

پرسش ۱: [محاسبات کامپیوتری زمان تقریبی پاسخگویی ۱۵ دقیقه]

یک کامپیوتر با قابلیت پردازش اعداد ممیزشناور ۳۲ بیتی داریم. مقدار عددی عدد ممیزشناور ذخیره شده در کلمه ۳۲ بیتی $b_{31} b_{30} \dots b_0$ از رابطی زیر به دست می‌آید. مقدار کوچکترین و بزرگترین عدد مثبت قابل نمایش در این ماشین را به دست آورید. توجه داشته باشید که در این سیستم نمایش اعداد نرمال نیستند.

ID	Value	
0	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{25} 2^{l-26} b_l\right) 2^S$	$S = -16 + \sum_{l=26}^{30} 2^{l-26} b_l$
1	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{24} 2^{l-25} b_l\right) 2^S$	$S = -32 + \sum_{l=25}^{30} 2^{l-25} b_l$
2	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{23} 2^{l-24} b_l\right) 2^S$	$S = -64 + \sum_{l=24}^{30} 2^{l-24} b_l$
3	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{22} 2^{l-23} b_l\right) 2^S$	$S = -128 + \sum_{l=23}^{30} 2^{l-23} b_l$

$b_{31} \ b_{30} \dots b_{24} \ b_{23} \dots b_0$
 علامت مخ ماشین

کوچکترین مقدار مثبت :

* کوچکترین مخ مخ ممکن :

+ ~ ماشین ممکن :

* علامت مثبت :

$$S = -64$$

$$1$$

$$\frac{1}{2}$$

$$\leftarrow 00\dots0$$

$$\leftarrow 00\dots0$$

$$\leftarrow 0$$

$$\frac{1}{2} * 1 * 2^{-64} \leq 2^{-65}$$

کوچکترین مقدار مثبت

2D. 2

برترین مقدار است :

$$\underbrace{11 \dots 1}_{\text{11 بار}}$$

* برترین توان ممکن :

$$S = -64 + 127 = 63$$

* حاصل می شود :

$$M = 101 \dots 1 = 2 - 2^{-24}$$

$\underbrace{\hspace{1.5cm}}_{\text{25 بار}}$

$$\frac{1}{2} \leftarrow \text{در } 2^0 \text{ : حاصل می شود } *$$

$$\frac{1}{2} (2 - 2^{-24}) + 2^{63} = (1 - 2^{-25}) + 2^{63}$$

$$= \underbrace{2^{63} - 2^{38}}_{\rightarrow}$$

برترین مقدار است :

2D: 2

پرسش ۲: [محاسبات کامپیوتری، زمان تقریبی پاسخگویی ۱۵ دقیقه]

ID	0	1	2	3
Dividend	+215	+219	+223	+227
Divisor	+21	+22	+23	+24

101001

تقسیم دو عدد علامت دار مثبت داده شده را انجام دهید.

00001101111

مقسوم (مبنای ۲):

010111

مقسوم علیه (مبنای ۲):

Step	E	A	Q
		000011	011111
① Shift		000110	111110
Sub		101001	
EC=0	0	101111	
Add		010111	
		000110	
② Shift		001101	111100
Sub		101001	
EC=0	0	110110	
Add		010111	
		001101	
③ Shift		011011	111000
Sub		101001	
EC=1	1	000100	
Q=1			111001
④ Shift		001001	110010
Sub		101001	
EC=0	0	110010	
Add		010111	
		001001	

Step	E	A	Q
⑤ Shift		010011	100100
Sub		101001	
EC=0	0	111100	
Add		010111	
		010011	
⑥ Shift		100111	001000
Sub		101001	
EC=1	1	010000	
Q=0			001001

9

خارج قسمت (مبنای ۱۰):

16

باقیمانده (مبنای ۱۰):

001001

خارج قسمت (مبنای ۲):

010000

باقیمانده (مبنای ۲):

2D, 2

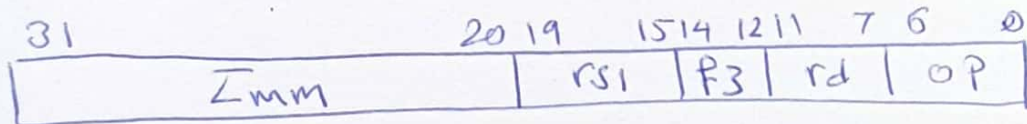
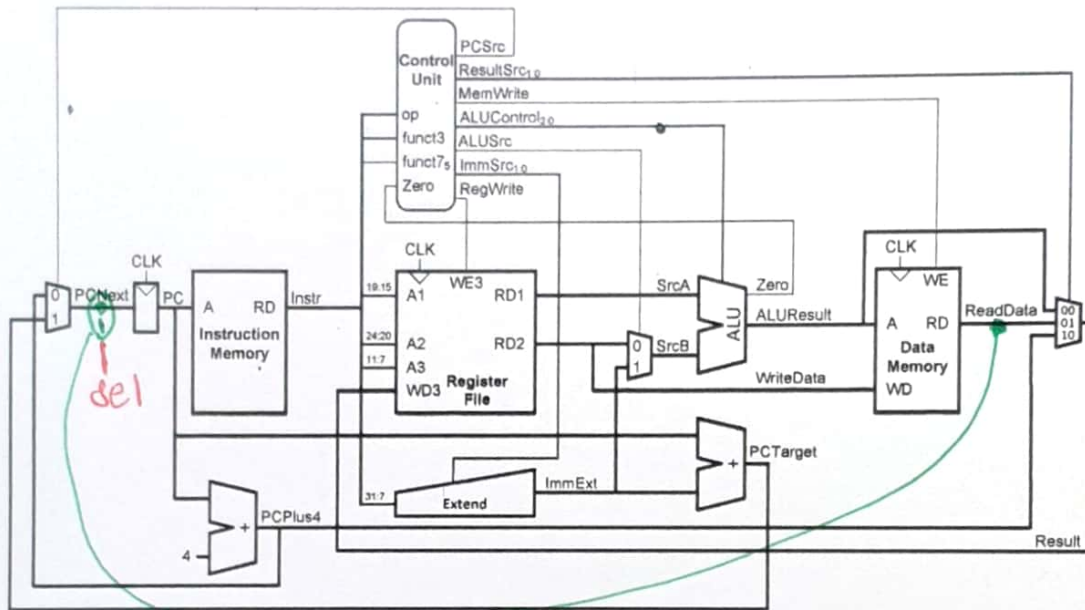
پرسش ۳: [معماری مجموعه دستورات پردازنده، زمان تقریبی پاسخگویی ۱۵ دقیقه]
برنامه‌ی زیر را به اسمبلی RISC-V بنویسید. A و B دو آرایه‌ی ۱۰۰ عنصری از اعداد صحیح ۳۲ بیتی بدون علامت هستند.

ID	0	1
	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] > B[i]) sum += A[i]; else sum += B[i];</pre>
ID	2	3
	<pre>sum = 0; for (i=99; i>=0; i--) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for (i=99; i>=0; i--) if (A[i] > B[i]) sum += A[i]; else sum += B[i];</pre>

add 80, zero, zero ; sum = 0
addi 81, zero, 396 ; i = 0
Loop: blt 81, zero, END-Loop, *توقف در صورتی که i از 0 کوچکتر شود*
lw t0, A(81) *خواندن A[i] به t0*
lw t1, B(81) *خواندن B[i] به t1*
bltu t0, t1, IF *تقریباً ۵۰٪ احتمال دارد که A[i] < B[i] باشد*
add 80, 80, t1 *در صورتی که A[i] < B[i] باشد، A[i] را به sum اضافه کن*
J END-IF *در صورتی که A[i] < B[i] باشد، به END-IF برو*
IF: add 80, 80, t0 *در صورتی که A[i] > B[i] باشد، B[i] را به sum اضافه کن*
END-IF: addi 81, 81, -4 *i را ۴ واحد کوچکتر کن*
J Loop
END-Loop:

حل سوال ۴ قسمت پ

ب- شکل زیر پیاده‌سازی Single-Cycle مسیر داده و کنترل پردازنده‌ی RISC-V را نشان می‌دهد. حداقل تغییرات لازم را در مسیر داده و کنترل اعمال کنید تا پردازنده توانایی اجرای دستور $\text{jmp_ind } \text{adr}(\text{rs1})$ را داشته باشد. این دستور به آدرس ذخیره شده در خانه‌ای از حافظه به آدرس $\text{rs1} + \text{adr}$ پرش می‌کند. برای این دستور از قالب I-Type استفاده کنید. مقدار تمام سیگنال‌های کنترلی را مشخص کنید.



برای اجرای دستور $\text{jmp_ind } \text{adr}(\text{rs1})$ در PC سیگنال‌ها:

Reg Write	Imm Src	ALU Src	Mem Write	Result Src	Branch	ALU op	del
0	00	1	0	xx	0	00	1