top.vhd:

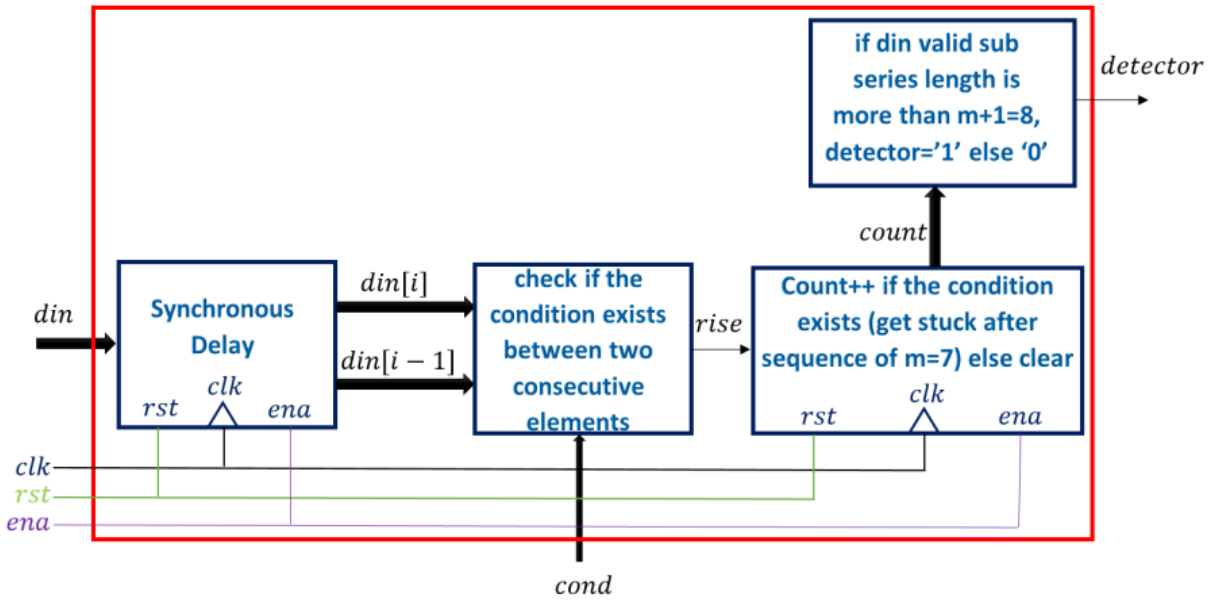
בקובץ הזה נמצא Entity בשם top בעל הכניסות והמוצאים הבאים:

שם הפורט	סוג	הסבר
rst	Input bit	ביט שתפקידו לאפס את המערכת
clk	Input bit	סיגנל מחזורי שמסנכרן את כל המערכת
ena	Input bit	סיגנל אשר מפעיל את פעולת המערכת
din	Input vector (n bits)	ווקטור הכניסה, נדגם בכל עליית שעון (clk)
cond	Input integer (range 0-3)	מספר אשר פוקד איזה סוג תת-סדרה אנו בודקים
detector	Output bit	ביט שמאשר שתת-הסדרה תקינה

מכיל את ה-processes הבאים: updateCondProcess, counterProc, delayProc,.

בעצם היצוג של תת מודולים הבאים: Adder.vhd, Counter.vhd,Cond.vhd, detector_val.vhd, SynchronousDelay.vhd הנמצאים בתיקיה **MODELING כלל TB.**

איור המודול:



updateCondProcess:

process זה אחראי לעדכון ביט הכניסה (cinSIG) ואחד משני וקטורי הכניסה למודול Adder (adderInSIG). למעשה, יש לעדכן את ערכים אלו כל פעם כאשר הערך של cond משתנה – כי אנחנו מתאימים אותם רק ל-cond.בנוסף הוא אחרי על בדיקת שינוי בערך של המוצא של מודול Adder (adderS). כל פעם שיש שינוי בערך הסיגנל adderS, אנחנו בודקים אם adderS העדכני שווה ל-D_next (שהוא בעצם din[i]). כאשר הם שווים, זה אומר ששני סיגנלים עוקבים מקיימים את תנאי תת-הסדרה הנתונה, ונעלה את הביט riseSig להיות '1'.

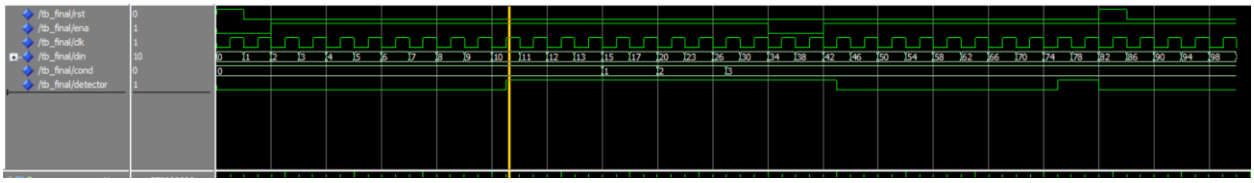
:counterProc

process שאחראי על ספרות כמות הפעמים שרצף תקין הופיע.

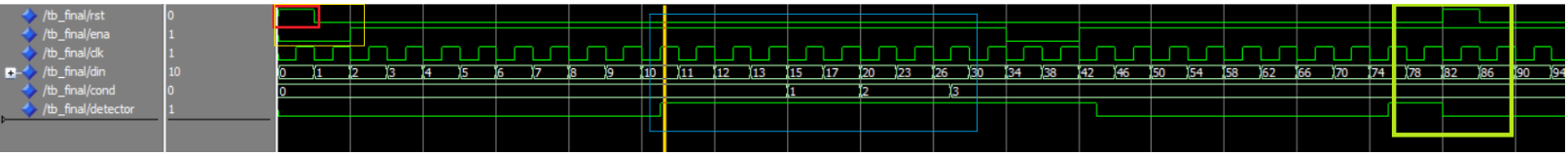
:delayProc

מייצר דילאי של מחזור ומוציא את הקלט הנוכחי וקלט שהיה מחזור 1 קודם.

דיאגרמה נקייה:



דיאגרמה עם הסבר:



ניתן לראות מה שמסומן באדום וצהוב בהתחלה זה מצב של RST של המעגל והדלת ENABLE FLAG.אחרי זה ניתן לראות שקיים רצף של 7 תנאי שהתקיימו לכן detector עולה ל1.בTB הנוכחי ניתן לראות מצב מעניין שמסומן בצבע כחול. COND השתנה וdetector נשאר על 1 למרות שלא הופיע רצף של 7 באותו COND אף ניתן לראות שקלט השתנה גם כך ש RISE עדיין 1 ולכן detector נשאר על 1.בריבוע האחרון בצבע ירוק ניתן לראות שכאשר הדליקו דגל RST אזי פלט של מעגל התאפס.

דיאגרמת ה-LIST כדי שנוכל לראות את טבלת האמת של גרף ה-wave:

ps	delta	/tb_final/rst	/tb_final/ena	/tb_final/clk	/tb_final/din	/tb_final/cond	/tb_final/detector
55000000	+1	0	1	1	00000101	0	0
60000000	+1	0	1	0	00000110	0	0
65000000	+1	0	1	1	00000110	0	0
70000000	+1	0	1	0	00000111	0	0
75000000	+1	0	1	1	00000111	0	0
80000000	+1	0	1	0	00001000	0	0
85000000	+1	0	1	1	00001000	0	0
90000000	+1	0	1	0	00001001	0	0
95000000	+1	0	1	1	00001001	0	0
100000000	+1	0	1	0	00001010	0	0
105000000	+1	0	1	1	00001010	0	0
105000000	+3	0	1	1	00001010	0	1
110000000	+1	0	1	0	00001011	0	1
115000000	+1	0	1	1	00001011	0	1
120000000	+1	0	1	0	00001100	0	1
125000000	+1	0	1	1	00001100	0	1
130000000	+1	0	1	0	00001101	0	1
135000000	+1	0	1	1	00001101	0	1
140000000	+1	0	1	0	00001111	1	1
145000000	+1	0	1	1	00001111	1	1
150000000	+1	0	1	0	00010001	1	1
155000000	+1	0	1	1	00010001	1	1
160000000	+1	0	1	0	00010100	2	1
165000000	+1	0	1	1	00010100	2	1
170000000	+1	0	1	0	00010111	2	1
175000000	+1	0	1	1	00010111	2	1
180000000	+1	0	1	0	00011010	2	1
185000000	+1	0	1	1	00011010	3	1
190000000	+1	0	1	0	00011110	3	1
195000000	+1	0	1	1	00011110	3	1
200000000	+1	0	0	0	00100010	3	1
205000000	+1	0	0	1	00100010	3	1
210000000	+1	0	0	0	00100110	3	1
215000000	+1	0	0	1	00100110	3	1
220000000	+1	0	1	0	00101010	3	1
225000000	+1	0	1	1	00101010	3	1
225000000	+3	0	1	1	00101010	3	0
230000000	+1	0	1	0	00101110	3	0
235000000	+1	0	1	1	00101110	3	0
240000000	+1	0	1	0	00110010	3	0
245000000	+1	0	1	1	00110010	3	0

Adder.vhd:

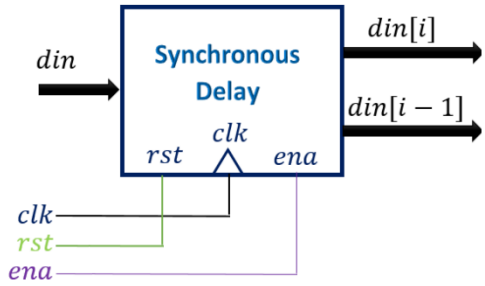
מודול זה בעצם מקבל 2 וקטורים באורך n וביט בשם cin ותפקידו פשוט לחבר ביניהם. אנו משתמשים בו לשם ווידוא ששני וקטורי כניסה עוקבים מקיימים את תנאי התת-הסדרה שאותה אנחנו רוצים לבדוק (לפי הערך של cond).
בפועל, אנחנו מחברים אליו לווקטור אחד את din[i-1], ובכניסה לווקטור השני ולביט הכניסה אנחנו מחברים אות שמתאים לתת-הסדרה שאנחנו בוחנים. לבסוף, אנחנו משווים את המוצא ל-din[i] וכך נדע אם התנאי מתקיים.

SynchronousDelay process:

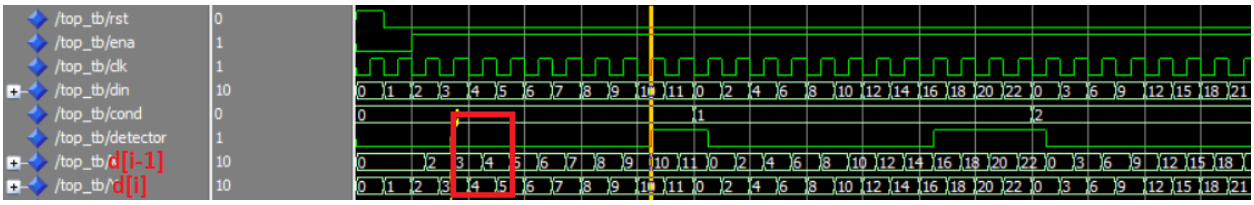
מודול זה אחראי על דגימת ערכי ווקטור הכניסה din ולאחר מכן שמירת הערך הנוכחי והערך הקודם. הדגימה נעשית בכל עליית שעון וכל עוד ena='1'. כאשר rst='1' ערכי din[i] ו din[i-1] מתאפסים.
פירוט הכניסות והמוצאים של המודול:

שם הפורט	סוג	הסבר
rst	Input bit	ביט שתפקידו לאפס את המערכת
clk	Input bit	סיגנל מחזורי שמסנכרן את כל המערכת
ena	Input bit	סיגנל אשר מפעיל את פעולת המערכת
din	Input vector (n bits)	ווקטור הכניסה, נדגם בכל עליית שעון (clk)
din_i	Output vector (n bits)	ווקטור הכניסה ברגע הנתון din[i]
din_iMinus	Output vector (n bits)	ווקטור הכניסה ברגע הקודם din[i-1]

איור המודול:



דיאגרמה עם הסבר:



ניתן לראות במלבן האדום בתמונה למעלה שחלק של דילאי עובד כמו שהוגדר בתרגיל.הוא סינכרוני לCLK ואסינכרוני לRST.בנוסף ניתן לראות שכאשר קיימת עליית שעון אזי מודול הזה מוציא את הקלט בדילאי .

דיאגרמה של מודול מחוץ למערכת:

ניתן לראות את הדילאי כפי שהוגדר בתרגיל.

דיאגרמת ה-LIST בדי שנוכל לראות את טבלת האמת של גרף ה-wave:

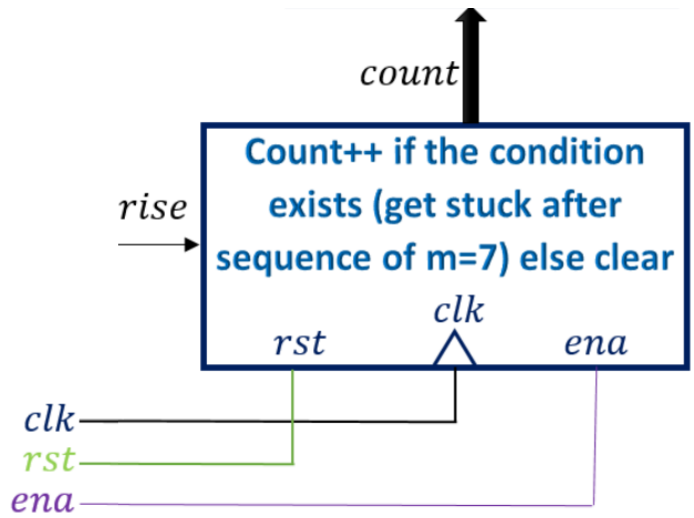
ps	delta	/synchronousdelay_tb/L0/rst	/synchronousdelay_tb/L0/ena	/synchronousdelay_tb/L0/clk	/synchronousdelay_tb/L0/din	/synchronousdelay_tb/L0/din_i	/synchronousdelay_tb/L0/din_iMinus
0	+0	U	U	U	UUUUUUUU	UUUUUUUU	UUUUUUUU
0	+1	1	0	0	00000000	UUUUUUUU	UUUUUUUU
0	+3	1	0	0	00000000	00000000	00000000
5000000	+1	1	0	1	00000000	00000000	00000000
10000000	+1	0	0	0	00000001	00000000	00000000
15000000	+1	0	0	1	00000001	00000000	00000000
20000000	+1	0	1	0	00000010	00000000	00000000
25000000	+1	0	1	1	00000010	00000000	00000000
25000000	+3	0	1	1	00000010	00000010	00000000
30000000	+1	0	1	0	00000011	00000010	00000000
35000000	+1	0	1	1	00000011	00000010	00000000
35000000	+3	0	1	1	00000011	00000011	00000010
40000000	+1	0	1	0	00000100	00000011	00000010
45000000	+1	0	1	1	00000100	00000011	00000010
45000000	+3	0	1	1	00000100	00000100	00000011
50000000	+1	0	1	0	00000101	00000100	00000011
55000000	+1	0	1	1	00000101	00000100	00000011
55000000	+3	0	1	1	00000101	00000101	00000100
60000000	+1	0	1	0	00000110	00000101	00000100
65000000	+1	0	1	1	00000110	00000101	00000100
65000000	+3	0	1	1	00000110	00000110	00000101
70000000	+1	0	0	0	00000111	00000110	00000101
75000000	+1	0	0	1	00000111	00000110	00000101
80000000	+1	0	0	0	00001000	00000110	00000101
85000000	+1	0	0	1	00001000	00000110	00000101
90000000	+1	0	0	0	00001001	00000110	00000101
95000000	+1	0	0	1	00001001	00000110	00000101
100000000	+1	0	1	0	00001010	00000110	00000101
105000000	+1	0	1	1	00001010	00000110	00000101
105000000	+3	0	1	1	00001010	00001010	00000110
110000000	+1	0	1	0	00001011	00001010	00000110
115000000	+1	0	1	1	00001011	00001010	00000110
115000000	+3	0	1	1	00001011	00001011	00001010
120000000	+1	0	1	0	00000000	00001011	00001010
125000000	+1	0	1	1	00000000	00001011	00001010
125000000	+3	0	1	1	00000000	00000000	00001011
130000000	+1	1	1	0	00000010	00000000	00001011
130000000	+3	1	1	0	00000010	00000000	00000000
135000000	+1	1	1	1	00000010	00000000	00000000
140000000	+1	1	1	0	00000100	00000000	00000000
145000000	+1	1	1	1	00000100	00000000	00000000
150000000	+1	0	1	0	00000110	00000000	00000000
155000000	+1	0	1	1	00000110	00000000	00000000
155000000	+3	0	1	1	00000110	00000110	00000000

Counter process:

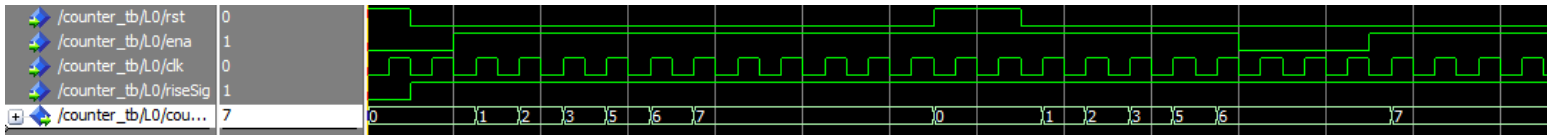
מודול זה אחראי על ספירת כמות הפעמים שהביט rise היה מורם. הספירה היא בכל עליית שעון, כל עוד ena='1' וכל עוד riseSig='1'. פירוט הכניסות והמוצאים של המודול:

שם הפורט	סוג	הסבר
rst	Input bit	ביט שתפקידו לאפס את המערכת
clk	Input bit	סיגנל מחזורי שמסנכרן את כל המערכת
ena	Input bit	סיגנל אשר מפעיל את פעולת המערכת
riseSig	Input bit	סיגנל ביט כניסה, מורה על כך שתת סדרה מתקיימת ברגע הנתון
counterMax	input bit	סיגנל שמסמן שהיה רצף באורך רצוי
counterResult	Output vector (k bits)	ווקטור מוצא שסופר כמה פעמים תת-הסדרה התקיימה

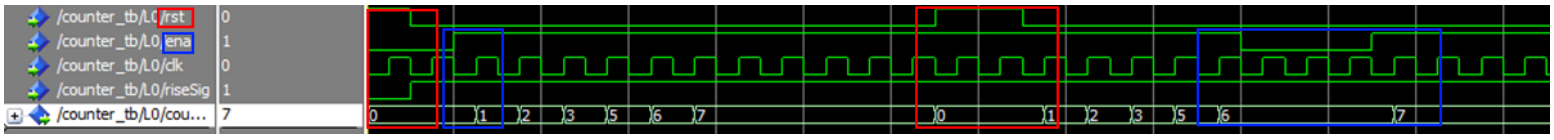
איור המודול:



דיאגרמה נקייה עבור :

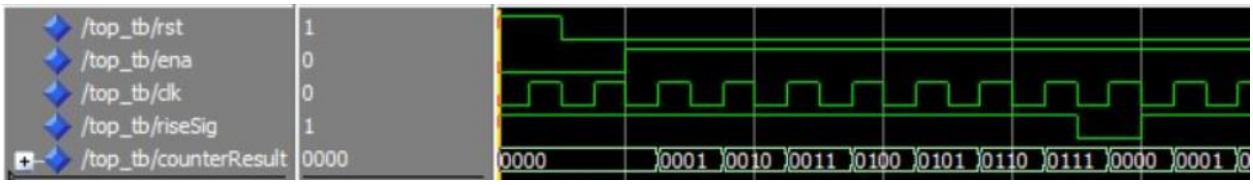


דיאגרמה עם הסבר:



המסומן באדום: ניתן לראות שברגע ש'rst='1 המוצא ישר מתאפס, ללא תלות ב-clk.
המסומן בכחול: ניתן לראות שברגע ש'ena='0 המוצא שומר על הערך שהיה בו לפני השינוי של ביט ena. אחרי שהוא חוזר להיות '1' המערכת ממשיכה לפעול כרגיל.
בכללי, כל עוד riseSig='1' ויש עלייה של clk מוצא הcounter עולה ב-1.

דיאגרמה של תת מודול מחוץ למערכת:



ניתן לראות שCOUNTER עובד כפי שמוגדר

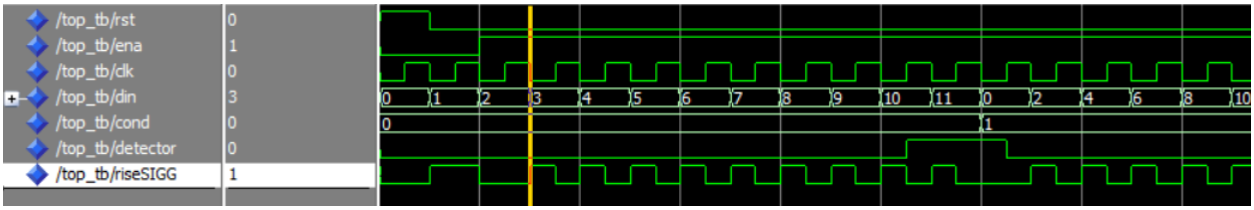
דיאגרמת ה-LIST כדי שנוכל לראות את טבלת האמת של גרף ה-wave:

ps	delta	/counter_tb/L0/rst	/counter_tb/L0/ena	/counter_tb/L0/clk	/counter_tb/L0/riseSig	/counter_tb/L0/counterResult
0	+0	U	U	U	U	UUU
0	+1	1	0	0	0	UUU
0	+3	1	0	0	0	000
5000000	+1	1	0	1	0	000
10000000	+1	0	0	0	1	000
15000000	+1	0	0	1	1	000
20000000	+1	0	1	0	1	000
25000000	+1	0	1	1	1	000
25000000	+3	0	1	1	1	001
30000000	+1	0	1	0	1	001
35000000	+1	0	1	1	1	001
35000000	+3	0	1	1	1	010
40000000	+1	0	1	0	1	010
45000000	+1	0	1	1	1	010
45000000	+3	0	1	1	1	011
50000000	+1	0	1	0	1	011
55000000	+1	0	1	1	1	011
55000000	+3	0	1	1	1	101
60000000	+1	0	1	0	1	101
65000000	+1	0	1	1	1	101
65000000	+3	0	1	1	1	110
70000000	+1	0	1	0	1	110
75000000	+1	0	1	1	1	110
75000000	+3	0	1	1	1	111
80000000	+1	0	1	0	1	111
85000000	+1	0	1	1	1	111
90000000	+1	0	1	0	1	111
95000000	+1	0	1	1	1	111
100000000	+1	0	1	0	1	111
105000000	+1	0	1	1	1	111
110000000	+1	0	1	0	1	111
115000000	+1	0	1	1	1	111
120000000	+1	0	1	0	1	111
125000000	+1	0	1	1	1	111
130000000	+1	1	1	0	1	111
130000000	+3	1	1	0	1	000
135000000	+1	1	1	1	1	000
140000000	+1	1	1	0	1	000
145000000	+1	1	1	1	1	000

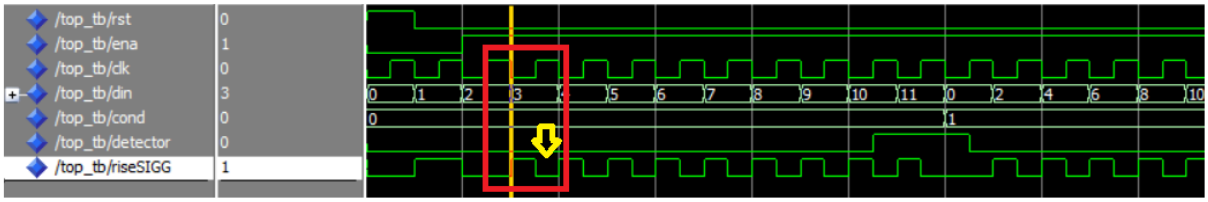
Condition process:

מודול זה בעצם לוקח שתי סיגנלים d[i] וגם d[i-1] ובודק האם הם מקיימים את cond . אם כן אזי RISE עולה ל1 אחרת 0.

דיאגרמה נקייה עבור :

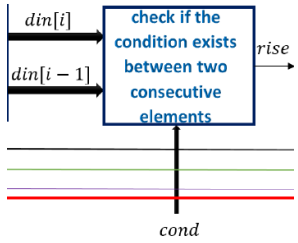


דיאגרמה עם הסבר:

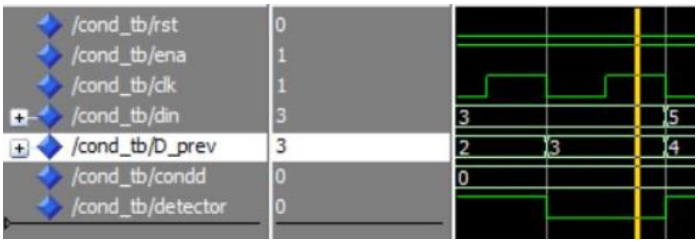


אנו עובדים בלוגיקה של עליית שעון לכן אנו דאגנו שסיגנל של RISE יהיה תקין לקראת דגימה של חלק הcounter.במלבן האדום ניתן לראות שRISE שווה לקראת עליית השעון מפני ש d_i=3 וd_i-1=2 וCOND בזמן הזה שווה ל0 וזה בדיוק מקיים את התנאי לכן RISE שווה ל1, לאחר עליית שעון חלק של דילאי מוציא 3 ואז כבר תנאי לא מתקיים לכן RISE יורד ל0.

איור המודול:



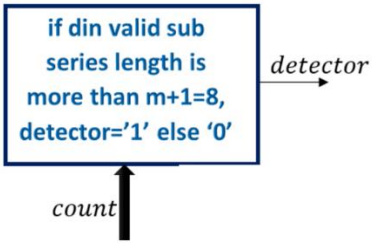
דיאגרמה של תת מודול מחוץ למערכת:



ניתן לראות שבאשר תנאי מתקיים DETECTOR עולה ל1 אחרת0

Detector:

מודול זה בעצם לוקח את כמות הפעמים שתתי-הסדרות קיימו את התנאי הרצוי, ומוציא '1' אם כל הסדרה תקינה – כלומר אם אורך הסדרה הוא m+1=8 אחרת 0.החלטנו לממש את זה לא בPROCESS. איור המודל:



דיאגרמה של תת מודול בנפרד מכל המערכת: