



تمرین سری سوم

- تمرین‌های خود را در قالب فایل تایپ‌شده و ذخیره شده به فرمت *PDF* به صورت یک فایل *Zip* با نام *1STDID_HW* که *STDID* شماره‌ی دانشجویی شما است، در صفحه‌ی درس در *CourseWare (CW)* بارگذاری کنید.
- دانشجویان مجاز هستند دو تمرین را حداکثر با دو روز تاخیر ارسال کنند.
- سؤالات خود را صرفاً در فروم مربوطه در *CW* بپرسید.



تمرین سری سوم

1- فرض کنید بیت صفر (کم ارزش) در ورودی *Write Register* از بانک ثبات همیشه دچار خطای "Stuck at one" شده‌اند. توضیح دهید آیا هنوز می‌توان از پردازنده استفاده کرد (و چگونه). توجه شود که پردازنده در صورتی قابل استفاده است که بتوان دستور دچار خطا را با دستور یا مجموعه‌ای از دستور یا مجموعه‌ای از دستورهای دیگر جایگزین نمود.

2- در جدول زیر تاخیر واحدهای منطقی مسیر داده (*datapath*) داده شده است. چنانچه بتوان 10 درصد از تاخیر یکی از بلوک‌های مسیر داده (*datapath*) را کاهش داد، بیشترین تسریع (*Speed-up*) چقدر است؟

| I-Mem | Add | Mux | ALU | Regs | D-Mem | Sign-Extend | Shift Left 2 | ALU Ctrl |
|-------|------|------|------|------|-------|-------------|--------------|----------|
| 200ps | 70ps | 20ps | 90ps | 90ps | 250ps | 15ps | 10ps | 30ps |

3- مسیر داده (*datapath*) پردازنده *MIPS* را در نظر بگیرید.

الف – برای یک دستور *beq* مقادیر سیگنال‌های خواسته‌شده را مشخص کنید.

| <i>RegWrite</i> | <i>RegDst</i> | <i>Branch</i> | <i>MemRead</i> | <i>MemtoReg</i> | <i>MemWrite</i> | <i>ALUSrc</i> |
|-----------------|---------------|---------------|----------------|-----------------|-----------------|---------------|
| | | 1 | | | | |

ب – فرض کنید یک دستور از نوع *R-type* داریم. می‌دانیم که شماره ثبات مقصد در بیت‌های 11 تا 15 است. با توجه به این نکات مقادیر سیگنال‌های خواسته‌شده را برای این دستور تعیین کنید.

| <i>RegWrite</i> | <i>RegDst</i> | <i>Branch</i> | <i>MemRead</i> | <i>MemtoReg</i> | <i>MemWrite</i> | <i>ALUSrc</i> |
|-----------------|---------------|---------------|----------------|-----------------|-----------------|---------------|
| 1 | | 0 | 0 | 0 | 0 | 1 |

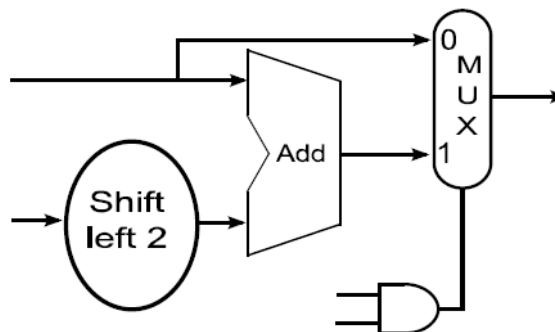
4- در پردازنده *MIPS* می‌خواهیم دستور *bne* را در جمع دستورات پشتیبانی‌شده اضافه کنیم. بخش نشان‌داده‌شده در مسیر داده (*datapath*) باید به گونه‌ای اصلاح شود که این مهم تحقق یابد. هم‌چنین سیگنال کنترلی *Branch* دیگر کافی نخواهد بود. به جای آن باید دو سیگنال کنترلی داشته باشیم.

bne \$rs, \$rt, immediate *#if valueOf(rs) != valueOf(rt) then PC = PC + 4 + immediate << 2*

با اعمال تغییرات گفته‌شده به جزئیات بگوئید چه تغییراتی در مسیر داده (*datapath*) ایجاد می‌شود، سیگنال‌های کنترلی جدید چه هستند، مقدار آن‌ها چگونه تعیین می‌شود، سیگنال کنترل‌کننده *MUX* موجود در شکل چه خواهد بود؟ کاملاً پاسخ خود را شرح کنید و هر آنچه که باید را تعیین کنید.



تمرین سری سوم



شکل مربوط به سوال 4

5- فرض کنید که می‌خواهیم دستور *I-type* جدید $getpc \$rt$ را به *ISA* پردازنده *MIPS* اضافه کنیم. این دستور مقدار *pc* را درون ثبات $\$rt$ قرار می‌دهد. تغییرات لازم برای مسیر داده (*datapath*) جدید که بتواند این دستور را پشتیبانی کند را نمایش دهید. سیگنال‌های کنترلی‌ای که این دستور چه مقادیری به خود می‌گیرند؟

6- دستور $add \$20, \$9, \$10$ در یک پردازنده *single cycle MIPS* در حال اجرا شدن می‌باشد در صورتی که در ثبات‌های $\$9$ و $\$10$ به ترتیب مقادیر یک و دو ذخیره شده باشند و مقدار *pc* برابر با $0x0012$ باشد. مسیر حرکت این دستور در مسیر داده (*datapath*) را با نشان دادن مقادیر سیم‌ها و گذرگاه‌های پردازنده نمایش دهید.

سوال عملی - یک واحد *ALU* (محاسبه و منطق) ترکیبی با عملکرد ذیل را طراحی کنید.

| OpCode | Instruction |
|--------|-----------------------------------|
| 000 | $OUT = DATA1 + DATA2$ |
| 001 | $OUT = DATA1 - DATA2$ |
| 010 | $OUT = DATA1 \text{ NAND } DATA2$ |
| 011 | $OUT = DATA1$ |
| 100 | $OUT = DATA1 \ll SHAMT$ |
| 101 | $OUT = DATA1 \gg SHAMT$ |
| 110 | Check Equal |
| 111 | Find the Greater |

در این واحد، در صورت انتخاب دستورات با *OpCode* برابر با 100 و یا 101 مقدار *Data1* به اندازه ورودی *SHAMT* (*Shift Amount*) به سمت چپ شیفت داده می‌شود.



تمرین سری سوم

در دستور *Check Equal* در صورتی که دو مقدار *Data1* و *Data2* با هم مساوی باشند خروجی *E* برابر 1 می شود و مقدار یکی از آن ها با خروجی منتقل می شود. در غیر این صورت خروجی *E* برابر 0 خواهد بود و مقدار خروجی تغییری نمی کند.

در دستور *Find the Greater* دو مقدار *Data1* و *Data2* با هم مقایسه می شوند هر کدام که بزرگ تر باشد بر روی خروجی قرار می گیرد. در صورت تساوی مقدار یکی از آن ها بر روی مساوی قرار می گیرد.

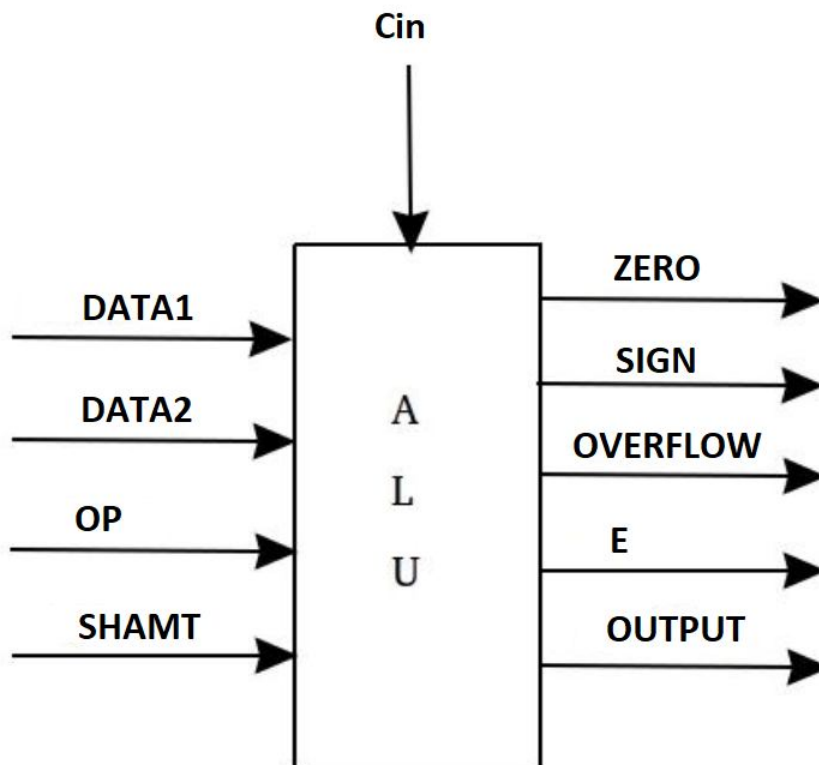
خروجی *Sign* در صورتی که خروجی منفی باشد مقدار 1 می گیرد در غیر این صورت مقدار 0 خواهد داشت.

خروجی *Zero* در صورتی که خروجی برابر صفر باشد 1 خواهد بود و در غیر این صورت مقدار 0 خواهد داشت.

در هنگام اجرای عملیات جمع یا تفریق اگر *overflow* رخ دهد سیگنال خروجی آن 1 می شود در غیر این صورت این سیگنال مقدار 0 خواهد داشت.

ورودی *Cin* برای دادن رقم نقلی ورودی به کار می رود.

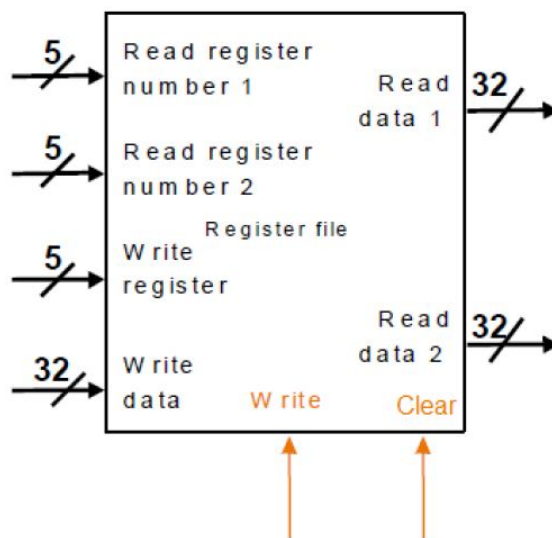
توجه: در این واحد محاسبه و منطق کلیه عملیات مطابق سیستم مکمل دو صورت می گیرد.





تمرین سری سوم

سوال عملی - یک بانک ثبات 32 بیتی با 32 عدد ثبات مطابق شکل زیر و آنچه که در معماری MIPS به کار می‌رود را پیاده‌سازی نمایید. این بانک ثبات توانایی خواندن دو عدد ثبات و نوشتن در یک ثبات را به طور همزمان داراست. افزون بر قابلیت‌های بانک ثبات MIPS، این بانک ثبات قابلیت پاکسازی سنکرون تمامی ثبات‌ها را با یک شدن سیگنال Clear دارا است. در صورتی که سیگنال Write و Clear همزمان فعال باشند، اولویت با سیگنال Clear است.



طراحی شما باید مطابق شکل زیر باشد و هیچ کدام از اسامی تغییر نیابد. برای این بانک ثبات، یک *waveform* طراحی کرده و با انجام تعدادی نوشتن و خواندن، صحت عملکرد را بررسی نمایید.

