VLSI DESIGN FLOW: RTL TO GDS

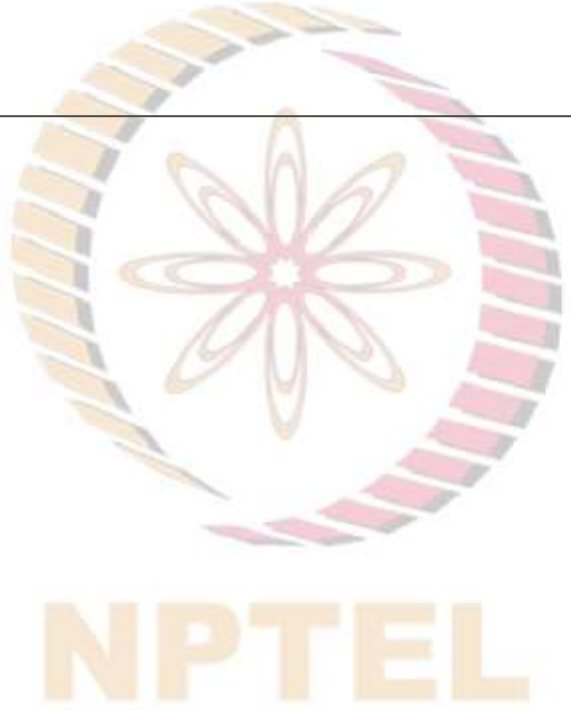
Lecture 5
Overview of VLSI Design Flow: III

Sneh Saurabh
Electronics and Communications Engineering
IIIT Delhi

Lecture Plan

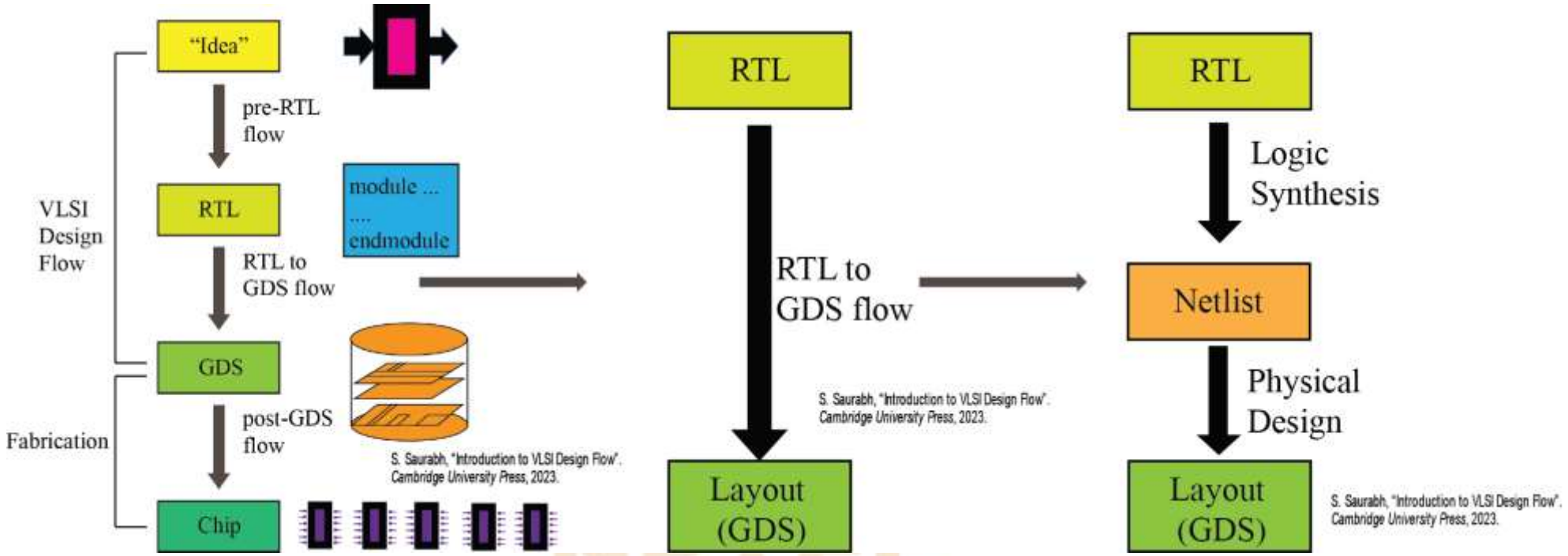
Overview of VLSI Design Flow

- RTL to GDS Implementation
  - Logic Synthesis

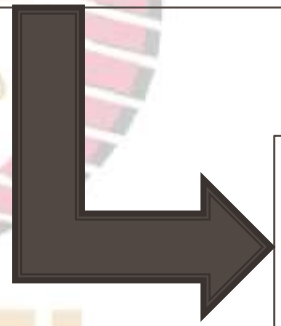


Overview of RTL to GDS Flow

VLSI Design Flow: RTL to GDS Flow



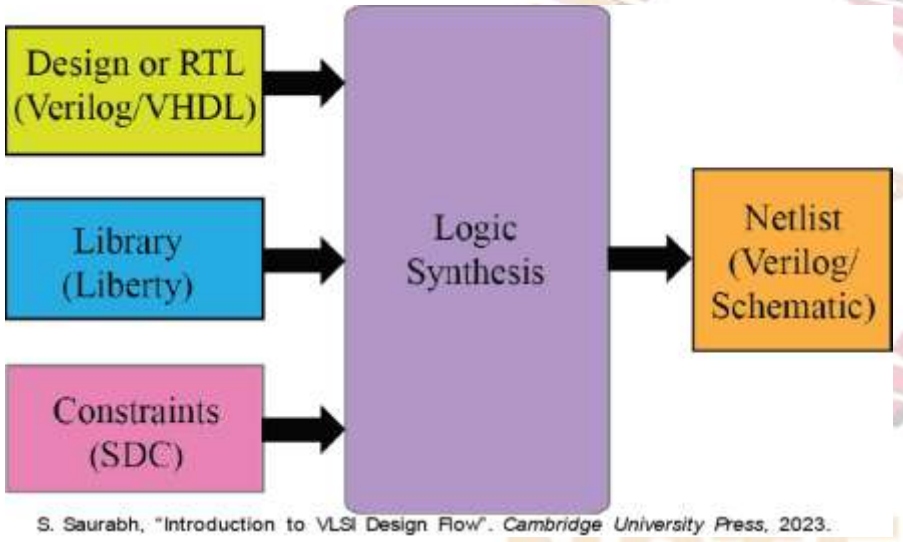
Overview of RTL to GDS Flow



Logic Synthesis

RTL to GDS flow: Logic Synthesis

- **Logic Synthesis**: process by which RTL is converted to an equivalent circuit as interconnection of logic gates



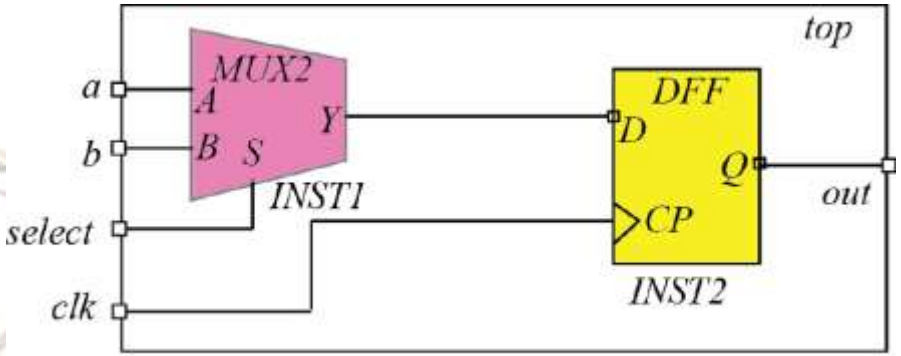
S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.

- **RTL**: given design (Verilog, VHDL)
- **Library**: standard cells and macros (Liberty)
- **Constraints:** design goals, expected timing behavior, environment (SDC)

- **Netlist**:
  - Interconnection of logic gates
  - Usually represented using Verilog constructs or schematic

Logic Synthesis: Illustration

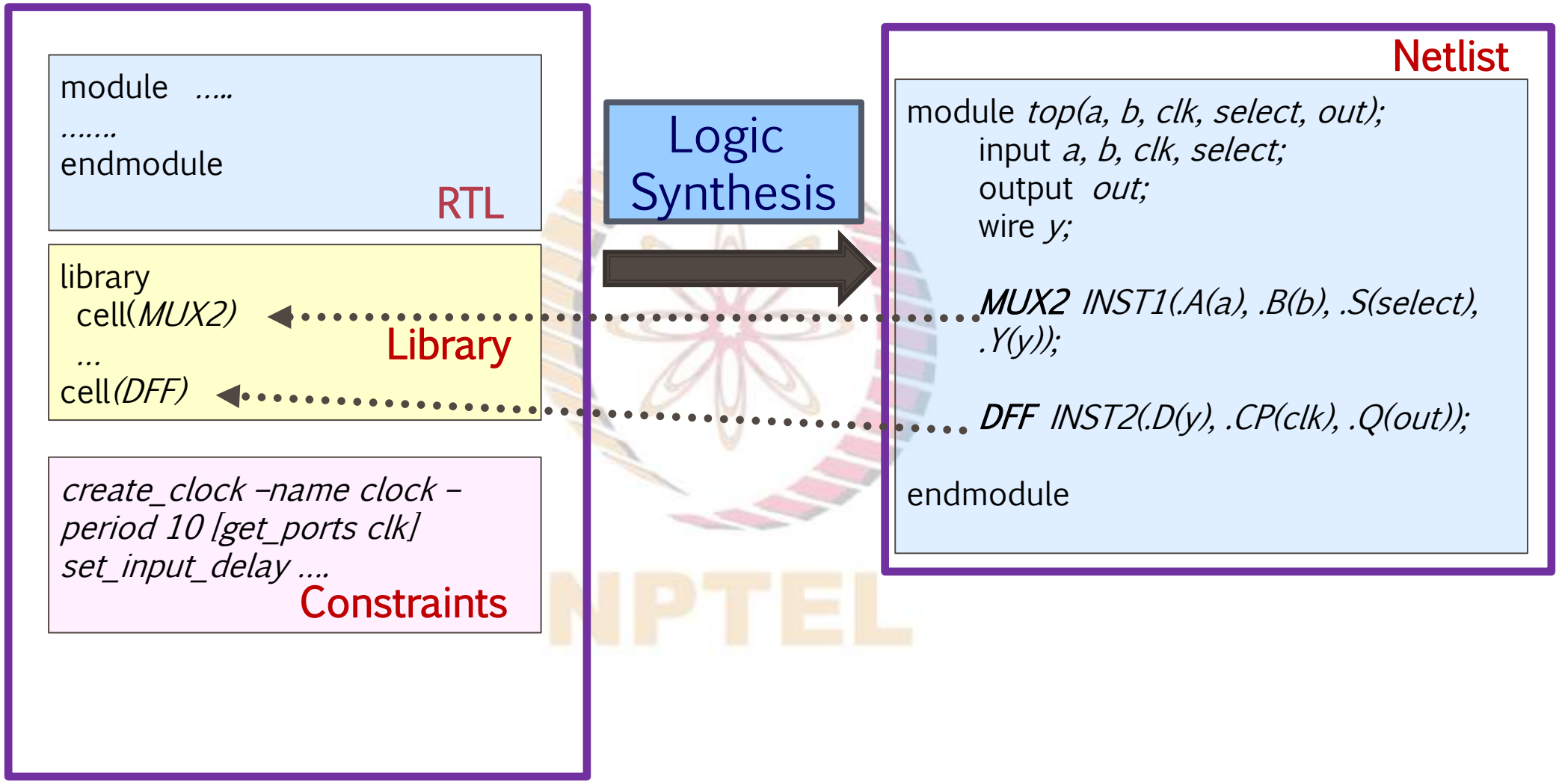
```
module top(a, b, clk, select, out);

        input a, b, clk, select;
        output out;
        reg out;
        wire y;

        assign y = (select) ? b : a;

        always @(posedge clk)
                begin
                        out <= y;
                end

endmodule
```



S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.

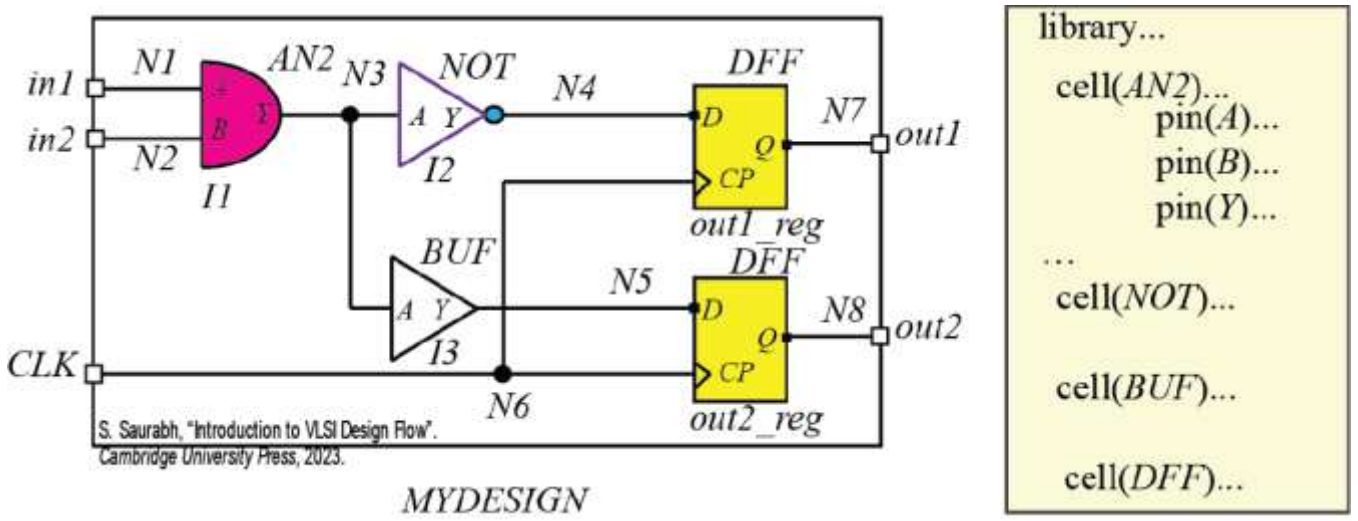
```
module top(a, b, clk, select, out);
        input a, b, clk, select;
        output out;
        wire y;

        MUX2 INST1(.A(a), .B(b), .S(select), .Y(y));
        DFF INST2(.D(y), .CP(clk), .Q(out));

endmodule
```

Logic Synthesis: Inputs and Outputs



Netlist Terminologies

Netlist Terminologies: Design and Ports

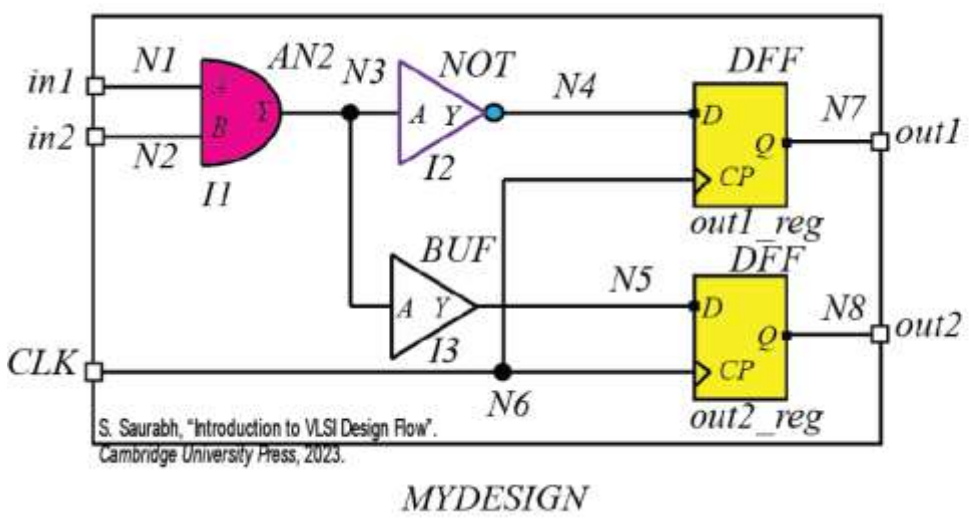


Design: Top level entity that represents the circuit. Example: *MYDESIGN*

Ports: The interfaces of the Design through which it communicates with the external world. Example: *in1, in2, CLK, out1, out2*

- **Input Ports**: Signals going inside the design. Example: *in1, in2, CLK*
- **Output Ports**: Signals going outside from the design. Example: *out1, out2*

Netlist Terminologies: Cells

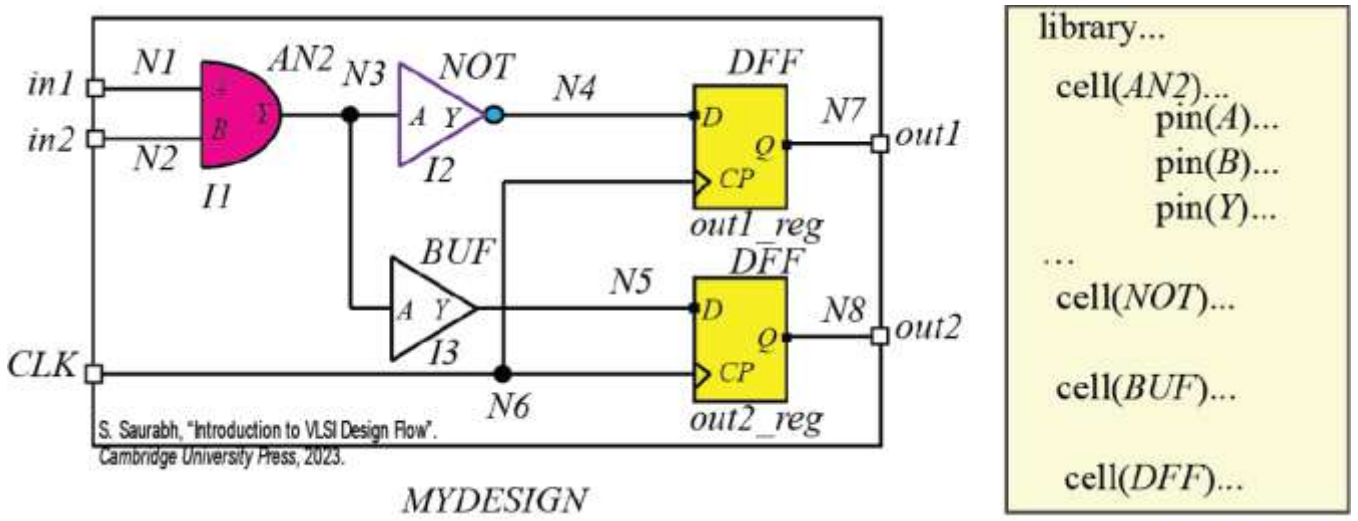



```
library...
  cell(AN2)...
        pin(A)...
        pin(B)...
        pin(Y)...
  ...
  cell(NOT)...

  cell(BUF)...

  cell(DFF)...
```

- **Cells**: basic entity delivering combinational or sequential function contained in libraries.

  Examples: *AN2, NOT, BUF, DFF*
- Design is composed of multiple *cells* connected together

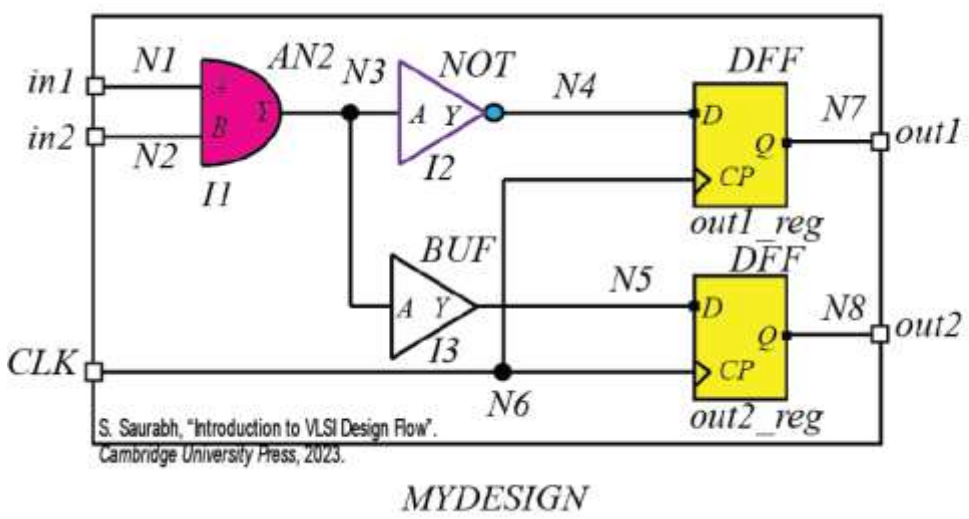
Netlist Terminologies: Instances




```
library...
  cell(AN2)...
        pin(A)...
        pin(B)...
        pin(Y)...
  ...
  cell(NOT)...

  cell(BUF)...

  cell(DFF)...
```

- **Instances**: cells when used inside a design are called *instances*.
  Examples: *I1, I2, I3, out1_reg, out2_reg*.
- Using a cell in a design is called ***instantiation***.
- The same cell can be instantiated multiple times.
  Example: *out1_reg* and *out2_reg* are instances of the same cell *DFF*.

Netlist Terminologies: Pins

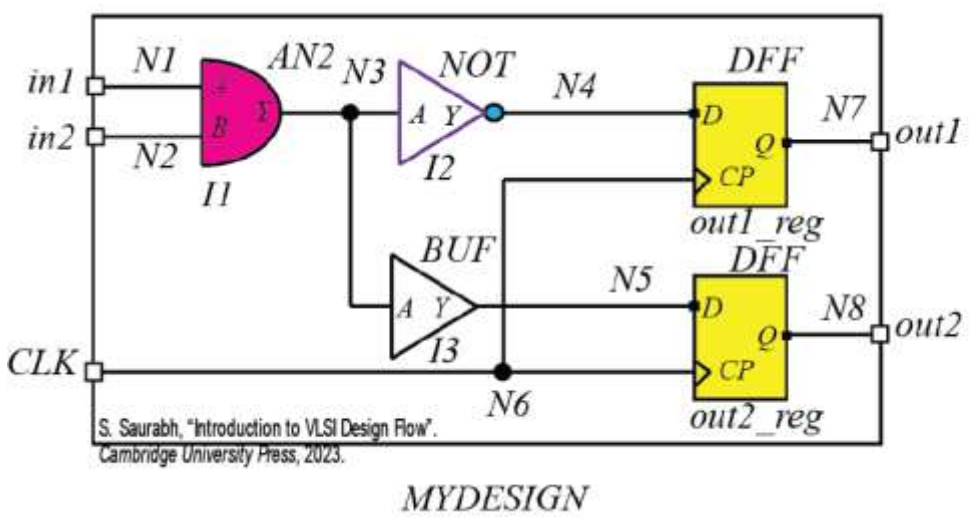



```
library...
  cell(AN2)...
        pin(A)...
        pin(B)...
        pin(Y)...
  ...
  cell(NOT)...

  cell(BUF)...

  cell(DFF)...
```

- **Pin**: An interface of a library cell or instance through which it communicates with the other components is called a pin.
- Examples: *A, B, Y* are the pins of the cell *AN2* and the instance *I1*

- **Library Pin** and **Instance Pin** (if we want to be explicit)
  - ➢ Often apparent from the context
- Input Pin or Output Pin based on direction of flow of signal to cell/instance

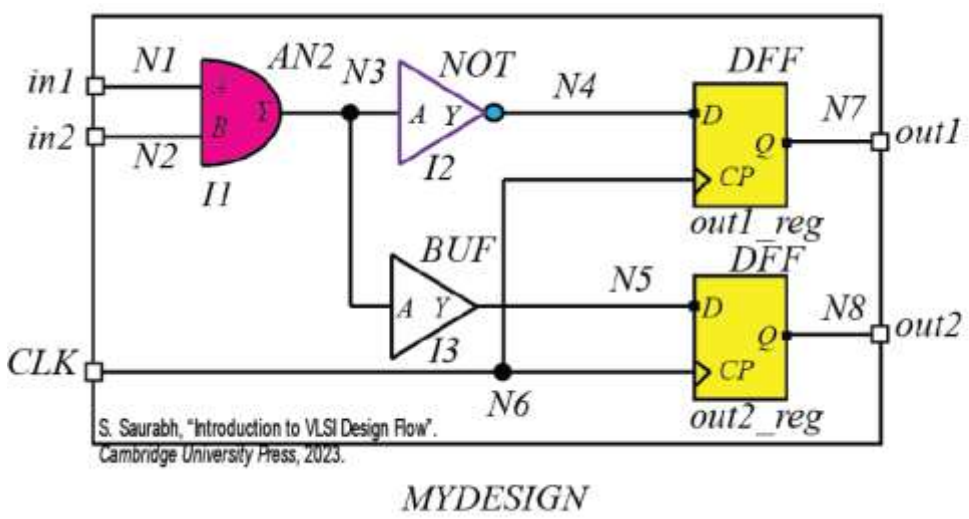
Netlist Terminologies: Pin Names




```
library...
  cell(AN2)...
        pin(A)...
        pin(B)...
        pin(Y)...
  ...
  cell(NOT)...

  cell(BUF)...

  cell(DFF)...
```

- **Instance pin name**: typically specified as combination of instance name and pin name separated by /
- Examples: *I1/A, I1/B , out1_reg/Q*

Netlist Terminologies: Nets

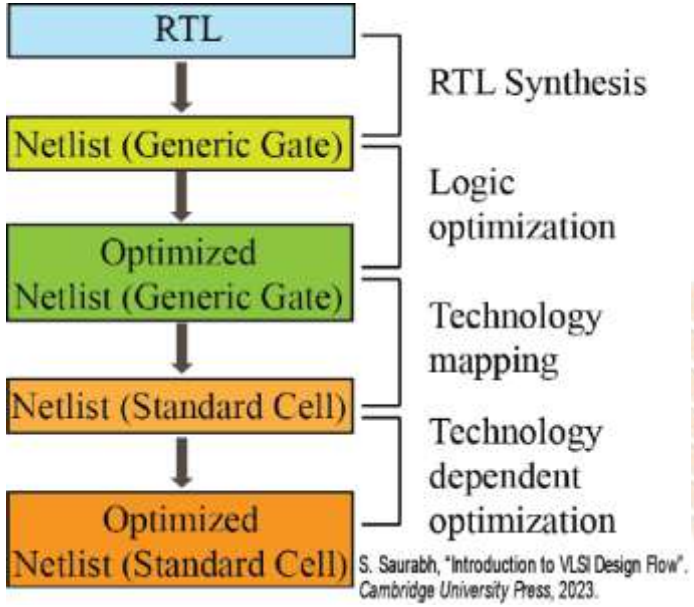



```
library...
  cell(AN2)...
        pin(A)...
        pin(B)...
        pin(Y)...
  ...
  cell(NOT)...

  cell(BUF)...

  cell(DFF)...
```

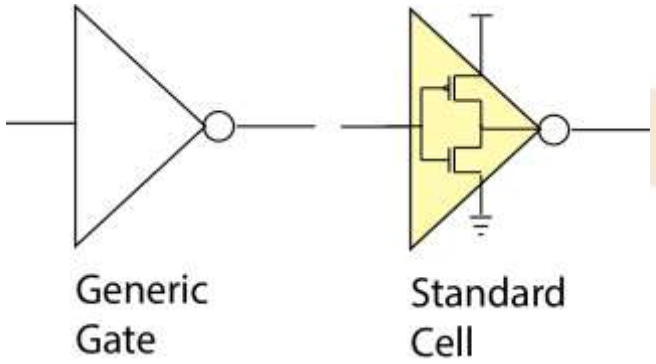
- **Net**: The wire that connects different instances and ports is a called net.
  Examples.: *N1, N2, N3, …N8*

Logic Synthesis Tasks

Logic Synthesis Tasks: RTL Synthesis



S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.



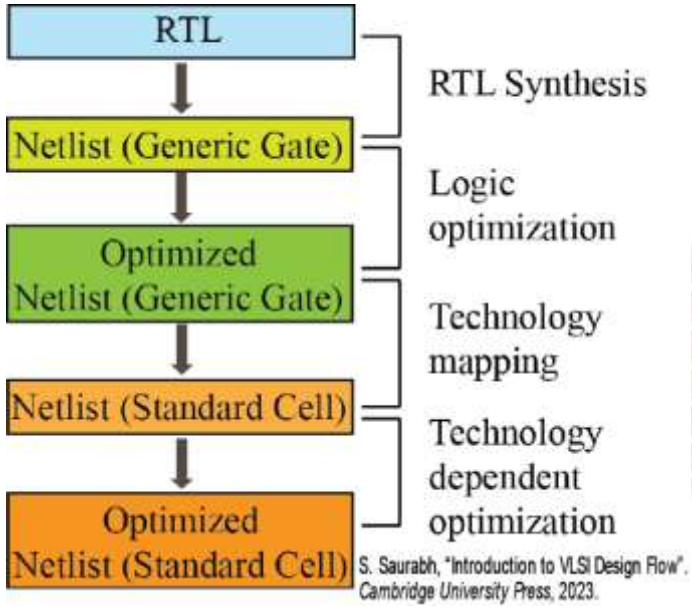
RTL synthesis:

- Initial part of logic synthesis consisting of translating an RTL to a netlist of generic logic gates

Generic Logic Gates

- A generic logic gate has a well-defined Boolean function.
  - AND, NAND, XOR, multiplexer, demultiplexer etc.
  - Latches and flip-flops.
- Does not have a fixed transistor-level implementation
  - Does not have a well-defined area, delay, and power attributes

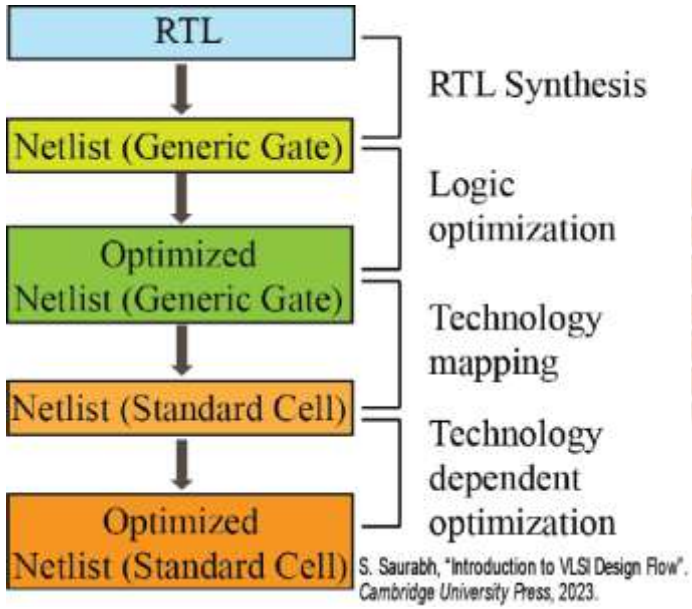
Logic Synthesis Tasks: Logic Optimization



Logic Optimization:

- Optimizations on a generic gate netlist
- Typically area-driven

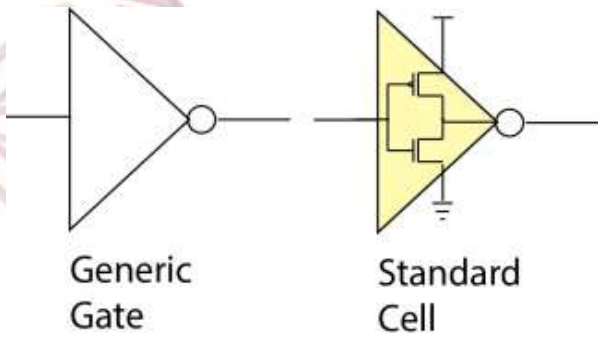
Logic Synthesis Tasks: Technology Mapping and Optimization



S. Saurabh, "Introduction to VLSI Design Flow", Cambridge University Press, 2023.

Technology Mapping:

- Map a netlist consisting of generic logic gates to the standard cells in the given technology library
- Obtain a netlist consisting of standard cells



Technology-dependent optimization:

- PPA can be estimated more accurately after technology mapping
- Perform timing, area, and power optimizations over netlist consisting of standard cells

References

- G. D. Micheli. “Synthesis and Optimization of Digital Circuits”. *McGraw-Hill Higher Education*, 1994.
- S. Saurabh, “Introduction to VLSI Design Flow”. Cambridge: *Cambridge University Press*, 2023.