Q1) Multiplexers:

Consider the function given by the following truth table:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Truth table

Implement that function using:

- An 8:1 multiplexer
- A 4:1 multiplexer and one inverter
- A 2:1 multiplexer, one NOR gate and one AND gate

You can use the ground and Vdd as inputs to the multiplexer. Signal fanouts are allowed.

Q2) Adders:

a) An incrementer adds 1 to an N-bit number. Build a 4-bit incrementer using only half adders. You can connect input ports of the half adder to Vdd or GND. The circuit you will compose will have 5 labeled inputs and 5 labeled outputs:

- 1-bit input M – Mode: when set to 0, S=A, and when set to 1 S=A+1.
- 4 bit input A[3:0] – Input of the incrementer
- 4-bit output S[3:0] – Output of the incrementer
- 1-bit output O – Overflow: 1 if incrementer overflows, 0 otherwise

Recall: half adders have two inputs (A,B) and two outputs (Co and S) given by:

- \( Co = A \cdot B \)
- \( S = A \oplus B \)
You can represent the half adder using the following representation:

\[
\begin{array}{ccc}
A & & S \\
& \text{HA} & \\
B & & \text{Co}
\end{array}
\]

Figure 1: Half adder block diagram

b) If an AND gate delay is 500ps and an XOR gate delay is 1.5ns. What is the delay of a half adder? Of the 4-bit incrementer?

c) Let’s build a 16-bit counter. In part a), you built the following 4-bit incrementer:

Using 4-bit incrementers, and pos-edge triggered flip-flops with active-high asynchronous reset, design a 16-bit counter that adds 1 at each positive clock edge. The counter has a clock input and an asynchronous reset input. The outputs should include the 16-bit counter value, and an overflow bit. Note: you can represent your 16 flip flops as one monolithic register which assumes all clocks and reset signals are wired together.

Change log:
EDIT 4/24/2013: in Q2 fixed the Co equation to AND. Also updated the gate delays.
EDIT 4/25/2013: in Q2 part c, labeled the input of the 4-bit incrementer correctly