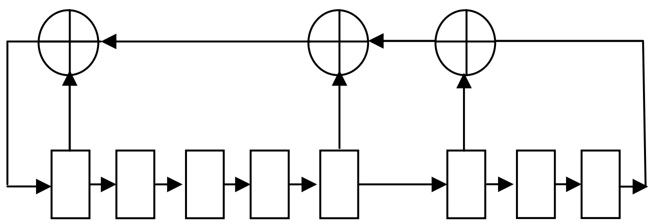
COE 205, Term 051

Computer Organization & Assembly Programming
Programming Assignment# 4

Due date: Wednesday, Dec. 14, 2005

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 **READB** to read a binary number and store it in the specified parameter. The macro should report an error message and ask the user to reenter the number if the value read is 0. Also, the user does not have to enter the whole 8-bits. If he enters less than 8 bits and hits return the remaining most significant bits should be assumed 0. Also, if any digit entered is other than a binary digit, an error message should be reported.

(ii) Write a procedure **DISPD** to display the content of an 8-bit number in decimal. The number to be displayed must be passed using the stack

(iii) Write a macro, **RAND8**, that implements an 8-bit pseudo random generator. The macro should be given the Feedback polynomial, and the seed as parameters and it should generate the next random number. Assume also that the generated number will be returned on a specified parameter.

(iv) Ask the user to enter an 8-bit feedback ploynomial and an 8-bit seed in binary. Use the macro READB for this purbose. 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 least 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 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 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!!**. Encrypt this with the feedback polynomial 10001101 and a seed of 0001111.

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).