

معماری کامپیوتر (۴۰۳۲۳) – نیم سال دوم ۹۸-۹۹

Computer Architecture

دانشکده مهندسی کامپیوتر

مدرس: دکتر اسدی



پروژه پایانی

- گزارش پروژه خود را به صورت تایپ شده و در فرمت pdf در قسمت مربوطه در سامانه CW بارگذاری نمایید.
- نام فایل ارسالی خود را حتما به شکل proj-studentid.pdf قرار دهید.
- پرسش های خود را در تالارهای ایجاد شده در صفحه CW درس مطرح کنید.

پروژه پایانی

پروژه نهایی دارای دو بخش است:

بخش اول - طراحی پردازنده پایه به کمک Quartus

در این بخش، شما باید با استفاده از نرم‌افزار Quartus، یک پردازنده طراحی کنید. شما باید با استفاده از ماژول‌های طراحی شده در تمرین‌های ۶ و ۷ (بانک ثبات و PC) و افزودن ماژول‌های ALU و دیکودر، یک پردازنده ساده را شبیه‌سازی نمایید. طراحی شما بایستی به صورت Multi-Cycle باشد. تعداد کلاک‌ها وابسته به نوع طراحی شماست (در گزارش خود تعداد سیکل‌هایی که در طراحی این پردازنده در نظر گرفته‌اید ذکر کنید).

پردازنده بایستی دستورات زیر را پشتیبانی کند:

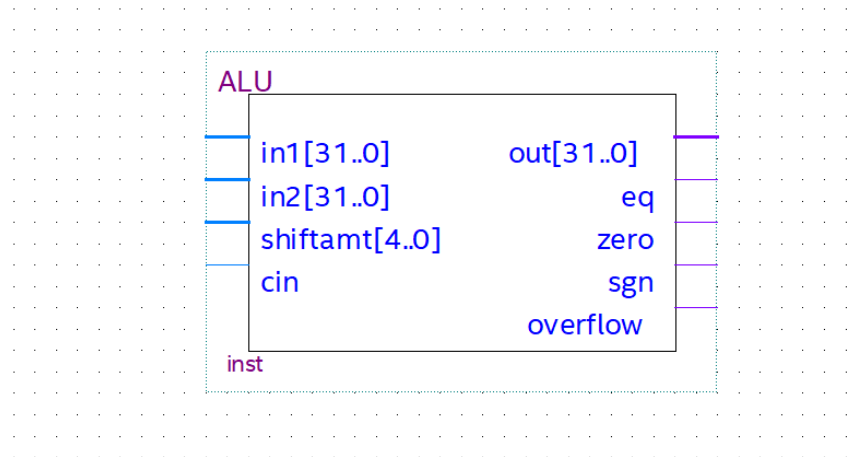
Opcode	Instruction	Explanation
0000	$out = in1 + in2$	جمع دو ورودی
0001	$out = in1 - in2$	تفریق دو ورودی
0010	$out = in1 \gg shiftamt$	شیفت به راست ورودی اول به اندازه‌ی $shiftamt$
0011	$out = in1 \ll shiftamt$	شیفت به چپ ورودی اول به اندازه‌ی $shiftamt$
0101	$out = in1 \sim in2$	$NAND$ دو ورودی
0110	$out = \min(in1, in2)$	پیدا کردن عدد کوچک‌تر بین دو ورودی
0111	$out = in1 < in2 ? 1:0$	دستور $set\ on\ less\ than$
1000	Load value to reg bank	ذخیره مقدار $Shamt$ در بانک ثبات

ورودی‌های این ALU به صورت زیر هستند:

Input	Name[bits]
ورودی اول	in1[0..31]
ورودی دوم	in2[0..31]
مقداری که باید به اندازه‌ی آن شیفت بدهیم	shiftamt[0..4]
مقدار $opcode$ دستور را مشخص می‌کند	opcode[0..2]
مقدار cin برای عملیات جمع	Cin

خروجی‌های این ALU به صورت زیر هستند:

Output	Name[bits]
حاصل عملیات	out[0..31]
پرچم برابری	Eq
پرچم صفر	Zero
پرچم علامت	Sgn
پرچم سرریز شدن	Overflow



در طراحی ALU به نکات زیر توجه داشته باشید:

- در صورت نیاز شما می‌توانید از یکی از Opcode های رزرو به عنوان دستور No-op استفاده کنید (در صورت استفاده در گزارش خود ذکر کنید).
- در هنگام انجام عملیات shift right، بیت‌های خالی شده را صفر قرار دهید.
- عملیات NAND به صورت bitwise انجام می‌شود.
- در عملیات amin، باید از بین دو ورودی کوچک‌ترین آن‌ها برگردانده شود و در صورتی که دو ورودی با هم برابر باشند ورودی اول بازگردانده شود.
- دستور set on less than، مشابه دستور set on less than در mips است به طوریکه اگر ورودی اول از ورودی دوم کوچک‌تر باشد مقدار خروجی یک و در غیر این صورت مقدار خروجی صفر می‌شود.
- از ورودی cin، تنها در عملیات جمع به عنوان cin استفاده می‌شود.
- پرچم eq، در صورتی که ورودی‌ها ALU با هم برابر باشند مقدار آن یک می‌شود و غیر این صورت مقدار آن صفر است.
- پرچم zero، در صورتی که حاصل عملیات ALU صفر باشد مقدار آن یک می‌شود و در غیر این صورت مقدار آن صفر است.
- پرچم sgn، در صورتی که حاصل عملیات ALU منفی باشد مقدار آن یک می‌شود و در غیر این صورت مقدار آن صفر است.
- پرچم overflow فقط برای عملیات جمع استفاده می‌شود و در صورتی که overflow اتفاق افتاده باشد مقدار آن یک می‌شود در غیر این صورت مقدار آن صفر است.

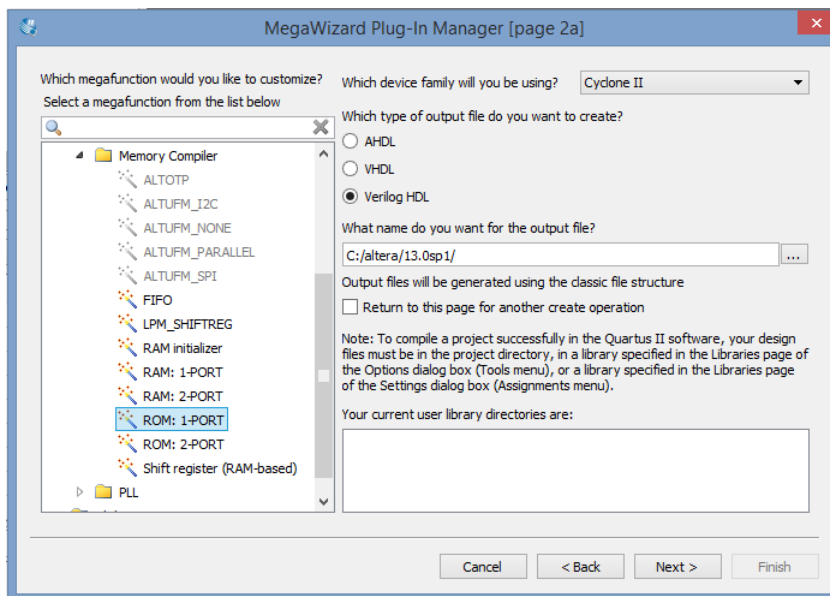
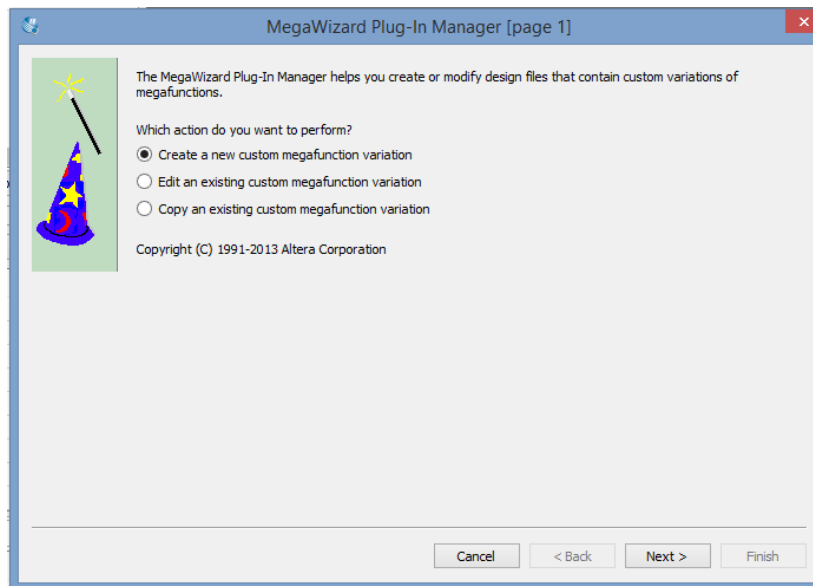
این پردازنده از دستورات ۲۰ بیتی استفاده می‌کند. ساختار دستورات محاسباتی/منطقی دستور به شکل زیر است:

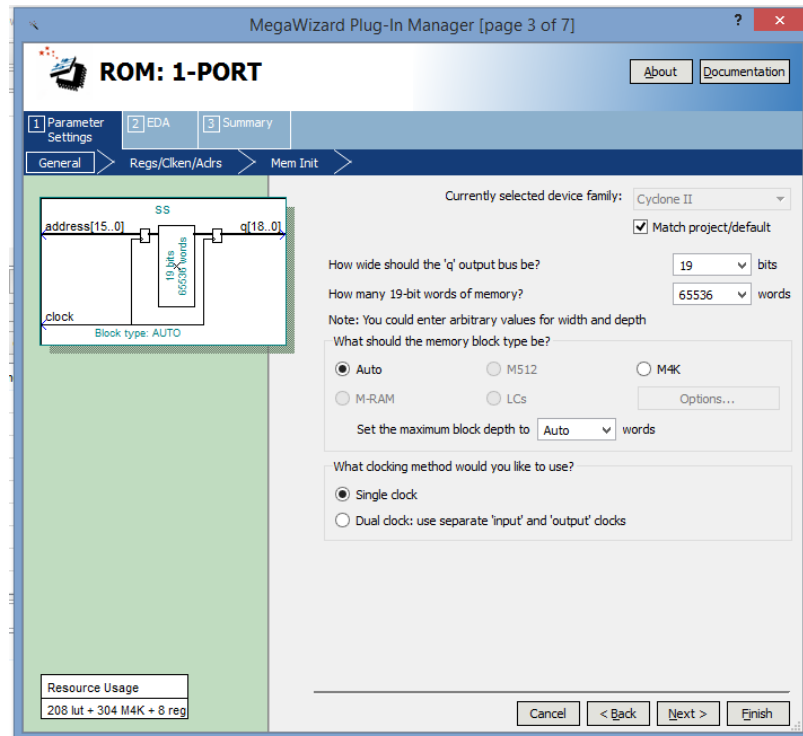
19	16	15	14	10	9	5	4	0
Opcode (3-bit)	cin (1-bit)	Operand 1 Address (5-bit)			Operand 2 Address or Shamt (5-bit)		Output Address (5-bit)	

تمامی عملوندهای این پردازنده، از نوع ثبات هستند. در دستورات دو عملوندی (Opcode های ۲ و ۳)، مقدار shamt بر اساس بیت های ۵ تا ۹ تعیین می شود. در دستور بارگذاری، بیت های ۵ تا ۱۴ در بانک ثبات قرار می گیرند:

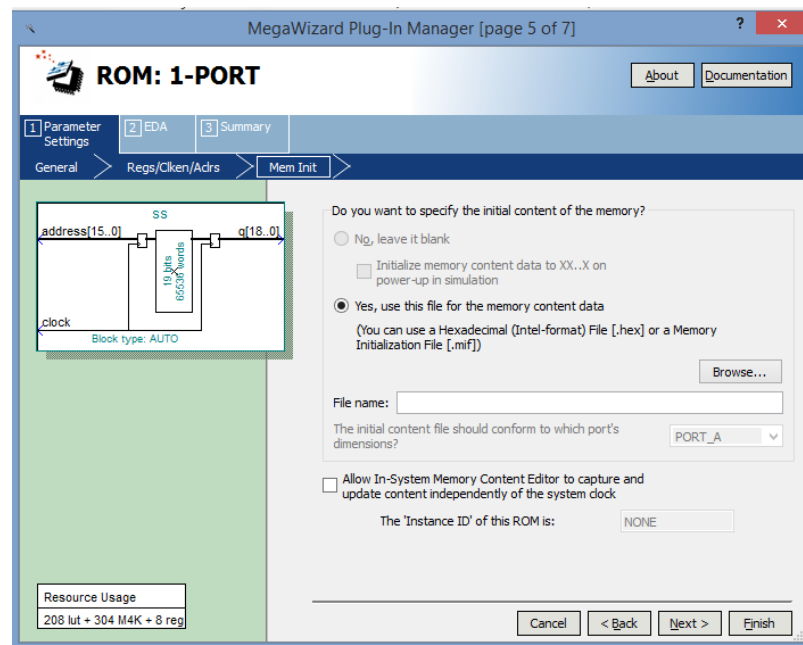
$\text{Reg}[\text{IR}[4:0]] = \text{IR}[14:5]$

۲۲ بیت دیگر ثبات را برابر با صفر قرار دهید. به منظور طراحی حافظه دستور، می توانید از منوی Tools و انتخاب MegaWizard Plug-in Manager، یک حافظه RAM یا ROM ایجاد کنید.





[با توجه به ۳۲ بیتی بودن PC، می‌توانید از دو ROM استفاده کنید]



[به منظور مقداردهی اولیه ROM، نیاز به یک فایل hex یا mif دارید. جهت ایجاد یک فایل mif می‌توانید از این لینک

[استفاده نمایید: Memory Initialization File \(.mif\)](#)

در انتها حتما تیک مربوط به تولید فایل bsf را فعال کنید. استفاده از روش‌های دیگر جهت ایجاد حافظه دستور نیز مجاز است.

* با توجه به اینکه اعداد ورودی ۳۲ بیتی هستند برای راحت تر شدن انجام این تمرین پیشنهاد می شود از سبیل سازی ابزار Quartus استفاده کنید. همچنین استفاده از انواع lpm برای طراحی این پردازنده بلامانع است.

** برای آزمون این پردازنده، طراحی و ساخت یک waveform، ارزیابی تمامی دستور عمل های ALU و چک کردن حالاتی که منجر به تغییر پرچم های خروجی می شوند، الزامی است.

بخش دوم – ارزیابی اثر پارامترهای مختلف حافظه و دستورات خاص منظوره بر کارایی برنامه ها با استفاده از شبیه ساز gem5

هدف از این پروژه، بهبود کارایی به وسیله افزودن دستورات خاص منظوره به پردازنده و همچنین ارزیابی اثر ساختارهای مختلف حافظه ای می باشد. به این منظور، الگوریتم SHA-256 مدنظر قرار می گیرد. مجموعه SHA-2 مجموعه ای از توابع درهم سازی در رمزنگاری محسوب می شود SHA-256. به عنوان یکی از معروف ترین شاخه های آن محسوب می شود. این پروژه شامل چهار گام زیر است:

- در ابتدا، تحلیل خود را از دستوراتی که اضافه شدن آنها به ISA می تواند کارایی این برنامه را بهبود دهد، ارائه دهید و یک دستور را جهت اضافه شدن به ISA انتخاب کنید. دستور انتخابی شما نباید جزو دستورات رایج ISA باشد.
- دستور مدنظر را به ISA اضافه کنید. شما باید سه مقدار تاخیر متفاوت را برای دستور اضافه شده مدنظر قرار بدهید و نشان دهید اضافه شدن دستور اختصاصی، نسبت به ISA اصلی، چه تاثیری بر روی کارایی برنامه گذاشته است.
- در گام بعدی، با طراحی یک فایل تنظیمات (Config)، تاثیر افزودن حافظه نهان دو سطحی بر اجرای برنامه در حالت های عادی و تغییر یافته (افزودن دستور خاص منظوره و استفاده از یکی از مقادیر تاخیر) را در حضور حافظه نهان، تحلیل کنید. سطح اول این حافظه نهان دارای دو حافظه جداگانه برای دستور و داده به اندازه های ۳۲ و ۶۴ کیلوبایت می باشد. حافظه سطح دوم از یک حافظه برای دستورات و داده استفاده کرده و اندازه ای آن ۱ مگابایت است. فرکانس کاری پردازنده، نوع پردازنده و حافظه اصلی مورد استفاده را نیز مطابق جدول زیر تعیین نمایید:

CPU	Frequency	Memory
TimingSimpleCPU	1	DDR3_1600_8x8

- برای سنجیدن عملکرد سیستم با تنظیمات مختلف برای حافظه نهان بهتر است پارامترهایی مانند 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، مقادیر ۳۲ و ۶۴ را برای cacheline در نظر بگیرید و تاثیر هر مقدار بر کارایی برنامه را گزارش کنید. سپس برای associativity حافظه نهان سطح دوم مقادیر ۸ و ۱۶ را فرض کرده و نتایج آزمایش را به همراه تحلیل خود را گزارش کنید (مجموعاً ۴ شبیه سازی برای این گام لازم است).

* علاوه بر گزارش کار، شما بایستی فایل های زیر را نیز ارسال نمایید:

- فایل `cpp` مربوط به الگوریتم SHA-256.
 - اسکریپت تنظیمات.
 - فایل `stats.txt` تولید شده برای هر کدام از اجراها با تنظیمات مختلف با اسم گذاری مناسب.
- **** با توجه به این که هدف پروژه، مسلط شدن شما بر روی شبیه ساز `gem5` است، می توانید از پیاده سازی های آماده الگوریتم SHA-256 نیز استفاده کنید.

******* با توجه به مشخص نبودن امکان تحویل حضوری پروژه، لطفا گزارشی در مورد تمرین عملی به همراه فایل های طراحی، ارسال نمایید. گزارش شما باید شامل توضیح کامل رویکرد و ساختار طراحی و توضیح نتایج باشد. لطفا گزارش خود را در ادامه پاسخ به سوالات تئوری بنویسید و تنها یک فایل `pdf` ارسال نمایید.