

LAB 6: The Flags Register

Objectives:

- a) Understand the purpose of the eflags register.
- b) Understand the types of instructions that affect various bits in the eflags register.
- c) Understand the categories of flags in the register.

- 1) **Background:** In this lab you will investigate the eflags register. This register contains status information of the processor in that each bit indicates a state of the system. Some flags the status of last operation, some are control flags and some are system flags. eflags is 32 bits wide. In Frances-A we will only consider the first 12 bits. Since we will view the value of the eflags register as a three digit hexadecimal value. These 12 bits of the register contain all of the information that we are interested in for the application programs that we will consider in this lab. These bits are set to 1 or cleared to 0 depending on the status of the processor. The bits provide condition codes as follows:

<u>Bit#</u>	<u>Flag</u>	<u>Purpose</u>
0	Carry Flag (CF)	set if an arithmetic carry or borrow is generated in the most significant bit.
1	1	Reserved – Do not use
2	Parity Flag (PF)	set if the number of set bits in the result of the last operation is even.
3	0	Reserved – Do not use
4	Auxiliary Flag (AF)	set if an arithmetic carry or borrow is generated in the 4 least significant bits.
5	0	Reserved – Do not use
6	Zero Flag (ZF)	set if the last mathematical operation resulted in zero.
7	Sign Flag (SF)	set if the result of last mathematic operation resulted in a negative value.
8	Trap Flag (TF)	When set, the processor will execute only one instruction then call interrupt 1 (debug interrupt) to allow a debugger to inspect the program as it executes.
9	Interrupt Flag (IF)	if set interrupts will be handled, otherwise they are ignored.
10	Direction Flag (DF)	if set instructions that need to move through memory locations will decrease (move backwards) otherwise they will increase (move forward).
11	Overflow Flag (OF)	is set when the Most Significant Bit is set or cleared. This indicates an overflow has occurred in the last operation.

2) Exercises:

a) Go to the Frances-A website and compile the default program. Convert the hexadecimal value in the eflags register to binary.

- i) Which flags are set?
- ii) Why?

b) Step through the code.

- i) At which line does the value of eflags change?
- ii) What is the new value?
- iii) Why did it change?

c) Write and compile the following program.

```
int main(){
    int x=2147483648;
    x=x*2;
    x=x+1;
    x=x-1;
}
```

- i) Why is the value 2,147,483,648 significant?
- ii) What is the value of the memory location storing the value of x after the execution of step 8?
- iii) What is the value of the eflags register after the execution of step 8?
- iv) Convert the value to binary. What flags changed from the previous step?
- v) What is the value of the memory location storing the value of x after the execution of step 9?
- vi) What is the value of the eflags register after the execution of step 9?
- vii) Convert the value to binary. What flags changed from the previous step?
- viii) What is the value of the memory location storing the value of x after the execution of step 10?
- ix) What is the value of the eflags register after the execution of step 10?
- x) Convert the value to binary. What flags changed from the previous step?
- xi) What is the value of the memory location storing the value of x after the execution of step 11?
- xii) What is the value of the flags register after the execution of step 11?
- xiii) Convert the value to binary. What flags changed from the previous step?