

امتحان سوال نرم افزار و زبان کامپیوتر

۲، ۲، ۱۳۹۳

نیت : دو ساعت

لطفاً تمیز و بدون خط خوردگی بنویسید.

۱ - پنج مدل آدرس دهی MIPS را در یک جدول نام برده ، مدل یا نزدیکترین مدل آدرس دهی پیشین به آن ؟ را به همراه توضیح مختصر در جدول ثبت و یک تفاوت جزئی آن ؟ ارائه دهید.

۲ - معنای مجموعه دستورالعمل در یک چار وتری یک پردازنده را مشخص می کند. آن ؟ (بسته) را نام ببرید.

۳ - کدام دستورات یا ویژگی (حداقل ۲ مورد را نام ببرید) وجه تمایز پردازنده های CISC و RISC به هم می ریزد ؟

۴ - در MIPS دستورالعمل `ror rdest, rsrc1, rsrc2` که عمل دوران را نسبت به بیت مبدأ `rsrc1` اندازه `rsrc2` و نتیجه را در `rdest` انجام می دهد به وقوع یک شبه دستور (pseudo instruction) است یعنی دستور واقعی اسمبلر نیست . بنیان اسمبلر مدل این شبه دستور را با کمک دستور واقعی MIPS (مثلاً بایه کاپی دستور `srlv`) در حداقل ۴ خط بنویسید.

۵ - مشخص کنید بنیان اسمبلر متبل کدام

گزاره زبان C را در پردازنده MIPS بنویسید

```
lw $s0, a
lw $s1, b
lw $s2, c
lw $s3, j
loop:
    add $s4, $s1, $s2
    lw $s4, 0($s4)
    bge $s0, $s4, skip
    bge $s0, $s3, skip
    b out
skip:
    add $s4, $s1, $s0
    sw $s2, 0($s4)
    addi $s0, $s0, 2
    b loop
    sw $s0, i
```

۶ - تفاوت پیوند زنی ایستا و دینامیک (Dynamic link) را برابری/شدنی (Object) ۲ به افتضاح بیان و برتری یا کالته هر یک را ذکر کنید.

۷ - مشخص کنید در "اسمبل" کوانتیک "گذر یک مرحله‌ای" (One pass assembly) مدول نماد؟ (symbol table) چگونه بروز رسانی می‌شود (حفظه برای هر چسبایی ۲ که در جلو یا بعد از دست‌جاری و کار دارند).

۸ - دستور `Beq 34h` در آدرس `10FE` قرار دارد. مشخص کنید اگر پس از انجام کد به بی ۲ خواهیم رفت (یعنی مقدار PC چه خواهد بود).

۹ - می‌خواهیم مقدار PC را در بیت‌بیم به ۱۰۰۰ تغییر دهیم. چه کار کنیم یا به کد توانی ۲ را بدین منظور اجرا کنیم؟

۱۰ - فرق بین وقفه‌دهی، پرش به زیر برنامه و فراخواندن هم‌روال (coroutine) ۴ به طور دقیق در چیست؟

۱۱ - مشخص کنید مقدار `eax`، `ebx` و `ecx` بعد از اجرای برنامه مقابل چه خواهد بود ۴

```
section .data
x:    dd 32
y:    dd 17
```

```
section .text
mov    eax, x
mov    ebx, [eax]
mov    ecx, [eax + 4]
add    ebx, [x]
add    ecx, ebx
mov    eax, [eax]
```

۱۲ - یک مثال برای در کردن پارامتر ۴ به زیر برنامه C به کمک اسمبلی بنویسید که شامل در کردن آدرس یک متغیر و پس از آن مقدار متغیر و برسی و پس گرفتن مقدار نتیجه خواهد بود. ارائه نمایید.

تعمیم امتحان میان ترم سفتا و زبان کامپیوتر

موضوع: ۹۳، ۲، ۲

۱ -

معماری MIPS	شرح اجزای	معماری Pentium	ملاحظات
۱) Immediate: آنی مثال: <code>addi \$s1, \$s0, -1</code>	<code>[op rs rt immediate]</code> مقدار عملوند در داخل دستور ۳۲ بیتی قرار دارد و ۱۶ بیتی است	آنی مثال: <code>mov AX, 1234</code>	مقدار عملوند آنی می تواند ۱۶، ۳۲ و ۶۴ بیتی باشد زیرا دستور با بیتی طول ثابت ندارد
۲) Register addr. بیتی مثال: <code>slt \$t2, \$t0, \$t1</code>	یک عمل منطقی یا جابجایی فقط با ۱ انجام می شود.	بیتی مثال: <code>mov AX, BX</code>	در بینیم تنوع کمتری از بیتی و دستورات این چنین نیست MIPS داریم
۳) Base addr. مثال: <code>lw \$t2, 4(\$t1)</code>	عملوند از آدرسی که از جمع مقدار ثابت و یک مقدار ثابت در دستور العمل به دست می آید به بیتی مقدار منتقل می شود.	Register relative: بیتی مثال: <code>mov AX, [BX+1000h]</code>	در بینیم مقدار ثابت اجزایی می تواند ۱۶ بیتی یا ۳۲ بیتی باشد حال آنکه در MIPS محدود به ۱۶ بیتی است
۴) PC-relative addr. آدرس دهی بیتی (PC) مثال: <code>beq \$t0, \$zero, exit</code> (مقدار exit برابر با آدرس بعد از این است)	عمل آدرس دهی از جمع مقدار PC و عدد ثابت داخل دستور (که offset نامیده می شود) به دست می آید	به این نام آدرس دهی داریم مقدار عدد در دستور relative یعنی بیتی بیتی شکل بیتی نیز به طور کلی PC انجام می شوند.	تا حد زیادی به است
۵) Pseudo direct addr. آدرس دهی شبه مستقیم مثال: <code>J done</code>	عمل پرش از ترکیب یک آدرس ۲۶ بیتی موجود در دستور (پرشی) که دو بیت به چپ جابجایی شده است و ۴ بیت با بیتی PC به هم می پیوندد یک آدرس ۳۲ بیتی مستقیم می شود	مستقیم مثال: <code>mov [1234h], AX</code>	در اینجا آدرس کامل تعیین می شود و فقط فاصله بیتی نیست مثال MIPS

۲ - مبنای مجموعه دستور العمل ISA: Instruction Set Architecture و بیتی زیر

۱. مشخص می کند:

— تعداد بیتی

— طول دستور

— تعداد دستور و انواع آن

— به نام آدرس دهی

— گونه (type) داده

— تعداد عملوندهای قابل استفاده در دستور

— میزان بیتی هر دستور العمل

(ص ۱)

۳- ویدر دستور load/store فقط برابر در آید حافظه

تعداد بیت زیاد

طول دستور ثابت

[و تعداد نسبتاً کم دستور و مقدار در سردهی]

۲

وشرقی است متناظر کننده RISC

نسبت به پردازنده های CISC می باشد.

مقدار جابجایی ثابت مبدأ ثابت مقصد
ror rodest, rsrc1, rsrc2

ع - ثابت مبدأ: rsrc1

اندازه مقدار تعیین شده

در نسبت rsrc2 دوران می کند

و در نسبت rodest ذخیره می شود

۲

⇔ { sub \$t1, \$zero, \$rsrc2
sllv \$t1, \$rsrc1, \$t1
srlv \$rodest, rsrc1, rsrc2
or rodest, rodest, \$t1

دستور اول مکمل مقدار جابجایی را (به نسبت ۳۲ بیت)

در نسبت موقت \$t قرار می دهد مثلاً اگر ۴ بیت در اندازی فوایم

در \$t مقدار ۲۸ قرار می دهد. سپس نسبت مبدأ را ۲۸ بیت به چپ جابجایی می کند (sllv: shift logical left variable)

تا ۴ بیت سمت راست به سمت چپ در \$t آید.

در دستور دوم صرفاً یک جابجایی به راست به اندازه rsrc2 انجام می شود.

و دستور آخر آن ۴ بیت (مثلاً) ذخیره کنده موقت را به سمت چپ (نسبت چپ) rodest می جابجاند.

یعنی دستور یکبار هم ROR عمل می کند که به اجرا و تبار با جابجایی نسبت چپ می باشد.

۵ -

for (i=a; (i<b[c] || (i<2) i+=2)

⇔ b(i)=c و

۲

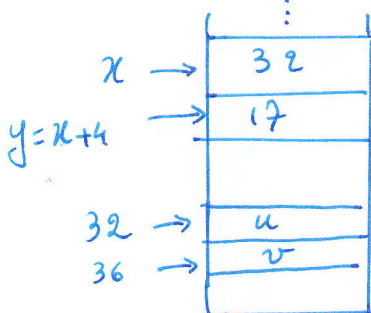
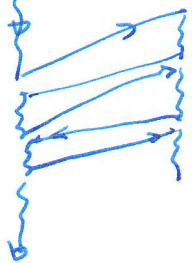
- (1) lw \$s0, a
- (2) lw \$s1, b
- (3) lw \$s2, c
- (4) lw \$s3, d

- (5) loop add \$s4, \$s1, \$s2
- (6) lw \$s4, 0(\$s4)
- (7) bge \$s0, \$s4, skip
- (8) bge \$s0, \$s3, skip
- (9) b out
- (10) skip add \$s4, \$s1, \$s0
- (11) sw \$s2, 0(\$s4)
- (12) addi \$s0, \$s0, 2
- (13) b loop
- (14) sw \$s0, n

مثلاً ۵ و در ابتدا به مقدار ۵ برابر است و در خط ۱۲ دوباره آن افزوده می شود تا طلعه بهر رابطه کد دستور ۱۳ نگارنده. پایال طلعه نیز نوکتن \$s0 در مقیاس فواید بود. نسبت ۴/۵، ۵ و ۵ نیز به ترتیب در دستور ۲، ۳ و ۴ مقدار b، c و n را دریافت می کنند. در خط ۵ آدرس b[c] را حساب می کنند و در خط ۶ خود [c] طای می شود که در خط ۷ با n (تبار حجم) مقایسه می شود و اگر کوچکتر بود نیز به بولان نیز از n (در خط ۸) کوچکتر نبود خطه ۱۰ نگاری در وگرنه خارج می شود.

۱- در وقفه دهی، ثابت CS و IP در پشته ذخیره می‌شوند و بعد بریم وقفه IF صرف شده، بزبان یار دال وقفه براساس شماره وقفه که در جدول وقفه مقادیر CS و IP آن مشخص می‌شود. البته امکان پوچ شدن mask وقفه و عدم اجرای آن وجود دارد ولی زیر بزبان صحت در حدت فراخوانی اجرا می‌شود. زیر بزبان call می‌تواند از نوع نزدیک near و ذخیره موقت IP در پشته و برتر: آرر برتر بزبان باشد یا از نوع دور far که ذخیره موقت CS و IP در پشته زیر بزبان را می‌دهد. بنابراین در این نوع دستور محل برپش مشخص می‌شود حال آنکه در وقفه صرفاً یک شماره یا کوکط بزبان داده می‌شود (مثلاً $\text{int } 255$) یا سنت افزا خود تعیین می‌کند که نام دال وقفه (یا آرر سی که از جدول وقفه استخراج می‌کند) باید اجرا گردد. وقفه هم نرم افزاری مادل یک call far هستند یا این تفاوت یا امتیاز که فقط ۲ بیتی اند حل می‌کند مادل call که بیتی و طولانی تر است. هم دال بزبان می‌گفتی که آنکه که اگر می‌توان اجرای آن بزبان اصلی برگردیم و دوباره آن را فراخوانی کنیم از محل انقطاع قبلی کار ادامه پیدا می‌کند.

هم دال = coroutine بزبان اصلی



section .data

x: dd 32

y: dd 17

section .text

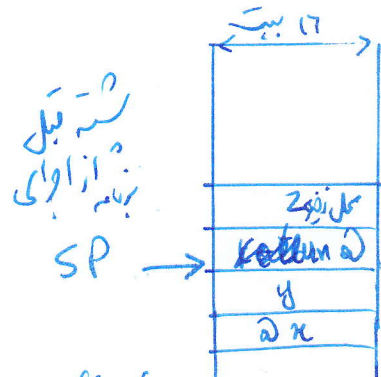
```
mov eax, x          ; eax ← 32
mov ebx, [eax]       ; ebx ← u
mov ecx, [eax+4]     ; ecx ← v
add ebx, [x]          ; ebx ← 2xu
add ecx, ebx          ; ecx ← 2xu + v
mov eax, [eax]       ; eax ← u
```

۱۱

۲

۱۲- فرض کنید تابع F_1 آرر بر متغیر x و مقدار متغیر y را می‌گیرد و مقدار $z = x + y$ را حساب کرده بازمی‌گرداند. همین طوری فرض کنید که متغیر z از نوع int و ۱۶ بیتی باشد. بزبان اسمبلی مادل F_1 می‌تواند چنین باشد:

$z = F_1(*\text{int } x, \text{int } y)$



```
{
- Asm
} mov BP, SP
mov AX, [BP-2] ; AX ← y
mov SI, [BP-4] ; SI ← x
mov BX, [SI]   ; BX ← x
add AX, BX     ; AX ← x + y
mov [BP+2], AX ; ذخیره در پشته
}
```

۳

(صحت)