CS211 Midterm Example

October 28, 2025

Example Midterm

Name: _____

Instructions: Print your name at the top of each page. Mark your answers directly on the test, with an x on your chosen solutions. Multiple choice questions have one solution each.

```
addq %rax, %rbx
movq (%rbx), %rax
```

1) Given the above code, which statement is **true**?
(a) There is a data hazard for rbx, whose value must be updated before being used in movq instruction.
(b) There is a data hazard for rax, whose value must be updated before being used in the movq instruction.
(c) The data hazard between the instructions **cannot** be resolved by forwarding.
(d) There is no hazard in the above instructions.

2) Which of the following statements about pipelines is **true**?
(a) Our ability to design longer pipelines is only limited by transistor density in silicon.
(b) Shorter pipelines have faster clock rates.
(c) Longer pipelines resolve data dependency hazards.
(d) None of the above are true.

3) Which instruction allocates memory on the stack?
(a) `pushq %rbp`
(b) `movq %rsp, %rbp`
(c) `subq $16, %rsp`
(d) `movq %rdi, -8(%rbp)`

4) How are program arguments passed?
(a) In agreed upon registers, such as %rdi.
(b) On the stack, if there are not enough registers.
(c) Always on the stack.
(d) A and B

5) Which of the following will **not** resolve a read-after-write hazard between two instructions?
(a) A no-op bubble in between them.
(b) Data forwarding from the first to the second.
(c) An out of order executed instruction in between them.
(d) Branch prediction.

6) Incrementing the largest floating point value less than 1 will:
(a) Overflow from the significand and increment the exponent.
(b) Result in an exponent value equal to the bias.
(c) Leave the value in the significand equal to 0.
(d) All of the above.

7) What is immediate addressing?
(a) When an operand loads data from an offset into the memory referred to by a register.
(b) When the operand is specified within the instruction itself.
(c) When an operand is within one of the general purpose registers.
(d) None of the above.

8) Which of these statments is **false** about the x86-64 instruction set?
(a) It supports C-style array access through indirect memory access instructions.
(b) Compared to other instruction sets, the x86-64 ISA has a large number of instructions.
(c) The x86-64 ISA only supports 64-bit operands.
(d) The same registers can be referred to with multiple names to use different amounts of the registers.

```
pushq %rbp
movq %rsp, %rbp
```

9) Which statement accurately describes the code above?
(a) Before calling a function, the caller adds an argument onto the stack.
(b) This allocates memory on the stack.
(c) In a called function, the code saves the caller's base stack pointer before replacing it with the function's new base stack pointer.
(d) This frees memory on the stack.

```
movq %rbp, %rsp
popq %rbp
```

10) Which statement **does not** accurately describe the code above?
(a) In a called function, this restores the callers %rbp and entry %rsp, restoring the frame of the caller.
(b) This code prevents a stack overflow when stack-using functions are called repeatedly.
(c) This allocates memory on the stack.
(d) This frees memory on the stack.

11) Which of these is **not** an advantage of floating point values over fixed point?
(a) Floating point operations are supported directly in hardware.
(b) Floating point values can express much greater ranges given the same number of bits.
(c) Floating point formats are standardized, making them easily shared between systems.
(d) Floating point values can represent fractional values.

12) Two consecutive addq instructions use the same destination operand. What is **true** about this situation?
(a) In a non-pipelined CPU, there is no hazard.
(b) In a pipelined CPU, there is no data hazard.
(c) Consecutive instructions of that type are not allowed.
(d) The instructions will need to be reordered.

13) The instruction '`movq (%rax), %rax`' is issued during the instruction fetch stage on clock cycle 0. In the five stage pipeline discussed in class, which is the **earliest stage** after which the new value of %rax will be **available for forwarding** to a subsequent instruction? (a) Move instructions cannot be forwarded.
(b) Clock cycle 2.
(c) Clock cycle 4.
(d) None of the above.

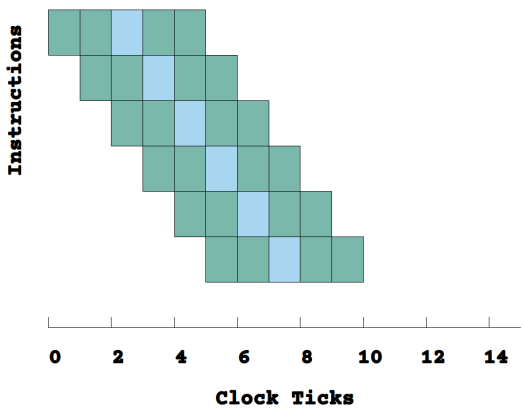

Figure 1: For question 13

Name: _____

14) Consider the five stage pipeline discussed in class. A designer tells you that they plan to split the execution phase into two steps, with each step taking half of the current 10ms clock period. All of the other stages run in less than 5ms. However, the two new execution stages will require a pipeline buffer with a delay of 0.1ms. Which of the following is true?
(a) The execution stage cannot be split because it would make forwarding too complicated.
(b) A new clock rate can be used that is twice the current clock rate.
(c) A new clock rate can be used that is slightly less than twice the current clock rate.
(d) A new clock rate can be used that is slightly faster than twice the current clock rate.

15) Consider a superscalar pipeline two instructions wide and 10 instructions deep. Through prefetching, the pipeline is always filled with instructions. If a data hazard causes an unavoidable bubble of two instructions before the next instruction can run, but does not affect any other instructions, what is the lowest that CPU utilization will drop to because of the bubble?
(a) 20%
(b) 50%
(c) 80%
(d) 90%

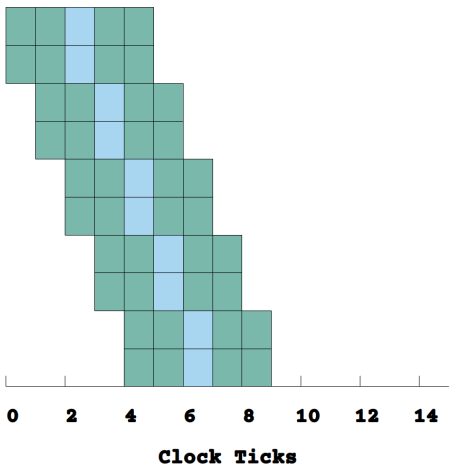

Figure 2: For question 15

16) Consider this code:

```
float mathFloats(float a, float b) {
    return a + b - a;
}
```

Which statement about this code is **true**?
(a) `return b;` would return the same value as the function.
(b) `return a - a + b;` would return the same value as the function.
(c) This code will not compile.
(d) None of the above are true.

17) Consider an fp-8 type with 1 bit sign, 4 bits exponent, and 3 bits significand. This type does not have a way to represent NaN or +/- infinity. Exponent bias is 8. What is the maximum value this type can store? You may write the equation rather than reducing it to a single number.

Answer: _____

18) Consider this code:

```
for (int i = 0; i < strlen(argv[1]); ++i) {
    puts(argv[i]);
}
```

What would make this code run faster?
(a) Changing the `++i` to `i++`
(b) Changing the for loop to a while loop.
(c) Convert the loop into a recursive function.
(d) Move the call to `strlen` out of the loop.

19) Write a function that returns nothing and accepts two arguments: a pointer to an int array and the number of elements of that array. The function should set every element in the array with the index of that element squared (e.g. $0^2, 1^2, ...$)

20) This code will push a value onto the stack:

```
long long value;
asm("push %0" : : "r" (value) :);
```

This code will retrieve a value from the stack:

```
long long out;
asm("pop %0" : "=r" (out) : :);
```

Write a function that returns nothing and accepts two arguments: a pointer to a long long array and the number of elements of that array. Use the stack to reverse the array. Use no variables other than loop counters and the stack.