We have examined the types of *operands* and *operations* that may be specified by *machine instructions*. Now we have to see how is the address of an operand specified, and how are the *bits* of an instruction organized to define the *operand addresses* and operation of that instruction.

Addressing Modes:

The most common addressing techniques are:

- **Immediate**
- **Direct**
- **Indirect**
- **Register**
- **Register Indirect**
- **Displacement**
- **Stack**

Addressing Mode

Two modes that need no address field at all :

1. <u>Implied Mode</u>

- **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.

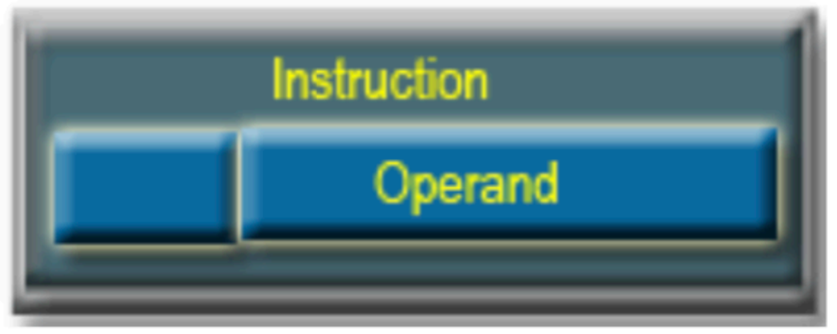
- No need to specify address in the instruction
- However, operand itself needs to be specified
- Sometimes, require more bits than the address
- Fast to acquire an operand

Immediate Addressing:

The simplest form of addressing is immediate addressing, in which the operand is actually present in the instruction:

OPERAND = A

This mode can be used to define and use constants or set initial values of variables. The advantage of immediate addressing is that no memory reference other than the instruction fetch is required to obtain the operand. The disadvantage is that the size of the number is restricted to the size of the address field, which, in most instruction sets, is small compared with the world length.



Addressing Mode

3. Register Mode

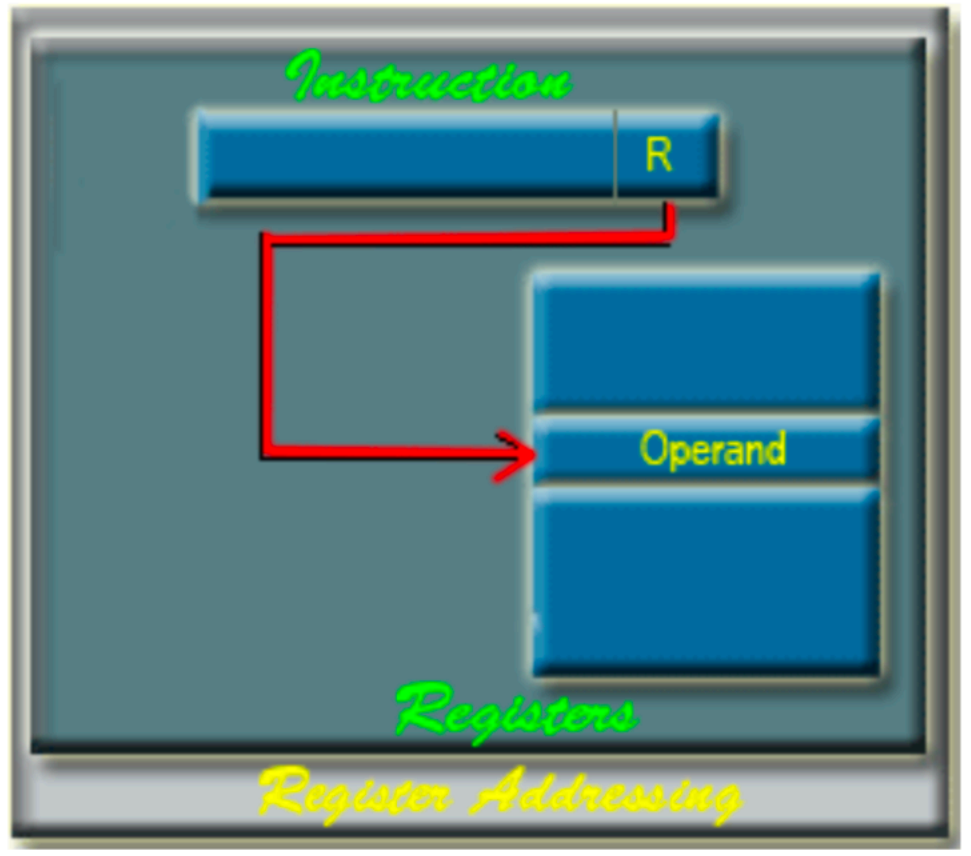
- When address field specifies a processor register, it is said to be in register mode
- Designated operand need to be in a register
- Shorter address than the memory address
- Saving address field in the instruction
- Faster to acquire an operand than the memory addressing
- EA = IR(R) (IR(R): Register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

Register Addressing:

Register addressing is similar to direct addressing. The only difference is that the address field referes to a register rather than a main memory address:

EA = R

The advantages of register addressing are that only a small address field is needed in the instruction and no memory reference is required. The disadvantage of register addressing is that the address space is very limited.



4. Register Indirect Mode

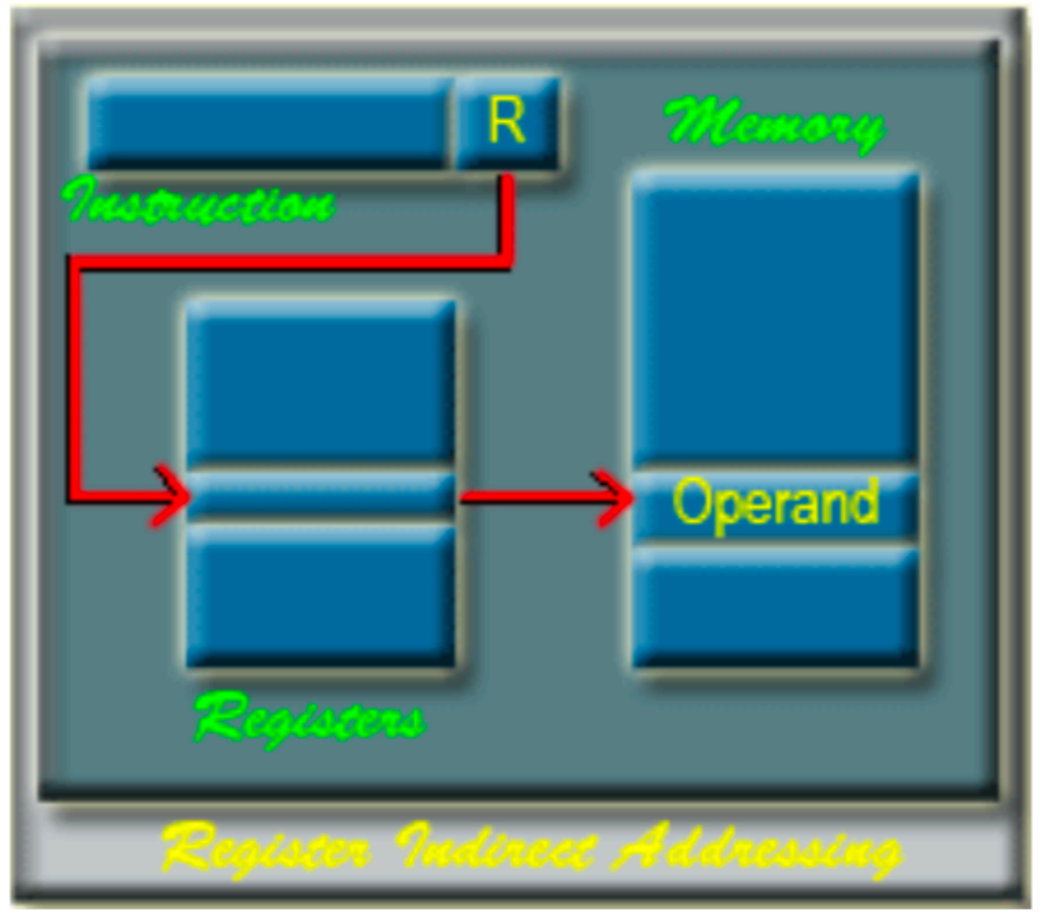
- Instruction specifies a register which contains the memory address of the operand
- Saving instruction bits since register address is shorter than the memory address
- Slower to acquire an operand than both the register addressing or memory addressing
- Adv : Fewer address bit reqd. compared to memory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

Register Indirect Addressing:

Register indirect addressing is similar to indirect addressing, except that the address field refers to a register instead of a memory location.
It requires only one memory reference and no special calculation.

EA = (R)

Register indirect addressing uses one less memory reference than indirect addressing. Because, the first information is available in a register which is nothing but a memory address. From that memory location, we use to get the data or information. In general, register access is much more faster than the memory access.



5. Autoincrement or Autodecrement Mode

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

Addressing Mode

6. Direct Address Mode

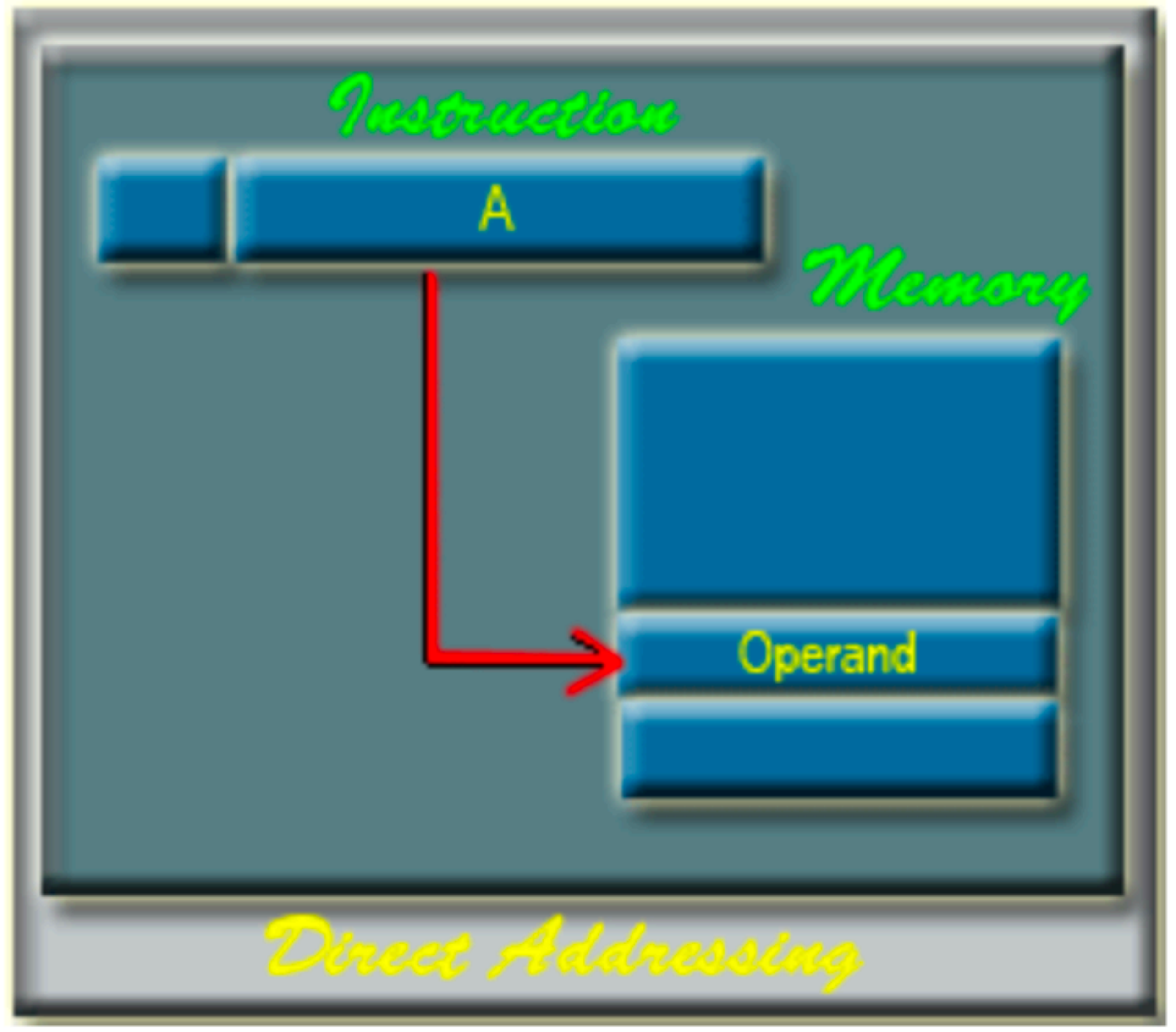
- Instruction specifies the memory address which can be used directly to access the operand
- Faster than the other memory addressing modes
- Too many bits are needed to specify the address for a large physical memory space
- EA = IR(addr) where (IR(addr): address field of IR)

Direct Addressing:

A very simple form of addressing is direct addressing, in which the address field contains the effective address of the operand:

EA = A

It requires only one memory reference and no special calculation.



7. Indirect Addressing Mode

- The address field of an instruction specifies the address of a memory location that contains the address of the operand

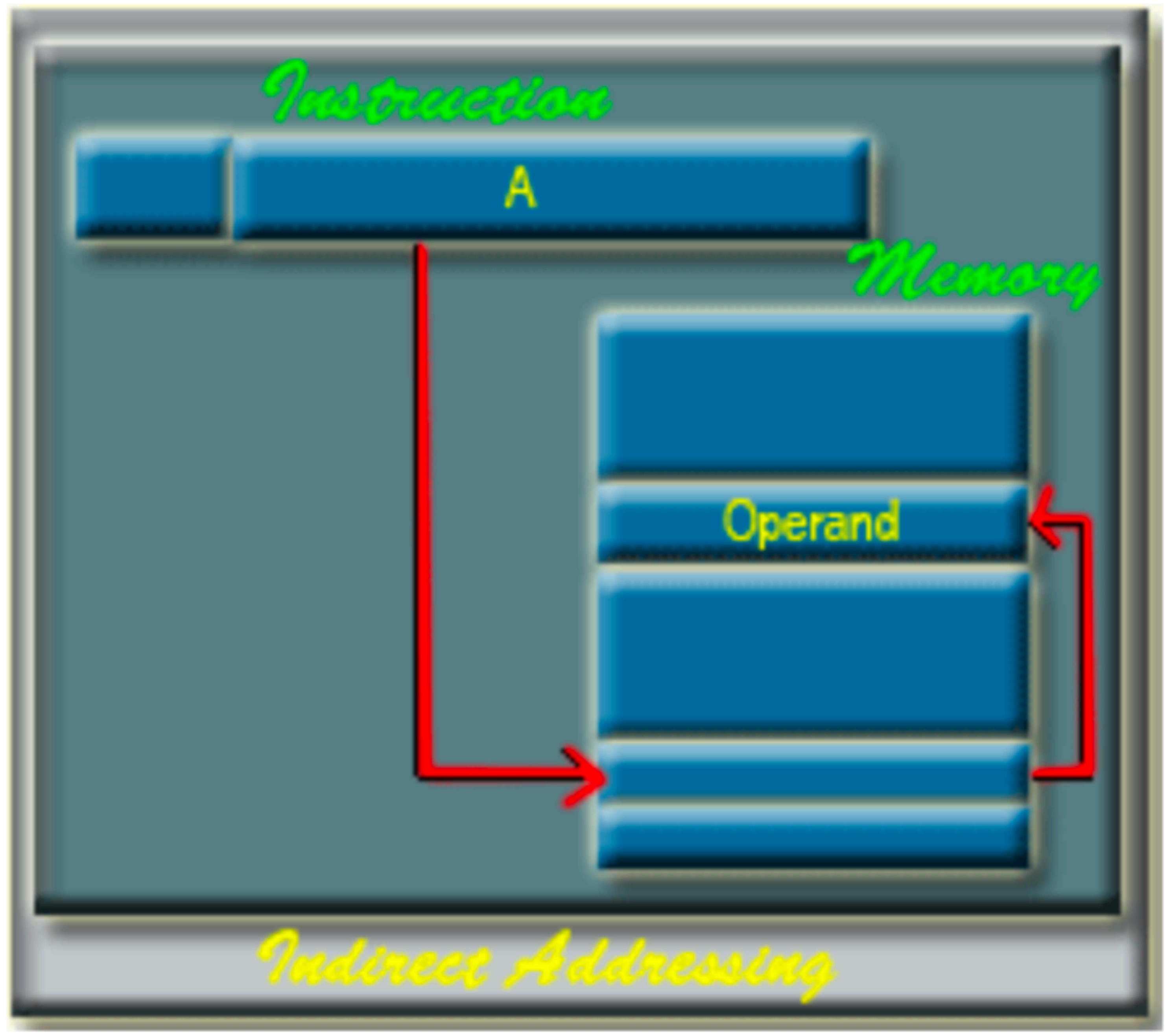
- When the abbreviated address is used large physical memory can be addressed with a relatively small number of bits

- Slow to acquire an operand because of an additional memory access
- EA = M[IR(address)]

Indirect Addressing:

With direct addressing, the length of the address field is usually less than the word length, thus limiting the address range. One solution is to have the address field refer to the address of a word in memory, which in turn contains a full-length address of the operand. This is know as indirect addressing:

EA = (A)



Diaplacement Addressing:

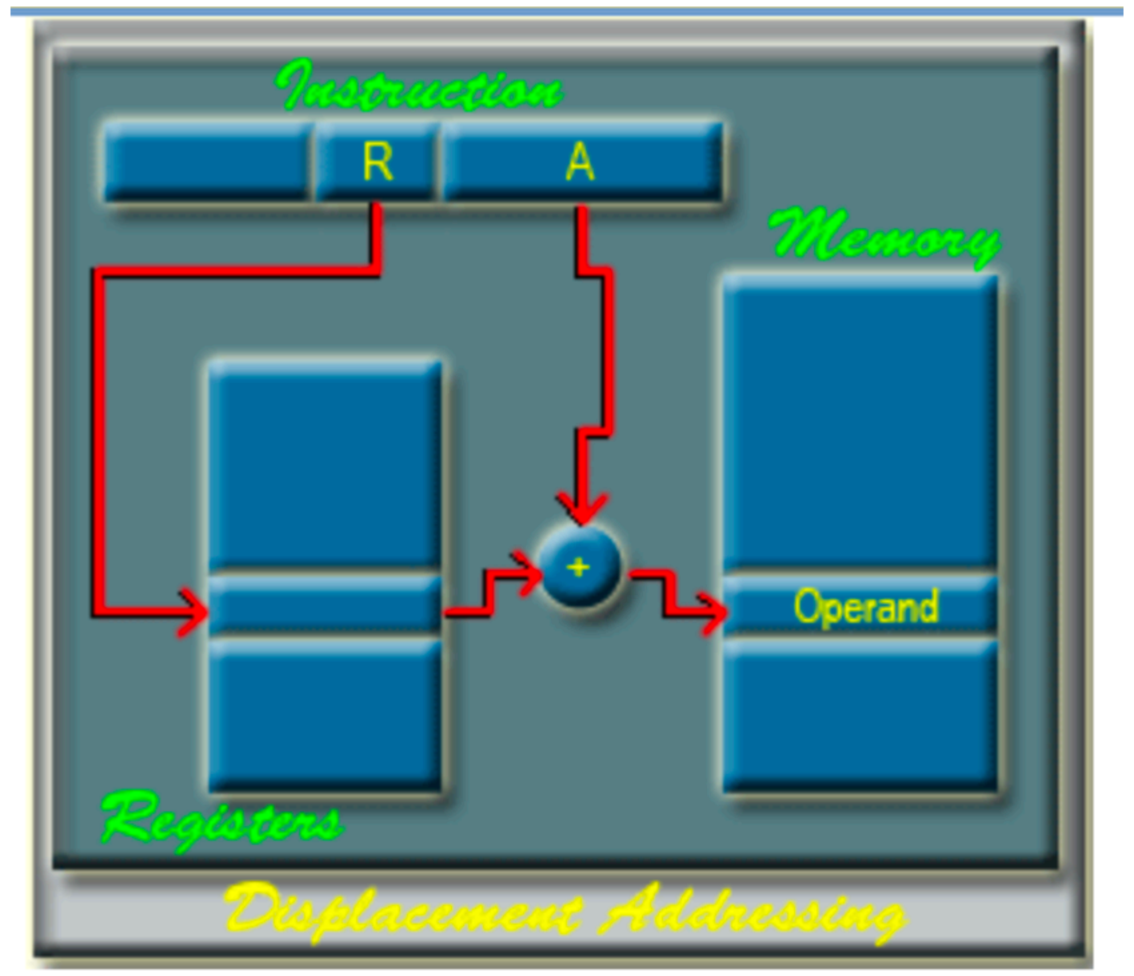
A very powerful mode of addressing combines the capabilities of direct addressing and register indirect addressing, which is broadly categorized as displacement addressing:

$$EA = A + (R)$$

Displacement addressing requires that the instruction have two address fields, at least one of which is explicit. The value contained in one address field (value = A) is used directly. The other address field, or an implicit reference based on opcode, refers to a register whose contents are added to A to produce the effective address.

Three of the most common use of displacement addressing are:

- Relative addressing
- Base-register addressing
- Indexing



Q1. Which addressing mode execute its instructions within CPU without the necessity of reference memory for operands?

a. Indirect Mode
b. Immediate Mode
c. Direct Mode
d. Register Mode

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

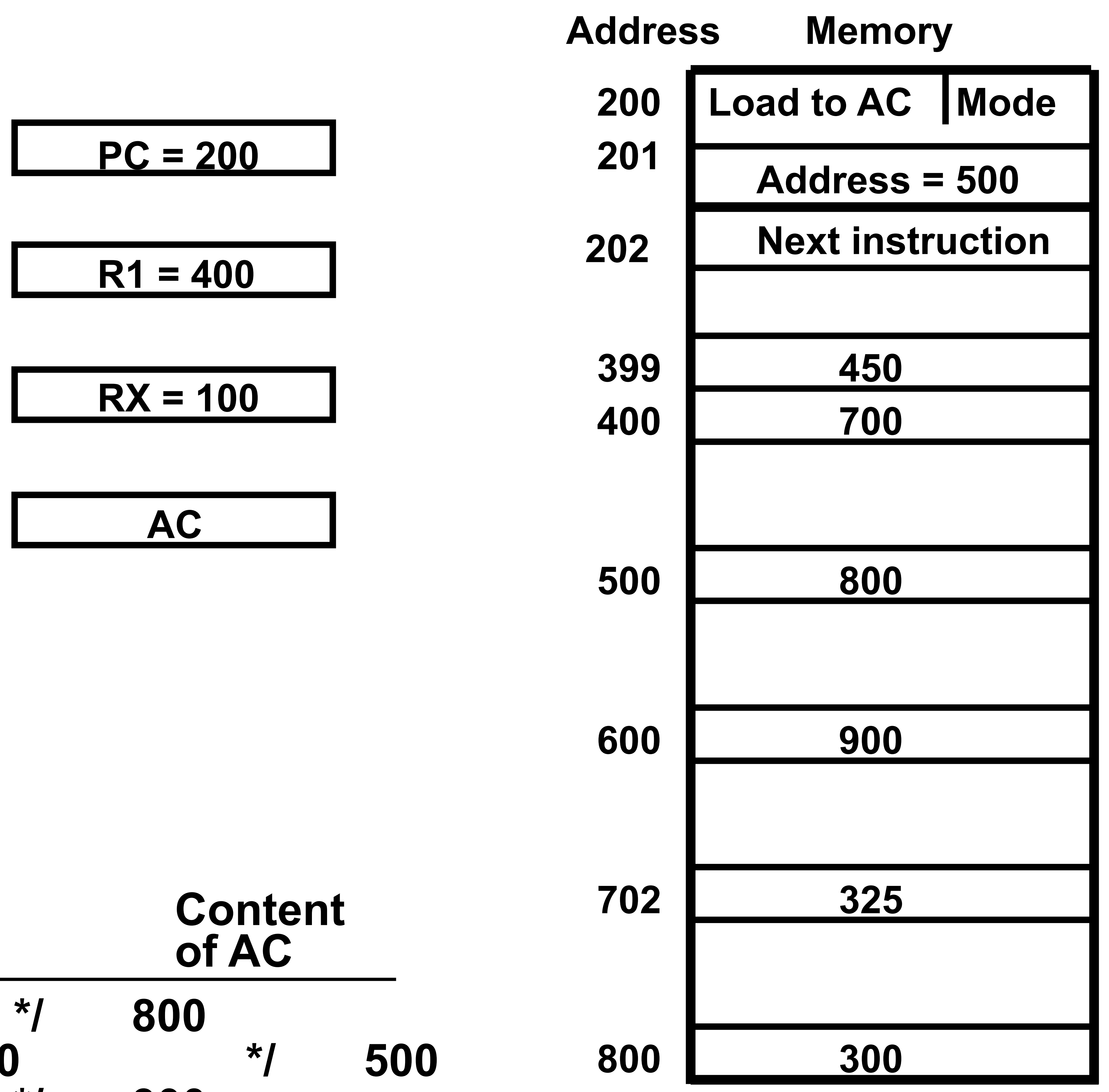
-3 different Relative Addressing Modes depending on R;

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	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

Reduced Set Instruction Set Architecture (RISC) –
The main idea behind is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating and storing operations just like a load command will load data, store command will store the data.

Complex Instruction Set Architecture (CISC) –
The main idea is that a single instruction will do all loading, evaluating and storing operations just like a multiplication command will do stuff like loading data, evaluating and storing it, hence it’s complex.

Both approaches try to increase the CPU performance

- **RISC:** Reduce the cycles per instruction at the cost of the number of instructions per program.
- **CISC:** The CISC approach attempts to minimize the number of instructions per program but at the cost of increase in number of cycles per instruction.

Characteristic of RISC –

1. Simpler instruction, hence simple instruction decoding.
2. Instruction come under size of one word.
3. Instruction take single clock cycle to get executed.
4. More number of general purpose register.
5. Simple Addressing Modes.
6. Less Data types.
7. Pipeling can be achieved.

Characteristic of CISC –

1. Complex instruction, hence complex instruction decoding.
2. Instruction are larger than one word size.
3. Instruction may take more than single clock cycle to get executed.
4. Less number of general purpose register as operation get performed in memory itself.
5. Complex Addressing Modes.
6. More Data types.

Example – Suppose we have to add two 8-bit number:

- **CISC approach:** There will be a single command or instruction for this like ADD which will perform the task.
- **RISC approach:** Here programmer will write first load command to load data in registers then it will use suitable operator and then it will store result in desired location.

RISC	CISC
Focus on software	Focus on hardware
Uses only Hardwired control unit	Uses both hardwired and micro programmed control unit
Transistors are used for more registers	Transistors are used for storing complex Instructions
Fixed sized instructions	Variable sized instructions
Can perform only Register to Register Arthmetic operations	Can perform REG to REG or REG to MEM or MEM to MEM
Requires more number of registers	Requires less number of registers
Code size is large	Code size is small
A instruction execute in single clock cycle	Instruction take more than one clock cycle
A instruction fit in one word	Instruction are larger than size of one word

Difference Between CISC and RISC

Architectural Characterstics	**Complex Instruction Set Computer(CISC)**	**Reduced Instruction Set Computer(RISC)**
Instruction size and format	Large set of instructions with variable formats (16-64 bits per instruction).	Small set of instructions with fixed format (32 bit).
Data transfer	Memory to memory.	Register to register.
CPU control	Most micro coded using control memory (ROM) but modern CISC use hardwired control.	Mostly hardwired without control memory.
Instruction type	Not register based instructions.	Register based instructions.
Memory access	More memory access.	Less memory access.
Clocks	Includes multi-clocks.	Includes single clock.
Instruction nature	Instructions are complex.	Instructions are reduced and simple.

The	CISC stands for	___________
a)	Computer Instruction Set Compliment	
b)	Complete Instruction Set Compliment	
c)	Computer Indexed Set Components	
d)	Complex Instruction set computer	