تمرین (۲) معماری کامپیوتر　　اشکان شکیبا　　۹۹۳۱۰۳۰

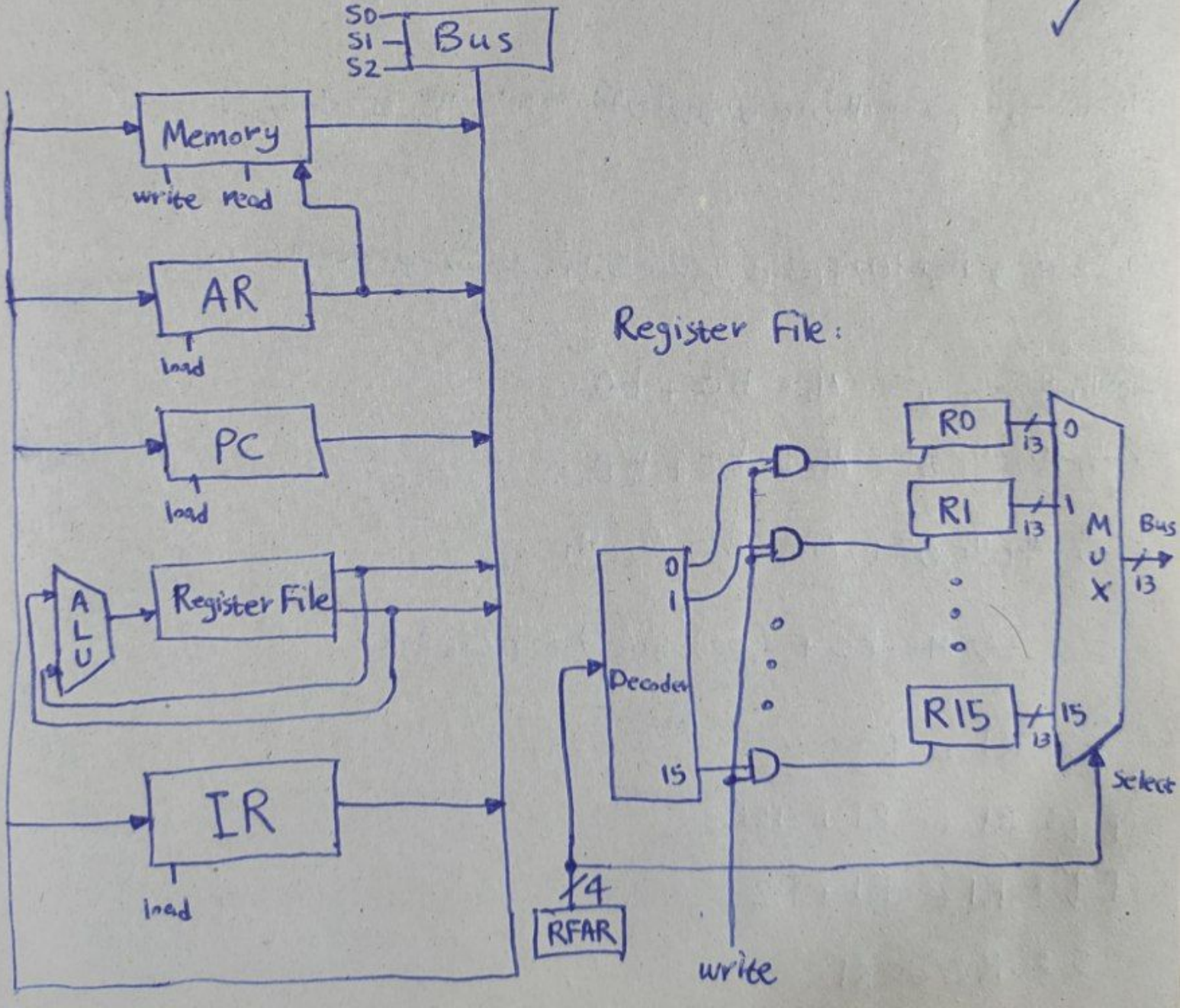
۱) ✓ دستورالعمل‌های نوع اول (۲ عملوند ثباتی) با ۴ بیت opcode : $2^4 = 16$

دستورالعمل‌های نوع دوم (یک عملوند حافظه) با ۳ بیت opcode : $2^3 = 8$

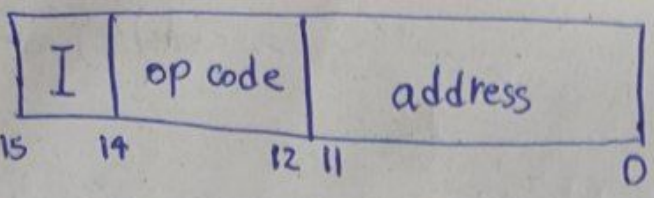
✓

ثبات‌های خاص منظوره	ثبات‌های عام منظوره	حافظه
PC 9 bit	13 bit	512 x 13 bit
AR 9 bit	x16 ← تعداد	
IR 13 bit		
RFAR 4 bit		

✓



۲) ✓ دستورالعمل‌های ۱۶ بیتی به فرمت روبرو:



سه نوع دستورالعمل داریم:

I) memory reference: اگر opcode≠111 که اگر I=0 آدرس‌دهی مستقیم به حافظه و اگر I=1 آدرس‌دهی غیرمستقیم خواهیم داشت. address حاوی آدرس عمل‌وند در حافظه است.

II) register reference: اگر opcode=111 و I=0، که با یک کردن یکی از بیت‌های ۰ تا ۱۱، نوع عملیات مشخص می‌شود.

III) I/O operation: اگر opcode=111 و I=1، که با یک کردن یکی از بیت‌های ۷ تا ۱۱ نوع عملیات مشخص می‌شود.

✓ در دستورالعمل‌های memory ref. امکان آدرس‌دهی به دو شکل مستقیم و غیرمستقیم را به ما می‌دهد.

در حالت opcode=111 مشخص‌کننده تفاوت دستورالعمل‌های register ref. و I/O op. است.

۳)

خروجی مالتی‌پلکسر بالا چپ: $M_1 = b'c' + bD$

خروجی مالتی‌پلکسر پایین چپ: $M_2 = bc + b'D$

خروجی مالتی‌پلکسر راست: $M = M_2 + M_2' M_1$

$$\Rightarrow M = bc + (bc)'(b'c') = bc + b'c'$$

۴)

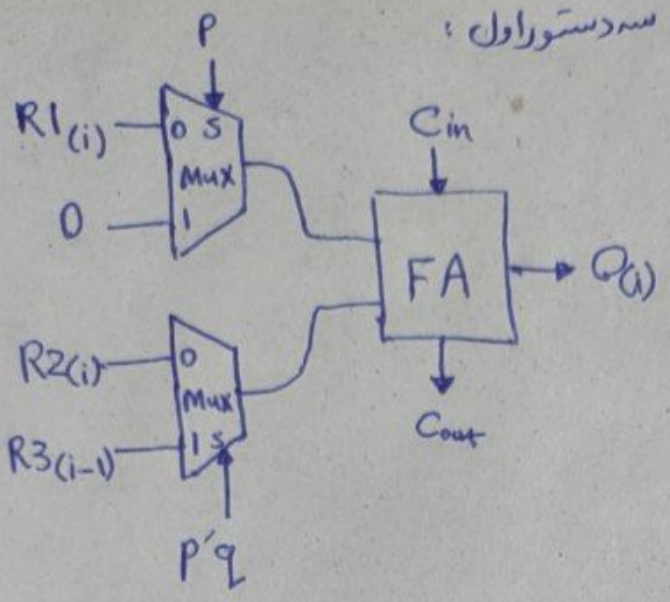
p: $R1 \leftarrow R2$

p'q: $R1 \leftarrow R1 + ShL\,R3$

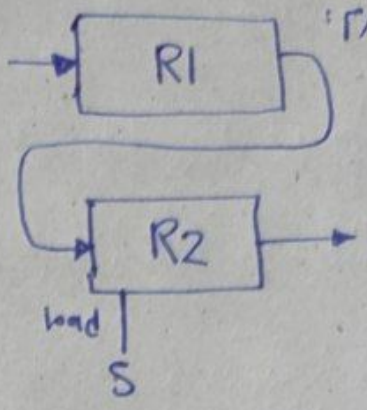
p'q': $R1 \leftarrow R1 + R2$

s: $R2 \leftarrow R1$

سه دستور اول :



دستور چهارم :



۵) ✓ نادرست ، هر خروجی دیکودر به لود یک رجیستر وصل است و ۷ رجیستر داریم .
پس یک دیکودر ۳x۸ می تواند کافی باشد.

✓ درست ، به ازای هربیت از هر رجیستر یک بافر سه حالته نیاز داریم پس ۱۶x۷ بافر نیاز است . همچنین برای هر رجیستر باید یک خروجی دیکودر داشته باشیم که با توجه به ۷ رجیستر نیاز به یک دیکودر ۳x۸ داریم .

✓ نادرست ، زبان های سطح بالا مستقل از معماری و ISA هستند و کامپایلر آنها می تواند یک برنامه یکسان را برای ماشین های متفاوت ترجمه کند .

✓ نادرست ، اگر در مدار جمع کننده ، R1 با q ، and بشود و فلیپ فلاپ ها در لبه بالا رونده کلاک load شوند ، conflict رخ نمی دهد .

۶) ✓ ISA تعیین می کند کامپیوتر چه کارهایی می تواند کند و کامپایلر از ISA برای نوشتن برنامه به زبان ماشین استفاده می کند . ریزمعماری یک سطح پایین تر از ISA است و مشخص می کند که پیاده سازی سخت افزاری ISA چگونه باشد بنابراین یک ISA یکسان می تواند با ریزمعماری های مختلف پیاده سازی شود .

1) ISA 2) MicroArch. 3) ISA, MicroArch. 4) MicroArch. ✓