

# Final - 2024

1

1 point

Given a cache with 256 blocks. How many bits are needed for the Cache reference?	8
Given each block is 1024 bytes. How many bits are needed for the offset in the block?	10
What is the Tag size in bits	<input type="text" value="8"/>
What is the <b>DATA</b> size of one block in <b>bits</b> ?	<input type="text"/>
How many total <b>bits</b> are needed for one block?	<input type="text"/>
For the full cache, how many bits are needed? (no commas)	<input type="text"/>

2

1 point

The memory address is 8632ACF5	
Given: Cache <b>Block size</b> (in bytes) is	8192
The Block Offset width (in bits) is	<input type="text"/>
Given: The TAG width (in bits) is	11
What is the <b>Cache Index</b> size in bits	<input type="text"/>
How many Blocks are there in Cache	<input type="text"/>
What is the TAG for this instruction?	<input type="text"/>
What is the INDEX for this instruction?	<input type="text"/>
What is the byte offset of the block	<input type="text"/>

The memory address is 68CD51F1

Given Cache **Block** size (in bytes) is

32768

The **Offset** width (in bits) is

Given The **Tag** width (in bits) is

8

What is the **Cache Index** size in bits

How many Blocks are there in the Cache?

What is the **TAG** for this instruction?

What is the **INDEX** for this instruction?

What is the byte **Offset** of the block

Given: The **cache** contains 4096 blocks  
Each **block** holds 4096 bytes

What is the size (in bits) of the **Tag**

Given this memory address.  
01111000001101011100101011001011

What is the tag

What is the Index

What is the Offset

What is the address in hex (all CAPS)

How many bits is the **Offset**?

How many bits are needed for the **Cache Index**?

5

1 point

Answer the following question about this cache configuration

<b>Tag</b> is 7 bits how many different tags will there be?	<input type="text"/>
<b>Index</b> is 18 bits how many different <b>cache</b> blocks are there?	<input type="text"/>
How many bits wide will the <b>offset</b> be?	<input type="text"/>
Given the above information how big is the <b>block size</b> (in <b>BITS</b> )?	<input type="text"/>

6

1 point

PC	Tag	Index	Address (Jersey #)	Value
0	Bulls	100	28	Obi Enechionya
1	Heat	011	35	Aric Holman
2	Magic	111	15	Sindarius Thornwell
3	Hawks	010	10	Jeremiah Martin
4	Jazz	011	11	Mike Conley
5	Magic	111	15	Rodney Purvis
6	76ers	001	65	John Petrucelli
7	Magic	000	40	Braian Angola-Rodas
8	Mavericks	111	23	Alessandro Pajola
9	76ers	101	13	R.J. Hampton
10	Mavericks	100	20	Rashad Vaughn
11	Heat	000	0	Yante Maten

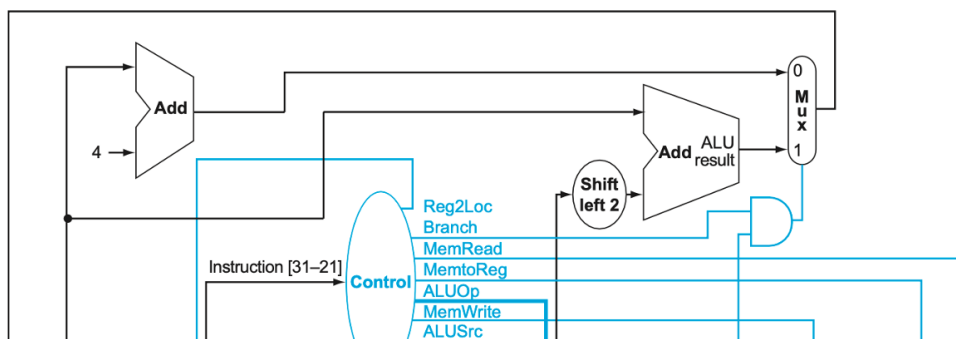
12	Jazz	011	19	Xavier Sneed
13	Hawks	000	32	Reid Travis
14	Heat	010	18	DeAndre Liggins
15	Magic	111	15	Sindarius Thornwell
16	Celtics	011	43	Javonte Green
17	Bulls	010	34	Justin Lewis

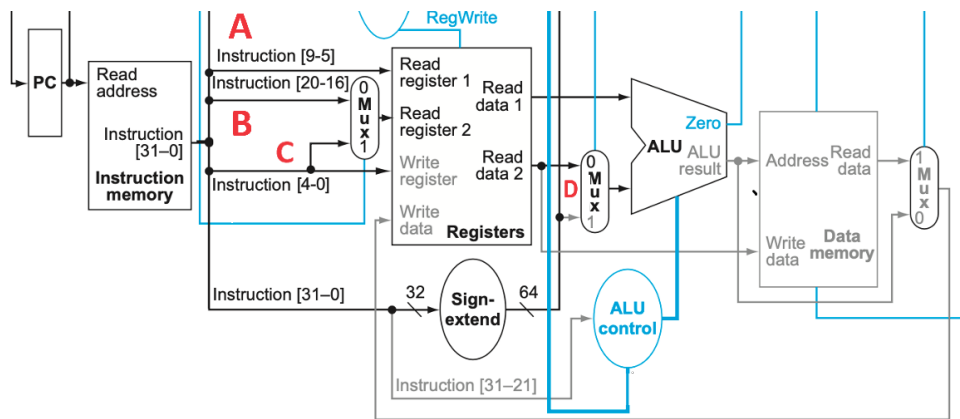
What will be the state of the cache after Fetching this 18 memory locations?

Tag	Index	Jersey #	State <u>H</u> it or <u>M</u> iss
	000		
	001		
	010		
	011		
	100		
	101		
	110		
	111		

7

1 point





Given this line of code, how will the Signal flags be set?

STUR

X3,

[X19, 96]

Signal Flags

Value

Reg2Loc

ALUSrc

ALUOp

ALU Control

MemRead

MemWrite

MemToReg

RegWrite

What Type of Instruction  
(R, I, CB, D)

D

Specify the Value used at the following checkpoints			
Checkpoint	Value		
A - Reg 1	<input type="text"/>		
B - Reg 2			
C - alt Reg 2			
D - Src 2, what value is passed to the ALU	<input type="text"/>		

8

1 point

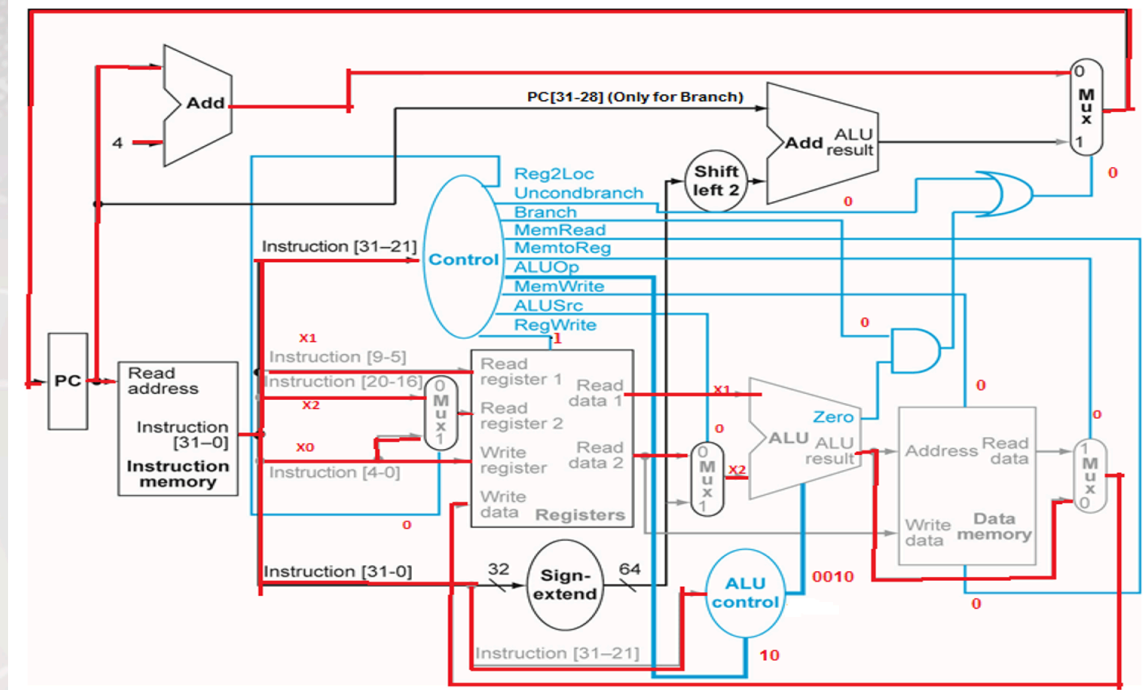
Look at the two slides below. One instruction is an Add, the other is a SUBI, one uses three registers the other uses two. Explain the difference in the data paths and the similarities.

Provide at least two differences - include the values of the Control Signals

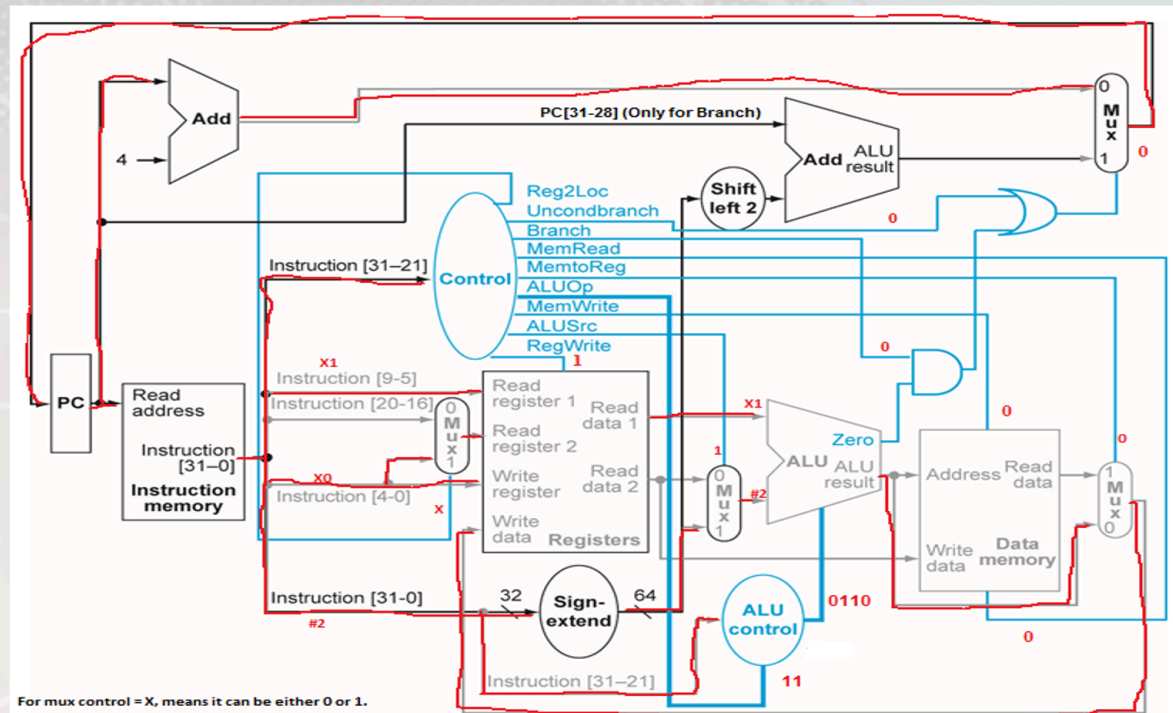
Provide at least four similarities - include the values of the Control Signals



## Tracing Datapath Ex: ADD X0, X1, X2



## Tracing Datapath Ex: SUBI X0, X1, #2



For mux control = X, means it can be either 0 or 1.

Edit View Insert Format Tools Table

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾  $T^2$  ▾ |

 ▾  ▾  ▾  ▾ |  ▾ |  ▾  ▾  ▾ |

  ▾  $\sqrt{x}$  

p

  |   

9

1 point

Which instructions have the signal RegWrite = 0

- ☐ STUR
- ☐ CBZ
- ☐ LDUR
- ☐ AND

10

1 point

Which instructions have the signal ALUSrc = 1

- ☐ STUR
- ☐ ANDI
- ☐ LDUR
- ☐ AND

11 1 point

Which instructions have the signal Reg2Loc = 1

- ☐ STUR
- ☐ ANDI
- ☐ LDUR
- ☐ AND
- ☐ CBZ

12 1 point

Match the Hazard with the cause and possible solution

Structure Cause



Control Cause



Data Cause



Possible answers

⋮ Instruction cannot execute because data that are needed to execute the instruction are not yet available.

⋮ When an instruction cannot execute because the hardware does not support the combination of instructions that are set to execute.

⋮ Occurs when the pipeline makes incorrect branch prediction decisions, resulting in instructions entering the pipeline that must be discarded.

13

1 point

Match the Hazard with the cause and possible solution

Structure Solution



Control Solution



Data Solution



Possible answers

⋮ Reorder lines of code if possible or add in some NOPs

⋮ Have two different kinds of memory

⋮ Rewrite code to take the 'false' path as frequently as possible

14

1 point

Match the Hazard with the cause and possible solution

Control Example



Data Example



Structure Example



Possible answers

⋮ A CBZ takes the 'true' path and skips over the immediately following instructions

⋮ One type of memory used for Instruction and Data - Fetch and Data access can not happen at the same time

⋮ An R-Type instruction (ADD X1, X2, X3) uses a register that an LDUR instruction has not retrieved the value from memory (LDUR X1, [X19, #24])

15

1 point

For each action listed on the left select the Stage in which it is performed

If the MemRead signal is set then the address from the ALU is used to retrieve the value from memory

The ALU executes the desired instruction (ADD, ORR, SUB,...)

Current PC is incremented by 4

Instruction Memory is accessed to retrieve the next instruction

If the MemWrite signal is set then the value from 'Read data 2' is saved to memory

Registers to be used are identified and values are retrieved from the Register File

If the RegWrite signal is set then the value passed by the MemoryMux is saved to the desired Reg

The ALU calculates the address of the memory value to be used in the MEM stage

Possible answers

⋮ ID

⋮ EX

⋮ MEM

⋮ IF

⋮ WB

16

1 point

Convert Decimal to 32-bit FP

For example:

Decimal #	Sign bit	Exponent	Fraction
-4.75	1	10000001	0011
10.125000	<input type="text"/>	10000010	<input type="text"/>
43.015625	<input type="text"/>	10000100	<input type="text"/>
86.031250	<input type="text"/>	<input type="text"/>	01011000001
-98.17187500	<input type="text"/>	<input type="text"/>	<input type="text"/>

17

1 point

Convert to FP binary

-0.390625

Convert Integer to Binary

Convert Fraction to Binary



What is the sign bit?

How many places did you need to move the decimal?

<input type="text"/>	Positive to the left. Negative to the right
What is the decimal value of the Bias	
<input type="text"/>	
What is the binary value of the Bias? 8 bits	
<input type="text"/>	
What then is the significant (digits to the right of the decimal) No trailing 0s	
<input type="text"/>	
What is the full binary value of the FP # 1 bit for the sign 8 bits for the exponent Remaining bits for the fraction. No trailing 0s for the fractional portion	
<input type="text"/>	<input type="text"/>
<input type="text"/>	
What is the HEX value of the number (all CAPS)	
<input type="text"/>	



Convert to FP binary. For the Fractional portion do NOT include the trailing 0s

413.031250

Convert Integer to binary

Convert Fraction to binary (no trailing 0s)

00001

What is the sign bit?

How many places did you need to move the decimal?

Positive to the left. Negative to the right

What is the decimal value of the Bias

What is the binary value of the Bias

What then is the significant (digits to the right of the decimal)  
No trailing 0s

1001110100001

What is the full binary value of the FP # 1 bit, 8 bits, No trailing 0s for the fractional portion

What is the Hex value of the number ALL CAPS

19

1 point

Convert to FP binary

No not include the Trailing 0s of the Fractional portion

37.15625

Convert Integer to binary

Convert Fraction to binary. No trailing 0s

What is the sign bit?

How much did you need to move the decimal?

Remember: Pos to the left. Neg to the right

What is the decimal value of the Bias

What is the binary value of the Bias

What then is the significant (digits to the right of the decimal). No not include trailing 0

What is the full binary value of the FP #

Sign

Exponent

Significand

What would be the value of the number in HEX.. Use all caps. For this you will need to add in the trailing zeros to make a 32 bit number.

20

1 point

Convert to FP binary

Do not include Trailing 0s in the Fractional portion

27.359375

Integer

Fraction

010111

What is the sign bit?

How much did you need to move the decimal?  
Remember: Pos to the left. Neg to the right

What is the decimal value of the Bias

What is the binary value of the Bias

What then is the significant (digits to the right of the decimal)

1011010111

What is the full binary value of the FP #. No trailing 0s

Sign

Exponent

Significant

What would the HEX value of the number? In ALL CAPS. This will be an 8 digit HEX number

21 1 point

Associate the line of code with the category

Triple the value of X1

Multiply by 33

Create the 2s complement of register X3

Integer divide by 8

Possible answers

⋮ ADD X3,X3,X3,LSL 5

⋮ MULI X1,X1, 3

⋮ Multiply X3 by 16

⋮ Calculate the Reminder

⋮ LSR X3,X3,#3

⋮ MULI X3,X3,#-1

⋮ XORI X3,X3,#-1. followed by ADDI X3,X3,#1

⋮ DIVI X3,X3,#8

⋮ ADD X1,X1,X1,LSL 1

⋮ MULI X3,X3, 33

22 1 point

Below is code for a simple looping program. The table below contains the Machine Code. You need to decipher the code and fill in the table with the values of the different cells. If a field is all zeros include ALL zeros.

		loop	
80001238	F10005DF	CMP	x14, #1
8000123C	540001AD	B.LE	Exit ; 0x80001270
80001240	910021AD	ADD	x13, x13, #8
80001244	F84001A9	LDUR	x9, [x13, #0]
1. 80001248	D10005CE	SUB	x14, x14, #1
8000124C	8B09014A	ADD	x10, x10, x9
80001250	EB09017F	CMP	x11, x9
80001254	5400006A	B.GE	check_smallest ; 0x80001260
80001258	AA0903EB	MOV	x11, x9
8000125C	17FFFFF7	B	loop ; 0x80001238

•

R Format						
Opcode - 11 bits	Rm - 5 bits	Shamt - 6 bits	Rn - 5 bits	Rd - 5 bits	Operation	Hex Instruction value
				01010	ADD X10, X10, X9	8B09014A
		000000	11111		ORR X11, XZR, X9	AA0903EB
CB Format						
Opcode - 8 bits	Address - 19 bits			Rt - 5		
					B.GE CheckSmallest	5400006A
				01101	B.LE Exit	540001AD
I Format						
Opcode - 10 bits	Immediate - 12 bits		Rn - 5	Rd - 5		
			01101		ADDI X13, X13, #8	910021AD
D Format						
Opcode - 11	Address - 9	Op2 - 2	Rn - 5	Rt - 5		
		00			LDUR X9, [X13, #0]	F84001A9

•

23 1 point

List 4 different kinds of **I - Type** instructions.  
Include any registers, addresses or immediate values needed

Edit View Insert Format Tools Table

12pt Paragraph | **B** *I* U A ∨  ∨  $\text{T}^2$  ∨ |

 ∨  ∨  ∨  ∨ |  ∨ |  ∨  ∨  ∨ |

  ∨  $\sqrt{x}$  

p

  |   

24 1 point

Given this line of Code (which is an R Format) what are the values for the respective fields

	opcode	Rm	shamt (enter all 6 characters to fill the field)	Rn	Rd	Hex Value
AND X11, X9,X13						



25

1 point

Given the Cycles per Instruction Type. Determine The CPI per Instruction Type and the Weighted CPI. All answers should have TWO decimal places. Even if the second decimal is 0. If the number is less than 1 include the leading 0.

Examples 2.00 1.50 1.66, 0.10, 0.25

Instruction Type	Add	Store	Branch	FP	
Cycles per Instruction Type	1	12	5	2	
Instruction Type Mix	40%	20%	15%	25%	
CPI per Instruction Type	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Weighted CPI <input type="text"/>



Order chips from slowest to fastest

Slowest

☐

⋮ 2.0 GHz cycles per second

☐

⋮ 1 ns seconds per cycle

☐

⋮ 303 ns seconds per cycle

☐

⋮  $20 \times 10^{-15}$  seconds per cycle

☐

⋮ 4.0 Ghz cycles per second

☐

⋮  $400 \times 10^{-12}$  seconds per cycle

☐

⋮  $1 \times 10^{-15}$  seconds per cycle

fastest

27

1 point

Application A on Computer A	
CPU Time (secs)	20.00
Cycles	70.00E+9
Cycle Rate (cycles / sec)	3.50E+9
We have created a new chip and have tested Application A. It runs in 12 secs and requires an additional 20% in instruction cycles	
Application A on Computer B	
CPU Time (secs)	12.00
Cycles (enter whole number no decimals)	<input type="text"/> x $10^9$
Cycle Rate (enter whole number no decimals)	<input type="text"/> x $10^9$

28

1 point

Given three pieces of information about the application parameters calculate the missing piece. If number is less than 1 include the leading 0. Only include the significant fractional portion, no trailing zeros. Do not include commas or spaces. 0.5, 1.55, 2, 1.7

CPI	# of Instructions	Clock Rate Cycles/Sec	CPU Time
2.25	200000000	900000000	0.5
3	100000000	1000000000	<input type="text"/>
1.5	100000000	<input type="text"/>	0.3
1.5	125000000	<input type="text"/>	0.75
3.5	<input type="text"/>	350 000 000	1.25
4	<input type="text"/>	5 000 000	120
<input type="text"/>	150 000 000	5 000 000	66
<input type="text"/>	12 000	240 000	0.3

29 1 point

Compare Performance of Chip X to Chip Z

	Chip X	Chip Z
Instruction Cycles	27 000 000 000	<input type="text"/>
Cycle Rate (Cycles/Sec)	4.2 GHz	3.5 GHz
If the new chip design for Computer Z <b>decreases</b> instruction cycles by 20% and has a cycle time of 3.5 GHz,		
Calculate the run time on Computer X. Only include <b>two decimals</b> .		<input type="text"/> Sec
Calculate the total cycles on the new chip Z. <b>No commas or spaces</b>		<input type="text"/>
Calculate the run time on the new chip Z. Only include <b>two decimals</b> .		<input type="text"/> Secs
Which chip ran the application faster? (X or Z)		<input type="text"/>

30 1 point



The cells in Yellow are steps along the way which you will need to solve for the Green cells.

Do NOT include commas or spaces, If the answer does not have a decimal portion do not include it. DO include the leading zero is answer is less than 1

BAD BAD BAD

12,000,000 12.0 12.10 12 000 000 .4 0.40

GOOD

12000000 12 12.1 12000000 0.4 0.4

Processor B runs App X

		Run Time	0.0002 secs		
# of Cycles	# of Instructions	Instr Type	Cycles per Instructions	% of Instructions	CPI for Instruction Type
<input type="text"/>	500,000	Add/Mov Instr	1	<input type="text"/>	<input type="text"/>
<input type="text"/>	250,000	Branch	2	<input type="text"/>	<input type="text"/>
<input type="text"/>	500,000	Multiply	3	0.4	1.2
Total Cycles <input type="text"/>	Total # of Instructions <input type="text"/>				Average CPI <input type="text"/>
	Clock Rate <input type="text"/> GHz	Clock Rate (cycles / sec)			
	Cycle Time (Sec/Cyc) <input type="text"/> x 10 <sup>-12</sup>	Clock Cycle Time (sec / cycle)			