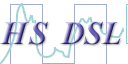
הטכניון – מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל



מעבדות 1, 1ח

|  |
| --- |
| **ניפוי תקלות בחומרה (DEBUG)**  **דו"ח מכין - שאלות ותרגילי הכנה** |



הניסוי פותח בחסות המעבדה למערכות ספרתיות מהירות

גרסה 1.0 (אביב 2018)

עורכים: ארמנד שוקרון, ליאת שורץ

על פי החוברת המקורית של עמוס זסלבסקי

הנחיות

* קובץ זה הוא גם תבנית לדו"ח המכין, יש לשמור ב- PDF ולהגיש במודל.

|  |  |
| --- | --- |
| תאריך הגשת דו"ח ההכנה | 12/5/18 |
| שם המדריך | דודי |

|  |  |  |
| --- | --- | --- |
| סטודנט | שם פרטי | שם משפחה |
| 1 | חגי | קריטי |
| 2 | יהונתן | סרלואי |

תוכן עניינים של דו"ח מכין   
ניפוי תקלות בחומרה (DEBUG)

[1 מכונת RANDOM 2](#_Toc476300016)

[2 ממשק למקלדת 3](#_Toc476300017)

[2.1 תכן יחידת ה - **BITREC** 3](#_Toc476300018)

[2.2 סימולציה 8](#_Toc476300019)

[3 חישוב עומק הזכרון עבור הנתח הלוגי 8](#_Toc476300020)

[4 מטלת תכן עם מקלדת (זיהויNUMLock ) 9](#_Toc476300021)

[5 פרויקט 13](#_Toc476300022)

[5.1 סכמת מלבנים 13](#_Toc476300023)

[5.2 רשימת תהליכים (מלבנים) עיקרית 13](#_Toc476300024)

[5.3 סיפתח (אִסְתִפְתָאח =  ) 14](#_Toc476300025)

# מכונת RANDOM

בהתייחס למכונה ליצירת מספר אקראי RANDOMשתוארה בחומר הרקע ענה על השאלות הבאות:

**א. הסבר מדוע היציאה RANDOM[7..0] היא מספר אקראי?**

**תשובה: מכיוון שה – count8 סופר מהר מאוד ביחס (50MHz) לקצב הלחיצה על מקש key0 כל לחיצה עליו תוציא מספר אקראי כלשהו בין 255 ל0**

**ב. מהו תפקידה של היחידה vrise?**

**תשובה: להוציא '1' בכל פעם שיש עליה בpulse- ולאפשר שמירה בregout-**

**ג. למה משמש הקבוע MAXCOUNT?**

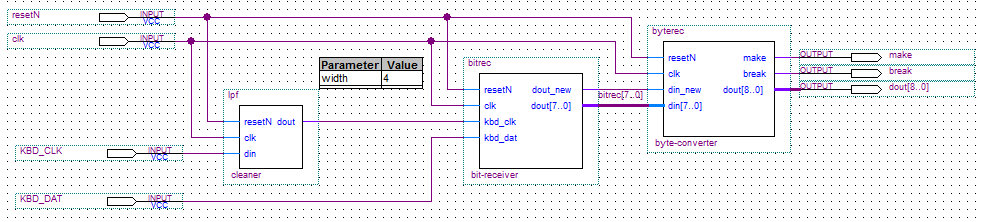
**תשובה: לטעינה חדשה של ה-counter**

**ד. כיצד ניתן לשנות את המכונה כך שתגריל מספרים זוגיים בלבד?**

**תשובה: ניתן לאפס את הLSB וכך נקבל מספרים זוגיים בלבד**

# ממשק למקלדת

כפי שהוסבר בחומר הרקע לניסוי זה, התכן הסינכרוני הבא נבחר למימוש **ממשק חומרה למקלדת**.



כל אחת מהיחידות הנ"ל כתובה בשפת VHDL ותשמש לבנית הממשק שלך למקלדת במעבדה זו.

להלן הקבצים שבהם תעשה שימוש:

1. יחידת מסנן מעביר נמוכים: lpf.vhd, lpf.bsf – נתונה לך במודל
2. יחידת המקלט ברמת ה Bit: bitrec.vhd - נתון שלד שלה במודל
3. יחידת המקלט ברמת ה – Byte: byterec.vhd, byterec.bsf – נתונה לך במודל

**הערה חשובה: יש להביא למעבדה את כל הרכיבים אותם אתה כותב במסגרת עבודת ההכנה.**

**צור פרוייקט חדש** בשם KBDINTF, העתק אליו את הקבצים הנ"ל מהמודל.

## תכן יחידת ה - **BITREC**

**רקע למטלה**: כמו שהוסבר בחומר הרקע תפקידה של היחידה שמטפלת בתשדורת הטורית BITREC הוא, להפיק מהמידע הטורי שמגיע לכניסות kbd\_clk ו kbd\_dat, מידע מקבילי ביציאה dout, יחד עם יציאת חיווי שפעילה למשך מחזור שעון אחד ושנקראת dout\_new. דיאגרמת הזמנים הבאה מתארת אותות אלו אחד ביחס לשני וביחס לאות השעון:



|  |
| --- |
| **נתון לך הקובץ bitrec.vhd שהוא שלד המכיל את כל החלקים הדרושים כפי שהוסבר בחומר הרקע פרט למכונת המצבים.**  **שים לב! השתמש אך ורק בקובץ הנתון לך כעת במודל ולא בגרסאות אחרות מסמסטרים קודמים!**  **הוסף לקובץ זה את הקוד של מכונת המצבים, כפי שתתואר להלן, במקומות בקובץ שבהם כתובה ההערה:**  **---- WRITE YOUR CODE HERE** |

**מכונת מצבים** (מסוג Moore) משמשת כבקר של היחידה. דיאגרמת המצבים הבאה מתארת את התנהגותה.



בדיאגרמה הנ"ל השתמשנו **בקיצורים** הבאים:

* clk מציין את האות Kbd\_CLK בגבוה, ו- clk! בנמוך
* Data מציין את האות Kbd\_DAT בגבוה, ו- Data! בנמוך
* ok מציין את הסיגנל parity\_ok במצב true
* Error מציין את הסיגנל parity\_ok במצב false
* counter מונה את מספר הביטים של קוד המקש שמגיעים בקו הסריאלי

**הדרכה ודרישות:**

**כתוב קוד ב- VHDL** המתאר את מכונת המצבים באמצעות תהליך סינכרוני בלבד. פתח את הקובץ bitrec.vhd תוך הוספה שלו לפרויקט הקיים KBDINTF והגדר אותו כהיררכיה עליונה. הוסף לקובץ bitrec.vhd את הקוד שלך בלבד בהתאם להנחיות להלן.

**שים לב: אין צורך לשנות חלקים אחרים משלד הקוד הנתון ב- bitrec.vhd!**

**חשב מהו NUM\_OF\_BITS.**

**תשובה**: 11bits = 8databits + 1paritybit + 1startbit + 1stopbit

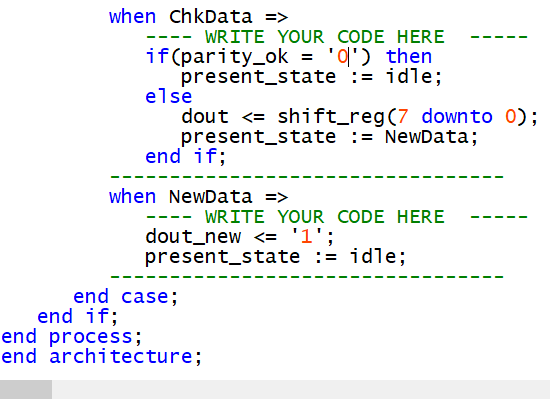
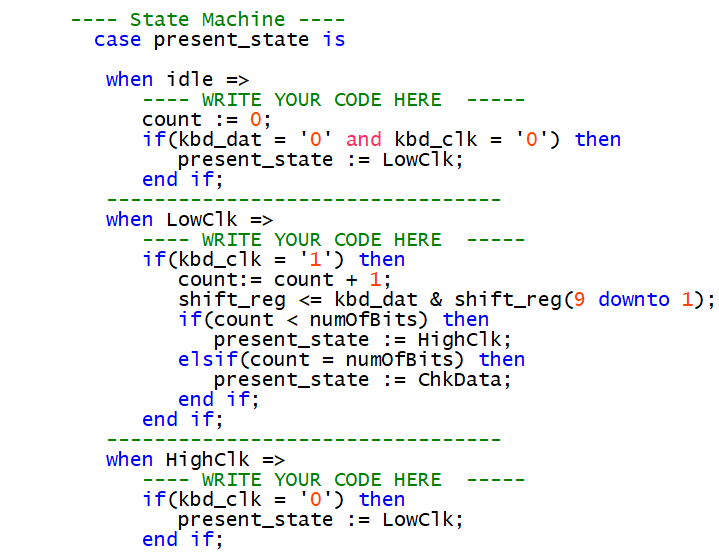
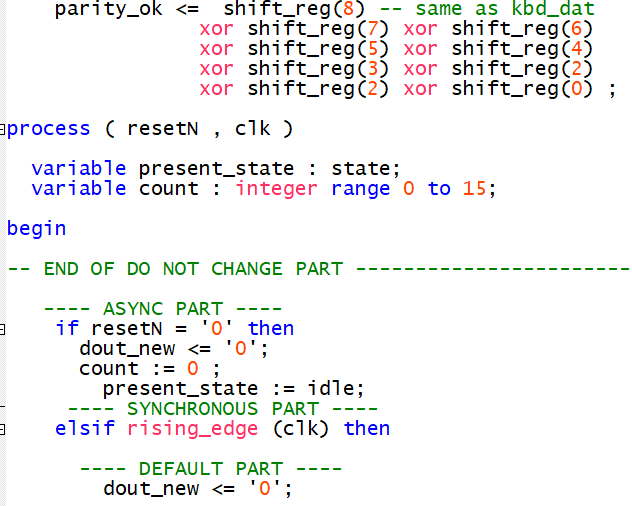
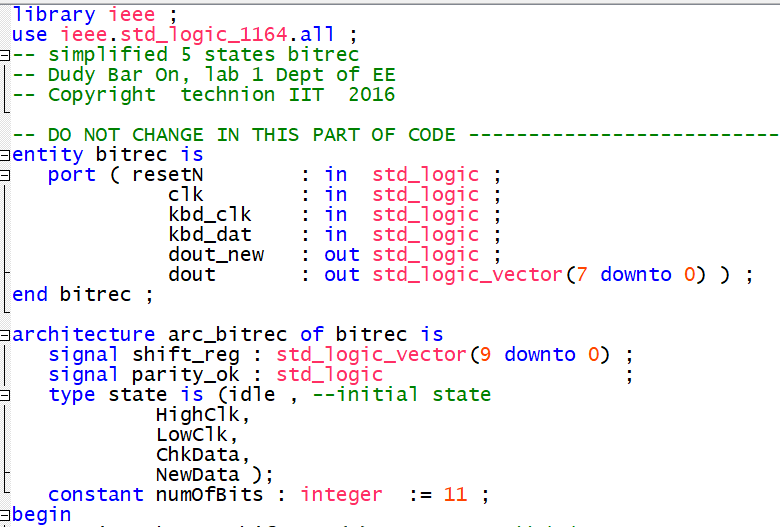
בטבלה הבאה מפורטים המצבים שבמכונה והפעולות לביצוע בכל מצב.

**מלא את העמודה האחרונה בטבלה לפי הדוגמה שבשורה הראשונה:**

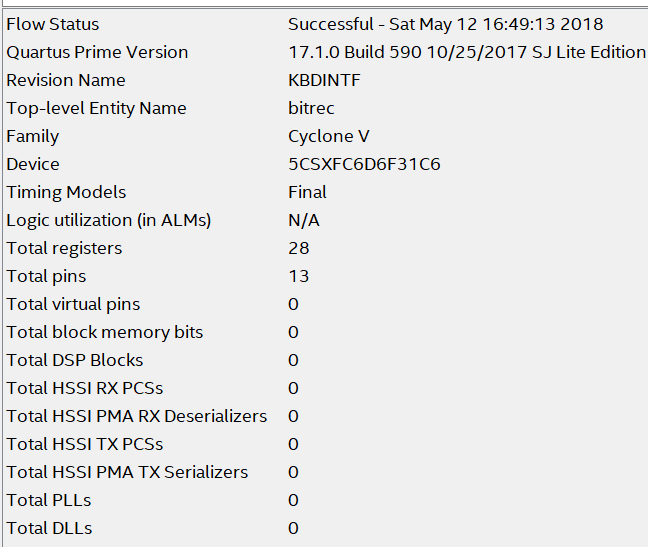
|  |  |  |
| --- | --- | --- |
| **שם המצב** | **פעילות עיקרית** | **לאיזה מצב עוברים מהמצב הנוכחי ובאילו תנאים – למלא את התאים הריקים** |
| Idle | מאפסים את המונה count. ממתינים לתו חדש: אם יש ירידה באות השעון Kbd\_CLK וגם באות הנתונים Kbd\_DAT אז עוברים למצב הבא. | **עוברים** ל**-** LowClk  **עם** ירידה בשעון Kbd\_CLK וגם ירידה ב- Kbd\_DAT (סימן שמתחיל להגיע תו חדש) |
| LowClk | ממתינים לאות שעון גבוה כי זה אומר שהביט הבא כבר הגיע.  אם Kbd\_CLK גבוה:  - משרשרים למקום האחרון ברגיסטר ההזזה shift\_reg את הסיבית החדשה שהגיעה מה- Kbd\_DAT.  shift\_reg <= kbd\_dat & shift\_reg(9 downto 1);  - מקדמים את המונה count ב-1  - ובודקים אם הגיעו כל הביטים. אם כן עוברים למצב בדיקת הנתונים אם לא מחכים לירידת השעון הבא כדי להמשיך לקבל ביטים. | **עוברים ל-HighClk**  **אם Kbd\_Clk גבוה וגם count<numOfBits**  עוברים לChkData אם Kbd\_Clk גבוה וגם count = numOfBits |
| HiClk | ממתינים לביט הבא. אם מגיע ביט, מסומן ע"י ירידה ב- Kbd\_CLK עוברים למצב הבא, קבלת הביט. | **עוברים ל-LowClk**  **אם kbd\_clk נמוך** |
| ChkData | בודקים את נכונות הנתונים ובהתאם לתוצאת הבדיקה עוברים למצב הבא. רק אם בדיקת הזוגיות (ה- parity) טובה מעדכנים את המוצא בתכולת הרגיסטר  dout <= shift\_reg(7 downto 0); | **עוברים ל-NewData**  **אם parity גבוה**  עוברים ל – idle  אם parity נמוך |
| NewData | מודיעים על מילה חדשה  dout\_new <= '1’;  ועוברים מצב | **עוברים ל-idle**  **עם עליית שעון** |

**בצע קומפילציה.**

**צרף את הקוד של BITREC הכולל את מכונת המצבים המלאה:**



**צרף לכאן צילום מסך של תוצאות קומפילציה מוצלחת של המעגל.**



**צור SYMBOL** של קובץ זה אחרי קומפילציה מוצלחת.

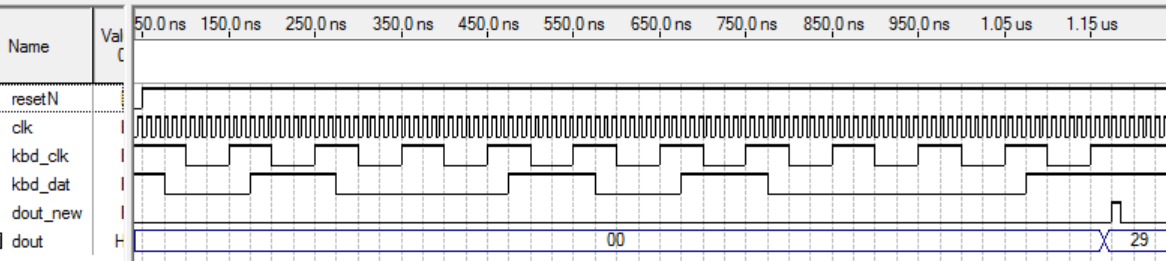
## סימולציה

**בצע סימולציה** ב- Quartus כדי לדבג את מכונת המצבים שתכננת.

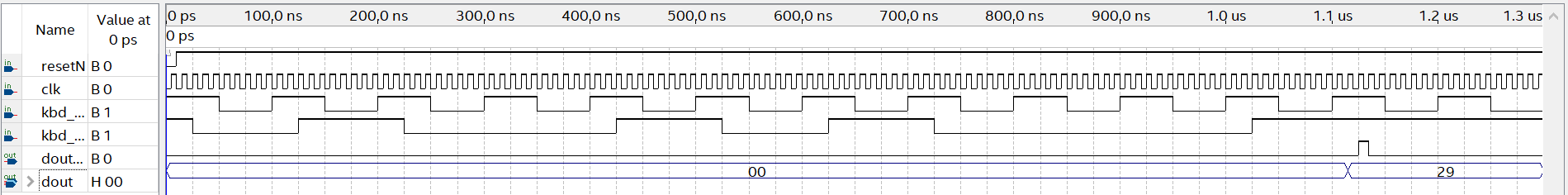
**הדרכה לסימולציה: מומלץ** להגדיר את **שעון** המערכת (clk) מהיר פי 10 משעון המקלדת (Kbd\_CLK): למשל, קבע בשעון המערכת period=10nsec ובשעון המקלדת period=100nsec. השתמש בגודל **grid** של nsec 25 ושים לב שהשינוי ב- Kbd\_DAT מתרחש בזמנים ששעון המקלדת ב- '1' לוגי!

**הראה שבסימולציה** שלך התוצאות זהות לדוגמה הנתונה להלן. הראה שאם מכניסים רצף טורי של קוד מקש נתון ב- Kbd\_DAT, למשל **H29 הקוד של מקש הרווח**, מתקבל ב- dout H29 מקבילי (הצג אות זה ב- radix hexadecimal) ומתקבל '1' במשך מחזור שעון אחד שמודיע על מקש חדש ב- dout\_new אחרי שה- clk האחרון הסתיים (אחרי ה- Stop bit).

**חשוב מאד: לביצוע הסימולציה יש להזין אך ורק את אות המבוא KBD\_DAT כפי שנתון בדוגמה להלן ובקובץ הנתון לכם במודל!**



**צרף לכאן צילום מסך של תוצאות סימולציה מוצלחת.**



# חישוב עומק הזכרון עבור הנתח הלוגי

**רקע למטלה:** על מנת לדבג את המערכת רוצים לדגום באמצעות הנתח הלוגי את אות המבוא Kbd\_DAT של יחידת ה- BITREC בזמן הקשה על מקש כלשהו.

ברוב המקשים קוד המקש מכיל 11 סיביות, אך במקשים מהסוג החדש, הקוד מכיל 11 סיביות נוספות ומחזור שעון הפסקה (למשל הקוד של מקש Down Arrow מהסוג החדש הוא  
72 E0)). כמו כן, שעון המקלדת Kbd\_CLK, שמשמש לסנכרון סיביות הנתונים של Kbd\_DAT, עובד בתדר של 12.5 KHz. **לביצוע החישוב היעזר בהסבר המפורט מחומר הרקע**.

**חשב מה צריך להיות עומק הזכרון המינימלי בנתח הלוגי הדרוש לקליטת כל הקוד במקרה זה.**

**חישוב ותשובה:**

**NumOfBits\*Time\*Frequency = Memory**

**(11+1+11)\*(1/12.5K)\*50M = 92K**

**לכן נבחר 128K ע"פ האפשרויות מהחומר רקע.**

# מטלת תכן עם מקלדת (זיהויNUMLock )

**רקע למטלה**: יישומים המשתמשים במקלדות בדרך כלל מקצים למקשים תפקידים מיוחדים של החלפת תפקידיהם של מקשים אחרים במקלדת, כמו מקש ה- NumLock. מקש זה גורם בדרך כלל להחלפה של מספרים וחיצים. כל הקשה עליו, גורמת להחלפה אחת בלבד של המצב, גם כאשר מדובר בלחיצה ארוכה.

בחומר העזר **נתונה דוגמא** ב-VHDL למימוש ישום המשתמש במקש ה- CapsLock להחלפה אחת של מצבו של לד בלוח DE10, בין הדלקה לכיבוי.

**ממש** מערכת שמבצעת פעולה דומה למקש NumLock, **בעזרת מכונת מצבים**. כל לחיצה על מקש זה (גם אם מדובר בלחיצה ארוכה) תגרום להחלפה אחת בלבד של מצב הנורית בלוח DE10.

**הערה חשובה**: הקבוע "001011000"**)**) מופיע בגוף הקוד ולא כCONSTANT, עליכם לתקן שגיאה זו גם כן.

**שים לב** שיישום זה ישתמש במערכת הממשק למקלדת שיצרת קודם, ז"א אותות המבוא שלו הם אותות המוצא של הממשק למקלדת (מהמוצא של BYTEREC).

**בהמשך, המערכת תידרש לתמוך במקש כלשהוא לא רק ב-NUMLOCK ולכן עליך לאפשר הזנה של קוד המקש המבוקש ממודול חיצוני, כגון רכיב LPM\_CONSTANT (אותו תחבר למערכת במעבדה).**

**הדרכה ודרישות:**

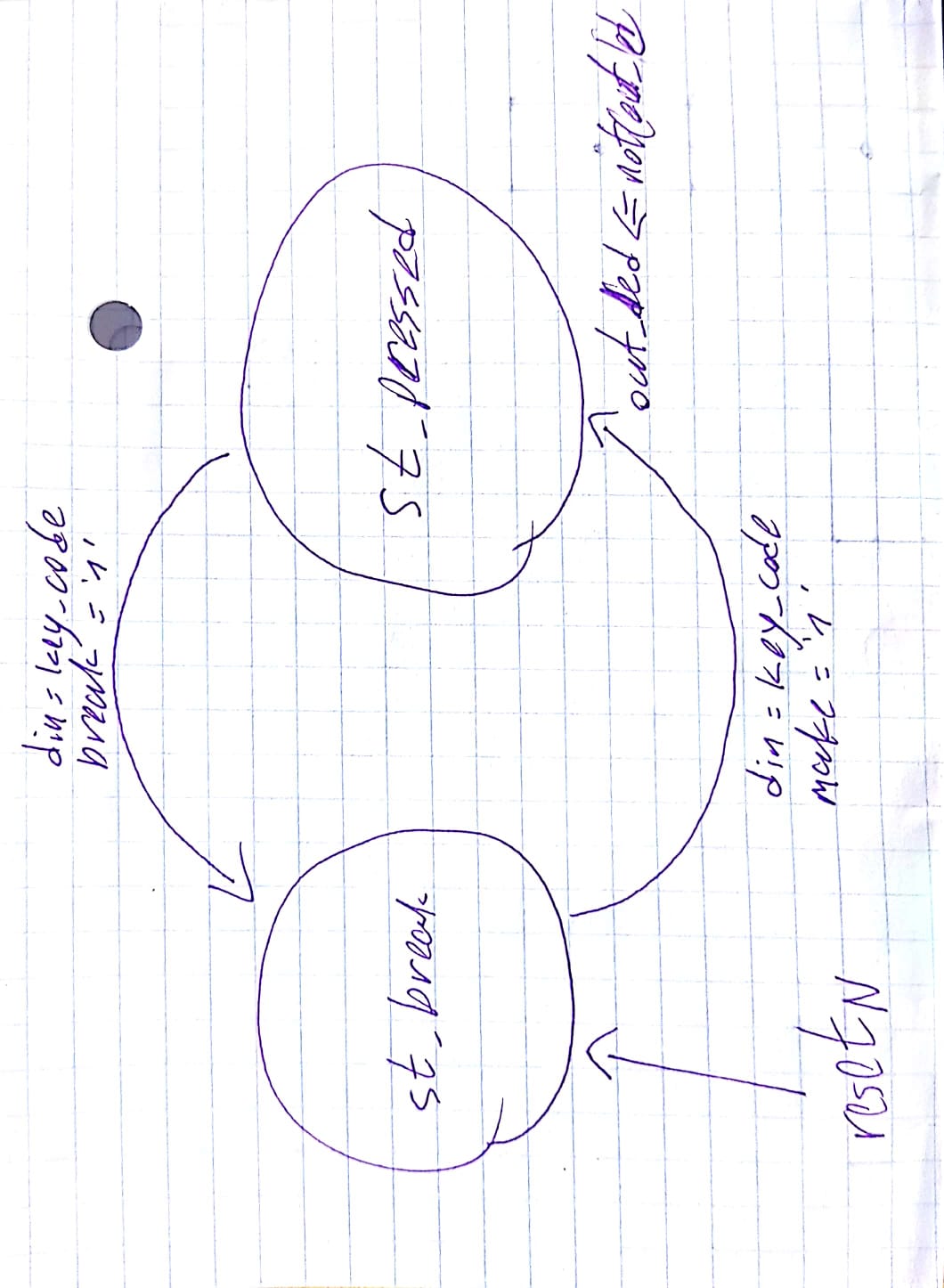
**פתח** קובץ VHDL חדש ב – Quartus, באותו פרויקט KBDINTF. קרא לקובץ בשם NUM\_LOCK.VHD וודא שהוא שייך לפרויקט והוא מוגדר **כהיררכיה עליונה**.

**כתוב** את הקוד של ישום ה- NumLock בשפת VHDL בקובץ שפתחת. תוכל להיעזר בדוגמת הקוד לישום כזה הנתונה בנספח של חומר הרקע לניסוי.

**בשונה מהדוגמה** הנתונה, בתוכנה שלך הגדר **וקטור כניסה** **in** בשם key\_code באורך 8 סיביות, שיקבל את קוד המקש הרצוי.

**בשונה מהדוגמה** הנתונה, יש לתכנן את היישות בעזרת **מכונת מצבים**.

**צרף לכאן דיאגרמת מצבים עליה התבססת למימוש היישות.**

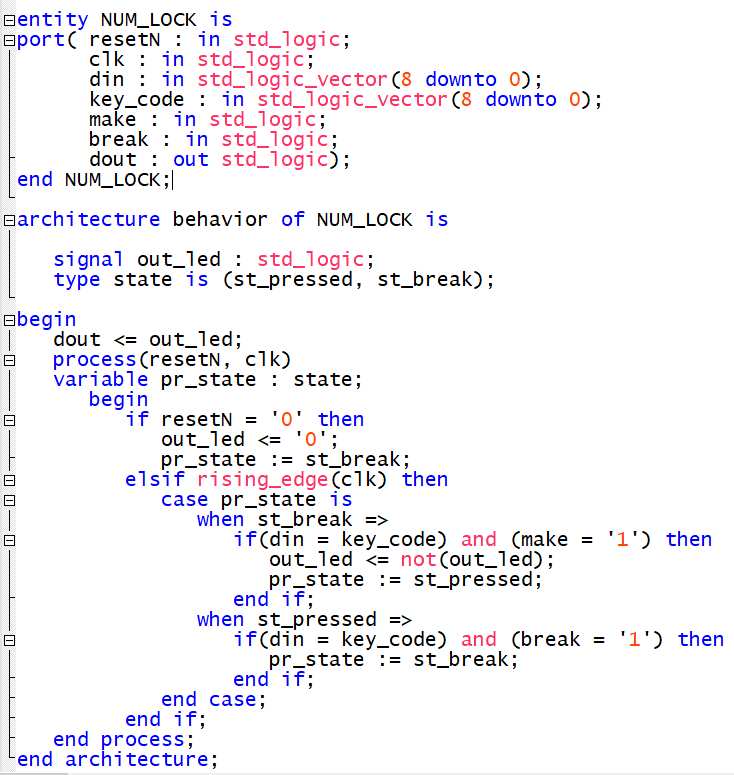


**שים לב ששם ה- ENTITY צריך להיות זהה לשם הקובץ.**

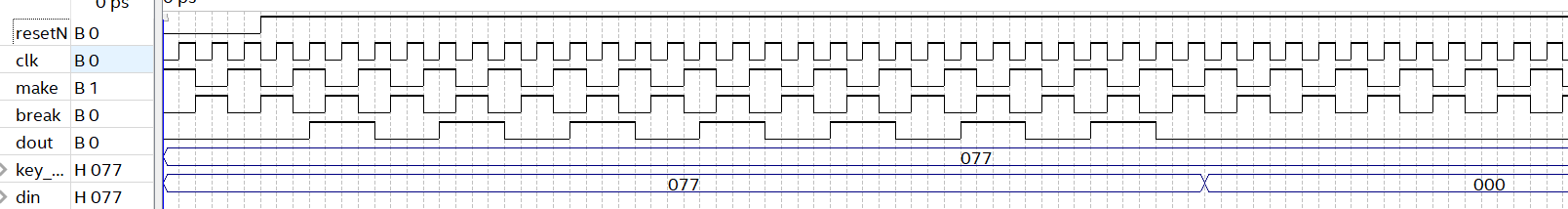
**קמפל** את הקוד. אחרי שהקומפילציה עברה בהצלחה, **צור** SYMBOL של קובץ זה.

**בצע סימולציה** והראה שהיישום עובד כמתוכנן. הראה שבכל לחיצה על מקש ה- NumLock (קוד H77) יש שינוי אחד במצב הנורית led\_out (מ- '0' ל- '1' ואחר כך מ- '1' ל- '0'). לשם כך הזן ל- din ול- key\_code את אותו מספר, H77.

**צרף את קטע הקוד הרלוונטי להדלקה/כיבוי לד בכל הקשה על מקש ה-** NumLock **:**

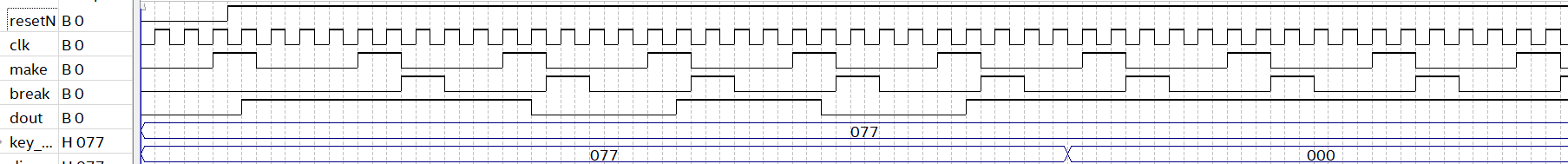


**צרף לכאן צילום מסך של תוצאות סימולציה מוצלחת.**



**בצע סימולציה נוספת** והראה שבלחיצה ארוכה על מקש ה- NumLock יש רק שינוי אחד במצב הנורית. לחיצה ארוכה על מקש מדמים על ידי מספר פולסי make רצופים (הראה לפחות שנים) ללא פולס break ביניהם.

**צרף לכאן צילום מסך של תוצאות סימולציה מוצלחת.**



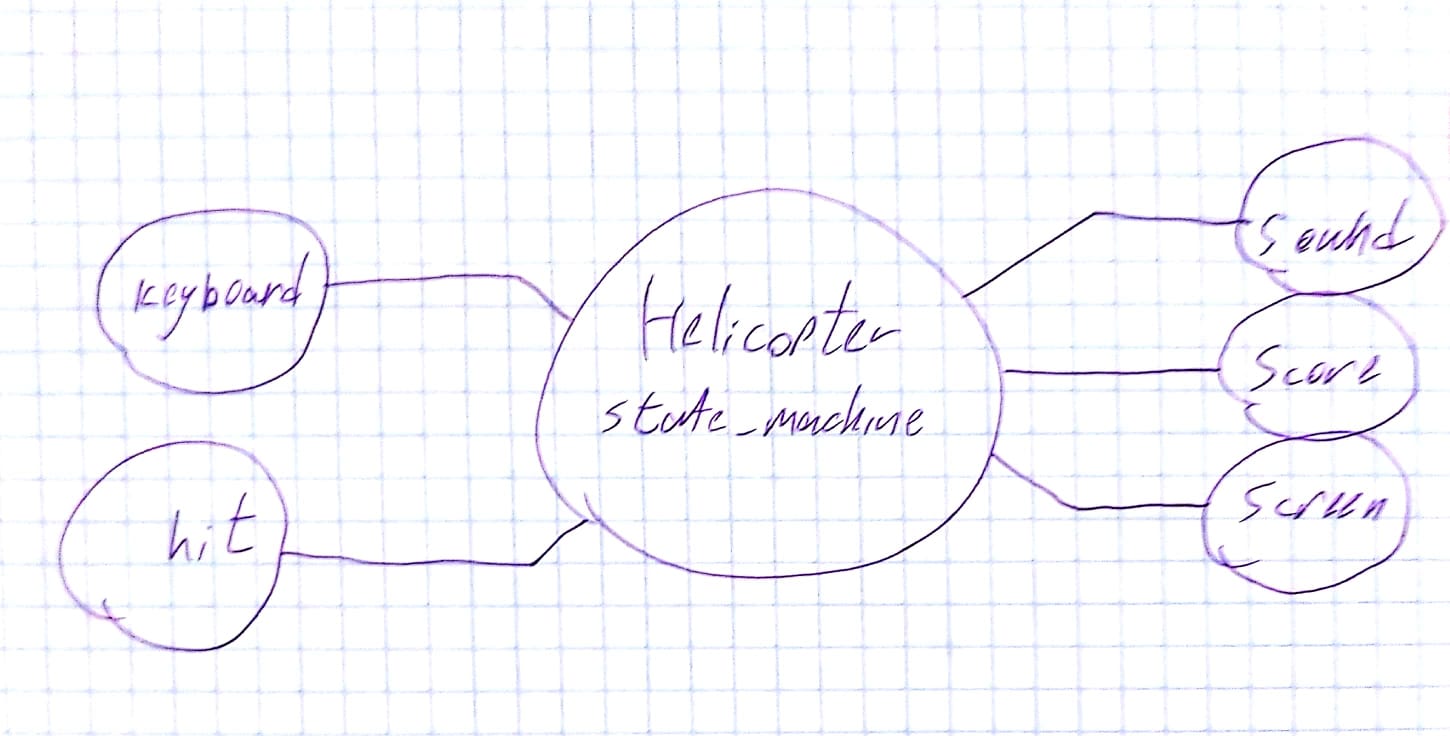
# פרויקט

(לא חלק מהציון של דו"ח זה)

|  |
| --- |
| *נושא הפרוייקט* |
| *Helicopter* |

## סכמת מלבנים

הוסף סכמת מלבנים עיקרית של הפרוייקט – 5-10 מלבנים משמעותיים



## רשימת תהליכים (מלבנים) עיקרית

רשום את כל הרכיבים (תהליכים) העיקריים בפרוייקט, לכל רכיב רשום את תפקידיו ואת הכניסות והיציאות העיקריות

|  |  |  |  |
| --- | --- | --- | --- |
| שם | תאור | כניסות | יציאות |
| 1. Keyboard | ממשק של המקלדת למשחק והגדרת תפעול כללי של המקשים | ממשק למקלדת מחשב | Make, brake, key\_data\_out |
| 1. Game\_templates | הגדרת מספר טמפליטים של מסך המשחק בזכרון | State\_machine |  |
| 1. hit | תצורת מסך נוכחית דינאמית הכוללת את תזוזת חלקי המשחק | State\_machine  Game\_templates | State\_machine |
| 1. Helicopter\_state\_machine | הגדרת מעבר בין מצבי המשחק השונים ( התחלה וסיום משחק, פסילה וכו,) | חיווי מהמקלדת ותצורת המסך הנוכחית ( או מטמפלייט בתחילת ובסוף המשחק) | Sound, score, present\_state |
| 1. sound | מימוש התנהגות הצלילים במשחק | State\_machine | רמקול |
| 1. score | תצוגת הניקוד במשחק | State\_machine | 7seg |
| 1. screen | תצוגת המשחק | State\_machine  Game\_templates | VGA |

## סיפתח (אִסְתִפְתָאח = )

הגדר מהו החלק שתממש כסיפתח לפרוייקט,

תצוגה המכילה מסך פשוט והוספת מסוק במיקום קבוע בלחיצה של לחצן ספציפי

רשום את כל התליכים (מלבנים) העיקריים בסיפתח , לכל רכיב רשום את תפקידיו ואת הכניסות והיציאות העיקריות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| שם | תאור | כניסות | יציאות | |
| 1. מסך |  |  |  |
| 1. מקלדת |  |  |  |
| 1. Game\_templates |  |  |  |
| 1. State\_machine |  |  |  |

שרטט את ההירארכיה העליונה של הסיפתח – אין צורך לממש

***לאחר שסיימת - לחץ על ה LINK ומלא בבקשה את השאלון המצורף***

|  |
| --- |
|  |
| |  | | --- | | [**מלא את הטופס**](https://docs.google.com/forms/d/1tO1v_J1GNnuBFqGURbuZZzX8uuGXzKp8RdkdBOaakfY/viewform?c=0&w=1&usp=mail_form_link) | |
|  |