IC design Tools

Short overview of layout , simulation , synthesis and design creation tools

This is additional material to chapters 1 and 2 from De Micheli

Please read chapters 1 and 2 of De Micheli before next Tuesday

- Logic Synthesis is only a part of the entire design process.
- It has links with behavioral and system synthesis on top and physical and circuit synthesis on the bottom.
- So, we need some superficial understanding of the whole process and its tools

Cell Development comes first

Cell development tools

- **Schematic entry** (transistor symbols)
- **Analog simulation** (SPICE models)
- **Layout** (layer definitions)
- **Design Rule Checking**, DRC (design rules)
- **Extraction** (extraction rules and parameters)
- **Electrical Rule Checking**, ERC (ERC rules)
- **Layout Versus Schematic**, LVS (LVS rules)

Cell development tasks

What is involved in cell development:

- Analog simulation.
- **Characterization:** delay, setup, hold, loading sensitivity,etc.
- Generation of **digital simulation** model with back annotation.
- Generation of a synthesis model
- Generation of symbol and black-box for place & route

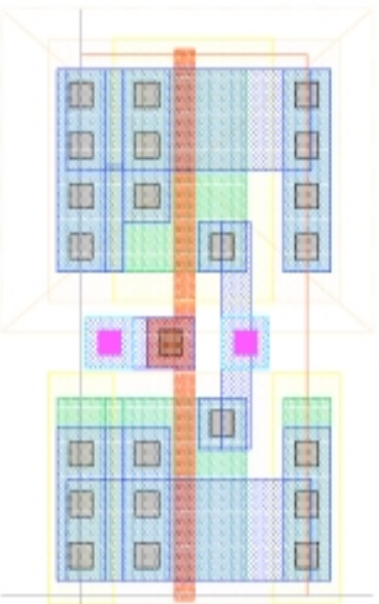
Digital design tools

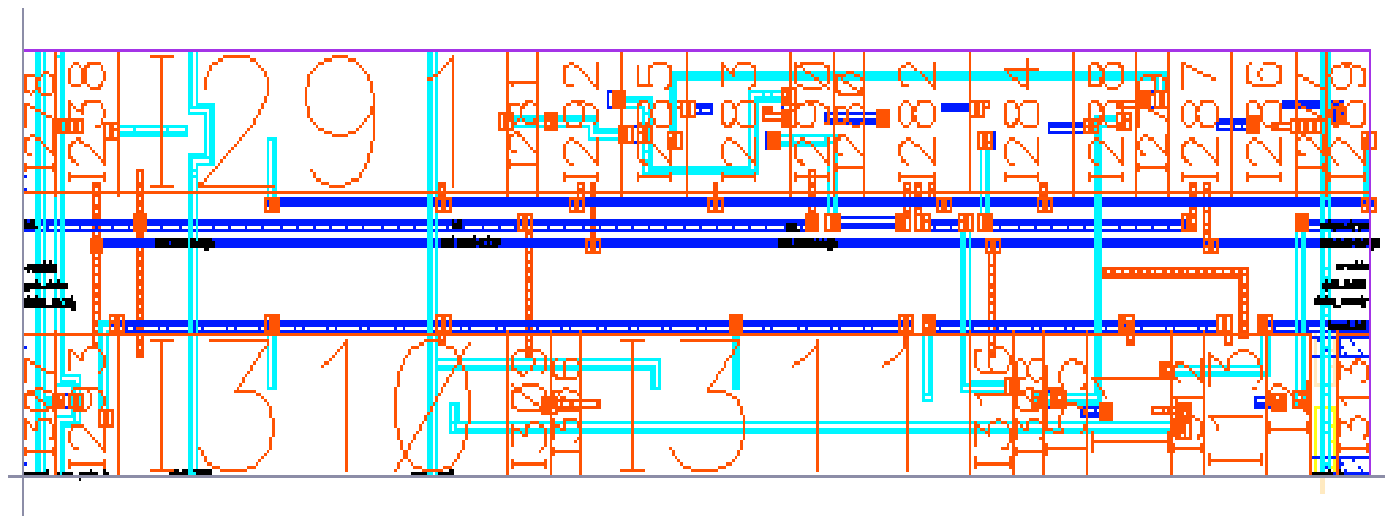
- **Behavioral simulation**
- **Synthesis** (synthesis models)
- Gate level simulation (gate models)
- **Floor planning**
- **Loading estimation** (loading estimation model)
- Simulation with estimated back-annotation
- **Place and route** (place and route rules)
- **Design Rule Check**, DRC (DRC rules)
- Loading **extraction (**rules and parameters)
- Simulation with real back-annotation
- Design export
- Testing: Test generation, Fault simulation, Vector translation

} Or direct **schematic entry**

Design entry

- Layout
  - Drawing geometrical shapes:
    - Defines layout hierarchy
    - Defines layer masks
      - Requires detailed knowledge about CMOS technology
      - Requires detailed knowledge about design rules
      - Requires detailed knowledge about circuit design
      - Slow and tedious
      - Optimum performance can be obtained
      - No yield guarantee from manufacturer when making full custom cells





- Schematic
  - Drawing electrical circuit: Defines electrical hierarchy
    Defines electrical connections
    Defines circuit: transistors, resistors,,,

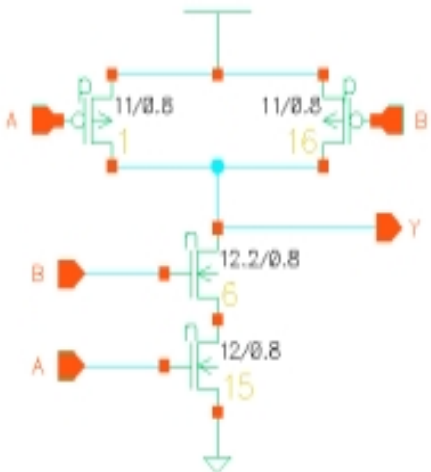
    Requires good circuit design knowledge for analog design

    Requires good logic design knowledge for digital design (boolean logic, state machines)

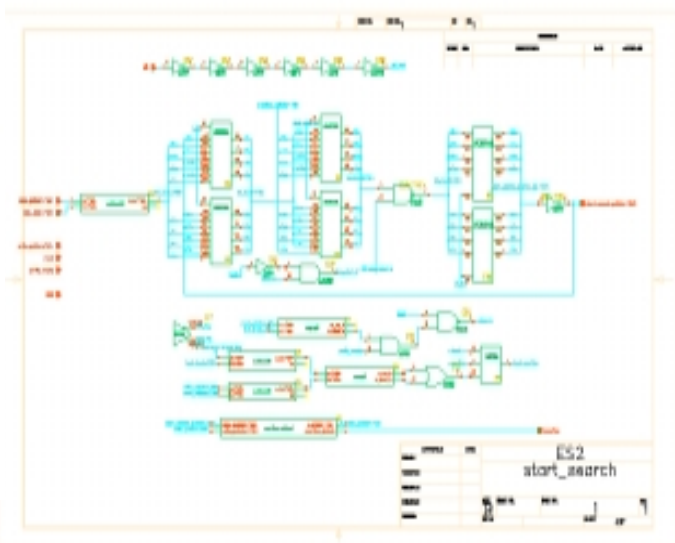
    Gives good overview of design hierarchy

    Significant amount of time used for manual optimization

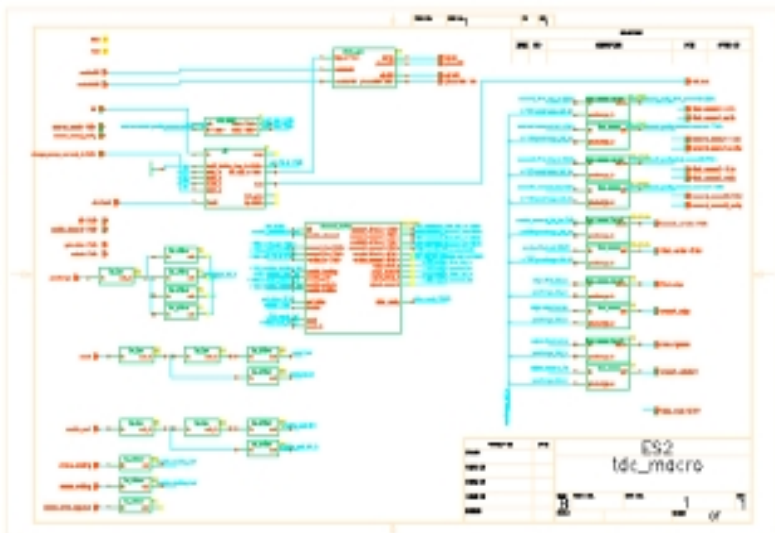
Transistor level



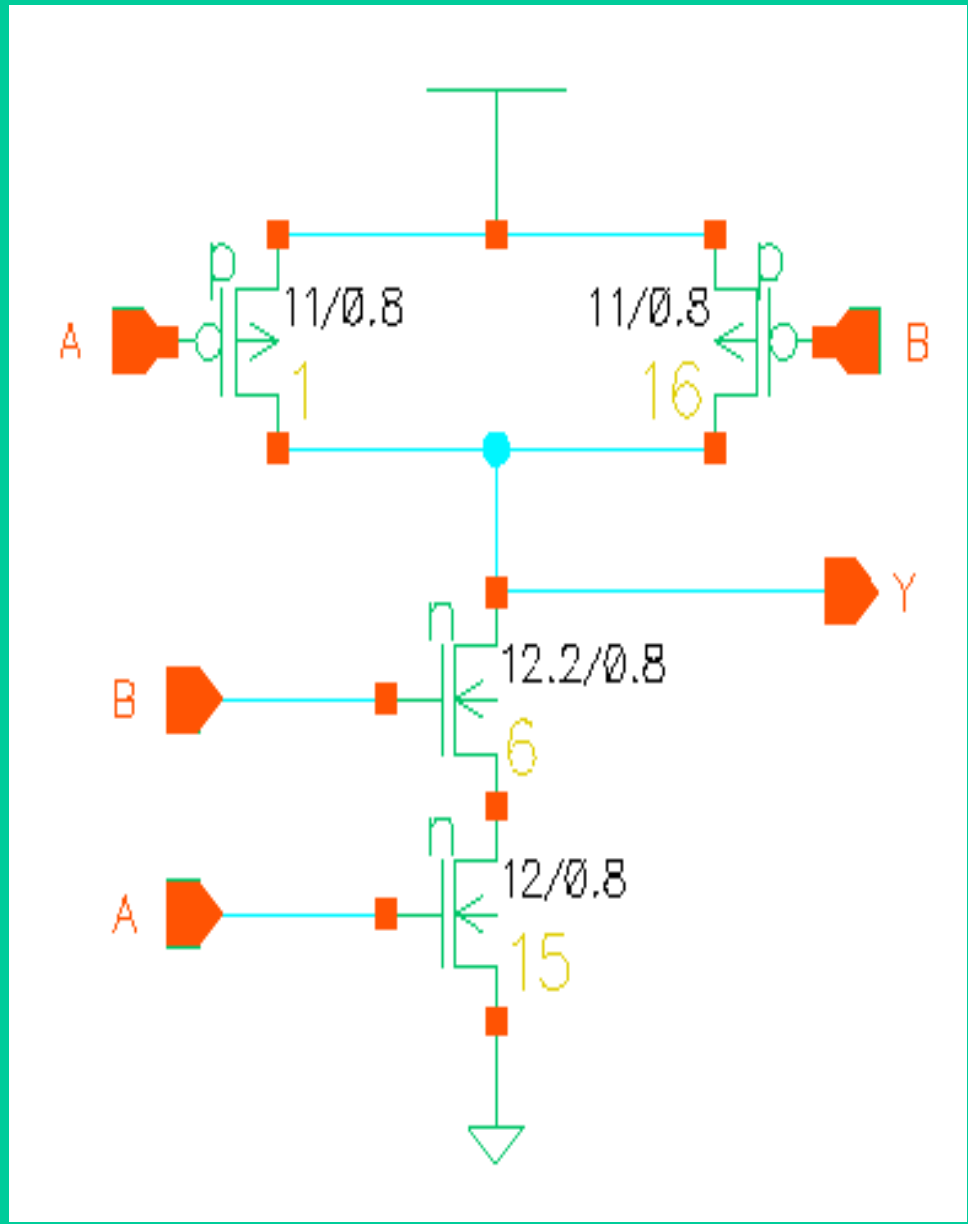
Gate level



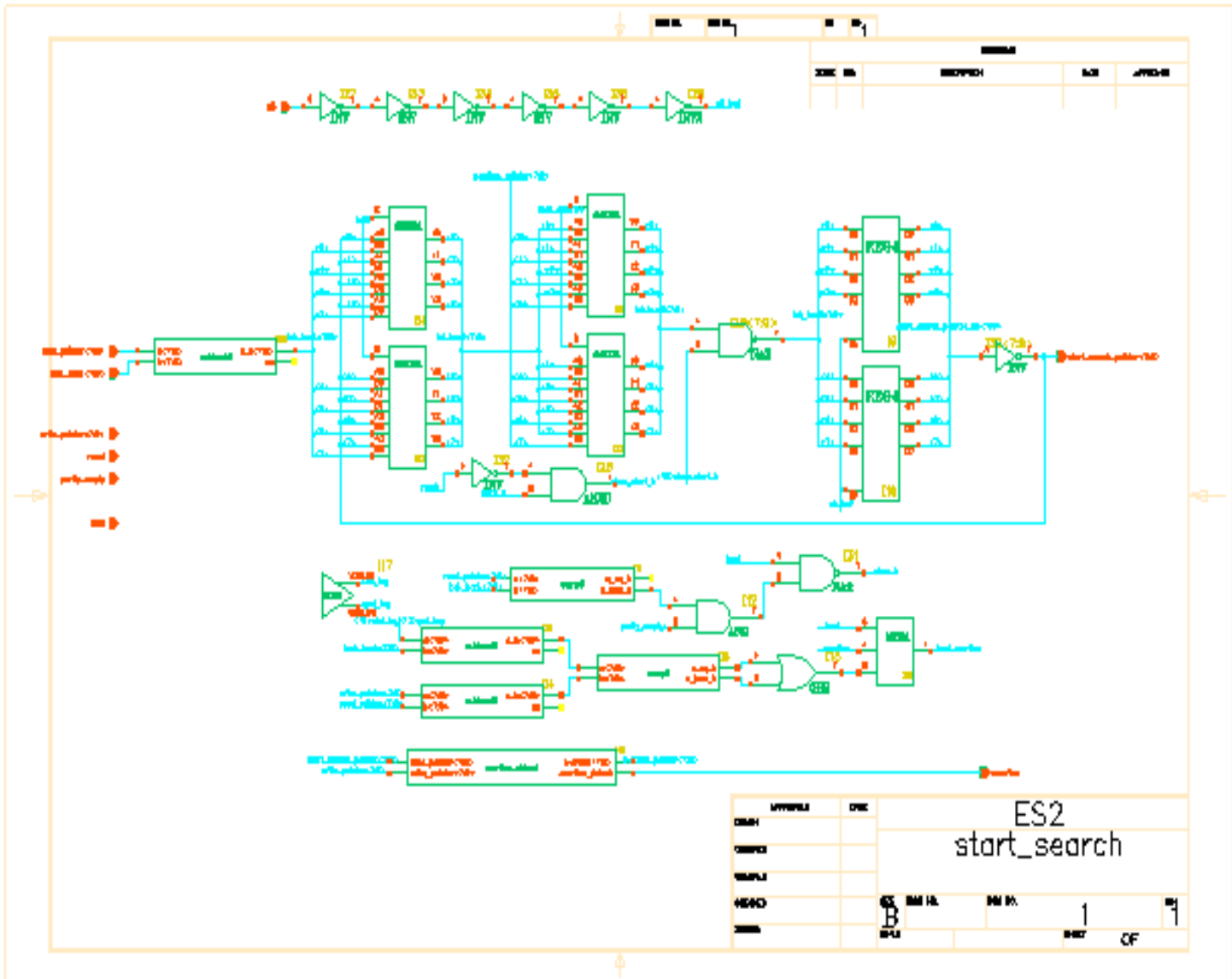
Module level



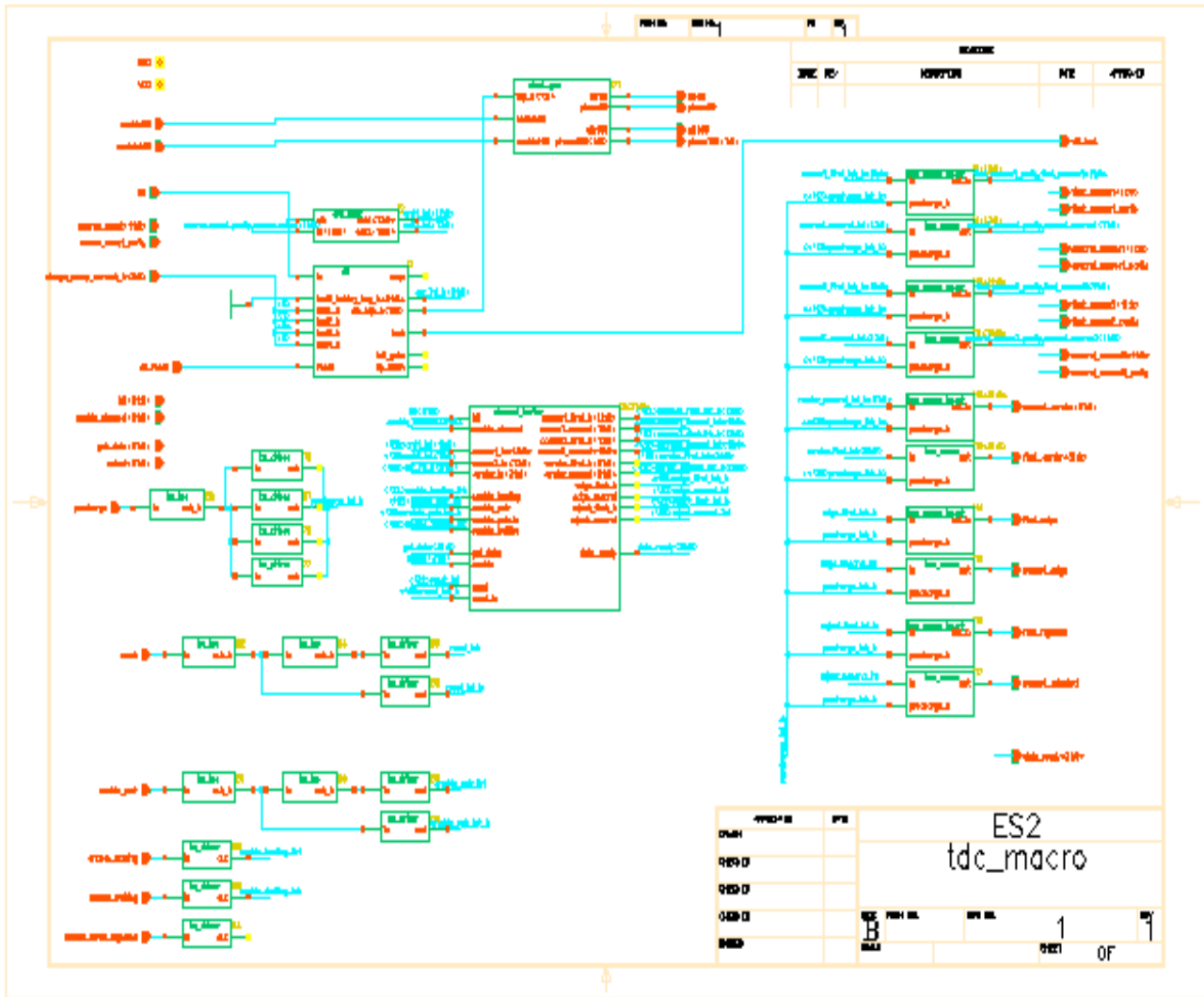
Schematic Drawing -Transistor level



Schematic Drawing - Gate level



Schematic Drawing Module level



Behavioral

- Writing behavior (text): Defines behavioral hierarchy
  Defines algorithm
  Defines architecture
- Synthesis tool required to map into gates
- Often integrated with graphical block diagram tool.

```
assign #(test.logic_delay)
bsr_clk = ~(m_extest | m_sample | m_intest) | clk_dr,
bsr_shift = (m_extest | m_sample | m_intest) & shift_dr,;
```

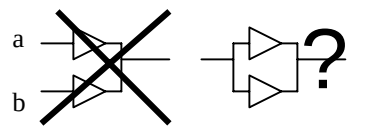
```
always @(posedge clk)
        begin
        if (set) coarse <= #(test.ff_delay) offset;
        else if (coarse == count_roll_over)
                coarse <= #(test.ff_delay) 0;
        else coarse <= #(test.ff_delay) coarse + 1;
        end
```

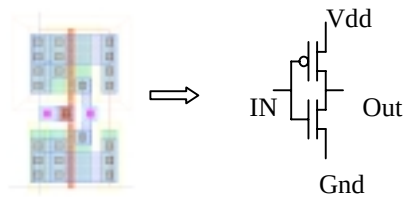
Example of behavioral specification in Verilog

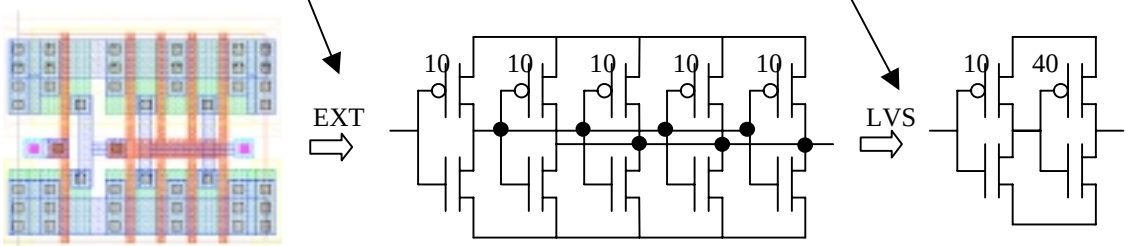
```
module add_and_mult( a,b,c, out)
input[31:0]  a,b,c;
output[31:0]   out;
wire[31:0] internal_add;

adder32      add1(a,b, internal_add);
multiplier32              mult1( internal_add, c, out);
endmodule
```

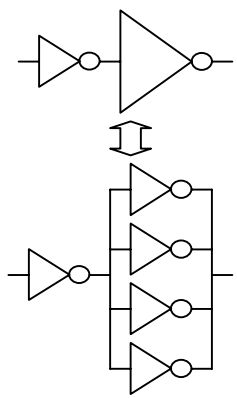
Verification

- Design Rule Check:
  Checks geometrical shapes: width, length, spacing, overlap, etc.
- Electrical rule check:
  Checks electrical circuit: unconnected inputs
  shorted outputs
  correct power and ground connection
- Extraction:
  Extracts electrical circuit: transistors, connections, capacitance, resistance
- Layout versus schematic:
  Compares electrical circuits: transistors: parallel or serial
  (schematic and extracted layout)









Types of Simulation

- Simulates behavior of designed circuit
  - Input: Models (transistor, gates, macro)
    Textual netlist (schematic, extracted layout, behavioral)
    User defined stimulus
  - Output: Circuit response (waveforms, patterns)
    Warnings
- **1.** Transistor level simulation using analog simulator (SPICE)
  - Time domain
  - Frequency domain
  - Noise

Types of Simulation (cont)

- 2. Gate level simulation using digital simulator
  - Logic functionality
  - Timing: Operating frequency, delay, setup & hold violations
- 3. Behavioral simulation
  - System and IC definition (algorithm, architecture)
  - Partitioning
  - Complexity estimation

Normally same simulator

Gate level models

- They are a border between transistor domain (analog) and digital domain
- Digital gate level models are introduced to speed up simulation.
- Gate level model contains:
  - Logic behavior
  - Delays depending on: operating conditions, loading, signal slew rates
  - Setup and hold timing violation checks
- Above gate level model parameters are extracted from transistor level simulations and characterization of real gates.

Gate data sheet

NA3

3–input NAND

Truth Table

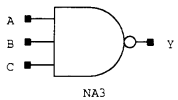
A	B	C	Y
0	X	X	1
X	0	X	1
X	X	0	1
1	1	1	0

$Y = \overline{A \text{ AND } B \text{ AND } C}$

Datasheet Version: 2.0

Code	Parameter	Value	Unit
X	Length	13.800	um
Y	Height	38.000	um
Ntran	Transistor count	6	trans
Power	AC Power dissipation	3.25	uW/MHz

Symbol



NA3

Input/Output Specifications

Input	Description	Fanin	Unit	Output	Description	Fanout	Unit
A	Data in	0.058	pF	Y	Data out	0.79	pF
B	Data in	0.057	pF				
C	Data in	0.057	pF				

Propagation Delays

Code	From	To	Min	Typ	Max	Mil	Unit
tplh	A	Y	0.10	0.20	0.41	0.47	ns
tphl	A	Y	0.06	0.13	0.25	0.29	ns
tplh	B	Y	0.08	0.17	0.35	0.40	ns
tphl	B	Y	0.08	0.16	0.32	0.38	ns
tplh	C	Y	0.06	0.13	0.26	0.30	ns
tphl	C	Y	0.09	0.19	0.37	0.43	ns
dtplh	ANY	Y	0.54	1.16	2.29	2.66	ns/pF
dtphl	ANY	Y	0.56	1.20	2.37	2.76	ns/pF

NO3S

3–input NOR

Truth Table

A	B	C	Y
1	X	X	0
X	1	X	0
X	X	1	0
0	0	0	1

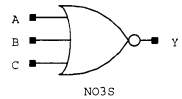
$Y = \overline{A \text{ OR } B \text{ OR } C}$

Reduced drive strength.

Datasheet Version: 2.0

Code	Parameter	Value	Unit
X	Length	14.700	um
Y	Height	38.000	um
Ntran	Transistor count	6	trans
Power	AC Power dissipation	3.40	uW/MHz

Symbol



NO3S

Input/Output Specifications

Input	Description	Fanin	Unit	Output	Description	Fanout	Unit
A	Data in	0.058	pF	Y	Data out	0.43	pF
B	Data in	0.058	pF				
C	Data in	0.058	pF				

Propagation Delays

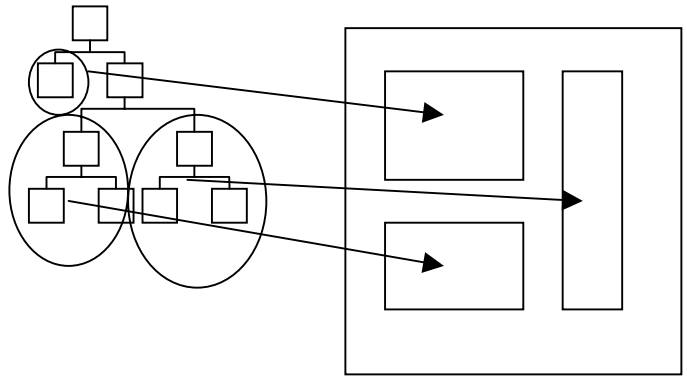
Code	From	To	Min	Typ	Max	Mil	Unit
tplh	A	Y	0.11	0.24	0.47	0.54	ns
tphl	A	Y	0.09	0.19	0.38	0.45	ns
tplh	B	Y	0.10	0.21	0.41	0.47	ns
tphl	B	Y	0.08	0.17	0.34	0.40	ns
tplh	C	Y	0.07	0.15	0.29	0.34	ns
tphl	C	Y	0.06	0.13	0.25	0.29	ns
dtplh	ANY	Y	0.96	2.05	4.05	4.71	ns/pF
dtphl	ANY	Y	0.36	0.77	1.52	1.76	ns/pF

Place and Route

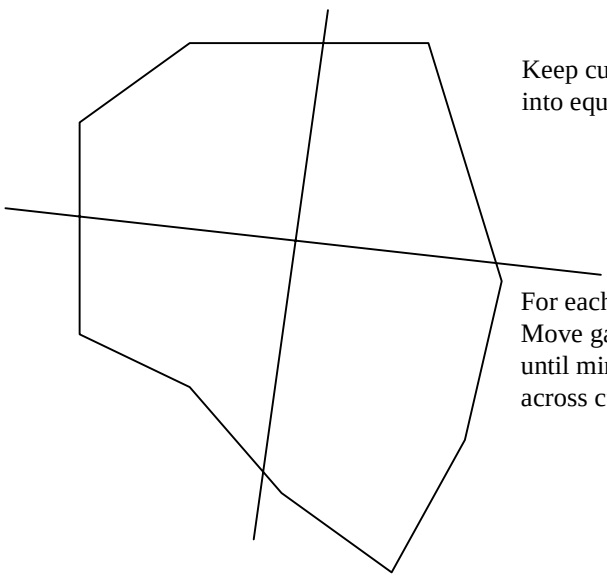
- Generates final chip from gate level netlist
  - Goals: Minimum chip size
    Maximum chip speed.
- Placement:
  - Placing all gates to minimize distance between connected gates
    - Floor planning tool using design hierarchy
    - Specialized algorithms (min cut, simulated annealing, etc.)
    - Timing driven
    - Manual intervention
  - Very compute intensive

Place and Route

Hierarchy based floor planning



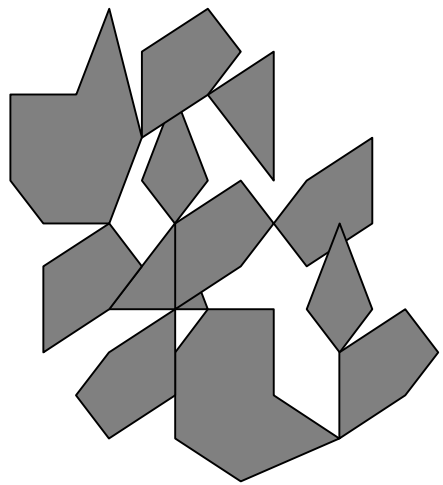
Min cut



Keep cutting design
into equal sized pieces

For each cut:
Move gates around
until minimum connection
across cut

Simulated annealing

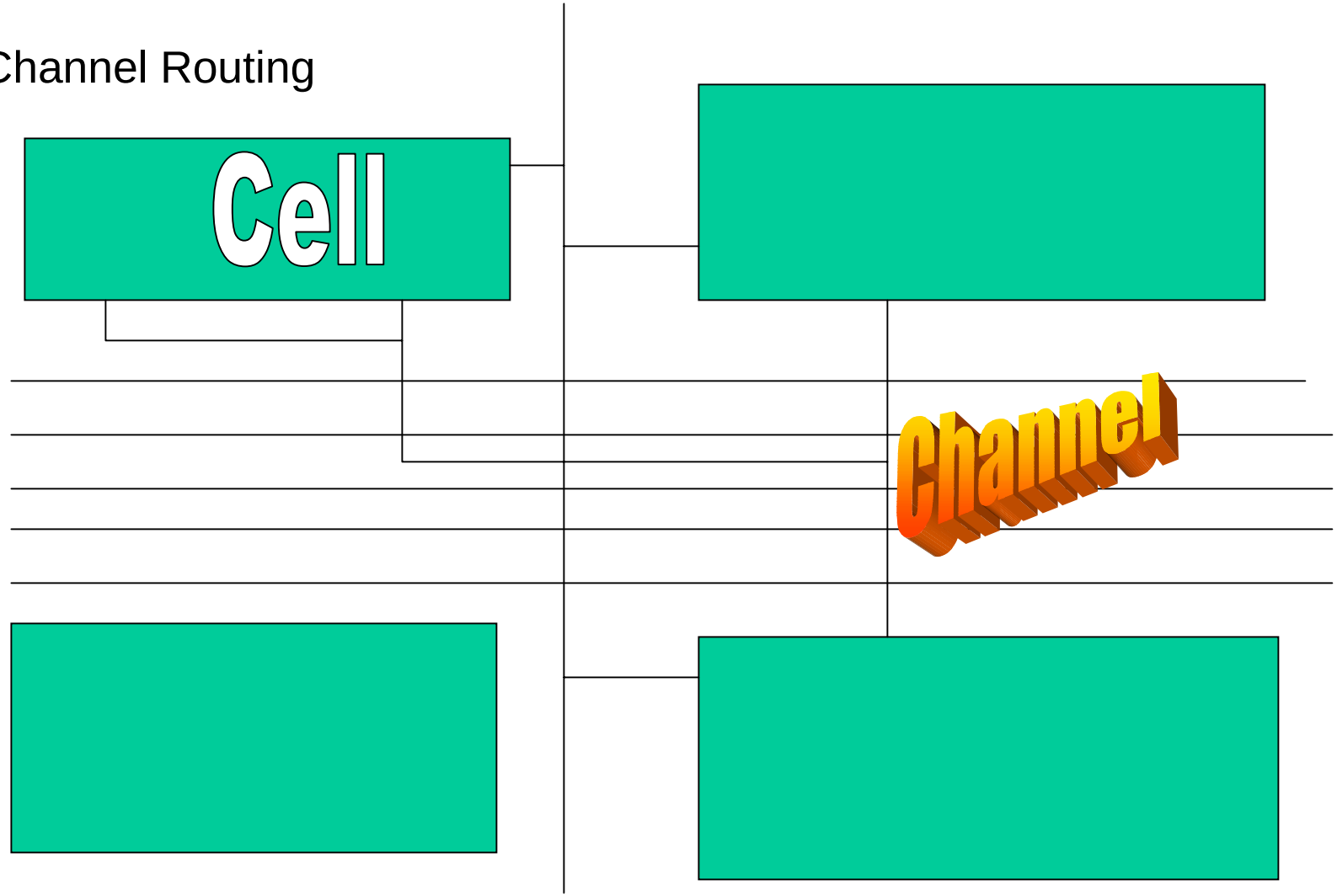


High temperature:
move gates randomly

Low temperature:
Move gates locally

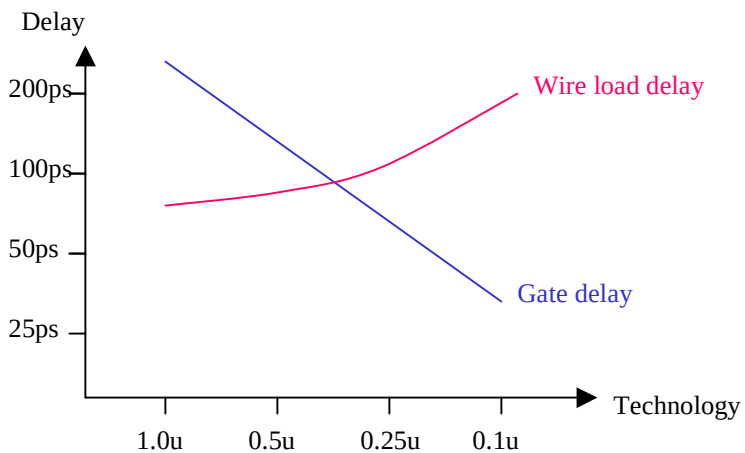
- Routing:
  - Channel based:
    - Routing only in channels between gates (few metal layers: 2)
  - Channel less:
    - Routing over gates (many metal layers: 3 - 5)
  - Often split in two steps:
    - Global route:
      - Find a coarse route depending on local routing density
    - Detailed route:
      - Generate routing layout

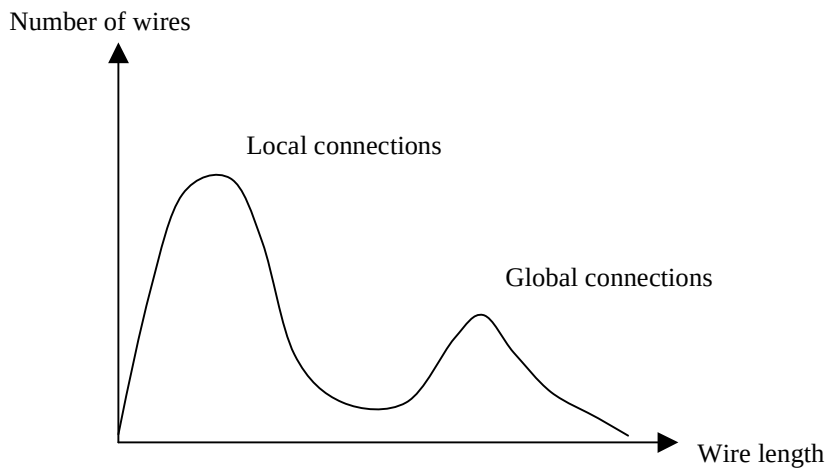
- Channel Routing



Importance of physical design

- Performance of sub-micron CMOS IC's is to a large extent determined by place & route.
  - Loading delays are bigger than intrinsic gate delays
  - Wire R-C delays start to become important in sub-micron
  - Clock distribution over complete chip gets critical at operating frequencies above 100Mhz.





Design tool framework

- Design tools from one vendor normally integrated into a framework which enables tools to exchange data.
  - Common data base
  - Automatic translation from one type to another
  - (Allows third party tools to be integrated into framework)
- Few standards to allow transport of designs between tools from different vendors.
  - VHDL and Verilog behavioral models and netlists
  - EDIF netlist, SPICE netlist for analog simulation
  - GDSII layout
  - Standard Delay Format (SDF) for gate delays.
  - Small vendors must be compatible with large vendors.

Transporting designs between tools from different vendors often cause problems

Required tools for different designs

- **FPGA**

  A: PC based schematic entry with time estimator and simple Place & route

  B: Behavioral modeling with synthesis, simulation and place & route.

- **Gate array**

  A: Schematic entry and simulation

  B: Behavioral modeling with synthesis and simulation.

  - Place and route performed by vendor

- **Full custom**

  – Layout, DRC, extraction and transistor level simulation

- **Standard cell, macro and full custom**

  – All tools described required

Source of CAE tools

- **Cadence**
  - Complete set of tools integrated into framework
- **Mentor**
  - Complete set of tools integrated into framework
- **Synopsys**
  - Power full synthesis tools
  - VHDL simulator
- **Avant!**
  - Power full place and route tools
  - Hspice simulator with automatic characterization tools
- **Diverse commercial**:
  - View-logic, Summit, Tanner, etc.

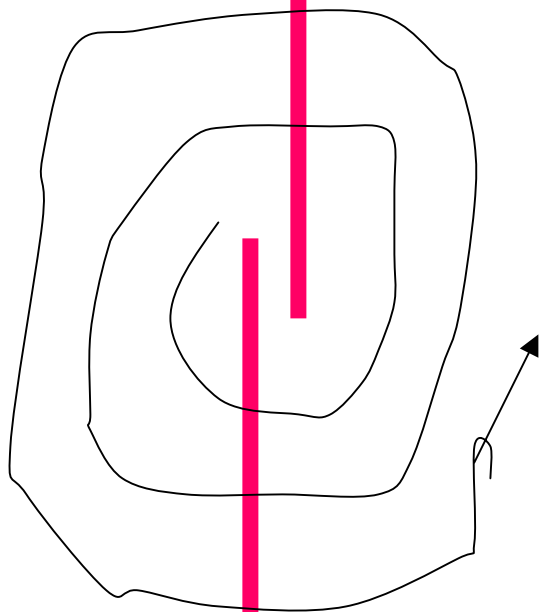
- **Free shareware**:
  - Spice, Magic, Berkley IC design tools, Aliance
  - Diverse from the web.
- Complete set of commercial high performance CAE tools cost ~1 M$ per seat ! (official list price).
- University programs: Complete set of tools ~10K$
  - Europe: Eurochip
  - US: Mosis
  - Japan:?

- **Design methodologies**:
  - Top-down design (military)
  - Bottom-up (revolutions in middle ages)
  - Inside-out (political parties)
- **Top-down** (from High-level description to layout
- **Bottom-up** (from cells to blocks to systems)
- **Inside-out** (from register-transfer simultaneously up to system description and down to cells and layout

Top Down

Bottom Up

Inside - Out

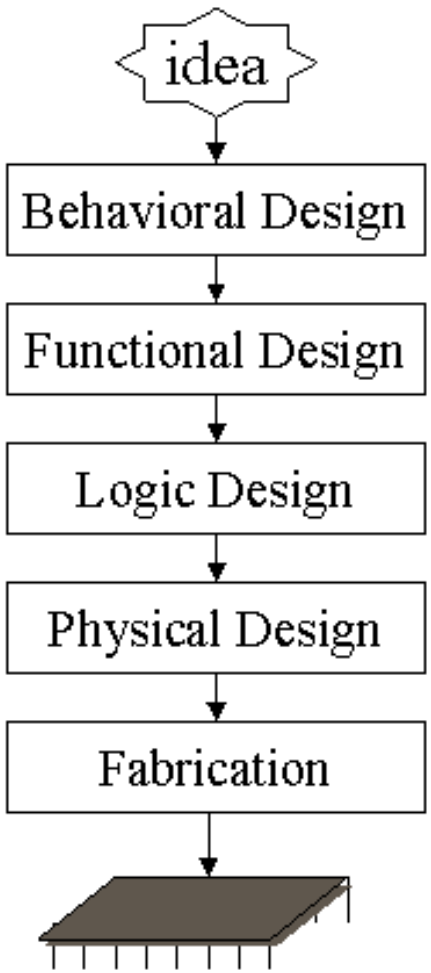


System Level Description

Register Transfer Level Description

Logic Cell Level Description

Example of Top-Down Design Flow



Architectural Design High- level design
- Flow chart, Pseudo code, Algorithmic description

Data Path Design, RTL(Register Transfer Level) Design
- Bus, Register

Gate Level Design
- NAND, NOR, INV, AND, OR, XOR etc.

Transistor Level Design, Circuit Level Design
- Transistor, Capacitor, Resistor

Manufacturing

Design Styles

- Full- custom layout
- Macro- cell (Building block) layout
- Standard- cell layout
- Gate- array layout
- PLA (Programmable Logic Array)
- FPGA (Field Programmable Gate Array)

Now we will illustrate these styles

Full- custom layout

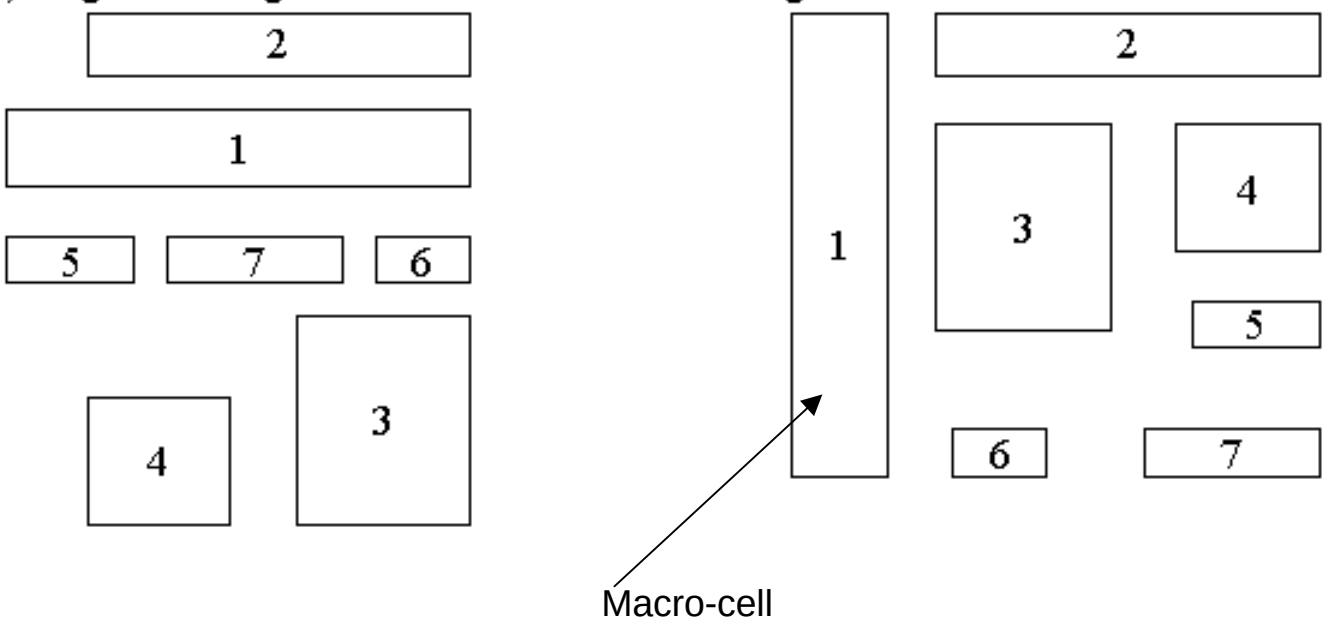
- Manual layout design using layout editor
- time- consuming & difficult
- Custom chip high performance

  Ex) Microprocessor
- layout editor
  - DRC (Design Rule Check)
    - width rule Ex) metal line min 4 μm
    - space rule Ex) wire
  - λ- based design rule

- **ASIC**

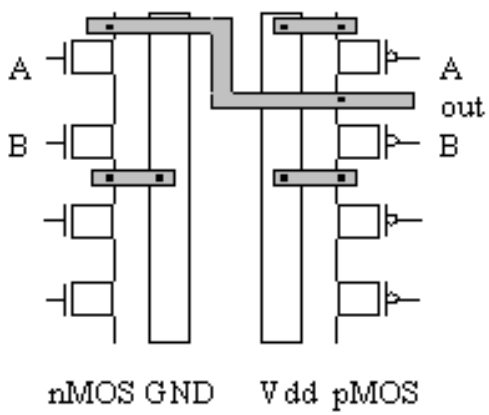
Macro- cell layout

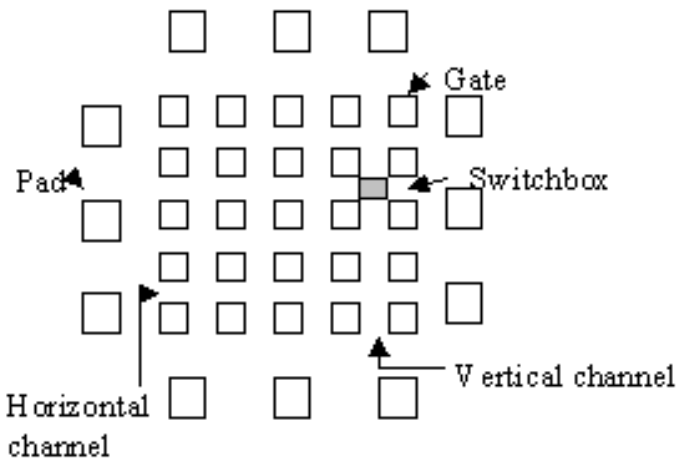
- Macro- cell design style (Building- block design style)
  : permit cells to vary in both dimensions
- Advantage
  : cells of significant complexity are permitted in the library
  Ex) register, register files, arithmetic- logic units, memories etc .



Gate array layout

- **Gate- array (Mask Programmed Gate- Arrays (MPGAs))**
  - A large of number of transistors which have been prefabricated on a wafer in the form of a regular two- dimensional array
- **Personalizing**
  - process of adding metal wires to a gate array
  - Two types of interconnection
    - intra- cell wiring & inter- cell wiring
- **Sea of gate (channel- less gate- arrays)**
  - Routed over the transistors



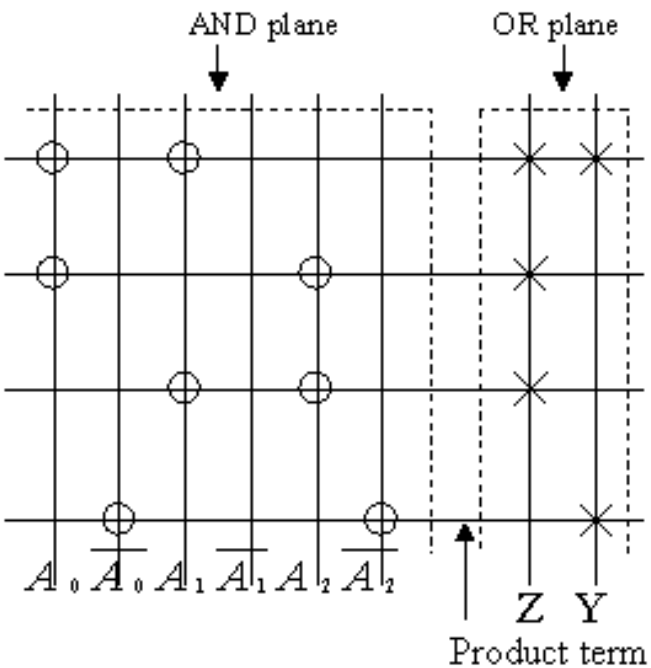


PLA(Programmable Logic Array)

- AND plane & OR plane
- Fixed architecture
- Easy to automate the generation of PLA layouts

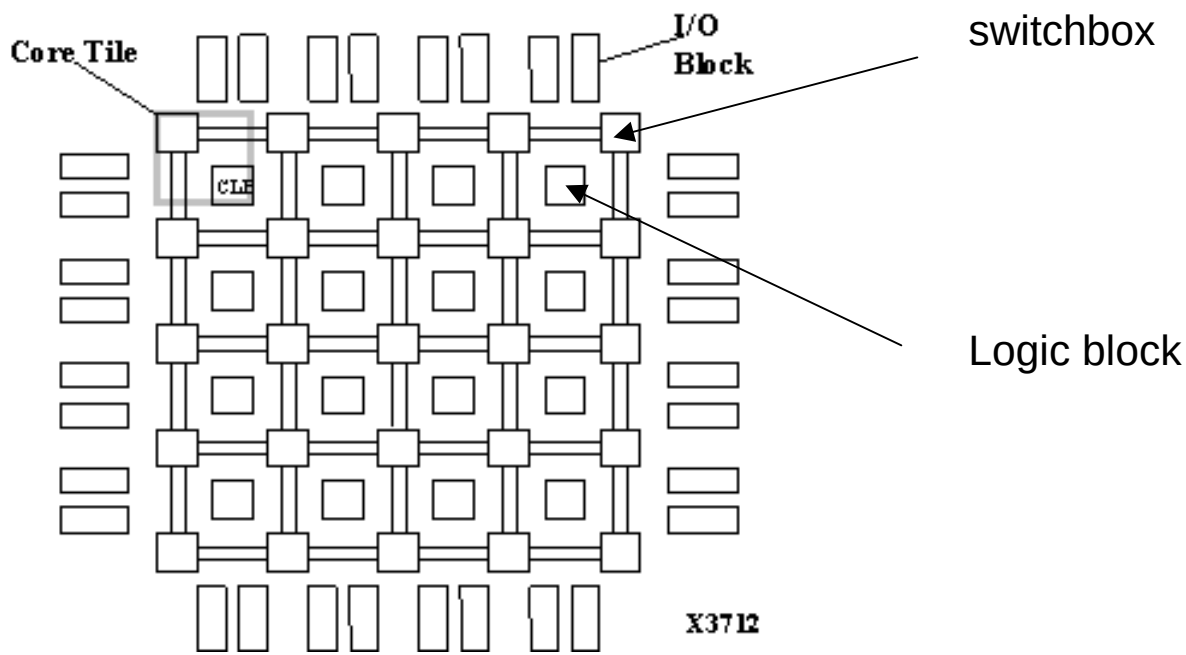
Ex) $Z = A_0 \cdot A_1 + A_0 \cdot A_2 + A_1 \cdot A_2$

$Y = A_0 \cdot A_1 + \overline{A_1} \cdot \overline{A_2}$



FPGA (Field-Programmable Gate Array)

- A two- dimensional array of logic blocks
- CLBs (Configurable Logic Blocks)
  - Can be programmed to implement any logic function
- Interconnect
  - Contain programmable switches



FPGA (Field-Programmable Gate Array)

- Design steps
  1. Technology mapping
  2. Placement
  3. Routing
  4. Generate the bit patterns
- Advantage
  - Lower prototyping cost & shorter production times
- Disadvantage
  - Lower speed of operations & lower gate density
  - Expensive for mass- production

- Class projects can be of this type

What have we learnt?

- Tools for layout, simulation, design creation and synthesis
- Many level of synthesis
- Designer of circuits should understand the role and function of all tools
- Designer of tools should understand the problems that occur in design, and also to same extent the technology

Sources

- Christian from CERN
- Korean slides