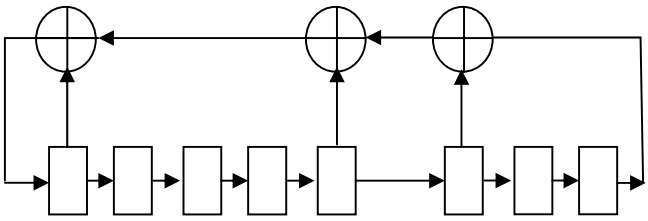
**COE 205, Term 052**

**Computer Organization & Assembly Programming**
**Programming Assignment# 3**

Due date: Saturday, May 6, 2006

You are required to write an 8086 assembly program to implement a pseudo random generator using Liner Feedback Shift Register (LFSR). An example of an 8-bit LFSR is shown below:



Two important characteristics of an LFSR are the Feedback Polynomial, which determines the FFs that are XORed to compute the shifted bit, and the seed which determines the initial content of the FFs. Depending on the Feedback polynomial, the LFSR can generate a maximal-length sequence without repetition, or it may not. The seed can be any number other than 0.
The 8-bit LFSR shown above is a maximal-length i.e. it is guaranteed to generate a random sequence in the range from 1 to 255 before it repeats again.
The Feedback polynomial for the above LFSR can be represented as 10001101. Note that 1 indicates that there is feedback connection, while 0 indicates that there is no feedback connection.

(i) Write a macro, RAND8, that implements an 8-bit pseudo random generator. The macro should be given the Feedback polynomial, and the seed, and it should generate the next random number. Assume that the Feedback polynomial, the seed and returned random number should be passes as macro parameters.

(ii) Write a procedure that gets as parameters the address of an Array and the size of the Array and scans the array content and determines after how many elements the first number stored in the Array is repeated. For example, suppose that the Array content is 1, 5, 6, 1, 4, 2, then the procedure should return 3. You can store the returned number as you like.

(iii) Using the macro and procedure in (i) and (ii), generate the first 256 random numbers that will be obtained from the feedabck polynomial 10001101 and an initial seed of 00000001. Determine after how many numbers the first number is repeated.

(iv) Using the macro and procedure in (i) and (ii), generate the first 256 random numbers that will be obtained from the feedabck polynomial 10001101 and an initial seed of 1010101. Determine after how many numbers the first number is repeated.

(v) Using the procedure and macro in (i) and (ii), generate the first 256 random numbers that will be obtained from the feedabck polynomial 10000001 and an initial seed of 00000001. Determine after how many numbers the first number is repeated.

(vi) Ask the user to enter a feedback ploynomial and a seed. Then, ask the user to enter a string of characters. Then, encrypt the string using RAND8 as follows. Each character is encrypted by XORing the least significant 4-bits of the ASCII code of the character with the lest significant 4 bits and the most significant 4 bits of the generated random number. For example, assume the character to be encrypted is A=41H and the random number is A1H. Then, the encrypted character will be obtained by keeping the most significant digit as is and computing the least significant digit as A XOR1 XOR 1= A. Thus, the encrypted charactaer will have the ASCII code 4AH = J. To decrypt the character, the decrypted character 4AH=character J, will be XORed with the same corresponding random number used for encryption i.e. A1 and this will generate the original character 41H= character A, as A XOR 1 XOR A=1. As an example show the encryption of the string **This is the last Assignment!!**. Then, rerun your program giving it the encrypted string and it should correctly decrypt it to **This is the last Assignment!!**. Try this with the feedback polynomial 10001101 and a seed of 10101010.

***The solution should be well organized and your program should be well documented. Submit a soft copy of your solution in a zip file. The soft copy should include a Readme file indicating the file names containing the solution and whether it works or not. The Readme file should also contain your name and ID. Submit both source code file (i.e. .asm) and the executable file (i.e. .exe).***