

מערכות אלקטרוניות ומחשבים ט'

למתמחים במערכות אלקטרוניות במגמת הנדסת אלקטרוניקה ומחשבים
(כיתה י"ג)

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 בבהירות.
- 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-19 עמודי נספחים.

ההנחיות בשאלון זה מנוסחות בלשון זכר,

אך מכוונות הן לנבחנות והן לנבחנים.

בהצלחה!

◀ המשך מעבר לדף

elec4u.co.il

השאלות

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

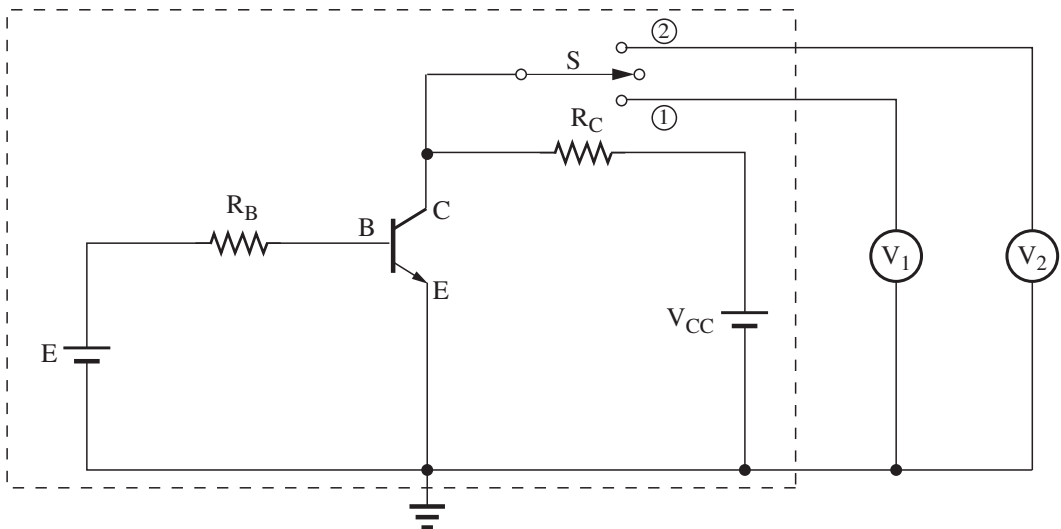
פרק ראשון: מדידות ומכשור

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

שאלה 1

- במעגל החשמלי, המתואר באיור לשאלה 1, המתח בין הנקודות E ו- C הוא $V_{CE} = 10\text{ V}$. מודדים את המתח הזה באמצעות שני מדי-מתח, (V_1) ו- (V_2) :
- כאשר מעבירים את המתג S למצב ① – המתח הנמדד הוא $V_{CE} = 9.189\text{ V}$.
 - כאשר מעבירים את המתג S למצב ② – המתח הנמדד הוא $V_{CE} = 9.890\text{ V}$.

מעגל חשמלי

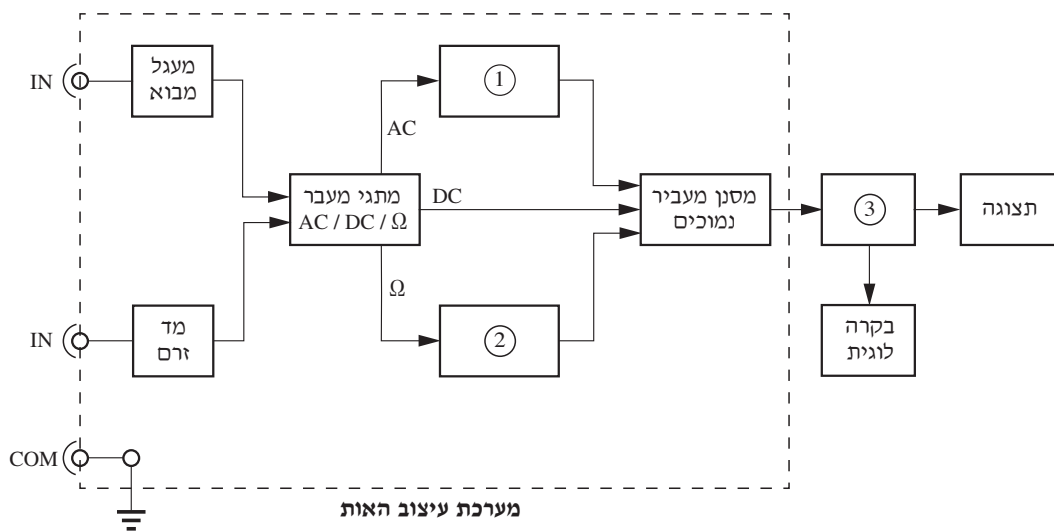


איור לשאלה 1

- חשב את השגיאה המוחלטת בכל אחת מן המדידות.
- חשב את השגיאה היחסית בכל אחת מן המדידות.
- מהי הסיבה לשגיאות המדידה של מדי-המתח, ובאיזה מהם רצוי להשתמש במדידת המתח?

שאלה 2

באיור לשאלה 2 נתון תרשים מלבנים של רב־מודד ספרתי.

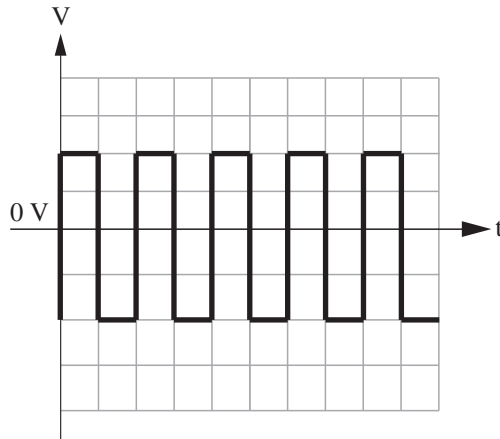


איור לשאלה 2

- א. רשום את שמה והסבר את תפקידה של כל אחת מן היחידות המסומנות בספרות $(1) \div (3)$.
- ב. מדוע קיימים ברב־מודד ספרתי מבואות שונים למדידת מתח וזרם?
- ג. התצוגה של רב־המודד הזה היא בעלת שלוש ספרות. למכשיר שני תחומי מדידה של מתח ישר: $0 \div 10 \text{ V}$ ו- $0 \div 100 \text{ V}$.
- מהו המתח המרבי שיכול להופיע בתצוגה בכל אחד מתחומי המדידה?
 - מהו כושר ההבחנה של המכשיר בכל אחד מן התחומים?
 - מעוניינים למדוד באמצעות המכשיר את המתח 8.12 V . באיזה תחום תבצע את המדידה? נמק את תשובתך.

שאלה 3

באיור לשאלה 3 מתואר האות המתקבל על מסך של משקף תנודות. המסך מחולק ל-8 שורות ו-10 עמודות.



איור לשאלה 3

הבחון המקוץ במכשיר נמצא במצב $\times 10$ (מנחית את אות-המבוא פי 10).

מצב הבוררים במשקף התנודות הוא:

- בורר המבוא: במצב צימוד AC
- בורר רגישות הזמן: $0.5 \text{ msec} / \text{div}$
- בורר רגישות המתח: $5 \text{ V} / \text{div}$
- מקור הדירבון (Trigger Source): במצב של ערוץ 1 (CH1).
- בורר השיפוע (Slope): במצב +
- בורר ההתנעה: במצב NORMAL

א. 1. הסבר את תפקידו של הבחון המקוץ.

2. היעזר באיור וקבע אם הבחון המקוץ מכויל. נמק את קביעתך.

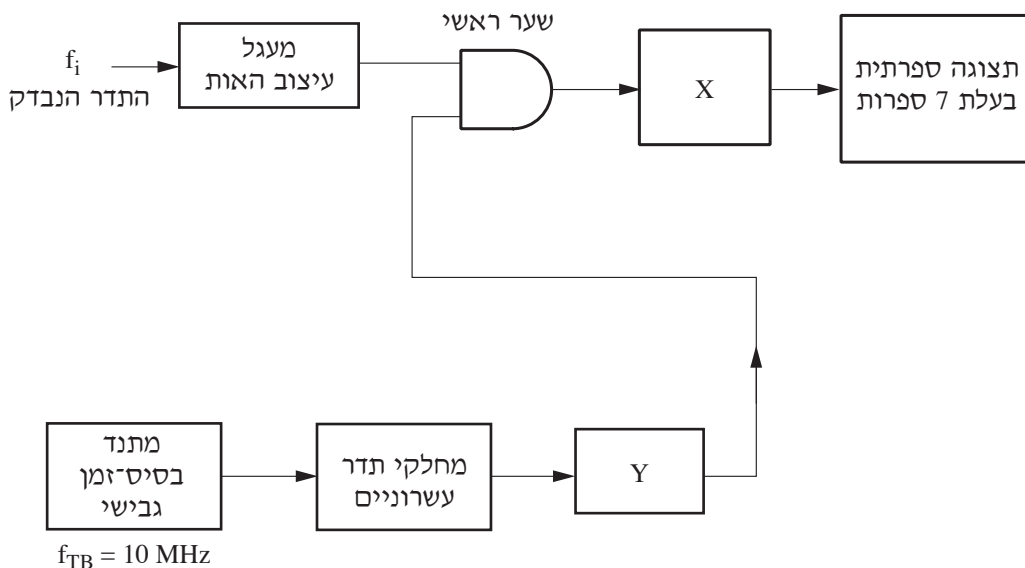
ב. מהו המתח משיא לשיא (p.t.p) של האות הנבדק?

ג. חשב את תדר האות הנבדק.

ד. אילו שינויים יש לבצע במצב הבוררים במשקף על-מנת למדוד את רמת ה-DC של אות אחר?

שאלה 4

באיור לשאלה 4 נתון תרשים מלבנים עקרוני של מד-תדר.



איור לשאלה 4

- א. ציין את שמות היחידות X ו-Y, ורשום את התפקיד של כל אחת מהן.
- ב. להלן תוצאת מדידה (ביחידות kHz) שהתקבלה בתצוגה הספרתית בת שבע הספרות של מד-התדר:

0	2	9	7	5	5	.5
---	---	---	---	---	---	----

- מהו התדר המרבי שניתן להציג במד-התדר הזה, כאשר הנקודה העשרונית ממוקמת בין שתי הספרות הימניות בתצוגה?
- מהו כושר-ההבחנה (הרזולוציה) של המכשיר במדידה הזו?
- התדר של מתנד בסיס-זמן במכשיר הוא $f_{TB} = 10 \text{ MHz}$. חשב את זמן המדידה של התוצאה הזו.
- מעוניינים לשפר את כושר-ההבחנה של המכשיר הזה פי 10. כיצד ניתן לבצע זאת?

פרק שני: מיקרו־בקרים ושפה עילית

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

שאלה 5

להלן תת־שגרה הכתובה בשפת־הסף של המיקרו־בקר 8051 :

```
1.      START:    MOV      DPTR,#1000H
2.              MOV      R7,#0AH
3.              MOVX     A,@DPTR
4.              MOV      40H,A
5.              MOV      41H,#01H
6.      AGAIN:    INC      DPTR
7.              MOVX     A,@DPTR
8.              CJNE     A,40H,NOTE
9.              INC      41H
10.     NOTE:     DJNZ     R7,AGAIN
11.              RET
```

א. הסבר את ההוראות שבשורות 4, 5, 8 ו־9.

ב. בטבלה שלהלן נתונים תכני התאים שכתובותיהם $1000H \div 100AH$ בזיכרון החיצוני של המיקרו־בקר, לפני ביצוע התת־שגרה.

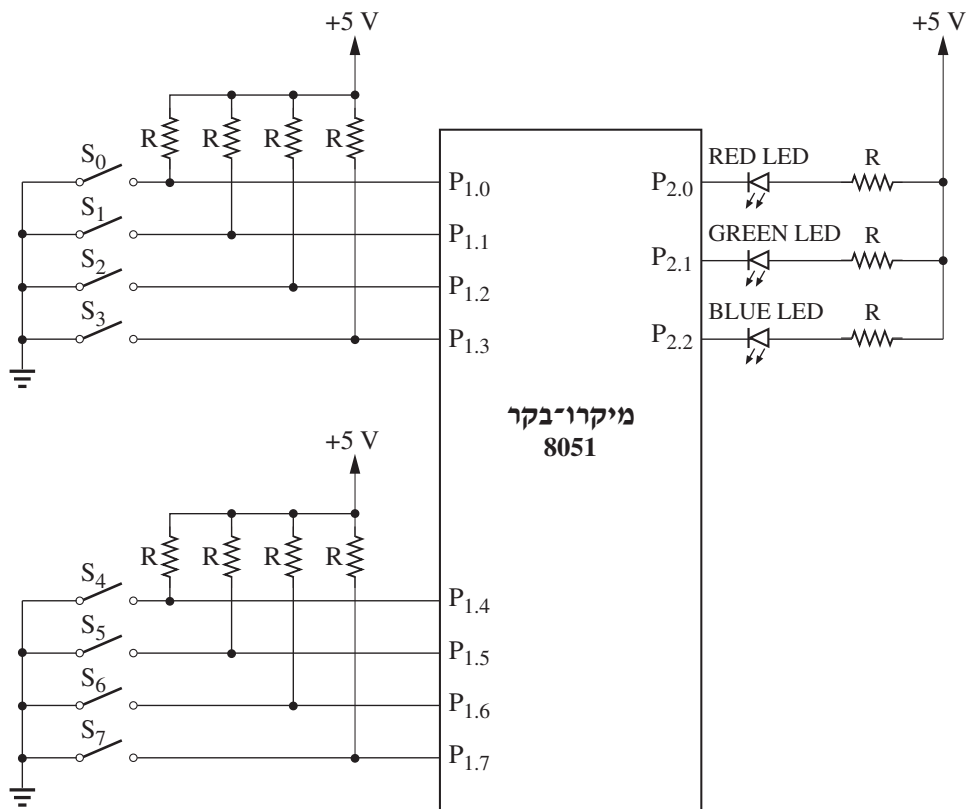
כתובת התא	1000H	1001H	1002H	1003H	1004H	1005H	1006H	1007H	1008H	1009H	100AH
תוכן התא	A4H	A3H	A3H	A6H	A3H	A5H	FFH	A4H	A3H	FFH	A3H

מה יהיו תכני התאים, שכתובותיהם בזיכרון הפנימי של המיקרו־בקר הן $40H$ ו־ $41H$, לאחר ביצוע התת־שגרה? נמק את תשובתך במילים או הצג טבלת־מעקב אחר כל האוגרים הלוקחים חלק בתת־השגרה.

ג. מה צריכים להיות תכני התאים שכתובותיהם $1000H \div 100AH$ בזיכרון החיצוני של המיקרו־בקר, כדי שתוכן התאים $40H$ ו־ $41H$ בזיכרון הפנימי של המיקרו־בקר יהיה $0BH$ לאחר ביצוע התת־שגרה? נמק את תשובתך.

שאלה 6

באיור לשאלה 6 נתון מיקרו־בקר 8051. ההדקים $P_{1.0} \div P_{1.3}$ שלו חוברו, בהתאמה, למפסקים $S_0 \div S_3$, וההדקים $P_{1.4} \div P_{1.7}$ שלו חוברו, בהתאמה, למפסקים $S_4 \div S_7$. כמו־כן חוברו נורית LED אדומה להדק $P_{2.0}$, נורית LED ירוקה להדק $P_{2.1}$, ונורית LED כחולה להדק $P_{2.2}$.



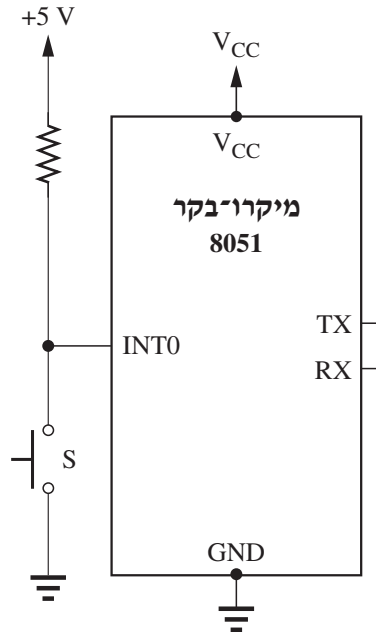
איור לשאלה 6

כתוב תכנית בשפת C של המיקרו־בקר 8051, שתבצע את הפעולות שלהלן:

1. תקלוט את מצב המפסקים $S_0 \div S_7$.
2. אם כל המפסקים פתוחים – היא תדליק את נורית ה־LED הכחולה בלבד.
3. אם לפחות אחד מהמפסקים סגור – התכנית תבדוק אם מצב המפסקים $S_0 \div S_3$ זהה למצב המפסקים $S_4 \div S_7$, בהתאמה.
אם הוא זהה – התכנית תדליק את נורית ה־LED הירוקה בלבד.
אם הוא אינו זהה – התכנית תדליק את נורית ה־LED האדומה בלבד.

שאלה 7

באיור לשאלה 7 נתון לחצן S המחובר להדק המבוא INT0 במיקרו־בקר 8051. מצבו הרגיל של הלחצן הוא פתוח (N.O.).



איור לשאלה 7

להלן תכנית בשפת C של המיקרו־בקר 8051, העושה שימוש ב־UART ובפסיקות חומרה חיצוניות:

```
1. #include <8051.H>
2. unsigned char CH,I;
3. void int0() interrupt0
4. {
5.     I++;
6.     if(I%3==0)
7.         CH = 'B';
8.     else
9.         CH = 'C';
10. }
11. void main(void)
```

```

12. {
13.     CH='A';
14.     I=0;
15.     TMOD=0x20;
16.     SCON=0x40;
17.     TH1=0xFD;
18.     TL1=0xFD;
19.     TR1=1;
20.     TI=0;
21.     IE=0x81;
22.     IT0=1;
23.     while(1)
24.     {
25.         SBUF=CH;
26.         while(!TI);
27.         TI=0;
28.     }
29. }

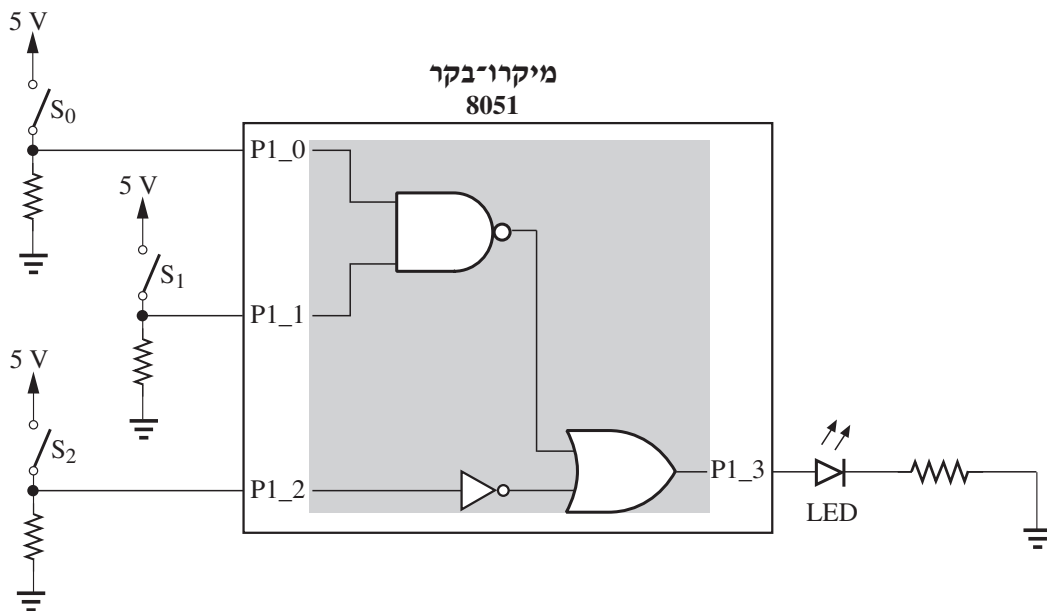
```

- א. הסבר את ההוראות שבשורות 1, 3, 6 ו-26.
- ב. הסבר את המשמעות של השורות $22 \div 15$ בקוד התכנית. היעזר בדף הנוסחאות של המיקרו-בקר לצורך ניתוח מילות הבקרה.
- ג. העתק למחברתך את הטבלה שלהלן, השלם בה את הנתונים שישודרו במוצא ה-UART בכל אחד מן המצבים, ונמק את תשובותיך. התעלם מבעיית הריטוטים העשויים להתרחש בעת לחיצה על הלחצן S.

נימוק	הנתונים שישודרו במוצא ה-UART	
		לפני הלחיצה הראשונה
		לאחר הלחיצה הראשונה
		לאחר 2 לחיצות
		לאחר 3 לחיצות
		לאחר 4 לחיצות
		לאחר 5 לחיצות
		לאחר 6 לחיצות
		לאחר 7 לחיצות

שאלה 8

באיור לשאלה 8 נתון מיקרו־בקר 8051. מעוניינים לממש בתוכנה את המעגל הלוגי המתואר בתוכו.



איור לשאלה 8

א. כתוב תכנית בשפת־הסף של המיקרו־בקר 8051 שתממש את המעגל הלוגי המתואר באיור. על התכנית:

1. לקלוט את הרמה הלוגית של המפסקים $S_0 \div S_2$ דרך שלושת ההדקים $P1_0 \div P1_2$, בהתאמה.

2. לחשב את הרמה הלוגית במוצא המעגל הלוגי, ולהציג אותה על־ידי נורית ה־LED במוצא ההדק $P1_3$.

ב. תדר הגביש של המיקרו־בקר הוא 12 MHz. חשב את הזמן שיעבור מרגע שינוי מצב המפסקים במבוא הבקר ועד לרגע שינוי מצב ה־LED, בהתאם לקוד התכנית שכתבת בסעיף א'.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.

אין להעתיק או לפרסם אלא ברשות משרד החינוך.

אין להעביר את הנוסחאון
לנבחן אחר

מקום לנכתוב תשובות

נוסחאון במדידות ומכשור לכיתה י"ג

(3 עמודים)

שגיאות מדידה

א. שגיאה אקראית

$\bar{X}$ - הערך הקרוב ביותר
לערך האמיתי
(ממוצע חשבוני
משוקלל)

X_i - ערך מדידה i

n - מספר המדידות

f_i - שכיחות מדידה i

k - מספר השכיחויות

σ - סטיית התקן

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} = \frac{X_1 + X_2 + \dots + X_n}{n}$$

$$\bar{X} = \frac{\sum_{i=1}^k f_i \cdot X_i}{\sum_{i=1}^k f_i} = \frac{f_1 \cdot X_1 + f_2 \cdot X_2 + \dots + f_k \cdot X_k}{f_1 + f_2 + \dots + f_k} \quad \text{או}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$$

$$= \sqrt{\frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \dots + (X_n - \bar{X})^2}{n - 1}}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^k f_i \cdot (X_i - \bar{X})^2}{\left(\sum_{i=1}^k f_i\right) - 1}} \quad \text{או}$$

$$= \sqrt{\frac{f_1 \cdot (X_1 - \bar{X})^2 + f_2 \cdot (X_2 - \bar{X})^2 + \dots + f_k \cdot (X_k - \bar{X})^2}{\left(\sum_{i=1}^k f_i\right) - 1}}$$

ב. שגיאה שיטתית

$$\Delta = \sqrt{\Delta_1^2 + \Delta_2^2}$$

Δ - שגיאה כוללת של מכשיר המדידה

ג. שגיאת מדידה כוללת

$$\Delta X = \sqrt{\Delta^2 + \sigma^2}$$

Δ_1 - שגיאה הנובעת מדיוק מכשיר המדידה

Δ_2 - כושר ההבחנה של מכשיר המדידה

$$\delta\% = 100\% \cdot \frac{\Delta X}{\bar{X}}$$

ΔX - שגיאת המדידה הכוללת

$\delta\%$ - שגיאה כוללת יחסית (באחוזים)

$\bar{X}$ - הערך הקרוב ביותר לערך האמיתי (ממוצע חשבוני משוקלל)

מונה

$$f = \frac{n}{t}$$

f [Hz] - תדירות ממוצעת של אירועים נמדדים

n - מספר האירועים

t [sec] - משך המדידה

S_T [1 / °C] - גורם היציבות של מתנד השעון של המונה

$$S_T = \frac{\Delta f_0 / f_0}{\Delta T}$$

f_0 [Hz] - התדר של מתנד השעון

Δf_0 [Hz] - השינוי בתדר של מתנד השעון

ΔT [°C] - השינוי בטמפרטורה

משקף תנודות

א. זמן העלייה של משקף תנודות

זמן העלייה - t_r [sec]

רוחב-פס - BW [Hz]

$$t_r = \frac{0.35}{BW}$$

ב. זמן העלייה הכולל של משקף תנודות

זמן העלייה הכולל של
משקף תנודות, בהתחשב
בזמן העלייה של האות

$$t_{r_s} = \sqrt{t_r^2 + t_{r_1}^2}$$

זמן העלייה של האות הנמדד - t_{r_1}

ג. מדידת הפרש מופע באמצעות משקף תנודות

הפרש מופע - $\Delta\phi$ [°]

הפרש זמנים - Δt [sec]

זמן מחזור - T [sec]

$$\Delta\phi = 360 \cdot \frac{\Delta t}{T}$$

ד. בחון מקוזה ומכיל

התנגדות הבחון - R_1 [Ω]

קיבול הבחון - C_1 [F]

התנגדות המבוא של
משקף התנודות

קיבול המבוא של
משקף התנודות

מקדם הניחות של הבחון - k

$$R_1 C_1 = R_2 C_2$$

$$k = \frac{R_2}{R_1 + R_2}$$

בהצלחה!

נוסחאון במיקרו-בקר 8051

לכיתה י"ג

(15 עמודים)

אין להעביר את הנוסחאון
לנבחן אחר

מקום למחברות נא

נוסחאון בשפת ASM-51

ARITHMETIC OPERATIONS				DATA TRANSFER (cont.)			
Mnemonic	Description	Byte	Cyc	Mnemonic	Description	Byte	Cyc
ADD A,Rn	Add register to Accumulator	1	1	MOVC A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
ADD A,direct	Add direct byte to Accumulator	2	1	MOVC A,@A+PC	Move Code byte relative to PC to A	1	2
ADD A,@Ri	Add indirect RAM to Accumulator	1	1	MOVX A,@Ri	Move External RAM (8-bit addr) to A	1	2
ADD A,#data	Add immediate data to Accumulator	2	1	MOVX A,@DPTR	Move External RAM (16-bit addr) to A	1	2
ADDC A,Rn	Add register to Accumulator with Carry	1	1	MOVX @Ri,A	Move A to External RAM (8-bit addr)	1	2
ADDC A,direct	Add direct byte to A with Carry flag	2	1	MOVX @DPTR,A	Move A to External RAM (16-bit addr)	1	2
ADDC A,@Ri	Add indirect RAM to A with Carry flag	1	1	PUSH direct	Push direct byte onto stack	2	2
ADDC A,#data	Add immediate data to A with Carry flag	2	1	POP direct	Pop direct byte from stack	2	2
SUBB A,Rn	Subtract register from A with Borrow	1	1	XCH A,Rn	Exchange register with Accumulator	1	1
SUBB A,direct	Subtract direct byte from A with Borrow	2	1	XCH A,direct	Exchange direct byte with Accumulator	2	1
SUBB A,@Ri	Subtract indirect RAM from A w/Borrow	1	1	XCH A,@Ri	Exchange indirect RAM with A	1	1
SUBB A,#data	Subtract immed. data from A w/Borrow	2	1	XCHD A,@Ri	Exchange low-order Digit ind. RAM w/A	1	1
INC A	Increment Accumulator	1	1	BOOLEAN VARIABLE MANIPULATION			
INC Rn	Increment register	1	1	Mnemonic	Description	Byte	Cyc
INC direct	Increment direct byte	2	1	CLR C	Clear Carry flag	1	1
INC @Ri	Increment indirect RAM	1	1	CLR bit	Clear direct bit	2	1
DEC A	Decrement Accumulator	1	1	SETB C	Set Carry flag	1	1
DEC Rn	Decrement register	1	1	SETB bit	Set direct Bit	2	1
DEC direct	Decrement direct byte	2	1	CPL C	Complement Carry flag	1	1
DEC @Ri	Decrement indirect RAM	1	1	CPL bit	Complement direct bit	2	1
INC DPTR	Increment Data Pointer	1	2	ANL C,bit	AND direct bit to Carry flag	2	2
MUL AB	Multiply A & B	1	4	ANL C/bit	AND complement of direct bit to Carry	2	2
DIV AB	Divide A by B	1	4	ORL C,bit	OR direct bit to Carry flag	2	2
DA A	Decimal Adjust Accumulator	1	1	ORL C/bit	OR complement of direct bit to Carry	2	2
LOGICAL OPERATION				MOV C,bit	Move direct bit to Carry flag	2	1
Mnemonic	Destination	Byte	Cyc	MOV bit,C	Move Carry flag to direct bit	2	2
ANL A,Rn	AND register to Accumulator	1	1	PROGRAM AND MACHINE CONTROL			
ANL A,direct	AND direct byte to Accumulator	2	1	Mnemonic	Description	Byte	Cyc
ANL A,@Ri	AND indirect RAM to Accumulator	1	1	ACALL addr11	Absolute Subroutine Call	2	2
ANL A,#data	AND immediate data to Accumulator	2	1	LCALL addr16	Long Subroutine Call	3	2
ANL direct,A	AND Accumulator to direct byte	2	1	RET	Return from subroutine	1	2
ANL direct,#data	AND immediate data to direct byte	3	2	RETI	Return from interrupt	1	2
ORL A,Rn	OR register to Accumulator	1	1	AJMP addr11	Absolute Jump	2	2
ORL A,direct	OR direct byte to Accumulator	2	1	LJMP addr16	Long Jump	3	2
ORL A,@Ri	OR indirect RAM to Accumulator	1	1	SJMP rel	Short Jump (relative addr)	2	2
ORL A,#data	OR immediate data to Accumulator	2	1	JMP @A+DPTR	Jump indirect relative to the DPTR	1	2
ORL direct,A	OR Accumulator to direct byte	2	1	JZ rel	Jump if Accumulator is Zero	2	2
ORL direct,#data	OR immediate data to direct byte	3	2	JNZ rel	Jump if Accumulator is Not Zero	2	2
XRL A,Rn	Exclusive-OR register to Accumulator	1	1	JC rel	Jump if Carry flag is set	2	2
XRL A,direct	Exclusive-OR direct byte to Accumulator	2	1	JNC rel	Jump if No Carry flag	2	2
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	1	JB bit,rel	Jump if direct Bit set	3	2
XRL A,#data	Exclusive-OR immediate data to A	2	1	JNB bit,rel	Jump if direct Bit Not set	3	2
XRL direct,A	Exclusive-OR Accumulator to direct byte	2	1	JBC bit,rel	Jump if direct Bit is set & Clear bit	3	2
XRL direct,#data	Exclusive-OR immediate data to direct	3	2	CJNE A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CLR A	Clear Accumulator	1	1	CJNE A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CPL A	Complement Accumulator	1	1	CJNE Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
RL A	Rotate Accumulator Left	1	1	CJNE @Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
RLC A	Rotate A Left through the Carry flag	1	1	DJNZ Rn,rel	Decrement register & Jump if Not Zero	2	2
RR A	Rotate Accumulator Right	1	1	DJNZ direct,rel	Decrement direct & Jump if Not Zero	3	2
RRC A	Rotate A Right through Carry flag	1	1	NOP	No operation	1	1
SWAP A	Swap nibbles within the Accumulator	1	1	Notes on data addressing modes:			
DATA TRANSFER				Rn	-Working register R0-R7		
Mnemonic	Description	Byte	Cyc	direct	-128 internal RAM locations, any I/O port, control or status register		
MOV A,Rn	Move register to Accumulator	1	1	@Ri	-Indirect internal RAM location addressed by register R0 or R1		
MOV A,direct	Move direct byte to Accumulator	2	1	#data	-8-bit constant included in instruction		
MOV A,@Ri	Move indirect RAM to Accumulator	1	1	#data16	-16-bit constant included as bytes 2 & 3 of instruction		
MOV A,#data	Move immediate data to Accumulator	2	1	bit	-128 software flags, any I/O pin, control or status bit		
MOV Rn,A	Move Accumulator to register	1	1	Notes on program addressing modes:			
MOV Rn,direct	Move direct byte to register	2	2	addr16	-Destination address for LCALL & LJMP may be anywhere within the 64-Kilobyte program memory address space.		
MOV Rn,#data	Move immediate data to register	2	1	addr11	-Destination address for ACALL & AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.		
MOV direct,A	Move Accumulator to direct byte	2	1	rel	-SJMP and conditional jumps include an 8-bit offset byte. Range is +127/-128 bytes relative to first byte of the following instruction.		
MOV direct,Rn	Move register to direct byte	2	2				
MOV direct,direct	Move direct byte to direct	3	2				
MOV direct,@Ri	Move indirect RAM to direct byte	2	2				
MOV direct,#data	Move immediate data to direct byte	3	2				
MOV @Ri,A	Move Accumulator to indirect RAM	1	1				
MOV @Ri,direct	Move direct byte to indirect RAM	2	2				
MOV @Ri,#data	Move immediate data to indirect RAM	2	1				
MOVDPTR,#data16	Load Data Pointer with a 16-bit constant	3	2				

INSTRUCTIONS THAT AFFECT FLAG SETTINGS'

INSTRUCTION	FLAG	INSTRUCTION	FLAG
	C 0V AC		C 0V AC
ADD	X X X	CLR C	0
ADDC	X X X	CPL C	X
SUBB	X X X	ANL C,bit	X
MUL	0 X	ANL C/bit	X
DIV	0 X	ORL C,bit	X
DA	X	ORL C/bit	X
RRC	X	MOV C,bit	X
RLC	X	CJNE	X
SETB C	1		

Special Function Registers

P3-Alternate Special Functions of Port 3

(MSB)

(LSB)

RD	WR	T1	T0	INT1	INT0	TXD	RXD
----	----	----	----	------	------	-----	-----

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
RD	P3.7	Read data control output. Active low pulse generated by hardware when external data memory is read.	INT1	P3.3	Interrupt 1 input pin. Low-level or falling-edge triggered.
			INT0	P3.2	Interrupt 0 input pin. Low-level or falling-edge triggered.
WR	P3.6	Write data control output. Active low pulse generated by hardware when external data memory is written.	TXD	P3.1	Transmit Data pin for serial port in UART mode. Clock output in shift register mode.
T1	P3.5	Timer/counter 1 external input or test pin.	RXD	P3.0	Receive Data pin for serial port in UART mode. Data I/O pin in shift register mode.
T0	P3.4	Timer/counter 0 external input or test pin.			

TMOD-Timer/Counter Mode Register

(MSB)

(LSB)

GATE	C / $\bar{T}$	M1	M0	GATE	C / $\bar{T}$	M1	M0
------	---------------	----	----	------	---------------	----	----

TIMER1

TIMER0

M1

M0

Operating Mode

MCS-48 Timer "TLx"
 serves as five bit
 prescaler.

0

0

16-bit timer/counter.
 "THx" and "TLx" are
 cascaded; there is no
 prescaler.

1

0

8-bit auto-reload timer/
 counter. "THx" holds
 a value which is to be
 reloaded into "TLx"
 each time it overflows.

GATE Gating control. When set, Timer/
 counter "x" is enabled only while
 "INTx" pin is high and "TRx"
 control bit is set. When cleared,
 timer/counter is enabled whenever
 "TRx" control bit is set.

1

1

(Timer 0) TL0 is
 an eight-bit timer/
 counter controlled by
 the standard Timer 0
 control bits. TH0 is an
 eight-bit timer
 only controlled by
 Timer 1 control bits.

C / $\bar{T}$ Timer or Counter Selector. Cleared
 for Timer operation (input from
 internal system clock).
 Set for Counter operation (input
 from "Tx" input pin).

1

1

(Timer 1) Timer/
 counter 1 stopped.

TCON-Timer/Counter Control/Status Register

(MSB)

(LSB)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on timer/ counter overflow. Cleared when interrupt processed.	IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.	IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on timer/ counter overflow. Cleared when interrupt processed.	IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.	IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SCON-Serial Port Control/Status Register

(MSB)

(LSB)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
SM0	SCON.7	Serial port Mode control bit 0. Set/cleared by software (see note).	RB8	SCON.2	Receive Bit 8. Set/cleared by hardware to indicate state of ninth data bit received.
SM1	SCON.6	Serial port Mode control bit 1. Set/cleared by software (see note).	TI	SCON.1	Transmit Interrupt flag. Set by hardware when byte transmitted. Cleared by software after servicing.
SM2	SCON.5	Serial port Mode control bit 2. Set by software to disable reception of frames for which bit 8 is zero.	RI	SCON.0	Receive Interrupt flag. Set by hardware when byte received. Cleared by software after servicing.
REN	SCON.4	Receiver Enable control bit. Set/cleared by software to enable/disable serial data reception.			
TB8	SCON.3	Transmit Bit 8. Set/cleared by hardware to determine state of ninth data bit transmitted in 9-bit UART mode.			

Note — the state of (SM0, SM1) selects:

- (0,0) — Shift register I/O expansion.
- (0,1) — 8 bit UART, variable data rate.
- (1,0) — 9 bit UART, fixed data rate.
- (1,1) — 9 bit UART, variable data rate.

קצב העברת הנתונים ב־UART באופני־פעולה Mode 1 או Mode 3:

$$\text{Baud Rate} = \frac{\text{Clock Frequency}}{12 \cdot 32 \cdot (256 - \text{TH1})}$$

Baud Rate [baud] – קצב העברת הנתונים

Clock Frequency [Hz] – תדר השעון

TH1 – תוכן הבית העליון של Timer 1 (בבסיס 10)

IE-Interrupt Enable Register

(MSB)

(LSB)

EA	—	—	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
EA	IE.7	Enable All control bit. Cleared by software to disable all interrupts, independent of the state of IE.4-IE.0.	EX1	IE.2	Enable External interrupt 1 control bit. Set/cleared by software to enable/disable interrupts from INT1.
—	IE.6	(reserved)	ET0	IE.1	Enable Timer 0 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 0.
—	IE.5	(reserved)	EX0	IE.0	Enable External interrupt 0 control bit. Set/cleared by software to enable/disable interrupts from INT0.
ES	IE.4	Enable Serial port control bit. Set/cleared by software to enable/disable interrupts from TI or RI flags.			
ET1	IE.3	Enable Timer 1 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 1.			

IP-Interrupt Priority Control Register

(MSB)	—	—	—	PS	PT1	PX1	PT0	PX0	(LSB)
Symbol	Position	Name and Significance			Symbol	Position	Name and Significance		
—	IP.7	(reserved)			PX1	IP.2	External interrupt 1		
—	IP.6	(reserved)					Priority control bit. Set/cleared by software to specify high/low priority interrupts for INT1.		
—	IP.5	(reserved)							
PS	IP.4	Serial port Priority control bit. Set/cleared by software to specify high/low priority interrupts for Serial port.			PT0	IP.1	Timer 0 Priority control bit. Set/cleared by software to specify high/low priority interrupts for timer/counter 0.		
PT1	IP.3	Timer 1 Priority control bit. Set/cleared by software to specify high/low priority interrupts for timer/counter 1.			PX0	IP.0	External interrupt 0 Priority control bit. Set/cleared by software to specify high/low priority interrupts for INT0.		

Register	Address	Function
P0	80H*	Port 0
SP	81H	Stack Pointer
DPL	82H	Data Pointer (Low)
DPH	83H	Data Pointer (High)
TCON	88H*	Timer register
TMOD	89H	Timer Mode register
TL0	8AH	Timer 0 Low byte
TL1	8BH	Timer 1 Low byte
TH0	8CH	Timer 0 High byte
TH1	8DH	Timer 1 High byte
P1	90H*	Port 1
SCON	98H*	Serial Port Control register
SBUF	99H	Serial Port data Buffer
P2	0A0H*	Port 2
IE	0A8H*	Interrupt Enable register
P3	0B0H*	Port 3
IP	0B8H*	Interrupt Priority register
PSW	0D0H*	Program Status Word
ACC	0E0H*	Accumulator (direct address)
B	0F0H*	B register

Interrupt Source	Service Routine Starting Address
(Reset)	0000H
External 0	0003H
Timer/Counter 0	000BH
External 1	0013H
Timer/Counter 1	001BH
Serial Port	0023H

*=bit addressable register

נוסחאון בשפת C של המיקרו־בקר 8051

נוסחאון זה מתאים למהדר Keil uVision3. חלקים ממנו מתאימים גם למהדרים אחרים.

Data Types (טיפוסי נתונים)

Name	Description	תאור	Size	Range
bit	One Bit	ביט בודד	1 bit	0 to 1
char	Character or small integer	תו בודד או בית	1 byte	-128 to 127
unsigned char	Unsigned small integer	בית אחד ללא סימן	1 byte	0 to 255
int	Integer	מספר שלם	2 bytes	-32768 to 32767
unsigned int	Unsigned integer	מספר שלם ללא סימן	2 bytes	0 to 65535
long	Long integer	מספר שלם ארוך	4 bytes	-2147483648 to 2147483647
unsigned long	Unsigned long integer	מספר שלם ארוך ללא סימן	4 bytes	0 to 4294967295
float	Floating point number	מספר ממשי	4 bytes	+/- 1.175494E-38 to +/- 3.402823E+38
sbit	Special Bit	ביט מיוחד	1 bit	0 to 1
sfr	8 bits special Function Registers	בית מיוחד	1 byte	0 to 255
sfr16	16 bits special Function Registers	בית כפול מיוחד	2 bytes	0 to 65535

דוגמאות:

```
unsigned char a;
int b, c;
sfr P1=0x90;
sbit P1_7 = 0x97;
```

Memory Areas (אזורי הזיכרון)

Name	Description	Example
data	places the variable in directly addressable RAM in the micro core	unsigned int data dnum;
xdata	places the variable in external RAM	unsigned char xdata xnum_at_0x8000; *
idata	places the variable in indirectly addressable memory within the micro core	int idata inum;
code	places the variable in program memory	unsigned char code cnum=0xAA;

* _at_ – places the variable in absolute address.

Absolute Memory Access Macros (גישה לזיכרון באמצעות פקודות מקרו)

Syntax	Description	Example
XBYTE[addr]	XBYTE macro accesses individual bytes in the xdata memory.	XBYTE[0x8000]=0x34;
DBYTE[addr]	DBYTE macro accesses individual bytes in the data/idata memory.	DBYTE[0x40]=0x34;

Preprocessor-directives (הנחיות לקדם-מהדר)

Description	Syntax	Example
macro definitions	#define identifier replacement	#define LED P 1_7

identifier – מזהה ; replacement – תחליף

Operators (אופרטורים)

Description	תאור	Operator
Assignment	השמה	=

Initialization of variables (אתחול משתנים)

```
unsigned char 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 מ־ / שווה	> =
Less than or equal to	קטן מ־ / שווה	< =

Logical operators (אופרטורים לוגיים בין ביטויים)

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

Bitwise Operators (אופרטורים על סיביות)

Description	תאור	Operator
AND	וגם	&
Inclusive OR	או כולל	
Exclusive OR (XOR)	או מוציא	^
Byte inversion	היפוך בית	~
Bit inversion	היפוך סיבית	!
Shift Left	הזזה שמאלה	<<
Shift Right	הזזה ימינה	>>

Conditional Structures (מבני בקרה – משפטי תנאי)

Description	Syntax	Example
if	<pre>if (condition) { statements ; }</pre>	<pre>if (d == 100) { P1 = 0xFF; }</pre>
if .. else	<pre>if (condition) statement ; else statement ;</pre>	<pre>if (d == 100) P1 = 0xFF; else P1=0;</pre>
if .. else if .. else	<pre>if (condition) statement ; else if (condition) statement ; else statement ;</pre>	<pre>if (d > 0) P1=4; else if (d < 0) P1=2; else P1=1;</pre>

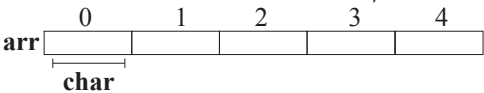
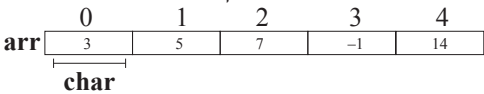
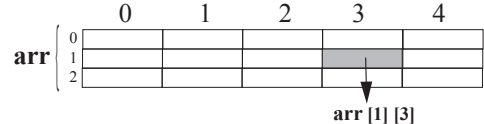
הצהרה – statement ; תנאי – condition

Iteration Structures (מבני בקרה - לולאות)

Description	Syntax	Example
while loop	<pre>while (expression) { statements ; }</pre>	<pre>while (n>0) { P1=n; n- -; }</pre>
do-while loop	<pre>do { statements ; } while (condition);</pre>	<pre>do { n=P1; } while (n != 0);</pre>
for loop	<pre>for (initialization; condition; increase) { statements ; }</pre>	<pre>for (i=0; i<10; i++) { P1=i; }</pre>

קידום - increase ; אתחול - initialization ; הצהרה - statement ; תנאי - condition

Arrays (מערכים)

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

elements – פרטים ; ערך – value

Structure of a program (מבנה כללי של תוכנית)

```
#include <8051.h>           //including headers for the SFR definitions
                             of your microcontroller

void main()
{
    while(1)
    {
        //Your code
    }
}
```

Functions (פונקציות)

Description	Syntax	Example
Functions with no type and no argument	<pre>void name (void) { statements; }</pre>	<pre>void OutData(void) { P1=0xAB; } void main() { OutData(); }</pre>
Functions with no type	<pre>void name (parameter1, parameter2, ...) { statements; }</pre>	<pre>void OutData(unsigned char a, unsigned char b) { P0=a; P1=b; } void main() { OutData(0xF,0xF0); }</pre>
Functions with type and argument	<pre>type name (parameter1, parameter2, ...) { statements; }</pre>	<pre>unsigned char InOutData(unsigned char a) { P1=a; return P0; } void main() { unsigned char r; r = InOutData(63); P2=r; }</pre>

argument – טיעון ; type – טיפוס ; statement – הצהרה ; parameter – ערך המועבר לפונקציה

Interrupt Service Routines (שגרות לטיפול בפסיקות)

Interrupt Number	Description	Address
0	External INT 0	0003h
1	Timer 0	000Bh
2	External INT 1	0013h
3	Timer 1	001Bh
4	Serial port	0023h

דוגמה:

```
void int2sub () interrupt 2
{
    // Interrupt code
}
```

נספח: מילון מונחים

לשאלון 711911, אביב תשע"ו

תרגום המונח			המונח
אנגלית	רוסית	ערבית	
clip probe	Компенсационный пробник	مجسّ תואזן	בחון מקוּזז
crystal	Кристалл	בִּלּוֹר / כריסטל	גביש
external memory	Внешняя память	الذاكرة الخارجيّة	זיכרון חיצוני
internal memory	Внутренняя память	الذاكرة الداخليّة	זיכרון פנימי
resolution	Резолюция	דִּقָּה התבּיין	כושר הבחנה
push-button	Пусковая кнопка	זֶרֶקָבֵס	לחצן
frequency meter	Измеритель частоты	מְקִיָּאס התרֵדד	מד־תדר
microcontroller	Аналого-дигитальный контроллер	מְתַחַקֵּם דָּקִיק	מיקרו־בקר
logic system	Логическая схема	דֹּאִירָה מִנְלִיקִיָּה	מעגל לוגי
switches	Выключатель	מִפְתִּיחַ	מפסקים
oscilloscope	Осциллоскоп	רֹאסֵם הַזִּבְזָבָת	משקף תנודות
oscillator	Осциллятор	מְזִבְזֵב	מתנד
program code	Кодирование	קוּד הַבְּרֹאמַג	קוד התכנית
digital multi-meter	Цифровой много целевой измеритель	מְקִיָּאס רִקְמִי מֵעַדָּד	רב־מודד ספרתי
logic level	Логический уровень	מִסְטוֹי מִנְלִיקִי	רמה לוגית
relative error	Относительная ошибка	חֲטֵא נִסְבִּי	שגיאה יחסית
absolute error	Абсолютная ошибка	חֲטֵא מִלְק	שגיאה מוחלטת
block diagram	Блок-схема	מִחְפָּט סִנְדוּקִי / מִסְתַּטִּילָת	תרשים מלבנים
subroutine	Подпрограмма	רוּתִין פְּרָעִי	תת־שגרה