بسمه تعالی

معماری کامپیوتر

نیمسال دوم ۱۴۰۱-۱۴۰۲

تمرین امتیازی

دانشگاه صنعتی امیرکبیر **دانشکده مهندسی کامپیوتر**

مهلت تحویل:

نام و نام خانوادگی: **شماره دانشجویی:**

۱) کد زیر را در نظر بگیرید (۰.۵)

```
register int i, j;  /* i, j are in the processor registers */
register float sum1, sum2;
float a[64][64], b[64][64];

for (i = 0; i < 64; i++) {                  /* 1 */
        for (j = 0; j < 64; j++) {          /* 2 */
                sum1 += a[i][j];            /* 3 */
        }
        for (j = 0; j < 32; j++) {          /* 4 */
                sum2 += b[i][2*j];          /* 5 */
        }
}
```

موارد زیر را فرض کنید:

- یک کش بی نقص وجود دارد. یعنی نگران زمان دسترسی به دستورالعمل نباشید.
- int و float هر دو اندازه ۴ بایت هستند.
- فقط دسترسی‌ها به مکان‌های آرایه a_{ij} و b_{i2j} بارهای کاری را در حافظه پنهان ایجاد می‌کنند. بقیه متغیرها همگی در ثبات‌ها تخصیص داده می‌شوند.
- حافظه نهان داده LRU کاملاً انجمنی را با ۳۲ خط فرض کنید که در آن هر خط ۱۶ بایت دارد.
- در ابتدا، کش داده‌ها خالی است.
- برای ساده نگه داشتن موارد، فرض می‌کنیم که دستورات در کد بالا به صورت متوالی اجرا می‌شوند. زمان اجرای خطوط (۱)، (۲) و (۴) ۴ کلاک برای هر فراخوانی است. خطوط (۳) و (۵) ۱۰ کلاک طول می‌کشند تا اجرا شوند و ۴۰ کلاک اضافی برای منتظر ماندن داده‌ها در صورت عدم وجود حافظه پنهان داده‌ها.
- یک دستورالعمل پیش واکشی داده با فرمت (prefetch(array[index وجود دارد. این دستور کل بلوک حاوی کلمه آرایه[index] را از قبل در حافظه پنهان واکشی می‌کند. ۱ کلاک طول می‌کشد تا پردازنده این دستور را اجرا کند و به حافظه پنهان داده ارسال کند. سپس پردازنده می‌تواند ادامه دهد و دستورالعمل‌های بعدی را اجرا کند. اگر داده‌های از پیش واکشی شده در حافظه پنهان نباشد، ۴۰ چرخه طول می‌کشد تا داده‌ها در حافظه پنهان بارگذاری شوند.
- آرایه‌های a و b به صورت ردیف اصلی ذخیره می‌شوند.

- آرایه‌های a و b هر دو از مرزهای خط کش شروع می‌شوند.

به سوالات زیر پاسخ دهید

بخش A

اگر از پیش واکشی استفاده نکنیم قطعه کد بالا چند چرخه طول‌می کشد تا اجرا شود؟

قسمت B

درج دستورالعمل پیش واکشی داده برای دو حلقه داخلی به ترتیب آرایه‌های a و b را در نظر بگیرید. توضیح دهید که چرا ممکن است بخواهیم حلقه‌ها را برای پیش واکشی، unroll کنیم. حداقل تعداد دفعاتی که باید برای هر یک از دو حلقه برای این منظور unroll کنید چقدر است؟

قسمت C

حلقه‌های داخلی را برای تعداد دفعاتی که در قسمت b مشخص شده است unroll کنید و حداقل تعداد پیش واکشی‌های نرم افزاری را وارد کنید تا زمان اجرا به حداقل برسد. تکنیک درج پیش واکشی مشابه با خط لوله کردن نرم افزار است.

قسمت D

اجرای کد قسمت (c) چند کلاک طول‌می کشد؟ میانگین speedup کد را بدون واکشی اولیه محاسبه کنید. فرض کنید واکشی‌های اولیه در کد راه اندازی وجود ندارد. زمان اضافی مورد نیاز توسط واکشی‌های اولیه که بیش از پایان زمان اجرای حلقه اجرا می‌شوند، نباید شمارش شود.

بخش E

آیا تکنیک دیگری وجود دارد که بتوان از آن برای دستیابی به همان هدفی مانند حلقه باز کردن در این مثال استفاده کرد، اما با دستورالعمل‌های کمتر؟ این تکنیک را توضیح دهید و کاربرد آن را برای کد قسمت (ج) نشان دهید.

بسمه تعالی

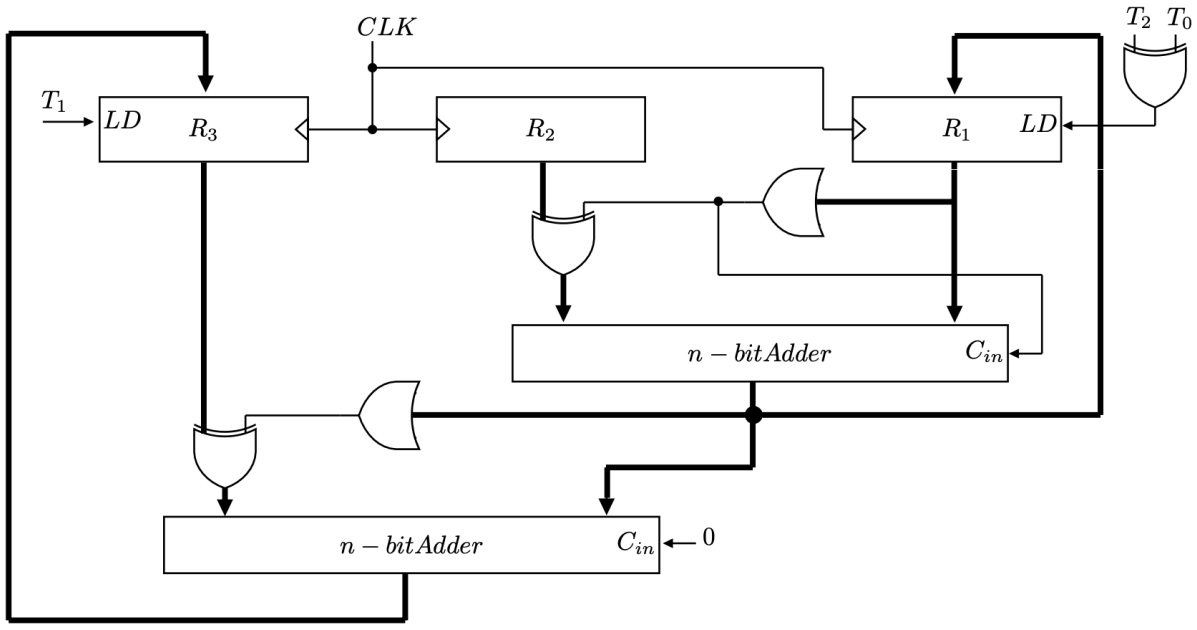
معماری کامپیوتر

نیمسال دوم ۱۴۰۱-۱۴۰۲

تمرین امتیازی

دانشگاه صنعتی امیرکبیر	**دانشکده مهندسی کامپیوتر**
مهلت تحویل:	
نام و نام خانوادگی:	**شماره دانشجویی:**

۲) عملکرد مدار زیر را ابتدا به صورت شبه‌کد، سپس آن شبه‌کد را به زبان RTL بنویسید. (۰.۲۵)



۳) کامپیوتر پایه ۳۲ بیتی نیاز داریم که عملیات ریاضی مشخص شده در جدول را انجام دهد (A و B دو عدد integer و C, D دو عدد با نمایش floating-point است). ابتدا کامپیوتر را طراحی کنید و بعد کد عملیات زیر را با کمک micro operationهای طراحی شده بنویسید. (حافظه سامانه ۶۴ کیلو ردیف چهار بایتی است) (۰.۵۵)

Y	func
A+B	۰۰۰۰
A-B	۰۰۰۱
C*D	۰۰۱۰
C/D	۰۰۱۱
sqrt(A)	۰۱۰۰
NOT A	۰۱۰۱
A AND B	۰۱۱۰
A OR B	۰۱۱۱
PUSH A	۱۰۰۱
POP	۱۰۱۰

- فرض کنید دستور sqrt وجود دارد و تنها لازم است در ریز عملیات خود مورد استفاده قرار بگیرد.
- تمامی ورودی‌های توابع در داخل Register File می‌باشند و آدرس آنها در قالب دستورالعمل‌ها به صورت ورودی داده می‌شود. برای خواندن از Register File از ثبات انتزاعی RFAR یا Register File Address Register می‌توانید استفاده کنید.
- با RF[RFAR] مقدار هرکدام را انتخاب کنید.
- ثبات‌های عام منظوره باید در Register File و از شماره ۰ الی ۱۲ وجود داشته باشند.
- ثبات Stack Pointer, Link Register, Program Counter در Register File قرار دارند.

بسمه تعالی

معماری کامپیوتر

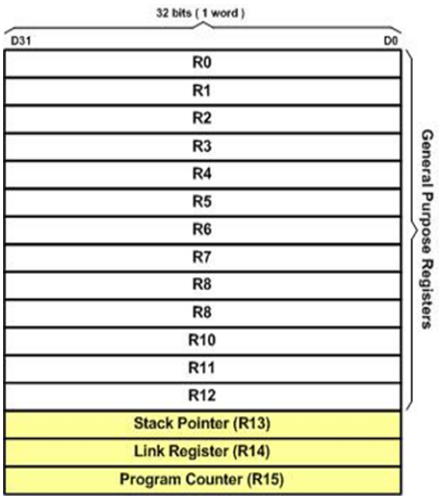
نیمسال دوم ۱۴۰۱-۱۴۰۲

تمرین امتیازی

دانشگاه صنعتی امیرکبیر **دانشکده مهندسی کامپیوتر**

مهلت تحویل:

نام و نام خانوادگی: **شماره دانشجویی:**



مطلوبست

۱. طراحی قالب دستورالعمل بهینه برای این سیستم.

۲. ترسیم مسیر داده این رایانه.

۳. ریز عملیات‌های لازم برای اجرای هرکدام از دستورات جدول را بنویسید.

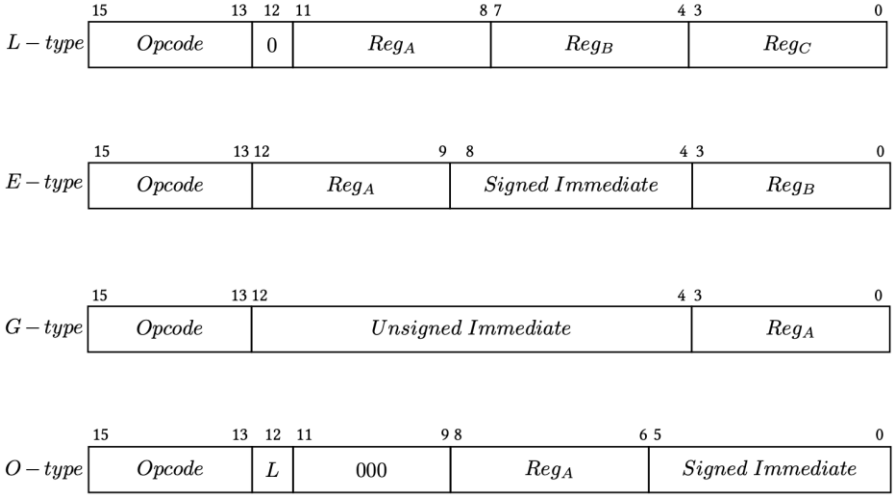
٤. ترسیم فلوچارت فرایند اجرای تمام دستورالعمل‌ها طبق الگوریتم فون‌نیومن.

٥. با توجه به وجود و مورد استفاده قرار گرفتن پایه‌های Clear, Decrement, Increment Load, طراحی واحد کنترل این رایانه برای تمامی ثبات‌ها را انجام دهید.

٦. کد عملیات ریاضی زیر را با کمک ریز عملیات‌های طراحی شده بنویسید. (توجه کنید اعداد باید از Stack خوانده شوند)

$$\frac{30.4}{7.2 * 2.1 + 1} - 3$$

۴) می‌خواهیم پردازنده‌ای ۱۶ بیتی به نام LEGO طراحی کنیم که دارای ۴ نوع دستور زیر است: (۵۵.۰)



همچنین پردازنده LEGO از دستورات جدول زیر پشتیبانی‌می‌کند:

دستور	نوع	کد دودویی	کد اسمبلی	عملیات
SUB	L-type	0000	$SUB\ Reg_A, Reg_B, Reg_C$	$Reg_A \leftarrow Reg_B - Reg_C$
ADD	L-type	0001	$ADD\ Reg_A, Reg_B, Reg_C$	$Reg_A \leftarrow Reg_B + Reg_C$
DIV	L-type	0010	$DIV\ Reg_A, Reg_B, Reg_C$	$Reg_A \leftarrow \frac{Reg_B}{Reg_C}$
MUL	L-type	0011	$MUL\ Reeg_A, Reg_B, Reg_C$	$Reg_A \leftarrow Reg_B \times Reg_C$
SW	L-type	0100	$SWReg_A, Reg_B, Reg_C$	$Mem[Reg_B + Reg_C] \leftarrow Reg_A$
LW	L-type	0101	$LW\ Reg_A, Reg_B, Reg_C$	$Reg_A \leftarrow Mem[Reg_B + Reg_C]$
SLT	L-type	0110	$SLT\ Reg_A, Reg_B, Reg_C$	$if\ (Reg_B < Reg_C)\ then\ Reg_A \leftarrow 1\ else\ Reg_A \leftarrow 0$
JAL	E-type	0111	$JAL\ Reg_A, Reg_B$	$PC \leftarrow 2 \times Reg_A + 2 \times S.E(imm), Reg_B \leftarrow PC + 2$
LUI	G-type	1000	$LUI\ Reg_A, imm$	$Reg_A \leftarrow imm \times 128$
MOV	G-type	1001	$MOV\ Reg_A, imm$	$Reg_A \leftarrow Z.F(imm)$
JMP	O-type	1100	$JMP\ L, Reg_A, imm$	$if\ (L = 1)\ then\ PC \leftarrow 2 \times Reg_A + 2 \times S.E(imm)$

بخش مسیر داده پردازنده را به صورتی طراحی کنید که هر عملیات در یک سیکل انجام شود (single Cycle). ریز عملیات هر دستور را بنویسید و همچنین سیگنال‌های کنترلی هر دستور را مشخص نمایید.

- در جدول بالا منظور از S.E همان Sign Extend و منظور از Z.F همان Zero Filling یا در واقع قرار دادن تعدادی صفر در سمت چپ عدد است.