

תכן חומרה בשפת VHDL

משימה מסכמת:

מימוש של מעבד mips 32 bit pipeline

מבוא: יש לממש את כל היחידות של הצנרת כ-components ולחברן יחד ביחידת ה-top. להלן רשימת היחידות למימוש:

1. יחידת ה-fetch

2. Decode

3. Execute

4. Memory access

5. Write back

6. Forwarding unit

7. Hazard detection

8. יחידת הבקרה הראשית ("הביצה הגדולה")

הנחיות למימוש:

- בקרת ה-ALU ("הביצה הקטנה") יש לממש בחלק מיחידת ה-Execute
- זיכרון הפקודות (instruction memory) המהווה חלק מיחידות ה-fetch
- זיכרון הנתונים (data memory) מהווה חלק ויחידת ה-memory access
- **את הזיכרונות תקבלו כיחידות הסגורות (black boxes) ללא צורך במימוש.**

מגבלות ההקלות:

1. יש לכתוב קוד אשר עובר סינתיזה בהצלחה!!! אין לכתוב מודל התנהגותי!!!
2. עבודה ברמת הפורטים עם טיפוס std_logic/vector בלבד! סיגנלים פנימיים מותר מכל טיפוס על פי החלטתכם.
3. במימוש של יחידת הביצוע (execute) לצורך כתיבת ה-ALU יש להשתמש בפקודות המובנות בשפה כגון "+", "-", וכו' ללא צורך בתיאור פרטני של מבנים האריתמטיים.
4. ה-TB אמור לתת תמונה כללים של עבודת המעבד ולבצע שגרה קצרה עם משמעות כלשהי ותוצאה ידוע מראש! השגרה אמורה לכלול כ-8-10 פקודות שונות. לצורך כך יש להכין באופן ידני קבועים ולהגדירם בטסט. הקבועים האלה מייצגים בבסיס בינארי את הפקודות אשר נבחרו לצורך הטסט על פי הפורמט המוגדר במעבד Mips
5. סט הפקודות הנתמך על ידי המעבד כפי שמוגדר במעבד הצנרת (ראה נספח)
6. חומר עזר הכולל מצגות הרלוונטיות וקוד של הזיכרונות באתר הקורס

נספחים :

פקודת בפורמט R :

Opcode	Rs	Rt	Rd	Shift amount	func
6bit	5bit	5bit	5bit	5bit	6bit

0	rs	rt	rd	shift <i>n</i>	function
31-26	25-21	20-16	15-11	10-6	5-0

operation	syntax	The Action	# Function
add	add \$1,\$2,\$3	$\$1 = \$2 + \$3$	32=0x20
sub	sub \$1,\$2,\$3	$\$1 = \$2 - \$3$	34=0x22
and	and \$1,\$2,\$3	$\$1 = \$2 \& \$3$	36=0x24
or	or \$1,\$2,\$3	$\$1 = \$2 \$3$	37=0x25
nor	nor \$1,\$2,\$3	$\$1 = \sim \$2 \$3$	39=0x27
Slt set on less than	slt \$1,\$2,\$3	if ($\$s2 < \$s3$) $\$s1 = 1$ else $\$s1 = 0$	42=0x2a

פקודת בפורמט IMM :

Opcode	Rs	Rt	Address \ Immediate
6bit	5bit	5bit	16 bit

beq \$X, \$Y, const

4	rs	rt	address
31-26	25-21	20-16	15-0

lw \$rt,add(\$rs) # \$rt=MEM (add+\$rs)

35	rs	rt	address
31-26	25-21	20-16	15-0

sw \$rt,add(\$rs) # MEM (add+\$rs)=\$rt

43	rs	rt	address
31-26	25-21	20-16	15-0

פקודות בפורמט JUMP :

J label

2(opcode)	Address
26-31	0-25

בקרה הראשית

Inst.	Op-code	Reg-Dst	ALU-Src	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp
R-type	0	1	0	0	1	0	0	0	10
lw	35	0	1	1	1	1	0	0	00
sw	43	X	1	X	0	0	1	0	00
beq	4	X	0	X	0	0	0	1	01

ALU בקרת

R-type	Op	Rs	Rt	Rd	Shamt	Funct
I-type	Op	Rs	Rt	Address offset		

Instruction	Op	Funct	(ALUOp)	ALU ctrl	Function
lw	35	-	00	010	ADD
sw	43	-	00	010	ADD
beq	4	-	01	110	SUB
addu	0	33	10	010	ADD
sub	0	34	10	110	SUB
and	0	36	10	000	AND
or	0	37	10	001	OR
slt	0	42	10	111	SLT

ריכוז קווי הבקרה (לא כולל JUMP)

Instruction	Execution/Address Calculation stage control lines				Memory access stage control lines			Write-back stage control lines	
	Reg Dst	ALU Op1	ALU Op0	ALU Src	Branch	Mem Read	Mem Write	Reg write	Mem to Reg
R-format	1	1	0	0	0	0	0	1	0
lw	0	0	0	1	0	1	0	1	1
sw	X	0	0	1	0	0	1	0	X
beq	X	0	1	0	1	0	0	0	X