CS 2200 - Introduction to Systems

Fall 2016

Homework 2

Rules:

- Please print a copy of the assignment and handwrite your answers. No electronic submissions are allowed. **Please print as one double-sided page.**
- This is an individual assignment; No collaboration is permitted.
- **Due Date: 14th September 2016 – 6:05 PM (At the start of Recitation)**. Bring your **Buzzcards.**

Name (please print): ANSWER KEY **GTLogin:** answerkey2 **Section** —

1. For this question, use the LC-2200 calling convention defined in the textbook
   a) Assume for the code below, that we are in the program state where the processor is about to execute the first actual executable statement in the bar function. Identify whether the caller (foo) or callee (bar) is responsible for placing the item on the stack.
   b) Given that the value of $sp was 0x2040 just before foo started setting up the stack, use the table below to record the layout of the stack from 0x2040 and above when point B is reached. Be as inclusive as possible. You may not necessarily need to use all of the rows.

```
int foo(int x, int y) {
      int z;
      /* some code that uses $s1, $s2 and $t0 */
      z = bar(x, y, z, 1);
      /* some more code that uses $s1 and $t0 */
}

int bar(int a, int b, int c, int d) {
      int k, j;
      /* POINT B */
      /* some code that uses $s1, $s2
and $t0 */
}
```

item:	who placed it:
$s2	Foo and Bar
$ra	Foo
d	Foo
k	Bar
z	Foo
$t0	Foo
$s1	Foo and Bar
j	Bar
$fp	Foo and Bar

Note: "Foo and Bar" also accepting only "Bar"

(+2) for each correct answer

	k
	j
	$s2
	$s1
	$FP
	$ra
	1
	$t0
	z
	$s2
	$S1
	$FP
0x2040	(Top of frame for caller of foo)

Bar puts on stack (+10)

At this point Foo is about to call Bar

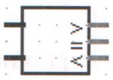
Put on stack by Foo (+20)

(+2) – order is correct and no extra elements.

2. Classify each of the following Logisim components as either **Sequential Logic** or **Combinational Logic**.

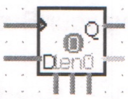
C → combinational
S → sequential

Comparator



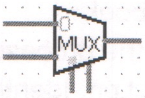
a) C

D Flip-Flop



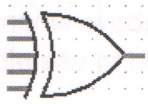
b) S

Multiplexer



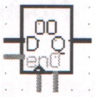
c) C

XOR Gate



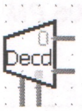
d) C

Register



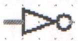
e) S

Decoder



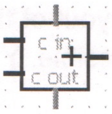
f) C

NOT Gate



g) C

Adder

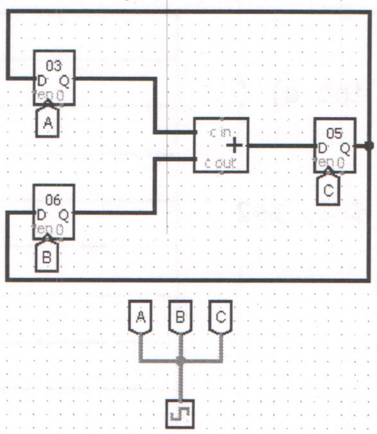


h) C

+ 3.5 for (b) and (e)

+ 3 for everything else

3. The following circuit contains three 8-bit registers A, B, and C, each initialized with 0x03, 0x06, and 0x05, respectively, as well as a clock and an adder. Fill in the table below to indicate the values for A, B, and C across 3 clock cycles.



Register	Clock Cycle 0	Clock Cycle 1	Clock Cycle 2	Clock Cycle 3
A	0x03	0x5	0x9	0xA
B	0x06	0x5	0x9	0xA
C	0x05	0x9	0xA	0x12

8 points per clock cycle

+2 A (should be C(n-1))

+2 B (should be C(n-1))

+4 C (should be A(n-1) + B(n-1))

+1 - for perfectly matching the solution.