



מערכות תוכנה להנדסאים - הנדסת אלקטרוניקה

הנחיות לנבחנים

- א. משך הבחינה: ארבע שעות
- ב. מבנה השאלון ומפתח הערכה:
- בשאלון 10 שאלות.
יש לענות על 5 שאלות בלבד בהתאם לפירוט הבא:
חלק א' - שאלות 1-4: יש לענות על 2 שאלות בלבד.
חלק ב' - שאלות 5-10: יש לענות על 3 שאלות בלבד.
כל השאלות שוות בערך - 20 נקודות.
סה"כ 100 נקודות
- ג. חומר עזר מותר בשימוש:
- מחשבון (אין להשתמש במחשב נייד, מחשב כף יד או במחשבון בעל תקשורת חיצונית).
 - קלסר **אחד בלבד** עם חומר ההרצאות. אין להוציא דפים מהקלסר.
 - שני ספרי לימוד.
- ד. הוראות מיוחדות:
- אם צוין בשאלון הבחינה שאלות המכילות השלמת טבלה יש לרשום את הטבלה במחברת הבחינה.**
 - יש לציין את המקור ומספר העמוד במקרים בהם ניתנה תשובה מתוך ספרי הלימוד.
 - יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בגמר הבחינה יש לרשום בעמוד זה את מספרי השאלות שיבדקו לבקשתך על ידי מעריכי הבחינה. לא יבדקו תשובות עודפות על הנדרש.
- ה. הוראות כלליות:
- יש לקרוא בעיון את כל השאלות.
 - את התשובות יש לכתוב במחברת **בעט בלבד**, בכתב יד ברור ונקי. (גם בכך תלויה הערכת הבחינה).
 - יש להתחיל כל תשובה בעמוד חדש תוך ציון מספר השאלה והסעיף. אין צורך להעתיק את השאלה או הסעיף.
 - טיוטה תעשה במחברת הבחינה בלבד. יש לכתוב "טיוטה" ולהעביר קו על התשובה כדי שלא תבדק.
 - יש להציג פתרון מלא ומנומק כולל חישובים כשצריך. הצגת תשובה סופית ללא דרך הפתרון לא תזכה לניקוד.
 - אם להערכתך חסר בשאלה נתון, יש לציין זאת בתשובה ולבחור ערך מתאים שיאפשר לך להמשיך בפתרון השאלה. חובה להסביר ולנמק את הבחירה.

בהצלחה !

חלק א': תוכנה C/C++

יש לענות על 2 שאלות בלבד, מתוך 4 השאלות הבאות (שאלות 1 - 4)

שאלה מספר 1 (20 נקודות)

יש לכתוב מחלקה המכילה מחרוזת בגודל $N=20$.

א) (7 נקודות) יש לרשום את הגדרת המחלקה. המחלקה צריכה להכיל את הפונקציות הבאות:

- ✓ בנאי ברירת מחדל (default constructor) – מאתחל את המחרוזת ל"xxxx".
- ✓ בנאי המקבל כקלט מחרוזת.
- ✓ פונקציה המחזירה את מספר התווים במחרוזת.
- ✓ פונקציה המדפיסה את המחרוזת למסך

בסעיף זה אין לממש פונקציות

- ב) (6 נקודות) יש לממש את בנאי ברירת המחדל, המאתחל את המחרוזת ל"xxxx".
- ג) (7 נקודות) יש לכתוב פונקציה main המגדירה אובייקט אחד מהמחלקה המכיל את המחרוזת "Hello, c" ומדפיסה אותה למסך. אין צורך לממש את פונקציות המחלקה, אלא להשתמש בהן בלבד.

שאלה מספר 2 (20 נקודות)

נתונות ההגדרות הבאות:

```
#define N 20
void print(int *p, int size)
{
    int i;
    for(i=0; i<size; ++i)
        cout << p[i] << ",";
    cout << endl;
}
ובתוך פונקציה
int a[N] = {1, 2, 3, 4, 4, 3, 2, 1};
int *p = a;
int i=0;
```

בכל אחד מהסעיפים הבאים, ישנו קטע קוד המופיע לאחר ההגדרות הנ"ל. כל קטע הוא עצמאי ואינו תלוי בקטעים האחרים. בכל אחד מהסעיפים יש לרשום באופן ברור את הפלט בסיום הקטע.

(א) (7 נקודות) מה יודפס בסיום הקטע הבא?

```
print(p+1, 4);
print(p, 4);
```

(ב) (6 נקודות) מה יודפס בסיום הקטע הבא?

```
i = 2;
a[i+1] = a[a[i-1]];
print(p, 8);
```

(ג) (7 נקודות) מה יודפס בסיום הקטע הבא?

```
for(i=0; i<7; ++i)
{
    *p = *(p+1);
    ++p;
}
print(a, 8);
```

שאלה מספר 3 (20 נקודות)

יש לכתוב לכל אחד מהסעיפים הבאים פונקציה (נפרדת!) ב ++C הבודקת / מדפיסה את המבוקש:

(א) (6 נקודות) פונקציה **רקורסיבית** המקבלת מערך חד מימדי, ואת אורכו, ומחזירה את סכום האיברים במערך.

(ב) (6 נקודות) פונקציה המקבלת מערך דו מימדי בגודל $N \times N$ (N הוא קבוע), ומחזירה את סכום אברי האלכסון הראשי. האלכסון הראשי הוא האלכסון המתחיל בפינה שמאלית-עליונה.

(ג) (8 נקודות) פונקציה המקבלת מערך דו מימדי בגודל $N \times N$ (N הוא קבוע). הפונקציה מחזירה 1 אם יש במערך לפחות שורה אחת שכולה אפסים, אחרת מחזירה הפונקציה 0.

שאלה מספר 4 (20 נקודות)
נתונה התוכנית:

```
#include <iostream.h>
class rec{
    double side1, side2;
public:
    rec(double a, double b){ side1=a; side2=b;}
    virtual void print(){ cout<<side1<<"", "<<side2<<endl; }
    double a(){ return side1*side2; }
    double getl(){ return side1; }
    double getw(){ return side2; }
};
class sq: public rec{
public:
    sq(double a):rec(a,a){}
    virtual void print(){ cout<<getl()<<endl; }
};
void main(){
    rec r(3, 5);
    sq b(4);
    r.print();
    cout << "a = " << r.a() << endl ;
    b.print();
    cout << "a = " << b.a() << endl ;
}
```

(א) (10 נקודות) מהו הפלט של התוכנית? להסביר בקצרה.

(ב) (7 נקודות) מה יהיה הפלט, אם נחליף את הפונקציה main, בפונקציה הבאה?

```
void main(){
    rec *p1 = new rec(2,3), *p2 = new sq(1);
    p1 -> print();
    p2 -> print();
}
```

(ג) (3 נקודות) מה יהיה הפלט בסעיף ב', אם נמחוק את המילה virtual מהגדרת הפונקציה print?

חלק ב': VHDL

יש לענות על 3 שאלות בלבד, מתוך 6 השאלות הבאות (שאלות 5 - 10)

שאלה מספר 5 (20 נקודות) -

נתונה התוכנית הבאה:

א) (10 נקודות) יש לצייר במדויק את החומרה שתיווצר מהתוכנית הנתונה, הכוללת רישום של כל ה"אקטואלים" וה"פורמלים" מהקוד. נא להקפיד על תרשים ברור!

```
library ieee;
use ieee.std_logic_1164.all;
entity tff is
    port (clk: in std_logic;
          t: in std_logic;
          q: out std_logic);
end;

architecture rtl of tff is
    signal sig: std_logic:='0';
begin
    process (clk)
    begin
        if clk'event and clk = '1' then
            if t='1' then
                sig<=not sig;
            end if;
        end if;
    end process;
    q<=sig;
end;

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity test5A is
    GENERIC (Size: INTEGER:= 3);
    port (t, clk: in std_logic;
          q: out std_logic_vector (Size-1 downto 0));
end;
```

```

architecture for_gen of test5A is
    COMPONENT tff
        port(clk: in std_logic; q :out std_logic);
    end component;
    SIGNAL qSig: std_logic_vector (Size-1 DOWNT0 0) := (others=>'0');
    SIGNAL tSig: std_logic_vector (Size-1 DOWNT0 0) := (others=>'0');

begin
    tSig(0)<=t;
    out_loop: FOR i in 0 to size-1 GENERATE
        boxs: tff port map(t=>tSig(i), clk=>clk, q=>qSig(i));
    end GENERATE out_loop;

    -----
    z: FOR i IN 1 TO size-1 GENERATE
        tsig(i)<=qsig(i-1) AND NOT tsig(i-1);
    END GENERATE z;

    -----
    q<=qSig;
end;

```

ב) (10 נקודות) יש לרשום את סדרת המוצאים של התוכנית במספרים עשרוניים, כאשר ניתן להניח שהמערכת מתחילה מ-0, והכניסה t מקבלת '1' באופן קבוע. אם סדרה זו מחזורית, רשום את מחזוריה. אם איננה מחזורית, יש לרשום את 8 המצבים הראשונים בלבד (כולל ה-0).

שאלה מספר 6 (20 נקודות)

נתונה התוכנית הבאה, כאשר בכל הסעיפים נתון: `input = "1000"`. יש להשלים את הטבלאות להלן וגם לצייר את צורות הגלים של הפלט `div_by` ביחס ל-`count`. יש להניח שניתן איפוס קצר לפני הופעת מחזור השעון הראשון:

```
library ieee;
use ieee.STD_LOGIC_1164.all;
use ieee.STD_LOGIC_ARITH.all;
use ieee.STD_LOGIC_UNSIGNED.all;
entity div is
    generic (size: NATURAL:= 4);
    port (clk: in BIT; rst: in BIT;
          input: in std_logic_vector(size-1 downto 0);
          div_by: out std_logic);
end div;
architecture div of div is
    signal count: std_logic_vector(size-1 downto 0);
begin
    process (rst, clk)
    begin
        if rst = '1' then
            count<= input;
            div_by<='1';
        elsif clk'event and clk = '1' then
            if count = 1 then
                count<=input;
                div_by<='0';
            else
                if count < SHR(input,"01")-1 then
                    div_by<='0';
                else
                    div_by<='1';
                end if;
                count<=count-1;
            end if;
        end if;
    end process;
end;
```

(א) (10 נקודות) עבור התוכנית לעיל כמו שהיא, יש לצייר טבלה במחברת ע"י רישום ערכו של div_by ביחס ל-count וגם לצייר את צורות הגלים:

count	div_by
8	
7	
6	
5	
4	
3	
2	
1	

(ב) (10 נקודות) משנים את השורה המודגשת בתוכנית, ובמקומה רושמים את השורה הבאה:

```
if count > conv_integer(input)/2 then
```

מומלץ לרשום לעצמך שורה זו במילים, כדי לוודא ששמת לב לשינויים!
יש לצייר טבלה חדשה במחברת ע"י רישום ערכו של div_by ביחס ל-count וגם לצייר את צורות הגלים:

count	div_by
8	
7	
6	
5	
4	
3	
2	
1	

הערות:

1. הפונקציה SHR קיימת בספריות המצורפות לתוכנית ומבצעת הזזה ימינה.
2. הפונקציה conv_integer קיימת בספריות המצורפות לתוכנית ומתרגמת מווקטור ל-integer. לידיעתך, מותר להשתמש בחילוק ב-2 עבור הסוג integer.
3. הרבה טועים בשאלה זו מאחר והם "שוכחים" שמדובר בדלגלים, בבקשה נא לא לשכוח!

שאלה מספר 7 (20 נקודות)

(20 נקודות) לפניך מכונת מצבים סינכרונית, עם איפוס אסינכרוני.

נתון שמחזור השעון הוא 100ns והוא סימטרי (כלומר: 50ns ב-'0' ואח"כ 50ns ב-'1').

עליך לכתוב עירור שיבדוק 4 ממצבי המכונה, ע"י תוכנית כרצונך (הקל ביותר הוא לכתוב תוכנית עצמאית שהממשק נתון להלן).

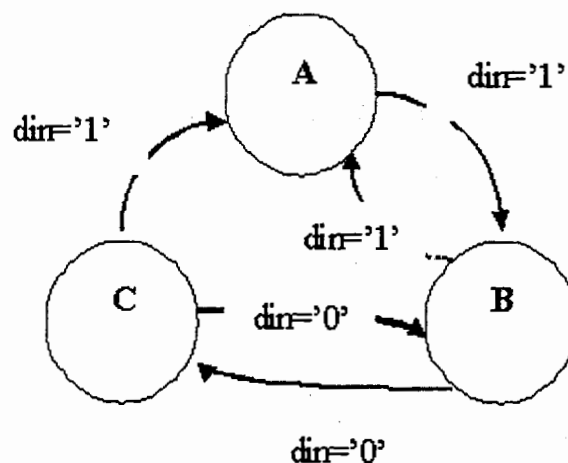
למשל: ממצב A ל-B, ל-C וחזרה ל-A זו אפשרות אחת מתוך ה-4 (בכל אפשרות תמיד חוזרים בסוף ל-A).

סימונים לשאלה זו:

מצב A: "01"

מצב B: "10"

מצב C: "11"



```

ENTITY test7 IS
  PORT(rst, clk: OUT bit;
        din: OUT bit;
        otp: IN bit);
END;
```

שאלה מספר 8 (20 נקודות)

נתון הממשק הבא:

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY cnt IS
    PORT (clk, rst: IN STD_LOGIC;
          vec: IN STD_LOGIC_vector(3 downto 0);
          ones: OUT STD_LOGIC_vector(2 downto 0));
END;
```

א) (10 נקודות) יש לכתוב פונקציה שתספור את מספר האחדים במילה `vec` שבתוכנית לעיל. מותר להשתמש בכל אלגוריתם.

ב) (10 נקודות) כעת, יש להוסיף ארכיטקטורה הקוראת לפונקציה. יש להניח כי בכל מחזור שעון מגיעה כניסה חדשה `vec`.

עליך האחריות לבדוק שכל חלקי הקוד מתאימים והכל עובד כמו שצריך. יש להסביר בעברית את השיטה בה התוכנית כולה עובדת.

שאלה מספר 9 (20 נקודות)

(א) (10 נקודות) יש לצייר את החומרה שתיווצר מתוכנית זו:

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY test09A_What IS
    PORT(clk, rst, inp: IN STD_LOGIC;
          otp: OUT std_logic);
END;

ARCHITECTURE arc OF test09A_What IS
    SIGNAL a, b, c: std_logic;
BEGIN
    otp<= b WHEN inp='0' ELSE
        c;
    process(clk, rst)
    begin
        if rst= '1' then
            a<= '1'; b<= '0';
        elsif rising_edge(clk) then
            a<=inp; b<= a; c<=b;
        end if;
    end process;
end;

```

(ב) (10 נקודות) יש לצייר את החומרה שתיווצר מתוכנית זו:

```

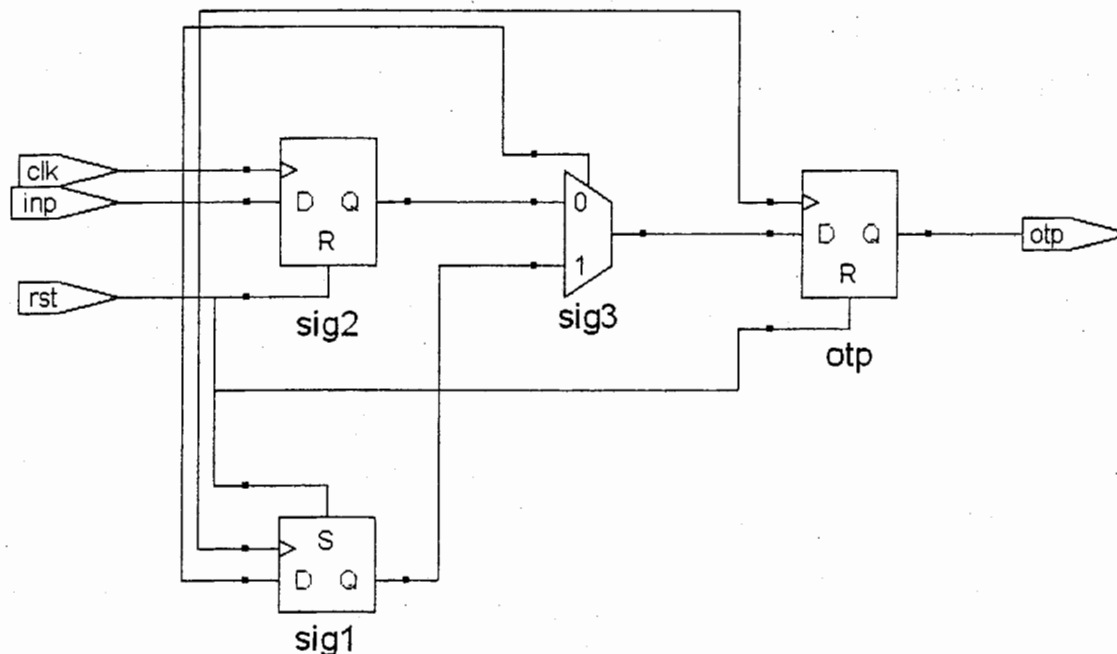
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY test09B_What IS
    PORT(clk, inp1, inp2: IN STD_LOGIC;
          otp: OUT std_logic);
END;

ARCHITECTURE arc OF test09B_What IS
    SIGNAL a, b, c: std_logic;
BEGIN
    process(clk)
    begin
        if rising_edge(clk) then
            a<= inp2; otp<= b; c<=a;
        end if;
    end process;
    b<= inp2 WHEN inp1= '1' ELSE
        c;
end;

```

שאלה מספר 10 (20 נקודות)

בשאלה זו, אסור לכתוב קוד המכיל חיווטים כלשהם.



הערות חשובות לגבי הסימונים בתרשים

1. בדלגלים שבתרשים R=Reset, S=Set, ואילו sig1, sig2, otp הם מוצאי הדלגלים. sig3 הוא מוצא המרבב.
2. המשתנה inp נכנס גם לדלגל שמוצאו sig2 וגם לדלגל שמוצאו sig1 וגם הבורר של המרבב הוא inp.
3. ניתן להניח שכל האותות מסוג bit.
4. חובה להשתמש בסימונים שעל התרשים, אחרת יורדו נקודות.

(20 נקודות) יש לכתוב קוד המכיל תהליך מדורבן שעון אחד (Clocked Process) עבור הדלגלים, ועוד פקודות מקביליות לתיאור המרבב שבתרשים. סך כל הקוד שלך יתאר במדויק את התרשים לעיל.

בהצלחה