**22C: 060 Spring 2008**

**Sample solution to Examination 1**

**Answer 1** $s_0$ contains 400, $s_2$ contains 0, and $M[400]$ contains the 32-bit integer 13.

\[
\begin{align*}
\text{addi } s_1, \text{ zero, } 32 \\
\text{lw } t_1, 0(s_0) \\
\text{lui } t_2, 32768 / 65536 = 2^{15}
\end{align*}
\]

\[
\begin{align*}
\text{L:} & \quad \text{and } t_3, t_1, t_2 \\
& \quad \text{sll } t_1, t_1, 1 \\
& \quad \text{subi } s_1, s_1, 1 \\
& \quad \text{beq } t_3, \text{ zero, LL} \\
& \quad \text{addi } s_2, s_2, 1 \\
\text{LL:} & \quad \text{beq } s_1, \text{ zero, L}
\end{align*}
\]

(End of program)

(a) When the program ends, $s_2 = 3$ (the number of 1’s in 13, $13 = 1101$)
(b) The program computes the number of 1’s in the 32-bit word stored in $M[400]$.

**Answer 2a.** Note that J L will not work since the offset will not fit into 28-bits. One solution is to use the indirect jump instruction.

\[
\begin{align*}
\text{lui } t_0, \quad 3FFF \text{ (hex)} \\
\text{addi } t_0, \quad t_0, \text{ FFFF (hex)} & \quad \text{$t_0$ now contains 30 1's} \\
\text{jr } t_0 & \quad \text{jump to the address stored in $t_0$}
\end{align*}
\]

Other solutions are possible too. (A solution that stores a 1 in $t_0$ and shifts it by 31 places to the left, before using jr $t_0$ is also correct, but painfully slow)

**Answer 2b.** Assume that the integer is in $s_0$. Use a mask to filter out the LSB. If the result is non-zero then the original number is odd

\[
\begin{align*}
\text{andi } t_0, s_0, 1 \\
\text{bne } t_0, \text{ zero, odd} & \quad \text{even: <the number is even>} \\
\text{even: <the number is even>} & \quad \text{j next} \\
\text{odd: <the number is odd>} & \quad \text{next: <done>}
\end{align*}
\]

**Answer 3a**

(a) $A + B + C = 0$ implies $ABC = 111, 100, 110$

**Answer 3b**

\[
\begin{align*}
F & = (A + B + C). (A + B \overline{C}) \\
& = A + (B+C). B \overline{C} \quad (*) \\
& = A + B \cdot \overline{C}
\end{align*}
\]

*(A+B). (A+C) = A+B.C (You can also solve this using a truth table)*
Answer 4.

Half-adder: Adds two bits (A, B), generates a sum and a carry.

Full adder: Adds three bits (including the carry Cin from the previous stage), generates a sum and a carry.

Answer 5.

Many completely messed up the connections with the inputs of the 1-to-2 decoder.

Grade summary (with approximate perception of the letter grades)