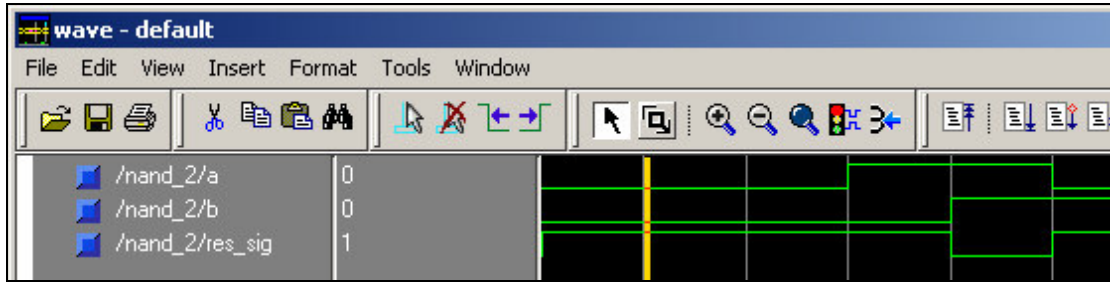


## חלק 2: מודל התנהגותי

השינוי בקוד: הביטוי not(a) הוחלף ב not(a and b).  
תוצאת הסימולציה המלאה:



## חלק 3: testbenches וסימולציה

הערה: לפני ביצוע התרגיל, קראו היטב את הוראות ההגשה המופיעות בעמוד האחרון.

### .A testbench פשוט לבדיקת הרכיב NAND

השינוי בקוד:  
השורות:

A <= '0' after 0 ns, '1' after 20 ns;  
B <= '0' after 0 ns, '1' after 20 ns, '0' after 40 ns;

שנו ל:

A <= '0' after 0 ns, '1' after 20 ns, '0' after 60 ns;  
B <= '0' after 0 ns, '1' after 20 ns, '0' after 40 ns, '1' after 60 ns;

כדי להיות תואמים לוקטור הבדיקות שנבחר:

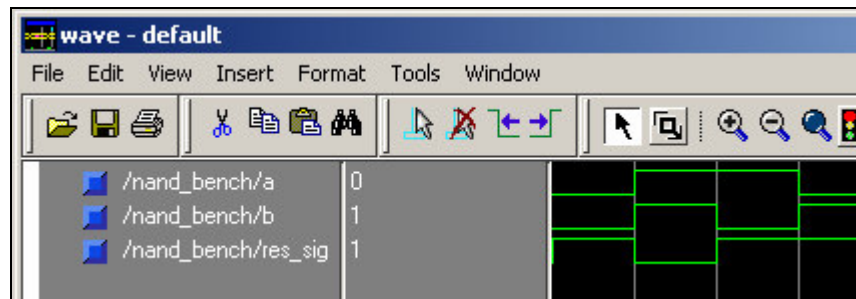
Time (ns)	0	20	40	60
B	0	1	0	1
A	0	1	1	0

השורה:

circuit: nand\_2 port map (Res\_sig, A,B);

שונתה ל:

circuit: nand\_2 port map (A, B, Res\_sig);



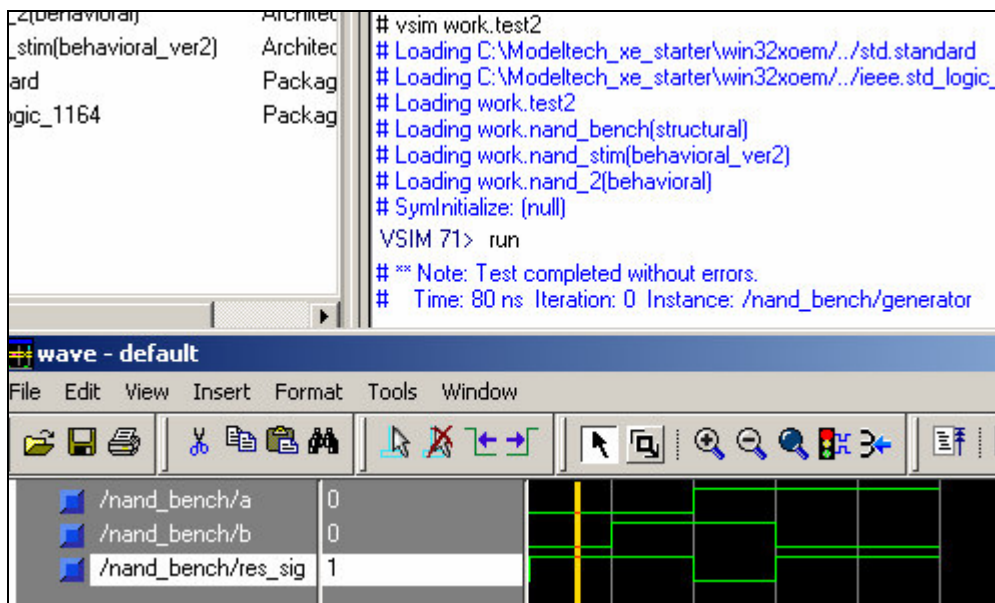
השורה

;'('1','0','0')

תוקנה ל:

;'('1','0','1')

כדי שטבלת האמת של שער ה NAND תהיה נכונה.



### חלק 4: מודל מבני של מעגלים בשפת VHDL

השורות הללו הוסרו כי אין בהם שימוש והן לא קשורות:

-- Or gate functionality (with delays)

```
ARCHITECTURE behavioral_delayed OF and_gate IS
BEGIN
    Out1 <= In1 or In2 AFTER 4 ns;
END behavioral_delayed;
```

מיפוי הסיגנלים הנכון:

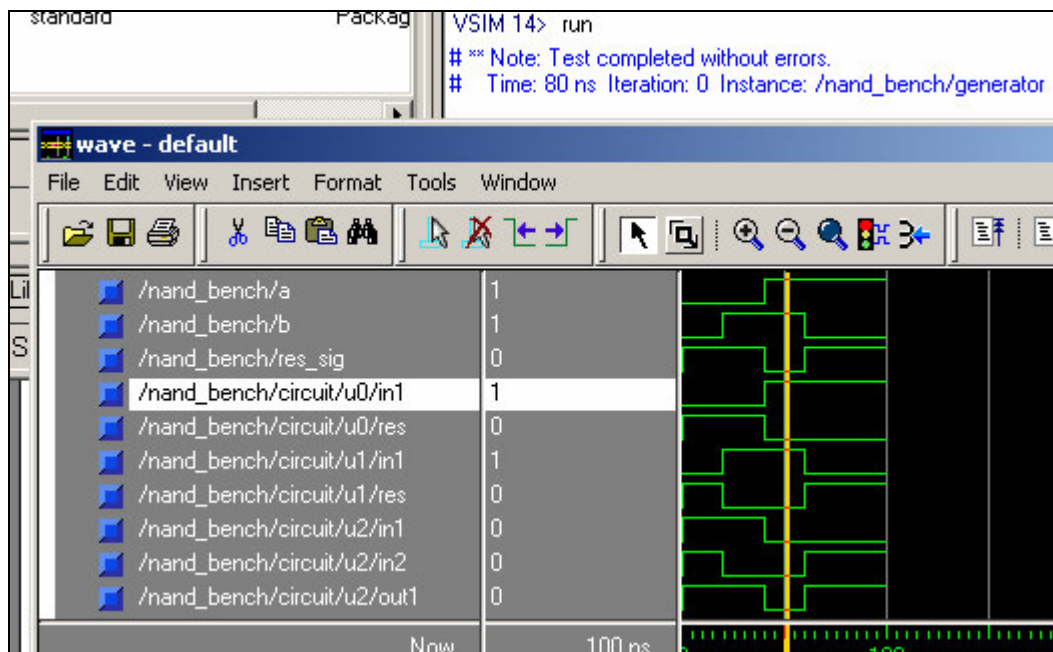
```
BEGIN
    u0: not_gate PORT MAP (A, A_not);
    u1: not_gate PORT MAP (B, B_not);
    u2: or_gate PORT MAP (A_not, B_not, Res_Sig);
END structure;
```

הסבר:

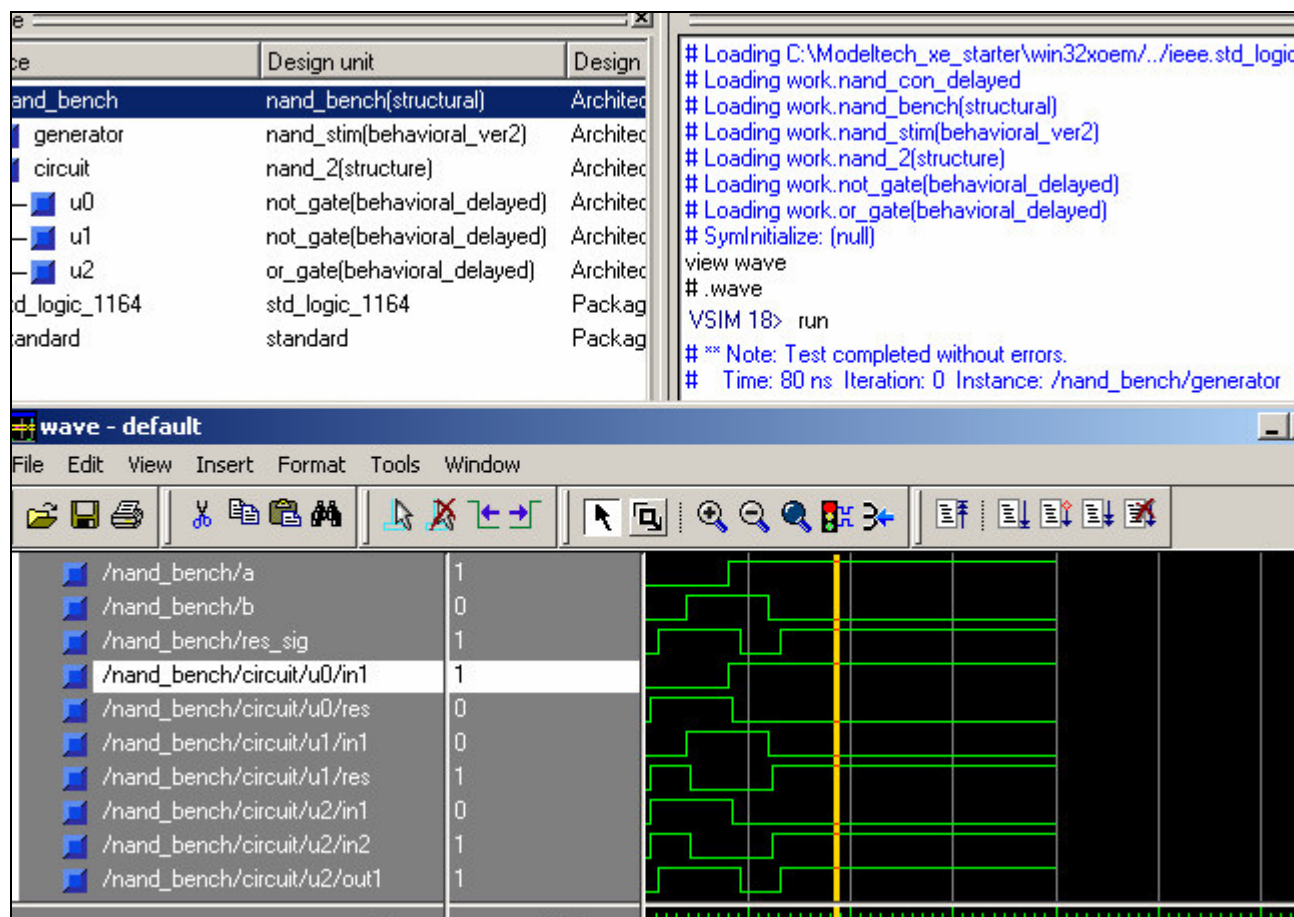
נריץ את A ו B בשער ה NOT, ואת שתי התוצאות נשים ב OR אחד, לקבלת NAND כמוצר סופי. אין צורך בסיגנלים temp\_res\_1 ו temp\_res\_2. שינויים נוספים:

- הוספת המילים behavioral\_delayed בשני סוגריים ריקים בהגדרת .nand\_con\_delayed.
- השמטת השימוש במעגל u3 שלא היה נחוץ.

הרצה ללא delay:



הרצה עם delay:



ניתן לראות את ה delay שהכנסנו משפיע על כל מוצא רלוונטי.