Processor Control Instructions (Machine Control Instructions)

Processor control instructions are the instruc-tions, which controls the operation of the processor. 8086 processor supports variety of processor control instructions

1) CLC : Clear Carry
2) STC : Set Carry
3) CMC : Complement carry
4) CLD : Clear Direction flag

5) STD: Set Direction flag

6) CLI: Clear Interrupt flag

7) STI: Set Interrupt flag

8) HLT: Halt

9) NOP: NO Operation

10) WAIT: Wait

11) LOCK: Lock resources.

Rajeshwari. B.S
Assistant Professor
CSE Dept, BMSCE
Bangalore

1) CLC: Clear carry

CLC instruction clears the carry flag to 0.

Syntax: CLC

Function: CF ← 0

2) STC: Set Carry

STC instruction sets the carry flag to 1

Syntax: STC

Function: CF ← 1

3) CMC: Complement Carry

CMC instruction complement the carry flag content ie, if CF=0 before execution of CMC, after

execution CF=1. If CF=1 before execution of CMC after execution CF=0.

Syntax: CMC

Function: If (CF=0) then

CF ← 1

If (CF=1) then

CF ← 0

4) CLD: Clear Direction flag

CLD instruction clears the Direction flag to 0. If DF=0, then SI & DI register are automatically incremented after the execution string instructions to point to next element of the string

Syntax: CLD

Function: DF ← 0

5) STD: Set Direction flag

STD instruction sets the direction flag to 1 If DF=1, then SI & DI register are automatically decremented to point to the previous element after the execution of string instruction.

Syntax: STD

Function: DF ← 1

6) CLI: Clear Interrupt flag

CLI instruction clears the interrupt flag to 0. If IF=0, then processor will not respond to any interrupt signal on INTR pin given by external devices. CLI instruction can be used while executing any critical part of the program.

Syntax: CLI

Function: IF ← 0

7) STI: Set Interrupt flag

STI instruction sets the interrupt flag to 1. If IF=1, then processor accepts any interrupt signal on INTR pin given by external signal. This instruction can be used in the program which allows external devices to interrupt its execution.

Syntax: STI Function: IF ← 1

8) HLT: Halt

The HLT instruction causes the processor to enter into halt state. When processor executes HLT

instruction, it stops fetching & executing of next instructions. the processor can be brought out from the halt state only after the occurence of any of the following events

1) Signal on INTR pin, if IF=1
2) Signal on NMI pin
3) Reset signal on RESET pin

Syntax: HLT

9) NOP: No operation

The NOP instruction makes the Processor to wait for 3 clock cycles & proceeds with the execution of next instruction. This instruction is useful in implementing delay procedure within the program.

Syntax: NOP

10) WAIT:

The WAIT instruction causes the Processor to enter into wait state & continues to remain the that state until the occurence of any of the following events

1) Signal on INTR pin, if IF=1

2) Signal on NMI pin

3) Signal on TEST input pin

This instruction is used to synchronize 8086 processor with an 8087 math coprocessor. Every 8087 coprocessor instruction is prefixed with WAIT instruction. 8086 processor after execution of WAIT instruction gives next coprocessor floating point instruction to 8087 math processor & enters into wait state. 8087 processor, after execution of floating point instruction sends signal to 8086 processor on TEST pin. 8086 processor after receiving signal on TEST pin, resumes its execution & continues with the execution of next instructions.

Syntax: WAIT

Eg:

```
=
=
ADD AX, BX
MUL BL
WAIT
FADD
=
=
```

11) Lock: Lock resources

The LOCK instruction will prevent external devices from taking control of shared resources.

The Lock instruction is used as a prefix to the critical instruction. The Lock instruction ensures that, when the processor is in the middle of critical instruction execution that uses shared resources like bus is not to be taken over by other processor (external devices).

Syntax: Lock

Eg: LOCK XCHG A, AL ; Lock & then execute XCHG instruction.

ASSEMBLER DIRECTIVES

Assembly Language program is composed of two types of statements

* Instructions
* Directives

Rajeshwari. B.S
Assistant Professor
Dept of CSE
BMSCE, Bangalore

Instructions are the statements, which are translated into machine code by the assembler

Directives are the statements, which gives direction (information) to