Homework 0 Solutions

1. (10 points) Design a digital logic circuit that implements the 4-input Boolean function described by the truth table below. You may use three inverters to complement the inputs and up to four other gates.

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$\overline{f(x_0, x_1, x_2, x_3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The Karnaugh map for this truth table is:

Which yields a boolean expression of $F = \overline{x_0} \cdot x_1 + \overline{x_0} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$ and the following combinational logic circuit.
2. (10 points) Consider the sequential logic circuit below. Assume that the flip-flops are triggered on the rising edge of the clock signal. Complete the timing diagram.

The behavioral table of a T (toggle) flip-flop is given below.

<table>
<thead>
<tr>
<th>T</th>
<th>Q_{t-}</th>
<th>Q_{t+}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Extra credit: Use the schematic capture tool in Xilinx ISE to construct the circuit shown and use the simulator to verify your answer.

3. (10 points) Convert the following C code to MIPS assembly instructions. Use the minimum number of instructions necessary. Assume that variables f, g and h are 32-bit integers stored in registers $t0$, $t1$ and $t2$ respectively and that the base address of arrays A and B are in registers $s3$ and $s4$ respectively.
(a) \[ f = g - h; \]
\[
\text{sub } \$t0, \$t1, \$t2
\]

(b) \[ f = g + (h - 5); \]
\[
\text{addi } \$t0, \$t2, -5
\]
\[
\text{add } \$t0, \$t1, \$t0
\]

*Note: other correct two instruction solutions are also acceptable.*

(c) \[ f = g - A[1]; \]
\[
\text{lw } \$t0, 4(\$s3)
\]
\[
\text{sub } \$t0, \$t1, \$t0
\]

*Note: other correct two instruction solutions are also acceptable.*

(d) \[ B[2] = A[f-h]; \]
\[
\text{sub } \$t0, \$t0, \$t2
\]
\[
\text{sll } \$t0, \$t0, 2
\]
\[
\text{add } \$t0, \$s3, \$t0
\]
\[
\text{lw } \$t0, \$t0, 0
\]
\[
\text{sw } \$t0, 8(\$s4)
\]

*Note: other correct solutions are also acceptable.*

4. (10 points) Consider the following 32-bit binary items.

\[
\begin{array}{cccccccc}
0000 & 0010 & 0011 & 0001 & 1000 & 1000 & 0010 & 0000 \\
0000 & 0001 & 0000 & 1010 & 0100 & 0000 & 0010 & 0010 \\
\end{array}
\]

(a) Assuming the items above are instructions, what MIPS assembly instructions do they represent?

\[
\begin{array}{cccccc}
\text{op} & \text{rs} & \text{rt} & \text{rd} & \text{shamt} & \text{func} \\
000000 & 10001 & 10001 & 10001 & 00000 & 100000 \\
000000 & 01000 & 01010 & 01000 & 00000 & 100010
\end{array}
\]

\[ \Rightarrow \text{add } \$s1, \$s1, \$s1 \]

\[ \Rightarrow \text{sub } \$t0, \$t0, \$t2 \]

(b) Assuming the items above are data (i.e. integers), what value do they represent?

36,800,544 and 17,448,994

(c) How can you tell if the bits represent instructions or data? There is no way to tell just by looking at the bits.
(d) Could the bit patterns represent anything else besides instructions and integers? If so, what? Floating point number. Signed or unsigned integer (though the value represented would be the same for both of these).

5. (5 points) Show the hexadecimal representation of the following MIPS instructions.

(a) addi $t0, $t0, 4
(b) sw $t0, 32($t2)

\[
\begin{array}{cccc}
\text{op} & \text{rs} & \text{rt} & \text{imm} \\
\hline
\text{addi $t0, $t0, 0} & 001000 & 01000 & 01000 & 0000000000000100 & \Rightarrow 0x2108 \ 0004 \\
\text{sw $t1, 32($t2)} & 101011 & 01010 & 01000 & 0000000000100000 & \Rightarrow 0xAD48 \ 0020 \\
\end{array}
\]

6. (10 points) Consider the following changes to the MIPS instruction set. Assuming no changes are made to the other instruction fields, draw the new R-type instruction format. Be sure to label each field and include its size.

(a) Change from 32 to 128 registers

Each register field would need to grow from 5-bits to 7-bits. R-type instructions have three register fields so the overall instruction length would need to grow by 6-bits giving a new instruction size of 38-bits.

\[
\begin{array}{cccccc}
\text{op} & \text{rs} & \text{rt} & \text{rd} & \text{shamt} & \text{func} \\
\hline
6\text{-bits} & 7\text{-bits} & 7\text{-bits} & 7\text{-bits} & 5\text{-bits} & 6\text{-bits} \\
\end{array}
\]

(b) Change to support four times as many instructions

*be sure to think about the opcode and the funct value!*

Both the opcode and function code fields should be extended from 6-bits to 8-bits. The overall instruction length would need to grow by 4-bits giving a new instruction size of 36-bits.

\[
\begin{array}{cccccc}
\text{op} & \text{rs} & \text{rt} & \text{rd} & \text{shamt} & \text{func} \\
\hline
8\text{-bits} & 5\text{-bits} & 5\text{-bits} & 5\text{-bits} & 5\text{-bits} & 8\text{-bits} \\
\end{array}
\]