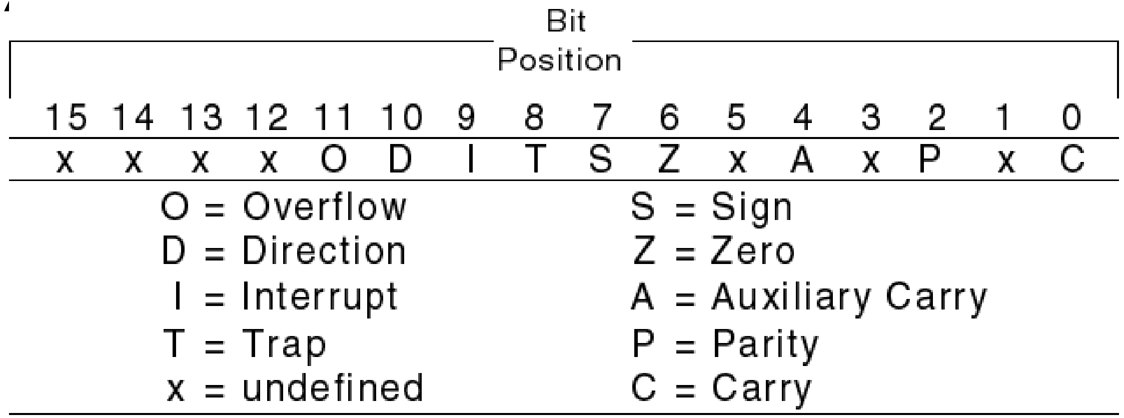
Flag Register

- **Flags Register:** A 16-Bits register specify status of CPU and information about the results of the arithmetic operations.
- Flags Register determines the current state of the processor.
- It is modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program.
- Generally you cannot access these registers directly.



Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	O	D	I	T	S	Z	x	A	x	P	x	C

O = Overflow
D = Direction
I = Interrupt
T = Trap
x = undefined

S = Sign
Z = Zero
A = Auxiliary Carry
P = Parity
C = Carry

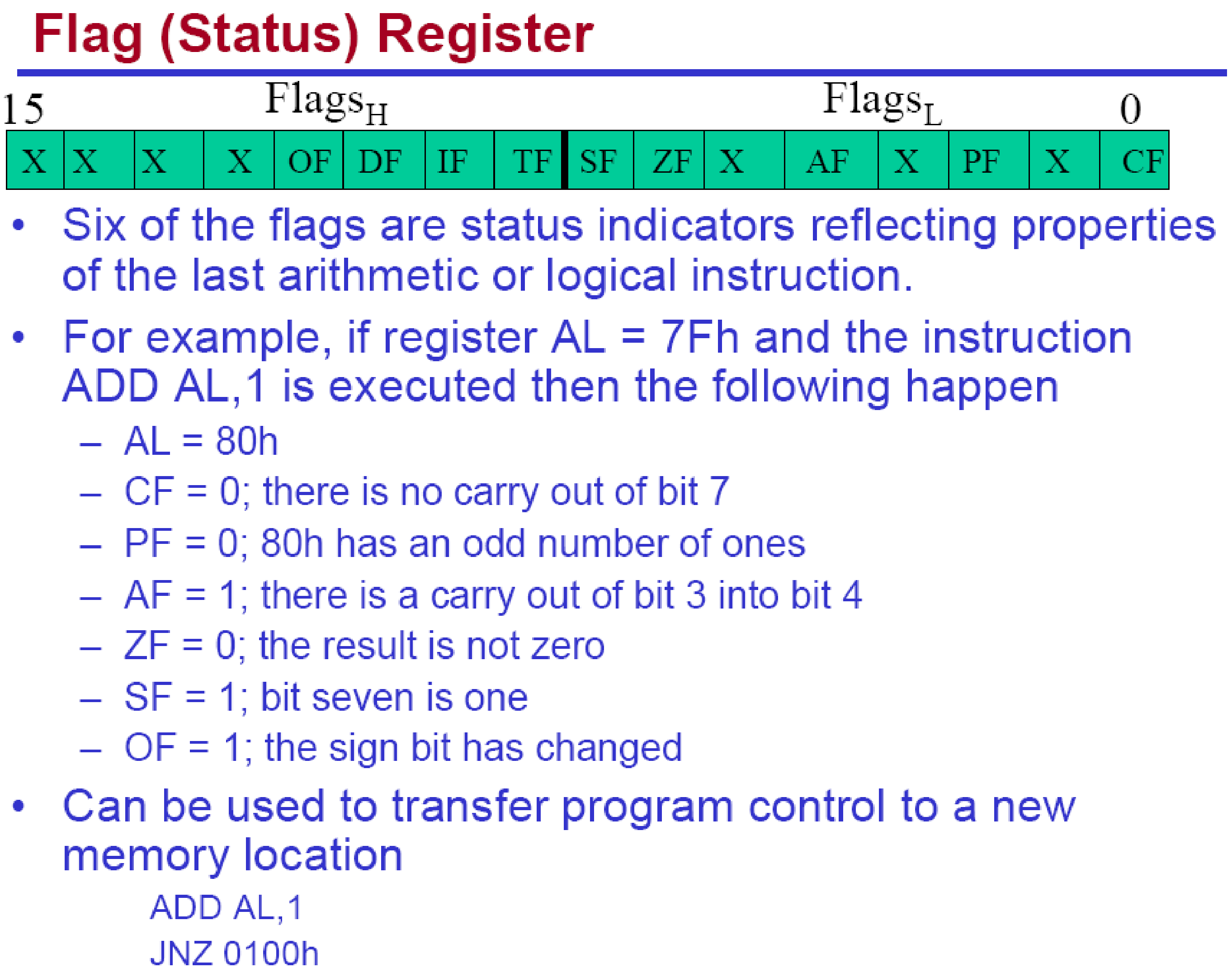
Flag Register

- **Carry Flag (CF)** - this flag is set to ’1’ when there is an unsigned overflow. For example when you add bytes 255 + 1 (result is not in range 0...255). When there is no overflow this flag is reset to 0.
- **Parity Flag (PF)** - this flag is set to ’1’ when there is even number of one bits in result, and reset to ’0’ when there is odd number of one bits.
- **Auxiliary Flag (AF)** - set to ’1’ when there is an unsigned overflow for low nibble (4 bits).
- **Zero Flag (ZF)** - set to ’1’ when result is zero. For non-zero result this flag is reset to ’0’.

Flag Register

- **Sign Flag (SF)** - set to '1' when result is negative. When result is positive it is reset to '0'. (This flag takes the value of the most significant bit).
- **Trap Flag (TF)** - Used for on-chip single-step debugging.
- **Interrupt enable Flag (IF)** - when this flag is set to '1' CPU reacts to interrupts from external devices.
- **Direction Flag (DF)** - this flag is used by some instructions to process data chains, when this flag is set to '0' - the processing is done forward, when this flag is set to '1' the processing is done backward.
- **Overflow Flag (OF)** - set to '1' when there is a signed overflow. For example, when you add bytes 100 + 50.

Flag Register (Example)



Flag (Status) Register

15 Flags_H Flags_L 0

X	X	X	X	OF	DF	IF	TF	SF	ZF	X	AF	X	PF	X	CF

- Six of the flags are status indicators reflecting properties of the last arithmetic or logical instruction.
- For example, if register AL = 7Fh and the instruction ADD AL,1 is executed then the following happen
  - AL = 80h
  - CF = 0; there is no carry out of bit 7
  - PF = 0; 80h has an odd number of ones
  - AF = 1; there is a carry out of bit 3 into bit 4
  - ZF = 0; the result is not zero
  - SF = 1; bit seven is one
  - OF = 1; the sign bit has changed
- Can be used to transfer program control to a new memory location

```
ADD AL,1
JNZ 0100h
```