



הפקולטה להנדסת חשמל
ע"ע אנדרו וארנה ויטרבי



הטכניון
מכון טכנולוגי לישראל

מעבדה בהנדסת חשמל
044157 א'1

ניסוי VGA - ממשק מסך, מקלדת וצלילים

תדריך מעבדה ודוח סיכום

עם ממשק מקלדת נתון

גרסה 1.6
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

1	מבוא	2
2	הכרה של פלטפורמת ה-VGA	3
2.1	חיבור המערכת	3
2.2	הפעלת יישום ה-VGA	3
3	שינוי BITMAP - הפיכת ה-SMILEY	5
4	חיבור מכלולים נוספים	6
4.1	הוספת מלבן מעל הרקע הסטטי ומכונת RANDOM	6
4.2	תרגול שימוש בנתח הלוגי, ה-SIGNAL_TAP	9
4.3	הוספת ספרות ליישום	10
4.4	אינטגרציה ובקרת משחק	11
5	עבודה על הפרויקט إستفتاح - סיפתח	13
5.1	מטרות הספתח	13
5.2	תיאור הספתח	13
5.3	דיון עם המדריך	14
6	הרחבת רשות: הוספת צלילים	15
7	גיבוי העבודה	16

1 מבוא

מטרות מעבדה זאת הן:

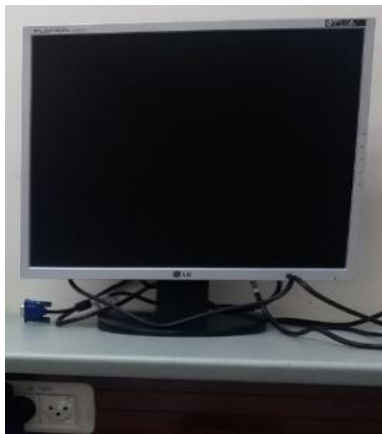
- הכרה של ממשק ה-VGA דרך יישום נתון
- הרחבת היישום על ידי שינויים במודולים שונים
- עבודה על הפרויקט הספציפי – ביצוע סיפתח
- נושא אפשרי להרחבה – שמוש בצלילים

שימו לב: בדוח הסופי של הפרויקט אתם נדרשים להוסיף חלקים שבצעתם במעבדה זו. השלימו בהתאם להנחיות שם. בסוף המעבדה הזאת יש להגיש מסמך זה (לא את הדוח הסופי של הפרויקט) כמו תמיד כ-PDF.

2 הכרה של פלטפורמת ה- VGA

2.1 חיבור המערכת

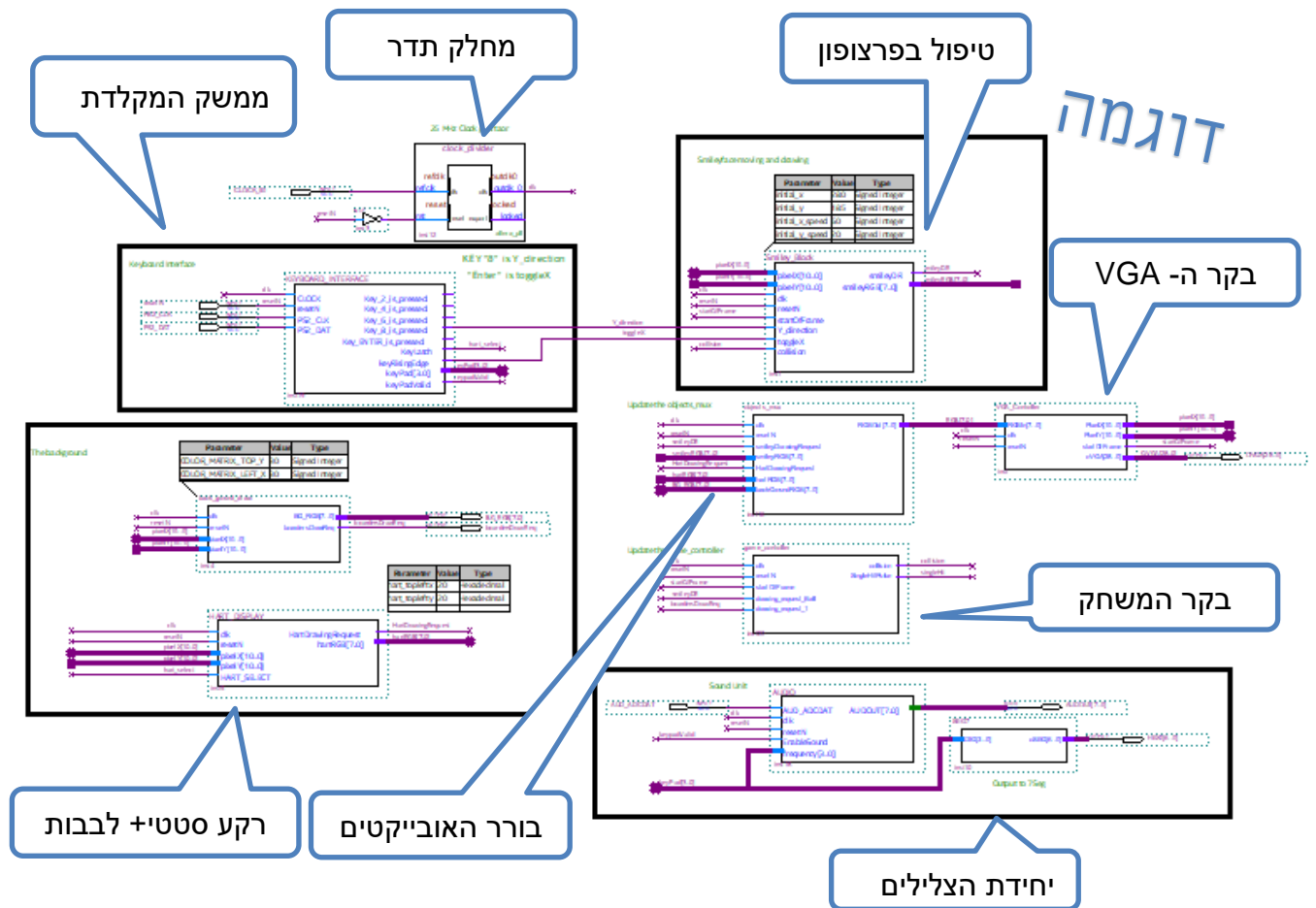
- הגדר תיקייה למעבדת VGA. הורד מהמודל את קובץ הארכיב של המעבדה ופתח אותו בתיקיה שיצרת. קובץ זה ישמש גם כבסיס לפרוייקט הסיום שלך.
- הפעל את כרטיס הניסוי. בדוק שהכרטיס והמסך הנוסף שנמצא בכל עמדת עבודה על המדף (מעל שני המסכים של העמדה) נדלקים.
- כמו כן, גם מקלדת מספרים מחוברת לכרטיס, וודא שחיבור זה אכן קיים.



2.2 הפעלת יישום ה- VGA

- בפרוייקט שנפתח פתח את ה- TOP: TOP_VGA_DEMO_KBD.bdf.
- ה- TOP מחולק לאזורים (המפורטים באיור הבא) ובכל שלב במעבדה זו נתמקד באזור אחר.
- הממשק למקלדת** נתון לך כקופסה סגורה במעבדה זו KEYBOARD_INTERFACE.
- הכניסות שלה נתונות ואין לשנותן.**
- היציאות**, חלקן מחוברות ומשמשות לתפעול היישום וחלקן אינן מחוברות אבל אפשר להשתמש בהן לפי הצורך. **יציאות הממשק למקלדת המספרים הן:**
- Key_2_is_pressed – נותנת 1 לוגי כל עוד מקש 2 (חץ למטה) לחוץ
 - Key_4_is_pressed – כנ"ל עבור מקש 4 (חץ שמאלה) לחוץ
 - Key_6_is_pressed – כנ"ל עבור מקש 6 (חץ ימינה) לחוץ
 - Key_8_is_pressed – כנ"ל עבור מקש 8 (חץ למעלה) לחוץ
 - Key_Enter_is_pressed – נותן 1 לוגי כל עוד מקש ENTER לחוץ
 - KeyLatch – יש פעולת TOGGLE בכל לחיצה על המקש המוגדר שהוא ENTER
 - keyRisingEdge – נותנת פולס של שעון אחד כל לחיצת המקש המוגדר - שהוא ENTER
 - keyPad[3..0] – נותנת את המספר של המקש הלחוץ, אם הוא Valid (מספר)
 - keyPadValid – מוציאה 1 לוגי כל עוד אחד המקשים החוקיים (אחד המספרים) לחוץ

דוגמה



תחילה נבדוק שיישום ה-Smiley בכללותו עובד נכון.

- **בצע סינתזה ל-TOP**, הרץ קובץ הדקים (tcl) ואז הרץ קומפילציה מלאה.
- **צרו** את הפרוייקט לכרטיס, ודא שיישום ה-Smiley עובד נכון ואשר זאת עם המדריך.
- **התבונן** בישום שמופעל על המסך:
- ברקע שמאל למעלה יש מטריצה של 16X16 ריבועים (כל ריבוע מכיל 8X8 פיקסלים) בה מציגים את 256 הצבעים האפשריים (3- ביט אדום, 3 ביט ירוק, 2 ביט כחול)
- במרכז הרקע יש ביטמאפ של לב שמורכב מלבבות קטנים. ניתן לבחור את סוג הלב הקטן שירכיב את הלב הגדול.
- ישנו רכיב מחלק תדר – שמוציא תדר של 25MHz המתאים לעבודה עם מערכת ה-VGA, כך ששעון זה נכנס לכל רכיבי הפרויקט.

זהה את הסיגנל שבוחר את סוג הלב הקטן והבן את פעולתו. מה הסיגנל ואיזה מקש מפעיל אותו?

תשובה:

TOGGLEX - מופעל על ידי מקש "ENTER" אשר משנה את סוג הלבבות

תתבקש להסביר על הפעולה של בחירת הלב למדריך.

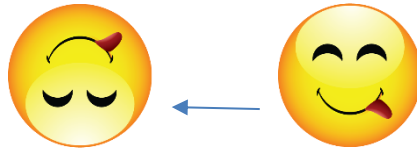
קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 8:52

3 שינוי BITMAP - הפיכת ה-SMILEY

משימה: לשנות את קובץ ה-BITMAP. ספציפית יש לסובב את הביטמאפ כתמונת ראי:

אם תרצה לא לדרוס את הקובץ המקורי שמור אותו בשם אחר ומכאן והלאה תעבוד איתו. תוסיף/תעדכן מודולים כפי שיתואר בהמשך.

- פתח את מודול הביטמאפ smileyBitMap.sv שבתוך מודול Smiley_Block והבן את אופן פעולתו.
- שנה את הקוד כך שיציג את הסמיילי הפוך כמו בתמונה.



הנחיה: בקובץ הביטמאפ שים לב באיזה מהצירים יש לעשות שינוי כך שמה שהיה למעלה עכשיו יוצג למטה.

הוסף לדו"ח את השינויים שביצעת בקוד ה-SV של מודול הפרצופון המסובב.

```
logic [OBJECT_HEIGHT_Y-1:0] [0:OBJECT_WIDTH_X-1] [7:0] object_colors = {  
8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF  
8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF, 8'hFF
```

- קמפל, צרוב לכרטיס ובדוק שהיישום עובד נכון אחרי השינוי.

שים לב! מכאן והלאה המשך לעבוד עם הפרצופון המסובב.

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 9:04

4 חיבור מכלולים נוספים

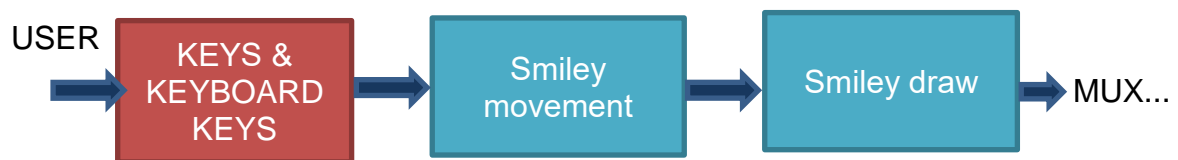
- שים לב ששני מקשים הוגדרו כמשפיעים על תנועת הפרצופון.
- בדוק אותם.

פרט מה תפקידם.

תשובה:

TOGGLEX - מופעל על ידי מקש "ENTER" אשר משנה את סוג הלבבות ומחליף כיוון בציר הא

Ydirection - מופעל על ידי מקש "8" - מקדם את הסמיילי בכיוון ציר ה



4.1 הוספת מלבן מעל הרקע הסטטי ומכונת RANDOM

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

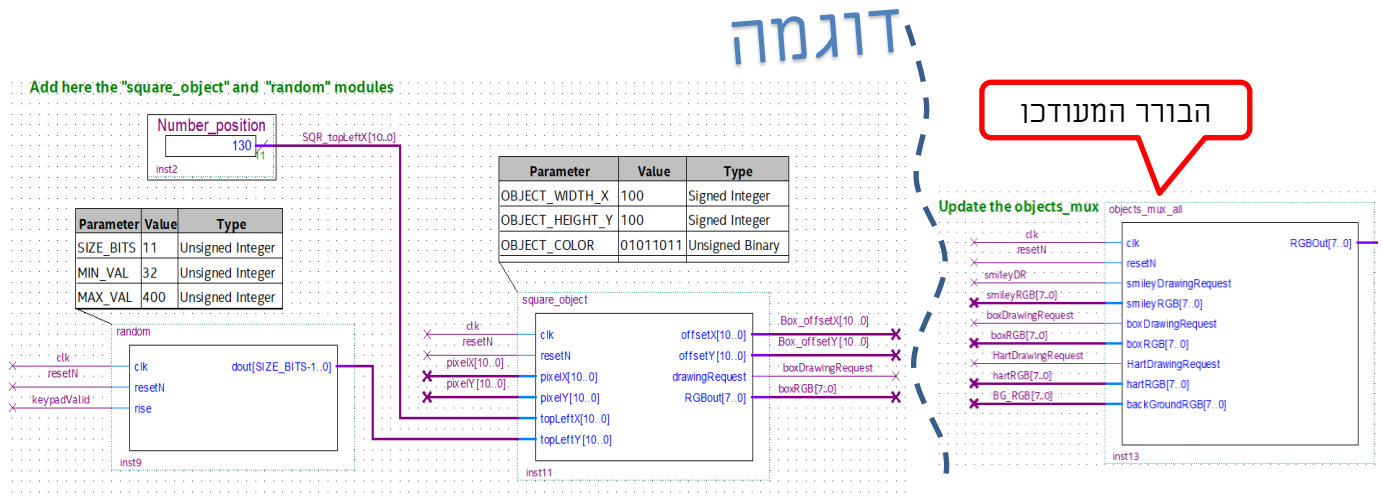
- הוסף ל- TOP של הפרויקט שלך, רכיב (instance) נוסף מסוג square_object (הרכיב כבר קיים בפרויקט אך יש ליצור Symbol שלו). ראה שרטוט להלן.
- ניתן לקבוע את גודלו ולבחור את צבעו של המלבן כרצונך, באמצעות פרמטרי הרכיב.
- יש למקם את המלבן החדש במיקום אופקי קרוב למסלול תנועתו של הפרצופון, כך שהוא והמלבן יגעו אחד בשני לפחות לרגע קט במהלך תנועת הפרצופון.
- קבע את הקואורדינטה **topLeftX** על ידי מודול שמוציא מספר קבוע.
 - לשם כך תשתמש ברכיב מסוג LPM_CONSTANT הקיים בשרטוט.
 - (בעתיד, להוספת עוד רכיב מסוג זה ניתן להעזר ב - COOK BOOK).
- הקואורדינטה **topLeftY** תיקבע על ידי מודול שמוציא מספר אקראי.
 - לשם כך חבר מכונת RANDOM (קיימת בפרויקט, צור Symbol שלה) אשר תופעל בלחיצת כל לחצן מקלדת (זהה את הסיגנל שמדווח על כך).
 - שים לב שיש להתאים את הפרמטרים שלו:
 - את גודל וקטור המוצא של מכונת RANDOM לגודל וקטור הכניסה של מודול המלבן (על יד קביעת פרמטר SIZE_BITS של הרכיב). ציין: 11
 - ואת טווח התנועה שרוצים לאפשר בכיוון Y, בהתאם לגודל המסך או אחרת. ציין מה קבעת: ערך תחתון 0 , ערך עליון 379 .
- כמו כן עליך להוסיף עוד זוג כניסות ולוגיקה מתאימה לבורר העדיפויות (mux) לטיפול במלבן. בלוגיקה קבע עדיפותו מעל הרקע ומתחת לפרצופון.

פתח קובץ נתון בפרויקט שלך objects_mux, שמור אותו (אפשר לשמור בשם אחר כדי לא לדרוס אותו) והשלם בו את הכניסות והלוגיקה הדרושות עבור האובייקט הנוסף (המלבן).

צור Symbol למודול זה.

בהירארכיה העליונה החלף את הבורר הישן בחדש וחבר אליו את המלבן.

אחרי שינויים אלה אזור זה ביישום שלך צריך להיראות כך:



קמפל, צרוב לכרטיס ובדוק שהיישום עובד נכון אחרי השינוי.

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

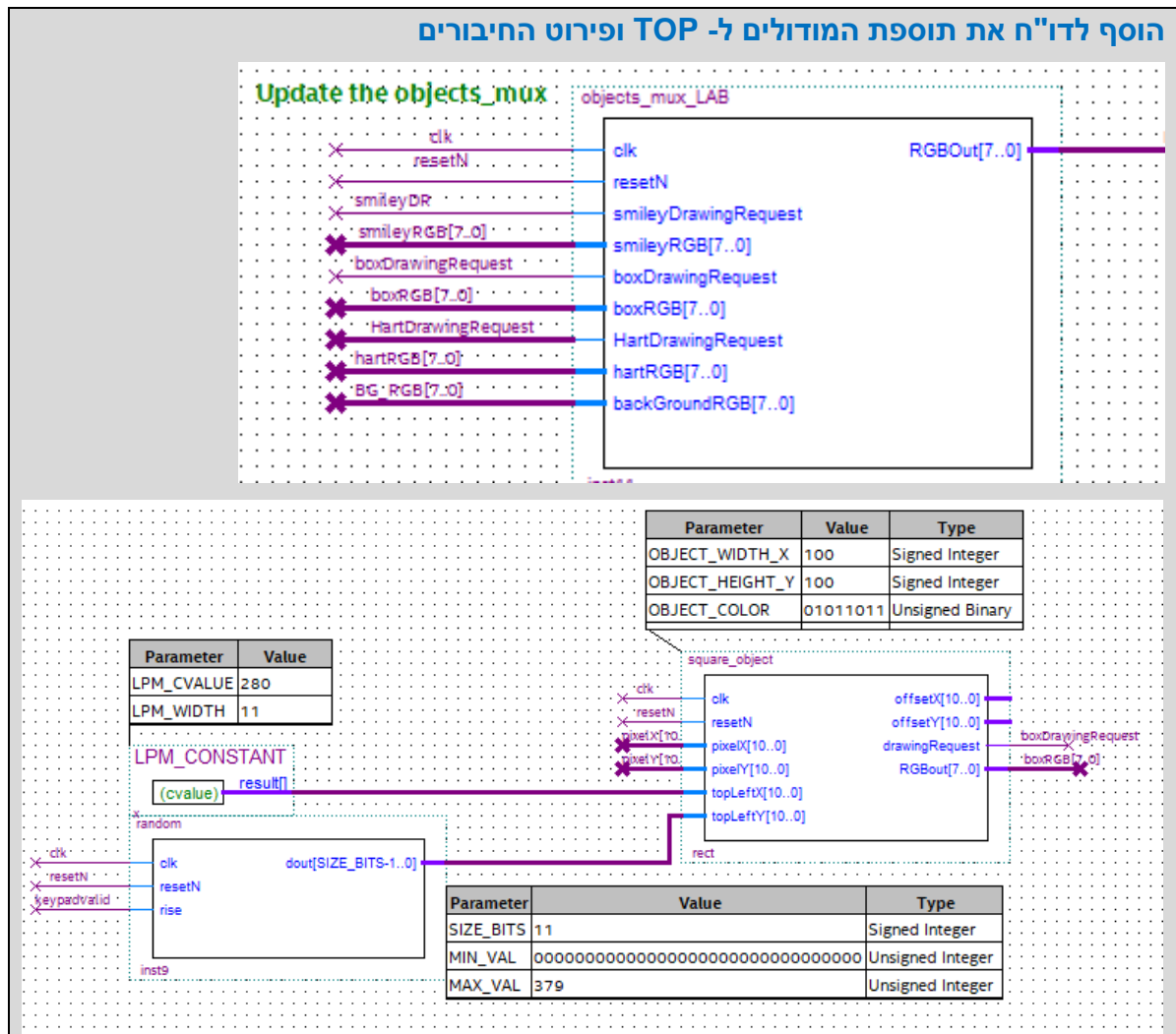
```
// add the box here
input logic boxDrawingRequest, // two set of inputs per unit
input logic [7:0] boxRGB,

always_ff@(posedge clk or negedge resetN)
begin
    if(!resetN) begin
        RGBout <= 8'b0;
    end
    else begin
        if (smileyDrawingRequest == 1'b1)
            RGBout <= smileyRGB; //first priority

        // add logic for box here
        else if (boxDrawingRequest == 1'b1)
            RGBout <= boxRGB; //second priority

        else if (HartDrawingRequest == 1'b1)
            RGBout <= hartRGB;
        else
            RGBout <= backGroundRGB ; // last priority
        end ;
    end
endmodule
```

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



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

4.2 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 רלוונטיים במקרה זה.

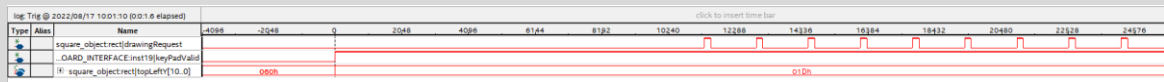
כיצד תקבע את תנאי ה- Trigger במקרה זה?

תשובה:

עליית "rise" אשר נשלט על ידי לחיצה על אחד המקשים – כלומר `keyPadValid=1`

- המתן להתרחשות האירוע שמשפיע על `rise`, מהו? וראה כיצד זה משפיע על מיקום המלבן בזמן שהמערכת פעילה.

הוסף לדו"ח רכישה מ- *SIGNAL TAP*



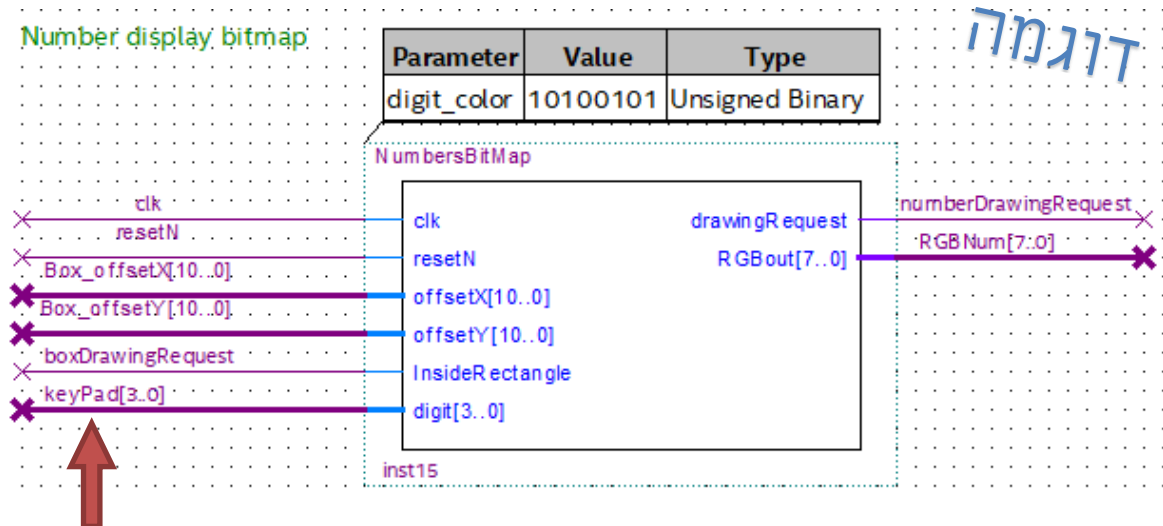
קרא למדריך, רשום את השעה בה הוא ראה את המעגל:

4.3 הוספת ספרות ליישום

מטרה: להוסיף ספרות כאובייקטים גרפיים מעל הרקע.

נתון לך מודול שמכיל ביטמאפ של ספרות ואותיות 0 עד 9, `NumbersBitMap`.

- פתח אותו והבן את פעולתו.



שים לב: ממשיק הכניסה מחובר למקלדת שתתן את הספרה אותה רוצים להציג

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.

- **השתמש** במלבן מסעיף קודם (הוספת אובייקט מלבן מעל הרקע הסטטי):
 - **קבע** את גודלו של המלבן ל- $x = 16, y = 32$, כדי להתאים לגודל הביטמאפ של האותיות והספרות שנרצה להציג
 - בקשת השרטוט של המלבן נכנסת לכניסת הביטמאפ של המספרים (**חיבור זה כבר קיים באמצעות שמות החוטים**)
- **חבר** את ה- `drawing request` של ביטמאפ המספרים לכניסה של ה- `object_mux` במקום הבקשה לשרטוט של המלבן (אפשר גם באמצעות שמות חוטים).
- **חבר** גם את צבע המספרים `RGBNum[7..0]` ל- `object_mux` במקום צבע המלבן (אופציונלי, אם רוצים צבע שונה מהמלבן).
- **קמפל**, **צרוב** לכרטיס **ובדוק** שהיישום עובד נכון אחרי השינוי.
- **בדוק** שהספרות מוצגות נכון על המסך.
- **בדוק** את פעולת המקשים 0-9

4.4 אינטגרציה ובקרת משחק

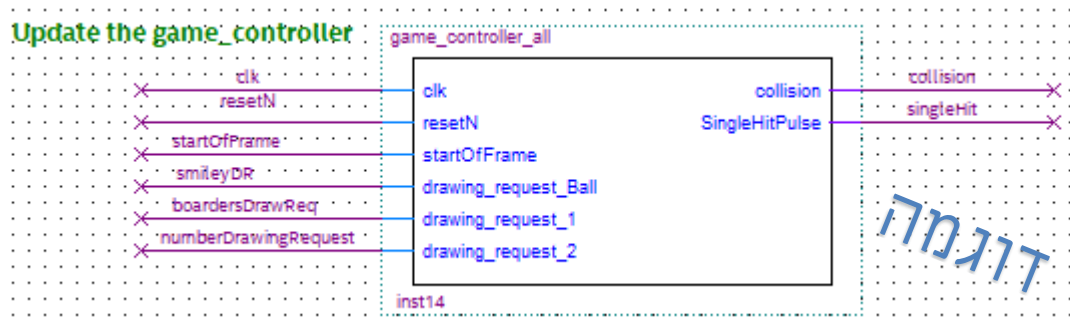
משימה: להוסיף תכונות לבקר המשחק: כאשר הפרצופון מתנגש באובייקט (המספר שמוצג מעל הרקע) הפרצופון ישנה כיוון תנועה וימשיך את תנועתו. כמו כן, נרצה כי בכל התנגשות על המסך בין ה- smiley למספר: מיקומו של המספר על המסך יוגרל מחדש.

- לשם כך יש לחבר את יציאת ה- SingleHitPulse של הבקר לכניסה rise במקום סיגנל keypadValid כך שבכל התנגשות בין הפרצופון למספר מכונת ה-RANDOM תגריל מחדש את מיקומו של המספר.

בבקר המשחק

התאם את הרכיב **game_controller.v** הקיים לדרישות הבאות (אפשר לשמור אותו בשם **game_controller_all.v**, כדי לא לדרוס את הקובץ הקיים):

- זיהוי התנגשות כלשהי כך ש- collision יתקיים אם יהיה drawing request של הסמיילי ושל השוליים או של מלבן המספר.
- collision_smiley_number יתקיים אם יהיה drawing request של הסמיילי רק עם מלבן המספר.
- **הכניסות שלו:** בקשות השרטוט של כל האובייקטים: הפרצופון, המלבן הנוסף (עם המספרים) והשוליים.
- **היציאות שלו:**
 - **collision** - שיהיה 1 לוגי אם יש התנגשות בין אובייקטים לפי הכללים הנ"ל (collision_smiley_number לא צריך להיות כיצאה).
 - **SingleHitPulse** - המוציא 1 למשך מחזור שעון יחיד כאשר יש התנגשות בין הפרצופון ומלבן המספרים.



- **קמפל וצור Symbol** עבורו.

ב- TOP של היישום

- **הוסף** את המודול המעודכן **game_controller** 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 game_controller_lab (
    input logic clk,
    input logic resetN,
    input logic startOfFrame, // short pulse every start of frame 30Hz
    input logic drawing_request_Ball,
    input logic drawing_request_1,
    // add input from box of numbers here
    input logic drawing_request_digit,

    output logic collision, // active in case of collision between two objects
    output logic singleHitPulse // critical code, generating A single pulse in a frame
);

// drawing_request_Ball --> smiley
// drawing_request_1 --> brackets
// drawing_request_2 --> number/box

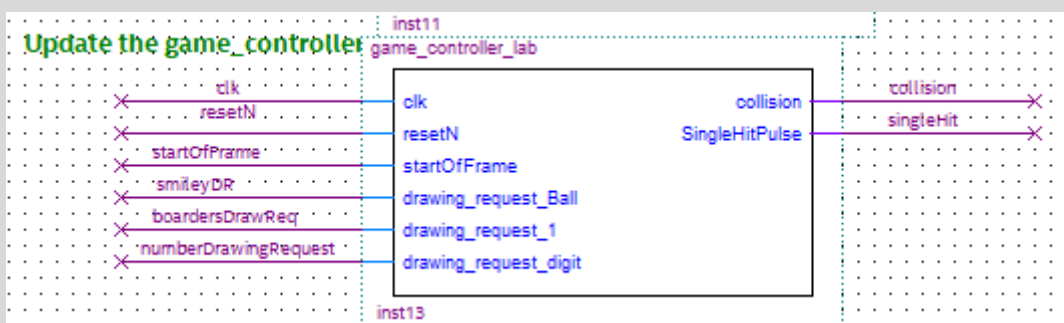
logic collision_smiley_digit;
assign collision_smiley_digit = ( drawing_request_Ball && drawing_request_digit );
assign collision = ( drawing_request_Ball && drawing_request_1 ) || collision_smiley_digit ;// any collision

flag <= 1'b0 ; // reset for next time |
// change the section below to collision between number and smiley

if ( collision_smiley_digit && (flag == 1'b0)) begin
    flag <= 1'b1; // to enter only once
    singleHitPulse <= 1'b1 ;
end ;

```

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



Number display bitmap

Parameter	Value	Type
digit_color	00	Hexadecimal



קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 11:15

5 עבודה על הפרוייקט إستفتاح -סיפתח

- שב עם המדריך לדון על השקף המתאים מהמצגת עבור סכמת המלבנים הכללית של הפרויקט שלך.
- **שמור** את קובץ ה TOP בשם חדש (צור עותק) עבור הפרויקט.
- **שנה** את הקוד כך שיתאים לתנועת אחד האובייקטים בפרויקט שלך.
- **למשל** - נוון את תנועת הפרצופון כך:
 - בציר X - ינוע משמאל לימין, בין המסגרות הפנימיות
 - בציר Y – יהיה במקום קבוע, בחצי העליון של המסך
 - כל פעם שהוא יתנגש במסגרת הימנית, ישנה את מיקומו למסגרת השמאלית ושוב ינוע באותה צורה
 - עצור/המשך את תנועת הפרצופון בעזרת אחד הלחצנים או המתגים
- **מלא** את הסעיפים הבאים ולאחר מכן העתק אותם, לפי הצורך, לדוח המסכם של הפרויקט (במודל)

5.1 מטרות הספתח

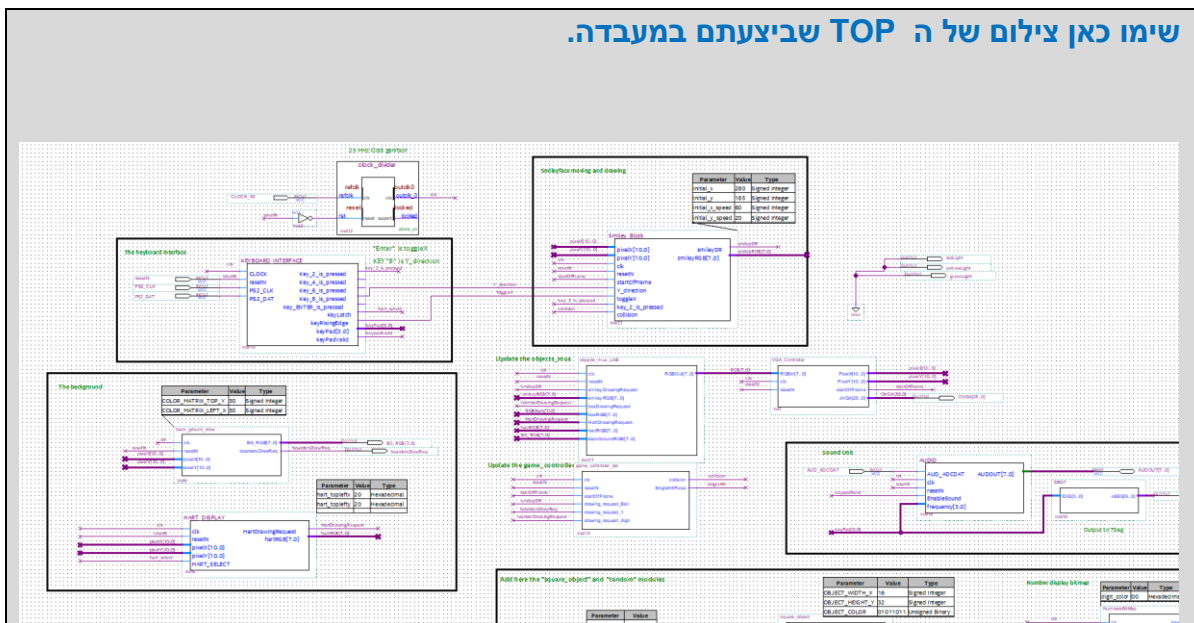
רשמו כאן מה אתם מצפים להשיג מהספתח.

תשובה:

לגרום לתנועת הסמיילי בלחיצה על כפתור ספציפי, אחרת לא נע.

5.2 תיאור הספת

שימו כאן צילום של ה TOP שביצעתם במעבדה.



5.3 דיון עם המדריך

רשמו כאן את עיקרי הדברים, ודגשים חשובים להמשך העבודה, מה אתם הולכים לעשות עד המעבדה הבאה.

תשובה:

להחליף את הסמיילי לכדור לבן,

להוסיף כדור נוסף

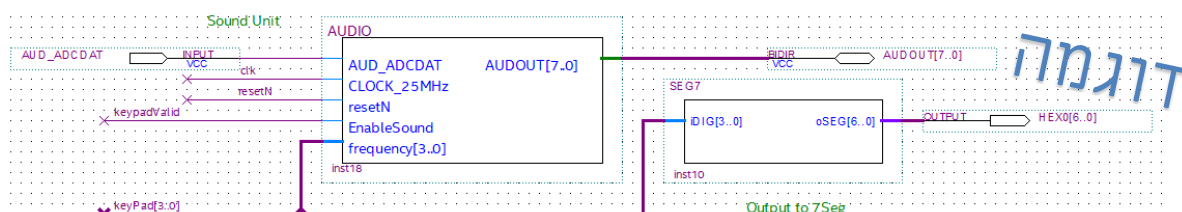
כאשר קורית התנגשות אז להוסיף את הכיוון הנכון למהירויות כתוצאה מההתנגשות

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 12:43

6 הרחבת רשות: הוספת צלילים

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

בשלב זה נתמקד בממשק השמע, המודול AUDIO. פירוט על אופן פעולתו מופיע בחומר הרקע למעבדה זו. בקצרה, יישום זה משתמש 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.



- **התבונן** במודולים של הקובץ AUDIO וזהה את הרכיבים שלו.
- **התבונן** בחיבורים אל ממשק השמע והבן כיצד המשתמש משנה את הצליל הנשמע וכיצד הוא מפעיל/מפסיק את הצליל.

הסבר:

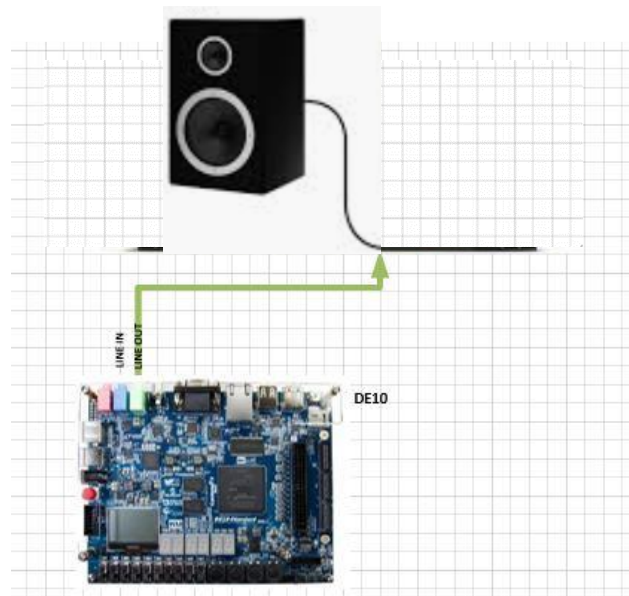
- כיצד המשתמש יכול לשנות את הצליל הנשמע?

תשובה:

- כיצד המשתמש יכול להפעיל/להפסיק את הצליל?

תשובה:

במעבדה ישנם רמקולים מחוברים לכרטיס. השליטה בעוצמת השמע היא דרך כפתור בחלק האחורי של אחד הרמקולים.
בבית אפשר לחבר אוזניות או רמקולים חיצוניים ליציאת השמע (הירוקה) של הכרטיס (כמוסבר באיור הבא).



- הורד את היישום לכרטיס ובדוק שהוא עובד נכון. השמע צלילים שונים.

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

7 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 השמירה, למשל VGA_Siftah_110422.
 העלה את קובץ הארכיב למודל. קובץ זה ישמש לקובץ הבסיס עבור הפרויקט שלך.
 תפתח את הפרויקט שלך כהמשך לפרויקט ה- VGA.

בזמן פיתוח הפרויקט שמור שלבים שונים שעובדים כקבצי ארכיב עם שמות הכוללים פרטים ברורים או תאריכים.

גבה את הדוח והפרויקט גם באמצעים אחרים.

רשום את השעה בה סיימת את המעבדה: 12:45