

دستور کار آزمایشگاه مجازی مدارهای منطقی و معماری کامپیوتر

کامپیوتر پایه دارای ویژگی‌های زیر است:

- هشت عدد رجیستر ۸ بیتی به نام‌های R_0 تا R_7 . این رجیستر بانک دارای دو پورت برای خواندن و یک پورت برای نوشتن است و خواندن از آن به صورت آسنکرون و نوشتن در آن به صورت سنکرون با کلاک و در لبه بالارونده کلاک است. این رجیستر بانک باید توسط دانشجویان در یک مازول کد زده شود و در مازول اصلی از آن نمونه گرفته و با آن کار شود.
- حافظه دستور (Instruction memory) که از این پس IM گفته می‌شود) به اندازه ۲۵۶ خانه ۱۶ بیتی. (پس طول آدرس IM هشت بیتی خواهد بود یعنی PC هشت بیتی است) همچنین با این حساب دستورات ۱۶ بیتی خواهند بود و در نتیجه IR (Instruction Register) نیز ۱۶ بیتی است. دانشجویان می‌توانند برای IM از IP Core های آماده موجود در ISE استفاده کنند. توجه کنید که عناصر حافظه موجود در IP Core ها سنکرون با کلاک هستند. (پیشنهاد می‌شود دیتاشیت آن‌ها را مطالعه مختصری کنید).
- حافظه داده (Data Memory) که از این پس به آن DM گفته می‌شود) به اندازه ۲۵۶ خانه ۸ بیتی. (پس طول آدرس DM هم هشت بیتی خواهد بود) DM نیز همانند IM از IP Core ها استفاده می‌شود. بدیهی است که باید از مازول های IM و DM هم نمونه‌ای گرفته شود و سیگنال‌های ورودی و خروجی آن به جاهای مناسب وصل شود.
- دیتا باس ۸ بیتی. یعنی همه کار با داده‌ها به صورت بیتی انجام می‌شود. (طول رجیسترها، عرض DM و طول داده‌هایی که به صورت مستقیم داده می‌شوند همگی ۸ بیتی است)
- کامپیوتر دارای ۴ فلگ است (CF-ZF-SF-OF) که این فلگ‌ها بعد از انجام بعضی از دستورات به روز می‌شوند. این که چه دستوری کدام فلگ را تحت تاثیر قرار می‌دهد در جدولی که در ادامه آمده است گفته می‌شود.
- کامپیوتر به صورت دو عملوندی است. این دو عملوند یا هر دو رجیستر هستند یا یکی رجیستر و دیگری یا آدرس (می‌تواند آدرس DM یا IM باشد) است یا داده. بیت MSB دستورات اگر ۰ باشد دستور از نوع دو رجیستری یا یک رجیستر و یک دیتا ۳ بیتی است و اگر ۱ باشد دستور از نوع یک رجیستر و دیگری دیتا یا آدرس است. در واقع در دستورات نوع اول دو عملوند ۳ بیتی هستند اما در دستورات نوع دوم عملوند اول ۳ بیتی و عملوند دوم ۸ بیتی است. این که قسمت عملوند دوم در نوع دوم دستورات آدرس IM یا آدرس DM و یا داده است از روی Opcode مشخص می‌شود و برای آن بیت اضافی در نظر گرفته نشده است. (به دلیل کم بودن تعداد بیت)

دستورات کامپیوتر به همراه Operand ها هر دستور، تعریف آن، کد اسمبلی آن و این که هر دستور چه Flag هایی را تحت تاثیر قرار می‌دهد، در جدول زیر نشان داده شده است:

	Instruction	Operands	Description	Flags Affected	assembly code
1	ADD	REG, REG	operand1 = operand1 + operand2	ALL Flags	0- 000000001- RRRRRR
2	AND	REG, REG	Logical AND between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0- 000000010- RRRRRR
3	SUB	REG, REG	operand1 = operand1 - operand2 sub with 2's complement add	ALL Flags	0- 000000011- RRRRRR

4	OR	REG, REG	Logical OR between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0- 000000100- RRR RRR
5	XOR	REG, REG	Logical XOR between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0- 000000101- RRR RRR
6	MOV	REG, REG	Copy operand2 to operand1	NONE	0- 000000110- RRR RRR
7	XCHG	REG, REG	Exchange values of two operands	NONE	0- 000000111- RRR RRR
8	NOT	REG	One's complement negate operand1 = ~ operand1	NONE	0- 000001000- RRR XXX
9	SAR	REG , immediate	Shift Arithmetic operand1 Right. The number of shifts is set by operand2. Shift all bits right, the bit that goes off is set to CF. The sign bit that is inserted to the left-most position has the same value as before shift.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0- 000001001- RRR III
10	SLR	REG , immediate	Shift operand1 Right. (Logical) The number of shifts is set by operand2. Shift all bits right, the bit that goes off is set to CF. Zero bit is inserted to the left-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0- 000001010- RRR III
11	SAL	REG , immediate	Shift Arithmetic operand1 Left. The number of shifts is set by operand2. Shift all bits left, the bit that goes off is set to CF. Zero bit is inserted to the right-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0- 000001011- RRR III
12	SLL	REG , immediate	Shift operand1 Left. The number of shifts is set by operand2. Shift all bits left, the bit that goes off is set to CF. Zero bit is inserted to the right-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0- 000001100- RRR III

13	ROL	REG , immediate	Rotate operand1 right. The number of rotates is set by operand2. shift all bits right, the bit that goes off is set to CF and the same bit is inserted to the left-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-000001101- RRRIII
14	ROR	REG , immediate	Rotate operand1 left. The number of rotates is set by operand2. shift all bits left, the bit that goes off is set to CF and the same bit is inserted to the right-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-000001110- RRRIII
15	INC	REG	operand = operand + 1. Increment by 1	ALL (same as ADD)	0-000001111- RRRXXX
16	DEC	REG	operand = operand - 1. Decrement by 1	ALL (same as SUB)	0-000010000- RRRXXX
17	NOP	NONE	NONE	NONE	0-000000000- XXXXXX
18	ShowR	REG	Show Value of Register (operand) on LEDs	NONE	0-000010010- RRRXXX
19	ShowR seg	REG	Show Value of Register (operand) on 7 segments	NONE	0-000010011- RRRXXX
20	CMP	REG, REG	operand1 - operand2 Result is not stored anywhere, flags are set (OF, SF, ZF, CF) according to result.	ALL	0-000010100- RRRRRR
21	JE	Address of Data Memory (IM)	address of memory that jumps to this if jump happened in operand2 and operand1 is don't care Condition: ZF = 1	NONE	1-0000- XXXAAAA AAAA
22	JB	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: CF = 1 Used for unsigned compare	NONE	1-0001- XXXAAAA AAAA
23	JA	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: CF = 0 , ZF=0 Used for unsigned compare	NONE	1-0010- XXXAAAA AAAA
24	JL	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: SF not equal to OF Used for signed compare	NONE	1-0011- XXXAAAA AAAA

25	JG	Address of Data Memory (IM)	address of memory that jumps to this if jump happened in operand2 and operand1 is don't care Condition: SF=OF , ZF=0 <i>Used for unsigned compare</i>	NONE	1-0100- XXXAAAA AAAA
26	JMP	Address of Data Memory (IM)	Address of memory that jumps to this in operand2 and operand1 is don't care. Jump without any condition	NONE	1-0101- XXXAAAA AAAA
27	LI	REG , immediate	Copy immediate value (operand2) to Register (operand1)	NONE	1-0110- RRRIIIIII
28	LM	REG, Address of Data Memory (DM)	Load data from data memory (address of memory in operand2) and copy to REG (operand1)	NONE	1-0111- RRRAAAA AAAA

نکات مربوط به دستورات فوق:

۱. مقدار immediate در دستورات ۹ تا ۱۴ حداکثر ۸ است. به همین دلیل فیلد immediate در این دستورات ۳ بیتی است و همانند یک رجیستر فضا اشغال می‌کند. به همین دلیل این نوع دستورات را همانند REG-REG در نظر می‌گیریم. یعنی در این دستورات بیت MSB کد، صفر است.
۲. از دستور ۲۱ دستورات نوع دوم (یکی از عملوندها ۳ بیتی و دیگری ۸ بیتی) شروع می‌شوند.
۳. در نمایش اسمبلی دستورات X به معنی Don't care، A به معنی آدرس (یا DM و یا IM)، R به معنی آدرس یکی از رجسترها و I به معنی مقدار می‌باشد. این مقدار در بعضی از دستورات ۳ بیتی و در بعضی از دستورات دیگر ۸ بیتی است.

❖ در این آزمایش بایستی پردازنده ساده تک چرخه‌ای پیاده‌سازی شود و تمامی مراحل طراحی مربوط به یک پردازنده با توجه به مطالب فراگرفته شده دانشجو در درس معماری کامپیوتر (دارای شمارنده برنامه، واحد کنترل، مسیر داده، برداشت دستور، دیکد و) رعایت شود. تعداد دستورها زیاد است اما تقریباً مشابه هم هستند و به راحتی قابل جایگزینی هستند، بنابراین تمامی دستورات در جدول توسط پردازنده پشتیبانی شود. دانشجویان گرامی کد مدار توصیف شده به همراه گزارش مناسبی از انجام آزمایش را تا مهلت تعیین شده توسط استاد مربوطه برای ایشان ارسال کنند. در گزارش ابتدا راجع به آزمایش (طرح مدار، جدول درستی، مزایا و معایب) توضیحی مفید ارائه کنید و سپس کد مدار توصیف شده را به همراه توضیحات مناسبی در گزارش شرح دهید. پس از اینکه طرح توصیف شده را با نوشتن Test Bench برای مدار شبیه‌سازی کردید، با انتخاب چند نمونه از ورودی‌ها، شکل‌های موج خروجی را در گزارش توضیح دهید. مجموع فایل‌های کد و گزارش را در قالب یک فایل فشرده شده برای استاد مربوطه ارسال کنید. نام فایل فشرده: Lab Number, Group Name.Zip