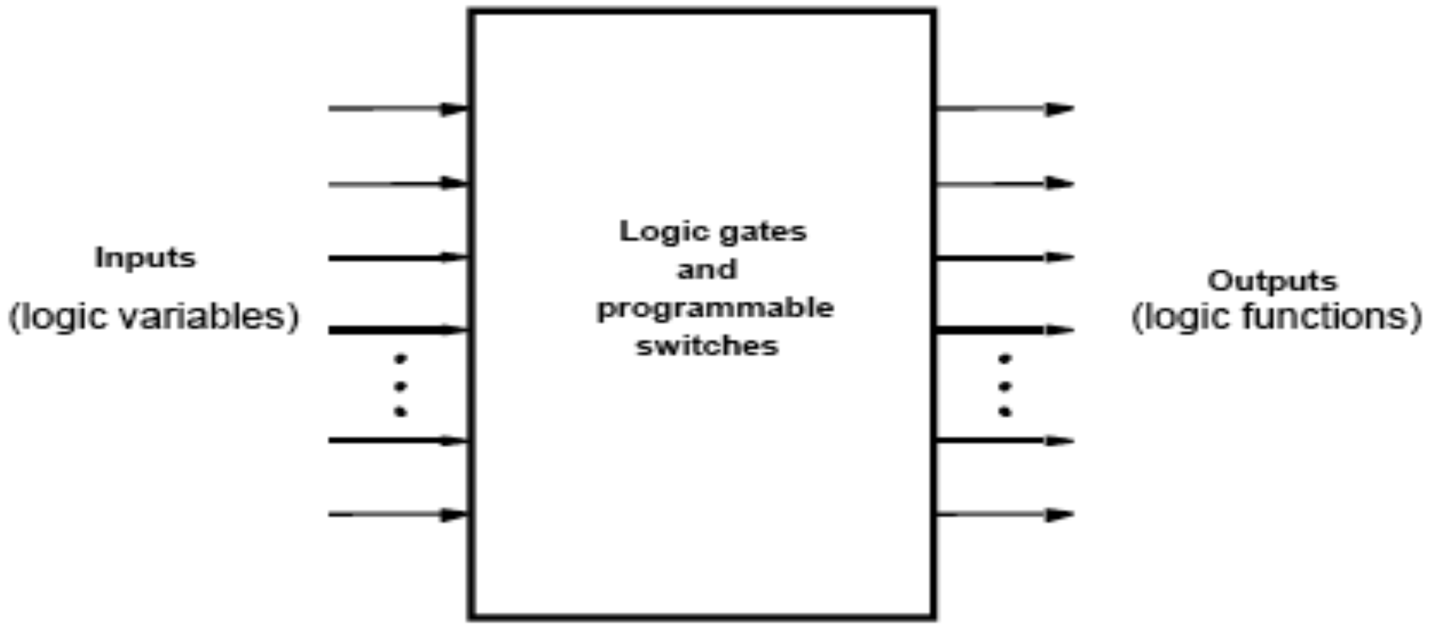
ECE327

PROGRAMMABLE LOGIC DEVICES

Programmable Logic Device Black Box



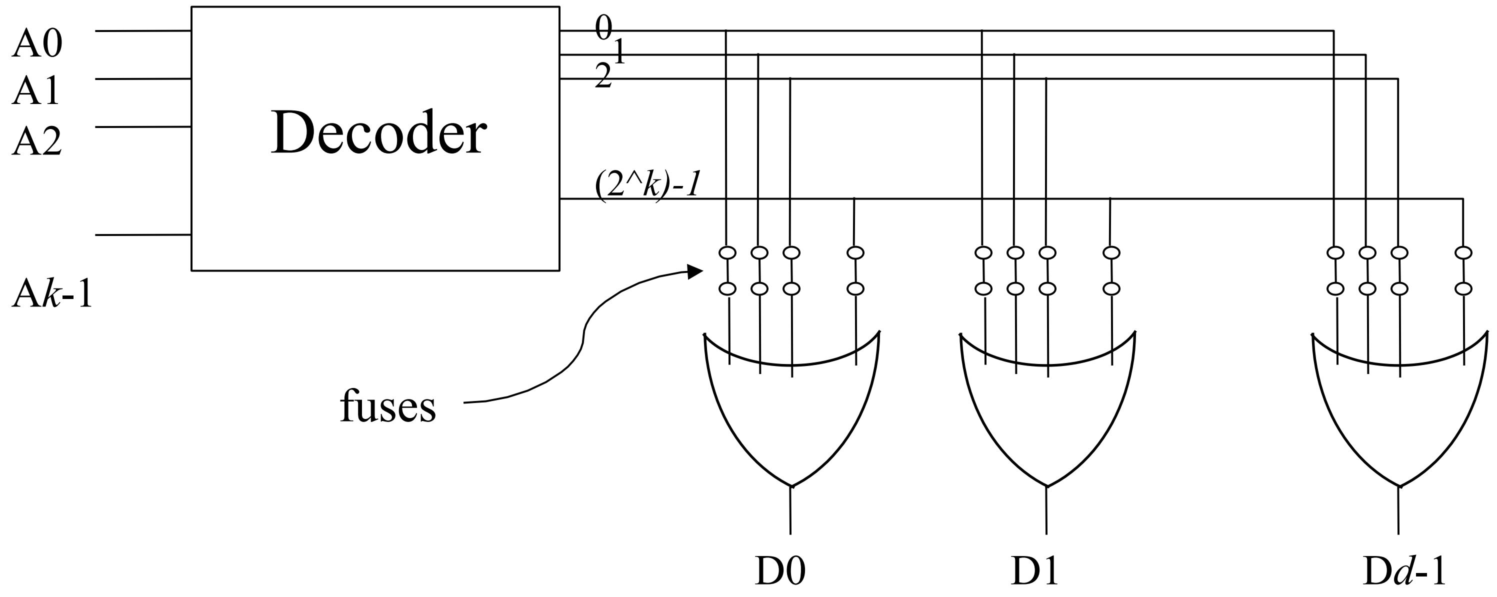
Programmable Logic Devices

- Read Only Memory (ROM)
  - mask programmable
  - PROM, EPROM, EEPROM
- Programmable Logic Array (PLA)
- Programmable Array Logic (PAL)
- Field Programmable Gate Array (FPGA)
  - Mux based
  - LUT based

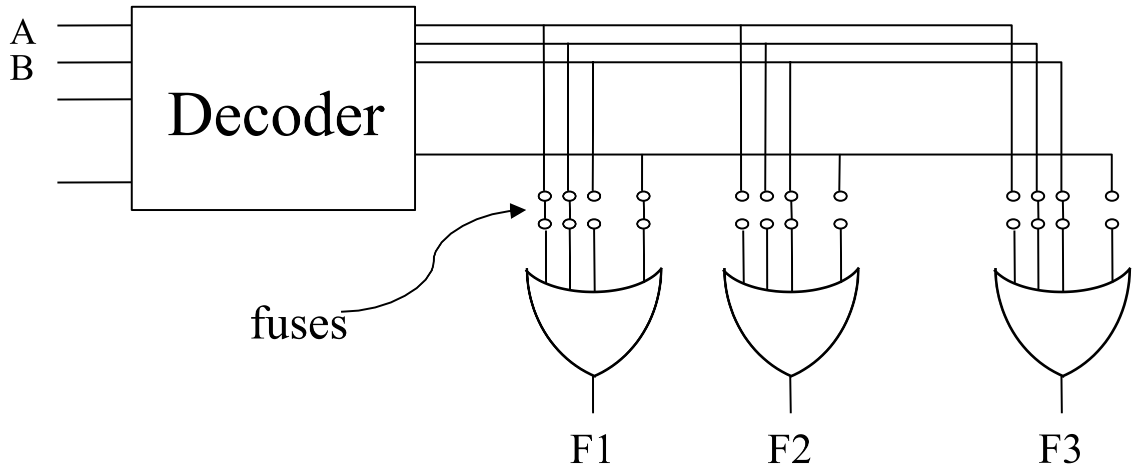
ROMs

- Input: k Address lines (A)
- Output: d Data lines (D)
- Programmable OR array
- Function:
  - Each possible value of A [0..(2^k)-1] has a unique set of *d* bits that are output on D when the corresponding "address" is provided on A

ROMs



Implementing Logic in ROMs

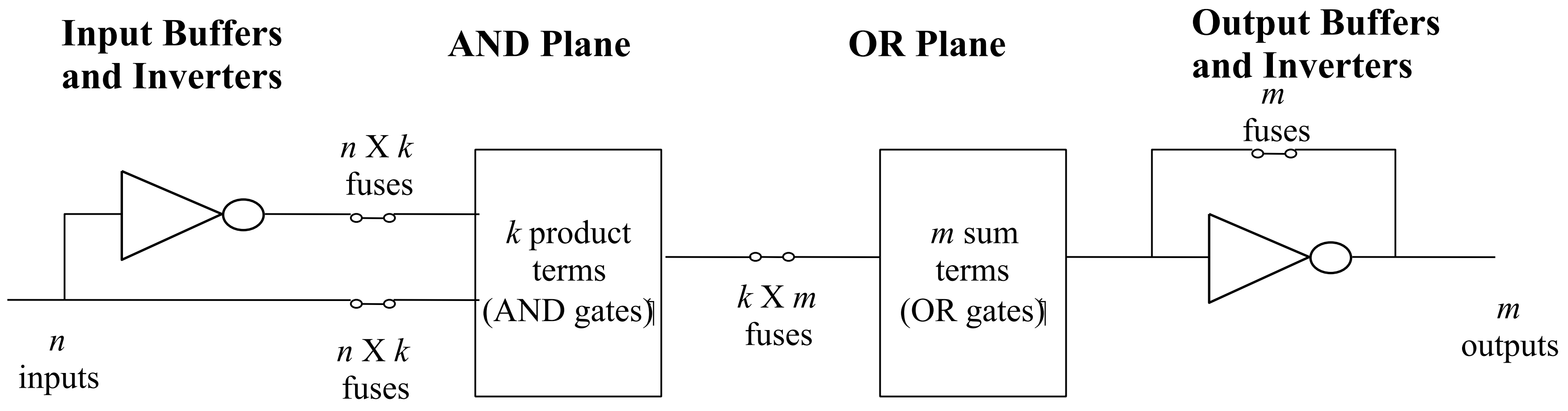


A	B	F1	F2	F3
0	0	1	0	0
0	1	1	0	1
1	0	0	1	1
1	1	1	0	0

ROM Types

- Mask programmable ROM
  - fuses programmed during manufacture
- Programmable ROM (PROM)
  - 0's programmed by blowing fuses or "burning"
- Erasable PROM (EPROM)
  - programming erased by UV light
- Electrically erasable PROM (EEPROM)
  - programming erased via control signals

Programmable Logic Array (PLA)



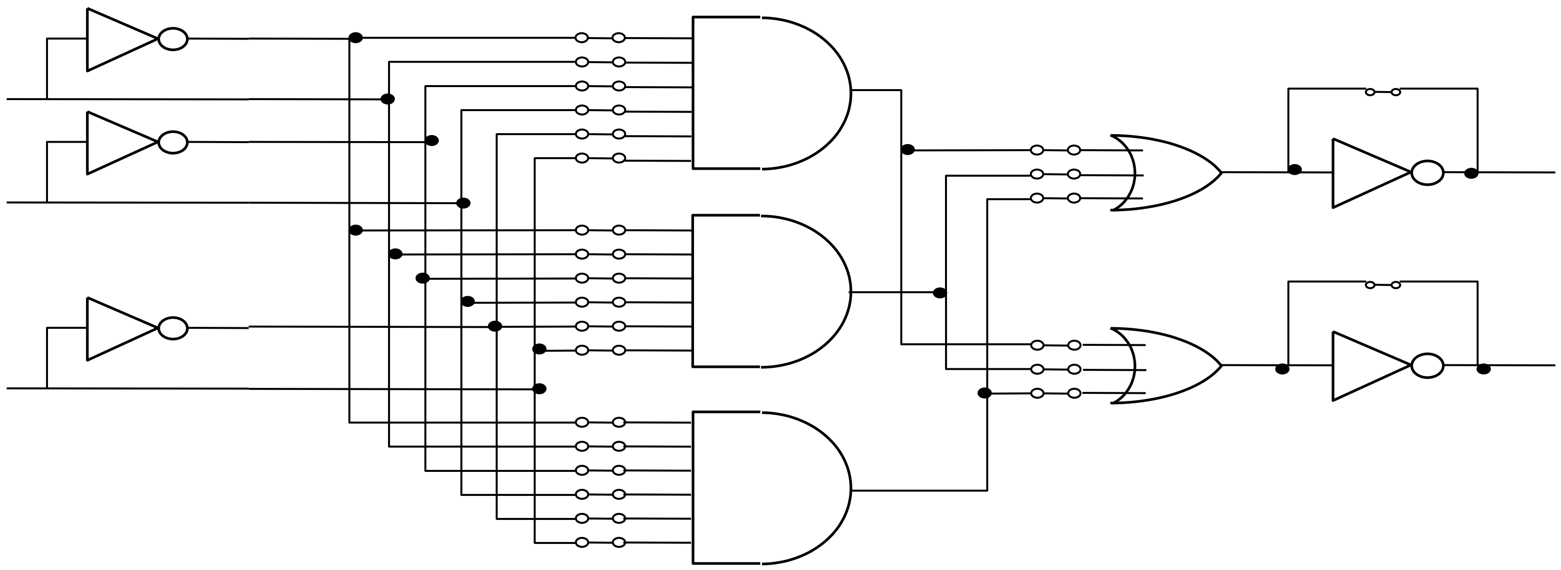
Programmable Logic Array (PLA)

Input Buffers and Inverters

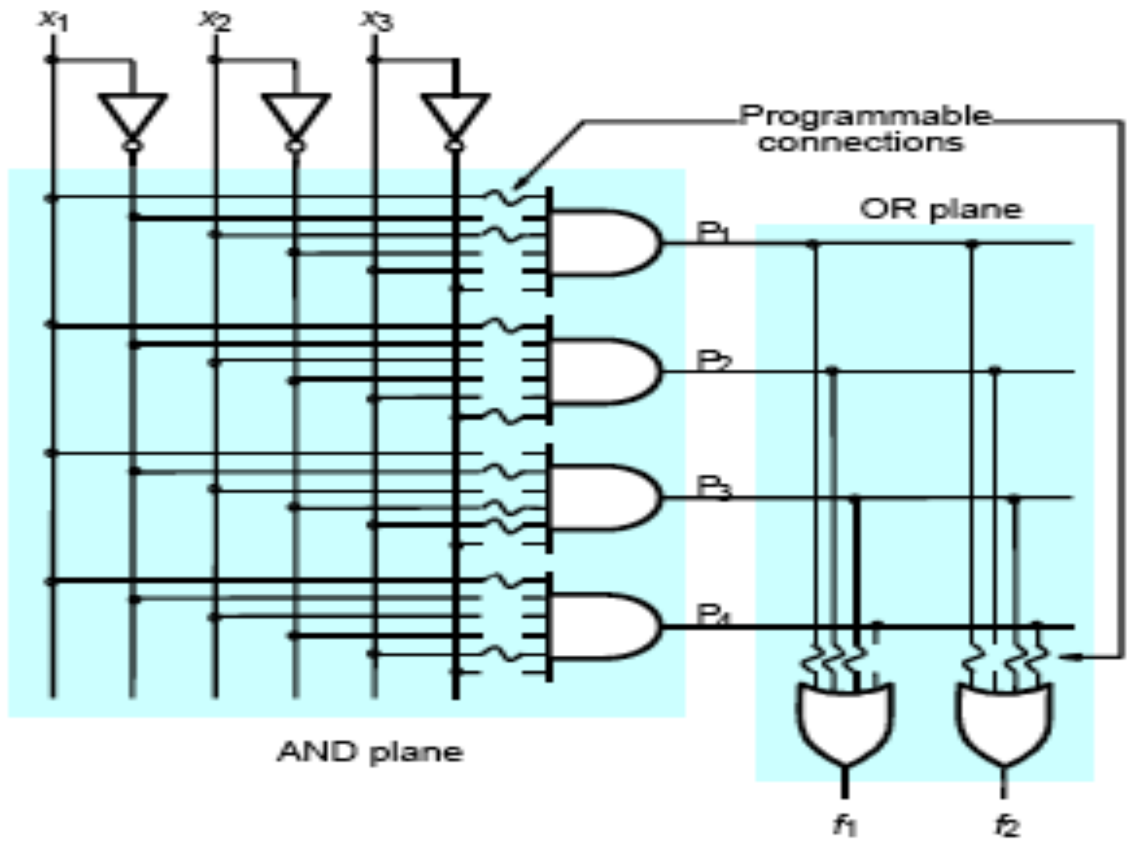
AND Plane

OR Plane

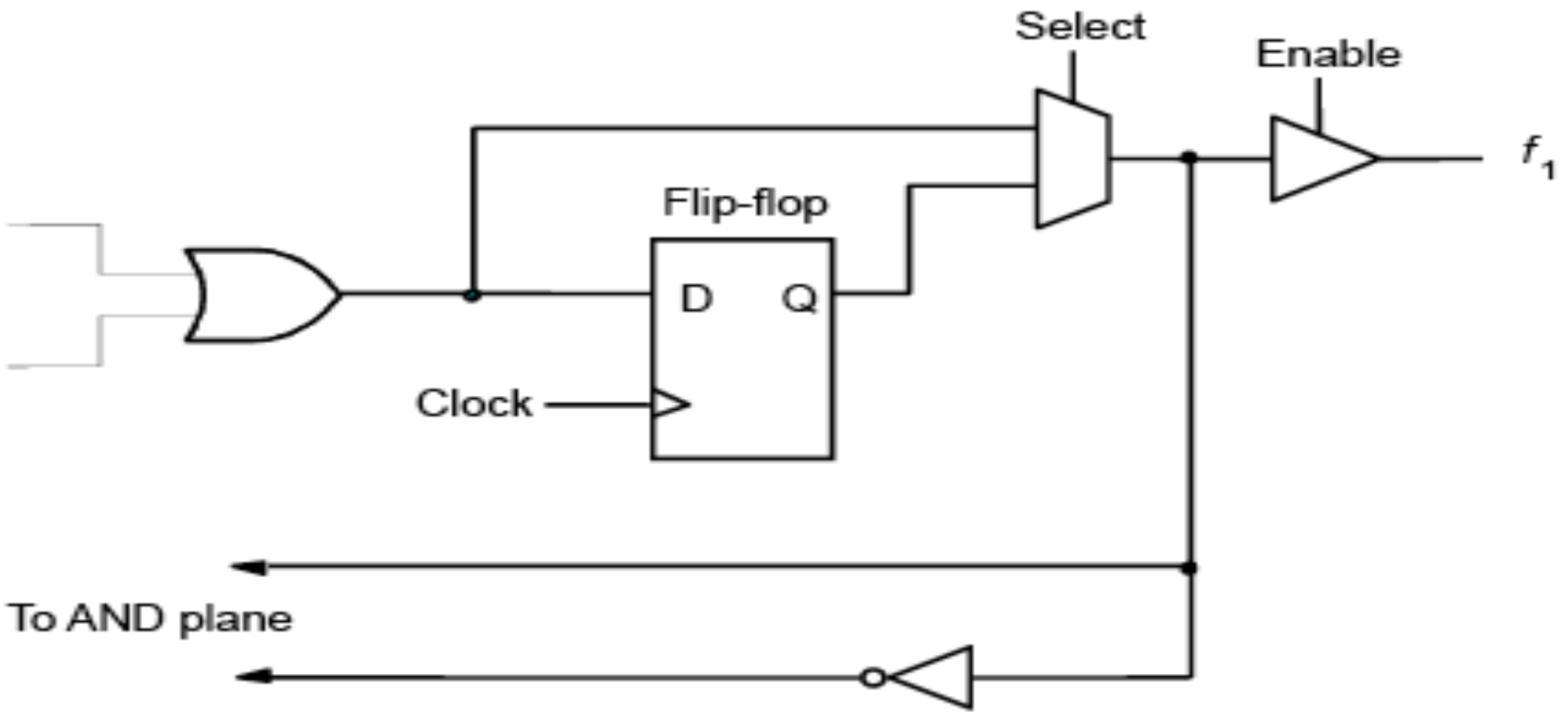
Output Buffers and Inverters



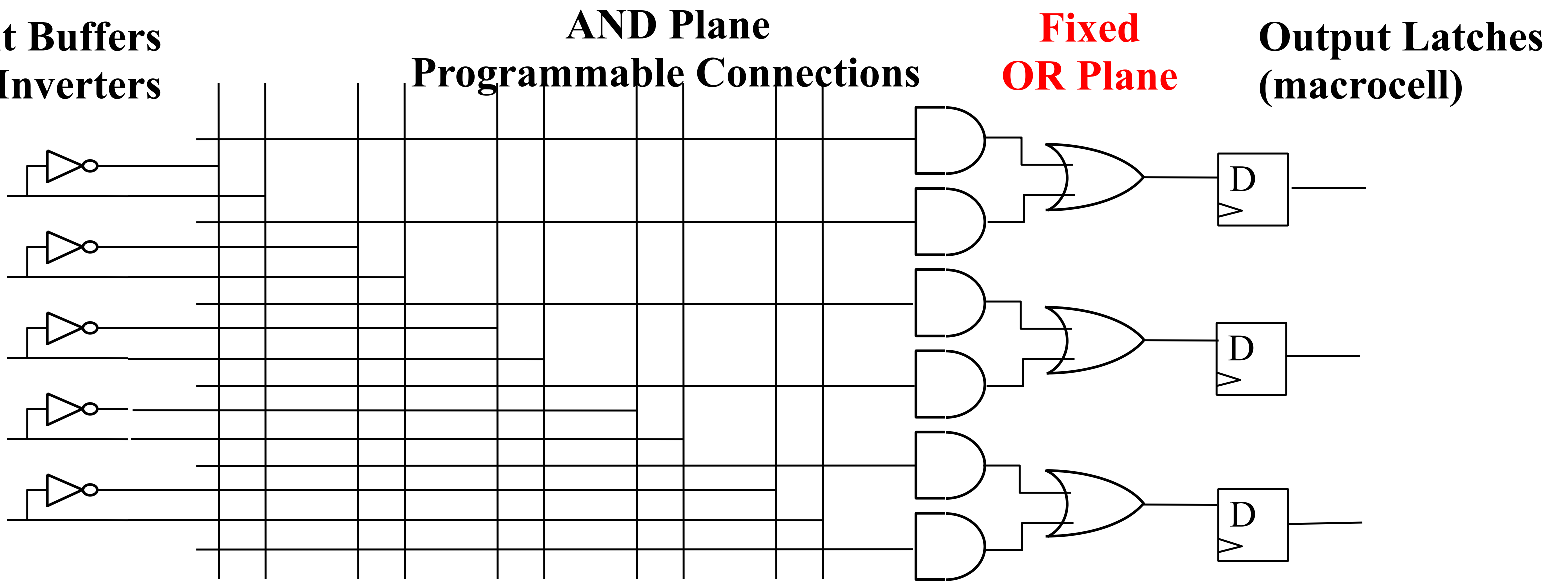
Gate Level PLA Structure



Typical PLA Output Circuitry



Programmable Array Logic (PAL)

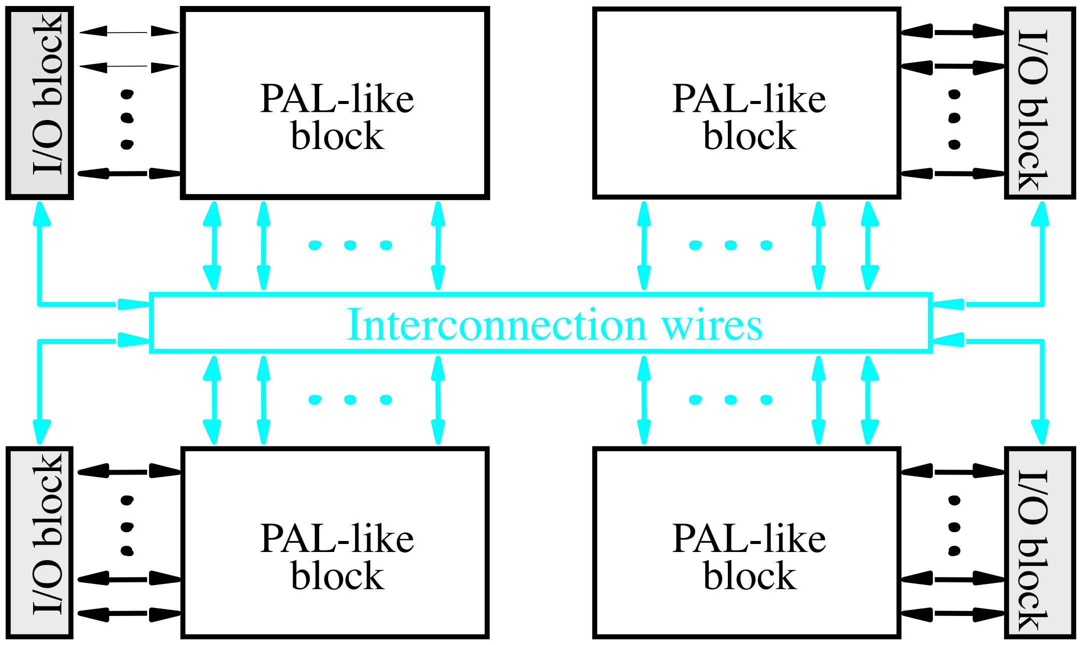

Input Buffers and Inverters
AND Plane Programmable Connections
Fixed OR Plane
Output Latches (macrocell)
D
D
D

Summary of PLD Devices

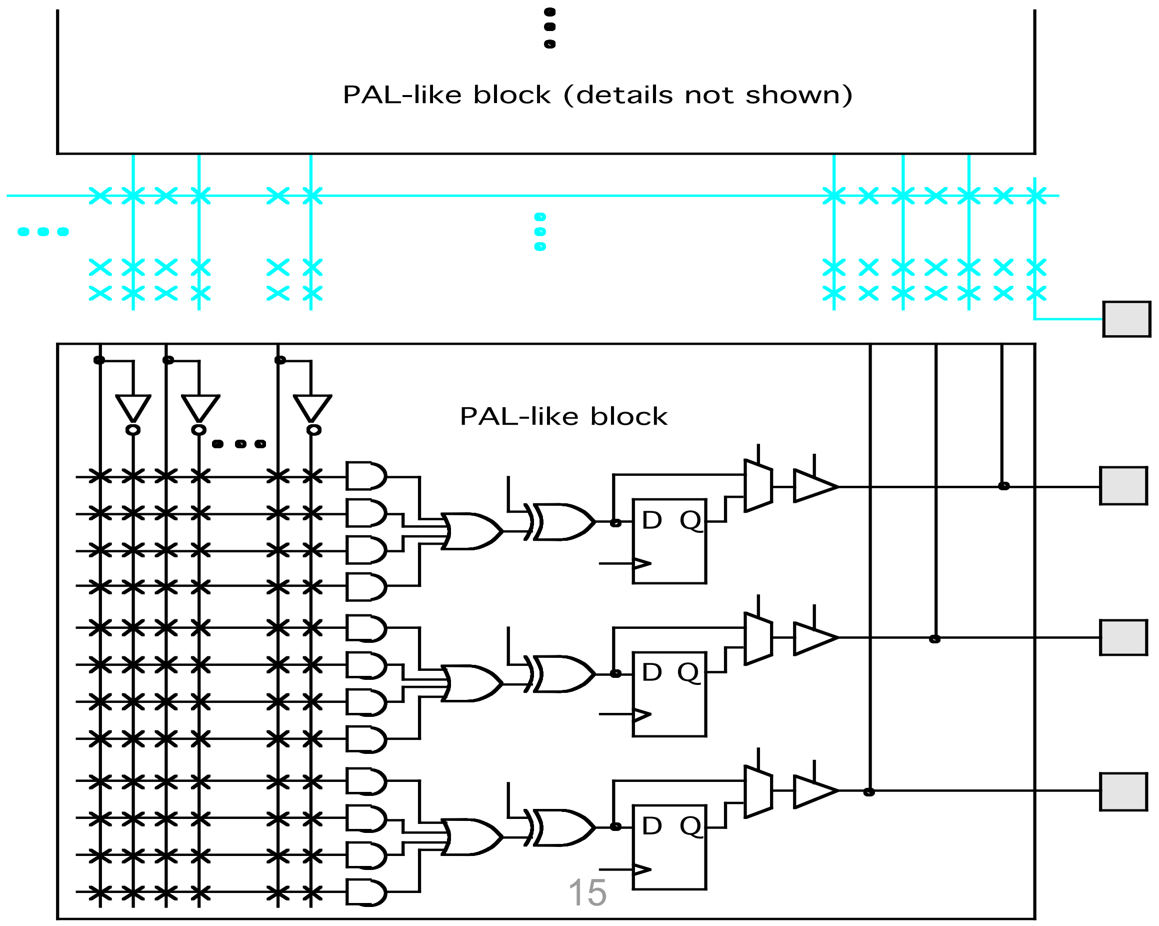
- ROM: programmable OR array
- PLA: programmable AND ***<u>***and***</u>*** programmable OR arrays
- PAL: programmable AND array
- Each type may have a macrocell that can invert the function, or provide a register at the output

Complex Programmable Logic Devices (CPLD)

- Multiple (PLA or PAL-like) logic blocks per chip
- Useful for implementing more complex circuits



Section of a CPLD



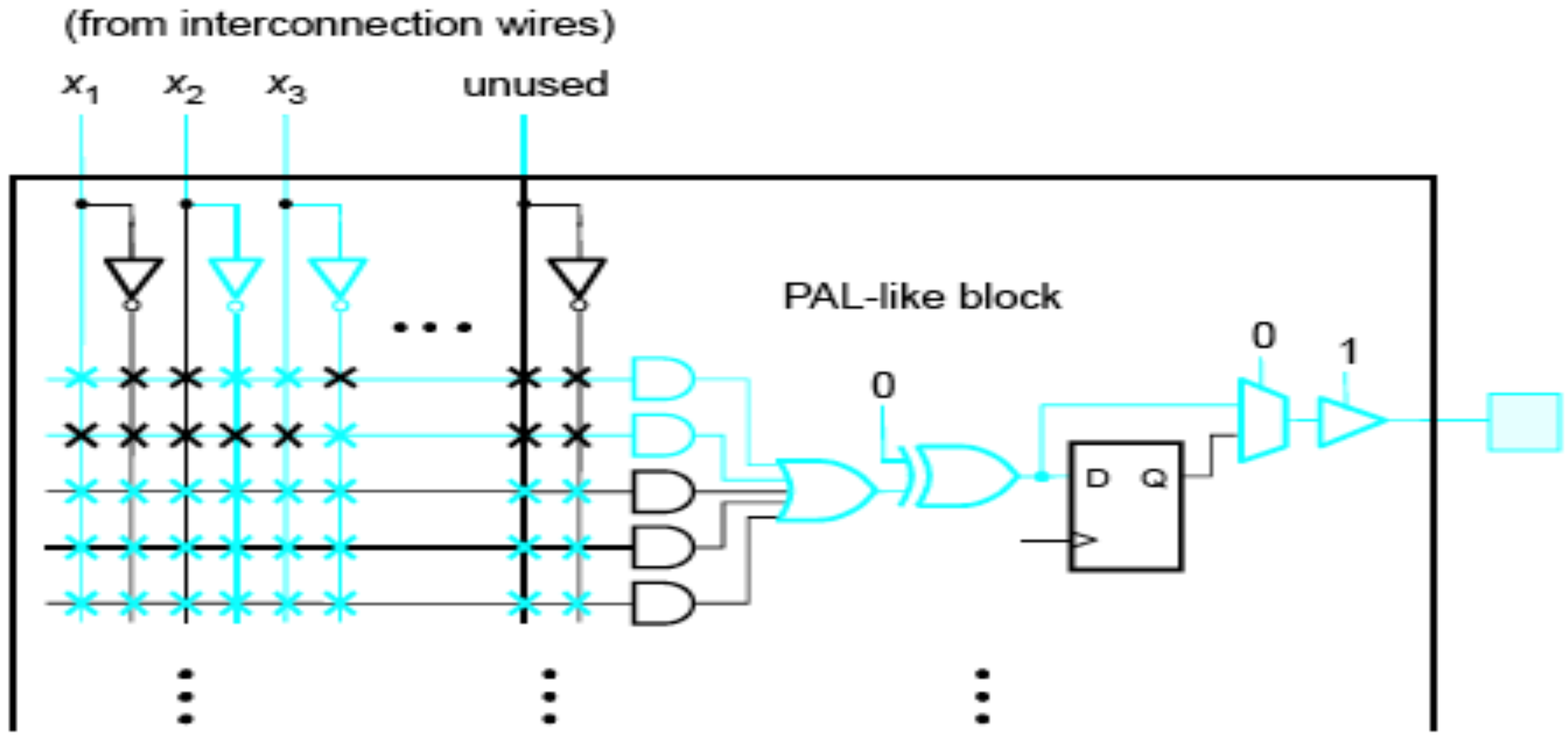
VHDL code for the function $f = \Sigma m(0,2,4,5,6)$

```
LIBRARY ieee;
USE ieee_std_logic_1164.all;

ENTITY func3 IS
   PORT(x1,x2,x3   : IN   STD_LOGIC;
        f       : OUT STD_LOGIC);
END func3;

ARCHITECTURE LogicFunc OF func3 IS
BEGIN
   f <= (NOT x1 AND NOT x2 AND NOT x3) OR
        (NOT x1 AND x2 AND NOT x3) OR
        (x1 AND NOT x2 AND NOT x3) OR
        (x1 AND NOT x2 AND x3) OR
        (x1 AND x2 AND NOT x3);
END LogicFunc;
```

VHDL implementation of $f = \Sigma m(0,2,4,5,6)$



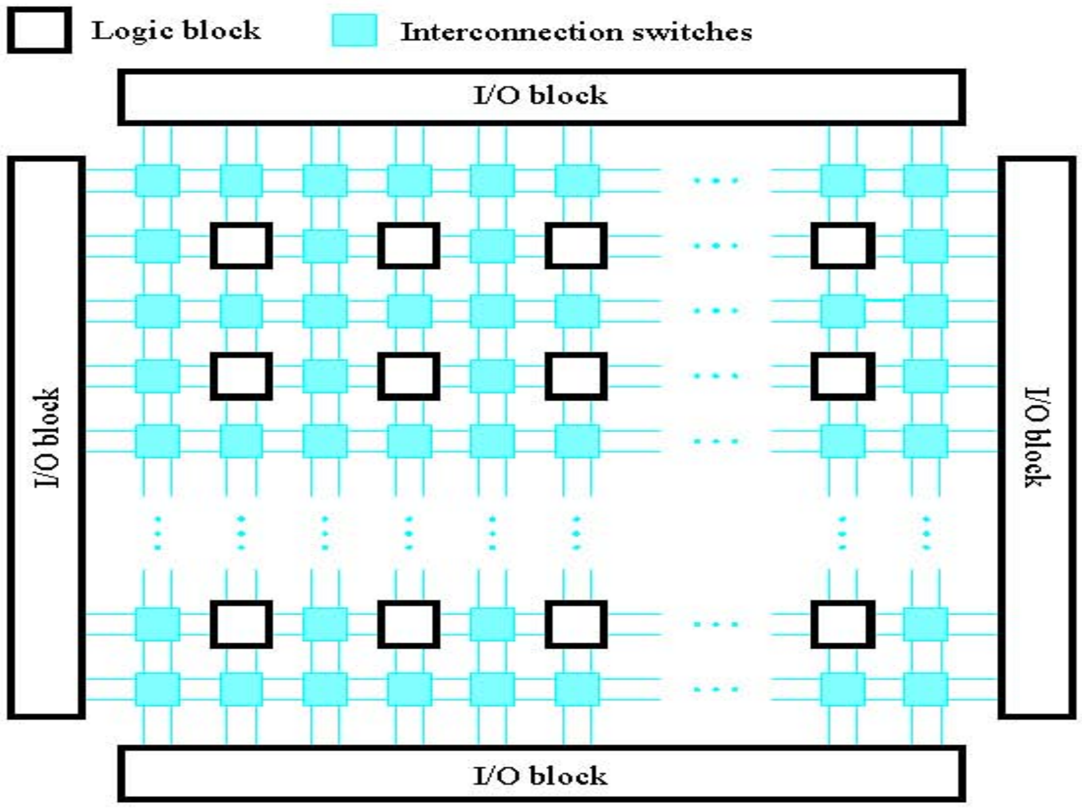
FPGAs

- The field-programmable gate array (FPGA) is completely manufactured by the IC vendor
  - The device is design independent
- Each manufacturer has their own proprietary architecture for their devices that includes
  - Programmable blocks connected to
  - Programmable switch matrices
- A device is configured to implement a particular design by programming the switch matrices to route signals between programmable logic blocks

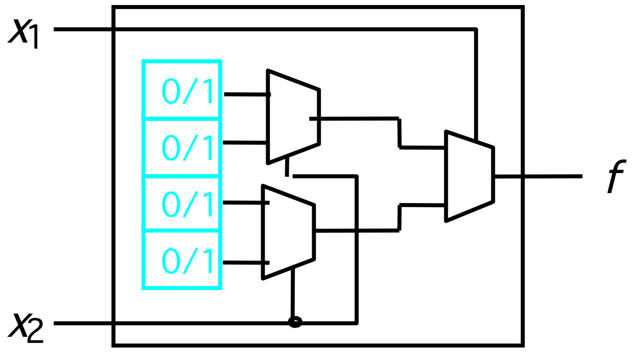
FPGAs

- Logic Implementation Types
  - Look-up Tables (LUTs)
  - Multiplexers
  - Array Logic (PAL)
- Programming Styles
  - Programmable (but not re-programmable)
  - Re-programmable (whole chip)
  - Partially Re-programmable (partial chip)

Structure of an FPGA



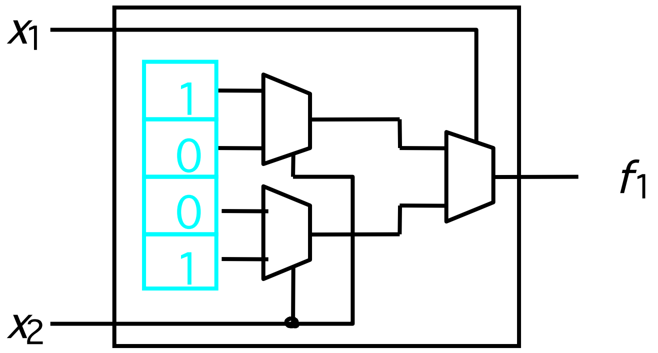
A 2-input Lookup Table



(a) Circuit for a two-input LUT

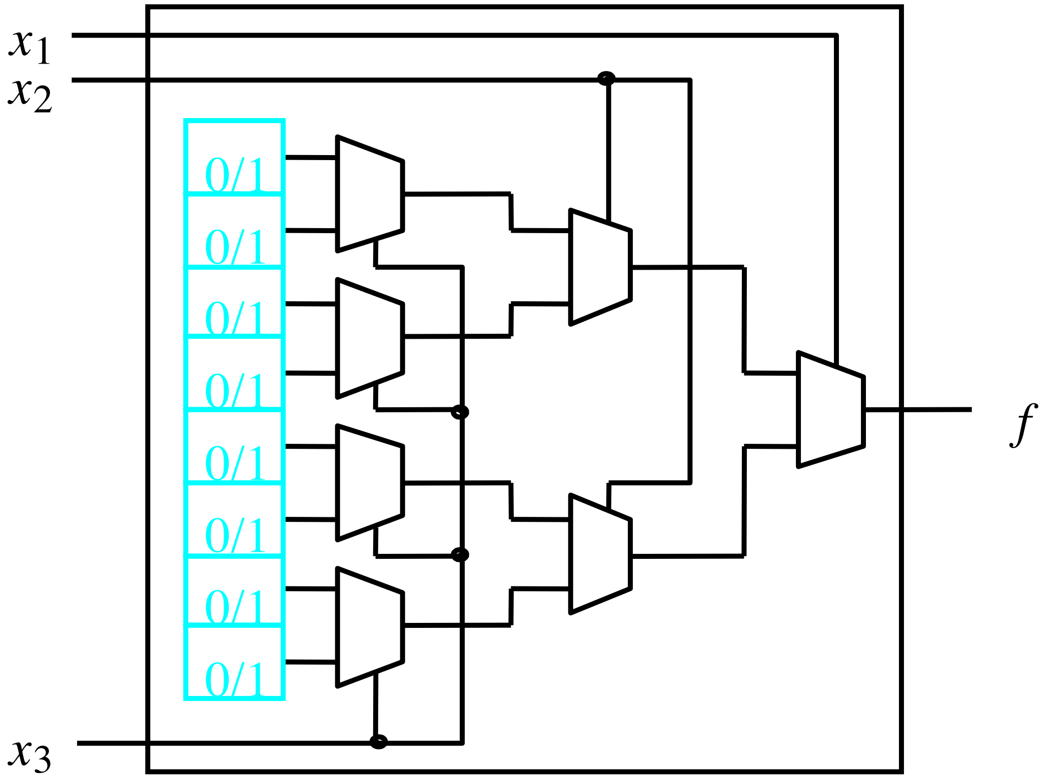
x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \overline{x_1}\,\overline{x_2} + x_1 x_2$

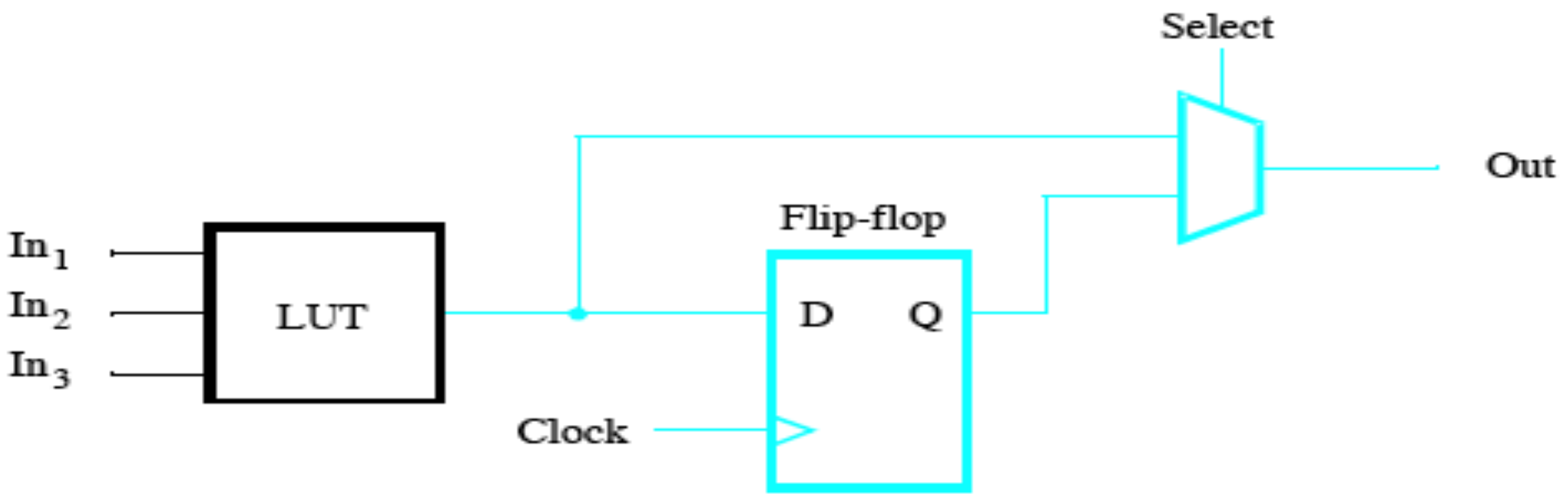


(c) Storage cell contents in the LUT

A 3-input Lookup Table

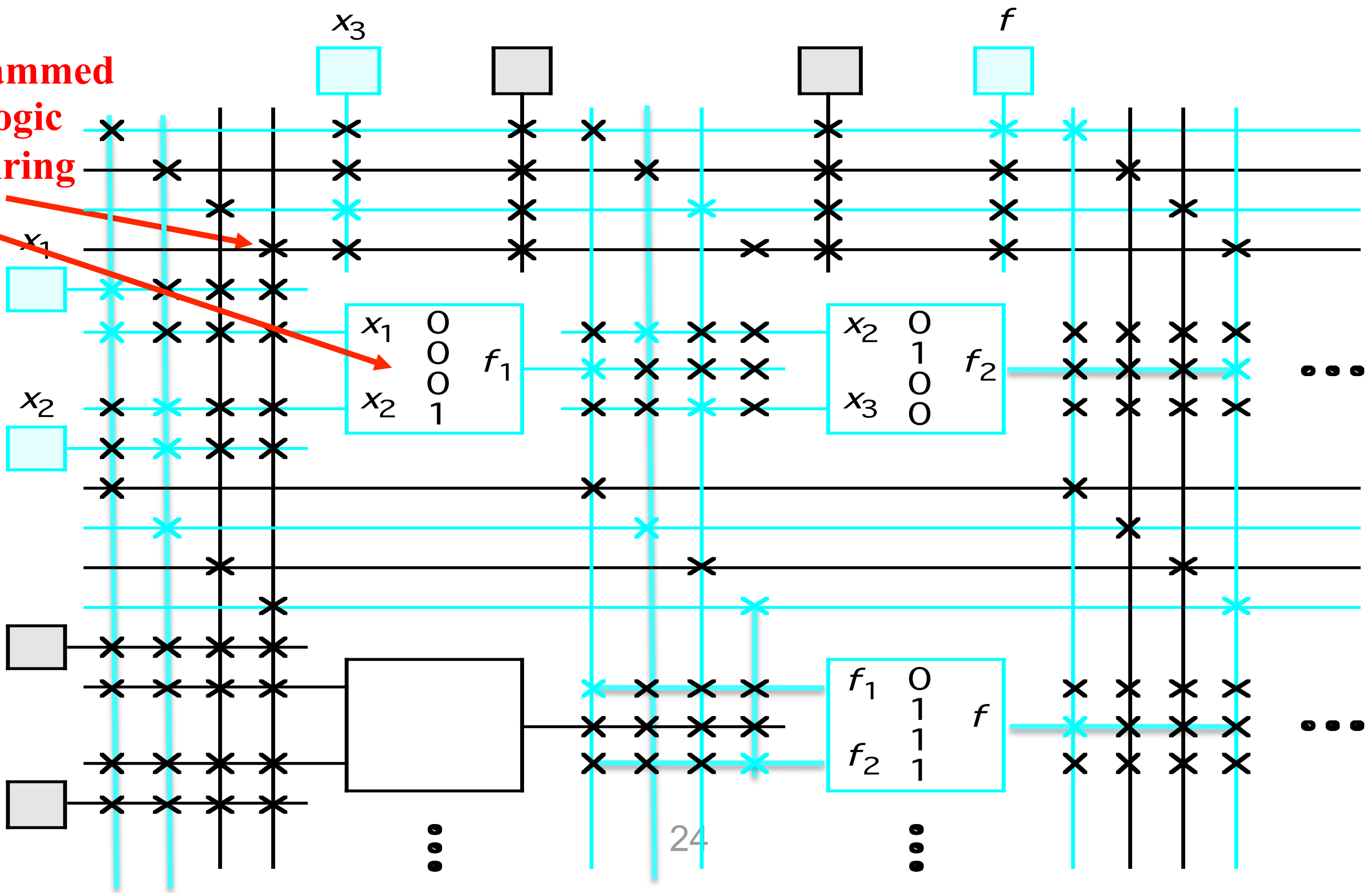


Inclusion of a flip-flop with a LUT

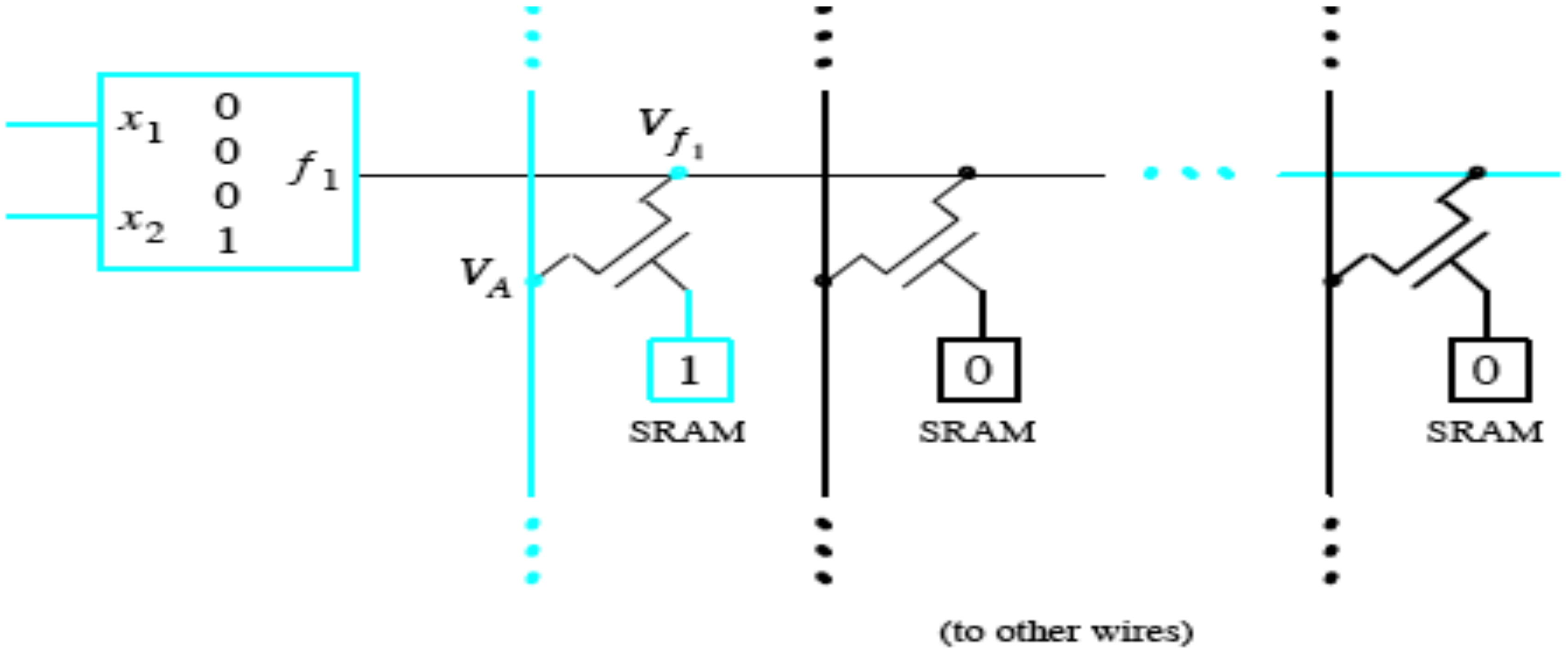


Section of a Programmed FPGA

Shows programmed states of the logic blocks and wiring switches



Pass Transistor Switches in an FPGA



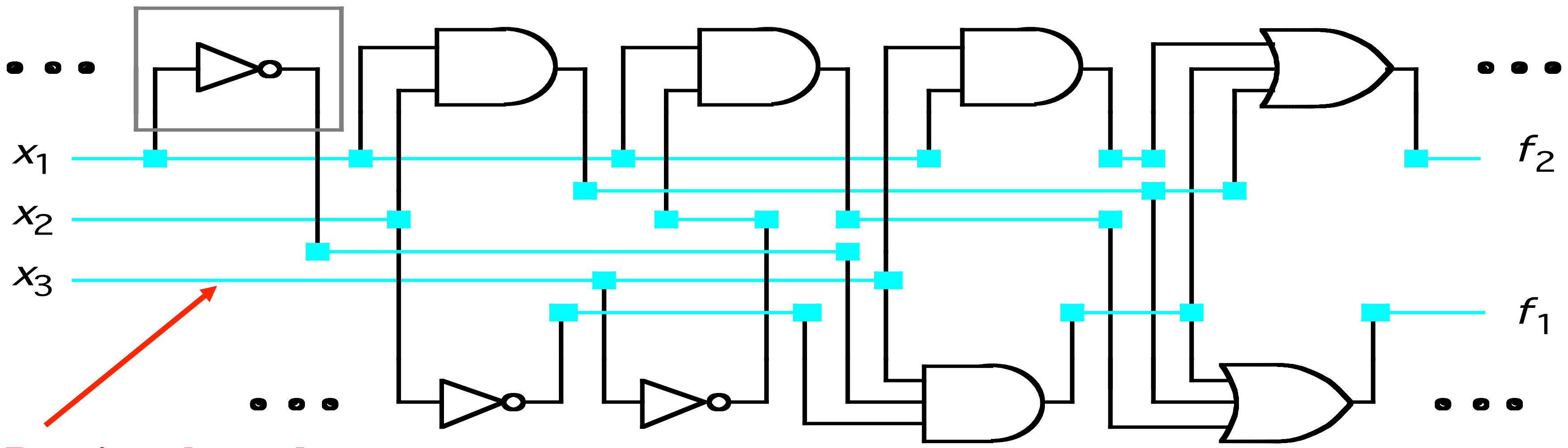
ASICs

- Application Specific Integrated Circuits
  - Standard Cells
  - Semi-Custom and Full Custom
  - Sea-of-gates

- NOT REPROGRAMMABLE!

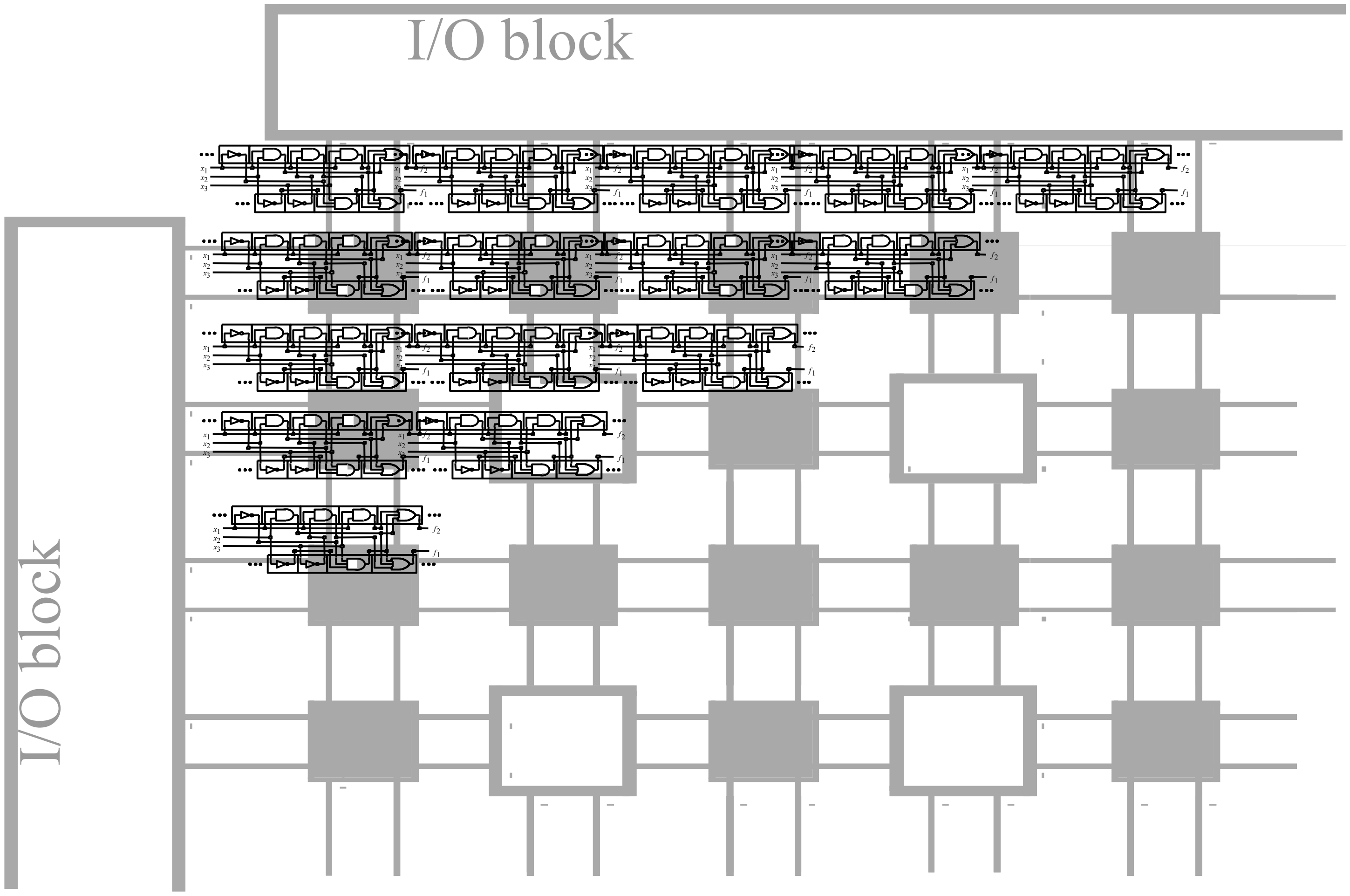
ASIC Types

- An Sea of Gates is partially manufactured by an ASIC vendor
  - Integrated circuit with an array of unconnected transistors; basic cells for implementing logic
  - Library of cells: primitive logic gates, registers, MSI and LSI complexity level cells (adders, multiplexers, etc.)
  - Final manufacturing process requires connecting transistors together to implement a specific design
- Standard Cell, Semi-Custom, and Full Custom
  - No concept of a basic cell
  - No components prefabricated on the IC
  - Silicon used more efficiently



Routing channels
using multiple metal layers

A section of two rows in a standard-cell chip



FPGA and Standard Cell Compared

FPGAs vs. PLDs

- PLD functions (the AND/OR planes) are tied to output pins, FPGA functions are routed to output pins
- FPGAs have more functions than pins (can result in I/O bounded problems)
- FPGAs typically have more FFs per equivalent gates of logic than PLDs
- FPGAs include routing resources to create arbitrary interconnect of functions

FPGAs vs. PLDs

- PLDs
  - Functions typically fit in a single logic block,
  - CAD tools only need to place logic in appropriate blocks - no routing performed
- FPGAs
  - Functions typically spread across multiple logic blocks
  - CAD tools must place AND route (harder)

FPGAs vs. ASICs

- ASICs
  - Functionality fixed by the designer for a particular application
  - Requires a manufacturing process step to layout the design
  - Faster, Denser, Cheaper (in large quantities)
  - Expensive to design (NRE, Errors difficult to correct …)
    - After the initial cost, the per unit cost can be low
- Structured ASICs
  - Some lower levels of the device are prefabricated
  - Upper levels are based on a given (user) design
  - Mix between ASIC and FPGA
- FPGAs
  - Cheaper (in small quantities, higher per unit cost than ASICs)
  - Re-programmable, “field” programmable by the designer
  - Useful for prototyping, able change or upgrade as needed (w/i logic limits)

FPGAs vs. ASICs

- If production volume is low then FPGA is probably the implementation choice
- For high volume devices, ASICs may be more cost effective
- For the same design, ASICs will generally hold a performance advantage over FPGAs
- Alternately –FPGA may be used in prototyping and ASIC used in high volume production