CmpE102
Spring 2017
Programming Assignment 6

Due Date: 19 April
Goal: Linear feedback shift register

Instructions:

Step 1. Creating the assembly language file

Everything should be done the same way as the previous assignment.

Step 2. The application area

The wikipedia article is a bit math-heavy, but gives a good description of the many application areas.
https://en.wikipedia.org/wiki/Linear-feedback_shift_register

Here is a more hardware-oriented description.
http://www.eetimes.com/document.asp?doc_id=1274550

Here is a summary explanation that does not involve a hardware description. You are familiar with the shift left and shift right instructions. These move all the bits in the register to the left or right respectively by a specified number of bit positions. Let's suppose that we shift by one position. Then a new bit value will enter at the least or most significant end of the register respectively. The value of the new bit would be zero if shifting to the left. Shifting to the right, the new bit is a zero for a logical shift, and a duplicate of the sign bit for an arithmetic shift.

A linear feedback shift register (LFSR) is a shift register whose input bit is
a linear function of its previous state.

The only linear function of single bits is XOR, thus it is a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value.

In a LFSR, the new bit is neither of those, but some arbitrary function of the existing bits in a register. Depending on the function, as the shifts repeat, the successive values in the register will go through a repeating sequence. It can be shown that the maximum length of the sequence for an n-bit register is 2^n-1 if you chose the best function. Other functions may give shorter sequences.

Step 3. What your code should do

1. We will use an 8-bit register so the output is limited to 255 values.
2. There are two type of LFSRs, with external and internal feedback. We will use the external feedback kind.
3. We will shift to the right by one position on each step.
4. The new bit entering at the left (new bit b_7) is a function of the old bits (before the shift occurs).
5. The function for the new bit is XOR(b_4,XOR(b_3,XOR(b_0,b_2))).
6. You might as well use a register to hold your value.
7. Initialize the register to 1. (If it is zero, the register contents will never change.)
8. One more time--remember that the new bit value must be calculated before the shift takes place, and then placed in bit position b_7 after the shift takes place.
9. Print out the sequence in both binary and hex using **`printf()`**.
10. **`printf()`** does not support a binary output format. You will have to write a function to do this. (See http://stackoverflow.com/questions/6373093/how-to-print-binary-number-via-printf)
11. Print out one time through the sequence.

Hints

The following skills are exercised in this assignment:

- Bit manipulation
- Function call
- **`printf()`**

Step 4. Turning in the assignment

Turn in the following:

1. Commented listing file.
2. Output of one time through the sequence.