



پروژه نهایی

✓ در این پروژه ۲ مسئله تعریف شده است که دانشجویان به اختیار خود می‌توانند یک مورد را برای پیاده‌سازی انتخاب کنند.

✓ دقت کنید هر فرد مجاز به پیاده‌سازی تنها یک مورد است و در صورت پیاده‌سازی بیش از یک مورد نمره اضافه‌ای به فرد تعلق نخواهد گرفت.

✓ دانشجویان مجاز هستند مورد اول را با هر زبان برنامه‌نویسی که مایل هستند پیاده‌سازی کنند.

۱. JVM Emulator

در تمرین سوم درس نسخه‌ای کوچک از ماشین مجازی جاوا^۱ پیاده‌سازی کردید. پردازنده مذکور قادر به اجرای دسته‌ای از دستورات بود. شما بایستی در این قسمت شبیه‌سازی^۲ ارائه دهید که قادر باشد برنامه ساده‌ای که کاربر با استفاده از دستورات مذکور نوشته است را دریافت و به درستی اجرا کند و خروجی را در اختیار کاربر قرار دهد. شبیه‌ساز شما بایستی توانایی خواندن و اجرای تمامی دستورات ذکر شده در ISA تمرین سوم را داشته باشد. یک نمونه کامل از پردازنده طراحی شده در تمرین سوم برای استفاده دانشجویان بر روی سایت درس قرار خواهد گرفت اما هر فرد مجاز است از پردازنده خود نیز استفاده کند. علاوه بر این بایستی حین اجرای برنامه گزارشی از تغییراتی که در داخل پردازنده در جریان است به کاربر ارائه دهید. به عنوان مثال تغییراتی که در DataPath ایجاد می‌شود و مقادیر موجود در ثبات‌ها بایستی به نحوی به کاربر نشان داده شود. در این قسمت نشان دادن محتوا و تغییرات ثبات‌های PC، SP، ثبات آدرس حافظه اصلی (MAR)، ثبات نگه‌داری داده (MDR) و کلیه ثبات‌های موجود در پردازنده الزامی است. همچنین فعال شدن سیگنال‌های کنترلی مختلف درون پردازنده نیز بایستی به اطلاع کاربر برسد. کد ورودی شبیه‌ساز بایستی در قالب یک فایل دریافت شود. نمایش گزارش‌ها و تغییرات ثبات‌ها و سیگنال‌های کنترلی بایستی برخط^۳ باشد. اجرای کد بایستی خط به خط صورت گیرد و پس از اجرای هر خط از کد محتوای ثبات‌های ذکر شده و سیگنال‌های کنترلی که برای اجرای این دستور فعال شده اند به ازای هر کلاک نشان داده شود. به عنوان مثال اگر اجرای خطی از کد، ۴ کلاک به طول انجامد بایستی تمام مقادیر مورد نیاز در هر ۴ کلاک چاپ شوند. در صورتی که از رابط گرافیکی استفاده نمی‌کنید فرمت نمایش دادن نتایج بایستی بدین شکل باشد:

Register_name: register_value in decimal

Signal_name: true or false for this clock

موارد امتیازی:

^۱ JVM

^۲ emulator

^۳ online



- موارد مذکور به صورت گرافیکی برای کاربر نمایش داده شوند.
 - معیاری از **throughput** و **utilization** پردازنده در نهایت برای کاربر ارائه شود.
 - اضافه کردن حافظه نهان بین پردازنده و حافظه اصلی و استفاده از توابعی که در تمرین پنجم درس برای شبیه‌سازی حافظه نهان زده شد به عنوان واسط بین پردازنده و حافظه اصلی و ارائه آمار از نرخ **hit** و **miss**، هم‌چنین ارائه گزارشی از تغییرات در حال اعمال در حافظه نهان به کاربر.
- توجه:** حداکثر نمره امتیازی تعلق گرفته به این موارد ۵ درصد خواهد بود.

۲. وقفه و اعداد ممیز شناور

- با مطالعه بیشتر درباره ماشین مجازی جاوا و مکانیزم رسیدگی به وقفه‌ها در آن، مکانیزمی برای رسیدگی به وقفه در پردازنده‌ای که در تمرین سوم و چهارم درس طراحی کردید طراحی کنید. برای طراحی این مکانیزم از ماشین مجازی جاوا ایده بگیرید اما پیاده‌سازی ۳ لایه به جای ۱۵ لایه کفایت می‌کند. دقت کنید تغییرات ایجاد شده در پردازنده سیم‌کشی شده بایستی حداقلی باشد و تلاش بر این باشد که این قسمت با ریزبرنامه‌نویسی پیاده‌سازی شود. دستوراتی که در جدول زیر ذکر شده‌اند بایستی به **ISA** پردازنده قبلی اضافه شوند. علاوه بر آن‌چه که پیش‌تر ذکر شد، در این قسمت دانشجویان بایستی مکانیزمی برای انجام محاسبات ممیز شناور به پردازنده خود بیافزایند. برای این قسمت نیز دانشجویان مجاز هستند یک واحد ممیز شناور سخت افزاری به پردازنده قبلی اضافه نمایند و مابقی تغییرات را بایستی از طریق ریزبرنامه‌نویسی اعمال کنند. پیاده‌سازی عمل ضرب ممیز شناور اجباری است و برای این مورد یک **ALU** جدید در اختیار دانشجویان قرار خواهد گرفت.

Instruction	opcode	description
fadd	0x62	Adds 2 float numbers
fsub	0x66	Subtracts 2 float numbers
input	0x5f	Enables the input interrupt
fmul	0x6a	Multiplies 2 float numbers

- دقت کنید که اعداد ممیز شناور در قالب ۳۲ بیت نمایش داده خواهند شد. هر عدد ممیز شناور دارای ۴ بخش، است: ۱. mantissa، ۲. exponent، ۳. radix و ۴. sign که mantissa نشانگر جز بامعنی عدد است، exponent نشانگر توان مربوط به مبنای عدد است، radix مبنایی است که عدد در آن بیان شده و sign علامت مربوط به عدد است. دقت کنید که mantissa عددی همواره نامنفی است که در مبنای دو نمایش داده میشود و در نمایش ۳۲ بیتی عدد ممیز شناور، ۲۳ بیت کم‌ارزش را اشغال می‌کند. از طرفی exponent عددی ست که ۸ بیت بعدی را اشغال می‌کند و می‌تواند مثبت یا منفی باشد، این عدد را در

