

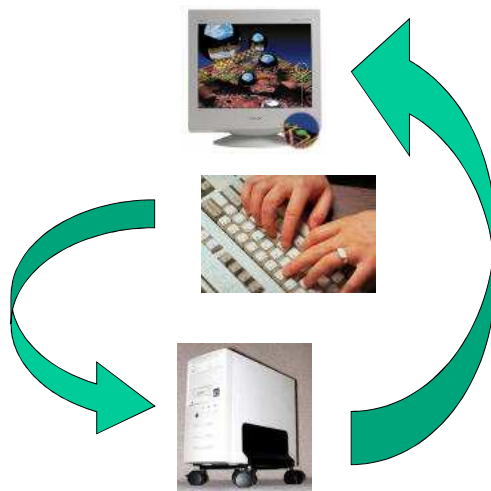
# תרגול 9

## קלט/פלט סינכרוני

מבוסס על שקפים מאת יאן ציטרין

1

### הסכמה של קלט / פלט



©את"ם - תרגול מס' 10

2

## ממשק בין המחשב לבין התקני קלט/פלט

- הממשק בין המחשב לבין התקני קלט/פלט ממומש ע"י ארבע מילים מיוחדות הנקראות "interface registers".
- לכל רגיסטר כזה יש כתובת קבועה בזיכרון וניתן לפנות אליו כאל חלק מהזיכרון.
- למרות שהמילים הללו מכונות רגיסטרים הם למעשה אינם אוגרים במובן רגיל.

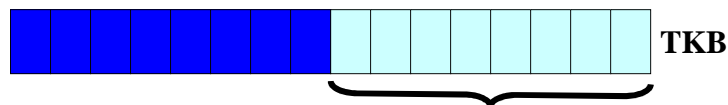
TKS	177560	המידע מגיע מחומרה בלבד
TKB	177562 (Read Only)	המידע מיועד למדפסת בלבד
TPS	177564	
TPB	177566 (Write Only)	

©את"ם - תרגול מס' 10

3

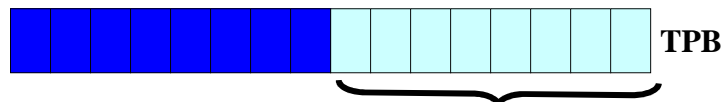
## TKB & TPB

התו מגיע מהמקלדת אל תוך **הבית הנמוך של ה-TKB**. התוכנית קוראת את התו ע"י פקודות רגילות שניגשות לזיכרון.



כאן מאוחסן קוד ה-ascii של התו שהוקלד

כדי להדפיס את התו יש להכניס אותו **לבית הנמוך של ה-TPB**. חומרת המדפסת מזהה את הגישה ל-TPB ומתחילה להדפיס את התו.



כאן מאוחסן קוד ה-ascii של התו שנשלח להדפסה

©את"ם - תרגול מס' 10

4

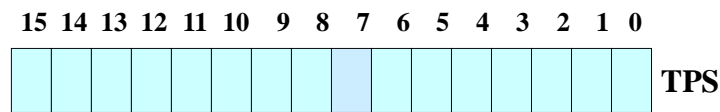
## TPS

### מדוע זקוקים לבקרה על המדפסת?

מכיוון שהמחשב הרבה יותר מהיר מהמדפסת, היא אינה יכולה להדפיס תווים בקצב שלו, אז צריך לחכות לאחר כל תו שנשלח, לסיום תהליך ההדפסה.

- דרוש אינדיקטור כלשהו שמציין את סיום ההדפסה הקודמת.

TPS הנו חוצץ שמספק מנגנון בקרה על המדפסת, והוא מכיל אינדיקטור כזה.



**ready**

הביט הזה הוא 1 כאשר המדפסת מוכנה לקבל תווים להדפסה

5

## TKS

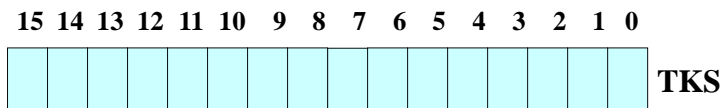
### ומדוע יש צורך בבקרה על המקלדת?

לא נרצה שייכנס אל תוך TKB תו חדש, לפני שהספקנו לקרוא את הקודם.

TKS הנו חוצץ שמספק מנגנון בקרה על המקלדת.

באופן דומה ל-TPS הוא מציין שהוקלד תו חדש ע"י המשתמש והתוכנית יכולה לקבל אותו.

קיימים 3 דגלים שאחראים על האינטראקציה



**RE –read enable**

חייב לעלות מ-0 ל-1 כדי שהמקלדת תוכל להכניס תו ל- TKB

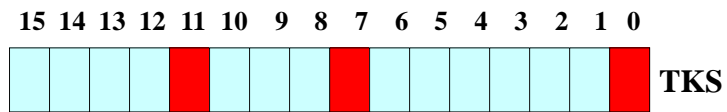
**busy**

מציין שהתו טרם הועבר מהמקלדת ל- TKB

**done**

מציין שקוד ה-ascii של התו שהוקלד מוכן ב- TKB

## TKS



כיצד התוכנית תבדוק האם יש תו מוכן עבודה?

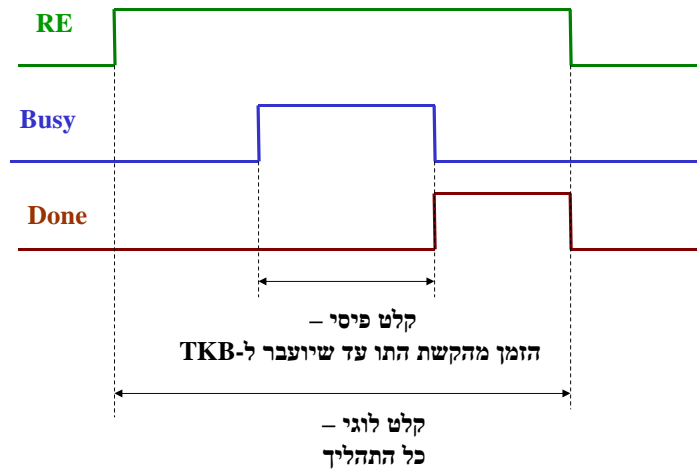
- הפתרון הנאיבי: ע"י פקודת bic (bit).
- ע"י התייחסות לTKS כשני בתיים במקום כמילה אחת.  
(בדיקת דגל N עבור הבית הנמוך)

## המנגנון של הקלט

Done	Busy	Read Enable (RE)	אחריות	פעולה
		↑	התוכנית	$RE \leftarrow 1$
	↑(*)		המקלדת	המשתמש מקיש תו
↑(*)	↓(*)		המקלדת	התו מוכן ב-TKB
↓(*)		↓(*)	התוכנית	קוראים את התו מ-TKB

(\*) – הפעולה מתבצעת אוטומטית ע"י החומרה

## המנגנון של הקלט (המשך)



©את"ם - תרגול מס' 10

9

## המנגנון של הפלט

1. ממתינים עד שדגל ה-Ready יעלה.
2. מכניסים תו ל-TPB. המדפסת מורידה את ה-Ready.



©את"ם - תרגול מס' 10

10

## דוגמה

```

tkb = 177560
tkb = 177562
tps = 177564
tpb = 177566
. = torg + 1000
main:  mov  #msg, r1
         mov  #6, r2
r_loop: mov  #1, @#tkb
busy_w: tstb  @#tkb
         bpl   busy_w
         movb @#tkb, (r1)+
         sob   r2, r_loop
    
```

```

mov  #msg, r1
mov  #6, r2
w_loop: tstb  @#tps
         bpl   w_loop
         movb (r1)+, @#tpb
         sob   r2, w_loop
         halt
msg:  .blkw  10
    
```

התוכנית מקבלת מהמשתמש 6 תווים,  
כותבת אותם לזיכרון (msg),  
לאחר מכן קוראת את אותם התווים  
ומדפיסה אותם.  
משתמשת במנגנון busy wait

©את"ם - תרגול מס' 10

11

## דוגמה נוספת (מהמבחן)

נתונה התוכנית שמבצעת את המשימה הבאה:

```

Loop:  read character c from keyboard
        if ( c == '/' ) then stop
        if ( c > '9' or c < '0' ) then Error
        translate character c to the corresponding digit i.
        print table[i]
        go to Loop
    
```

```

Error:  print '*'
        go to Loop
    
```

©את"ם - תרגול מס' 10

12

## דוגמה מהמבחן (המשך)

```

1.      tks      =      177560
2.      tkb      =      177562
3.      tps      =      177564
4.      tpb      =      177566

5. main: mov     #1, @#tks      ; RE ← 1
6. in:   tstb    @#tks      ; While ( there is no input)
7.      bpl     in           ;      wait for input      } busy
8.      movb    @#tkb, r1     ; R1 ← input character    } wait
9.      bic     mask1, r1     ; Clear irrelevant bits (high byte)
10.     cmpb    r1, end       ; if ( r1 == '/' )
11.     beq     sof          ;      go to sof
12.     cmpb    r1, #60       ; if ( r1 < '0' )
13.     blt     err          ;      go to err
14.     cmpb    r1, #71       ; if ( r1 > '9' )
15.     bgt     err          ;      go to err
16.     bic     mask2, r1     ; Conversion: '0' → 0, '1' → 1, etc.
17.     movb    table(r1), r2 ; R2 ← Character to be printed

```

©את"ם - תרגול מס' 10

13

## דוגמה מהמבחן (המשך)

```

18. out:  tstb    @#tps      ; while ( the printer is busy )
19.      bpl     out         ;      wait for printer to be ready
20.      movb    r2, @#tpb   ; print R2
21.      br      main       ; proceed to the next input

22. err:  movb    star, r2   ; R2 ← error message
23.      br      out        ; go to printing

24. sof:  halt             ; Stop the program

25. mask1: .word 177600
26. mask2: .word 177760
27. end:   .ascii</>
28. star:  .ascii<*>
29. table: .byte 12
30.       .ascii<abcdefghi>

```

©את"ם - תרגול מס' 10

14

דוגמה מהמבחן (המשך) -שאלות

מה יודפס עבור הקלט "192b" ?

**aib\***