



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۲-۱۴۰۳
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل ۱۴۰۲/۱۰/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

دستور کار:

- ❖ پس از ۱۰ روز تاخیر مجاز در مجموع کل تمرینات (کامپیوتری + دستی)، تحویل تمرین با تاخیر شامل جریمه می باشد (هر روز 25 درصد).
- ❖ نام فایل ارسالی را در قالب ACA_HW(number)_studentID بارگذاری شود.
- ❖ تمامی پروژه ها از لحاظ شباهت، کنترل و بررسی می شوند بنابراین از کپی کردن خودداری فرمایید چنانچه در صورت مشاهده صفر لحاظ خواهد شد و نیز در مجموع نمرات جریمه خواهید شد.
- ❖ پاسخ خود را در ادامه سوال و داخل فایل word قرار گرفته در سامانه قرار دهید و برای اسکن کردن پاسخ های خود از CamScanner استفاده کنید و طبق فرمت خواسته شده فایل را pdf شده اپلود کنید.
- ❖ راه ارتباطی با حل تمرین:

sara.zamani73@aut.ac.ir



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۲-۱۴۰۳
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل ۱۴۰۲/۱۰/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

1- جدول زیر را برای توماسولو و اسکوربرد کامل کنید.

| Hazard | Scoreboard | Tomasulo |
|--------|------------|----------|
| FU | | |
| RAW | | |
| WAR | | |
| WAW | | none |

2- جدول دستورات را برای Tomasulo ای با مشخصات یادشده کامل کنید.

a. یک واحد Integer برای Store/Load با تاخیر 2

b. یک واحد جمع (Sub/Add) با تاخیر 2

c. یک واحد ضرب کننده با تاخیر 10

d. یک واحد تقسیم کننده با تاخیر 40

LD F6 24+ R2

LD F2 44+ R3

MULT F0 F2 F4

SUBD F8 F6 F2

DIVD F10 F0 F6

ADDD F6 F8 F2



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۲-۱۴۰۳
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل ۱۴۰۲/۱۰/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

| Instruction | Issue | Execution Complete | Write Result |
|-------------|-------|--------------------|--------------|
| LD | 1 | 3 | |
| LD | 2 | | |
| MULT | | | |
| SUBD | | | |
| DIVD | | | |
| ADDD | | | |

3- در سیکل کلاک 14 و 21 وضعیت Status Unit Functional و status result Register را بکشید.

4- در این تمرین، به این می پردازیم که چگونه تکنیک های نرم افزاری می توانند موازی سازی سطح دستورالعمل (ILP) را در یک حلقه برداری مشترک استخراج کنند. حلقه زیر به اصطلاح حلقه DAXPY (aX با دقت دوگانه به اضافه Y) است و عملیات مرکزی در حذف گاوسی است. کد زیر پیاده سازی DAXPY عملیات، $Y=aX+Y$ را برای طول برداری 10 پیاده سازی می کند. در ابتدا، $R1$ به آدرس پایه آرایه X و $R2$ به آدرس پایه Y تنظیم می شود:

```

addi    x4,x1,#80 ; x1 = upper bound for X
foo: fld      F2,0(x1) ; (F2) = X(i)
fmul.d  F4,F2,F0 ; (F4) = a*X(i)
fld      F6,0(x2) ; (F6) = Y(i)
fadd.d  F6,F4,F6 ; (F6) = a*X(i) + Y(i)
fsd      F6,0(x2) ; Y(i) = a*X(i) + Y(i)
addi    x1,x1,#8 ; increment X index
addi    x2,x2,#8 ; increment Y index
sltu    x3,x1,x4 ; test: continue loop?
bnez    x3,foo ; loop if needed

```



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۲-۱۴۰۳
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل ۱۴۰۲/۱۰/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

تأخیرهای functional unit را همانطور که در جدول زیر نشان داده شده است فرض کنید. یک one-cycle delayed را فرض کنید که در مرحله ID انجام می‌شود. فرض کنید که نتایج به طور کامل گذشتن.

| Instruction producing result | Instruction using result | Latency in clock cycles |
|----------------------------------|--------------------------|-------------------------|
| FP multiply | FP ALU op | 6 |
| FP add | FP ALU op | 4 |
| FP multiply | FP store | 5 |
| FP add | FP store | 4 |
| Integer operations and all loads | Any | 2 |

(a) یک single-issue pipeline را در نظر بگیرید. نشان دهید که حلقه چگونه بدون زمانبندی توسط کامپایلر و پس از زمانبندی کامپایلر برای عملیات ممیز شناور و تأخیرهای branch، از جمله هرگونه stall یا clock cycle بیکار به نظر می‌رسد. زمان اجرا (بر حسب cycle) برای هر عنصر بردار خروجی، Y، زمانبندی نشده و زمانبندی شده چقدر است؟ برای اینکه سخت افزار پردازنده به تنهایی با بهبود عملکرد به دست آمده توسط کامپایلر زمانبندی مطابقت داشته باشد، clock چقدر باید سریعتر باشد؟ (از هر گونه تأثیر احتمالی افزایش سرعت clock بر عملکرد سیستم حافظه چشم‌پوشی کنید).

(b) یک single-issue pipeline را در نظر بگیرید. حلقه را هر چند بار که لازم است باز کنید تا بدون هیچ stallی زمانبندی شود و دستورالعمل‌های بالای حلقه را جمع کنید. چند بار باید حلقه باز شود؟ زمانبندی را نشان دهید. زمان اجرای هر عنصر نتیجه چقدر است؟

(c) همانطور که در شکل زیر نشان داده شده است، یک پردازنده VLIW با دستورالعمل‌هایی که شامل پنج عملیات است، فرض کنید. ما دو دور باز کردن حلقه را با هم مقایسه خواهیم کرد. ابتدا حلقه را 6 بار باز کنید تا ILP استخراج شود و آن را بدون هیچ توقفی (یعنی چرخه‌های issue کاملاً خالی) برنامه ریزی کنید، دستورالعمل‌های بالای حلقه را جمع کنید، و سپس روند را تکرار کنید اما حلقه را 5 بار باز کنید. branch delay slot را نادیده بگیرید. زمانبندی را نشان دهید. زمان اجرای هر عنصر بردار خروجی برای هر زمانبندی چقدر است؟ چند درصد



دانشکده مهندسی کامپیوتر

بسمه تعالی
معماری کامپیوتر پیشرفته
نیمسال اول ۱۴۰۲-۱۴۰۳
تمرین (۳)



دانشگاه صنعتی امیرکبیر

مهلت تحویل ۱۴۰۲/۱۰/۱۵

شماره دانشجویی:

نام و نام خانوادگی:

از slot عملیات در هر زمانبند استفاده می شود؟ اندازه کد بین دو زمانبندی چقدر تفاوت دارد؟ مجموع تقاضای رجیستر برای این دو برنامه چقدر است؟

| Memory reference 1 | Memory reference 2 | FP operation 1 | FP operation 2 | Integer operation/branch |
|--------------------|--------------------|--------------------|-------------------|--------------------------|
| fld f0,0(x1) | fld f6,-8(x1) | | | |
| fld f10,-16(x1) | fld f14,-24(x1) | | | |
| fld f18,-32(x1) | fld f22,-40(x1) | fadd.d f4,f0,f2 | fadd.d f8,f6,f2 | |
| fld f26,-48(x1) | | fadd.d f12,f0,f2 | fadd.d f16,f14,f2 | |
| | | fadd.d f20,f18,f2 | fadd.d f24,f22,f2 | |
| fsd f4,0(x1) | fsd f8,-8(x1) | fadd.d f28,f26,f24 | | |
| fsd f12,-16(x1) | fsd f16,-24(x1) | | | addi x1,x1,-56 |
| fsd f20,24(x1) | fsd f24,16(x1) | | | |
| fsd f28,8(x1) | | | | bne x1,x2,Loop |

Figure 3.20 VLIW instructions that occupy the inner loop and replace the unrolled sequence. This code takes 9 cycles assuming correct branch prediction. The issue rate is 23 operations in 9 clock cycles, or 2.5 operations per cycle. The efficiency, the percentage of available slots that contained an operation, is about 60%. To achieve this issue rate requires a larger number of registers than RISC-V would normally use in this loop. The preceding VLIW code sequence requires at least eight FP registers, whereas the same code sequence for the base RISC-V processor can use as few as two FP registers or as many as five when unrolled and scheduled.