دانشگاه صنعتی امیرکبیر

«به نام خدا»

درس معماری کامپیوتر و ریزپردازنده

پروژه درسی ۱: طراحی و توصیف سخت افزار یک پردازنده

دانشکده مهندسی برق

دکتر احمد شعبانی

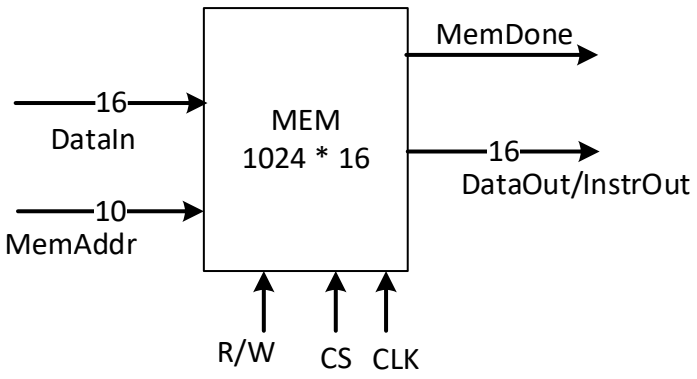
نیمسال اول سال تحصیلی ۱۴۰۲

نگارش: سید علی عبداللهیان (aliansgp@gmail.com)

هدف از این پروژه توصیف سخت‌افزار یک پردازنده ساده با توانایی اجرای چند دستورالعمل است. این سخت‌افزار از لحاظ معماری شباهت زیادی به معماری مانو دارد ولی ساده‌سازی زیادی در رابطه با آن انجام گرفته است. بطور کلی معماری هدف این پروژه شامل بخش مسیر داده پردازنده (Datapath)، کنترلر پردازنده (ماشین حالت) و حافظه است که در نهایت هر یک از قسمتهای نامبرده را مطابق توضیحات داده شده از طریق زبان VHDL توصیف و شبیه‌سازی می‌کنیم.

بخش اول: طراحی واحد حافظه

در این معماری حافظه داده و دستورالعمل بصورت مشترک است. شکل ۱ شمای کلی حافظه مورد انتظار را نشان می‌دهد. این حافظه دارای ۱۰۲٤ خانه ۱٦ بیتی است و در نتیجه تعداد بیتهای آدرس ما ۱۰ بیت است. فرض شود که آدرس ۰ تا ۵۱۱ مربوط به بخش ذخیره‌سازی دستورالعمل و مابقی خانه‌های حافظه از آدرس ۵۱۲ تا ۱۰۲۳ مربوط به بخش ذخیره‌سازی داده است. برای تست و شبیه‌سازی اولیه عمکرد حافظه می‌توانید مقدار اولیه چند خانه حافظه مربوط به بخش دستورالعمل و داده را بصورت دلخواه مطابق با اطلاعات جدول ۱ تنظیم نمایید. این مقادیر اولیه مشابه یک تست بنچ برای شما به شمار می‌روند و بر اساس مقدار آنها می‌توانید عملکرد کد خود را تست و شبیه‌سازی نمایید.



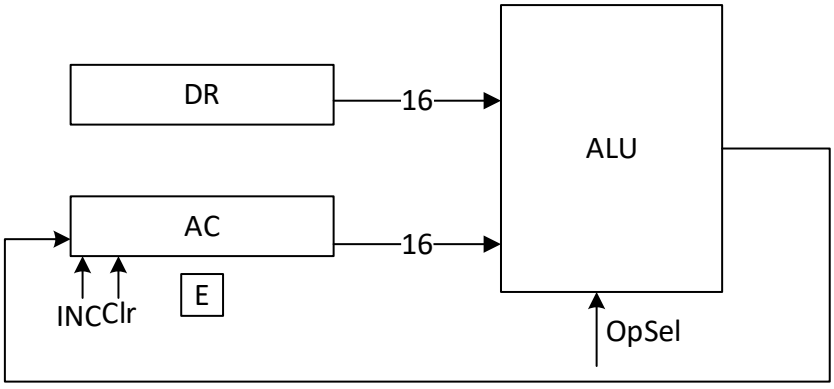
شکل ۱. ورودی–خروجی بخش حافظه پردازنده پروژه

اگر CS= 1 باشد یعنی عملیات حافظه داریم (مانند سیگنال Enable برای حافظه بشمار می‌رود) در این صورت آدرس خانه مورد نظر برای عملیات نوشتن یا خواندن را از روی ۱۰ بیت آدرس ورودی خوانده و پس از مشخص کردن نوع عملیات داده یا دستورالعمل ۱۶ بیتی را از خانه حافظه خوانده و روی باس داده خروجی قرار می‌دهیم (عملیات Load یا Fetch دستورالعمل) یا اینکه داده ۱۶ بیتی ورودی را گرفته و بر روی خانه مورد نظر در حافظه می‌نویسیم (عملیات Store). نوع عملیات خواندن و نوشتن را سیگنال R/W مشخص می‌کند. R/W=1 به معنای عملیات خواندن و R/W=0 به معنای عملیات نوشتن در حافظه است. در صورتیکه CS=0 باشد هیچ عملیاتی روی حافظه انجام نمی‌گیرد. سیگنال خروجی حافظه MemDone نشان دهنده وضعیت اتمام عملیات دسترسی به حافظه است در صورتیکه این سیگنال یک باشد به معنای اتمام دسترسی است بدین معنی که در عملیات خواندن، داده مورد نظر روی باس خروجی قرار گرفته است یا اینکه عملیات نوشتن به اتمام رسیده است. در صورتیکه هنوز داده آماده نباشد باید این سیگنال وضعیت صفر را نشان دهد. زمانبندی و تاخیر عملیات دسترسی به حافظه به دلخواه انجام گردد. در نهایت بر اساس فرضیات فوق، با زبان VHDL حافظه مورد را توصیف نمایید.

بخش دوم: طراحی و توصیف مسیر داده

مسیر داده در این کامپیوتر فرضی مطابق با شکل ۲ می‌باشد. این مسیرداده شامل تعدادی ریجیستر و واحد منطق و محاسبات (ALU) است. ریجسترهای مورد نظر شامل ریجسترهای ۱۶ بیتی DR, AC, IR و دو ریجستر ۱۰ بیتی PC و AR است. مسیر داده با گرفتن اطلاعاتی از بخش کنترلر نظیر OpSel تصمیم می‌گیرد که چه عملیاتی را انجام دهد. بعنوان مثال، پردازنده بعد از عملیات Decode نوع عملیات و داده مورد نظر برای عملیات را

مشخص و آماده می‌کند. بعد از آن، داده‌ها برای اجرای دستور وارد بخش ALU شده و متناسب با نوع Opcode عملیات انجام گرفته و نتیجه آن روی ریجیستر مقصد که همان AC است، نوشته می‌شود. دقت شود که برای عملیات Load و Store نیازی به مراجعه به بخش ALU نیست و در واقع ALU کاری در این حالت انجام نمی‌دهد و پردازنده صرفاً به بخش حافظه رجوع می‌کند. جدول ۱ نوع دستورالعملهای قابل پشتیبانی این کامپیوتر مشابه با دستورات مانو بهمراه Opcode دستورالعملها را نشان می‌دهد. برای فهم بهتر عملکرد هر یک از دستورات جدول ۱، به بخش مورد نظر در درس مراجعه نمایید.



شکل ۲. شماتیک مسیر داده پردازنده فرضی شامل ریجسترها و بخش ALU

جدول ۱. فهرست دستوالعملهای پردازنده فرضی و توصیف اجزای آن

اسم دستور	نیازمندی به آدرس	Opcode
AND	yes	000001
Store	Yes	000010
Load	Yes	000011
ADD	Yes	000100
Increment AC (INC)	No	000101
Clear AC (CLA)	No	000110
Clear E (CLE)	No	000111
Circular Left Shift (CIL)	No	001000
Circular Right Shift (CIR)	No	001001
Halt	No	001010

مطابق با اطلاعات جدول ۱، در این کامپیوتر پایه، دستورات ما ۱۶ بیتی هستند و از دو بخش آدرس داده ۱۰ بیتی و بخش Opcode ۶ بیتی تشکیل شده‌اند. ۱۰ بیت سمت راست یعنی بیت ۰ تا ۹ مخصوص آدرس داده دستورالعمل

(Operand) است و به دستوراتی است که نیازمند به آدرس هستند آدرس خانه حافظه داده می‌شود. درصورتیکه دستورالعمل نیاز به آدرس نداشته باشد (نظیر Clear AC) تمام ۱۰ بیت آدرس آن صفر است. ٦ بیت سمت چپ Opcode است یعنی مشخص کننده نوع دستور، بعنوان مثال دستور ۰۰۰۰۰۱ نشان دهنده دستور AND است و این دستور نیازمند آدرس برای مشخص کردن Operand نیز است یکی از متغیرها برای عملیات در AC موجود است و دیگری در حافظه به آدرسی است که بخش آدرس دستورالعمل مشخص می‌کند (رجوع شود به بخش پردازنده در درس). توجه داشته باشید که کامپیوتر مورد نظر در صورت دریافت سیگنال ورودی START= 1 از خانه صفر حافظه شروع کرده و در هر بار عملیات Fetch یک دستورالعمل را از حافظه خوانده و عملیات Decode و اجرای آن را انجام می‌دهد. در نهایت یکی به آدرس دستورالعمل (PC) خود اضافه کرده و فرآیند مجددا تا رسیدن به دستور Halt برای دیگر دستورالعملها تکرار می‌گردد. در صورت دریافت دستور Halt پردازنده وارد حالت Halt شده و هیچکاری انجام نمیدهد تا مجددا سیگنال Reset را دریافت کند. پیاده سازی ثبات E نیز فراموش نشود.

بخش سوم: طراحی بخش کنترلر پردازنده (ماشین حالت):

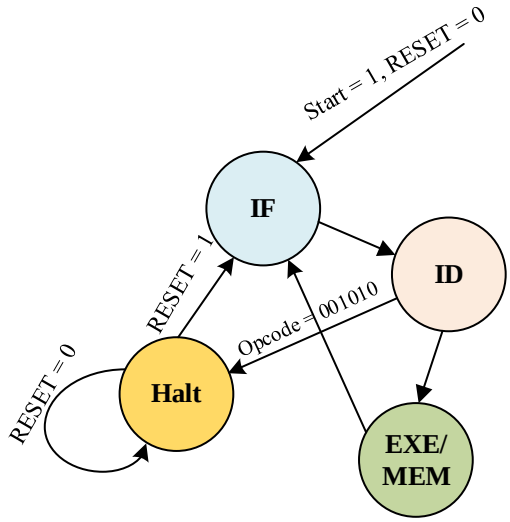
برای اینکه واحد مرکزی ما به درستی کار کند و تداخلی در برنامه به وجود نیاید نیازمند آن است که تمامی مراحل اجرا در این پردازنده با سیگنال کلاک سنکرون شود. بخش کنترلر این پردازنده شامل چهار حالت (State) کلی است که در شکل ۳ نمایش و در زیر توضیح داده شده است.

۱. حالت Fetch: در این حالت دستورات از خانه صفر حافظه (مقدار اولیه PC برابر صفر است) شروع به لود شدن می‌کنند و در یک ریجستر IR ریخته می‌شوند. آدرس دسترسی به خانه حافظه برای دستورالعمل در ریجستر PC قرار دارد. در هر بار عملیات Fetch مقدار ریجیستر PC یکی اضافه می‌شود.

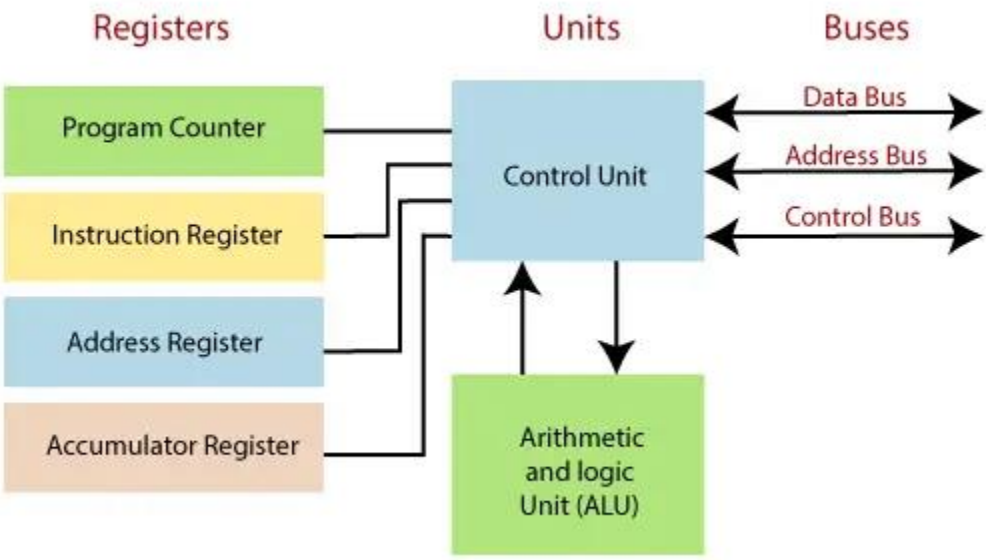
۲. حالت Decode: در این بخش دستورات خوانده شده که در ریجستر IR وجود دارد بررسی شده و نوع عملیات و آدرس دسترسی به Operand دستورالعمل (در صورت لزوم) مشخص می‌شود. در صورتیکه نیاز به Operand داریم عملیات دسترسی به حافظه انجام می‌گیرد و در نهایت Operand مربوطه روی ریجستر DR ریخته می‌شود. تعداد کلاک لازم در این مرحله به دلخواه توسط کاربر می‌تواند تعیین شود.

۳. حالت EXE/MEM Access : در این بخش عملیات مربوط به اجرای دستورالعمل انجام می‌گیرد. این عملیات یا از طریق خود ALU انجام می‌گیرد یا اینکه عملیات مربوط به دسترسی به حافظه است.

٤. حالت Halt: در این حالت پردازنده stall می‌شود و هیچکاری انجام نمی‌دهد. تنها سیگنالی که می‌تواند پردازنده را از این حالت خارج کند سیگنال RESET است. در صورت دریافت Opcode = 001010 در مرحله دیکود پردازنده مستقیما وارد حالت Halt می‌شود و اجرای برنامه در این مرحله به پایان می‌رسد.



شکل ۲. ماشین حالت کنترلر پردازنده فرضی



در شکل بالا ارتباط میان واحد های مختلف دیده می شود که کنترلر وظیفه تحلیل دستورات و ارتباط میان رجیستر ها و ALU و حافظه را (با باس دیتا و آدرس) بر عهده دارد. توجه داشته باشید که شما باید مراحل مختلف ماشین حالت را مرحله به مرحله پیاده سازی کنید بدین صورت که در مرحله اول دستور از خانه حافظه که PC آن را مشخص می‌کند خوانده شده و در رجیستر IR ریخته شود، پس از آن دستور داخل رجیستر IR دیکد شده و مشخص می‌شود که آیا Operand مربوط به دستور، نیاز به مراجعه به حافظه دارد یا خیر. در نهایت نیز عمل مربوط به واحد ALU رفته یا در صورتی که نیاز به دسترسی به حافظه داشته باشد، عمل آن با توجه به مقدار خوانده شده از حافظه انجام می پذیرد.

برای درک بهتر نحوه پیاده سازی ماشین حالت در پردازنده حتما این لینک را مطالعه فرمایید:

https://vhdlwhiz.com/finite-state-machine/

نکته ۱ : این شکل کدها نیازمندی به Test Bench ندارند و تغیرات مستقیم در حافظه شکل می‌گیرد و می‌توان خروجی را در سطح برنامه و حالتهای ماشین حالت مشاهده کرد.

نکته ۲: استفاده از کد های موجود در اینترنت در صورت تغییر و درک آنها بلا مانع است.

تاریخ تحویل: ساعت ۲۳:۵۵ روز ۱۲ بهمن ماه

بارمبندی

- **بخش ۱: ۳۰ نمره**
- **بخش ۲: ۳۵ نمره**
- **بخش ۳: ۳۵ نمره**

با آرزوی موفقیت برای شما