IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 01: INTRODUCTION

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

About the Course – VLSI Physical Design

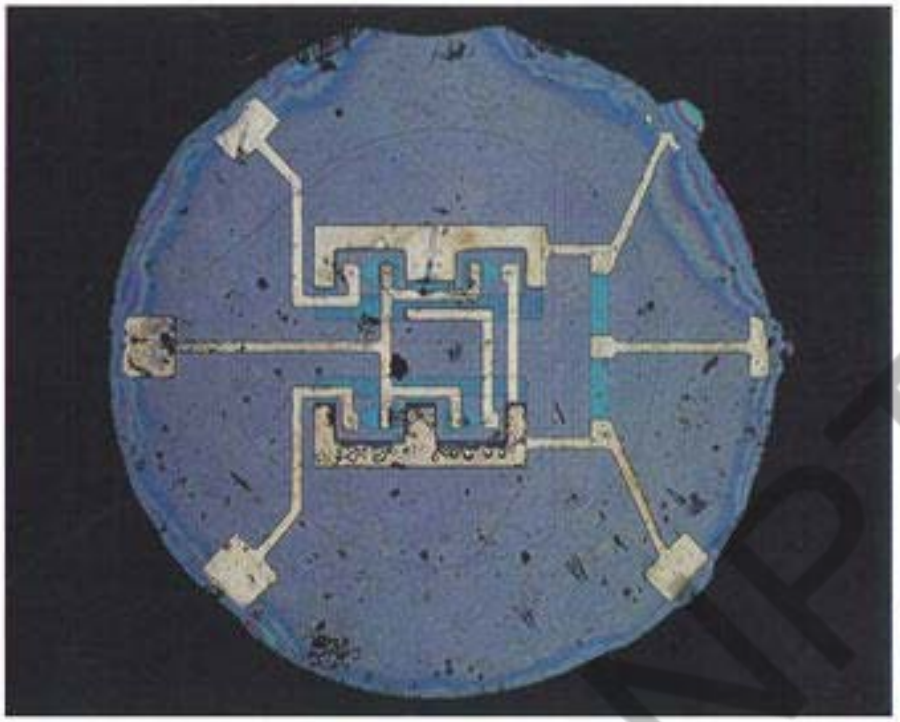
- Module 1: Introduction to physical design automation
- Module 2: Floorplanning and Placement
- Module 3: Routing
- Module 4-5: Static timing analysis
- Module 6: Signal integrity and crosstalk issues
- Module 7-8: Clocking issues and clock tree synthesis
- Module 9-10: Low power design issues
- Module 11: Noise analysis and layout compaction
- Module 12: Physical verification and sign-off

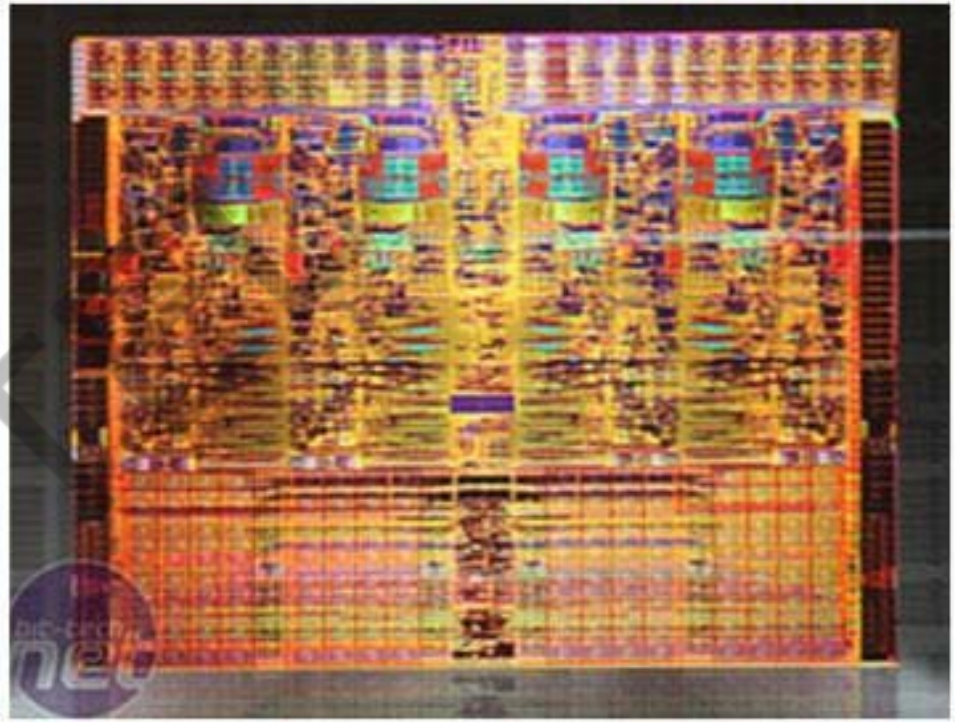
Main Focus of the Course

1. Digital design
   - Many of the techniques discussed also apply to analog and mixed signal circuits.
2. Physical design automation
   - Starting with a circuit netlist, how to move down up to the layout level?

Digital Design Process

- Design complexity increasing rapidly
  - Increased size and complexity
  - Fabrication technology improving
  - CAD tools are essential
  - Too many CAD tools to choose from
- The present trend
  - Standardize the design flow
  - Emphasis on low-power design, and increased performance

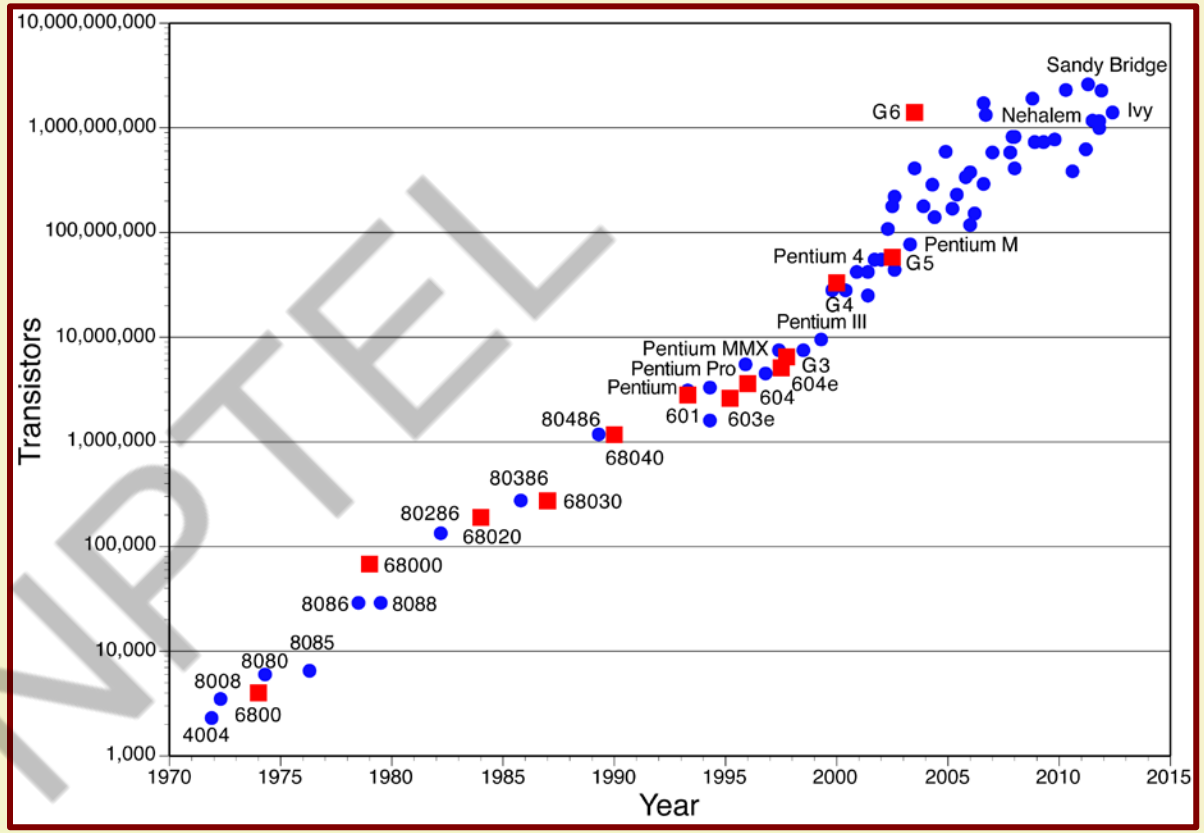


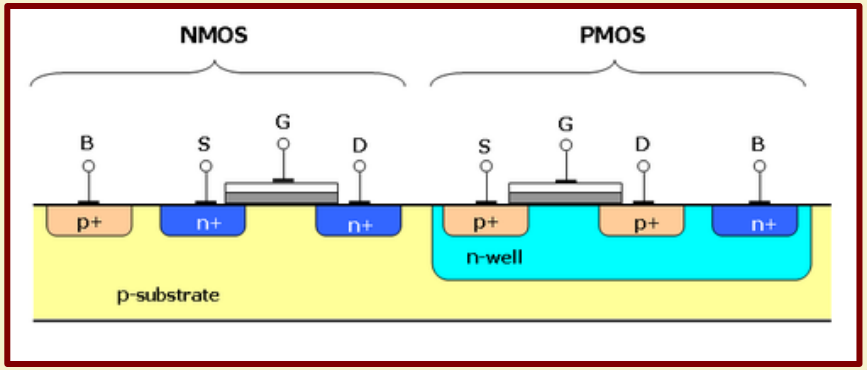


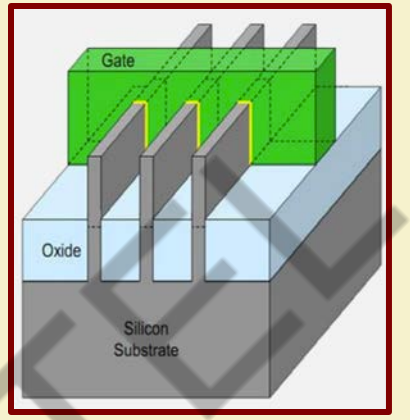
First Planar IC (1961) and Intel Nehalem Quad Core Die

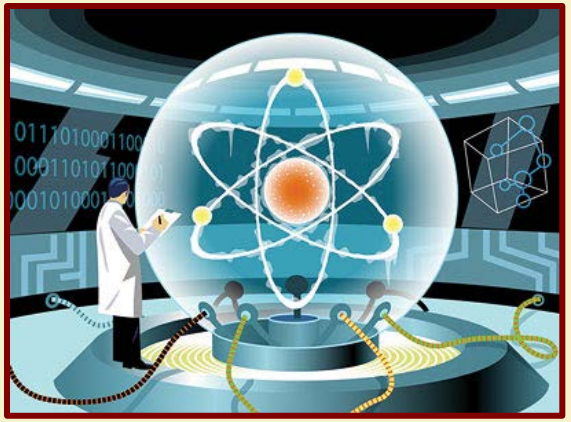
Moore's Law

- Exponential growth
- Design complexity increases rapidly
- Automated tools are essential
- Must follow well-defined design flow









CMOS
(up to 22nm)

FinFET
(14nm)

QUANTUM?

What is design flow?

- Standardized design procedure
  - Starting from the design idea down to the actual implementation.
- Encompasses many steps:
  - Specification
  - Synthesis
  - Simulation
  - Layout
  - Testability analysis
  - and many more ……

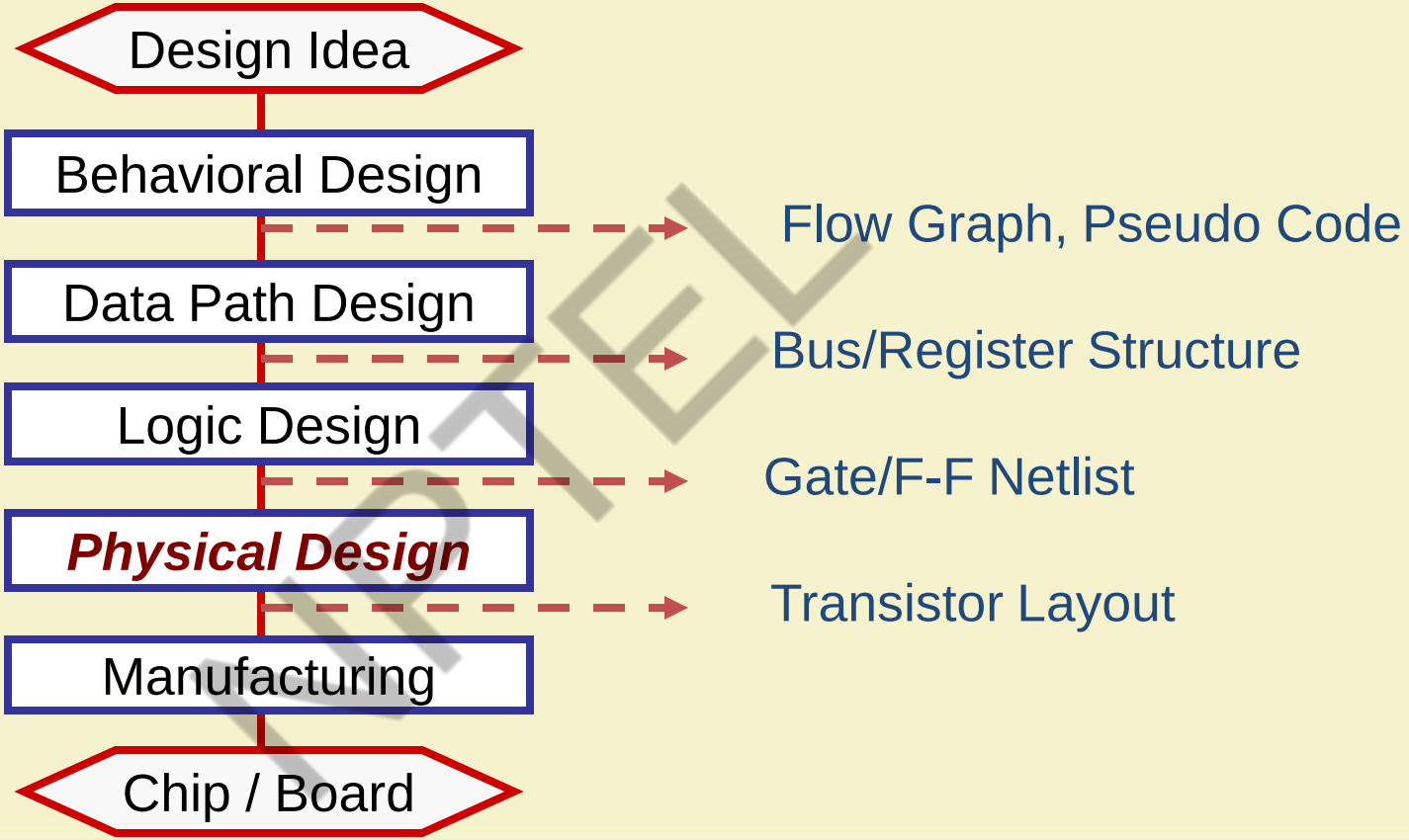
- Need to use Computer Aided Design (CAD) tools.
  - Based on Hardware Description Language (HDL).
  - HDLs provide formats for representing the outputs of various design steps.
  - CAD tool transforms its HDL input into a HDL output that contains more hardware information.
    - Behavioral level to register transfer level
    - Register transfer level to gate level
    - Gate level to transistor level (***PHYSICAL DESIGN***)

Two Competing HDLs

1. Verilog
2. VHDL

Designs are created typically using HDLs, which get transformed from one level of abstraction to the other as the design flow progresses.

Simplistic View of Design Flow



Steps in the Design Flow

- Behavioral design
  - Specify the functionality of the chip
- Data path design
  - Generate a netlist of register transfer level components
- Logic design
  - Generate a netlist of gates/flip-flops or standard cells
- Physical design
  - Generate the final layout
- Manufacturing
  - Fabricate the chip

Other Steps in the Design Flow

- Simulation for verification
  - At various levels: logic level, switch level, circuit level
- Formal verification
  - Used to verify the designs through formal techniques
- Testability analysis and Test pattern generation
  - Required for testing the manufactured devices

END OF LECTURE 01

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

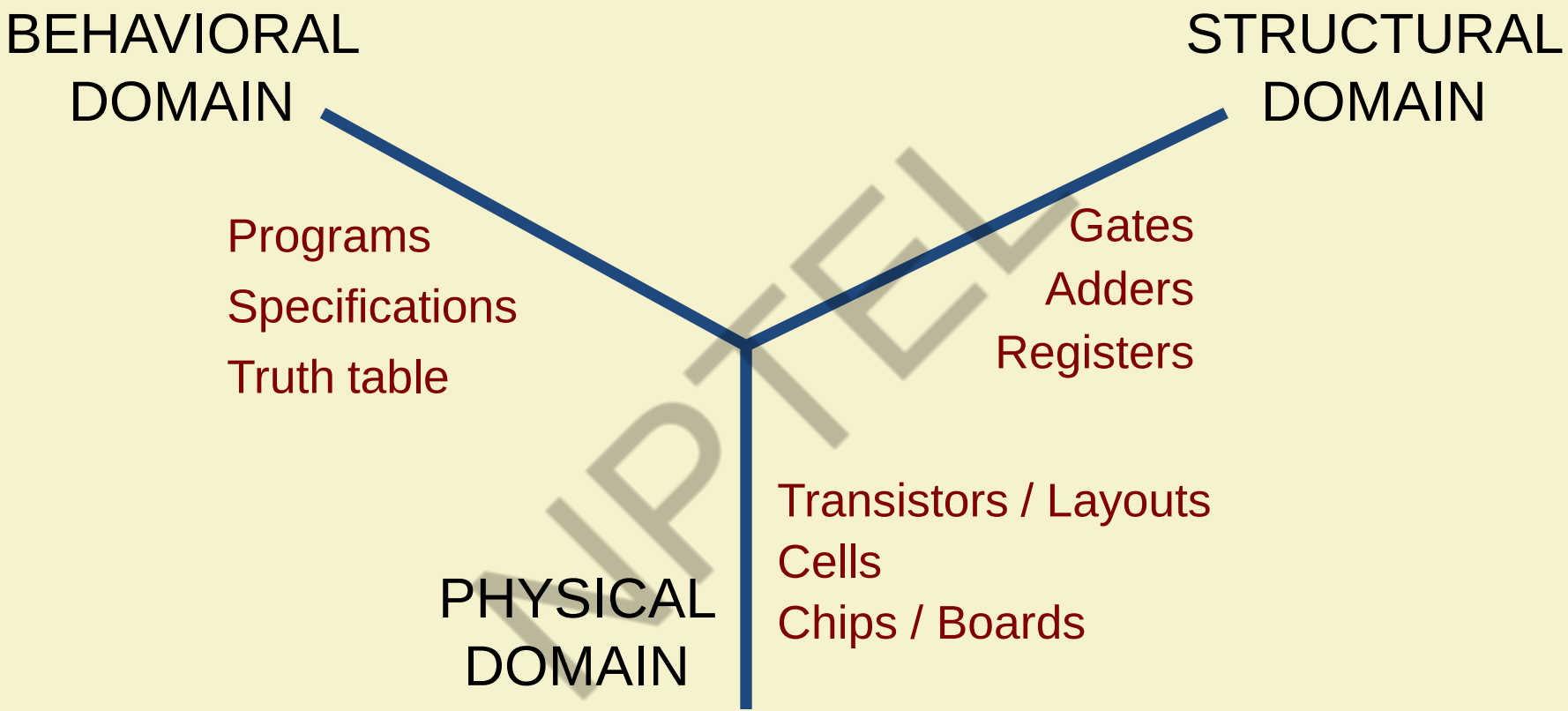
Lecture 02: DESIGN REPRESENTATION

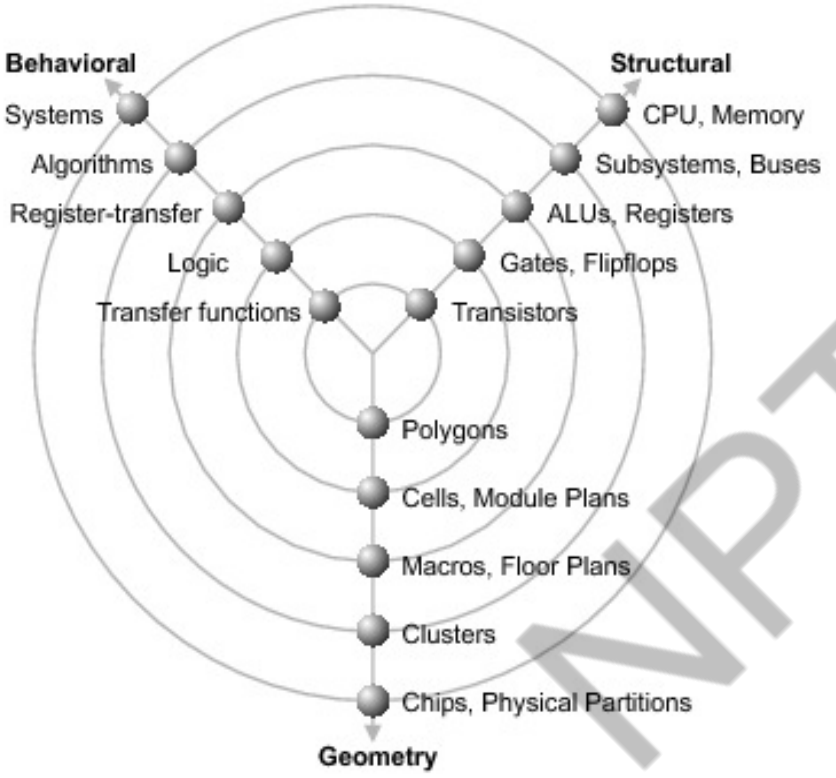
PROF. INDRANIL SENGUPTA

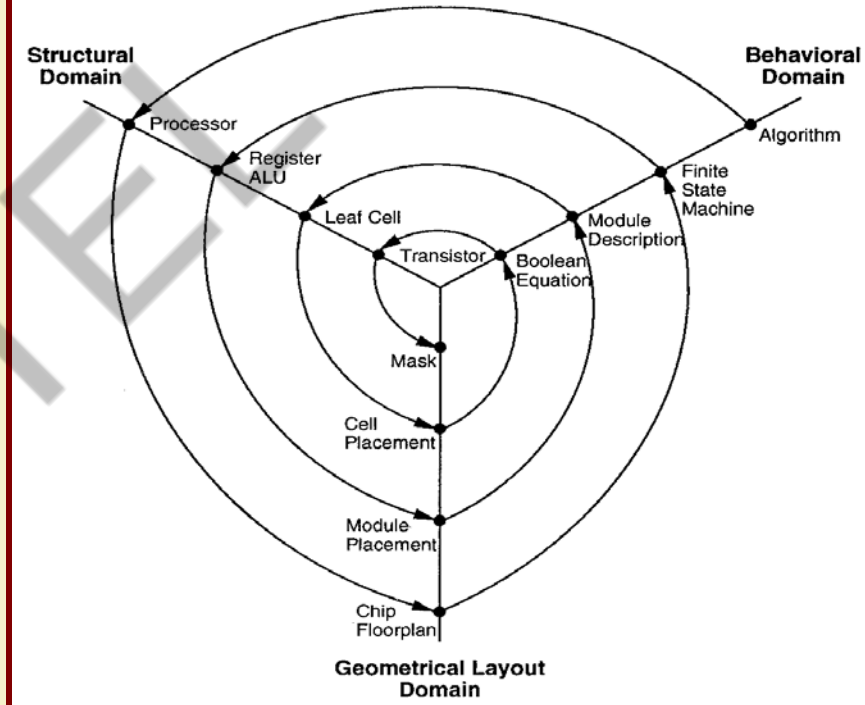
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Design Representation

- A design can be represented at various levels from three different angles:
    1. Behavioral
    2. Structural
    3. Physical
- Can be represented by Y-diagram.


BEHAVIORAL DOMAIN
STRUCTURAL DOMAIN
Programs
Specifications
Truth table
Gates
Adders
Registers
PHYSICAL DOMAIN
Transistors / Layouts
Cells
Chips / Boards


Behavioral
Structural
Systems
CPU, Memory
Algorithms
Subsystems, Buses
Register-transfer
ALUs, Registers
Logic
Gates, Flipflops
Transfer functions
Transistors
Polygons
Cells, Module Plans
Macros, Floor Plans
Clusters
Chips, Physical Partitions
Geometry


Structural Domain
Behavioral Domain
Processor
Register ALU
Algorithm
Finite State Machine
Leaf Cell
Module Description
Transistor
Boolean Equation
Mask
Cell Placement
Module Placement
Chip Floorplan
Geometrical Layout Domain

Behavioral Representation

- Specifies how a particular design should respond to a given set of inputs.
- May be specified by:
  - Boolean equations
  - Tables of input and output values
  - Algorithms written in standard HLL like C
  - Algorithms written in special HDL like Verilog or VHDL

Behavioral Representation :: Example

A n-bit adder constructed by cascading n 1-bit adders:

A 1-bit adder has

- two operand inputs A and B
- a carry input C
- a carry output Cy
- a sum output S

$S = A.B'.C' + A'.B'.C + A'.B.C' + A.B.C = A \text{ xor } B \text{ xor } C$

$Cy = A.B + A.C + B.C$

An algorithmic level description of Cy

```
module  carry (cy, a, b, c);
    input a, b, c;
    output cy;
    assign
        cy = (a&b) | (b&c) | (c&a);
endmodule
```

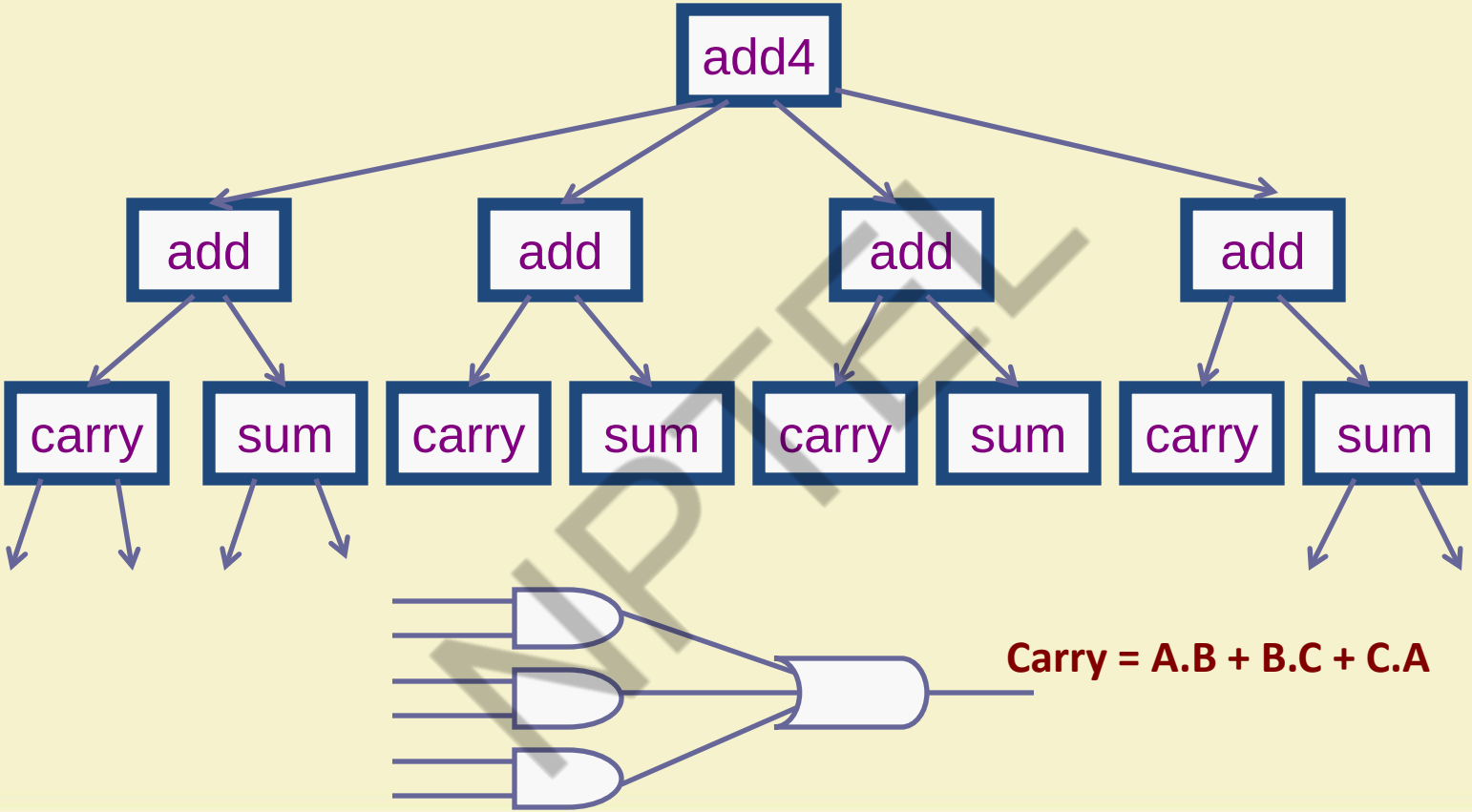
Boolean behavioral specification for Cy

```
primitive  carry (cy, a, b, c);
        input a, b, c;
        output cy;
        table
          //  a  b  c    cy
              1  1  ?  :  1 ;
              1  ?  1  :  1 ;
              ?  1  1  :  1 ;
              0  0  ?  :  0 ;
              0  ?  0  :  0 ;
              ?  0  0  :  0 ;
      endtable
endprimitive
```

Structural Representation

- Specifies how components are interconnected.
- In general, the description is a list of modules and their interconnects:
  - called *netlist*.
  - can be specified at various levels.

- At the structural level, the levels of abstraction are:
  - the module (functional) level
  - the gate level
  - the switch level
  - the circuit level
- In each successive level more detail is revealed about the implementation.



Carry = A.B + B.C + C.A

Structural Representation :: Example

4-bit adder

```
module  add4 (s, cy4, cy_in, x, y);
        input [3:0] x, y;
        input cy_in;
        output [3:0] s;
        output cy4;
        wire [2:0]  cy_out;
            add  B0  (cy_out[0], s[0], x[0], y[0], ci);
            add  B1  (cy_out[1], s[1], x[1], y[1], cy_out[0]);
            add  B2  (cy_out[2], s[2], x[2], y[2], cy_out[1]);
            add  B3  (cy4, s[3], x[3], y[3], cy_out[2]);
 endmodule
```

```
module  add  (cy_out, sum, a, b, cy_in);
    input a, b, cy_in;
    output  sum, cy_out;
        sum   s1 (sum, a, b, cy_in);
        carry  c1 (cy_out, a, b, cy_in);
 endmodule
```

```
module  sum  (sum, a, b, cy_in);
    input a, b, cy_in;
    output  sum;
    wire t;
        xor   x1 (t, a, b);
        xor   x2 (sum, t, cy_in);
endmodule
```

```
module carry (cy_out, a, b, cy_in);
    input a, b, cy_in;
    output  cy_out;
    wire  t1, t2, t3;
        and  g1 (t1, a, b);
        and  g2 (t2, a, c);
        and  g3 (t3, b, c);
        or    g4 (cy_out, t1, t2, t3);
endmodule
```

Physical Representation

- The lowest level of physical specification.
  - Photo-mask information required by the various processing steps in the fabrication process.
- At the module level, the physical layout for the 4-bit adder may be defined by a rectangle or polygon, and a collection of ports.

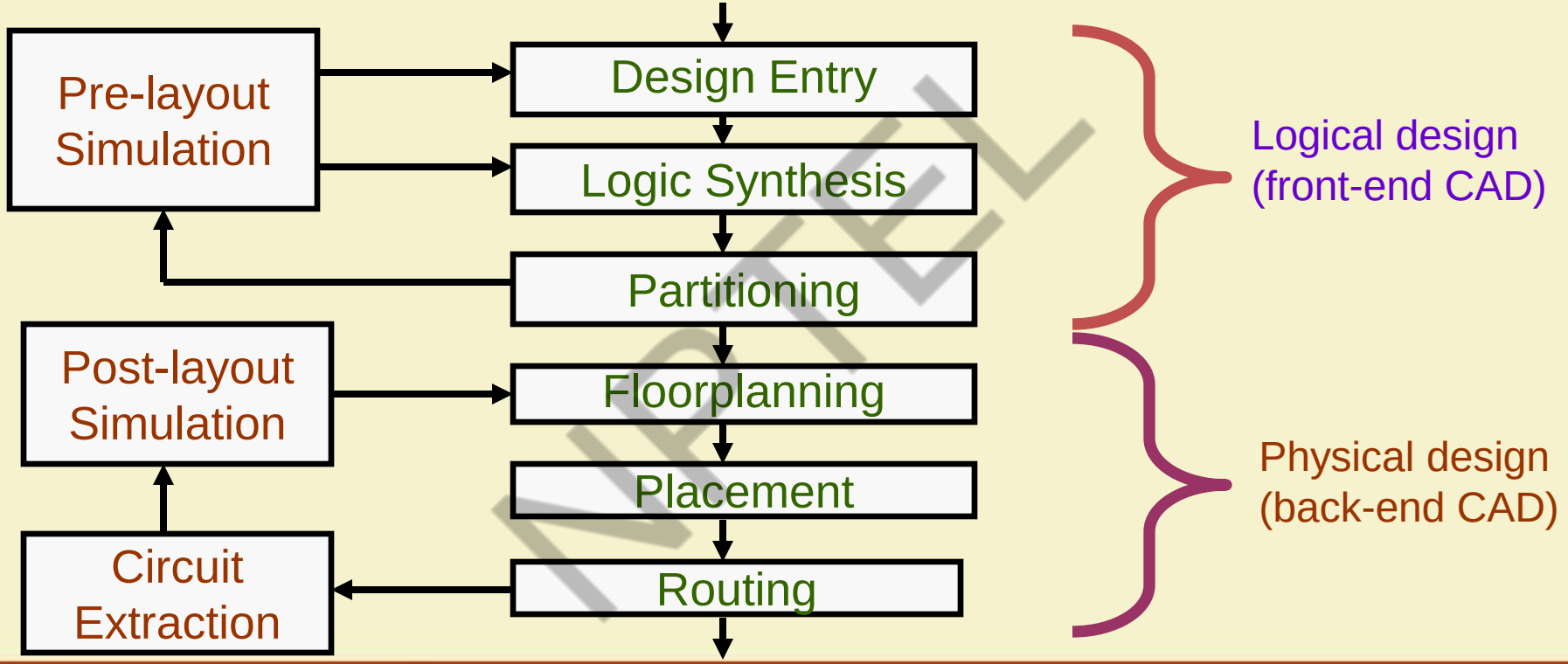
Physical Representation :: Example

A possible (partial) physical description for 4-bit adder

```
module  add4;
     input x[3:0], y[3:0], cy_in;
     output s[3:0], cy4;
     boundary [0, 0, 130, 500];
     port  x[0]     aluminum  width = 1 origin = [0, 35];
     port  y[0]     aluminum  width = 1 origin = [0, 85];
     port   cy_in  polysilicon width = 2 origin = [70, 0];
     port  s[0]     aluminum  width = 1  origin = [120, 65];

     add  a0  origin  =  [0, 0];
     add  a1  origin  =  [0, 120];
 endmodule
```

Digital IC Design Flow: A quick look



END OF LECTURE 02

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

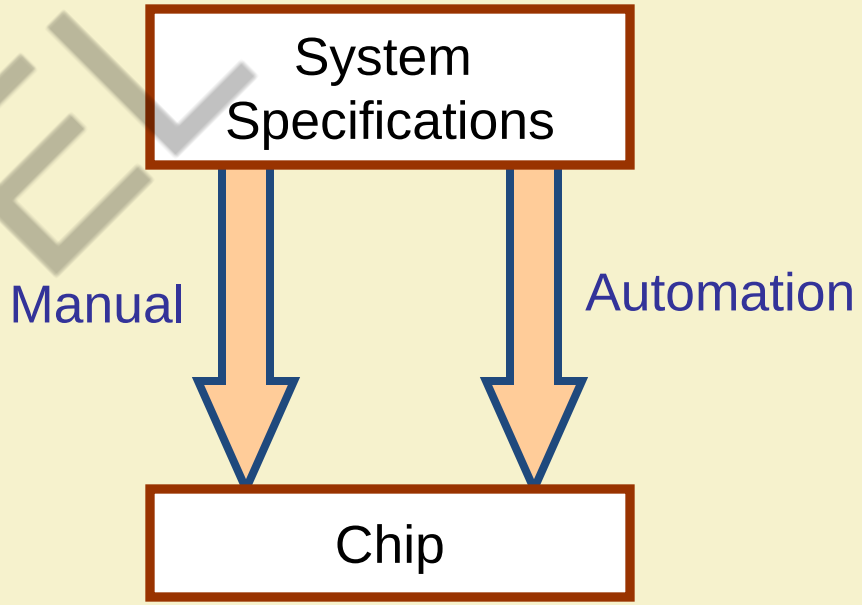
Lecture 03: VLSI DESIGN STYLES (PART 1)

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VLSI Design Cycle

- Large number of devices
- Optimization requirements for high performance
- Time-to-market competition
- Cost



VLSI Design Cycle (contd.)

1. System specification
2. Functional design
3. Logic design
4. Circuit design
5. ***Physical design***
6. Design verification
7. Fabrication
8. Packaging, testing, and debugging

Physical Design

- Converts a circuit description into a geometric description.
  - This description is used for fabrication of the chip.
- Basic steps in the physical design cycle:
  1. Partitioning, floorplanning and placement
  2. Routing
  3. Static timing analysis
  4. Signal integrity and crosstalk analysis
  5. Physical verification and signoff

Various Design Styles

- Programmable Logic Devices
  - Field Programmable Gate Array (FPGA)
  - Gate Array
- Standard Cell (Semi-Custom Design)
- Full-Custom Design

Which Design Style to Use?

- Basically a tradeoff among several design parameters.
  - Hardware cost
  - Circuit delay
  - Time required
- Optimizing on these parameters is often conflicting.

Field Programmable Gate Array (FPGA)

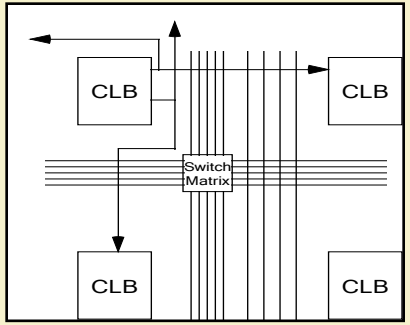
Introduction

- User / Field Programmability.
  - Array of logic cells connected via routing channels.
  - Different types of cells:
    - Special I/O cells.
    - Logic cells (Mainly lookup tables (LUT) with associated registers).
  - Interconnection between cells:
    - Using SRAM based switches.
    - Using antifuse elements.

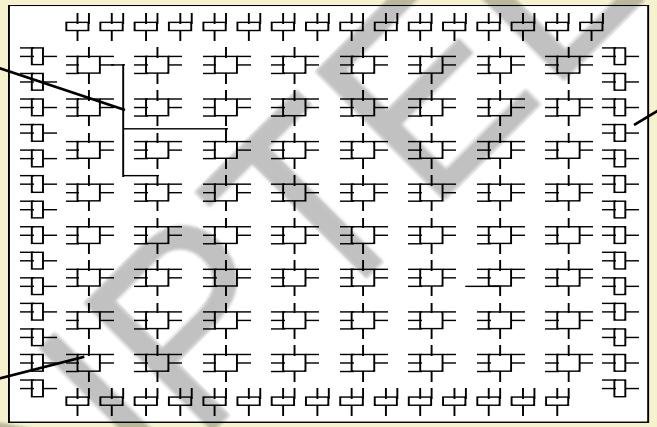
Availability

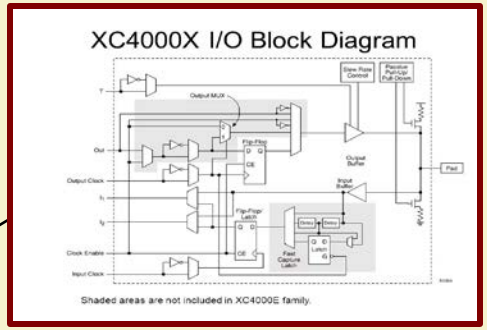
- FPGA chips are manufactured by a number of vendors:
  - Xilinx, Altera, Actel, etc.
  - Products vary widely in capability.
- FPGA development boards and CAD software available from many sellers.
  - Allows rapid prototyping in laboratory.

Xilinx XC4000 Architecture

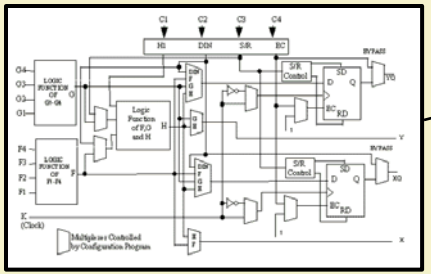


Programmable Interconnect



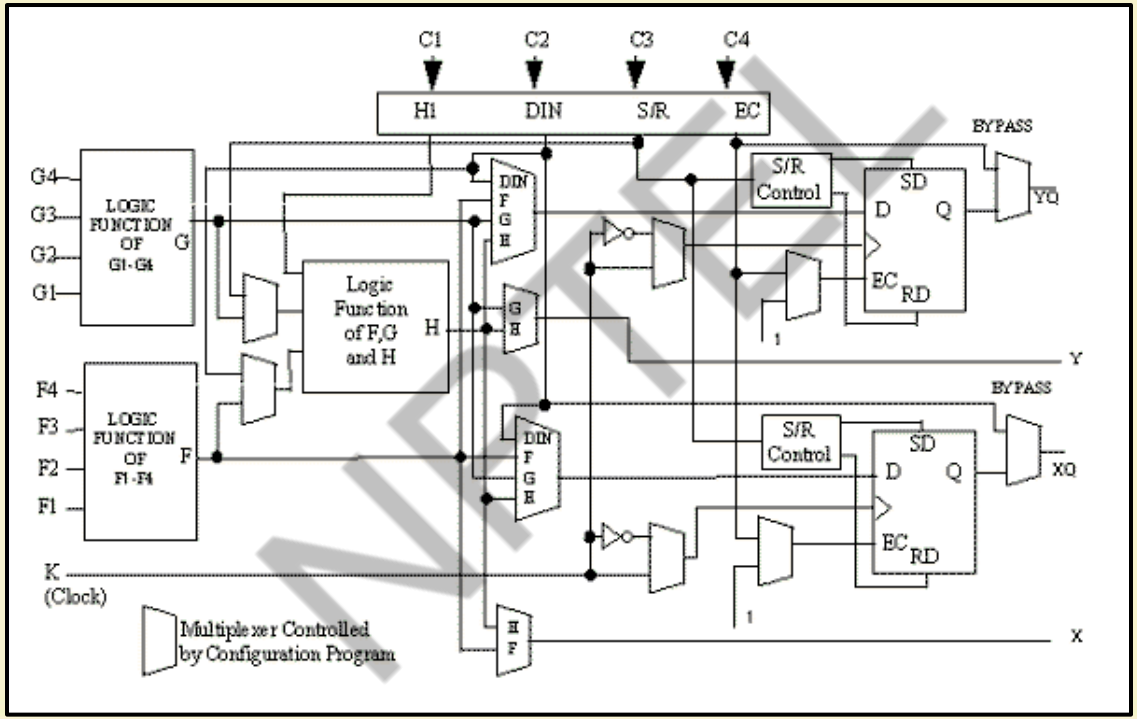


I/O Blocks (IOBs)



Configurable Logic Blocks (CLBs)

XC4000E Configurable Logic Block

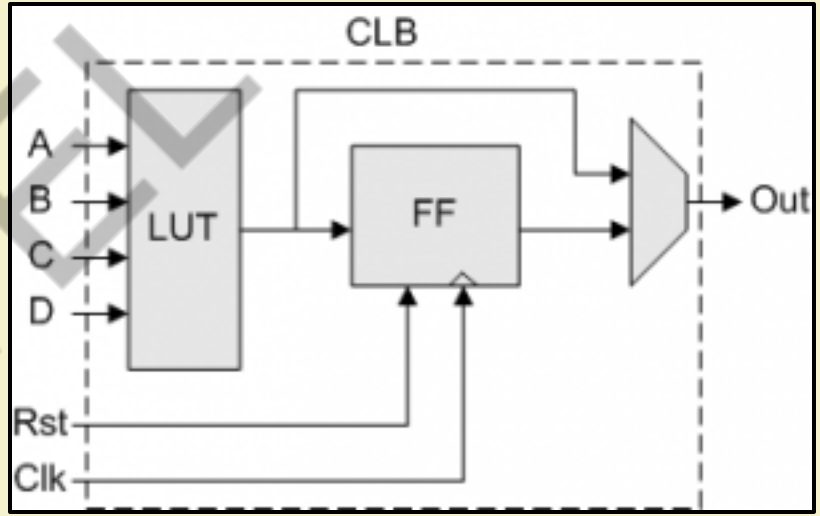


CLB Functionalities

- Two 4-input function generators
  - Implemented using Lookup Tables using 16x1 RAM.
  - Can also implement 16x1 memory.
- Two 1-bit registers
  - Each can be configured as flip-flop or latch.
  - Independent clock polarity.
  - Synchronous and asynchronous Set / Reset.

Look Up Tables (LUT)

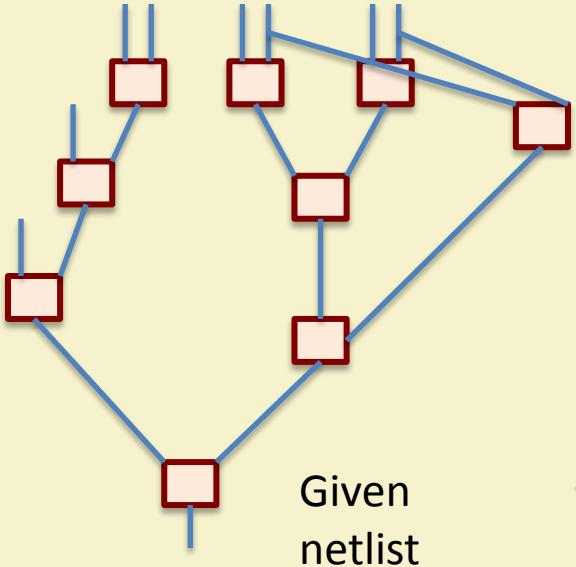
- Combinatorial Logic is stored in 16x1 SRAM Look Up Tables (LUTs) in a CLB.
- Capacity is limited by number of inputs, not complexity.
- Choose to use each function generator as 4-input logic (LUT) or as high-speed RAM.



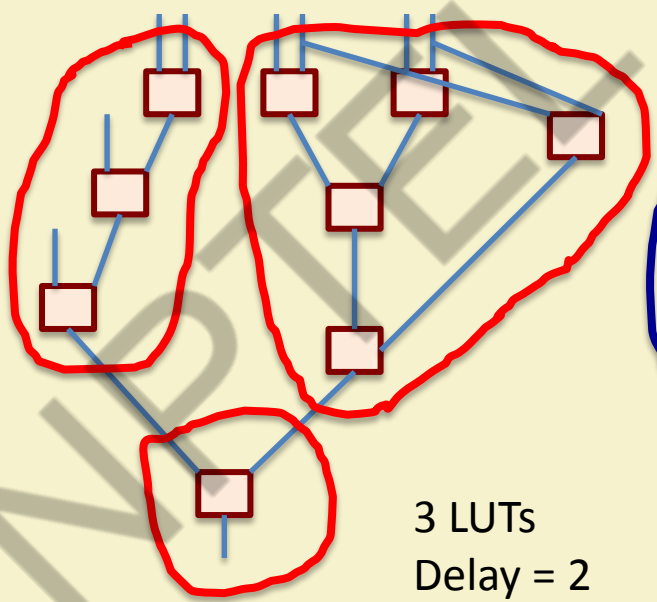
LUT Mapping: An Example

- A function: $f = A'.B + B'.C.D$
- The mapping process:
  - Create the truth table of the 4-variable function.
  - Load the output column into the SRAM corresponding to the LUT.
  - Apply the function inputs to the LUT inputs.
- Any 4-variable function can be realized.
  - Netlist to LUT mapping is an interesting design tradeoff.

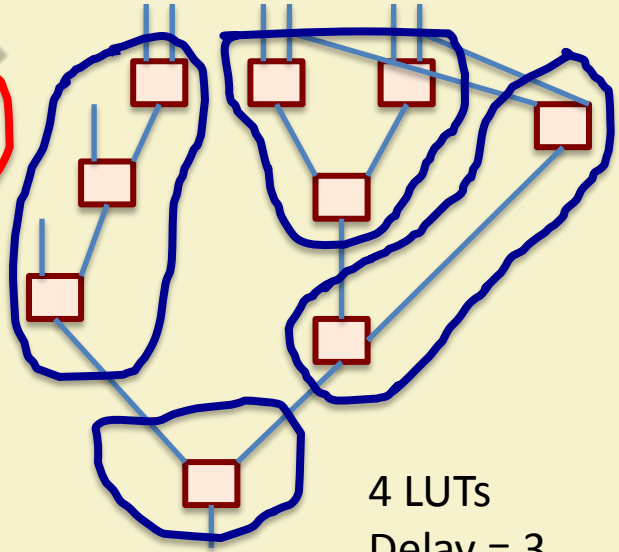
Area - Delay Tradeoff



Given
netlist

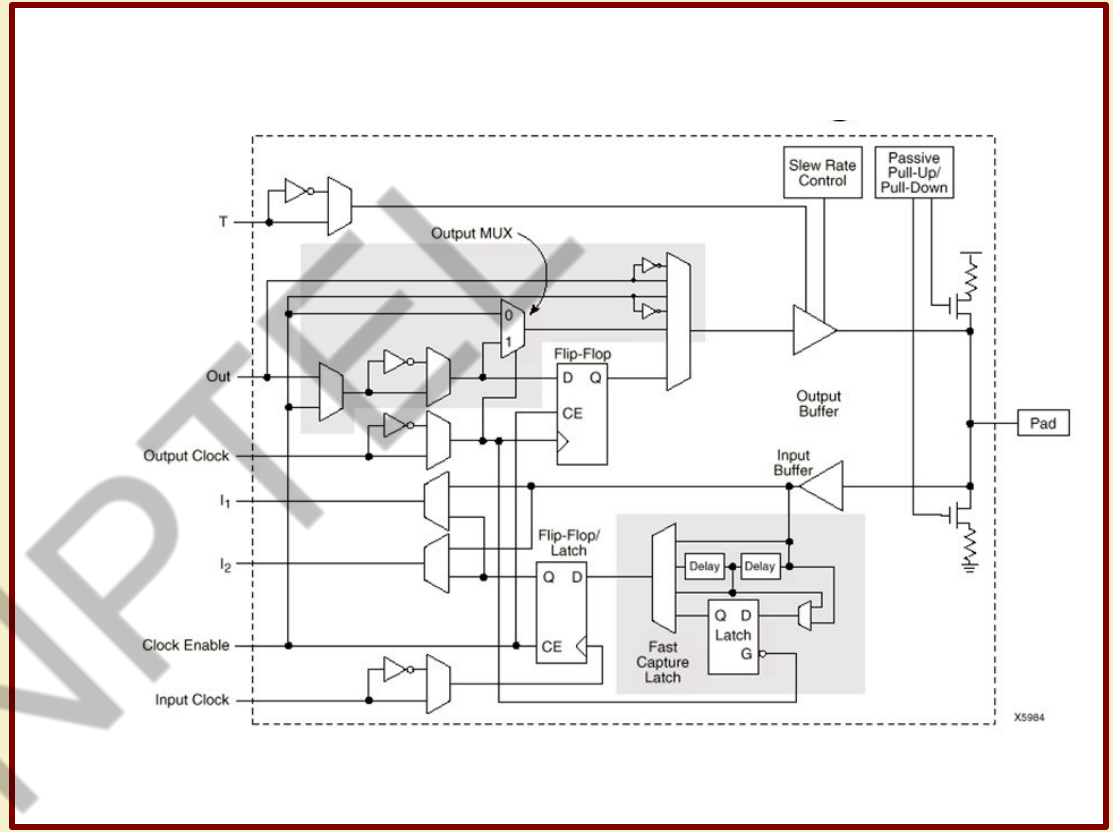


3 LUTs
Delay = 2



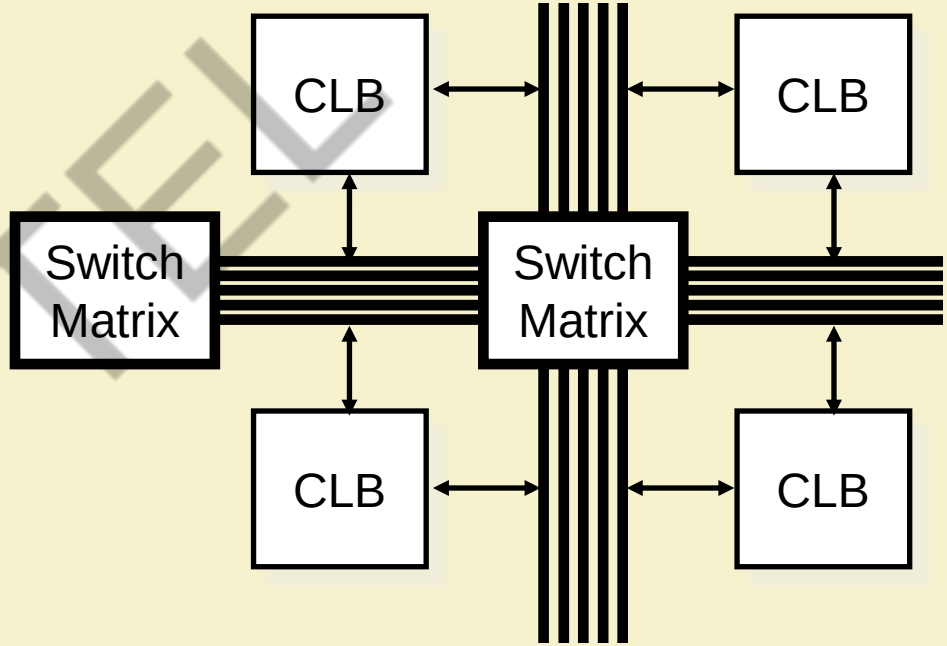
4 LUTs
Delay = 3

XC4000X I/O Block Diagram



Xilinx FPGA Routing

1. Fast Direct Interconnect - CLB to CLB
2. General Purpose Interconnect - Uses switch matrix



FPGA Design Flow

- Design Entry
  - In schematic, VHDL, or Verilog.
- Implementation
  - Placement & Routing
  - Bitstream generation
  - Analyze timing, view layout, simulation, etc.
- Download
  - Directly to Xilinx hardware devices with unlimited reconfigurations.

END OF LECTURE 03

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 04: VLSI DESIGN STYLES (PART 2)

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

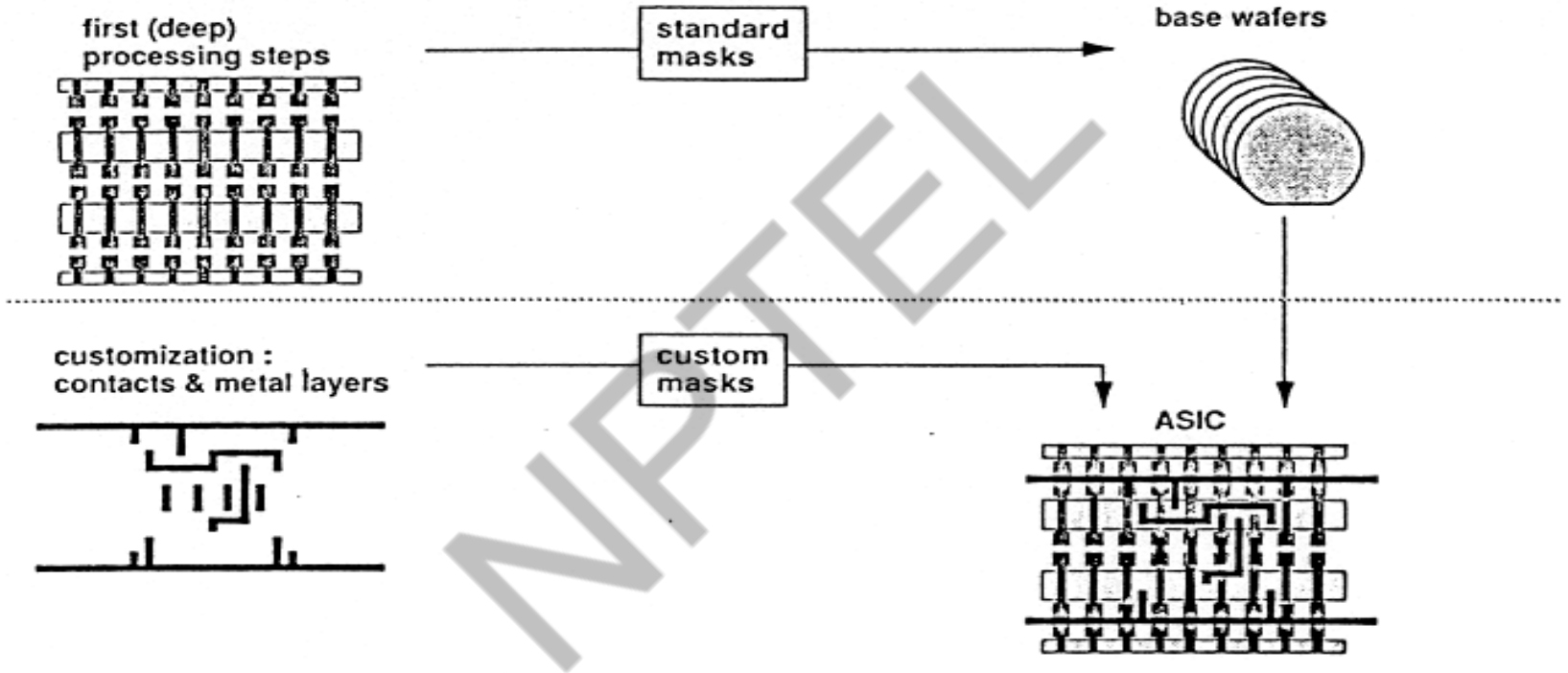
Gate Array

Introduction

- In view of the speed of prototyping capability, the gate array (GA) comes after the FPGA.
- Design implementation of
  - FPGA chip is done with user programming,
  - Gate array is done with metal mask design and processing.

- Gate array implementation requires a two-step manufacturing process:
  a) The first phase, which is based on generic (standard) masks, results in an array of uncommitted transistors on each GA chip.
  b) These uncommitted chips can be customized later, which is completed by defining the metal interconnects between the transistors of the array.

two-step manufacture :



- The GA chip utilization factor is higher than that of FPGA.
  - The used chip area divided by the total chip area.
- Chip speed is also higher.
  - More customized design can be achieved with metal mask designs.
- Typical gate array chips can implement millions of logic gates.

Standard Cell Based Design

Introduction

- One of the most prevalent design styles.
  - Also called semi-custom design style.
  - Requires developing full custom mask set.
- Basic idea:
  - Commonly used logic cells are developed, and stored in a standard cell library.
  - Typical library may contain a few hundred cells (*Inverters, NAND gates, NOR gates, AOI gates, OAI gates, 2-to-1 MUX, D-latches, flip-flops, etc.*).

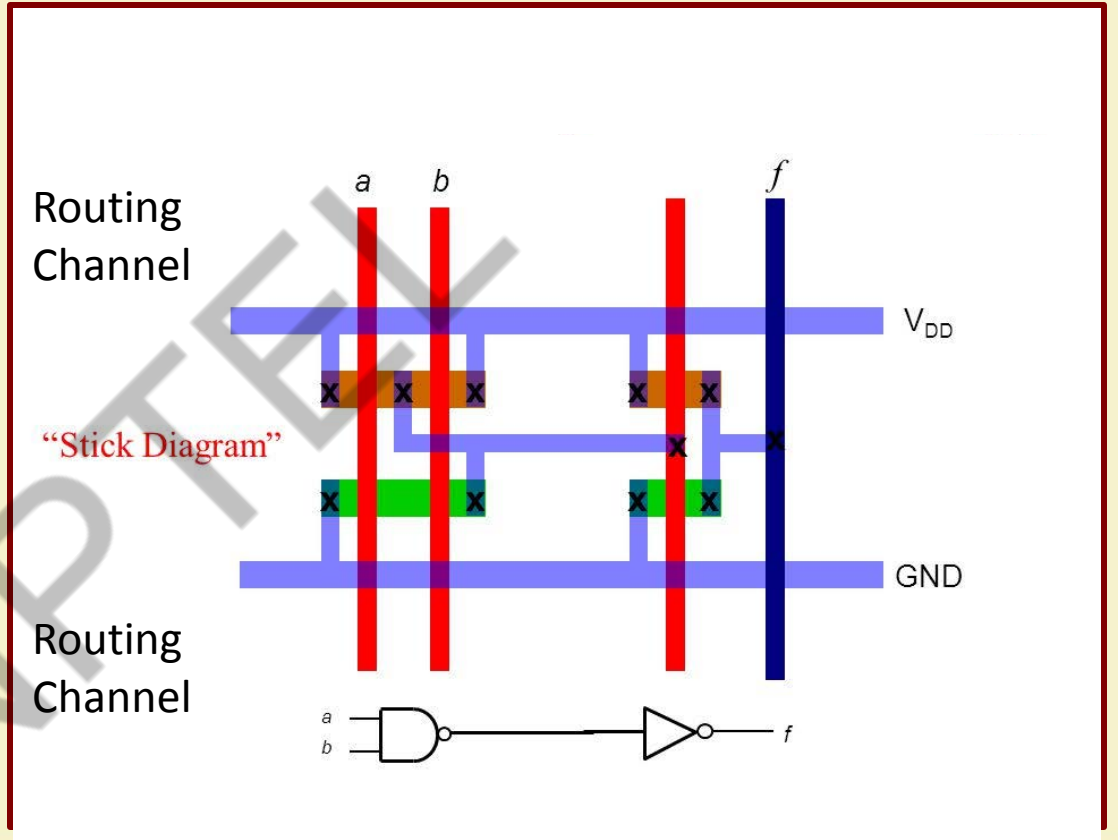
Characteristic of the Cells

- Each cell is designed with a fixed height.
  - To enable automated placement of the cells, and routing of inter-cell connections.
  - A number of cells can be abutted side-by-side to form rows.
- The power and ground rails typically run parallel to upper and lower boundaries of cell.
  - Neighboring cells share a common power and ground bus.
- The input and output pins are located on the upper and lower boundaries of the cell.

Standard Cell Example

Made to stack side by side

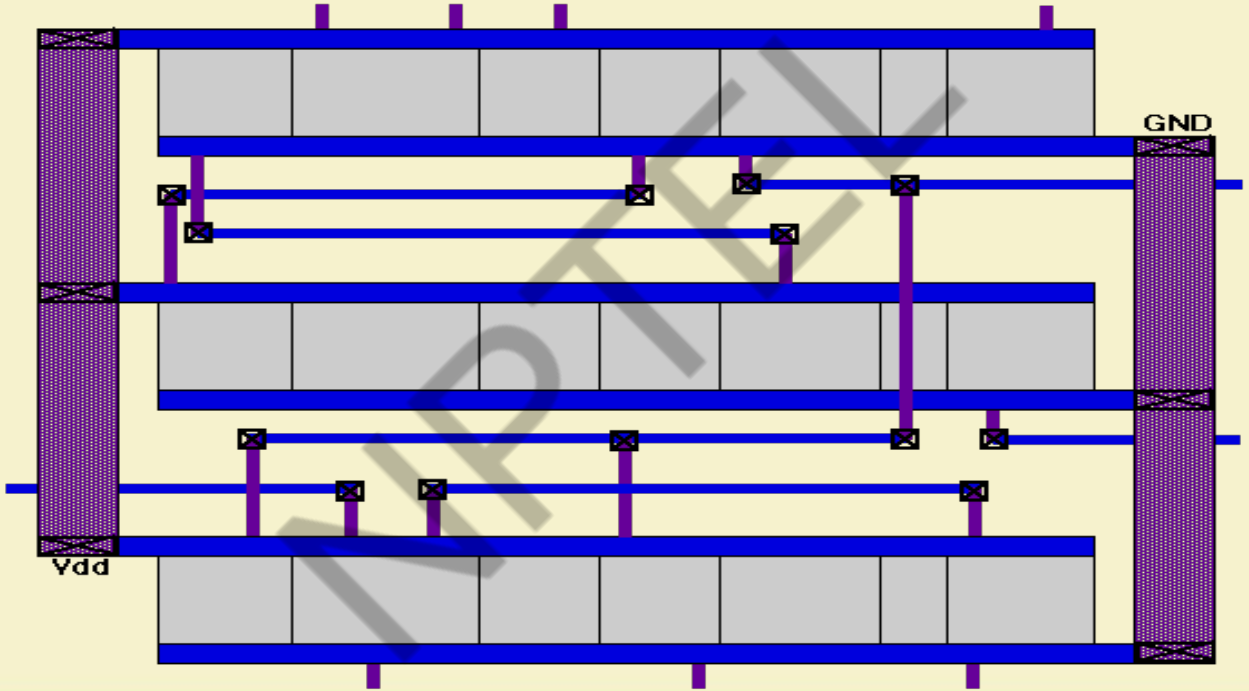
- Fixed height
- Width can vary
- Can abut at V_{DD} and GND



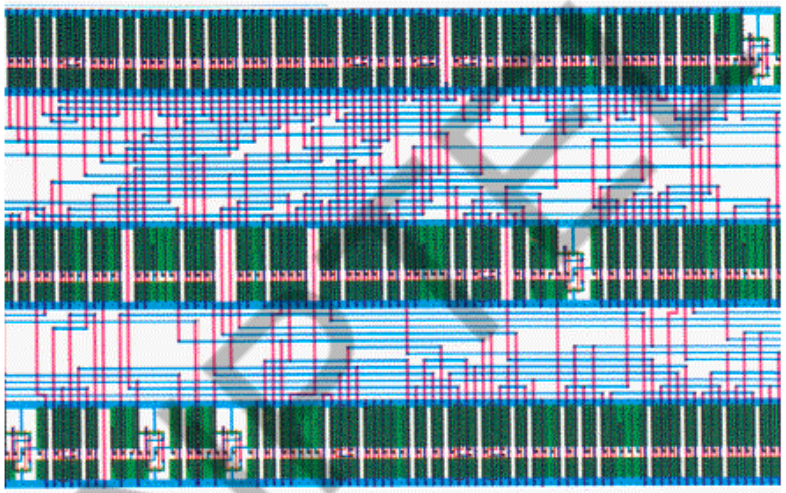
Floorplan for Standard Cell Design

- Inside the I/O frame which is reserved for I/O cells, the chip area contains rows or columns of standard cells.
  - Between cell rows are channels for routing.
  - Over-the-cell routing is also possible.
- The physical design and layout of logic cells ensure that
  - When placed into rows, their heights match.
  - Neighboring cells can abut side-by-side, which provides natural connections for power and ground lines in each row.

Standard Cell Layout



Standard Cell Layout



Full Custom Design

Introduction

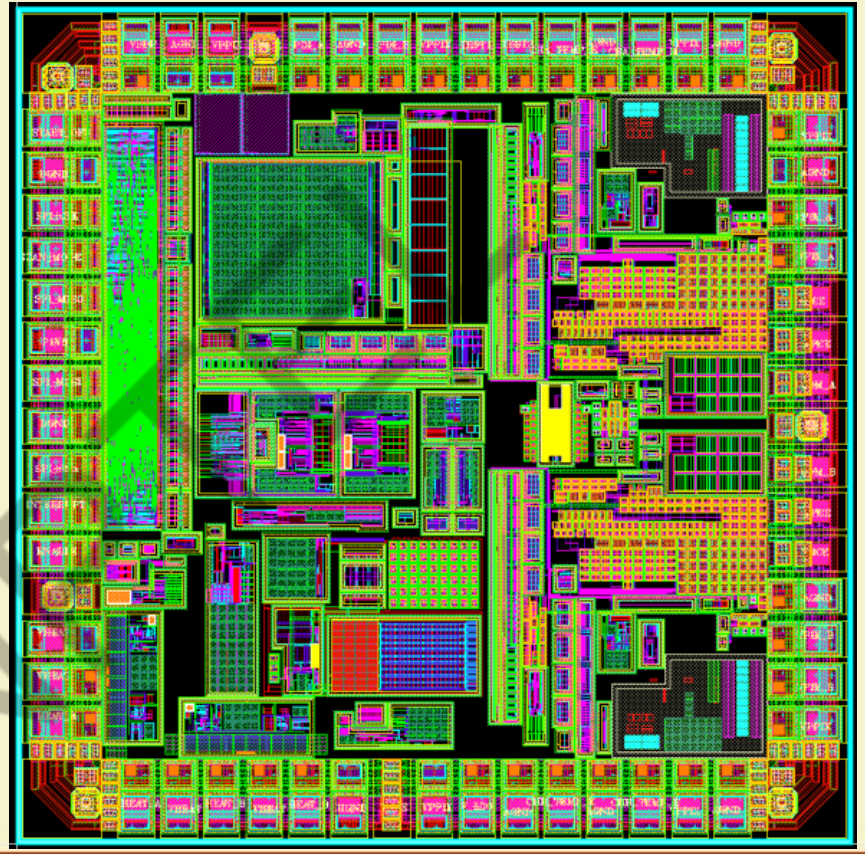
- Standard-cells based design is often called semi custom design.
  - The cells are pre-designed for general use.
- In the full custom design, the entire mask design is done anew without use of any library.
  - The development cost of such a design style is prohibitively high.
  - The concept of design reuse is becoming popular to reduce design cycle time and cost.

Contd.

- The most rigorous full custom design can be the design of a memory cell.
  - Static or dynamic.
  - Since the same layout design is replicated, there would not be any alternative to high density memory chip design.
- For logic chip design, a good compromise can be achieved by using a combination of different design styles on the same chip.
  - Standard cells, data-path cells and PLAs.

- In real full-custom layout in which the geometry, orientation and placement of every transistor is done individually by the designer.
  - Design productivity is usually very low (typically 10 to 20 transistors per day, per designer).
- In digital CMOS VLSI, full-custom design is rarely used due to the high labor cost.
  - Exceptions to this include the design of high-volume products such as memory chips, high-performance microprocessors and FPGA masters.

A full custom layout



Comparison Among Various Design Styles

	Design Style			
	FPGA	Gate array	Standard cell	Full custom
Cell size	Fixed	Fixed	Fixed height	Variable
Cell type	Programmable	Fixed	Variable	Variable
Cell placement	Fixed	Fixed	In row	Variable
Interconnect	Programmable	Variable	Variable	Variable
Design time	Very fast	Fast	Medium	Slow

END OF LECTURE 04

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 05: VLSI PHYSICAL DESIGN AUTOMATION (PART 1)

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

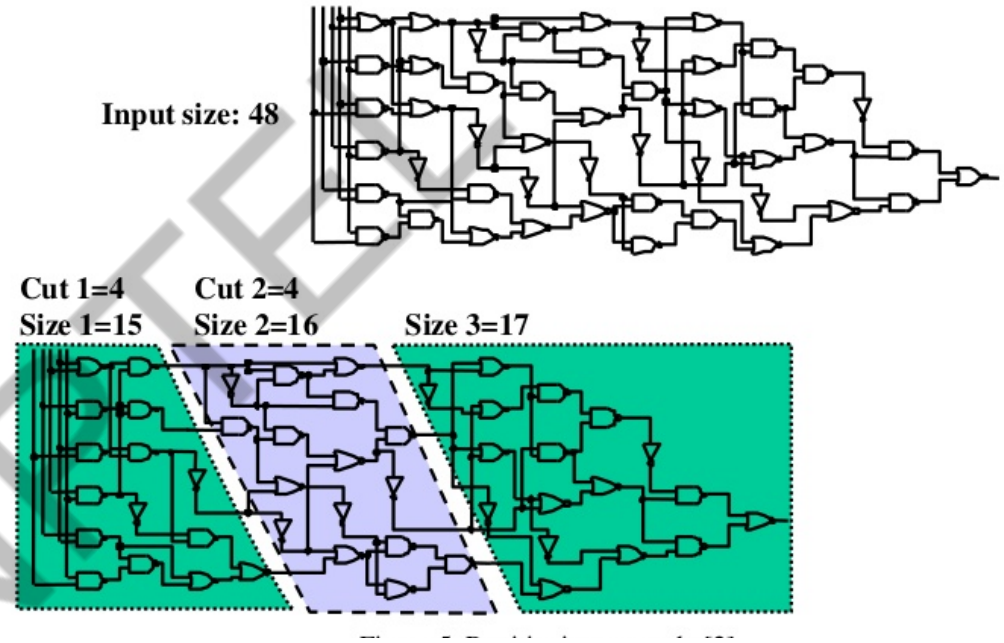
Introduction

- Main steps in VLSI physical design
  1. Partitioning and Floorplanning
  2. Placement
  3. Routing
  4. Static timing analysis
  5. Signal integrity and crosstalk issues
  6. Clock and power timing issues
  7. Physical verification and design sign-off

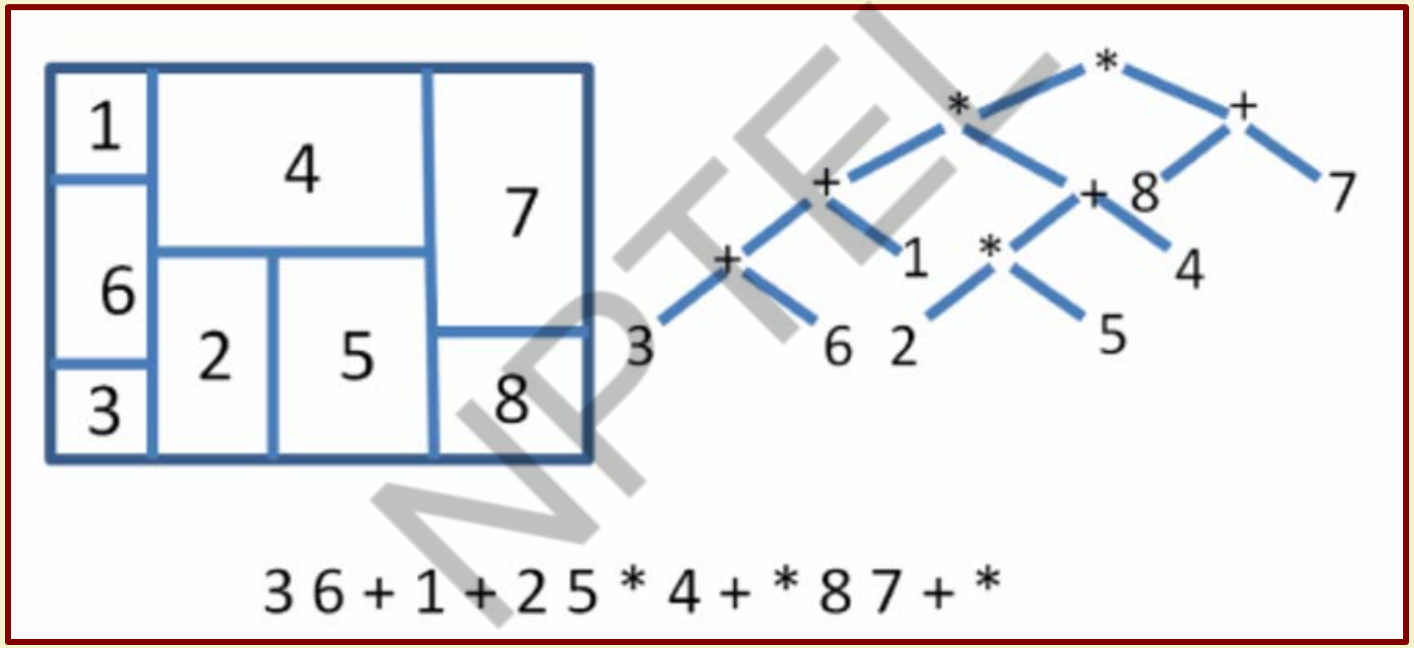
(a) Floorplanning

- A number of circuit blocks have to be laid out on the silicon floor.
- Determine the rough position for each of the blocks.
- Determine the shapes of the blocks that are flexible.
- Determine the pin locations of the blocks so that the interconnections can be easily routed in the future.
- This is a mandatory step for full-custom design style.

Example of Partitioning that Precedes Floorplanning



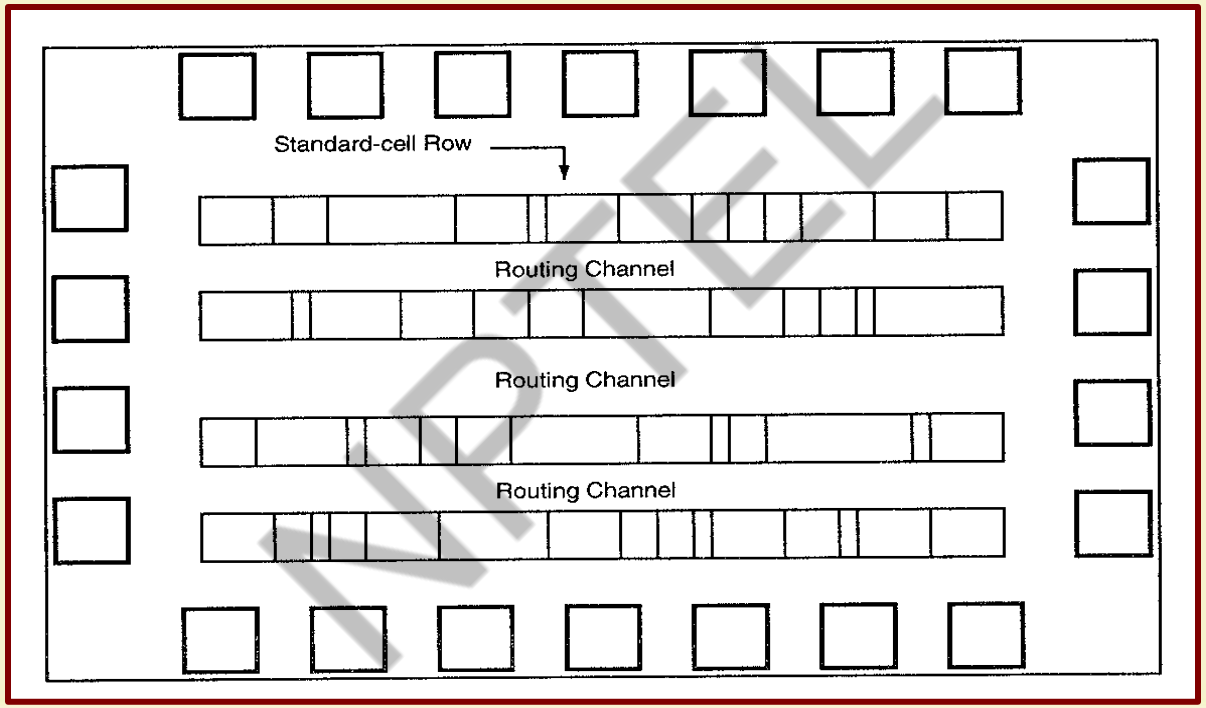
An Example Floorplan and Representation

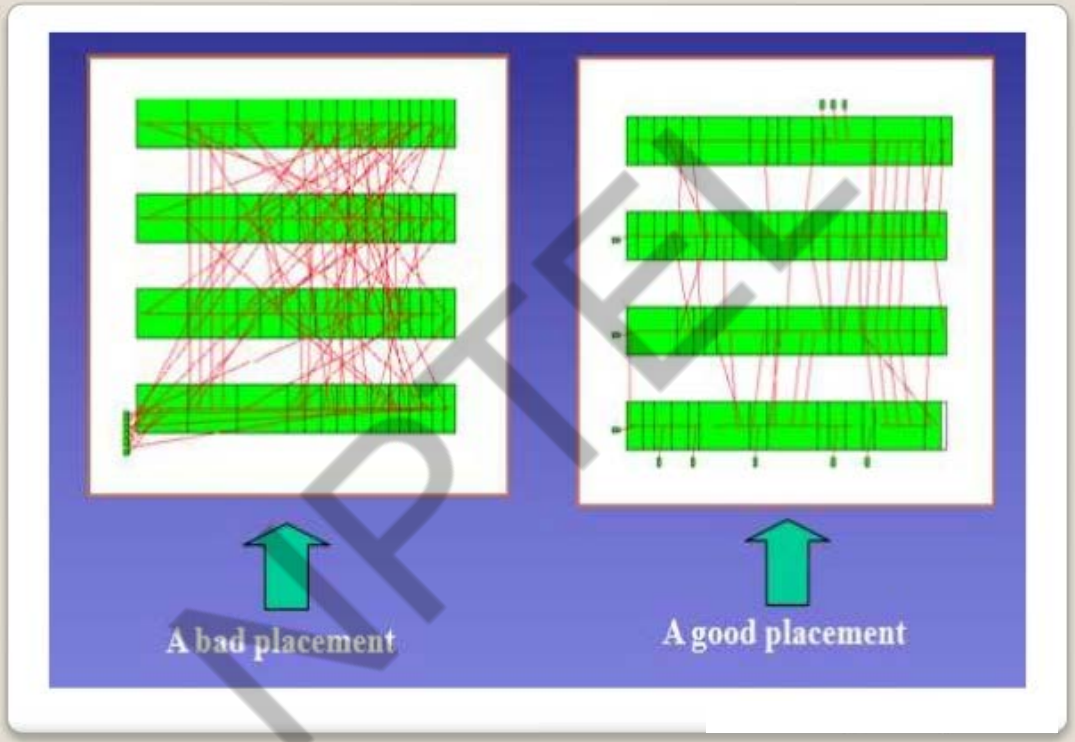


(b) Placement

- Given a set of blocks with defined shapes and pin locations, determine their positions on the layout.
- Keep adequate space between the blocks to run the interconnection lines.
- Interconnection modeling and cost estimation are important in this step.
- For some design styles (e.g. FPGA, semi-custom), floorplanning and placement are identical.

An Example Standard Cell Placement

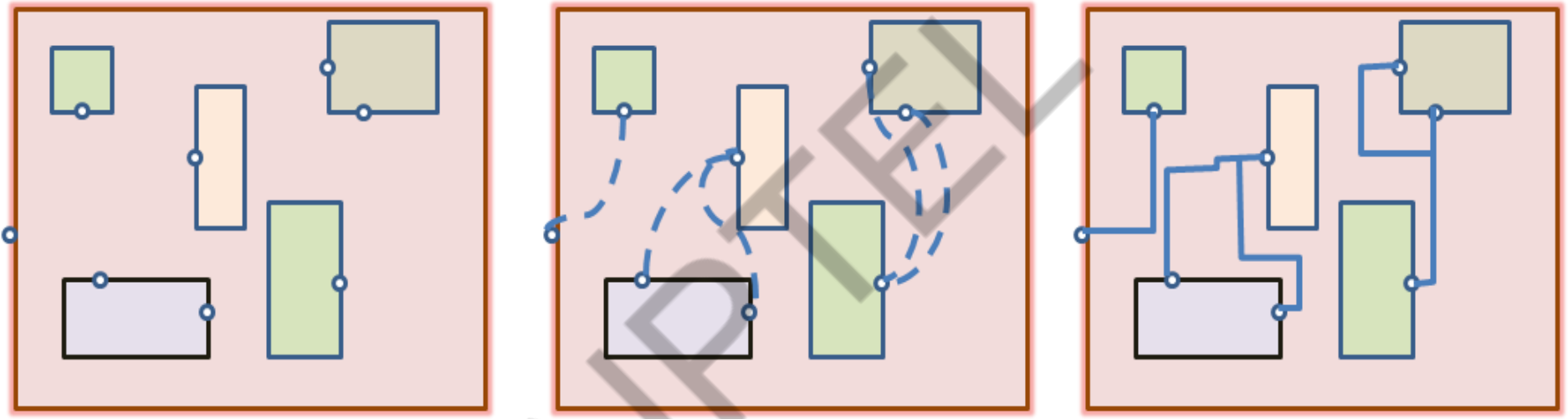




(c) Routing

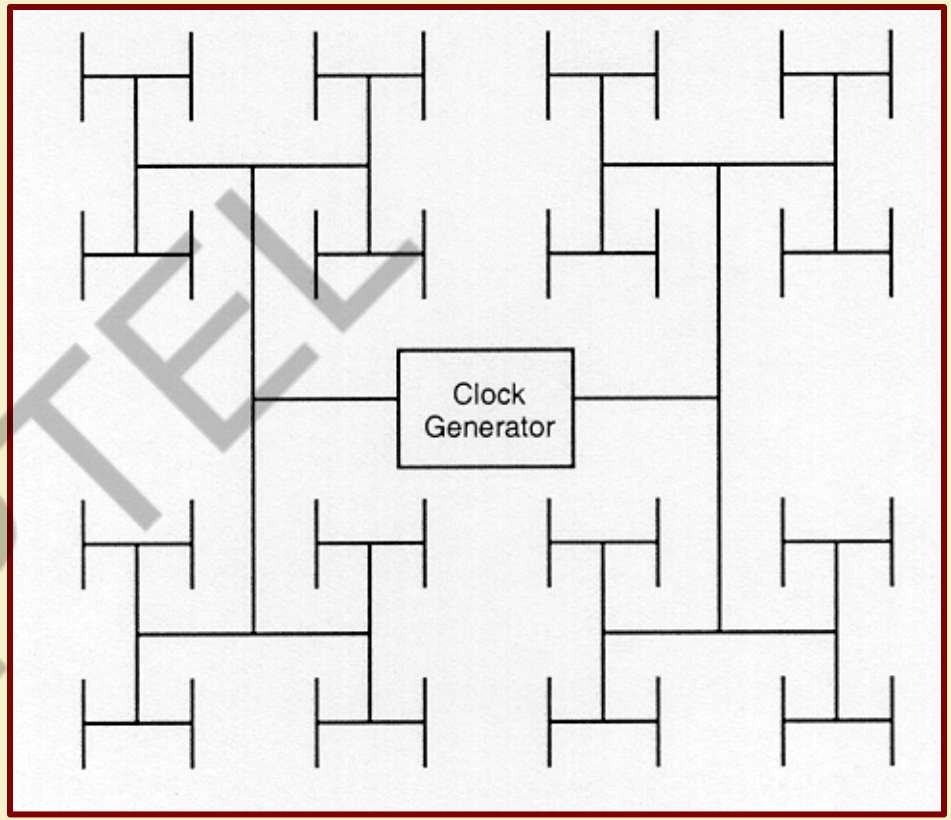
- Once all the blocks have been placed, run the interconnection lines so as to complete all the netlists.
- Various categories of routing:
  - Grid routing
  - Global routing
  - Detailed routing
  - Clock and power routing (needs special considerations)
- A bad placement can make routing very difficult.

Example of Global and Detailed Routing



An example showing – (i) Placement of standard cells, (ii) After global routing, (iii) After detailed routing

Example of Clock Routing



END OF LECTURE 05

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 06: VLSI PHYSICAL DESIGN AUTOMATION (PART 2)

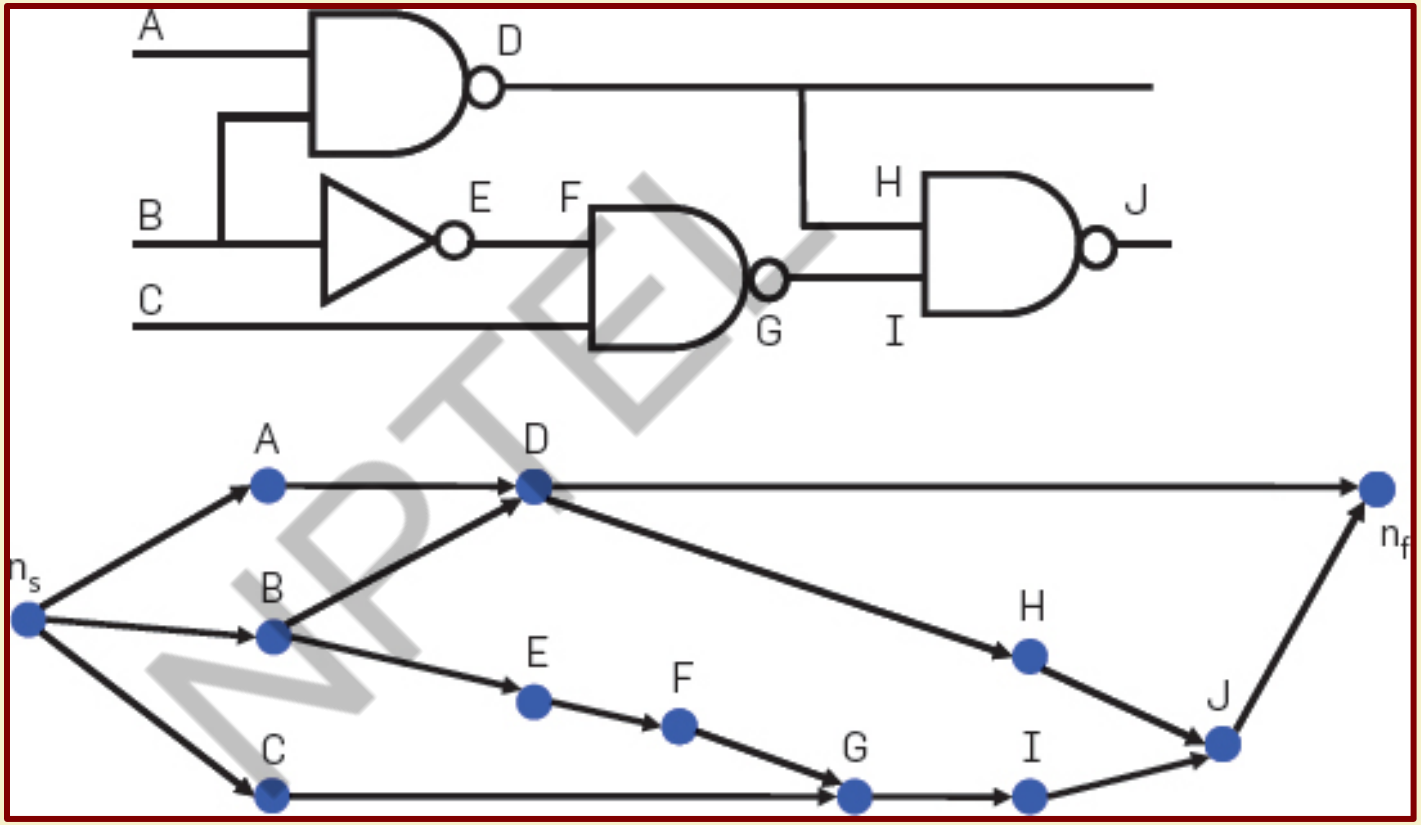
PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(d) Static Timing Analysis

- Analyze a circuit netlist to determine worst-case circuit delays.
  - Predict maximum clock frequency.
- Carry out timing optimization so that a circuit can run faster.
- An essential step in modern-day high performance systems.

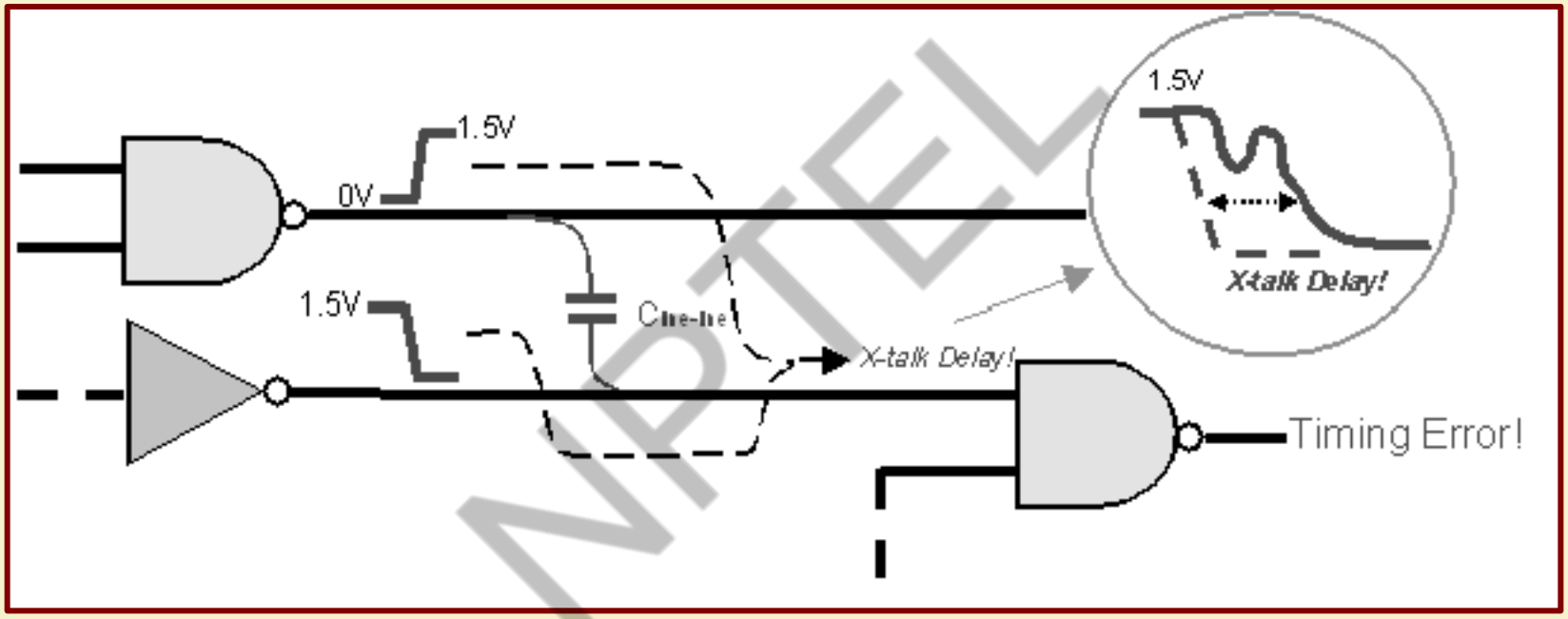
Example of Static Timing Analysis



(e) Signal Integrity and Crosstalk Issues

- Various sources of noise in signal lines, and ways to limit their impact for high performance systems.
- Identify design rules that may limit the noises to acceptable levels.
- Essential for high performance systems.

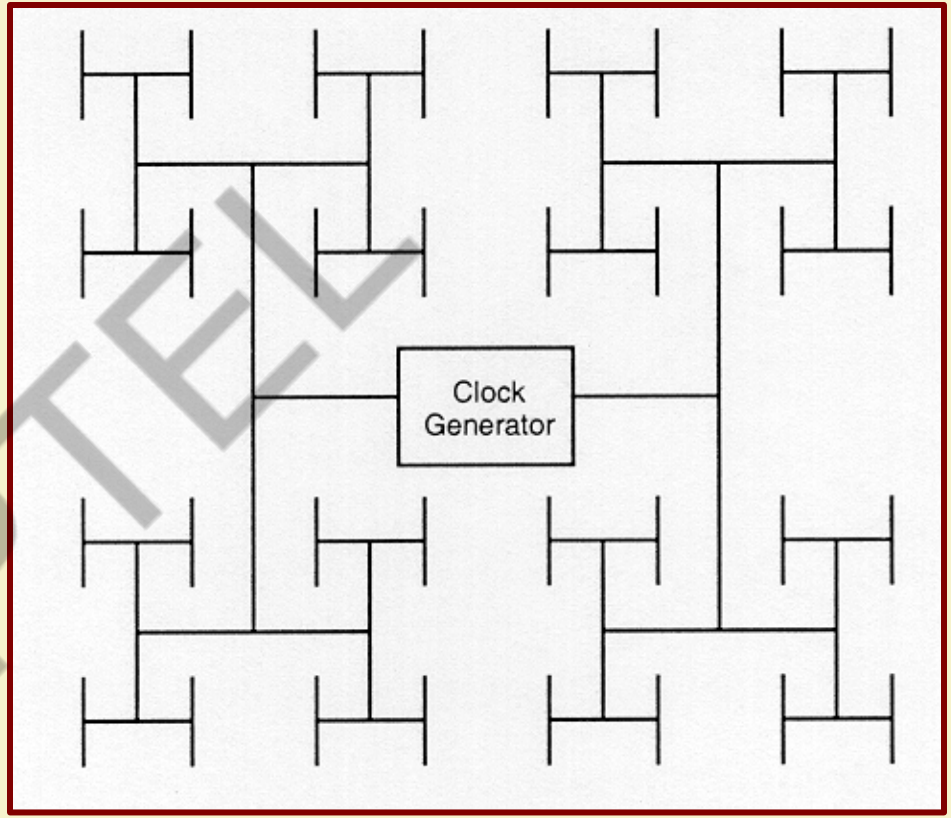
Example of Signal Crosstalk



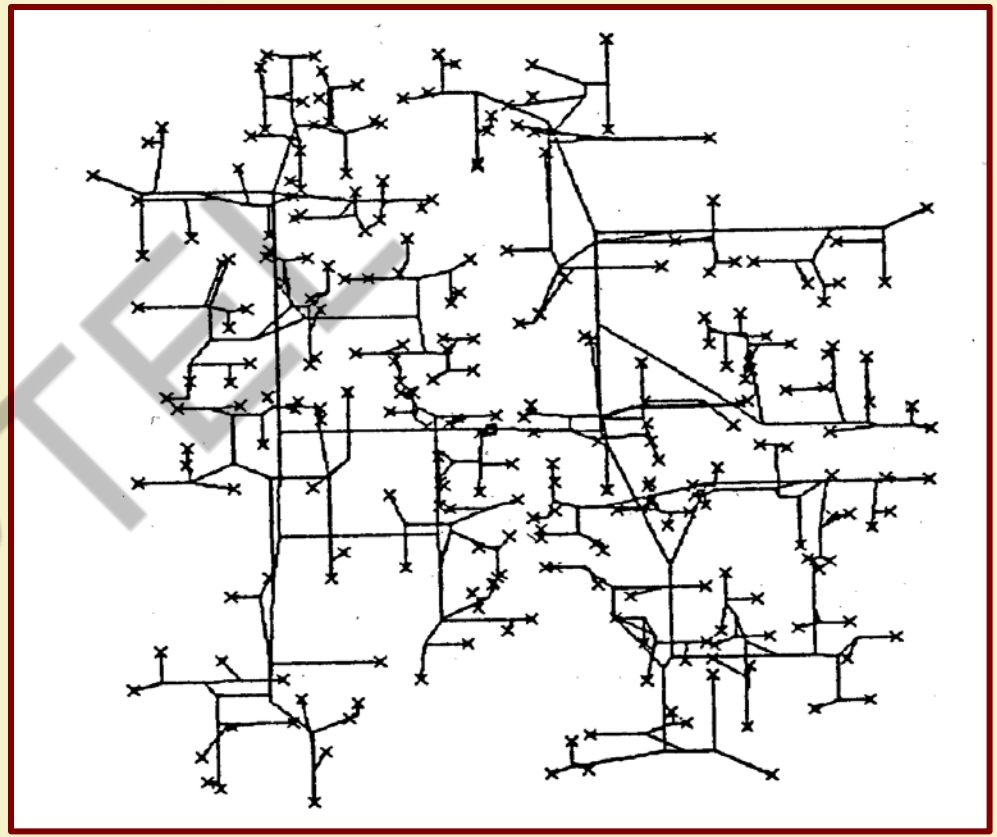
(f) Clock and Power Routing Issues

- Clock routing
  - Needs to consider issues like delays, skews and hazards.
  - Clock signals must reach various parts of the chip at the same time.
- Power routing
  - Power requirement may vary for different parts of the chip.
  - The Vdd and GND lines must have adequate widths to carry the required currents.
  - Requires a priori power analysis.

Example of Clock Routing



Clock Routing in a Typical Design



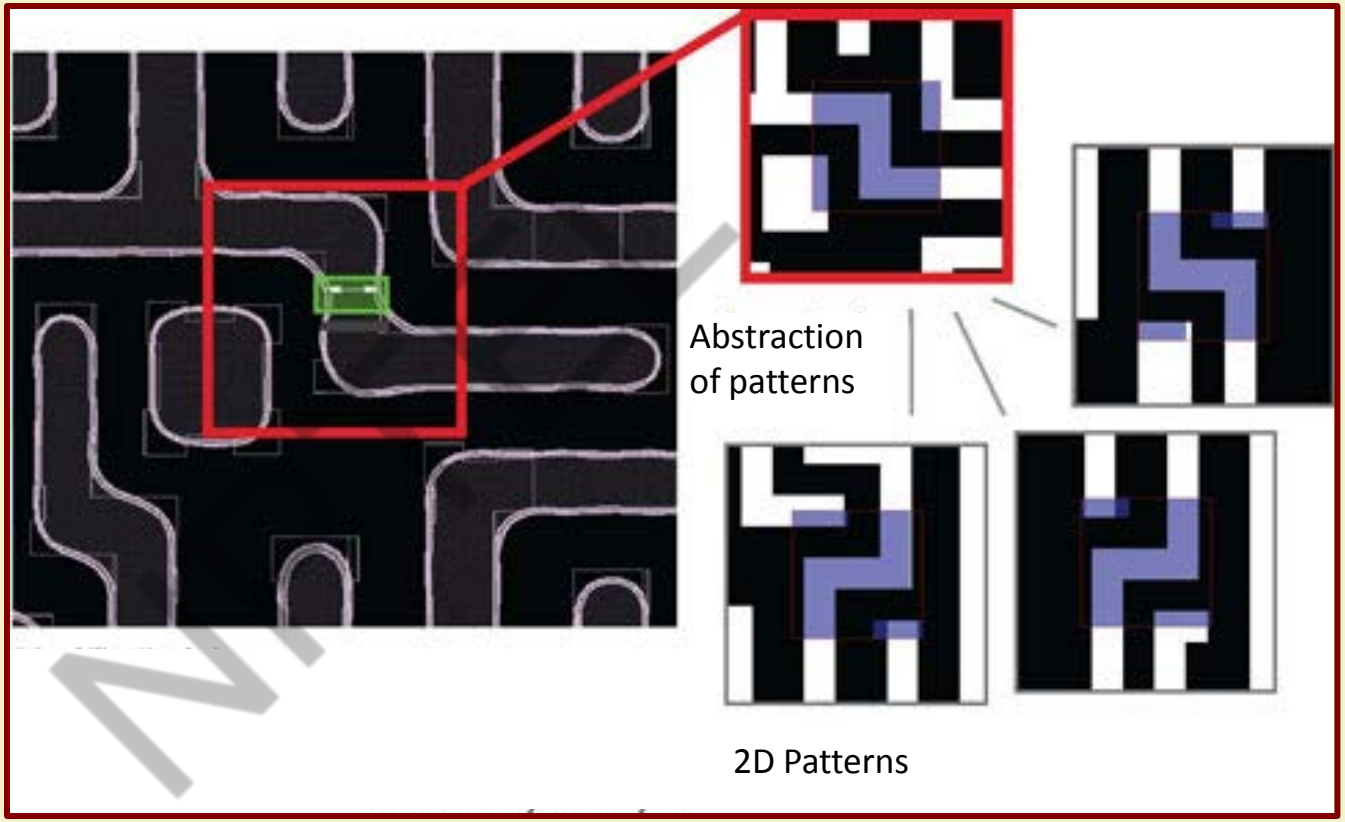
- Issues in routing Vdd and GND nets.
  - Various segments can have variable widths, depending on the maximum currents that are expected to flow.
  - Line width calculation is a complex process.
  - Typically laid out on separate metal layer.

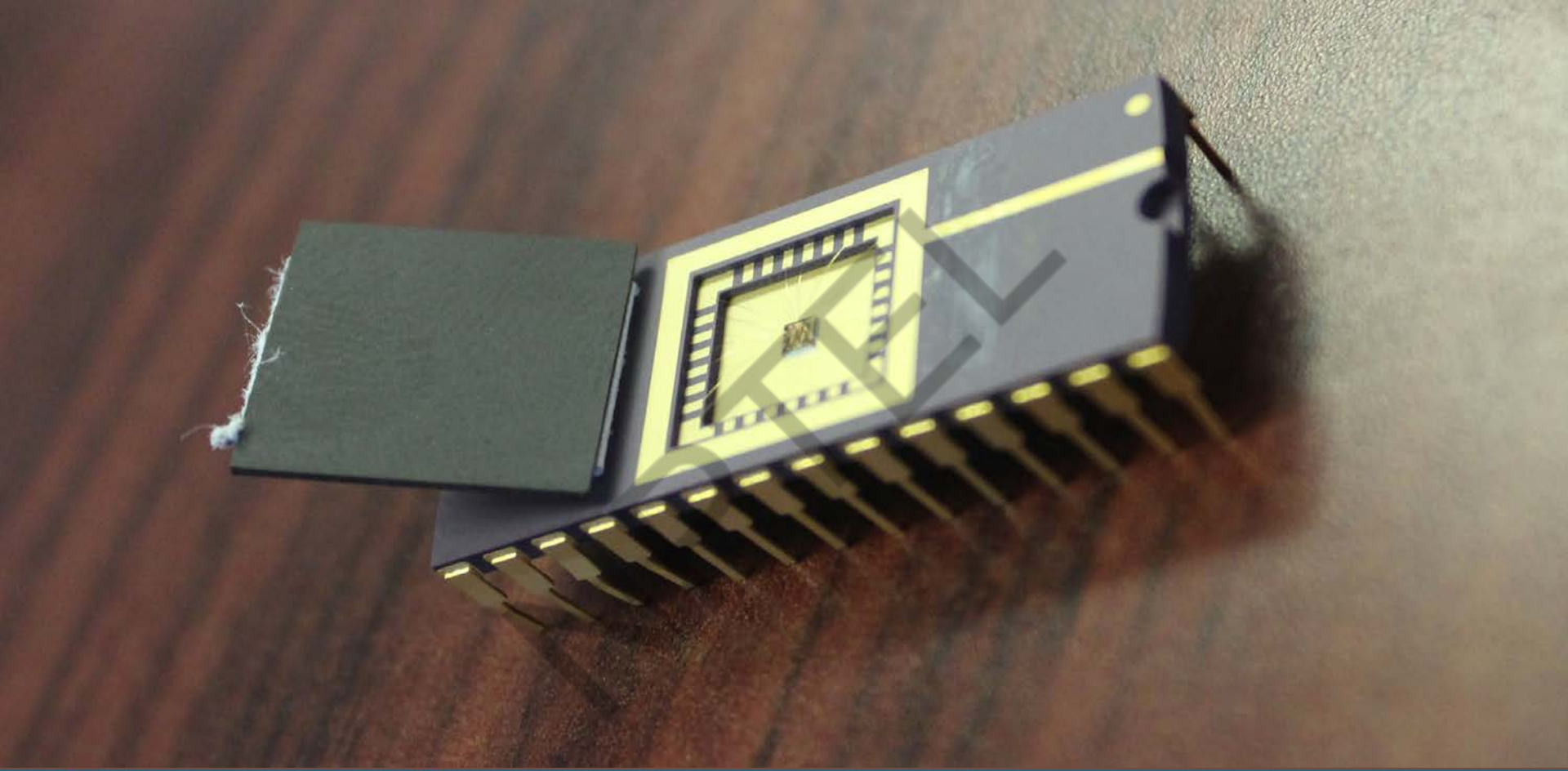
(g) Physical Verification and Design Signoff

- Because of the size and complexity of modern-day VLSI chips, physical verification tools and methodologies are essential before a design is sent for fabrication (i.e. taped out).
- Iterative process involving incremental fixes across the design in one or more check type and retesting as required.

- Design rule check is a very useful analysis process.
  - Minimum width of lines on various layers.
  - Minimum separation between lines on same layer.
  - Minimum separation between lines on different layers.
  - Minimum size of various types of contacts.
- Usually template based matching is carried out on the entire netlist to look for anomalies.

Example of Physical Layout Verification





END OF LECTURE 06