Week 8

Memory and Memory Interfacing
Semiconductor Memory Fundamentals

- In the design of all computers, semiconductor memories are used as primary storage for data and code.
- They are connected directly to the CPU and they are the memory that the CPU asks for information (code or data).
- Among the most widely used are RAM and ROM.
- **Memory Capacity**
  - The number of bits that a semiconductor memory chip can store is called its chip capacity (bits or bytes).
- **Memory Organization**
  - Each memory chip contains $2^x$ locations where $x$ is the number of address pins on the chip.
  - Each location contains $y$ bits, where $y$ is the number of data pins on the chip.
  - The entire chip will contain $2^x \times y$ bits.
  - Ex. Memory organization of $4K \times 4$: $2^{12} = 4096$ locations, each location holding 4 bits.
- **Memory Speed** (access time)
Memory

• Each memory device has at least one *chip select* (CS) or *chip enable* (CE) or *select* (S) pin that enables the memory device.
  – This enables read and/or write operations.

• Each memory device has at least one control pin.
  – For ROMs, an *output enable* (OE) or *gate* (G) is present. The OE pin enables and disables a set of tristate buffers.
  – For RAMs, a *read-write* (R/W) or *write enable* (WE) and *read enable* (OE) are present.
Memory Types

• ROM (Read Only Memory)
  - ROM is the type of memory that does not lose its contents when power is turned off. It is also called nonvolatile memory.
  - PROM (Programmable Memory)
    - User programmable (one-time programmable) memory
    - If the information burned into PROM is wrong, it needs to be discarded since internal fuses are blown permanently.
    - Special equipment needed: ROM burner or ROM programmer
  - EPROM (Erasable Programmable ROM) 2,000 times
    - Allows making changes in the contents of PROM after it is burned
    - One can program the memory chip and erase it thousands of times
    - Erasing its contents can take up to 20 minutes; the entire chip is erased
    - All EPROM chips have a window that is used to shine ultraviolet (UV) radiation to erase its contents
    - Also referred to as UV-EPROM
Memory Types

- **EEPROM (Electrically Erasable ROM)** 500,000 times
  - Method of erasure is electrical
  - Moreover, one can select which byte to be erased
  - Cost per bit is much higher than for UV-EPROM

- **Flash Memory EPROM**
  - First, the process of erasure of the entire contents takes less than a second, or one might say in a flash, hence its name: flash memory
  - When flash memory’s contents are erased, the entire device is erased.
  - Even though flash memories are writeable, like EPROMs they find their widest use in microcomputer systems for storage of firmware

- **RAM (Random Access Memory)** infinite times
  - RAM memory is called volatile memory since cutting off the power to the IC will mean the loss of data.
  - Also referred to as R/WM (Read And Write Memory)
Minmode 8088 Microcomputer system memory circuitry

Figure 9-37  (a) Minimum-mode 8088 system memory interface. (Reprinted with permission of Intel Corporation, Copyright/Intel Corp. 1981) (b) Minimum-mode 8086 system memory interface. (Reprinted with permission of Intel Corporation, Copyright/Intel Corp. 1979) (c) Maximum-mode 8088 system memory interface. (Reprinted with permission of Intel Corporation, Copyright/Intel Corp. 1981)
Minmode 8086 Microcomputer system memory circuitry
Maxmode 8088 Microcomputer system memory circuitry

Figure 9–37 (continued)
Memory Interface
Bank Write Control Logic

\[ \overline{\text{BHEL}} \quad \overline{\text{MWRC}} \quad \overline{\text{A}_{\text{OL}}} \]

\[ \overline{\text{WR}_U} \quad \overline{\text{WR}_L} \]

\[ \text{7432} \]

\[ \text{7432} \]
Bank Read Control Logic
Address Bus Configuration with Address Decoding
Address Latch Circuit
Generation of MEMRD & MEMWR in Minmode

Control Signal Generation Circuit
8088 Memory and I/O address spaces

We first look at the memory addressing.

Figure 8-46 8088/8086 memory and I/O address spaces.
Address Selection

(a)

(b)
74F139 2-line to 4-line decoder

(a) 2-line to 4-line decoder

(b) Diagram showing the 2-line to 4-line decoder with enable and select inputs and data outputs.

(c) Truth table for the 74F139 decoder, showing the inputs (Enable, Select) and outputs (Y0, Y1, Y2, Y3).
Memory Address Decoding

3-8 Decoder
(for example: 74LS138)
Address Decoder Circuit

Figure 8–35  Address decoder circuit
Example on Address Decoding

A circuit containing 32KB of RAM is to be interfaced to an 8088 based system, so that the first address of the RAM is at 48000H. What is the entire range of the RAM Address? How is the address bus used to enable the RAMs? What address lines should be used?

<table>
<thead>
<tr>
<th>A_19</th>
<th>A_18</th>
<th>A_17</th>
<th>A_16</th>
<th>A_15</th>
<th>A_14</th>
<th>A_13</th>
<th>A_12</th>
<th>A_11</th>
<th>A_10</th>
<th>A_9</th>
<th>A_8</th>
<th>A_7</th>
<th>A_6</th>
<th>A_5</th>
<th>A_4</th>
<th>A_3</th>
<th>A_2</th>
<th>A_1</th>
<th>A_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

These 5 address lines set the base address of the memory.

These 15 address lines will select one of $2^{15}$ (or 32,768) locations inside the RAMs.
Example on Address Decoding

Memory Address Decoder for 48000 to 4FFFF Range
Memory Address Decoding

IO/M

Logic 0 when $A_{11}$ through $A_{19}$ are all 1.

RD of 8088/86
Or MRDC bus signal.

2716
(2K X 8)
EPROM

$A_0$
$A_1$
$A_2$
$A_3$
$A_4$
$A_5$
$A_6$
$A_7$

Address Bus

Data Bus

$A_{19}$
$A_{18}$
$A_{17}$
$A_{16}$
$A_{15}$
$A_{14}$
$A_{13}$
$A_{12}$
$A_{11}$
Memory Address Decoding

$A_{13}$ through $A_{15}$ select a 2764
$A_{16}$ through $A_{19}$ enable the decoder

Address space F000H-FFFFFH

2764 (8K X 8) EPROM

Address Bus

Data Bus
Memory Addressing

512 K of SRAM 00000-7FFFF
Partial Address Decoding

- Not all the address lines need to be used. (A14-A19 not used). So FFFF0, 3BFF0, 07FF0 pr C3FF0 get the same data.
- (+) The purpose is to get the job done in minimum hardware.
- (-) Feature expansion of the memory is impossible, and may cause invalid data reads due to overlapping memory segment reads (a fatal error)
Partial Address Decoding

### Foldback memory exists

<table>
<thead>
<tr>
<th>A_{19}</th>
<th>A_{18}</th>
<th>A_{17}</th>
<th>A_{16}</th>
<th>A_{15}</th>
<th>A_{14}</th>
<th>A_{13}</th>
<th>A_{12}</th>
<th>A_{11}</th>
<th>A_{10}</th>
<th>A_{9}</th>
<th>A_{8}</th>
<th>A_{7}</th>
<th>A_{6}</th>
<th>A_{5}</th>
<th>A_{4}</th>
<th>A_{3}</th>
<th>A_{2}</th>
<th>A_{1}</th>
<th>A_{0}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**For DECODER**

**To EPROM**

**Foldback memory exists**
Generating Wait States in Hardware

Ready signal can be generated by special hardware using the SEL' signal from the address decoder.

The circuit above will generate two clock periods of zero signal for the READY output.
A complete RAM/EPROM Memory

[Diagram of RAM/EPROM Memory]
Examples: Find different addressing for CS (A0-A13 used by memories)

10x1 10/xx xxxx xxxxxxxxxx

x011 0x/xx xxxx xxxxxxxxxx

01xx x0/xx xxxx xxxxxxxxxx

(A19 and A18=0) or A16=1 or A14=1

xx1x xx/xx xxx x xxxxxxxxxx
ROM
Figure 9–4  Read-only memory interface.
Intel 2716 EPROM (2K x 8):

- **Pin(s)**
  - A<sub>0</sub>-A<sub>10</sub>: Address
  - PD/PGM: Power down/Program
  - CS: Chip Select
  - O<sub>0</sub>-O<sub>7</sub>: Outputs

- **Function**
  - Address
  - Power down/Program
  - Chip Select
  - Outputs

**V<sub>PP</sub>** is used to program the device by applying 25V and pulsing **PGM** while holding **CS** high.
EPROM Critical Timing

2716 Timing diagram:

Address

CS

High Z

Data Out Valid

Read Mode (PD/PGM = VIL)

tACC1

tOH

tDF

Sample of the data sheet for the 2716 A.C. Characteristics.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Limits</th>
<th>Unit</th>
<th>Test Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>tACC1</td>
<td>Addr. to Output Delay</td>
<td>250</td>
<td>450</td>
<td>ns</td>
</tr>
<tr>
<td>tOH</td>
<td>Addr. to Output Hold</td>
<td>0</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>tDF</td>
<td>Chip Deselect to Output Float</td>
<td>0</td>
<td>100</td>
<td>ns</td>
</tr>
</tbody>
</table>

This EPROM would need a READY generation to work with a 8086 with 5Mhz.
RAM types

• **SRAM (Static RAM)**
  – Storage cells are made of flip-flops and therefore they do not require refreshing to keep their data
  – Cells handling one bit requires 6 or 4 transistors each, which is too many
  – SRAMS are widely used for cache memory and battery-backed memory systems.
  – Speeds as fast as 10ns. But limited in size ~256Kx8

• **DRAM (Dynamic RAM)**
  – Uses MOS capacitors to store a bit
  – Requires constant refreshing due to leakage (every 2ms – 4ms)
  – Advantages
    • High density (capacity) ~1GBX8
    • Cheaper cost per bit
    • Lower power consumption
  – Disadvantage
    • While it is being refreshed, data cannot be accessed
    • Larger access times
    • Too may pins due to large size
Figure 9-12  16K x 16-bit SRAM circuit.
SRAM

<table>
<thead>
<tr>
<th>SRAM</th>
<th>Density (bits)</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>4361</td>
<td>64K</td>
<td>64K x 1</td>
</tr>
<tr>
<td>4363</td>
<td>64K</td>
<td>16K x 4</td>
</tr>
<tr>
<td>4364</td>
<td>64K</td>
<td>8K x 8</td>
</tr>
<tr>
<td>43254</td>
<td>256K</td>
<td>64K x 4</td>
</tr>
<tr>
<td>43256A</td>
<td>256K</td>
<td>32K x 8</td>
</tr>
<tr>
<td>431000A</td>
<td>1M</td>
<td>128K x 8</td>
</tr>
</tbody>
</table>

**Figure 9–13**
devices.

**Figure 9–14**  (a) 4364 pin layout. (b) 43256A pin layout.
**DRAM**

<table>
<thead>
<tr>
<th>DRAM</th>
<th>Density (bits)</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2164B</td>
<td>64K</td>
<td>64K × 1</td>
</tr>
<tr>
<td>21256</td>
<td>256K</td>
<td>256K × 1</td>
</tr>
<tr>
<td>21464</td>
<td>256K</td>
<td>64K × 4</td>
</tr>
<tr>
<td>421000</td>
<td>1M</td>
<td>1M × 1</td>
</tr>
<tr>
<td>424256</td>
<td>1M</td>
<td>256K × 4</td>
</tr>
<tr>
<td>44100</td>
<td>4M</td>
<td>4M × 1</td>
</tr>
<tr>
<td>44400</td>
<td>4M</td>
<td>1M × 4</td>
</tr>
<tr>
<td>44160</td>
<td>4M</td>
<td>256K × 16</td>
</tr>
<tr>
<td>416800</td>
<td>16M</td>
<td>8M × 2</td>
</tr>
<tr>
<td>416400</td>
<td>16M</td>
<td>4M × 4</td>
</tr>
<tr>
<td>416160</td>
<td>16M</td>
<td>1M × 16</td>
</tr>
</tbody>
</table>

**Figure 9–19** Standard DRAM devices.

**Figure 9–21** Block diagram of the 2164B DRAM.

**Figure 9–20** (a) 2164B pin layout. (b) 21256 pin layout. (c) 421000 pin layout.
In DRAM, the 8 address lines are latched accordingly by the strobe of the RAS and CAS signals.

For example: To load a 16 bit address into the DRAM 8 bits of the address are first latched by pulling RAS low, then other 8 bits are presented to A0-A7 and CAS is pulled low.
Refresh time example:
For a 256K X 1 DRAM with 256 rows, a refresh must occur every 15.6us (4ms/256).
For the 8086, a read or write occurs every 800ns.
This allows 19 memory reads/writes per refresh or 5% of the time.
DRAM Addressing

Figure 9-22  64K × 16-bit DRAM circuit.
Larger DRAMs are available which are organized as $1M \times 1$, $4M \times 1$, $16M \times 1$, $64M \times 1$, $256M \times 1$. DRAMs are typically placed on SIMM (Single In-line Memory Modules) boards. 

30-pin SIMMs come in $1M \times 8$, $1M \times 9$ (parity), $4M \times 8$, $4M \times 9$.
72-pin SIMMs come in $1/2/3/8/16M \times 32$ or $1M \times 36$ (parity).

Pentiums have a 64-bit wide data bus. The 30-pin and 72-pin SIMMs are not used on these systems. Rather, **64-bit DIMMs** (Dual In-line Memory Modules) are the standard. These organize the memory 64-bits wide. The board has DRAMs mounted on both sides and is **168 pins**
DRAM can be refreshed by an external circuitry including an 8 bit counter.

- HOLD/HLDA used

- Only the columns of the matrix (256 x 256 for a 64K bit matrix) is needed to be refreshed.

- The refresh rate can be adjusted using a 555 timer circuitry.
DRAM in PC

Figure 10-21. 640K Bytes of DRAM with odd and even banks designation

Figure 10-22. 16-bit Data Connection in the 80286 System
Parity circuits

Figure 9–23  Data-storage memory interface with parity-checker generator.
Parity circuits

(a) Block diagram of the 74AS280. (Texas Instruments Incorporated)
(b) Function table. (Texas Instruments Incorporated)
(c) Even-parity checker/generator connection.
Parity Error Detection Circuit

64Kx8

Parity Generate

64Kx1

Parity Detect
Checksum byte (used for ROM)

✓ Add the bytes together and drop the carries
✓ Take the 2’s complement of the total sum, and that is the checksum byte, which becomes the last byte of the stored information.
✓ To perform the checksum operation add all the bytes, including the checksum byte. The result must be zero. If it is not zero, one or more bytes of data have been changed (corrupted)

Example: Assume that we have 5 bytes of hexadecimal data: 1A, 14, 82, FC, 3E.

a) Find the checksum byte
b) Perform the checksum operation to ensure integrity
c) If the 3rd byte is changed to 44 show how the error is detected?

a) The checksum is: \(1A + 14 + 82 + FC + 3E = 1EA\) drop 1 \(\rightarrow\) EA , take 2’s comp \(\Rightarrow\) 16
b) \(1A + 14 + 82 + FC + 3E + 16 = 00\)
c) \(1A + 14 + 44 + FC + 3E + 16 = 1C2\) \(\Rightarrow\) Error!
IBM PC Memory Map

- **00000h – 9FFFFh**: RAM (640 Kb)
  - The first 1K used for the interrupt vector table (00000h to 003FFh)
  - 00400h to 004FFh is set aside for the BIOS temporary area
  - 00500h to 005FFh is set aside for the temporary storage of certain parameters in DOS and BASIC
  - A certain number of Kbytes is occupied by the operating system itself
- **A0000h – BFFFFh**: Video Display RAM (128 Kb)
  - A total of 128 Kbytes is allocated for video
  - Of that 128K, only a portion is used for VDR, the amount depending on which type of video adapter card is installed in the system
- **C0000h – FFFFFh**: ROM (256 Kb)
  - 256 K is set aside for ROM
  - Used in
    - BIOS ROM, Basic language compiler ROM, hard disk controller, other peripheral board ROMS and the rest for expansion by the user
IBM PC Memory Map

Figure 11-6. Memory Map of the IBM PC

Figure 11-8. PC XT Detailed Memory Map
## IBM PC Memory Map

### Table 11-5: System Identification Byte for Some IBM Products

<table>
<thead>
<tr>
<th>Product</th>
<th>BIOS Date</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>04/24/81</td>
<td>FF</td>
</tr>
<tr>
<td>PC</td>
<td>10/19/81</td>
<td>FF</td>
</tr>
<tr>
<td>PC</td>
<td>10/27/82</td>
<td>FF</td>
</tr>
<tr>
<td>PC XT</td>
<td>11/08/82</td>
<td>FE</td>
</tr>
<tr>
<td>PC XT</td>
<td>01/10/86</td>
<td>FB</td>
</tr>
<tr>
<td>PC XT</td>
<td>05/09/86</td>
<td>FB</td>
</tr>
<tr>
<td>PC jr</td>
<td>06/01/83</td>
<td>FD</td>
</tr>
<tr>
<td>AT</td>
<td>01/10/84</td>
<td>FC</td>
</tr>
<tr>
<td>AT</td>
<td>06/10/85</td>
<td>FC</td>
</tr>
<tr>
<td>AT</td>
<td>11/15/85</td>
<td>FC</td>
</tr>
<tr>
<td>PC XT Model 286</td>
<td>04/21/86</td>
<td>FC</td>
</tr>
<tr>
<td>PC Convertible</td>
<td>09/13/85</td>
<td>F9</td>
</tr>
<tr>
<td>Personal System 2 Model 30</td>
<td>09/02/86</td>
<td>FA</td>
</tr>
<tr>
<td>Personal System 2 Model 50</td>
<td>*</td>
<td>FC</td>
</tr>
<tr>
<td>Personal System 2 Model 60</td>
<td>*</td>
<td>FC</td>
</tr>
<tr>
<td>Personal System 2 Model 80</td>
<td>*</td>
<td>F8</td>
</tr>
<tr>
<td>Personal System 2 Model 80</td>
<td>*</td>
<td>F8</td>
</tr>
</tbody>
</table>

### Memory Location

- F000:E000 - F000:E07F: Power-On Start-Up Tests (POST)
- F000:E0F2 - F000:E078: Boot strap loader (INT 19H)
- F000:E729 - F000:E72D: RS-232 I/O (INT 14H)
- F000:E82E - F000:E881: Keyboard I/O (INT 16H)
- F000:E882 - F000:E986: Keyboard scan code tables
- F000:E987 - F000:EC58: Keyboard (INT 9H)
- F000:EC59 - F000:EFD1: Diskette I/O (INT 13H)
- F000:EFD2 - F000:F044: Printer I/O (INT 17H)
- F000:F045 - F000:F840: Video I/O (INT 10H)
- F000:F841 - F000:F84C: Memory check (INT 12H)
- F000:F84D - F000:F85B: Equipment check (INT 11H)
- F000:F85C - F000:FA6D: Cassette I/O (INT 15H) - not used on XT
- F000:FA6E - F000:FE6D: Graphics character table
- F000:FE6E - F000:FEF2: Time of day (INT 1AH)
- F000:FEF3 - F000:FF52: Interrupt vector table
- F000:FF53 - F000:FF53: Dummy return point for unused interrupts
- F000:FF54 - F000:FFD9: Print screen (INT 5H)
- F000:FFF0 - F000:FFF4: First code executed after power-on
- F000:FFF5 - F000:FFFD: BIOS release date

(Reprinted by permission from "IBM Technical Reference" c. 1987 by International Business Machi

Figure 11-11. PC/XT BIOS ROM Memory Map