

מעבדה בהנדסת חשמל
1א' 044157

ניסוי DEBUG - ניפוי תקלות בחומרה

תדריך מעבדה ודוח סיכום

גרסה 3.51
אביב תשפ"ד 2024

מועד	ביצוע עד סעיף	שם המדריך בפועל	תאריך
ביצוע הניסוי			
השלמת חלקים חסרים			

סטודנט	שם פרטי	שם משפחה
1		
2		

תוכן עניינים של תדריך מעבדה ודוח סיכום

DEBUG - ניפוי תקלות בחומרה

3	1 לימוד מודרך של SignalTap בעזרת מונה
4	2 לימוד עצמאי של SignalTap בעזרת מכונת RANDOM
4	2.1 בדיקת ה-RANDOM על הכרטיס
4	2.2 קונפיגורציה בסיסית של ה-SignalTap
5	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
8	2.3 צריבת הפרויקט מתוך חלון ה-SignalTap
8	2.4 הפעלת ה-SignalTap
9	2.4.1 הפעלת ה-SignalTap במצב Autorun
10	2.5 שימוש ב-Trigger הכולל כמה תנאים
12	3 ניפוי תקלה ובנית ממשק למקלדת
12	3.1 שימוש ב-SignalTap לניפוי תקלה – ביחידה BITREC
17	3.2 בדיקת היחידה המתוקנת
18	3.3 יצירת סימבול של הממשק למקלדת
19	4 שימוש בממשק למקלדת
20	5 גיבוי העבודה

רשום את השעה בה התחלת את המעבדה:

1 לימוד מודרך של SignalTap בעזרת מונה

מטרה: בחלק זה של הניסוי תלמד להשתמש בכלי חשוב של Quartus לדיבוג מערכות בחומרה, נתח לוגי, באנגלית SignalTap. הלימוד המודרך יעשה בעזרת מונה פשוט.

במעבדה:

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.

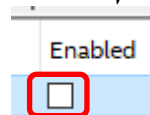
הערה: יש לפתוח את ה- **QUARTUS COOKBOOK** בפרק על הנתח הלוגי ולהיעזר בו במקביל. כמו כן, **להעשרה** ניתן להיעזר גם בסרטונים שבמודל (במיוחד בסרטונים 1-3):

סרטונים להבהרת השימוש בנתח הלוגי, ה- Signal Tap 

בתרגיל זה המדריך ירכז את תשומת הלב של הסטודנטים וכולם ביחד, כל זוג בעמדתו על המחשב שלו, יבצעו את הפעולות על פי הנחיות המדריך, כך שבאופן מרוכז יעברו פעולה פעולה. שלב זה יתבצע פרונטלי, **לא** עם סרטונים.

פתיחת הקובץ	פתיחת קובץ הפרויקט שנשמר בעבודת ההכנה וגם QUARTUS COOKBOOK קביעת המונה הפשוט כ- TOP להריץ סינתזה, TCL, קומפילציה מלאה (להסתכל על מספר רכיבים וזכרון)
הפעלת הנתח הלוגי	פתיחת ה- SIGNAL TAP קביעת רכיב (השני), הגדרת קובץ צריבה
קביעת פרמטרים	שעון דגימה (לקבוע clk) עומק זכרון - כמות דגימות – שיקולים (לקבוע 64) סוג דרבון (לקבוע POST TRIGGER)
בחירת אותות לדגימה	בחירת אותות בחירת אות בתת ספריה HEX ל RADIX קביעת TRIGGER (לקבוע ירידה ב- resetN)
הרצה	קומפילציה מלאה (להסתכל על מספר רכיבים וזכרון) שמירת קובץ STP בתיקיה הראשית (לא ב- rtl) צריבה מחלון הנתח הלוגי הרצה יחידה והרצה מחזורית
שינוי דרבון	אחרי שינוי TRIGGER אין צורך בקומפילציה מלא: לשנות לעליה ב- resetN ול- PRE-TRIGGER

הערה: אם פתחת קובץ של הנתח הלוגי מסוג STP בזמן ההדגמה, יש למחוק אותו מהפרויקט ומהדיסק לפני שממשיכים לסעיפים הבאים. לחילופין, ניתן



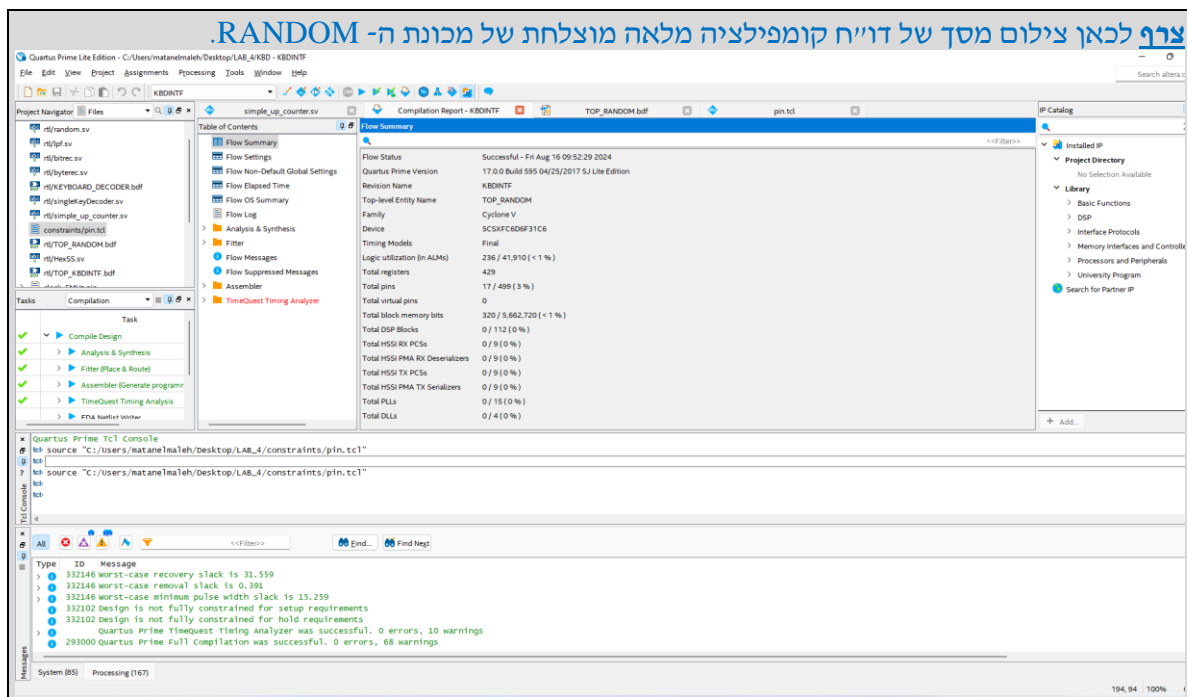
לבטל את האפשרות שלו בחלון הנתח הלוגי. ראה הסבר מפורט מאד בפרק "השבתת ה- SIGNAL TAP" שב- QUARTUS COOKBOOK.

2 לימוד עצמאי של SignalTap בעזרת מכונת RANDOM

מטרה: בחלק זה של הניסוי תלמד להשתמש באופן עצמאי בנתח הלוגי, 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**.



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

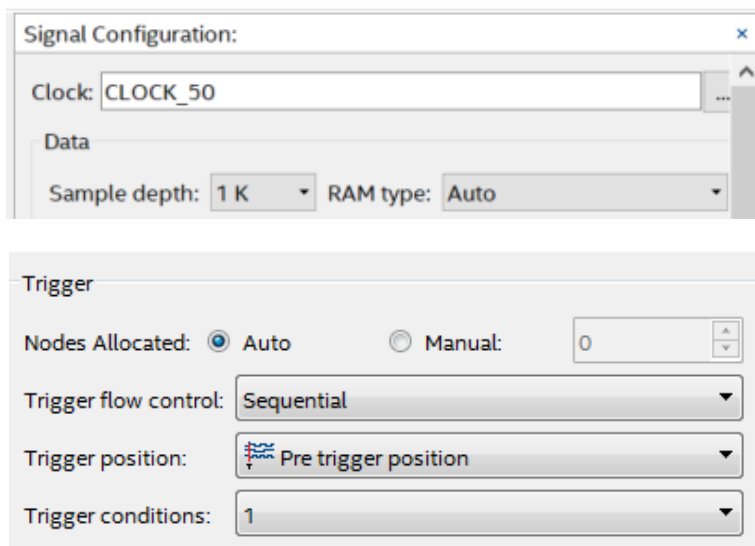
לבצוע המטלות בניסוי זה העזר בפרק " הנתח הלוגי SIGNAL TAP ב-COOK BOOK. **הפעל** את הנתח הלוגי (SignalTap II Logic analyzer -> Tools).

2.2.1 קביעת השעון, עומק הזכרון ואופן ההקלטה

קבע את הקונפיגורציה הבסיסית של הנתח הלוגי :

- ☐ שעון הנתח הלוגי – יהיה אות השעון CLOCK_50 של המערכת. (בדרך כלל משתמשים באות השעון 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 :

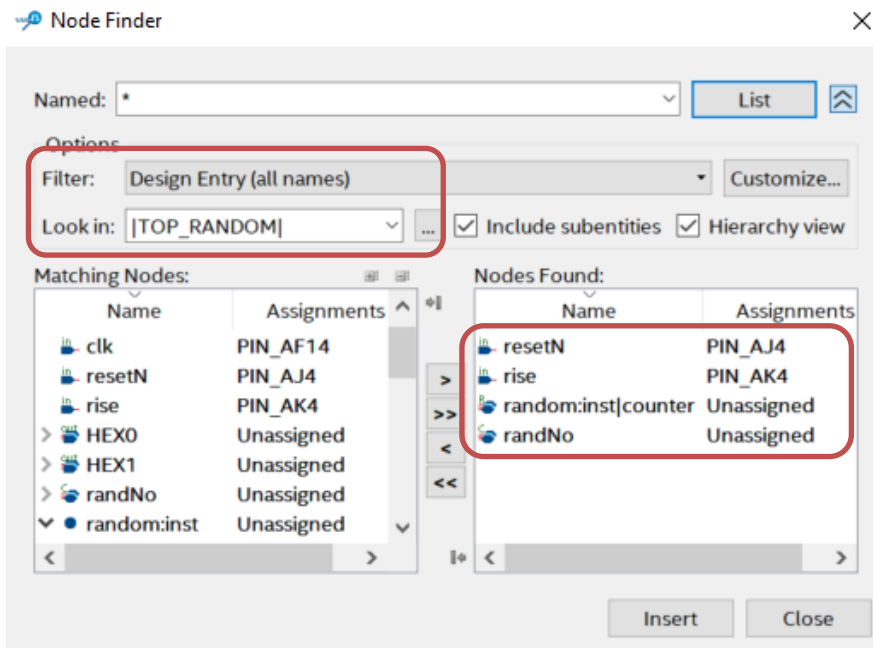


2.2.2 קביעת האותות להקלטה ותנאי הדרבון

קבע את האותות אותם רוצים להקליט בחלון ה- Setup.

- ☐ resetN
- ☐ אות הכניסה rise
- ☐ וקטור הספירה counter (מתוך המודול random.sv)
- ☐ וקטור היציאה של מספרים אקראיים randNo

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



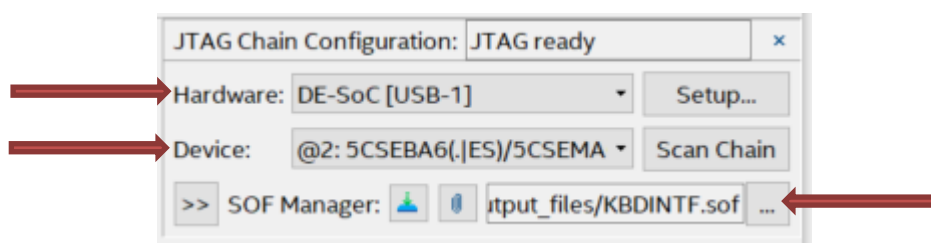
קבע את תנאי ה- Trigger (דברון) על אחד מהאותות להקלטה: **עליה באות rise**.

חלון ה- Setup (עם האותות להקלטה ותנאי הדברון):

Type	Alias	Node Name	Data Enable	Trigger Enable	Trigger Condition
in		resetN	✓	✓	Basic AI
in		rise	✓	✓	✓
+		random:inst counter[7..0]	✓	✓	XXh
+		randNo[7..0]	✓	✓	XXh

2.2.3 קביעת ההתקן לקראת צריבה

קבע את תצורת התקשורת עם הרכיב (JTAG Chain Configuration) בחלק הימני העליון של חלון הנתח הלוגי, 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

Instance	Status	Enabled
auto_signaltap_0	Not running	✓

ודא שהקובץ מאופשר (V ב- Enabled):

קמפל שוב את הפרויקט.

מלא את הטבלה ושים לב לצריכת המשאבים הגבוהה יותר.

Resource Name	Used without SignalTap	Used with SignalTap	Added by SignalTap
Logic utilization (in ALMs)	236	314	78
Total block memory bits	0	18432	18432

חשב את תוספת הזכרון שמוסיף הנתח הלוגי למערכת. הסבר מאיפה נובע הפרש זה.

חישוב: 18432

הסבר: לפני כן לא נדרשנו לזכרון, לא היינו צריכים לשמור מידע.

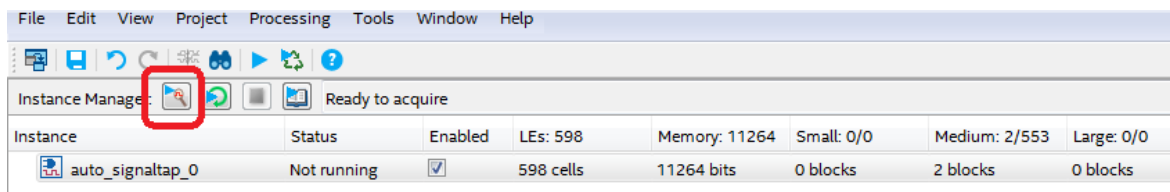
שים לב: כעת מערכת מכילה גם את הרכיבים שנוספו על ידי הנתח הלוגי וכאשר מורידים לכרטיס המערכת כוללת גם אותם.

2.3 צריבת הפרויקט מתוך חלון ה-SignalTap

צריב 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:

TypeAlias	Name	-2	-1	0	1	2	3	4	5	6	7	8
	resetN											
	rise											
	random:inst[counter[7..0]]	B3h	B4h	B5h	B6h	B7h	B8h	B9h	BAh	BBh	BCh	
	randNo[7..0]		31h								B5h	

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

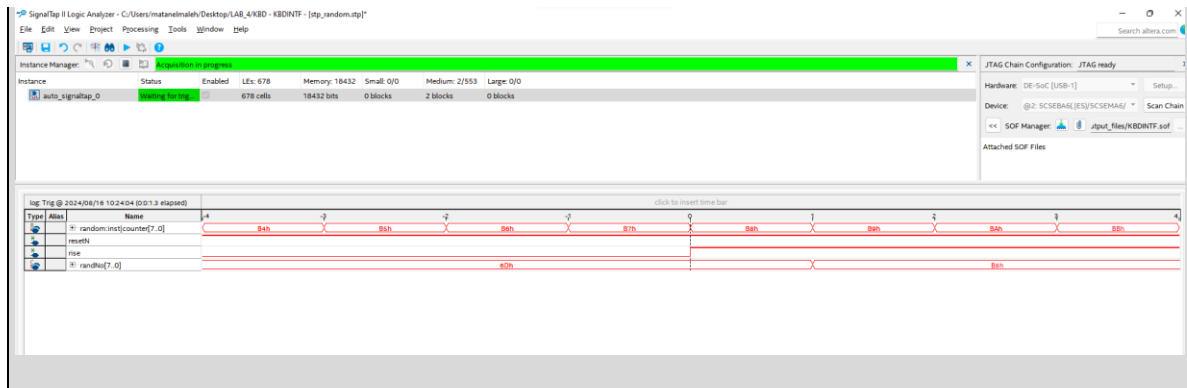
- ☐ העלייה באות rise (שגורמת להיווצרות ה-Trigger)
- ☐ אות המוצא שמקבל, בעלית השעון הבאה, את המספר האקראי שהיה באותו זמן במונה counter (בדוגמה לעיל זה 85h).
- ☐ הערך שהיה במוצא לפני העלייה באות rise (בדוגמה לעיל זה 31h). במערכת שלך כמובן שיתקבלו ערכים אחרים.

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.

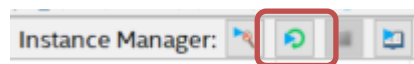
הראה למדריך את הדגימה.

הוסף תדפיס לדוח הסיכום שלך.



2.4.1 הפעלת ה-SignalTap במצב Autorun

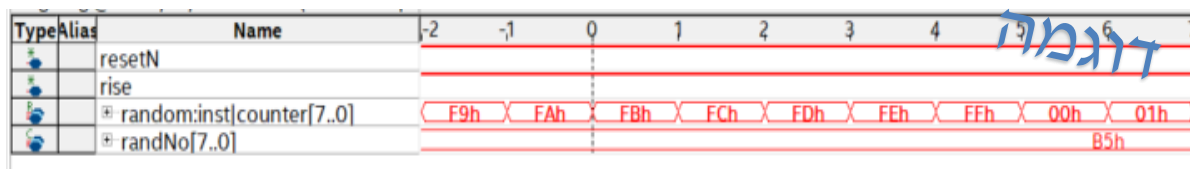
להפעלה אוטומטית של הנתח הלוגי אחרי כל Trigger הקש על הכפתור הריצה האוטומטית ה-Autorun Analysis.



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**:

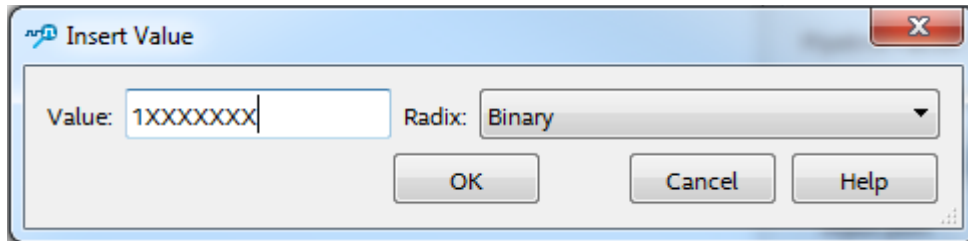
עם עצירת הנתח הלוגי שלא בעקבות Trigger כל שהוא, תתקבל תצוגה אקראית, הדומה ל:



רשום את השעה בה המדריך ראה את הצלחת הרכישה: 10:26

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 במחצית מהלחיצות על KEY0 תהיה הקלטה.

להלן דוגמה של הקלטה בה מתקיימים שני התנאים.

Type	Alias	Name	-2	-1	0	1	2	3	4	5	6	7	8	9
		resetN												
		rise												
		random:inst[counter[7..0]]	141	142	143	144	145	146	147	148	149	150	151	
		randNo[7..0]		99								143		

הוסף את תדפיס מסך ההגדרות המראה את הגדרת ה- TRIGGER.

trigger: 2024/08/16 10:29:45 #0				Lock mode: <input checked="" type="checkbox"/> Allow all changes	
Type	Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
R		random:inst[counter[7..0]]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 <input checked="" type="checkbox"/> Basic AND 1XXXXXXXb
*		resetN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
*		rise	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
C		randNo[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXb

הוסף לכאן תדפיס לדוגמה של ה- SignalTap.

3 ניפוי תקלה ובנית ממשק למקלדת

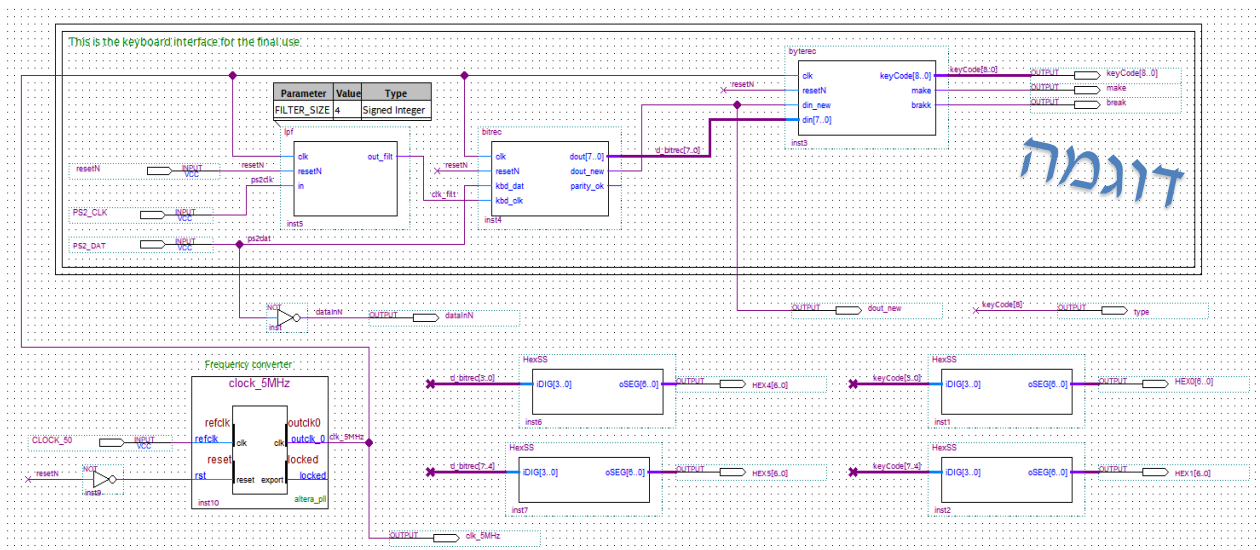
מטרה: בחלק זה של הניסוי תבנה ממשק למקלדת, תמצא באמצעות הנתח הלוגי, את התקלה ששורבבה בו, תתקן את התקלה ולבסוף תבדוק שהממשק המתוקן עובד נכון.

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!

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



הערה: במעגל זה משתמשים ברכיב מחלק תדר של תדר ה- FPGA (50MHz) ל 5MHz כדי להאט את קצב הדגימה; כך נוכל לראות את כל הדגימות של מקש (מקסימום 3 מילים) בזכרון המקסימלי 128K של הנתח הלוגי. היציאה של רכיב זה היא שעון של המערכת, לכן אפשר להשתמש בו כשעון איטי יותר עבור הנתח הלוגי.

הרץ סינתזה.

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.

הרץ את קובץ הפינים ובצע קומפילציה מלאה לתכן.

הורד את התכן לכרטיס **ובדוק** אותו.

השתמש במקלדת המספרים PS/2 שקיבלת בערכה (ראה הסבר בחומר הרקע).

במקלדת המספרים ישנם 18 מקשים , 2 מהסוג המורחב ("Enter" , "/") וכל השאר רגילים.

בדוק ארבעה מקשים , לפחות 2 רגילים לבחירתך ו- 2 מורחבים.

מה התגובה לכל אחד מהמקשים? ציין מה מראה התצוגה של 7Seg ומה מראה הנורית האדומה

(ראה איזו נורית בדיוק) שמעידה על הגעת הנתונים (dataIn).

תשובה: נורית שמעידה על extended ב enter ולא ב back_slash.

האם יש מקשים שעובדים נכון, ואם כן איזה?

תשובה: כל המקשים פרט ל 4,8,6 ו back_slash.

הפעל את הנתח הלוגי, ה- SignalTap II logic analyzer כדי לבדוק את פעולת הממשק למקלדת.
קבע את הפרמטרים הבאים:

☐ שעון הנתח הלוגי: אות מחלק התדר – 5MHz

☐ עומק הזיכרון: 8K

☐ מצב הדרבון: 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)" וודא שהאפשרות "Include subentities" מסומנת.

PS2_CLK	○	- אות כניסת השעון מהמקלדת
PS2_DAT	○	- אות כניסת המידע מהמקלדת
dout_new	○	- אות היציאה שמודיע על מקש חדש
d_bitrec	○	- אות היציאה מה- BITREC – קוד מקש
parity_ok	○	- אות בדיקת הזוגיות - אות מה- BITREC
shift_reg	○	- הרגיסטר של מכונת המצבים – אות פנימי של BITREC
cntr	○	- המונה של מכונת המצבים – אות פנימי של BITREC

• אחרי שהכנסתם אותות אלה דרך חלון ה- Node Finder יש להוסיף את המצבים של מכונת המצבים:

○ SMbits - המצבים של מכונת המצבים של bitrec

באופן הבא:

Edit -> Add state machine nodes -> bitrec:inst4| SMbits

(ראה גם הוראות ב - COOK BOOK)

• קבע **כתנאי דרבון**: התחלה של העברת הנתונים (ירידה ב- ps2dat)

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	31	31	1 <input checked="" type="checkbox"/> Basic AND
		PS2_CLK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		PS2_DAT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		dout_new	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		bitrec:inst4 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 shift_reg[9..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXh
		bitrec:inst4 cntr[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		bitrec:inst4 SMbits	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh

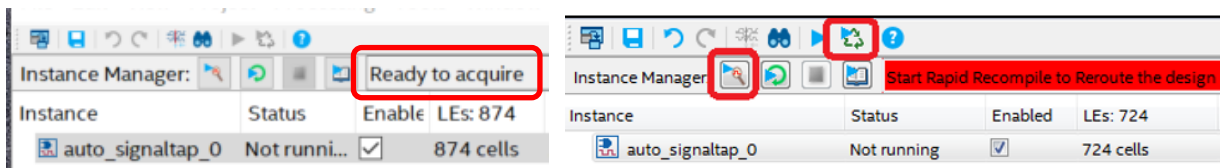
שים לב: בהצגת SMbits ניתן לעשות ZOOM IN כדי לראות מצבים שהם מאד קצרים בזמן או ללחוץ על ה- (+) כדי לפתוח אותו ולראות כל מצב בשורה נפרדת.

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 ארוך יותר כיוון שהתוכנית כוללת גם את הנתח הלוגי.
2. היעזר בשורת ההודעות, כגון "יש לקמפל", "יש לצרוב לכרטיס", "מוכן לדגימה" וכו', ופעל לפיהן. אם שורת ההודעות אדומה – הקומפילציה טרם הושלמה או שיש שגיאה!

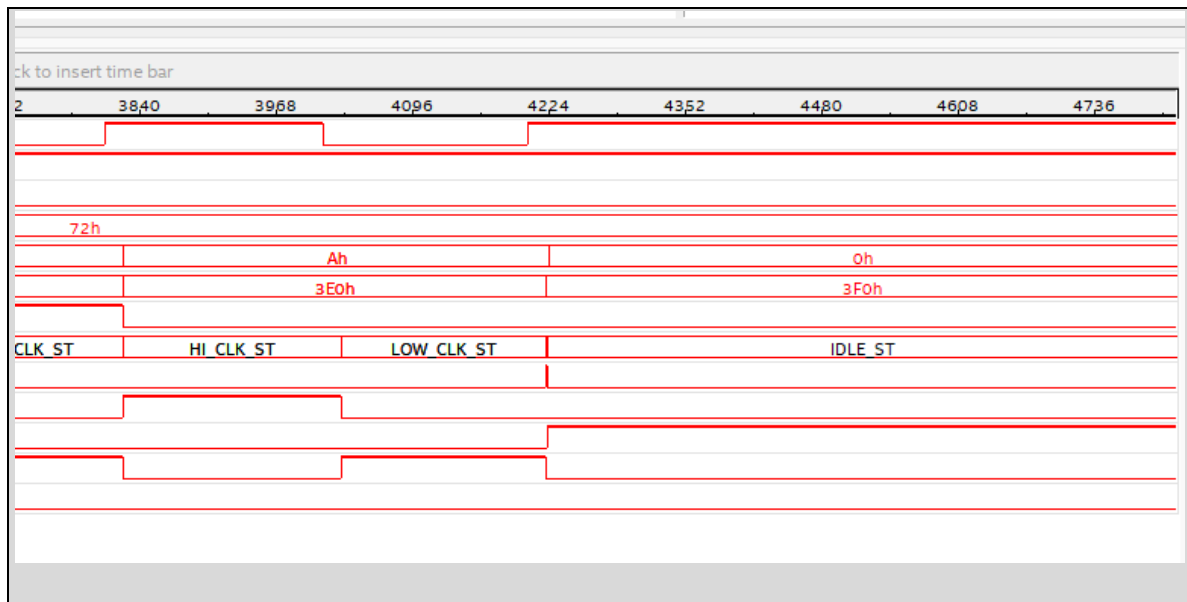


כדי לדבג את המערכת **הפעל** את ה- SignalTap (Auto Run) **ובצע הקלטה** של האותות השונים. **נסה** כמה מקשים, אחד שעובד נכון ואחר שלא עובד, וראה את ההבדל בין התוצאות. התבונן באותות השונים המוצגים ב-SignalTap וחשוב כיצד תוכל להסיק מהם מה התקלה.

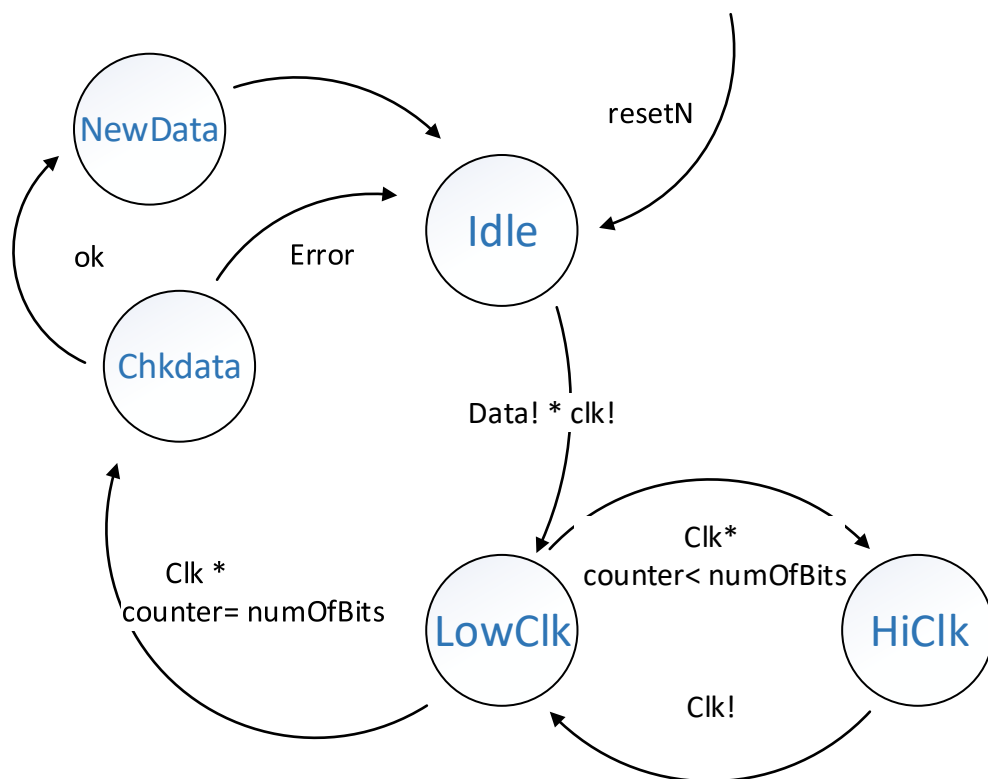
הוסף תדפיס מסך של הרכישה ב-SignalTap במצב תקול לדוח.

Type	Alias	Name	1024	-1023	-1022	-1021	-1020	-1019	-1018	-1017	-1016
		PS2_CLK									
		PS2_DAT									
		dout_new									
		bitrec:inst4 d_bitrec[7..0]									
		bitrec:inst4 cntr[3..0]									
		bitrec:inst4 shift_reg[9..0]									
		bitrec:inst4 parity_ok									
		bitrec:inst4 SM_BITS									

בצע ZOOM על הדגימה בסוף ה-FRAME, לפני המעבר ל-IDLE כך שרואים בדיוק את המעברים של המצבים האחרונים (שהם קצרים מאד בזמן) של מכונת המצבים.



מה היית מצפה שיהיה במוצא האותות dout_new ו-d_bitrec ומה המוצא בפועל?
 תשובה: הייתי מצפה שהדout_new יעל לי ל-1 והוא לא עולה, והייתי מצפה שהbitrec יתעדכן על הערך החדש של המספר 4, וזה לא קורה.



התבונן במצבים של מכונת המצבים. לאילו מצבים לא הגענו (ב-SignalTap אין אות מוצא)?
 תשובה: כאשר לוחצים על 4, המצב של NEW_DATA לא עובד. (באופן טבעי)

על סמך שתי התשובות הנ"ל באיזה מצב אתה חושב שיש תקלה ומה התקלה?
תשובה: התקלה היא ב parity bit והמצב התקול הוא המבצ של check_data.

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

רשום את השעה בה המדריך ראה את המערכת:

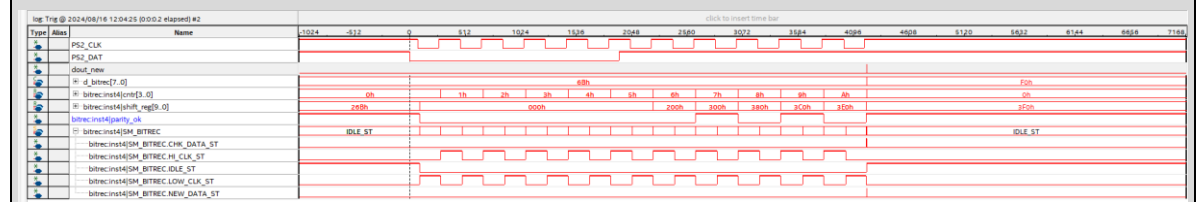
תקן את הקוד של יחידת BITREC **קמפל וצורב** לכרטיס.
בדוק שכעת המערכת עובדת נכון. **בדוק** שמקשים שלא נתנו תוצאה נכונה קודם נותנים את הקוד הנכון עכשיו.

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

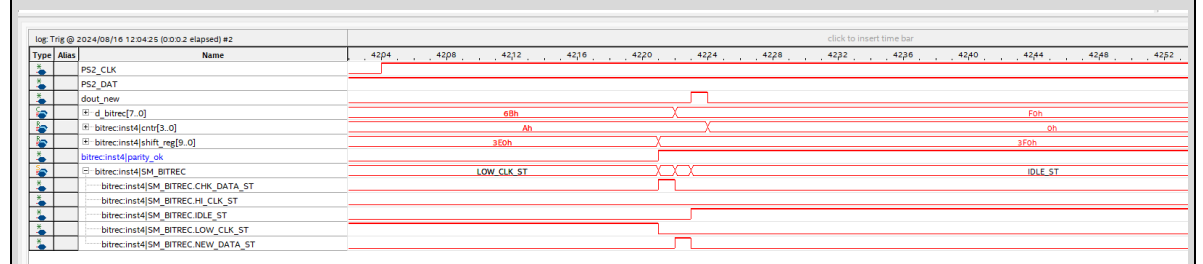
```
126
127 // parity Calc
128 assign parity_ok = shift_reg[8] ^ shift_reg[7] ^ shift_reg[6] ^ shift_reg[5] ^ shift_reg[3]
129     ^ shift_reg[2] ^ shift_reg[4] ^ shift_reg[1] ^ shift_reg[0];
130
131
132 endmodule
```

הפעל את הנתח הלוגי. הקש על מקשים שונים וראה שקוד המקש המתקבל נכון.

הוסף לכאן תדפיס מסך של הרכישה במצב תקין.



בצע ZOOM על האזור המעניין **והוסף** לדוח.



הראה את הצלחת הרכישה במעגל המתקון למדריך המעבדה.

רשום את השעה בה המדריך ראה את המערכת: 12:06

3.2 בדיקת היחידה המתוקנת

עדכון האותות להקלטה:

הורד מרשימת הנתח הלוגי את האותות שהיו נחוצים לדיבוג cnt, parity_ok, shift_reg ומצבי מכונת המצבים.

הוסף את אותות המוצא של BYTEREC: make, break, ו- keyCode.
שנה את תנאי הדרבון לחיווי של לחיצה על מקש (עליה באות make).

עדכון פרמטרים של הנתח הלוגי:

שנה את עומק הזכרון ל- 64K

קבע את מצב הדרבון ל-Post-Trigger, בחלון Setup באזור Trigger.

הבדיקה:

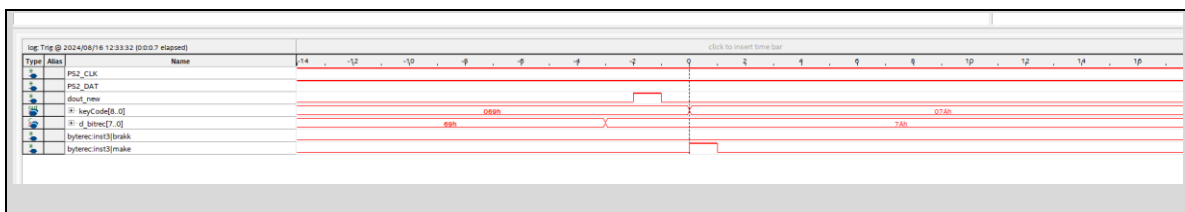
קמפל והורד לכרטיס.

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 על הפעלת מקש רגיל "down arrow 2" או מקש מהסוג המורחב כמו "Enter"?
תשובה: keycode: מקבלים 15Ah וב- d_bitrec קיבלנו 5Ah.

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
ENTER	E0,5A	E0,F0,5A
KP2	72	F0,72

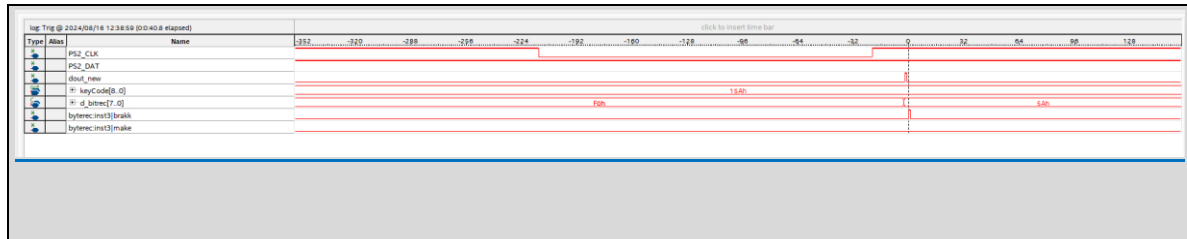


שנה את תנאי הדרבון לעליה באות break (חיווי של עזיבת מקש)

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.

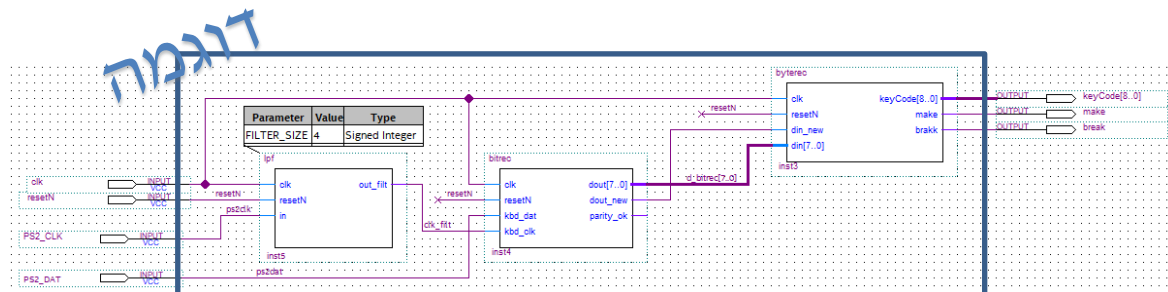
מה הפעם הקודים שמתקבלים ב- d_bitrec וב- keyCode בלחיצה על מקש מסוג מורחב ומקש רגיל?
תשובה: במקש מורחב ENTER מה שכתוב למעלה בירוק, ואותו דבר גם ב- 2.



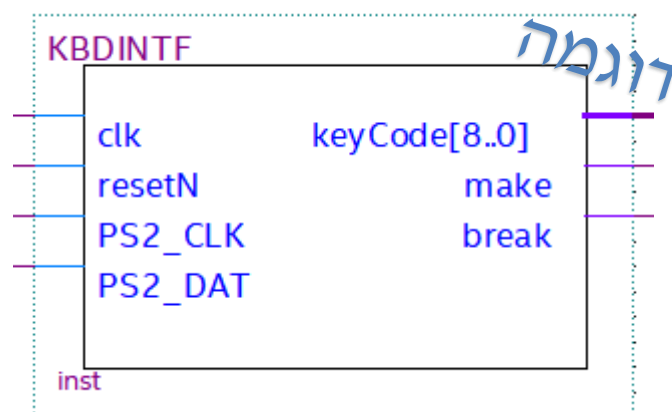
רשום את השעה בה המדריך ראה את המערכת: 12:39

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 העזר (כולל כניסות של מחלק תדר ויציאות לתצוגת 7-Segment). הוסף או עדכן כניסת שעון כמו בדוגמה להלן.



צור Symbol גרפי לממשק למקלדת. סימבול זה ימשש למעגלים ולפרויקטים בהמשך המעבדה.



רשום את השעה בה המדריך ראה את המערכת: 12:58

4 שימוש בממשק למקלדת

המטרה: להדגים שמוש בממשק למקלדת.

פתח את הקובץ הגרפי **KEYBOARD_DECODER.bdf** שעבדת עליו בעבודת ההכנה והגדר אותו כ- TOP.

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

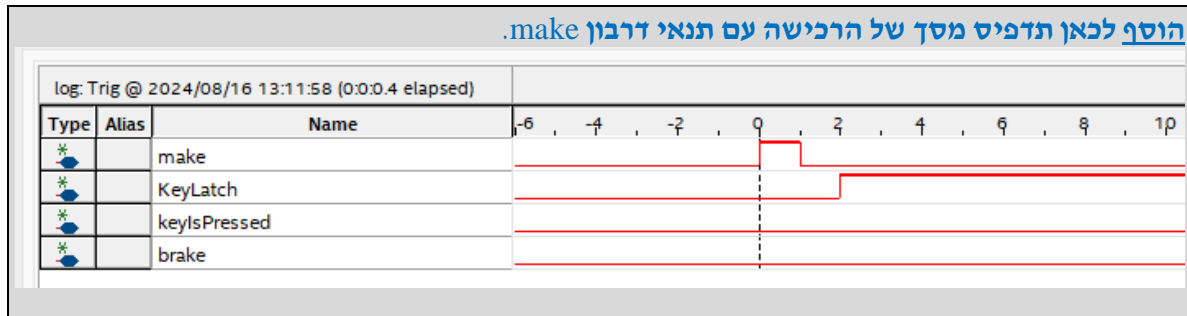
בדוק שכל החיבורים נכונים וכניסות/יציאות קיימות כמו שצריך.

הרץ אנליזה וסינתזה, הרץ קובץ הדקים, קמפל, הורד לכרטיס **נבדוק** עבודה תקינה. למשל, בדוק: ☐ לחיצה על מקש "6" תדליק את LEDR(0) כל זמן הלחיצה. ☐ לחיצה על מקש אחר לא תשפיע על LEDR(0).

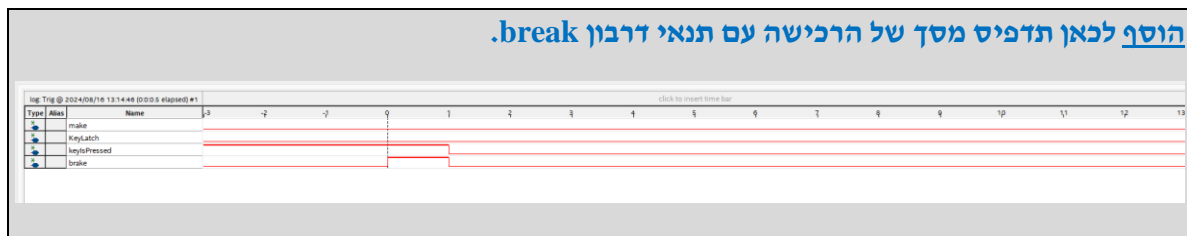
רשום להלן את המקשים שנקבעו לשתי הפעולות, keyIsPressed ו- keyLatch.

מקש	קוד ב- HEX	פעולה
ENTER	A5	מדליק את ledr1 key_latch=
6	74	מדליק את ledr0 =ledr0 keyispressed

הפעל את הנתח הלוגי על **KEYBOARD_DECODER.bdf** והוסף אותות לפי הצורך. **הדגם את הפעולות של שני המקשים והצג** את שתי היציאות הנ"ל של הרכיב **singleKeyDecoder** עם **make** תנאי דרבון.



חזור על הפעלת היישום עם תנאי דרבון **break**.



הראה את הצלחת המשימה למדריך המעבדה שלך.

רשום את השעה בה המדריך ראה את המערכת:

5 גיבוי העבודה

שמור דוח זה רגיל וכ- PDF והעלה את קובץ ה- PDF למודל.

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

שים לב לשנות את השם שמציע הקוורטוס לשם קצר, שאינו מכיל: עברית, רווחים ו/או את הסימן '-' ומכיל את התאריך ושעה של הדחיסה, למשל DEBUG_LABWORK_230323. העלה את קובץ הארכיב למודל, כי תצטרך אותו בהמשך.

גבה את הדוח והפרויקט גם באמצעים אחרים.

רשום את השעה בה סיימת את המעבדה: 13:15