**הטכניון - מכון טכנולוגי לישראל**

**הפקולטה להנדסת חשמל**



מעבדה 1

MSS

שאלות ודו"ח הכנה

DE-10

גרסה 1.45

קיץ 2018

מחברים: אלכס גרישנפון

|  |  |
| --- | --- |
| תאריך הגשת דו"ח ההכנה |  |
| שם המדריך |  |

|  |  |  |
| --- | --- | --- |
| סטודנט | שם פרטי | שם משפחה |
| 1 | ברק | זן |
| 2 | בועז | טייטלר |

תוכן עניינים

[1 מימוש מונה כתובות 2](#_Toc521424922)

[2 מימוש מיישר חצי גל, גל שלם 3](#_Toc521424923)

[3 שימוש בטבלאות למימוש פונקציה מורכבת 7](#_Toc521424924)

[3.1 שימוש בטבלה ב VHDL](#_Toc521424925) 6

[4 DPR 8](#_Toc521424926)

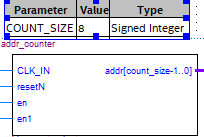
הנחיות

* קובץ זה הוא גם התבנית לדו"ח המכין, יש לשמור ב PDF ולהגיש במודל.
* במעבדה זו נבחן מימושם שונים של כניסות ויציאות אנלוגיות ועיבוד אותות, בחלקים מהם נשתמש בפרויקט לצורך יצירת צלילים.

# מימוש מונה כתובות

* תכנן מונה לפי התיאור הבא: הרכיב ***addr\_counter*** הוא מונה 8 ביט שמספק את 256 הכתובות. המונה יהיה סינכרוני ויעבוד לפי עלית שעון , יספור אך ורק כאשר גם כאשר **en** וגם **en1**מקבלים .’1’ כרגיל, resetN הינה כניסת איפוס   
  א-סינכרונית הפעילה בנמוך. השתמש בקובץ addr\_counter.vhd שנמצא בmoodle כחלק מהפרויקט.
* שימו לב: אורך וקטור היציאה (המוגדר כעת ל-8 ביטים) הינו גודל גנרי וניתן לשינוי. בתכן שלכם עליכם להתייחס לגודל גנרי כללי כך שגם אם נשנה את אורך וקטור היציאה, ל-16 ביטים למשל, המונה יעבוד כהלכה. לשם כך, ניתן להשתמש בכך שכאשר וקטור בגודל מסויים מנסה לגדול מעבר ערך המקסימלי שלו (ב-8 ביטים- 255) אזי מתבצע wraparound והוקטור חוזר לערך 0. כך למשל, כאשר המונה יהיה "11111111" וננסה להוסיף לו 1, נקבל "00000000".

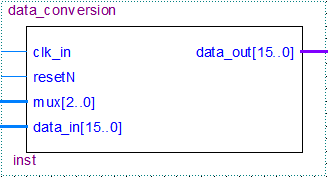
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| count[COUNT\_SIZE..0] | en1 | en | resetN | CLOCK\_50 |
| 0000 | x | x | 0 | **x** |
| count | 1 | 0 | 1 | **↑** |
| count+1 | 1 | 1 | 1 | **↑** |
| count | 0 | 1 | 1 | **↑** |



# מימוש מיישר חצי גל, גל שלם

בניסויים האנלוגיים, ראינו ונראה שימושים למגברי שרת במגברים הופכי מופע, מעגלי יישור חד דרכי ודו דרכי, מעגלי קטימה ועוד.

כתוב ב-VHDL יחידה (Entity+Architecture) אשר מממשת באופן סיפרתי "מעבד אותות ספרתי". שמור את הקובץ בשם *data\_conversion.vhd* .



היחידה מקבלת 16 ביט בכניסה ומוציאה 16 ביט לאחר עיבוד. לא לשכוח עובדים ב2s complement

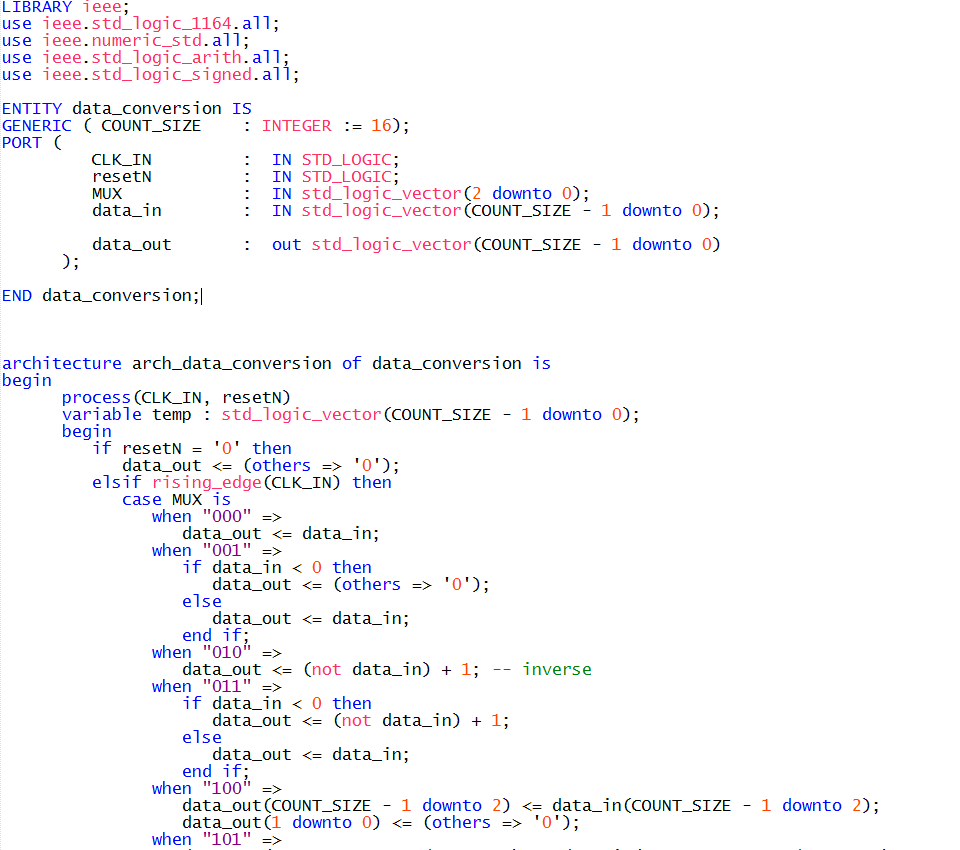
המערכת תהיה סינכרונית – ותוציא את המידע אחרי השהיה של שעון אחד, שים לב לשימוש נכון ב SIGNALS וב VARIABLES

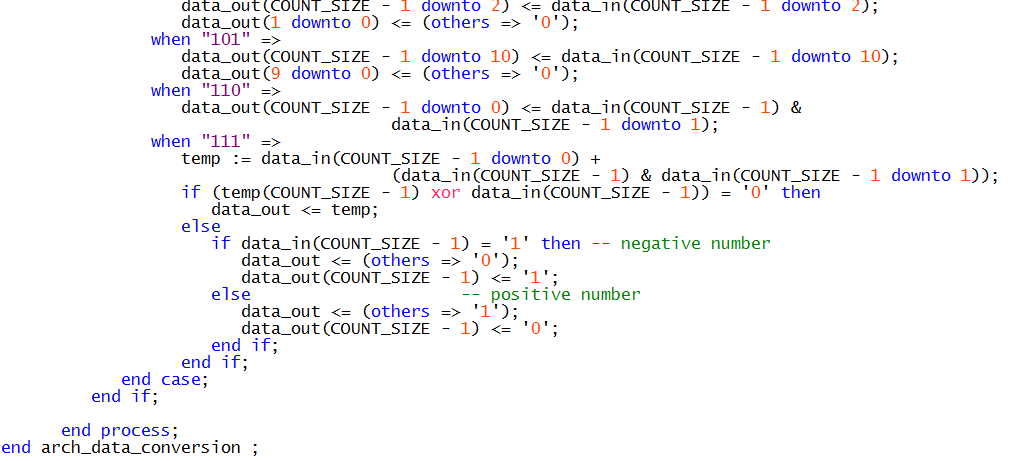
למיישר כניסת פיקוד: mux[2..0] בעלת מספר מצבים

הכניסות והיציאות למיישר תהינה אותות בינריים של 16 ביט בייצוגsigned integer .

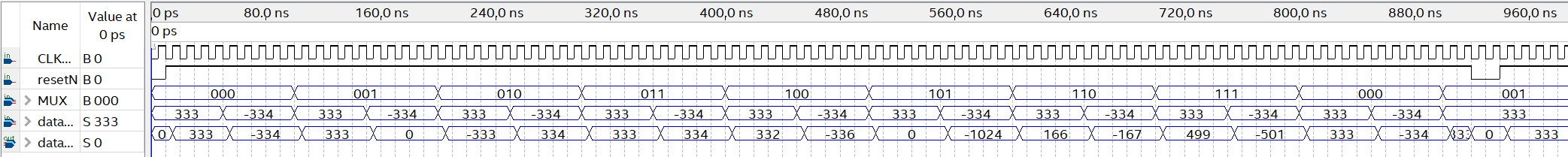
כל התכן חייב להיות סינכרוני ובעל כניסת איפוס אסינכרונית הפעילה בנמוך.

|  |  |  |  |
| --- | --- | --- | --- |
| תאור | שרטוט | כניסת  SW4- SW2  הMUX | description |
| אות כניסה , מהמחולל  אמפליטודה +/- 400mV  תדר 1000 Hz |  |  |  |
| אות יציאה זהה לאות כניסה  Bypass |  | 000 | Bypass  מעביר את האות ללא שינויים |
| אות יציאה. מעביר חלק חיובי  Half wave |  | 001 | - Half Wave מישר חצי גל. רק חלק חיובי, כל ערך שלילי == 0 |
| אות יציאה. הופך מופע invert |  | 010 | Invert  הופך את האות. (מכפיל ב -1) |
| אות יציאה**.** מעביר חלק חיובי והופך שליליfull wave |  | 011 | - Full Wave מיישר גל מלא. חלק'ם שלילים יהפכו להיות חיוביים לפי 2s complement |
| Quantization-Two |  | 100 | -Quantization-Two - מאפסת את שתי הסיביות התחתונות של המספר כך שהרזולוציה קטנה יותר ( המספר 0x4777 יהפוך ל 0x4774 ) |
| Quantization-Eleven. |  | 101 | Quantization-ELEVEN - מאפסת את 10 הסיביות התחתונות של המספר.  ( המספר 0x4777 יהפוך ל 0x4400 ) |
| HALF - מחלקת בשנים | Y = x /2 | 110 | HALF- מחלקת בשנים את המספר כך שהתחום הדינמי קטן בחצי .  ( המספר 0x4300 יהפוך ל 0x2180 ) |
| OneFiftyPrecent - מגדיל את המספר ל 150% מערכו | Y = x \* 3/2 | 111 | OneFiftyPrecent - מגדיל את המספר ל 150% מערכו שהתחום הדינמי גדל פי אחד וחצי במידת הצורך המעגל נכנס ל"רוויה".  ( המספר 0x3480 יהפוך ל 0x4EC0 |





בצע סימולציה והראה שכל אחד מהמצבים עובד. צרף את תוצאות הסימולציה של כל אחד מהמצבים לדו"ח ההכנה (יש להביא את הקובץ למעבדה על מנת לבדוק שהתכן עובד על גבי הפלטפורמה).

שימו לב שכל החישובים נעשים ב ב2s complement!

|  |
| --- |
|  |

**על הקוד שכתבתם יש לבצע פעולת ARCHIVE ב QUARTUS (כמתואר בפרק 16 של** [**quartus 17 cook book**](https://moodle.technion.ac.il/mod/resource/view.php?id=616928) **במודל).**

**את הקובץ המכווץ שתקבלו מפעולה זו יש להעלות במודל ל**



# שימוש בטבלאות למימוש פונקציה מורכבת

לפעמים הדרך הפשוטה לממש פונקציה מסובכת היא על ידי טבלה של קבועים שמכינים מראש

בחומר העזר במודל נתון קובץ בשם ***SinTable.vhd*** בו מופיעה טבלה לאות סינוס הממירה את מספר השורה לערך של סינוס הזווית.

דוגמה לבנית טבלה:

כיוון שאיןfloating point ב VHDL כל הערכים מנורמלים למספרים שלמים. סינוס הזווית מבוטאת במספר בין 0x0000ובין ,0x3E80משורה ראשונה עד שורה 256, והמוצא מופיע ב-SIGNED INT

החישובים הדרושים לבניית האות נעשים בקובץ אקסל, ***Sinetable.XLSX*** , הנתון גם. הטבלה בקובץ ה- VHDL נוצרה על ידי העתקת העמודה המסומנת בצהוב, הנותנת את ערכי אות הסינוס ב-*hexadecimal* מטבלת האקסל. ראה את הנוסחאות המשמשות ליצירת הטכסט בקובץ האקסל הנתון.

## שימוש בטבלה ב VHDL

במודל מופיע מימוש מלא של המעגל מלבד הקוד של מונה הכתובת אותו יש עליכם לכתוב , כמתואר בסעיף 1, ולצרף אותו לתכן המוכן שקיבלתם.

במונה זה תשתמשו גם בניסוי שתעשו במעבדה.

עליך לכתוב ב-VHDL יחידה (Entity Architecture) בשם *SquareTable.vhd* אשר מממשת באופן סיפרתי פונקציה של ערך חזקה שניה. הפונקציה מקבלת ערך**8 ביט** Signed int שלX שערכו בין -1.0 ל- 1.0. ומוציאה את הערך של , ( 16 ביט) , שערכו בין . -1.0 ל- 1.0.

בנה את הפונקציה המתאימה בקובץ אקסל, לפי דוגמת הסינוס. מה יהיו הערכים המנורמלים עבור קובץ ה-VHDL ? רשום להלן את 20 הערכים הראשונים. (המתחילים ב X= -1.0)

תשובה תשובה:

הכניסות יהיו בין -128 ל 128 (שולחים כתובת בין 0 ל 125), והיציאות בין 0 ל 16384

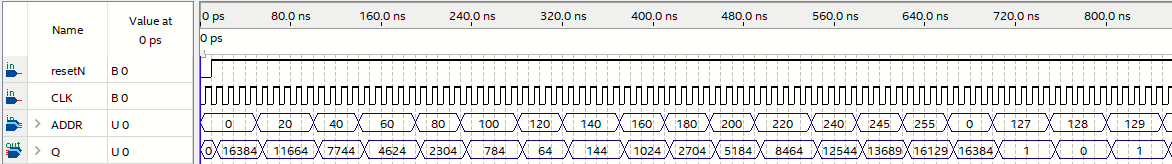
X"4000",X"3F01",X"3E04",X"3D09",X"3C10",X"3B19",X"3A24",X"3931",

X"3840",X"3751",

התאם את הקוד שלך בקובץ *SquareTable .vhd* לפי דוגמת הסינוס על ידי בחירת העמודה המתאימה והעתקתה לקובץ ה- VHDL.

סמלט אותו על ידי הכנסת ערכים בכניסה (למשל כל ערך עשירי) והוסף את הסימולציה לדו"ח.

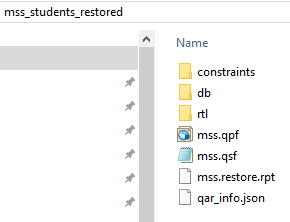
|  |
| --- |
|  |

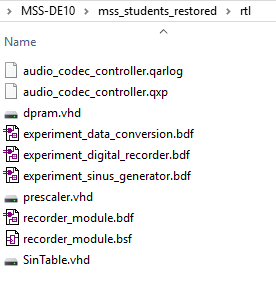


שמור את הקבצים והבא אותם למעבדה על מנת לבדוק שהתכן עובד על גבי הפלטפורמה.

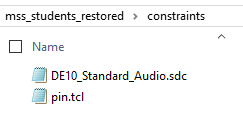
# DPR

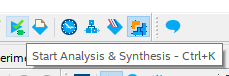
השתמשו במודול recorder\_module.bdf אשר נמצא בmoodle בקובץ mss\_students.qar:

* פתח מהמודל את הפרויקט mss\_students.qarובחר לאן לחלץ את קבצי הפרויקט.



* קובץ הפינים בשם pin.tcl נמצא ב:

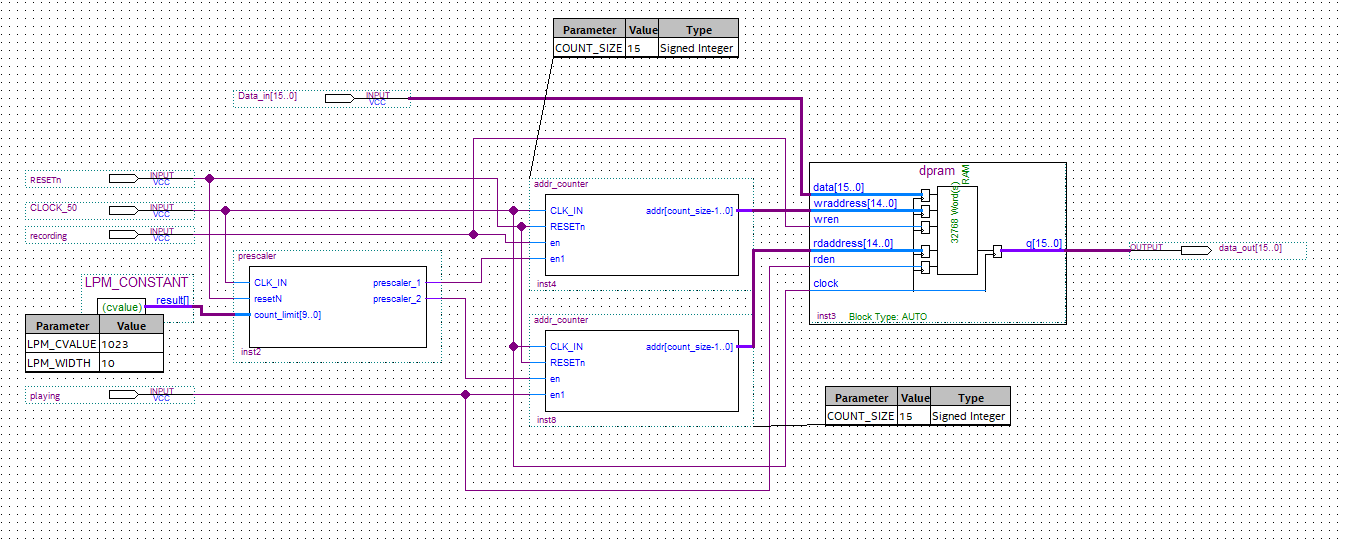


* העתק לתיקיה rtl את הקובץaddr\_counter.vhd שתכננת בעבודת הכנה.
* עבור לתפריט Project Navigator – Files והגדר את הקובץ recorder\_module.bdf הנמצא בתיקיה rtl כ-TOP-Level Entity - (CTRL-SHIFT-V)
* קמפל – )מספיק להריץ רק סינטזה(
* 

דרוש לבנות מעגל השהיה, שייקח דגימות של אות ויוציא אותן מושהה וללא שינוי צורה ביציאה :

כדי ליצור השהייה מקסימאלית בין אות הכניסה לבין אות היציאה. לרשותך התקן זיכרון מסוג *Dual Port RAM* בעל גודל קבוע של N מילים.

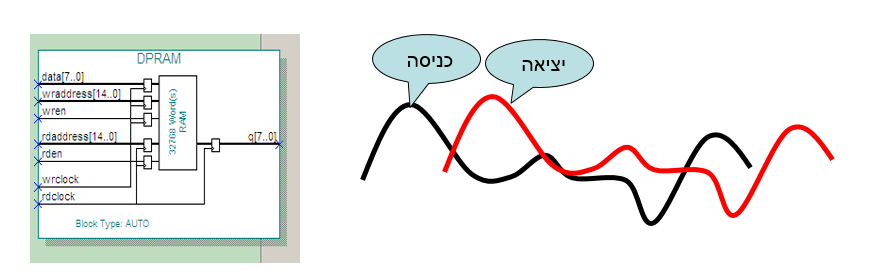
למעגל שתי כניסות מפסקים READ ENABLEו- WRITE ENABLE שכאשר שניהם הם ב-‘1’ המונים מתקדמים (מאופשרים).



שים לב שליחידהPRESCALE יש שתי יציאות בהפרש של שעון, הדבר נועד למנוע מצב של קריאה וכתיבה בו זמנית מהזיכרון, דבר שיהיה בעייתי אם שתי הכתובות הן זהות.

שנה את ערך ה- LPM\_CVALUE של prescaler ל63 .

סמלץ את המעגל, וודא שהוא מתפקד כראוי.



באיזה אופן תכתוב לזיכרון ותקרא ממנו על מנת לקבל השהייה מכסימלית.

תשובה: על מנת לקבל השהייה מקסימלית ננצל את התכונה של הזכרון לזכור עד N מילים. תחילה נכתוב את ה-N מילים הראשונות, ואז ננסה לסנכרן בין קריאה מהזכרון לבין כתיבת המילה הבאה מאות הכניסה לתוך הזכרון. בסופו של דבר נקבל השהייה של N מילים או N עליות שעון.

במודל מופיע מימוש מלא של המעגל, עליך רק לבצע לו סימולציה חכמה שתבדוק את מצבי המערכת

סמלט את המעגל, הכנס מידע משתנה בכניסה שנה את כניסות ה ENABLE וודא שהמעגל מבצע את פעולת ההקלטה. הוסף את תוצאות הסימולציה הבוחנות את כל המקרים ומראות שהמעגל אכן מתפקד כראוי:

|  |
| --- |
|  |

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

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