COE 205 Term 011

The Stack and Stack Handling Instructions

The Stack:

The stack is a special segment in memory used to facilitate subroutine handling. The SS register contains the Stack Segment number where the stack is stored. The ".STACK" directive instructs the assembler to reserve a stack segment of a desired size. To reserve a stack segment of size 80 bytes, ".STACK 50H" is used before the ".CODE" directive. In this case, the ".STACK" directive initializes the Stack Pointer (SP) to 50H. If the ".STACK" directive is missing from a program, the assembler issues the warning: "LINK: Warning L4021: no stack segment".

The stack always starts at a high address and grows towards the beginning of the stack segment at a lower address. When a program starts, the stack is empty, and its size is zero. The microprocessor stores data on the stack as needed, and uses the SP register to point to the last item stored on the stack. The stack size dynamically changes as data is stored or retrieved from the stack.

Stack Handling Instructions:

The PUSH instruction is used to store the content of a 16-bit register, or memory location, on the stack. It first decreases the content of SP by two and then stores the data into the two bytes on the top of the stack. The high order byte of the data goes to the high addressed byte in the stack.

Instruction	Example	Meaning
PUSH	PUSH AX	[SP-1]
POP	POP NUM1	[NUM1]
PUSHF	PUSHF	[SP-1]
POPF	POPF	LSB(FR)

Note: FR = Flag Register

<u>Table 13. 1</u>: Summary of the Stack Handling Instructions

The PUSHF instruction is similar to the PUSH instruction, except that the PUSHF is used to push the contents of the flag register onto the stack. The POP and POPF instructions have a reverse action of the PUSH and PUSHF, respectively. The POP instruction retrieves a word from the stack and then increases SP by two. The POPF has the same effect, except that the word retrieved is saved to the flag register