# CS 3330 Exam 1 Fall 2018

EXAM KEY Name:

Computing ID: <u>KEY</u>

Letters go in the boxes unless otherwise specified (e.g., for C 8 write "C" not "8"). Write Letters clearly: if we are unsure of what you wrote you will get a zero on that problem. Bubble and Pledge the exam or you will lose points.

**Assume** unless otherwise specified:

- little-endian 64-bit architecture
- %rsp points to the most recently pushed value, not to the next unused stack address.
- questions are single-selection unless identified as select-all

Variable Weight: point values per question are marked in square brackets.

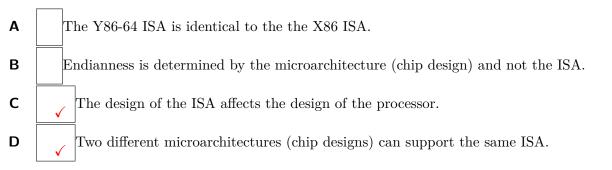
Mark clarifications: If you need to clarify an answer, do so, and also add a **\*** to the top right corner of your answer box.

.....

Question 1 [2 pt]: Suppose one takes an unpipelined multiply circuit which takes 500 nanoseconds to perform a multiplication, and divides into a pipelined multiply circuit which has ten stages, each of which requires 100 nanoseconds, including register delays.

If we use this pipelined circuit to perform many multiplications as fast as possible, how long after the result of the first multiplication is available will the result of the third multiplication be available? Write your answer as a base-10 number of nanoseconds, or write "unknown" if there is not enough information to answer.

Question 2 [2 pt]: Which of the following are true about Instruction Set Architectures (ISAs)? Place a  $\checkmark$  in each box corresponding to a correct answer and leave other boxes blank.



Question 3 [2 pt]: The linker sometimes needs to know where the first instruction of functions are in the object files it reads. The most likely way for it to find the location of the function **foo** is by

**A** searching for instructions that save callee-saved registers in the appropriate object file's machine code

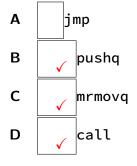
searching the object file for the string "foo:" Β

С finding the name **foo** in the symbol table of the object file that calls the function foo

finding the relocation table entry corresponding to the call of the function foo D

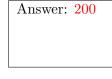
finding the name **foo** in the symbol table of the object file that defines the function foo

Question 4 [2 pt]: In our textbook's single-cycle processor, which of the following Y86 instructions require the result of a calculation typically performed by the ALU? Place a  $\checkmark$  in each box corresponding to a correct answer and leave other boxes blank.



Ε

Answer: E



KEY

Variant E page 3 of 8

#### Information for questions 5–6

Consider the following Y86-64 program and its corresponding machine code. (Each line of machine code is written as a sequence of hexadecimal byte values, prefixed by the memory address of the first byte.)

0x000: 30 f4 30 00 00 00 00 00 00 00 | irmovq \$0x30, %rsp 0x00a: 30 f0 25 00 00 00 00 00 00 00 | irmovg \$0x25, %rax 0x014: 61 04 subg %rax, %rsp 0x016: b0 0f popq %rax 0x018: 00 halt

Assume any memory location not specified above is initially 0. 0x25 in decimal is 37 and 0x30 in decimal is 48. (Recall that subq A, B computed B-A.)

Question 5 [2 pt]: (see above) After this assembly snippet runs, what is the value of %rsp?

- Α 0x00
- **B** 0x05
- **C** 0x0B (11 in decimal)
- **D** 0x13 (19 in decimal)
- **E** 0x1B (27 in decimal)
- F  $0 \times 25$  (37 in decimal)
- **G** 0x28 (40 in decimal)
- **H** none of the above

Question 6 [2 pt]: (see above) After this assembly snippet runs, what is the value of %rax?

- 0x0 Α
- 0x0B (11 in decimal) В
- С **0x13** (19 in decimal)
- D  $0 \times 25$  (37 in decimal)
- Ε 0x25f0
- F 0x30f0
- G 0xf025
- H 0xf025 0000 0000 0000
- L there is not enough information to determine this
- none of the above J

Question 7 [2 pt]: Assume we have a variable arr which was set as follows:

char \*arr[5] = {"We", "live", "in", "a", "society"};

Which of the following C expressions are true? Place a  $\checkmark$  in each box corresponding to a correct answer and leave other boxes blank.

A 
$$\checkmark$$
 \*(\*(arr+4)+3) == arr[2][0] 'i' in "society" == 'i' in "in"  
B  $\checkmark$  \*\*(arr+4) + 3 == arr[4][3] would come out to be 'v' == 's' because we add 3 to the  
ascii value for 's'  
C  $\checkmark$  \*(arr[1]) == arr[0][1] 'l' != 'e' -> makes sure the student applies the correct order of  
operations  
D  $\checkmark$  \*((\*arr)+1) == \*(arr[1] + 3) 'e' from "We" == 'e' from "live"

Answer: D	

Answer: E

#### Information for questions 8–9

Suppose we wanted to extend the single-cycle processor design discussed in lecutre and our textbook to support a new instruction **pop2g** rA, rB which would pop two values from the stack. So, for example,

```
irmovq $0x1000, %rsp
irmovg $0x10, %rax
irmovq $0x20, %rbx
pushq %rax
pushq %rbx
pop2q %r8, %r10
```

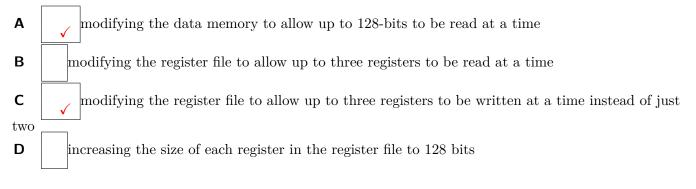
would set %r8 to 0x20 and %r10 to 0x10 and %rsp to 0x1000. (Recall that subg A, B computes B-A.)

(see above) The new pop2q instruction could have the same machine code layout Question 8 [2 pt]: as which of the following Y86-64 instructions: dropped (outside of TPEGS)

- Α rrmovq
- popq В
- С irmovq
- D pushq
- Е none of the above

Answer: A

Question 9 [2 pt]: (see above) Adding this instruction to the single-cycle processor (while still having every instruction execute in a single cycle) would require which of the following? Place a  $\sqrt{in}$  each box corresponding to a correct answer and leave other boxes blank.



Information for questions 10–13

For the following questions, consider the following C code and assume doubles take up 8 bytes, and all pointers take up 8 bytes.

```
double array[6] = {-1, 0, 1, 2, 3, 4};
double *pointer;
pointer = array + 1;
pointer += 1;
*pointer = 0;
pointer += 3;
*pointer *= pointer[0];
```

Question 10 [2 pt]: (see above) What is the final value of array?

A {-1, 0, 0, 2, 3, 16}
B {-1, 0, 0, 2, 3, -4}
C {-1, 1, 1, 5, 3, 4}
D {-1, 0, 1, 2, 3, 4}

 ${\sf E}$   $\,$  none of the above; it will have some other value or the code is likely to crash  $\,$ 

Question 11 [1 pt]: (see above) What is the value of sizeof(\*array)?

A it depends whether pointer += 1 has been executed yet

- **B** 48
- **C** 8
- **D** it's a compile-time error
- ${\bm \mathsf E} \quad {\rm none \ of \ the \ above}$

### **Question 12** [1 pt]: (see above) What is the value of sizeof(array)?

A it depends whether pointer += 1 has been executed yet

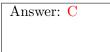
- **B** 8
- $\boldsymbol{\mathsf{C}} \quad \mathrm{it's \ a \ compile-time \ error}$
- **D** 48
- ${\bm \mathsf E} \quad {\rm none \ of \ the \ above}$

Question 13 [2 pt]: (see above) Which of the following is/are true?

A sizeof(pointer) == sizeof(pointer[0])
B if the array goes out of scope, one could still safely access its values using a copy of pointer
C the code above may crash (such as from a segmentation fault) because space for pointer is not
allocated before assigning array + 1 to it

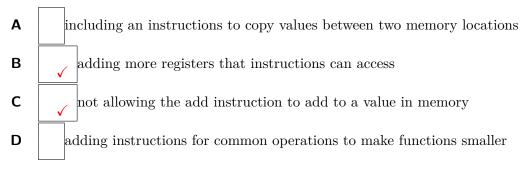
D / \*pointer == pointer[0]

Answer: A



Answer: D

Question 14 [2 pt]: Which of the following is more typical of a RISC instruction set than a CISC instruction set? Place a  $\checkmark$  in each box corresponding to a correct answer and leave other boxes blank.



### Information for questions 15–16

Consider the following Y86-64 code:

```
start:
    subq %rbx, %rax
    irmovq $0xb, %rdi
    subq %rax, %rdi
    jle start
```

(Recall that subq A, B computes B-A.)

Question 15 [2 pt]: (see above) If %rax corresponds to the variable a and %rbx corresponds to the variable b, which of the following C snippets is equivalent to this code? (You may assume integer overflow does not occur and that all variables are 8-byte signed long ints.)

Α	<pre>do { a -= b; } while (a &gt;= 11);</pre>
В	<pre>do { a -= b; } while (a &lt;= 11);</pre>
С	<pre>do { a -= b; } while (a &lt; -11);</pre>
D	<pre>do { a -= b; } while (a &gt; 11);</pre>
Е	none of the above

Answer: A

Question 16 [2 pt]: (see above) What describes the possible values of the condition codes when the above assembly code breaks out the loop?



- **B** either SF = 0 or ZF = 1
- **C** either SF = 1 or ZF = 1
- **D** both SF = 0 and ZF = 0
- **E** none of the above

Answer: D

Question 17 [2 pt]: Given a long \*a stored in the register %rax and a long \*b stored in the register %rbx, the C code \*a = \*(b + 1) + 1 is equivalent to the x86-64 assembly snippet:

## Α

```
movq 8(%rbx), %r8
leaq 1(%r8), %r8
movq %r8, (%rax)
B
leaq 8(%rbx), %r8
addq %r8, 8(%rax)
C
movq 8(%rbx), %r8
leaq 8(%r8), %r8
movq %r8, (%rax)
D
leaq 1(%rax, %rbx, 8), %rax
E
movq 8(%rbx), %r8
leaq 8(%r8), %rax
```

Answer: A

Question 18 [2 pt]: When executing the ret instruction on the single-cycle processor described in lecture and our textbook, the value of the stack pointer in the register file is updated \_\_\_\_\_\_ return address is read from the data memory.

- **A** before
- **B** at about the same time
- $\mathbf{C}$  after
- ${\boldsymbol{\mathsf{D}}}$   $\,$  it depends on the relative speeds of the ALU and data memory
- ${\ensuremath{\mathsf{E}}}$  the ret instruction doesn't change the stack pointer

**Question 19 [2 pt]:** Which of the following statements are true about the registers files included with HCLRS and used in our textbook's processor designs?

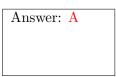
- **A** register values are written in the register file on the rising edge of the clock.
- **B** register values are written in the register file during the clock cycle.
- **C** the register file only allows reads.
- **D** the register file only stores values for one clock cycle.
- **E** register values are written in the register file on the falling edge of the clock.

Question 20 [2 pt]: Given a 16-bit unsigned short x, which of the following C snippets result in x with the least significant 4 bits and most significant 4 bits swapped? (For example, if 0xABCD should become 0xDBCA.)

```
A (((x<<12) >> 12) | (x>>12) | (x & (~0xF00F)) & 0xFFFF
```

- **B** ((x >> 12) | (x << 12)) & 0xFFFF
- C ((x & 0xF) << 12) | ((x >> 12) & 0xF) | (x & 0x0FF0)
- **D** (x & 0x0FF0) | ((x >> 12) & 4) | ((x & 4) << 12)
- ${\bf E} \quad {\rm none \ of \ the \ above}$

Answer: C



Answer: C

Email ID: <u>KEY</u>

Question 21 [2 pt]: Consider the following HCLRS snippet:

```
register i0 {
    a : 64 = 10;
    b: 64 = 1;
}
i_a = 0_a - 0_b;
i_b = 0_b + 1;
```

Answer: 7

which defines a register **a** with input wire **i\_a** and output wire **0\_a** and a register **b** with input wire i\_b and output wire 0\_b. During cycle 1, the value of 0\_a is 10 and the value of **0\_b** is 1. What is the value of **0\_a** during cycle 3? Write your answer as a base-10 number.

Question 22 [2 pt]: If x and y are unsigned ints between 0 and 1000 inclusive, then which of the following C expressions are always true? Place a  $\sqrt{in}$  each box corresponding to a correct answer and leave other boxes blank.

.....

Α	(x & y) < y
В	✓ (x   y) >= x
	((x >> 3) & 0x7) == ((x & 0x3F) >> 3)
D	((x + y) << 1) >= (x   (y << 1))

# Pledge:

On my honor as a student, I have neither given nor received aid on this exam.

Your signature here