


סימולציה 1

		
		<u>דניאל טרדלר</u>

חלק יבש:

2.

2.1. בהרצאה ראינו כי פונקציה של בורר היא: $z = s' * d0 + s * d1$

ולכן טבלת אמת:

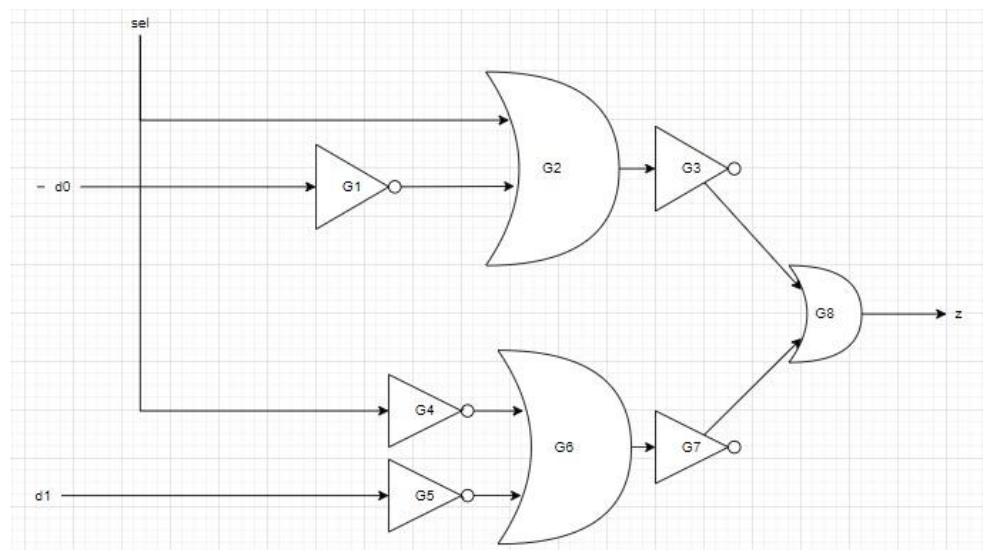
d0	d1	sel	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

אבל מכיוון שעלינו לעבוד ללא שערי AND, החלפנו כל שער AND ב: $(X' + Y)'$

כעת נראה ש: $X * Y = (X' + Y)'$

X	Y	$X \cap Y$	$X' + Y'$	$(X' + Y)'$
0	0	0	1	0
1	0	0	1	0
1	1	1	0	1
0	1	0	1	0

הפונקציה שנקבל היא: $Z = (s + d0')' + (s' + d1)'$



ערכים עבור השהיות:

	Tpd _{lh}	Tpd _{hl}
NOT	10	5
OR2	8	1
XOR2	1	6

שינוי בכניסה של d0:

Path	d0	d1	sel	g1	g2	g3	g8	Tpd
d0->g1->g2->g3->g8->z	0->1	0	0	1->0	1->0	0->1	0->1	5+1+10+8=24
	1->0	0	0	0->1	0->1	1->0	1->0	10+8+5+1=24
d0->g1->g2->g3->g8->z	0->1	1	0	1->0	1->0	0->1	0->1	5+1+10+8=24
	1->0	1	0	0->1	0->1	1->0	1->0	10+8+5+1=24

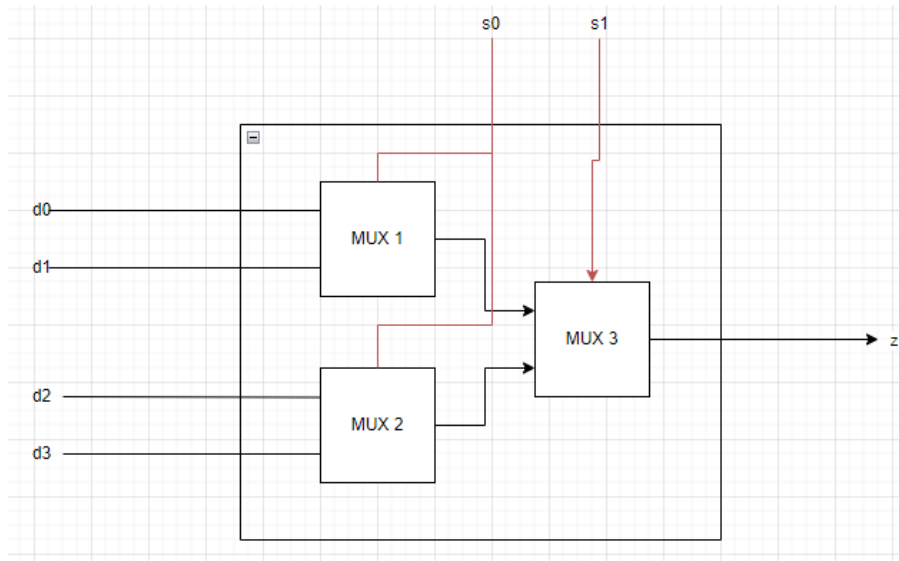
שינוי בכניסה של d1:

Path	d0	d1	sel	g5	g6	g7	g8	Tpd
d1->g5->g6->g7->g8->z	0	0->1	1	1->0	1->0	0->1	0->1	5+1+10+8=24
	0	1->0	1	0->1	0->1	1->0	1->0	10+8+5+1=24
d1->g5->g6->g7->g8->z	1	0->1	1	1->0	1->0	0->1	0->1	5+1+10+8=24
	1	1->0	1	0->1	0->1	1->0	1->0	10+8+5+1=24

שינוי בכניסה של sel: (ה-tpd מחושב על סמך המסלול הארוך מבין השניים)

Path	d0	d1	sel	g4	g6	g7	g8	Tpd
sel->g4->g6->g7->g8->z	0	1	0->1	1->0	1->0	0->1	0->1	5+1+10+8=24
	0	1	1->0	0->1	0->1	1->0	1->0	5+8+10+1=24
sel->g4->g6->g7->g8->z	1	0	0->1	1->0	1->0	0->1	1->1	5+1+10=16
	1	0	1->0	0->1	0->1	1->0	1->1	10+8+5=23

2.2. נבנה מימוש לmux 4-1 ע"י שימוש בmux שבנינו בסעיף הקודם בלבד.

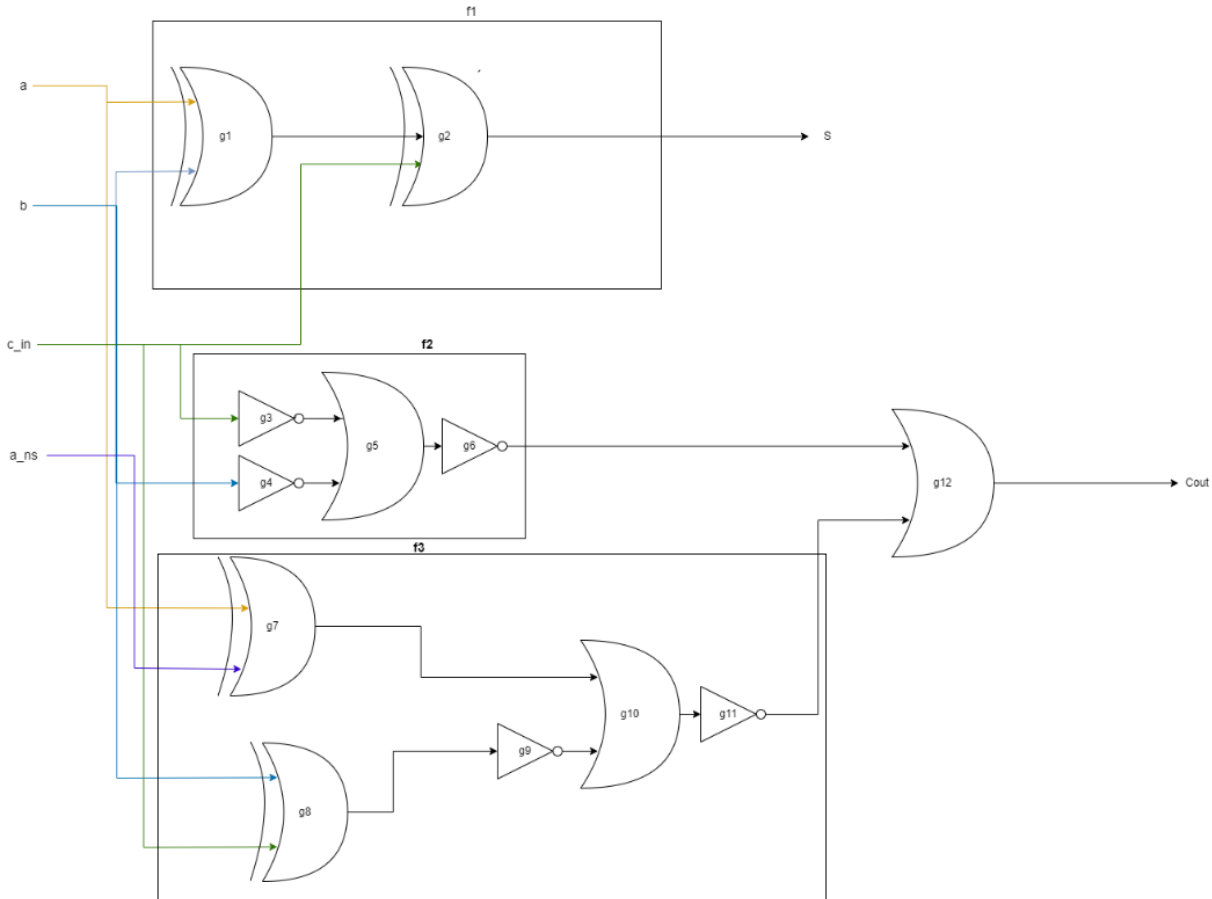


נשים לב ש[0]s קובע את LSB – הוא בוחר בכל MUX על איזה כניסה נסתכל
 ו[1]s קובע את MSB – כלומר קובע לMUX3 איזה תוצאה מmux1 וmux2 לקחת.

Path	d0	d1	d2	d3	s0	s1	Tpd
d0->mux1->mux3->z	0->1	0	0	0	0	0	$2(5+1+10+8)=48$
d0->mux1->mux3->z	1->0	0	0	0	0	0	$2(10+8+5+1)=48$

	t _{PDLH}	t _{PDHL}
Not	10	10
OR	8	8
XOR	6	6

2.3. הציעו מימוש ליחידת Full Adder/Subtractor



2.3.1. נציין כמו בסעיפים הקודמים ש- $X*Y = (X'+Y)'$ מהווה תחליף לAND

2.3.2. נבנה טבלאות קרנו:

1. $C_{out} = B * CIN + (AXNORA_NS) * (BXORCIN)$

2. $S = CIN XOR AXOR B$

		AB			
CIN,ANS		00	01	11	10
	00	0	1	0	0
	01	0	0	1	0
	11	0	1	1	1
	10	1	1	1	0

		AB			
CIN,ANS		00	01	11	10
	00	0	1	0	1
	01	0	1	0	1
	11	1	0	1	0
	10	1	0	1	0

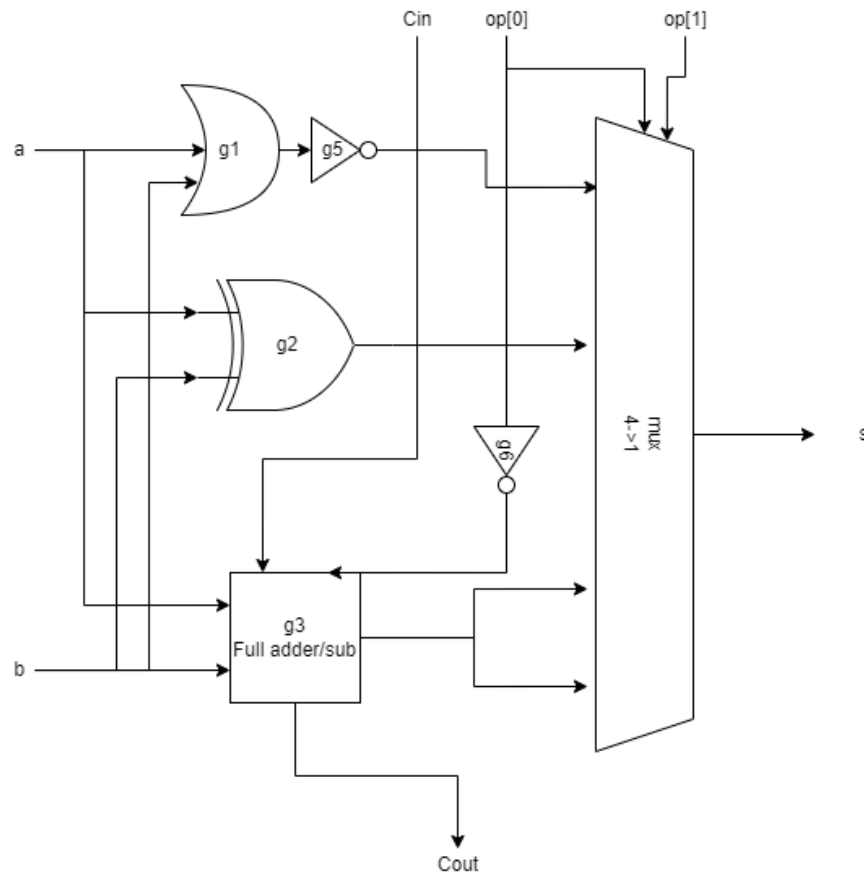
2.3.3. טבלת אמת

a	b	cin	a_ns	f1	s	f2	f3	g7	c_out
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0
0	1	0	0	1	1	0	1	1	1
1	1	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	1
1	0	1	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1	1
1	1	1	0	1	1	1	0	1	1
0	0	0	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	0
0	1	0	1	1	1	0	0	0	0
1	1	0	1	0	0	0	1	1	1
0	0	1	1	1	1	0	0	0	0
1	0	1	1	0	0	0	1	1	1
0	1	1	1	0	0	1	0	1	1
1	1	1	1	1	1	1	1	1	1

2.3.4. נחשב את ההשהיות המקסימליות מכל כניסה ליציאה:

- $Tpd(a \rightarrow s): tpd(xor) + tpd(xor) = 6 + 6 = 12$
- $Tpd(a \rightarrow c_out): tpd(xor) + tpd(or) + tpd(not) + tpd(or) = 32$
- $Tpd(b \rightarrow s): tpd(xor) + tpd(xor) = 6 + 6 = 12$
- $Tpd(b \rightarrow c_out): tpd(xor) + tpd(not) + tpd(or) + tpd(not) + tpd(or) = 42$
- $Tpd(cin \rightarrow s): tpd(xor) = 6$
- $Tpd(cin \rightarrow c_out): tpd(xor) + tpd(not) + tpd(or) + tpd(not) + tpd(or) = 42$
- $Tpd(a_ns \rightarrow s): 0$
- $Tpd(a_ns \rightarrow c_out): tpd(xor) + tpd(or) + tpd(not) + tpd(or) = 32$

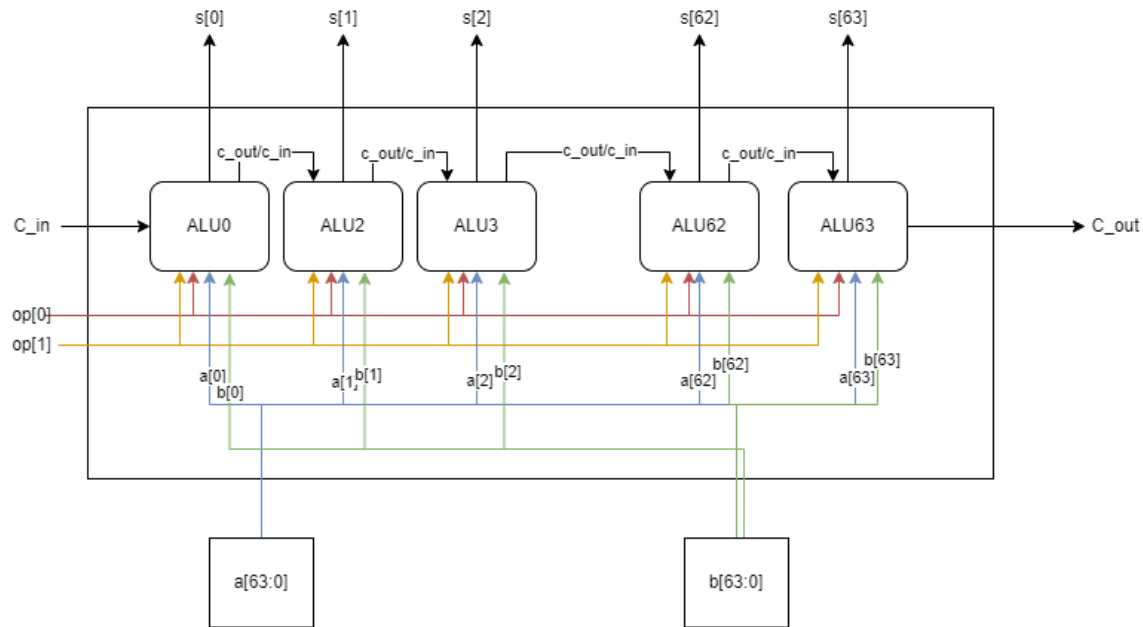
2.4. המימוש עבור ALU:



ניתן להבחין כי עבור כניסות הבקרה $op[0]$ ו- $op[1]$ יש השפעה על full adder/sub. $g[0]$ אחראי על האם full adder/sub יבצע חיבור או חיסור, ולכן קיים פיצול. בחרנו לעבוד עם בבורר $4 \rightarrow 1$. הכנסנו לכניסה השלישית והרביעית את תוצאת $g3$ על מנת שאם $op[1]$ יהיה שווה ל-1 אזי הבורר צריך להעביר את תוצאת החיבור/ החיסור, בהתאם ל- $op[0]$ שדאג לבצע את החיבור/חיסור שלב קודם לכן. ההשהיה המקסימלית עבור בורר $4 \rightarrow 1$ היא: ps48. (בורר $4 \rightarrow 1$ בנוי מ-3 בוררים $2 \rightarrow 1$, קיים מסלול שעבורו זמן ההשהיה המקסימלי הוא 48). ההשהיות המקסימלית מכל כניסה אל כל יציאה:

- $Tpd(a \rightarrow s): tpd(full\ adder/sub) + tpd(mux) = 12 + 48 = 60$
- $Tpd(a \rightarrow c_out): tpd(full\ adder/sub) = 32$
- $Tpd(b \rightarrow s): tpd(or) + tpd(not) + tpd(mux) = 10 + 8 + 48 = 66$
- $Tpd(b \rightarrow c_out): tpd(full\ adder/sub) = 42$
- $Tpd(cin \rightarrow s): tpd(full\ adder/sub) + tpd(mux) = 6 + 48 = 60$
- $Tpd(cin \rightarrow c_out): tpd(full\ adder/sub) = 42$
- $Tpd(op[0] \rightarrow s): tpd(not) + tpd(full\ adder/sub) + tpd(mux) = 10 + 48 = 58$
- $Tpd(op[0] \rightarrow c_out): tpd(not) + tpd(full\ adder/sub) = 10 + 32 = 42$
- $Tpd(op[1] \rightarrow s): tpd(mux) = 48$
- $Tpd(op[1] \rightarrow c_out): 0$

2.5. המימוש יהיה זהה למימוש בסעיף למעלה רק נשנה את מספר הכניסה ונחבר את הרכיבי ALU כמו שראינו בהרצאות:



נחשוב על הסיטואציה ששינוי בביט בודד יעבור את כל רכיבים הALU שלנו.

או במילים אחרות שפעולה בALU הראשון $a[0]$ תשפיע על כל שאר הALU ותבחר את הדרך הארוכה ביותר דרכם.

נסמן $a=111...1$ ו $b=000...0$ (64 סיביות כל אחד) וכאשר נשנה את הסיבית של $b[0]$ ל-1 זה יגרום להשהיה מקסימלית מכניסה ליציאה.

\leq נבחר $op[0]=0$ ו $op[1]=1$ (חיבור לפי הטבלה)

\leq $c_{in}=1 \Rightarrow c_{out}=1$ של ALU1 וALU1 נקבל את אותה תוצאה הלאה, עד לALU האחרון.

\leq $s[i]$ בהתאם לכל ALUi שנהיה, ישתנה מ-1 ל-0

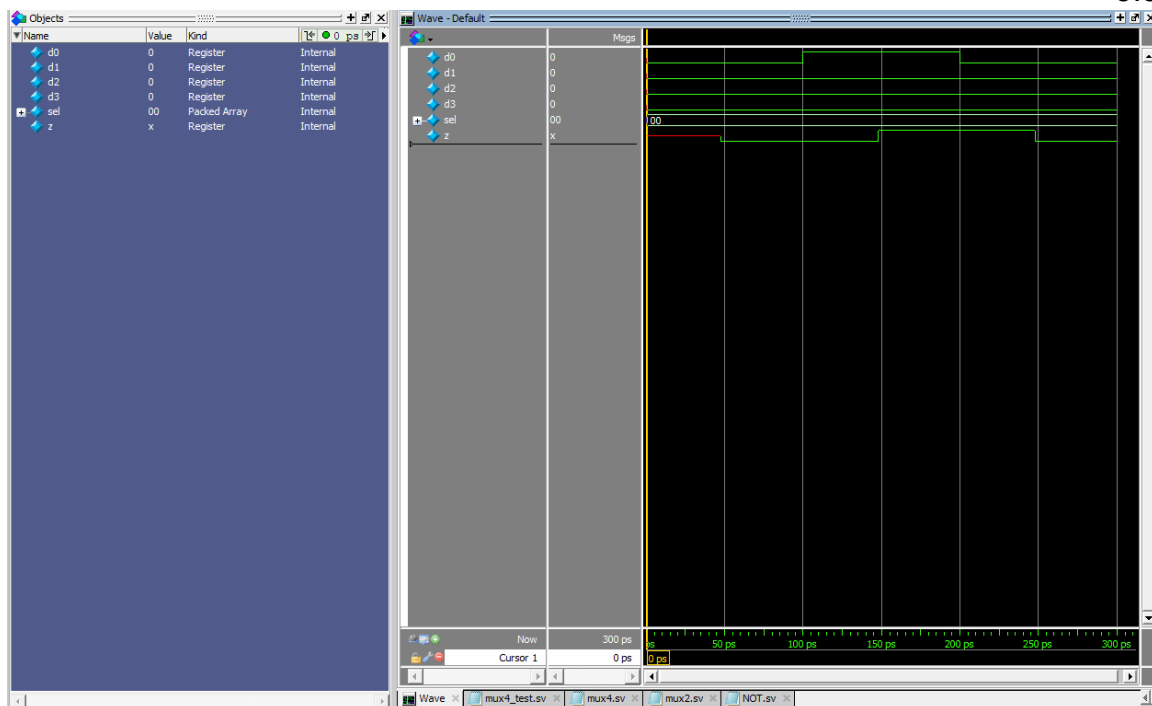
וכמובן בסוף בסוף ניקח את היציאה $Tpd(c_{in} \rightarrow s[63])$ שהיא יותר ארוכה מ $Tpd(c_{in} \rightarrow c_{out})$ לכן נחשב:

$$Tpd(a[0] \rightarrow c_{out}) + 62 * Tpd(c_{in} \rightarrow c_{out}) + Tpd(c_{in} \rightarrow s[63]) =$$

נקבל ש: שהשהיה המקסימלית שווה ל :

$$(tpd(b[0] \rightarrow cout) + 62 * tpd(cin \rightarrow cout) + tpd(cin \rightarrow s[63])) = 42 + 42 * 62 + 48 = 2700$$

-3.3



```
initial begin
    d0= 0;
    d1= 0;
    d2=0;
    d3=0;
    sel[0]=0;
    sel[1]=0;
```

הכנסנו את הנתונים הבאים :

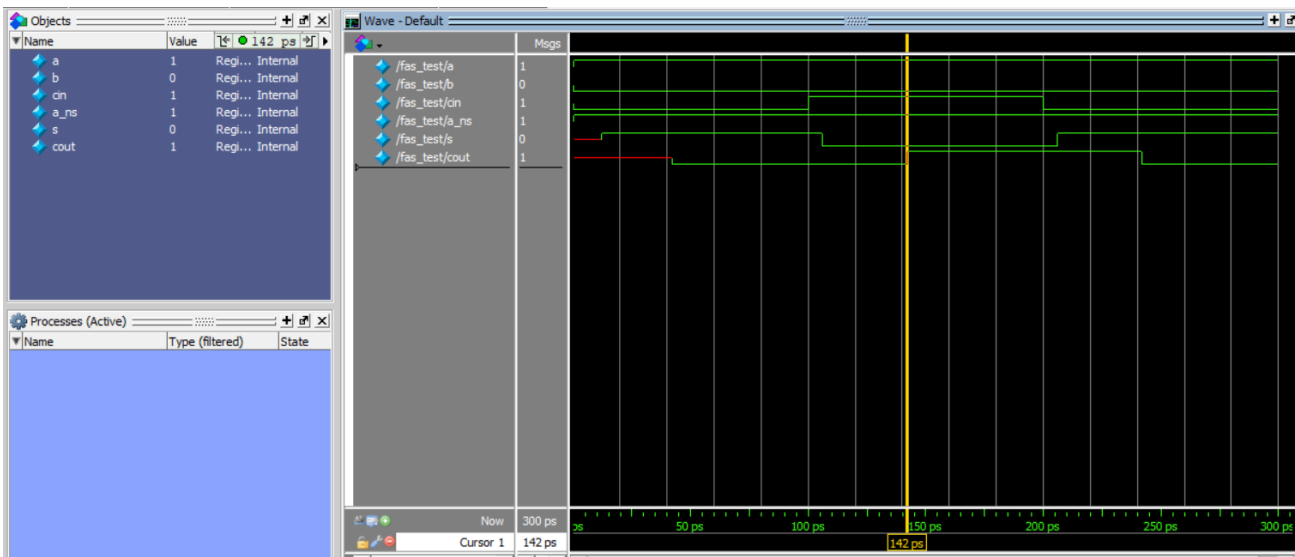
כאשר z אמור לייצג את הערך של d0 (זאת אומרת שהבורר מראה כפלט את ערכו של d0).

ניתן לראות שעד זמן 100 הערך של d0 הוא 0 (כמצופה)

לאחר שינוי של d0 ל1, לקח בדיוק 48 זמן(כל שינוי בנתיב הזה בכל mux2 הוא בדיוק 24 שניות, כמו שרשום בסעיפים למעלה) עד שקיבלנו את השינוי ביציאה z.

לאחר 100 זמן נוספים שינוי את d0 ל1, וכמו מקודם בהתאם לטבלה שלנו, קיבלנו אחרי 48 שניות שינוי בז .

3.4.



```
#100          initial begin
a=1;          a=1;
b=0;          b=0;
cin=1;        cin=0;
a_ns=1;       a_ns=1;
```

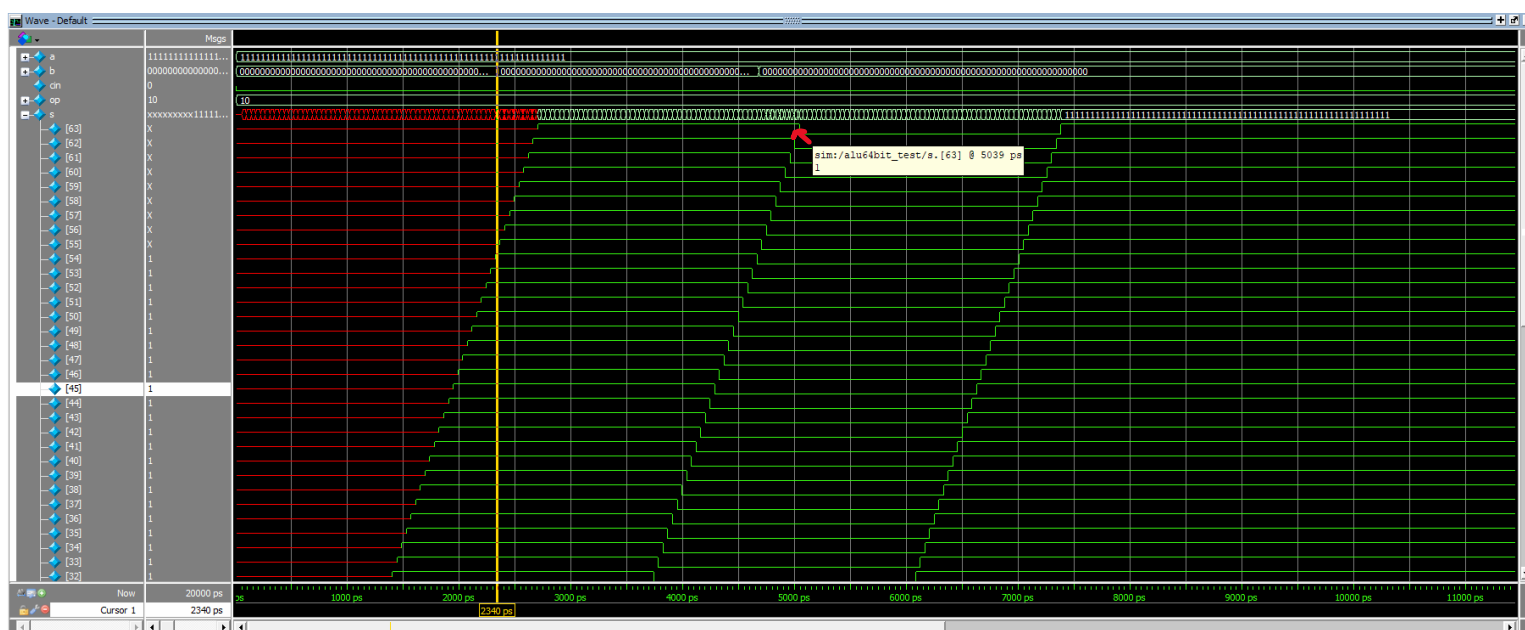
והשינוי הוא:

הכנסנו את הנתונים הבאים:

$a_ns = 1$ מייצג את פעולת החיבור.

לאחר השינוי הערך של S אמור להיות 0 כי פעולת XOR עבור $a=1$ ו- $b=0$ נותנת $w=1$ יחד עם $Cin=1$ ב-XOR התוצאה תהיה 0.

הערך של Cout אמור להיות 1 לאחר התייצבות, כי בפעולת החיבור כגון במסלול זה ($a_ns=1$) עבור $a=1$ ו- $b=0$, $Cin=1$ אמורה להתקבל שארית 1 כמצופה לפי החישובים שלנו, וכן ניתן לראות שלאחר 100 ps השינוי מתרחש ולאחר 42 ps ישנה התייצבות עם כל התוצאות המצופות הנ"ל. השהיה המקסימלית היא 42 ps והיא מתרחשת במסלול מ $Cout \rightarrow b$.



#2340

```

a  = {64{1'b1}} ;      a = {64{1'b1}} ;
b  = 64'h1;            b  = 64'h0;
op = 2'b10;             op = 2'b10;
cin = 1'b0;             cin = 1'b0;

```

והשינוי

הכנסנו את הנתונים הבאים :

נשים לב הסמן (הצהוב) עומד על 2340, זהו הרגע שאני משנה את $b[0]$ לערך 1.

מהרגע הזה ניתן להבחין שיש שינוי באיברי $s[i]$ (מתחלפים מ1 ל0) כמצופה.

כמו בחישוב התיאורטי אנחנו מצפים שמרגע השינוי עד ליציאה $s[63]$ יעבור 2700 זמן.

ניתן לראות בריבוע הלבן (שעומד על נקודת השינוי **בערך**) שהוא עומד על 5040, כמצופה.

בתמונה הבאה ניתן לראות גם את השינוי בcout שמשנה את ערכו רק אחרי $s[62]$ אבל קצת לפני הפלט של $s[63]$, ניתן לראות בזמנים:

