CPU Architecture

Task 1 report

ALU Design with VHDL

עומר לוקסמבורג 205500390

עילי נוריאל 312538580

**תוכן עניינים**

[**הגדרת תכנון המערכת** 3](#_Toc40713112)

[**תיאור הרכיבים** 5](#_Toc40713113)

[**תיאור הבדיקות** 6](#_Toc40713114)

[**File based simulation of DUT** 11](#_Toc40713115)

# **הגדרת תכנון המערכת**

התבקשנו לכתוב תיאור חומרה למערכת הבאה:

תכנון ALU בסיסי, עם רגיסטרים בכניסה וביציאה שלו. ה-ALU מורכב מ:

1. יחידה אריתמטית המבצעת פעולות מתמטיות.

2. יחידת הזזה – shift המבצעת פעולות הזזה שונות.

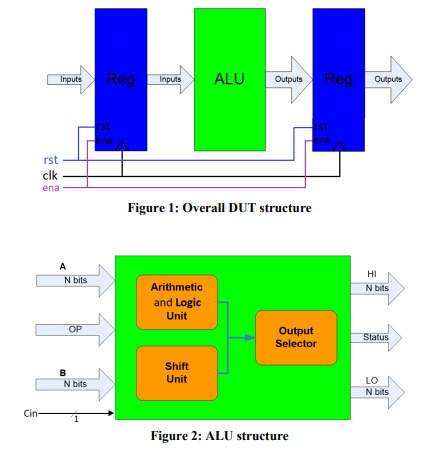
3. בורר יציאה – Selector – בין היחידה האריתמטית ל-shift unit.

המערכת מקבלת שני Inputים A,B באורך ברירת מחדל של 8 ביט, OPCODE באורך 5 ביט ו-Cin ומוציאה את תוצאות הפעולה התואמת ל-OPCODE לרגיסטר RES המורכב משני רגיסטרים באורך 8 ביט (HIGH ו-LOW).

**דרישות נוספות:**

* השכבה העליונה של ה-ALU צריכה להיות Structural.
* יש להשתמש בקבצים top ו-aux\_package שניתנו עם המשימה.
* יש להשתמש במודול אחד של כפל ומודול אחד של חיבור-חיסור בלבד.
* נדרש לכתוב Test Bench למערכת כולה.

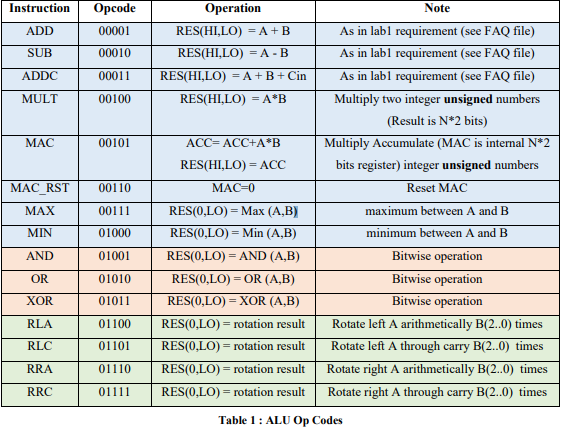
**תרשים המערכת:**



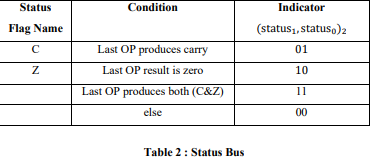
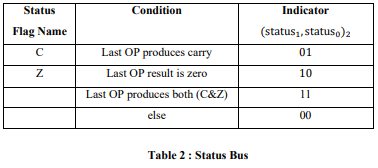
**חיבורים:**

A, B – ערוצי הכניסה בעלי n ביטים כל אחד.

Cin - כניסה של ביט נוסף (carry).

OPCODE - בחירת הפעולה שתתבצע על הכניסות הנקבעת באופן הבא:

Hi, Low – ערוצי היציאה באורך n ביטים כל אחד.

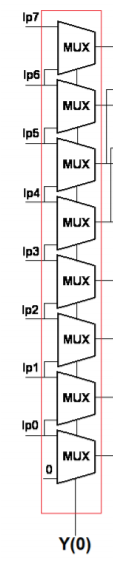
Status – יציאה באורך 2 ביטים המייצגת דגלי Zero ו-Carry של הפעולה האחרונה שהתבצעה:

# **תיאור הרכיבים**

לפי הגדרת המשימה הגדרנו את המודולים ותתי המודולים הבאים (מסודרים לפי היררכיית השכבות):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| top | | | | aux\_package | |
| ALU | | | |
| Arithmetic\_Logic\_Unit | Shift\_Unit | | Selector | |  |
| Adder\_Substructor | RLA | RRA |  | |
| FA | Yblock | Yblock |
| MUX2 | MUX2 |

**הערות – לגבי המודולים ממעבדה 1**:

* מודול ה-Yblock כתוב בצורה גנרית ומתאר שכבה אחת ברכיב הshifter- (כמתואר בדרישות המעבדה הראשונה) כלומר עמודה כזו:
* מודול ה-FA מתאר רכיב Full Adder לשני ביטים ומודול הMUX2 מתאר מרבב 1 🡪 2.

**תיאור הלוגיקה של המודולים בעמוד הבא**

# **תיאור הבדיקות**

## **File based simulation of DUT**

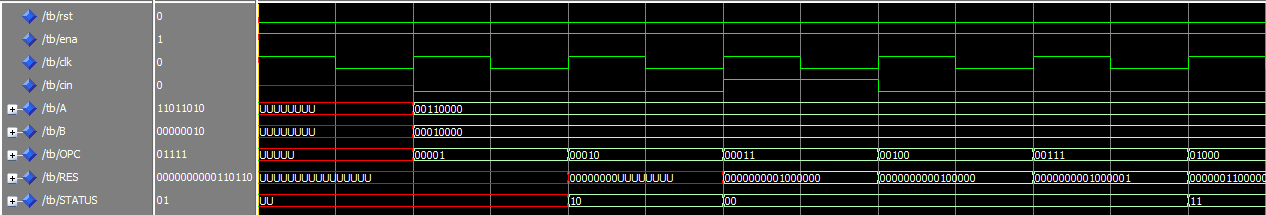
כתבנו מודול tb היוצר מקובץ הinput.txt- סימולציה למערכת.

בקובץ הinput- הכנסנו תרחישים לכל הפקודות השונות אותן המערכת צריכה לבצע (כל הopcodeים), וכן מקרי קצה.

כלומר הוא מכיל ערכים לכל הקלטים של המערכת. כמו כן המודול tb כותב את הפלטים לקובץ output.txt.

**ניתוח תוצאות הסימולציה בתצוגת wave form:**

בדיקת הפעולות ADD, SUB, ADDC, MULT, MAX, MIN, AND, OR, XOR:



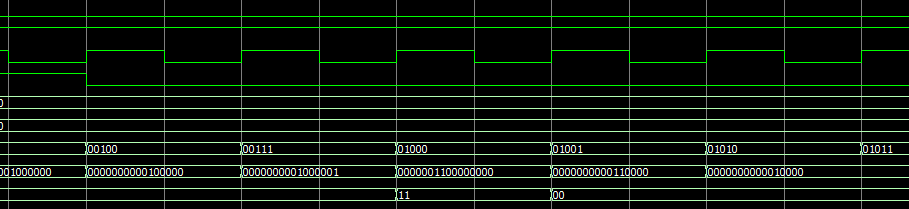
שני מחזורי שעון

הopcode משתנה ל1 כלומר פעולת ADD.

נשים לב כי הפלט התואם לפקודה הזו ייצא רק לאחר שני מחזורים שעון בעקבות ההשהיה של רגיסטרי הכניסה והיציאה.

לאחר שני מחזורי שעון ערך היציאה הוא אכן

המשך

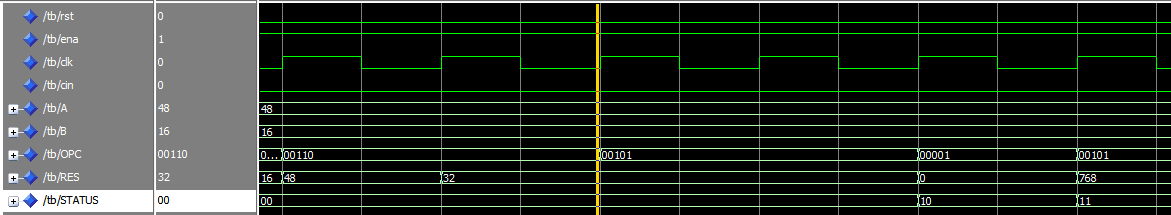


הopcode משתנה ל100 כלומר פעולת MULT.

נשים לב כי הפלט התואם לפקודה הזו ייצא רק לאחר שני מחזורים שעון בעקבות ההשהיה של רגיסטרי הכניסה והיציאה.

לאחר שני מחזורי שעון ערך היציאה הוא אכן

, כאשר A=32 B=48,

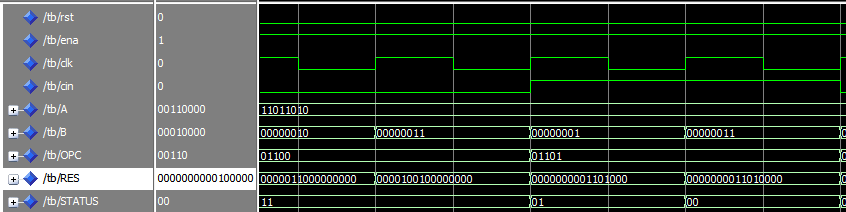
בדיקת הפעולות MAC, MAC\_RST:

2. הopcode משתנה ל101 כלומר פעולת MAC ונשאר במשך עוד שני מחזורים.

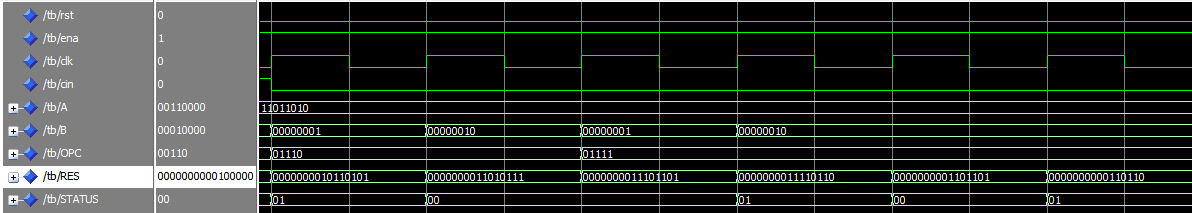
החץ האדום מציג את המצב בו ACC=0 – הערך 0 הוא המוצא עדיין.

החץ הירוק מציג את הערך שבו ACC כבר שונה מ-0.

1. הopcode משתנה ל110 כלומר פעולת MAC RST. ערך הRES לא משתנה

בדיקת הפעולות RLA, RRA, RRC, RLC:

הopcode משתנה ל1100 כלומר פעולת RLA. 2B= לכן יש הזזה של 2 עם הוספת ,"00" מימין.



שני מחזורי שעון

המשך

הopcode הוא 01111 כלומר פעולת RRC.

A=11011010 B=2 - כלומר יתבצעו 2 הזזות על A כאשר cin=**0**. נשים לב שזה אומר שביט הCarry צריך להיות 1 (הביט האחרון שיצא מ-RES(LO).

בנוסף התוצאה תיהיה RES(LO)=0**0**110110

כאשר ה-0 האדום הוא ביט הקארי ומשמאל אליו ביט ה-LSB.