Basic Assembly

Basic Conditional branching

Objectives

- We will learn about the JZ/JNZ conditional jump instructions, and see example of their usage.

- We will briefly mention some other basic conditional jumps.

Jumping according to flags

- The JMP instruction changes the value of eip, unconditionally.
- We would like to be able to “jump” only on certain conditions.
- There is a family of instructions of the form **Jcc**, where the “cc” is replaced by some condition.
  - The jump is taken only if the condition is fulfilled.
  - The condition is usually based on the values inside the flags register.

Jump Zero (JZ)

- **JZ** label.
- Takes the jump only if the zero flag is set.
  - Only if the result of the last calculation was zero.
- Otherwise flow continues as usual.
- Examples:

```
            mov     ax,1
            dec     ax
            jz      my_label
            add     ax,5
my_label:
            add     ax,2

; The jump is taken.
; ax == 2
```

```
            mov     ax,1
            inc     ax
            jz      my_label
            add     ax,5
my_label:
            add     ax,2

; The jump is not taken.
; ax == 9
```

- The **JNZ** instruction does the opposite.
  - Jumps only if the zero flag is cleared.

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
????????	????????	?

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
⟹      mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000000	????????	?

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000000	00000003	?

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
⟹       dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000003	00000003	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
⟹       jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000003	00000002	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000003	00000002	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000003	00000002	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
⟹      dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000005	00000002	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
⟹       jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000005	00000001	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000005	00000001	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000005	00000001	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
⟹       dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000006	00000001	0

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
⟹       jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000006	00000000	1

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000006	00000000	1

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000006	00000000	1

- Calculates: 1 + 2 + 3 = 6.

Jump Zero (Example)

- Simple loop:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

eax	ecx	ZF
00000006	00000000	1

- Calculates: 1 + 2 + 3 = 6.
- How could you change the program to make it calculate 1 + 2 + 3 + … + 100 ?

Using JNZ

- We could use JNZ instead of JZ, to get simpler code:

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jz      outside
        jmp     again
outside:
        ...
```

→

```
        mov     eax,0
        mov     ecx,3
again:
        add     eax,ecx
        dec     ecx
        jnz     again

        ...
```

- Same behavior, simpler code.

Basic conditional jumps

- Some other basic conditional jumps:

Conditional jump	Description
JS / JNS	Jump if the **sign** flag is set / cleared.
JC / JNC	Jump if the **carry** flag is set / cleared.
JO / JNO	Jump if the **overflow** flag is set / cleared.

- We will get to using those later.

Summary

- The conditional jump instruction Jcc allows us to take branch decisions based on the flags register.
- We created a loop that sums 1+2+3.
- The conditional jump instructions are an indirect way of reading the flags register.

Exercises

- Code reading.
- Code writing.

- Have fun :)