

## EC010 506 MICROPROCESSORS AND APPLICATIONS

### Course Notes

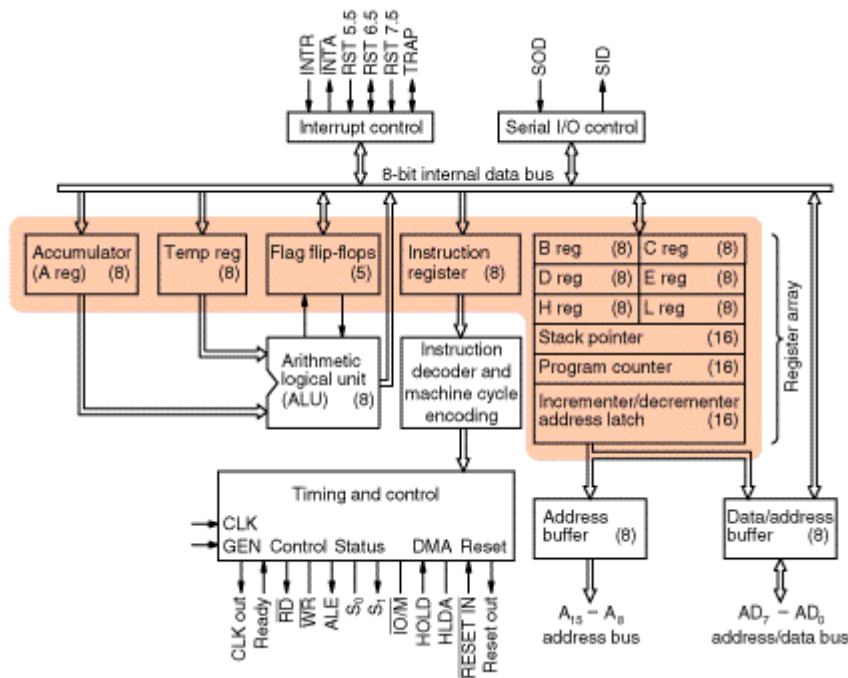
#### Module I (12hours)

Introduction to microprocessors and microcomputers: Function of microprocessors- organisation of a microprocessor based system – microprocessor architecture and its operations – memory – I/O devices - pin configuration and functions of 8085 – tristate bus concept - control signals– de-multiplexing AD<sub>0</sub>-AD<sub>7</sub> – flags - memory interfacing - I/O addressing - I/O mapped I/O - memory mapped I/O schemes - instruction execution - fetch/execute cycle - instruction timings and operation status.

#### Introduction to microprocessors and microcomputers

Microprocessors are regarded as one of the most important devices in our everyday machines called computers. Before we start, we need to understand what exactly microprocessors are and their appropriate implementations. **Microprocessor** is an electronic circuit that functions as the central processing unit (CPU) of a computer, providing computational control. Microprocessors are also used in other advanced electronic systems, such as computer printers, automobiles, and jet airliners

#### Function of microprocessors and organisation of a microprocessor based system



## Control Unit

Generates signals within uP to carry out the instruction, which has been decoded. In reality causes certain connections between blocks of the uP to be opened or closed, so that data goes where it is required, and so that ALU operations occur.

## Arithmetic Logic Unit

The ALU performs the actual numerical and logic operation such as 'add', 'subtract', 'AND', 'OR', etc. Uses data from memory and from Accumulator to perform arithmetic. Always stores result of operation in Accumulator.

## Registers

The 8085/8080A-programming model includes six registers, one accumulator, and one flag register, as shown in Figure. In addition, it has two 16-bit registers: the stack pointer and the program counter. They are described briefly as follows.

The 8085/8080A has six general-purpose registers to store 8-bit data; these are identified as B,C,D,E,H, and L as shown in the figure. They can be combined as register pairs - BC, DE, and HL - to perform some 16-bit operations. The programmer can use these registers to store or copy data into the registers by using data copy instructions.

## Accumulator

The accumulator is an 8-bit register that is a part of arithmetic/logic unit (ALU). This

register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator. The accumulator is also identified as register A.

### Flags

The ALU includes five flip-flops, which are set or reset after an operation according to data conditions of the result in the accumulator and other registers. They are called Zero(Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags; they are listed in the Table and their bit positions in the flag register are shown in the Figure below. The most commonly used flags are Zero, Carry, and Sign. The microprocessor uses these flags to test data conditions.

For example, after an addition of two numbers, if the sum in the accumulator is larger than eight bits, the flip-flop used to indicate a carry -- called the Carry flag (CY) -- is set to one. When an arithmetic operation results in zero, the flip-flop called the Zero(Z) flag is set to one. The first Figure shows an 8-bit register, called the flag register, adjacent to the accumulator. However, it is not used as a register; five bit positions out of eight are used to store the outputs of the five flip-flops. The flags are stored in the 8-bit register so that the programmer can examine these flags (data conditions) by accessing the register through an instruction.

These flags have critical importance in the decision-making process of the microprocessor.

The conditions (set or reset) of the flags are tested through the software instructions. For example, the instruction JC (Jump on Carry) is implemented to change the sequence of a program when CY flag is set. The thorough understanding of flag is essential in writing assembly language programs.

### Program Counter (PC)

This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer. Memory locations have 16-bit addresses, and that is why this is a 16-bit register.

The microprocessor uses this register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location

### Stack Pointer (SP)

The stack pointer is also a 16-bit register used as a memory pointer. It points to a

memory location in R/W memory, called the stack. The beginning of the stack is defined by loading 16-bit address in the stack pointer. The stack concept is explained in the chapter "Stack and Subroutines."

#### Instruction Register/Decoder

Temporary store for the current instruction of a program. Latest instruction sent here from memory prior to execution. Decoder then takes instruction and 'decodes' or interprets the instruction. Decoded instruction then passed to next stage.

#### Memory Address Register

Holds address, received from PC, of next program instruction. Feeds the address bus with addresses of location of the program under execution.

#### Control Generator

Generates signals within uP to carry out the instruction which has been decoded. In reality causes certain connections between blocks of the uP to be opened or closed, so that data goes where it is required, and so that ALU operations occur.

#### Register Selector

This block controls the use of the register stack in the example. Just a logic circuit which switches between different registers in the set will receive instructions from Control Unit.

#### General Purpose Registers

uP requires extra registers for versatility. Can be used to store additional data during a program. More complex processors may have a variety of differently named registers.

#### Microprogramming

How does the  $\mu$ P know what an instruction means, especially when it is only a binary number? The microprogram in a uP/uC is written by the chip designer and tells the uP/uC the meaning of each instruction uP/uC can then carry out operation.

## **2. 8085 System Bus**

Typical system uses a number of busses, collection of wires, which transmit binary numbers, one bit per wire. A typical microprocessor communicates with memory and other devices (input and output) using three busses: Address Bus, Data Bus and Control Bus.

#### Address Bus

One wire for each bit, therefore 16 bits = 16 wires. Binary number carried alerts memory to 'open' the designated box. Data (binary) can then be put in or taken out. The Address Bus consists of 16 wires, therefore 16 bits. Its "width" is 16 bits. A

16 bit binary number allows 2<sup>16</sup> different numbers, or 32000 different numbers, ie 0000000000000000 up to 1111111111111111. Because memory consists of boxes, each with a unique address, the size of the address bus determines the size of memory, which can be used. To communicate with memory the microprocessor sends an address on the address bus, eg 0000000000000011 (3 in decimal), to the memory. The memory then selects box number 3 for reading or writing data. Address bus is unidirectional, ie numbers only sent from microprocessor to memory, not other way.

**Question?:** If you have a memory chip of size 256 kilobytes (256 x 1024 x 8 bits), how many wires does the address bus need, in order to be able to specify an address in this memory? **Note:** the memory is organized in groups of 8 bits per location, therefore, how many locations must you be able to specify?

#### Data Bus

Data Bus: carries 'data', in binary form, between  $\mu$ P and other external units, such as memory. Typical size is 8 or 16 bits. Size determined by size of boxes in memory and  $\mu$ P size helps determine performance of  $\mu$ P. The Data Bus typically consists of 8 wires. Therefore, 28 combinations of binary digits. Data bus used to transmit "data", ie information, results of arithmetic, etc, between memory and the microprocessor. Bus is bi-directional. Size of the data bus determines what arithmetic can be done. If only 8 bits wide then largest number is 11111111 (255 in decimal). Therefore, larger numbers have to be broken down into chunks of 255. This slows microprocessor. Data Bus also carries instructions from memory to the microprocessor. Size of the bus therefore limits the number of possible instructions to 256, each specified by a separate number.

#### Control Bus

Control Bus are various lines which have specific functions for coordinating and controlling  $\mu$ P operations. Eg: Read/NotWrite line, single binary digit. Control whether memory is being 'written to' (data stored in mem) or 'read from' (data taken out of mem) 1 = Read, 0 = Write. May also include clock line(s) for timing/synchronising, 'interrupts', 'reset' etc. Typically  $\mu$ P has 10 control lines. Cannot function correctly without these vital control signals.

The Control Bus carries control signals partly unidirectional, partly bi-directional. Control signals are things like "read or write". This tells memory that we are either **reading from** a location, specified on the address bus, or **writing to** a location specified. Various other signals to control and coordinate the operation of the system.

Modern day microprocessors, like 80386, 80486 have much larger busses. Typically 16 or 32 bit busses, which allow larger number of instructions, more memory location, and faster arithmetic. Microcontrollers organized along same lines, except: because microcontrollers have memory etc inside the chip, the busses may all be internal. In the microprocessor the three busses are external to the chip (except for the internal data bus). In case of external busses, the chip connects to the busses via buffers, which are simply an electronic connection between external bus and the internal data bus.

### **3. 8085 Pin description.**

Properties

Single + 5V Supply

4 Vectored Interrupts (One is Non Maskable)

Serial In/Serial Out Port

Decimal, Binary, and Double Precision Arithmetic

Direct Addressing Capability to 64K bytes of memory

The Intel 8085A is a new generation, complete 8 bit parallel central processing unit (CPU). The 8085A uses a multiplexed data bus. The address is split between the 8bit address bus and the 8bit data bus. Figures are at the end of the document.

Pin Description

The following describes the function of each pin:

A6 - A1s (Output 3 State)

Address Bus; The most significant 8 bits of the memory address or the 8 bits of the I/O address, 3 stated during Hold and Halt modes.

AD0 - 7 (Input/Output 3state)

Multiplexed Address/Data Bus; Lower 8 bits of the memory address (or I/O address) appear on the bus during the first clock cycle of a machine state. It then becomes the data bus during the second and third clock cycles. 3 stated during Hold and Halt modes.

ALE (Output)

Address Latch Enable: It occurs during the first clock cycle of a machine state and enables the address to get latched into the on chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. ALE can also be used to strobe the status information. ALE is never 3stated.

SO, S1 (Output)

Data Bus Status. Encoded status of the bus cycle:

S1 S0

0 0 HALT

0 1 WRITE

1 0 READ

1 1 FETCH

S1 can be used as an advanced R/W status.

RD (Output 3state)

RD; indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer.

WR (Output 3state)

WR; indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of WR. 3stated during Hold and Halt modes.

READY (Input)

If Ready is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If Ready is low, the CPU will wait for Ready to go high before completing the read or write cycle.

HOLD (Input)

HOLD; indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request, will relinquish the use of buses as soon as the completion of the current machine cycle. Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, RD, WR, and IO/M lines are 3stated.

### HLDA (Output)

HOLD ACKNOWLEDGE; indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle. HLDA goes low after the Hold request is removed. The CPU takes the buses one half clock cycle after HLDA goes low.

### INTR (Input)

INTERRUPT REQUEST; is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

### INTA (Output)

INTERRUPT ACKNOWLEDGE; is used instead of (and has the same timing as) RD during the Instruction cycle after an INTR is accepted. It can be used to activate the 8259 Interrupt chip or some other interrupt port.

### RST 5.5

### RST 6.5 - (Inputs)

### RST 7.5

RESTART INTERRUPTS; These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted.

RST 7.5 ~~ Highest Priority

RST 6.5

RST 5.5 o Lowest Priority

The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR.

### TRAP (Input)

Trap interrupt is a nonmaskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or Interrupt Enable. It has the highest priority of



any interrupt.

#### RESET IN (Input)

Reset sets the Program Counter to zero and resets the Interrupt Enable and HLDA flipflops. None of the other flags or registers (except the instruction register) are affected. The CPU is held in the reset condition as long as Reset is applied.

#### RESET OUT (Output)

Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.

#### X1, X2 (Input)

Crystal or R/C network connections to set the internal clock generator. X1 can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.

#### CLK (Output)

Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X1, X2 input period.

#### IO/M (Output)

IO/M indicates whether the Read/Write is to memory or I/O. Tristated during Hold and Halt modes.

#### SID (Input)

Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

#### SOD (output)

Serial output data line. The output SOD is set or reset as specified by the SIM instruction.

Vcc

+5 volt supply.

V<sub>ss</sub>

Ground Reference.

### **8085 Functional Description**

The 8085A is a complete 8 bit parallel central processor. It requires a single +5 volt supply. Its basic clock speed is 3 MHz thus improving on the present 8080's performance with higher system speed. Also it is designed to fit into a minimum system of three IC's: The CPU, a RAM/ IO, and a ROM or PROM/IO chip.

The 8085A uses a multiplexed Data Bus. The address is split between the higher 8bit Address Bus and the lower 8bit Address/Data Bus. During the first cycle the address is sent out. The lower 8bits are latched into the peripherals by the Address Latch Enable (ALE). During the rest of the machine cycle the Data Bus is used for memory or I/O data.

The 8085A provides RD, WR, and IO/Memory signals for bus control. An Interrupt Acknowledge signal (INTA) is also provided. Hold, Ready, and all Interrupts are synchronized. The 8085A also provides serial input data (SID) and serial output data (SOD) lines for simple serial interface.

In addition to these features, the 8085A has three maskable, restart interrupts and one non-maskable trap interrupt. The 8085A provides RD, WR and IO/M signals for Bus control.

#### Status Information

Status information is directly available from the 8085A. ALE serves as a status strobe.

The status is partially encoded, and provides the user with advanced timing of the type of bus transfer being done. IO/M cycle status signal is provided directly also.

Decoded So, S1 Carries the following status information:

HALT, WRITE, READ, FETCH

S1 can be interpreted as R/W in all bus transfers. In the 8085A the 8 LSB of address are multiplexed with the data instead of status. The ALE line is used as a strobe to enter the lower half of the address into the memory or peripheral address latch. This also frees extra pins for expanded interrupt capability.

#### Interrupt and Serial I/O

The 8085A has 5 interrupt inputs: INTR, RST5.5, RST6.5, RST 7.5, and TRAP. INTR is identical in function to the 8080 INT. Each of the three RESTART inputs, 5.5, 6.5.

7.5, has a programmable mask. TRAP is also a RESTART interrupt except it is nonmaskable.

The three RESTART interrupts cause the internal execution of RST (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The non-maskable TRAP causes the internal execution of a RST independent of the state of the interrupt enable or masks. The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending as follows: TRAP highest priority, RST 7.5, RST 6.5, RST 5.5, INTR lowest priority This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt a RST 7.5 routine if the interrupts were re-enabled before the end of the RST 7.5 routine. The TRAP interrupt is useful for catastrophic errors such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both edge and level sensitive.

#### Basic System Timing

The 8085A has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8bits of address on the Data Bus. Figure 2 shows an instruction fetch, memory read and I/ O write cycle (OUT). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address. As in the 8080, the READY line is used to extend the read and write pulse lengths so that the 8085A can be used with slow memory. Hold causes the CPU to relinquish the bus when it is through with it by floating the Address and Data Buses.

#### System Interface

8085A family includes memory components, which are directly compatible to the 8085A CPU. For example, a system consisting of the three chips, 8085A, 8156, and 8355 will have the following features:

- 2K Bytes ROM
- 256 Bytes RAM
- 1 Timer/Counter
- 4 8bit I/O Ports
- 1 6bit I/O Port
- 4 Interrupt Levels
- Serial In/Serail Out Ports

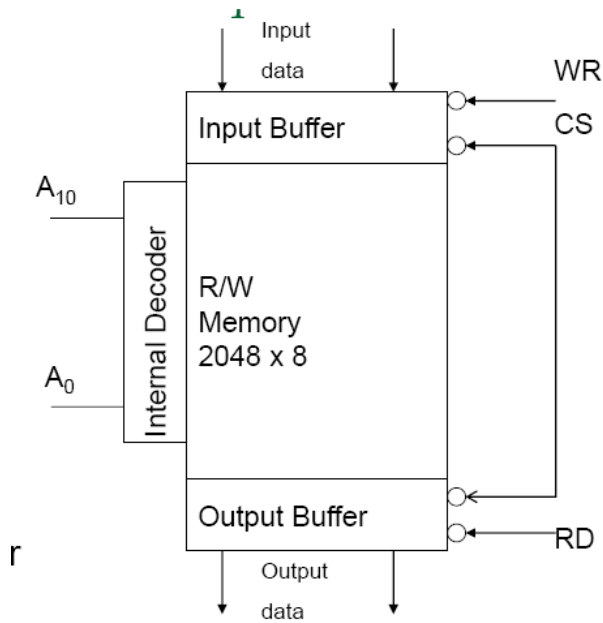
In addition to standard I/O, the memory mapped I/O offers an efficient I/O addressing technique. With this technique, an area of memory address space is assigned for I/O address, thereby, using the memory address for I/O manipulation. The 8085A CPU can also interface with the standard memory that does not have the multiplexed address/data bus.

### **Memory Interfacing**

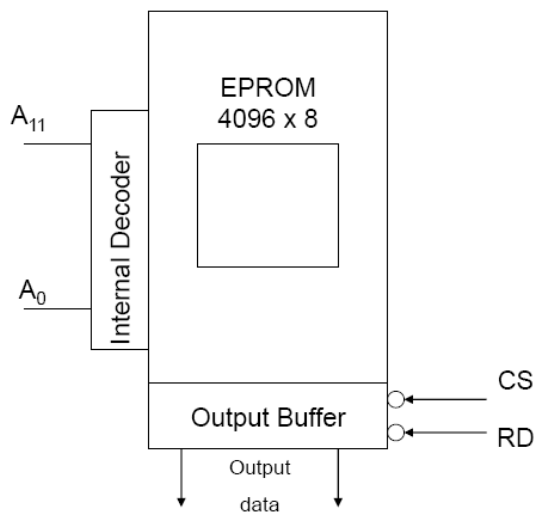
Memory are made up of (**registers**).

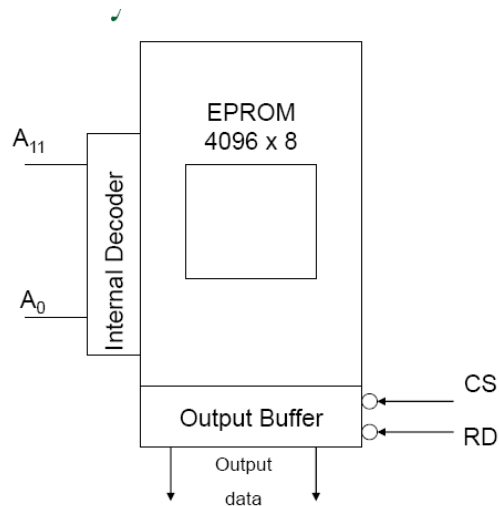
- Each register consists of one storage location
- Each location consists of an **address**
- The number of storage locations from few hundreds to several mega or giga locations
- The total number of memory storage is called memory **capacity** and measured in Bytes
- Each register consists of storage element (FF, capacitor for semiconductor)
- A storage element is called **cell**
- The data could be read from or written to memory

Memory Structure and its requirements



If memory is having 12 address lines and 8 data lines, then Number of registers/  
 memory locations (capacity) =  $2^N = 2^{12} = 4096$   
 Word length = M bit  
 = 8 bit





### **I/O addressing - I/O mapped I/O - memory mapped I/O schemes**

Memory mapping for IO and memory are of two types

#### Memory Mapped I/O

- 16-bit device address
- Data transfer between any general-purpose register and I/O port.
- The memory map (64K) is shared between I/O device and system memory.
- More hardware is required to decode 16-bit address
- Arithmetic or logic operation can be directly performed with I/O data
- separate locations are allotted for memory and IO

#### IO mapped IO

- IO is treated as part of memory
- 8-bit device address
- Data is transfer only between accumulator and I.O port
- The I/O map is independent of the memory map; 256 input device and 256. output device can be connected
- Less hardware is required to decode 8-bit address
- Arithmetic or logical operation cannot be directly performed with I/O data