



תום קיסוס – 206018749

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

דן בן עמי – 316333079

אוניברסיטת בן – גוריון

הפקולטה להנדסה

המחלקה להנדסת מחשבים

דו"ח מסכם – מטלת ארכיטקטורת יחידת עיבוד מרכזית

Final project

מגישים: דן בן עמי – 316333079

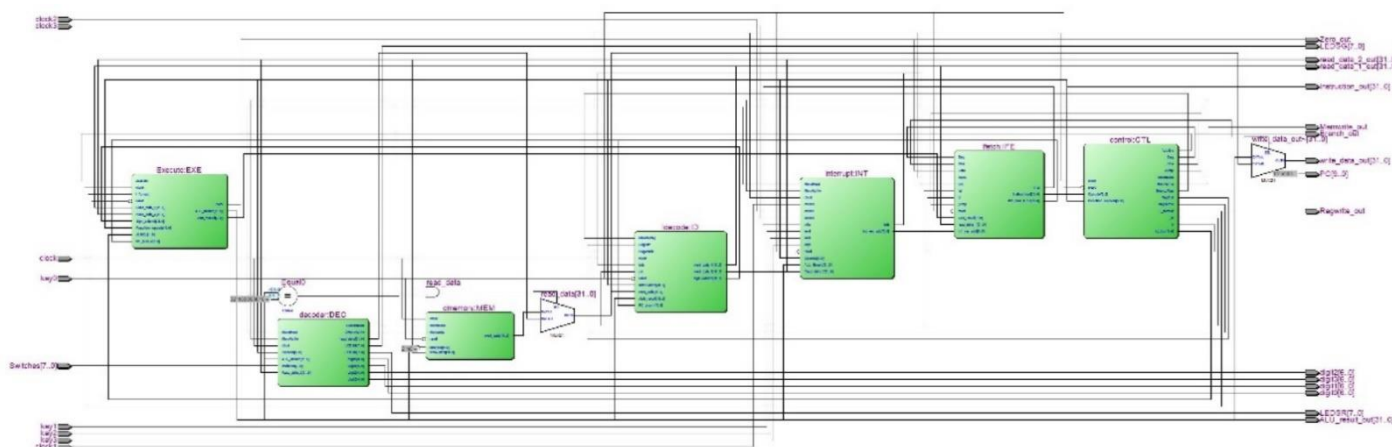
תום קיסוס – 206018749

מדריך: חנן ריבוא, מור אבי אהרון

תאריך הגשה: 16.09.20



Top level block diagram - RTL viewer - top level entity:



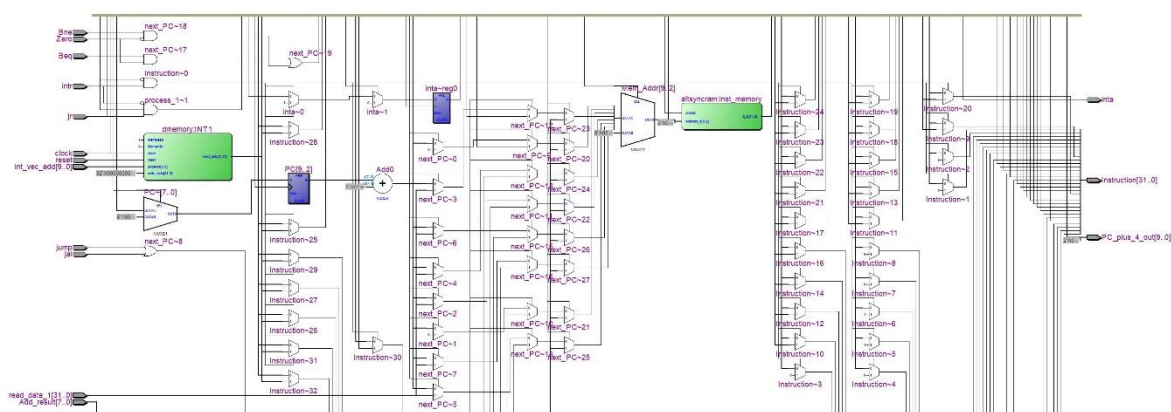
איור 1 RTL viewer - top level

Instruction fetch:

בלוק זה אחראי על הבאת הפקודה במיקום של הרגיסטר PC (program counter) מזיכרון הפקודות (instructions memory) אל הבלוק Decode.

במקביל להבאת הפקודה מתבצע קידום של רגיסטר PC ב-4 (היות ואורך כל פקודה הוא 4 בייטים) ומערכת לוגית צירופית הקובעת את ערך הרגיסטר PC בשעון הבא בהתאם לקווי הבקרה ולערך פקודת ההסתעפות.

RTL viewer:

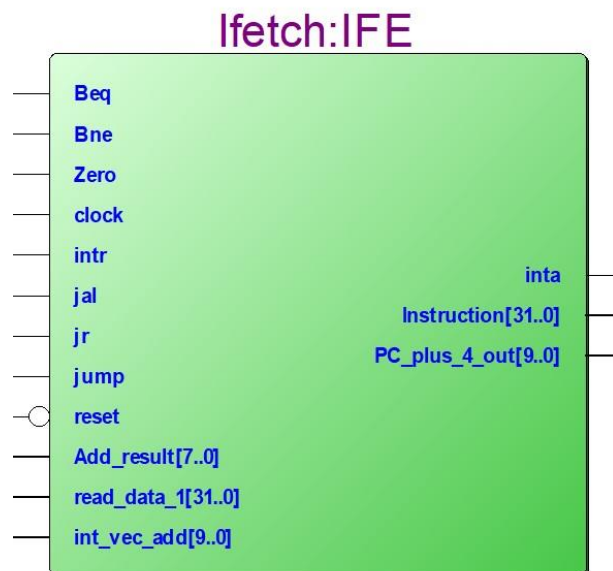


איור 2 RTL viewer - IFETCH

Logic usage:

Flow Summary	
Flow Status	Successful - Wed Sep 09 12:59:09 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	Ifetch
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	73 / 18,752 (< 1 %)
Total combinational functions	73 / 18,752 (< 1 %)
Dedicated logic registers	9 / 18,752 (< 1 %)
Total registers	9
Total pins	102 / 315 (32 %)
Total virtual pins	0
Total memory bits	57,344 / 239,616 (24 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 3 Ifetch Logic usage

Graphical description:

איור 4 Ifetch Graphical description

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
Instruction	32	buffer	פקודה מהזיכרון
jump	1	IN	קו בקרה העולה ל-1 בהינתן פקודת JUMP
PC_plus_4_out	10	OUT	הערך הבא של ה-PC במידה ואין הסתעפות.
Add_result	8	IN	הכתובת להסתעפות בהינתן פקודת הסתעפות.
Beq	1	IN	קו בקרה העולה ל-1 בהינתן פקודת BEQ
Bne	1	IN	קו בקרה העולה ל-1 בהינתן פקודת BNE
Zero	1	IN	קו בקרה העולה ל-1 אם תוצאת פעולת ה-ALU היא 0.
jal	1	in	קו בקרה העולה ל-1 בהינתן פקודת JAL
jr	1	in	קו בקרה העולה ל-1 בהינתן פקודת JR
read_data_1	32	in	המידע שנקרא מהרגיסטר
clock	1	IN	שעון
reset	1	IN	קו בקרה העולה ל-1 בהינתן לחיצה על key0. מביא את ה-PC לתחילת התוכנית.
intr	1	buffer	קו בקרה העולה ל-1 בהינתן בקשת פסיקה
int_vec_add	10	in	הכתובת המפנה ל dmemory אל המיקום בה נמצא כתובת רוטנית השירות.
inta	1	buffer	קו בקרה העולה ל-1 כאשר המעבד מטפל בפסיקה

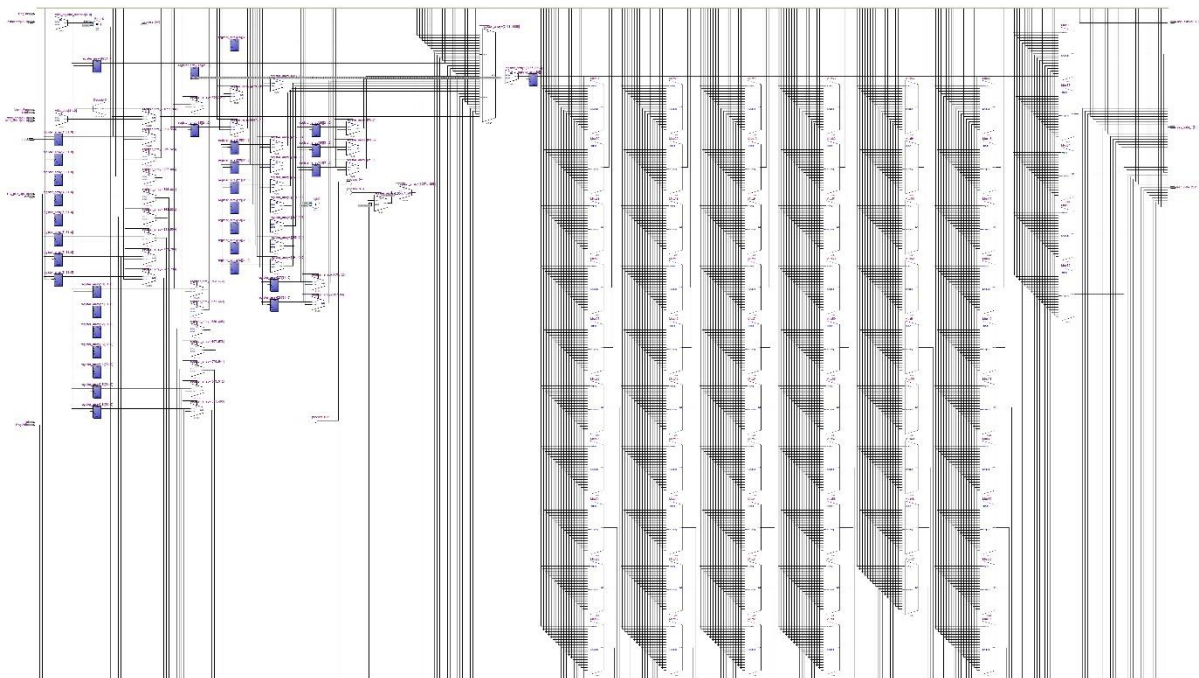
Table 1 – port table Fetch



Instruction decode:

בלוק זה אחראי על פיענוח הפקודה מתוך 32 הביטים המובאים כפקודה ב Instruction fetch ולאחר מכן מוציא מהרגיסטר פייל את ערכי הרגיסטרים המתאימים לפקודה זו. במקביל, בשלב זה מתבצעת הרחבת סימן (sign extend) של הקבוע הנתון בפקודה לכדי 32 ביטים על מנת שיהיה ניתן לבצע באמצעותו פעולות שונות בבלוק הבא (למשל בפקודות i-type, j-type). בנוסף, בבלוק זה אנו מבצעים גם את פעולת ה write back כאשר נדרש לבצע כתיבה של תוצאה אל הרגיסטרים (לדוגמא בפקודות מסוג R-type).

RTL viewer:

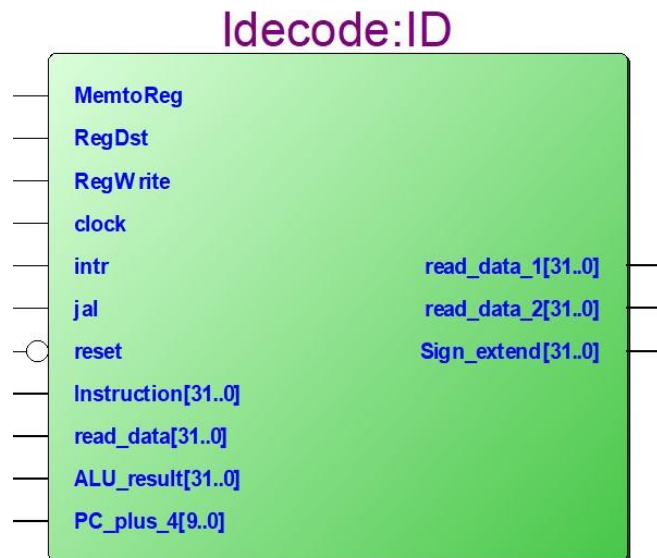


איור 5 Idecode RTL viewer

Logic usage:

Flow Summary	
Flow Status	Successful - Wed Sep 09 13:05:52 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	Idecode
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	1,485 / 18,752 (8 %)
Total combinational functions	1,485 / 18,752 (8 %)
Dedicated logic registers	992 / 18,752 (5 %)
Total registers	992
Total pins	209 / 315 (66 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 6 Idecode logic usage

Graphical description:

איור 7 Idcode Graphical description

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
read_data_1	32	OUT	המידע שנקרא מהרגיסטר
read_data_2	32	OUT	המידע שנקרא מהרגיסטר
Instruction	32	IN	פקודה לביצוע
read_data	32	IN	המידע שנקרא מ-Data memory.
ALU_result	32	IN	תוצאת החישוב של ALU
RegWrite	1	IN	קו בקרה לכתיבה לרגיסטרים
MemtoReg	1	IN	קו הבקרה השולט על בחירת המידע שנכתב לרגיסטרים.
RegDst	1	IN	קו בקרה לבחירת הרגיסטר לכתיבה.
Sign_extend	32	OUT	תוכן שדה immediate לאחר הרחבה ל 32 ביט
jal	1	in	קו בקרה העולה ל-1 בהינתן פקודת JAL
intr	1	in	קו בקרה העולה ל-1 בהינתן בקשת פסיקה
PC_plus_4	10	in	הערך הבא של ה-PC במידה ואין הסתעפות.
clock	1	IN	שעון
reset	1	IN	קו בקרה העולה ל-1 בהינתן לחיצה על key0. מאתחל כל רגיסטר לאחסן את מספרו הסידורי ב- register file.

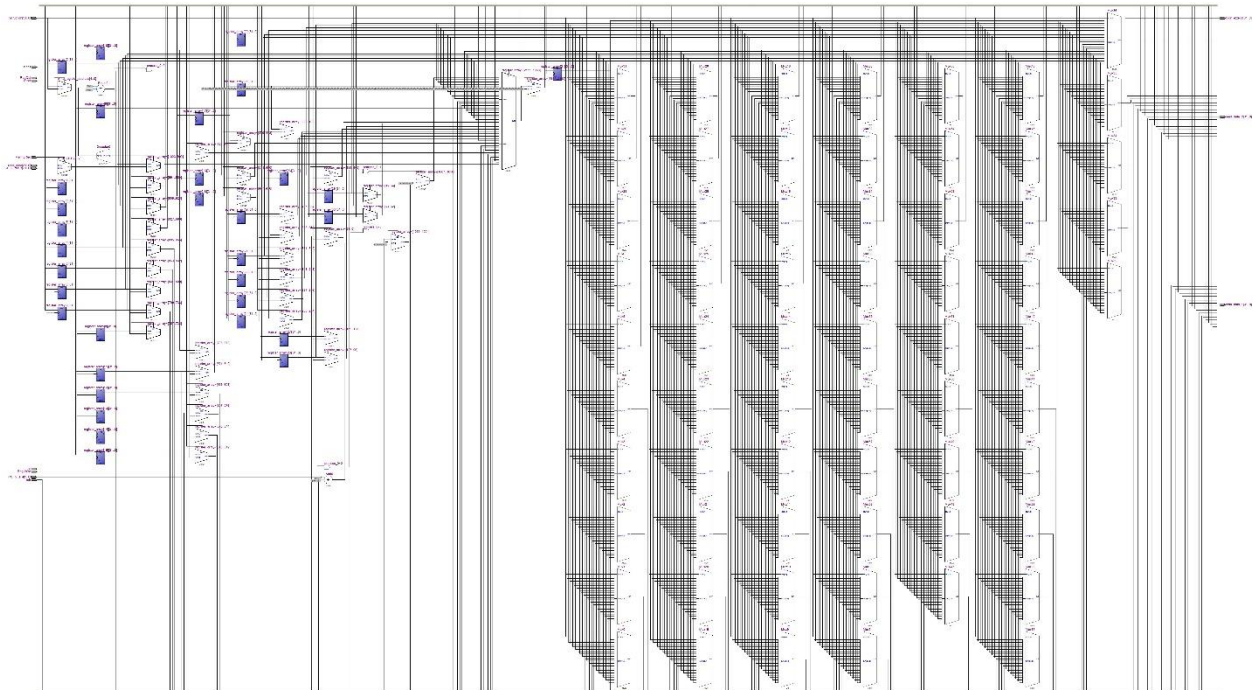
Table 2 – port table Decode



Control:

בלוק זה אחראי על הפעלת קווי הבקרה לרכיבי MUX השונים במערכת על מנת שינתבו את המידע (Data path) בהתאם לפקודה הנוכחית. בנוסף בלוק זה מעביר גם את קווי הבקרה לרכיב ה ALU שבאמצעותם ה ALU מפעיל את האופרציה הנכונה על האופרנדים השונים בהתאם לפקודה.

RTL viewer:



איור 8 control RTL viewer

Logic usage:

Flow Summary	
Flow Status	Successful - Wed Sep 09 13:05:52 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	Icode
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	1,485 / 18,752 (8 %)
Total combinational functions	1,485 / 18,752 (8 %)
Dedicated logic registers	992 / 18,752 (5 %)
Total registers	992
Total pins	209 / 315 (66 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 9 control logic usage

Graphical description:

Graphical description of control 10 איור

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
Opcode	6	IN	שדה ה- opcode מתוך ה- instruction
Function_opcode	6	IN	שדה ה- Function_opcode מתוך ה- instruction
RegDst	1	OUT	קו בקרה לבחירת הרגיסטר לכתובה.
ALUSrc	1	OUT	קו בקרה לבחירת מבוא ה- ALU.
MemtoReg	1	OUT	קו הבקרה השולט על בחירת המידע שנכתב לרגיסטרים.
RegWrite	1	OUT	קו בקרה לכתובה לרגיסטרים
MemRead	1	OUT	קו בקרה המורה אם לקרוא מהזיכרון
MemWrite	1	OUT	קו בקרה המורה אם לכתוב להזיכרון
Beq	1	buffer	קו בקרה העולה ל-1 בהינתן פקודת BEQ
Bne	1	buffer	קו בקרה העולה ל-1 בהינתן פקודת BNE
Jump	1	OUT	קו בקרה העולה ל-1 בהינתן פקודת JUMP
ALUOp	2	OUT	מורה ל- ALU איזו פעולה לבצע.
jal	1	buffer	קו בקרה העולה ל-1 בהינתן פקודת JAL
jr	1	OUT	קו בקרה העולה ל-1 בהינתן פקודת JR
i_format	1	OUT	קו בקרה העולה ל-1 בהינתן פקודת I-format.
clock	1	IN	שעון
reset	1	IN	קו בקרה העולה ל-1 בהינתן לחיצה על key0. מביא את ה- PC לתחילת התוכנית.

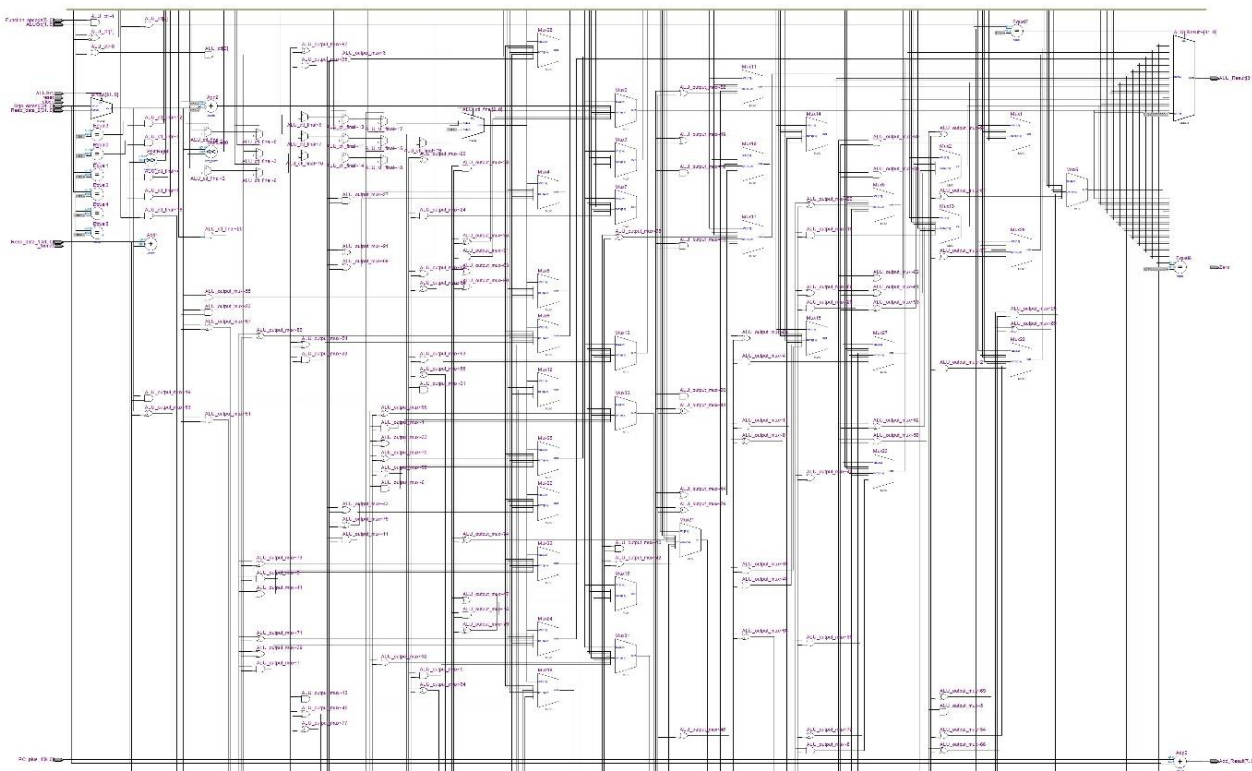
Table 3 – port table Execute



Exectute:

בלוק זה אחראי על ביצוע הפעולות האריתמטיות בין הערכים השונים שהיו ברגיסטרים או בחלק של הקבוע בפקודה. בעזרת הקו בקרה המתאים נבחרת הפעולה המתאימה ב ALU מבין מספר פעולות כמו חיבור, חיסור, AND, OR, הזזה וכו'. הבלוק מכיל בנוסף ל ALU גם רכיב Adder על מנת לחשב את ערך הכתובת להסתעפות במידה והפקודה היא פקודת branch והסתעפות אכן נלקחת.

RTL viewer:

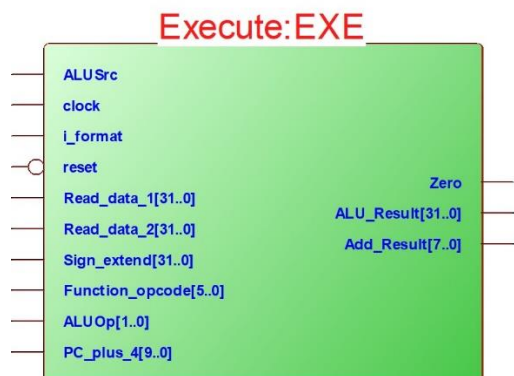


איור 11 Execute RTL viewer

Logic usage:

Flow Summary	
Flow Status	Successful - Mon Aug 24 17:11:57 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	Execute
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	607 / 18,752 (3 %)
Total combinational functions	607 / 18,752 (3 %)
Dedicated logic registers	0 / 18,752 (0 %)
Total registers	0
Total pins	159 / 315 (50 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 12 Execute logic usage

Graphical description:

איור 13 Execture Graphical description

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
read_data_1	32	IN	המידע שנקרא מהרגיסטר
read_data_2	32	IN	המידע שנקרא מהרגיסטר
Sign_extend	32	IN	תוכן שדה immediate לאחר הרחבה ל 32 ביט
Function_opcode	6	IN	שדה ה- Function_opcode מתוך ה- instruction
ALUOp	2	IN	מורה ל- ALU איזו פעולה לבצע.
ALUSrc	1	IN	קו בקרה לבחירת מבוא ה- ALU.
Zero	1	OUT	קו בקרה העולה ל-1 אם תוצאת פעולת ה- ALU היא 0-.
ALU_result	32	OUT	תוצאת החישוב של ALU
Add_Result	8	OUT	הכתובת לקפיצה בהינתן פקודת Branch.
PC_plus_4	10	IN	הכתובת הבאה לפקודת Branch.
i_format	1	IN	קו בקרה העולה ל-1 בהינתן פקודת i-format.
clock	1	IN	שעון
reset	1	IN	קו בקרה העולה ל-1 בהינתן לחיצה על key0. מביא את ה- PC לתחילת התוכנית.

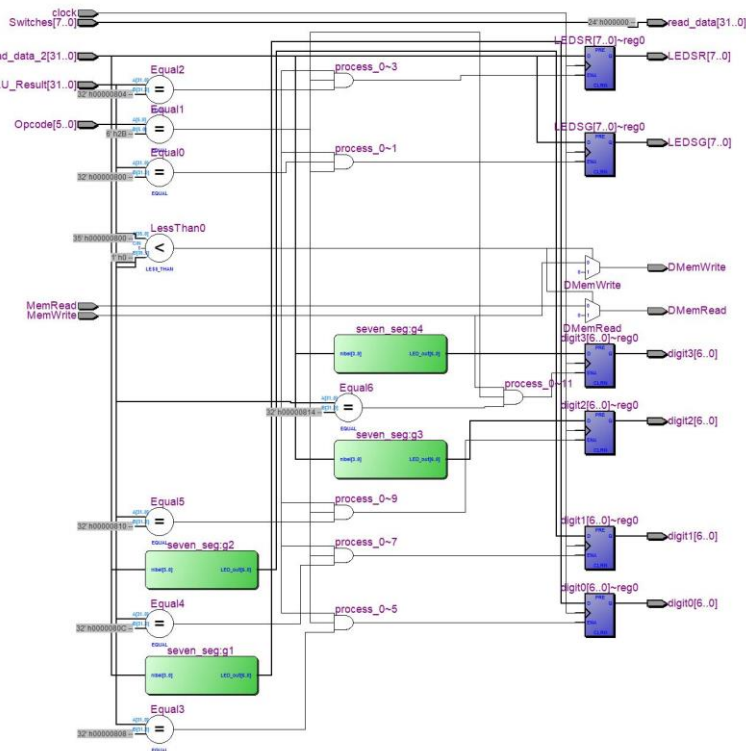
Table 4 – port table Execute



I/O decoder:

בלוק זה אחראי על הממשק בין המערכת למשתמש, הוא אחראי על פיענוח הקלט הנכנס בקלטים כאשר התוכנית בזיכרון מבקשת לטעון את קלטי המשתמש וכן על הוצאת הפלט אל הלידים והמסך בהתאם לפקודות התוכנית המאוחסנת בזיכרון. בנוסף, בלוק זה אחראי על פיענוח יעד הפקודות LW, SW, האם הן מתכוונות להתממשקות עם הזיכרון נתונים או עם מתכוונות להתממשקות עם הקלט/פלט של המשתמש.

RTL viewer:



איור 14 I/O decoder RTL viewer

Logic usage:

Flow Summary	
Flow Status	Successful - Mon Aug 24 17:15:59 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	decoder
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	65 / 18,752 (< 1 %)
Total combinational functions	28 / 18,752 (< 1 %)
Dedicated logic registers	44 / 18,752 (< 1 %)
Total registers	44
Total pins	159 / 315 (50 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 15 I/O decoder Logic usage

Graphical description:

I/O decoder Graphical description / 16 איור

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
Opcode	6	IN	שדה ה- opcode מתוך ה- instruction
ALU_result	32	IN	תוצאת החישוב של ALU
Switches	8	IN	שמונת המתגים בלוח.
read_data_2	32	IN	המידע שנקרא מהרגיסטר
MemRead	1	IN	קו בקרה המורה אם לקרוא מהזיכרון
MemWrite	1	IN	קו בקרה המורה אם לכתוב להזיכרון
clock	1	IN	שעון
DMemRead	1	OUT	קו בקרה המורה אם לקרוא מהזיכרון או לקרוא מ IO
DMemWrite	1	OUT	קו בקרה המורה אם לכתוב לזיכרון או לכתוב ל IO
read_data	32	OUT	המידע שנקרא מ- Data memory.
LEDSG	8	OUT	שמונת לדים ירוקים
LEDSR	8	OUT	שמונת לדים אדומים
Digit0	7	OUT	ספרה 0 בצג
Digit1	7	OUT	ספרה 1 בצג
Digit2	7	OUT	ספרה 2 בצג
Digit3	7	OUT	ספרה 3 בצג

Table 5 – port table decoder



Data Memory:

בלוק זה אחראי על הממשק של המערכת עם הזיכרון נתונים. בפקודות כמו LW, SW המערכת יוצרת גישה לזיכרון הנתונים באמצעות בלוק זה ומביאה/מאחסנת שם את הנתונים בכתובת המחושבת בבלוק ה- Execute.

RTL viewer:

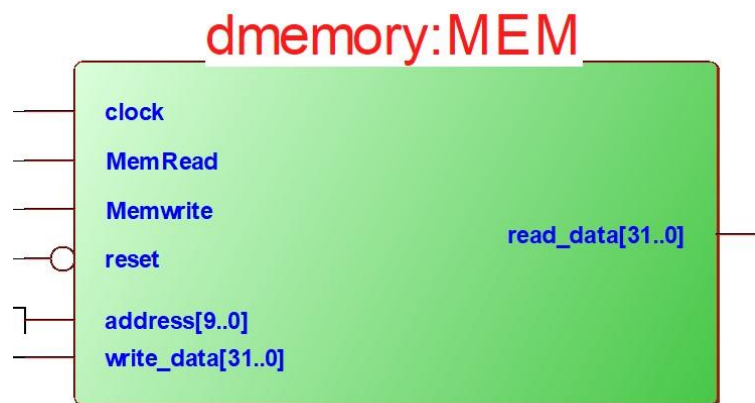


איור 17 Dmemory RTL viewer

Logic usage:

Flow Summary	
Flow Status	Successful - Mon Aug 24 17:17:14 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	dmemory
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	0 / 18,752 (0 %)
Total combinational functions	0 / 18,752 (0 %)
Dedicated logic registers	0 / 18,752 (0 %)
Total registers	0
Total pins	78 / 315 (25 %)
Total virtual pins	0
Total memory bits	32,768 / 239,616 (14 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 18 Dmemory Logic usage

Graphical description:

איור 19 Dmemory Graphical description

Port table:

שימוש	כניסה/יציאה	מספר ביטים	שם
המידע שנקרא מ- Data memory.	OUT	32	read_data
הכתובת לקריאה או כתיבה.	IN	10	address
המידע שיכתב לזיכרון.	IN	32	write_data
קו בקרה המורה אם לקרוא מהזיכרון	IN	1	MemRead
קו בקרה המורה אם לכתוב להזיכרון	IN	1	Memwrite
שעון	IN	1	clock
קו בקרה העולה ל-1 בהינתן לחיצה על key0. מביא את ה- PC לתחילת התוכנית.	IN	1	reset

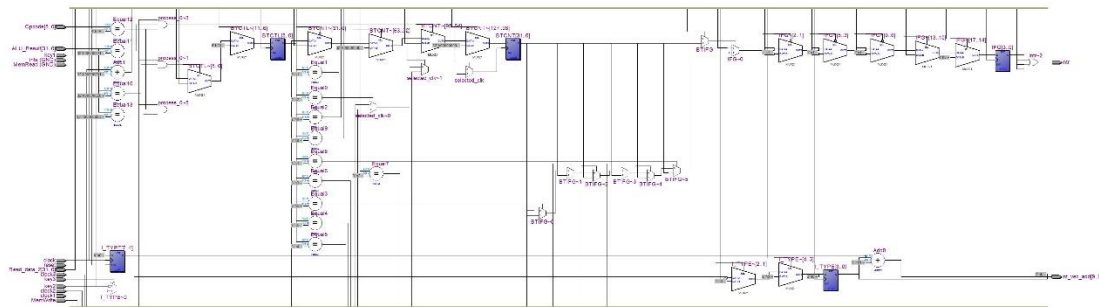
Table 6 – port table Dmemory



Interrupt controller:

רכיב זה הוא רכיב האחראי למתן פסיקות למעבד מכפתורים על לוח ה IDE ופסיקות מטיימר. הטיימר הוא טיימר בסיסי, המוציא פסיקה כאשר המונה של הטיימר מגיע לרוויה שנקבעת באמצעות רגיסטר הבקרה (BTCTL). בתוך רכיב זה נמצאים הטיימר עצמו (רגיסטר מונה, רגיסטר בקרה) וכן רגיסטר איפשור פסיקות ורגיסטר דגלי הפסיקות.

RTL viewer:

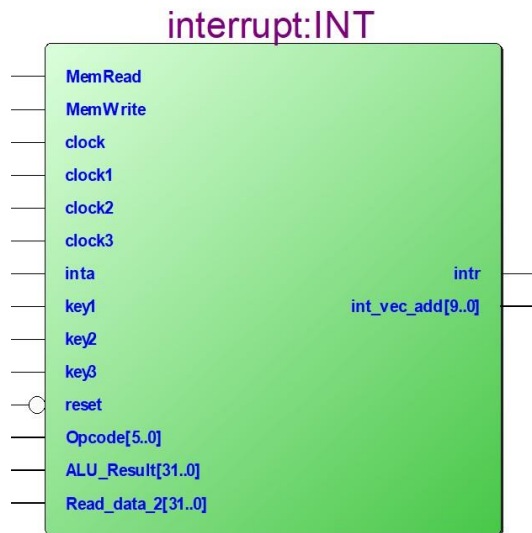


איור 20 RTL viewer - interrupt/timer

Logic usage:

Flow Summary	
Flow Status	Successful - Wed Sep 09 12:14:23 2020
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	task3final
Top-level Entity Name	interrupt
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	70 / 18,752 (< 1 %)
Total combinational functions	69 / 18,752 (< 1 %)
Dedicated logic registers	38 / 18,752 (< 1 %)
Total registers	38
Total pins	92 / 315 (29 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

איור 21 Logic usage interrupt/timer

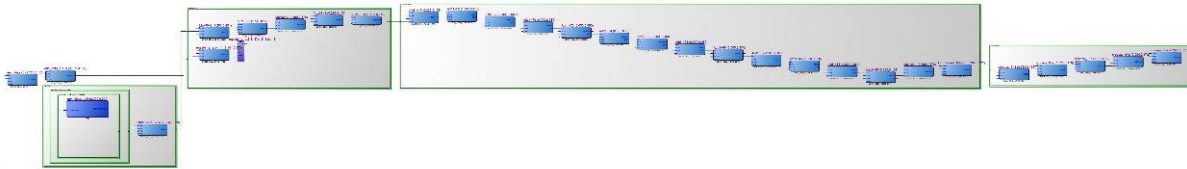
Graphical description:

איור 22 interrupt/timer Graphical description -

Port table:

שם	מספר ביטים	כניסה/יציאה	שימוש
Opcode	6	IN	שדה ה- opcode מתוך ה- instruction
ALU_result	32	IN	תוצאת החישוב של ALU
read_data_2	32	IN	המידע שנקרא מהרגיסטר
MemRead	1	IN	קו בקרה המורה אם לקרוא מהזיכרון
MemWrite	1	IN	קו בקרה המורה אם לכתוב להזיכרון
inta	1	buffer	קו בקרה העולה ל- 1 כאשר המעבד מטפל בפסיקה
Key1	1	IN	לחצן פסיקה מספר 1
Key2	1	IN	לחצן פסיקה מספר 2
Key3	1	IN	לחצן פסיקה מספר 3
clock	1	IN	שעון
reset	1	IN	קו בקרה העולה ל-1 בהינתן לחיצה על key0. מאפס את הרגיסטרים של הטיימר ובקר הפסיקות.
intr	1	buffer	קו בקרה העולה ל- 1 בהינתן בקשת פסיקה
int_vec_add	10	in	הכתובת המפנה ל dmemory אל המיקום בה נמצא כתובת רוטנית השירות.

Table 7 – port table Execute

**מסלול קריטי:**

איור 23 critical path

ניתן לראות כי המסלול הקריטי שתוכנת ה- Quartus מציגה אינו תואם את הציפיות שלנו. מהאיור עולה כי הנתיב הקריטי הוא בפקודת SW לרגיסטרי פסיקה\טיימר. אולם מניסיוננו ומהרצאות הקורס עולה כי דווקא פקודת LW אמורה להוות את הנתיב הקריטי מכיוון שזאת עוברת תהליך של חישוב הכתובת ב- ALU ולאחר מכן קוראת מ- Data memory ולאחר מכן צריכה לכתוב את המידע לרגיסטר. נתיב זה אמור להיות הארוך ביותר מפאת זמני גישה ארוכים יחסית לזיכרון המידע.

תדר מקסימלי:

Slow Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	18.13 MHz	18.13 MHz	clock	

איור 24 תדר מקסימלי

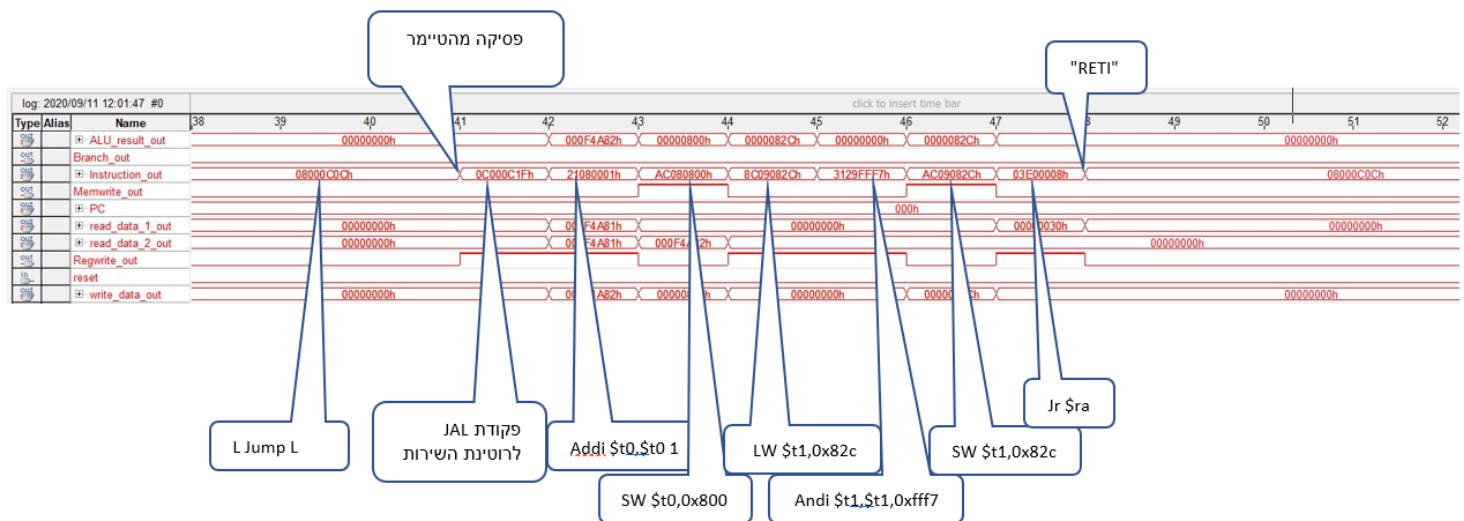
נתיב מינימלי:

הנתיב המינימלי יהיה בפקודת Jump מכיוון שזאתי לא ניגשת כלל לזיכרון ולא עוברת תהליך חישוב ב- ALU. הפקודה מעדכנת את ערך ה- PC לערך הקפיצה המבוקשת.



Signal tap:

בדוגמא מטה הרצנו את קובץ ה test2 שקיבלנו במודל לבדיקת תקינות המערכת. בדוגמא זו בחנו את מנגנון הפסיקות של הטיימר. ניתן לראות כי כאשר מתקבלת פסיקה מהטיימר, התוכנית קופצת לרוטינת השירות של הטיימר ומבצעת את הנדרש. לאחר סיום הפסיקה התוכנית חוזרת למקום ממנה התבצעה הפסיקה, כמצופה.



איור 25 Signal tap