به نام خداوند بخشنده مهربان

معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دکتر اسدی
بهار ۱۴۰۳

مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶

تمرین اول

سوال ۱

۱. در بخشی از درس با یک کلاس میانی از ماشین‌ها که دارای هسته RISC و رابط میانی[1] از نوع CISC بودند، آشنا شدید. یک مثال واقعی از چنین پردازنده‌هایی بیابید و در مورد نحوه‌ی کارکرد آن‌ها توضیح دهید.

یک مثال واقعی از این نوع پردازنده ها Crusoe Transmeta است که در واقع یک پردازنده ی RISC است ولی به برنامه نویس ها دستوراتی مشابه دستورات CISC ارائه می دهد.
این پردازنده به چند دلیل توانسته این نوع معماری را داشته باشد.

- code morphing

  این پردازنده به صورت پویا دستورات ۸۶x را گرفته و آنها را به دستورات داخلی خود یعنی VLIW ترجمه می کند

- ترجمه کد و بهینه کردن آن منطبق با دستورات RISC

  این پردازنده ها در داخل ساختار خود یک لایه ی سخت افزاری دارند که به واسطه ی آن می توانند دستورات پیچیده ی CISC را به دنباله ای از دستورات RISC تبدیل کنند. چه قبل اجرا و چه در زمان اجرا.

سوال ۲

۲. به پرسش‌های زیر به صورت کامل پاسخ دهید.

۱. کلاس‌های موازی‌سازی را نام ببرید.
۲. کلاس‌های موازی‌سازی در اپلیکیشن را نام ببرید و هر کدام را توضیح دهید.
۳. کلاس‌های موازی‌سازی در معماری کامپیوتر را نام ببرید و هر کدام را توضیح دهید.

۱.

۱) موازی سازی در سطح دیتا
۲) موازی سازی در سطح ترد
۳) موازی سازی در سطح درخواست

۲.

۱) موازی سازی در سطح دیتا
چند داده را بتوان همزمان دستکاری کرد و عملیات بر روی آنها انجام داد
۲) موازی سازی در سطح تسک
موازی سازی تسک هایی که مستقل از یکدیگر هستند

۳.

۱) موازی سازی در سطح دستوز
این نوع موازی سازی سعی می کند که به اجرای یک برنامه سرعت بخشد. با استفاده از تکنیک هایی مانند pipelining ، که به معنی به صف کردن دستوراتی که قرار است اجرا شوند، پیشبینی پرش و اجرای خارج از ترتیب.
۲) موازی سازی در سطح ترد
همان مفهوم multithreading که در زبان های برنامه نویسی پیشرفته تر نیز مشاهده می شود. به این معنی که از موازی کردن تعدادی ترد استفاده کنیم تا throughput کلی سیستم را افزایش دهیم
۳) موازی سازی در سطح درخواست
موازی سازی در سطح درخواست به این معنی است که اجرای برنامه های بزرگ و تسک های کلی و حجیم را به صورت موازی انجام دهیم. این مورد توسط برنامه نویس یا os مشخص می شود
۴) معماری برداری و GPU
به معنی انجام یک دستور بر روی مجموعه ای از داده ها است

سوال ۳

۳. فرض کنید برنامه‌ای از دو بخش ترتیبی و قسمت موازی تشکیل شده است که این برنامه با سیستمی با n پردازنده اجرا می‌شود.

آ) اگر اجرای برنامه روی یک پردازنده ۱ واحد زمانی طول بکشد، زمان اجرا روی n پردازنده (با فرض اینکه پردازنده‌ها یکسان بوده‌اند) و میزان تسریع را بدست آورید.

ب) اگر اجرای برنامه روی n پردازنده ۱ واحد زمانی طول بکشد، زمان اجرای برنامه روی یک پردازنده (با فرض اینکه پردازنده‌ها یکسان بوده‌اند) و میزان تسریع را بدست آورید.

ج) فرض کنید قدرت محاسباتی پردازنده iام متناسب با u_i است و به عبارتی زمان اجرا روی پردازنده iام برابر $\frac{1}{u_i}$ است. در این صورت پاسخ‌های قسمت آ و ب چگونه خواهد بود.

آ)

t = زمان بخش موازی
q = زمان بخش ترتیبی
p = قدرت پردازشی
w = میزان پردازش مورد نیاز بخش موازی
m = میزان پردازش مورد نیاز بخش ترتیبی

$$۱ = t + q, q = \frac{m}{p}, t = \frac{w}{p}$$

$$t \to \frac{t}{n}, t = ۱ - q$$

پس زمان اجرا $q + \frac{۱-q}{n}$ می شود و میزان ترسیع برابر است با $\frac{۱}{\frac{۱-q}{n}+q}$

ب)

$$t = \frac{w}{n * p}, \quad n * p \to p \quad \Rightarrow \quad t \to t * n$$

زمان اجرا $q + n(۱ - q)$ و میزان ترسیع $\frac{۱}{n(۱-q)+q}$

ج)

برای بخش الف)

$$t = \frac{w}{u_k}, \quad u_k \to \Sigma_{i=۰}^{n} u_i \quad \Rightarrow \quad t \to t * \frac{u_k}{\Sigma_{i=۰}^{n} u_i}$$

زمان اجرا برابر می شود با $q + (۱ - q) * \frac{u_k}{\Sigma_{i=۰}^{n} u_i}$ و میزان ترسیع $\frac{۱}{(۱-q)*\frac{u_k}{\Sigma_{i=۰}^{n} u_i}+q}$

برای بخش ب)

$$t = \frac{w}{\Sigma_{i=۰}^{n} u_i}, \quad \Sigma_{i=۰}^{n} u_i \to u_k \quad \Rightarrow \quad t \to t * \frac{\Sigma_{i=۰}^{n} u_i}{u_k}$$

زمان اجرا برابر می شود با $q + (۱ - q) * \frac{\Sigma_{i=۰}^{n} u_i}{u_k}$ و میزان ترسیع $\frac{۱}{(۱-q)*\frac{\Sigma_{i=۰}^{n} u_i}{u_k}+q}$ برای بخش ب)

سوال ۴

۴. یک پردازنده با مشخصات زیر پیاده‌سازی شده است:

۱. واکشی[۲] دستورات ۲ پالس ساعت به طول می‌انجامد.
۲. اجرای دستورات ۳ پالس ساعت به طول می‌انجامد.
۳. پردازنده در سرعت ۱۰۰ مگاهرتز کار می‌کند.

ایجاد تغییرات زیر ممکن است:

۱. انجام واکشی در یک پالس ساعت که باعث می‌گردد سرعت پردازنده به ۸۰ مگاهرتز کاهش یابد.
۲. اجرای دستورات در دو پالس ساعت که باعث می‌گردد سرعت پردازنده به ۷۵ مگاهرتز کاهش یابد.
۳. واکشی در ۳ پالس ساعت و اجرا در ۴ پالس ساعت که باعث می‌گردد سرعت تا ۱۵۰ مگاهرتز افزایش یابد.

چنانچه برنامه‌ای با تعداد دستور مشخص روی ساختارهای فوق اجرا شود، کدام ساختار به کمترین زمان اجرا منجر می‌شود؟

c = تعداد دستورات
t = زمان اجرای کل
f = فرکانس
p = زمان پردازش و اجرا به کلاک
v = زمان واکشی به کلاک

$$t = \frac{c * (p + v)}{f}$$

تحت ساختار ۱)

$$t = \frac{c(۳ + ۱)}{۸ * ۱۰^{۷}} = c * ۵ * ۱۰^{-۸}$$

تحت ساختار ۲)

$$t = \frac{c * (۲ + ۲)}{۷/۵ * ۱۰^{۷}} = c * ۵/۳۳ * ۱۰^{-۸}$$

تحت ساختار ۳)

$$t = \frac{c * (۴ + ۳)}{۱/۵ * ۱۰^{۸}} = c * ۴/۶۶ * ۱۰^{-۸}$$

با مقایسه ی t های به دست آمده می توان دید که تحت ساختار ۳ کمترین زمان اجرا را برای یک برنامه با تعداد دستور ثابت خواهیم داشت

سوال ۵

۵. فرض کنید زیرروال[۳] نوشته شده در قطعه کد اسمبلی MIPS زیر، با مقدار اولیه ۴ در ثبات a0 به واسطه دستور jal صدا زده شود. با فرض اینکه فرکانس کلاک ماشینی که برنامه روی آن اجرا می‌شود، ۹GHz باشد و همچنین با در نظر گرفتن جدول پایین، زمان اجرای این زیرروال را محاسبه کنید.

Instruction Type	CPI
ALU	0.5
Load	1.5
Store	0.9
Jump	1
Branch	0.8

```
fibonacci:
addi $sp, $sp, -8
sw $ra, 4($sp)
sw $a0, 0($sp)

li $t0, 1
ble $a0, $t0, base_case

addi $a0, $a0, -1
jal fibonacci
move $s1, $v0
lw $a0, 0($sp)
addi $a0, $a0, -2
jal fibonacci
add $v0, $s1, $v0

lw $ra, 4($sp)
lw $a0, 0($sp)
addi $sp, $sp, 8
jr $ra

base_case:
move $v0, $a0
addi $sp, $sp, 8
j epilogue

epilogue:
jr $ra
```

محاسبه می کنیم که از هر کلاس دستوری چند دستور استفاده می شود:

$$fibonacci(i) = \begin{cases} (ALU : ۳, Load : ۱, Store : ۲, Jump : ۲, Branch : ۱) & i < ۲ \\ fibonacci(i-۱) + fibonacci(i-۲) + (ALU : ۶, Load : ۴, Store : ۲, Jump : ۳, Branch : ۱) & o.w \end{cases}$$

$$base = (۳, ۱, ۲, ۲, ۱), regular = (۶, ۴, ۲, ۳, ۱)$$

حال ما $fibonacci(۴)$ را میخواهیم

$$\begin{aligned} fibonacci(۴) &= fibonacci(۳) + fibonacci(۲) + regular \\ &= fibonacci(۲) + fibbonaci(۱) + regular + fibonacci(۱) + fibonacci(۰) + regular + regular \\ &= fibonacci(۱) + fibonacci(۰) + regular + base + regular + base + base + regular + regular \\ &= ۵ * base + ۴ * regular \\ &= (۳۹, ۲۱, ۱۸, ۲۲, ۹) \end{aligned}$$

حال CPI برنامه را حساب می کنیم

$$CPI = \frac{۳۹ * ۰/۵ + ۲۱ * ۱/۵ + ۱۸ * ۰/۹ + ۲۲ * ۱ + ۹ * ۰/۸}{۱۰۹} = \frac{۹۶/۴}{۱۰۹}$$

$$t = \frac{CPI * IC}{f} = \frac{۹۶/۴}{۹ * ۱۰^۹} \approx ۱۰/۷ * ۱۰^{-۹}$$

سوال ۶

۶. یک طراح کامپایلر می‌خواهد از میان دو توالی دستورالعمل[۴] یکی را برای یک پردازنده‌ی خاص انتخاب کند. شرکت سازنده‌ی پردازنده، جدول زیر را که نشان‌دهنده‌ی CPI پردازنده برای کلاس‌های مختلف دستورالعمل‌ها است را در اختیار طراح کامپایلر قرار داده است:

Instruction Class	A	B	C
CPI	1	2	3

در جدول زیر این توالی دستورالعمل‌ها و تعداد دستورالعمل‌هایی که از هر کلاس از دستورالعمل‌ها دارند را می‌بینید:

	A	B	C
Sequence 1	2	2	2
Sequence 2	6	1	1

حال به سوالات زیر جواب دهید:

۱. کدام توالی دستورات بیش‌تری دارد؟

۲. کدام توالی روی پردازنده‌ی ذکرشده سریع‌تر اجرا خواهد شد؟

۳. CPI هر کدام از این توالی‌ها را به دست آورید.

۱)
توالی ۱ دارای ۶ دستور است و توالی ۲ شامل ۸ دستور است. پس توالی ۲
۳)

$$CPI_{seq۱} = \frac{i_A * CPI_A + i_B * CPI_B + i_C * CPI_C}{IC_{seq۱}} = \frac{۲*۱ + ۲*۲ + ۲*۳}{۶} = ۲$$

$$CPI_{seq۲} = \frac{i_A * CPI_A + i_B * CPI_B + i_C * CPI_C}{IC_{seq۲}} = \frac{۶*۱ + ۱*۲ + ۱*۳}{۸} = ۱/۳۷۵$$

۲)

$$t = \frac{CPI * IC}{f}$$

$$t_{seq۱} = \frac{۲*۶}{f}, t_{seq۲} = \frac{۱/۳۷۵*۸}{f}$$

با مقایسه ی زمان هر یک از توالی ها مشاهده می کنیم که توالی دوم سریع تر اجرا خواهد شد.