The Design of Micro-Electronic Circuits and Systems
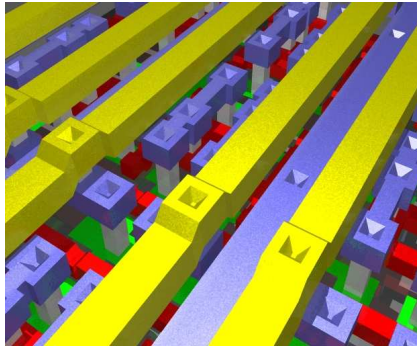## Automation, physical verification

Nick van der Meijs

EWI/CAS

nick@cas.et.tudelft.nl

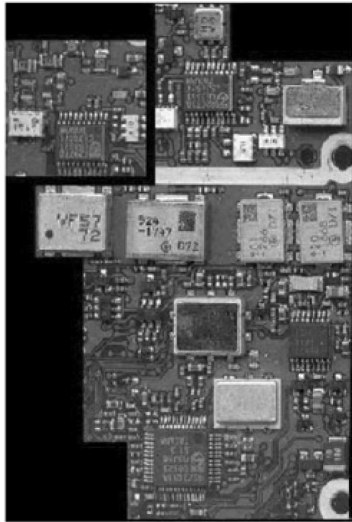**TU**Delft
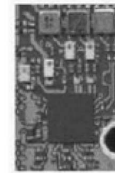

From Walkman to iPod

16TH CENTURY WALKMAN.

1979

[http://pocketcalculatorshow.com/walkman/sony/]
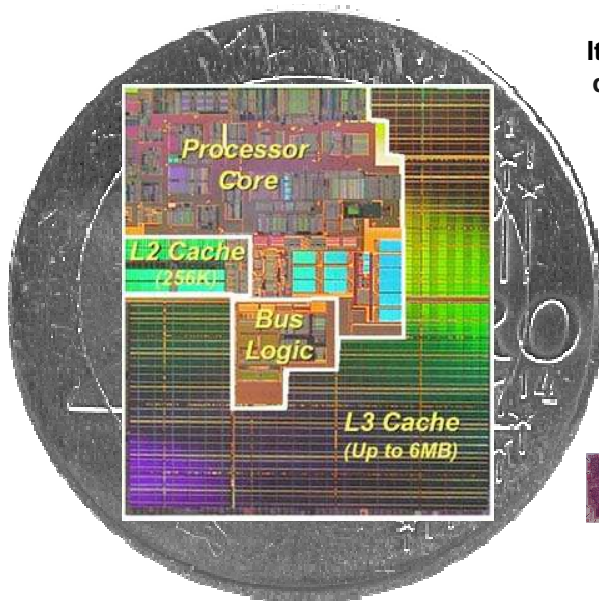
Integration Density

Ericsson GSM (S888, 1997)

Ericsson GSM (R520, 2000) same scale

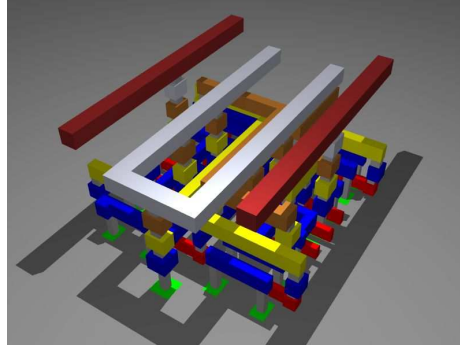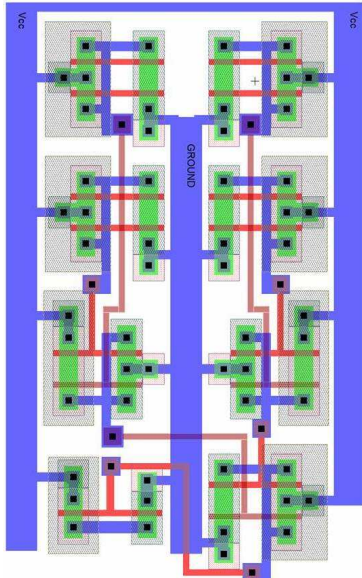**Whole point is to put more and more functionality into a chip**

Evolution

Intel 4004 (1971) and Itanium (McKinley, 2002) die compared to 2€ coin

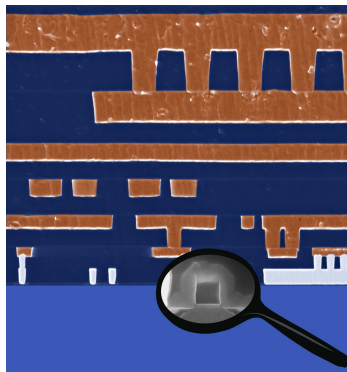Processor Core

L2 Cache (256K)

Bus Logic

L3 Cache (Up to 6MB)

Chip Anatomy

(Rendering of) result after fabrication

Not 1-to-1, sorry

Layout of chip, final design result

SEM Images

Cross-section
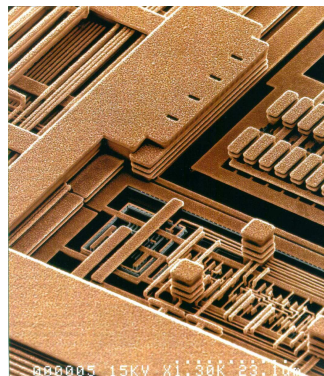
3d perspective

Design Challenge

■ **System Complexity**

Dealing with the sheer size of the system
- ■ $> 10^8$ components (transistors)
- ■ Compare boeing 747-400: 6×10^6 components
- ■ >> 10 km of interconnect
- ■ Compare boeing 747-400: 274 km wiring, 8 km tubing

■ **Silicon Complexity**

Dealing with the physical aspects
- ■ Features < 0.0000001 m = 100nm
- ■ Actually far from ideal behavior
  More like building spaghetti bridges then steel bridges
- ■ Lots of unwanted parasitics
- ■ Manufacturing tolerances, …

[http://www.boeing.com/commercial/747family/pf/pf_facts.html]

How To Cope?

■ **Abstraction and modeling**
- ■ **Focus** on separate issues of design at separate times
- ■ And by separate designers
- ■ What are proper abstractions?

■ **Modularity and Reuse**
- ■ Decompose a design in manageable pieces
- ■ Make sure you can understand / handle pieces in part
- ■ … and their composition
- ■ … on all the relevant abstraction levels
- ■ Reuse modules whenever possible

■ **Automation**
- ■ The computer is a designer's best friend
- ■ Implement methodology and design flow
- ■ Electronic Design Automation is fantastic research field

Abstractions

Behavior / Digital **Structure / Electrical** **Physical / Layout**

X	NOT X
0	1
1	0
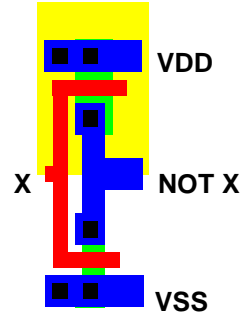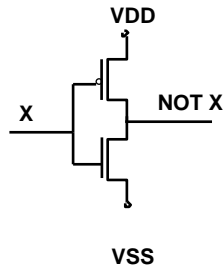
More Abstractions

```
if x = '0' then
    reg1 <= a;
else
    reg1 <= b;
end if;
```

$$i = C \frac{dV}{dt}$$

BEHAVIORAL DOMAIN   **STRUCTURAL DOMAIN**

Systems
Algorithms   Processors
Register transfers   ALU's, RAM, etc.
                                            Gates, flip-flops, etc.
        Logic   Transistors
Transfer functions

Transistor layout
Cell layout
Module layout
Floor plans
Physical partitions

PHYSICAL DOMAIN

5

Generic Chip Design Flow

```
if x = '0' then
    reg1 <= a;
else
    reg1 <= b;
end if;
```

(not so) detailed specs

architectural, system design

floorplan

register-transfer design

logic design

circuit design

layout

functional/ performance verification

testability

tapeout

VDD

In   Out

VSS

Design Actions

synthesis

verification

Synthesis:

> **increasing information about the design by providing more detail (within the same or another design domain).**

Verification:

> **checking whether a synthesis step has left the specification intact.**

Analysis:

> **collecting information on the quality of the design.**

Optimization:

> **increasing the quality of the design by rearrangements in a given description.**

Design Management:

> **storage of design data, cooperation between tools, design flow, etc.**

Design Automation Improvements

SOC Design Cost Model

(fictitious) design costs of low power pda chip

Reduced cost from better automation software

In-House P&R
Tall Thin Engineer
Small Block Reuse
Large Block Reuse
IC Implementation tools
Intelligent Testbench
ES Level Methodology

$$42,417,579$$
$$15,066,373$$

Total Design Cost (log scale)

$100,000,000,000
$10,000,000,000
$1,000,000,000
$100,000,000
$10,000,000

1985   1990   1995   2000   2005   2010   2015   2020

Year

◆ RTL Methodology Only
■ With all Future Improvements

Source: itrs 2001

Methodologies and Tools

- **Electronic Design Automation is key enabling technology!**
- **The software tools to do the actual IC design and engineering**
- **These tools need to be designed and developed, too!**
- **Need constant improvement to cope with technology progress**

NVIDIA Example (A.D. 2000)

- ~ **850 employees** (worldwide total incl. sales, mgmt, …)
- ~ **$85M of CAD tools**
- ~ **$20M emulation**
- **Engineering Compute Resources**
  - **Desktops: 200 Sun / 2150 pc's**
  - **Servers:**
    278 Sun / 634 Linux / 496 Gbytes RAM
  - **14 Terabytes of storage**

1 Terabyte (TB) = 1000 GB

~ 1.000.000 large digital photos

~ 10 km of books on a shelf

~ 1 year of music ~ 200 iPods

NVIDIA - 2005

3000 compute servers　　　　　**1700 engineers**
2000 employees

NAS

Fibre Channel
SAN

Sacramento DR Site

Clustered NetApp
Storage Systems

SnapMirror

High End
NetApp FAS Systems

Mid Range
NetApp FAS Systems

Tier 3 Storage
NetApp NearStore
Systems

2 NearStore Systems
84TB
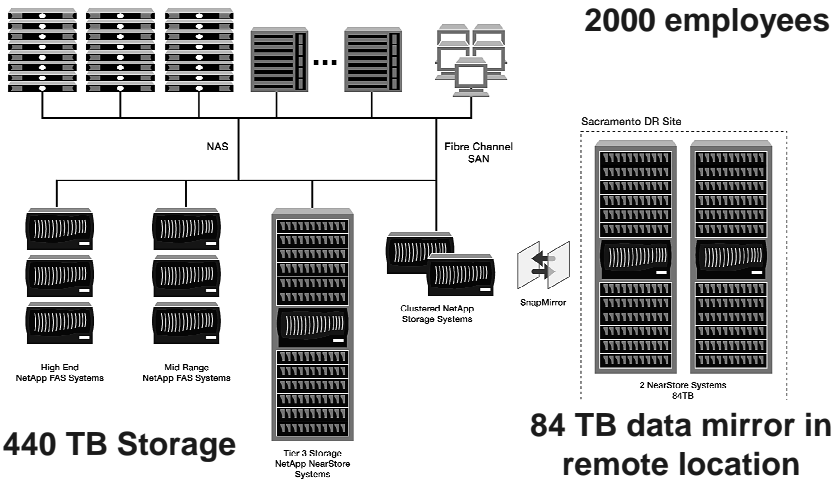
440 TB Storage　　　　　**84 TB data mirror in
remote location**

[http://www.netapp.com/library/cs/nvidia.pdf, 061004]

Summarizing ...

- **System Complexity**
- **Silicon Complexity**

- **Abstraction and Modeling**
- **Modularity and Reuse**
- **Automation**

My Research

- **Main topic: physical / electrical modeling of IC's**

- **Design of IC's is making approximations, assumptions, short-cuts, simplifications**
- **Real world physics is too complex to handle**
- **Need independent verification steps to check a design before fabrication**
  - **Will the chip actually work?**
  - **Is it within performance, power, reliability, etc. budget?**

- **See www.space.tudelft.nl**

- **You might remember this when choosing your MSc topic**

ASIC First Spin Errors

Functional Logic Error	43%
Analog Tuning Issue	20%
Signal Integrity Issue	17%
Clock Scheme Error	14%
Reliability Issue	12%
Mixed Signal Problem	11%
Uses Too Much Power	11%
Has Path(s) Too Slow	10%
Has Path(s) Too Fast	10%
IR Drop Issues	7%
Firmware Error	4%
Other Problem	3%
Require at least one re-spin	61%

Source:
Aart de Geus,
Chairman & CEO, Synopsis
ESNUG keynote, Sep 8, 2003

Introduction micro electronics - EDA - © 2005 NvdM   *19*

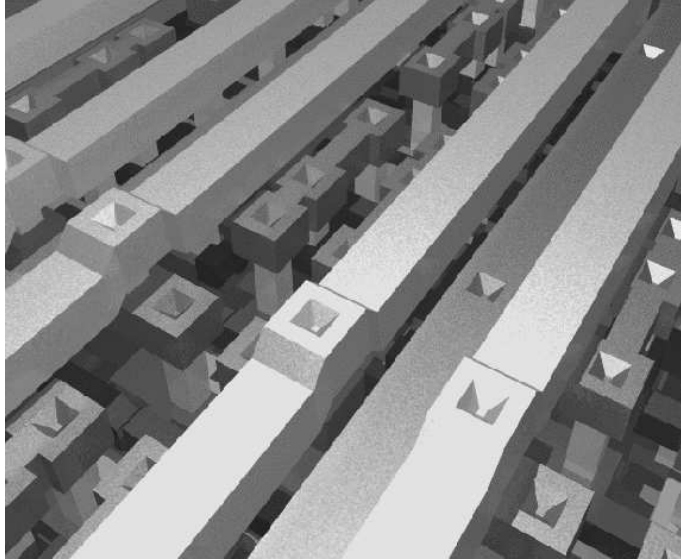# Physical Verification

3

Layout-to-Circuit Extraction

```
m1 11 12 13 14 penh_0 w=250n l=250n
m2 15 16 17 14 penh_0 w=250n l=250n
r1 15 4 329.7622
c1 15 18 1.096202f
r2 11 4 307.2622
c2 11 18 1.028856f
c3 4 18 2.004567f
c4 14 18 16.71091f
m3 19 20 21 22 nenh_0 w=250n l=250
m4 23 24 25 22 nenh_0 w=250n l=250
r3 13 8 1.709575k
c5 13 18 1.435885f
r4 19 26 157.2081
r5 19 8 3.320258k
c6 19 18 394.353a
r6 26 8 3.233061k
c7 26 18 418.6736a
r7 16 24 213.8774
r8 16 8 7.274456k
c8 16 18 110.7094a
...
```
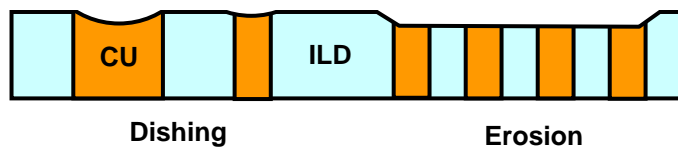
simeye

File   Edit   Simulate   View   Options   Help

viewing: rand_cnt.ana   spice   Analog

simulation ready

fb_in
in[0]
in[1]
in[2]
in[3]
in[4]
in[5]
in[6]
in[7]
out[0]
out[1]
out[2]
out[3]
out[4]
out[5]
out[6]
out[7]
vdd_l1
P_ld_l
run_l
phi1_l
phi2_l
vdd_l2
vdd_r1
P_ld_r
run_r
phi1_r
phi2_r
vdd_r2
vdd_r3
vdd_l3

e-07   0   5   10   15   20   25   30

Simulation

Computation of Capacitance

Unsolved Physical Verification Issue
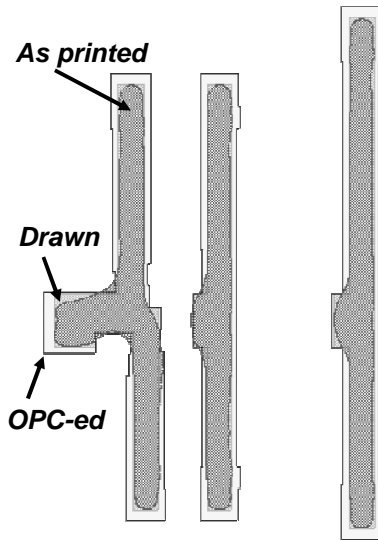# Metal CMP Effects

- **Systematic, layout-process interaction based shape variability**
- **Has a pronounced effect on capacitance values**
- **Effect can be modeled with 'local layout density' concept**
- **(Effective) metal height**

Dishing　　　　　　**Erosion**

Lithography Effects on C

As printed

Drawn

OPC-ed

From Ren et. al., CEM2006

- **90 nm technology, 193 nm litho**
- **About 5% difference in self-C**
- **About 9% difference for some coupling C**
- **Systematic effect, can be modeled in theory**

- **Need extraction solutions**

My Lectures:
ET 4255 Electronic Design Automation

- **Principles, methods, algorithms for Computer-Aided design of electronic systems**
- **Here: IC design**
- **Not about design**
- **But: how do I built the tools to do the design**

- **Lecture period: 4th quarter**

Electronic Design Automation

- **Principles, methods, algorithms for Computer-Aided design of electronic systems**
- **Here: IC design**
- **Not about design**
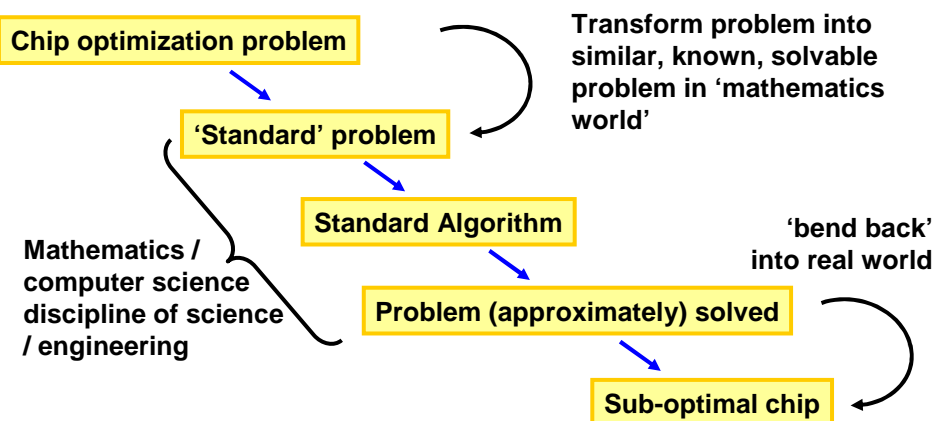- **But: how do I built the tools to do the design**

Why Education in EDA?

- **Without EDA no chips**
- **Without chips no …**
- **Without electrical engineers no EDA**

- **Very advanced software**
    - **Example: design rule checking**
    - **Example: Pentium fdiv bug**
- **You will learn a lot**
- **That you can apply at numerous other occasions**
    - **Abstraction, modeling are key**

- **Why electrical engineers for EDA?**
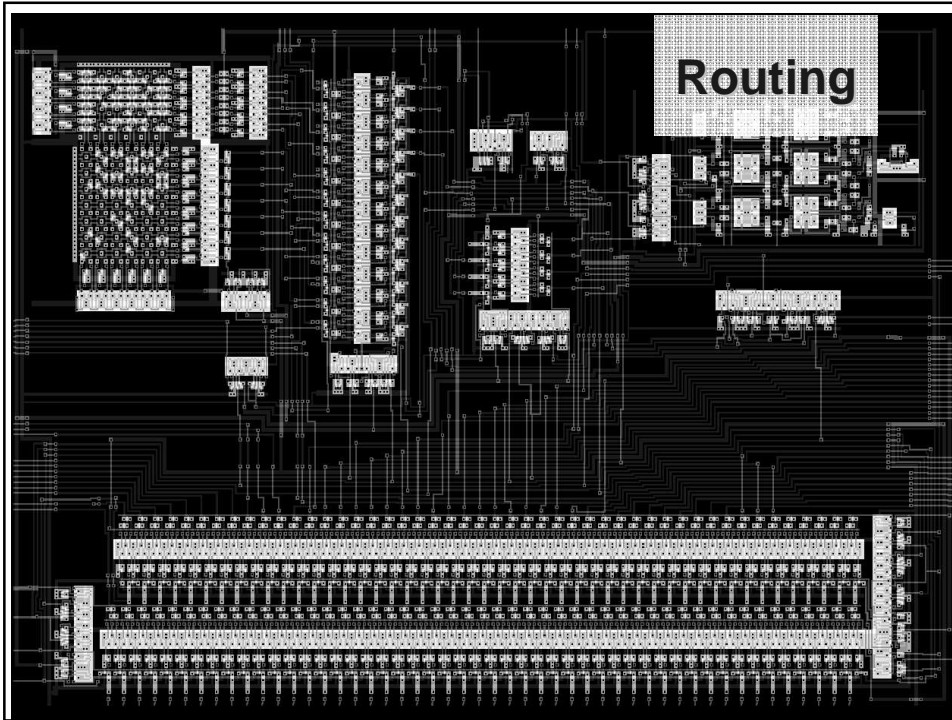- **You will have to understand the underlying issues!!!**

EDA Education Goals

- **Show current and future CAD tool users what is going on inside tools**
    - **Appreciate possibilities and limitations**
    - **Complexity of problems**
- **First introduction to students that want to specialize in CAD**
- **Example of solving complex technical automation problems**
    - **Useful for other branches of technology**
    - **Motivation for other algorithm courses**
    - **Also for Non-EE students**
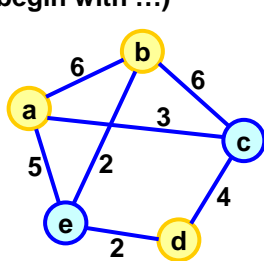
Typical EDA Optimization Setting

Chip optimization problem

Transform problem into similar, known, solvable problem in 'mathematics world'

'Standard' problem

Standard Algorithm

'bend back' into real world

Mathematics / computer science discipline of science / engineering

Problem (approximately) solved

Sub-optimal chip

Optimization often part of synthesis
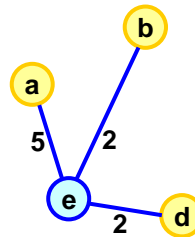
Routing

Example 1: Steiner Tree

Steiner tree is 'standard' abstraction of typical routing applications

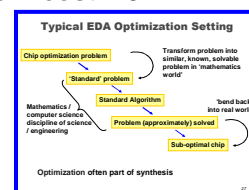**Problem: Steiner tree problem is NP complete (as is routing to begin with …)**

Solution: cost = 9

G = {a, b, c, d, e}

R = {a, b, d}

Determine a Minimum Steiner Tree on R in G
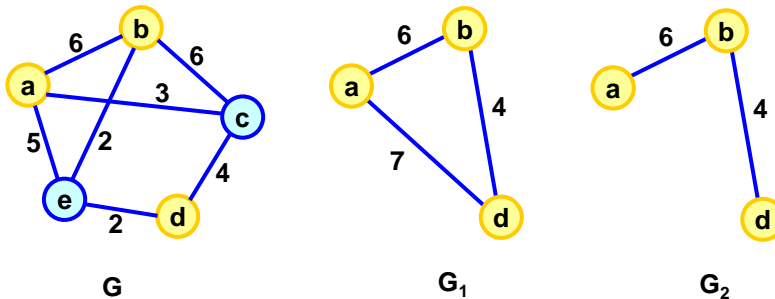
Steiner Tree Heuristic (simple)

Input: graph G = (V, E)

net R = ⊂ V

Output: approximation of MStT on R, in G

Algorithm:

1. **Determine complete distance graph G_1 among vertices of R. Weight of edge (v, w) is shortest path between v and w in G.**
2. **Determine MST G_2 of G_1**
3. **Replace each edge in G_2 by corresponding path in G. This is G_3**
4. **Determine MST G_4 of G_3**
5. **Remove all leaves of G_4 which are not in R. This is the approximation of MStT.**

Steiner Tree Heuristic (simple)

1. **Determine complete distance graph G_1 among vertices of R. Weight of edge (v, w) is shortest path between v and w in G.**
2. **Determine MST G_2 of G_1**

Steiner Tree Heuristic (simple)
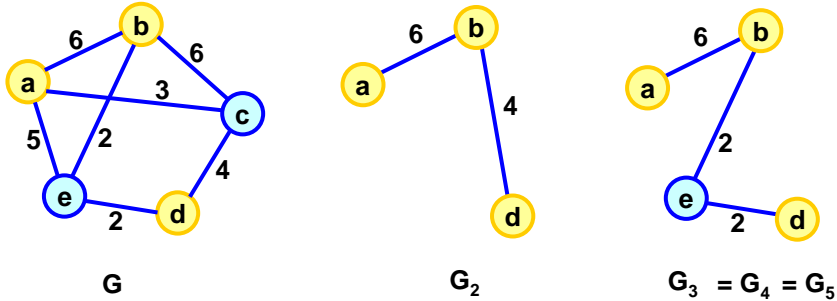
G G_2 $G_3 = G_4 = G_5$

3. Replace each edge in G_2 by corresponding path in G. This is G_3
4. Determine MST G_4 of G_3
5. Remove all leaves of G_4 which are not in R. This is the approximation of MStT, G_5.

Cost = 10: Sub-optimal!

Example 2: Handling Trade-Offs

× Pareto points
O other points
— trade-off curve

D=3+7=10
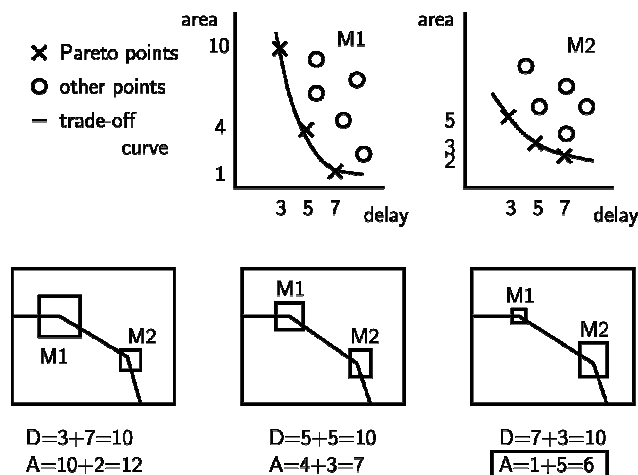A=10+2=12

D=5+5=10
A=4+3=7

D=7+3=10
A=1+5=6

Figure 2.3: *Different modules will have different area-delay points. Only the Pareto points are of interest and span up an area-delay trade-off curve for the module. Combining two modules several areas are possible at equal delay.*

[Dirk Jan Jongeneel]

That's All

- **VLSI Design is challenging**
- **Issues arise in digital, analog, mixed signal and RF design**
- **Always want to be at the edge of technologically possible**
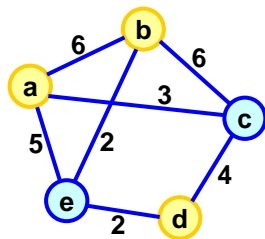
- **Need continuous research and education**

Backup

ET4255 Programming Contest

This year's contest will feature Steiner Tree Generation

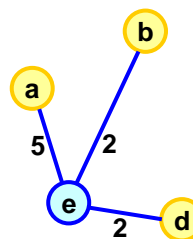**Given:**   **a (large) set of points**

Required: **a minimum length connection among a subset of these points**
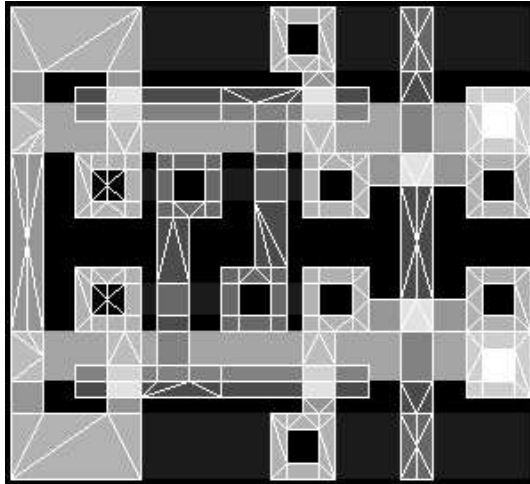
Who can write the fastest program?

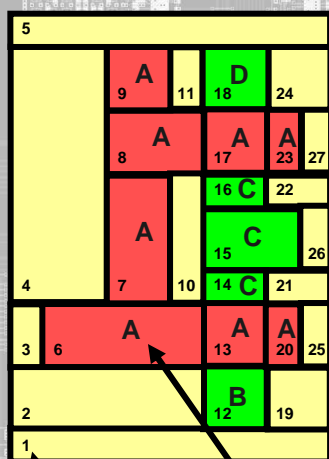G = {a, b, c, d, e}
R = {a, b, d}

Solution: cost = 9
(sub-optimal)

Layout-to-Circuit Extraction Examples

Node Numbering

Tile number   **Net 'number'**

- **Essential step during layout-to-circuit extraction**

- **Layout is composed of rectangles**

- **Each rectangle has a so-called tile-number.**

- **Give each rectangle a second number (the net-number) for the electrical net to which is belongs.**

- **Rules for two-layer simplification:**

  - **All red is connected (if it abuts)**

  - **All green is connected (if it abuts)**

  - **Grey is empty space**

- **Consider scalability to billions of rectangles.**

Node Numbering (2)

find all neighbors of tile
in constant time

Algorithm
Input: **A corner-stitched layout data structure**
Result: **Each tile is numbered with node number**
(numbers equal if and only if tiles are electrically connected)

```
n := 1        # start with node number 1
for all tiles t
   if t contains interconnect and t.node = nill then
      extract_node (t, n)
      n := n + 1

extract_node (t, n)
   t.node := n
   for all connected neighbors b of t
      if b contains interconnect and b.node = nill then
         extract_node (b, n)
      if b contains a contact and b.node = nill then
         o := corresponding tile in other plane
         extract_node (o, n)
```

Node Numbering (3)
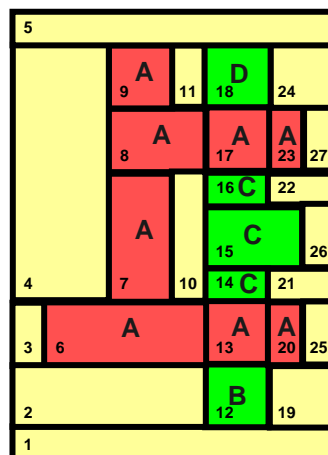

```
extract node (6,A)
     3  4  7  10  13  2
   extract node (7,A)
       4  8  10  16
     extract node (8,A)
         4  9  11  17  10  7
       extract node (9,A)
           4  5  11  8
       extract node (17,A)
           8  18  23  16
         extract node (23,A)
             17  24  26  22
   extract node (13,A)
       6  14  20  12
     extract node (20,A)

extract node (12,B)
   ...
```
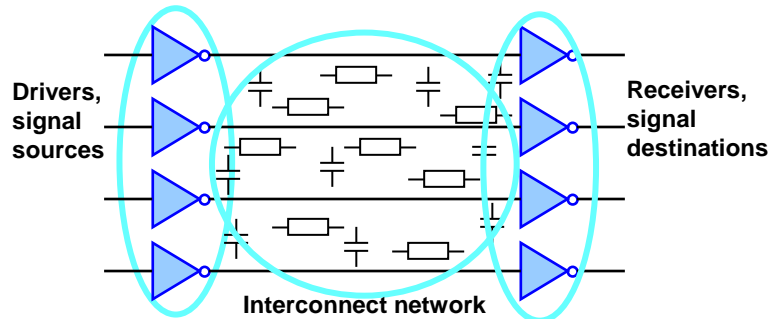
Interconnect Parasitics

■ **Chip Timing dominated by interconnect R, C**

Drivers, signal sources

Receivers, signal destinations

Interconnect network

■ **R and C distributed along interconnect**
■ **Need initial, fine-granularity RC mesh**
■ **Huge data sets**
■ **Impossible to analyze directly**

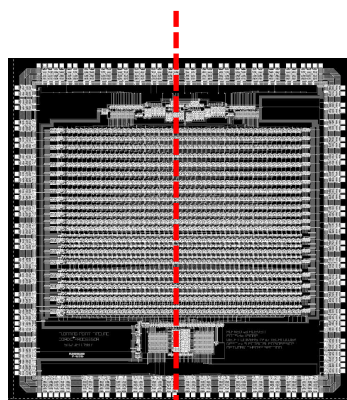## Need Layout Parasitics Extraction + Model Order Reduction

---

Fundamental Approach

Based on Scanline Algorithm

■ **Operations take place in a narrow band sliding over layout from left to right**
■ **Layout data read in A.L.A.P.**
■ **Circuit data written out A.S.A.P.**
■ **Sublinear memory complexity**
■ **Near-linear time complexity**

**Scanline**

Interconnect Resistances

- **FEM to determine resistances**
- **Solution only necessary at 'boundary nodes'**
- **Model reduction via Gaussian elimination of internal nodes (matrix based)**
- **FE Mesh ⇔ resistance network**
- **Gaussian elimination ⇔ star-delta transformation: network based**
- **Node can be eliminated when all its neighbors are known**
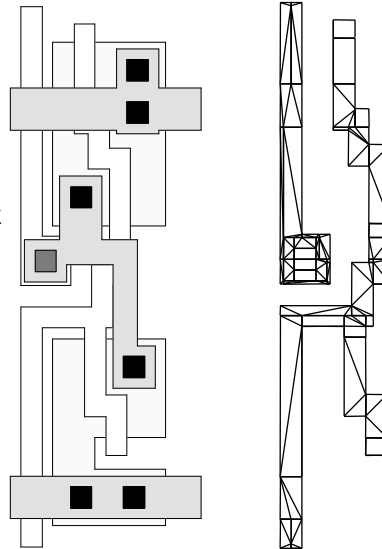
Gaussian Elimination for RC Networks
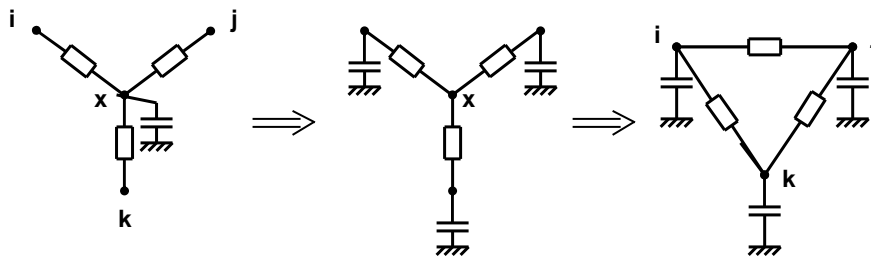
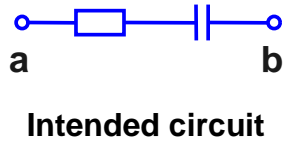$$C_i := C_i + C_x \frac{G_{xi}}{\sum_{n \notin x} G_{xn}} \qquad G_{ij} := G_{ij} + \frac{G_{xi}G_{xj}}{\sum_{n \notin x} G_{xn}}$$

- **Gaussian Elimination = Star-delta transformation**
- **Elmore delay (first moment of impulse response) is preserved**
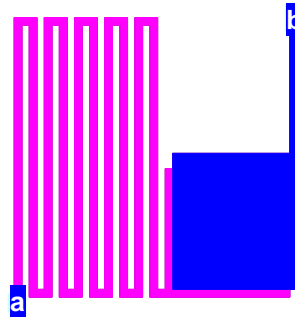- **Also for coupling capacitances**
- **On-the-fly elimination**

SNE Example: Snake RC

a ———— **b**

Intended circuit

Layout

Initial 3D extraction ⇨ **lumped RC mesh**
109 nodes
164 resistors
720 capacitors

SNE circuit results for Snake RC

1 MHz   **100 MHz**   **200 MHz**
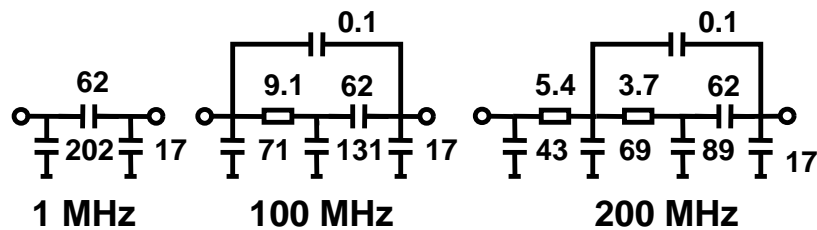
| Exact | SNE | | | AWE |
Poles	100 MHz	200 MHz	500 MHz	N=3
$8.8 \cdot 10^{08}$	$7.6 \cdot 10^{08}$	$8.4 \cdot 10^{08}$	$8.7 \cdot 10^{08}$	$8.8 \cdot 10^{08}$
$9.9 \cdot 10^{09}$		$8.4 \cdot 10^{09}$	$8.9 \cdot 10^{09}$	$1.0 \cdot 10^{10}$
$2.4 \cdot 10^{10}$			$2.1 \cdot 10^{10}$	$2.1 \cdot 10^{10}$

SNE simulation results for Snake RC

---
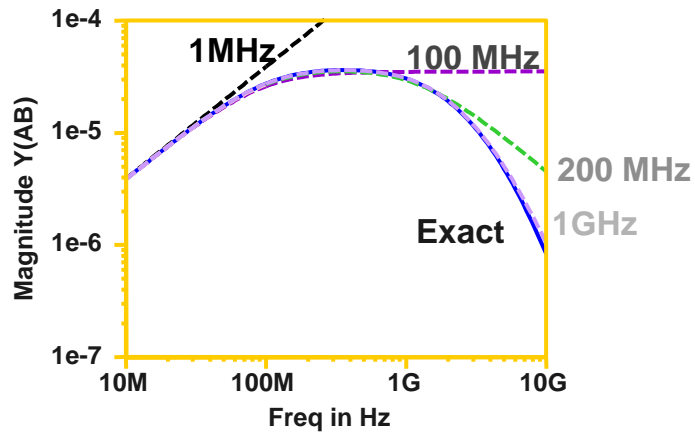
4 Conclusion
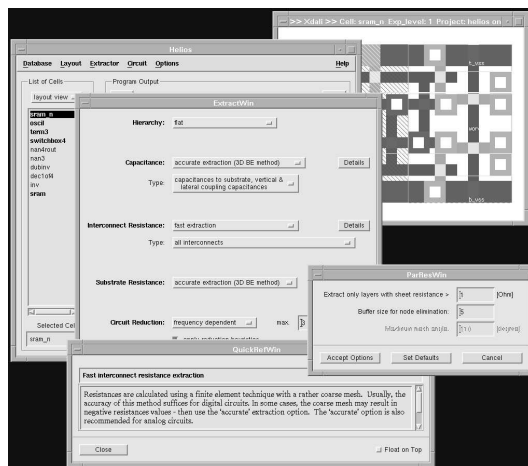
- **Physical verification of Integrated Circuits**
- **From geometry (chip layout) to reduced order model (netlist)**
- **Devices, interconnect (R, C), substrate**
- **Model Order Reduction as soon as possible**
- **Consistency of models remains a challenge**
- **Being used in practice**

space.tudelft.nl

Backup

IC	Integrated circuit
LSI	Large scale integration
VLSI	Very large scale integration
ULSI	Ultra large scale integration
DSM	Deep sub micron
VDSM	Very deep sub micron

Design Flow Changes

Required Advance in Design System Architecture

Today 180 nm → Tomorrow 50 nm

- Verification moves to earlier higher levels, followed in lower levels by equivalence checking and assertion driven optimizations
- Design optimized over many constraints with tightly integrated analysis and optimization
- Integration through modular open architecture with industry standard interface for data control
- Shared data in memory to eliminate disk accesses in critical loops with common data for cooperating applications
- Incremental analysis and optimization

■ IC Industry Economics

Fab costs >> $1.000.000.000

- ■ **Time to market is <mark>EVERYTHING</mark> (nearly)**
  - ■ **New game station after the Christmas season can mean bankruptcy**
- ■ **Design has to be right first time**
  - ■ **Not a second chance**
  - ■ **Very reliable design procedures needed**

Modeling

model (REPRESENTATION) /£ ˈmɒd.əl, $ ˈmɑː.dəl/ *noun* [C]
a representation of something, either as a physical object which is usually smaller than the real object, or as a simple description of the object which might be used in calculations
a plastic model aircraft
By looking at this model you can get a better idea of how the bridge will look.
to construct a statistical/theoretical/mathematical model
No model **of** the economy can predict when the next recession will be.
Computer models have been used to predict long−term climatic changes.

model /£ ˈmɒd.əl, $ ˈmɑː.dəl/ *verb* [T]
to model animals out of clay
to model clay into animal shapes
The whole car can be modelled on a computer before a single component is made. [T]

(*Cambridge International Dictionary of English*)

Modeling

- An abstraction of (the properties) of something to help understanding and predicting its behavior
- Domain Specific: weather, climate, economy, stock market, ...
- Different models for something to answer different questions
- Black-Box modeling vs. Physically Based
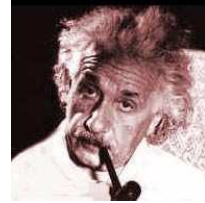
- After Einstein: **[a model] should be as simple as possible, but not simpler**

Abstraction/Modeling is key enabling factor in the success (E)E.