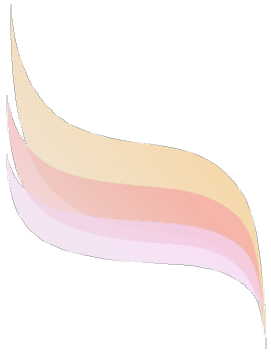
Digital System Design

Lecture 1
Design Concepts and Programmable Logic

Design Concepts



Reading Assignment:

- Brown, “Fundamentals of Digital Logic with VHDL, pp. 2 - 16, pp. 56 - 65

Learning Objective:

- A general introduction to the process of designing digital systems
- Explain CAD tools

Chip Complexity

- 1963: transistor size = 50μm
- 1975: transistor size = 10μm
- 1985: transistor size = 2μm
- 1995: transistor size = 0.35μm
- 2004: transistor size = 90nm
- 2012: transistor size = 36nm
- 2018: transistor size = 10nm

(1um = 10^3 nm)

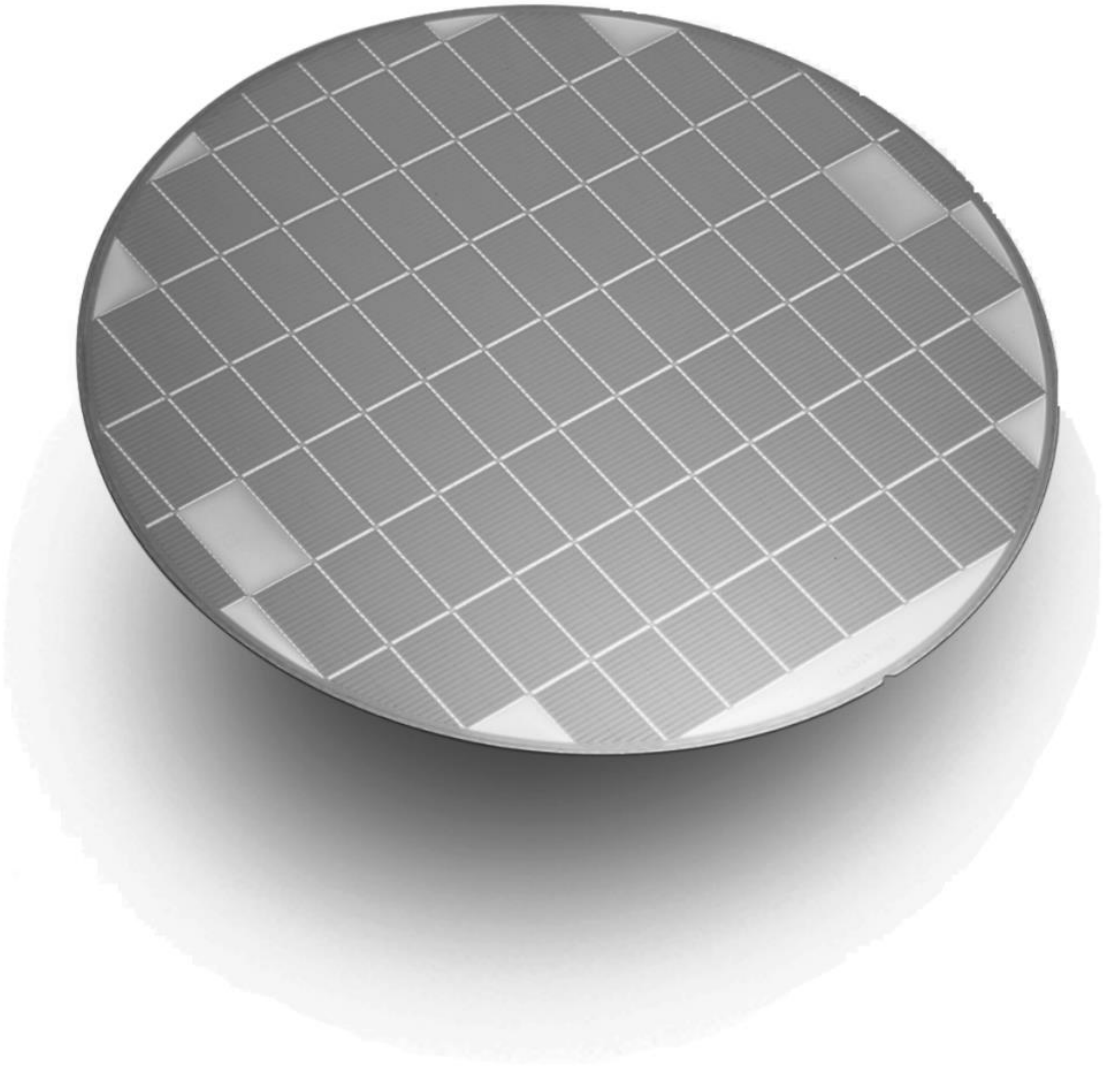
decreasing

SIA Roadmap

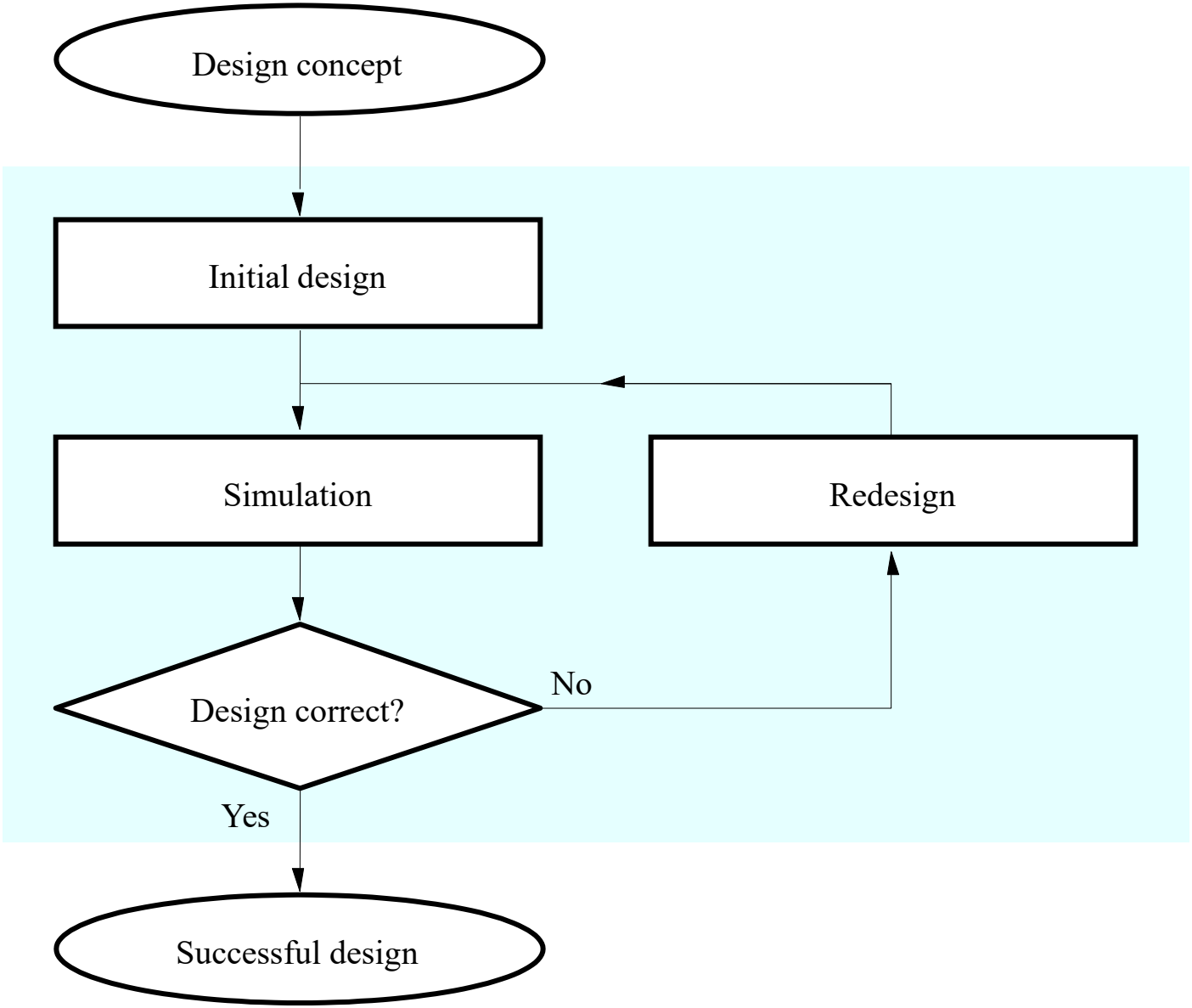
Table 1.1 A sample of the International Technology Roadmap for Semiconductors.

	Year					
	2006	2007	2008	2009	2010	2012
Technology feature size	78 nm	68 nm	59 nm	52 nm	45 nm	36 nm
Transistors per cm^2	283 M	357 M	449 M	566 M	714 M	1,133 M
Transistors per chip	2,430 M	3,061 M	3,857 M	4,859 M	6,122 M	9,718 M

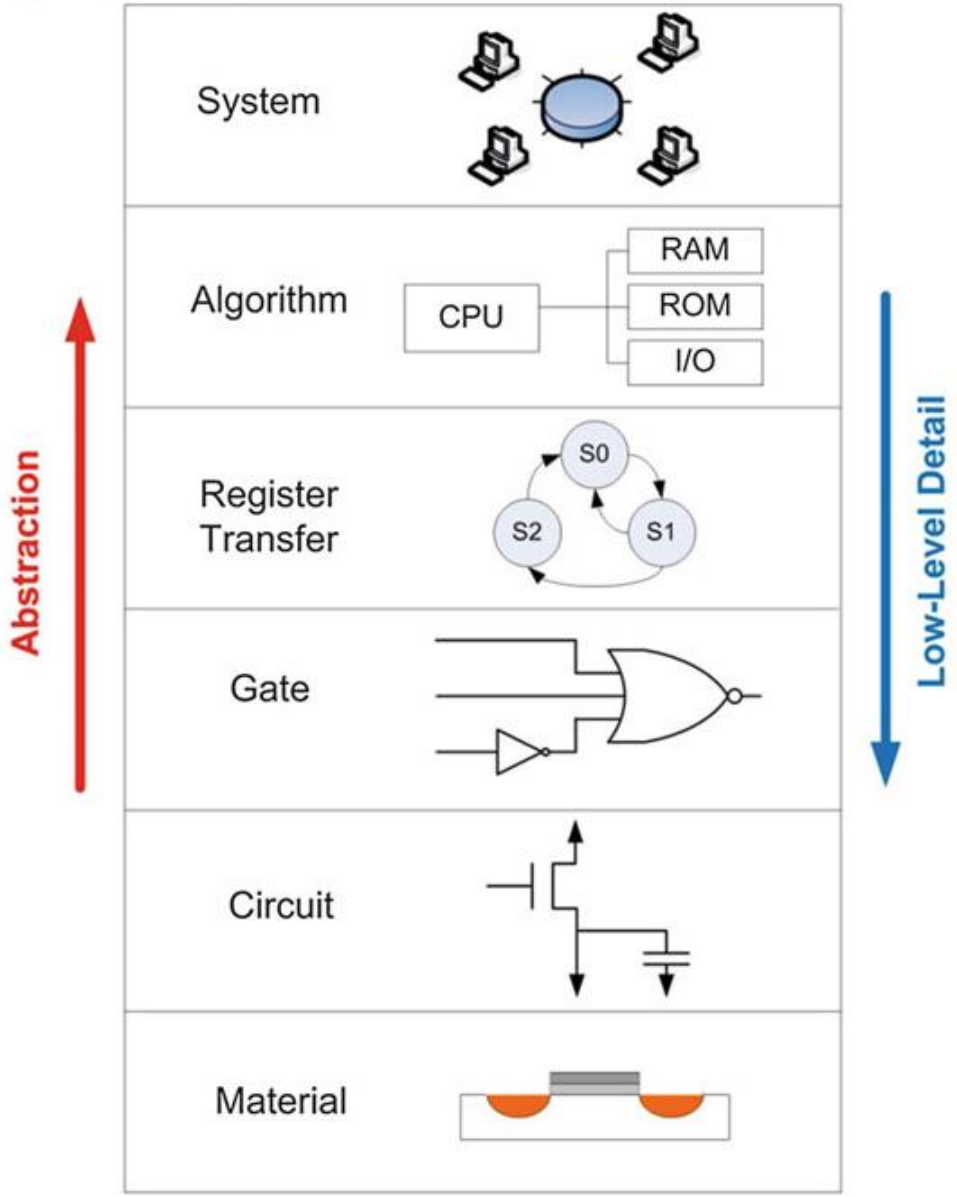
A silicon wafer



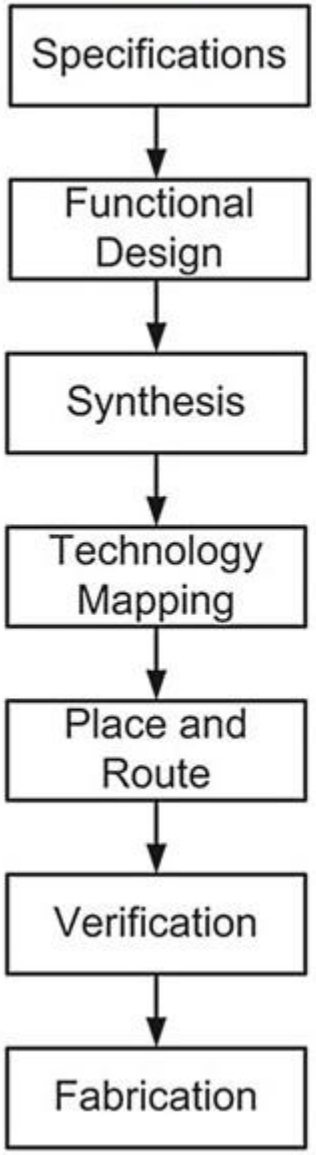
The basic design loop



Levels of design abstraction



Generic design flow



Description of Tasks at Each Step

- State the desired behavior of the design using broad, high-level specifications.

- Describe the high-level architecture of the design (e.g., block diagrams for inputs/outputs, sub-systems) and generic behavior (truth tables, state diagrams and/or algorithms).

- Create the gate-level connection (schematic or netlist) of the design using logic synthesis processes (e.g., K-maps or automated CAD tools).

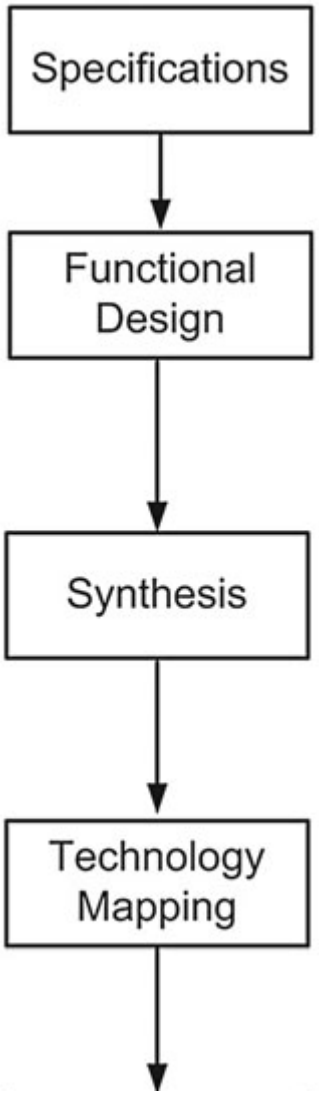
- Select the logic technology that will achieve the specifications (e.g., 74HC family, 32nm CMOS ASIC). Manipulate the gate-level netlist/schematic into a form that is suitable for this technology (e.g., DeMorgan's NAND/NOR).

- Arrange the components to minimize the area needed (on a board or chip) and wire all connections to minimize interconnect length and crossings.

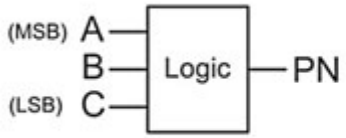
- Once a technology is chosen and the routing is complete, the gate and wiring delays can be used to estimate whether the final design meets the timing and power consumption requirements of the original specifications.

- Once the design is verified it can be implemented. (ASIC, programmable device, board-level, discrete parts)

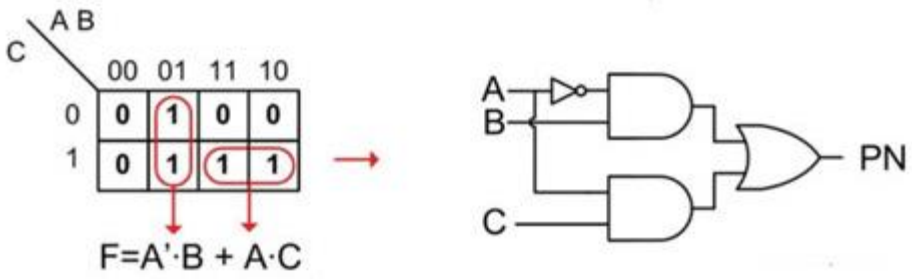
Classical design flow



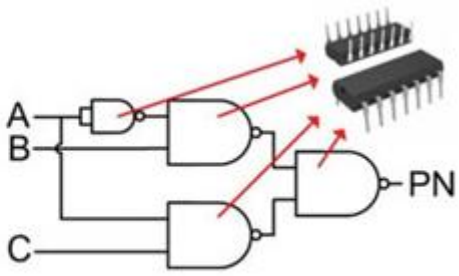
- Design a "Prime Number Detector" that takes in values from 0_{10} to 7_{10}. The circuit should be able to indicate a prime number with a delay less than 200ns.



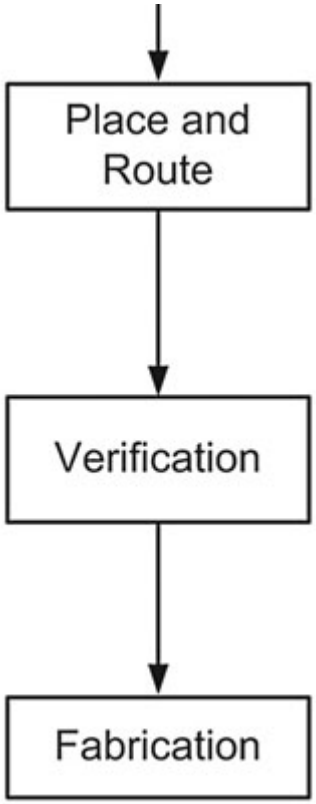
A	B	C	PN	
0	0	0	0	
0	0	1	0	
0	1	0	1	← 2
0	1	1	1	← 3
1	0	0	0	
1	0	1	1	← 5
1	1	0	0	
1	1	1	1	← 7



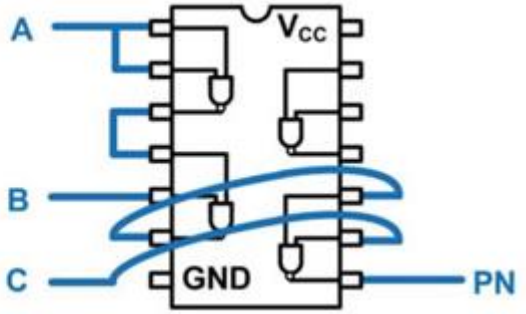
- It is decided that a 74HC logic family will be the most cost-effective technology for this design. To minimize the number of parts, the logic will be implemented with only NAND-gates.



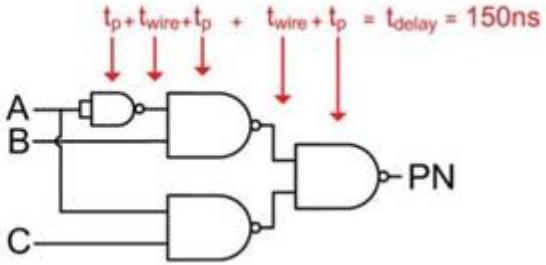
Classical design flow



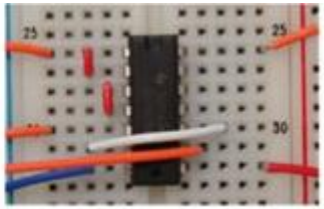
- The circuit to be implemented is placed in a floor plan and an estimate of the connections are made.



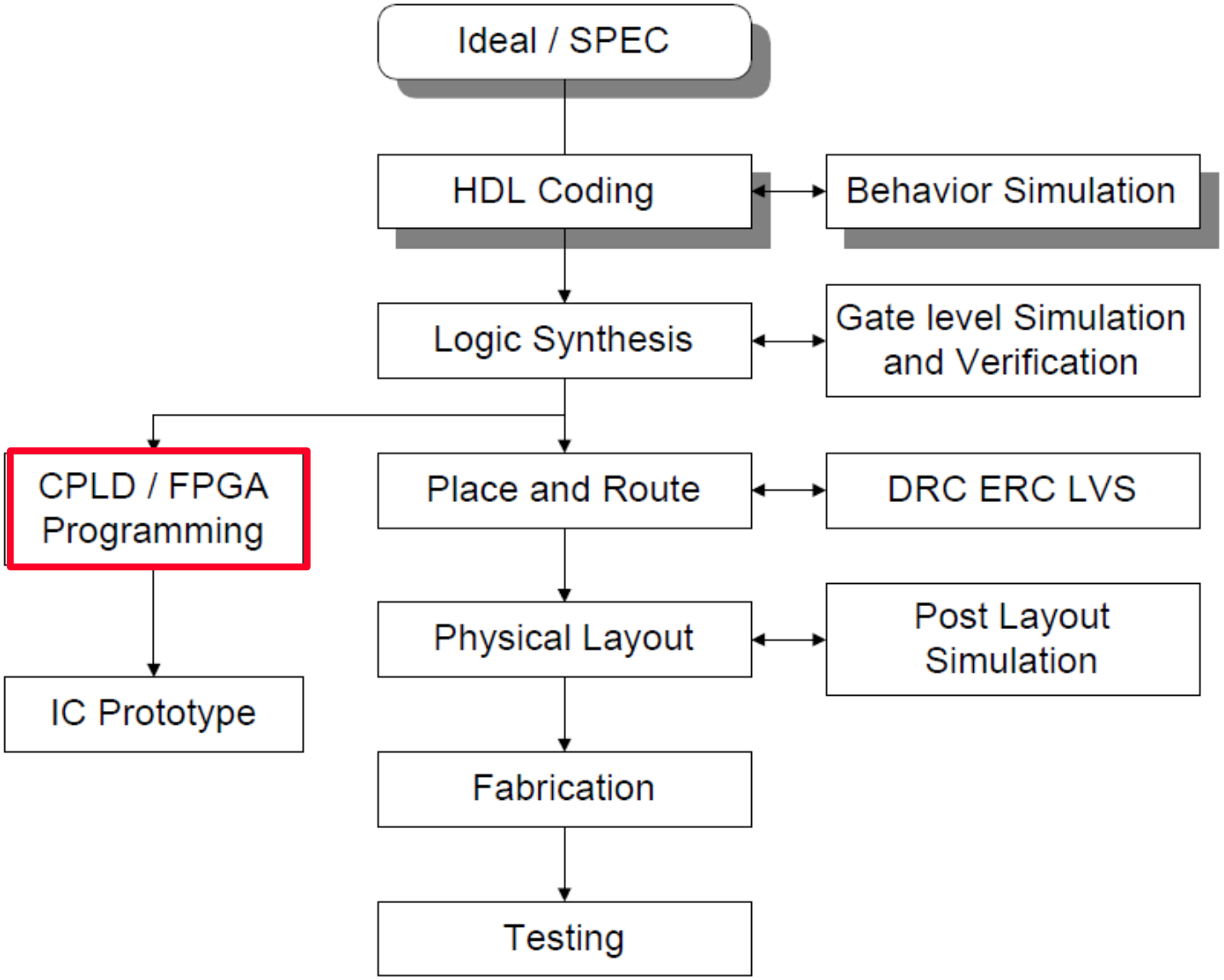
- Based on the layout, the wire delays are found. The delays of the gates are taken from the data sheet.



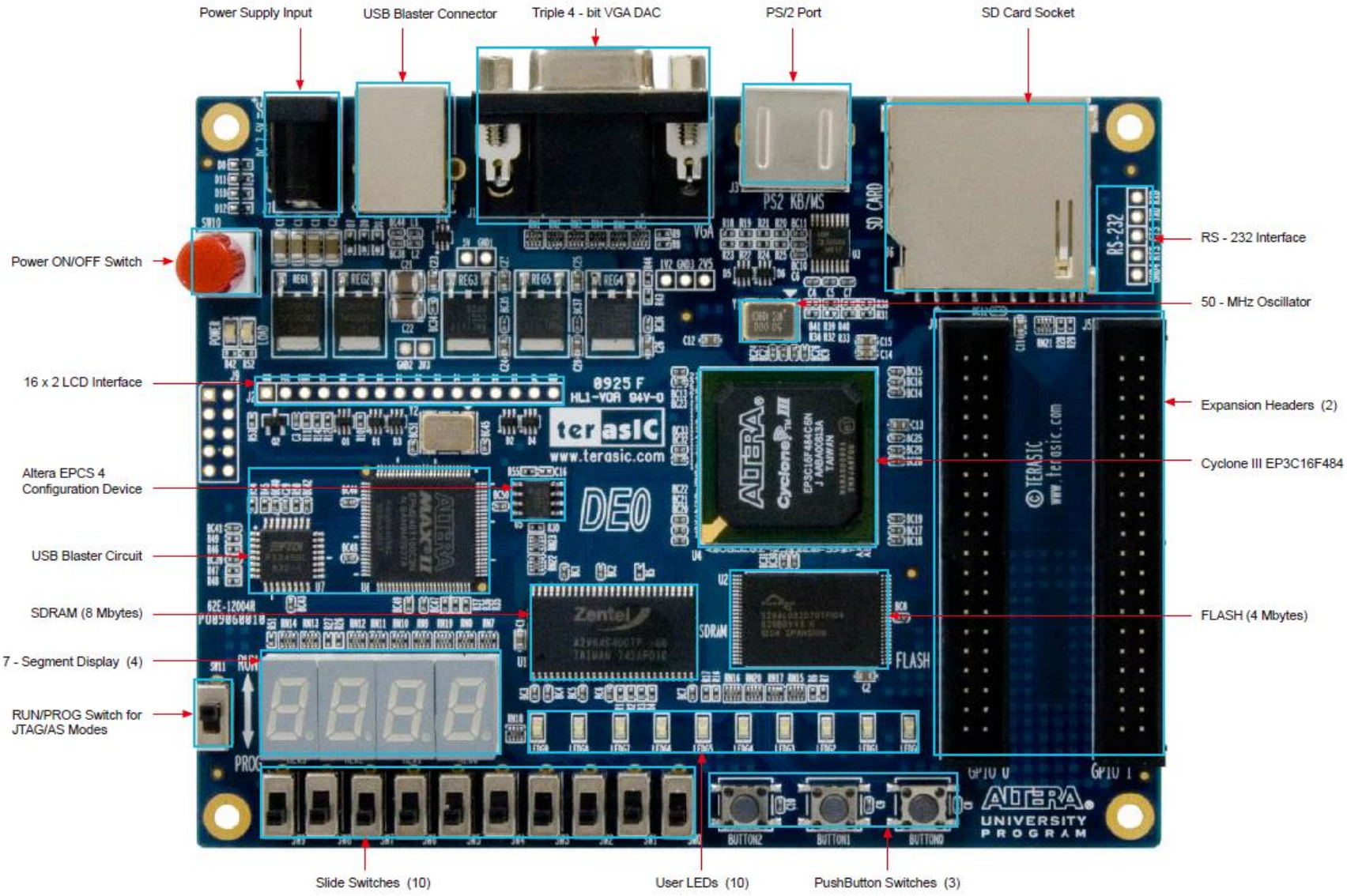
- The verified circuit is implemented in hardware.



Modern design flow



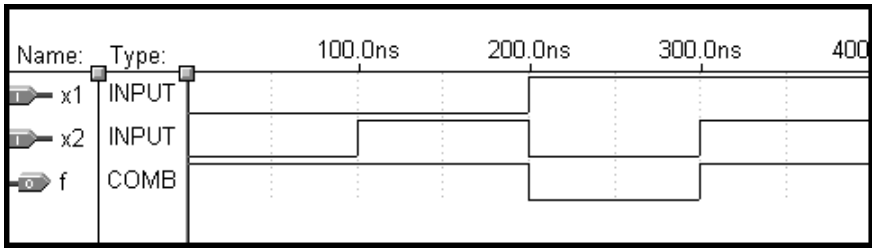
Altera DE0

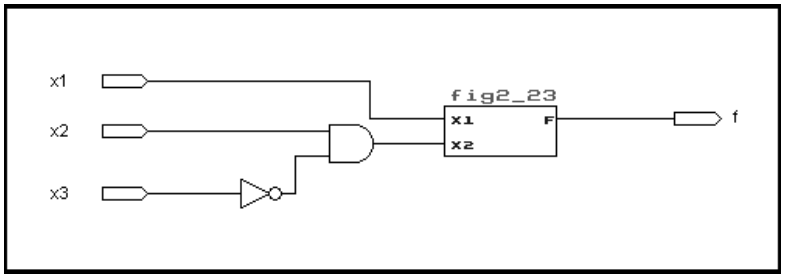

Power Supply Input
USB Blaster Connector
Triple 4 - bit VGA DAC
PS/2 Port
SD Card Socket
Power ON/OFF Switch
RS - 232 Interface
50 - MHz Oscillator
16 x 2 LCD Interface
Expansion Headers (2)
Altera EPCS 4 Configuration Device
Cyclone III EP3C16F484
USB Blaster Circuit
SDRAM (8 Mbytes)
FLASH (4 Mbytes)
7 - Segment Display (4)
RUN/PROG Switch for JTAG/AS Modes
Slide Switches (10)
User LEDs (10)
PushButton Switches (3)
DE0
terasIC
www.terasic.com
PS2 KB/MS
SD CARD
RS-232
VGA
SDRAM
FLASH
GPIO 0
GPIO 1
BUTTON2
BUTTON1
BUTTON0
ALTERA UNIVERSITY PROGRAM

Introduction to CAD tools

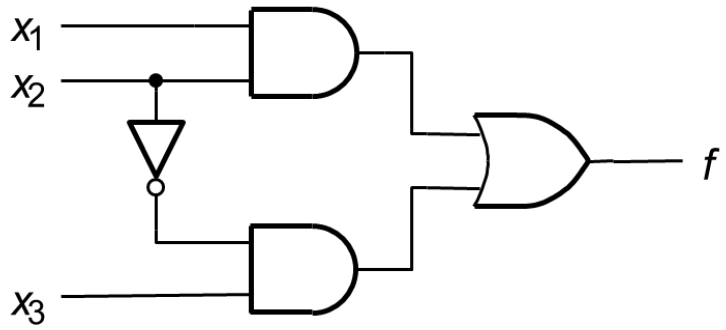
Design Entry

- Truth tables
  - Using Waveform Editor
- Schematic capture
  - Interconnect symbols in some *library*.
- Hardware description languages (HDLs).
  - Similar to a programming language.
  - VHDL and Verilog HDL.
  - Allow for sharing and design reuse.
  - Support hierarchical design.





Introduction to CAD tools (cont.)



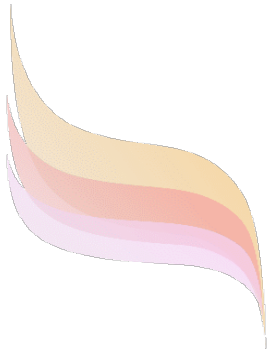

```
ENTITY example1 IS
      PORT ( x1, x2, x3   : IN     BIT ;
             f            : OUT    BIT ) ;
END example1 ;

ARCHITECTURE LogicFunc OF example1 IS
BEGIN
      f <= (x1 AND x2) OR (NOT x2 AND x3) ;
END LogicFunc ;
```

Introduction to CAD tools (cont.)

- **Logic synthesis or logic optimization**
  - is process to translate a truth table, schematic, or VHDL code into a network of logic gates.

- **Simulation**
  - Functional simulation is used to determine if designed circuit operates functionally correct.
    - Zero delay
  - Timing simulation is used to check the performance of a design
    - Real delay

Programmable Logic



Reading Assignment:

- Brown, “Fundamentals of Digital Logic with VHDL, pp. 98 - 114

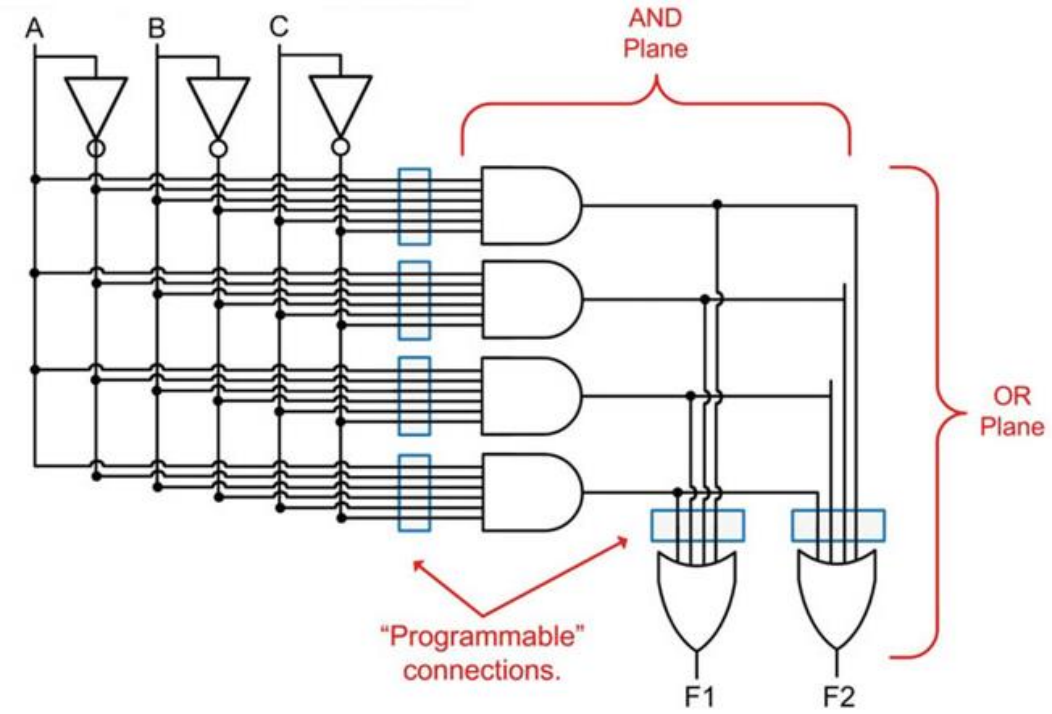
Learning Objective:

- Describe the genesis of programmable logic devices
- List the differences between complex programmable logic devices (CPLDs) and field programmable gate arrays (FPGAs) and describe the basic organization of each

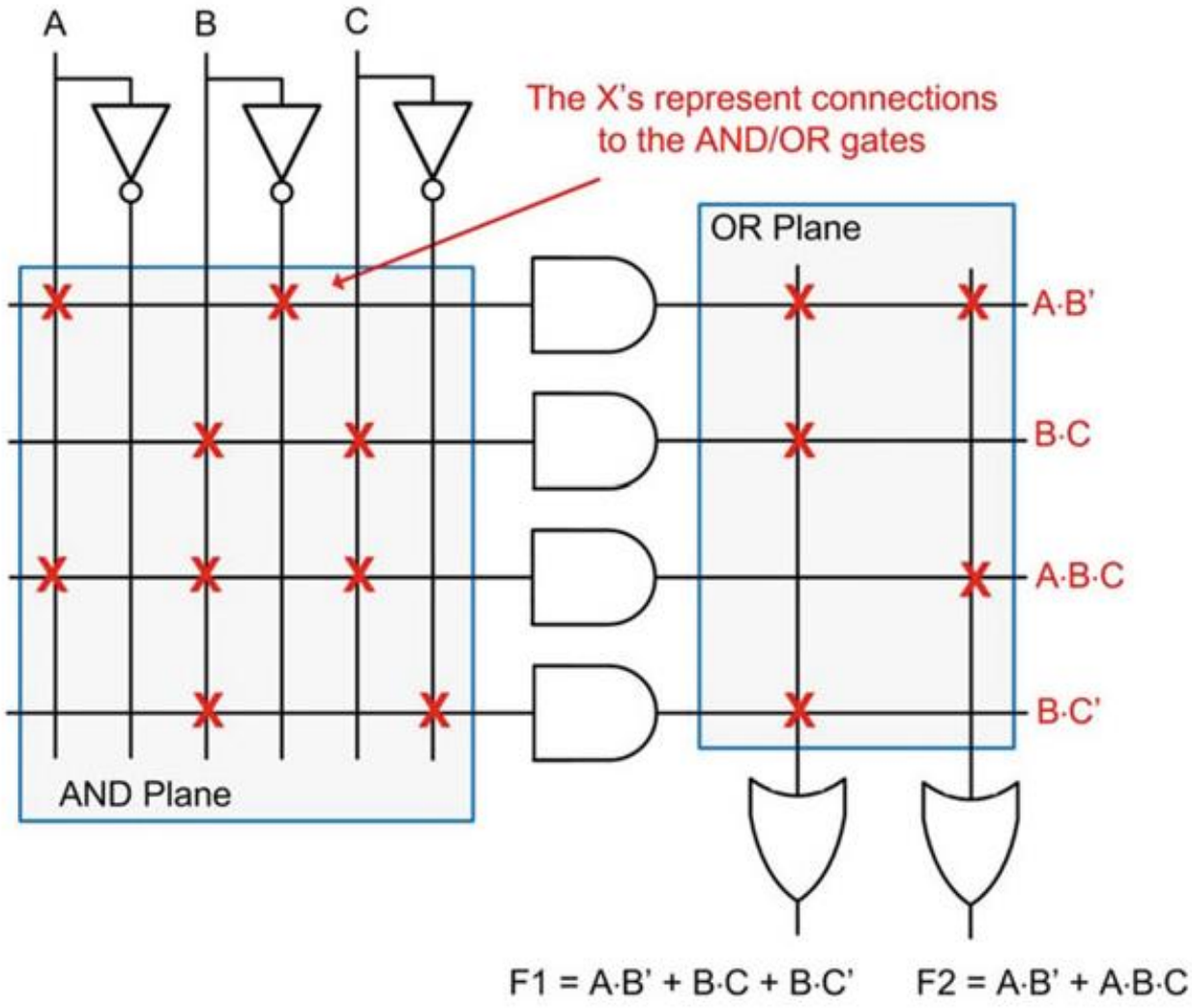
Overview

PLA ⊆ PLD

- The first programmable logic devices (PLDs) were programmable logic arrays (PLAs)
- PLAs are combinational, two-level AND-OR devices that can be programmed to realize sum-of-products expression
- General structure of a PLA

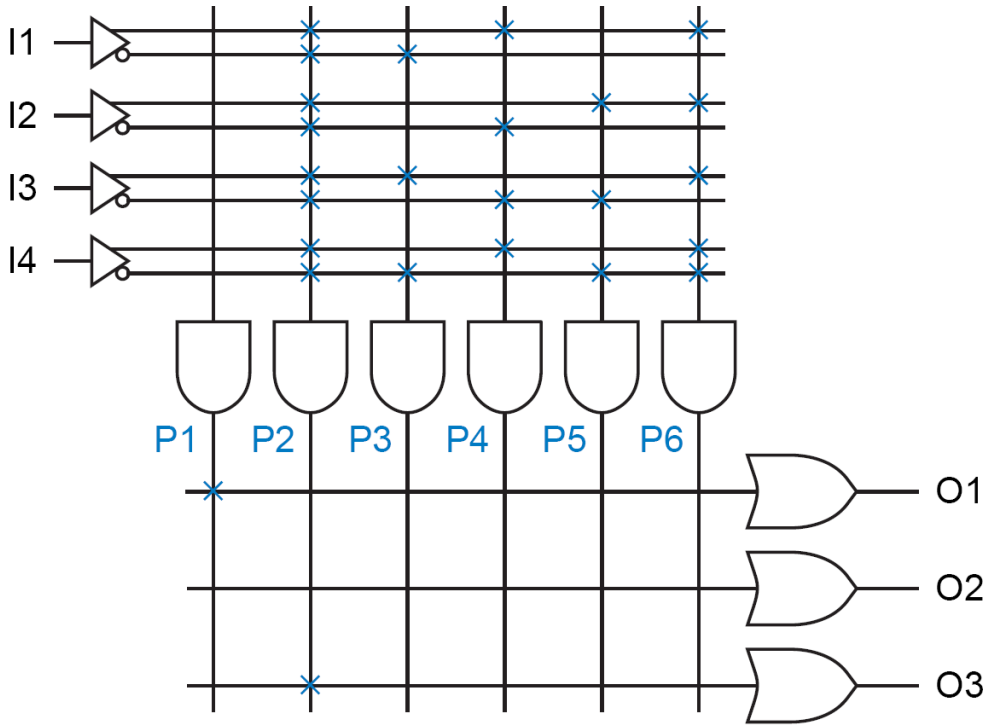


Simplified Programmable Logic Array (PLA)



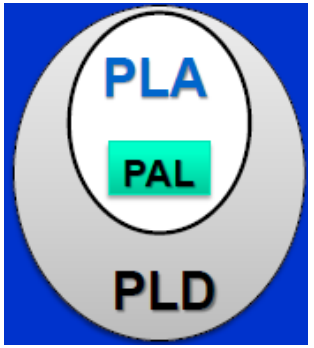
Constants

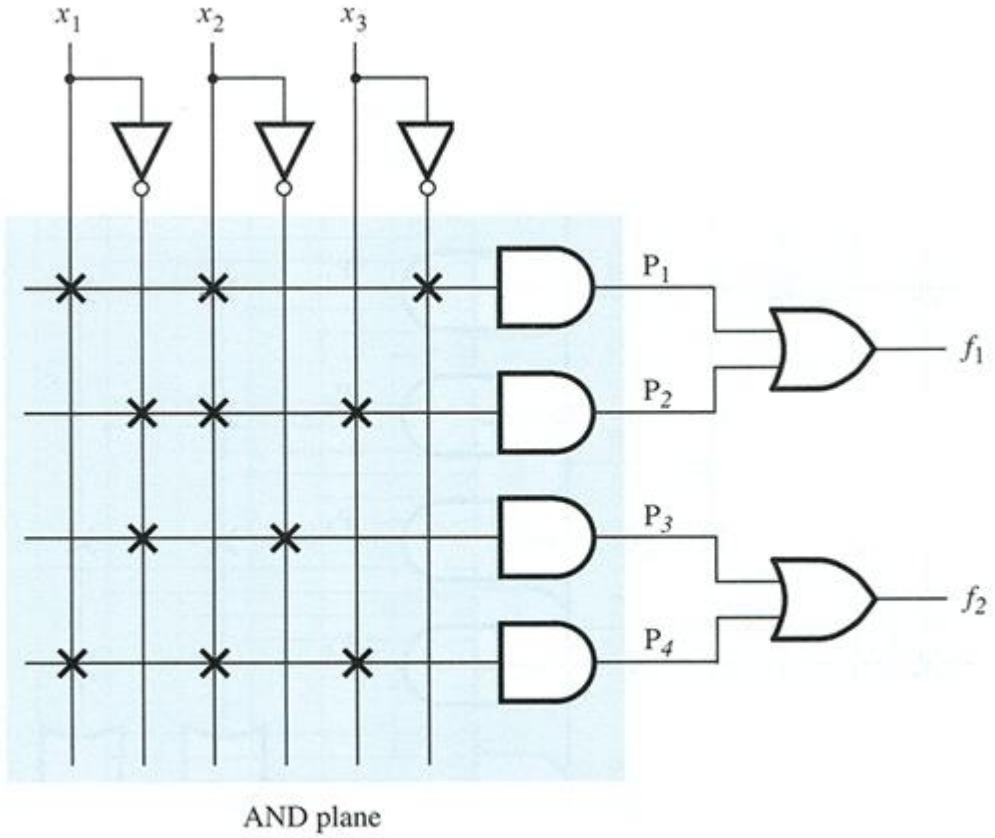
- Sometimes a PLA output must be programmed to be a constant 1 or a constant 0.
  - P1 is always 1 because its product line is connected to no inputs and is therefore always pulled HIGH;
  - this constant-1 term drives the O1 output.
- No product term drives the O2 output, which is therefore always 0.
- Another method of obtaining a constant-0 output is shown for O3.



Programmable Array Logic (PAL)

- A special case of PLA is the programmable array logic (PAL)
- Unlike a PLA, a PAL device has a fixed OR array (i.e. AND gates can not be shared)



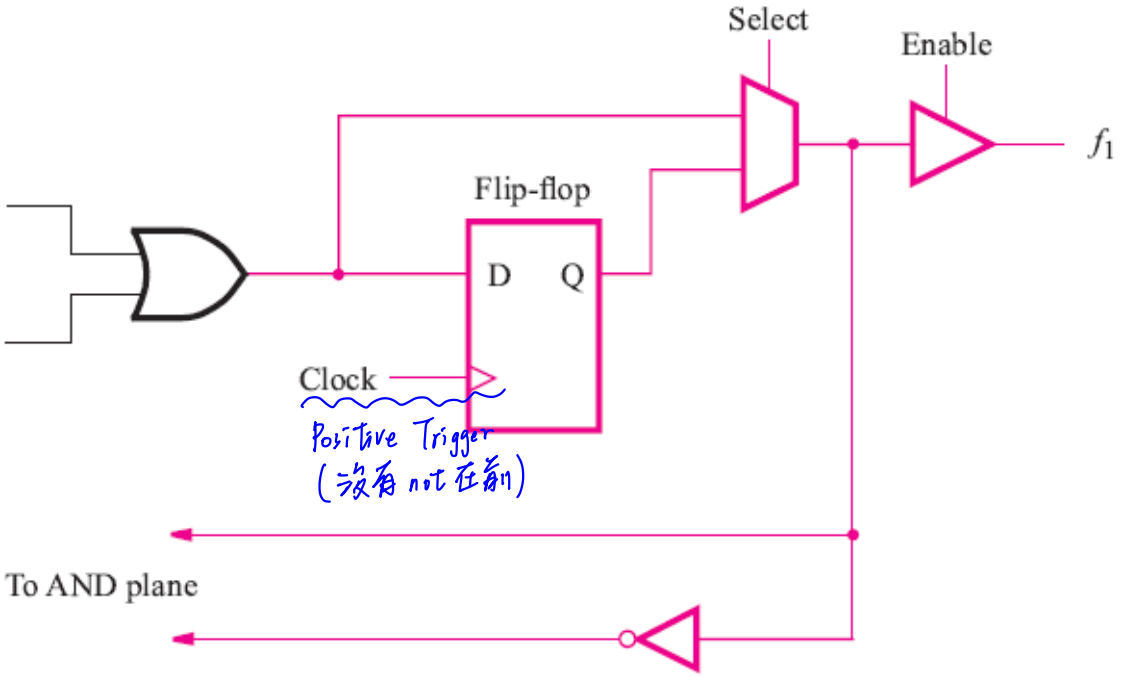


An example of a PAL

- PAL v.s. PLA

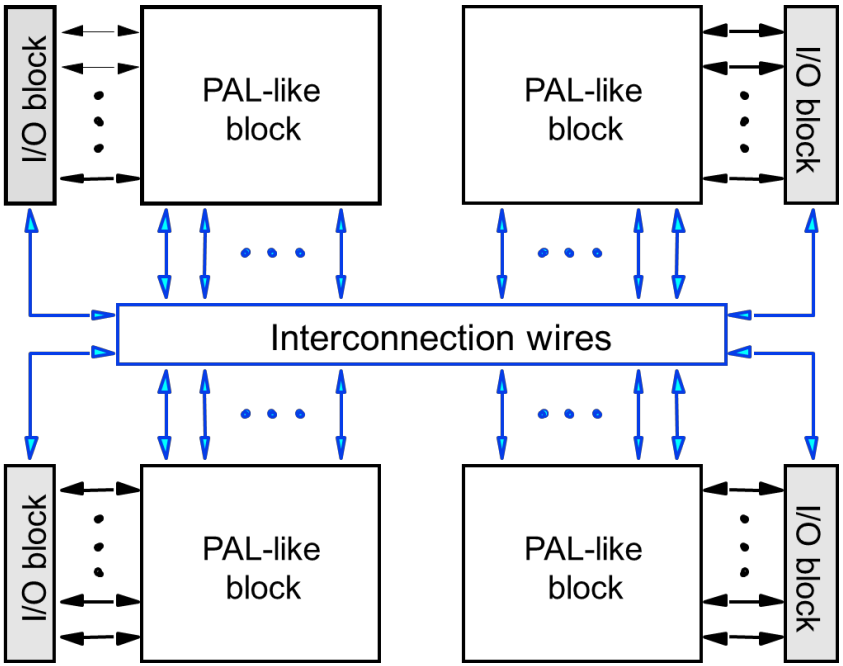
Macrocell (一組)

- In many PALs, extra circuitry is added at the output of each OR gate to provide additional flexibility.
- It is customary to use the term macrocell to refer to the OR gate combined with the extra circuitry.

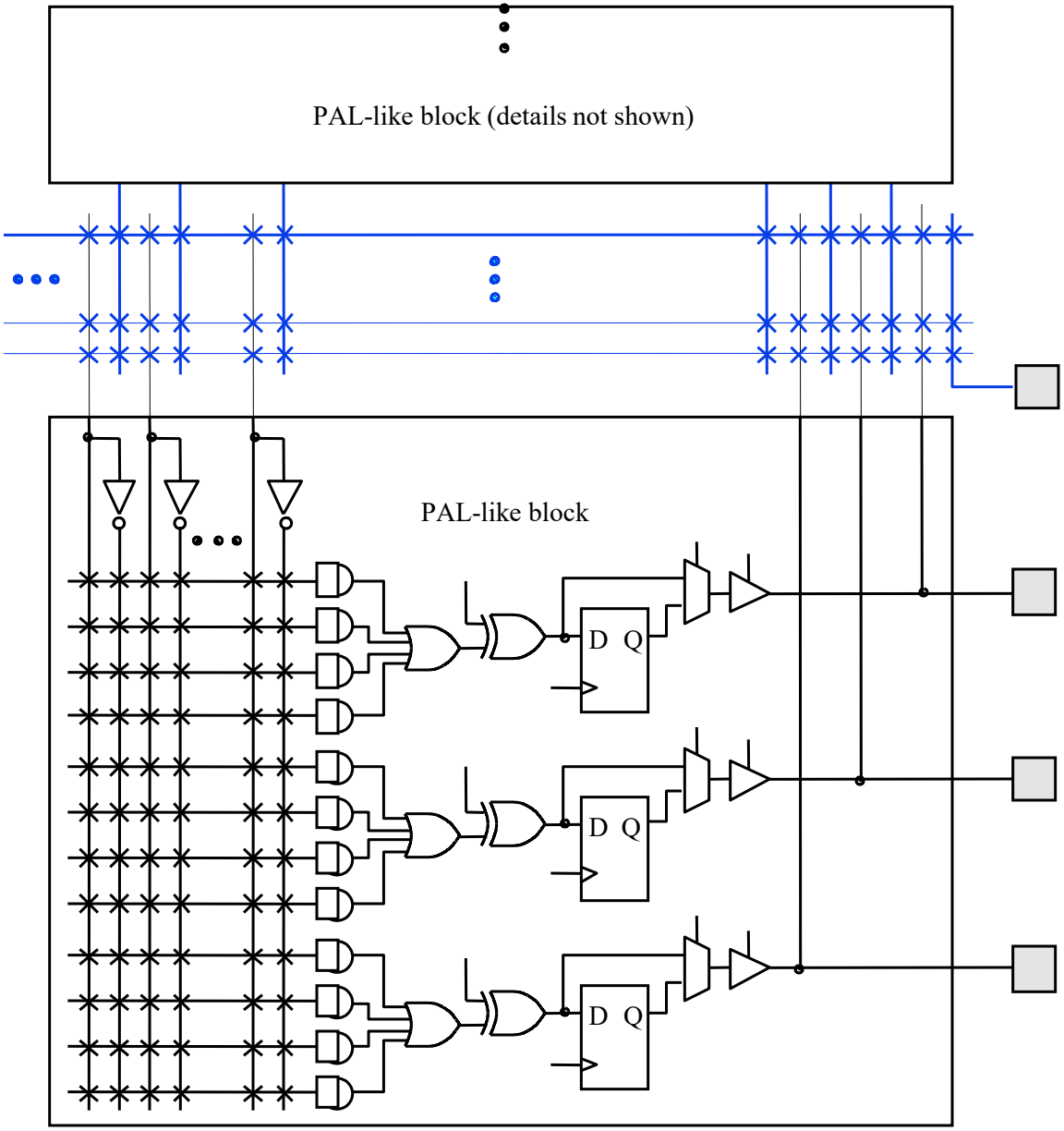


Complex PLDs (CPLDs)

- PLAs and PALs are limited to fairly modest sizes, typically supporting a combined number of inputs plus outputs of not more than 32.
- For implementation of circuits that require more inputs and outputs, a more sophisticated type of chip, called a complex programmable logic device (CPLD), can be used
- A CPLD comprises multiple PAL-like circuit blocks on a single chip, with internal wiring resources to connect the circuit blocks.

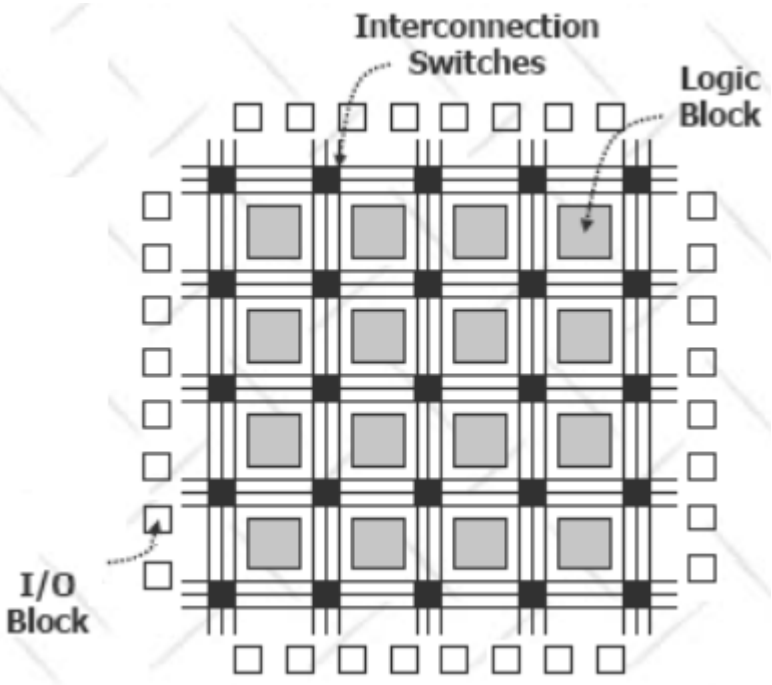


A section of a CPLD(3 macrocell)



Field Programmable Gate Array (FPGA)

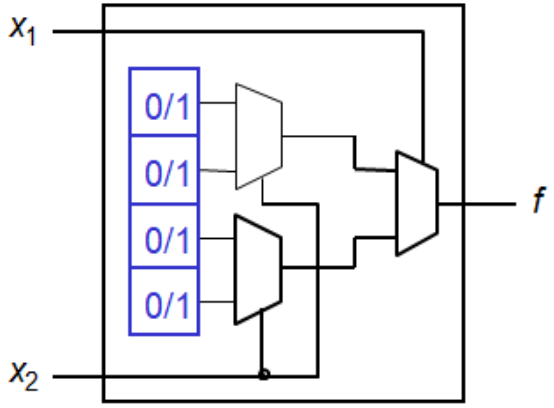
- By modern standards, a logic circuit with 10,000 gates is not large. To implement larger circuits, it is convenient to use a different type of chip that has a larger logic capacity.
- Field-programmable gate array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuits.
- FPGAs provide logic blocks for implementation of the required functions.



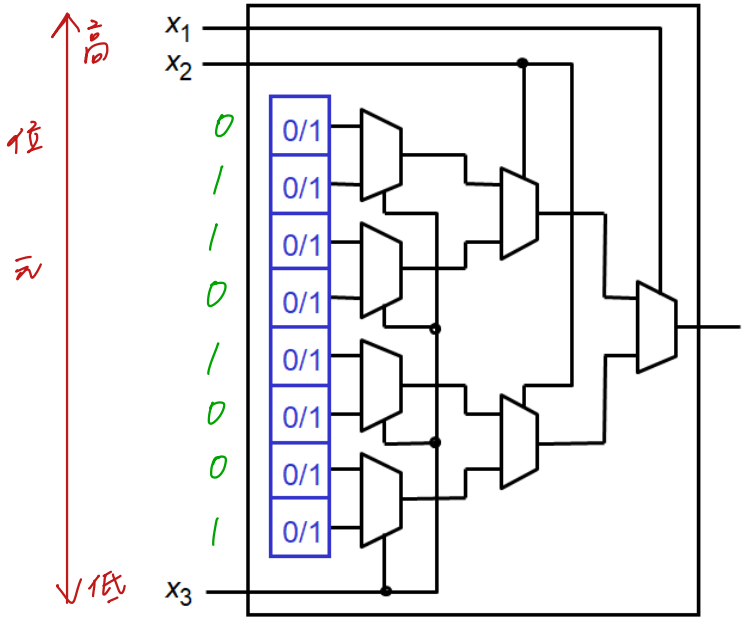
Field Programmable Gate Array (FPGA)

- Each logic block in an FPGA typically has a small number of inputs and outputs.
- The most commonly used logic block is a lookup table (LUT), which contains storage cells that are used to implement a small logic function.
- Each cell is capable of holding a single logic value, either 0 or 1.
- LUTs of various sizes may be created, where the size is defined by the number of inputs.

2 variable's LUT



高位元放靠近輸出



0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

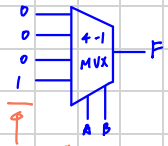
可知為 $(A \oplus B \oplus C)$

" odd 1's output 1

otherwise → 0

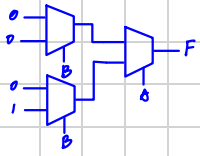
Using MUX implement AND Gate

- 4-1 MUX



设置值走的结果

- 2-1 MUX

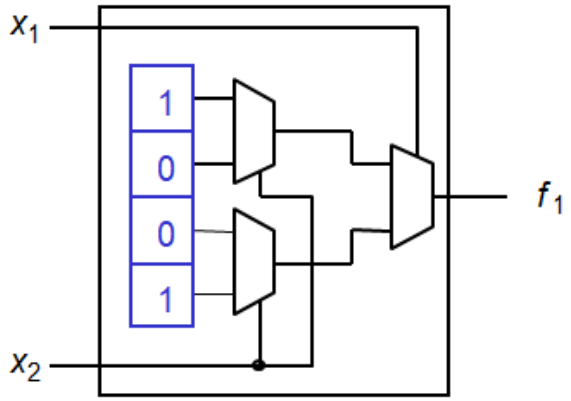


Field Programmable Gate Array (FPGA)

- A two-input lookup table (LUT).

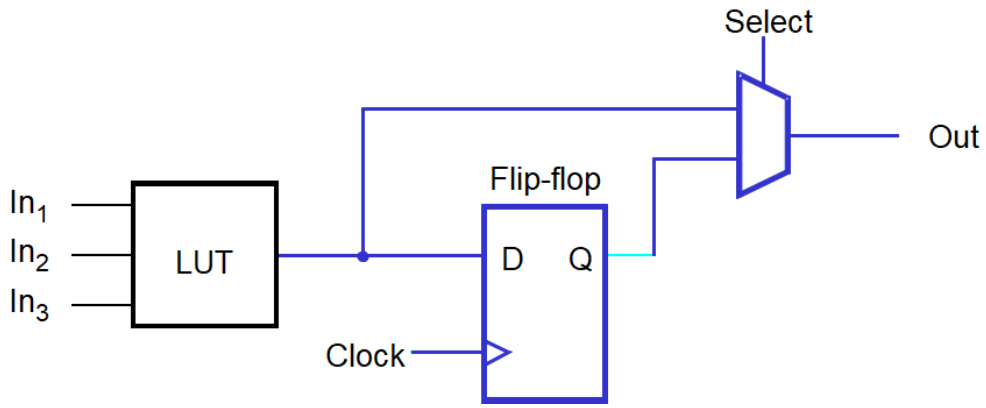
x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$



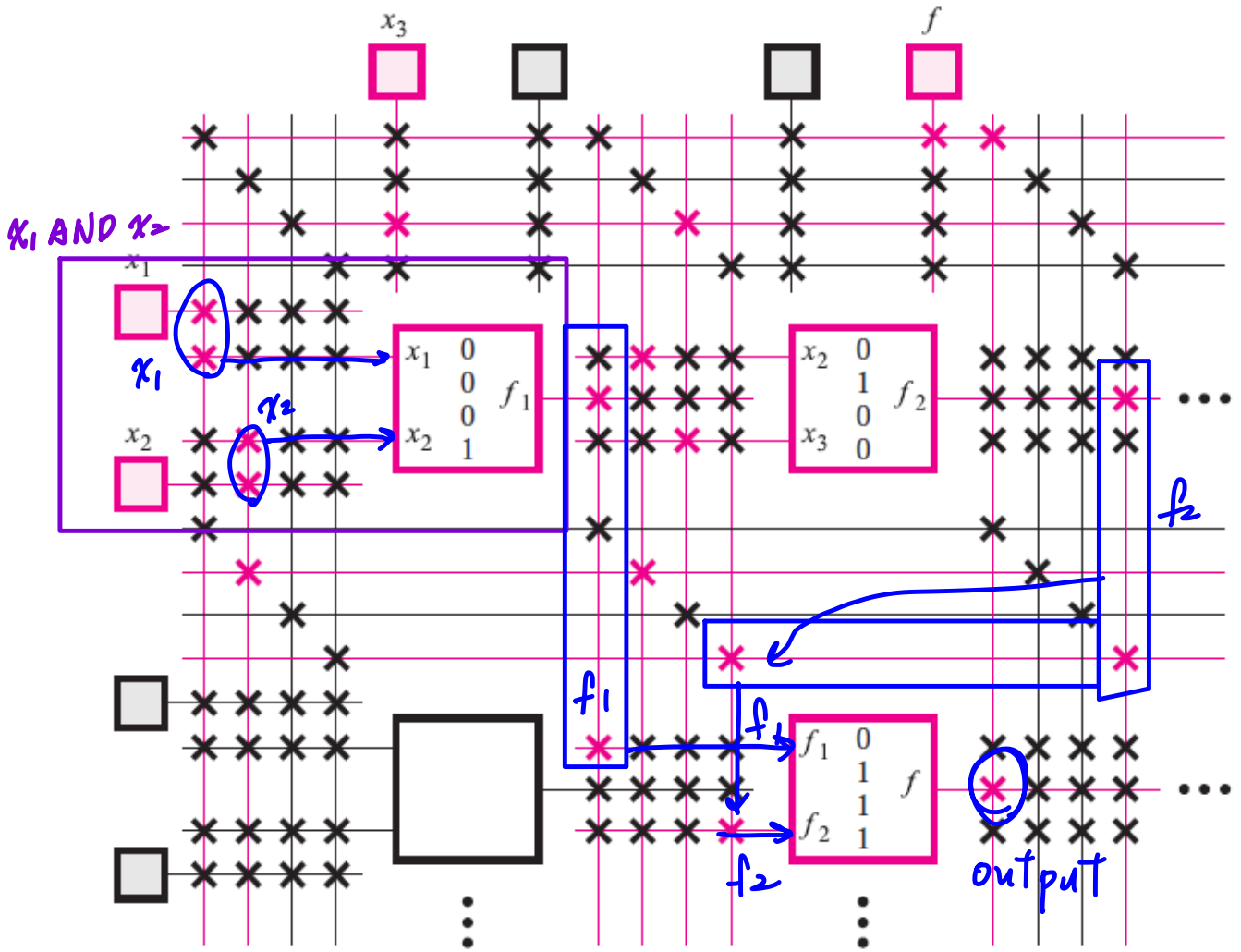
(c) Storage cell contents in the LUT

- Inclusion of a flip-flop with a LUT



A section of a programmed FPGA

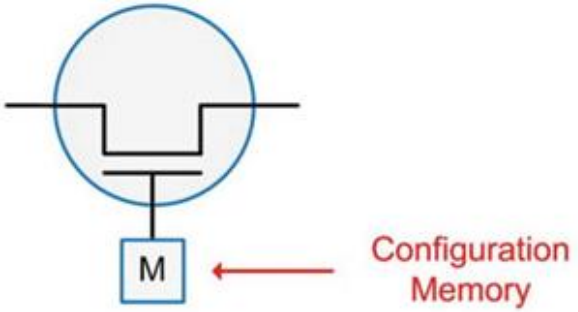
$f = f_1 + f_2 = x_1x_2 + x_2'x_3$

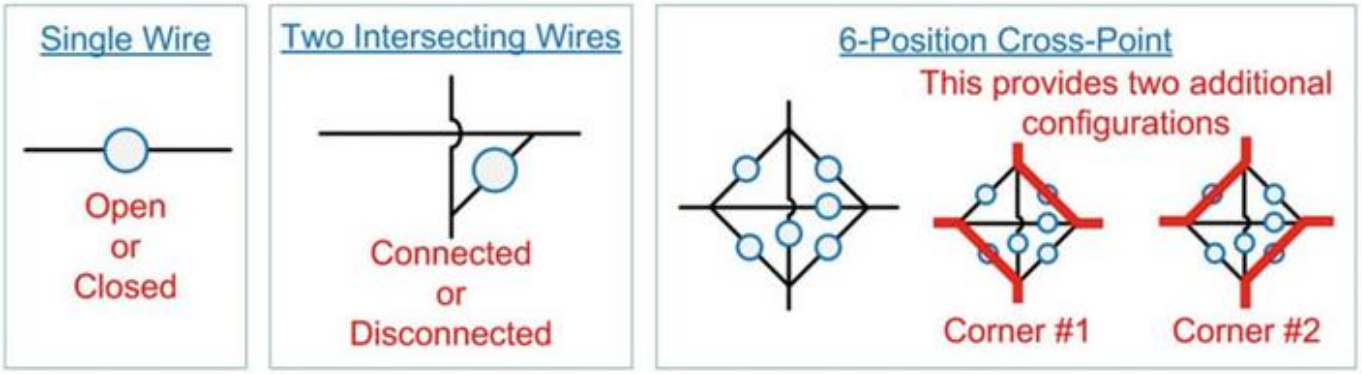


FPGA Programmable Interconnect

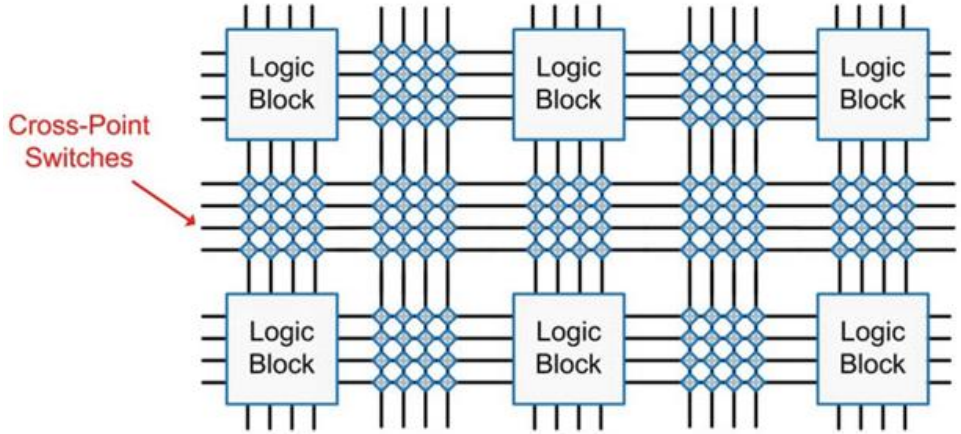
A simple model for a programmable interconnection is an NMOS transistor that connects or disconnects two wires. The switch is controlled using a configuration bit.

This can be used in a variety of configurations.

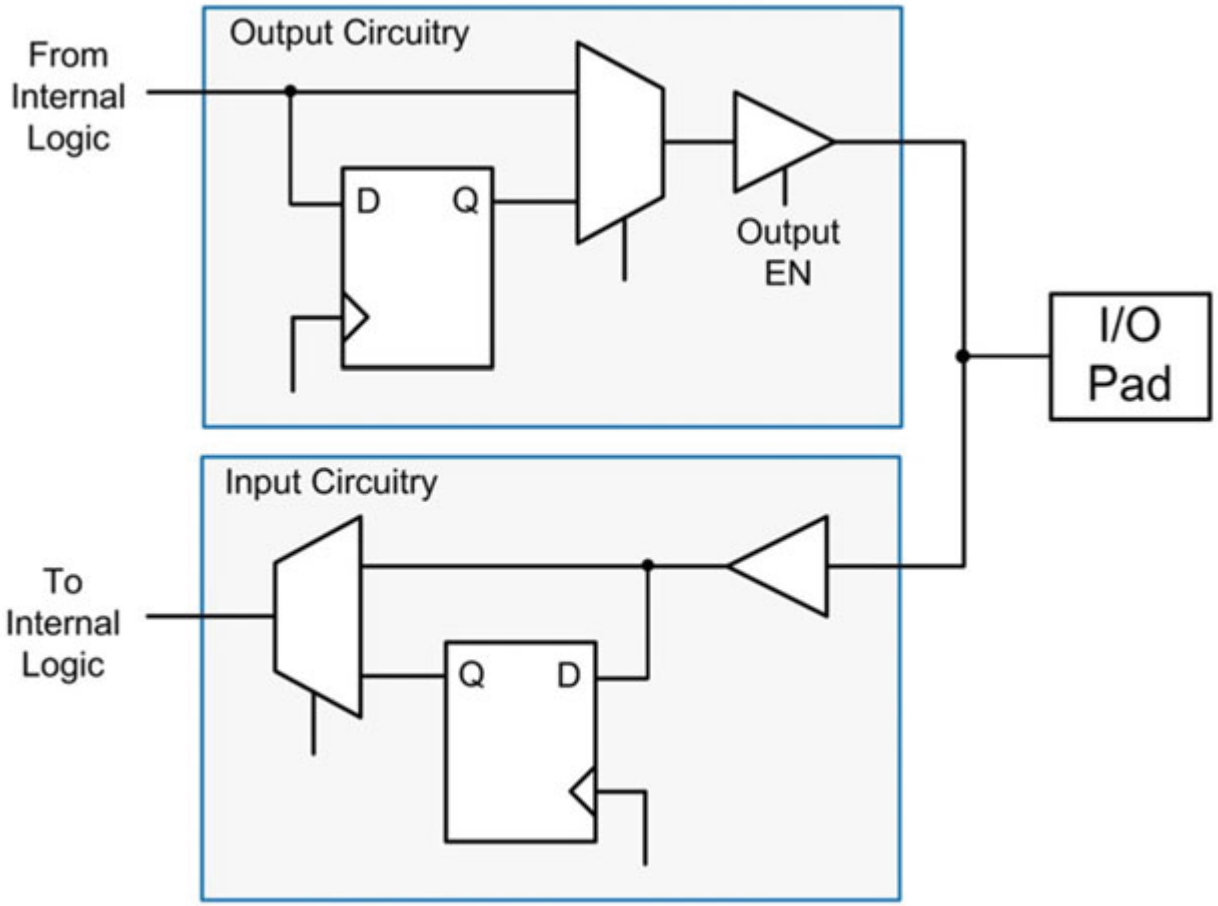




The switches are placed at the intersections of horizontal and vertical routing lanes on the FPGA.



FPGA I/O Blocks



Summary

- There are currently two types of programmable logic devices in common use:
  - CPLDs
    - in-circuit programmable
    - non-volatile (retains configuration information when powered down)
    - "instant on" (no external configuration ROM or boot sequence required)
    - less dense (fewer programmable logic blocks) than comparably sized FPGA
  - FPGAs
    - in-circuit programmable
    - volatile (loses configuration when powered down)
    - requires external configuration ROM and "boot" sequence to initialize
    - more dense (greater number programmable logic blocks) than comparably sized CPLD