Electronics and Communication Systems
Electronics Systems

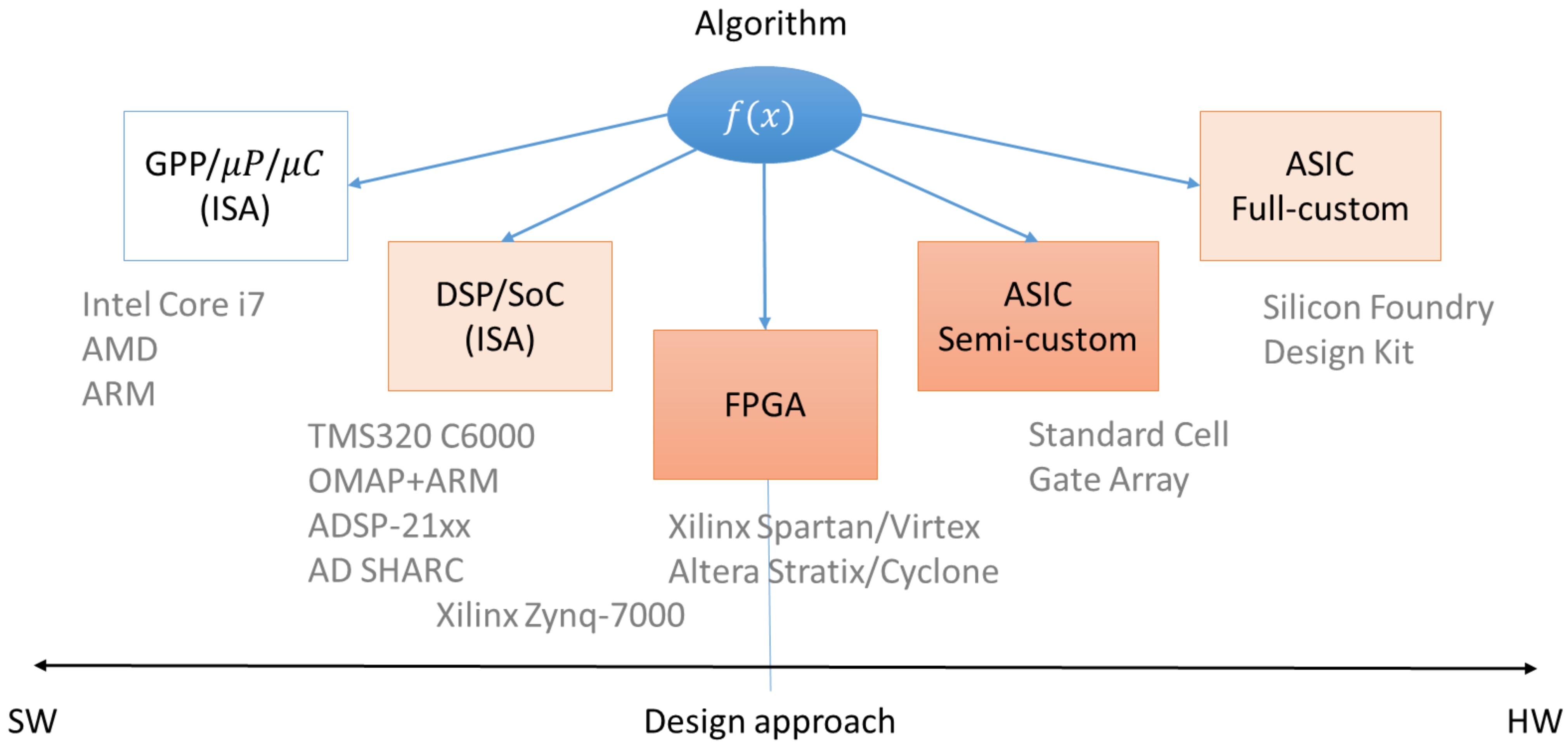
Master Degree in Computer Engineering
https://computer.ing.unipi.it/ce-lm

Luca Fanucci
Dipartimento Ingegneria dell'Informazione
Via Caruso 16 - I-56122 - Pisa - Italy
Phone: +39 050 2217 668
Fax: +39 050 2217 522
Mobile: +39 347 5013837
Email: luca.fanucci@unipi.it

Outline

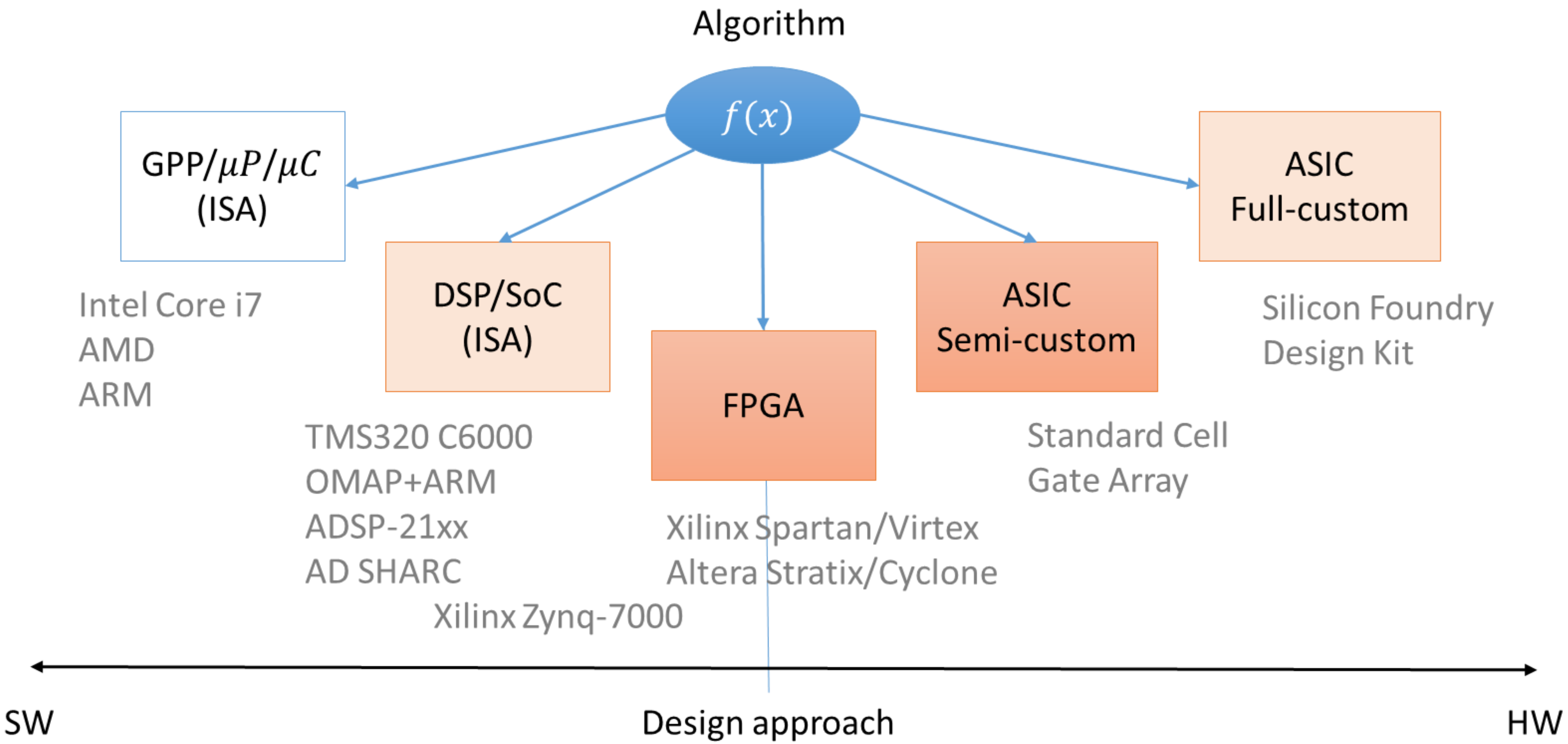
- Design Metrics
- Managing engineering trade-offs

How to map a system/algorithm to hw/sw design



Design is the art of managing engineering trade-offs !

How to map a system/algorithm to hw/sw design



Algorithm

$f(x)$

GPP/$\mu P/\mu C$ (ISA)

Intel Core i7
AMD
ARM

DSP/SoC (ISA)

TMS320 C6000
OMAP+ARM
ADSP-21xx
AD SHARC

Xilinx Zynq-7000

FPGA

Xilinx Spartan/Virtex
Altera Stratix/Cyclone

ASIC Semi-custom

Standard Cell
Gate Array

ASIC Full-custom

Silicon Foundry
Design Kit

SW

Design approach

HW

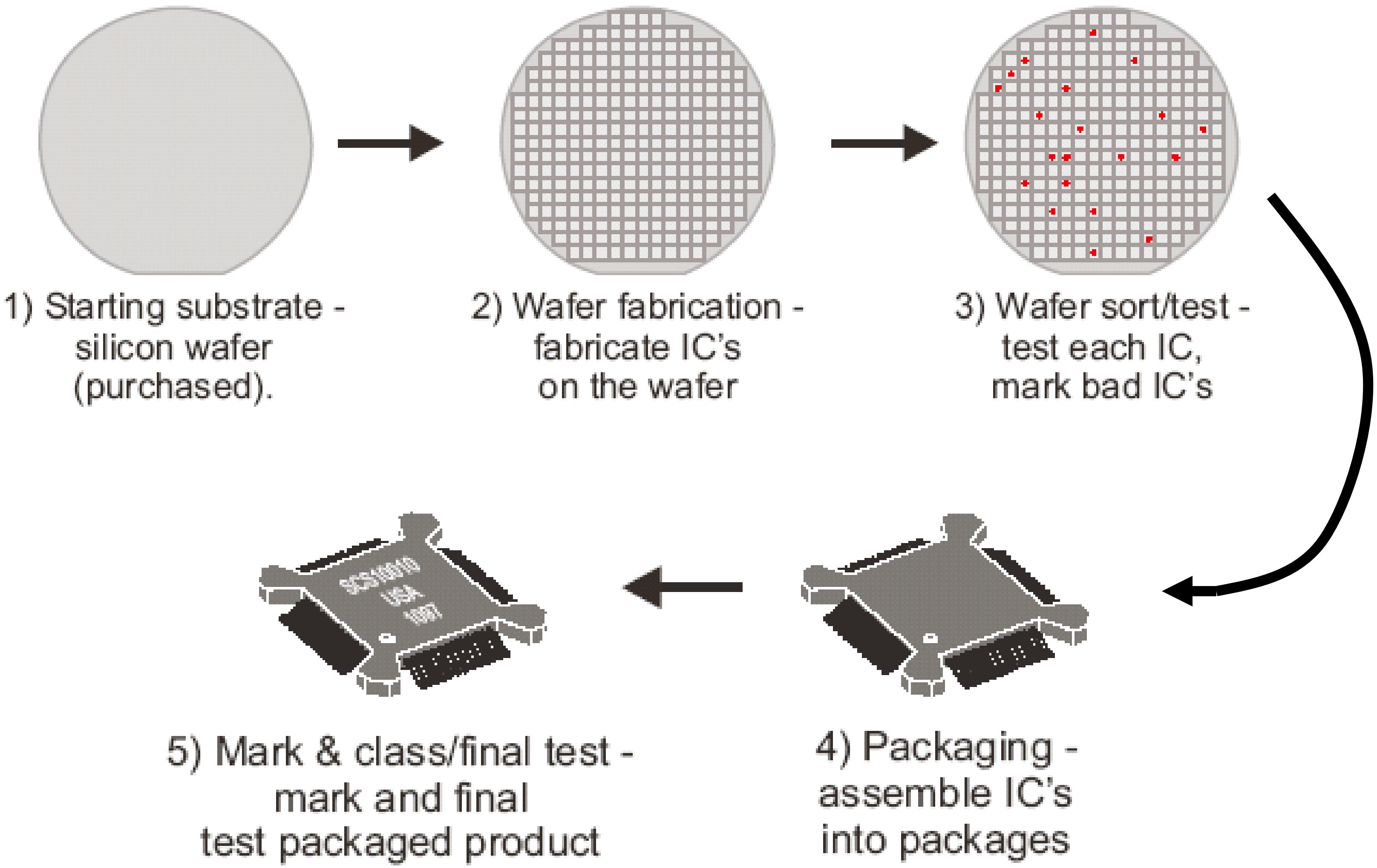
Let's see some fundamental design metrics !!

Fundamental Design Metrics

- Functionality
- Cost
  - NRE (fixed) costs - design effort
  - RE (variable) costs - cost of parts, assembly, test
- Reliability, robustness
  - Noise margins
  - Noise immunity
- Performance
  - Speed (delay)
  - Power consumption; energy
- Time-to-market

Design is the art of managing engineering trade-offs !

How a chip is manufactured?



1) Starting substrate - silicon wafer (purchased).

2) Wafer fabrication - fabricate IC's on the wafer

3) Wafer sort/test - test each IC, mark bad IC's

4) Packaging - assemble IC's into packages

5) Mark & class/final test - mark and final test packaged product

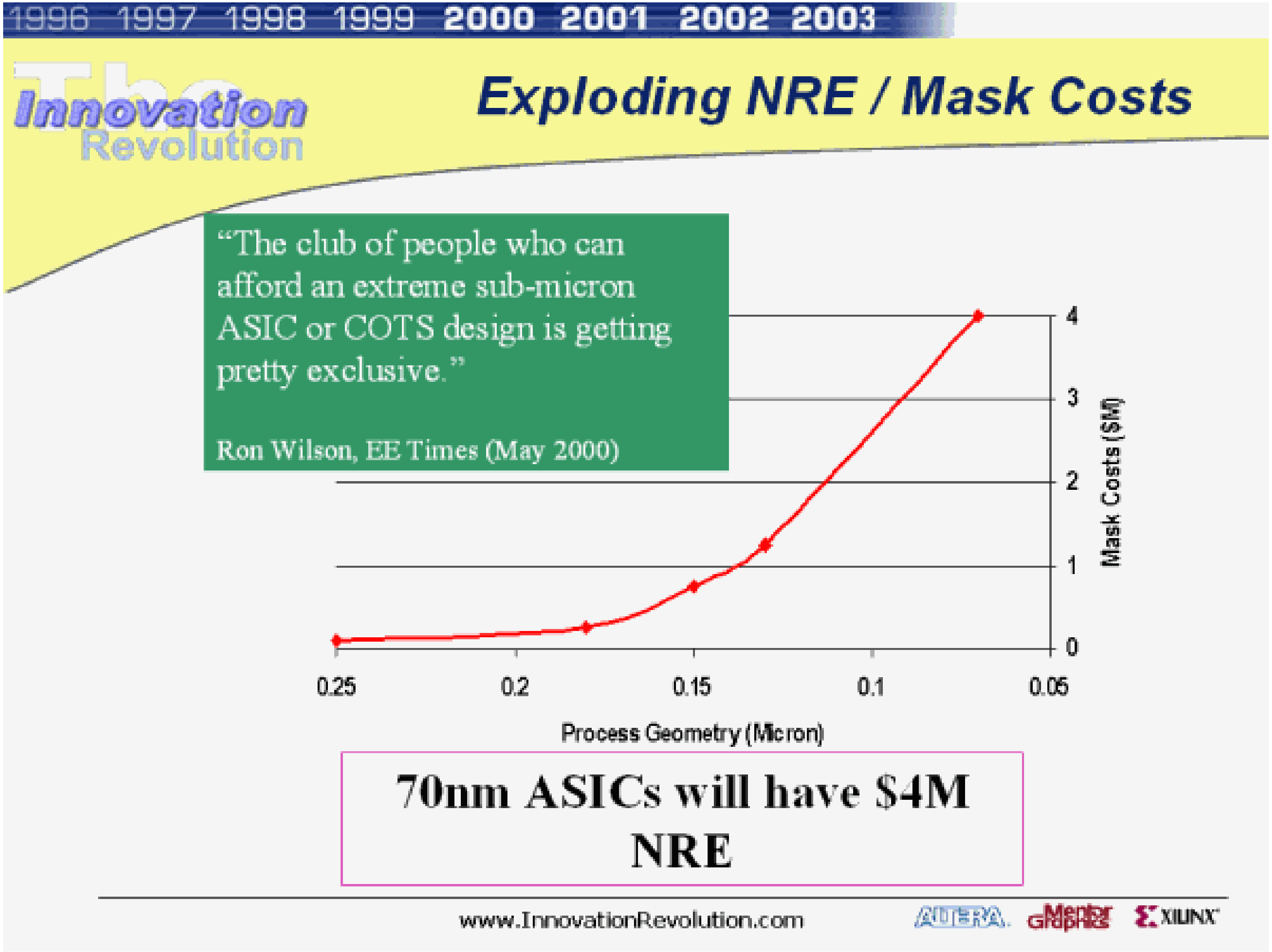
Cost of Integrated Circuits

- NRE (non-recurring engineering) costs
  - Fixed cost to produce the design
    - design effort
    - design verification effort
    - mask generation
  - Influenced by the design complexity and designer productivity
  - More pronounced for small volume products

- Recurring costs – proportional to product volume
  - silicon processing
    - also proportional to chip area
  - assembly (packaging)
  - test

$$\text{cost per IC} = \text{variable cost per IC} + \frac{\text{fixed cost}}{\text{volume}}$$

NRE Cost is Increasing



Exploding NRE / Mask Costs

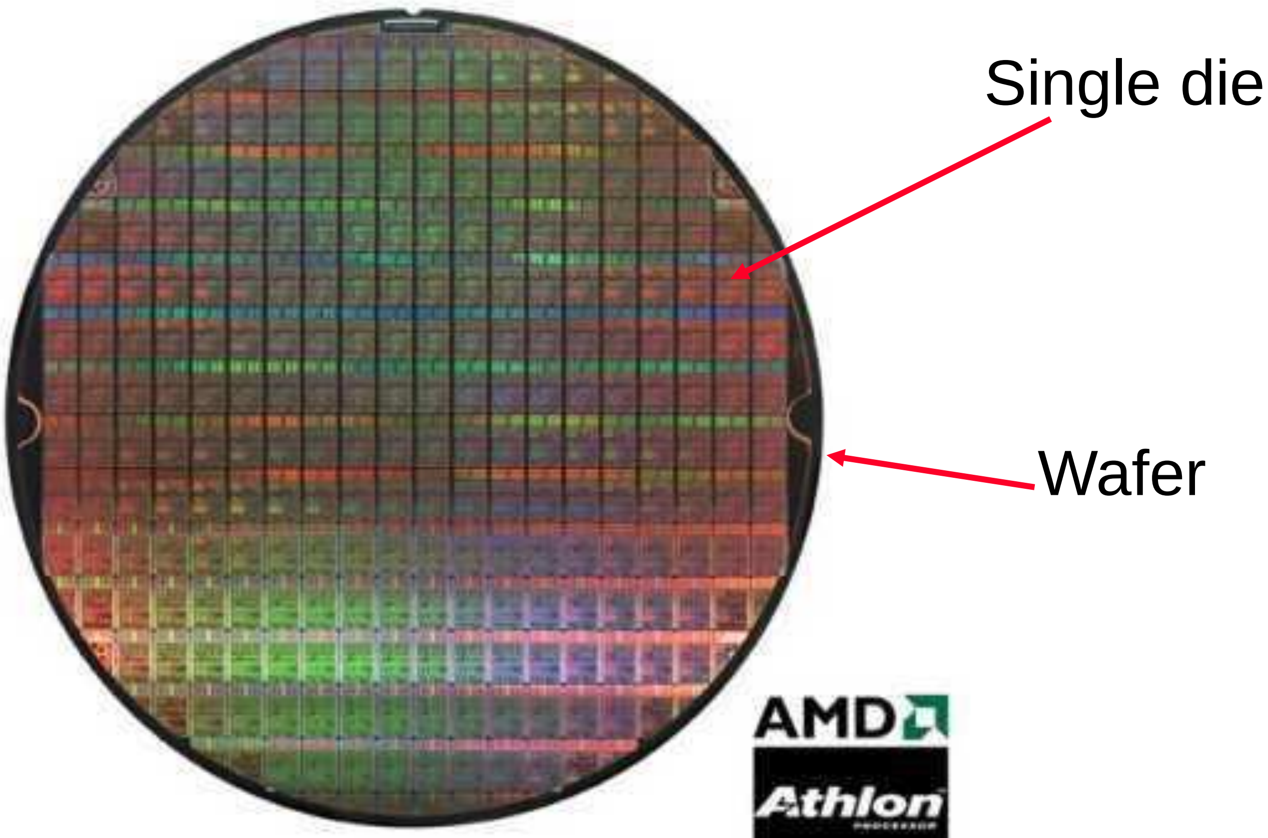
"The club of people who can afford an extreme sub-micron ASIC or COTS design is getting pretty exclusive."

Ron Wilson, EE Times (May 2000)

70nm ASICs will have $4M NRE

www.InnovationRevolution.com

Silicon Wafer



Single die

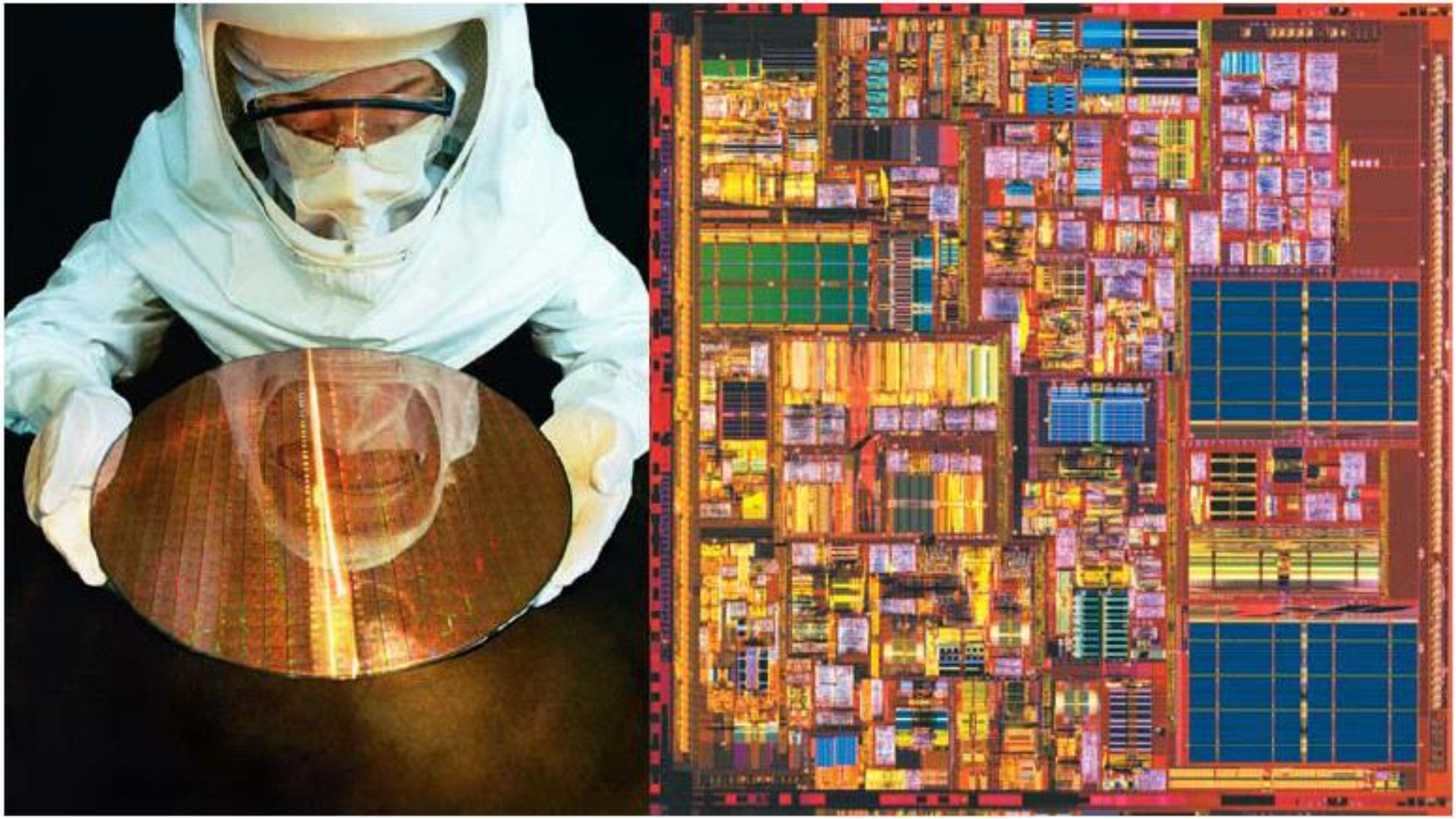
Wafer

AMD

Athlon

From http://www.amd.com

Silicon Wafer



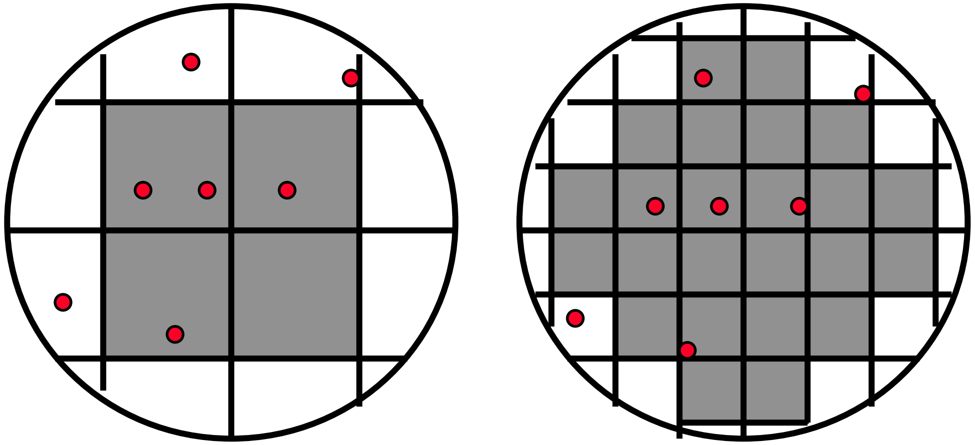
300mm wafer and Pentium 4 IC.   Photos courtesy of Intel.

Recurring Costs

$$\text{variable cost} = \frac{\text{cost of die} + \text{cost of die test} + \text{cost of packaging}}{\text{final test yield}}$$

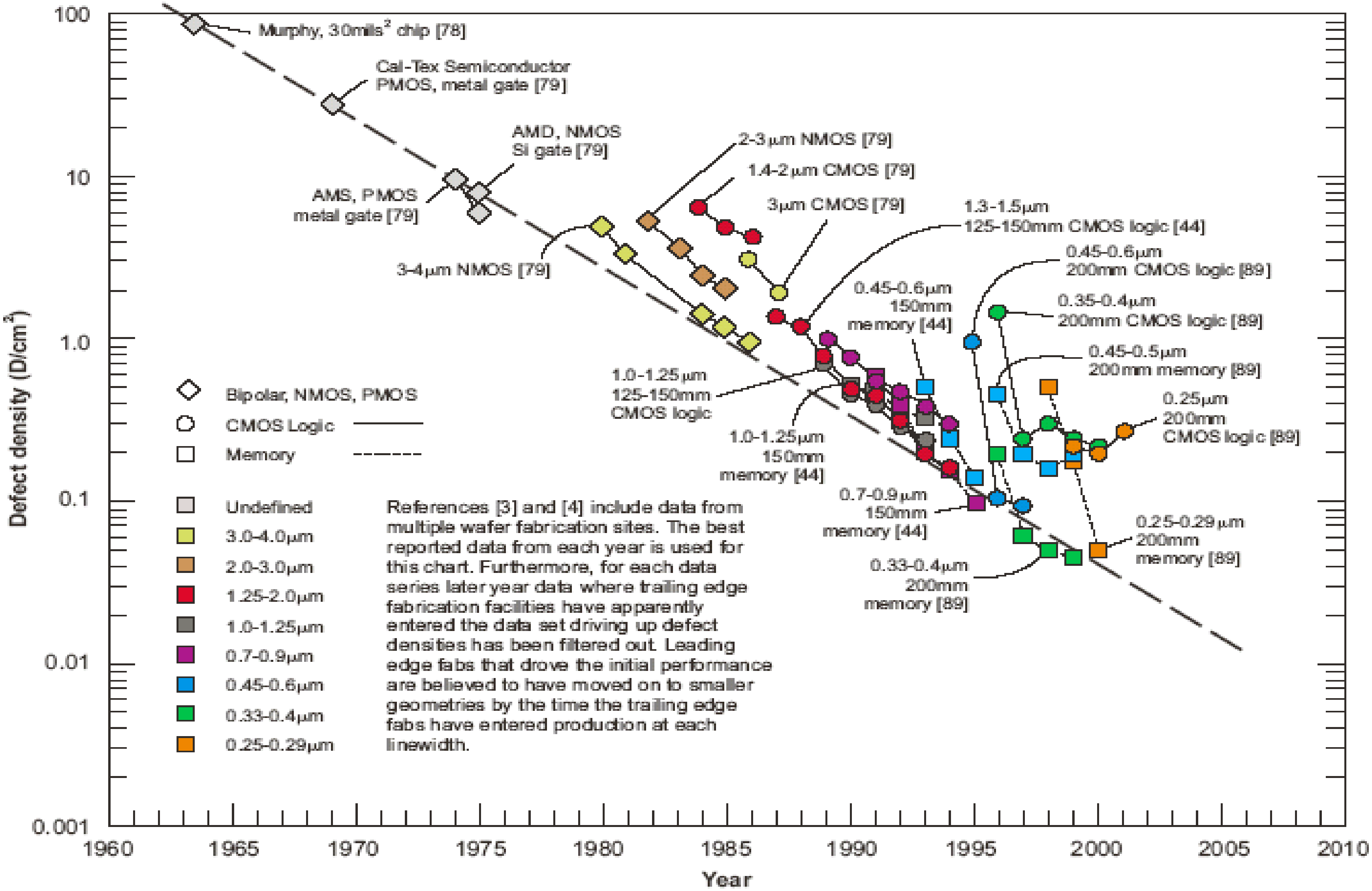
$$\text{cost of die} = \frac{\text{cost of wafer}}{\text{dies per wafer} \times \text{die yield}}$$

$$\text{dies per wafer} = \frac{\pi \times (\text{wafer diameter}/2)^2}{\text{die area}} - \frac{\pi \times \text{wafer diameter}}{\sqrt{2 \times \text{die area}}}$$



$$\text{die yield} = (1 + (\text{defects per unit area} \times \text{die area})/\alpha)^{-\alpha}$$

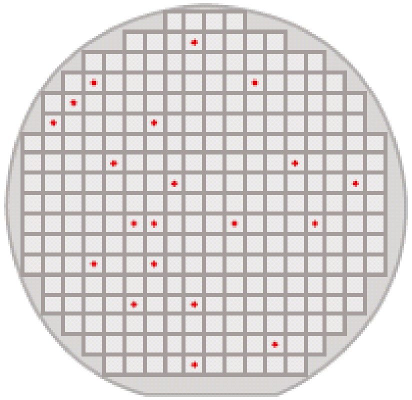
Defect Density Trends



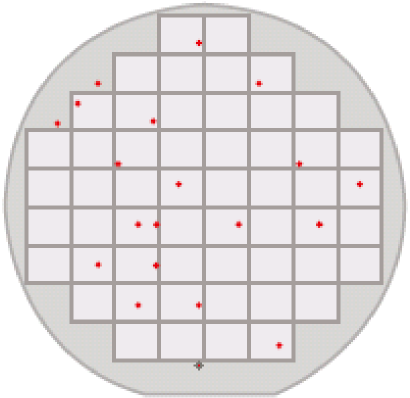
References [3] and [4] include data from multiple wafer fabrication sites. The best reported data from each year is used for this chart. Furthermore, for each data series later year data where trailing edge fabrication facilities have apparently entered the data set driving up defect densities has been filtered out. Leading edge fabs that drove the initial performance are believed to have moved on to smaller geometries by the time the trailing edge fabs have entered production at each linewidth.

Yield Example

- Example
  - wafer size of 12 inches, die size of 2.5 cm^2, 1 defects/cm^2, α = 3 (measure of manufacturing process complexity)
  - 252 dies/wafer (remember, wafers round & dies square)
  - die yield of 16%
  - 252 x 16% = only 40 dies/wafer die yield !



20 Defects
20 Bad Die
264 Gross Die
92% Yield



20 Defects
16 Bad Die
54 Gross Die
70% Yield

- Die cost is strong function of die area
  - proportional to the third or fourth power of the die area

Examples of Cost Metrics (circa 1994)

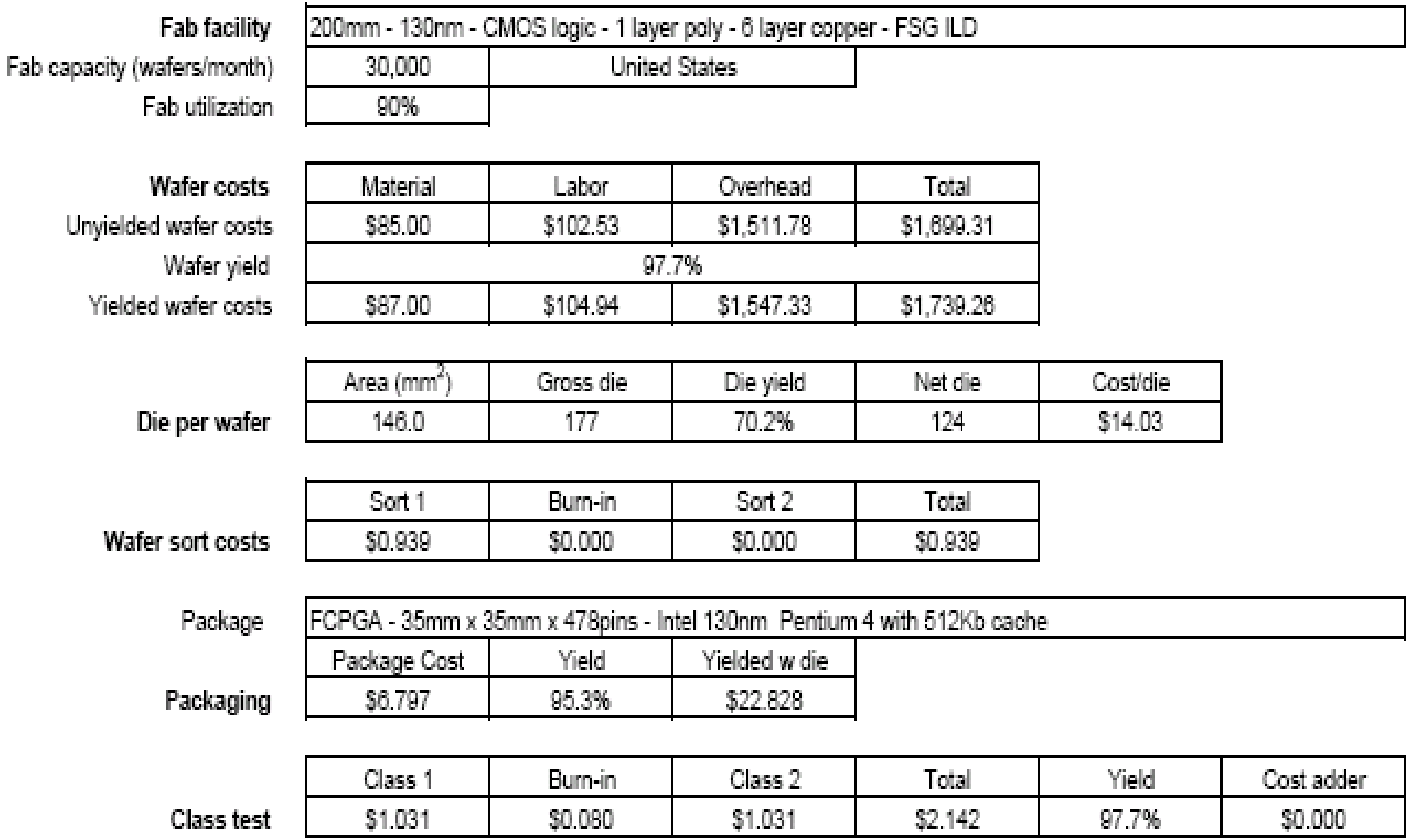
Chip	Metal layers	Line width	Wafer cost	Defects /cm^2	Area (mm^2)	Dies/ wafer	Yield	Die cost
386DX	2	0.90	$900	1.0	43	360	71%	$4
486DX2	3	0.80	$1200	1.0	81	181	54%	$12
PowerPC 601	4	0.80	$1700	1.3	121	115	28%	$53
HP PA 7100	3	0.80	$1300	1.0	196	66	27%	$73
DEC Alpha	3	0.70	$1500	1.2	234	53	19%	$149
Super SPARC	3	0.70	$1700	1.6	256	48	13%	$272
Pentium	3	0.80	$1500	1.5	296	40	9%	$417

Examples of Wafer Cost (2003)

Wafer size	Line- width (µm)	Mask layers					
			18	20	22	24	26
200mm	0.25		$890	$980	$1,070	$1,155	-
	0.18		-	$1,320	$1,440	$1,565	-
	0.13		-	-	$1,815	$1,970	$2,130
300mm	0.13		-	-	$2,500	$2,690	$2,890
	0.09		-	-	-	$2,860	$3,065

Source: Icknowledge.com

Examples of Cost Metrics (2003) Intel Pentium4



Fab facility	200mm - 130nm - CMOS logic - 1 layer poly - 6 layer copper - FSG ILD	
Fab capacity (wafers/month)	30,000	United States
Fab utilization	90%	

Wafer costs	Material	Labor	Overhead	Total
Unyielded wafer costs	$85.00	$102.53	$1,511.78	$1,699.31
Wafer yield	97.7%			
Yielded wafer costs	$87.00	$104.94	$1,547.33	$1,739.28

	Area (mm²)	Gross die	Die yield	Net die	Cost/die
Die per wafer	146.0	177	70.2%	124	$14.03

	Sort 1	Burn-in	Sort 2	Total
Wafer sort costs	$0.939	$0.000	$0.000	$0.939

Package	FCPGA - 35mm x 35mm x 478pins - Intel 130nm Pentium 4 with 512Kb cache		
	Package Cost	Yield	Yielded w die
Packaging	$6.797	95.3%	$22.828

	Class 1	Burn-in	Class 2	Total	Yield	Cost adder
Class test	$1.031	$0.080	$1.031	$2.142	97.7%	$0.000

Final cost: 25.57$ / chip source:www.icknowledge.com

Fundamental Design Metrics

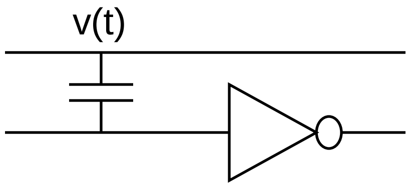
- Functionality
- Cost
  - NRE (fixed) costs - design effort
  - RE (variable) costs - cost of parts, assembly, test
- **Reliability, robustness**
  - Noise margins
  - Noise immunity
- Performance
  - Speed (delay)
  - Power consumption; energy
- Time-to-market

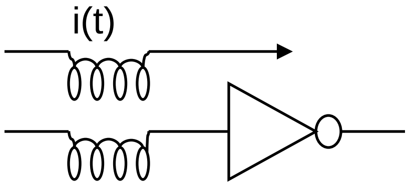
Reliability

Noise in Digital Integrated Circuits

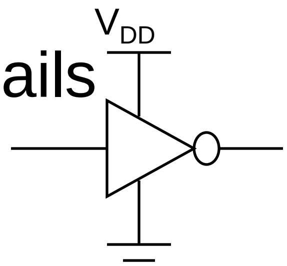
- Noise – unwanted variations of voltages and currents at the logic nodes

- From two wires placed side by side
  - capacitive coupling
    - voltage change on one wire can influence signal on the neighboring wire
    - cross talk
  - inductive coupling
    - current change on one wire can influence signal on the neighboring wire





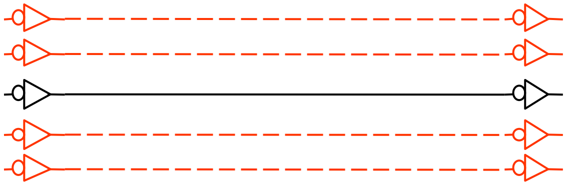
- From noise on the power and ground supply rails
  - can influence signal levels in the gate



Example of Capacitive Coupling

- Signal wire glitches as large as 80% of the supply voltage will be common due to crosstalk between neighboring wires as feature sizes continue to scale

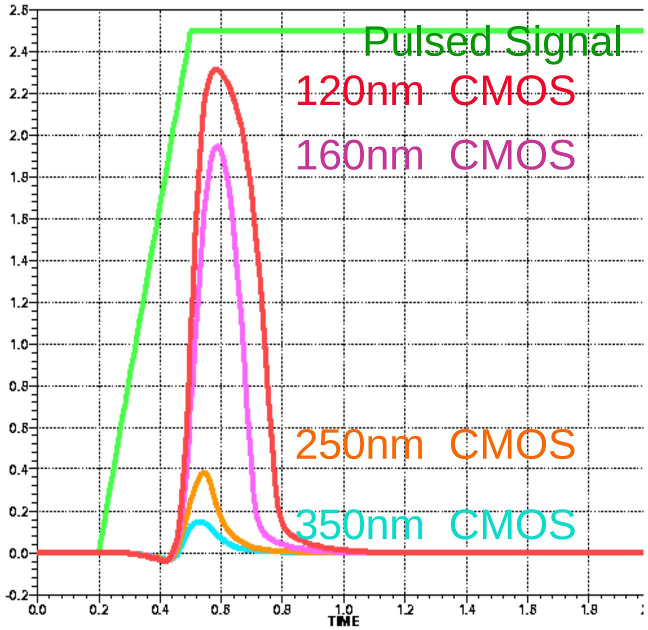
Crosstalk vs. Technology



Black line quiet

Red lines pulsed

Glitches strength vs technology



From Dunlop, Lucent, 2000

Static Gate Behavior

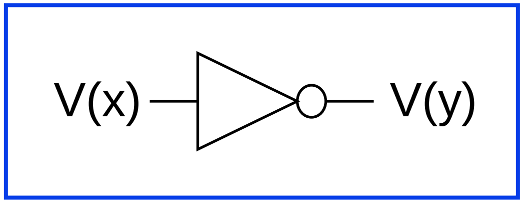
- Steady-state parameters of a gate – *static behavior* – tell how robust a circuit is with respect to both variations in the manufacturing process and to noise disturbances.

- Digital circuits perform operations on Boolean variables

$$x \in \{0,1\}$$

- A logical variable is associated with a *nominal voltage level* for each logic state

$$1 \Leftrightarrow V_{OH} \text{ and } 0 \Leftrightarrow V_{OL}$$



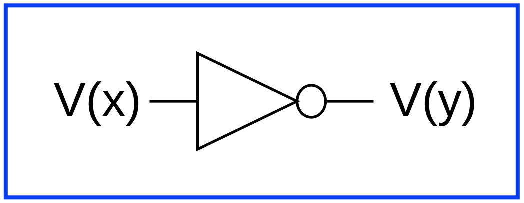
$$V_{OH} = !\,(V_{OL})$$
$$V_{OL} = !\,(V_{OH})$$

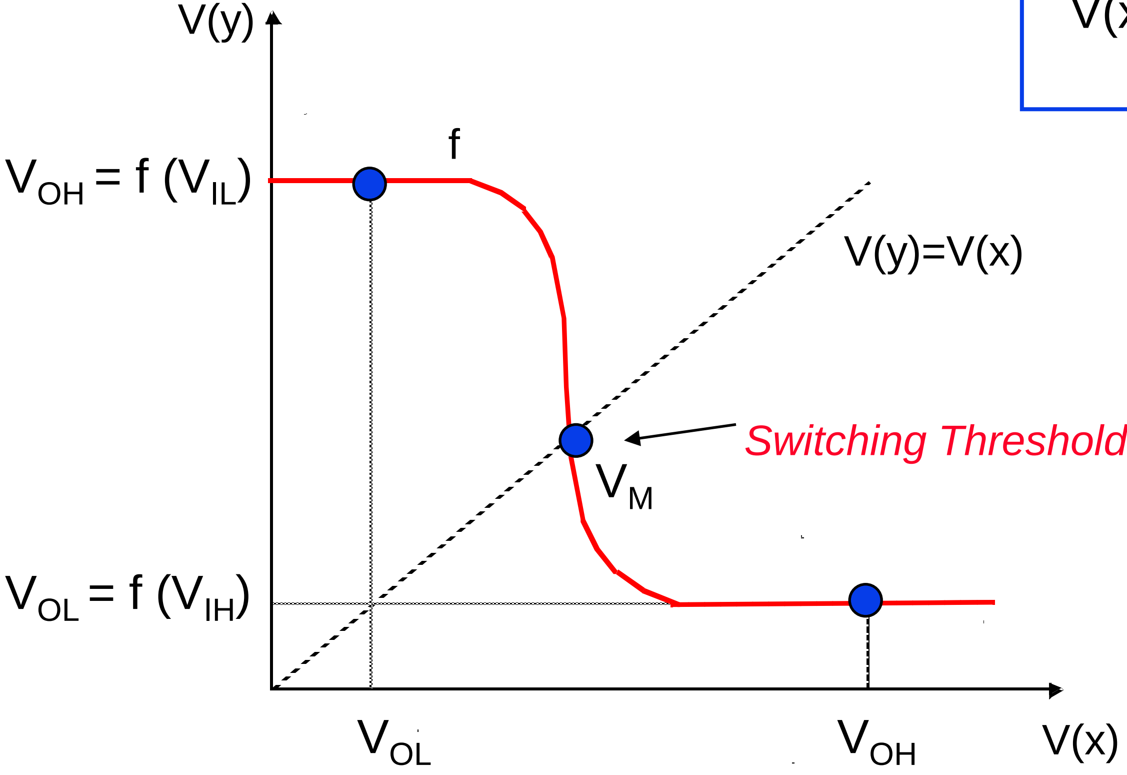
- Difference between V_{OH} and V_{OL} is the logic or *signal swing* $V_{sw} = V_{OH} - V_{OL}$

DC Operation

Voltage Transfer Characteristics (VTC)

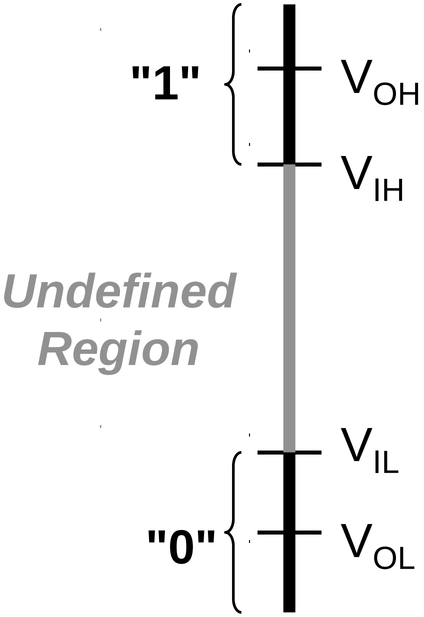
- Plot of output voltage as a function of the input voltage

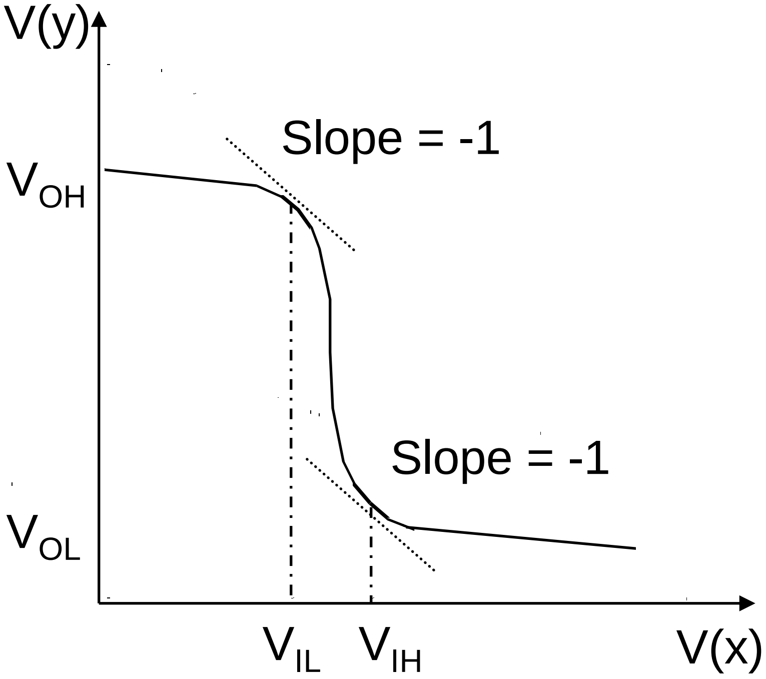




Mapping Logic Levels to the Voltage Domain

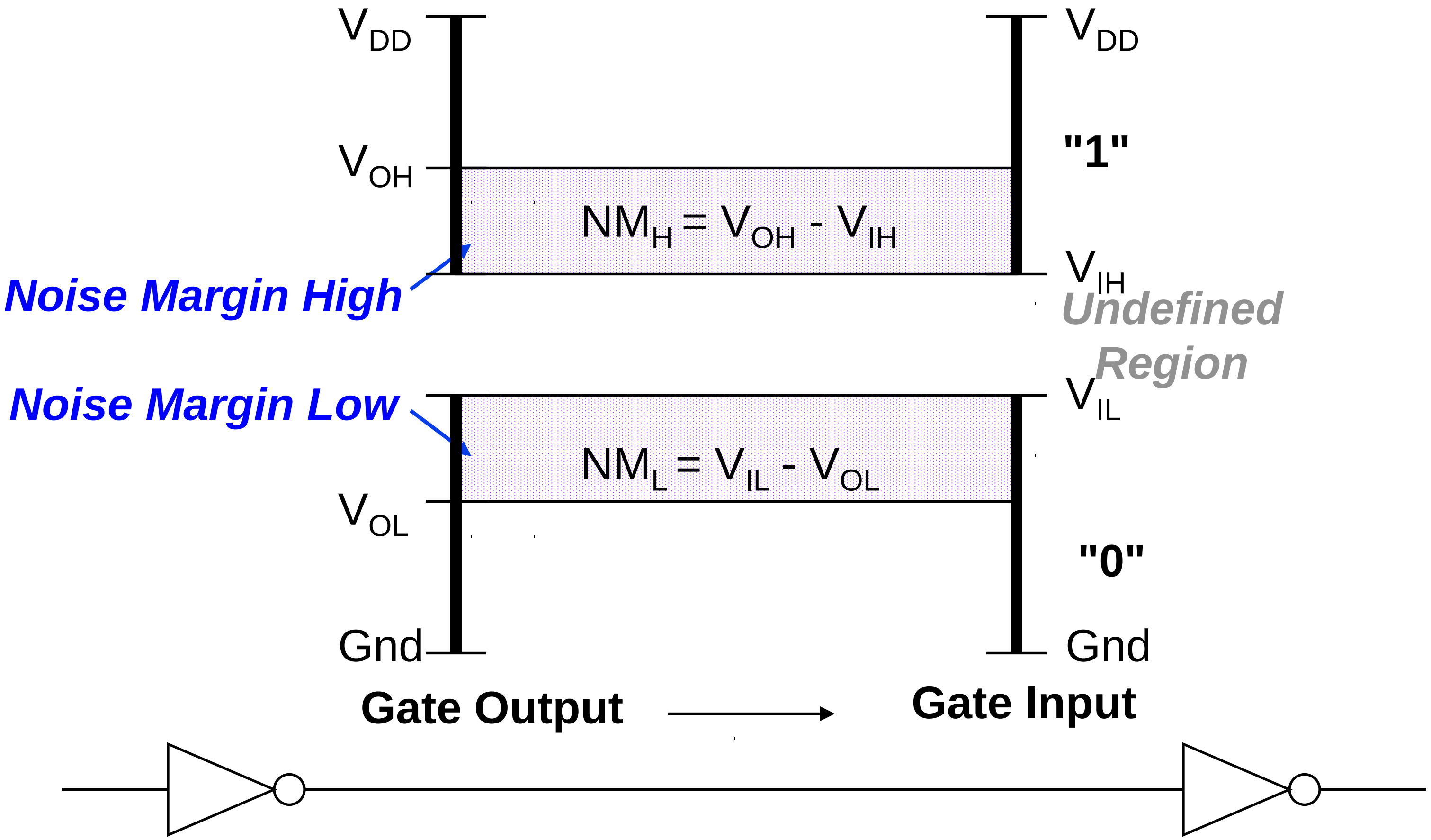
- The regions of acceptable high and low voltages are delimited by V_{IH} and V_{IL} that represent the points on the VTC curve where the gain = -1





Noise Margins

- For robust circuits, want the “0” and “1” intervals to be as large as possible



- Large noise margins are desirable, but not sufficient …

Noise Immunity

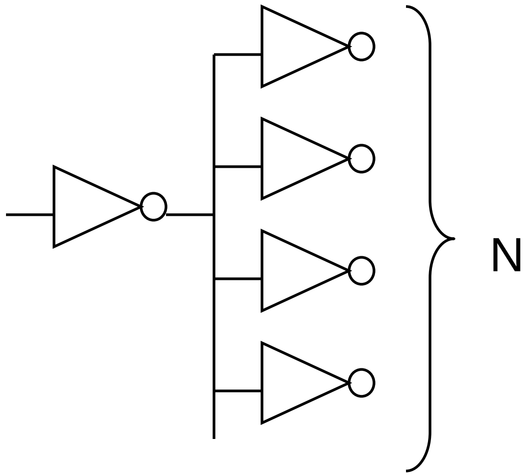
- Noise margin expresses the ability of a circuit to overpower a noise source
  - noise sources: supply noise, cross talk, interference, offset
- Absolute noise margin values are deceptive
  - a floating node is more easily disturbed than a node driven by a low impedance (in terms of voltage)
- *Noise immunity* expresses the ability of the system to process and transmit information correctly in the presence of noise
- For good noise immunity, the signal swing (i.e., the difference between V_{OH} and V_{OL}) and the noise margin have to be large enough to overpower the impact of fixed sources of noise

Fundamental Design Metrics

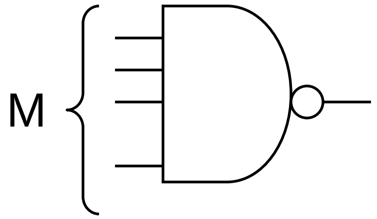
- Functionality
- Cost
  - NRE (fixed) costs - design effort
  - RE (variable) costs - cost of parts, assembly, test
- Reliability, robustness
  - Noise margins
  - Noise immunity
- **Performance**
  - Speed (delay)
  - Power consumption; energy
- Time-to-market

Fan-In and Fan-Out

- Fan-out – number of load gates connected to the output of the driving gate
  - gates with large fan-out are slower



- Fan-in – the number of inputs to the gate
  - gates with large fan-in are bigger and slower



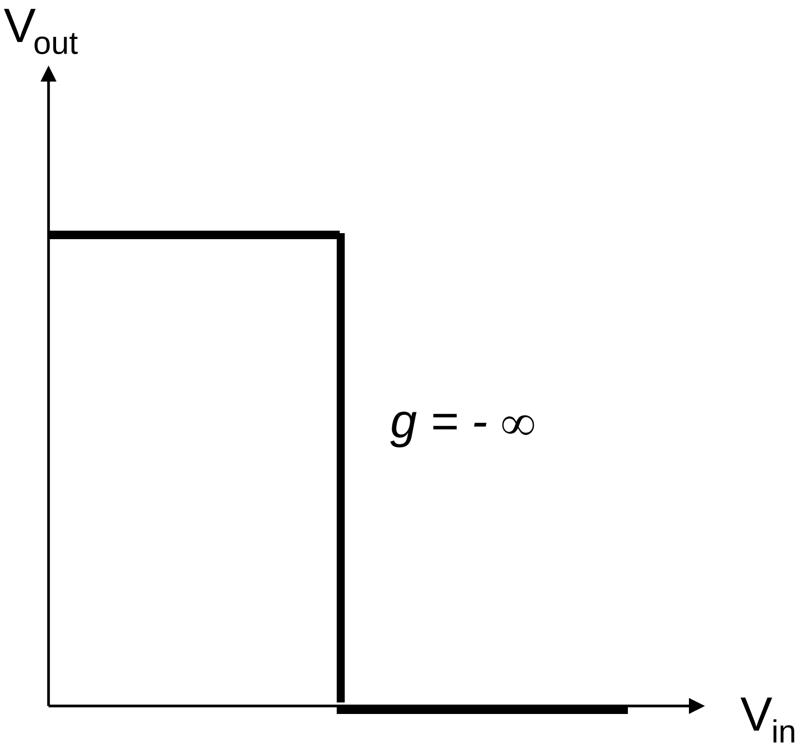
Directivity

- A gate must be *undirectional*: changes in an output level should not appear at any unchanging input of the same circuit
  - In real circuits *full* directivity is an illusion (e.g., due to capacitive coupling between inputs and outputs)

- Key metrics: *output impedance* of the driver and *input impedance* of the receiver
  - ideally, the output impedance of the driver should be zero and
  - input impedance of the receiver should be infinity

The Ideal Inverter

- The ideal gate should have
  - infinite gain in the transition region
  - a gate threshold located in the middle of the logic swing
  - high and low noise margins equal to half the swing
  - input and output impedances of infinity and zero, resp.



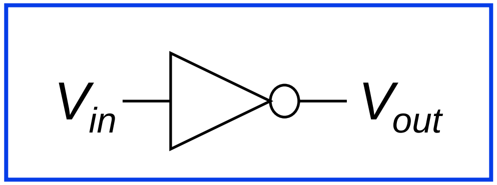
$R_i = \infty$

$R_o = 0$

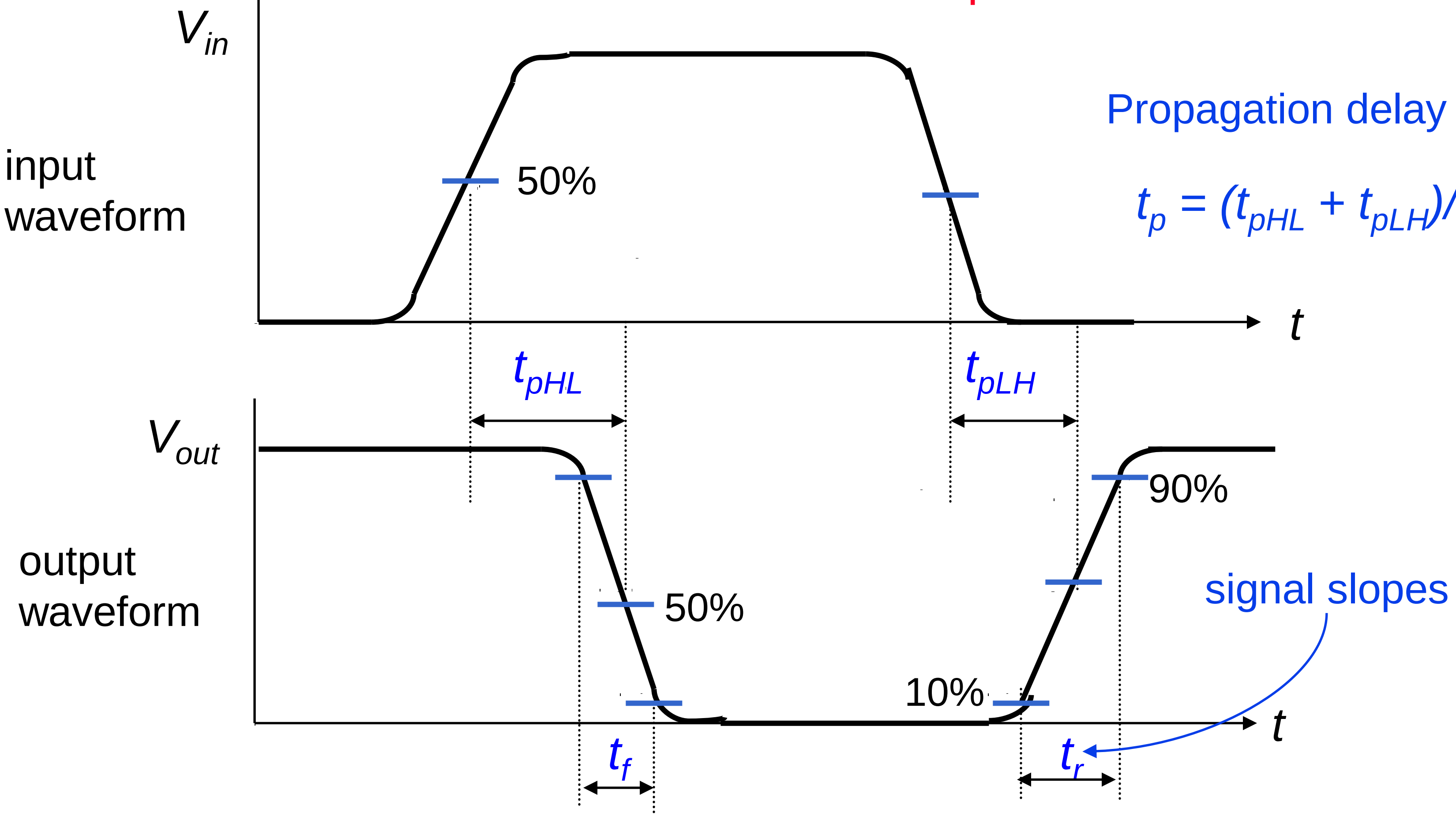
Fanout = ∞

$NM_H = NM_L = VDD/2$

Delay Definitions



T_p is an artificial gate quality metric,no physical meaning per se

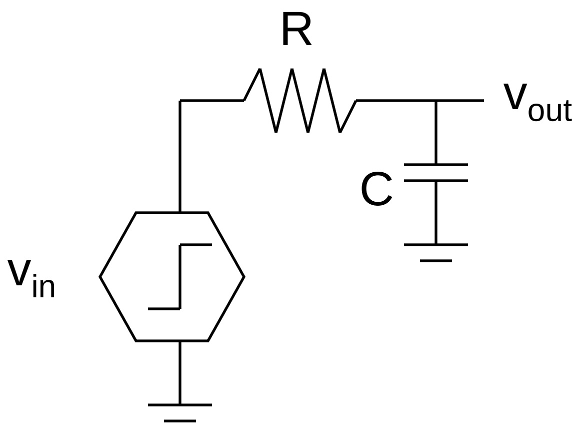


Propagation delay

$t_p = (t_{pHL} + t_{pLH})/2$

Modeling Propagation Delay

- Model circuit as first-order RC network



$$v_{out}(t) = (1 - e^{-t/\tau})V$$

where $\tau = RC$

Time to reach 50% point is
$t = \ln(2)\,\tau = 0.69\,\tau$

Time to reach 90% point is
$t = \ln(9)\,\tau = 2.2\,\tau$

- Matches the delay of an inverter gate

Power and Energy Dissipation

- Power consumption: how much energy is consumed per operation and how much heat the circuit dissipates
  - supply line sizing (determined by peak power)

$$P_{peak} = V_{dd} i_{peak}$$

Power and Energy Dissipation

- Power consumption: how much energy is consumed per operation and how much heat the circuit dissipates
  - supply line sizing (determined by peak power)

    $$P_{peak} = V_{dd} i_{peak}$$

  - battery lifetime (determined by average power dissipation)

    $p(t) = v(t)i(t) = V_{dd}i(t)$ $\qquad P_{avg} = 1/T \int p(t)\,dt = V_{dd}/T \int i_{dd}(t)\,dt$

  - packaging and cooling requirements

- Two important components: static and dynamic

$$P\ (\text{watts}) = C_L\ V_{dd}^2\ f_{0\rightarrow 1} + t_{sc} V_{dd}\ I_{peak}\ f_{0\rightarrow 1} + V_{dd}\ I_{leakage}$$

$$f_{0\rightarrow 1} = P_{0\rightarrow 1} * f_{clock}$$

Power and Energy Dissipation

- Propagation delay and the power consumption of a gate are related

- Propagation delay is (mostly) determined by the speed at which a given amount of energy can be stored on the gate capacitors
  - the faster the energy transfer (higher power dissipation) the faster the gate

- For a given technology and gate topology, the product of the power consumption and the propagation delay is a constant
  - Power-delay product (PDP) – energy consumed by the gate per switching event

- An ideal gate is one that is fast and consumes little energy, so the ultimate quality metric is
  - Energy-delay product (EDP) = power-delay ^{2}

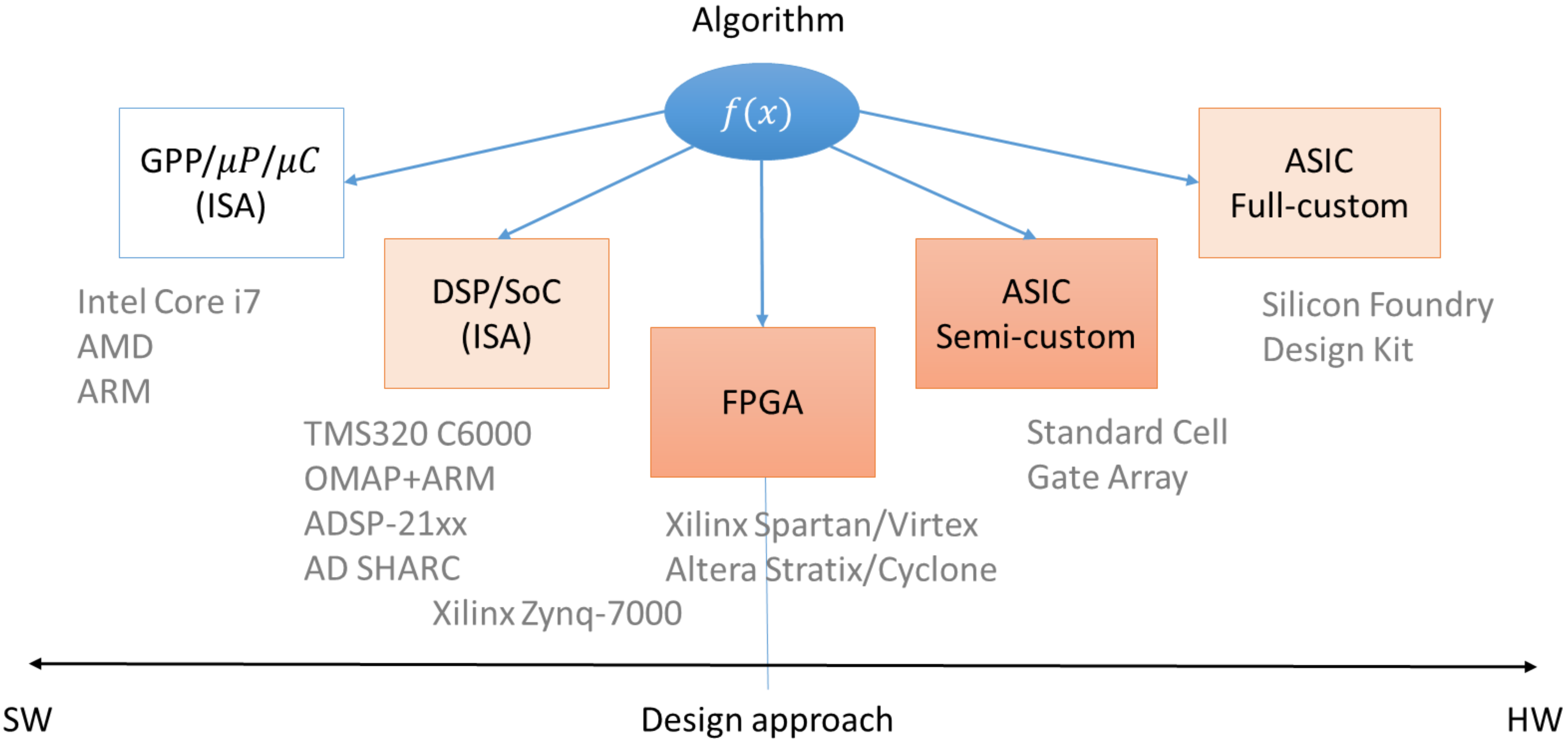
Summary

- Digital integrated circuits have come a long way and still have quite some potential left for the coming decades
- Some interesting challenges ahead
  - Getting a clear perspective on the challenges and potential solutions is the purpose of this course
- Understanding the design metrics that govern digital design is crucial
  - Cost, reliability, speed, power and energy dissipation

Outline

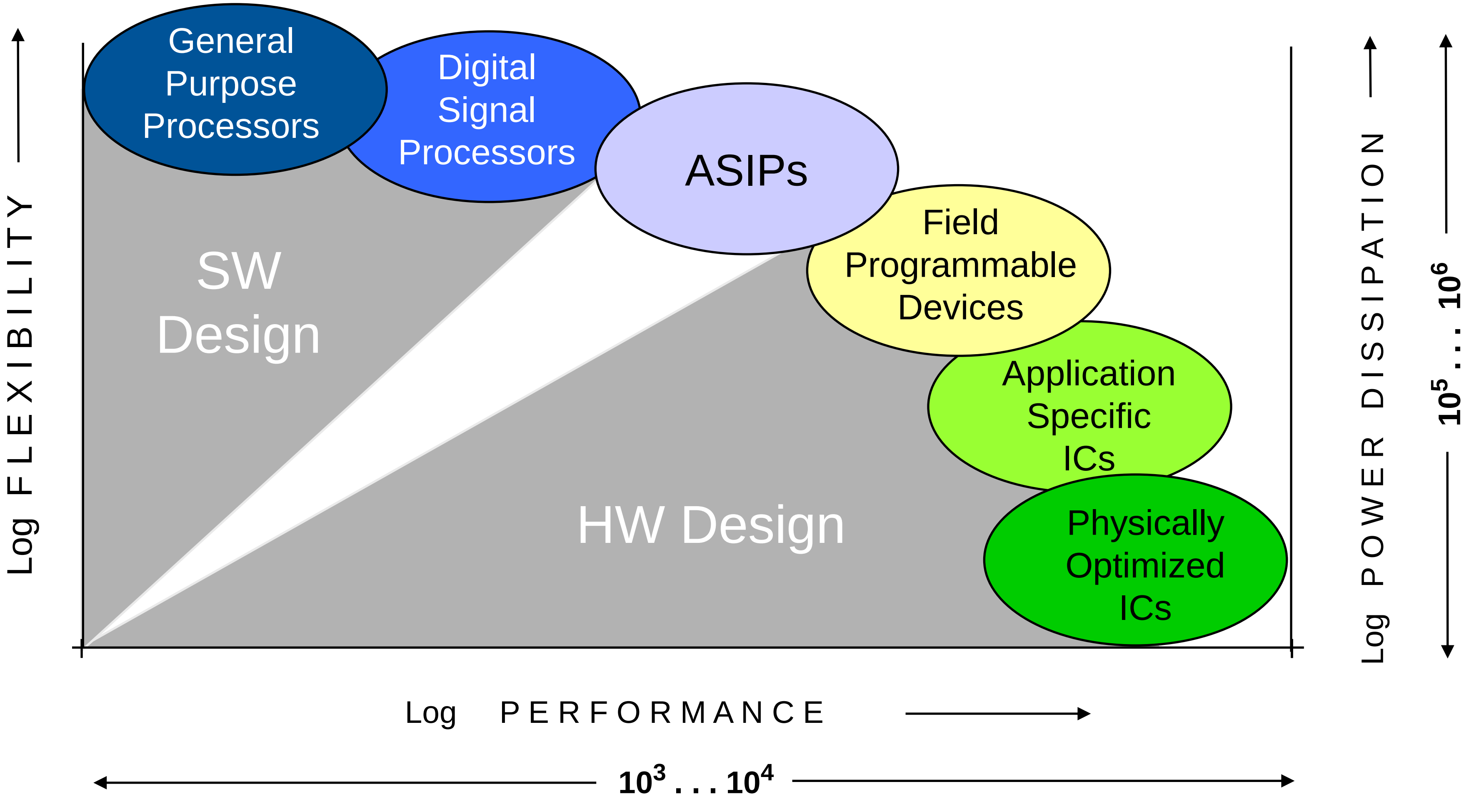
- Design Metrics
- Managing engineering trade-offs

How to map a system/algorithm to hw/sw design



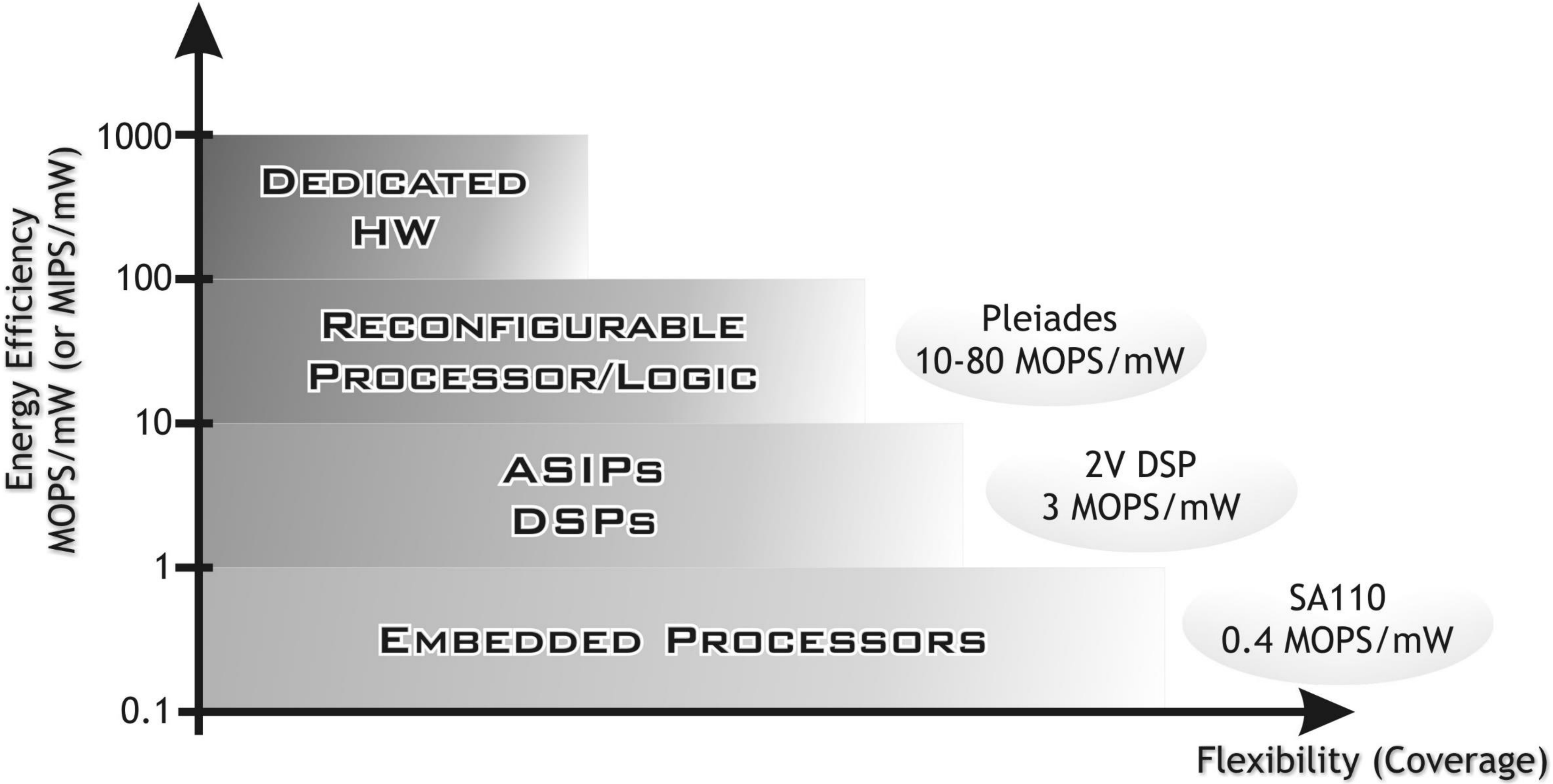
Design is the art of managing engineering trade-offs !

The Energy-Flexibility Gap



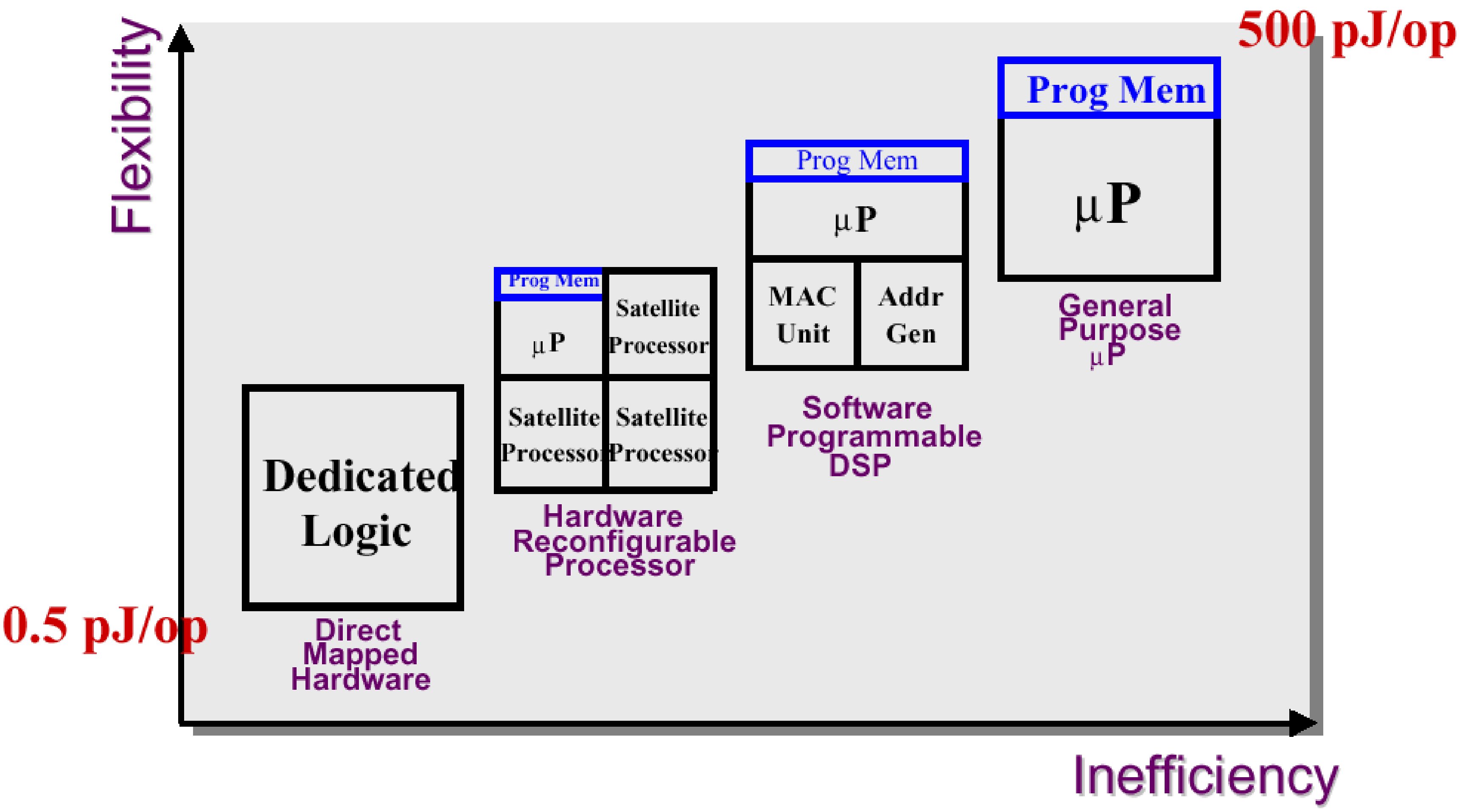
Source: T.Noll, RWTH Aachen

The Energy-Flexibility Gap



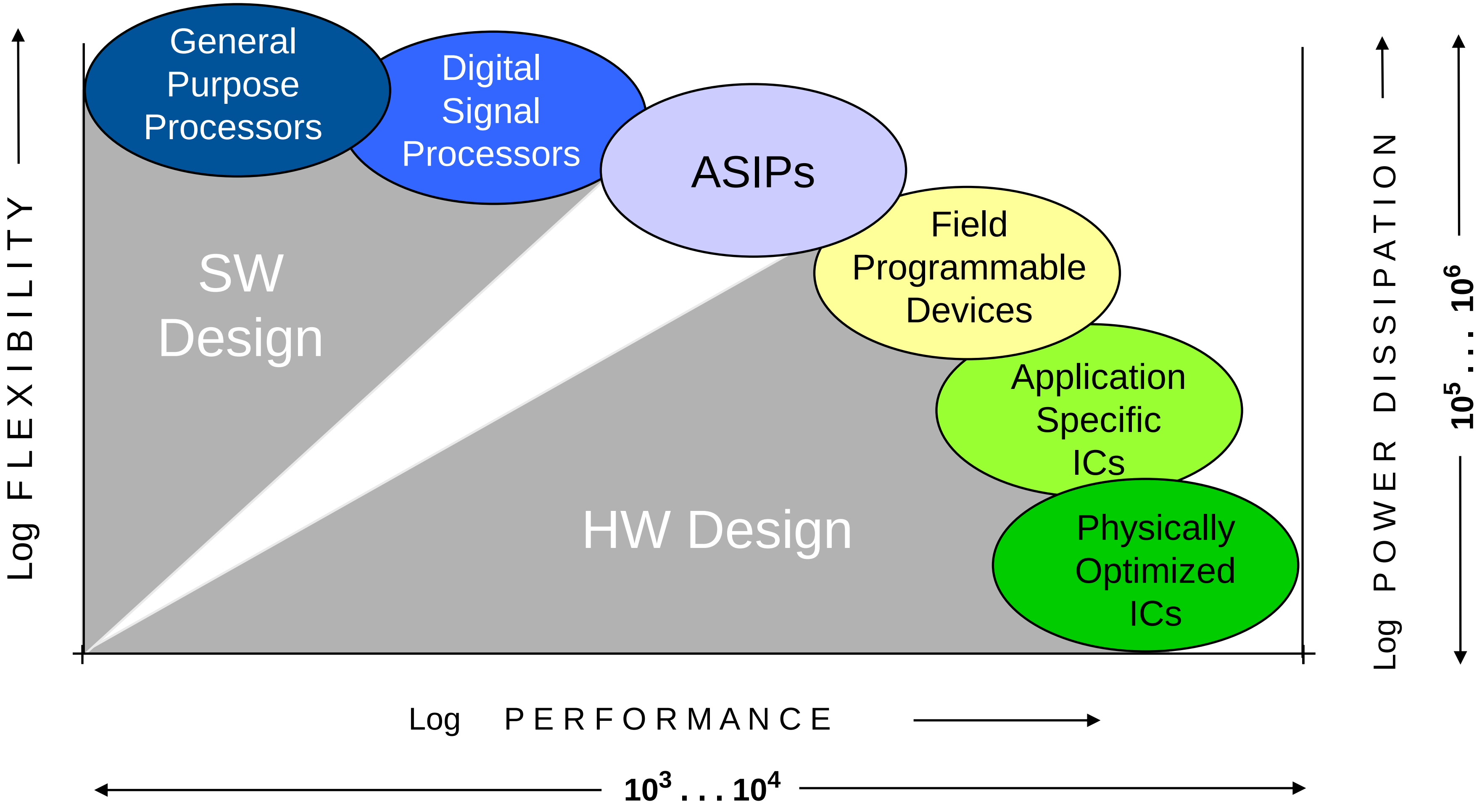
Source: Rabaey, BRWC

The Energy-Flexibility Gap



Source: Rabaey, BRWC

The Energy-Flexibility Gap



Source: T.Noll, RWTH Aachen

SoC elements

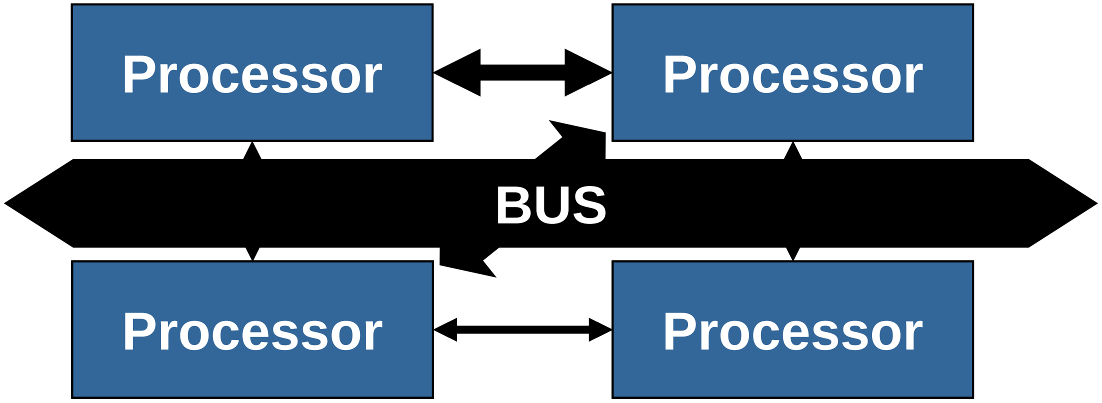
programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

How to address growing SoC complexity?

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

MP-SoC solution:

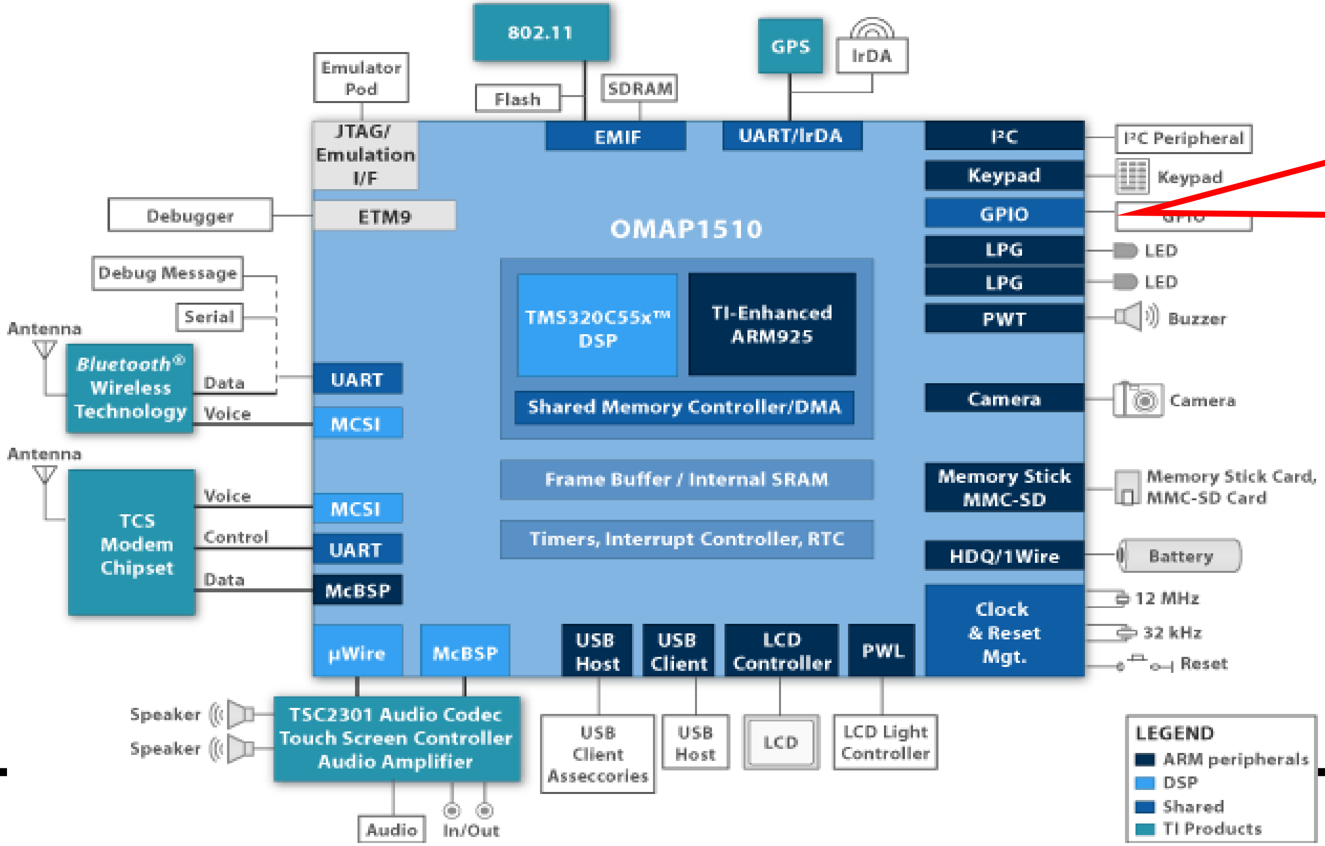


- + Programmable solution
- + Utilizing existing tools

- - Scheduling and Mapping may be difficult
- - Throughput constraints not achievable

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

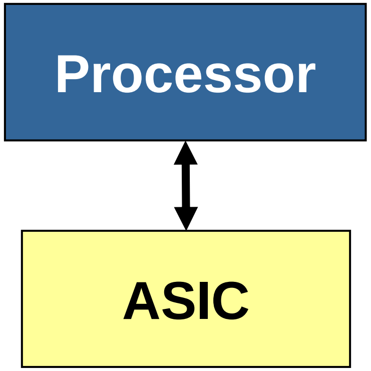


Commercial Example:
TI OMAP (1510)

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

Programmable Solution and Accelerator:


Processor

ASIC

+ Programmable solution

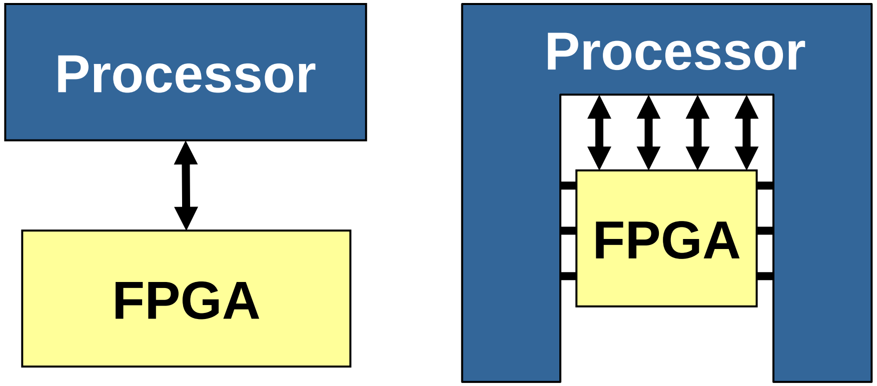
- Utilizing existing tools ?

- Flexibility bound to given limitations:
  - Instruction Set
  - Processor/ASIC Interface
  - Memory Interface
  - Architecture Timing

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

Programmable Solution and Accelerator:



loosely coupled

tightly coupled (embedded into infrastructure)

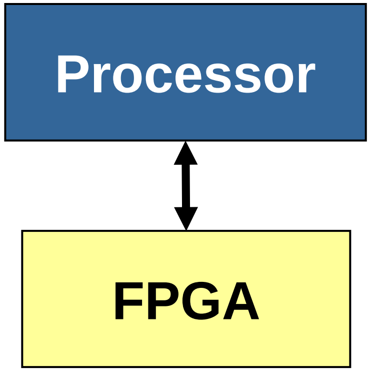
+ Programmable solution
+ High Performance

- Utilizing existing tools ?

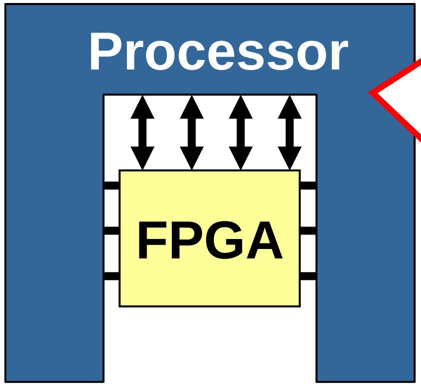
- Flexibility bound to given limitations:
  - Instruction Set
  - Processor/FPGA Interface
  - Memory Interface
  - Architecture Timing

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	



loosely coupled

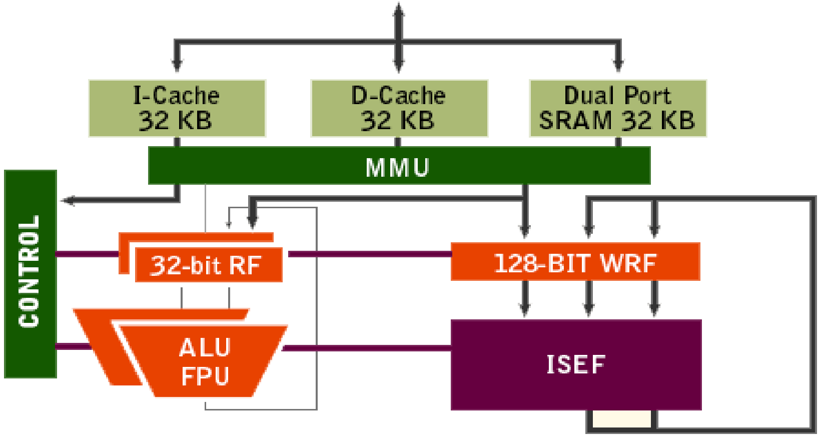


tightly coupled (embedded into infrastructure)

Comercial Example: Stretch Inc.

load/store up to 3x/2x 128 bit

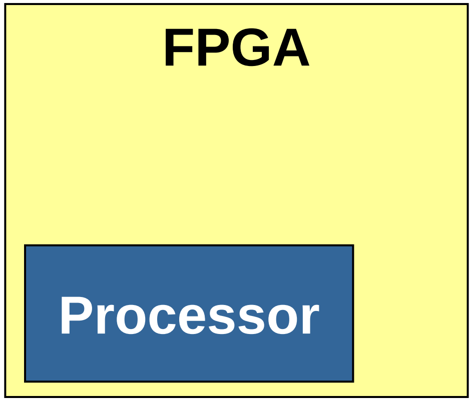
User-defined extensions to the core ISA



SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

Programmable Solution and Accelerator:



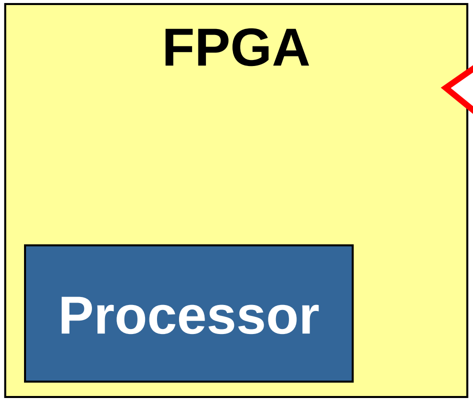
+ Programmable solution

- Utilizing existing tools ?

- Energy Efficiency
- Area/Timing Efficency
- Flexibility bound to given limitations:
  - Instruction Set
  - Processor/FPGA Interface
  - Memory Interface
  - Architecture Timing

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	



Comercial Example:

XILINX, Virtex-II Pro

Single or multiple IBM PowerPC Processors are embedded in the FPGA (Platform FPGA).

XILINX, Virtex-4 FPGA family

Logic Cells, RAM, XtremeDSP™ Slices, PowerPC Processor Blocks,
10/100/1000 Ethernet MAC Blocks, RocketIO™ Serial Transceivers

SoC elements

programmable	reconfigurable	fixed
DSP, µC, RISC, GPP, …	FPGA / eFPGA	ASIC
Compiler, Assembler, Linker, Simulator, Debugger	Gate-Level Synthesis Tools, Power Analysis Tools, HW-Simulator	

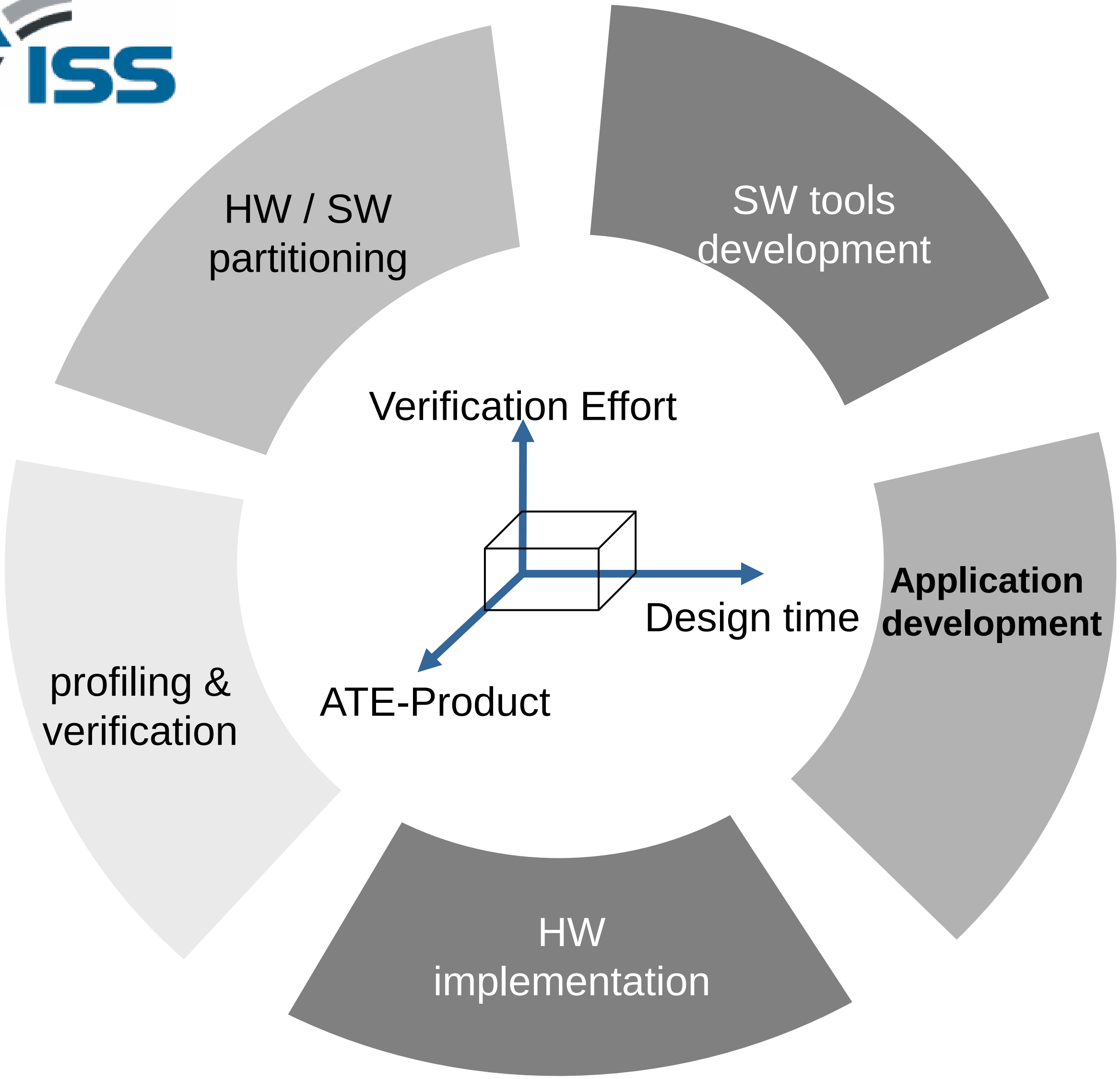
To map the Architecture to the Application means:

>> Consider jointly HW and SW for an efficient solution
>> Identify hotspots of the application and ...
>> implement complex instructions.

Without being limited to given structures or interfaces!

Application Specific Instruction Set Processor (ASIP)

How to move through the design space?



Trade-offs:

Area vs. Timing vs. Energy

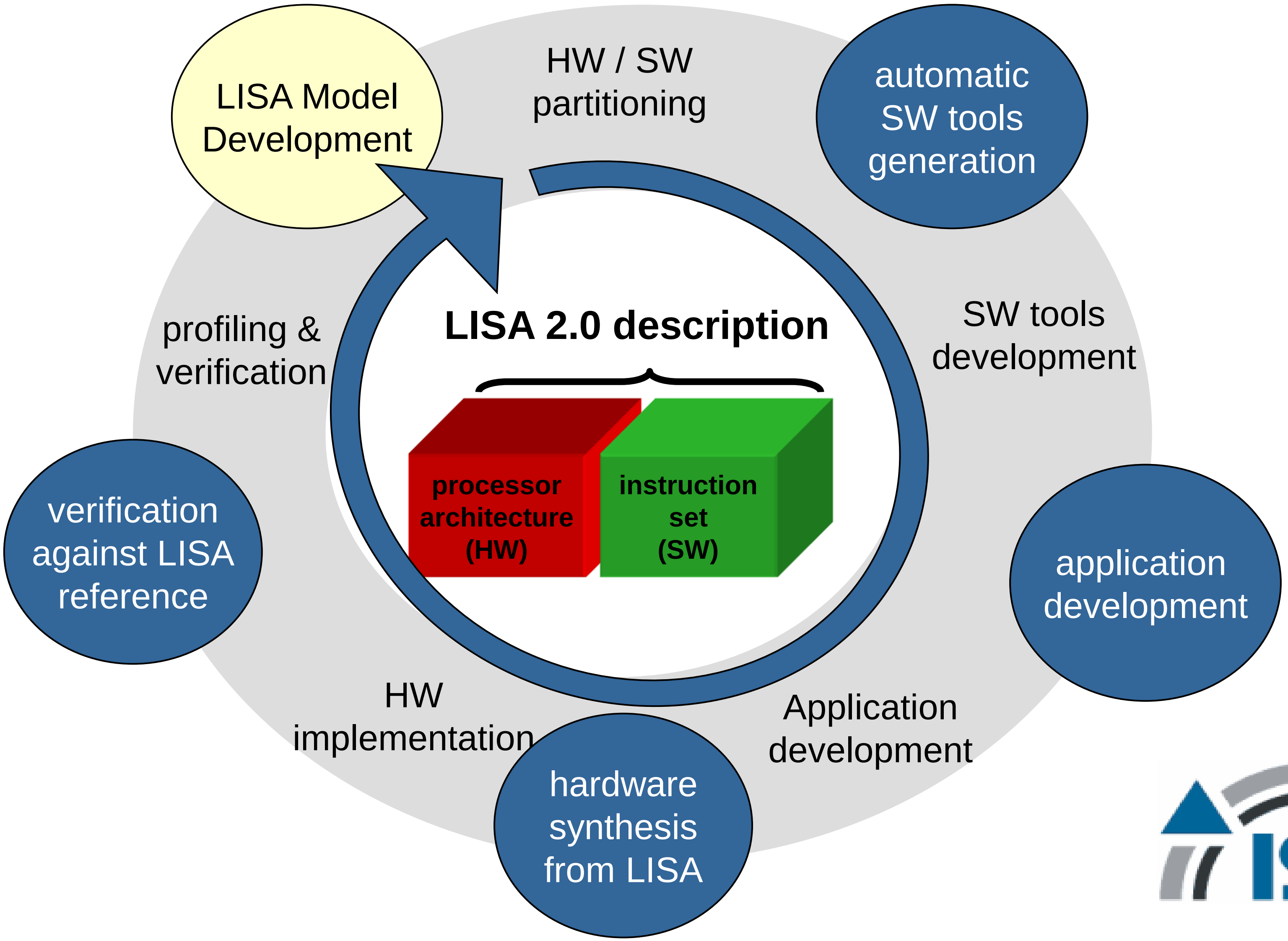
Design Time vs. ATE-Product

Architectural Features vs. Verification Effort

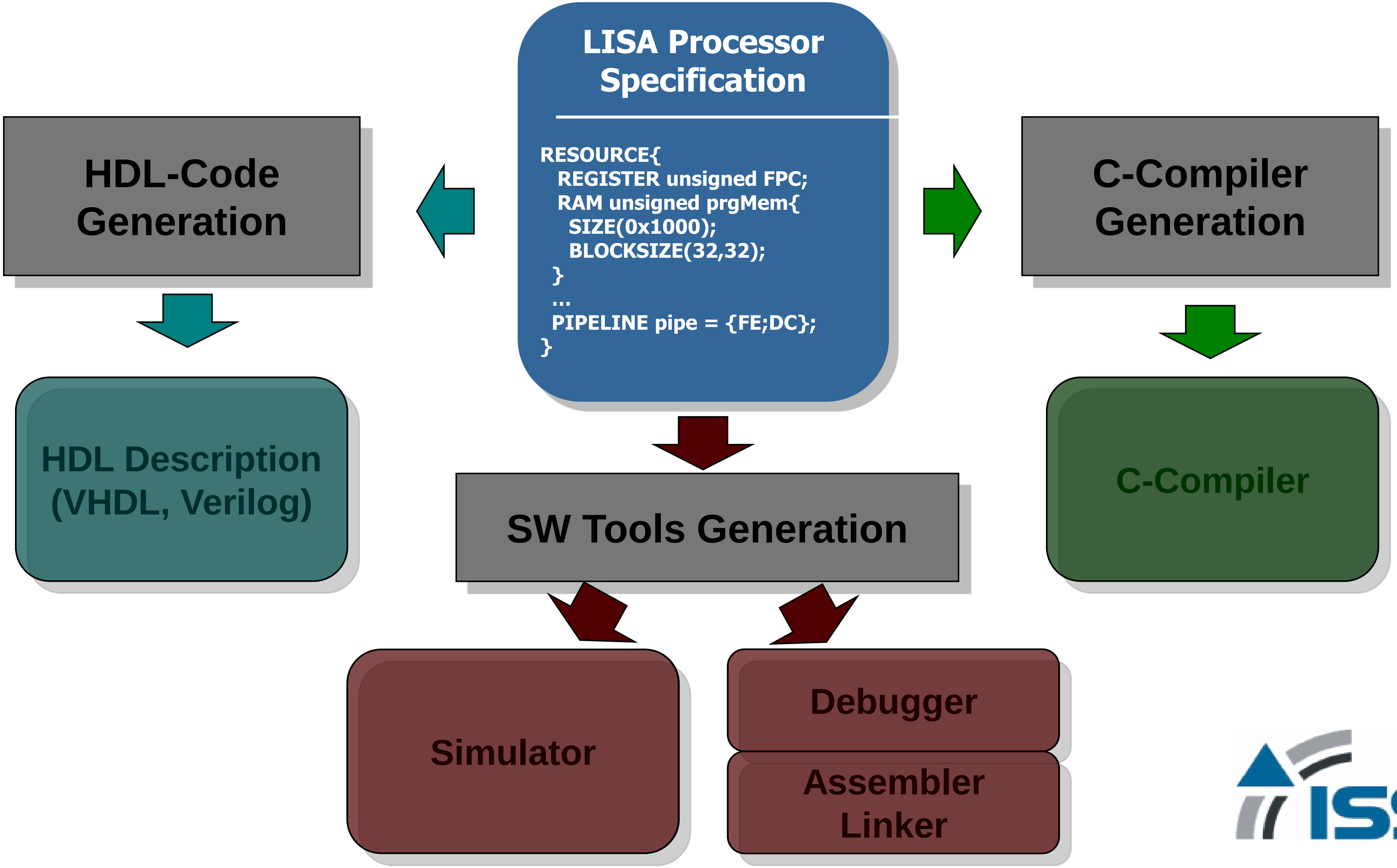
Verification Effort vs. ATE-Product

Flexibility vs. ATE-Product

ADL (Architecture Design Language) LISA



Centralized Processor Design with LISA



LISA Processor Specification

```
RESOURCE{
  REGISTER unsigned FPC;
  RAM unsigned prgMem{
    SIZE(0x1000);
    BLOCKSIZE(32,32);
  }
  ...
  PIPELINE pipe = {FE;DC};
}
```

End, Questions ?

- Design Metrics
- Managing engineering trade-offs