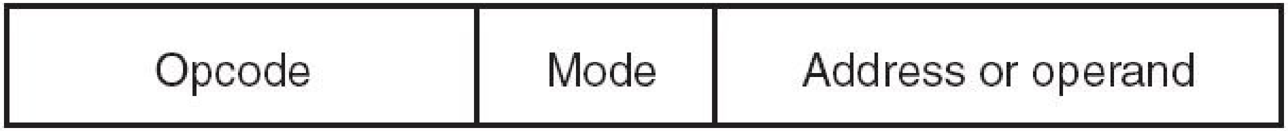
Instruction formats

What is instruction format?

- An instruction is normally made up of a combination of an operation code and some way of specifying an operand, most commonly by its location or address in memory



Opcode	Mode	Address or operand

Fig. 9-3 Instruction Format with Mode Field

Types of address instructions

Three address instructions

- Memory addresses for the two operands and one destination need to be specified.
- It is also called General register organization.
- **Instruction: ADD R1, R2, R3**

Microoperation: R1 ← R2 + R3

EVALUATE X=(A+B)*(C+D)

• ADD • ADD	R1, A, B R2, C, D	**R1 ← M[A]+M[B]** **R2 ← M[C]+M[D]**
• MUL	X, R1, R2	**M[X] ← R1*R2**

X=A+(B.C)+(D.E)

Two address instructions

- Two address registers or two memory locations are specified
- Assumes that the destination address is the same as that of the first operand.
- ***Instruction: ADD R1, R2***

  Microoperation: R1 ← R1 + R2

EVALUATE X=(A+B)*(C+D)

• MOV	R1, A	**R1 ← M[A]**
• ADD	R1, B	**R1 ← R1+M[B]**
• MOV	R2, C	**R2 ← M[C]**
• ADD	R2, D	**R2 ← R2+M[D]**
• MUL	R1,R2	**R1 ← R1*R2**
• MOV	X,R1	**M[X] ← R1**

X=A+(B.C)+(D.E)

One address instructions

- One address can be a register name or memory address.
- **SINGLE ACCUMULATOR ORGANIZATION**
- It uses AC register for all data manipulation
- **Instruction: ADD X**

Microoperation: AC ← AC + M[X]

EVALUATE X=(A+B)*(C+D)

• LOAD	A	AC ← M[A]
• ADD	B	AC ← AC+M[B]
• STORE	T	M[T] ← AC
• LOAD	C	AC ← M[C]
• ADD	D	AC ← AC+M[D]
• MUL	T	AC ← AC*M[T]
• STORE	X	M[X] ← AC

X=A+(B.C)+(D.E)

Zero address instruction

- Stack is used. Arithmetic operation pops two operands from the stack and pushes the result.
- Also called ***stack organization***

Zero address instruction

- Evaluate $X = (A + B) * (C + D)$

• PUSH	A	TOS ← A
• PUSH	B	TOS ← B
• ADD		TOS ← (A+B)
• PUSH	C	TOS ← C
• PUSH	D	TOS ← D
• ADD		TOS ← (C+D)
• MUL		TOS ← (C+D)*(A+B)
• POP	X	M[X] ← TOS

Push A
Push B
ADD
Push C
Push D
ADD
Mult
Store

Addressing Modes

Addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

Well known addressing modes-

- Implied mode
- Immediate mode
- Register mode
- Register indirect mode
- Autoincrement or Autodecrement
- **Direct address mode**
- **Indirect address mode**
- Relative address mode
- Indexed addressing mode
- Base register addressing mode

Implied Mode

- In this mode the operands are specified implicitly in the definition of the instruction.
- All register reference instructions that use an accumulator are implied mode instruction.
- Example: CMA compliment accumulator.

Immediate Addressing

- Data needed by the processor is contained in the instruction
- Operand= address field
- e.g-ADD 5

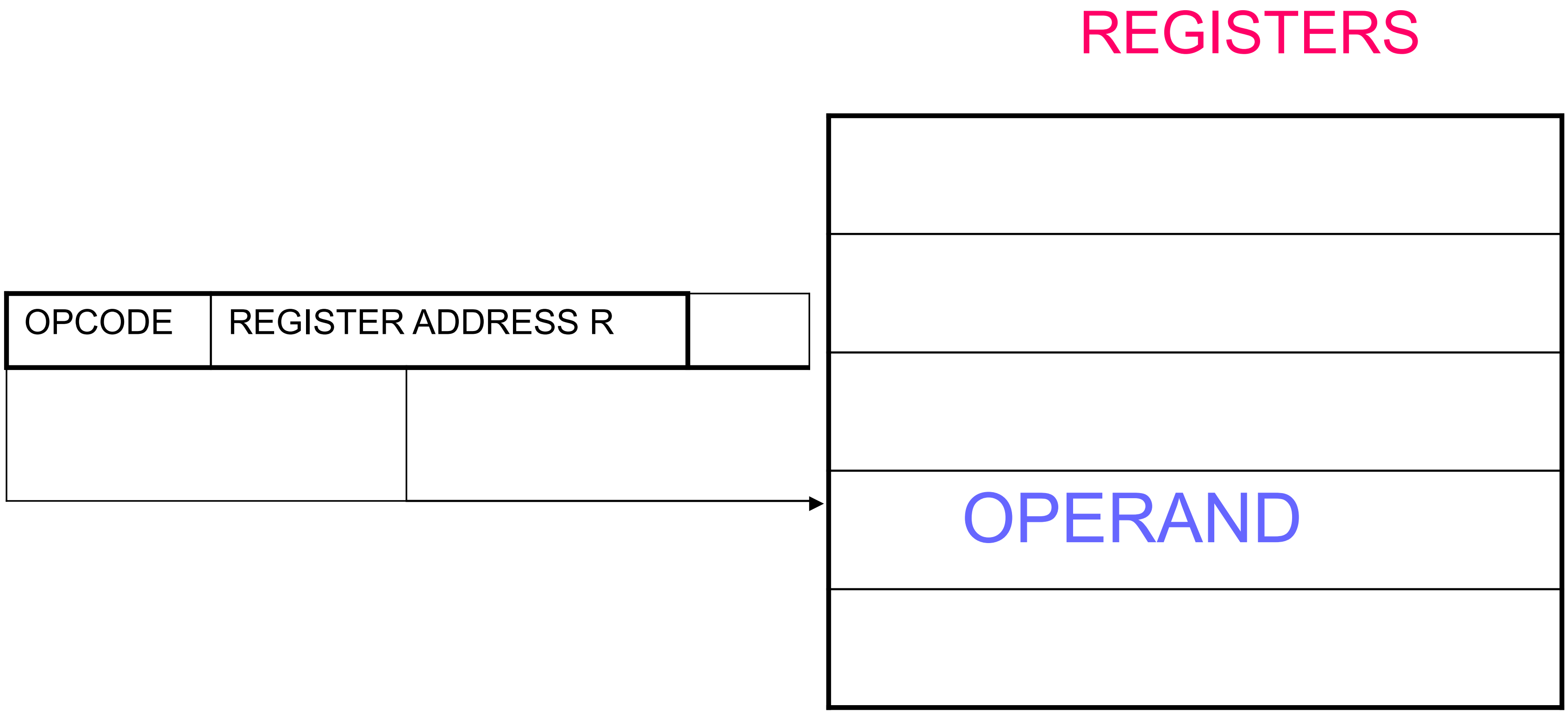
  -add 5 to the content of accumulator

  -5 is the operand

opcode	Operand (5)

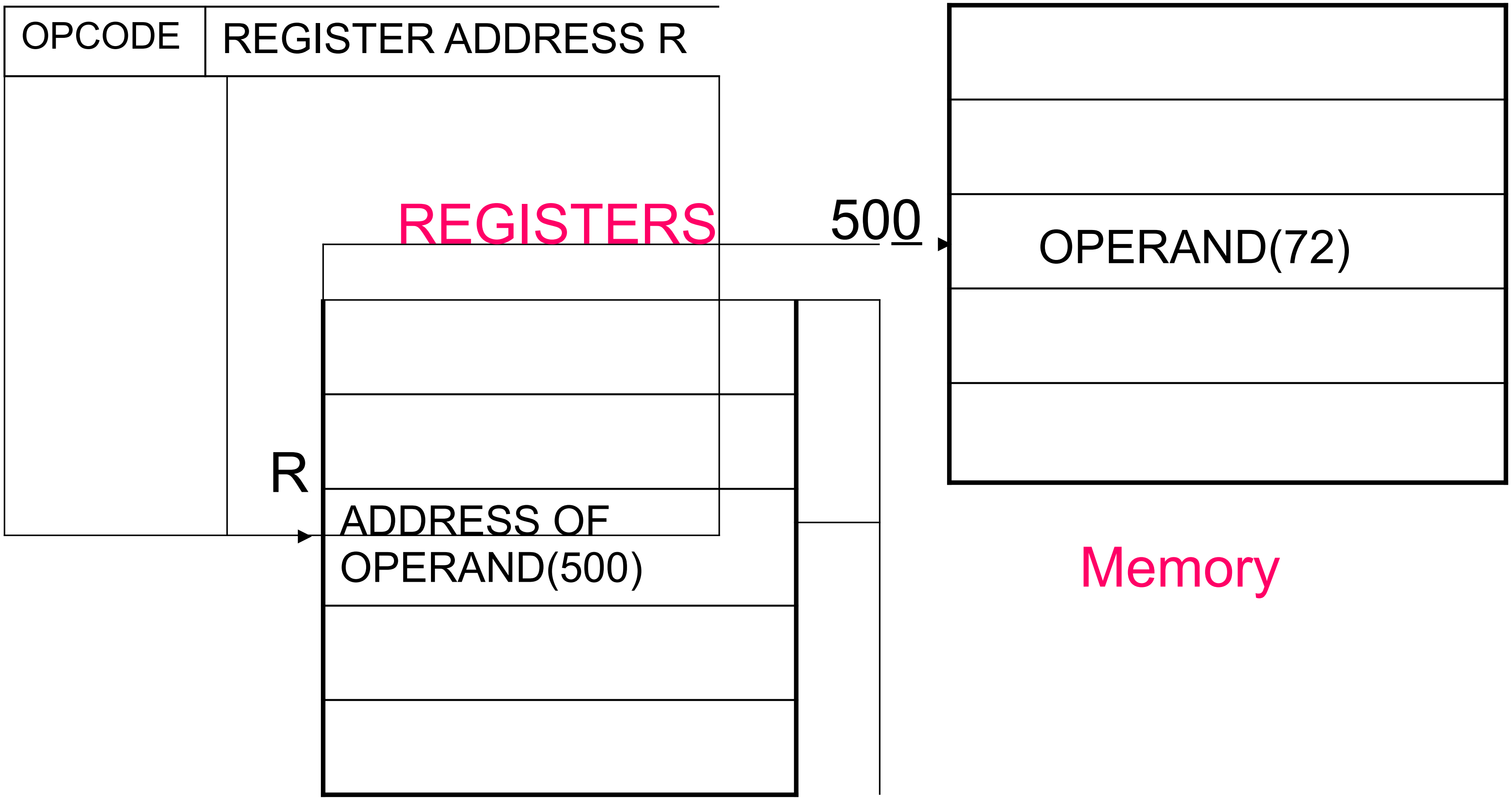
Register Addressing

- Operand is held in register named in address field
- A k-bit field can specify any one of 2^k registers



Register Indirect Addressing

- The selected register contains the address of the operand rather than the operand itself.



Auto Increment or Auto Decrement Mode

- This is similar to the register indirect mode except that the register is incremented or decremented after or before its value is used to access memory.

R1=400

1. Auto incr-E.A=400

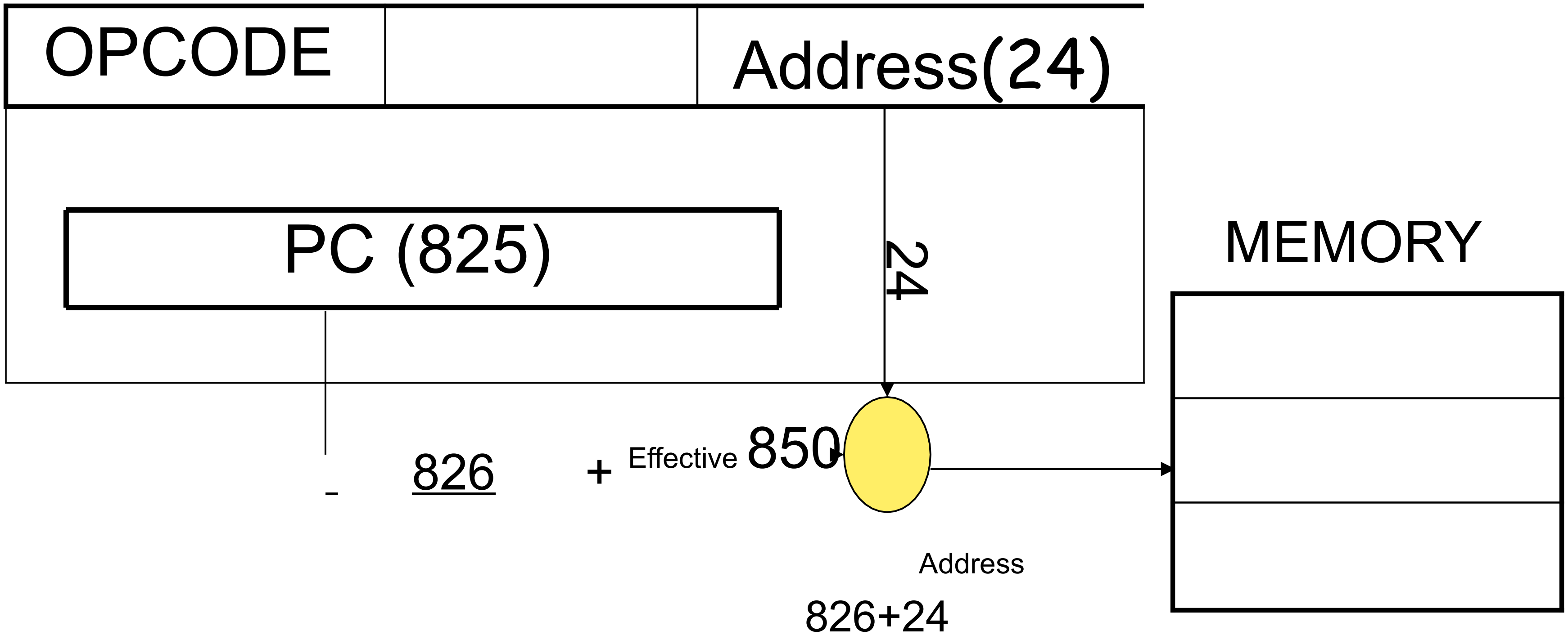
R1 is incremented to 401after the execution

2. Auto decr-R1 is decremented prior to the execution,therefore,R1=399 and E.A=399

399	450
400	700

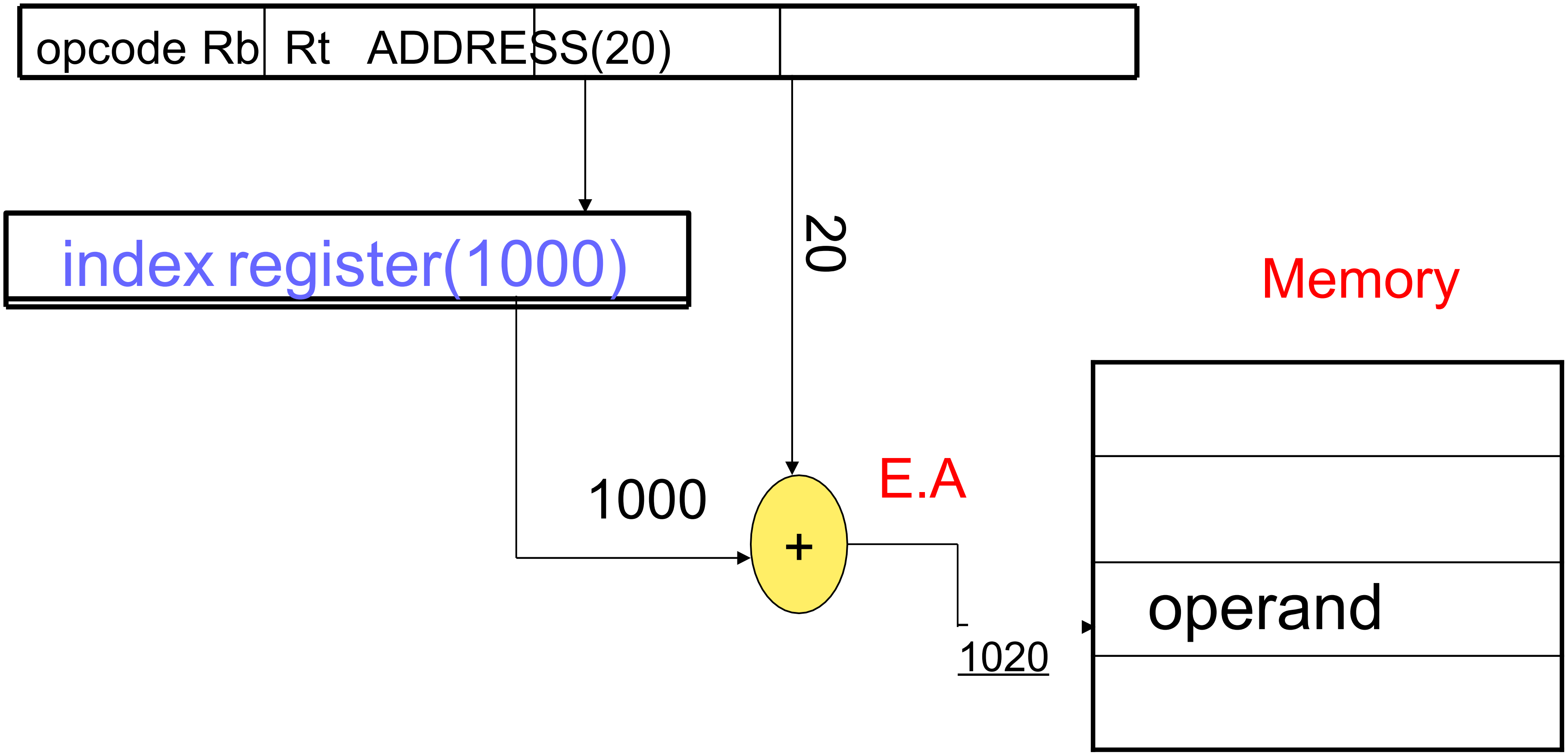
RELATIVE ADDRESS MODE

- In this mode the content of the program counter is added to the address part of the instruction to obtain the effective address.



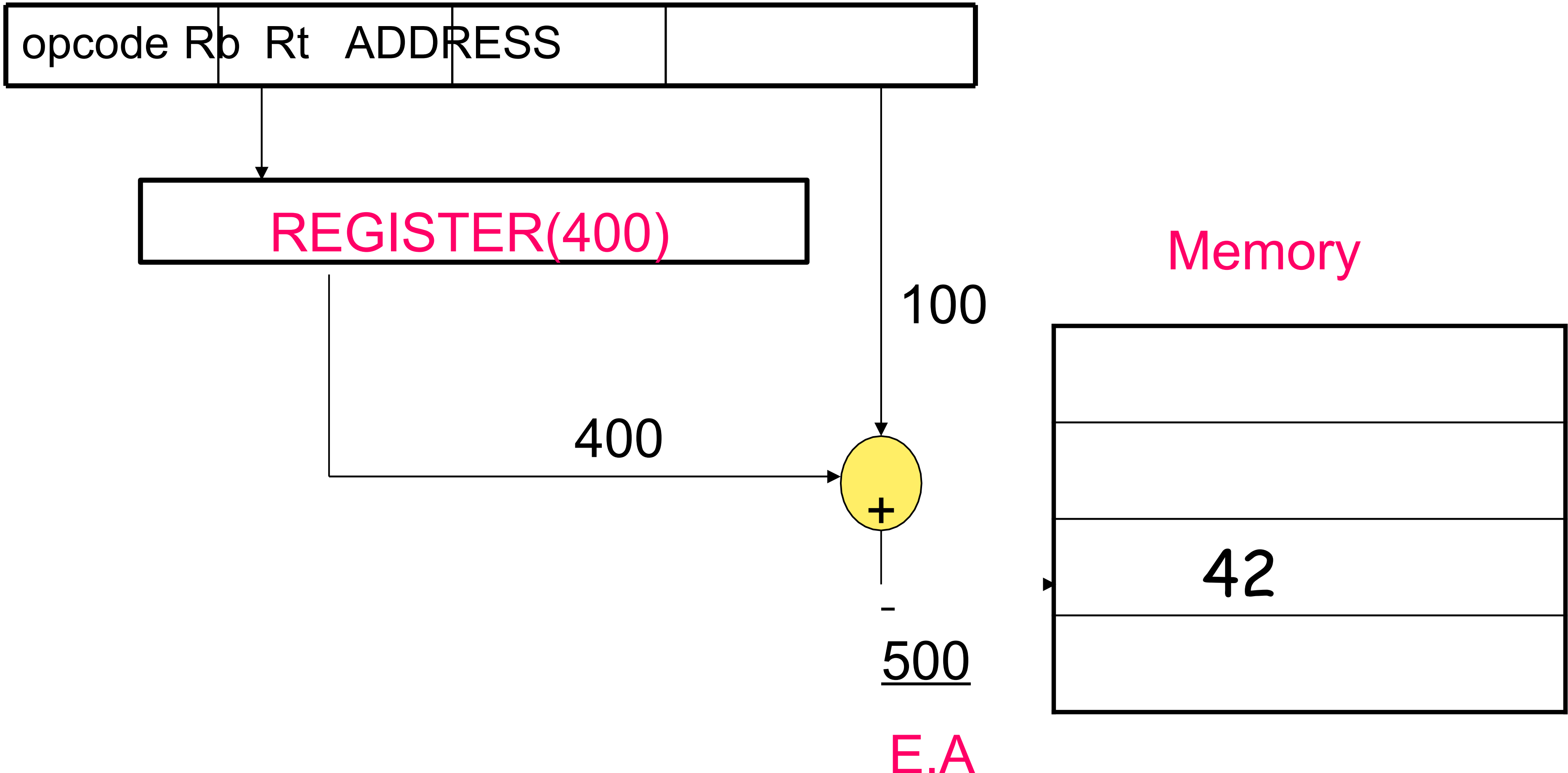
Indexed addressing mode

- In this mode the content of an index register is added to the address part of the instruction to obtain the effective address.



BASE REGISTER ADDRESSING MODE

- In this mode the content of a base register is added to the address part of the instruction to obtain the effective address.


opcode Rb Rt ADDRESS
REGISTER(400)
Memory
100
400
+
42
500
E.A