Introduction to Memories

ROM, RAM

Asynchronous & Synchronous

FIFOs, CAMs, Flash, DRAM
2. MEMORIES

SEMICONDUCTOR MEMORY CLASSIFICATION

- RANDOM ACCESS MEMORIES (RAM)
- SEQUENTIAL ACCESS MEMORIES (SAROM)
- RAMs
- READ ONLY MEMORIES (ROM)
- READ MOSTLY MEMORIES (PMRAM)
- READ WRITE MEMORIES (RWM)
- E2-PROM
- EEPROM
- STATIC (SRAM)
- DYNAMIC (DRAM)

E2-PROM = ELECTRICALLY ERASABLE AND PROGRAMMABLE ROM (ALSO KNOWN AS EEPROM)

ROM / EPROM
is used for holding permanent program/data.

RWM
is used for holding program/data which is not permanent. RWM is required for stack operation.

DEDICATED DEVICES WITH A SPECIFIC FUNCTION
(example: intelligent point-of-sale terminal, etc.)

HAVE MORE OF ROM AND LESS OF RWM AS COMARED TO A GENERAL PURPOSE COMPUTER.

EPROM is a high density low cost device but erasing and reprogramming requires the chip to be removed from the board and exposed to UV light for about 30 minutes. E2-PROM is both alterable right in the circuit, but it is high cost low density device. In May 1988, Intel announced FLASH MEMORY which has EPROM density and E2-PROM functionality.

WRITE ENDURANCE: The number of write operations allowed to any single location.
Conceptual Structure of a Memory Cell
An 8-Bit Register as a 1-D RAM Array

FIGURE 7.5 A 4 x 8 2-D Memory Cell Array
Memory can be viewed as an array of addressable registers.

Memory size: $2^n \times m$
- i.e. $2^n$ words each of $m$ bits.
- Address Range: 0 to $2^n-1$
- Bit designation: $b_{m-1} b_{m-2} \ldots b_1 b_0$

### PINCOUNT

<table>
<thead>
<tr>
<th>$A_{n-1}$ to $A_0$</th>
<th>$A_{n-1}$ to $A_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>CS</td>
</tr>
<tr>
<td>WR</td>
<td>WE</td>
</tr>
<tr>
<td>RD</td>
<td>OE</td>
</tr>
</tbody>
</table>

A memory of $2^n \times m$ size shall have:
- $n$ address inputs
- $m$ bidirectional data lines
- Chip select
- Write enable
- Output enable
- VCC & GND

**Example:** HM 6116

- $2^n = 2 \times 8$
- Address: $n = 11$
- Data: $m = 8$
- $n + m + 5 = 11 + 8 + 5 = 24$

Note: Some chips may have separate data inputs and separate data outputs.

Some chips may not have OE (output enable) control input. Then, when WE is low, data is written into the chip and when WE is high, data is read out of the chip.

### 2.2.2 Memory Internal Organization

Internally, a RW memory shall have:

1. An array of memory cells, each of which can store a single bit.
2. Logic to address any location in the memory.
3. Circuitry to allow reading of the contents of any memory location.
4. Circuitry to allow writing into any location.

**WORD ORGANIZED ARRAY with LINEAR SELECTION**

- **W**: NUMBER OF LOGICAL WORDS
- **B**: NUMBER OF BITS PER LOGICAL WORD
BUILDING LARGER MEMORIES USING SMALLER MEMORY CHIPS:

The memory required in a system may be larger than provided by a single chip.
The number of words required may be more or the length of the word required may be more.

INCREASE THE WORD LENGTH

Word length is increased by placing the outputs of two or more devices in parallel.
Ex: Build 1k×8 memory using 1k×1 chips.

INCREASE THE NUMBER OF WORDS

Note that the data lines are 3-state type
and when the chip is deselected (CS = 1), the deselected chip presents a HIGH-Z to the common DATA BUS.

Technique: Higher order memory interleaving
Example: Build 8k×8 ROM using 1k×8 chips.

The 8 different combinations of A12 A11 A10 will select the 8 different chips, one at a time.
Width Expansion

Given 2 of $1K \times 4$ build $1K \times 8$
Width Expansion

Given 2 of 1K×4 build 1K×8
Depth Expansion

Given $2 \times 1K \times 4$ build $2K \times 4$
Depth Expansion

Given 2 of 1K x 4, build 2K x 4

Diagram showing connections between A10-A0, A9-0, WE, OE, and D3-D0.
Width & Depth Expansion

Given 4 $1k \times 4$ build $2k \times 8$

Diagram:

- $Aq-A0$
- $D_3-D_0$
- $\overline{WE}$
- $\overline{OE}$
- $\overline{CS}$
First expand Width and then Depth
First expand WIDTH and then DEPTH
First expand DEPTH and then WIDTH
First expand **DEPTH** and then **WIDTH**.
Exercises on Memory Organization

(i) Memory internal organization:
Two level decoding of a 4Kx4 memory

(ii) Block diagram & labels of a 16Kx8 EPROM

(iii) Block diagram & labels of a 1Kx4 ROM

(iv) Build a 16Kx8 memory using 4Kx4 ROM chips and a 2 to 4 decoder (74LS139).
Exercises on Memory organization

(i) Memory Internal Organization:
Two level decoding of a 4K x 4 memory.

(ii) Block diagram & labels of a 16K x 8 EPROM

(iii) Block diagram & labels of a 1K x 4 RAM

(iv) Build a 16K x 8 memory using 4K x 4 ROM chips and a 2 to 4 decoder (74LS139).
Processor & Memory Map

- **8088** (simplified)
  - A19-A0
  - D7-D0
  - Mw
  - MR

- **Address Space**
  \[ 2^{20} \times 8 \]
  \[ = 1 \text{ M} \times 8 = 1 \text{ MByte} \]

- **2^{20} = 2^4 \times 2^{16}**
- **1 \text{ MByte} = 16 \times 64 \text{KB}**
- **Addresses**
  \[ 00000_4 \text{ } - FFFFF_4 \]
<table>
<thead>
<tr>
<th>Top</th>
<th>Bottom</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFF</td>
<td>00000</td>
</tr>
<tr>
<td>F0000</td>
<td>0FFFF</td>
</tr>
<tr>
<td>EFFFF</td>
<td>10000</td>
</tr>
<tr>
<td>E0000</td>
<td>1FFFF</td>
</tr>
<tr>
<td>DFFFF</td>
<td>20000</td>
</tr>
<tr>
<td>D0000</td>
<td>2FFFF</td>
</tr>
<tr>
<td>CFFFF</td>
<td>30000</td>
</tr>
<tr>
<td>C0000</td>
<td>3FFFF</td>
</tr>
<tr>
<td>BFFFF</td>
<td>40000</td>
</tr>
<tr>
<td>B0000</td>
<td>4FFFF</td>
</tr>
<tr>
<td>AFFFF</td>
<td>50000</td>
</tr>
<tr>
<td>A0000</td>
<td>5FFFF</td>
</tr>
<tr>
<td>9FFFF</td>
<td>60000</td>
</tr>
<tr>
<td>90000</td>
<td>6FFFF</td>
</tr>
<tr>
<td>8FFFF</td>
<td>70000</td>
</tr>
<tr>
<td>80000</td>
<td>7FFFF</td>
</tr>
<tr>
<td>7FFFF</td>
<td>80000</td>
</tr>
<tr>
<td>70000</td>
<td>8FFFF</td>
</tr>
<tr>
<td>6FFFF</td>
<td>90000</td>
</tr>
<tr>
<td>60000</td>
<td>9FFFF</td>
</tr>
<tr>
<td>5FFFF</td>
<td>A0000</td>
</tr>
<tr>
<td>50000</td>
<td>AFFFF</td>
</tr>
<tr>
<td>4FFFF</td>
<td>B0000</td>
</tr>
<tr>
<td>40000</td>
<td>BFFFF</td>
</tr>
<tr>
<td>3FFFF</td>
<td>C0000</td>
</tr>
<tr>
<td>30000</td>
<td>CFFFF</td>
</tr>
<tr>
<td>2FFFF</td>
<td>D0000</td>
</tr>
<tr>
<td>20000</td>
<td>DFFFF</td>
</tr>
<tr>
<td>1FFFF</td>
<td>E0000</td>
</tr>
<tr>
<td>10000</td>
<td>EFFFF</td>
</tr>
<tr>
<td>0FFFF</td>
<td>F0000</td>
</tr>
<tr>
<td>00000</td>
<td>FFFFF</td>
</tr>
</tbody>
</table>

**Memory Map**

TOP is the HIGHEST address, BOTTOM is the LOWEST address.
Memory addresses (in hex & bin) and address ranges

$00_h - FF_h \Rightarrow 8 \text{ bits going from } 8 \text{ zeros to } 8 \text{ 1's}$

$\Rightarrow 2^8 \Rightarrow 256 \text{ locations}$

$0000_h - FFFF_h = \square$

$0_h - F_h$ can be divided into 2 equal parts

$0_h - 7_h, \ 8 - F_h$

$00_h - FF_h$ can be divided into 2 equal parts

$00_h - , \ - FF_h$
Natural ranges

In decimal system 7400 to 7499 is a natural range of hundred as it can be written as 74 x x

7450 to 7549 is an unnatural range.

A 32K range starting at 4000\text{H} will end at

State a natural range of 32K which includes 77777\text{H} to
Exercise:
Say, we are going to have only 3 memory chips interfaced to the 8088 processor.
Exercise: Say, we are going to have only 3 memory chips interfaced to the 8088 processor.

64KB

FFFE
F 0000
F =

FFFE
F 0000
O =
Say, we are going to have only 3 memory chips interfaced to the 8088 processor.

SOLUTION

64kB
Exercise: Given a 32KB chip, a 8KB chip, and another 8KB chip, build a 48KB (= 32 + 8 + 8) memory, starting at 50000H.

8KB

A19 A8 A7 A6 8KB

32KB

5TH

MR

OE

CE

50000

32KB

MW

D7-D0

MR

WE

OE

CE

5TH

5
SOLUTION: Given a 32kB chip, a 8kB chip, and another 8kB chip, build a 48kB (\(= 32 + 8 + 8\)) memory, starting at 50000h.

5BFFFF
A0000
upper 7 bits = 0101 - 101
A15 A14 A13

A12-Ao
A12-Ao

A19 A8 A17 A16
8kB

57FFFF
00000
upper 5 bits = 01010
A14-A0
32kB

59FFFF
80000
upper 7 bits = 0101 - 100
A15 A14 A13

A12-Ao
A12-Ao

MW WE D7-D0
D7-D0

5TH
MR
WE
OE
CE

5TH
MR