

Chapter-2

Question - 1:

Construct the equivalent RISC-V code of the following C code. Once you have the RISC-V code, identify type of each instruction and encode them accordingly.

$$A[7] = A[2] + A[B[8]] + 10;$$

$$B[i] = A[3] - 8;$$

Base addresses of array A and B are in register X_{20} and X_{21} and i is in register X_{22}

Line-1

```

ld  X5, 64(X2)    // B[8]
slli X5, X5, 3     // offset calculation (X5 & 2^3)
add X6, X20, X5   // physical loc. (Base + offset)
ld  X6, 0(X6)      // A[B[8]]

ld  X5, 16(X20)   // A[2]

add X5, X5, X6    // A[B[8]] + A[2]

addi X5, X5, 10   // A[B[8]] + A[2] + 10

sd  X5, 56(X20)   // A[7] = A[B[8]] + A[2] + 10

```

$$A = X_{20}$$

$$B = X_{21}$$

$$i = X_{22}$$

Line-2

```

ld  X6, 24(X20)   // A[3]
addi X6, X6, -8   // A[3] - 8

slli X7, X22, 3    // offset calc  $\Rightarrow X_{22} \& 2^3$ 
add X7, X7, X21   // Physical Add. calc

sd  X6, 0(X7)      // B[i] = A[3] - 8

```

No.	Instruction	Type	Encoding
1	ld $x_5, 64(x_2)$	I ✓	0000 0100 0000 10101 imm 1011 funct3 00101 rd XXXX opcode
2	Slli $x_5, x_5, 3$	I ✓	0000 00 imm 00 0011 00101 XXXX 00101 XXXX
3	Add x_6, x_{20}, x_5	R ✓	XXXXXXX 00101 10100 XXXX 00110 XXXX
4	ld $x_6, 0(x_6)$	I	
5	ld $x_5, 16(x_{20})$	I	
6	Add x_5, x_5, x_6	R	
7	Addi $x_5, x_5, 10$	I	
8	Sd $x_5, 56(x_{20})$	S ✓	0000 001 imm [1:5] 00101 10100 XXXX 11000 imm [4:0] XXXX
9	ld $x_6, 18(x_{20})$	I	
10	Addi $x_6, x_6, -8$	I ✓	1111 1111 1000 imm 00110 XXXX 00110 XXXX
11	Slli $x_7, x_{22}, 3$	I	
12	Add $x_7, x_7, 21$	R	
13	Sd $x_6, 0(x_7)$	S	

$$8 = 1000$$

$$+8 = 01000$$

$$+8 = 0000 0000 1000$$

$$\begin{array}{r} 1111 1111 0111 \\ +1 \end{array}$$

$$-8 = \overline{1111 1111 1000}$$

Encode rest of the instructions yourself.

Question - 2:

Construct the equivalent RISC-V code of the following C code.

```
for (i = 8; i > 0 ; i--) {  
    if ( A[i] == i){ }  
    } A[2] = A [B[3]]; }
```

Base addresses of array A and B are in register X_{20} and X_{21} . Also consider i is in register X_{22} .

Addi $X_{22}, X_0, 8$
Loop:
Beq X_{22}, X_0, Exit

$A = X_{20}$
$B = X_{21}$
$i = X_{22}$

If :

```
Slli  $X_5, X_{22}, 3$  // offset calc.  
Add  $X_5, X_5, X_{20}$  // Base + offset  
Id  $X_6, 0(X_5)$  //  $X_6 = A[i]$   
Bne  $X_6, X_{22}, \text{inc}$ 
```

```
Id  $X_7, 24(X_{21})$  //  $X_7 = B[3]$   
Slli  $X_7, X_7, 3$  //  $X_7 = X_7 \times 2^3$   
Add  $X_7, X_7, X_{20}$  // Base + offset  
Id  $X_5, 0(X_7)$  //  $X_5 = A[B[3]]$   
Sd  $X_5, 16(X_{20})$  //  $A[2] = X_5$ 
```

inc :

```
Addi  $X_{22}, X_{22}, -1$   
Beq  $X_0, X_0, \text{Loop}$ 
```

Exit :

Question - 3:

Construct the equivalent RISC-V code of the following C code.

```
if ( A[i] < i){  
    A[2] = A [B[3]] ;  
}
```

Base addresses of array A and B are in register X_{20} and X_{21} . Also consider i is in register X_{22} .

If :

```
Slli x5, x22, 3      // offset calc.  
Add  x5, x5, x20      // Base + offset  
Ld   x6, 0(x5)       // x6 = A[x]  
BGE x6, x22, Exit
```

$$\begin{cases} A = X_{20} \\ B = X_{21} \\ i = X_{22} \end{cases}$$

```
Ld   x7, 24(x2)      // x7 = B[3]  
Slli x7, x7, 3        // x7 = x7 * 23  
Add  x7, x7, x20      // Base + offset  
Ld   x5, 0(x7)       // x5 = A[B[3]]  
Sd   x5, 16(x20)     // A[2] = x5
```

Exit :

Question - 4:

Construct the equivalent RISC-V code of the following C code.

2

```
if ( A[3] != A[6]) {  
    if (A[3] == 0) {  
        A[3] = A[3] + 2;  
    } else {  
        A[6] = A[6] / 16;  
    }  
} else {  
    A[6] = A[6] * 8  
}
```

Base addresses of array A and B are in register X_{20}

If 1:

ld x_{28} , 24(x_{20}) // $x_{28} = A[3]$

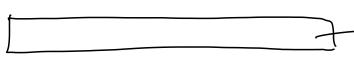
ld x_{29} , 48(x_{20}) // $x_{29} = A[6]$

Breq x_{28} , x_{29} , Else1

If 2:

ld x_5 , 24(x_{20}) // $x_5 = A[3]$

BNE x_5 , x_0 , Else2

 $A[3]$ already loaded into x_5 and x_5 is unchanged till now.

Addi x_5 , x_5 , 2

sd x_5 , 24(x_{20}) // $A[3] = x_5 + 2$

Breq x_0 , x_0 , Exit

Else 2:

ld x_5 , 48(x_{20}) // $x_5 = A[6]$

SRLI x_5 , x_5 , 4 // $x_5 = x_5 / 2^4$

sd x_5 , 48(x_{20}) // $A[6] = x_5$

Breq x_0 , x_0 , Exit

Else 1:

ld x_{28} , 48(x_{20}) // $x_{28} = A[6]$

SLLI x_{28} , x_{28} , 3 // $x_{28} = x_{28} \# 8$

sd x_{28} , 48(x_{20}) // $A[6] = x_{28}$

Exit:

Question - 5:

Translate the following code written in C programming language into instructions sequence written in RISC-V Assembly. The values in the variables a, b, and c inside the add function are stored in the argument registers \$X₁₀, \$X₁₅, and \$X₁₂ respectively. Also, the values in the variables x and y inside the are stored in the argument registers \$X₁₃ and \$X₁₄ respectively. The return value is stored in the return register \$X₁₁ for both functions.

x_{10} x_{15} x_{12}	x_{13} x_{14}
<pre>int max(int x, int y) { if(x > y) return x; else return y; }</pre>	<pre>int calc(int a, int b, int c) { return a + max(b,c) }</pre>

$x_{13} < x_{14}$

Calc:

Addi \$P, \$P, -8

sd \$1, 0(\$P)

Addi \$13, \$15, 0

$$[x_{13} = x_{15} + 0]$$

b

Addi \$14, \$12, 0

$$[x_{14} = x_{12} + 0]$$

c

Jal \$1, max

Add \$11, \$10, \$11
a return from max

ld \$1, 0(\$P)

Addi \$P, \$P, 8

Jalr \$0, 0(\$1)

Max:

bit \$13, \$14, maxElse

Addi \$11, \$13, 0

beq \$0, \$0, ExitMax

max Else:

Addi \$11, \$14, 0

ExitMax:

Jalr \$0, 0(\$1)

Ques - 6

Write RISC-V assembly code that checks if the number stored in register X_{25} is **even** or not. If **even** then store **1** in register X_{26} otherwise store **0**.

$$0000 \dots 0001 = 1$$

Addi $X_{27}, X_0, 1$

And $X_{27}, X_{25}, X_{27} \text{ } || \text{ } \text{LSB}$

BNE X_{27}, X_0, else bit required

Addi $X_{26}, X_0, 1$

Beq X_0, X_0, Exit

else:

Addi $X_{26}, X_0, 0$

Exit:

You can also use
Slli / Srahi to figure it
out.

Ques - 7

ADD X₂₅, X₂₅, X₀. Can you make this instruction faster? If yes, Write the updated instruction?

Ans, ADDI X₂₅, X₂₅, 0

0 is hardwired here.

Ques - 8

Memory Location	Code	Line Number	Machine Code																								
	ADDI X ₅ , X ₀ , 5	1																									
#7064	ADDI X ₆ , X ₀ , 1	2																									
#7068	ADDI X ₂₅ , X ₀ , 0	3																									
#7072	Loop: BLT X ₅ , X ₆ , loopBreak r ₀₁ r ₀₂	4 -3x4 = -12	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>000 000</td><td>00110</td><td>00101</td><td>XXX</td><td>1000</td><td>0</td><td>XXXXXXXX</td> </tr> <tr> <td>immn</td><td>immn</td><td>r₀₂</td><td>r₀₁</td><td></td><td>immn</td><td>immn</td><td></td> </tr> <tr> <td>12</td><td>[10:5]</td><td></td><td></td><td></td><td>14:0</td><td>D1</td><td></td> </tr> </table>	0	000 000	00110	00101	XXX	1000	0	XXXXXXXX	immn	immn	r ₀₂	r ₀₁		immn	immn		12	[10:5]				14:0	D1	
0	000 000	00110	00101	XXX	1000	0	XXXXXXXX																				
immn	immn	r ₀₂	r ₀₁		immn	immn																					
12	[10:5]				14:0	D1																					
#7076	✓ ADDI X ₂₅ , X ₂₅ , 1	5 ✓																									
#7080	✓ ADDI X ₅ , X ₅ , -1	6 ✓																									
	✓ BEQ X ₀ , X ₀ , Loop	7 ✓	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td><td>111 111</td><td>00000</td><td>00000</td><td>XXX</td><td>1010</td><td>1</td><td>XXXXXXXX</td> </tr> <tr> <td>12</td><td>[10:5]</td><td></td><td></td><td></td><td>9:1</td><td>11</td><td></td> </tr> </table>	1	111 111	00000	00000	XXX	1010	1	XXXXXXXX	12	[10:5]				9:1	11									
1	111 111	00000	00000	XXX	1010	1	XXXXXXXX																				
12	[10:5]				9:1	11																					
	loopBreak:	8 ✓ +4x4 = 16																									

- a) What is the value of PC while executing line2? Answer:

7064 [2]

- b) Fill up the machine codes corresponding to line4 and line7 in the table above. [5]

$$16 = 0000 \ 0001 \ 0000$$

$$+16 = 0 \ 0000 \ 0001 \ 0000$$

12 11 10 5 4 1 X
 | | | | | | |

$$12 = 0000 \ 0000 \ 1100$$

$$+12 = 0 \ 0000 \ 0000 \ 1100$$

$$= 1 \ 1111 \ 1111 \ 0011$$

$$+ 1$$

$$-12 = \overline{1 \ 1111 \ 1111 \ 1010 \ 0}$$

12 11 10 5 4 1 0X
 | | | | | | |

Ques - 9

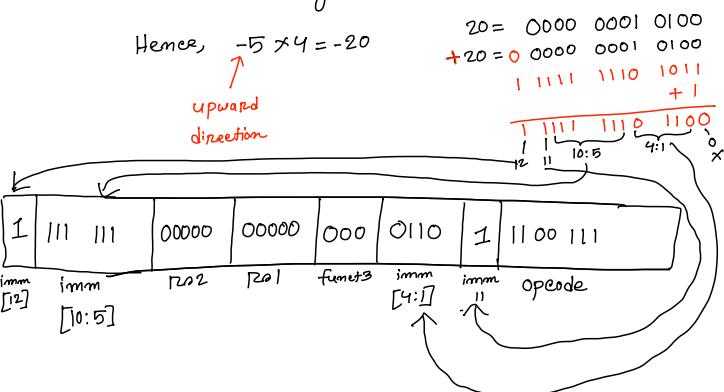
# Location		Line No.
	Loop :	
# 80000	Slli $x_{10}, x_{22}, 3$ ✓	1
# 80004	Add x_{10}, x_{10}, x_{25} ✓	2
# 80008	Ld $x_0, 0(x_{10})$ ✓	3
# 80012	Bne x_0, x_{24}, Exit ✓	4
# 80016	Addi $x_{22}, x_{22}, 1$ ✓	5
# 80020	Beq x_0, x_0, loop ✓	6
	Exit :	
# 80024	→ ✓	7

SB type instructions are - Beq x_0, x_0, loop
 Bne x_0, x_{24}, Exit

For Beq,

We are moving 5 instruction upward.

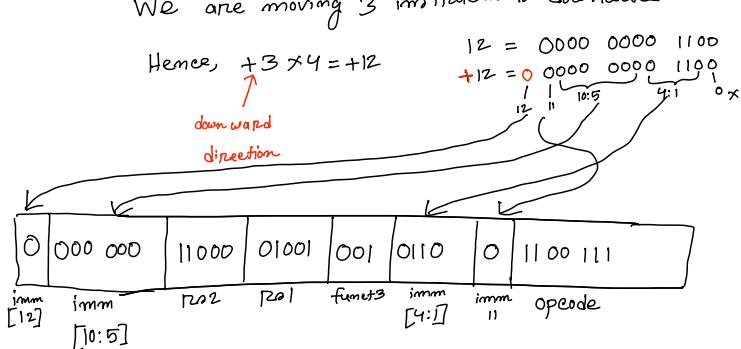
Hence, $-5 \times 4 = -20$



For Bne,

We are moving 3 instructions downward

Hence, $+3 \times 4 = +12$



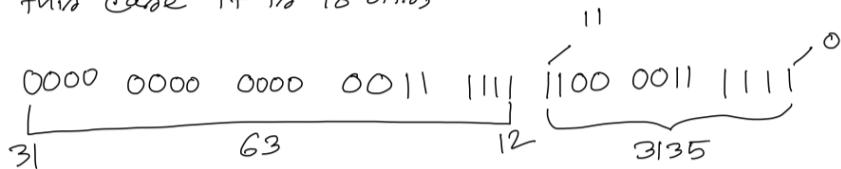
Ques - 10

Write necessary RISC-V instructions to store the value $(1111\ 1111\ 0000\ 1111\ 11)_2$ in X20 register.

Answer2:

Check if the given number requires more than 12 bits to store.

In this case it is 18 bits,



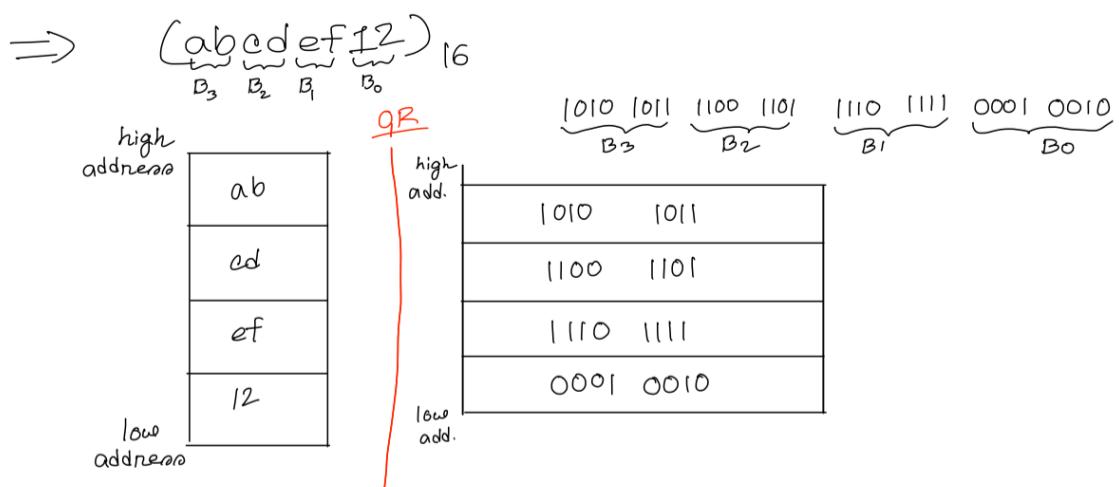
Lui X20, 63

Ques- 13

Show how the value 0xabcdef12 would be arranged in memory in RISC-V machine.

Answer3:

RISC-V follows Little endian approach to store data in memory



Queso - 12

For the RISC-V assembly instructions below, what is the corresponding C/high level statement?

$slli x30, x5, 3 \Rightarrow x_{30} = f \times 8$ $add x30, x_{10}, x_{30} \Rightarrow x_{30} = Base_A + 8f$ $slli x31, x6, 3 \Rightarrow x_{31} = g \times 8$ $add x31, x_{11}, x_{31} \Rightarrow x_{31} = Base_B + 8g$ $ld x5, 0(x30) \Rightarrow f = A[f]$ $addi x12, x30, 8 \Rightarrow x_{12} = x_{30} + 8$ $ld x30, 0(x12) \Rightarrow x_{30} = A[f+1] + A[f]$ $add x30, x30, x5 \Rightarrow x_{30} = A[f+1] + A[f]$ $sd x30, 0(x31) \Rightarrow B[g] = A[f+1] + A[f]$	Assume that the variables f, g, h, i, and j are assigned to registers x5, x6, x7, x28, and x29, respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively.
---	---

$$B[g] = A[f+1] + A[f]$$