Computer Interfacing

Chapter-6: External Memory Interfacing

External Memory Interfacing

Interface

- An interface is a concept that refers to a point of interaction between components, and is applicable at the level of both hardware and software.
- This allows a component, (such as a graphics card or an Internet browser), to function independently while using interfaces to communicate with other components via an input/output system and an associated protocol.

Semiconductor Memory

Semiconductor Memories are classified according to the type of data storage and the type of data access mechanism into the following two main groups.

1. **Non-volatile Memory (NVM)** also known as Read-Only Memory (ROM) which retains information when the power supply voltage is off.

   NVM are divided into the following groups:

   - **Mask programmed ROM**
     The required contents of the memory are programmed during fabrication,
   - **Programmable ROM (PROM)**
     The required contents are written in a permanent way by burning out internal interconnections (fuses). It is a one-off procedure.
   - **Erasable PROM (EPROM)**
     Data is stored as a charge on an isolated gate capacitor (“floating gate”). Data is removed by exposing the PROM to the ultraviolet light.
   - **Electrically Erasable PROM (EEPROM)**
     Also known as Flash Memory. The contents can be re-programmed by applying a suitable voltage to the EEPROM pins.

2. **Read/Write (R/W) memory** also known as Random Access Memory (RAM).

RAM are divided into two main groups:

- **Static RAM**
  Data is retained as long as there is power supply on.
- **Dynamic RAM**
  Data is stored on capacitors and requires a periodic refreshment.

The memory consists of the following basic blocks:

- The array of 1-bit memory cells,
- The row decoder which selects a single word line for a given n-bit row address a [1: n]
- The column decoder which selects a single bit line for a given m-bit column address b[1:m], and routs a 1-bit data to or from a selected memory cell.

Key Notes

- **Memory Capacity**
  - The number of bits that a semiconductor memory can store is call chip capacity.
  - Units: Kbits, Mbits.
- **Memory Organization**
  - A memory chip contains 2^x, where x is the number of address pins.
  - 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, where x is the number of address pins and y is the number of data pins on the chip.
  - **Examples of calculating memory organization**

Given, 512 bytes RAM
$512 = 2^9$
So 9 address lines are required.
Memory capacity: $512 \times 8 = 4096$ bits
Memory organization is 512×8

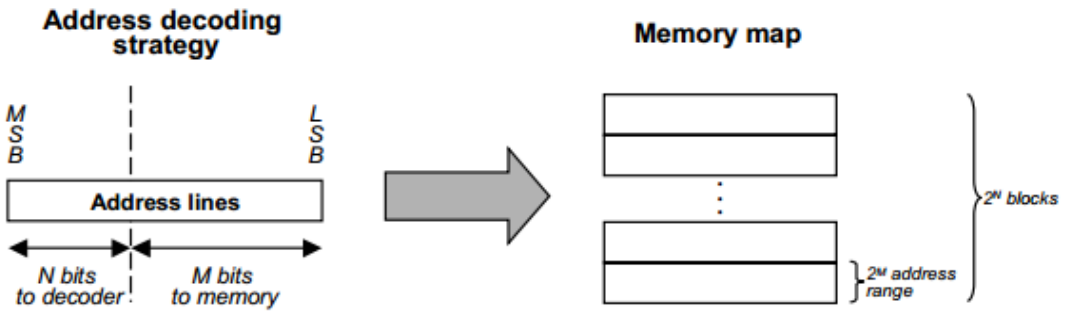
Given, 8 Kbytes RAM
$1\ K = 1024 = 2^{10}$
$8\ K = 2^3 \times 2^{10} = 2^{13}$
So 13 address lines are required.
Memory capacity: $213 \times 8 = 65{,}536$ bits
Memory organization: $8\ K \times 8$

X	**2^x**
5	32
10	1K
11	2K
13	8K
20	1M

- **Speed**
  - **Access data from memory**
    - To access data, the address is presented to the address pins.
    - READ pin is activated.
    - After a certain amount of time has elapsed, the data shows up at the data pins.
    - The speed of the memory chip commonly referred as access time.

Memory Address Decoding

- CPU provides the address of the data, but it is the job of decoding circuit to locate the selecting memory block.
- Address decoding is the process of generating chip select (CS) signals from the address bus for each device in the system.
- The address bus lines are split into two sections:
  - n the N most significant bits are used to generate the CS* signals for the different devices
  - n the M least significant signals are passed to the devices as addresses to the different memory cells or internal registers



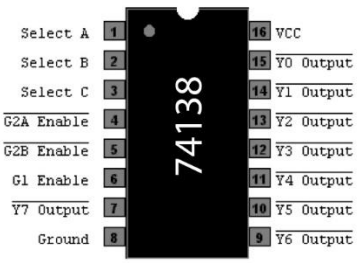
- Memory chips have one or more pins called CS (chip select). Which must be activated for the memory's contents to be accessed.
- **In connecting a memory chip to the CPU, note the following points:**
  - The data bus of the CPU is connected directly to the data pins of memory chip.
  - Control signal RD (read) and WR (memory write) from the CPU are connected to the OE (output enable) and WE (write enable) pins of the memory chip, respectively.
  - In the case of the address buses, while the lower bits of the addresses from the CPU go directly to the memory chip address pins, the upper ones are used to activate the CS pin of the memory chip.
- Address Decoding can be achieved using:
  - Combinational Logic, (AND, NAND, OR, NOR and INVERTERS)
  - Address Decoders (2 to 4, 3 to 8, etc.),
  - Programmable Logic Device:
    - Programmable Logic Array (PLA),
    - Programmable Array Logic (PAL),
    - Gated Array Logic(GAL).

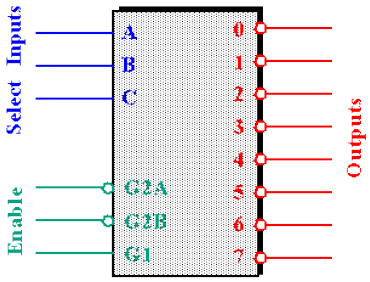
Simple NAND Gate Decoder

- When 2K X 8 EPROM is used, address connections A10 – A0 of the 8051 are connected to address inputs A10 – A0 of the EPROM.
- 2K X 8 = 2K bytes = $2 \times 2^{10} = 2^{11}$
- Remaining nine address pins (A19 – A11) are connected to the inputs of a NAND gate decoder.
- The decoders select the EPROM for one of the many 2Kbyte sections of the entire 1Mbyte address range of the 8051 microcontroller.
- A NAND gate outputs a logic 0 when all of its inputs are at logic 1. so, when A19 – A11 are all at logic 1 then a logic 0 will be applied at CE (Chip Enable) of 2716 EPROM which will enable the memory.
- For getting output from memory RD = 0 will be applied at OE (Output Enable) pin of the EPROM.
- NAND gates are rarely used to decode memory because each memory device requires its own NAND gate decoder. Because of excessive cost of the NAND gate decoder and of the inverters that are often required, this method is very expensive in the systems where large amount of memory is to be used.

The 3 to 8 Line Decoder (74LS138)

- 74LS138 (3 to 8 Line Decoder) is a common decoder used in microcontroller based systems.
- To be active, the G2A and G2B inputs must be low (logic 0), and G1 must be high (logic 1).
- Once the 74LS138 is enabled, the address inputs (C, B and A) select which output pin goes low.
- Imagine eight EPROM's CE inputs connected to the eight outputs of the decoder. This is very powerful device because it selects eight different memory devices at the same time.





Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

- Here the decoder selects eight 8Kbytes blocks of memory for a total of 64Kbytes of memory.
- All the address connections from the 8051 are connected to this circuit.
  - A12 – A0 are directly connected to 2764 EPROMs.
  - A15 – A13 selects one of EPROMs.
  - A19 – A16 enables the decoder.

Programmable Decoders (PLD)

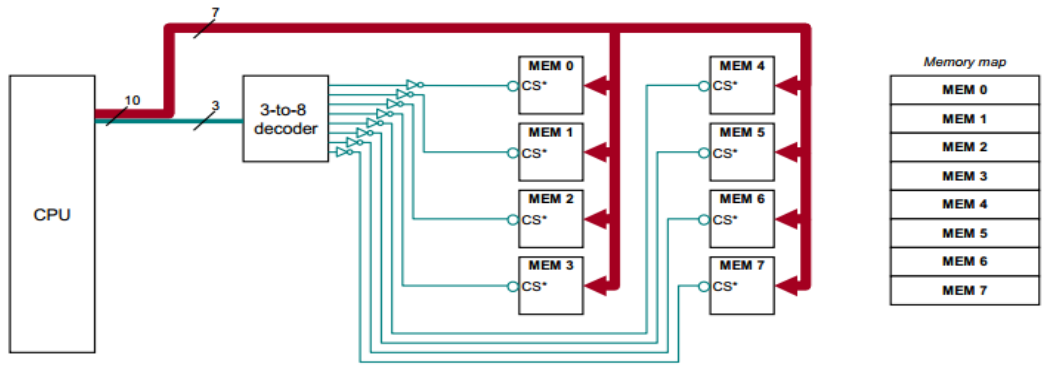
- PLD means Programmable Logic Devices.
- There are three PLD devices that function in the same manner but have different names: PLA (Programmable Logic Array), PAL (Programmable Array Logic) and GAL (Gated Array Logic).
- The PAL and the PLA are fuse-programmed as is the PROM, and some PLD devices are erasable devices (as are EPROMs).
- All the three devices are arrays of logic elements that are programmable.

Address Decoding Methods

- There are some situations where only a portion of the addressable space is going to be implemented there are two basic address decoding strategies.
- **Full address decoding**
  - All the address lines are used to specify a memory location
  - Each physical memory location is identified by a unique address
- **Partial address decoding**
  - Since not all the address space is implemented, only a subset of the address lines are needed to point to the physical memory locations
  - Each physical memory location is identified by several possible addresses (using all combinations of the address lines that were not used)

Address Decoding Examples

- Let's assume a very simple microcontroller with 10 address lines (1KB memory). We wish to implement all its memory space and we use 128x8 memory chips.

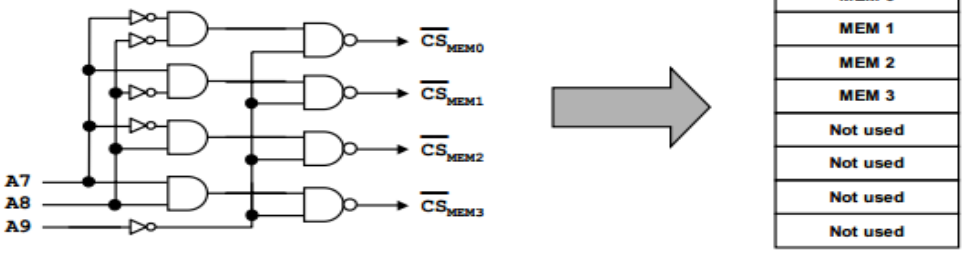


- We will need 8 memory chips (8x128=1024)
- We will need 3 address lines to select each one of the 8 chips
- Each chip will need 7 address lines to address its internal memory cells

- Let's assume the microcontroller with 10 address lines (1KB memory) However, this time we wish to implement only 512 bytes of memory

Address Decoding *on each memory IC*

Device	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
MEM 0	0	0	0	X	X	X	X	X	X	X
MEM 1	0	0	1	X	X	X	X	X	X	X
MEM 2	0	1	0	X	X	X	X	X	X	X
MEM 3	0	1	1	X	X	X	X	X	X	X



- We still must use 128-byte memory chips
- Physical memory must be placed on the upper half of the memory map