סימולציה 1:

מגישים:

עידו טאוסי	214008997
נעם ביטון	213745953

סעיף 2.1

טבלת האמת המתאימה לבורר הנ"ל:

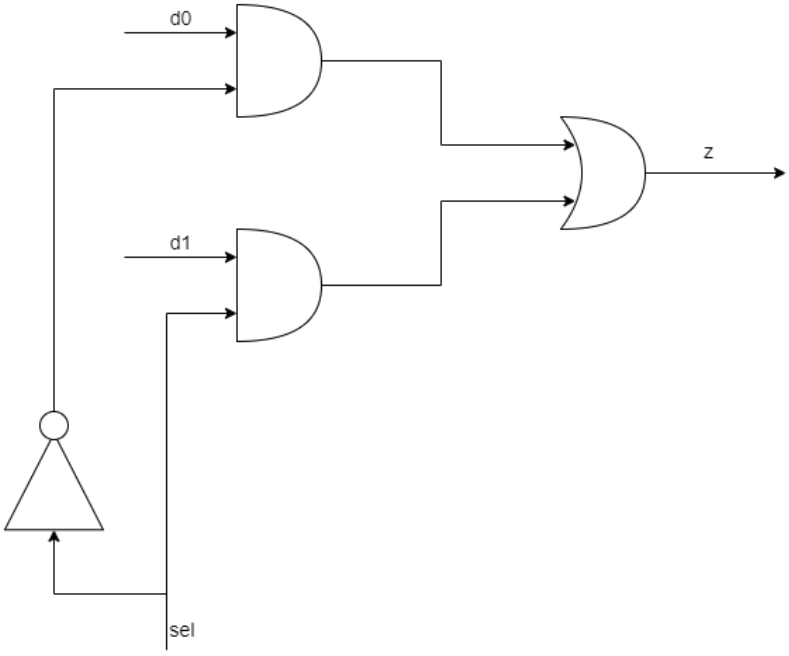
d_0	d_1	sel	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

כדי למצוא את z כפונקציה של הכניסות נעזר במפת קרנו המתאימה לטבלת האמת שהצגנו:

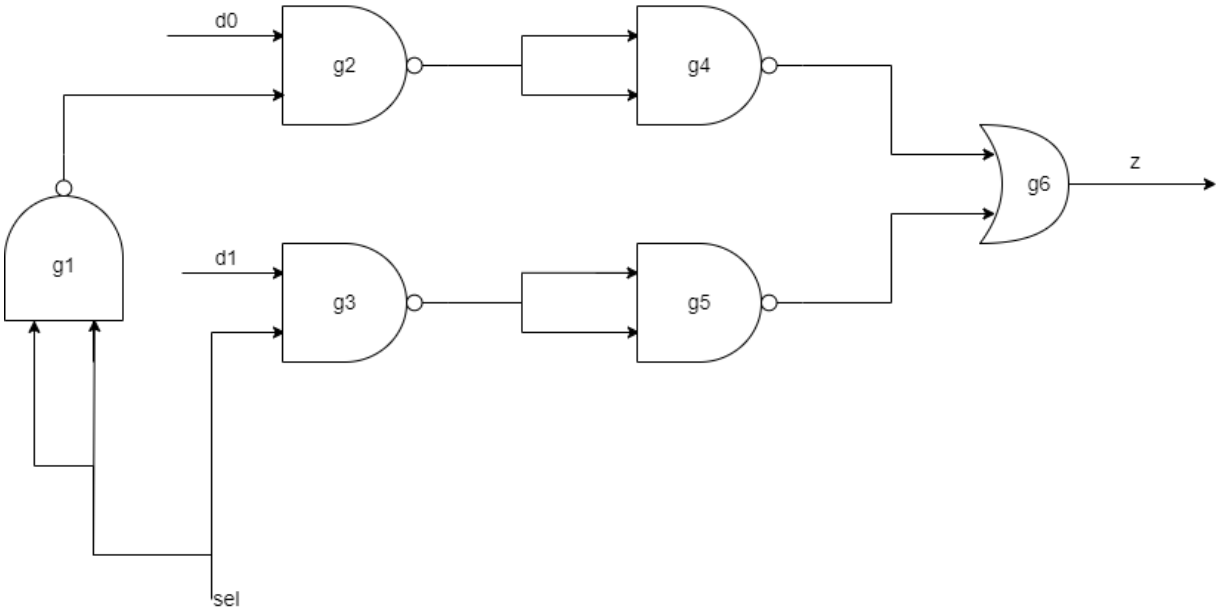
sel \ d_0d_1	00	01	11	10
0			1	1
1		1	1	

לכן הפונקציה המצומצמת של z לפי מפת קרנו היא: $z(d_0, d_1, sel) = d_0 \cdot \overline{sel} + d_1 \cdot sel$

לכן המימוש של הבורר הרגיל, ללא הגבלה על שימוש בשערים הוא:



כעת נשתמש בשער NAND על מנת להמיר שער AND ו- NOT. ידוע כי כאשר נותנים את אותו ערך בשתי הכניסות של ה-NAND, מקבלים שער NOT, ובנוסף ניתן לשרשר שערי NAND ו-NOT שיצרנו, על מנת לקבל מימוש של שער AND, לכן המימוש שלנו לבורר 2 ל-1 הוא:



נחשב בטבלה הבאה את כל ההשהיות של כל המסלולים מכל הכניסות למוצא z בטבלה הבאה:

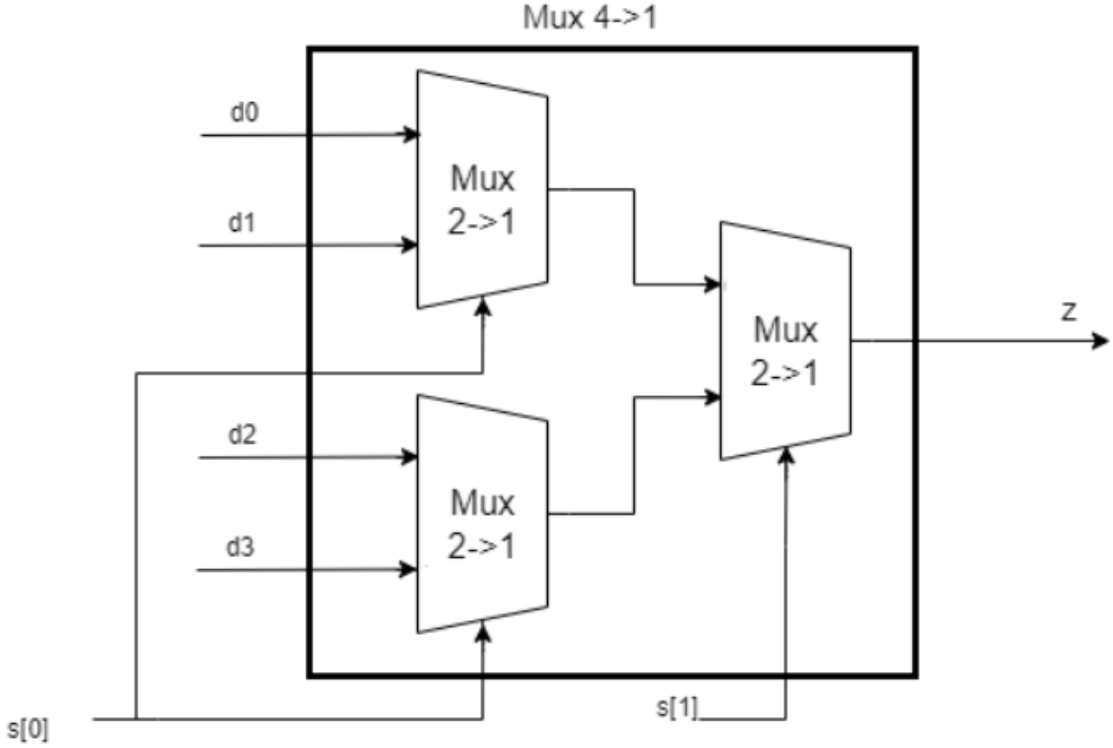
path	$\mathbf{d_0}$	$\mathbf{d_1}$	**sel**	**Tpd**

d_0→g2→g4→g6→z	0→1	0	0	$E+B+C=15$
d_0→g2→g4→g6→z	1→0	0	0	$B+E+F=19$
d_0→g2→g4→g6→z	0→1	1	0	$E+B+C=15$
d_0→g2→g4→g6→z	1→0	1	0	$B+E+F=19$
d_1→g3→g5→g6→z	0	0→1	1	$B+E+C=15$
d_1→g3→g5→g6→z	0	1→0	1	$E+B+F=19$
d_1→g3→g5→g6→z	1	0→1	1	$E+B+C=15$
d_1→g3→g5→g6→z	1	1→0	1	$B+E+F=19$
sel→g3→g5→g6→z	0	1	0→1	$E+B+C=15$
sel→g3→g5→g6→z	0	1	1→0	$E+B+F=19$
sel→g1→g2→g4→g6→z	1	0	0→1	$3E+F=38$
sel→g1→g2→g4→g6→z	1	0	1→0	$2B+E+C$=16

בחישוב ההשהיות התייחסנו רק למסלולים עבורם שינוי הכניסה ישנה את המוצא בסוף המסלול,

ואת המוצא הכולל של המערכת, ולכן לא כל האפשרויות מופיעות בטבלה.

סעיף 2.2



נבחר לשנות את d_0 ונבחר את המצב הקבוע הבא:

$$d_1 = 0,\ d_2 = 0,\ d_3 = 0$$

$$sel = 00 \ \Rightarrow sel[0] = 0,\ sel[1] = 0$$

$$0 \to 1:\ t_{pd,\,LH} = t_{pd,\,LH}\left(Mux\ \ 2 \to 1\right) + t_{pd,\,LH}\left(Mux\ \ 2 \to 1\right) =$$

$$2*(\ t_{pd}(NAND) + t_{pd}(NAND) + t_{pd}(OR)) = 2*(2*max\{B,E\} + max\{C,F\}) = 2*(2*10+8) = 56$$

$$1 \to 0:\ t_{pd,\,HL} = t_{pd,\,HL}\left(Mux\ \ 2 \to 1\right) + t_{pd,\,HL}\left(Mux\ \ 2 \to 1\right) =$$

$$2*(\ t_{pd}(NAND) + t_{pd}(NAND) + t_{pd}(OR)) = 2*(2*max\{B,E\} + max\{C,F\}) = 2*(2*10+8) = 56$$

סעיף 2.3:

נבנה מפת קרנו עבור cout:

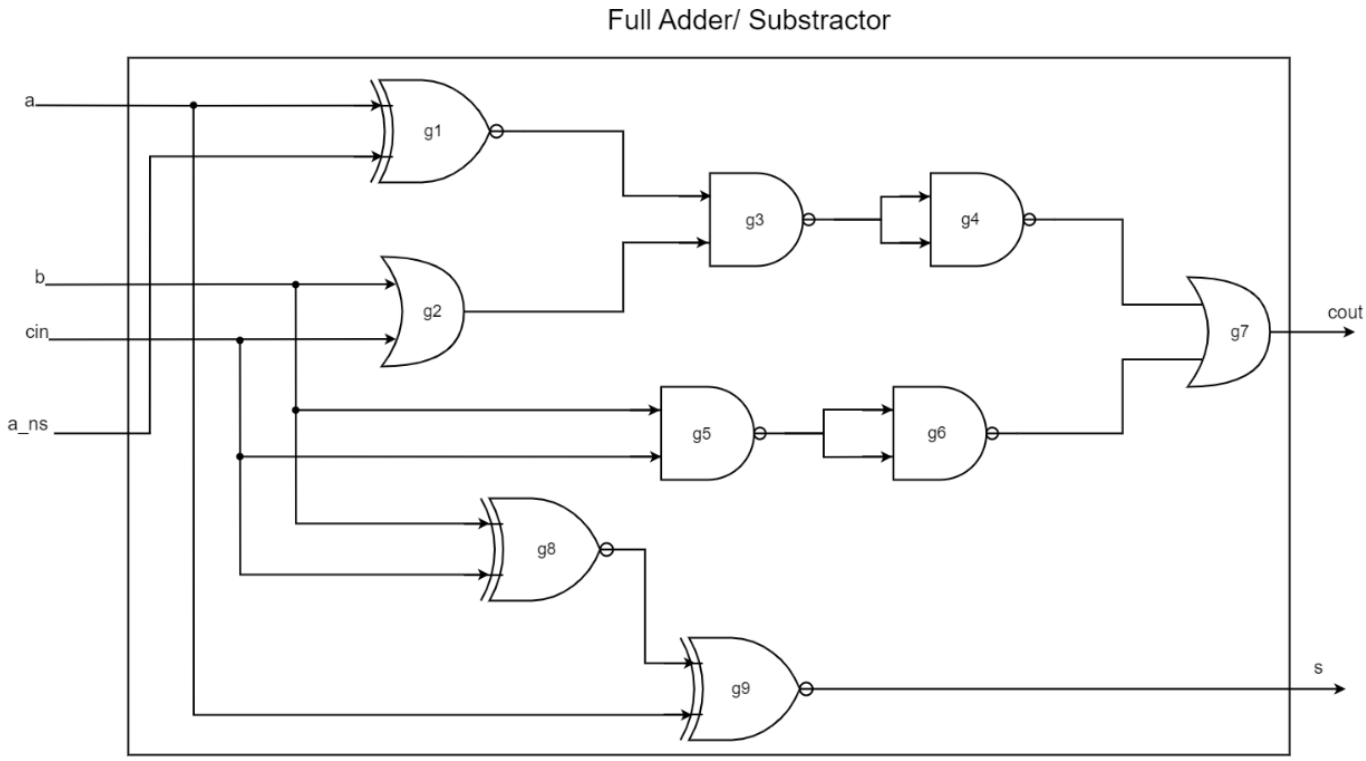
a b / cin a_ns	00	01	11	10
00		1		
01			1	
11		1	1	1
10	1	1	1	

$$cout(a,b,cin,a\_ns) = \overline{a}\cdot b\cdot\overline{a\_ns} + \overline{a}\cdot cin\cdot\overline{a\_ns} + a\cdot b\cdot a\_ns + a\cdot cin\cdot a\_ns + cin\cdot b =$$
$$= \overline{a}\cdot\overline{a\_ns}(b+cin) + a\cdot a\_ns(b+cin) + cin\cdot b$$
$$= (b+cin)\cdot\left(\overline{a}\cdot\overline{a\_ns} + a\cdot a\_ns\right) + cin\cdot b$$
$$= (b+cin)\cdot XNOR\left(a,a\_ns\right) + cin\cdot b$$

נבנה מפת קרנו עבור s:

a b / cin a_ns	00	01	11	10
00		1		1
01		1		1
11	1		1	
10	1		1	

$$s\left(a,b,cin,a\_ns\right) = \overline{a}\cdot\overline{b}\cdot cin + \overline{a}\cdot b\cdot\overline{cin} + a\cdot b\cdot cin + a\cdot\overline{b}\cdot\overline{cin}$$
$$= \overline{a}(\overline{b}\cdot cin + b\cdot\overline{cin}) + a(b\cdot cin + \overline{b}\cdot\overline{cin})$$
$$= \overline{a}\cdot XOR(b,cin) + a\cdot XNOR(b,cin)$$
$$= XNOR(a, XNOR(b,cin))$$



טבלת חישוב ההשהיות המקסימליות מכל כניסה לכל יציאה:

זמני ה-t_{pd} החדשים של המעגלים הם:

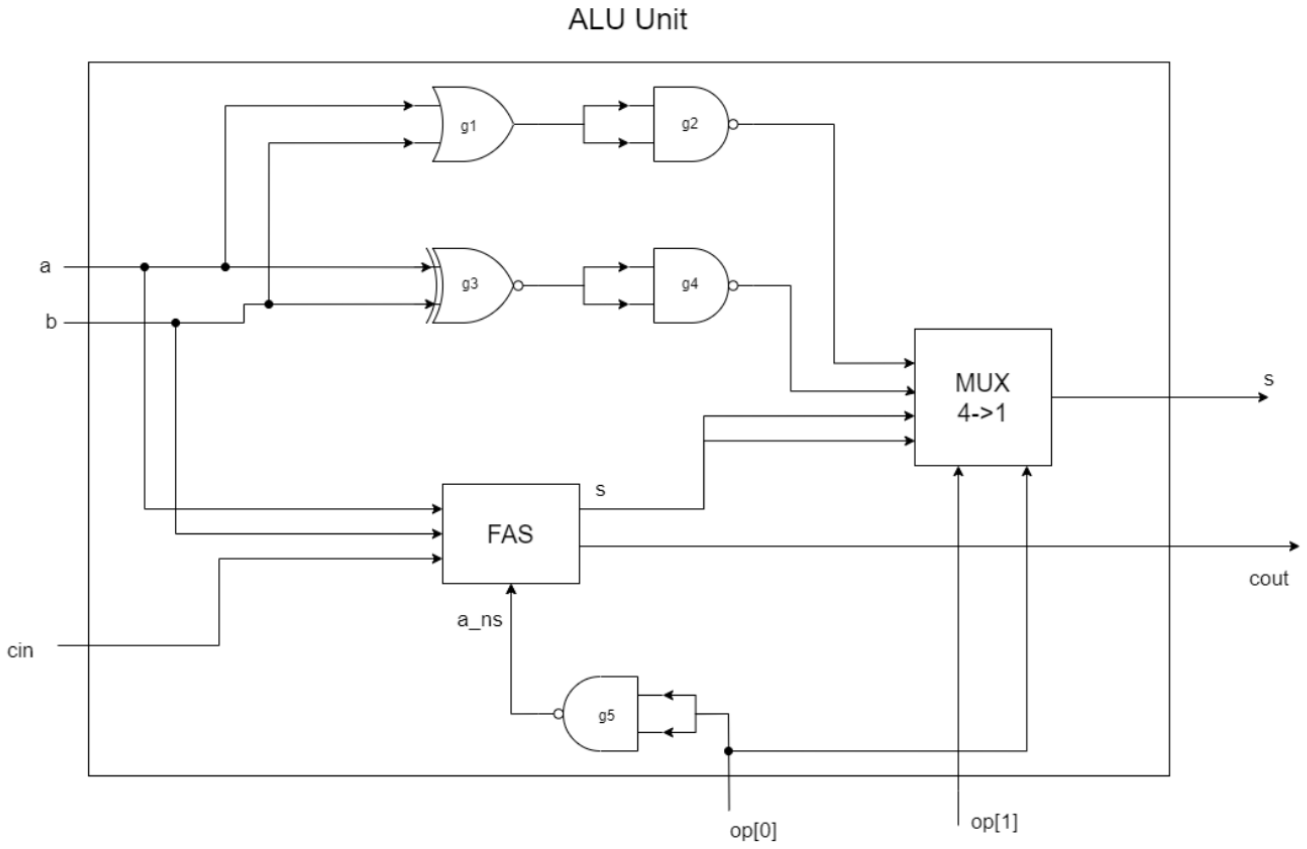
$$NAND2 = 10$$

$$OR2 = 8$$

$$XNOR2 = 10$$

כניסה	יציאה	Tpd
a	s	$XNOR2 = 10$
a	$cout$	$XNOR2 + 2 \cdot NAND2 + OR2 = 38$
b	s	$2 \cdot XNOR2 = 20$
b	$cout$	$2 \cdot NAND2 + 2 \cdot OR2 = 36$
cin	s	$2 \cdot XNOR2 = 20$
cin	$cout$	$2 \cdot NAND2 + 2 \cdot OR2 = 36$
a_ns	s	0 (אין מסלול מ a_ans ל- s)
a_ns	$cout$	$XNOR2 + 2 \cdot NAND2 + OR2 = 38$

סעיף 2.4



טבלת חישוב ההשהיות המקסימליות מכל כניסה לכל יציאה:

זמני ה-t_{pd} החדשים של המעגלים הם:

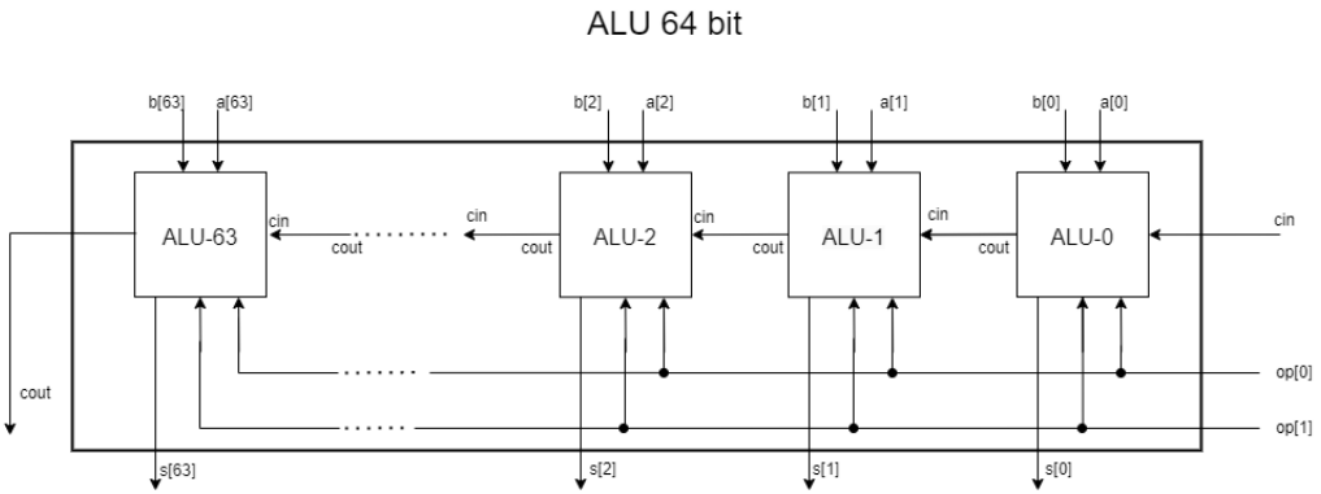
$$NAND2 = 10$$
$$OR2 = 8$$
$$XNOR2 = 10$$
$$MUX4 = 56$$
$$FAS = 38$$

כניסה	יציאה	Tpd
a	s	$FAS + MUX4 = 10 + 56 = 66$
a	$cout$	$FAS = 38$
b	s	$FAS + MUX4 = 20 + 56 = 76$

b	$cout$	$FAS = 36$
cin	s	$FAS + MUX4 = 20 + 56 = 76$
cin	$cout$	$FAS = 36$
$op[0]$	s	$NAND2 + FAS + MUX4 = 10 + 0 + 56 = 66$
$op[0]$	$cout$	$NAND2 + FAS = 10 + 38 = 48$
$op[1]$	s	$MUX4 = 38$
$op[1]$	$cout$	0 (אין לו השפעה על $cout$)

סעיף 2.5

שרטוט מימוש ה-ALU בעל כניסות $data$ ברוחב 64 ביט:



נבחר את המסלול שיגרום לזמן ההשהייה הארוך ביותר. נרצה ששינוי בכניסה אחת יגרור כמה שיותר שינויים אחריו ולכן נבחר לשנות את אחת הכניסות של הרכיב הראשון כך שנקבל זמן השהייה מקסימלי. הרכיב שיגרור השהייה מקסימלית ממנו אל היציאה $cout$ (כי רק ל-$cout$ יש השפעה על הרכיבים שלאחר מכן ולכן תגרור זמני השהייה נוספים עבור שאר יחידות הALU) הוא $op[0]$ ולכן נשנה אותו מ-0 ל-1, כאשר a הוא וקטור באורך 64 שכל איבריו 1, b הוא וקטור שכולו 0 באורך 64, cin הוא 1, $op[1]$ הוא 1. שינוי זה יגרור שינויים ב-$cout$ של כל אחד מהרכיבים.

לכן, כל אחד מהרכיבים $ALU_1 - ALU_{62}$ ייצטרך להמתין לשינוי ב-cin שמתקבל ביציאת ה$cout$ של הרכיב שלפניו, ולכן עבור כל אחד מ62 הרכיבים הנ"ל נחכה את השהייה שלהם: $cin \to cout$. אנחנו לא מתייחסים לזמן ההשהייה של $cin \to s$ עבור הרכיבים הללו, מפני שנחכה בכל מקרה ברכיב האחרון ALU_{63} ל-s שלו, שיסיים בהכרח אחרי שהם יחושבו כבר, לכן ניתן להתייחס רק אליו.

עבור הרכיב האחרון ALU_{63} נמצא את זמן ההשהייה המקסימלי מהכניסה cin שלו עד לאחת היציאות שלו- עד לשלב זה הוא כבר קיבל את כל האלמנטים והוא חיכה רק לcin, ולכן מקבלת הcin ההשהייה היא המקסימום בין הזמן של $cin \to cout$ לבין הזמן של $cin \to s$ כי בסוף זמן זה נקבל את $cout,\ s$ שלאחר סיום חישובים יהיה בידנו את כל התוצאות הרלוונטיות עבור חישוב הALU עם $data$ בגודל 64 ביטים.

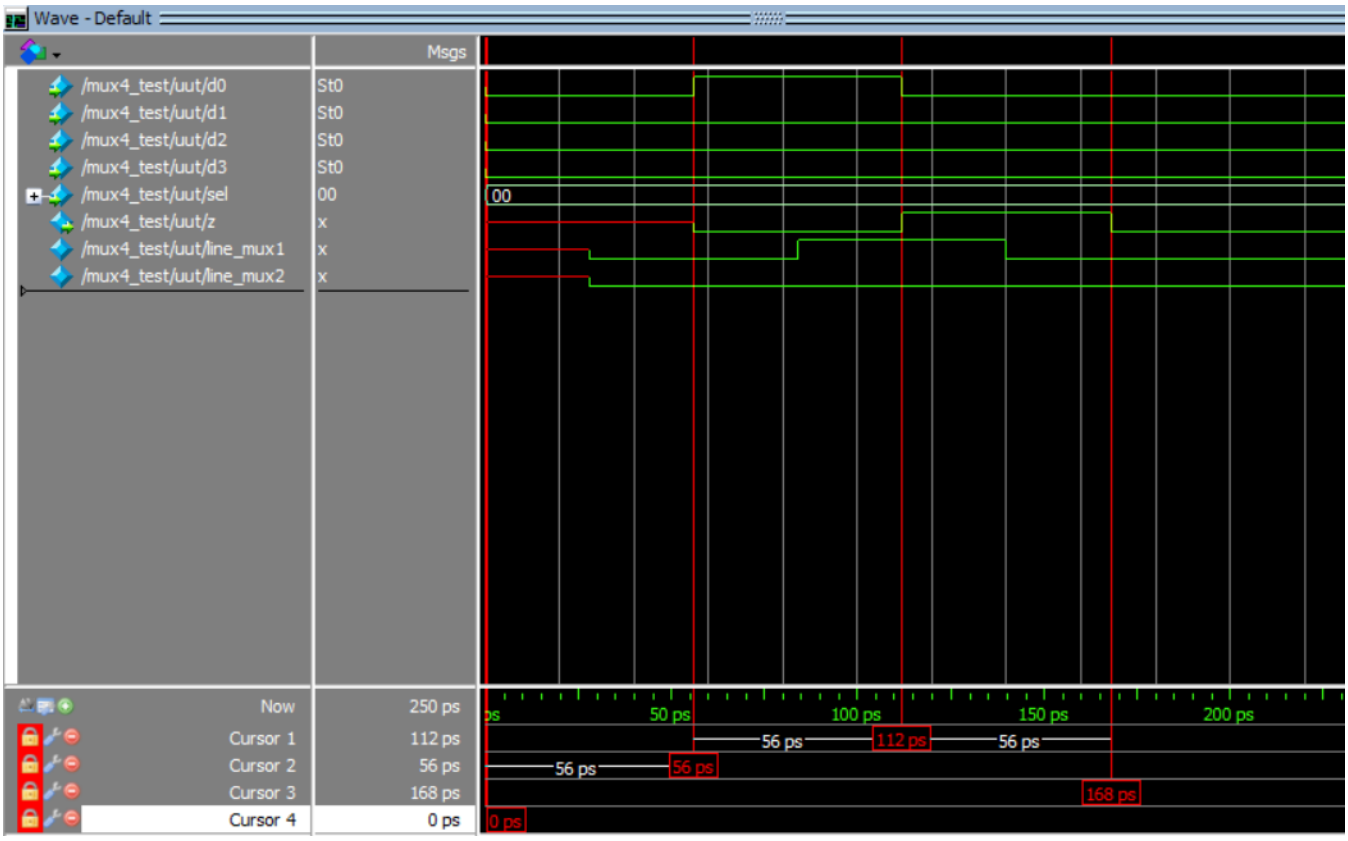
לכן חישוב המסלול שגורם להשהייה מקסימלית בין כניסה ליציאה יבוצע באופן הבא:

$$\underbrace{48}_{op[0] \to cout[0]} + 62 \cdot \underbrace{36}_{cin[i] \to cout[i]} + \underbrace{76}_{cin[63] \to s[63]} = 2356$$

חלק רטוב

סעיף 3.3:

תוצאות הסימולציה של $mux4\_test$:



<u>הסבר התוצאות:</u>

נסתכל על הכניסה d_0 ועל היציאה z.

בסימולציה הצבנו ערכי 0 בכל הכניסות ושינינו את הערך של d_0 מ-0 ל-1 ולאחר מכן בחזרה את הערך של d_0 מ-1 ל-0.

נשים לב כי כש-d_0 קיבל את הערך 0, z קיבל את הערך 0 לאחר 56 שניות.

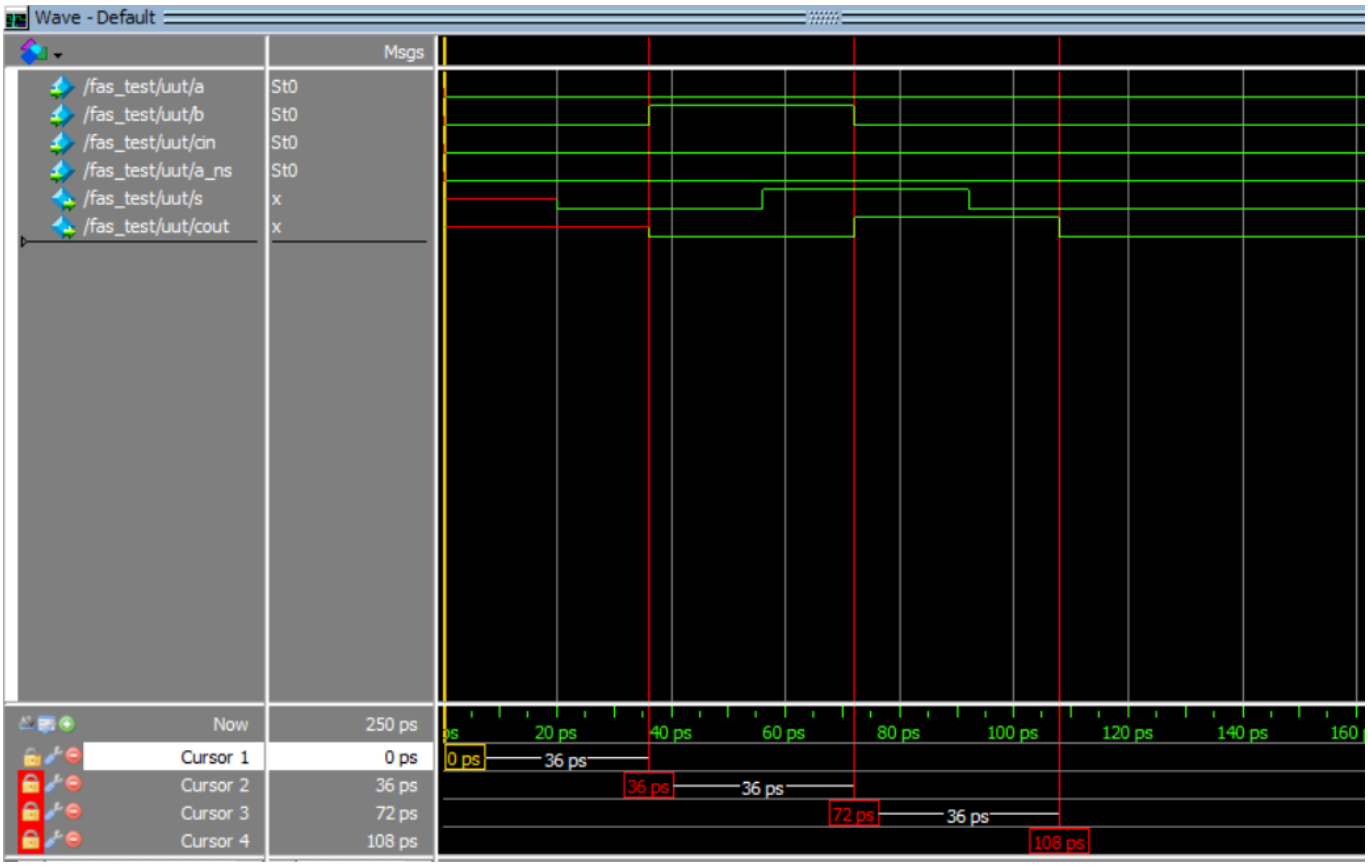
בנוסף, כש-d_0 השתנה מ-0 ל-1, השינוי ב-z מ-0 ל-1 קרה לאחר 56 שניות.

בנוסף, כש-d_0 השתנה מ-1 ל-0, השינוי ב-z מ-1 ל-0 קרה לאחר 56 שניות.

תוצאה זו היא התוצאה אשר קיבלנו גם בחלק היבש- כלומר תוצאות הסימולציה הן שזמן ההשהייה המקסימלי הוא 56 בהתאמה לתוצאות להן ציפינו בחלק היבש.

סעיף 3.5:

תוצאות הסימולציה של fas_test:



הסבר התוצאות:

נסתכל על הכניסה b ועל היציאה $cout$.

בסימולציה הצבנו ערכי 0 בכל הכניסות ושינינו את הערך של b מ-0 ל-1 ולאחר מכן בחזרה את הערך של b מ-1 ל-0.

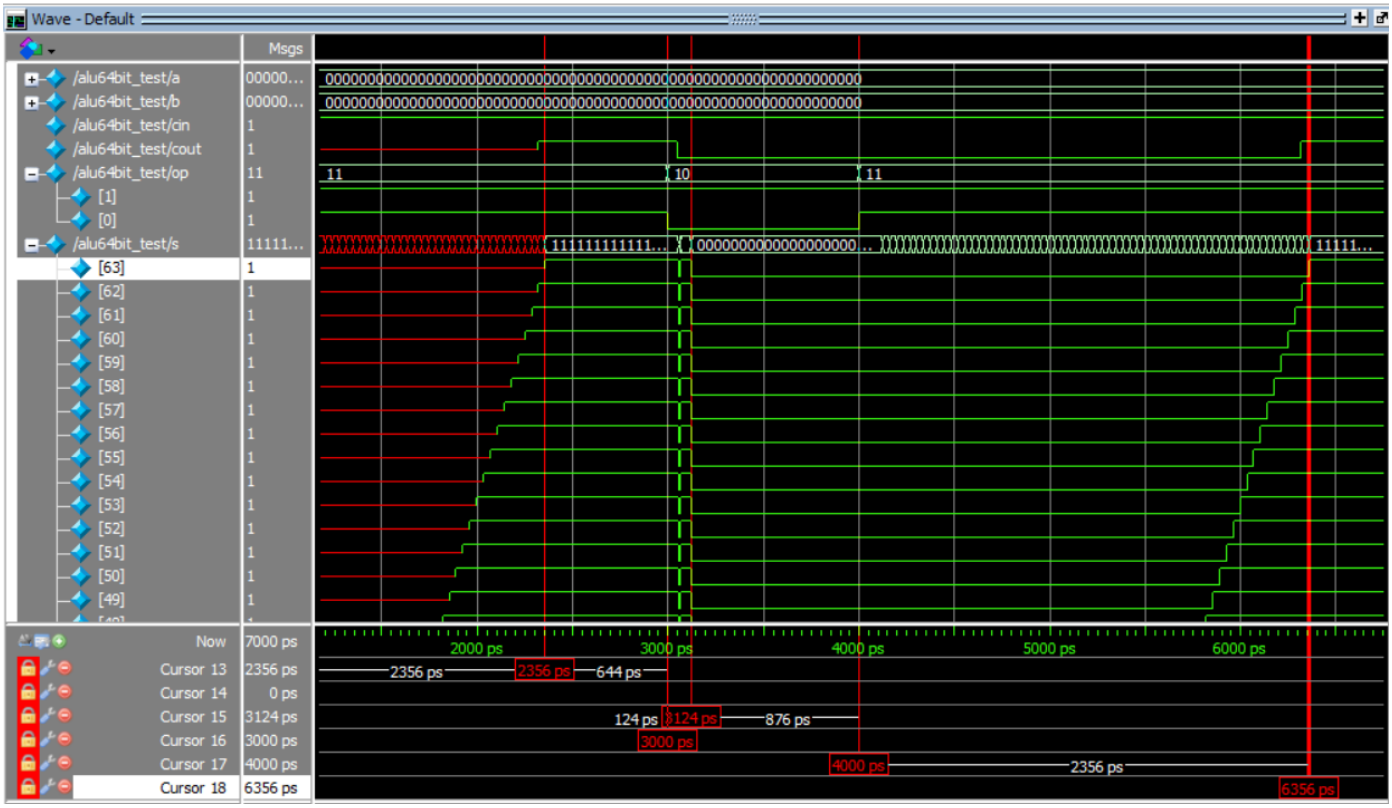
נשים לב כי כאשר b קיבל את הערך 0, $cout$ קיבל את הערך לאחר 36 שניות.

בנוסף, כש-b השתנה מ-0 ל-1, השינוי ב-$cout$ מ-0 ל-1 קרה לאחר 36 שניות.

בנוסף, כש-b השתנה מ-1 ל-0, השינוי ב-$cout$ מ-1 ל-0 קרה לאחר 36 שניות.

תוצאה זו היא התוצאה אשר קיבלנו גם בחלק היבש- כלומר תוצאות הסימולציה הן שזמן ההשהייה המקסימלי הוא 36 בהתאמה לתוצאות להן ציפינו בחלק היבש.

סעיף 3.8:

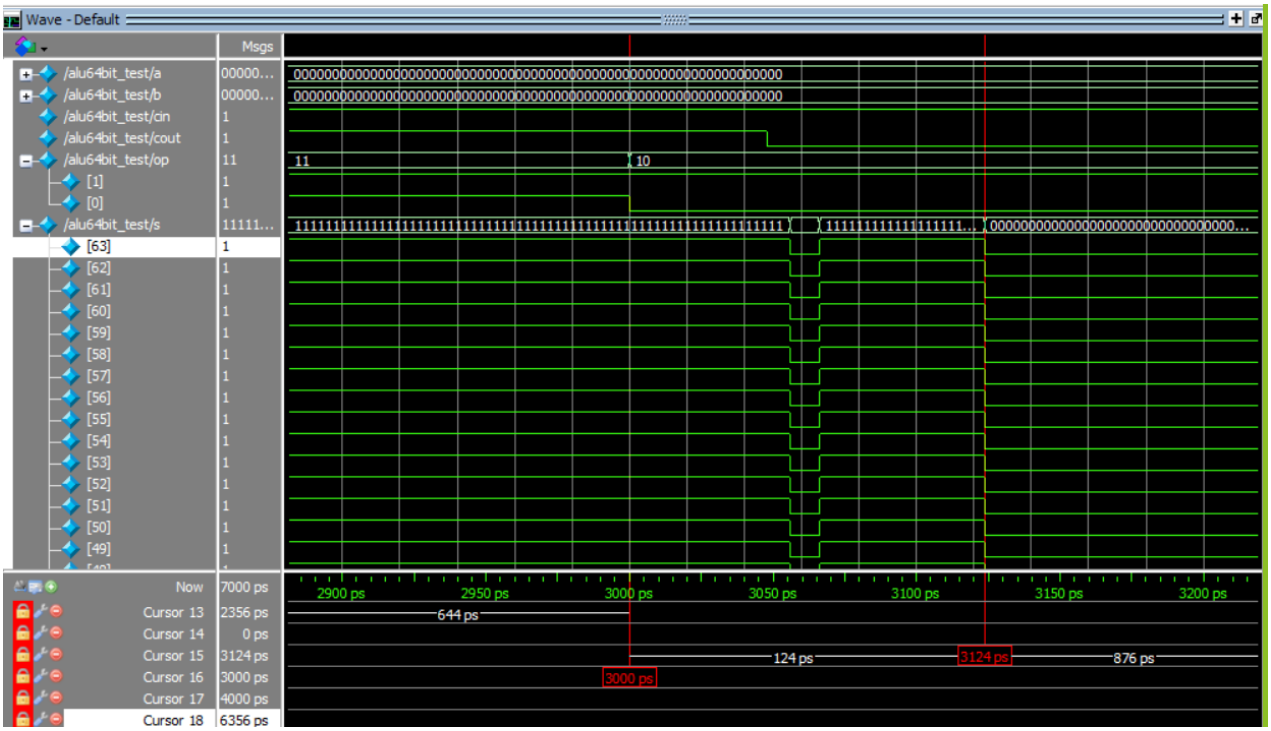


הסבר התוצאות:

בסימולציה הגדרנו את a כוקטור של 1 באורך 64, את b כוקטור שכולו 0 באורך 64, את cin בתור 1, את $op[1]$ בתור 1.

ראשית, נראה שלאחר $2356ps$ קיבלנו שהתוצאה התייצבה כפי שחישבנו בסעיף 2.5.

לאחר מכן, ברגע שעברו $3000\ ps$ מתחילת הטסט, נשנה את $p[0]$ מ-1 ל-0, ונראה כי קיבלנו הבהוב (האזארד) בזמן ההשהייה של הרכיב:



כעת, ברגע שעברו $4000\ ps$ מרגע תחילת הטסט, נשנה את $op[0]$ מ-0 ל-1, ונראה כי זמן ההשהייה של הרכיב הוא אכן $2356\ ps$ כפי שציפינו ועל פי החישוב שלנו בסעיף 2.5: (בהמשך)