Midterm Review Solutions

CS 465- Fall 2021

Prof. Daniel A. Menasce

Department of Computer Science

Questions

- What is CPI?

- Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?

Questions

- What is CPI?
  - A: cycles per instruction

- Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
  - A: 0.15*2 + 0.25*3+0.6*1 = 1.65

Question

- What is CPI?
- Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
- Suppose the same program above executes 1,000,000 instructions. How many cycles would it take to execute the program?

Question

- What is CPI?
- Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
- Suppose the same program above executes 1,000,000 instructions. How many cycles would it take to execute the program?
  - 1,000,000 * 1.65 = 1,650,000 cycles

Question

- Suppose that the above program runs on a machine that has a cycle time of 200 ps. What is the execution time of the program on this machine?

Question

- Suppose that the above program runs on a machine that has a cycle time of 200 ps. What is the execution time of the program on this machine?

1,650,000 cycles * 200 * 10^{-12} = 3.3 * 10^{-4} sec = 0.33 msec

Question

- Suppose that compiler optimization is used to compile the same program as before. The optimization reduces the total number of instructions by 10% and now 12% of the instructions of the program take 2 cycles, 28% take 3 cycles, and 60% take 1 cycle. What is the execution time of the program now?

Question

- Suppose that compiler optimization is used to compile the same program as before. The optimization reduces the total number of instructions by 10% and now 12% of the instructions of the program take 2 cycles, 28% take 3 cycles, and 60% take 1 cycle. What is the execution time of the program now?

$0.9 * 1{,}000{,}000 * (0.12 * 2 + 0.28 * 3 + 0.6 * 1) * 200 * 10^{-12} = 3.024 * 10^{-4}$ sec

Question

- Consider that 20% percent of a program's instructions are branch instructions and that the CPI for these instructions is 2. The CPI for the remaining instructions is 1.8. What would be the CPI of the program if the hardware designers improved the branch prediction algorithm so that the CPI of branch instructions went down to 1.2?

Question

- Consider that 20% percent of a program's instructions are branch instructions and that the CPI for these instructions is 2. The CPI for the remaining instructions is 1.8. What would be the CPI of the program if the hardware designers improved the branch prediction algorithm so that the CPI of branch instructions went down to 1.2?
  - 0.2 * 1.2 + 0.8 * 1.8 = 1.68

Question

- Which of these elements can influence the number of instructions executed by a program?
  - The algorithm
  - Its input data
  - The language in which it is written
  - The compiler
  - The ISA

Question

- Which of these elements can influence the number of instructions executed by a program?
  - The algorithm
  - Its input data
  - The language in which it is written
  - The compiler
  - The ISA

Answer: all

Question

- How would you compute the CPU time of a program as a function of the number of instructions, the CPI, and the clock cycle duration?

Question

- How would you compute the CPU time of a program as a function of the number of instructions, the CPI, and the clock cycle duration?

CPU time = # instructions * CPI * clock cycle duration

Question

- What is the motivation to design multicore computers?

Question

- What is the motivation to design multicore computers?

The power wall. Processors are using too much power and dissipating too much heat at current clock frequencies

Question

- Is there any instruction in the MIPS ISA that allows a number in main memory to be added to a number in a register?

Question

- Is there any instruction in the MIPS ISA that allows a number in main memory to be added to a number in a register?

No. MIPS only operates on registers.

Question

- What is the MIPS instruction needed to load element A[4] of array A into register $t0 assuming the address of the array is stored at register $s0 and that each element of the array is a 4-byte integer?

Question

- What is the MIPS instruction needed to load element A[4] of array A into register $t0 assuming the address of the array is stored at register $s0 and that each element of the array is a 4-byte integer?

- lw $t0, 8 ($s0)

Questions

- Why MIPS does not have a subtract immediate instruction?

- How are negative integer numbers represented in MIPS and in the majority of processors?

Questions

- Why MIPS does not have a subtract immediate instruction?

Because this can be accomplished by a addi in which one of the operands is negative

- How are negative integer numbers represented in MIPS and in the majority of processors?

Questions

- Why MIPS does not have a subtract immediate instruction?

Because this can be accomplished by a addi in which one of the operands is negative

- How are negative integer numbers represented in MIPS and in the majority of processors?

2's complement

Questions

- What is the sign bit of negative integer numbers in 2's complement?

- 1

- How do you negate a number?

Questions

- What is the sign bit of negative integer numbers in 2's complement?

  - 1

- How do you negate a number?

- Flip the bits and add 1

Question

- How would you compile the statement into MIPS using 3 instructions?

```
if (i==j) f = g;
else f = h;
```

where i, j, are in $t0, $t1, f, g, and h are stored in $s0, $s1, and $s2.

Question

- How would you compile the statement into MIPS using 3 instructions?

```
if (i==j) f = g;
else f = h;
```

where i, j, are in $t0, $t1, f, g, and h are stored in $s0, $s1, and $s2.

```
        add     $s0, $zero, $s1     # f = g
        beq     $t0,$t1,LABEL       # skip else if  i = j
ELSE    add     $s0, $zero, $s2     # f = h
LABEL   -------
```

Question

Consider the instructions

```
    slt $t0, $s1, $s2
    bne $t0, $zero, L1
L2  ……
    ……
L1
```

And consider that $s1 = 3 and $s2 = 5. What is the address branched to by the bne instruction?

Question

Consider the instructions

```
      slt $t0, $s1, $s2
     bne $t0, $zero, L1
L2    ……
      ……
L1
```

And consider that $s1 = 3 and $s2 = 5. What is the address branched to by the bne instruction?

$t0 is set to 1. Then, branch to L1.

Question

- What is the purpose of the instruction below and what it does?

```
jal Label
```

Question

- What is the purpose of the instruction below and what it does?

```
jal Label
```

It saves the address of the instruction following the jal in the $ra register and changes the PC to the address of the instruction that corresponds to Label

Question

- What is the purpose of the instruction below and what it does?

  jr $ra

Question

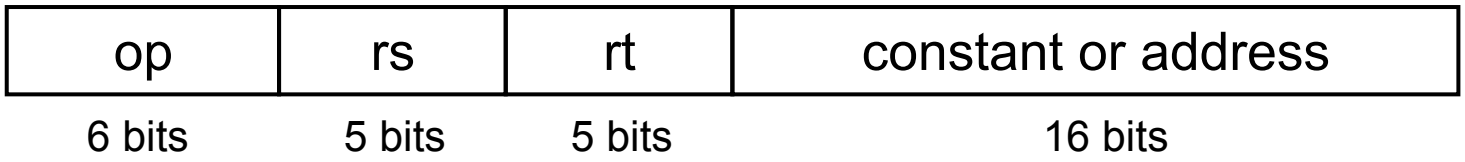
- What is the purpose of the instruction below and what it does?

  jr $ra

It jumps to the address stored in the register $ra

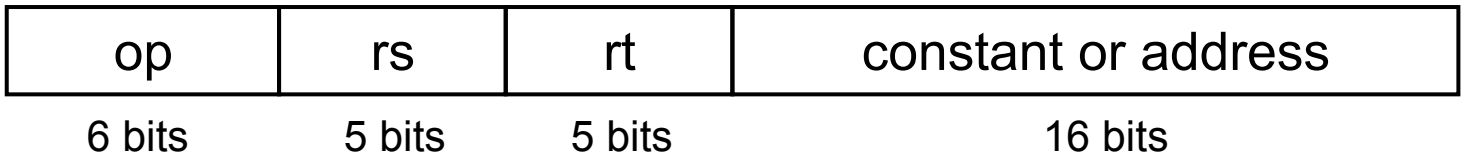
Question

- Consider the beq instruction stored at address 1000_{10}. The value of the address field is 200_{10}. What is the address of the next instruction if rs and rt are equal?



Question

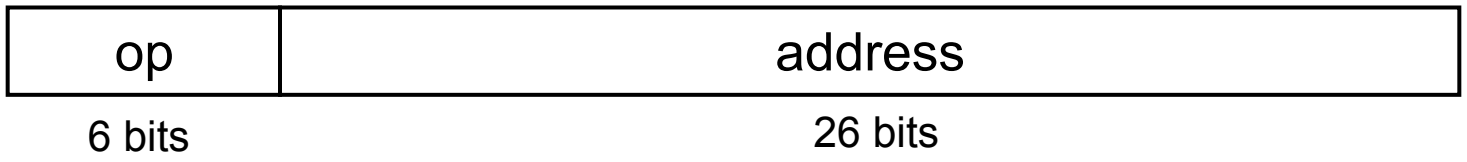
- Consider the beq instruction stored at address 1000_{10}. The value of the address field is 200_{10}. What is the address of the next instruction if rs and rt are equal?



- If rs=rt, the target address of the branch is $(1000+4) + 200*4 = 1804_{10}$

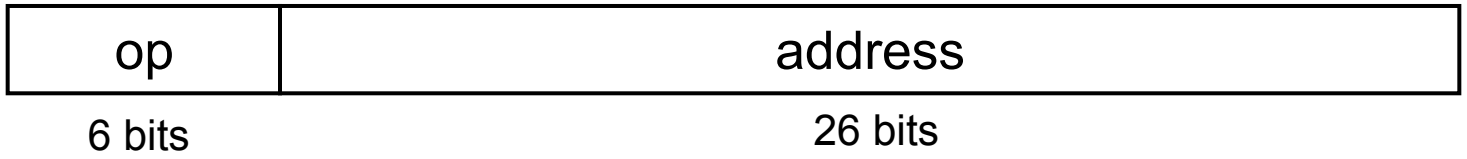
Question

- Consider the jump instruction stored at address $A80094_{16}$. The value of the address field is 800_{10}. What is address in binary of the next instruction to be executed?



Question

- Consider the jump instruction stored at address $A80094_{16}$. The value of the address field is 800_{10}. What is address in binary of the next instruction to be executed?



$A8009004_{16} + 4_{10} = A8009008_{16}$

=> A = 1010_2; $800*4=3200_{10}$ = 000 …110010000000_2

Target address = 1010: 800*4 =>1010000 …110010000000_2

Question

- Consider adding the numbers -100 and -64 represented in 2's complement using 8 bits. What is the result of the computation?

Question

- Consider adding the numbers -100 and -64 represented in 2's complement using 8 bits. What is the result of the computation?

- $100_{10} = 01100100_2 \Rightarrow -100_{10} = 10011100_2$
- $64_{10} = 01000000_2 \Rightarrow -64_{10} = 11000000_2$
- $-100-64 = 01011100_2$

- Adding two negative numbers results in a positive number => overflow

Question

Consider a 2 x 3 matrix stored in memory in *column major order*, i.e., elements are stored column by column. Each element is 4-bytes long. What is the byte offset of element i,j?

Question

Consider a 2 x 3 matrix stored in memory in *column major order*, i.e., elements are stored column by column. Each element is 4-bytes long. What is the byte offset of element i,j?

Byte offset of [i,j] =[j * 2 + i] * 4 because before [i,j] there are j full columns and i elements

Question

Write a minimal set of MIPS assembly instructions that does the identical operation as the C code below. Assume the base address of C is in $s1 and that A is in $s2. Use the minimum number of registers. Do not destroy the contents of $s1 or $s2.

A = C[0] << 4;

Question

Write a minimal set of MIPS assembly instructions that does the identical operation as the C code below. Assume the base address of C is in $s1 and that A is in $s2. Use the minimum number of registers. Do not destroy the contents of $s1 or $s2.

A = C[0] << 4;

```
lw      $t1, 0($s1)     # $t1 <- C[0]
sll     $t1, $t1,4      # $t1 <- $t1 << 4
sw      $t1, 0($s2)     # A <- $t1
```

Exercise 2.26.1

Consider the following MIPS code with the following initial values: $t1 = 10 and $s2 = 0.

```
LOOP: slt    $t2, $0, $t1
      beq    $t2, $0, DONE
      subi   $t1, $t1, 1
      addi   $s2, $s2, 2
      j      LOOP
DONE:
```

What is the final value of $s2?

Exercise 2.26.1

Consider the following MIPS code with the following initial values: $t1 = 10 and $s2 = 0.

```
LOOP: slt   $t2, $0, $t1
      beq   $t2, $0, DONE
      subi  $t1, $t1, 1
      addi  $s2, $s2, 2
      j     LOOP
DONE:
```

What is the final value of $s2?

Number of loop executions:

$t1 at top = 10; $t1 at bottom = 9

…

t1 at top = 1; $t1 at bottom = 0 → 10 executions →$s2 = 2x10 = **20**

Question

Describe what the following MIPS code does.

```
        addi    $s2,$0,$0
        addi    $t1,$0,$0
LOOP    lw      $s1,0($s0)
        add     $s2,$s2,$s1
        addi    $s0,$s0,4
        addi    $t1,$t1,1
        slti    $t2,$t1,100
        bne     $t2,$0,LOOP
DONE:
```

Question

Describe what the following MIPS code does.

```
        addi    $s2,$0,$0
        addi    $t1,$0,$0
LOOP    lw      $s1,0($s0)
        add     $s2,$s2,$s1
        addi    $s0,$s0,4
        addi    $t1,$t1,1
        slti    $t2,$t1,100
        bne     $t2,$0,LOOP
DONE:
```

Code meaning: store in $s2 the sum of all 100 words stored starting at address $s0

Question

Consider a multiprocessor with p processors. Assume that 25% of the instructions of a program can be executed in parallel using all p processors. The remaining 75% of the instructions have to be executed sequentially. Assume that the time to execute the program sequentially (i.e., using only one processor) is Ts. Give an expression for S(p), the speedup obtained when using p processors.

What is the maximum possible speedup? i.e.(lim S(p) when p -> ∞)

Question

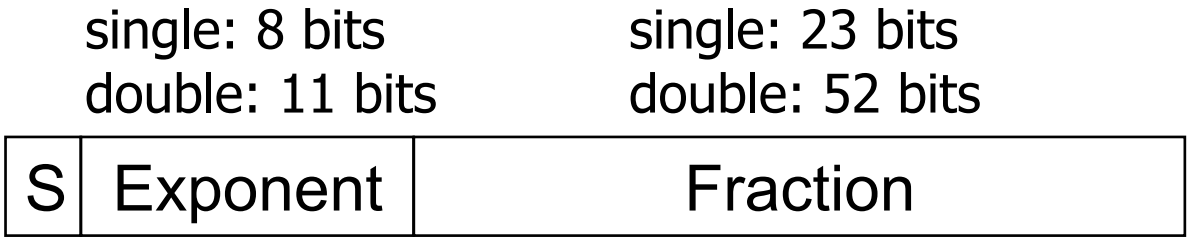
Consider a multiprocessor with p processors. Assume that 25% of the instructions of a program can be executed in parallel using all p processors. The remaining 75% of the instructions have to be executed sequentially. Assume that the time to execute the program sequentially (i.e., using only one processor) is Ts. Give an expression for S(p), the speedup obtained when using p processors.

What is the maximum possible speedup? i.e.(lim S(p) when p -> ∞)

S(p) = Ts / (0.75 Ts + 0.25 Ts/p) = 1 / (0.75 + 0.25/p)

lim S(p) when p -> ∞ = 1 / 0.75 = 4/3 = 1.33

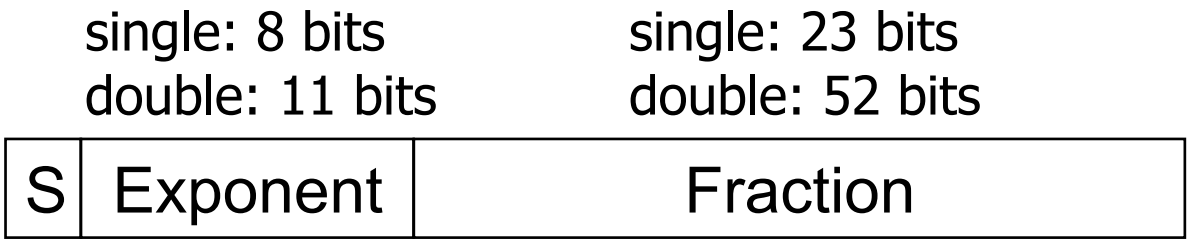
Floating Point



Single: Bias = 127; Double: Bias = 1023

What is the value of the exponent field and the fraction for the single precision representation of 1.75?

Floating Point



Single: Bias = 127; Double: Bias = 1023

What is the value of the exponent field and the fraction for the single precision representation of 1.75?

1.75 = 1 + 0.75 = 1 + 0.5 + 0.25
Fraction= 1100000…000
Exponent = actual + bias = 0 + 127 = 127 = 01111111_2

The Processor

- What is a single cycle datapath?
- What is the duration of a cycle in a single-cycle datapath?
- How does a pipelined architecture differ from a single cycle datapath?
- What is the duration of a cycle in a pipelined architecture?

The Processor

- What is a single cycle datapath?

All instructions take one cycle to execute.

- What is the duration of a cycle in a single-cycle datapath?
- How does a pipelined architecture differ from a single cycle datapath?
- What is the duration of a cycle in a pipelined architecture?

The Processor

- What is a single cycle datapath?

All instructions take one cycle to execute.

- What is the duration of a cycle in a single-cycle datapath?

The time to execute the longest instruction

- How does a pipelined architecture differ from a single cycle datapath?

- What is the duration of a cycle in a pipelined architecture?

The Processor

- What is a single cycle datapath?

All instructions take one cycle to execute.

- What is the duration of a cycle in a single-cycle datapath?

The time to execute the longest instruction

- How does a pipelined architecture differ from a single cycle datapath?

Instruction execution is broken down into stages. Different instructions can be at different stages of execution

- What is the duration of a cycle in a pipelined architecture?

The Processor

- What is a single cycle datapath?

All instructions take one cycle to execute.

- What is the duration of a cycle in a single-cycle datapath?

The time to execute the longest instruction

- How does a pipelined architecture differ from a single cycle datapath?

Instruction execution is broken down into stages. Different instructions can be at different stages of execution

- What is the duration of a cycle in a pipelined architecture?

The time needed to execute the longest stage

The Processor

- What is the purpose of the control unit?

Generate selector bits that control the various multiplexers and units of the processor.

- Discuss the inputs of the control unit

- Discuss some of the outputs of the control unit

The Processor

- What is the purpose of the control unit?

Generate selector bits that control the various multiplexers and units of the processor.

- Discuss the inputs of the control unit

The bits of the instruction (e.g., opcode and function codes)

- Discuss some of the outputs of the control unit

The Processor

- What is the purpose of the control unit?

Generate selector bits that control the various multiplexers and units of the processor.

- Discuss the inputs of the control unit

The bits of the instruction (e.g., opcode and function codes)

- Discuss some of the outputs of the control unit

MemRead, MemWrite, RegWrite, Branch, ALUSrc, MemToReg

The Processor

- What are the phases of a MIPS pipeline?

Instr. Fetch, Instr Decode, Execute, Memory Access, WriteBack

- What is duration of each phase in cycles?

- Consider the following instruction sequence:

  add $t5, $t1, $t2

  add $t6, $t3, $t4

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

The Processor

- What are the phases of a MIPS pipeline?

Instr. Fetch, Instr Decode, Execute, Memory Access, WriteBack

- What is duration of each phase in cycles?

1 cycle

- Consider the following instruction sequence:

  add $t5, $t1, $t2

  add $t6, $t3, $t4

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

The Processor

- What are the phases of a MIPS pipeline?

Instr. Fetch, Instr Decode, Execute, Memory Access, WriteBack

- What is duration of each phase in cycles?

1 cycle

- Consider the following instruction sequence:

  add $t5, $t1, $t2

  add $t6, $t3, $t4

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

No data hazard

The Processor

- Consider the following instruction sequence:

  add $t3, $t1, $t2

  add $t6, $t3, $t5

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

Yes by 2 cycles

- Consider the instruction sequence above:

  Is there a data hazard assuming forwarding is used? If yes, by how many cycles?

The Processor

- Consider the following instruction sequence:

  add $t3, $t1, $t2

  add $t6, $t3, $t5

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

Yes by 2 cycles

- Consider the instruction sequence above:

  Is there a data hazard assuming forwarding is used? If yes, by how many cycles?

No. $t3 can be sent by the end of EX of the first add to the input of EX for the second add

The Processor

- Consider the following instruction sequence:

  lw $t3, 16($t3)

  add $t6, $t3, $t5

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

- Consider the instruction sequence above:

  Is there a data hazard assuming forwarding is used? If yes, by how many cycles?

The Processor

- Consider the following instruction sequence:

  lw $t3, 16($t3)

  add $t6, $t3, $t5

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

Yes, 2 cycles

- Consider the instruction sequence above:

  Is there a data hazard assuming forwarding is used? If yes, by how many cycles?

The Processor

- Consider the following instruction sequence:

  lw $t3, 16($t3)

  add $t6, $t3, $t5

Is there a data hazard assuming no forwarding? If yes, by how many cycles?

Yes, 2 cycles

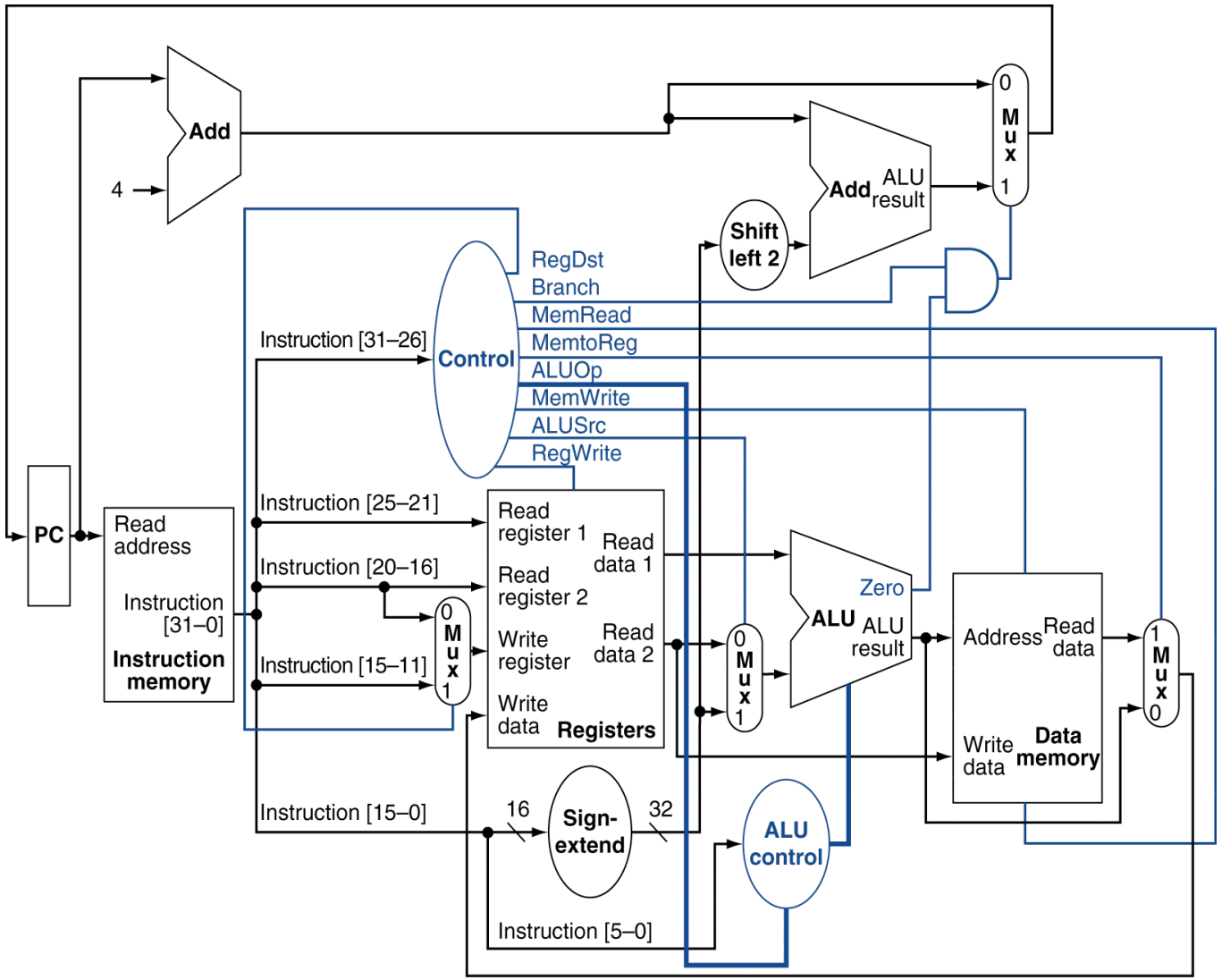
- Consider the instruction sequence above:

  Is there a data hazard assuming forwarding is used? If yes, by how many cycles?

Yes, one cycle

What is the value of RegDst for add $t1,$t2,$t3?

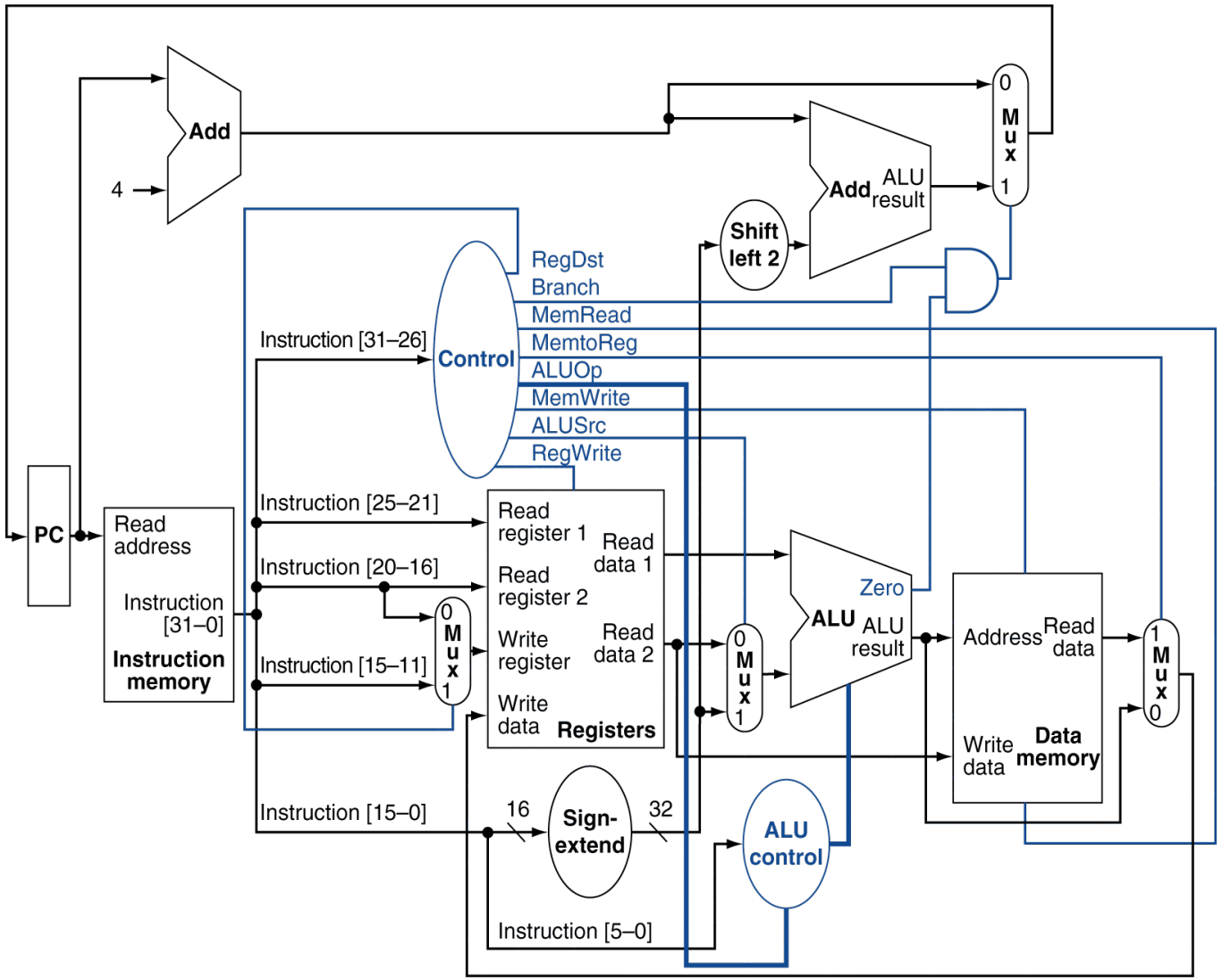
Hint: destination address in bits 15-11.



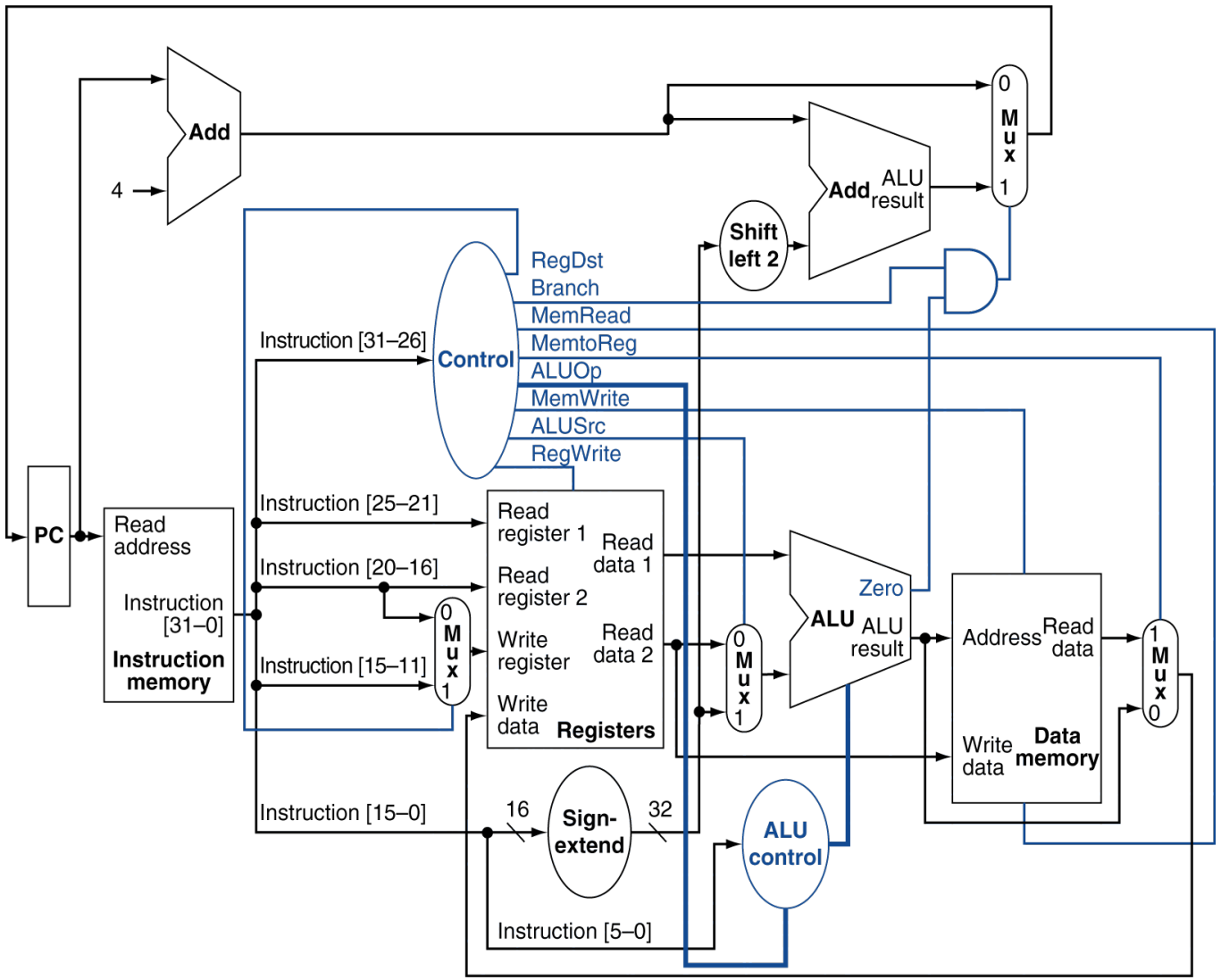
What is the value of RegDst for add $t1,$t2,$t3?

Answer: 1

Hint: destination address in bits 15-11.

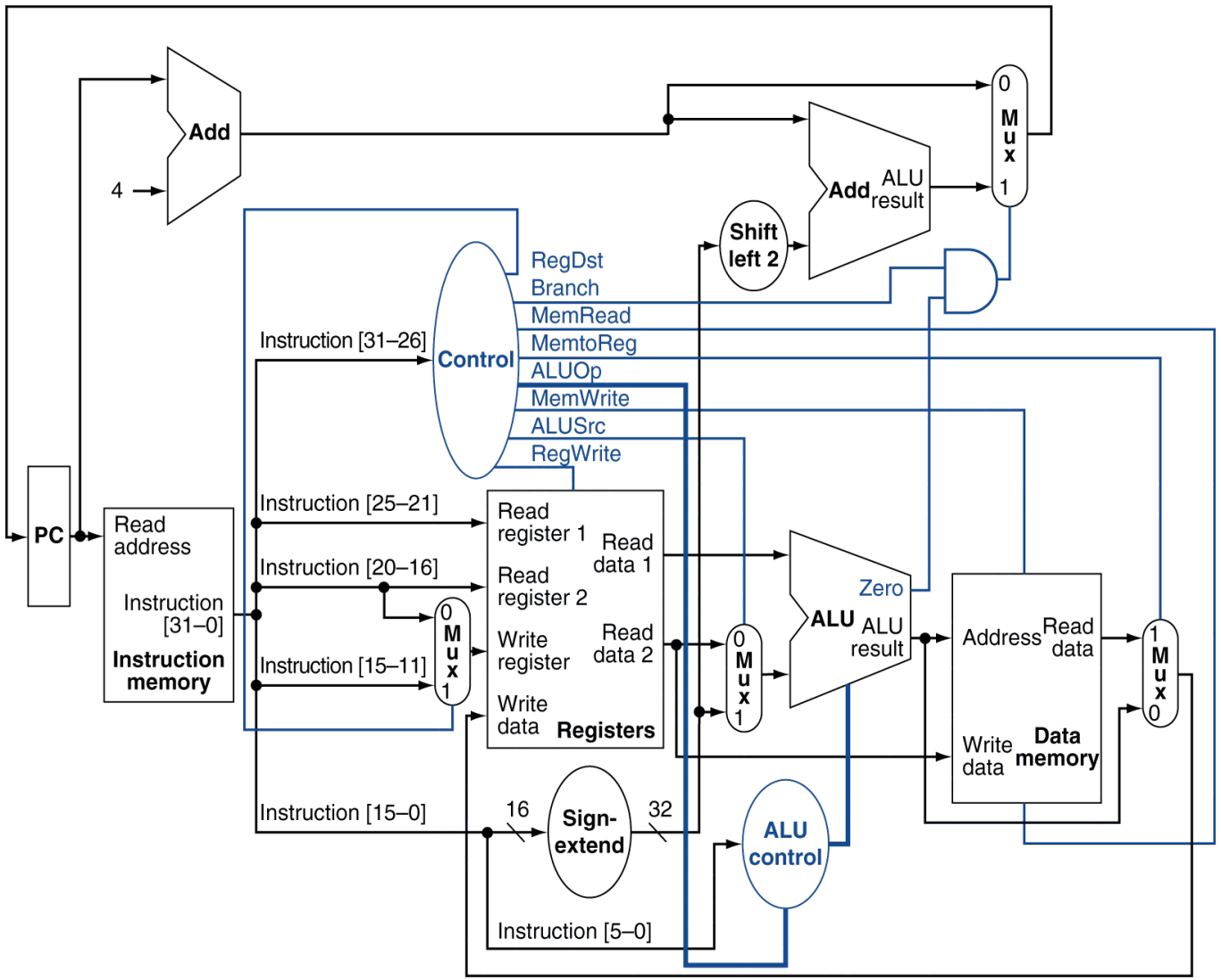


What is the value of ALUSrc for addi $t1,$t2,4?

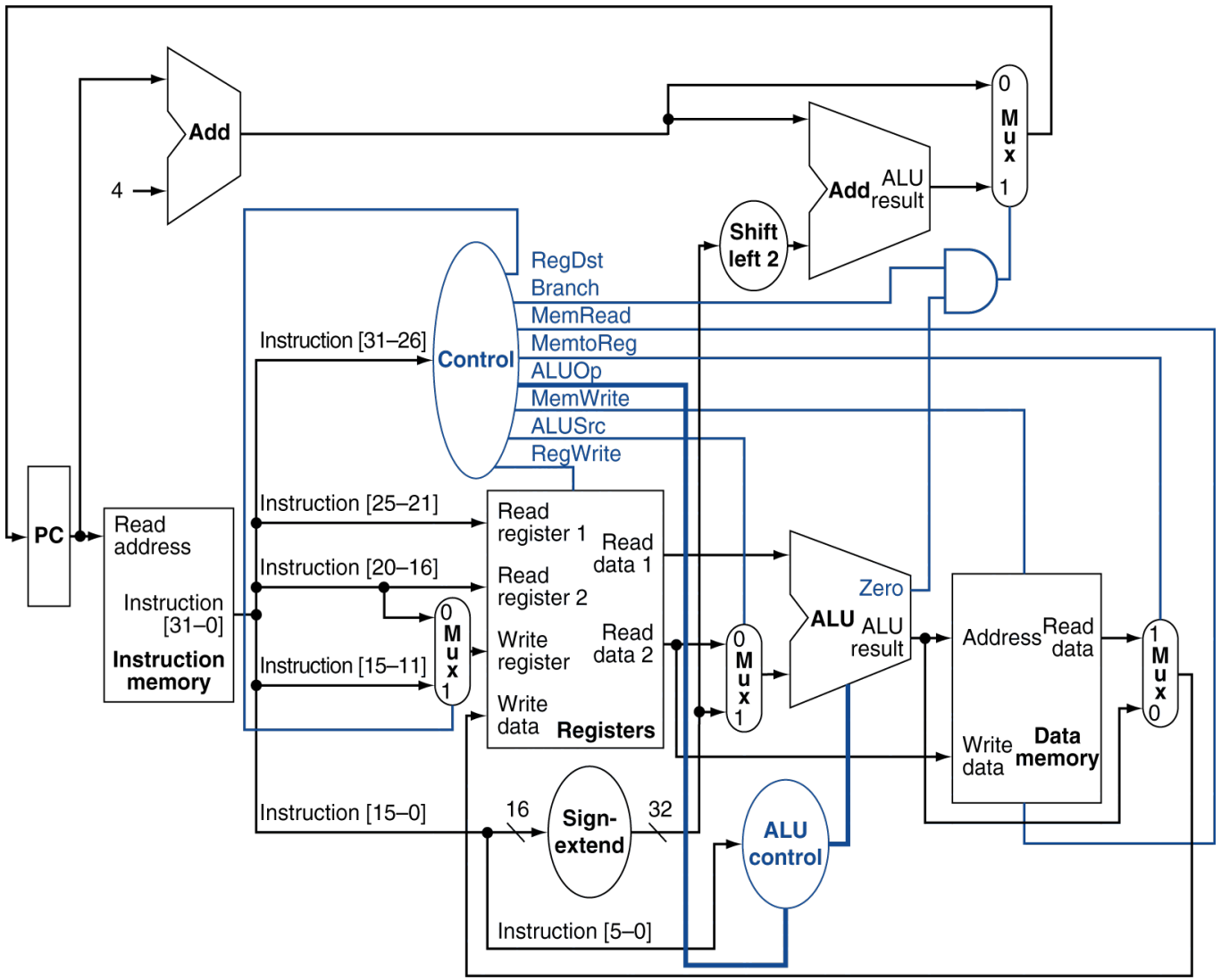


What is the value of ALUSrc for addi $t1,$t2,4?

A: 1

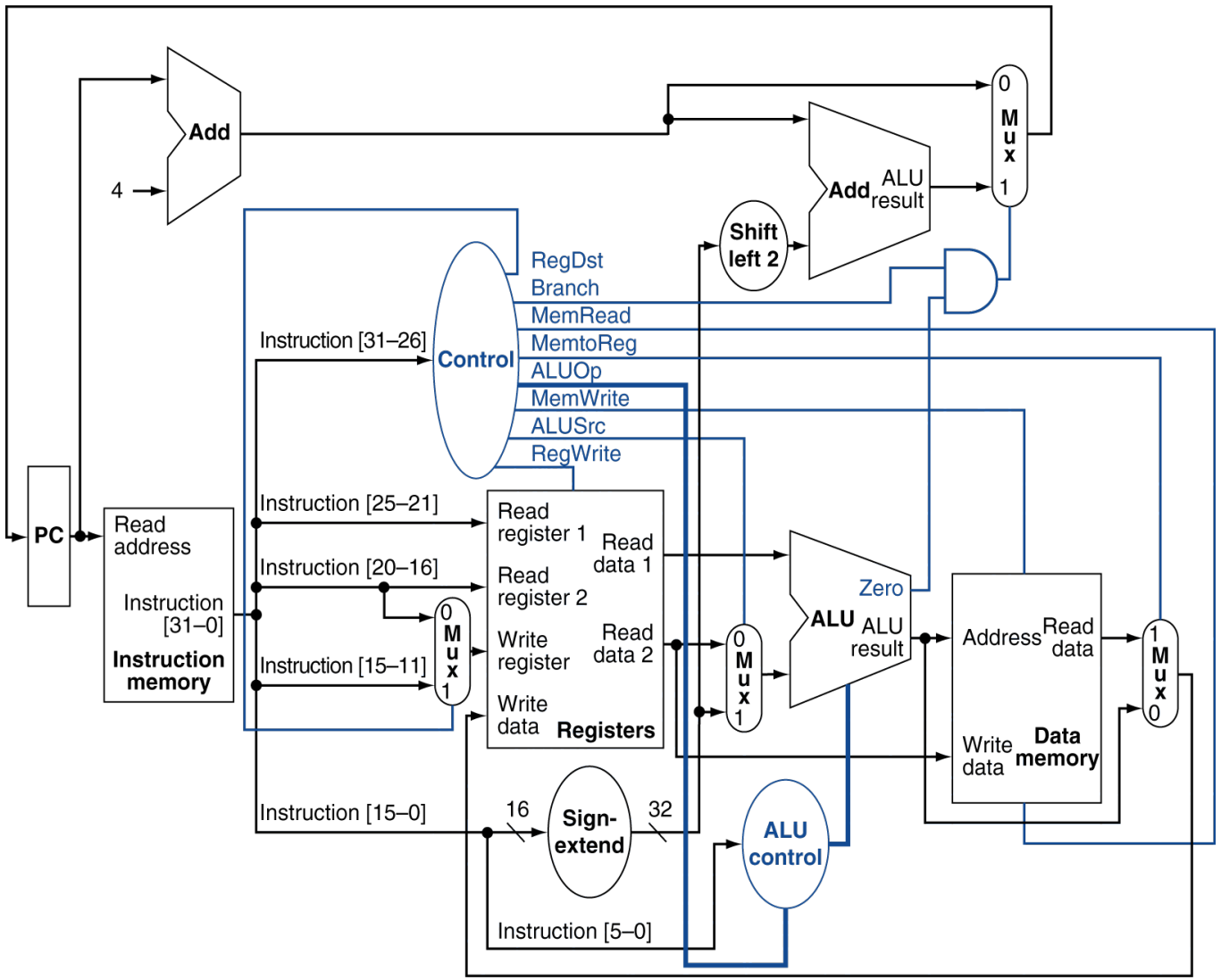


What is the value of ALUSrc for lw $t2,4($t3)?

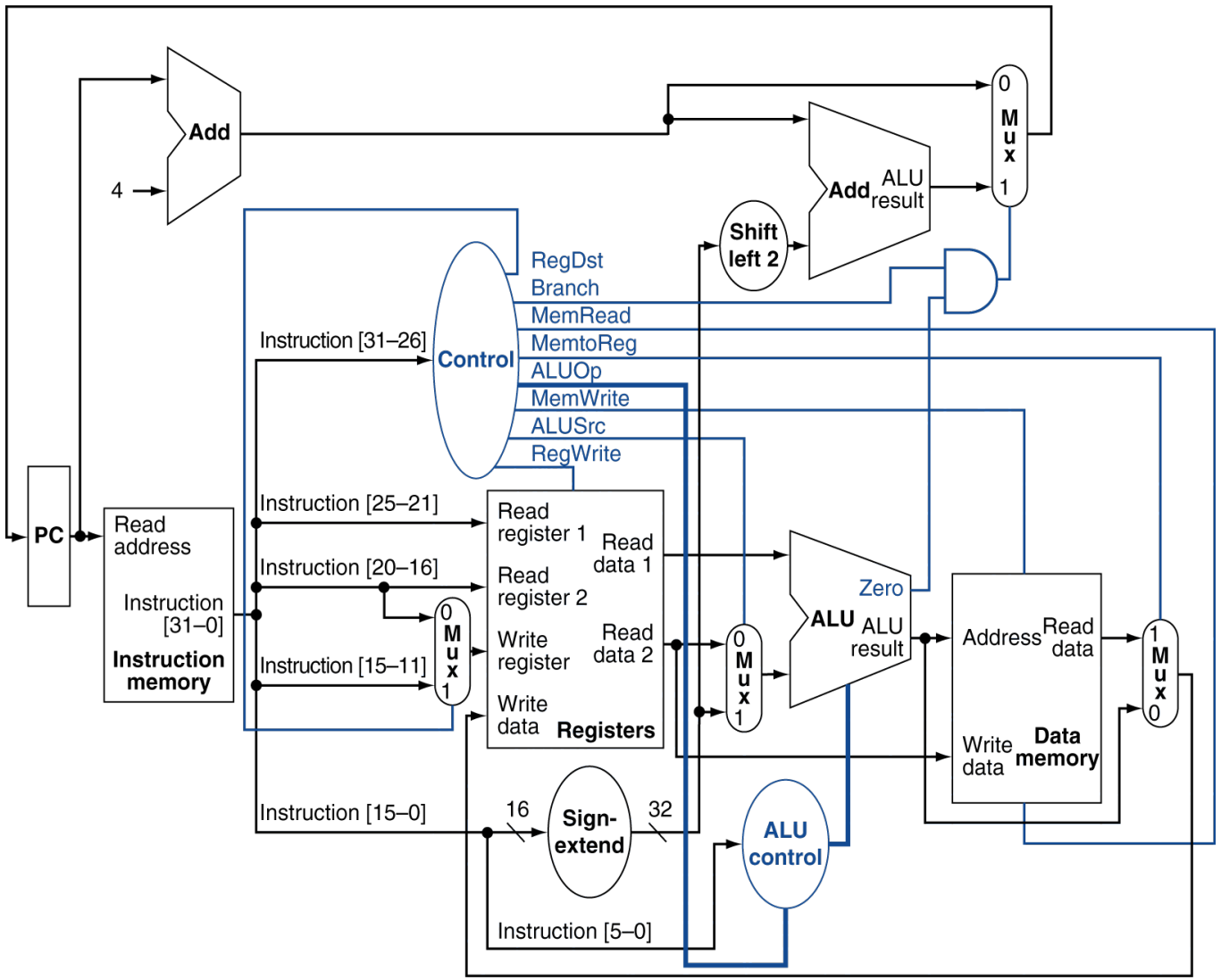


What is the value of ALUSrc for lw $t2,4($t3)?

A: 1



What is the value of ALUSrc for beq $t2,$t3,exit?



What is the value of ALUSrc for beq $t2,$t3,exit?

A: 0