



Pre3

רועי שחמון 206564759 נעם קליינר 316015411 | מעבדת ארכיטקטורה ומאיצי חומרה | 13.05.23



הגדרת תכנון המערכת :

1. Control Unit and DataPath
2. Multi Cycle CPU
3. Control unit based FSM

תרשים של המערכת :

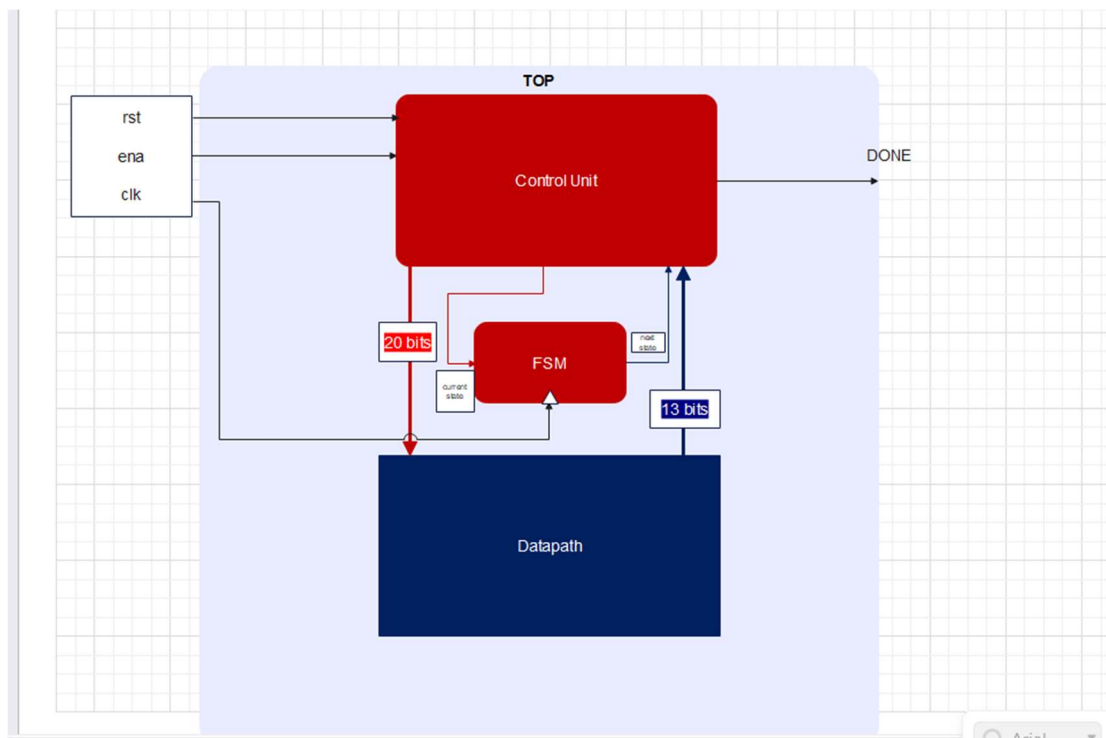


Figure 1: Graphical description for all system

תיאור המערכת באופן כללי :

המערכת מורכבת מ-2 יחידות מרכזיות. יחידת בקרה ויחידת נתיב המידע. יחידת הבקרה היא בעצם מכונת מצבים גדולה אשר המצב הבא שלה תלוי במצב הנוכחי ובאותות הכניסה (מכונת MEALY). כאשר אותות הכניסה הן אותות המצב של יחידת נתיב המידע. בהתאם למצב הנוכחי והסטטוס של המערכת יחידת הבקרה מנתבת את אותות הבקרה המתאימים אל יחידת נתיב המידע.



מכוון שליחידת נתיב המידע יש תקשורת וגישה לזיכרונות וליחידה אריתמטית לוגית, על ידי אותות הבקרה היא מממשת כראוי את ISA של המעבד. שכן המעבד עובד בארכיטקטורה . MULTI-CYCLE

תיאור הסימולציות(בדיקות)

בדיקת יחידת הבקרה

מכוון שיש הרבה פקודות חילקנו את הבדיקות לפי סוגים. התחלנו מ-I-type הרצנו את כל הסייקלים ווידאנו שקווי הבקרה מתעדכנים כמתוכנן

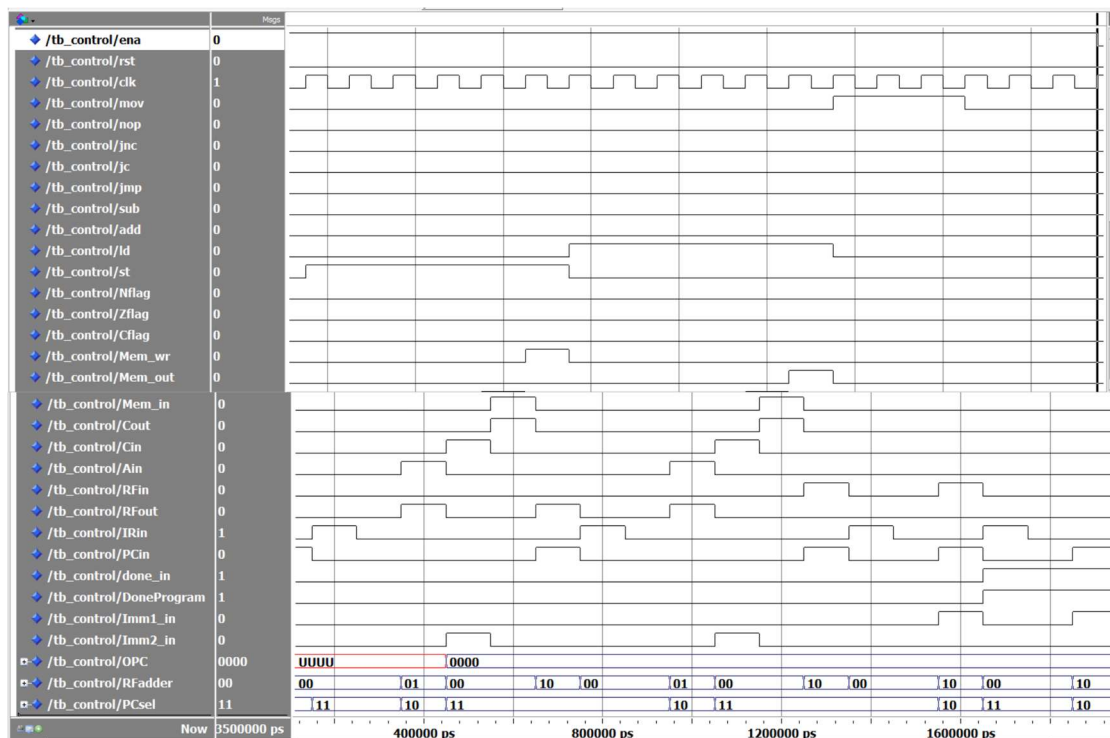


Figure 2: wave test control I-type instructions



ps	del	tb_control/ena	tb_control/jnc	tb_control/ld	tb_control/Mem_wr	tb_control/Ain	tb_control/done_in	tb_control/OPC	tb_control/rst	tb_control/jc	tb_control/st	tb_control/Mem_out	tb_control/RFin	tb_control/DoneProgram	tb_control/PCsel	tb_control/clk	tb_control/jmp	tb_control/Nflag	tb_control/Mem_in	tb_control/RFout	tb_control/Imm1_in	tb_control/mov	tb_control/sub	tb_control/zflag	tb_control/Cout	tb_control/IRin	tb_control/Imm2_in	tb_control/nop	tb_control/add	tb_control/Cflag	tb_control/Cin	tb_control/PCIn	tb_control/RFadder
0	+1	U	1	0	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
50000	+1	U	1	1	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
100000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
150000	+5	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200000	+4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
250000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
300000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
350000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
400000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
450000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
500000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
550000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
600000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
650000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
700000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
750000	+5	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
800000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
850000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
900000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
950000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1000000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1050000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1100000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1150000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1200000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1250000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1300000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1350000	+5	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1400000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1450000	+4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1500000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1550000	+4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1600000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1650000	+6	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1700000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1750000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1800000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1850000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1900000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1950000	+5	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2050000	+4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2100000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2150000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2200000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2250000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2300000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2350000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2400000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2450000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2500000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2550000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2600000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2650000	+1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2700000	+1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2750000	+1	1	0	0	1	0	0	0																									



בדיקת יחידת נתיב המידע:

בשביל לבדוק את נתיב המידע, טענו פקודות ל- program-memory וטענו מידע התחלתי ל- data-memory. בשביל לוודא שהפעולות נעשות כמו שצריך, בטסט עצמו רק שינינו את קווי הבקרה בהתאם לפקודות. כלומר שיחקנו את התפקיד של יחידת הבקרה.
הפקודות שטענו הן:

```

while counter < 15 loop
  if counter = 0 then
    TempWprog := "1001000100000001";      --ld
  elsif counter = 1 then
    TempWprog := "1001001000000111";      --ld
  elsif counter = 2 then
    TempWprog := "0000011000010010";      --add r6 <- r2+r1
  elsif counter = 3 then
    TempWprog := "0001000101100010";      --sub r1 = r6-r2
  elsif counter = 4 then
    TempWprog := "0010110000110010";      --nop
  elsif counter = 5 then
    TempWprog := "0100000000000000";      -- jmp offset 0
  elsif counter = 6 then
    TempWprog := "0101000000000001";      -- jc 1
  elsif counter = 7 then
    TempWprog := "0110000000000011";      -- jnc 3
  elsif counter = 8 then
    TempWprog := "1000111100110010";      --mov r15 <- 00110010
  elsif counter = 9 then
    TempWprog := "0001000100010010";      -- sub r1 = r1-r2
  elsif counter = 10 then
    TempWprog := "0100111111111101";      -- jmp -3
  elsif counter = 11 then
    TempWprog := "1010000100001100";      -- str [r0+12] <- r1
  elsif counter = 12 then
    TempWprog := "1010001000111100";      -- str [r3+12] <- r2
  elsif counter = 13 then
    TempWprog := "1010111100101101";      -- str [r2+13] <- r15
  elsif counter = 14 then
    TempWprog := "1010011001101111";      -- str [r6+15] <- r6
  end if;

```

figure 4: instructions

בהערות רואים את סדר הפקודות.

המידע ההתחלתי שטענו הוא

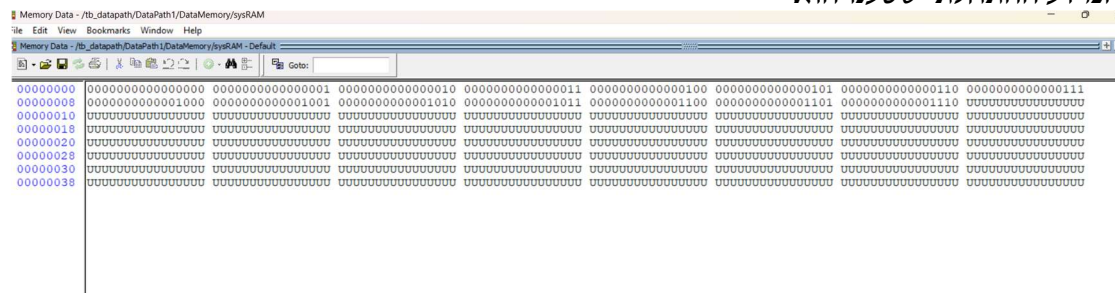


Figure 5: Data-memory initial



ניתן לראות את האותות בזמן טעינת הזיכרונות.

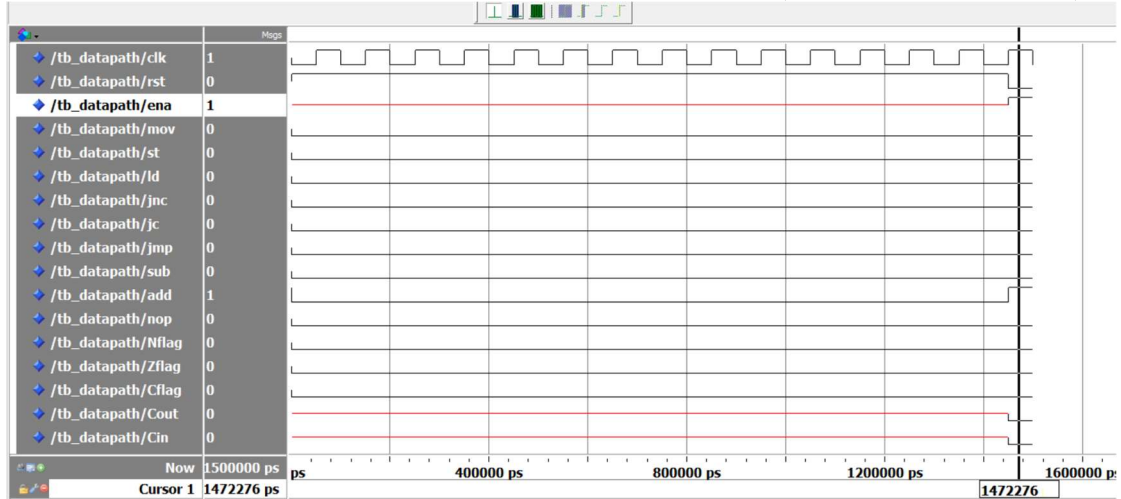


Figure 6: wave test data booting time

ניתן לראות את אותות היחידה בזמן פקודות ld ופקודת add, ומחזור אחד של פקודת sub. כלומר בזמן ריצת התוכנית.

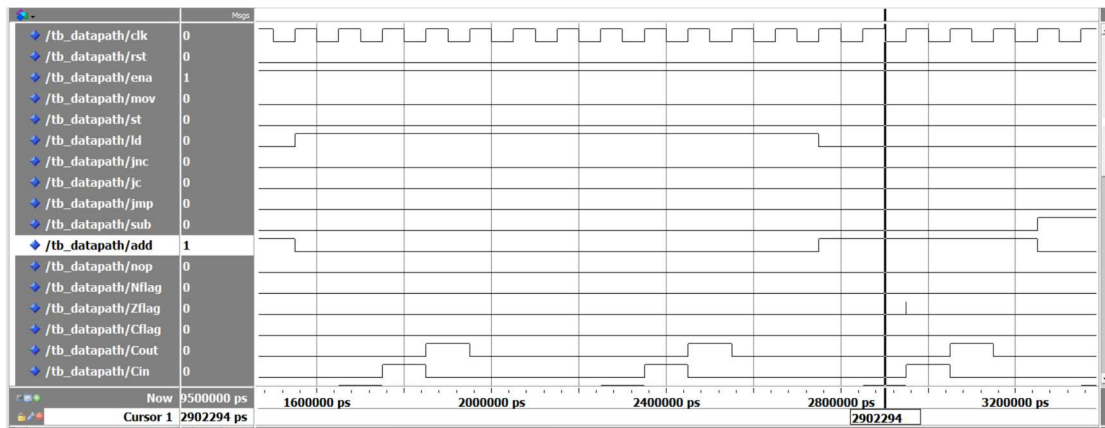


Figure 7: wave test data running time



את טבלת האמת חילקנו ל-2 חלקים (יש לה המשך שלא צירפנו בשביל לא להעמיס).
ניתן לראות בצד שמאל את רציפות הזמן בין חלקי הטבלה.

ps delta	/tb_datapath/clk_	/tb_datapath/ld_	/tb_datapath/add_	/tb_datapath/Cout_	/tb_datapath/IRin_	/tb_datapath/DoneProgram_	/tb_datapath/RFac	
	/tb_datapath/rst_	/tb_datapath/jnc_	/tb_datapath/nop_	/tb_datapath/Cin_	/tb_datapath/PCin_	/tb_datapath/Mem wr_	/tb_datapath/i	
	/tb_datapath/ena_	/tb_datapath/jc_	/tb_datapath/Nflag_	/tb_datapath/Ain_	/tb_datapath/Imm1_in_	/tb_datapath/Mem out_	/tb_dat	
	/tb_datapath/mov_	/tb_datapath/jmp_	/tb_datapath/Zflag_	/tb_datapath/RFin_	/tb_datapath/Imm2_in_	/tb_datapath/Mem in_	/tb_dat	
	/tb_datapath/st_	/tb_datapath/sub_	/tb_datapath/Cflag_	/tb_datapath/RFout_	/tb_datapath/done_in_	/tb_datapath/OPC_		
0 +2	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
50000 +4	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
100000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
150000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
200000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
250000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
300000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
350000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
400000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
450000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
500000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
550000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
600000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
650000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
700000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
750000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
800000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
850000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
900000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
950000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1000000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1050000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1100000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1150000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1200000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1250000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1300000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1350000 +2	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1400000 +1	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1450000 +3	1 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1500000 +1	0 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0

Figure 8: wave test data part A

ps delta	/tb_datapath/clk_	/tb_datapath/ld_	/tb_datapath/add_	/tb_datapath/Cout_	/tb_datapath/IRin_	/tb_datapath/DoneProgram_	/tb_datapath/RFac	
	/tb_datapath/rst_	/tb_datapath/jnc_	/tb_datapath/nop_	/tb_datapath/Cin_	/tb_datapath/PCin_	/tb_datapath/Mem wr_	/tb_datapath/i	
	/tb_datapath/ena_	/tb_datapath/jc_	/tb_datapath/Nflag_	/tb_datapath/Ain_	/tb_datapath/Imm1_in_	/tb_datapath/Mem out_	/tb_dat	
	/tb_datapath/mov_	/tb_datapath/jmp_	/tb_datapath/Zflag_	/tb_datapath/RFin_	/tb_datapath/Imm2_in_	/tb_datapath/Mem in_	/tb_dat	
	/tb_datapath/st_	/tb_datapath/sub_	/tb_datapath/Cflag_	/tb_datapath/RFout_	/tb_datapath/done_in_	/tb_datapath/OPC_		
1550000 +4	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1600000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1650000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1700000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1750000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0
1800000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0
1850000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1900000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
1950000 +5	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 1 0	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0
2000000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 1 0	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0
2050000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2100000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2150000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2200000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2250000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2300000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2350000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0
2400000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 0 0 0	0 0 0 0 0	0 0 1 0 0	0 0 0 0 0
2450000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2500000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2550000 +5	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 1 0	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0
2600000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0 0	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0
2650000 +2	1 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2700000 +1	0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2750000 +4	1 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2800000 +1	0 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2850000 +2	1 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2900000 +1	0 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 0 1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
2950000 +12	1 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 1 0 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
3000000 +1	0 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	0 1 0 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
3050000 +2	1 0 1 0 0	0 0 0 0 0	1 0 0 0 0	0 0 0 0 0	1 0 0 1 0	0 1 0 0 0	0 0 0 0 0	0 0 0 0 0

Figure 9: list test data part B

בסיום התוכנית בדקנו את המידע שהשתנה ואת תוכן ה-RF. ראינו כי המידע בשתי הרכיבים
השתנה כצפוי. בנוסף עברנו על מחזורי שעון אקראיים ובדקנו שכל האותות מוציאים ערך תקין
אל יחידת הבקרה.

Memory Data - /tb_datapath/DataPath/DataMemory/sysRAM									
File Edit View Bookmarks Window Help									
Memory Data - /tb_datapath/DataPath/DataMemory/sysRAM - Default									
Goto:									
00000000	0000000000000111	0000000000000001	0000000000000010	0000000000000011	0000000000000010	0000000000000011	0000000000000110	0000000000000011	0000000000000011
00000008	0000000000000100	0000000000000001	0000000000000010	0000000000000011	0000000000000011	0000000000000011	0000000000000011	0000000000000011	0000000000000011
00000010	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000018	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000020	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000028	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000030	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000038	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

Figure 10: data-memory final

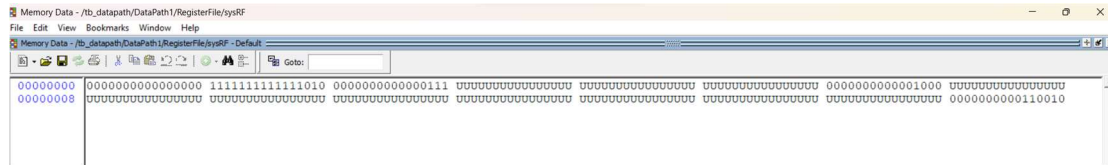


Figure 11: RF final

סיכום הסימולציה:

התוכנית שסימלצנו בעצם טוענת מהזיכרון את הערך 1 לרגיסטר 1 ואת הערך 7 לרגיסטר 8. לאחר מכן אנחנו סוכמים את שניהם ומאחסנים אותם ברגיסטר 6. כלומר את הערך 8. לאחר מכן מחסרים מרגיסטר 6 את מה שיש ברגיסטר 7 ודורסים את מה שיש ברגיסטר 1. זה נשאר הערך 1. לאחר מכן יש כל מיני קפיצות שבניהן יש פקודת movi לרגיסטר 15. לאחר מכן הערך ברגיסטר 1 משתנה שוב ל מינוס 6 בעקבות $r1=r1-r2=1-7=-6$. לבסוף טוענים את ערכים אלו לזיכרון. מקבלים התאמה בין התכנון לבין הסימולציה!

בדיקת ה-top.

לבסוף בדקנו את כל המערכת על ידי טסט מבוסס קבצים. לקחנו את הקבצים מהדרייב של המשימה, טענו אותם לזיכרונות. הרצנו את התוכנית וכתבנו בסוף את תוכן זיכרון הנתונים לקובץ ריק. קיבלנו התאמה בין מה שצפינו שיקרה לבין תוכן הקובץ.

זו התוכנית שהרצנו(לקוחה מהדרייב של המשימה)

5. Running Code Example:

Assembly code

```
1 ld r2,2(r0)
2 ld r4,4(r0)
3 ld r9,9(r0)
4 ld r13,13(r0)
5 mov r1,1
6 add r3,r2,r13
7 add r2,r4,r9
8 sub r6,r2,r3
9 jc 2
10 add r6,r1,r0
11 jmp 1
12 add r6,r0,r0
13 st r6,14(r0)
14 done
15 nop
16 jmp -2
```

Equivalent Pseudo code

```
int arr[14]={0,1,2,3,4,5,6,7,8,9,10,11,12,13}
int res=0;

void main() {

    R[3] = arr[2]+arr[13];
    R[2] = arr[4]+arr[9];

    if(R[2] >= R[3])
        res=0;
    else
        res=1;

    loop_forever;
}
```

Figure 12: code example

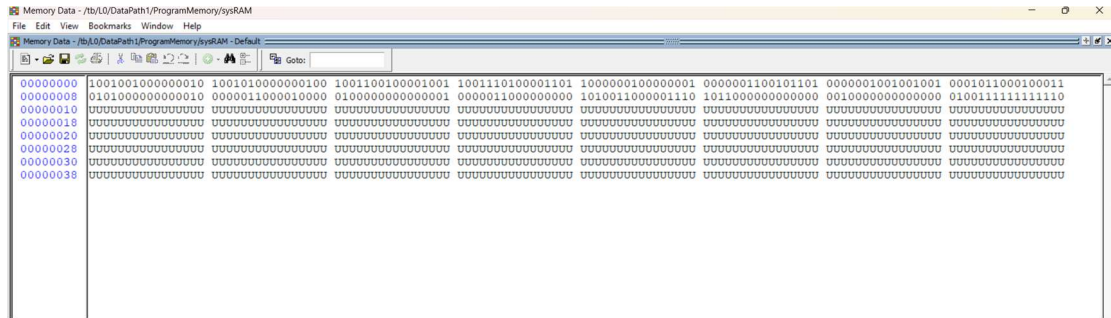


Figure 13: program memory in system

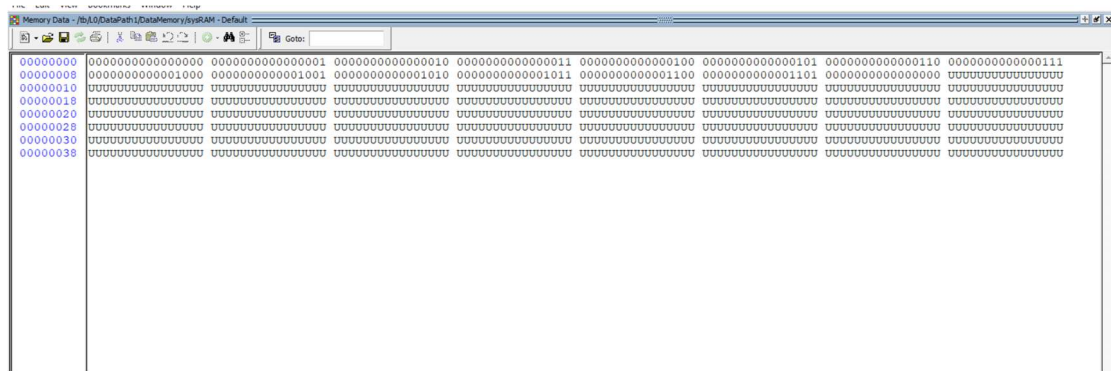


Figure 14: data memory initial in system

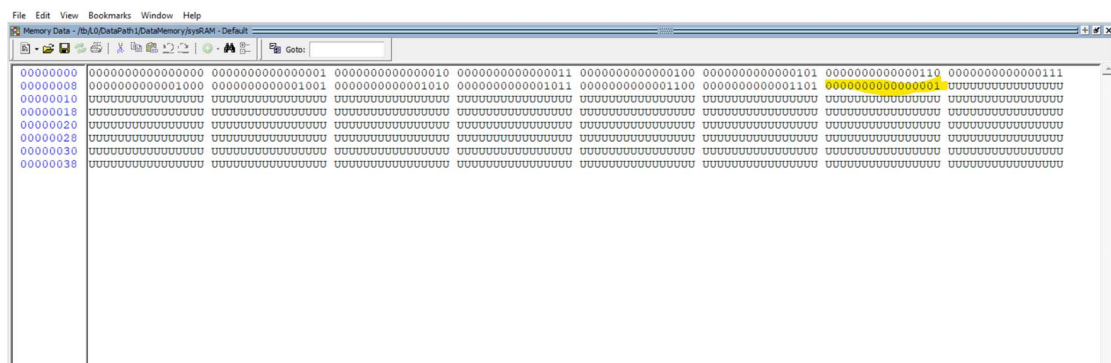
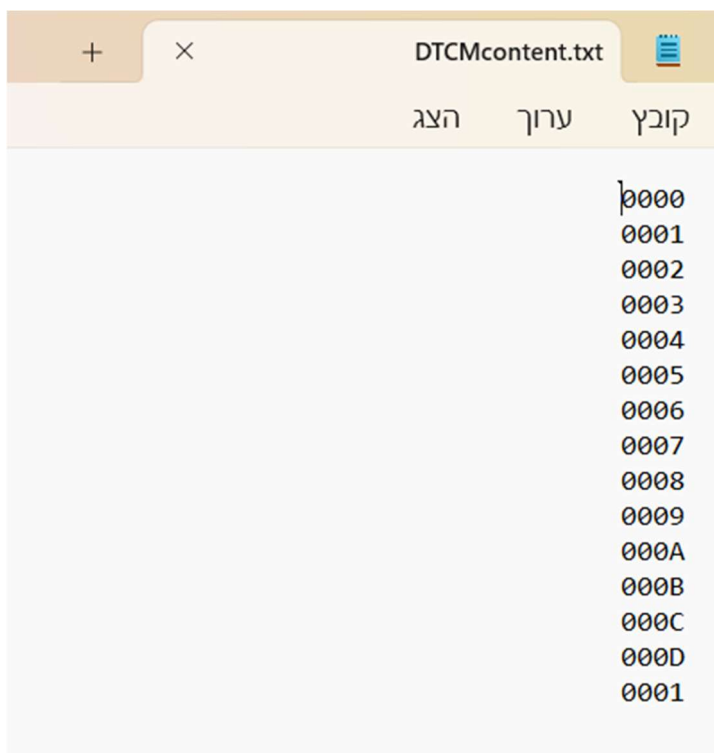


Figure 15: data memory final in system



סימנו את השינוי לאחר ריצת התוכנית



קובץ	ערוך	הצג
0000		
0001		
0002		
0003		
0004		
0005		
0006		
0007		
0008		
0009		
000A		
000B		
000C		
000D		
0001		

Figure 16: DTCMcontent

לא צירפנו תרשים גלים ורשימה של הבדיקה הזו, מכיוון שזה הרגיש לנו מיותר.
צירפנו קבצי DO.