

2D: 3

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

یک کامپیوتر با قابلیت پردازش اعداد مغیرشناور ۳۲ بیتی داریم. مقدار عددی عدد مغیرشناور ذخیره شده در کلمه ۳۲ بیتی $b_{31} b_{30} \dots b_0$ از رابطه‌ی 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	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{22} 2^{l-23} b_l\right) 2^S$	$S = -128 + \sum_{l=23}^{30} 2^{l-23} b_l$

$b_{31} \quad b_{30} \dots b_{23} \quad b_{22} \dots b_0$
 علامت توان مانده

کوچکترین مقدار مثبت :

$$S = -128$$

۱

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

* کوچکترین توان ممکن : ۰۰...۰ ←

* مانده ~ : ۰۰...۰ ←

* علامت مثبت : ۰ ←

$$\frac{1}{2} * 1 * 2^{-128}$$

$$2^{-129}$$

کوچکترین مقدار مثبت

2D:3

زیرترین مقدار مثبت :

Δ
11...1

* زیرترین مقدار منفی :

$$S = -128 + 255 = 127$$

* \sim مابین مقادیر :

$$M = 101 \dots 1 = 2 - 2^{-23}$$

$\underbrace{\hspace{1.5cm}}_{\text{Cov 23}}$

* $\frac{1}{2}$ ← Δ : Cov 24 و Cov 23

$$\frac{1}{2} (2 - 2^{-23}) * 2^{127} = (1 - 2^{-24}) * 2^{127}$$

$$\begin{array}{r} 2^{127} \\ - 2^{103} \\ \hline \end{array}$$

\rightarrow

زیرترین مقدار مثبت

2D. 3

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

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

تقسیم دو عدد علامت‌دار مثبت داده شده را انجام دهید.

مقسوم (مبنای ۲): ۰۰۰۰۱۱۰۰۰۱۱

مقسوم علیه (مبنای ۲): ۰۱۱۰۰۰

۱۰۱۰۰۰ ←

Step	E	A	Q
		۰۰۰۰۱۱	۱۰۰۰۱۱
① shl sub E=0 Add		۰۰۰۱۱۱ ۱۰۱۰۰۰	۰۰۰۱۱۰
	0	۱۰۱۱۱۱ ۰۱۱۰۰۰	
		۰۰۰۱۱۱	
② shl sub E=0 Add		۰۰۱۱۱۰ ۱۰۱۰۰۰	۰۰۱۱۰۰
	0	۱۱۰۱۱۰ ۰۱۱۰۰۰	
		۰۰۱۱۱۰	
③ shl sub E=1 Q=1		۰۱۱۱۰۰ ۱۰۱۰۰۰	۰۱۱۰۰۰
	1	۰۰۰۱۰۰	۰۱۱۰۰۱
④ shl sub E=0 Add		۰۰۱۰۰۰ ۱۰۱۰۰۰	۱۱۰۰۱۰
	0	۱۱۰۰۰۰ ۰۱۱۰۰۰	
		۰۰۱۰۰۰	

Step	E	A	Q
⑤ shl sub E=0 Add		۰۱۰۰۰۱ ۱۰۱۰۰۰	۱۰۰۱۰۰
	0	۱۱۱۰۰۱ ۰۱۱۰۰۰	
		۰۱۰۰۰۱	
⑥ shl sub E=1 Q=1		۱۰۰۰۱۱ ۱۰۱۰۰۰	۰۰۱۰۰۰
	1	۰۰۰۱۰۱	۰۰۱۰۰۱

9
11

خارج قسمت (مبنای ۱۰):

باقیمانده (مبنای ۱۰):

۰۰۱۰۰۱
۰۰۱۰۱۱

خارج قسمت (مبنای ۲):

باقیمانده (مبنای ۲):

2D: 3

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.

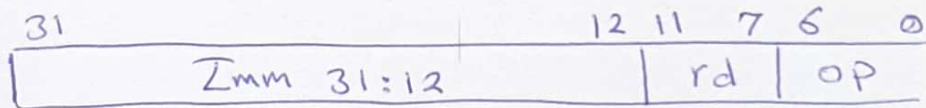
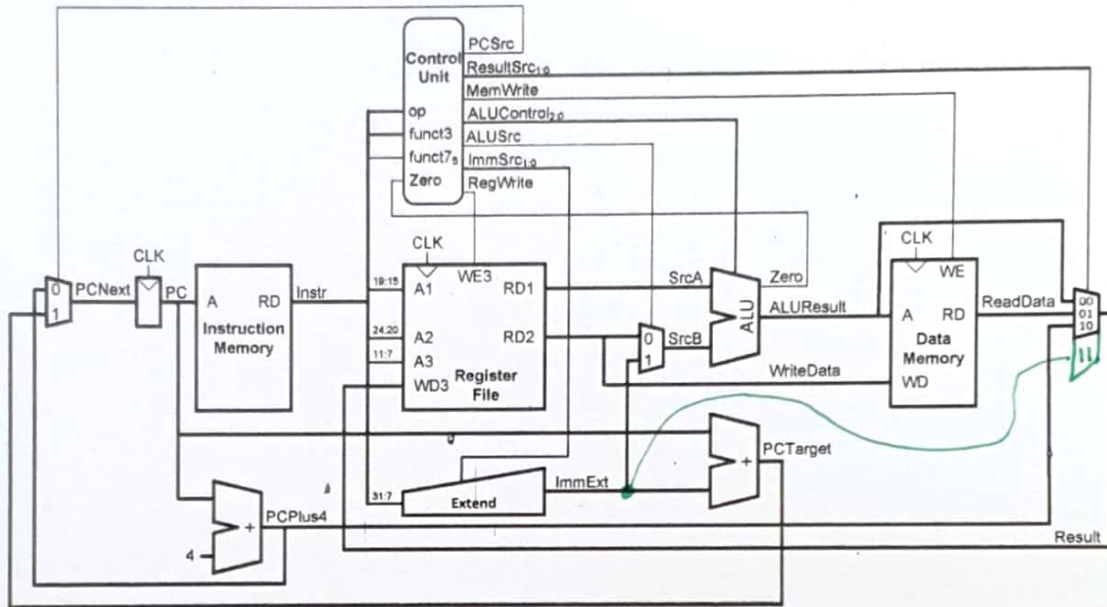
ID	0	1
	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] > B[i]) sum += A[i]; else sum += B[i];</pre>
ID	2	3
	<pre>sum = 0; for (i=99; i>=0; i--) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for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

add \$0, zero, zero ; sum = 0
 addi \$1, zero, 396 ; i = 0
 Loop: blt \$1, zero, END-Loop; شرط ختم شدن
 lw \$t0, A(\$1)
 lw \$t1, B(\$1) } در این مرحله
 bltu \$t1, \$t0, IF
 add \$0, \$0, \$t1
 j IF
 IF: add \$0, \$0, \$t0
 END-IF: addi \$1, \$1, -4
 j Loop
 END-Loop:

حل سوال ۴: مسأله الف

پرسش ۴: [طراحی، پردازنده، زمان تقریبی پاسخگویی ۳۰ دقیقه]

الف- شکل زیر پیاده‌سازی Single-Cycle مسیر داده و کنترلر پردازنده‌ی RISC-V را نشان می‌دهد. حداقل تغییرات لازم را در مسیر داده و کنترلر اعمال کنید تا پردازنده توانایی اجرای دستور lui data_20bit را داشته باشد. برای این دستور از قالب U-Type استفاده کنید. مقدار تمام سیگنال‌های کنترلی را مشخص کنید.



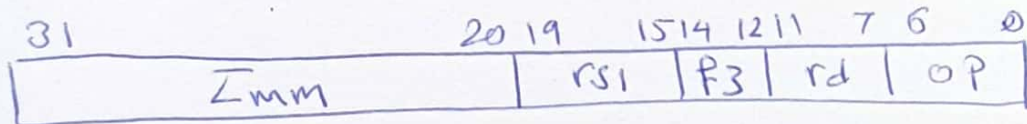
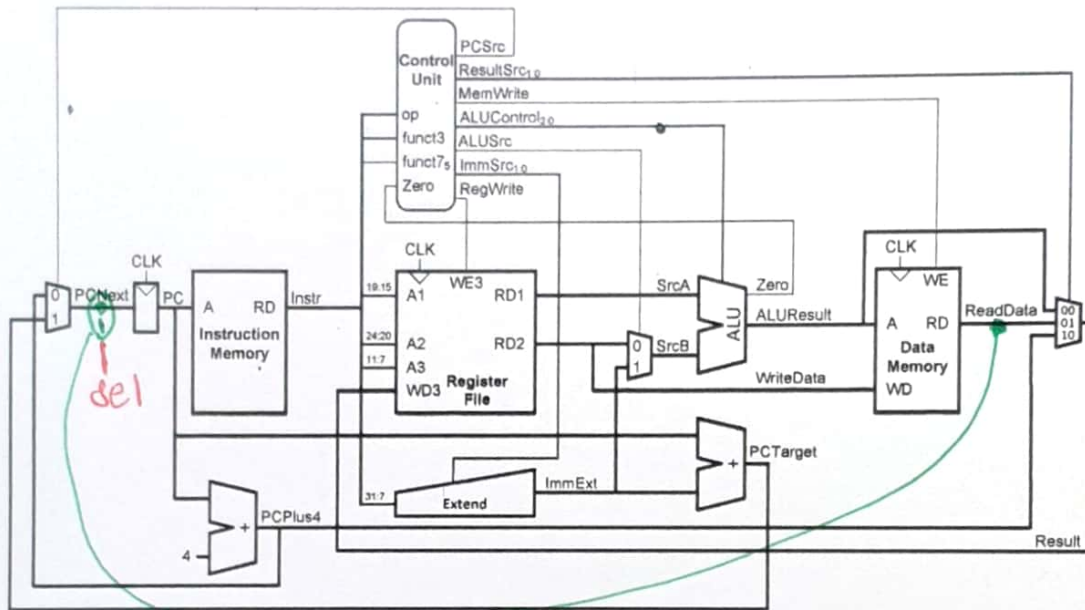
داده ۲۰ بیت می‌تواند از این ۲۰ بیت ($Imm[31:12]$) استفاده کند. تغییراتی که بتواند این ۲۰ بیت ($Imm[31:12]$) را تغییر دهد.

Imm_src	Imm_Ext	Imm_Type
100	$\{ \{ 12 \} Imm[31:12] \}$	U-Type

RegWrite	Imm_src	ALU src	MemWrite	Result src	Branch	ALU op
1	100	x	0	11	0	xx

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