|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CMP** |  |  |  |  | **Compare**  **Byte Data Comparison** |

**[Instruction format]** CMP dst, src

**[Operation]** dst – src

**[Operand]**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Mnemonic | Operand (dst, src) | | | CMP | A, #byte |  | | !addr16, #byte |  | | ES:!addr16, #byte |  | | saddr, #byte |  | | A, r | **Note** | | r, A |  | | A, !addr16 |  | | A, ES:!addr16 |  | | A, saddr |  | | |  |  | | --- | --- | | Mnemonic | Operand (dst, src) | | CMP | A, [HL] | | A, ES:[HL] | | A, [HL+byte] | | A, ES:[HL+byte] | | A, [HL+B] | | A, ES:[HL+B] | | A, [HL+C] | | A, ES:[HL+C] | |

**Note**  Except r = A

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* The source operand (src) specified by the 2nd operand is subtracted from the destination operand (dst) specified by the 1st operand.

The subtraction result is not stored anywhere and only the Z, AC and CY flags are changed.

* If the subtraction result is 0, the Z flag is set (1). In all other cases, the Z flag is cleared (0).
* If the subtraction generates a borrow out of bit 7, the CY flag is set (1). In all other cases, the CY flag is cleared (0).
* If the subtraction generates a borrow for bit 3 out of bit 4, the AC flag is set (1). In all other cases, the AC flag is cleared (0).

**[Description example]**

**CMP FFE38H, #38H;** 38H is subtracted from the contents at address FFE38H and only the flags are changed (comparison of contents at address FFE38H and the immediate data).

|  |  |  |
| --- | --- | --- |
| **BR** |  | **Branch**  **Unconditional Branch** |

|  |  |
| --- | --- |
| **[Instruction format]** | BR target |
| **[Operation]** | PC  target |

**[Operand]**

|  |  |
| --- | --- |
| Mnemonic | Operand (target) |
| BR | AX |
| $addr20 |
| $!addr20 |
| !addr16 |
| !!addr20 |

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* This is an instruction to branch unconditionally.
* The word data of the target address operand (target) is transferred to PC and branched.

**[Description example]**

**BR !!12345H;** Branch to address 12345H.

|  |  |  |
| --- | --- | --- |
| **BC** |  | **Branch if Carry**  **Conditional Branch with Carry Flag (CY = 1)** |

|  |  |
| --- | --- |
| **[Instruction format]** | BC $addr20 |
| **[Operation]** | PC  PC+2+jdisp8 if CY = 1 |

**[Operand]**

|  |  |
| --- | --- |
| Mnemonic | Operand ($addr20) |
| BC | $addr20 |

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* When CY = 1, data is branched to the address specified by the operand.

When CY = 0, no processing is carried out and the subsequent instruction is executed.

**[Description example]**

**BC $00300H;** When CY = 1, data is branched to 00300H (with the start of this instruction set in the range of

addresses 0027FH to 0037EH).

|  |  |  |
| --- | --- | --- |
| **BZ** |  | **Branch if Zero**  **Conditional Branch with Zero Flag (Z = 1)** |

**[Instruction format]**  BZ $addr20

**[Operation]** PC  PC+2+jdisp8 if Z = 1

**[Operand]**

|  |  |
| --- | --- |
| Mnemonic | Operand ($addr20) |
| BZ | $addr20 |

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* When Z = 1, data is branched to the address specified by the operand.

When Z = 0, no processing is carried out and the subsequent instruction is executed.

**[Description example]**

**DEC B**

**BZ $003C5H;** When the B register is 0, data is branched to 003C5H (with the start of this instruction set in the range of addresses 00344H to 00443H).

|  |  |  |
| --- | --- | --- |
| **BH** |  | **Branch if Higher than Conditional branch by numeric value comparison ((Z**  **CY) = 0)** |

**[Instruction format]**  BH $addr20

**[Operation]** PC  PC+3+jdisp8 if (Z  CY) = 0

**[Operand]**

|  |  |
| --- | --- |
| Mnemonic | Operand ($addr20) |
| BH | $addr20 |

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* When (Z  CY) = 0, data is branched to the address specified by the operand.

When (Z  CY) = 1, no processing is carried out and the subsequent instruction is executed.

* This instruction is used to judge which of the unsigned data values is higher. It is detected whether the first operand is higher than the second operand in the CMP instruction immediately before this instruction.

**[Description example]**

**CMP A, C**

**BH $00356H;** Branch to address 00356H when the A register contents are greater than the C register

|  |  |  |
| --- | --- | --- |
| **INC** |  | **Increment**  **Byte Data Increment** |

|  |  |
| --- | --- |
| **[Instruction format]** | INC dst |
| **[Operation]** | dst  dst + 1 |

**[Operand]**

|  |  |
| --- | --- |
| Mnemonic | Operand (dst) |
| INC | r |
| !addr16 |
| ES:!addr16 |
| saddr |
| [HL+byte] |
| ES:[HL+byte] |

**[Flag]**

|  |  |  |
| --- | --- | --- |
| Z | AC | CY |
|  |  |  |

**[Description]**

* The destination operand (dst) contents are incremented by only one.
* If the increment result is 0, the Z flag is set (1). In all other cases, the Z flag is cleared (0).
* If the increment generates a carry for bit 4 out of bit 3, the AC flag is set (1). In all other cases, the AC flag is cleared (0).
* Because this instruction is frequently used for increment of a counter for repeated operations and an indexed addressing offset register, the CY flag contents are not changed (to hold the CY flag contents in multiple-byte operation).

**[Description example]**

**INC B;** The B register is incremented.