

מדור בחינות ומערכת שעות

המחלקה להנדסת תוכנה

תאריך: 23/06/2022

שעה: 09:00-12:00

ארכיטקטורת מחשבים II

מבחן - מועד א'

פרופ' שלמה גרינברג, ד"ר נינה ביימל

תשפ"ב – הנדסת תוכנה - סמסטר ב

חומר עזר: מחשבון.

יש לענות על כל השאלות.
המבחן מכיל 5 שאלות. משקלן של השאלות שונה והניקוד מצויין עבור כל סעיף בנפרד.
השאלון מכיל 11 עמודים (כולל עמוד זה).

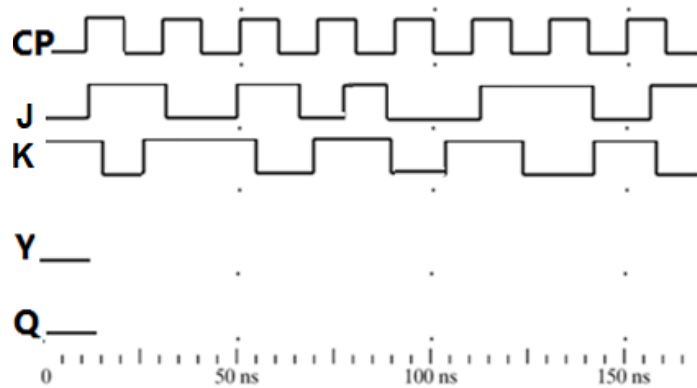
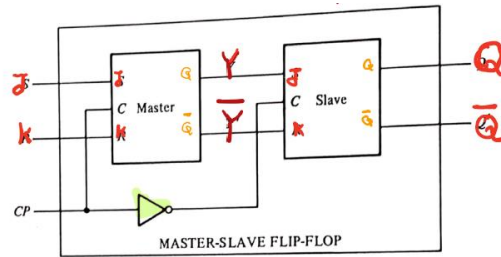
הערות:

1. יש לכתוב את התשובות בכתב יד ברור ומסודר בדפים ממוספרים בלבד.
2. בכל השאלות יש להציג את דרך הפתרון. תשובה סופית בלבד לא תתקבל.

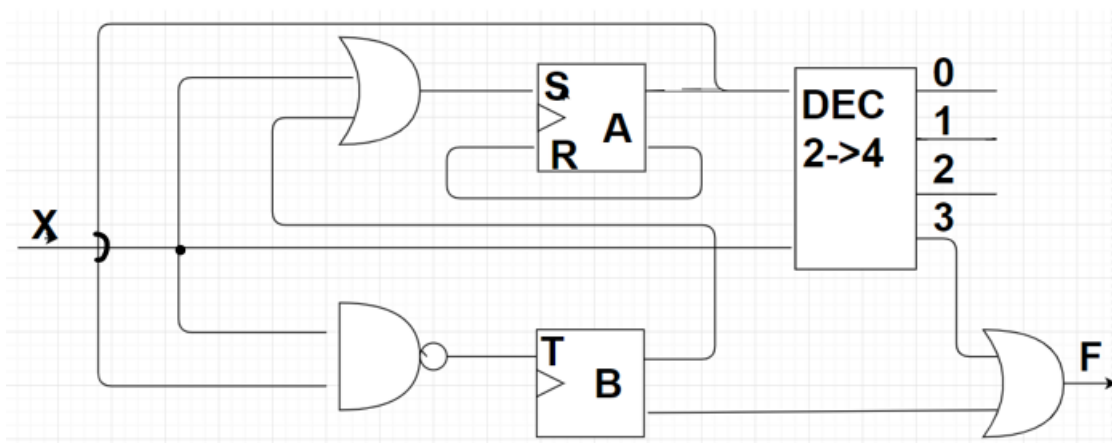
בהצלחה !

שאלה 1: (25 נק') - אלמנטי זיכרון וניתוח מעגלים סדרתיים.

1.1 (5 נק') השלימו את דיאגרמת הזמנים עבור דלגלג מסוג **JK Master-Slave**. נתונים ערכי התחלה כדלקמן: $J=0, K=1, Y=0, Q=0$. יש להניח שכאשר $CP=1$ עדכון ערך המוצא יבוצע פעם אחת בלבד.



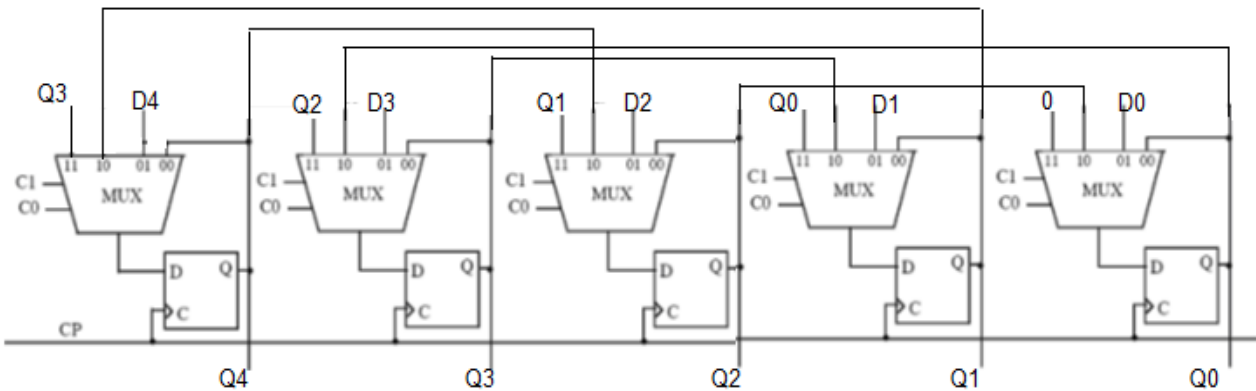
1.2 (20 נק') נתון תרשים של מעגל סדרתי המכיל שני דלגלים: דלגלג A מסוג SR ודלגלג B מסוג T. למעגל יש כניסת קלט X ומוצא יחיד F.



- 1.2.1 (5 נק') רישמו ביטוי לפונקציות הכניסה לדלגלים (S_A, R_A, T_A) ולפונקציית המוצא F.
- 1.2.2 (5 נק') רישמו את טבלת המצבים המתארת את התנהגות המערכת.
- 1.2.3 (5 נק') שרטטו את דיאגרמת המצבים.
- 1.2.4 (5 נק') מצאו את משוואות המצב Q_A, Q_B . פשטו באמצעות מפת קרנו.

שאלה 2: (25 נק') תכנון אוגרים ומונים

2.1 (5 נק') האיור להלן מתאר את חיבורי הבקרה של אוגר בן 5 סיביות. יש לציין בטבלה את הפונקציה המבוצעת באוגר כתלות בקווי הבקרה: C0 ו-C1.



פונקציה		C1	C0
		0	0
		0	1
		1	0
		1	1

2.2 (20 נק') יש לתכנן מונה סינכרוני up/down עם כניסת בקרה x הקובעת את כיוון המנייה.

עבור $x=0$ המונה סופר **UP**: 2, 3, 5, 6, 7 (משמאל לימין).

עבור $x=1$ המונה סופר **Down**: 7, 6, 5, 3, 2 (משמאל לימין).

הנח השמת מצבים בינרית.

יש לתכנן את המערכת כך שכאשר היא נקלעת למצב לא מוגדר המערכת תעבור למצב 2.

2.2.1 (5 נק') שרטט את דיאגרמת המצבים

2.2.2 (5 נק') תאר את טבלת המצבים של המערכת.

2.2.3 (5 נק') מצא את משוואות הכניסה לדלגלים.

2.2.4 (5 נק') ממש את המונה באמצעות דלגלים מטיפוס **JK** ושערים לוגיים כנדרש.

שאלה 3: (15 נק') מיפוי זיכרון

במערכת מחשב מבוססת מעבד עם 16 קווי כתובת הזיכרון החיצוני מיושם על ידי הרכיבים הבאים:

- ארבעה רכיבי ePROM בגודל 8 KB כל אחד. הרכיבים ממופים ברצף החל מכתובת 0.
- שני רכיבי RAM בגודל 8 KB כל אחד. הרכיבים ממופים החל מכתובת 9000H.

3.1 (2 נק') רשום את מרחב הזכרון של כל אחד מרכיבי ה-ePROM (בקידוד HEX)

3.2 (3 נק') רשום את מרחב הזכרון של כל אחד מרכיבי ה-RAM (בקידוד HEX)

3.3 (10 נק') נתונים 2 מפענחים מסוג 3:8 ושני שערים לוגיים לצורך מיפוי ויצירת CS לרכיבי הזיכרון.

ציין את קווי הכניסה לכל מפענח. שרטט את החיבור של מוצא המפענחים ליצירת 6 קווי ה-CS לרכיבי הזיכרון.

שאלה 4: (10 נקודות)

נתון מבנה של Datapath ומבנה הפקודות של יחידת הבקרה (ראה דפי עזר).

ה-Datapath מכיל 16 אוגרים (השדות: DA, AA, BA הם בני 4 ביט כל אחד).

קידוד פקודות ה-ALU מפורט בטבלת ה-FS.

שדה ה-Constant הוא בן 8 ביטים ותומך בערכים שלמים חיוביים ושיליים.

אותות הבקרה הם: DA, AA, BA, FS, MB, MD, RW, MW, Constant.

נתונה תכנית בשפת RTL: (Register Transfer Language).

$$R4 \leftarrow R2 - R5$$

$$R4 \leftarrow R3 + 20$$

$$R7 \leftarrow M[R3]$$

$$R2 \leftarrow R2 \oplus R3$$

4.1 (5 נק') ציינו בטבלה את מצב אותות הבקרה הנדרש לביצוע של כל פקודה. (שורה עבור כל פקודה).
ראו דוגמא עבור הפקודה $R2 \leftarrow R1 \text{ AND } R2$.

4.2 (5 נק') הסבירו את הפעולה המתבצעת ואת מסלול העברת הנתונים עבור הפקודה $R7 \leftarrow M[R3]$.

Pseudo-code	Control Signals								
	DA	AA	BA	FS	MB	MD	WR	MW	Const.

$R2 \leftarrow R1 \text{ AND } R2$ 2 2 1 1000 0 0 1 0 x

שאלה 5: (25 נקודות) שפת אסמבלי

5.1 (10 נק') כתבו תוכנית בשפת ASM86 המדפיסה על המסך משולש ישר זווית שווה שוקיים (מיושר לשמאל). התוכנית קוראת לפרוצדורה בשם `show_triangle` המקבלת דרך המחסנית את ערך האוגר `AX` המייצג את אורך הבסיס של המשולש. הנח ש `AX=10h` כך שהתוכנית מדפיסה משולש שבסיסו שווה ל 16.

הערה: יש לכתוב תוכנית אחת הכוללת את הקריאה לפרוצדורה ואת הפרוצדורה עצמה. ממש לולאה מקוננת באמצעות שימוש במחסנית ופקודת `loop` בלבד.

5.2 (10 נק') כתבו קטע קוד בשפת ASM86 שמבצע היפוך של הספרות של שני מספרים (בייצוג הקסדצימלי) השמורים באוגרים `AH,AL` ומחשב את החיסור של שני המספרים (לאחר ההיפוך). תוצאת החיסור נשמרת באוגר `CX`.

הנח שערכם של האוגרים הוא: `ah = 58h al = 43h`.
 כלומר התוכנית אמורה לבצע את החישוב הבא: `cx = 85h-34h`.
הערה: יש להשתמש בפקודות **הזזה** במימוש הקוד.

5.3 (5 נק') להלן נתון קטע של `data segment` הכולל משתנה אנונימי (ללא שם).
 הדגם בקצרה כיצד ניתן לקרוא את ערך המשתנה האנונימי (30h) ולהציבו באוגר `dl`.

```
A db 2,3,12
B dw ?, ?
  db 30h
C db 12h
```

בהצלחה !

ארכיטקטורת מחשבים II דפי עזר

טבלאות מאפיינות:

JKFF

J	K	$Q(t+1)$	
0	0	$Q(t)$	No Change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

SRFF

S	R	$Q(t+1)$	
0	0	$Q(t)$	No Change
0	1	0	Reset
1	0	1	Set
1	1	?	Undef.

DFF

D	$Q(t+1)$	
0	0	Reset
1	1	Set

TFF

T	$Q(t+1)$	
0	$Q(t)$	No Change
1	$Q'(t)$	Complement

טבלאות עירור:

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

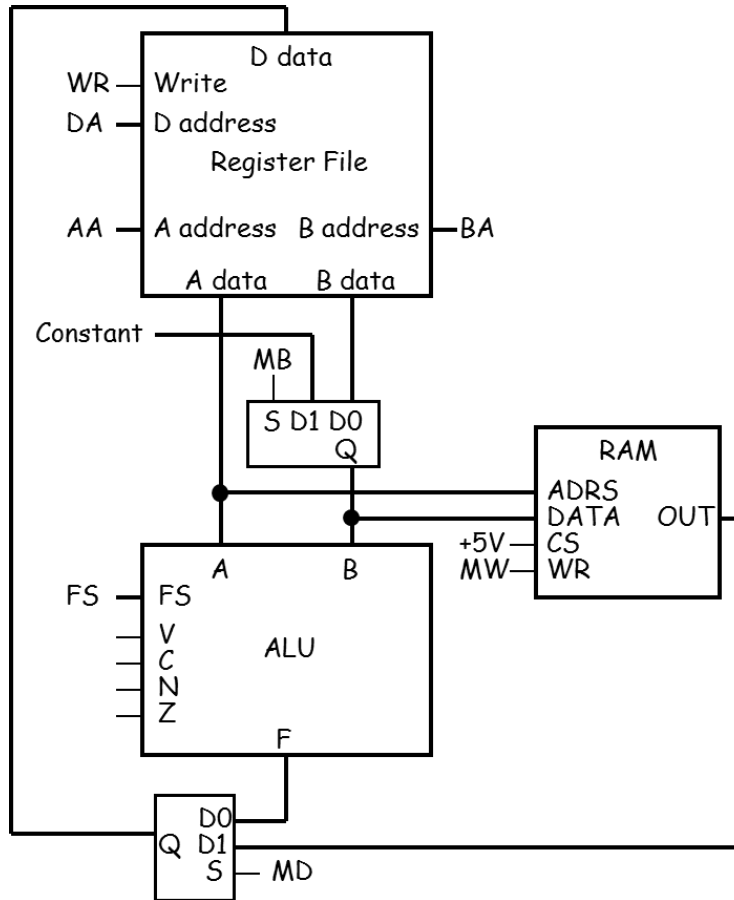
Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

גודל זכרון:

$$1K = 2^{10}; 1M = 2^{20}; 1G = 2^{30}$$

מבנה מעבד: Datapath ופעולות ALU



פעולות ALU: ביטים 9-12 ב Opcode

FS	Operation
00000	$F = A$
00001	$F = A + 1$
00010	$F = A + B$
00011	$F = A + B + 1$
00100	$F = A + B'$
00101	$F = A + B' + 1$
00110	$F = A - 1$
00111	$F = A$
01000	$F = A \wedge B$ (AND)
01010	$F = A \vee B$ (OR)
01100	$F = A \oplus B$
01110	$F = A'$
10000	$F = B$
10100	$F = sr\ B$ (shift right)
11000	$F = sl\ B$ (shift left)

מבנה מעבד: קידוד פקודות

קידוד הפקודות בשפת מכונה:

15	9	8	6	5	3	2	0
Opcode	Destination Register (DR)		Source Register A (SA)		Source Register B (SB)		
Opcode	Destination Register (DR)		Source Register A (SA)		Operand (OP)		
Opcode	Address Bits 5-3 (AD)		Source Register A (SA)		Address Bits 2-0 (AD)		

Instruction type	Opcode bits		
	15	14	13
Register-format ALU operation	0	0	0
Register-format shift operation	0	0	1
Memory write (from registers)	0	1	0
Memory read (to registers)	0	1	1
Immediate ALU operation	1	0	0
Immediate shift operation	1	0	1
Conditional branch	1	1	0
Jump	1	1	1

פעולות ALU: ביטים 9-12 ב- Opcode

פקודות BRANCH - ביטים 9-11 עפ"י הטבלה הבאה:

Condition	BC
If carry set	000
If negative	001
If overflow	010
If zero	011
If carry clear	100
If positive	101
If no overflow	110
If non-zero	111

אסמבלי 8086

פקודות קפיצה והשוואה:

טבלת קפיצות לפי דגלים

מס'	דגל	פקודה
1	Z=1	JE/JZ
2	Z=0	JNE/JNZ
3	C=1	JC
4	C=0	JNC
5	S=1	JS
6	S=0	JNS
7	O=1	JO
8	O=0	JNO

טבלת קפיצות

	>	>=	<	<=	==	!=
Unsigned (לא מסומן)	JA	JAE	JB	JBE	JE/JZ	JNE/JNZ
Signed (מסומן)	JG	JGE	JL	JLE	JE/JZ	JNE/JNZ

מבנה תכנית מלאה כולל מחסנית ו- Data Segment:

```

data segment
; here you type the variables
data ends
stack segment stack
db 100h dup (0)
stack ends
code segment
assume ds:data, cs:code, ss:stack
start: mov ax,data
      mov ds,ax

      ; your code here!

      mov ah,4ch
      int 21h

code ends
end start
    
```

פסיקות:

פסיקת יציאה למערכת ההפעלה

הפסיקה נדרשת בסוף כל תכנית שנרצה להריץ כיוון שללא הבקשה הזו, נקבל שגיאה מתוכנית שיוצאת באופן לא חוקי.

```

mov ah, 4CH
int 21H
    
```

פסיקה להצגת הודעה על הצג

יש לרשום משתנה המכיל הודעה, ההודעה תהיה בין גרשיים ובסופה, בתוך הגרשיים, יירשם סימן דולר.

```
Msg db 'Good morning $'
```

```
mov dx, offset Msg
mov ah, 9
int 21h
```

```
data segment
MSG db 'Hello!!!$'
data ends
code segment
assume cs: code, ds: data
main: mov ax, data
      mov ds, ax

      mov dx, offset MSG
      mov ah, 9
      int 21h

      mov ah, 4CH
      int 21h

code ends
end main
```

זהירות: פסיקה זו "מקלקלת" את תוכנו של האוגר AL.

פסיקה לקליטת מקש מהמקלדת

התו יוצג על הצג:

```
mov ah, 1
int 21h
```

התו לא יוצג על הצג:

```
mov ah, 7
int 21h
```

קוד ASCII של המקש יירשם באוגר AL.

פסיקת להצגת תו בודד על הצג

אפשרות א':

יש לאחסן ב-DL את קוד ה-ASCII של התו שרוצים להקריין.

```
mov dl, 35h ← קוד האסקי של הספרה 5 הינו 35
mov ah, 2
int 21h
```

זהירות: פסיקה זו "מקלקלת" את תוכנו של האוגר AL.

אפשרות ב':

נרשום את התו שברצוננו להציג בין גרשיים:

```
mov dl, '5'
mov ah, 2
int 21h
```

תבנית פרוצדורה:

```
my_proc_name proc
; code for proc at the end of the code segment only!
my_proc_name endp
```

קריאה לפרוצדורה:

```
call my_proc_name
```