INSTRUCTION SET ARCHITECTURE AND DESIGN

Danang Wahyu Utomo
danang.wu@dsn.dinus.ac.id
+6285 740 955 623

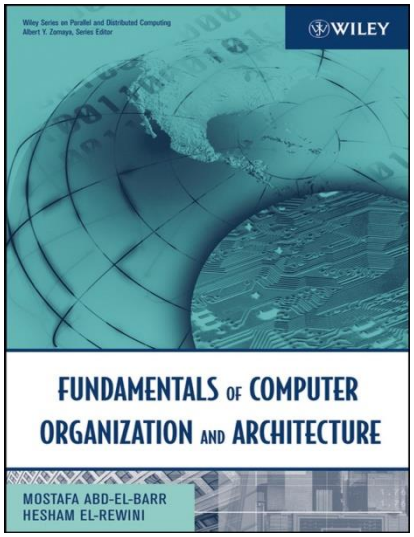
RENCANA KEGIATAN PERKULIAHAN SEMESTER

W	Pokok Bahasan
1	Organisasi dan Arsitektur Komputer
2	Sistem Komputer
3	**Instruction Set Architecture and Design**
4	
5	Computer Arithmetic
6	
7	Review Materi 1-6
8	**Ujian Tengah Semester**

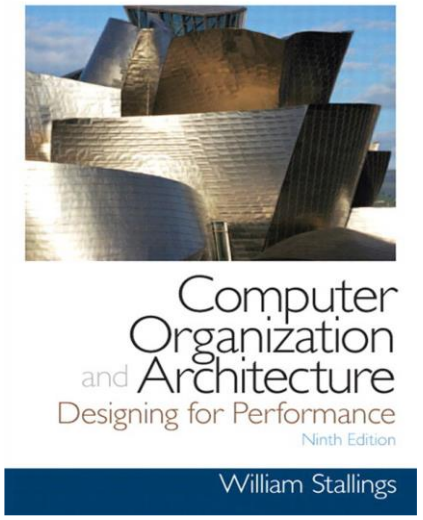
W	Pokok Bahasan
9	Desain Unit Pemrosesan
10	
11	Desain Sistem Memory
12	
13	Desain dan Organisasi Input Output
14	
15	Teknik Desain Pipelining
16	**Ujian Akhir Semester**

Reference

- Mustafa Abd-el-Bhar, Hesham El Rewini – Fundamentals of Computer Organization and Architecture 9th Edition (2004)



- William Stallings – Computer Organization and Architecture Designing For Performance 9th Edition (2013)



Content

- Memory Location And Operation
- Addressing Mode
- Instruction Types

Review Last Week

- Computer ?
- Architecture ?
- Organization ?
- Main component of **Computer** ?
- Main component of **CPU ?**

Memory Locations and Operations

- Main memory can be modeled as an array of millions of adjacent cells
- The cells storing a binary digit (**bit**), having value 1 or 0
- The cells are organized in the form of groups of fixed number
  - An entity consisting of 8 bits is called **byte**
- The entity of *n* bits that can be stored and retrieved in and out of the memory using one basic memory operation is called **word**
- word is the smallest addressable entity
- Two basic memory operations :
  - Write Operation
  - Read Operation

Write Operation

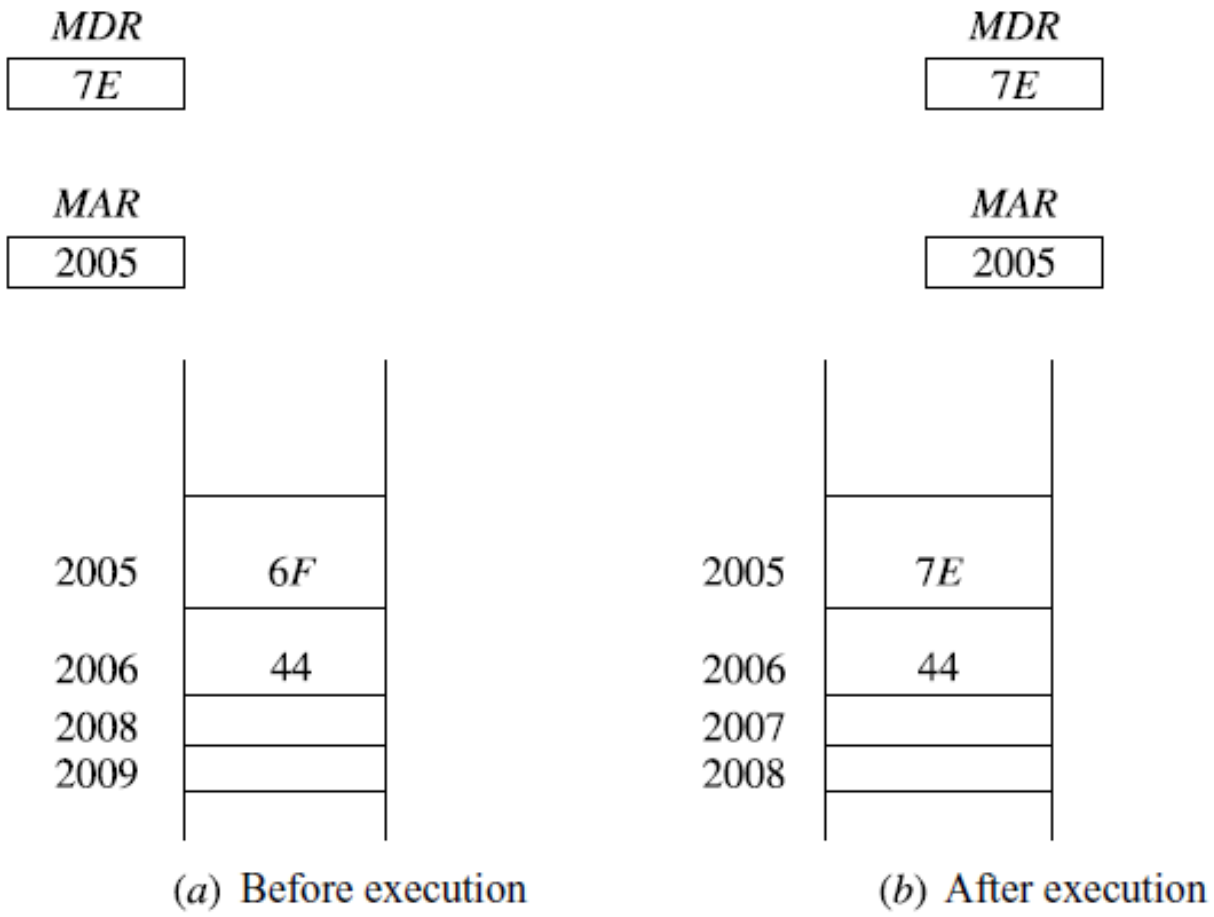
- Move a word in and out of the memory, an address has to be assigned to each word
- The address used to determine the location in the memory in which a given word is to be stored
- Three basic step :
  1. MDR (Memory Data Register)

     the word to be stored into the memory location is first loaded by the CPU into a specified register
  2. MAR (Memory Address Register)

     the address of the location into which the word is to be stored is loaded by the CPU into a specified register
  3. Signal, called write.

     CPU indicate that the word stored in MDR is to be stored in the memory location whose address loaded in the MAR

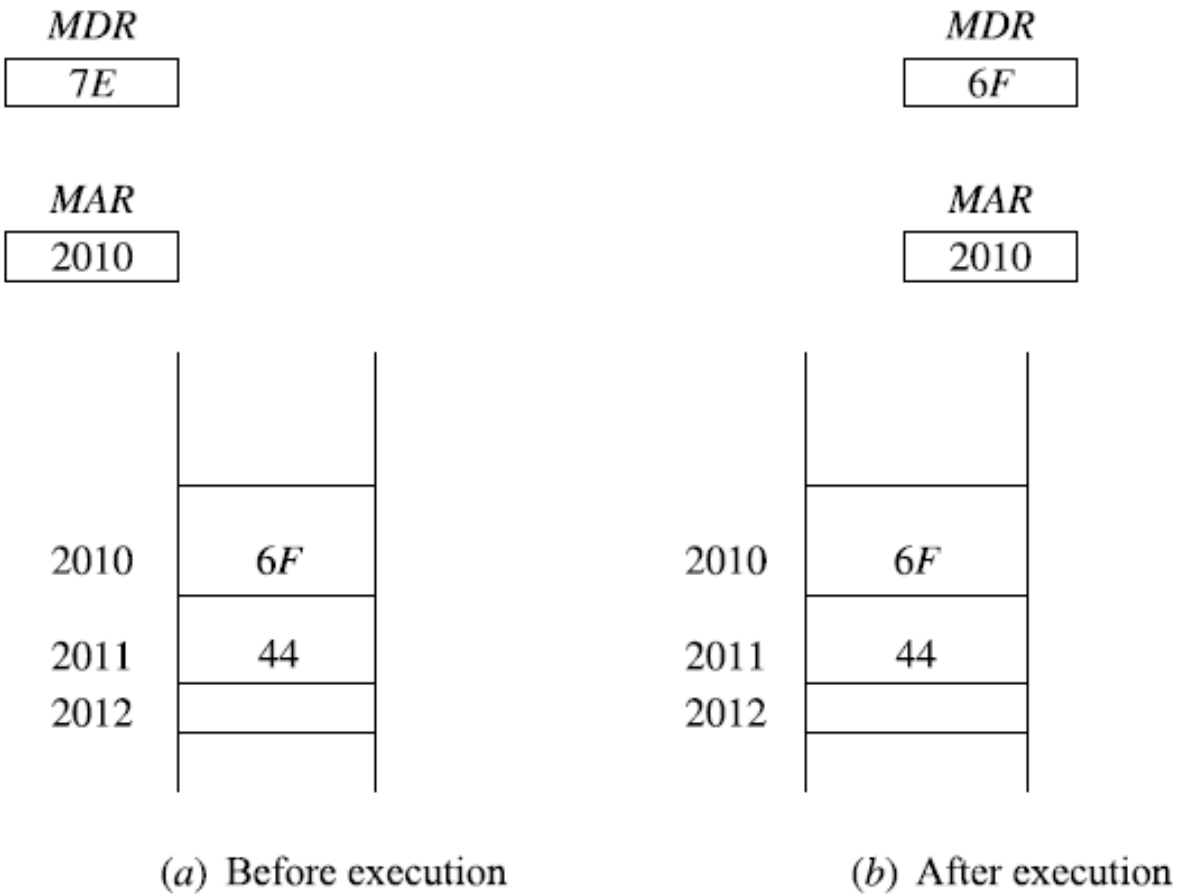
Write Operation



Read Operation

- The address will be used to determine the memory location from which a word is to be retrieved from memory
- Three basic step :
  1. **The address** of the location from which the word is to be read is loaded into the MAR
  2. A Signal, called read. CPU indicate that the word whose address is in the MAR is to be read in the MDR
  3. **The required word** will be loaded by the memory into the MDR ready for use by the CPU

Read Operation

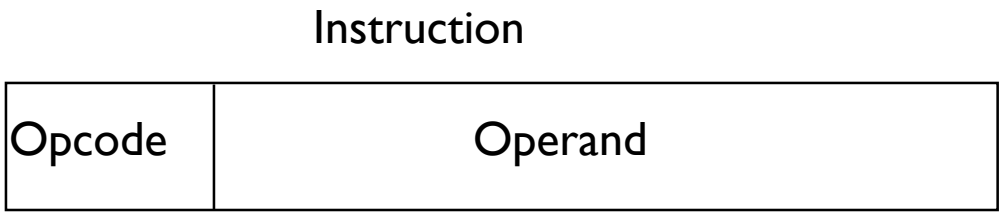


Addressing Mode

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

Immediate Addressing

- No memory reference
- Fast
- Limited Range
- Ex : Operand = A



Immediate Addressing

- Contoh :

  LOAD #9

  the result ACC ← 9

  the value in the behind of # is considered as operand

  ADD Y, #2, #3

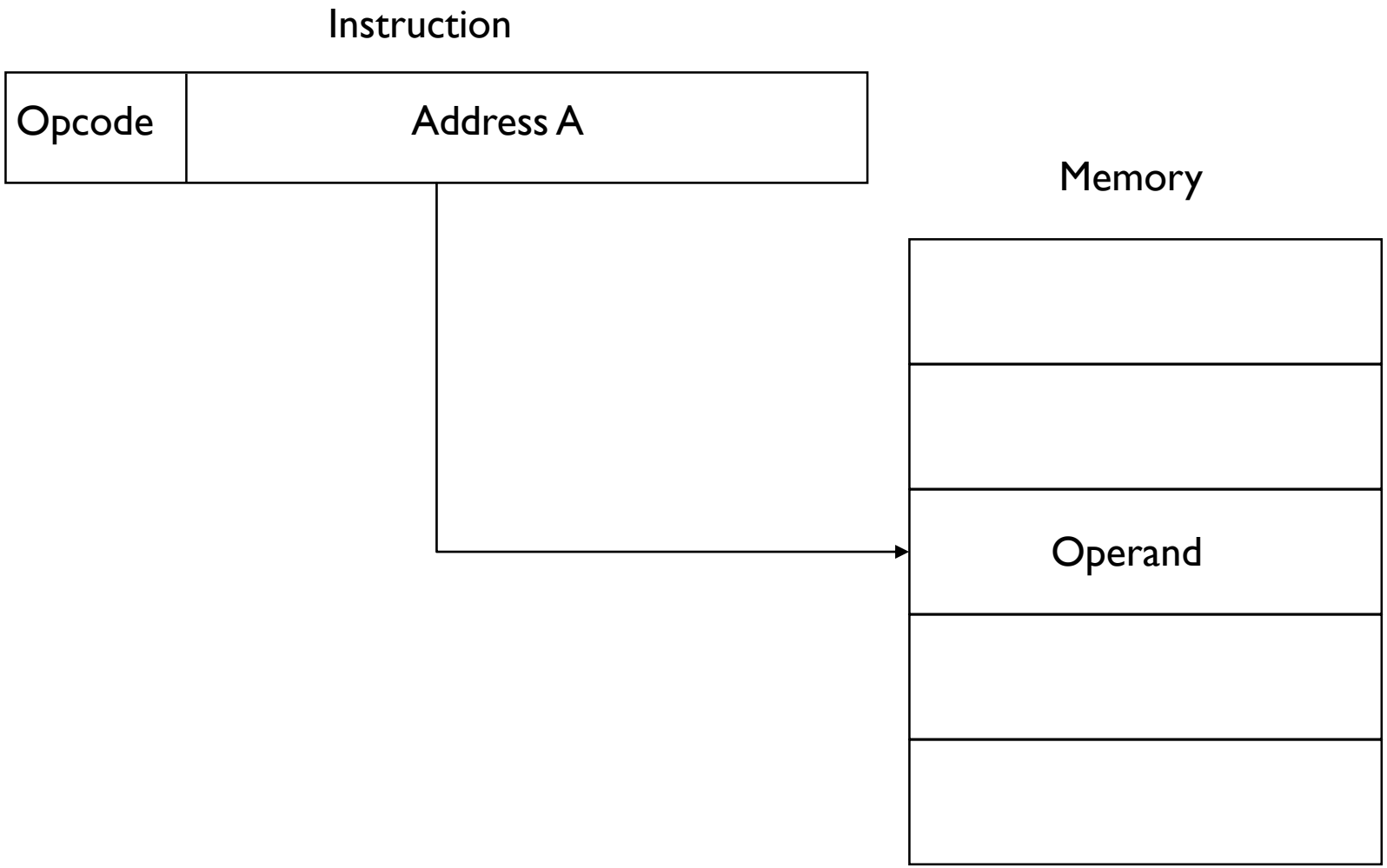
  result Y ← 5, two operand is 2 and 3

  so the result is, 2 + 3 = 5

Direct Addressing

- Address field contains address of operand
- Single memory reference to access data
- No additional calculation to work out effective address
- Limited address space
- Ex : ADD A
  - Add content of A to accumulator
  - Look in memory at address A for operand

Direct Addressing



Direct Addressing

- LOAD 3

  the result

  ACC ← 43

  address of memory 3

  contain 43

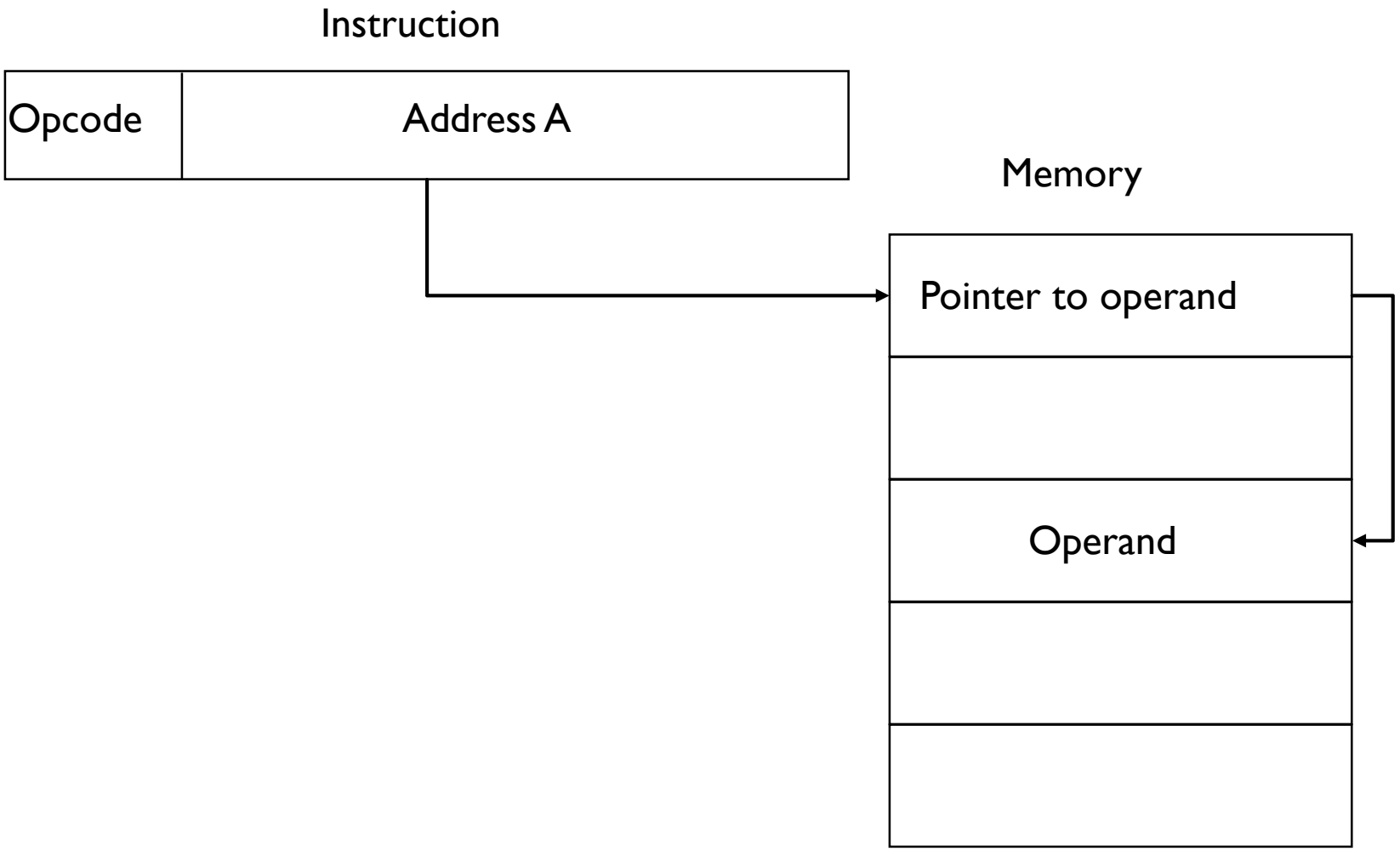
Register	
A	
B	
C	50
D	51
...	
Y	
Z	
100	5
101	6
102	15

Memory	
0	52
1	1
2	12
3	43
...	
50	3
51	7
52	10
...	
100	5
101	6
102	15

Indirect Addressing

- Memory cell pointed to by address field contain the address of (pointer to) the operand
- Large address space
  - 2^n , which n = word length
- Multiple memory accessed to find operand
- Hence slower
- Example : EA = (A)
  - Look in A, find address (A) and look there for operand

Indirect Addressing



Indirect Addressing

- LOAD (50)

  result

  ACC ← 43

  address of memory 50 contain address of memory 3

  address of memory 3 contain 43

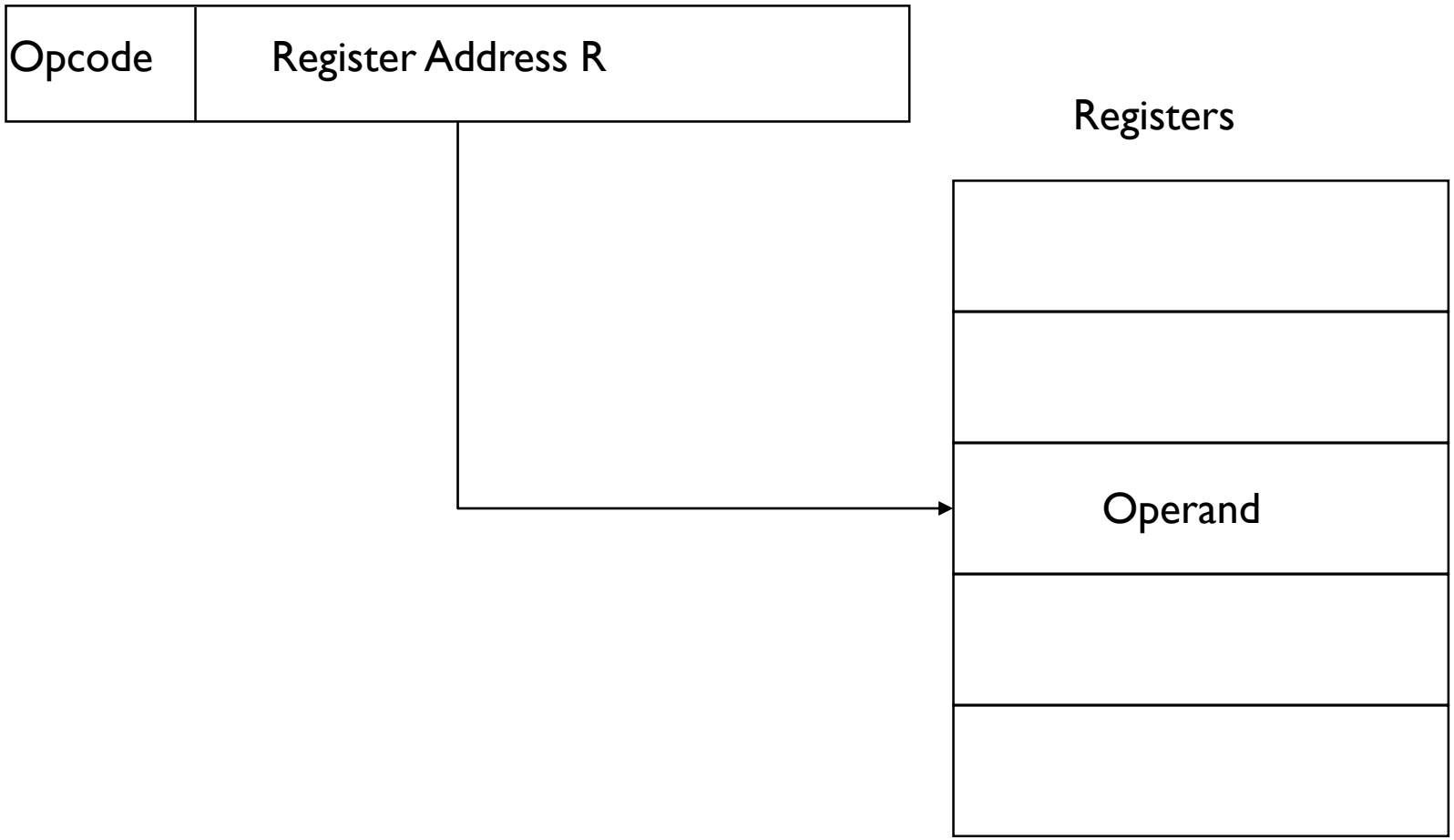
Register	
A	
B	
C	50
D	51
...	
Y	
Z	
100	5
101	6
102	15

Memory	
0	52
1	1
2	12
3	43
...	
50	3
51	7
52	10
...	
100	5
101	6
102	15

Register Addressing

- Similar to direct addressing
- Address field refers to a register
- Limited number of register
- No memory access
- Very fast execution
- Very limited address space
- Multiple register helps performance
- Very small address field needed
  - Shorter instructions
  - Faster instruction fetch
- Ex : EA = R

Register Addressing



Register Addressing

- LOAD C

  result

  ACC ← 50

  addr. of register C contain 50

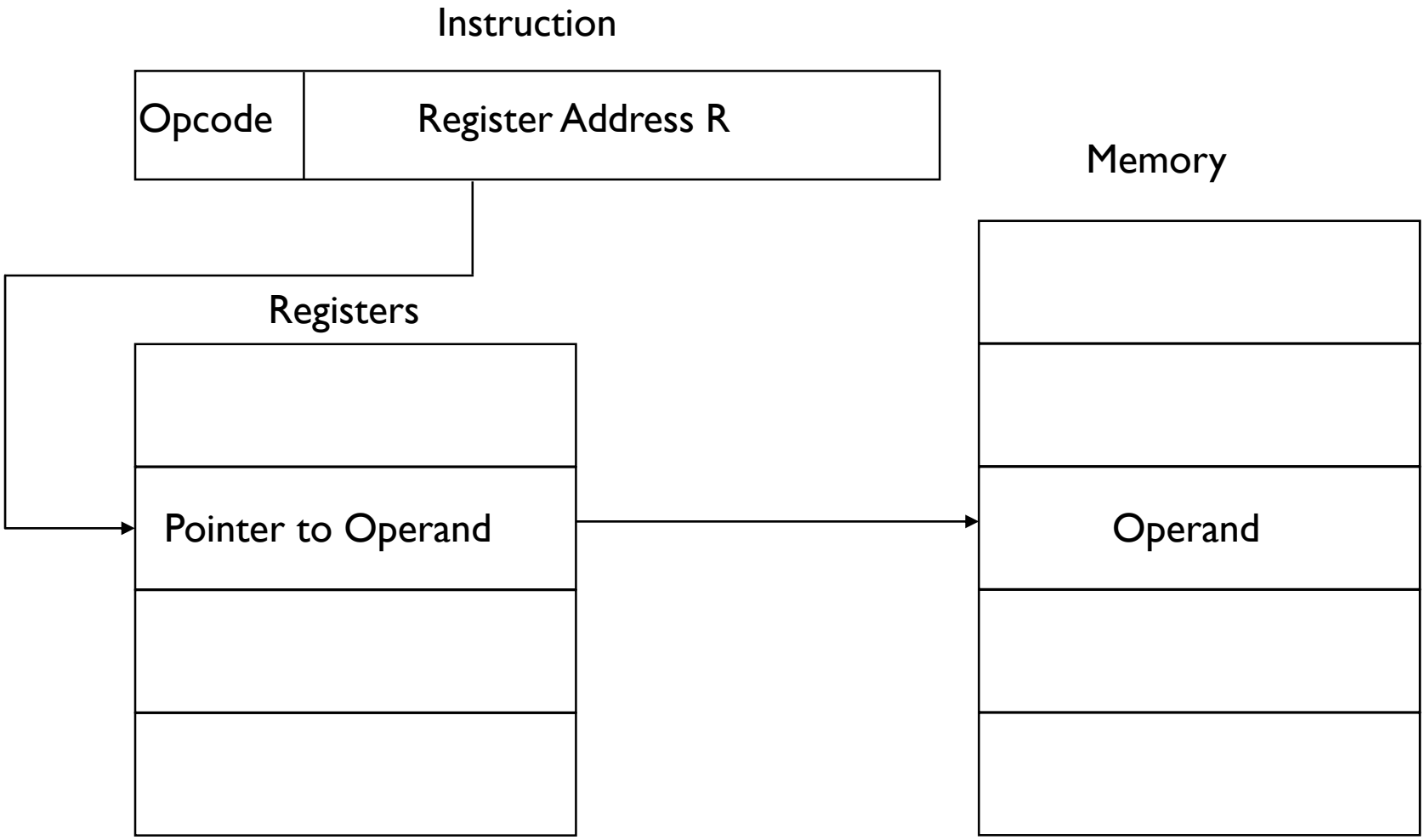
Register	
A	
B	
C	50
D	51
…	
Y	
Z	
100	5
101	6
102	15

Memory	
0	52
1	1
2	12
3	43
…	
50	3
51	7
52	10
…	
100	5
101	6
102	15

Register Indirect Addressing

- Operand is in memory cell pointed to by content of register R
- Large address space 2^n
- One fewer memory access than indirect indirect address

Register Indirect Addressing



Register Indirect Addressing

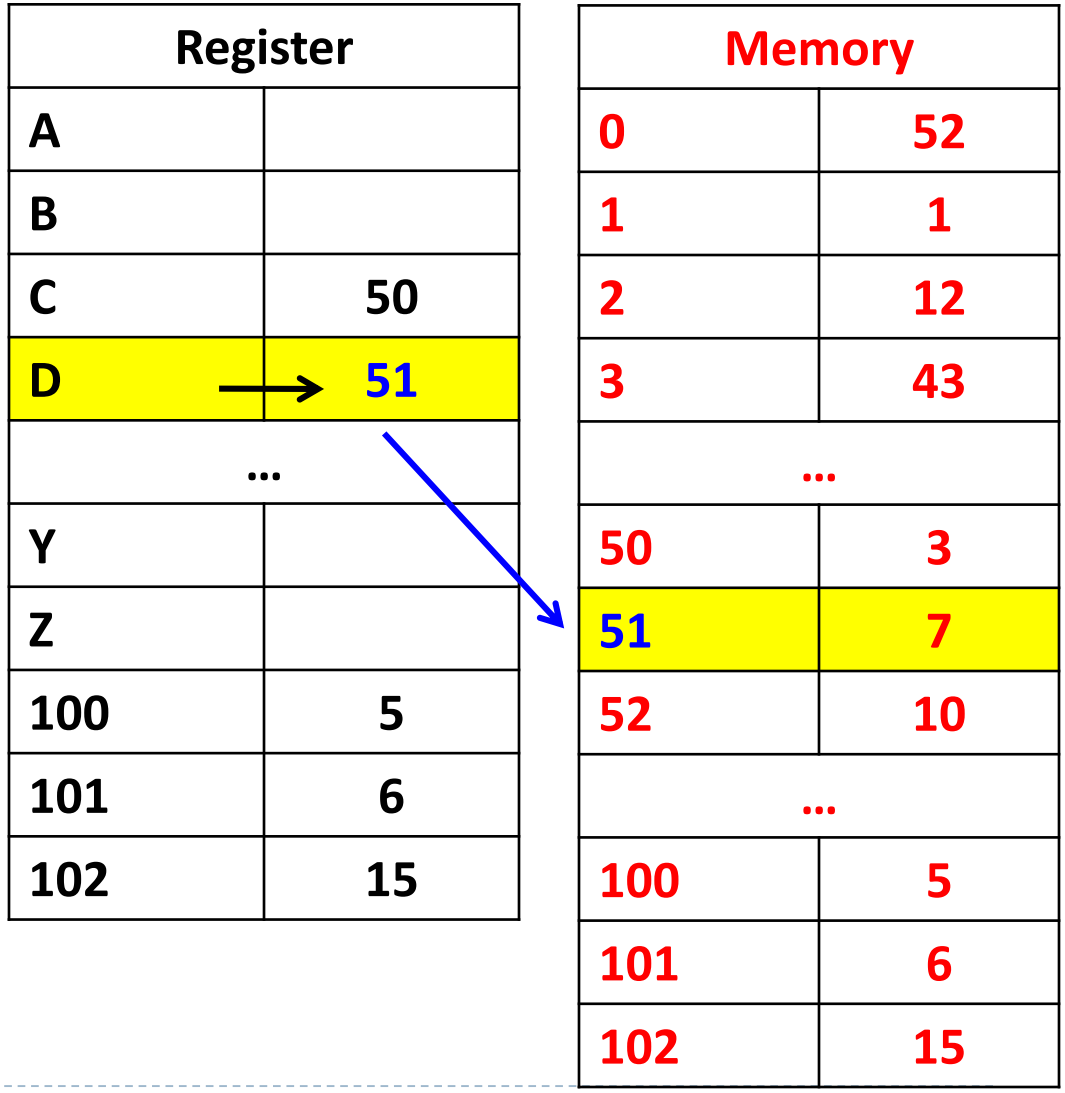
- LOAD (D)

  result

  ACC ← 7

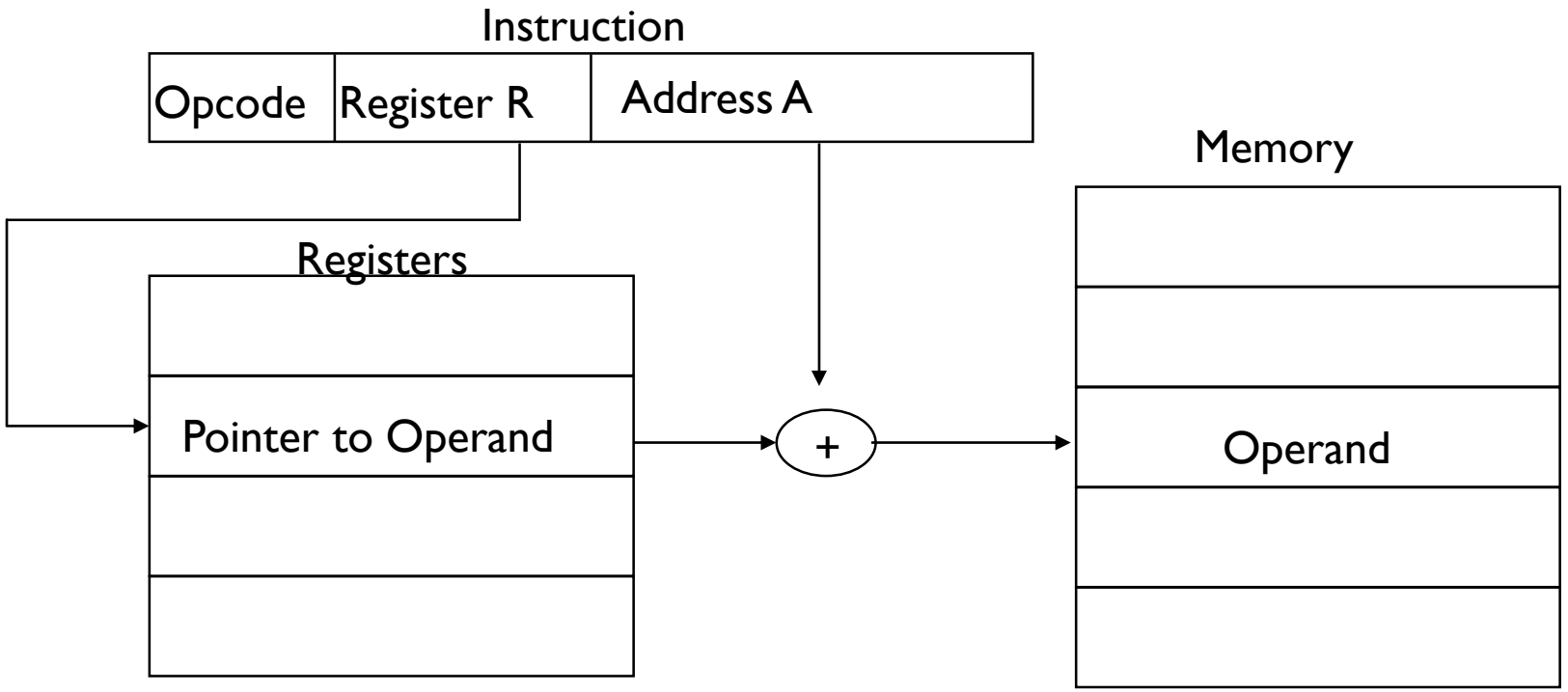
  addr. of register D contain addr of memory 51

  addr. Of memory 51 contain 7



Displacement Addressing

- EA = A + (R) ; Address field hold two values
  - A = base value
  - R = register that holds displacement



Displacement Addressing

- LOAD (C) + 50

  contain

  ACC ← 5

  addr. of register C contain addr. 50,

  addressee within +50

  50 + 50 = 100

  address 100 contain 5

Register	
A	
B	
C	50
D	51
…	
Y	
Z	
100	5
101	6
102	15

Memory	
0	52
1	1
2	12
3	43
…	
50	3
51	7
52	10
…	
100	5
101	6
102	15

Contoh

- calculate register Y ?

- Instruction

  LOAD (C)

  ADD 3

  SUB (0)

  MPY (C) + 50

  DIV #9

  STOR Y

Register	
A	
B	
C	50
D	51
…	
Y	
Z	
100	5
101	6
102	15

Memory	
0	52
1	1
2	12
3	43
…	
50	3
51	7
52	10
…	
100	5
101	6
102	15

Contoh

- calculate register Y ?

	ALGORITMIK	ISI ACC
LOAD C	AC ← 3	3
ADD 3	AC ← AC + 43	46
SUB (0)	AC ← AC – 10	36
MPY (C) + 50	AC ← AC x 5	180
DIV #9	AC ← AC / 9	20
STOR Y	**Y ← 20**	

Latihan

- Calculate Y

instruction

LOAD (C)

ADD 3

SUB (2)

ADD (0)

MPY (C) + 50

ADD #10

STOR Y

Register	
A	
B	
C	50
D	51
...	
Y	
Z	
100	54
101	66
102	151

Memory	
0	51
1	1
2	3
3	102
...	
50	2
51	75
52	107
...	
100	54
101	66
102	151

TERIMA KASIH