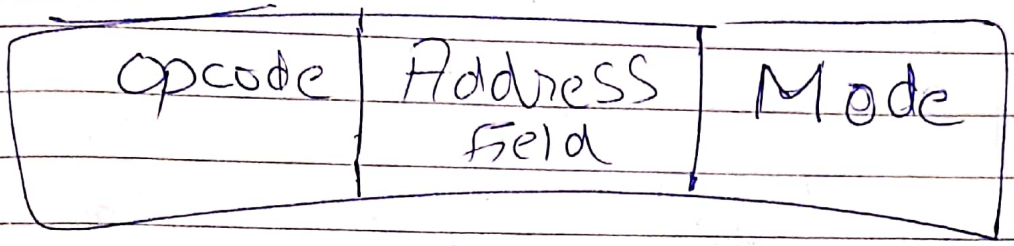
ASHOKA
Date:
Page No.

ADDRESSING MODES

Instruction =>



1. Immediate Mode

Operand Directly Given

eg. $R_1 \leftarrow 5$

2. Implied Mode

Operands present in special registers of CPU.

eg. ~~R...~~ PC

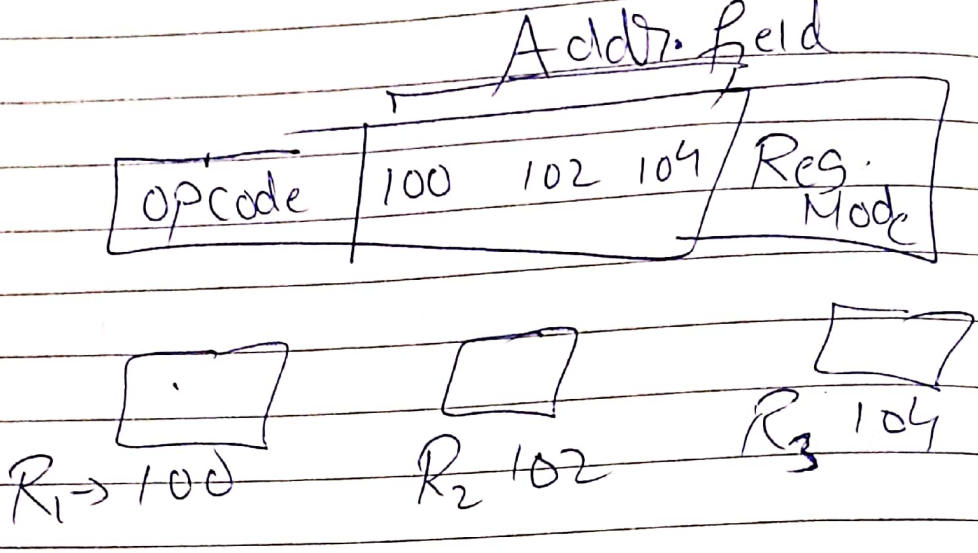
=

AC ← AC + 1

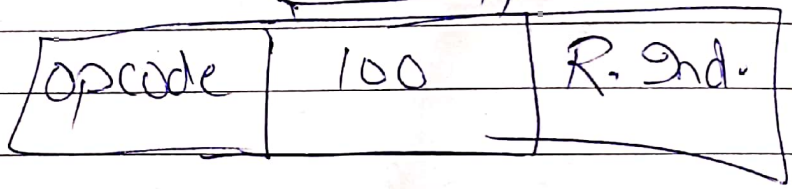
3. Register Mode

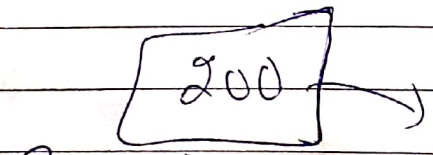
Addr. field contains address of registers.

e.g. $R_1 \leftarrow R_2 + R_3$

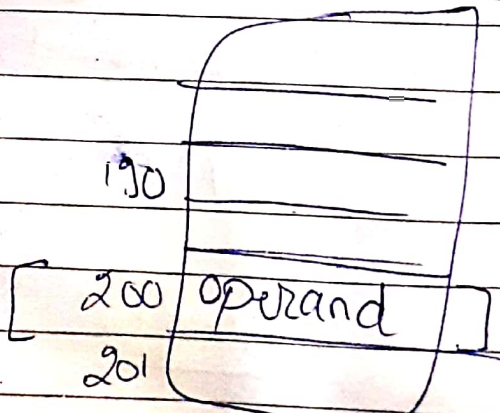


4- Register Indirect Mode





go to memory 200

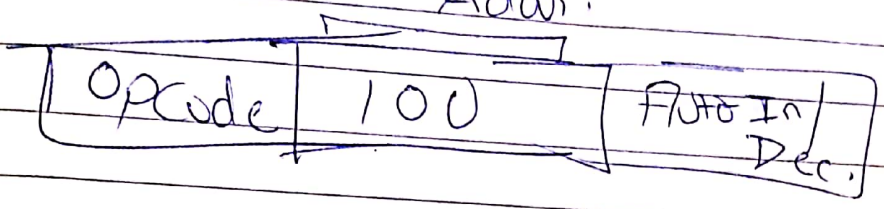


Why to use reg. indirect ?

Less no. of bits in address field.

Date:
Page No.

5. Autoincrement / Autodecrement mode

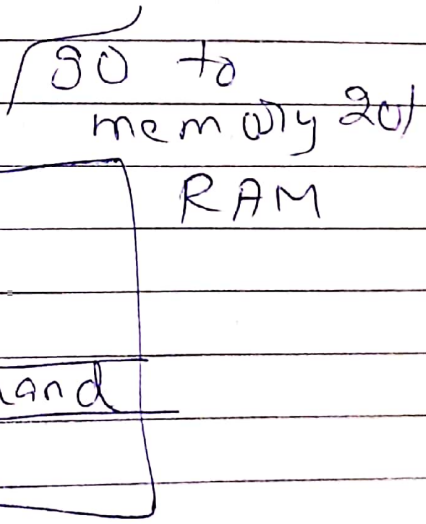


Suppose INCrement

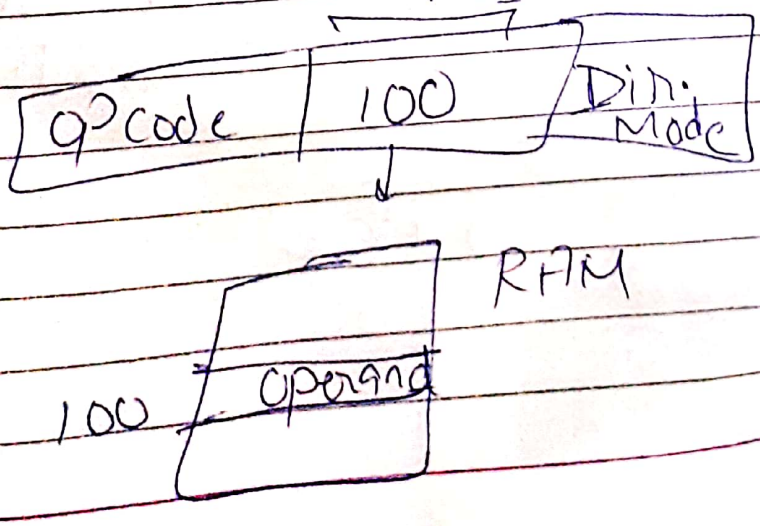
~~100 → 101~~

200 → Inc. 200 → 201

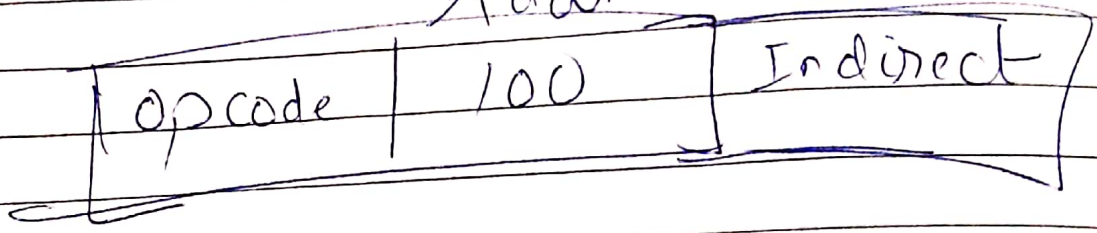
R → 100

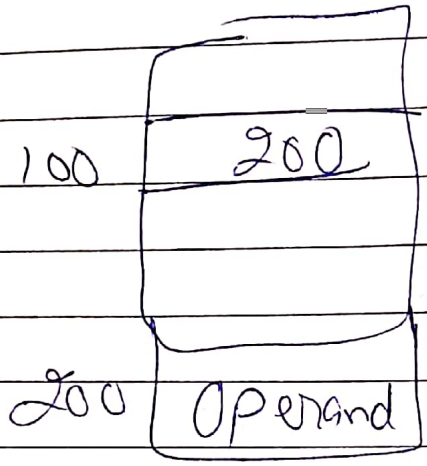


6. Direct Mode



7. Indirect Mode





Two memory access

8. Relative Mode

Operand Effective Address = PC + Addr. field.

Eg. PC [825] → Fetch instruction at address 825

PC++ → PC = [826]

ASHOKA
Date:
Page No.

825 Instruction => | Opcode | 100 | Ref. |
Addr.

Operand = 826 + 100 = 926 memory location.

9 Indexed Mode

Operand Eff. Addr. = Index Register + Addr. field.

| Opcode | 100 | Index |
Addr.

Index 20

Operand at 120 memory location.

Beneficial in arrays

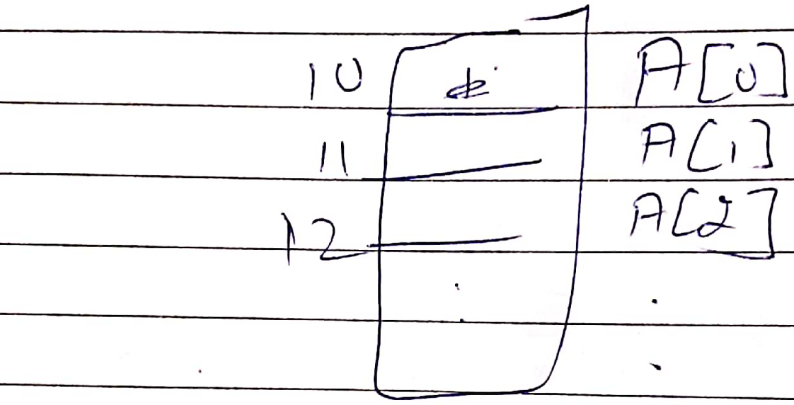
| opcode | 10 | Index |

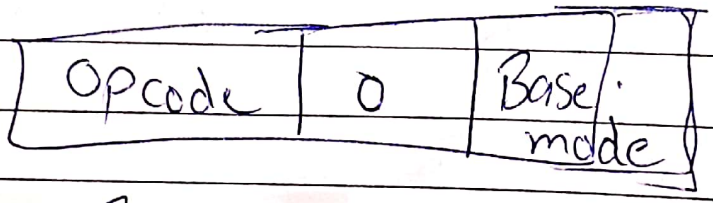
Index 0

10	A[0]
11	A[1]
12	A[2]
.	.
.	.

10. Base Register Mode

Operand's Addr = Base Reg. + Addr. field





Base Reg | 10

Beneficial ?

In Program relocation.

Load to AC	500	Mode

ASHOKA
Date:
Page No.

Q

200	Load to AC	Mode
201	Addr = 500	

Single instruction

399	450
400	700
500	800
600	900
702	325
800	300

Index 100

Determine the content of AC if Mode is

1. Direct Mode
   AC ← 800
2. Immediate Mode
   AC ← 500
3. Indirect Mode
   AC ← 300

4. Relative Mode

operand = PC + Addr.
Addr
= 202 + 500
Address = 702

AC ← 325

5. Index Mode

Operand Addr = 100 + 500
= 600

AC ← 900