Notes about ENEL 353 Fall 2017 Midterm Question 8
November 8, 2017, by Steve Norman

This problem, as presented on the midterm paper, cannot be solved. I apologize for putting such a problem on a test—it's unfair to set a question that is intended to be a challenge, but turns out to be an impossible challenge.

Marking. In consultation with Norm Bartley, I decided to reduce the weight of this question to 3 marks, and to change the maximum possible score on the test from 50 marks to 48 marks. The idea was to make the question count for less within the overall test score while still giving credit to students who made good efforts to solve the problem.

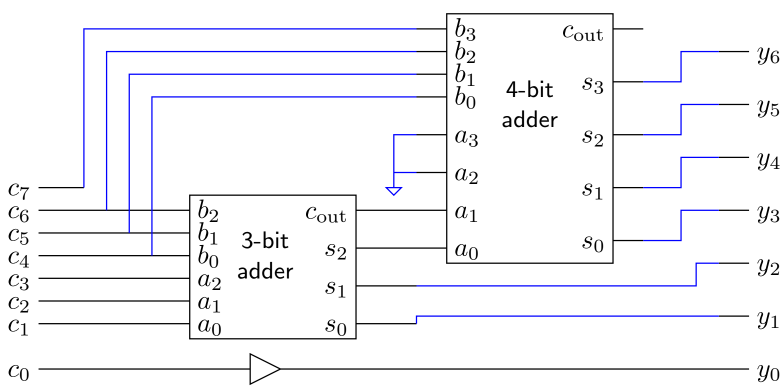
Marks up to a maximum of 2 were given for choosing inputs to the 4-bit adder, as follows:

2	choice would work for a large number of input values
1.5	inputs to 4-bit adder complete and reasonable
1	inputs to 4-bit adder somewhat incomplete or unreasonable
0.5	inputs to 4-bit adder mostly incomplete or unreasonable

Marks up to a maximum of 1 were given for connecting wires to the outputs $y_6 \ldots y_0$, as follows:

1	output choices complete and reasonable
0.5	output choices somewhat incomplete or unreasonable

Here is an answer that some students found, that got 3 marks:



It generates correct results for all two-digit BCD inputs with $c_7 = 0$, but not for two-digit BCD inputs with $c_7 = 1$. In other words, it works when the input bits represent a number between 0 and 79_{10}, but not when they represent a number between 80_{10} and 99_{10}.

Problem background. The idea behind the problem was to find out if students could recognize a few things:

- the circuit would need to do binary arithmetic to compute
  $10 \times c_7c_6c_5c_4 + c_3c_2c_1c_0$;
- that arithmetic result could be computed as
  $(c_3c_2c_1c_0 + 2 \times c_7c_6c_5c_4) + 8 \times c_7c_6c_5c_4$;
- using multi-bit adders to solve the problem was similar in some ways similar to the 4-bit $\times$ 4-bit multiplication problem of Lab 2.

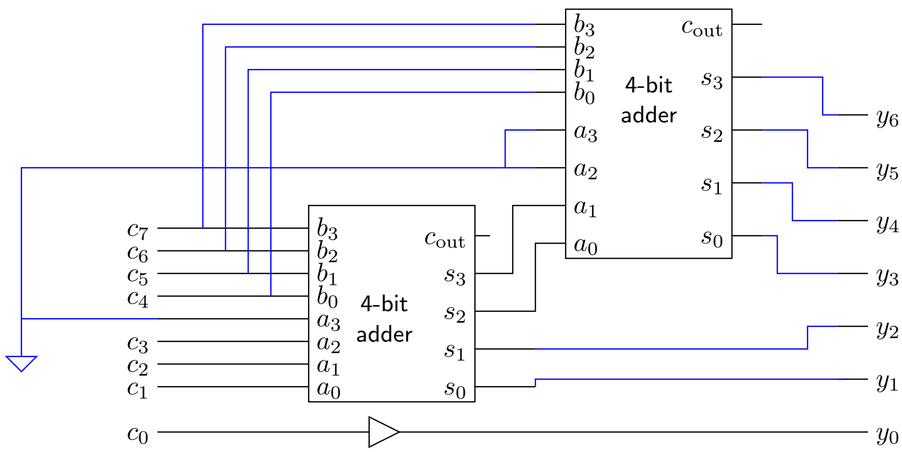
I realize that those are a lot of puzzle pieces to put together under time pressure while writing a test, but the goal was to provide a challenge at the end of a test that was made mostly of questions that were fairly recognizable in form from problem sets and/or old midterms.

Here's the layout of the computation:

$$\begin{array}{rccccccccl} & 0 & 0 & 0 & c_3 & c_2 & c_1 & c_0 & \\ + & 0 & 0 & c_7 & c_6 & c_5 & c_4 & 0 & (2 \times c_7c_6c_5c_4) \\ + & c_7 & c_6 & c_5 & c_4 & 0 & 0 & 0 & (8 \times c_7c_6c_5c_4) \\ \hline & y_6 & y_5 & y_4 & y_3 & y_2 & y_1 & y_0 & \end{array}$$

Bits of the sum of the first two rows can be found with a multi-bit adder. A second multi-bit adder could take bits from that first sum and add them to bits of the third row to generate $y_6y_5y_4y_3$. The flaw in the problem design was incorrectly believing that that the first adder could be only 3 bits wide.

Here is a circuit with two 4-bit adders that does the job correctly for all inputs between 0 and 99_{10}:



A problem based on the above circuit, along with some hints, would have made a better midterm problem, but there's no way to back in time and fix that now!