




מעבדה 1א' 044157

ניפוי תקלות בחומרה (DEBUG) System_Verilog

תדריך מעבדה ודו"ח סיכום
עם אמולטור למקלדת

הניסוי פותח בחסות המעבדה למערכות ספרתיות מהירות 

גרסה 3.23 – קיץ 2020

עדכנו: דודי בר-און, ארמנד שוקרון, ליאת שורץ
על פי החוברת המקורית של עמוס זסלבסקי

מועד	ביצוע עד סעיף	שם המדריך בפועל	תאריך
ביצוע הניסוי	DEBUG	אלון	20/8
השלמת חלקים חסרים - 1			
השלמת חלקים חסרים - 2			

סטודנט	שם פרטי	שם משפחה
1	ליאור	דביר
2	נועם	אילתה

הנחיות: קובץ זה הוא תדריך מעבדה וגם תבנית לדו"ח המסכם. בסוף המעבדה יש לשמור ב-PDF ולהגיש במודל.

תוכן עניינים של תדריך מעבדה ודו"ח סיכום ניפוי תקלות בחומרה (DEBUG)

3	1 לימוד מודרך של SignalTap בעזרת מונה
3	2 לימוד עצמאי של SignalTap בעזרת מכונת RANDOM
3	2.1 בדיקת ה- RANDOM על הכרטיס
4	2.2 קונפיגורציה בסיסית של ה- SignalTap
4	2.2.1 קביעת השעון, עומק הזכרון ואופן ההקלטה
5	2.2.2 קביעת האותות להקלטה ותנאי הדרבון
6	2.2.3 קביעת ההתקן לקראת צריבה
6	2.2.4 משאבי החומרה הדרושים
7	2.3 צריבת הפרויקט מתוך חלון ה- SignalTap
7	2.4 הפעלת ה- SignalTap
8	2.4.1 הפעלת ה- SignalTap במצב Autorun
9	2.5 שימוש ב- Trigger הכולל כמה תנאים
11	3 ניפוי תקלה ובנית ממשק למקלדת
11	3.1 שימוש ב- SignalTap לניפוי תקלה – ביחידה BITREC
17	3.2 בדיקת היחידה המתוקנת
20	3.3 יצירת סימבול של הממשק למקלדת
21	4 שימוש בממשק למקלדת
21	5 גיבוי העבודה
22	6 פרויקט

רשום את השעה בה התחלת את המעבדה: 8:52

1 לימוד מודרך של SignalTap בעזרת מונה

מטרה: בחלק זה של הניסוי תלמד להשתמש בכלי חשוב של Quartus לדיבוג מערכות בחומרה, נתח לוגי, ה-SignalTap. הלימוד המודרך יעשה בעזרת מונה פשוט. בניסוי זה השתמש בקובץ הארכיב שיצרת בעבודת ההכנה בבית. יש לפתוח את ה-QUARTUS COOKBOOK ולהיעזר בו במקביל.

הגדר תיקייה למעבדת DEBUG והורד אליה את קובץ הארכיב ששמרת בסוף עבודת ההכנה בבית. **פתח** את הארכיב לפרויקט.

הגדר את הקובץ [simple_up_counter.sv](#) כ-Top.

לצורך המשימה העזר גם בסרטונים של DEBUG.

פתיחת הקובץ	פתיחת הקובץ שנשמר בעבודת ההכנה
הפעלת הכלי	המונה הפשוט כ- TOP להריץ TCL, קומפילציה (להסתכל על מספר רכיבים וזכרון) פתיחת ה- SIGNAL TAP לקבוע רכיב, קובץ צריבה
קביעת פרמטרים	שעון דגימה (לקבוע clk) עומק זכרון - כמות דגימות – שיקולים (לקבוע 128) סוג דרבון (לקבוע POST TRIGGER)
בחירת סיגנלים	בחירת אותות בחירת אות בתת ספריה HEX ל RADIX קביעת TRIGGER (לקבוע ירידה ב- resetN)
הרצה	קומפילציה (להסתכל על מספר רכיבים וזכרון) שמירת STP לתיקיה צריבה מהכלי לשינוי TRIGGER - אין צורך בקומפילציה הרצה יחידה הרצה אוטומטית

2 לימוד עצמאי של SignalTap בעזרת מכונת RANDOM

מטרה: בחלק זה של הניסוי תלמד להשתמש באופן עצמאי בנתח לוגי, ה-SignalTap. הלימוד יעשה בעזרת מערכת ליצירת מספר אקראי, RANDOM.

2.1 בדיקת ה-RANDOM על הכרטיס

הגדר את הקובץ [TOP-RANDOM.bdf](#) כ-Top. בעבודת ההכנה הכרת את אופן פעולתו של [random.sv](#). הקובץ הגרפי כולל בנוסף 2 יחידות תצוגה על 7Seg. **הרץ אנליזה וסינתזה.**

קבע את ההדקים על ידי הרצת קובץ הדקים [pin.tcl](#). **הרץ קומפילציה מלאה.**

צרף לכאן צילום מסך של דו"ח קומפילציה מוצלחת של מכונת ה-RANDOM.

Flow Summary	
Flow Status	Successful - Thu Aug 20 08:59:32 2020
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	KBDINTF
Top-level Entity Name	TOP_RANDOM
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	245 / 41,910 (< 1 %)
Total registers	452
Total pins	17 / 499 (3 %)
Total virtual pins	0
Total block memory bits	640 / 5,662,720 (< 1 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

התבונן ב - Compilation Report (Summary) המפרט, בין היתר, את המשאבים בהם נעשה שימוש עבור המעגל הנתון: **Logic utilization (in ALMs)** ו- **Total block memory bits**.

רשום מספרים אלה מדו"ח הקומפילציה בטבלה להלן שבפרק 2.2.4 (בהמשך תתבקש להתייחס אליהם).

הורד את התכן לכרטיס.

בדוק שבעת לחיצה על לחצן rise נוצר מספר אקראי שמוצג על גבי ה- 7-Seg. לחץ כמה פעמים על לחצן זה וראה את שינוי המספרים.

2.2 קונפיגורציה בסיסית של ה - SignalTap

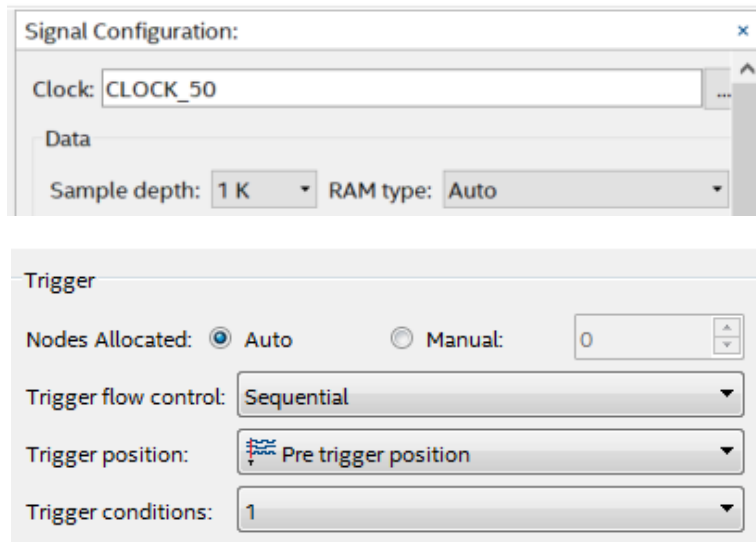
לבצוע המטלות בניסוי זה העזר בפרק "הנתח הלוגי SIGNAL TAP ב-COOK BOOK".
הפעל את הנתח הלוגי (SignalTap II Logic analyzer).

2.2.1 קביעת השעון, עומק הזכרון ואופן ההקלטה

קבע את הקונפיגורציה הבסיסית של הנתח הלוגי:

- ☐ **שעון הנתח הלוגי** – יהיה אות השעון clk של המערכת. (בדרך כלל משתמשים באות clk מכיוון שהוא מהיר יותר מכל אות אחר במערכת שלנו).
- ☐ **עומק הזיכרון של הדגימות** (Sample depth) ל - 1K דגימות.
- ☐ **מצב ה- Trigger** למצב Pre Trigger Position.

אחרי קביעות אלה חלון הקונפיגורציה צריך להיראות כך:

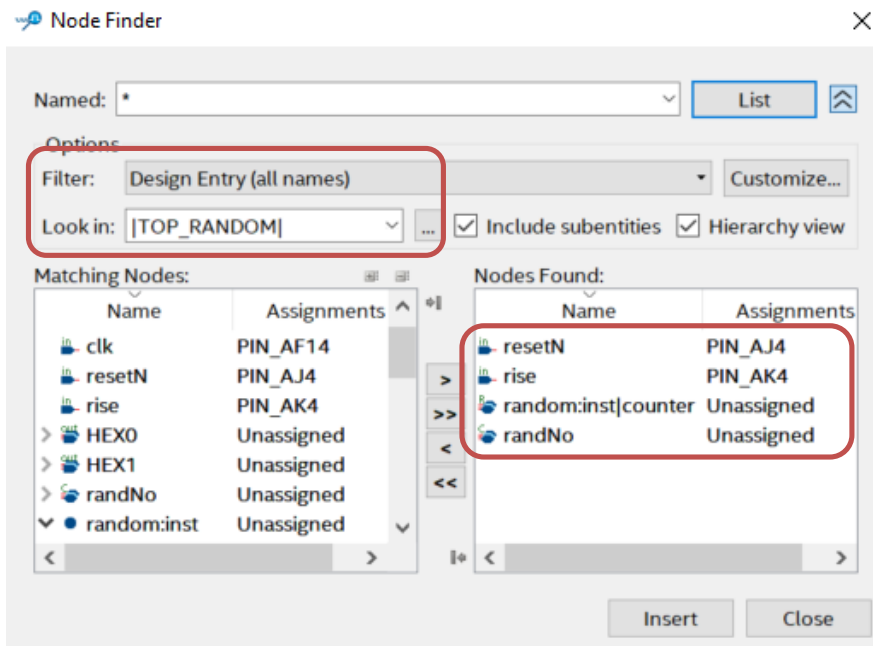


2.2.2 קביעת האותות להקלטה ותנאי הדרבון

קבע את האותות אותם רוצים להקליט עם הנתח הלוגי בחלון ה- Setup.


- ☐ resetN
- ☐ אות הכניסה rise
- ☐ וקטור הספירה count
- ☐ וקטור היציאה של מספרים אקראיים dout

הנחיה לביצוע : פתח את ה Node Finder על ידי הקלקה כפולה על חלון האותות, ותחת התפריט Filter בחרו ב"Design Entry (All Names)".



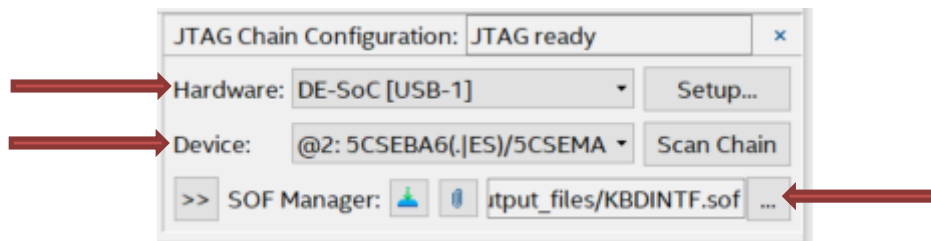
קבע את תנאי ה- Trigger (דרבון) על אחד מהאותות להקלטה : עליה באות rise.

חלון ה- Setup ייראו כך (עם האותות להקלטה ותנאי הדרבון) :

Node			Data Enable	Trigger Enable	Trigger Condition
Type	Alias	Name	18	18	1 Basic AI
in		resetN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
in		rise	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
in		random:inst counter[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh
in		random:inst counter[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh

2.2.3 קביעת ההתקן לקראת צריבה

קבע את **תצורה התקשורת עם הרכיב** (JTAG Chain Configuration) בחלק הימני העליון של חלון ה-SignalTap, את סוג ה-Device ואת הקובץ הדרוש לצריבה *.sof. בעקבות פעולה זו יתקבל חלון שנראה כך.



שמור את קובץ הנתח הלוגי בשם stp1.stp כפי שהתוכנה מציעה כברירת מחדל, תוך שיוכו לפרויקט.

2.2.4 משאבי החומרה הדרושים

קמפל שוב את הפרויקט. מלא את הטבלה ושים לב הפעם לצריכת המשאבים הגבוהה יותר.

Resource Name	Used 'without SignalTap	Used with SignalTap	Added by SignalTap
Logic utilization (in ALMs)	245	315	70
Total block memory bits	640	18,432	17,792

כיצד תחשב את תוספת הזיכרון שמוסיף הנתח הלוגי למערכת?
תשובה: נכפיל את עומק הדגימה במס הביטים שאנו דוגמים.

חשב את תוספת הזכרון. האם התוצאה הגיונית?

תשובה: כן $18 \times 1k = 18,432$

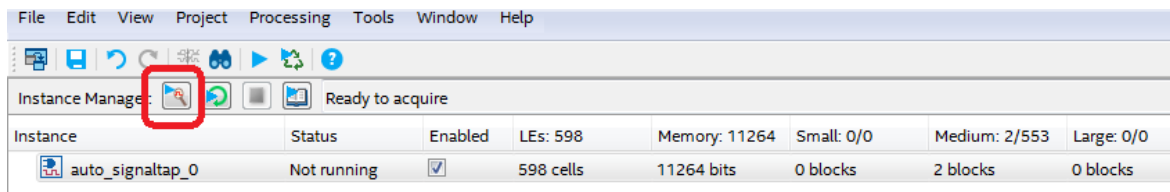
שים לב: כעת מערכת מכילה גם את הרכיבים שנוספו על ידי הנתח הלוגי וכאשר מורידים לכרטיס המערכת כוללת גם אותם.

2.3 צריבת הפרויקט מתוך חלון ה-SignalTap

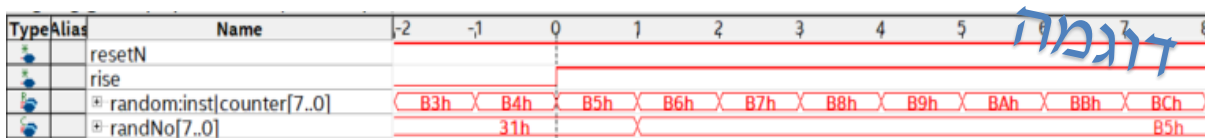
צרו את הפרויקט מתוך חלון ה-SignalTap. **העזר ב** COOK BOOK **בדוק** שהמערכת מתנהגת כנדרש, כלומר מתקבלים מספרים אקראיים כמו קודם.

2.4 הפעלת ה-SignalTap

להפעלת הנתח הלוגי, כלומר ליצירת קשר עם החמרה, הקש על מקש ה-Run Analysis.



להתחלת ההקלטה יש להפעיל את תנאי הדריבון (האות rise). כדי לראות את האותות המוקלטים על ידי ה-SignalTap עבור מ-Setup Tab ל-Data Tab. בחלון ה-Data תתקבל תצוגה דומה ל:



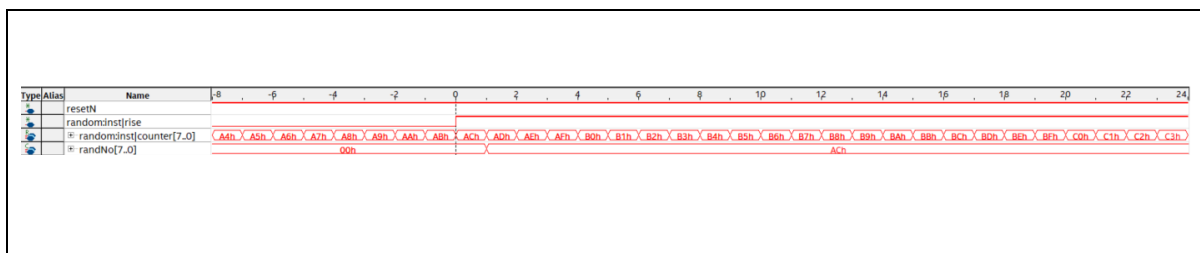
פעולת **Left-Click** ו **Right-Click** גורמות לפעולות **Zoom-In** ו **Zoom-Out** בהתאמה. **בתצוגה** שתקבל צריכים לראות בבירור את:

- ☐ העלייה באות rise (שגורמת להיווצרות ה-Trigger)
- ☐ אות המוצא שמקבל, בעלית השעון הבאה, את המספר האקראי שהיה באותו זמן במונה counter (בדוגמה זה 85h).
- ☐ הערך שהיה במוצא לפני העלייה באות rise (בדוגמה זה 31h). במעבדה כמובן שיתקבלו ערכים אחרים.

הבא את הסמן לאזור התצוגה **ובצע Zoom-In**. תתקבל תצוגה שבה רואים את המונה במפורט.

ניתן להציג את ספירת המונה גם באמצעות **אותות בודדים**. הקש על הסימן + שנמצא בצדו השמאלי שלו.

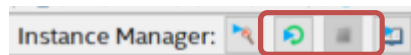
הראה למדריך את הדגימה והוסף תדפיס לדוח הסיכום שלך.



2.4.1 הפעלת ה-SignalTap במצב Autorun

להפעלה אוטומטית של הנתח הלוגי אחרי כל Trigger הקש על הכפתור הריצה האוטומטית ה-Autorun Analysis.

בהפעלה כזו כל פעם שה- Trigger מתקיים יש הקלטה. בכל לחיצה על ה-rise תתקבלנה תוצאות אחרות.



הפסק את פעולת המצב Autorun ע"י הקשה על **Stop Analysis**:

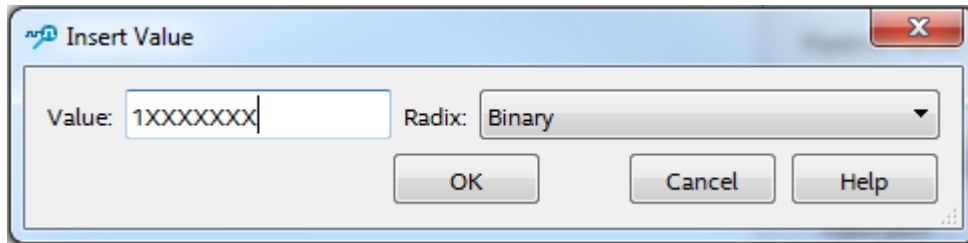
עם עצירת הנתח הלוגי שלא בעקבות Trigger כל שהוא, תתקבל תצוגה אקראית, הדומה ל:

Type	Alias	Name	-2	-1	0	1	2	3	4	5	6	7
		resetN										
		rise										
		random:inst[counter[7..0]]	F9h	FAh	FBh	FCh	FDh	FEh	FFh	00h	01h	
		randNo[7..0]										85h

רשום את השעה בה המדריך ראה את הצלחת הרכישה: 9:47

2.5 שימוש ב - Trigger הכולל כמה תנאים

הנתח הלוגי של Quartus מאפשר ליצור תנאי Trigger מורכבים יותר.
הוסף תנאי שני, שהמונה יהיה גדול מ - 128, לתנאי הדרבון הקודם:
☐ מקש ימני בחלון ה Setup על התא אותו רוצים לשנות,
☐ בחירת Insert Value ברשימה המתקבלת ושינוי ערכו.
 לאחר השינוי, חלון ה- Insert Value ייראה כך.



סימני ה - X הם Don't Care, לכן ייבחרו כל הצירופים שבהם סיבית ה - MSB היא 1 ושאר הסיביות הנמוכות יותר מהוות צרף כל שהוא (128, 129 .. 255). פעולת AND נעשית בין שני התנאים, כלומר ההקלטה תופעל כל פעם שנלחץ על rise וספירת המונה גדולה מ - 128.

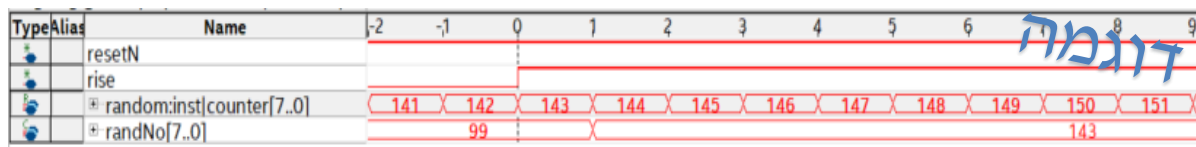
שנה את תצוגת המספרים ל - Unsigned Decimal :

☐ סמן את האות, מקש ימני, בחירת Bus Display Format ברשימה המתקבלת,

☐ ובחירת Unsigned Decimal ברשימה הנוספת.

בדוק את התנאי החדש, הפעל את המערכת במצב Autorun. שים לב שעכשיו בממוצע רק כמחצית מהלחיצות על KEY0 תהיה הקלטה.

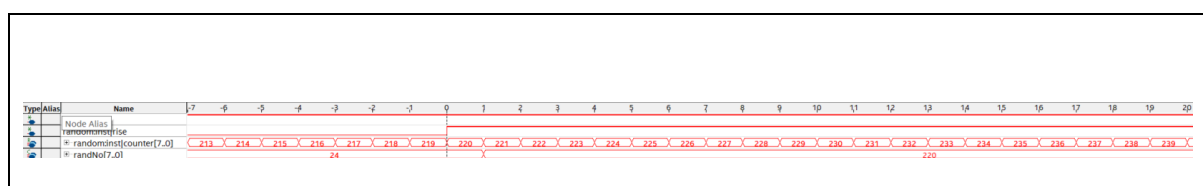
להלן דוגמה של הקלטה בה מתקיימים שני התנאים.



הוסף את תדפיס מסך ההגדרות המראה את הגדרת ה TRIGGER

trigger: 2020/08/20 09:59:50 #0		Lock mode: Allow all changes		
Type/Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
	resetN	18	18	1 Basic AN
	random:inst[rise]	✓	✓	
	random:inst[counter[7..0]]	✓	✓	1XXXXXXXb
	randNo[7..0]	✓	✓	XXXXXXXXb

הוסף לבאן תדפיס לדוגמה של ה - SignalTap.



רשום את השעה בה המדריך ראה את התוצאות: 10:09

3 ניפוי תקלה ובנית ממשק למקלדת

מטרה: בחלק זה של הניסוי תבנה ממשק למקלדת, תמצא, באמצעות הנתח הלוגי, את התקלה ששורבבה בו, תתקן את התקלה ולבסוף תבדוק שהממשק המתוקן עובד נכון.

חשוב מאוד בניסוי זה לפעול על פי ההוראות בקפידה ולבצע את הניסוי בדיוק לפי השלבים הנתונים!

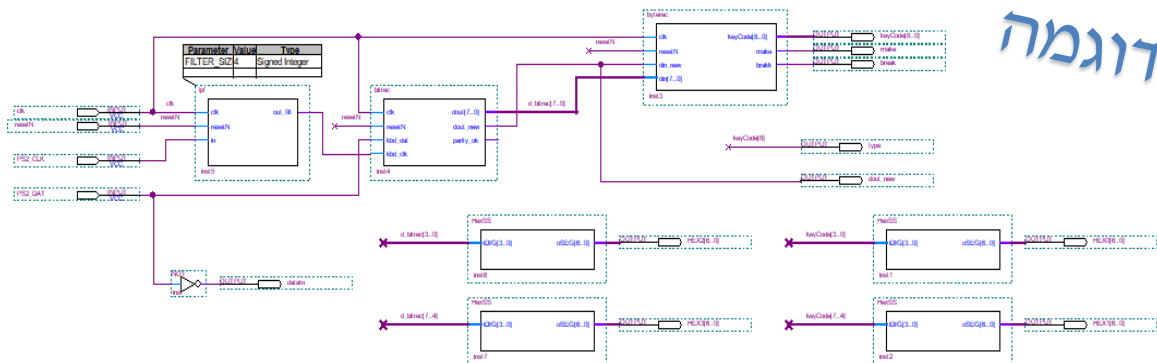
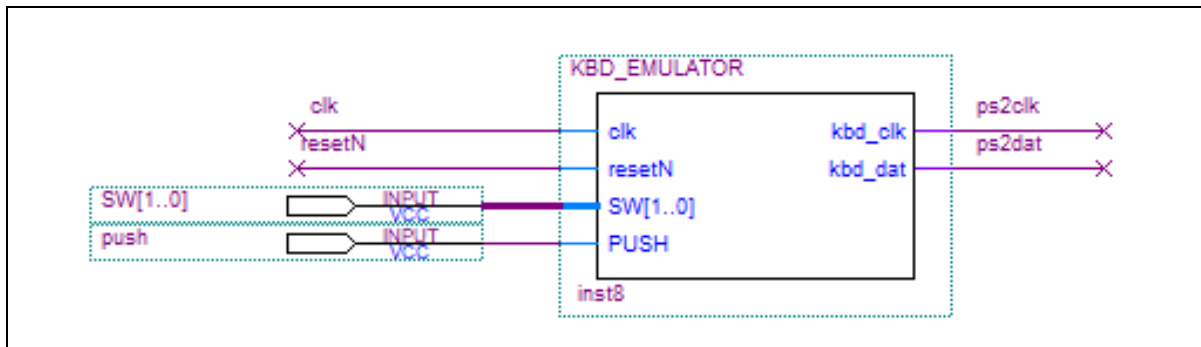
3.1 שימוש ב-SignalTap לניפוי תקלה – ביחידה BITREC

שים לב! ביחידה BITREC אותה הכנת בבית שורבבה תקלה! למרות שבסימולציה שעשית בבית היחידה עבדה נכון, צפוי שכאשר תעבוד עם מקלדת אמיתית המערכת לא תעבוד נכון!

המטרה בשלב ראשון היא למצוא את התקלה שקיימת ביחידת ה-BITREC שתכנתת בבית ולתקן אותה. לשם כך יש להשתמש בנתח הלוגי, ה-SignalTap.

פתח את קובץ הממשק למקלדת TOP_KBDINTF.bdf שנתון לך. הגדר אותו כ-TOP.

שים לב – כיוון שאין לכם מקלדת, הוספנו את מכלול KBD_EMULATOR ל-TOP



הורד את קובץ האמולטור מהמודל והוסף אותו לפרוייקט שלך.
צור עבורו סימבול ועדכן את הסימבול שנתון לך בשרטוט.

כדי לחבר את האמולטור לממשק המקלדת יש לבטל את הדקי הכניסה של המקלדת, PS2_CLK ו-PS2_DAT, ולתת את השמות המתאימים לשתי היציאות של האמולטור, ps2clk ו-ps2dat. **הרץ סינתיזה.**



הערה: כדי לראות אותות פנימיים בנתח הלוגי יש להצמיד שמות לחוטים של אותם אותות, או לחילופין לחבר הדק יציאה output לאות הרצוי.

השתמש בקובץ הפינים הנתון pin.tcl שמכיל כבר את ההקצאות. **שים לב** שהשמות שלך תואמים לשמות שקיימים בקובץ הפינים. אם לא עדכן בהתאם. **הרץ** את קובץ הפינים ובצע **קומפילציה מלאה** לתכן.

הורד את התכן לכרטיס ובדוק את התכן :

עבור האמולטור השתמש בשני מתגים ולחצן אחד של הכרטיס במקום מקלדת PS/2. **האמולטור מסוגל לדמות ארבעה "מקשים" באמצעות קומבינציה של שני מתגים, SW0 ו-SW1, ולחצן KEY1:**

1. SW0 בורר בין מקש "רווח" ומקש "חץ ימינה"
2. SW1 בורר בין מקש מסוג ישן או חדש כמתואר להלן:

SW0	SW1	KeyCode [HEX]	Bit for extended key	KeyCode [9 bit Binary]	Key on real keyboard
0	0	29	0	000101001	Space
0	1	29	1	100101001	Non-existent
1	0	74	0	001110100	 number pad
1	1	74	1	101110100	 extended

כדי להפעיל מקש מסוים יש לשים את המתגים במצב הרצוי ואז ללחוץ על לחצן KEY1 **הערה:** במהלך המעבדה יכול להיות שיהיו איזכורים למקש מקלדת כלשהו. התעלם מכך והתיחס לשווה ערך שלו באמולטור (לחצן/מתגים של הכרטיס).

בדוק את ארבעת המקשים הנתונים.

האם יש תגובה להקשה על מקשים, ציין מה מראה התצוגה של 7Seg ומה מראה הנורית האדומה שמעידה על הגעת הנתונים (dataIn). **תשובה:**

SPACE מראה 29 29 ומסומן DATAIN

EX-NONE מראה 29 ומספר לא קיים משמאל. EXTENDED וכלל לגבי DATAIN

חץ ימינה מסמן DATAIN אך אינו אינו מספר

חץ ימינה מורחב מציג DATAIN ומסמן E0 ב7SEG מצד שמאל, אך לא מספר

איזה מקש עובד נכון?

הפעל את הנתח הלוגי, ה-SignalTap II logic analyzer כדי לבדוק את פעולת הממשק למקלדת.
בצע את הקביעות הבאות:

- ☐ שעון הנתח הלוגי: אות השעון **CLOCK_50**
- ☐ עומק הזיכרון: **32K**
- ☐ מצב הדריבון: **Pre-trigger position**
- ☐ הצג את האותות הבאים ב-SignalTap

הערה: לשם קביעת האותות המוצגים, בחלון ה-Node Finder השתמש ב-**Filter** בשם **"Design entry (all names)"** ודא שהאפשרות **"Include subentities"** מסומנת.

- ☐ ps2clk - אות כניסת השעון מהאמולטור
- ☐ ps2dat - אות כניסת המידע מהאמולטור
- ☐ dout_new - אות היציאה שמודיע על מקש חדש
- ☐ d_bitrec - אות היציאה מה-BITREC – קוד מקש
- ☐ parity_ok - אות בדיקת הזוגיות
- ☐ cntr - המונה הפנימי של BITREC
- ☐ Shift_reg - הרגיסטר של BITREC

- אחרי שהכנסתם אותות אלה דרך חלון ה-Node Finder יש להוסיף את:
 - ☐ cur_st - המצבים של מכונת המצבים שדרושים לניפוי התקלה דרך Edit -> Add state machine nodes -> bitrec:inst4|cur_st (ראה גם הוראות ב-COOK BOOK)
- קבע כתנאי דריבון: התחלה של העברת הנתונים (ירידה ב-ps2dat)

Node		Data Enable	Trigger Enable	Trigger Condition	
Type	Alias	Name	31	31	1 Basic AI
		ps2clk	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		ps2dat	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		dout new	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		⊕ d bitrec[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh
		bitrec:inst4 parity ok	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		⊕ bitrec:inst4 cntr[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		⊕ ...ec:inst4 shift reg[9..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXh
		⊕ bitrec:inst4 cur st	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh

שים לב: בהצגת cur_st ניתן לעשות ZOOM IN כדי לראות מצבים שהם מאד קצרים בזמן או ללחוץ על ה-(+) כדי לפתוח אותו ולראות כל מצב בשורה נפרדת.

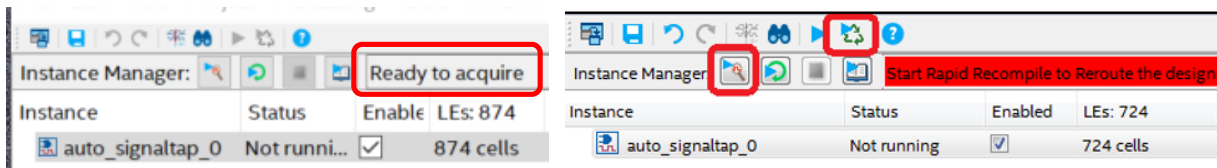
בצע את שתי הפעולות הבאות דרך ה-SignalTap:

- **קמפל** שוב (אחרי סיום האתחול של הנתח הלוגי)
- **הורד** את התכן לכרטיס.

שים לב:

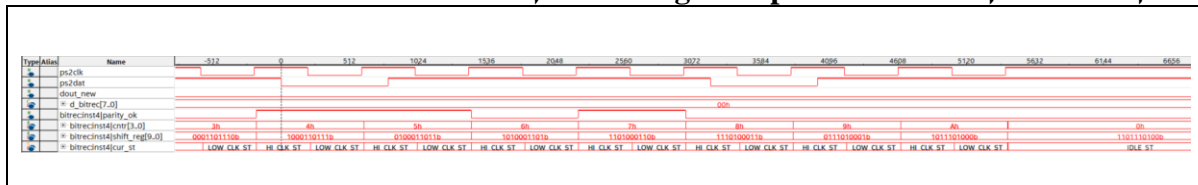
1. זמן הקופילציה הפעם הוא יותר ארוך כיוון שהתוכנית כעת כוללת גם את הנתח הלוגי.

2. היעזר בשורת ההודעות, כגון יש לקמפל, יש לצרוב לכרטיס, מוכן לדגימה וכו', ופעל לפיהן. כל עוד שורת ההודעות אדומה פעולת ההרצה לא הסתיימה או שהיא לא תקינה!

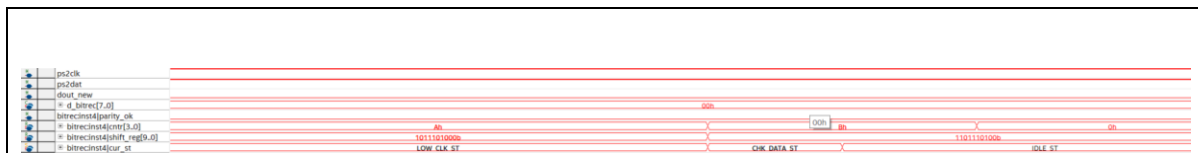


כדי לדבג את המערכת **הפעל** את ה - SignalTap (Auto Run) **ובצע הקלטה** של האותות השונים. הקש על כמה מקשים, אחד שעובד נכון ואחר שלא עובד, וראה את ההבדל בין התוצאות. התבונן באותות השונים המוצגים ב- SignalTap וחשוב כיצד תוכל להסיק מהם מה התקלה.

הוסף תדפיס מסך של הרכישה ב- SignalTap במצב תקול לדוח.



בצע ZOOM על הדגימה בסוף ה FRAME, לפני המעבר ל IDLE כך שרואים בדיוק את המעברים



מה היית מצפה שיהיה במוצא האותות dout_new ו- d_bitrec ומה המוצא בפועל?

תשובה: מצפים:

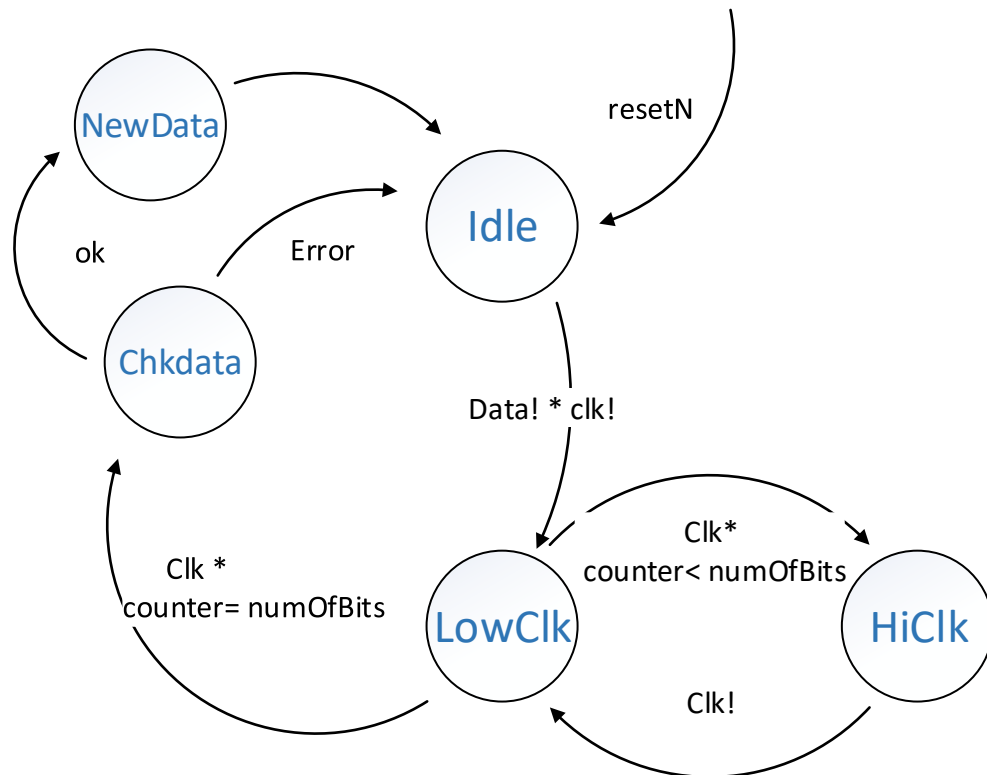
dout_new = 1'b1

d_bitrec = 8'h74

בפועל:

dout_new = 1'b0

d_bitrec = 8'h0



התבונן במצבים של מכונת המצבים. לאילו מצבים לא הגענו (ב-SignalTap אין אות מוצא)?
תשובה: לא הגענו לNEWDATA

על סמך שתי התשובות הנ"ל באיזה מצב אתה חושב שיש תקלה ומה התקלה?
תשובה: התקלה בCHKDATA והינה ב.PARITYBIT
ניתן לראות בסימולציה שPARITYBIT הינו 0 למרות שאמור להיות 1

פתח את הקוד של BITREC ונסה לזהות את התקלה. קרא למדריך ואמור לו מה התקלה.

רשום את השעה בה המדריך ראה את המערכת: 11:00

תקן את הקוד של יחידת BITREC קמפל וצורב לכרטיס.
בדוק שהמקלדת עובדת: הקש על מקשים שונים וראה שהתצוגות מראות את הקוד הנכון.

הוסף כאן את כל הקובץ המתוקן. סמן בצהוב את המקום ששיניית לתיקון התקלה.

```

module bitrec
(
    input logic clk,
    input logic resetN,
    input logic kbd_dat,
    input logic kbd_clk,
    output logic [7:0] dout,
    output logic dout_new,
    output logic parity_ok
);

enum logic [2:0] {IDLE_ST, // initial state
                  LOW_CLK_ST, // after clock low

```

[illegible]


```

        CHK_DATA_ST: begin
            if (parity_ok) begin
                nxt_st = NEW_DATA_ST;
                Next_Dout = shift_reg[7:0];
            end else
                nxt_st = IDLE_ST;
            end

        NEW_DATA_ST: begin
            nxt_st = IDLE_ST;
            dout_new = 1'b1;
        end

    endcase

end

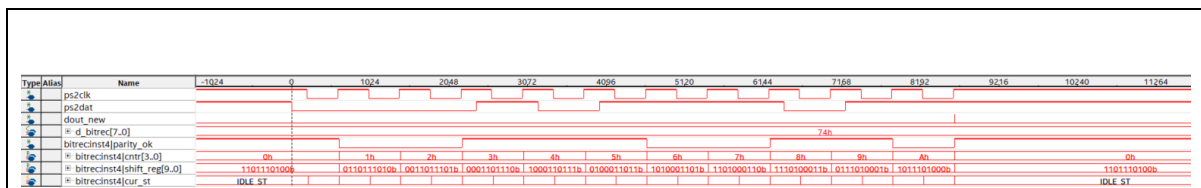
// parity Calc
assign parity_ok = shift_reg[8] ^ shift_reg[7] ^ shift_reg[6] ^
shift_reg[5]
^ shift_reg[3] ^ shift_reg[2] ^ shift_reg[4] ^ shift_reg[1] ^
shift_reg[0];

endmodule

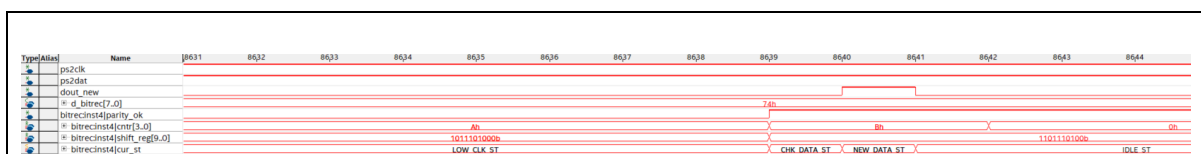
```

הפעל את הנתח הלוגי. הקש על מקש הרווח ואחרים וראה שקוד המקש המתקבל נכון.

הוסף לכאן תדפיס מסך של הרכישה במצב תקין.



בצע ZOOM על האזור המעניין



וודא שקוד המקש זהה לקוד הצפוי :

הראה את הצלחת הרכישה במעגל המתוקן למדריך המעבדה.

רשום את השעה בה המדריך ראה את המערכת:

3.2 בדיקת היחידה המתוקנת

הורד מרשימת הנתח הלוגי את האותות שהיו נחוצים לדיבוג cntr, parity_ok, ומצבי מכונת המצבים. **הוסף** את אותות המוצא של BYTEREC: make, break, ו-keyCode.

- עמוד 17 - DEBUG דוח מסכם

שנה את תנאי הדרבון להודעה על לחיצה על מקש (עליה באות make).
קבע את מצב הדרבון ל-Post-Trigger, בחלון Setup באזור Trigger.

הפעל את הנתח הלוגי במצב Autorun ולחץ של מקשים שונים. שים לב לתוצאה ב- d_bitrec וב- keyCode.

מה הקודים שמתקבלים ב- d_bitrec וב- keyCode בלחיצה על מקש right arrow או אחר מהסוג הישן ומהסוג החדש?

תשובה:

עבור ימינה ישן

D_bitrec = 74h

keycode = 074h

ימינה חדש

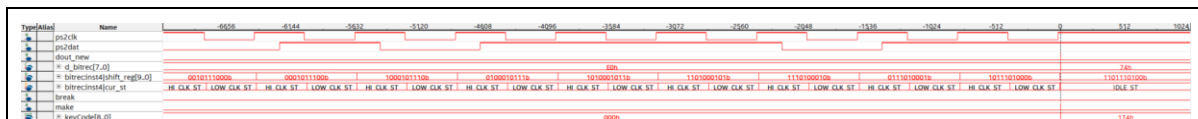
D_bitrec = E0h74h

keycode = 174h

בצע ZOOM OUT והראה את כל התווים שנקלטו מהמקלדת החל מה START BIT של התו הראשון (מצורפת חלק מהטבלה שבחומר העזר לדוגמה)

KEY	MAKE	BREAK
DOWN	E0,72	E0,F0,72
KP 2	72	F0,72

הוסף לכאן תדפיס מסך של הרכישה עם תנאי דרבון make. למקש "מסוג חדש" (בעל קוד כפול)



שנה את תנאי הדרבון לעליה באות break (המציין עזיבת מקש)

הערה: אין צורך לקמפל שוב כאשר משנים תנאי או מצב דרבון!

הפעל את הנתח הלוגי במצב Autorun ולחץ של מקשים שונים. **שים לב** לתוצאה ב- d_bitrec וב- keyCode.

מה הפעם הקודים שמתקבלים ב- d_bitrec וב- keyCode בלחיצה על מקש right arrow או אחר מהסוג הישן ומהסוג החדש?

תשובה:

עבור ימינה ישן

D_bitrec = F0h74h

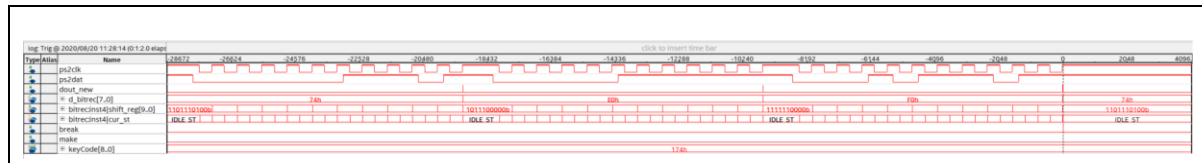
keycode = 074h

ימינה חדש

D_bitrec = E0hF0h74h

keycode = 174h

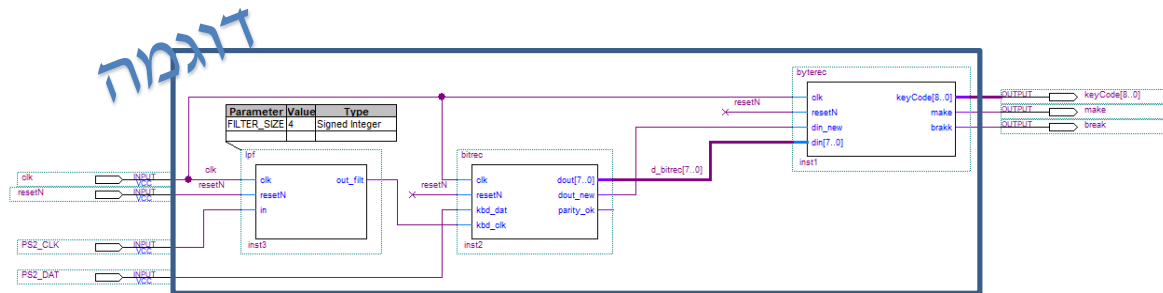
--



רשום את השעה בה המדריך ראה את המערכת: 11:30

3.3 יצירת סימבול של הממשק למקלדת

ליצירת סימבול עבור הממשק למקלדת שמור את הקובץ הגרפי בשם אחר, KBDINTF.bdf, בקובץ החדש בטל את יציאות העוזר (כולל יציאות לתצוגת 7-Segment). **נתק** גם את ה KBD EMULATOR והשאר רק את שלושת הרכיבים שמרכיבים את הממשק למקלדת. **החזר** את היציאות של חיבורי המקלדת האמיתיים, PS2_CLK ו- PS2_DAT.



צור Symbol גרפי לממשק למקלדת.
סימבול זה ישמש למעגלים ולפרויקטים בהמשך המעבדה. – אין צורך להעתיקו לדוח

רשום את השעה בה המדריך ראה את המערכת: 11:37

4 שימוש בממשק למקלדת

המטרה: להדגים שמוש בממשק למקלדת.

פתח את הקובץ הגרפי **TOP_KBD_DEMO.bdf** שעבדת עליו בעבודת ההכנה.

עדכן (Update Symbol) את הסימבול של הממשק למקלדת עם הסימבול שלך.
עדכן גם את הסימבול של האמולטור עם הסימבול המעודכן.

חבר את האמולטור לממשק המקלדת. (בטל את כניסות המקלדת PS2_CLK ו- PS2_DAT)

קמפל, הורד לכרטיס **ובדוק** עבודה תקינה:

☐ לחיצה על מקש SPACE תדליק את LEDR(0) כל זמן הלחיצה.

☐ לחיצה על מקש אחר לא תשפיע.

אפשר לשנות את אחד הקבועים של הרכיב **keyToggle decoder** לקוד מקש שממומש באמולטור, על ידי:

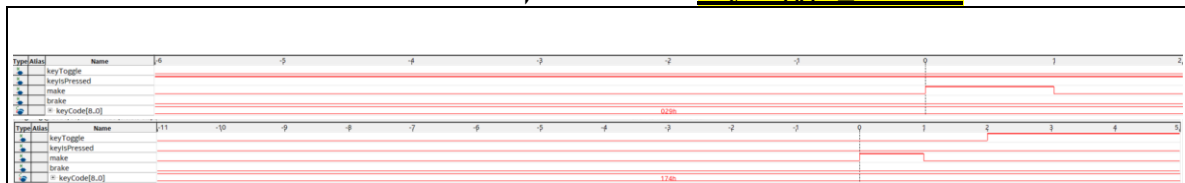
☐ מקש ימני על טבלת PARAMETER בשרטוט, בחירת parameter → properties

☐ מקש ימני על value ושינוי ערכו לקוד המקש המבוקש.

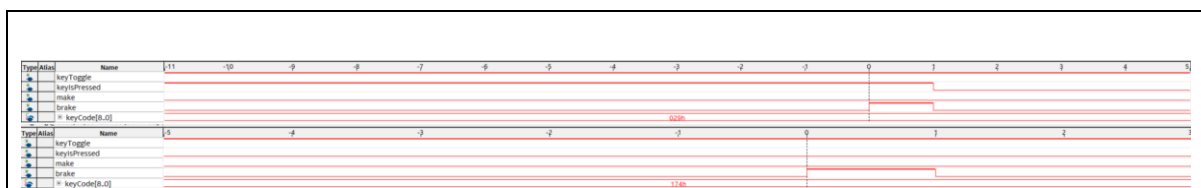
רשום את המקש שנקבע ולאיוזו פעולה (keyIsPressed או keyToggle).

מקש	קוד ב- HEX	פעולה
SPACE	29h	keyIsPressed
Extended SPACE	129h	keyToggle

הפעל את הנתח הלוגי על **TOP_KBD_DEMO.bdf** והוסף אותות לפי הצורך. הצג את שתי היציאות של רכיב **keyToggle decoder** עם תנאי דרבון **make**.



הצג את שתי היציאות עם תנאי דרבון **break**.



הראה את הצלחת המשימה למדריך המעבדה שלך.

רשום את השעה בה המדריך ראה את המערכת: 12:00

5 גיבוי העבודה

שמור את הפרויקט רגיל וגם כארכיב (Project -> Archive Project).

תגובה את קובץ הארכיב וגם העלה אותו למודל למקום המתאים.

שמור וגובה את הדו"ח שלך רגיל.

שמור את הדוח גם כ- PDF והעלה אותו למודל.

6 פרויקט

חלוקת הנושאים :

נושא הפרוייקט	בתקווה <i>Witcher 3: WILD HUNT</i>
---------------	------------------------------------

רשום את השעה בה סיימת את המעבדה : 12:08