**הטכניון - מכון טכנולוגי לישראל**

**הפקולטה להנדסת חשמל**



מעבדה 1

ניסוי 1VHDL

דוח הכנה למעבדה

גרסה 1.71

קיץ 2018

עורכים: דודי בר-און, אברהם קפלן , ליאת שורץ

על פי חוברות של עמוס זסלבסקי

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

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

תוכן עניינים

[1 תרגיל תכנון MUX בשיטות שונות 2](#_Toc508779233)

[1.1 מימוש Selected Assignment MUX 2](#_Toc508779234)

[1.2 מימוש MUX Conditiona Assignment 2](#_Toc508779235)

[2 תרגיל תכנון MUX הירארכי 3](#_Toc508779236)

[3 מחלק תדר 3](#_Toc508779237)

[4 מחלק תדר- כניסת TURBO 4](#_Toc508779238)

[5 מונה עולה פשוט 5](#_Toc508779239)

[6 תצוגת 7Segment עם הדלקה מלאה וכיבוי 6](#_Toc508779240)

הערות:

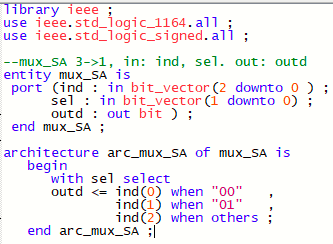
1. בכתיבת הקוד חובה להשתמש בשמות ה- entities, הכניסות והיציאות המופיעים בהגדרת התרגילים.
2. יש לתת שמות קבצים ותיקיות באנגלית בלבד וללא רווחים
3. שם הקובץ צריך להיות כשם ה-Entity
4. בכל פעם יש להגדיר את הקובץ כ- TOP
5. **יש לקמפל את הקוד** בקווארטוס, לראות שאין שגיאות סינטקס
6. בסוף התהליך יש להעתיק את הקוד בצורה **קריאה** מהNOTEPAD++ לקובץ התשובות

# תרגיל תכנון MUX בשיטות שונות

## מימוש Selected Assignment\_MUX

כתוב קוד בשם mux\_SA. המערכת הינה Multiplexer בעל ממדים 3=>1. כניסות המידע ind, כניסות בחירה - sel ויציאת המידע – outd.

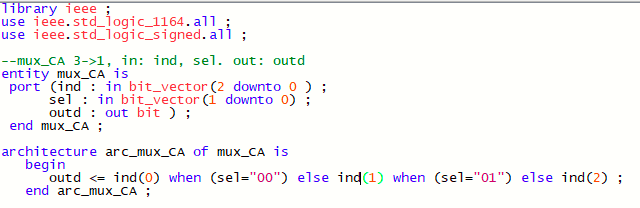
כתוב את הקוד שמתאר את הרכיב באמצעות **התניית** Selected Assignment .



## מימוש MUX Conditional Assignment

כתוב קוד בשם mux\_CA. המערכת הינה Multiplexer בעל ממדים 3=>1. כניסות המידע ind, כניסות בחירה - sel ויציאת המידע – outd.

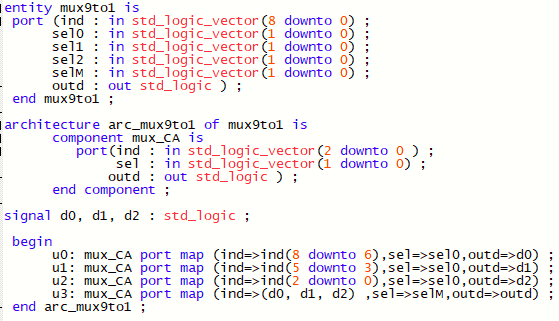
כתוב את הקוד שמתאר את הרכיב באמצעות **התניית** Conditional Assignment.



# תרגיל תכנון MUX הירארכי

כתוב קוד בשם: mux9to1. שיממש Multiplexer בעל 9 כניסות מידע ind, 4 כניסות בחירה selויציאה אחת outd. המערכת מורכבת מ 4 רכיביMultiplexer מהתרגיל הקודם בעלי ממדים 3>1, אותות הכניסות הן מסוג std\_logic\_vector

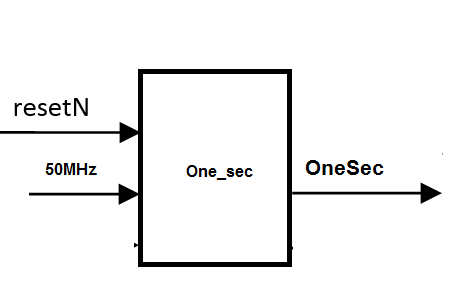
ממש תכן הירארכי בVHDL השתמש בקוד מהתרגיל הקודם

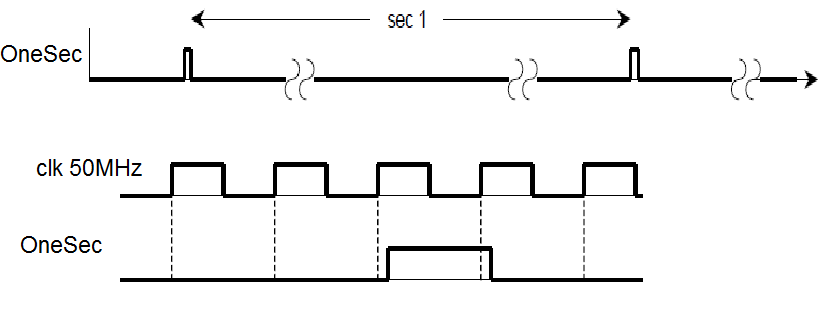


# מחלק תדר

כתוב מחלק תדר המקבל שעון של 50MHz ומוציא פולס שעון צר בתדר של 1Hz בשיטה סינכרונית. רוחב הפולס הצר יהיה כזמן מחזור יחיד של השעון המהיר .

כמתואר באיור הבא:





העזר בקוד הבא:

יש להקפיד לא להשתמש ב MOD וחילוק שהם יקרים למימוש

**signal** one\_sec\_flag **:** std\_logic **;**

**process(**CLK**,**RESETN**)**

**variable** one\_sec**:** integer **;**

-- constant sec: integer := 50000000 ; -- for Real operation

**constant** sec**:** integer **:=** 5 **;** -- for simulation

**begin**

**if** RESETN **=** '0' **then**

one\_sec **:=** 0 **;**

one\_sec\_flag **<=** '0' **;**

**elsif** **rising\_edge(**CLK**)** **then**

one\_sec **:=** one\_sec **+** 1 **;**

**if** **(**one\_sec **\*\*fill)** **then**

one\_sec\_flag **\*\*fill** **;**

one\_sec **\*\*fill;**

**else**

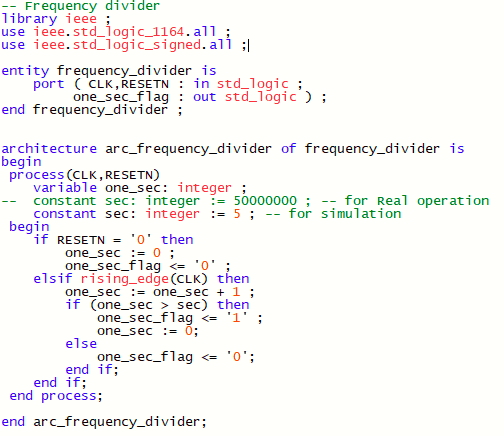
one\_sec\_flag **\*\*fill;**

**end** **if;**

**end** **if;**

**end** **process;**

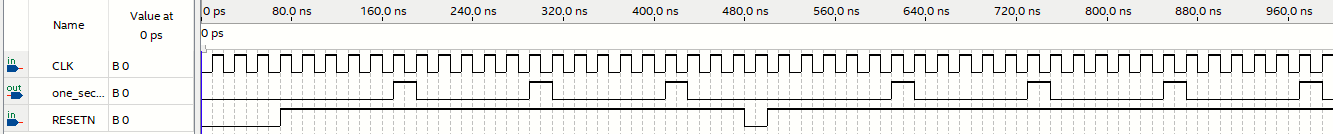
**end** Counter\_arch**;**



תכנן ופרט להלן מה תרצה לבדוק בסימולציה, אילו אותות ומצבים (מומלץ לבדוק מצב עבודה רגיל ומקרי קצה, כמו מצב RESET, סיום ספירה והתחלתה, וכו').

נרצה לבדוק שהמחלק תדר סופר רק כאשר RESETN ב 1, בדיקה של כמה מחזורים ברצף ובדיקה שהספירה מתאפסת כאשר RESETN עולה ומתחילה מההתחלה כשהוא יורד

צור קובץ WAVEFORM והרץ סימולציה של המעגל (שנה את הקבועSEC לזמן סימולציה קצר סביר)

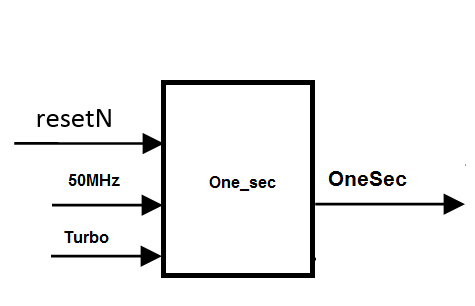


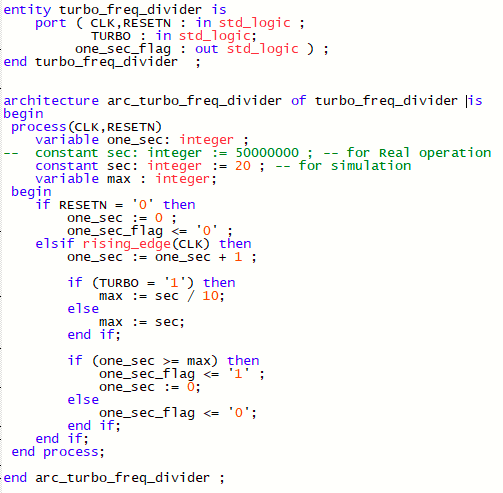
דד

# מחלק תדר- כניסת TURBO

הדגמת רכיב על הכרטיס עם שעון שמתחלף פעם בשניה היא איטית ומייגעת ולכן מומלץ להוסיף כניסת טורבו שמאיצה את תהליך ההדגמה פי -10.

לשם כך הוסף למחלק התדר כניסת טורבו: כשהיא ב-1 תדר הפולס הופך ל- Hz 10 וב-0 ללא שינוי. הקפד על קוד יעיל וסינכרוני.





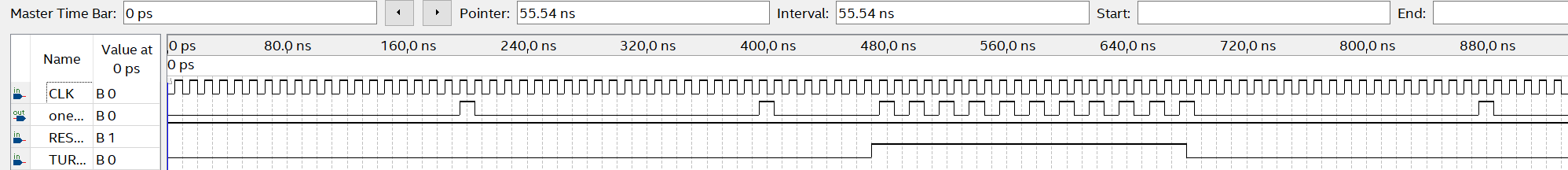
הגדר מה תרצה לראות בסימולציה

נצפה לראות כי כאשר כניסת Turbo דולקת, התדר מהיר פי 10 ביחס למתי שכניסת ה-Turbo כבוייה.

הרץ סימולציה של המעגל בקווארטוס במצבFunctional העזר ב COOK-BOOK (שנה את הקבועSEC לזמן סימולציה קצר סביר)

שנה את כניסת טורבו **במהלך הריצה** לשני הכיוונים וראה שהמונה אינו מתבדר

**יש לבדוק גם את המצב שבו TURBO משתנה כשערך המונה גדול מהSEC החדש**



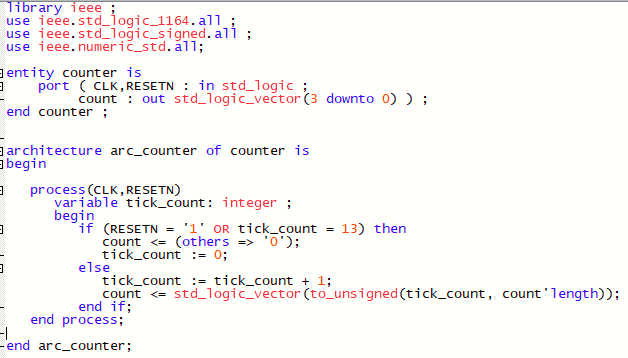
# מונה עולה פשוט

כתוב יישות (entity) של מונה בינארי סינכרוני עולה, שהוא בעל מחזור של 14. זאת אומרת המונה מתחיל לספור מ- 0, סופר עד 13, חוזר ל- 0 ושוב סופר עד 13, וכן הלאה בצורה מחזורית.

למונה כניסות: שעון ו- resetN שמאפס את יציאת המונה

ויציאות: count – וקטור של 4 ביט המראה את מצב הספירה בכל פולס שעון.

העתק את הקוד שכתבת כאן.



צור קובץ WAVEFORM והרץ סימולציה של המעגל והוסף את תוצאות הסימולציה לדו"ח.

פלט סימולציה

# תצוגת 7Segment עם הדלקה מלאה וכיבוי

באחד הניסויים הקודמים ראינו שקיימים ב - Quartus רכיבים מוכנים שממירים מקוד BCD לתצוגות Seven Segment (רכיבים תואמי TTL שנקראו 7447 ו 7448). רכיבים אלו היו מאוד מוגבלים, היות ולא יכולנו להמיר באמצעותם צירופי כניסה שהם גדולים מ:

(1001)2 = (9)10

בשאלת תכן זו יהיה עליך לתכנן ממיר צירופי פשוט מקוד בינארי ברוחב ארבע סיביות + 2 סיביות נוספות, לתצוגת Seven Segment עבור כל 16 הצירופים האפשריים של 4 סיביות. תפקיד הסיביות הנוספות יוסבר בהמשך.

לרכיב זה קרא בשם HEXSS והוא יהיה שימושי עבורנו בניסויים שבהמשך.

לתצוגות Seven Segment יש את המבנה המרחבי הבא:



שבעת המקטעים נקראים בדרך כלל בשמות של הספרות 0 – 6 (או האותיות a עד (g. לעתים קיימת בצד שמאל או בצד ימין של המבנה הנ"ל גם תצוגה קטנה נוספת שנקראת dp ושמשמשת כנקודה עשרונית (Decimal Point). נקודות dp קיימת בתצוגות שבלוח DE10 אך היא לא מחוברת לרכיב Cyclone.

במערכת שתתכנן השתמש בווקטור באורך 7 עבור היציאות ל 7Segment. קטע הקוד הבא מתאר חלק מהדקי הרכיב שעליך לתכנן עבור תצוגה של ספרה אחת:

-- HEX to Seven-Segment (active low) – vector out

**library** ieee **;**

**use** ieee**.**std\_logic\_1164**.all** **;**

**entity** hexss **is**

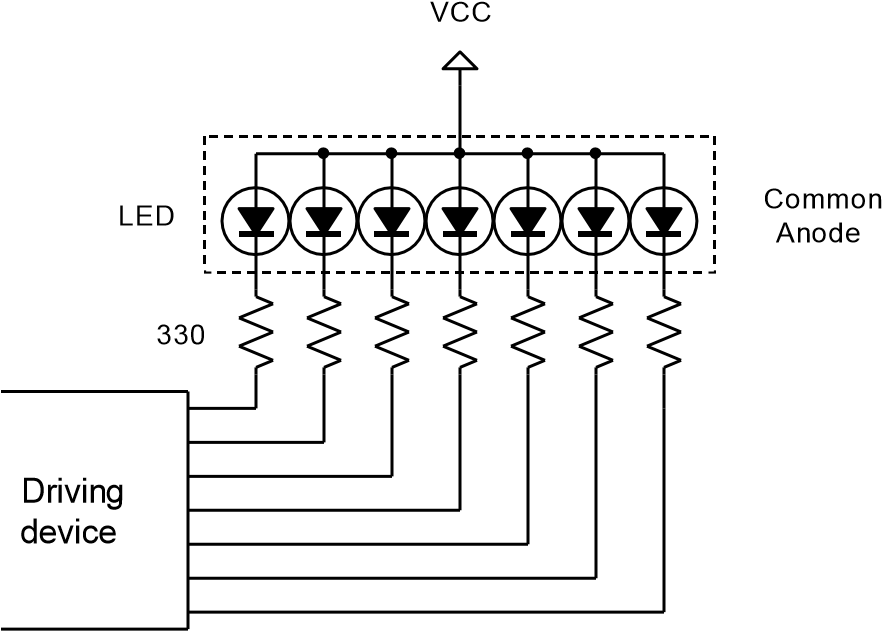
**port** **(** din **:** **in** std\_logic\_vector**(**3 **downto** 0**)** **;**

ss **:** **out** std\_logic\_vector**(**6 **downto** 0**)** **);**

-- ADD additional signals

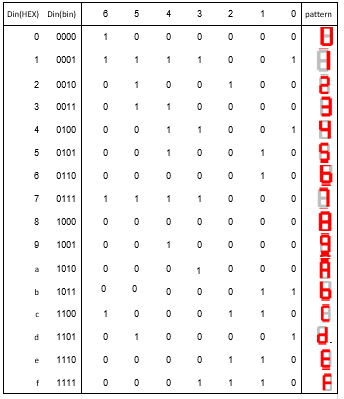
**end** hexss **;**

כזכור לך מהניסוי הראשון, מבחינה חשמלית קיימים שני סוגים של תצוגות. סוג אחד נקרא תצוגת מסוג CA) Common Anode) והיא מחוברת באופן הבא:



לצורת פעולה כזו קוראים לעתים בשם Active Low. התצוגות בלוח התרגול DE10 הן מסוג CA והן כולן פועלות בנמוך (Active Low).

טבלת האמת הבאה מתארת את הלוגיקה הרצויה של רכיב ההמרה מספרה הקסדצימלית (ארבע סיביות ) לספרת Seven-Segment שפועלת ב :Active Low –



כתוב קובץ VHDL בשם HEXSS.VHD שמקבל 4 + 2 סיביות ומציג ספרה הקסדצימלית אחת כמתואר לעיל.

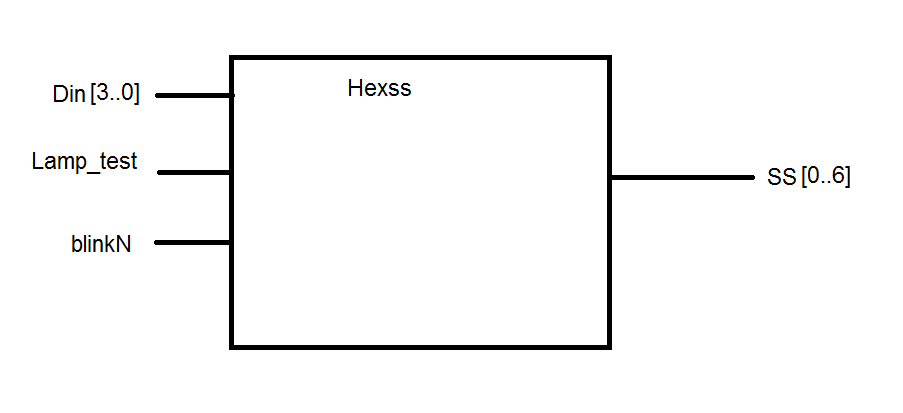
תפקידן של 2 סיביות הכניסה הנוספות הוא:

* במקרה של blinkN=0- יש להחשיך את הספרה - “ “ ללא תלות בכניסה
* במקרה של LAMP\_TEST=1 יש להדליק את כל הסגמנטים –“8” ללא תלות בכניסה
* אם שניהם מאופשרים יש להדליק את התצוגה

הקפד על כתיבת קוד צירופי חוקי שיעבור קומפילציה.

**הערה: מותר להיעזר בקוד HEXSS שקיבלת במעבדה סכמתי1 /ספרתי 1 בתור דוגמה**

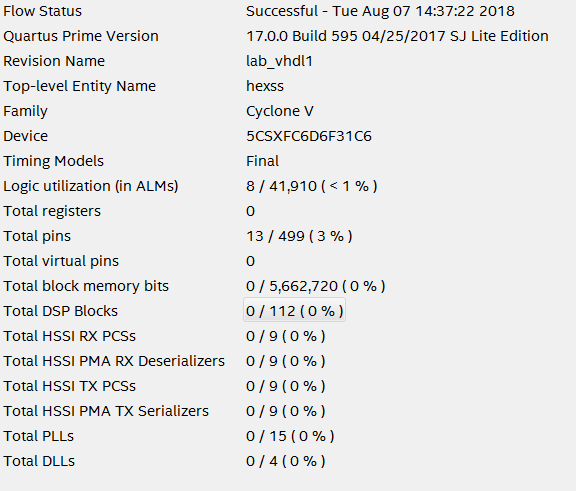
סרטט **בעפרון** סמל גרפי של הרכיב (כניסות ויציאות)



קוד VHDL בעמוד הבא



העתק דוח קומפילציה



היות ומדובר במערכת פשוטה שאותה נבדוק מיד בתחילת הניסוי באופן ישיר בחמרה, אין צורך לבצע סימולציה כל שהיא.

**להזכירכם יש להביא למעבדה את כל קבצי הקוד והפרוייקטים שכתבתם, כי תשתמשו בהם**

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

|  |
| --- |
|  |
| |  | | --- | | [**מלא את הטופס**](https://docs.google.com/forms/d/e/1FAIpQLScKIokZgowmcDuf0l79Qzn_sybx6sq9v_V_CBx9J30Exvg08w/viewform?c=0&w=1)  שמור כPDF והעלה למודל | |