4.1 Introduction :

In chapter 3, we had already developed microprocessor. It was our version. The same should be replica of Intel 8085. So let's start study of the same.

4.2 Features of Intel 8085 :

The features of 8085 include :

(1) It is an 8 bit microprocessor i.e. it can accept or provide 8 bit data simultaneously.

(2) It is a single chip, NMOS device implemented with 6200 transistors.

(3) It requires a single +5V power supply.

(4) It provides on chip clock generator, hence it does not require external clock generator, but it requires external tuned circuit like LC, RC or crystal.

(5) It requires two phase, 50% duty cycle, TTL clock. These clock signals are generated by an internal clock generator. (Refer Fig. 4.1).

(6) The maximum clock frequency is 3 MHz and minimum clock frequency is 500 KHz.

(7) It provides 74 instructions with the following addressing modes : register, direct, immediate, indirect and implied.

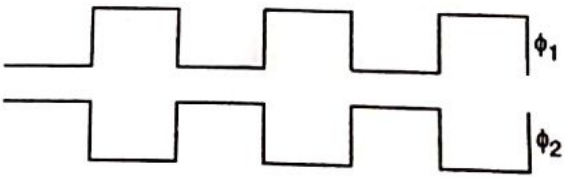

Fig. 4.1 : 2 Phase TTL clock

(8) The data bus is multiplexed with address bus, hence it requires external hardware to separate data lines from address lines.(This is the drawback of 8085).

(9) It provides 16 address lines, hence it can access 2^{16}= 64 K bytes of memory.

(10) It generates 8 bit I/O address, hence it can access $2^8 = 256$ input ports and 256 output ports.

(11) It performs the following arithmetic and logical operations :
8 bit, 16 bit binary addition, 2 digit BCD addition, 8 bit binary subtraction logical AND, OR, EXOR, complement and shift operations.

(12) It provides 5 hardware interrupts : TRAP, RST 5.5, RST 6.5, RST 7.5, INTR

(13) By providing external hardware one can increase interrupt capability of it.

(14) 8085 has capability to share its bus with external bus controller (Direct memory access controller); for transferring large amount of data from memory to I/O and vice versa.

(15) It provides one accumulator, one flag register, 6 general purpose registers and two special purpose registers.

(16) It provides status for advanced control signals. (Advanced control signals are used in large system).

(17) It can be used to implement three chip microcomputer (8085, 8155 and 8355).

(18) It provides two serial I/O lines viz SOD and SID, hence serial peripherals can be interfaced with 8085 directly.

4.3 Pin Diagram of 8085 :

The 8085A is an 8 bit general purpose microprocessor having 40 pins and works on +5V single power supply. Fig. 4.2 shows the pin diagram of 8085A. To study the pin diagram, we group the signals as shown in Fig. 4.3. They are as follows :

(1) Address bus
(2) Data bus
(3) Status signals
(4) Control signals
(5) Power supply
(6) Clock signals
(7) Interrupt signals
(8) Serial input output signals
(9) DMA request signals
(10) Reset signals.

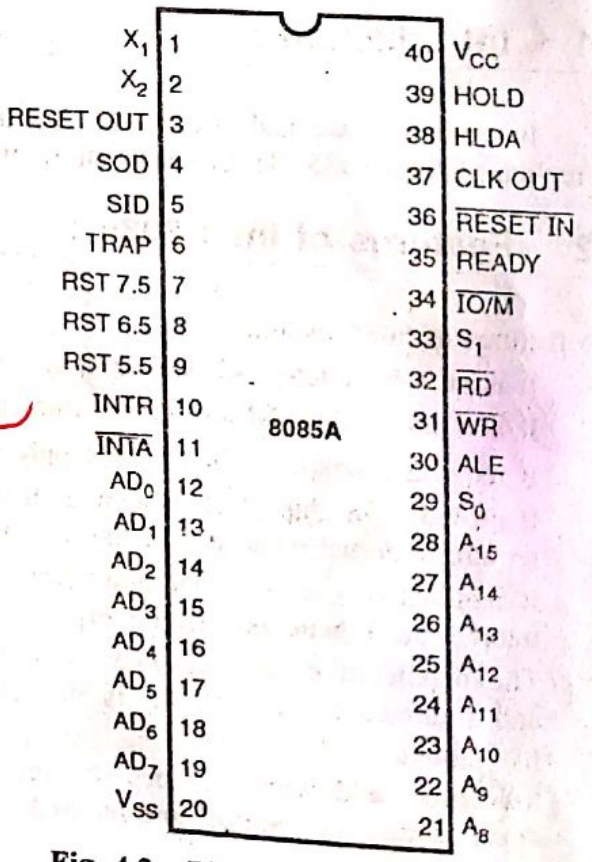

Fig. 4.2 : Pin diagram of 8085

Note : Observe pin configuration carefully. It matches with our microprocessor. Obviously, functionwise identical with our microprocessor.

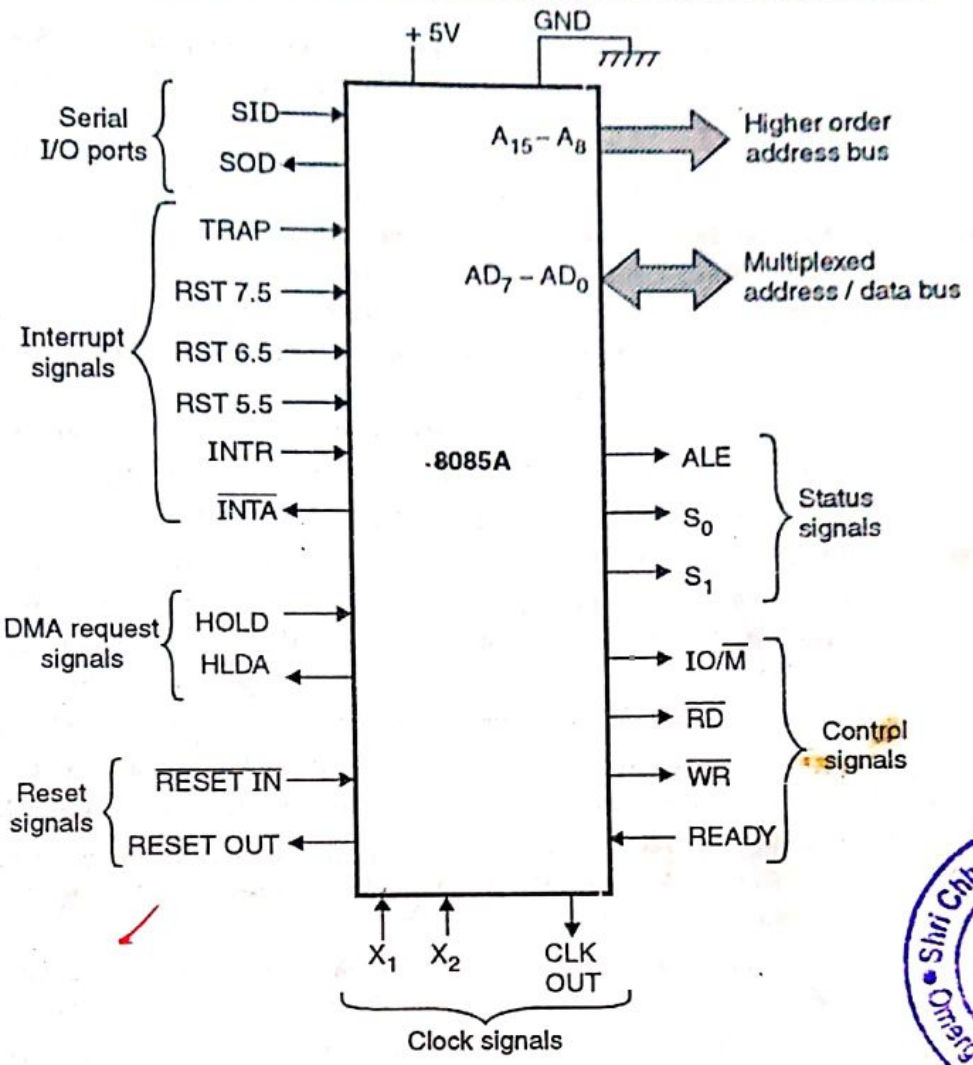

Fig. 4.3 : Groups of 8085 signals

	Symbol	Name and function
(1)	$A_8 - A_{15}$	**Address bus :** These are output, tri-state signals used as higher order 8 bits of 16 bit address. These signals are unidirectional meaning that the address is given by 8085 to select a memory or an I/O device.
(2)	$AD_0 - AD_7$	**Multiplexed address/data bus :** These signals are I/O tristatable signals. These pins provides multiplexed or time shared address and data bus. Address is lower order of total 16 bit address i.e. $A_7 - A_0$. Refer Fig. 4.4, as shown the bus works in conjuction with ALE. In earlier part it will output address. Address will remain there for finite time. After that it will vanish. In later part it is used as data bus; i.e. either for reading or writing.

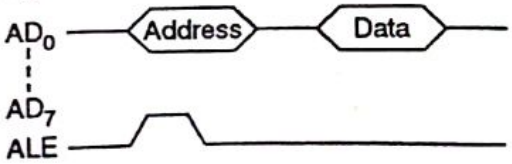

Fig. 4.4 : Multiplexed address/data bus

But external peripherals requires separate address and data, therefore to DEMULTIPLEX we use external latch. Thus advantage of multiplexed bus is, microprocessor requires less number of pins. But disadvantage is, we require extra IC (Latch) for demultiplexing. Second disadvantage is, due to multiplexing, little bit more time required for read/write operation(s).

(3) ALE

Address latch enable : This is an output signal, used to give information of $AD_0 - AD_7$ contents. It is a positive going pulse, generated when a new operation is started by a microprocessor. When pulse is HIGH, it indicates that the contents of $AD_0 - AD_7$ are address. When it is LOW, it indicates that the contents are data. (Refer Fig. 4.4)

The ALE signal is used to separate $AD_0 - AD_7$ (i.e. demultiplex) to $A_0 - A_7$ and $D_0 - D_7$. To do this separation a latch is connected to $AD_0 - AD_7$ lines. The latch is controlled by ALE signal.

(4) IO/$\overline{M}$

Input Output/memory : This is an output status signal, used to give information of operation to be performed with memory or I/O device. When $IO/\overline{M} = 0$ the microprocessor is performing a memory related operation and when $IO/\overline{M} = 1$ the microprocessor is performing an I/O device related operation. This is the signal which separates memory and I/O devices. To implement this; $IO/\overline{M}$ signal is combined with read and write control signals and two sets of signals are generated; one for memory and other for I/O device.

The function performed by $IO/\overline{M}$ can also be performed by using one address line. But in this case the number of I/O devices and memory which can be connected to 8085 will be halved.

Instead of having $IO/\overline{M}$ line, the manufacturer could have increased one address line. But the main problem with this approach is that the total number of address lines now becomes 17. All the internal sources of address lines are only 16 bits so it is not possible to generate 17 bit address using 16 bits. So instead of adding one address line for $IO/\overline{M}$, it is used to address two separate sections i.e. memory and I/O ports.

(5) S_1 and S_0

Status signals S_0 and S_1 : These are output status signals used to give information of operation performed by microprocessor. These are generally not used in small systems but can be used to generate advanced control signals for large systems.

The S_0 and S_1 lines specifies four different conditions of 8085 machine cycles. These 4 cycles are as follows :

(1) OPCODE fetch (Instruction read from memory) $S_0 = 1$, $S_1 = 1$
(2) Read (Data read from memory) $S_0 = 0$, $S_1 = 1$
(3) Write $S_0 = 1$, $S_1 = 0$ (4) Halt $S_0 = 0$, $S_1 = 0$

When S_0 and S_1 is combined with $IO/\overline{M}$ we get status of all the machine cycles (operations) performed by 8085 as shown in Table 1

(6) $\overline{RD}$

Read : This is an active low, output control signal used to read data from memory or an I/O device. To read data from device, microprocessor selects a device, make data bus available for data transfer and then generates signal RD

(7) $\overline{WR}$ to read data from selected device.

Write : This is an active low, output control signal used to write data to memory or an I/O device. To write data to device, microprocessor selects a device and transfers data on data lines and then generates signal $\overline{WR}$. The signal $\overline{WR}$ indicates that the contents of data bus is to be written in the selected device.

Table 1 : 8085 machine cycle status and control signals

Machine cycle status				
IO / $\overline{M}$	S_1	S_0	Status	Control signal used
0	1	1	OPCODE fetch	$\overline{RD}$ = 0
0	1	0	Memory read	$\overline{RD}$ = 0
0	0	1	Memory write	$\overline{WR}$ = 0
1	1	0	I/O read	$\overline{RD}$ = 0
1	0	1	I/O write	$\overline{WR}$ = 0
1	1	1	Interrupt acknowledge	$\overline{INTA}$ = 0
Z	0	0	Halt	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA}$ = 1
Z	X	X	Hold	
Z	X	X	Reset	
Note : Z-Tristate (High impedance) condition X-Unspecified condition				

(8) **READY** This is an active high input control signal. It is used by microprocessor to detect whether a peripheral has completed the data transfer or not. If ready pin is HIGH, the microprocessor will complete the operation and proceed for next operation. But if ready pin is LOW (i.e. peripheral has not yet completed the operation), microprocessor will **WAIT** until it goes HIGH. The main function of this pin is to synchronize slower peripheral to the faster microprocessor.

(9) **TRAP** This is an active high level, edge triggered, non maskable, highest priority interrupt. When TRAP line is active microprocessor performs internal restart automatically at address 0024. The net effect of TRAP is, it transfers program control at address 0024.

(10) **RST 7.5, RST 6.5 and RST 5.5** **Restart interrupts :** These are active high, edge (RST 7.5) or level (RST 6.5 and RST 5.5) triggered maskable interrupt. The priorities of these are TRAP, RST 7.5, RST 6.5, RST 5.5. When RST 7.5, RST 6.5, RST 5.5 is active microprocessor performs internal restart automatically at address 003C, 0034, 002C respectively. The net effect is, it transfers program control at address, specified above.

(11) **INTR and $\overline{INTA}$** **Interrupt request :** INTR is an active high, level triggered general purpose interrupt. When INTR is active, microprocessor generate an interrupt acknowledge signal $\overline{INTA}$. The INTR and $\overline{INTA}$ signals are basically used to expand interrupt system to more than 5.

> Note : The details of TRAP, RST 7.5, RST 6.5, RST 5.5, INTR and $\overline{INTA}$ are covered in chapter 10.

(12) HOLD and HLDA HOLD is an active high, input signal used by other controller to request microprocessor about use of address, data and control buses. When the microprocessor receives HOLD request signal, it completes current machine cycle and will relinquish use of the bus. To implement this, the microprocessor will tristate its address, data and control signals ($\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$); and stop using them. The microprocessor in response to HOLD generates a signal that acknowledges the requesting device by HLDA signal. When HLDA is active it indicates that microprocessor has received HOLD request and will relinquish the buses in next clock cycle. The other controller will use buses and upon completion of work will deactivate HOLD signal, because of this, microprocessor will also make HLDA LOW. The microprocessor takes control of buses half clock cycle after HLDA goes LOW.

(13) $\overline{RESET\ IN}$ This is an active low, input reset signal. When $\overline{RESET\ IN}$ = 0, it clears program counter i.e. 0000 and makes address, data and control lines tristated. After reset the status of internal register and flags are unpredictable. The CPU is held in the reset condition as along as RESET IN is applied. After reset the microprocessor starts executing instructions from 0000 H onwards. Thus 0000 H is also referred as RESET address of microprocessor.

(14) RESET OUT This is an active high, output signal used to indicate that the microprocessor is resetted. This signal is used as system reset, to reset other devices connected in the system.

(15) SID **Serial input data :** This is an active high, serial input port pin, used to accept serial 1 bit data under software control. When a RIM instruction is executed, the SID pin data is loaded in bit D_7 of accumulator.

(16) SOD **Serial output data :** This is an active high, serial output port pin, used to transfer serial 1 bit data under software control. When a SIM instruction is executed the SOD pin is set or reset depending on D_7 and D_6 bits of accumulator. The details of SID and SOD are covered in chapter 14.

(17) X_1 X_2 These are clock input signals, connected to crystal, LC or RC network. The X_1 and X_2 pins drive the internal clock generator circuit. The frequency is divided by 2 and used as operating frequency. Generally, the 6.014 MHz crystal is connected to X_1 and X_2. So the operating frequency is 3.07 MHz. i.e. ($\div 2$) 3.007

(18) CLK OUT This is an output signal, used as a system clock. The internal operating frequency i.e.($\div 2$), is available on CLK OUT pin.

(19) V_{CC} and V_{SS} Power supplty V_{CC} – +5V,
V_{SS} – Ground reference

4.4 8085 CPU Architecture :

Alas ! we have reached to architecture. Hold your breath and observe it carefully. ..... And you have won the game. Except few points you are acquainted with each and every block. Compare Fig. 3.23 in chapter 3, with Fig. 4.5.

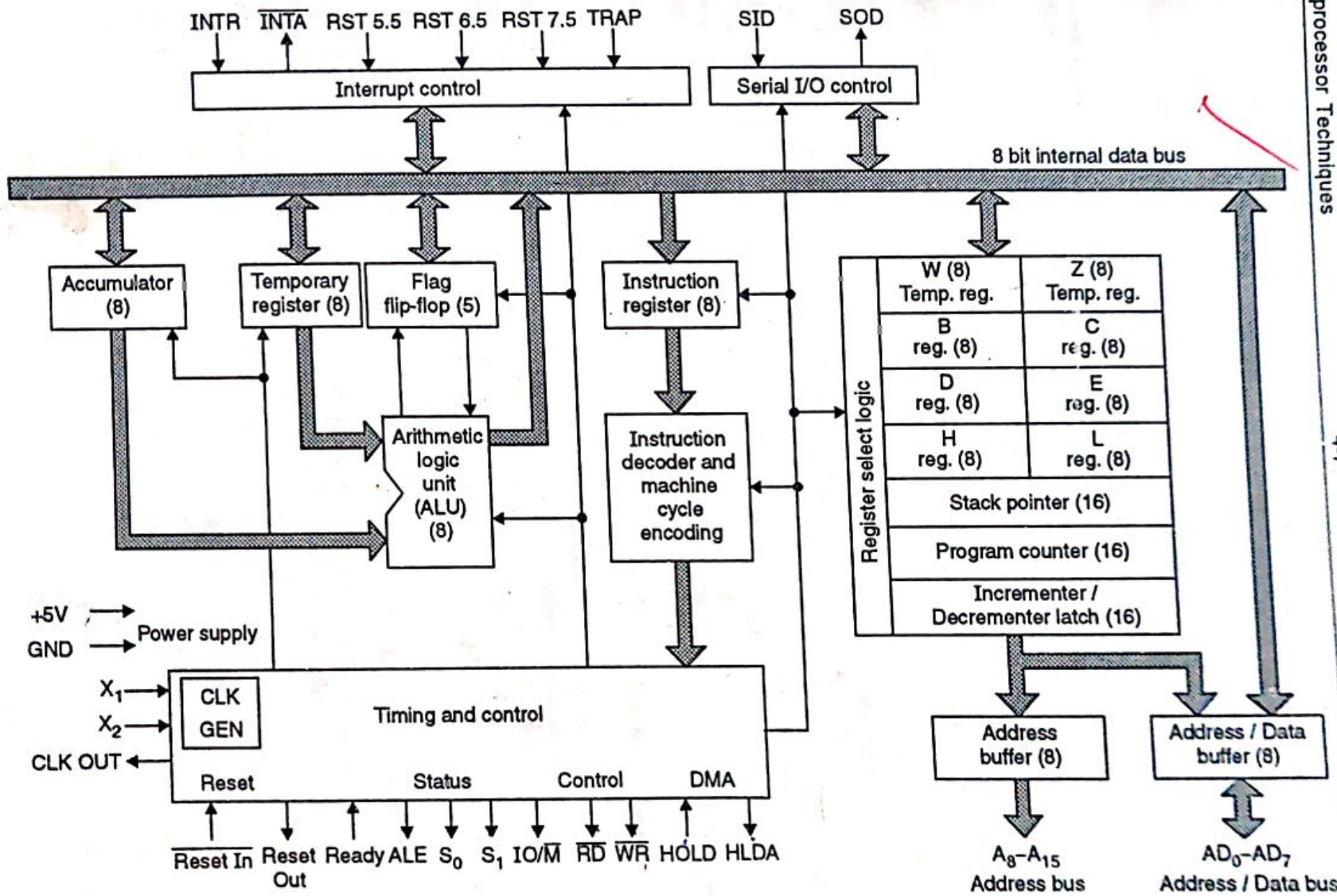

Fig. 4.5 : Functional block diagram of 8085

(1) Output store register in our microprocessor, has been replaced by temporary register.
(2) Two more temporary registers i.e. W and Z are added by Intel corporation.
(3) Remaining full architecture is same.

Eventhough we know everything, we will again go through each and every block and will make our concepts, more strong.

The architecture is divided in different groups as follows :

(1) Arithmetic and logical group (2) Register group (3) Interrupt control group
(4) Serial I/O control group (5) Instruction register, decoder, timing and control group.

4.5 Arithmetic and Logical Group :

This group consists of ALU, accumulator or A register, temporary register and flag register.

4.5.1 ALU :

The ALU performs arithmetic operations such as addition, subtraction and logical operations such as ANDing, ORing, EXORing, etc. This block can be well visualised by using ALU IC 74181. In it the arithmetic and logical operations are performed on input data A and B. Both the A and B inputs are 4 bits each. The data is provided at A and B inputs and a function desired is selected by using select lines. Similarly in 8085, to ALU the inputs are provided by accumulator and a temporary registers; both are of 8 bits. The operation to be performed on these data bytes is selected by control logic. The ALU will perform the operation and output result on internal data bus. The ALU is of 8 bits so at a time operation on 8 bit data can only be performed.

4.5.2 Accumulator :

The accumulator is a 8 bit general purpose register connected to internal data bus and to ALU. It is also called as A register. As it is connected as one of the inputs to ALU, it is used in most of the arithmetic and logic instructions. After performing an operation, the ALU places its result on internal data bus, from there it is generally stored in accumulator. So accumulator is integral part in performing different operations. Because of its general purpose nature it can also be used to store 8 bits of data temporarily.

4.5.3 Temporary Register :

The other input to ALU is given by temporary register. This register is not available for user. It is only used internally by microprocessor. To perform arithmetic and logical operations microprocessor assumes one data is available in accumulator and takes another data from other register (depends on instruction) into temporary register and then performs operation on the two data bytes.

Example : ADD B instruction, adds A reg and B reg contents, the result is stored in A reg. In this case one data is available in A reg. The other data is available in B register, this other data from B register is transferred to temporary register and then add operation is performed on that. This temporary register is also used for other operations such as register to register data transfer, etc.

4.5.4 Flag Register (PSW) :

The flag register is nothing but a group of flip-flops used to give status of different operations result. The flag register is connected to ALU. When an operation is performed by ALU, the result is transferred on internal data bus and status of result will be stored in flip flops. It is clear that for all other operations, except related to ALU, the flags doesn't get affected. It will only give status if an operation is performed in ALU.

> **Note :** Output of flag register is not directly accessible to the user. But we can get the flag register and manipulate INDIRECTLY, with software techniques. But wait till we study, instruction set.

The different flags and their positions in flag register are as shown in Fig. 4.6.

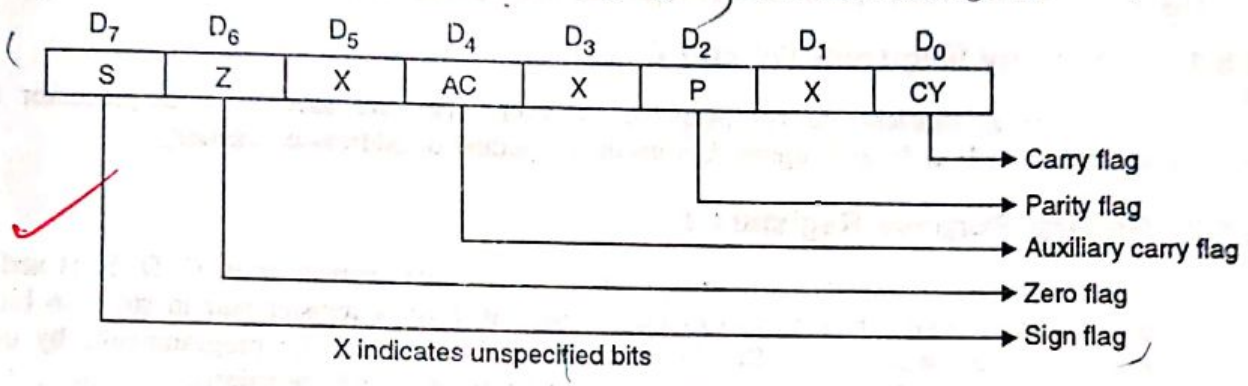

Fig. 4.6 : 8085 flag register

(1) **CY** **Carry flag :** If an operation performed in ALU, generates a carry from D_7 bit, the CY flag is SET. It works as 9th bit for addition and as borrow flag for subtraction. If there is no carry/borrow, out of MSB bit, i.e. D_7, of the result, CY flag is RESET.

(2) **AC** **Auxiliary carry flag :** If an operation performed in ALU generates a carry from lower nibble (i.e. D_0 to D_3) to upper nibble (i.e. D_4 to D_7) the AC flag is set i.e. a carry given by D_3 bit to D_4 is a AC flag.

This is not a general purpose flag, it is only used internally by microprocessor to perform binary to BCD conversion. It is not available for programmer for any decision making.

(3) **Z** **Zero flag :** If an operation in ALU results in zero, the zero flag is SET. If the result is not zero, the zero flag is RESET.

(4) **S** **Sign flag :** In sign magnitude format, the sign of a number is indicated by MSB bit. If MSB bit = 0, the number is positive and if MSB bit = 1, the number is negative. In 8085 MSB bit is D_7 bit. The sign flag is exact replica of D_7 bit of the result. If $D_7 = 1$, the flag is set and if $D_7 = 0$, the flag is reset. This flag can be used to perform operation on signed numbers.

(5) **P** **Parity flag :** This bit is used to indicate the parity of the result. If the result contain even number of 1's this flag is set. If the result contains odd number of 1's this flag is reset. i.e. by insertion of flag bit microprocessor maintains odd parity for result.

> **Note :** In 8085 overflow flag is not present. Basically, the overflow flag and carry flag are not same. The overflow flag is used by other processor Ex. 8086. It is used to indicate overflow condition. It is set if the result of a signed operation is too large to fit in the number of bits available. When we want to represent signed numbers (i.e. positive and negative numbers) we use 2's complement sign magnitude form. In this form the MSB is used as a sign bit and remaining 7 bits are used as magnitude. So for an 8 bit number the allowed range of numbers is + 127 to –128. When result of any arithmetic operation exceeds this limit, this indicates an overflow condition. In 8085 overflow condition can be checked by checking D_7 bit of operands and result.

4.6 Register Group :

This group consists of 3 types of registers :
(i) Temporary registers (ii) General purpose registers (iii) Special purpose registers.

4.6.1 Temporary Registers (W and Z) :

The W and Z registers are not available for user. They are used by microprocessor for internal operations such as to store operand, immediate operand or address of memory.

4.6.2 General Purpose Registers :

The 8085 contains 6 general purpose registers of 8 bits each, named as B, C, D, E, H and L. These can be used to store 8 bits of data or can be used to form a register pair to store 16 bit of data. The register pairs available are BC, DE and HL. These registers are programmable by user. User can store any data in these registers and use it to perform different operations.

4.6.3 Special Purpose Registers :

The 8085 contains 3 special purpose registers such as program counter, stack pointer and incrementer / decrementer latch.

(a) Program counter : This is a 16 bit register used for execution of program. This register always points to address of memory from where the next instruction is to be fetched and executed. When microprocessor performs one operation of taking instruction i.e. fetching, the PC contents are automatically incremented by one to point to next location. In this way, PC keeps the track for execution of program. ***Upon reset PC contents are set to 0000 H, so after reset operation, microprocessor will start execution of program from 0000 H onwards.***
The program counter is of 16 bits. The main reason behind this is that the 8085 contains 16 address lines. By using 16 address lines one can select any memory location in the memory map of 8085.

(b) Stack pointer : This is a 16 bit register used to define the stack starting address. Stack is a reserved portion of memory where register pair information can be stored or taken back under software control. The stack pointer is used to keep track of data stored on stack. [Refer section 3.8.7 in chapter 3 for operation in stack].

(c) Incrementer/decrementer latch : This 16 bit register is used to increment or decrement the contents of PC and SP registers. In coordination with these registers, two buffers are used.

(A) Address buffer : This is an 8 bit unidirectional buffer used for A_8 to A_{15} address lines. These are used to output higher order address on A_8 to A_{15}. When they are not in use or under certain conditions such as reset, hold, halt, this buffer is used to tristate A_8 to A_{15} address lines.

(B) Address/data buffer : This is an 8 bit bidirectional buffer used for address/data. The address/data signals are multiplexed on AD_0 to AD_7 lines. In earlier part it is used to output lower order address A_0 to A_7 and in later part it is used to input or output data D_0 to D_7. The address is taken from address lines and data is taken or transferred on internal data bus. Under certain condition such as reset, hold, halt this buffer is used to tristate AD_0 to AD_7 address/data lines. The various sources of addresses for the address register includes program counter, stack pointer, temporary registers, BC pair, DE pair and HL pair. These are as shown in Fig. 4.7.

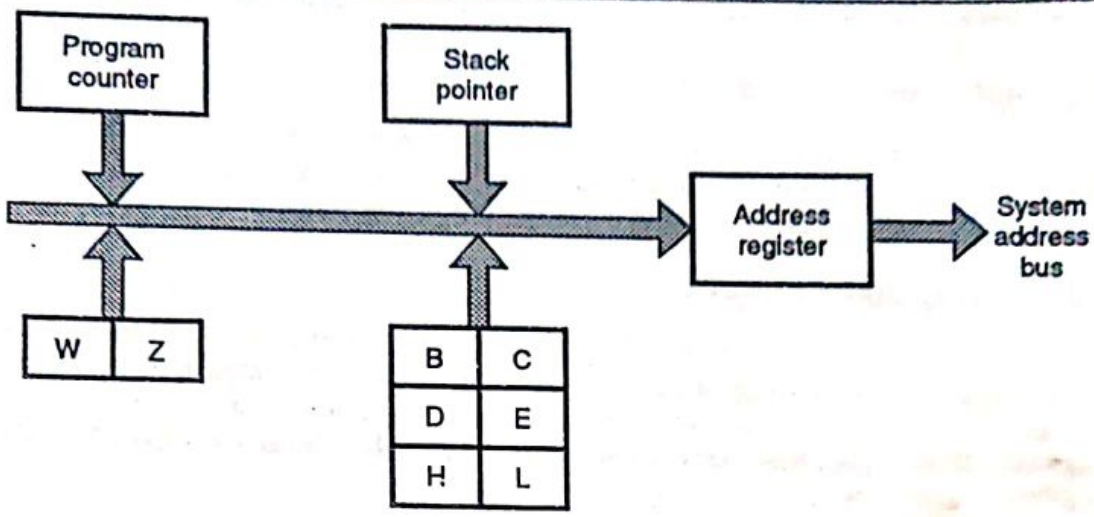

Fig. 4.7 : Various sources of addresses

> **Note :** As mentioned to you, PC points to program memory and SP points to data memory. In short, any register that is used as a pointer, to point memory, have to generate *"Address"* ONLY. Same way, as BC, DE and HL pairs can be used as *"Data pointer (Pointing to data memory)"*, it will output address.

4.7 Interrupt Control :

This block accepts different interrupt request inputs such as TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR. When a valid interrupt request is present, it informs control logic to take action in response to each signal.

> **Note :** Presently this is enough. At this stage too much involvement in interrupt part will not be much comfortable.

4.8 Serial I/O Control Group :

The data transferred on D_0 to D_7 lines is a parallel data, but under certain condition it is advantageous to use serial data transfer. 8085 implements this by using SID and SOD signals. The data on these lines is accepted or transferred under software control by serial I/O control block. In 8085 to perform serial data transfer there are two special instructions RIM and SIM.

4.9 Instruction Register, Decoder and Control Group :

4.9.1 Instruction Register :

When an instruction is fetched from memory it is loaded in instruction register. These contents are then provided to decoder for decoding. This register is only activated when an instruction code or OPCODE is available on internal data bus. It is non programmable register i.e. not available for programmers use. Remember it accepts only OPCODE of instruction, operands are not accepted by this register.

4.9.2 Instruction Decoder :

This accepts a bit pattern from instruction register, decodes it and gives the decoded information to control logic. The information includes what operation is to be performed, who is going to perform it, how many operand bytes the instruction contains, etc.

4.9.3 Timing and Control :

This is a control section of 8085. This accepts information from instruction decoder and generates microsteps to perform it (so 8085 is called as microprogrammed). In addition to this, the block accepts clock inputs, performs sequencing and synchronising operations. The synchronization is required for communication between microprocessor and peripheral devices. To implement this, it uses different status and control signals.

4.10 Questions from Previous Examination Papers :

Q. 1 Draw the functional architecture diagram of 8085 microprocessor neatly labelling the data paths, registers etc. Briefly indicate the function of each.
(Nov. 99, 16 Marks) (Dec. 2001, 12 Marks)

Ans. : Refer section 4.4.

Q. 2 Draw and explain the block diagram of 8085 microprocessor and label all the registers.
(June 2000, 10 Marks)

Ans. : Refer section 4.4.

Q. 3 What is the use of program counter and stack pointer in 8085 ? **(June 2000, 4 Marks)**

Ans. : Refer section 4.6.3.

Q. 4 Explain the function of RESET, READY and HOLD pins of 8085. **(June 2000, 4 Marks)**

Ans. : Refer section 4.3.

Q. 5 Explain the ***'Flag register'*** of 8085 microprocessor. **(June 2000, 4 Marks)**

Ans. : Refer section 4.5.4.

Q. 6 Draw and explain the functional block diagram of 8085 microprocessor. Label various registers.
(May 2001, 8 Marks)

Ans. : Refer section 4.4.

Q. 7 Explain the functioning of the following pins of 8085 microprocessor. Indicate their activation status :

(i) RST 7.5 (ii) ALE (iii) Ready (iv) IO / $\overline{M}$ (v) Reset. **(May 2001, 8 Marks)**

Ans. : Refer section 4.3.

Q. 8 What is reset address of 8085 microprocessor ? **(May 2001, 2 Marks)**

Ans. : Refer section 4.3.

Q. 9 What is the significance of ALE pin in 8085 microprocessor ? **(Dec. 2001, 2 Marks)**

Ans. : Refer section 4.3.

❑❑❑