EITF35: Introduction to Structured VLSI Design

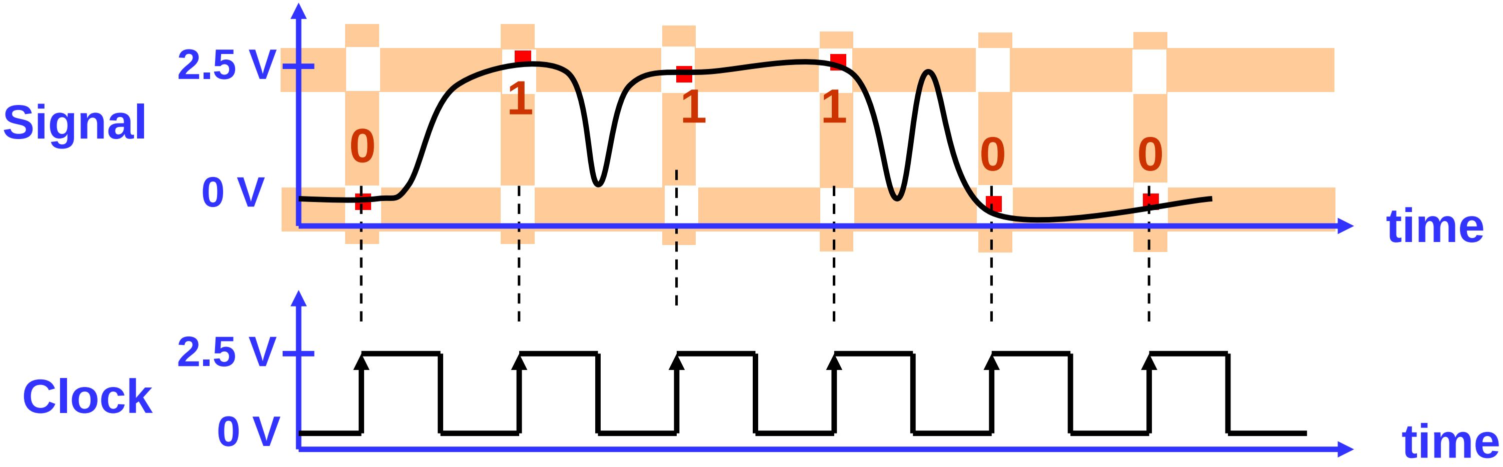
Part 1.1.2: Introduction (Digital VLSI Systems)

Liang Liu
liang.liu@eit.lth.se

Outline

- Digital VLSI
- Roadmap
- Device Technology & Platforms
- System Representation
- Design Flow
- RTL (register transfer level) Basics

Digitalization

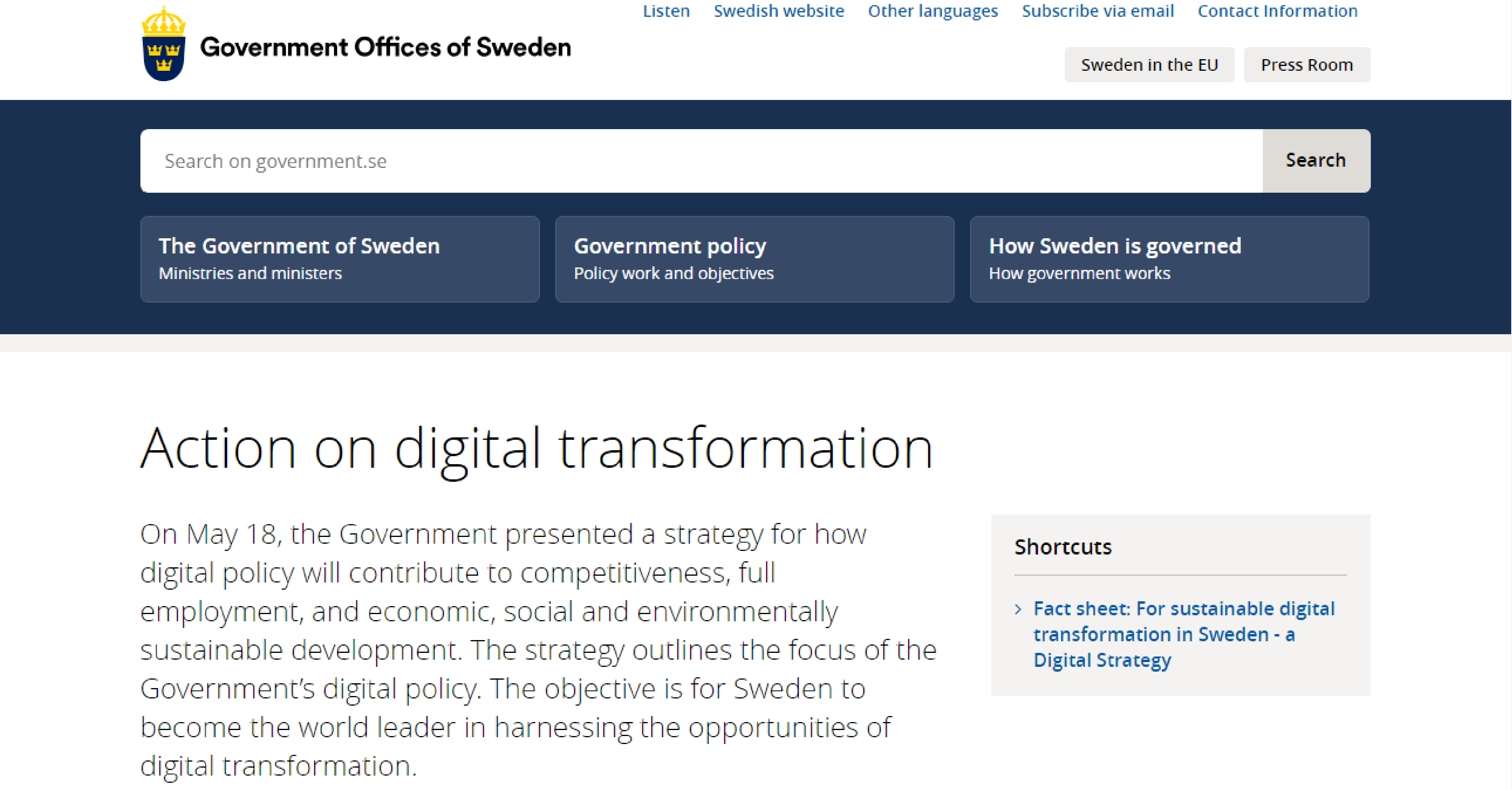


□ Digital is an abstraction

- Discrete in time: Sampling
- Discrete in value: Quantization

□ Digital vs. Analog

- Flexibility & functionality: easier to store and manipulate information
- Reliability: tolerant to noise, mismatch, variations, etc.
- Economic: “easy” to design, and friendly to technology evolvement



Action on digital transformation

On May 18, the Government presented a strategy for how digital policy will contribute to competitiveness, full employment, and economic, social and environmentally sustainable development. The strategy outlines the focus of the Government's digital policy. The objective is for Sweden to become the world leader in harnessing the opportunities of digital transformation.

Shortcuts

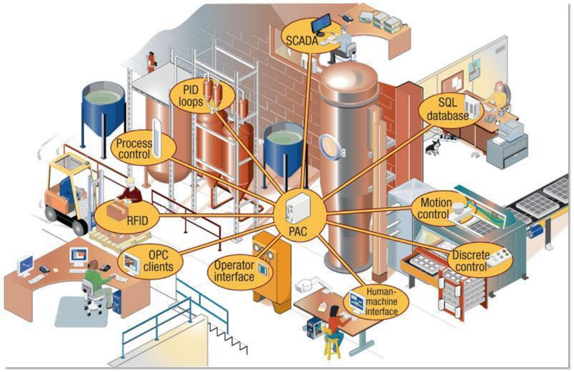
- **Fact sheet: For sustainable digital transformation in Sweden - a Digital Strategy**

Applications 4C:CCCC

Computation

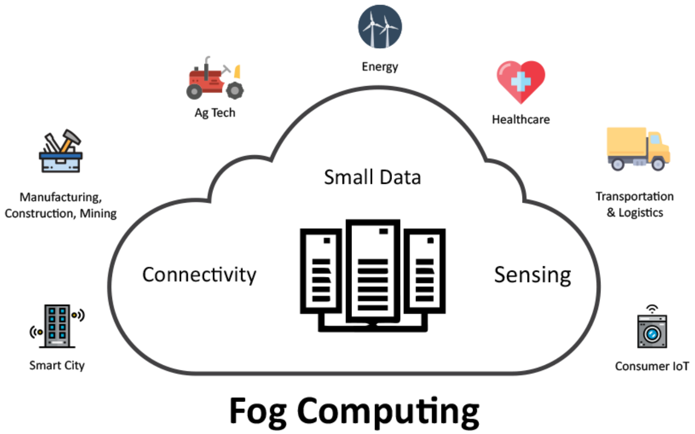
Communication

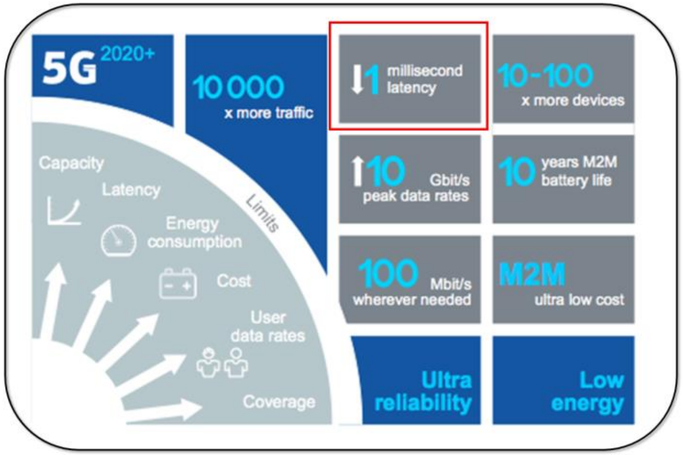
Consumer



Control

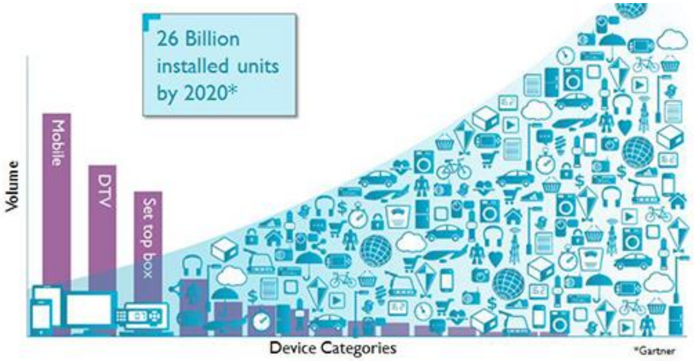
Applications 4C (evolution):CCCC

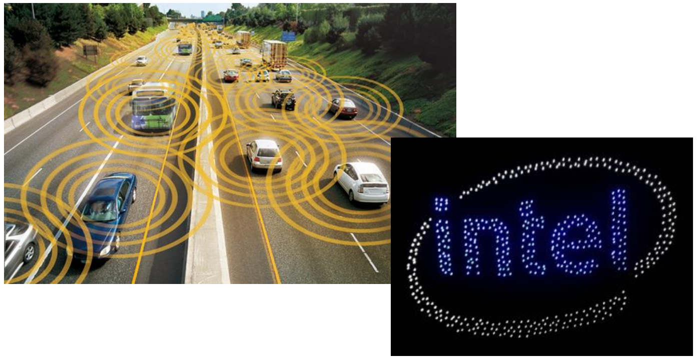




Computation (Cloud, Mobile, Fog, Edge)

Communication (5G & Beyond)





Consumer (IoT, IoE)

Control (AI, ML, DL)

Outline

- □ Why Digital?
  - Advantages
  - Some applications
- □ **Roadmap (Enabling Tech.)**
- □ Device Technology & Platforms
- □ System Representation
- □ Design Flow
- □ RTL Basics

Enabling Technology

1981

1946

Why machine learning hot now?

Big Data Avaliability

facebook

350M images/day

Walmart

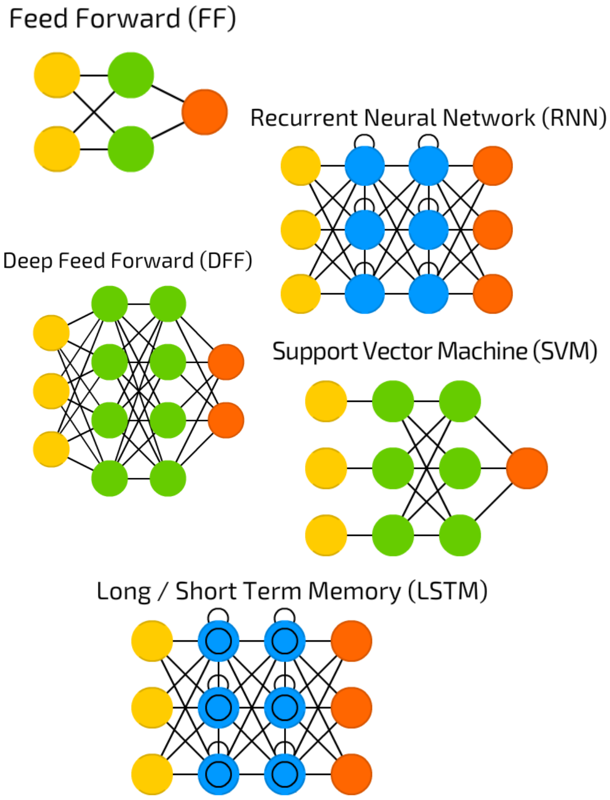
Save money. Live better.

2.5 Petabytes of customer data hourly

IoT

20.4 billion connected things by 2020

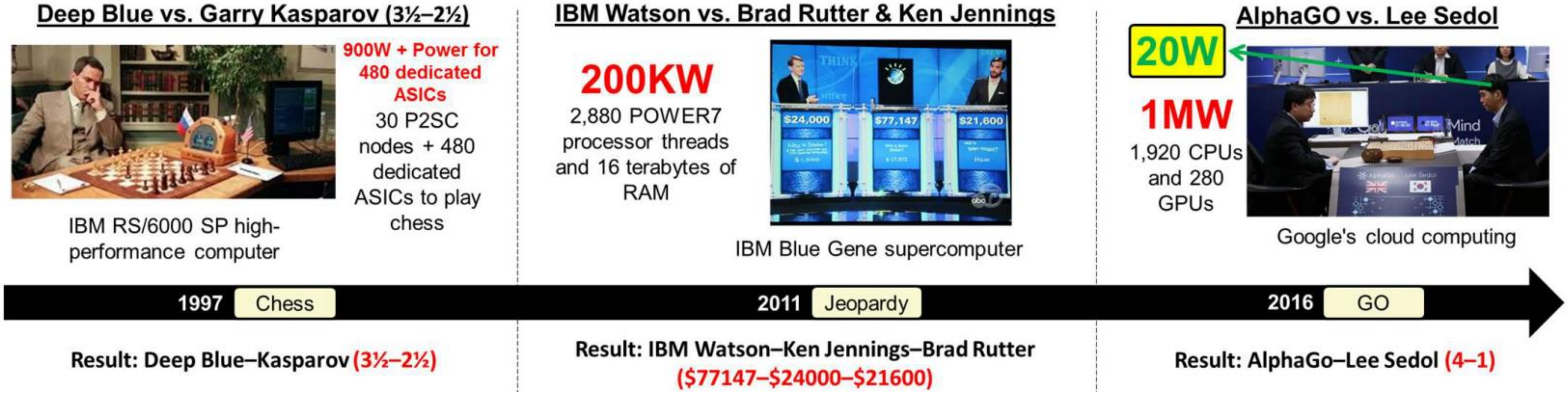
ML Algorithms and Methods


Feed Forward (FF)
Recurrent Neural Network (RNN)
Deep Feed Forward (DFF)
Support Vector Machine (SVM)
Long / Short Term Memory (LSTM)

Source: http://www.asimovinstitute.org/neural-network-zoo/

Hardware Acceleration

Implementation charllenges: power consumption



Deep Blue vs. Garry Kasparov (3½–2½)

900W + Power for 480 dedicated ASICs

30 P2SC nodes + 480 dedicated ASICs to play chess

IBM RS/6000 SP high-performance computer

1997 Chess

Result: Deep Blue–Kasparov (3½–2½)

IBM Watson vs. Brad Rutter & Ken Jennings

200KW

2,880 POWER7 processor threads and 16 terabytes of RAM

IBM Blue Gene supercomputer

2011 Jeopardy

Result: IBM Watson–Ken Jennings–Brad Rutter ($77147–$24000–$21600)

AlphaGO vs. Lee Sedol

20W

1MW

1,920 CPUs and 280 GPUs

Google's cloud computing

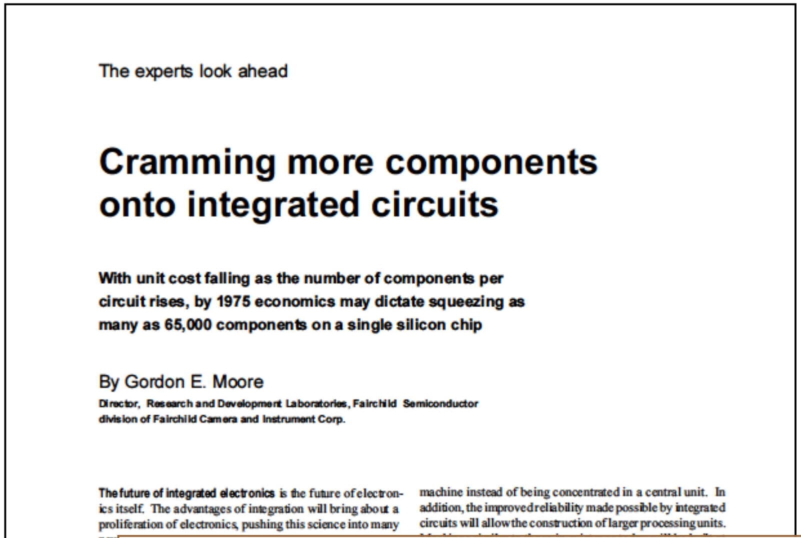
2016 GO

Result: AlphaGo–Lee Sedol (4–1)

Mobile: Battery life

Source: M. Shafique, "An Overview of Next-Generation Architectures for Machine Learning"

Moore’s Law



The experts look ahead

Cramming more components onto integrated circuits

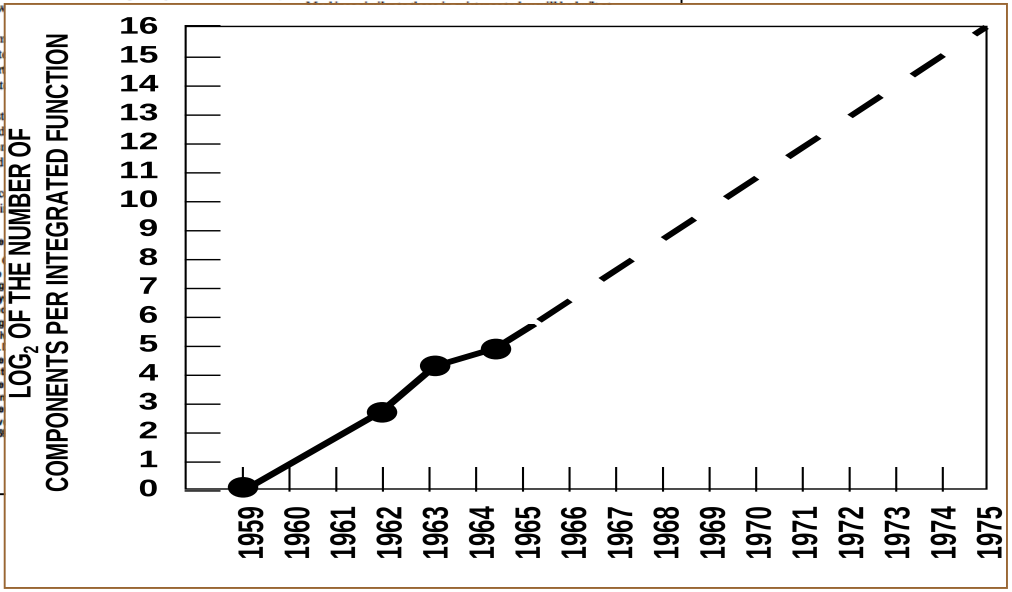
With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

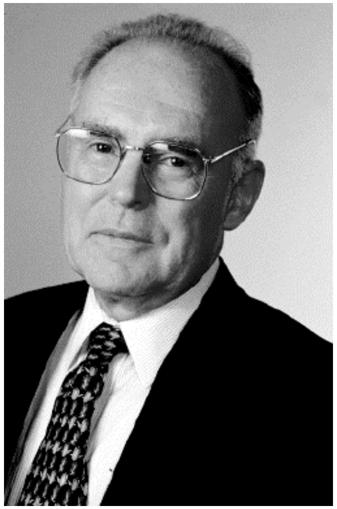
The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many

machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units.

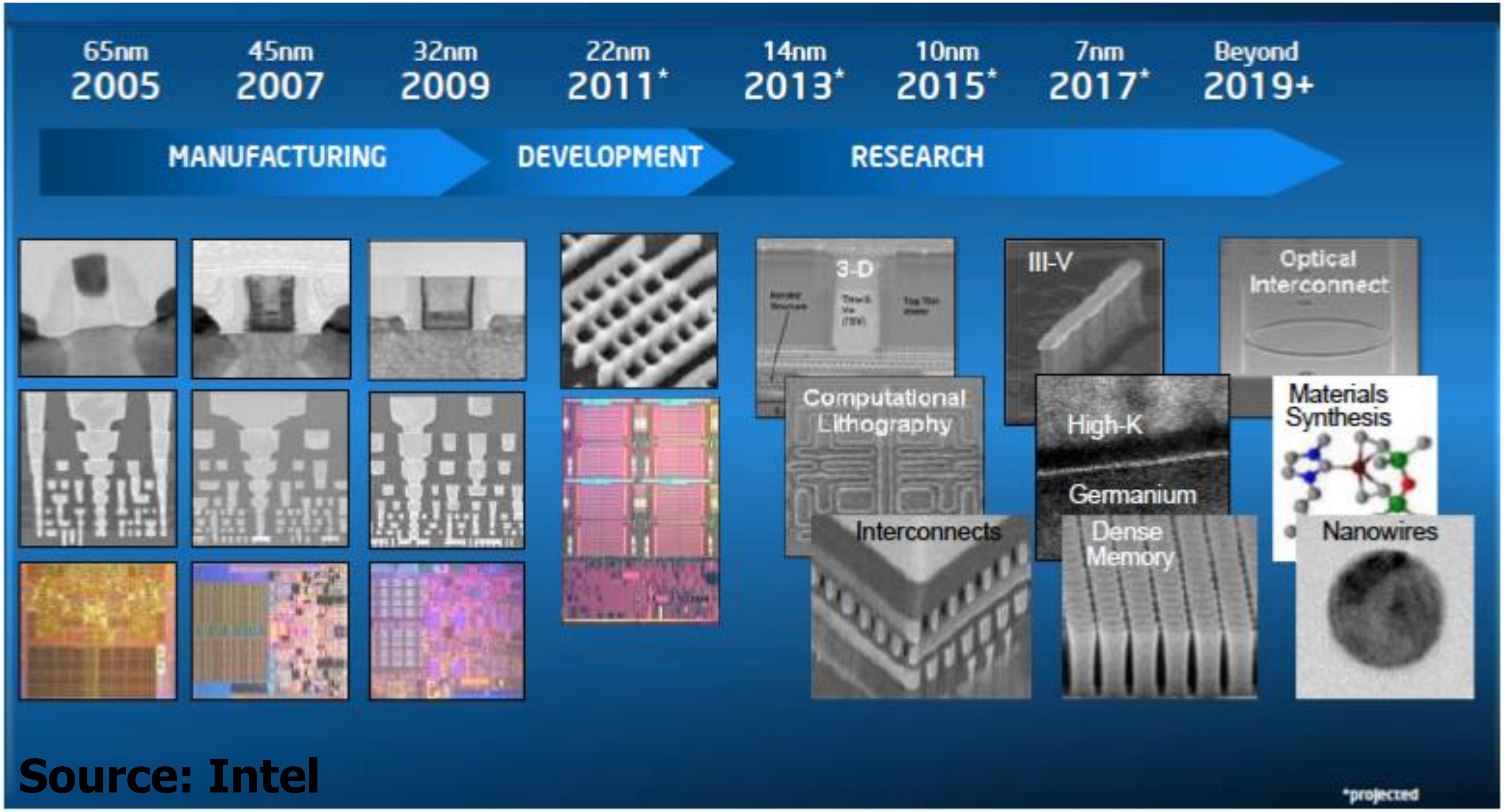


- **Electronics, Apr. 19, 1965**

Gordon Moore (co-founder of Intel) made a prediction that semiconductor technology will double its effectiveness every 18 months



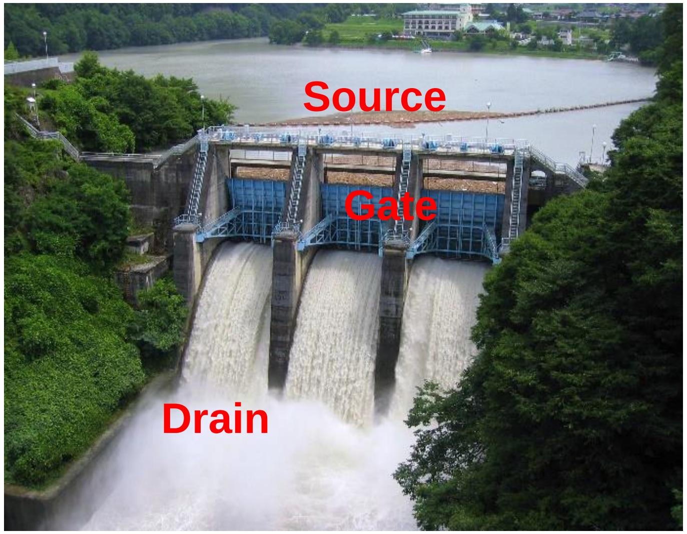
On-Time 2 Year Cycles

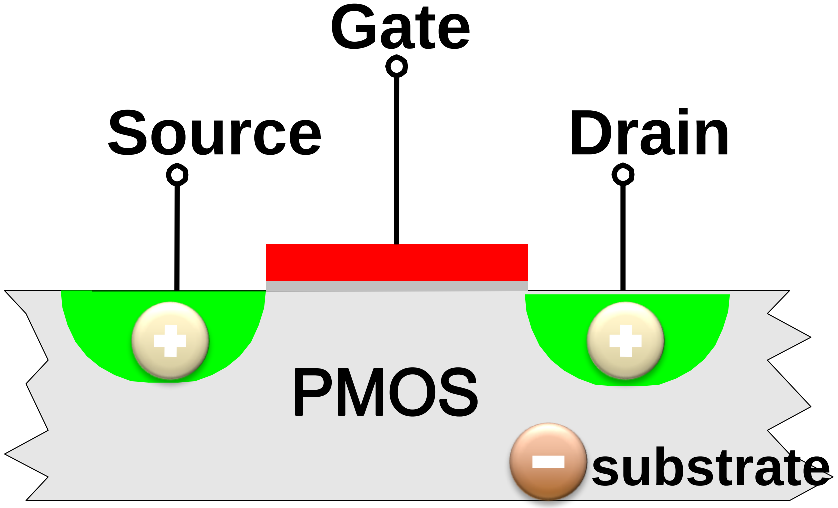


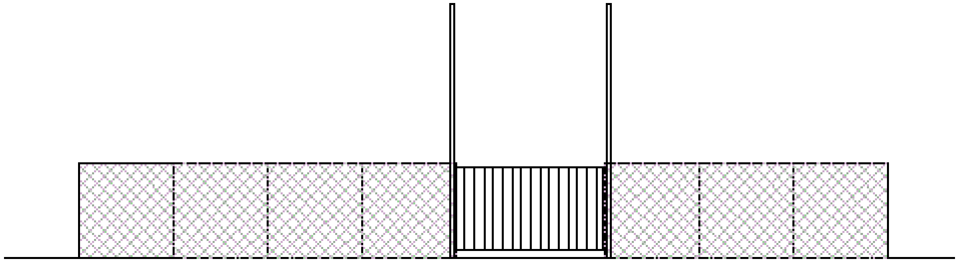
Source: Intel

*projected

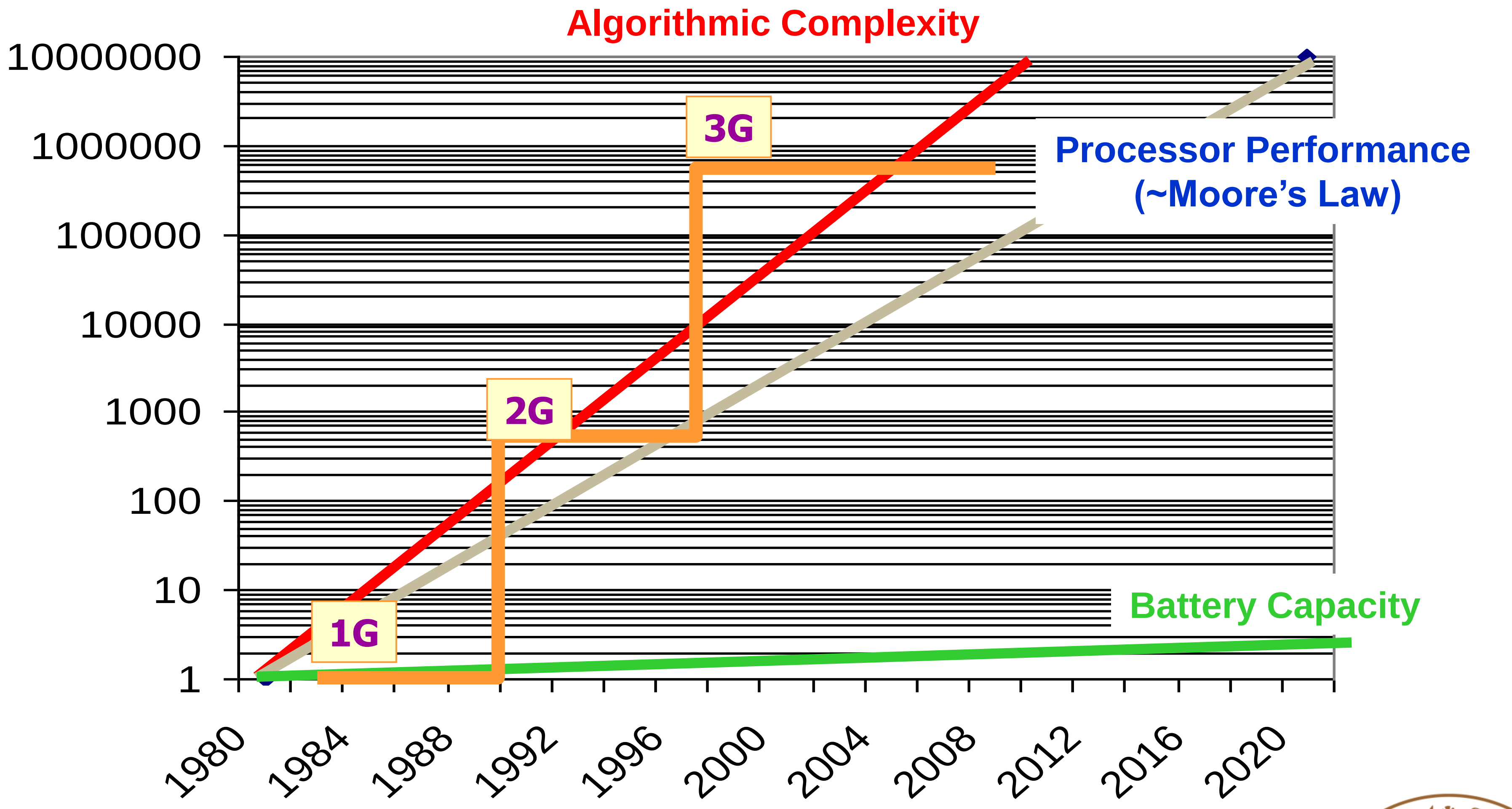
FinFET





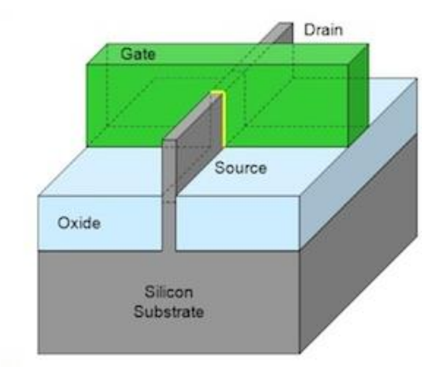


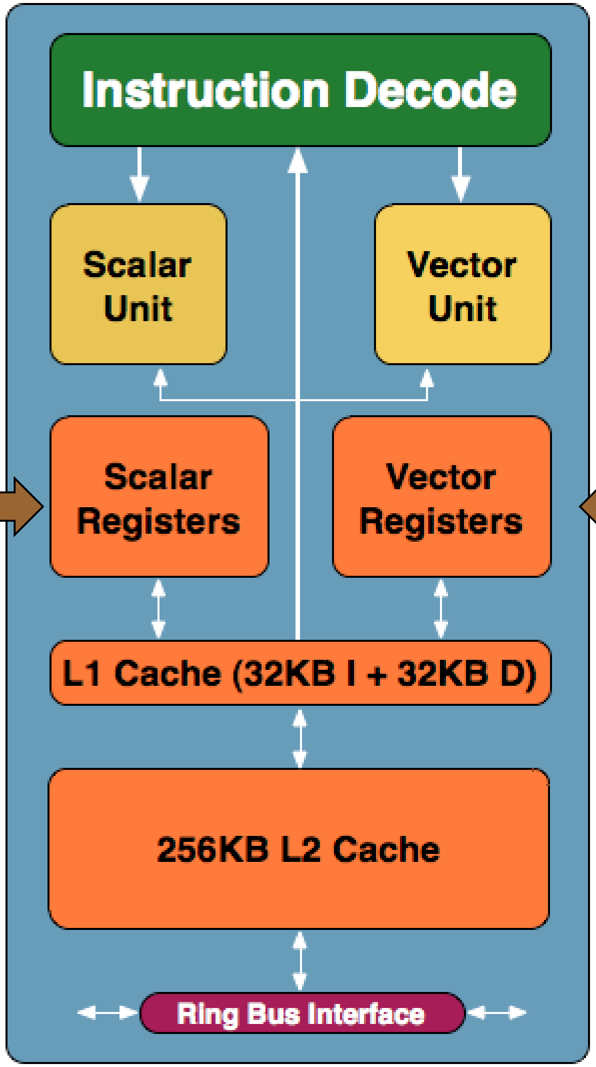
Algorithms beats Moore beats Chemists



Courtesy: Ravi Subramanian (Morphics)

VLSI Architecture: how to decide?




Instruction Decode

Scalar Unit

Vector Unit

Scalar Registers

Vector Registers

L1 Cache (32KB I + 32KB D)

256KB L2 Cache

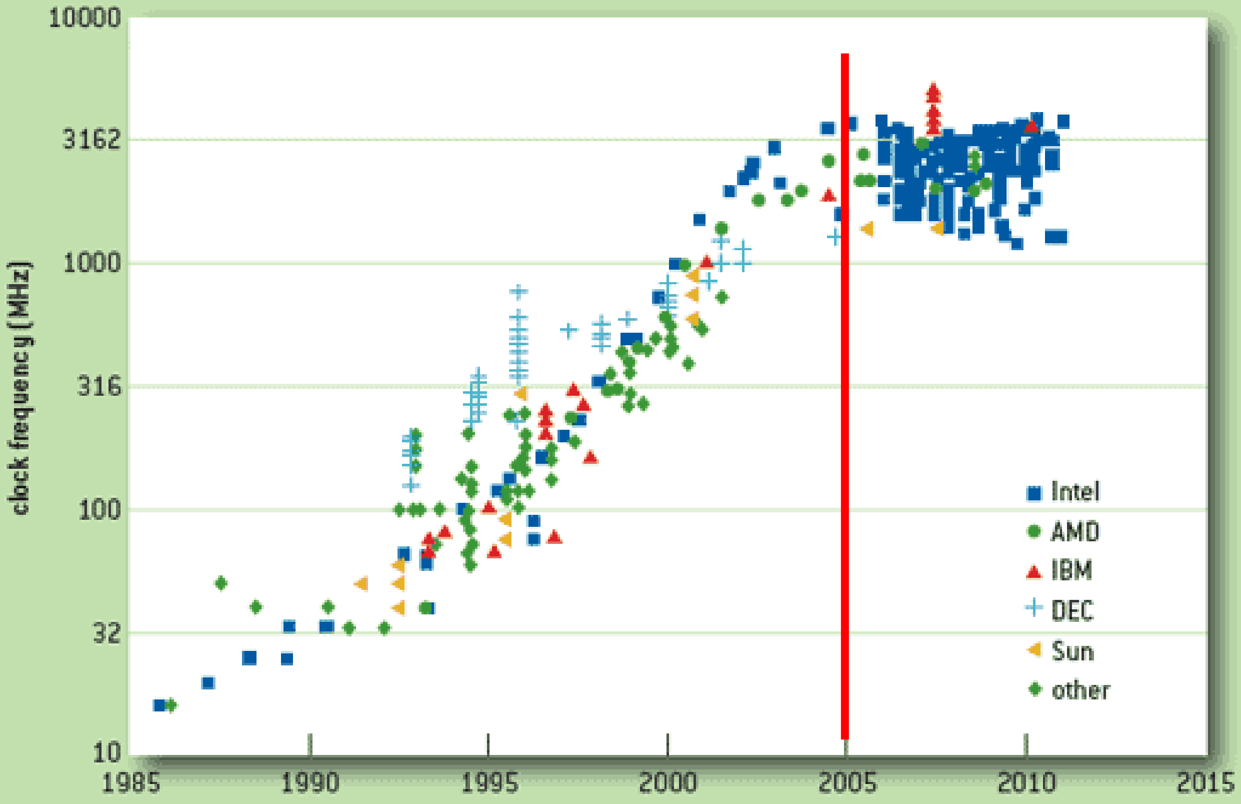
Ring Bus Interface

Moore’s Law: frequency

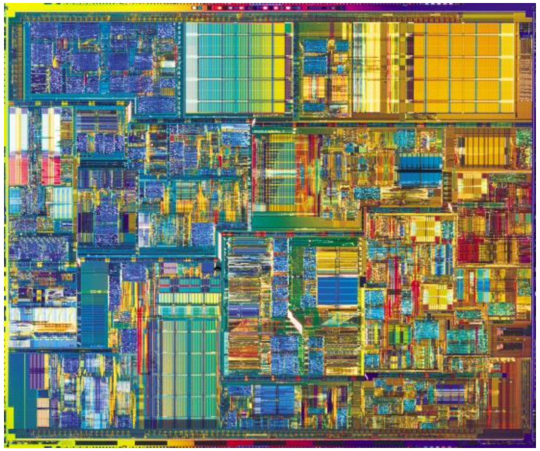
FIGURE 7

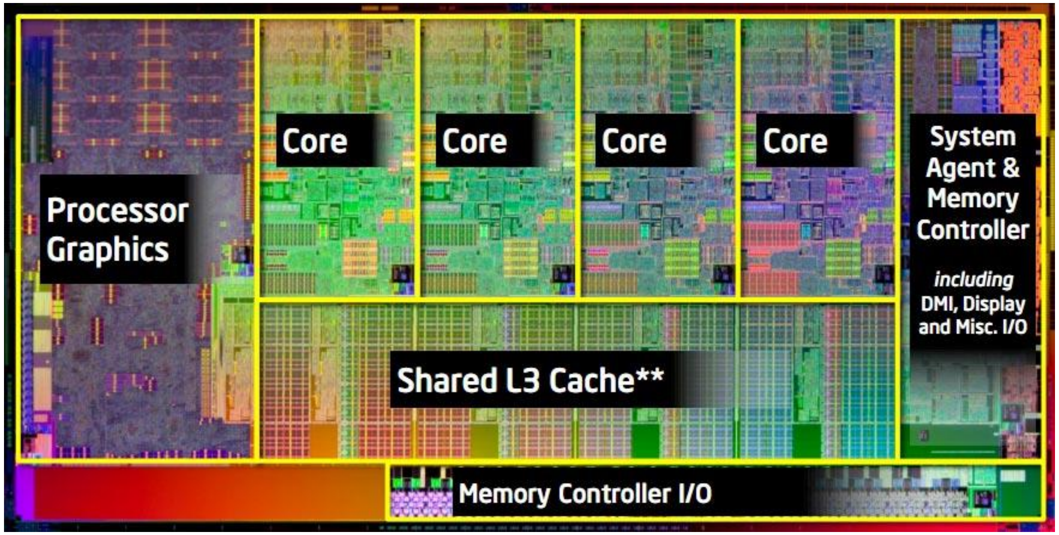
Source: CPU DB: Recording Microprocessor History

Processor Frequency Scaling Over Time

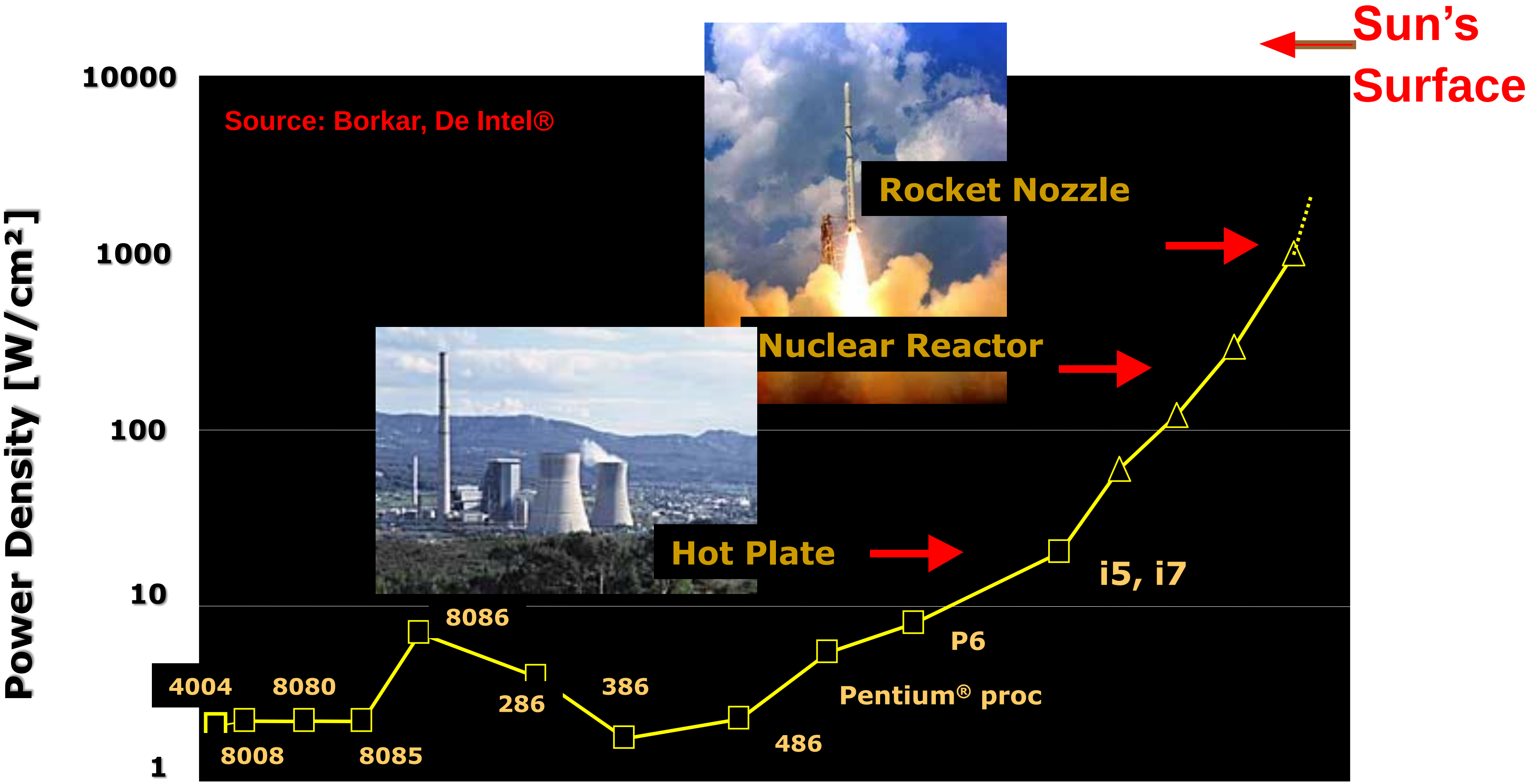


Architecture change due to technology limitations





Moore’s Law: power density



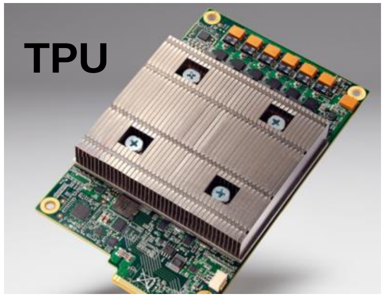
Architecture change due to new applications (Power)

intel

ARM

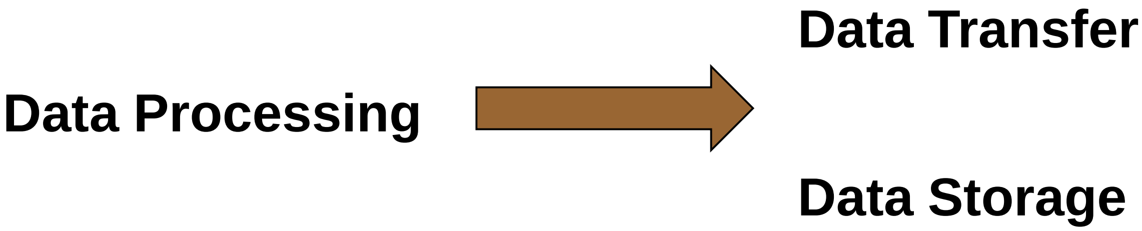
?

Architecture change due to new applications (Performance)



Architecture change due to new applications (big data)

Architecture change due to new applications (big data)



Data Processing

Data Transfer

Data Storage

Outline

□ **Why Digital?**

- Advantages
- Some applications

□ **History & Roadmap**

□ **Device Technology & Platforms**

□ **System Representation**

□ **Design Flow**

□ **RTL Basics**

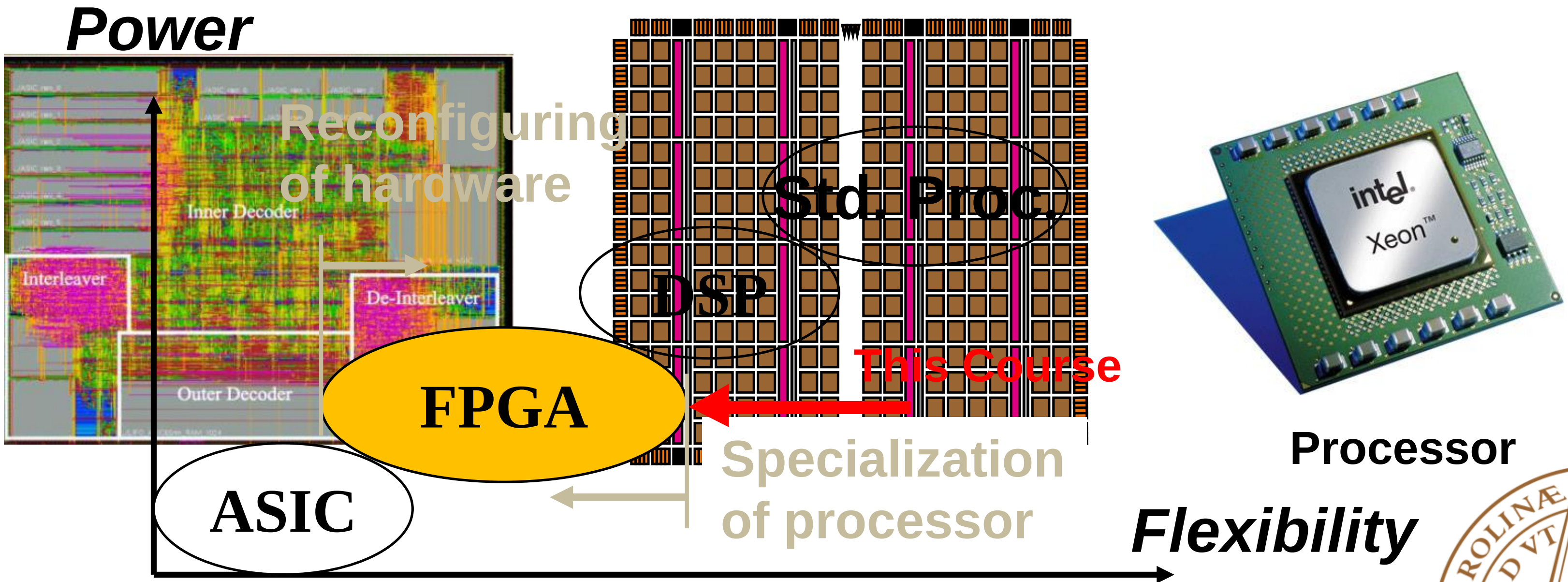
Devices

□ **Programmable integrated circuits**

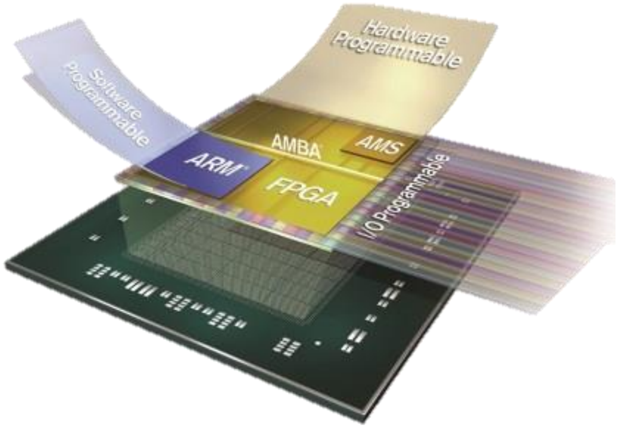
- **Microprocessors**, DSP, ***FPGA*** and memories

□ **Application-specific integrated circuits (ASIC)**

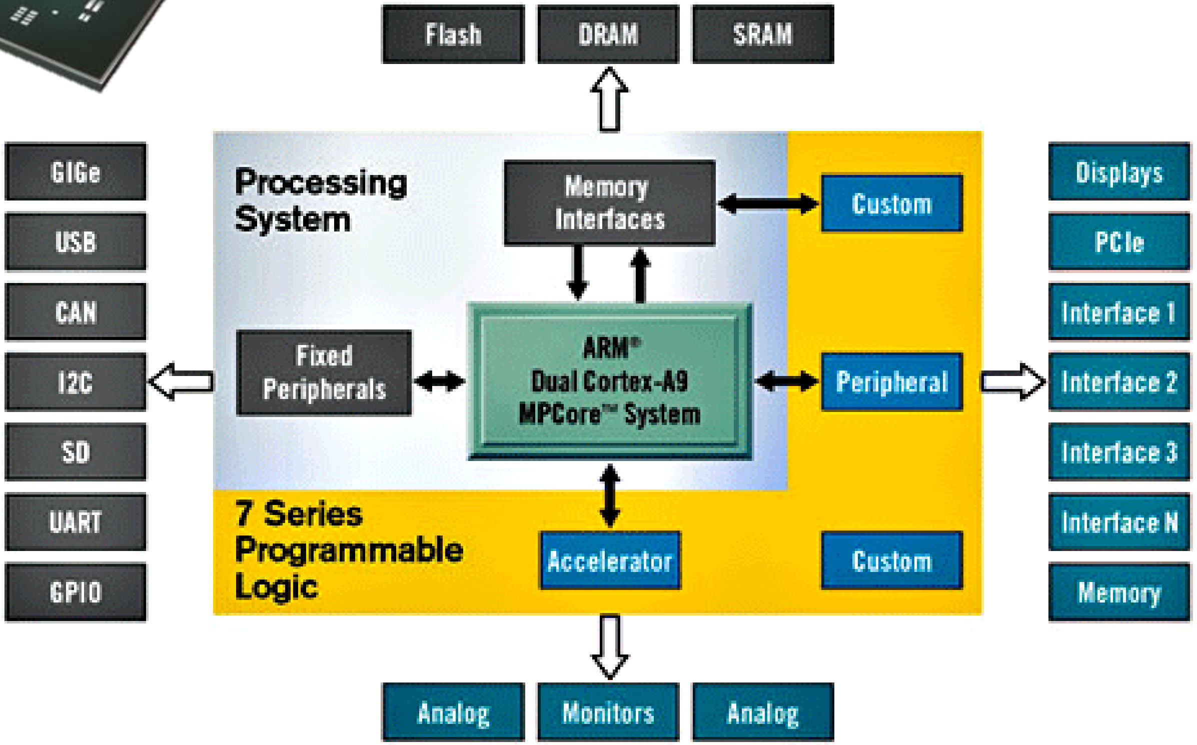
- Full-custom ASIC
- ***Standard-cell ASIC (IP)***



Devices (heterogeneous)

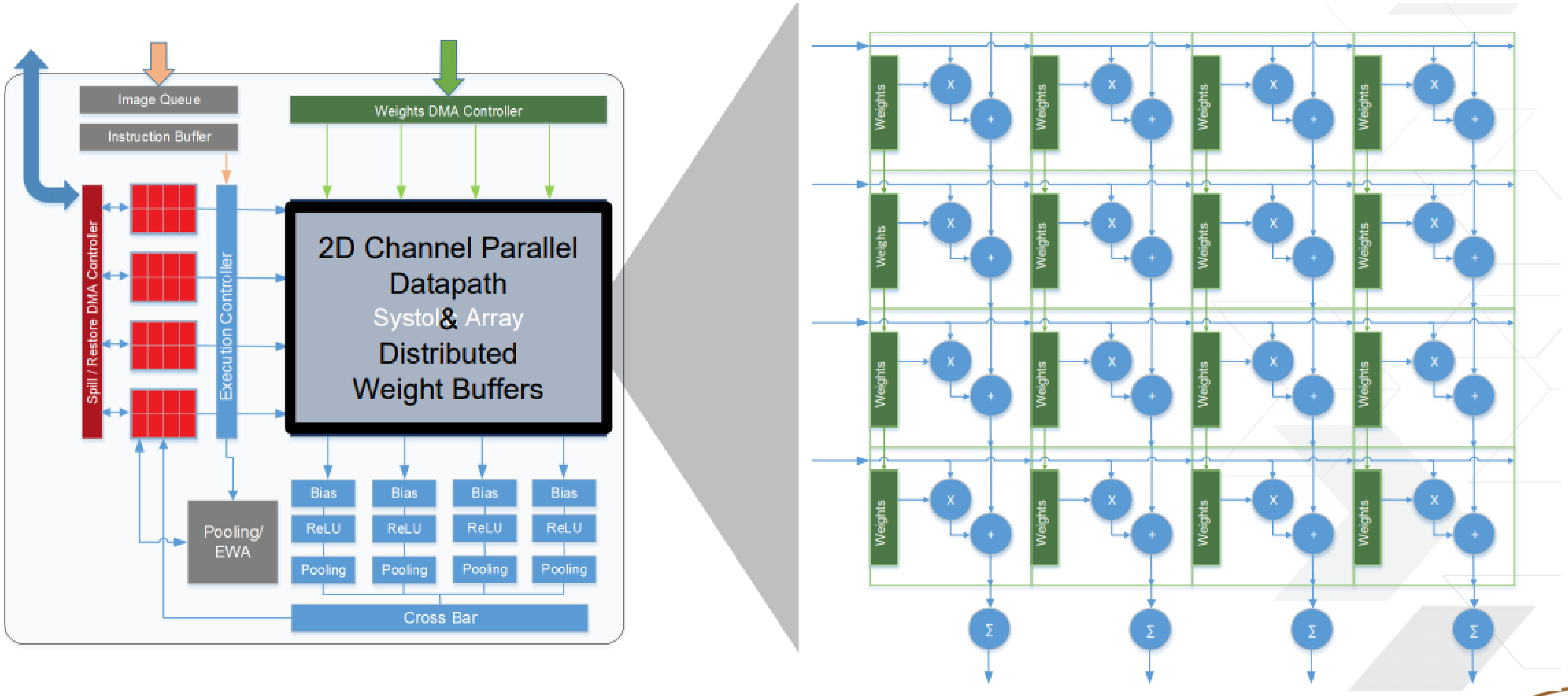


All Programmable SoC

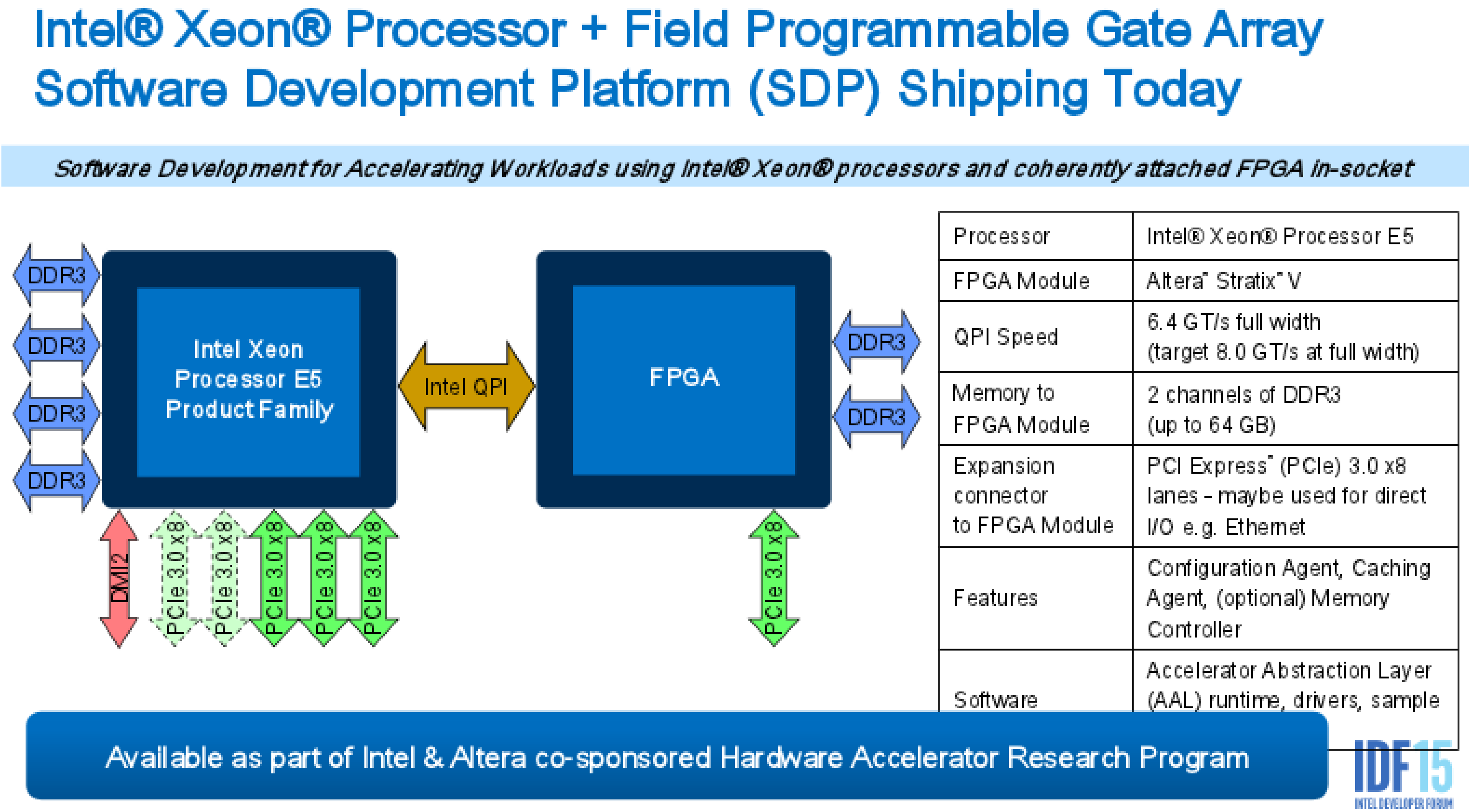


Devices (heterogeneous)

Xilinx Announces Project Everest: The 7nm FPGA SoC Hybrid



Devices



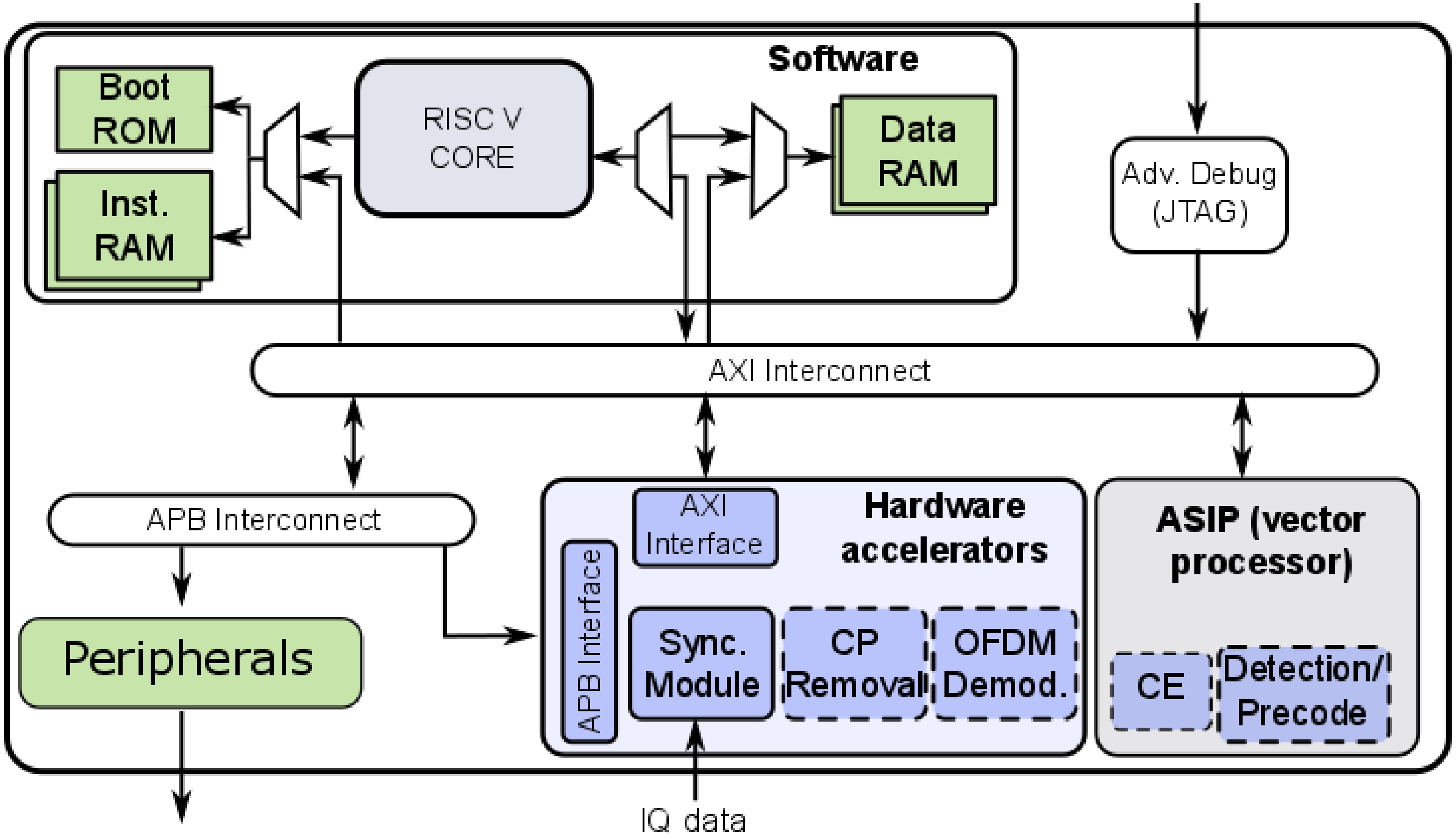
Intel® Xeon® Processor + Field Programmable Gate Array Software Development Platform (SDP) Shipping Today

Software Development for Accelerating Workloads using Intel® Xeon® processors and coherently attached FPGA in-socket

Processor	Intel® Xeon® Processor E5
FPGA Module	Altera* Stratix* V
QPI Speed	6.4 GT/s full width (target 8.0 GT/s at full width)
Memory to FPGA Module	2 channels of DDR3 (up to 64 GB)
Expansion connector to FPGA Module	PCI Express* (PCIe) 3.0 x8 lanes - maybe used for direct I/O e.g. Ethernet
Features	Configuration Agent, Caching Agent, (optional) Memory Controller
Software	Accelerator Abstraction Layer (AAL) runtime, drivers, sample

Available as part of Intel & Altera co-sponsored Hardware Accelerator Research Program

xG MPSoC @ EIT

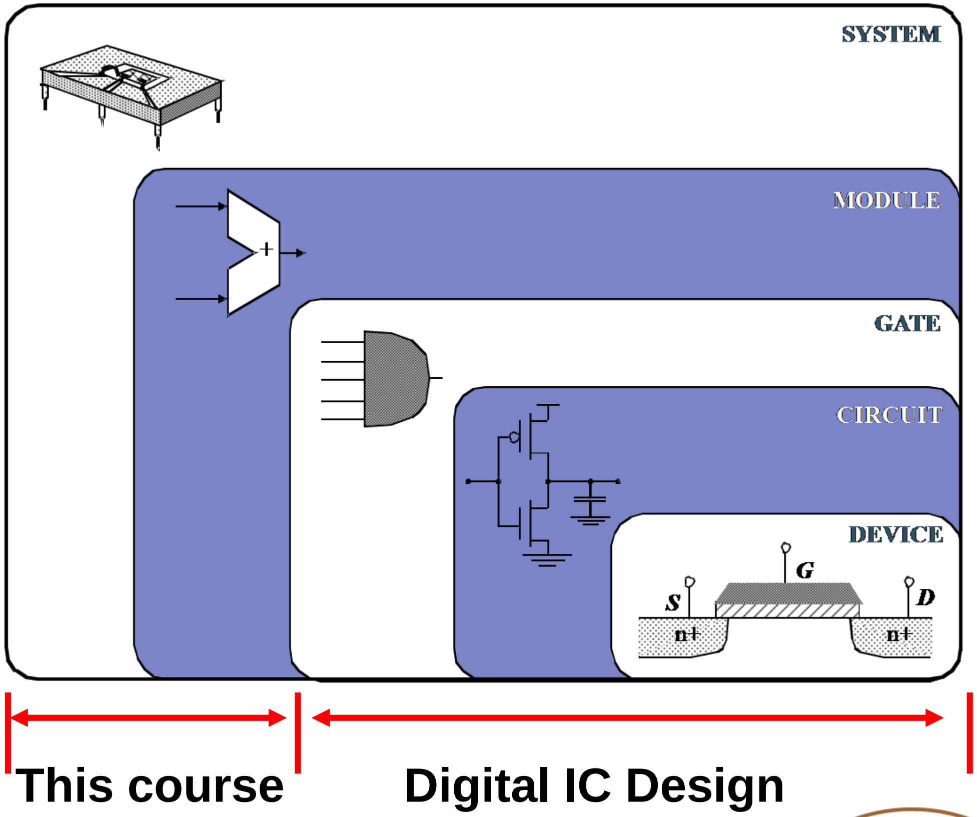

Software
Boot ROM
Inst. RAM
RISC V CORE
Data RAM
Adv. Debug (JTAG)
AXI Interconnect
APB Interconnect
Peripherals
AXI Interface
Hardware accelerators
APB Interface
Sync. Module
CP Removal
OFDM Demod.
ASIP (vector processor)
CE
Detection/ Precode
IQ data

Outline

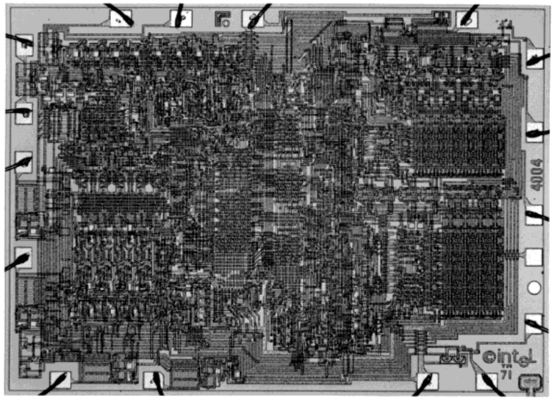
- □ Why Digital?
  - Advantages
  - Some applications
- □ History & Roadmap
- □ Device Technology & Platforms
- □ **System Representation**
- □ Design Flow
- □ RTL Basics

System Representation

- □System
  - SoC: a CPU chip …
- □Module
  - Macro cell in a chip：ALU…
- □Gate
  - Basic logic block：xor, nor…
- □Circuit
  - Transistors
- □Device
  - Gate, source, drain

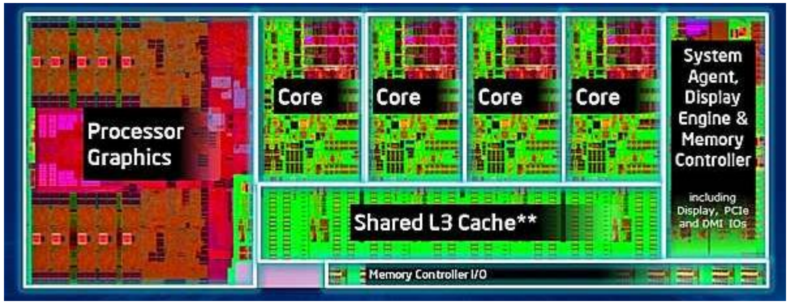


View a Design in a Proper Way



Intel 4004 (2.3K transistors)

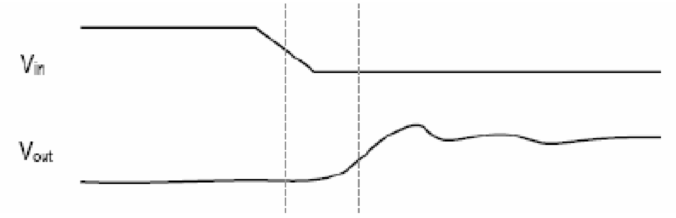
Full-custom

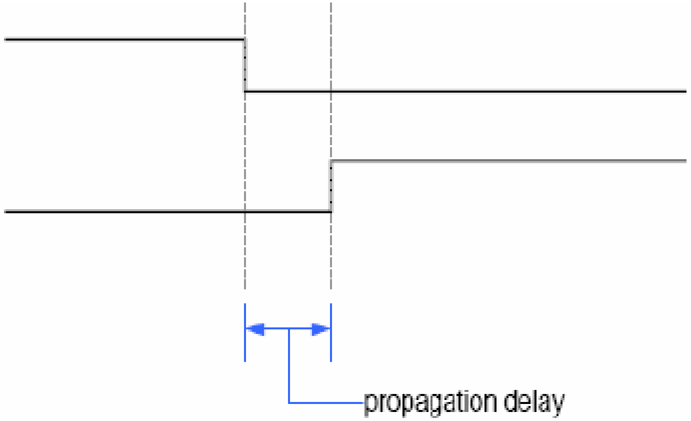

Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O

Intel Haswell (1.4B transistors)

?

- □ **Abstraction: simplified model of a system**
  - **Show the selected features accurate enough**
  - **Ignore the others**

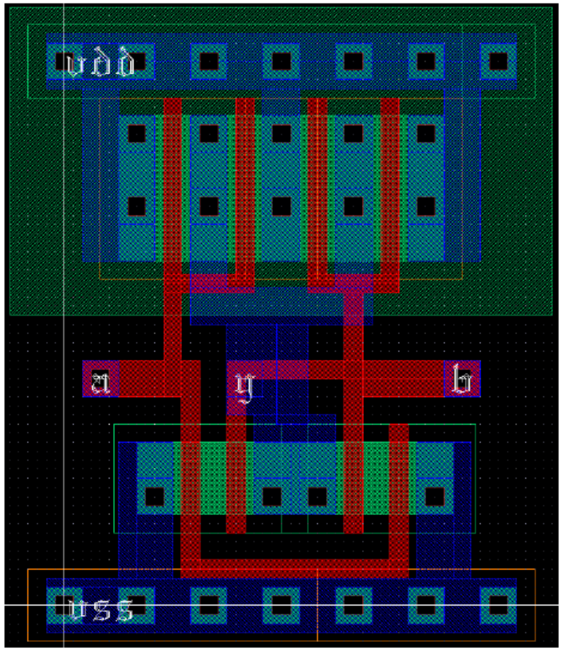




VLSI Design Flow

□ Evolution of circuit design (Design Hierarchy)

- Full-custom⇒Design-automation
  - □ ***Based on library cells and IPs***
  - □ ***Top-down methodology***
- Design abstraction⇒“Black box” or “Model”
  - □ ***Parameter simplification***
  - □ ***Accurate enough to meet the requirement***



```
    module HS65_GH_NAND2AX14 (Z, A, B);
       output Z;
       input A,B;
       not    U1 (INTERNAL1, B) ;
       or   #1 U2 (Z, A, INTERNAL1) ;
       specify
          (A +=> Z) = (0.1,0.1);
          (B -=> Z) = (0.1,0.1);
       endspecify
    endmodule // HS65_GH_NAND2AX14
```

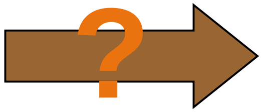
Outline

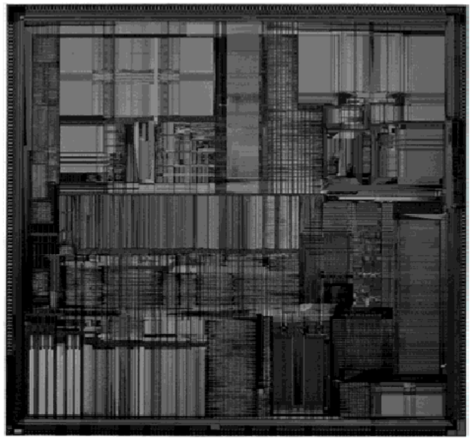
- □ Why Digital?
    - Advantages
    - Some applications
- □ History & Roadmap
- □ Device Technology & Platforms
- □ System Representation
- □ **Design Flow**
- □ RTL Basics

VLSI Design Flow

□Set of specification:

- What does the chip do?
- How fast does it run?
- How reliable will it be?
- How is the silicon area?
- How much power will it consume?
- ……

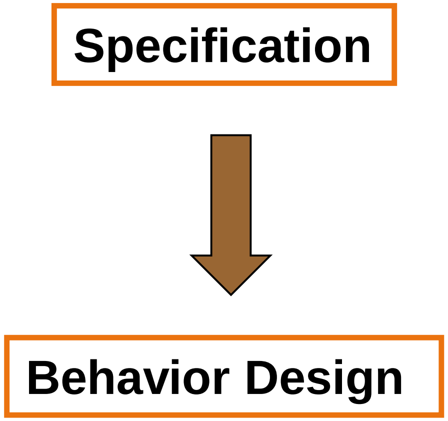




Time-to-Market
Parasitics
Millions of Gates
Power
Reliability

VLSI Design Flow

- An iterative process that transfer the specification to a manufacturable chip through at least five levels of design abstraction.



Specification

Function, performance, definition: English

Behavior Design

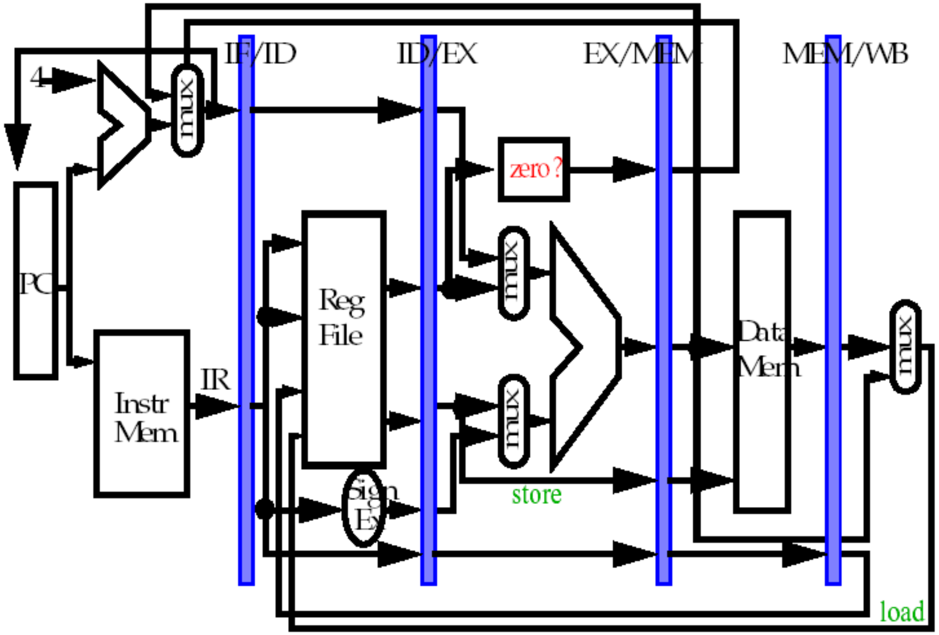
Behavior, algorithm: C/C++, SystemC, Matlab …

Behavior simulation

Architecture
VHDL
Verilog

Register Transfer Level Design

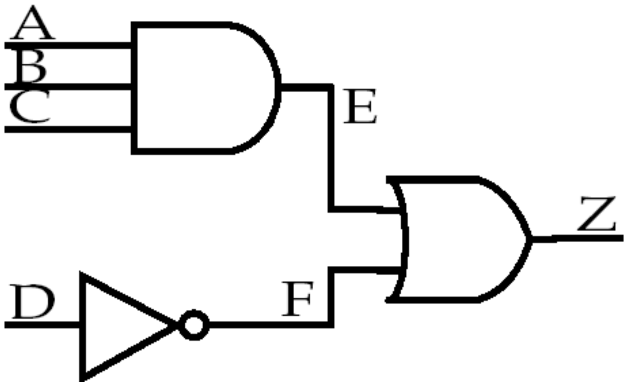
RTL simulation



Synthesis

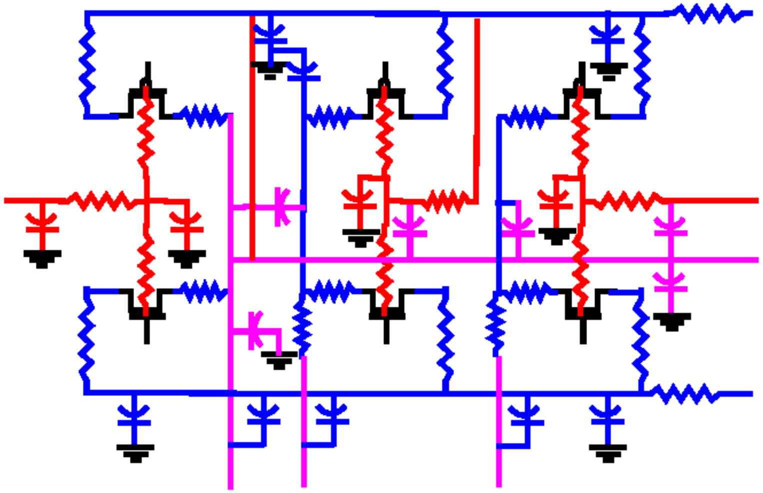
Logic Design

Gate-level simulation
Timing analysis
Power analysis



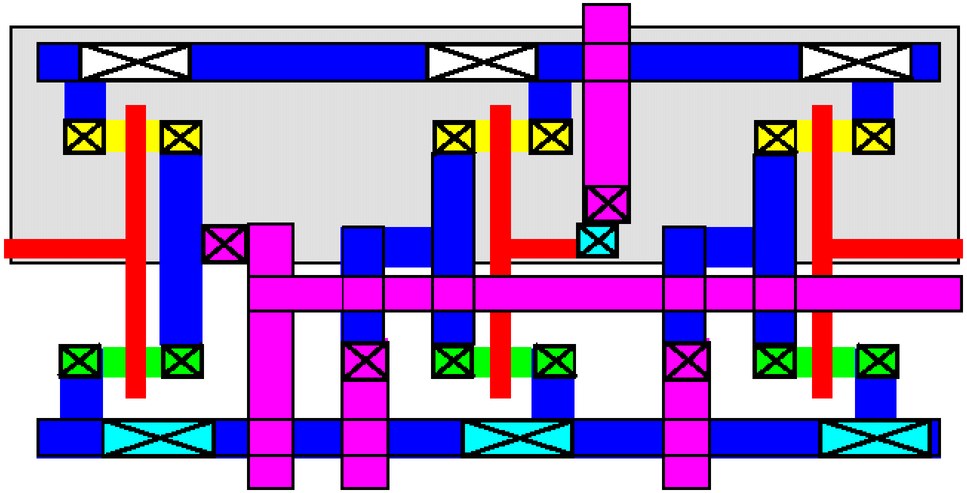
Custom Design

Circuit Design



Layout
(Place & Route)

Physical Design



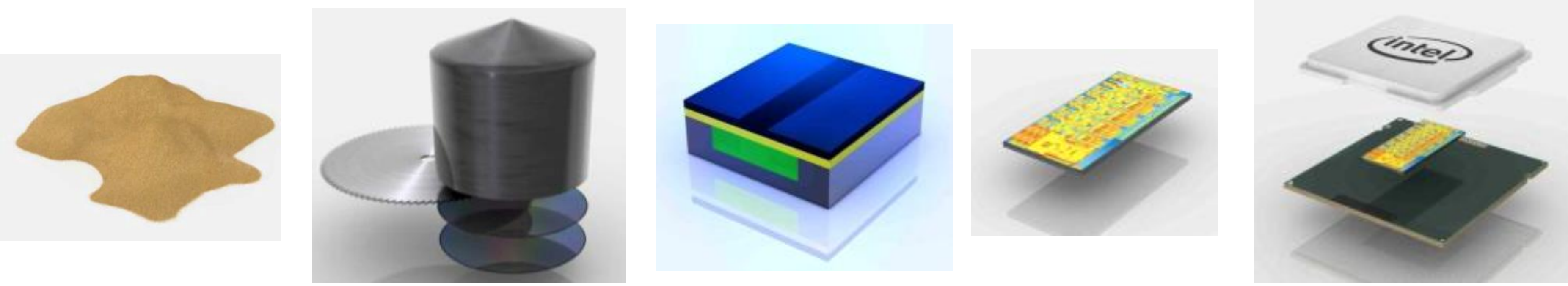
Design rule checking
Post layout simulation

Verification at ALL levels !!!

- □ **Verification**
  - Check whether a design meets the specification and performance goals
- □ **Two aspects**
  - ***Functionality***
  - ***Performance (timing/power/area)***
- □ **Method of Verification**
  - **Simulation**
    - □ ***Spot check: cannot verify the absence of errors***
    - □ ***Can be computation intensive***
  - **Timing analysis**
    - □ ***Just check delay***
  - Formal verification
    - □ ***E.g, equivalence checking***
  - **Hardware emulation**
  - **…**

Fabrication

From Sand to Silicon: the Making of a Chip | Intel

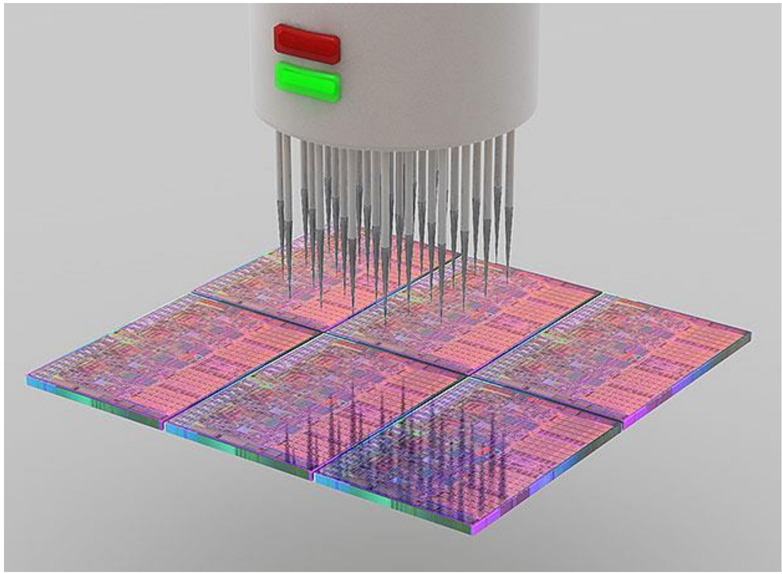


http://download.intel.com/newsroom/kits/chipmaking/pdfs/Sand-to-Silicon_32nm-Version.pdf

https://www.youtube.com/watch?v=Q5paWn7bFg4

Testing

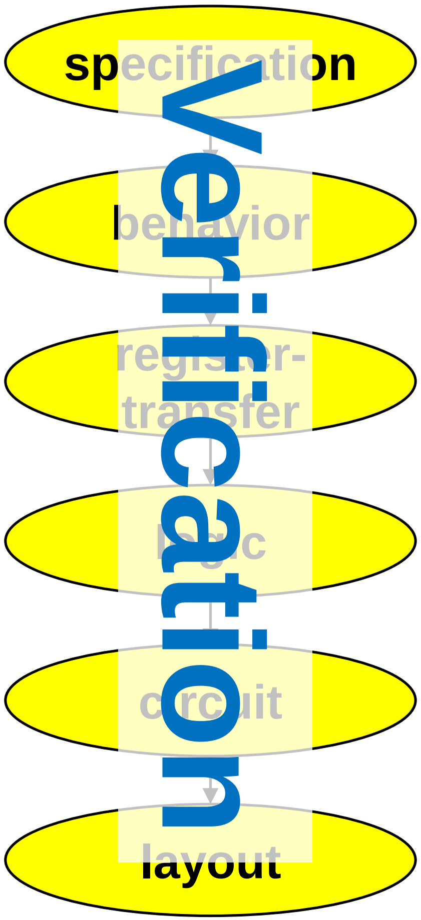
- □Testing is the process of detecting physical defects of a die or a package occurred at the time of manufacturing
- □Testing and verification are different tasks.
- □Difficult for large circuit
    - •Need to add auxiliary testing circuit in design
    - •E.g., built-in self test (BIST), scan chain etc.



VLSI Design Flow: Tools

- **Algorithm**
  - **Matlab**
- **RTL Simulation**
  - **Modelsim, Mentor**
  - VCS, Synopsys
  - VerilogXL, Cadence
- **Logic Synthesis**
  - **Design Compiler, Synopsys**
  - Blast Create, Magma
- **Transistor Simulation**
  - **Hspice/Starsim, Synopsys**
  - Spectra, Cadence
  - Eldo, Mentor
- **Mixed-Signal Simulation**
  - **AMS Designer, Cadence**
  - ADMS, Mentor
  - Saber, Synopsys
- **Place & Route**
  - Astro, Synopsys
  - **Silicon Encounter, Cadence**
  - Blast Fusion, Magma
- **Layout**
  - **Icfb/Dracula, Cadence**
  - ICstation/Calibre, Mentor
- **FPGA**
  - ***Vivado, Xilinx***
  - Quatus, Altera

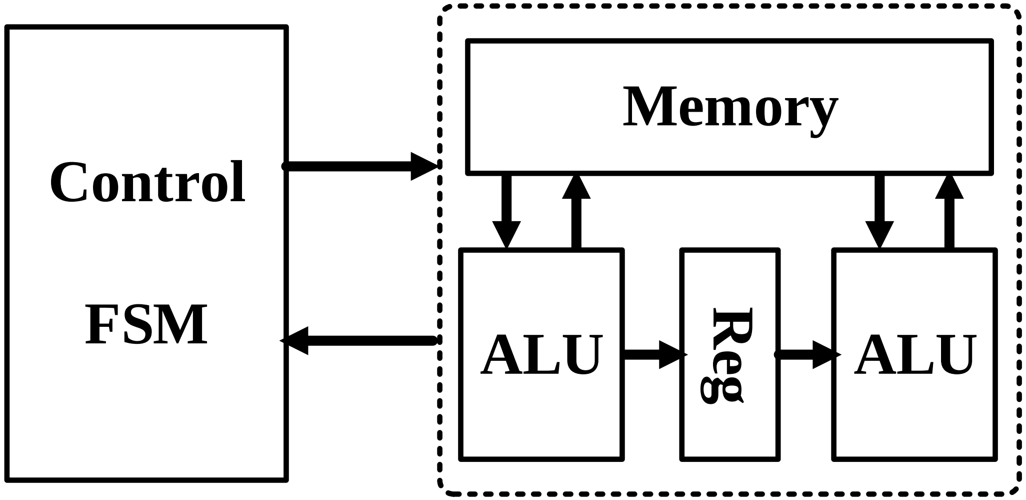
VLSI Design Flow (This course): Summary



English	PDF
Executable program	Matlab/C/Pen&Paper
HDL	Emacs/UltraEdit/Modelsim
Logic gates	
Transistors	Xilinx Vivado
Rectangles	

Following slides should fresh up your memory

Overall VLSI Structure



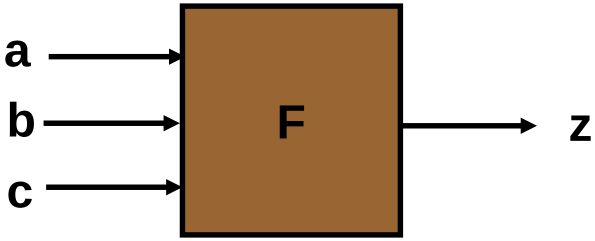
```
IF (a>10)
   b = c + d;
ELSE
   b = c - d;
```

- Scheduling / ordering / sequencing of operations
- Mapping / allocation:
  - Variables -> {Reg1, ... ,RegN}
  - Operations -> {MUL, ADD, ALU, ... ,}

We will implement something similar in this course

Two Basic Digital Components (What)

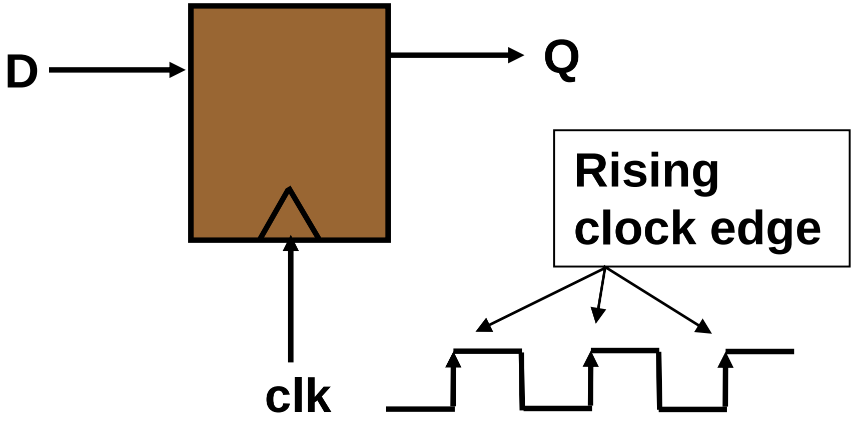
Combinational Logic



```
Always:
            z <= F(a, b, c);
```

i.e. a function that is always evaluated when an input changes. Can be expressed by a truth table.

Register



```
if clk’ event and clk=‘1’ then
      Q <= D;
```

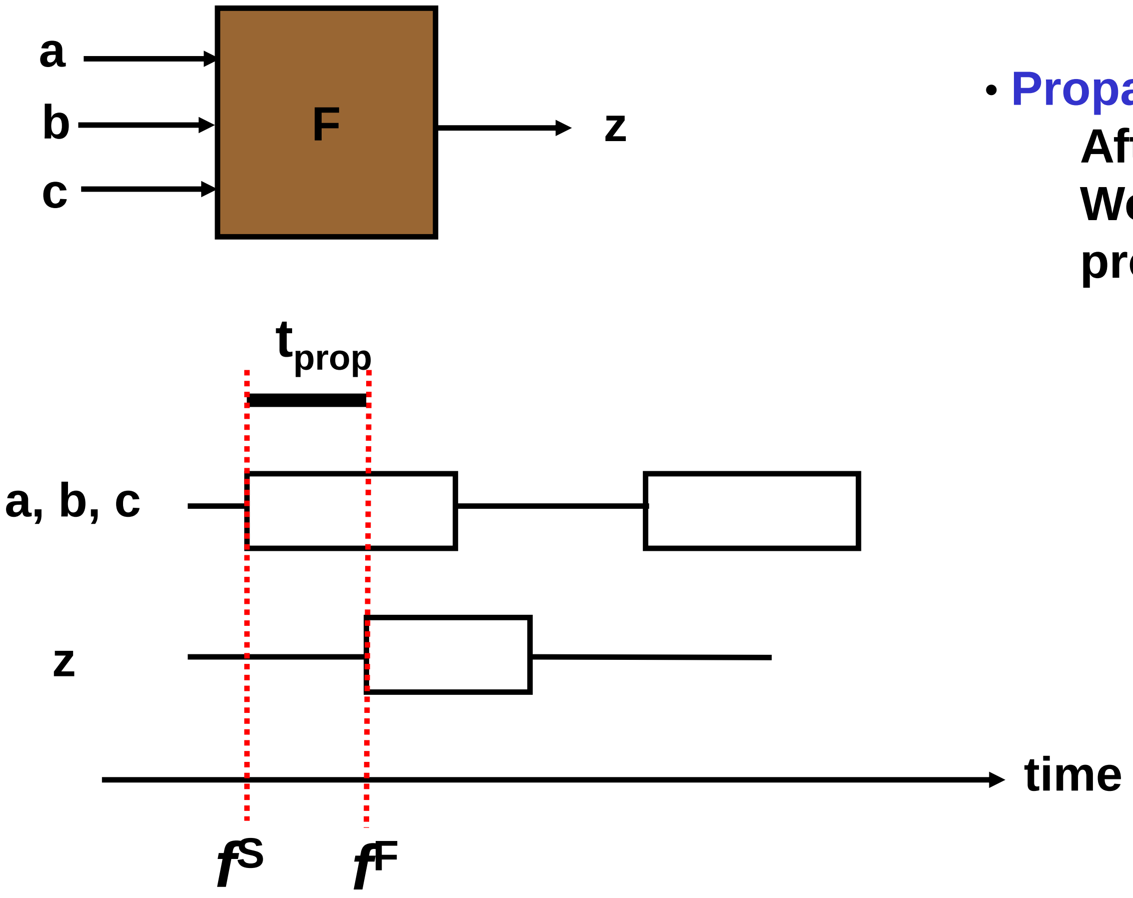
i.e. a stored variable, Edge triggered D Flip-Flop with enable.

Timing (When)

Only if we guarantee to meet the timing requirements

... do the components guarantee to behave as intended.

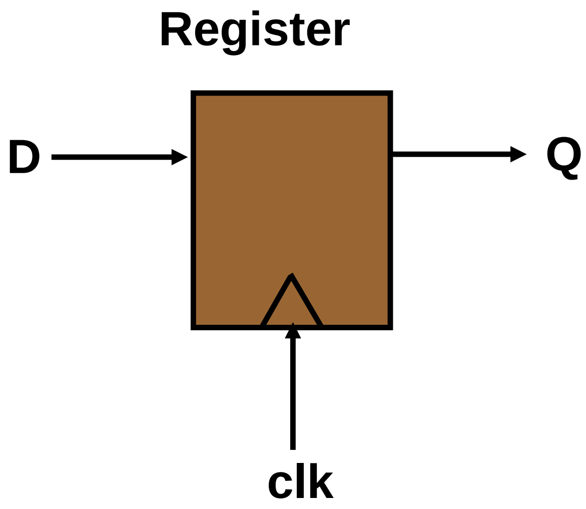
Combinational Logic Timing

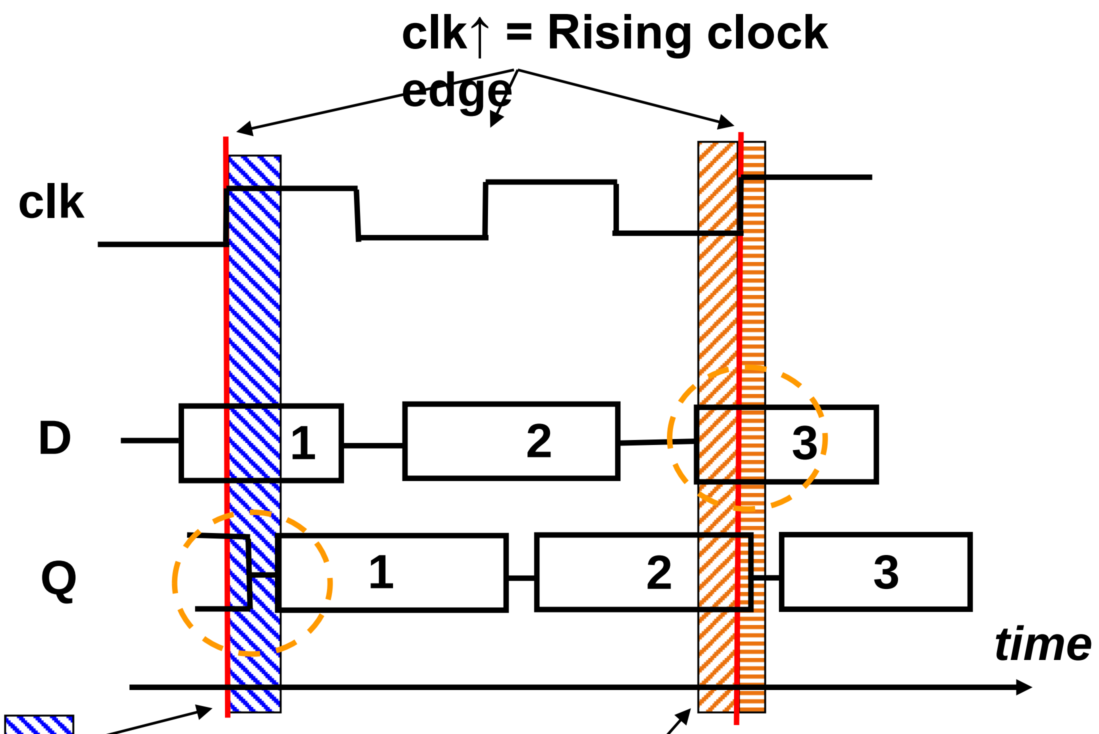


- **Propagation delay:**

  After presenting new inputs
  Worst case delay before
  producing correct output

Register timing





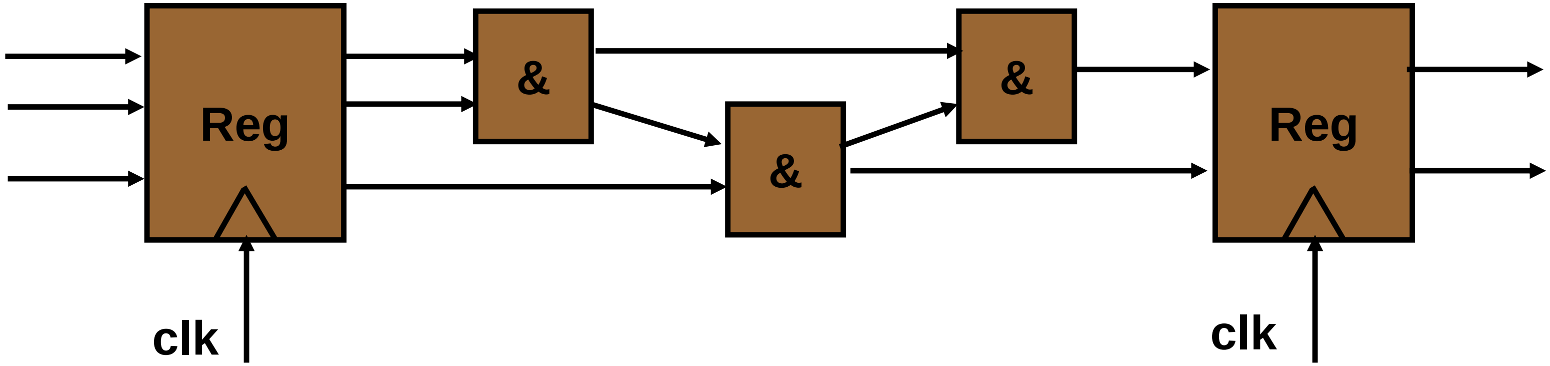
Propagation delay (clk_to_Q): ▨
Worst case (maximum) delay after clk↑ before new output data is valid on Q.

Setup time: ▨
Minimum time input must be stable before clk↑

Hold time: ▤
Minimum time input must be stable after clk↑

Clock Frequency (RTL)

- What is the maximum clock frequency?



Register

Propagation delay:	Tckl-Q	250ps
Setup time:	Tsu	200ps
Hold time:	Th	100ps

AND-gate

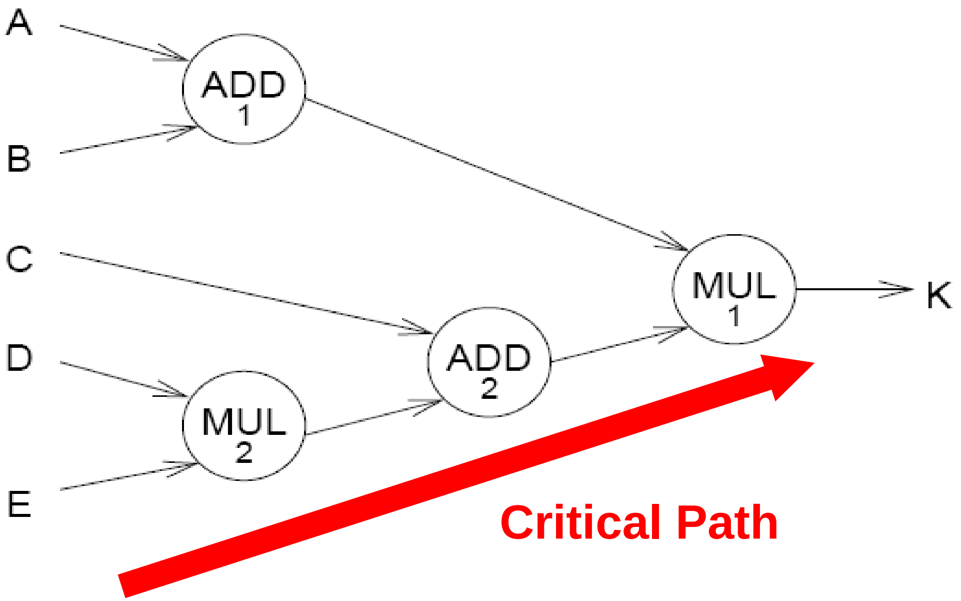
Propagation delay:	Tprop	250ps

250+250×3+200=1.2ns

***f*=833MHz**

Critical path

- …begin to explore the construction of digital systems with complex behavior
  - Example: $K = (A +_1 B) *_1 (C +_2 D *_2 E)$
- Combinational circuit:



Critical Path

Thanks