



מסמך מלווה – מעבדה 5 מעבדת מסלול מחשבים

מטרת מסמך זה הוא הסבר על מודולי המערכת.

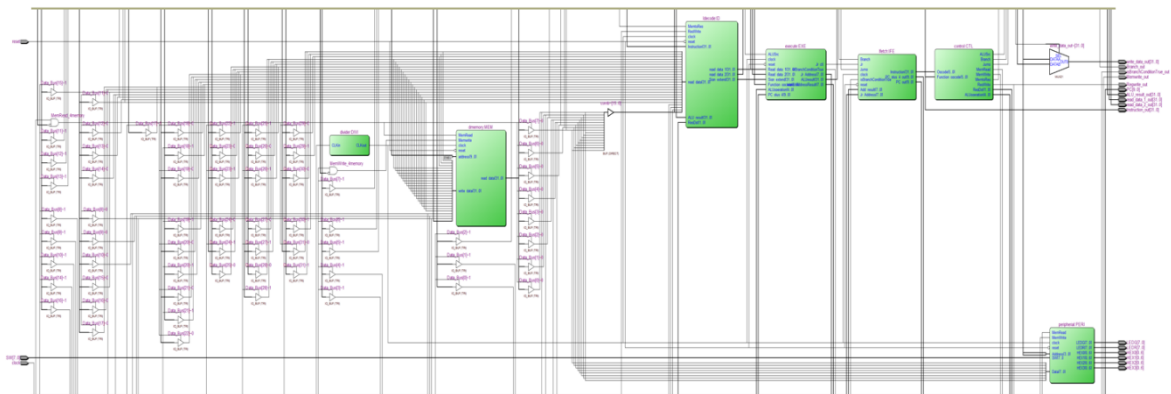
מגשים: רון טל וכפיר כהן.

מודול MIPS:

מעבד single cycle, מורכב מארבעה תתי מודולים מרכזיים:

Fetch, decode, control, execute, dmemory

תומך ב27 פקודות אסמבלי.



המודול נבדק בעזרת קוד אסמבלי במבצע transpose למטריצה בגודל $M \times M$

כאשר $M=4$.



פקודות שנתמכות:

Required MIPS Assembly Instructions Set

Arithmetic Instructions			
Instruction	Example	Meaning	Comments
add	add \$1,\$2,\$3	$\$1 = \$2 + \$3$	
subtract	sub \$1,\$2,\$3	$\$1 = \$2 - \$3$	
add immediate	addi \$1,\$2,100	$\$1 = \$2 + 100$	
Multiply (without overflow)	mul \$1,\$2,\$3	$\$1 = \$2 * \$3$	Result is only 32 bits!

Logical Instructions			
Instruction	Example	Meaning	Comments
and	and \$1,\$2,\$3	$\$1 = \$2 \& \$3$	Bitwise AND
or	or \$1,\$2,\$3	$\$1 = \$2 \$3$	Bitwise OR
xor	xor \$1,\$2,\$3	$\$1 = \$2 \wedge \$3$	Bitwise XOR
and immediate	andi \$1,\$2,100	$\$1 = \$2 \& 100$	Bitwise AND with immediate value
or immediate	ori \$1,\$2,100	$\$1 = \$2 100$	Bitwise OR with immediate value
xor immediate	xori \$1,\$2,100	$\$1 = \$2 \wedge 100$	Bitwise XOR with immediate value
shift left logical	sll \$1,\$2,10	$\$1 = \$2 \ll 10$	Shift left by constant number of bits
shift right logical	srl \$1,\$2,10	$\$1 = \$2 \gg 10$	Shift right by constant number of bits

Data Transfer Instructions			
Instruction	Example	Meaning	Comments
move	move \$1,\$2	$\$1 = \2	Pseudo-instruction (provided by MARS Assembler, not processor!) Copy from register to register.
load word	Lw \$1,100(\$2)	$\$1 = \text{Memory}[\$2 + 100]$	
store word	Sw \$1,100(\$2)	$\text{Memory}[\$2 + 100] = \1	
load upper immediate	lui \$1,100	$\$1 = 100 \times 2^{16}$	Load constant into upper 16 bits. Lower 16 bits are set to zero.

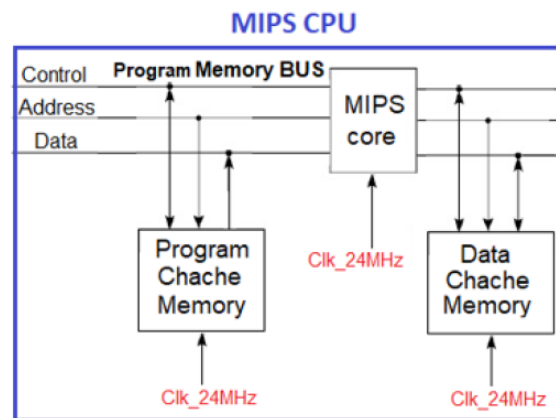
Conditional Branch Instructions			
Instruction	Example	Meaning	Comments
branch on equal	Beq \$1,\$2,100	if($\$1 == \2) go to PC+4+100	Test if registers are equal
branch on not equal	Bne \$1,\$2,100	if($\$1 \neq \2) go to PC+4+100	Test if registers are not equal

Comparison Instructions			
Instruction	Example	Meaning	Comments
set on less than	slt \$1,\$2,\$3	if($\$2 < \3) $\$1 = 1$; else $\$1 = 0$	Test if less than. If true, set \$1 to 1. Otherwise, set \$1 to 0.
set on less than immediate	slti \$1,\$2,100	if($\$2 < 100$) $\$1 = 1$; else $\$1 = 0$	Test if less than. If true, set \$1 to 1. Otherwise, set \$1 to 0.

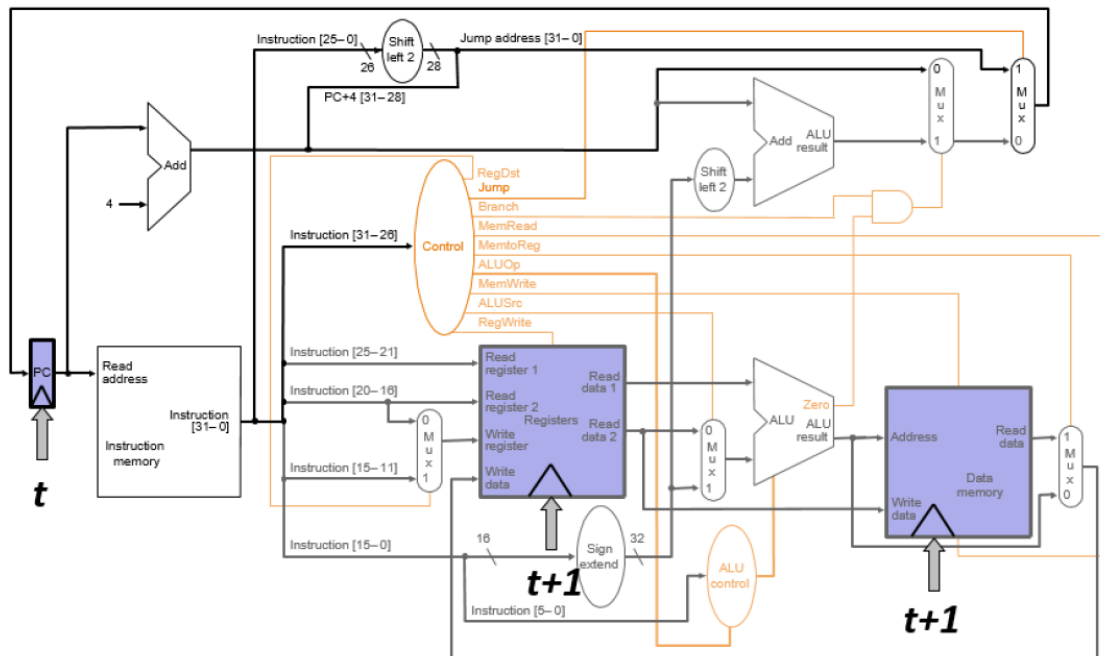
Unconditional Jump Instructions			
Instruction	Example	Meaning	Comments
jump	j 1000	go to address 1000	Jump to target address
jump register	jr \$ra	go to return address stored in \$ra	procedure return
jump and link	jal 1000	\$ra=PC+4; go to procedure call which starts in address 1000	Use when making procedure call. This saves the return address in \$ra



תצורת אגירת הזכרון הינה Harvard, כלומר שזכרון המידע מופרד מזכרון התוכנית:



איור מופשט של המודול:





תתי מודולים:

Execute:

מכיל ALU אשר מבצע פעולות על שני האופרנדים בכניסות Ainput, Binput בהתאם לALUoperation שמגיע ממודול control.



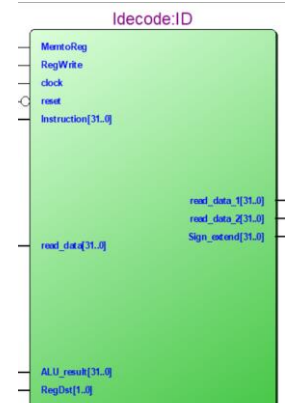
:Control

מוציא את קווי הבקרה ההכרחיים לפעולת המערכת בהתאם לפקודה שמגיעה ממודול FETCH.



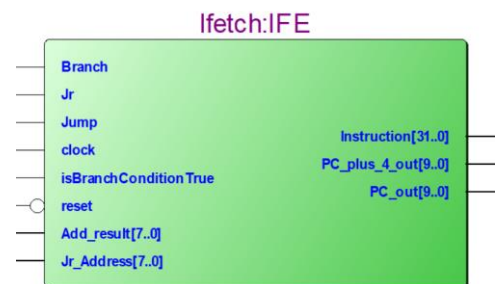
:DECODE

מודול זה מפענח את הפקודה שהתקבלה ממודול FETCH. מודול זה מכיל את ה-Register File ולאחר פענוח הפקודה מוציא ומכניס את הדאטה מהרגיסטרים המתאימים.



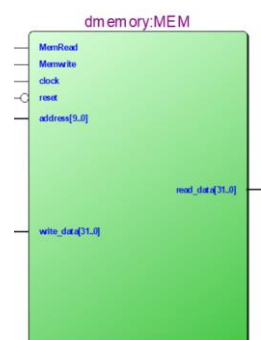
:FETCH

מכיל את ה-Register PC, אחראי להביא את הפקודה הבאה בתוכנית מה-Program Memory שנמצא בו בתצורת ram (קריאה בלבד).



:Dmemory

מכיל את ה-data memory בתצורת ram (קריאה וכתובה)



מודול peripheral קיים אך אינו בשימוש ניתן לגשת דרכו לרכיבי IO בכרטיס (LEDs, hex screen..), מודול divider מחלק את השעון או במספר גדול יותר אם נרצה (ניתן לעקוף ע"י קצר בין רגל הכניסה והמוצא)



תוצאת קוד Assembly

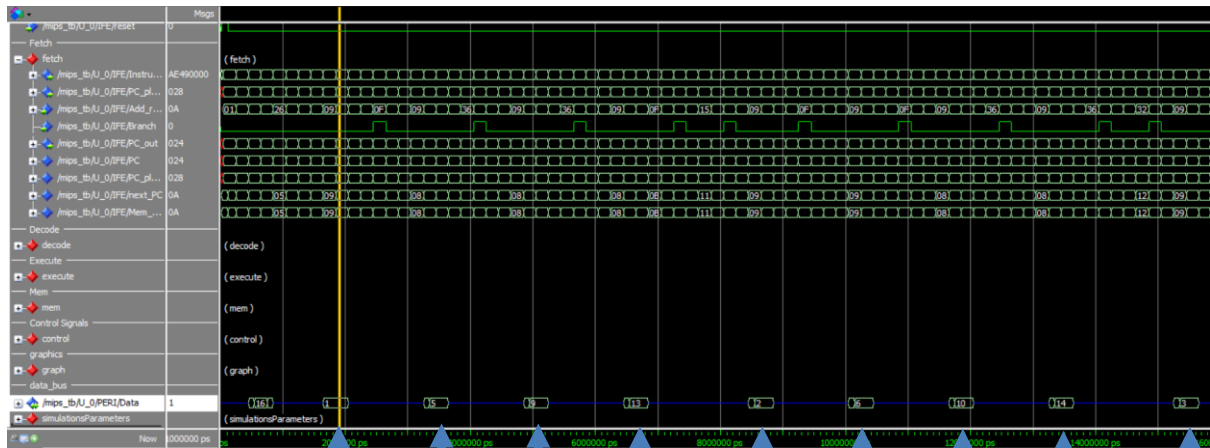
הזכרון לפני הרצת הקוד:

```
0000002d 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000027 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000021 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
0000001b 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000015 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
0000000f 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000009 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000003 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
```

הזכרון אחרי הרצת הקוד:

```
00000027 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000021 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
0000001b 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000015 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
0000000f 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000009 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
00000003 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
```

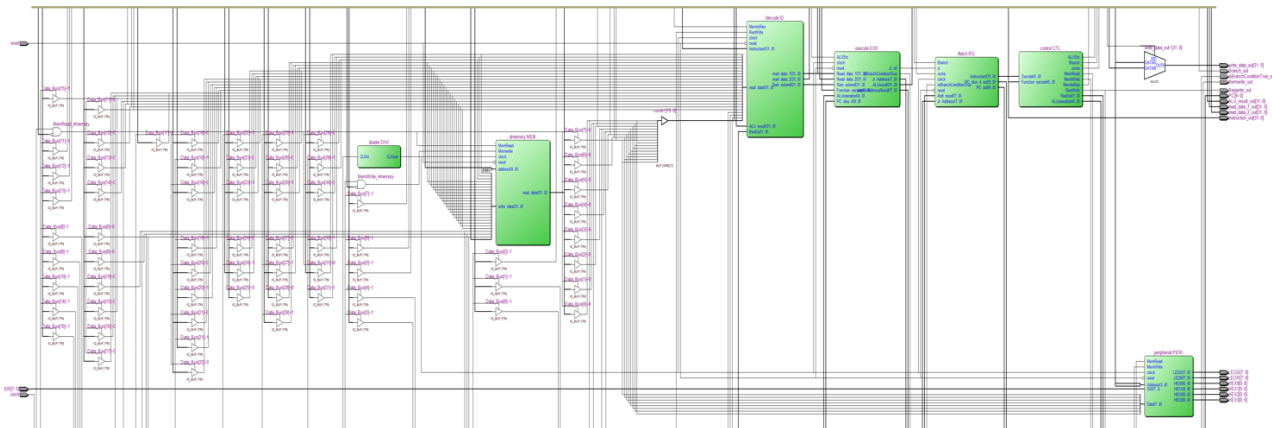
Assembly Code in ModelSim:



החיצים מצביעים על הערך שעובר בדatabus של ערכי המטריצה לאחר ביצוע פעולת transpose



RTL Viewer:



Usage:

Flow Summary	
Flow Status	Successful - Wed Jun 29 12:38:11 2022
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	lab5_2022
Top-level Entity Name	MIPS
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	2,918 / 18,752 (16 %)
Total combinational functions	2,428 / 18,752 (13 %)
Dedicated logic registers	1,049 / 18,752 (6 %)
Total registers	1049
Total pins	228 / 315 (72 %)
Total virtual pins	0
Total memory bits	65,536 / 239,616 (27 %)
Embedded Multiplier 9-bit elements	6 / 52 (12 %)
Total PLLs	0 / 4 (0 %)

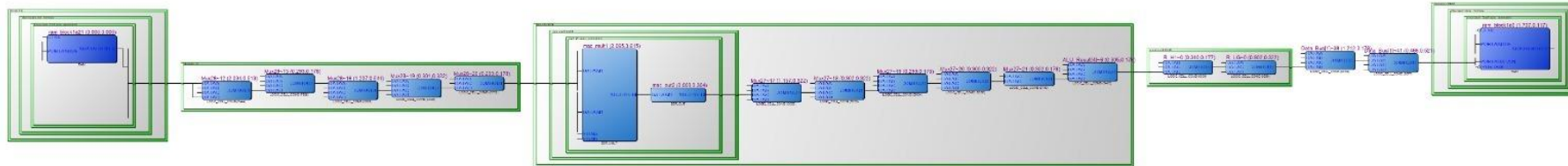
תדר שעון מקסימלי אפשרי:

	Fmax	Restricted Fmax	Clock Name	Note
1	24.64 MHz	24.64 MHz	clock	



Critical path:

המסלול הארוך ביותר הינו של הסיגנלים שעוברים דרך המכפל, מסלול זה הוא המאלץ את התדר המקסימלי.



Signal Tap:

הבדיקה בסיגנל tap אינה של הקוד אותו כתבנו של היפוך מטריצה עקב מחסור בזמן.

הקוד המופעל כאן הינו של מיון מערך המכיל את התעודת זהות של רון:

תז ראשוני: 20851603

תז לאחר מיון: 0 0 1 2 3 5 6 8

כפי שניתן לראות ב databus

