



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۳-۱۴۰۲
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل: ۱۴۰۲/۰۹/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

دستور کار:

- ❖ پس از ۱۰ روز تاخیر مجاز در مجموع کل تمرینات (کامپیوتری + دستی)، تحویل تمرین با تاخیر شامل جریمه خواهد بود (هر روز ۲۵ درصد).
- ❖ نام فایل ارسالی در قالب ACA_CA(number)_studentID بارگذاری شود.
- ❖ تمامی پروژه‌ها از لحاظ شباهت کنترل و بررسی می‌شوند، بنابراین از کپی کردن خودداری فرمایید. در صورت مشاهده شباهت در پاسخ دانشجویان، نمره تمرین صفر خواهد بود و در مجموع نمرات جریمه در نظر گرفته خواهد شد.
- ❖ راه ارتباطی با حل تمرین

sara.zamani73@aut.ac.ir



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۳-۱۴۰۲
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل: ۱۴۰۲/۰۹/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

هدف این تمرین آشنایی با زبان gem5 برای توصیف مجموعه دستورات است. برای این کار فایل‌های ISA موجود در مسیر src/arch/x86/isa مورد بررسی قرار می‌گیرند. شما ابتدا مطابق توضیحات یک دستور از معماری x87 با نام (FSUBR) را برای x86 پیاده‌سازی می‌کنید. سپس برای تست دستور پیاده‌سازی شده، یک برنامه می‌نویسید که از این دستور خاص با استفاده از ویژگی Inline assembly استفاده کند. سپس برنامه را با استفاده از gem5 شبیه‌سازی می‌کنید.

توضیح مراحل

در دستورات x86 معمولاً هر دستور به عنوان ترکیبی از بخش‌های کوچکتر پیاده‌سازی می‌شود. به طور کلی هر دستور به عنوان (macro-op) و بخش‌های کوچکتر به عنوان (micro-op) شناخته می‌شوند. برای پیاده‌سازی یک دستورالعمل در gem5، ابتدا اطلاعات مربوط به macro-op به دیکودر ISA ارائه می‌شود. سپس macro-op به صورت تعدادی micro-op پیاده‌سازی می‌شود و در نهایت، micro-opهایی که قبلاً پیاده‌سازی نشده‌اند پیاده‌سازی می‌شوند. این مراحل برای پیاده‌سازی دستور FSUBR مورد استفاده قرار می‌گیرد و مشابه پیاده‌سازی FSUB خواهد بود که از قبل در gem5 موجود است.

در x86 ISA، دستورات به چندین روش مختلف کدگذاری می‌شوند. که در این تمرین تمرکز بر روی زیرمجموعه x87 است (برای مطالعه بیشتر در مورد کدگذاری دستورات می‌توانید به [فایل راهنمای ارائه شده توسط AMD](#) مراجعه کنید).

ابتدا با مراجعه به فایل src/arch/x86/isa/decoder/one_byte_opcodes.isa می‌توان بررسی کرد که در gem5، دیکودر دستورات ISA x86 به چه نحو انجام می‌شود. در این فایل دستورات با ساختار مشخصی تعریف شده‌اند و محتوای آن در نهایت به یک Switch Case در زبان C++ تبدیل می‌شود. برای این کار ابتدا ۵ بیت اول بایت مربوط به opcode دیکود می‌شود که امکان ایجاد ۳۲ حالت مختلف را دارد که به ترتیب مشخص شده است. تمام دستورات x87 با یک بایت opcode در محدوده 0xD8 تا 0xDF شروع می‌شوند. بنابراین



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۳-۱۴۰۲
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل: ۱۴۰۲/۰۹/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

پنج بیت اول دستورات x87 همیشه 0x1B هستند. همانطور که قابل مشاهده است برای این حالت، فایل دیگری با آدرس `src/arch/x86/isa/decoder/x87.isa` به فایل حاضر `include` شده است که با مراجعه به آن شروع به دیکود سه بیت باقی مانده از بایت مربوط به opcode می‌کنیم. می‌توانید به جدول A-15 (صفحه ۴۴۳) در [فایل راهنما](#) برای بررسی دستورات مشخص شده توسط مقادیر مختلف سه بیت مذکور مراجعه کنید. به عنوان مثال، دستورات FSUB و FSUBR با رمزهای 0xDC و 0xD0 مشخص می‌شوند. برای تشخیص تفاوت عملکرد opcode‌های ارائه شده برای یک دستور مشابه، شما باید مفهوم فیلد ModRM مربوط به دستورات را بدانید (برای این کار می‌توانید به [فایل راهنما](#) مراجعه کنید). در فایل `x87.isa`، می‌توانید بررسی کنید که ما دستور FSUB را برای مقادیر 0x0 و 0x4 داریم. همچنین مشاهده می‌شود که پیاده‌سازی دستور FSUBR حذف شده است.

در گام اول، تفاوت بین دو پیاده‌سازی دستور FSUBR را بررسی کنید: یکی با بایت opcode برابر با D8h و دیگری برابر با DCh (توضیحات مربوط به دستور FSUBR را در [فایل راهنما](#) دستورات x87 مطالعه کنید). سپس، در سه مکان مختلف در فایل `x87.isa`، جملات مرتبط با دستور FSUBR را جستجو کنید و با جملاتی مشابه جملات مشخص شده برای دستور FSUB جایگزین کنید. با استفاده از عبارتی مانند `Inst::FSUB`، درخواست می‌کنید که از این دستور به جای دستور پیش‌فرض که فقط یک هشدار عدم پیاده‌سازی چاپ می‌کند، استفاده شود.

در گام دوم، باید پیاده‌سازی macro-op برای FSUBR به صورت micro-opها انجام شود. مجدداً می‌توان از پیاده‌سازی دستور FSUB الگوبرداری کرد. به دایرکتوری `src/arch/x86/isa/insts/x87/arithmetic/` مراجعه کنید. این دایرکتوری شامل تعریف دستورات مختلف حسابی x87 به صورت micro-opها است. نگاهی به نحوه پیاده‌سازی دستور FSUB با استفاده از micro-opها بندهاید، FSUB1 و FSUB2 به دو opcode مختلف اشاره دارند. برای هر نوع، باید سه پیاده‌سازی مختلف فراهم شود: یکی که فقط از ثبات‌ها استفاده می‌کند، یکی که یکی از عملوندها را با استفاده از آدرس موجود در دستور از حافظه می‌خواند و آخرین مورد که از آدرس



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۳-۱۴۰۲
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل: ۱۴۰۲/۰۹/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

اشاره گر دستور برای خواندن عملوند استفاده می کند. `micro-op` های مورد استفاده برای این سه پیاده سازی باید به سادگی قابل درک باشند. روشی که پارسر دستور در `gem5` کار می کند، باعث الزام تعریف هر سه پیاده سازی برای دستور `FSUBR` می شود. در کل، شما باید شش بلوک کد جداگانه برای `FSUBR` داشته باشید، مانند آنچه برای `FSUB` مشخص شده است.

سرانجام، باید یک پیاده سازی برای `micro-op subfp` ارائه شود. می توانید بررسی کنید که این پیاده سازی در حال حاضر در فایل `src/arch/x86/isa/micro-ops/fpop.isa` موجود است. بنابراین، برای این مرحله نیازی به انجام کاری نیست.

`gem5` را برای معماری `x86` کامپایل کنید تا اطمینان حاصل شود که در پیاده سازی هیچ اشتباهی رخ نداده است.

در نهایت لازم است پیاده سازی دستور `FSUBR` تست شود. برای این کار، یک برنامه `C` می نویسد که یک فایل با دو عدد اعشاری می خواند، آنها را از یکدیگر کم می کند و خروجی را چاپ می کند. به منظور اطمینان از استفاده از دستور `FSUBR` برای تفریق، از ویژگی `Inline assembly` کامپایلر `GCC` استفاده کرده و به طور مشخص از دستور `FSUBR` در کد استفاده می کنید.

مواردی که لازم است در فایل پاسخ تمرین موجود باشند:

- فایل `c` استفاده شده به منظور تست دستور `FSUBR`.
- فایل `patch` شامل تغییرات اعمال شده به `src/arch/x86/isa/insts/x87/arithmetic/` و `src/arch/x86/isa/decoder/x87.isa`.
- فایل `patch` را می توان با استفاده از دستور زیر ایجاد کرد:
`hg diff src/arch/x86/isa > /tmp/changes.patch.`
- گزارشی از مراحل انجام کار و نتایج حاصل از شبیه سازی.



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۳-۱۴۰۲
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل: ۱۴۰۲/۰۹/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

*دانشجویان گرامی در صورتی که ابهامی در توضیح گام‌های انجام تمرین وجود داشت لطفاً برای رفع ابهام و اطلاع بیشتر از جزئیات تمرین به آدرس زیر مراجعه کنید:

https://www.gem5.org/documentation/learning_gem5/gem5_101/homework-2

موفق باشید.