Logic and Computer Design Fundamentals

Chapter 3 – Combinational Logic Design

Part 2 – Programmable Implementation Technologies

Charles Kime & Thomas Kaminski

Overview

- **Part 1 - Implementation Technology and Logic Design**
  - **Design Concepts and Automation**
    - **Fundamental concepts of design and computer-aided design techniques**
  - **The Design Space**
    - **Technology parameters for gates, positive and negative logic and design tradeoffs**
  - **Design Procedure**
    - **The major design steps: specification, formulation, optimization, technology mapping, and verification**
  - **Technology Mapping**
    - **Mapping from AND, OR, and NOT to other gate types**
  - **Verification**
    - **Does the designed circuit meet the specifications?**
- **Part 2 - Programmable Implementation Technologies**
  - **Read-Only Memories, Programmable Logic Arrays, Programmable Array Logic**
    - **Technology mapping to programmable logic devices**

Overview

- **Why programmable logic?**
- **Programmable logic technologies**
- **Read-Only Memory (ROM)**
- **Programmable Logic Array (PLA)**
- **Programmable Array Logic (PAL)**
- **VLSI Programmable Logic Devices -**
  covered in VLSI Programmable Logic Devices reading supplement

Why Programmable Logic?

- **Facts:**
  - **It is most economical to produce an IC in large volumes**
  - **Many designs required only small volumes of ICs**
- **Need an IC that can be:**
  - **Produced in large volumes**
  - **Handle many designs required in small volumes**
- **A programmable logic part can be:**
  - **made in large volumes**
  - **programmed to implement large numbers of different low-volume designs**

Programmable Logic - Additional Advantages

- **Many programmable logic devices are *field-programmable*, i. e., can be programmed outside of the manufacturing environment**
- **Most programmable logic devices are *erasable* and *reprogrammable*.**
  - **Allows "updating" a device or correction of errors**
  - **Allows reuse the device for a different design - the ultimate in re-usability!**
  - **Ideal for course laboratories**
- **Programmable logic devices can be used to prototype design that will be implemented for sale in regular ICs.**
  - **Complete Intel Pentium designs were actually prototype with specialized systems based on large numbers of VLSI programmable devices!**

Programming Technologies

- **Programming technologies are used to:**
  - **Control connections**
  - **Build lookup tables**
  - **Control transistor switching**
- **The technologies**
  - **Control connections**
    - **Mask programming**
    - **Fuse**
    - **Antifuse**
    - **Single-bit storage element**

Programming Technologies

- The technologies (continued)
  - Build lookup tables
    - Storage elements (as in a memory)
  - Transistor Switching Control
    - Stored charge on a floating transistor gate
      - Erasable
      - Electrically erasable
      - Flash (as in Flash Memory)
    - Storage elements (as in a memory)

Technology Characteristics

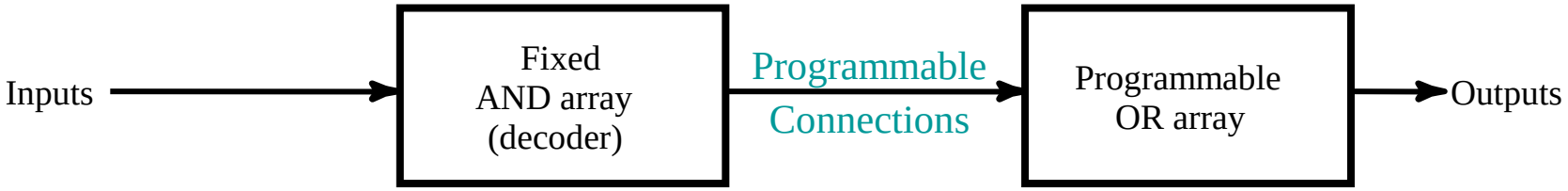
- **Permanent - Cannot be erased and reprogrammed**
  - **Mask programming**
  - **Fuse**
  - **Antifuse**
- **Reprogrammable**
  - **Volatile - Programming lost if chip power lost**
    - **Single-bit storage element**
  - **Non-Volatile**
    - **Erasable**
    - **Electrically erasable**
    - **Flash (as in Flash Memory)**

Programmable Configurations

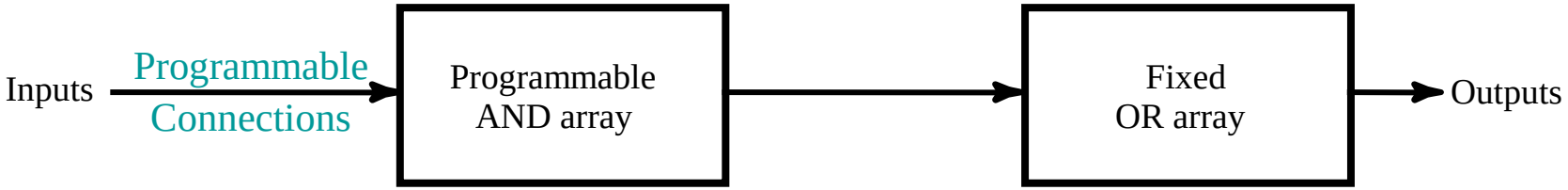
- ***Read Only Memory* (*ROM*) - a fixed array of AND gates and a programmable array of OR gates**
- ***Programmable Array Logic* (*PAL*)® - a programmable array of AND gates feeding a fixed array of OR gates.**
- ***Programmable Logic Array* (*PLA*) - a programmable array of AND gates feeding a programmable array of OR gates.**
- ***Complex Programmable Logic Device* (*CPLD*) /*Field- Programmable Gate Array* (*FPGA*) - complex enough to be called "architectures" - See VLSI Programmable Logic Devices reading supplement**

PAL is a registered trademark of Lattice Semiconductor Corp.

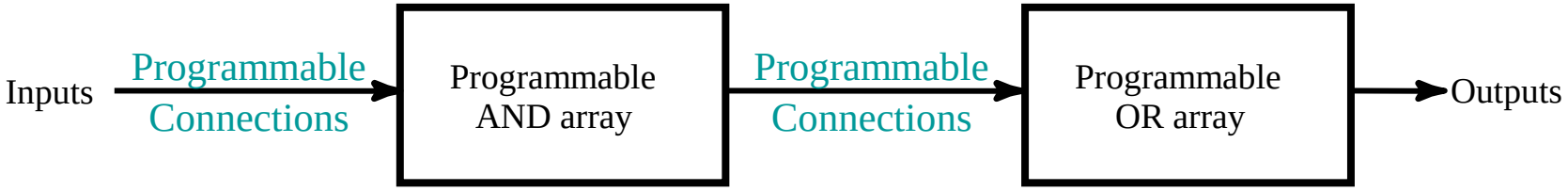
ROM, PAL and PLA Configurations



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL) device



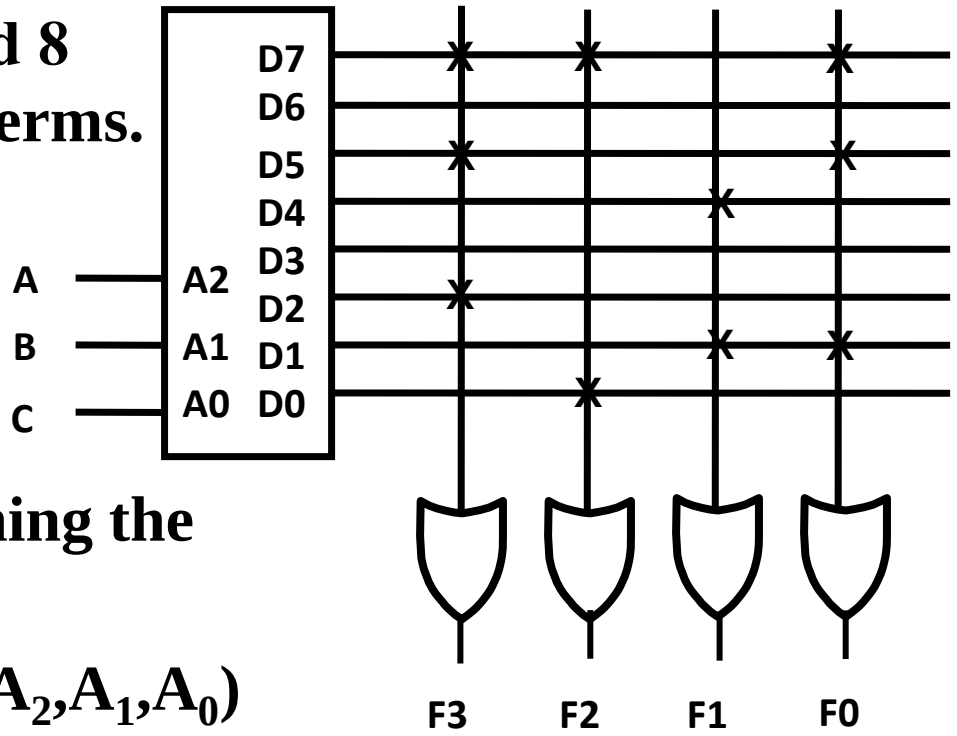
(c) Programmable logic array (PLA) device

Read Only Memory

- **Read Only Memories (ROM) or Programmable Read Only Memories (PROM) have:**
  - **N input lines,**
  - **M output lines, and**
  - **2^N decoded minterms.**
- **Fixed AND array with 2^N outputs implementing all N-literal minterms.**
- **Programmable OR Array with M outputs lines to form up to M sum of minterm expressions.**
- **A program for a ROM or PROM is simply a multiple-output truth table**
  - **If a 1 entry, a connection is made to the corresponding minterm for the corresponding output**
  - **If a 0, no connection is made**
- **Can be viewed as a *memory* with the inputs as *addresses* of *data* (output values), hence ROM or PROM names!**

Read Only Memory Example

- Example: A 8 X 4 ROM (N = 3 input lines, M= 4 output lines)
- The fixed "AND" array is a "decoder" with 3 inputs and 8 outputs implementing minterms.
- The programmable "OR" array uses a single line to represent all inputs to an OR gate. An "X" in the array corresponds to attaching the minterm to the OR



- Read Example: For input (A_2,A_1,A_0) = 011, output is (F_3,F_2,F_1,F_0) = 0011.
- What are functions F_3, F_2 , F_1 and F_0 in terms of (A_2, A_1, A_0)?

Programmable Array Logic (PAL)

- **The PAL is the opposite of the ROM, having a programmable set of ANDs combined with fixed ORs.**
- **Disadvantage**
  - **ROM guaranteed to implement any M functions of N inputs. PAL may have too few inputs to the OR gates.**
- **Advantages**
  - **For given internal complexity, a PAL can have larger N and M**
  - **Some PALs have outputs that can be complemented, adding POS functions**
  - **No multilevel circuit implementations in ROM (without external connections from output to input). PAL has outputs from OR terms as internal inputs to all AND terms, making implementation of multi-level circuits easier.**

Programmable Array Logic Example

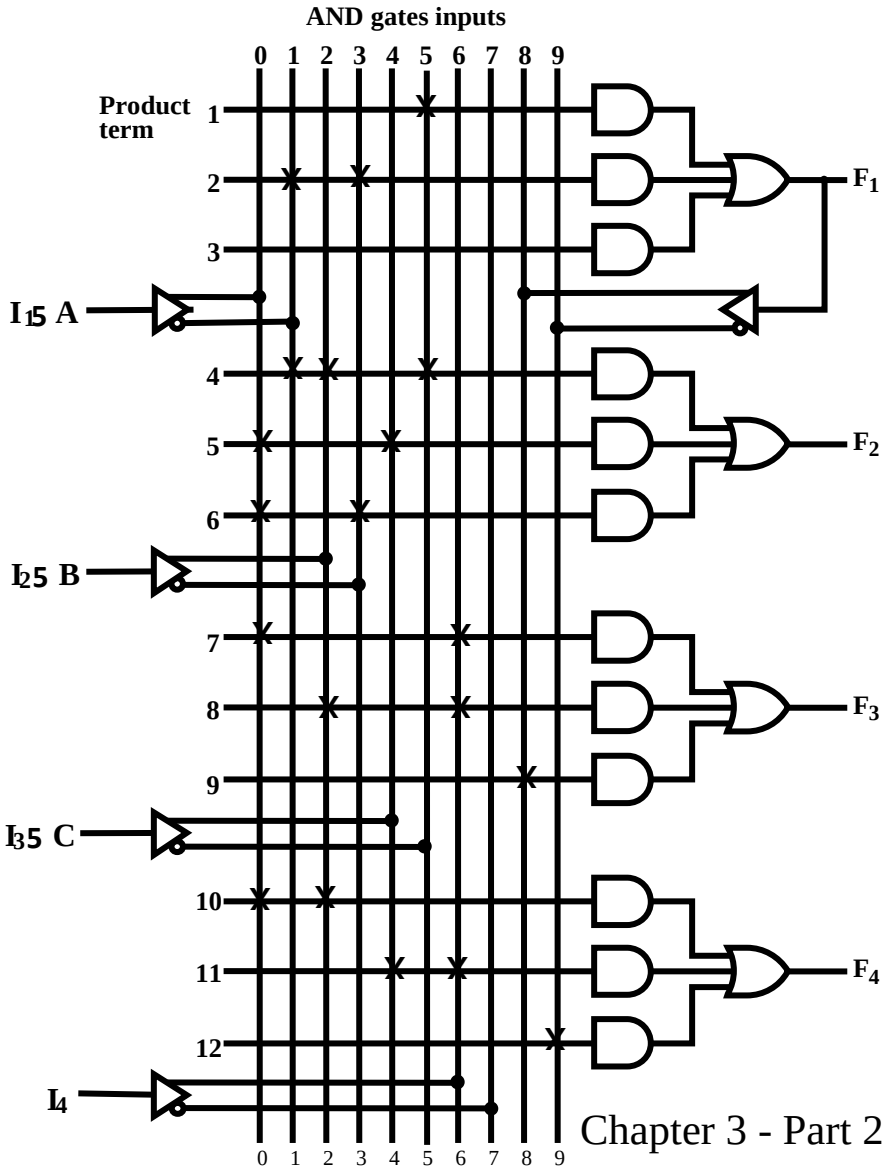
- **4-input, 3-output PAL with fixed, 3-input OR terms**
- **What are the equations for F1 through F4?**

  F1 = $\overline{A}\,\overline{B} + \overline{C}$

  F2 = $\overline{A}B\,\overline{C}$ + AC + AB

  F3 =

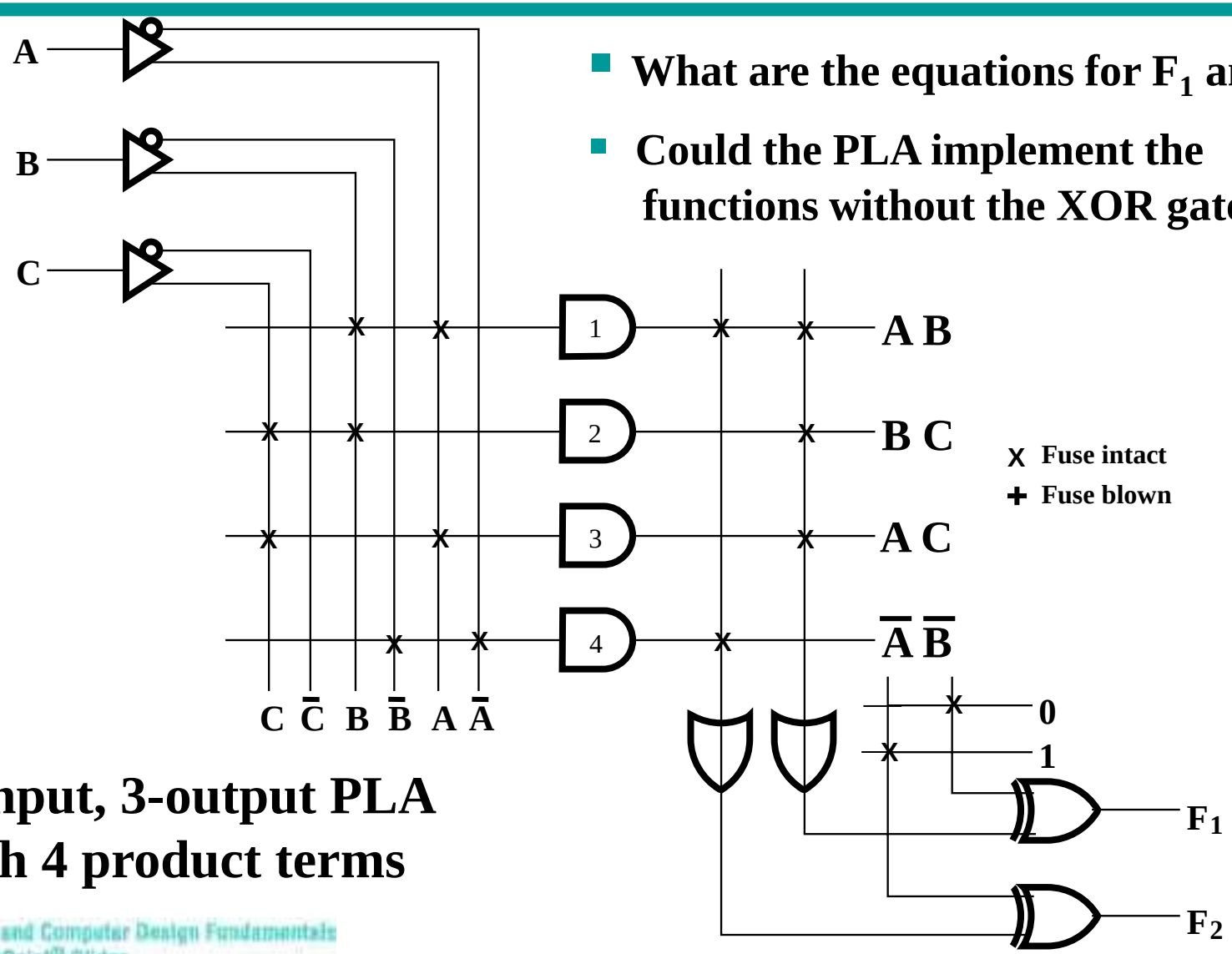
  F4 =



Programmable Logic Array (PLA)

- **Compared to a ROM and a PAL, a PLA is the most flexible having a programmable set of ANDs combined with a programmable set of ORs.**
- **Advantages**
  - **A PLA can have large N and M permitting implementation of equations that are impractical for a ROM (because of the number of inputs, N, required**
  - **A PLA has all of its product terms connectable to all outputs, overcoming the problem of the limited inputs to the PAL Ors**
  - **Some PLAs have outputs that can be complemented, adding POS functions**
- **Disadvantage**
  - **Often, the product term count limits the application of a PLA. Two-level multiple-output optimization reduces the number of product terms in an implementation, helping to fit it into a PLA.**

Programmable Logic Array Example



- What are the equations for F_1 and F_2?
- Could the PLA implement the functions without the XOR gates?

- 3-input, 3-output PLA with 4 product terms

Terms of Use

-
- The following terms of use apply in addition to the standard Pearson Education Legal Notice.
- Permission is given to incorporate these materials into classroom presentations and handouts only to instructors adopting Logic and Computer Design Fundamentals as the course text.
- Permission is granted to the instructors adopting the book to post these materials on a protected website or protected ftp site in original or modified form. All other website or ftp postings, including those offering the materials for a fee, are prohibited.
- You may not remove or in any way alter this Terms of Use notice or any trademark, copyright, or other proprietary notice, including the copyright watermark on each slide.
- Return to Title Page