

## LAB2 - VHDL

נפרט את כל אחד מהתהליכים אותם מימשנו ואת מוצא ה – testbench עבור כל אחד מהם:

Process 1: בתהליך זה מימשנו השהייה של מחזור שעון אחד, ושל שני מחזורי שעון עבור אות הכניסה של המע':  $X$  (וקטור בינארי באורך  $n$ ), כאשר תתי הסדרות המושהות – מועברות לתהליך 2.

מימשנו תהליך זה באמצעות התנייה על קווי הבקרה ( reset, clock, enable ), כך שבהנתן '0' = enable – לא תתבצע השהייה והאותות במוצא התהליך ישארו כבמחזור השעון הקודם.

אחרת – אם – '1' – reset נאפס את האותות במוצא

ואחרת – בעליית שעון – תתבצע ההשמה של  $x[j-1] := x[j] \& x[j-2]$   $x[i-1] := x[j]$  כך שנקבל את התתי הסדרות הנדרשות לתהליך 2.

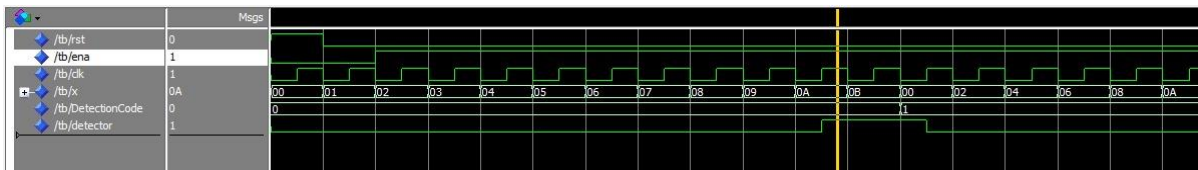
Process 2: בתהליך זה מימשנו פעולת חיסור בין 2 תתי הסדרות (ע"י שימוש ברכיב Adder ומימוש החיסור ע"י חיבור של אחת הסדרות בשיטת המשלים לאחד) ובדקנו אם ההפרש (ייצוגו הדצימלי של תוצאות החיסור בין הסדרות) – מתאים לקוד המתקבל לתהליך – במידה כן – יועבר '1' לוגי לתהליך 3 – אחרת '0' (תחת סיגנל שקראנו לו valid).

Process 3: בתהליך זה תחזקנו מונה, המונה את מספר הפעמים ברצף שהתקבל valid bit חיובי, כלומר '1'.

בכל פעם שהתקבל valid bit שלילי, איפסנו את המונה ונתחיל למנות מחדש.

בהנתן valid bit חיובי  $m$  פעמים ברצף – יתקבל '1' במוצא המערכת, אחרת '0'.

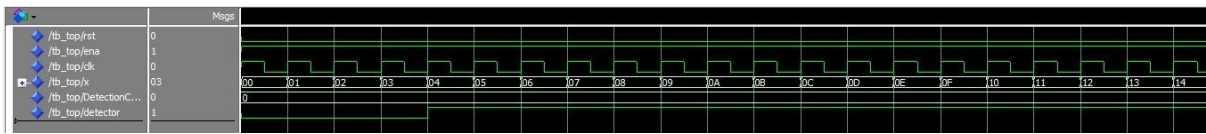
להלן דיאגרמת wave וlist של ביצוע testbench למודול:



ps	delta	/tb/rst	/tb/ena	/tb/clk	/tb/DetectionCode	/tb/x	/tb/detector
0	+1	1	0	0	00000000	0	0
50000	+1	1	0	1	00000000	0	0
100000	+1	0	0	0	00000001	0	0
150000	+1	0	0	1	00000001	0	0
200000	+1	0	1	0	00000010	0	0
250000	+1	0	1	1	00000010	0	0
300000	+1	0	1	0	00000011	0	0
350000	+1	0	1	1	00000011	0	0
400000	+1	0	1	0	00000100	0	0
450000	+1	0	1	1	00000100	0	0
500000	+1	0	1	0	00000101	0	0
550000	+1	0	1	1	00000101	0	0
600000	+1	0	1	0	00000110	0	0
650000	+1	0	1	1	00000110	0	0
700000	+1	0	1	0	00000111	0	0
750000	+1	0	1	1	00000111	0	0
800000	+1	0	1	0	00001000	0	0
850000	+1	0	1	1	00001000	0	0
900000	+1	0	1	0	00001001	0	0
950000	+1	0	1	1	00001001	0	0
1000000	+1	0	1	0	00001010	0	0
1050000	+2	0	1	1	00001010	0	1
1100000	+1	0	1	0	00001011	0	1
1150000	+1	0	1	1	00001011	0	1
1200000	+1	0	1	0	00000000	1	1
1250000	+2	0	1	1	00000000	1	0
1300000	+1	0	1	0	00000010	1	0
1350000	+1	0	1	1	00000010	1	0
1400000	+1	0	1	0	00000100	1	0
1450000	+1	0	1	1	00000100	1	0

בדוגמא זו ניתן לראות כי עבור  $m = 7$  ועבור  $\text{detectionCode} = 0$  למשך  $1\mu s$ , נקבל לבסוף, בהתאם לקונפיגורציה של קוי הבקרה '1'  $\text{detector} = 1$  אחרי 8  $(m+1)$  מחזורי שעון (החל מ'1'  $\text{enable} = 1$ ).

דוגמא נוספת,



ps	delta	/tb_top/rst	/tb_top/x	/tb_top/ena	/tb_top/clk	/tb_top/DetectionCode	/tb_top/detector
0	+1	0	1	1	00000000	0	0
50000	+1	0	1	0	00000000	0	0
100000	+1	0	1	1	00000001	0	0
150000	+1	0	1	0	00000001	0	0
200000	+1	0	1	1	00000010	0	0
250000	+1	0	1	0	00000010	0	0
300000	+1	0	1	1	00000011	0	0
350000	+1	0	1	0	00000011	0	0
400000	+2	0	1	1	00000100	0	1
450000	+1	0	1	0	00000100	0	1
500000	+1	0	1	1	00000101	0	1
550000	+1	0	1	0	00000101	0	1
600000	+1	0	1	1	00000110	0	1
650000	+1	0	1	0	00000110	0	1
700000	+1	0	1	1	00000111	0	1
750000	+1	0	1	0	00000111	0	1
800000	+1	0	1	1	00001000	0	1

עבור סימולציה זו,  $m = 2$  ו  $detectionCode = 0$  ניתן לראות כי אכן לאחר  $3(m+1)$  מחזורי שעון, בהם ההפרש בין שני זמנים הוא 0, קיבלנו 1 ב `detector`.

נשים לב, כי בסימולציה זו, השורה הראשונה לא נקראת מאחר והיא מתארת איטרציה ב `falling edge`, בעוד המע' קוראת את הערכים רק ב `rising edge`.