Addressing Mode

Addressing Modes

- Specifies a rule for interpreting or modifying the address field of the instruction (before the operand is actually referenced)

Computer uses Addressing mode to accommodate one or both of the following provision :

(1) To give flexibility to programmer by providing facilities such as pointer to memory, counter for loop control, indexing of data, program relocation

(2) To reduce no. of bits in addressing fields of the instruction

Addressing Mode

Two modes that need no address field at all :

1. Implied Mode

- **Address of the operands are specified implicitly in the definition of the instruction**
- **No need to specify address in the instruction**
- **EA = AC, or EA = Stack[SP]**
- **Examples from Basic Computer - CLA, CME**

2. Immediate Mode

- **Instead of specifying the address of the operand, operand is specified in the instruction itself.**

- **No need to specify address in the instruction**
- **However, operand itself needs to be specified**
- **Sometimes, require more bits than the address**
- **Fast to acquire an operand**

Addressing Mode

3. Register Mode

- When address field specifies a processor register, it is said to be in register mode
- Designated operand need to be in a register
- Shorter address than the memory address
- Saving address field in the instruction
- Faster to acquire an operand than the memory addressing
- EA = IR(R) (IR(R): Register field of IR)

4. Register Indirect Mode

- Instruction specifies a register which contains the memory address of the operand
- Saving instruction bits since register address is shorter than the memory address
- Slower to acquire an operand than both the register addressing or memory addressing
- Adv : Fewer address bit reqd. compared to memory address
- EA = [IR(R)] ([x]: Content of x)

5. Autoincrement or Autodecrement Mode

- Similar to Register Indirect but When the address in the register is used to access memory, the value in the register is incremented or decremented by 1 automatically

Addressing Mode

6. Direct Address Mode

- Instruction specifies the memory address which can be used directly to access the operand
- Faster than the other memory addressing modes
- Too many bits are needed to specify the address for a large physical memory space
- EA = IR(addr) where (IR(addr): address field of IR)

7. Indirect Addressing Mode

- The address field of an instruction specifies the address of a memory location that contains the address of the operand
- When the abbreviated address is used large physical memory can be addressed with a relatively small number of bits
- Slow to acquire an operand because of an additional memory access
- EA = M[IR(address)]

Addressing Mode

8. Relative Addressing Modes

- The Address fields of an instruction specifies the part of the address (abbreviated address) which can be used along with a designated register to calculate the address of the operand
- Address field of the instruction is short
- Large physical memory can be accessed with a small number of address bits
- EA = f(IR(address), R), R is sometimes implied

-3 different Relative Addressing Modes depending on R;

PC Relative Addressing Mode (R = PC)

- EA = PC + IR(address)

Adv : Shorter Address Field

Indexed Addressing Mode (R = IX, where IX: Index Register)

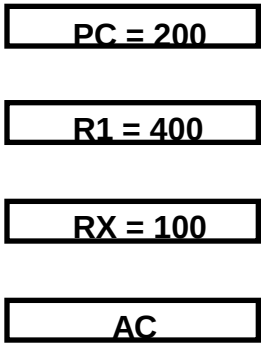
- EA = IX + IR(address)

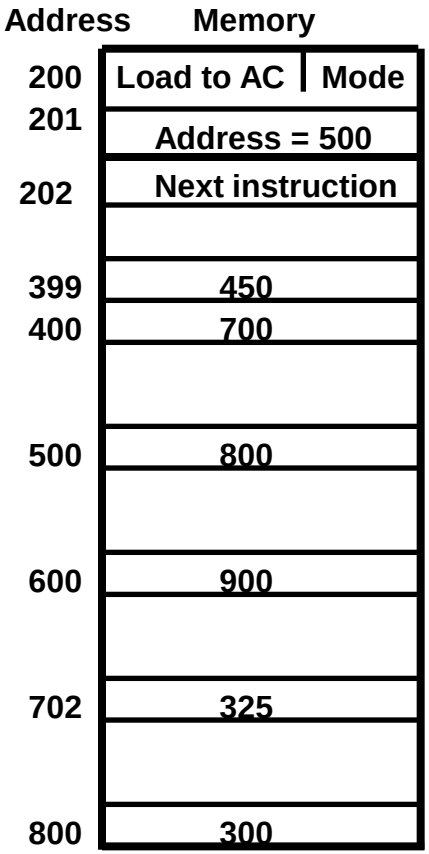
Base Register Addressing Mode

(R = BAR, where BAR: Base Address Register)

- EA = BAR + IR(address)

Addressing Mode - Example





Addressing Mode	Effective Address		Content of AC
Direct address	500	/* AC ← (500) */	800
Immediate operand	-		
Indirect address			
Relative address			
Indexed address			
Register	--		
Register indirect			
Autodecrement			
Autoincrement			

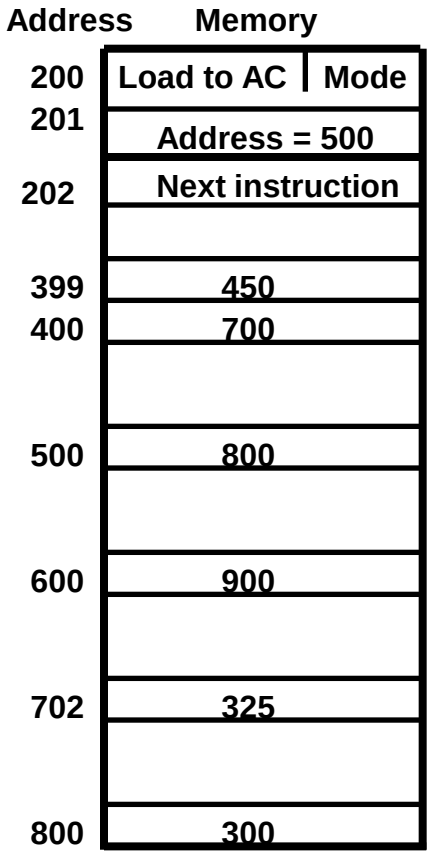
Addressing Mode - Example

PC = 200

R1 = 400

RX = 100

AC



Addressing Mode	Effective Address		Content of AC
Direct address	500	/* AC ← (500) */	800
Immediate operand	-	/* AC ← 500 */	500
Indirect address	800	/* AC ← ((500)) */	300
Relative address	702	/* AC ← (PC+500) */	325
Indexed address	600	/* AC ← (RX+500) */	900
Register	-	/* AC ← R1 */	400
Register indirect	400	/* AC ← (R1) */	700
Autodecrement	399	/* AC ← -(R) */	450
Autoincrement	400	/* AC ← (R1)+ */	700