Very Large Scale Integration II - VLSI II

Digital Design Methodology

ITU VLSI Laboratories

Istanbul Technical University

Outline

- Properties of Digital Circuits
- Implementation Methods
  - Programmable Logic
  - Semi Custom
  - Full Custom
- Project Constraints
- Digital Circuit Representations
- Digital Design Considerations
- Digital Design Flow

Properties of Digital Circuits

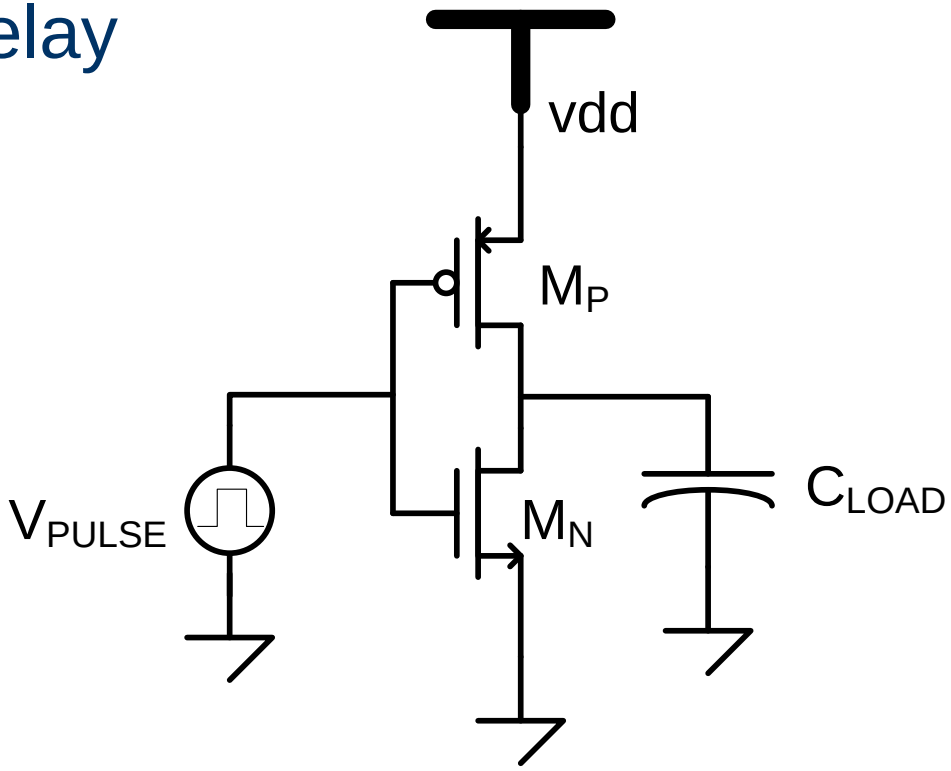
- Most general form
  - Inputs + Present State → Output + Next State
- For most cases, only 1 and 0
  - No 0.5 or Z
- Less specifications
  - Delay
  - Transition
  - Power
  - Area

Properties of Digital Circuits

- Combinational
  - Inputs → Output
- Sequential
  - Moore Type
    - Inputs → Next State
    - Present State → Output
  - Mealy Type
    - Inputs + Present State → Output + Next State

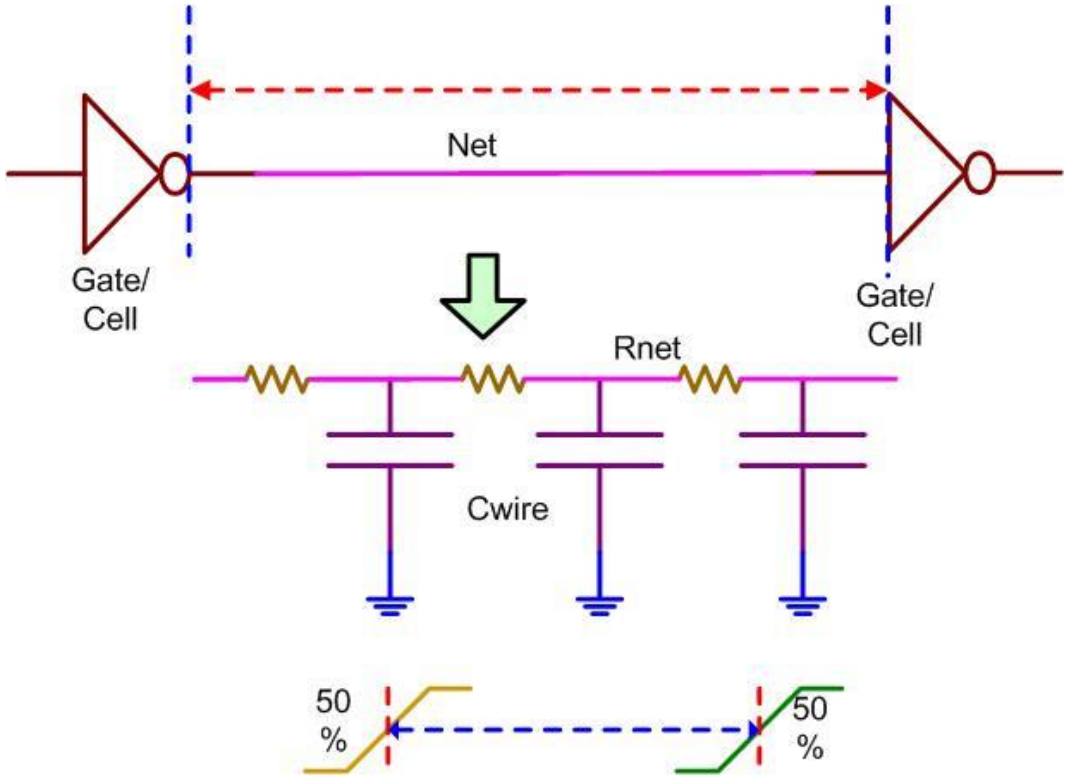
Properties of Digital Circuits

- Delay = Gate Delay + Interconnect Delay
- Gate Delay



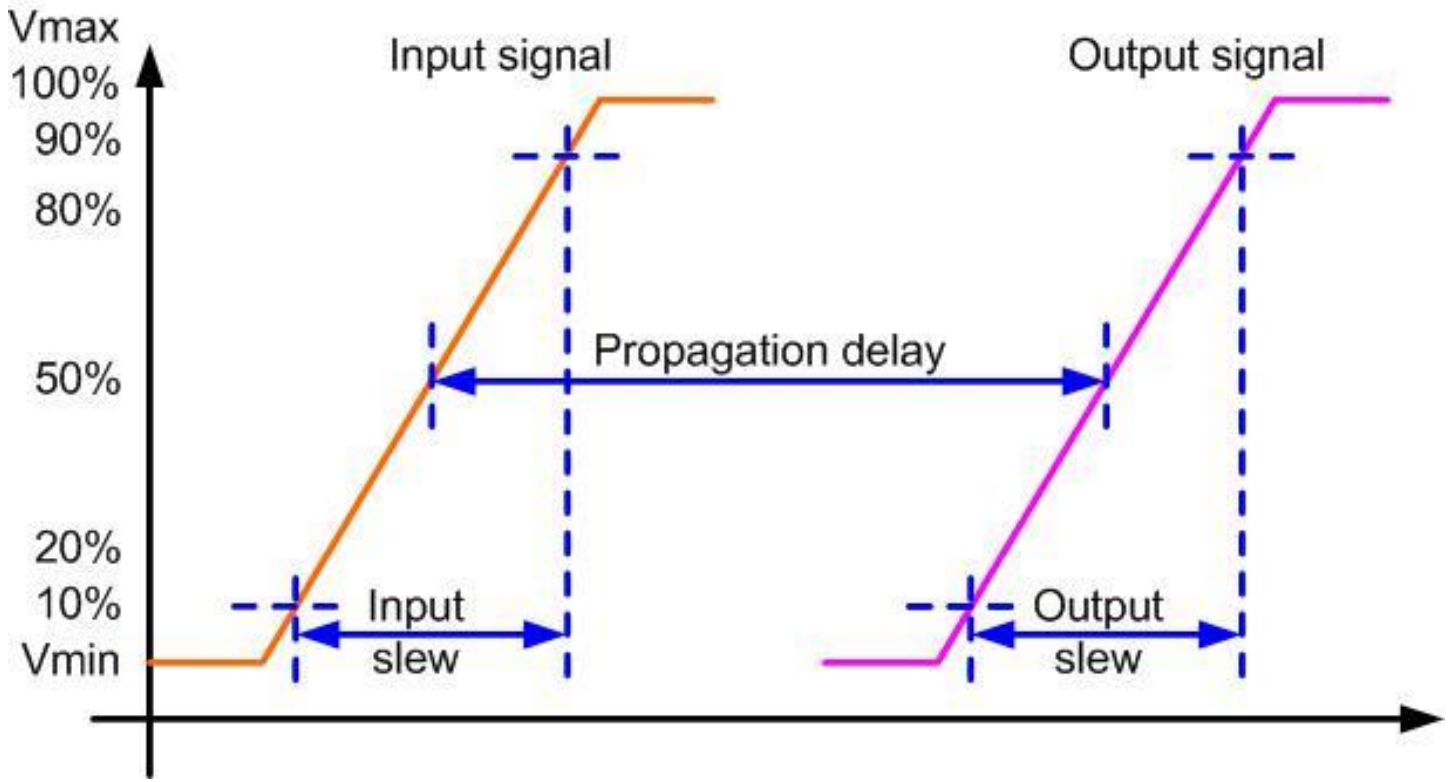
Properties of Digital Circuits

- Interconnect Delay



Properties of Digital Circuits

- Transition

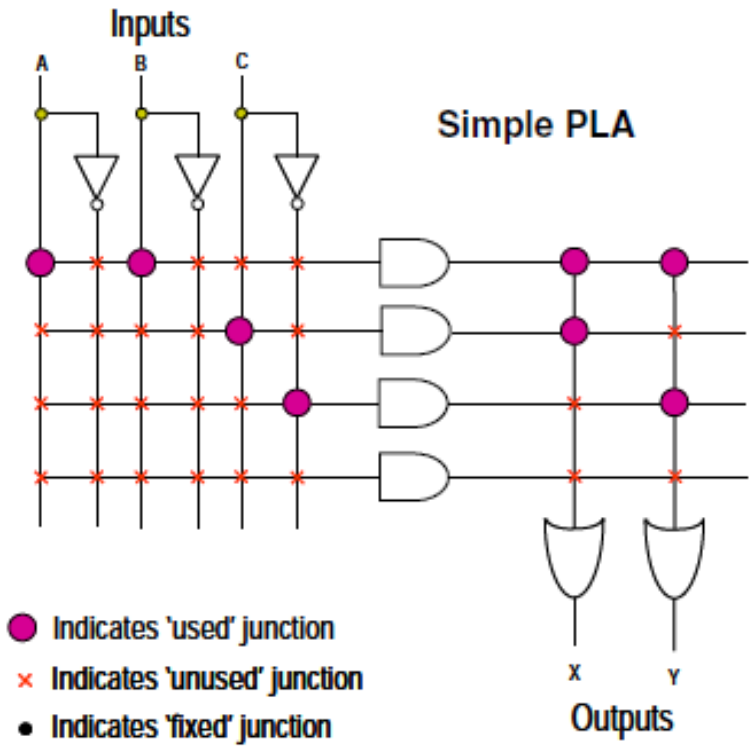


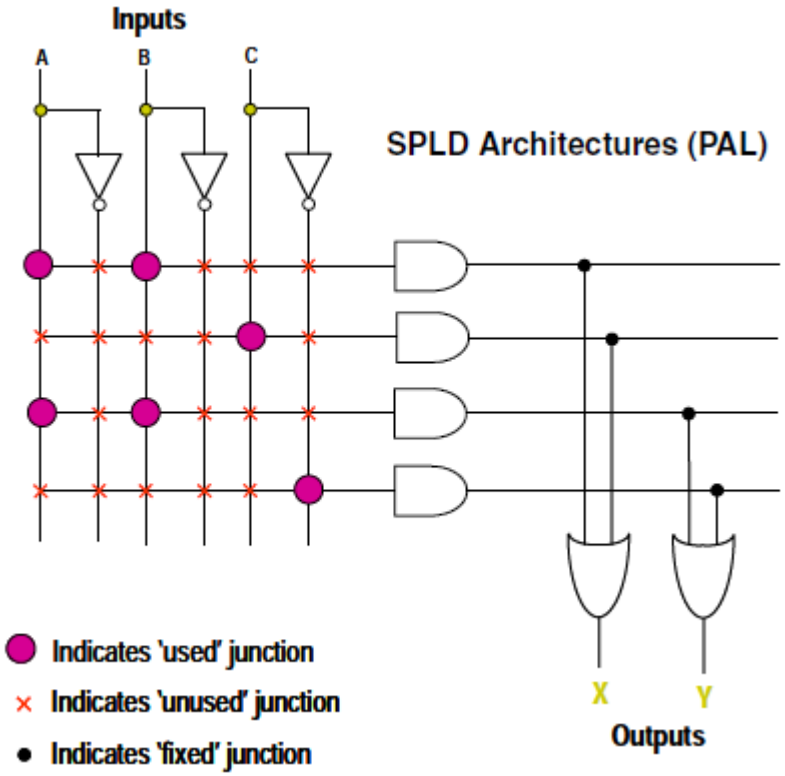
Implementation Methods

- Programmable Logic (FPGA, CPLD, PLA, PAL)
- Semi Custom (ASIC, Standard Cell)
- Full Custom

Programmable Logic

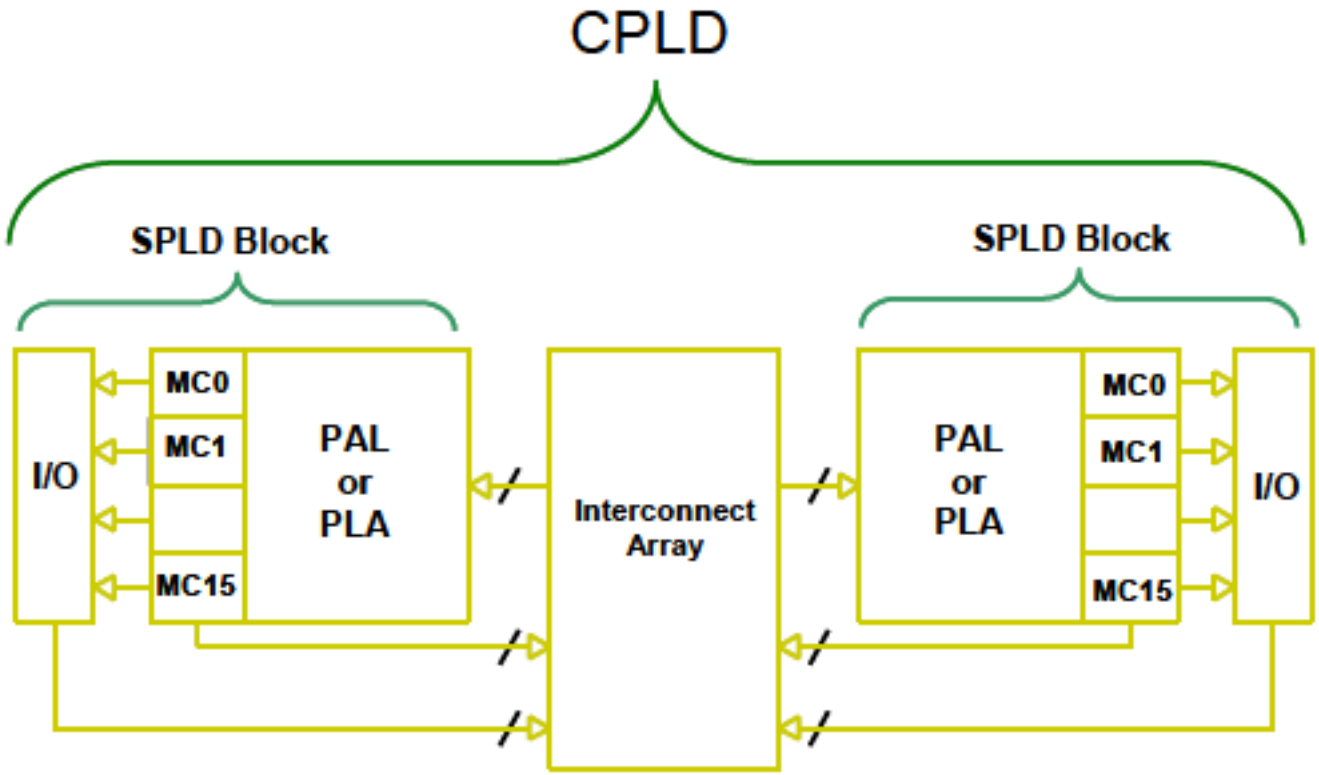
- PLA and PAL





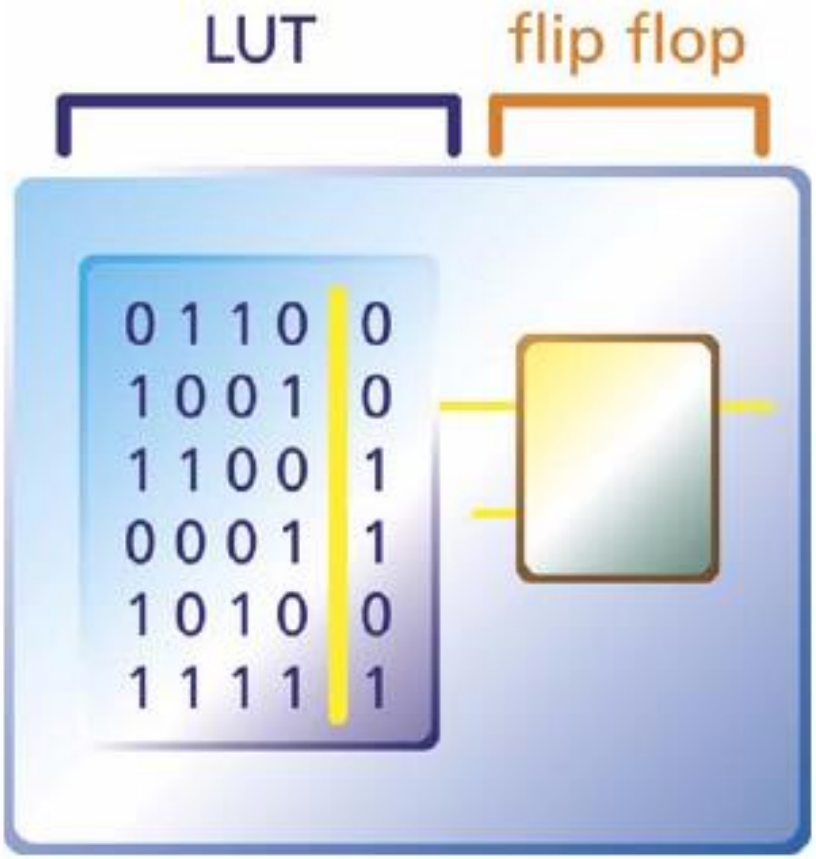
Programmable Logic

- CPLD



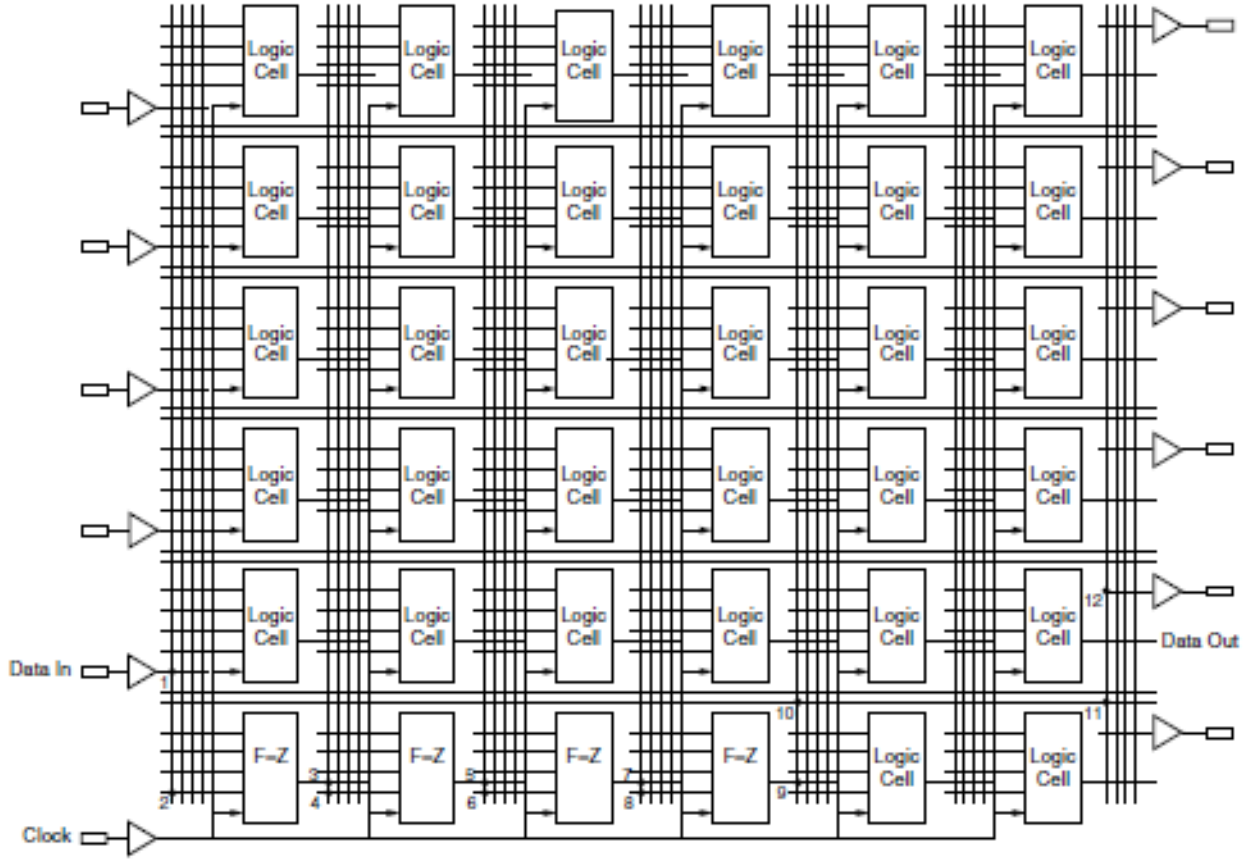
Programmable Logic

- FPGA
  - **CLB (LUT)**
  - Interconnect
  - PLL, DCM
  - I/O



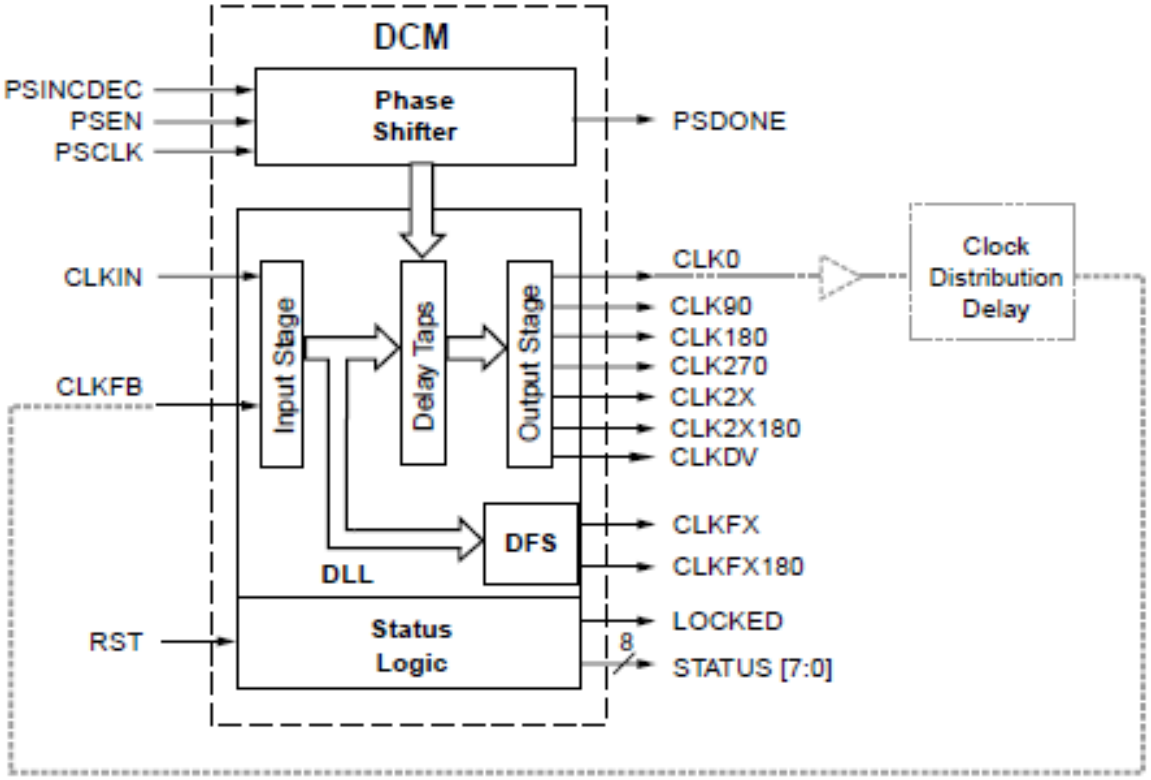
Programmable Logic

- FPGA
  - CLB (LUT)
  - **Interconnect**
  - PLL, DCM
  - I/O



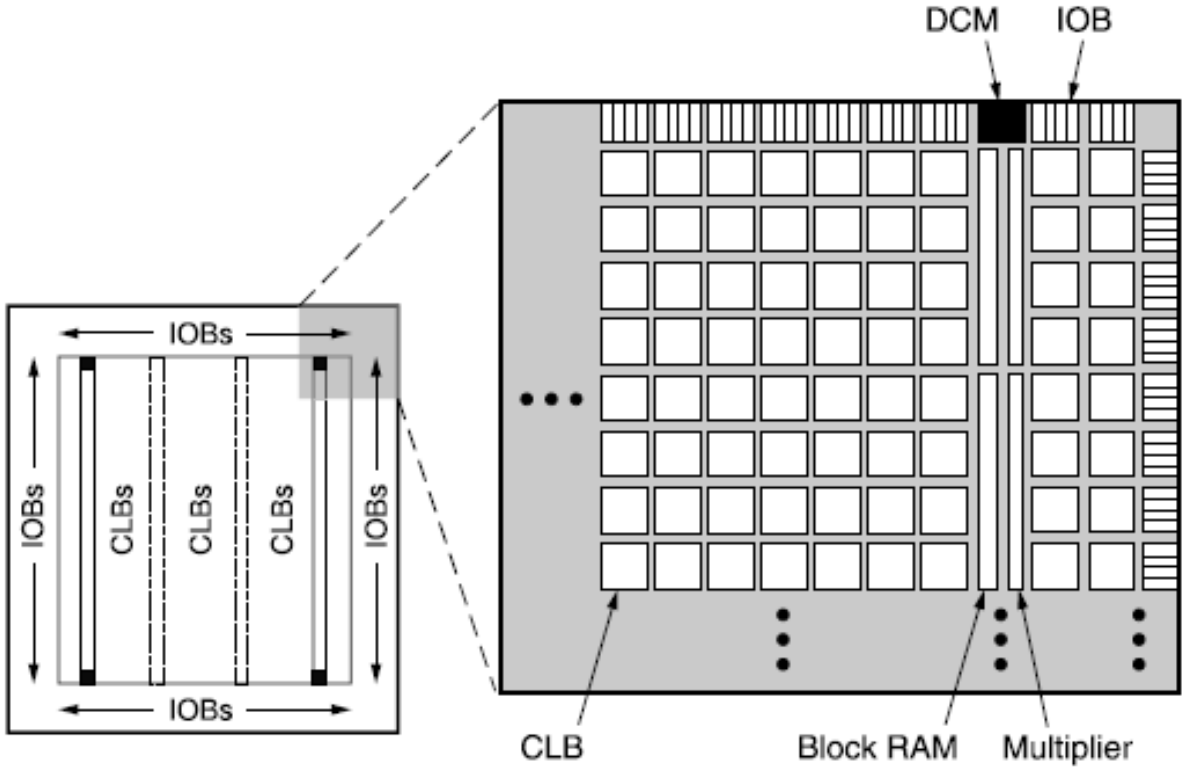
Programmable Logic

- FPGA
  - CLB (LUT)
  - Interconnect
  - **PLL, DCM**
  - I/O



Programmable Logic

- FPGA
  - CLB (LUT)
  - Interconnect
  - PLL, DCM
  - **I/O**



Programmable Logic

- FPGA Design
  - Architecture Driven
    - Dedicated Multipliers
    - FFT
    - CPU
  - No Area Control
    - %1 utilization cost = %100 utilization cost
  - Limited Power Control

Semi Custom

- Logic Cells from Process
- Technology Specific
- Generic Gates
  - Function
  - Driving Capacity
- Well Characterized Delay & Transition
  - Supply
  - Temperature
  - Process Corner
  - Load
- Systematic Layout (LEGO Parts)

Full Custom

- Analog/RF design methods
- Design from scratch
- Custom schematic
- Custom layout

	Full-Custom	Semi-Custom	ASIC
			
Gate Sizing & logic opt.	All gates manually sized	Only gate that effect tiling	All Gates auto sized
Placement & Routing	Mostly custom	Only place & route critical parts of the design and let tool handle remaining	Mostly Auto
Development time	Months	Days / Weeks	Days
Itteration time	Days / weeks	Hours	Hours
Top level visibility	Black Box	Glass Box / White Box	White Box

Project Constraints

	Time to market	Volume	Performance
FPGA	☺	☹	☹
Semi Custom	😐	☺	😐
Full Custom	☹	☺	☺

Digital Circuit Representations

TOP DOWN

- Behavioral
  - Define what the digital circuit does
- Register Transfer Level (RTL)
  - Define the logical operations of busses and wires
- Structural
  - Define the gates and the connections between them
- Gate Level
  - Schematic representation at the gate level
- Transistor Level
  - Schematic representation at the transistor level

BOTTOM UP

Digital Circuit Representations

- Behavioral

```
module FA_behavioral(a, b, carry_in, carry_out, sum);

        input a, b, carry_in;
        output carry_out, sum;

        reg [1:0] tmp;
        reg carry_out, sum;

        always@(a or b or carry_in) begin
                tmp = a + b + carry_in;
                sum = tmp[0];
                carry_out = tmp[1];
        end
endmodule
```

Digital Circuit Representations

- Register Transfer Level (RTL)

```
module FA_RTL(a, b, carry_in, carry_out, sum);

        input a, b, carry_in;
        output carry_out, sum;

        wire x1, x2, x3;

        assign x1 = a ^ b;
        assign sum = x1 ^ carry_in;
        assign x2 = x1 & carry_in;
        assign x3 = a & b;
        assign carry_out = x2 | x3;

endmodule
```

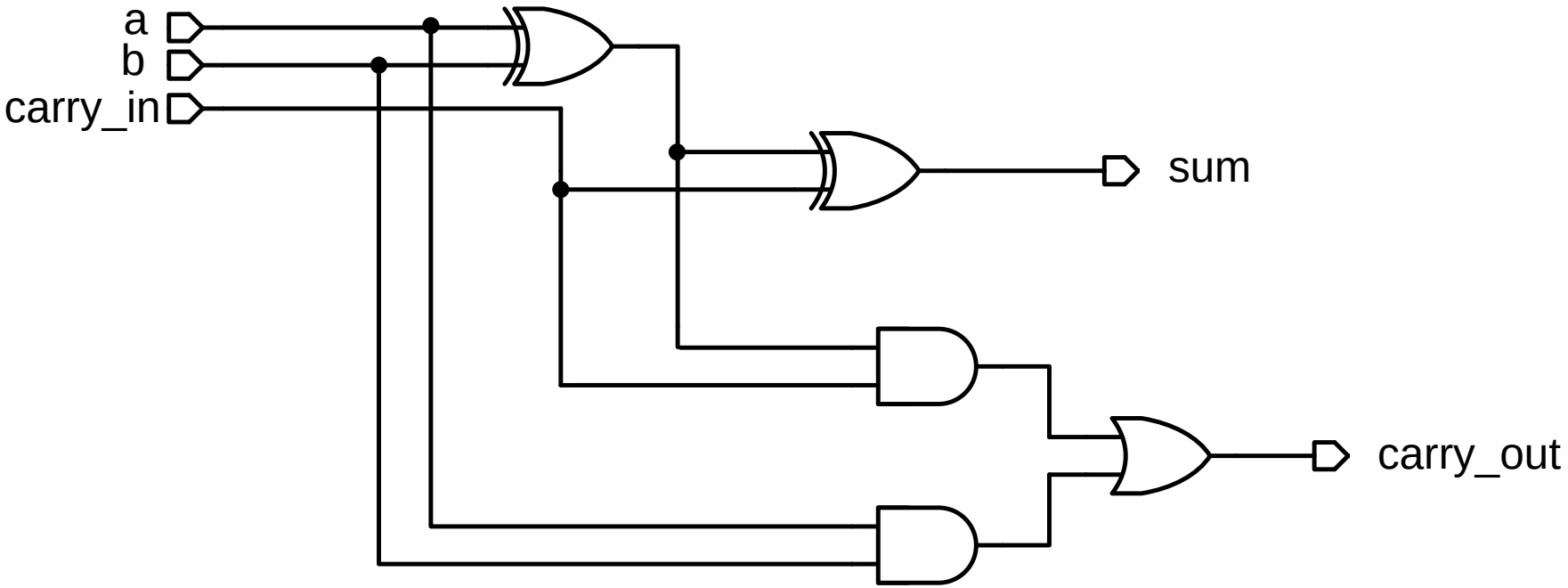
Digital Circuit Representations

- Structural

```
module FA_structural(a, b, carry_in, carry_out, sum);

        input a, b, carry_in;
        output carry_out, sum;

        wire x1, x2, x3;

        XOR U1(a, b, x1);
        XOR U2(x1, carry_in, sum);
        AND U3(x1, carry_in, x2);
        AND U4(a, b, x3);
        OR U5(x2, x3, carry_out);

endmodule
```

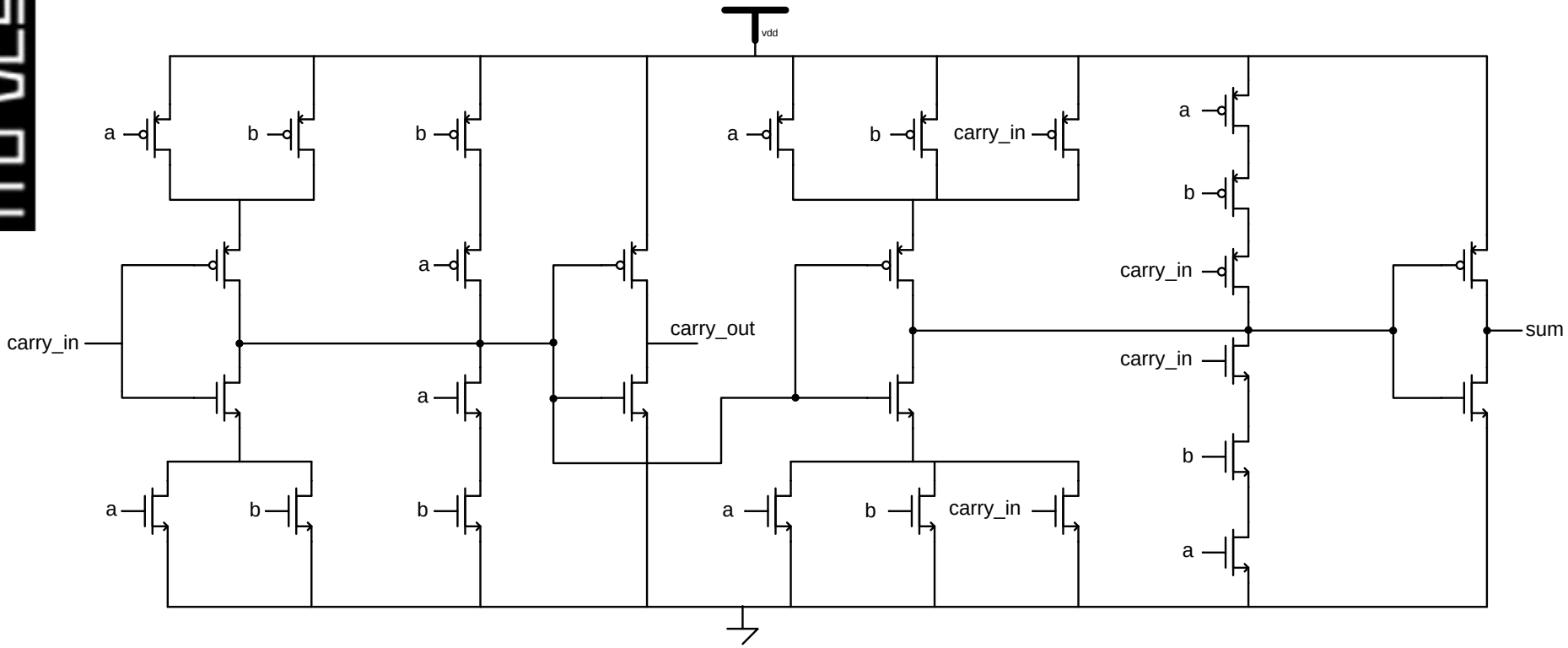
Digital Circuit Representations

- Gate Level



Digital Circuit Representations

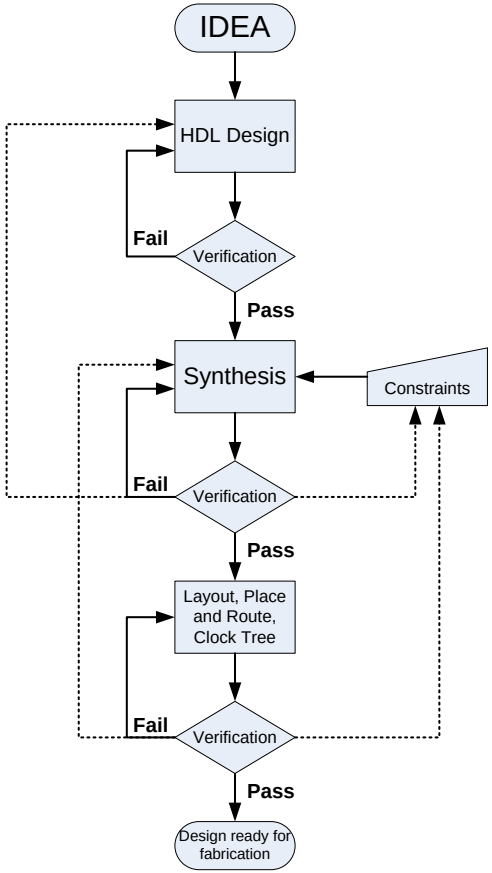
- Transistor Level



Digital Design Considerations

- Digital Design ≠ “Let do and let pass, the world goes on by itself!” Vincent de Gournay
- Digital Design ≠ Do-it-all tools
- One should:
  - Consider code style
  - Consider coding for an architecture
  - Investigate the tool outputs
  - Know how synthesizer thinks

Digital Design Flow



Use human brain to come up with something
Use text editor to write the HDL code
Functional verification (Behavioral Simulation)
Use synthesizer to convert behavioral code to structural code
Functional and Timing Verification (Post-Synthesis Simulation)
Place and Route tool converts structural code to physical layout
Functional and Timing Verification (Post-PAR Simulation)
Send to fab for manufacturing

References

- http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-111Spring2004/F19D2F2D-5FE9-4C59-AA67-D2113639B022/0/l6_fsm_toCMS_080904.pdf
- http://asic-soc.blogspot.com/2008/12/transition-delay-and-propagation-delay.html
- http://asic-soc.blogspot.com/2008/10/net-delay.html
- http://people.bu.edu/rknepper/sc571/chapter4_b.ppt
- http://www.xilinx.com/publications/products/cpld/logic_handbook.pdf
- http://en.wikipedia.org/wiki/Adder_(electronics)
- http://www.askkia.com/articles/how-to-create-a-single-bit-full-adder.html