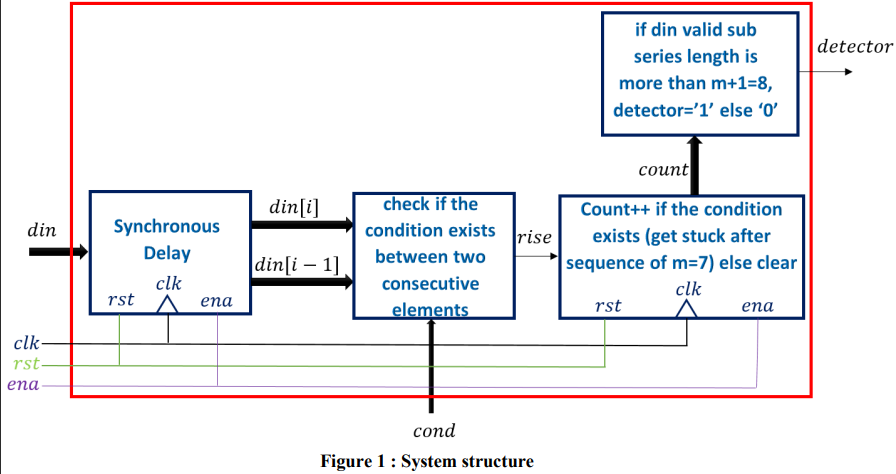
**top.vhd:**

בקובץ הזה נמצא Entity בשם top בעל הכניסות והמוצאים הבאים:

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

המודול מכיל תת מודולים שנמצאים בקבצים: Adder.vhd, Counter.vhd, detector\_val.vhd, SynchronousDelay.vhd.

על-כן, הוא מכיל את ה-processes הבאים: updateCondProcess, sProcess.

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.

sProcess:  
process זה אחראי לבדיקת שינוי בערך של המוצא של מודול Adder (adderS). כל פעם שיש שינוי בערך הסיגנל adderS, אנחנו בודקים אם adderSהעדכני 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'.  
הסבר על trigger

דיאגרמה נקייה:

דיאגרמה עם הסבר:

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

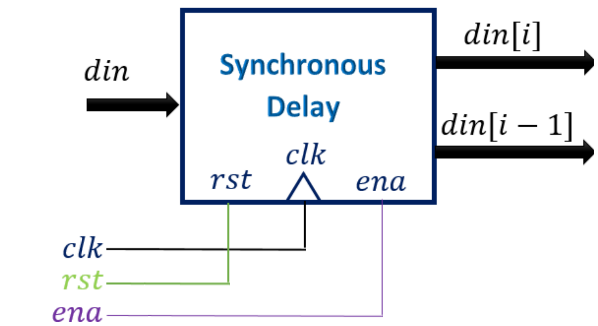
**Adder.vhd:**

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.

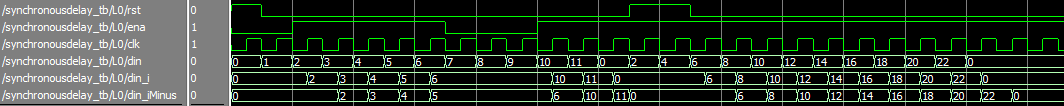
**SynchronousDelay.vhd:**

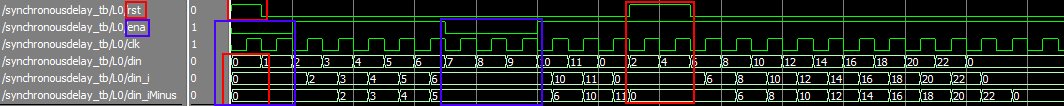
מודול זה אחראי על דגימת ערכי ווקטור הכניסה din ולאחר מכן שמירת הערך הנוכחי והערך הקודם. הדגימה נעשית בכל עליית שעון וכל עוד ena=’1’. כאשר rst=’1’ ערכי din[i] ו din[i-1]מתאפסים.  
פירוט הכניסות והמוצאים של המודול:

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



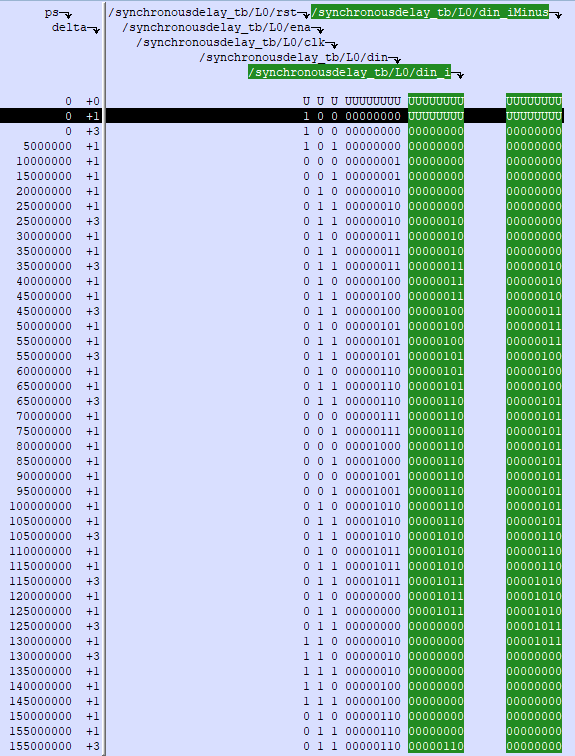
איור המודול:

דיאגרמה נקייה:

דיאגרמה עם הסבר:

המסומן באדום: ניתן לראות שברגע ש rst=’1’המוצאים ישר מתאפסים, ללא תלות ב-clk.  
המסומן בכחול: ניתן לראות שברגע ש ena=’0’המוצאים שומרים על הערך שהיה בהם לפני השינוי של ביט ena. אחרי שהוא חוזר להיות ‘1’ המערכת ממשיכה לפעול כרגיל.

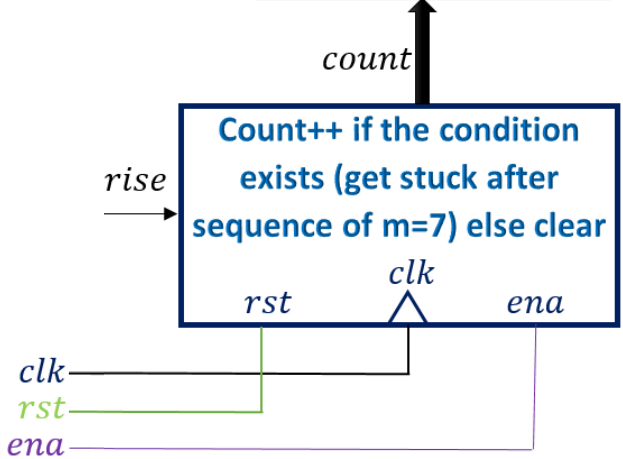
דיאגרמת ה-LIST כדי שנוכל לראות את טבלת האמת של גרף ה-wave:



**Counter.vhd:**

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:

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

איור המודול:

דיאגרמה נקייה:

דיאגרמה עם הסבר:

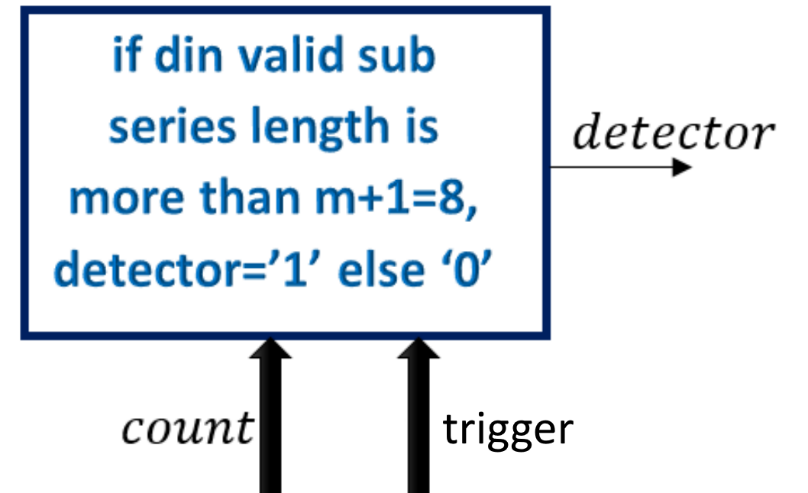
דיאגרמת ה-LIST כדי שנוכל לראות את טבלת האמת של גרף ה-wave:

**Detector\_val.vhd:**

מודול זה בעצם לוקח את כמות הפעמים שתתי-הסדרות קיימו את התנאי הרצוי, ומוציא '1' אם כל הסדרה תקינה.  
פירוט הכניסות והמוצאים של המודול:

|  |  |  |
| --- | --- | --- |
| הסבר | סוג | שם הפורט |
| ווקטור כניסה שמכיל כמה פעמים תת-הסדרה התקיימה | Input bit | Trigger |
| ווקטור כניסה שמכיל כמה פעמים תת-הסדרה התקיימה | Input bit | counterResult |
| סיגנל אשר בודק אם כל הסדרה תקינה | Output bit | isOne |

איור המודול:



דיאגרמה נקייה:

דיאגרמה עם הסבר:

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