Fundamentals of VLSI

Introduction

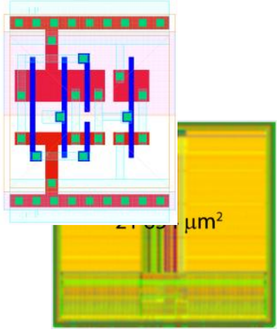
Andreas Burg

Telecommunications Circuits Laboratory,
EPFL, Switzerland

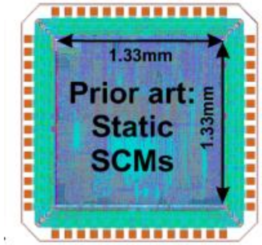
Course Content and Objectives

How to design a complex digital integrated circuits using modern design tools:

- The basic building blocks (transistor-level design)
  - The basic building blocks (logic gates, memories, …)
  - Basic performance metrics and behavior of digital circuits



- Designing complex circuits with design abstraction
  - Design abstraction with Hardware Description Languages
  - Froentend: From HDL to a digital integrated circuit
  - Backend: Physical design



At the end of the course, you should be able to

- Understand all steps involved in the design process
- Design an integrated circuit from specification to tapeout
- Understand its behavior and optimize its performance

Course Content and Objectives

The course is divided into two parts:

Part-1: Fundamentals of digital integrated circuits

- Design on transistor level => **Full custom design**
- Consider the basic building blocks
- Investigate the behavior of digital circuits on a small scale

Part-2: Managing complexity with abstraction

- Design on the gate level => **Semicustom design**
- Handle complex designs with millions of transistors
- Understand and learn how to use the design tools
- Design and optimization techniques

Detailed Content

Introduction

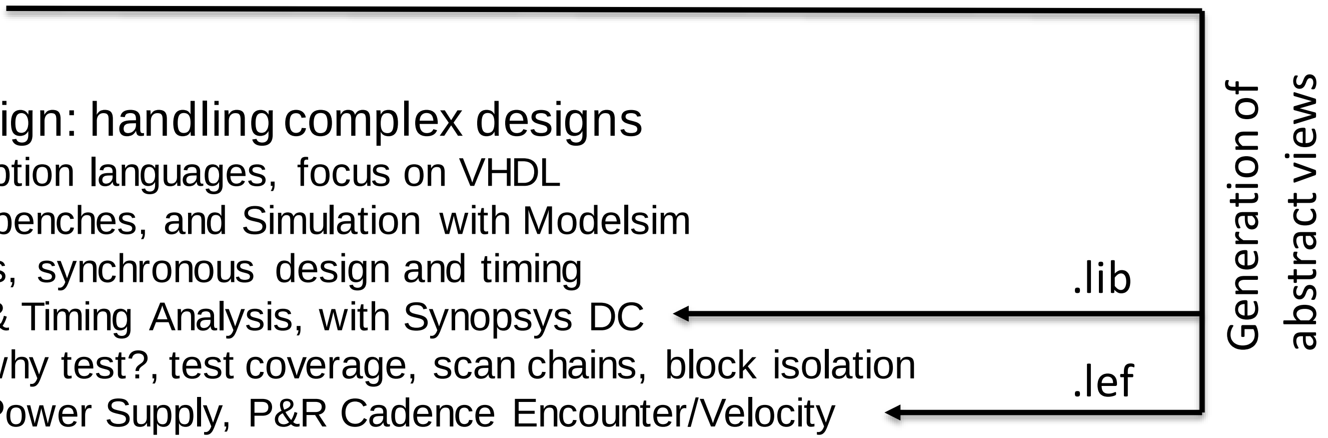
- Course outline, history and trends in VLSI design
- How to design a chip: objectives, metrics, design methodology, tools, and ecosystem

Fullcustom design: basic logic gates and their characteristics

- CMOS inverter and other logic gates: static and dynamic behavior, power consumption
- Sequential elements: timing requirements
- Standard cells

Semicustom design: handling complex designs

- Hardware description languages, focus on VHDL
- Verification, Testbenches, and Simulation with Modelsim
- Timing disciplines, synchronous design and timing
- Logic synthesis & Timing Analysis, with Synopsys DC
- Design for test: why test?, test coverage, scan chains, block isolation
- Floor Planning, Power Supply, P&R Cadence Encounter/Velocity
- Clock Tree Insertion and Advanced Clocking: clock skew, clock gating, multiple clocks, post-CTS timing analysis and timing optimization, reset tree
- Power analysis, Postlayout simulation
- Design for low power
- DRC/LVS



Logistics

- Classroom lectures cover 50% of the course hours and can be Thursday and/or Friday

- Mandatory exercises/Labs in irregular intervals based on content of the course and can also be Thursday and/or Friday (announced on Moodle)
- Homework is the completion of the exercises

- **We highly recommend you attend also the optional EDA-TP on Tuesady since it provides a lot of practical background and additional practice for the exercises**

Logistics: Exams

- Graded exercises will count 50% toward the final grade
- Final exam counts 50% of your grade
- Final exam will be similar to the exercises plus a quizz
- Hint: make sure you attend all lectures and exercises

Fundamentals of VLSI

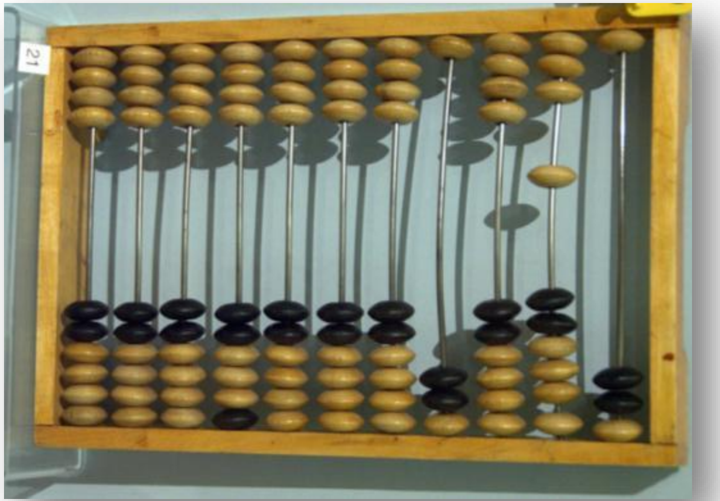
Lecture 1

Andreas Burg

Telecommunications Circuits Laboratory,
EPFL, Switzerland

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

History of Digital Circuits



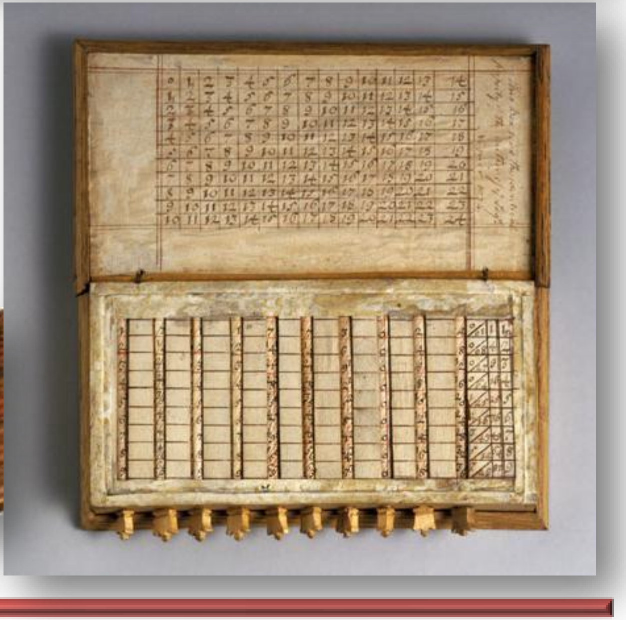
The Abacus:

- The first computation device
- Invented around 2400 BCE
- Still in use today

❑ Napier’s Bones:

» Invented by ***John Napier*** (~1590)

» Addition, Multiplication, Logarithms



❑ Slide Rule:

» Introduced in 1620

» Analog Computer

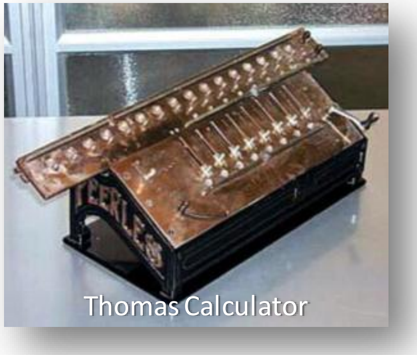
History of Digital Circuits

Binary Logic

- ***Pingala*** discovered the ***Binary Numeral System*** (~300 BCE India)
- ***Leibniz*** described ***Binary Logic*** (~1650 Germany)
- ***Boolean Algebra*** was published by ***George Boole*** in 1854

Gottfried Wilhelm von Leibniz

❑ Mechanical Calculators



Thomas Calculator

» First ***calculator*** by ***Schickard*** (1623), followed by ***Pascal*** and ***Leibniz***.

» First ***mass-produced calculator*** by ***Thomas*** (1820)

History of Digital Circuits

Punch Cards

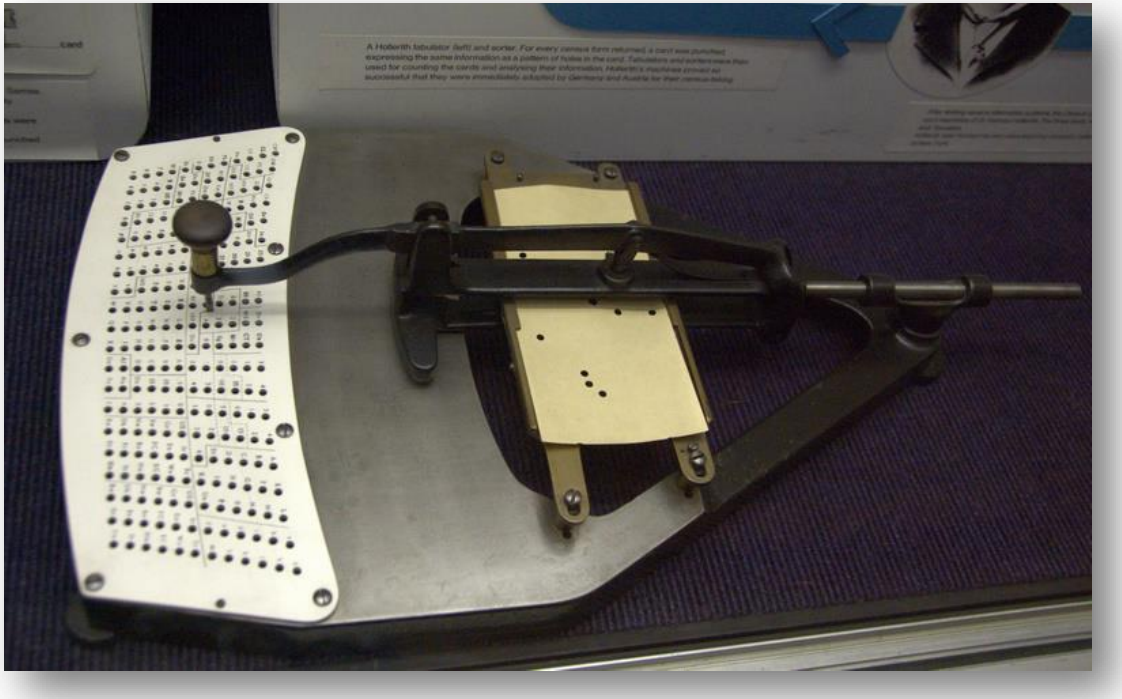
Jacquard Loom

- In 1725 ***Bouchon*** developed an ***Automatic Loom*** based on holes in paper.
- In 1801, ***Jacquard*** enabled using ***punch cards*** to control such a loom.
- In 1822, ***Charles Babbage*** described the ***Difference Engine,*** which is considered the first real computer design, though it was only made in 1991 (it is still operational at the London Science Museum).
- In 1834, ***Babbage*** described the ***Analytical Engine*** based on punch cards and a steam engine. It was the first general purpose programmable computer.

History of Digital Circuits

20th Century Milestones

- 1896 - ***Herman Hollerith*** establishes the ***Tabulating Machine Company***, later to become ***IBM*** (1924).



Hollerith Punch Card Machine

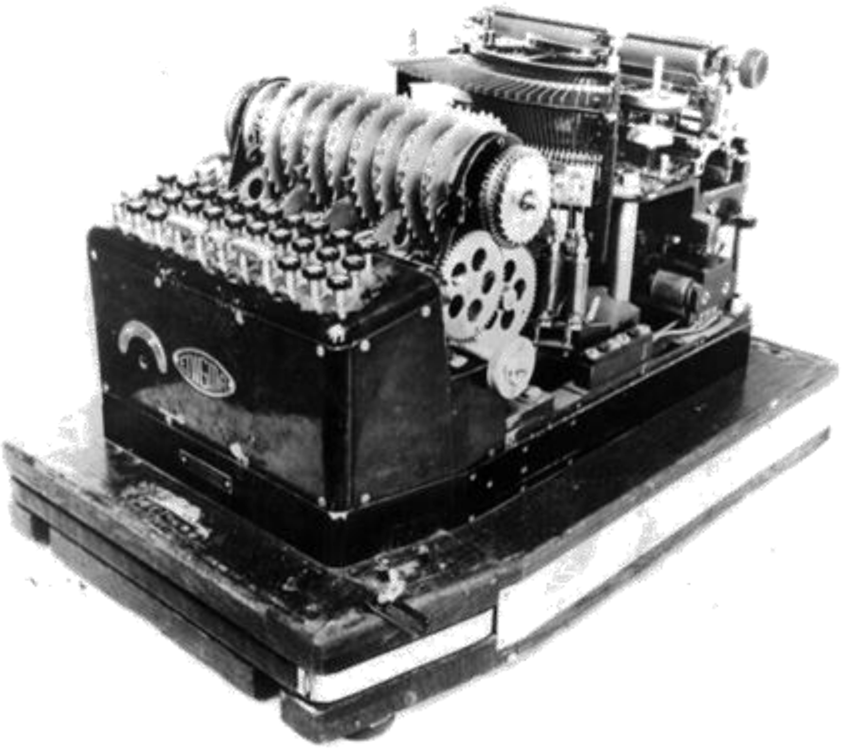
History of Digital Circuits

20th Century Milestones

- 1906 – The ***Electronic Valve*** is invented (***De Forest***). This is the switch that enabled the development of the digital computer.
- 1919 – The ***Flip Flop*** was proposed (***Eccles***, ***Jordan***).
- 1937 – ***Alan Turing*** publishes paper describing the "***Turing Machine***" and sets the basis for computer theory. Turing is considered "The Father of Modern Day Computing"

History of Digital Circuits

The Alan Turing Memorial Statue in Manchester



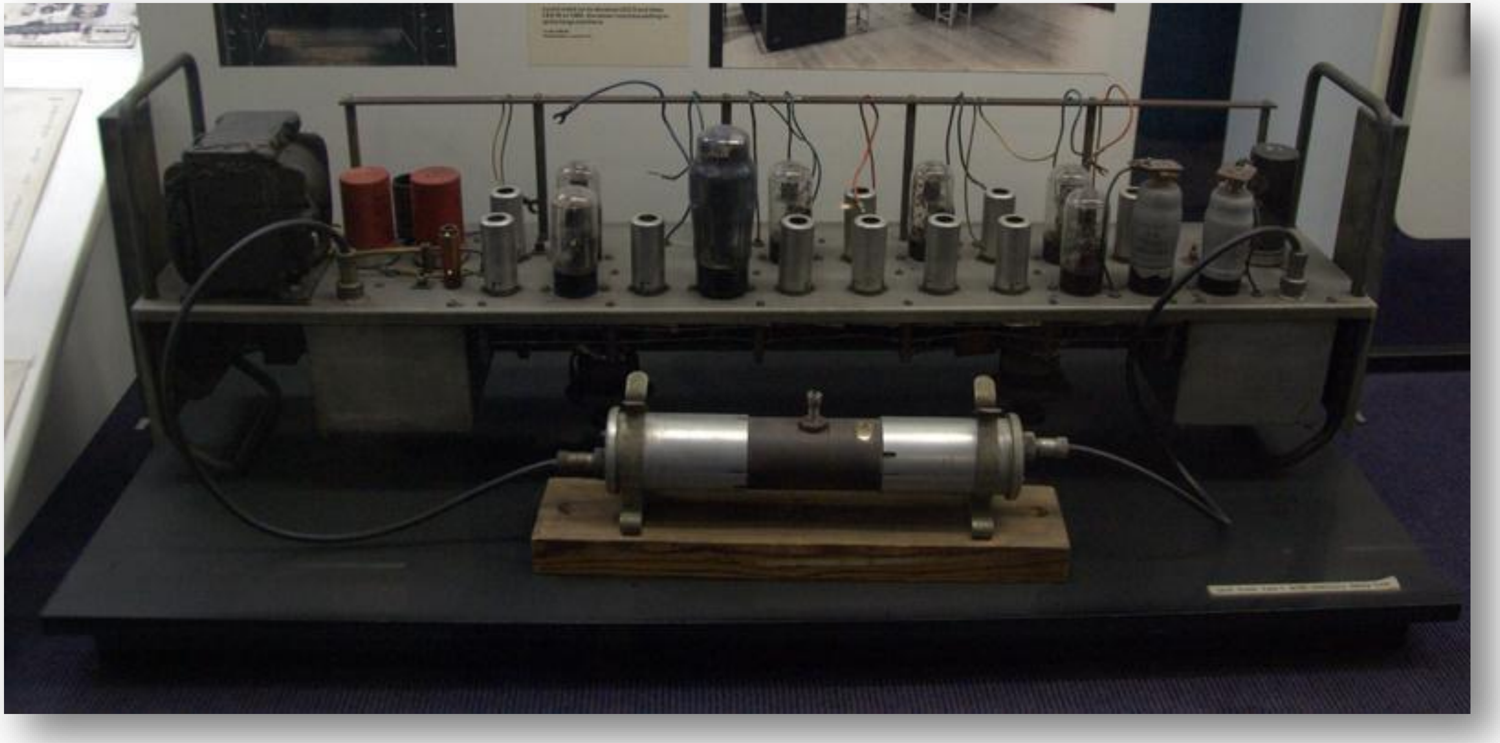
The Enigma

German Encryption Machine that Turing helped decipher

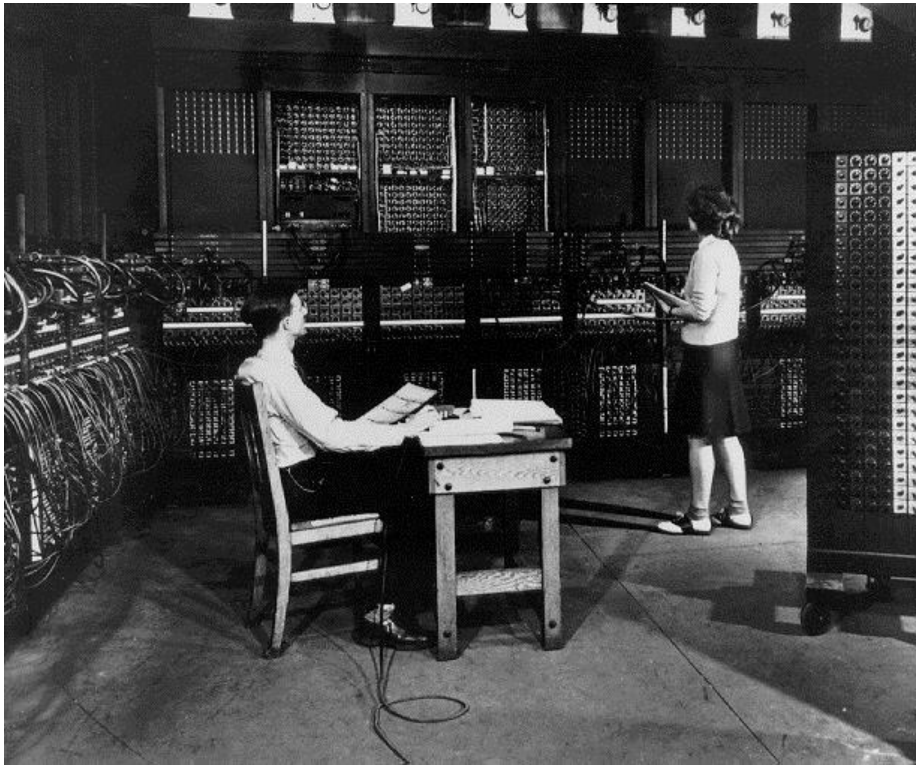
History of Digital Circuits

20th Century Milestones

- 1939 - First machine to calculate using ***vacuum tubes*** developed.



History of Digital Circuits



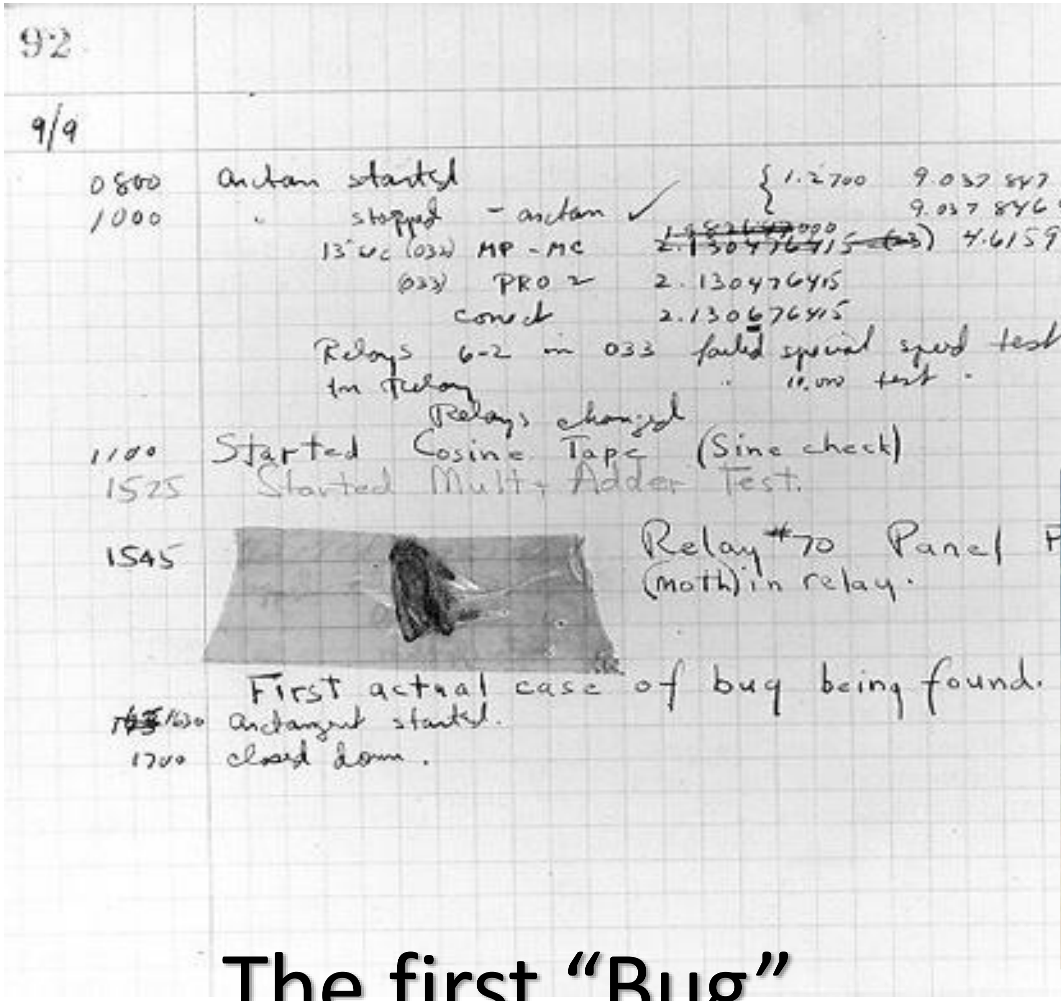
ENIAC (1946)

Considered the first Universal Electronic Computer. Used 18,000 electronic valves, weighed 30 Tons and consumed 25kW of power. Could do approximately 100,000 calculations a second. Intended to compute artillery firing tables (military)

UNIVAC-1 (1951)

First commercially successful electronic computer. Also, first general purpose computer. Worked with magnetic tapes.

History of Digital Circuits



92

9/9

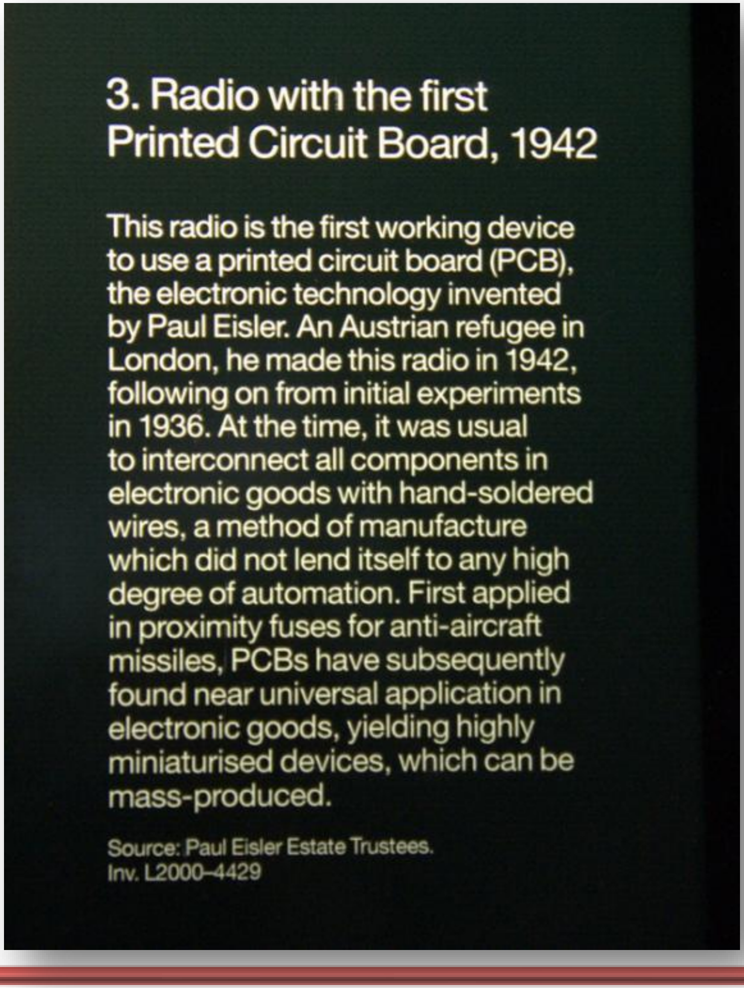
0800 Anctan started {1.2700 9.037 847 025
1000 " stopped - anctan ✓ 9.037 846 995 correct
13"uc (032) MP - MC ~~1.982147000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
In Relay " 11.000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult+ Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1600~~ 1630 Antangent started.
1700 closed down.

The first "Bug"

Grace Brewster Murray Hopper

Inventor of the infamous Bug!

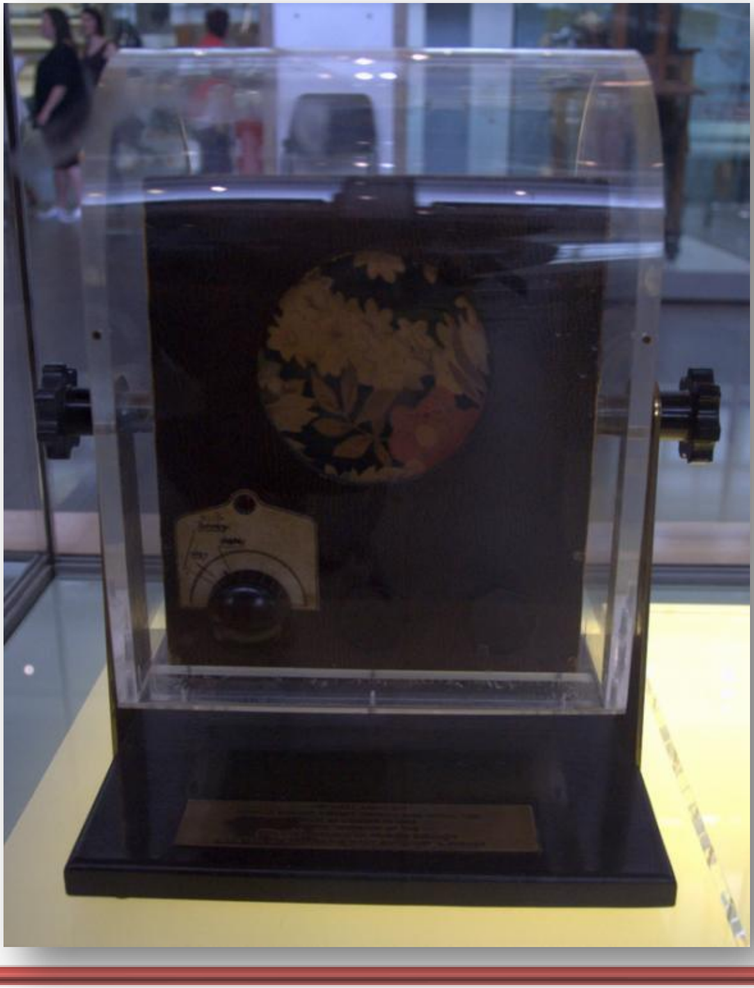
Radio with first Printed Circuit Board (1942)



3. Radio with the first Printed Circuit Board, 1942

This radio is the first working device to use a printed circuit board (PCB), the electronic technology invented by Paul Eisler. An Austrian refugee in London, he made this radio in 1942, following on from initial experiments in 1936. At the time, it was usual to interconnect all components in electronic goods with hand-soldered wires, a method of manufacture which did not lend itself to any high degree of automation. First applied in proximity fuses for anti-aircraft missiles, PCBs have subsequently found near universal application in electronic goods, yielding highly miniaturised devices, which can be mass-produced.

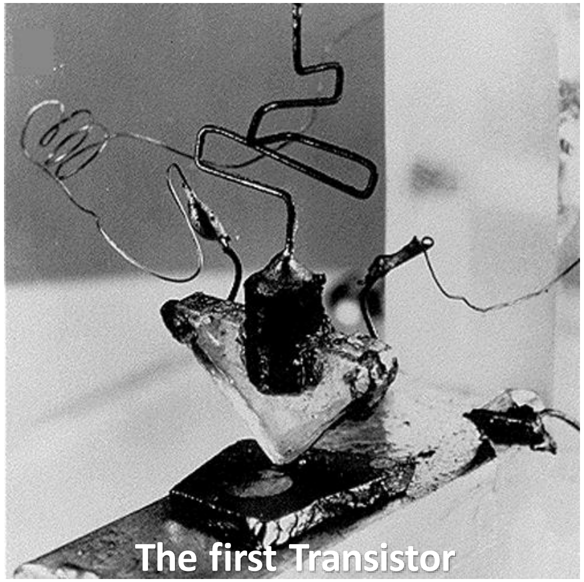
Source: Paul Eisler Estate Trustees.
Inv. L2000–4429



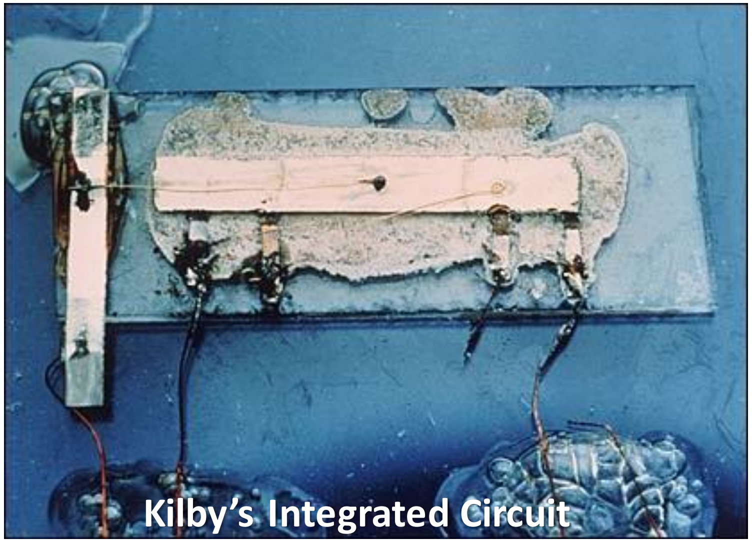
History of Digital Circuits

The Transistor Era

- 1947 – A group at Bell Labs, headed by ***Shockley***, invent the first ***transistor*** to replace the inefficient vacuum tube.
- 1952 – The idea of the ***Integrated Circuit*** was conceived by ***Dummer***.



The first Transistor



Kilby's Integrated Circuit

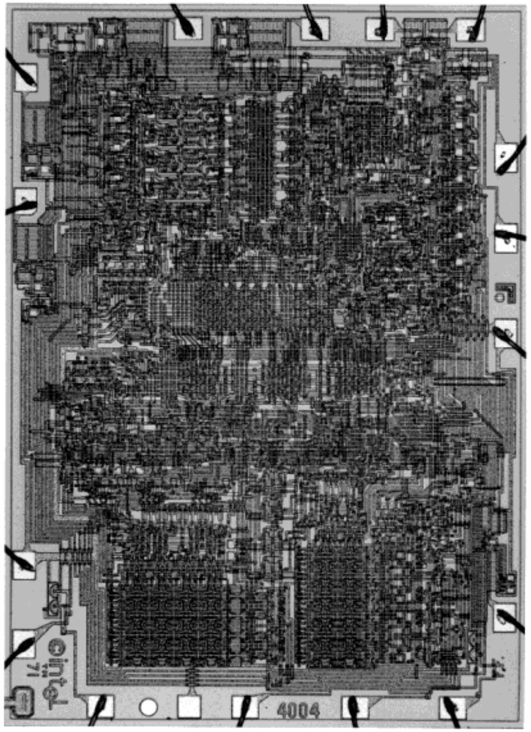
» 1958 – The first integrated circuit was invented by ***Jack Kilby*** of TI. The first ***silicon IC*** was invented by ***Robert Noyce*** of Fairchild half a year later.

History of Digital Circuits

The Transistor Era

- 1960 – First ***MOSFET*** Fabricated
- 1962 – ***TTL*** Invented
- 1963 – ***CMOS*** Invented (solve TTL Power issue)
- 1964 – 1-inch silicon wafers introduced
- 1965 – ***Moore's Law*** (more in a minute…)
- 1967 – ***Floating Gate*** invented
- 1970 – First commercial ***DRAM*** (1Kbit)
- 1971 – ***Microprocessor*** invented
- 1978 – Intel 8086/8088
- 1981 – ***IBM PC*** is introduced

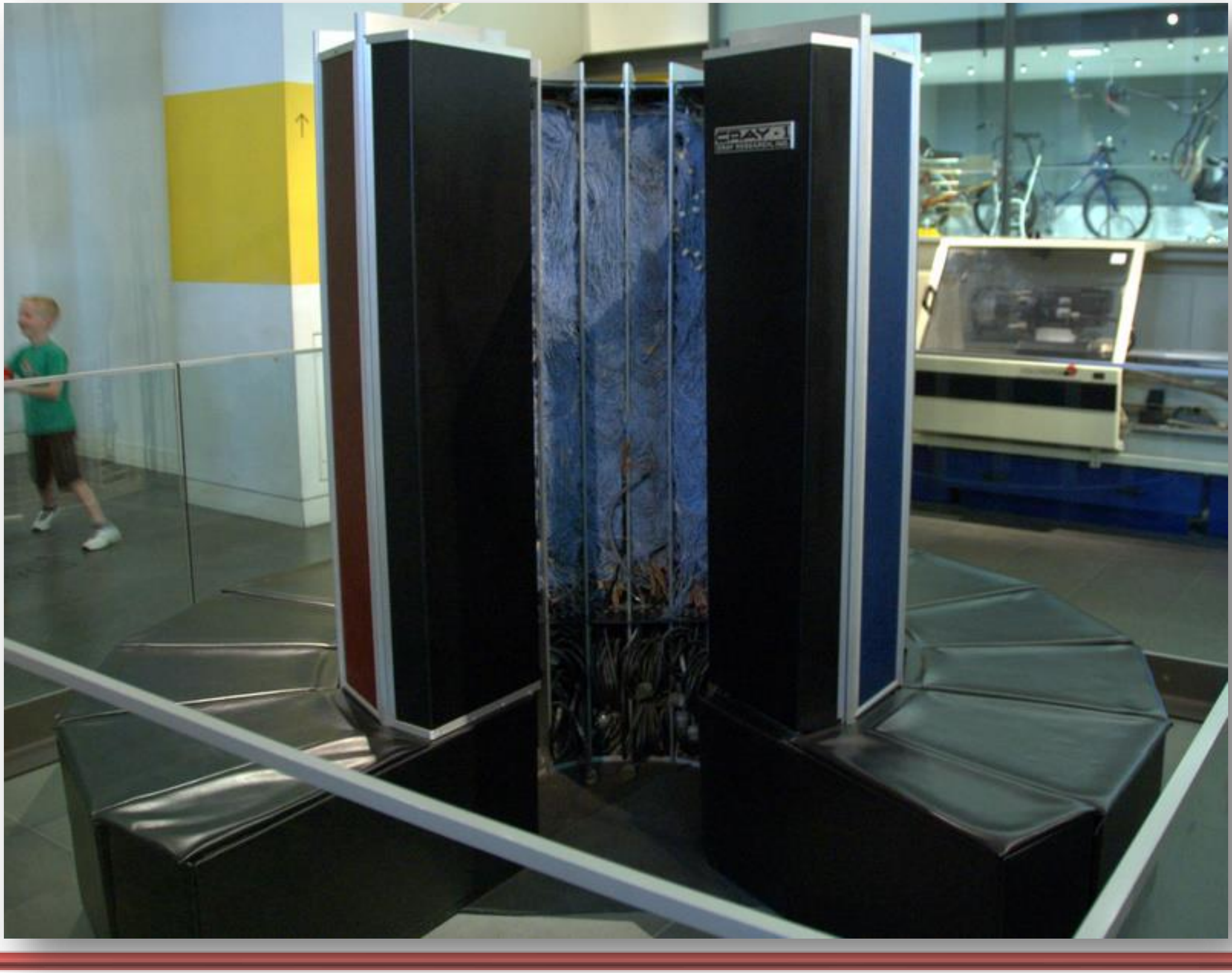
Intel 4004 (1971)
1000 transistors
1MHz operation



The Xerox Alto (1973)

- Mouse
- Graphical Display
- LAN
- WYSIWYG Editor
- Drawing Program
- Windows UI

Cray Supercomputer (1976)



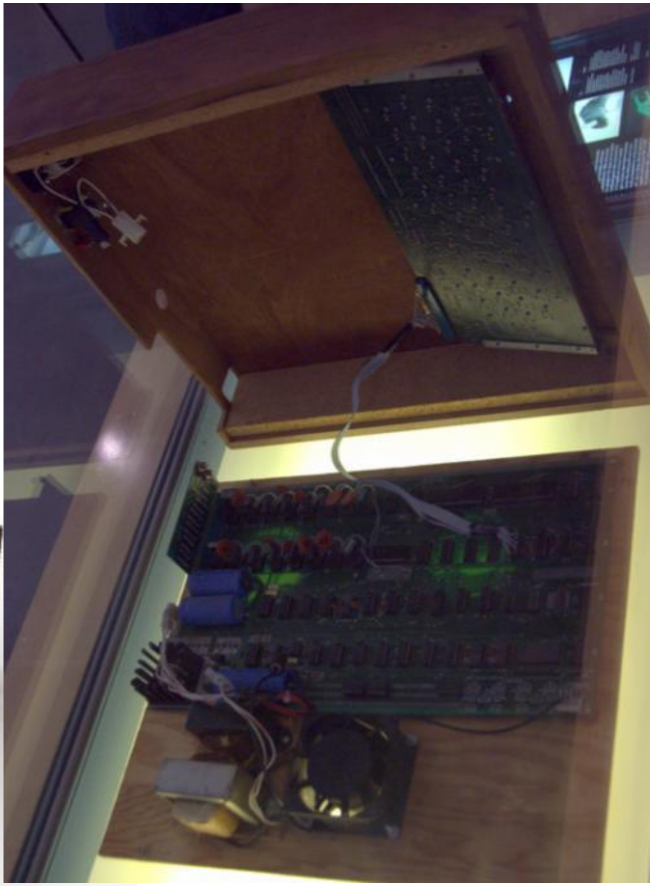
“Small” Computers

DEC PDP-8

The first “minicomputer”

The Apple 1

Great Great Great Grandfather of the iPhone…



History of Digital Circuits

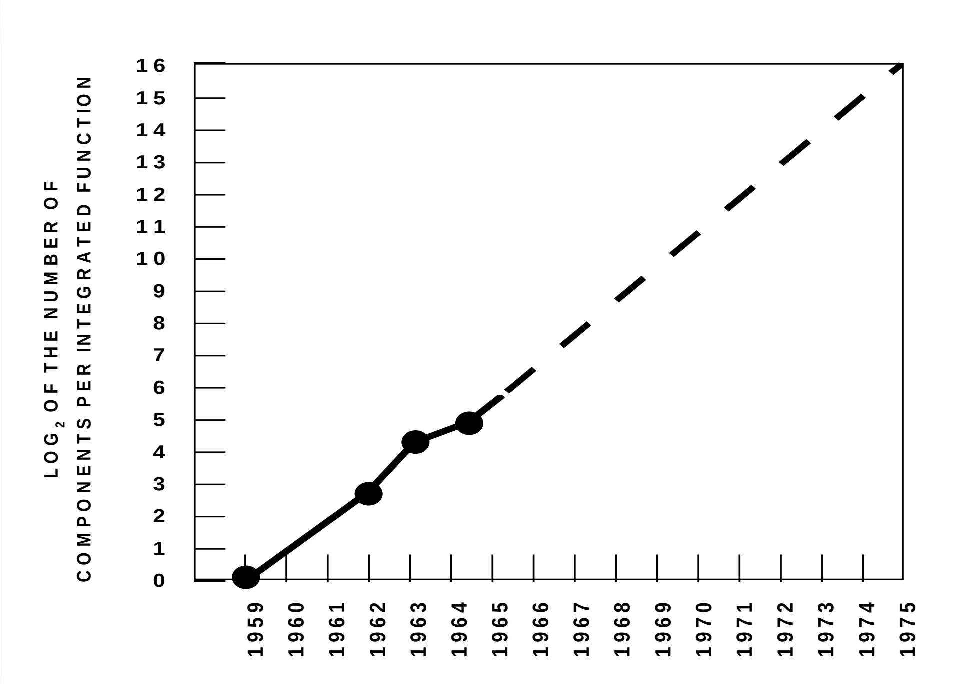
The IBM PC 5150 (1981)

Intel 8088

Moore's Law

In 1965, ***Gordon Moore*** noted that the number of transistors on a chip doubled every 18 to 24 months.

He made a prediction that semiconductor technology will double its effectiveness every 18 months



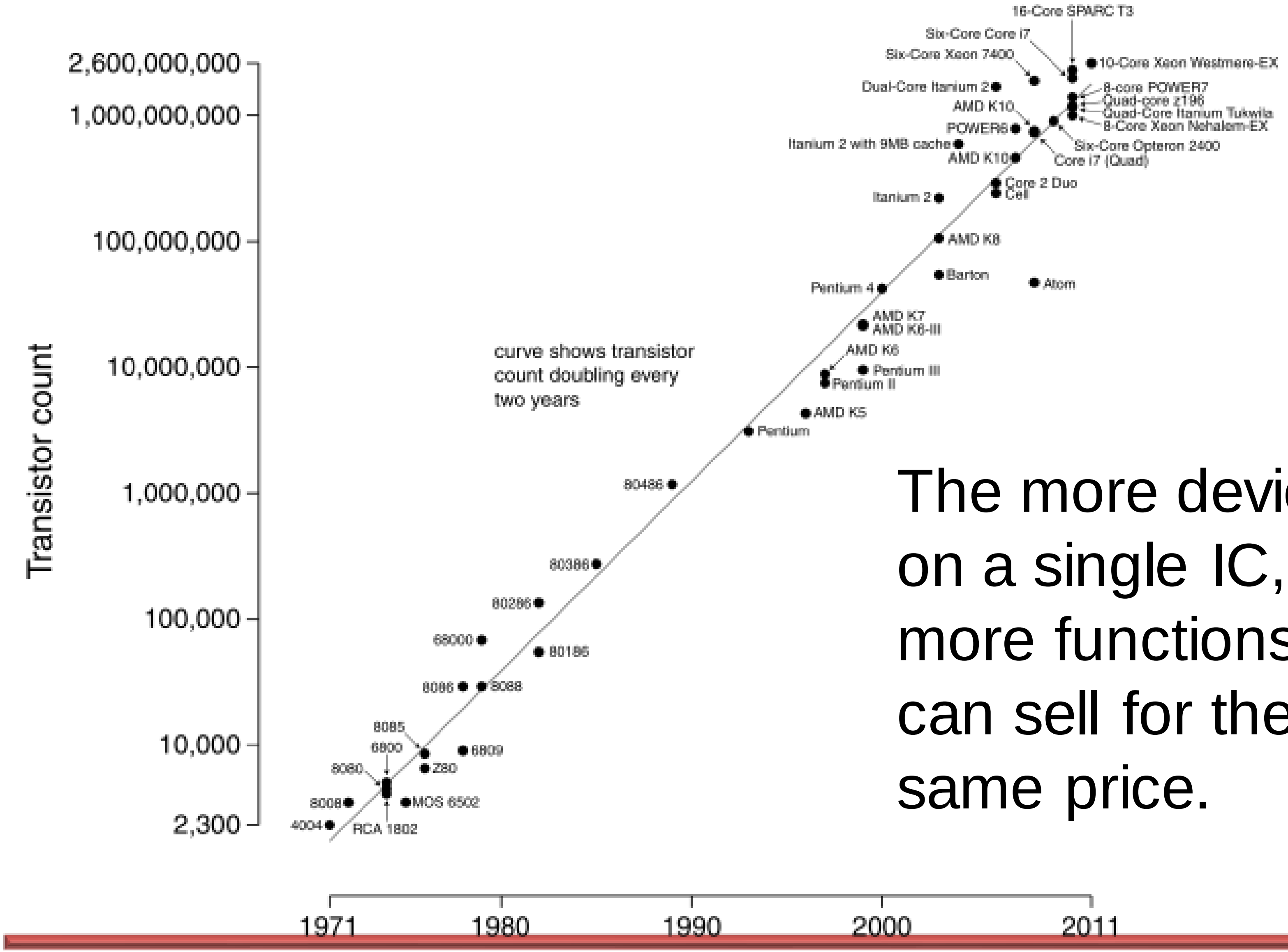
Electronics, April 19, 1965.

Moore’s Law Guides the Industry R&D

“*The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term, this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000*.”

Gordon Moore, Cramming more Components onto Integrated Circuits, (1965).

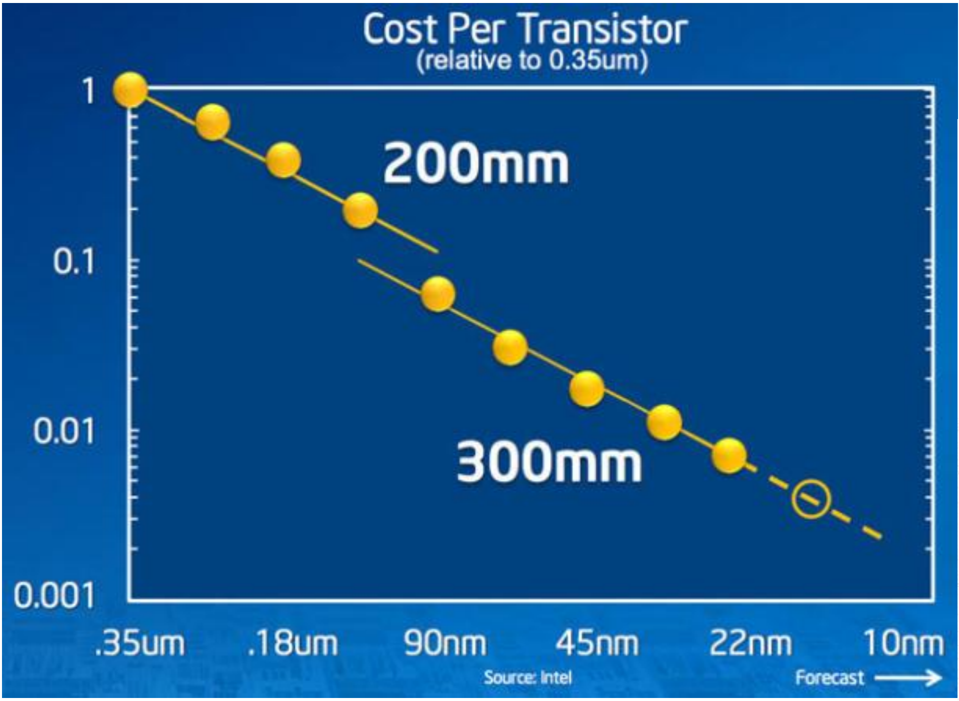
Moore’s Law – Transistor Count

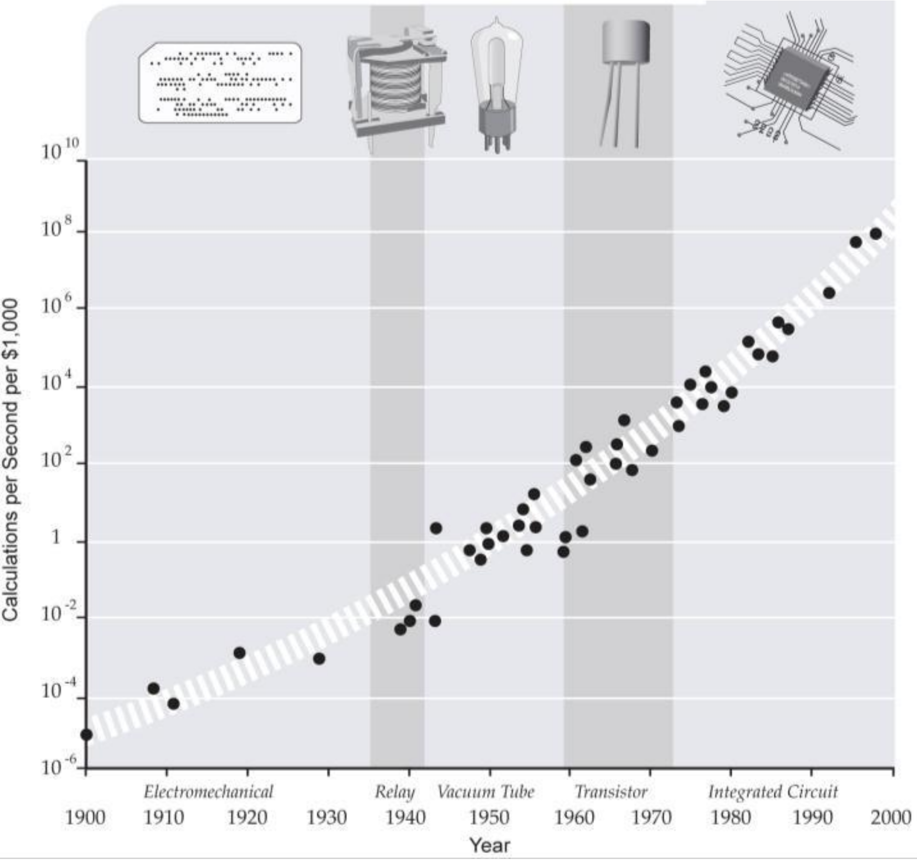


The more devices on a single IC, the more functions we can sell for the same price.

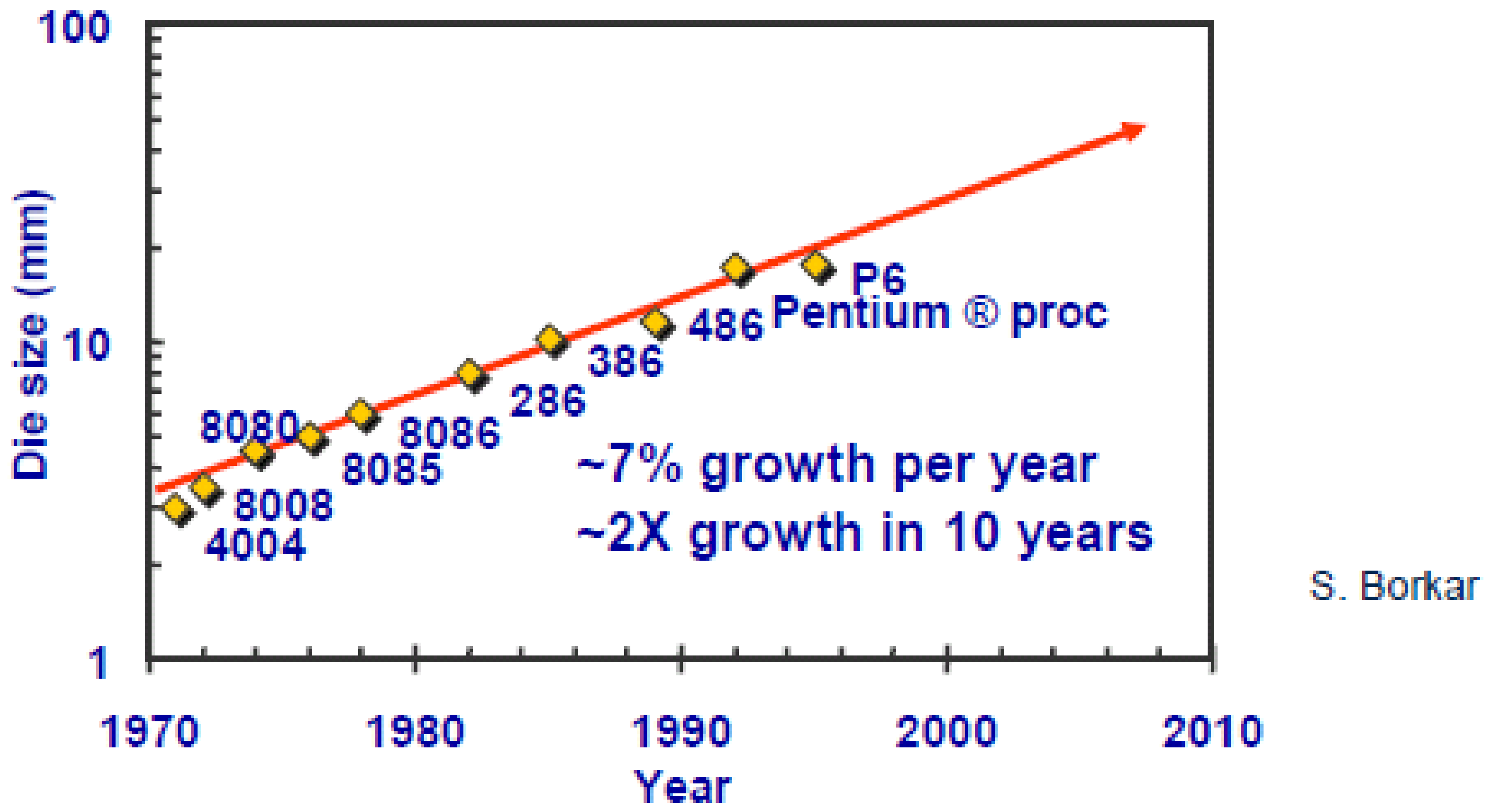
Moore's Law – Transistor Cost

Moore's Law is, at its base, an *Economical* law.





Moore's Law – Die Size



Die size growth by 14% to satisfy Moore's law

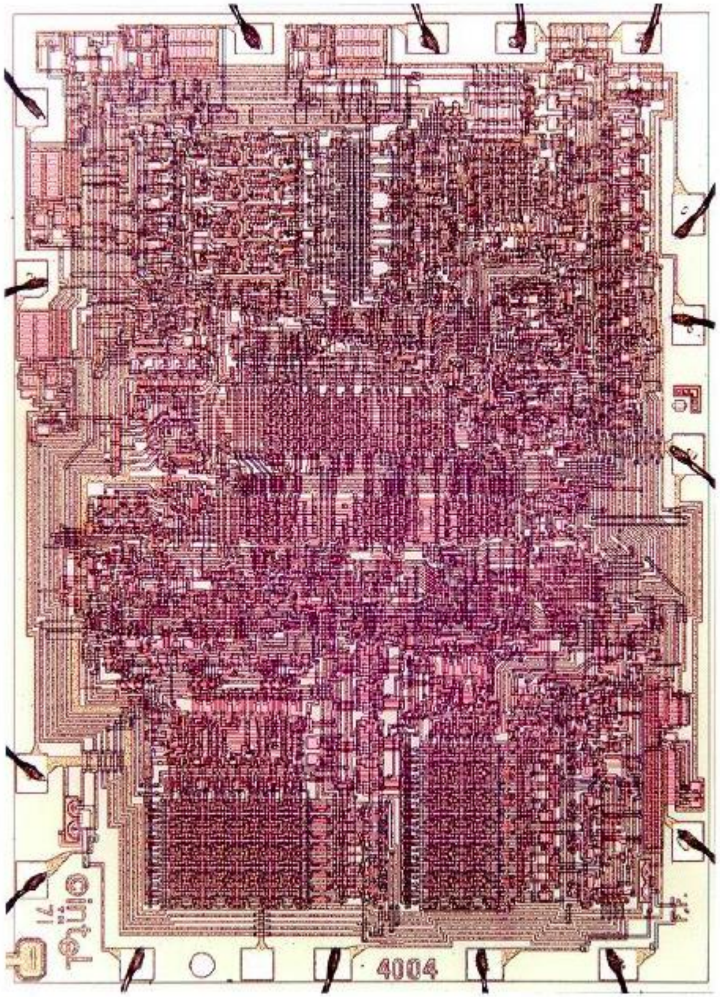
Illustration: Evoluation of Microprocessors

Intel 4004 Microprocessor

Introduced 1971
Clcok speed: 108 kHz

2300 Transistors

10um Technology



http://download.intel.com/pressroom/kits/IntelProcessorHistory.pdf

Illustration: Evoluation of Microprocessors

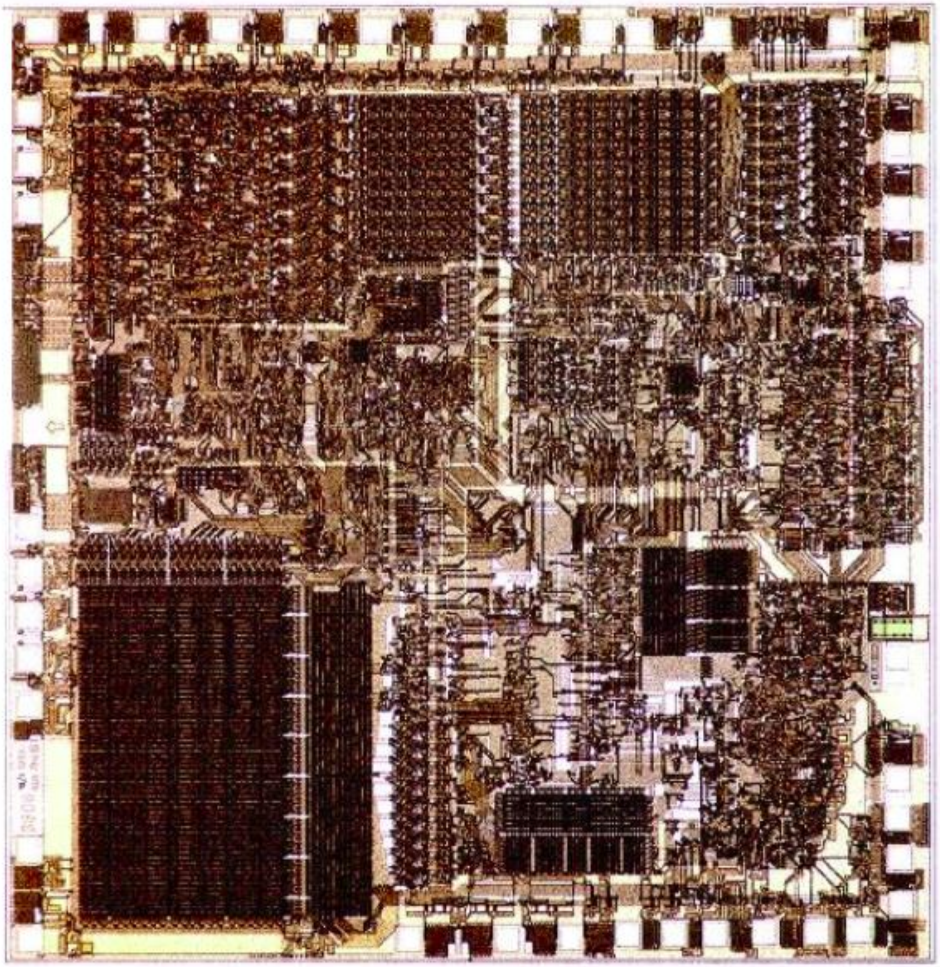
Intel 8088 Microprocessor

Introduced 1979

Clcok speed: 5 MHz

29’000 Transistors

3um Technology



http://download.intel.com/museum/exhibits/hist_micro/hof/large_jpegs/8088B1.jpg

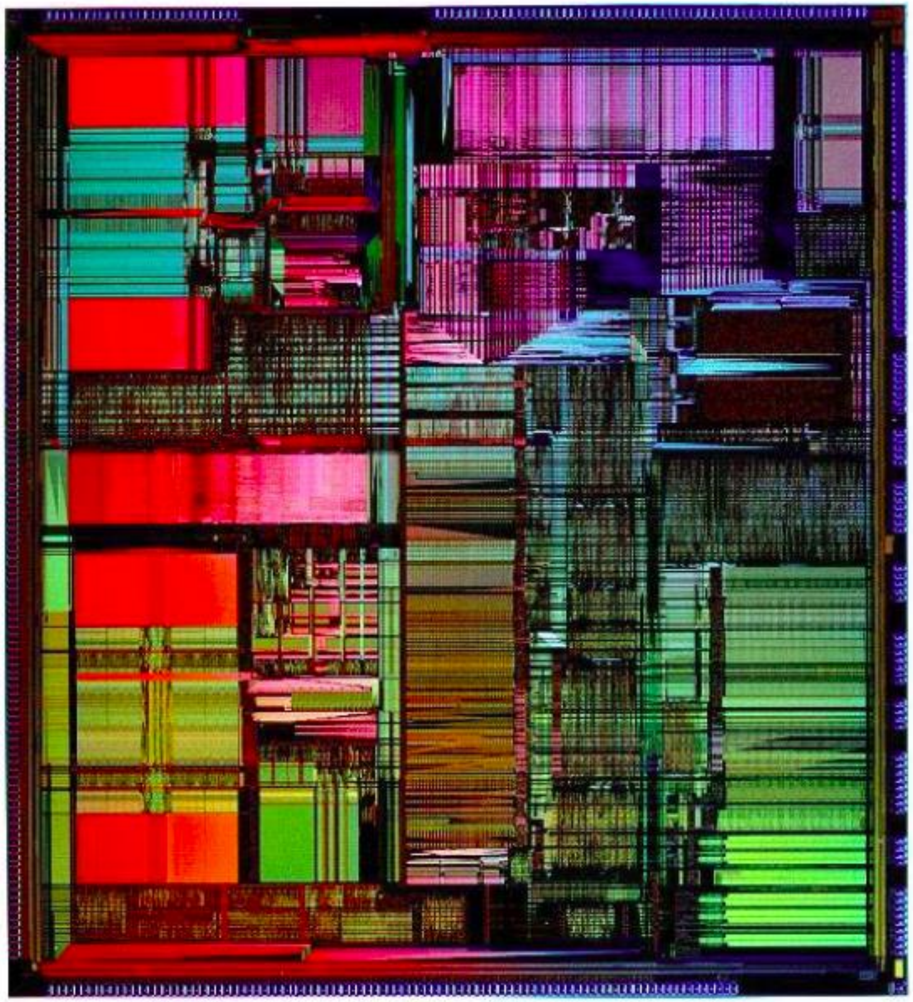
Illustration: Evoluation of Microprocessors

Intel Pentium Microprocessor

Introduced 1993
Clcok speed: 66 MHz

3.1 Mio Transistors

0.8um Technology



http://download.intel.com/museum/exhibits/hist_micro/hof/large_jpegs/pentB.jpg

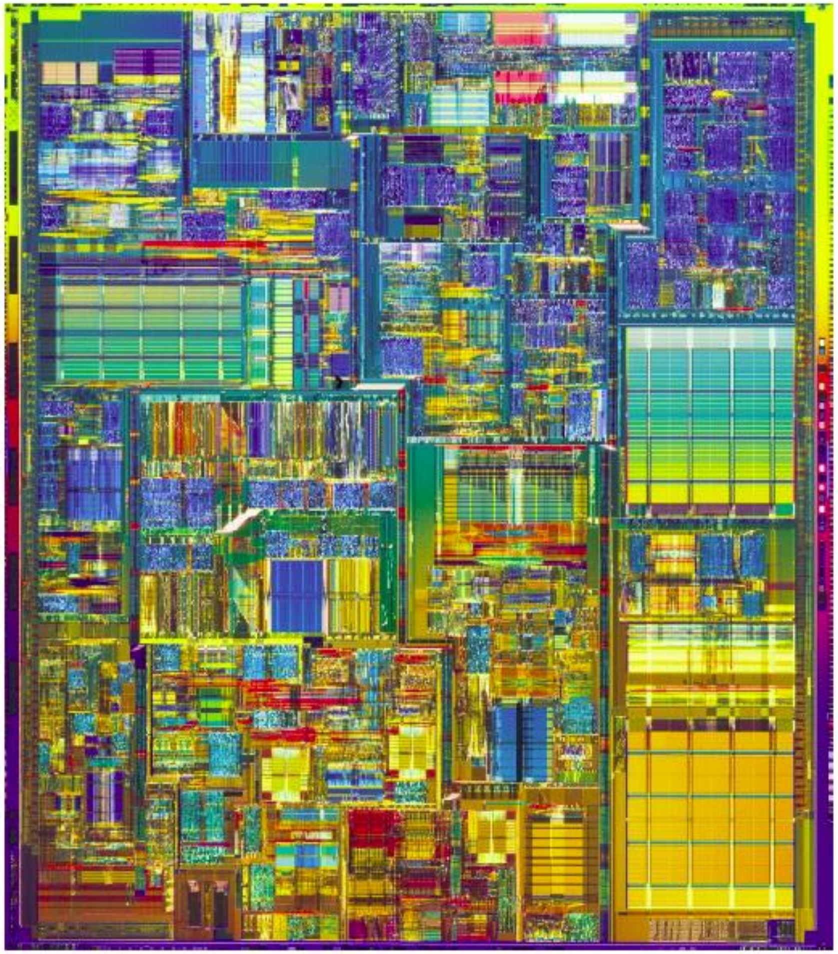
Illustration: Evoluation of Microprocessors

Intel Pentium 4 Microprocessor

Introduced 2000
Clcok speed: 1.5 GHz

42 Mio Transistors

0.18um Technology



http://download.intel.com/pressroom/kits/IntelProcessorHistory.pdf

Illustration: Evoluation of Microprocessors

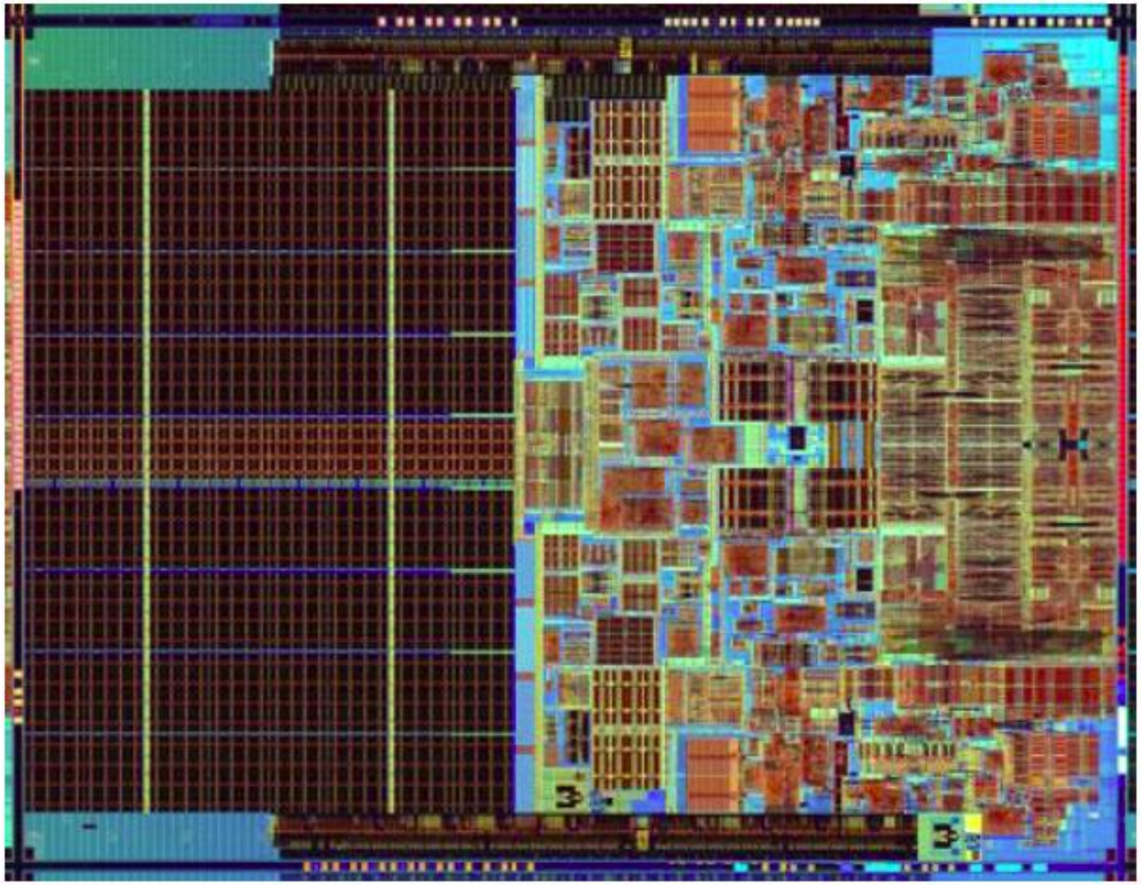
Intel Core 2 Duo Microprocessor

Introduced 2006
Clcok speed: 2.9 GHz

291 Mio Transistors

65nm Technology

Power: 65W



http://www.sudhian.com/img/intel/core2/core2.03.jpg

Illustration: Evoluation of Microprocessors

Quad-Core Intel Xeon Microprocessor

Introduced 2007
Clcok speed: >3 GHz

820 Mio Transistors

45nm Technology

Power: 45W

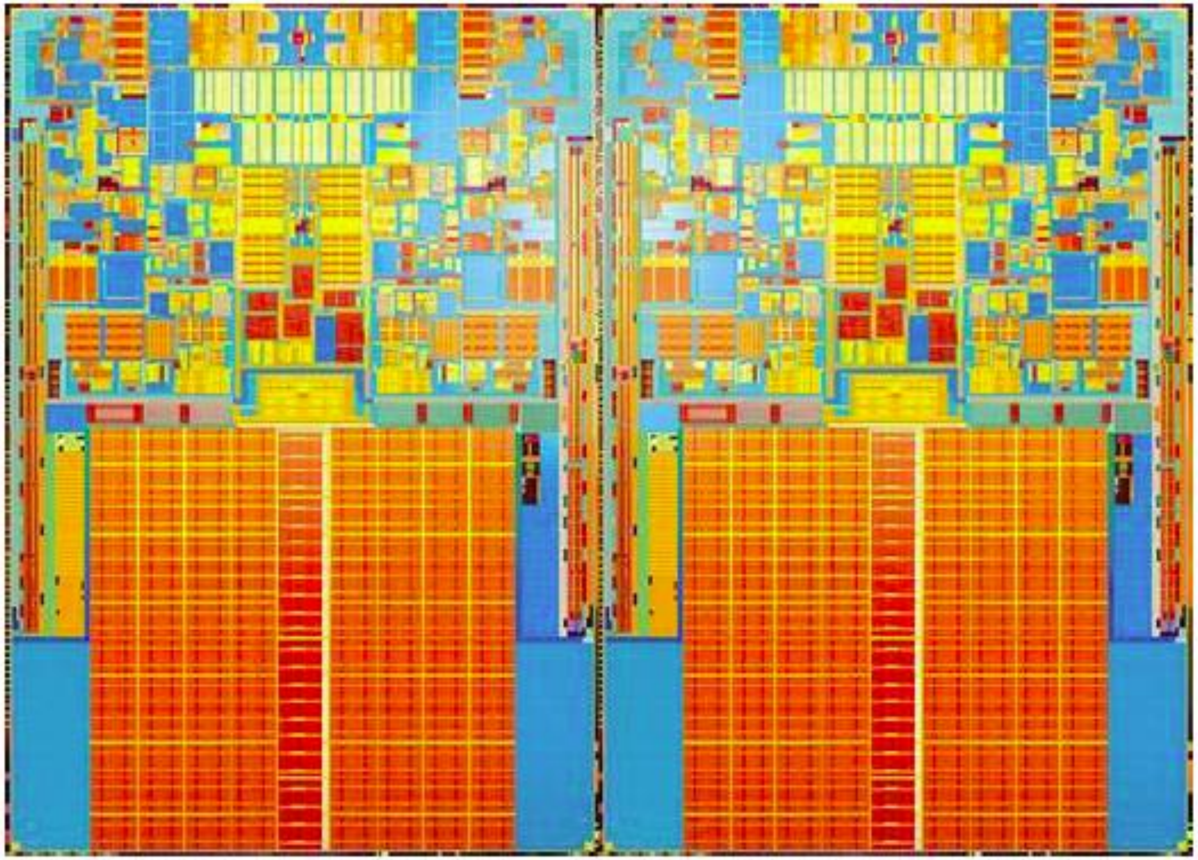


Illustration: Evoluation of Microprocessors

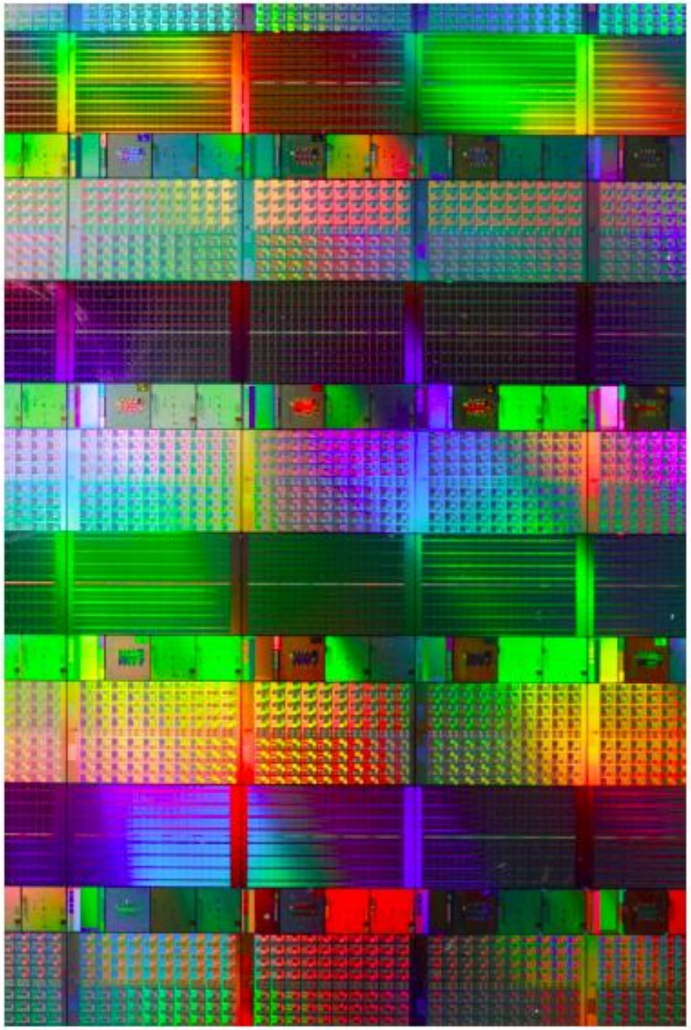
Teraflops Research Chip

Introduced 2006
65nm Technology

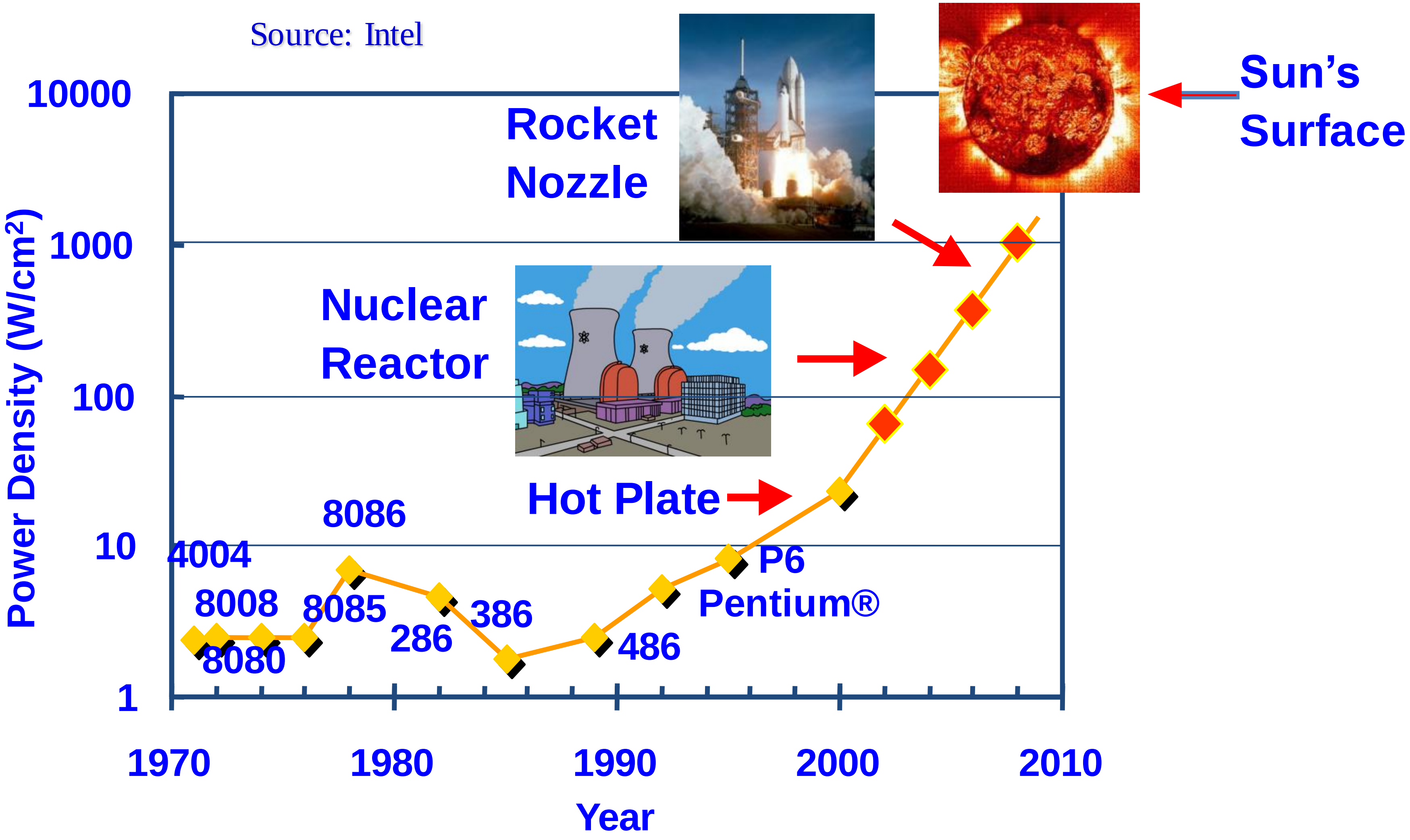
80 Processor cores

3.16 GHz	62W	1.0 Tflops
5.1 GHz	175W	1.6 TFlops
5.7 GHz	265W	1.8 TFlops

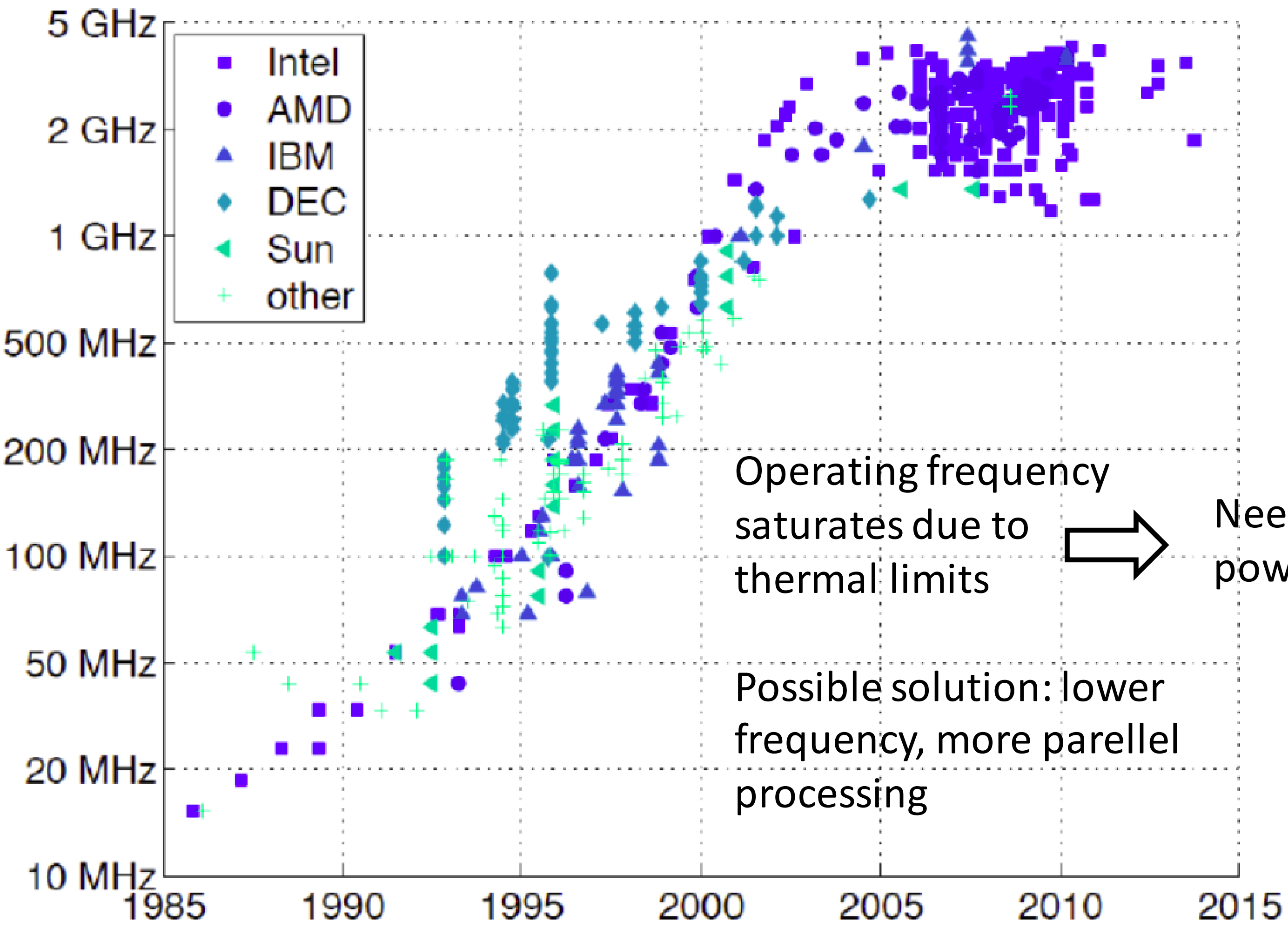
For comparison: ASCI Red was the first supercomputer to reach Tflops in 1996. That system used nearly 10,000 Pentium® Pro processors running at 200MHz and consumed 500kW of power plus an additional 500kW just to cool the room that housed it.



The Thermal Power Limit



The Thermal Power Limit

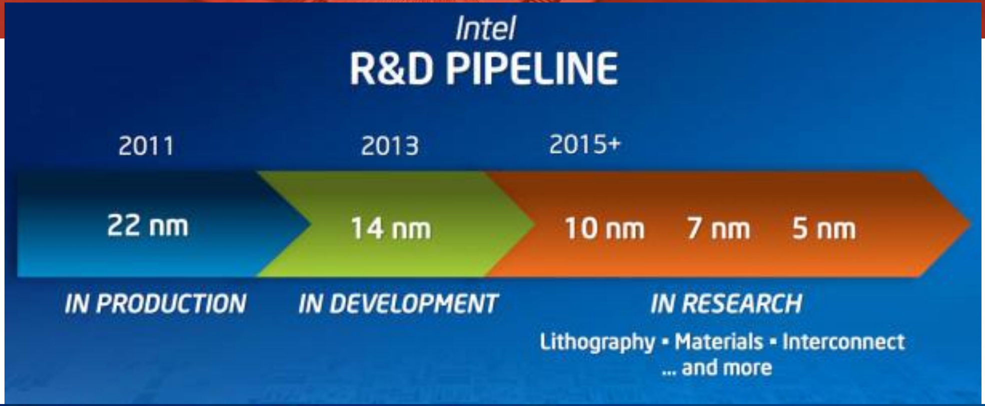


Operating frequency saturates due to thermal limits ⇨ Need for low-power design

Possible solution: lower frequency, more parellel processing

http://cpudb.stanford.edu/

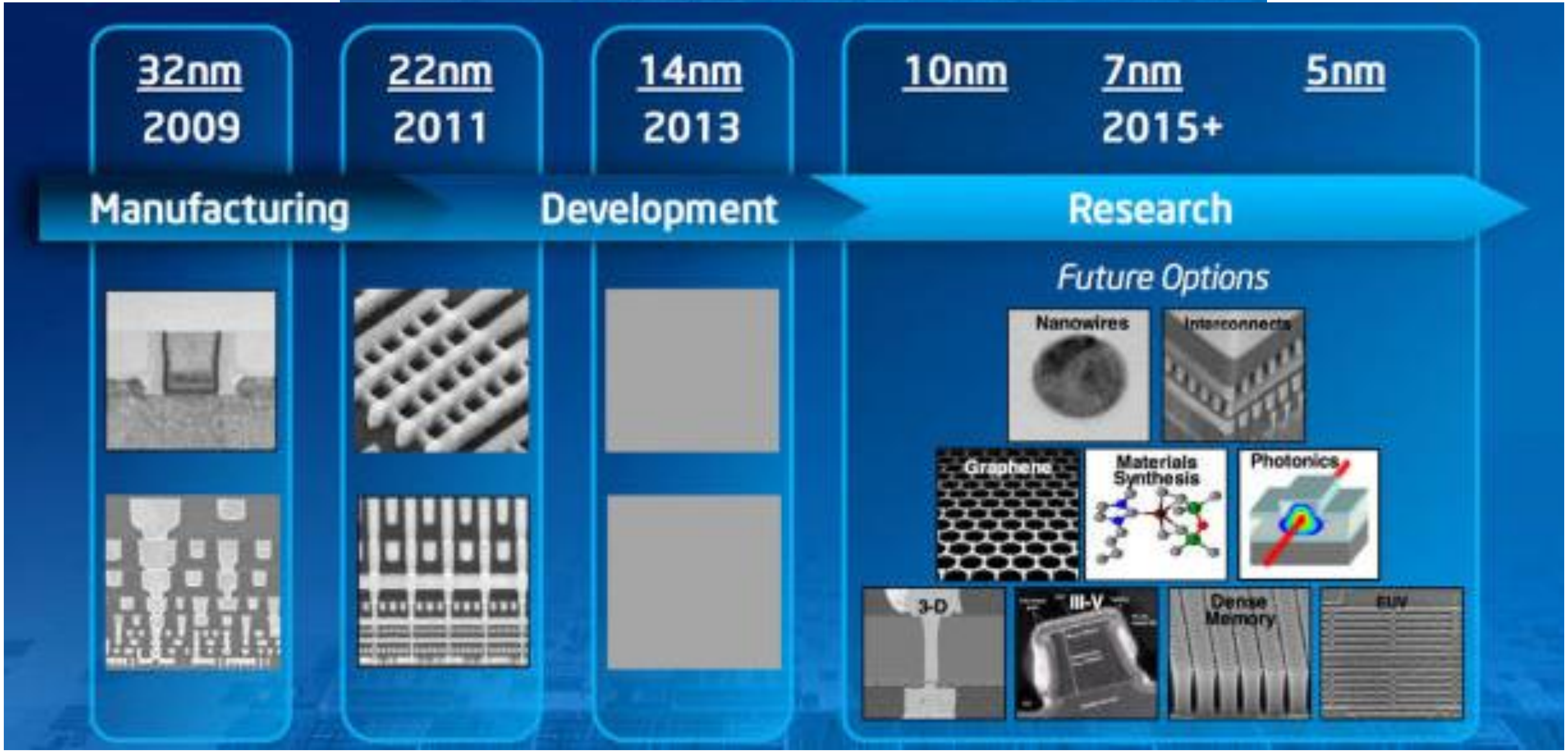
Moore's Law is Alive and Well



Intel
R&D PIPELINE

2011	2013	2015+
22 nm	14 nm	10 nm 7 nm 5 nm
IN PRODUCTION	*IN DEVELOPMENT*	*IN RESEARCH*

Lithography • Materials • Interconnect
... and more



32nm 2009	22nm 2011	14nm 2013	10nm 7nm 5nm 2015+
Manufacturing	Development		Research

Future Options

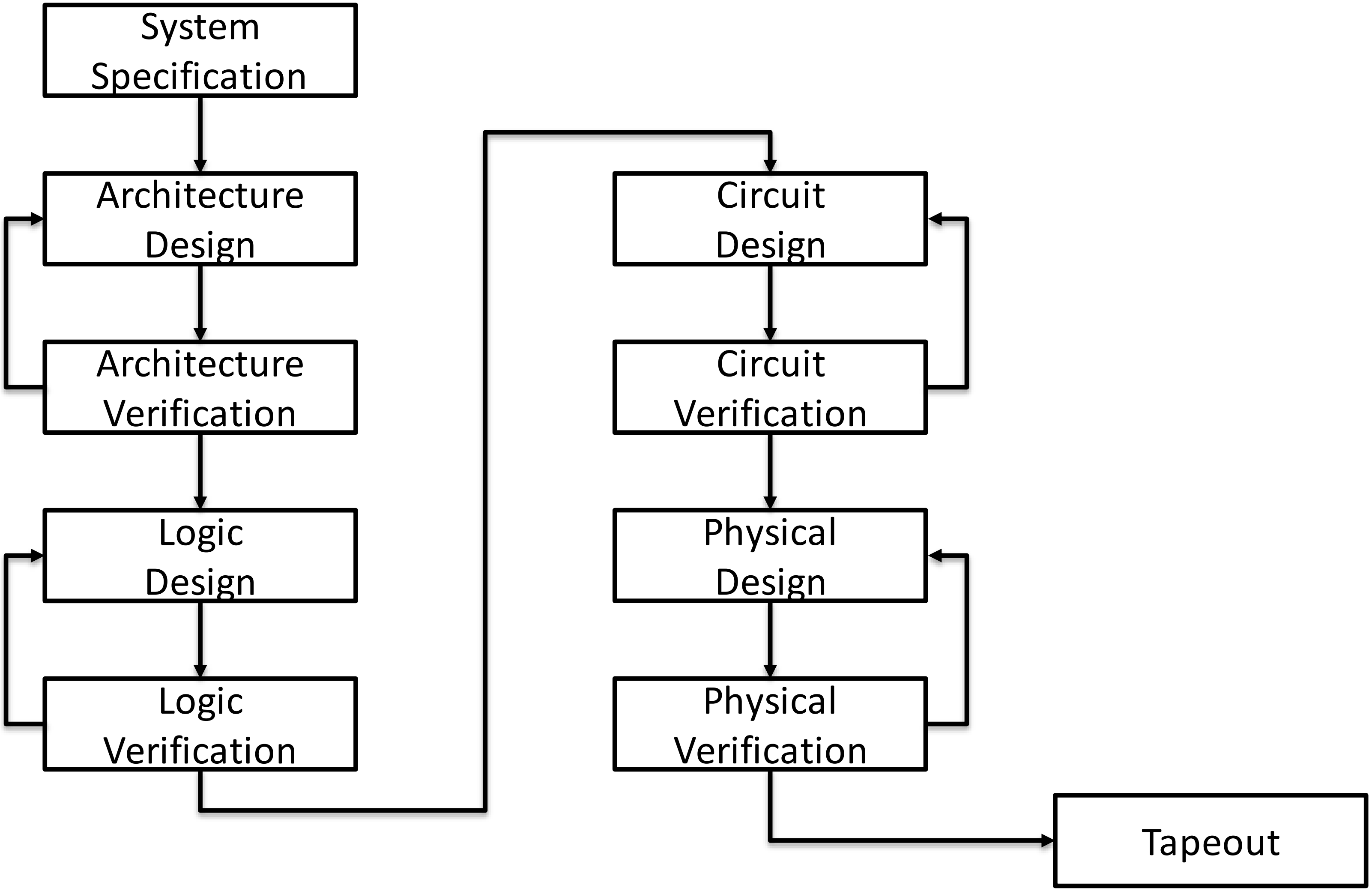
Very Large Scale Integration:

How to design a complex system?

⇩

Need for a well-defined *design methodology* to handle the complexity.

Overall Design Flow



System Specification

Architecture Design

Architecture Verification

Logic Design

Logic Verification

Circuit Design

Circuit Verification

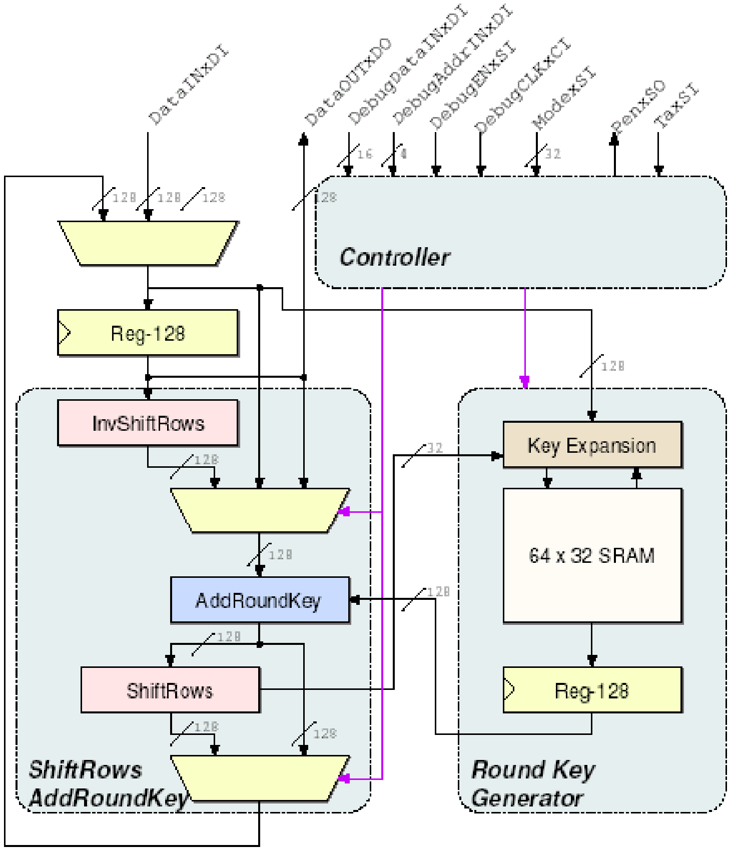
Physical Design

Physical Verification

Tapeout

Architecture Design

The golden rule: **ALWAYS START WITH A BLOCK DIAGRAM!**



Use hierarchy also in your block diagram

- **Identify blocks**
  What do we need to perform the functionality
- **Visualize structure**
  How are blocks connected
- **Find critical paths**
  Which block is the most critical (speed, area, power)?
- **Divide and Conquer**
  Use hierarchy, i.e., draw sub-block diagrams

Architecture Design: Structured Design Principle

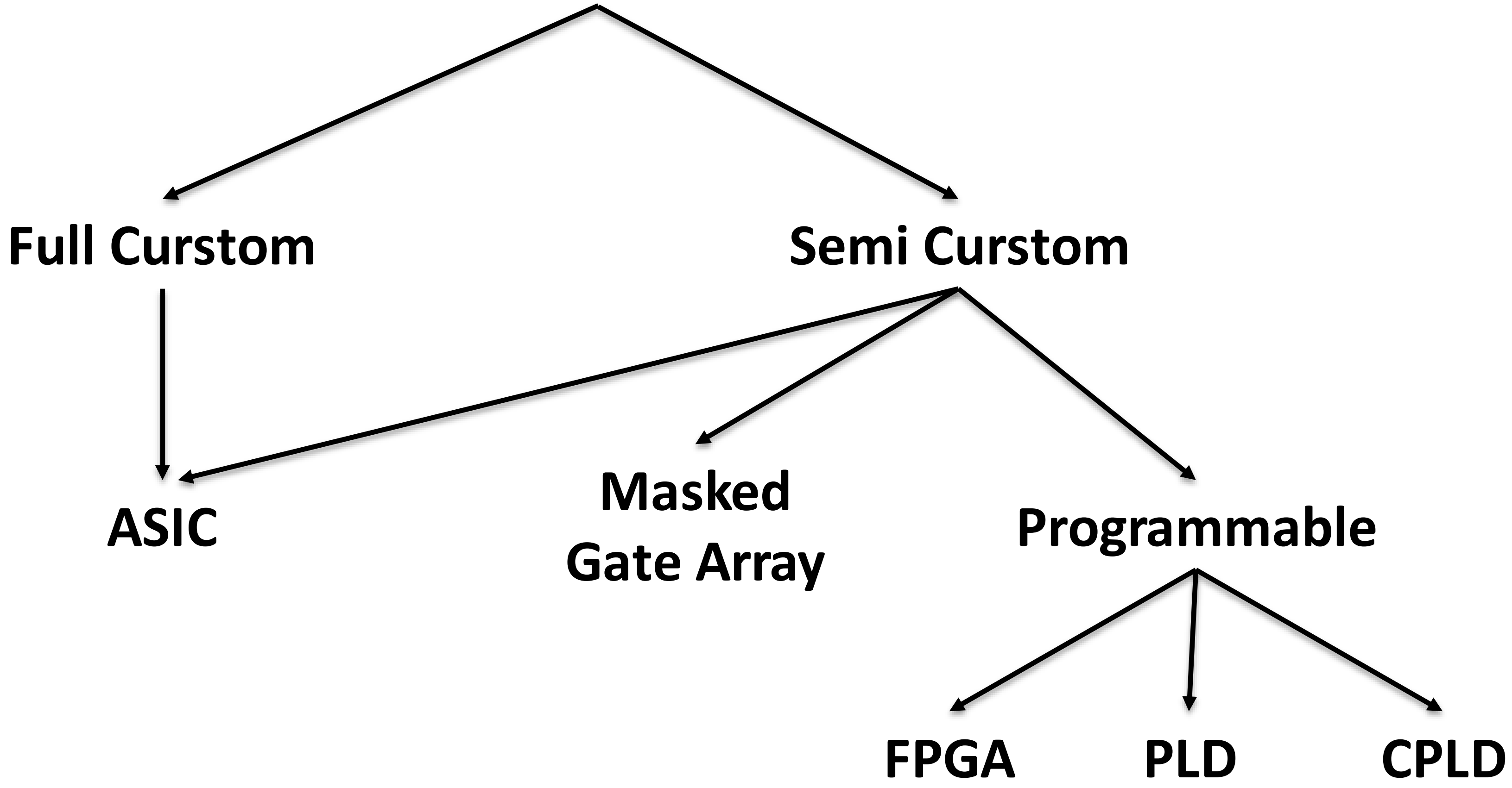
Hierarchy: “Divide and conquer” technique involves dividing a module into submodules and then repeating this operation on the sub-modules until the complexity of the smaller parts becomes manageable.

Regularity: The hierarchical decomposition of a large system should result in not only **simple**, but also **similar** blocks, as much as possible. Regularity usually reduces the number of different modules that need to be designed and verified, at all levels of abstraction.

Modularity: The various functional blocks which make up the larger system must have **well-defined functions** and **interfaces**.

Locality: Internal details remain at the local level. The concept of locality also ensures that connections are mostly between neighboring modules, **avoiding longdistance connections** as much as possible.

VLSI Design Styles for Logic/Circuit/Physical Design



Full Curstom

Semi Curstom

ASIC

Masked Gate Array

Programmable

FPGA

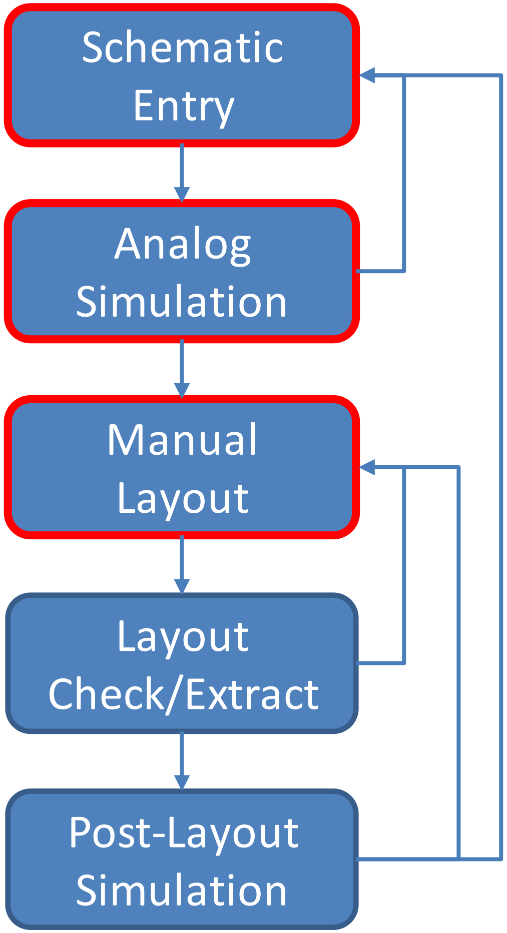
PLD

CPLD

VLSI Design Styles – Full Custom

Advantages

- Everything is done on device/transistor level
  - Full control over all device parameters
  - Full flexibility w.r.t. circuit topology
- Excellent performance
- Very high accuracy in simulations
- No strict separation between analog and digital parts
- Still the only option for analog design

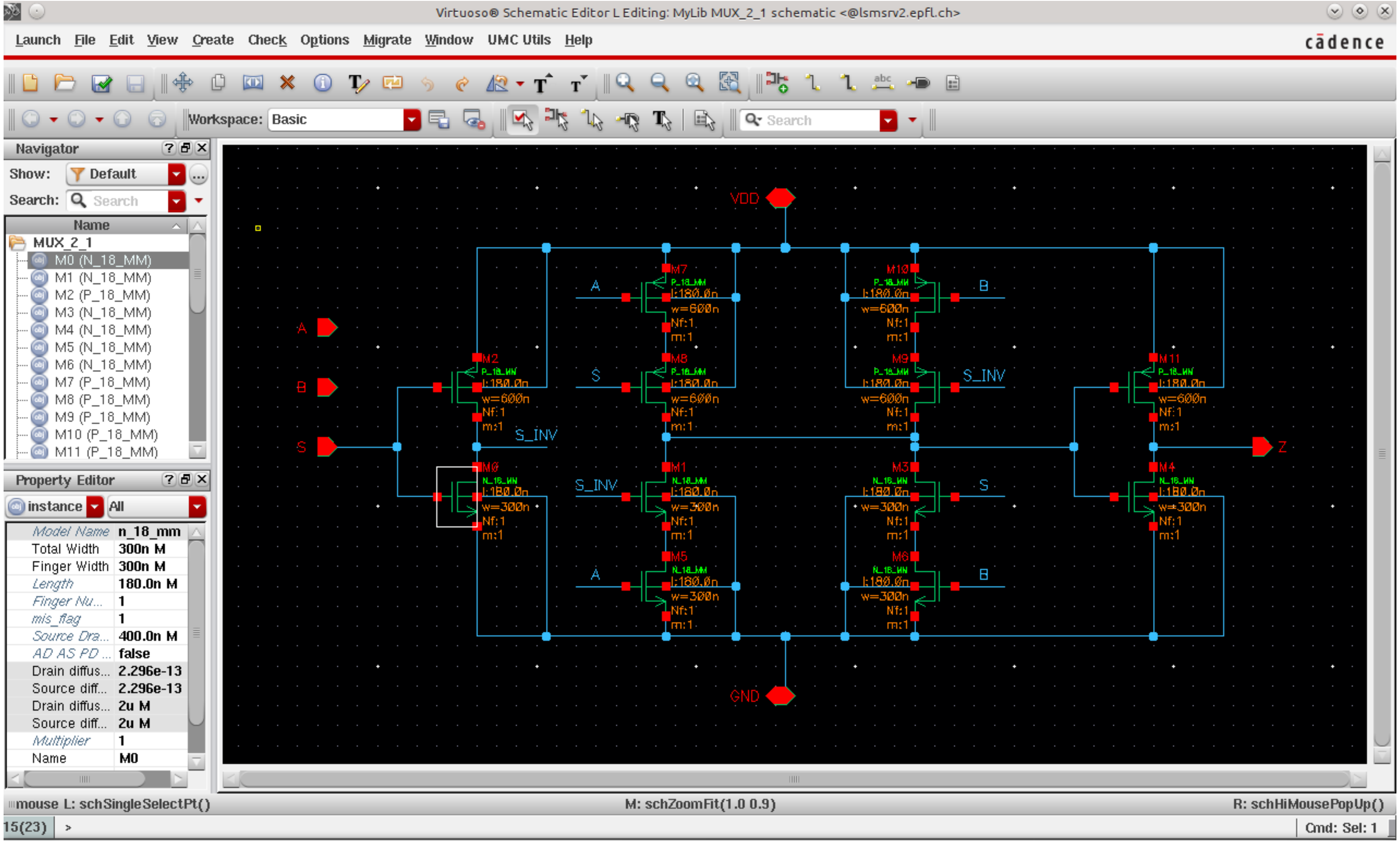


Disadvantages

- Everything is done manually
  - Limited design capacity (size) to few hundred devices
  - Long design time and limited re-use
- No abstraction/approximation
  - Slow

VLSI Design Styles – Full Custom

Schematic Entry



Virtuoso® Schematic Editor L Editing: MyLib MUX_2_1 schematic <@lsmsrv2.epfl.ch>

MUX_2_1
- M0 (N_18_MM)
- M1 (N_18_MM)
- M2 (P_18_MM)
- M3 (N_18_MM)
- M4 (N_18_MM)
- M5 (N_18_MM)
- M6 (N_18_MM)
- M7 (P_18_MM)
- M8 (P_18_MM)
- M9 (P_18_MM)
- M10 (P_18_MM)
- M11 (P_18_MM)

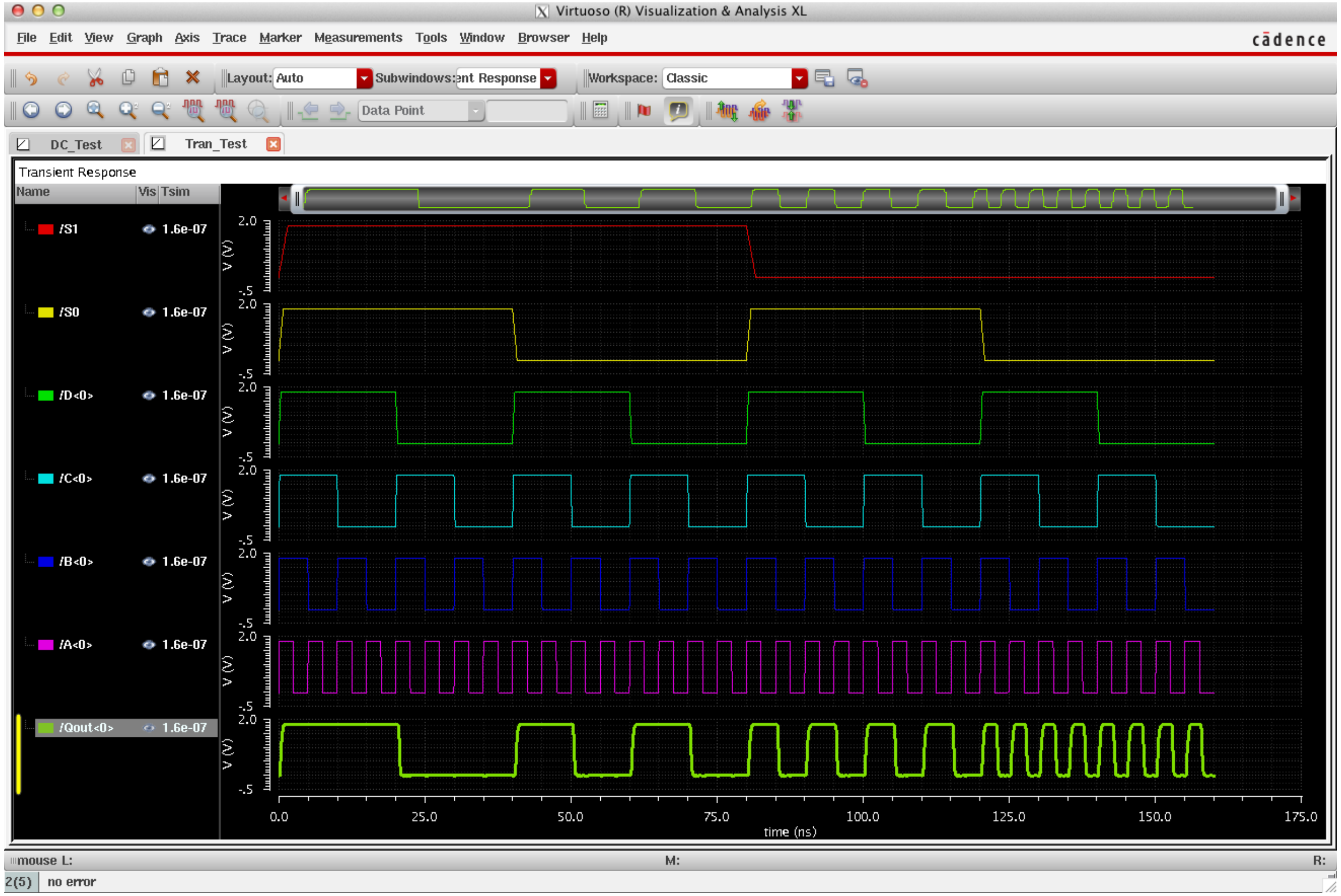
Model Name	n_18_mm
Total Width	300n M
Finger Width	300n M
Length	180.0n M
Finger Nu...	1
mis_flag	1
Source Dra...	400.0n M
AD AS PD ...	false
Drain diffus...	2.296e-13
Source diff...	2.296e-13
Drain diffus...	2u M
Source diff...	2u M
Multiplier	1
Name	M0

mouse L: schSingleSelectPt() M: schZoomFit(1.0 0.9) R: schHiMousePopUp()

15(23) > Cmd: Sel: 1

VLSI Design Styles – Full Custom

Simulation using Spice or Spectre



Layout: Auto

Subwindows: ent Response

Workspace: Classic

mouse L:

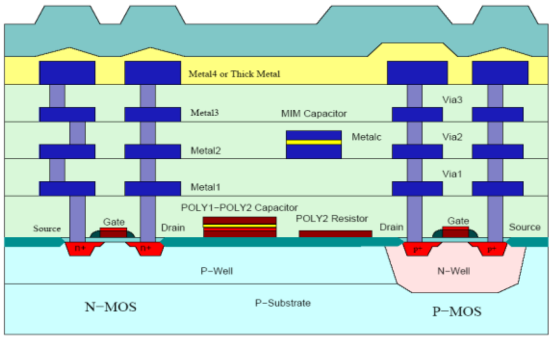
M:

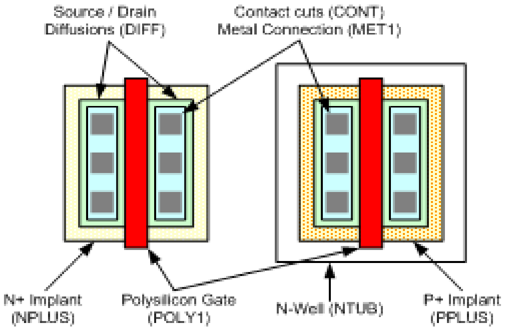
R:

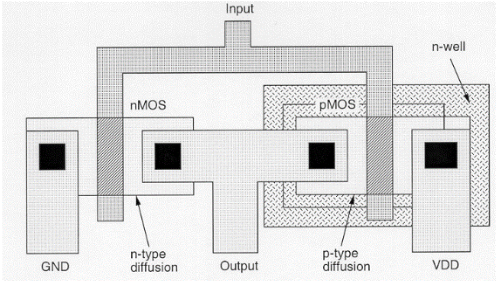
2(5) no error

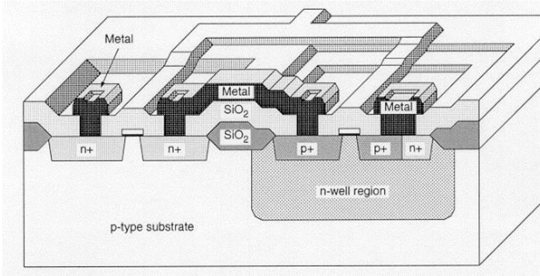
VLSI Design Styles – Full Custom

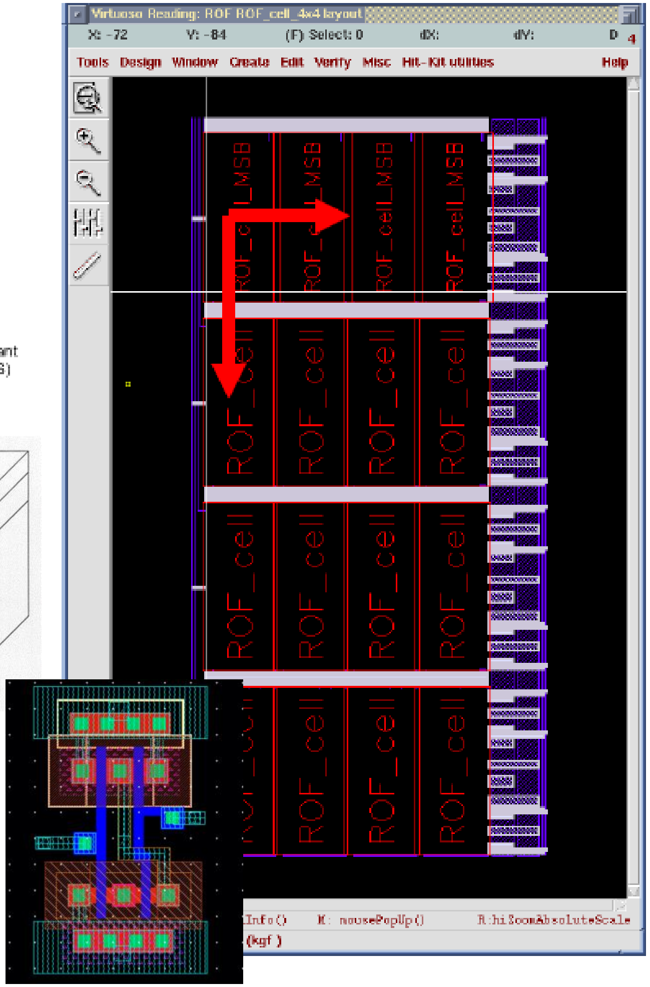
Layout: layers represent the masks for production




Source / Drain Diffusions (DIFF)
Contact cuts (CONT) Metal Connection (MET1)
N+ Implant (NPLUS)
Polysilicon Gate (POLY1)
N-Well (NTUB)
P+ Implant (PPLUS)







VLSI Design Styles – Full Custom

In full-custom style, the designer has many degrees of freedom to optimize a circuit design:

- Adjust individual transistor dimensions (width, length, aspect ratio, etc.) to satisfy:
  - DC specifications (voltage levels, switching thresholds)
  - Transient specifications (delay times, rise- and fall-times)

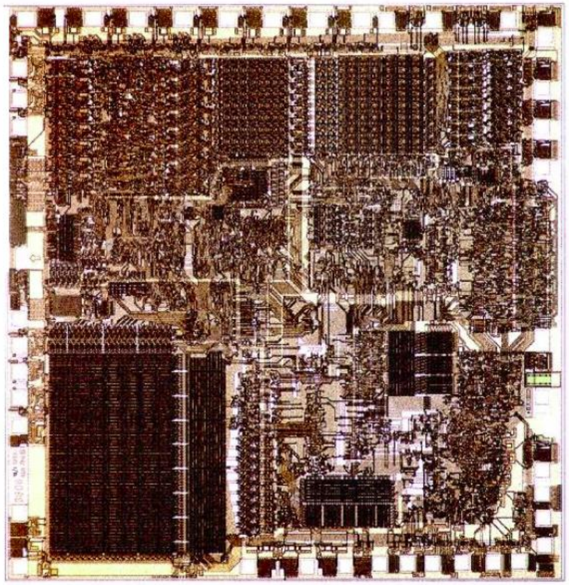
- Freely choose the most appropriate topology (placement and routing) for each circuit block.

- Decide on interconnection strategy between blocks.

- Decide for the global distribution of power, ground and clock.

Limits of Full Custom Design

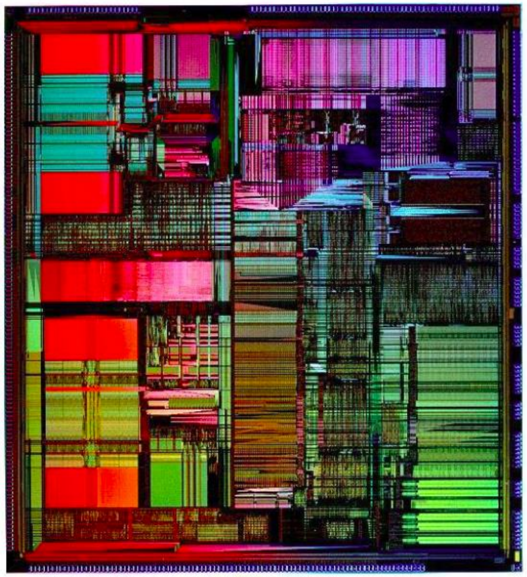
Increasing integration density no longer allows for design on transistor level, neither on schematic, nor on layout level



http://download.intel.com/museum/exhibits/hist_micro/hof/large_jpegs/8088B1.jpg

Intel 8088, 1979

Full-custom design



http://download.intel.com/museum/exhibits/hist_micro/hof/large_jpegs/pentB.jpg

Intel Pentium, 1993

Few macros, but mostly built using automatic tools

Need for a more automated that leaves the details to EDA tools

Principle Idea of Semicustom Design Flow

Render the design process more efficiently by using

- **Hierarchy**: build complex designs from a collection of smaller and much simpler components which by themselves are again hierarchical

- **Abstraction**: simplified *description*/characterization *of components as a model (black box)* to better use them on the next level of hierarchy

- **Design automation**: algorithms and tools to realize an abstract design description from components

Abstraction Levels: Technology

Layers of doped semiconductor material or metal interconnect

- Possible arrangemet of these layers is determined by manufacturing process
- Details are one of the best guarded secrets of a foundry (years to develop)

Composition of layers determines electrical and other characteristics

- Characterized by the complex laws of physics
- Technology Computer Aided Design (TCAD) tools used for analysis and simulation

TCAD

Technology

Abstraction Levels: Devices

Basic building blocks for circuit design that abstract the physics of the technology layer to electrical characteristics

Foundries supply a *Process Design Kit* (PDK) which provides

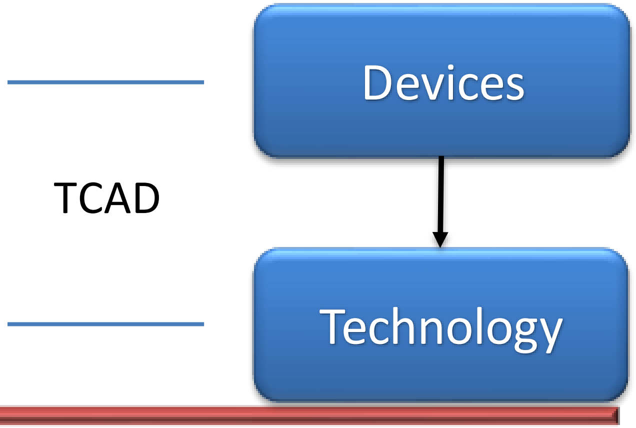
- Devices (Transistors, Resistors, Capacitors, Diodes)
- Layers (e.g., for interconnect)

Various "Flavors" of PDKs are available, e.g.:

- General Purpose/High Speed/Low Power
- RF/Image Sensor
- Flash/DRAM

Devices are abstracted to *compact electrical models* for circuit simulations and *design rules*

- Generated through TCAD and/or measurements



Abstraction Levels: Macros and Standard Cells

Basic analog or digital circuits as hierarchical building blocks for more complex circuits

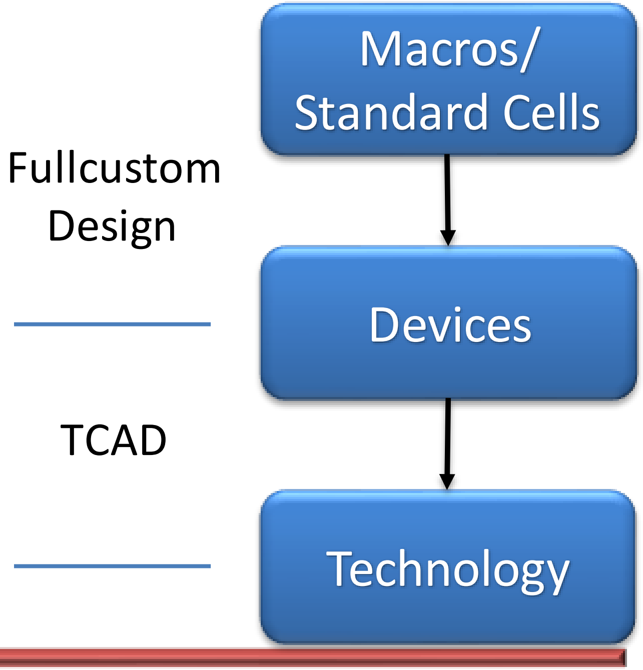
- Built from devices of the PDK with full control over all device parameters
- Characterized and optimized through circuit simulation, e.g., using SPICE
- Manual, carefully optimized compact layout

Library and IP providers offer for example:

- Standard cells: libraries of basic digital gates
- Digital macros such as RAMs/ROMs
- Analog macros such as ADCs/DACs/…

Standard cells/macros are described by abstract models that capture

- Functionality.
- Interface (logical and electrical).
- Performance (e.g., delay and power)
- Physical appearance (e.g., size and shape)



Abstraction Levels: Register Transfer Level (RTL) Design

Complex digital blocks (IPs) and complete complex digital ICs

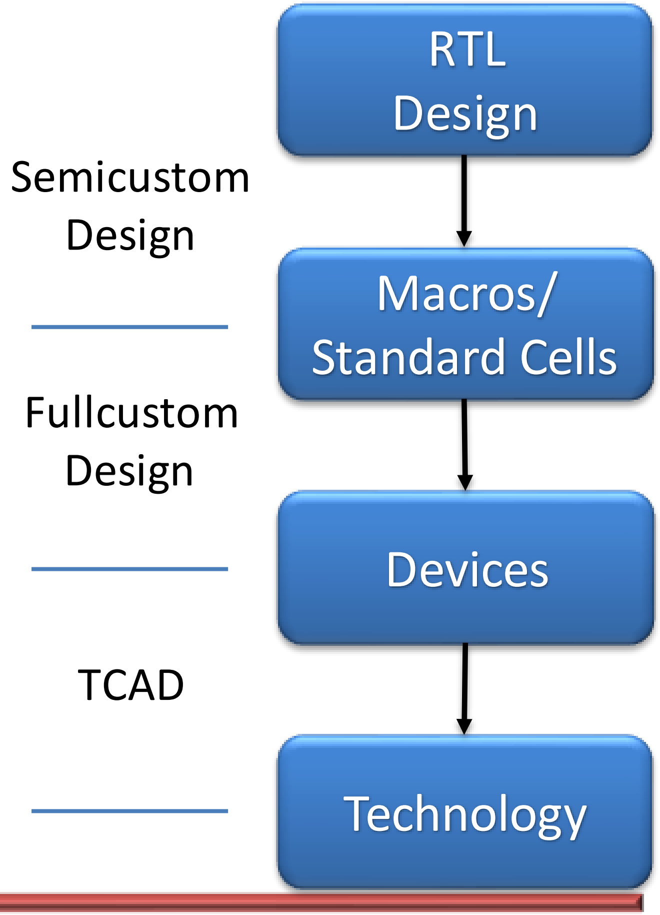
Specification of a digital circuit in a Hardware Description Language (HDL)

- Defines computational logic and storage elements in a abstract way

Semicustom design:

- Frontend: Automatic translation of HDL into a Gate-Level netlist (a circuit built from standard cells and macro blocks)
- Backend: Physical implementation of the design based on basic building blocks

RTL designs can be abstracted through behavioral models



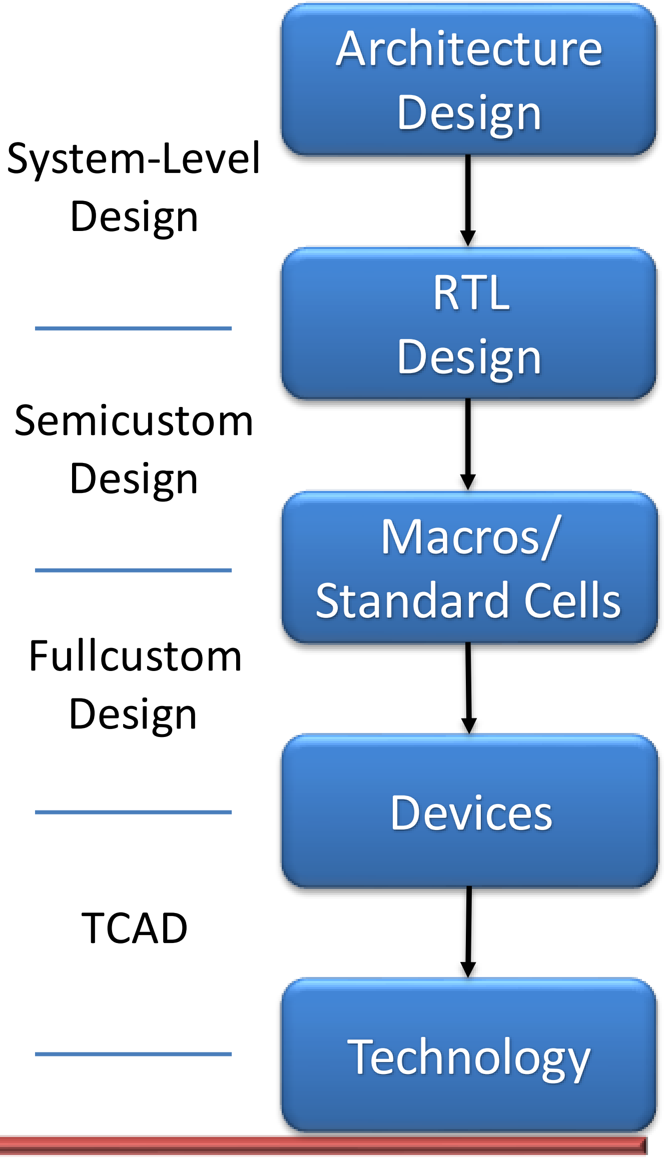
Abstraction Levels: Architecture

Entire system (e.g., cell phone)

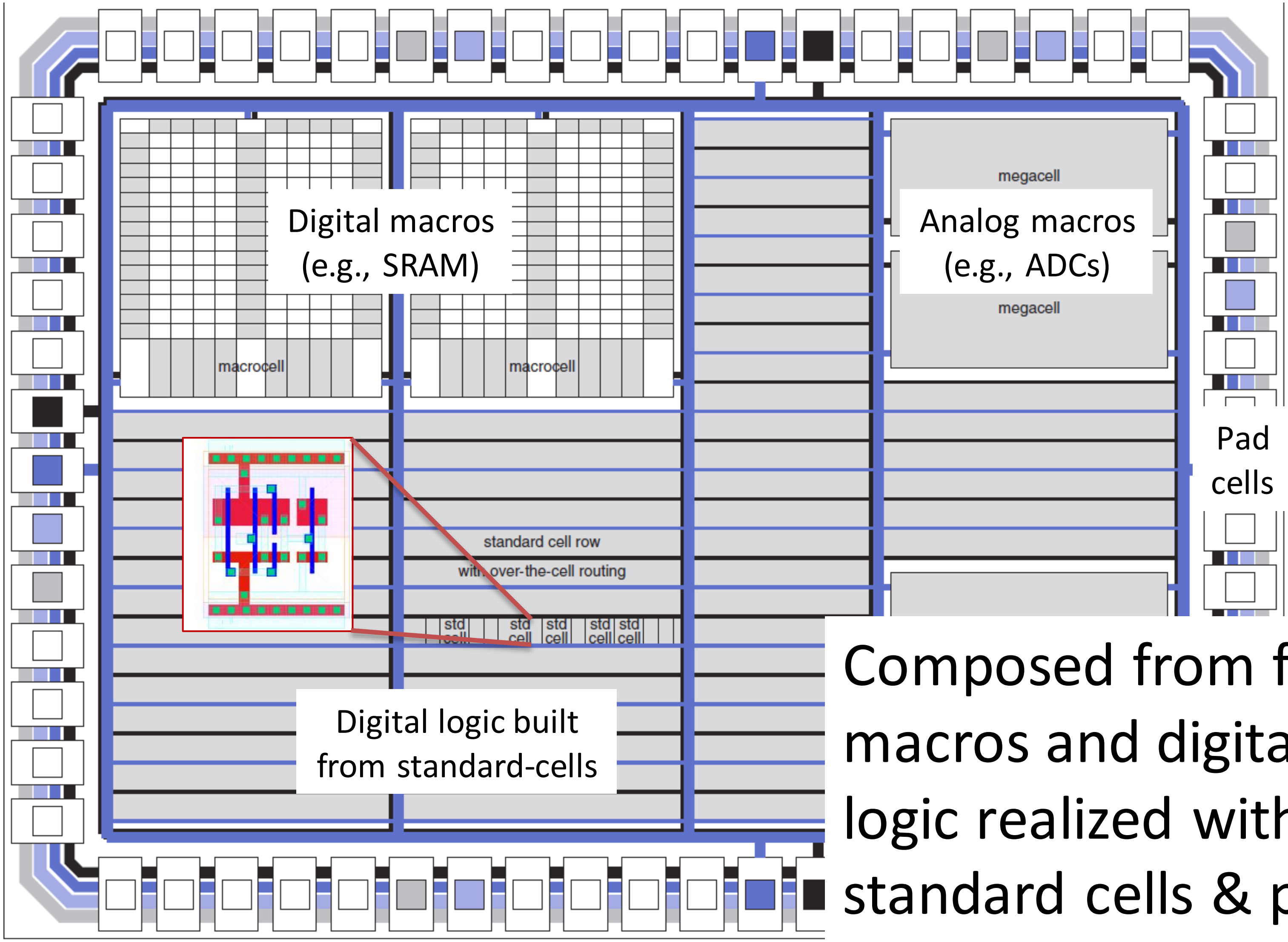
- Complex system comprised of individual (partially highly complex) components
- Communicating between components often through complex interfaces and protocols

Described either

- still in RTL by instantiating many other RTL components or
- in a high-level or HDL language instantiating behavioral models of the RTL components



A Digital IC with a Semicustom Design Flow



Digital macros (e.g., SRAM)

Analog macros (e.g., ADCs)

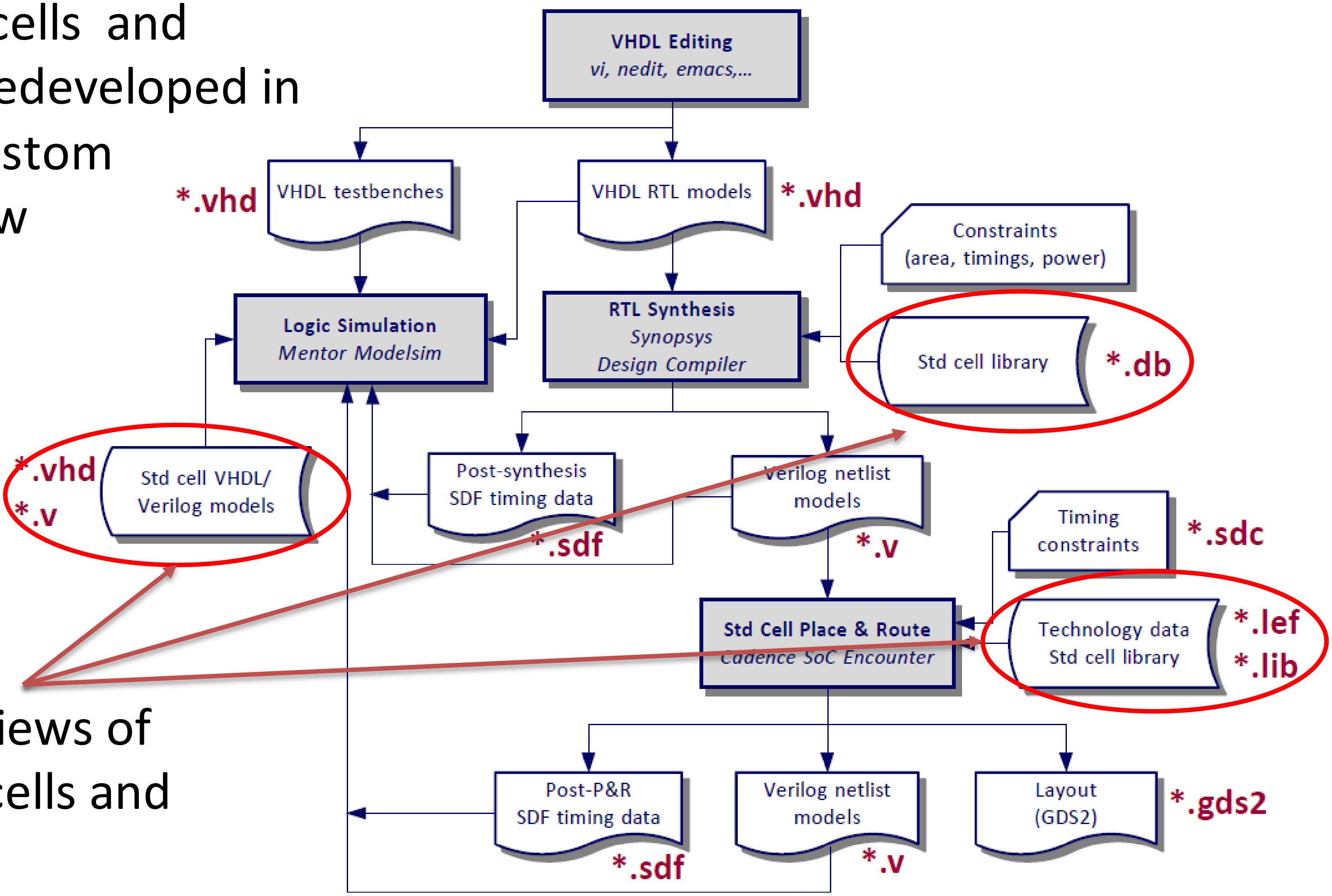
Pad cells

Digital logic built from standard-cells

Composed from few macros and digital logic realized with standard cells & pads

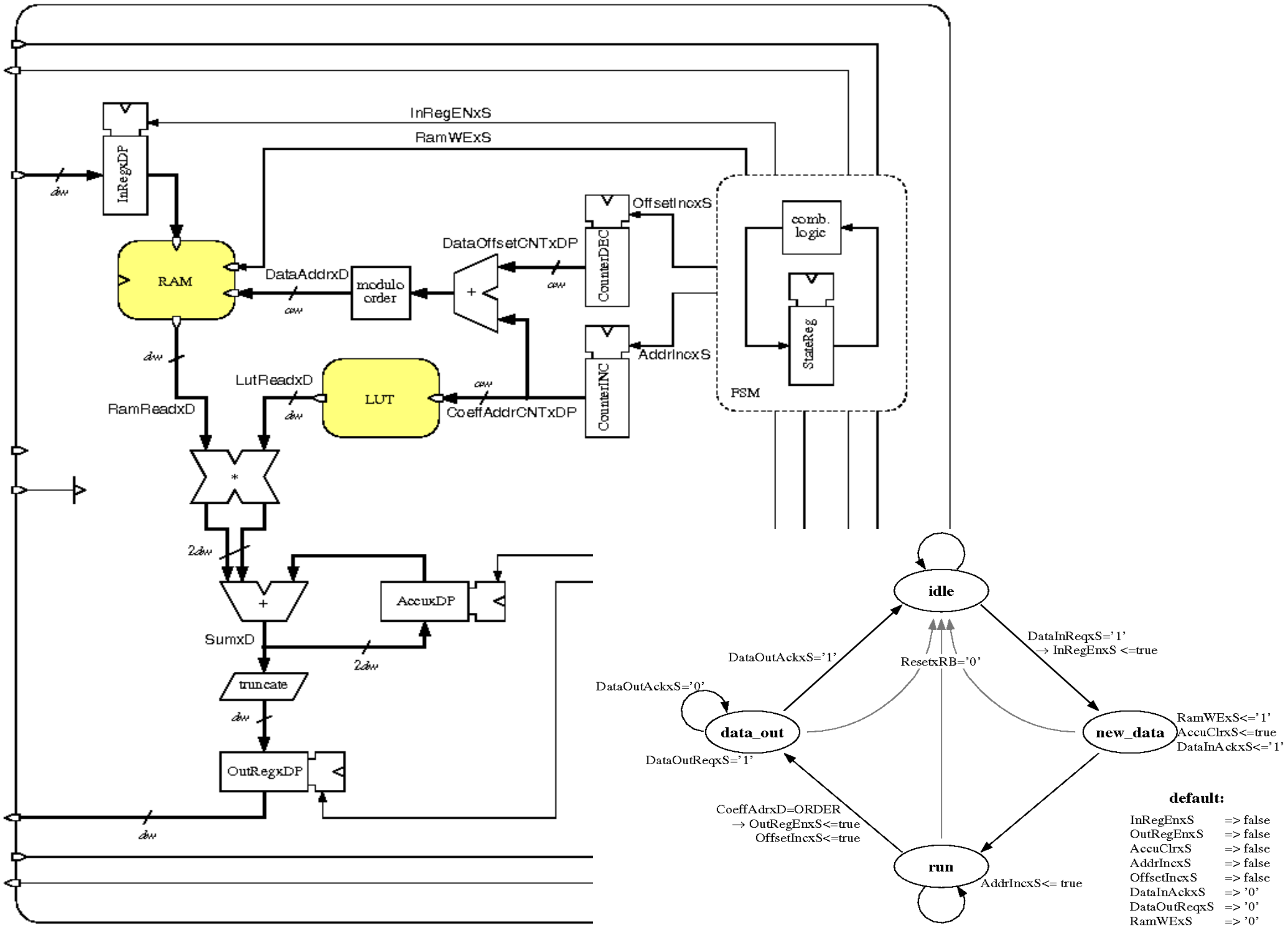
Semicustom Design Flow

Standard cells and macros aredeveloped in the full-custom design flow



Abstract views of standard cells and macros

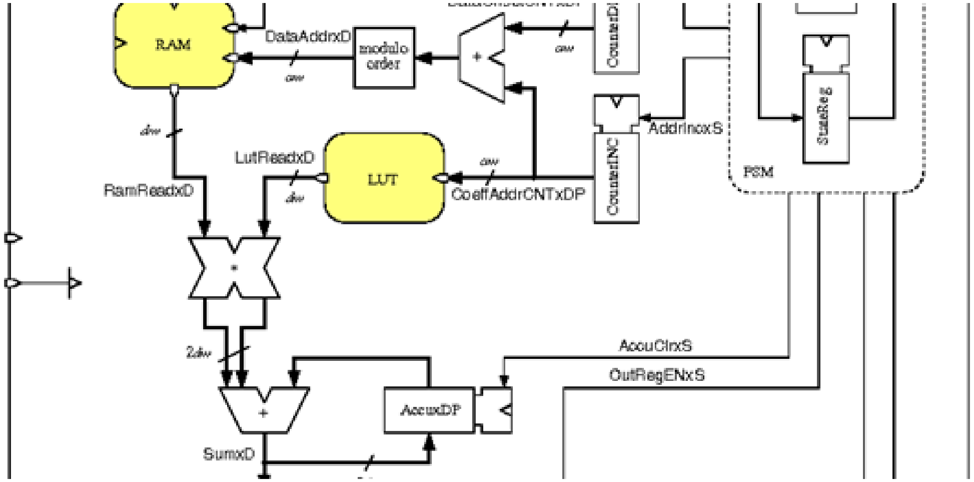
Semicustom Design Flow: Architecture Design

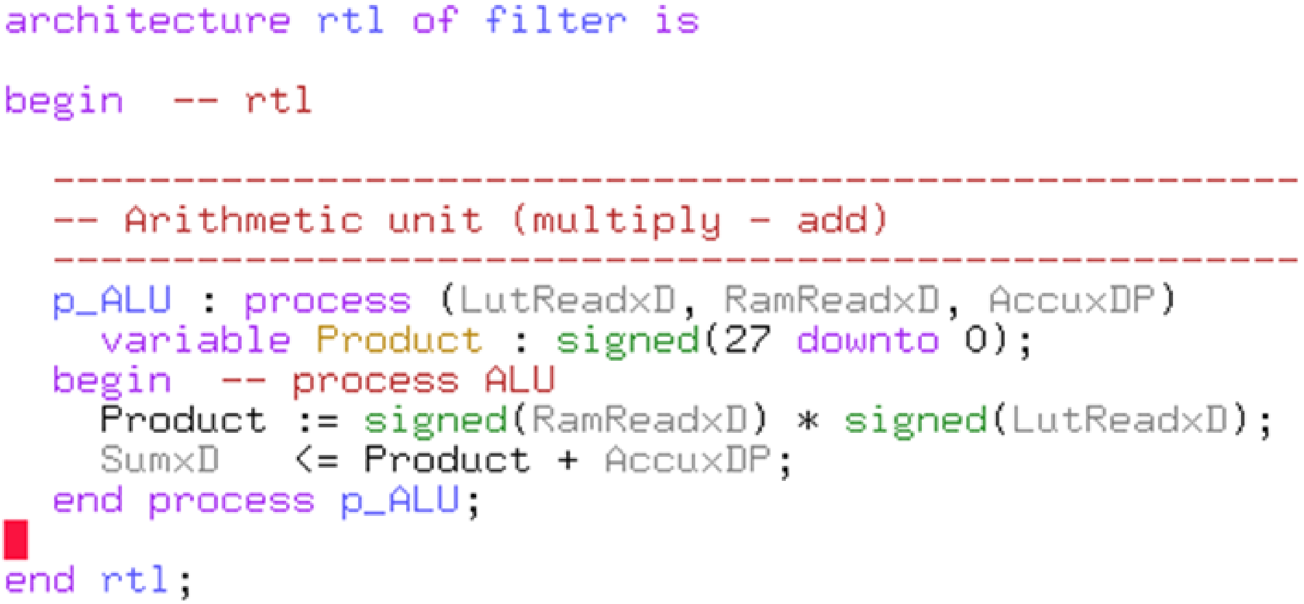


default:

InRegEnxS	=> false
OutRegEnxS	=> false
AccuClrxS	=> false
AddrIncxS	=> false
OffsetIncxS	=> false
DataInAckxS	=> '0'
DataOutReqxS	=> '0'
RamWExS	=> '0'

Semicustom Design Flow: Design Entry in VHDL





```
architecture rtl of filter is

begin  -- rtl

  -------------------------------------------------------
  -- Arithmetic unit (multiply - add)
  -------------------------------------------------------
  p_ALU : process (LutReadxD, RamReadxD, AccuxDP)
    variable Product : signed(27 downto 0);
  begin  -- process ALU
    Product := signed(RamReadxD) * signed(LutReadxD);
    SumxD   <= Product + AccuxDP;
  end process p_ALU;

end rtl;
```

Semicustom Design Flow: Design Entry in VHDL

Modern hardware description languages allow structural descriptions that basically determine all interconnections between well-defined blocks:

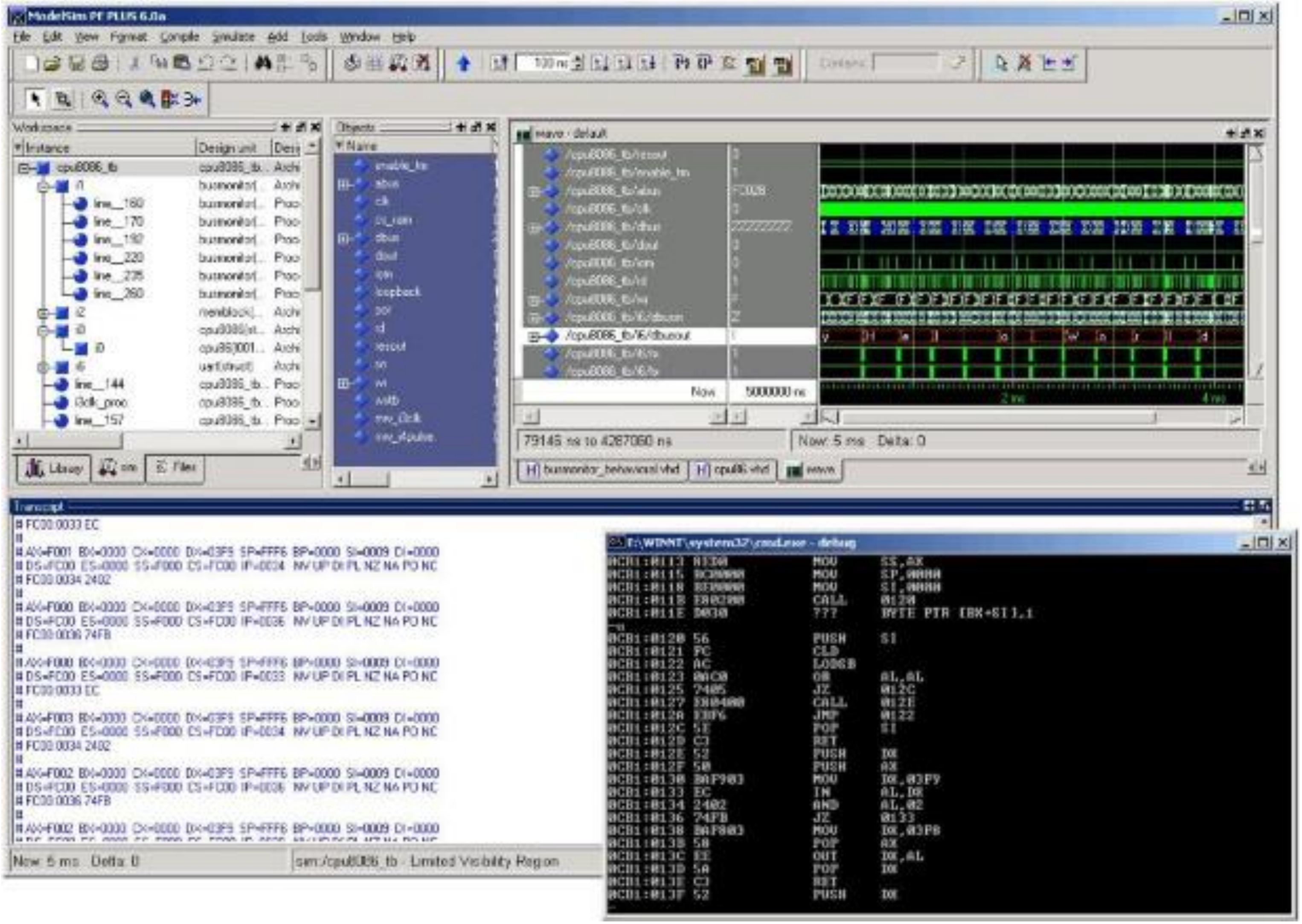
```
adder: component adder port map (a,b,ci,co,s);
```

as well as purely behavioral descriptions like:

```
sum <= a + b - c;
```

The synthesizer is responsible for converting the behavioral description into an optimized design.

Semicustom Design Flow: Verification



79145 ns to 4287060 ns

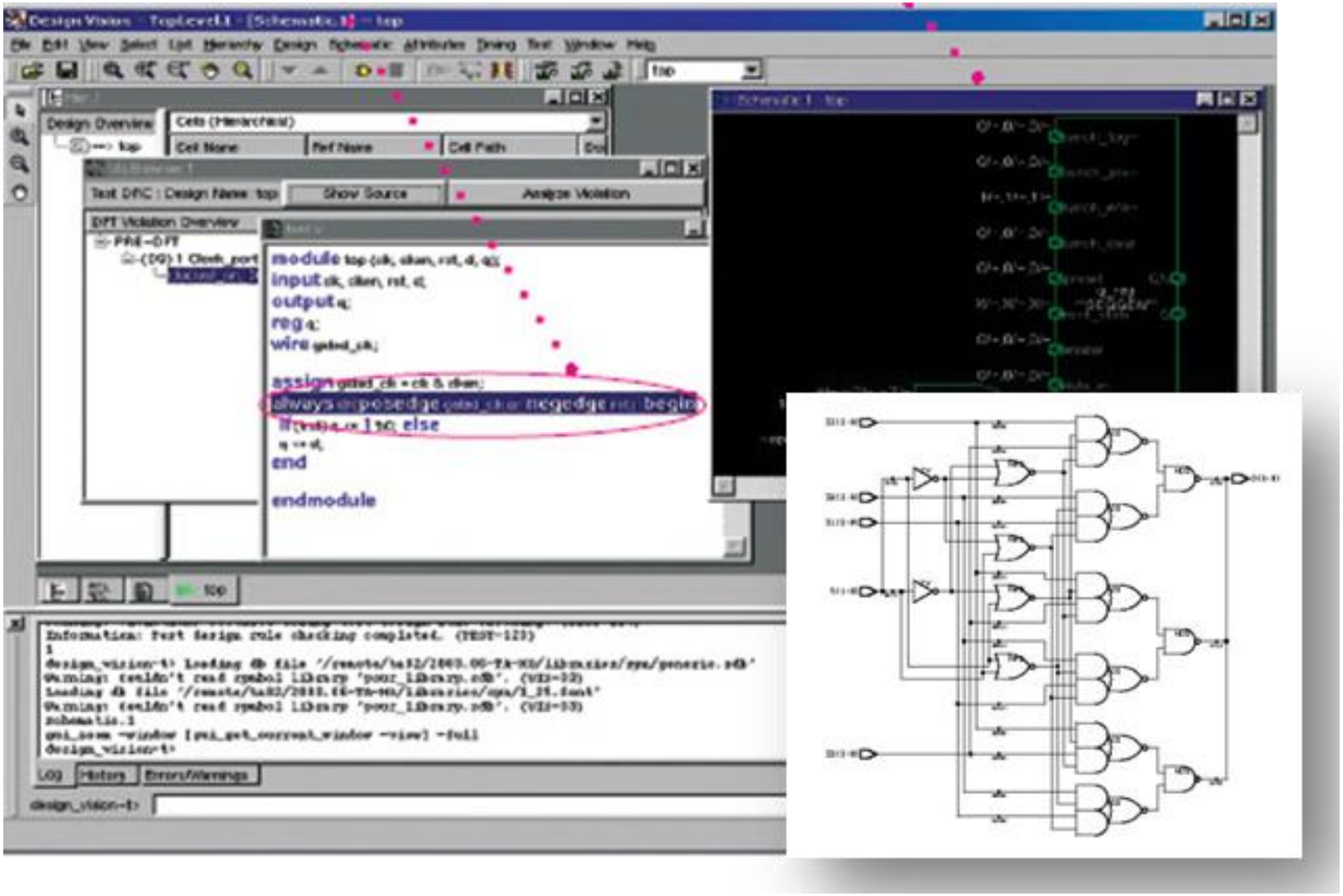
Now: 5 ms Delta: 0

```
# FC00:0033 EC
#
# AX=F001 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
# DS=FC00 ES=0000 SS=F000 CS=FC00 IP=0034  NV UP DI PL NZ NA PO NC
# FC00:0034 2402
#
# AX=F000 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
# DS=FC00 ES=0000 SS=F000 CS=FC00 IP=0036  NV UP DI PL NZ NA PO NC
# FC00:0036 74FB
#
# AX=F000 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
# DS=FC00 ES=0000 SS=F000 CS=FC00 IP=0033  NV UP DI PL NZ NA PO NC
# FC00:0033 EC
#
# AX=F003 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
# DS=FC00 ES=0000 SS=F000 CS=FC00 IP=0034  NV UP DI PL NZ NA PO NC
# FC00:0034 2402
#
# AX=F002 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
# DS=FC00 ES=0000 SS=F000 CS=FC00 IP=0036  NV UP DI PL NZ NA PO NC
# FC00:0036 74FB
#
# AX=F002 BX=0000 CX=0000 DX=03F9 SP=FFF6 BP=0000 SI=0009 DI=0000
```

Now: 5 ms Delta: 0   sim:/cpu8086_tb - Limited Visibility Region

```
0CB1:0113 8ED0          MOV     SS,AX
0CB1:0115 BC0000        MOV     SP,0000
0CB1:0118 BE0000        MOV     SI,0000
0CB1:011B E80200        CALL    0120
0CB1:011E D030          ???     BYTE PTR [BX+SI],1
-u
0CB1:0120 56            PUSH    SI
0CB1:0121 FC            CLD
0CB1:0122 AC            LODSB
0CB1:0123 0AC0          OR      AL,AL
0CB1:0125 7405          JZ      012C
0CB1:0127 E80400        CALL    012E
0CB1:012A EBF6          JMP     0122
0CB1:012C 5E            POP     SI
0CB1:012D C3            RET
0CB1:012E 52            PUSH    DX
0CB1:012F 50            PUSH    AX
0CB1:0130 BAF903        MOV     DX,03F9
0CB1:0133 EC            IN      AL,DX
0CB1:0134 2402          AND     AL,02
0CB1:0136 74FB          JZ      0133
0CB1:0138 BAF803        MOV     DX,03F8
0CB1:013B 58            POP     AX
0CB1:013C EE            OUT     DX,AL
0CB1:013D 5A            POP     DX
0CB1:013E C3            RET
0CB1:013F 52            PUSH    DX
-
```

Semicustom Design Flow: Logic Synthesis



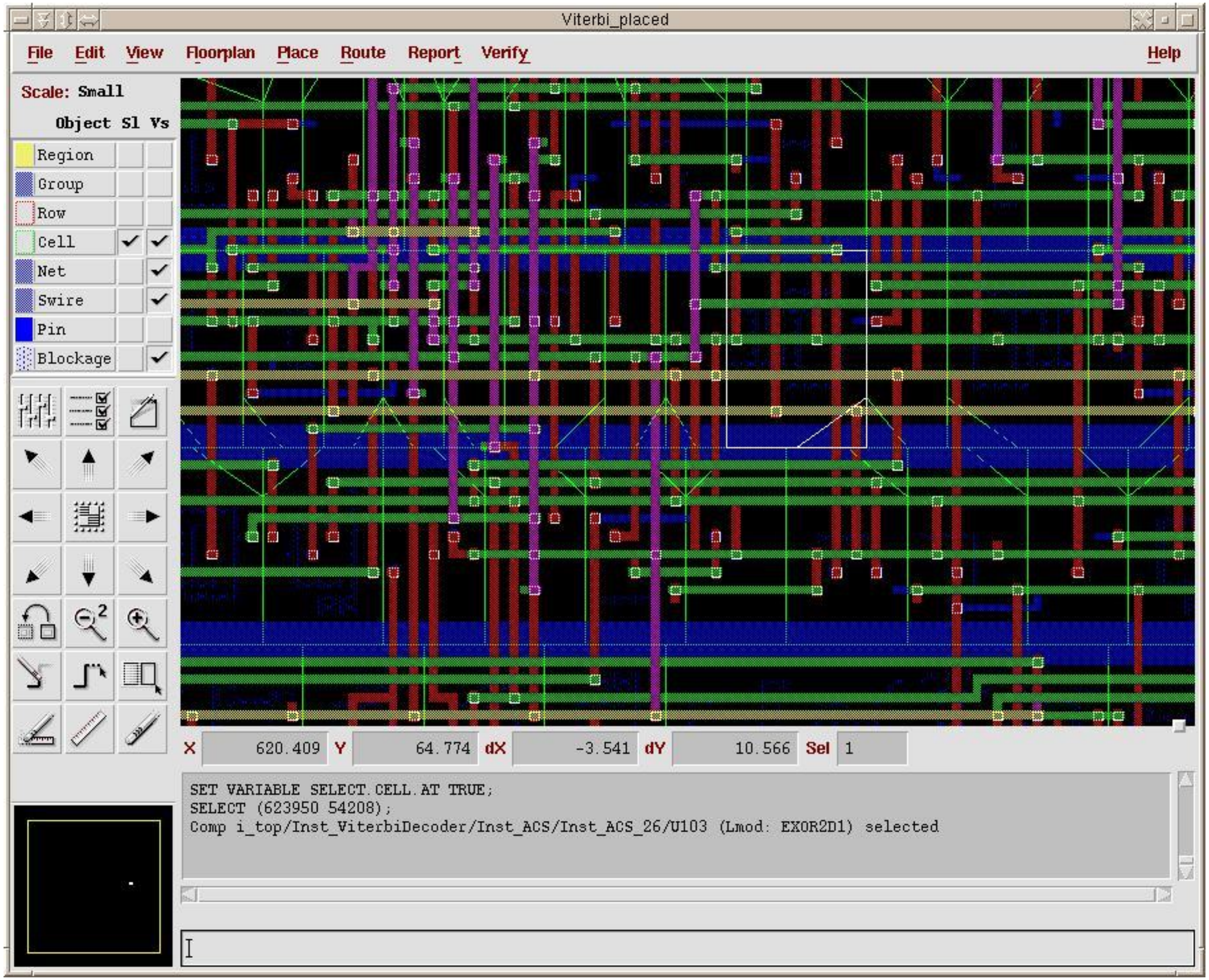
```
module top (clk, clken, rst, d, q);
input clk, clken, rst, d;
output q;
reg q;
wire gated_clk;

assign gated_clk = clk & clken;
always @(posedge gated_clk or negedge rst) begin
 if (!rst) q <= 1'b0; else
 q <= d;
end

endmodule
```

```
Information: Test design rule checking completed. (TEST-123)
1
design_vision-t> Loading db file '/remote/ta12/2003.06-TA-XX/libraries/syn/generic.sdb'
Warning: Couldn't read symbol library 'your_library.sdb'. (UID-33)
Loading db file '/remote/ta12/2003.06-TA-XX/libraries/syn/1_25.font'
Warning: Couldn't read symbol library 'your_library.sdb'. (UID-33)
schematic.1
gui_zoom -window [gui_get_current_window -view] -full
design_vision-t>
```

Semicustom Design Flow: Automatic Place & Route



```
SET VARIABLE SELECT.CELL.AT TRUE;
SELECT (623950 54208);
Comp i_top/Inst_ViterbiDecoder/Inst_ACS/Inst_ACS_26/U103 (Lmod: EXOR2D1) selected
```

The Semiconductor Ecosystem

Fabless Companies

- QUALCOMM®
- SanDisk
- MARVELL®
- TEXAS INSTRUMENTS
- BROADCOM.
- AMD
- PMC-SIERRA
- Micron®
- freescale™ semiconductor

CAD/EDA

- synopsys®
- cādence™
- Mentor Graphics®

IP Vendors

- ARM®
- FARADAY

Fab/Foundry

- TOWERJAZZ
- tsmc
- UMC
- SMIC™
- GLOBALFOUNDRIES

VLSI Performance Metrics and Objectives

In VLSI Design, we always have a trade-off between:

Speed

Power

Reliability

Cost (Process/Area)

For every circuit/system, we will have to evaluate the specifications and choose the best trade-off that meet any potential hard constraints on some metrics

In this class, we discussed the basic metrics and relationship between Speed, Power, and Cost.