تمرین ۳. طراحی و پیاده‌سازی یک پردازنده

- در این تمرین شما باید پردازنده‌ای با مشخصاتی که در ادامه می‌آید را طراحی و پیاده‌سازی کنید.
- برای پیاده‌سازی مسیرداده‌ی این ماشین یک ALU در اختیار شما قرار خواهد گرفت.
- این پردازنده، یک مدل ساده‌سازی‌شده از ماشین مجازی Java یا همان JVM است. این ماشین دارای ساختار Stack Machine است و تمام عملیات‌ها در این ساختار انجام می‌شود. به این صورت که، همه‌ی عملگرها ابتدا بر روی استک قرار می‌گیرند و سپس عملیات موردنظر برروی آن‌ها انجام شده و خروجی در بالای استک قرار داده می شود. برای مطالعه بیشتر در مورد ماشین‌های پشته‌ای می‌توانید از اینجا استفاده بکنید.
- یک ماژول حافظه در اختیارتان قرار خواهد گرفت که می‌توانید برای تست کردن پردازنده‌ی خود از آن استفاده کنید. برای تغییر دادن محتوای این حافظه کافی است به کد حافظه‌ی موردنظر نگاه کرده و تشخیص دهید که محتوای هر آدرس چگونه مشخص شده است. این ماژول حافظه پس از انجام عملیات خواندن و نوشتن سیگنال ready را تا ارسال درخواست بعدی فعال می‌کند و به طور کلی تأخیر آن مشخص نیست. این حافظه ۳ درگاه test دارد که بدون تاخیر محتویات ۳ آدرس حافظه را در اختیار شما قرار می‌دهد. این درگاه‌ها فقط برای تست پردازنده هستند و اجازه‌ی استفاده از آن‌ها را در پردازنده‌ی نهایی ندارید. پهنای هر کلمه در این حافظه ۳۲ بیت است و کوچکترین طول آدرس‌پذیر ۸ بیت می‌باشد.
- عملیات خواندن از حافظه با دو پهنای ۸ بیتی و ۳۲ بیتی امکان‌پذیر است. بدین صورت که عملیات خواندن دستورات برنامه بایست به صورت بایت به بایت از حافظه انجام شود ولی عملیات خواندن و نوشتن دیگر عملیات‌های پردازنده به صورت ۳۲ بیتی صورت بگیرد.
- دقت کنید ماژول حافظه‌ی شما جزء مسیر داده نیست و نباید در مسیر داده قرار بگیرد.
- برای سریع‌تر شدن فرآیند کامپایل فقط طرح‌های خود را سنتز کنید و نیازی به اجرای بقیه‌ی مراحل کامپایل نمی‌باشد. (به غیر از زمانی که می‌خواهید تست بگیرید)
- نوشتن RTL برای تمامی دستورات بخشی از نمره‌ی گزارش این تمرین است.
- برای ارزیابی این تمرین، تعدادی برنامه‌ی از پیش تعیین شده‌ی یکسان (benchmark) در ماژول حافظه‌ی شما قرار می‌گیرد که در آن‌ها از دستورات مختلف پردازنده استفاده شده است. به همین منظور در پیاده‌سازی پردازنده به نکات گفته شده دقت نمایید. دقت کنید اجرای صحیح تمامی دستورات پردازنده شامل نمره‌ی این تمرین می‌باشد.

دستورات واحد محاسبه و منطق

عملیات‌هایی که می‌توان با واحد محاسبه و منطق انجام داد به شرح زیر هستند:

F0	F1	ENA	ENB	INVA*	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\overline{A}$
1	0	1	1	0	0	$\overline{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	1	1	0	1	1	-A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	-1

*INVA: در صورت یک بودن مقدار ورودی A را منفی می‌کند.

دستورات پردازنده

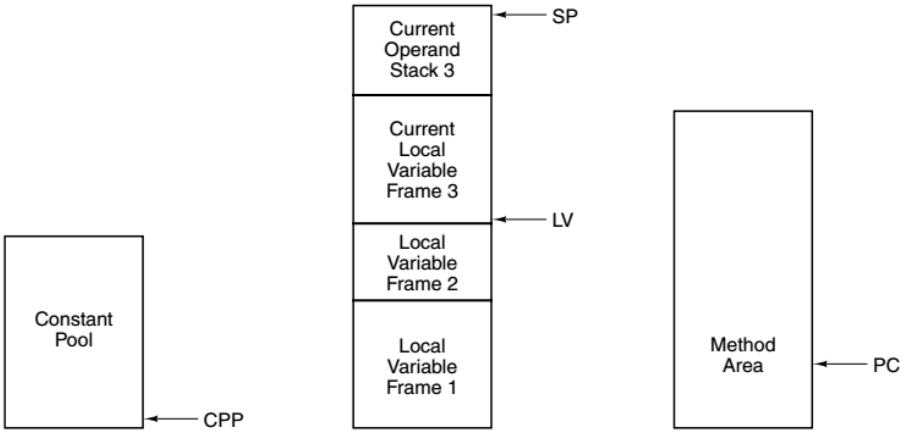
دستورهایی که بایست در این پردازنده پیاده سازی شود، در جدول زیر نشان داده شده‌اند. با توجه به توضیحات، تشخیص دستور و عملگر(های) آن به عهده‌ی خودتان می‌باشد.

HEX	Mnemonic	Meaning
0x10	BIPUSH byte	**Push byte onto stack**
0xA7	GOTO offset	**Unconditional branch**
0x60	IADD	**Pop two words from stack; push their sum**
0x99	IFEQ offset	**Pop word from stack and branch if it is zero**
0x9B	IFLT offset	**Pop word from stack and branch if it is less than zero**
0x9F	IF _ICMPEQ offset	**Pop two words from stack and branch if are equal**
0x84	IINC varnum const	**Add a constant to a local variable**
0x15	ILOAD varnum	**Push local variable onto stack**
0x36	ISTORE varnum	**Pop word from stack and store in local variable**
0x64	ISUB	**Pop two words from stack; push their difference**
0x00	NOP	**Do nothing**

نکات:

- مقادیر byte و const و varnum یک بایتی و offset دو بایتی هستند.
- هنگام انشعاب(branch) مقدار offset با مقدار کنونی PC جمع خواهد شد.

- مدل حافظه‌ی برنامه‌ها در معماری IJVM مانند شکل زیر است.



حافظه از ۴ قسمت مختلف تشکیل شده‌است که با استفاده از ۴ ثبات SP، LV، PC و PCC هر یک از این قسمت‌ها مشخص می‌شوند و با استفاده از این ۴ ثبات می‌توانید به قسمت‌های مختلف این نواحی دسترسی پیدا کنید. ثبات PC مشخص‌کننده‌ی قسمتی است که کد مورد نظر در آن قرار گرفته است. ثبات SP به بالای استک این معماری اشاره می‌کند. ثبات LV به ناحیه‌ای که برای ذخیره‌کردن متغیرهای محلی درنظر گرفته شده است اشاره می‌کند و ثبات CPP به ناحیه‌ای که مقادیر ثابت در آن ذخیره شده‌اند اشاره می‌کند.

- پیاده سازی ۴ ثبات اشاره شده در مسیرداده‌ی پردازنده‌ی شما ضروری است.
- می‌توانید برای ساده‌سازی فرض کنید ثبات‌های گفته‌شده همیشه دارای آدرس ثابتی هستند. مثلاً فرض کنید پشته در انتهای حافظه قرار دارد. لازم است که این فرض‌ها را در گزارش ذکر کنید.
- در طراحی مسیرداده از هر تعداد ثباتی که نیاز دارید بدون هیچ محدودیتی می‌توانید استفاده کنید.
- همه‌ی ثبات‌ها باید یک درگاه Reset آسنکرون داشته باشند که به ورودی Reset پردازنده وصل بوده و با یک شدن آن مقدار آن‌ها صفر شود.
- در نهایت گزارشی از روند کار و فرض‌های خود و RTL مربوط به دستورات را در قالب pdf و فایل های vwf. ، bsf. و bdf. را به صورت zip فشرده کرده و به شکل CA_Assignment3_StudentID نام‌گذاری و ارسال نمایید.

زمان تحویل

(به Courseware نگاه کنید.)

یادآوری‌های عمومی

لطفاً توجه داشته باشید که:

۱. به ازای هر روز دیرکرد در تحویل تمرین‌ها ۱۰٪ جریمه منظور خواهد شد.
۲. به‌هیچ عنوان تمرینی را از دیگران کپی نکنید و به دیگران کپی ندهید. درغیراین‌صورت نمره‌ی آن تمرین برای هر دو طرف (کپی‌دهنده و کپی‌گیرنده) ۱۰۰- منظور خواهد شد.

موفق باشید

سلیمانی و حق‌پناه