**הטכניון - מכון טכנולוגי לישראל**

**הפקולטה להנדסת חשמל**



מעבדה 1א'

ממשק מסך VGA וצלילים

תדריך מעבדה

גרסה .41

חורף 2019-20

נכתב על ידי: דודי בר-און, אלכס גרינשפון, ליאת שורץ

|  |  |  |  |
| --- | --- | --- | --- |
| מועד | ביצוע עד סעיף | שם המדריך בפועל | תאריך |
| ביצוע הניסוי |  |  |  |
| השלמת חלקים חסרים -1 |  |  |  |
| השלמת חלקים חסרים -2 |  |  |  |

|  |  |  |
| --- | --- | --- |
| סטודנט | שם פרטי | שם משפחה |
| 1 | אביב | כספי |
| 2 | בר | אלבו |

**תוכן עניינים**

[1 כללי 3](#_Toc25513274)

[2 הכרת פלטפורמת ה VGA 3](#_Toc25513275)

[2.1 חיבור המערכת 3](#_Toc25513276)

[2.2 הפעלת יישום ה- VGA 3](#_Toc25513277)

[3 ביצוע פעולות להרחבת היישום 4](#_Toc25513278)

[3.1 שינוי הרקע הסטטי 4](#_Toc25513279)

[3.2 שינוי BITMAP - הכפלת גודל ה- SMILEY 5](#_Toc25513280)

[4 חיבור מכלולים נוספים إستفتاح -סיפתח 6](#_Toc25513281)

[4.1 הממשק למקלדת 6](#_Toc25513282)

[4.1.1 הוספת הממשק למקלדת לפרויקט 6](#_Toc25513283)

[4.1.2 בדיקת קבלה לממשק המקלדת 7](#_Toc25513284)

[4.2 הוספת אובייקט מעל הרקע הסטטי 8](#_Toc25513285)

[4.3 תרגול שימוש בעורך הזכרון, ה - ISMCE 9](#_Toc25513286)

[4.4 תרגול שימוש בנתח הלוגי, ה-SIGNAL\_TAP 9](#_Toc25513287)

[4.5 הוספת ספרות/אותיות ליישום 10](#_Toc25513288)

[4.6 הוספת צלילים 11](#_Toc25513289)

[4.7 אינטגרציה ובקרת משחק 12](#_Toc25513290)

[5 עבודה על הפרוייקט 13](#_Toc25513291)

[6 גיבוי 13](#_Toc25513292)

***רשום את השעה בה התחלת את המעבדה: 8:34***

**מטרות הניסוי**:

• הכרת ממשק ה- VGA

* הכרת ממשק השמע
* הרחבת היישום
* אינטגרציה עם מקלדת והשמעת צליל

# כללי

הורד מהמודל את קובץ הארכיב של המעבדה ופתח אותו לפרויקט בתיקיה יעודית על הדיסק שלך (לא לפתוח פרויקט ב- DOWNLOADS, כדי שתהיה לך גישה אליו מכל עמדה אחרת).

# הכרת פלטפורמת ה- VGA

## חיבור המערכת

* הפעל את כרטיס הניסוי. תראה שהכרטיס והמסך הנוסף שנמצא בכל עמדת עבודה על המדף (מעל שני המסכים של העמדה) נדלקים.



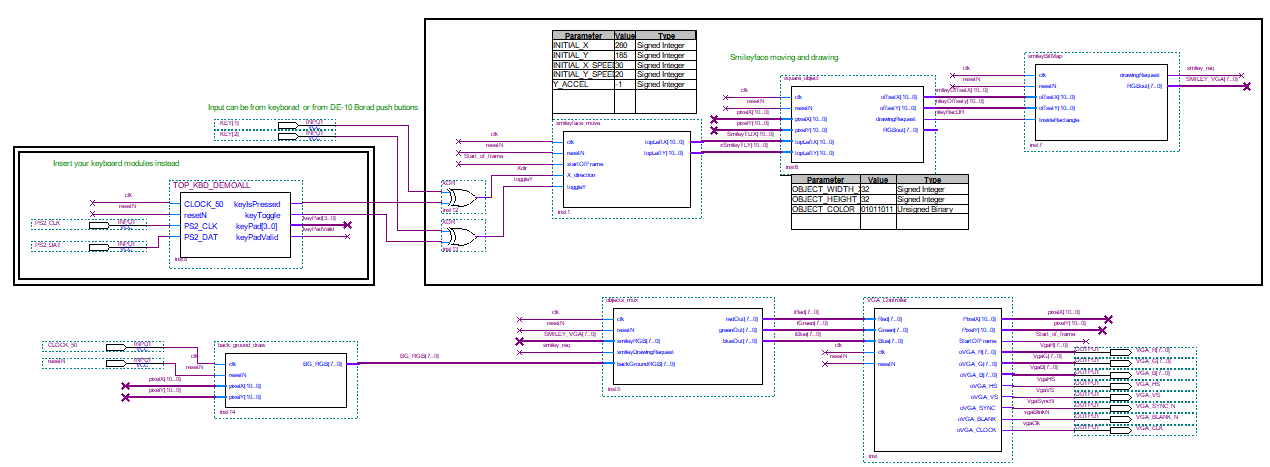
## הפעלת יישום ה- VGA

* הורד מהמודל את קובץ הארכיב של מעבדת VGA ופתח אותו בתיקייה משלו.
* בפרויקט שנפתח פתח את ה- :TOP TOP\_VGA\_DEMO\_WITH\_MSS.bdf.

ה-TOP מחולק לאזורים (כמו באיור הבא) ובכל שלב במעבדה זו נתמקד באזור אחר:

בשלב זה נתמקד בממשק ה- VGA ובמודולים השונים שמרכיבים אותו (ללא הצלילים). תחילה נבדוק שיישום ה- Smiley בכללותו עובד נכון.

* בצע סינתזה ל- TOP, הרץ קובץ הדקים ואז הרץ קומפילציה מלאה.
* צרוב את הפרוייקט לכרטיס, ודא שיישום ה- Smiley עובד נכון ואשר זאת עם המדריך.



ממשק למקלדת

טיפול בפרצופון

רקע סטטי

בורר אובייקטים

בקר ה- VGA

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 8:40

# ביצוע פעולות להרחבת היישום

כדי לא לדרוס את הקובץ המקורי שמור את קובץ היישום הנוכחי בשם אחר, למשל : TOP\_VGA\_DEMO\_WITH\_MSS\_ALL.bdf. מכאן והלאה תעבוד איתו ותוסיף/ תעדכן מודולים כפי שייתואר בהמשך.

## שינוי הרקע הסטטי

**משימה**: להוסיף לרקע הסטטי אלמנטים נייחים.

* פתח את קובץ הרקע הסטטי back\_ground\_draw ונסה להבין את פעולתו. שמור אותו בשם אחר, למשל back\_ground\_drawSquare.sv (עדכן גם שם מודול בהתאם)

**שים לב: שם הקובץ צריך להיות זהה לשם המודול**

* עשה בו את השינוי הבא:
  + הוסף לרקע הסטטי מלבן קטן ובעל צבע כלשהו, כמו באיור:



* קבע את גודל המלבן כרצונך.

מה גודל המלבן שקבעת (בפיקסלים)?

**תשובה** : 56X100

* קבע את צבע המלבן.

מה הצבע שקבעת, כקומבינציה של RGB? הצג את המילה ב- HEX ואת הצבע שיצא.

**תשובה** : **המילה ב- HEX ו- בינארי 00000000 ,00h**

**צבע שחור**

* קבע את מיקום המלבן על המסך.
* הוסף קוד למודול הרקע הסטטי כך שיכלול את המלבן הנוסף.
* קמפל את היישום והורד לכרטיס.
* ראה שהמלבן החדש מוצג ברקע כמו שהתכוונת. אם לא, תקן כנדרש, גודל, צבע וכו'.

הוסף לדו"ח את השינויים ב- SV של מודול הרקע הסטטי החדש

**if** **(**pixelX **<** 400 **&&** pixelX **>=** 300 **&&** pixelY **<** 256 **&&** pixelY **>=** 200**)** **begin**

blueBits **<=** 2'b00**;**

greenBits **<=** LIGHT\_COLOR**;**

redBits **<=** LIGHT\_COLOR**;**

**end**

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 9:00

## שינוי BITMAP - הכפלת גודל ה- SMILEY

**משימה**: לשנות את ה-BITMAP . ספציפית יש להכפיל את גודל הפרצופון (smiley) פי 2 בלי לשנות את גודל הביטמאפ.

* תחילה יש להגדיל את גודל המלבן (במודול המלבן) ל-פי שנים בשני הכיוונים, X ו- Y. לשם כך יש לעדכן את הפרמטרים של המודול (על ידי הקשה כפולה על טבלת הפרמטרים הצמודה למודול).
* פתח את מודול הביטמאפ smileyBitMap.sv ושמור אותו בשם אחר, למשל smileyBitMapX2.
* הבן את אופן פעולתו.
* במודול זה, שנה את גודל הפרצופון כך שכל פיקסל בו יהווה ארבעה פיקסלים (2X2) על המסך, על ידי שינוי האינדקסים של הגישה לBITMAP מבלי לשנות את ה BITMAP עצמו.

**הנחיה**: אפשר להשתמש בפקודת ההזזה ימינה או חלוקה ב- 2 של האינקסים Y,X וע"י כך כל פיקסל מוכפל לריבוע 2X2.

לדוגמה, פקודת ההזזה ימינה של offsetY במקום אחד: **offsetY >> 1**

הוסף לדו"ח את השינויים ב- SV של מודול הפרצופון המוכפל

RGBout **<=** object\_colors**[**offsetY **>>** 1**][**offsetX **>>** 1**];** //get RGB from the colors table

**שים לב!** מכאן והלאה המשך לעבוד עם הפרצופון המוכפל.

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 9:10

# חיבור מכלולים נוספים إستفتاح -סיפתח

## הממשק למקלדת

**משימה**: לחבר את המקלדת למערכת, על ידי הוספתו לפרוייקט זה של הממשק למקלדת שבנית במעבדת DEBUG.

### הוספת הממשק למקלדת לפרויקט

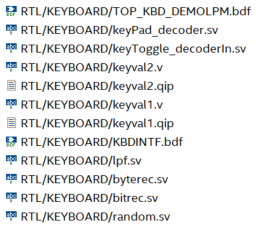
נתון לך מודול TOP\_KBD\_DEMOALL שמכיל את הקבצים:

* **keyToggle\_decoder.sv –** שהכרת במעבדה קודמת
* **keyPad\_decoder.sv** – קובץ נתון לך שמתרגם את המקשים (0-9 ו- a-f) לקוד לביטמאפ אותו ניתן להציג על המסך
* **KBDINTF** – קובץ הממשק למקלדת, שהוא ריק

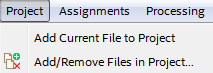
עליך ליבא את הקובץ KBDINTF שבנית במעבדת DEBUG לפרויקט זה.

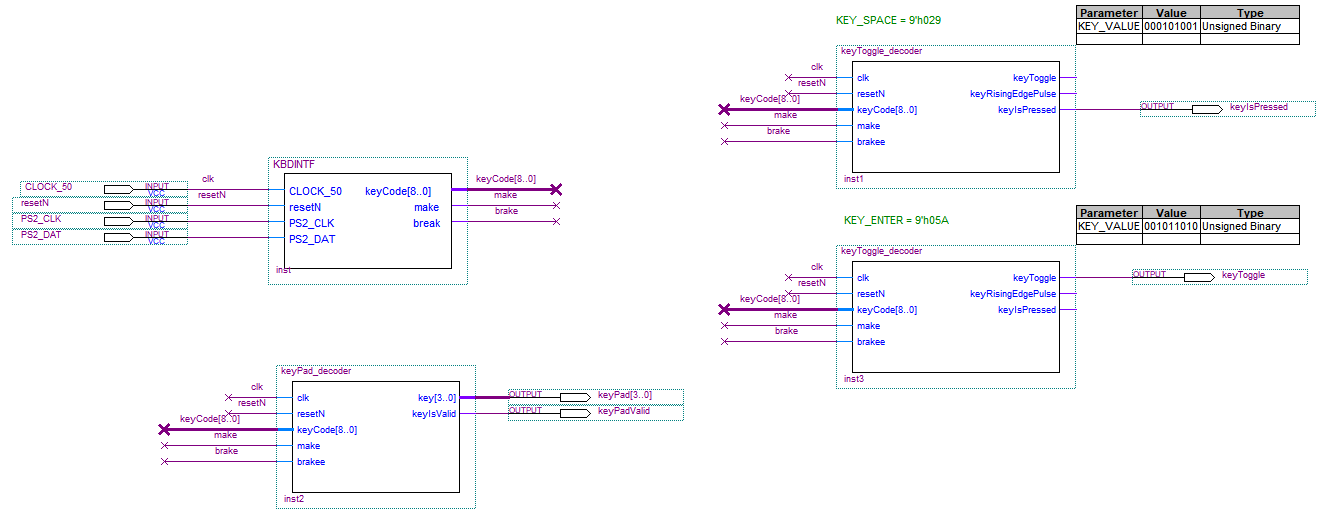
* לשם כך:
  + העתק מפרוייקט המקלדת (ממעבדת DEBUG) את הקבצים הרלוונטיים של הממשק למקלדת, לתיקית הפרויקט הנוכחי RTL/KEYBOARD:

קובץ זה כבר קיים בתיקיה של הפרויקט, יש להחליפו בשלך



* + הוסף את הקבצים שהעתקת לפרוייקט הראשי בעזרת:





נתונים

לעדכן ממשק למקלדת

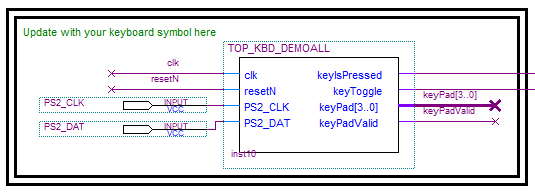
דוגמה

* בצע אנליזה לקובץ TOP\_KBD\_DEMOALL ועדכן את הסימבול שלו   
  (על ידי File – Create/Update – Create Symbol File for Current File ).

### בדיקת קבלה לממשק המקלדת

**משימה**: לבדוק תקינות של פעולת המקלדת ביישום הפרצופון

* חזור להירארכיה העליונה והגדר אותה כ- TOP.
* עדכן את הסימבול של ממשק המקלדת בהירארכיה העליונה של היישום (על ידי מקש ימני על הסימבול ובחירת הפעולה Update Symbol).



דוגמה

* בצע קומפילציה מלאה והורד את היישום לכרטיס.
* בדוק את פעולת המקשים ~~0-9, a-f~~,SPACE ,ENTER .
* שים לב ששני לחצנים הוגדרו כעוקפים של המקלדת. נסה אותם וראה איזה לחצן מתאים לאיזה מקש.

כיצד אתה משפיע על תנועת הפרצופון מהמקלדת ועם הלחצנים?

**תשובה**: על ידי לחיצה על רווח או על key1 אנחנו משנים את מהירות התנועה בציר X

על ידי לחיצה על אנטר או key2 אנחנו משנים את המהירות בציר Y

USER

KBD Intrfc.

Smiley movement

Big Smiley draw

MUX...

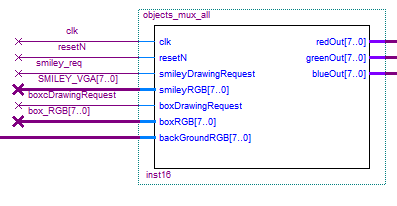
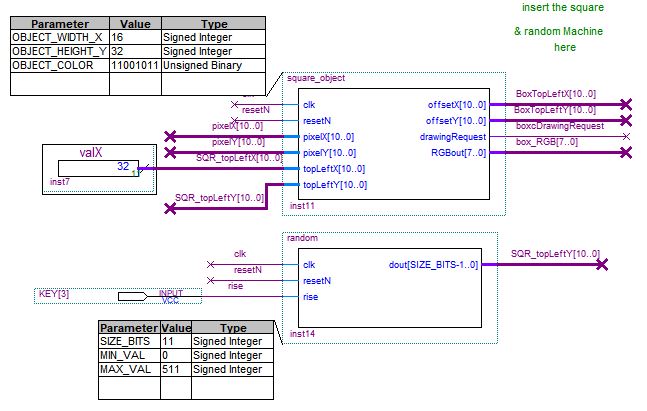
SPACE

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 9:23

## הוספת אובייקט מעל הרקע הסטטי

**משימה**: להוסיף מעל הרקע הסטטי מלבן נייח (אוביקט נפרד), שאפשר לקבוע את מיקומו חיצונית, ולחברו לבורר העדיפויות.

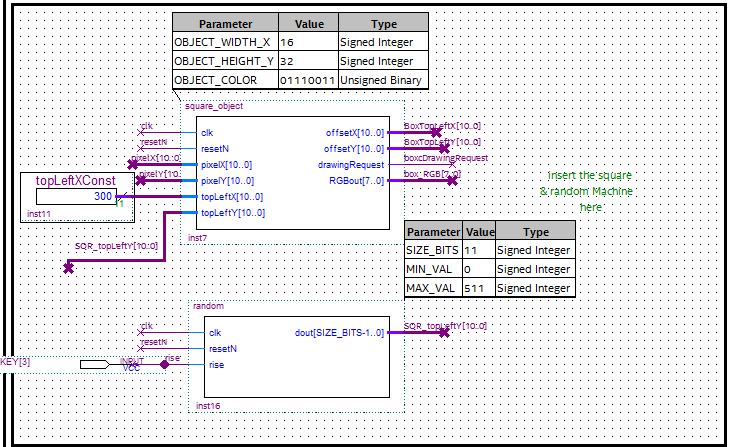
* הוסף ל- TOP רכיב נוסף - של מלבן square\_object (הרכיב קיים בפרויקט).
* קבע את גודלו ל- x = 16, y = 32 וצבעו של המלבן כרצונך, באמצעות פרמטרי הרכיב.
* יש למקם את המלבן החדש במיקום אופקי קרוב למסלול תנועתו של הפרצופון, כך שהוא והמלבן יגעו אחד בשני לפחות לרגע קט במהלך תנועת הפרצופון.
* קבע את הקואורדינטה topLeftX באופן ידני.
  + לשם כך חבר רכיב מסוג LPM\_CONTANT תוך איפשור עריכתו עם עורך הזכרון ה- ISMCE (העזר ב - COOK BOOK).
* הקואורדינטה topLeftY תיקבע באופן אקראי.
  + לשם כך חבר מכונת RANDOM (מודול שהובא לפרויקט יחד עם הממשק למקלדת ממעבדת DEBUG) אשר תופעל בלחיצה על 3KEY.
* כמו כן עליך להוסיף עוד זוג כניסות ולוגיקה מתאימה לבורר העדיפויות לטיפול במלבן. בלוגיקה קבע עדיפותו מעל הרקע ומתחת לפרצופון.
* אחרי שינויים אלה אזור זה ביישום שלך צריך להיראות כך:

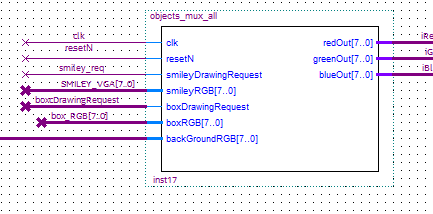
 

הבורר המעודכן

דוגמה

הוסף לדו"ח את תוספת המודולים ל- TOP ופירוט החיבורים





הוסף לדו"ח את ה- SV של מודול ה- MUX עם האובייקט החדש

**module** objects\_mux\_all **(**

// -------- Clock Input

**input** **logic** clk**,**

**input** **logic** resetN**,**

// smiley

**input** **logic** smileyDrawingRequest**,** // two set of inputs per unit

**input** **logic** **[**7**:**0**]** smileyRGB**,**

// add the box here

**input** **logic** boxDrawingRequest**,**

**input** **logic** **[**7**:**0**]** boxRGB**,**

// background

**input** **logic** **[**7**:**0**]** backGroundRGB**,**

**output** **logic** **[**7**:**0**]** redOut**,** // full 24 bits color output

**output** **logic** **[**7**:**0**]** greenOut**,**

**output** **logic** **[**7**:**0**]** blueOut

**);**

**logic** **[**7**:**0**]** tmpRGB**;**

**assign** redOut **=** **{**tmpRGB**[**7**:**5**],** **{**5**{**tmpRGB**[**5**]}}};** //-- extend LSB to create 10 bits per color

**assign** greenOut **=** **{**tmpRGB**[**4**:**2**],** **{**5**{**tmpRGB**[**2**]}}};**

**assign** blueOut **=** **{**tmpRGB**[**1**:**0**],** **{**6**{**tmpRGB**[**0**]}}};**

//======--------------------------------------------------------------------------------------------------------------=

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

**begin**

**if(!**resetN**)** **begin**

tmpRGB **<=** 8'b0**;**

**end**

**else** **begin**

**if** **(**smileyDrawingRequest **==** 1'b1 **)**

tmpRGB **<=** smileyRGB**;** //first priority

**else** **if** **(**boxDrawingRequest **==** 1'b1**)**

tmpRGB **<=** boxRGB**;**

**else**

tmpRGB **<=** backGroundRGB **;** // last priority

**end** **;**

**end**

**endmodule**

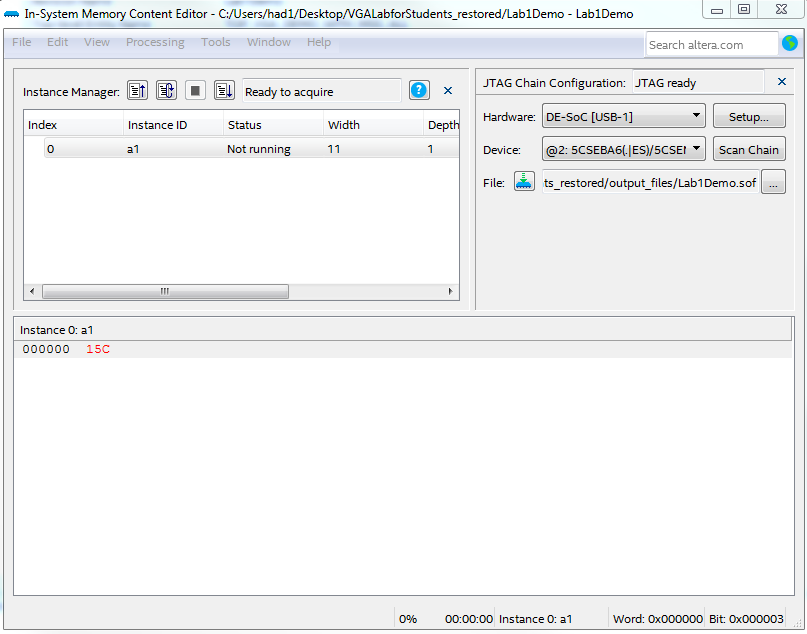
קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 9:50

## תרגול שימוש בעורך הזכרון, ה - ISMCE

**משימה**: לשנות את מיקומו של אובייקט המלבן באמצעות ה- ISMCE.

* הפעל את ה- ISMCE (העזר ב - COOK BOOK).
* שנה את הקואורדינטה topLeftX תוך כדי הפעלת היישום עד שהמלבן חותך את המסלול הפרצופון במקום אחד לפחות.

הוסף לדו"ח את מסך של ISMCE עם הקבוע שקבעת



## תרגול שימוש בנתח הלוגי, ה-SIGNAL\_TAP

**משימה**: לבדוק בעזרת הנתח הלוגי איך קובעים מיקום אקראי למלבן הנוסף

* הפעל את הנתח הלוגי (העזר ב - COOK BOOK) וקבע את כל הפרמטרים שלו.
* הצג בנתח הלוגי את האותות שנראים לך רלוונטים במקרה זה.

כיצד תקבע את תנאי ה- Trigger במקרה זה?

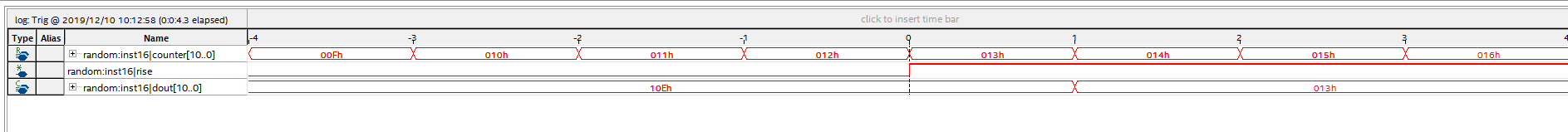
**תשובה**: נקבע את הTRIGGER שיפעל בעליית הערך של Rise ברכיב הRandom

כך נוכל לראות בדיוק את הספירה והערך שיצא מרכיב זה

* לחץ מספר פעמים על **rise** וראה כיצד זה משפיע על מיקום המלבן בזמן שהמערכת פעילה.

**שים לב**: הכניסה **rise** מקבלת אות מלחצן הכרטיס.

הוסף לדו"ח את רכישה מ- SIGNAL TAP במצב תקין



קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 10:13

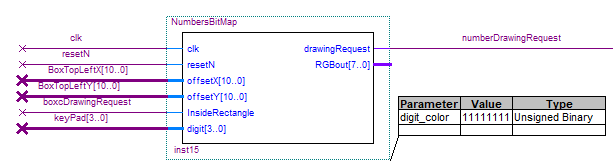
## הוספת ספרות/אותיות ליישום

**מטרה**: להוסיף ספרות/אותיות כאובייקטים גרפיים מעל הרקע.

נתון לך מודול שמכיל ביטמאפ של ספרות ואותיות 0 עד F, **NumbersBitMap**.

* פתח אותו והבן את פעולתו.

דוגמה



* חבר את יציאת הבקשה לשרטוט שלו לכניסת הבקשה לשרטוט של המלבן בבורר האובייקטים במקום זו של המלבן.
* בקשת השרטוט של המלבן נכנסת לכניסת הביטמאפ של המספרים (חיבור זה כבר קיים באמצעות שמות החוטים).

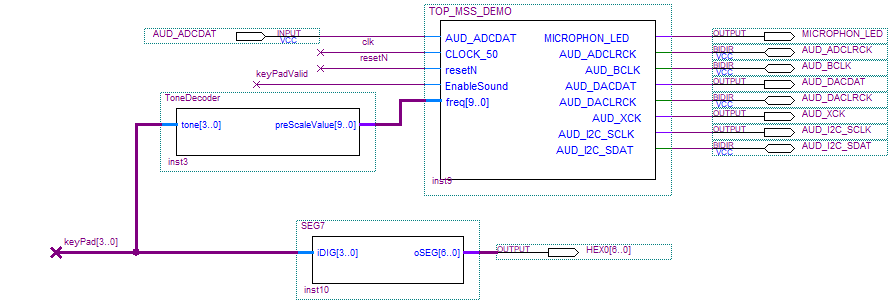
**שים לב**: צורת חיבור זו דומה לצורת החיבור של הפרצופון, מלבן שקובע את גבולות הצורה ומודול שנותן את התוכן שלה, כביטמאפ. לכן מימדי שני הרכיבים האלה צריכים להיות מתואמים.

* קמפל וצרוב לכרטיס.
* בדוק שהספרות /אותיות מוצגות נכון על המסך.
* בדוק את פעולת המקשים 0-9, a-f,

## הוספת צלילים

**משימה**: להפעיל את הצלילים של היישום על ידי הפעלת ממשק השמע.

בשלב זה נתמקד בממשק השמע, המודול TOP\_MSS\_DEMO. פירוט על אופן פעולתו מופיע בחומר הרקע למעבדה זו. בקצרה, יישום זה משתשמש בפלטפורמת ה- MSS של המערכת ומייצר אות דיגיטלי אותו המערכת ממירה לאות אנלוגי אותו ניתן לשמוע באוזניות או רמקולים. במקרה זה האות הוא סינוס דיגיטלי בתדר שהמשתמש יכול לשנות וכך לקבוע את הצליל הנשמע.



דוגמה

* התבונן במודולים של הקובץ TOP\_MSS\_DEMO וזהה את הרכיבים הנ"ל.
* התבונן בחיבורים אל ממשק השמע ונסה להבין כיצד המשתמש משנה את הצליל הנשמע.

הסבר כיצד המשתמש יכול לשנות את הצליל הנשמע.

**תשובה** : ניתן לשנות את הצליל שנשמע על ידי שינוי הכניסה freq , אשר מחובר ל Keypad[3..0]

כלומר נוכל להשפיע על התדר על ידי המקשים 0-9 A-F

* חבר את יצאת השמע של הכרטיס אל הרמקולים של המסך (כבל חיבור קיים על שולחן המעבדה שלך).

**שים לב**: השליטה בעוצמת השמע היא דרך שני הכפתורים השמאליים של המסך הימני



* הורד את היישום לכרטיס ובדוק שהוא עובד נכון. השמע צלילים שונים על ידי שינוי התדר.

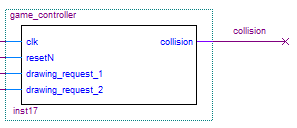
קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 10:30

## אינטגרציה ובקרת משחק

**משימה**: להוסיף מעין בקר משחק ליישום: כאשר הפרצופון מתנגש באובייקט (המלבן הנוסף שמעל הרקע) יש להביא את הפרצופון למקום ההתחלתי משם ימשיך את תנועתו כמקודם.

פתח קובץ SV חדש, וקרא לו למשל game\_controller.sv

* הכניסות שלו: שתי בקשות השרטוט של הפרצופון ושל המלבן הנוסף (עם המספרים)
* היציאה שלו: אות שיהיה 1 לוגי אם יש התגשות בין המלבן והפרצופון
* הוסף לוגיקה המזהה התנגשות בין הפרצופון והמלבן, כלומר שתי בקשות השרטוט מתקיימות באותו זמן באותו מקום. דוגמה לרכיב כזה:



דוגמה

* קמפל ועשה לו Symbol.

ב- smileyface\_move.sv

* הוסף למודול כניסה נוספת collision שמקבלת אות מיציאת ה- game\_controller
* הוסף לוגיקה המביאה את הסמיילי סינכרונית למקום שממנו הוא תמיד מתחיל את תנועתו ב- resetN, כתלות בכניסה זו.

ב- TOP של היישום

* הוסף את המודול החדש (הבקר) ליישום.
* חבר את הכניסות/יציאות החדשות לפי הצורך .
* קמפל, הורד לכרטיס ובדוק שהיישום עובד כמו שהתכוונת. אם לא, תקן בהתאם.

הוסף לדו"ח את השינויים ב- SV של המודולים שהשתנו

**module** smileyface\_move **(**

**input** **logic** clk**,**

**input** **logic** resetN**,**

**input** **logic** startOfFrame**,** // short pulse every start of frame 30Hz

**input** **logic** X\_direction**,** //change the direction in X

**input** **logic** toggleY**,** //toggle the y direction

**input** **logic** collision**,**

**output** **logic** **[**10**:**0**]** topLeftX**,**// output the top left corner

**output** **logic** **[**10**:**0**]** topLeftY

**);**

// a module used to generate a ball trajectory.

**parameter** **int** INITIAL\_X **=** 280**;**

**parameter** **int** INITIAL\_Y **=** 185**;**

**parameter** **int** INITIAL\_X\_SPEED **=** 30**;**

**parameter** **int** INITIAL\_Y\_SPEED **=** 20**;**

**parameter** **int** Y\_ACCEL **=** **-**1**;**

**const** **int** MULTIPLIER **=** 64**;**

// multiplier is used to work with integers in high resolution

// we devide at the end by multiplier which must be 2^n

**const** **int** x\_FRAME\_SIZE **=** 639 **\*** MULTIPLIER**;**

**const** **int** y\_FRAME\_SIZE **=** 479 **\*** MULTIPLIER**;**

**int** Xspeed**,** topLeftX\_tmp**;** // local parameters

**int** Yspeed**,** topLeftY\_tmp**;**

**logic** toggleY\_d**;**

// calculation x Axis speed

//======--------------------------------------------------------------------------------------------------------------=

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

**begin**

**if(!**resetN**)**

Xspeed **<=** INITIAL\_X\_SPEED**;**

**else** **begin**

**if** **((**topLeftX\_tmp **<=** 0 **)** **&&** **(**Xspeed **<** 0**)** **)** // hit left border while moving right

Xspeed **<=** **-**Xspeed **;**

**if** **(** **(**topLeftX\_tmp **>=** x\_FRAME\_SIZE**)** **&&** **(**Xspeed **>** 0 **))** // hit right border while moving left

Xspeed **<=** **-**Xspeed **;**

**end**

**end**

// calculation Y Axis speed using gravity

//======--------------------------------------------------------------------------------------------------------------=

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

**begin**

**if(!**resetN**)** **begin**

Yspeed **<=** INITIAL\_Y\_SPEED**;**

toggleY\_d **=** 1'b0 **;**

**end**

**else** **begin**

toggleY\_d **<=** toggleY **;** // for edge detect

**if** **((**toggleY **==** 1'b1 **)** **&&** **(**toggleY\_d**==** 1'b0**))** // detect toggle command rising edge from user

Yspeed **<=** **-**Yspeed **;**

**else** **begin** **;**

**if** **(**startOfFrame **==** 1'b1**)**

Yspeed **<=** Yspeed **-** Y\_ACCEL **;** // deAccelerate : slow the speed down every clock tick

**if** **((**topLeftY\_tmp **<=** 0 **)** **&&** **(**Yspeed **<** 0 **))** // hit top border heading up

Yspeed **<=** **-**Yspeed **;**

**if** **(** **(** topLeftY\_tmp **>=** y\_FRAME\_SIZE**)** **&&** **(**Yspeed **>** 0 **))** //hit bottom border heading down

Yspeed **<=** **-**Yspeed **;**

**end**

**end**

**end**

// position calculate

//======--------------------------------------------------------------------------------------------------------------=

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

**begin**

**if(!**resetN**)**

**begin**

topLeftX\_tmp **<=** INITIAL\_X **\*** MULTIPLIER**;**

topLeftY\_tmp **<=** INITIAL\_Y **\*** MULTIPLIER**;**

**end**

**else** **begin**

**if** **(**startOfFrame **==** 1'b1**)** **begin** // perform only 30 times per second

**if** **(**X\_direction**)** //select the direction

topLeftX\_tmp **<=** topLeftX\_tmp **+** Xspeed**;**

**else**

topLeftX\_tmp **<=** topLeftX\_tmp **-** Xspeed**;**

topLeftY\_tmp **<=** topLeftY\_tmp **+** Yspeed**;**

**end**

**if** **(**collision **==** 1'b1**)** **begin** // Checks collision

topLeftX\_tmp **<=** INITIAL\_X **\*** MULTIPLIER**;**

topLeftY\_tmp **<=** INITIAL\_Y **\*** MULTIPLIER**;**

**end**

**end**

**end**

//======--------------------------------------------------------------------------------------------------------------=

//get a better (64 times) resolution using integer

**assign** topLeftX **=** topLeftX\_tmp **/** MULTIPLIER **;** // note: it must be 2^n

**assign** topLeftY **=** topLeftY\_tmp **/** MULTIPLIER **;**

**endmodule**

**module** game\_controller **(**

**input** **logic** clk**,**

**input** **logic** resetN**,**

**input** **logic** drawing\_request\_1**,**

**input** **logic** drawing\_request\_2**,**

**output** **logic** collision

**);**

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

**begin**

**if(!**resetN**)** **begin**

collision **<=** 1'b0**;**

**end**

**else** **begin**

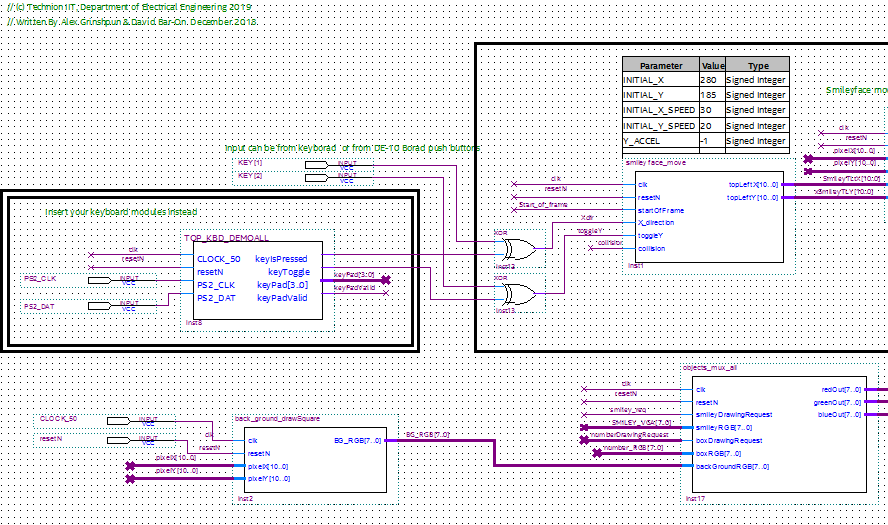
collision **<=** drawing\_request\_1 **&&** drawing\_request\_2**;**

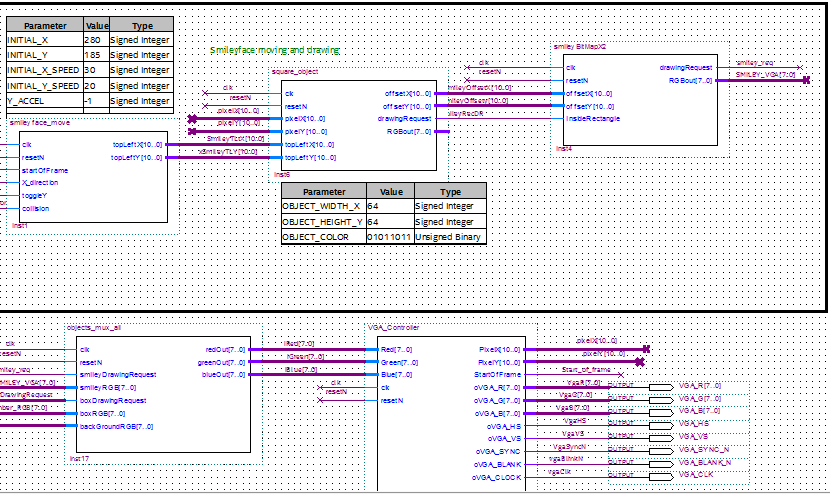
**end**

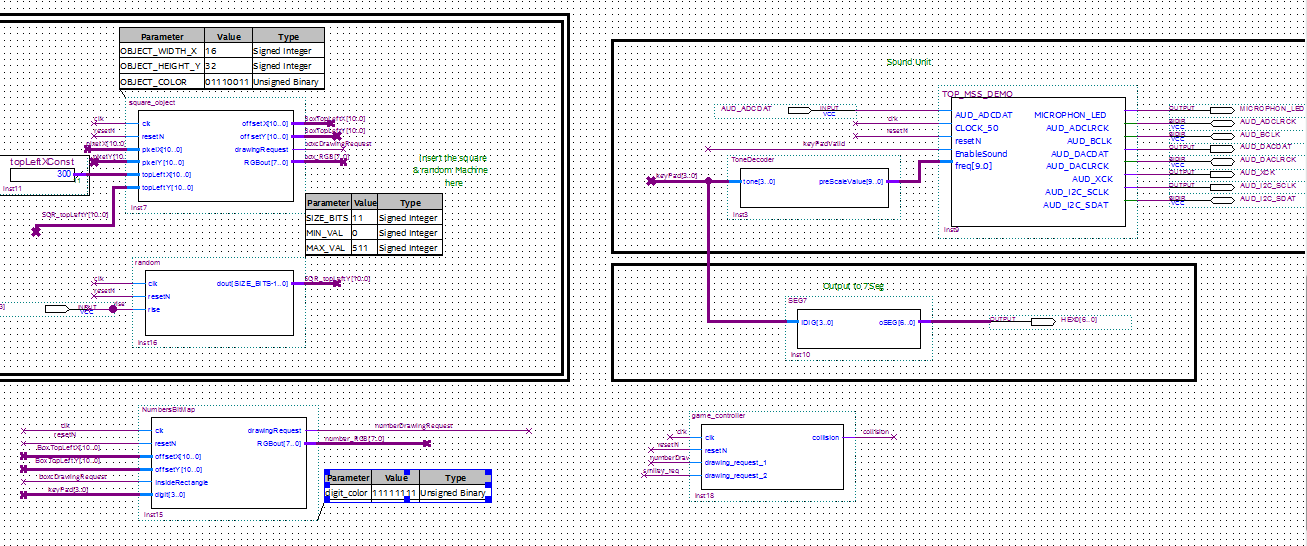
**end**

**endmodule**

הוסף לדו"ח את השרטוט הגרפי החדש עם המודולים הנוספים - (אפשר ביותר מתמונה אחת)







קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 11:02

# עבודה על הפרוייקט

* שב עם המדריך לדון על השקף המתאים מהמצגת עבור סכמת המלבנים הכללית של הפרויקט שלך – אותה תגיש במעבדה הבאה.

# גיבוי

* שמור את הפרוייקט כארכיבQAR והעלה למודל – הוא ישמש כבסיס לפרוייקט שלך
* שמור דוח זה כWORD והעלה למודל - אין עליו ציון – הוא ישמש אותך בלבד

רשום את השעה בה סיימת את המעבדה: 11:02