

EEAP 282
PROGRAMMING ASSIGNMENT #3

CRC SUBROUTINE

PURPOSE: This lab will introduce you to the use of subroutine and parameter passing using the stack. You will write a subroutine which performs CRC calculation.

LAB DESCRIPTION:

You will write a subroutine CRC which performs CRC calculation. The subroutine will receive three parameters on the stack, a 16 bit CRC mask, 16 bit number representing the number of bytes in the data block to be checked, 32 bit address of the starting block of the data to be checked.

We used the CCITT mask for calculating the CRC in Lab #2. Another popular choice for the mask is Xmodem (one of the old and slow, but popular data transfer protocol masks). The mask for the Xmodem CRC is \$A001, or %1010000000000001.

Your subroutine should return the CRC (16 bit) to the calling program on the stack. Your main program should set up the stack and push the parameters, and your subroutine should access and manipulate the stack. Your main program should retrieve the calculated CRC from the stack. Your subroutine should not destroy the contents of any registers when returning to the calling program.

In the lab report, include diagrams of the stack contents at start, just before calling CRC, after first few instructions of CRC, and after returning from CRC.

Test your CRC subroutine on the 3 different data sets given in Lab #2 using the two different CRC masks (CCITT and Xmodem).

Your lab report should use the following format:

1. signed title page with short lab abstract
2. assembler listing of your program - not all data needs to be included in the program listing
3. What is on the stack immediately before you call your subroutine, when you execute the first instruction in your subroutine, and immediately after you return to the main program from the CRC subroutine. A drawing is expected for each case.
4. Explain how you would modify your code to pass parameters using in-line coding of data.
5. Compare subroutine parameter passing using both data and address registers, in-line coding of parameters, and passing parameters on the stack. You should address issues of code size, execution speed, and memory usage.