



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

آزمایشگاه معماری کامپیوتر

پروژه پایانی درس

در پروژه‌ی نهایی این درس هدف جمع‌بندی مطالب ارائه‌شده در طول ترم و طراحی یک پردازنده‌ی AUT-MIPS است. در این پروژه یک پردازنده‌ی ۱۶ بیتی با ویژگی‌هایی که در ادامه می‌آید باید طراحی گردد. این پردازنده نسبت به پردازنده‌ی MIPS تغییراتی کرده است. طول دستورات این پردازنده ۱۶ بیت بوده و به همین دلیل در این پردازنده PC به جای آنکه با ۴ جمع شود، با ۲ جمع می‌شود. همچنین طول ثابت PC ۱۶ بیت است. دستورات این پردازنده به سه دسته‌ی کلی I-Type، R-Type و J-Type تقسیم می‌شود که در ادامه ویژگی‌های هر کدام ذکر می‌شود.

۱- R-Type: این دستور همانند دستور R-Type در پردازنده‌ی MIPS است که دو ثابت مبداء و یک ثابت مقصد دریافت می‌کند. ساختار دستور و جدول opcode این دستور در ادامه آمده است:

15.....12	11.....9	8.....6	5.....3	2.....0
4-bits	3-bit	3-bit	3-bit	3-bit
opcode	source 1	source 2	destination	function

شکل ۱: ساختار دستور R-Type

opcode	function	operation
0000	000	add
0000	001	sub
0000	010	and
0000	011	or
0000	100	xor
0000	101	nor
0000	110	slt
0000	111	jr

جدول ۱: مقدار opcode برای دستورات R-Type

۲- I-Type: دستورات I-Type مانند دستورات I-Type در MIPS می‌باشد با این تفاوت که مقدار immediate ۶ بیتی می‌باشد. ماشین کد دستورات I-Type در شکل ۲ آورده شده است. چهار بیت اول (بیت‌های ۱۲ الی ۱۵) opcode را مشخص می‌کنند. سه بیت بعد (بیت‌های ۹ الی ۱۱) شماره ثابت را مشخص می‌کنند. سه بیت بعد (بیت‌های ۶ الی ۸) شماره ثابت دوم را مشخص می‌کنند. ۶ بیت آخر (بیت‌های صفر الی ۵) مقدار immediate را مشخص می‌کنند.

برای دستورات addi، ori و andi چهار بیت اول (بیت‌های ۱۲ الی ۱۵) opcode را مشخص می‌کنند. سه بیت بعد (بیت‌های ۹ الی ۱۱) شماره ثابت مبداء را مشخص می‌کنند. سه بیت بعد (بیت‌های ۶ الی ۸) شماره ثابت مقصد را مشخص می‌کنند. ۶ بیت آخر (بیت‌های صفر الی ۵) مقدار immediate را مشخص می‌کنند. این مقدار ۶ بیتی sign extend می‌شود به یک مقدار ۱۶ بیتی و بعد با محتوای ثابت عملیات انجام می‌شود. مقادیر opcode برای دستورات addi، andi و ori در جدول ۲ آمده است.

15.....12	11.....9	8.....6	5.....0
4-bits	3-bit	3-bit	6-bit
opcode	register1	register2	Immediate

شکل ۲- قالب دستورات I-Type

opcode	Operation
0001	addi
0010	andi
0011	ori

جدول ۲- مقادیر opcode برای دستورات addi, andi و ori

برای این پردازنده دستورات شیفت جزء قالب I-Type حساب می شوند. برای دستورات sll (shift left logical), srl (shift right logical) و sra (shift right arithmetic) چهار بیت اول (بیت‌های ۱۲ الی ۱۵) opcode را مشخص می کند. سه بیت بعد (بیت‌های ۹ الی ۱۱) شماره ثبات مبدأ را مشخص می کند. سه بیت بعد (بیت‌های ۶ الی ۸) شماره ثبات مقصد را مشخص می کند. ۶ بیت آخر (بیت‌های صفر الی ۵) مقدار شیفت را مشخص می کند. مقدار شیفت یک مقدار بدون علامت می باشد. مقادیر opcode برای دستورات sll, srl و sra، در جدول ۳ آمده است.

Opcode	Operation
0100	sll
0101	srl
0110	sra

جدول ۳- مقادیر opcode برای دستورات sll, srl و sra

برای این پردازنده دستورات load (lhw) و store (shw) جزء قالب I-Type حساب می شوند. دستورات lhw و shw به ترتیب یک داده ۱۶ بیتی را از حافظه می خواند و یک داده ۱۶ بیتی را در حافظه می نویسد. در این پردازنده مانند MIPS از آدرس دهی displacement (محتوای ثبات + offset) برای محاسبه آدرس استفاده می شود. برای دستورات lhw چهار بیت اول (بیت‌های ۱۲ الی ۱۵) opcode را مشخص می کند. سه بیت بعد (بیت‌های ۹ الی ۱۱) شماره ثبات پایه را مشخص می کند. سه بیت بعد (بیت‌های ۶ الی ۸) شماره ثبات مقصد را مشخص می کند. ۶ بیت آخر (بیت‌های صفر الی ۵) مقدار offset را مشخص می کند که باید sign extend شود و بعد با ثبات پایه جمع شود و آدرس را تولید کند. برای دستورات shw چهار بیت اول (بیت‌های ۱۲ الی ۱۵) opcode را مشخص می کند. سه بیت بعد (بیت‌های ۹ الی ۱۱) شماره ثبات پایه را مشخص می کند. سه بیت بعد (بیت‌های ۶ الی ۸) شماره ثبات مبدأ را مشخص می کند. ۶ بیت آخر (بیت‌های صفر الی ۵) مقدار offset را مشخص می کند که باید sign extend شود و بعد با ثبات پایه جمع شود و آدرس را تولید کند. مقدار offset یک مقدار علامت دار می باشد. مقادیر opcode برای دستورات lhw و shw، در جدول ۴ آمده است.

opcode	Operation
0111	lhw
1000	shw

جدول ۴- مقادیر opcode برای دستورات lhw و shw

برای این پردازنده دستورات پرش (branch) جزء قالب I-Type حساب می شوند. دستورات پرش محتوای دو ثبات را با یکدیگر مقایسه می کنند و بر اساس نوع مقایسه، اگر نتیجه درست باشد پرش انجام می شود و در غیر اینصورت انجام نمی شود. آدرس پرش به اینصورت محاسبه می شود که مقدار ۶ بیتی immediate به مقدار ۱۶ بیتی sign extend می شود سپس در دو ضرب می گردد و بعد با  $PC+4$  جمع می شود. نتیجه این جمع آدرس محل پرش را مشخص می کند:

$$(PC+4+(sign\ extend\ (imm)*2))$$

جدول ۵ انواع دستورات پرش و opcode آنها را نشان می دهد.

opcode	instruction	Operation
1001	beq	Branch if equal
1010	bne	Branch if not equal
1011	blt	Branch if less than
1100	bgt	Branch if greater than
1101	ble	Branch if less than or equal
1110	bge	Branch if greater than or equal

جدول ۵- مقادیر opcode برای دستورات پرش

۳- J-Type: دستور jmp از نوع J-Type می باشد. ماشین کد دستورات J-Type در شکل ۳ آورده شده است. در این دستور ۱۲ بیت پایین دستور (بیت‌های صفر الی ۱۱) در ۲ ضرب می شود (تبدیل به یک مقدار ۱۳ بیتی می شود) و سپس سه بیت بالای PC به ابتدای آن اضافه می شود تا یک مقدار ۱۶ بیتی ایجاد گردد و سپس در PC نوشته می شود. Opcode دستور jmp برابر "1111" می باشد.

$$PC \leftarrow PC[15..13] \&\& (instr[11..0]) \&\& "0"$$

15.....12	11.....0
4-bits	12-bit
Opcode (1111)	jump address

شکل ۳- قالب دستورات J-Type

موفق باشید