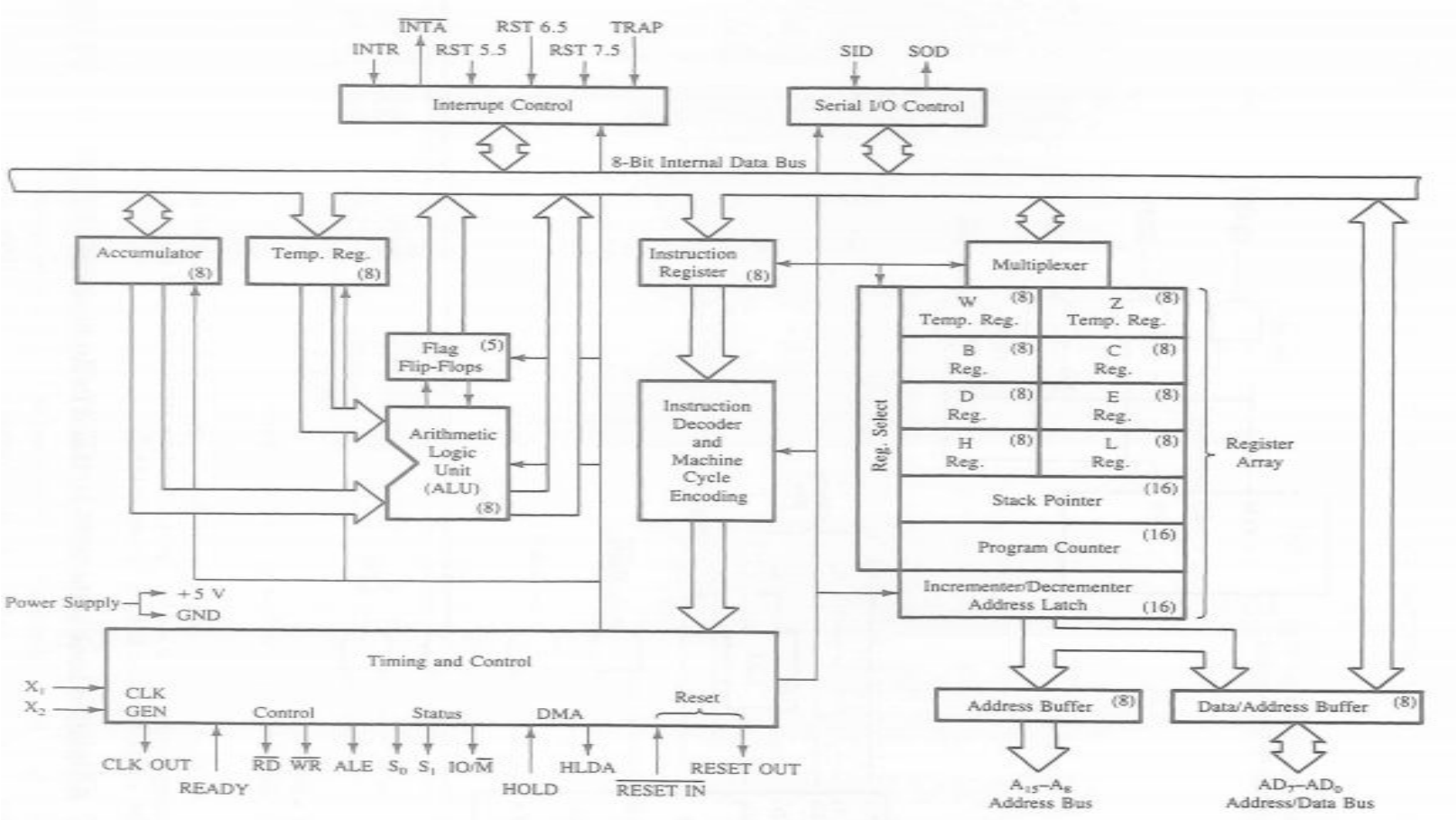
Microprocessor 8085 Instruction Set

Internal Architecture (functional block diagram) of 8085



8085 Architecture .. cont..

1. Register Array : 14 registers (12- 8bit registers , 2 - 16bit registers)

   They are classified into 4 types as

(a) General purpose register[B,C,D,E,H,L]
(b) Special purpose register [Accumulator (A), Instruction Register and Flag register]
(c) Temporary Register [W, Z, temporary data register]
(d) Pointer Register or special purpose [PC, SP]

Register array- general purpose registers , special purpose register

General purpose registers

- B, C, D, E, H, L are 8 bit registers
- Can be used as 16 bit register pairs when combined as BC, DE, HL
- Used to store the intermediate data and result
- H and L can be used to hold data as well as memory address.

Special purpose registers(A, Instruction register, flag register)

Accumulator (A)

- 8 bit register
- All the ALU operation are performed with reference to the contents of Accumulator
- Result of an operation is stored in A
- Store 8 bit data during I/O transfer

Instruction Register:

- When an instruction is fetched from memory, it is loaded in IR. then transferred to the decoder for decoding
- It is not programmable and can not be accessed through any instructio.
- Used by MPU internally.

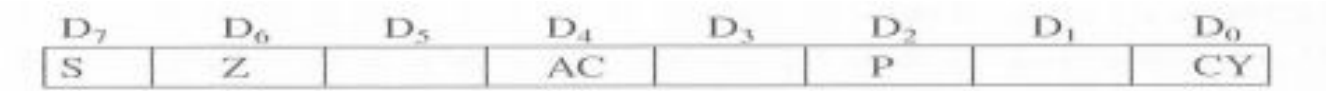
Slide 13

special purpose register - cont...

Flag Register (F)

- 8-bit register
- Indicates the status of the ALU operation
- ALU includes 5 flip flop, which are set or reset after an operation according to data conditions of the result in the accumulator.

The bit position of flag is given as:



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

S= sign bit

Z= zero bit

AC = Auxiliary carry

Slide 13

P = Parity

C = Carry

Flag register .. cont..

Flag	Significance
C or CY (Carry)	CY is set when an arithmetic operation generates a carry out, otherwise it is 0 (reset)
P (Parity)	**P= 1;** if the result of an ALU operation has an even number of 1's in A; **P= 0;** if number of 1 is odd.
AC (Auxiliary carry)	Similar to CY, AC= 1 if there is a carry from D_3 to D_4 Bit AC= 0 if there is a no carry from D_3 to D_4 Bit (not available for user)
Z(zero)	Z = 1; if result in A is 00H 0 otherwise
S(Sign)	S=1 if D_7 bit of the A is 1(indicate the result is -ive) S= 0 if D_7 bit of the A is 0(indicate the result is +ive)

Temporary Register

Temporary register**[W, Z, Temporary data register]**

W and Z register

- 8 bit
- Used to hold temporary addresses during the execution of some instructions

Temporary data register:

- 8 bit
- Used to hold temporary data during the ALU operations

Pointer register or special purpose [PC, SP]

Stack Pointer (SP)

- 16 bit register
- Holds the address of the data present at the top of stack memory
- Hold the content of PC when subroutines are used
- When there is a subroutine call or on an interrupt, i.e pushing the return address on a jump, and retrieving it after the operation is complete to come back to its original location

Program Counter (PC)

- 16 bit register.
- Used for execution of program
- Contain the address of the next instruction to be executed after fetching the instruction it is automatically incremented by 1

Instruction set 8085

An instruction is designed to do a specific function

The entire group of instructions that a microprocessor supports is called Instruction set

8085 has 246 instructions

Each instruction is represented by an 8-bit binary value

These 8-bits of binary value is called Op-code

Instruction types

Based on size :

1 -byte instruction - 8 bits

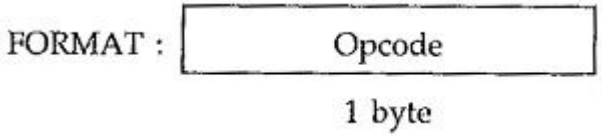
2- byte instruction - 16 bits

3 - byte instruction - 24 bits

Opcode format

Label	Memory address	opcode	operand

One byte instruction



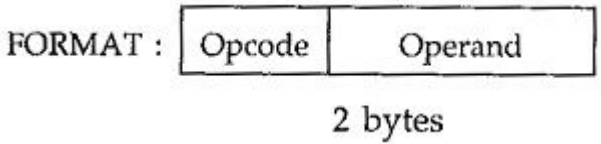
Example MOV A, B = 78H = 0111 1000

opcode	operand
MOV	A,B
78	

Opcode	Operand	Hex Code
MOV	C,A	4F

0100 1111

Two byte instruction

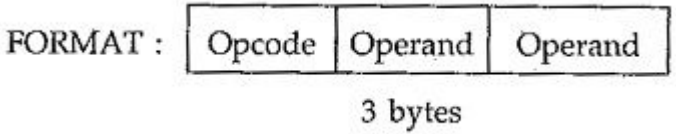


Example MVI B, 02H = 06 02H

opcode	operand
MVI B	02
06	02

06 02 = 0000 0110 0000 0010

Three byte instruction



Example JMP 6200H = C3 62 00H

opcode	operand	operand
JMP	62	00
C3	62	00

C3 62 00 = 1100 0011 0110 0010 0000 0000

8085 identifies all operations, registers and flags with a specific code. For example

Registers	Code
B	0 0 0
C	0 0 1
D	0 1 0
E	0 1 1
H	1 0 0
L	1 0 1
M (Memory)	1 1 0
A	1 1 1

Table 2.1(a)

Register Pairs	Code
BC	0 0
DE	0 1
HL	1 0
AF or SP	1 1

Table 2.1 (b)

Instruction formats

Instructions has two parts
Opcode (operation code) - first part explains the task or operation to be performed
Operand - the second part is the data to be operated on. This can be registers, data, or memory locations

type	task	op-code	operand	Binary code	Hex code
One byte	Copy the contents of the accumulator in the register C	MOV	C,A	0100 1111	$4F_H$
Two byte	Load an 8-bit data byte in the accumulator.	MVI	A,DATA	0011 1110 DATA	3E (first byte DATA (second byte
Three byte	Transfer the program sequence to the memory Location 2085H	JMP	2085H	1100 0011 1000 0101 0010 0000	C3 85 20

Types of Instruction

Since the 8085 is and 8-bit device it can have up to 2^8 (256) instructions.

- However, the 8085 only uses 246 combinations that represent a total of 74 instructions.

Types of instruction sets

- Data transfer operations
- Arithmetic operations
- Logic operations
- Branch operations
- Machine control operations

Instruction set

Data Transfer operation

These operations simply COPY the data from the source to the destination.

They transfer:

- Data between registers.
- Data Byte to a register or memory location.
- Data between a memory location and a register.
- Data between an I/O Device and the accumulator.

The data in the source is not changed.

Data transfer instructions never affect the flag bits.

e.g. LDA, STA, MOV, LDAX, STAX, MVI, LXI etc

Instruction set- cont..

Arithmetic operation

- These instruction perform addition, subtraction and compare operations.
- These operations are always performed with accumulator as one of the Operands.
- The status of the result can be verified by the contents of the flag register.

Addition: Any 8-bit number, or the contents of a register or the contents of a memory location can be added to the contents of the accumulator and the sum is stored in the accumulator. The instruction DAD is an exception; it adds 16-bit data directly in register pairs. Ex ADD, ADI

Subtraction - Any 8-bit number, or the contents of a register, or the contents of a memory location can be subtracted from the contents of the accumulator and the results stored in the accumulator. Subtraction is done by 2's compliment method and set carry flag to indicate borrow. SUB. SBI

Increment/Decrement - The 8-bit contents of a register or a memory location can be incremented or decremented by 1. Similarly, the 16-bit contents of a register pair (such as BC) can be incremented or decremented by 1. INR,DCR.

Instruction set- cont..

Logical operation

Perform 8-bit basic logical operations with the content of the accumulator

Logical instructions also modify the flag bits.

Op-codes for logical instructions include ANA, ANI, ORA, ORI, XRA,XRI, CMA, CMC, RAL, RLC, RAR, RRC, CMP, CPI etc.

AND, OR Exclusive-OR - Any 8-bit number, or the contents of a register, or of a memory location can be logically AND, Or, or Exclusive-OR with the contents of the accumulator. The results are stored in the accumulator.

Rotate- Each bit in the accumulator can be shifted either left or right to the next position.

Compare- Any 8-bit number, or the contents of a register, or a memory location can be compared for equality, greater than, or less than, with the contents of the accumulator.

Complement - The contents of the accumulator can be complemented.

Instruction set- cont..

Branch operations

These instructions are used to transfer the program control:

- To jump from one memory location to any other memory location within a program
- From one program to another program called as a subroutine.
- Alters the sequence of program execution either conditionally or unconditionally

Unconditional branch instructions- Transfer the program to the specified label or address JMP unconditionally i.e. without satisfying any condition.

Unconditional Program control instructions are

Call & RET

Conditional branch instructions -Transfer the program to the specified label or address when certain condition is satisfied.

JNC, JC, JNZ, JZ, JP, JM, JPE, JPO

CNC, CC, CNZ, CZ, CP, CM, CPE, CPO

RNC, RC, RNZ, RZ, RP, RM, RPE, RPO

Instruction set- cont..

Machine control operations

These instructions include special instructions such as I/O data transfer, perform machine related operation

- HLT – To halt the CPU
- NOP – To perform no operation
- SIM – To set the masking of hardware interrupts and serial output data
- RIM – To read the status of interrupt mask and serial input data
- EI – Enable Interrupt
- DI – Disable Interrupt