VLSI microarchitecture

Scaling

Toni Juan

Short index

- **Administrative**
  - glass schedule
  - projects?
- **Topic of the day: Scaling**

Administrative

Class Schedule

Day	Topic	Lecturer
06 Nov 00	Introduction Tech. & μArch.	T+E
13 Nov 00	Introduction To MOS VLSI	E
20 Nov 00	Scaling of wires & Trans.	T
27 Nov 00	Implem./Techn. MOS	E
04 Dec 00	Delay μArchitecture	T
11 Dec 00	Delay VLSI	E
18 Dec 00	Power μArchitecture	T
08 Dec 00	Power VLSI	E
15 Jan 01	Area μArchitecture	T
18 Jan 01	Case studies & alternatives	T+E

Projects

- **In groups of two people**

- **Take your favorite structure and:**
  - architect several implementations (block level)
  - find critical path
  - delay estimation
  - delay measurement (SPICE)
  - power estimation
  - area estimation
  - write report

Introduction

Why Scaling?

- **We care about technology improvements**
- **Scaling (reduction of minimum feature size)**
  - faster devices
  - more devices per area unit
- **Other important things (that we won't consider...)**
  - die size
  - metal levels
  - new conductors and dielectrics
  - ...

SCALING for Architects

- **What is scaling**
  - summary of impact of technology improvements
- **How is it measured, what does it mean**
- **Scaling for**
  - gates
  - wires
- **Limits for scaling**
- **Impact on**
  - delay
  - power
  - area

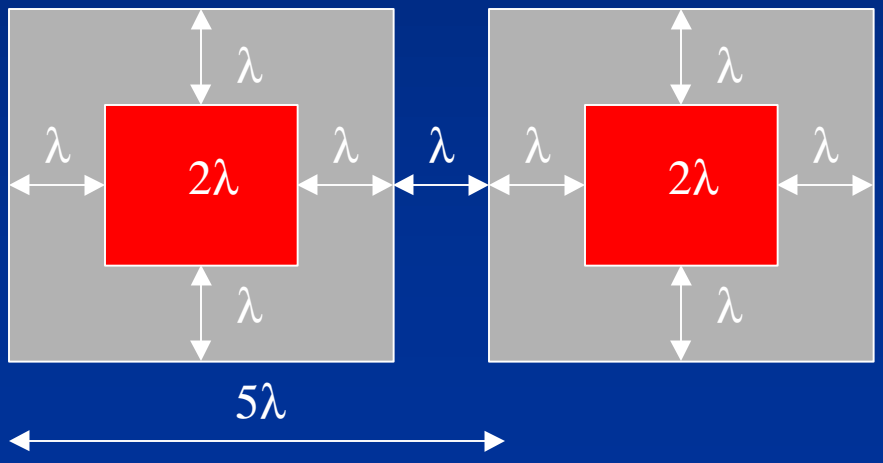
Scaling

Disclaimer

- **We refer to CMOS VLSI scaling**
- **Very simple models for μarchitects**
  - Gates
  - Wires

Scaling factor and technology

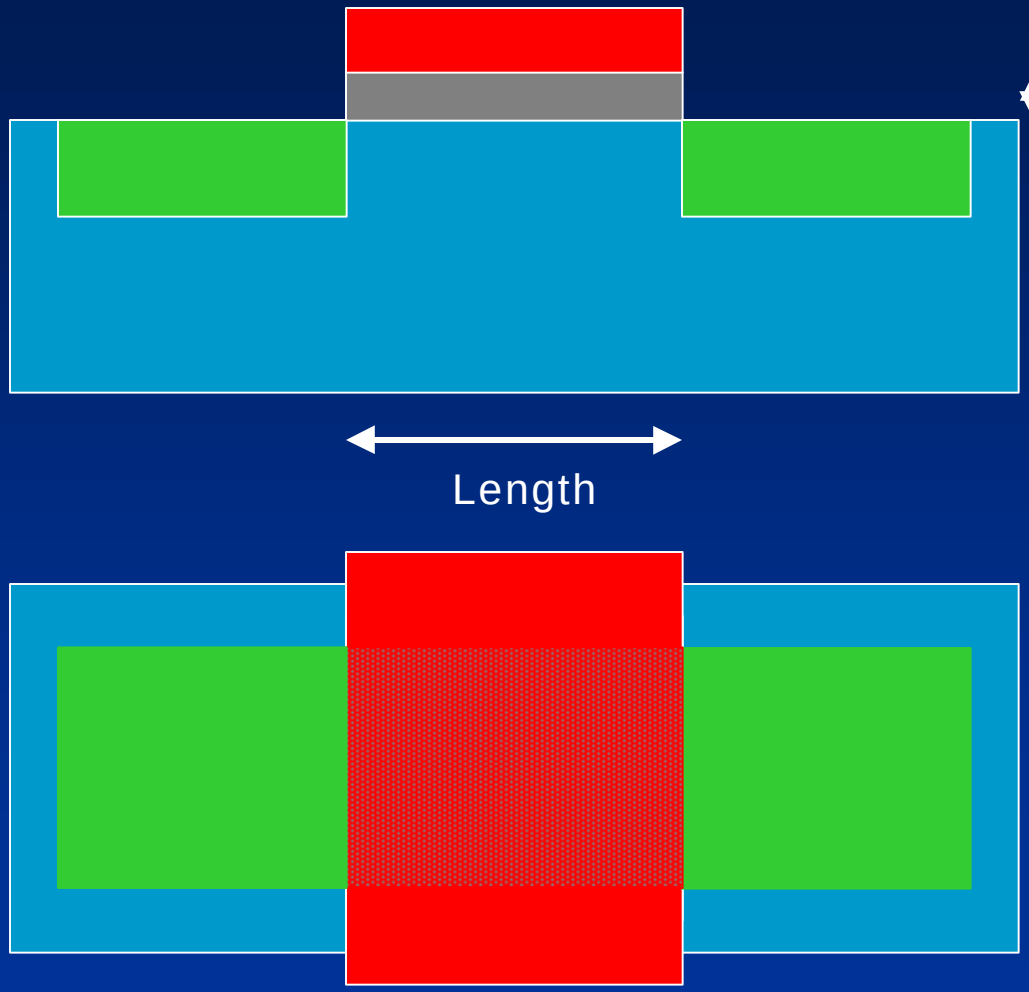
- **Depends on the litography and components tech.**
- **f: Minimum feature size**
  - f= 2 λ
  - λ is the mask registration variation
  - smallest device is 5 λ x 5 λ
  - S: scale= f_{old}/f_{new}



Gate scaling

How does the gate speed depend on technology?

Shape of transistors



R1= L/W

C1= W x L/Tgox

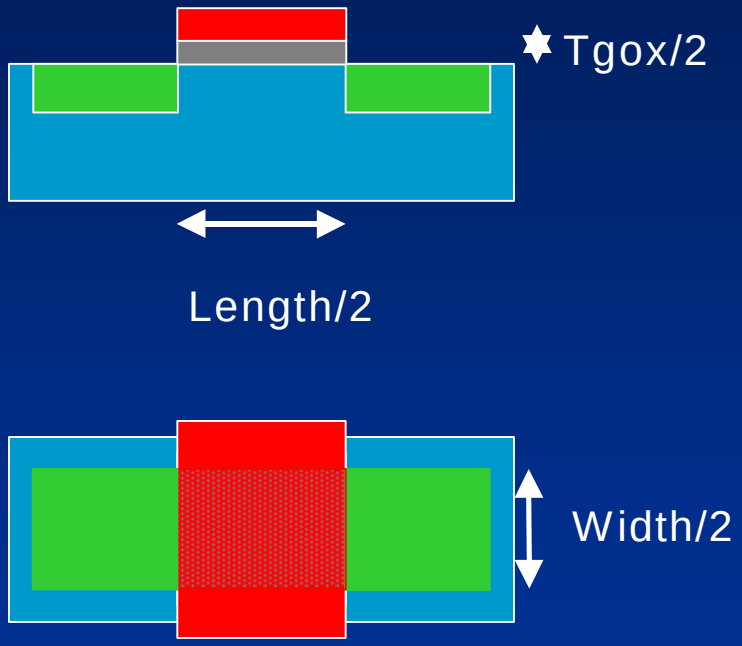
Delay= R1 x C1

Ideal scaling of MOS transistors

Dimensions (W, L, Tgox)	**1/S**
Voltages (Vdd, Vtn, Vtp)	**1/S**
Current per device (Ids)	**1/S**
Gate capacitance	**1/S**
Transistor on-resistance	**1**
Intrinsic gate delay	**1/S**
Area per device (A= WxL)	**$1/S^2$**
Power dissipation per gate (IxV)	**$1/S^2$**
Power-delay product per gate	**$1/S^3$**
Power-dissipation density (IxV/A)	**1**

Scaling transistors

$\alpha = 0.5 = ½$ (S= 2)



R2= (L/S)/(W/S)= R1

C2= (L/S x W/S) /(Tgox/S)= C1/S

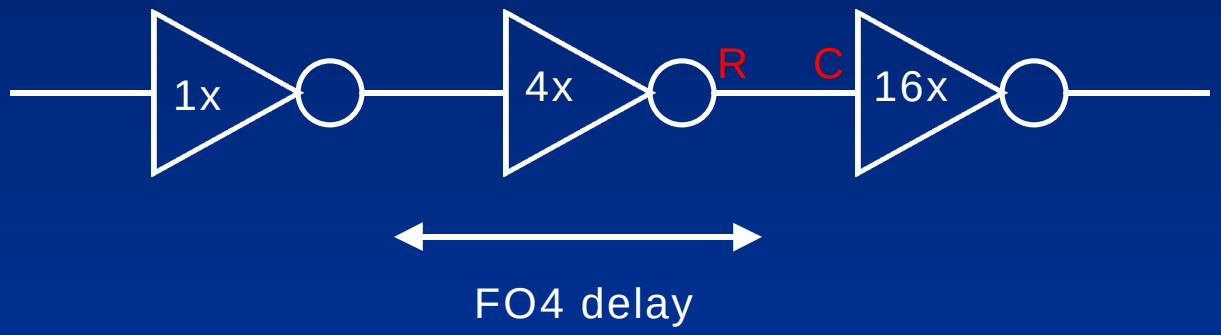
Delay= R2 x C2= R1 x C1/S

Gate delay (ideal scaling)

- **R remains constant**
- **C is divided by the scaling factor**
- **Delay= RC is divided by S when scaling by S**

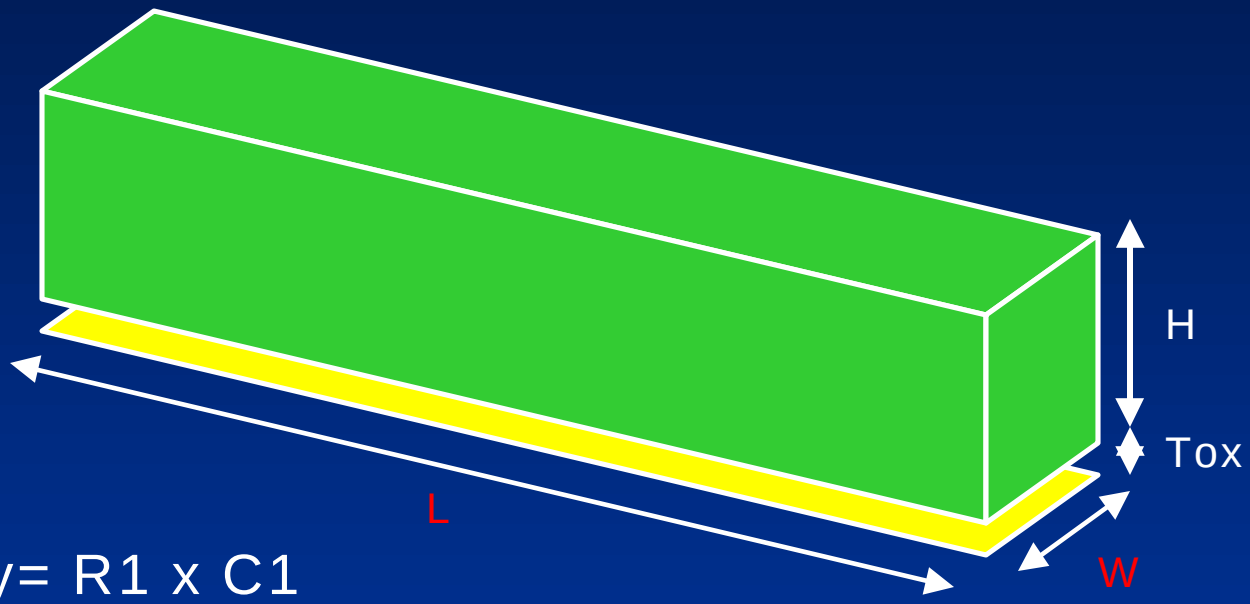
Fanout of 4 inverter metric

- **Delay of an inverter with $C_{out}/C_{in}= 4$**



Wire scaling

Shape of a wire



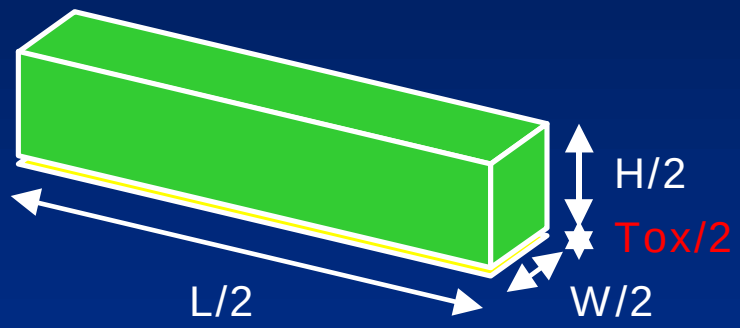
Delay= R1 x C1

R1= L/(WxH) R1/μm= ρ/(WxH)

C1= LxW C1/μm= εxW

Scaling of a wire (uniform)

$\alpha = 0.5 = ½$ (S= 2)



Delay= R2 x C2= R1 x C1

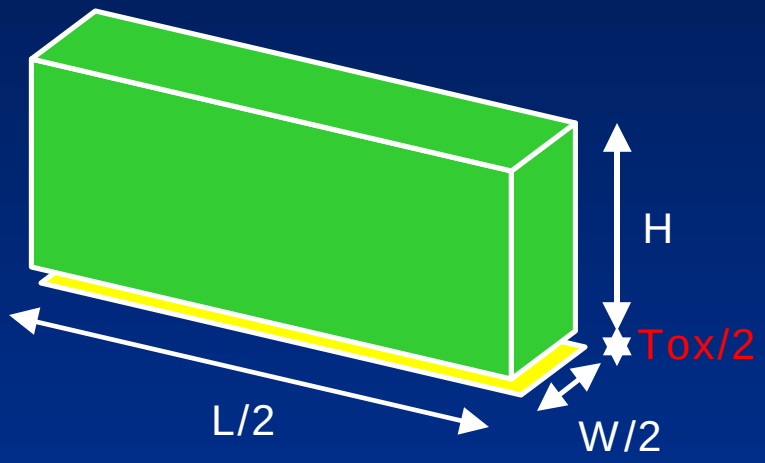
R2= R1xS R2/μm= R1xS^2

C2= C1/S C2/μm= C1

Scaling of a wire (non uniform)

Aspect ratio= 2

α = 0.5= ½ (S= 2)

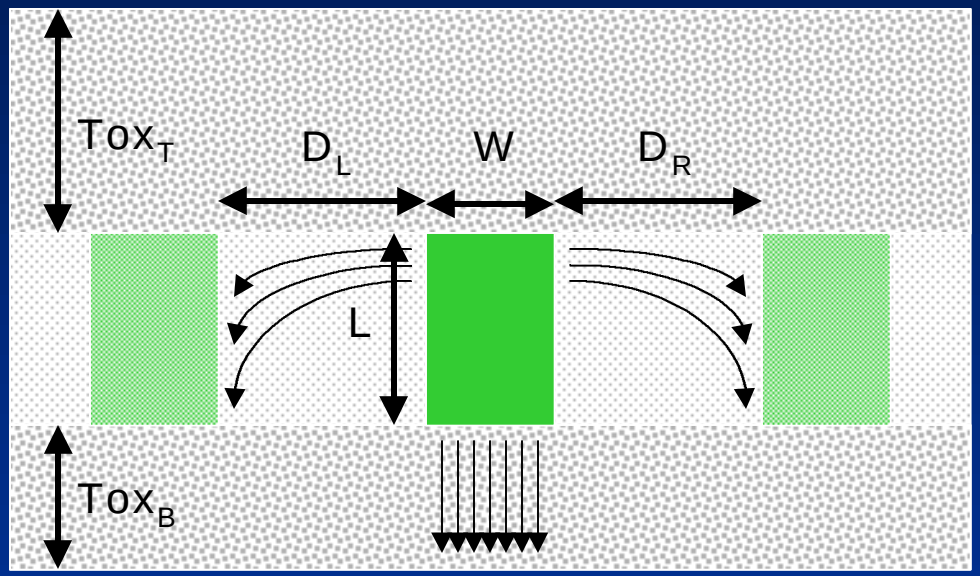


Delay= R3 x C3= R1 x C1/S

R3= R1 R3/µm= R1xS

C3= C1/S C3/µm= C1/S^2

Not so easy ...



Even worse

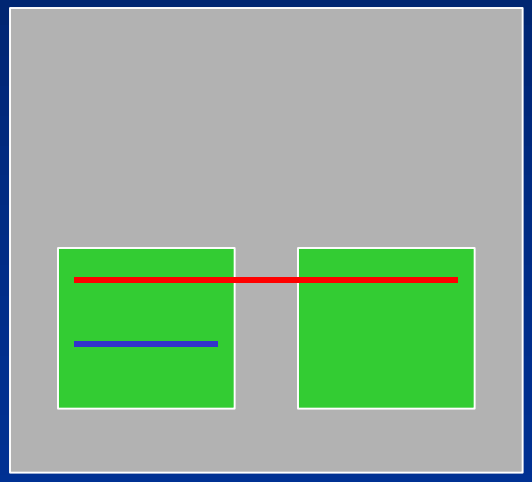
- **Three types of levels**
  - M1
    - Local interconect
    - Highest resistance, finest pitch
  - M2, M3, M4 (...)
    - Intra block
  - M5, M6, ...
    - Global wires
    - Thick coarse metal
- **When scaling introduces too thinner metal**
  - Create new top layer

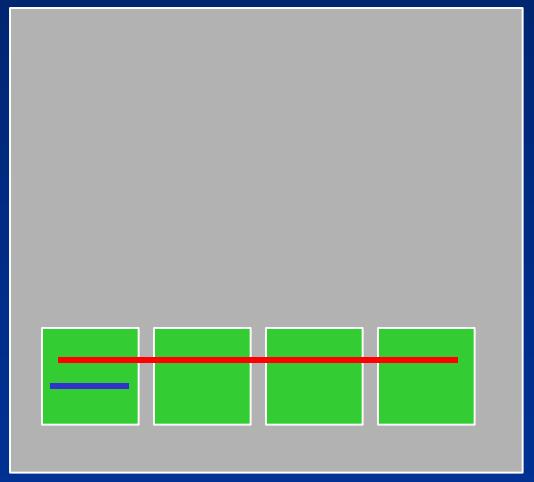
Performance of wires?

- **Measured relative to logic gates**
- **Divide speed of circuit by speed of FO4 inverter**
- **Many options and complex trade-offs**
- **Local wiring vs Global wiring**

Scaling of global wires

- **R gets worse with scaling**
- **C remains constant (+ or -)**
- **RC grows**



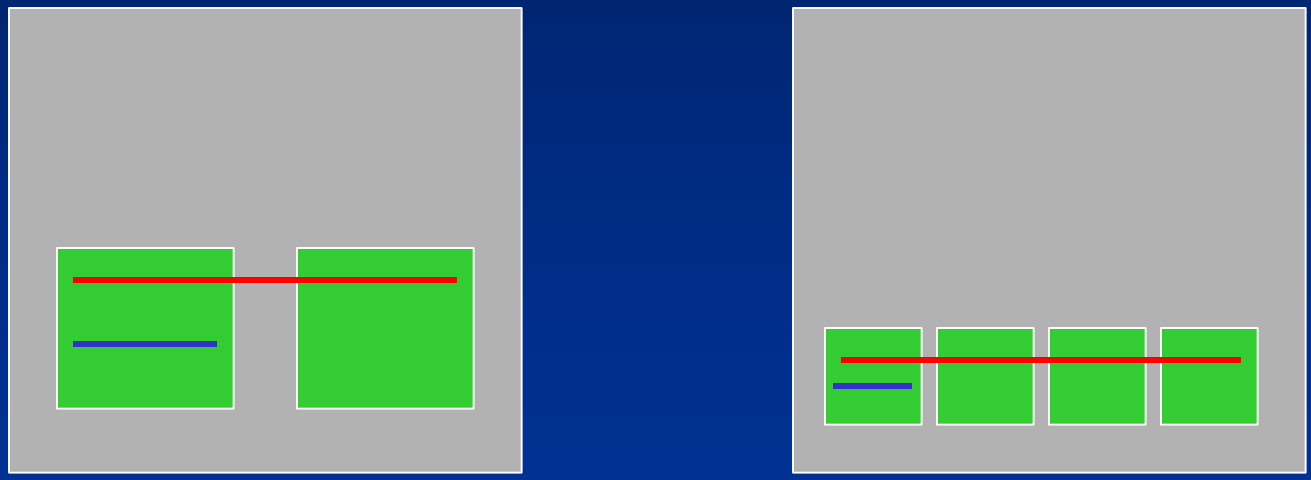


Solution to global wires

- **Use wider or thicker wires**
  - Higher metal levels are faster
- **Use repeaters**
  - Break wire into segments
  - Delay becomes linear with length
- **Repeaters introduce delay and complexity**

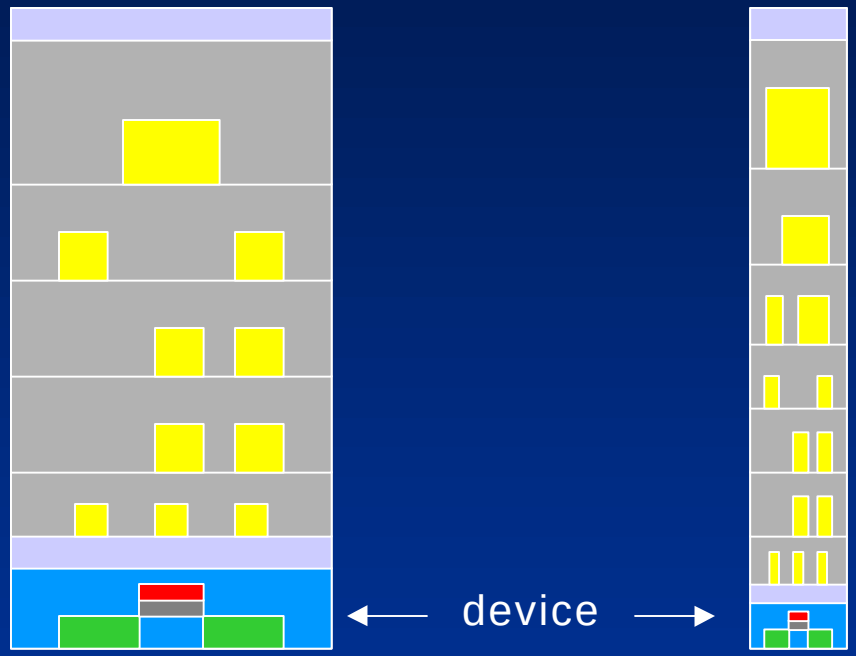
Scaling of local wires

- **R is constant (+ or -)**
- **C falls linearly**
- **RC is reduced like gate delay (+ or -)**



All together

Gates & Wires



Impact of scaling

From the point of view of an architect

Main topics

- **Delay**
- **Power**
- **Area**
- **μArchitecture**
- **Performance**
- **Complexity**

The equation

- **T= I*CPI*Tc**
  - When we scale we improve Tc (reduce it)
  - We also add more stages to further reduce Tc
  - But more stages increase CPI
    - load misses, branches, …
  - So we put better predictors and caches and ...
    - to compensate for the lost CPI due to more stages
- **Future**
  - some more Tc reduction due to technology
    - where is the limit?
  - no more reduction on number of FO4 per stage?
    - only new stages for wire delays???? where is the limit?
  - so … we better improve the µarchitecture...

The equation (II)

- **Now**
  - $T_1 = I \times CPI_1 \times Tc_1$
- **Usual improvements**
  - $T_2 = I \times CPI_1 * \alpha_1 * \beta_1 \times Tc_1 * \gamma * \delta$

    $\alpha_1 > 1$ (more stages because of $Tc_1 * \gamma * \delta$)

    $\beta_1 < 1$ (better μarchitecture)

    $\gamma < 1$ (better technology)

    $\delta < 1$ (reduced FO4 per stage)
- **Future**
  - $T_3 = I \times CPI1 * \alpha_2 * \beta_2 \times Tc_1 * \gamma$
    - $\alpha_2 > 1$ (more stages because of $Tc_1 * \delta$)

The equation (III)

- **Another improvement in Tc comes from improvements in circuit design**
  - dynamic vs static logic
  - latches with embedded logic
  - …

Impact on μarchitecture

- **When using global resources we run into wire limitations….**
- **Maintain number of stages**
  - Multilevel (to maintain number of
    - Caches
    - Predictors
    - Register files
  - Reduce structure size
- **Maintain structure size**
  - Increase pipe stages
- **Multibank**
- **Clustering**

More impact on μarchitecture

- **As geometries get smaller and stored charge gets smaller…**
  - soft errors (alpha particles, gamma radiation, ...) may happen frequently (we need fault tolerance)
- **Memory will be at 1000 cycles from processor**
  - and caches can not grow as we would like
  - and for some codes caches don't help at all
    - low temporal locality codes (like data bases)

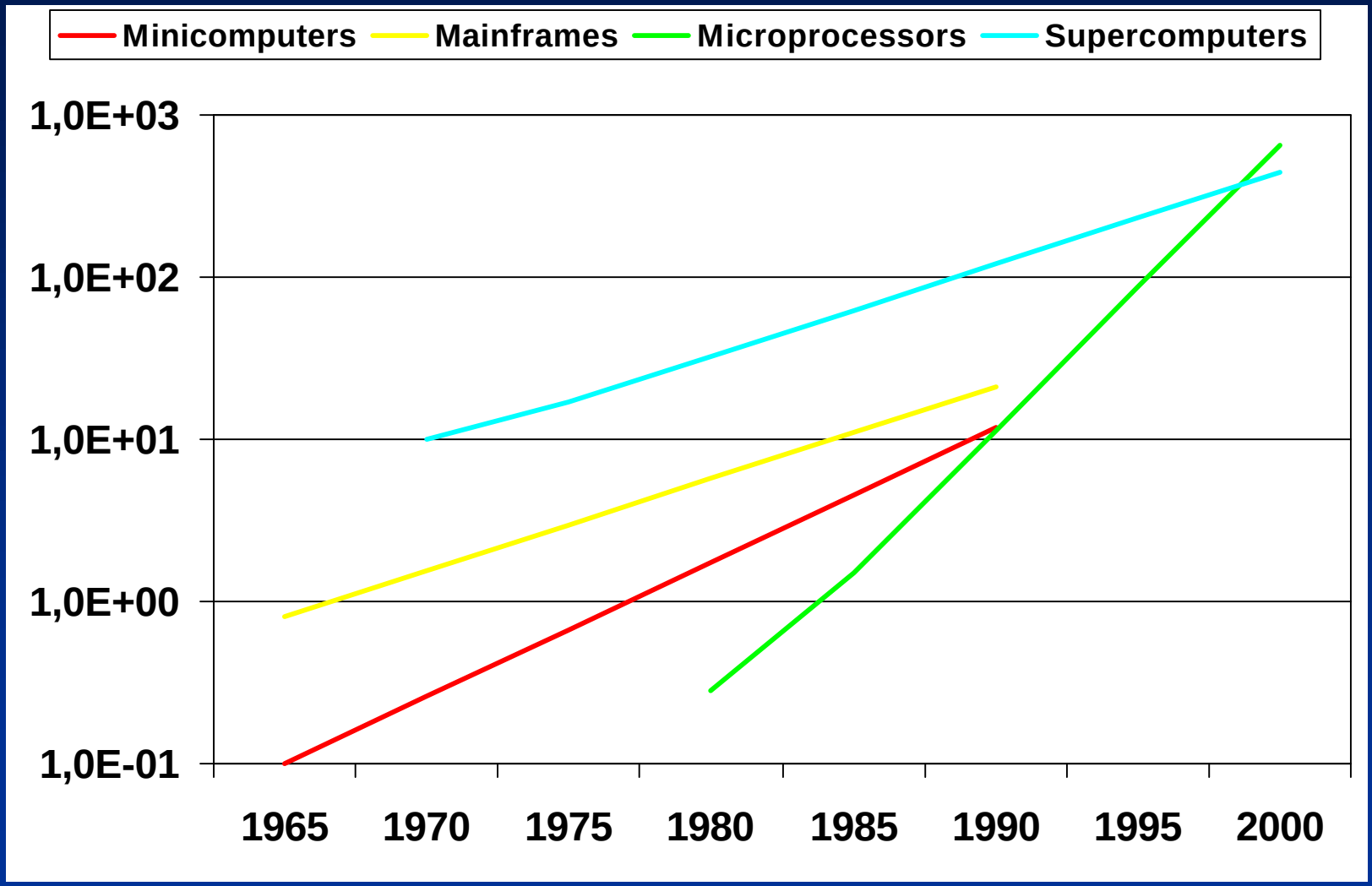
More architecture stuff

- **More ILP**
  - bigger Reg file
  - bigger IQ
  - more functional units
  - more ports
- **Longer distance, global wires, slow …**
  - amount of state that can be reached in 1 clock

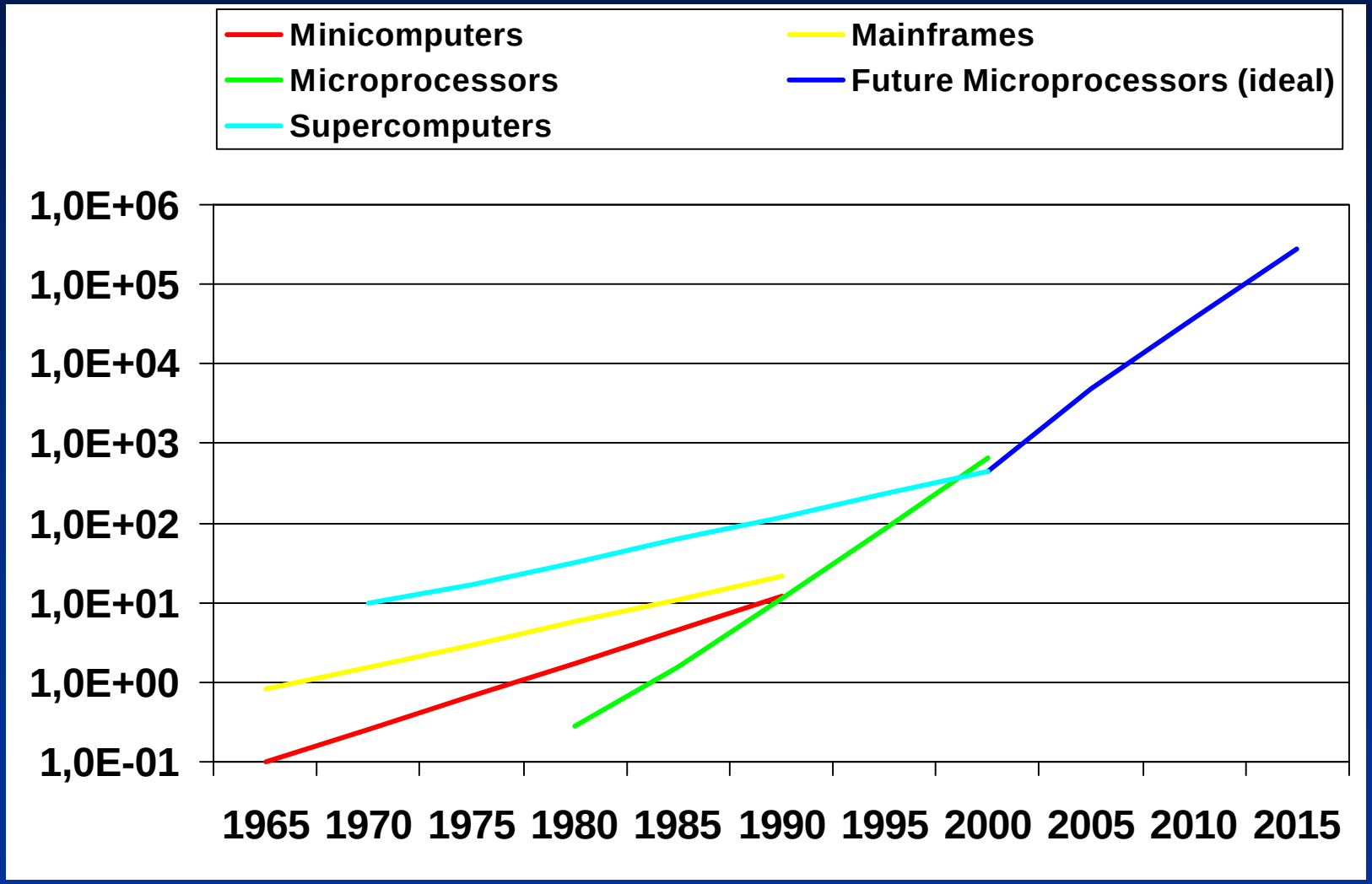
It's a 2D world

- **Perimeter vs Area**
  - as time goes by there are more transitors per pin
    - how do you communicate them?
    - how do you debug them?
  - avg. distance between communicating transistors
    - try to get as many communications as possible local
  - similar to living things and 3D (area vs volume)
- **At some point bigger is not better**
  - the extra transistors do not compensate
    - because of the communication latency

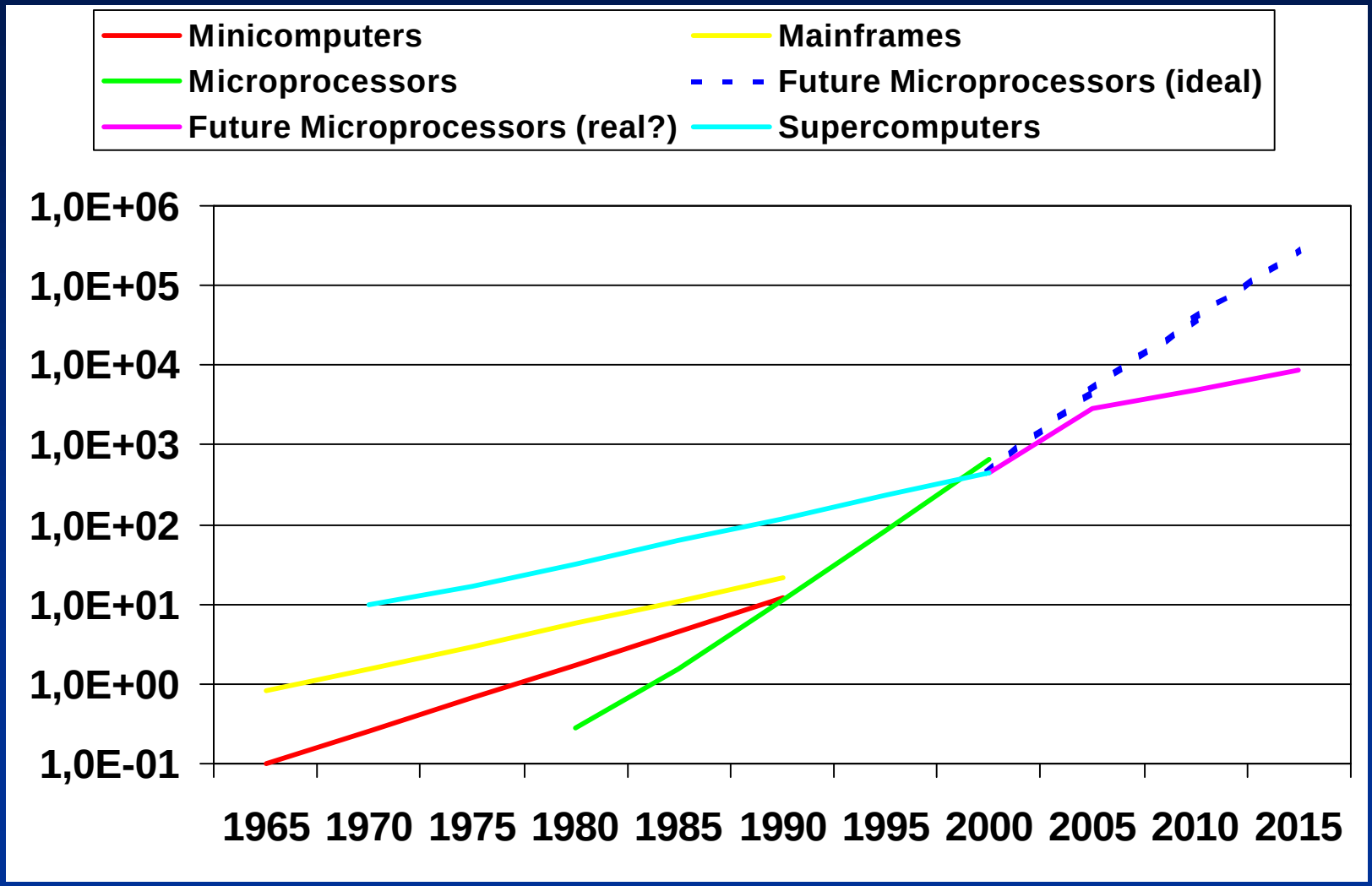
Performance trends



Ideal future



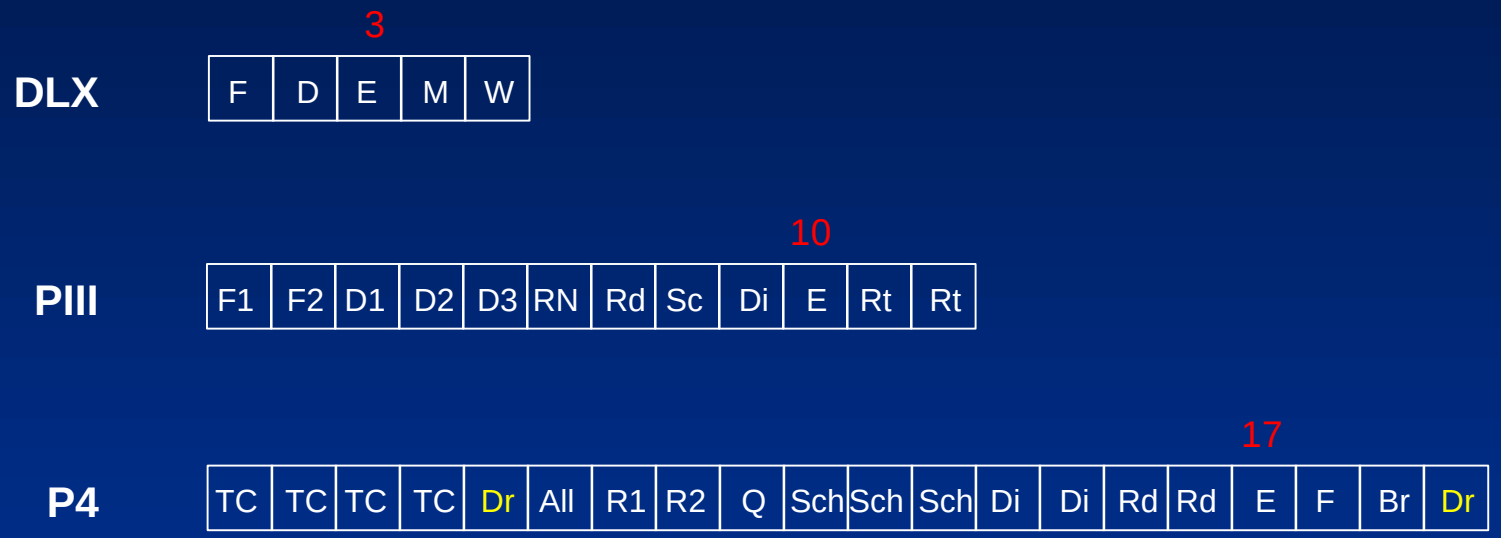
Real future?


Minicomputers
Mainframes
Microprocessors
Future Microprocessors (ideal)
Future Microprocessors (real?)
Supercomputers
1,0E+06
1,0E+05
1,0E+04
1,0E+03
1,0E+02
1,0E+01
1,0E+00
1,0E-01
1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015

Cicle time in FO4

- **Lots of performance from reduced FO4s per stage**
  - Used to be 100 FO4
  - Now close to 16 FO4
  - Difficult below (8 FO4)
    - Fastest 64 bit add 5.5 F04
    - Add latch overhead bypass delay and so on ...
  - SIA aggresive predictions assume 6 FO4

X86 Pipelines



Ppro, PII, PIII share the basic same core

Impact on Complexity

- **Size of design**
- **Size of teams**
- **Worse yield**
- **Cost is growing exponentially**
  - design+development cycle from 1 to 6+ years
  - time to market -> miss the market
  - fab costs
- **Intellectual complexity**
  - number of transistors per person?
  - Who understands the whole design?
  - Number (and complexity) of bugs
  - lack of tools

Bibliography

- Mark Horowitz (ISCA2K panel)
- Jim Smith (ISCA2K panel)

- Clock Rate vs IPC: The End of the Road for Conventional Microarchitectures,
  Agarwal et Al., ISCA2k

- Circuits, Interconnections and Packagin for VLSI,
  H.B. Bakoglu, 1990

- Computer Technology and Architecture: An Evolving Interaction,
  John L. Hennessy and Norman P. Jouppi,
  Sept. 1991, Computer