HONOR CODE

- I have not used any online resources during the exam.
- I have not obtained any help either from anyone in the class or outside when completing this exam.
- No sharing of notes/slides/textbook between students.
- **CANVAS ANSWERS MAY BE LOCKED AFTER 1ST TRY.**

Questions Sheet.

Read all of the following information before starting the exam:

- For each question fill out the appropriate choice or write text on page. Also type clearly on in the exam on the appropriate text.
- IF THE MULTIPLE CHOICE ANSWER IS WRONG WE WILL MARK THE ANSWER WRONG. IF THE MULTIPLE-CHOICE ANSWER IS CORRECT, WE WILL READ THE WRITTEN PORTION.
- 1 pt Qs (0 or 1). 2 or 3pt Qs (if no explaination only 1 pt.)
- Show all work, clearly and in order, if you want to get full credit.
- We reserve the right to take off points if we cannot see how you logically got to the answer (even if your final answer is correct). 1 or 2 sentences atmost.
- I will take points off for rambling and for incorrect or irrelevant statements.

        - 53. What hazards existing between line 3 and 4 ? (1)
        - 54. What hazards existing between line 4 and 5 ? (1)
        - 55. What is the instruction in IF stage t = 6 ? line ____ (1)
        - 56. What is the instruction in IF stage of t = 7 ? line ____ (1)
        - 57. How many cycles does this program take to complete ? ______ (2)
    - Part 2.
        - 58. What is the instruction in IF stage t = 6 ? line _____ (1)
        - 59. What is the instruction in IF stage of t = 7 ? line _____ (1)
        - 60. How many cycles does this program take to complete ? ______ (2)
    - Part 3.
        - 61. What is the instruction in IF stage t = 6 ? (1)
        - 62. What is the instruction in IF stage of t = 7 ? (1)
        - 63. How many cycles does this program take to complete ? ______ (2)
    - Part 4.
        - 64. What is the instruction in IF stage t = 6 ? (1)
        - 65. How many cycles does this program take to complete ? ______ (2)
- F. Pipeline CPU.
    - 66. Assuming that this CPU is NOT Pipelined (i.e. it is a singlecycle CPU), what is the
    - 67. Assuming that this CPU is Pipelined, what is the

Section A Virtual Memory 15 points. Canvas Q1-Q28

```
Refer slide deck L21-VM-III Week 8 if you need to.
```

The chart below shows how memory accesses are treated in a system. The table below describes the parameters int he memory system.
Please use the data below to answer question groups Q1,Q2,Q3,Q4 on canvas.

```
CAUTION 1: When converting from binary to hex you can always pad the MSB
e.g., 10 1010 (6 bit field) in hex is 0010 1010 (2 0s padded in MSB)
is 0x2a .
```

```
always use lower case and prefix 0x for hex .
Correct: 0xabcdef Incorrect: 0xABCDEF
```

Flowchart

Parameter	Value
Physical address bits	18
Size of page	64 bytes
Virtual address bits	18
-------------------------	---------------------
TLB Sets	8
TLB Ways	2
-----------------------	-------------------
Cache size	256 bytes
Cache Sets	16
Cache Ways	2

- VPN - Virtual page number
- Index (Set index of cache or TLB)
- PPN - Physical page number
- INVALID. TLB entry is invalid
- TLB-T (TLB Tag)

TLB

Way 0	TLB-T	PPN
Set 0	[0x1cb]	0x958
Set 1	[0x1be]	---
Set 2	[0xa4]	0xe76
Set 3	[0x74]	----
Set 4	[0x11d]	0x2b5
Set 5	[0xbc]	0xa36
Set 6	[0x1fb]	---
Set 7	[0xf1]	0xfea

Way 0	TLB-T	PPN
Set 0	[0x177]	0x47b
Set 1	[0x24]	---
Set 2	[0x12]	0x84a
Set 3	[0x16c]	---
Set 4	[0x192]	0xe76
Set 5	[0xd1]	---
Set 6	[0xab]	0xec3
Set 7	[0xc6]	---

Page Table (Partial)

```
CAUTION: Only partial table relevant to the questions are shown.
```

VPN	PPN	Valid
0xbb8	0x47b	1
0xe58	0x958	1
0x522	e76	1
0xdf1	---	0
0xfde	---	0
0x78f	fea	1
0x383	0xa3	1
0x68d	---	0
0x565	0x636	1

Cache

Way 0	Tag	0	1	2	3	4	5	6	7
Set 0	[0x322]	0x13	0xc8	0xce	0xc6	0x76	0x7c	0xde	0xf3
Set 1	[0x45a]	0xbc	0x0f	0xd8	0x93	0xe7	0x69	0xea	0xb1
Set 2	[0x6f6]	0x56	0xf3	0x3d	0xd6	0x7b	0x2c	0x98	0x34
Set 3	[0x276]	0xf3	0x8c	0x94	0x8d	0xaf	0x5c	0x02	0x3b
Set 4	[0x23c]	0x3a	0x09	0x3c	0x3b	0x35	0x3f	0x85	0x1e
Set 5	[0x425]	0x19	0x65	0xaf	0x3e	0xb4	0x8a	0x8a	0xcf
Set 6	[0x62e]	0x88	0x87	0xad	0x73	0xac	0x70	0xeb	0x77
Set 7	[0x4ac]	0x0c	0x02	0x16	0xfe	0x7b	0x34	0xd6	0x91
Set 8	[0x761]	0xc9	0x97	0x01	0x6d	0xea	0x55	0x59	0x73
Set 9	[0x242]	0xe2	0x55	0x38	0xd0	0x84	0x6c	0x16	0x5b
Set 10	[0x494]	0x76	0x78	0x19	0x6b	0xb6	0xf3	0xa4	0xfb
Set 11	[0x23d]	0xf5	0xac	0x7f	0xf5	0x1f	0xb8	0x03	0x20
Set 12	[0x7b9]	0x11	0x9a	0xd4	0x8d	0x85	0xe8	0xb3	0xb1
Set 13	[0x15a]	0x4e	0xaa	0x51	0x0f	0x61	0xc3	0x8f	0x0e
Set 14	[0x739]	0x77	0x71	0x3b	0xb8	0xa7	0x70	0x18	0x15
Set 15	[0x627]	0x7f	0x65	0xe0	0x34	0x1a	0x90	0xb5	0x19

Way 1	Tag	0	1	2	3	4	5	6	7
Set 0	[0x7f5]	0x11	0xbf	0xe8	0x3e	0xad	0x26	0x2e	0xaa
Set 1	[0x73b]	0xe3	0x08	0x0b	0x3a	0xc6	0x98	0x67	0x17
Set 2	[0x31b]	0x39	0x16	0x2e	0xbc	0xde	0x90	0xb5	0x61
Set 3	[0x755]	0x2b	0x39	0x46	0xf5	0x95	0xb7	0x43	0x6d
Set 4	[0x627]	0x51	0x8f	0x28	0xb4	0x1d	0xac	0x8c	0x3d
Set 5	[0x2af]	0xf5	0x49	0x67	0x1d	0xcb	0x75	0x8e	0x05

Way 1	Tag	0	1	2	3	4	5	6	7
Set 6	[0xf2]	0xc1	0xcd	0x99	0xce	0x27	0xb1	0x9b	0xaf
Set 7	[0x236]	0xbb	0xe4	0xda	0xcd	0x43	0xa6	0xa0	0x11
Set 8	[0x1c1]	0x02	0x29	0xe3	0x33	0xa9	0x24	0x89	0x1e
Set 9	[0x335]	0x32	0x44	0x68	0x76	0x7b	0xfb	0x7d	0xc9
Set 10	[0x64]	0x3c	0x84	0x70	0x27	0x98	0x88	0x96	0x73
Set 11	[0x51]	0x50	0x82	0x6c	0xda	0x93	0x3a	0x77	0x8b
Set 12	[0x56f]	0x9c	0xd1	0xc7	0xba	0x62	0xde	0x1e	0x37
Set 13	[0x654]	0xd5	0x9c	0x66	0x8f	0x95	0xa6	0x3f	0x4c
Set 14	[0x1e4]	0xc9	0x96	0xb0	0x3b	0xfb	0x76	0xa3	0x77
Set 15	[0x5d8]	0xb3	0x4f	0x91	0xe9	0x6e	0xa5	0x91	0x8a

Common questions. Canvas Q1-Q2

1. How many bits is the VPN. decimal (1pt) ?

12

2. How many bits is the PPN. decimal (1pt) ?

12

For the virtual address 0x15957 answer the following Canvas Q3-Q12. All in hex. (0.5 pt each)

- **What is the VPN . 0x15957**

  |-----VPN (12 bits)------|-----Page Offset (6 bits)-----| 0x15957 = 010101100101 (VPN) 010111 (offset)

010101100101 = 0x565

What is the TLB tag. 0x0ac (since 9 bits)

VPN = |----TLB Tag----|-----TLB Index----| The system specifies 8 sets for the TLB; therefore 3 bits for the index, as log8 = 3)

010101100101 (VPN) -> 00010101100 (TLB Tag) 101 (TLB Index)

is split:Tag: Higher bits (10101100 = 0x0ac in hex).

- **Is it a TLB hit or miss**

Miss. TLB Set is 101 (5). Tags in set 5 do not match 0xac.

is it a page fault No. Page table lookup using VPN (0x565-> 0x636)

- **What is the PPN ?** |---PPN----|----Page offset (from VPN question above) ---|

  Page table lookup using VPN (0x565-> 0x636)

  Physical page address (| PPN | offset |) -> | 0110 0011 0110 | 010111 | -> 0x18d97. Used for calculateing next set of cache operations.

- **what is the cache tag ? 0x31B (Ask yourself? How many bits in cache tag)**

  PPA = 01100011011 0010 111 -> | Cache Tag (Remaining) | Cache Set (4bits cause cache has 16 sets) | Block offset (3 bits cause block size is 8 bytes)

  01100011011 (Cache tag) 0010 (cache index) 111 (cache offset)

- what is the cache index

0x2

- What is the byte offset

0x7 0x31b found in cache set (or row 2). Read 7th byte from data array

- Is it a cache hit or miss

Hit. See Row/Set id = 2 in the cache and try to find tag 0x31B.

- What is the data byte

0x61

For the virtual address 0x2ee19 answer the following. Canvas Q13-Q22. All in hex (0.5 pt each)

- What is the VPN

0xbb8

- What is the TLB tag.

0x177

- Is it a TLB hit or miss

Hit

- Is it a page fault

No

- What is the PPN ?

0x47b

- what is the cache tag ?

0x23d

- what is the cache index

0xb

- What is the byte offset

0x1

- Is it a cache hit or miss

Hit

- What is the data byte

0xac

For the virtual address 0x1a344 answer the following. All in hex (Canvas: 23-28) (0.5 pt each)

00110 1000 1101 000100

- What is the VPN
  0x68d
- What is the TLB tag.
  0xd1
- Is it a TLB hit or miss
  hit
- Is it a page fault
  yes
- What is the PPN ?
  N/A
- Is it a cache hit ?
  No

B. Easy. RISCV Blackbox. [10 Points]

WARNING: FOR THIS QUESTION MAKE SURE YOU WRITE YOUR CORRECT ANSWER AT THE BEGINNING.

ALSO WRITE ONE/TWO SENTENCES REASONING YOUR ANSWER. OTHERWISE WE WILL SIMPLY ZERO IT OUT.

Assume we have two arrays input and result.

a0=message, a1=result . Answer questions below.

```
char *message = "ABCDE"
char result[20];
```

Answer questions below for the code shown.

```
  lbu   s0 0(a0)
  slli s1 s0 8
  add   s1 s1 s0
  sh    s1 0(a1)
```

29. _____ bytes are stored to memory? [1]

2

30. The string result is "_______" ? [1]

AA

Load the First Byte of message: The program reads the first byte (lbu a1, 0(a0)) of the string pointed to by a0. This is the ASCII value of the first character in message.

Shift and Duplicate the Byte:
Left Shift (slli): The loaded byte is shifted left by 8 bits. This moves the original value to the upper 8 bits of a register while leaving the lower 8 bits as zero.
Concatenate: The original byte is concatenated with its shifted version. This duplication effectively places the byte twice in a 16-bit word:
[Original Byte | Original Byte]

Answer questions below for the code

```
BLACKBOX:
  lbu   s0 0(a0)
  slli s1 s0 8
  addi s0 s0 1
  add   s1 s1 s0
  slli s2 s1 16
  addi s2 s2 s0
  sw    s2 0(a1)
```

31. _____ bytes are stored to memory? [1]

4

32. The string result is "_______" ? [2]

Program Analysis
lbu s0, 0(a0)
slli s1, s0, 8   s1 = 'A'

s1 = [s0 << 8 | 0x00]
addi s0, s0, 1   s0 = 'B'

add s1, s1, s0 s1 = 'AB'

This concatenates the shifted byte (s1) with the incremented original byte:
s1 = [Original Byte << 8]
slli s2, s1, 16 .   s2 = 'AB00' (note that 0x0 in hex is \0 in char)

addi s2, s2, s0 s2 = `AB\0B'

sw s2, 0(a1)

Store the value in s2 at the address pointed to by a1.

AB\0B

Answer questions below for the code shown.

33. _____ bytes are stored to memory? [2]

```
.text
BLACKBOX:
  lbu  s0 0(a0)
  beq  s0 x0 End
  slli s1 s0 8
  add  s1 s1 s0
  sh   s1 0(a1)
  sb   s0 2(a1)
  addi a0 a0 1
  addi a1 a1 3
j BLACKBOX
end:
  sb x0 0(a1)
```

16 (5 letters × 3 + null character)

34. The string result is _______ ? [3]

"AAABBBCCCDDDEEE"

C. Lets Cache I (14pts)

Assume we are working in a 4GB physical address space.
We will be studying the behavior of different loops on two different caches.

Cache-A	Direct-mapped, 4KB, 64 sets
Cache-B	Set-associate, 4KB, 2 ways, 32 sets

Both caches use write-back and write-allocate policies.
Hit ratio is written as #Hits:#Accesses
Hit-Miss ratio is: #Hits:#Misses
Hit % : #Hits:#Accesses * 100.

Answer questions below

```
int size = 4096;
// int is 4 bytes
int a[size];
// long long int is 8 bytes
long long int a_long[size];

/* loop 1 */
for (int i = 0; i < size; i++) {
  a[i] = i;
}
/* loop 2 */
for (int i = 0; i < size; i++) {
  a_long[i] = i;
}

/* loop 3 */
for (int i = 0; i < size/2; i += 1) {
  a[(size/2+i] = a[i];
}
```

35. The hit ratio of loop 1 for Cache A is ---- ? (1 pts)

Write it as Hits:Accesses

15:16 It's because in the first iteration you access a[0] and a[2048], both of which will be misses but will go to different blocks in the same set (for the two-way cache). Then the next 15 iterations you access a[1-15] and a[2049-2063] which all are hits since they belong to the same blocks as a[0] / a[2048]. So each 16 iterations you have two misses followed by 30 hits. Remember that each block is 64 bytes so it can hold 16 ints.

36. The hit ratio of loop 1 for Cache B is ---- ? (1 pts)

15:16

37. The hit ratio of loop 2 for Cache A is ---- (assume loop 1 has already run) ? (1 pts)

7:8

38. The hit ratio of loop 2 for Cache B is ---- (assume loop 1 has already run) ? (1 pts)

7:8

39. The hit ratio of loop 3 for Cache A is ---- (assume loop 1 and 2 has already run) ? (2 pts)

0

MM. each array element is long long int (8 bytes). So element 2048 is 2048*8=16k away from element 0. So all index bits are the same.

40. The hit ratio of loop 3 for Cache B is ---- (assume loop 1 and 2 has already run) ? (2 pts)

15:16

MMHHHH(repeats HH 15 times). Cache A is direct-mapped so you have conflict misses while cache B is 2-way set associative so you avoid these conflict misses. More explanation (see in comments bar or adobe pdf reader)

41. What is the hit ratio if loop 3 changes to the one below (on Cache A) ? (2 pts)

```
 1  /* loop 3 */
 2  for (int i = 0; i < size/2; i += 4) {
 3  int tmp0 = a_int[i+0]
 4  int tmp1 = a_int[i+1]
 5  int tmp2 = a_int[i+2]
 6  int tmp3 = a_int[i+3]
 7  a_int[size/2+i] = tmp0
 8  a_int[size/2+(i+1)] = tmp1
 9  a_int[size/2+(i+2)] = tmp2
10  a_int[size/2+(i+3)] =
11  tmp3 }
```

3/4

MHHHMHHH

The array accesses in lines 7-10 should have been corrected to be the same as four iterations of the original loop 3, so line 7 should be accessing a[size/2+i+0], line 8 should be accessing a[size/2+i+1], etc.

In this case, in the first iteration of the new loop 3, line 3 (a[0]) is a miss followed by three hits for a[1], a[2] and a[3] (same cache block as a[0]).

Line 7 (a[2048] is a miss because it has a conflict with a[0] (similar to the original loop 3 in question 39), then a[2049], a[2050], a[2051] will be hits.

Q: Loop3 : if a[2048] is a miss because it has conflict with a[0] would a[2049], a[2050], and a[2051] not have the same issues as well?

When you access a[2048] on line 7, you would have a conflict with a[0] and evict it to bring the block containing a[2048]. But the next three lines of code (8-10) will access a[2049], a[2050], a[2051] which all belong to the same block as a[2048] so they will be hits. Remember that a cache block here has 16 ints so it all of these elements belong to the same block.

42. What is the hit ratio if loop 3 is same as Q41 for lines 7-10. (on Cache B) ?(2 pts)

15:16 1st iteration MHHHMHHH 2nd interation HHHH HHHH ... 5th iteration MHHHMHHHH

43. What is hit ratio for the code below. Assume Cache B ? (2 pts)

```
struct int_and_long {
  int a_int;
  long long int a_long;
};
int size = 4096;
// int is 4 byte aligned
// long long int is 8 byte aligned

/* loop 1 */
for (int i = 0; i < size/2; i += 1) {
 s[size/2+i].a_int = s[i].a_int
}
```

Struct has to add 4 byte padding to a_int so total size of struct is 16 bytes.

Each cache line in A can hold 4 structs or 4 a_ints.

3:4 hits. e.g, s[0] and s[4095]

M,M,H,H,H,H,H,H. Each 4 structs that start with an index divisible by 4 are in the same block. So s[0-3] are in one block, s[4-7], s[8-11] etc.So s[4095] will bring in s[4092]-s[4095]. Not s[4095]-s[4098].1st iteration: s[4095] Miss, rings in s[4092-4095], s[0] Miss, brings in s[0-3]. 2nd iteration: s[4094] Hit, s[1] Hit3rd: s[4093] Hit, s[2] Hit4th: s[4092] Hit, s[3] Hit The same pattern repeats every 4 iterations.

D. RISC-V Single Cycle Datapath [15 points]

We wish to introduce a new instruction into our RISC-V datapath.

`badd` (branch and add). This instruction is a conditional add instruction. The instruction works as follows. The instruction is encoded as follows.

Add instruction						Opcode
f7	rs2	rs1	0x0	f3	rd	0110011

new badd instruction

0x10	rs2	rs1	0x0	f3	rd	0001011

```
if (R[rs1] > R[rs2]) {
  R[rd]++
} else  {
 R[rd] = R[rd]
}
```

It combines the semantics of branch and R-type instruction.

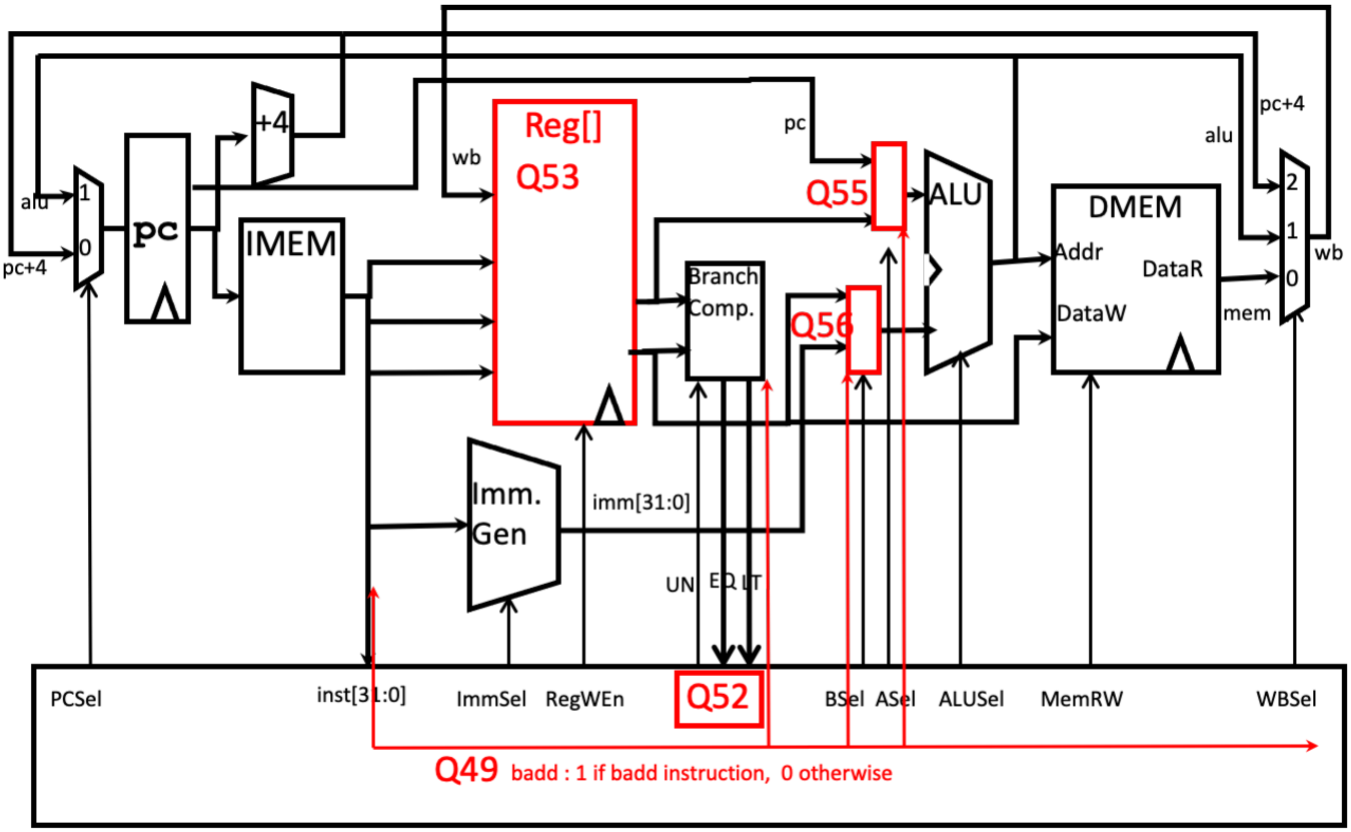
- Like a branch it performs a comparison.
- If comparison succeeds it increments rd. rd is destination and source.
- If comparison fails, it simply retains the value in R[rd].
- We have a new control signal `badd` which is 1 if the instruction being decoded is a badd.

Given the single cycle datapath below, select the correct modifications in parts such that the datapath executes correctly for this new instruction (and all other instructions!). You can make the following assumptions:

```
Caution 2: Pay careful attention to which input line is 1 and which line is 0 in the muxes.
Some muxes choose top-most input as 0, some choose bottom-most input as 0
```

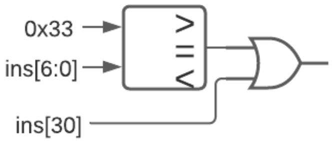
```
Hint: YOU DO NOT REQUIRE TRUTH TABLES
Try writing down in plain english or reading out the logic
to yourself e.g, !(A<=B) is A is not equal to B and A is not LT (less than) B
```

Pipeline with badd (Red boxes indicate questions)

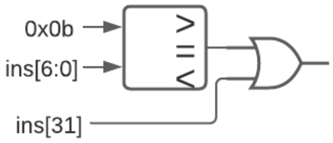


44. What is the logic for badd signal ? (1)

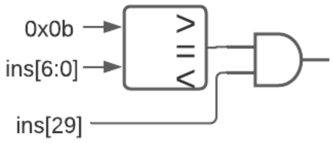
(a)



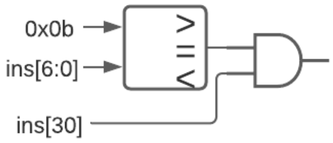
(b)



(c)



(c)



C ins[6:0] == 0x0b && ins[29] == 0x1, The instruction encoding given in the question was that the opcode (bits 0-6) is 0001011 (i.e, 0x0b) and that func7 bits (bits 25-31) are 0x10 = 0010000 (bit 29 is 1 while bits 30-31 and 25-28 are zeros). So in order to determine that the instruction is a "badd" instruction,

45. _____ is the number of registers badd needs to access in a single cycle ?(1)

3. You need to read rs1, rs2 and rd. You also need to write rd.

46. What is the RegWen signal for badd ? (1)

0 You need to look at Q46m 48 together. The correct answer for Q48 is d. Which means that if you set RegWen to 1 then you will always write to Rd even when you don't need to (since the output of the OR gate will always be 1). But if you set RegWen to 0 then you only write to Rd if (badd and !EQ and !LT) is true

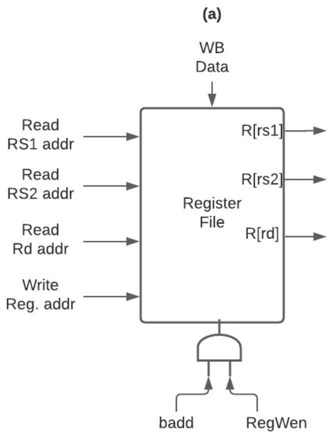
For some of these questions, you have to consider two signals together since the answer to one signal may affect the answer to another. We'll try to point this out in the questions, but you should try to think about how the whole datapath works together.

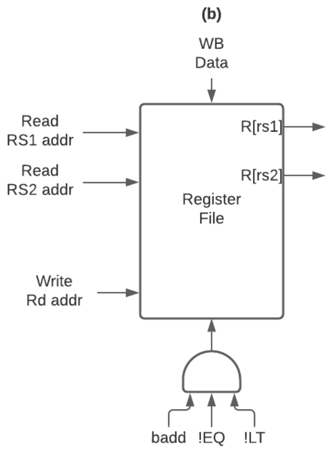
47. What is the comparison logic we are interested in ? (2)

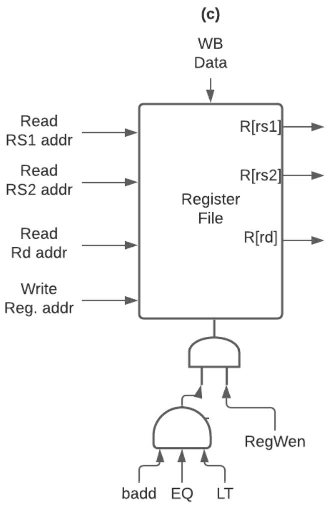
!(EQ OR LT) The comparison checks if rs1 > rs2. This logic is equivalent to !(EQ OR LT) because EQ means rs1 == rs2 and LT means rs1 < rs2.

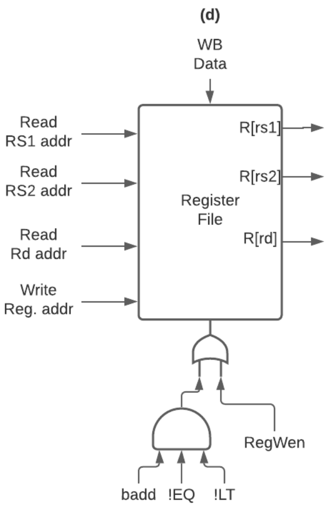
48. Consider the following modifications to the Reg[] register file. (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?









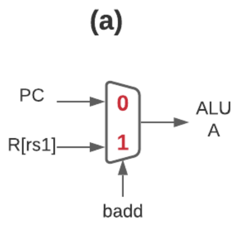
(d)

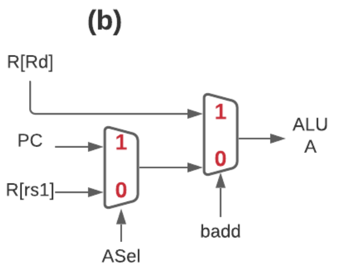
49. What is ASel and BSel (2)

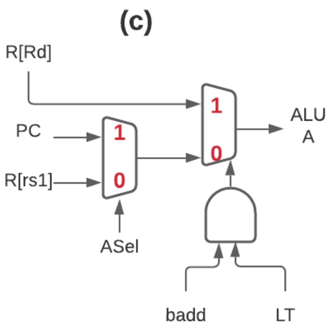
Don't care. They don't matter. Cause see 50 and 51. In ASel case we always send through Rd in second mux.

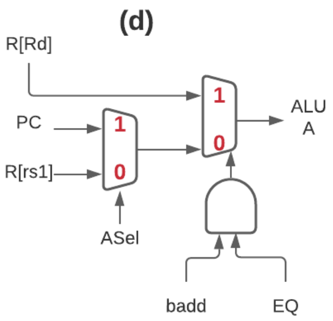
50. What are the changes to mux-A ? (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?







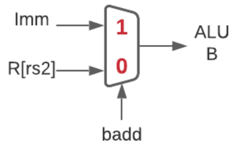


B. adds a control path that checks if the instruction is badd (via the control signal). If it is, Mux-A routes the correct comparison inputs (rs1 or rs2) to the ALU for evaluation. For all other instructions, Mux-A behaves as in the standard datapath, ensuring no disruption to existing functionality.

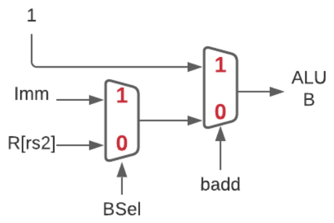
51. What are the changes to mux-B ? (2)

Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?

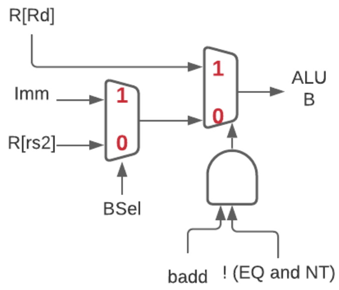
(a)



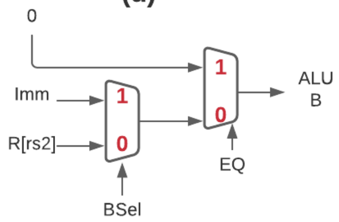
(b)



(c)



(d)



B **For Standard Instructions:** Mux-B behaves as usual, selecting inputs like rs2 (for R-type instructions) or immediate values (for I-type instructions), ensuring compatibility with existing functionality. **For badd:** Mux-B selects rs2 to provide the value needed for the ALU comparison logic (!(EQ OR LT)), allowing the instruction to function correctly

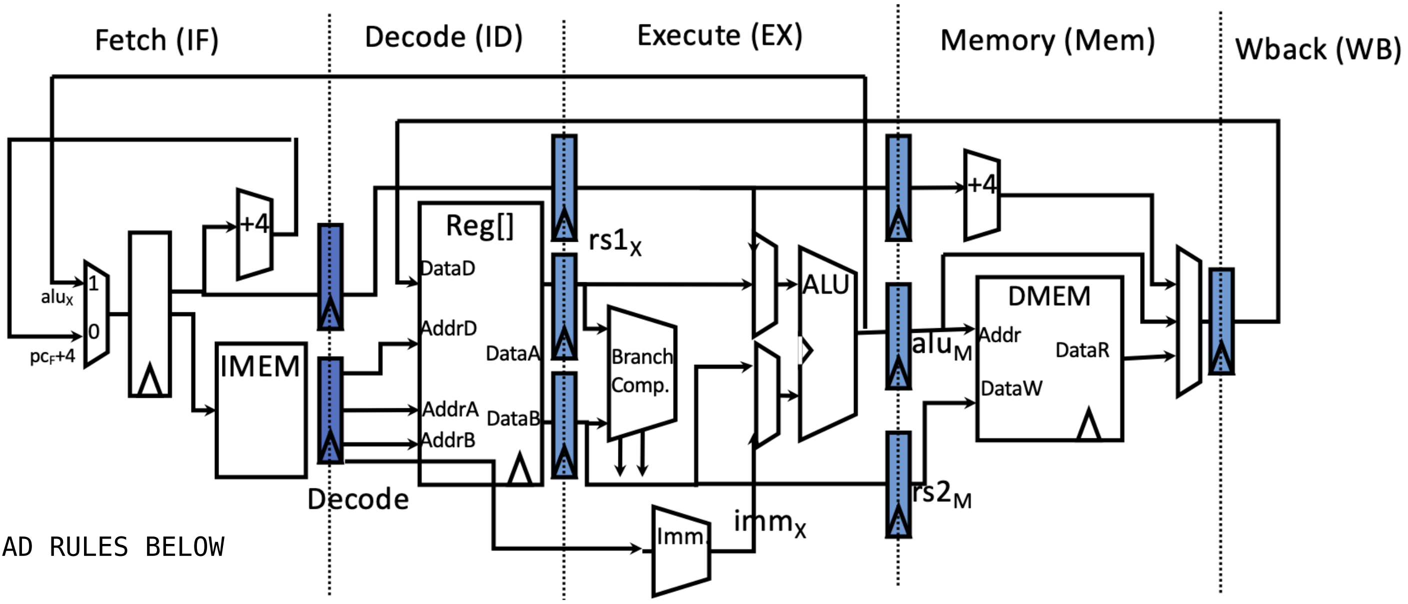
52. The WBsel select is _____ ? (2)

ALU

F. RISC-V Pipeline 17 points.

```
 Refer slide deck L29–Hazard Week 11 if you need to.
```

Consider a typical 5-stage (Fetch,Decode,EXecute,Memory,WriteBack) pipeline.Assume pipeline registers exist where the dotted lines are.



```
READ RULES BELOW
```

- Forwarding/Bypassing is implemented to EX stage only

- Branches targets are calculated in the EX stage.
- Branch comparison in the EX stage.
- We can read and write the register in a cycle.
- A stall is the number of extra cycles an instruction wastes cause of a hazard.
- Assume at t=0 instruction `addi t1,t1,1` is in IF stage when calculating cycles below.

Part I

```
if (a0 == 1) {
   // Choice 1
} else if (a0 == 2) {
   // Choice 2
} else if (a0 == 3) {
  // Choice 3
}
```

The assembly code below implements the following

```
1  main:
2    addi a0,zero,1
3    addi t1,t1,1
4    beq  a0,t1,choice1
5    addi t1,t1,1
6    beq  a0,t1,choice2
7    addi t1,t1,1
8    beq  a0,t1,choice3
9
10 choice1: # Jumped into choice 1
11         j end
12 choice2:  # Jumped into choice 2.
13   j end
14 choice3:
15   j end
16 end:
17   addi a0, zero, 10
18   ecall # terminate ecall
```

```
Use the provided linux numbers in code listing
when answering questions below
```

53. What hazards existing between line 3 and 4 ? (1)

Data. Line 3 rd is t1 and in Line 3. rs1 = t1

54. What hazards existing between line 4 and 5 ? (1)

Control. Line 5 may or may not run based on line 4.

55. What is the instruction in IF stage t = 6 ? line ____ (1)

```
only write line number e.g., 2 if instruction is addi a0,zero,1
```

13: jal zero, end . See spread sheet at the end.

56. What is the instruction in IF stage of t = 7 ? line ____ (1)

15: jal zero, end

57. How many cycles does this program take to complete ? ______ (2)

16

Part 2.

For questions below assume we have changed line 2 to `addi a0,zero,2` i.e., we are going into choice 2.

58. What is the instruction in IF stage t = 6 ? line _____ (1)

line 8:beq a0,t1,choice3

59. What is the instruction in IF stage of t = 7 ? line _____ (1)

line 13: j end

60. How many cycles does this program take to complete ? ______ (2)

18

Part 3.

For questions below assume we have changed line 2 to `addi a0,zero,3` i.e., we are going into choice 3.

61. What is the instruction in IF stage t = 6 ? (1)

line8:beq a0,t1,choice3

62. What is the instruction in IF stage of t = 7 ? (1)

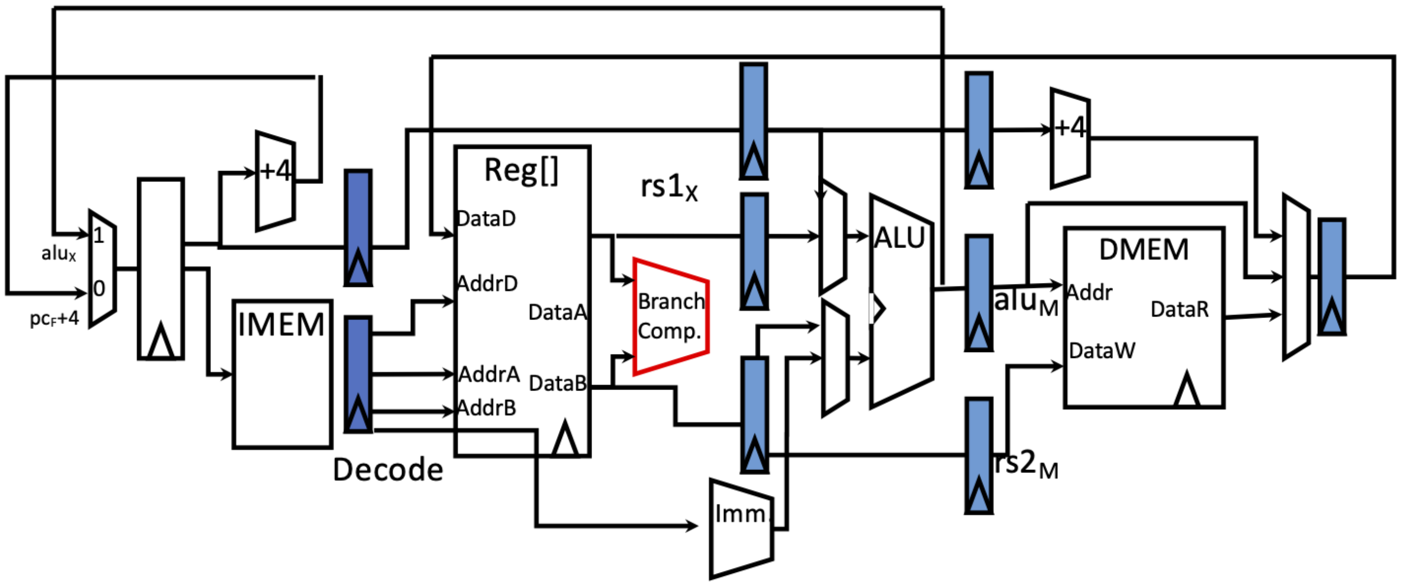
line 11: j end

63. How many cycles does this program take to complete ? ______ (2)

20 Q63 see more details at the end of exam.

Part 4.

Now consider the pipeline modification below



- The branch comparison has been moved to the Decode stage
- Note that branch addresses still get resolved only in the EX stage. Answer questions
- below. `Hint: Think about data hazards for the branch instruction`

64. What is the instruction in IF stage t = 6 ? (1)

`Assume first instruction is addi a0,zero,2` line6:beq a0,t1,choice2.

In this question, the branch comparison is being done in the ID stage. So any branch instruction needs both source registers to be available at ID otherwise it will stall until they are. addi (line 3) computes its result t1 in EX (cycle 3) while beq (line 4) is expecting the result in the same cycle when it is supposed to be in ID, So beq will stall for a cycle after IF since it can't go to ID until addi completes EX. This is the case for other branches as well.

65. How many cycles does this program take to complete ? ______ (2)20 cycles

F. Pipeline CPU.

Consider the pipeline cpu above with the stage timings as shown below.

MUX	ALU	DMEM	Regfile	IMMGEN	BRANCH	IMEM

MUX	ALU	DMEM	Regfile	IMMGEN	BRANCH	IMEM
20	220	230	100	30	30	200

66. Assuming that this CPU is NOT Pipelined (i.e. it is a singlecycle CPU), what is the

shortest clock period possible to execute the store instruction? ______

200+100+220+20+230 = 770ps . Question instructions (given at exam time) was to ignore mux in IF stage.

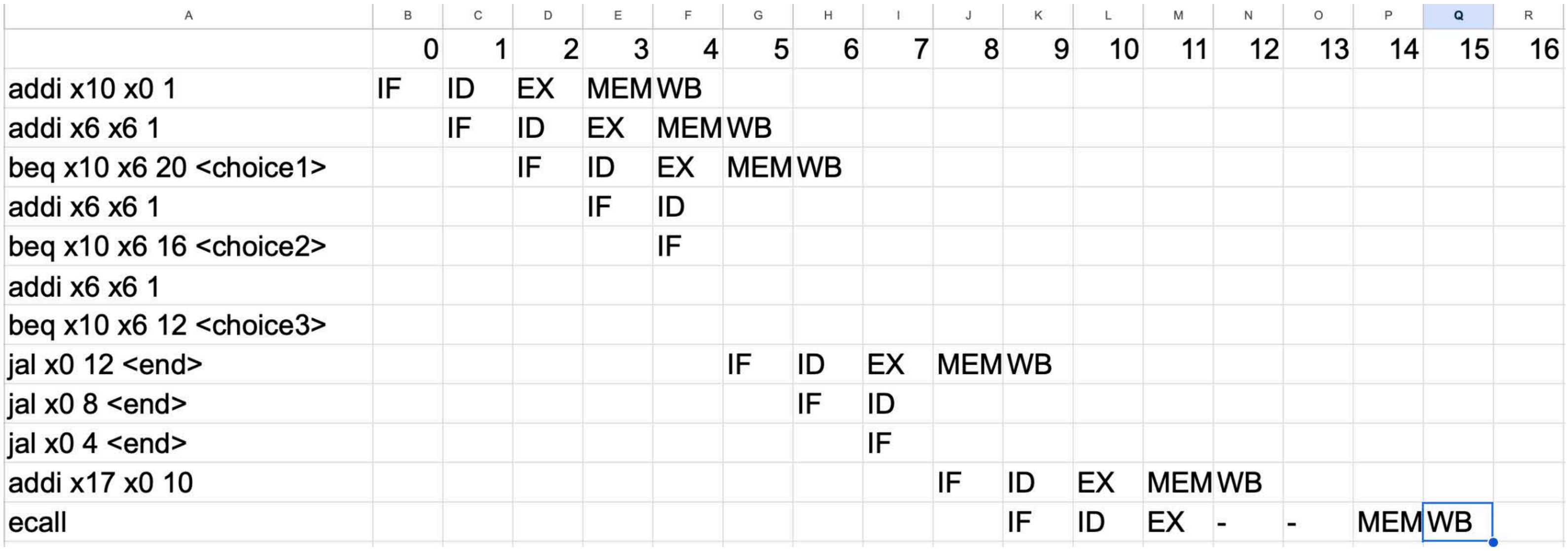
IN the exam the question was specific to store instruction (so no writeback required).

67. Assuming that this CPU is Pipelined, what is the

shortest clock period possible to execute the instruction? ______

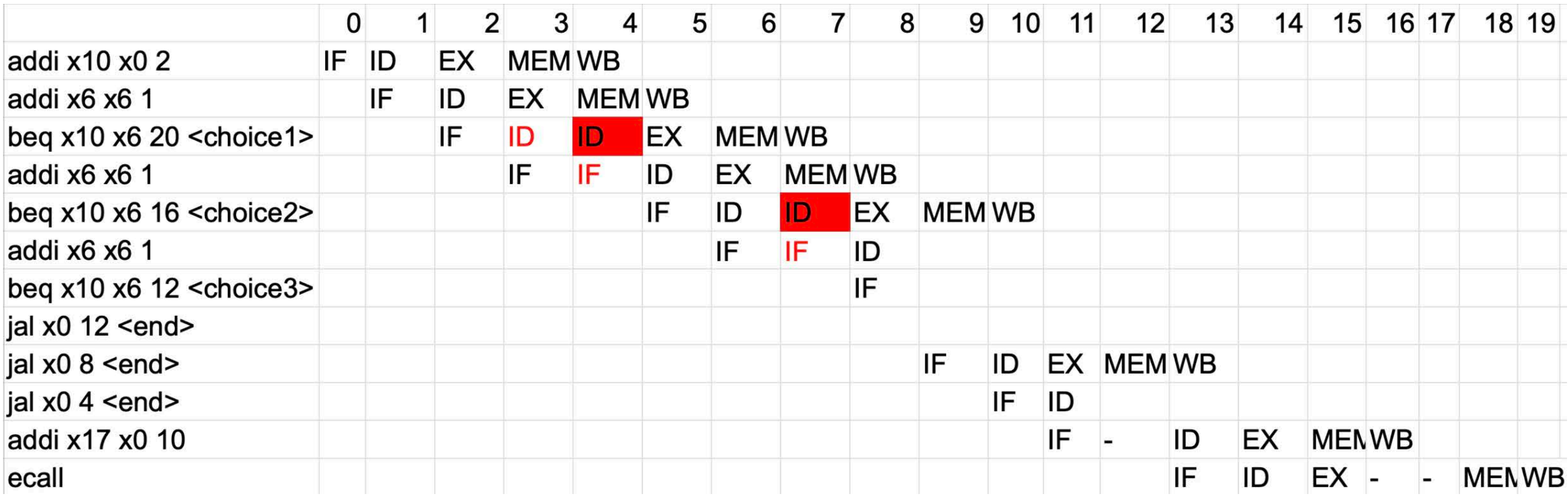
240ps

Q53-57



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
addi x10 x0 1	IF	ID	EX	MEM	WB												
addi x6 x6 1		IF	ID	EX	MEM	WB											
beq x10 x6 20 <choice1>			IF	ID	EX	MEM	WB										
addi x6 x6 1				IF	ID												
beq x10 x6 16 <choice2>					IF												
addi x6 x6 1																	
beq x10 x6 12 <choice3>																	
jal x0 12 <end>						IF	ID	EX	MEM	WB							
jal x0 8 <end>							IF	ID									
jal x0 4 <end>								IF									
addi x17 x0 10									IF	ID	EX	MEM	WB				
ecall										IF	ID	EX	-	-	MEM	WB	

Q64-Q65



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
addi x10 x0 2	IF	ID	EX	MEM	WB															
addi x6 x6 1		IF	ID	EX	MEM	WB														
beq x10 x6 20 <choice1>			IF	ID	ID	EX	MEM	WB												
addi x6 x6 1				IF	IF	ID	EX	MEM	WB											
beq x10 x6 16 <choice2>						IF	ID	ID	EX	MEM	WB									
addi x6 x6 1							IF	IF	ID											
beq x10 x6 12 <choice3>									IF											
jal x0 12 <end>																				
jal x0 8 <end>										IF	ID	EX	MEM	WB						
jal x0 4 <end>											IF	ID								
addi x17 x0 10												IF	-	ID	EX	MEM	WB			
ecall													IF	ID	EX	-	-	MEM	WB	