



הפקולטה להנדסת חשמל
ע"ש אנדרו וארנה ויטרבי



הטכניון
מכון טכנולוגי לישראל

מעבדה בהנדסת חשמל
044157 א'1

ניסוי DEBUG - ניפוי תקלות בחומרה

תדריך מעבדה ודוח סיכום

גרסה 3.44
קיץ תשפ"ב 2022

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

סטודנט	שם פרטי	שם משפחה
1	עמיחי	שטרנליכט
2	יקיר	לוגסי

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
5	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
11	3 ניפוי תקלה ובנית ממשק למקלדת
11	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: 15:10

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.

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.

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

 סרטונים להבהרת השימוש בנתח הלוגי, ה-Signal Tap

או בקישור:

<https://youtube.com/playlist?list=PLQsoPmPFZRI-sHU7HmV4w5q2JCSVENzdC>

בתרגיל זה המדריך מרכז את תשומת הלב של הסטודנטים וכולם ביחד, כל זוג בעמדתו על המחשב שלו, מבצעים את הפעולות על פי הנחיות המדריך, כך שבאופן מרוכז עוברים פעולה פעולה. שלב זה מבוצע פרונטלי, לא עם סרטונים.

פתיחת הקובץ	פתיחת הקובץ שנשמר בעבודת ההכנה המונה הפשוט כ- TOP להריץ סינתזה, TCL, קומפילציה מלאה (להסתכל על מספר רכיבים וזכרון)
הפעלת הנתח הלוגי	פתיחת ה SIGNAL TAP לקבוע רכיב (השני), להגדיר קובץ צריבה
קביעת פרמטרים	שעון דגימה (לקבוע clk) עומק זכרון - כמות דגימות – שיקולים (לקבוע 128) 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 לשינוי TRIGGER - אין צורך בקומפילציה הרצה יחידה הרצה מחזורית

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

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.**

Flow Summary	
Flow Status	Successful - Mon Aug 15 15:41:37 2022
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Standard Edition
Revision Name	KBDINTF
Top-level Entity Name	TOP_RANDOM
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	12 / 41,910 (< 1 %)
Total registers	18
Total pins	17 / 499 (3 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
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

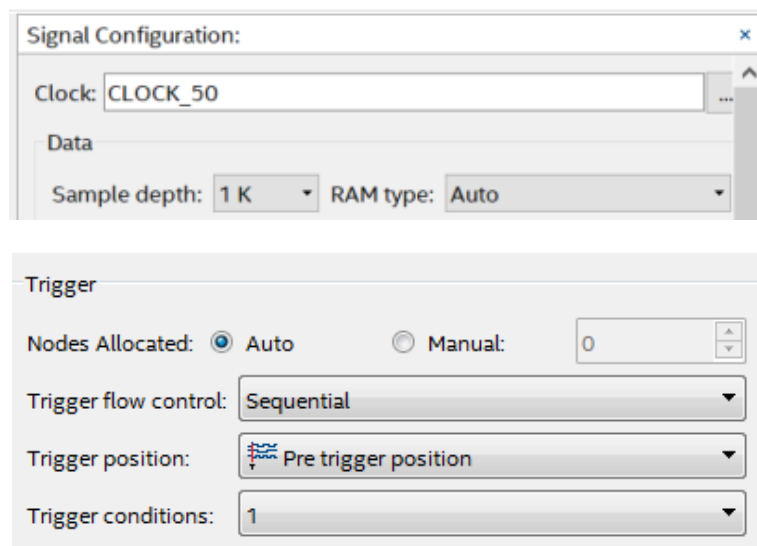
לבצוע המטלות בניסוי זה העזר בפרק "הנתח הלוגי SIGNAL TAP ב-COOK BOOK".
הפעל את הנתח הלוגי (SignalTap II Logic analyzer -> Tools).

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:

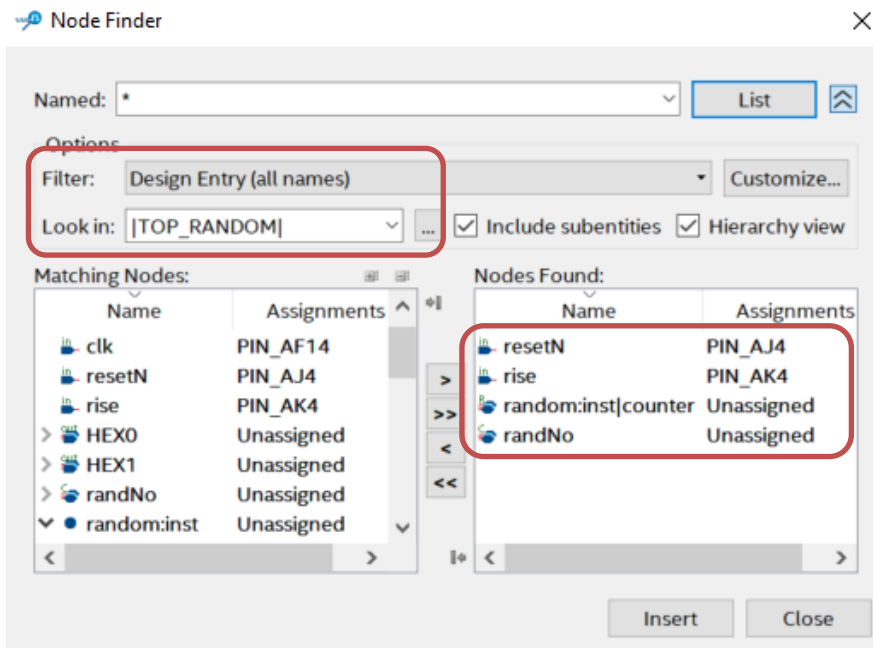


2.2.2 קביעת האותות להקלטה ותנאי הדרבון

קבע את **האותות אותם רוצים להקליט** 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 בחר ב"Design Entry (All Names)".



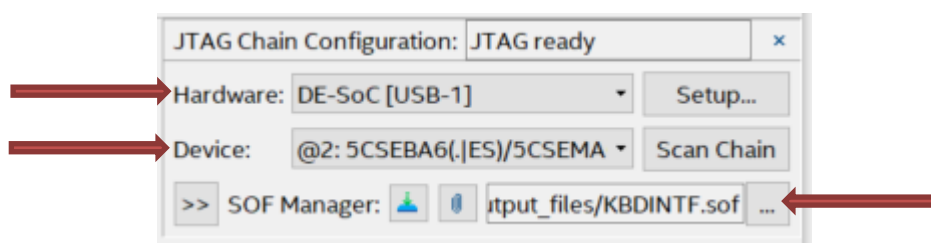
קבע את תנאי ה- Trigger (דברון) על אחד מהאותות להקלטה: **עליה באות rise**.

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

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)	12	314	302
Total block memory bits	0	18432	18432

חשב את תוספת הזכרון שמוסיף הנתח הלוגי למערכת. הסבר את החישוב והאם התוצאה הגיונית?
תשובה:

התוצאה הגיונית מכיוון שנוסף עוד חומרה אשר בודקת את הסיגנלים ועושה אמולציה. פירוט –
אנחנו עוקבים אחרי 4 סיגנלים: resetN ו- rise ברוחב סיבית כל אחד, ו- randNo, counter ברוחב 8 סיביות כל אחד. לכן מדובר ב- $1+1+8+8=18$ סיביות בסה"כ שאנחנו דוגמים.
מכיוון שעומק הדגימה שהוגדרה לנו הוא 1K (1024 סיביות) אזי כמות הביטים הדרושה לזכור היא:
 $18432 = 1024 * 18$ סיביות, כצפוי.

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

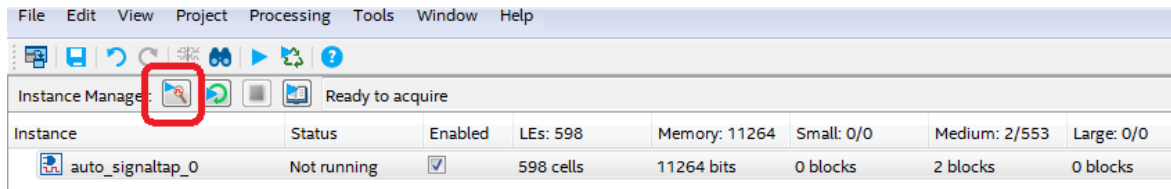
2.3 צריבת הפרויקט מתוך חלון ה-SignalTap

צרו את הפרויקט מתוך חלון ה-SignalTap. העזר ב-COOK BOOK.

בדוק שהמערכת מתנהגת כנדרש, כלומר מתקבלים מספרים אקראיים כמו קודם.

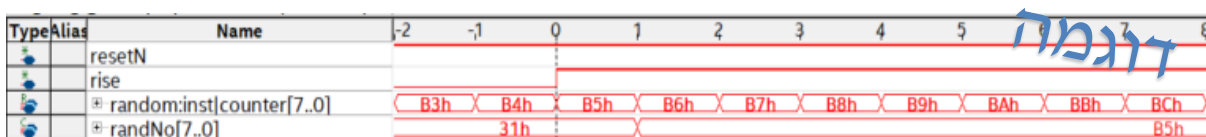
2.4 הפעלת ה-SignalTap

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



להתחלת ההקלטה יש להפעיל את תנאי הדריבון (האות rise).

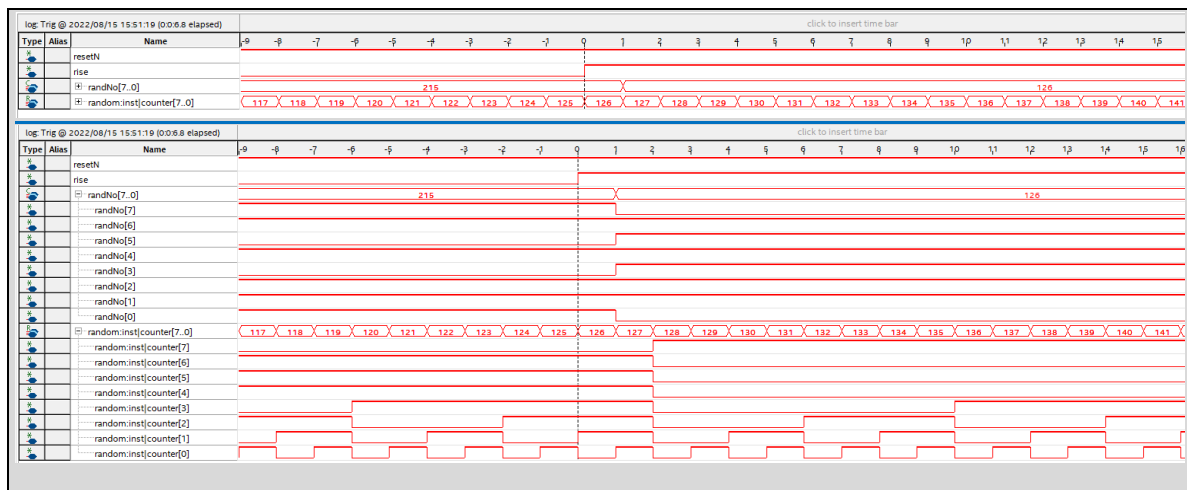
כדי לראות את האותות המוקלטים על ידי ה-SignalTap עבור מ-Setup Tab ל-Data Tab. בחלון ה-Data תתקבל תצוגה דומה ל:



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

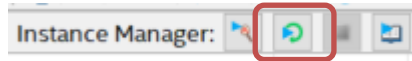
- ☐ העלייה באות rise (שגורמת להיווצרות ה-Trigger)
- ☐ אות המוצא שמקבל, בעלית השעון הבאה, את המספר האקראי שהיה באותו זמן במונה counter (בדוגמה לעיל זה 85h).
- ☐ הערך שהיה במוצא לפני העלייה באות rise (בדוגמה לעיל זה 31h). במערכת שלך כמובן שיתקבלו ערכים אחרים.

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.



2.4.1 הפעלת ה-SignalTap במצב Autorun

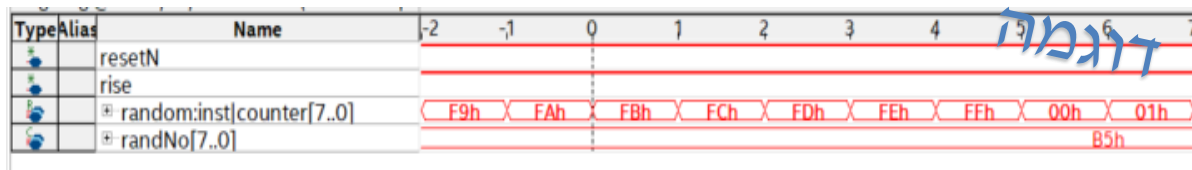
להפעלה אוטומטית של הנתח הלוגי אחרי כל Trigger הקש על הכפתור הריצה האוטומטית ה-Autorun Analysis.



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**:

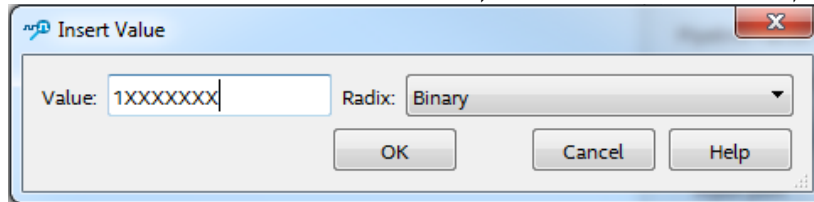
עם עצירת הנתח הלוגי שלא בעקבות Trigger כל שהוא, תתקבל תצוגה אקראית, הדומה ל:



רשום את השעה בה המדריך ראה את הצלחת הרכישה: 15:50

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.



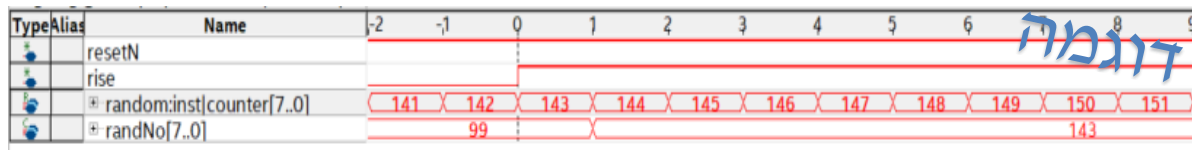
סימני ה - X הם Don't Care, לכן ייבחרו כל הצירופים שבהם סיבית ה - MSB היא 1 ושאר הסיביות הנמוכות יותר מהוות צרף כל שהוא (128, 129 .. 255). פעולת AND נעשית בין שני התנאים, כלומר ההקלטה תופעל כל פעם שנלחץ על rise וספירת המונה גדולה מ - 128.

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.

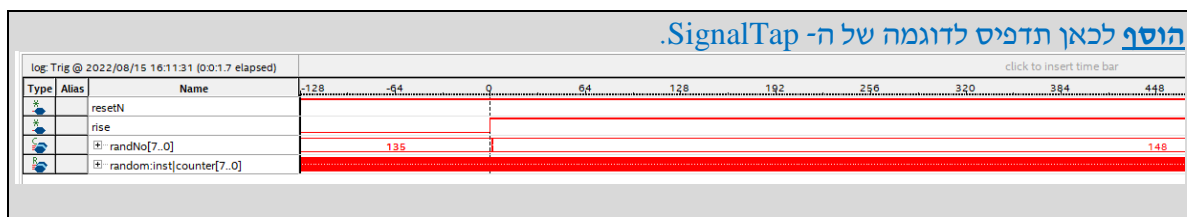
להלן דוגמה של הקלטה בה מתקיימים שני התנאים.



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

trigger: 2022/08/15 16:02:41 #0					Lock mode: Allow all changes
Type	Alias	Name	Data Enable	Trigger Enable	Trigger Conditions
		resetN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		rise	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		+ randNo[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXb
		+ random:inst[counter[7..0]]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1XXXXXXXXb

הוסף לכאן תדפיס לדוגמה של ה- SignalTap.



רשום את השעה בה המדריך ראה את התוצאות: 16:10

3 ניפוי תקלה ובנית ממשק למקלדת

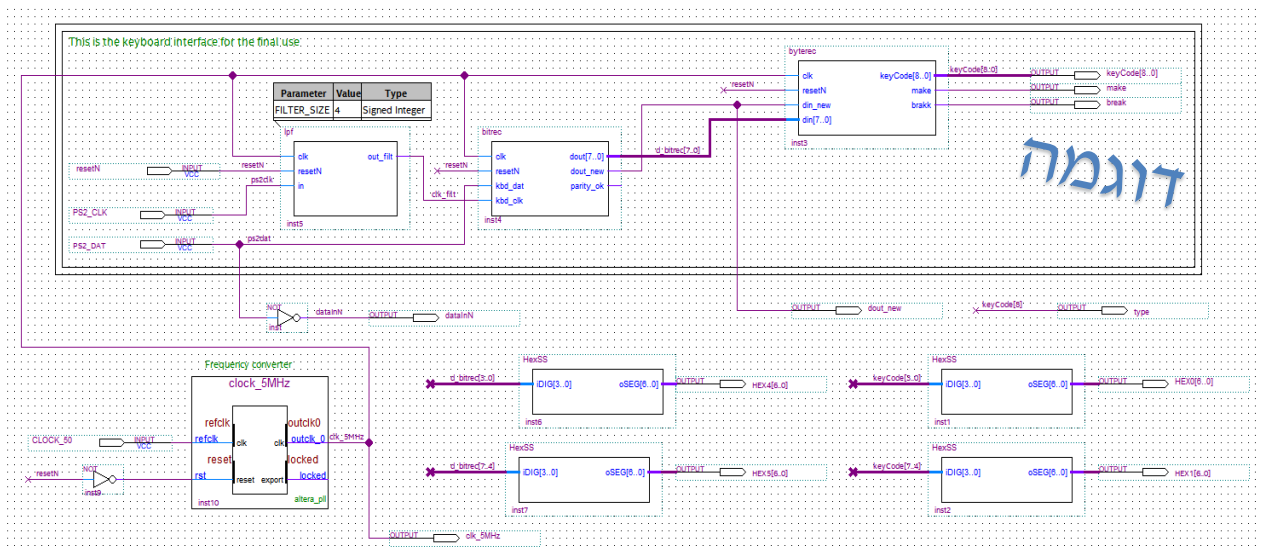
מטרה: בחלק זה של הניסוי תבנה ממשק למקלדת, תמצא באמצעות הנתח הלוגי, את התקלה ששוררבה בו, תתקן את התקלה ולבסוף תבדוק שהממשק המתוקן עובד נכון.

חשוב מאד 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- 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.tel** 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** 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:

LED[8] מציינת את הגעת הנתונים, LED[9] מציינת את סוג המקש – מורחב. רגיל

האם יש מקשים שעובדים נכון, ואם כן איזה?

תשובה:

NUMLOCK, ENTER, 1, 2, 3, 5, 7 עובדים

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)"** ודא 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_PS	○	- הרגיסטר של מכונת המצבים – אות פנימי של BITREC
cntr_PS	○	- המונה של מכונת המצבים – אות פנימי של BITREC

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

○ **pres_st** - המצבים של מכונת המצבים של bitrec

באופן הבא:

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

(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 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"/>	
		+ 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_PS[9..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXh
		+ bitrec:inst4 cntr_PS[3..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		+ bitrec:inst4 pres_st	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXh

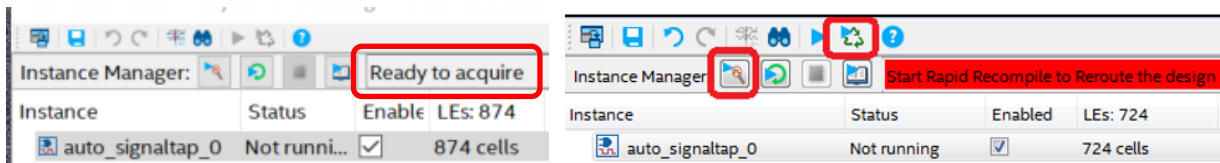
שים לב: בהצגת **pres_st** ניתן לעשות ZOOM IN כדי לראות מצבים שהם מאד קצרים בזמן או ללחוץ על ה- (+) כדי לפתוח אותו ולראות כל מצב בשורה נפרדת.

בצע את שתי הפעולות הבאות דרך ה- SignalTap:

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

שים לב:

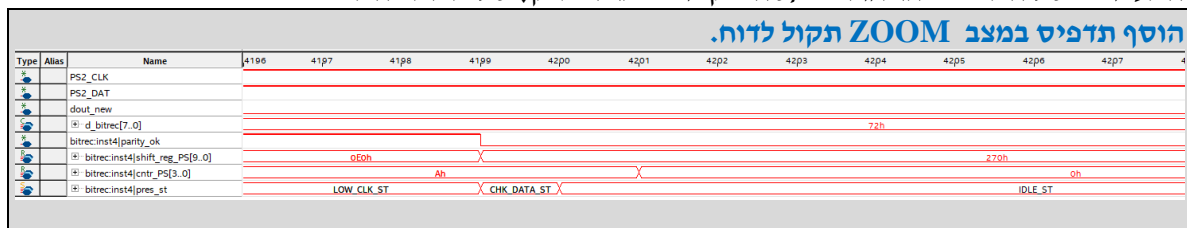
1. זמן הקופילציה הפעם הוא ארוך יותר כיוון שהתוכנית כוללת גם את הנתח הלוגי.
2. היעזר בשורת ההודעות, כגון "יש לקמפל", "יש לצרוב לכרטיס", "מוכן לדגימה" וכו', ופעל לפיהן. אם שורת ההודעות אדומה – הקומפילציה טרם הושלמה או שיש שגיאה!



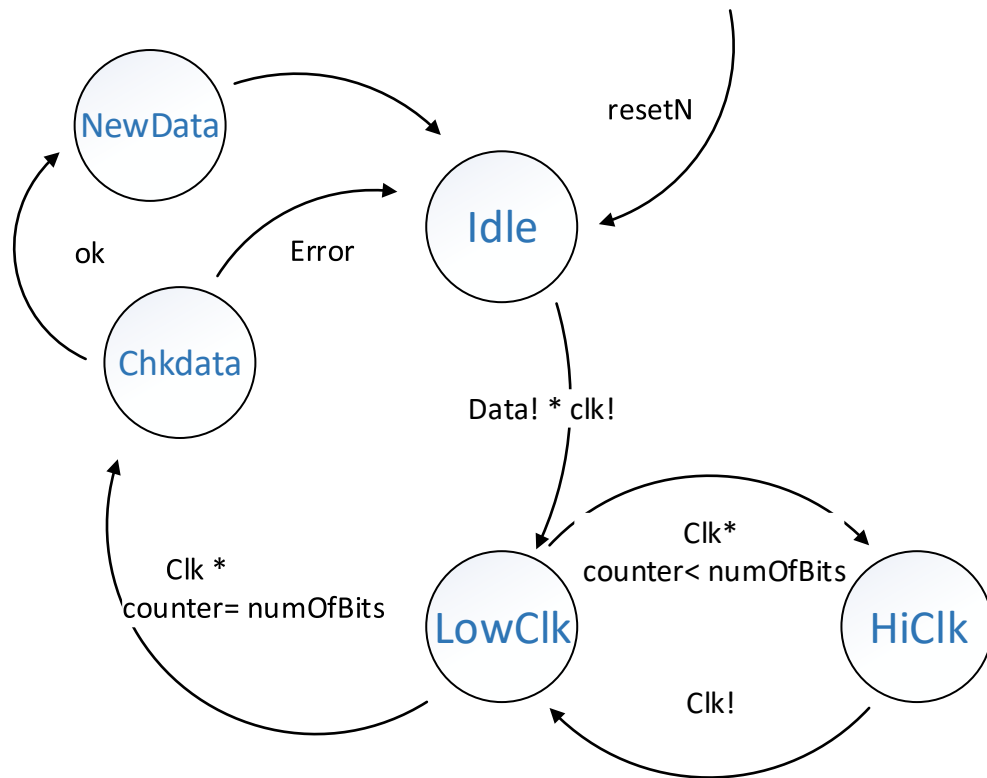
כדי לדבג את המערכת **הפעל** את ה- SignalTap (Auto Run) **ובצע הקלטה** של האותות השונים. **נסה** כמה מקשים, אחד שעובד נכון ואחר שלא עובד, וראה את ההבדל בין התוצאות. התבונן באותות השונים המוצגים ב-SignalTap וחשוב כיצד תוכל להסיק מהם מה התקלה.



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:
 dout_new אמור להיות 1 מכיוון שקיבלנו לחיצה על מקש, d_bitrec אמור להיות טעון בקוד של המקש שלחצנו ולא בערך של המקש האחרון שעבד.



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

על סמך שתי התשובות הנ"ל באיזה מצב אתה חושב שיש תקלה ומה התקלה?
 תשובה:
 התקלה נמצאת במצב CHKDATA בגלל שבדיקת parity אינה נכונה ולכן לא מתבצע מעבר ל newdata כנדרש.

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. שמך בצהוב את המקום ששינית 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
                  HI_CLK_ST, // after clock hi
                  CHK_DATA_ST, // after all bits recieved
                  NEW_DATA_ST // valid parity laod new data
                } pres_st /* for simulation --> synthesis keep = 1 */,
                next_st /* synthesis keep = 1 */;

logic [3:0] cntn_PS, cntn_NS /* for simulation -->synthesis keep = 1 */ ;
logic [9:0] shift_reg_PS, shift_reg_NS /* for simulation -->synthesis keep = 1 */ ;
logic [7:0] Dout_NS /* for simulation -->synthesis keep = 1 */ ;

localparam NUM_OF_BITS = 10 ; // &&&&& fill please the right number

always_ff @(posedge clk or negedge resetN)
begin: fsm_sync_proc
    if (resetN == 1'b0) begin
        pres_st <= IDLE_ST ;
        cntn_PS <= 4'h0 ;
        shift_reg_PS <= 10'h000 ;
        dout <= 8'h00 ;

    end
    else begin
        pres_st <= next_st;
        cntn_PS <= cntn_NS ;
        shift_reg_PS <= shift_reg_NS ;
        dout <= Dout_NS ;

    end ;
end // end fsm_sync_proc

always_comb
begin
    // default values
    next_st = pres_st ;
    cntn_NS = cntn_PS ;
    shift_reg_NS = shift_reg_PS ;
    Dout_NS = dout ;
    dout_new = 1'b0 ;

    case(pres_st)
        IDLE_ST:
            begin
                //-----
                cntn_NS = 4'h0 ;
                if ( (kbd_clk == 1'b0) && (kbd_dat == 1'b0) ) //check start bit
                    next_st = LOW_CLK_ST;

            end

        LOW_CLK_ST:
            begin
                //-----
                if (kbd_clk == 1'b1)
                    begin
                        shift_reg_NS = {kbd_dat, shift_reg_PS [9:1]};
                        if (cntn_PS < NUM_OF_BITS)
                            begin
                                cntn_NS <= cntn_PS + 1;
                                next_st = HI_CLK_ST;

                            end
                        else if (cntn_PS == NUM_OF_BITS)
                            begin
                                next_st = CHK_DATA_ST;

                            end
                        end

                    end

                HI_CLK_ST:
                    begin
                        //-----
                        if (kbd_clk == 1'b0)

```

```

begin
    next_st = LOW_CLK_ST;
end

end

CHK_DATA_ST:
begin
//-----
    if ((kbd_clk == 1'b1) && (kbd_dat == 1'b1)) //check stop bit
    begin
        if (parity_ok == 1'b1)
        begin
            Dout_NS = shift_reg_PS[7:0];
            next_st = NEW_DATA_ST;
        end
        else
        begin
            next_st = IDLE_ST;
        end
    end
end

NEW_DATA_ST:
begin
    dout_new = 1'b1;
    next_st = IDLE_ST;
//-----
end

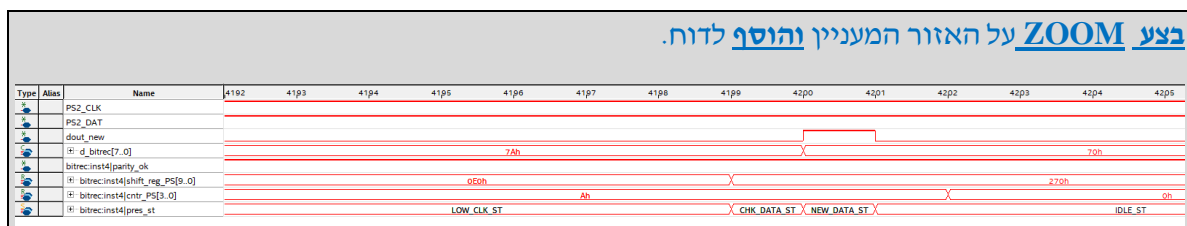
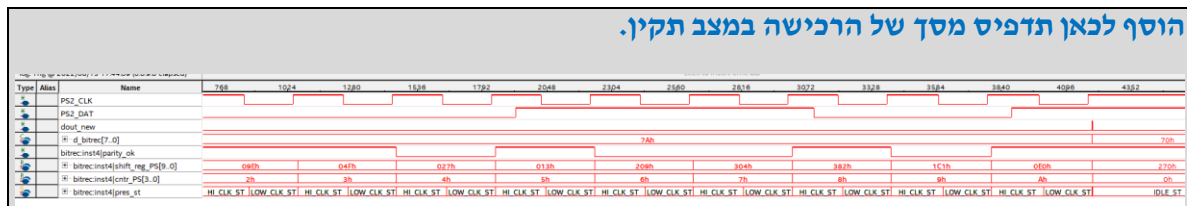
endcase
end

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

endmodule

```

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



הראה את הצלחת הרכישה במעגל המתוקן למדריך המעבדה.

רשום את השעה בה המדריך ראה את המערכת: 17:47

3.2 בדיקת היחידה המתוקנת

עדכון האותות להקלטה:

הורד מרשימת הנתח הלוגי 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.

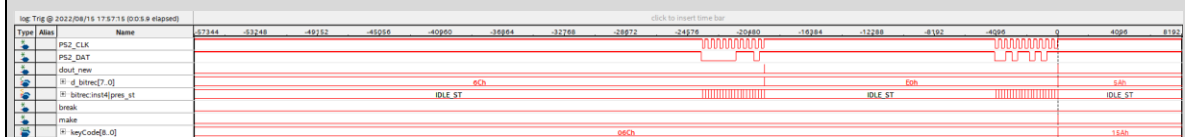
הפעל את הנתח הלוגי במצב Autorun ולחץ של מקשים שונים. שים לב לתוצאה ב- d_bitrec וב- keyCode.

מה הקודים שמתקבלים ב- d_bitrec וב- keyCode על הפעלת מקש רגיל "down arrow 2" או מקש מהסוג המורחב כמו "Enter"?
תשובה:
עבור "down arrow 2" מקבלים ב- d_bitrec קוד 72 וב- keyCode קוד 072 (כי זה מקש אינו מורחב).
עבור Enter מקבלים ב- d_bitrec קוד E05A וב- keyCode קוד 15A (כי זה מקש מורחב).

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

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



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:
עבור "down arrow 2" מקבלים ב- d_bitrec קוד F072 וב- keyCode קוד 072 (כי זה מקש אינו מורחב).
עבור Enter מקבלים ב- d_bitrec קוד E0F05A וב- keyCode קוד 15A (כי זה מקש מורחב).

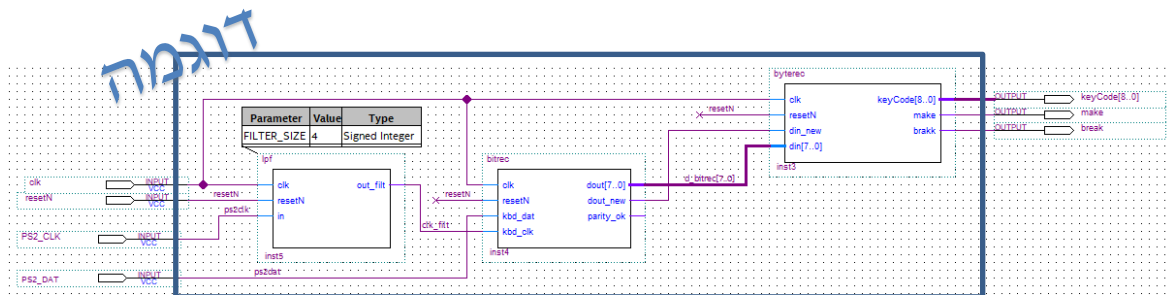
הוסף לכאן תדפיס מסך של הרכישה עם תנאי דרבון break. וודא שרואים את כל התווים שנקלטים.
בצע ZOOM OUT כך שרואים את כל אות הכניסה, החל מה- START BIT של התו הראשון.



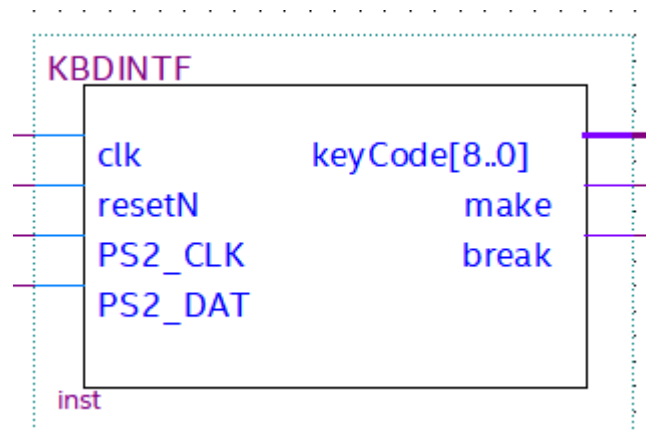
רשום את השעה בה המדריך ראה את המערכת: 18:05

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 ימשך למעגלים ולפרויקטים בהמשך המעבדה.



רשום את השעה בה המדריך ראה את המערכת: 18:13

4 שימוש בממשק למקלדת

המטרה: להדגים שמוש בממשק למקלדת.

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

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

צור סימבול למודול שהשלמת בעבודת ההכנה, **singleKeyDecoder**.

עדכן גם את הסימבול של **singleKeyDecoder** עם הסימבול שלך.

בדוק שכל החיבורים נכונים וכניסות/יציאות קיימות כמו שצריך.

קמפל, הורד לכרטיס **ובדוק** עבודה תקינה. למשל, בדוק:

☐ לחיצה על מקש "6" תדליק את LEDR(0) כל זמן הלחיצה.

☐ לחיצה על מקש אחר לא תשפיע על LEDR(0).

אפשר לשנות את אחד הקבועים של הרכיב **singleKeyDecoder** לקוד מקש אחר, על ידי:

☐ מקש ימני על טבלת **Parameter** בשרטוט, בחירת **Properties → Parameter**

☐ הקלקה כפולה על **Value** ושינוי ערכו לקוד המקש המבוקש.

רשום את המקש שנקבע ולאיוזו משתי הפעולות, **keyIsPressed** או **keyLatch**.

מקש	קוד ב- HEX	פעולה
"6"	074	keyIsPressed
"ENTER"	15A	keyLatch

הפעל את הנתח הלוגי על **KEYBOARD_DECODER.bdf** והוסף אותות לפי הצורך. הצג את שתי

היציאות הנ"ל של הרכיב **singleKeyDecoder** עם תנאי דרבון **make**.

הוסף לכאן תדפיס מסך של הרכישה עם תנאי דרבון make.
לחיצה על "6"

Type	Alias	Name	25672	25624	24976	22328	20480	18632	16984	14336	12288	10240	8192	6144	4096	2048	4096
make																	
keyCode[0]										15A0							0740
keyIsPressed																	
keyLatch																	
keyRisingEdge																	
resetH																	
brake																	

לחיצה על enter

Type	Alias	Name	25672	25624	24976	22328	20480	18632	16984	14336	12288	10240	8192	6144	4096	2048	4096
make																	
keyCode[0]										15A0							
keyIsPressed																	
keyLatch																	
keyRisingEdge																	
resetH																	
brake																	

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

הוסף לכאן תדפיס מסך של הרכישה עם תנאי דרבון break.
לחיצה על "6"

Type	Alias	Name	25672	25624	24976	22328	20480	18632	16984	14336	12288	10240	8192	6144	4096	2048	4096
make																	
keyCode[0]										0740							
keyIsPressed																	
keyLatch																	
keyRisingEdge																	
resetH																	
brake																	

לחיצה על enter

Type	Alias	Name	25672	25624	24976	22328	20480	18632	16984	14336	12288	10240	8192	6144	4096	2048	4096
make																	
keyCode[0]										15A0							
keyIsPressed																	
keyLatch																	
keyRisingEdge																	
resetH																	
brake																	

הראה את הצלחת המשימה למדריך המעבדה שלך.

רשום את השעה בה המדריך ראה את המערכת: 18:43

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 18_10_2021_DEBUG_LABWORK. העלה את קובץ הארכיב למודל, כי תצטרך אותו בהמשך.

גבה את הדוח והפרויקט גם באמצעים אחרים.

רשום את השעה בה סיימת את המעבדה: 18:45