



معماری کامپیوتر (بهار ۹۸)

مدرس: دکتر حسین اسدی

پروژه پایان ترم

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

موعد تحویل^۱: ۱۵ تیر ۹۸

لطفا به نکات زیر توجه نمایید:

- یکی از پروژه‌ها را انتخاب کرده و در این مسیر شماره دانشجویی اعضای گروه (گروه‌های دو نفره در تمرین ۵) را در مقابل پروژه وارد نمایید. با توجه به تعداد نفرات کلاس انتظار می‌رود ۱۲ گروه در پروژه اول و ۱۲ گروه در پروژه دوم ثبت نام کنند. اولویت با دانشجویانی است که زودتر اقدام به ثبت نام نمایند.
- خروجی پروژه را به همراه مستندات کافی و تصاویر لازم در قالب فایل فشرده و با عنوان Proj_X_Y که X و Y شماره دانشجویی اعضای گروه است، در سامانه CW بارگزاری نمایید.
- در صورت استفاده از منابع، حتما به آنها ارجاع دهید.
- پروژه تحویل حضوری دارد و هر دو عضو گروه بایستی روی پروژه تسلط کافی داشته باشند.
- مهلت پروژه به هیچ عنوان تمدید نخواهد شد.
- سوالات و اشکالات پروژه را تنها در فروم مشخص شده در CW مطرح نمایید.

پروژه اول: پردازنده خط لوله

در این پروژه هدف طراحی و آزمایش عملکرد یک پردازنده‌ی خط لوله است.

- استفاده از تمامی ماژول‌های طراحی شده در تمرین‌های قبلی اعضای گروه مجاز است.
- استفاده از هر تعداد ثبات میانی با هر اندازه‌ای در پروژه مجاز است. دقت کنید مانند تمرین قبل تمامی ثبات‌ها باید ساخته‌ی خودتان باشد و همچنان استفاده از ماژول‌های آماده‌ی dpm، یا استفاده از ثبات‌های ساخته شده توسط دیگر دانشجویان مجاز نیست.
- تمامی ثبات‌های پردازنده، به همراه بانک ثبات اصلی باید دارای یک درگاه ریست سنکرون باشند که به سیگنال ریست پردازنده وصل شده است و با یک شدن آن، در لبه‌ی کلاک تمامی ثبات‌ها مقدار صفر به خود می‌گیرند.
- در این پروژه برای پیاده‌سازی حافظه، چهار فایل در پوشه‌ی memory پیوست شده است. این فایل‌ها را به پوشه‌ی پروژه‌ی خود منتقل کرده و از ماژول آنها در پروژه استفاده نمایید.
- مانند تمرین‌های درس، طراحی ماژولار و قابل فهم، بخش مهمی از نمره‌ی شما را در بر می‌گیرد.

مشخصات ماژول حافظه

- دو حافظه‌ی مجزا برای حافظه‌های داده و دستور با نام‌های instmemory و datamemory در نظر گرفته شده است.
- این حافظه‌ها دارای یک درگاه آدرس اصلی با نام addr به طول ۱۶ بیت و یک درگاه اصلی ورودی داده به نام datain و به طول ۳۲ بیت است. همچنین یک درگاه یک بیتی با نام write نیز مشخص کننده‌ی عمل خواندن (write=1) و یا عمل نوشتن (write=0) خواهد بود.
- این ماژول دارای ۱۶۲ ردیف ۳۲ بیتی از اعداد است. بدین ترتیب آدرس‌دهی به حافظه با اعداد ۱۶ بیتی امکان‌پذیر است و پهنای هر خط از حافظه ۳۲ بیت است.

^۱مهلت به هیچ عنوان تمدید نخواهد شد.

- برای مقداردهی اولیه به این حافظه، فایل با پسوند v. را مشاهده نموده و داده‌ی مربوط به آدرس مورد نظر را تغییر دهید.
- همچنین می‌توانید برای خانه‌هایی که مقداردهی نشده‌اند خط مربوطه را به کد اضافه نمایید.
- درگاه reset حافظه به صورت سنکرون و حساس به لبه‌ی مثبت است.
- سیگنال‌های پرچم که در تمرین ساخت ALU طراحی شدند، همچنان در این پردازنده به همان حالت حضور داشته و استفاده می‌شوند.
- بانک ثبات استفاده شده در این پردازنده، مشابه تمرین‌های قبلی شما شامل ۳۲ ثبات ۳۲ بیتی است.

بررسی صحت عملکرد

برای بررسی صحت عملکرد این پردازنده، باید دو برنامه‌ی مجزا بنویسید و فایل حافظه‌ی شامل این دو برنامه را به همراه پروژه بارگذاری نمایید.

برنامه‌ی اول، ضرب ماتریس: این برنامه برای ضرب دو ماتریس پنج در پنج است. ماتریس اول به صورت row-order major از خانه‌ی صفر تا ۲۴ حافظه، و ماتریس دوم از خانه‌ی ۲۵ تا ۴۹ حافظه ذخیره شده است که باید به دلخواه به آن مقدار دهید. در نهایت نتیجه‌ی ضرب دو ماتریس باید به صورت یک ماتریس ۵ در ۵ از خانه‌ی ۵۰ تا ۷۴ حافظه ذخیره شود.

برنامه‌ی دوم، مرتب‌سازی آرایه: این برنامه به منظور مرتب‌سازی عناصر یک آرایه به طول ۱۰۰ است که در خانه‌ی صفر تا ۹۹ حافظه ذخیره شده است. اعضای آرایه باید به صورت صعودی از خانه‌ی ۱۰۰ تا ۱۹۹ حافظه ذخیره شوند. الگوریتم مورد استفاده در مرتب‌سازی دلخواه است.

- به منظور تایید صحت عملکرد پردازنده، نیاز خواهد بود تا مقدار ثبات‌های میانی خط لوله را به عنوان خروجی پردازنده هم در نظر بگیرید تا بتوان مقادیر آنها را حین اجرای پردازنده توسط فایل Waveform بررسی کرد.
- پردازنده‌ی طراحی شده باید قابلیت اجرای forwarding هایی شامل ALU-ALU و Mem-ALU را داشته باشد.
- نکته‌ی مهم: لازم است خروجی حافظه‌ی داده به عنوان خروجی اصلی پردازنده در نظر گرفته شود و پس از اتمام برنامه، داده‌های مورد نظر در سیکل‌های متوالی از حافظه خوانده و نمایش داده شوند.

پروژه دوم: Gem5

هدف از این پروژه، بهبود کارایی به‌وسیله افزودن دستورات خاص منظوره به پردازنده و همچنین افزودن حافظه نهان (Cache) می‌باشد. به این منظور، الگوریتم SHA-256 مدنظر قرار می‌گیرد. مجموعه SHA-2 مجموعه‌ای از توابع درهم‌سازی در رمزنگاری محسوب می‌شود. SHA-256 به عنوان یکی از معروف‌ترین شاخه‌های آن محسوب می‌شود. در این پروژه، ISA پردازنده X86 برای اجرای SHA-256 بهینه‌سازی می‌شود. پروژه شامل بخش‌های زیر می‌باشد:

- ۱- در ابتدا، تحلیل خود را از دستوراتی که اضافه شدن آن‌ها به ISA می‌تواند کارایی این برنامه را بهبود دهد، ارائه دهید و یک دستور را جهت اضافه شدن به ISA انتخاب کنند. دستور انتخابی شما نباید جزو دستورات رایج ISA باشد.
- ۲- در مرحله بعد، بر اساس توضیحات ارائه شده در کلاس، دستور مدنظر را به ISA اضافه کنید. شما باید سه مقدار تأخیر متفاوت را برای دستور اضافه شده مدنظر قرار بدهید و نشان دهید اضافه شدن دستور اختصاصی، نسبت به ISA اصلی، چه تأثیری بر روی کارایی برنامه گذاشته‌است.
- ۳- در این مرحله برای اجرا از فایل تنظیمات (config) نوشته شده در تمرین قبلی (تمرین دوم gem5) استفاده کنید. مشخصات حافظه و پردازنده را می‌توانید به شکل دلخواه تغییر دهید.
- ۴- در مرحله بعد، یک حافظه نهان دو سطحی به فایل تنظیمات اضافه کرده و تأثیر اجرای برنامه در حالت عادی و تغییر یافته (بخش دوم) را در حضور حافظه نهان، تحلیل کنید. سطح اول این حافظه‌ی نهان دارای دو حافظه‌ی جداگانه برای دستور و داده به اندازه‌های ۳۲KB و ۶۴KB می‌باشد. حافظه‌ی سطح دوم از یک حافظه برای دستورات و داده استفاده کرده و اندازه‌ی آن ۱MB است.

۵- برای سنجیدن عملکرد سیستم با تنظیمات مختلف برای حافظه‌ی نهان راحت‌تر است پارامترهایی مانند اندازه و cacheline و associativity را به فایل تنظیمات خود اضافه کرده و به جای هر بار تغییر کد فایل تنظیمات، این پارامترها را از طریق ترمینال تغییر دهید. بعد از اضافه کردن این پارامترها ابتدا اندازه‌ی حافظه‌ی نهان داده و دستور سطح اول و اندازه‌ی حافظه‌ی سطح دوم را به صورت جداگانه با فرض ثابت بودن سایر اندازه‌ها دو برابر کنید. استنباط خود از این نتایج را گزارش کنید. سپس، مقادیر حافظه‌ها را همان مقادیر اولیه در نظر گرفته و برای cacheline مقدار ۳۲ و ۶۴ فرض کرده و آزمایش را تکرار کنید. در نهایت برای associativity حافظه‌ی نهان سطح دوم مقادیر ۸ و ۱۶ را فرض کرده و نتایج آزمایش با تحلیل خود را گزارش کنید (مجموعاً ۷ شبیه‌سازی برای این بخش لازم است).

نکات:

- با توجه به این که هدف از این پروژه، ارزیابی تسلط شما بر روی شبیه‌ساز gem5 است، استفاده از پیاده‌سازی‌های آماده الگوریتم SHA-256 نیز مجاز است.
- تمامی اطلاعات مورد نیاز جهت چگونگی انجام موارد فوق در ارائه‌های کلاسی بیان شده‌است. در صورت نیاز به اطلاعات بیشتر، می‌توانید از اطلاعات موجود در سایت gem5 استفاده کنید.

خروجی مورد انتظار

- مستندات شامل فایل‌های word و pdf به همراه توضیحات کافی. مستندات بایستی به نحوی تهیه شوند که نشان‌دهنده مراحل انجام کار توسط اعضای گروه باشند. مرتب بودن مستندات (شامل فصل‌بندی، استفاده از فونت‌های مناسب، زیرنویس شکل و... همگی شامل نمره اضافه هستند).
- گزارش نهایی شما از دو جنبه کیفیت پیاده‌سازی و رعایت استانداردهای نگارشی مورد ارزیابی قرار خواهد گرفت. لطفاً زمان کافی به تهیه گزارش اختصاص دهید تا حق شما ضایع نشود.
- کلیه فایل‌های پیاده‌سازی
- تحویل حضوری (زمان تحویل متعاقباً اعلام خواهد شد).

موفق باشید