8085 MICROPROCESSOR ARCHITECTURE

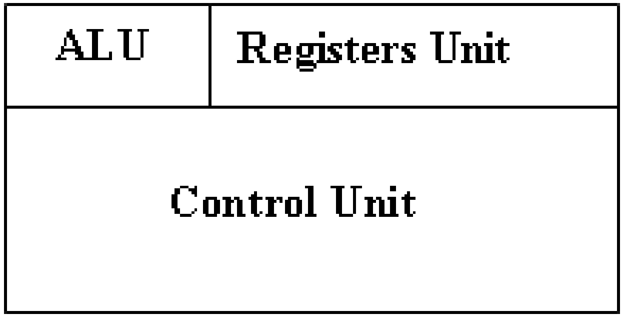
8085 Architecture , Address, Data and Control Bus, Pin Functions, Demultiplexing of bus, Generation of Control Signals, Instruction Cycle , Machine Cycle .

Introduction

A single chip CPU is called as Microprocessor

The CPU is made of two units namely the Arithmetic and Logic Unit and the Control Unit

It performs functions like executing the instructions given by the user program, controlling the I/O operations and the functions of peripheral devices

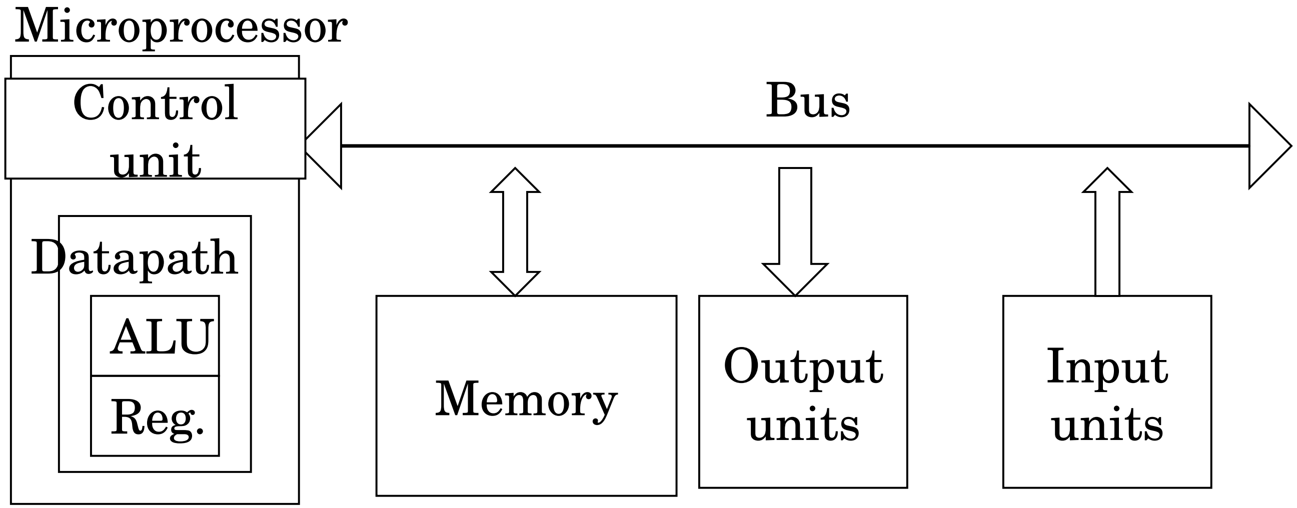


Block diagram of Microprocessor

What are microprocessor-based systems?

❑ Microprocessor-based systems are electrical systems consisting
of microprocessors, memories, I/O units, and other peripherals.

- Microprocessors are the brains of the systems
- Microprocessors access memories and other units through buses
- The operations of microprocessors are controlled by instructions stored in memories



MICROPROCESSOR ARCHITECTURE(CONT..)

- **Address Bus:**
- Consist of 16,20,24 32 parallel signal lines
- CPU sends the address of the memory location that is to be written or read from on address line
- CPU with 16 address lines can address 2^{16} that is 65,536 memory location
- **Data Bus :**
- Consist of 8,16 or 32 parallel signal lines
- Bidirectional
- CPU can read data in from memory or from a port or it can send data out to memory or to a port on this lines
- Many device has their output connected to data bus but only one device at a time will have its o/p enabled

MICROPROCESSOR ARCHITECTURE(CONT..)

- **Control Bus:**
- Consist of 4 to 10 parallel signal lines
- CPU sends out the signal on control bus to enable the output of addresed memory device
- Control signals are:

1. Memory Read
2. Memory Write
3. I/O Read
4. I/O Write

ARCHITECTURE OF GENERIC MICROPROCESSOR

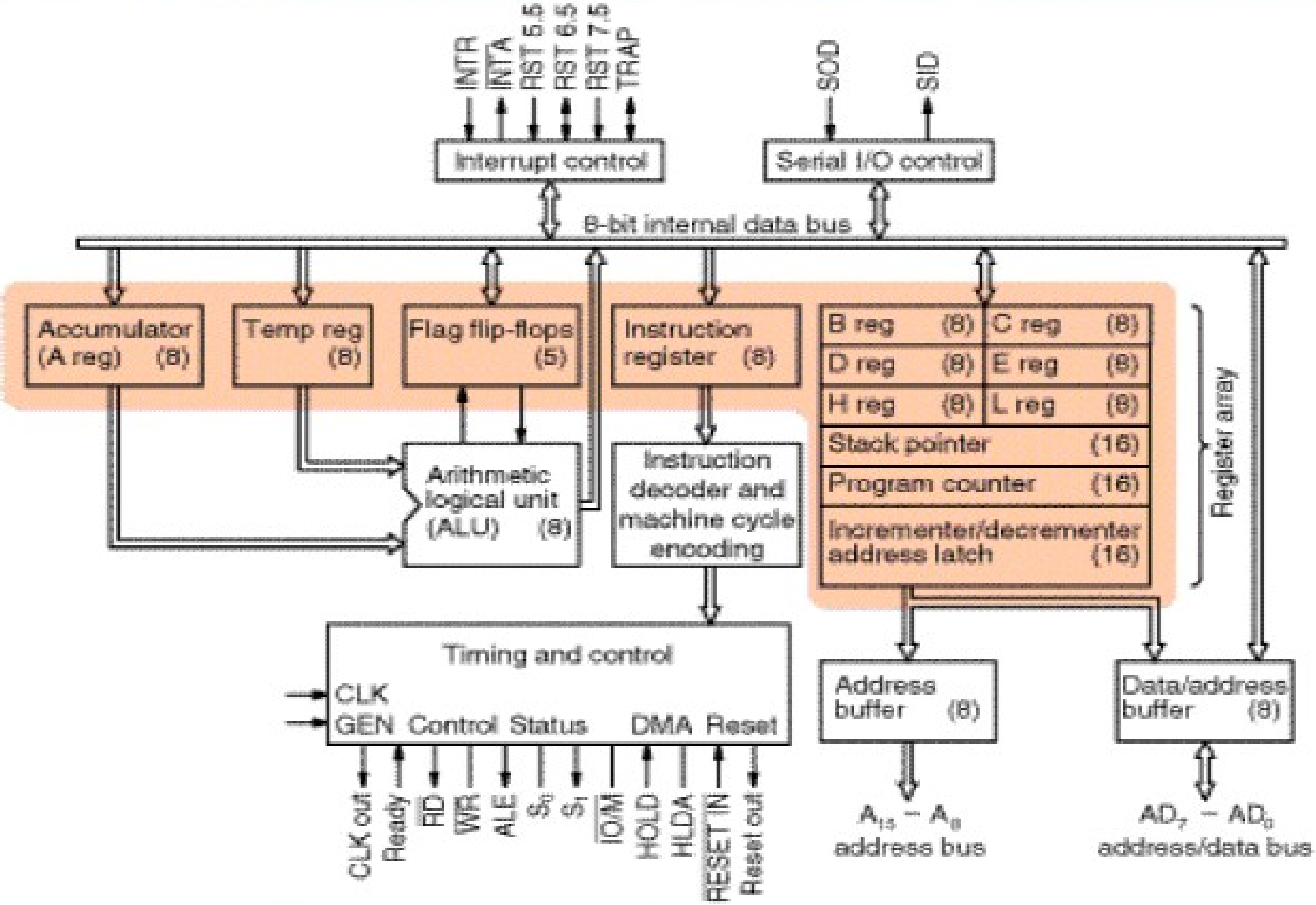
- It is heart of the system
- Performs primarily following function

1. Fetch-decode & execute program instruction in order.
2. Perform data transfer (From & to)
3. Interrupt service to external devices
4. Overall timing & control in the system

THE THREE CYCLE INSTRUCTION EXECUTION MODEL

- To execute a program, it must loaded into memory .
- The microprocessor “reads” each instruction from memory, stores in into internal register.(it is known as fetching an insruction)
- After fetching,mp interpretes it to find what insruction it is.(this is known as decoding)
- Finally mp perform the work as indicated by instrucion)
- To use the right names for the cycles:
  - The microprocessor **fetches** each instruction,
  - **decodes** it,
  - Then **executes** it.

8085 ARCHITECTURE



OPERATION TYPES IN A MICROPROCESSOR

- All of the operations of the microprocessor can be classified into one of **three types**:
  - Microprocessor Initiated Operations
  - Internal Operations
  - Peripheral Initiated Operations

MICROPROCESSOR INITIATED OPERATIONS

- Step 1: Identify the peripheral or the memory location(With its address)
- Step 2 :Transfer binary information (data and Instruction)
- Step 3 :provide timing or synchronization signal

THE 8085 ARCHITECTURE

- The 8085 uses three separate busses to perform its operations
  - The **address bus**.
  - The **data bus.**
  - The **control bus**.

THE ADDRESS BUS

- 16 bits wide (A_0 A_1…A_{15})
  - Therefore, the 8085 can access locations with numbers from 0 to 65,536. Or, the 8085 can access a total of 64K addresses.

- "Unidirectional".
  - Information flows out of the microprocessor and into the memory or peripherals.

- When the 8085 wants to access a peripheral or a memory location, it places the 16-bit address on the address bus and then sends the appropriate control signals.

THE DATA BUS

- 8 bits wide (D_0 D_1…D_7)
- “Bi-directional”.
  - Information flows both ways between the microprocessor and memory or I/O.

- The 8085 uses the data bus to transfer the binary information.

- Since the data bus has 8-bits only, then the 8085 can manipulate data 8 bits at-a-time only.

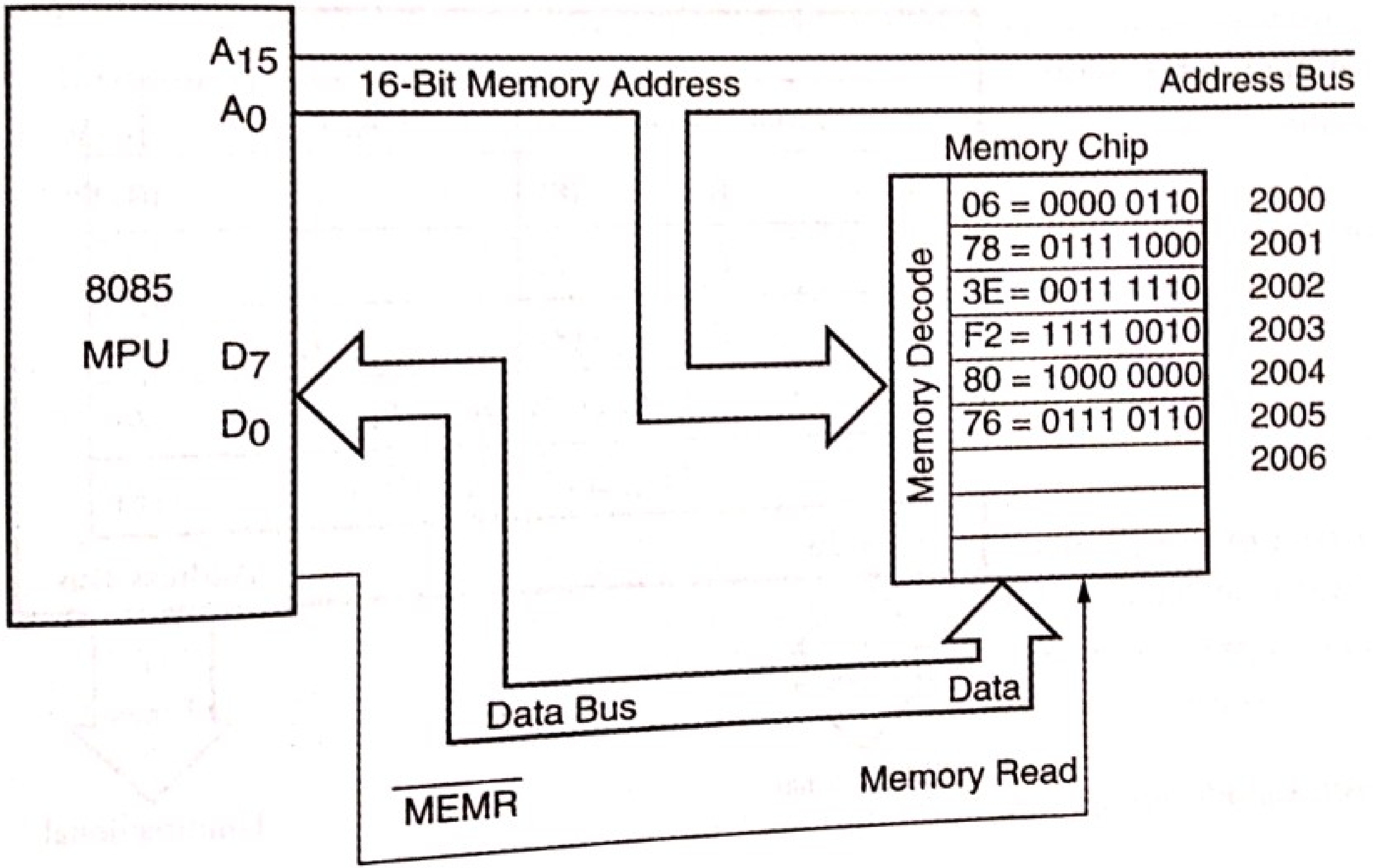
THE CONTROL BUS

  - There is no real control bus. Instead, the control bus is made up of a number of single bit control signals.
- These are usually one of **4 operations**:
  - Memory Read
  - Memory Write
  - I/O Read (Get data from an input device)
  - I/O write (Send data to an output device)

THE CONTROL BUS (CONT..)

- These are operations that the microprocessor itself starts.
- These are usually one of **4 operations**:
  - ***Memory Read***:Reads the data(or instrcuction) from memory
  - ***Memory Write***: Writes the data (or instruction)into memory
  - ***I/O Read:*** (Get data from an input device):Accepts the data from Input Device
  - ***I/O write :***(Send data to an output device):Sends data to output device

THE READ OPERATION



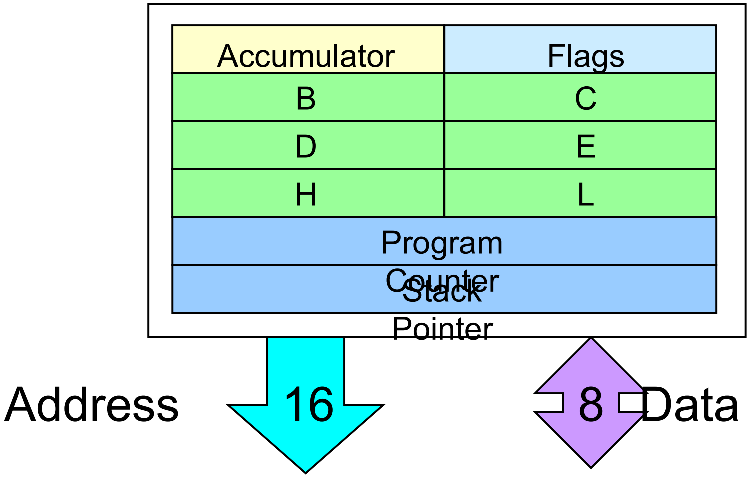
06 = 0000 0110	2000
78 = 0111 1000	2001
3E = 0011 1110	2002
F2 = 1111 0010	2003
80 = 1000 0000	2004
76 = 0111 0110	2005
	2006

THE READ OPERATION

- To read the contents of a memory location, the following steps take place:
  - The microprocessor places the 16-bit address of the memory location on the address bus.
  - The microprocessor activates a control signal called “memory read” which enables the memory chip.
  - The memory decodes the address and identifies the right location.
  - The memory places the contents on the data bus.
  - The microprocessor reads the value of the data bus after a certain amount of time.

INTERNAL DATA OPERATIONS

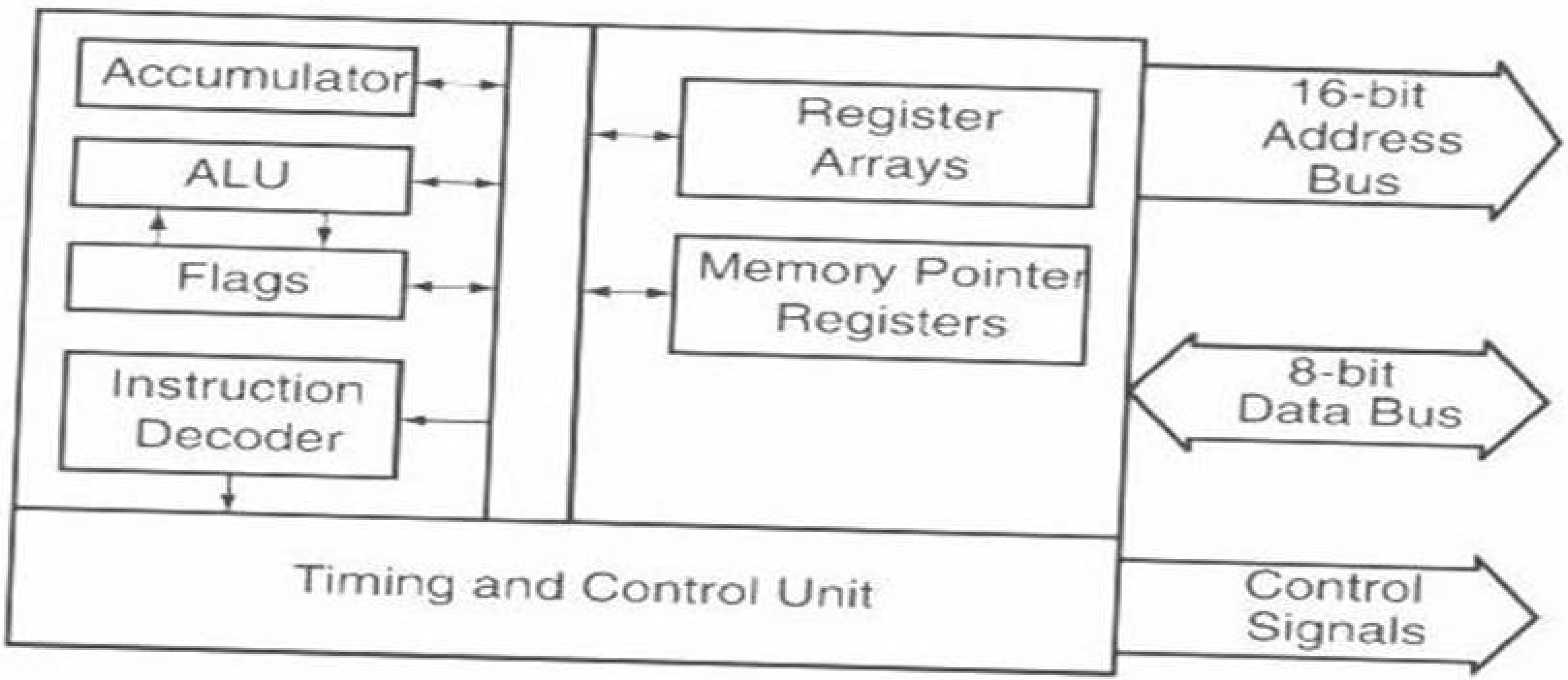
- The 8085 can perform a number of internal operations. Such as: storing data, Arithmetic & Logic operations, Testing for condition, etc.
  - To perform these operations, the microprocessor needs an internal architecture similar to the following:



THE 8085 HARDWARE/PROGRAMMING MODEL

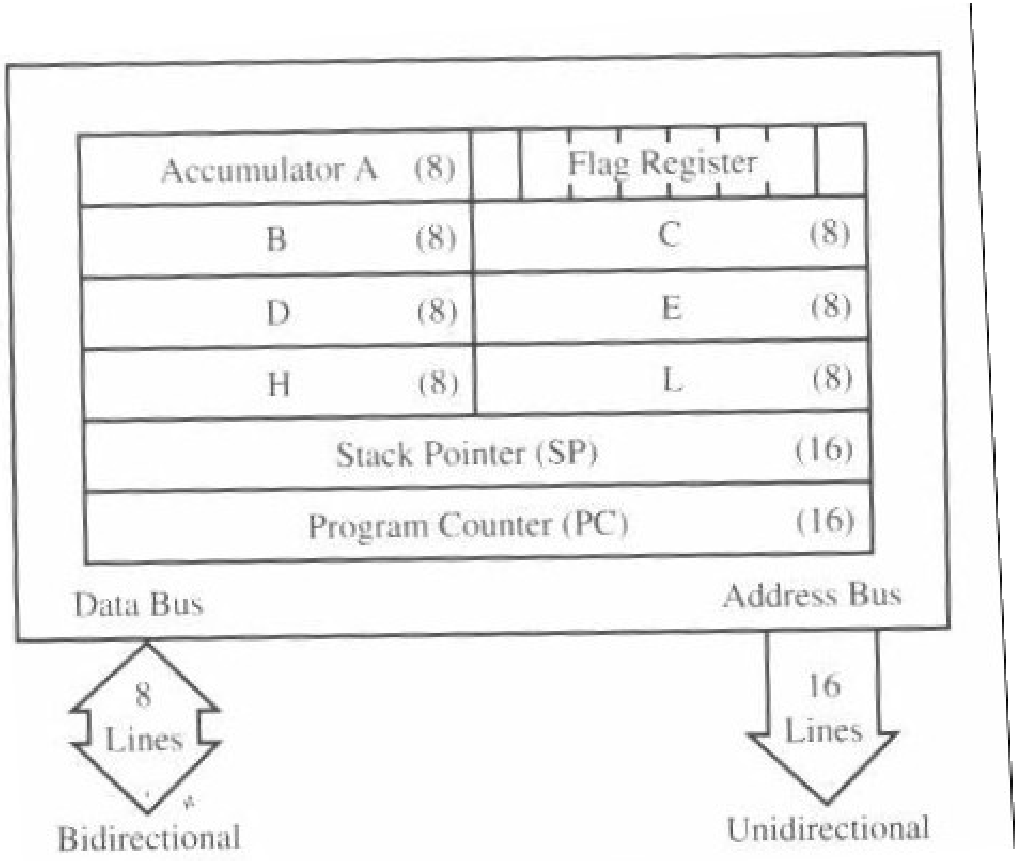
- A model is a conceptual representation of a real object.
- The microprocessor can be represented in terms of its hardware (physical electronic components) and a programming model (information needed to write programs).

8085 HARDWARE MODEL



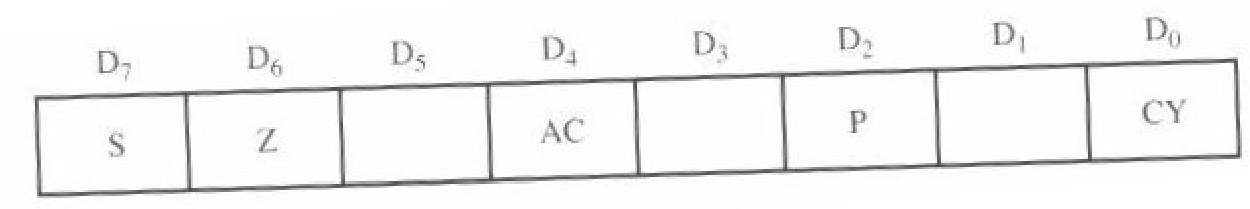
(a)

8085 PROGRAMMING MODEL



Accumulator A (8)	Flag Register
B (8)	C (8)
D (8)	E (8)
H (8)	L (8)
Stack Pointer (SP) (16)	
Program Counter (PC) (16)	

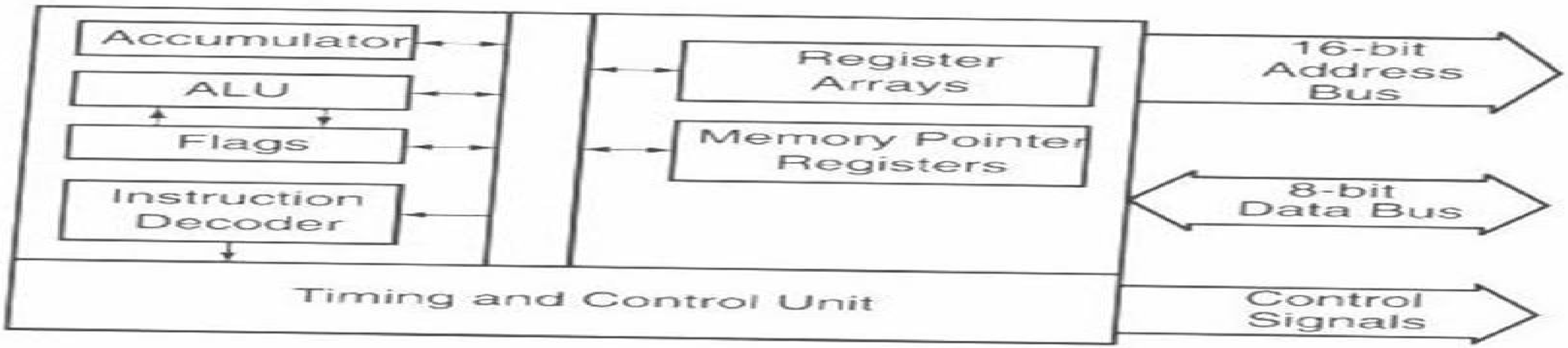
8085 FLAG REGISTER



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

8085 HARDWARE MODEL

- Two major segments:
  - One segment includes the arithmetic/logic unit (ALU) and an 8-bit register called an accumulator, instruction decoder, and flags.
  - The second segment shows 8-bit and 16-bit registers.
  - Both segments are connected with various internal connections called an internal bus.
  - The arithmetic and logical operations are performed in the ALU. Results are stored in the accumulator, and flip-flops, called flags, are set or reset to reflect the results



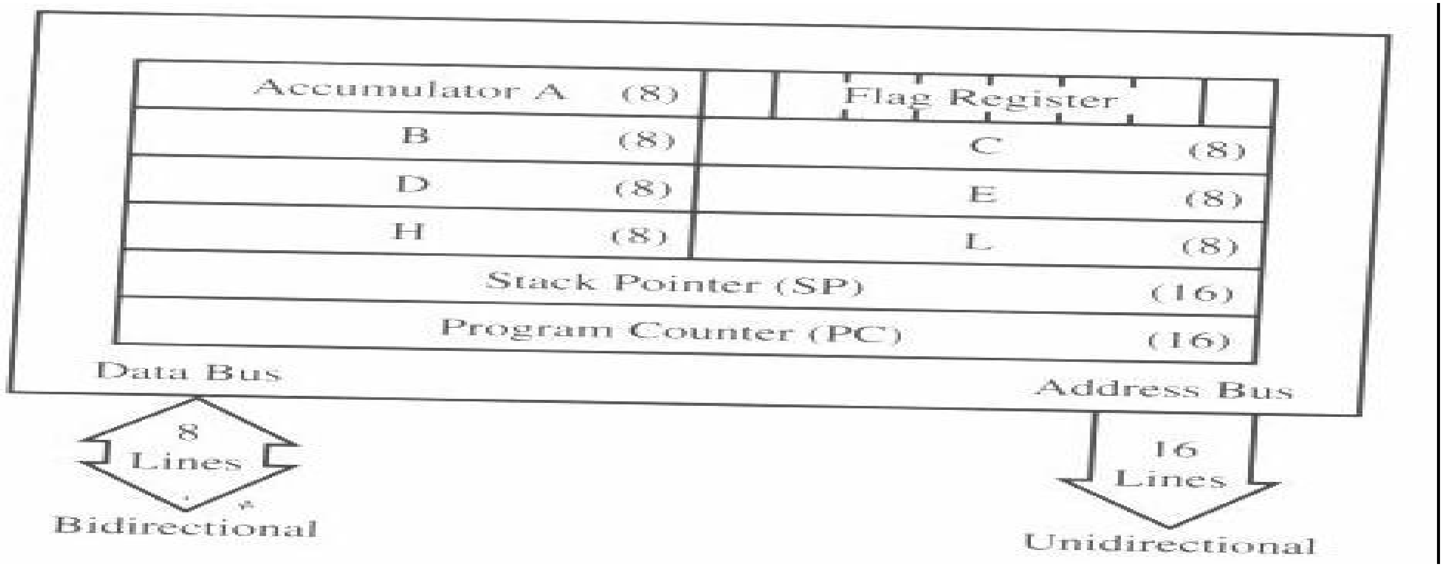
(a)

8085 HARDWARE MODEL

- There are three buses:
  - a 16-bit unidirectional address bus to send out memory addresses;
  - an 8-bit bidirectional data bus, and a control bus to transfer data, and.
  - the control bus for timing signals.

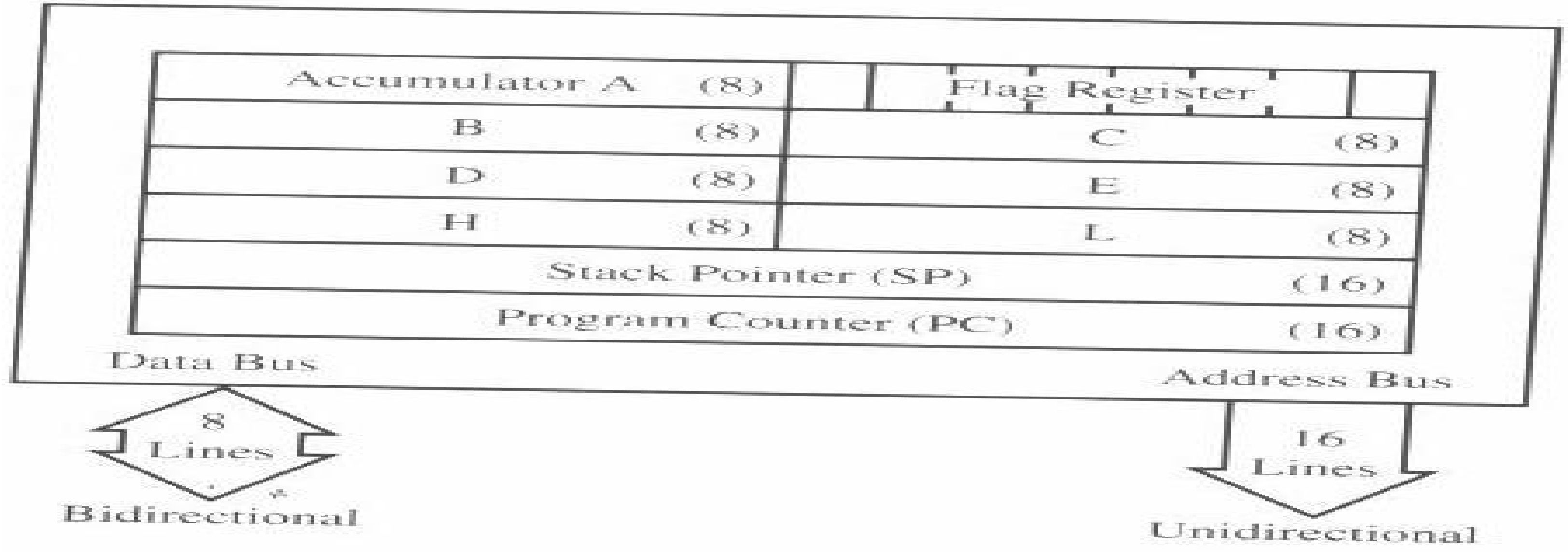
8085 PROGRAMMING MODEL

- The model includes six 8-bit registers (B, C, D, E, H & L), one accumulator, and one flag register.
- It also has two 16-bit registers:
  - the stack pointer (SP);
  - the program counter (PC).



GENERAL-PURPOSE REGISTERS

- The 8085 has six general-purpose registers to store 8-hit data;
  - B, C, D, E, H, and L.
- 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.



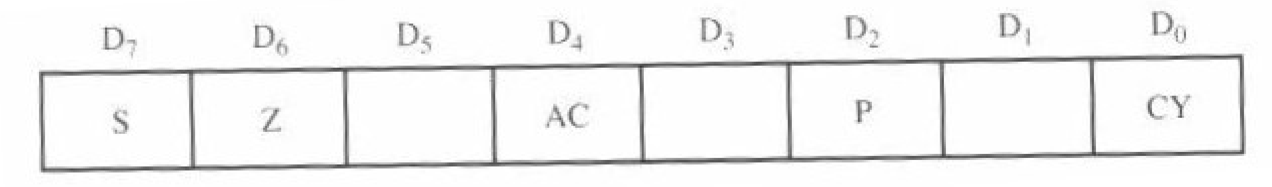
Accumulator A (8)	Flag Register
B (8)	C (8)
D (8)	E (8)
H (8)	L (8)
Stack Pointer (SP) (16)	
Program Counter (PC) (16)	

ACCUMULATOR

- The accumulator is an 8-bit register that is part of the 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;



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		CY

FLAGS

- The most commonly used flags are Zero, Carry, and Sign. The microprocessor uses these flags to test data conditions.
- These flags have critical importance in the decision-making process of the microprocessor.
  - E.g., the instruction JC (Jump On Carry) is implemented to change the sequence of a program when the CY flag is set.

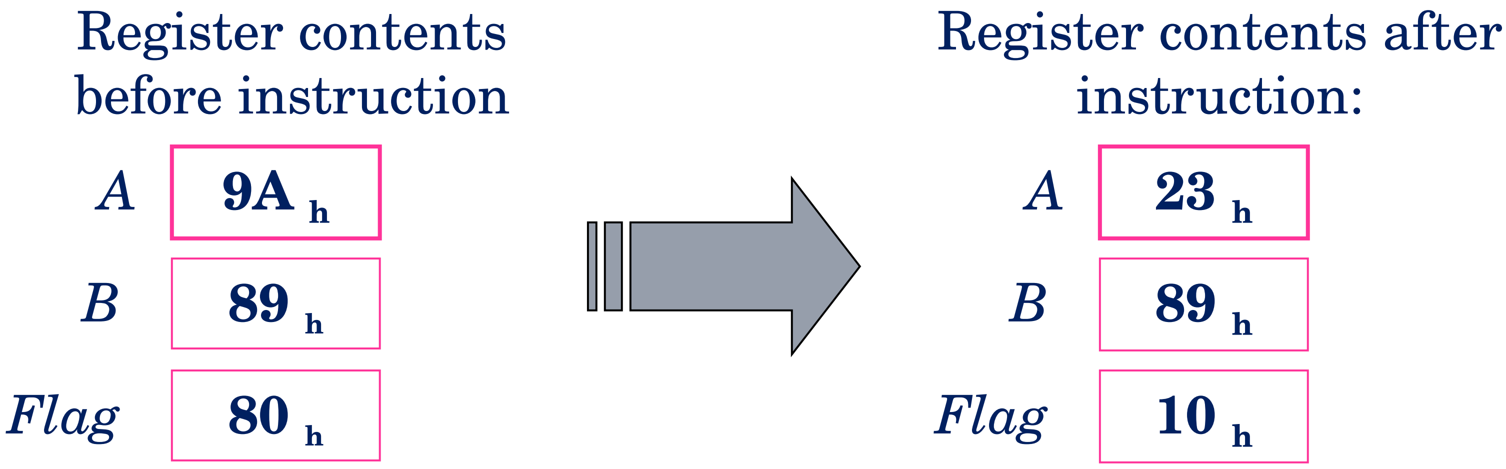
FLAGS

- The following flags are set or reset after the execution of an arithmetic or logic operation; data copy instructions do not affect any flags. See the instruction set (Appendix F) to find how flags are affected by an instruction.
  - Z-Zero: The Zero flag is set to 1 when the result is zero; otherwise it is reset.
  - CY - Carry: If an arithmetic operation results in a carry, the CY flag is set; otherwise it is reset.

FLAGS

- S - Sign: The Sign flag is set if bit D7 of the result = 1; otherwise it is reset.
- P - Parity: If the result has an even number of 1s, the flag is set; for an odd number of 1s, the flag is reset.
- AC - Auxiliary Carry: In an arithmetic operation, when a carry is generated by digit D3 and passed to digit D4, the AC flag is set. This flag is used internally for BCD (binary-coded decimal) operations; there is no Jump instruction associated with this flag.

Instruction: **ADD B**



Register contents before instruction

A	**9A** $_h$
B	**89** $_h$
Flag	**80** $_h$

Register contents after instruction:

A	**23** $_h$
B	**89** $_h$
Flag	**10** $_h$

```
 1 0 0 1   1 0 1 0
 1 0 0 0   1 0 0 1
 0 0 1 0   0 0 1 1
```

Note: All flags are modified to reflect the result of the addition.

Flag: S=0, Z=0, AC=1 , P=0 and C=1,

PROGRAM COUNTER (PC) AND STACK POINTER (SP)

- These are two 16-bit registers used to hold memory addresses.
- PC:
  - The function of the PC 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.
  - It stores the address of next instruction to be executed.
  - It sequences the execution of instruction

PROGRAM COUNTER (PC) AND STACK POINTER (SP)

- SP:
  - It points to a memory location in R/W memory, called the stack.
  - It stores the address of last location in the stack.
  - So it can be read first
  - Also known as top of the stack.

STACK

- The stack is one of the most important things you must know when programming.
  - Think of the stack as a deck of cards. When you put a card on the deck, it will be the top card. Then you put another card, then another.
  - When you remove the cards, you remove them backwards, the last card first and so on.
  - The stack works the same way, you put (push) words (addresses or register pairs) on the stack and then remove (pop) them backwards.
  - That's called LIFO, Last In First Out.

STACK

- There are instructions that allow you to modify SP contents but you should NOT change the contents of that register if you don't know what you're doing!
  - PUSH
  - POP

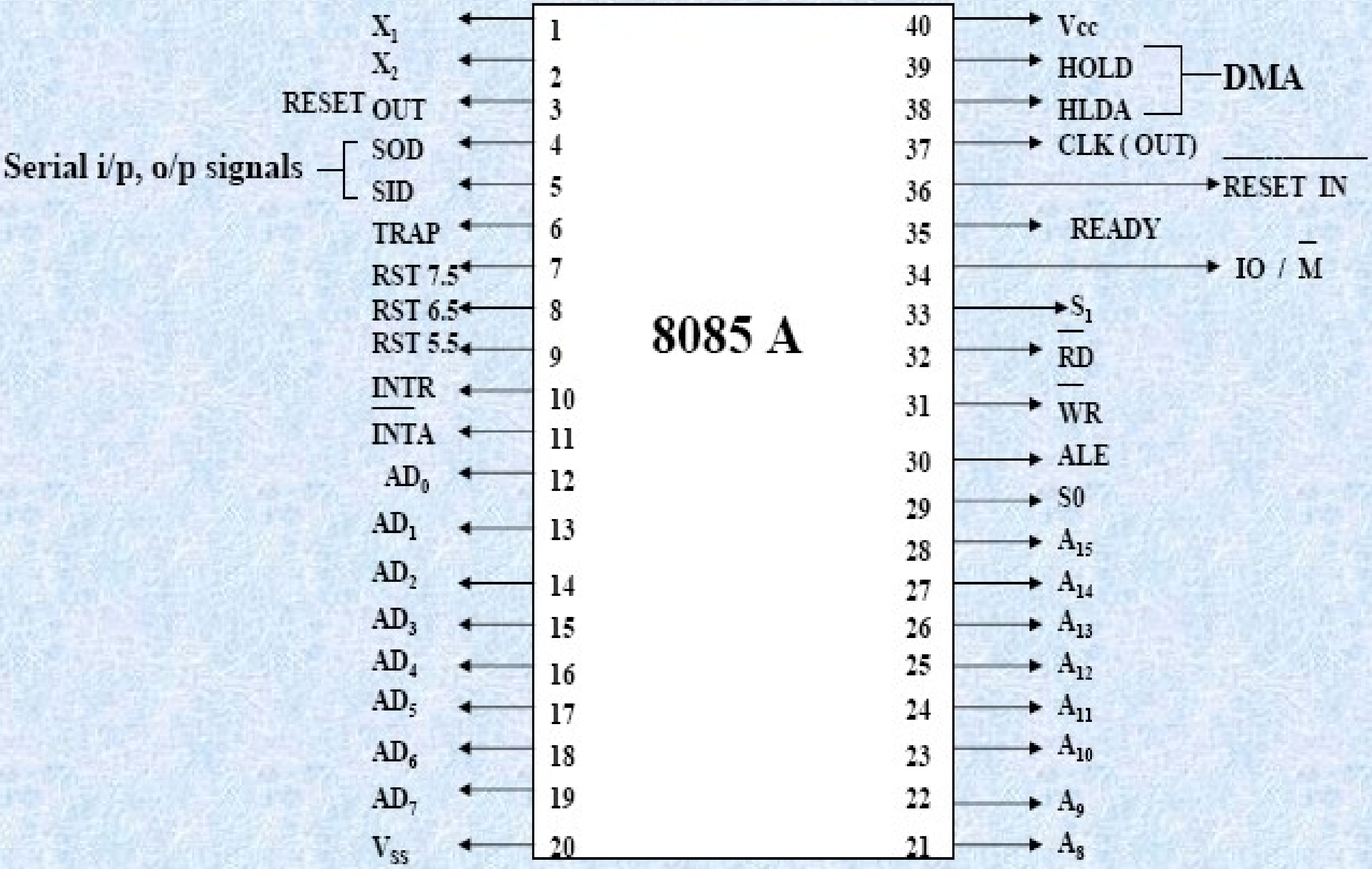
INTERNAL DATA OPERATION

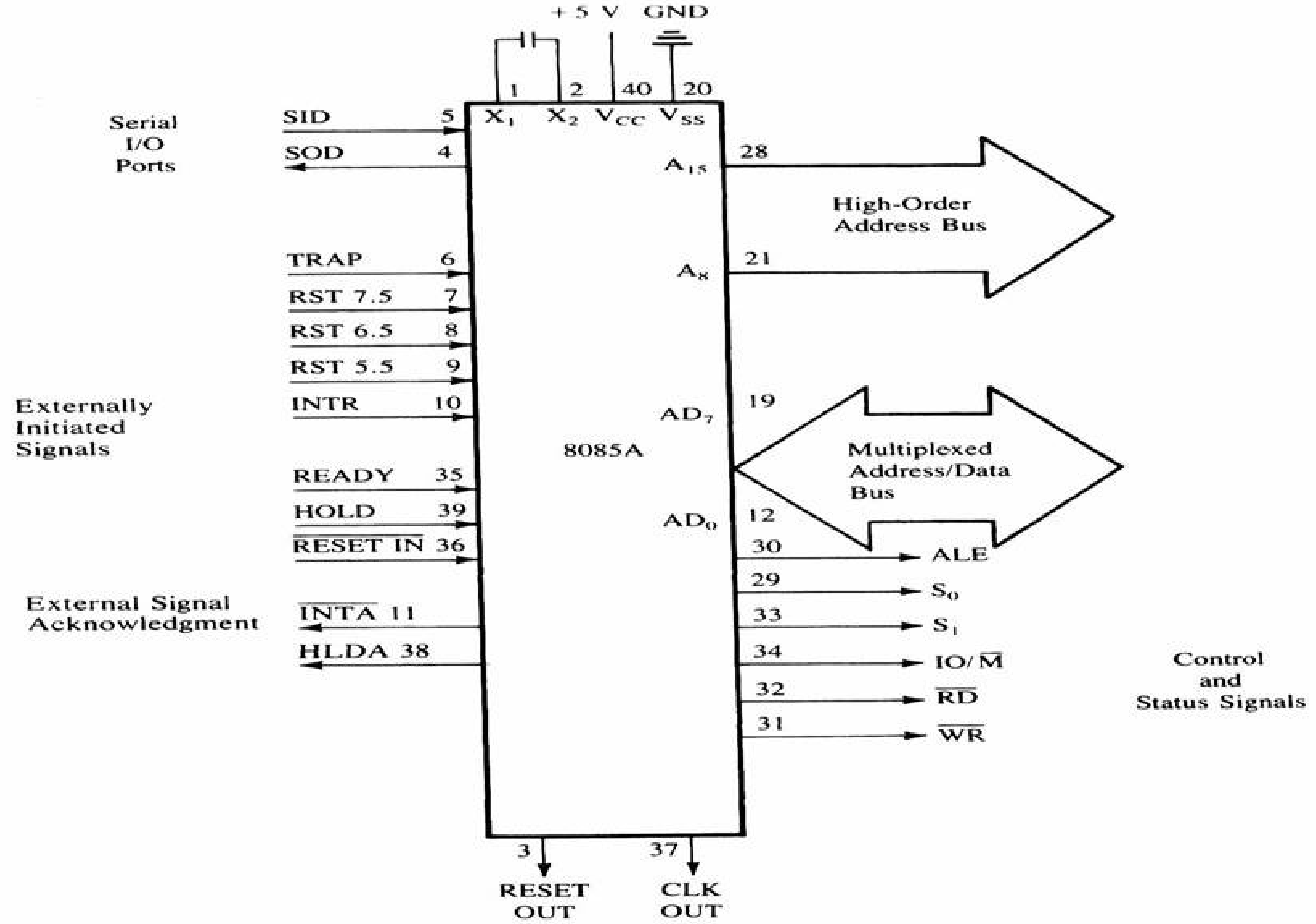
- Store 8 bit data
- Perform arithmetic and logical operation
- Test for condition
- Sequence the execution
- Store the data temp. during execution in the defined R/W memory location called stack

INTERNAL DATA OPERATION

MVI B ,76H	O6H	2000
	76H	2001
MVI A,34H	3EH	2002
	34H	2003
HLT	76H	2004

Pin Diagram of 8085





Signals and I/O Pins

CLASSIFICATION OF SIGNALS

- The 8085 is an 8-bit general purpose microprocessor that can address 64K Byte of memory.
- It has 40 pins and uses +5V for power.
  - The pins on the chip can be grouped into 6 groups:
    - Address Bus.
    - Data Bus.
    - Control and Status Signals.
    - Power supply and frequency.
    - Externally Initiated Signals.
    - Serial I/O ports
    - External acknowledgement signals

CLASSIFICATION OF SIGNALS

- **Data Signals:** Signals associated with data bus comes under this type.
- **Control and Status Signals**: Signals which are associated with timing and control unit such RW’, WR’ etc. comes under this type of signals
- **Interrupt Signals**: We know that signals like TRAP, RST 5.5 etc. are interrupt signals. Such signals come under this category.

CLASSIFICATION OF SIGNALS

- **Classification of Signals**
- The various signals in a microprocessor can be classified as
- **Power supply and Frequency signals**: Signals which aids in supplying power and generating frequency are associated with this type. Pins like Vcc and ground are classified under this type.
- **Address signals**: Signals associated with the lower order address bus and time multiplexed higher order address bus comes under this type of signals.
- .

CLASSIFICATION OF SIGNALS

- **Serial I/O signals:** These signals are used for giving serial input and output data. Signals like SID, SOD come under this category.
- **Acknowledgement Signals**: Signals like INTA’, HLDA acts as acknowledgement signal for 8085 microprocessor.

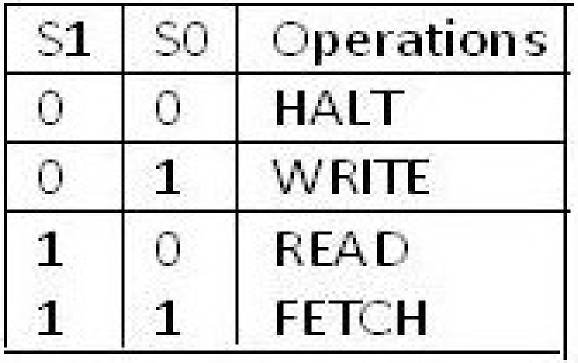
THE ADDRESS AND DATA BUSSES

- The address bus has 8 signal lines A8 – A15 which are unidirectional.
- The other 8 address bits are multiplexed (time shared) with the 8 data bits.
  - So, the bits AD0 – AD7 are bi-directional and serve as A0 – A7 and D0 – D7 at the same time.
    - During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits.
  - In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes.

THE CONTROL AND STATUS SIGNALS

- There are 4 main control and status signals. These are:
  - ALE: Address Latch Enable. This signal is a pulse that become 1 when the AD0 – AD7 lines have an address on them. It becomes 0 after that. This signal can be used to enable a latch to save the address bits from the AD lines.
  - RD: This is an active low signal. That is, an operation is performed when the signal goes low. This signal is used to control READ operation of the microprocessor. When this pin goes low the microprocessor reads the data from memory or I/O device.
  - WR: WR' is also an active low signal which controls the write operations of the microprocessor. When this pin goes low, the data is written to the memory or I/O device.

- IO/M: This signal specifies whether the operation is a memory operation (IO/M=0) or an I/O operation (IO/M=1).
- S1 and S0 : Status signals to specify the kind of operation being performed .Usually un-used in small systems.



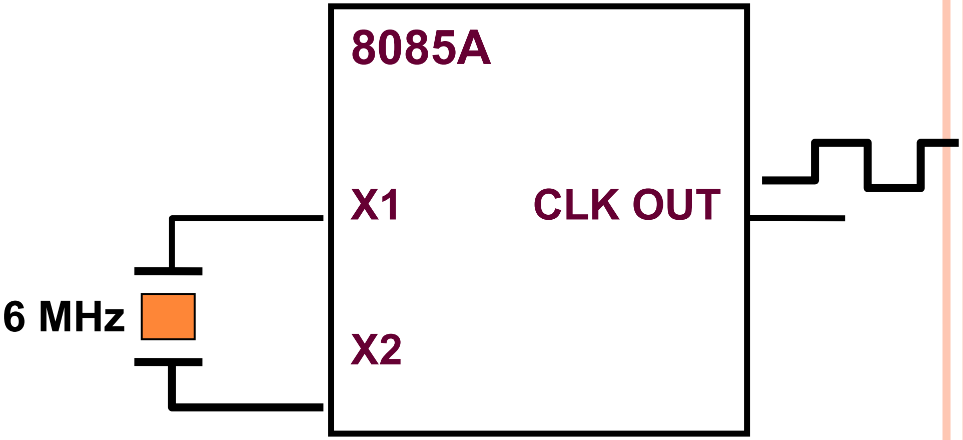
S1	S0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

FREQUENCY CONTROL SIGNALS

- There are 3 important pins in the frequency control group.
  - X0 and X1 are the inputs from the crystal or clock generating circuit.
    - The frequency is internally divided by 2.
      - So, to run the microprocessor at 3 MHz, a clock running at 6 MHz should be connected to the X0 and X1 pins.
  - CLK (OUT): An output clock pin to drive the clock of the rest of the system.

CLOCK PINS

- 8085 MPU has 3 pins that control or present the clock signal.
  - X1 and X2 pins determine the clock frequency.
  - CLK OUT is a TTL square-wave output clock.
- The CLOCK OUT is one-half the crystal frequency.



CONTROL AND STATUS SIGNALS.

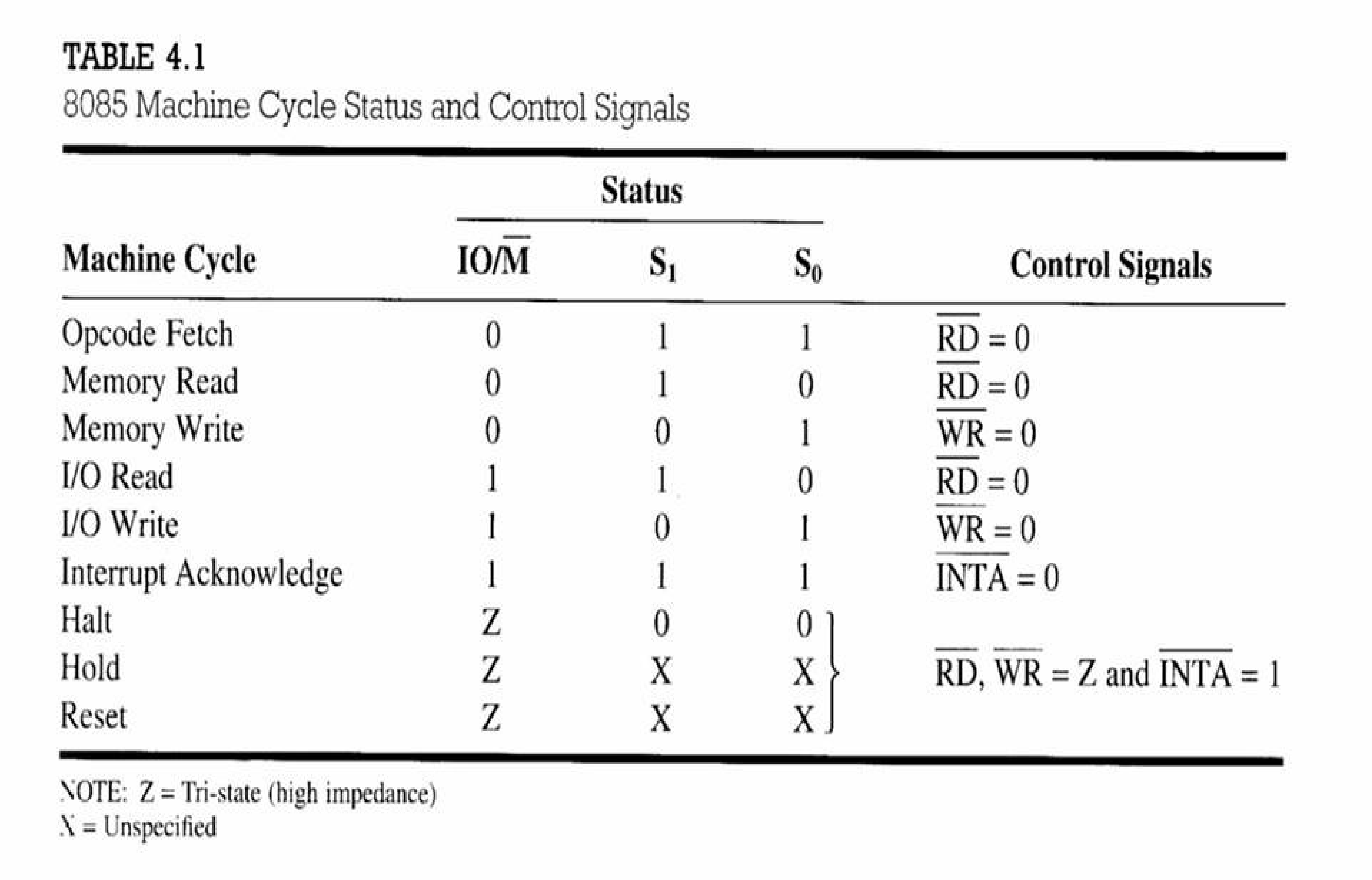


TABLE 4.1

8085 Machine Cycle Status and Control Signals

	Status			
Machine Cycle	**$IO/\overline{M}$**	**S_1**	**S_0**	**Control Signals**
Opcode Fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O Write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA}$ = 1
Hold	Z	X	X	
Reset	Z	X	X	

NOTE: Z = Tri-state (high impedance)
X = Unspecified

EXTERNALLY INITIATED OPERATIONS & INTERRUPTS SIGNALS

- **Ready**
  - The 8085 has a pin called RDY. This pin is used by external devices to stop the 8085 until they catch up.
  - As long as the RDY pin is low, the 8085 will be in a wait state.(to synchronize slower peripherals with MP)
- **Hold**
  - The 8085 has a pin called HOLD. This pin is used by external devices to gain control of the busses.
  - When the HOLD signal is activated by an external device, the 8085 stops executing instructions and stops using the busses.
  - This would allow external devices to control the information on the busses. Example **DMA**.

- **HLDA:**
- HLDA is the acknowledgment signal for HOLD. It indicates whether the HOLD signal is received or not. After the execution of HOLD request, HLDA goes low.
- **INTR:**
- INTR is an interrupt request signal. It has the lowest priority among the interrupts. INTR can be enabled or disabled by using software. Whenever INTR goes high the microprocessor completes the current instruction which is being executed and then acknowledges the INTR signal and processes it.

- **INTA’:**
- Whenever the microprocessor receives interrupt signal. It has to be acknowledged. This acknowledgement is done by INTA’. So whenever the interrupt is received INTA’ goes high.
- **RST 5.5, 6.5, 7.5:**
- These are nothing but the restart interrupts. They insert an internal restart function automatically.
- All the above mentioned interrupts are maskable interrupts. That is, they can be enabled or disabled using programs.

- **TRAP:**
- Among the interrupts of 8085 microprocessor, TRAP is the only non-maskable interrupt. It cannot be enabled or disabled using a program. It has the highest priority among the interrupts.
- PRIORITY ORDER (From highest to lowest)
- TRAP
- RST 7.5
- RST 6.5
- RST 5.5
- INTR

- **RESET IN’:**
- This pin resets the program counter to 0 and resets interrupt enable and HLDA flip-flops. The CPU is held in reset condition until this pin is high. However the flags and registers won’t get affected except for instruction register.
- **RESET OUT:**
- This pin indicates that the CPU has been reset by RESET IN’.

- **SID:**
- This pin provides serial input data. The serial data on this pin is loaded into the seventh bit of the accumulator when RIM instruction is executed.
- RIM stands for READ INTERRUPT MASK, which checks whether the interrupt is masked or not.
- **SOD:**
- This pin provides the serial output data. The serial data on this pin delivers its output to the seventh bit of the accumulator when SIM instruction is executed.

- **Vcc and Vss:**
- Vcc is +5v pin and Vss is ground pin.