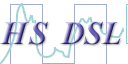
הטכניון – מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל



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)

[1 לימוד מודרך של SignalTap בעזרת מונה 3](#_Toc33634234)

[2 לימוד עצמאי של SignalTap בעזרת מכונתRANDOM 3](#_Toc33634235)

[2.1 בדיקת ה- RANDOM על הכרטיס 3](#_Toc33634236)

[2.2 קונפיגורציה בסיסית של ה - **SignalTap** 4](#_Toc33634237)

[2.2.1 קביעת השעון, עומק הזכרון ואופן ההקלטה 4](#_Toc33634238)

[2.2.2 קביעת האותות להקלטה ותנאי הדרבון 5](#_Toc33634239)

[2.2.3 קביעת ההתקן לקראת צריבה 6](#_Toc33634240)

[2.2.4 משאבי החומרה הדרושים 6](#_Toc33634241)

[2.3 צריבת הפרויקט מתוך חלון ה - **SignalTap** 7](#_Toc33634242)

[2.4 הפעלת ה - **SignalTap** 7](#_Toc33634243)

[2.4.1 הפעלת ה- SignalTap במצב Autorun 8](#_Toc33634244)

[2.5 שימוש ב - **Trigger** הכולל כמה תנאים 9](#_Toc33634245)

[3 ניפוי תקלה ובנית ממשק למקלדת 11](#_Toc33634246)

[3.1 שימוש ב - **SignalTap** לניפוי תקלה – ביחידה**BITREC** 11](#_Toc33634247)

[3.2 בדיקת היחידה המתוקנת 17](#_Toc33634248)

[3.3 יצירת סימבול של הממשק למקלדת 20](#_Toc33634249)

[4 שימוש בממשק למקלדת 21](#_Toc33634250)

[5 גיבוי העבודה 21](#_Toc33634251)

[6 פרויקט 22](#_Toc33634252)

***רשום את השעה בה התחלת את המעבדה: 8:52***

# 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 הCOOKBOOK QUARTUS 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  RADIX לHEX  קביעת TRIGGER (לקבוע ירידה ב- resetN) |
| הרצה | קומפילציה (להסתכל על מספר רכיבים וזכרון)  שמירת STP לתיקיה  צריבה מהכלי  לשינויTRIGGER - אין צורך בקומפילציה  הרצה יחידה  הרצה אוטומטית |

# לימוד עצמאי של SignalTap בעזרת מכונתRANDOM

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

## בדיקת ה- RANDOM על הכרטיס

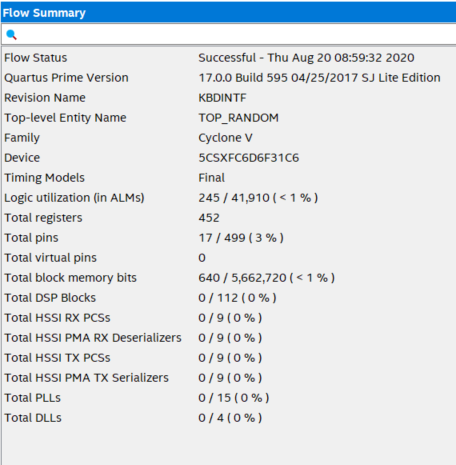
**הגדר** את הקובץ **TOP-RANDOM.bdf** כ- Top. בעבודת ההכנה הכרת את אופן פעולתו של **random.sv**. הקובץ הגרפי כולל בנוסף 2 יחידות תצוגה על 7Seg.

**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.**



**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.

## קונפיגורציה בסיסית של ה - **SignalTap**

*לבצוע המטלות בניסוי זה העזר בפרק " הנתח הלוגי SIGNAL TAP ב-COOK BOOK*.

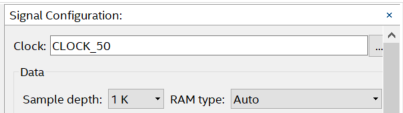
**הפעל** את הנתח הלוגי (SignalTap II Logic analyzer).

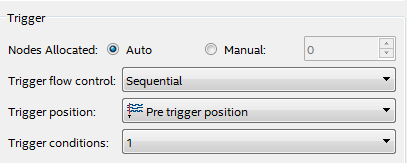
### קביעת השעון, עומק הזכרון ואופן ההקלטה

**קבע** את הקונפיגורציה הבסיסית של הנתח הלוגי:

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

אחרי קביעות אלה חלון הקונפיגורציה צריך להיראות כך:



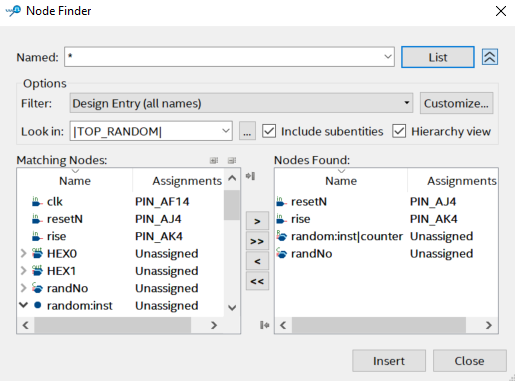


### קביעת האותות להקלטה ותנאי הדרבון

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

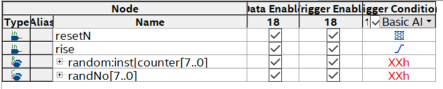
1. resetN
2. אות הכניסה rise
3. וקטור הספירה count
4. וקטור היציאה של מספרים אקראיים dout

**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**.

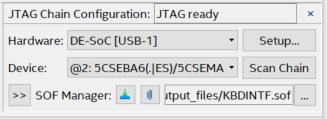
חלון ה - Setup ייראו כך (עם האותות להקלטה ותנאי הדרבון):



### 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.

### משאבי החומרה הדרושים

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

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

**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** 18,432 **= 1k \* 18**

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

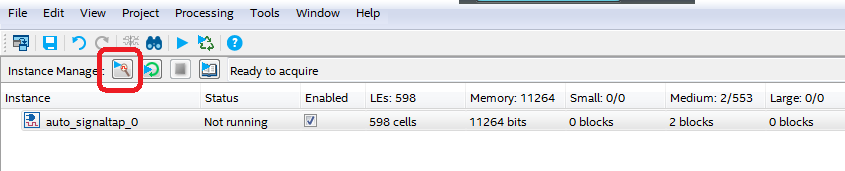
## צריבת הפרויקט מתוך חלון ה - **SignalTap**

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

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

## הפעלת ה - **SignalTap**

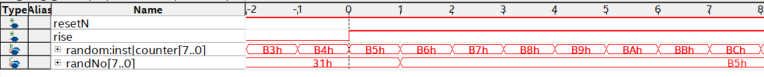
**להפעלת הנתח הלוגי**, כלומר ליצירת קשר עם החמרה, הקש על מקש ה- Run Analysis.



**להתחלת ההקלטה** יש להפעיל את תנאי הדרבון (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:

דוגמה



פעולת **Right-Click ו Left-Click** גורמות לפעולות **Zoom-Out ו Zoom-In** בהתאמה.

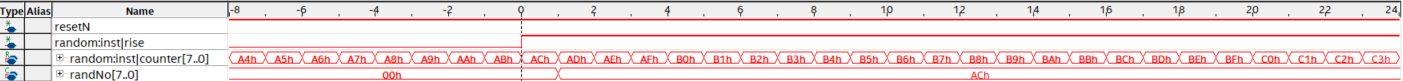
**בתצוגה** שתקבל צריכים לראות בבירור את:

1. העלייה באות rise (שגורמת להיווצרות ה - Trigger)
2. אות המוצא שמקבל, בעלית השעון הבאה, את המספר האקראי שהיה באותו זמן במונה counter (בדוגמה זה 85h).
3. הערך שהיה במוצא לפני העליה 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.**



### 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.

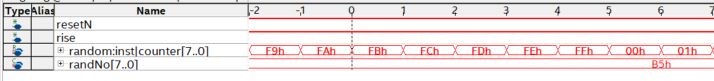
בכל לחיצה על ה- rise תתקבלנה תוצאות אחרות.



**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:

דוגמה



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

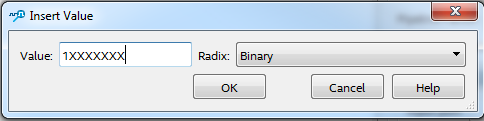
## שימוש ב - **Trigger** הכולל כמה תנאים

הנתח הלוגי של Quartus מאפשר ליצור תנאי Trigger מורכבים יותר.

**הוסף תנאי שני**, שהמונה יהיה גדול מ – 128, לתנאי הדרבון הקודם:

1. מקש ימני בחלון ה Setup על התא אותו רוצים לשנות ,
2. 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 צרוף כל שהוא (128, 129 .. 255). פעולת AND נעשית בין שני התנאים, כלומר ההקלטה תופעל כל פעם שנלחץ על rise וספירת המונה גדולה מ - 128.

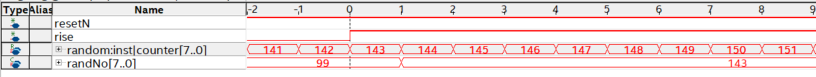
**שנה** את תצוגת המספרים ל – Unsigned Decimal:

1. סמן את האות, מקש ימני, בחירת Bus Display Format ברשימה המתקבלת,
2. ובחירת Unsigned Decimal ברשימה הנוספת.

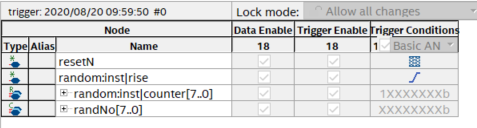
**בדוק** את התנאי החדש, הפעל את המערכת במצב Autorun. שים לב שעכשיו בממוצע רק כמחצית מהלחיצות על KEY0 תהיה הקלטה.

להלן דוגמה של הקלטה בה מתקיימים שני התנאים.

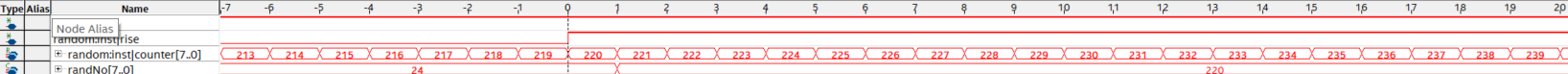
דוגמה



הוסף את תדפיס מסך ההגדרות המראה את הגדרת ה TRIGGER



**הוסף לכאן תדפיס לדוגמה של ה- SignalTap.**



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

# 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 מאד בניסוי זה לפעול על פי ההוראות בקפידה ולבצע את הניסוי בדיוק לפי השלבים הנתונים!**

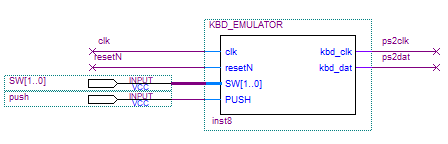
## שימוש ב - **SignalTap** לניפוי תקלה – ביחידה**BITREC**

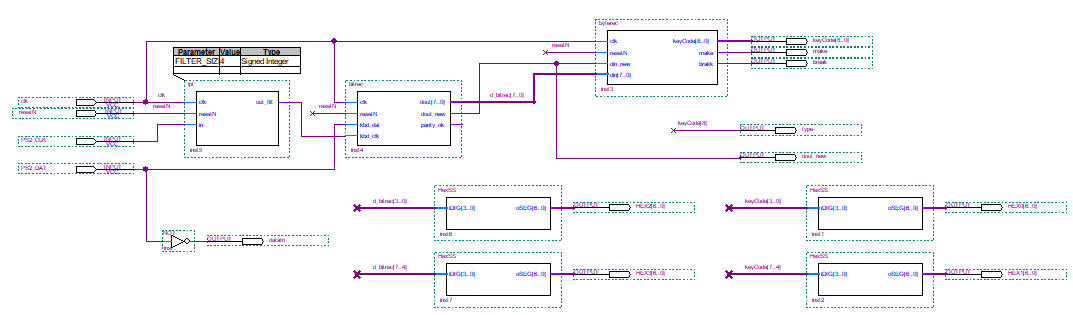
**שים לב! ביחידה BITREC אותה הכנת בבית שורבבה תקלה! למרות שבסימולציה שעשית בבית היחידה עבדה נכון, צפוי שכאשר תעבוד עם מקלדת אמיתית המערכת לא תעבוד נכון!**

**המטרה** בשלב ראשון היא למצוא את התקלה שקיימת ביחידת ה- BITREC שתכננת 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





דוגמה

**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 **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.

**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 ו**בדוק** 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, 0SW ו- 1SW, ולחצן 1KEY**:

1. 0SW בורר בין מקש "**רווח"** ומקש "**חץ ימינה**"
2. 1SW בורר בין מקש מסוג ישן או חדש

כמתואר להלן:

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

**כדי להפעיל מקש מסויים** יש לשים את המתגים במצב הרצוי ואז ללחוץ על **לחצן 1KEY**

**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**

**NONE-EX מראה 29 ומספר לא קיים משמאל. EXTENDED וכנל 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?**

**תשובה:** SPACE רגיל

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

1. שעון הנתח הלוגי: אות השעון **CLOCK\_50**
2. עומק הזיכרון: **32K**
3. מצב הדרבון:**position** **Pre-trigger**.
4. **הצג את האותות** הבאים ב – SignalTap.

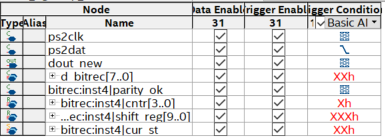
**הערה: לשם קביעת האותות המוצגים, בחלון ה- Node Finder השתמש ב- Filter בשם   
 "Design entry (all names)" וודא שהאפשרות "Include subentities" מסומנת.**

* + 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 דרבון:** התחלה של העברת הנתונים (**ירידה ב- ps2dat)**



דוגמה

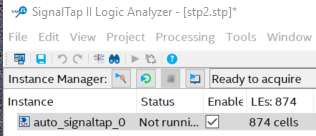
**שים לב**: בהצגת **cur\_st** ניתן לעשות ZOOM IN כדי לראות מצבים שהם מאד קצרים בזמן או ללחוץ על ה- (+) כדי לפתוח אותו ולראות כל מצב בשורה נפרדת.

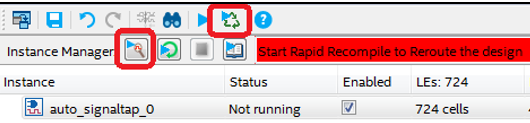
**בצע את שתי הפעולות הבאות דרך ה- SignalTap :**

* **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!



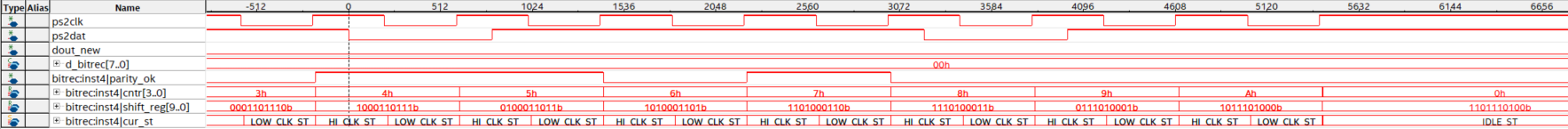


כדי לדבג את המערכת **הפעל** את ה - SignalTap (Auto Run ) **ובצע הקלטה** של האותות השונים.

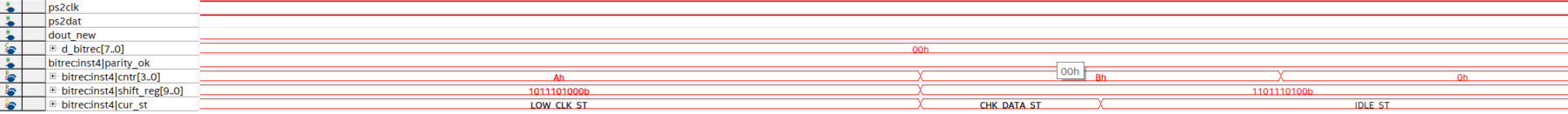
הקש על כמה מקשים, אחד שעובד נכון ואחר שלא עובד, וראה את ההבדל בין התוצאות.

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 בPARITYBIT.  
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 וצרוב** לכרטיס.

**בדוק** שהמקלדת עובדת: **הקש** על מקשים שונים וראה שהתצוגות מראות את הקוד הנכון.

**הוסף** כאן את כל הקובץ המתוקן. **סמן** 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 // if valid parity announce new data

**}** nxt\_st**,** cur\_st /\* synthesis keep = 1 \*/**;**

**logic** **[**3**:**0**]** cntr**,** nextCntr /\* synthesis keep = 1 \*/ **;**

**logic** **[**9**:**0**]** shift\_reg**,** Next\_Shift\_Reg /\* synthesis keep = 1 \*/ **;**

**logic** **[**7**:**0**]** Next\_Dout /\* synthesis keep = 1 \*/ **;**

**localparam** NUM\_OF\_BITS **=** 4'ha **;** // &&&&&&&&&&&&&& fill please

**always\_ff** **@(posedge** clk **or** **negedge** resetN**)**

**begin:** fsm\_sync\_proc

**if** **(**resetN **==** 1'b0**)** **begin**

cur\_st **<=** IDLE\_ST **;**

cntr **<=** 4'h0 **;**

shift\_reg **<=** 10'h000 **;**

dout **<=** 8'h00 **;**

**end**

**else** **begin**

cur\_st **<=** nxt\_st**;**

cntr **<=** nextCntr**;**

shift\_reg **<=** Next\_Shift\_Reg**;**

dout **<=** Next\_Dout**;**

**end**

**end** // end fsm\_sync\_proc

**always\_comb**

**begin**

// default values

nxt\_st **=** cur\_st **;**

nextCntr **=** cntr **;**

Next\_Shift\_Reg **=** shift\_reg **;**

Next\_Dout **=** dout **;**

dout\_new **=** 1'b0 **;**

**case(**cur\_st**)**

IDLE\_ST**:** **begin**

nextCntr **=** 4'h0 **;**

**if(** **(**kbd\_clk **==** 1'b0**)** **&&** **(**kbd\_dat **==** 1'b0**)** **)**

nxt\_st **=** LOW\_CLK\_ST**;**

**end**

LOW\_CLK\_ST**:** **begin**

**if** **(**kbd\_clk **==** 1'b1**)** **begin**

Next\_Shift\_Reg **=** **{**kbd\_dat**,**shift\_reg **[**9**:**1**]};**

nextCntr **=** cntr **+** 1**;**

**if** **(**cntr **<** NUM\_OF\_BITS**)**

nxt\_st **=** HI\_CLK\_ST**;**

**else** **if** **(**cntr **==** NUM\_OF\_BITS**)**

nxt\_st **=** CHK\_DATA\_ST**;**

**end**

**end**

HI\_CLK\_ST**:** **begin**

**if** **(!**kbd\_clk**)**

nxt\_st **=** LOW\_CLK\_ST**;**

**end**

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

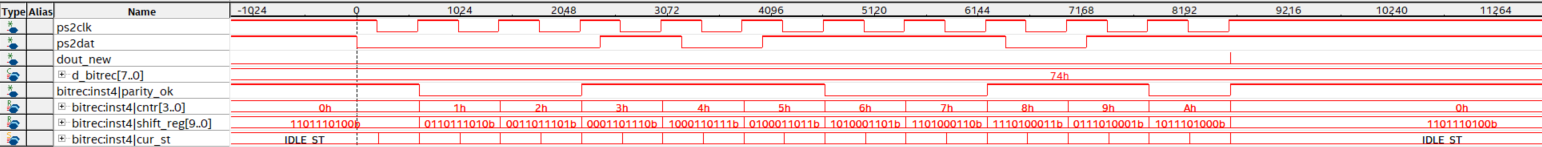
**assign** parity\_ok **=** shift\_reg**[**8**]** **^** shift\_reg**[**7**]** **^** shift\_reg**[**6**]** **^** shift\_reg**[**5**]**

**^** shift\_reg**[**3**]** **^** shift\_reg**[**2**]** **^** shift\_reg**[**4**]** **^** shift\_reg**[**1**]** **^** shift\_reg**[**0**];**

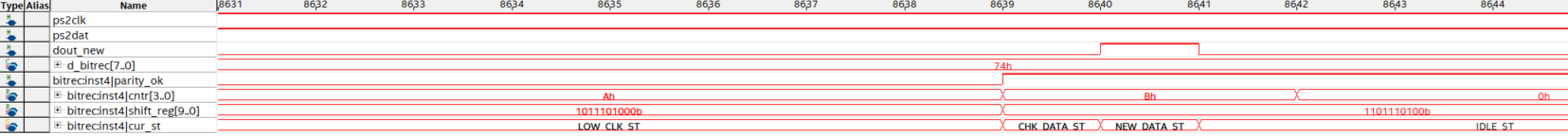
**endmodule**

**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:***

## 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.**

**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

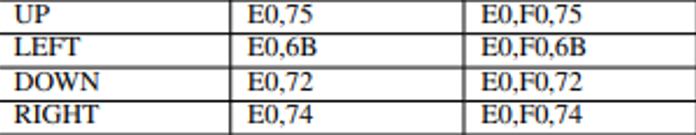
D\_bitrec = 74h  
keycode = 074h

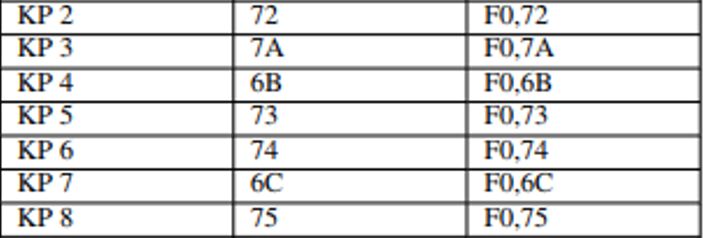
ימינה חדש

D\_bitrec = E0h74h  
keycode = 174h

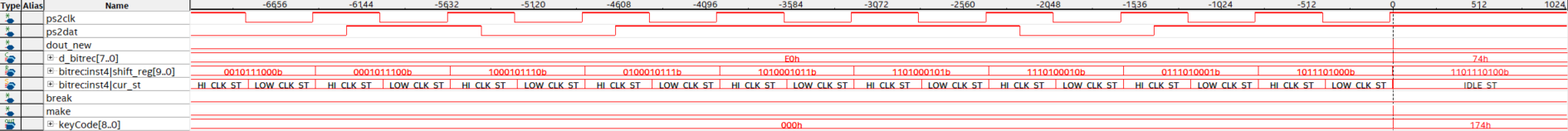
בצע ZOOM OUT והראה את כל התווים שנקלטו מהמקלדת החל מה START BIT של התו הראשון (מצורפת חלק מהטבלה שבחומר העזר לדוגמה)







**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

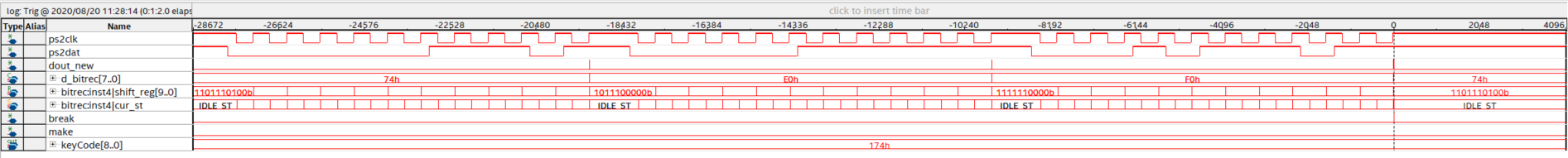
D\_bitrec = F0h74h  
keycode = 074h

ימינה חדש

D\_bitrec = E0hF0h74h  
keycode = 174h

**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: 11:30***

## יצירת סימבול של הממשק למקלדת

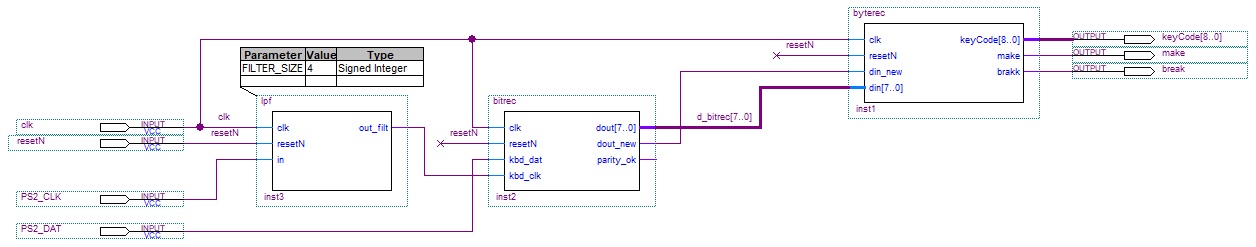
**ליצירת סימבול עבור הממשק למקלדת שמור** את הקובץ הגרפי בשם אחר, KBDINTF.bdf.

בקובץ החדש בטל את יציאות העזר (כולל יציאות לתצוגת 7-Segment).

**נתק** גם את ה KBD EMULATOR והשאר רק את שלושת הרכיבים שמרכיבים את הממשק למקלדת.

**החזר** את היציאת של חיבורי המקלדת האמיתיים, PS2\_CLK ו- PS2\_DAT.

דוגמה



**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***

# 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:

1. לחיצה על מקש SPACE תדליק את LEDR(0) כל זמן הלחיצה.
2. לחיצה על מקש אחר לא תשפיע.

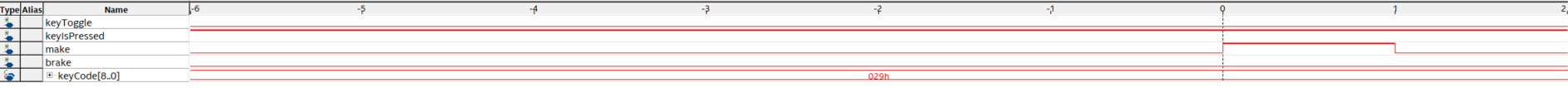
**אפשר לשנות** את אחד הקבועים של הרכיב **keyToggle\_decoder** לקוד מקש שממומש באמולטור, על ידי:

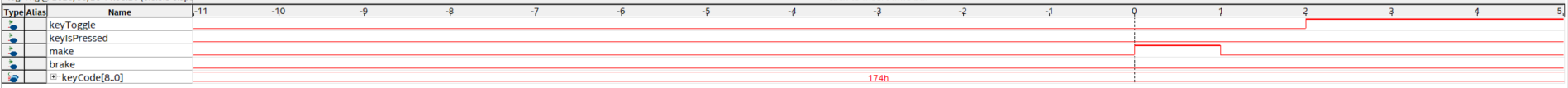
1. מקש ימני על טבלת PARAMETER בשרטוט, בחירת properties 🡪parameter
2. מקש ימני על value ושינוי ערכו לקוד המקש המבוקש.

**רשום** את המקש שנקבע ולאיזו פעולה keyIsPressed) או keyToggle).

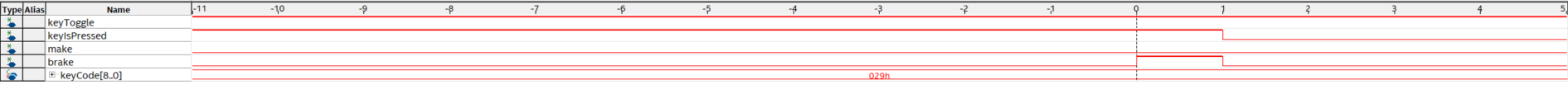
|  |  |  |
| --- | --- | --- |
| מקש | קוד ב- HEX | פעולה |
| SPACE | 29h | keyIsPressed |
| Extended SPACE | 129h | keyToggle |

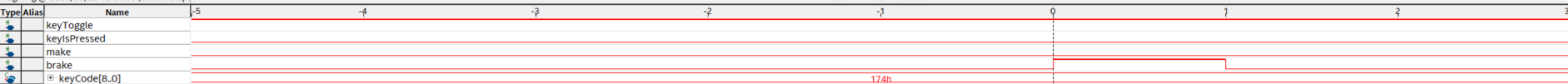
**הפעל את הנתח הלוגי על TOP\_KBD\_DEMO.bdf והוסף אותות לפי הצורך. הצג את שתי היציאות של רכיב keyToggle\_decoder עם תנאי דרבון make.**





**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***

# גיבוי העבודה

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

**תגבה** את קובץ הארכיב וגם העלה אותו למודל למקום המתאים.

**שמור וגבה** את הדו"ח שלך רגיל.

שמור את הדוח גם כ- PDF והעלה אותו למודל.

# פרויקט

חלוקת הנושאים:

|  |  |
| --- | --- |
| *נושא הפרוייקט* | *בתקווה Witcher 3: WILD HUNT* |

***רשום את השעה בה סיימת את המעבדה: 12:08***