

מערכות תוכנה

הנדסאים – הנדסת אלקטרוניקה

הנחיות לנבחן

- א. משך הבחינה: ארבע שעות.
- ב. מבנה השאלון ומפתח ההערכה:
- בשאלון זה 10 שאלות. יש לענות על 5 שאלות בלבד.
חלק א': 40 נקודות
שאלות 1-4: יש לענות על **שתי** שאלות בלבד.
ערך כל שאלה – 20 נקודות.
חלק ב': 60 נקודות
שאלות 5-10: יש לענות על **שלוש** שאלות בלבד.
ערך כל שאלה – 20 נקודות.
בסך הכול: 100 נקודות.
- ג. חומר עזר מותר לשימוש: 1. מחשבון. (אין להשתמש במחשב כף יד או במחשבון המאפשר תקשורת חיצונית).
2. קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר.
3. שני ספרי לימוד.
- ד. הוראות מיוחדות: יש לציין את המקור ואת מספר העמוד במקרים שבהם ניתנה תשובה מתוך ספרי הלימוד.
- ה. הוראות כלליות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות.
2. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות.
3. יש לכתוב את התשובות במחברת הבחינה בעט בלבד, בכתב יד ברור.
4. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה.
5. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק.
6. יש להציג פתרון מלא ומנומק כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד.
7. אם להערכתך חסר נתון בשאלה, ציין זאת בתשובה והוסף נתון מתאים שיאפשר לך להמשיך בפתרון השאלה. חובה להסביר ולנמק את הבחירה.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!
ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות לנבחנות ולנבחנים כאחד.

חלק א' – תוכנה C++ (40 נקודות)

ענה על שאלה אחת מבין שאלות 1-2 ועל שאלה אחת מבין שאלות 3-4. (ערך כל שאלה – 20 נקודות).

שאלה 1

להלן פונקציה רקורסיבית:

```
int what(char* s)
{
    if (*s == '\0')
        return 0;

    if (*s >= 'a' && *s <= 'z')
        return *s + what(s + 1);
    else
        return what(s + 1);
}
```

- 2 נק') א. מה תהיה תוצאת הרצת הפונקציה עם הערך `s="aAbC"`?
- 5 נק') ב. תאר במשפט אחד מה מבצעת הפונקציה, יש להתייחס לכל המקרים האפשריים.
(בלי להסביר "איך הפונקציה מבצעת את העבודה" ובלי להסביר "כל פקודה בנפרד").
- 13 נק') ג. כתבו את הפונקציה הרקורסיבית הבאה:

```
void fillArray(int arr[], int size)
```

הפונקציה מקבלת מערך של מספרים שלמים `arr` שרק שני האיברים הראשונים בו מכילים ערכים, וגודלו `size`. הפונקציה תמלא את כל איברי המערך החל מהאיבר באינדקס 2, כך שכל איבר יכיל את סכום שני האיברים שלפניו.

דוגמה

```
int arr[10] = { 1, 1 };
```

עבור המערך:

הפונקציה תעדכן אותו להיות עם הערכים הבאים:

```
{ 1 1 2 3 5 8 13 21 34 55 }
```

שאלה 2

כתבו את הפונקציה הבאה :

```
bool isMatrixHasSquareAscendingValues(int mat[][SIZE],
                                       int rows, int cols);
```

הפונקציה מקבלת מטריצה שממדיה הם SIZExSIZE (ניתן להניח כי SIZE מוגדר כקבוע כלשהו) וכן שני מספרים שלמים, שנתון כי ערכם קטן מ-SIZE. הפרמטר rows מעיד כמה שורות מתוך SIZE שורות המטריצה בשימוש, והפרמטר cols מעיד כמה עמודות מתוך SIZE עמודות במטריצה בשימוש.

הפונקציה תתייחס לכל רביעיית איברים צמודים כאל איבר אחד (כלומר, האיברים: $mat[0][0]$, $mat[0][1]$, $mat[1][0]$, $mat[1][1]$ יחשבו כאיבר הראשון). יש לשים לב שאיברי רביעייה צריכים להיות בזוג שורות ועמודות אחד (כלומר, לא ניתן להתחיל רביעייה בשורה מסוימת ולסיים אותה בשורה אחרת).

הפונקציה תבדוק האם הערכים בכל רביעייה בסדר משמאל לימין ומלמעלה למטה בעלי מספר סדרתי רציף עולה, החל מהערך 1.

דוגמאות:

עבור המטריצה הבאה יוחזר true :

1	1	2	2	3	3
1	1	2	2	3	3
4	4	5	5	6	6
4	4	5	5	6	6

עבור המטריצה הבאה יוחזר false, מאחר וממדי המטריצה הינם אי-זוגיים ולכן אינם מאפשרים לענות על התנאי.

1	1	2	2	3	3	4
1	1	2	2	3	3	4
5	5	6	6	7	7	8
5	5	6	6	7	7	8

שאלה 3

להלן main :

```

void main()
{
    A a(3);
    a.show();
    cout << "-----\n";

    B b(4, 5);
    b.show();
    cout << "-----\n";

    A* ab = new B(6, 7);
    ab->show();
}

```

A ו-B הם classes
להלן פלט הרצת התוכנית :

```

x=3
-----
x=4
y=5
-----
x=6
y=7

```

עליכם לכתוב את המחלקות המתאימות לקיומו של main זה עבור הפלט הנ"ל.

להלן קטע קוד המכיל 2 מחלקות ו-main. עליכם לכתוב מה יהיה פלט התוכנית.

```
class A
{
protected:
    int x;
public:
    A(int x) : x(x) {}
    ~A() { cout << "In A::~~A\n"; }

    void show() const { cout << "In A::show(const) x=" << x << "\n"; }
    virtual void show() { cout << "In A::show x=" << x << "\n"; }

    virtual const A& operator++() { x++; return *this; }
};

class B : public A
{
public:
    B(int x = 3) : A(x) {}

    ~B() { cout << "In B::~~B\n"; }

    void show() const { cout << "In B::show(const) x=" << x << "\n"; }
    void show() { cout << "In B::show x=" << x << "\n"; }

    virtual const A& operator++() { x += 2; return *this; }
};
```

```
void main()
{
    A a1(3);
    const A a2(4);

    ++a1;
    a1.show();
    a2.show();

    B b1(5);
    const B b2(6);

    ++b1;
    b1.show();
    b2.show();

    A* a3 = new B(7);
    ++(*a3);
    a3->show();

    delete a3;
}
```

חלק ב' – VHDL (60 נקודות)

ענה על שלוש מבין השאלות 5-10 (ערך כל שאלה – 20 נקודות).

שאלה 5

(10 נק') א. שרטט במדויק תרשים המתאר את החומרה שתיווצר על ידי התוכנית הבאה, והסבר האם ניתן להגדיר את כל האותות כ- BIT במקום STD_LOGIC.

```
library ieee ;
use ieee.std_logic_1164.all ;
entity test51 is
  port ( a, b : in std_logic ;
        x0, x1, y1, y2, z0 : in std_logic;
        x, y, z : out std_logic ) ;
end test51 ;
architecture test51 of test51 is
begin
  x <= x0 when a = b else x1 ;
  y <= y1 when a < b else y2 ;
  z <= z0 when a = b else 'Z' ;
end;
```

הערה

הרכיבים שעומדים לרשותך במימוש הם שערים, בוררים, דלגלים, gated Latch, רכיבי Tri-State, נגדים וגם מתחי ספק וחוטטים. אין צורך להשתמש בכל הרכיבים הנ"ל.

(10 נק') ב. שרטט במדויק תרשים המתאר את החומרה שתיווצר על ידי התוכנית הבאה, והסבר מתי נקבל gated Latch.

```
entity test52 is
  port ( a, b, c : in bit ;
        x0, y1 : in bit;
        x, y, z : out bit ) ;
end test52 ;

architecture test52 of test52 is
begin
  x <= x0 when c = '1' ;

  process(a,b,c)
  begin
    if a <= b then
      y <= y1 ;
    end if ;
  end process;

  process(a,b)
  begin
    if a <= b then
      z <= '1' ;
    else
      z <= '0' ;
    end if ;
  end process ;
end ;
```

הערה

הרכיבים שעומדים לרשותך במימוש הם שערים, בוררים, דלגלים, gated Latch, רכיבי Tri-State, נגדים וגם מתחי ספק וחוטרים. אין צורך להשתמש בכל הרכיבים הנ"ל.

שאלה 6

(8 נק') א. צייר סכמה לוגית (RTL) של מעגל החומרה המתואר על ידי התוכנית הבאה :

```
library ieee;
use ieee.std_logic_1164.all;
entity tar6 is port(
    a,b,clr,pre, clk : in std_logic;
    out1 : buffer std_logic;
    out2 : out std_logic);
end tar6;
architecture tar6 of tar6 is
    signal input : std_logic_vector(1 downto 0);
begin
    input <= a&b;
    process (clr, pre, clk)
    begin
        if clr = '0' then out1<='0';
        elsif pre = '0' then out1 <='1' ;
        elsif rising_edge(clk) then
            case input is
                when "00" => null;
                when "10" => out1<='1';
                when "01" => out1<='0';
                when "11" => out1<=not out1;
            end case;
        end if;
        out2<= not out1;
    end process;
end;
```

הערה

הרכיבים שעומדים לרשותך במימוש הם שערים, בוררים, דלגלים, gated Latch, רכיבי Tri-State, נגדים וגם מתחי ספק וחוטטים. אין צורך להשתמש בכל הרכיבים הנ"ל.

(4 נק') ב. הסבר איך כתיבת ההוראה `out2<= not out1` לפני `end if` משפיעה על התוכנית הנ"ל.

(8 נק') ג. כתוב FULL TEST BENCH עבור התוכנית שבסעיף א'.

שאלה 7

(10 נק') א. בעזרת מכונת מצבים כתוב תכנית VHDL לתיאור המעגל העובד בצורה הבאה :

Input	rst	'1'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'
	clk	'~'	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Output	Q(0)	'0'	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'	'1'	...
	Q(1)	'0'	'0'	'1'	'1'	'1'	'1'	'0'	'0'	'0'	'0'	...
	Q(2)	'0'	'0'	'0'	'1'	'1'	'1'	'1'	'0'	'0'	'0'	...
	Q(3)	'0'	'0'	'0'	'0'	'1'	'1'	'1'	'1'	'0'	'0'	...

מבוא : clk, rst

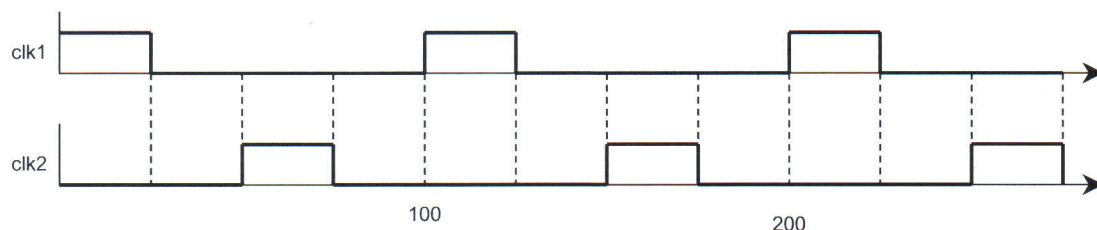
מוצא : Q (4 סיביות)

(6 נק') ב. שרטט את דיאגרמה המצבים (דיאגרמת "בועות") של המכונה.

(4 נק') ג. הסבר מהו סוג המכונה בתוכנית שלך.

שאלה 8

(10 נק') א. כתוב תהליך שאינו מיועד לסינתזה ואשר מייצר את צורות האותות clk1, clk2 כאשר מדובר בתהליך אין סופי.



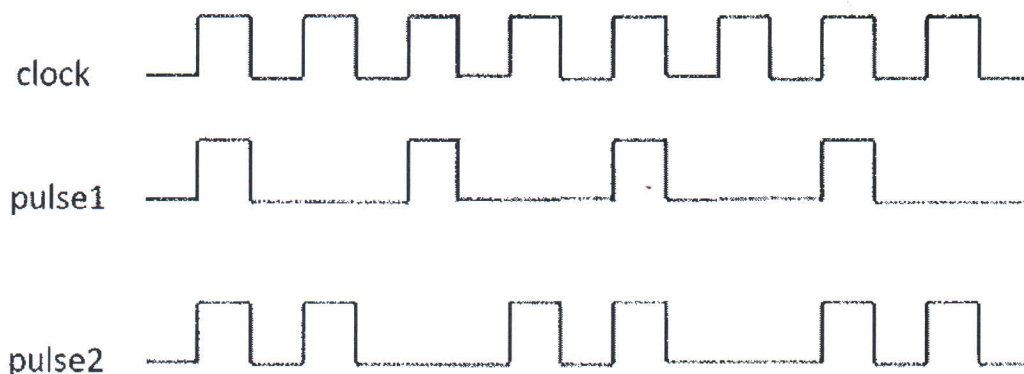
הערה: ציר הזמן הוא ב- ns

(10 נק') ב. קטע הקוד הבא מתאר מערכת שאינה אמורה לעבור סינתזה:

```
process
begin
    output <= '0';
    wait until clock'event and clock = '1';
    output <= '1';
    wait until pulse1'event and pulse1 = '0';
    output <= '0';
    wait until pulse2'event and pulse2 = '1';
    output <= '1';
    wait until clock'event and clock = '0';
end process;
```

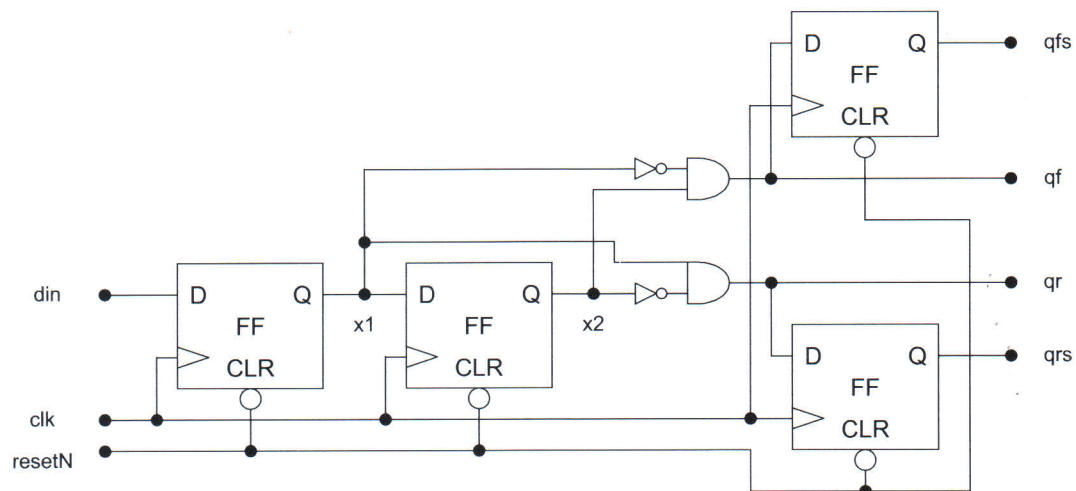
למערכת יש שלוש כניסות שנקראות clock, pulse1 ו-pulse2 והיציאה output.

העתק את דיאגרמת הזמנים הבאה למחברת הבחינות וציין את האות output.



שאלה 9

(15 נק') א. כתוב את קוד VHDL (ישות וארכיטקטורה מלאים) בצורה התנהגותית לתיאור המערכת באיור הבא :



- מבוא : din, clk, resetN
- מוצא : qfs, qf, qr, qrs

הערות

- אסור להשתמש במשתנים או אותות נוספים, פרט לאלו שמצוינים באיור.
- אסור להשתמש בשמות של האותות שרשומים בתוך הרכיבים.

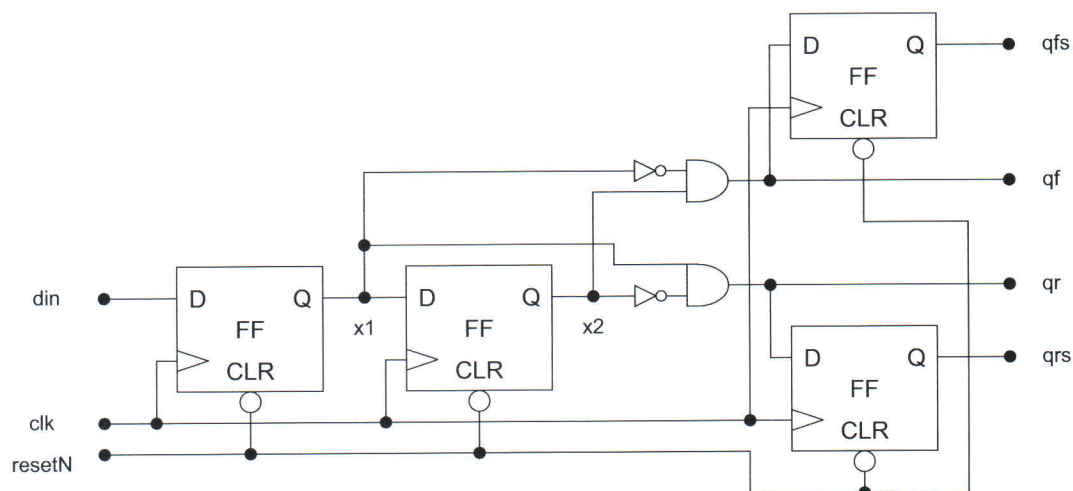
(5 נק') ב. הסבר בקצרה את ההבדל בין INOUT, OUT ו-BUFFER.
(התייחס גם ל-VHDL-2008)

שאלה 10

5 נק') א. כתוב תכנית מלאה ב-VHDL לתיאור דלגלג מסוג D העובד לפי טבלת אמת :

INPUTS			OUTPUT
CLR	CLK	D	Q
'0'	'-'	'-'	'0'
'1'	↑	'0'	'0'
'1'	↑	'1'	'1'

15 נק') ב. בעזרת הרכיב מסעיף א' כ- COMPONENT, כתוב תכנית מלאה ב-VHDL לתיאור המעגל שבאיור הבא :



- מבוא : din, clk, resetN

- מוצא : qfs, qf, qr, qrs

הערה : אסור להשתמש באותות נוספים עבור מיפוי הרכיבים.

בהצלחה !

© כל הזכויות שמורות למה"ט