COE 202: Digital Logic Design
Memory and Programmable Logic Devices

Dr. Ahmad Almulhem
Email: ahmadsm AT kfupm
Phone: 860-7554
Office: 22-324
Objectives

- Memory
- Programmable Logic Devices (PLD)
Memory

- **Memory**: A collection of cells capable of storing binary information (1s or 0s) – in addition to electronic circuit for storing (writing) and retrieving (reading) information.

- n data lines (input/output)
- k address lines
- $2^k$ words (data unit)
- Read/Write Control
- Memory size = $2^k \times n$

Ahmad Almulhem, KFUPM 2010
Memory (cont.)

Two Types of Memory:

• Random Access Memory (RAM):
  • Write/Read operations
  • **Volatile:** Data is lost when power is turned off

• Read Only Memory (ROM):
  • Read operation (no write)
  • **Non-Volatile:** Data is permanent.
  • PROM is programmable (allow special write)
Programmable Logic Devices

- **Programmable Logic Device** (PLD) is an integrated circuit with internal logic gates and/or connections that can in some way be changed by a programming process
  - Examples:
    - PROM
    - Programmable Logic Array (PLA)
    - Programmable Array Logic (PAL) device
    - Complex Programmable Logic Device (CPLD)
    - Field-Programmable Gate Array (FPGA)

- A PLD’s function is not fixed
  - Can be programmed to perform different functions
Why PLDS?

• Fact:
  • It is most economical to produce an IC in large volumes

• But:
  • Many situations require only small volumes of ICs
  • Many situations require changes to be done in the field, e.g. Firmware of a product under development

• A programmable logic device can be:
  • Produced in large volumes
  • Programmed to implement many different low-volume designs
PLD Hardware Programming Technologies

• In the Factory - Cannot be erased/reprogrammed by user
  • Mask programming (changing the VLSI mask) during manufacturing

• Programmable only once
  • Fuse
  • Anti-fuse

• Reprogrammable (Erased & Programmed many times)
  • Volatile - Programming lost if chip power lost
    • Single-bit storage element
  • Non-Volatile - Programming survives power loss
    • UV Erasable
    • Electrically Erasable
      • Flash (as in Flash Memory)
Most PLD technologies have gates with very high fan-in

Fuse map: graphic representation of the selected connections
Programmable Logic Devices (PLDs)

All use AND-OR structure- differ in which is programmable

Programmable read-only memory (PROM)

Programmable array logic (PAL) device

Programmable logic array (PLA)

Ahmad Almulhem, KFUPM 2010
Read-Only Memory (ROM)

• **ROM**: A device in which “permanent” binary information is stored using a special device (programmer)

\[ k \text{ inputs (address)} \rightarrow 2^k \times n \text{ ROM} \rightarrow n \text{ outputs (data)} \]

• k inputs (address) \( \rightarrow \) \( 2^k \) words each of size n bits (data)

• ROM DOES NOT have a write operation \( \Rightarrow \) ROM DOES NOT have data inputs

**Word**: group of bits stored in one location

Ahmad Almulhem, KFUPM 2010
ROM Internal Logic

- The decoder stage produces ALL possible minterms
- 32 Words of 8 bits each
- 5 input lines (address)
- Each OR gate has a 32 input
- A contact can be made using fuse/anti-fuse

Ahmad Almulhem, KFUPM 2010
Programming a ROM

- Every ONE in truth table specifies a closed circuit
- Every ZERO in truth table specifies an OPEN circuit
- Example: At address 00011 → The word 10110010 is stored

Ahmad Almulhem, KFUPM 2010
Combinational Circuit Implementation with ROM

- ROM = Decoder + OR gates
- Implementation of a combinational circuit is easy
  - Store the truth table by programming the ROM
- Only need to provide the truth table
Example 1

**Example**: Design a combinational circuit using ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the number.

**Solution**: Derive truth table:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 A1 A0</td>
<td>B5 B4 B3 B2 B1 B0 SQ</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 0 0 0 1 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 0 1 0 0 4</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 0 1 0 0 1 9</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0 0 0 0 16</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1 1 0 0 1 25</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 0 0 1 0 0 36</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 0 0 0 1 49</td>
</tr>
</tbody>
</table>

Ahmad Almulhem, KFUPM 2010
Example 1 (cont.)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 A1 A0</td>
<td>B5 B4 B3 B2 B1 B0 SQ</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 0 0 0 1 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 0 1 0 0 4</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 0 1 0 0 1 9</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0 0 0 0 16</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1 1 0 0 1 25</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 0 0 1 0 0 36</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 0 0 0 1 49</td>
</tr>
</tbody>
</table>

ROM truth table – specifies the required connections

B1 is ALWAYS 0 → no need to generate it using the ROM
B0 is equal to A0 → no need to generate it using the ROM
Therefore: The minimum size of ROM needed is $2^3 \times 4$ or 8X4
**Example 2**

**Problem:** Tabulate the truth for an 8 X 4 ROM that implements the following four Boolean functions:

- \( A(X,Y,Z) = \Sigma m(3,6,7) \); \( B(X,Y,Z) = \Sigma m(0,1,4,5,6) \)
- \( C(X,Y,Z) = \Sigma m(2,3,4) \); \( D(X,Y,Z) = \Sigma m(2,3,4,7) \)

**Solution:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X )</td>
<td>( Y )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 3 (Size of a ROM)

**Problem:** Specify the size of a ROM (number of words and number of bits per word) that will accommodate the truth table for the following combinational circuit: An 8-bit adder/subtractor with Cin and Cout.

**Solution:**
- Inputs to the ROM (address lines) = 8 (first number) + (8 second number) + 1 (Cin) + 1 (Add/Subtract) \(\Rightarrow\) 18 lines
- Hence number of words in ROM is 218 = 256K
- Size of each word = number of possible functions/outputs
  \[= 16 \text{ (addition/subtraction)} + 1 \text{ (Cout)}\]
  \[= 17\]

Hence ROM size = 256K X 17
Sequential Circuit Implementation with ROM

• sequential circuit = combinational circuit + memory
• Combinational part can be built with a ROM as shown previously
  • Number of address lines = No. of FF + No. of inputs
  • Number of outputs = No. of FF + No. of outputs
Example

Example: Design a sequential circuit whose state table is given, using a ROM and a register.

State Table

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_2$</td>
<td>$Q_1$</td>
<td>$X$</td>
<td>$Q_2^+$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

We need a 8x3 ROM (why?)
3 address lines and 3 data lines

Exercise: Compare design with ROMs with the traditional design procedure.
Types of ROMs

A ROM programmed in four different ways:

- **ROM: Mask Programming**
  - By a semiconductor company

- **PROM (Programmable ROM)**
  - User can blow/connect fuses with a special programming device (PROM programmer)
  - Only programmed once!

- **EPROM (Erasable PROM)**
  - Can be erased using Ultraviolet Light

- **Electrically Erasable PROM (EEPROM or E²PROM)**
  - Like an EPROM, but erased with electrical signal
Other PLDs

All use AND-OR structure- differ in which is programmable

Fixed AND array (decoder)  Programmatic connections  Programmable OR array  Outputs

Programmable read-only memory (PROM)

Programmable AND array  Fixed OR array  Outputs

Programmable array logic (PAL) device

Programmable AND array  Programmatic connections  Programmable OR array  Outputs

Programmable logic array (PLA)

Ahmad Almulhem, KFUPM 2010
Programmable Logic Array (PLA)

- AND array and OR array are programmable
- XOR is available to complement an output if needed

Example:
- 3 inputs/2 outputs
- \( F_1 = A B' + A C + A' B C' \)
- \( F_2 = (AC + BC)' \)
Programmable Array Logic (PAL)

- Fixed OR array and programmable AND array
  - Opposite of ROM
- Feed back is used to support more product terms
- AND output can not be shared here!

Example:
- 4 inputs/4 outputs with fixed 3-input OR gates
- \( W = A \cdot B \cdot C' + A' \cdot B' \cdot C \cdot D' \)
- \( X = ? \)
- \( Y = ? \)
- \( Z = ? \)

Source: Mano’s textbook
Field Programmable Gate Array (FPGA)

Xilinx FPGAs

- Configurable Logic Block (CLB)
  - Programmable logic and FFs
- Programmable Interconnects
  - Switch Matrices
  - Horizontal/vertical lines
- I/O Block (IOB)
  - Programmable I/O pins

Source: Mano’s textbook
More on PLDs

- Read Section 6.8 in the textbook
- Wikipedia/Youtube