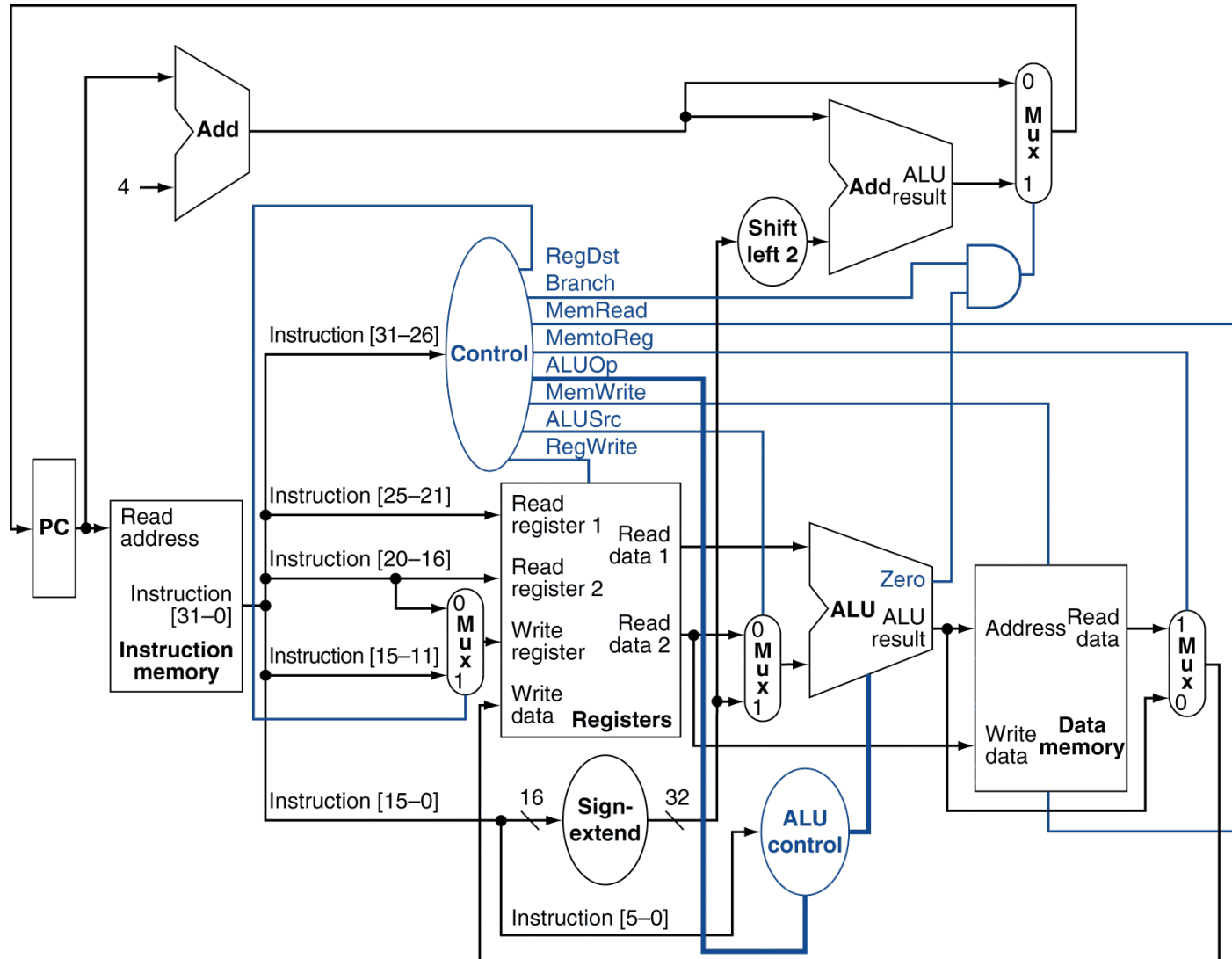


Topic 15

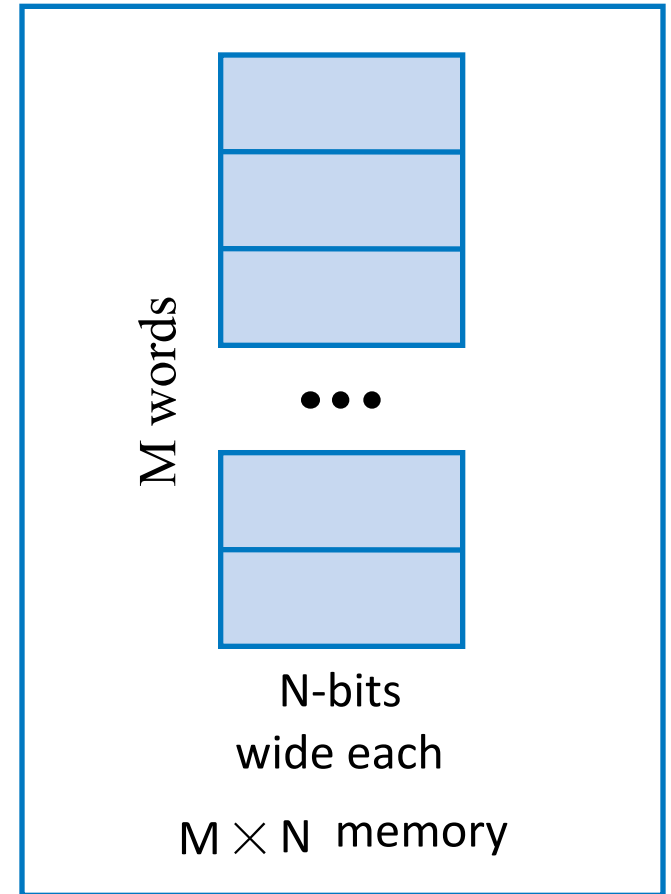
Memory and PLD

Big Picture – Simplified Computer



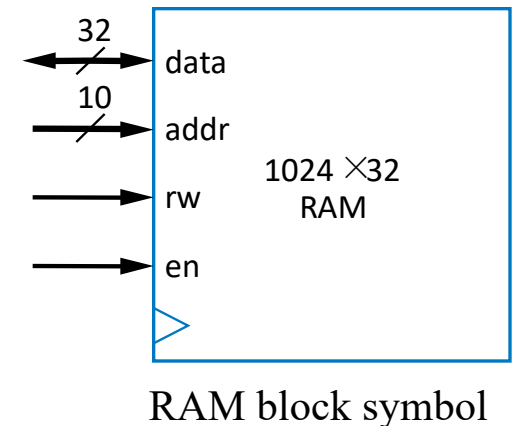
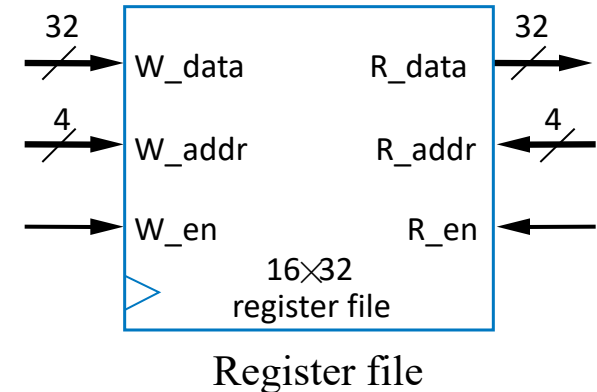
Memory Components

- ***MxN memory***
 - M words (row)
 - N bits (column) wide each
- Types of memory
 - RAM
 - ROM

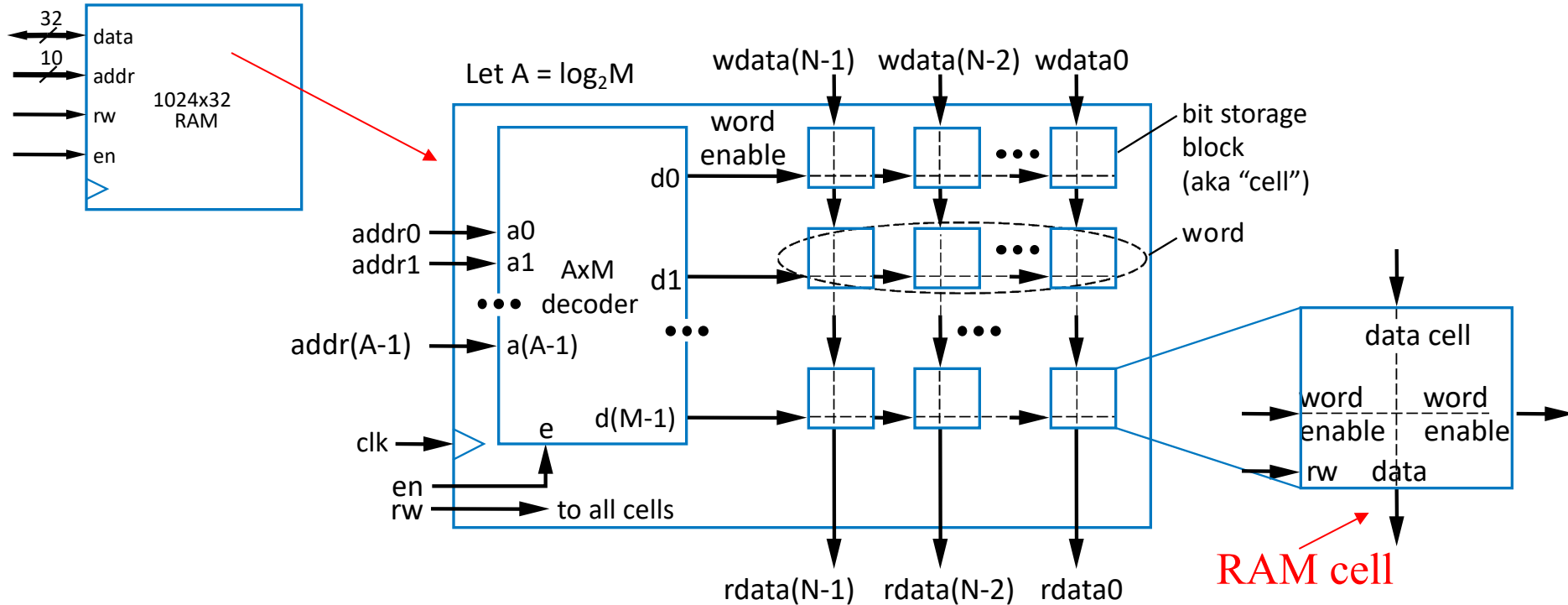


Random Access Memory (RAM)

- 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

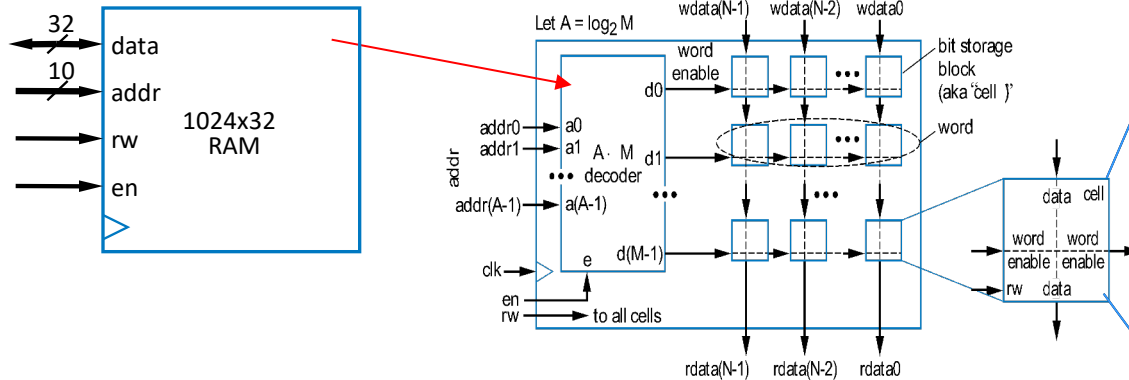


RAM Internal Structure

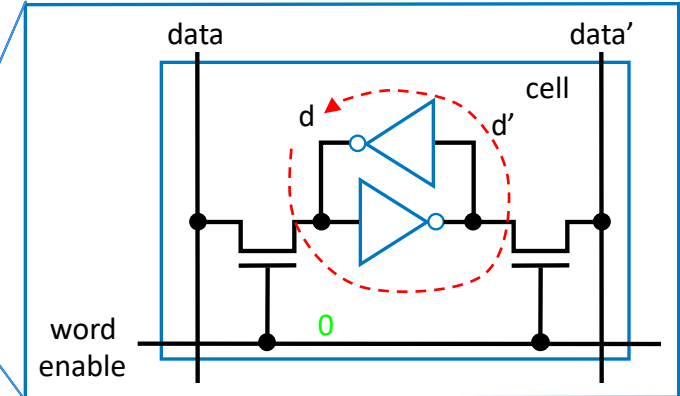


- Similar internal structure as register file
 - Decoder enables appropriate word based on address inputs
 - rw controls whether cell is written or read

Static RAM (SRAM)

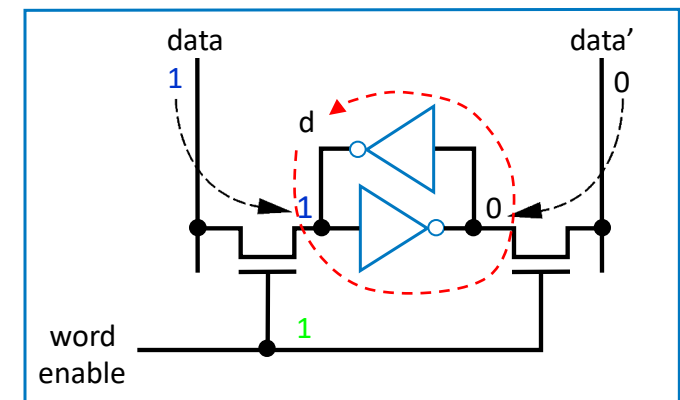


SRAM cell

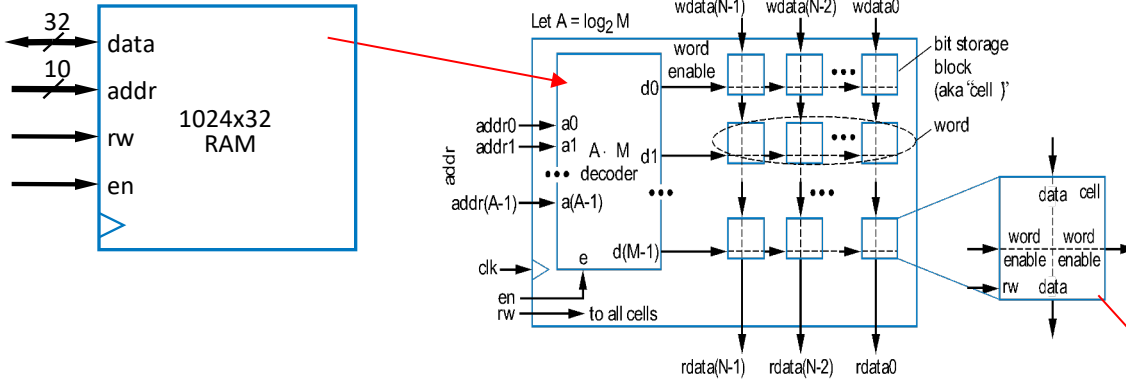


- “Static” RAM cell
 - 6 transistors (recall inverter is 2 transistors)
 - Writing this cell
 - *word enable* input comes from decoder
 - When 0, value *d* loops around inverters
 - That loop is where a bit stays stored
 - When 1, the *data* bit value enters the loop
 - *data* is the bit to be stored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

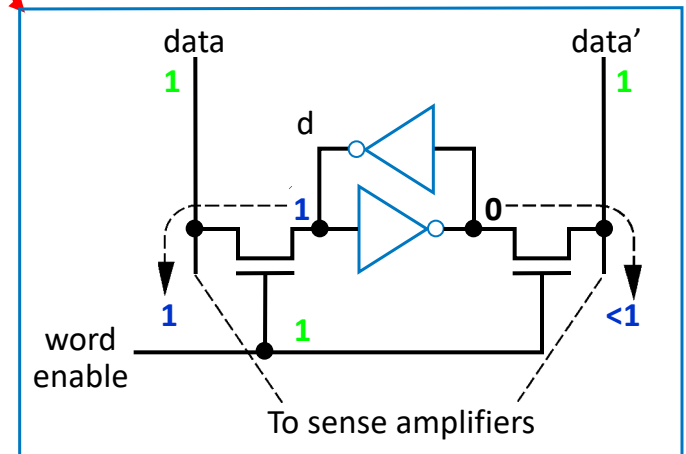
SRAM cell



Static RAM (SRAM)



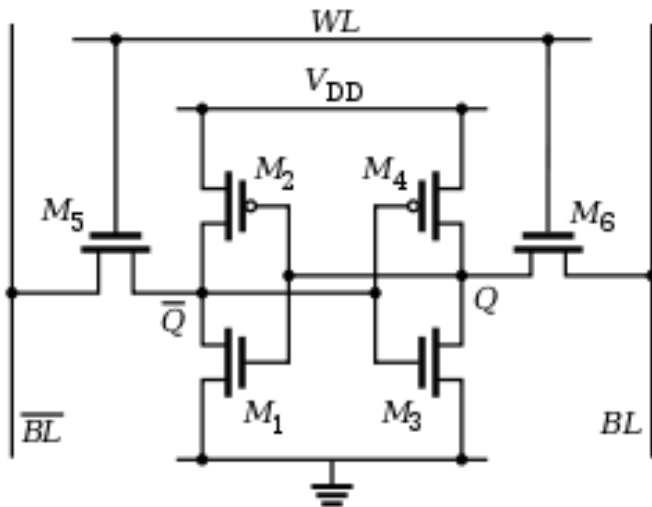
SRAM cell



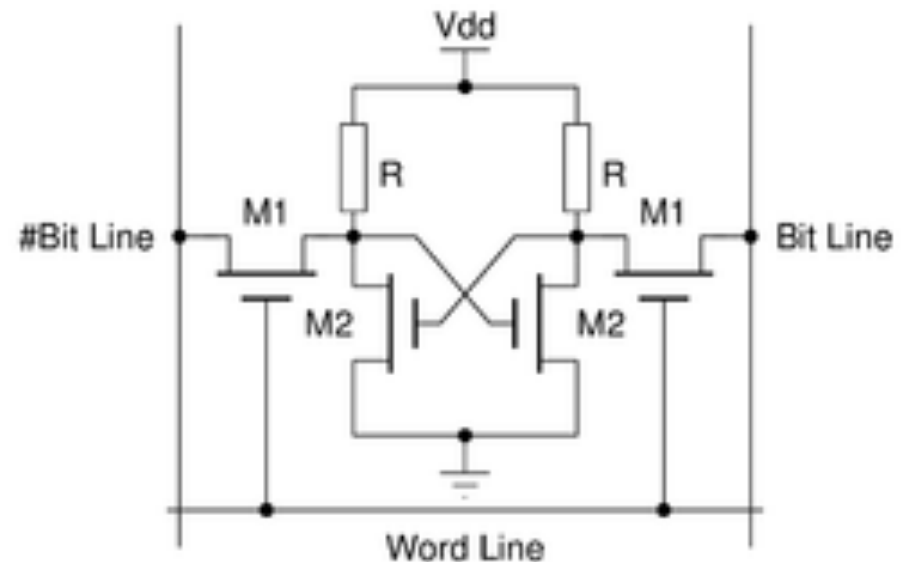
- “Static” RAM cell
 - Reading this cell
 - Somewhat trickier
 - When *rw* set to read, the RAM logic sets both *data* and *data'* to 1
 - The stored bit *d* will pull either the left line or the right line down slightly below 1
 - “Sense amplifiers” detect which side is slightly pulled down

Static RAM (SRAM)

Implementation with 6 transistors

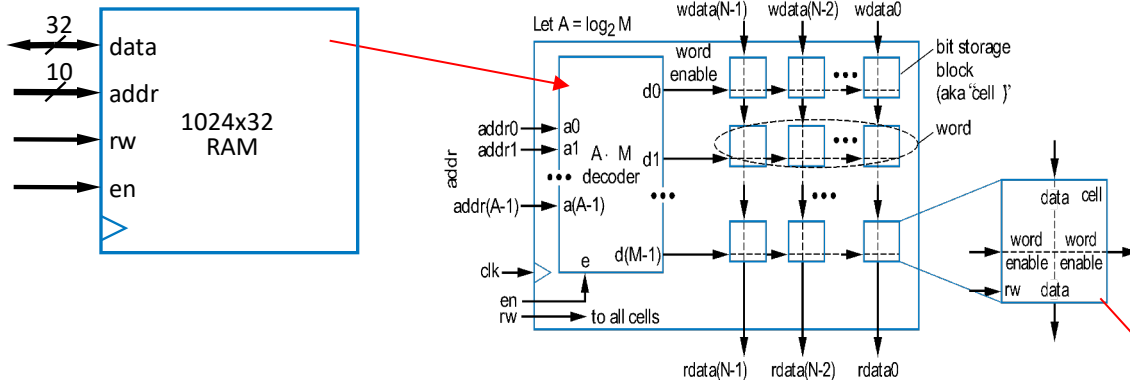


Implementation with 4 transistors



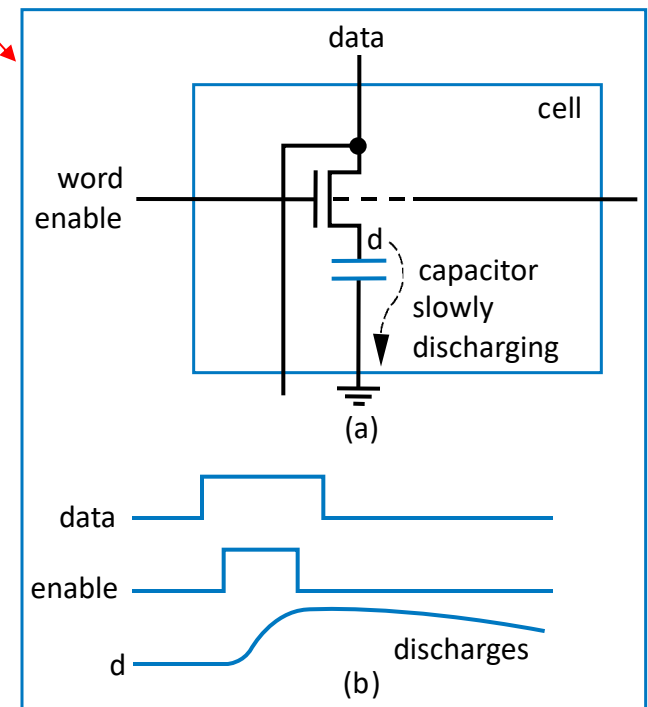
Source: wikipedia

Dynamic RAM (DRAM)



DRAM cell

- “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



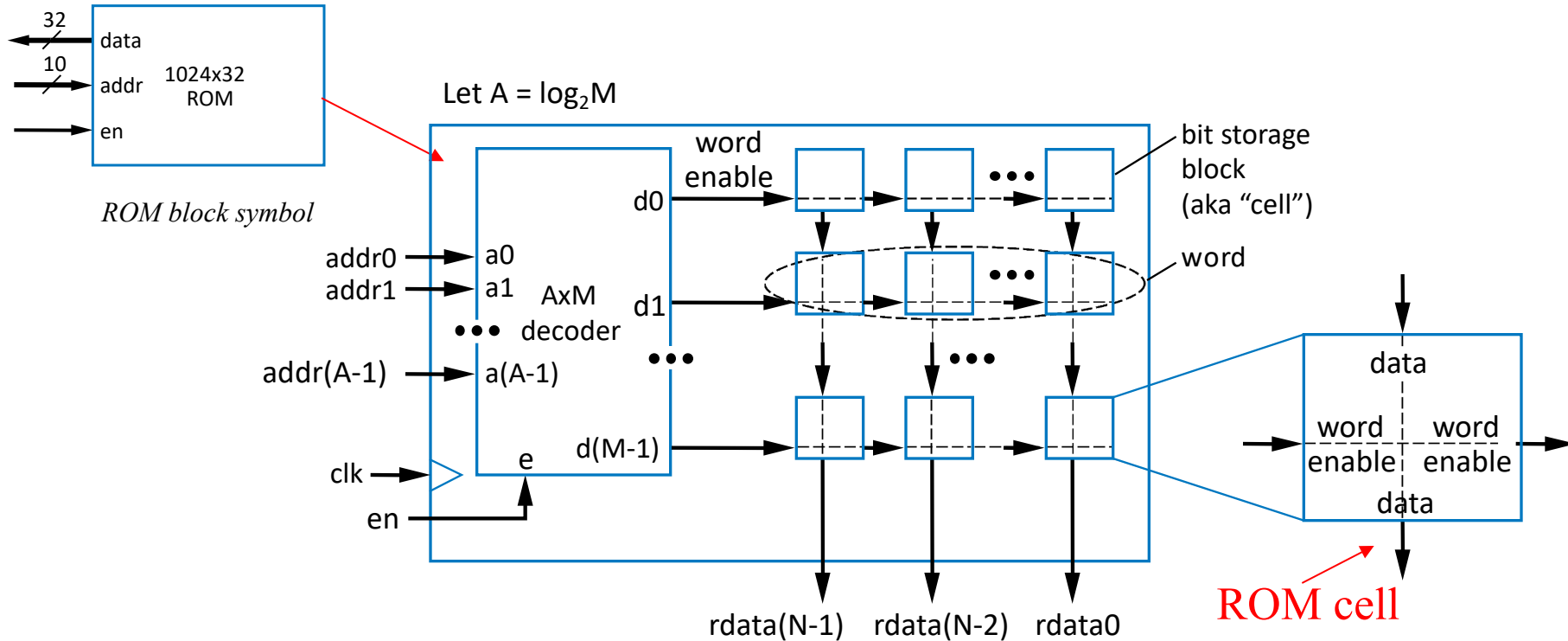
Comparing Memory Types

- 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)

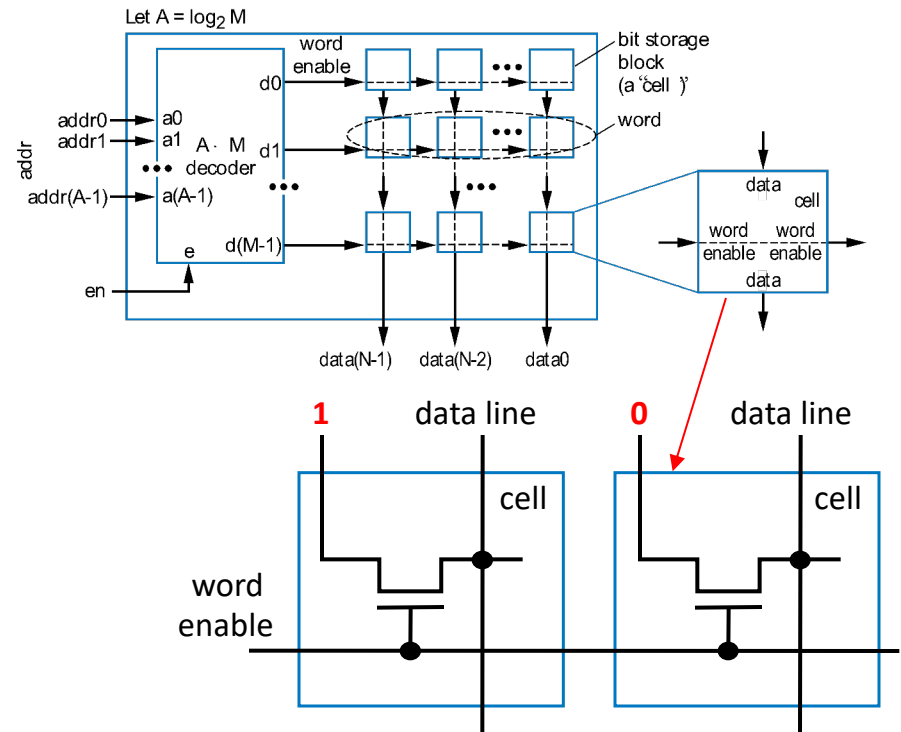
Read-Only Memory – ROM



- 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

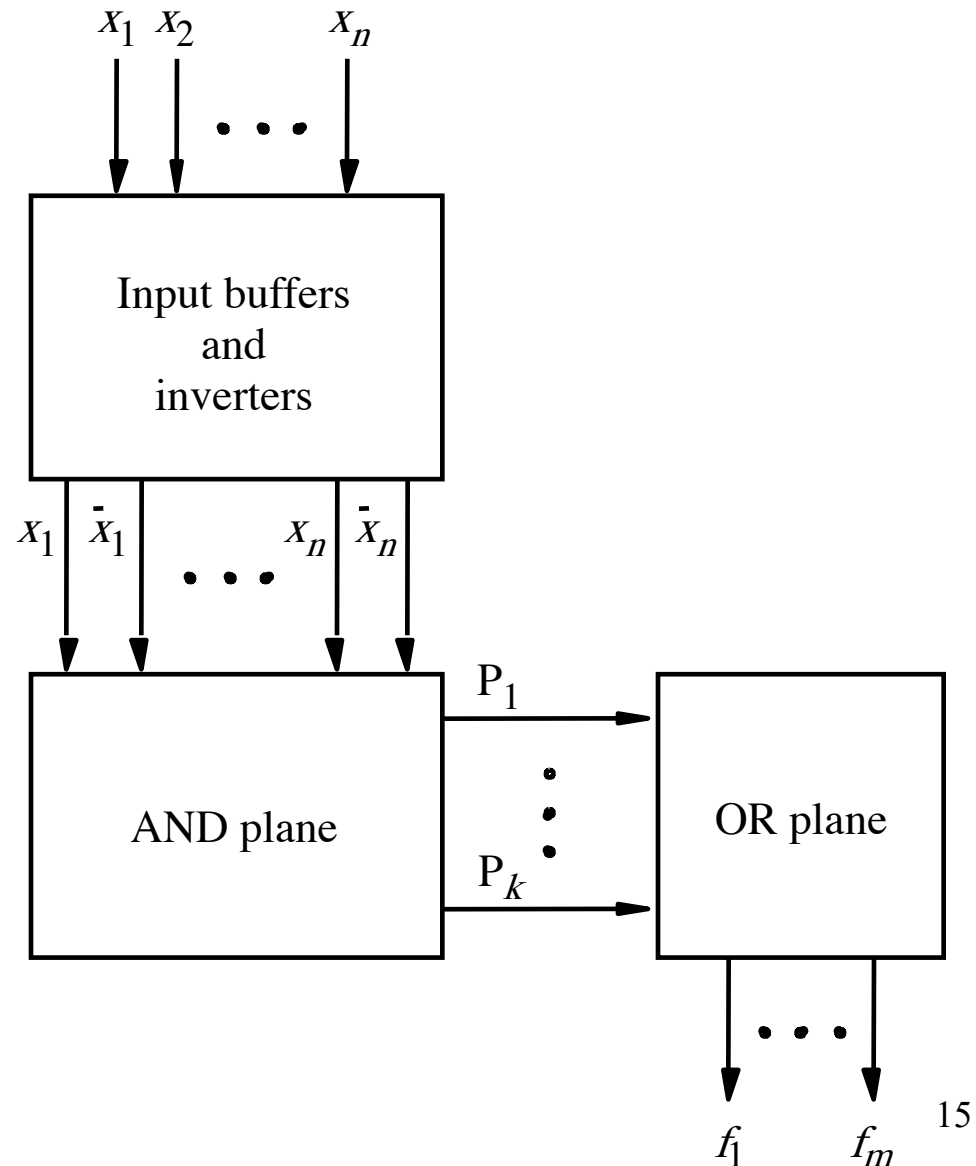


Programmable Logic Devices

- 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)

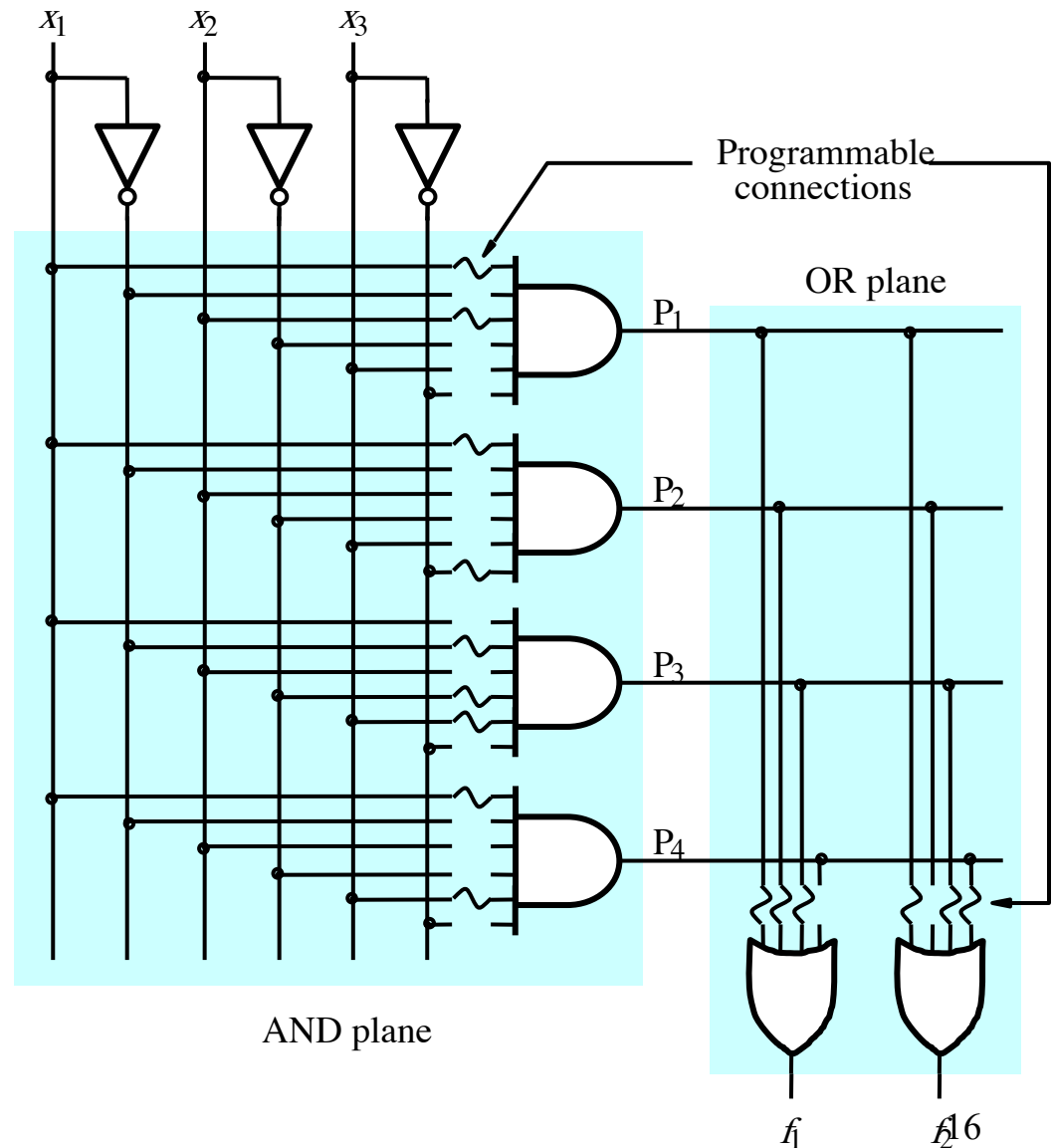
Programmable Logic Array (PLA)

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



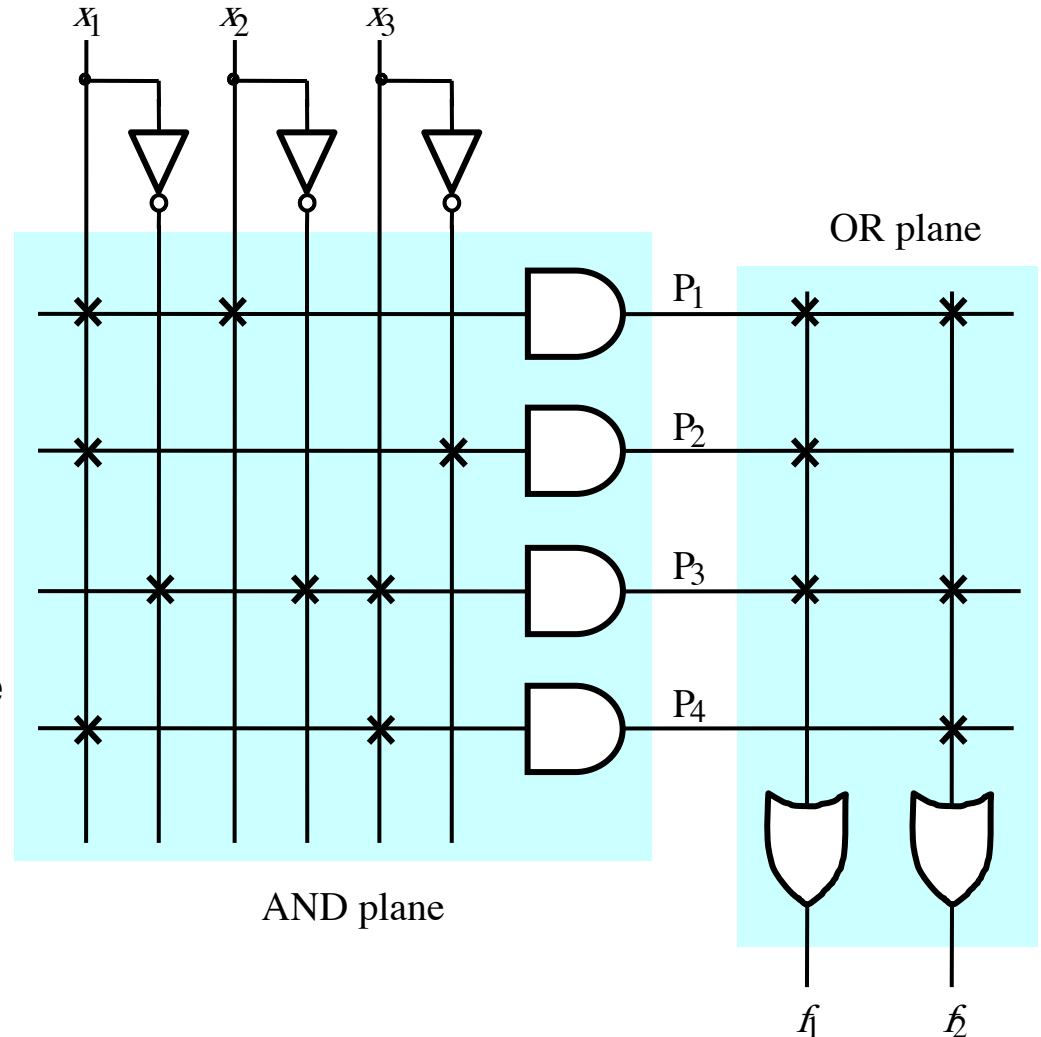
Programmable Logic Array (PLA)

- 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:
 - $P_1 = x_1x_2$
 - $P_2 = x_1x_3'$
 - $P_3 = x_1'x_2'x_3$
 - $P_4 = x_1x_3$
 - $F_1 = P_1 + P_2 + P_3$
 - $F_2 = P_1 + P_3 + P_4$



Programmable Logic Array (PLA)

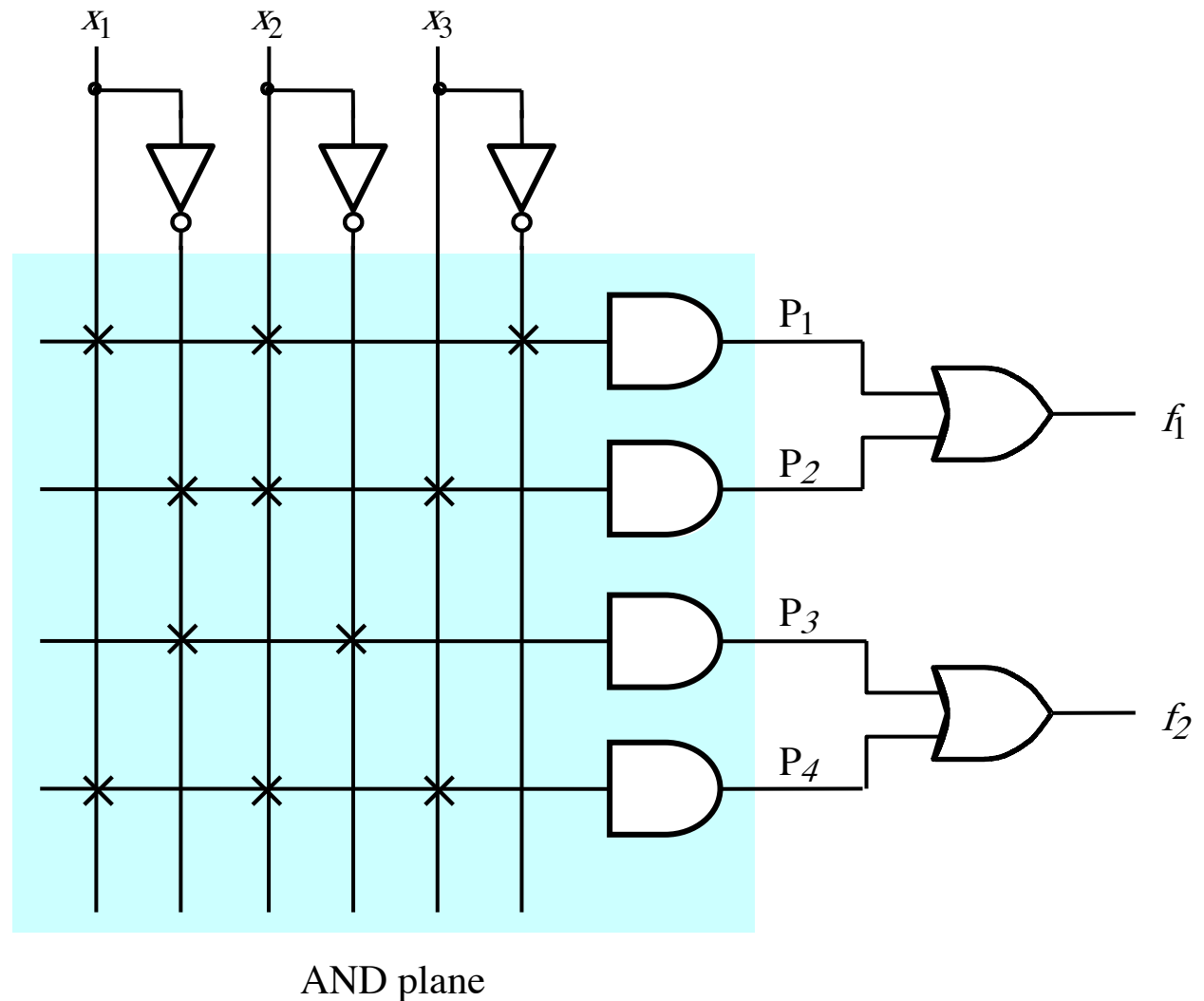
- Each AND gate has $2 \times N$ inputs
 - N , number of primary inputs
- Each OR gate has M inputs
 - M , number of and gates
- Problem:** size of the inputs
- Commercially available PLAs typically have:
 - 16 inputs
 - 32 AND gates
 - 8 OR gates
- Connections replaced by single lines, “x” indicates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



Programmable Array Logic (PAL)

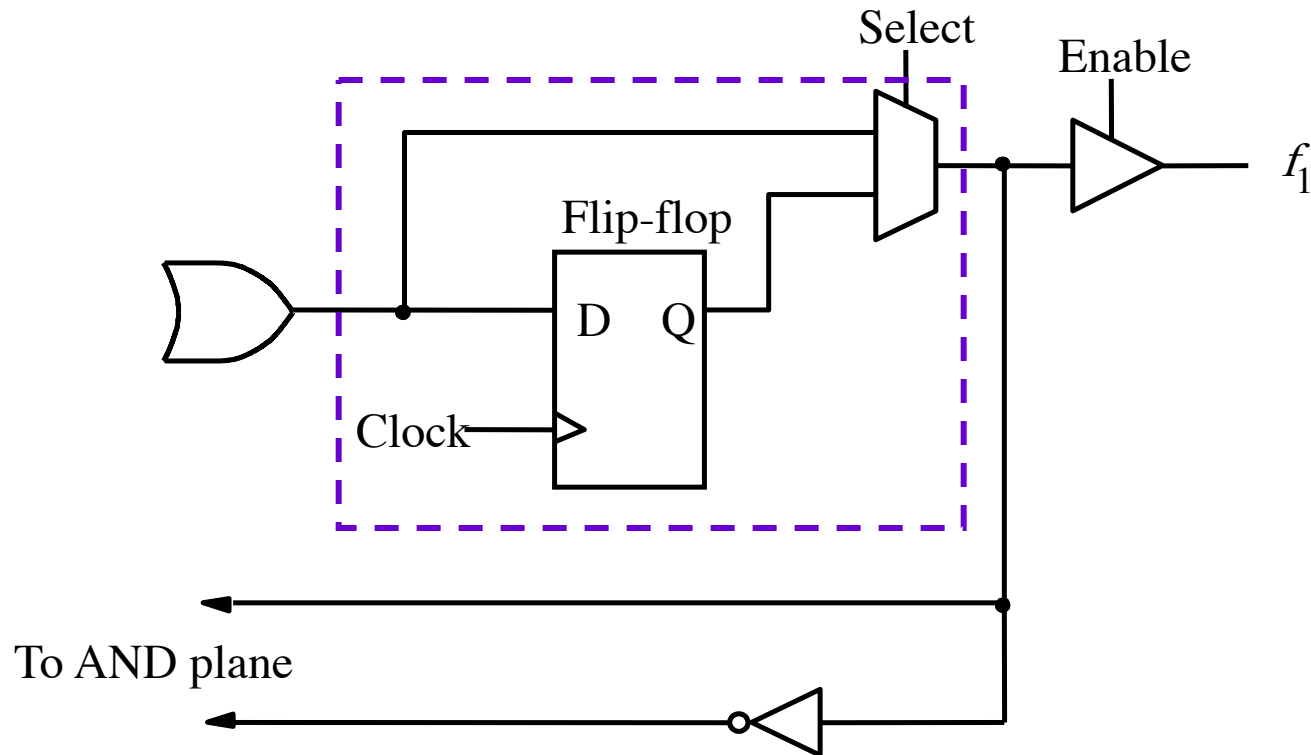
- Drawbacks of PLA
 - Hard to fabricate correctly due to the programmable connections
 - Special implementation of the programmable connections reduce the speed of circuits in PLA
- Solution: fix the OR plane – PAL
 - Less expensive
 - Better performance
 - Became popular in practical applications

Programmable Array Logic (PAL)



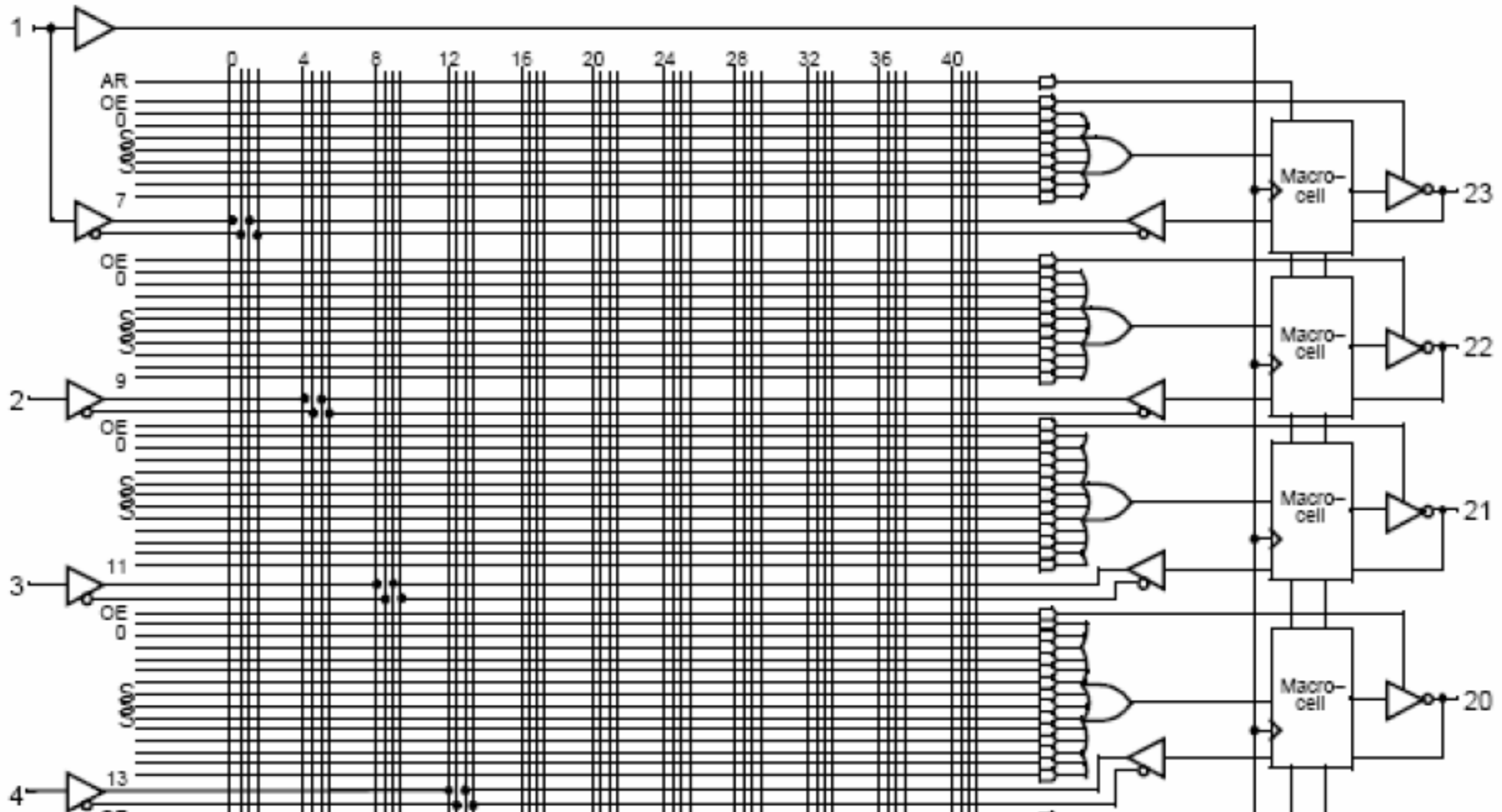
Programmable Array Logic (PAL)

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



PAL Example

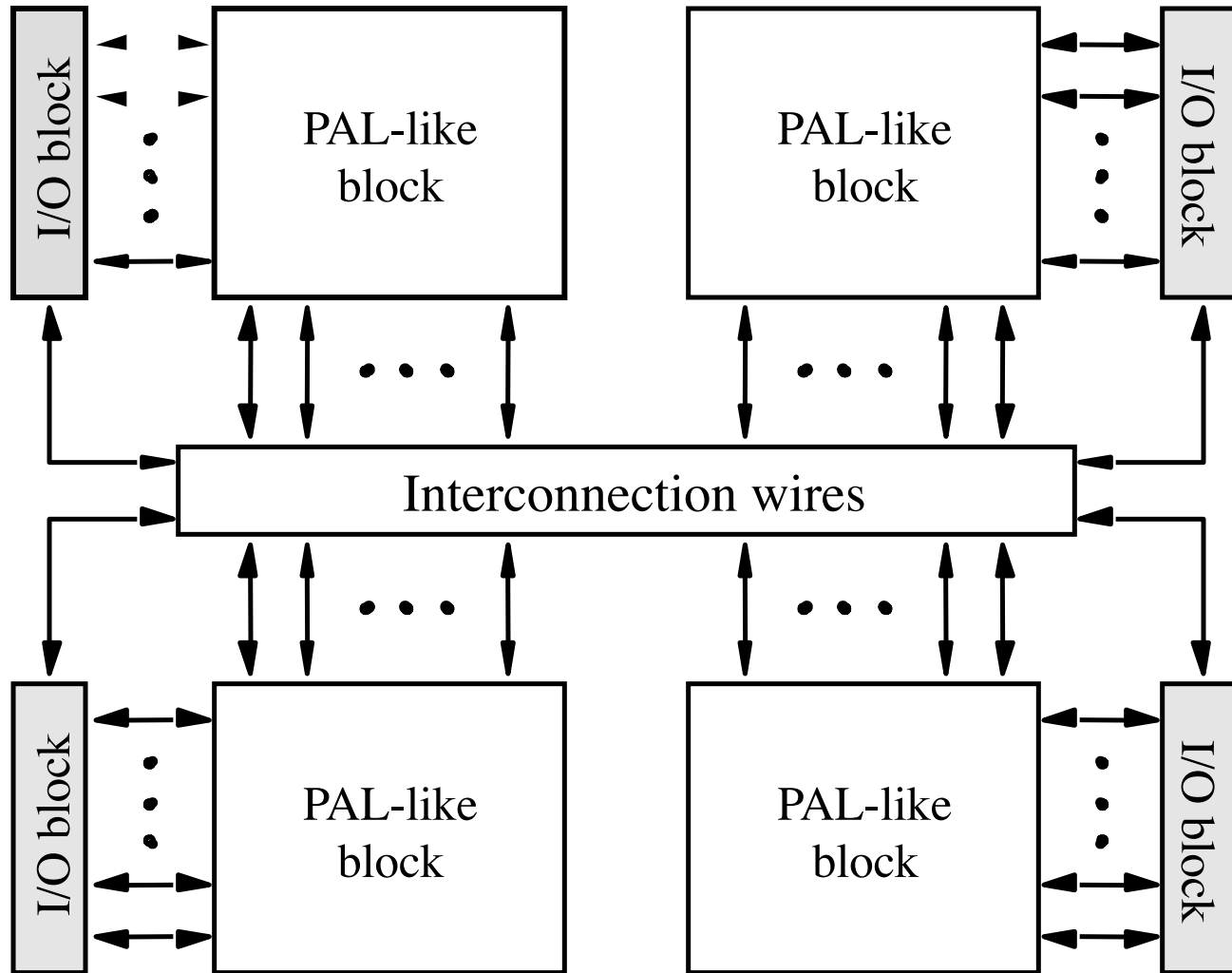
- Compensate for the reduced flexibility
 - Various numbers of inputs to the OR gates

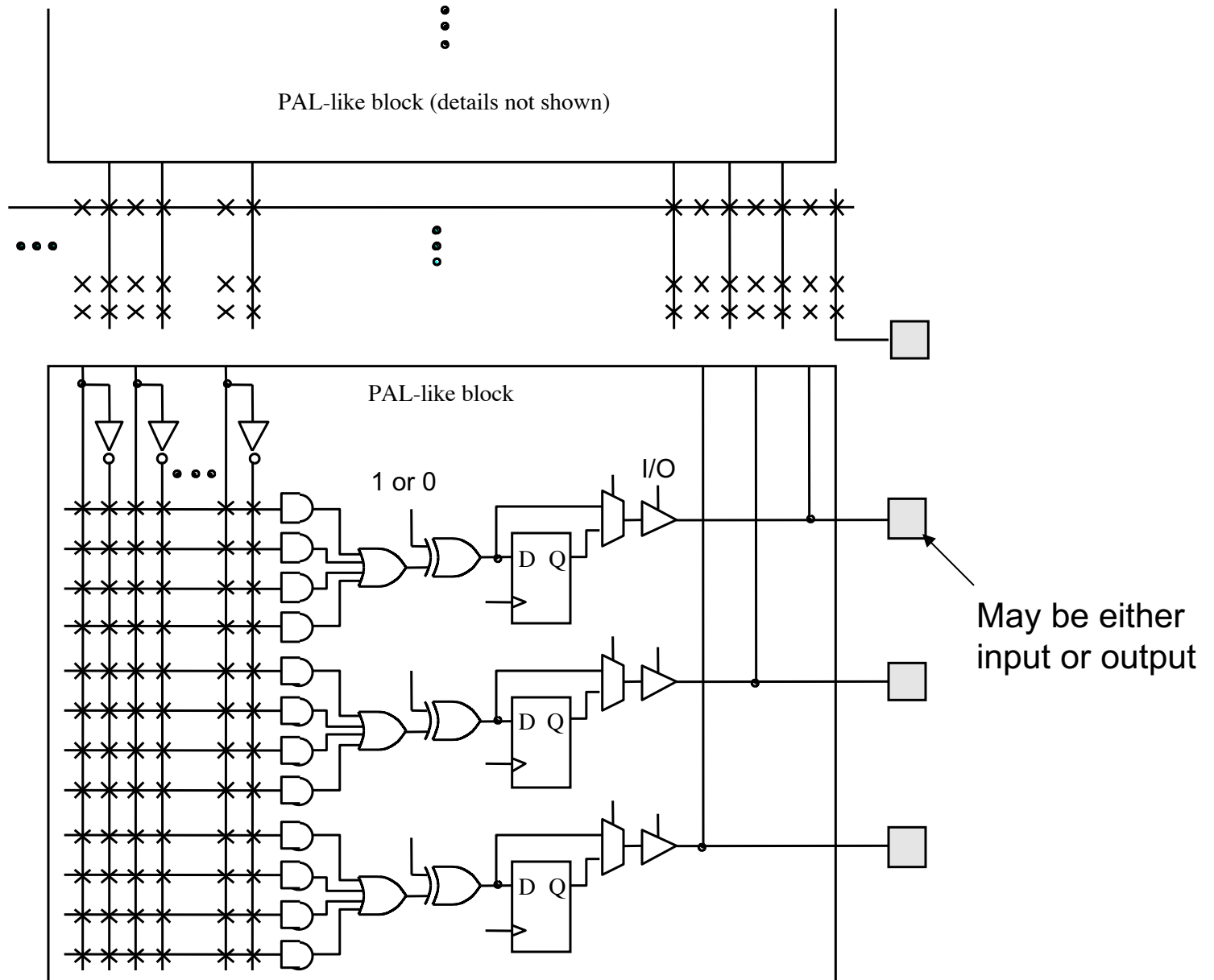


Complex Programmable Logic Devices (CPLD)

- 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





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

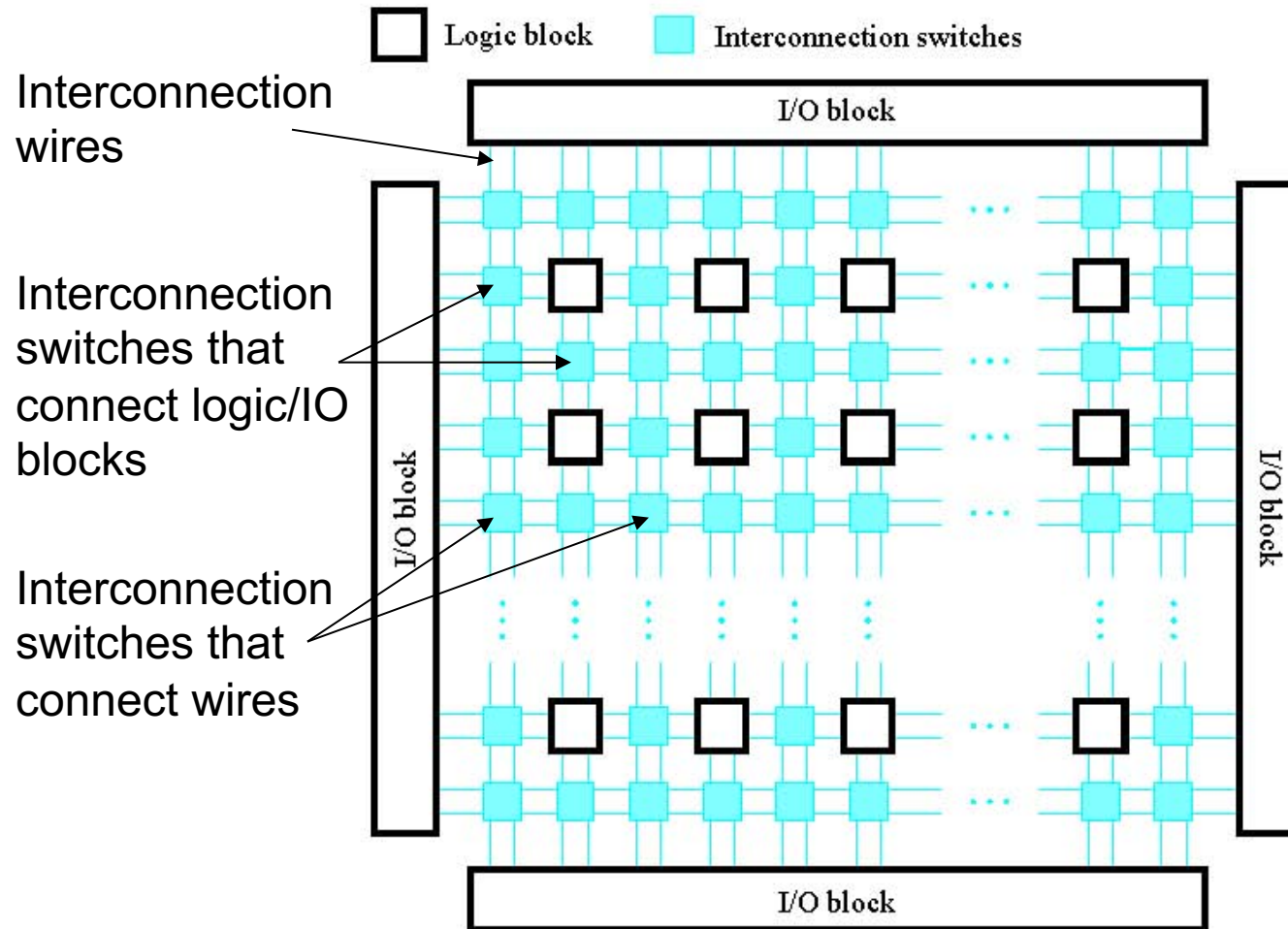
- Reprogrammable
 - SRAM-based FPGA
 - Volatile, often the best choice for prototyping and development
 - Supports in-system-programming (ISP)
 - What we used in the labs
 - EEPROM-based (Flash-based) FPGA
- One-Time Programmable (OTP)
 - Anti-Fuse-based FPGA
 - EPROM-based FPGA

Internal Structure

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

FPGA Architecture

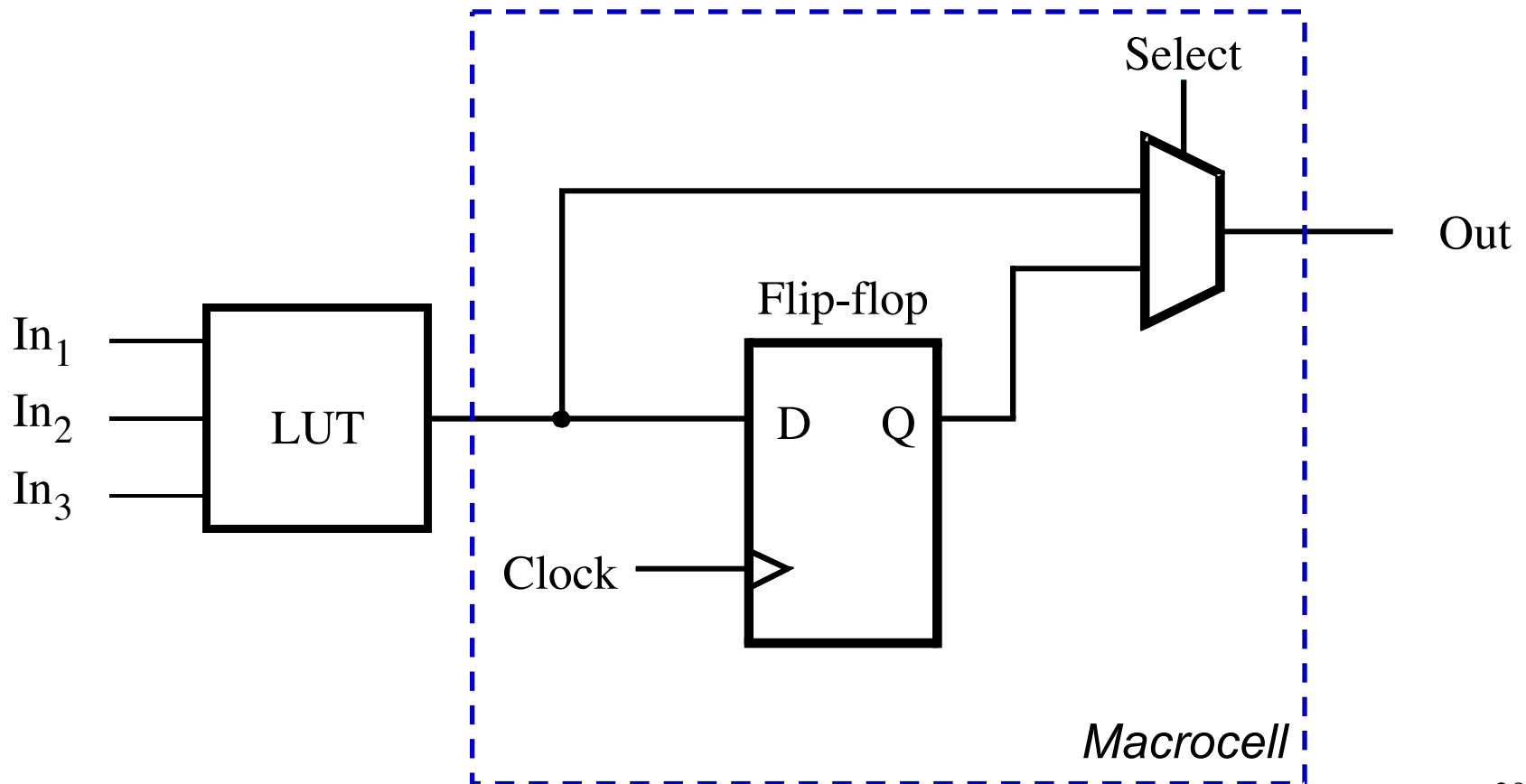
- Typical FPGA architecture



Courtesy of Xilinx Inc.

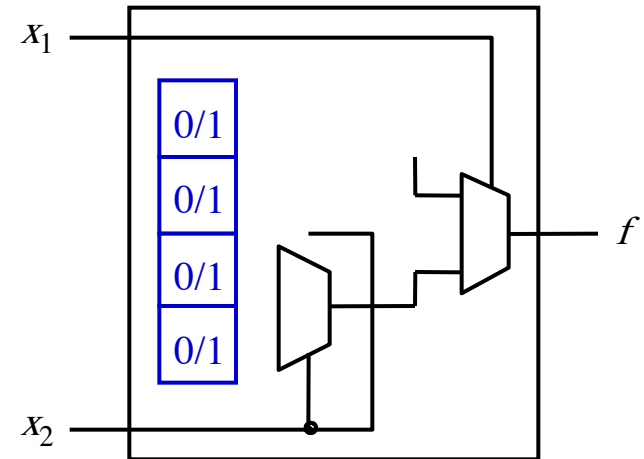
Programmable (Configurable) Logic Block

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



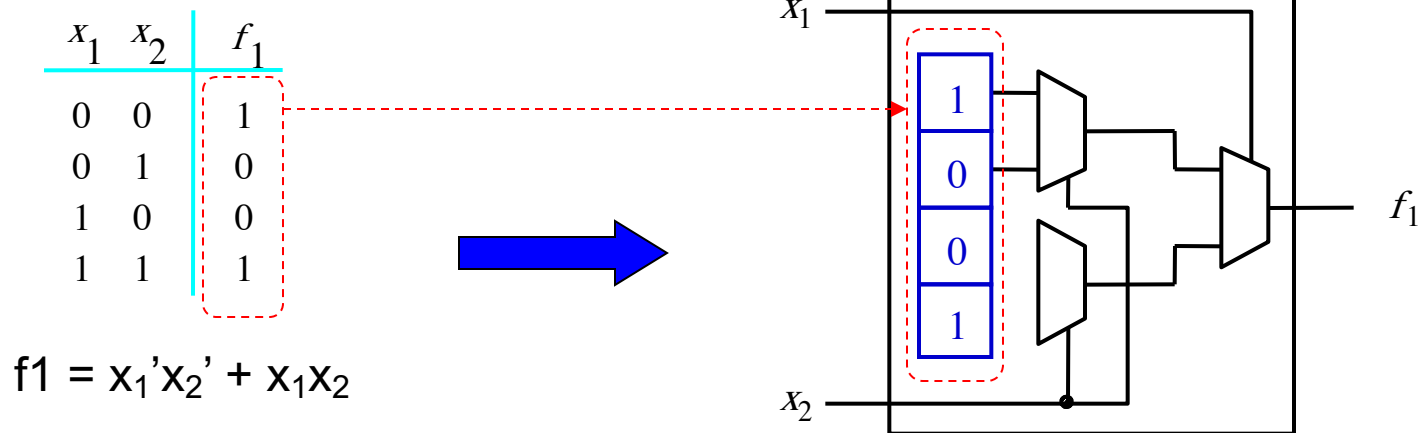
Look Up Table (LUT)

- 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



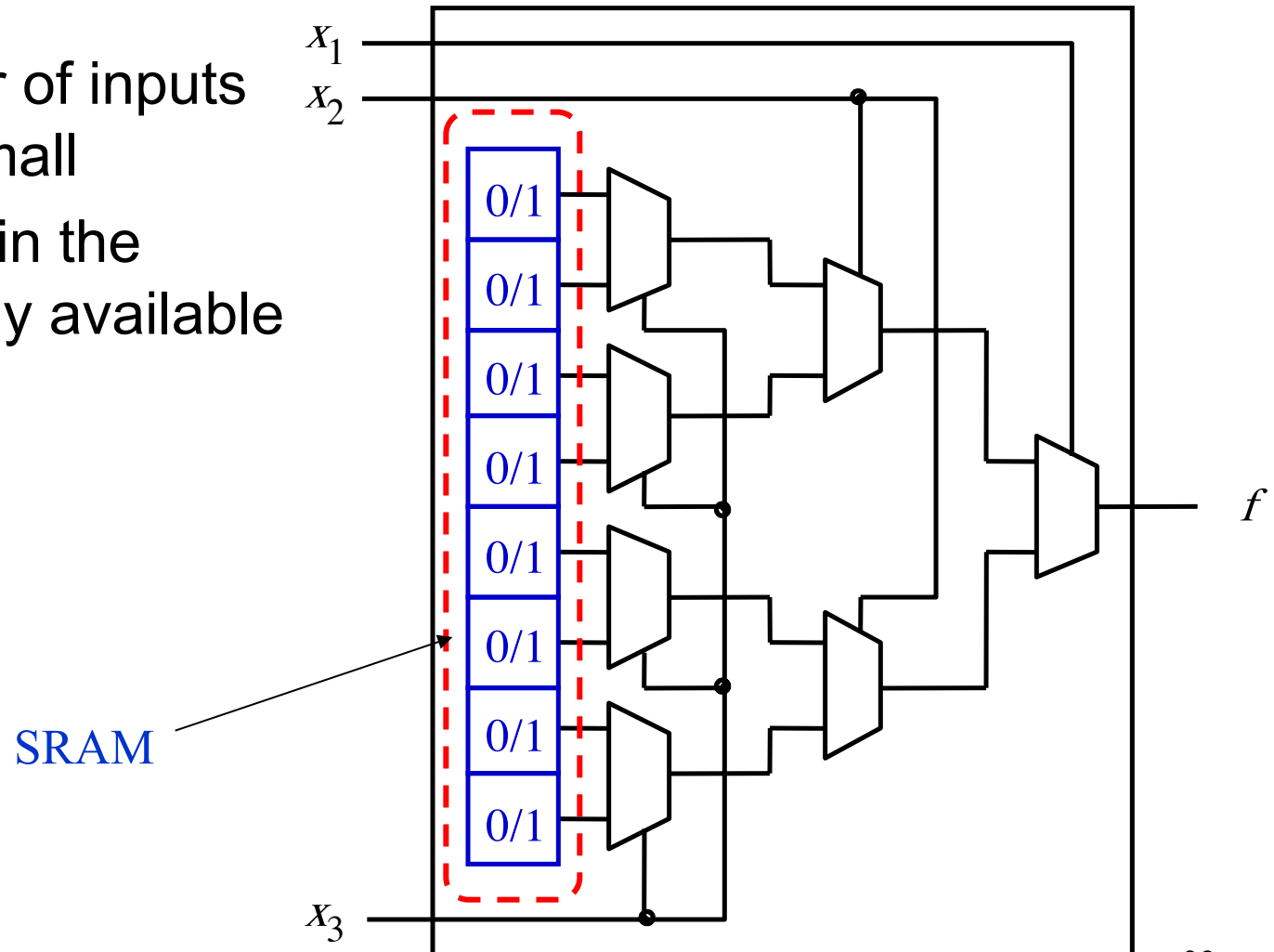
LUT Example

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



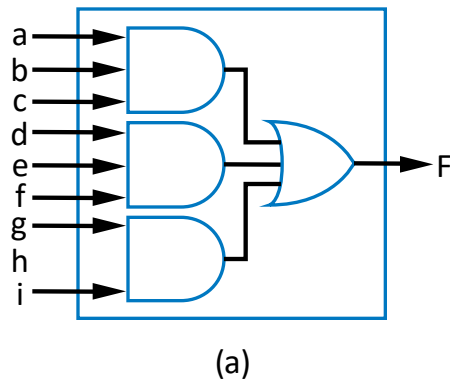
3-input LUT

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

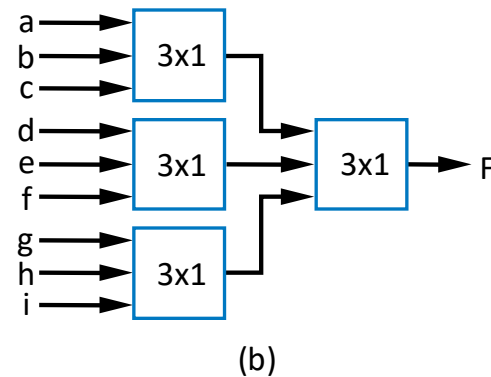


FPGA Internals: Lookup Tables (LUTs)

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



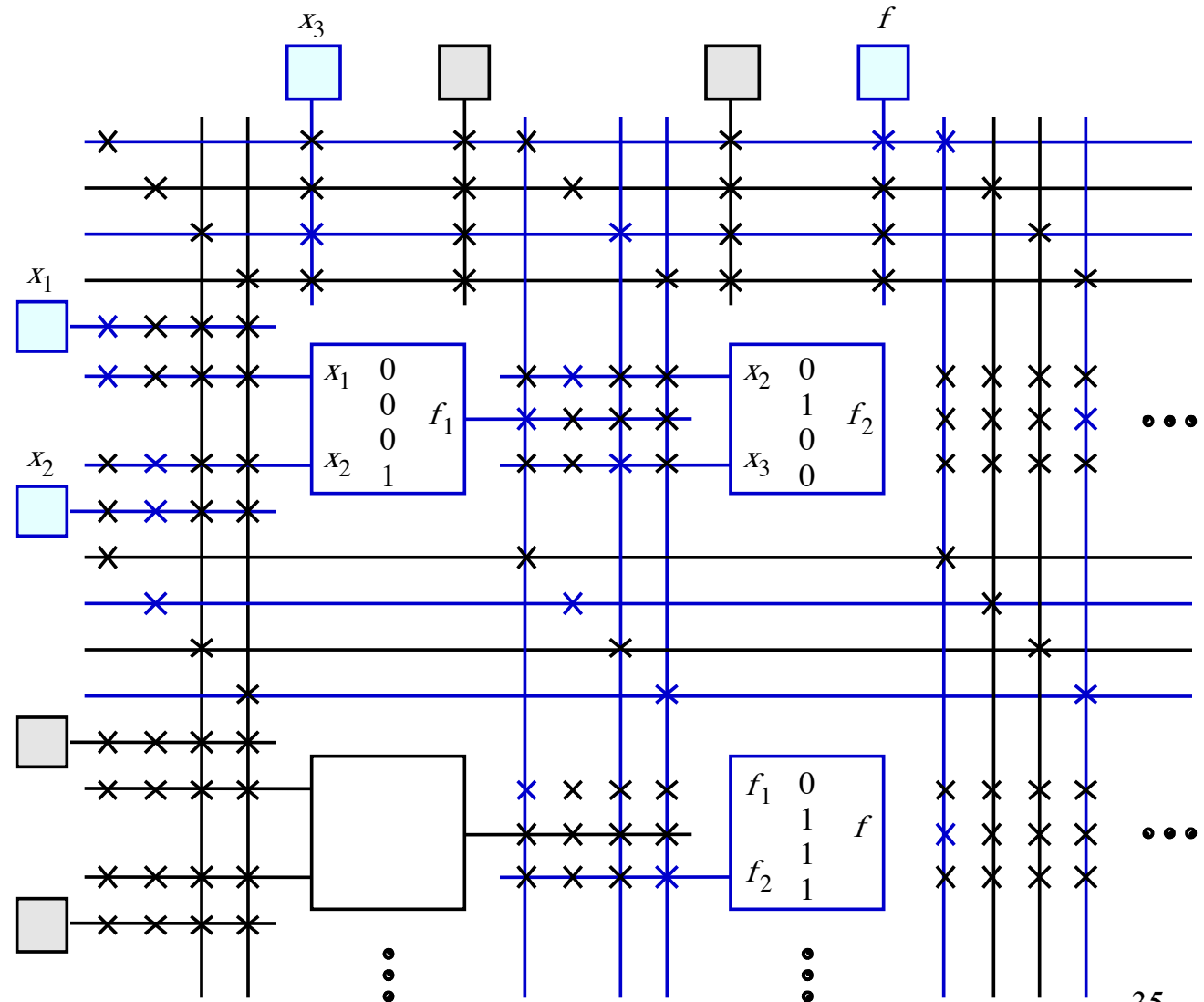
Original 9-input circuit



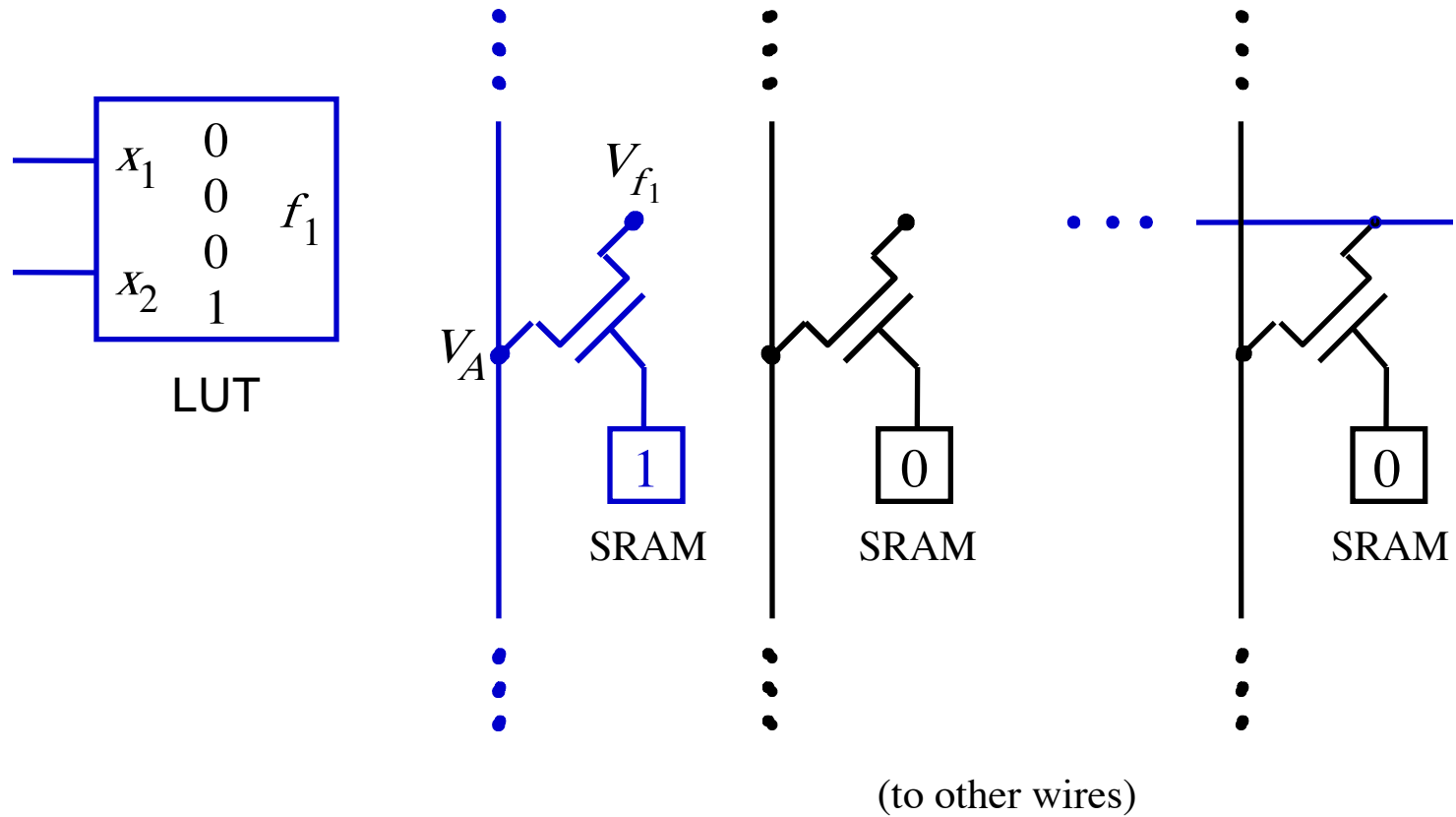
Implemented with
3x1 LUTs

FPGA Configuration Example

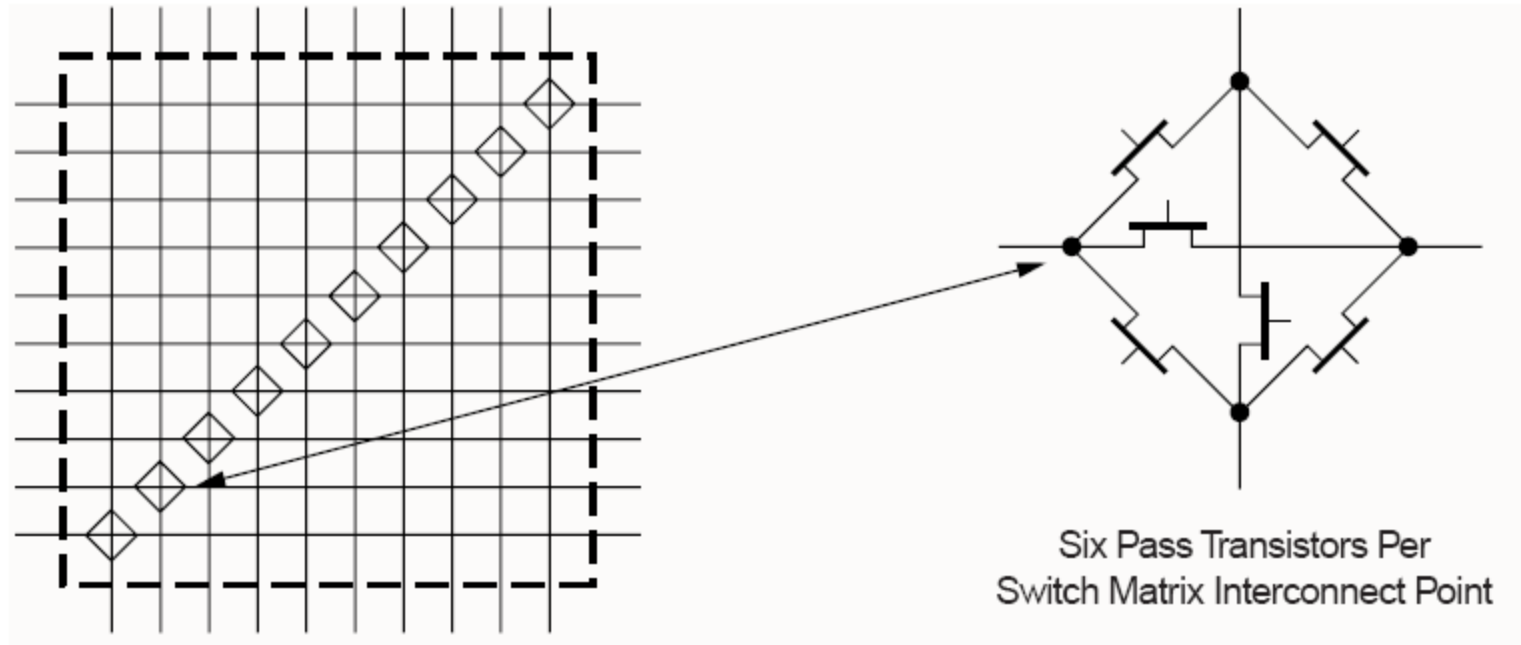
- Blue x indicates a connection
- $f_1 = x_1 x_2$
- $f_2 = x_2' x_3$
- $f = f_1 + f_2$



Programmable Switch

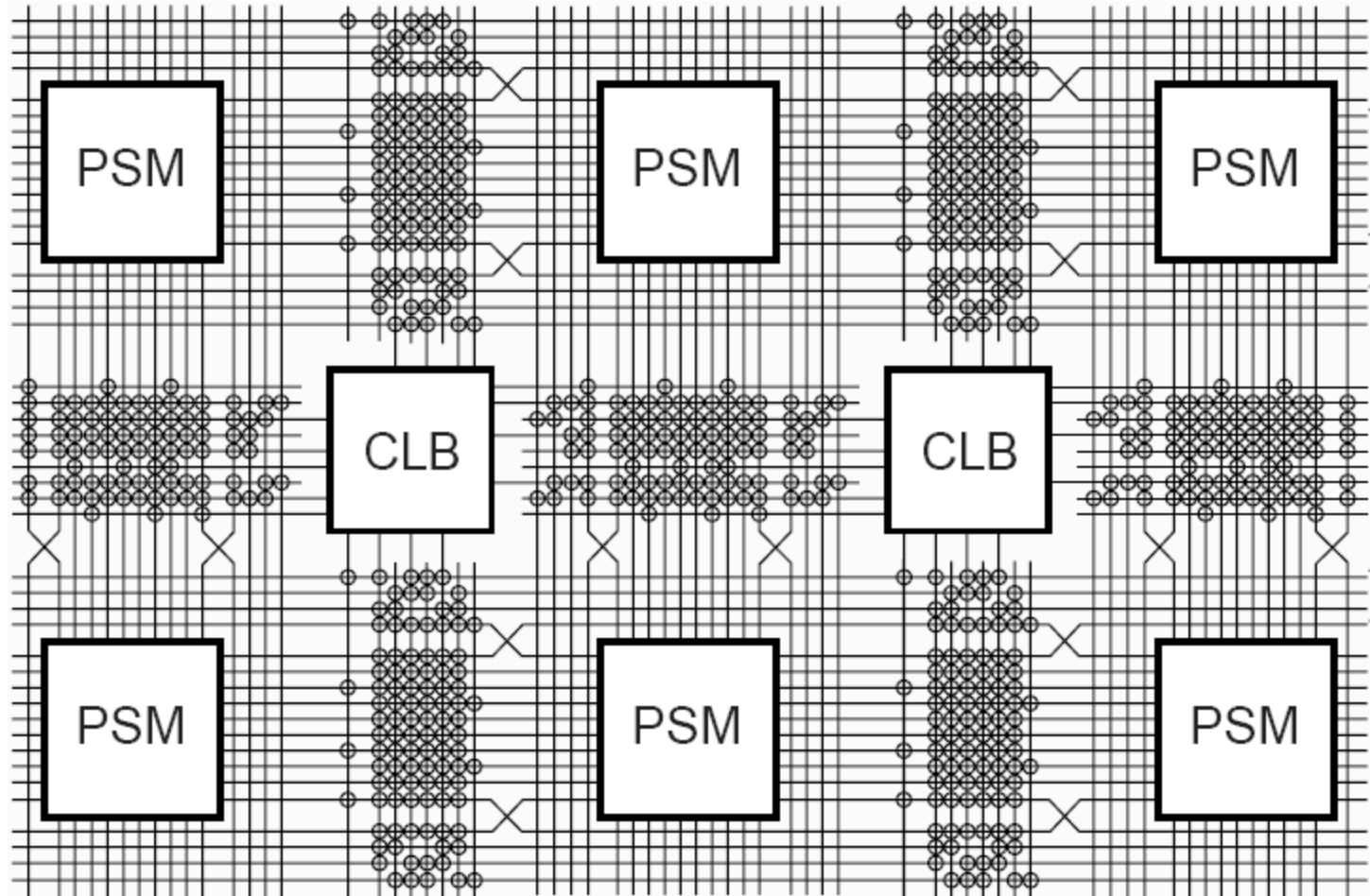


Xilinx FPGA PSM



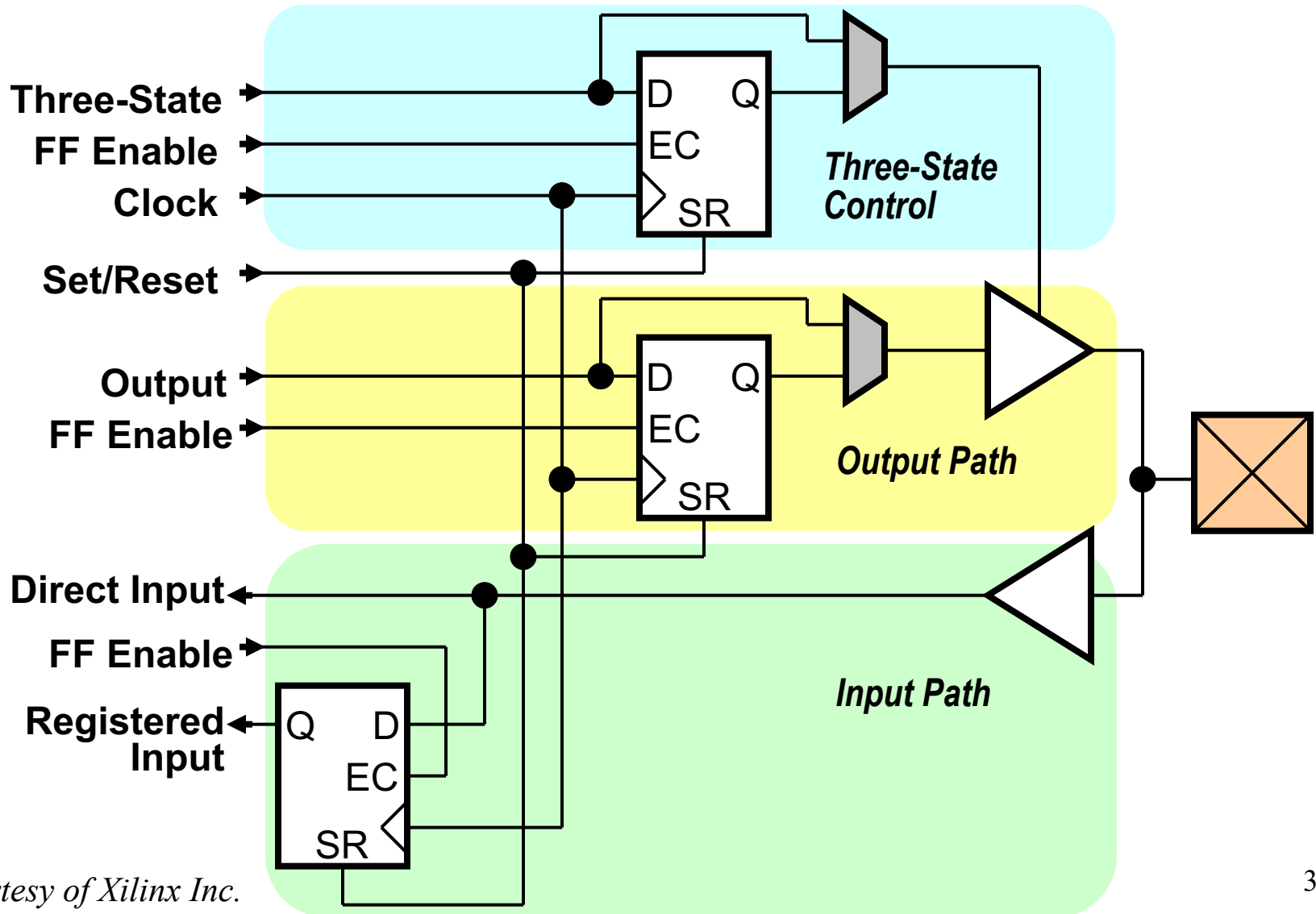
Programmable Switch Matrix (PSM)

Xilinx FPGA Switches and Wires



Courtesy of Xilinx Inc.

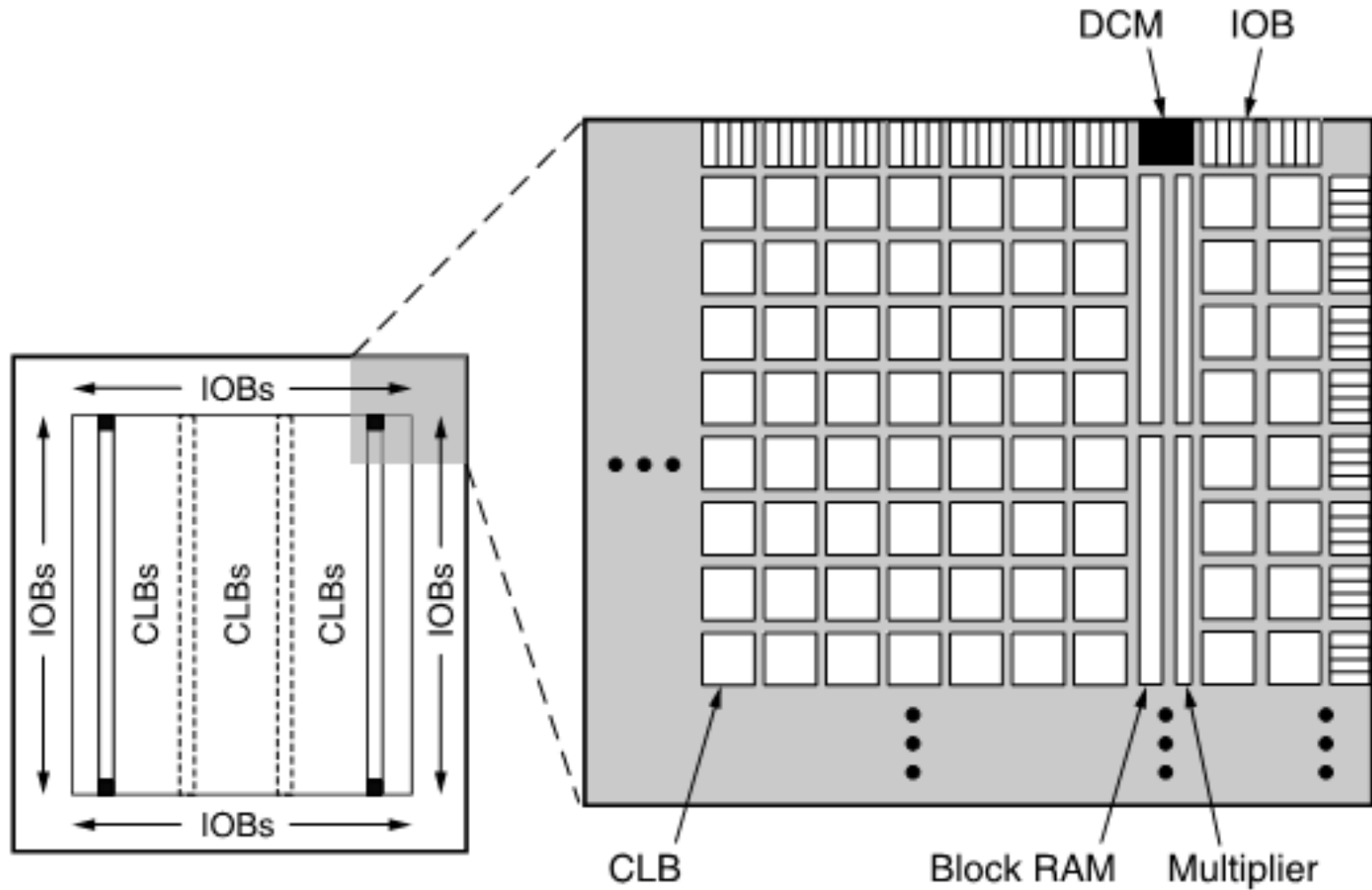
Basic I/O Block Structure



Advanced FPGA Features

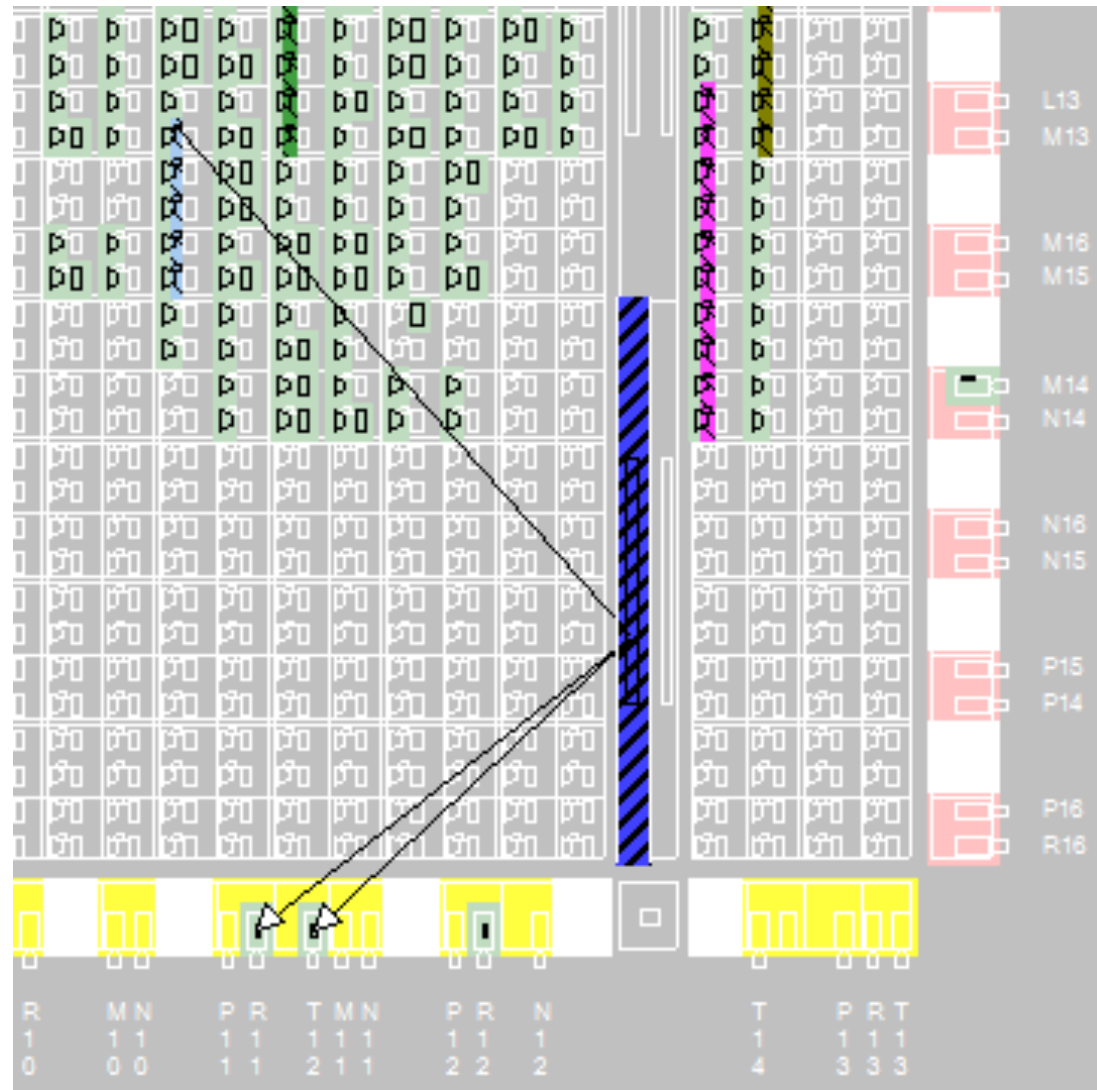
- Dozens of millions of equivalent logic gates
- Enhanced clock features
- Flexible embedded memory blocks
- Intellectual property (IP) cores
- Embedded processors (hard and soft)
- Digital signal processing (DSP) blocks, tools, design flows (specific FPGA vendors)
- Dedicated hardware multipliers
- 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