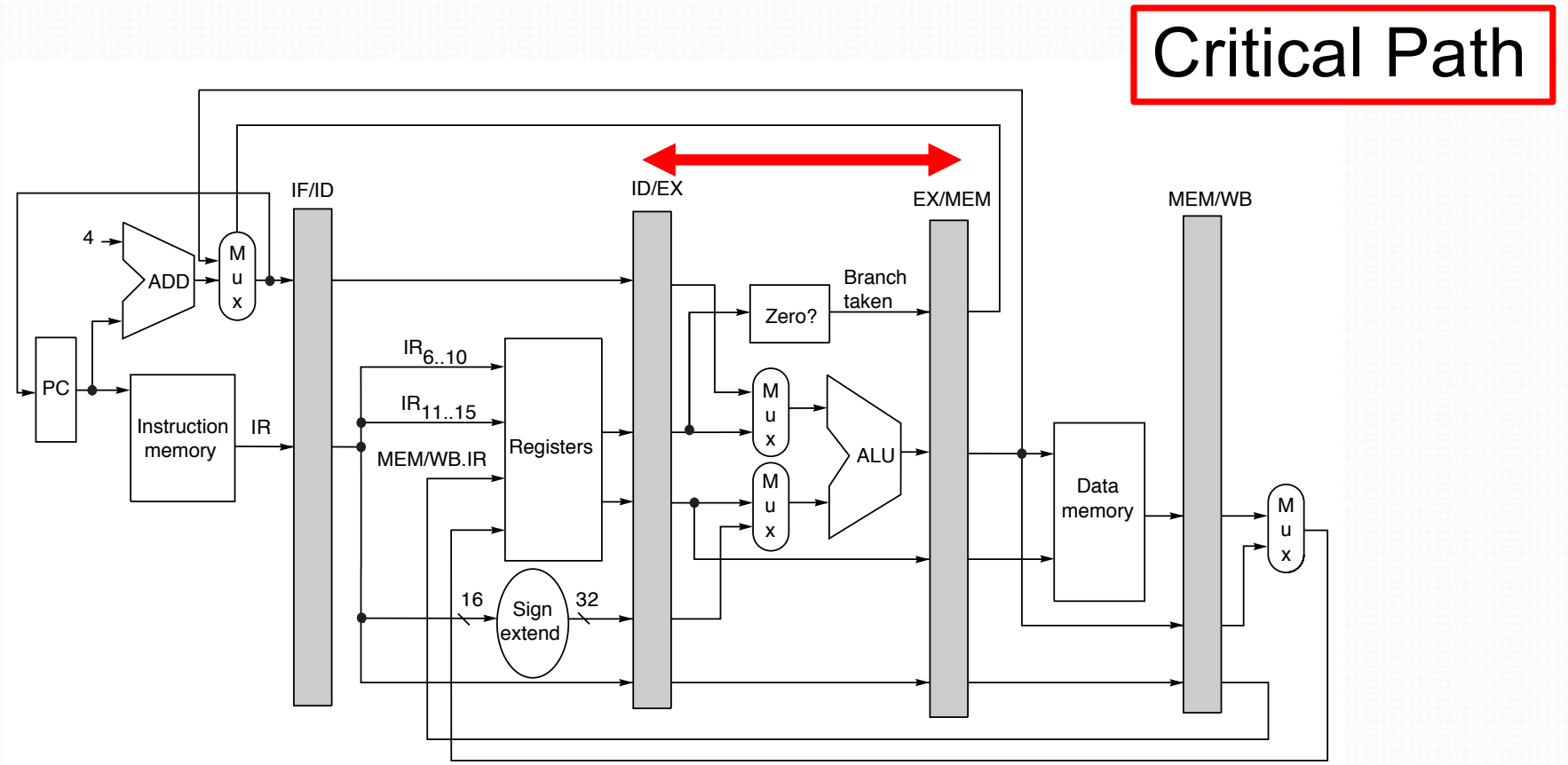
$$Tcp = \underset{\forall i}{MAX}(D_i), \text{ with } D_i \text{ Delay of path } i$$

- Maximal Frequency

$$Fclk_{\max} = \frac{1}{Tcp + Tp + Tsetup}$$

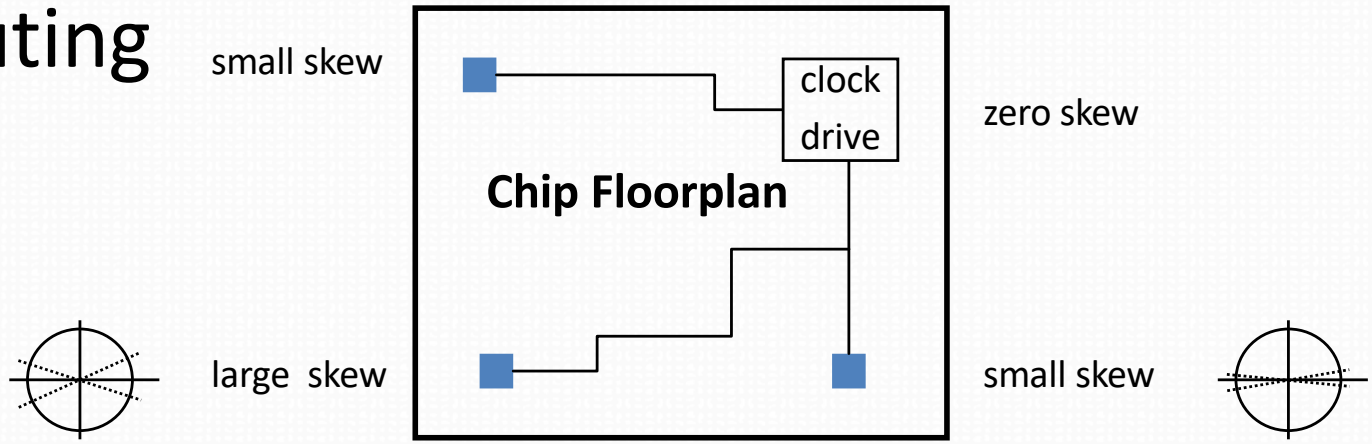
Critical Path in Processor Pipelines

- A typical (yet simple) processor pipeline



Clock Skew

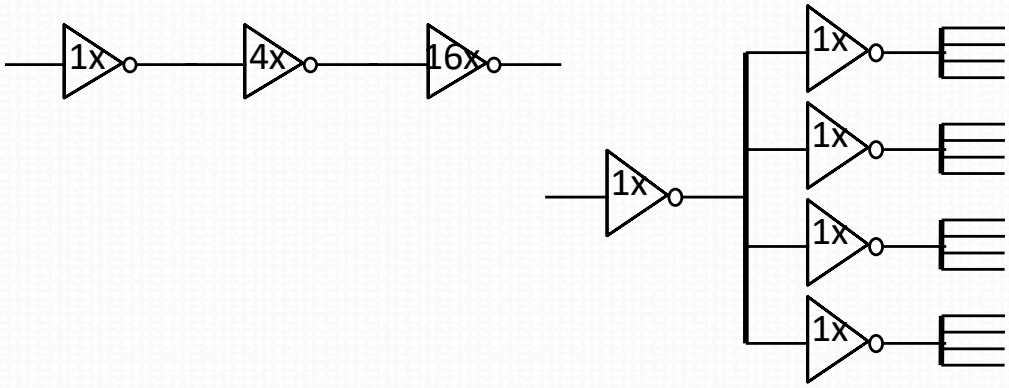
- Every FF receives the clock edge at a different time
- Clock routing



- Reduction of maximal frequency: $Fe_{max} = \frac{1}{Tcc + Tp + Tsetup + \delta}$
- Maximal skew for circuit operation: $\delta < t_{P1} + MIN(t_{comb}) - t_{hold}$
  - Race between data and clock

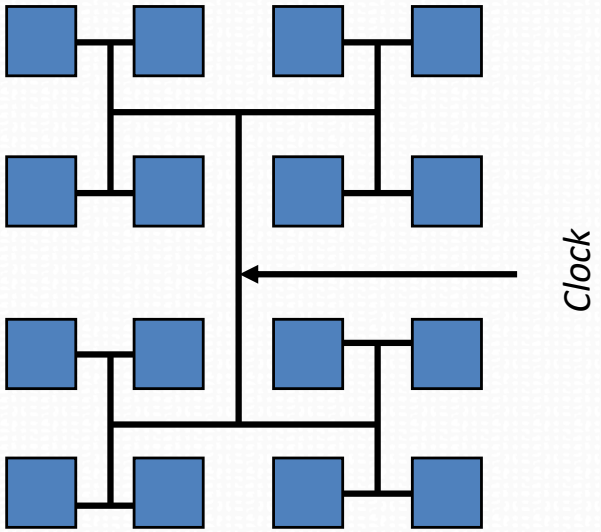
Clock Distribution

- Geometric buffering

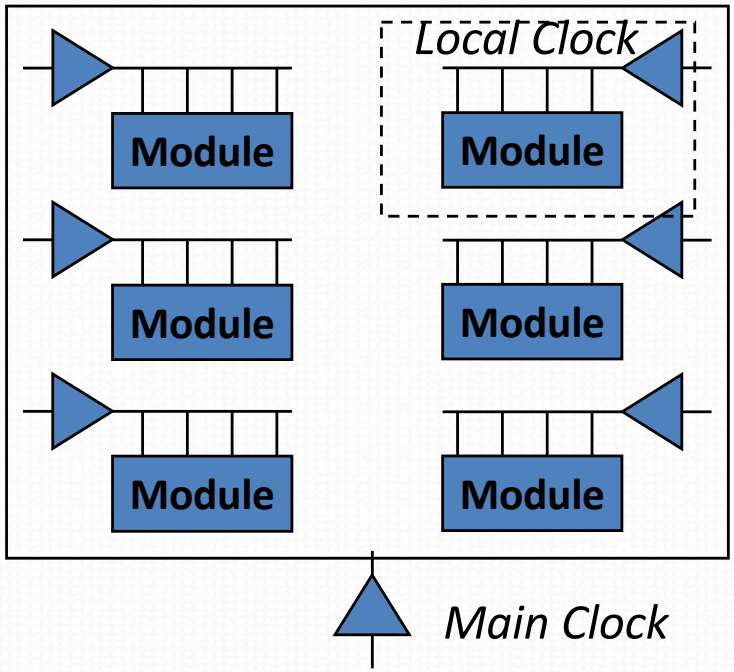


- Tree-based

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



Buffering: local reduction of skew



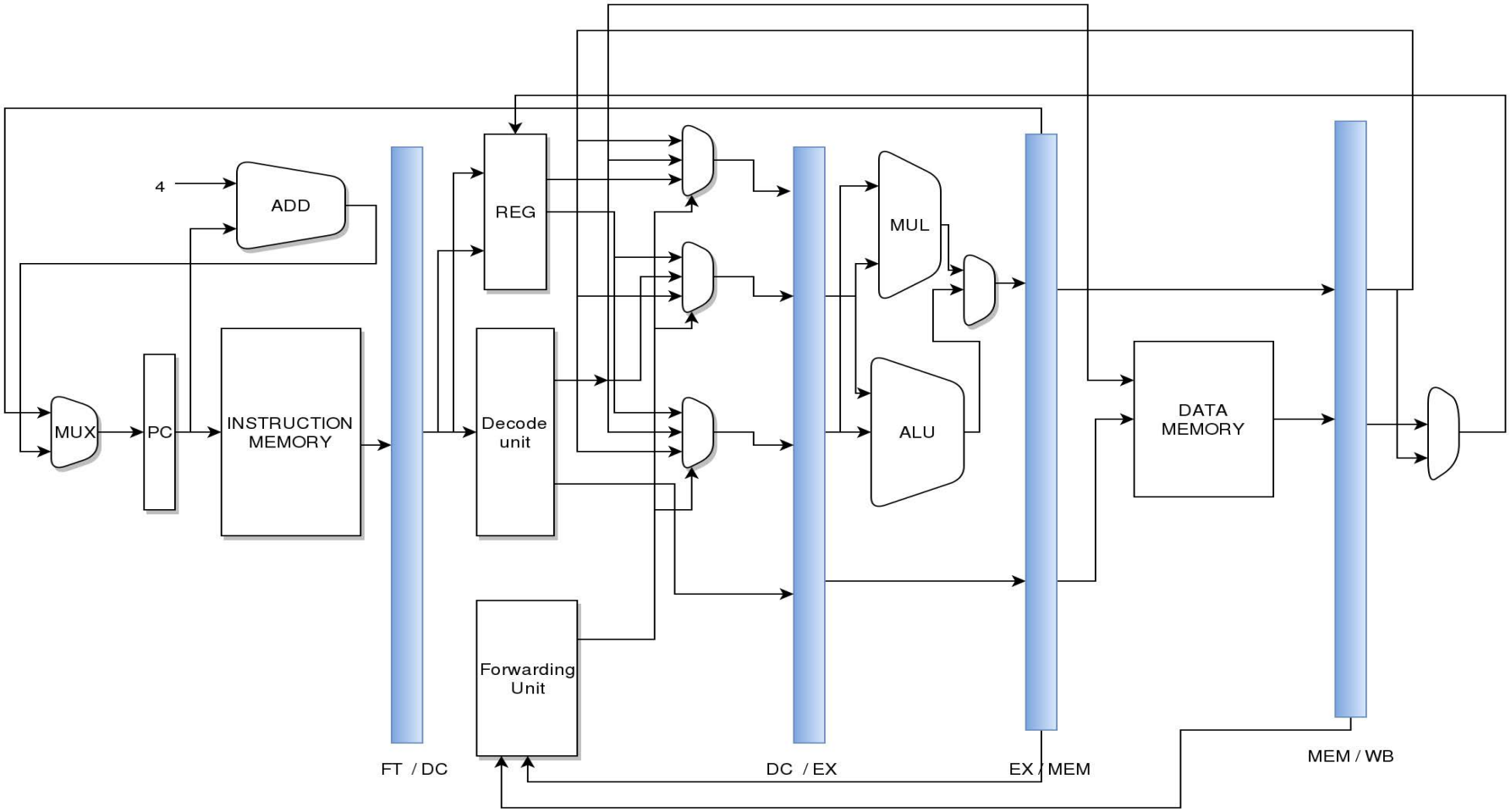
Outline

- The Fundamental Element: MOSFET Transistor
- Design of CMOS Cells: Combinatorial Logic
- Memory Cells
- Delay
- Power Consumption
- Synchronous Design
- Multicore: power and utilization walls
- Hardware accelerators

Inside (Simple) Processor Architecture

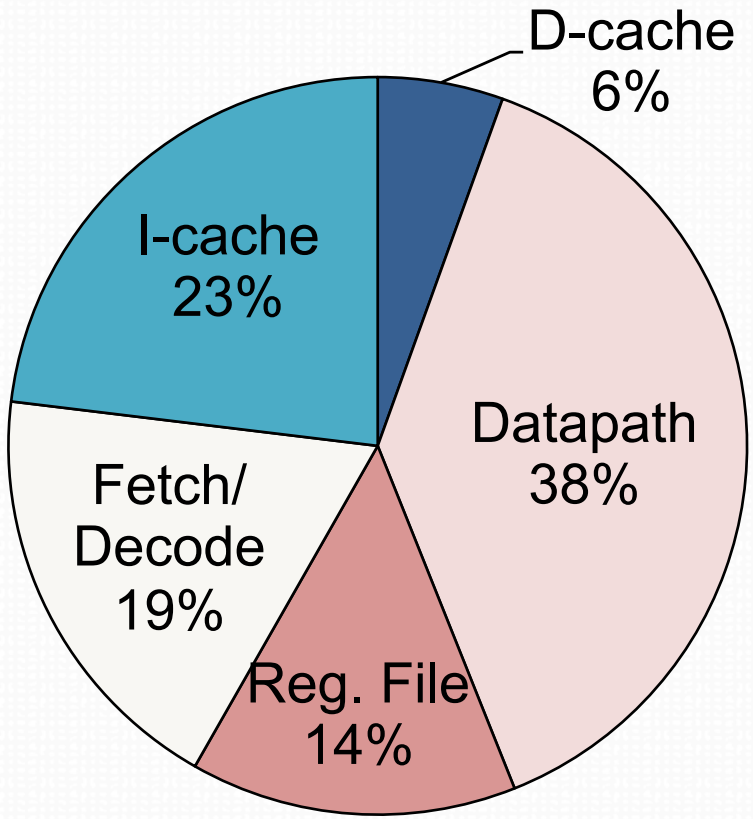
Microarchitecture Pipeline

- Microarchitecture defines how instructions are executed (not unique)



Energy Cost in a Processor

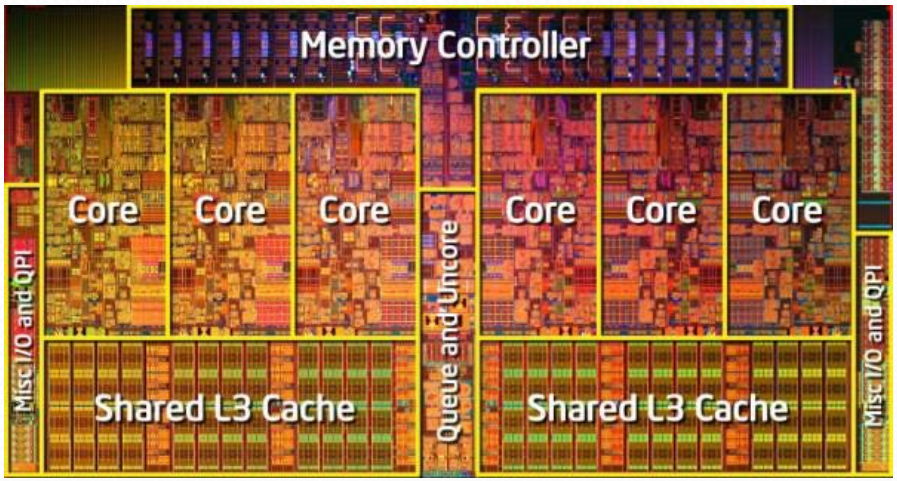
- Operation:
  - 32-bit addition: 0.05pJ
  - 16-bit multiply: 0.25pJ
  - 64-bit FPU: 20pJ/op
- Instruction:
  - fetch, decode, read 2 operands from RF, execute, write back



MIPS Proc.
91 pJ/instr.

Achieving Higher Performance

- Pushing clock frequency…
- Branch/value prediction
- Cache memory
- In-core parallelism
  - Multiple FUs
  - Out of order execution
  - VLIW+good compilers

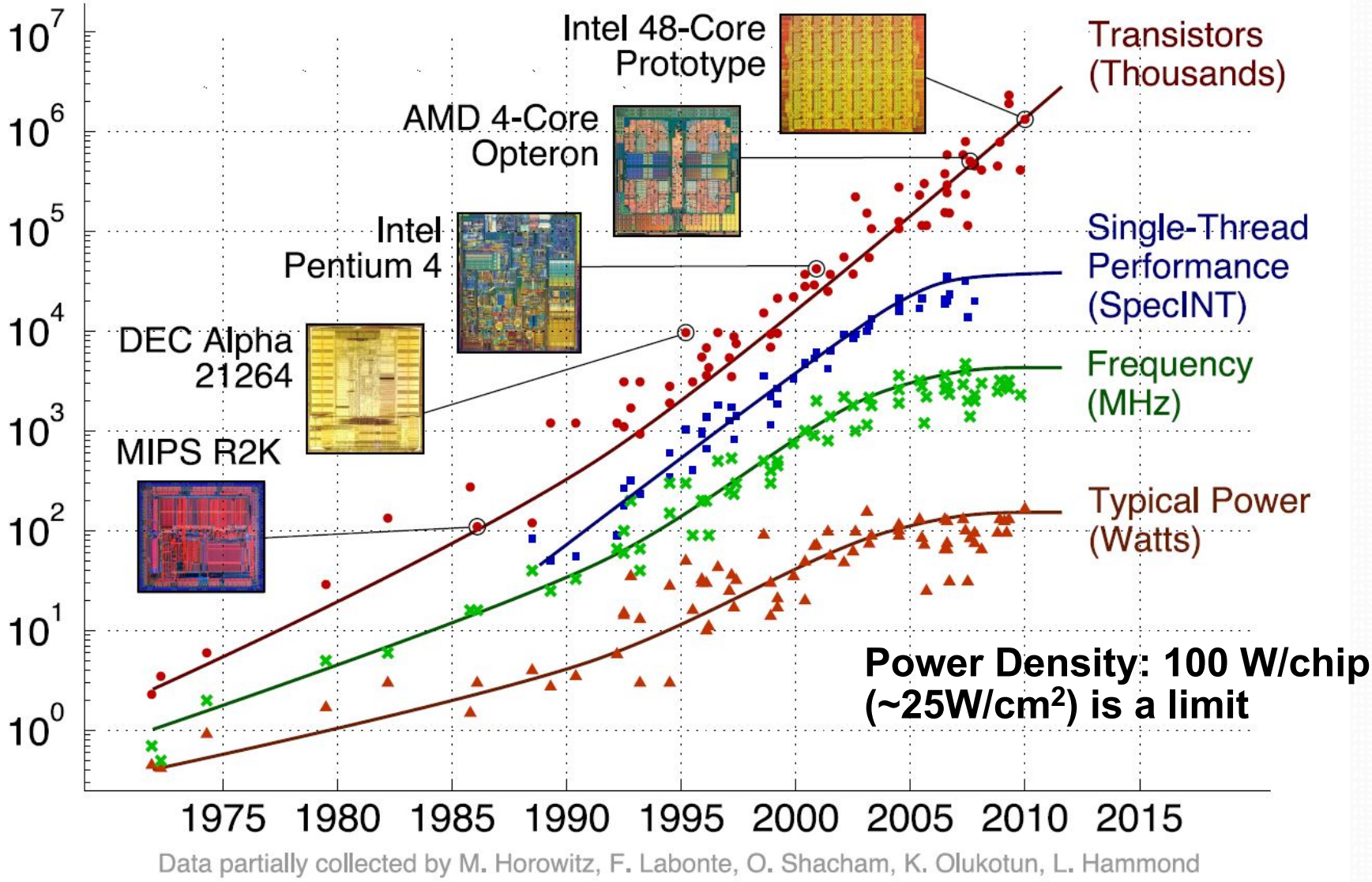


- Multiple cores on a single chip

Multicore: it's all a trick!

Power and Utilization Walls

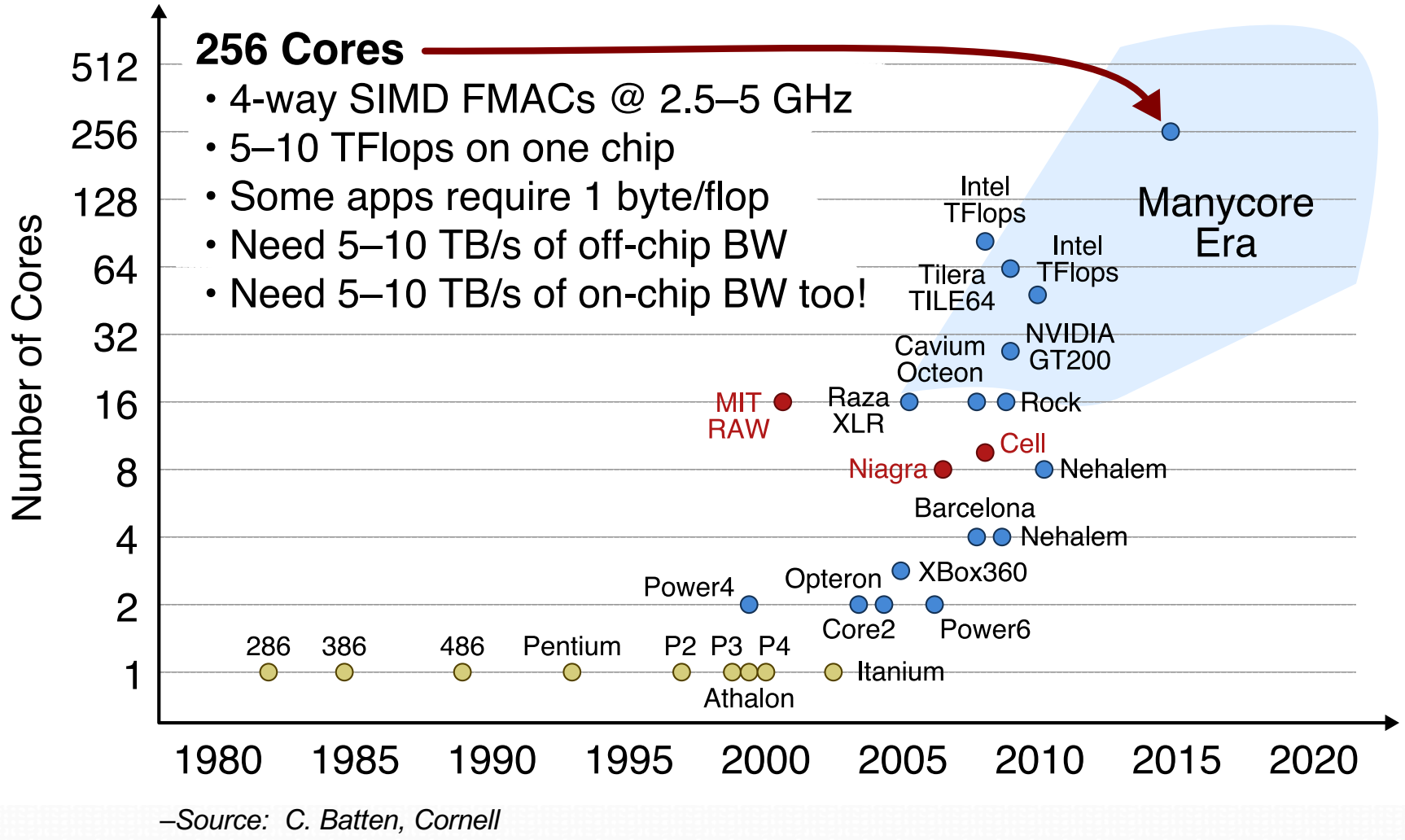
And then came the "Power Wall"



Source: C. Batten, Cornell

and the "Multicore Era"

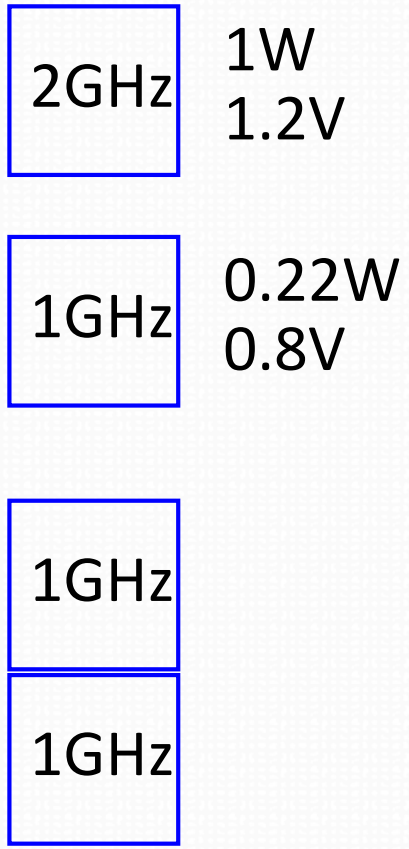
- Increasing performance by increasing # of cores



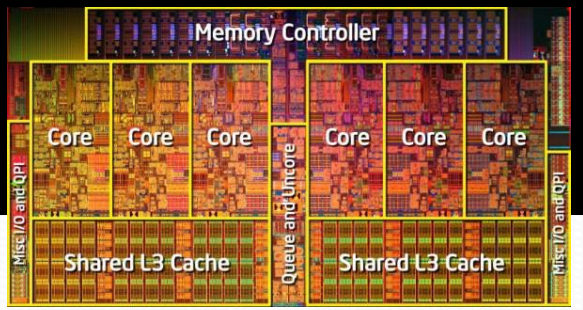
–Source: C. Batten, Cornell

Moving to multicore

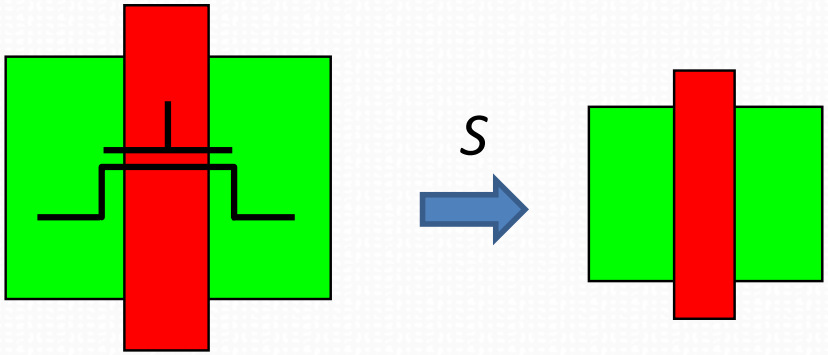
- 1 core@2GHz@1.2V@1W
- 1 core@1GHz@0.8V@0.25W
- 2 cores@1GHz@0.8V@0.5W
- But… twice area (and not so simple)
- Advanced technology nodes?

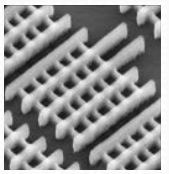


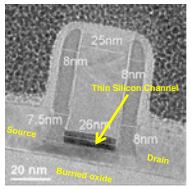
Technology Scaling



Intel's Xeon Chip

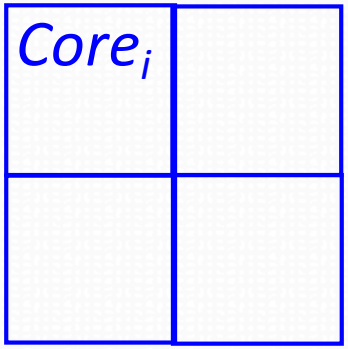




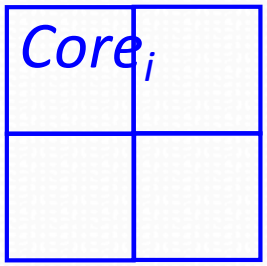


Classical (Dennard's) scaling

Device count	S^2
Device frequency	S
Capacitance, Vdd	1/S
Device power	$1/S^2$
Utilization	**1**



100W@*f*



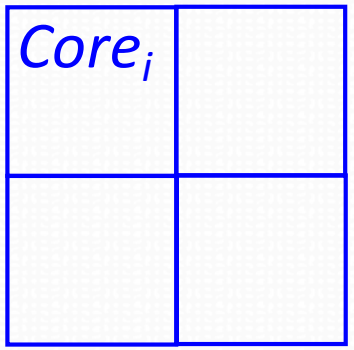
50W@*1.4.f*

End of Dennard's Scaling

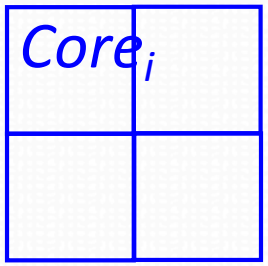
- Energy efficiency is not scaling along with integration capacity

Leakage limited scaling

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

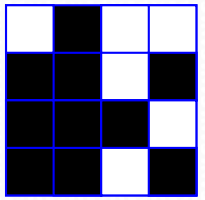


100W@*f*

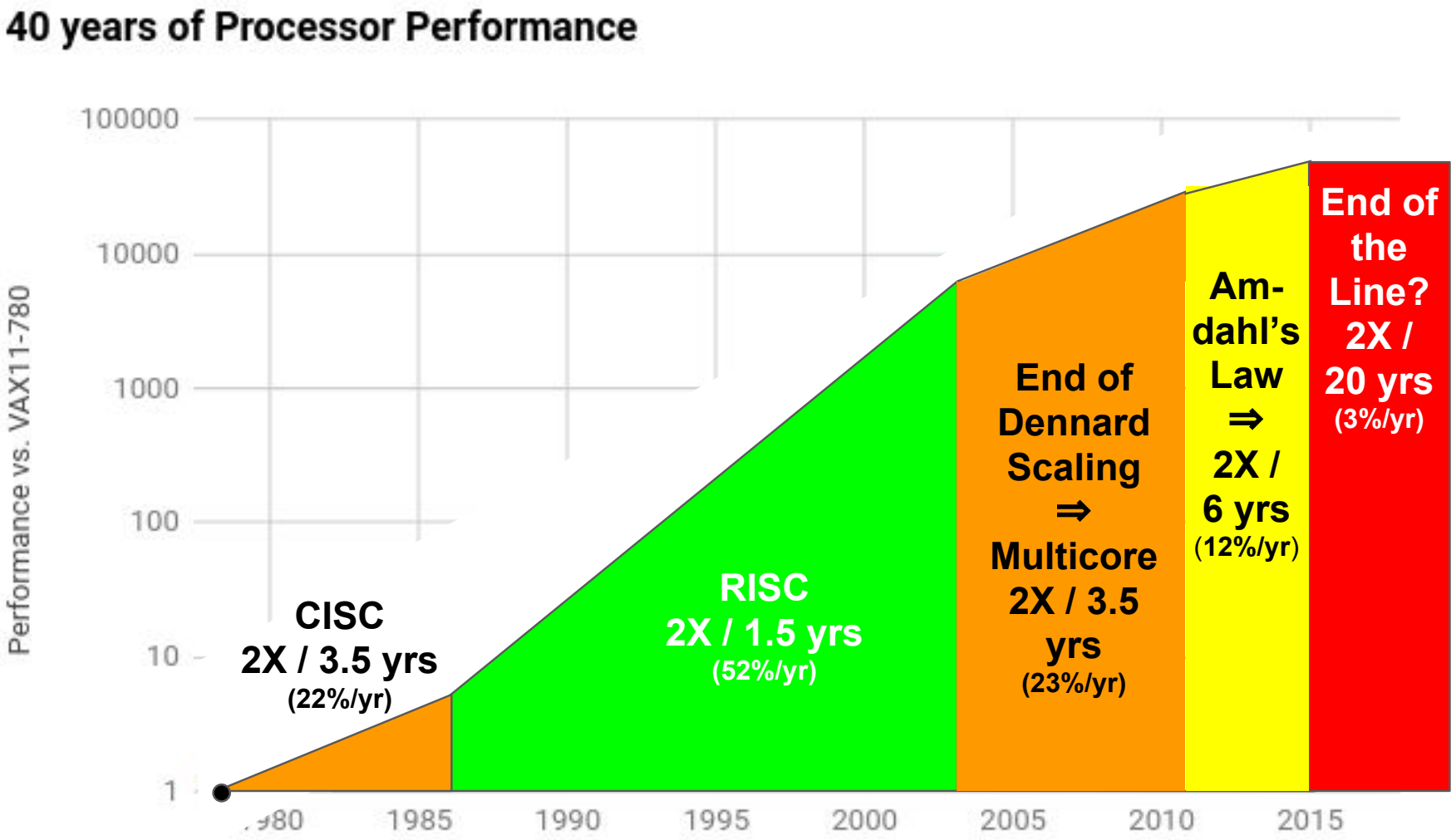


100W@*1.4.f*
(w/o) leakage

- Utilization Wall: percentage of a chip that can switch at full frequency drops exponentially
- Replace dark cores with specialized cores (10-100x more energy efficient)



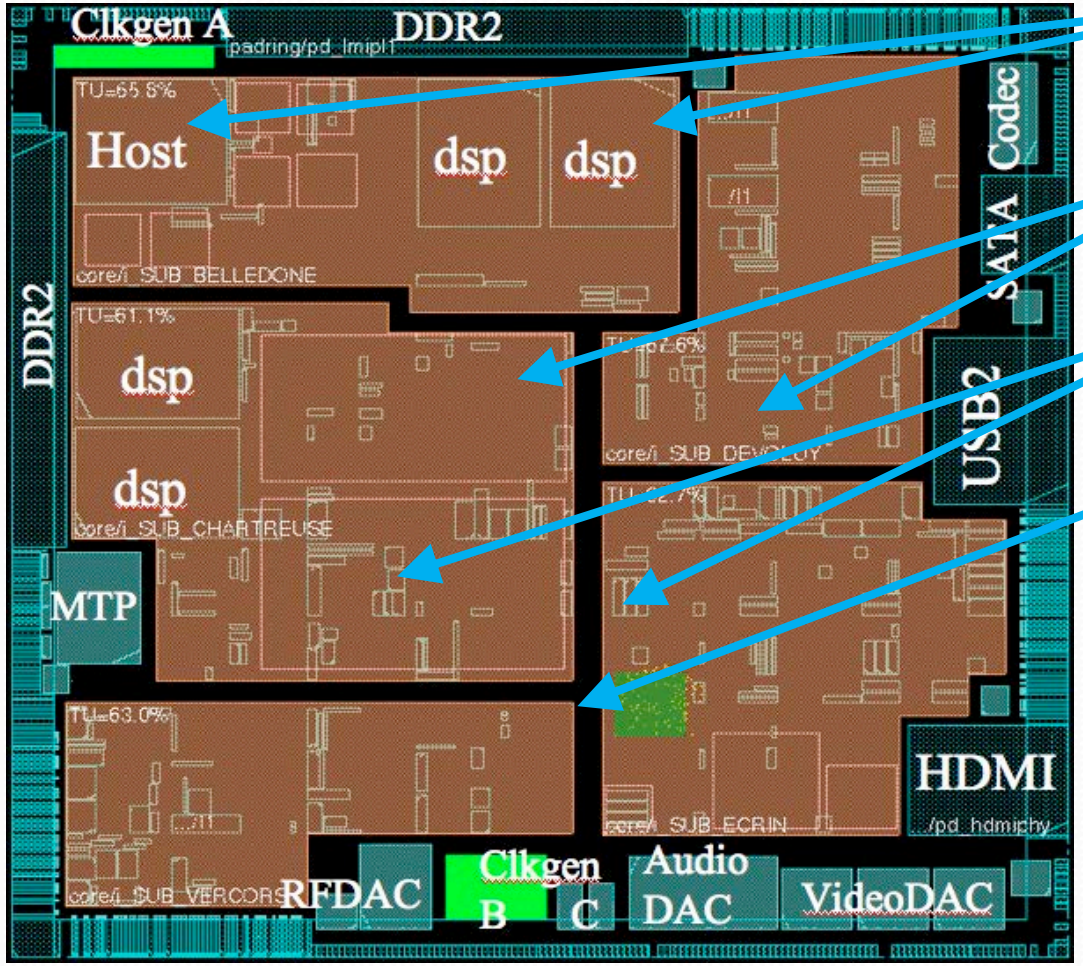
End of Growth of Speed?



Based on SPECintCPU. Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

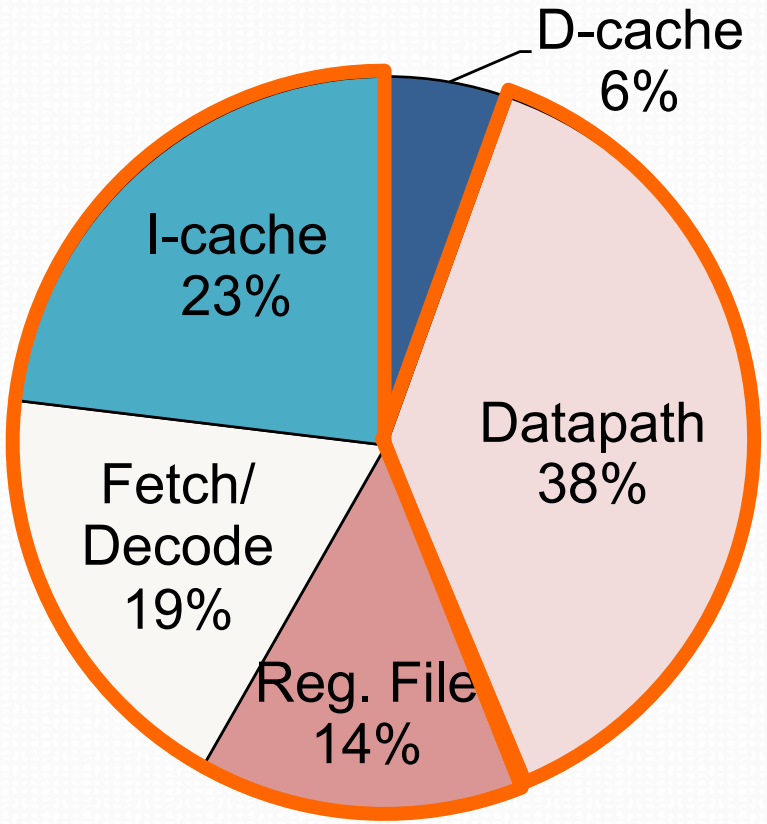
Pushing the Accelerator!

What is a HW accelerator?

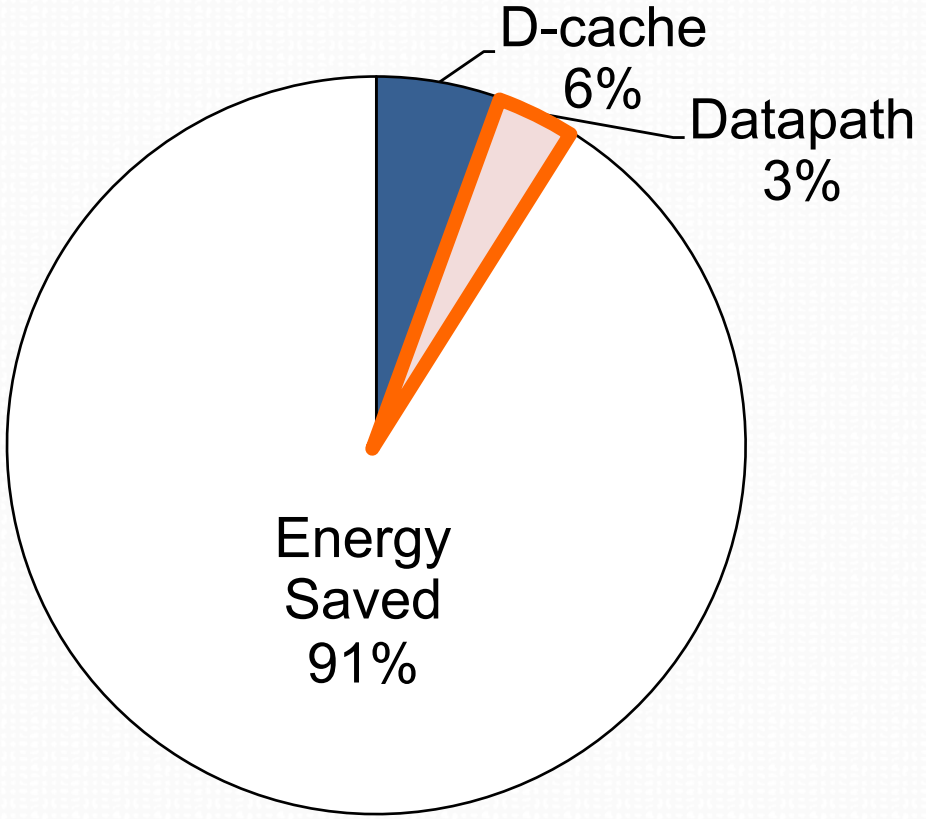


- 16 processors
- 38 HW blocks
- 140 memory blocks
- 5 Gbytes/s on-chip interconnection network

Energy Savings in Specialized HW



MIPS baseline
91 pJ/instr.



Specialized core
8 pJ/instr.

[Goulding et al., Hot Chips'10]

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

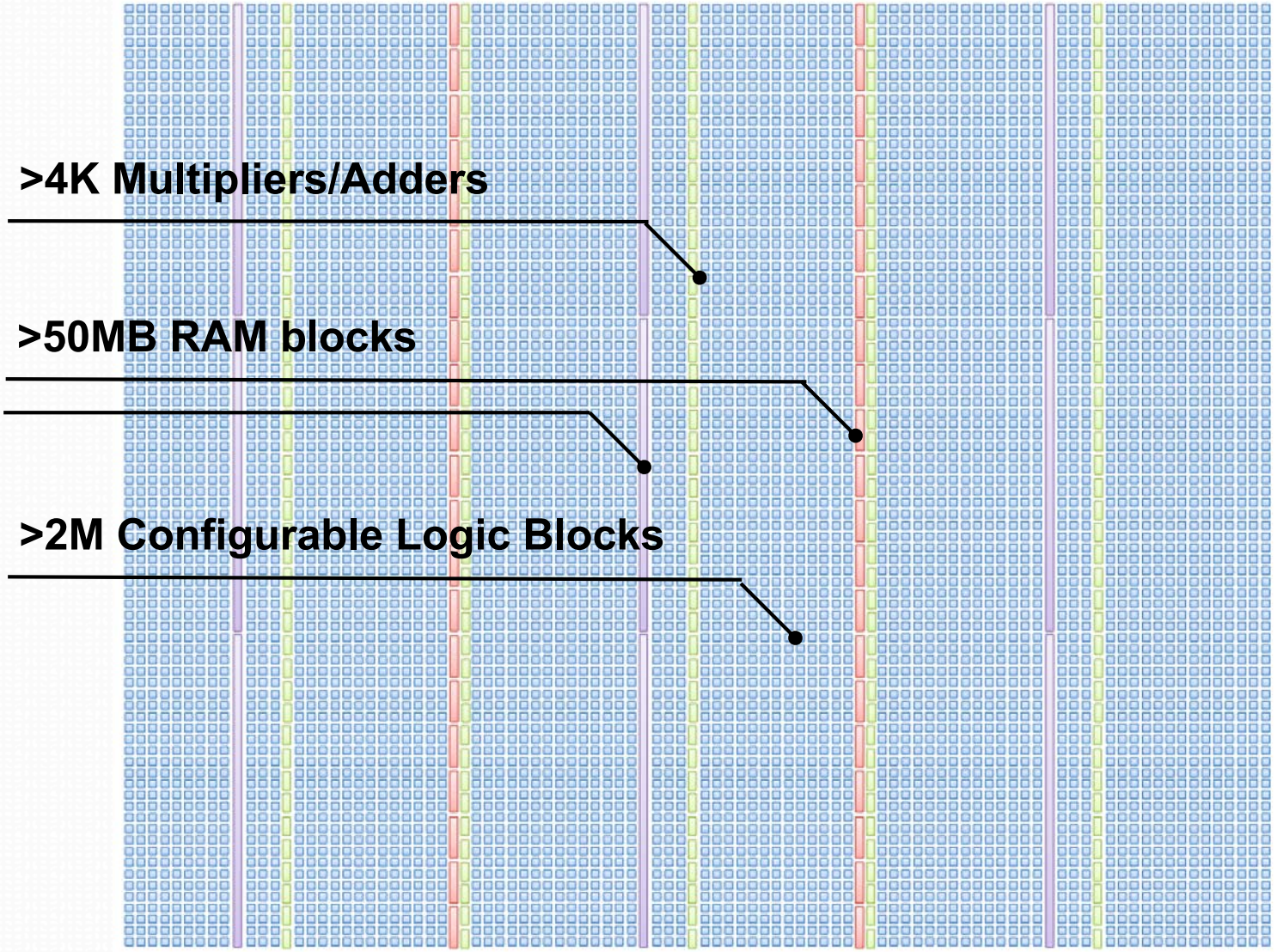
Making ANN Inference more Efficient

- Main motivation: AlphaGo consumes around 250,000 Watts!
- Bring Logic and Memory closer
- Compute less precisely

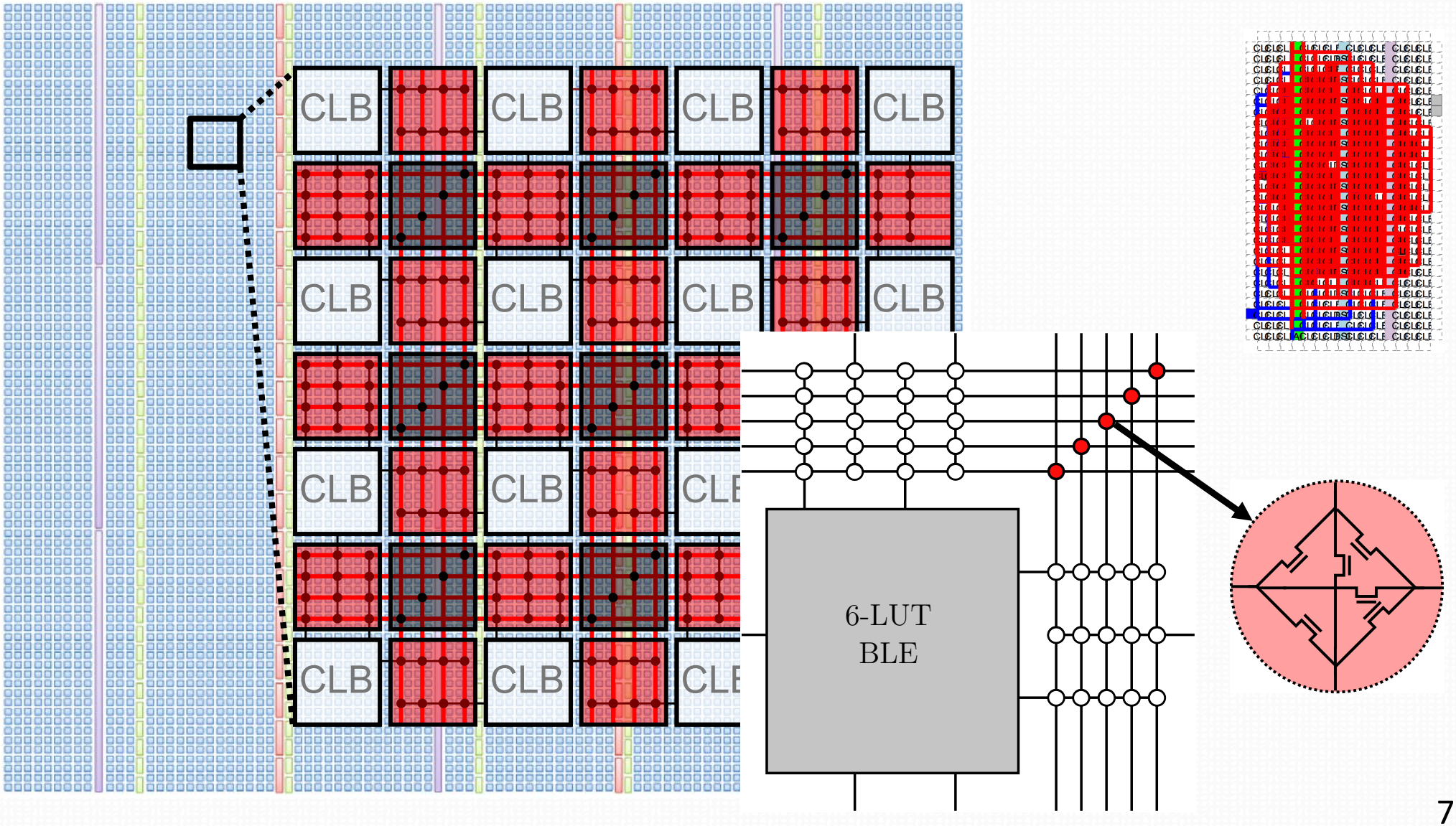
- Google Tensor Processing Units (TPU)
  - Computations close to memory
  - 8 bit operations

Reconfigurable Hardware Accelerators

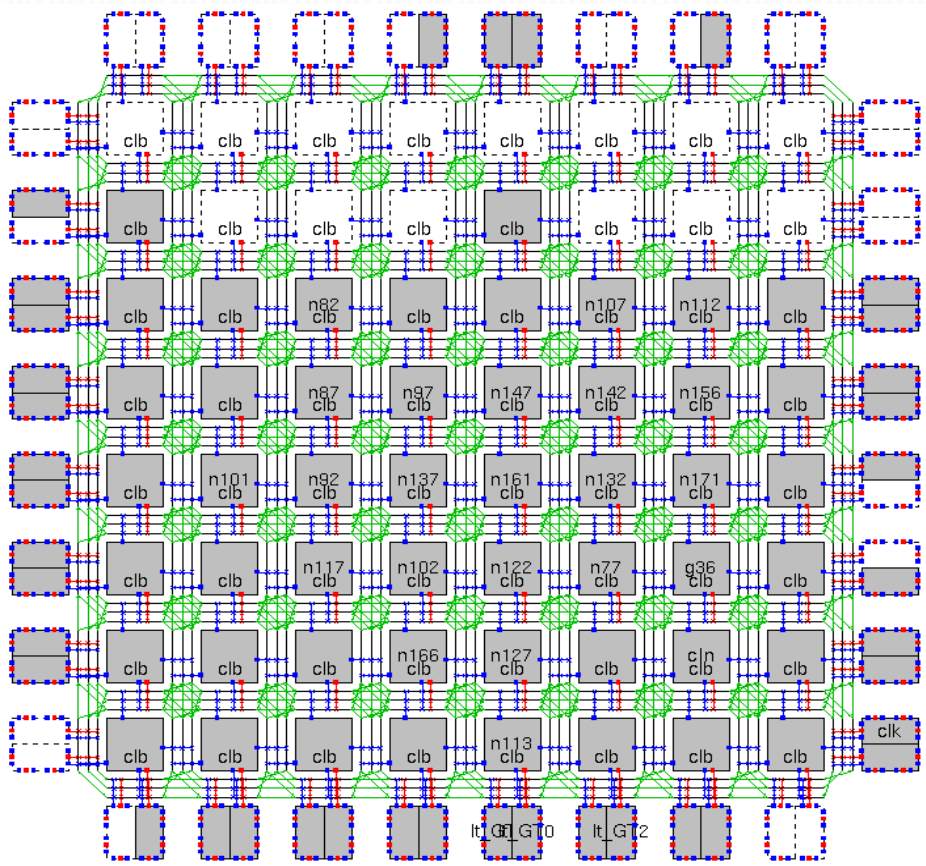
Field Programmable Gate Array (FPGA)



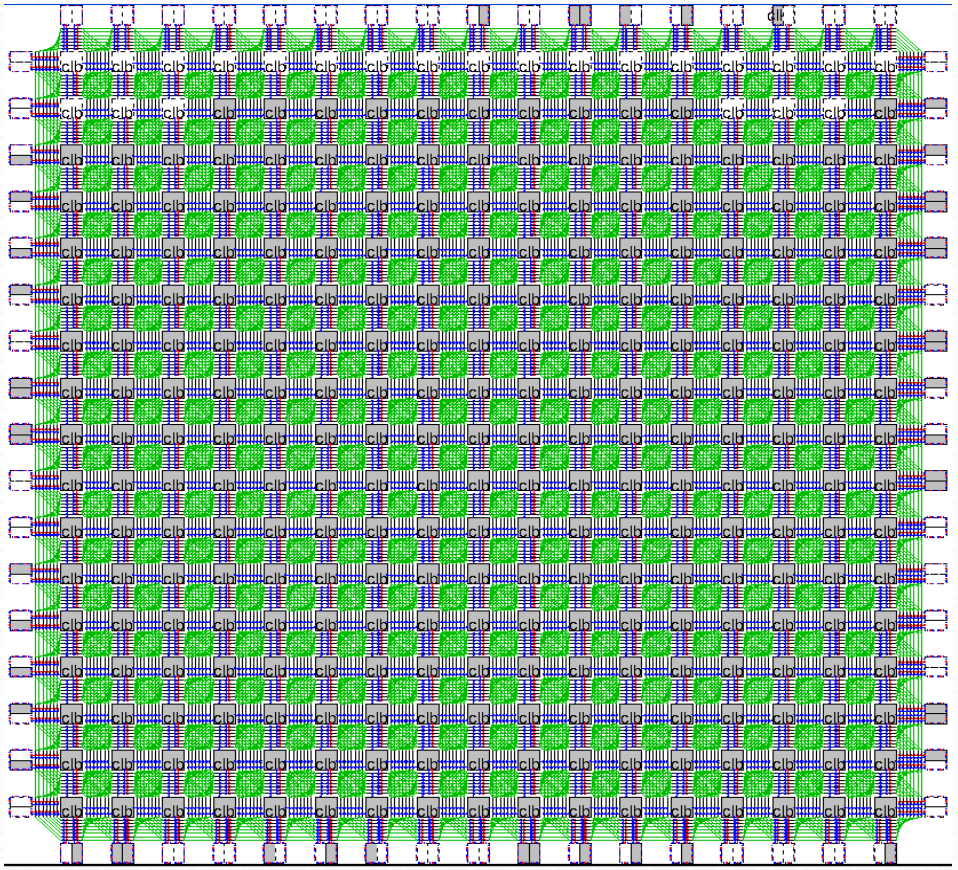
Field Programmable Gate Array (FPGA)



The Program is the Configuration

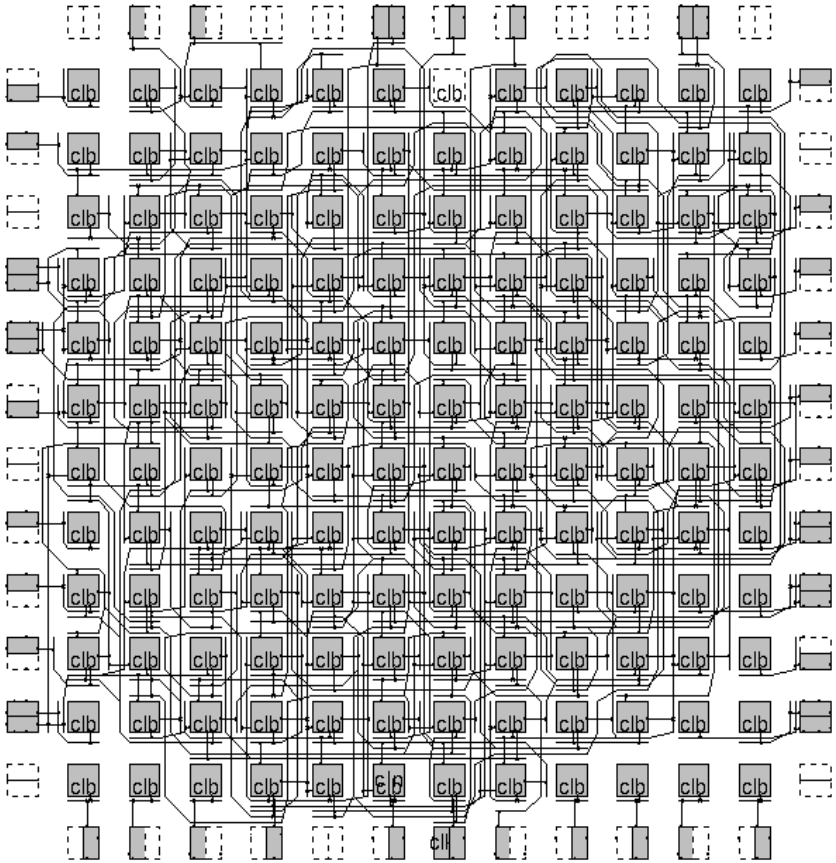


(a) abs

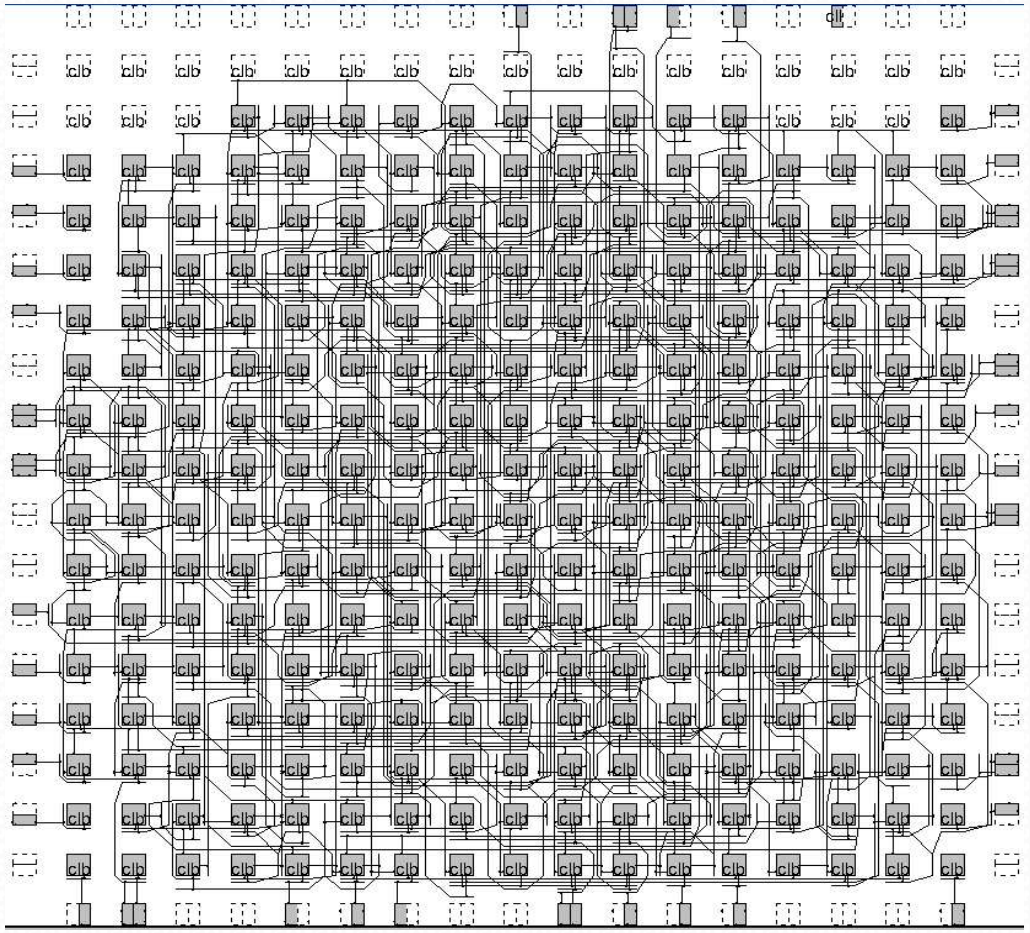


(b) calcNeighbor

The Program is the Configuration



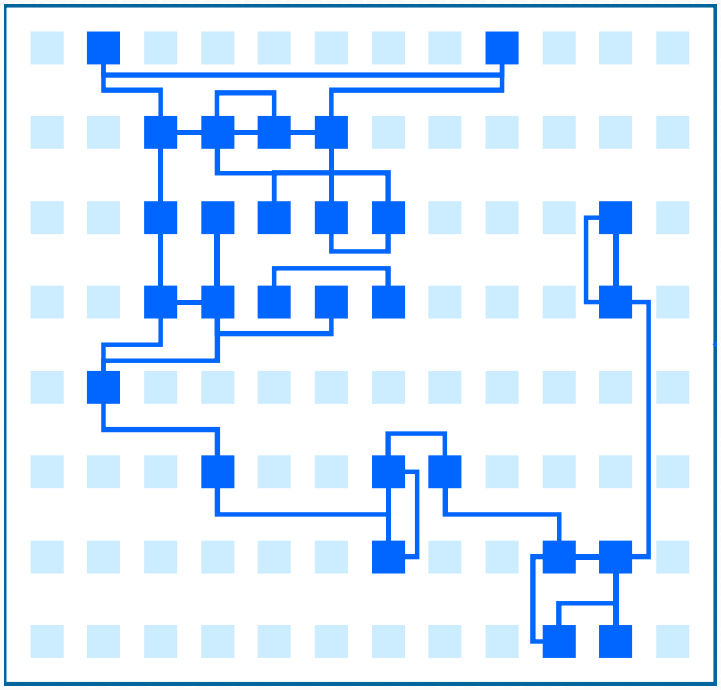
(a) Crc16



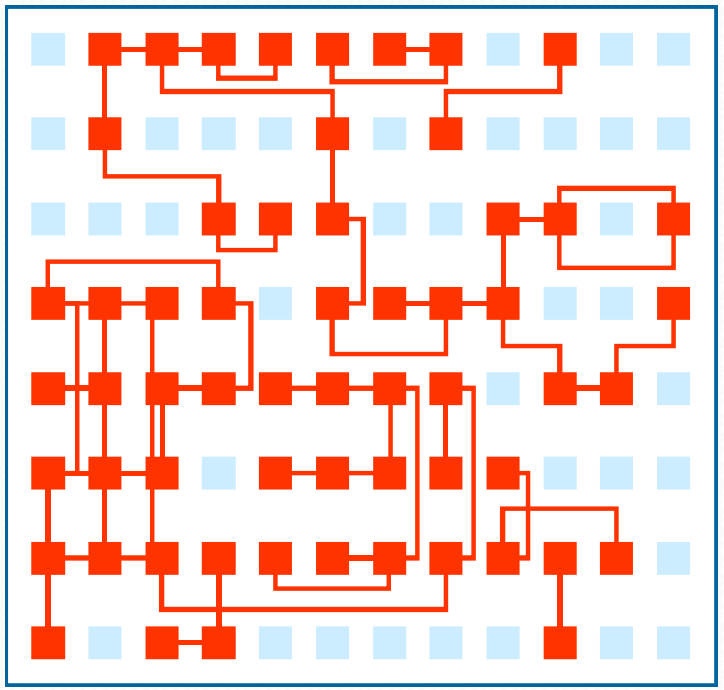
(b) calcNeighbor

Space-Time Computation

```
for(i=1; i<length; i++) {
     if (max < T[i]) {
        max = T[i];
     }
}
```

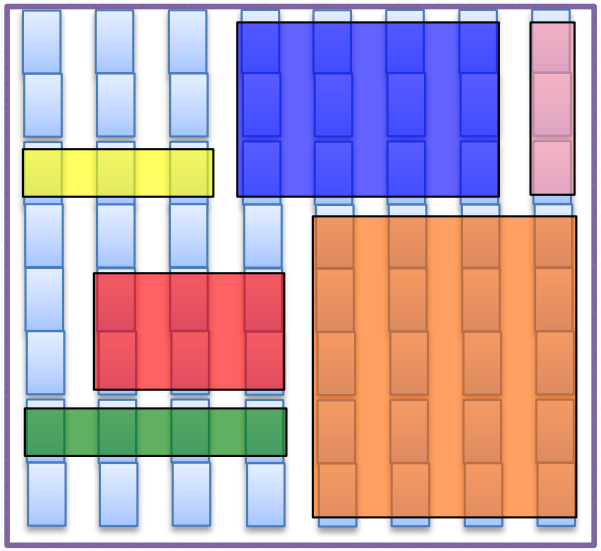


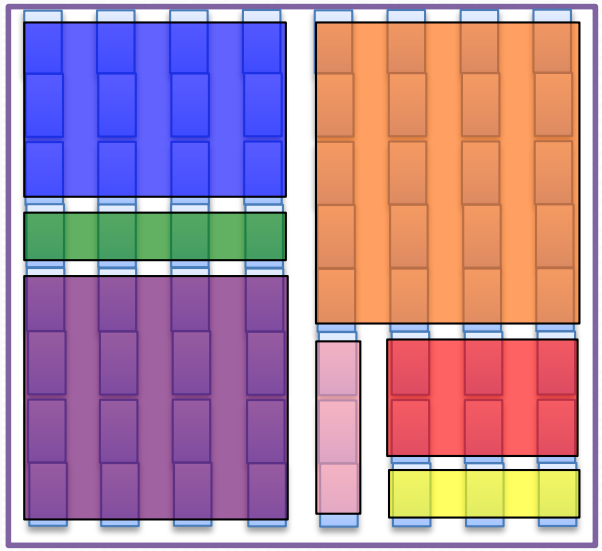
```
for(i=1; i<N; i++) {
    for(j=1; j<M; j++) {
        y[i][j]+=x[i][j]*h[j][i]
    }
}
```

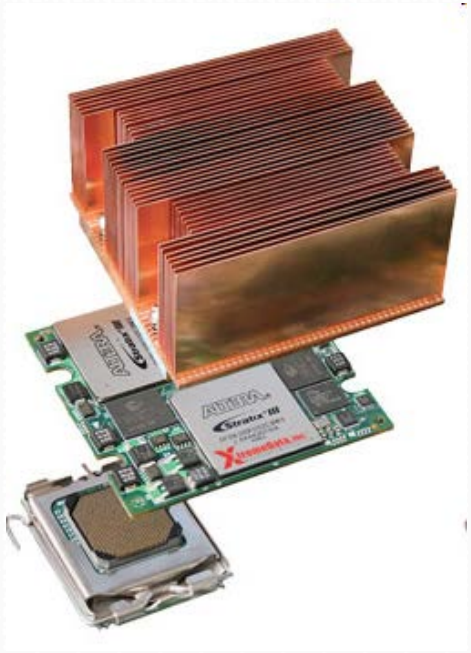


FPGA Acceleration

- FPGAs can run multiple tasks in parallel





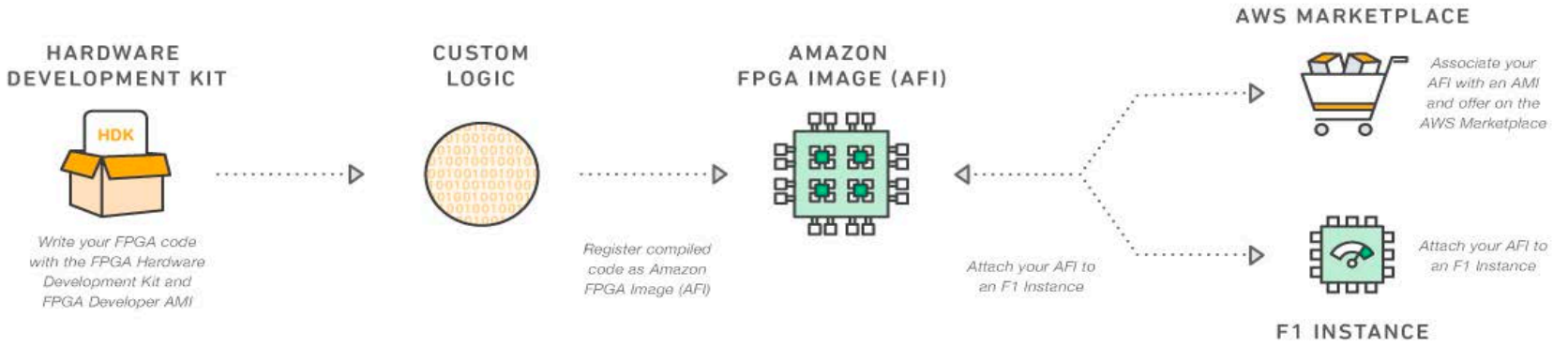


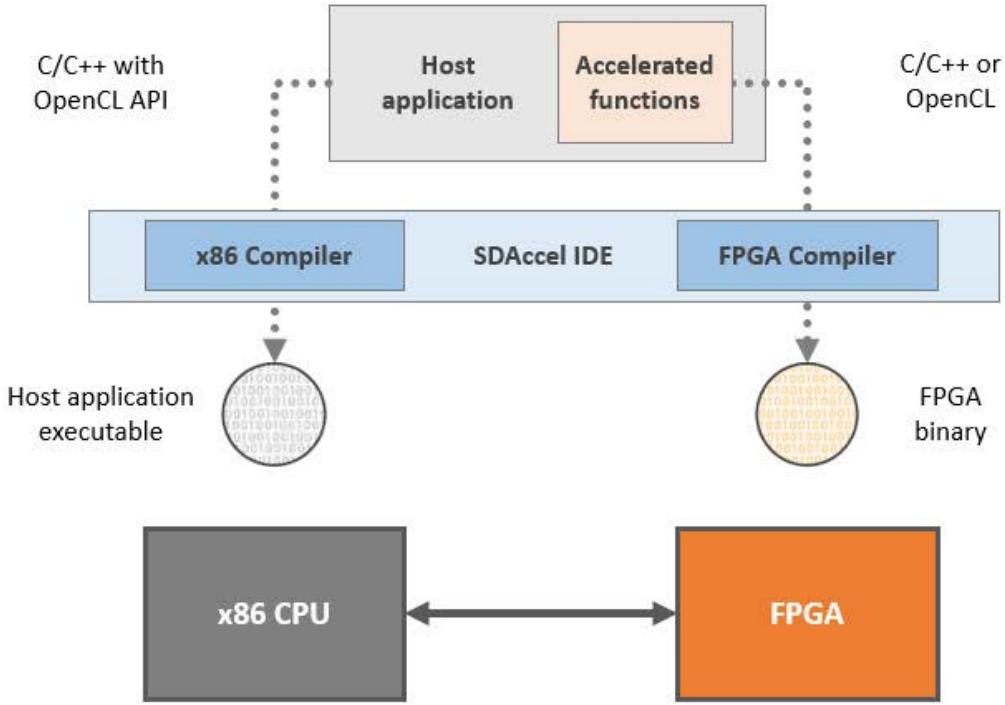
FPGA accelerators for HPC/Cloud

- Towards heterogeneous multicores

Amazon AWS EC2 F1

aws.amazon.com





AWS EC2 F1 Platform

Instance Size	FPGAs	DDR-4 (GiB)	vCPUs	Instance Memory (GiB)	NVMe Instance Storage (GB)	Network Bandwidth
f1.2xlarge	1	4 x 16	8	122	1 x 470	Up to 10 Gbps
f1.16xlarge	8	32 x 16	64	976	4 x 940	25 Gbps

- Up to 8 Xilinx UltraScale+ FPGA devices in a single EC2/F1 instance

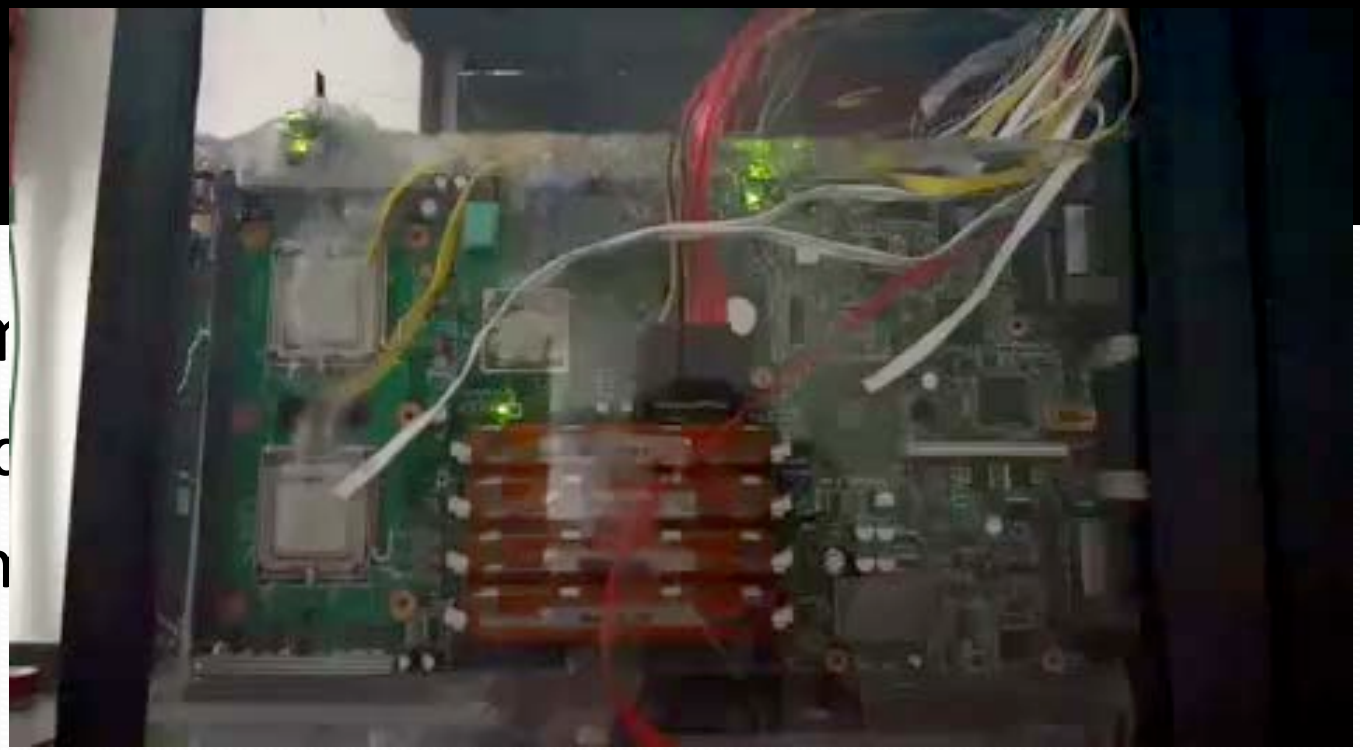
Time has Come for Specialization

- 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%

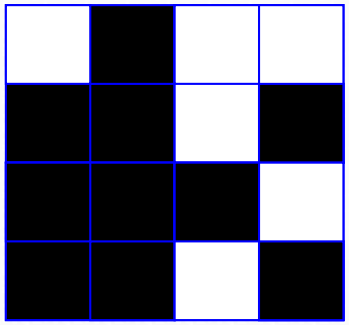
Conclusions

- 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

Conclusions



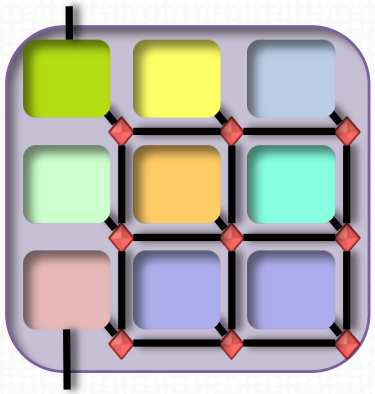
- Energy consum
  - True in embed
  - True in HPC, n
- End of Moore's law…
- Multicores but utilization wall
  - Percentage of a chip that can switch at full frequency drops exponentially

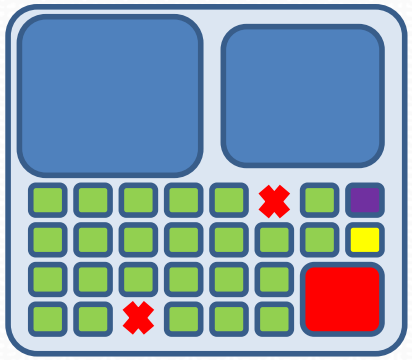


Dark Silicon

What's next?

- Dark Silicon is also an opportunity
  - Heterogeneous manycore architectures





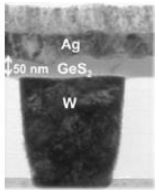
- Efficiency of hardware specialization
  - Domain-specific architectures and languages
- Computing just right
  - @design-time or @run- time

What's next?

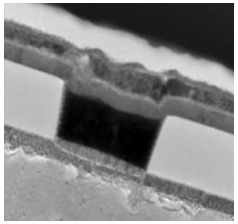
- Emerging devices

- Cells, brain, neurons have "analog" behavior
- And compute with very low precision

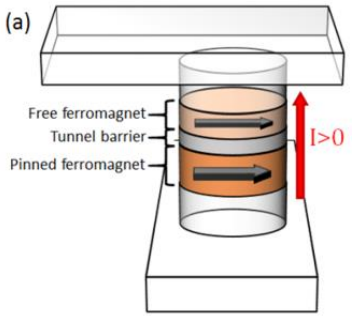
- Making neuromorphic computing more efficient



Phase Change Memory



Memristors, Oxide Resistive Memory



Spin Torque Magnetic Memory