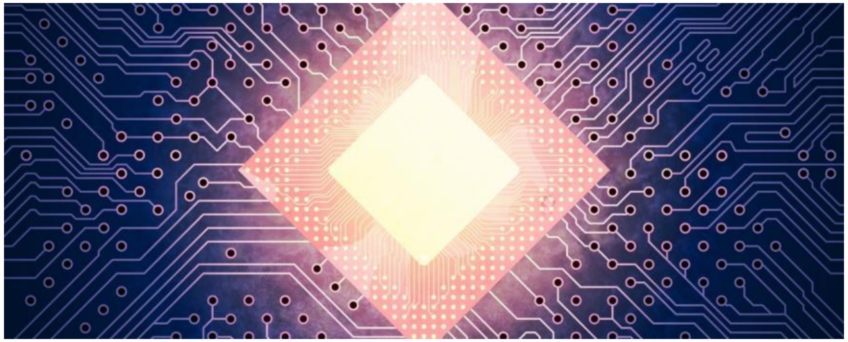


3

ET2223 Microprocessors Microcontrollers & Embedded Systems

Interfacing Techniques and Circuits

Sandali Goonatilleke

Department of Computer Engineering
General Sir John Kotelawala Defence University

Interfacing

1. **What is Interfacing?**
2. **What is the aim or purpose of interfacing?**

Interfacing

Interface is the path for communication between two components

Interfacing is of two types;

- Memory Interfacing
- I/O interfacing

Wires Vs Busses

Wires

- A single strand of metal that carries power or data from one location to another location
- Uni-directional or bi-directional
- One line may represent multiple wires

Wires Vs Busses

Bus

- In computer architecture, a bus is a communication system that transfers data between components inside a computer, or between computers
- A set of wires with a single function

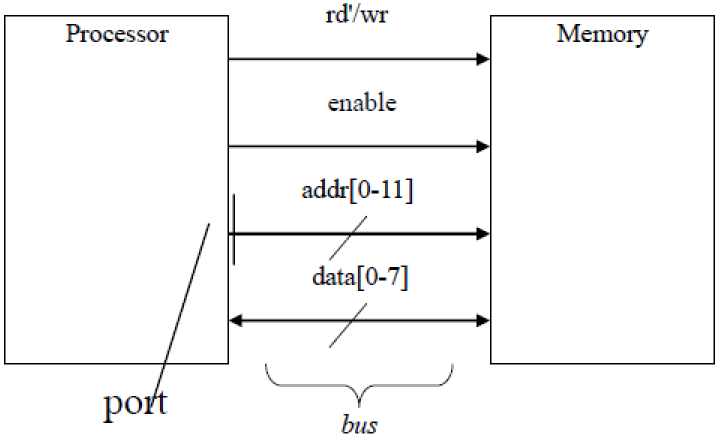
- Address Bus

- Data Bus

- Control Bus

Ports

- Conducting device on periphery
- Connects bus to processor or memory
- Often referred to as a pin



Memory Interfacing

- When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory
- For this, both the memory and the microprocessor requires some signals to **read from** and **write to** **registers**

Memory Interfacing

- The interfacing process includes some key factors to match with the memory requirements and microprocessor signals
- The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor

Memory Interfacing in 8085

Memory Structure and its Requirements

- Read/write memories consist of an array of registers
- Each register has unique address
- **The size of the memory is N x M**, where N is the number of registers and M is the word length, in number of bits

Memory Interfacing in 8085

Basic Concepts in Memory Interfacing

1. Microprocessor 8085 can access 64Kbytes memory and address bus is 16-bit. But it is not always necessary to use full 64Kbytes address space. The total memory size depends upon the application

2. Generally, EPROM(s) is used as a program memory and RAM(s) as a data memory. When both, EPROM and RAM are used, the total address space 64Kbytes is shared by them

Memory Interfacing in 8085

Basic Concepts in Memory Interfacing

3. The capacity of program memory and data memory depends on the application.

4. It is not always necessary to select 1 EPROM and 1 RAM. Can have multiple EPROMs and multiple RAMs as per the requirement of application.

Memory Interfacing in 8085

Basic Concepts in Memory Interfacing

5. Can place EPROM/RAM anywhere in full 64 Kbytes address space. But program memory (EPROM) should be located from address 0000H since reset address of 8085 microprocessor is 0000H.

6. It is not always necessary to locate EPROM and RAM in consecutive memory.

Memory Interfacing in 8085

The memory interfacing requires to;

- Select the chip
- Identify the register
- Enable the appropriate buffer

Memory Interfacing in 8085

- Microprocessor system contains memory devices and I/O devices
- **It is important to note that microprocessor can communicate (read/write) with only one device at a time**, since the data, address and control buses are common for all the devices
- In order to communicate with memory or I/O devices, it is necessary to decode the address from the microprocessor
- Due to this each device (memory or I/O) can be accessed independently

Memory Interfacing in 8085

Address Decoding Techniques;

- Absolute decoding/Full Decoding
- Linear decoding/Partial Decoding

Memory Interfacing in 8085

Address Decoding Techniques;

1. Absolute Decoding

- All the higher address lines are decoded to select the memory chip
- Memory chip is selected only for the specified logic levels on these high-order address lines; no other logic levels can select the chip

Memory Interfacing in 8085

Address Decoding Techniques;

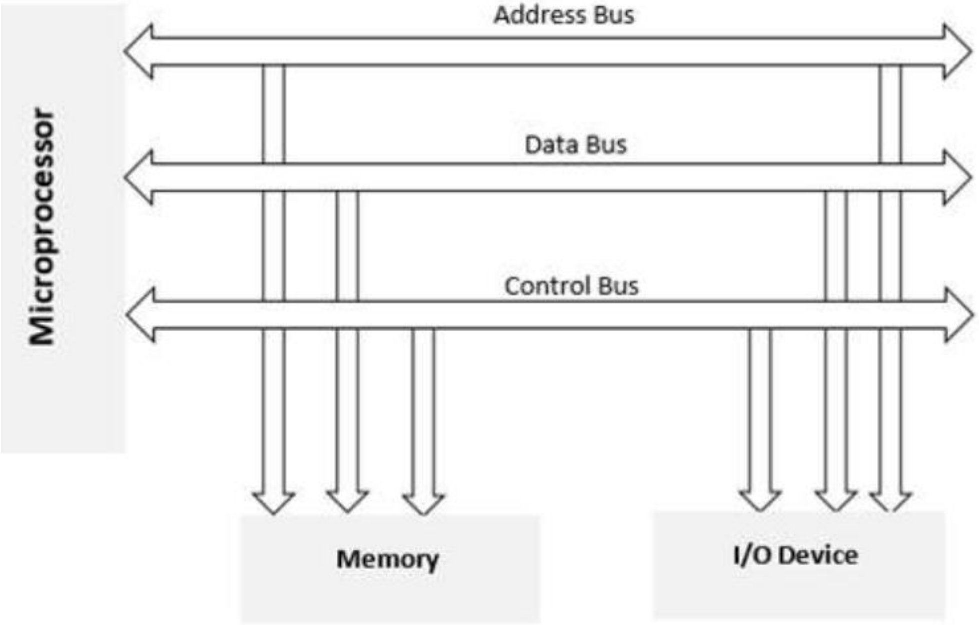
2.Linear Decoding

- In small systems, hardware for the decoding logic can be eliminated by using individual high-order address lines to select memory chips
- Also called partial decoding
- Reduces the cost of decoding circuit, but it has a drawback of multiple addresses (shadow addresses)

I/O Interfacing

- There are various communication devices like the keyboard, mouse, printer etc.
- So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers
- This type of interfacing is known as **I/O interfacing**

Memory Interfacing and I/O Interfacing



Block Diagram of Memory and I/O Interfacing

8279 Programmable Keyboard

- The Intel 8279 is a programmable keyboard interfacing device
- 8279 is designed to interfaces a keyboard with the CPU
- The keyboard first scans the keyboard and identifies if any key has been pressed
- It then sends their relative response of the pressed key to the CPU and vice-a-versa
- 8279 has been designed for the purpose of 8-bit Intel microprocessors

8279 Programmable Keyboard

- 8279 has two sections namely **keyboard section** and **display section**
- The function of the **keyboard section** is to interface the keyboard which is used as input device for the microprocessor
- It can also interface toggle or thumb switches
- The purpose of the **display section** is to drive alphanumeric displays or indicator lights
- It is directly connected to the microprocessor bus

8279 Programmable Keyboard

- The Keyboard can be interfaced either in the **interrupt** or the **polled mode**
- In the **Interrupt mode**, the processor is requested service only if any key is pressed, otherwise the CPU will continue with its main task
- In the **Polled mode**, the CPU periodically reads an internal flag of 8279 to check whether any key is pressed or not with key pressure

8279 Programmable Keyboard Features

- Simultaneous keyboard display operations
- Scanned sensor mode
- Scanned keyboard mode
- 8-character keyboard FIFO
- Strobed input entry mode
- 2-key lock out or N-key roll over with contact debounce
- Single 16-charcter display
- Dual 8 or 16 numerical display
- Interrupt output on key entry
- Programmable scan timing and mode programmable from CPU

8257 DMA Controller

- The data transfer between I/O devices and memory through the accumulator is a time consuming process
- For this situation, the **Direct Memory Access (DMA)** technique is preferred
- In DMA data transfer scheme, data is directly transferred from an I/O device to RAM or from RAM to an I/O device

8257 DMA Controller

- Using a DMA controller, the device requests the CPU to hold its address, data and control bus
- Accordingly, the device becomes free to transfer data directly to/from the memory
- The DMA data transfer is initiated only after receiving HLDA signal from the CPU

How DMA Operations are Performed

- The device send DMA request (DRQ) to DMA controller for sending the data between the device and the memory
- The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU for the HLDA
- When CPU gets the HLDA signal then, it leaves the control over the bus and acknowledges the HOLD request through HLDA signal

How DMA Operations are Performed

- Now the CPU is in the HOLD state and the DMA controller has to manage the operations over the buses between the CPU, memory and I/O devices

Intel 8257

- The **Intel 8257 is a programmable DMA controller**
- It is a 4 channel programmable **Direct Memory Access (DMA)** controller
- It is a 40 pin IC package and requires +5V supply for its operation
- Can perform three main operations; read, write, and verify
- Each channel incorporates two 16-bit registers, namely DMA address register and byte count register

Intel 8257

- Each channel can transfer data up to 64KB and can be programmed independently
- It operates in 2 modes; Master mode and Slave mode

Intel 8257

Exercise 1: Explore the architecture of Intel 8257

Exercise 2: Draw the pin configuration of Intel 8257

Exercise 3: Describe each pin of the controller