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.
- NO SMARTPHONES.
- **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 Canvas 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.
- Show all work, clearly and in order, if you want to get full credit.
- I reserve the right to take off points if I cannot see how you logically got to the answer (even if your final answeris correct).
- Circle or otherwise indicate your final answers.
- Please keep your written answers brief; be clear and to the point.
- I will take points off for rambling and for incorrect or irrelevant statements. This test has six problems.

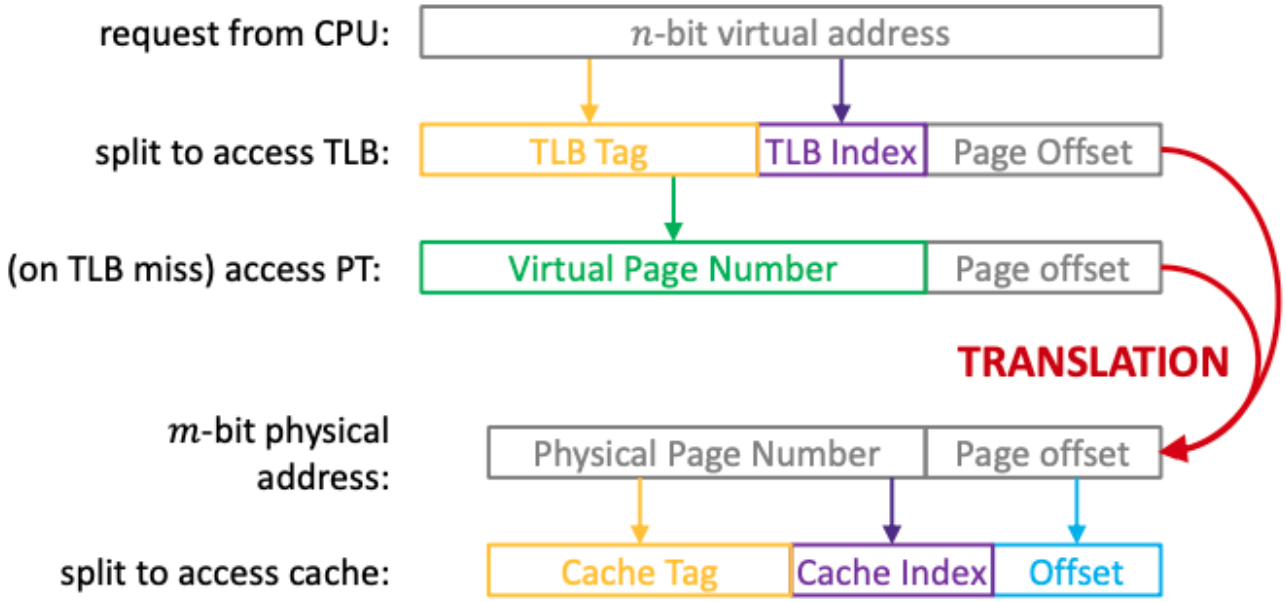
- HONOR CODE
- Questions Sheet.
- Section Virtual Memory 17 points. Canvas Q1-Q31
  - Common questions. Canvas Q1-Q2
  - For the virtual address 0x11a39 answer the following Canvas Q3-Q12. (0.5 pt each)
  - For the virtual address 0xab6e7 answer the following. Canvas Q13-Q22 (0.5 pt each)
  - c. For the following virtual address 0x02974 answer the following (Canvas: 23-32) (0.5 pt each)
- B. Easy. RISCV Blackbox. [10 Points]
  - 33. What is the value of the registers a0,a1, a2 on line 5: CHECK ? (show your working and explain your answer). Write down in hex [5]
  - 34. Lets say message* is now a 2 character string "J-". The value of a2 on line 5:CHECK is 0. What is the mystery character "-". It has to be an ascii character between '0'--'9' (note not value 0-9). ? Write down the digit. [5]
- C. Lets Cache I (10pts)
  - 35. What is the number of tag bits in Cache-A and Cache-B ? (1pt)
  - 36. What is the number of index bits in Cache-A and Cache-B ? (1pt)
  - 37. What is the number of offset bits in Cache-A and Cache-B ? (1pt)
  - 38. What is the hit rate for Cache-A and Cache-B for loop 1? What types of misses do we get? (2 pts)
  - 39. What is the hit rate for Cache-A and Cache-B when you execute loop 2? (5 pts)
- D. Lets Cache II (10pts)
  - 40. What is the bits for virtual address? (1)
  - 41. What is the bits for physical address? (1)
  - 42. Which associativity below will maximize cache size while maintaining the same Tag:Index:Offset? (1)
  - 43. Which block size below will maximize cache size while maintaining the same Tag:Index:Offset? (1)
  - 44. Assuming associativity and block size from questions 42 and 43 what is the number of blocks? (1)
  - 45. Now we're working with a direct mapped cache with same TIO and block size as 43. What is miss rate for code below (3)
  - 46. You add an L2 and shrink the L1. L1 hit latency is 3 cycles. L2 hit latency is 50 cycles. L2 hit rate is 90%. L1 hit rate is 25%. Memory is 100 cycles What is AMAT ? (2)
- F. RISC-V Single Cycle Datapath
  - 47. What is the number of register that beqjalr needs to read and write in a single cycle ? (1)
  - 48. What is the RegWEn signal ? (1)
  - 49. What is the branch comparison signal we are interested in? (1)
  - 50. Which fields are passed to the branch comparison ? (1)
  - 51. What is the logic for beqjalr signal ? (1)
  - 52. Consider the following modifications to the Reg[] register file. (2)

Section Virtual Memory 17 points. Canvas Q1-Q31

```
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: 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 .
```



Parameter	Value
Physical address bits	16
Size of page	256 bytes
Virtual address bits	20
----------	----------
TLB Sets	4
TLB Ways	2
TLB Size	8 entries
----------	----------

Parameter	Value
Cache block	8 bytes
Cache size	64 bytes
Cache Sets	8
Cache Ways	1

Terminology

- 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	PPN
Index:0 TLB-T:[0x69]	---
Index:1 TLB-T:[0x396]	---
Index:2 TLB-T:[0x46]	eb
Index:3 TLB-T:[0x29c]	4a

Way 1	0
Index:0 TLB-T:[0x3b]	---
Index:1 TLB-T:[0x010]	88
Index:2 TLB-T:[0x2ad]	4b
Index:3 TLB-T:[0x6]	dc

Page Table (Partial)

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

VPN	PPN	Valid
0x1a4	---	0

VPN	PPN	Valid
0xe59	---	0
0x11a	0xeb	1
0xa73	0x4a	1
0xec	0x00	1
0x29	0x12	1
0xab6	0x4b	1
0x1b	0xdc	1
0x8ff	0x91	1
0xbf1	0x47	1
0x016	0x99	1
0x010	0x88	1

Cache

- Way 0

	0	1	2	3	4	5	6	7
Index: 0 [0x3ae]	0x39	0x24	0x76	0x62	0x17	0x92	0x72	0x38
Index: 1 [0x373]	0x24	0x6c	0x38	0x47	0x8e	0x43	0xc2	0x29
Index: 2 [0x2c7]	0x08	0x9d	0xd1	0xf8	0x53	0x05	0x14	0x16
Index: 3 [0xdc]	0xfa	0x8c	0x3d	0x11	0x14	0xf4	0xe9	0xc3
Index: 4 [0x12b]	0x11	0xfa	0x4c	0x9d	0xb3	0x1d	0xb6	0x87
Index: 5 [0x27e]	0xdc	0x0f	0x44	0x85	0x6b	0xfc	0x04	0x4f
Index: 6 [0x49]	0x38	0xde	0xfb	0x10	0x3d	0xfe	0x05	0xc5
Index: 7 [0x3ac]	0x49	0x52	0xe0	0xdd	0xf0	0x0d	0xa6	0xe0

Common questions. Canvas Q1-Q2

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

- 12

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

- 8

For the virtual address 0x11a39 answer the following Canvas Q3-Q12. (0.5 pt each)

What is the VPN

- 0x11a

What is the TLB tag.

- 0x46
- Is it a TLB hit or miss
  - Hit
- Is it a page fault
  - NO
- What is the PPN ?
  - 0xeb
- what is the cache tag ?
  - 0x3ac
- what is the cache index
  - 0x7
- What is the byte offset
  - 0x1
- Is it a cache hit or miss
  - Hit
- What is the data byte
  - 0x52

For the virtual address 0xab6e7 answer the following. Canvas Q13-Q22 (0.5 pt each)

- What is the VPN
  - 0xab6

- What is the TLB tag.
  - 0x2ad
- Is it a TLB hit or miss
  - Hit
- Is it a page fault
  - NO
- What is the PPN ?
  - 0x4b
- what is the cache tag ?
- 0x12F (0x4be0: 01 0010 1111 010 000)
- what is the cache index
  - 0x4
- What is the byte offset
  - 0x7
- Is it a cache hit or miss
  - Miss
- What is the data byte
  - N/A

c. For the following virtual address 0x02974 answer the following (Canvas: 23-32) (0.5 pt each)

- What is the VPN
  - 0x029
- What is the TLB tag.
  - 0x0a
- Is it a TLB hit or miss
  - Miss
- Is it a page fault
  - NO
- What is the PPN ?
  - 0x12

  00 0100 1001 110 000
- what is the cache tag ?
  - 0x049
- what is the cache index
  - 0x6
- What is the byte offset
  - 0x04
- Is it a cache hit or miss

    - Hit
- What is the data byte
    - 0x3d

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 output.

```
char *message = "MESSAGE"
```

Study the following RISC-V code shown below and answer the questions. You can assume the data segments starts at 0x10000000 and only contains message. You can assume a0 contains value of message* at the start.

```
.text
main:
  li  a2,0xFF
  jal BLACKBOX
  # CHECK
  li a0,10
  ecall

BLACKBOX:
  addi sp,sp,-4
  sw ra, 0(sp)

  lbu  a1,0(a0)
  beq a1,zero,end
  and a2,a1,a2
  addi a0,a0,1
  jal BLACKBOX

end:
  lw ra, 0 (sp)
  addi sp,sp,4
  ret
```

33. What is the value of the registers a0,a1, a2 on line 5: CHECK ? (show your working and explain your answer). Write down in hex [5]

The program simply bitwise ands the ascii value of every letter in this string
a0 : contains index in each recursive iteration of the loop
a1: contains the actual index
a2: final result

a0: 0x10000007

a1: 0x0 (always NULL; final character)

a2: 0x41 (for message) 4D & 65 & 73 & 73 & 61 & 67 & 65.

If you write the program ands the letters we gave you 2.5 pts. More explanation

34. Lets say message* is now a 2 character string "J-". The value of a2 on line 5:CHECK is 0. What is the mystery character "-". It has to be an ascii character between '0'--'9' (note not value 0-9). ? Write down the digit. [5]

The mystery digit is one which is ~ of J's ascii value.

`J` is 4A : 01001010

Hence the mystery digit has to be 00110101

Answer is `5` : ASCII 53: 00110101

C. Lets Cache I (10pts)

Assume we are working in a 4GB physical address space.

Cache-A	Direct-mapped, 4KB, 512 byte blocks
Cache-B	Set-associate, 4KB, 2 ways, 512 byte block

Both caches use write-back and write-allocate policies.

Answer questions below

```
int size = 2 * 1024;
// long long int is 8 bytes
long long int array[size];
/* loop 1 */
for (int i = 0; i < size; i++) {
  array[i] = i;
}

/* loop 2 */
for (int i = 0; i < size; i += 8) {
// Order in which access happens
//3rd       1st         2nd
array[i] =  array[i] * arr[0];
}
```

35. What is the number of tag bits in Cache-A and Cache-B ? (1pt)

Direct mapped : 20 and

Set associative: 21

36. What is the number of index bits in Cache-A and Cache-B ? (1pt)

3 and 2.

37. What is the number of offset bits in Cache-A and Cache-B ? (1pt)

9 and 9

38. What is the hit rate for Cache-A and Cache-B for loop 1? What types of misses do we get? (2 pts)

63/64. Compulsory misses

39. What is the hit rate for Cache-A and Cache-B when you execute loop 2? (5 pts)

- `Hint: You should try considering the miss rate of each of the accesses in loop 2 independently.`
- `Assume that you have executed loop 1`
- `Assume the worst case ordering of accesses within a single iteration of the loop`

Expressions are sufficient. You do not need to calculate fractions.
You need to show your working and reasoning to obtain points.

Array size is 16KB and Cache size is 4KB. After loop 1, final 1/4th of array left in the cache.
array[size-$\frac{1}{4}size$] to array[size]

In loop 2, stride is 64B. block size is 512B. Thus we have 8 accesses per block. Each iteration has 3 accesses.
In the direct mapped cache

- First quarter of array. array[0...$\frac{1}{4}size$], every new block is a miss but rest of accesses are hits.

Access stream
Iteration 0: array[0]:M,array[0]:H,array[0]:H
Iteration 1: array[8]:H, array[0]:H, array[8]:H all the way to array[56] from array[64] new cycle starts all the way to
array[512] when cache fills up

- From 2nd/quarter to array size i.e., array[512]....array[2047]
  now it's useful

Consider 1st read to array[i]. All accesses will be misses.
Consider 1st read to array[0] this will end up evicting array[i] since the cache is full and they map to the same set (worst case). This is also a miss
Consider the write to array[i] this will evict array[0] and end up being a miss

However for array[i+8] in the next iteration will be a hit since the previous iteration brought in the block.

Hence

	1st array[i]	2nd array[0]	3rd array[i]
ith iteration	Miss	Miss	Miss
i+1	Hit	Miss	Miss
i +2	Hit	Miss	Miss

all the way to array[i+56] when a new cycle will start.
In the new cycle however, array[0] and array[i+64] will map to different sets. Hence its just the first touch to the new blocks that will miss. $\frac{23}{24}$ hits Overall: $\frac{29}{32} \times \frac{23}{24} + \frac{3}{32} \times \frac{7}{24}$

Set associative:
array[0] and array[i] map to different sets. No conflict.
23/24 accesses will be hits.

1. 2 pts - if they said in direct mapped. array[0] and array[i] will conflict. But set associative does not.
2. 4 points - if they said 1 and reason that array[i] in subsequent iteration will hit based on previous iteration. More explanation

D. Lets Cache II (10pts)

We have a mystery cache. A mystery, byte addressed cache has Tag:Index:Offset (T:I:O) = 12:4:3.

40. What is the bits for virtual address? (1)

N/A

41. What is the bits for physical address? (1)

19

42. Which associativity below will maximize cache size while maintaining the same Tag:Index:Offset? (1)

43. Which block size below will maximize cache size while maintaining the same Tag:Index:Offset? (1)

8 bytes. The exam had multiple choices. To maximize cache size choose the largest associativity and largest block size.

44. Assuming associativity and block size from questions 42 and 43 what is the number of blocks? (1)

128

45. Now we're working with a direct mapped cache with same TIO and block size as Q43. What is miss rate for code below (3)

```
#define LEN 2048
int ARRAY[LEN];
int main() {
    for (int i = 0; i < LEN - 32; i+=32) {
       ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+32];
       ARRAY[i] += 10;
}
```

Every iteration it's
ARRAY[i] read MISS
ARRAY[i+1] read HIT
ARRAY[i+32] CONFLICT MISS with
ARRAY[i] CONFLICT MISS
ARRAY[i] read HIT
ARRAY[i] write HIT
3 MISSES, 3 HITS. 50% hit rate

46. You add an L2 and shrink the L1. L1 hit latency is 3 cycles. L2 hit latency is 50 cycles. L2 hit rate is 90%. L1 hit rate is 25%. Memory is 100 cycles What is AMAT ? (2)

$3 + 0.75 * (50 + (0.10 * 100)) = 48 cycles$

F. RISC-V Single Cycle Datapath

We wish to introduce a new instruction into our RISC-V datapath.
`beqjalr` . This instruction is a conditional jal instruction. Typical branch instruction has limited offsets

since all offsets are PC-relative. Jalr however can specify register relative. The instruction works as follows. The instruction is encoded as follows.

- We no longer use an immediate for the branch offsets
- We revert imm[4:1|11] to specify a register

`original beq instruction`

imm[12	10:5]	rs2	rs1	0x0	imm[4:1

`new beqjalr instruction`

f7 (0x40)	rs2	rs1	0x0	rd	1100011

```
if (R[rs1] == R[rs2]) {
 R[ra] = PC + 4;
 PC = R[rd];
} else  {
// Zero out the rd register and destroy
 R[rd] = 0
}

```

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

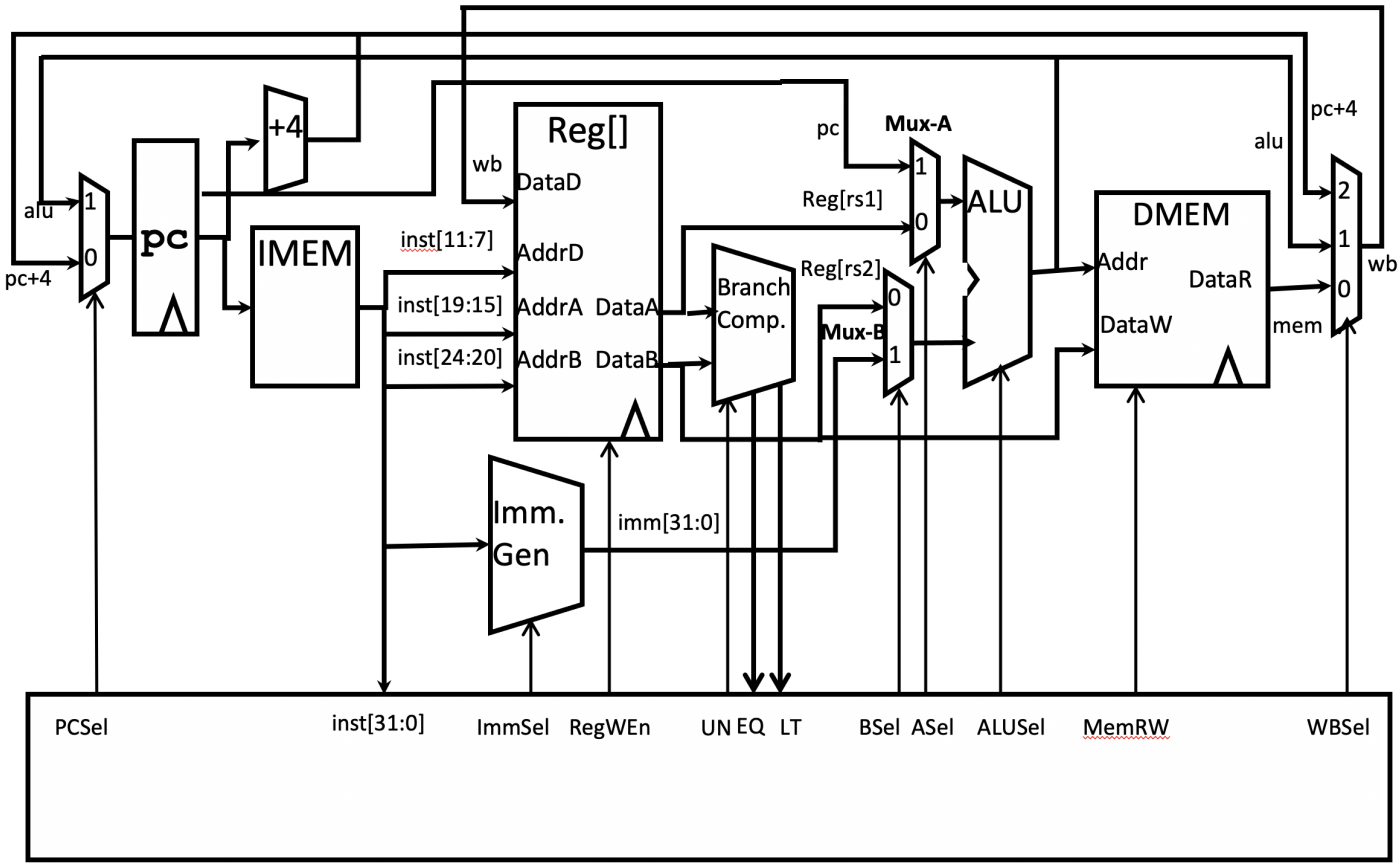
- Like a branch it performs a comparison.
- If comparison succeeds it performs a JALR, the destination is contained in the rd register. `rd is treated as a source .`
- If comparison fails, it simply zeros out the register which contains the PC it should jump to. `rd is treated as a destination`
- We have a new control signal `beqjalr` which is 1 if the instruction being decoded is a beqjalr

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:

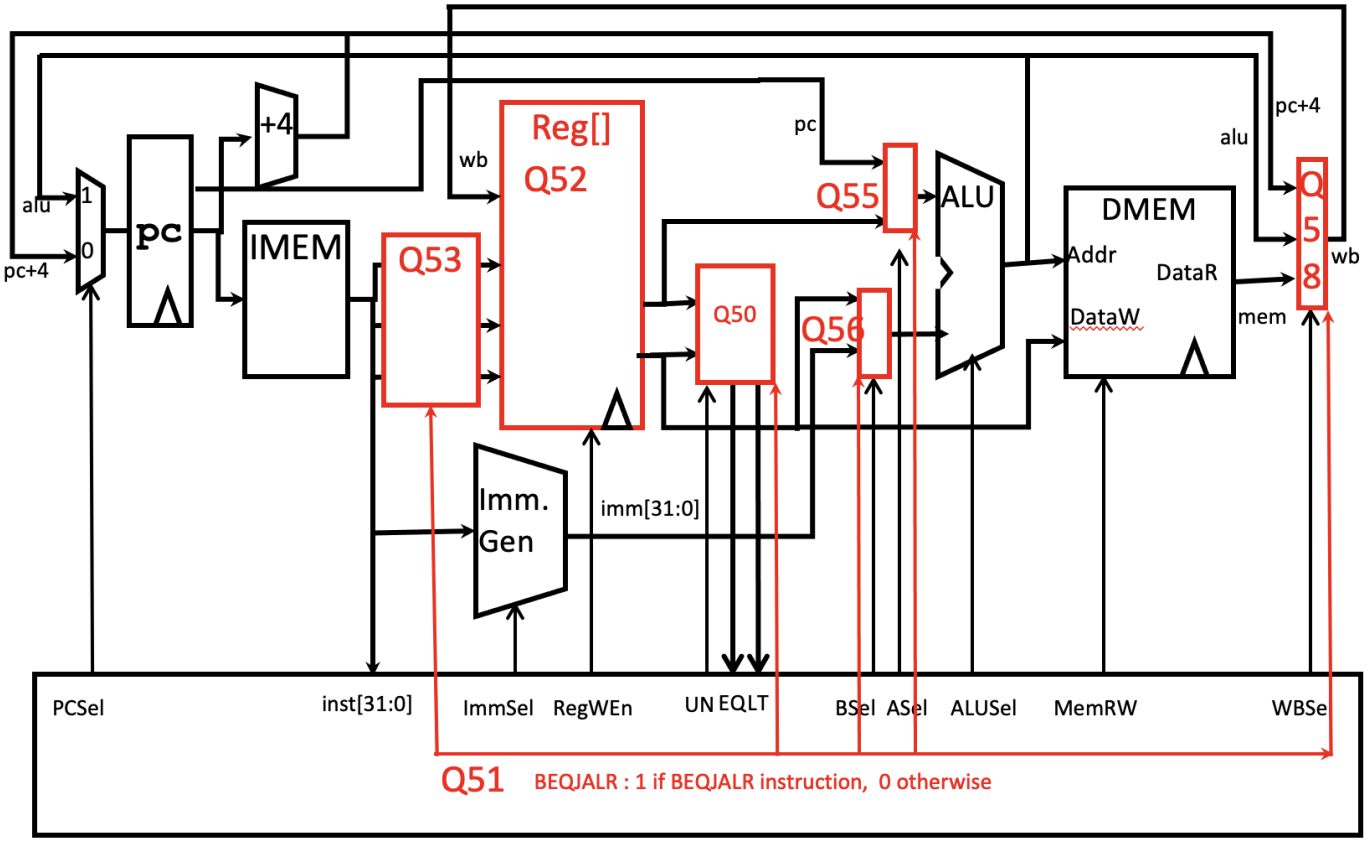
`Caution 2: Pay careful attention to which input line is 1 and which line is 0 in the muxes. Some muxes choose top-most input as 0, some choose bottom-most input as 0`

`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`

Baseline Pipeline



Pipeline with beqjalr (Red boxes indicate questions)



47. What is the number of register that beqjalr needs to read and write in a single cycle ? (1)

3 reads 1 write

48. What is the RegWEn signal ? (1)

1

49. What is the branch comparison signal we are interested in? (1)

EQ & ~UN

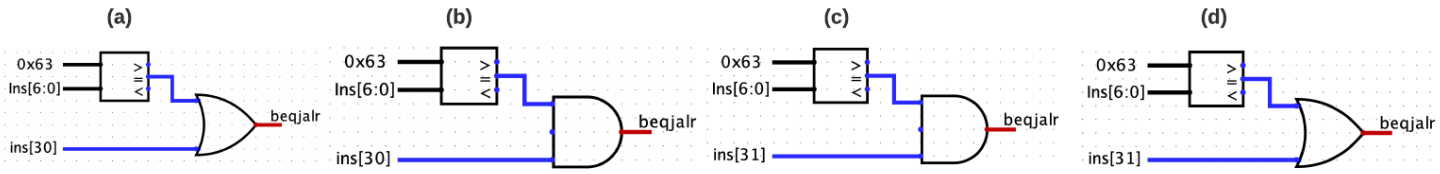
or

EQ and NOT UN

50. Which fields are passed to the branch comparison ? (1)

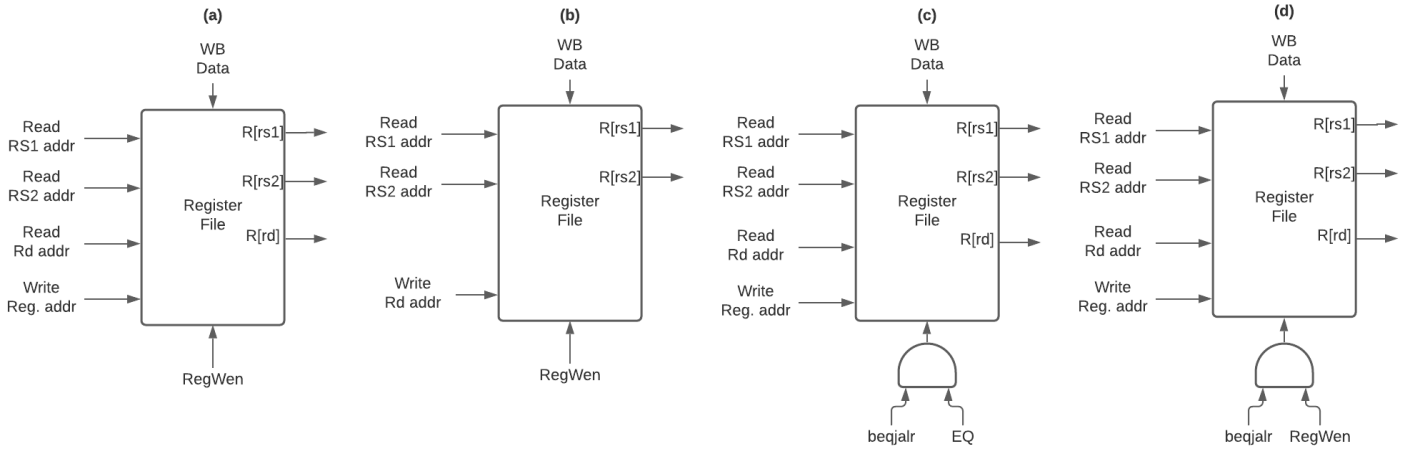
R[rs1] and R[rs2]

51. What is the logic for beqjalr signal ? (1)



(c) The comparison is done in the branch comparison block since it is exactly the same function (comparing R[rs1] and R[rs2]). The ALU needs to compute the new value for R[rd] which is R[rd] + 1, so you need to select "1" from the B-mux for this instruction.
A useful hint for these types of questions is to try to figure out what the ALU needs to do. It is usually the computation that cannot be done at any other block. In this case comparing rs1 and rs2 is done by the branch comparison block already, but no other block can be used to increment a value by 1.

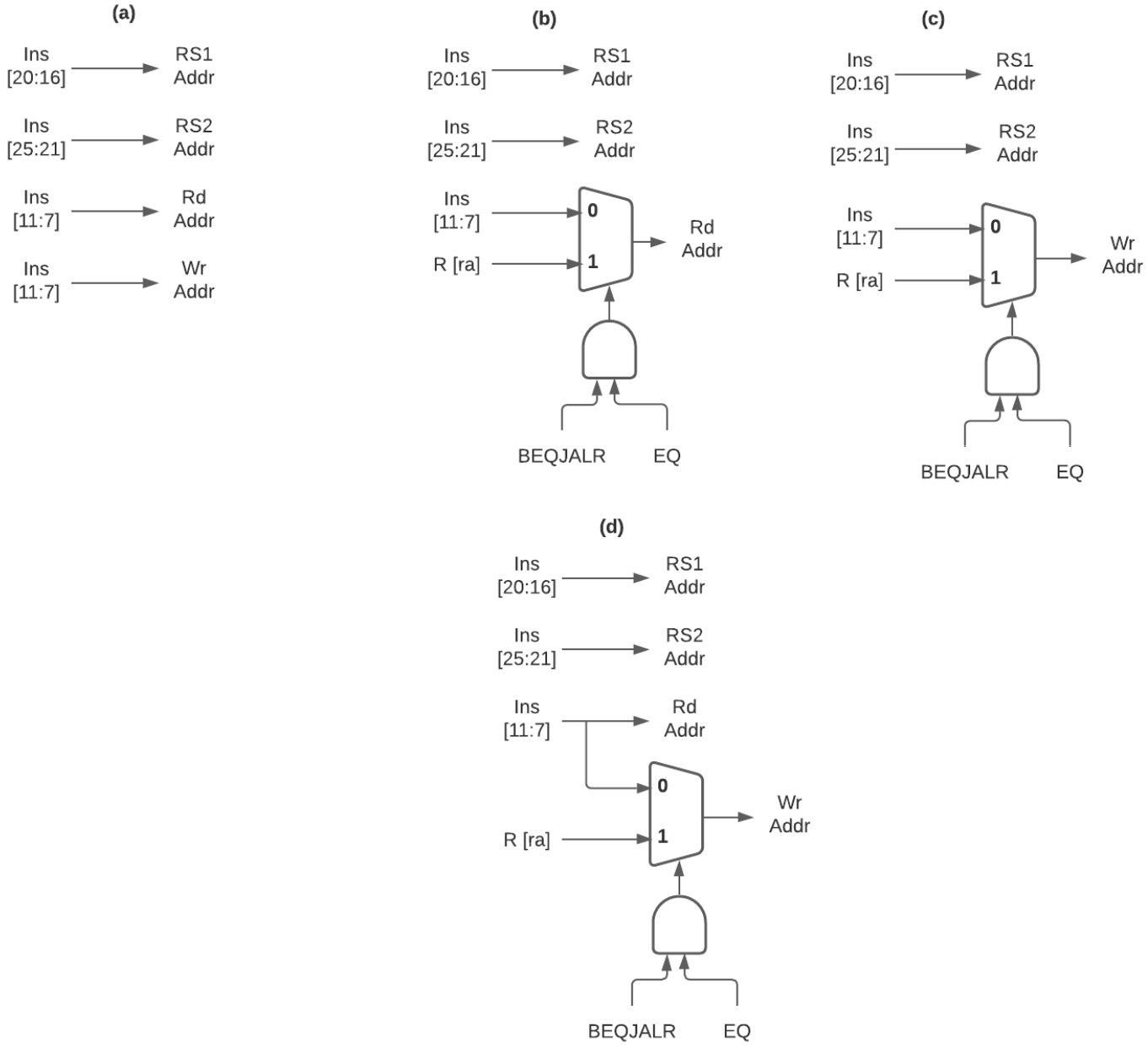
52. Consider the following modifications to the Reg[] register file. (2)Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?



(a) . 3 reads and 1 write. Always writes whether condition succeeds or fails.

53. What are the inputs to the register file ? (2)

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



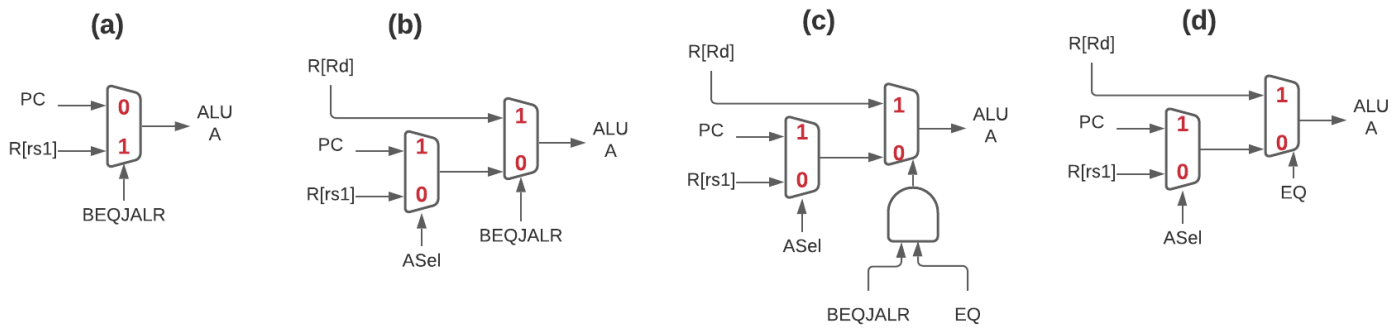
(d). If BEQJALR instruction and EQ (i.e., R[rs1] == R[rs2] then we need to select R[ra], otherwise we select Rd to be written.
We also need to read Rs1 and Rs2 and pass it to branch unit; Rd is read and passed onto ALU.

54. What are the inputs to the ALU ? (1)

R[Rd] and 0

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

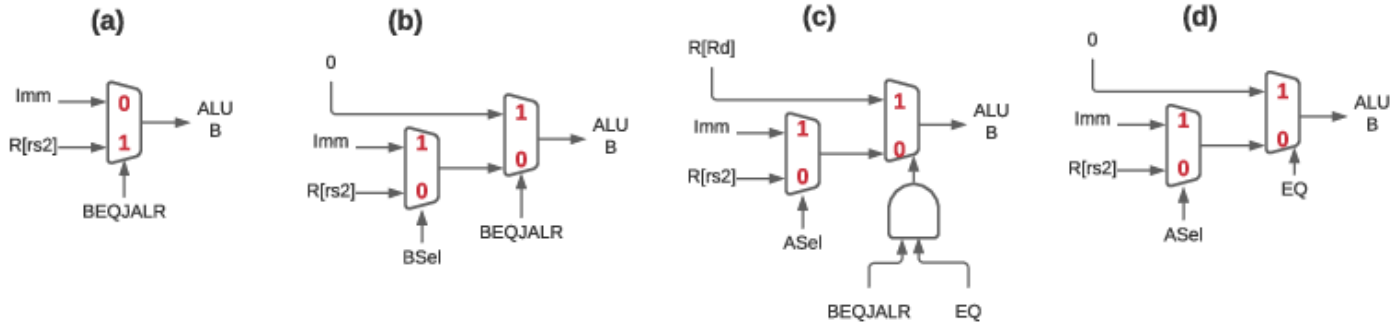
Which configuration will allow this instruction to execute correctly without breaking the execution of other instructions in our instruction set?



(b) The ALU is computing R[rd}+0 (from Question 54). So the ASel mux needs to perform the same function for all other instructions (choose between PC and R[rs1), and it needs to choose R[rd] if the instruction is BEQJALR. Answer b chooses the original ASel mux for all other instructions (BEQJALR = 0) and chooses R[rd] for the new instruction (BEQJALR = 1)

56. 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?



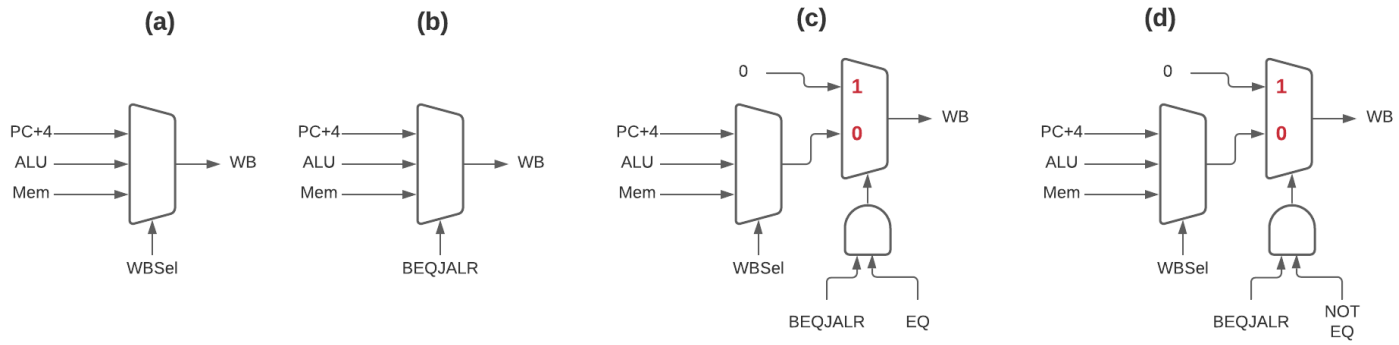
(b)

57. For beqjalr instruction signal does WBsel choose ? (2)

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

PC+4

58. What are the changes to the writeback stage ? (2)

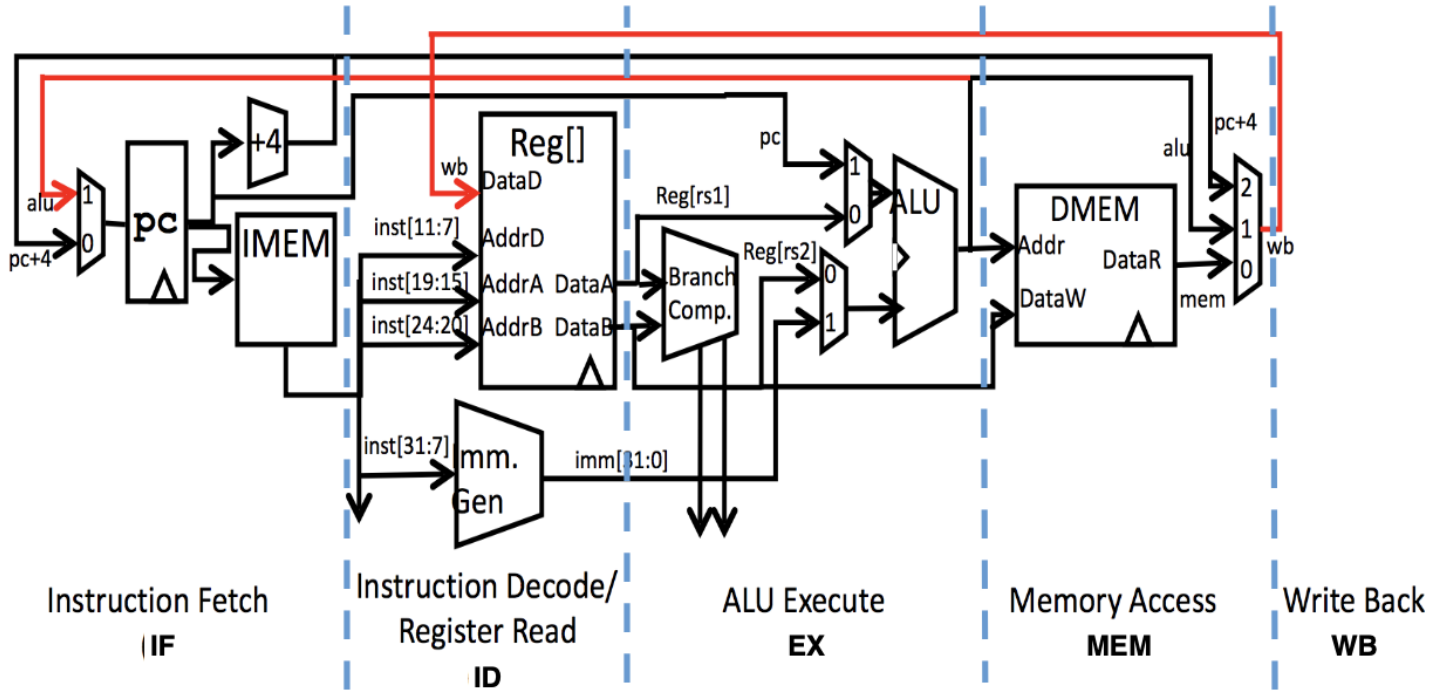


(d)

E. RISC-V Pipeline 12 points.

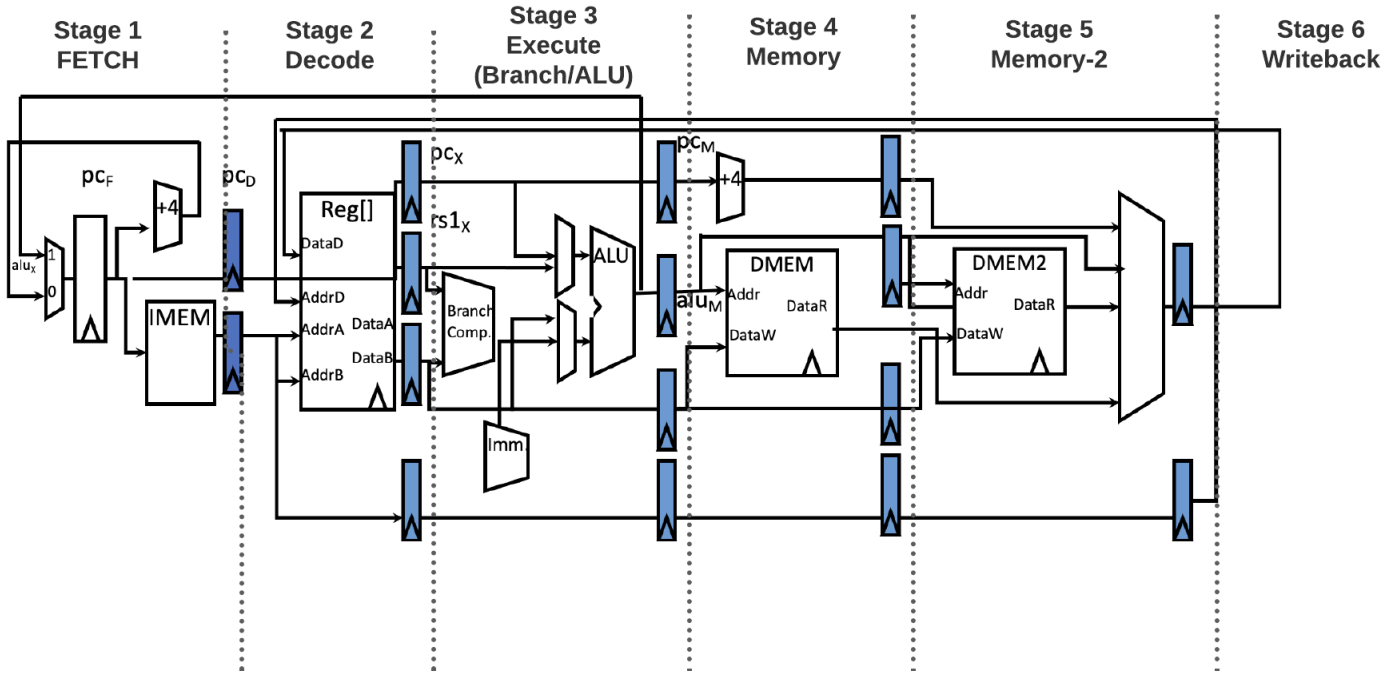
`Refer slide deck L29-Hazard Week 11 if you need to.`

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



This pipeline is more simple than the one you dealt with in the assignment. `READ RULES BELOW`

- Forwarding/Bypassing is not implemented; dependent instructions will have to wait for WB.
- Following a branch, the next instructions stalls until branch is resolved.
- A stall is the number of cycles after the previous instruction that the current instruction can start.
- We modify this pipeline to add another memory stage and create a six-stage pipeline.



Answer questions based on the following program

For the given code, what hazards might exist and due to which lines?

Assume all registers have been initialised and that all labels are defined. a0 = 0.

```
1  bneq a0,zero,end
2  addi t0,t0,1
3  lw   s0,0x10(t0)
4  mul  s1,s0,a0
5  add  s1,s0,a0
```

59. What hazards existing between instruction 1 and 2 ? (1)

Control

60. What hazards existing between instruction 2 and 3 ? (1)

Data

61. What hazards existing between instruction 3 and 4 ? (1)

Data

62. What hazards existing between instruction 4 and 5 ? (1)

None

63. How many cycles does instruction 2 stall for ? (2)

2

If you answered for instruction 1 and said N/A we still gave you point.

64. How many cycles does instruction 3 stall for ? (2)

4 If you answered for instruction 2 and said 2 we still gave you point.

65. How many cycles does instruction 4 stall for ? (2)

4. If you answered for instruction 3 and said 4 we still gave you point.

66. How many cycles does instruction 5 stall for ? (2)

0. (always count stall once instruction reaches ID).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
bneq a0,zero,en	IF	ID	EX	MEM	MEM	WB													
addi t0,t0,1		IF	ID	IF	ID	EX	MEM	MEM	WB										
lw s0,0x10(t0)			IF		IF	ID	ID	ID	ID	ID	EX	MEM	MEM	WB					
mul s1,s0,a0						IF	IF	IF	IF	IF	ID	ID	ID	ID	ID	EX	MEM	MEM	WB	
add s1,s0,a0											IF	IF	IF	IF	IF	ID	EX	MEM	MEM	WB

This branch is not taken (a0 is zero. see question branch checks for a0!=0) , so typically we would continue to run these instructions. But in this case they stall for branch to be resolved in EX (see question). Both options are shown

G. Lets RISC-V II 10 points

67. We are designing a new RISC-V instruction with 16 registers. Assuming that you use the extra bits to extend the immediate field, what is the range of `half-word` instructions that can be reached using a branch instruction in this new format? (2)

-8192 - 8192 instructions. You need to examine the encoding for branch instructions (B-type) in the RISC-V card. It includes two source registers (rs1 and rs2) and bits 12-1 of the immediate field (bit 0 is always zero). So a branch can have a target at a half-word boundary since the last bit of the immediate field is always zero. If the last two bits were zeros, then the branch target is at a word boundary. If no bits are zeros, then the branch target is at a byte boundary.

What the immediate field provides is the offset from the current PC where the branch target is. So in the original branch instruction format, with a 12-bit immediate field, you can have a target address from PC-2^12 to PC+2^12.

In the new RISC-V with only 16 registers, you only need 4 bits to encode a register number. So both rs1 and rs2 will have 4 bits each instead of 5. The two bits you saved from the registers will be added to the immediate field, so it will become 14 bits. So your target addresses will be from PC-2^13 (which is -8192) to PC+2^13 (which is 8192).

66- refer code below

```
add s0, x0, x0
addi s1, x0, 4
loop:
  beq s0, s1, end
  add a0, a0, s0
  jal ra, square
  jal ra, printf
n:
  addi s0, s0, 1
  j loop
end:
  ecall
square:
  mul a0, a0, a0
  ret
```

68. What is the PC address of `instruction 4:beq` (hex) ?

0x08

69. What is the PC address of `instruction 9:addi` (hex) ?

0x18

70. What is the PC address of `instruction 12:ecall` (hex) ?

0x20

71. What is the PC address of `instruction 14:mul` (hex) ?

0x24

72. What is the machine code for instruction at address 0x1C (hex) ? (2)

j loop = jal x0, -20

0xFEDFF06F

Imm[20|10:1|11|19:12] rd opcode
0b 0 0000001010 0 00000000 00001 1101111

0x00 | add s0, x0, x0
0x04 | addi s1, x0, 4
0x08 | loop: beq s0, s1, end
0x0C | add a0, a0, t0

0x10 | jal ra, square
0x14 | jal ra, printf
0x18 | n: addi s0, s0, 1
0x1C | j loop
0x20 | end: ecall
|
0x24 | square: mul a0, a0, a0
0x28 | ret

73. After the first pass of the assembler instruction at 0d20 does not have the label resolved. (1)

True

74. After the first pass of the assembler instruction at 0d28 does not have the label resolved. (1)

False

72 translation from pseudo instruction to base instruction is from the RISCV Card page 4 (which you will be given in your exam booklet). The offset (-20) is because the instruction your are jumping to is 5 instructions before and each instruction is 4 bytes.
For question 73, the label "printf" hasn't been reached when the instruction was assembled in the first pass, so the label is not resolved.
For question 74, the label "loop" has been reached and is known at the time the instruction was assembled in the first pass.

Look at the assembly in Question 72 above. j loop goes back to line 4 in the original listing which translates to address 0x08 in the assembly.
"label resolved" means that when the assembler reaches the line "j loop" that requires going back to the "loop" label, it had already seen that label and knows what address it is at. So the assembler can translate the instruction using the correct offset. If the "loop" label was after that instruction (which is the case for jumping to printf) then the assembler wouldn't know what address loop is at so it can't come up with the correct offset in the generated assembly instruction.

Note: This is decimal 28 not hex 28.
0d28 = 0x1c which is the "j loop" instruction.