# ECE 331 – Hardware organization and Design

## Department of Electrical and Computer Engineering

## University of Massachusetts, Amherst

## Fall 2023

## Project 1

The goal of this project is to design a simple encryption/decryption system. The "Encryption" part of the system works as follows:

a) It works on text documents only. All characters are coded using ASCII codes. The input text is referred to as the "Plain Text".
b) It uses a 4-character (32 bits) password (also referred to as a "Secret Key").
c) The encryption algorithm works as follows:
    I. Four characters are packed in a 32-bit register.
    II. The password is also stored in a 32-bit register.
    III. To process each 4-character package, each character is considered as two 4-bit unsigned binary numbers and if both numbers are even or both numbers are odd, the corresponding password character is also divided into 4-bit numbers and these 4-bit numbers in password are swapped. Then bits 0 and 4 of all four generated password characters are cleared to 0. Notice that the original password will not change, just the current copy being used for these four characters.
    IV. After stage iii is finished for all four characters of the data package, the data package and the produced password (each being 32 bits) are XOR'ed and the results are stored in memory.
    V. The text generated in step IV is referred to as "Cipher Text". It generally should be very different from the original "Plain Text" and convey no useful information unless it is "decrypted" by another system that undo's the encryption process.

d) Here is what you need to do for this project:
    I. Use your firstname_LASTNAME in a sentence such as: "john_DOE is a senior CompE student at UMASS, Amherst." And use it as the "Plain Text" followed by a byte of all 0's (this byte is called a NULL in some high-level programming languages) and define it in your assembly program as an array of ASCII characters (first name in lower case and last name in upper case separated by underscore "_").
    II. Use the first four characters of your family name (in lower case) as the password: ("doe0" in the above example). If your family name is less than 4 characters pad with zeros.
    III. Write a "main" program that after initialization of the variables, has three tasks only:
        1. The first task is to call a procedure called ENCRYPT with six parameters: the starting address of the "Plain Text", the starting address of the "Cipher Text" (where the generated "Cipher Text" should be stored in memory) and the four characters used as

the password. This procedure should convert the “Plain Text” to the “Cipher Text” using the steps outlined in part (c) above.

2. The second task is to call a procedure called DECRYPT with six parameters: the starting address of the generated “Plain Text” (different from the original “Plain Text” in memory), the starting address of the “Cipher Text” and the four characters used as the password. This procedure should convert the “Cipher Text” to the “Plain Text” using the steps that you need to propose. This is not hard if you understand the encryption steps well.
3. The third task is to call a procedure called “COMPARE” which compares the original “Plain Text” with the generated “Plain Text” by the ENCRYPT procedure and returns a “1” if they are exactly the same or a “0” if they are not.

   **NOTE**: In the three procedures “ENCRYPT”, “DECRYPT”, and “COMPARE”, try to break down the required tasks into smaller tasks and implement each of those smaller tasks using a procedure. Try to maximize the sharing of the procedures among the “ENCRYPT”, “DECRYPT”, and “COMPARE” procedures.

e) Use a RISC-V assemble and simulator (you may use BRISC-V Simulator, which is a free online assembler-simulator, or any other assembler-simulator of your choice for this part) to test your program by showing the original “Plain Text”, the generated “Cipher Text”, and the generated “Plain Text” produced by the “DECRYPT” procedure and comparison results of the original “Plain Text” and the generated “Plain Text” after decryption.

f) Deliverables: Please submit a report with the followings:
   I. The decryption algorithm (similar to part (c) above).
   II. The RISC-V source code for all the procedures (including the main program).
   III. Screen shots of the simulator run showing the original “Plain Text”, generated “Cipher Text”, and the generated “Plain Text.”