



**הטכניון**

מכון טכנולוגי

לישראל

הפקולטה להנדסת חשמל ע"ש אנדרו ורנה ויטרבי

מעבדה בהנדסת חשמל  
044157 1א'



**פרויקט סיום  
תבנית דוח מסכם**

**גרסה 2.1**

**קיז 2020**

**מחברים : אברהם קפלן, דודו בר-און, נעם ליבוביץ עציון**

סטודנט	שם פרטי	שם משפחה
1	נדב	أشكוני
2	איתית	סלפיאן

Bumpy	שם הפרויקט
אייל לב	שם המדריך הקבוע

## תוכן עניינים – פרויקט

3	נספח מנהלי	1
3	תכנון זמינים	1.1
3	סיכום פגישות	1.2
4	הקדמה	2
4	צילום של הפרויקט	2.1
4	הנחיות כללוות	2.2
5	סקר ספרות	2.3
5	אפיון הפרויקט	3
5	הדרישות המקוריות של הפרויקט (כמו במצבת)	3.1
5	החלק היצורתי	3.2
6	ארכיטקטורה - ממשקים לעולם החיצון	3.3
7	IMPLEMENTATION – מימוש הספתקה ודיון עם המדריך	4
7	מטרות הספתקה	4.1
7	תיאור הספתקה	4.2
8	דיון ומסקנות עם המדריך	4.3
<b>Error! Bookmark not defined.</b>		4.3.1
9	עדכון טבלת התוכנו שבסעיף 1	5
9	ארכיטקטורה וסכמה מבנים פנימית	5.1
10	סקמות מבנים	5.2
10	רשימת חמשת המכלולים העיקריים, תפקידם וסדר ביצועם	5.3
10	פרוטוגדרת שני מודולים העיקריים למאגרים במצבת	5.3.1
11	מודול ראשון [החלף בשם המודול]	5.3.2
11	מודול שני [החלף בשם המודול]	5.3.3
12	שלבים במימוש הפרויקט - MVP - minimal viable project	6
12	מימוש ה- MVP (כמו במצבת)	6.1
13	תיאור מפורט של שני מודולים (כמו במצבת) – שם המודול – שם הסטודנט האחראי	7
14	מודול ראשון – [שם המודול] – שם הסטודנט האחראי	7.1
14	דיאגרמות מבנים	7.1.1
14	דיאגרמת מצבים ( bubble diagram – בועות )	7.1.2
14	מסך(י) סימולציה של המודול	7.1.3
16	מודול שני – [שם המודול] – שם הסטודנט האחראי	7.2
16	דיאגרמות מבנים	7.2.1
17	דיאגרמת מצבים ( bubble diagram – בועות )	7.2.2
18	מסך(י) סימולציה	7.2.3
20	(S.T.) Signal Tap	8
22	דיוון ומסקנות עם המדריך במהלך מעבדת אינטגרציה	8.1
<b>Error! Bookmark not defined.</b>		8.1.1
23	עדכון טבלת התוכנו - התוכנות לסיום הפרויקט	9
23	IMPLEMENTATION – התוכנות לסיום הפרויקט	9.1
27	צריכת משאבים	9.2
28	סיכום ומסקנות	10
28	המלצות לשנה הבאה (אם יש)	11
28	נספחים : דפי נתונים, דפי מידע שונים בהם השתמשת.	12

## 1 נספח מנהלי

### 1.1 תכנון זמנים

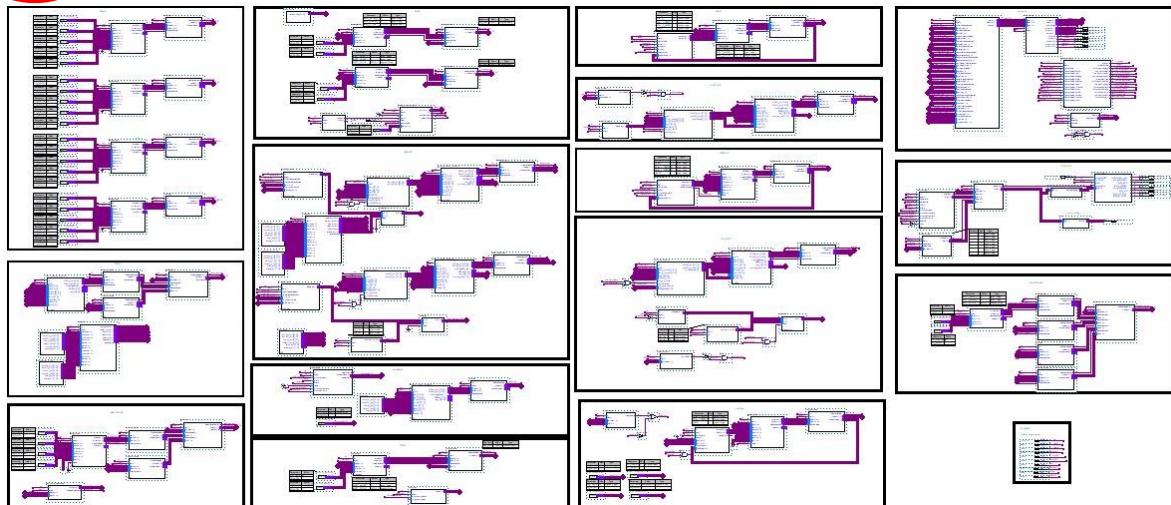
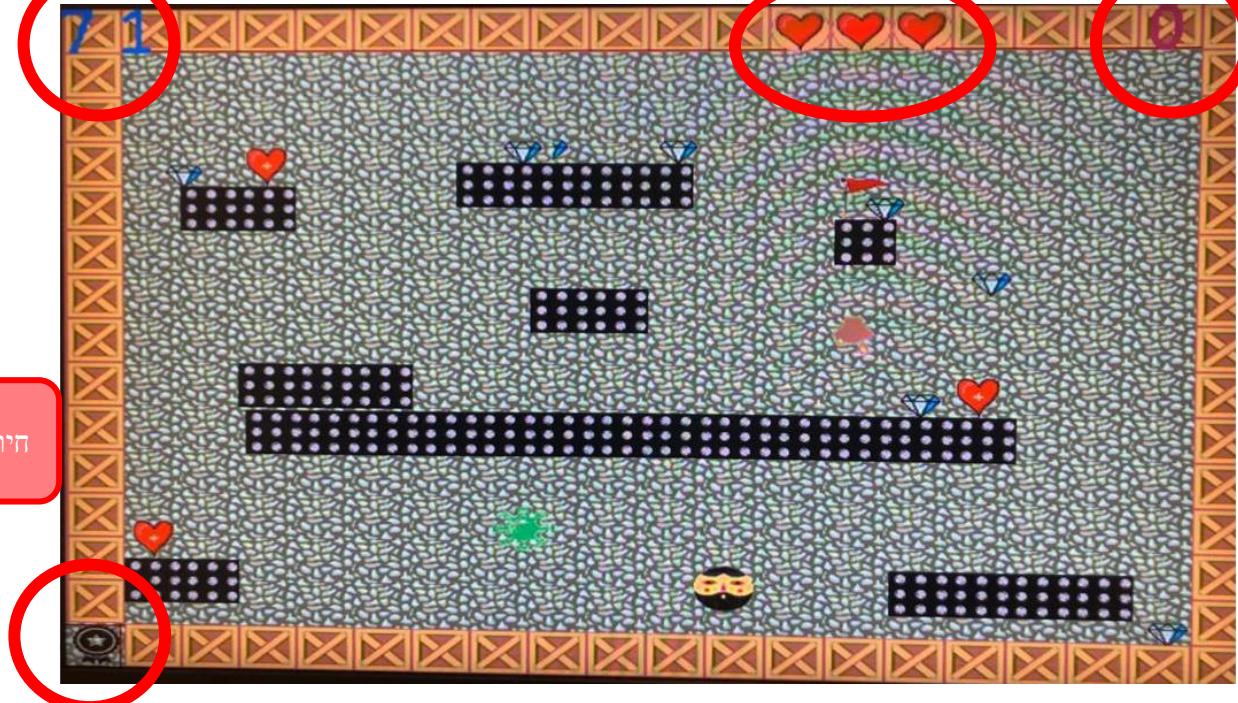
תיאור	מתקנים	תאריך בפועל	תאריך יכובן	הערות ומסקנות
דיוון בהגדרת הפרויקט	25/8	25/8		
מיימוש ספתח	25/8	25/8		
סקמת מלבנים MVP	29/8	29/8		במהלך המימוש אנו מעריכים שני אלץ להתמודד עם שינויים בסיסיים design לפי הדרישות שיעלו, ולפיכך אנו נוטים לתת הערכת יתר.
כתיבת מכונות המכבים של הפרויקט	בוצע בכל משך הפרויקט	29/8		בדומה להערה לעיל, מכונות מכבים, להערכתו ייכבו בתאריך המצוין, אך יתווסף מכונות נוספות בהתאם לדרישות והאלוצים של הפרויקט.
מיימוש MVP	31/8	30/8		עיקר הפרויקט מביחסת משך הזמן הנדרש
CODE REVIEW	12/9	11/9		לאחר שטירות הפרויקט הושגו, והתקבל משחק תקין ע"פ לוגיקה מוגדרת, היינו מסוגלים לבחון באופן מדויק יותר את האינטגרציה בין כל הרכיבים שבנינו, הודות להבנה המعمיקה שלנו במבנה.
דיונים עם מדריך על בעיות	10/9 , 3/9	3/9		התיעזויות בוצעו במהלך שעות הקבלה של המדריך, כמו גם בשעות קבלה של שאר חברי הסגל ובהתכתבויות במיל.

### 1.2 סיומים פגישות

תיאור	נושא לשיחה	bumbedat	שם המדריך	תאריך בפועל	צפי לתוצאות	הערות ומסקנות
דיוון בהגדרת הפרויקט	מספרת הפרויקט	VGA	אייל לב	25/9	תוכנית עבודה לשבעה הראשון לפרויקט	התחייבנו לבנות עד 31/8 : Bumpyת תנעה מוגדרת לעט Bitmpar לרקע המשחק Bumpy Bitmap לדמויות Bitmpar שורת מדינות תקינה
דיוון בארכיטקטורה	סקמת מלבנים בעפוץ	VGA	אייל לב	31/8	משמעותם של המרכיבים	הוחלט לשנות מעט את לוגיקת התנטשות Bumpy עם גבולות המסך
דיוון ב- NVP	TOP	VGA	אייל לב	31/8	משמעותם של המרכיבים	הוחלט לגבי ההפניות שנבנה במהלך הפרויקט : מערך חיים מערך נקודות Bumpy מערכ אובי שפוגע בעפוץ שלבי משחק שימושיים ע"פ מערכים אלו
דיונים על בעיות	באגים שיצרו במהלך הבדיקה	כאשר ניתקל בחן	דוד בר- אוון	3/9	פתרון בעיות	Timing Bugs, Generate inside 'for' loop,
CODE REVIEW ראשוני	TOP מכלולים	איתנגרציה	אייל לב	14/9	משמעותם של המרכיבים	
CODE REVIEW מצגת	כל הפרויקט	בחינה	סgal הקורס	14/9	המועד הסופי	

## 2 הקדמה

### 2.1 צילום של הפרויקט



### 2.2 הנחיות כלליות

- מטרת הדוח לתקן ולתעד ביצורה מלאה את פרויקט הסיום שבוצעו.
- יש לכתוב ביצורה מלאה וברורה, כך שננתן יהיה להבין את הפרויקט על סמך קריית הדוח.
- יש לוודא שכל הشرطויות, הנסיבות, האירורים, הגрафים, התמונות וכיוצא ברורים ומובנים. שרטוט מ-Print-Screen ע"י: סימון הشرطויות, העתק, הדבק, ולא
- בכל אחד מפרקיה הדוח, יש לציין את החלק השיך לתוספת היצירתית.

## 2.3 סקיי ספרות

Bumpy Tutorial: <https://www.youtube.com/watch?v=Pf23jQ9aDro>

Online game simulator:

<https://playclassic.games/games/puzzle-solving-dos-games-online/play-bumpys-arcade-fantasy-online/play/>

## 3 אפיון הפרויקט

### 3.1 הדרישות המקוריות של הפרויקט (כמו במצבה)

- שורה של מדרגות בתתית המסך
- תזוזה של באMPI עם הלחצנים ימינה ושמאליה בין המדרגות, כל לחיצה תזיז במדרגה אחת.
- קפיצה לשורה שנייה
- שורה אחת של ארבע מדרגות מעל השורה התחתונה עם הפתעה אחת
- איסוף של הפתעה יגרום להופעת מהילה באחת מהמדרגות התחתונות
- מפגש של באMPI עם המילה יוביל לניצחון
- מספר שורות של מדרגות עם הפתעות
- לא כל המדרגות קיימות – מוגרל רנדומלי
- קפיצה ונפילה חופשית לתזוזה בין שורות המדרגות באמצעות הלחצנים
- מונה זמן מטה (אם כל הפתעות לא נאספו בתוך זמן שקבע מראש אז יש פסילה).

לאחר התייעצות שלנו עם אייל הגענו למסקנה שrndomilitot של המדרגות תפגום בחווית המשתמש, ועל כן הוחלט למשם רנדומליות ברכיב הפתעות.

### 3.2 החלק הייצירתי

#### מערך חיים של Bumpy

בתחילת המשחק מוצעים שלושה חיים (לב מוקף בעיגול מסמן חיים אחד).

לדמות יורד חיים אם קורה אחד מהباءים: הסטיימה הספרה לאחרר בחלק השמאלי העליון בمسך, או פגעה באויב (בצורה של אף), או פגעה ב

covid

.

הדמות יכולה לקבל חיים אחד אם היא נמצאת בשלב השני וגם קורה אחד מהباءים: הדמות עתה לשלב הבא, או הדמות השיגה חיים עיי' התגשותה לב שמדובר באזורי המשחק.

#### מערך נקודות של Bumpy

בתחילת כל שלב כמות הנקודות של הדמות מתאפסת. הדמות משיגה נקודות אם היא התרנשה באחד הילומיים שופיעו באזורי המשחק.

#### מערך שלבים

קיים במשחק שני שלבים, כאשר על מנת לעبور מהאחד לשני על השחקן להשיג ארבע נקודות. שני השלבים נבדלים במיקום, גודל וצורת המדרגות, הרקעים, מיקום הילומיים, ובכך שהשלב השני ניתן גם לאסוף הפתעה מסווג חיים. בנוסף, מספר הלבבות שייפלו לאזורי המשחק נקבע באופן רנדומלי.

לאחר שהשחקן השיג ארבע נקודות יופיע לו על המסך במקום מוגדר דגל, שעליו לגעת בו, ורק כך יעבור לשלב הבא.

כמו כן, ניתן למשך כל זמן המשחק חיוי רציף באיזה שלב השחקן נמצא, בתתית השמאלית במסך. סימן של גולגולת משמעו הפסד, סימן של מדלית הצעירות משמעו ניצחון.

## מערכת האויב

במהלך המשחק מופיע אויב בצורה של אפ', נקרא לו obstacle , שלוגיקת התנועה שלו דומה ל-smiley-.  
בגיאת דמותה העוקבת בו תגרום להורדה של חיים אחד. דמותה העוקבת מסוגלת להתמודד עם ע"י יריה לכיוונו. לאחר שהיריה פוגעת ב-obstacle הוריד obstacle, בנוסף, bumpty משחרר בזמן רנדומלי אויבים קטנים – נקרא להם covid-ים, אשר לוגיקת התנועה שלהם דומה ל-bumpty. אם הם נוגעים בשחקן הוא מביך חיים אחד והם נעלמים. הדורך שלו להתגבר עליהם הוא לחסל את obstacle ביריה (כך ייעלמו כל האויבים), או לחסל את covid ביריה.

## מודול סאונד

במהלך המשחק ישנו צלילי חיוי למשתמש, בכל פגיעה של bumpty בהפתעה / אויב, יריה ומעבר שלב.

### **3.3 אובייקטולוגיה - ממשקים לעולם החיצון**

בפרויקט השתמשנו בקרטיס FPGA מסוג DE-10 Standard, להלן שרטוט ופירוט על אמצעי הקלטי והפלט:

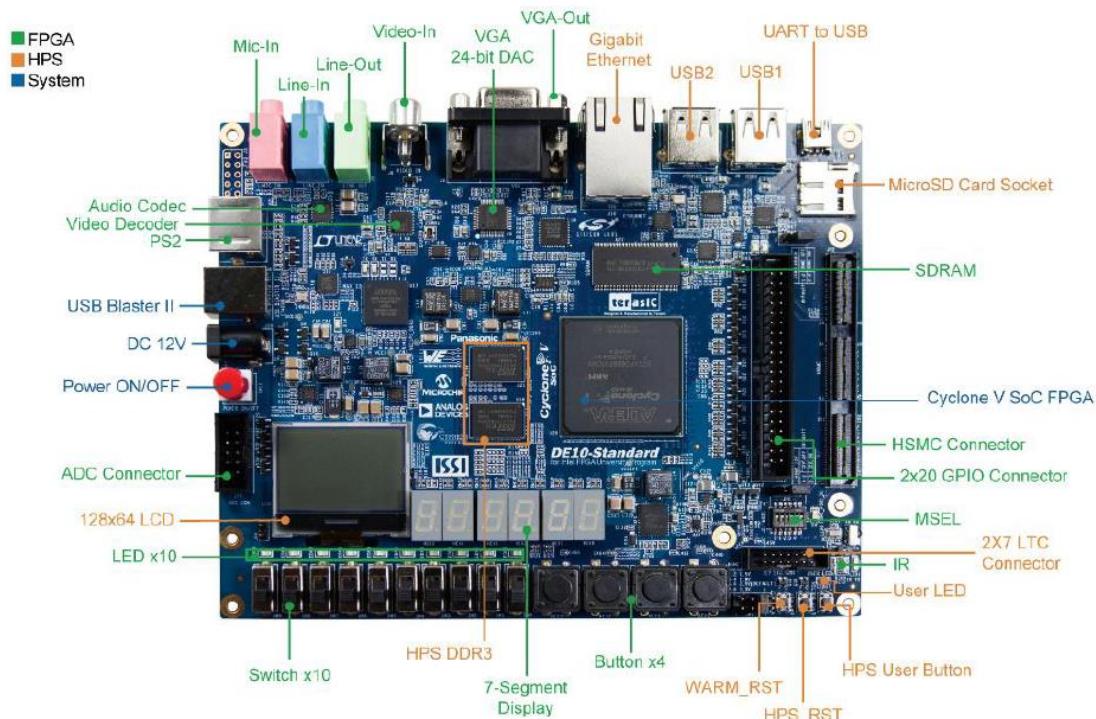
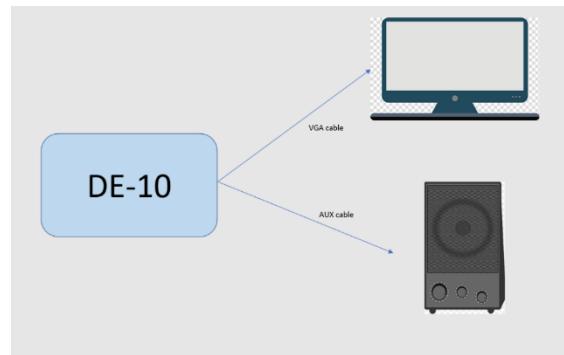


Figure 2-1 DE10-Standard development board (top view)

על מנת לשחק, ממשקנו לפרויקט את ארבעת הלחצנים בתחתית ה الكرטיס, כאשר הימני ממושך במקום ה-0. יש לציין שהלחצנים עובדים ב-0. בנוסף, אנו משתמשים בשלושת המפסקים שצמודים להחצן [KEY], אשר הם נמצאים למטה הערך הלוגי 0'. על מנת לראות את המשחק, יש לחבר בשפה העליונה של ה الكرטיס כבל VGA, ולהבררו למסך. על מנת לקבל חיויי סאונד יש לחבר רמקולים ל الكرטיס באמצעות כבל Aux.

شرطוט המכשור בפרויקט להלן:

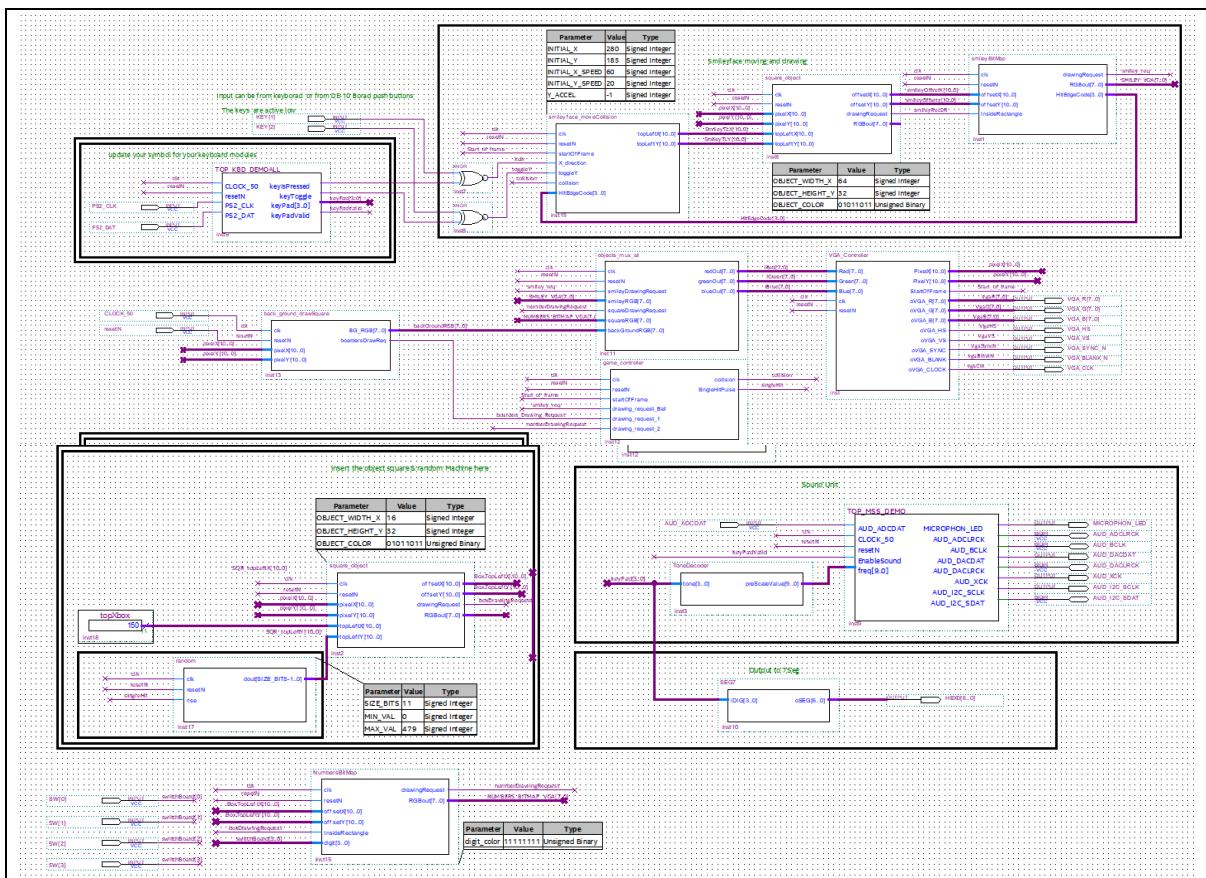


## 4 מימוש הספרת ודיוון עם המדריך

### 4.1 מטריות הספרת

- מסגרת למסך
- כללי של המשחך Top
- דמויות שזזה לפי לוגיקה מוגדרת
- שורת מדרגות
- שורה של Bitmap

## 4.2 תיאור הSENT

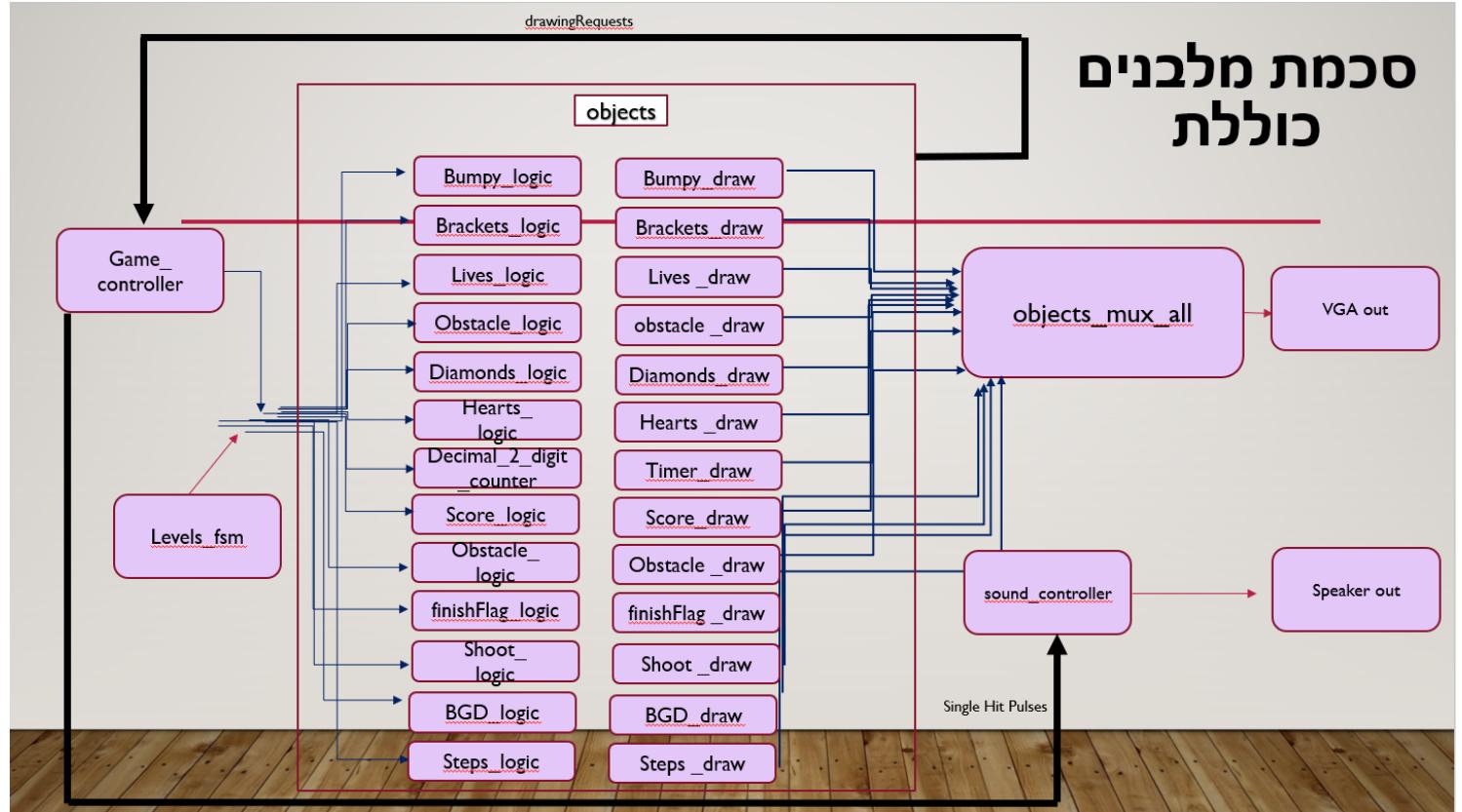


## 4.3 דיוון ומסקנות עם המדריך

הגענו למסקנה כי תחילת עליינו לוודא שלוגיקת התנועה של דמות ה-Bumpkyunk היגיונית, ואך יהיה מעניין להכניס אלמנט פיזי-קל, כגון תאוצת כוח הכבידה. כמו כן ניצור מנגן שלבים שմבדיל בין שני שלבים באמצעות סט מדרגות שונה. בנוסף עליינו להפקיד על תכנות מודולרי, חלוקה היגיונית לקופסאות שחורות שיופיעו בימיוש כמה רכיבים בעלי מאפיינים דומים, על מנת למנוע שכפול קוד.

## 5 ארכיטקטורה וסכמת מבנים פנימית

### 5.1 סכמת מבנים



## 5.2 רישימת חמשת המכלולים העיקריים, תפקידם וסדר ביצועם

מספר	שם	תקפיך	תקפיך מנוען	סדר	סיבוכיות	הfcn	עbor ה MVE	הfcn ביצוע	סדר ביצוע
1	Mux_all	לנהל את היציאה לאג'ג, לבחור ע"פ סדר מוגדר מראש מכל הכניסות המייצגות את רכיבי המשחק.	בחירה מבין: באמפי, רקע ומדרגות	קל	1				
2	Game controller	lezohot ולקבוע האם קיימות התנשויות ב프וטוקול VGA מבין רכיבי המשחק	זיהוי התנשויות של באמפי בمسجدת או המדרגות ובאיוז כיוון.	בינוני	2				
3	Bumpy_logic	לקבוע את pozot הבאמפי ע"פ לוגיקה פיזיקלית מוגדרת מראש ולהחליט כיצד יוז בעת פגיעה.	לקבוע את pozot הבאמפי ע"פ הבאמפי ע"פ לוגיקה физיקלית מוגדרת ראש ולהחליט כיצד יוז בעת פגיעה.	קשה	3				
4	Level_controller	לנהל מעבר בין שלבים, בחור מבין קומבינציות המדרגות, הרקעים וההפטעות	לא רלוונטי למשחק המנוען שכן הוא מכין שלב בודד.	קשה	4				
5	Shoot_controller	מנהל מחסנית של 8 כדורים שנוראים לעבר האויב, הופעה והעלמות שלהם, עשה reload כשנגמרה המחסנית	להוציא ירייה בודדת, ולודוא שהיא נעלמת לאחר פגיעה בעצם כלשהו	קשה	5				

## 5.3 פלוט הגדלת שני מודולים העיקריים לנצח

### 5.3.1 שיקולי בחירה

מודול covid\_controller נבחר מאחר שיש בו תחכים מסוימים שימושו בקוד באמצעות שילוב בין לוגיקה סינכרונית, קומבינטורית ואלמנט דגימה. בנוסף הכנסנו במודול זה גם אלמנט רנדומלי. لكن אלו מעוניינים להתמקד בו בעת הצגת הפרויקט.

מכונות המצלבים שאחריות על השלבים גם היא נבחרה. על פניו מכונות המצלבים עצמה איננה מסובכת, אך האינטגרציה שלה בפרויקט בסדר גודל כזה הייתה עבורהנו אתגר לא פשוט, ולבסוף אנו גאים מאוד בתוצאות שקיבלונו ממנה. האתגר שילב אילוצי timing, כמו גם בפתרונות אחרים נוספים. מודול זה מנהל את שלבי המשחק, וכפועל יוצא גם הדפסות רבות, כגון החיוויי התרמי לשחקן שמודפס תחתית השמאלית במסך, ומנגנונים נוספים.

שקלנו אם להכניס את מנגנון ההפטעות של הלבבות, אך לבסוף הוחלט שעדיין יהיה להראות רכיבים שהיוו עבורהנו אתגר גדול יותר.

### 5.3.2 מודול ראשון [covid\_controller]

שם הסטודנט	נדב אשכנזי
תפקיד מפורט של המודול	בעת פגעה ב-object.bumpy מנהל את פעלות הירייה של obstacle על bumpy, ומוריד חיים
למה הוא חשוב	המודול מכיל מגנון SMBus פועלת reload למחסנית של שמו אובייקטי קורונה, והמיושן שלו נראה לנו מעניין ודורש הבנה טובת של המערכת, תוך התחשבות בשיקולי timing. כמו כן המודול עוקב אחרי תנועת obstacle ודוגמת את מיקומו לפי הזמן ונדומל.
IMPLEMENTATION (MVP)	כ��פיך מנוון בניית המודול כך שיאפשר הוצאה של אובייקט בעל לוגיקת תנועה פשוטה מהמיקום העדכני של obstacle. ללא הורדת חיים ואפשרויות מתקדמות נוספות.
אופן המימוש	ኒצ'ר ייחידה שתוציא enable בעת לחיצה על המקש, לאובייקט כללי (general) object שינווע בלוגיקת תנועה זהה לעט bumpy, כאשר לכניסת המיקום ההתחלתי ייכנס המיקום העכשווי של obstacle. ברגע הפגיעה של האובייקט עם שחקן ייזהה gameController את התגניות, ויוציא pulse, שיוריד את bumpy enable לאובייקט הגרפי הזה ויוריד חיים.
כニיסות עיקריות	obstacle_TopLeftX, obstacle_TopLeftY, shootN (button press by timer), SHP_shootCovid, SHP_covidCollision, resetNLevelCovid – reset whenever level-related data should be reset.
יציאות עיקריות	Covid_req covid_VGA

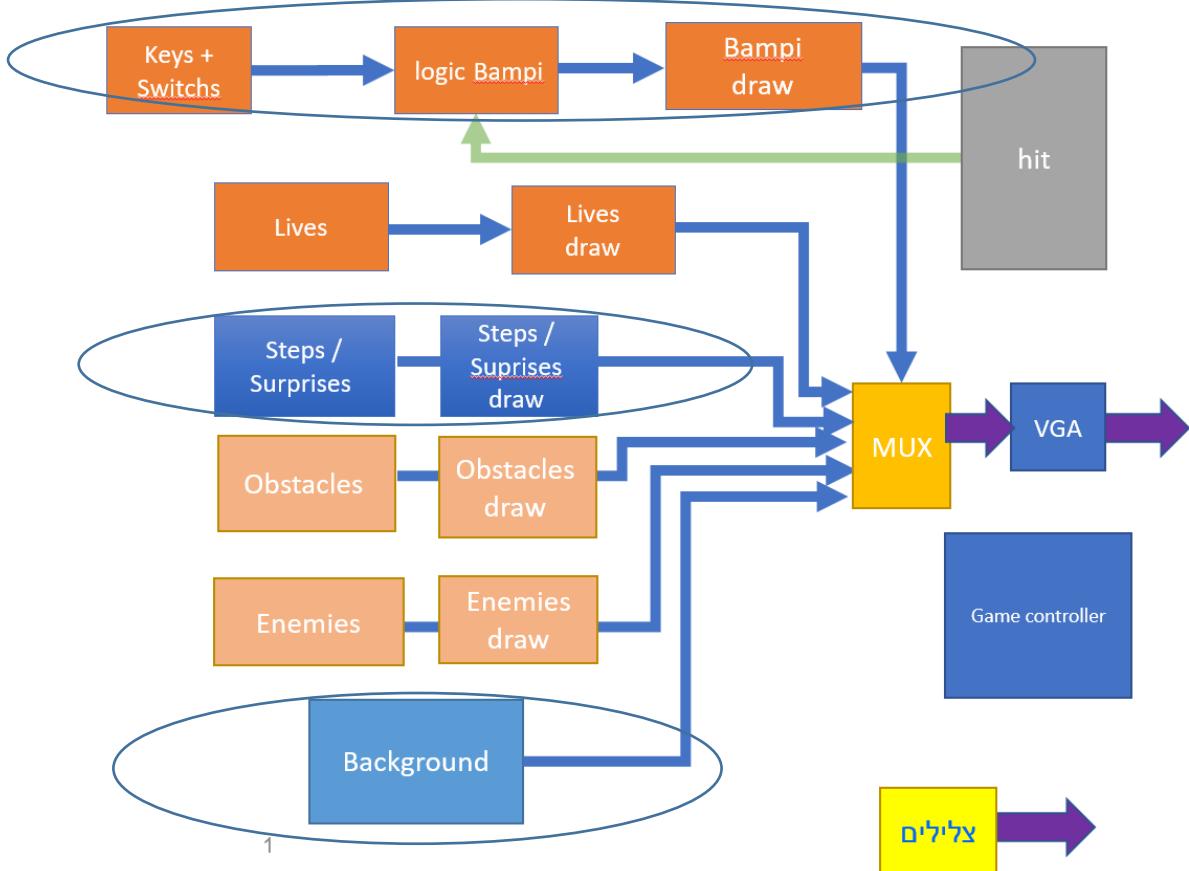
### 5.3.3 מודול שני [level\_fsm]

שם הסטודנט	אייתי סלפיאן
תפקיד מפורט של המודול	מודול זה מנהל את כל האלמנטים שימושיים בין שני שלבי המשחק, ביניהם bitmap של מדרגות, מיקום המדרגות, רקע, חיוויי השלבים בתחריטת המסך מצד שמאל, אילו הפתעות יופיעו בכל שלב. בנוסף, המודול מעדכן את מערך החיים והנקודות.
למה הוא חשוב	מודול זה מכיל ומתממשק לרכיבים רבים, ועל כן האינטגרציה שלו בפרויקט היוותה עבורנו אתגר מעניין.
IMPLEMENTATION (MVP)	לא מימושו את מודול זה MVP, לאחר שסיכמנו עם אייל להתמקד תחילית בבנייה שלב אחד, תוך הוצאה כלל הסיגנלים שיופיעו לרלוונטיים לFSM בעtid.
אופן המימוש	ኒצ'ר FSM שבעת קבלת סיגנל של 4 נקודות, הוא מאפשר הדפסה למסך של דגל שעל bumpy להשיג, ונכנס במצב שהוא מוכנה להתגניות. לאחר מכן מקבל המכונה את סיג널 ההתגניות, ותעביר במצב שלב שני. אם התהליך קורה פעם נוספת בשלב השני – המכונה תעבור למצב ניצחון. אם במהלך אחד השלבים יורדים לדמות החיים – המכונה תעבור למצב Game over .Game over
כニיסות עיקריות	gameOver – indicates life is over finishFlag – indicates collision between bumpy and finishFlag
יציאות עיקריות	levelUp – single pulse indicating level up levelCode[1:0] – two level codes are supported

## 6 שלבים בIMPLEMENTATION הפרויקט - MVP

### 6.1 IMPLEMENTATION ה- MVP

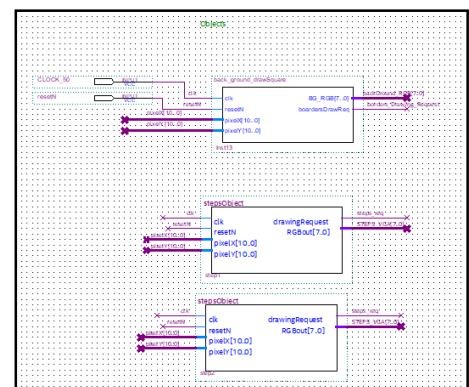
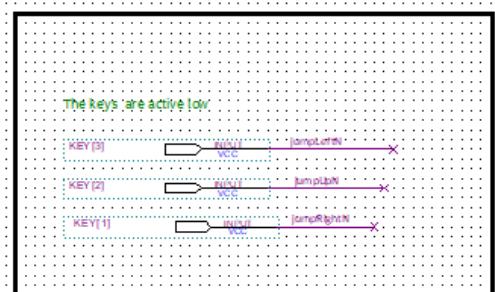
במצב המיניימי אנו מצפים שתיהיה לנו דמות שאננו בוחרים (כלומר בונים עבורה bitmap), שנעה לפי תנועה מוגדרת בתוך המסגרת של המסך, ומתאיימה את תנועתה לפי התנשויות עם עצמים במתחב (מסך, מחרוזים וכו'). כמו כן יופיעו על המסך מדרגות צבעים קבועים שכמובן מוגדרות עבורה התנשויות.



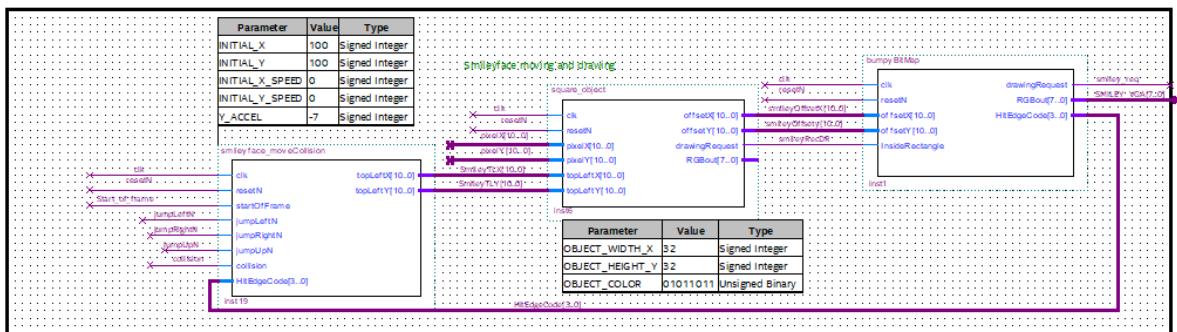
scheme of the upper hierarchy of the MVP in QUARTUS

membrane - connection of the inputs  
FPGA

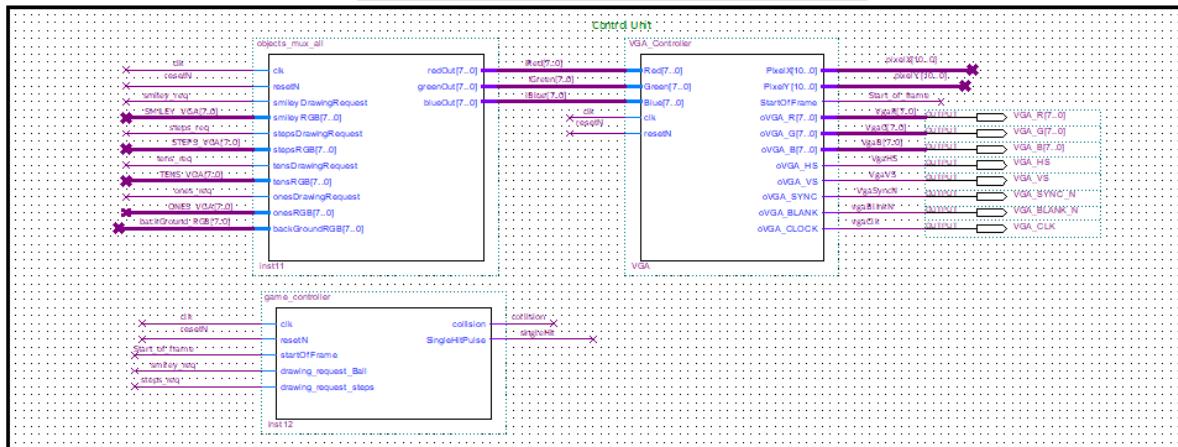
rock and two oblique  
grades



### Bumpy logic + draw



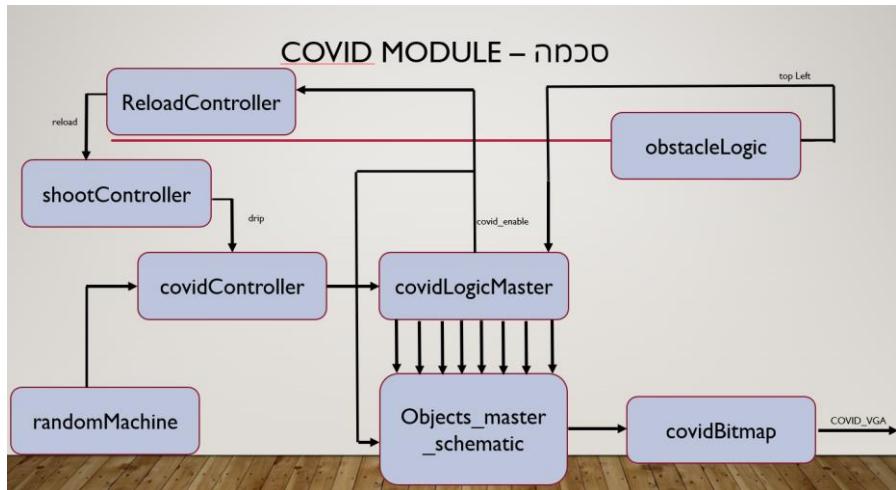
### Control unit – mux, VGA\_Controller, game controller



## 7 תיאור מפורט של שני מודולים (כמו במצגת)

### 7.1 מודול ראשון - [covid] - [נדב אשכנזי]

#### 7.1.1 דיאגרמת מבנים



#### 7.1.2 דיאגרמת מצבים (bubble diagram - בועות)

מאחר שלא מימושנו את המודול באמצעות מצבים, נפרט כאן בהרחבה על הרכיבים והאותות החיצוניים והפנימיים המשמשותיים ביוטר.

תפקידו להזקיה לירייה ע"פ תזמון שעון. – **shoot\_controller**

תפקדו לבחור באופן רנדומלי אילו אובייקטים יוכלו להופיע – **random\_controller**

תפקדו להחליט אילו אויבים יופיעו על המסך ברגע נתון. – **covid\_controller**

מקבל כניסה enable, וגם את TLX, TLY של ObstacleTLX, TLY של covidLogicMaster – **covidLogicMaster** – מנה את תנועת היריות כולל תאוצה.

מכיל שמונה אובייקטים שמקבלים enable ומופיעים על המסך עם Bitmap של Objects master schematic מזרק

הרכיב מקבל את אותן enable, וכאשר לירייה האחורה יורץ reloadController (כלומר פוגעת בעצם כלשהו במשחק) נגמרת המחסנית של שמונה האובייקטים. ברגע זה המודול עושה את חול למחסנית האובייקטים כך שניתן יהיה להמשיך ליריות.

**דgeshtim ו הערות לגבי השרטוט:**

המודול מתרսט על ידי האות resetNLevelCovid, שיורד ל-0 לוגי כאשר עוברים לשלב הבא, או כאשר נדרשים לבצע פעולה טעונה של היריות (לאחר שמונה יריות).

N shootN זהו וקטור שיוצאת מobjects\_master\_schematic אשר מאנד בתוכו את אותן המסללים את יירה מסויימת נורמה ומופעה כעת על המסך, עבר כל אחד משמונה האובייקטים שהמודול מחזיק.

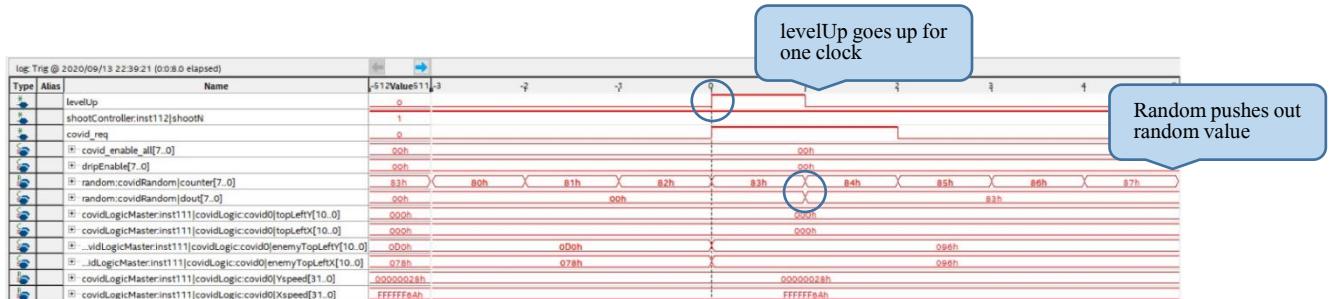
#### 7.1.2.1 פרט את המצבים העיקריים

לא מימושנו את המודול באמצעות מצבים, ופירטנו קודם לכן על המודול ועל פועלתו במפורט כנדרש.

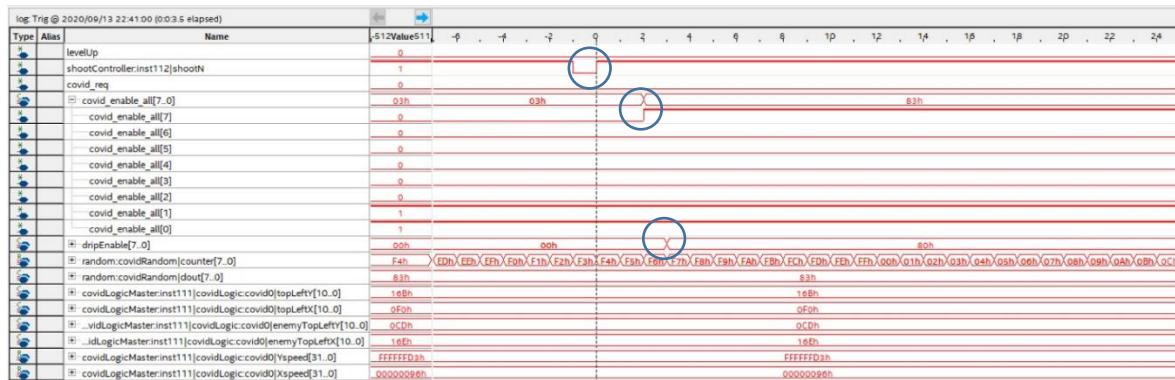
#### 7.1.3 מסך(י) סימולציה של המודול

יש לבדוק את כל הכניסות והיציאות, כל מקרי הקצה וכל המקרים המיוחדים.

אם יש צורך, הצג את תוכניות הסימולציה במספר חלונות. מעל כל חלון כתוב מה הוא בודק. סמן **בעזרת חיצים** על **דיאגרמת הזמן**, את **מקום הבדיקה**. ולמה אתם מצפים (ראו בדוגמה למטה) וודאו שבחלון הסימולציה רואים את רשימת האותיות ואת ציר הזמן.



בחלק זה בבדיקה בדקנו אם מכונת random אכן מוציאה ערך רנדומלי של enable לשמונה אלמנטי covid כאשר עלויים לשלב הבא. ואכן, כאשר ערך levelUp עולה, ייציאת מכונת random דוגם ומוציאה את הערך מהזור השעון הקודם, צפוי.

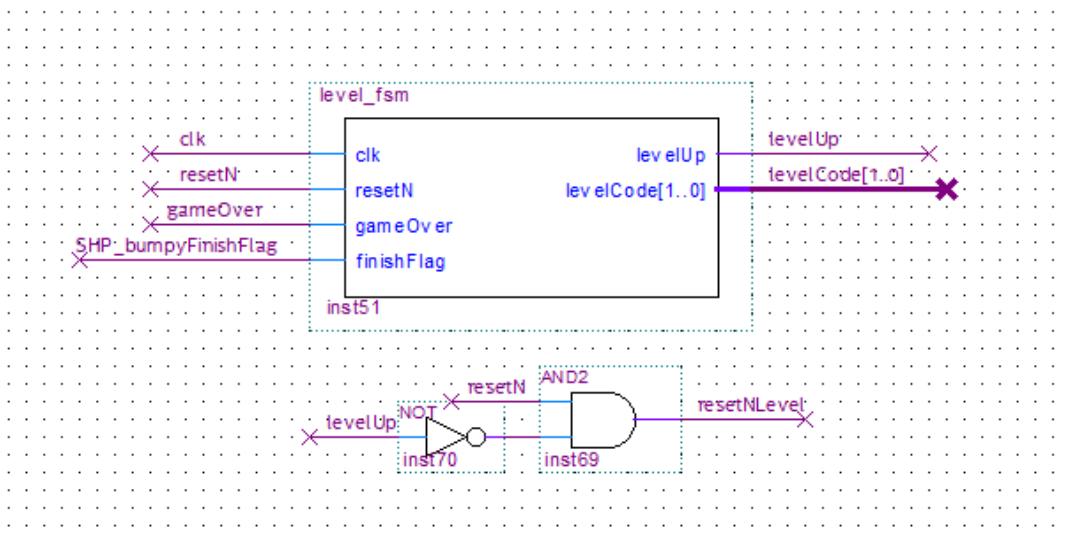


בדיקה זו בchner אם בעט ניסיון של obstacle covid (עליה של shoot, מסומנת בעיגול הראשון), אכן covid\_enable משנה את ערכו (מסומן בעיגול השני). וקטור זה מאפשר, בנוסף לכך שינוי covid\_enable\_all על הממד covid. וכן, אם רואים שלאחר מחוור שעון אחד ערך [7,0,0,0,0,0], שידפס על המסך covid\_enable\_all. וכן, אם רואים שלאחר מחוור שעון נוסף רואים covid\_enable\_all עלה ל-1, כלומר covid שבע יכול להיות מודפס (כמו גם 0,1). ולבסוף, לאחר שעון נוסף רואים שdrip\_enable גם שיינה את ערכו מ-0 ל-80. לבסוף שני וקטוריים אלו ייכנסו לשער and שיחליט באופן סופי אם, כאשר drip\_enable יינתן (באופן אוטומטי על ידי המערכת), אכן covid יודפס למסך. כך בדקנו בהצלחה את המערכת הרנדומלית שלן לירית covid.

## 7.2 מודול שני - [level\_fsm] - [אייתי שלפיאן]

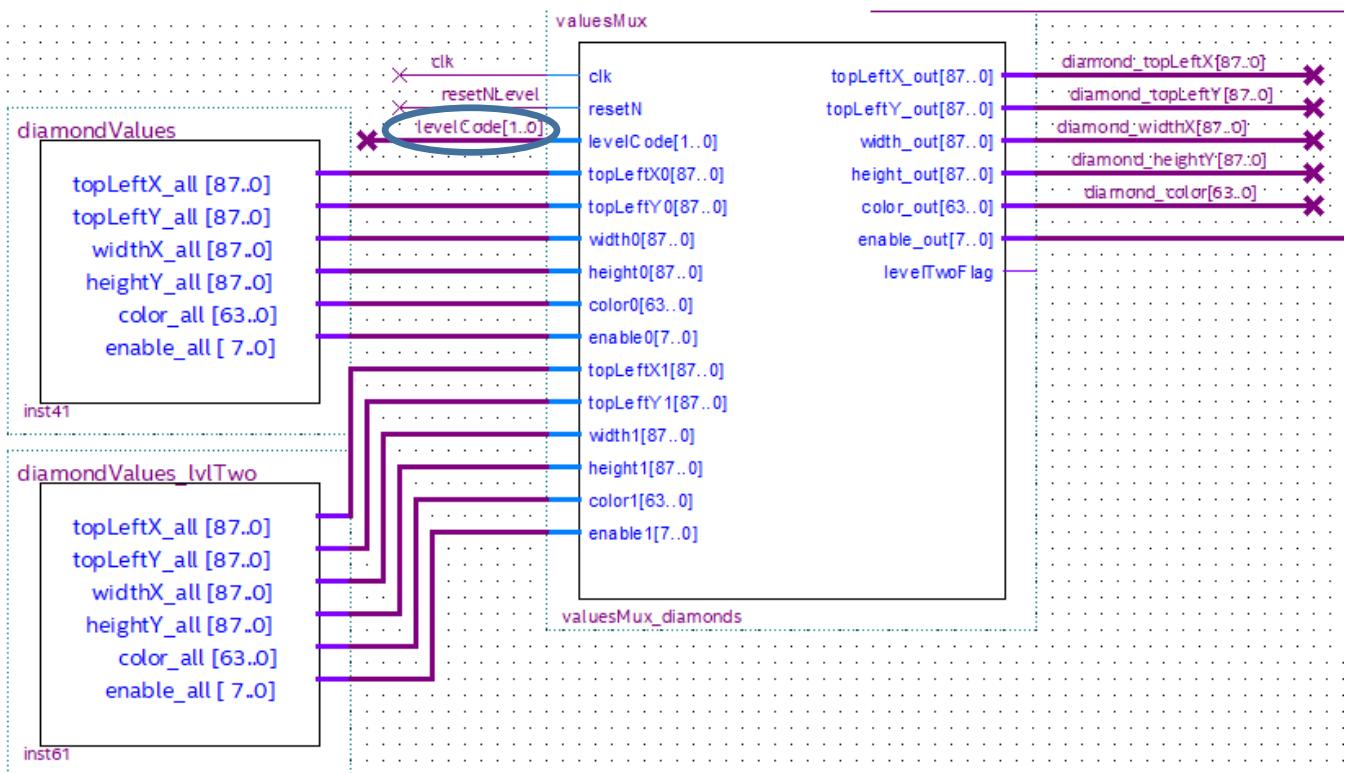
### 7.2.1 דיאגרמת מלבים

מכונת המצלבים עצמה מחליטה על מעבר בין מצבים לפי שני אותות – gameOver – gameOver שמתתקבל מ-SHP\_bumpyFlaggameController ו-SHP\_bumpyFlaggameController שמתתקבל מ-SHP\_bumpyFlaggameController. ומסמן שהדמות התנשאה בדגל שמעבירו שלב. כזכור מבניית המודול, בנינו קו שמרסט את כל האלמנטים שאמורים להתרסט במהלך מעבר שלב, וכך יחולם, לבבות ויריות – resetNlevel. שקרה רק אם לא מתבצע resetNlevel וגם עולמים שלב.



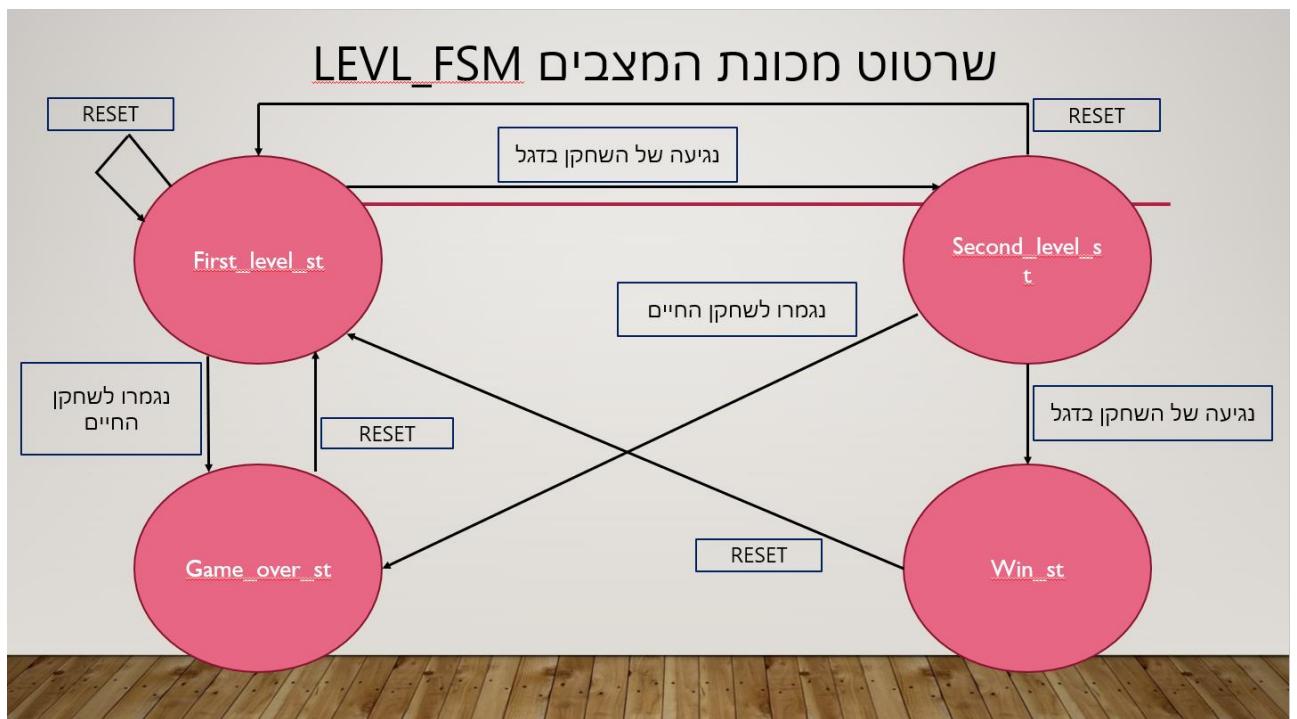
החלק המעניין לטענו הוא המנגנוןים המשמשים ביציאות ה-fsm. נפרט על שניים מהם :  
1. יציאת levelUp (שעליה ממשך מחזור שעון אחד) היא זו שוצרתת את ספירת מכונת random ובקצ מגדרה את המיקום הרנדומלי של הילומים שיוחזרו למשחק.

2. חלק ממנגנון מעבר השלבים, הוגדרו אונט-ים שבוררים אלמנטים משתנים במהלך מעבר שלב השני, כגון ערכי const של מיקום המדרגות והילומים שפוזרים לאורץ ציר x, bitmap של הרקעים והמדרגות, שרגלי sel שלדים מתקובלים מ-fsm. לדוגמה :



## 7.2.2 דיאגרמת מצבים (bubble diagram - ביעות) (bubble diagram - outputs)

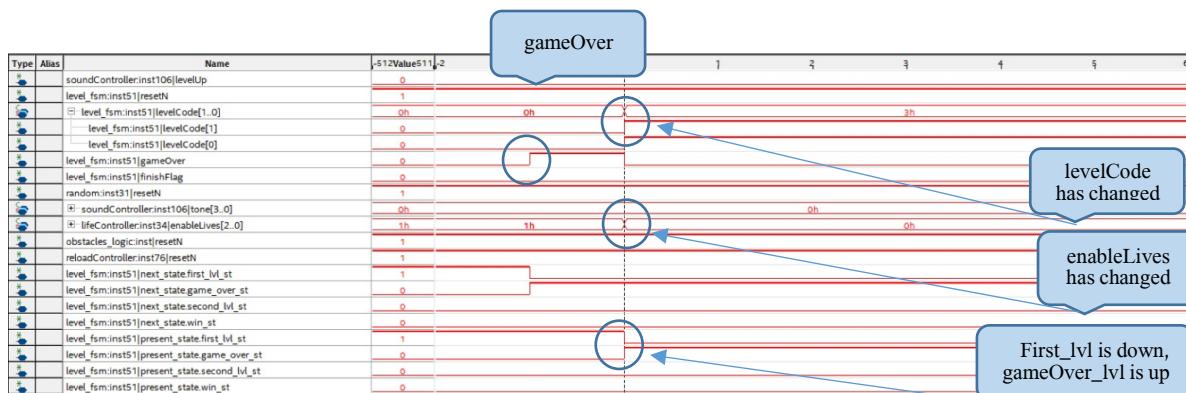
לתחליכים אותם מימשת בעזרת מכונת מצבים, צייר את דיאגרמת המצבים



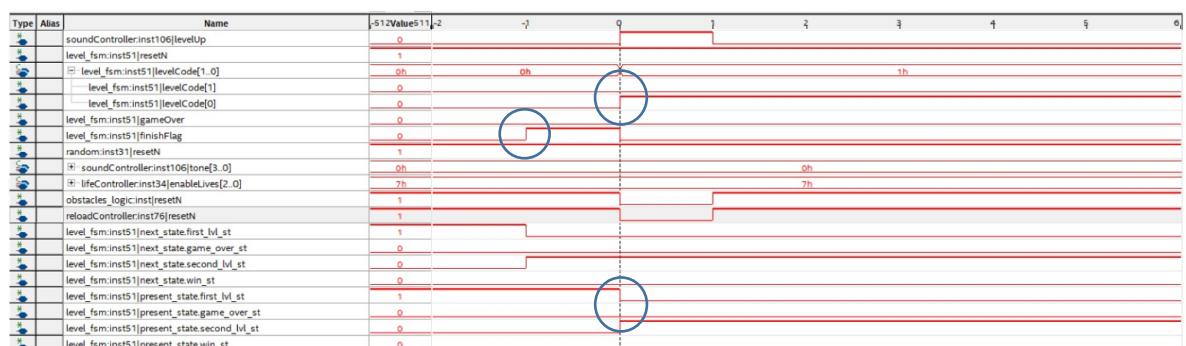
### 7.2.2.1 פרט את המצביעים העיקריים -

שם המצביע	פעילות עיקרית	לאייזה מצב עוברים מהמצביע הנוכחי ובאיזה תנאים
First_level_st	מעבר אותו שינוי מצב ומשנה את קוד השלב ל 00	עובר למצב second_level_st כאשר כניסה SHP_bumpyFinishFlag עליה (כלומר הדמות נוגעת בדגל) ואנחנו לא נמצאים.reset
Second_level_st	מעבר אותו שינוי מצב ומשנה את קוד השלב ל 01	עובר למצב win_st כאשר כניסה SHP_bumpyFinishFlag עליה (כלומר הדמות נוגעת בדגל) ואנחנו לא נמצאים.game_over_st כאשר כניסה gameOver reset
Win_st	מעבר אותו שינוי מצב ומשנה את קוד השלב 10	עובר למצב first_level_st.resetN כאשר ניתן
Game_over_st	מעבר אותו שינוי מצב ומשנה את קוד השלב 11	עובר למצב first_level_st.resetN כאשר ניתן

### 7.2.3 מסך(י) סימולציה



בבדיקה זו הינו מעוניינים לבדוק שיציאת fsm אן תקינה ומעברו שלב בעת קבלת סיגנל. אנו רואים שנitinן fsm אוות gameOver, ומחזיר לנו אחד לאחריו אן מעביר ממו 00 (first\_lvl\_st) למצביע 11 (game\_over\_st). כמו כן החיכים ירדו, וכמובן שגם המצביעים עצם כפויים שמסומן על גבי צילום המיסך.



בבדיקה זו ראיינו שאן כאשר הגיעו לשם finishFlag bumpy (מסומן בעיגול השני מלמעלה), אן מעלה את המשחק לשלב השני (מעלה את levelCode כמסומן בעיגול הראשון), והצביעים גם עוברים באופן תקין (העיגול השלישי).

Type	Alias	Name	-512Value511	-2	-1	0	1	2	3	4	5	6
soundController:inst106	levelUp		0									
level_fsm:inst51	resetN		1									
level_fsm:inst51	[levelCode[1..0]		1h	1h	1h	1h	1h	1h	1h	1h	1h	1h
level_fsm:inst51	[levelCode[1]		0									
level_fsm:inst51	[levelCode[0]		1									
level_fsm:inst51	gameOver		0									
level_fsm:inst51	finishFlag		0									
random:inst31	resetN		1									
soundController:inst106	tone[3..0]		0h									
lifeController:inst34	enableLives[2..0]		7h									
obstacles_logic:inst1	resetN		1									
reloadController:inst76	resetN		1									
level_fsm:inst51	[next_state.first_lv_st		0									
level_fsm:inst51	[next_state.game_over_st		0									
level_fsm:inst51	[next_state.second_lv_st		1									
level_fsm:inst51	[next_state.win_st		0									
level_fsm:inst51	[present_state.first_lv_st		0									
level_fsm:inst51	[present_state.game_over_st		0									
level_fsm:inst51	[present_state.second_lv_st		1									
level_fsm:inst51	[present_state.win_st		0									

בבדיקה זו בחנו אם המערכת מביאה מבייה lvl\_two\_st lvl\_two\_st. אנו רואים שהמכונה מתחילה במצב , ובעיגול השני המסומן המערכת מזזה התנגשות .finishFlag. لكن המערכת, כמצופה, מעבירה למצב win\_st (מסומן בעיגול השלישי), ומשנה את levelCode (מסומן בעיגול הראשון).

## (S.T.) Signal Tap 8

התקלה שביצענו בה signal-tap הייתה כאשר ניסינו להוסיף אויב קורונה.

הקדמה לנבי האופן שבו בנוינו את המודול שמכיל אובייקטים של אויב קורונה:

יצורנו instantiation של מודול בשם objects\_master\_schematic שמכיל שמונה אובייקטים, שמחוברים לBitmap אחד של covid, ומתקבלים את אותן התוצאות. דרך מודול זה הם מוצאים בקשוט collision\_all\_objects. נציין שלכל אובייקט קטן יש כניטת collision. מבנהו הופעה על המסך: האויב המרכזי (שאינו נחשב covid) מוצג כל 4 שניות אויב covid, שלוגיקת התנועה שלו דומה לעט Bumpy בפרויקט. כך יוצא שעל המסך מופיעים בו ומנית מספר אויב covid קטנים.

תיאור הבאג:

הbag שגילינו היה שכאשר אחד מאובי covid התנגש בקרקע, כל אויב covid שנמצא בمسך היו מסתנכרנים לתנועה שלו. כך יצא שאם לדוגמה covid1 מתנגש בקרקע כאשר covid2 נמצא באוויר, האחרון גם היה מתחילה לפוץ אליו המיקום שלו בziej ה-*הגע* אליו הרגע רצפה, על אף שהוא נמצא באוויר באותו הרגע.

:signal tap

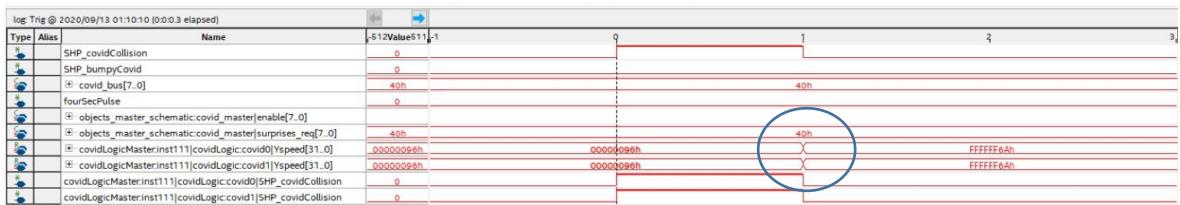
הכנסנו שני אובייקטים פנימיים SHP\_covidCollision, Yspeed של שני covid-instantiation, ובנוסף את SHP\_covidCollision הכללי (שנמצא בשורה הראשונה), שנדלק כאשר אחד מאובייקטי covid מدلיק SHP\_covidCollision והוא גם יהיה הטויגו.

Type	Alias	Node	Data Enable	Trigger Enable	
*		SHP_covidCollision	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
*		SHP_bumpyCovid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
C		+ covid_bus[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
*		fourSecPulse	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
C		+ objects_master_schematic:covid_master enable[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
C		+ objects_master_schematic:covid_master surprises_req[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
R		+ covidLogicMaster:inst111 covidLogic:covid0 Yspeed[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
R		+ covidLogicMaster:inst111 covidLogic:covid1 Yspeed[31..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
*		covidLogicMaster:inst111 covidLogic:covid0 SHP_covidCollision	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
*		covidLogicMaster:inst111 covidLogic:covid1 SHP_covidCollision	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

:signal tap

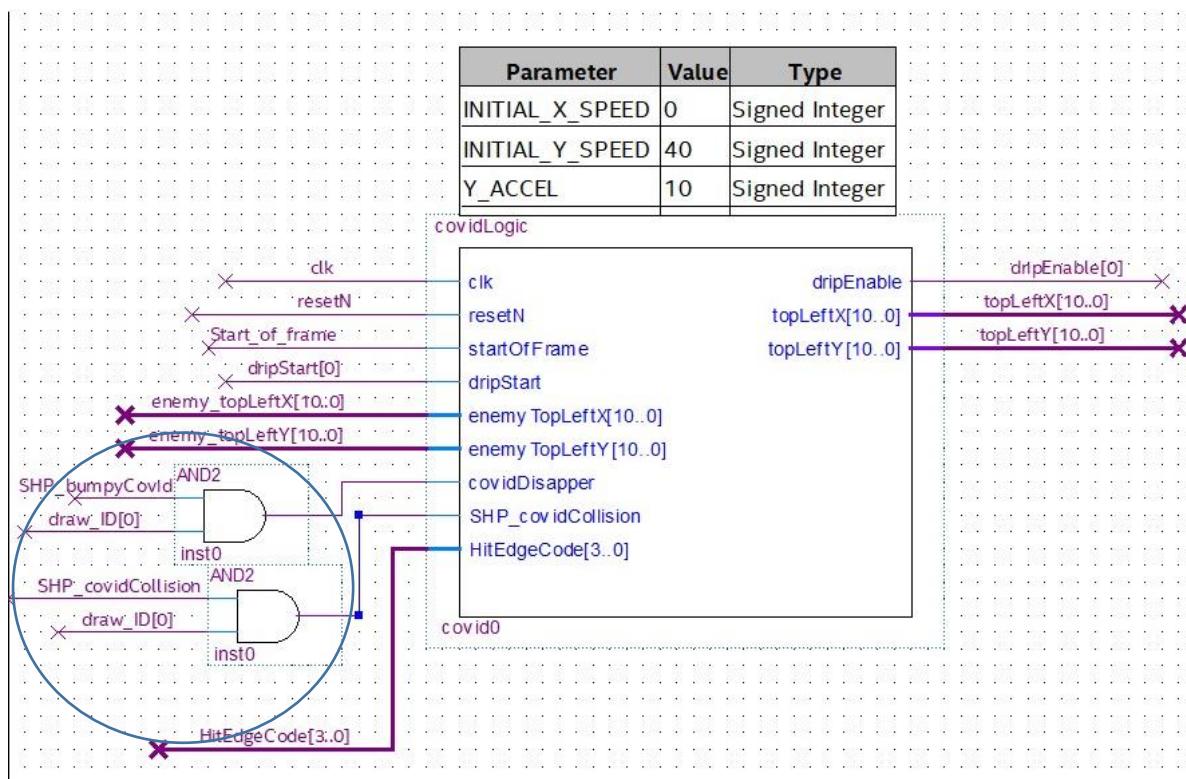
מצאו (כפוי) שאות הכניסה SHP\_covidCollision שמקבלים שני covid-instantiation-ים עולה ביחד ובנוסף מהירות שני האובייקטים משתנה בזמנית.

כמoven שבתקן VGA הדבר בלתי אפשרי – הרי שינוי המהירות מוגדר לקרות רק בעת התנגשות. לאחר שלא הגדרנו התנגשות בין שני covid-ים, בהכרח הפגיעה הייתה בין אחד covid-ים לרצפה. בתקן VGA התנגשות מזוהה על ידי ניסיון של שני אובייקטים לכתוב על אותו פיקסל, וכך אנו מסיקים שהסיבה לבאג היא שכל האובייקטים קיבלו את אותן התוצאות collision.



### הצעה לפתרון:

הגענו בנוסך גם להבנה שכנית covidDisappear צריכה לקבל גם כו שער and covidCollision נספ שבו לעתים אויב covid לא נעלם למרות שהיה צרי. השער collision נקבע כאות כניסה למשתנה draw\_ID[i] בין draw\_and SHP\_covidCollision ו-objects\_master\_schematic. על ידי הגדרת שער i המיקום של האובייקט במודול objects\_mux\_all מתקבל draw\_ID המכיל את בקשות drawingRequest שמנשלחות גם(objects\_. mux\_all). כך אנו מבטיחים שכל אובייקט covidCollision יקבל אם ורק אם הוא מבקש להיכתב בנutan, וגם המערכת מזזה covidCollision.



תוצאות signal tap לאחר השינוי:

אכן אנו רואים כתף, שכאשר אותן SHP\_covidcollision עולה, רק אחד האובייקטים משנה את מהירותו בציר ה- $y$  כפי שהיינו רוצים, ובכך הבאג נפתר!



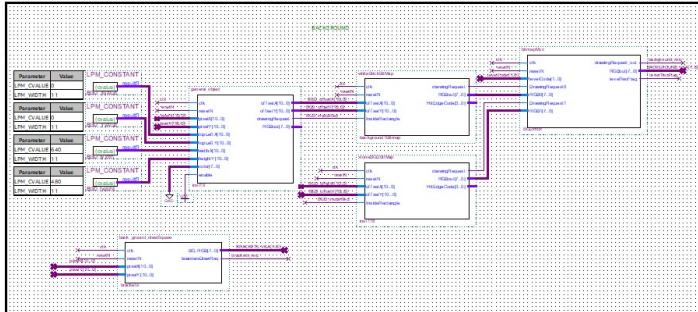
## 8.1 דיוון ומסקנות עם המדריך במהלך מעבדת אינטגרציה

- עליינו להתאים את מודול המדרגות כך שנוכל לבחור מבוחץ את מספר המדרגות, את הקפם של חן ומיקומן.
- עליינו להכניס אלמנט רנדומלי בפרויקט
- עליינו ליצור אובייקט נע בפרויקט
- ניתן להמשיך להפעלה או למודול הבא אך ורק לאחר שהמודול עליו עבדנו תקין לחלווטין

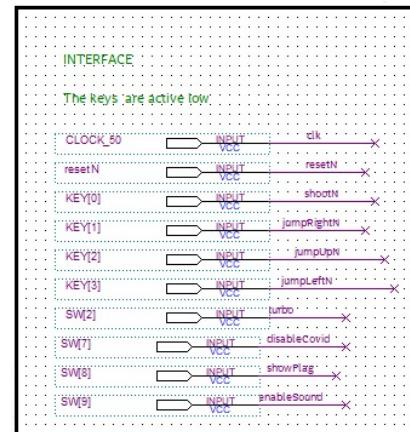
## 9 מימוש היררכיה עליונה - התוכנות לסיום הפרויקט

### 9.1 שלדים

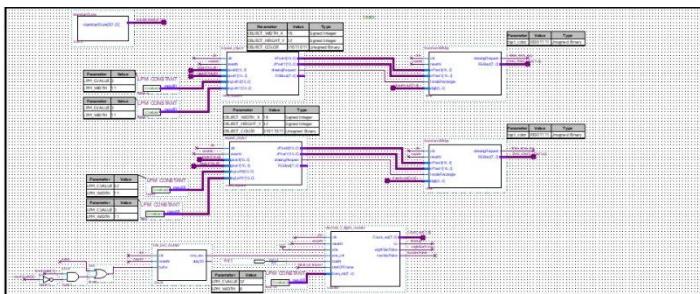
**יחידת הרקע – אחראית  
על רהט החשחה**



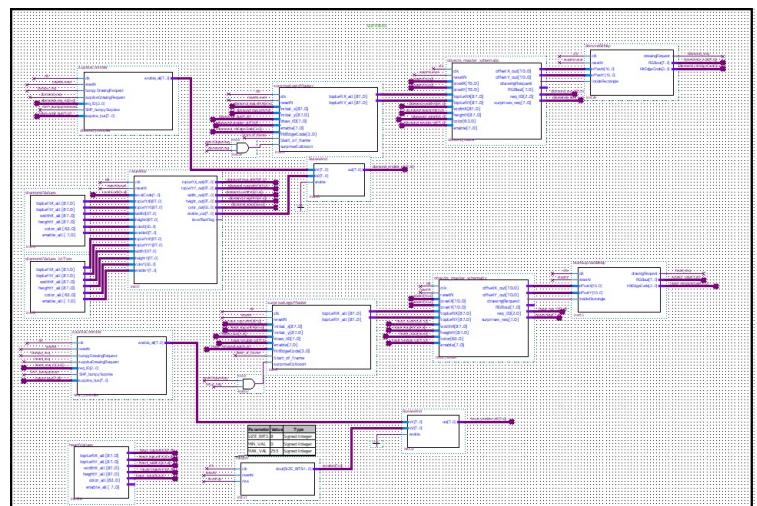
**מקשי ומתקי הכניסה**



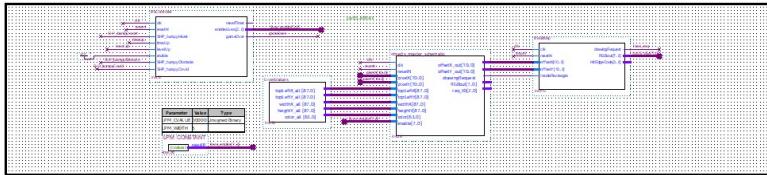
**שעון עצם – מופיע על  
המסך וסופר אחריה בכל  
שלב**



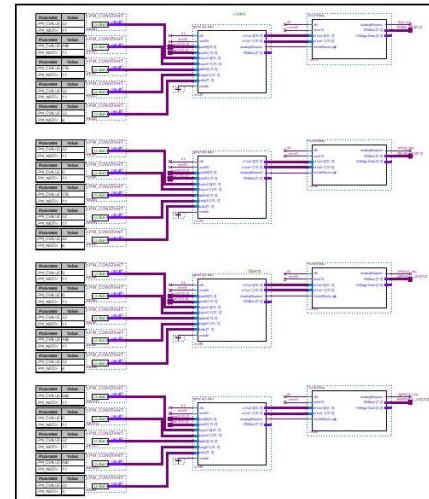
**מערכת הפתעות  
(יהלומים, לבבות)**



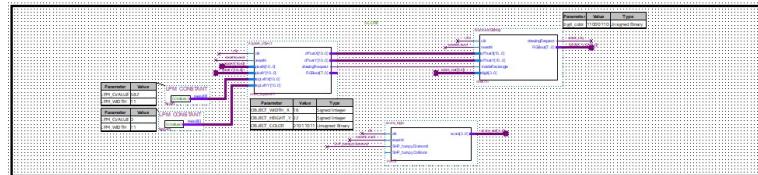
### מערכת החיים



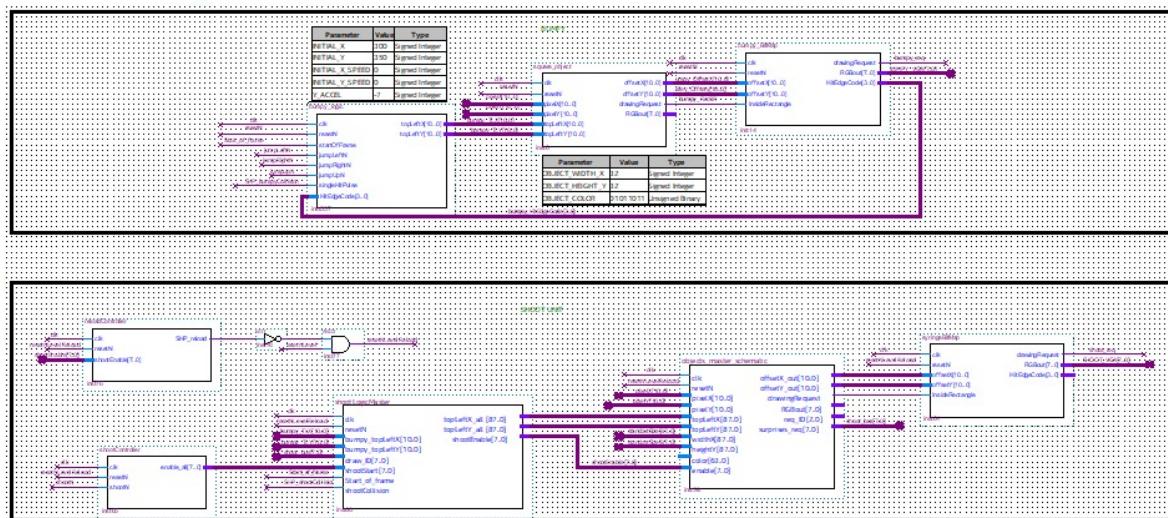
### מסגרת המסך



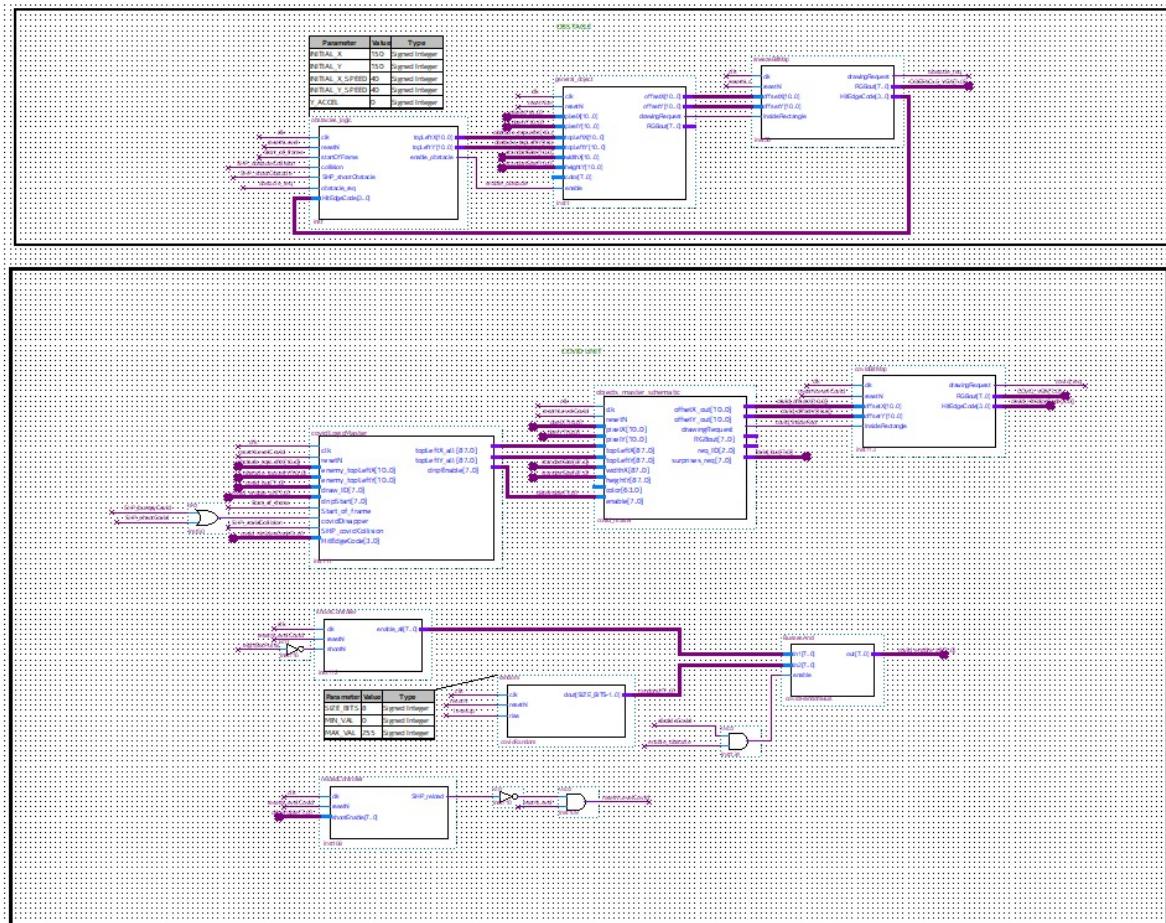
### מערכת הנקודות



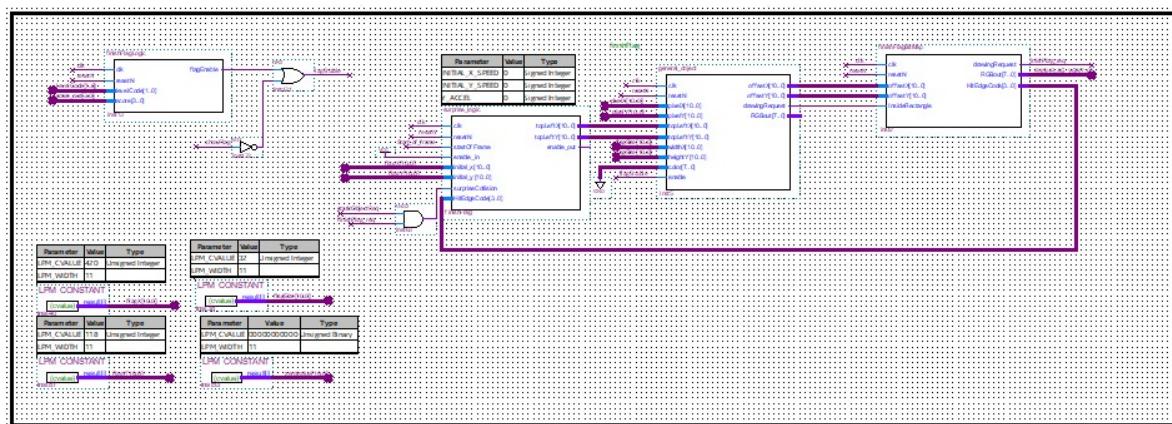
**Bumpy & shoot – לוגיקת התנועה וניהול של numpy, יריות  
שיטות ממנו, כמו גם הופעה והעלמות שלו, ותאוצה**



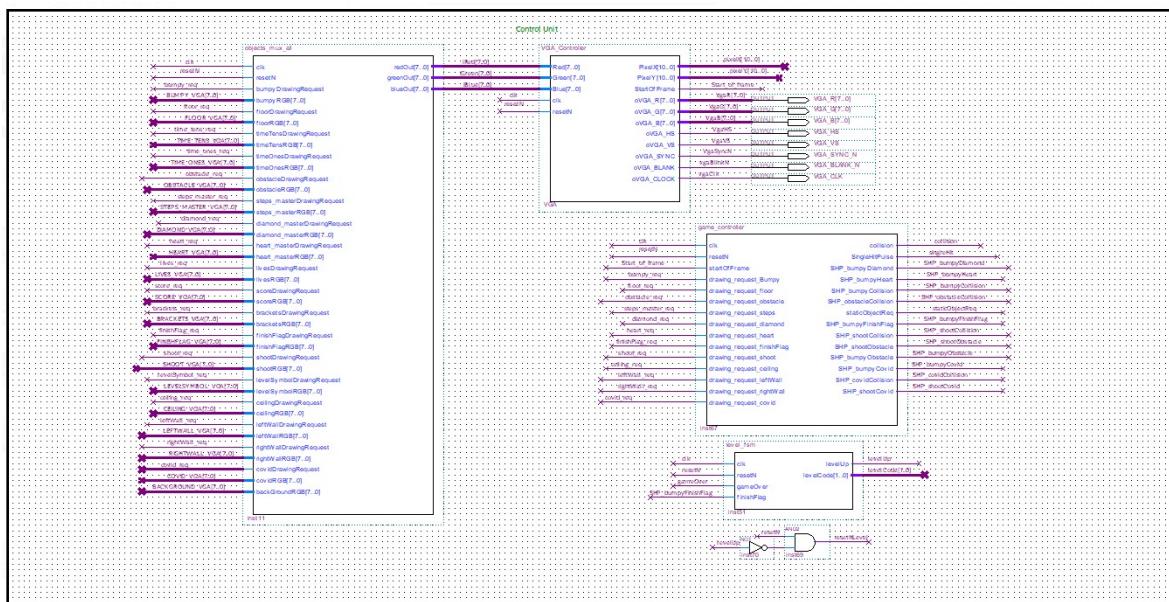
## Obstacle & covid - לוגיקת התנועה וניהול obstacle וגם אויבי covid שיווצאים ממנו



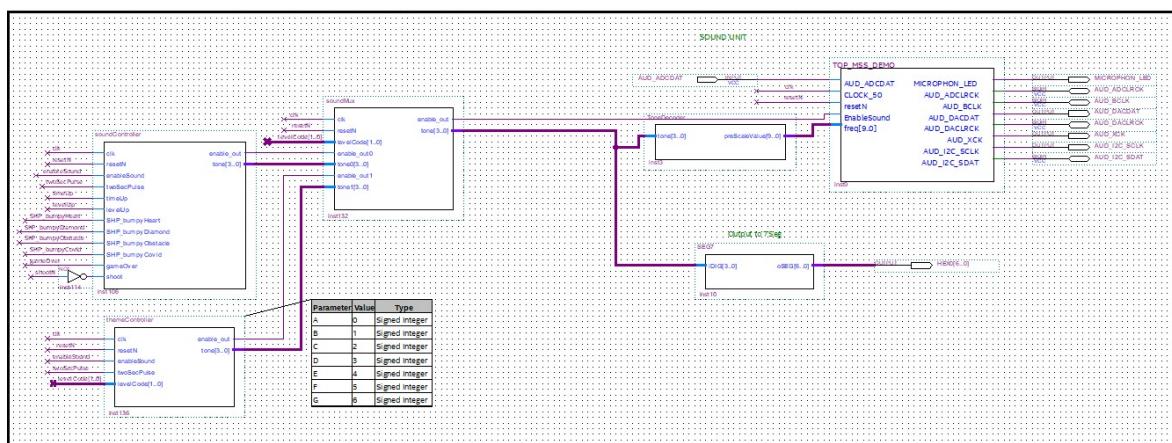
## מנהל את הופעת flag ודייוו flagController – gameController



## Control unit



## Sound unit



## 9.2 צריכת משאבים

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Sep 14 12:18:36 2020
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	Lab1Demo
Top-level Entity Name	TOP_VGA_DEMO_WITH_MSS_ALL
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	8,168 / 41,910 ( 19 % )
Total registers	4827
Total pins	56 / 499 ( 11 % )
Total virtual pins	0
Total block memory bits	16,512 / 5,662,720 (< 1 % )
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 % )



אם נסתכל על נתוני צריכת המשאבים נראה שמדובר ב-19% מהמשאבים, שזו צריכה יחסית סבירה של המשאבים הנתונים. ככל הנראה הרוב המוחלט של המשאבים מגיע מכאן שהם צורכים המונע ייחדות זיכרון. עם זאת, אנו עומדים בדרישה של הפרויקט לקמפול לפחות 50 דקוט.

## **10 סיכום ומסקנות**

- עמידה בדרישות, קשיים, פתרונות אחרים, שימוש בכלים, מסקנות.
- הפרויקט הסופי אכן עמד בדרישות ההתחלתיות.
  - הוקדש מאמץ נרחב לחלק היצירתי של הפרויקט – מערך שלבים, מערך חיים ונכודות, יהלומים ולבבות, שעון עצם, אובי קורונה נעים, מערך יריות, מערך שמע, והתמסחות בין מערכיהם אלו.
  - הפרויקט מכיל אף חלקים שלא נכללו בחשيبة הראשונית. גילינו שלא ניתן תמיד לחזות איך התוצר ייראה כבר בתחלת העבודה עליו. אופן העבודה צריך להיות, תחילת בניית signal tap לשלבים הבסיסיים, וכל פרק זמן יש לבצע design משודרג.
  - כל הדיבוג signal tap סייע לנו מאוד – שווה להשקיע זמן לפני העבודה על הפרויקט, ולהבין איך במשבצות המוקדמות להבין איך עובדים איתו וזה יחסוך לכם שעות רבות של דיבוג.

## **11 המלצות לשנה הבאה (אם יש)**

אמנם בתחילת signal tap נראה כי מעט מפחיד, אך הוא אמצעי הנדר לדיבוג. השקיעו כבר במשבצות המוקדמות להבין איך עובדים איתו וזה יחסוך לכם שעות רבות של דיבוג.

## **12 נספחים: דפי נתוניים, דפי מידע שונים בהם השתמשת.**

DE-10 Standard User Overview:

<https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=1081&PartNo=2>