

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל



מעבדה 1א'
044157

ניסוי SV1
תדריך ודוח מעבדה

גרסה 1.42
קיץ 2020

על פי חוברות של עמוס זסלבסקי, 2009

תאריך	שם המדריך בפועל	ביצוע עד סעיף	
16/08/2020	אלון מזרחי	הכל	ביצוע הניסוי המקורי
			השלמת חלקים חסרים - 1
			השלמת חלקים חסרים - 2

שם משפחה	שם פרטי	סטודנט
דביר	ליאור	1
אילתה	נועם	2

תוכן עניינים

3	1	כתיבת SV וסינתזה
3	1.1	פתיחת ארכיב מדוח ההכנה
4	2	מונה סינכרוני עם קפיצות
5	2.1	צריבה לכרטיס
6	3	מונה מתנפח
8	3.1	מימוש משווה
8	3.1.1	כתיבת קוד
8	3.1.2	סימולציה ל Control-Path
10	3.1.3	סימולציה ל Data-Path
11	3.2	מימוש מונה (slow/fast)
12	3.3	מימוש המונה המתנפח כתכן הירארכי בקוד SystemVerilog
14	3.4	שרטוט "גרפי" של המימוש ב-SV
15	3.5	סימולציה למונה המתנפח
16	4	המונה המתנפח בתכן הירארכי גרפי
16	4.1	שעון – לקצב של 1 Hz
18	4.2	שרטוט המכלול המלא – גרפי
19	4.3	סימולציה למכלול המלא
20	4.4	צריבה לכרטיס והדגמה
21	5	גיבוי העבודה

מטרת הניסוי:

במעבדה זו נתרגל לראשונה כתיבת קוד ב-SV, כתיבת מונים סינכרוניים ותכן הירארכי.

רשום את השעה בה התחלת את המעבדה:

1 כתיבת SV וסינתזה

1.1 פתיחת ארכיב מדוח ההכנה

הורד מהמודל ופתח את קובץ הארכיב ששמרת בסיום עבודת ההכנה למעבדת SV1. שמור את הקובץ ב desktop שלך בתיקייה שתייצר עבור מעבדה זו ופתח אותו לפרויקט בתיקייה הנ"ל. (הקפד לא להשאירו בתיקית downloads).
שים לב להתשמש בתיקייה קרובה ל DESKTOP ושאינה עמוק בעץ.

פתח את הקובץ של מימוש ה-MUX בעזרת משפט IF קומבינטורי mux_4to1_if.v. צמצם את המודול לשתי כניסות מידע וכניסת select של סיבית אחת ושנה את שמו ל-mux_2to1_if.
שמור את הקובץ המצומצם בשם mux_2to1_if.v תחת אותו פרויקט והגדר אותו כ-TOP.

```
module mux_2to1_if
(
    input logic [1:0] datain,
    input logic sel,
    output logic outd
);

    always_comb
    begin
        if (sel == 0) begin
            outd = datain[0];
        end
        else begin
            outd = datain[1];
        end
    end
end

endmodule
```

בצע אנליזה לקוד שלך Analysis & Elaboration.

בדוק ששמות הכניסות/יציאות שלך מתאימים לקובץ ההדקים הנתון לך. אם לא, התאם לפי הצורך.
הרץ את קובץ ה-tcl.*.

בצע קומפילציה מלאה לתכן Compilation.

שים לב! וודא בסיכום הקומפילציה שהוקצו יותר מ-0 מודולים (ALMS) לפרויקט.

הצג את ה-PIN PLANNER הכולל את הקצאת ההדקים ובדוק שהיא נכונה. ושלל הכניסות ויציאות הוגדרו בכלכה

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate
datain[1]	Input	PIN_AA15	3B	B3B_N0	PIN_AA15	3.3-V LVTTL		16mA ...ault)	
datain[0]	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	3.3-V LVTTL		16mA ...ault)	
outd	Output	PIN_AA24	5A	B5A_N0	PIN_AA24	3.3-V LVTTL		16mA ...ault)	1 (default)
select	Input	PIN_AB30	5B	B5B_N0	PIN_AB30	2.5 V		12mA ...ault)	

וודא שלל הפינים הוגדרו 3.3V אם לא, בצע Remove Assignment והקצאת הדקים מחדש.
הורד את התכן לכרטיס.

הגדר מה תרצה לבדוק על הכרטיס, כדי להראות פעולה תקינה של המערכת.

תשובה: נרצה לבדוק טבלת אמת עבור מוקס 2<-1

הגדר מה תרצה לבדוק על הכרטיס כ- TRUE NEGATIVE (בוצע למרות שלא ביקשנו) או FALSE POSITIVE (ביקשנו ולא בוצע)

תשובה:

True Negative – בחירת select של יציאה אחת לא מובילה לשנייה

False Positive – בחירת יציאה ספציפית מובילה לכניסה שלה

בדוק שהמערכת פועלת כנדרש.

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 10:57

2 מונה סינכרוני עם קפיצות

קבע את קובץ המונה עם הקפיצות jmp_counter.sv מעבודת ההכנה כ- TOP של הפרויקט. בצע אנליזה.

בדוק ששמות הכניסות/יציאות שלך מתאימים לקובץ ההדקים הנתון לך. אם לא, התאם לפי הצורך והרץ קובץ הדקים.

שים לב: - אם עשית שינויים בקובץ ההדקים יש להסיר הדקים תחילה

(Remove assignments) ואז להריץ קובץ הדקים (Run TCL Script)

- השעון clk ידני וה- resetN נקבעו לשני לחצנים

בצע קומפילציה מלאה לתכן. הוסף את טבלת ההדקים הרלוונטים וסיכום הקומפילציה לדו"ח.

שים לב! תמיד לוודא בסיכום הקומפילציה המלאה שהוקצו יותר מ- 0 מודולים (ALMS).

Node Name	Direction	Location	I/O Bank	REF Group	Port Location	I/O Standard	Reserved	Current Strength	Drive Rate	Ascent
clk	Input	PI...K4	3B	B3B_NO	PI...K4	3.3-V LVTTL		16m...lt)		
count[3]	Output	PI...22	5A	B5A_NO	PI...22	3.3-V LVTTL		16m...lt) 1 (...lt)		
count[2]	Output	PI...24	4A	B4A_NO	PI...24	3.3-V LVTTL		16m...lt) 1 (...lt)		
count[1]	Output	PI...24	4A	B4A_NO	PI...24	3.3-V LVTTL		16m...lt) 1 (...lt)		
count[0]	Output	PI...25	4A	B4A_NO	PI...25	3.3-V LVTTL		16m...lt) 1 (...lt)		
resetN	Input	PIN_AJ4	3B	B3B_NO	PIN_AJ4	3.3-V LVTTL		16m...lt)		

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun Aug 16 11:03:00 2020
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	SV1Exs
Top-level Entity Name	jmp_counter
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	8
Total pins	6 / 499 (1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
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 %)

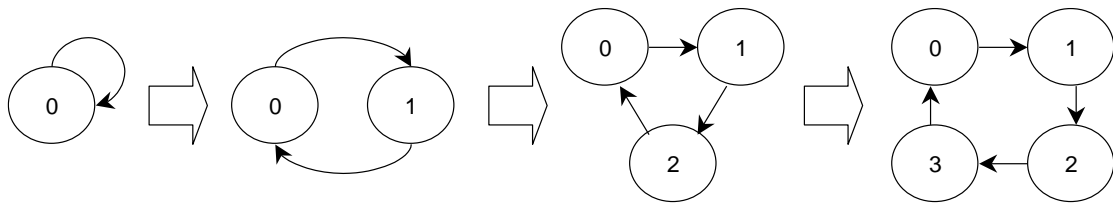
2.1 צריבה לכרטיס

צרוב את תכן המונה עם הקפיצות לכרטיס.
בדוק שהמונה עובד נכון.

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

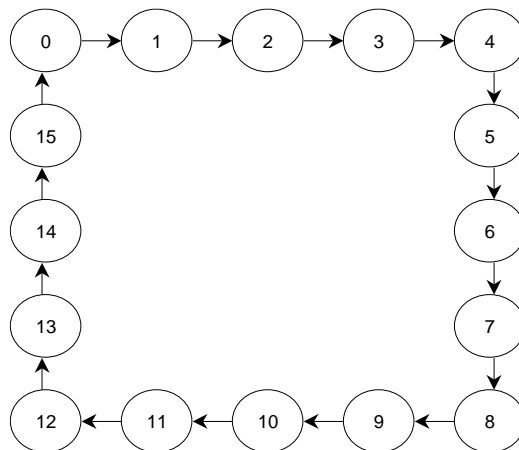
3 מונה מתנפח

בתרגיל זה יש לממש מונה ששיא הספירה שלו הולך וגדל. בתחילת הספירה (מיד לאחר איפוס המונה), הספירה המרבית של המונה מגיעה ל - 0:



במחזור הספירה הבא, הספירה המרבית מגיעה ל - 1, במחזור הספירה הבא היא מגיעה ל - 2, אחר כך ל - 3 וכן הלאה. בסופו של דבר מחזור הספירה עולה ומגיע לספירה מרבית עד 15:

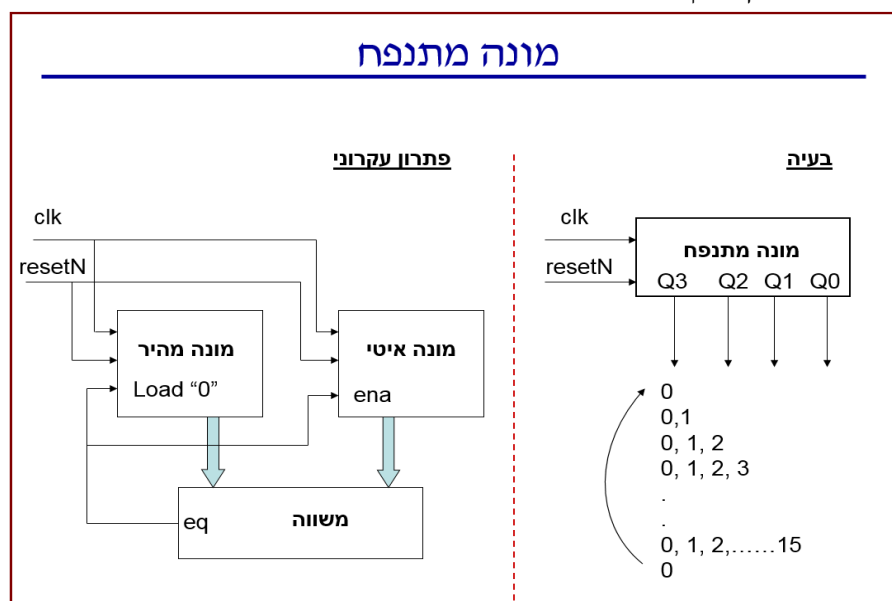
לאחר מכן, המונה מתאפס והספירות המרבית שלו שוב עולה 0, 1, 2, 3 וכו'.



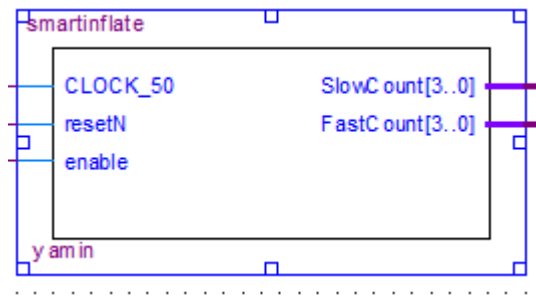
הגדרת הדרישות:

בשלב הראשון יש לבנות מונה מתנפח בעל שתי יציאות בינאריות. לשם כך יש להשתמש בשני מונים בינאריים עולים וברכיב משווה.

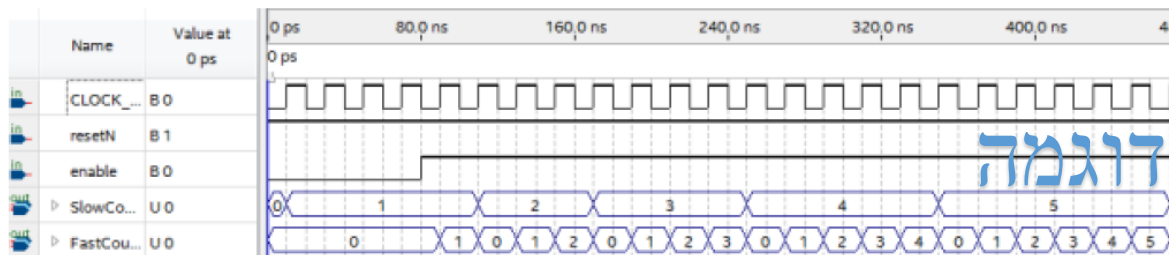
הבעיה והפתרון העקרוני שלה מוצגים בדיאגרמה הבאה.



הסימבול של מונה מתנפח צריך להיראות כך:



להלן דוגמה לתוצאות סימולציה של מונה מתנפח:



בשלב השני נרחיב מונה מתנפח בהירארכיה עליונה בעזרת השעון המהיר של הכרטיס ותצוגת 7Seg בדומה למעבדת תכן סכמתי.

הגדרת Interface של המונה המתנפח בשלב הראשון:

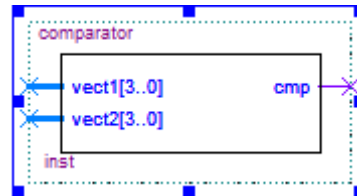
SIGNAL		
clk	Manual	Input from key
resetN	Active low – asynchronous	Key
enable	Active high - synchronic	Key
SlowCount[3:0]	Output vector	Leds
FastCount[3:0]	Output vector	Leds

נתחיל עם הבניה של כל אחד מהרכיבים הדרושים למונה המתנפח, תחילה המשווה אחר כך המונה. אחר כך נחבר את שלושת הרכיבים למימוש המונה המתנפח.

3.1 מימוש משווה

❖ בתכנון הנתון לך `comparator.sv` השלם את החיבורים החסרים לפי הלוגיקה והוסף את התכנון השלם לדו"ח.

	cmp
<code>Vect1[3..0] == Vect2[3..0]</code>	1
<code>Vect1[3..0] != Vect2[3..0]</code>	0



3.1.1 כתיבת קוד

לקראת סימולציה בצע סינתזה מוצלחת והוסף את הקוד שלך לדו"ח.

```
module comparator
(
  // Input, Output Ports
  input logic [3:0] vect1,
  input logic [3:0] vect2,
  output logic cmp
);
  assign cmp = (vect1 == vect2);
endmodule
```

הגדר מה המצבים שתוצאה לסמלט .
כיון שיש 256 אפשרויות , נחלק את הסימולציה לשני חלקים:

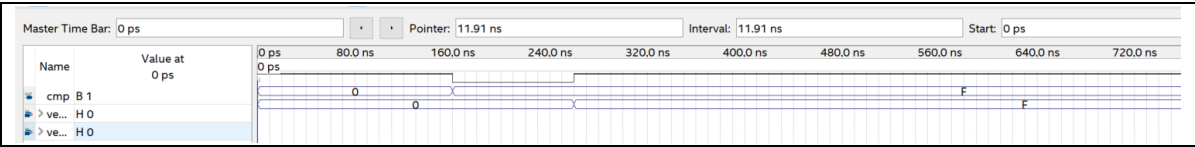
3.1.2 סימולציה ל Control-Path

בשלב הראשון נבדוק את הלוגיקה, יש לה שני מצבים: כניסות שוות וכניסות שונות

יש לפרט את המצבים

תשובה: כניסות שוות כלומר הביטים באינדקסים שווים בשני וקטורי הסיביות הם שווים.
כניסות שונות כלומר לפחות אחד מן הביטים באינדקסים שווים הם שונים.

בצע סימולציה. **שים לב** להציג את התוצאות כ UNSIGNED או HEX ולא כמספר בינארי.

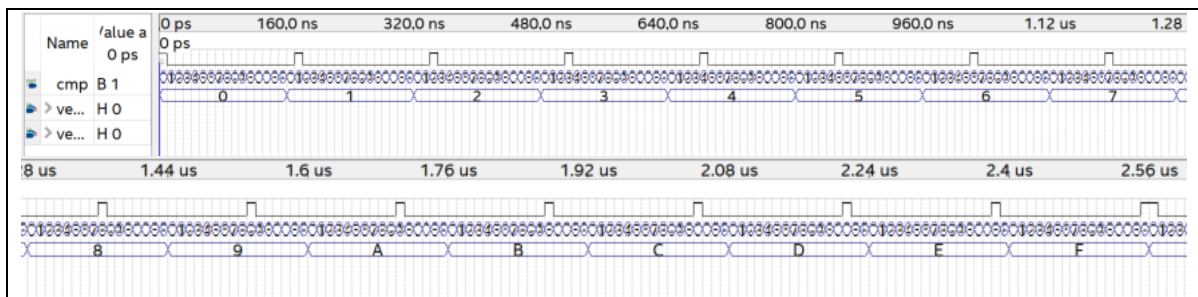


3.1.3 סימולציה ל Data-Path

בשלב השני נבדוק את כל צירופי הכניסות: יש לכסות את כל $16 \times 16 = 256$ האפשרויות. כדי להיות יעילים מומלץ להשתמש בסימולציה בשני מונים. הסבר כיצד תגדיר מונים אלה.

תשובה: נגדיר מונה שרץ בין 0 ל 15 עבור אחת הכניסות ומונה נוסף עבור הכניסה השנייה בין 0 ל 15 שעולה כל מחזור של המונה הראשון.

בצע סימולציה. **שים לב** להציג את התוצאות כ UNSIGNED או HEX ולא כמספר בינארי. ויש להציג תמונה כללית של הסימולציה.



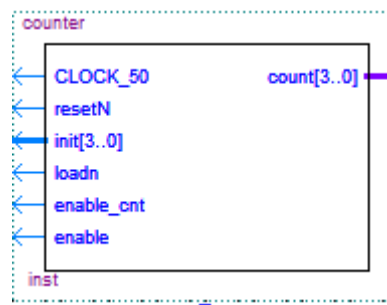
3.2 מימוש מונה (slow/fast)

יש לבנות מונה לפי טבלת האמת:

resetN	CLOCK_50	enable	loadn	enable_cnt	init[3..0]	count[3..0]	
0	x	x	x	x	x	0000	Reset
1	0, 1, ↓	x	x	x	x	previous count	
1	↑	0	x	x	x	previous count	
1	↑	1	0	x	Init[3..0]	Init[3..0]	Load
1	↑	1	1	0	x	previous count	
1	↑	1	1	1	x	count+1	Increment

שימו לב לממש את ה IF לפי סדר ההירארכיה של הטבלה. ולבצע הזחות IDENT קדי שהקוד יהיה קריא

פתח את קובץ המונה **up_counter**. הוסף לו לוגיקה על פי הדרישות בטבלת האמת. הקפד להשתמש בלוגיקה סינכרונית בלבד.



היות ונרצה לבצע סימולציה, בצע סינתזה **Analysis & Synthesis** והוסף את הקוד שלך לדו"ח.

```

module up_counter
(
    // Input, Output Ports
    input logic clk,
    input logic resetN,
    input logic loadN,
    input logic enable_cnt,
    input logic enable,
    input logic [3:0] init,
    output logic [3:0] count
);

always_ff @( posedge clk , negedge resetN )
begin

    if ( !resetN ) begin
        count <= 0;
    end else if (!enable) begin
        count <= count;
    end else if (!loadN) begin
        count <= init;
    end
end

```

```

        end else if (enable_cnt) begin
            count <= count + 1;
        end else begin
            count <= count;
        end
    end

end
endmodule

```

הגדר מה המצבים שתוצאה לסמלט. (יש לפרט את המצבים ולכסות את כל האפשרויות)

תשובה: את כלל האפשרויות:

הפעלת resetN

חוסר פעולה ללא Enable

הפעלת Enable

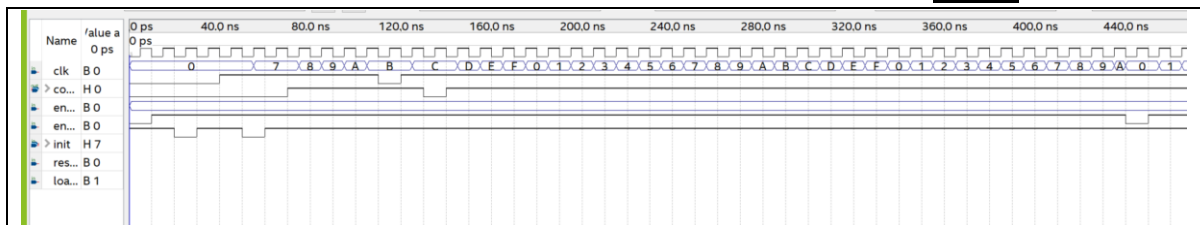
הפעלת LoadN (ננסה רק מקרה יחיד של init)

חוסר פעולה ללא enable_cnt

הפעלת enable_cnt

ספירה תקינה של count

בצע סימולציה. **שים לב** להציג את המונה כ- UNSIGNED או HEX ולא כמספר בינארי.



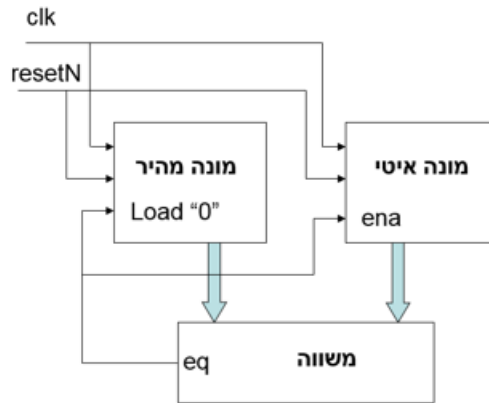
3.3 מימוש המונה המתנפח כתכן הירארכי בקוד SystemVerilog

יש לממש את המונה המתנפח על ידי חיבור שני מונים ומשווה. תבנה מימוש זה כתכן הירארכי בתוכנה, ב-SV, על ידי instantiation (הפעלה) של המודולים הנ"ל.

השתמש בקובץ השלד הנתון לך **inflating_counter** והוסף את החלקים החסרים תוך קביעת החיבורים המתאימים בין שני המונים והמשווה.

שים לב לחבר את כניסת ה- ENABLE החיצונית לשני המונים, נשתמש בה להפעלת המערכת רק פעם אחת בשניה.

פתרון עקרוני



בצע סינתזה לקוד שלך והוסף אותו לדו"ח.
יש לוודא שלכל הסיגנלים שמות מובנים ומשמעותיים (coding convention)

```
module inflating_counter
(
    // Input, Output Ports
    input logic clk,
    input logic resetN,
    input logic enable,

    output logic [3:0] FastCount,
    output logic [3:0] SlowCount
);

    logic enable_cnt;

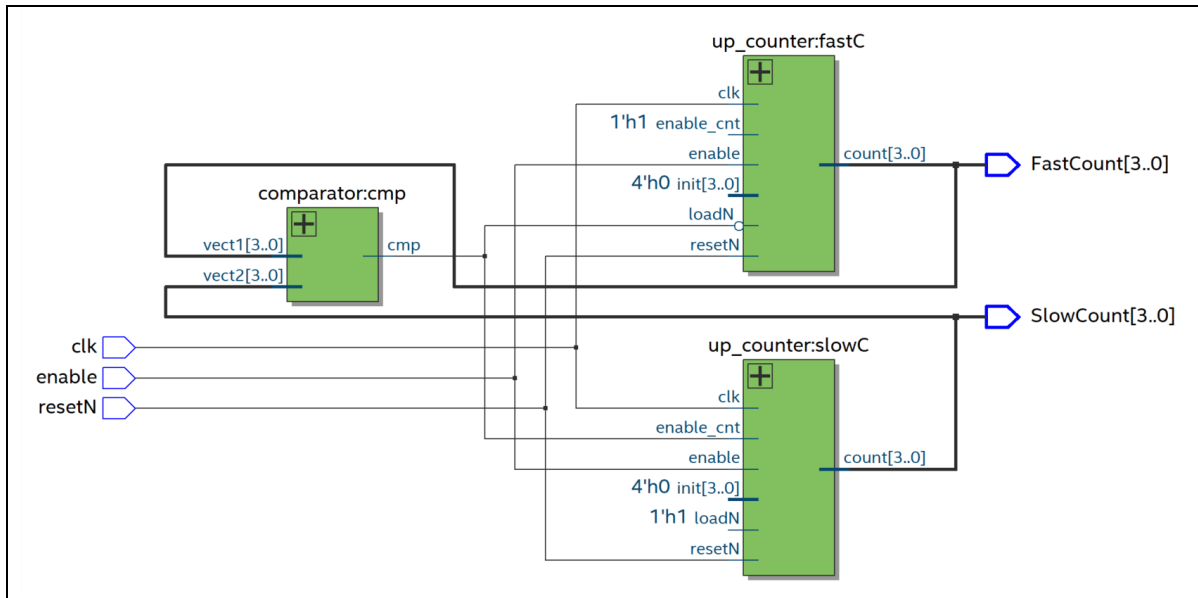
    // Fast counter instantiation
    up_counter fastC(.clk(clk), .resetN(resetN),
        .loadN(!enable_cnt), .enable_cnt(1'b1),
        .enable(enable),
        .init(4'h00), .count(FastCount)
    );

    // Slow counter instantiation
    up_counter slowC(.clk(clk), .resetN(resetN),
        .loadN(1'b1), .enable_cnt(enable_cnt),
        .enable(enable),
        .init(4'h00),
        .count(SlowCount)
    );

    // Comparator instantiation
    comparator cmp
        (.vect1(FastCount), .vect2(SlowCount),
        .cmp(enable_cnt)
    );
endmodule
```

3.4 שרטוט "גרפי" של המימוש ב-SV

בדומה לדו"ח ההכנה, הצג בצורה גרפית את מימוש המונה בעזרת ה- RTL Viewer ובדוק שהחיבורים שלך נכונים. אם לא, תקן בהתאם.



3.5 סימולציה למונה המתנפח

הגדר מה תרצה לבדוק בסימולציה.

תשובה:

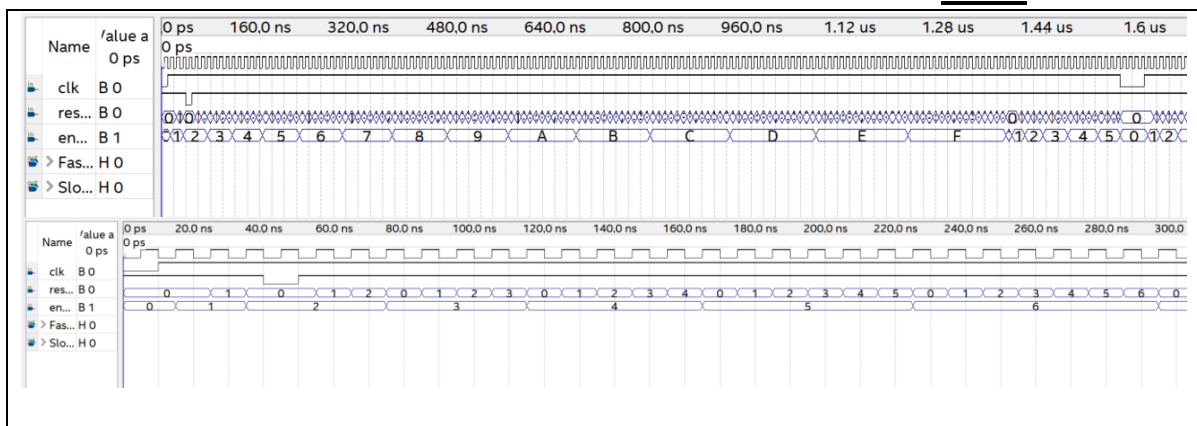
הפעלת resetN

חוסר פעולה ללא Enable

הפעלת Enable

ספירה תקינה של מונה מתנפח

בצע סימולציה **שים לב** להציג את התוצאות כ- UNSIGNED או HEX ולא כמספר בינארי



וודא שבדקת את כניסת ENABLE , אם לא חזור על הסימולציה

צור סימבול של המונה המתנפח.

אין צורך להוריד לכרטיס בשלב זה!

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 12:28

4 המונה המתנפח בתכן הירארכי גרפי

בתכן הירארכי עליון נרצה לחבר את המונה המתנפח במערכת שתאפשר הפעלה קלה שלו. מכיוון שקשה ללחוץ על לחצן השעון עשרות פעמים, וקשה לקרוא תצוגת נוריות שמשתנה במהירות, נרחיב את מימוש המונה המתנפח בהירארכיה עליונה, על ידי שימוש בשעון פנימי 50MHz של הכרטיס (להפעלה אוטומטית), נוסף שעון של שניה אחת (להאטת הקצב) ותצוגות של 7Seg להצגת ספרות (מעבודת ההכנה).

4.1 שעון – לקצב של 1 Hz

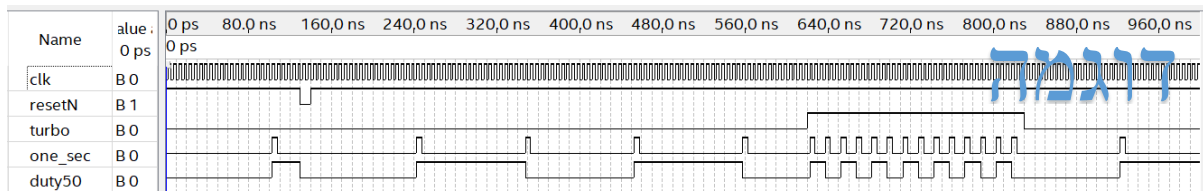
נתון לך מודול של מונה מחלק תדר, בשם `one_sec_counter.sv`, שסופר פולסי שעון, וממיר את קצב השעון המאד מהיר של הכרטיס DE10 (50 MHz), לפולס צר בקצב איטי יותר של 1 Hz. מודול זה בעל הכניסות:

- clk – שעון (50 MHz של הכרטיס)
- resetN – כניסת איפוס אסינכרונית
- TURBO – כניסה של ביט אחד. כאשר היא ב-1 לוגי תדר הפולס עולה ל-10 Hz, ב-0 לוגי התדר נשאר ללא שינוי

והיציאות:

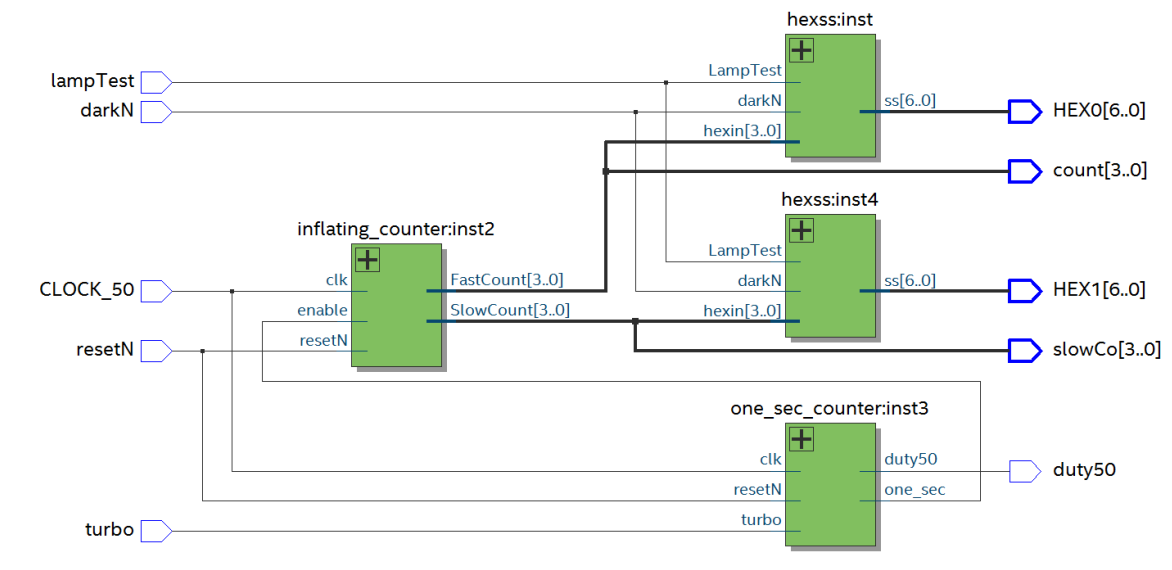
- one_sec פולס צר מאד, ברוחב של 10 nsec (לפי 50 MHz), בתדר של 1 Hz, מעין "שעון איטי"
- duty50 – פולס רחב, ברוחב של שניה, בתדר של 0.5 Hz.

במצב TURBO שתי היציאות מהירות פי 16, כפי שניתן לראות להלן בסימולציה של מודול זה.



פתח את הקובץ הגרפי `inflating_cnt_top` וקבע אותו כ- TOP.

המערכת אותה תממש לצורך הבדיקה, **דומה** למערכת הבאה, אבל תוך שימוש ברכיבים שלך.



הערה: הכניסות darkN ו-LampTest מחוברות לכניסות. לאן תחברן בכרטיס ה-DE-10 ומה אמור להיות מצבן כדי שאפשר יהיה להשתמש במעגל?

תשובה: למתגים כדי, כדי שישארו במצב שלהם ללא צורך בלחיצה תמידית.

לאיזה רכיב שעל הכרטיס תחבר את היציאות של המודול HEXSS?

תשובה: לנורות seg7 ברכיב.

4.2 שרטוט המכלול המלא – גרפי

הסבר: נתון לך שלד של המערכת בהירארכיה עליונה, בקובץ `inflating_cnt_top.bdf`.
לצורך השלמת השרטוט בהירארכיה עליונה תחילה יש ליצור סימבול גרפי לכל אחד מהמודולים שכתבת, המודול `inflating_counter.sv` ו-`hexss.sv`, ולהוסיף אותם למערכת.

לשם כך בצע:

- צור סימבולים למודולים שכתבת, העזר ב COOK BOOK
- השלם את ההירארכיה העליונה עם הרכיבים והחיבורים הנדרשים. היעזר בסכימת ה-RTL הנתונה

חיבורי הכניסות (חלקם כבר קיימים):

- חבר את כניסת ה- resetN ללחצן KEY0
- חבר את כניסת השעון למתנד 50MHZ פין

PIN_AF14 -to CLOCK_50

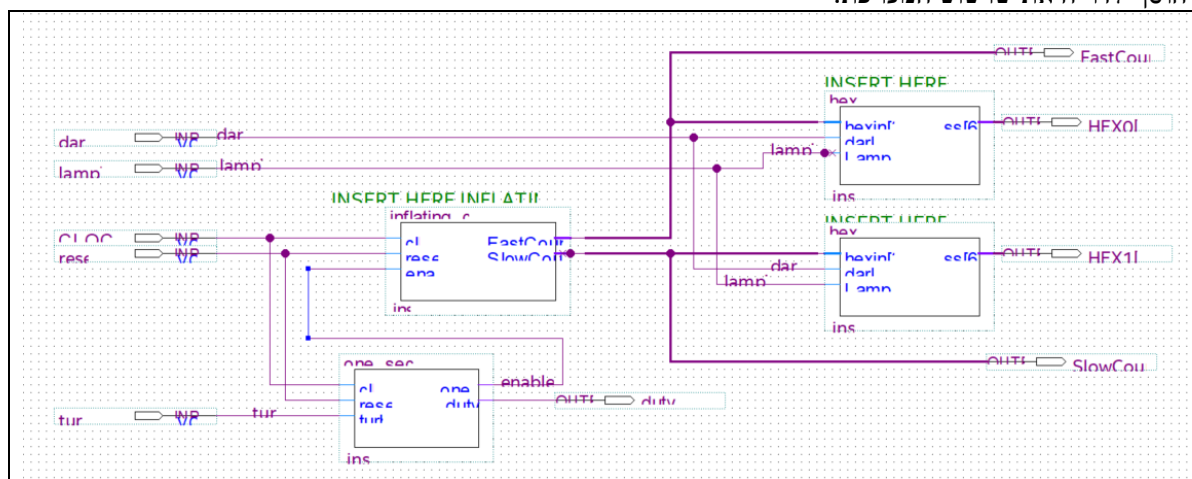
- ולא למפסק (או שתלחצו עליו 50 מיליון פעמים ☺)
- חבר את כניסת הטורבו למפסק (לא ללחצן)
- היציאה `one_sec` של שעון השניות מהווה כניסה ל- `enable` של המונה המתנפח

חיבורי היציאות (חלקם כבר קיימים):

- היציאה `duty50` של השעון `one_sec_counter` - לנורית LEDR אדומה
- היציאות של המונה המתנפח מחוטים באופן ישיר לנוריות LEDR ולכניסות של HEXSS
- היציאות של HEXSS - ל- 7Seg – דרך קובץ ההדקים

חיבורים אלה כבר קיימים בקובץ ההדקים הנתון. אך בדוק שהשמות שלך תואמים למה שנתון.
אפשר להעזר בתכן ההירארכי שבנית במעבדת תכן סכמתי.

הוסף לדו"ח את שרטוט המערכת:



שים לב! היות והשלב הבא הוא סימולציה, לפני ביצוע סינתזה יש להקטין את הקבוע במונה מחלק התדר `one_sec_counter.sv` מ-50,000,000 למספר קטן יותר, למשל 32.

לקראת סימולציה בצע סינתזה מוצלחת וצרף את הסיכום לדו"ח.

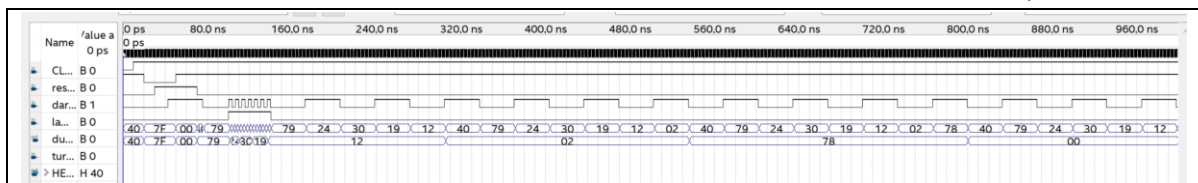
Flow Summary	
Flow Status	Successful - Sun Aug 16 12:41:17 2020
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	SV1Exs
Top-level Entity Name	inflating_cnt_top
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	42
Total pins	28
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

4.3 סימולציה למכלול המלא

הגדר מה תרצה לבדוק בסימולציה רשום את כל מצבי הכניסות ויציאות המיוחדים

תשובה:
RESET
TURBO
DarkN
LampTest
ספירה תקינה

בצע סימולציה, שים לב להציג את תוצאות המונים כ UNSIGNED INTEGER או HEX ולא כמספר בינארי.



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

4.4 צריבה לכרטיס והדגמה

הרץ את קובץ ההדקים המעודכן. TCL

שים לב! לקראת הצריבה החזר את הקבוע במונה מחלק התדר ל- 50,000,000 עבור פעולה עם השעון 50 Hz של הכרטיס.

בצע קומפילציה מלאה לתכן.

הצג את ה- PIN PLANNER הכולל את הקצאת ההדקים הנכונה.

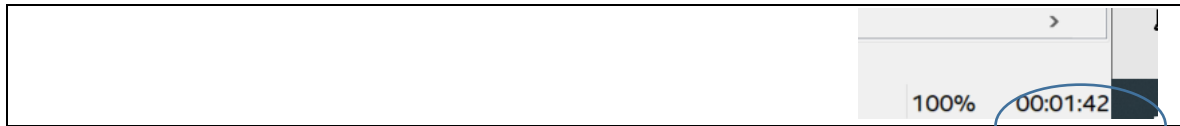
בדקו שכל הרגליים הרלוונטיות מוגדרות ל V 3.3

Node Name	Direction	Location	I/O Bank	REF Group	Port Location	I/O Standard	Reserved	Output Strength	Drive Rate	Differential
CLOCK_50	Input	PI...14	3B	B3B_NO	PI...14	3.3...TTL		16m...lt)		
darkN	Input	PI...28	5B	B5B_NO	PI...28	3.3...TTL		16m...lt)		
duty50	Output	PI...22	4A	B4A_NO	PI...22	3.3...TTL		16m...lt)	1 (...lt)	
Fast...t[3]	Output	PI...25	4A	B4A_NO	PI...25	3.3...TTL		16m...lt)	1 (...lt)	
Fast...t[2]	Output	PI...24	4A	B4A_NO	PI...24	3.3...TTL		16m...lt)	1 (...lt)	
Fast...t[1]	Output	PI...23	4A	B4A_NO	PI...23	3.3...TTL		16m...lt)	1 (...lt)	
Fast...t[0]	Output	PI...23	5A	B5A_NO	PI...23	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[6]	Output	PI...18	4A	B4A_NO	PI...18	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[5]	Output	PI...18	4A	B4A_NO	PI...18	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[4]	Output	PI...17	4A	B4A_NO	PI...17	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[3]	Output	PI...16	4A	B4A_NO	PI...16	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[2]	Output	PI...17	4A	B4A_NO	PI...17	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[1]	Output	PIN_V18	4A	B4A_NO	PIN_V18	3.3...TTL		16m...lt)	1 (...lt)	
HEX0[0]	Output	PI...17	4A	B4A_NO	PI...17	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[6]	Output	PIN_V17	4A	B4A_NO	PIN_V17	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[5]	Output	PI...17	4A	B4A_NO	PI...17	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[4]	Output	PI...18	4A	B4A_NO	PI...18	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[3]	Output	PI...17	4A	B4A_NO	PI...17	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[2]	Output	PI...16	4A	B4A_NO	PI...16	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[1]	Output	PIN_V16	4A	B4A_NO	PIN_V16	3.3...TTL		16m...lt)	1 (...lt)	
HEX1[0]	Output	PI...16	4A	B4A_NO	PI...16	3.3...TTL		16m...lt)	1 (...lt)	

Node Name	Direction	Location	I/O Bank	REF Group	Port Location	I/O Standard	Reserved	Output Strength	Drive Rate	Differential
HEX1[2]	Output	PI...16	4A	B4A_NO	PI...16	3.3-V LVTTL		16m...lt)	1 (...lt)	
HEX1[1]	Output	PIN_V16	4A	B4A_NO	PIN_V16	3.3-V LVTTL		16m...lt)	1 (...lt)	
HEX1[0]	Output	PI...16	4A	B4A_NO	PI...16	3.3-V LVTTL		16m...lt)	1 (...lt)	
lampTest	Input	PI...30	5B	B5B_NO	PI...30	3.3-V LVTTL		16m...lt)		
resetN	Input	PIN_AJ4	3B	B3B_NO	PIN_AJ4	3.3-V LVTTL		16m...lt)		
Slow...t[3]	Output	PI...22	5A	B5A_NO	PI...22	3.3-V LVTTL		16m...lt)	1 (...lt)	
Slow...t[2]	Output	PI...24	4A	B4A_NO	PI...24	3.3-V LVTTL		16m...lt)	1 (...lt)	
Slow...t[1]	Output	PI...24	4A	B4A_NO	PI...24	3.3-V LVTTL		16m...lt)	1 (...lt)	
Slow...t[0]	Output	PI...25	4A	B4A_NO	PI...25	3.3-V LVTTL		16m...lt)	1 (...lt)	
turbo	Input	PIN_Y27	5B	B5B_NO	PIN_Y27	3.3-V LVTTL		16m...lt)		

רשום כמה זמן לקחה הקומפילציה (פינה ימנית למטה במסך) 1:42

ציין את מערכת ההפעלה 7WIN או 10WIN



הורד את התכנן לכרטיס. בדוק שהמערכת פועלת כנדרש:

- המונה מתנפח מ- 0 עד 15, מתאפס ומתחיל להתנפח מחדש.
- בדוק איפוס ב- .resetN
- בדוק שבמצב TURBO המונה מתקדם בקצב מהיר פי 10.
- בדוק lampTest ו- darkN.

אם המערכת אינה פעולת כנדרש מצא באיזה חלק יש בעיה ותקן אותה.

קרא למדריך, רשום את השעה בה הוא ראה את המעגל: 12:55

5 גיבוי העבודה

שמור את הפרויקט רגיל וגם כארכיב (באמצעות Project -> Archive Project).
והעלה אותו למודל כי תצטרך אותו בהמשך.

שמור את הדו"ח רגיל וכ- PDF והעלה אותו למודל.

גבה את הדו"ח והפרויקט גם באמצעים אחרים (כגון העלאה ל- Google drive).

רשום את השעה בה סיימת את המעבדה: 13:04