68000 Addressing Modes

(Specify the location of an operand)

1. **Absolute Mode:**

   Effective address (EA) is in instruction.

   e. g. ADD B, D2;

   [B] + [D2] → D2, B is the address of the operand, D2 is data register

2. **Immediate Mode**

   Operand is in instruction

   e. g. ADD #50, D2;

   #50 + [D2] → D2, #50 is the operand, D2 is data register

3. **Indirect Mode**

   Effective address (EA) is in address register

   e. g. ADD (A2), D2;

   [[A2]] + [D2] → D2, A2 is address register and D2 is data register

4. **Register Mode**

   Operand is in data register

   e. g. ADD D0, D1;

   [D0] + [D1] → D1, D0 and D1 are data register

5. **Index Mode**

   a) Basic index mode

   Effective address (EA) is generated by adding a constant value to the content of an address register (the constant has 16 bits).

   e. g. ADD -$100(A4), D2;

   [[A4] - $100] + [D2] → D2

   b) Full index mode

   Effective address (EA) is generated by adding a constant value to the contents of two registers (the constant has 8 bits only).

   e. g. SUB 5(A3, D0), D1;

   EA = 5 + [A3] + [D0], [D1] - [EA] → D1

6. Autoincrement Mode

Effective address (EA) is the content of a register; after accessing, register is automatically incremented to point to next item

e. g. ADD (A1)+, D2;

[[A1]] + [D2] → D2, [A1] + 2 → A1 (note: the increment could be 1, 2 or 4 depending on the length of the operand).

7. Autodecrement Mode

The register is decremented first, and the decremented content of the register is the effective address (EA)

e. g. CLR -(A2);

[A2] - 2 → A2, 0 → [[A2]] (note: the decrement could be 1, 2 or 4 depending on the length of the operand).

8. Relative Mode

Effective address (EA) is generated by adding a constant value to the content of the Program Counter (PC)

- Same as index mode except constant added to PC, not An (address register)
- Relative to current instruction in the program
- 68000 has two relative modes
    a) Basic relative mode: ADD -$100(PC), D2;

    [[PC]-$100] +[D2] → D2

    b) Full relative mode: SUB 5(PC, D0), D1

    [D1] - [[PC] + [D0] + 5] → D1