MD ANIK ISLAM

1512132642

CSE331L H.W-4

*MOV-MOV Destination, Source

The MOV instruction copies a word or byte of data from a specified source to a specified destination. The destination can be a register or a memory location. The source can be a register, a memory location or an immediate number. The source and destination cannot both be memory locations. They must both be of the same type. MOV instruction does not affect any flag.

*LEA-LEA Register, Source

This instruction determines the offset of the variable or memory location named as the source and puts this offset in the indicated 16-bit register. LEA does not affect any flag.

*ADD-ADD Destination, Source

ADC-ADC Destination, Source

These instruction add number from some source to a number in some destination and put the result in the specified destination. The ADC also adds the status of the carry flag to the result. The source may be an immediate number, a register, or a memory location. The destination may be a register or a memory location. The source and the destination must be of the same type. If you want to add a byte to a word, you must copy the byte to a word location and fill the upper byte of the word with 0's before adding. Flags affected : AF, CF, OF, SF, Z

SUB - SUB Destination, Source

SBB - SBB Destination, Source

These instruction subtract the number in some source from the number in some destination and put the result in the destination. The SBB instruction also subtracts the contents of carry flag the destination. The destination can also be a register or a memory location. However, the source and the destination cannot both be memory location. The source and the destination cannot both be the same type. If you want to subtract a byte from a word, you must first move the byte to a word location such as a 16 bit register and fill the upper byte of the word with 0's. Flags affected: AF, CF, OF, PF, SF, ZF.

MUL - MUL Source:

This instruction multiples an unsigned byte in some source with an unsigned byte in AL register or an unigend word in some source with an unsigned word in AX register. The source can be aregister or a memory location. When a byte is multiplied by the content of AL, the result is put in AX. When a word is multiplied by the content of AX, the result in DX and AX registers. If the most significant byte.

byte of a 16-bit result or most significant word of 32-bit result is 0, CF and OF will both be 0's, AF, PF, SF and ZF are undefined after a MUL instruction.

DIV - DIV Source:

This instruction is used to divide an unsigned word by a byte or to divide an unsigned double word by a word. When a word is divided by a byte, the word must be in the AX register. The divisor can be in a register or a memory location. After the division, AL will contain the 8-bit quotient, and AH will contain the 8-bit remainder. When a double word is divided by a word, the most significant word of the double word must be in DX, and the least significant word of the double word must be in AX. After the division, AX will contain the 16-bit quotient and DX will contain the 16-bit remainder. If an attempt is made to divide by 0 or if the quotient is too large to fit in the destination. The 8086 will generate a type 0 interrupt. All flags are undefined after DIV instruction

INC-INC Destination

The INC instruction add 1 to a specified register or to a memory location. AF, OF, PF, SF and ZF are updated, but CF is not affected. This means that if an 8-bit destination containing FFH or a 16-bit destination containing FFFFH is incremented, the result will be 0's with no carry.

DEC - DEC Destination

This instruction subtracts 1 from the destination word or byte. The destination can be a register or a memory location. AF, OF, SF, PF and ZF are updated, but CF is not affected. This means that if an 8-bit destination containing 00H or a 16-bit destination containing 0000H is decremented, the result will be FFH or FFFFH with no carry.

DAA

This instruction is used to make sure the result of adding two packed BCD numbers is adjusted to be a legal BCD number. The result of the addition must be in AL for DAA to work correctly. If the lower nibble in AL after an addition is greater than 9 or AF was set by the addition, then the DAA instruction will add 6 to the lower nibble in AL. If the result in the upper nibble of AL is now greater than 9 or if the carry flag was set by the addition or correction, then DAA instruction will add 60H to AL.

AAA

Numerical data coming into a computer from a terminal is usually in ASCII code. In this code, the numbers 0 to 9 are represented by the ASCII codes 30H to 39H. The 8086 allows you to add the ASCII codes for two decimal digits without masking off 3 in the upper nibble of each. After the addition, the AAA instruction is used to make sure the result is the correct unpacked BCD.

AND - AND Destination, Source

This instruction AND each bit in a source byte or word with the same numbered bit in a destination byte or word. the result is put in the specified destination.

The content of the specified source is not changed.

OR-OR Destination, Source.

This instruction OR's each bit in a source byte or word with the same numbered bit in a destination byte or word. The result is put in the specified destination. The content of the specified source is not changed.

XOR-XOR Destination, Source

This instruction Exclusive-OR each bit in a source byte or word with the same numbered bit in a destination byte or word. The result is put in the specified destination. The content of the specified source is not changed.