Introduction to Microcomputer and Microprocessor

By
Dr. Sikder Sunbeam Islam
EEE, IIUC
Microcomputer System: Introduction

- Microcomputer is used to describe a system that includes a minimum of a microprocessor, program memory, data memory, and input/output (I/O). It is called ‘micro’ because of its small size.

- Power of the Microcomputer is determined by the capabilities of microprocessor. It’s clock frequency determines the speed of the microcomputer. Microprocessor is the CPU (central processing unit) of the microcomputer.

- Microcomputer has three basic blocks:
  - (a) Microprocessor Unit
  - (b) Memory Unit, and
  - (c) Input /output unit
Microcomputer System: Introduction cont.

Figure 1: General Architecture of a Microcomputer System
Microcomputer System: Microprocessor Unit/CPU

- **Microprocessor Unit/CPU**: The CPU executes all instructions and performs arithmetic and logic operations on data. The CPU of microcomputer is called Microprocessor. CPU is the "brain" of the microcomputer.

- CPU has two parts, the arithmetic and logic unit (ALU), and control unit.

- CPU also has several storage places called registers.

- In program execution, the CPU reads and executes the programs instructions one by one from the main memory. The execution of instructions may involve the arithmetic/logic operations and/or transfer data between CPU and main memory (or I/O ports).
Microcomputer System: Memory Unit

- The **memory unit** contains both program (list of instructions) and data.

- **Memory unit:**
  
  (a) Processor Memory: Registers
  (b) Primary (or main) memory: RAM and ROM
  (c) Secondary memory: Hard disks, CD, floppy disks, tape

- **Processor Memory** refers to a set of CPU registers. These registers are useful to hold temporary results when a computation is in progress.

- **Primary memory** is the memory that the CPU can access directly. Examples of main memory include RAM, ROM, etc.

- **RAM (Random Access Memory)** can be read and written. It does not retain its content when power is turn off. It is used to store data which are temporary and might change during the course of execution.
Microcomputer System: Memory and I/O Unit

- **ROM (Read Only Memory)** can be read and it retains its content when power is turned off. It is used to store program and data which do not change during the course of execution.

- The **Secondary memory** cannot be addressed directly (cannot access specified memory location) by the CPU. Examples of secondary include floppy disk, hard disk, CD, etc. The information in the secondary memory must be copied to the main memory so that CPU can access it.

- **Input Unit and Output Unit:** Input and Output units are the means by which the MPU communicates with the external devices.
  - Examples: Input unit: keyboard, mouse, scanner etc.
    - Output unit: Monitor, printer etc.
    - Bi-directional ports: Modem, Network etc.

- Each input and output port has a unique address. Some microprocessors assign the memory and I/O with separate address space (e.g., Intel x86 processors). In this case, one may use memory address X and also I/O address X the same time.
Microcomputer System: How Computer Works with CPU?

- Both program and data are fed into the CPU and transferred to their respective memory locations.
- The CPU reads the first instruction (Like, instruction may be simple ADD two numbers) from the program memory with help of address of memory location and control signal and then execute it.
- After the data manipulation is completed result is transferred to the output of the computer.
Microcomputer System: Microprocessor and Microcontroller

- **Microprocessor** is a multipurpose, programmable register based electronic device which read binary instructions from memory, processes the input data as per instructions and provides output. It is an IC which has only the CPU inside them. Such as Intel’s Pentium 1,2,3,4, core 2 duo, i3, i5 etc. These microprocessors don’t have RAM, ROM, and other peripheral on the chip.

- **Microcontroller** is a device that includes microprocessor, memory and input/output devices on a single chip. Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip.
### Comparison between Microcontroller and Microprocessor

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>No.</th>
<th>Microprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller having inbuilt RAM or ROM and inbuilt timer.</td>
<td>1</td>
<td>Do not have inbuilt RAM or ROM and timer.</td>
</tr>
<tr>
<td>Input and output ports are available.</td>
<td>2</td>
<td>Input and output ports are not available</td>
</tr>
<tr>
<td>Separate memory to store program and data.</td>
<td>3</td>
<td>Program and data are stored in same memory.</td>
</tr>
<tr>
<td>Many functional pins on the IC.</td>
<td>4</td>
<td>Less multifunction pins on IC.</td>
</tr>
<tr>
<td>It takes few instructions to read and write data from external memory.</td>
<td>5</td>
<td>It takes many instructions to read and write data from external memory.</td>
</tr>
</tbody>
</table>
The common way of categorizing microprocessor is by the number bits that their ALU can work with, at a time. Over time, five standard data widths have evolved for microprocessors: 4-bit, 8-bit, 16-bit, 32-bit, 64-bit.

Like, 8-bit microprocessor means that ALU can work with 8-bit number at a time or, data width of this microprocessor is 8-bit.

Depending on registers microprocessor can be classified in two types:

a) Accumulator based: Use accumulator register as one data source for ALU operation. 8-bit Microprocessors are accumulator based. Example: Intel 8085 and Motorola 6809.

b) General purpose register based: Registers can be used for holding data, address, results. 16-and 32-bit Microprocessors are General purpose register based. Like, Intel 8086, Motorola 6800.
Microcomputer System: Evaluation of Microprocessors

- We can divide the years of development of microprocessors as 5 generations (based on data width):

- The first commercial microprocessor was introduced by Intel in 1971 named Intel 4004. It was 4 bit device, evolved from the development effort while designing a calculator chip. They were fabricated using PMOS technology, supported only 45 instructions and not compatible with TTL.

- In 1972 Intel came out with 8008 which was capable of working using 8-bit words.

- Second Generation (1974 - 1978): In 1974 Intel announced 8-bit microprocessor 8080 which had much larger instruction set. It was NMOS technology based and faster than 8008. It is called 2nd generation microprocessors. Some of the popular 2nd generation processors were: Motorola’s 6800 and 6809; Intel’s 8085; Zilog’s Z80.
Microcomputer System: Evaluation of Microprocessor contin....

- Third generation microprocessors (1979 - 80): Age of 16 - bits microprocessors. Example: Intel’s 8086/80186/80286 and Motorolla’s 68000/68010. They were designed using HMOS technology. Four times better than that of NMOS.


- Fifth Generation (1995 - till date): Introduction of 64-bit microprocessors. Intel leads the show here with Pentium, Celeron and very recently dual and quad core processors working with up to 3.5GHz speed.
Figure: Simplified Microprocessor Architecture
Generally microprocessor unit ( MPU) contain storage elements called registers, computational circuitry called ALU, decoding and control Unit. These are the main units of typical microprocessor.

**Internal Registers:** This is a small amount of internal memory that is used for the quick storage and retrieval of data and instructions.

**Program Counter (PC):** It is the register that contains the address of the instruction or operation code. This register is used to hold the memory address of the next instruction that has to execute in a program. The width of the PC is the same as the width of the address bus.

Upon activiting the microprocessor reset input, the address of the first instruction to be executed is loaded into the PC. In order to execute the instruction, the microprocessor typically places the contents of the PC on the address bus and reads (fetches) the contents of this address (i.e. instruction)from memory. The PC contents are automatically incremented by microprocessor internal logic. The microprocessor thus executes the program sequentially.
Microcomputer System: Microprocessor Architecture contin....

- **Accumulator (A, or ACC):** Accumulator is the register associated with ALU operations and sometimes I/O operations. It is used to store results after most ALU operations. 8-bit microprocessors are usually accumulator based. Some processors (16 or 32-bits MPU) have general purpose registers that may be used as accumulator.

- **Status Register (Flags):** Available in all microprocessors. The individual bits in this register is called Flag. The conditions of the Flags change with respective ALU operations.

- **General Purpose Registers:** General purpose registers may be used to temporary store data and hold address or result of ALU operations. It can store address or data for indefinite period of time then to retrieve address or data when needed. 16 or 32-bit microprocessors (Like, Intel 8086/80386 or Motorolla 68000/68020) are usually general purpose register based.
Microcomputer System: Microprocessor Architecture conti...:

- **Instruction Register (IR):** This register stores an Instruction. After fetching an instruction from memory, the microprocessor stores it in the IR. This 8-bit register is decoded internally by microprocessor which then perform the desired operations.

- **Memory Buffer Register (MBR):** When an instruction or data is obtained from the memory or elsewhere, it is first placed in the memory buffer register. The next action to take is then determined and carried out, and the data is moved on to the desired location.

- **Stack Pointer Register:** It is a specialized register that keeps truck of next available memory location in the stack. Stack is reserved area in the RAM used for temporary storage data, return address and contents of registers. Stack is used during subroutine call and interrupts.
Microcomputer System: Microprocessor Architecture continua....

- **Arithmetic and Logic Unit (ALU):** This unit of microprocessor performs arithmetic, logic and rotate operations which affects the status register (Flags). The results from ALU are placed back in the accumulator via the internal bus. The size of the ALU confirms the type of the microprocessor. Accumulator and temporary registers many times considered as a part of ALU.

  ALU basically performs the following tasks:

  - Addition, subtraction, Multiplication and division
  - Logical operations and comparisons
  - Bit shifting and rotating operations (move the content of the Accumulator or other register 1 bit to left or right)
  - Finding 1’s complement of data
Microcomputer System: Microprocessor Architecture contin....

- **Control Unit (CU):** This unit of microprocessor performs instructions interpreting and sequencing.

- Normally the instruction decoder interprets the contents of the instruction register to the control unit and then the control unit responds to the instruction by generating sequence of enable signals which activate the appropriate ALU logic blocks (Adder, Shifter, etc.).

- In sequencing phase, it also determines the address of the next instruction to be executed and load it into the program counter.
There are two prevalent instruction set architectures:

- **RISC (Reduced Instruction Set Computer) Architecture.**
- **CISC (Complex Instruction Set Computer) Architecture.**

<table>
<thead>
<tr>
<th>RISC</th>
<th>CISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small number of instructions in the instruction set compared to CISC.</td>
<td>CISC chips have a large amount of different and complex instructions</td>
</tr>
<tr>
<td>RISC chips use simpler instructions sets to achieve higher clock frequencies and process more instructions per clock cycle than CISC processors</td>
<td>In common, CISC chips process smaller instructions per clock cycle than RISC processors, but to do same task number of instruction is smaller in CISC than RISC.</td>
</tr>
<tr>
<td>It’s hardware design is easy but software (programming) is difficult because you have to write code using a small number of instructions.</td>
<td>It’s hardware design is difficult but programming is easy as it has a large number of instructions.</td>
</tr>
<tr>
<td>Apple and SUN use RISC architecture.</td>
<td>Intel and AMD develop CISC processors. 8086 microprocessor has CISC architecture.</td>
</tr>
</tbody>
</table>
Microcomputer System: System Bus

- **System Bus**: Bus refers to the group of wires that interconnect components (processor, memory and peripherals) in a microcomputer system.

- The system bus consists of three different groups of wiring, called the data bus, control bus and address bus.
Microcomputer System: System Bus contin...

- **Control Bus:** The control bus contains lines that select the memory or I/O and cause them to perform a read or write operation. In most computer systems, there are four control bus connections:
  - MRDC (Memory read control)
  - MWTC (Memory Write control)
  - IORC (I/O read control)
  - IOWC (I/O write control)

- **Data Bus:** This is used for the exchange of data between the processor, memory, and peripherals, and is bi-directional so that it allows data flow in both directions along the wires. Data bus may be 8 bits, 16 bits, 32 bits or 64 bits.

- **Address Bus:** This is a unidirectional bus. This bus is usually 8 to 32 bits wide. Information transfer takes place from the microprocessor to the memory or I/O elements. For a 16 bit address bus, microprocessor can generate $2^{16} = 65,536$ different possible address.
Microcomputer System: Instruction Execution

- To execute an instruction processor must follow minimum of four steps: (1) Instruction Fetch (2) Instruction Decode (3) Instruction execute (4) Store or, Store of Result

- **Instruction Fetch**: Control unit collects the instructions from main memory and puts them in CPU register. This is called instruction fetch.

- **Instruction Decode**: When instruction reaches in processor register, CU decodes or interprets the instruction and sends necessary signals and data to ALU.

- **Instruction Execute**: ALU processes the data with arithmetic and logic operations and gives a result according to instructions.

- **Store Result**: Finally CU stores result in Accumulator or main memory.

- These four steps to execute an instruction are called **machine cycle**.
Classification of Memory

- Main purpose of memory unit is to hold programs and data.
- The cost of memory unit is so prohibitive that it is practically not feasible to design a large memory unit with one technology that guarantees a high speed.
- Memory is usually designed with different technologies such as solid state, magnetic and optical.
Classification of Memory

- A microcomputer memory system logically divided into **three** groups:

- **Processor Memory:** Refers to CPU registers. There is no speed disparity between the registers and the microprocessor because they are fabricated using the same technology.

- **Primary Memory:** CPU can access directly in primary memory. All program and data are saved in here prior to execution.

- **Secondary Memory:** Holds large data files and huge programs such as, compilers and database management systems. Also referred to as auxiliary or backup.

Fig. Memory classifications
**Classification of Memory**

- **RAM (Random Access Memory):** One type of volatile memory. RAM can be made of magnetic core or by semiconductor. Semiconductor RAM's are two types:
  1. Static RAM
  2. Dynamic RAM

- Difference between Static and Dynamic RAMs:

<table>
<thead>
<tr>
<th>Static RAM</th>
<th>Sl.</th>
<th>Dynamic RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bipolar transistors used for constructions.</td>
<td>1</td>
<td>MOS transistors used for constructions.</td>
</tr>
<tr>
<td>Information is stored in the form of voltage levels in flip-flops.</td>
<td>2</td>
<td>Information is stored in the form of electrical charges in capacitors</td>
</tr>
<tr>
<td>Power is required even when the chip is in standby mode.</td>
<td>3</td>
<td>Refresh logic is inbuilt, so draws less power comparatively.</td>
</tr>
<tr>
<td>These voltage levels do not get drifted away</td>
<td>4</td>
<td>Has tendency to be leaked away (Stored 1 would become 0).</td>
</tr>
<tr>
<td>Four times larger in size compared to an equivalent dynamic cell.</td>
<td>5</td>
<td>Four times more bits can be accommodated than a static RAM chip.</td>
</tr>
<tr>
<td>No refresh logic is needed</td>
<td>6</td>
<td>Refresh logic is necessary to prevent information loss</td>
</tr>
</tbody>
</table>
Classification of Memory

- **ROM: Bipolar: Mask ROM:** Contents are programmed by the manufacturer. Example: Character generator ROM2513.

- **ROM: Bipolar: PROM(Programmable ROM):** User can program this ROM but for only one time. Can not be reprogrammed.

- **ROM: MOS: EPROM(Erasable PROM):** Programs are entered using electrical impulses and stored information is erased by UV rays.

- **ROM: MOS: EAROM/EEPROM(Electrically Alterable ROM/Electrically Erasable PROM):** Can be programmed even when they are in circuit board. Slower write times than read times.
Main Memory Array Design: Addressing Memory

- Using 3-bit we can address $2^3=8$ distinct memory location. So, n number of bits can address $2^n$ memory locations.

- 1K Byte = 1024 Byte.
  
  $2^{10}=1024$

- So, for addressing 1KB of memory we need at least 10 bits. And the addresses will be:
  
  0 to 1023 (decimal) or, 000H to 3FFH (HEX) or, 0000 0000 0000 to 0011 1111 1111 (Binary). Similarly, for addressing 64 KB of memory we need at least 16 bits.

![Memory Array Diagram]
How many bits are necessary to address 1MB memory?

- 1MB = 1024 Kbyte = 1,048,576 Byte = $2^{20}$ Byte
- So, we need 20 bits to address 1MByte memory. Addresses will be-
- 0 to 1,048,575 (Decimal)
- 00000H to FFFFFH (Hex)
- 0000 0000 0000 0000 0000 0000 0000 0000 to 1111 1111 1111 1111 1111 1111 1111 1111 B (Binary)
- Microprocessor 8086 has 20-bit address bus. So, it can addresses 1 MB memory.

In many applications, a large size capacity is often realized by interconnecting several small size memory blocks. There are two kinds of techniques used for designing the main memory in such cases. They are: (1) Linear Decoding. (2) Fully Decoding.
Main Memory Array Design: Typical RAM

- To grasp the idea of interconnection technique first we should know about a small memory block. Let us consider the block diagram of a typical RAM IC.

- The capacity of this chip is 1Kbytes (=1024 bytes)
- 10-bit address lines A9 – A0; so can be accessed $2^{10}=1024$ different memory locations;
- 8-bit bidirectional data bus (D7– D0)
- Operation of this chip is governed by two control inputs: $\overline{WE}$ (Write Enable) and $\overline{CS}$ (Chip Select)
Main Memory Array Design: Typical RAM

The following truth table describes the operation of this chip:

<table>
<thead>
<tr>
<th>CS</th>
<th>WE</th>
<th>MODE</th>
<th>Status (D7 – D0)</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>X</td>
<td>Not selected</td>
<td>High Impedance Acts as</td>
<td>Standby</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>Write</td>
<td>Input Bus Acts as</td>
<td>Active</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>Read</td>
<td>Output Bus</td>
<td>Active</td>
</tr>
</tbody>
</table>

- When CS is high, chip is not selected at all, hence D7 to D0 are driven to **high impedance** state.
- When CS = 0 and WE = 0, data on lines D7 – D0 acts as **input** bus (data are written into the word addressed by A0 through A9).
- When CS = 0 and WE = 1, data on lines D7 – D0 acts as **output** bus (the contents of memory word whose address is on A9 – A0 will appear on lines D7 – D0.)
Main Memory Array Design: Linear Decoding

- This technique uses the unused address lines of the microprocessor as chip selects for the memory chip.

- A simple way to connect an 8-bit microprocessor to a 6K RAM system using linear decoding is shown in figure, where:

- Address lines A9 through A0 of the microprocessor used as common input to address lines of each 1 KB RAM chip.

- The data lines of microprocessor are connected to data lines of all memory chips.

- The remaining address lines (A10-A15) are used to select one of the chips (\( \overline{CS} \)) at a time. For example,

\[
\begin{align*}
A15 & A14 A13 A12 A11 A10 \\
0 & 0 0 0 0 1 0
\end{align*}
\]

\[\Rightarrow\] RAM Chip 2 will be selected

Fig. Linear Decoding
**Main Memory Array Design: Linear Decoding**

<table>
<thead>
<tr>
<th>Binary Address Pattern</th>
<th>Selected Device</th>
<th>Address Assignment in HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 1 0 0 0 0 0 0 0 0 0</td>
<td>RAM 1</td>
<td>0400 to 07FF</td>
</tr>
<tr>
<td>0 0 0 0 0 1 1 1 1 1 1 1 1 1 1</td>
<td>RAM 2</td>
<td>0800 to 0BFF</td>
</tr>
<tr>
<td>0 0 0 0 1 0 0 0 0 0 0 0 0 0 0</td>
<td>RAM 3</td>
<td>1000 to 13FF</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 1 1 1 1 1 1 1 1</td>
<td>RAM 4</td>
<td>2000 to 23FF</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>RAM 5</td>
<td>4000 to 43FF</td>
</tr>
<tr>
<td>0 1 0 0 0 0 1 1 1 1 1 1 1 1 1</td>
<td>RAM 6</td>
<td>8000 to 83FF</td>
</tr>
</tbody>
</table>
Main Memory Array Design: Linear Decoding

- **Advantage:-**
  - Linear decoding does not need any decoding circuit (Hardware).

- **Disadvantages:-**
  - a) With 16 bit address bus we can connect $2^{16}=65,536$ =64 KB of RAM but here we are able to interface only 6KB of RAM so, waste of address space.
  - b) Address map is not contiguous. It is sparsely distributed. (such as, 13FF+1=1400 but started from 2000, so not contiguous).
  - c) Bus conflicts occur if both A11 and A10 become high at the same time (as both RAM1 and RAM2 are selected).
  - d) If all unused address lines are not used as chip selectors for memory then these unused pins become don’t cares (can be 0 or 1). This results in foldback, meaning a memory location will have its image in memory map. For example, if A15 is don”t care, then address 0000H is same as address 8000H. It wastes memory space.
Main Memory Array Design: Fully Decoding

- The problems of bus conflict and sparse address distribution are eliminated by the use of **fully decoded** addressing technique.

- The Figure shows a organization where we used 2 to 4 decoder and interface the 8-bit microprocessor with 4 KB of RAM.

- If we observe A10 and A11, 2-to-4 decoder would be an obvious choice for CS signals. We can write the truth table as follows:

- The above hardware makes sure that the memory system enables only when the lines A15-A12 will be **Zero**. If any line becomes high memory system disable and by this way fold bask is removed. Above that, address space is not wasted since the unused lines (A15-A10) can be used in future by making use of higher decoder.
### Main Memory Array Design: Fully Decoding

#### Figure: Address map

<table>
<thead>
<tr>
<th>Binary Address Pattern</th>
<th>Device selected</th>
<th>Address Assignment in HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>RAM 1</td>
<td>0000 to 03FF</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0</td>
<td>RAM 2</td>
<td>0400 to 07FF</td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0</td>
<td>RAM 3</td>
<td>0800 to 0BFF</td>
</tr>
<tr>
<td>0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>RAM 4</td>
<td>0C00 to 0FFF</td>
</tr>
<tr>
<td>0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Main Memory Array Design: The 3-to-8 Line Decoder (74LS138)

- The truth table shows that only one of the eight outputs ever goes low at any time. For each low output the decoder must have to enabled by--,

  \( G_2A \) and \( G_2B \) inputs must both be low (logic 0) and \( G_1 \) must be high (logic 1).

- Once the 74LS138 is enabled, the address inputs (C, B and A) select which output pin goes low.
**Problem:** Interface 4KB memory to 8085 microprocessor with starting address A000H using 74LS138 decoder.

A0-A11 address lines (12 lines) are used as $2^{12} = 4KB$.

A12-A15 are used for selecting memory chips.
A15 line is used for enabling decoder chip. (A15-should be High)

When A12, A13, A14 lines are 010; output should be "0". This is provided at O2 pin of 74LS138 chip.

The range of address is found as follows:

<table>
<thead>
<tr>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
<th>A10</th>
<th>A9</th>
<th>A8</th>
<th>A7</th>
<th>A6</th>
<th>A5</th>
<th>A4</th>
<th>A3</th>
<th>A2</th>
<th>A1</th>
<th>A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</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>
</tr>
<tr>
<td>1</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>
</tr>
</tbody>
</table>
Exercise:
Problem: Find the range of address for RAM1 and RAM2 in the following figure.
Exercise:

1. Interface an 8-bit microprocessor with a 2K×8 ROM Chip and two 1K×8 Chips such that the following address map is realized:

<table>
<thead>
<tr>
<th>Device</th>
<th>Size</th>
<th>Address range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM Chip0</td>
<td>2K×8</td>
<td>0800-0FFF</td>
</tr>
<tr>
<td>ROM Chip1</td>
<td>1K×8</td>
<td>1000-13FF</td>
</tr>
<tr>
<td>ROM Chip2</td>
<td>1K×8</td>
<td>2800-2BFF</td>
</tr>
</tbody>
</table>

2. Interface an 8-bit microprocessor with a 1K×8 ROM Chip, two 512×8 RAM Chips and two RAM 256×8 Chips.
Exercise:
Problem: Considering the figure below, determine the address map of this system.
References

- Microprocessor and Interfacing Programming and Hardware; Douglas V Hall, 2nd Edition.
- Microprocessors and Microcomputer-Based System Design; Mohammed Rafiquzzaman, revised edition.
- Microprocessor & Interfacing Lecture Materials; Mohammed Abdul kader, IIUC.