

INTEL 8085 Microprocessor Architecture

Dr. Girdhari Singh

Introduction

- The microprocessor - semiconductor device consisting of electronic logic circuits manufactured through either a large-scale integration (**LSI**) or very-large-scale integration (**VLSI**) technique.
- It contains registers, Arithmetic unit, logic unit, flip-flops and timing and control circuits.

Introduction to 8085



- Introduced in 1976-77.
- It is an **8-bit MP (μP)**
- It is a 40 pin dual-in-line chip.
- It uses a single +5V supply for its operations.
- Its clock speed is 3MHz.

General Architecture- Microprocessor

- 8085 microprocessor works using **Von Neumann Architecture** (in which instructions and data are stored in the same memory, unlike **Harvard architecture**, which uses two separate memories for storing instructions and data). The architecture in which the same memory is used for storing the program as well as data. “**stored program execution**”.
- In this architecture - **the processor fetches instructions from the memory – decodes it - generates appropriate control signals and finally the instruction is executed.**
- The program is stored consecutively in the memory locations.

MPU Communication and Bus Timing

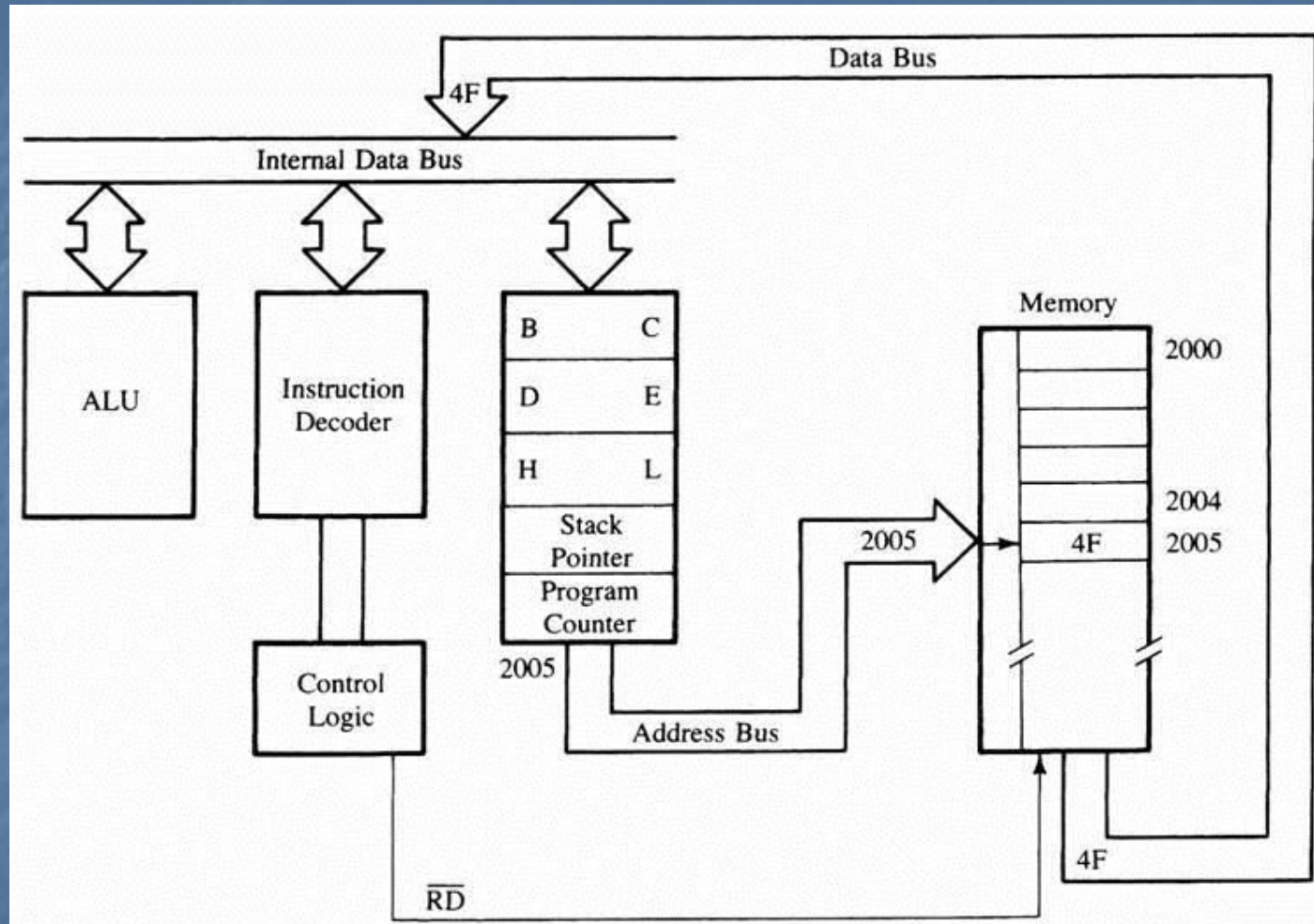


Figure 3: Moving data form memory to MPU using instruction **MOV C, A** (code machine 4FH = 0100 1111)

General Architecture – Microprocessor Contd..

- The execution steps - repeated for all the instructions of the program until the last instruction.
- The data required may either be taken from memory or from input ports - results - will be stored in the memory or given out through output ports.

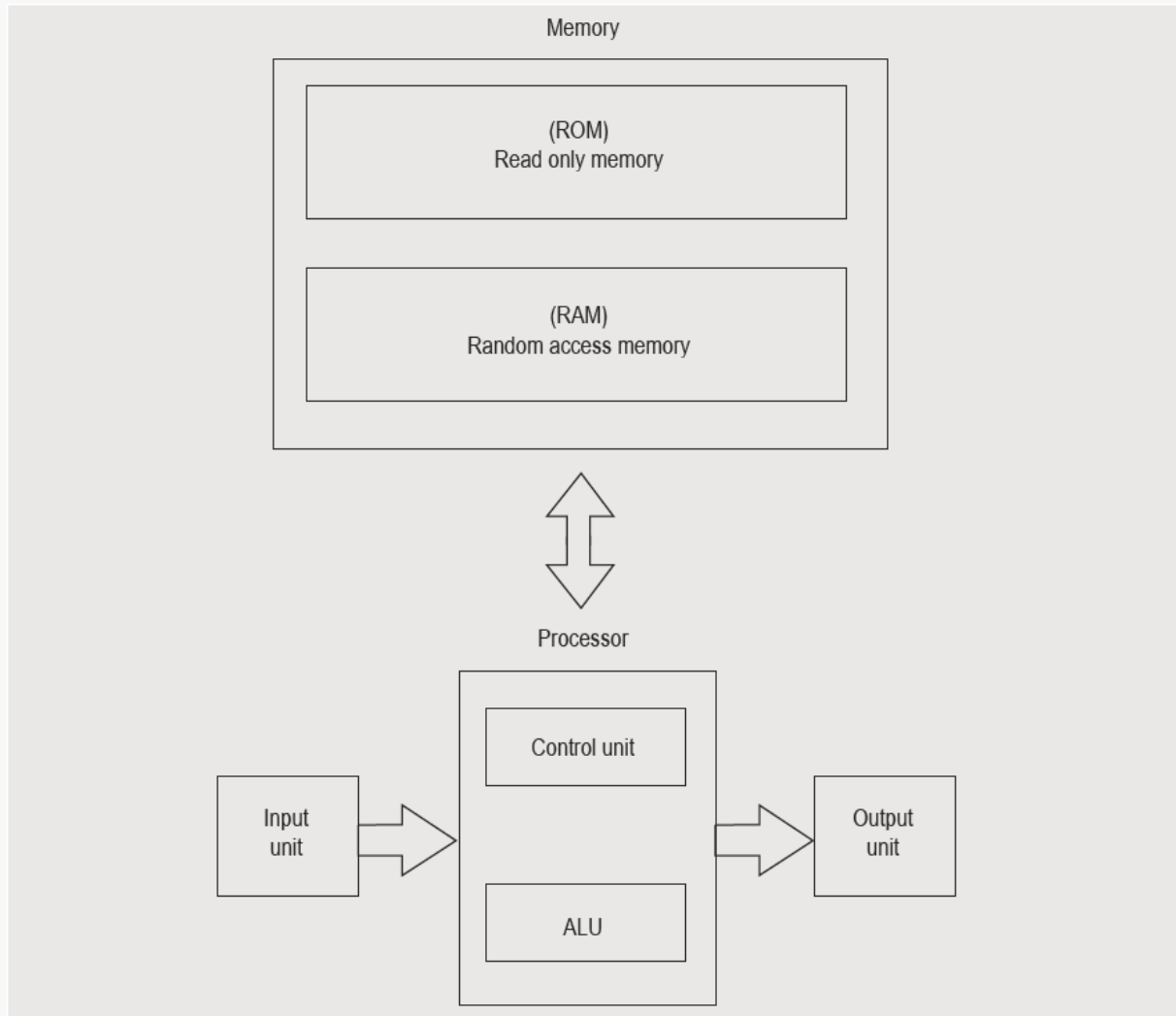
Program Execution Flow

- Before the start of execution - complete program must be stored in the primary memory.
- **Assume** - starting address of the stored program - 8000H (for example).
- While running the program - microprocessor must be directed to 'go' to 8000H.
- Once it has executed the instruction in 8000H, it goes to the next address 8001H (assuming single byte instructions) and then 8002H etc., until it reaches the end of the program.

Introduction – 8085 Microprocessor

- Intel 8085 is an eight bit microprocessor of INTEL Corporation, usually called as a general purpose 8-bit Micro-Processor (**8-bit MP** or **8-bit μP**).
- Several (faster) versions of 8085 microprocessor was developed - 8085AH, 8085AH-1, 8085AH-2
- Microprocessor system consist of three functional blocks - a Central Processing Unit (CPU), input/output, - memory units

A microprocessor system



Third Generation

During 1978-80

HMOS technology \Rightarrow Faster speed, Higher packing density
16 bit processors \Rightarrow 40/ 48/ 64 pins
Easier to program
Dynamically relatable programs
Processor has multiply/ divide arithmetic hardware
More powerful interrupt handling capabilities
Flexible I/O port addressing

Intel 8086 (16 bit processor)

First Generation

Between 1971 – 1973

PMOS technology, non compatible with TTL
4 bit processors \Rightarrow 16 pins
8 and 16 bit processors \Rightarrow 40 pins
Due to limitations of pins, signals are multiplexed

Fifth Generation Pentium

Fourth Generation

During 1980s

Low power version of HMOS technology (HCMOS)
32 bit processors
Physical memory space 2^{24} bytes = 16 Mb
Virtual memory space 2^{40} bytes = 1 Tb
Floating point hardware
Supports increased number of addressing modes

Intel 80386

Second Generation

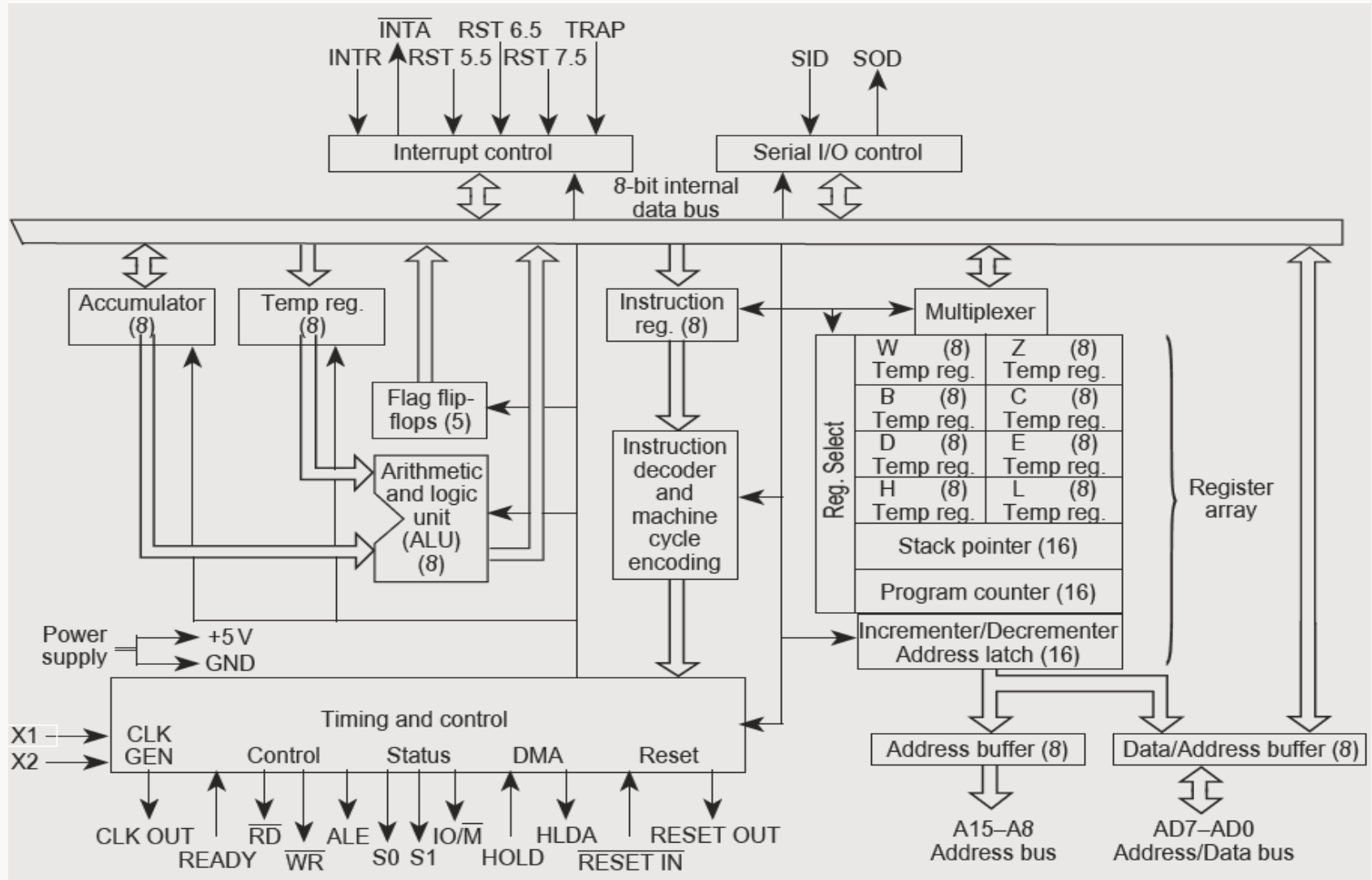
During 1973-78

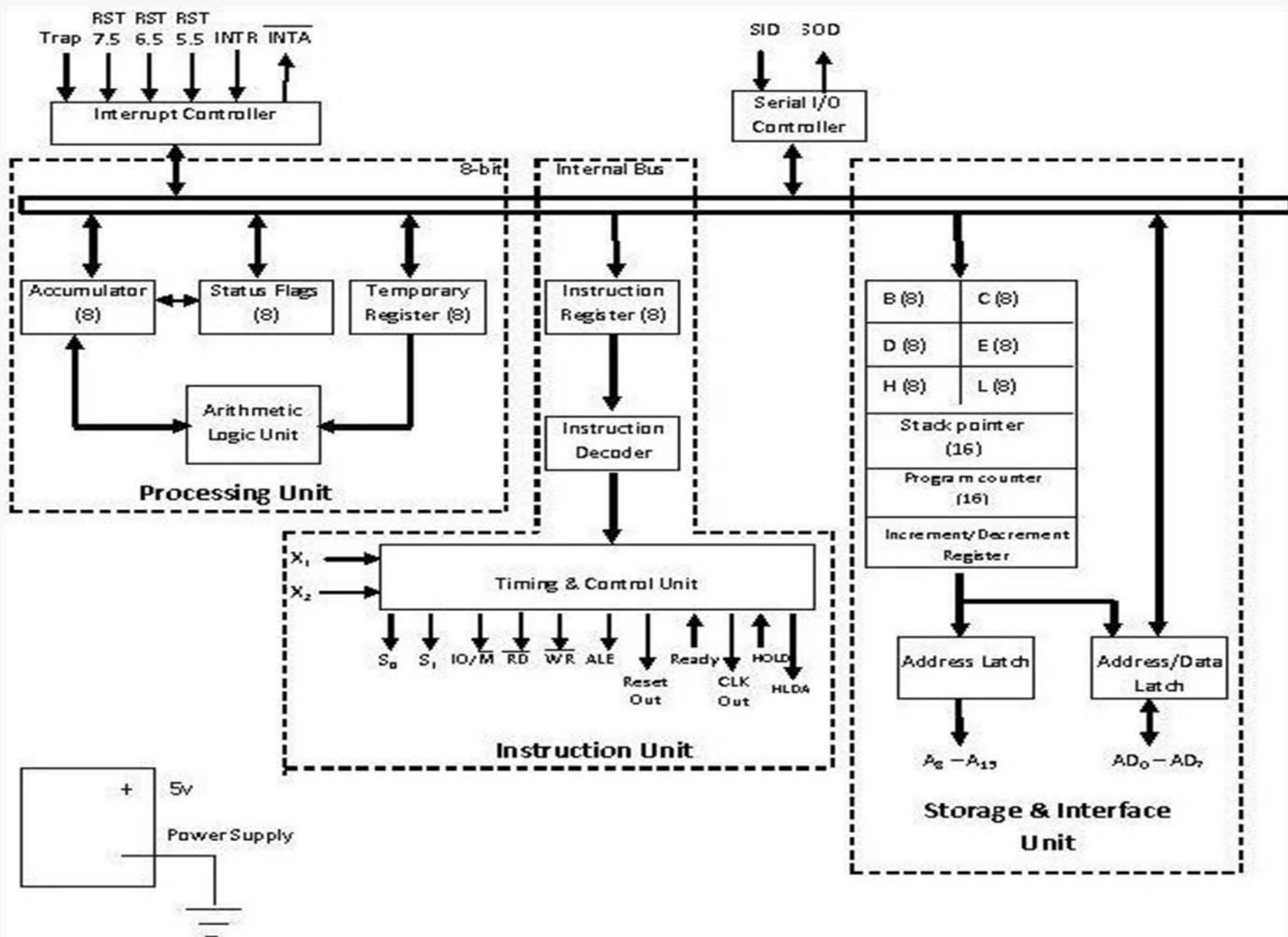
NMOS technology \Rightarrow Faster speed, Higher density, Compatible with TTL
4/ 8/ 16 bit processors \Rightarrow 40 pins
Ability to address large memory spaces and I/O ports
Greater number of levels of subroutine nesting
Better interrupt handling capabilities

Intel 8085 (8 bit processor)

Architecture of 8085

Block Diagram – 8085





Functional Description of 8085

- It is a 40 pin IC package and uses +5V for power.
- It can run at a (maximum) frequency of 3MHz.
- The modified versions of the 8085 processor have these minimum common feature and functional similarities.
- 8-bit processor - **data length is 8-bit** - has a data bus of 8-bits wide
- Addressing capability of 16-bit
- That means it can address $2^{16} = \mathbf{64K}$ Bytes of memory (1Kbyte = 1024 byte).

Functional Units of 8085 processor

- The processor consists of five functional units:-
- **Arithmetic and logic Unit**
- **General purpose registers**
- **Special Purpose Registers**
- **Instruction register and decoder and**
- **Timing and control unit.**

Arithmetic Logic Unit (ALU)

- ALU is the circuitry which performs the actual arithmetic and logical operations.
- Addition (ADD), subtraction (SUB), increment (INR), decrement (DCR) etc. are the arithmetic operations available with 8085 microprocessor.
- The possible logical operations in 8085 are AND (ANA), OR (ORA), EXLUCIVE OR (XRA), COMPARE(CMP), COMPLEMENT (CMA) etc.

Arithmetic Logic Unit (ALU) Contd..

- The data is fetched from the memory, the operation executed with the content of the accumulator - results stored in the Accumulator
- ALU of 8085 is called **accumulator oriented ALU** - one of the data for arithmetic and logical operations - must be stored in **accumulator**.
- If the operation needs **only one** data, then that data **usually be stored in accumulator**.

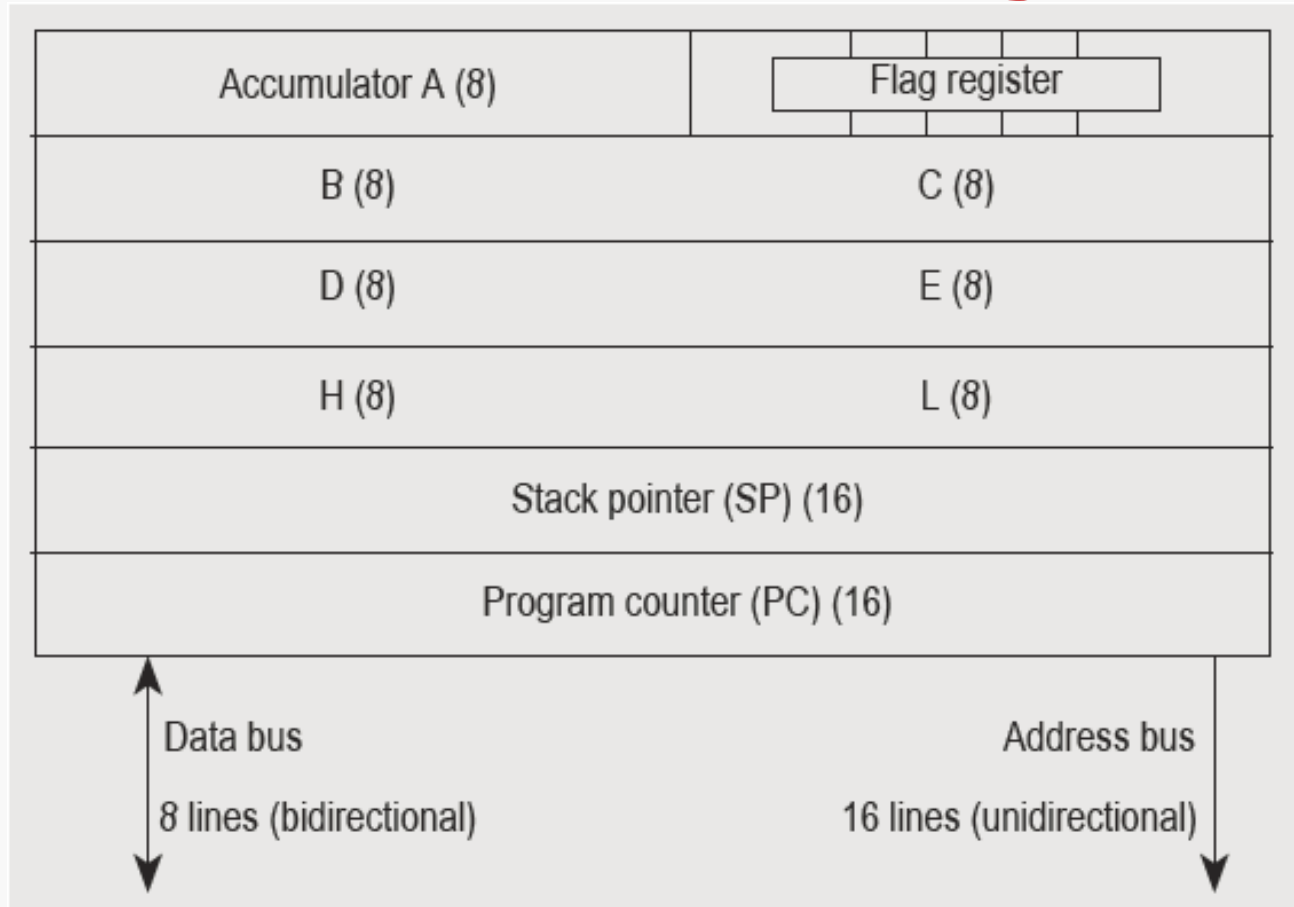
The general-purpose registers

- A register - collection of 8 flip-flops with **parallel in and parallel** out operation.
- A flip-flop can only store one bit so to handle 8 bits at a time, **8 flip-flops** and named as an 8-bit register.
- Registers - storage areas inside the microprocessor
- they differ by the purpose of storage.

The general-purpose registers Contd..

- The **general purpose registers** - store only the data that is used by the currently running program - results obtained from the currently running program
- These general purpose registers - **user accessible** by programs.
- Registers B, C, D, E, H and L - general purpose registers of 8085
- They can also be called as **Scratch Pad** registers.
- registers W and Z are internal registers. which are controlled internally and not available for user access.

General purpose & Special Purpose Registers



The general-purpose registers Contd..

- **All most in all** arithmetic and logical operations - registers - used as the second operands while the first operand is being **accumulator (A)**.
- The general-purpose registers - all 8-bit registers but they can be handled as 16-bit registers.
- They can be combined as **register pairs** – B and C, D and E, and H and L - to perform 16-bit operations

The general-purpose registers Contd..

- They are named respectively - **register pairs BC, DE, and HL**
- Among these pairs **HL** has a significance.
- Memory (**M**) related instructions of 8085 use - **HL pair as memory pointer**.
- For example, the instruction “**MOV A, M**” transfers the content of memory location pointed by the HL pair to accumulator.
- The HL pair should be pre-loaded with the memory address in which data is available.

The special -purpose registers

- There are also **special purpose registers** that are dedicated to particular/specific function.
- The Accumulator, **flag register**, **Program Counter (PC)** and **Stack Pointer (SP)** constitute the special registers in the 8085 Microprocessor.

Accumulator (A)

- The accumulator - 8-bit register - part of arithmetic/logic unit (ALU) - most important register.
- Used to store - 8-bit data - perform arithmetic and logical operations. The result of an operation is also stored in the accumulator.
- The accumulator is identified as register A in the instruction set of 8085.
- The programmer can use it at any time to store an 8-bit binary number.

Accumulator (A) Contd..

- Being an 8-bit register, it only hold one byte at a time. Any previous data stored in this register will be overwritten as soon as anything else is stored.
- Also the 8085 microprocessor communicates with input/output devices only through A. For input (**IN**) and for output **OUT** instruction is used.

The flag register (F)

- This is a special 8-bit register.
- **Each bit of flag register is quite independent of each other.**
- In all other registers, each bit is just part of a single binary byte value, but in flag register, every bit has different meaning/ value.
- The flag is an 8-bit register used to indicate the **status of the recent arithmetic or logic operation.** It may be set or reset after an arithmetic or logical operation according to the condition of the processed data.

- S = Sign Flag
- Z = Zero Flag
- AC = Auxiliary Carry Flag
- P = Parity Flag
- CY = Carry Flag

The flag register Contd..

- The five flag bits are Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags.
- The remaining three bits (D1, D3 and D5) of the flag register remain unassigned and they are marked with an X to show that they are not used and are don't cares.
- **It provides the status of μP after every instruction.**

Flag Register Bit Position

S	Z	X	AC	X	P	X	CY
D7	D6	D5	D4	D3	D2	D1	D0

The flag register Contd..

- Any flag register bit is said to be 'set' when its value is 1 and is 'cleared' when it is 0. The most commonly used flags are **Zero**, **Carry**, and **Sign**. **AC** can not be accessed externally.
- The **S** flag is **set** if the result of the last mathematical operation is **negative** while reset when the result is positive.

- **Sign Flag (S):**

- It tells the sign of result stored in Accumulator after the operation is performed.
- If result is **−ve**, sign flag is **set** (1).
- If result is **+ve**, sign flag is **reset** (0).

Zero flag (Z)

- The **zero** flag is **set** if the result of an arithmetic operation results in a **zero**. Otherwise it is reset.
- It 'sets' or changes to a binary 1 if it sees a zero result in accumulator (i.e. all 8 bits of accumulator are 0) and stays at binary 0 at all other times.

- **Zero Flag (Z):**

- It tells whether the result stored in Accumulator is zero or not after the operation is performed.
- If result is **zero**, zero flag is **set** (1).
- If result is **not zero**, zero flag is **reset** (0).

Carry flag (C)

- The carry flag is **set** when a **carry** is generated in the process of an arithmetic operation out of accumulator. Otherwise reset.
- When addition is carried out, it sometimes results in a 9th bit being carried over to the next byte.
- It also reflects the value of the '**borrow**' in subtractions.

Auxiliary carry flag (AC)

- The auxiliary carry flag is set when an auxiliary carry is generated in the process of an arithmetic operation in the accumulator.
- That is when a carry results from bit **D3** and passes on to **D4** (i.e. from the lower 4 bits/nibble to the higher 4 bits/nibble). Used in BCD arithmetic.
- This carry is also called as **half-carry**.
- D7 D6 D5 **D4** ← **D3** D2 D1 D0

Parity flag(P)

- The parity flag is **set** if the content of the accumulator after an arithmetic operation has an **even** number of 1's.
- Otherwise the parity flag is reset. It is set for operation in the even parity mode.
- **Parity is number of 1's**
 - If parity is **even**, parity flag is **set** (1).
 - If parity is **odd**, parity flag is **reset** (0).

Program counter (PC)

- PC is a 16-bit register, which always points to the **address of next instruction to be executed**.
- In other words this register is used to sequence the execution of the instructions.
- At the end of execution of any instruction, the **content** of the memory location indicated by PC is moved to **instruction register** and the PC is loaded with the new address to indicate the next address.
- As it keeps track of program by always counting the memory address from which the next byte is to be fetched, a self-explanatory name is given.

Stack Pointer (SP)

- Stack is an array of memory locations organized in the form of '**Last In First Out**' or 'First In Last Out' fashion.
- Stack is very much essential in any microprocessor based system.
- Stack is mainly used to store the return address of the main program when **subroutines** are called.

Stack Pointer (SP) Contd..

- Stack Pointer (SP) is a 16-bit register that holds the **address of the memory location of top of the stack.**
- The architecture of 8085 offers flexibility to define the stack locations by the programmer.
- The programmer can allocate a series of RAM locations to be used as a **stack** and an address counter must be employed to keep track of what address of the stack is to be used next.

Stack Pointer (SP) Contd..

- This counter, called stack pointer (SP) points to the next free location in the stack to be used.
- The users are allowed to use the stack but the microprocessor takes priority and also uses the stack.
- Care must be taken by the programmer that the data stored in the stack is retrieved properly so that the data stored in the stack by the processor is not affected.
- the range of stack memory locations must be chosen carefully such that it would not affect program space.

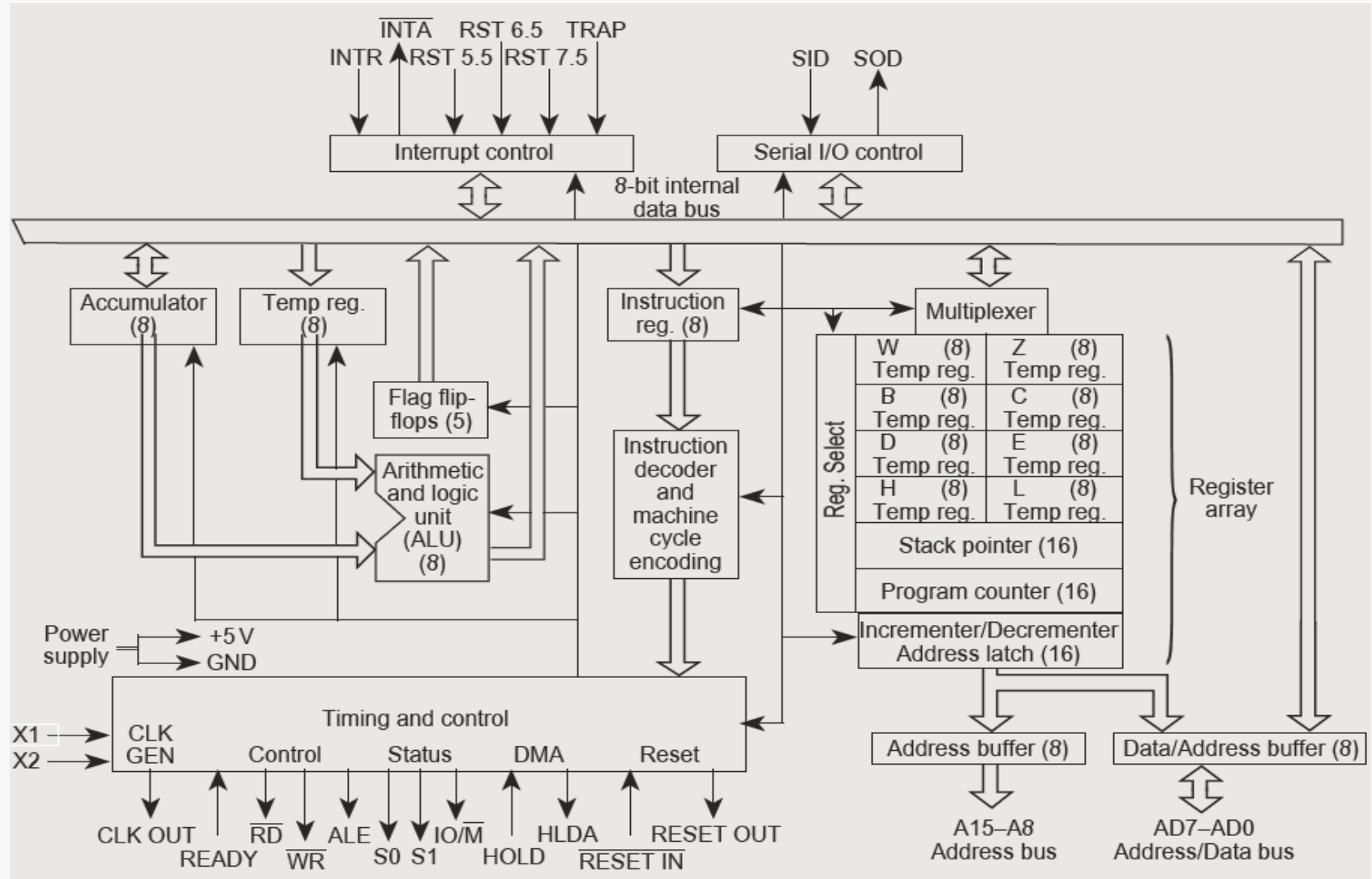
Instruction Register and Decoder

- It is an 8-bit register that usually offers a temporary storage for the instructions fetched from memory locations before the actual execution of it.
- The content of the register is decoded by the decoder circuitry, where the nature of the operation that is to be performed is decided (interpreted).

Instruction Register and Decoder ...

- The internal machine command is interpreted to decide the next course of action.
- In addition, there are two temporary registers **W** and **Z** which are controlled internally and **not available for user access**.

Block Diagram – 8085



Data Bus

- The microprocessor performs its functions using wires or lines called buses.
- For example, an 8-bit microprocessor (8085) uses 8 wires to carry the data between the microprocessor and the memory (or I/O).

Data Bus Contd...

- Microprocessor communicates with memory and other devices (input and output) using these busses.
- There are three types of buses namely the **address bus**, the **data bus** and the **control bus**.
- The data bus (**D0-D7**) is a two-way (**bi-directional**) bus carrying data around the system.
- Information going into the microprocessor and results coming out of the microprocessor are through this data bus.

Data Bus Contd...

- It is used for transfer of binary information between the microprocessor, memory and peripherals.
- The **lower group of eight address lines A0-A7** is multiplexed with the data bus in order to reduce the **pin count**.
- Therefore the multiplexed **lower group** of address lines and data lines is more generally denoted as **AD0-AD7**.

Address Bus

- The address bus carries addresses and is a **one-way** bus from the microprocessor to the memory or other devices (**uni-directional**).
- It is a group of sixteen unidirectional lines that allow flow of address from the processor to its peripheral devices.
- Each peripheral or a memory location is identified by a 16 bit binary number called an address.
- It follows that the maximum addressing capability of the 8085 processor $2^{16}=64\text{KB}$. Its basic function is to identify a peripheral or **memory location**.

Address Bus Contd..

- The address bus is a group of 16 lines generally identified as **A0 to A15**.
- The address bus has 8 higher order address lines **A8 – A15** which are unidirectional.
- The **lower order 8 lines** are multiplexed (time shared) with the 8 data lines (**AD0 – AD7**) and hence they are **bi-directional**.

Address Bus Contd..

- During the execution of the instruction, these lines carry the address bits during the **early part** (initially), and then during the **late parts** of the **execution**, they carry the 8 **data** bits.
- In order to separate the address from the data, a **latch is used externally** to save the **address** before the function (of the bits) changes. **(ALE)**

- Address and Data Buses:
- A8 - A15 (Output only) - **higher order address bus**-
 - The most significant 8 bits of the (memory) address or the 8 bits of the I/O addresses, **unidirectional**.
- AD0-AD7 (Input/output)- multiplexed address/data bus) **bidirectional**.
- 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.**

MPU Communication and Bus Timing

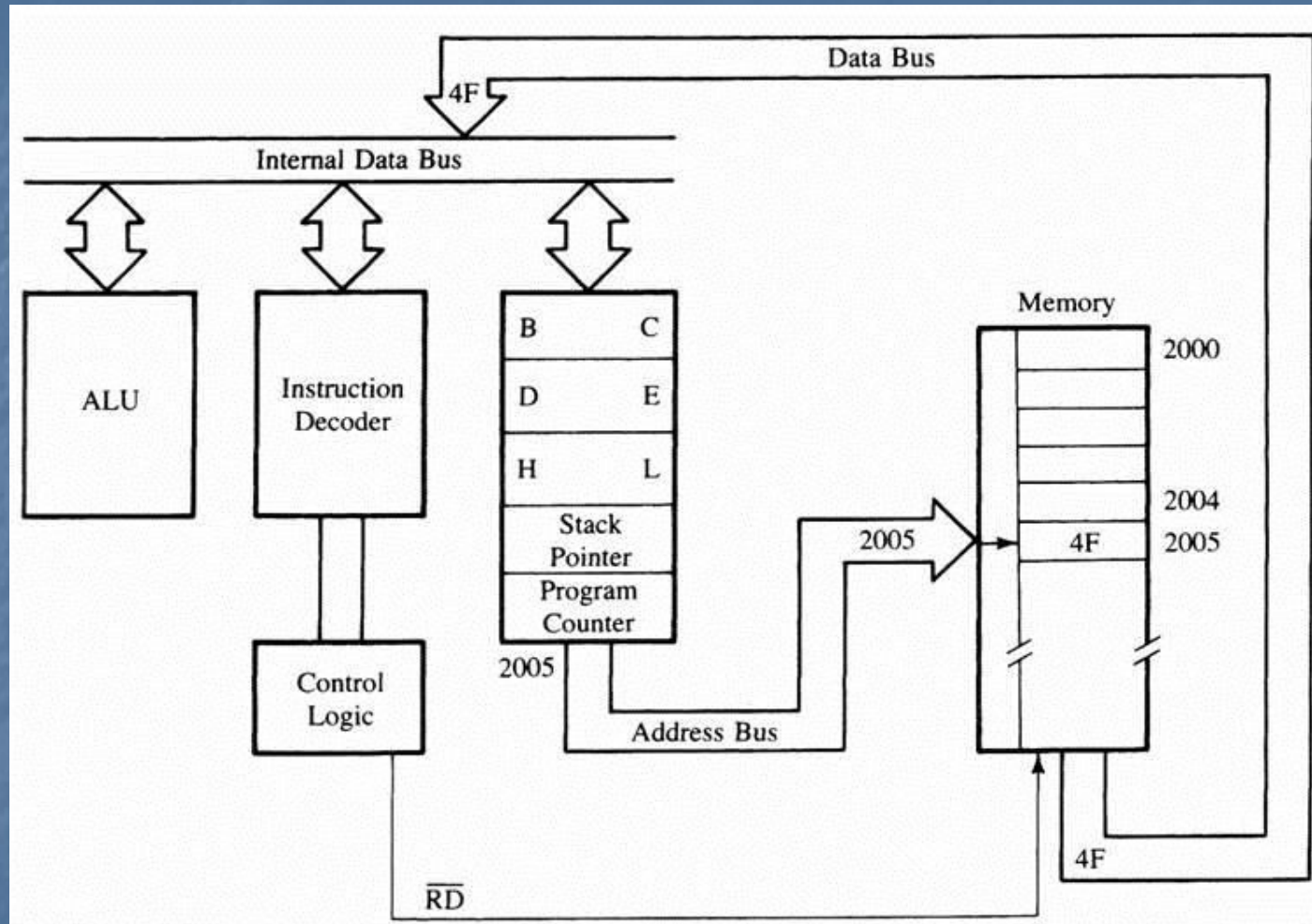


Figure 3: Moving data form memory to MPU using instruction **MOV C, A** (code machine 4FH = 0100 1111)

Control Bus

- The control bus comprises of various **single** lines that have specific functions for coordinating and controlling microprocessors operations.
- For example, a **Read** or **Write** control signal will indicate whether memory is being 'written to' (data stored in memory) or 'read from' (data taken out of memory).
- Thus, they are individual lines that provide a pulse to indicate operation of the microprocessor.

Control Bus Contd...

- In fact the microprocessor generates specific control signals for **every operation**, which in turn is used to identify the **type of device** the processor intends to communicate.
- The microprocessor cannot function correctly without these vital control signals.
- The Control Bus carries control signals which are partly bi-directional (**some are input and others are output**).

Control signals - 8085

- Typically microprocessor has 12 control lines.
- There are **four** main control and status signals.

They are:

ALE (Address Latch Enable), *output*

It is a pulse that is provided (**made high**) when an address appears on the lower 8 (AD0 – AD7) lines, after which it becomes 0. This signal can be used to enable a **latch** to save the address bits from the AD lines so that **later**, they can be used as **data buses**.

Control Signals - 8085

- **RD~** (Read - **Active low output**), *output*

It indicates that the data is read from the selected memory or I/O device and then data available/moved on the data bus.

- **WR~** (Write - **Active low output**), *output*

It indicates that the data on the data bus are to be written into a selected memory location or I/O.

Control Signals - 8085

- **IO/ \overline{M} (output)**
 - a) It is a signal that **distinguishes** between a memory operation and an IO operation.
 - b) An active low on this signal shows it is a **memory operation ($\text{IO}/\overline{M}=0$)** and a high on this line indicates an **IO operation ($\text{IO}/\overline{M}=1$)**.

Control Signals – 8085

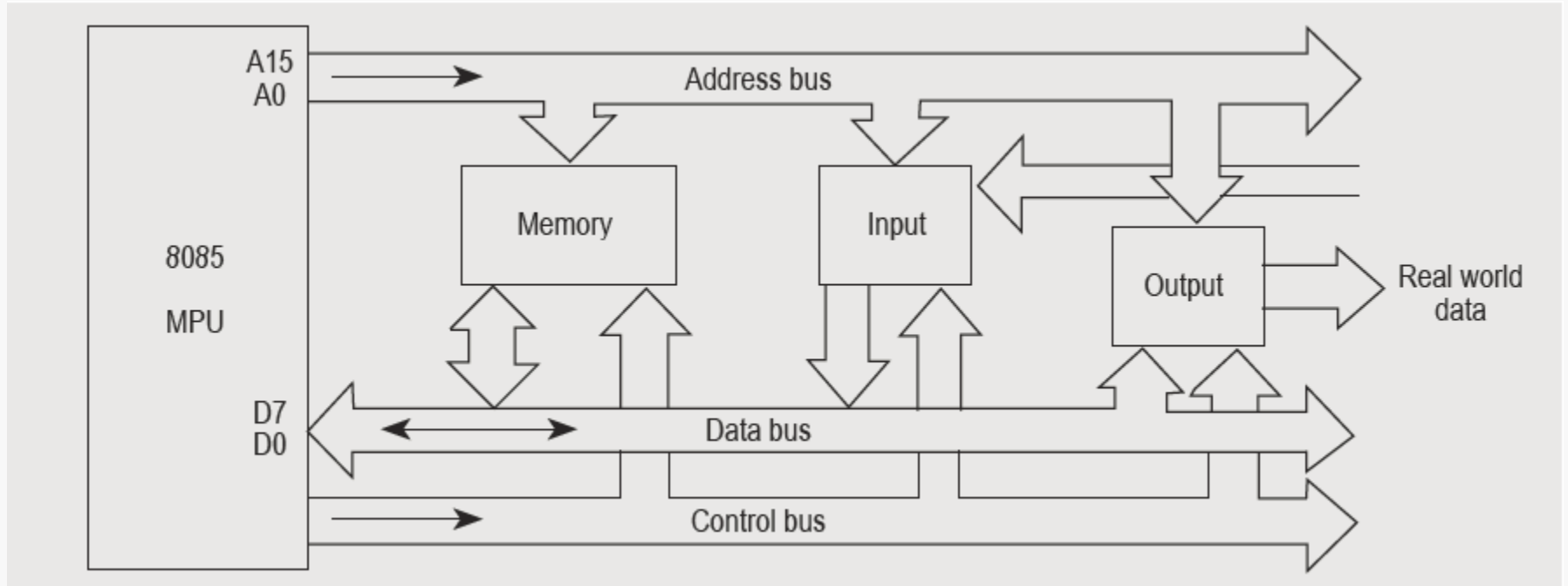
S1 and S0 (output)

- They are **status signals**, used to specify the kind of operation being performed. (Also Indicate- **Data Bus Status**)
- The status signals combined with I/O signals to govern the different operations as listed in Table
- If both S0 and S1 are **low**, the operation of the processor tends to **halt**.
- If S1 **low** and S0 is **high**, the processor **reads** the data. While the processor **writes** data on to a memory or I/O, **S1 is high** and **S0 is low**.
- If both S0 and S1 are **high**, the **instruction fetch** operation is performed.

Status signals-operations

S1	S0	States
0	0	Halt
0	1	Write
1	0	Read
1	1	Fetch

8085 bus structure



Introduction – 8085 Bus Structure

- The schematic representation of the 8085 bus structure explains how the movement of data within the computer is accomplished by a series of buses.
- Address information, data and control signals have to be carried around inside the microprocessor as well as in the external system and hence the buses will be seen both internal as well as external.

Interrupts

It means ***interrupting*** the normal execution of the microprocessor.

- When microprocessor receives interrupt signal, it **discontinues** whatever it was executing.
- It **starts** executing new program indicated by the interrupt signal.
- Interrupt signals are generated by external peripheral devices.
- After execution of the new program, microprocessor **goes back** to the previous program.

- Microprocessor **completes** execution of current instruction of the program.
- It stores the content of PC in the **stack**.
- PC is loaded with address of the new program.
- After executing the new program, the microprocessor **returns** back to the previous program.
- It goes to the previous program by reading the top value of **stack**.

Interrupts of 8085

- An important function to make the microprocessor, to respond to the **high priority** externally initiated signals is called as **Interrupt**.
- Whenever an interrupt signal is sensed by the **processor**, the processor **suspends** the current program execution and **executes** the program corresponding to the interrupt signal.
- There are five interrupt signals (**INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP**) that are available to facilitate the processor to realize and acknowledge the interrupt call of peripherals.

Interrupt contd...

- **INTR: (Input)**

It is a general purpose interrupt request signal. It is an active high signal

- **INTA~: (output)**

It is used to acknowledge an interrupt. It is an active low signal.

Interrupt Signals

- **INTR (Input)**

- a) INTERRUPT REQUEST is used as a **general purpose** interrupt.
- b) If it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA^{\sim} will be issued.
- c) During this cycle a RESTART (RST) or CALL instruction can be inserted to jump to the interrupt service routine (**ISR**).

They are used to make the processor execute a subroutine at a predefined address called ISR

Interrupt contd...

- **RST (Restart Interrupts): (Input)**
 - These are interrupts that transfer the program control to specific memory locations. These are all maskable.
 - They have higher priority than INTR interrupts. The **priority order is RST 7.5, RST 6.5 and RST 5.5**
- **TRAP: (Input)**
 - It is a **non-maskable** interrupt and has the highest priority.
It is unaffected by any mask or Interrupt Enable.
It is the **highest priority** interrupt.

continued...

- In addition there are three other signals- **RESET**, **HOLD** and **READY**.
- RESET-IN~:
 - When the signal on this pin goes low the **program counter** set to zero and the processor is reset. It is active low signal.

Reset Signals

- **RESET IN[~] (Input)**
 - a) Reset sets the **Program Counter** to zero (0000H) and resets the Interrupt Enable and HLDA flip flops.
 - b) It is an essential signal for any microprocessor system because it determines the **address** at which program execution **begins**
 - c) In most microprocessor-based systems, RESET IN signal is applied as soon as the power is turned on.

continued...

- **RESET-OUT**
 - This signal can be used to reset other device. It is an active high signal.
- **HOLD (INPUT)**
 - This signal indicate that a peripheral such as a Direct Memory Access (DMA controller) is requesting for the use of address and data bus.

Externally initiated signals

- **HOLD (Input):**

- a) It is an active high signal used in the direct transfer of data between a peripheral device and memory locations.
- b) This type of data transfer is called as direct memory access (DMA) data transfer.
- c) During this transfer, the microprocessor loses control over the address and data buses.
- d) Logic 1 on the Hold pin indicates that another controller, generally the DMA controller, is requesting the use of the address and data buses.
- e) The CPU, upon receiving the hold request, will relinquish the use of the buses after completing the current instruction.

contd...

- During HOLD state, the peripheral (I/O) devices get control over the data and address buses for data transfer with memory.
- This operation is called as direct memory access (DMA) i.e. direct data transfer between memory and I/O devices.
- DMA is useful when high speed peripherals want data transfer with memory and during which the processor need not make any intervention.

Externally initiated signals

- **HLDA (Output)**
 - a) HOLD ACKNOWLEDGE, an active high signal, indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle.
 - b) It is an **acknowledgement** signal in response to the HOLD request.
 - c) The Cycle stealing mode and burst mode of DMA will be discussed later with DMA controller.

Interrupt contd...

- **READY (input)**

It is a signal that is used for synchronizing the slow speed peripheral devices.

The READY signal must be synchronized with the processor clock.

Externally Initiated Signals

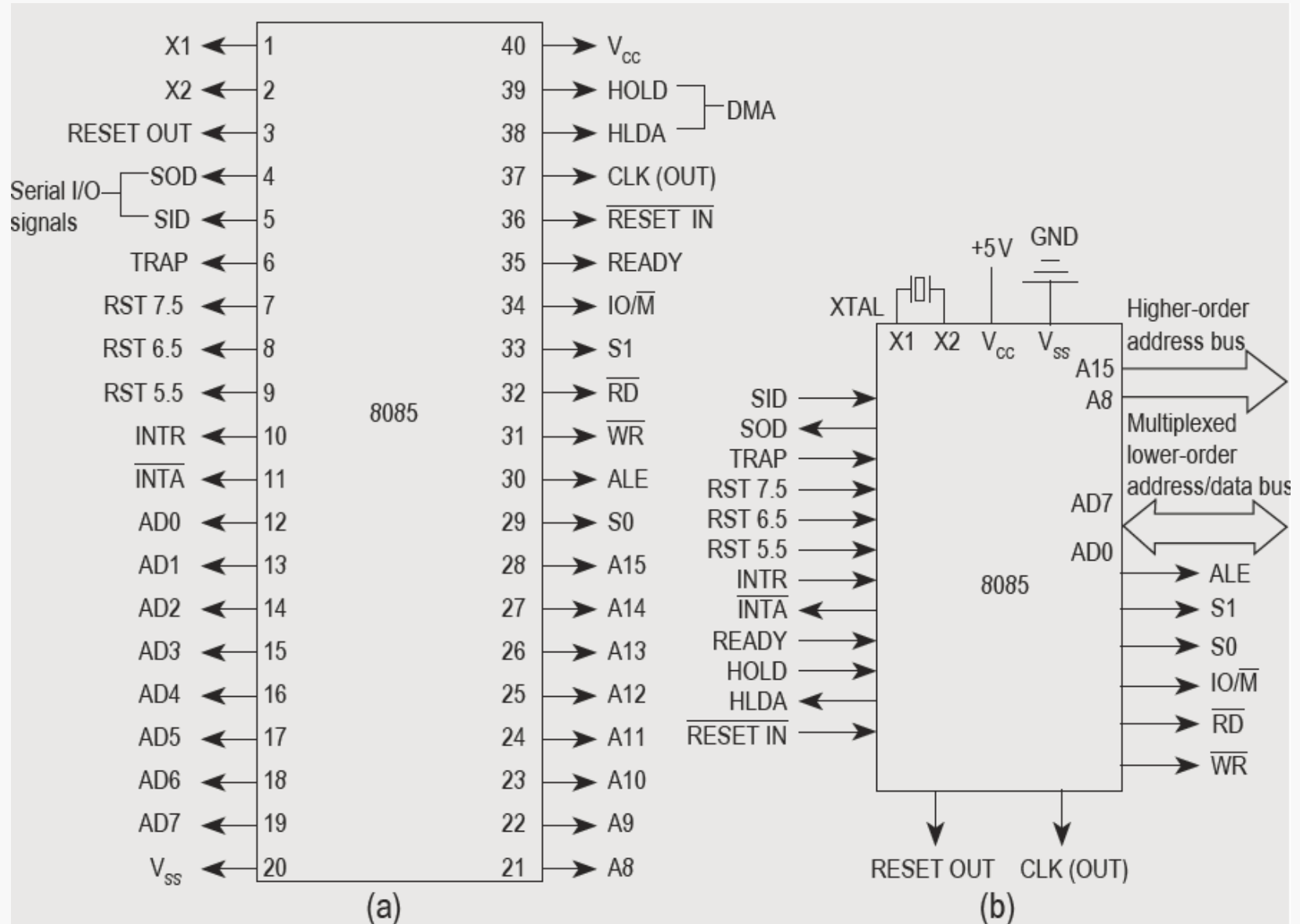
Ready (input)

- It is used to interface slow peripheral devices with the fast microprocessor.
- If Ready input is high, it indicates that the memory or peripheral is **ready** to send or receive data.
- If Ready is low, the CPU will wait for Ready signal to go high before completing the read or write cycle.

Pin Layout and Description of Signals Contd..

- The size, shape and number of pins on the microprocessor depend on the amount of data that it is designed to handle.
- It is available with 40 pins DIP (Dual in line package).

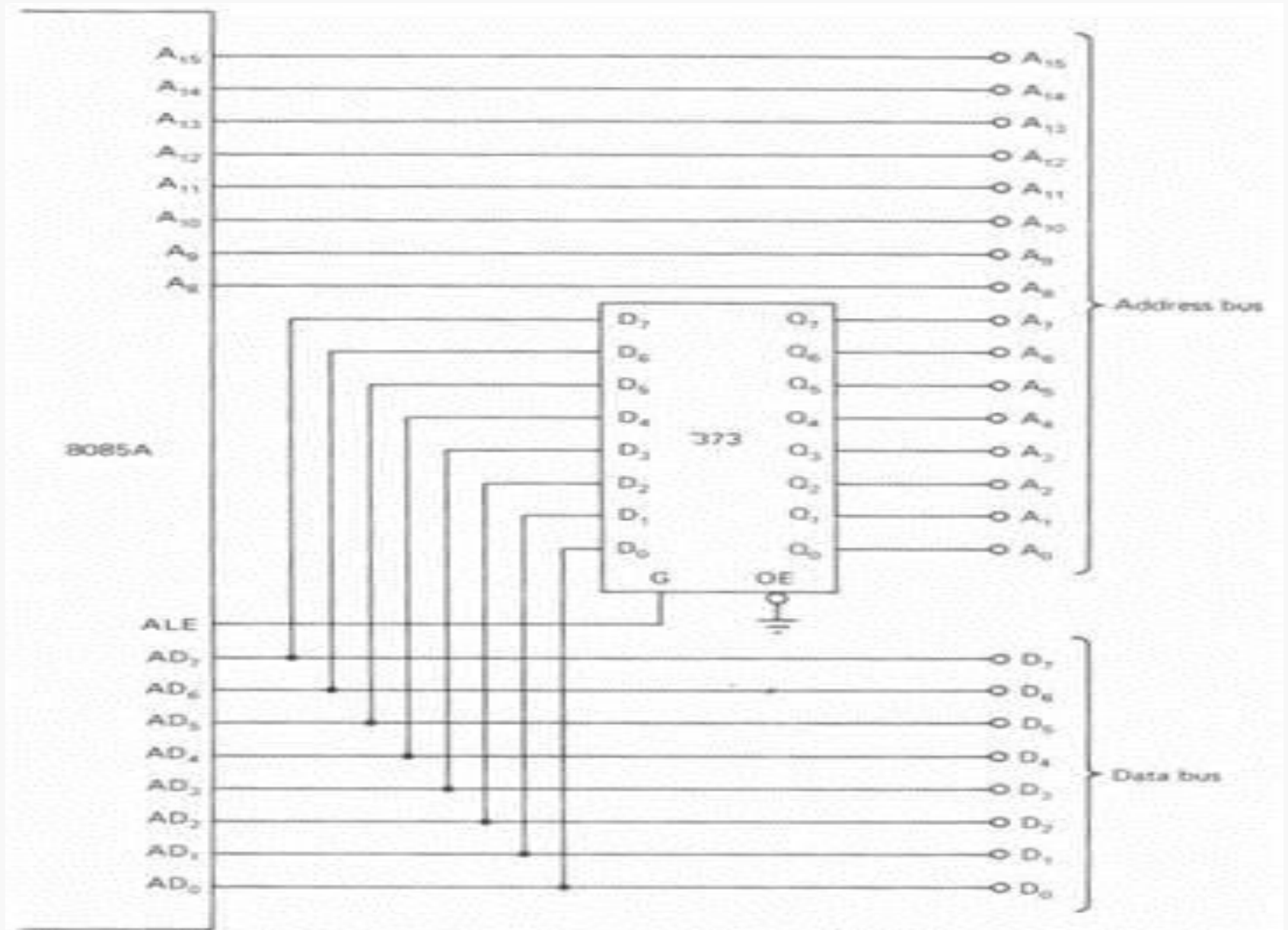
8085 Pin Diagram



Pin Functionality

- Address and Data Buses:
- A8 - A15 (Output - higher order address bus)-
 - The most significant 8 bits of the memory address or the 8 bits of the I/O addresses.
- AD0-AD7 (Input/output- multiplexed address/data bus)
 - **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.
 - Lower order 8 bits of the address lines are multiplexed with data lines, and requires de-multiplexing, which needs some additional time (though it is very small). Therefore the lower 8 bits are sent first.

ALE used to demultiplex address/data bus



Power supply and System Clock

- **X1, X2 (Input)**

- a) A microprocessor needs a square wave (clock) signal to ensure that all internal operations are synchronized.
- b) A **quartz Crystal** or R-C or L-C network is connected to these two pins.
- c) The crystal frequency is internally divided by two to give the operating frequency

Power supply and System Clock contd..

So, to run the microprocessor at 3 MHz, a clock running at 6 MHz should be connected to the X1 and X2 pins.

- Clock circuitry includes a **divide-by-two** circuit, so that double-frequency crystal can be used.

Power supply and System Clock contd..

- The **crystal** is preferred as a clock source because of its high stability, large Q (quality factor), and **absence** of frequency drifting with aging.
- Without a clock signal, the microprocessor cannot execute any program.
- **CLK (Output)**
- This output clock pin is used to provide the clock signal to the rest of the system, typical frequency is 3 MHz.
- Power supplies
- **VCC**—+5 V supply; **VSS**—ground reference

Serial I/O Signals

- **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.

8085 based system

- To make a complete system, the 8085 microprocessor needs to be interfaced with the memories and input and output devices.
- The memories required are of two types – RAM and ROM.
- ROM is necessary to store some amount of fixed programs.
- These programs are executed when the system is powered on and are essential for the system.

8085 based system Contd..

- RAM is required in any system to store temporary programs and data.
- A particular memory location is selected from the memory devices by properly issuing the address for that memory location and issuing the control signals discussed in the previous section.

8085 based system contd..

- The microprocessor is the master in any microcomputer system and the microprocessor issues the required control signals to the peripherals.
- Each location in a memory is given a number, called an address. The maximum number of locations that can be addressed will depend on the number of bits in the address.
- In general, 2^n is the number of memory locations addressed where n is the number of bits in the address.

8086

- Paging (physical division of program and memory)
 - Pages and page frames
- Segmentation (Code Segment (IP), Data Segment (BX/SI/DI), Stack Segment (SP), Extra segment)
- Fragmentation (wastage of memory)
- Swapping
- Thrashing