

# תכנון מתקדם של מעגלי VLSI

פרוייקט חלק א'

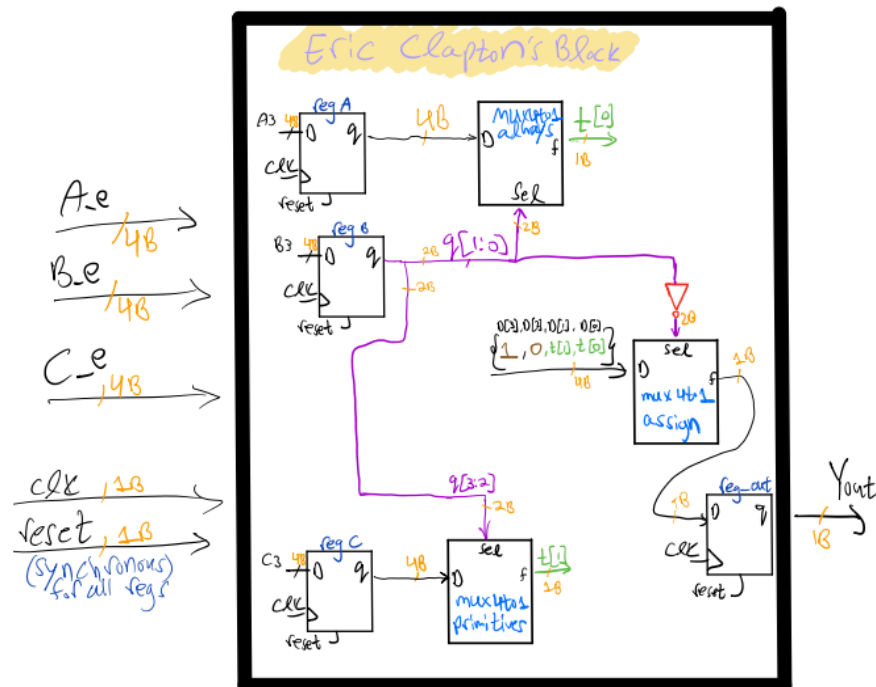
מגישים: team04c

205664014	idodatner	עידו דטנר
204765614	galgur1	גל גור
204620702	omri	עמרי אלעד
208410969	iristaubkin	איריס טאובקין

## תיאור כללי

ראש הצוות הוא עמרי אלעד ואת כלל הסביבה שבה פיתחנו את המודולים השונים ניתן למצוא תחת הכתובת:  
[/project/advvlsi/users/omri/ws/pro\\_1/](https://github.com/advvlsi/users/omri/ws/pro_1/) או ב [GitHub](#)  
הפרויקט מורכב מ3 תתי בלוקים שנקראים : dire\_straits, sultans\_of\_swing, ו-eric\_clapton.  
הבלוק העליון שכולל את שלושת תתי הבלוקים הנ"ל הינו - wembley\_88.

על כל בלוק נפרט באופן נפרד ממה הוא מורכב, ונצרף דיאגרמות גלים רלוונטיות אשר מאשרות את תקינותו אשר מתחתן יש הסבר שמוכיח את תקינותו \ תקינות אופן פעולת הבלוק.

הבלוק כפי שמתואר במטלה

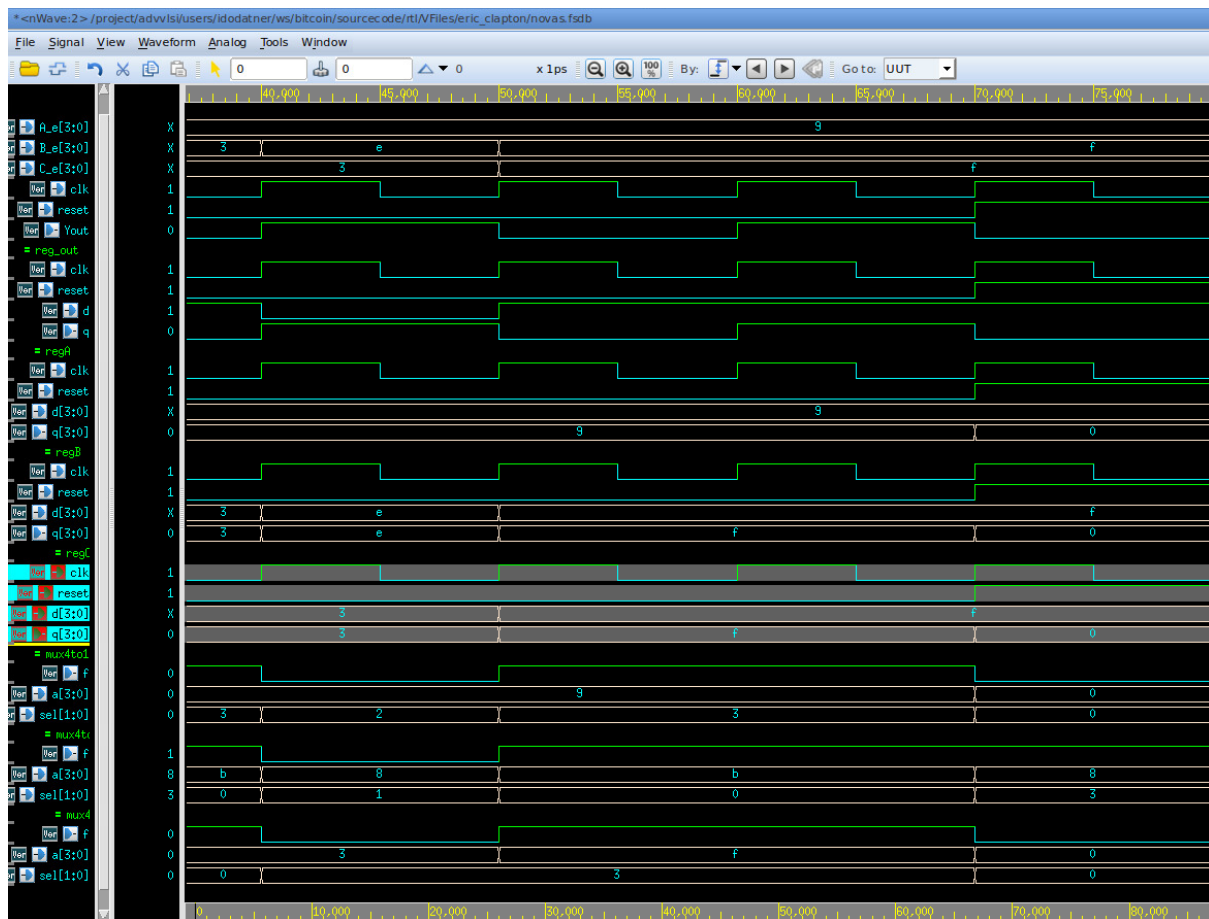
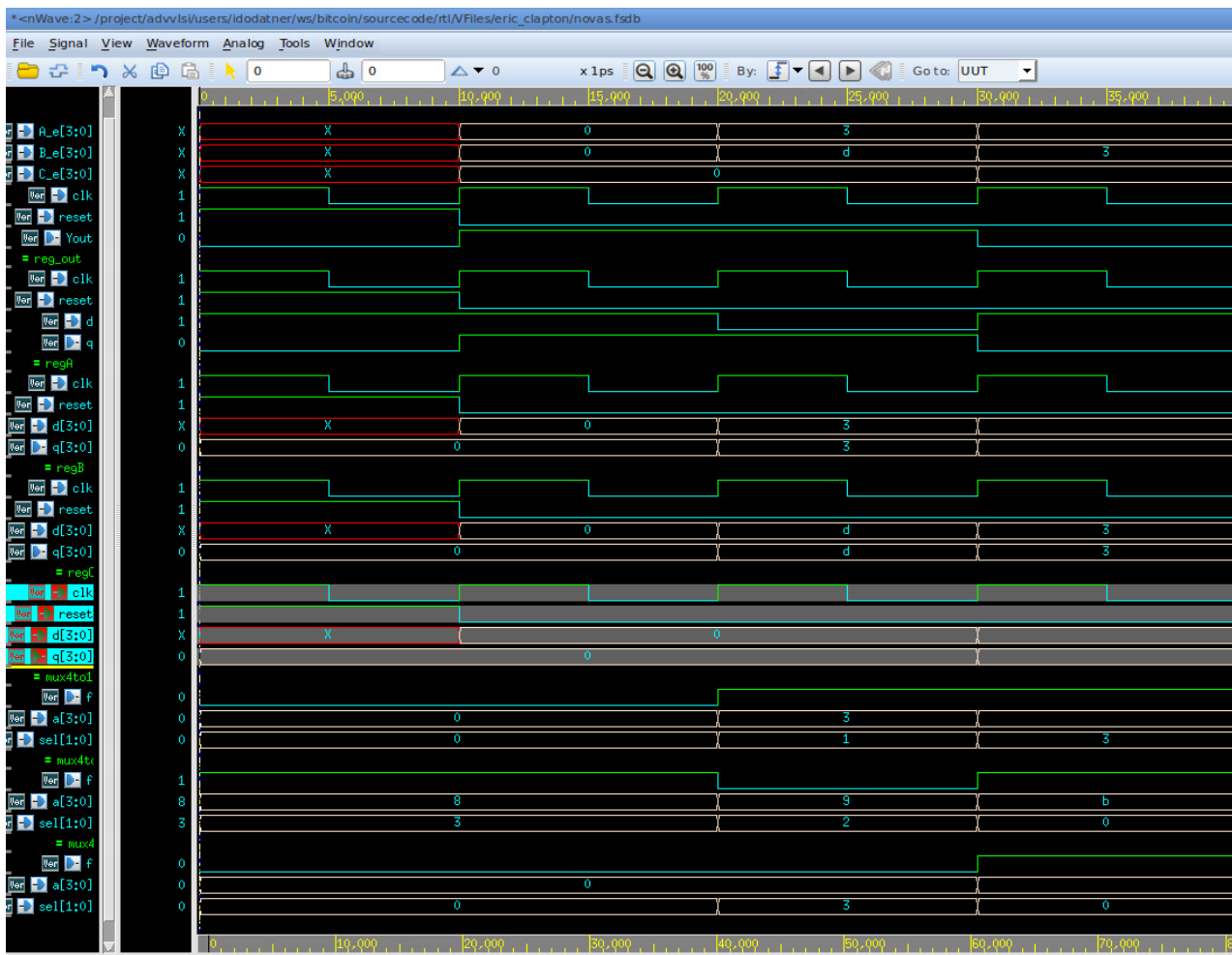
לפני שהתחלנו לייצר את הבלוק, היה עלינו לייצר 8 רכיבים שונים שמהם מורכב הבלוק הנ"ל וגם הבלוקים הבאים. הרכיבים האלו הינם:

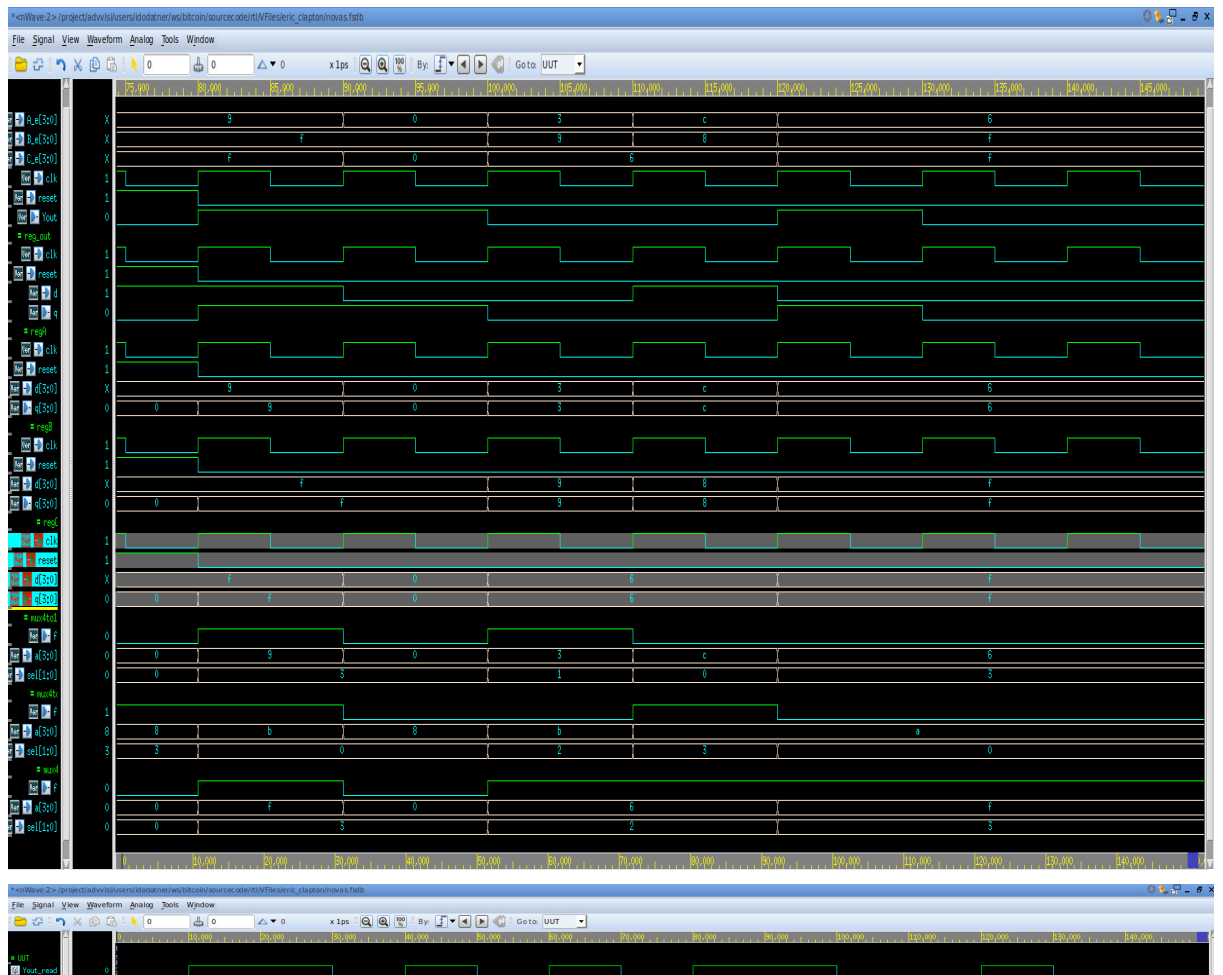
1. mux4to1\_assign
2. mux2to1\_assign
3. mux4to1\_primitives
4. mux2to1\_primitives
5. mux4to1\_always
6. mux2to1\_always
7. reg4
8. reg1

הסבר כללי של הבלוק

- רכיבי reg4 הינם הרכיבים שמתחברים לסיגנלים של כניסת הבלוק eric\_clapton, ומשמשים כרגיסטרים של סיגנלים אלו.
- ה mux4to1\_always מחובר לארבעת הביטים של היציאה של רגיסטר A ונשלט ע"י שני ביטי LSB של רגיסטר B.
- ה mux4to1\_primitives מחובר לרגיסטר C ונשלט ע"י 2 ביטי MSB של מוצא רגיסטר B.
- ה mux4to1\_assign מחובר בשני ביטי MSB שלו ל 1 ו-0 באופן קבוע, ולסיגנלים t0 ו t1 שיהם היציאות של ה mux4to1\_always ו mux4to1\_primitives בהתאמה, ובחרים את אחד מארבעת הסיגנלים האלו ע"י חיבור של 2 ביטי LSB של רגיסטר B שמעבירים אותם בשער NOT לתוך select של ה mux.
- reg1 מחובר ליציאה של ה mux4to1\_assign ומשמש כרגיסטר של יציאת המעגל כאשר מצידו השני הוא מחובר ל Yout.

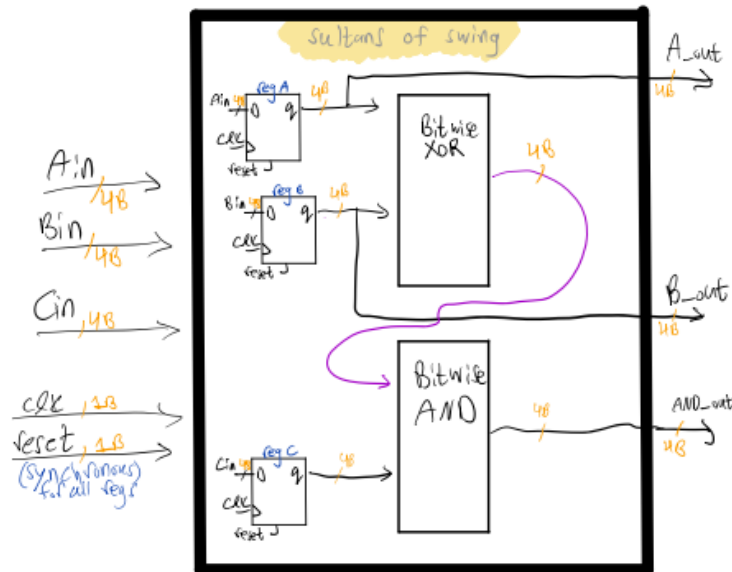
צילום סיגנלים להוכחת תקינות פעולת הבלוק (10 ווקטורים שונים שהוכנסו לבלוק לשם כך):





- ניתן לשים לב שהתחלנו "למדוד" את הסיגנלים רק לאחר שהייה קטנה של 10ns בכדי שהמערכת כולה תוכל להתאפס מבחינת reset, קלטים, פלטים וכו'.
- ניתן לראות שהסיגנלים שנכנסים לבלוק נדגמיים על ידי הרגיסטרים (A, B, ו-C) בעליית השעון הראשונה.
- נשים לב שאם נעקוב אחרי כניסת הselect של כל אחד מהמוקסים הרלוונטיים, המוצא של כל אחד מהם הוא בדיוק ע"פ כניסות הselect שלהם.
- ניתן לראות שה mux4to1\_assign במוצא שלו הינו סיגנל המוצא לאחר שעובר מחזור שעון, מכיוון שהוא מחובר לרגיסטר המוצא ולכן סיגנל המוצא מתעדכן ב'איחור' של מחזור אחד לעומת המוצא של mux4to1\_assign.
- נשים לב כי סיגנל ה-Yout\_read הוא בעצם החישוב שחישבנו של הסיגנל שאמור להתקבל לאחר המעבר בבלוק בהתאם לכניסות הלוגיקיות בכל רגע נתון - והוא תואם באופן מוחלט לסיגנל Youtn, שזהו הסיגנל שמייצג את מוצא הבלוק האמיתי באותו הרגע - ולכן ניתן לסכם כי הבלוק אכן מבצע את פעולתו כנדרש.

הבלוק כפי שמתואר במטלה

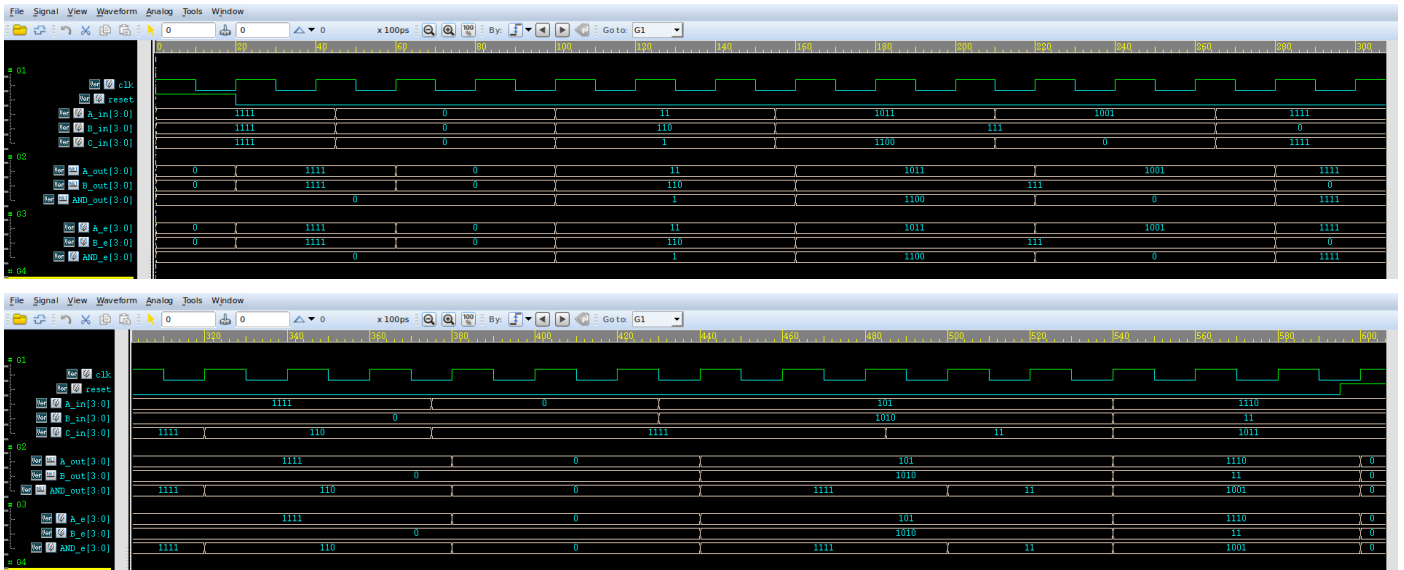


הסבר כללי של הבלוק

הבלוק Sultans of Swing הינו בלוק סינכרוני בעל דגימה עבור הכניסות (עושה שימוש בבלוק Reg4 שתואר בEric Clapton) ובפעולות Bitwise (פעולות לוגיות על ביטים).  
 לבלוק 3 כניסות דאטא: Ain, Bin, Cin ברוחב 4 ביטים כל אחת, ושתי כניסות נוספות: clk, reset ברוחב ביט יחיד כל אחת.  
 לבלוק 3 מוצאים: A\_out, B\_out, AND\_out ברוחב 4 ביטים על אחת.  
 הוצאים A\_out, B\_out מעבירים לבלוק הבא את הדגימה של הכניסות Ain, Bin בעוד שהמוצא AND\_out הוא מימוש של פונקציה לוגית על הדגימה של שלושת הכניסות: לכל ביט מתקיים:

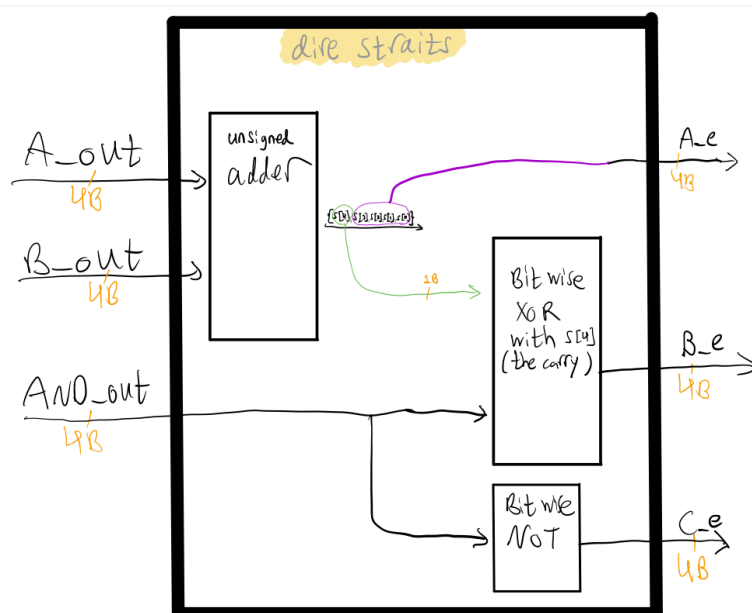
Ain [i] [sampled]	Bin [i] [sampled]	Cin [i] [sampled]	AND_out [i]
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

## צילום סיגנלים להוכחת תקינות פעולת הבלוק (10 ווקטורים שונים שהוכנסו לבלוק לשם כך):



בנוסף, ה Testbench מדפיס את הערכים שציפה לקבל (ואת אלה שקיבל בפועל) ומשווה ביניהם. במידה והתקבלה סתירה הוא עוצר ומודיע על כך, אחרת הוא מודיע שסיים בהצלחה.

TestBench for Sultan Of Swing:														
Cycle	Signal	Expected	Computed	V										
1	A out	0000	0000	1	11	A out	1011	1011	1	22	A out	0000	0000	1
1	B out	0000	0000	1	11	B out	0111	0111	1	22	B out	0000	0000	1
1	AND out	0000	0000	1	11	AND out	1100	1100	1	22	AND out	0000	0000	1
2	A out	1111	1111	1	12	A out	1001	1001	1	23	A out	0101	0101	1
2	B out	1111	1111	1	12	B out	0111	0111	1	23	B out	1010	1010	1
2	AND out	0000	0000	1	12	AND out	0000	0000	1	23	AND out	1111	1111	1
3	A out	1111	1111	1	13	A out	1001	1001	1	24	A out	0101	0101	1
3	B out	1111	1111	1	13	B out	0111	0111	1	24	B out	1010	1010	1
3	AND out	0000	0000	1	13	AND out	0000	0000	1	24	AND out	1111	1111	1
4	A out	0000	0000	1	14	A out	1001	1001	1	25	A out	0101	0101	1
4	B out	0000	0000	1	14	B out	0111	0111	1	25	B out	1010	1010	1
4	AND out	0000	0000	1	14	AND out	0000	0000	1	25	AND out	1111	1111	1
5	A out	0000	0000	1	15	A out	1111	1111	1	26	A out	0101	0101	1
5	B out	0000	0000	1	15	B out	0000	0000	1	26	B out	1010	1010	1
5	AND out	0000	0000	1	15	AND out	1111	1111	1	26	AND out	0011	0011	1
6	A out	0011	0011	1	16	A out	1111	1111	1	27	A out	0101	0101	1
6	B out	0110	0110	1	16	B out	0000	0000	1	27	B out	1010	1010	1
6	AND out	0001	0001	1	16	AND out	1111	1111	1	27	AND out	0011	0011	1
7	A out	0011	0011	1	17	A out	1111	1111	1	28	A out	1110	1110	1
7	B out	0110	0110	1	17	B out	0000	0000	1	28	B out	0011	0011	1
7	AND out	0001	0001	1	17	AND out	0110	0110	1	28	AND out	1001	1001	1
8	A out	0011	0011	1	18	A out	1111	1111	1	29	A out	1110	1110	1
8	B out	0110	0110	1	18	B out	0000	0000	1	29	B out	0011	0011	1
8	AND out	0001	0001	1	18	AND out	0110	0110	1	29	AND out	1001	1001	1
9	A out	1011	1011	1	19	A out	1111	1111	1	30	A out	1110	1110	1
9	B out	0111	0111	1	19	B out	0000	0000	1	30	B out	0011	0011	1
9	AND out	1100	1100	1	19	AND out	0110	0110	1	30	AND out	1001	1001	1
10	A out	1011	1011	1	20	A out	0000	0000	1	Success: Testbench Finished Successfully				
10	B out	0111	0111	1	20	B out	0000	0000	1					
10	AND out	1100	1100	1	20	AND out	0000	0000	1					
					21	A out	0000	0000	1					
					21	B out	0000	0000	1					
					21	AND out	0000	0000	1					

הבלוק כפי שמתואר במטלההסבר כללי של הבלוק

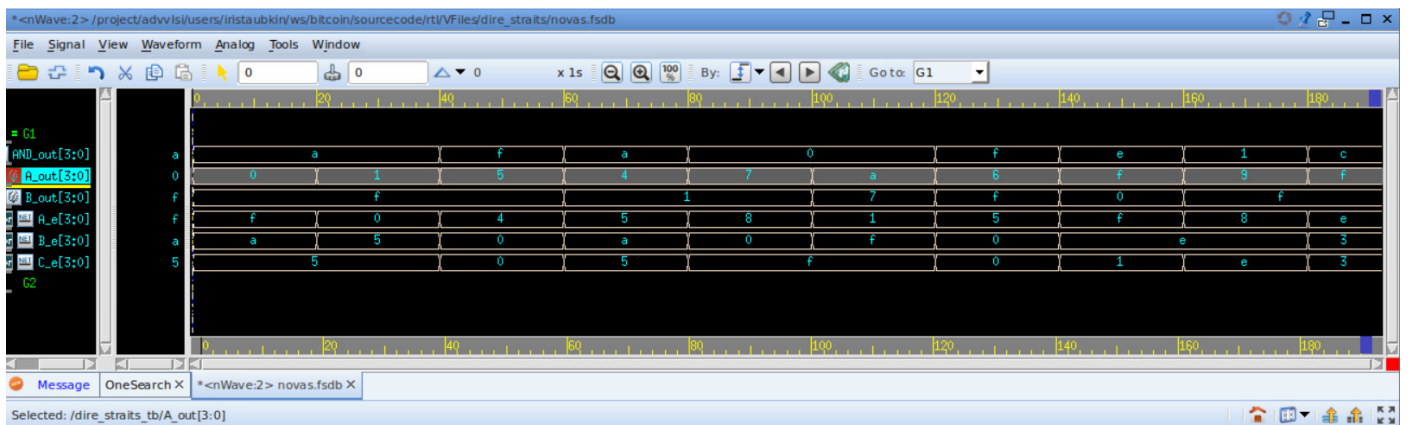
הבלוק הינו בלוק א-סינכרוני אשר כולל רק לוגיקה קומבינטורית בלבד. לבלוק 3 כניסות: A\_out, B\_out, AND\_out ו-3 מוצאים: A\_e, B\_e, C\_e. הכניסות A\_out ו-B\_out עוברות דרך unsigned adder. המוצא A\_e הוא למעשה ארבעת הביטים התחתונים של תוצאת החיבור. המוצא B\_e הוא תוצאת bitwise XOR בין ה-MSB של תוצאת החיבור לבין הכניסה AND\_out ואילו המוצא C\_e הוא Bitwise NOT של AND\_out.

## צילום סיגנלים להוכחת תקינות פעולת הבלוק (10 ווקטורים שונים שהוכנסו לבלוק לשם כך):

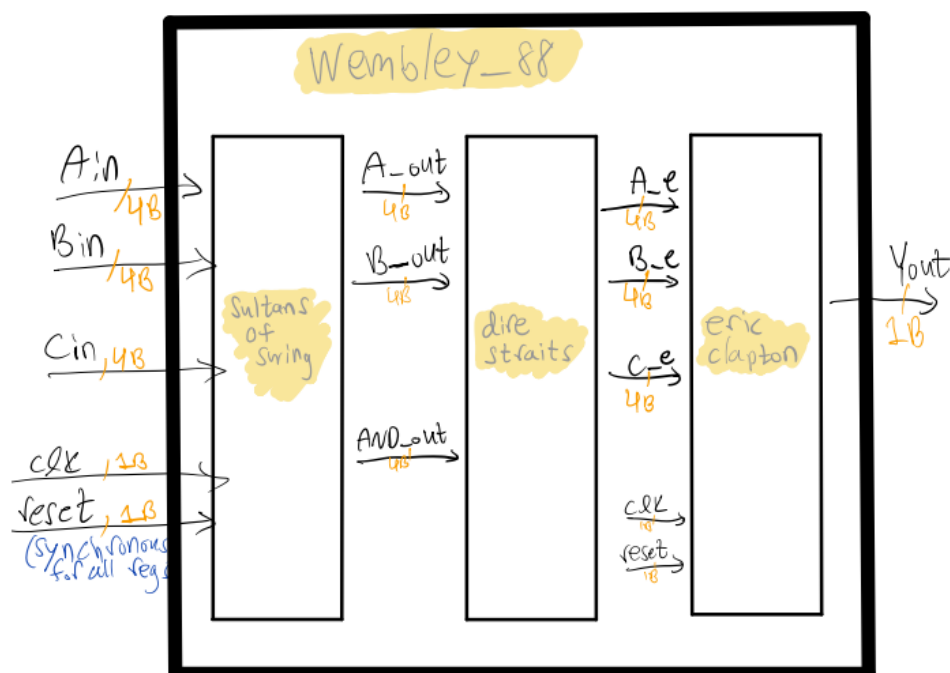
ה-testbench למעשה בודק את 10 הקומבינציות הבאות ועבור כל קומבינציה בודק את שלושת המוצאים ומדפיס אם אחד מהם אינו נכון.  
הקומבינציות:

A_out	B_out	AND_out	A_e	B_e	C_e
0000	1111	1010	1111	1010	0101
0001	1111	1010	0000	0101	0101
0101	1111	1111	0100	0000	0000
0100	0001	1010	0101	1010	0101
0111	0001	0000	1000	0000	1111
1010	0111	0000	0001	1111	1111
0110	1111	1111	0101	0000	0000
1111	0000	1110	1111	1110	0001
1001	1111	0001	1000	1110	1110
1111	1111	1100	1110	0011	0011

הגלים:





הבלוק כפי שמתואר במטלההסבר כללי של הבלוק

הבלוק הנ"ל מכיל את שלושת הבלוקים הנ"ל, ואת הוקטורים הפעם וידאנו בעזרת בניית כלל המעגלים בlogisim ווידוא המוצא של כל המעגלים כאשר הם מחוברים אחד לשני ויוצרים את הבלוק הנ"ל.

- נשים לב בגלים כי בבלוק הראשון - sultans of swing - יש רגיסטרים בכניסה ולכן עדכון כלל הערכים במוצא של הבלוק הראשון יקרו בעליית השעון הראשונה לאחר היציאה מreset.
- בבלוק השני - dire straits - לא קיימים רגיסטרים ולכן ערכי המוצא מתעדכנים ישר באופן מיידי באותו המחזור שערכי הכניסה נכנסים אליו.
- בבלוק השלישי - eric clapton - הסיגנל "מתעכב" בגלל שני רגיסטרים שהוא עובר דרכם, ולכן ייקח לסיגנל שני מחזורים מהרגע שהגיע לכניסה ועד שיצא סיגנל מהמוצא של הבלוק הזה, ז"א נצפה כי המוצא יתעדכן לאחר 2 עליות שעון מרגע שייכנסו ערכים בכניסה לבלוק.
- לסיכום, ניתן לראות כי מהרגע שייכנס סיגנל לבלוק ועד הרגע שייצא, עוברים כשלושה מחזורים (3 עליות שעון).
- הערה: לאחר שקיבלנו ערכי מוצא רלוונטיים מהlogisim, הכנסו אותם לתוך הtbl והוספנו הדפסות שיקרו במידה והערך שיוצא מהבלוק לא תואם לערך שאמור להתקבל במוצא של הבלוק כולו - ניתן לראות כי אף הדפסה כזו לא קרתה וכל ערכי המוצא מהבלוק תאמו לערכים להם ציפינו וחישבנו מראש :

```

FIRST CHECK
A_in is 0
B_in is 15
C_in is 10
SECOND CHECK
A_in is 8
B_in is 15
C_in is 10
THIRD CHECK
A_in is 9
B_in is 8
C_in is 10
FOURTH CHECK
A_in is 2
B_in is 11
C_in is 11
FIFTH CHECK
A_in is 10
B_in is 15
C_in is 15
SIXTH CHECK
A_in is 1
B_in is 3
C_in is 15
SEVENTH CHECK
A_in is 9
B_in is 11
C_in is 15
EIGHTH CHECK
A_in is 13
B_in is 2
C_in is 10
NINTH CHECK
A_in is 6
B_in is 11
C_in is 1
TENTH CHECK
A_in is 6
B_in is 12
C_in is 9
$finish called from file ".././../tb/VFiles/wembley_88/wembley_88_tb.v", line 194.
$finish at simulation time          3300
      V C S   S i m u l a t i o n   R e p o r t
Time: 330000 ps
CPU Time:      0.350 seconds;      Data structure size:   0.0Mb
Fri May 13 15:46:55 2022

```

צילום סיגנלים להוכחת תקינות פעולת הבלוק (10 ווקטורים שונים שהוכנסו לבלוק לשם כך):

