Memory Interface

Allmost all systems contain two main types of memory :

- read-only memory (ROM) → system software and permanent system data
- random access memory (RAM) or read/write → memory temporary data and application software

Memory Devices

- Types of Memory :
  - read-only memory (ROM)
  - flash memory (EEPROM)
  - static random access memory (SRAM)
  - dynamic random access memory (DRAM)
- All memory devices have :
  - address inputs and outputs, or just outputs
  - a pin for selection
  - one or more pins that control the operation of the memory

Memory Devices (cont'd)

- Memory Pin Connections
  - **Address Connections**
    - **are used to select one of the memory location within the device.**
  - **Data Connections**
    - **are used to enter information to be stored in a memory location and also to retrieve information read from a memory location.**
    - **Manufacturers list their memory as, for example, 4K x 4, which means that the device has 4K memory locations (4096) and 4-bits are stored in each location.**

Memory Devices (cont'd)

- Selection Connections
  - Memory selection is accomplished via a chip selection pin (CS) on many RAMs or a chip enable pin (CE) on many EPROM or ROM memories.
- Control Connections
  - Memory function is selected by an output enable pin (OE), for reading data.
  - For writing data, memory function is selected by the write enable pin (WE).

Memory Devices (cont'd)

- **ROM Memory**

ROM permanently stores programs and data that are resident to the system and must not change when power is disconnected. This type of memory is often called *nonvolatile memory*, for example :

- EPROM (erasable programmable read-only memory) is programmed by an EPROM programmer and can be erased if exposed to ultraviolet light.

Memory Devices (cont'd)

  - The flash memory (EEPROM) is programmed in the system by using a 12V programming pulse.

- Static RAM (SRAM) Devices
  - retains data for as long as the system power system is attached.
  - these memory types are available in sizes up to 128K x 8
  - stores temporary data and is used when the size of the read/write memory is relatively small
  - Fig. 9-4 illustrates the 4016 SRAM, which is a 2K x 8 read/write memory

Memory Devices (cont'd)

- Dynamic RAM (DRAM) Memory
  - the size is up to 16M x 1
  - DRAM = SRAM, except that it retains data for only 2 or 4 ms on an integrated capasitor (see Fig. 9-7)
  - Another disanvantage of DRAM memory is that it requires so many address pins that the manufacturers have multiplexed the address inputs

Memory Devices (cont’d)

- as with SRAM, the $\overline{R}$/W pin writes data to the DRAM when a logic = 0, but there is no pin labeled G or enable. There also is no S (select) input to the DRAM

Address Decoding

To attach a memory device to the microprocessor → to decode the address from the microprocessor to make the memory function at a unique section or partition of the memory map.

- Why Decode Memory?
  - To corrects the mismatch between microprocessor and memory component (because of a difference in the number of address connections surfaces)

Address Decoding(cont'd)

- Commonly found address decoders include :
  - the 3-to-8 Line Decoder (74LS138)(see Fig. 9-13)
  - the 74LS139 2-to-4 line decoder (see Fig. 9-15)
  - programmed selection logic in the form of a PROM (the 82S147) or PLD (see Fig. 9-17)

8088 and 80188 (8-Bit) Memory Interface

- Basic 8088/80188 (8-bit) Memory Interface
  - Both has an 8-bit data bus
  - ideal as a simple controller
  - the memory system must decode the address to select a memory component
  - it must use the RD, WR and IO/M control signals provided by the 8088/80188 to control the memory system

8088 and 80188 (8-Bit) Memory Interface (cont'd)

- is used the maximum mode system :
  - IO/$\overline{\text{M}}$ is combined with $\overline{\text{RD}}$ to generate an $\overline{\text{MRDC}}$ signal
  - IO/$\overline{\text{M}}$ is combined with $\overline{\text{WR}}$ to generate an $\overline{\text{MWTC}}$ signal
  - these maximum mode control signals are developed inside the 8288 bus controller
- The minimum mode, the memory sees the 8088 or the 80188 as a device with 20 address connections (A_{19} - A_0), 8 data bus connections (AD_7 - AD_0) and control signals IO/$\overline{\text{M}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$

8088 and 80188 (8-Bit) Memory Interface (cont’d)

Interfacing EPROM to the 8088

- Fig. 9-19 illustrates an 8088 microprocessor connected to eight 2732 EPROMs, 4K x 8 memory devices
- the 2732 has one more address input (A_{11}) than the 2716 and twice the memory
- the decoder (74LS138) must add a NAND gate to generate a signal to enable the decoder and a signal for the wait state generator

8088 and 80188 (8-Bit) Memory Interface (cont'd)

- The 8088 will allow 660 ns for the EPROM to access data. The 660 ns is ample time for a 450 ns memory component to access data, even with the delays introduced by the decoder and any buffers added to the data bus

Interfacing RAM to the 8088

- RAM is a little easier to interface than EPROM (it doesn't require wait states)
- it contains vectors for interrupts

8088 and 80188 (8-Bit) Memory Interface (cont'd)

- Fig. 9-20, 16 62256 32K x 8 static RAMs are interfaced to the 8088, beginning at memory location 0000H
- the circuit board uses two decoders to select the sixteen different RAM memory components and a third to select the other decoders for the appropriate memory sections
- the first decoder (U_4) in this circuit selects the other two decoders

8088 and 80188 (8-Bit) Memory Interface (cont'd)

- Interfacing Flash Memory
  - for storing setup information on video cards
  - for storing the system BIOS in the PC
  - the only difference between a flash memory device and SRAM is that the flash memory device requires a 12V programming voltage to erase and write new data
  - Fig. 9-21 illustrates a 28F400 Intel Flash Memory interfaced to the 8088 microprocessor

8088 and 80188 (8-Bit) Memory Interface (cont’d)

- Parity for Memory Error Detection
  - Parity checking counts the number of 1’s in data and indicates whether there is an even or odd number
  - if all data are stored with even parity (with an even number of 1-bits), a 1-bit error can be detected
  - Fig. 9-22 illustrates the 74AS280 parity generator/detector integrated circuit

8088 and 80188 (8-Bit) Memory Interface (cont’d)

  - Fig. 9-23 illustrates a 64K memory system that containts a parity error detection circuit
- Error Correction
  - one such circuit is the 74LS636, an 8-bit error correction and detection circuit that corrects any single-bit memory read error and flags any 2-bit error
  - it corrects errors by storing five parity bits with each byte of memory data

8088 and 80188 (8-Bit) Memory Interface (cont’d)

- Fig. 9-24 depicts the pin-oit of 74LS636
- when a single error is detected, the 74LS636 goes through an error correction cycle: it places a 01 on S_0 and S_1 by causing a wait and then a read following error correction
- Fig. 9-25 illustrates an error detection and correction circuit using the 74LS636