

מטרת המעבדה:

במעבדה זו, השתמשנו בתוכנות MODELSIM ו QUARTRUS על מנת לבצע סינטזה למודלים שייצרנו במעבדות קודמות. את המעבדה הרצנו על הAKER FPGA CYCLONE של כרטיס DE10.

במעבדה זו מימושו שני מעבדים חשובים בתחום החומרה:

מעבד SINGLE CYCLE - מעבד אותו מימושו תחיליה כדי להווסף את הפקודות הנדרשות.

מעבד PIPELINE - מעבד אותו מימושו לאחר מימוש הCYCLE single, כדי לעבוד עם מעבד יעיל יותר.

כלומר במעבדה לבסוף מימושו מעבר MIPS הידוע לבצע ISA בסיסית של פקודות אסמבלי.

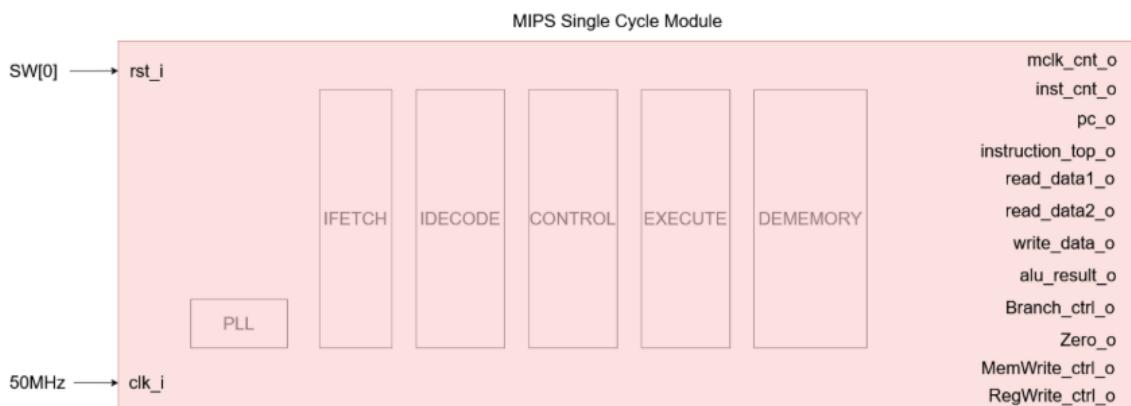


Figure 7: Single-cycle MIPS architecture top entity

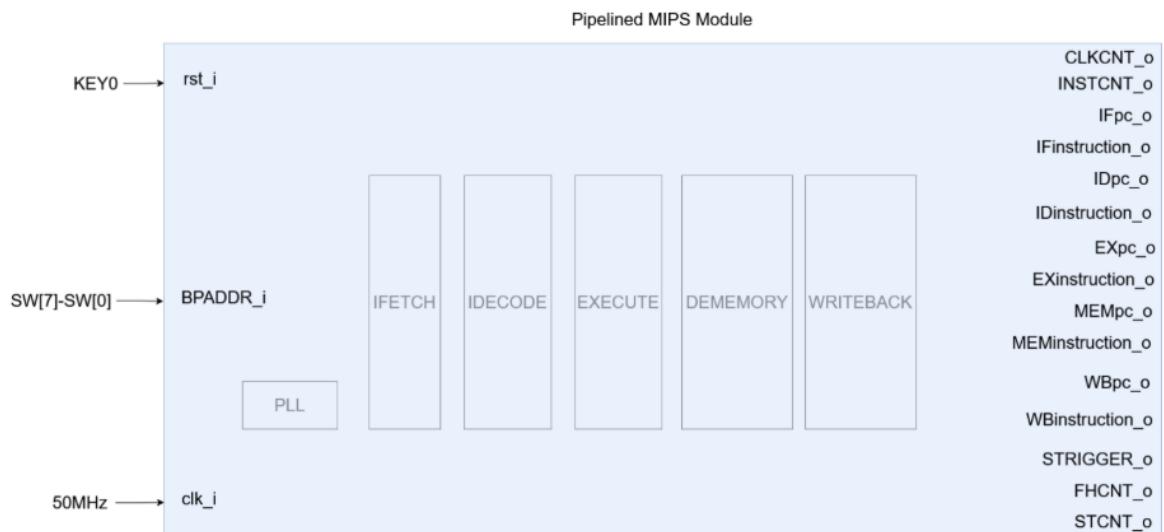
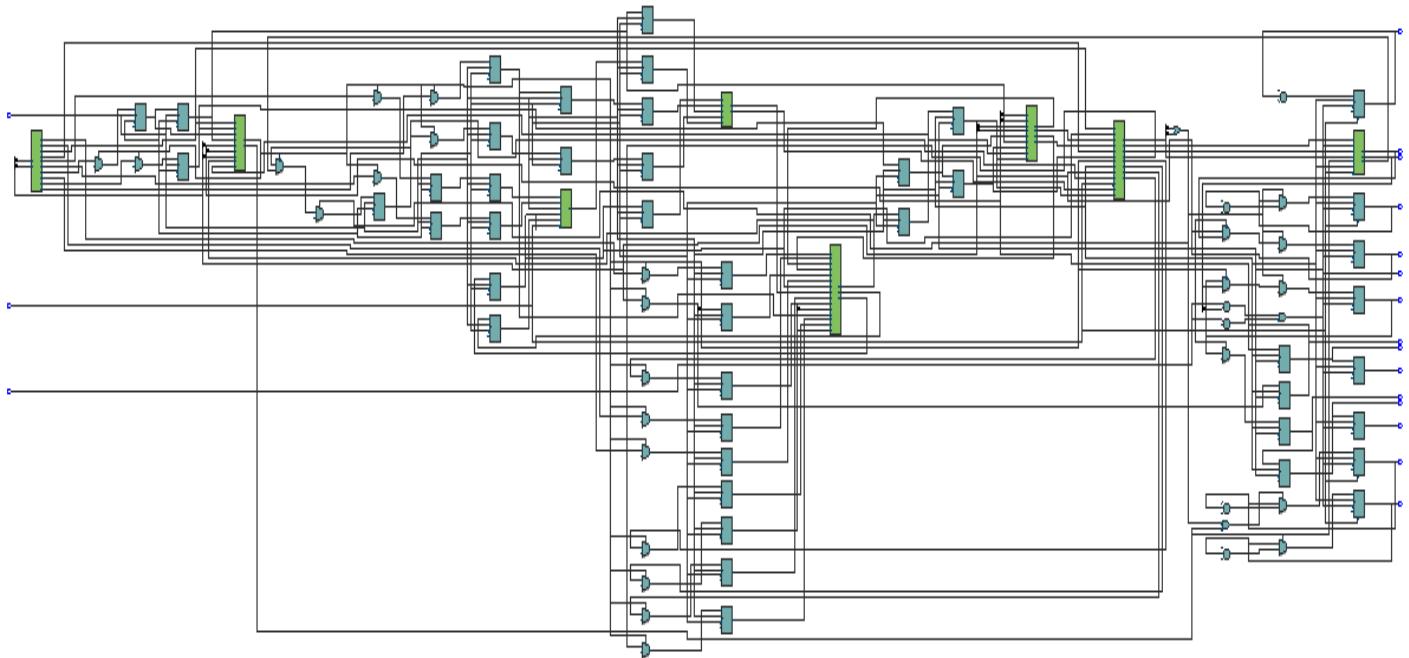


Figure 8: Pipelined MIPS architecture top entity

המעבד :

כעת, נסביר על המערכת שמיימשנו ועל המודולים השונים בה.

הRTL של המערכת שלנו :



BPADDER[0: 7], RST,CLK : INPUT •

. נכנס אליו כתובת BREAK POINT בהתאם לאיינפוט שבמוגדים 0-7.

CLKCNT, INSTCNT, IFPC, IFINSTRUCTION, IDPC, : OUTPUT •
IDINSTRUCTION, EXPC, EXINSTRUCTION, MEMPC, MEMINSTRUCTION,
.WBPC, WBINSTRUCTION, STRIGGER, FHCNT, STCNT

.FLUSH - מונה הסופר כמה פעמים בוצעה פקודת FHCNT

.CLKCNT - מונה הסופר כמה מהזורי שעון עברו.

.STALL - מונה הסופר כמה פעמים בוצעה פקודת STCNT

הסיבה בגיןה יש לכל שלב במעבד PIPELINE 5 שלבים גם את מוצא ה PC וגם את מוצא ה INSTRUCTION הינה כדי שנוכל לראות באופן יפה את מעבר הפקודות בין PIPELINE בין הרגיסטרים החוצצים את המעבד.

מיימוש ראשוןיו אותו התבקשנו לבצע במעבדה :

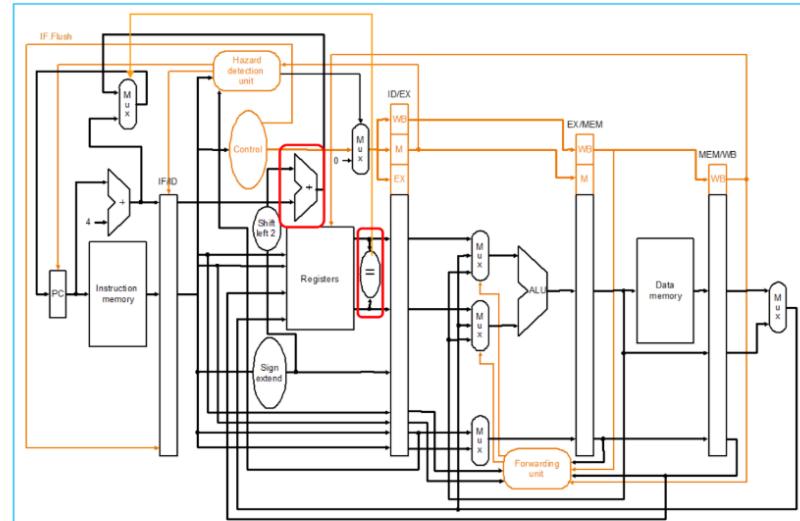


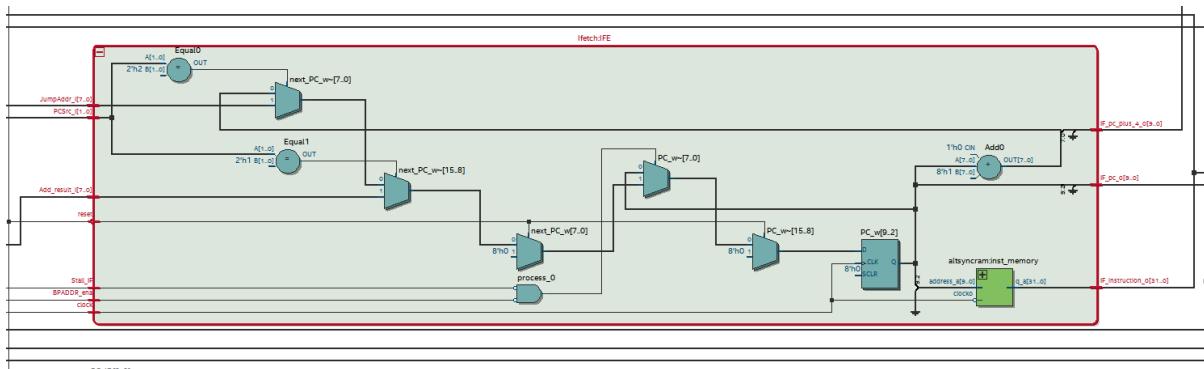
Figure 6: Five-stage pipelined MIPS architecture with forwarding and single delay slot support

השלבים במעבד :

שלב ה FETCH :

בשלב זה נרצה להביא פקודה בכתובת ה PC מהזיכרון ITCM. PC מटעדכן ל 3 מצבים הנבחרים על ידי JUMP, PCSEL ו BRANCH מפקודת PC ,4+PC . המצביעים הם : PCSEL, PC ו BRANCH .

شرطוט RTL :



מודול DECODE

בשלב זה נרצה לפענה את הפקודה המגיעה מה FETCH. את הפקודה אנחנו נפרק לפי 3 תבניות הנלמדו . I TYPE, J TYPE,R TYPE והן :

תבניות הפקודות :

Type	-31-	format (bits)					-0-
R	opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)	
I	opcode (6)	rs (5)	rt (5)	immediate (16)			
J	opcode (6)	address (26)					

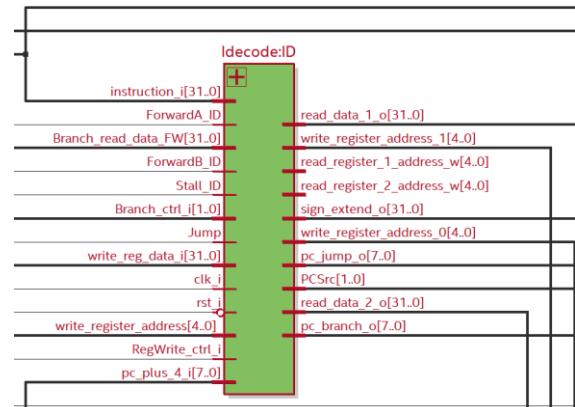
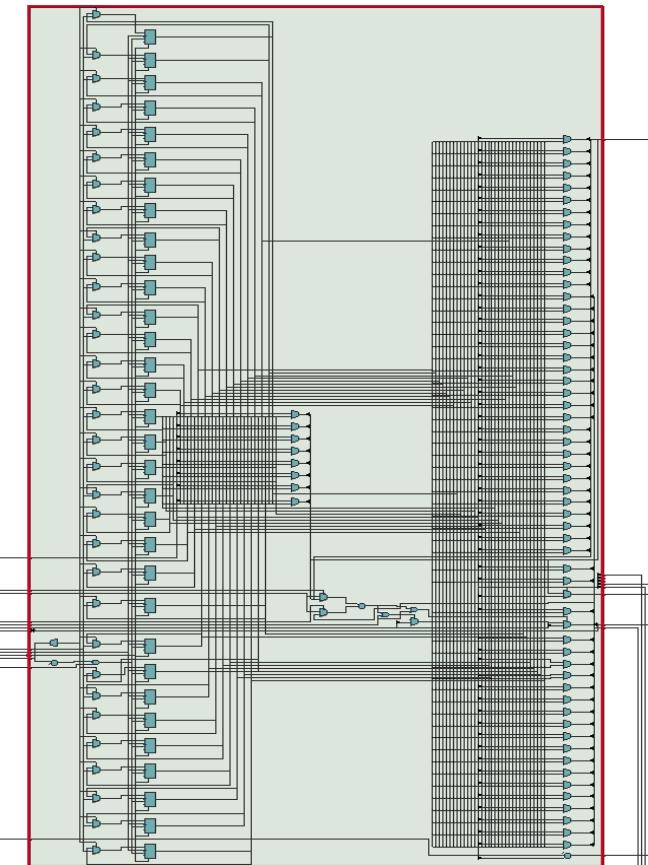
כך נשלוּוּ את הכתובת וריגיסטרים איהם נרצה לעבוד בשלבים המתקדמים במעבד. בנוסף בשלב זה נרצה לפענה את סוג הפקודה הרצiosa להבצע, וכך אותן הבקירה במודול הבא יעלו בהתאם.

כדי ליעיל את פעילות המעבד כבר בשלב זה נבצע את פעולות ההסתעפות : JUMP ו BRANCH, אם אחד מהם מתקיים. במידת הצורך קיבל נתונים מモודל ה FORWARDING .

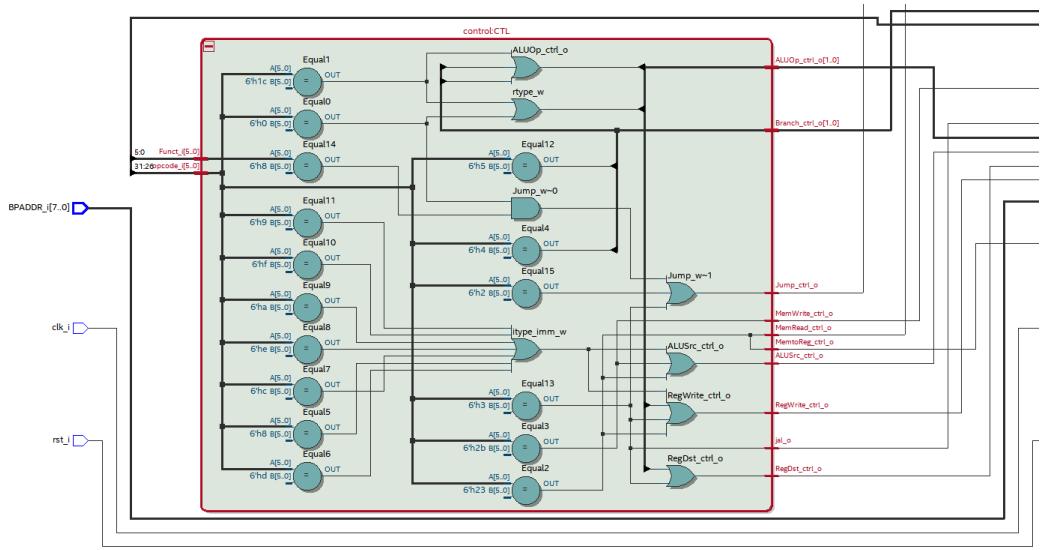
בנוסף בשלב זה יושבת יחידת CONTROL האחראית להעלות את כל קווי הבקירה הדורושים לפעולות התקינה של המעבד וביצוע הפקודות.

יחידה זאת מקבלת 2 שדות חשובים לפיענוח הפקודה והם שדה ה OPCODE ושדה ה FUNCT .

ה DECODE של RTL



ה : CONRTOL של יחידת RTL



מודול ה EXECUTE

בשלב זה במעבד הפקודות יוצאות לפועל, ככלומר בשלב זה נמצאים רכיבים חשובים במעבד כמו רכיב ה .SHIFTER ורכיב ה ALU.

בשלב זה מבצעים פקודות כמו חישוב כתובות או ביצוע פעולות אРИטמטיות.

כשעבדנו עם מעבד SINGLE CYCLE בוצעו בשלב זה גם חישוב ה PC הבא ופקודות ההסתעפות.

cut כשברנו למעבד PIPELINE עליינו להוסיף עוד 2 מודולים הכרחיים שהם HAZARD UNIT ו FORWARDING UNIT.

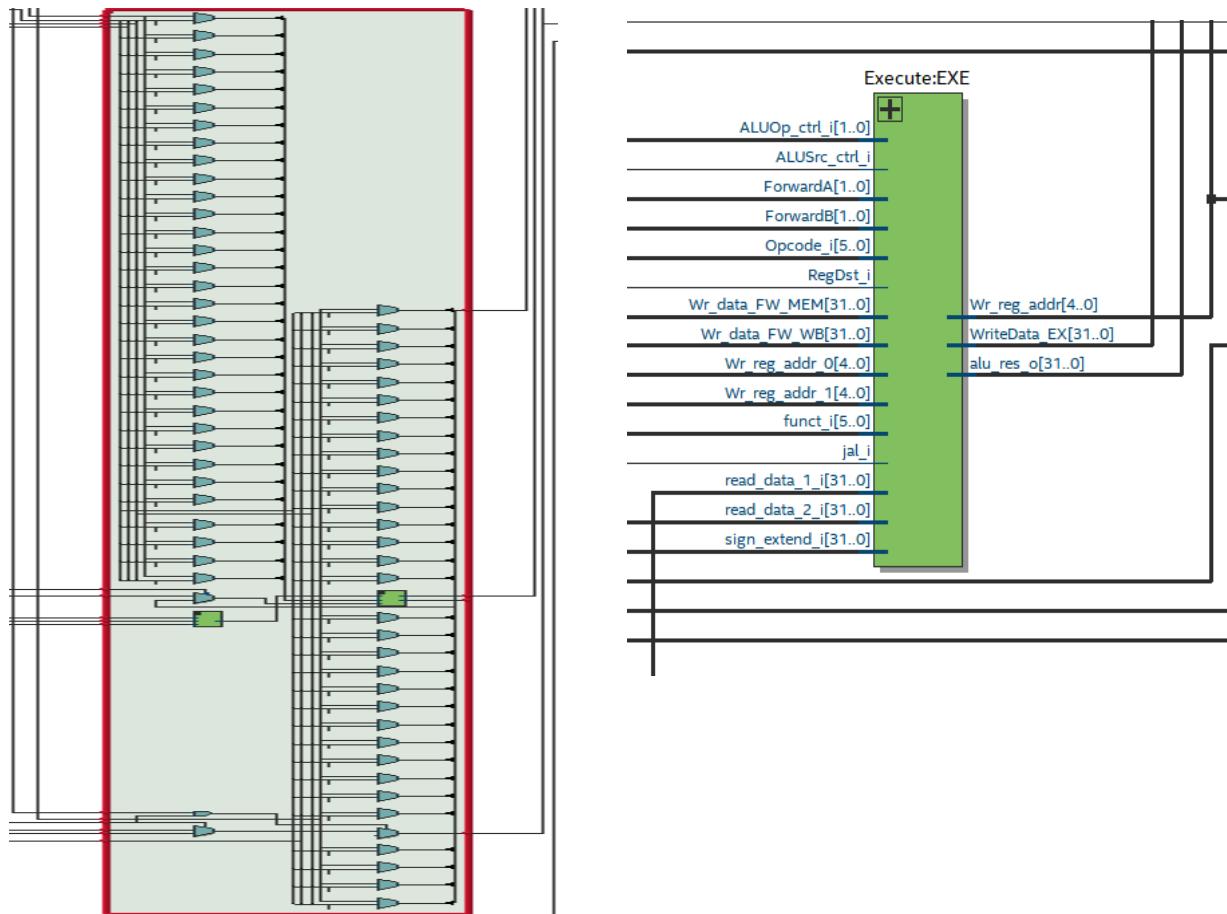
זאת מכיוון שיתכן שיחסבנו כתובות או נתונים במעבד לפקודה אחת אך לפקודה אחרת נדרש לשימוש נתונים אלה שיתכן שעוד לא הספיקו להישמר בכתובות או ברגיסטרים המתאימים. לשם כך, יש 2 אוטות חשובים שהם FLUSH ו STALL .

לדוגמא בפקודת BRUNCH הנמצאת בשלב DECODE ועתידה להתבצע יש בשלב ה FETCH פקודה חדשה. ברגע שההחליטה לבצע BRUNCH ההתבצעה יש לבצע FLUSH בשלב ה FETCH ולזרוק את הפקודה וה PC יתעדכן לערך הרצוי.

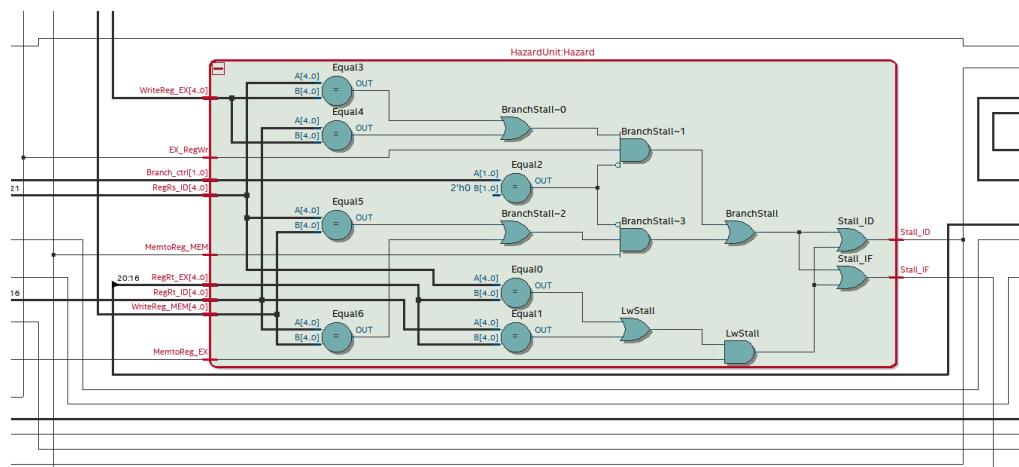
פקודת STALL היא פקודה המבצעת NOP, ככלומר היא פקודה הממתינה סייקל כדי שהميدע יוכל לעבור PIPELINE באופן תקין.

יחידת ה FORWARDING חשובה במעבד זה על מנת לפתור בעיות של DATA DEPENDENCIES .

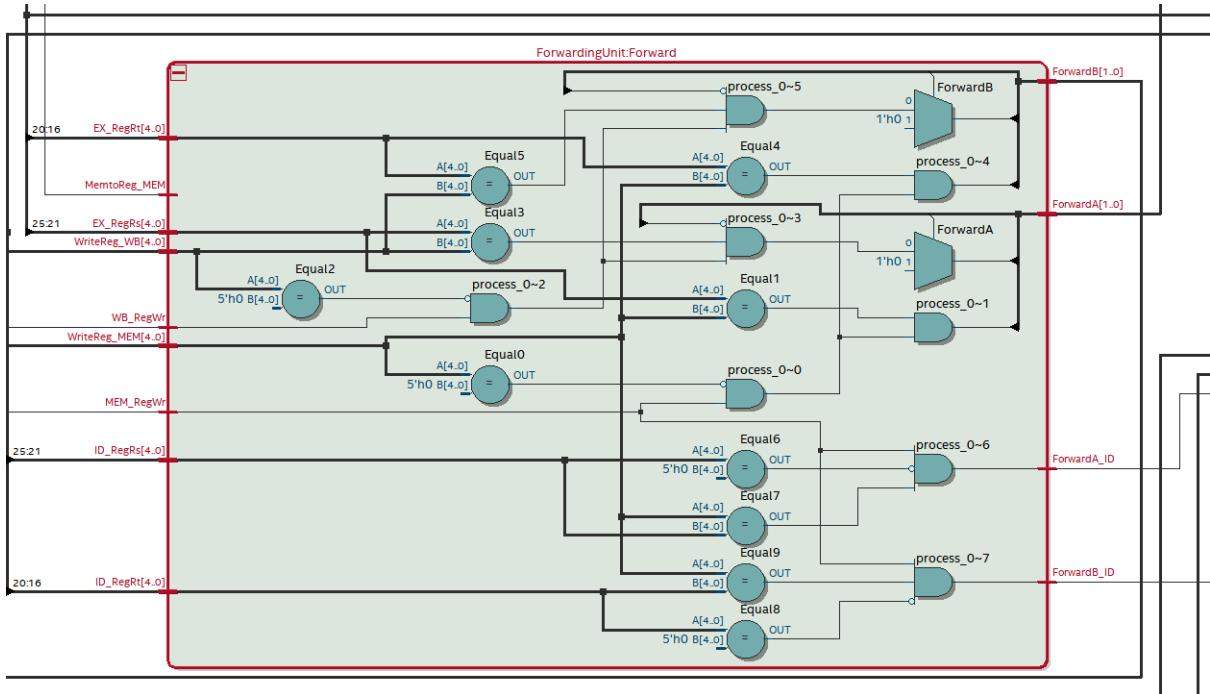
: EXECUTE של יחידת ה-RTL



: HAZARD של יחידת ה-RTL



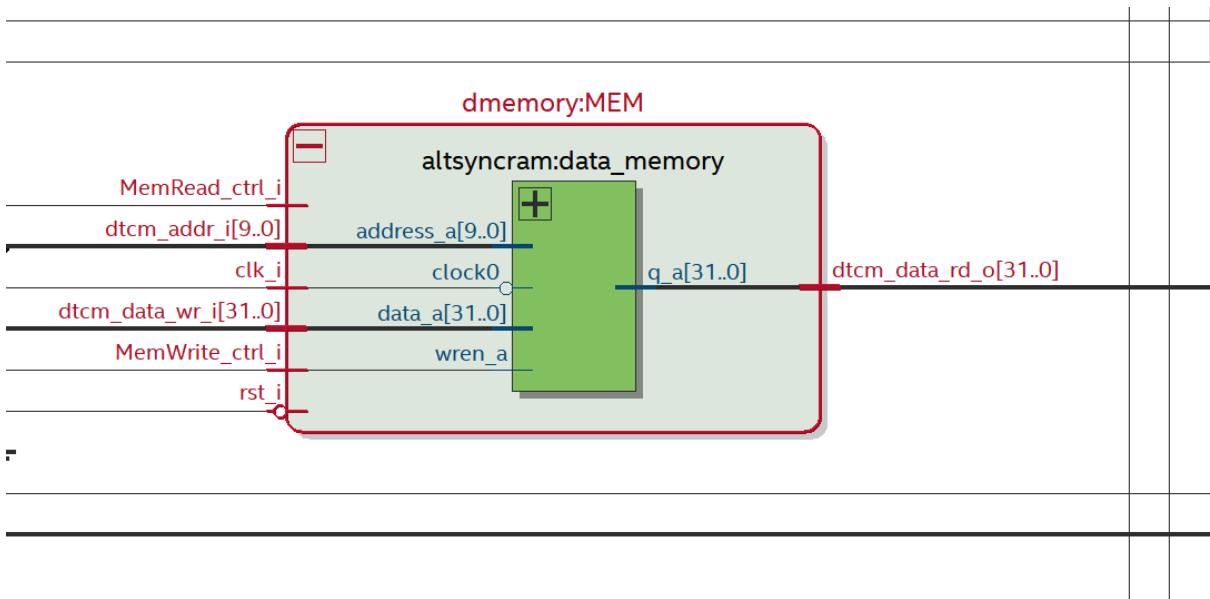
ה FORWARDING של יחידת RTL



מודול DATA MEMORY

מודול זה אחראי על ביצוע הכתיבה והקריאה מתוכן ה RAM שזהו בעצם זיכרון ה DATA MEMORY של המערכת שלנו.

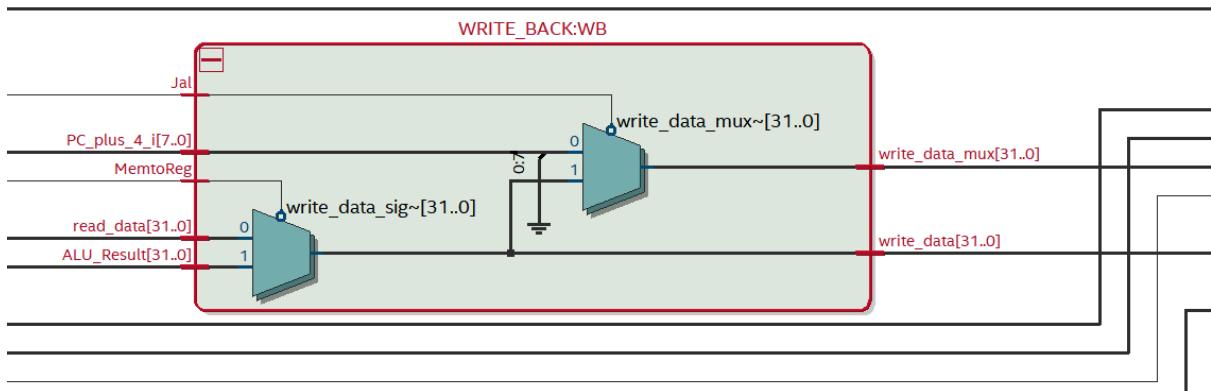
ה DATA MEMORY של RTL של יחידת RTL



מודול WRITE BACK

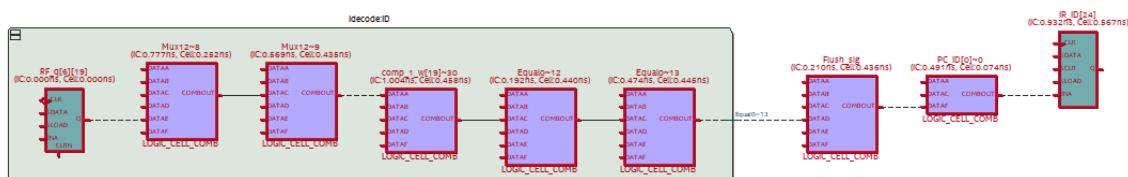
מודול זה אחראי על ניהול החזרת המידע וכטיבת המידע ל זיכרון בסוף שלבי PIPELINE. מודול זה מוציא את הכתובת ואת המידע כתוצאה מהפעולה שהמעבד ביצע ושולחת את התוצאה לרכיבים הרלוונטיים במעבד לצורך שמירת המידע.

ה WRITE BACK של יחידת RTL



המסלול הקרייטי:

מתוך תוכנת QUARTRUS נמצא המסלול הקרייטי במעבד שלנו. הנתיב הקרייטי אותו קיבלנו הינו .DATA MEMORY



מציאת תדר מקסימלי:

מתוך תוכנת QUARTRUS נמצא התדר המקסימלי של המעבד שלנו :

The screenshot shows the Quartus software interface with the following details:

- Table of Contents:** On the left, it lists various analysis and synthesis tools under the "Timing Analyzer" section.
- Slow 1100mV 85C Model Fmax Summary:** This panel displays Fmax values for different clocks. The table has columns: Fmax, Restricted Fmax, Clock Name, and Note. It shows three entries:

	Fmax	Restricted Fmax	Clock Name	Note
1	45.64 MHz	45.64 MHz	pll_inst1...TER divclk	
2	80.44 MHz	80.44 MHz	altera_reserved_tck	
3	137.21 MHz	137.21 MHz	clk_i	
- Slow 1100mV 0C Model Fmax Summary:** This panel displays Fmax values for a single clock. The table has columns: Fmax, Restricted Fmax, Clock Name, and Note. It shows one entry:

	Fmax	Restricted Fmax	Clock Name	Note
1	126.34 MHz	126.34 MHz	altera_reserved_tck	

SIGNLE TAP

בשלב זה ביצענו את הבדיקות על הקוד שכתבנו בעזרת תוכנת QUARTUS. לכל טסט הוצאנו קבצי ITCM ו DTCM מתאימים לביצוע התוכנית. קבצים אלה קבצים מותאמים לבקר שלנו אותם הוצאנו מתוך תוכנת MARS בקוד MIPS.

את הבקר קינפגנו כך ש RST מחובר לכפתור KEY0 והמתגים מחוברים לפי בית לתוכצתה BPADDER.

לאחר ביצוע קומpileציה וצריבת התוכנית הרצינו את הקוד וחיכינו לעצירתו כאשר נגיע לTRIGGER המתאים שהזנו.

להלן 6 טסטים שביצענו ופירוט על מה ניתן לראות בכל טסט.

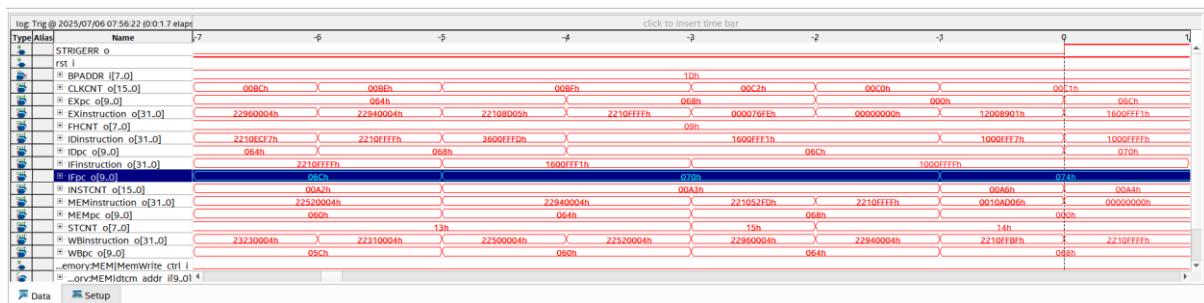
קבצי ה STP המצורפים ממוספרים לפי מספרי הטסטים המוצגים בקטע זה.

הסבר טסט 1 : נבחן כי בטסט זה הגדרנו את ה BPADDER להיות תוצאה ה \oplus בסיום הרצת התוכנית כמפורט. אכן, ניתן לראות כי הטסט ערך בהגעה לערך PC המתאים כלומר התוכנית ערכה בזמן המתאים.

עוד ניתן לראות את מעבר הפקודות PIPELINE בmozaic PC בכל מעבר וגייסטר.

ניתן לראות בסיום הרצת התוכנית את תוצאה הזיכרון שכוללות את המרכיבים המקוריים ולאחר מכן את תוצאות הפעולות שנעשו במעבד שהן פעולות חיבור, כפל וחישור ואת האחסון שלהם במקומות היעודים בזיכרון.

טסט 1



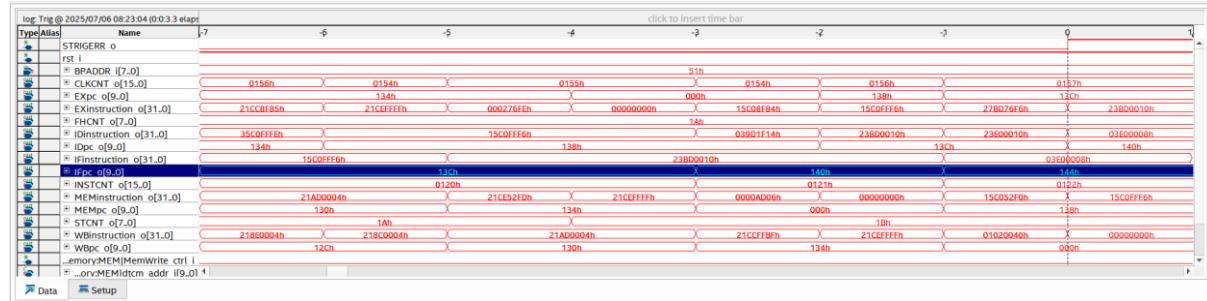
הסביר טסט 2 ו- 3: נבחין כי בטסטים הבאים הגדכנו את ה BPADDER להיות תוצאה ה Σ בסיום הרצת התוכנית כמבודק. אכן, ניתן לראות כי הטסט עזר בהגעה לערך PC המתאים כלומר התוכנית עצמה בזמן המתאים.

עוד ניתן לראות את מעבר הפקודות בPIPELINE בmooc'a PC בכל מעבר רגייסטר.

ניתן לראות בסיום הרצת התוכנית את תוכנת הזיכרון שכוללת את המרכיבים העיקריים ולאחר מכן את תוכנות הפעולות שנעשו במעבד שהן פועלות חיבור, כפל וחיסור הקורות מעבר על הקודך דרך לולאות וביצוע קפיצות (פעולות JUMP) ואת האיחסו שליהם במקומות הייעודיים בזיכרון.

בחוון כי קיבלו תוצאות זהות לטפסטים 3-1

הබב בינו טסיטים 3,2 הוא באופן בו אנו ניגשים לפעולות מבחןית הקודן, ולבו נצפה לתוכאה זהה גם כאן.



טסן 3



הסביר טסט 4 : כתוב התבקשנו לכתוב טסט מקובץ C הנitin לנו בדו"ח מכון. אזי מה שביצענו הוא לכתוב את הקוד כקוד MIPS ולאחר הרצתו ב MARS הוצאו ITCM ו DTCM מתאימים להרצת התוכנית החדשה.

הקוד מבצע את הפעולות הבאות:

```
#define M 4

void addMats(int Mat1[M][M], int Mat2[M][M], int resMat[M][M]){
    define it yourself ...
}

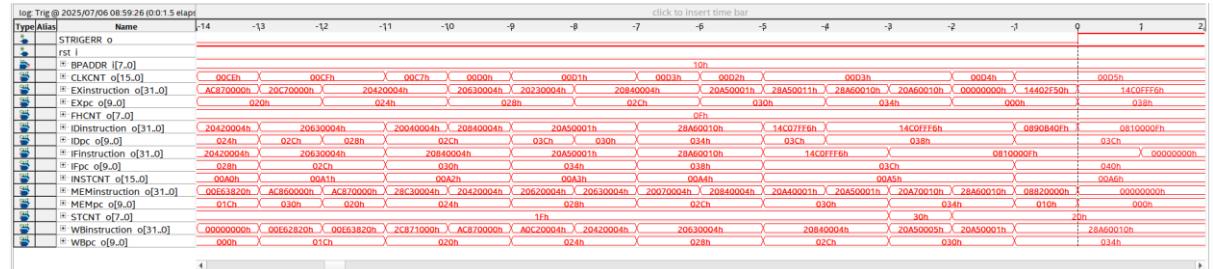
void main(){ //int=32bit
    int Mat1[M][M]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
    int Mat2[M][M]={{13,14,15,16},{9,10,11,12},{5,6,7,8},{1,2,3,4}};
    int resMat[M][M];

    addMats(Mat1,Mat2,resMat); // resMat = Mat1 + Mat2
}
```

בזיכרונו ניתן לראות את טיענת המרכיבים המקוריים ואת חישובי הפעולות שבוצעו במעבד ואוחסנו במקום הייחודי.

גם כאן הגדרנו את ה **BPADDER** להיות הכתובת האחורה של ה PC ובעלית הטריגר נעצרה התוכנית בכתובת הרצוייה.

4 טסע



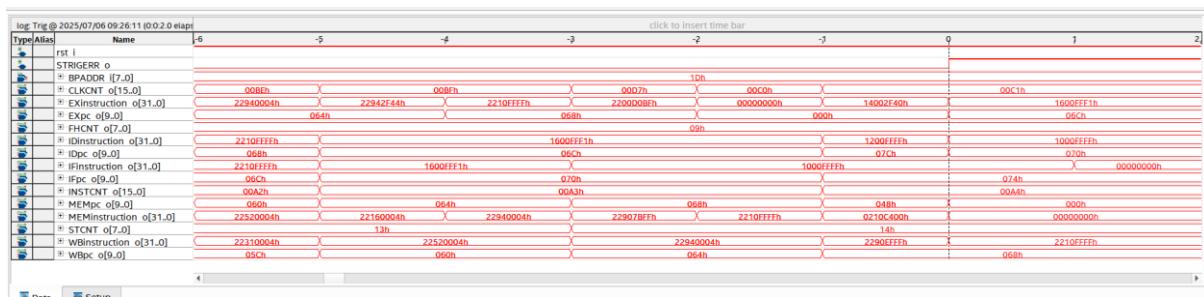
$$IPC = (166 - 15) / 213 = 0.708$$

הסבר טסט 5 : בבדיקה זו, הגדנו את רכיב ה- BPADDER כך שישקף את ערך ה- PC בסיום ביצוע התוכנית, בהתאם לדרישות. אכן, ניתן לראות כי ה- PC הגיע לערך המצוופה, מה ש מעיד שהתוכנית נעצרה בזמן הנכון.

כמו כן, במהלך הריצעה ניתן לעקוב אחר מעבר ההוראות לאורק שלבי ה PIPELINE דרך ערכיו ה-PC המופיעים במהלך ריגistration.

בסיום הריצה נצפו בזיכרון הנתונים המקוריים כלומר המרכיבים, ולאחריהם התוצאות של הפעולות הלוגיות שבוצעו במעבד, שהן AND, OR, XOR אשר אוחסנו במיקומים הייעודיים בזיכרון, בהתאם לתוכנית.

5 טסע



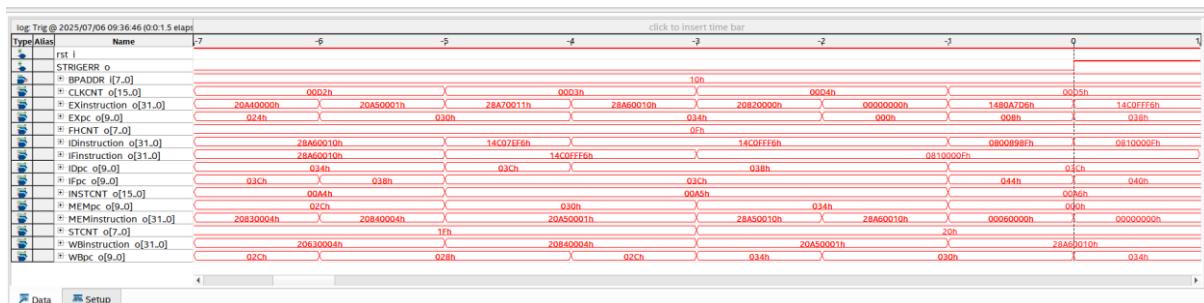
$$\text{JPC} = (164-9)/193 = 0.803$$

הסביר טסט 6 : בקטע קוד זה ביצעוו את פעולה ה SLT על שני מערכיים בגודל 16 ושמירת התוצאות בהתאם. בקטע זה השתמשנו ב 2 מערכיים בגודל 16 איחסנו אותם בזיכרון במקום המתאים ולאחר מכן

שמרנו את התוצאות במקומות היעודיים. גם כאן בדומה לשאר הטעדים הגדרנו את ה-BPADDR להיות סוף הרצת התוכנית מה שמעלה לנו את הטריגר במקום הנכון לפני ה-IFETCH של שלב ה-FETCH.

טסט 6

Instance 0: DTCM															
000000	00	00	00	01	00	00	00	00	00	00	00	00	00	00	03
000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000014	00	00	00	06	00	00	00	00	00	00	00	00	00	00	00
00001e	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000028	00	00	00	0B	00	00	00	00	00	00	00	00	00	00	00
000032	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00003c	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
000046	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000050	00	00	00	09	00	00	00	00	00	00	00	00	00	00	00
00005a	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000064	00	00	00	06	00	00	00	00	00	00	00	00	00	00	00
00006e	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000078	00	00	00	03	00	00	00	00	00	00	00	00	00	00	00
000082	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00008c	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000096	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000aa0	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00
0000aa	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000b4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000be	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000c8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000d2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000dc	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000e6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000fa	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000104	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00010e	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000118	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000122	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00012c	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



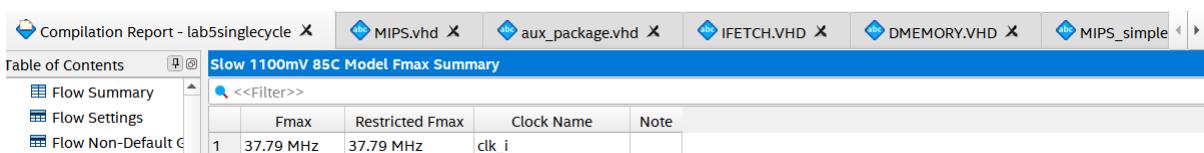
$$IPC = (166-15)/213=0.708$$

נצפה לקבל IPC זהה לשולט טסט 4 מכיוון שמבנה הטעט זהה - הפקודות הלוגיות שונות.

Single Cycle:

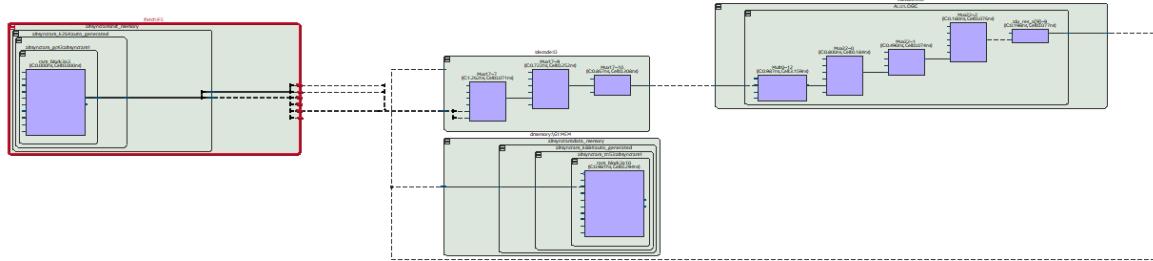
מציאת תדר מקסימלי :

מתוך תוכנת QUARTRUS נמצא את התדר המקסימלי של המעבד שלנו :

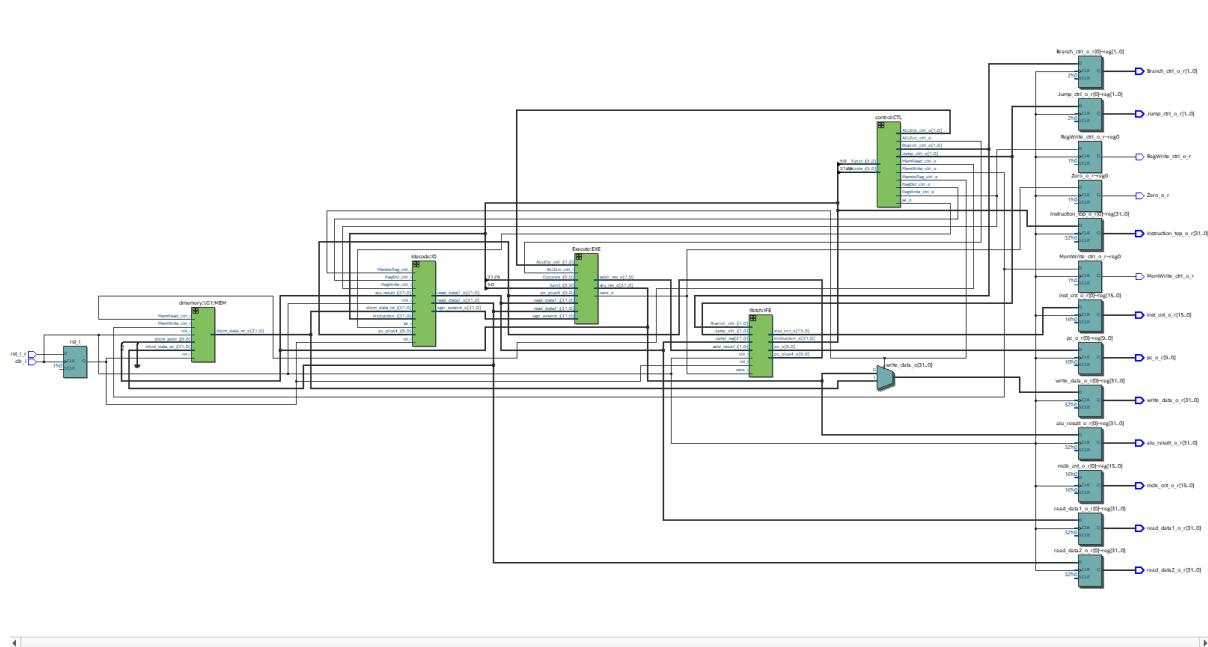


המסלול הクリיטי :

מתוך תוכנת ה QUARTRUS נמצא את המסלול הكريיטי במעבד שלנו. הנתיב הكريיטי אותו קיבלנו הינו .DATA MEMORY



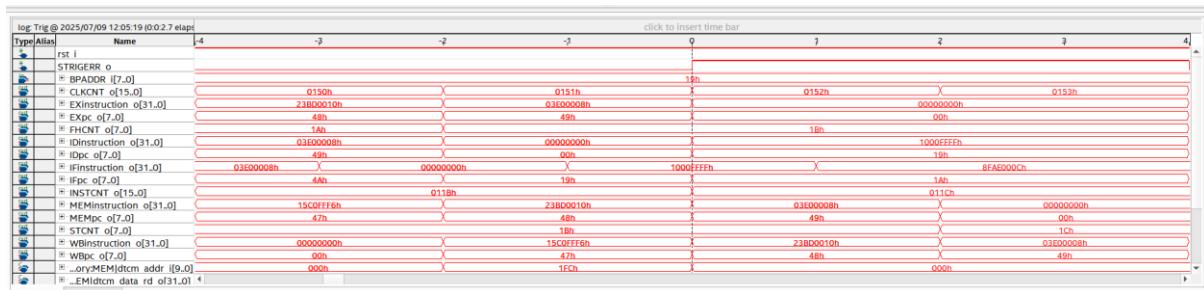
:RTL



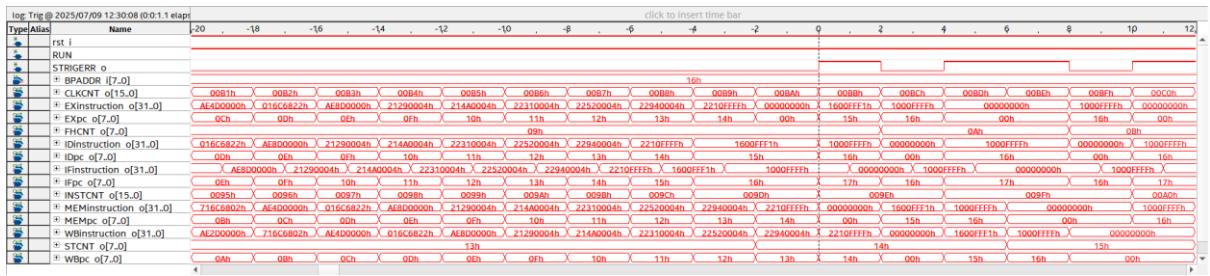
תיקון לאחר מעבר לפורמט הנכון:

2 טט

8STP



Instance 0: DTCM	
000000	00 00 00 0A 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 09
00000a	00 00 00 0A 00 00 00 64 00 00 00 63 00 00 00 62 00 00 00 61 00 00 00 60 00 00 00 5F 00 00 00 5E 00 00 00 5D 00 00 00 5C
000014	00 00 00 5B 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65 00 00 00 65
00001e	00 00 00 65 00 00 00 64 00 00 00 C6 00 00 01 26 00 00 01 84 00 00 01 E0 00 00 02 3A 00 00 02 92 00 00 02 E8 00 00 03 3C
000028	00 00 03 8E FF FF FF 9D FF FF FF 9F FF FF FF A1 FF FF FF A3 FF FF FF A5 FF FF FF A7 FF FF FF A9 FF FF FF AB FF FF FF AD
000032	FF FF FF AF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00003c	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000046	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00005a	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000064	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00006e	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000078	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000082	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00008c	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000096	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000aa	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000b4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000be	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000c8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000d2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000dc	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000e6	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000fa	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



Instance 0: DTCM

000000	00 00 00 01	00 00 00 02	00 00 00 03	00 00 00 04	00 00 00 05	00 00 00 06	00 00 00 07	00 00 00 08	00 00 00 64	00 00 00 63
00000a	00 00 00 62	00 00 00 61	00 00 00 60	00 00 00 5F	00 00 00 5E	00 00 00 5D	00 00 00 65	00 00 00 65	00 00 00 65	00 00 00 65
000014	00 00 00 65	00 00 00 65	00 00 00 65	00 00 00 65	FF FF FF 9D	FF FF FF 9F	FF FF FF A1	FF FF FF A3	FF FF FF A5	FF FF FF A7
00001e	FF FF FF A9	FF FF FF AB	00 00 00 64	00 00 00 C6	00 00 01 26	00 00 01 84	00 00 01 E0	00 00 02 3A	00 00 02 92	00 00 02 E8
000028	00 00 00 08	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000032	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00003c	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000046	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000050	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00005a	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000064	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00006e	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000078	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 20	00 00 00 80	00 00 00 08	00 00 00 00	00 00 00 00
000082	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00008c	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000096	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000a0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000aa	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000b4	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000be	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000c8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000d2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000dc	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000e6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000f0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0000fa	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

