



به نام خدا

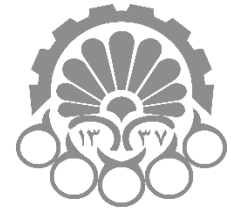
## پروژه اول

معماری کامپیوتر

دکتر شریعتمدار مرتضوی

نیمسال دوم 1401-1402

مهلت تحویل: 1402/02/15



دانشگاه صنعتی امیر کبیر

در پروژه اول می‌خواهیم به پردازش دو قسمت مهم از CPU پردازیم :

RAM & ROM (1

State Machine (2

بخش اول : RAM & ROM ✓

همانطور که میدانید در ROM دستورات ما ذخیره سازی میشود و در RAM دیتاهایی که می‌خواهیم را ذخیره سازی میکنیم پس یعنی در ROM ما فقط عملیات خواندن داریم و در RAM هم عملیات خواندن و هم عملیات نوشتن برای راحتی کار نیز دستورات را به صورت دستی میتوان درون ROM قرار داد و آن را تست کرد.

برای نوشتن کد و فهم بهتر برای نوشتن این قسمت به PDF داده شده مراجعه کنید.

بخش دوم : State Machine ✓

برای اینکه واحد مرکزی ما به درستی کار کند و تداخلی در برنامه به وجود نیاید نیازمند آن است که این با استفاده از Clock ما برنامه نویسی کنیم و برای برنامه نویسی با clk در زبان VHDL نیازمند به process هستیم. برای راهنمایی شما باید state 4 اصلی در نظر بگیرید :

- استیت دریافت دستور
- استیت دیکود کردن دستور
- استیت خواندن مقدار
- استیت انجام دستور

در این قسمت شما نیازمند یک سیگنال میانی هستید که کدام خانه از ROM هستید و بعد از دریافت دستورات یکی به این سیگنال اضافه شود تا در زمان استیت دریافت دستور بعدی، دستور بعدی را بخواند یا به زبان دیگر شما باید PC یا Program Counter را پیاده سازی کنید در این قسمت.

استیت شماره 1 برای خواندن دستورات از ROM است

استیت شماره 2 برای فهمیدن آن است که دستور خوانده شده به چه معنایی است و چه کاربردی دارد

استیت شماره 3 برای آن است که اگر دستور نیازمند به خواندن مقداری از RAM است آن را بخواند و در رجیستر DR (Data Register) ذخیره کند.

استیت شماره 4 انجام دستورات در واحد الگوریتم و منطق است (ALU).

واحد ALU ساخته شده از یک رجیستر 16 بیتی و یک رجیستر 1 بیتی است و بخش محاسبه است که دستورات مانند ADD یا And و ... را انجام میدهد

به رجیستر 16 بیتی AC و به رجیستر 1 بیتی E می گویند. رجیستر E رجیستر کمکی است که مقدار Carry جمع AC با DR را در خود ذخیره می کند

برای درک بهتر به لینک های زیر مراجعه کنید :

[ALU](#)

[AC](#)

قسمت سوم : دستورات

دستورات ما 16 بیت دارند و از دو بخش درست شده اند.

10 بیت سمت راست یعنی بیت 0 تا 9 مخصوص آدرس است و به دستوراتی است که نیازمند به آدرس هستند آدرس خانه RAM داده می شود. و اگر نیاز به آدرس نداشته باشد تمام 10 بیت صفر می شوند.

6 بیت سمت چپ Opcode است یعنی شماره دستور مثلا دستور 000001 دستور And است و این دستور نیازمند آدرس نیز است چون مقدار آدرس یک خانه را با AC ما And می کند و درون AC میریزد.

جدول دستورات به صورت زیر است :

	اسم دستور	نیازمندی به آدرس	Opcode	
ALU	AND	yes	000001	✓✓
	Store	Yes	000010	✓✓
	Load	Yes	000011	✓✓
ALU	Add	Yes	000100	✓✓
ALU	Increment AC	No	000101	✓✓
	Clear AC	No	000110	✓✓
	Clear E	No	000111	✓✓
ALU	Circular Left Shift	No	001000	✓
ALU	Circular Right Shift	No	001001	✓
skip if AC is possitive	SPA	No	001010	✓
skip if AC is negative	SNA	No	001011	✓
skip if E is zero	SZE	No	001100	✓
skip if AC is zero	SZA	No	001101	✓
ALU	linear Left Shift	No	001110	✓
ALU	Linear Right Shift	No	001111	✓
ALU	Multiply	Yes	010000	✓✓
ALU	SQR	Yes	100000	✓

دو دستور آخر با استفاده از ماژول‌هایی است که از قبل و در تمرین‌های قبلی آنها را ساخته‌اید.

نکته 1 : ماژول ضرب پیاده سازی شده در تمرین 6 بیت در 6 بیت است پس در اینجا ورودی آن نیز 6 بیت در 6 بیت می باشد و فقط یکی از 3 ماژول زده شده نیز مورد نیاز ما می باشد. خروجی آن نیز 12 بیت است و برای ذخیره سازی کافی است 0000 به سمت چپ آن اضافه شود.

نکته 2 : ماژول جذرگیر 16 بیتی می باشد ولی خروجی آن 8 بیت است و برای ذخیره سازی کافی است 8 بیت 0 به سمت چپ آن اضافه شود.

نکته 3 : پیاده سازی یکی از دو ماژول تمرین اجباری است یعنی یکی از ماژول های جذرگیر یا ضرب کننده باید در پروژه باشد و اگر جفت آنها باشند نمره امتیازی برای شما لحاظ می شود.

نکته 4 : به دلیل حجم زیاد کار پروژه اول به صورت گروه های 2 نفره می باشد و ارائه نیز گرفته می شود.

نکته 5 : این شکل کدها نیازمندی به Test Bench ندارند و تغییرات مستقیم در RAM و ROM شکل می گیرد و می توان خروجی را در سطح برنامه و stateها مشاهده کرد.

نکته 6 : کد و گزارش را یک نفر از اعضا گروه فقط آپلود کند.

نکته 7 : برای گزارش تمامی قسمت هایی که در تمرینات قبلی نوشته اید را بنویسید مثل نحوه عمل کرد برنامه و سطح مصرفی برنامه و ... و علاوه بر آن مشخص شود هر قسمت از برنامه را چه فردی انجام داده است.

نکته 8 : برای فهم بهتر دستورات به لینک [Inst](#) مراجعه کنید.

نکته 9 : چون 10 بیت مخصوص آدرس دهی ram داریم پس Ram ما  $16 \times 1024$  می باشد ولی برای راحتی کار شما  $16 \times 64$  در نظر بگیرید ولی دقت شود که دیگر آدرس های شما 6 بیتی می شود و باید 0000 بیت برای اینکه شکل ساختاری حفظ شود به سمت چپ آدرس های خود اضافه کنید.

موفق باشید