8086 Instruction Set

Instruction set of 8086 microprocessor

The **instruction set of the 8086 microprocessor** includes a variety of instructions grouped into categories based on their function. The 8086 uses a **CISC (Complex Instruction Set Computer)** architecture, meaning it has a large number of instructions with varying lengths and execution times.

Here's a categorized breakdown of the **8086 instruction set**:

🔹 1. Data Transfer Instructions

These are used to transfer data between registers, memory, and I/O ports.

Instruction	Description
`MOV`	Transfer data from source to destination
`PUSH`	Push data onto the stack
`POP`	Pop data from the stack
`XCHG`	Exchange data between two operands
`IN`	Input from port to accumulator
`OUT`	Output from accumulator to port
`LEA`	Load effective address
`LDS`/`LES`	Load pointer using DS/ES

🔹 2. Arithmetic Instructions

Perform basic arithmetic operations.

Instruction	Description
`ADD`	Add
`ADC`	Add with carry
`SUB`	Subtract
`SBB`	Subtract with borrow
`MUL`	Unsigned multiply
`IMUL`	Signed multiply
`DIV`	Unsigned divide
`IDIV`	Signed divide
`INC`	Increment
`DEC`	Decrement
`NEG`	Two's complement (negation)
`CMP`	Compare (subtract without storing result)

3. Logical Instructions

Perform bit-level logical operations.

Instruction	Description
`AND`	Logical AND
`OR`	Logical OR
`XOR`	Logical XOR
`NOT`	Bitwise NOT (complement)
`TEST`	AND without storing result (used for setting flags)

4. Shift and Rotate Instructions

Instruction	Description
`SHL` / `SAL`	Shift left (same effect)
`SHR`	Shift right (logical)
`SAR`	Shift right (arithmetic)
`ROL`	Rotate left
`ROR`	Rotate right
`RCL`	Rotate through carry left
`RCR`	Rotate through carry right

5. Branch (Jump) Instructions

Used for decision-making and looping.

Instruction	Description
`JMP`	Unconditional jump
`CALL`	Call procedure
`RET`	Return from procedure
`JC`, `JNC`	Jump if carry/no carry
`JZ`, `JNZ`	Jump if zero/non-zero
`JE`, `JNE`	Jump if equal/not equal
`JA`, `JAE`, `JB`, `JBE`, `JG`, etc.	Conditional jumps (above, below, greater, less, etc.)
`LOOP`, `LOOPE`, `LOOPNE`	Loop control instructions

6. String Instructions

Used for operations on strings.

Instruction	Description
`MOVS` / `MOVSB` / `MOVSW`	Move byte/word from DS:SI to ES:DI
`CMPS` / `CMPSB` / `CMPSW`	Compare strings

Instruction	Description
`SCAS` / `SCASB` / `SCASW`	Scan string
`LODS` / `LODSB` / `LODSW`	Load string
`STOS` / `STOSB` / `STOSW`	Store string

These can be used with `REP`, `REPE`, or `REPNE` prefixes for repeating operations.

🔹 7. Flag Manipulation Instructions

Instruction	Description
`STC`	Set carry flag
`CLC`	Clear carry flag
`CMC`	Complement carry flag
`STD`	Set direction flag
`CLD`	Clear direction flag
`STI`	Set interrupt flag
`CLI`	Clear interrupt flag

🔹 8. Control Instructions

Instruction	Description
`HLT`	Halt the CPU
`WAIT`	Wait for the TEST pin
`NOP`	No operation
`ESC`	Escape to external device (like a coprocessor)
`LOCK`	Lock bus for exclusive access

📝 Notes:

- **Operands** can be registers, immediate values, or memory locations.
- The **segment:offset** model is used to access memory.
- Many instructions affect **flags** (CF, ZF, SF, OF, etc.).

If you'd like, I can give you a **PDF cheat sheet**, a table for MCQ purposes, or a quiz on these instructions.

write work tabular form: 6 Status flags:

Carry flag(CF)
Parity flag(PF)
Auxiliary carry flag(AF)
Zero flag(Z)
Sign flag(S)

Overflow flag (O)
Status flags are updated after every arithmetic and logic operation.

3 Control flags:

Trap flag(TF)
Interrupt flag(IF)
Direction flag(DF)

Here’s a **tabular format** summarizing the **6 status flags** and **3 control flags** of the **8086 microprocessor**, including their names, symbols, and purposes:

🔹 Status Flags (Updated after arithmetic/logic operations)

Flag Name	Symbol	Purpose
Carry Flag	CF	Set if there is a carry out from the most significant bit (MSB) in addition or borrow in subtraction
Parity Flag	PF	Set if the number of 1-bits in the result is even
Auxiliary Carry Flag	AF	Set if there is a carry/borrow from the lower nibble (4 bits)
Zero Flag	ZF	Set if the result of the operation is zero
Sign Flag	SF	Set if the result is negative (i.e., MSB is 1)
Overflow Flag	OF	Set if the result of a signed operation is too large to fit in the destination operand

---

🔹 Control Flags (Used to control CPU operations)

Flag Name	Symbol	Purpose
Trap Flag	TF	Enables single-step execution for debugging (generates an interrupt after each instruction)
Interrupt Flag	IF	Enables or disables external interrupts
Direction Flag	DF	Determines direction for string operations: 0 = Auto-increment, 1 = Auto-decrement

---

Let me know if you'd like a visual diagram or MCQs based on these flags.

Do you like this personality?

Printed using ChatGPT to PDF, powered by PDFCrowd HTML to PDF API.