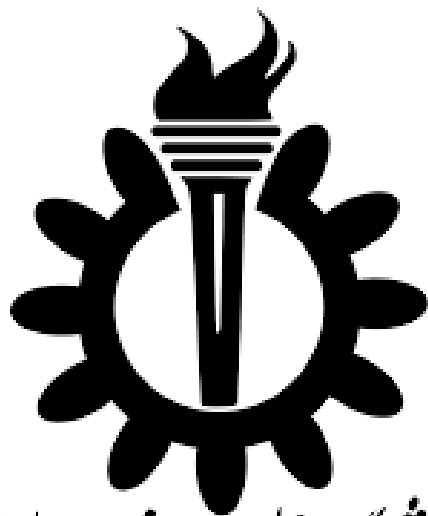


بنام خدا



دانشگاه علم و صنعت ایران

پروژه درس CAD

استاد: دکتر میثم عبدالحی

انجام دهنده :

الناز رضایی

حوریه سبزواری

تیر ۱۴۰۱

فهرست

مقدمه	3
بخش اول پروژه	4
بخش اول امتیازی	6
بخش دوم امتیازی	7
Test bench	8

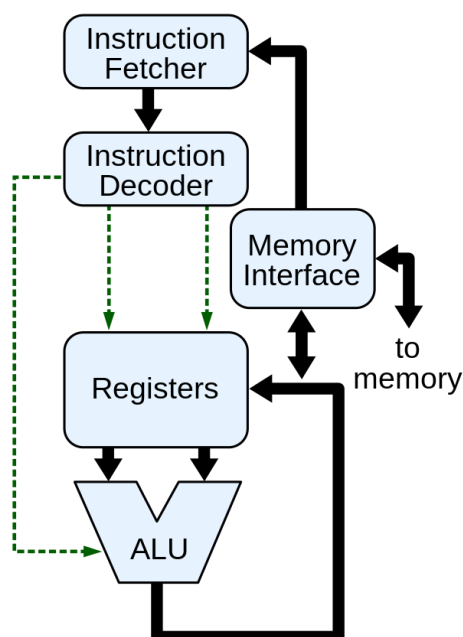
مقدمه

ALU پردازنده بیشتر به عنوان بخشی از پردازنده طراحی و ایجاد شده است تا حسابت را روی داده های منطقی از نوع اعداد صحیحی و یا Integer انجام دهد. این واحد تنها واحدی است که در پردازنده وظیفه محاسبه جمع و تفریق ها را بر عهده دارد. همچنین در خصوص عملیات ضرب ALU پردازنده می تواند به راحتی عملیات ضرب را بین دو عدد صحیح از نوع Integer انجام دهد.

بر اساس نوع محاسبه و نوع پردازنده ای که اجرا می شود ممکن است خروجی های واحد محاسبه و منطق در بخش های مختلف استفاده شوند. در رایانه ها خروجی این واحد عموماً بر روی RAM ها قرار می گیرد. واحد ALU خود از دو واحد داخلی تشکیل شده است و. این واحد ها با نام های:

۱. واحد محاسبه که بخش های محاسباتی ریاضی را انجام می دهد و با نام AU شناخته می شود که مخفف کلمه Arithmetic Unit است.

۲. واحد منطق که عملیات های منطقی را روی داده ها انجام می دهد و با نام LU که مخفف کلمه Logic Unit است شناخته می شود.



بخش اول پروژه

در این پروژه ما قرار است که یک ALU بسازیم که عملگرهای زیر را انجام دهد:

- ۱- جمع
- ۲- تفریق
- ۳- تقسیم
- ۴- ضرب
- ۵- لگاریتم بر مبنای دو
- ۶- جذر
- ۷- توان

کدام قرار است چهار عدد صحیح که دو به دو در پرانتز هستند را طبق عملگرهای ورودی محاسبه کند و جواب خروجی را نمایش دهد.

ابتدا ما قرار یک عدد ۴۱ بیتی را دریافت کرده سپس تا ۳۲ بیت اول ما برای ۴ عدد ۸ بیتی در نظر می‌گیریم و ۹ بیت بعدی برای ۳ عملگر ما می‌باشد.

در ادامه هر ۸ بیت را که یک عدد است به decimal تبدیل کرده و طبق عملگرها ابتدا پرانتز چپ و سپس پرانتز راست محاسبه می‌کنیم.

پاسخ حاصل از این دو پرانتز را در دو متغیر میانی ذخیره کرده و در قسمت پایانی عملگر بین دو پرانتز را در نظر گرفته و محاسبه می‌کنیم و در output قرار می‌دهیم.

برای مثال:

اگر دستور ورودی برابر

۰۰۰۰۱۰۱۰۰۰۰۰۱۱۰۰۰۰۰۰۱۱۰۰۰۰۰۰۱۰۰۰۰۰۰۱۰۰ باشد

حاصل ALU باید حاصل زیر را نمایش دهد: $(12+6)*(2^4)$

نکته: ما در انجام این پروژه از variable استفاده کردیم زیرا قرار است خط به خط کد را اجرا کند و نیاز به اجرای همزمان نیست.

بخش اول امتیازی

در بخش اول امتیازی ما تعداد بیت هارا تغییر می دهیم و به ۴۹ بیت می‌رسانیم. این ۴۹ بیت در واقع یک ۳ بیت آخر آن برای ثبت شدن در کدام رجیستر و تک بیت تک بیت برای ۴ عدد میباشد به گونه ای که عدد ۸ بیتی به ۹ بیتی تغییر کرده و بیت اخر برای این است که اگر ۰ بود خود همون عدد ورودی در ALU استفاده میکنیم اگر ۱ بود از رجیستر خوانده میشود. گونه ی خواندن به این طور است که عدد ورودی اگر بین ۰ تا ۳۲ بود از رجیستر اول ۳۳ تا ۶۴ از رجیستر دوم و به همین ترتیب تا رجیستر هشتم.

ما در ابتدا با ۸ portmap ۸ رجیستر ساختیم . در آخر نیز آن ۳ بیت selct که در انتهای آن ۴۹ بیت بود به ما نشان می‌دهد که در کدام رجیستر ذخیره شود و برای ذخیره سازی ما نیاز به کلاک داریم که در کلاک دوم با دیدن ۱ شدن اون write مورد نظر در رجیستر ذخیره شود.

بخش دوم امتیازی

ما یک module دیگر می‌سازیم برای استفاده از این ALU مورد نظر، اینگونه که ما برای مثال ۸ تا ورودی ۴۹ بیت دریافت کرده و یک n نیز میگیریم از ورودی برای اینکه بدانیم تا کدام ورودی ما از ALU استفاده کنیم. مثلاً اگر ۳ بود ما for تا ماکسیم ۳ میبریم و ۳ تا دستور اول را انجام می‌دهیم.

Test bench:

