הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל

מעבדה 1

פרויקט סיום
תבנית דוח מסכם

גרסה 1.0

חורף 2018-19

מחברים: אברהם קפלן, דודי בר-און

סטודנט	שם פרטי	שם משפחה
1	שחר	גוטליב
2	צחי	פרץ

שם הפרויקט	ביליארד
שם המדריך הקבוע	יוני ג'

תוכן עניינים – פרויקט

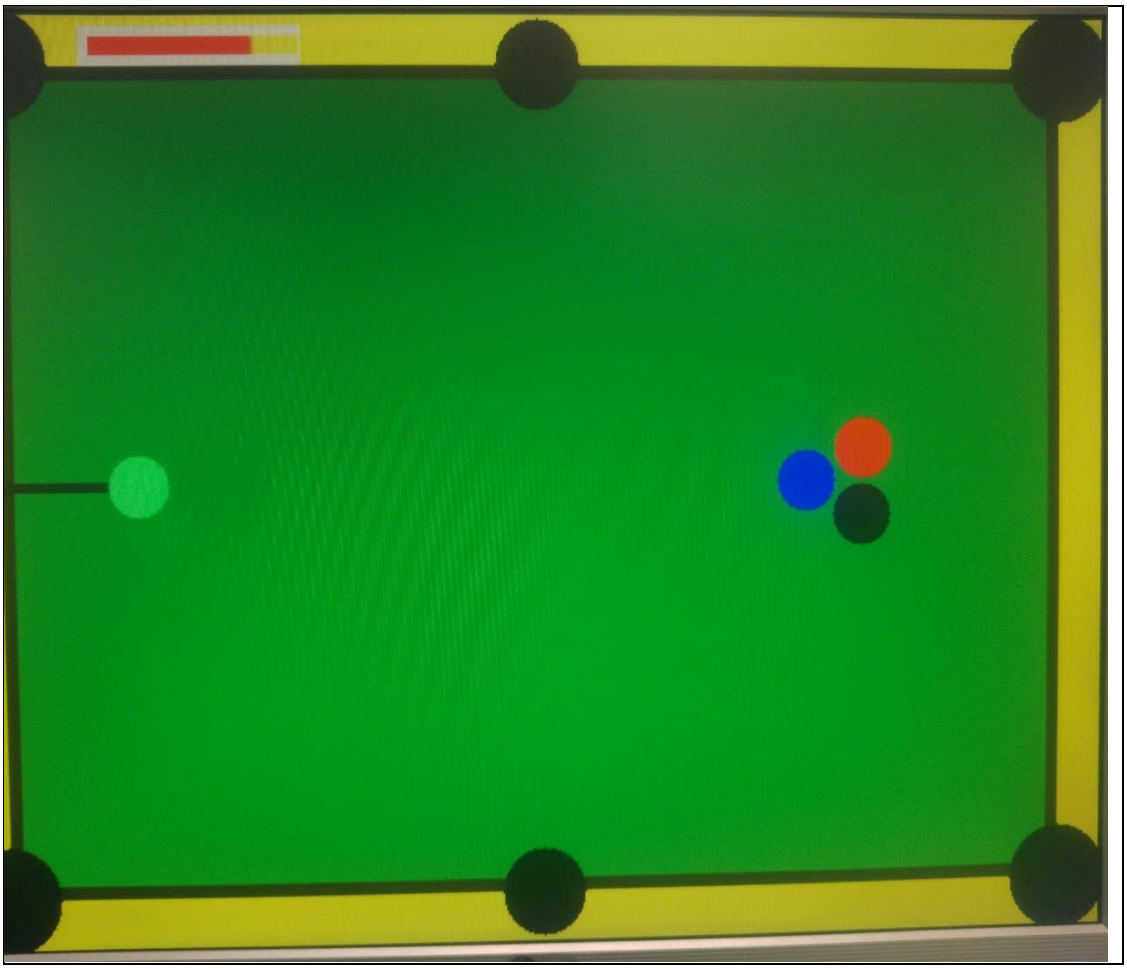
Contents

1 נספח מנהלתי

תיאור	תאריך	שם המדריך	הערות ומסקנות
דיון בהגדרת הפרויקט	30.4	יוני ג'	
סכמת מלבנים סיפתח	30.4	יוני ג'	
סכמת מלבנים PIPE	30.4	יוני ג'	
מכונת מצבים של כל הפרויקט	30.4	יוני ג'	
הגדרת שני המכלולים העיקריים	7.5	יוני ג'	
CODE REVIEW	28.5	יוני ג'	
דיונים על בעיות	28.5	יוני ג'	

2 הקדמה

2.1 צילום של הפרויקט



2.2 הנחיות כלליות

- מטרת הדוח לתעד בצורה מלאה את פרויקט הסיום שבצעתם.
- יש לכתוב בצורה מלאה וברורה, כך שנתן יהיה בעתיד על סמך קריאת הדוח, להבין את הפרויקט.
- יש לוודא שכל השרטוטים, הסכמות, האיורים, הגרפים, התמונות וכו' ברורים ומובנים. שרטוט מ QUARTUS ע"י: סימון השרטוט, העתק, הדבק, ולא Print-Screen.

- בכל אחד מפרקי הדוח, יש לציין את החלק השייך לתוספת היצירתית.

3 אפיון הפרויקט

3.1 הדרישות המקוריות מהפרויקט -(כמו במצגת)

- מסך (שולחן) ירוק עם 6 חורים בדפנות
- כדור לבן שנע בתנועה אלסטית (מתנגש בקירות) ונכנס לחורים
- כדור נוסף באינטראקציה של התנגשויות (אלסטית, שימור תנע)
- מוט נע ומקנה מהירות (בכיוון המוט) לכדור הלבן
- ניהול משחק – הוספת ניקוד.
- צלילים בסיסיים – כדור/חור, מכה.

במידה וחסרו פרטים בהגדרת בפרויקט, הוסף את ההנחות שלך לפיהם פעלת.

- מוט המשחק מסתובב סביב הכדור
- שליטה במוט בעזרת המקלדת
- שחקן יחיד

3.2 החלק היצירתי

הדרישות הנוספות מהפרויקט כתוצאה מהחלק היצירתי שהוספת.

- להוסיף מספר כדורים בצבעים שונים (3 כדורים נוספים מלבד הלבן)
- חישוב פיסיקלי של התנגשויות בין הכדורים (התנגשות אלסטית ריאלית), גילוי התנגשויות אמיתי.
- חיכוך בתנועה על המשטח, התלוי במהירות הכדור.
- שליטה בעוצמת המכה של המוט בכדור הלבן בכל תור.
- חוקי משחק – יש להכניס את השחור אחרון, ללא הכנסת הלבן במכה האחרונה.

נספח:

- **משוואות למימוש:**
  - התנגשות אלסטית (בין 2 כדורים):

$$v_1' = v_1 - \frac{\langle v_1 - v_2 \, , x_1 - x_2 \rangle}{\|x_1 - x_2\|^2}(x_1 - x_2)$$

$$v_2' = v_2 - \frac{\langle v_2 - v_1 \, , x_2 - x_1 \rangle}{\|x_2 - x_1\|^2}(x_2 - x_1)$$

  - מרחק נקודה (פיקסל) מקו המוגדר ע"י 2 נקודות (לציור מקל ברוחב):

$$dist^2 = \frac{\left|(y_2 - y_1)x_{pixel} - (x_2 - x_1)y_{pixel} + x_2 y_1 - y_2 x_1\right|^2}{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

  - מטריצת סיבוב (על מנת לסובב את המקל עם כיוון השעון):

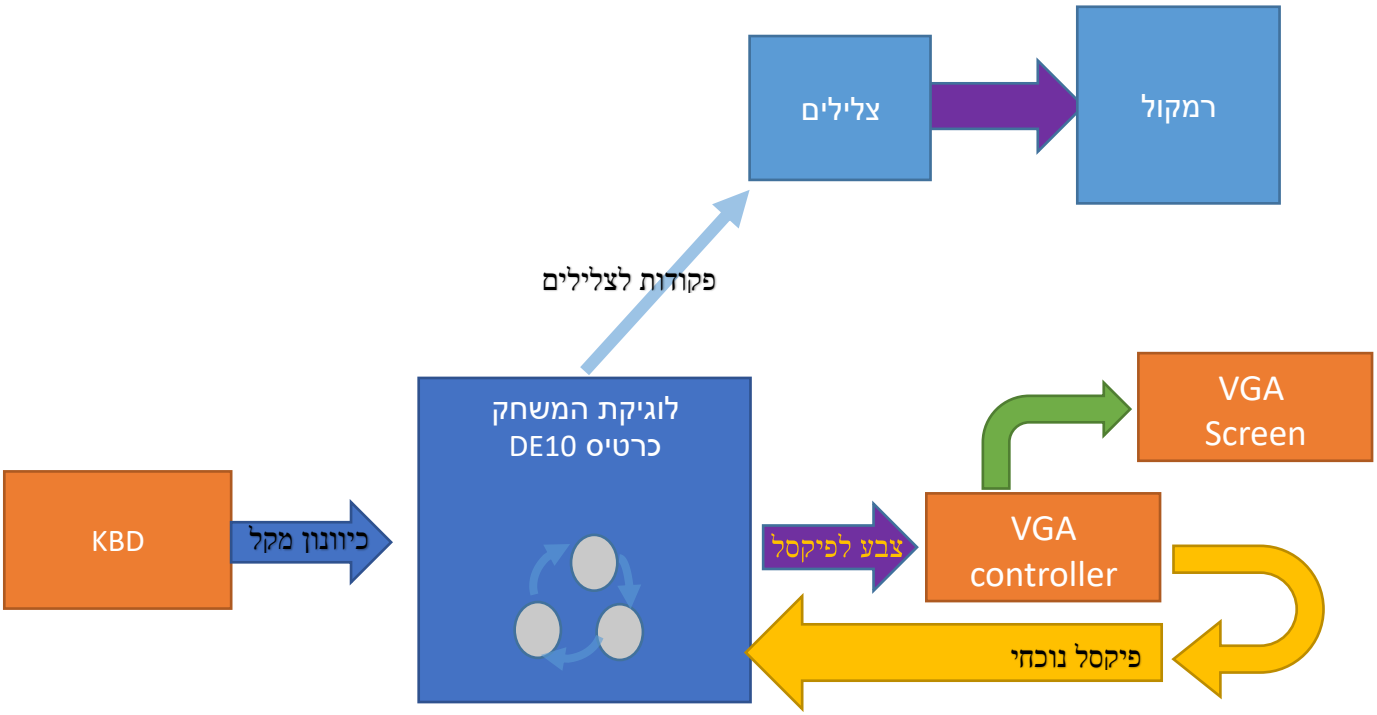
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

על מנת לסובב נגד כיוון השעון יש להשתמש במטריצהtranspose

4 ארכיטקטורה

היחידות מהן בנוי הפרויקט (כרטיסים, אמצעי קלט/פלט וכו') וזרימת הנתונים דרכן.
שרטוט המבנה והסבר תפקידה של כל יחידה. – *העזר ברכיבים מהמצגת ואל תגיש שרטוט בעפרון*

4.1 תפקיד היחידות:

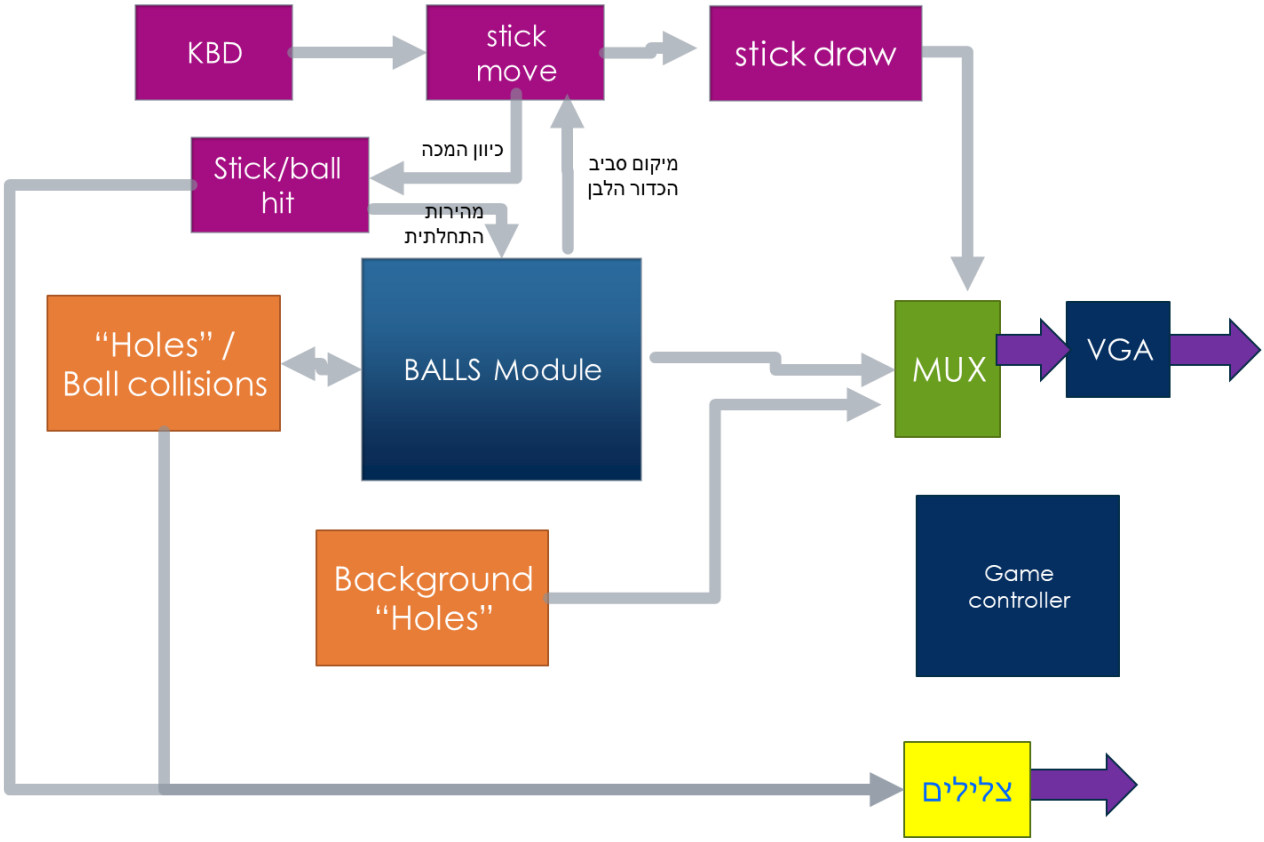


שם	תקציר פעולתה
כרטיס DE10 (קלט, פלט)	מחשב את הלוגיקה של, ואת החישובים המתמטיים הדרושים. מנהל את ומעביר את הפקודות למסך ולרמקול
מסך מחשב (פלט)	מציג את תצוגת המשחק, לוח הביליארד, הכדורים נעים, החורים וכו'
רמקול (פלט)	פולט את הצלילים במשחק. (התנגשות, ניצחון, זכייה בנקודה וכו')
מקלדת (קלט)	שימוש בחיצים על מנת לקבוע את מיקום המקל, שימוש במקש הרווח על מנת להכות בו.

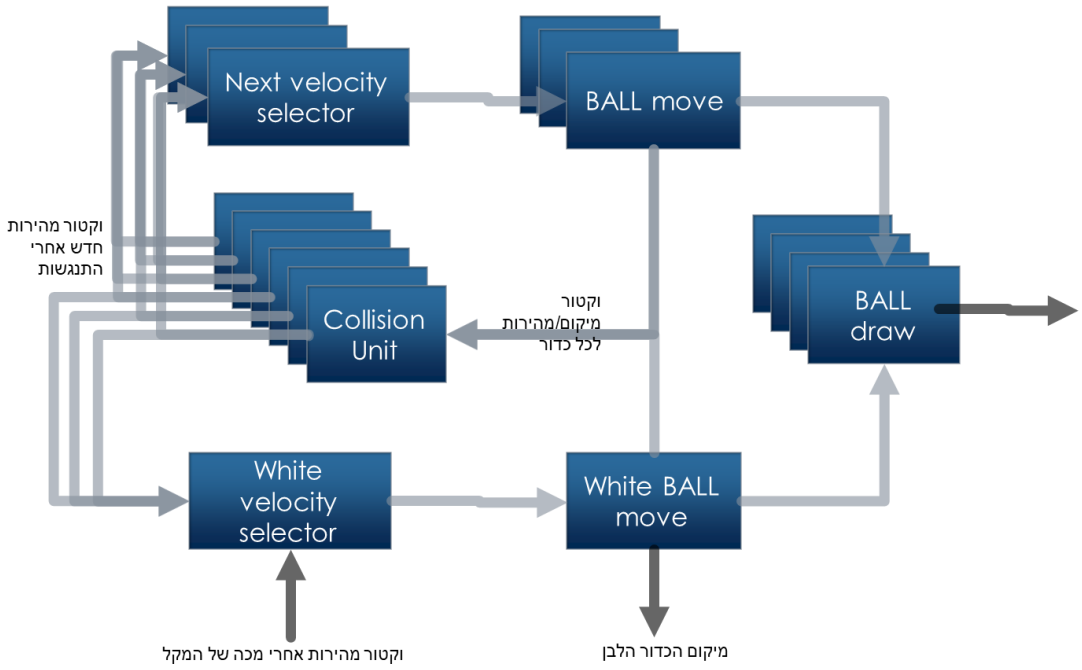
5 סכמת מלבנים פנימית

חלוקת הפרויקט למודולים פונקציונליים והקשרים ביניהם.

שרטוט ***סכמת המלבנים הכללית (VISIO או PPT)***



BALLS Module:



רשימת מכלולים (מלבנים) עיקריים, תפקידם וסדר ביצועם

פרט בטבלה את כל המכלולים העיקריים. פחות מעשרה

רצוי להתחיל עם ליבת הפרויקט (החלק הקשה/הארוך/המסובך של הפרויקט)

- בתפקיד מנוון רשום מה תעשה לפתיחת ה- PIPE
- לכל יחידה פרט את הסיבוכיות שתידרש לדעתך למימושה (קל בינוני כבד) \
- החלט מהו סדר המימוש שבחרת

מודול מס	שם	תפקיד	תפקיד מנוון PIPE	סיבוכיות התכן	סדר ביצוע
1	Collision Unit	זיהוי התנגשות בין 2 כדורים + חישוב מהירויות חדשות.	זיהוי התנגשות בנפרד לחישוב המהירות. חישוב מהירות אסינכרוני	קשה (נוסחה – התנגשות אלסטית)	1
2	Ball Move	תנועת כדור במרחב, עם חיכוך.	מעדכן וקטור מהירות ומיקום- 2 כדורים בלבד	קל	2
3	Stick draw	מצייר את המקל	מקבל 2 נקודות במרחב (קצות המקל) ומצייר אותו במרחב	קשה (נוסחה – מרחק נק' מישר)	3
4	Stick move	מזיז את המקל	מקבל אות ומסובב את המקל ב5 מעלות לכיוון המתאים	בינוני (נוסחה – כפל במטריצת סיבוב)	4
5	Stick collision	מתאר התנגשות מקל\כדור	מקבל עצמת מכה ומחשב מהירות התחלתית לכדור	קל	5
6	kbdTop	מקבל חץ לסיבוב המקל, רווח למכה (שליטה על עצמת המכה לפי משך ההחזקה)		בינוני	6
7	Game Controller	מכונת מצבים שמנהלת את המשחק	ע"פ המצב נותנת יציאות enable מתאימות לאובייקטים, וסיגנלים לסאונד \ תצוגה וכו'	בינוני-קשה	7

5.1 פרוט ארבעת המודולים העיקריים

רשום תת פרק לכל מודול אותו תתכננו (לא לבחור מודול שולי כמו ה MUX

5.1.1 [Collision Unit]

תפקיד מפורט	זיהוי התנגשות בין 2 כדורים וחישוב מהירויות חדשות לאחר ההתנגשות. החישוב מתבצע ע"פ משוואה לחישוב מהירות בהתנגשות אלסטית.
מימוש מצומצם (PIPE)	זיהוי התנגשויות ע"י פיקסלים בלבד, חישוב מהירות במודול אסינכרוני נפרד.
אופן המימוש- זיהוי התנגשות	בהינתן כדור A, נסמן וקטור מהירות Va , מיקום שלו .Xa הרדיוס יסומן R 2 הוקטורים מיוצגים ע"י INT FIXEDPOINT. זיהוי ההתנגשות – זיהוי ההתנגשות ע"י חפיפת פיקסל בלבד יוצר באגים ומכריח אותנו להשתמש בדיליי גדול בין התנגשות להתנגשות. משום האופי הלא צפוי של ההתנגשויות בביליארד השתמשנו בתנאי מתוחכם יותר. על מנת שתהיה התנגשות יש לוודא את התנאים הבאים: 1) חפיפה בין הכדורים: - או ע"י חפיפה בין פיקסלים - או ע"י מרחק בין מרכזי הכדורים: $\left\|\|Xa - Xb\|\right\|^2 \leq (2R)^2$ 2) כדורים נעים אחד לכיוון השני: $\left\|\|(Xa + Va) - (Xb + Vb)\|\right\|^2 \leq \left\|\|Xa - Xb\|\right\|^2$
אופי המימוש – חישוב מהירויות	המהירויות החדשות מחושבות בכל רגע, וסיגנל של "התנגשות" יוצא רק ברגע שהתנגשות מזוהה (כך הכדורים יודעים לשנות את מהירותם) חישוב המהירות: $v_1' = v_1 - \frac{\langle v_1 - v_2 , x_1 - x_2 \rangle}{\|x_1 - x_2\|^2}(x_1 - x_2)$ $v_2' = v_2 - \frac{\langle v_2 - v_1 , x_2 - x_1 \rangle}{\|x_2 - x_1\|^2}(x_2 - x_1)$
כניסות עיקריות	- CLK - RESETN - Start_of_frame - Xa (מיוצג ע"י 2 כניסות INT ל2 הרכיבים בוקטור) - Xb - Va - Vb - Draw_requestA - Draw_requestB
יציאות עיקריות	- Va’ - Vb’ - Collision_signal

[Collision Unit Multiplier Calculation]

תפקיד מפורט	תמיד רוצים להוסיף עוד ועוד כדורים למסך. לכל 2 כדורים צריך יחידת התנגשויות, ולכל יחידת התנגשויות צריך 8 מכפלים של INT. => 16 DSP BLOCKS לכל יח' התנגשויות. עם 4 כדורים על המסך יש צורך ב6 יח' התנגשויות => המון מכפלים. הבעיה – לחומרה יש משאבים מוגבלים (112 DSP BLOCKS) ולכן יש צורך בחישוב יעיל יותר. הפיתרון – יחידה מרכזית של חישובים לכל יחידות ההתנגשויות. היחידה הנ"ל מקבלת את הנתונים הדרושים לכל יח' התנגשות ומחשבת את המכפלות הבאות: $\|Xa - Xb\|^2$ $\|Xa - Xb + Va - Vb\|^2$ $\langle v_1 - v_2 , x_1 - x_2 \rangle$ $\langle v_1 - v_2 , x_1 - x_2 \rangle \cdot (x_1 - x_2)$ סה"כ 4 מכפלות וקטוריות (8 מכפלות של INT) . בכל מחזור שעון היחידה תפלוט את תוצאות החישובים ליחידת התנגשות אחרת
מימוש מצומצם (PIPE)	לא היה קיים בPIPE (לא היה צורך במעט כדורים)
אופן המימוש	שמירת מונה פנימי, ציקלי, הסופר עד 6 (מספר ההתנגשויות). בכל מצב של המונה חישוב המכפלות ליחידת התנגשות אחרת ופליטת "חתימה" של 4 ביט של היחידה המתאימה. למשל, אם החתימה היא 0101 אז המכפלות שייצאו כפלט באותו מחזור שעון הן של התנגשות כדור A עם C.
כניסות עיקריות	הפרש המיקומים והמהירויות של כל יחידת התנגשות: - X1-X2 - V1-V2 לכל יחידת התנגשות.
יציאות עיקריות	- 4 תוצאות המכפלות הנ"ל - חתימת ההתנגשות הנוכחית

[CueMove and CueObject]

תפקיד מפורט	התפקיד המרכזי הוא הצגה ושליטה במקל המשחק . הגדרת המקל מומשה בcueobject . השליטה על מיקום המקל ביחס לכדור נעשית באמצעות החצים אשר יכולים להזיז את המקל בשני כיוונים שונים (עם כיוון השעון או נגדו) ברזולציה יחסית גבוהה (שינוי בזווית של 5מעלות כל 'קליק').
מימוש מצומצם (PIPE)	הצגה 'חלקית' של אובייקט שיידע לבצע התנגשות בכדור הלבן -- שינוי של המקל בכיוון אחד בלבד
אופן המימוש	את המקל (cueobject) הגדרנו באמצעות 2 נקודות במישור : האחת היא הנקודה הקרובה לכדור הלבן והשניה היא הרחוקה ממנו. את הקאורדינטה הקרובה לכדור הלבן הגדרנו באמצעות מיקומו של הכדור וכאשר ידוע אורכו של המקל ניתן לחשב את מיקומה ההתחלתי של הקארדינטה הרחוקה.

	לגבי cue move כעת כל תזוזה במקל (נגד כיוון השעון לדוגמא) תיעשה באמצעות כפל משמאל במטריצת סיבוב- בצורה הבאה : $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$ כך למעשה נקבל את מיקומן של הנקודות החדשות . נציין שחישוב הנקודות עם כיוון השעון מתבצעת על ידי כפל במטריצת הטרנספוז של המטריצה לעיל. בנוסף , על מנת לא לגרור שגיאה מצטברת (כתוצאה מהמכפלה בסינוס ללא FP) כאשר הזוית היא 0 או 180 נחשב את קאורדינטות המקל בצורה ישירה (כמו האתחול הראשוני) ולא באמצעות הכפל המטריצי .
כניסות עיקריות	ל <u>cuemove</u> – מיקומו של הכדור הלבן בקשה לסיבוב עם כיוון השעון בקשה לסיבוב נגד כיוון השעון <u>לcueobject</u> תוצאת חישוב cuemove - מיקומן של הנקודות שמגדירות את המקל pixelx and pixel y
יציאות עיקריות	<u>cuemove</u> מיקומן של הנקודות שמגדירות את המקל <u>cueobject</u> drawing_request vga

[CuePower]

תפקיד מפורט	שליטה על העוצמה בה המקל יכה בכדור הלבן. מקבלת ערכים בין 1 ל120 . כאשר השחקן לוחץ על מקש הרווח, בר העוצמה מתחיל לרוץ הלוך ושוב, עד לנקודה בה המשתמש עוזב את מקש הרווח. הצגה של מד העוצמה על המסך
מימוש מצומצם (PIPE)	מכה בעוצמה קבועה > מימוש מד עוצמה שמגיע למקסימום ומפסיק
אופן המימוש	למודול נכנס הסיגנל start_of_frame (30 פעמים בשניה) הגדרנו מונה שברגע שמקש הרווח נלחץ הוא עולה/יורד בכל איטרציה של start_of_frame . את המימוש של המונה שיודע לעלות ולרדת עשינו באמצעות הגדרה של דגל שערכו הוא 0 או 2 בהתאם לכיוון המניה והחישוב הוא למעשה counter=counter +1 -flag
כניסות עיקריות	space_pressed_on space_pressed_off start_of_frame
יציאות עיקריות	power

5.2 בחירת המודולים למצגת סופית

מודול	Cue Move/Object
סטודנט	צחי פרץ
למה הוא חשוב	אובייקט במשחק – בעל נפח, שמסתובב סביב נקודה בלי לשנות את צורתו היחסית.
מה נציג	• אפיון בסיסי של אובייקט מקל (2 נקודות במרחב) • הצגה של מקל – מלבן מסובב בין 2 נקודות במרחב • סיבוב המקל (סיבוב כל נקודה סביב הראשית ע"י מטריצת סיבוב) • שמירת הזווית, שימוש ברזולוציה גבוהה, תיקון טעות נגררת

מודול	Ball Collision + calc
סטודנט	שחר גוטליב
למה הוא חשוב	ה"מנוע" הפיסיקלי במשחק – חישוב התנגשות אלסטית בין 2 כדורים בעלי נפח.
מה נציג	• מהו כדור – וקטור מיקום (מרכז), וקטור מהירות, רדיוס קבוע. • התנגשות אלסטית – נוסחה • מעבר לINT • זיהוי התנגשות – תנאי משולב כולל כיוון התקדמות • חיסכון במשאבים ע"י ייצוא החישוב

6 שלבים במימוש הפרויקט

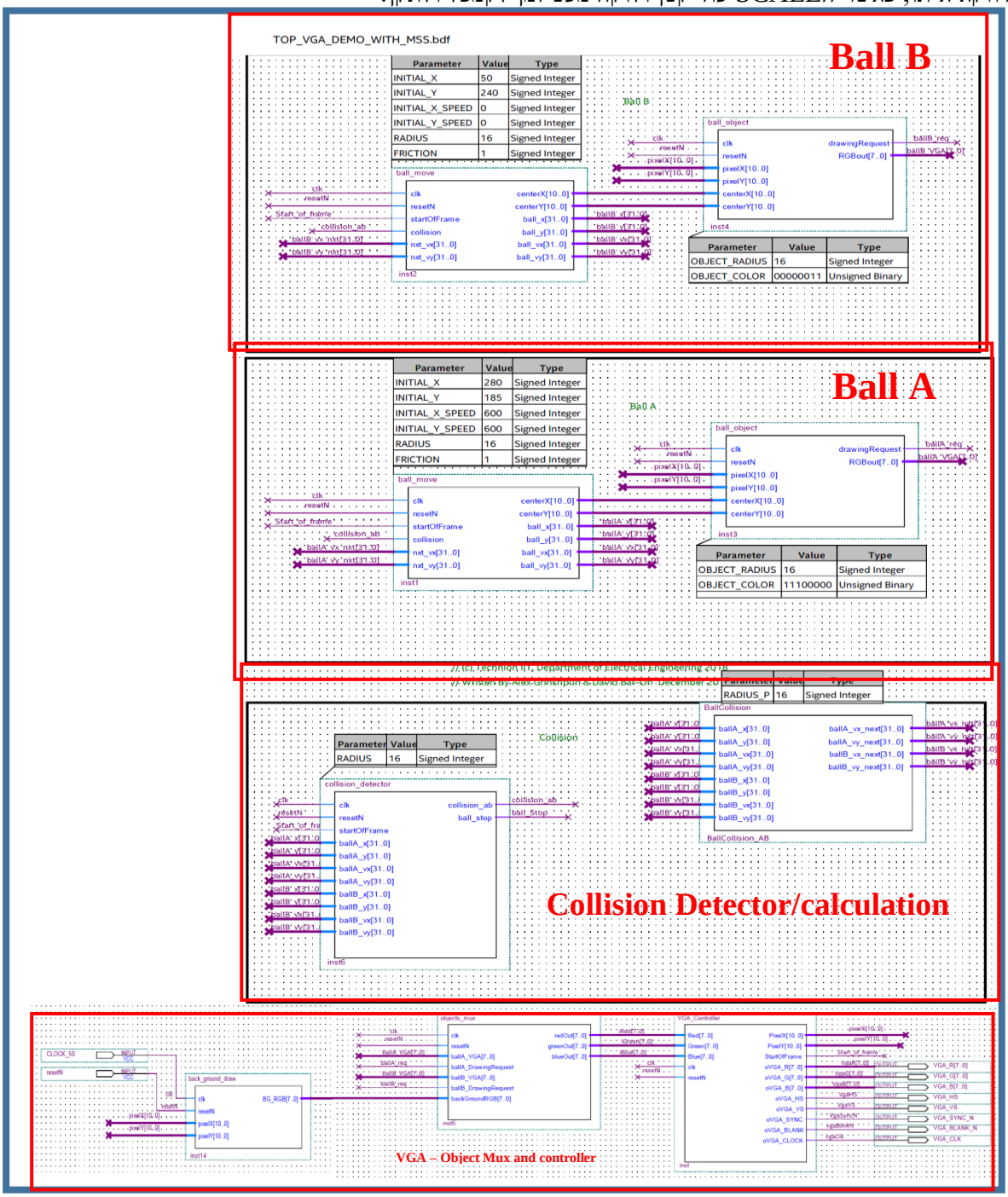
בגלל המורכבות של הפרויקט יחסית למה שתכננתם עד היום, וכדי שהפיתוח יעשה בצורה חלקה, ביצוע הפרויקט נעשה בשלושה שלבים, מהקל לכבד.

1. סיפתח – ביצוע פריט אחד או שניים הקשורים לממשקים של הפרויקט: תצוגה על מסך VGA וצליל.
2. PIPE – ביצוע מסלול שלם ומנוון של הפרויקט הדורש שיתוף מכלולים עיקריים שלו.
3. הפרויקט השלם.

חובה לבצע את כל השלבים בסדר שלמעלה וכל שלב יש לו חלק בציון על הפרויקט.
כל שלב הוא חלק מדוח הכנה בהתאם ללו"ז המופיע במודל.

6.1 סיפתח

TOP של הסיפתח – 2 כדורים שזזים על המסך ומתנגשים התנגשות אלסטית. זהו החלק הכי קשה ולכן התחלנו דווקא איתו, כאשר הSCALE עוד קטן ולוקח מעט זמן לקמפל ולתקן.

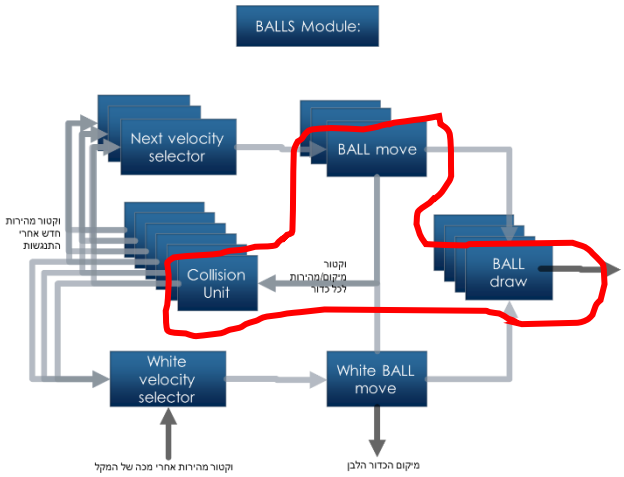


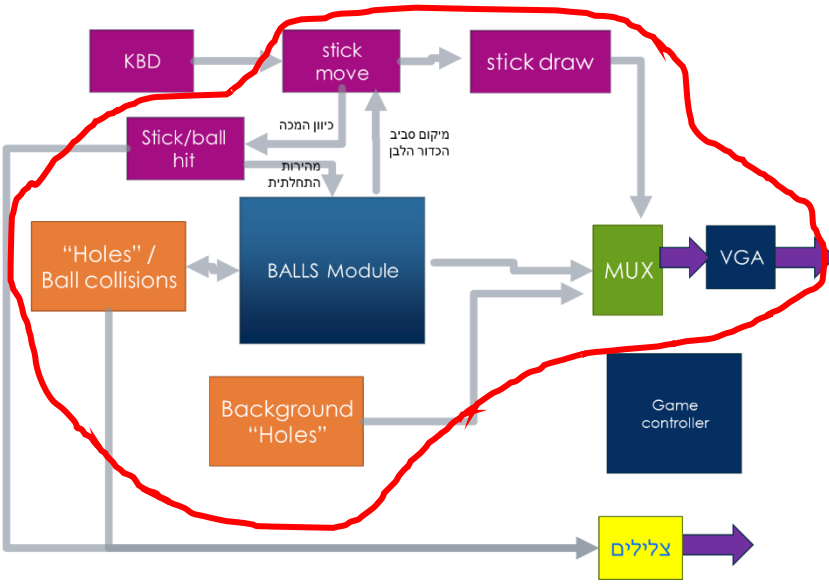
6.2 פתיחת PIPE

תאר מה יעשה ה PIPE,

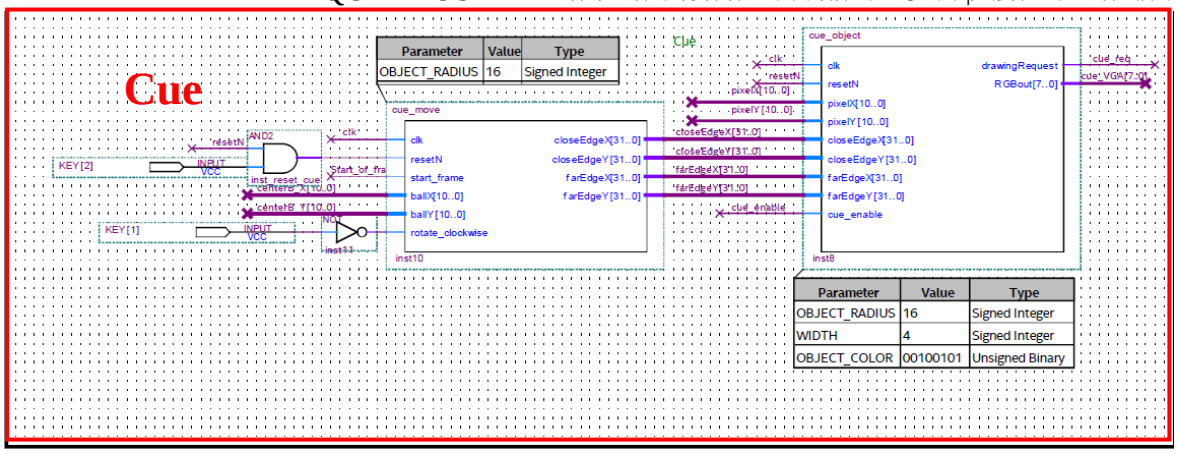
מסך ירוק (לא משופצר), עם 4 חורים בפינות. יש 2 כדורים ומקל על המסך (נשלט ע"י מקשים בכרטיס). מקל – מסתובב לכיוון יחיד, מקנה מהירות בכיוון שלו לאחד הכדורים בלחיצה על מקש. 2 כדורים – זזים עצמאית, מתנגשים בינהם, נעלמים כאשר נכנסים לחור. התנגשויות – יחידת התנגשות יחידה בין כדורים (לכן לא צריך בוררי מהירויות)

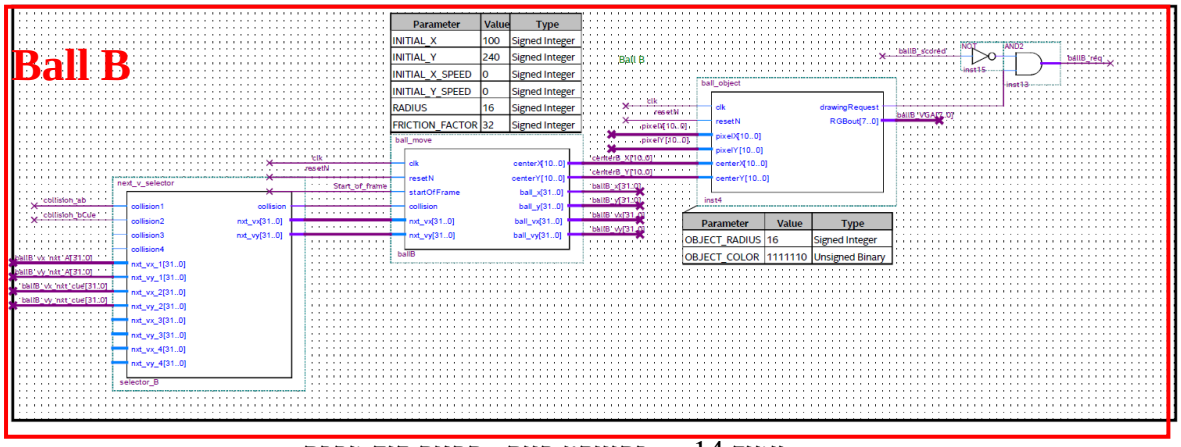
העתק לכאן את סכמת המלבנים הכללית וסמן עליה את המכלולים המשתתפים בביצוע ה PIPE

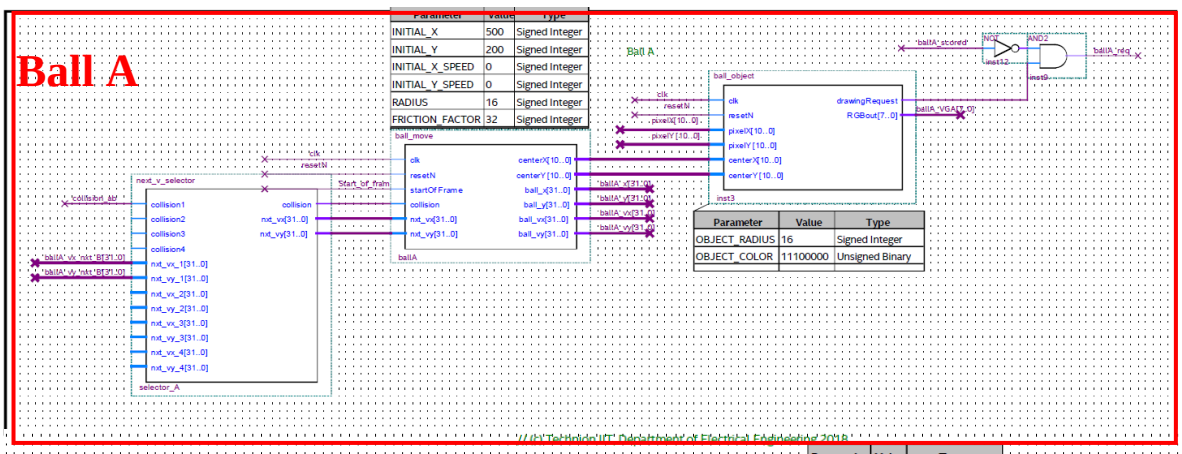


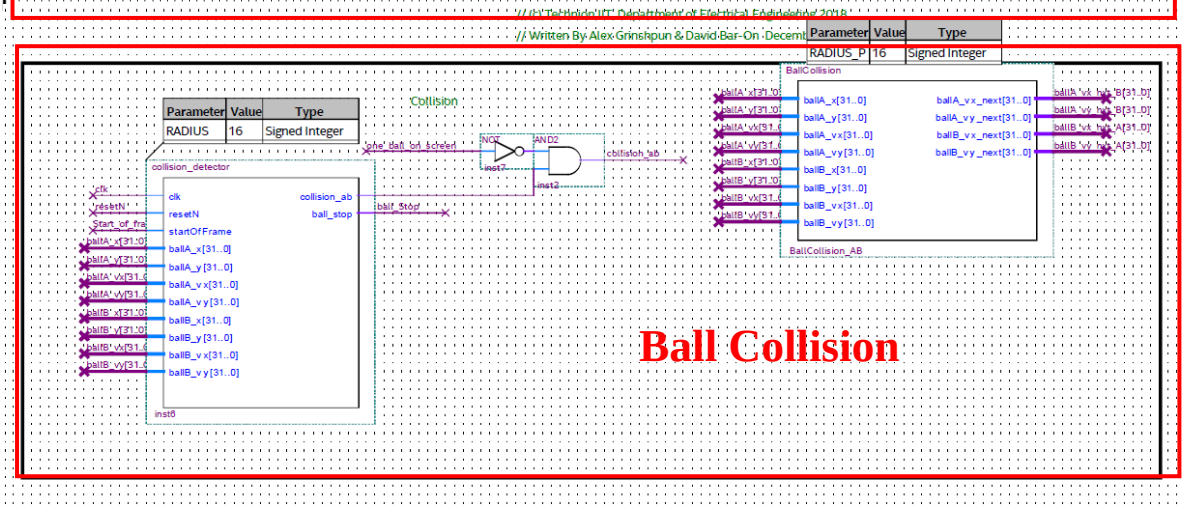


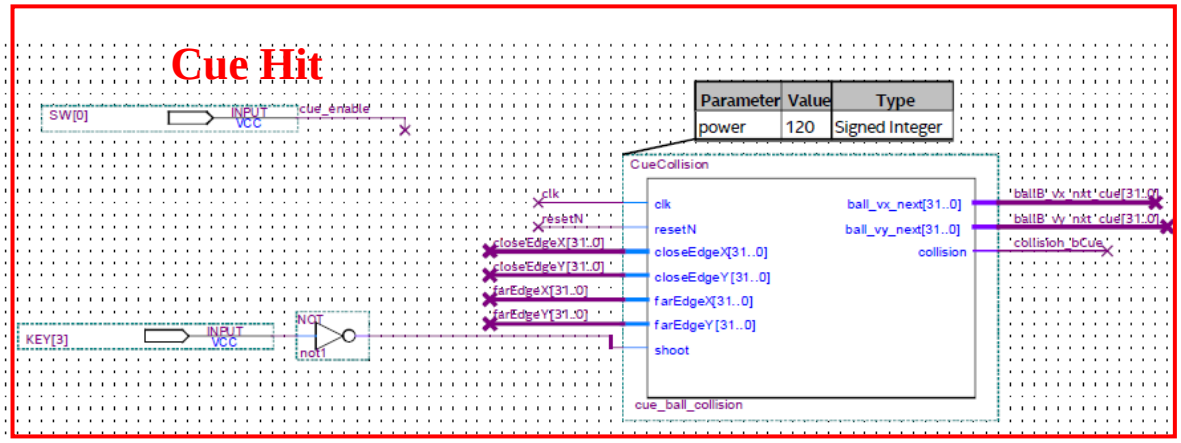
לאחר המימוש העתק את סכמת ההיראכיה העליונה של ה PIPEמ QUARTUS

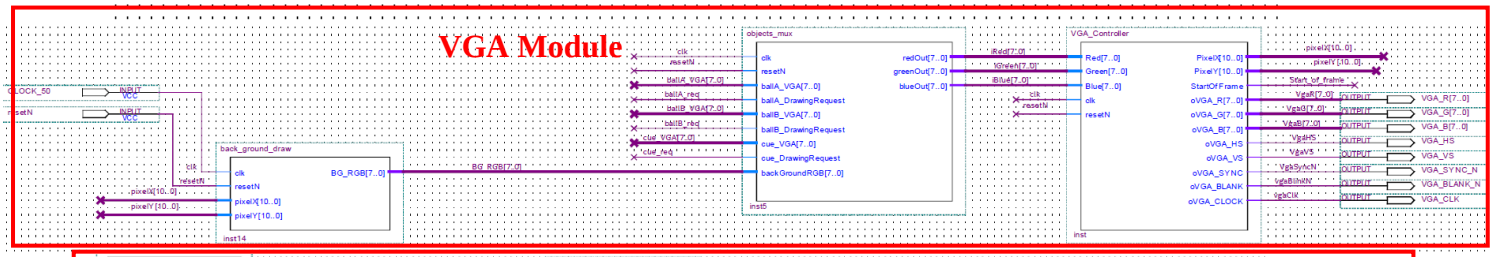


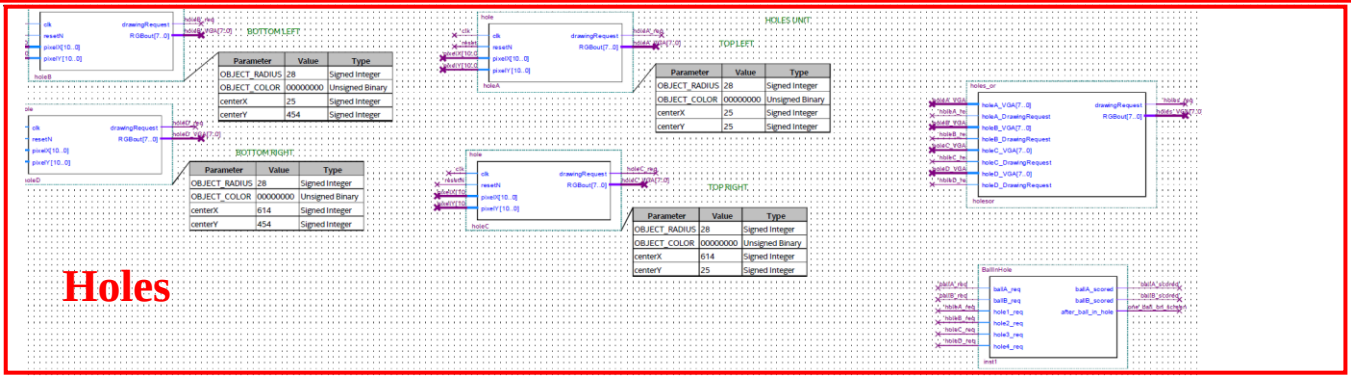



Ball A
INITIAL_X 500 Signed Integer
INITIAL_Y 200 Signed Integer
INITIAL_X_SPEED 0 Signed Integer
INITIAL_Y_SPEED 0 Signed Integer
RADIUS 16 Signed Integer
FRICTION_FACTOR 32 Signed Integer
OBJECT_RADIUS 16 Signed Integer
OBJECT_COLOR 11100000 Unsigned Binary
ball_move
ball_object
next_v_selector
selector_A


Ball Collision
RADIUS_P 16 Signed Integer
RADIUS 16 Signed Integer
BallCollision
collision_detector
BallCollision_AB


Cue Hit
power 120 Signed Integer
SW[0]
KEY[3]
CueCollision
cue_ball_collision


VGA Module
objects_mux
VGA_Controller
back_ground_draw


Holes
HOLES UNIT
BOTTOMLEFT
TOPLEFT
BOTTOMRIGHT
TOPRIGHT
holesor
BallInHole

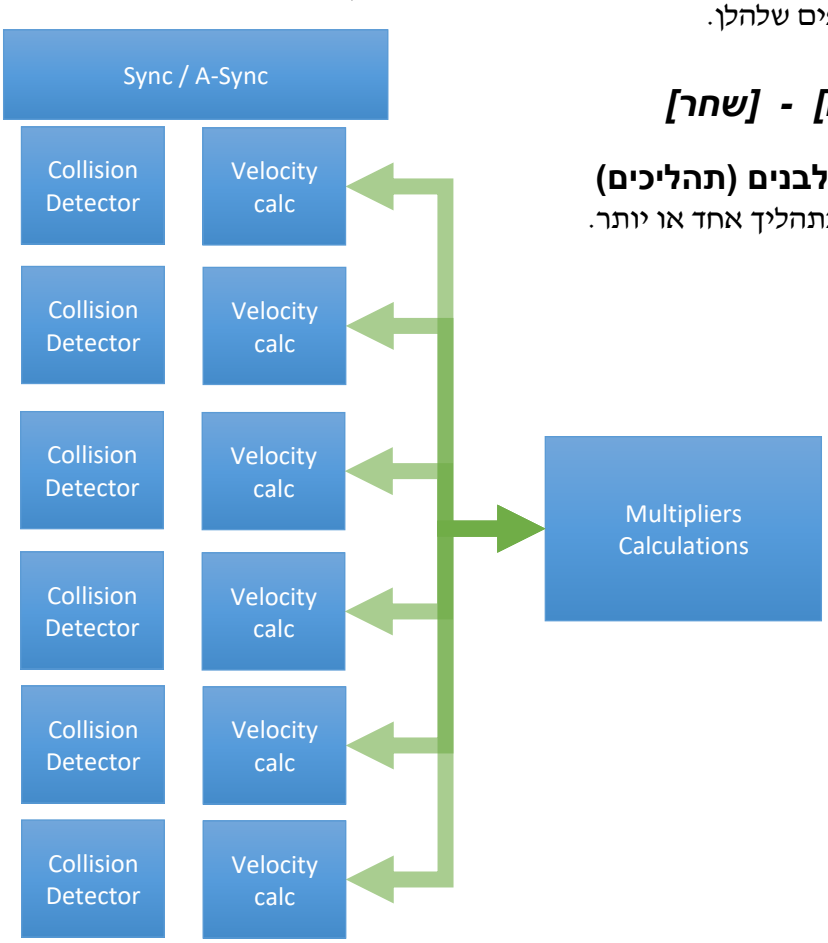
7 תיאור מפורט של שני מודולים -(כמו במצגת)

שימו לב שיש להקפיד לשים מודול אחד לכל סטודנט- (שיהיה תכנון שלו ועליו הוא יסביר)
יש לקחת מודולים מסובכים, רצוי כאלה המכילים המכילים מכונת מצבים , ולא קוד טרוויאלי
לכל מודול יש לבצע את הסעיפים שלהלן.

7.1 [Ball Collisions] - [שחר]

7.1.1 דיאגרמת מלבנים (תהליכים)

תאר את המודול כתהליך אחד או יותר.



התנגשות עלולה לקרות בין כל 2 כדורים, לכן לכל קומבינציה אפשרית של כדורים יש יחידת התנגשויות המורכבת מ2 תהליכים – תהליך סינכרוני וא-סינכרוני.

<u>גילוי התנגשות – תהליך סינכרוני -</u>

גילוי התנגשות קורה באופן סינכרוני בכל מחזור שעון, ומוגדר ע"י 3 תנאים:

- מצב – מצב ניתן לגילוי, התהליך נמצא במצב זה אם לא הייתה התנגשות זהה בטווח של X פריימים אחורה.
- חפיפה – חפיפה של פיקסלים **או** מרחק בין המרכזים - אם המרחק הריבועי בין מרכזי הכדורים קטן מ(2R)^2.

[מיקום הכדורים מחושב בHighPrecision (INT) לעומת המיקום ע"י פיקסל שמוגדר בLOWPRECISION. לכן התנאי המשולב מעניק עוד Robustness לזיהוי ההתנגשות]

- תנועה – 2 הכדורים נעים זה לקראת זה – התנאי הוא שהמרחק בין הכדורים בפריים הבא, קטן מהמרחק הנוכחי.

ברגע שמזוהה התנגשות, יוצא סיגנל המודיע על התנגשות לכדורים עצמם. כאשר הסיגנל הנ"ל יוצא, המהירויות החדשות של הכדורים כבר מעודכן ביציאה מהמודול ע"י התהליך הא-סינכרוני.

התנאי להתנגשות:

```
if (delay==0 && ( (AB_squre_dist_reg <= RADIUS_SQURE) || (ballA_draw_req && ballB_draw_req(( 
                    &&(AB_nxt_squre_dist_reg <= AB_squre_dist_reg) ) begin
            collision <= 1'b1;
```

<u>חישוב פיסיקלי – תהליך א-סינכרוני –</u>

על מנת שההתנגשות תיראה אמיתית יש לחשב את המהירויות המעודכנות אחרי התנגשות בין 2 כדורים.

החישוב הממומש הוא החישוב בנוסחה הבאה:

$$v_1' = v_1 - \frac{\langle v_1 - v_2 , x_1 - x_2 \rangle}{\|x_1 - x_2\|^2}(x_1 - x_2)$$

$$v_2' = v_2 - \frac{\langle v_2 - v_1 , x_2 - x_1 \rangle}{\|x_2 - x_1\|^2}(x_2 - x_1)$$

2 חישובים נוספים שיש לבצע הם המרחק הנוכחי בין הכדורים $\|x_2 - x_1\|^2$ והמרחק הבא בין הכדורים $\left||Xa - Xb + Va - Vb|\right|^2$

מגבלות בFPGA: חישוב בINT32, וחלוקה רק בחזקות של 2.

כאשר יש התנגשות, בתאוריה – המרחק בין 2 כדורים הוא תמיד $(2R)^2$ ולכן בחרנו רדיוס 16. בצורה כזאת אפשר לחלק בקבוע ידוע, שהוא חזקה של 2.

הקוד (כאשר norm =(2R)^2):

```
inner_prod = (((ballA_vy - ballB_vy) * (ballA_y - ballB_y)) + ((ballA_vx - ballB_vx) * (ballA_x - ballB_x)));
inter_x = (ballA_x-ballB_x) * inner_prod;
inter_y = (ballA_y-ballB_y) * inner_prod;
sub_x = (inter_x / norm);
sub_y = (inter_y / norm);
ballA_vx_next = ballA_vx - sub_x;
ballB_vx_next = ballB_vx + sub_x;
ballA_vy_next = ballA_vy - sub_y;
ballB_vy_next = ballB_vy + sub_y;

ABx = ballA_x-ballB_x;
ABy = ballA_y-ballB_y;
ABvx = (ballA_vx-ballB_vx);
ABvy = (ballA_vy-ballB_vy);
AB_squre_dist = ABx*ABx + ABy*ABy;
AB_nxt_squre_dist = (ABx+ABvx)*(ABx+ABvx) + (ABy+ABvy)*(ABx+ABvy);
```

נשים לב שבקוד הנ"ל יש 8 מכפלות של int32.

אם נרצה 4 כדורים, נגיע ל6 התנגשויות אפשריות => 48 מכפלות => 96 בלוקים ל DSP (מתוך 112 בכל ה כרטיס!). כדי לפתור את הנושא הזה הוספנו יחידה מרכזית שמבצעת את החישובים עבור כל היחידות.
החישובים שהוצאו החוצה הם אלו של: inter_x, inter_y, AB_squre_dist, AB_nxt_squre_dist .

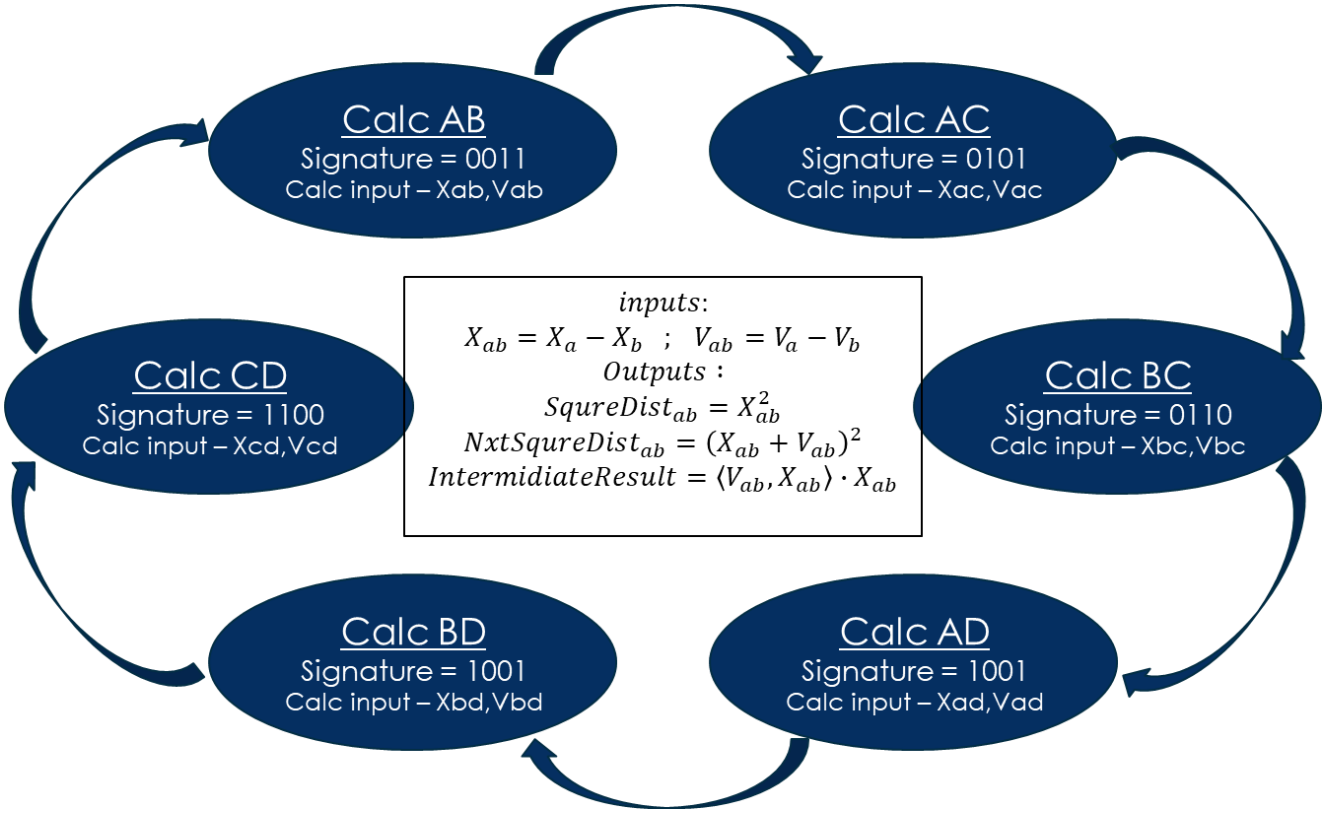
<u>יחידת חישוב מכפלות – תהליך סינכרוני:</u>

התנגשות יכולה לקרות לכל הפחות פעם בפריים.
יש 30 פריימים בשנייה, לעומת מחזורי שעון שיש 50,000 בשנייה!

אין שום סיבה לחשב את כל המכפלות האלה לכל התנגשות אפשרית בכל מחזור שעון.
הפיתרון – צמצום ליחידת חישוב מרכזית, ו"חלוקת" התוצאות באופן מחזורי ליחידות השונות.
זוהי למעשה מכונת מצבים ציקלית בעלת 6 מצבים. בכל מצב נחשב את 8 המכפלות עבור יחידת התנגשות אחרת.
יש הסכמה בין התהליכים על "חתימה" לכל מצב. התהליך יוציא את החתימה של החישוב שכרגע הוא ביצע, וההתנגשות המתאימה תדע (ע"פ החתימה) לקרוא את המידע (מועבר על BUS משותף) ולשמור אותו ברגיסטרים פנימיים.

7.1.2 דיאגרמת מצבים bubble diagram

לתהליכים אותם מימשת בעזרת מכונת מצבים, צייר את דיאגרמת המצבים

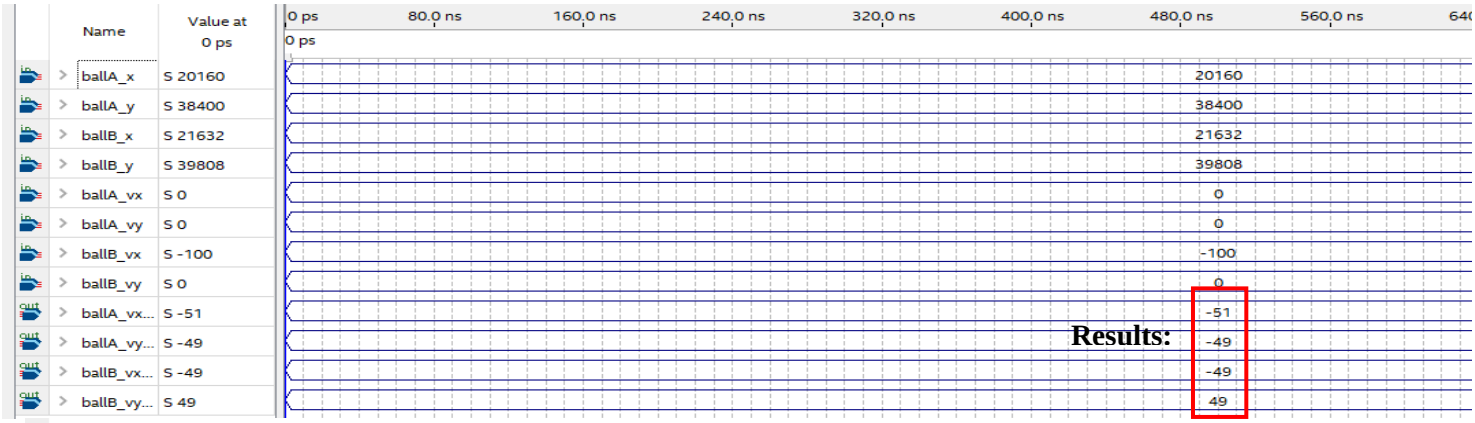


7.1.3 פרט את המצבים העיקריים -

שם המצב	פעילות עיקרית	לאיזה מצב עוברים מהמצב הנוכחי ובאילו תנאים
Calc(Z)(W)	חישוב המכפלות עבור יחידה הבודקת התנגשות בין כדור Z לכדור W . הוצאות החתימה המתאימה.	**עוברים** ל: Calc הבא בסדר עם עליית שעון

7.1.4 מסך(י) סימולציה

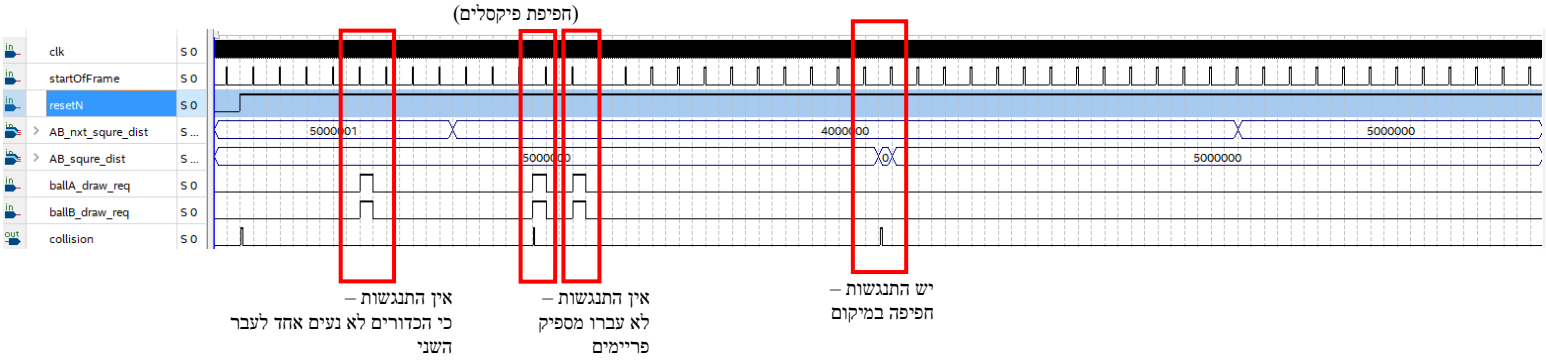
סימולציה 1- חישוב מתמטי של התנגשות (לפני ייצוא המכפלות), ווידוא תוצאות קרובות ע"י סקריפט פייתון



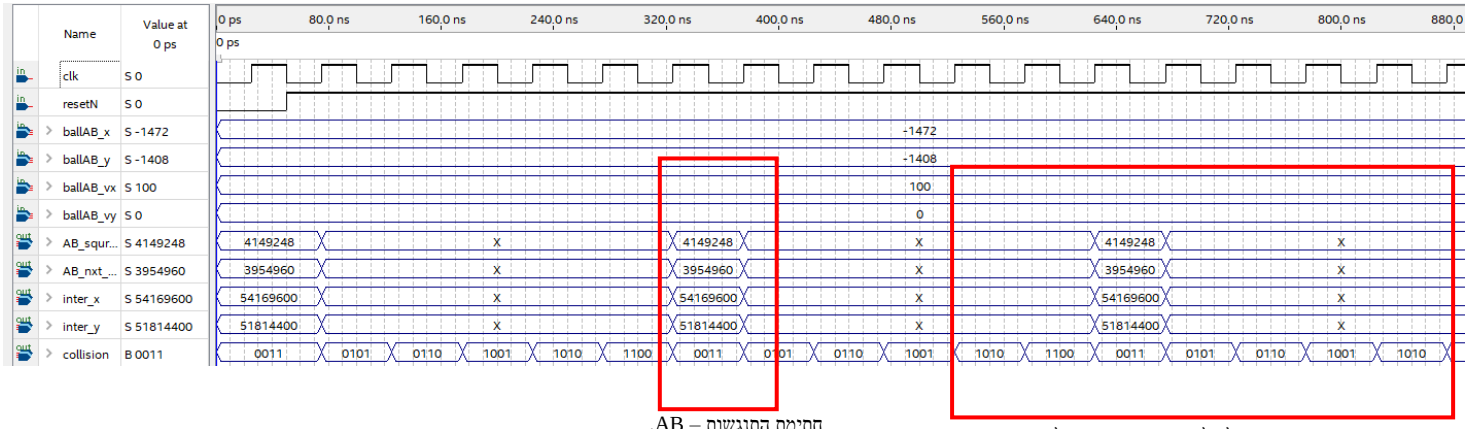

```
import numpy as np
Xa = np.array([315,600])
Xb = np.array([338,622])
Va = np.array([0,0])
Vb = np.array([-100,0])
Xa_HP = Xa*64;
Xb_HP = Xb*64;

Va_nxt = Va-np.dot(Va-Vb,Xa-Xb)*(Xa-Xb)/np.dot(Xa-Xb,Xa-Xb)
Vb_nxt = Vb-np.dot(Vb-Va,Xb-Xa)*(Xb-Xa)/np.dot(Xb-Xa,Xb-Xa)
print(' Xa_HP = ',Xa_HP,'\n','Xb_HP = ',Xb_HP,'\n','Va = ',Va,'\n','Vb = ',Vb)
print('EXPECTED RESULT :\n','Va\' = ',Va_nxt,'\n','Vb\' = ',Vb_nxt)
```

```
In [25]: runfile('C:/Users/Shahar/.spyde
Xa_HP =  [20160 38400]
Xb_HP =  [21632 39808]
Va =  [0 0]
Vb =  [-100    0]
EXPECTED RESULT :
Va' =  [-52.22112537 -49.95064166]
Vb' =  [-47.77887463  49.95064166]
```

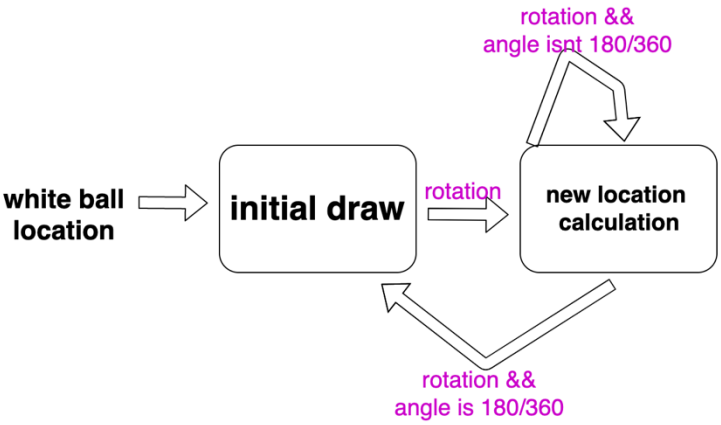
סימולציה 2- זיהוי התנגשות



סימולציה 3 – חישוב המכפלות באופן ציקלי והחלפת "חתימות"



7.2 [Cue Move/Draw] - [צחי]



7.2.1 דיאגרמת מלבנים

<u>אפיון והצגה של המקל</u>

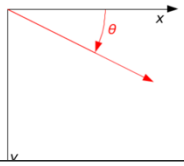
נקבל כקלט את המיקום של מרכז הכדור הלבן על הלוח. נאתחל משתנה שישמור על הזווית של המקל.
שתי הנקודות יאותחלו באופן הבא :

הנקודה הקרובה לכדור הלבן תקבל את קאורדינטת הy של הכדור הלבן ואת קאורדינטת הx פחות רדיוס הכדור
הנקודה הרחוקה תקבל את קאורדינטת הy ואת קאורדינטת הx פחות הרדיוס פחות אורך המקל.

<u>חישוב הנקודות לאחר כל שינוי במקל</u>

קיבענו את הזווית להיות 5 .כך למעשה קיבלנו רזולציה של 62 כיוונים אופציונאלים למיקום המקל .

בהינתן הנקודה ההתחלתית ,כל בקשה לשינוי במיקום המקל תתבצע באמצעות כפל במטריצת סיבוב . על פי הנוסחה הבאה :



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

```
closeEdgeX_tmp <=  (( (closeEdgeX_tmp*COS_5) - (closeEdgeY_tmp*SIN_5) ) / ANGLE_MULTIPLIER);
closeEdgeY_tmp <= ( ( (closeEdgeX_tmp*SIN_5) + (closeEdgeY_tmp*COS_5) ) / ANGLE_MULTIPLIER);
farEdgeX_tmp <=  ((( (farEdgeX_tmp*COS_5) - (farEdgeY_tmp*SIN_5) ) / ANGLE_MULTIPLIER));
farEdgeY_tmp <=  (( (farEdgeX_tmp*SIN_5) + (farEdgeY_tmp*COS_5) ) / ANGLE_MULTIPLIER);
```

כדי להתגבר על בעיה של חישוב של שבר כתוצאה מהמכפלה ב)5(cos ו)5(sin , נעזרנו בפרמטר multiplier=2^14 אשר מכפיל את ה sin ואת הcos ולבסוף חילקנו באותו multiplier.

<u>תיקון שגיאות נגררות</u>

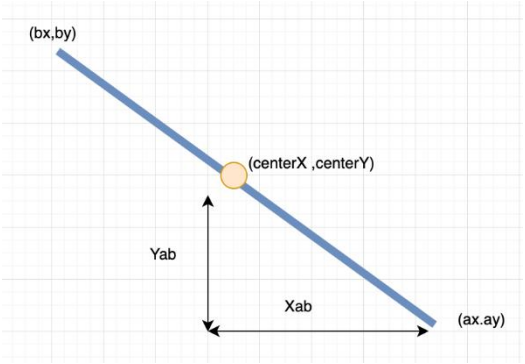
מכיוון שהחישוב הוא דיסקרטי ולא רציף , עשויה להצטבר שגיאה נגררת לאחר כמות של שינויים במקל.
התגברנו על הבעיה הזאת בעזרת דרישה לאתחול המקל כפי שאתחלנו אותו בצורה הראשונית כאשר המקל נמצא בזווית 180 או 360 .
אנחנו יודעים להפעיל את התנאי לאיתחול בעזרת הגדרה של משתנה אשר שומר בכל איטרציה את הזווית הנוכחית של המקל.

<u>הצגת המקל על המסך</u>

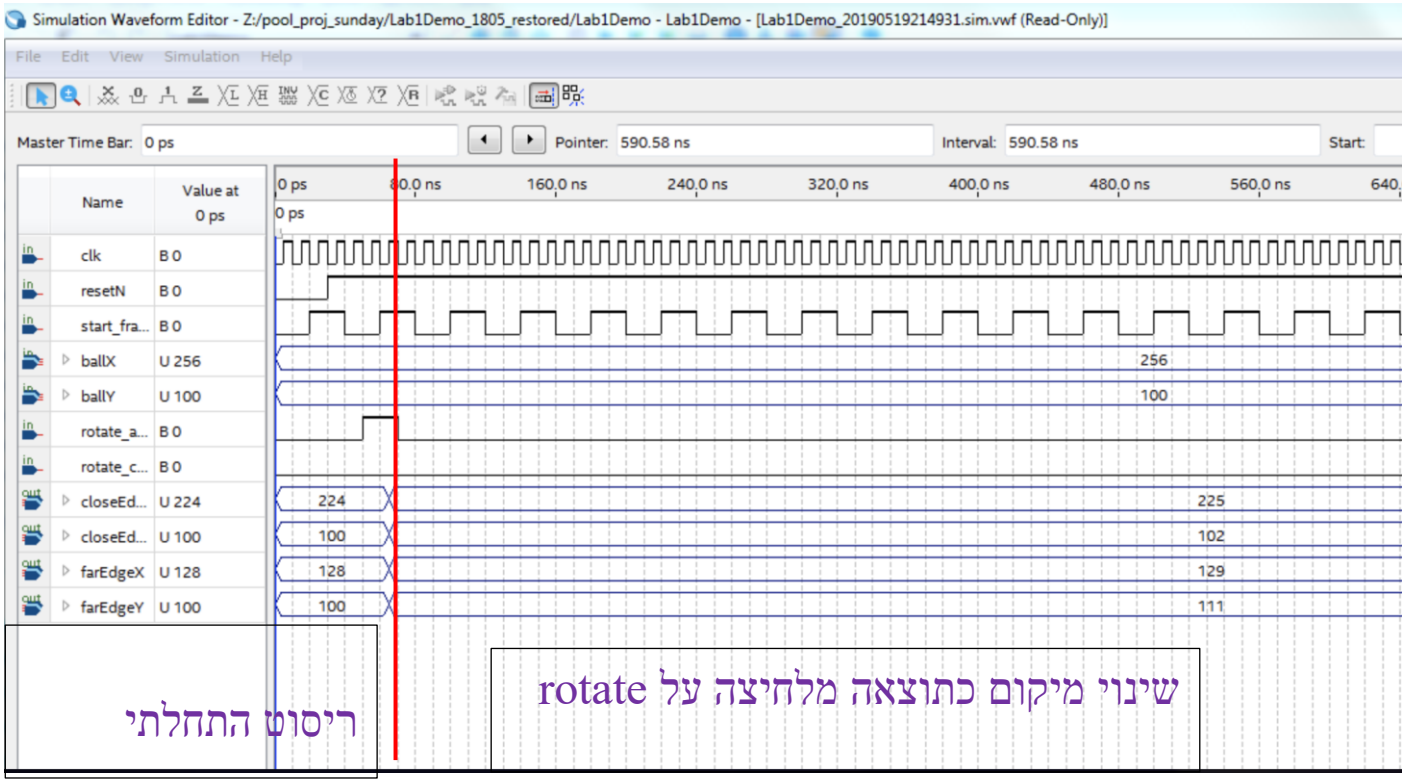
ציור המקל על המסך מתבצע בעזרת 2 נוסחאות.
הראשונה היא חישוב מרחק נקודה מקו המוגדר ע"י שתי נקודות

$$dist^2 = \frac{\left|(y_2 - y_1)x_{pixel} - (x_2 - x_1)y_{pixel} + x_2y_1 - y_2x_1\right|^2}{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

בחרנו את רוחב המקל הרצוי וכל נקודה שקטנה מ ^{2}(רוחב המקל) רלבנטית לנו.
כעת נקבל קו ישר ברוחב הדרוש על פני כל המסך. נרצה 'לחתוך' אותו לאורך המקל הרצוי.
נעשה זאת באמצעות תנאי שייצבע רק נקודות שמרחקן ממרכז המקל $(\frac{X_a+X_b}{2},\frac{Y_a+Y_b}{2})$ קטן מ$^2(\frac{\text{אורך המקל}}{2})$.



מסך סימולציה

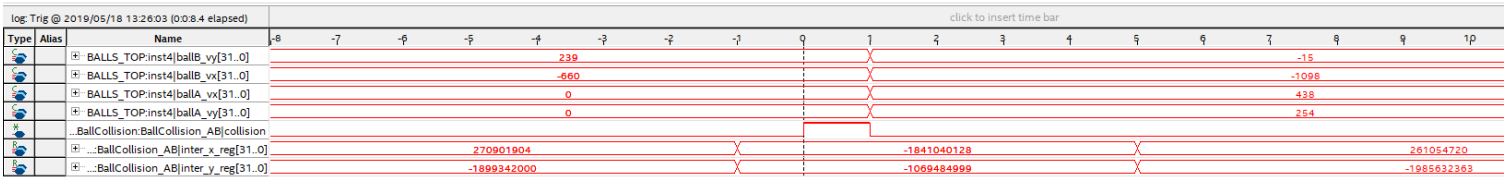


8 (S.T.) Signal Tap

אם השתמשת ב .S.T לזהות באג בחומרה, צרף מסך של ה .S.T בו זיהית את הבאג. הסבר מה היה הבאג, כיצד זיהית אותו וכיצד תקנת אותו.

אם לא השתמשת ב .S.T לזיהוי באג בחומרה, צרף מסך של ה .S.T בו מתבצעת פעולה סינכרונית והסבר אותה.

במהלך העבודה, זיהינו מידי פעם עלייה חדה במהירות הכדורים אחרי התנגשות.
תופעה זאת לא אמורה לקרות שכן קיים שימור תנע, ומהירות של כדור לא יכולה להיות גדולה יותר מסך גדלי המהירויות לפני התנגשות.
בעזרת SIGNAL TAP הצלחנו לשחזר את התקלה וזיהינו overflow באחד החישובים במודול ההתנגשות.



בציור -^ מסתכלים על המהירויות החדשות של כדורים A וB אחרי התנגשות (זהו הטריגר). ניתן לראות שהמהירויות החדשות נוגדות את שימור התנע. הבעיה כאמור – היא בחישוב של הרגיסטר inter_x , שם יש overflow.

המשוואה הבעייתית:

$$inter_x =< V_{ab}, X_{ab} >\cdot X_{ab_x}$$

לאחר מכן יש חלוקה של inter_x בגורם הנירמול.

מכיוון שהמיקום (X) מחושב בhigh precision , וכל החישובים נשמרים בInt32– עלול להתקבל overflow במקרי קיצון.

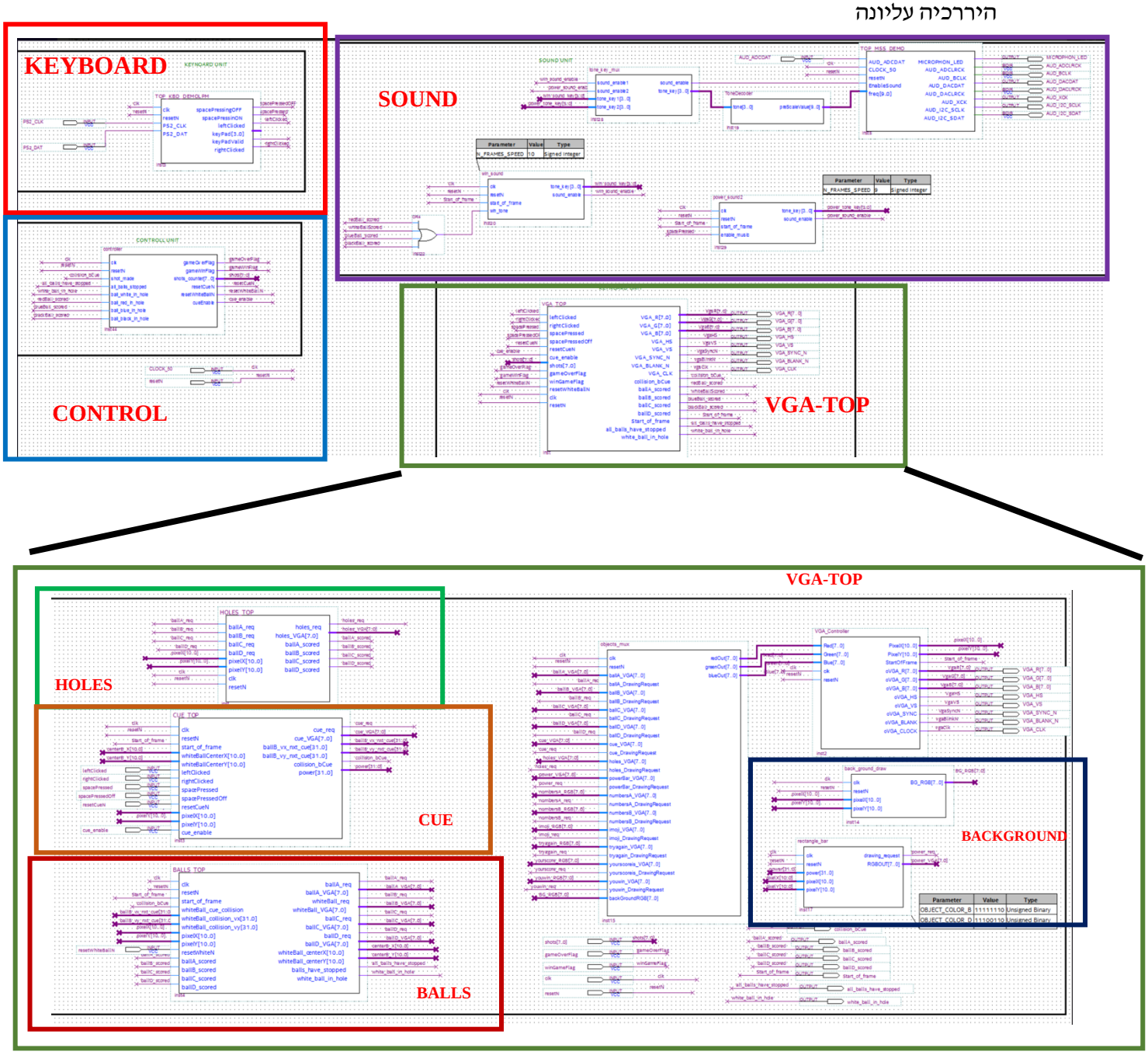
זה בדיוק הtradeoff בין דיוק גבוה לבין מספר הסיביות הנדרש לשמירה (וסיכון overflow). אחת הצרות כשאין FP...

פתרנו את הבעיה על ידי חלוקה של המכפלה הפנימית ב4 לפני כפל בXab (והכפלה של גורם הנרמול ב4).

9 מימוש ההיראכיה עליונה

9.1 שרטוט

היררכיה עליונה



Flow Summary

<<Filter>>

Flow Status	Successful - Sun May 19 22:12:55 2019
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Standard Edition
Revision Name	Lab1Demo
Top-level Entity Name	TOP_VGA_DEMO_WITH_MSS
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	4,329 / 41,910 (10 %)
Total registers	2499
Total pins	42 / 499 (8 %)
Total virtual pins	0
Total block memory bits	128 / 5,662,720 (< 1 %)
Total DSP Blocks	90 / 112 (80 %)
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 %)

זמן הקומפילציה הוא בערך 5 דקות
צרכנו יחסית הרבה יחידות dsp בשל החישוב הרב שנדרש להתנגשות הכדורים

10 סיכום ומסקנות

<u>ניהול עבודה נכון</u> – אפשר לבזבז המון זמן על פרוייקט כזה, זמן שאין לנו (קורסים אחרים, יומיים עבודה בשבוע ועוד) ולכן חייבים להשקיע זמן בתכנון - תכנון מראש של סדר כתיבת המודולים, תיעדוף עבודה, ושמירה על מודולריות זה לא סתם קלישאה. בחלקים שביצענו את זה בפרוייקט ראינו שעבדנו ביעילות רבה יותר, לעומת בחלקים בהם רצנו קדימה במימוש.
לדוגמא - מימוש כל ההתנגשויות והכדורים במסך. לכאורה החלק הכי מורכב בפרוייקט והתחלנו דווקא ממנו – הזמן קומפילציה היה נמוך וכך יכולנו לבצע תיקונים מהירים ולבדוק את הרכיב באופן יסודי. תחילה ע"י 2 כדורים שנעים על המסך ויחידת התנגשות אחת. בסוף - לבצע את ההתאמה ללוח עם 4 כדורים הייתה מאוד קלה ונוחה כי הקוד היה מודולרי ובעיקר נכון.

<u>פתירת באגים</u> - חווינו את ההרגשה של לפתור באגים בקוד לוקחת זמן רב יותר מכתיבת הקוד עצמו. גם כאן , סכמת פתרון הבעיות הייתה מאוד יעילה ברגע שהשתמשנו בה – הרצת סימולציה על המודול, עבודה עם signal tap וכו'...

<u>הצלחה מעבר לדרישות</u> – מראש ידענו שהמתרה שלנו היא לא רק לעמוד בדרישות המינימום, אלא להציג מוצר יותר מתוחכם (בזמן מאוד מוגבל). עמדנו בכל דרישות הפרוייקט והוספנו גם מעבר – ישנם 4 כדורים על הלוח שמבצעים התנגשות אלסטית ויודעים לנוע בתנועה עם חיכוך .יש מקל שמסתובב ומכה בצורה טובה, צלילים בכל פעולה של המשתמש , ומכונת מצבים ששולטת בשלבי המשחק .

הפרוייקט הזה (ביליארד) טומן בתוכו גם אתגרי מימוש מתמטיים
- מימוש משוואות מתמטיות מורכבות בfixed point.
- מציאת פתרונות ותחליפים לפעולת החילוק.
- ניהול משאבי חומרה מוגבלים, התמודדות עם overflow ועוד.

הצלחנו להתגבר על האתגרים הללו ולמדנו המון על כמה זה לא טריוואלי לממש חישוב פשוט בC.

11 המלצות לשנה הבאה

<u>לחיוב :</u>

-הפרויקט היה מאוד מאתגר, עם זאת ההכנה אליו הייתה מדורגת טוב.

-עצמאות של הסטודנטים בתקופת העבודה על הפרוייקט – בלי מטלות ביניים ובלי משימות נוספות.

-אווירה כללית חיובית וטובה בקרב המדריכים והסגל.

<u>רעיונות לשיפור</u>

-אולי שווה לתת לסטודנטים שיירצו לבנות משחק לגמרי שלהם ולא להגדיר להם פורמט מסויים.

-לתת לסטודנטים לחתום גם על מקלדת , ייקל מאוד על העומס במעבדה ועל הסטודנטים.

12 נספחים: דפי נתונים, דפי מידע שונים בהם השתמשת.

Equations and Theory:
https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line#Line_defined_by_two_points
https://en.wikipedia.org/wiki/Rotation_matrix#In_two_dimensions
https://en.wikipedia.org/wiki/Elastic_collision#Two-dimensional_collision_with_two_moving_objects
https://www.chipverify.com/verilog

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

דוגמה

מלא את הטופס