

## הרצאה 2 מודל המחשב

Edited by Tamer Salman 2008

1

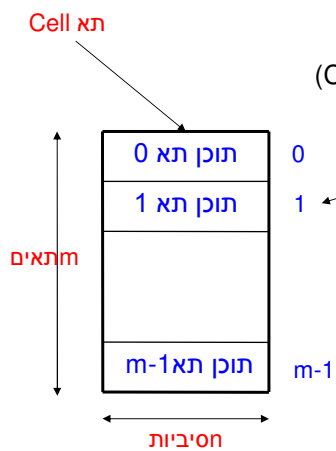
## חלקי המחשב

### • במחשב שני חלקים:

- זכרון (Memory)
- יחידת עיבוד מרכזית (Central Processing Unit)

### • הזכרון

- כתובת תיוצג ע"י A סיביות כאשר
- $2^A = m$
- כדי לאחסן כתובת בזכרון נדרש
- $n \geq A$



Edited by Tamer Salman 2008

2

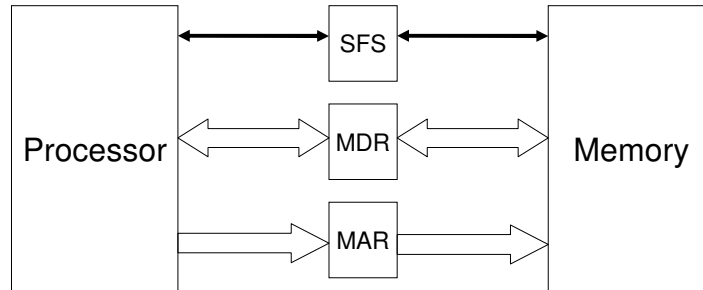
## תא הזכרון

- מה מכיל תא זכרון?
  - מספר (שלם, שבר, נקודה צפה).
  - תו (character) או מחרוזת (string).
  - כתובת של תא אחר.
  - פקודה.
- האם זכרון נתונים = זכרון תכנית?
  - כן.
  - משמעותו של תוכן תא זכרון נקבעת לפי השימוש באותו רגע.
- גודל תא זכרון:
  - משתנה ממחשב למחשב.
  - Burroughs B-1726 (1bit), IBM-370/PDP-11 (8bits),
  - Word (מילה) 12/16/18/32/48/60/64 bits.

## גישה לזכרון

- עקרון הגישה האקראית (Random Access Memory – RAM)
  - גישה ישירה לכל תא זכרון בזמן שווה לקריאה ולכתיבה.
  - אין חובת קריאה סדרתית.
- זכרון לקריאה בלבד (Read Only Memory – ROM)

## תקשורת מעבד - זכרון



### • אוגרי ממשק (Interface Registers)

- SFS: Store-Fetch Switch
- MDR: Memory Data Register
- MAR: Memory Address Register
- ניתן להניח שבאופן פונקציונלי עוברות מלים (words).

5

Edited by Tamer Salman 2008

## אוגרי הממשק

### • MAR

- מכיל את הכתובת ממנה / אליה רוצים לקרוא / לכתוב.
- גודלו  $\lceil \log_2[\text{גודל הזכרון}] \rceil$

### • MDR

- מכיל את תוכן התא הנקרא / הנכתב.
- גודלו כגודל תא זכרון.

### • SFS

- אוגר בקרה / מצב.
- גודלו 2 סיביות (2 bits)
- "00" כתיבה. "01" קריאה. "10" פקודה בוצעה. "11" תקלה.
- "00" ו- "01" יזמות ע"י המעבד. "10" ו- "11" יזמות ע"י הזכרון.

6

Edited by Tamer Salman 2008

## פרוטוקולי תקשורת מצד המעבד

- Read Protocol – CPU side

```
value FETCH (address a)
{
    /* wait until previous action is done */
    while (SFS != 2) ;

    MAR = a;
    SFS = 1;
    /* wait until action is done */
    while (SFS != 2);
    return MDR;
}
```

- Write Protocol – CPU side

```
void STORE (value d, address a)
{
    /* wait until previous action is done */
    while (SFS != 2) ;

    MAR = a;
    MDR = d;
    SFS = 0;
}
```

Edited by Tamer Salman 2008

7

## פרוטוקול תקשורת מצד הזכרון

- Read-Write Protocol – Memory side

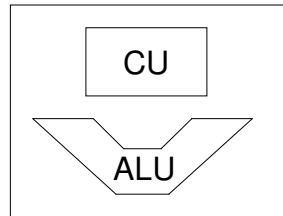
```
while (TRUE)          /* infinite loop */
{
    while (SFS == 2) ; /* wait until previous action is done */

    if (SFS == 0) {    /* write */
        a = MAR;
        MEM[a] = MDR;
        SFS = 2;
    }
    else {             /* read */
        a = MAR;
        MDR = MEM[a];
        SFS = 2;
    }
}
```

Edited by Tamer Salman 2008

8

## יחידת העיבוד המרכזית - CPU - Central Processing Unit



- יחידת הבקרה - **CU – Control Unit**  
– אחראית לקריאת הפקודות מהזכרון, פענחן וביצוען.
- היחידה האריתמטית-הלוגית **ALU – Arithmetical Logical Unit**  
– אחראית לביצוע פעולות אריתמטיות-לוגיות כל הנתונים.

Edited by Tamer Salman 2008

9

## סוגי פקודות

- אריתמטיות **ADD, SUBT**
- לוגיות **AND, NOT**
- העברה **MOVE, SWITCH**
- בקרה **JUMP, JSR**
- קלט/פלט **READ, WRITE**
- המסומנים בקו תחתון משתפים את ה-ALU.

Edited by Tamer Salman 2008

10

## מבנה פקודה

- **Op-code**
    - Operation code, מה לעשות.
  - **Operand/s**
    - אופרנדים, נתוני קלט.
  - **Result/s**
    - תוצאות.
  - **Next Instruction**
    - הפקודה הבאה.
- בפקודות אריתמטיות: שני אופרנדים, תוצאה אחת.

Edited by Tamer Salman 2008

11

## מבנה 1 + 3 כתובות

Op-code	Op #1	Op #2	Result	Next Inst.
---------	-------	-------	--------	------------

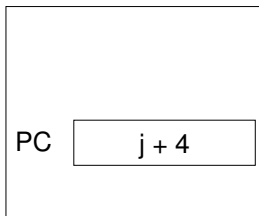
- **חסרון**
  - פקודה ארוכה מדי.
- לכן מבנה זה אינו נמצא בשימוש.
- משתמשים באוגר המצביע על הפקודה הבאה לביצוע  
PC – Program Counter –

Edited by Tamer Salman 2008

12

## מבנה 3 כתובות - דוגמא

• חישוב  $R = (X + Y - Z) / X$



⋮				
value of X				j
value of Y				j+1
value of Z				j+2
value of R				j+3
ADD	j	j+1	j+3	j+4
SUBT	j+3	j+2	j+3	j+5
DIV	j+3	j	j+3	j+6
⋮				

Edited by Tamer Salman 2008

13

## מבנה 2 כתובות

- אחד האופרנדים משמש לרישום התוצאה.
- יתרון: פקודה קצרה יחסית.
- חסרון: אחד האופרנדים נמחק.

ADD A, B  $\longleftrightarrow$  B := A + B

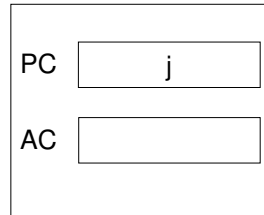
- דוגמאות למחשבים: PDP-11, IBM-370.

Edited by Tamer Salman 2008

14

## מבנה כתובת אחת

- אחד האופרנדים (מקום התוצאה) קבוע: הצובר Accumulator
- דרושות פקודות: LOAD address, STORE address.
- דוגמא



⋮		
LOAD	j + 6	j
ADD	j + 7	j+1
SUBT	j + 8	j+2
DIV	j + 6	j+3
STORE	j + 9	j+4
JUMP	j + 10	j+5
	value of X	j+6
	value of Y	j+7
	value of Z	j+8
	value of R	j+9
	⋮	

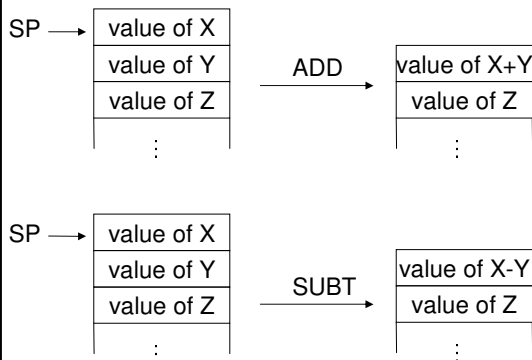
Edited by Tamer Salman 2008

15

## מבנה 0 כתובות (!)

- האופרנדים מצויים במחסנית (Stack).

דרוש:



- SP – Stack Pointer
- מצביע לראש המחסנית
- הפקודות: PUSH address
- POP address
- המכניסות ומוציאות
- איברים מראש המחסנית.

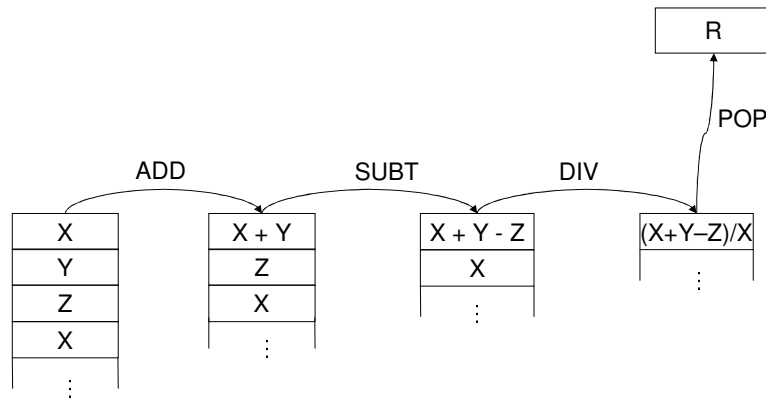
Edited by Tamer Salman 2008

16



## מבנה 0 כתובות - דוגמא

PUSH X  
PUSH Z  
PUSH Y  
PUSH X  
ADD  
SUBT  
DIV  
POP R



Edited by Tamer Salman 2008

17

## הערות

- בהכנסת נתונים בסדר נכון ניתן לבצע חישובים ללא משתני עזר.
- במספר מחשבים מבנה הפקודה ב- L1 – Microprogramming הינו 0-כתובות ובשפת המכונה L2 הינו 2-כתובות למשל.
- אורך הפקודה איננו בהכרח תא זכרון אחד.
- ה- PC יתקדם ע"פ אורך הפקודה הנקבע בעת פענוחה.

Edited by Tamer Salman 2008

18