

CPU Architecture

Task 2 report

FPGA based ALU Design

עומר לוקסמבורג 205500390

עילי נוריאל 312538580

הגדרת תכנון המערכת

מטלה זו היא מטלת המשך של מטלה 2 ונבצע בה סינתזה של רכיב ה-ALU שכתבנו במטלה הקודמת, על רכיב Cyclone II FPGA. התבקשנו לבצע שני מקרי בוחן לחומרה שתכננו:

מקרה בוחן ראשון - Performance Test Case

תרשים המערכת:

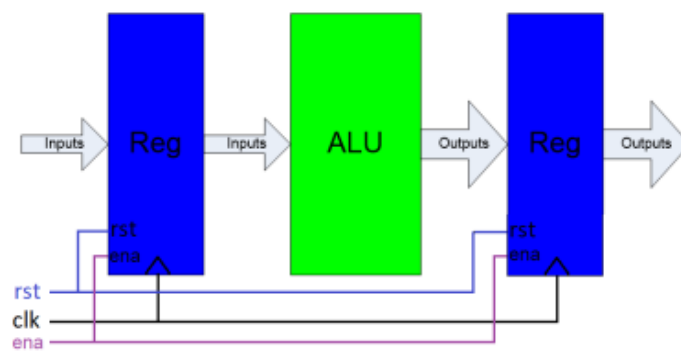


Figure 1: Overall DUT structure

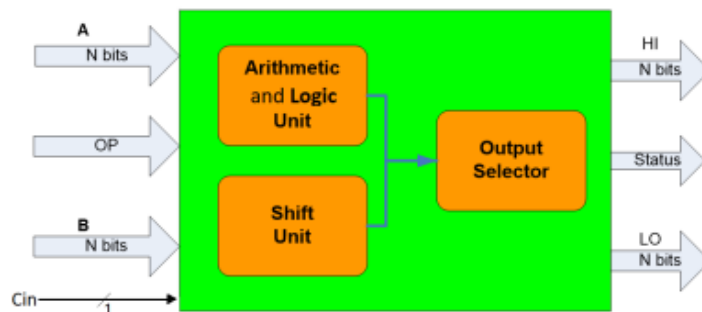
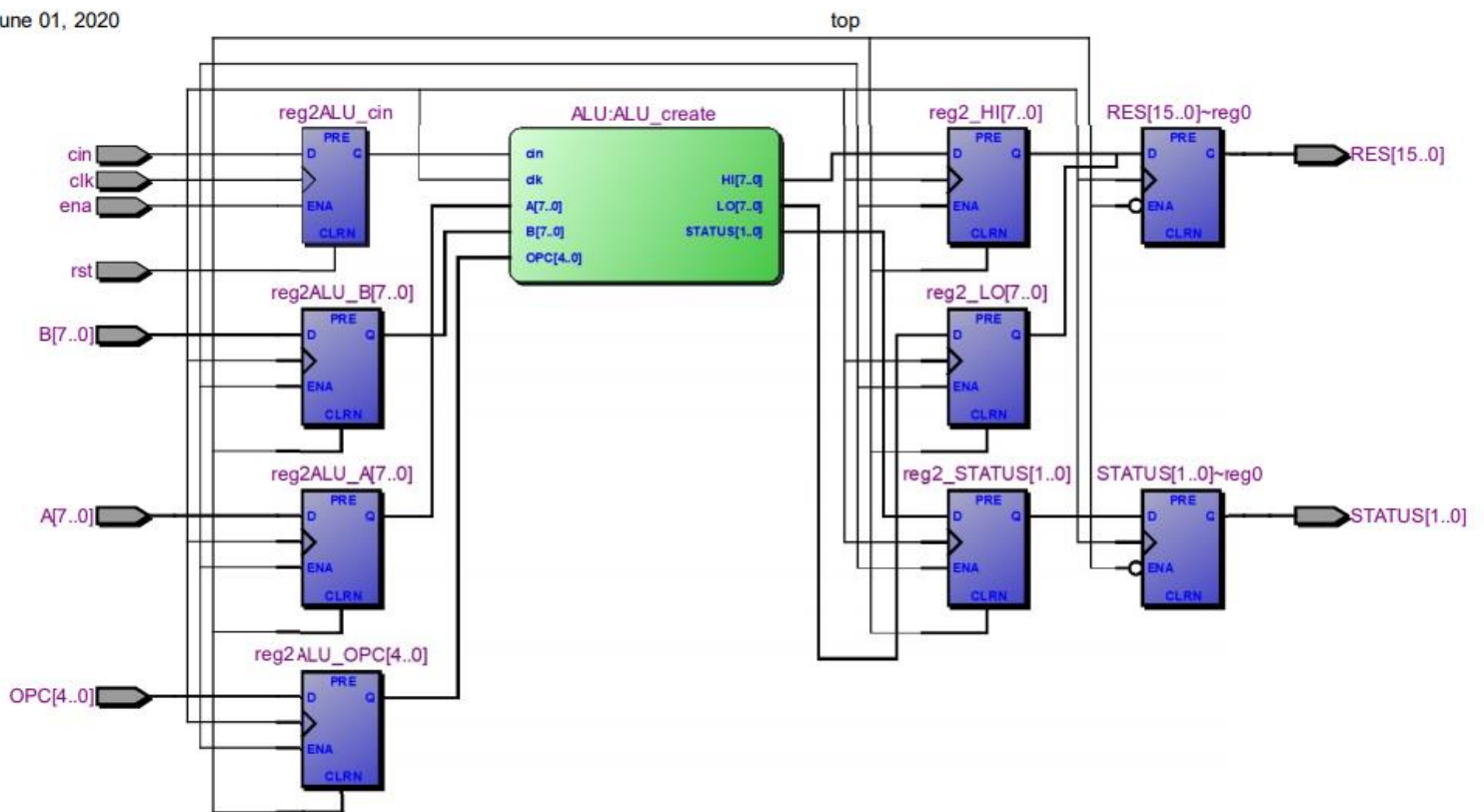


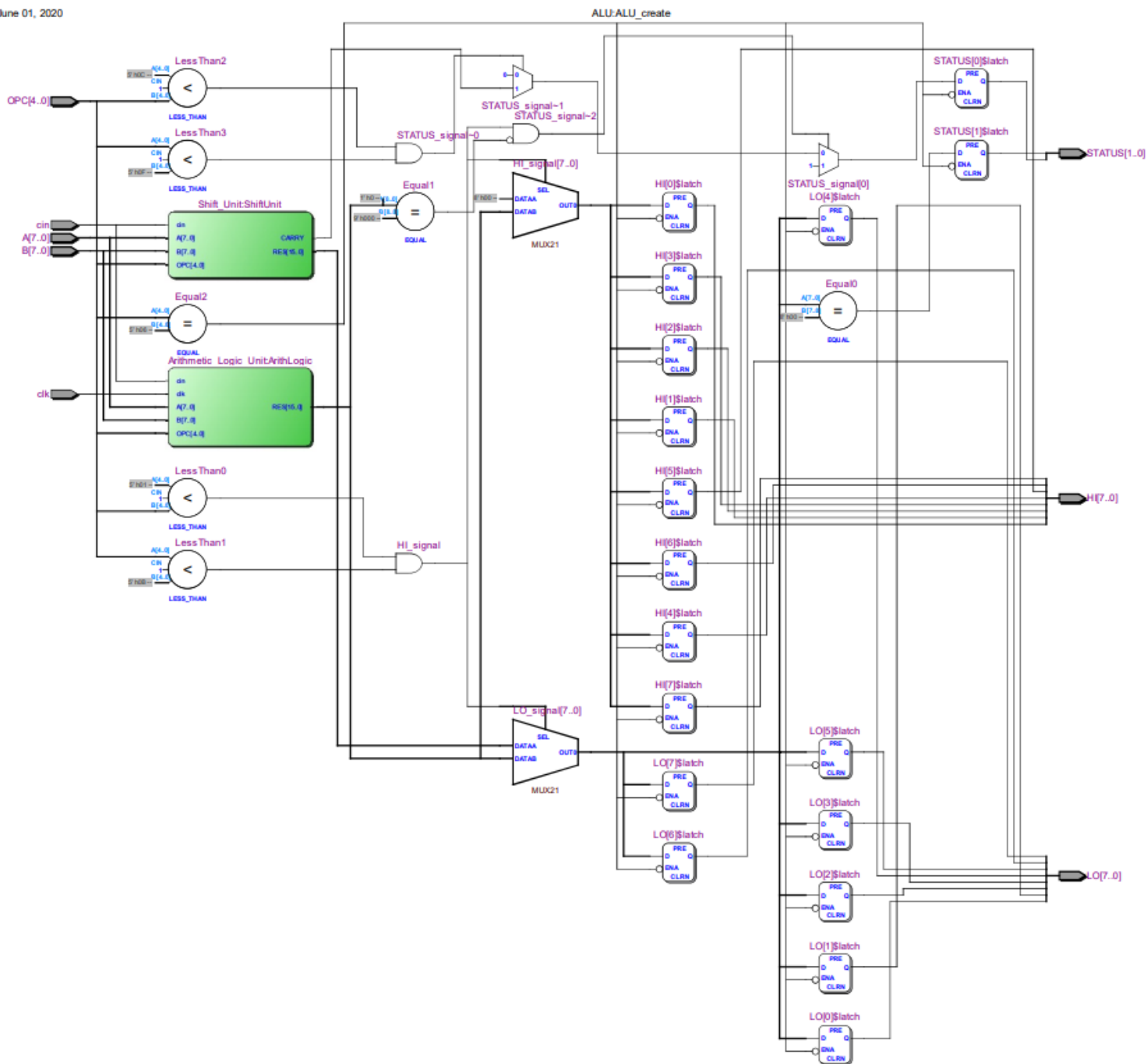
Figure 2: ALU structure

- סימולציית ה-ModelSim בוצעה במטלה 1 – ראה נספחים.
- אופטימיזציות שבוצעו על הקוד של מטלה 1 בכדי להתאימה למטלה 2 הן:
 - ביטול הרגיסטרים בכניסה העוטפים את ה-ALU והחלפתם ברגיסטרים ספציפיים לכל אחת מהכניסות הפועלים באמצעות קו enable. כמו כן DFF של פעולת ה-MAC נכנס קו enable נוסף והוא איננו מחובר לשעון.
 - ביטול הרגיסטרים במוצא – הפלט מגיע ישירות למוצא ללא סנכרון.
 - כתוצאה משינויים אלו המערכת אינה תלויה בשעון אלא רק בקווי ה-enable השונים. לכן יש חשיבות לסדר הפעולות המבצעים על רכיב ה-FPGA כאשר מפעילים אותו (טעינת הערכים לרגיסטרים A,B בחירת OPCODE וטעינתו, ובמקרה של MAC גם לחיצה על לחצן ה-clk).

ביצענו קומפילציה של התכנון שלנו ב-Quartus . ה-RTLים של הרכיבים:

Date: June 01, 2020





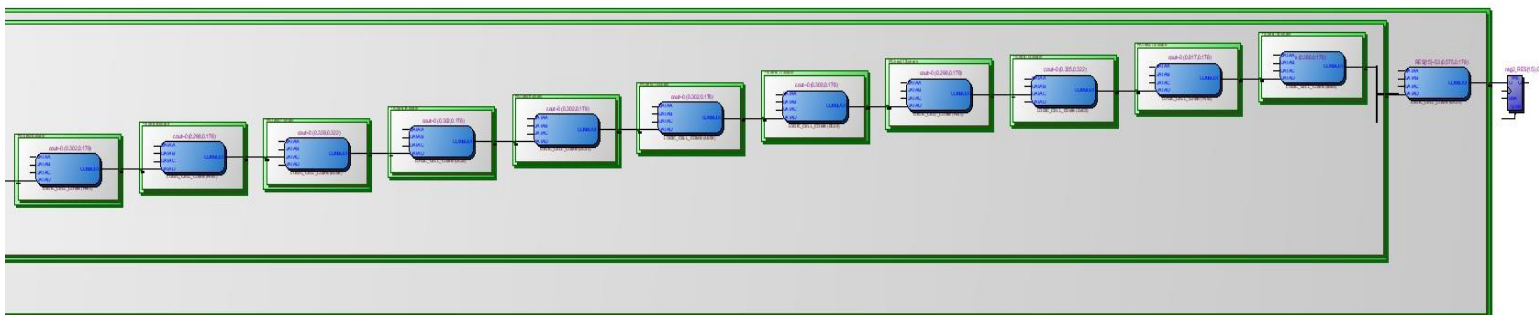
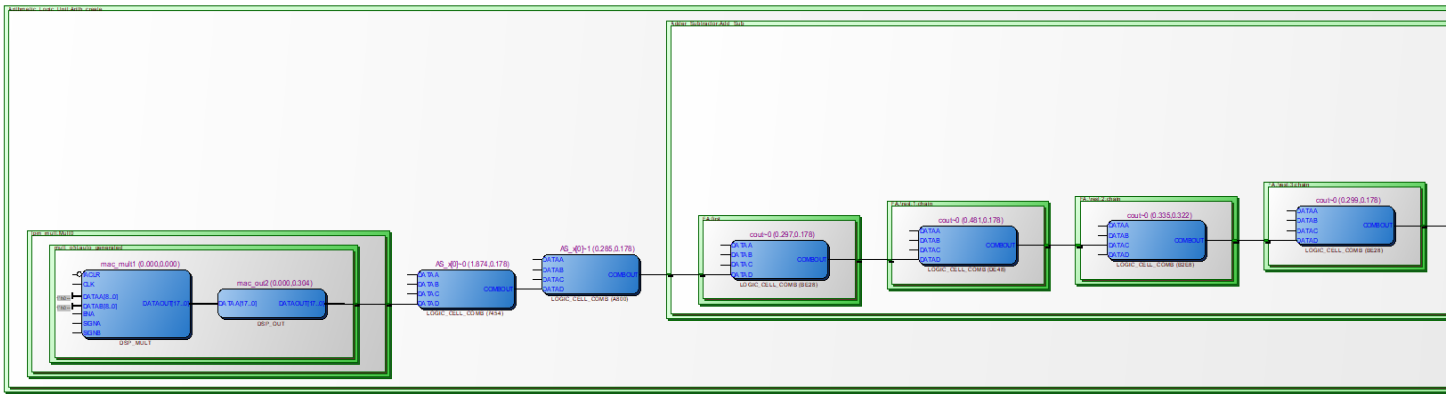
- תדר השעון המקסימלי שקיבלנו הוא 95 Mhz:

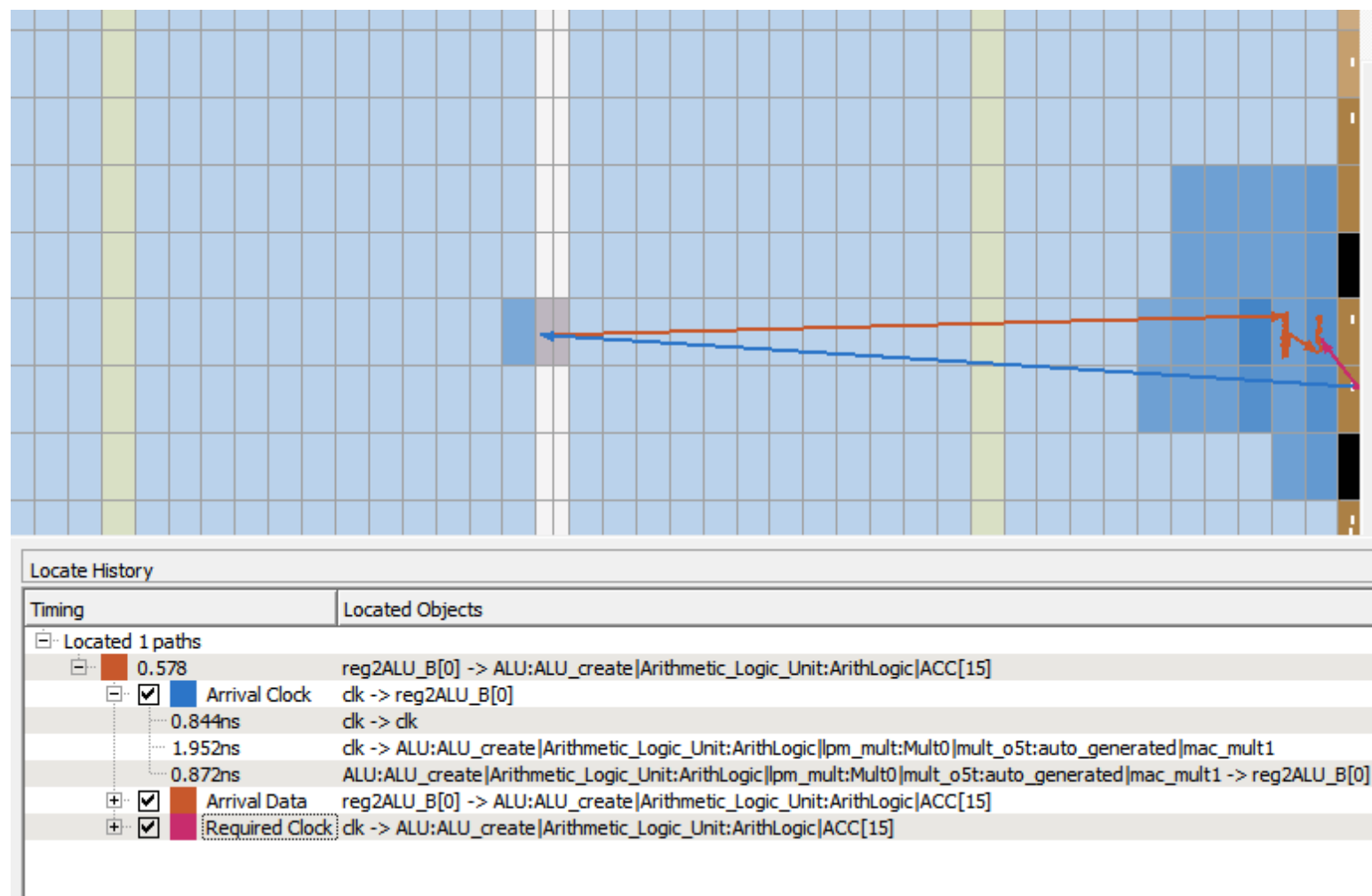
Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	95.95 MHz	95.95 MHz	clk	

לכן הגדרנו בקובץ SDC את זמן המחזור כ-

$$T = \frac{1}{95\text{Mhz}} = 10.526\text{ ns} \cong 11\text{ ns}$$

- ניתוח המסלול הקריטי:
המסלול הקריטי הוא של פקודת ה-MAC, ככל הנראה בעקבות השהיות שנוצרות מהרגיסטר בכניסה למכפל ופעולת המכפל עצמה.





- מבין ה- Entities הראשיים שלנו המסלול הקצר ביותר הוא המסלול של ה-shifter והתדר המקסימלי שניתן לקבל בו הוא 138 Mhz:

Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	138.41 MHz	138.41 MHz	clk	

- אנו משערים כי המסלול הכי קצר בכלל המערכת הוא של אחת מפעולות ה-bitwise, OR או AND (XOR ככל הנראה מעט ארוכה יותר), זאת מכיוון שהן מבוצעות ע"י מימוש פשוט של שער לוגי וללא פעולות אריתמטיות מסובכות.

- ניתוח השימוש הלוגי top:

Flow Summary	
Flow Status	Successful - Fri Jun 05 09:51:22 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	Test
Top-level Entity Name	top
Family	Cyclone II
Device	EP2C50U484C7
Timing Models	Final
Total logic elements	358 / 50,528 (< 1 %)
Total combinational functions	337 / 50,528 (< 1 %)
Dedicated logic registers	74 / 50,528 (< 1 %)
Total registers	74
Total pins	43 / 294 (15 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	1 / 172 (< 1 %)
Total PLLs	0 / 4 (0 %)

- ניתוח השימוש הלוגי shifter:

Flow Summary	
Flow Status	Successful - Fri Jun 05 10:38:18 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	Test
Top-level Entity Name	top_shifter
Family	Cyclone II
Device	EP2C50U484C7
Timing Models	Final
Total logic elements	156 / 50,528 (< 1 %)
Total combinational functions	145 / 50,528 (< 1 %)
Dedicated logic registers	35 / 50,528 (< 1 %)
Total registers	35
Total pins	43 / 294 (15 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	0 / 172 (0 %)
Total PLLs	0 / 4 (0 %)

- ניתוח השימוש הלוגי Arithmetic unit:

Flow Summary	
Flow Status	Successful - Fri Jun 05 10:27:19 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	Test
Top-level Entity Name	top_arith
Family	Cyclone II
Device	EP2C50U484C7
Timing Models	Final
Total logic elements	180 / 50,528 (< 1 %)
Total combinational functions	159 / 50,528 (< 1 %)
Dedicated logic registers	70 / 50,528 (< 1 %)
Total registers	70
Total pins	43 / 294 (15 %)
Total virtual pins	0
Total memory bits	0 / 594,432 (0 %)
Embedded Multiplier 9-bit elements	1 / 172 (< 1 %)
Total PLLs	0 / 4 (0 %)

מקרה בוחן שני - Hardware Test Case

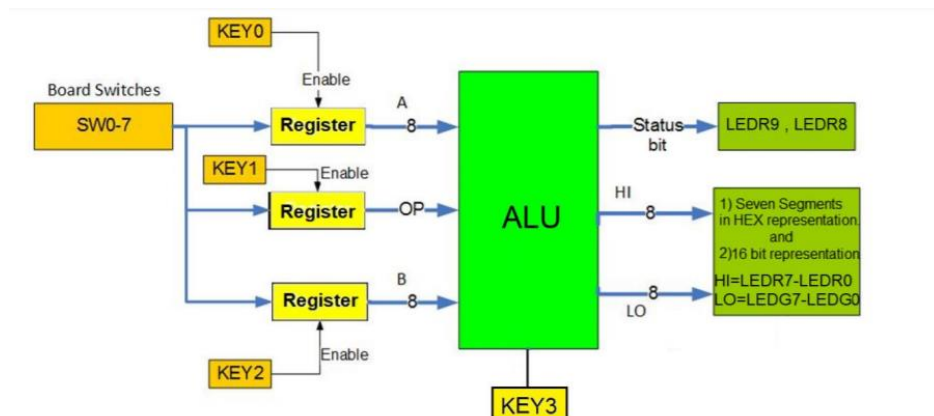


Figure 3: Hardware Test Case

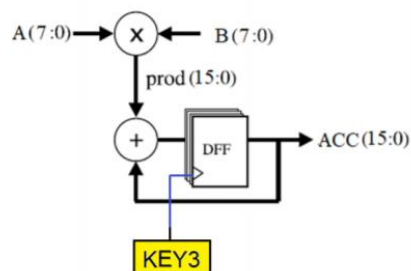


Figure 4: Hardware of MAC circuit

ביצענו אופטימיזציה של הקוד על מנת שיעבוד כראוי על ה-FPGA, והיא הוספת buffer שישמור את התוצאה האחרונה וכך כאשר מבצעים MAC RST הפלט יהיה התוצאה האחרונה לפני כן.

כמו כן היינו צריכים להוסיף סיגנלים שיתרגמו את תוצאת החישוב למסך ה-HEX, בקובץ ה-top.

ביצוע בדיקות Signal Tap:

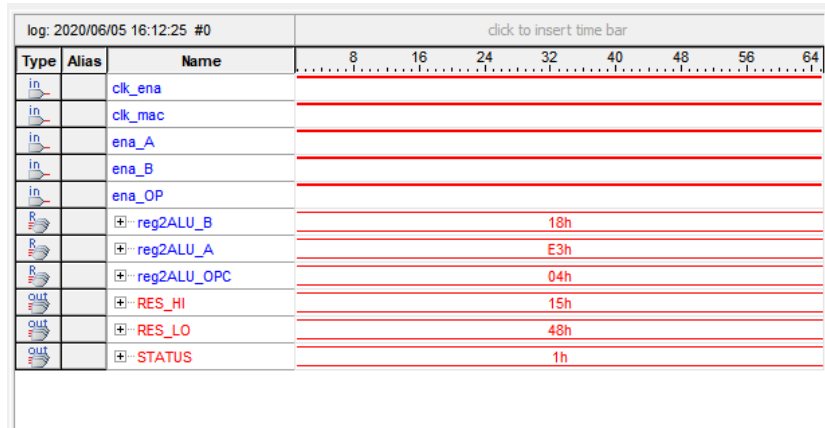
Clk_mac = key3

Enable_b = key2

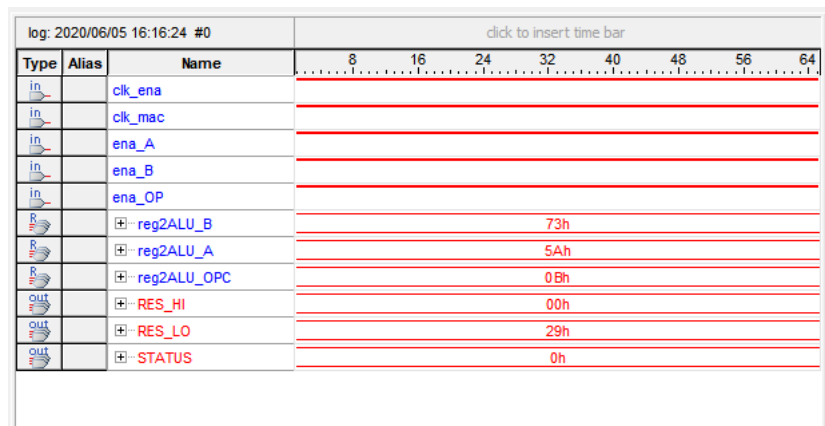
Enable_opc = key1

Enable_a = key0

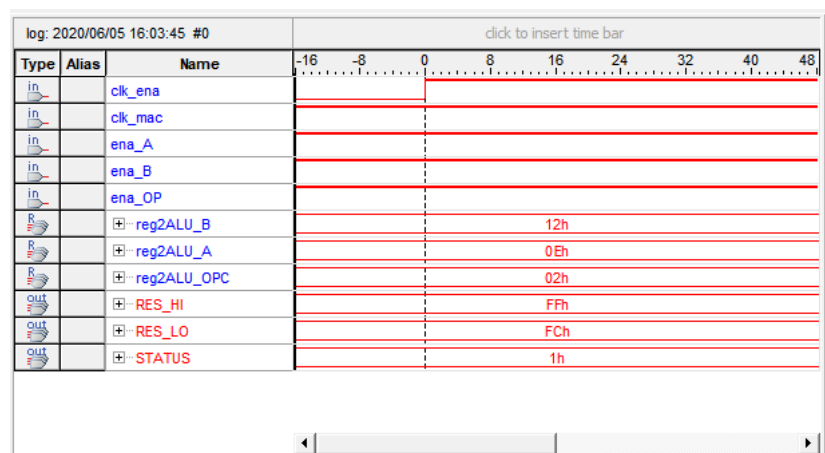
בדיקת הפעולה MULT



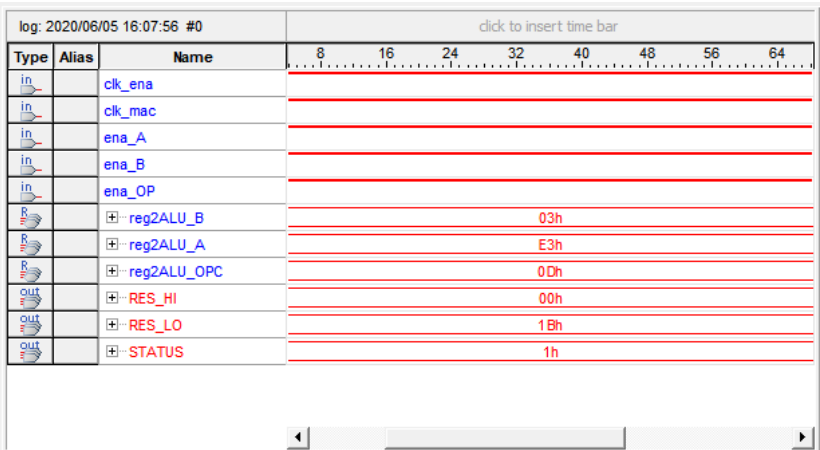
בדיקת הפעולה XOR



בדיקת הפעולה SUB



בדיקת הפעולה RLC



נספחים

דוח מטלה 1 וקבצי RTL ומסלול קריטי:

CPU Architecture

Task 1 report

ALU Design with VHDL

עומר לוקסמבורג 205500390

עילי נוריאל 312538580

תוכן עניינים

3	הגדרת תכנון המערכת
5	תיאור הרכיבים
6	תיאור הבדיקות
11	File based simulation of DUT

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 (LOW ו-HIGH).

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:

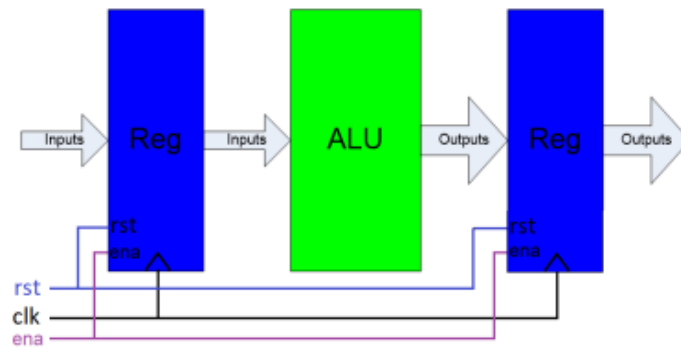


Figure 1: Overall DUT structure

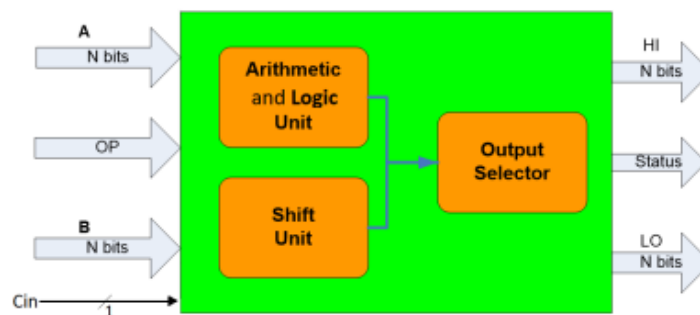


Figure 2: ALU structure

חיבורים:

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

C_{in} – כניסה של ביט נוסף (carry).

OPCODE – בחירת הפעולה שתבצע על הכניסות הנקבעת באופן הבא:

Instruction	Opcode	Operation	Note
ADD	00001	$RES(HI,LO) = A + B$	As in lab1 requirement (see FAQ file)
SUB	00010	$RES(HI,LO) = A - B$	As in lab1 requirement (see FAQ file)
ADDC	00011	$RES(HI,LO) = A + B + C_{in}$	As in lab1 requirement (see FAQ file)
MULT	00100	$RES(HI,LO) = A * B$	Multiply two integer unsigned numbers (Result is $N*2$ bits)
MAC	00101	$ACC = ACC + A * B$ $RES(HI,LO) = ACC$	Multiply Accumulate (MAC is internal $N*2$ bits register) integer unsigned numbers
MAC_RST	00110	$MAC = 0$	Reset MAC
MAX	00111	$RES(0,LO) = \text{Max}(A, B)$	maximum between A and B
MIN	01000	$RES(0,LO) = \text{Min}(A, B)$	minimum between A and B
AND	01001	$RES(0,LO) = \text{AND}(A, B)$	Bitwise operation
OR	01010	$RES(0,LO) = \text{OR}(A, B)$	Bitwise operation
XOR	01011	$RES(0,LO) = \text{XOR}(A, B)$	Bitwise operation
RLA	01100	$RES(0,LO) = \text{rotation result}$	Rotate left A arithmetically B(2..0) times
RLC	01101	$RES(0,LO) = \text{rotation result}$	Rotate left A through carry B(2..0) times
RRA	01110	$RES(0,LO) = \text{rotation result}$	Rotate right A arithmetically B(2..0) times
RRC	01111	$RES(0,LO) = \text{rotation result}$	Rotate right A through carry B(2..0) times

Table 1 : ALU Op Codes

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

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

Status Flag Name	Condition	Indicator (status ₁ , status ₀) ₂
C	Last OP produces carry	01
Z	Last OP result is zero	10
	Last OP produces both (C&Z)	11
	else	00

Table 2 : Status Bus

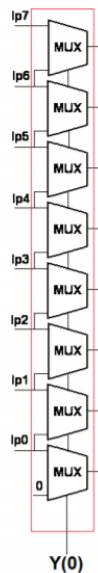
תיאור הרכיבים

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

aux_package	top			
	ALU			
	Selector	Shift_Unit		Arithmetic_Logic_Unit
		RRA	RLA	Adder_Substructor
		Yblock	Yblock	FA
		MUX2	MUX2	

הערות – לגבי המודולים ממעבדה 1:

- מודול ה-Yblock כתוב בצורה גרית ומתאר שכבה אחת ברכיב ה-shifter (כמתואר בדרישות המעבדה הראשונה) כלומר עמודה כזו:



- מודול ה-FA מתאר רכיב Full Adder לשני ביטים ומודול ה-MUX2 מתאר מרבב 1 → 2.

תיאור הלוגיקה של המודולים בעמוד הבא

תיאור הבדיקות

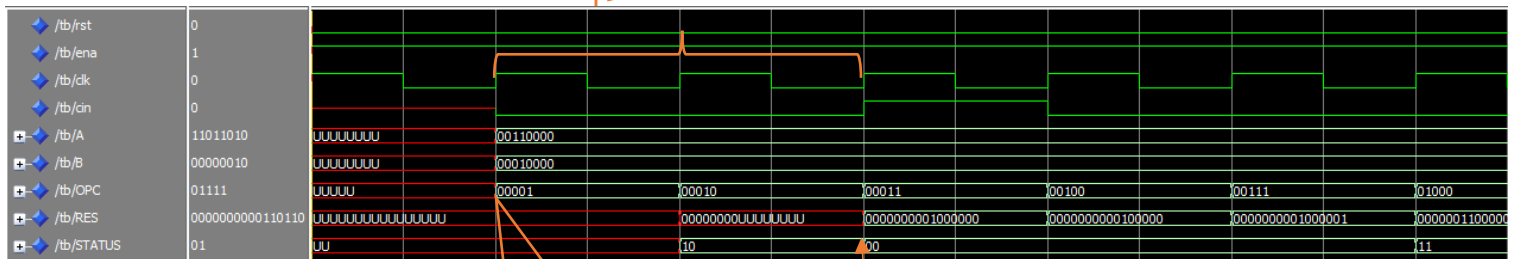
File based simulation of DUT

כתבנו מודול tb היוצר מקובץ ה-input.txt סימולציה למערכת.
בקובץ ה-input הכנסנו תרחישים לכל הפקודות השונות אותן המערכת צריכה לבצע (כל ה-opcodes),
וכן מקרי קצה.
כלומר הוא מכיל ערכים לכל הקלטים של המערכת. כמו כן המודול tb כותב את הפלטים לקובץ
.output.txt

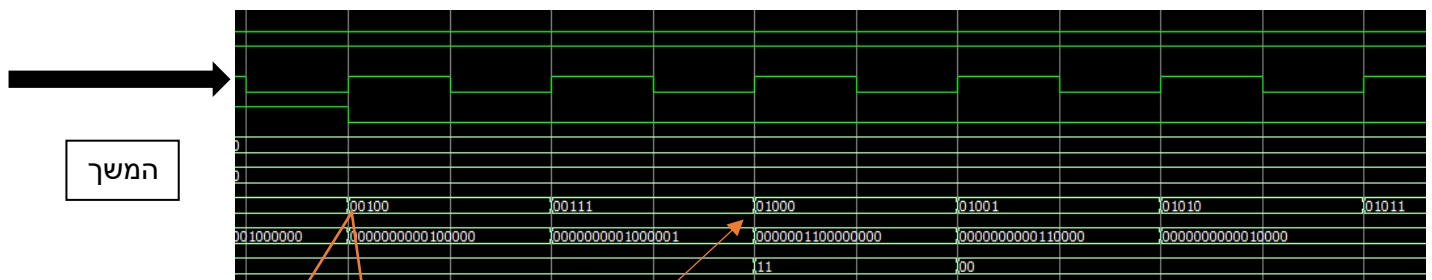
ניתוח תוצאות הסימולציה בתצוגת wave form :

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

שני מחזורי שעות

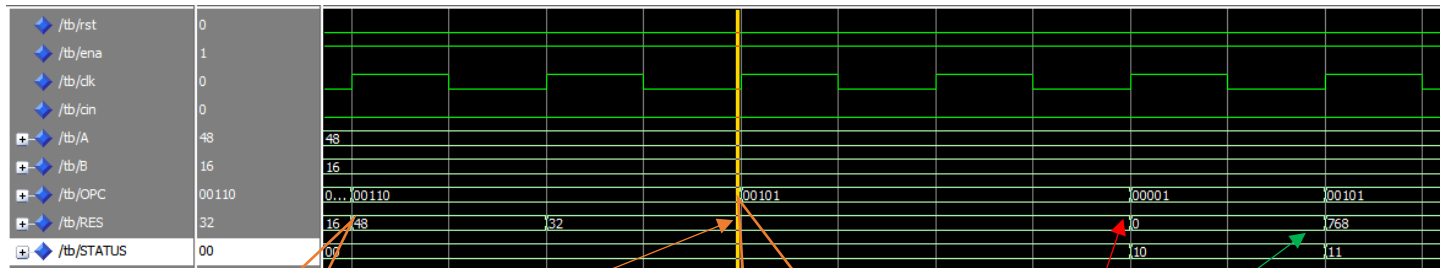


ה-opcode משתנה ל-1 כלומר פעולת ADD.
נשים לב כי הפלט התואם לפקודה הזו ייצא רק
לאחר שני מחזורים שעות בעקבות ההשהיה של
רגיסטרי הכניסה והיציאה.
לאחר שני מחזורי שעות ערך היציאה הוא אכן
 $A + B = 64 = (1000000)_2$



ה-opcode משתנה ל-100 כלומר פעולת MULT.
נשים לב כי הפלט התואם לפקודה הזו ייצא רק לאחר שני מחזורים שעות בעקבות
ההשהיה של רגיסטרי הכניסה והיציאה.
לאחר שני מחזורי שעות ערך היציאה הוא אכן
 $A \cdot B = 1536 = (11000000000)_{10}$, כאשר $A=32$, $B=48$

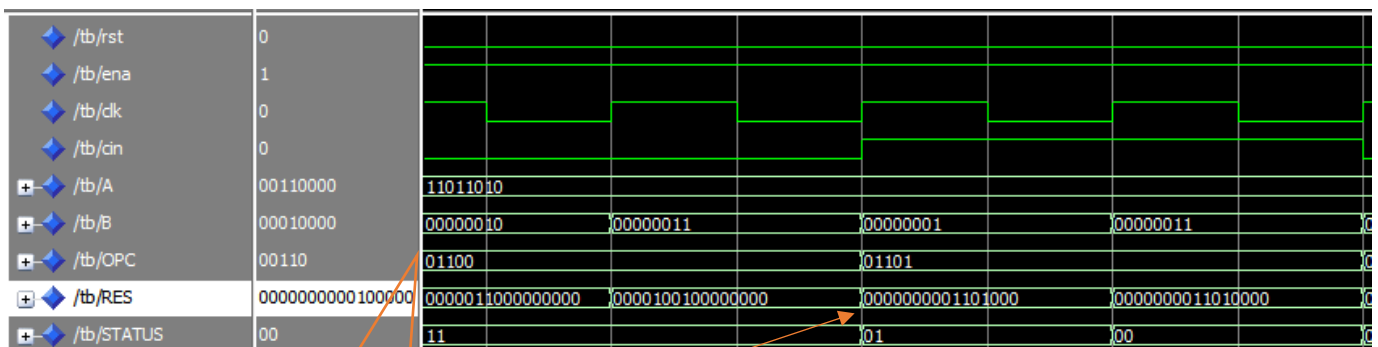
בדיקת הפעולות MAC, MAC_RST :



1. opcode משתנה ל110 כלומר פעולת MAC_RST. ערך RES לא משתנה

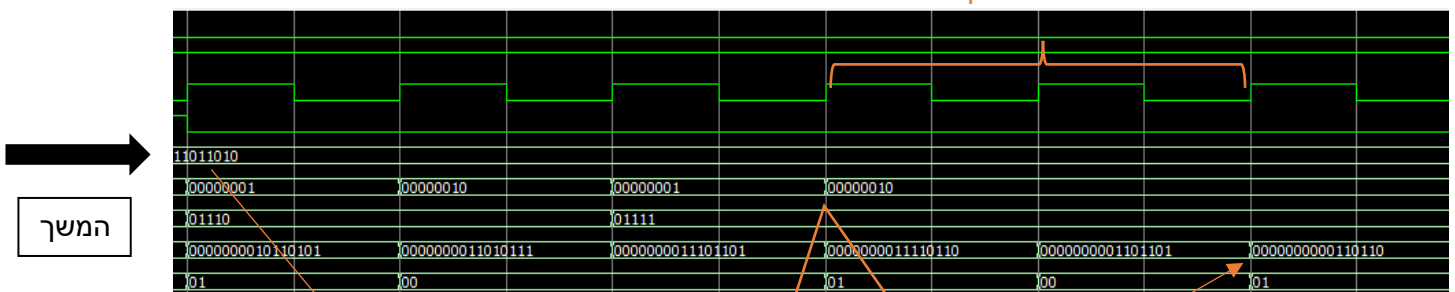
2. opcode משתנה ל101 כלומר פעולת MAC ונשאר במשך עוד שני מחזורים. החץ האדום מציג את המצב בו ACC=0 - הערך 0 הוא המוצא עדיין. החץ הירוק מציג את הערך שבו ACC כבר שונה מ-0.

בדיקת הפעולות RLA, RRA, RRC, RLC :



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

שני מחזורי שעון



המשך

opcode הוא 01111 כלומר פעולת RRC. B=2 A=11011010 - כלומר יתבצעו 2 הזזות על A כאשר cin=0. נשים לב שזה אומר שביט Carry צריך להיות 1 (הביט האחרון שיצא מ-RES(LO)). בנוסף התוצאה תהיה RES(LO)=01101101 כאשר ה-0 האדום הוא ביט הקארי ומשמאל אליו ביט ה-LSB.

