

אלקטרוניקה ומחשבים ה'

מגמת הנדסת אלקטרוניקה ומחשבים

(כיתה י"ד)

הוראות לנבחן

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים, ובהם שמונה שאלות. יש להשיב על **ארבע שאלות בלבד**, שאלה אחת לפחות מכל פרק.

לכל שאלה – 25 נקודות. סך-הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: מחשבון.

ד. הוראות מיוחדות:

- ענה על מספר השאלות הנדרש בשאלון. המעריך יקרא ויעריך את מספר השאלות הנדרש בלבד, לפי סדר כתיבתן במחברתך, ולא יתייחס לתשובות נוספות.
- התחל כל תשובה לשאלה חדשה בעמוד חדש.
- רשום את כל תשובותיך **אך ורק בעט**.
- הקפד לנסח את תשובותיך כהלכה ולסרטט את תרשימיך בהירות.
- כתוב את תשובותיך בכתב-יד ברור, כדי לאפשר הערכה נאותה של תשובותיך.
- אם לדעתך חסרים נתונים הדרושים לפתרון שאלה, אתה רשאי להוסיף אותם, בתנאי שתנמק מדוע הוספת אותם.
- בכתיבת פתרונות חישוביים, קבלת מֶרֶב הנקודות מותנית בהשלמת כל המהלכים שלהלן, בסדר שבו הם רשומים:
 - * רישום הנוסחה המתאימה.
 - * הצבה של כל הערכים ביחידות המתאימות.
 - * חישוב (אפשר באמצעות מחשבון).
 - * רישום התוצאה המתקבלת, יחד עם יחידות המידה המתאימות.
 - * ליווי הפתרון החישובי בהסבר קצר.
- לנוחותך, לשאלון זה מצורף מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכל להיעזר בו בעת הצורך.

בשאלון זה 10 עמודים ו-21 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר,
אך מכוונות הן לנבחנות והן לנבחנים.

בהצלחה!

◀ המשך מעבר לדף

השאלות

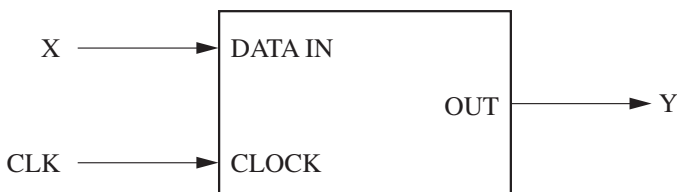
ענה על ארבע מבין השאלות 1-8. עליך לענות על שאלה אחת לפחות מכל פרק.

פרק ראשון: אלקטרוניקה ספרתית ב'

ענה על שאלה אחת לפחות מבין השאלות 1-4 (לכל שאלה - 25 נקודות).

שאלה 1

באיור לשאלה 1 נתונה מערכת עקיבה סינכרונית בעלת מבוא X ומוצא Y, שתמומש באמצעות דלגלי JK-FF ושערים לוגיים.



איור לשאלה 1

המערכת תפיק '1' במוצא Y בכל פעם שיתקבל רצף הנתונים '1' '1' '1' במבוא X (עבור כל '1' נוסף שיתקבל במבוא X לאחר הרצף הזה - המערכת תוסיף להפיק '1' במוצא Y. רק כאשר יתקבל '0' במבוא X לאחר הרצף הזה - המערכת תפיק '0' במוצא Y).

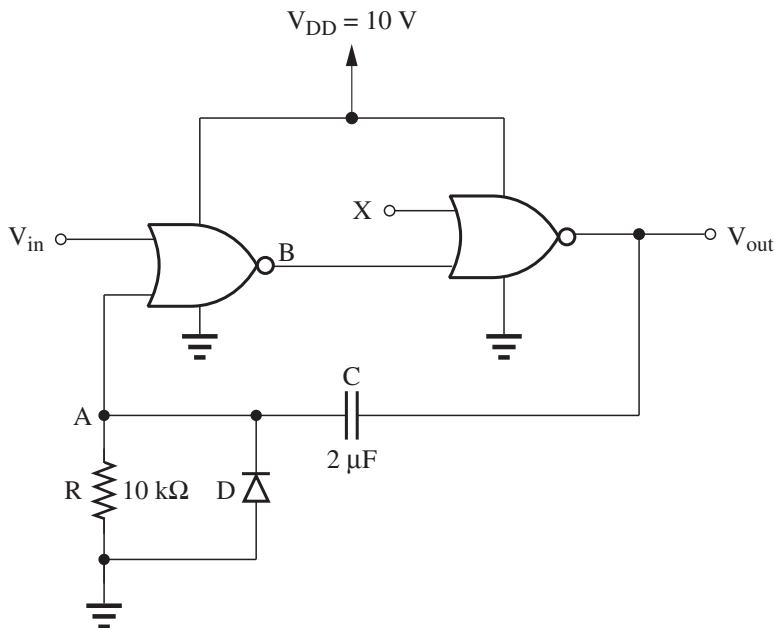
בדוגמה שלהלן, הזנת הנתונים למבוא X מתבצעת משמאל לימין:

X הנתון במבוא	0	1	0	1	1	0	1	1	1	0	1	1	1	1	0
Y הערך במוצא	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0

- סרטט דיאגרמת בועות, המתארת את הנדרש ממערכת העקיבה.
- רשום את טבלת המצבים ואת טבלת המעברים של המערכת הזו.
- רשום את הפונקציות של המבואות J ו-K בכל אחד מן הדלגלים, ואת פונקציית המוצא של המערכת.
- ממש את המערכת באמצעות דלגלים מסוג JK ושערים לוגיים.

שאלה 2

באיור לשאלה 2 נתון המעגל החשמלי של רב־רטט חד־יציב. הוא ממומש באמצעות שערים ממשפחת CMOS, המקיימים: $V_{TH} = \frac{1}{2} \cdot V_{DD}$.

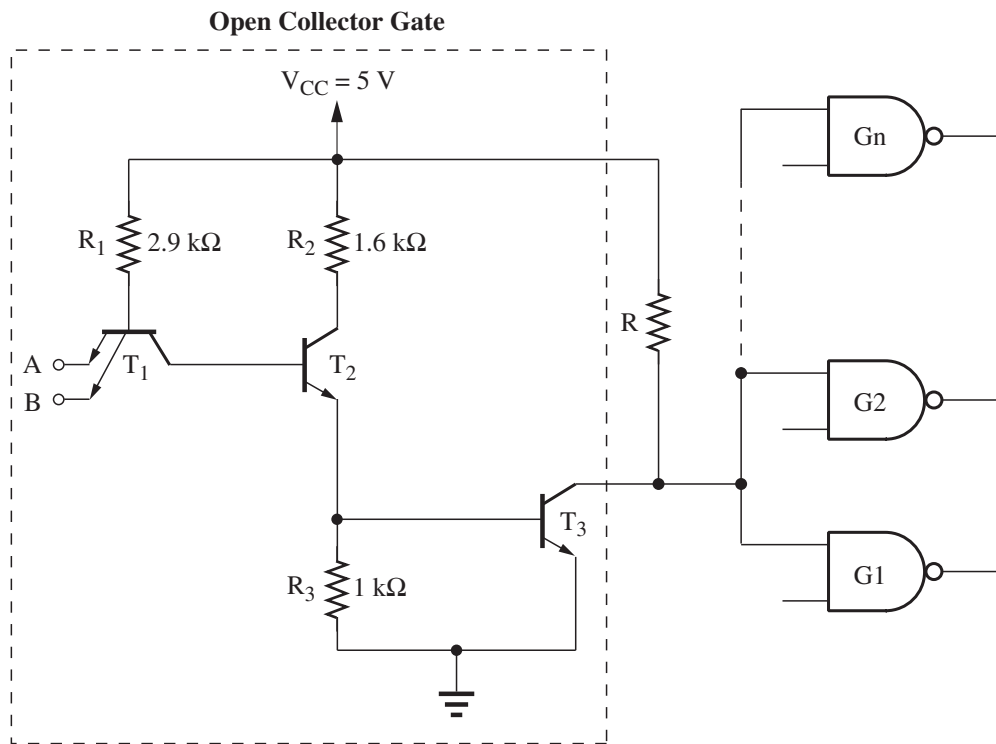


איור לשאלה 2

- א. מהי הרמה הלוגית שצריך לספק להדק X, כדי שהרב־רטט יפעל כחד־יציב?
- ב. מצא את המתחים V_{in} , V_A , V_B ו־ V_{out} במצב היציב.
- ג. 1. מספקים למבוא־המעגל דופק דירבון חיובי צר. סרטט, זה מתחת לזה בהתאמה, את המתחים V_{in} , V_A , V_B ו־ V_{out} . ציין בסרטוטך את הערך המרבי ואת הערך המזערי של כל אחד מן המתחים הללו.
2. חשב את רוחב הדופק המתקבל במוצא המעגל.
- ד. מהו תפקיד הדיודה D במעגל הזה?

שאלה 3

באיור לשאלה 3 נתון המעגל החשמלי של שער לוגי מסוג קולט פתוח (Open Collector Gate) ממשפחת TTL. שער זה מחובר למבואות של n שערים זהים.



איור לשאלה 3

כל הטרנזיסטורים זהים, ונתונים הם:

$$V_{CE_{SAT}} = 0.3 \text{ V} ; V_{BC_{ON}} = V_{BE_{ON}} = 0.7 \text{ V}$$

א. העתק את הטבלה שלהלן למחברתך, ורשום באיזה מצב נמצא כל אחד מן הטרנזיסטורים T_2 ו- T_3 ומהו הערך הלוגי של מוצא שער ה-TTL בכל אחד מן המקרים.

A	B	T_2	T_3	מוצא שער ה-TTL
'0'	'0'			
'0'	'1'			
'1'	'0'			
'1'	'1'			

ב. חשב את הזרם במבוא השער G1 :

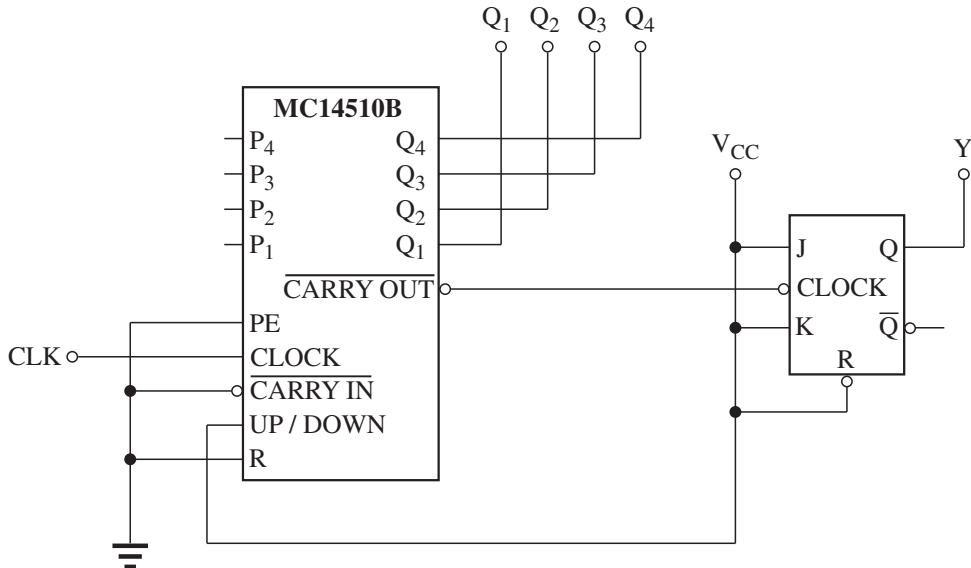
1. כאשר מוצא שער ה-TTL נמצא במצב '0'.

2. כאשר מוצא שער ה-TTL נמצא במצב '1'.

ג. חשב את זרם הבסיס של הטרנזיסטור T_3 כאשר $A = B = '1'$.

שאלה 4

באיור לשאלה 4 נתון מעגל חשמלי הכולל את המונה MC14510B ודלגלג מסוג JK. בנספח לשאלה 4 נתונים דפיי-המפרט של המונה. תדר הדפקים של האות CLK הוא 100 kHz. המצב ההתחלתי של מוצאי המונה הוא: $Q_1 = Q_2 = Q_3 = Q_4 = '0'$. המצב ההתחלתי של מוצא הדלגלג הוא $Q = '1'$. הדלגלג מגיב לירידת דופק השעון במבוא CLOCK שלו.



איור לשאלה 4

א. רשום את טבלת-האמת של דלגלג מסוג JK, וציין את תפקידו במעגל הזה.

ב. הסבר את ייעודו של המוצא $\overline{\text{CARRY OUT}}$ במונה, וציין באיזה מקרה הוא יהיה במצב '0'.

ג. חשב את תדר התנודות של האות במוצא המעגל Y.

פרק שני: שפה עילית

ענה על שאלה אחת לפחות מבין השאלות 5-8 (לכל שאלה - 25 נקודות).

שאלה 5

להלן תכנית הכתובה בשפת C :

```
1.  #include <stdio.h>
2.  void func(short *a)
3.  {
4.      short c, x;
5.      for (c = 15; c >= 0; c--)
6.      {
7.          x = *a >> c;
8.          if (x & 1)
9.              printf("1");
10.         else
11.             printf("0");
12.     }
13. }
14. void main()
15. {
16.     short n;
17.     scanf("%d", &n);
18.     func(&n);
19.     printf("\n");
20. }
```

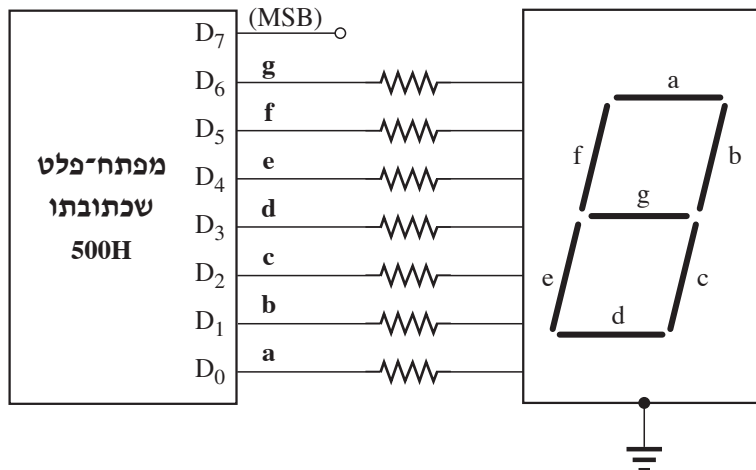
א. הסבר את ההוראות שבשורות 4, 7, 8 ו-18.

ב. הסבר מה מבצעת התכנית.

ג. קלט התכנית הוא $(54)_{10}$. מה יהיה פלט התכנית?

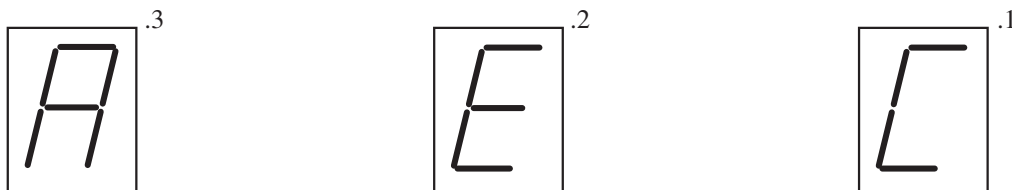
שאלה 6

למפתח-פלט שכתובתו 500H חיברו תצוגת שבעה מקטעים (7-seg) בחיבור קתודה משותפת (CC), כמתואר באיור לשאלה 6.



איור לשאלה 6

א. מהו הקוד שיש לשלוח למפתח-הפלט כדי להציג בתצוגת שבעת-המקטעים כל אחת מן האותיות שלהלן:

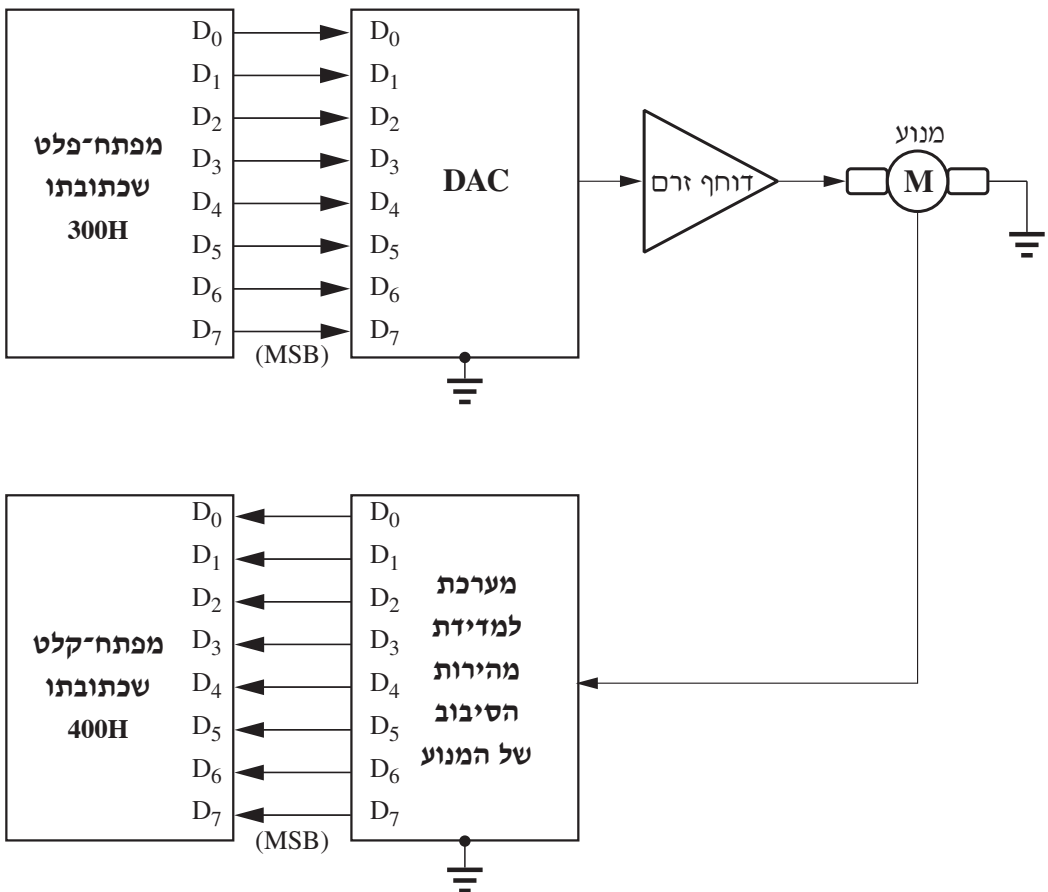


ב. נתון קובץ בשם chars.txt המכיל טקסט באנגלית. כתוב תכנית בשפת C, המבצעת את הפעולות האלה:

- קוראת מהקובץ את אותיות הטקסט בזו אחר זו.
- מציגה בתצוגת שבעת-המקטעים כל אחת מן האותיות A, C, ו-E למשך שנייה אחת, בכל פעם שהיא מופיעה בקובץ.
- סופרת את כמות האותיות A, את כמות האותיות C ואת כמות האותיות E בקובץ.
- מציגה בתצוגת שבעת-המקטעים למשך עשר שניות את האות המופיעה בטקסט יותר מהאחרות, מבין האותיות A, C, ו-E.

שאלה 7

נתונה מערכת הכוללת DAC בעל 8 סיביות, המחובר למפתח־פלט שכתובתו 300H. מוצא ה-DAC מחובר לדוחף זרם שמסובב מנוע, כמתואר באיור לשאלה 7. המנוע מחובר למערכת המודדת את מהירות הסיבוב שלו. המערכת למדידת מהירות הסיבוב מחוברת למפתח־קלט שכתובתו 400H. מהירות הסיבוב של המנוע (ביחידות סל"ד) שווה בערכה לערך המתקבל במפתח־הקלט. נתון שכאשר ה-DAC מקבל ערך של 240, מהירות המנוע היא 200 סיבובים לדקה, וכאשר ה-DAC מקבל ערך של 30, מהירות המנוע היא 25 סיבובים לדקה.



איור לשאלה 7

להלן תכנית הכתובה בשפת C:

```
1.  #include <stdio.h>
2.  short_stdcall Inp32(short PortAddress);
3.  void_stdcall Out32(short PortAddress,short data);
4.  void main()
5.  {
6.      byte dataIN;
7.      byte DACval=120;
8.      Out32(0x300,DACval);
9.      Sleep(1000);
10.     dataIN=Inp32(0x400);
11.     printf("Speed=%d\n",dataIN);
12.     while (1)
13.     {
14.         if(dataIN==100)
15.             printf("ok\n");
16.         else if(dataIN>100)
17.             while(dataIN>100)
18.             {
19.                 DACval--;
20.                 Out32(0x300,DACval);
21.                 Sleep(1000);
22.                 dataIN=Inp32(0x400);
23.             }
24.         else
25.             while(dataIN<100)
26.             {
27.                 DACval++;
28.                 Out32(0x300,DACval);
29.                 Sleep(1000);
30.                 dataIN=Inp32(0x400);
31.             }
32.     }
33. }
```



- א. הסבר את ההוראות שבשורות 8, 10, 11, 17 ו-19.
- ב. הסבר מה מבצע קטע התכנית שבין השורות $17 \div 23$.
- ג. מהי מטרת התכנית? נמק את תשובתך.
- ד. אילו שינויים היית מבצע בתכנית, כדי שהמנוע יסתובב במהירות של 50 סיבובים לדקה?

שאלה 8

כתוב תכנית בשפת C, שתבצע את הפעולות האלה:

1. תקלוט מהמקלדת עשרה מספרים שלמים וחיוביים.
2. תגדיר מערך חד-ממדי בשם R, שגודלו עשרה תאים.
3. תציב את עשרת המספרים במערך R, כך שכל המספרים הזוגיים ימוקמו בזה אחר זה החל מתא R[0] ומעלה, וכל המספרים האי-זוגיים ימוקמו בזה אחר זה החל מתא R[9] ומטה.
4. תציג על צג המחשב את כמות המספרים הזוגיים.

לדוגמה:

עבור עשרת המספרים 6, 1, 17, 2, 3, 1, 5, 4, 19 ו-8, הנקראים מימין לשמאל, ייראה המערך R בסיום ביצוע התכנית כך:

כתובת התא	R[0]	R[1]	R[2]	R[3]	R[4]	R[5]	R[6]	R[7]	R[8]	R[9]
תוכן התא	6	2	4	8	19	5	1	3	17	1

בהצלחה!

BCD Up/Down Counter

The MC14510B synchronous up/down BCD counter is constructed with MOS P-channel and N-channel enhancement mode devices in a monolithic structure. The counter consists of type D flip-flop stages with a gating structure to provide type T flip-flop capability.

This counter can be preset by applying the desired value in BCD to the Preset inputs (P1, P2, P3, P4) and then bringing the Preset Enable (PE) high. The direction of counting is controlled by applying a high (for up counting) or a low (for down counting) to the UP/DOWN input. The state of the counter changes on the positive transition of the clock input.

Cascading can be accomplished by connecting the Carry Out to the Carry In of the next stage while clocking each counter in parallel. The outputs (Q1, Q2, Q3, Q4) can be reset to a low state by applying a high to the Reset (R) pin.

This CMOS counter finds primary use in up/down and difference counting. Other applications include: (1) Frequency synthesizer applications where low power dissipation and/or high noise immunity is desired, (2) Analog-to-digital and digital-to-analog conversions, and (3) Magnitude and sign generation.

- Diode Protection on All Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Internally Synchronous for High Speed
- Logic Edge-Clocked Design — Count Occurs on Positive Going Edge of Clock
- Asynchronous Preset Enable Operation
- Capable of Driving Two Low-power TTL Loads or One Low-power Schottky TTL Load Over the Rated Temperature Range.

MAXIMUM RATINGS* (Voltages Referenced to V_{SS})

Symbol	Parameter	Value	Unit
V _{DD}	DC Supply Voltage	- 0.5 to + 18.0	V
V _{in} , V _{out}	Input or Output Voltage (DC or Transient)	- 0.5 to V _{DD} + 0.5	V
I _{in} , I _{out}	Input or Output Current (DC or Transient), per Pin	± 10	mA
P _D	Power Dissipation, per Package†	500	mW
T _{stg}	Storage Temperature	- 65 to + 150	°C
T _L	Lead Temperature (8-Second Soldering)	260	°C

* Maximum Ratings are those values beyond which damage to the may occur.

† Temperature Derating:

Plastic "P and D/DW" Packages: - 7.0 mW/°C From 65°C To 125°C

Ceramic "L" Packages: - 12 mW/°C From 100°C To 125°C

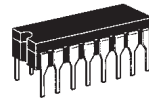
TRUTH TABLE

Carry In	Up/Down	Preset Enable	Reset	Clock	Action
1	X	0	0	X	No Count
0	1	0	0	↗	Count Up
0	0	0	0	↘	Count Down
X	X	1	0	X	Preset
X	X	X	1	X	Reset

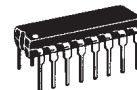
X = Don't Care

NOTE: When counting up, the Carry Out signal is normally high, and is low only when Q1 and Q4 are high and Carry In is low. When counting down, Carry Out is low only when Q1 through Q4 and Carry In are low.

MC14510B



L SUFFIX
CERAMIC
CASE 620



P SUFFIX
PLASTIC
CASE 648



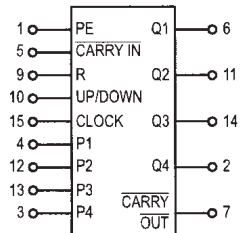
D SUFFIX
SOIC
CASE 751B

ORDERING INFORMATION

MC14XXXBCP Plastic
MC14XXXBCL Ceramic
MC14XXXBD SOIC

T_A = - 55° to 125°C for all packages.

BLOCK DIAGRAM



This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range V_{SS} ≤ (V_{in} or V_{out}) ≤ V_{DD}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Unused outputs must be left open.

ELECTRICAL CHARACTERISTICS (Voltages Referenced to V_{SS})

Characteristic	Symbol	V_{DD} Vdc	- 55°C		25°C			125°C		Unit
			Min	Max	Min	Typ #	Max	Min	Max	
Output Voltage "0" Level $V_{in} = V_{DD}$ or 0	V_{OL}	5.0	—	0.05	—	0	0.05	—	0.05	Vdc
		10	—	0.05	—	0	0.05	—	0.05	
		15	—	0.05	—	0	0.05	—	0.05	
"1" Level $V_{in} = 0$ or V_{DD}	V_{OH}	5.0	4.95	—	4.95	5.0	—	4.95	—	Vdc
		10	9.95	—	9.95	10	—	9.95	—	
		15	14.95	—	14.95	15	—	14.95	—	
Input Voltage "0" Level ($V_O = 4.5$ or 0.5 Vdc) ($V_O = 9.0$ or 1.0 Vdc) ($V_O = 13.5$ or 1.5 Vdc)	V_{IL}	5.0	—	1.5	—	2.25	1.5	—	1.5	Vdc
		10	—	3.0	—	4.50	3.0	—	3.0	
		15	—	4.0	—	6.75	4.0	—	4.0	
	"1" Level ($V_O = 0.5$ or 4.5 Vdc) ($V_O = 1.0$ or 9.0 Vdc) ($V_O = 1.5$ or 13.5 Vdc)	5.0	3.5	—	3.5	2.75	—	3.5	—	Vdc
		10	7.0	—	7.0	5.50	—	7.0	—	
		15	11	—	11	8.25	—	11	—	
Output Drive Current ($V_{OH} = 2.5$ Vdc) ($V_{OH} = 4.6$ Vdc) ($V_{OH} = 9.5$ Vdc) ($V_{OH} = 13.5$ Vdc)	Source	5.0	- 3.0	—	- 2.4	- 4.2	—	- 1.7	—	mAdc
		5.0	- 0.64	—	- 0.51	- 0.88	—	- 0.36	—	
		10	- 1.6	—	- 1.3	- 2.25	—	- 0.9	—	
		15	- 4.2	—	- 3.4	- 8.8	—	- 2.4	—	
	Sink	5.0	0.64	—	0.51	0.88	—	0.36	—	mAdc
		10	1.6	—	1.3	2.25	—	0.9	—	
Input Current	I_{in}	15	—	± 0.1	—	± 0.00001	± 0.1	—	± 1.0	μ Adc
Input Capacitance ($V_{in} = 0$)	C_{in}	—	—	—	—	5.0	7.5	—	—	pF
Quiescent Current (Per Package)	I_{DD}	5.0	—	5.0	—	0.005	5.0	—	150	μ Adc
		10	—	10	—	0.010	10	—	300	
		15	—	20	—	0.015	20	—	600	
Total Supply Current**† (Dynamic plus Quiescent, Per Package) ($C_L = 50$ pF on all outputs, all buffers switching)	I_T	5.0 10 15	$I_T = (0.58 \mu A/kHz) f + I_{DD}$ $I_T = (1.20 \mu A/kHz) f + I_{DD}$ $I_T = (1.70 \mu A/kHz) f + I_{DD}$							μ Adc

#Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

** The formulas given are for the typical characteristics only at 25°C.

†To calculate total supply current at loads other than 50 pF:

$$I_T(C_L) = I_T(50 \text{ pF}) + (C_L - 50) Vfk$$

where: I_T is in μA (per package), C_L in pF, $V = (V_{DD} - V_{SS})$ in volts, f in kHz is input frequency, and $k = 0.001$.

PIN ASSIGNMENT

PE	1	16	V_{DD}
Q4	2	15	C
P4	3	14	Q3
P1	4	13	P3
CARRY IN	5	12	P2
Q1	6	11	Q2
CARRY OUT	7	10	U/D
V_{SS}	8	9	R

PIN DESCRIPTIONS

INPUTS

P1, P2, P3, P4, Preset Inputs (Pins 4, 12, 13, 3) — Data on these inputs is loaded into the counter when PE is taken high.

Carry In, (Pin 5) — Active-low input used when cascading stages. Usually connected to Carry Out of the previous stage. While high, clock is inhibited.

Clock, (Pin 15) — BCD data is incremented or decremented, depending on the direction of count, on the positive transition of this signal.

OUTPUTS

Q1, Q2, Q3, Q4, BCD outputs (Pins 6, 11, 14, 2) — BCD data is present on these outputs with Q1 corresponding to the least significant bit.

Carry Out, (Pin 7) — Used when cascading stages, this pin is usually connected to Carry In of the next stage. This

synchronous output is active low and may also be used to indicate terminal count.

CONTROLS

PE, Preset Enable (Pin 1) — Asynchronously loads data on the Preset Inputs. This pin is active high and will inhibit the clock when high.

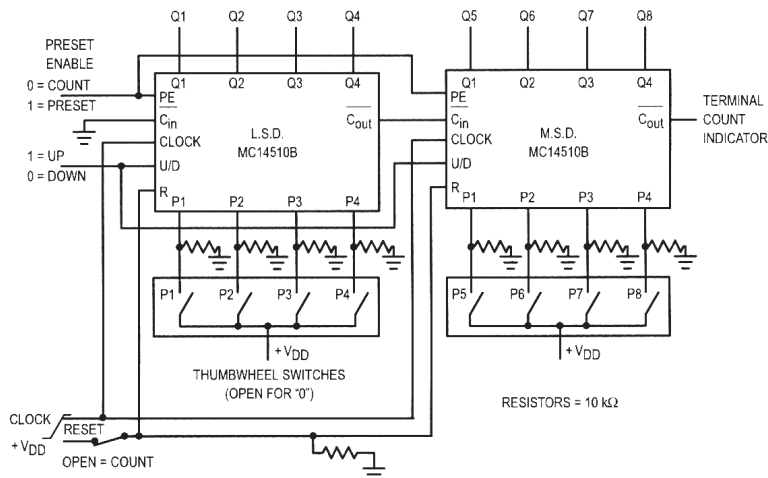
R, Reset, (Pin 9) — Asynchronously resets the Q outputs to a low state. This pin is active high and will inhibit the clock when high.

Up/Down, (Pin 10) — Controls the direction of count: high for up count, low for down count.

SUPPLY PINS

V_{SS}, Negative Supply Voltage, (Pin 8) — This pin is usually connected to ground.

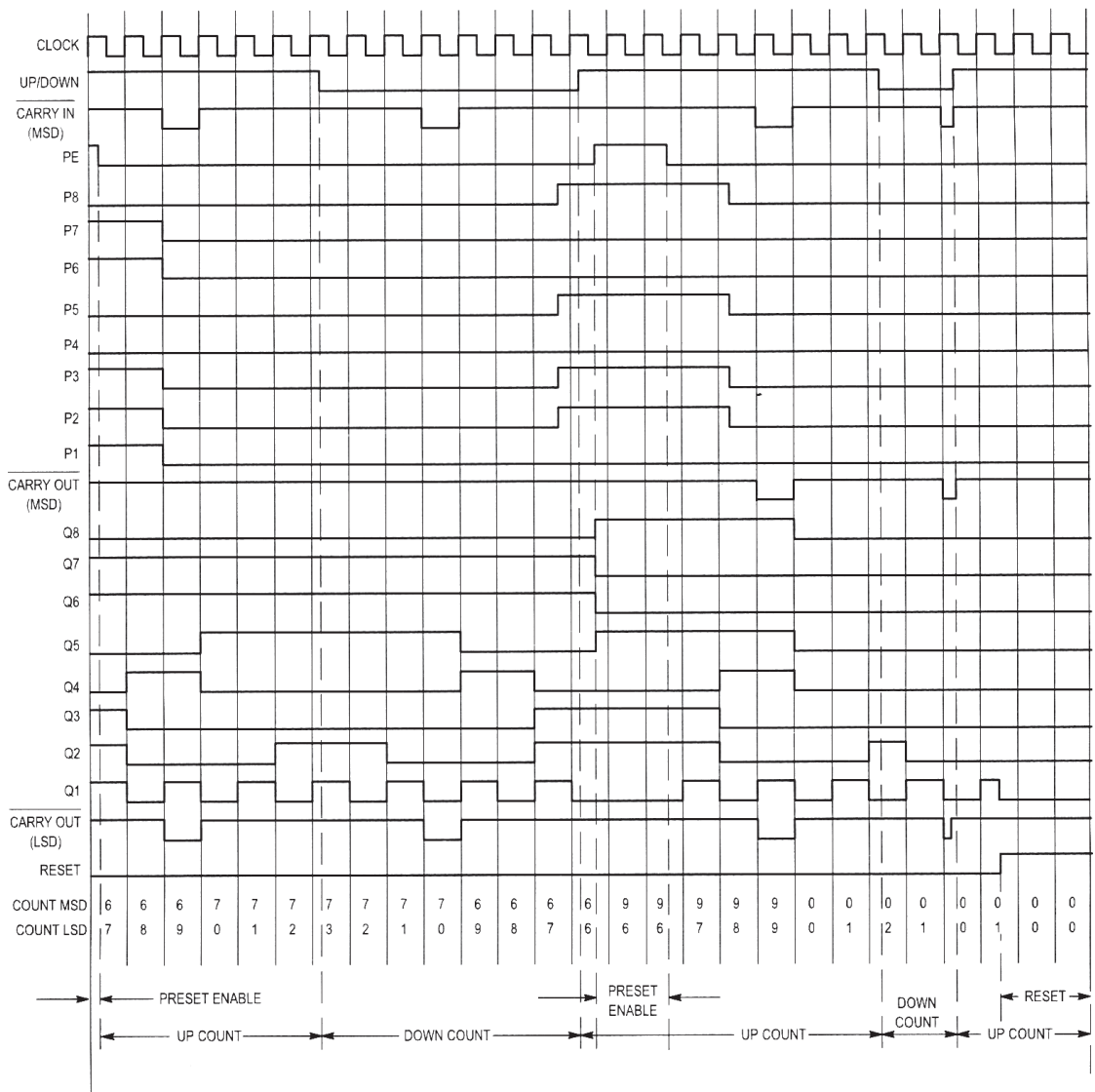
V_{DD}, Positive Supply Voltage, (Pin 16) — This pin is connected to a positive supply voltage ranging from 3.0 Vdc to 18.0 Vdc.



Note: The Least Significant Digit (L.S.D.) counts from a preset value once Preset Enable (PE) goes low. The Most Significant Digit (M.S.D.) does not change while $\overline{C_{in}}$ is high. When the count of the L.S.D. reaches 0 (count down mode) or reaches 9 (count up mode), $\overline{C_{out}}$ goes low for one complete clock cycle, thus allowing the next counter to decrement/increment one count. The L.S.D. now counts through another cycle (10 clock pulses) and the above cycle is repeated.

Figure 3. Presettable Cascaded 8-Bit Up/Down Counter

**TIMING DIAGRAM FOR THE PRESETTABLE
CASCADED 8-BIT UP/DOWN COUNTER**



אין להעביר נוסחאון זה
מנבחן אחד למשנהו!

מקור מציגות נכחן

נוסחאון באלקטרוניקה ספרתית ב' לכיתה י"ד

(4 עמודים)

משוואות הדפקים היסודית

- מתח מוצא - $V(t)$ [V]
 מתח סופי (עבור $t \rightarrow \infty$) - V_∞ [V]
 מתח התחלתי - V_{0+} [V]
 זמן - t [sec]
 קבוע הזמן - τ [sec]
 זרם מוצא - $I(t)$ [A]
 זרם סופי (עבור $t \rightarrow \infty$) - I_∞ [A]
 זרם התחלתי - I_{0+} [A]
 התנגדות שקולה ש"רואה" הרכיב
 ההיגבי, מחושבת לפי תבנית - R_{eq} [Ω]

$$V(t) = V_\infty - (V_\infty - V_{0+}) \cdot e^{-\frac{t}{\tau}}$$

$$I(t) = I_\infty - (I_\infty - I_{0+}) e^{-\frac{t}{\tau}}$$

$$\tau = R_{eq} \cdot C \quad ; \quad \tau = \frac{L}{R_{eq}}$$

$$t = \tau \cdot \ln \frac{V_\infty - V_{0+}}{V_\infty - V(t)} = \tau \cdot \ln \frac{I_\infty - I_{0+}}{I_\infty - I(t)}$$

תחום הרוויה בטרנזיסטור דו-נושאי

- מתח רוויה בין קולט לפולט - V_{CEs} [V]
 זרם בסיס - I_B [A]
 זרם קולט - I_C [A]
 הגבר זרם - β

$$V_{CE} = V_{CEs}$$

$$\beta \cdot I_B > I_C$$

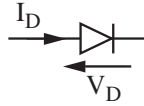
סף רוויה של התחום הפעיל בטרנזיסטור דו-נושאי

$$V_{CE} = V_{CEs}$$

$$\beta \cdot I_B = I_C$$

דיודת צומת

סימול:



א. דיודה אידיאלית:

ממתח קדמי - $V_D = 0$ (קָצֵר)

ממתח אחורני - $I_D = 0$ (נִתְק)

ב. קירוב באמצעות V_γ :

ממתח קדמי - $V_D = V_\gamma$

ממתח אחורני - $I_D = 0$ ($V_D < V_\gamma$)

ג. קירוב באמצעות V_γ ו- R_f :

ממתח קדמי - $V_D = I_D \cdot R_f + V_\gamma$ ($V_D > V_\gamma$)

ממתח אחורני - $I_D = 0$ ($V_D < V_\gamma$)

טעינה לינארית

טעינת קבל בזרם קבוע:

מתח הקבל	-	V_C	[V]
זרם הקבל	-	I_C	[A]
קיבול	-	C	[F]
זמן	-	t	[sec]

$$V_C = \frac{I_C}{C} \cdot t + V_C(0)$$
$$\Delta V_C = \frac{I_C}{C} \cdot \Delta t$$

טעינת סליל במתח קבוע:

זרם הסליל	-	I_L	[A]
מתח הסליל	-	V_L	[V]
השראות	-	L	[H]

$$I_L = \frac{V_L}{L} \cdot t + I_L(0)$$
$$\Delta I_L = \frac{V_L}{L} \cdot \Delta t$$

טבלת מצבים של JKFF

CLK	J	K	Q
1, 0	$\emptyset$	$\emptyset$	N.C
$\downarrow$	0	0	N.C
$\downarrow$	0	1	0
$\downarrow$	1	0	1
$\downarrow$	1	1	$\bar{Q}_{n-1}$ (שינוי מצב)

טבלת מצבים של SRFF (סינכרוני)

CLK	S	R	Q
0, 1	$\emptyset$	$\emptyset$	N.C
$\downarrow$	0	0	N.C
$\downarrow$	0	1	0
$\downarrow$	1	0	1
$\downarrow$	1	1	מצב אסור

טבלת מצבים של TFF

CLK	T	Q_n
0, 1	$\emptyset$	N.C
	0	N.C
	1	$\bar{Q}_{n-1}$ (שינוי מצב)

טבלת מצבים של DFF

CLK	D	Q
1, 0	$\emptyset$	N.C
	0	0
	1	1

טבלת עירור

PS			NS		JKFF		SRFF		TFF	DFF
q			→	Q	J	K	S	R	T	D
0	→	0		0	0	ϕ	0	ϕ	0	0
0	→	1		1	1	ϕ	1	0	1	1
1	→	0		0	ϕ	1	0	1	1	0
1	→	1		1	ϕ	0	ϕ	0	0	1

בהצלחה!

אין להעביר את הנוסחאון
לנבחן אחר

נוסחאון בשפת C לכיתה י"ד

(11 עמודים)

נוסחאון זה מתאים למהדר Microsoft Visual C++ 2010 Express Edition.
חלקים ממנו מתאימים גם למהדרים אחרים.

Data Types (טיפוסי נתונים)

Name	Description	תאור	Size*	Range*
char	Character or small integer	תו בודד	1 byte	-128 to 127
unsigned char	Unsigned small integer	תו בודד ללא סימן	1 byte	0 to 255
short	Short Integer	מספר שלם קטן	2 bytes	-32768 to 32767
unsigned short	Unsigned short integer	מספר שלם קטן ללא סימן	2 bytes	0 to 65535
int	Integer	מספר שלם	4 bytes	-2147483648 to 2147483647
unsigned int	Unsigned integer	מספר שלם ללא סימן	4 bytes	0 to 4294967295
float	Floating point number	מספר ממשי	4 bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double floating point number	מספר ממשי ארוך	8 bytes	+/- 1.7e +/- 308 (~15 digits)

*הערכים של עמודות אלו תלויים במבנה המחשב שבו נעשה הידור התוכנית.

```
char a;
float number;
int b, c;
unsigned short NewNumber;
```

המשך בעמוד 2

Preprocessor directives (הנחיות לקדם-מהדר)

Description	Syntax	Example
macro definitions	#define identifier replacement	#define ArrSize 100

identifier – מזהה ; replacement – תחליף

Operators (אופרטורים)

Description	תאור	Operator
Assignment	השמה	=

Initialization of variables (אתחול משתנים)

```
int d = 0;  
  
d=75;           // decimal number  
  
d=0x4b;         // hexadecimal number
```

Arithmetic operators (אופרטורים חשבוניים)

Description	תאור	Operator
Addition	חיבור	+
subtraction	חיסור	-
multiplication	כפל	*
division	חילוק	/
modulo	שארית	%

Relational and equality operators (אופרטורים להשוואה ויחסים)

Description	תאור	Operator
Equal to	שווה	==
Not equal to	שונה	!=
Greater than	גדול מ.	>
Less than	קטן מ.	<
Greater than or equal to	גדול שווה מ.	>=
Less than or equal to	קטן שווה מ.	<=

Logical operators (אופרטורים לוגיים בין ביטויים)

Description	תאור	Operator
NOT	היפוך	!
AND	וגם	&&
OR	או	

Bitwise Operators (אופרטורים על סיביות)

Description	תאור	ASM equivalent	Operator
AND	וגם	AND	&
Inclusive OR	או כולל	OR	
Exclusive OR	או מוציא	XOR	^
Bit inversion	היפוך	NOT	~
Shift Left	הזזה שמאלה	SHL	<<
Shift Right	הזזה ימינה	SHR	>>

Basic Input/Output (קלט/פלט בסיסי)

Description	Syntax	Example
Standard Output	<code>int putchar (int character);</code>	<code>int a='G' ;</code> <code>putchar(a) ;</code>
Standard Input	<code>int getchar (void);</code>	<code>int c;</code> <code>c=getchar() ;</code>

Formatted Input/Output (פלט לפי תבנית)

Description	Syntax	Example
Formatted output	<code>printf(format[,arg1,arg2,...]);</code>	<code>int num=10;</code> <code>printf("num=%d\n",num) ;</code>
Formatted Input	<code>scanf(format [,arg1,arg2,...]);</code>	<code>int num;</code> <code>scanf ("%d" ,&num) ;</code>

Specifier	Operator	פלט	Example
%c	Character	תו בודד	a
%d	Signed decimal integer	עשרוני שלם	133
%e	Scientific notation	עשרוני כולל נקודה וחזקה של 10	3.012e+4
%f	Decimal floating point	עשרוני כולל נקודה עשרונית	123.45
%s	String of characters	מחרוזת תווים	Hello
%x	Unsigned hexadecimal integer	הקסדצימלי ללא סימן	3fe

Conditional Structures (מבני בקרה – משפטי תנאי)

Description	Syntax	Example
if	<pre>if (condition) { statements ; }</pre>	<pre>if (d == 100) { printf("d is 100"); }</pre>
if .. else	<pre>scanf (condition) statement1; else statement2 ;</pre>	<pre>if (d == 100) printf("d is 100"); else printf("d is not 100");</pre>
if .. else if .. else	<pre>if (condition) statement1 ; else if (condition) statement2 ; else statement3 ;</pre>	<pre>if (d > 0) printf("d is positive"); else if (d < 0) printf("d is negative"); else printf("d is 0");</pre>

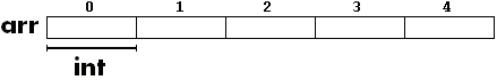
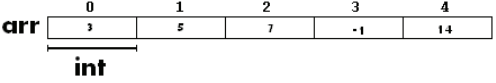
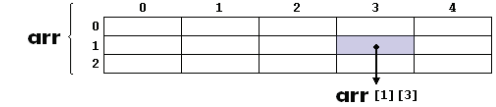
condition – תנאי ; הצהרה – statement

Iteration Structures (מבני בקרה - לולאות)

Description	Syntax	Example
while loop	<pre>while (expression) { statements ; }</pre>	<pre>while (n>0) { printf(" %d \n",n); n--; }</pre>
do-while loop	<pre>do { statements ; } while (condition);</pre>	<pre>do { printf("Enter 0 to end: "); scanf("%d",&n); }while (n != 0);</pre>
for loop	<pre>for (initialization; condition; increase) { statements ; }</pre>	<pre>for (i=0; i<10; i++) { printf(" %d \n",i); }</pre>

תנאי - condition ; הצהרה - statement

Arrays (מערכים)

Description	Syntax	Example
הגדרת מערך חד מימדי 	type name [elements];	int arr[5];
אתחול והצבת ערכים במערך 	type name [elements] = {value1,..valueN};	int arr[5] = {3,5,7,-1,14};
הגדרת מערך דו מימדי 	type name [elements, elements];	int arr[3][5];

elements – פרטים ; value – ערך

Structure of a program (מבנה כללי של תוכנית)

```
#include <stdio.h>

void main(void)
{

}
```

Hardware Input/Output (קלט/פלט בסיסי מחומרה)

Description	Syntax	Example
Hardware Output	Out32(hardware address, value);	Out32 (0x378, 0xAA) ;
Hardware Input	Inp32(hardware address);	int dataIN; dataIN=Inp32 (0x379) ;

hardware address – כתובת חומרה ; value – ערך

```
#include <stdio.h>

short _stdcall Inp32(short PortAddress);

void _stdcall Out32(short PortAddress, short data);

void main(void)
{
    int dataIN;

    Out32 (0x378, 0xAA) ;

    dataIN=Inp32 (0x379) ;
}
```

Sleep Function (פונקציית השהיה)

Description	Syntax	Example
Suspends the execution of the current thread until the time-out interval elapses	void Sleep (dword dwMilliseconds);	Sleep(2000);

*For windows 32-bit registry a DWORD is a 4-bytes unsigned int.

```
#include <windows.h>

void main(void)
{
    Sleep(2000) ;
}
```

המשך בעמוד 9

Functions (פונקציות)

Description	Syntax	Example
Functions with no argument	<pre>void name (void) { statements ; }</pre>	<pre>#include <stdio.h> void PrintHello(void) { printf("Hello"); } void main(void) { PrintHello(); }</pre>
Functions with no type	<pre>void name (parameter1, parameter2, ...) { statements ; }</pre>	<pre>#include <stdio.h> void multiplication(int a,int b) { int c; c=a*b; printf("%d*d=%d",a,b,c); } void main(void) { multiplication(2,8); }</pre>
Functions with type and argument	<pre>type name (parameter1, parameter2, ...) { statements ; }</pre>	<pre>#include <stdio.h> int multiplication(int a,int b) { int c; c=a*b; return c; } void main(void) { int r; r = multiplication(2,8); printf("%d",r); }</pre>

הצהרה – statement ; ערך המועבר לפונקציה – parameter
המשך בעמוד 10

Pointers (מצביעים)

Description	תאור	Operator
Reference operator	אופרטור הכתובת (כתובתו של)	&
Dereference operator	אופרטור המצביע (הערך המוצבע על-ידי)	*

דוגמה:

```
int a;
int *p_a;
p_a = &a;
*p_a = 10;
```

Data Structures (מבנים)

Description	Syntax	Example
הגדרת מבנה	<pre>struct structure_name { member_type1 c_name1; member_type2 member_name2; ... };</pre>	<pre>struct point { int x; int y; };</pre>
אתחול במבנה	structure_name object_name;	point MyPoint;
הצבת ערכים במבנה	object_name . member_name = value;	<pre>MyPoint.x=5; MyPoint.y=10;</pre>

מבנה - structure ; ערך - value ; איבר - member

file input/output (קלט/פלט עם קבצים)

Description	Syntax	Example
Opening a file	FILE * fopen(const char * File_Name , const char * Mode);	FILE *f; f=fopen ("MyFile.txt" , "w") ;
Closing a stream	int fclose(FILE * file);	fclose (f) ;
Reading from a stream using fgetc	int fgetc(FILE *fp);	char c; c=fgetc (f) ;
Writing to a stream using fputc	int fputc(int c, FILE *fp);	fputc ('A' , f) ;
Reading from a stream using fscanf	fscanf(FILE *fp , format [,arg1,arg2,...]);	int num; fscanf (f , "%d" , &num) ;
Writing to a stream using fprintf	fprintf(FILE *fp , format [,arg1,arg2,...]);	int a=10; fprintf (f , "a=%d" , a) ;

ערך - value ; כתובת חומרה - hardware address

Mode*	Description
r	open for reading
w	open for writing, creates file if it doesn't exist
a	open for appending, creates file if it doesn't exist

* The character string "Mode" specifies the type of access requested for the file.

בהצלחה!

נספח: מילון מונחים (2 עמודים)

לשאלון 711003, אביב תשע"ה

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
odd	Нечётный	فردِيّ	אי-זוגיים
current driver	Усилитель тока	مُحَفِّز تيارِيّ	דוחף זרם
stimulation pulse	Строблирующий сигнал	نبض التحفيز	דופק דירבון
clock pulse	Тактовый сигнал	نبض الساعة	דופק שעון
bubble diagram	Диаграмма состояний	مُخَطَّط الفقاعات / مُخَطَّط العلاقات	דיאגרמת בועות
flip-flop	Триггер	قَلَاب	דלגלג
even	Чётный	زوجِيّ	זוגיים
monostable	Одновибратор	أَحَادِيّ الاستقرار	חד-יציב
one dimension	Одномерный	واحد / أَحَادِيّ البُعد	חד-ממדי
truth table	Таблица истинности	جدول الصدق	טבלת אמת
transitions table	Таблица переходных состояний	جدول النقل	טבלת מעברים
states table	Таблица состояний	جدول الحالة	טבלת מצבים
rotational speed	Скорость вращения	سرعة الدوران	מהירות סיבוב
counter	Счётчик	عَدَاد	מונה
engine	Двигатель	مُحَرِّك	מנוע
array	Массив	مصفوفة	מערך
synchronous tracking system	Синхронный автомат последовательных состояний	نظام تتبّع غير متزامن	מערכת עקיבה סינכרונית
output port	Устройство вывода	مفتاح الإخراج	מפתח-פלט
logic value	Логический сигнал	قيمة منطقيّة	ערך לוגי
output	Выходные данные	الإخراج	פלט

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
file	Файл	ملف	קובץ
multi-vibrator	Мультивибратор	مُتَعَدِّد الاهتزاز	רב-רטט
logic level	Логический уровень	مستوى منطقي	רמה לוגית
vibrations frequency	Частота колебаний	تردد الذبذبات	תדר תנודות