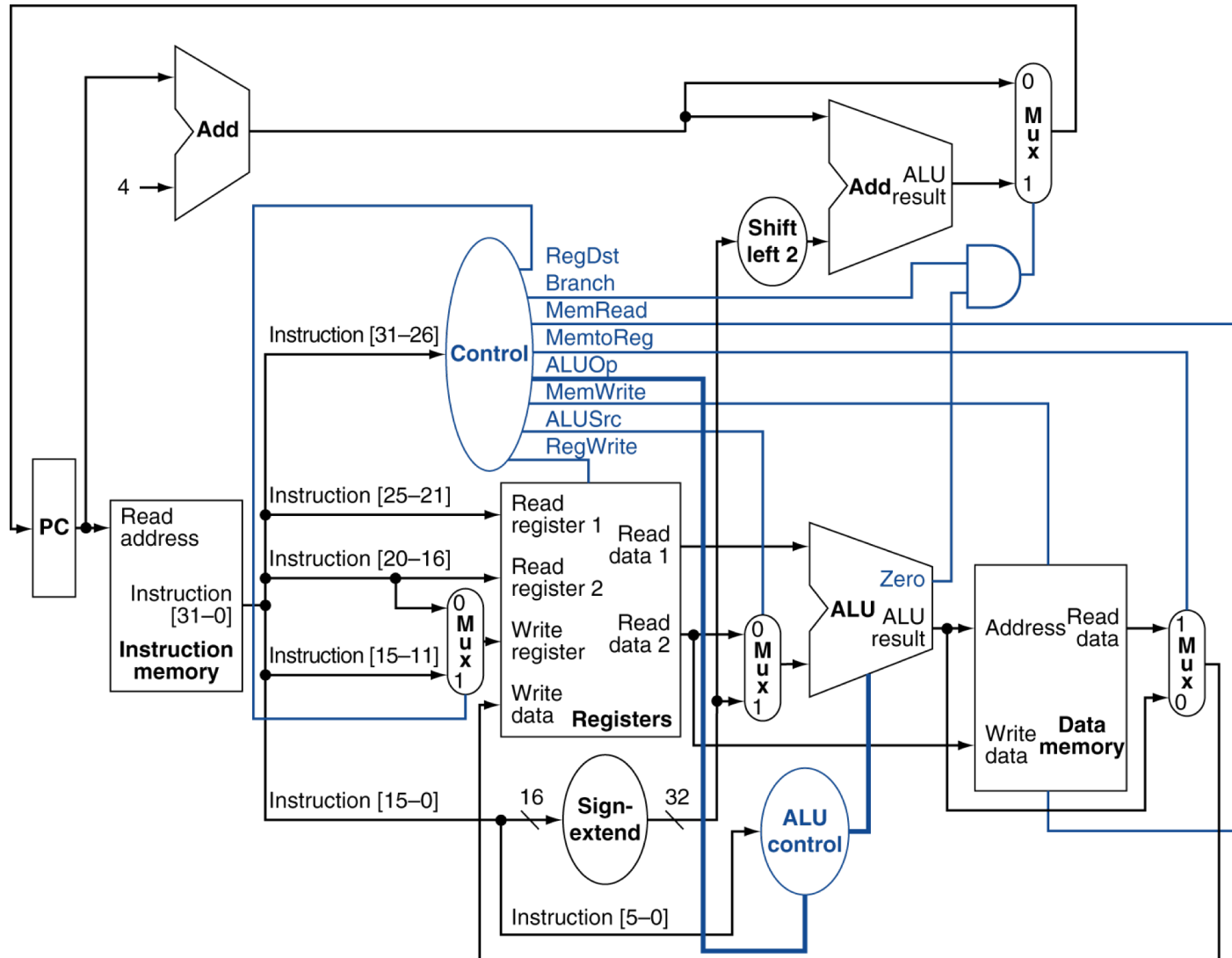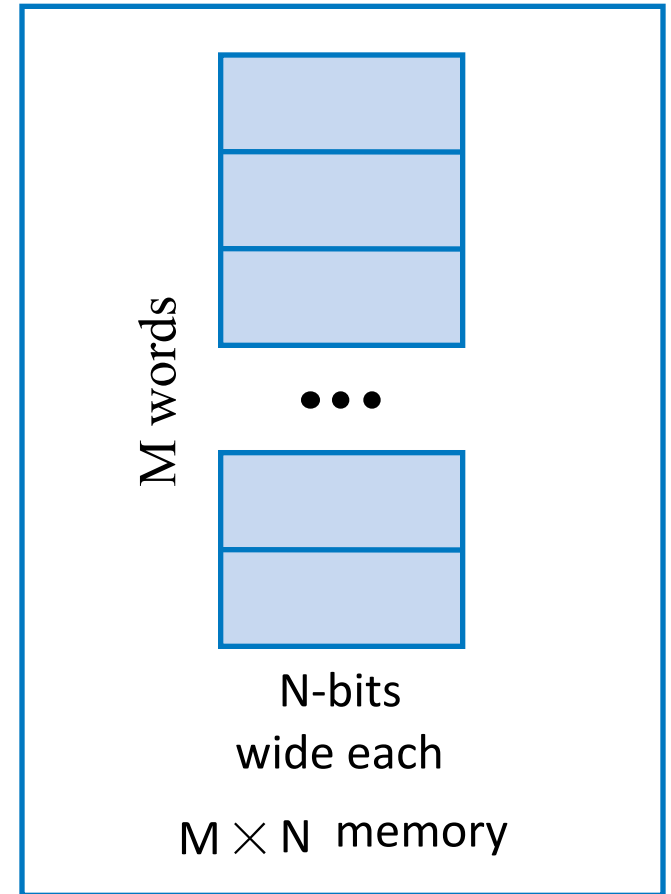# Topic 15

Memory and PLD

Big Picture – Simplified Computer

2

Memory Components
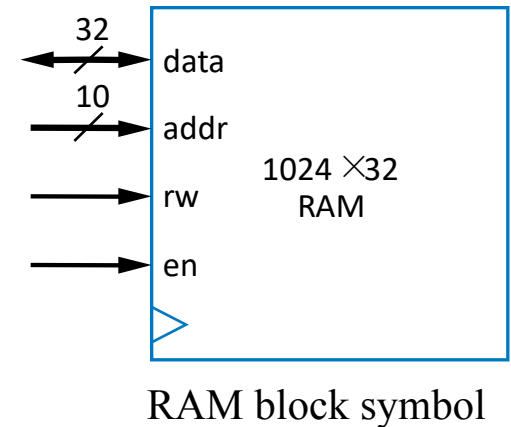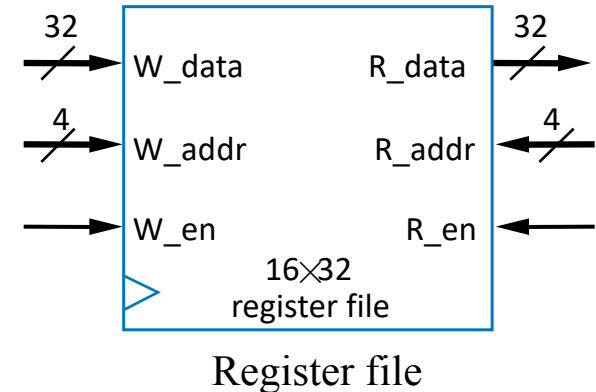
- **_MxN memory_**
  - M words (row)
  - N bits (column) wide each
- Types of memory
  - RAM
  - ROM

M words

N-bits
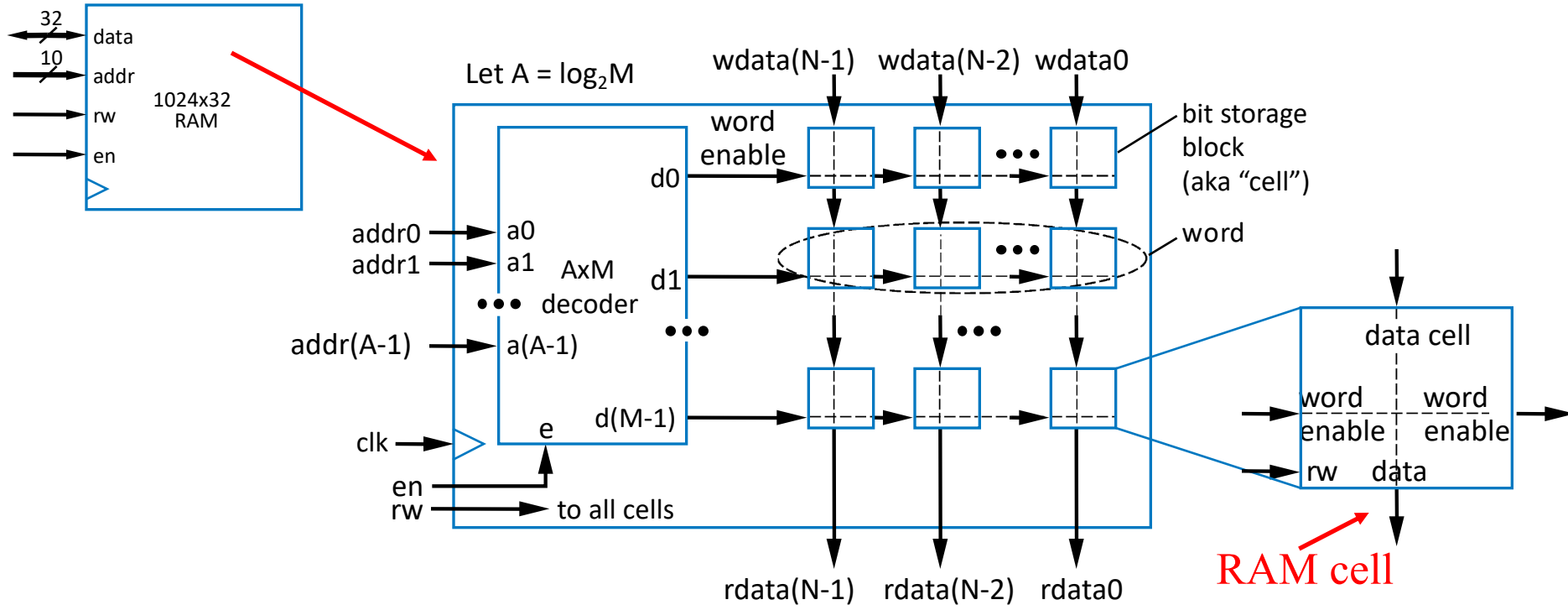wide each

M $\times$ N memory

Random Access Memory (RAM)

- RAM – readable and writable memory
  - Logically same as register file
    - Memory with address inputs, data inputs/outputs, and control
  - RAM vs. register file
    - RAM is typically larger
    - RAM typically stores bits more efficiently than flip flops
    - RAM typically implemented on a chip in a square rather than rectangular shape – keeps longest wires (hence delay) short

32 bit is a word
16 bit Half Word
8 bit byte

32   W_data   R_data   32

4   W_addr   R_addr   4

W_en   R_en

16×32
register file

Register file

32   data

10   addr

rw

en

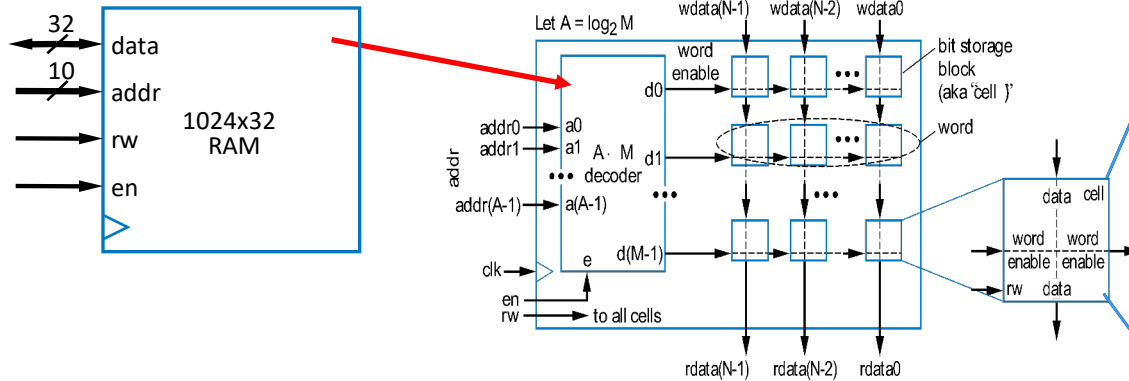1024×32
RAM

RAM block symbol
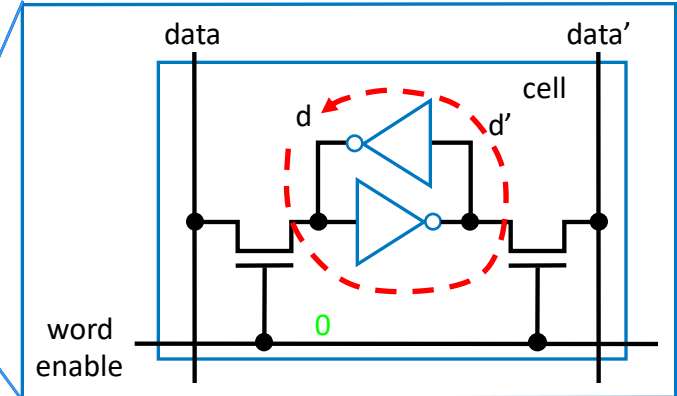
RAM Internal Structure

- Similar internal structure as register file
  - Decoder enables appropriate word based on address inputs
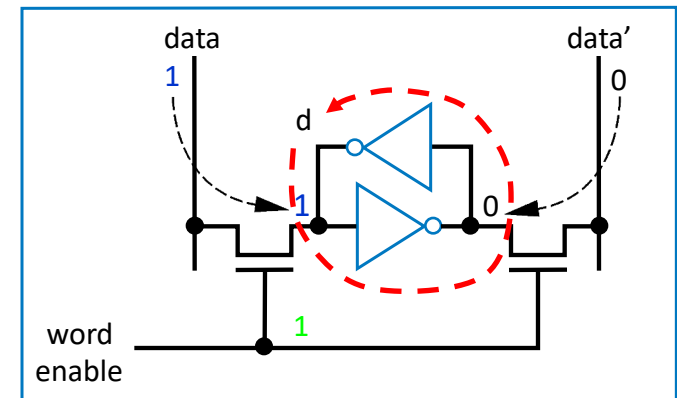 - rw controls whether cell is written or read

5

Static RAM (SRAM)

SRAM cell

SRAM cell

- **"Static" RAM cell**
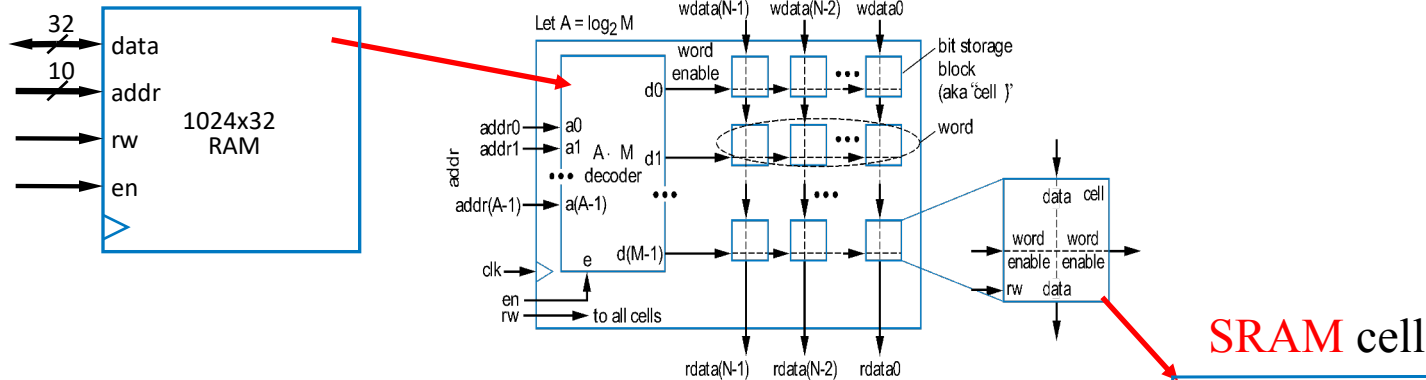 - 6 transistors (recall inverter is 2 transistors)
  - Writing this cell
    - *word enable* input comes from decoder
    - When 0, value *d* loops around inverters
      - That loop is where a bit stays stored
    - When 1, the *data* bit value enters the loop
      - *data* is the bit to be stored in this cell
      - *data'* enters on other side
      - Example shows a "1" being written into cell

6

Static RAM (SRAM)

32 data
10 addr

1024x32
RAM

rw

en

Let $A = \log_2 M$

wdata(N-1) wdata(N-2) wdata0

word enable

d0

bit storage block (aka 'cell')

addr0 → a0
addr1 → a1   word
$A \cdot M$ decoder   d1

addr(A-1) → a(A-1)

clk

e   d(M-1)

en
rw → to all cells

rdata(N-1) rdata(N-2) rdata0

data cell
word enable | word enable
rw data

SRAM cell

data **1**   data' **1**

d

1   **0**

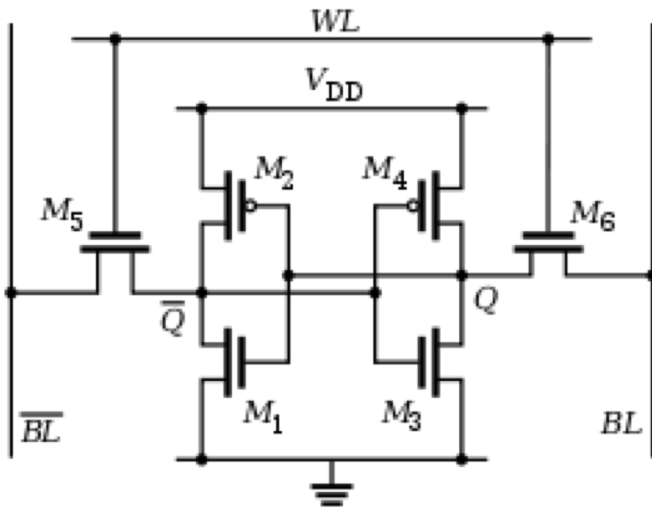**1** **1**   **<1**

word enable

To sense amplifiers

- "Static" RAM cell
  - Reading this cell
    - Somewhat trickier
    - When rw set to read, the RAM logic sets both *data* and *data'* to 1
    - The stored bit d will pull either the left line or the right line down slightly below 1
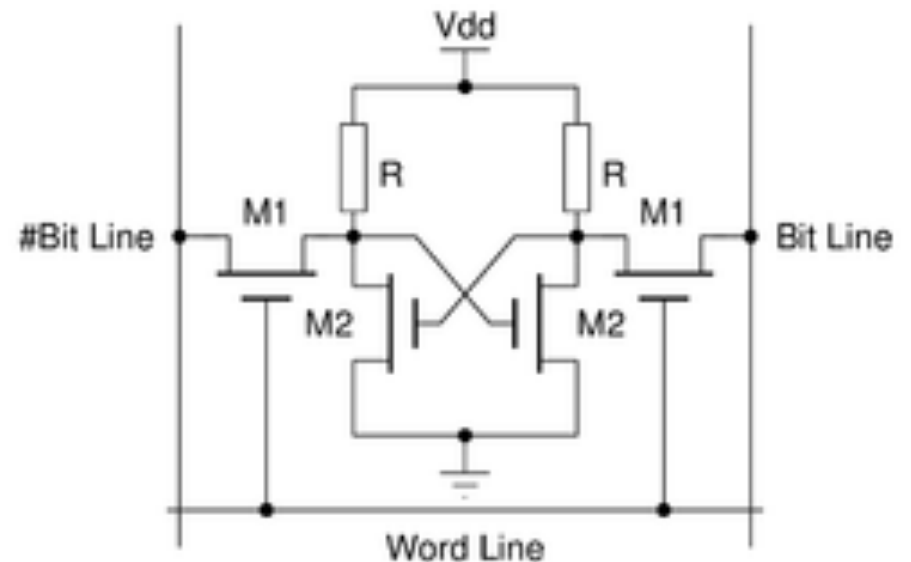   - "Sense amplifiers" detect which side is slightly pulled down
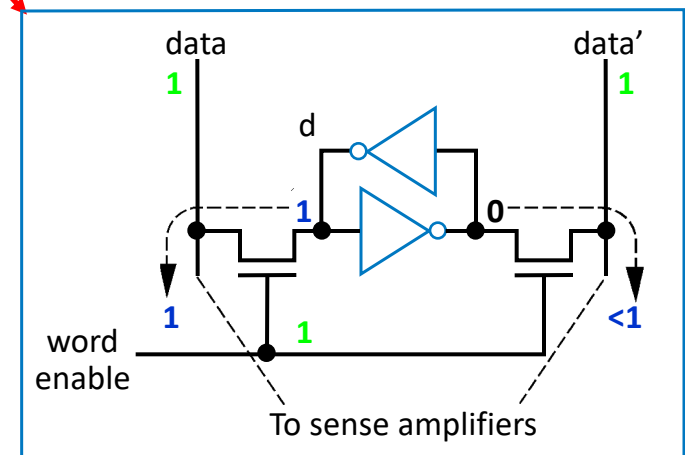
Static RAM (SRAM)
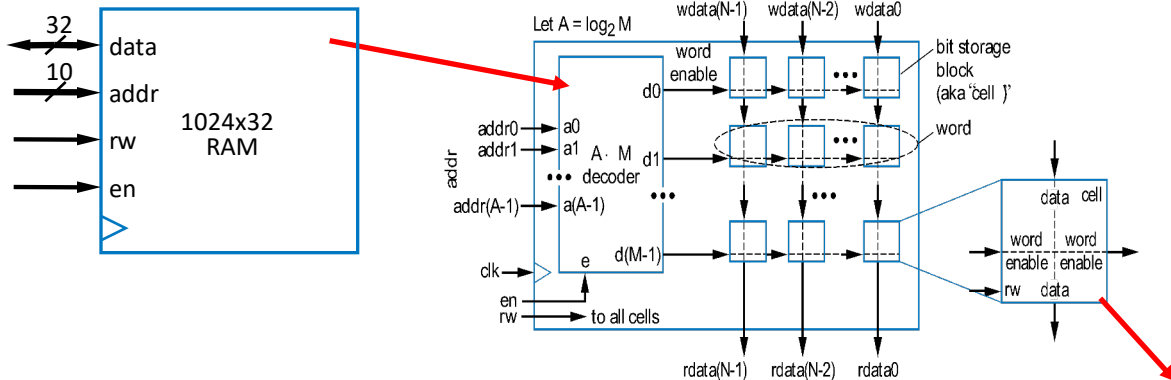
Implementation with 6 transistors


Implementation with 4 transistors

Source: wikipedia

Dynamic RAM (DRAM)

- ## "Dynamic" RAM cell
  - 1 transistor (rather than 6)
  - Relies on large capacitor to store bit
    - Write: Transistor conducts, data voltage level gets stored on top plate of capacitor
    - Read: sense amplifier on the data line
    - Problem: Capacitor discharges over time
      - Must "refresh" regularly, by reading d and then writing it right back

DRAM cell
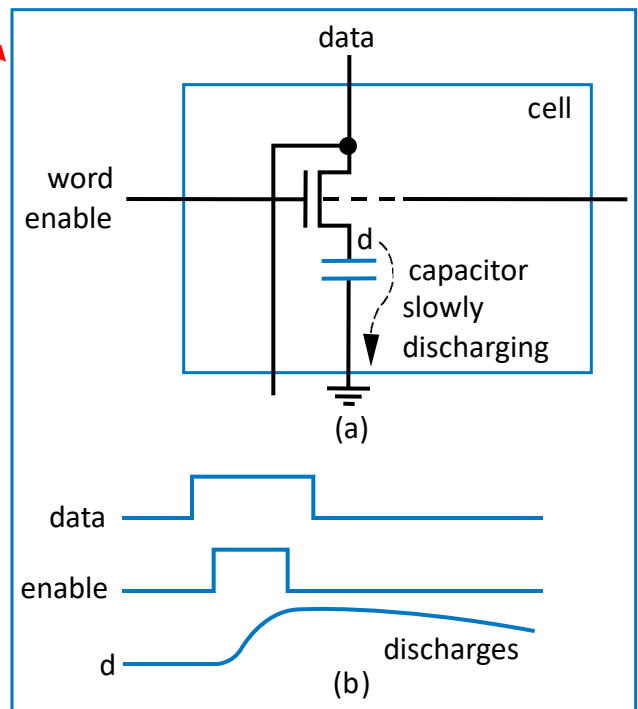
Comparing Memory Types

- Register file
  - Fastest
  - But small capacity and biggest size
- SRAM
  - Fast
  - More compact than register file
- DRAM
  - Slowest
    - And refreshing takes time
  - But very compact
- Use register file for small items, SRAM for large items, and DRAM for huge items
  - Note: DRAM's big capacitor requires a special chip design process, so DRAM is often a separate chip

Read-Only Memory – ROM

- Memory that can only be read from, not written to
  - Data lines are output only
  - No need for *rw* input
- Advantages over RAM
  - Compact: May be smaller
  - ***Nonvolatile***: Saves bits even if power supply is turned off
  - Faster Speed: especially than DRAM
  - Low power: Doesn't need power supply to save bits, so can extend battery life
- Choose ROM over RAM if stored data won't change (or won't change often)

Read-Only Memory – ROM

ROM block symbol

Let $A = \log_2 M$

word enable

bit storage block (aka "cell")

word

ROM cell
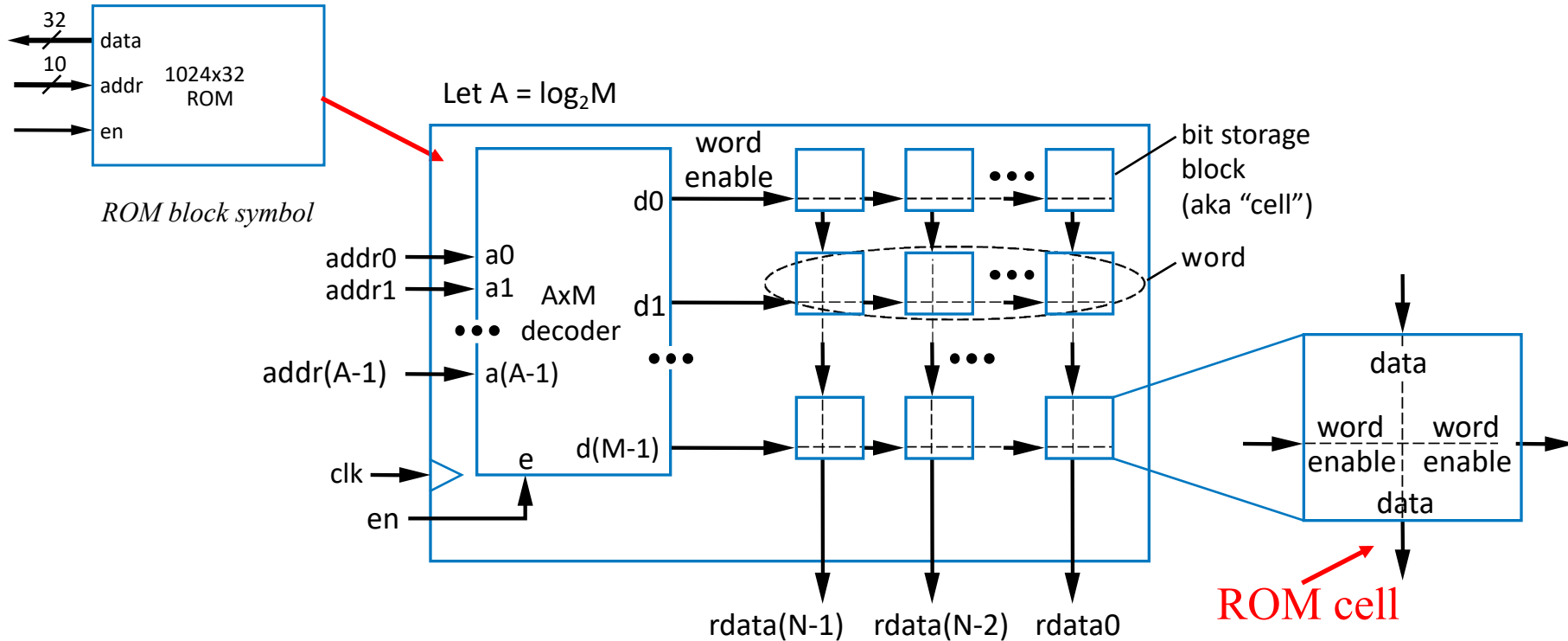
- Internal logical structure similar to RAM, without the data input lines

Read-Only Memory – ROM

- How are bits stored in ROM?
  - Storing bits in a ROM known as *programming*
  - Fuse-Based Programmable ROM (one time programming)
  - Erasable Programmable ROM (EPROM)
  - Electronically-Erasable Programmable ROM (EEPROM)
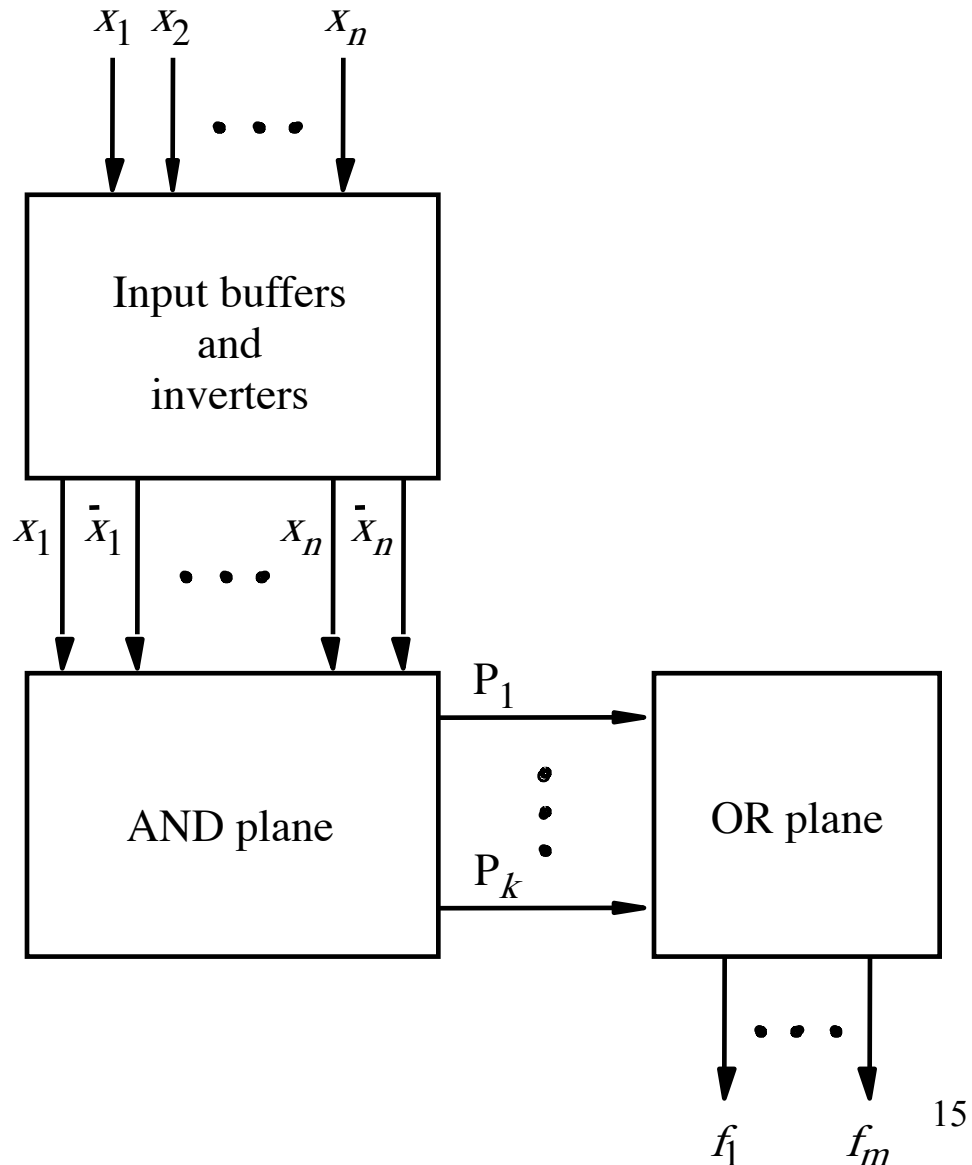   - Flash memory

13
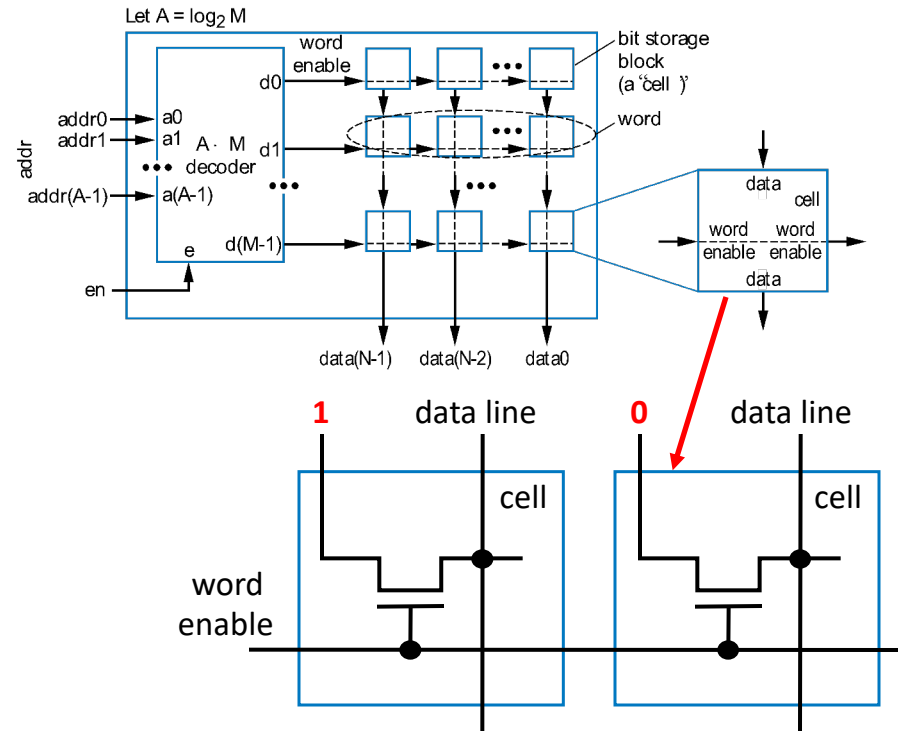
Programmable Logic Devices

- PLD
  - First introduced in 1970s
  - Can be viewed as a "black box" containing logic gates and programmable switches
    - The logic gates and programmable switches can be customized to implement specific logic circuit
  - Simple programmable logic devices (SPLD)
    - Programmable logic array (PLA)
    - Programmable array logic (PAL)
  - Complex programmable logic array (CPLD)
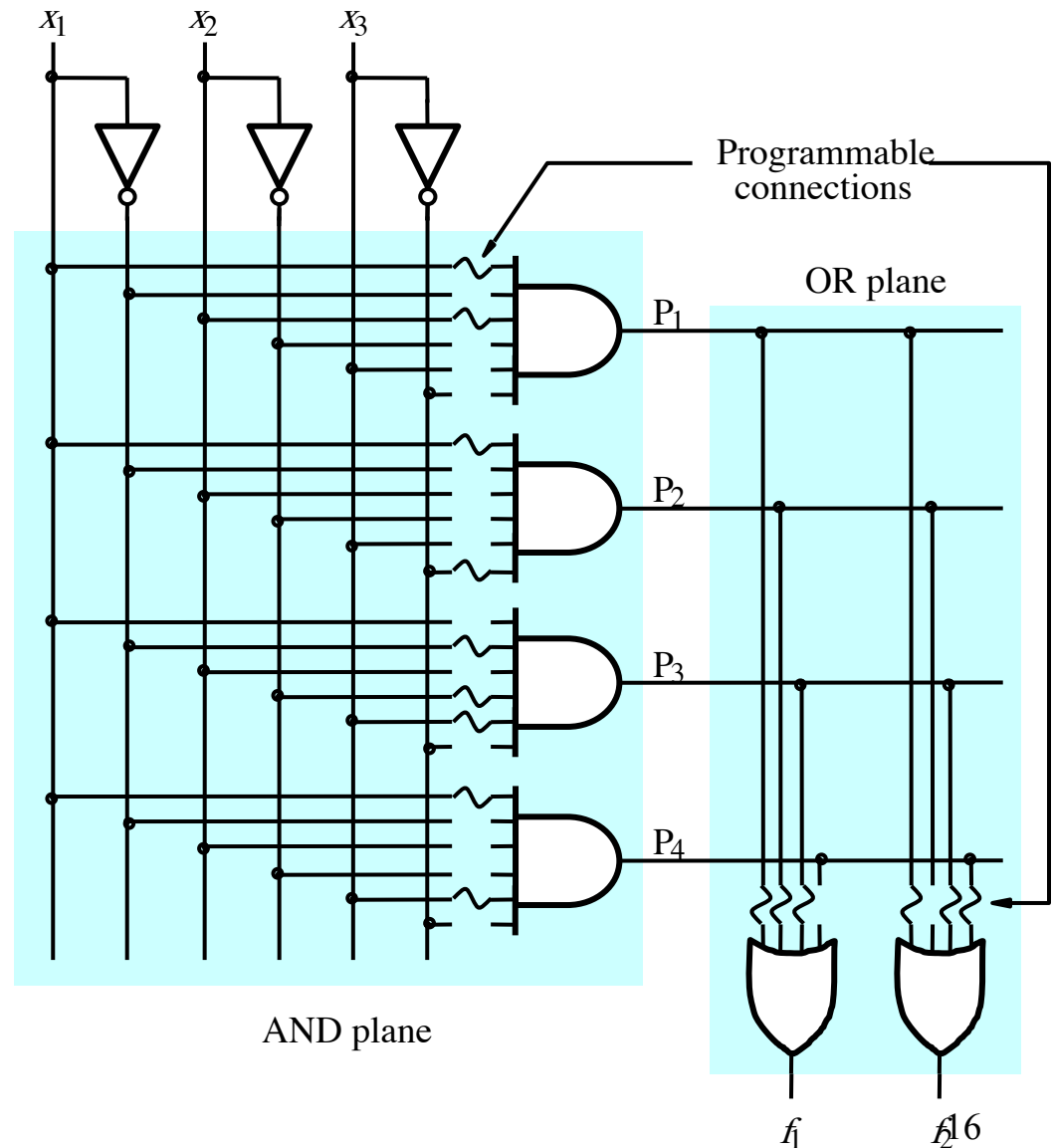 - Field-programmable gate array (FPGA)

Programmable Logic Array (PLA)

- Comprises a collection of buffers, inverters, AND gates, OR gates
- Can be used to realize logic circuit in sum-of-products (SOP) form,
- Example:
  f = x'yz + xy'z'

x_1 x_2 x_n

Input buffers and inverters

x_1 $\bar{x}_1$ x_n $\bar{x}_n$

AND plane

P_1

P_k

OR plane

f_1 f_m

15

Programmable Logic Array (PLA)

- Buffers and inverters provide both true value and complement of each input
- AND plane provides the product terms
- OR plane provides the sum of the product terms
- Example:
  - P1 = x1x2
  - P2 = x1x3'
  - P3 = x1'x2'x3
  - P4 = x1x3
  - F1 = P1 + P2 + P3
  - F2 = P1 + P3 + P4

Programmable connections

OR plane

AND plane

x_1 x_2 x_3

P_1 P_2 P_3 P_4

f_1 f_2

16

Programmable Logic Array (PLA)

- Each AND gate has 2xN inputs
  - N, number of primary inputs
- Each OR gate has M inputs
  - M, number of and gates
- **Problem**: size of the inputs
- Commercially available PLAs typically have:
  - 16 inputs
  - 32 AND gates
  - 8 OR gates
- Connections replaced by single lines, "x" indicates a connected input to the gate

OR plane

AND plane

x_1 x_2 x_3

P_1 P_2 P_3 P_4

f_1 f_2
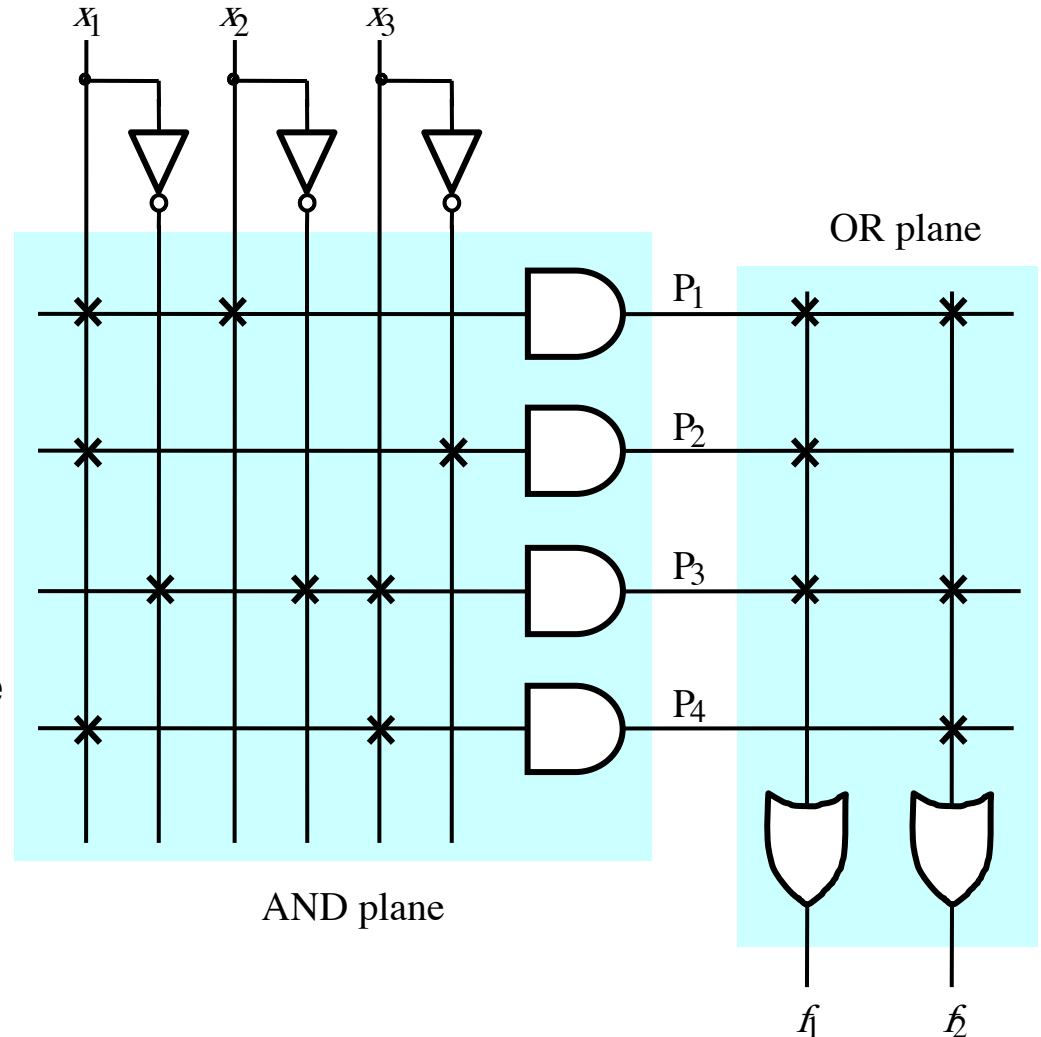
Programmable Array Logic (PAL)

- Drawbacks of PLA
  - Hard to fabricate correctly due to the programmable connections
  - Special implementation of the programmable connections reduce the speed of circuits in PLA

- Solution: fix the OR plane – PAL
  - Less expensive
  - Better performance
  - Became popular in practical applications

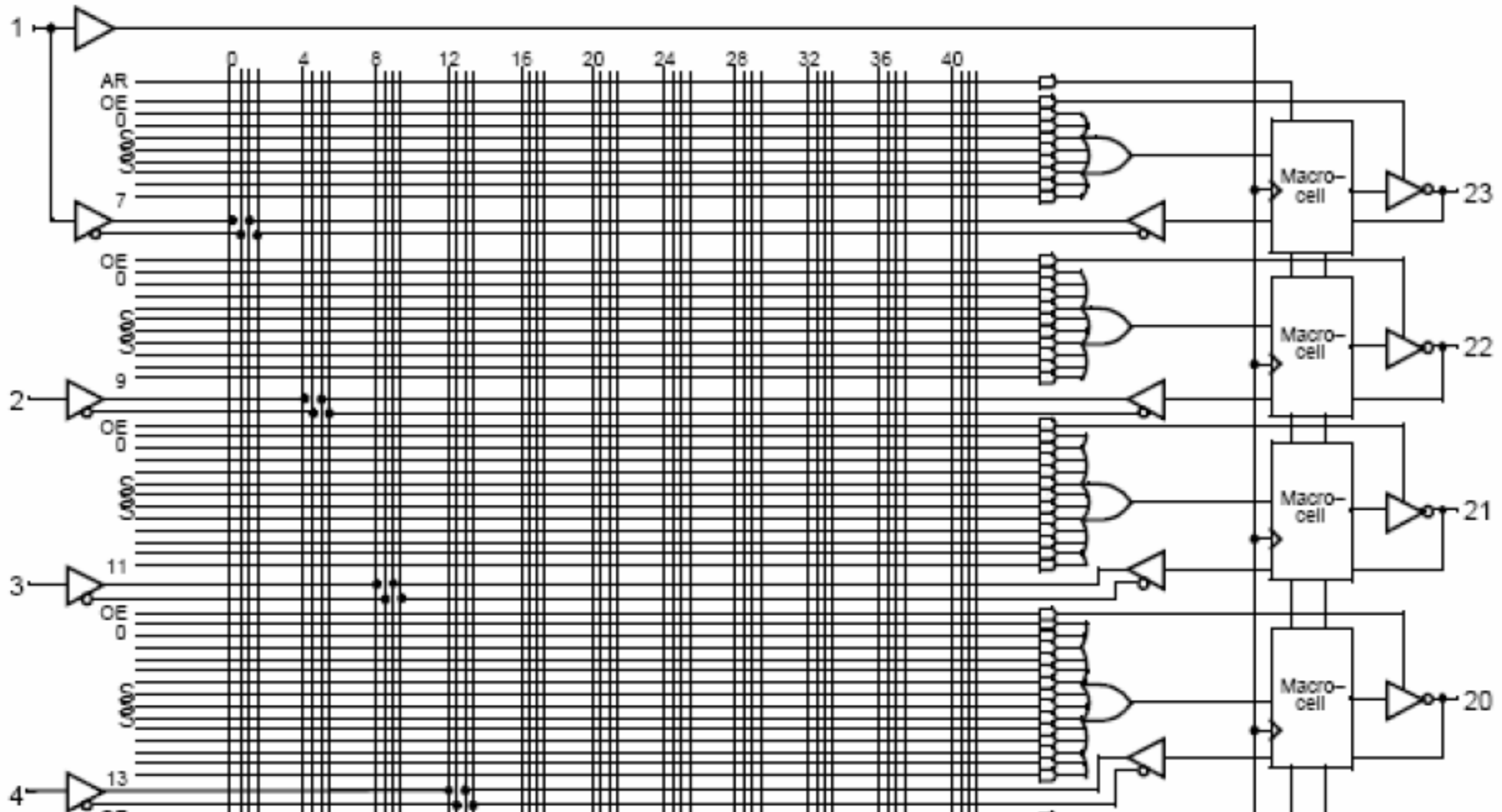# Programmable Array Logic (PAL)

AND plane
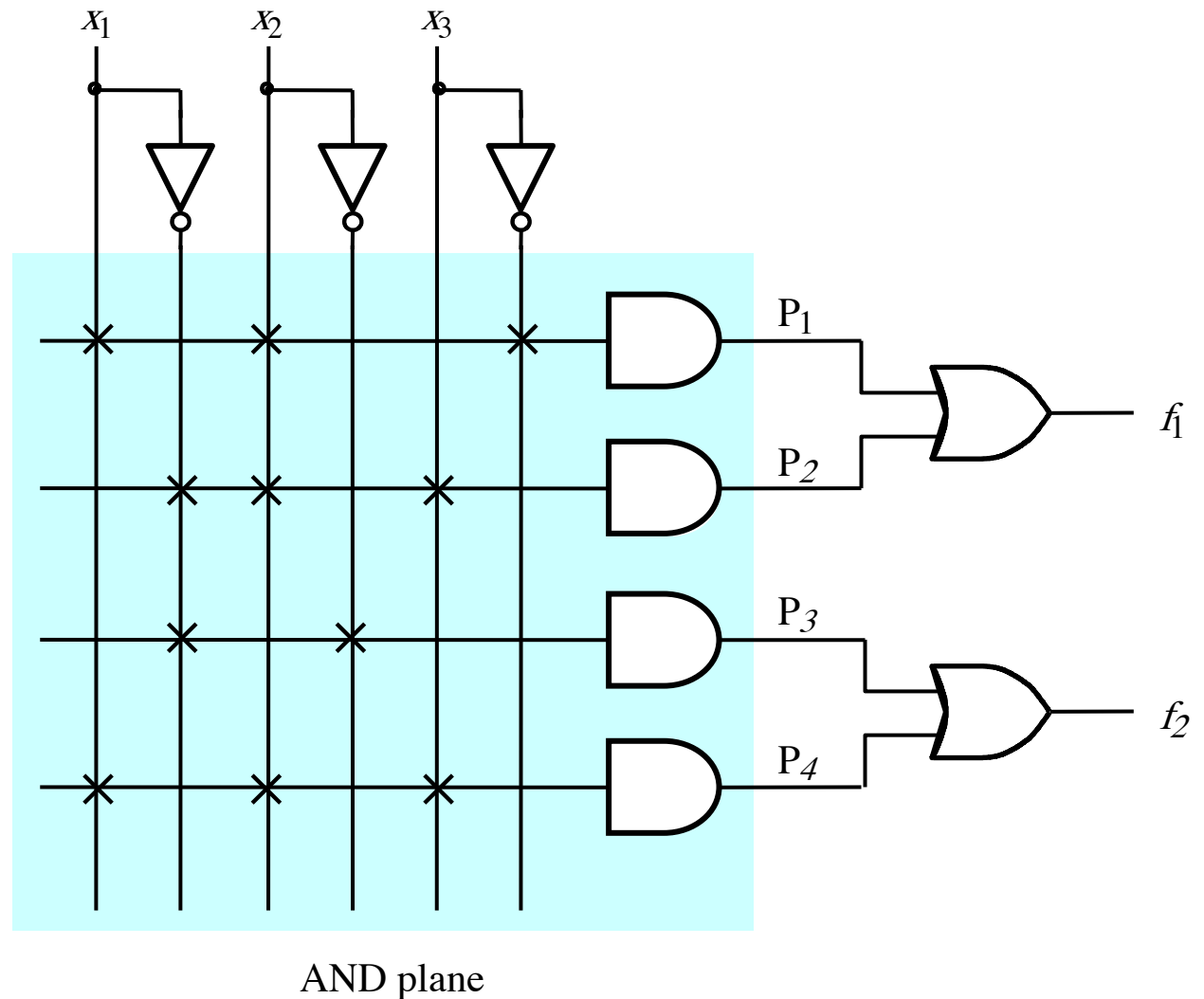
Programmable Array Logic (PAL)

- In order to provide additional flexibility, an extra circuit is inserted between the OR output and the chip pin - *Macrocell*
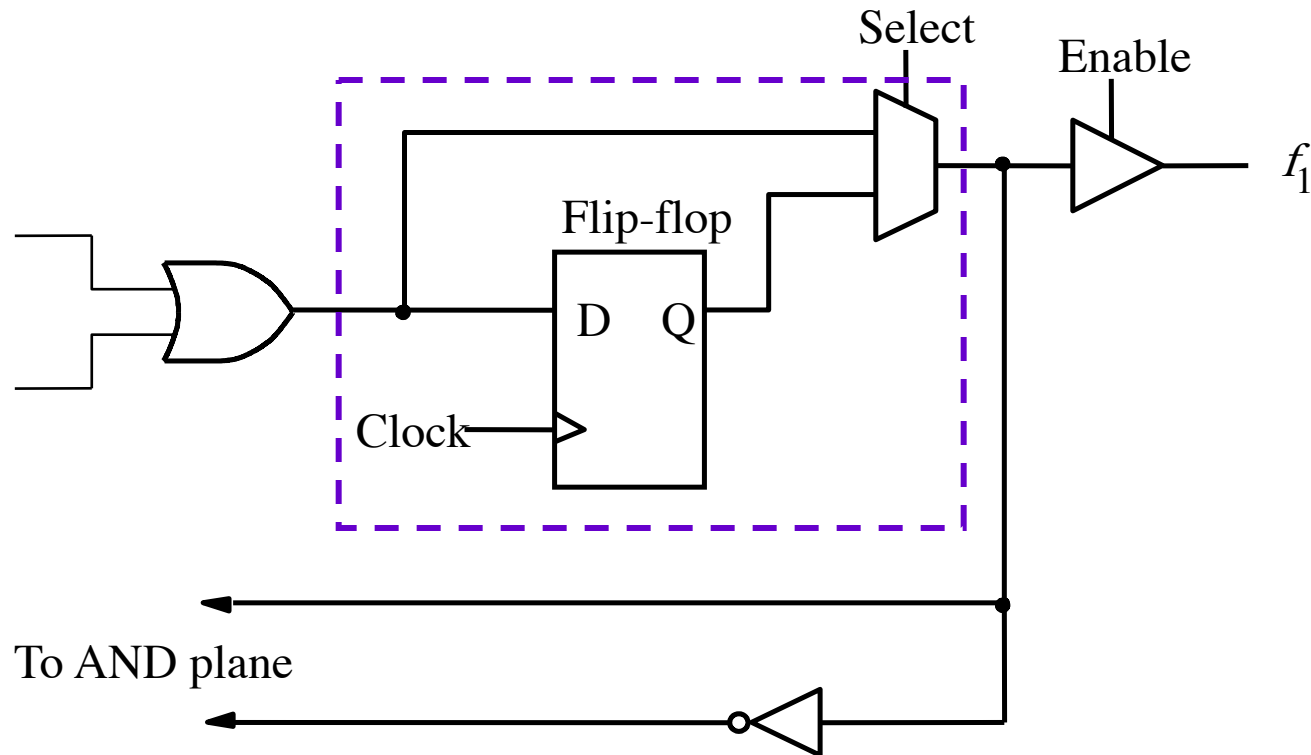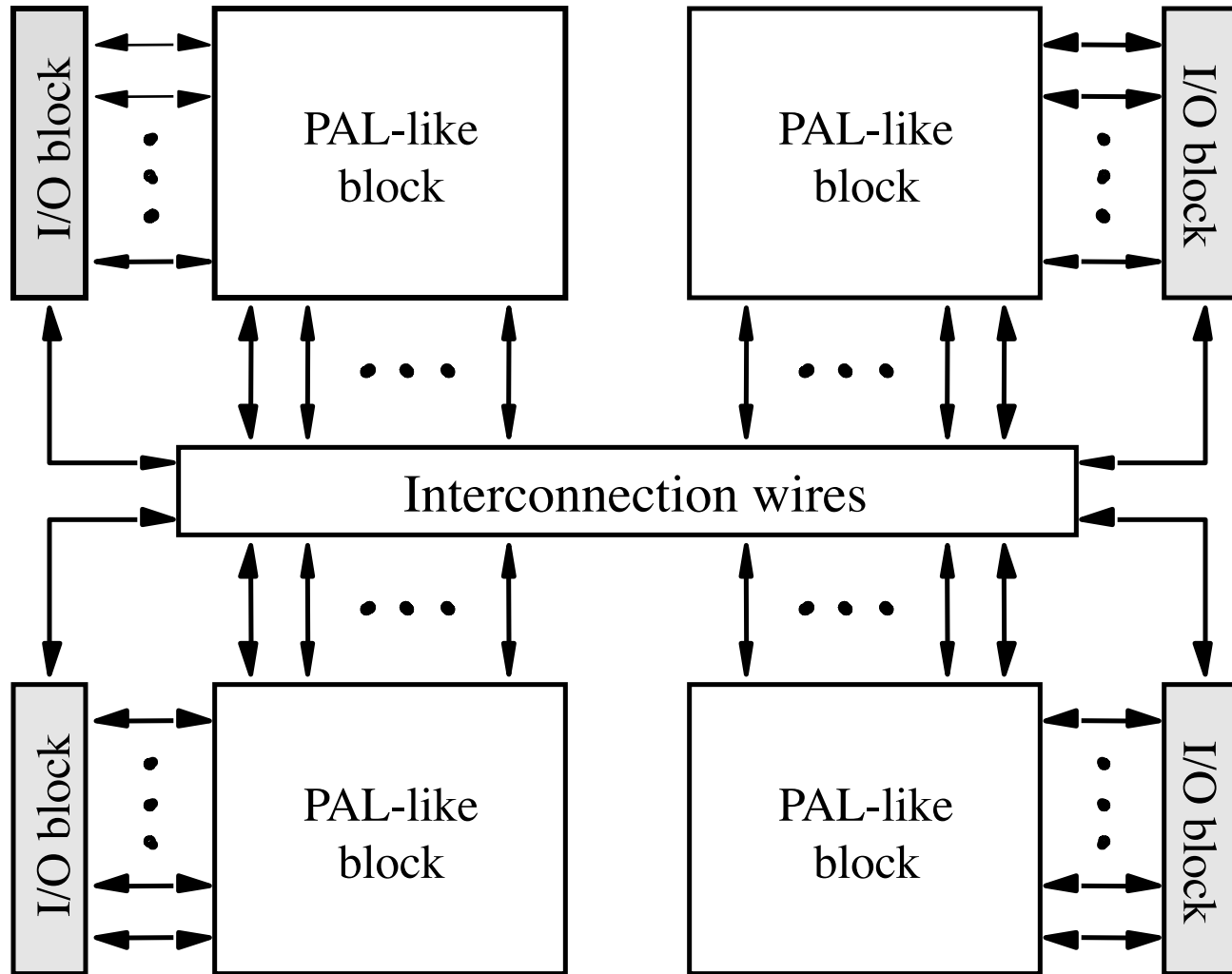
PAL Example

- Compensate for the reduced flexibility
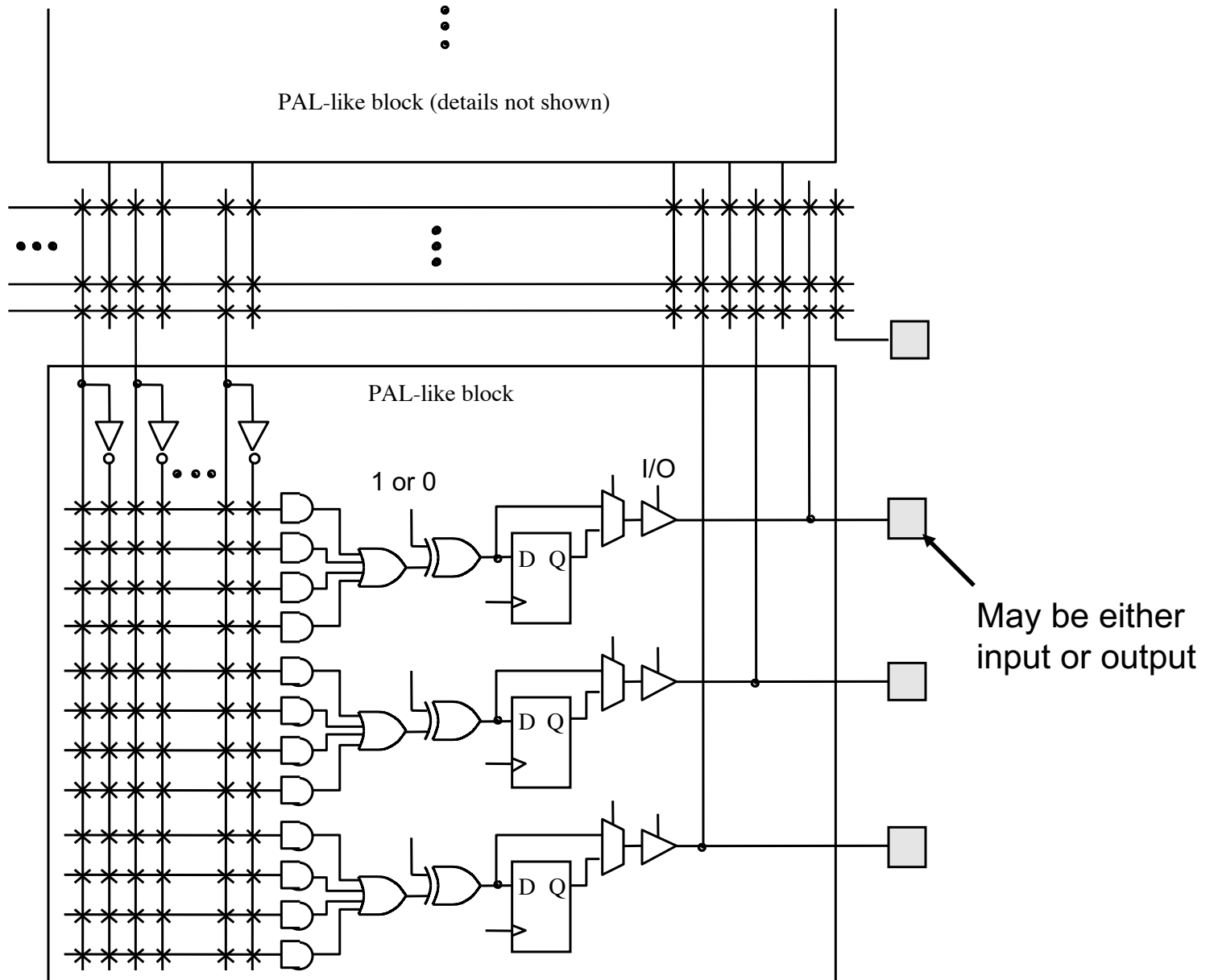 - Various numbers of inputs to the OR gates

Complex Programmable Logic Devices (CPLD)

- Composed of multiple PAL/PLA-like circuit blocks
  - Blocks are connected through a set of interconnection wires
  - Blocks are connected to the IC chip pins through a set of I/O blocks
  - Number of blocks may vary from 2 to over 100
- Provides more inputs and outputs
- Provides more flexibility
- May accommodate bigger size circuit

CPLD

PAL-like block (details not shown)

PAL-like block

1 or 0

I/O

D Q

D Q

D Q

May be either
input or output

24

Field Programmable Gate Array (FPGA)

- First introduced by Xilinx in 1985
- Most FPGA providers are "fabless", allows
  - focus on device capability
  - improvement of design software
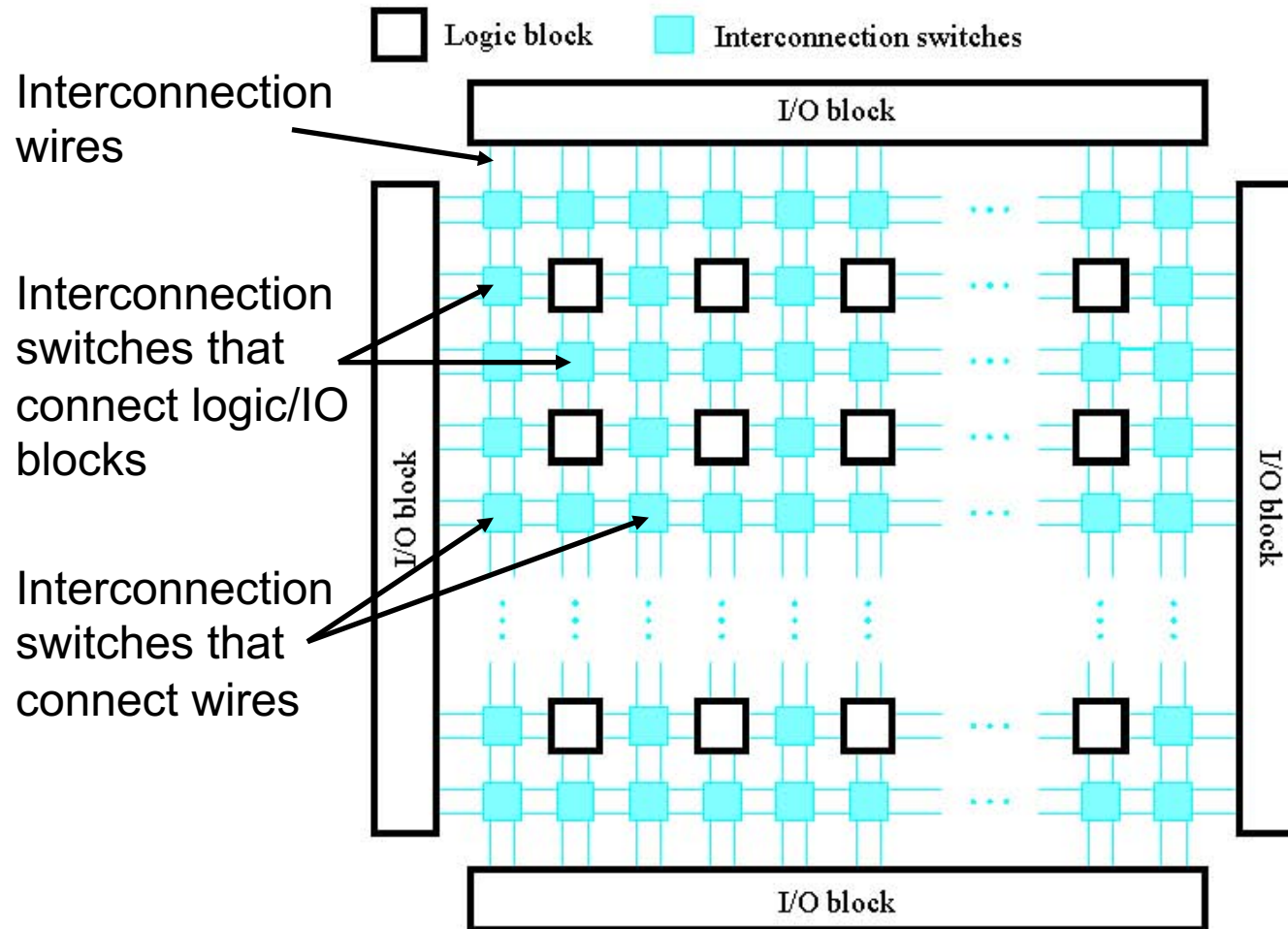 - offering IP cores

Types of FPGA

- ## Reprogrammable
  - ### SRAM-based FPGA
    - Volatile, often the best choice for prototyping and development
    - Supports in-system-programming (ISP)
    - What we used in the labs
  - ### EEPROM-based (Flash-based) FPGA

- ## One-Time Programmable (OTP)
  - ### Anti-Fuse-based FPGA
  - ### EPROM-based FPGA

Internal Structure

- Composed of logic blocks and wires
  - Configurable Logic blocks (CLB)
  - I/O blocks (IOB)
  - Interconnection wires and switches

FPGA Architecture

- Typical FPGA architecture

Interconnection wires

Interconnection switches that connect logic/IO blocks

Interconnection switches that connect wires

Logic block   Interconnection switches

I/O block

I/O block

I/O block

I/O block

Courtesy of Xilinx Inc.

Programmable (Configurable) Logic Block

- PLB (or CLB), like in SPLD and CPLD, macrocell is added to provide more flexibility

In$_1$

In$_2$

In$_3$

LUT

Select

Flip-flop

D   Q

Clock

Out

Macrocell
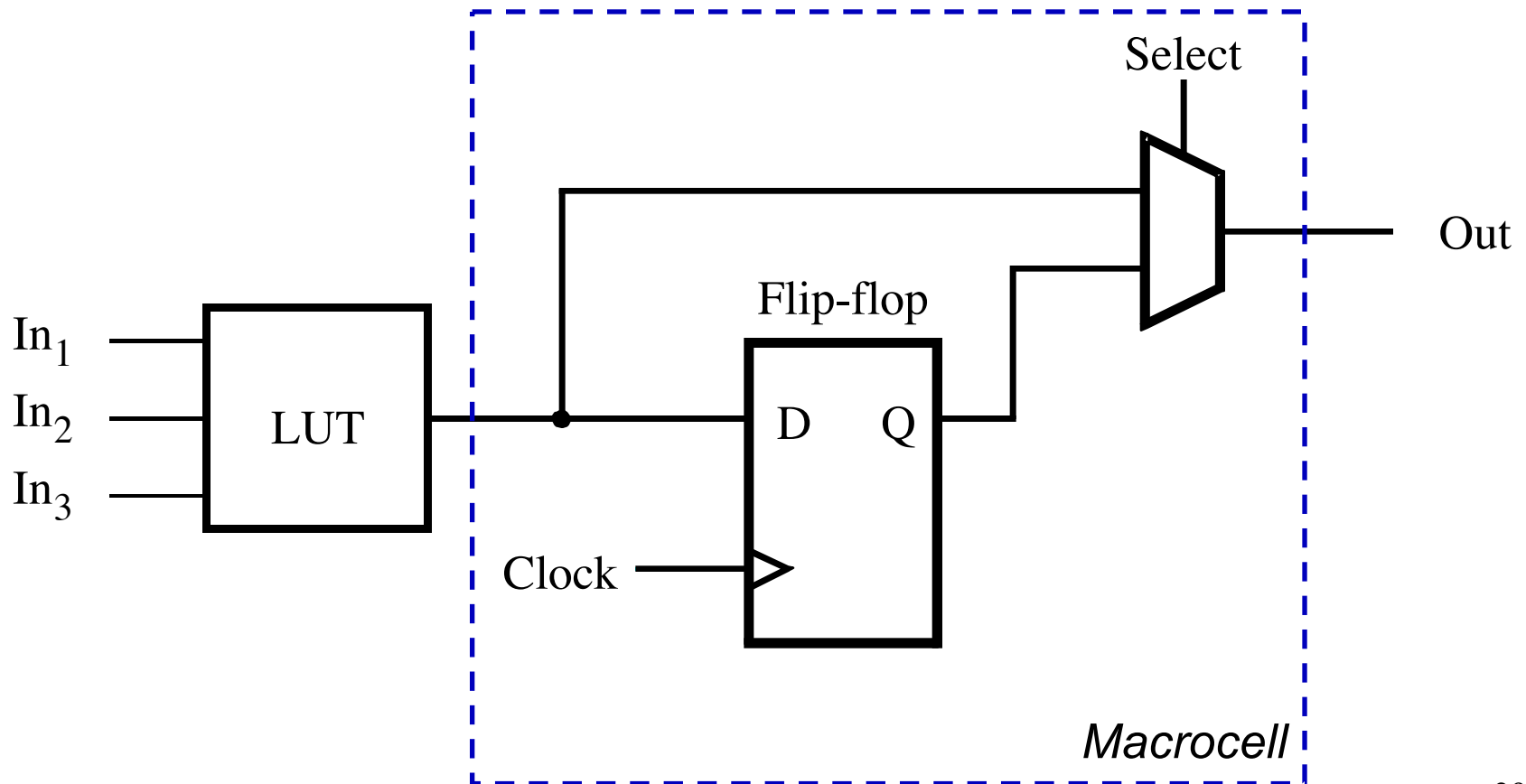
Look Up Table (LUT)

- A typical PLB has a LUT
- Each LUT contains 2^N storage cells, N is the number of inputs to the LUT
  - **SRAM** for storage cell
  - Each storage cell can hold a value, either "1" or "0"
  - The cells are programmed to implement particular logic functions
  - The cells may be reconfigured to implement a different logic function in the same LUT
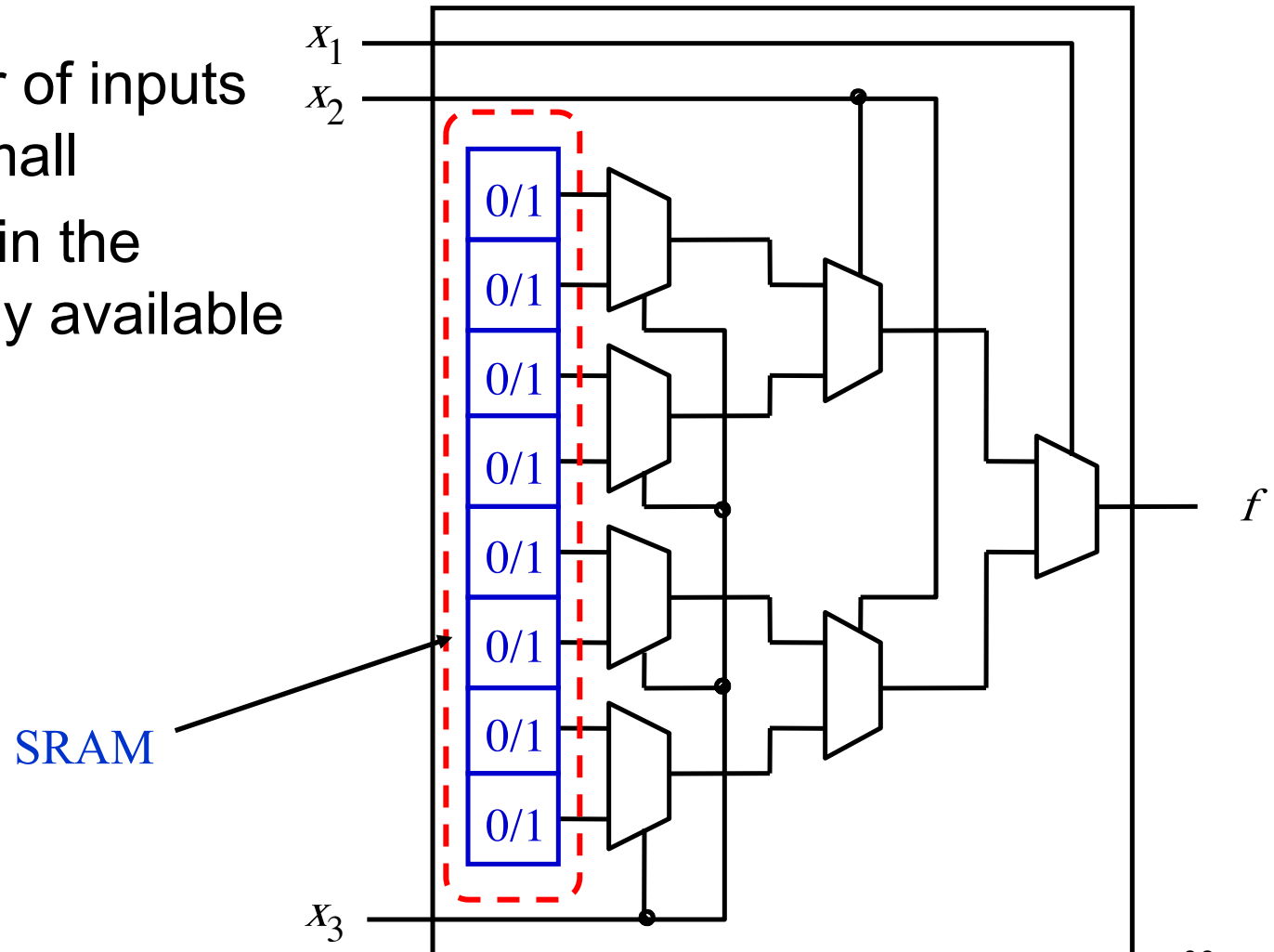- N input LUT can implement any N variable logic function

x_1

0/1

0/1

0/1

0/1

f

x_2
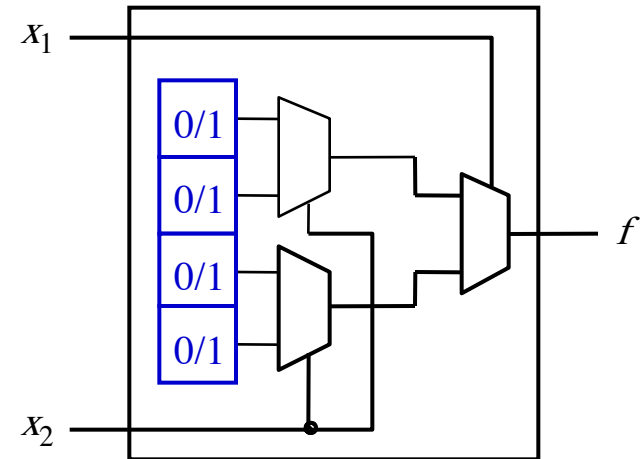
LUT Example

- Implement the logic function specified by the truth table using a 2-input LUT

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

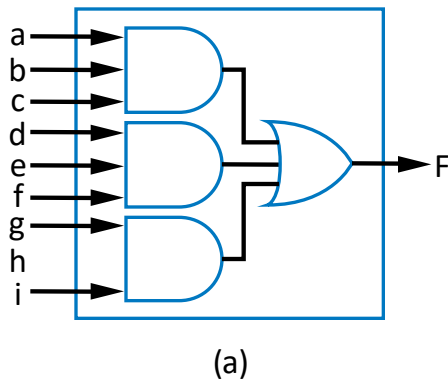f1 = x_1'x_2' + $x_1 x_2$
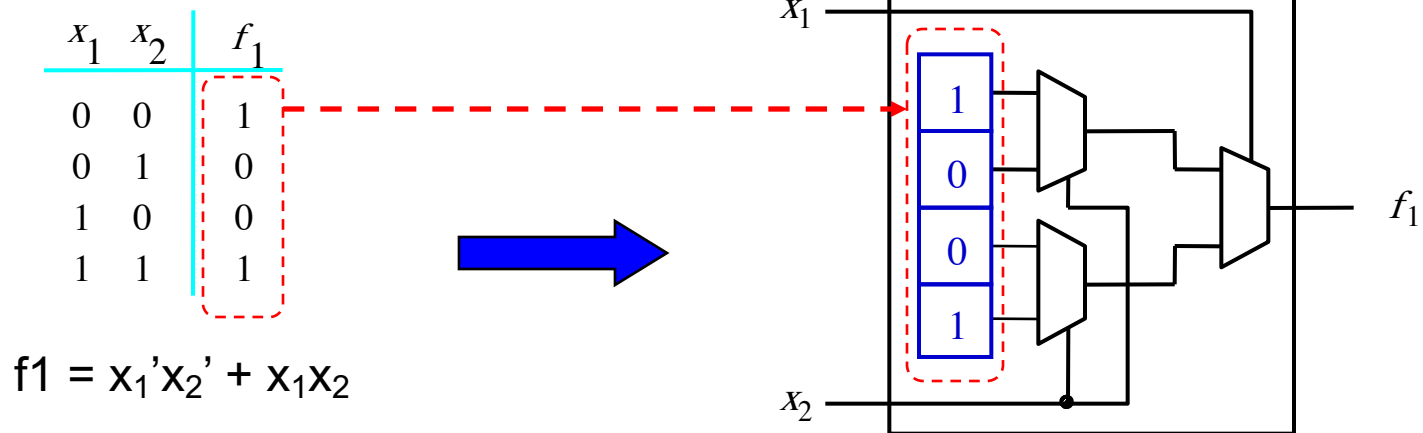
3-input LUT

- The number of inputs to LUT is small
- 4 - 6 inputs in the commercially available FPGAs
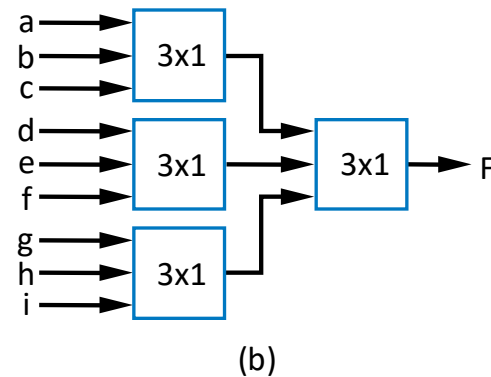

SRAM

FPGA Internals: Lookup Tables (LUTs)

- Implement bigger circuit with smaller LUTs
  - Example: 9-input circuit

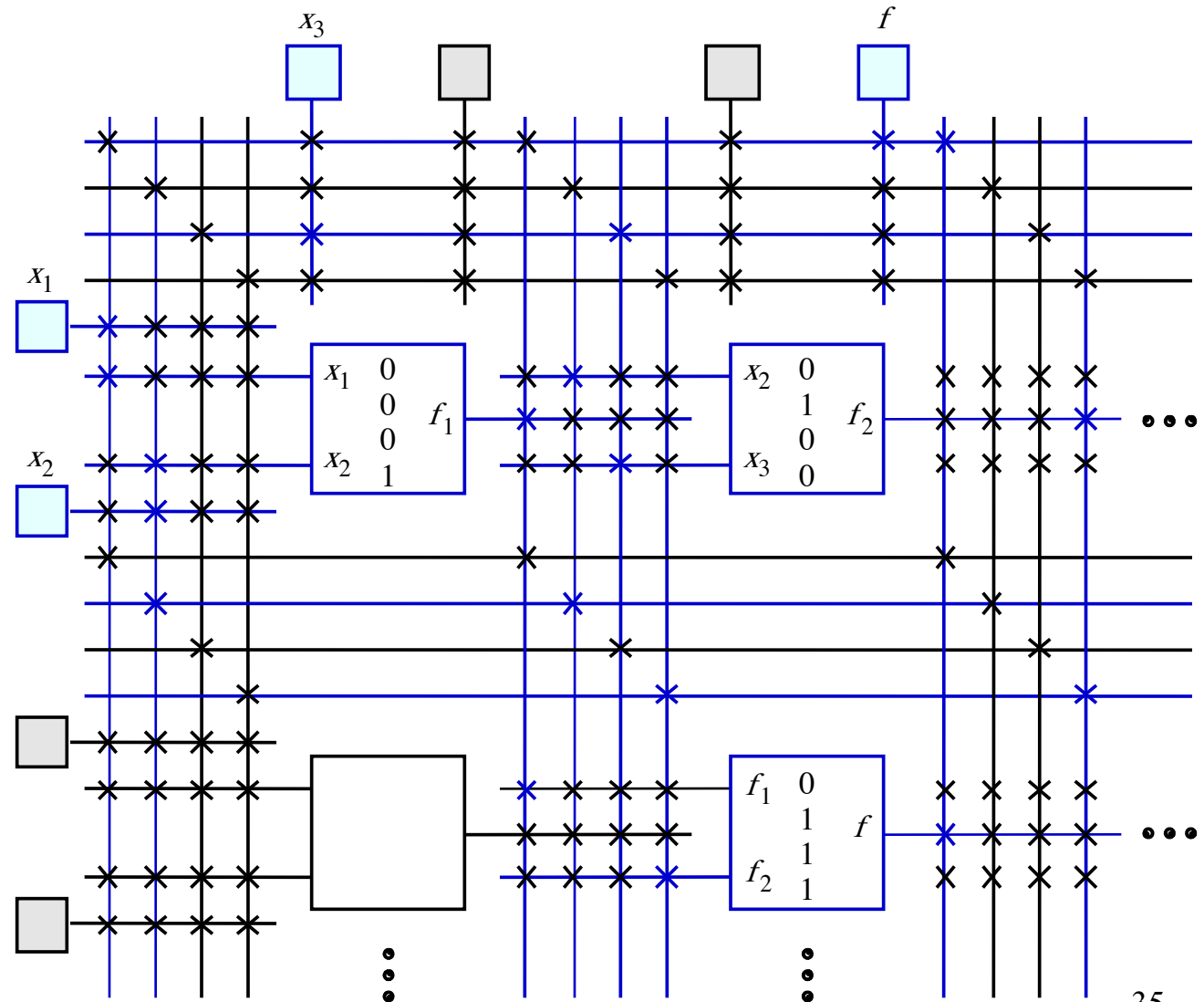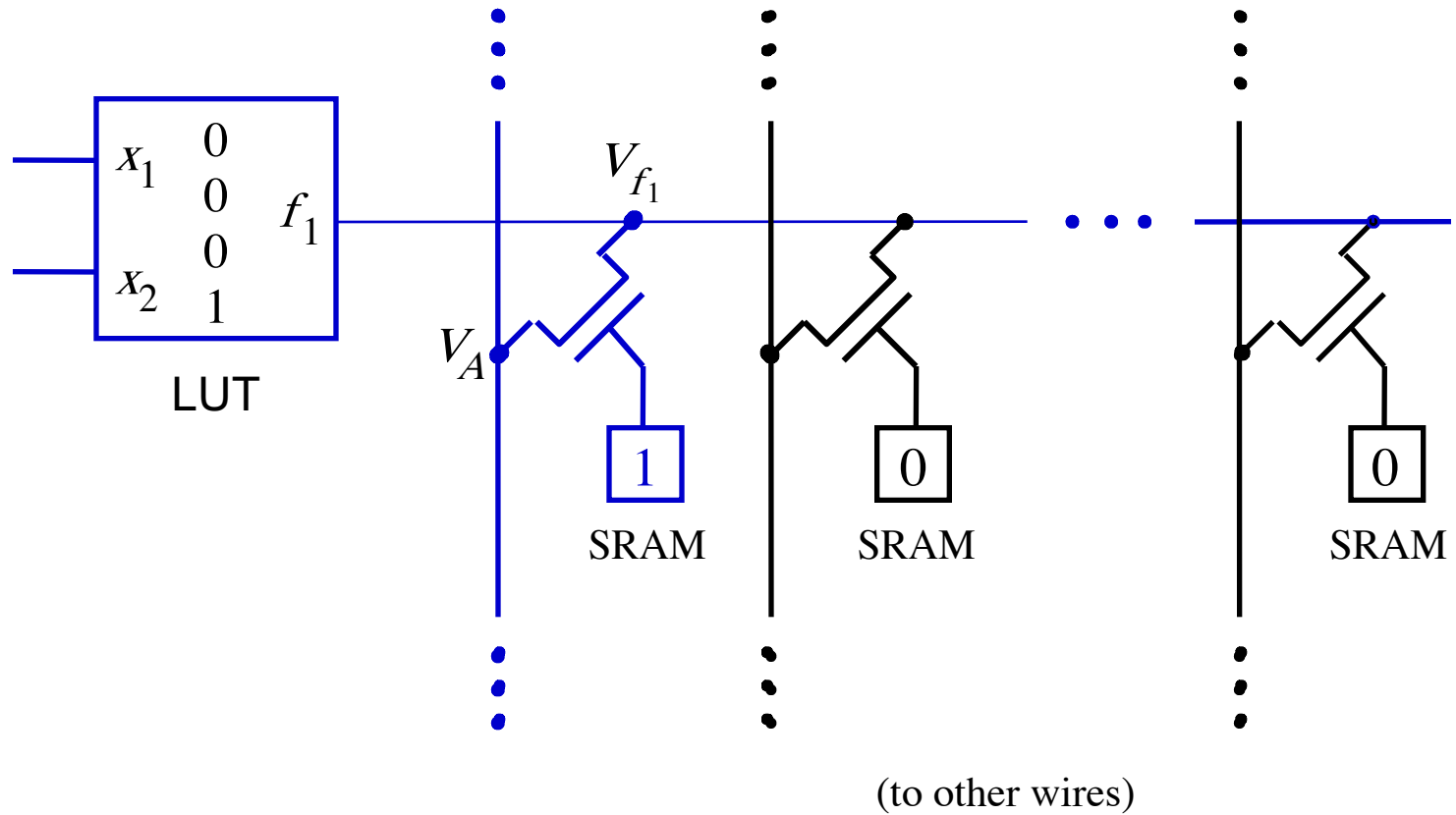Original 9-input circuit   Implemented with
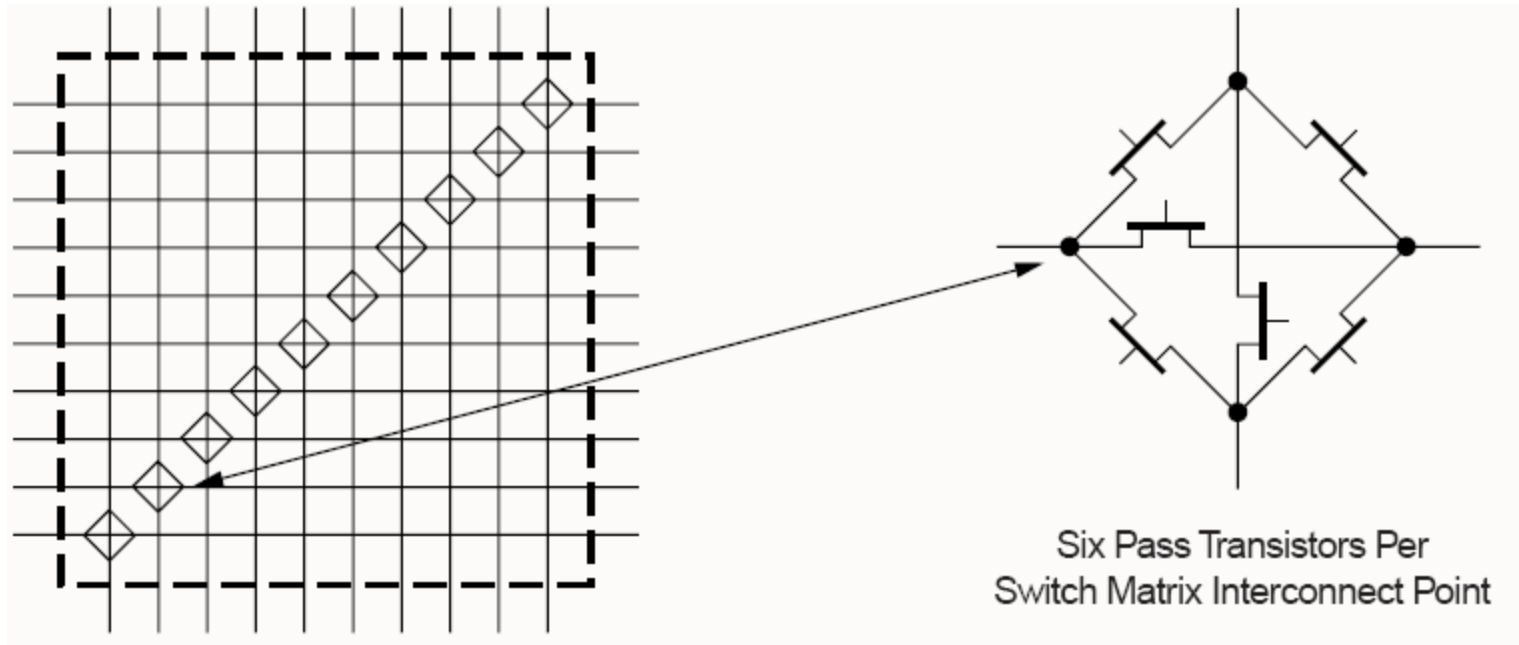   3x1 LUTs

FPGA Configuration Example

- Blue x indicates a connection
- f1 = $x_1 x_2$
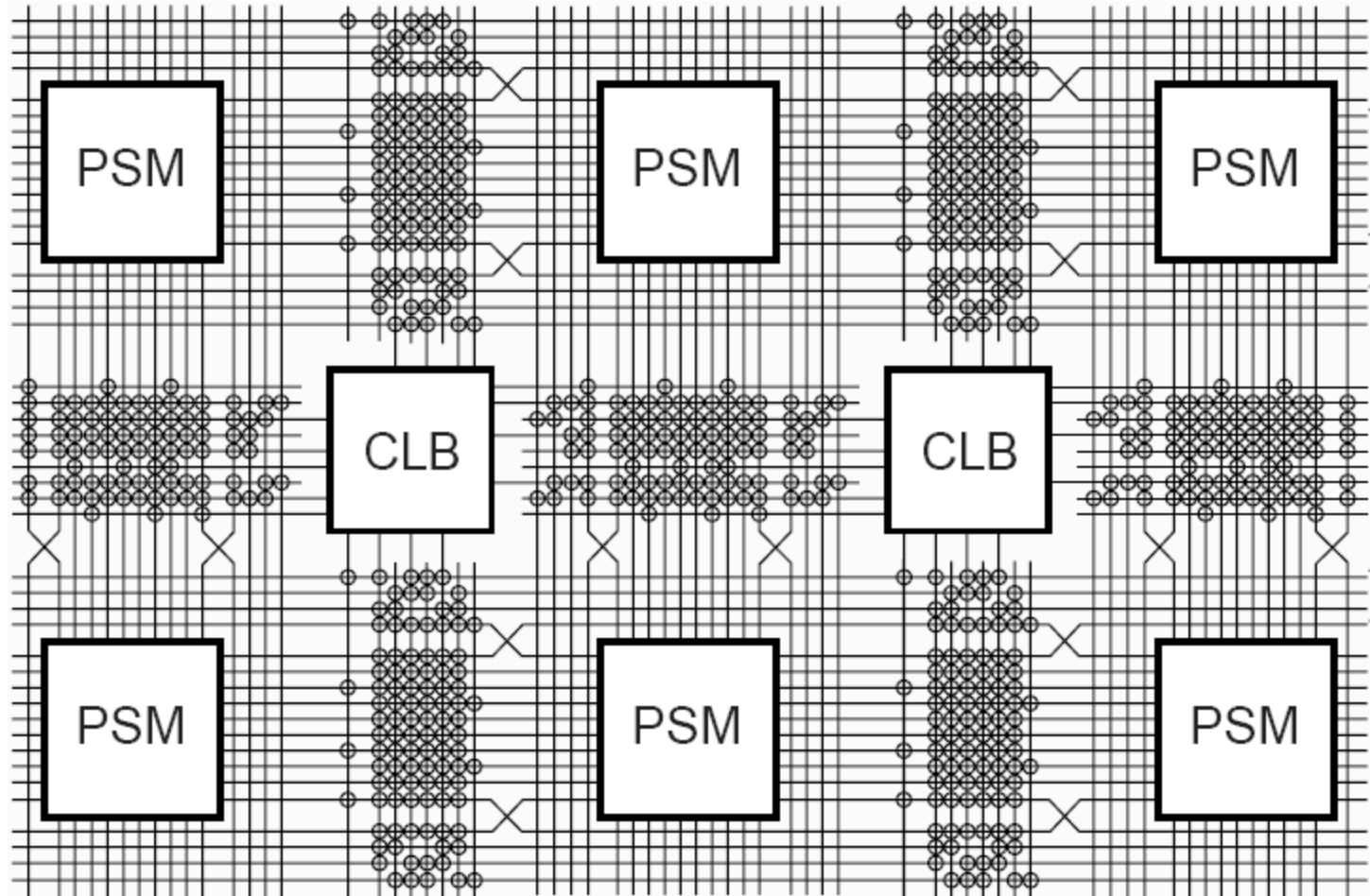- f2 = $x_2' x_3$
- f = f1 + f2

Programmable Switch

(to other wires)

Xilinx FPGA PSM

Six Pass Transistors Per
Switch Matrix Interconnect Point
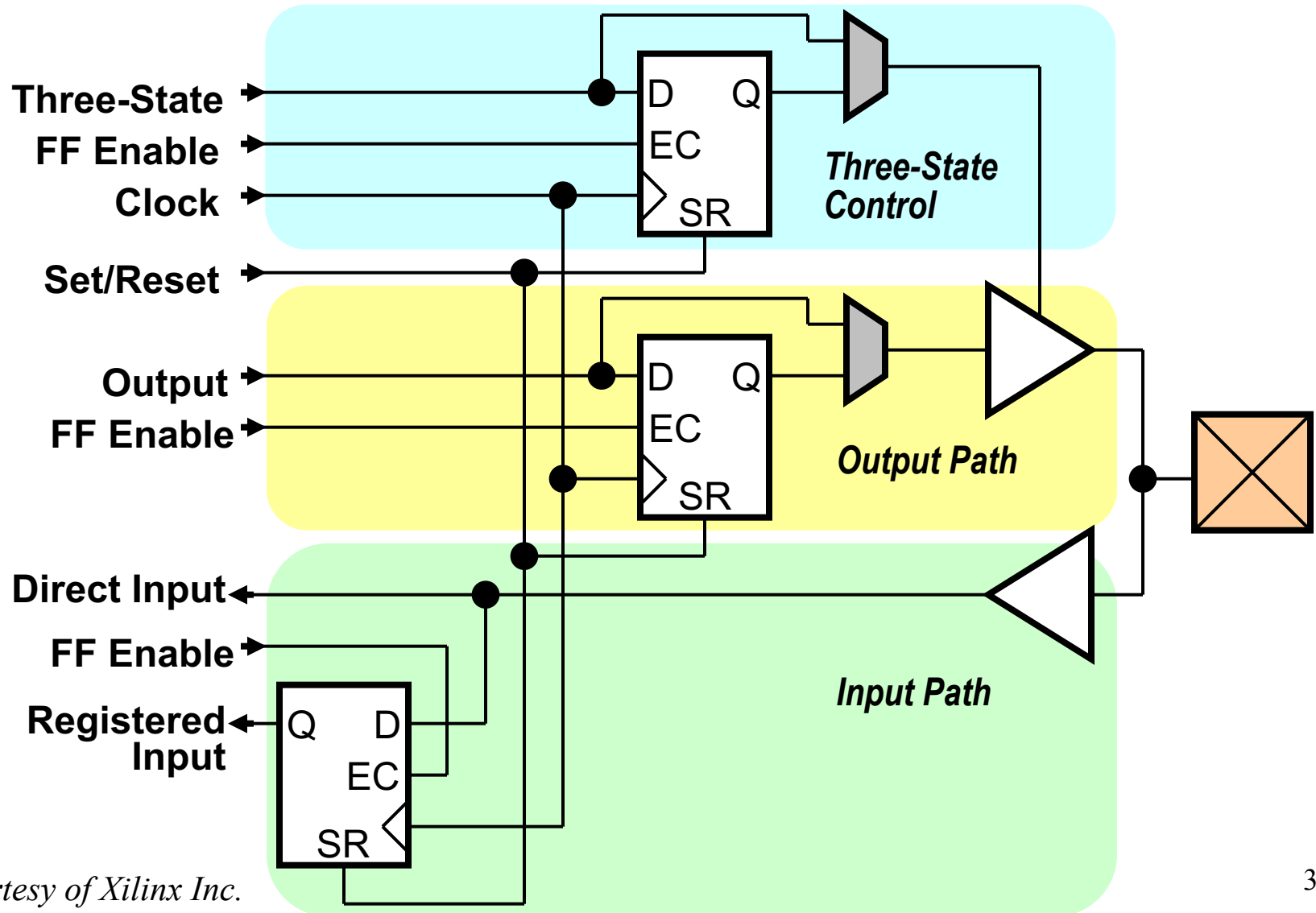
Programmable Switch Matrix (PSM)

Xilinx FPGA Switches and Wires

Courtesy of Xilinx Inc.

Basic I/O Block Structure

Three-State
Control

Output Path

Input Path

Three-State

FF Enable

Clock

Set/Reset

Output

FF Enable

Direct Input

FF Enable

Registered
Input

D   Q
EC
SR

Courtesy of Xilinx Inc.

39

Advanced FPGA Features

- Dozens of millions of equivalent logic gates

- Enhanced clock features

- Flexible embedded memory blocks

- Intellectual property (IP) cores

- Embedded processors (hard and soft)

- Digital signal processing (DSP) blocks, tools, design flows (specific FPGA vendors)

- Dedicated hardware multipliers

- high-speed communication capabilities

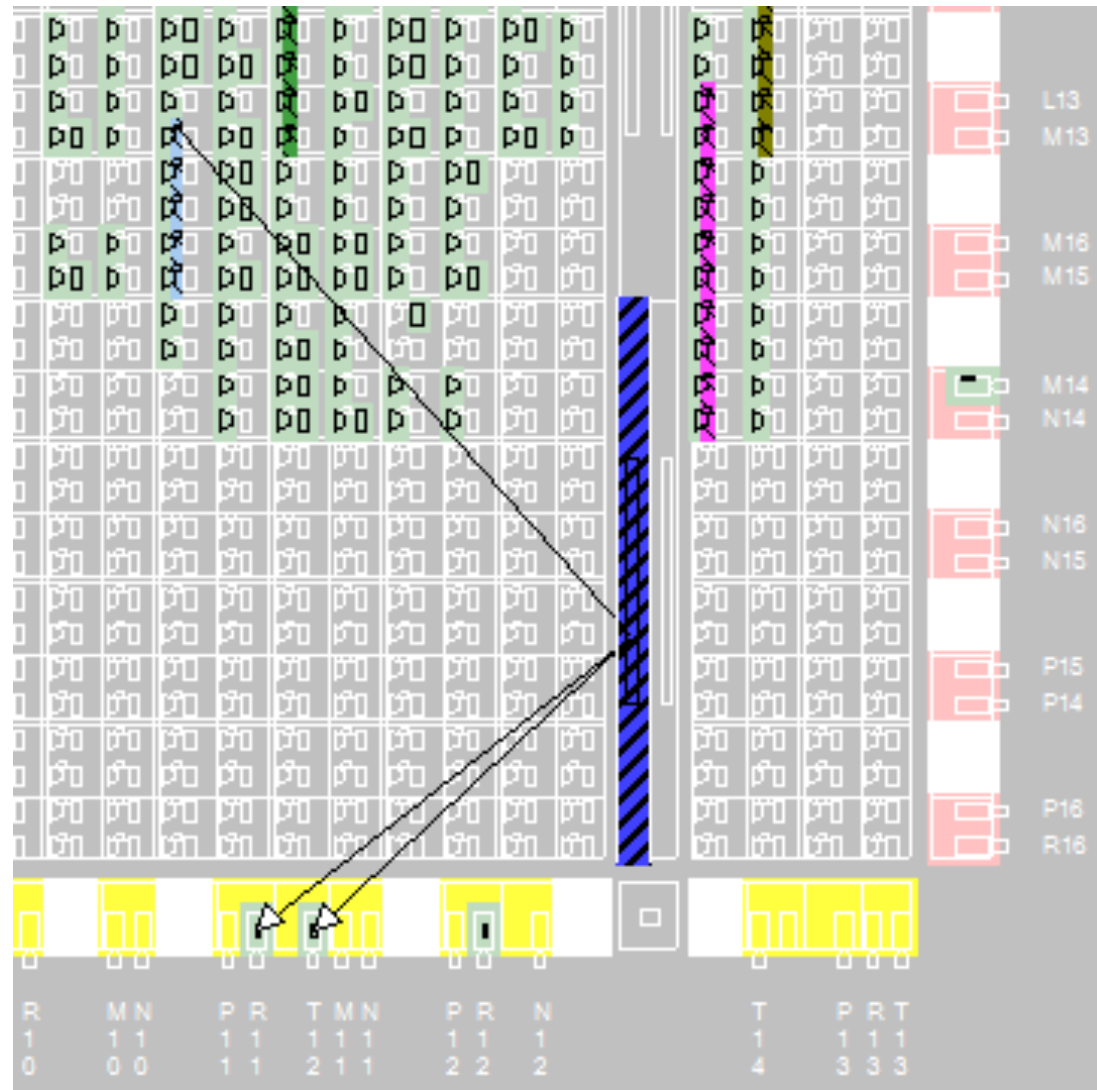- Advanced I/O standards and protocol support

FPGA Example – Spartan 3
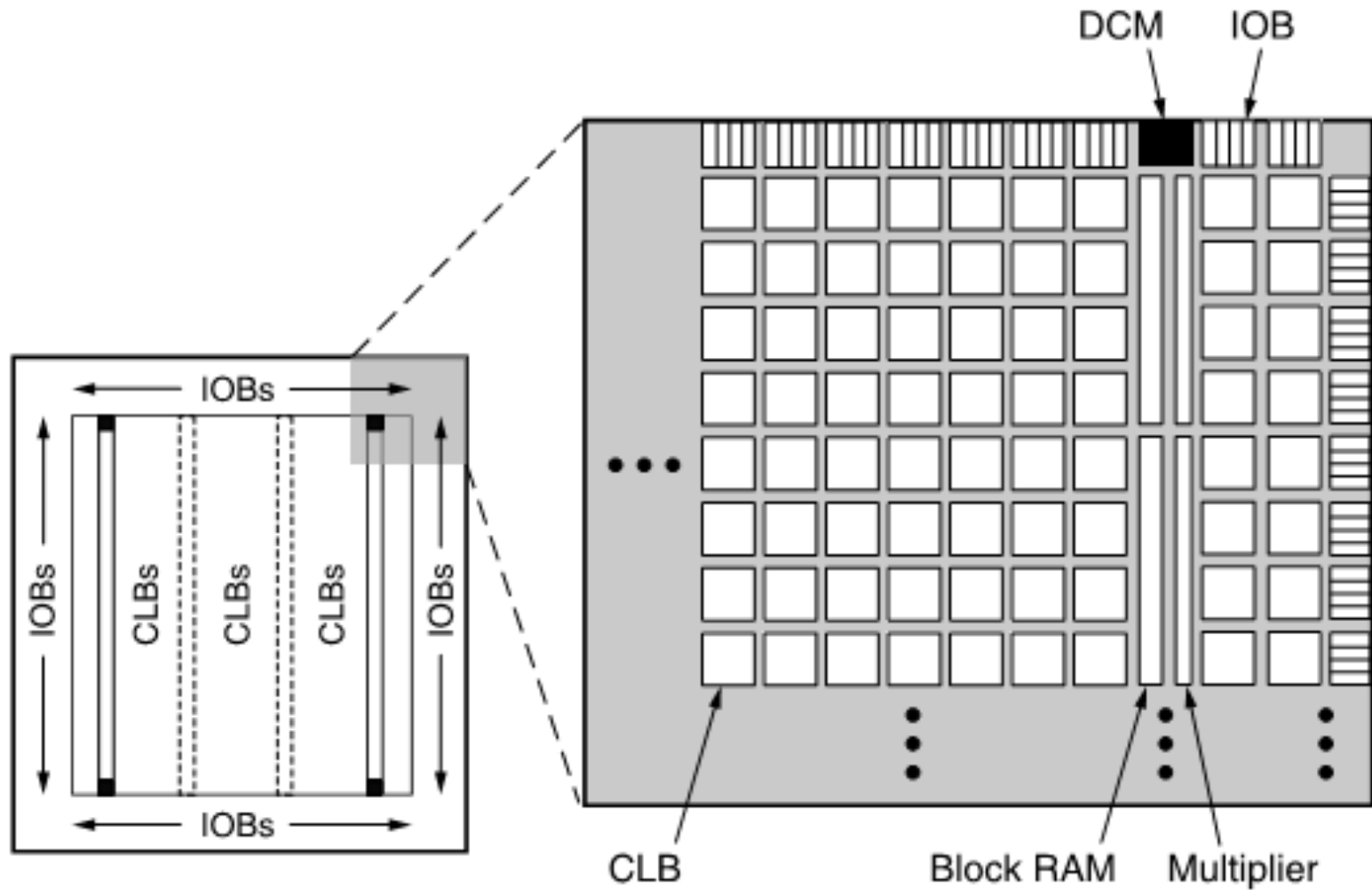
Courtesy of Xilinx Inc.

Example of FPGA Implementation

Pros and Cons of FPGA

- Pros
  - Fast turnaround.
  - Low NRE (non-recurring engineering) cost.
  - Low risk.
  - Effective design verification.
  - Low testing cost.
- Cons (compare to regular IC chip implementation, and are improving)
  - *Bigger chip size*
  - *Higher cost*
  - *Higher power consumption*
  - *Slower speed*
- Technology is still advancing