

2D: 1

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 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	$\left(\frac{1}{2} - b_{31}\right) \left(1 + \sum_{l=0}^{22} 2^{l-23} b_l\right) 2^S$	$S = -128 + \sum_{l=23}^{30} 2^{l-23} b_l$

$b_{31} \quad b_{30} \quad \dots \quad b_{25} \quad b_{24} \quad \dots \quad b_0$
 علامت توان ماشین

کوچکترین مقدار مثبت:

$$S = -32$$

← 00...0

* کوچکترین توان ممکن

1

← 00...0

* ۱ ماشین ممکن

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

← 0

* علامت مثبت

$$\frac{1}{2} * 1 * 2^{-32} = \frac{2^{-33}}{2}$$

کوچکترین مقدار مثبت

2D:1

برترین قدر مثبت :

$$\frac{4}{11-1}$$

* برترین تراخ ممکن :

$$8 \leq -32 + 63 \leq 31$$

* بیشترین ممکن :

$$M \leq 1 + \underbrace{1 + \dots + 1}_{25 \text{ بار}} \leq 2 - 2^{-25}$$

* 2^{-26} مثبت : $\frac{1}{2} \leftarrow 0$

$$\frac{1}{2} (2 - 2^{-25}) * 2^{31} \leq (1 - 2^{-26}) * 2^{31} \leq$$

$$\leq \underbrace{2^{31} - 2^5}_{\text{برترین قدر مثبت}}$$

2D.1

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): ۰۰۰۰۱۱۰۱۱۰۱۱

مقسوم علیه (مبنای ۲): ۰۱۰۱۱۰

۱۰۱۰۱۰ ←

۰۱۰۱۱۰

مقسوم علیه (مبنای ۲): ۰۱۰۱۱۰

Step	E	A	Q
		۰۰۰۰۱۱	۰۱۱۰۱۱
① shl		۰۰۰۰۱۱	۱۱۰۱۱۰
sub		۱۰۱۰۱۰	
Geo	0	۱۱۰۰۰۰	
↳ Add		۰۱۰۱۱۰	
		۰۰۰۱۱۰	
② shl		۰۰۱۱۰۱	۱۰۱۱۰۰
sub		۱۰۱۰۱۰	
Geo	0	۱۱۰۱۱۱	
↳ Add		۰۱۰۱۱۰	
		۰۰۱۱۰۱	
③ shl		۰۱۱۰۱۱	۰۱۱۰۰۰
sub		۱۰۱۰۱۰	
Geo	1	۰۰۰۱۰۱	
↳ Q=1			۰۱۱۰۰۱
④ shl		۰۰۱۰۱۰	۱۱۰۰۱۰
sub		۱۰۱۰۱۰	
Geo	0	۱۱۰۱۰۰	
↳ Add		۰۱۰۱۱۰	
		۰۰۱۰۱۰	

Step	E	A	Q
⑤ shl		۰۱۰۱۰۱	۱۰۰۱۰۰
sub		۱۰۱۰۱۰	
Geo	0	۱۱۱۱۱۱	
↳ Add		۰۱۰۱۱۰	
		۰۱۰۱۰۱	
⑥ shl		۱۰۱۰۱۱	۰۰۱۰۰۰
sub		۱۰۱۰۱۰	
Geo	1	۰۱۰۱۰۱	
↳ Q=1			۰۰۱۰۰۱

9

خارج قسمت (مبنای ۱۰):

21

باقیمانده (مبنای ۱۰):

۰۰۱۰۰۱

خارج قسمت (مبنای ۲):

۰۱۰۱۰۱

باقیمانده (مبنای ۲):

2D : 1

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.

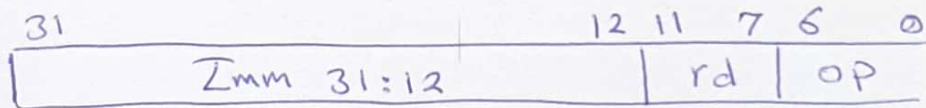
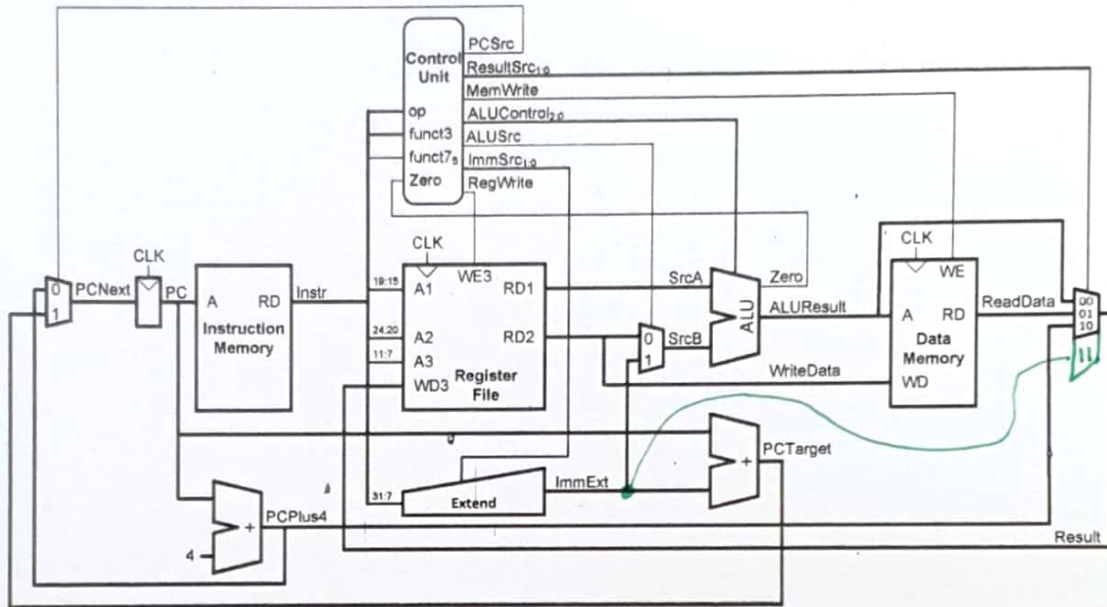
ID	0	1
	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for (i=0; i<100; i++) if (A[i] > B[i]) sum += A[i]; else sum += B[i];</pre>
ID	2	3
	<pre>sum = 0; for (i=99; i>=0; i--) if (A[i] < B[i]) sum += A[i]; else sum += B[i];</pre>	<pre>sum = 0; for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

add s0, zero, zero ; sum = 0
 add s1, zero, zero ; i = 0
 Loop: slti t0, s1, 400 } شرط حلقه
 beq t0, zero, END-loop } متوقف
 lw t0, A(s1) } خواندن عناصر A
 lw t1, B(s1) } خواندن عناصر B
 bltu t1, t0, IF } مقایسه و برقراری IF
 add s0, s0, t1 } جمع
 J END-IF } پرش
 IF: add s0, s0, t0 } جمع
 addi s1, s1, 4 } افزایش آرایه
 J Loop } پرش
 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 کنترلر را مشخص کنید.



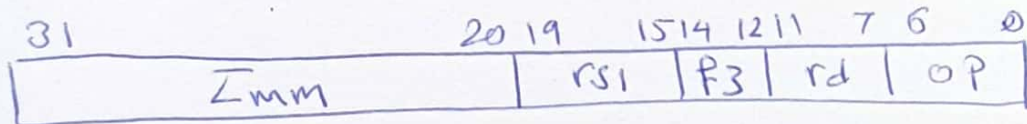
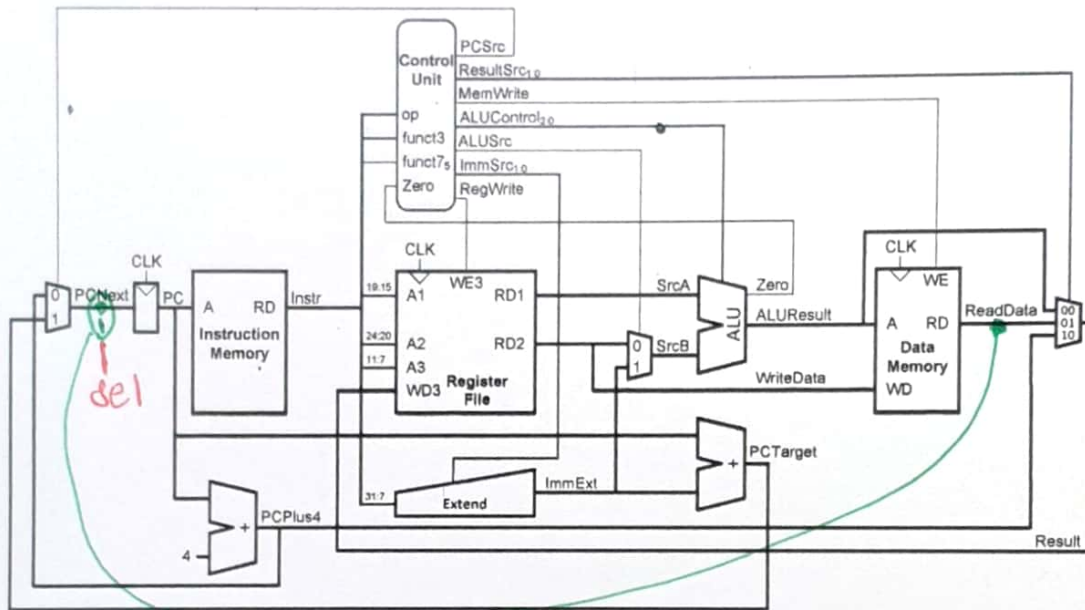
داده ۲۰ بیت می‌تواند از این ۲۰ بیت ($Imm[31:12]$) استفاده کند. تغییراتی که بتواند این ۲۰ بیت ($Imm[31:12]$) را تغییر دهد.

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