

Process Control Instructions in 8086:

The Process Control Instructions in 8086 are:

- ❖ **STC:** **ST** means **Set** (make it 1), **C** means Carry flag.
- ❖ **CLC:** **CL** means **Clear** (make it 0), **C** means Carry flag.
- ❖ **CMC:** **CM** means **Complement** (Inverse the value).
- ❖ **STD:** **Set Direction** flag.
- ❖ **CLD:** **Clear Direction** flag.
- ❖ **STI:** **Set Interrupt** flag.
- ❖ **CLI:** **Clear Interrupt** flag.

Below are the explanation of each type of these Instructions:

❖ **STC Instruction:**

This instruction sets the carry flag, STC does not affect any other flag.

❖ **CLC Instruction:**

This instruction resets the carry flag to zero. CLC does not affect any other flag.

❖ **CMC Instruction:**

This instruction complements the carry flag. CMC does not affect any other flag.

❖ **STD Instruction:**

This instruction is used to set the direction flag to one so that SI and/or DI can be decremented automatically after execution of string instructions. STD does not affect any other flags.

❖ **CLD Instruction:**

This instruction is used to reset the direction flag to zero, so that SI and/or DI can be incremented automatically after execution of string instructions. CLD does not affect any other flag.

❖ **STI Instruction:**

This Process Control Instructions in 8086 sets the interrupt flag to one. This enables INTR interrupt of the 8086. STI does not affect any other flag.

❖ **CLI Instruction:**

This instruction resets the interrupt flag to zero. Due to this Process Control Instructions in 8086 will not respond to an interrupt signal on its INTR input. CLI does not affect any other flag.

8086 External Hardware Synchronization Instructions:

The 8086 External Hardware Synchronization Instructions are:

- ❖ **HLT**
- ❖ **WAIT**
- ❖ **ESC**
- ❖ **LOCK**
- ❖ **NOP**

❖ **HLT Instruction:**

The HLT instruction will cause the 8086 to stop fetching and executing instructions. The 8086 will enter a halt state. The only ways to get the processor out of the halt state are with an interrupt

signal on the INTR pin, an interrupt signal on the NMI pin, or a reset signal on the RESET input.

❖ **WAIT Instruction:**

When this instruction executes, the 8086 enters an idle condition where it is doing no processing. The 8086 will stay in this idle state until a signal is asserted on the 8086 TEST input pin, or until a valid interrupt signal is received on the INTR or the NMI interrupt input pins. If a valid interrupt occurs while the 8086 is in this idle state, the 8086 will return to the idle state after the execution of interrupt service procedure. WAIT affects, no flags. The WAIT, instruction is used to synchronize the 8086 with 8086 External Hardware Synchronization Instructions such as the 8087 math coprocessor.

❖ **ESC Instruction:**

This instruction is used to pass instructions to a coprocessor such as the 8087 math coprocessor which shares the address and data bus with an 8086. Instructions for the coprocessor are represented by a 6-bit code embedded in the escape instruction. When the 8086 fetches an ESC instruction, the coprocessor decodes the instruction and carries out the action specified by the 6-bit code specified in the instruction. In most cases the 8086 treats the ESC instruction as a NOP. In some cases the 8086 will access a data item in memory for the coprocessor.

❖ **LOCK Instruction:**

In a multiprocessor system each microprocessor has its own local buses and memory. The individual microprocessors are connected together by a system bus so that each can access system resources such as disk drives, or, memory. Each microprocessor only takes control of the system bus when it needs to access some system resources. The LOCK prefix allows a microprocessor to make sure that another processor does not take control of the system bus while it is in the middle of a critical instruction which uses the system bus. The LOCK prefix is put in front of the critical instruction. When an instruction with a LOCK prefix executes, the 8086 will assert its bus lock signal output. This signal is connected to an 8086 External Hardware Synchronization Instructions bus controller device which then prevents any other processor from taking over the system bus. LOCK affects no flags.

❖ **NOP Instruction:**

At the time of execution of NOP instruction, no operation is performed except fetch and decode. It takes three clock cycles to execute the instruction. NOP instruction does not affect any flag. This instruction is used to fill in time delays or to delete and insert instructions in the program while trouble shooting.

