



دانشگاه تهران
دانشکده‌ی برق و کامپیوتر
معماری کامپیوتر، بهار ۹۲
تمرین کامپیوتری ۲
پیاده سازی پردازنده تک مرحله‌ای

تاریخ تحویل: ۹۲/۱/۲۰

تاریخ بارگذاری: ۹۱/۱۲/۱۵

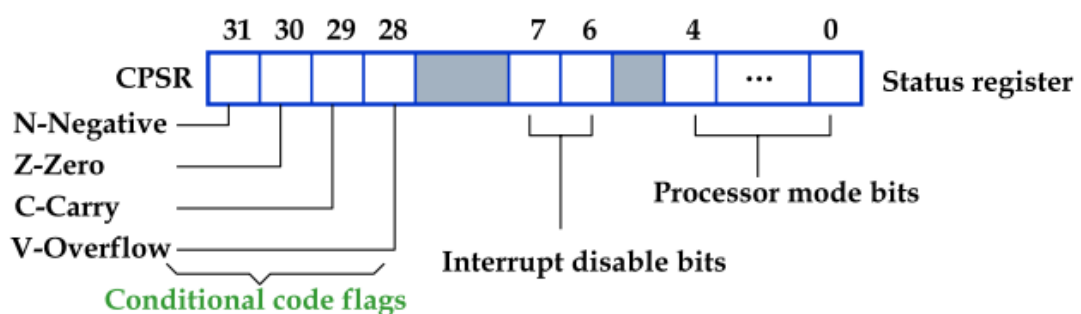
مقدمه

در این تمرین کامپیوتری باید پردازنده‌ی UTS92 که معماری شبیه پردازنده‌ی ARM دارد را پیاده‌سازی کنید. در این پردازنده همه‌ی **دستورات به صورت شرطی** اجرا می‌شوند و در صورتی که شرط مشخص شده در دستور با حالت کنونی پردازنده یکسان بود دستور اجرا می‌شود و در غیر این صورت دستور اجرا نمی‌شود. این قابلیت باعث بالا بردن چگالی کد^۱، کم کردن تعداد دستورات یک برنامه و در نتیجه بهره‌وری^۲ بیشتر پردازنده می‌شود.

در زیر مشخصات UTS92 توضیح داده شده است.

Registers:

- PC^۳ که مشخص می‌کند UTS92 در کجای حافظه‌ی دستورات قرار دارد.
- ۱۶ رجیستر ۳۲ بیتی همه منظوره^۴ (R0 همیشه صفر است)
- رجیستر ۳۲ بیتی مربوط به حالت^۵ پردازنده (شکل ۱) (در این تمرین تنها نیاز به بیت‌های ۲۸ تا ۳۱ داریم). که با استفاده از خروجی واحد پردازشی مقدار می‌گیرد.



شکل ۱: رجیستر حالت

^۱ Code Density

^۲ Performance

^۳ Program Counter

^۴ General Purpose Registers(GPR)

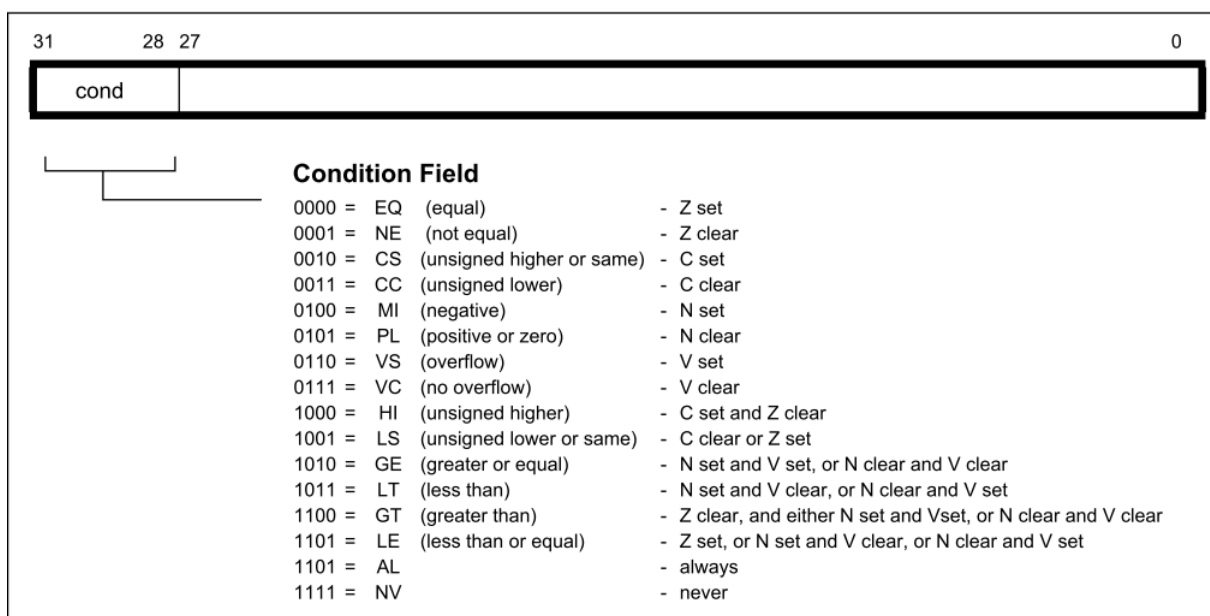
^۵ Statues Register

Hard Resetting:

UTS92 ورودی ریست دارد که اگر به اندازه‌ی یک سیکل ساعت^۱ باشد محتوای PC و CPSR صفر می‌شود. در هنگام ریست باید حافظه‌های پردازنده را با محتوای دو فایل "data.dat" و "inst.dat" با استفاده از دستورات ورودی خروجی وریلاگ^۶ پر کنید^۷ (\$readmemb).

Instruction set:

همان طور که در مقدمه گفته شد، تمام دستورات UTS92 به صورت شرطی اجرا می‌شوند و بر اساس اینکه پرچم^۸ های Z، N، C و V چه مقداری دارند پردازنده تصمیم می‌گیرد که دستور جاری اجرا شود یا خیر. در این تمرین فرض می‌شود (با توجه به شکل ۲) Condition Field تنها "0000"، "1100"، و "1101" می‌باشد. (AL, GT, EQ).



شکل 2: Condition Field در فریم دستورات UTS92

UTS92 سه نوع دستور را پشتیبانی می‌کند که عبارت‌اند از: دستورات پردازش داده^۹، انتقال داده^{۱۰} و پرش. در ادامه فرمت این دستورات و نحوه‌ی اجرای آنها توضیح داده شده است.

۱- دستورات پردازش داده

دستورات پردازش داده مربوط به UTS92 را در شکل (۳) ملاحظه می‌کنید. شما باید تمام دستوراتی که در شکل (۳) آمده است را با همان Op-code ها پیاده‌سازی کنید.

⁶ Verilog IO tasks

⁷ \$readmemb("<file_name>", <memory_name>")

⁸ Flag

⁹ Data processing instruction

¹⁰ Data transfer instruction

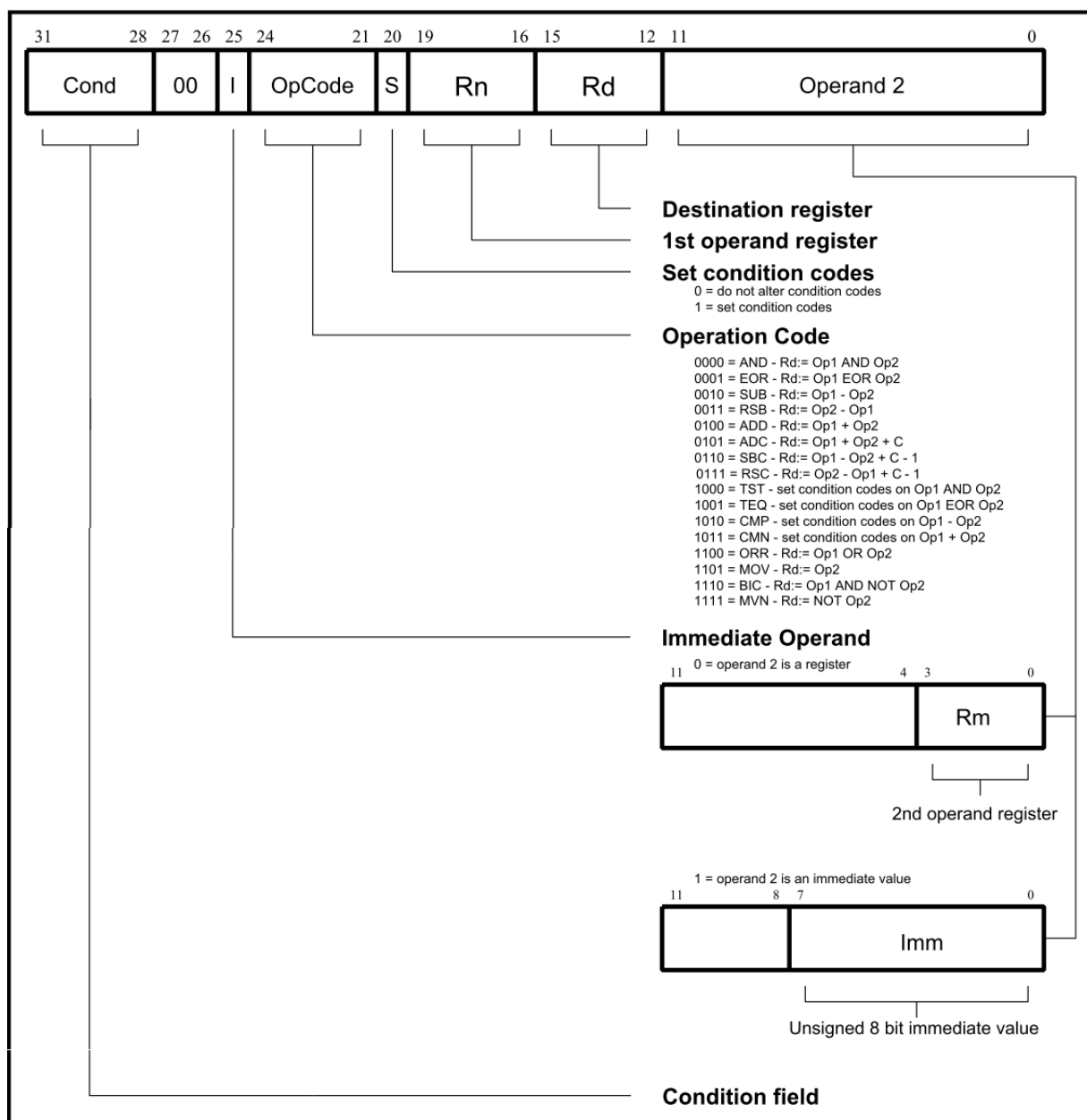
در صورتی که بیت ۲۰م در فریم مربوط به دستورات پردازش داده ۱ باشد با اجرای این دستورات رجیستر حالت (شکل ۱) تغییر می‌کند. ولی در صورتی که این بیت ۰ باشد با اجرای آن دستور رجیستر حالت تغییر نمی‌کند. در زیر نحوه‌ی تغییر رجیستر حالت با اجرای دستورات پردازش داده توضیح داده شده است:

پرچم Z: هنگامی ۱ می‌شود که خروجی ALU صفر شود.

پرچم N: هنگامی ۱ می‌شود که نتیجه‌ی دستورات پردازش داده منفی باشد.

پرچم V: این پرچم تنها با دستورات ADD, ADC, SUB, RSB تغییر می‌کند و در صورتی ۱ می‌شود که نتیجه‌ی عملیات دچار overflow شود.

پرچم C: این پرچم با خروجی سرریز^{۱۱} ALU تغییر می‌کند.



شکل 3: فرمت دستورات پردازش داده در UT92

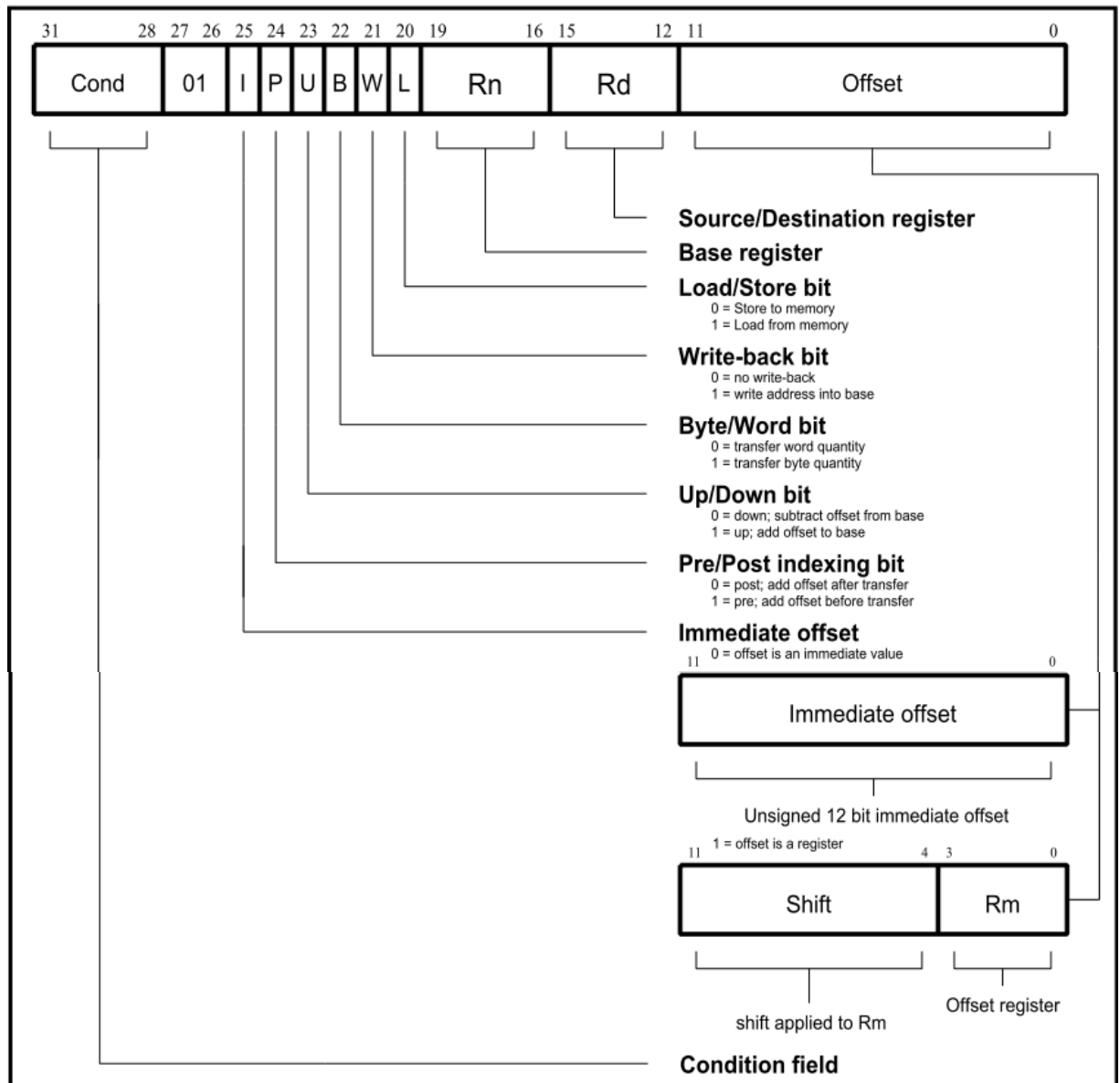
¹¹ Carry

۲- دستورات انتقال داده

در شکل (۴) فرمت دستورات انتقال داده^{۱۲} نشان داده شده است. این دستورات برای خواندن/از نوشتن/در حافظه استفاده می‌شوند. آدرس حافظه‌ی مورد نظر با اضافه کردن مقداری به یک رجیستر پایه^{۱۳} محاسبه می‌شود. در این تمرین فرض کنید بیت‌های ۲۱، ۲۲، ۲۳، ۲۴ و ۲۵ که مربوط به W, B, U, P, I در فریم شکل (۴) می‌باشد همیشه دارای مقادیر زیر هستند:

$W=0', B=0', U=1', P=1', I=0'$

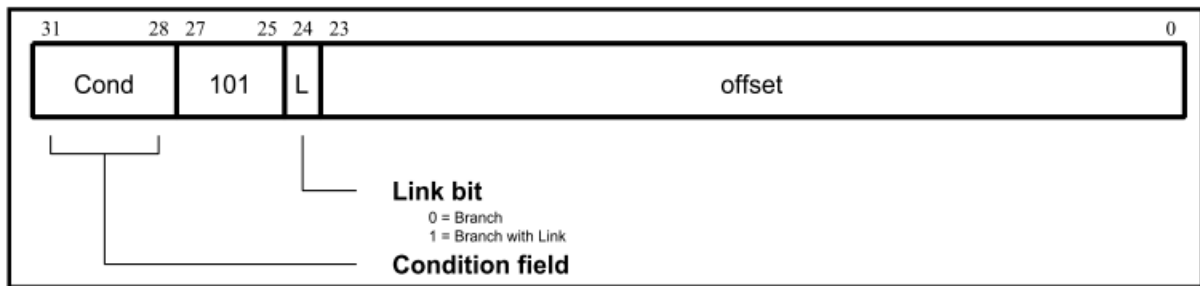
با توجه به اینکه $P=1'$ ، پس نحوه‌ی محاسبه‌ی آدرس Pre Index mode می‌باشد؛ به این معنی که آدرس برابر با مجموع offset و رجیستر پایه می‌باشد.



شکل 4: فرمت دستورات انتقال داده در UT92

¹² Data transfer instruction

¹³ Base register



شکل ۵: فرمت دستورات پرش در UTS92

دستورات پرش دارای یک آفست علامت‌دار ۲۴ بیتی می‌باشند. این آفست در هنگام اجرای دستور پرش به اندازه‌ی دو بیت به سمت چپ شیفت می‌خورد، sign extend می‌شود و با PC جمع می‌گردد. دستور Branch with Link مقدار PC پیشین را در R14 ذخیره می‌کند.

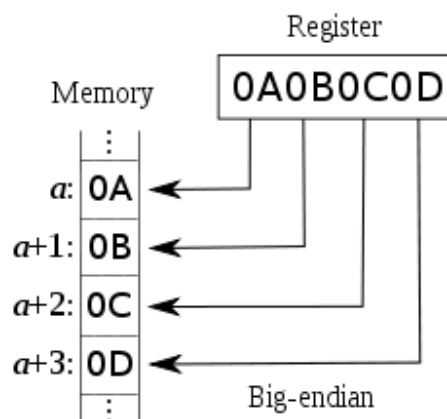
Memory Structure:

UTS92 دارای دو حافظه‌ی مجزا برای داده و دستور می‌باشد.

Instructions memory: 1K words instruction memory

Data memory: 8K words main memory

حافظه‌ها در UTS92 به صورت کلمه‌ای آدرس‌پذیر^{۱۴} هستند (یک کلمه^{۱۵} برابر ۳۲ بیت می‌باشد). همان‌طور که قبلاً گفته شد هر دو حافظه باید در reset با استفاده از فایل پرشوند. UTS92 تنها به صورت big-endian می‌تواند آدرس‌پذیر باشد (شکل ۶). برای اطلاعات بیشتر مربوط به endianness به <http://en.wikipedia.org/wiki/Endianness> مراجعه کنید.



شکل ۶: حالت آدرس‌دهی big-endian

¹⁴ Word addressable

¹⁵ Word

Testing:

نمره‌ی نهایی شما بر اساس این قسمت محاسبه می‌شود. بعد از پیاده‌سازی UTS92، شما باید آن را **درستی‌یابی**^{۱۶} کنید. برنامه‌ای بنویسید که مقادیر داخل حافظه‌ی داده را مرتب می‌کند (شکل ۷) و پردازنده را با آن تست کنید. دقت کنید که نوشتن این برنامه نمره‌ی جداگانه دارد. دستورات دیگر که در برنامه‌ی مرتب سازی استفاده نشده‌اند را نیز با نوشتن تست‌بنچ مناسب درستی‌یابی کنید. تمام تست‌بنچ‌های خود به همراه توضیح مختصری که هر خط تست چه کاری انجام می‌دهد را همراه دیگر فایل‌ها آپلود کنید.

```
1 Sort:
2 If(Mem[0]<0) then
3     Sort Mem[2] to Mem[Mem[1]] in descending order.
4 Else if(Mem >0) then
5     Sort Mem[2] to Mem[Mem[1]] in ascending order.
6 Else
7     Do nothing
```

شکل 7: شبه کد برنامه مرتب سازی

Notes:

- لطفا سوالات خود را در فروم درس بپرسید.
- این تمرین را می‌توانید در قالب گروه‌های حداکثر ۲ نفری انجام دهید.
- در هنگام تحویل حضوری مسیر داده و کنترلر خود را بر روی کاغذ همراه بیاورید. همچنین سعی کنید بر روی طرح خود مسلط باشید تا زحمتهایی که کشیده‌اید به نمره تبدیل شود!
- برای اطلاعات بیشتر به فایل "ARM Processor Instruction Set" که همراه با تمرین در سایت درس آپلود شده است مراجعه کنید.

References:

- [1] "ARM Processor Instruction Set," ARM7500FE Data Sheet, Preliminary
- [2] ARM Ltd., "ARM Information Center," Copyright 2012
- [3] <http://en.wikipedia.org/wiki/Endianness>

موفق باشید

¹⁶ Verify