

Computer Science & IT Engineering

Addressing Mode- Part II





Topics

to be covered



- 1 Addressing Modes & Types
- 2 GATE PYQ'S & Practice Questions



Recap of previous Lecture



- 1 Addressing Modes
- 2 Addressing Modes Types

Addressing Modes

- ① Immediate AM
- ② Direct | Absolute AM
- ③ Mem Indirect AM
- ④ Register Direct AM
- ⑤ Register Indirect AM
- ⑥ PC Relative AM
- ⑦ Index Reg AM
- ⑧ Base Reg AM
- ⑨ Implied | Implicit AM
- ⑩ Auto Decrement AM
- ⑪ Auto Increment AM.

Addressing Modes

① Immediate AM : In this AM operand are present in the Instruction Itself.



↓
'OPERAND'
(DATA)

Immediate AM is used to Access the Constant & Initialize the Register & Variable With Value.



- Immediate AM Can not be used as Destination Address

- n bit Unsigned = 0 to $2^n - 1$

Signed : -2^{n-1} to $+ (2^n - 1)$

Immediate AM

(e1)

MVI R₁ 6000

(e2)

MOV R₁ #6000

R₁ ∈ 6000

(e2)

ADD R₂ #500

(e3)

ADDL R₂ 500

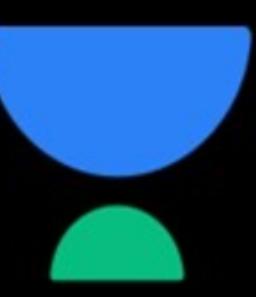
R₂ ∈ R₂ + 500

500

6000

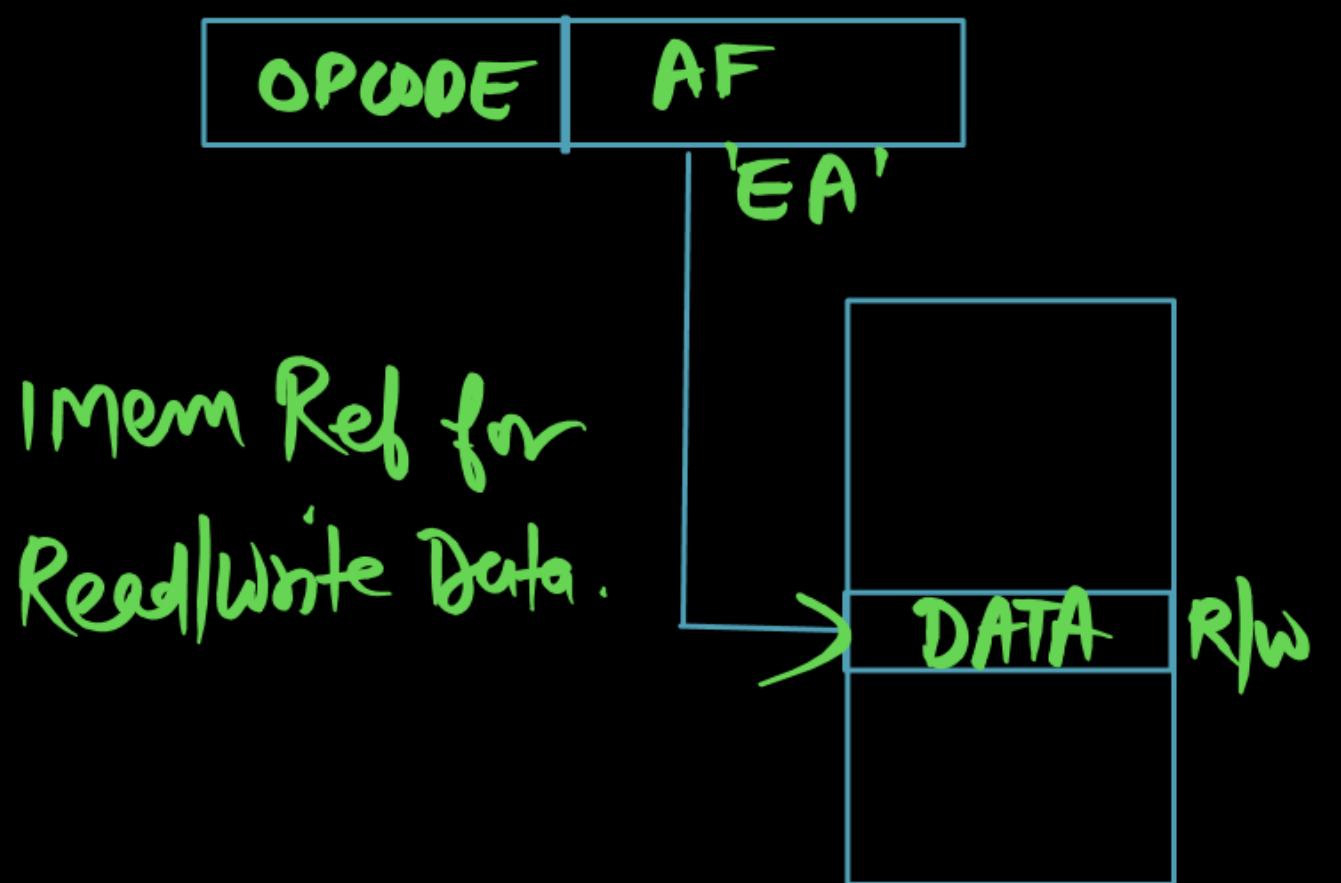
11

500



Addressing Modes

Memory Direct / Absolute AM



In this AM operand
are present in the
Memory & Instruction Contain
the effective Address.

Address field of the Instrn Contain EA

Note Direct AM Used to Access the Variable.

⑨1

MOV R₀ [6000]

R₀ ∈ M[6000]

R₀ = 500 Ans

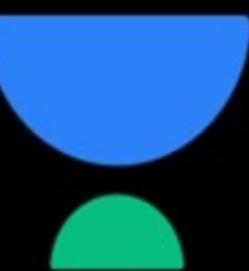
⑨2

ADD R₁ [500]

R₁ ∈ R₁ + M[500]

R₁ ∈ R₁ + 11 Ans

50	11
6000	500





2.5 ADDRESSING MODES

We have now seen some simple examples of assembly language programs. In general, a program operates on data that reside in the computer's memory. These data can be organized in a variety of ways. If we want to keep track of students' names, we can write them in a list. If we want to associate information with each name, for example to record telephone numbers or marks in various courses, we may organize this information in the form of a table. Programmers use organizations called *data structures* to represent the data used in computations. These include lists, linked lists, arrays, queues, and so on.

Programs are normally written in a high-level language, which enables the programmer to use constants, local and global variables, pointers, and arrays. When translating a high-level language program into assembly language, the compiler must be able to implement these constructs using the facilities provided in the instruction set of the computer in which the program will be run. The different ways in which the location of an operand is specified in an instruction are referred to as *addressing modes*. In this section we present the most important addressing modes found in modern processors. A summary is provided in Table 2.1.

Table 2.1 Generic addressing modes

Name	Assembler syntax	Addressing function
Immediate	#Value	Operand = Value
Register	R i	EA = R i
Absolute (Direct)	LOC	EA = LOC
Indirect	(R i)	EA = [R i]
	(LOC)	EA = [LOC]
Index	X(R i)	EA = [R i] + X
Base with index	(R i ,R j)	EA = [R i] + [R j]
Base with index and offset	X(R i ,R j)	EA = [R i] + [R j] + X
Relative	X(PC)	EA = [PC] + X
Autoincrement	(R i)+	EA = [R i]; Increment R i
Autodecrement	-(R i)	Decrement R i ; EA = [R i]

EA = effective address

Value = a signed number

Why Addressing Modes ??

Programs are normally written in a high-level language, which enables the programmer to use constants, local and global variables, pointers, and arrays. When translating a high-level language program into assembly language, the compiler must be able to implement these constructs using the facilities provided in the instruction set of the computer in which the program will be run. The different ways in which the location of an operand is specified in an instruction are referred to as *addressing modes*. In this section we present the most important addressing modes found in modern processors.

Tomorrow : 6PM

Sunday : 10AM

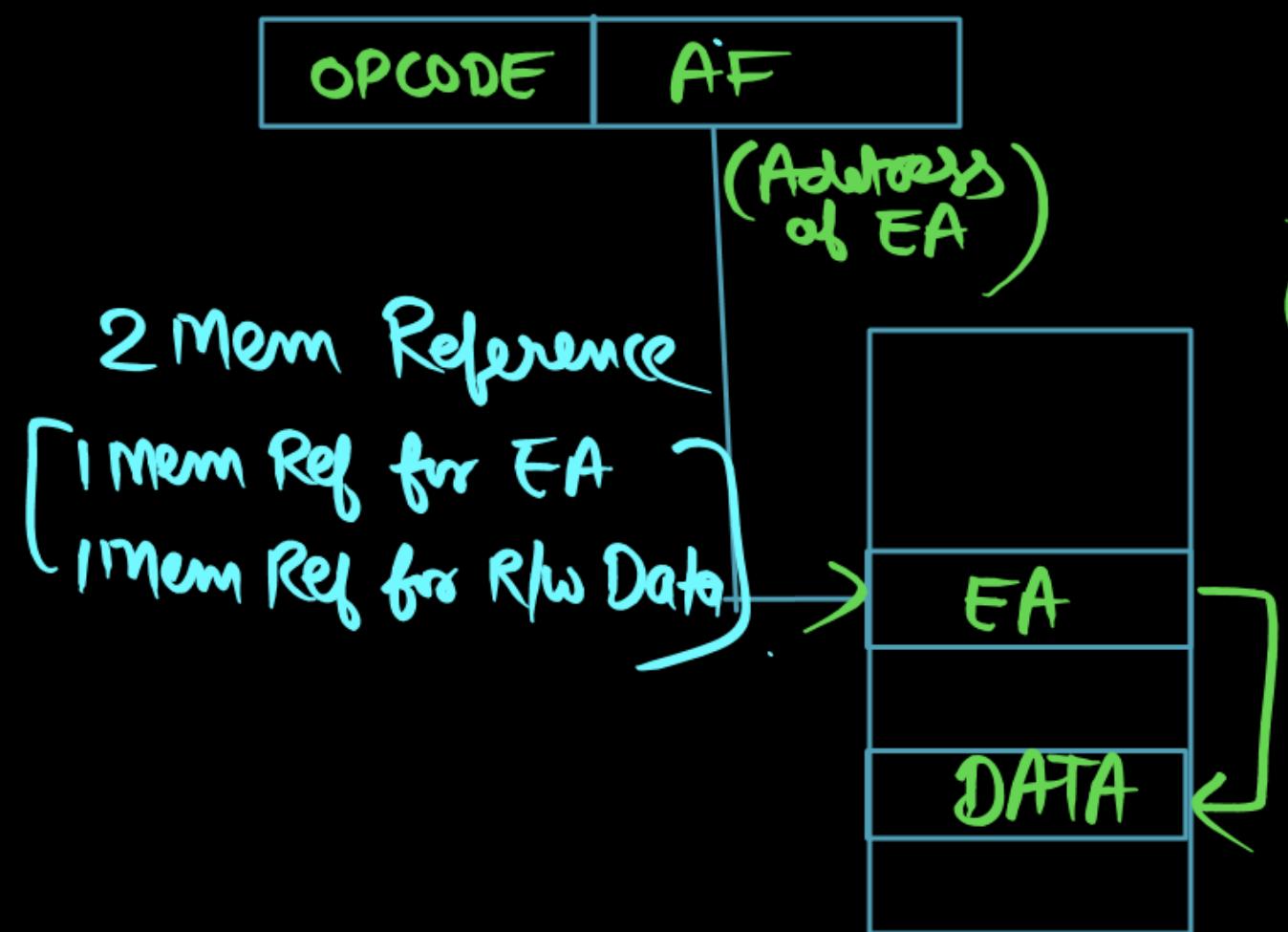
Addressing Modes

③ Memory Indirect AM

In

Indirect AM operand are present in the memory, effective Address(EA) is also present in the memory

Instruction Contain the Address of Effective Address.



Note

Indirect AM are Used to Implement the Pointers.

(e)

MOV R₀ @6000

R₀ \leftarrow M[6000]

R₀ \in M(500)

R₀ = 11 Ans

(e)

ADD R₂ ((600))

R₂ \leftarrow R₂ + M[6000]

R₂ \in R₂ + M(500)

R₂ \leftarrow R₂ + 11 Ans





- ① Immediate AM [$I | \#$]
- ② Direct | Absolut AM []
- ③ Mem Indirect AM (@ | ())

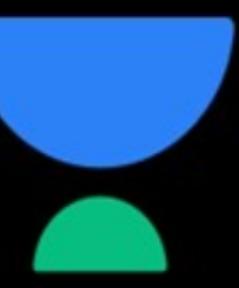
Addressing Modes

- ④ Register Direct AM : This AM is similar to Mem Direct AM But the Difference is Here Operand Present in the Register instead of Memory That Register Address (Ref. Name) is Maintained in the Address field of the Instruction.
- OPCODE AF (R₂)
- Ref Name
- R₀
R₁
R₂
R₃
R₄
DATA
⋮
- I Ref Reference for Read/Write Data
- Register file



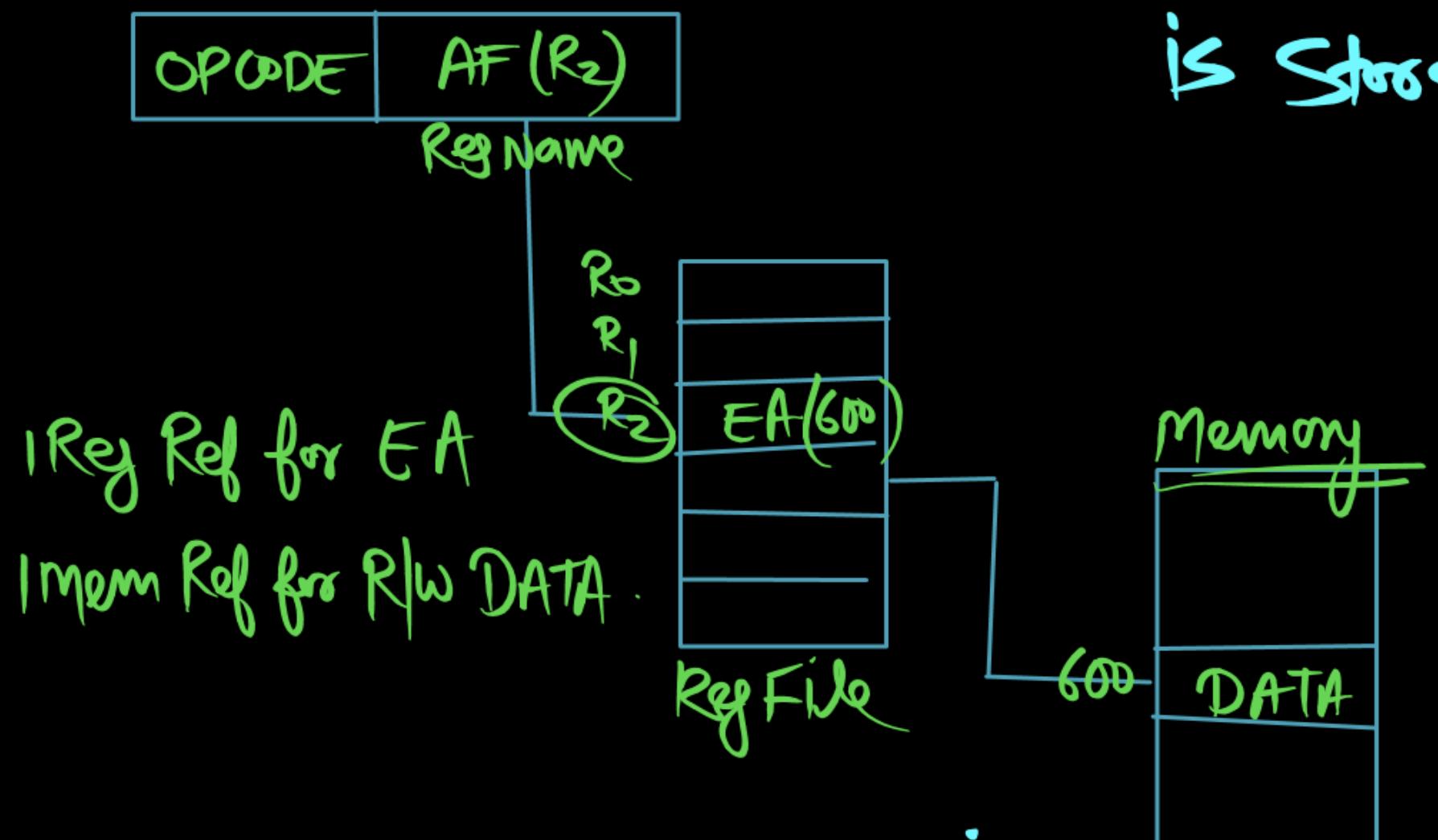
M0V R₀ R₁

R₀ ← R₁



Addressing Modes

⑤ Register Indirect AM



In this AM Operand are present in the memory & effective Address is stored in the Register.

~~MOV R₁, (R₂)~~

~~(OR)~~

MOV R₁ @R₂

R₁ ∈ @R₂

R₁ ∈ M[R₂]

R₁ ∈ M[4000]

R₁ ∈ 2L

Ans

R₂ = 4000

4000



Addressing Modes



Displacement AM

① PC Relative AM

$$EA = \frac{\text{Current PC Value}}{\text{Value}} + AF/offset$$

or

$$EA = \frac{\text{Updated Current PC Value}}{\text{Value}} + \frac{\text{Relative Value}}{AF/offset}$$

② Base Ref AM

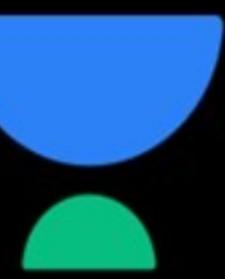
$$EA = \frac{BR}{\text{Value}} + \frac{OFFSET}{AF}$$

③ Index - Ref AM.

$$EA = \frac{XR}{\text{Value}} + AF/offset$$



- ① Immediate AM [$I | \#$]
- ② Direct | Absolut AM []
- ③ Mem Indirect AM (@ | ())



- ① Immediate AM
 - ② Mem Direct | Absolute AM
 - ③ Mem Indirect AM
 - ④ Register Direct AM
 - ⑤ Register Indirect AM
 - ⑥ PC-Relative AM
 - ⑦ Base Reg AM
 - ⑧ Index Reg AM
 - ⑨ Auto Decrement AM
 - ⑩ Auto Increment AM
 - ⑪ Implied | Implicit AM.
- Displacement AM.

Addressing Modes

④ 210 Auto Decrement & Increment AM: This AM is similar to Register Indirect AM in which Reg Value Decrement | Increment Respectively.

Decrement : Pre Decrement

Increment : Post Increment



Addressing Modes

⑪

Implied/Implicit AM

Operands are

Present in the opcode itself.

OPCODE



Type of
operation



CLC : Clear Carry [$CY = 0$]

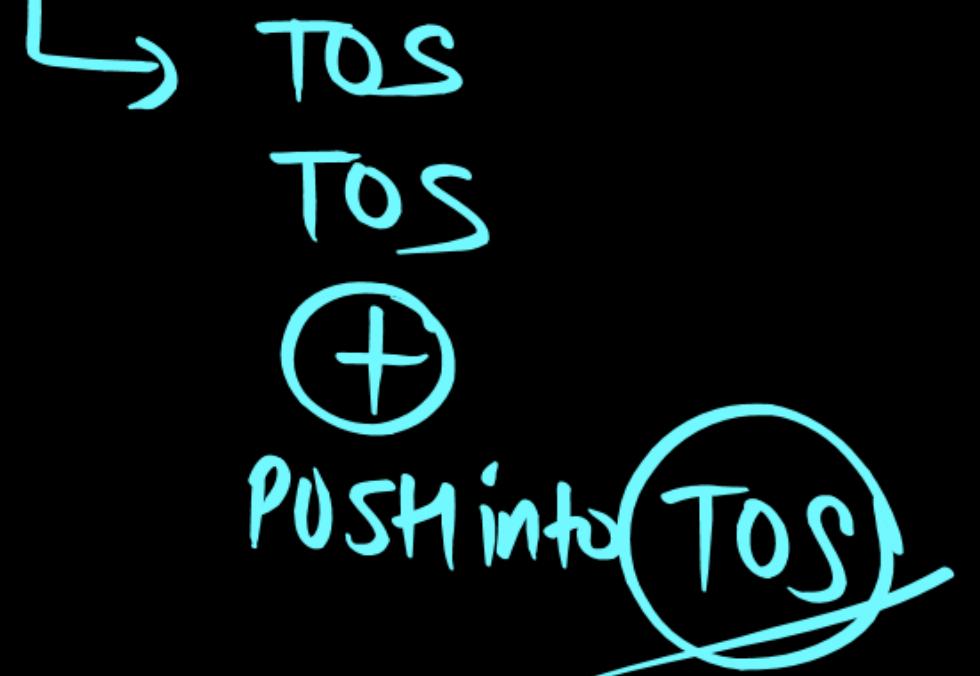
STC : Set Carry [$CY = 1$].

Addressing Modes

(e)

Stack Based org

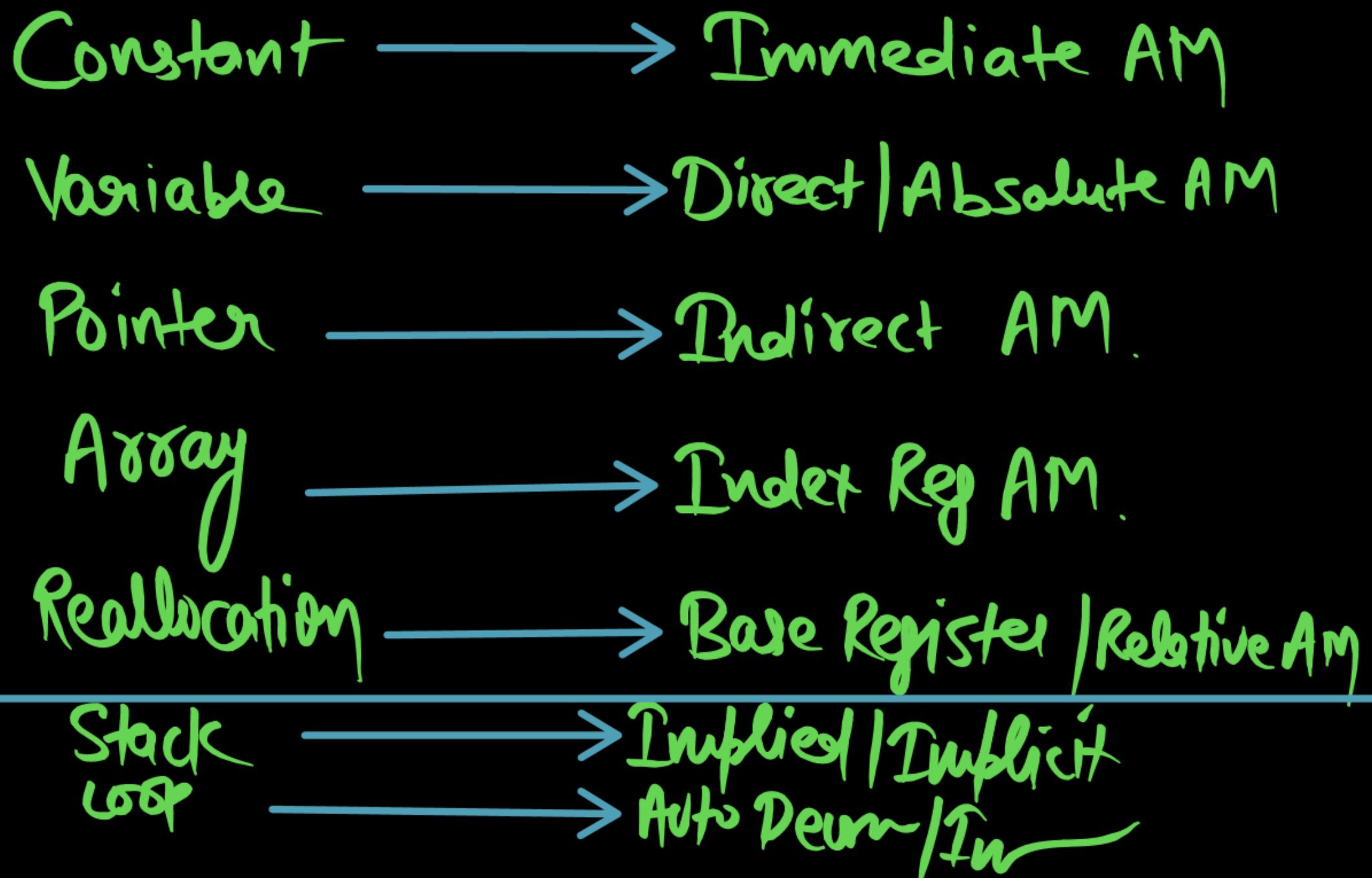
ADD



Stack Based Used
Implicit/Implied AM.

Addressing Modes

V.V.Imp.





Addressing Modes

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

Instruction

operand



(a) Immediate

Instruction

A EA

Memory



(b) Direct

Instruction

A

Memory

operand

EA



(c) Indirect

Instruction

R

operand

Registers

(d) Register

Instruction

R

Memory

EA

Memory

operand

Registers

(e) Register Indirect

Instruction

R

A

Memory

Registers

Memory

operand

(f) Displacement

Instruction

Implicit

Top of stack
Register

(g) Stack

13.1 ADDRESSING MODES

The address field or fields in a typical instruction format are relatively small. We would like to be able to reference a large range of locations in main memory or, for some systems, virtual memory. To achieve this objective, a variety of addressing techniques has been employed. They all involve some trade-off between address range and/or addressing flexibility, on the one hand, and the number of memory references in the instruction and/or the complexity of address calculation, on the other. In this section, we examine the most common addressing techniques, or modes:

- Immediate
- Direct
- Indirect
- Register
- Register indirect
- Displacement
- Stack



These modes are illustrated in Figure 13.1. In this section, we use the following notation:

A = contents of an address field in the instruction

R = contents of an address field in the instruction that refers to a register

EA = actual (effective) address of the location containing the referenced operand

(X) = contents of memory location X or register X

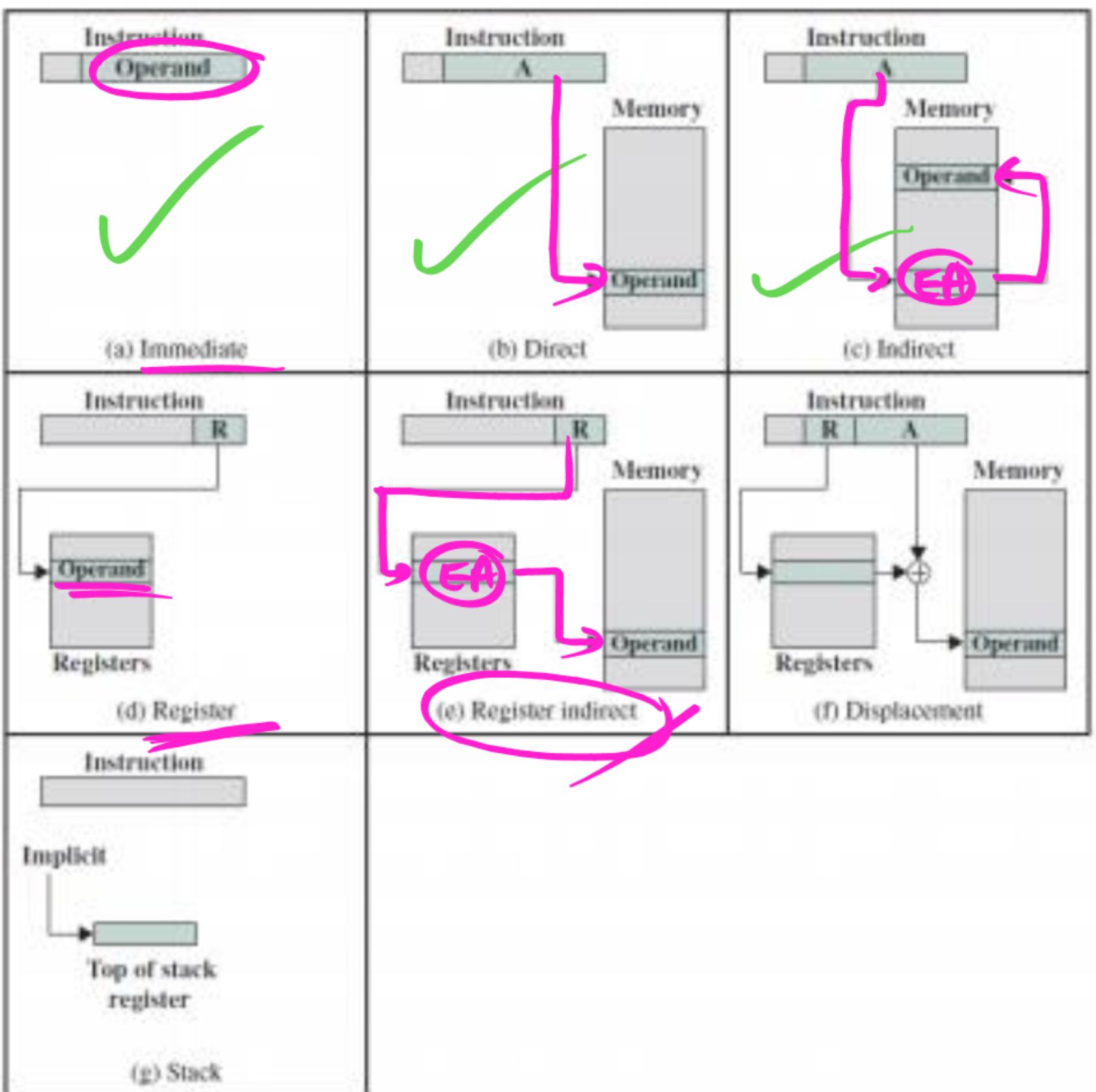
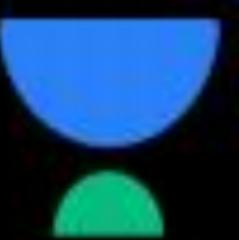


Figure 13.1 Addressing Modes

Table 13.1 Basic Addressing Modes

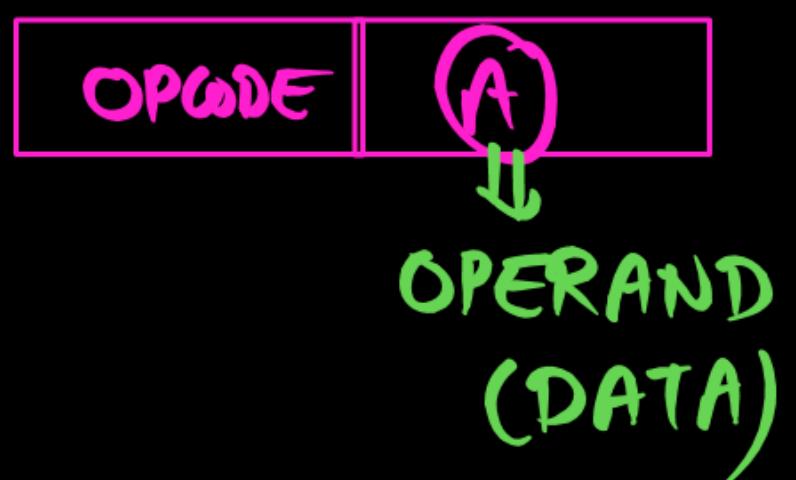
Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	EA = R	No memory reference	Limited address space
Register indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA = A + (R)	Flexibility	Complexity
Stack	EA = top of stack	No memory reference	Limited applicability

1
2
3
4
5

Immediate Addressing

The simplest form of addressing is **immediate addressing**, in which the operand value is present in the instruction

Operand = A



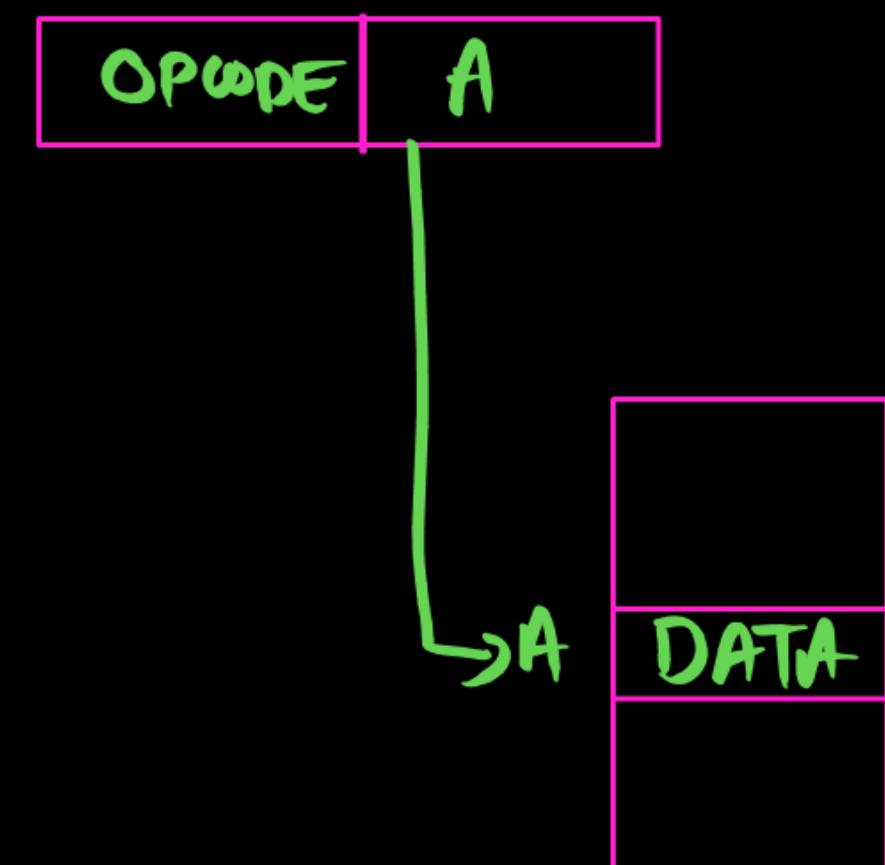


Direct Addressing



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

$$EA = A$$



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 known as indirect addressing:

$$EA = (A)$$



Register Addressing

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

$$EA = R$$

Register Indirect Addressing

Just as register addressing is analogous to direct addressing, **register indirect addressing** is analogous to indirect addressing. In both cases, the only difference is whether the address field refers to a memory location or a register. Thus, for register indirect address,

$$EA = (R)$$

Reg Indirect AM WHY ?

- ① To Short the Instruction Length .
- ② Faster Accessing of Register .

Problems

13.1 Given the following memory values and a one-address machine with an accumulator, what values do the following instructions load into the accumulator?

- Word 20 contains 40.
- Word 30 contains 50.
- Word 40 contains 60.
- Word 50 contains 70.

20	40
30	50
40	60
50	70

- (a) 20 Ans
- (b) 40 Ans
- (c) 60 Ans
- (d) 30 Ans
- (e) 50 Ans
- (f) 70 Ans

13.6 / KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

485

- a. LOAD IMMEDIATE 20
- b. LOAD DIRECT 20
- c. LOAD INDIRECT 20
- d. LOAD IMMEDIATE 30
- e. LOAD DIRECT 30
- f. LOAD INDIRECT 30

- 13.4** Consider a 16-bit processor in which the following appears in main memory, starting at location 200:

200	Load to AC	Mode
201		500
202		Next instruction

The first part of the first word indicates that this instruction loads a value into an accumulator. The Mode field specifies an addressing mode and, if appropriate, indicates a source register; assume that when used, the source register is R1, which has a value of 400. There is also a base register that contains the value 100. The value of 500 in location 201 may be part of the address calculation. Assume that location 399 contains the value 999, location 400 contains the value 1000, and so on. Determine the effective address and the operand to be loaded for the following address modes:

- a. Direct
- b. Immediate
- c. Indirect
- d. PC relative
- e. Displacement
- f. Register
- g. Register indirect
- h. Autoindexing with increment, using R1

Addressing Mode	Effective Address	Content Of AC	Address	Memory
Direct address		$PC = 200$	200	Load to AC Mode
Immediate Operand		$R1 = 400$	201	Address = 500
Indirect Address		$XR = 100$	202	Next instruction
Relative address		AC	399	450
Indexed address			400	700
Register			500	800
Register Indirect			600	900
Autoincrement			702	325
Autodecrement			800	300

Numerical example for addressing modes.

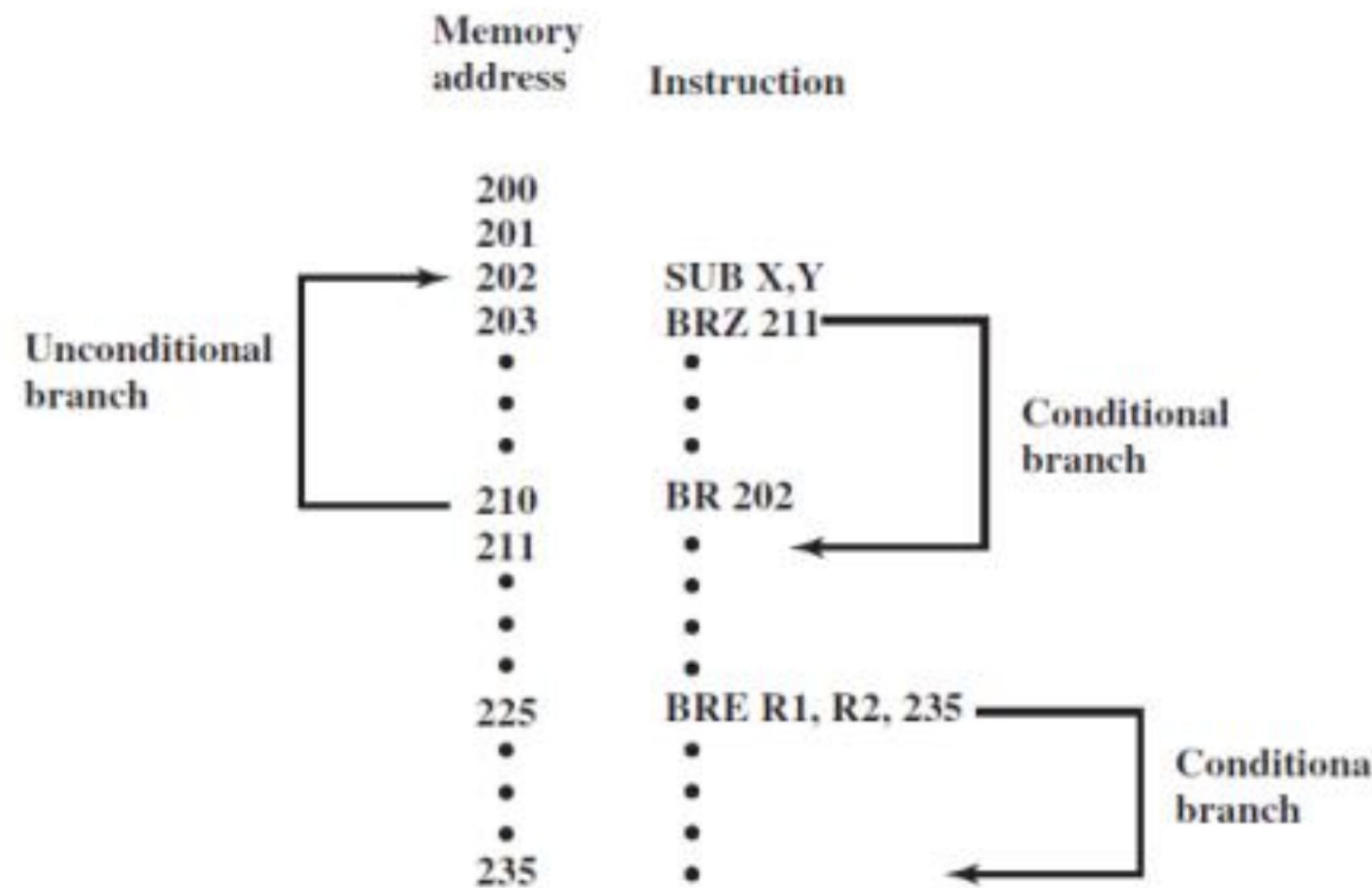


Figure 12.7 Branch Instructions

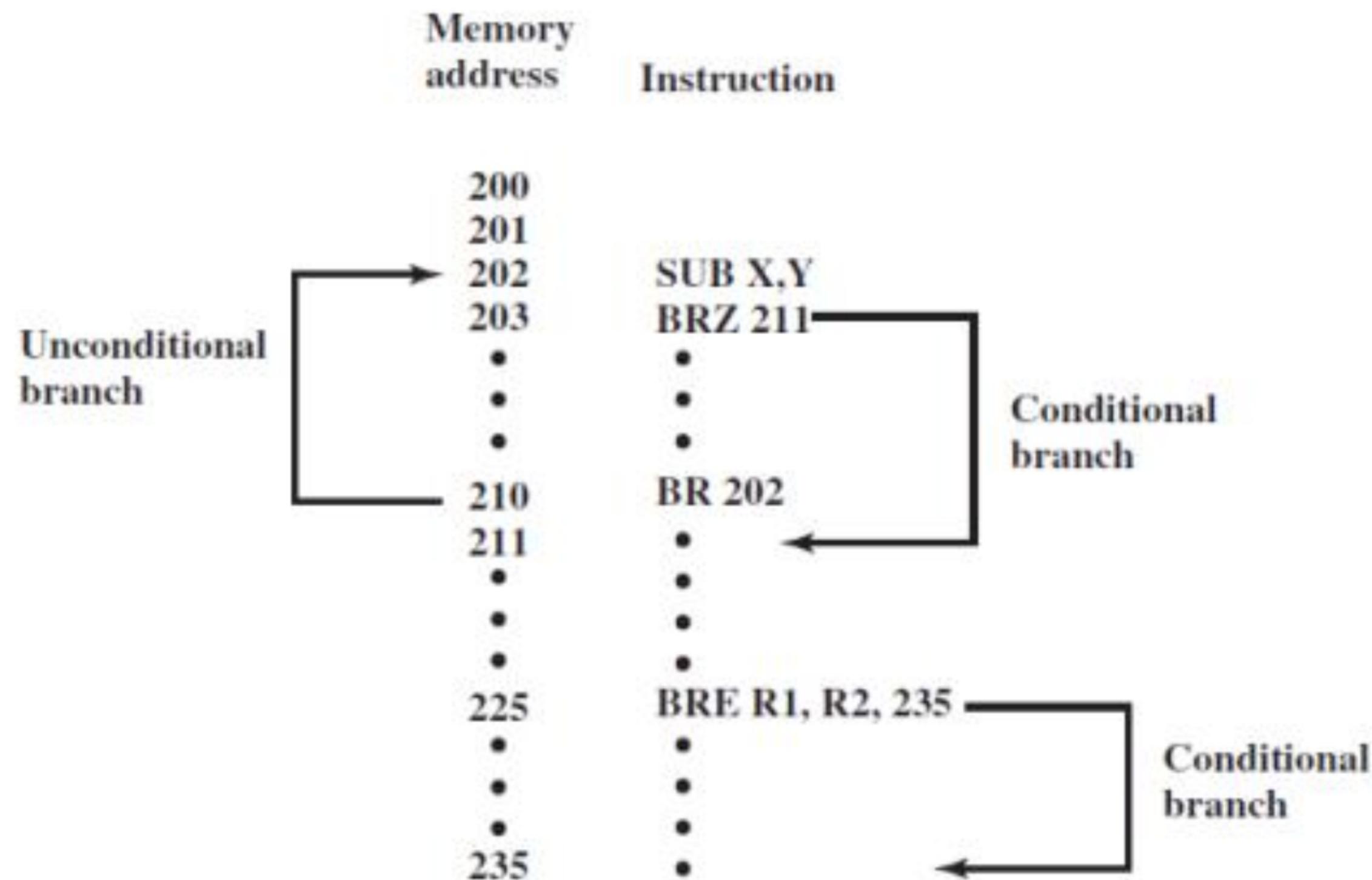


Figure 12.7 Branch Instructions

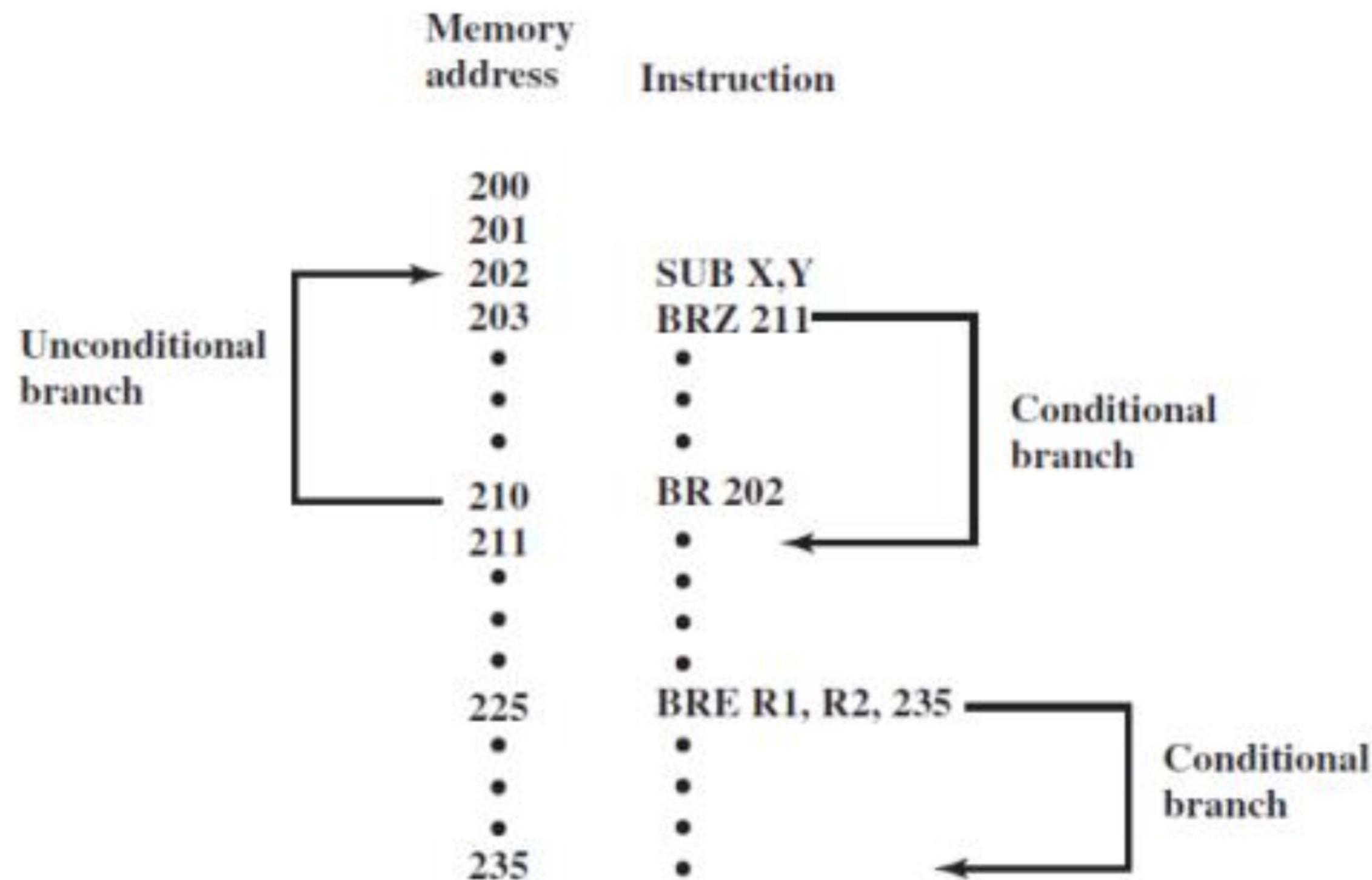


Figure 12.7 Branch Instructions

Eight addressing modes for the load instruction

Mode	Assembly Convention	Register Transfer
Direct address	LD ADR	$AC \leftarrow M[ADR]$
Indirect address	LD @ADR	$AC \leftarrow M[M[ADR]]$
Relative address	LD \$ADR	$AC \leftarrow M[PC + ADR]$
Immediate operand	LD#NBR	$AC \leftarrow NBR$
Index addressing	LD ADR(X)	$AC \leftarrow M[ADR + XR]$
Register	LD R1	$AC \leftarrow R1$
Register indirect	LD (R1)	$AC \leftarrow M[R1]$
Autoincrement	LD (R1)+	$AC \leftarrow M[R1], R1 \leftarrow R1 + 1$

Q.

In which of the following addressing modes, operand is NOT
A part of instruction?

[MSQ]

- A Immediate
- C Indirect

- B Direct
- D Register

OPCODE) AF
(DATA)

Immediate



Q.1

The most appropriate matching for the following pairs

- | | | |
|------------------------------|-----|--------------|
| X. Indirect addressing | - 2 | 1. Loops |
| Y. Immediate addressing | - 3 | 2. Pointers |
| Z. Auto decrement addressing | ↳ 1 | 3. Constants |

[GATE - 2000: 1 Mark]

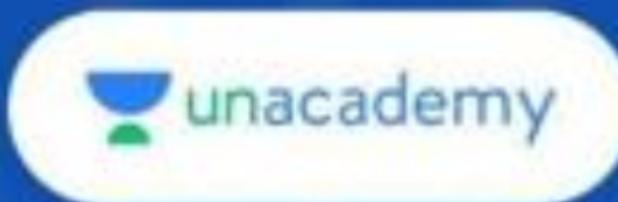
A X - 3 Y - 2 Z - 1

B X - 1 Y - 3 Z-2

C ✓ X - 2 Y - 3 Z - 1

D X - 3 Y - 1 Z - 2

Ans(C)



Unacademy Learners' excel in GATE 2024

7 Rankers secured AIR 1



AIR 1 EE

Shivam Garg

Unacademy Plus Learner



AIR 1 CSIT

Piyush Kumar

Unacademy Plus Learner



AIR 1 ECE

Raja Majhi

Unacademy Plus Learner



AIR 1 CH

Adarsh Rai

Unacademy Free Learner



AIR 1 IN

Rishabh Gupta

Unacademy Free Learner



AIR 1 PI

Kumar Vishesh

Unacademy Plus Learner



AIR 1 NM

Prince Kumar

Unacademy Plus Learner



GATE subscription is now at Rs. 5999 only!

Flat 50%* off on all GATE & ESE Subscriptions*

Offer extended till Nov 16

Subscribe Now

Use code
UNLOCK

For more details, contact: **85 85 85 85 85**

*T&C apply, as available on the platform.

CSDA



Ankit Doyal



Chandan Jha



Pankaj Sharma



Khaleel Khan



Mallesham Devsane



Satish Yadav



Rahul Joshi



Vijay Agarwal

GATE subscription is now at Rs. 5999 only!



Join our All in one Batch for GATE - CS 2025
started on 11th October

Target GATE 2025

Duration: 3 Months

Offer Price: ₹11,998/- ~~₹11,998/-~~ ₹5,999/-

Subscribe Now

Use code

CSDA

Offer extended till Nov 16

For more details, contact: 85 85 85 85 85

50%
OFF



GATE subscription is now at Rs. 5999 only!



Join our Samarth and Nischay for
GATE & PSU - CS 2026/27 starting on 15th November

Target GATE 2026

Duration: 15 months

Offer Price: ₹29,998/- ₹14,999/-

Subscribe Now

Target GATE 2027

Duration: 24 months

Offer Price: ₹39,998/- ₹19,999/-

Use code

Offer extended till Nov 16

For more details, contact: 85 85 85 85 85

50%
OFF

CSDA

Samarth Batch

for GATE 2026 - CS

 Starts on Nov 15th 2024

We Start With:

Linear Algebra - Part I By Rahul at 3:00 PM

Enroll Now

Use code:

*T&C apply, as available on the platform.

For More Details Call - 85858 58585



Ankit Kumar



Satish Kumar
Yadav



Pankaj Sharma



Khaleel Ur
Rehman Khan



Mallesham Devsane



Chandan Jha



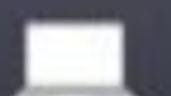
Rahul



Vijay Kumar Agarwal

3T/A Hinglish

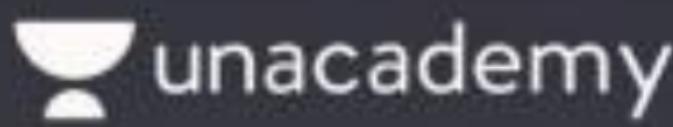
 Batch C



CS



Live



Nischay Batch

for GATE 2027 - CS

Starts on Nov 15th 2024

We Start With:

Linear Algebra - Part I By Rahul at 3:00 PM

Enroll Now

Use code:



Ankit Kumar



Satish Kumar
Yadav



Pankaj Sharma



Khaleel Ur
Rehman Khan



Mallesham Devsane



Chandan Jha

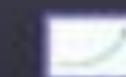


Rahul

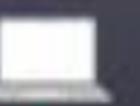


Vijay Kumar Agarwal

3I/A Hinglish



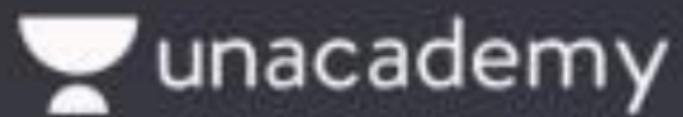
Batch C



CS



Live



All in one Practice

CS & IT

Starts on Oct 11th 2024

We Start With:

Operating Systems By Khaleel
at 6:30 PM

Enroll Now

CSDA
Use code:

*T&C apply, as available on the platform.

For More Details Call - 85858 58585



Ankit Kumar



Satish Kumar
Yadav



Pankaj Sharma



Khaleel Ur
Rehman Khan



Mallesham Devsane



Chandan Jha



Rahul



Vijay Kumar Agarwal

Hinglish

Batch D

CS & IT

Live + Recorded



UNACADEMY GATE SCHOLARSHIP TEST (UGST)

- Win Laptop, Tablet, exciting rewards
- Get up to 100% scholarship* on all GATE Subscriptions
- Assured Gift Vouchers to 50 Learners*

November 24 | 11 AM



GATE & ESE

All India Scholarship Test

Specially Curated by Top Educators
win upto 100% Scholarship



24th Nov



11:00 AM

Enroll Now

Rank 1 - 3

100% Scholarship

Rank 4 - 10

90% Scholarship

Rank 11 - 50

75% Scholarship

Rank 51 - 200

60% Scholarship

Rank 201 - 500

55% Scholarship

501 and Above

50% Scholarship

GET upto 90% Scholarship* for GATE 2025/26 Online Subscriptions!

Eligibility Criteria	Scholarship% (GATE 2025)
GATE Rank 1 - 500	90%
GATE Rank 501 - 1000)	80%
GATE Rank 1001 - 3000	70%
GATE Rank 3001 - 5000	60%
ESE Prelims Qualified	90%
ESE Mains Qualified	80%
B.E / B. Tech (80% & above Marks)	70%
Students of IITs & NITs	60%
Students of Top 50 NIRF Colleges	55%



Scan the QR code to
Fill the form
and crack the scholarship



Note : Valid for GATE 2023/24 candidates only

Avail Now

For more details, contact: 85 85 85 85 85

Plus Features



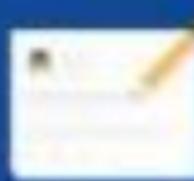
Raise a Hand



Class
Handouts



In-class
polls



Daily Practice
Problems



Educator
Community



Learn live from
the comfort of
your home



Study
Material
(PDF format)



Unlimited
access to all
courses



#1
Top Educators
of India on
one platform

CRACK GATE & ESE WITH UNACADEMY SUBSCRIPTION

PLUS

Basic plan to get started



- India's Best Educators
- Interactive Live Classes
- Structured Courses & PDFs
- Live Tests & Quizzes
- Digital Books Access

ICONIC

Take your GATE & ESE preparation to next level with personal



- Everything included in plus
- 1:1 Live mentorship
- Physical Books**
-  Everything included with PLUS

25% OFF

GATE PARIKSHA

TEST Series for CS & IT 2025

Total **103 Tests** & 3245 Questions

Test Name	Total No. of Tests	Questions per Test
Topic Wise Tests	47	15
Subject wise Tests	24	25
Multi-Subject tests	4	30
Full Syllabus Tests	15	65
GATE PYQ Tests	13	65

@₹1499!
only



25% OFF

GATE PARIKSHA

TEST Series for DS & AI 2025

Total **70 Tests** & 2100 Questions

Test Name	Total No. of Tests	Questions per Test
Topic Wise Tests	31	15
Subject wise Tests	19	25
Multi-Subject tests	4	30
Full Syllabus Tests	15	65
GATE PYQ Tests	1	65

@₹1499!
only





THANK YOU!

Here's to a cracking journey ahead!