3.4 Flags

There are many flags used for various reasons. One flag that we have already talked about is the Zero Flag (ZF).

Status Flags

Here are the flags you should know. Note that when I say a "flag is set" I mean the flag is set to 1 which is true/on. 0 is false/off.

- **Zero Flag (ZF)** - Set if the result of an operation is zero. Not set if the result of an operation is *not* zero.
- **Carry Flag (CF)** - Set if the last *unsigned* arithmetic operation carried (addition) or borrowed (subtraction) a bit beyond the register. It is also set if the operation is negative because this flag is for unsigned which is positive only. A negative could be a really big positive number.
- **Overflow Flag (OF)** - Set if a *signed* arithmetic operation is too big for the register to contain.
- **Sign Flag (SF)** - Set if the result of an operation is negative.
- **Adjust/Auxiliary Flag (AF)** - Same as the carry flag but for Binary Code Decimal (BCD) operations.
- **Parity Flag (PF)** - Set to 1 if the number of bits set in the last 8 bits is even. (10110100, PF=1; 10110101, PF=0)
- **Trap Flag (TF)** - Allows for single stepping of programs.

For a full list of flags see: https://www.tech-recipes.com/rx/1239/assembly-flags/

Examples

Basic Comparison

Here are some examples that demonstrate flags.

The most important flag in the following example is the zero flag (ZF). The code is trying to determine if RAX is equal to 4. On line 2 there is a CMP instruction which will set flags depending on the result. In this case, the ZF is going to be set to 1 because RAX is equal to 4. The JNE instruction is going to look at the flags to see if the previous operation resulted in the ZF being set.

```
mov RAX, 4
cmp RAX, 4
jne 5       ; Line 5 (ret)
call func1
ret
; ZF = 1, OF = 0, SF = 0
```

Subtraction Results Negative

The following would be for *signed* operations. The SF flag would be set to 1 because the subtraction resulted in a negative number. The same would be true if the SUB instruction was used because they both (essentially) do the same thing.

```
mov RAX, 2
cmp RAX, 8  ; 2 - 8 = -6.
; ZF = 0, OF = 0, SF = 1
```

The following is an example where the result is too big to fit into a register. Here I'm using 8-bit registers so we can work with small numbers. The biggest number that can fit in a signed 8-bit register is 128. AL is loaded with 75 then 60 is added to it. The result of adding the two together should result in 135, but the maximum number that it can hold is 128. Because of this, the number wraps around and AL is going to be -115. This sets the OF because the result was too big for the register, and the SF flag is set because the result is negative. If this was an unsigned operation CF would be set.

```
mov AL, 75
add AL, 60
; ZF = 0, OF = 1, SF = 1
```

Final Note

Hopefully, that gives you a good idea of what flags are and how they work. Remember that CMP will set flags depending on the result of the comparison. Conditional jumps will simply look at the flags. This tells us that a conditional jump does not need to be immediately proceeded with a CMP to work. Also, flags are set by things other than CMP instructions.

<- Previous Lesson
Next Lesson ->

Chapter Home