Life in the Synchronous Lane

Thoughts about Engineering Methodologies and CAD

Avi Kolodny

Intel Corporation, Haifa, Israel

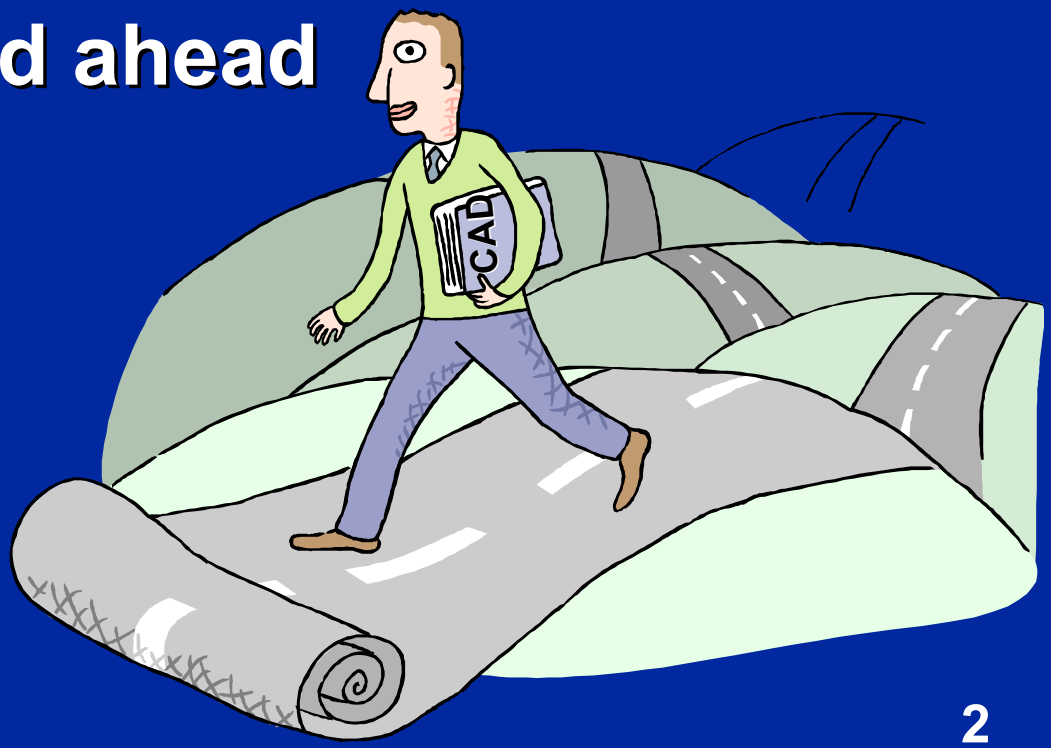
& Technion - I.I.T.

intel®

Async 2000

This talk is about a journey

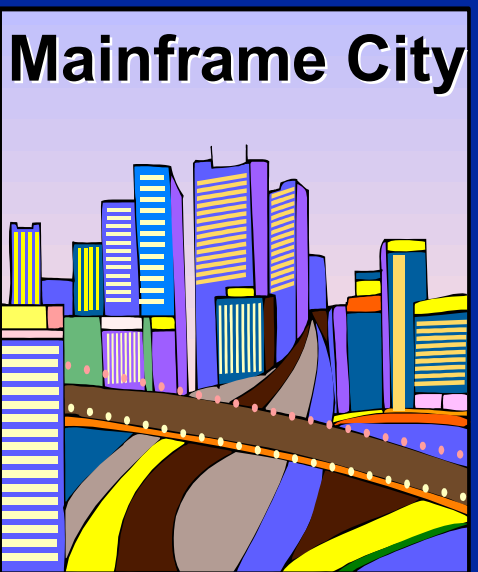
- Changes in design methods at Intel
  - Over the years
- Insights about CAD's role along the way
- Views of the road ahead



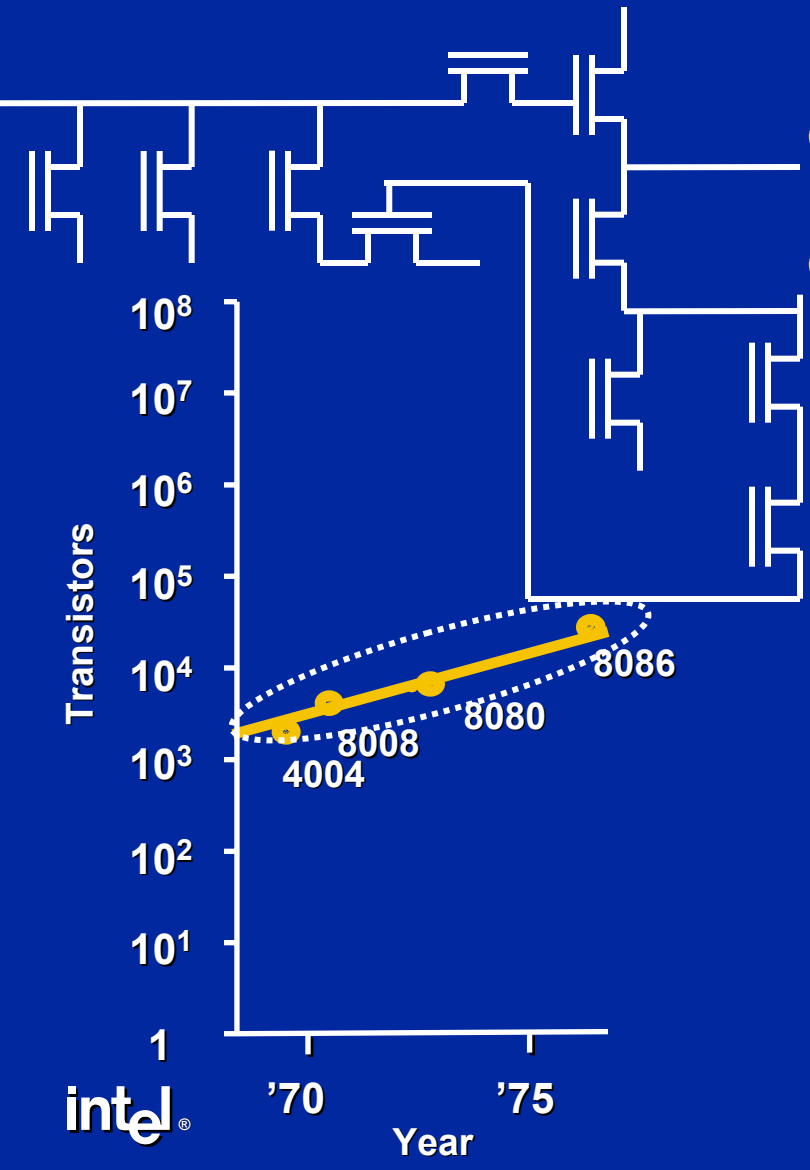
intel

How the journey started

- VLSI was driven by Silicon technologists
- They were not experts in logic design theory.
- Design had very few rules



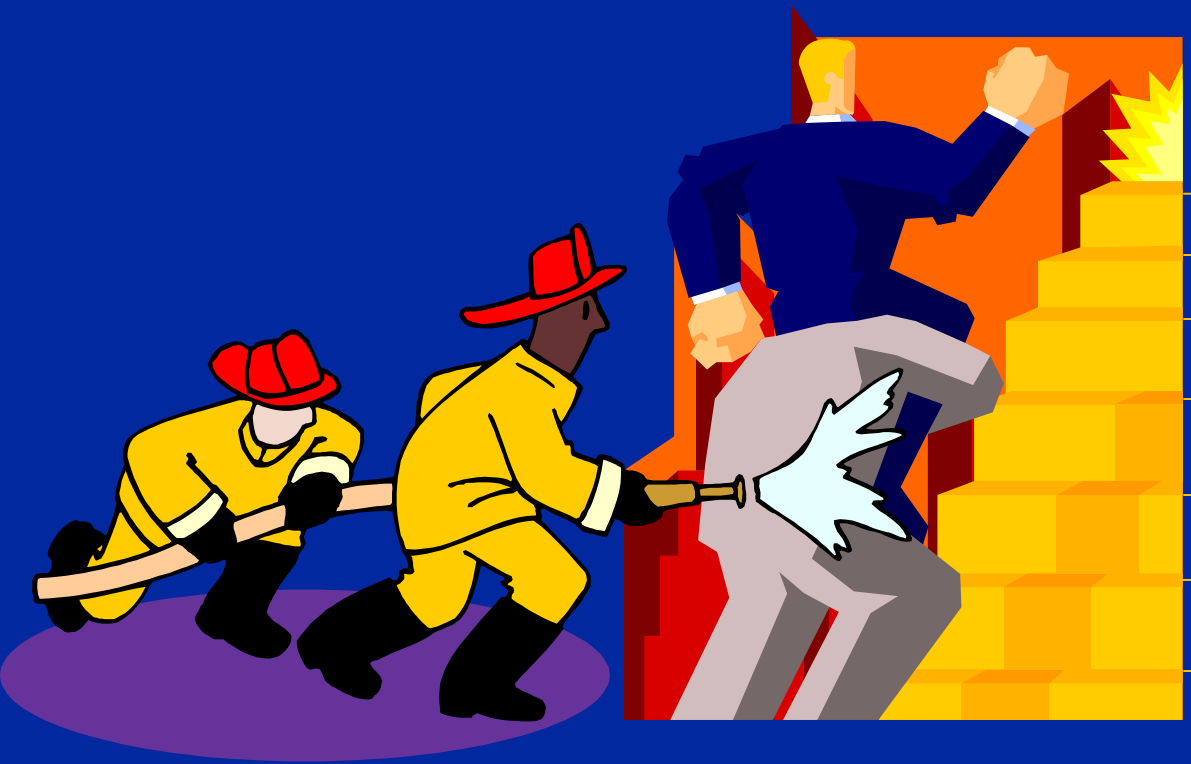
Design in the old days



- Transistor-level circuits
- Free mix of:
  - Ratioed-logic
  - Wired logic (contention based)
  - Race-based logic
  - Level / edge signaling
  - Dynamic storage nodes
  - Pass transistors
  - Clock input to logic
  - Loops in combinational logic

CAD Tools of the good old days

Fighting the Fire" step by step"



...What Next?...

Schematic editor

Layout editor

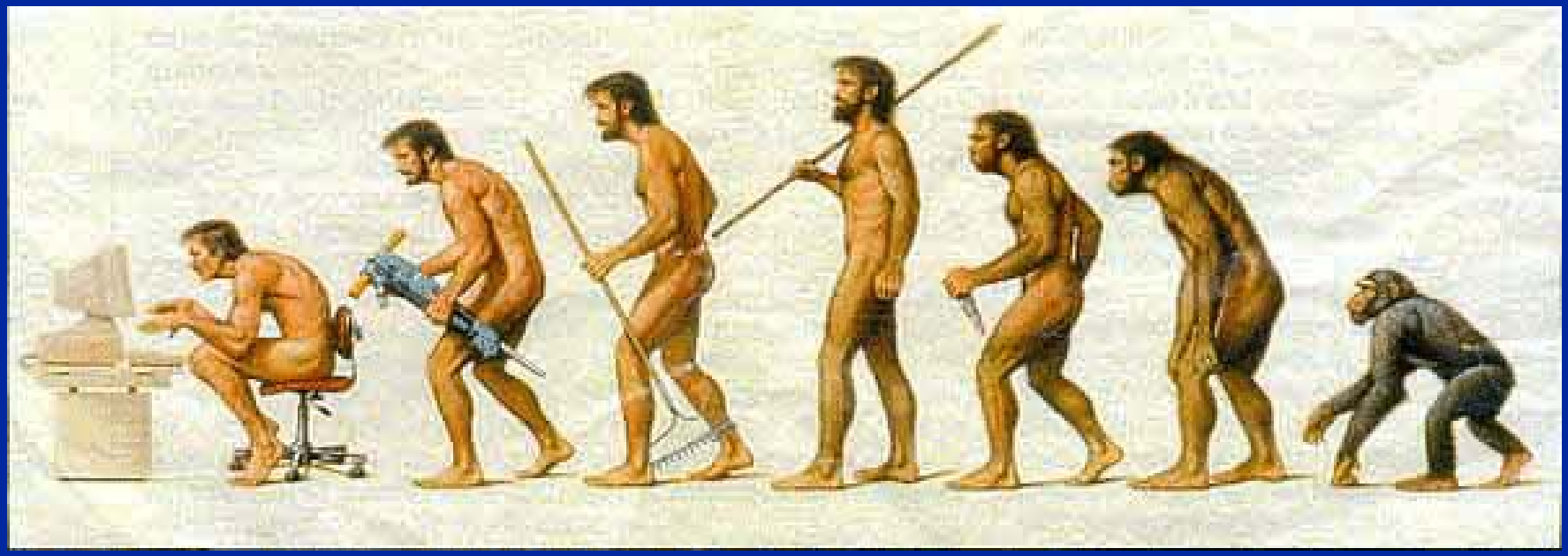
Design-Rule Checker

Circuit Simulator

Layout digitizer

intel

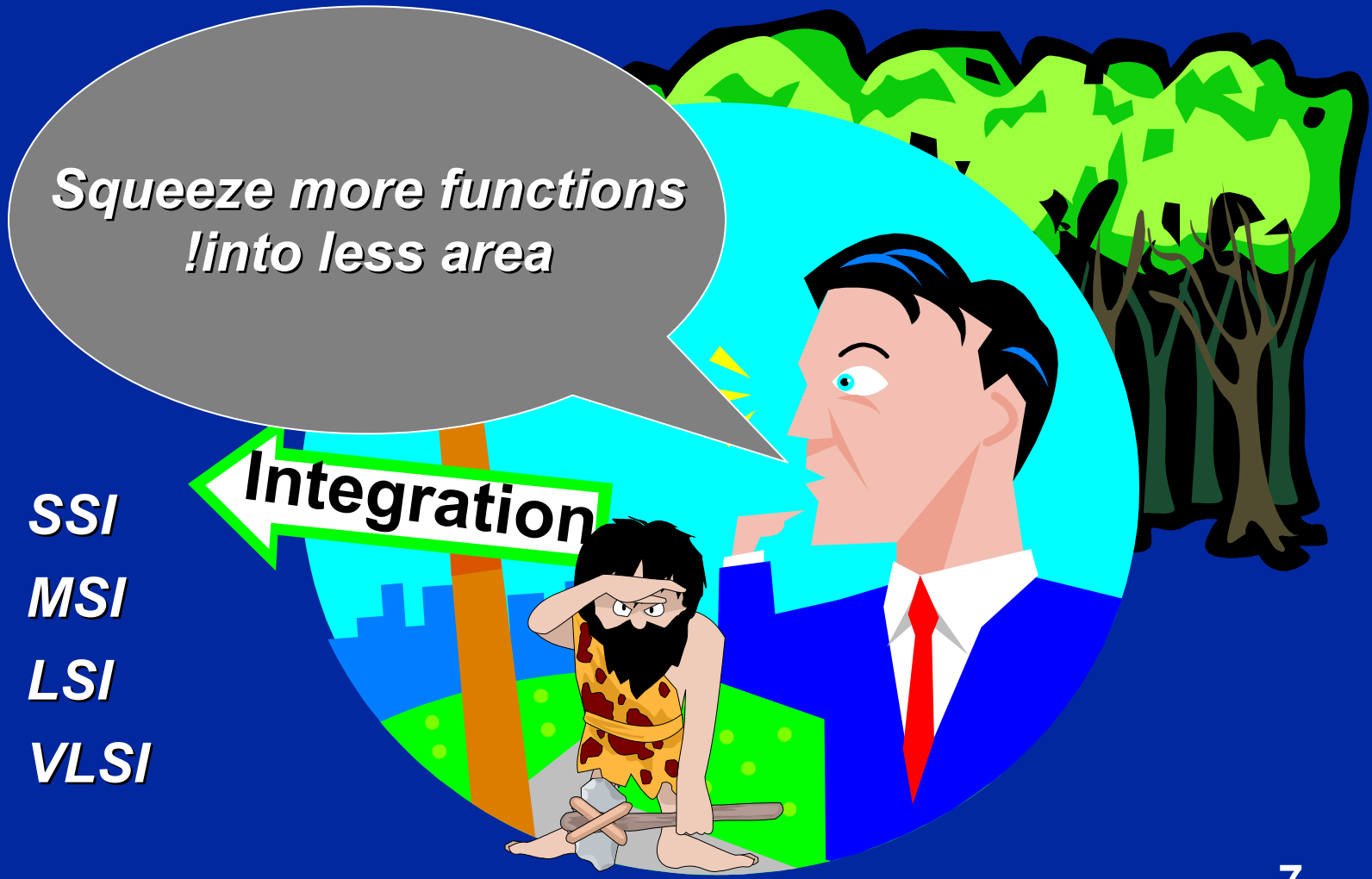
CAD is not planned.
It develops by Evolution



New CAD mutations emerge
to solve the painful problem of the era

intel®

The goal was: High Integration



SSI

MSI

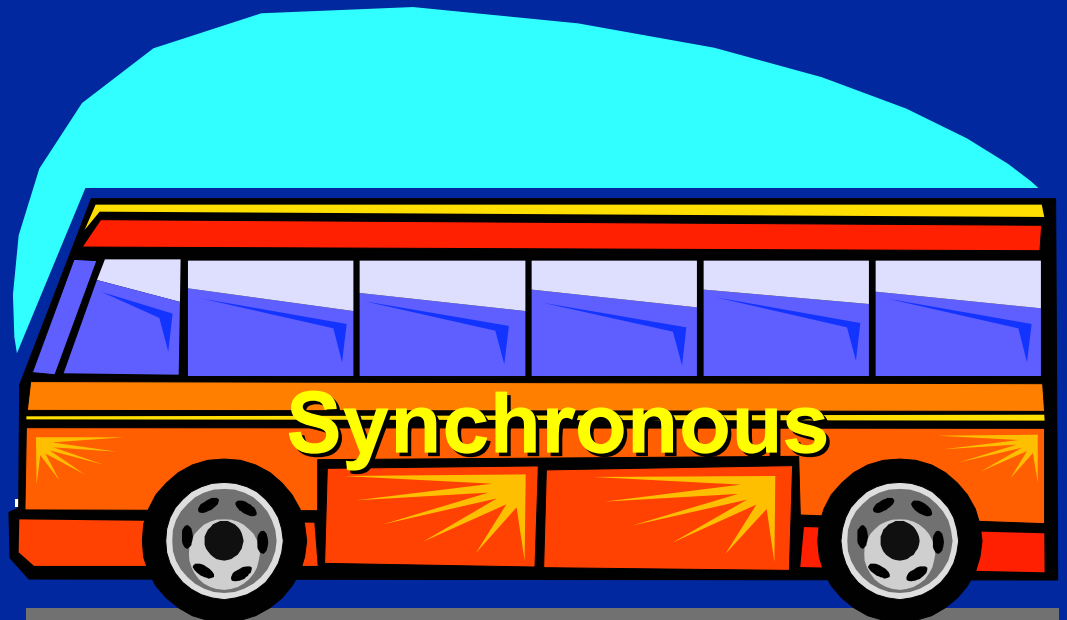
LSI

VLSI

intel

Which design style to use?

- Synchronous design was simpler for doing high integration

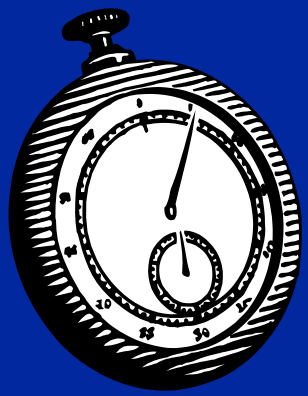


Asynchronous

Why synchronous design for hi-integration?

- "Time is assumed to come in discrete steps…..
  By providing a central 'clock' source …
  it is possible to organize even asynchronous components so that they act in the discrete time steps of a synchronous machine"
  [E.F. Moore, 1956]

Time as sequence (discrete)

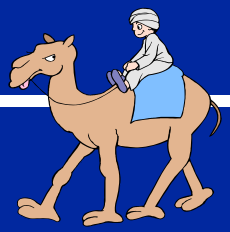


Time as duration (continuous)

intel

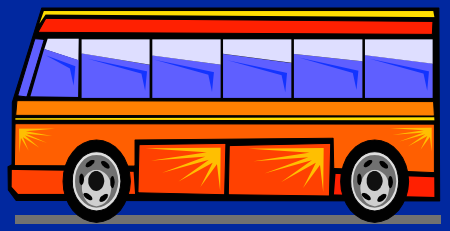
We had plenty of time...

- Speed was not an issue in product specs…..
  - Virtually no design for speed
  - The first PC ‘came out’ at 4.77 MHz
- So time was a free resource, used mainly to organize sequencing
  - A CAD tool limitation actually helped enforce synchronous design



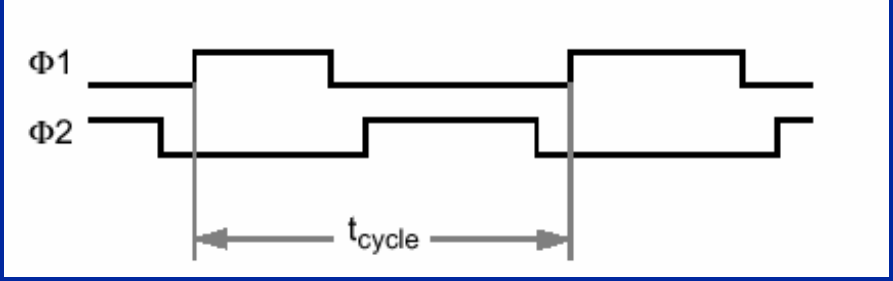
Async

Sync



The next CAD challenge: Logic verification

- Logic errors were too painful….
- Circuits were too large for SPICE
- An event-driven logic simulator was tried...
- It was too cumbersome
- Engineers wrote RTL models in a Pascal-like language
- The RTL simulator was born
- It could handle <u>synchronous design only</u>




```
If  (Phi1='1')  then
  Begin
    e :=  a AND b;
    f := c AND d;
     g := e OR f;
    x  := NOT g;
    y:= g  AND z;
         ……
     ……..
  End;
If  (Phi2 = '1')  then
  Begin
      e:=y;
      …..
  End.
```

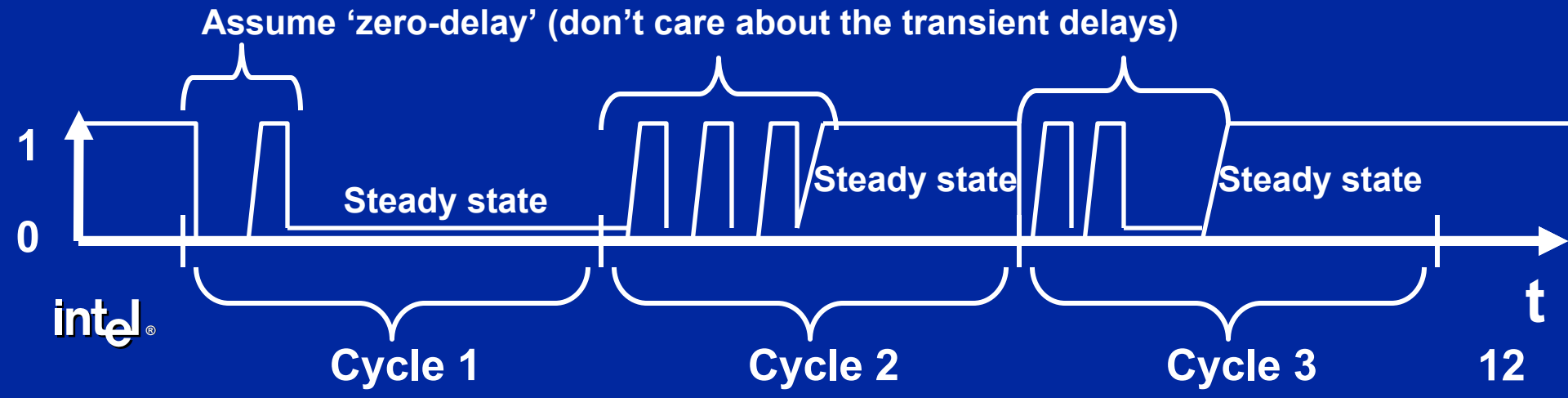
RTL modeling was a clever methodology !

- It ‘divorced’ functional behavior from timing
  - Functionality and timing could be verified separately
- But…. timing verification was ignored!

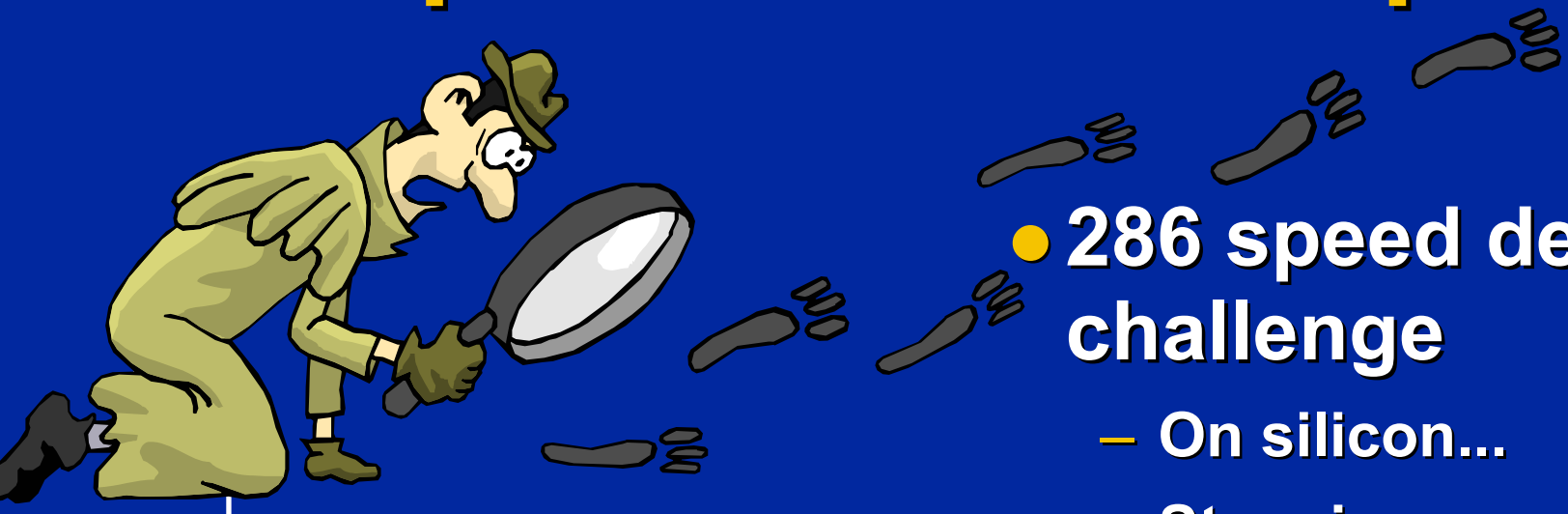
RTL model assumptions:

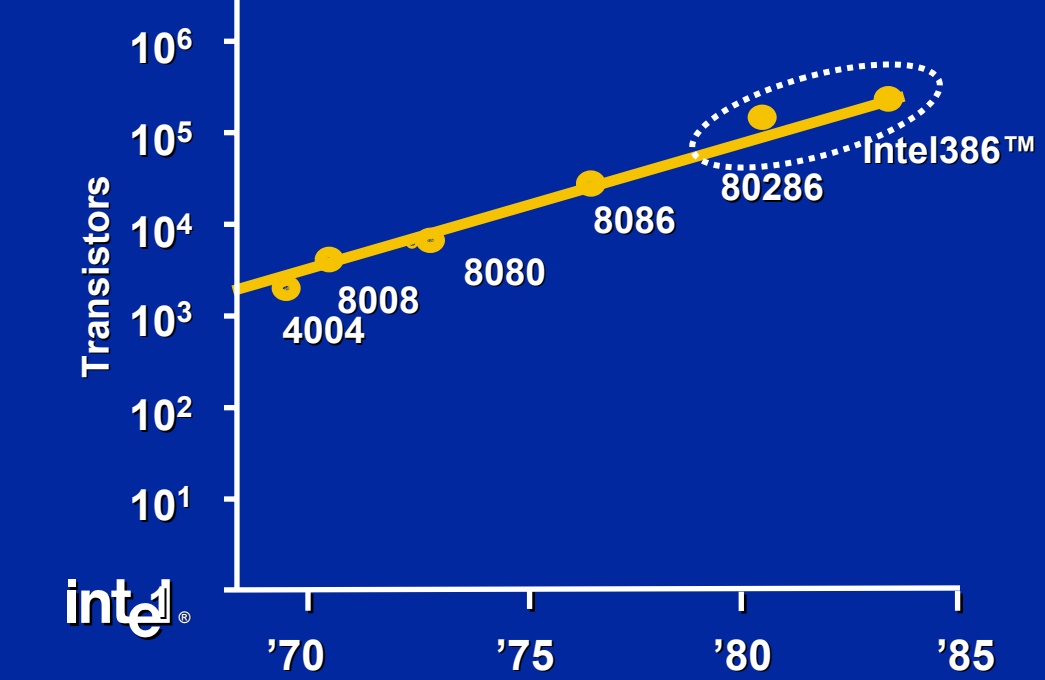
1) In each clock cycle, new values propagate until a steady-state is captured in registers.

2) Someone has guaranteed that the cycle-time is long enough to reach a steady state.



Then speed became important

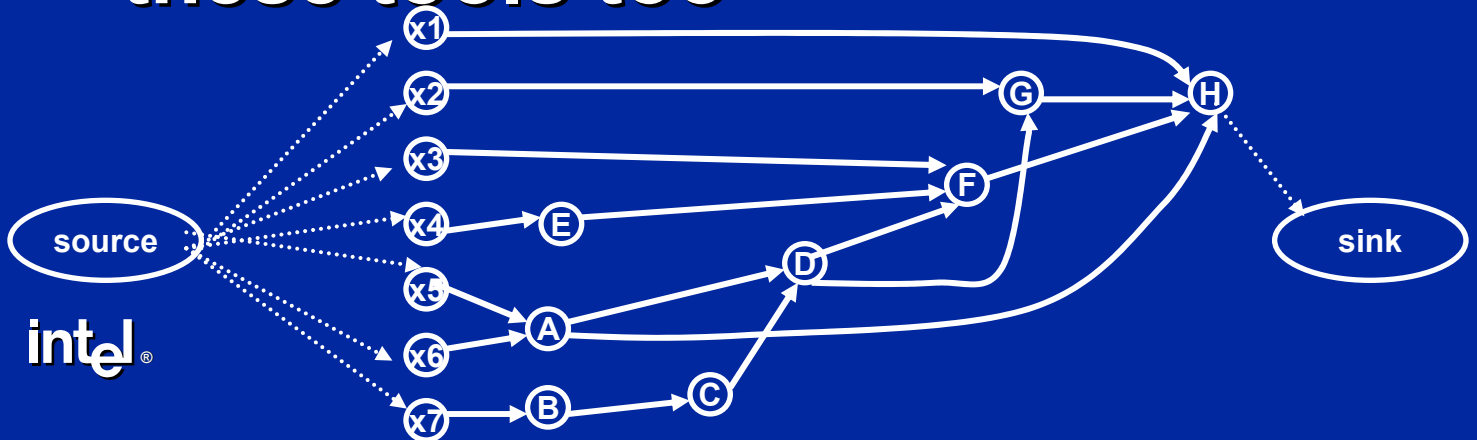




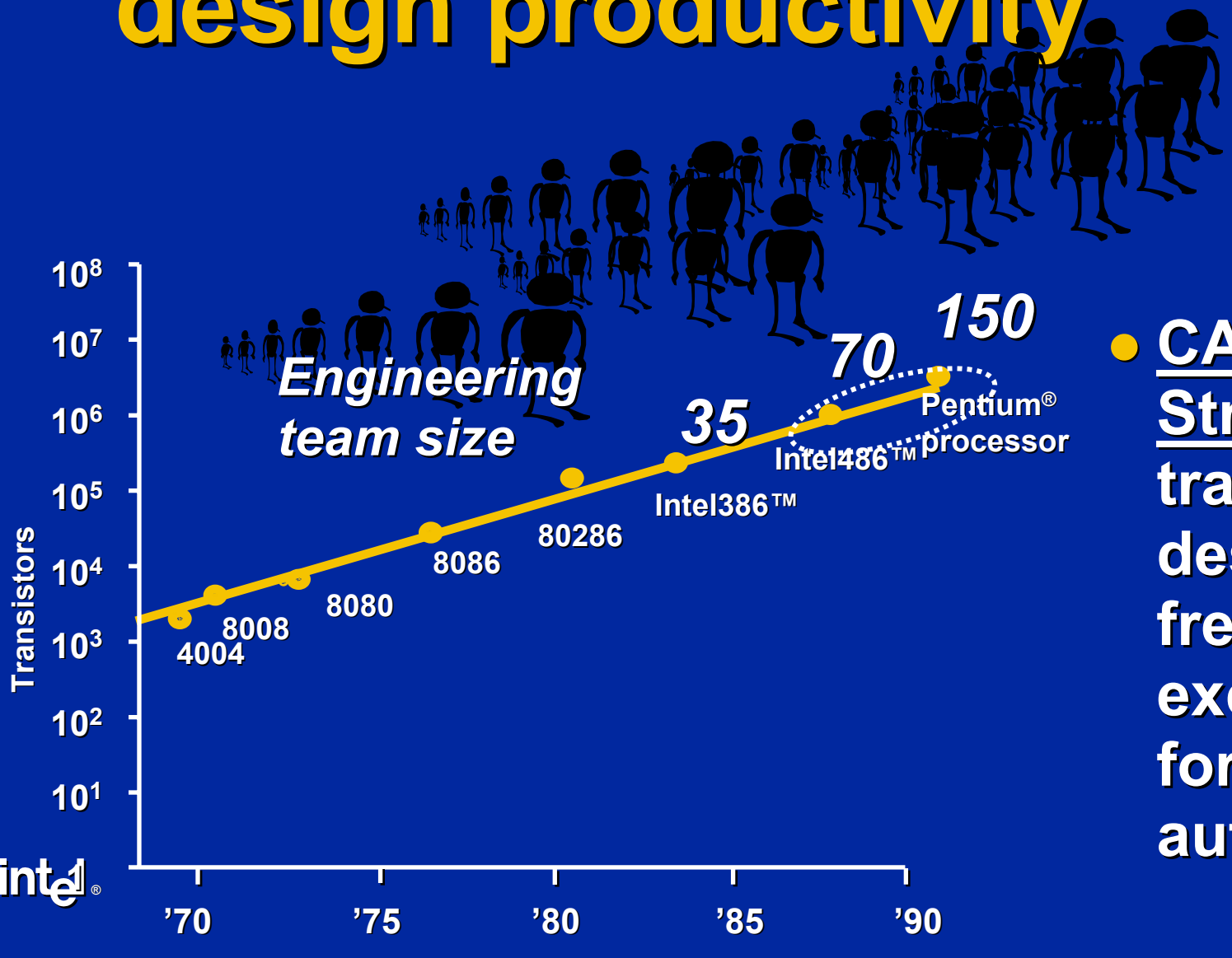
- 286 speed debug challenge
  - On silicon...
  - Steppings
  - Clock-stretcher testing
  - Pain & cost
- Duration of the clock cycle became a precious resource!

Design-for-speed Impact on methods & tools

- Time borrowing as a design technique (transparent latches)
- New static tools :
  - Critical Path Finder
  - Delay Analyzer
- Synchronous methodology enforced by these tools too



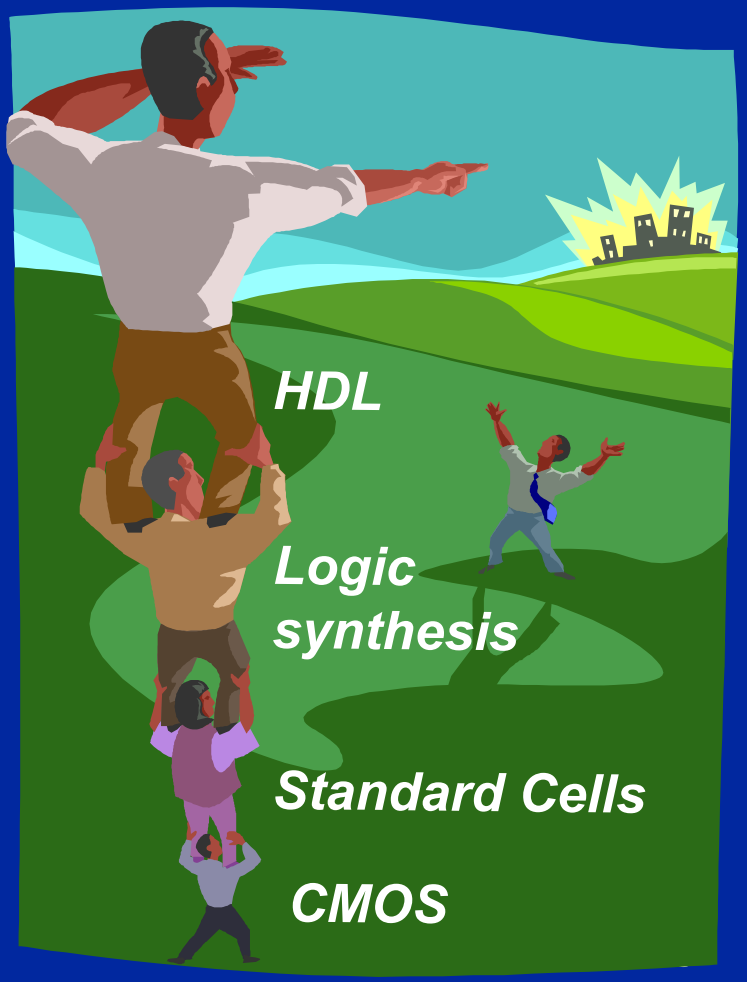
The next challenge: logic design productivity



- **CAD Strategy:** tradeoff design freedom in exchange for automation

The synthesis CAD solution

- Synergetic action of
  - HDL
  - Logic synthesis
  - Cell library + Place & Route tools
- A “package deal”
- Migration to Single-phase clock and Master-Slave Flip-Flops
- Current ASIC methodology

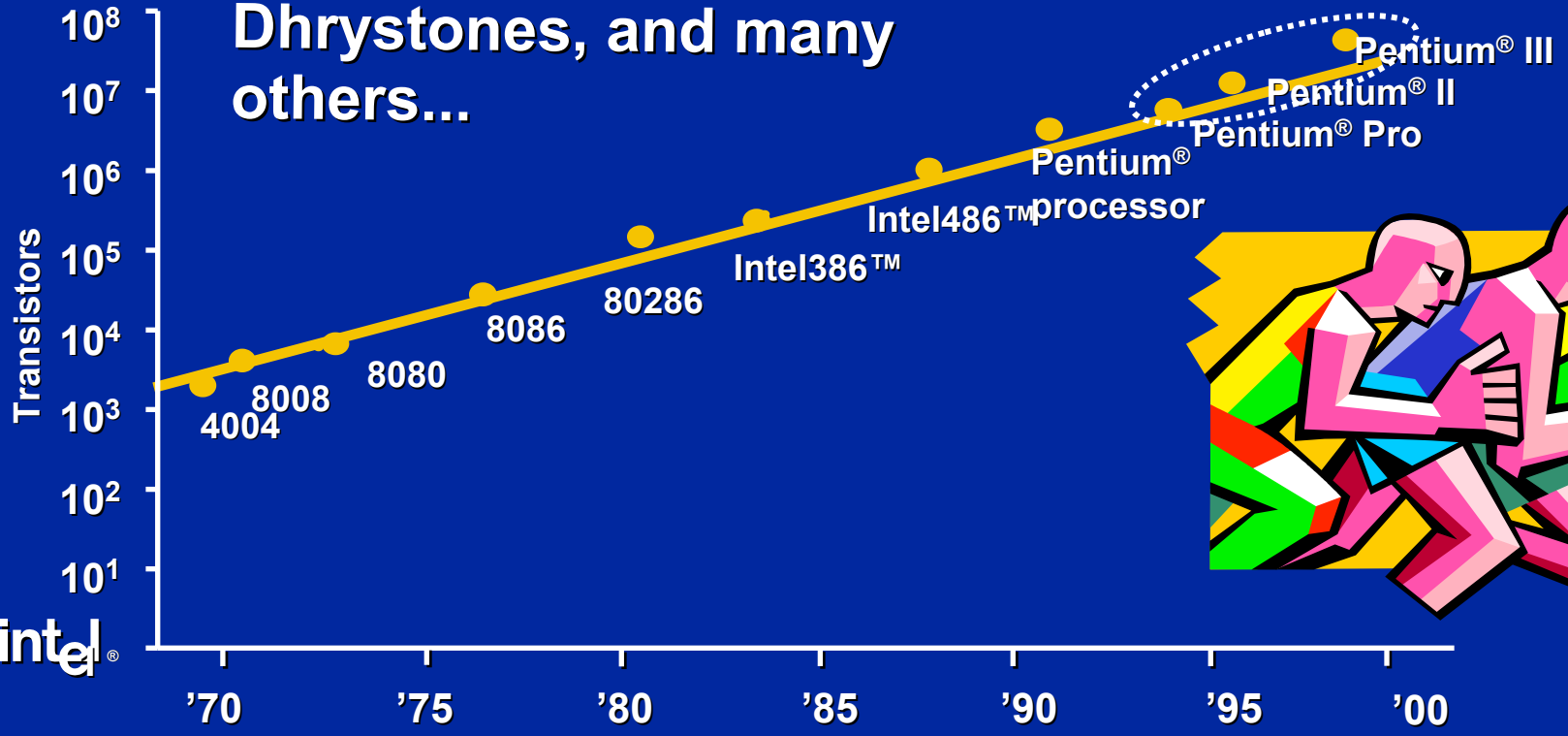


intel®

Then "MHz" became a marketing buzzword....

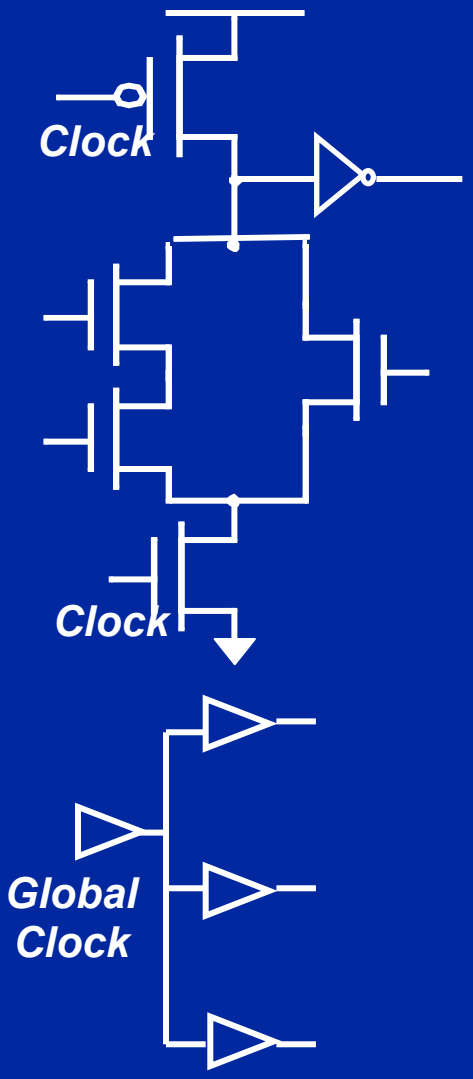
Year	Tech [micron]	Clock [MHz]
1992	0.8	66
1995	0.6	150
1997	0.35	266
1999	0.25	550
2000	0.18	1000

- Unlike MIPS, SPECS, Dhrystones, and many others...



Design for high clock rate

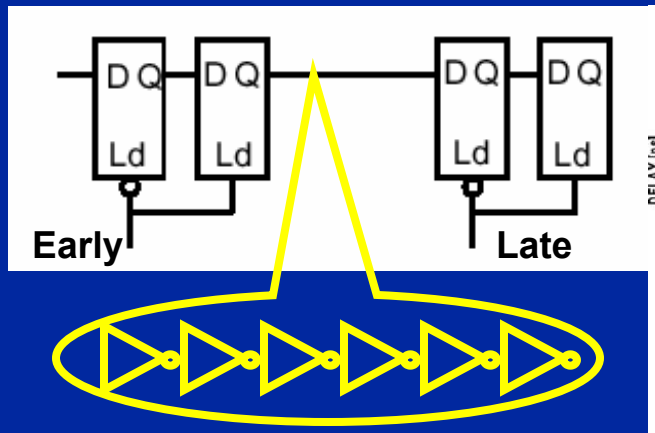
- Internal clock doubling
- Super-pipelined microarchitecture
- Domino logic
- Buffered clock-trees
- CAD: timing-driven everything
- *… new problems came* up

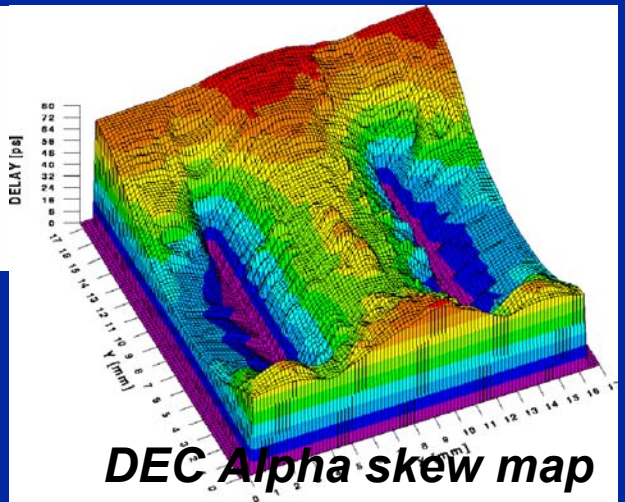


intel

Clock-skew problems

- Started systematic checking of minimum-delay violations
- Automatic insertions of buffers for delay padding
- 30 to 50% of gates are inverters!

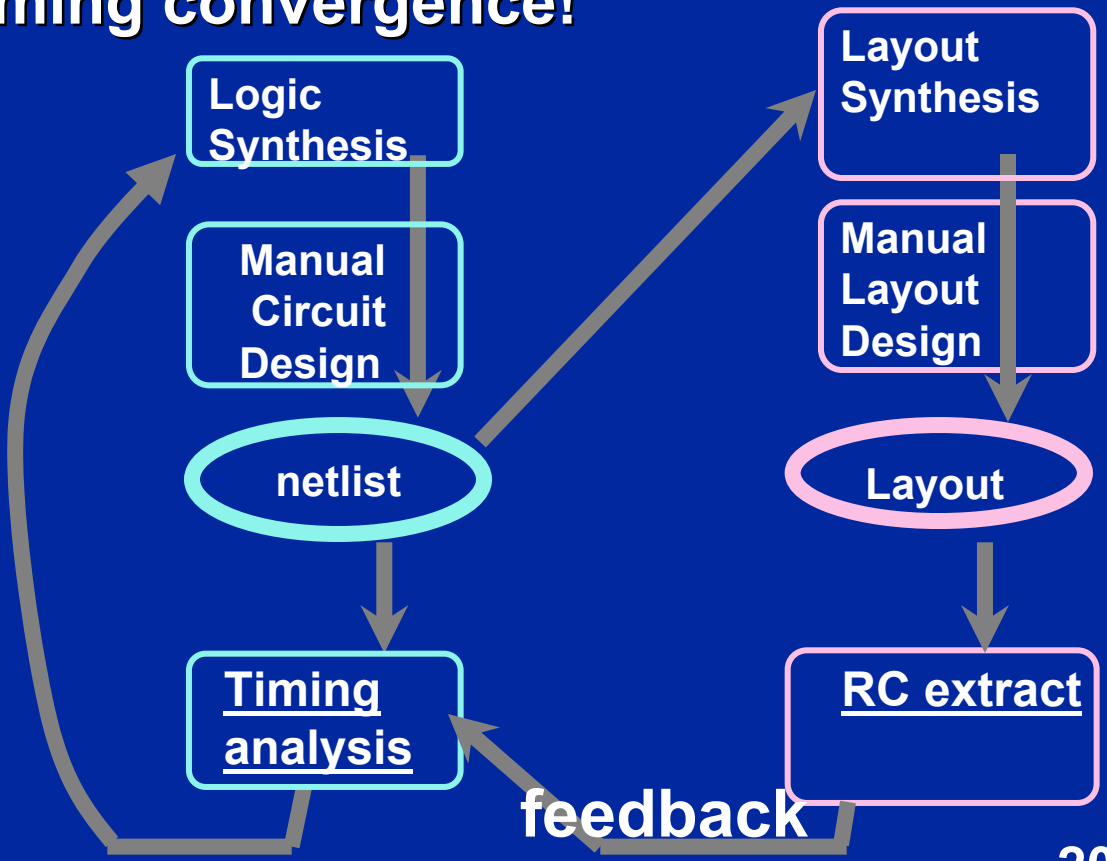




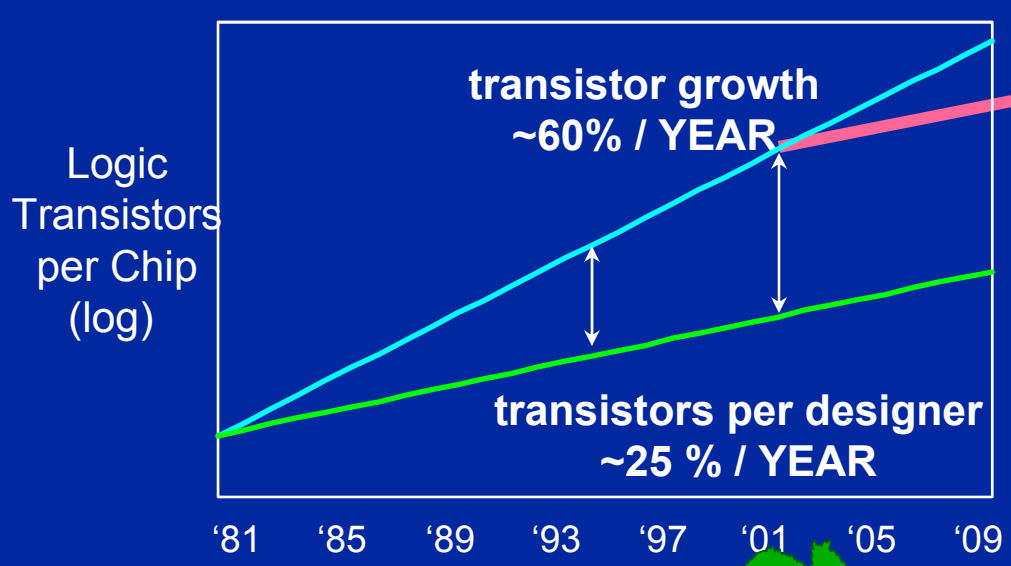
DEC Alpha skew map

Timing Convergence Problem

- Timing analysis with real layout RC is in a feedback-loop of the design flow
- Synthesis produces totally different solutions each time
- It's difficult to reach timing convergence!
- New approaches to solve the problem:
  - Wire planning with time budgeting
  - Layout-driven synthesis
  - Synthesis-driven layout

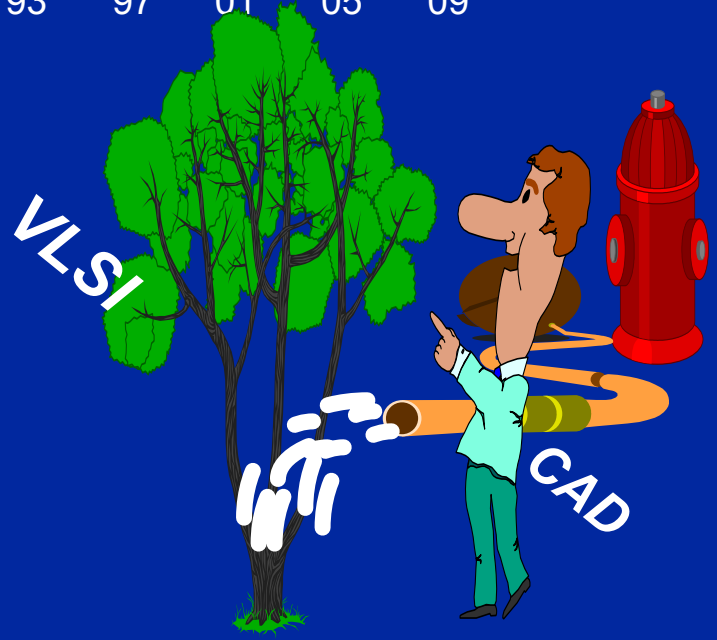


Design productivity gap keeps growing

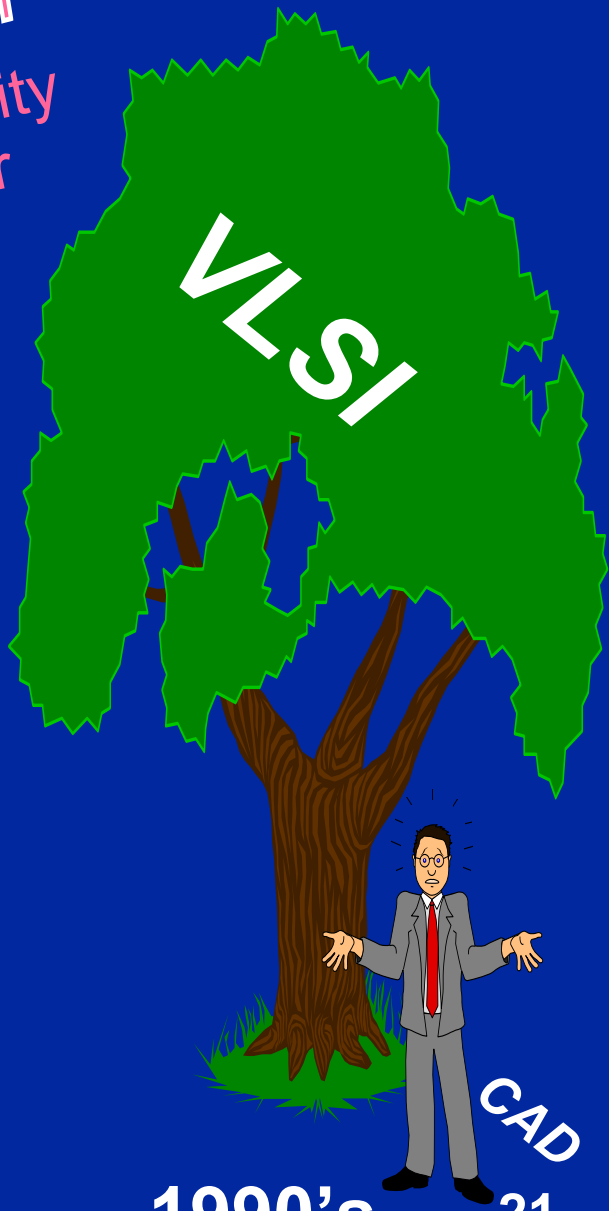


Complexity Limiter

1970's



1980's



1990's

Now the road starts climbing into big mountains…

- **Deep-Sub-Micron problems:**
  - **Wiring dominance:**
    - **R, L, C, delay, power, noise**
  - **System-On-Chip issues:**
    - **hierarchy, re-use**

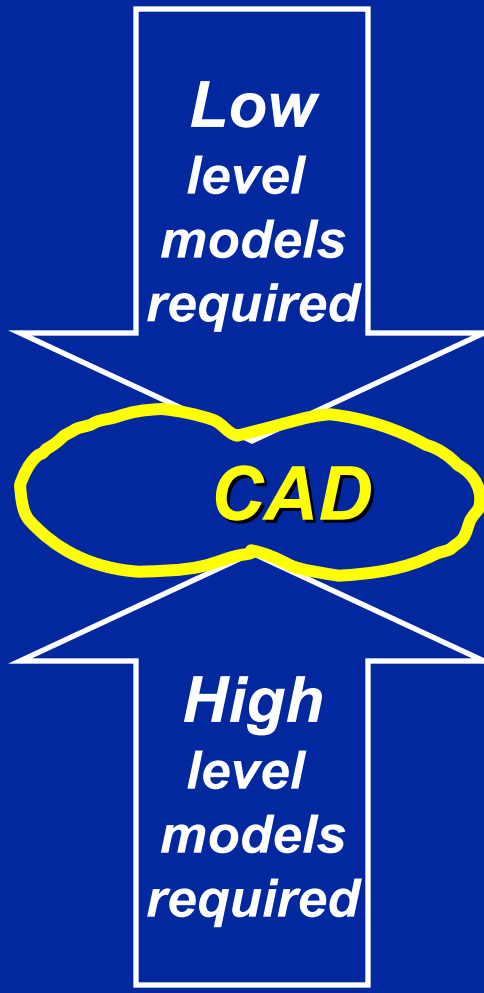
Future Technology Characteristics

(ITRS 99 - predictions for MPU chips)

Year	1999	2000	2008	2011
Technology [micron]	.18	.1	.07	.05
Transistors/chip	24M	190M	539M	1523M
Frequency [MHz]	1250	3500	6000	10,000
Wiring levels	7	9	9	10
Vsupply [V]	1.8	1.2	0.9	0.6
Power/chip [W]	90	160	170	174

Trends in *Deep Sub-Micron (DSM)*

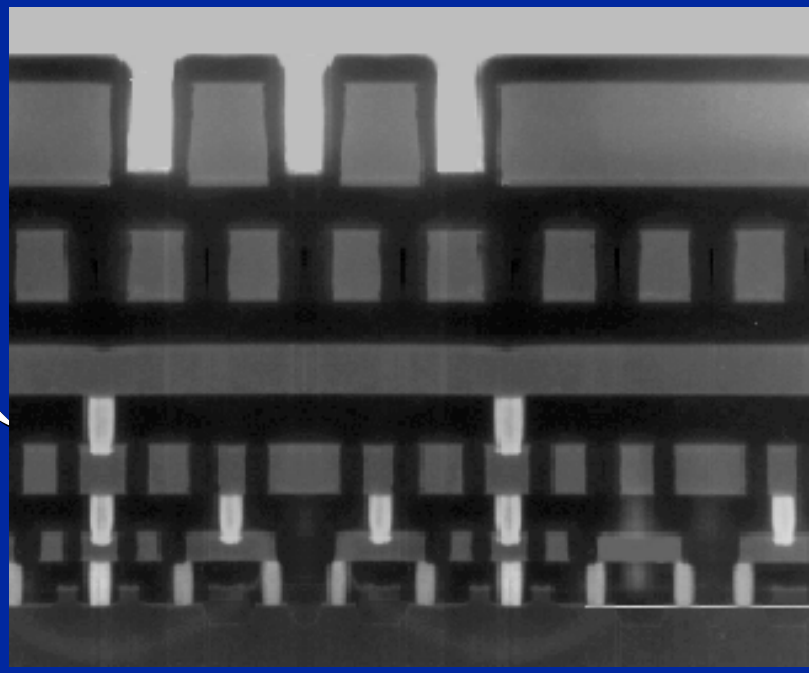
- **Physical effects at the circuit level:**
  - Interconnect design becomes critical
  - Crosstalk and switching noise
  - Heat dissipation (power) is severely limited
  - Complicated design rules and reliability requirements
- **System level requirements:**
  - System-On-Chip: Re-use, modularity, co-design of h/w and s/w
  - Accelerated development cycle



intel®

Noise is worse in DSM

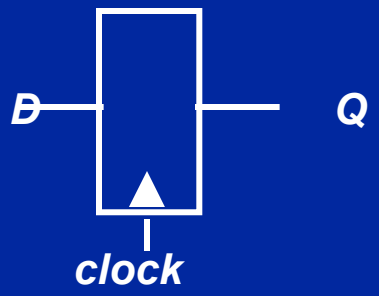
- Vsupply ↓ , Vthreshold/Vsupply ↑
  - Noise margins ↓
- Wire Resistance ↑, Cross-capacitance ↑
  - Crosstalk noise ↑
  - Must insert repeaters on wires to restore drive-strength
  - I*R voltage drop on power supply lines ↑
- Higher frequency → higher dV/dt
  - More coupling noise

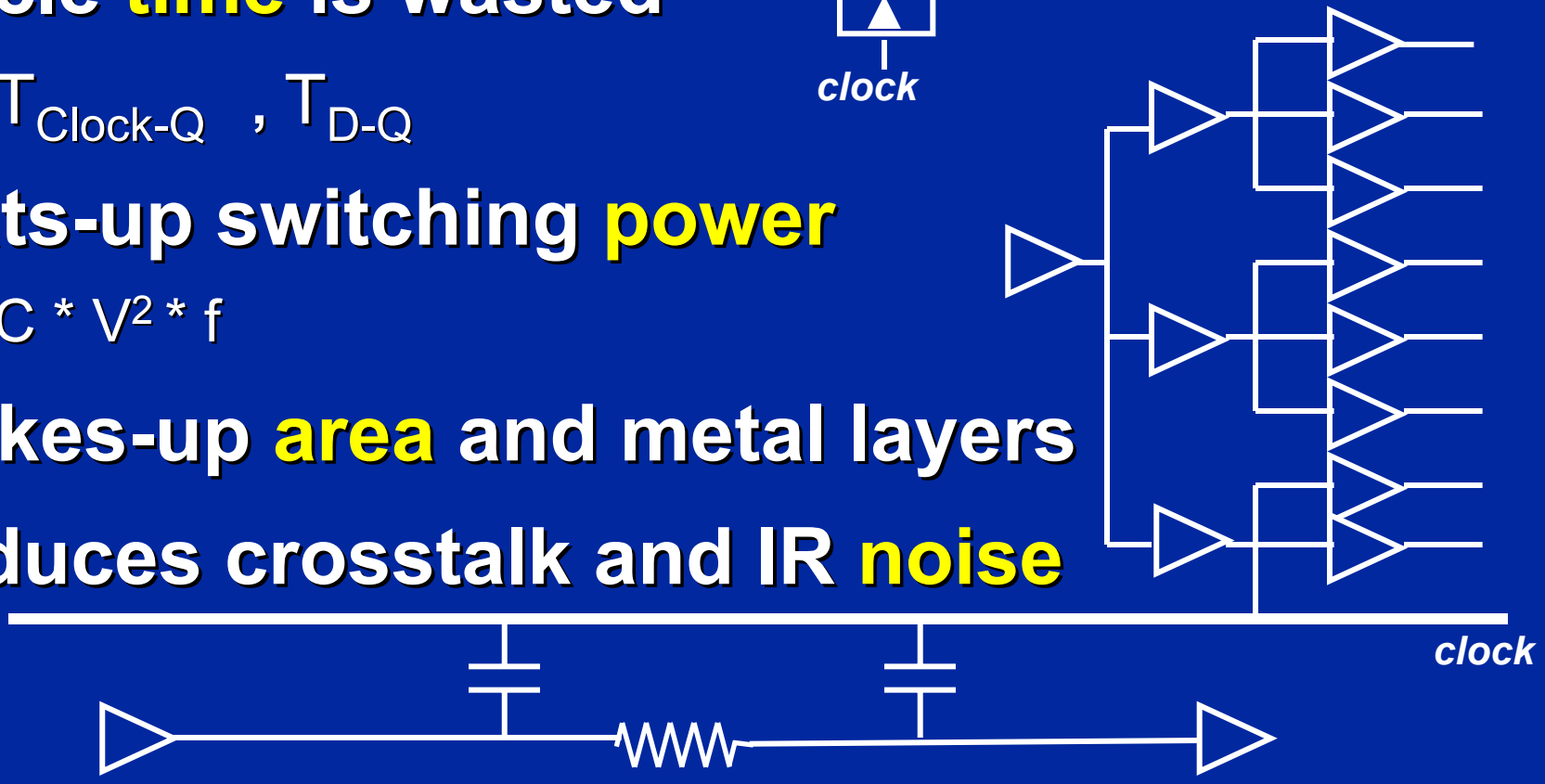


intel®

Clock is no longer a friend

- Significant part of cycle **time** is wasted
  - $T_{Clock-Q}$, T_{D-Q}
- Eats-up switching **power**
  - $C * V^2 * f$
- Takes-up **area** and metal layers
- Induces crosstalk and IR **noise**





Clock is no longer a friend 2

- People are used to clock-by- clock modeling
- This slows down the take-off of high-level design

intel®

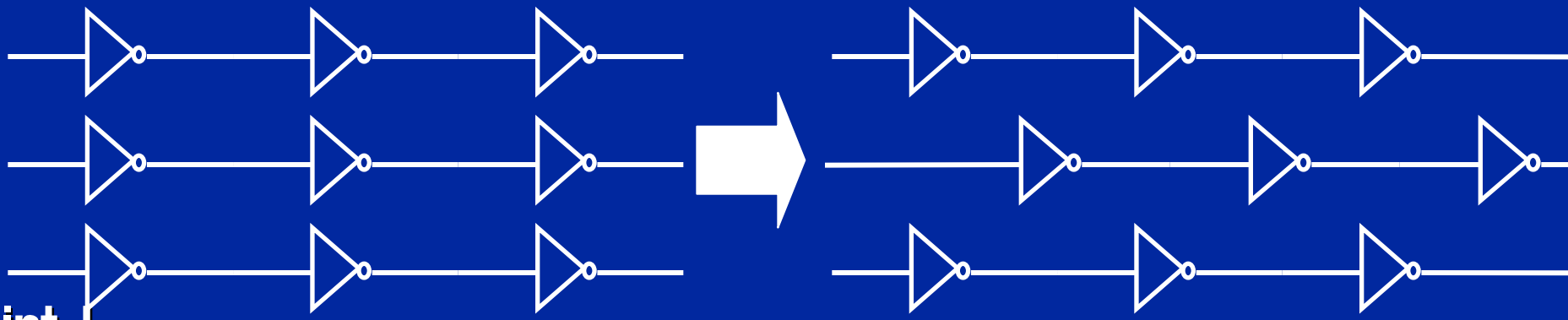
Workaround Approaches

- More domino logic
- Multiple frequencies on chip
- Selective stop-clock (power saving)
- Wave pipelining?
- Useful skew
- Master-slave FF losing favor?
  - Back to transparent latches
  - New latch designs for small Power*Delay
- Retiming (=“borrowing of logic”)
- Static noise analysis

intel®

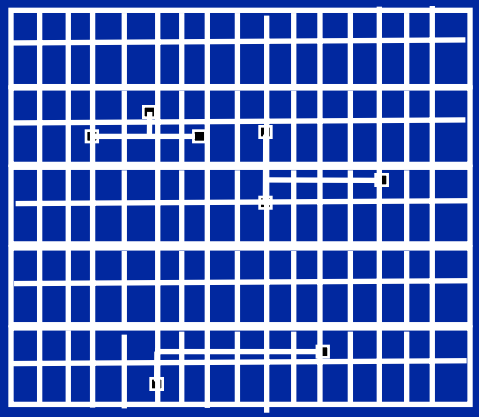
Interconnect-centered design

- Gates are ideal and free
- Wires require planning and optimization
  - Layer assignment, width, repeaters, driver sizing
- Noise+timing considerations



Time zones and chip assembly

- SOC is envisioned as a hierarchy of modules within several isochronous zones (clock domains)
- Design challenges:
  - Chip-assembly
  - Chip-level
    - delays
    - signal-integrity
    - power



intel®

The show must go on...

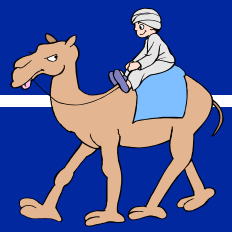
- More complex systems
- More performance required

‘10X’ technical forces

- Past:
  - Electronics → no moving parts & Relays
  - Solid-state → no vacuum Tubes
  - Integrated circuits → no coupling Capacitors
  - MOS technology → no Resistors
- Future:
  - DSM technology → no Clocks ???

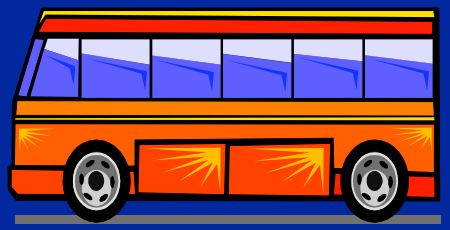
Synchronous Vs. Asynchronous: Will the lanes merge now?

- "The distinction ….. is very hazy in many cases of actual engineering interest" [G.H. Mealy, 1955]
- *Inertial delay* problem of learning by the engineering community….
  - Education on asynchronous techniques is a must!
  - Can CAD help?



Async

Sync



intel

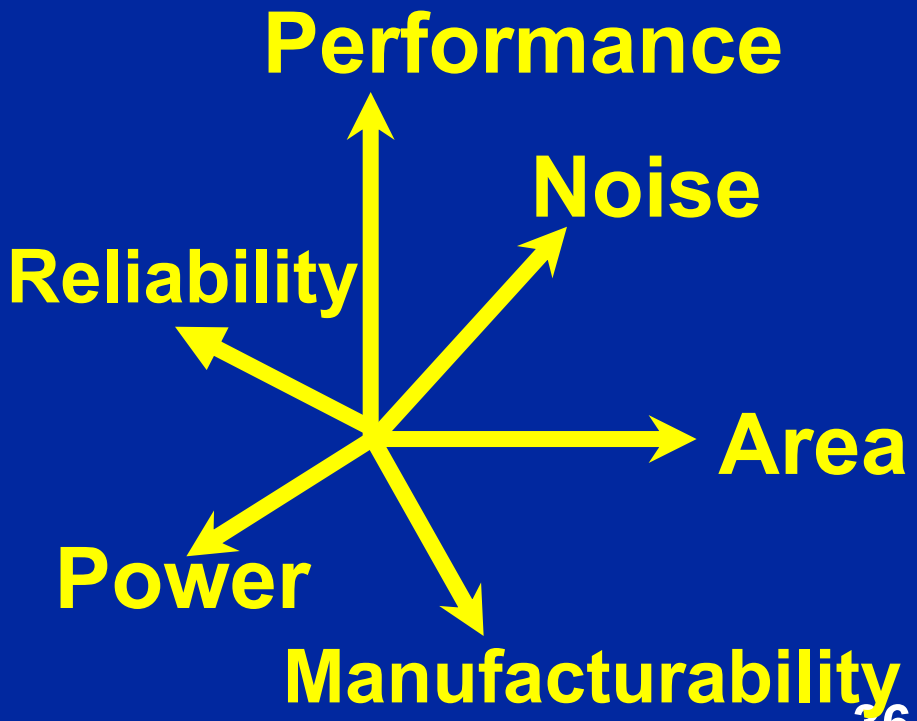
Insights on CAD

- Big productivity gains come from new design methods
- Tools and methods: chicken and egg
- CAD leverage
  - Evolution in tools causes revolutions in design work
- Successful tools take advantage of
  - Abstraction
  - Hierarchy
  - Regularity
  - Self-imposed restrictions
- Designers lose something and need to gain a lot in return
- 3 essentials:
  - Design capture
  - Synthesize
  - Verify

intel®

Summary

- Timing and synchronization issues will keep growing in importance
- Logic design will keep changing:
  - Modeling
  - Optimization goals
  - Methods
- 2 keys to success:
  - Education
  - CAD



intel