EE 3144
INTRODUCTION TO EMBEDDED SYSTEMS
LAB 4

TA: Issa Fattah (ifatta01@utopia.poly.edu),
Bernard Goldinger (bgold01@gmail.com)

The objective of this lab is to implement the 32-bit RC-5 encryption algorithm.

Design Flow:

The design flow to be used in this lab will remain similar to what was used for Lab 1

1. Type the code in the .asm file using an text editor (eg:notepad)
2. Assemble the code using ASM51
3. Simulate the code using JSIM
4. Simulate the code using Simulator2003
5. Download the code on to the board

Downloading the program to the Board:

The board which we will be using for this lab is the XS95+. The datasheet and other details can be seen at the manual posted on the Mypoly site.

To download the code to the board:

- ✓ Double click on the XSLOAD program on the desktop
- ✓ Double click on the directory you created which has the ASM, LST and the HEX files.
- ✓ The TA will provide you with another "svf" file to program the CPLD on the board, for proper functioning of the board.
- ✓ First, drag and drop the ".svf" file in to the XSLOAD window
- ✓ After receiving a successful programming message, drag and drop the .HEX file in the directory (with appropriate modifications, as explained by the TA, for 7-segment display purposes)
- ✓ The board is now programmed with your code

RC5 algorithm:

Here is a recap of the RC5 encryption algorithm.

```
A = A + S[0]
B = B + S[1]
for i = 1 to r do
A=   ((A xor B)<<< B)+   S[2* i]
B=   ((B xor A)<<< A)+   S[2* i+1]
```

where A and B are the two halves of the input plaintext and S[0], S[1], S[2]…… are the round keys.

Unlike in lab 2, we have to now implement the 32-bit version of the RC5 algorithm. This implies that all the quantities we deal with now are 32-bits in length. We now have 26 32-bit round keys. The additions and rotations are to be performed on 32-bit values now.

For simplicity, we will not implement the key-scheduling in this lab. We assume that the key scheduling has already been performed and the 26 round-keys are available:

S[0]=9bbbd8c8
S[1]=1a37f7fb
S[2]=46f8e8c5
S[3]=460c6085
S[4]=70f83b8a
S[5]=284b8303
S[6]=513e1454
S[7]=f621ed22
S[8]=3125065d
S[9]=11a83a5d
S[10]=d427686b
S[11]=713ad82d
S[12]=4b792f99
S[13]=2799a4dd
S[14]=a7901c49
S[15]=dede871a
S[16]=36c03196
S[17]=a7efc249
S[18]=61a78bb8
S[19]=3b0a1d2b
S[20]=4dbfca76
S[21]=ae162167
S[22]=30d76b0a
S[23]=43192304
S[24]=f6cc1431

S[25]=65046380
We now observe that the internal memory may not be enough to store all the required keys and to perform the various operations. So now, we need to make use of the external memory.

The first task is to transfer each of these keys in to the external memory. Transfer of data to and from the external memory has to be done using the MOVX instruction and the DPTR has to be used. Care has to be taken regarding selecting the order of the keys to be stored, as well as the format in which they are stored (little endian or big endian).

Once the keys are stored, the encryption can begin. 32-bit data A and B can be stored in consecutive locations in the internal memory. They can be brought in to the register banks/ accumulator to perform operations on them. The first operation to be performed is the 32-bit addition operation. This can be accomplished either by manually performing the 32-bit addition or simply calling the program written in lab 1 as a subroutine. The rotate function is performed by calling the program written in lab 3 as a subroutine. It also has to be taken in to account that the DPTR can only be incremented but not decremented.

Exercise 1:

Implement the 32-bit RC5 encryption algorithm making use of the previously written programs. Report the code size and the number of clock cycles used to peform the encryption.

Use the following test vectors:
For Plaintext A= 0x00000000 B=0x00000000 using the above 26 round keys provided, the results should be Ciphertext A=EEDB8A521 and B= 0x6D8F4D15

Answer the following in the Lab Report:

Question 1: How many clock cycles did the code take to execute? What is the size of the code in bytes?

General Lab Report Guidelines:

- Lab reports are due at the beginning of the lab session.
- Electronic version of both the lab reports and the code should be e-mailed to the TA at njoshi01@utopia.poly.edu
- Late submission will result in a 0 point grade for that lab, unless prior permission is obtained from the Professor/TA
- Apart from answers to the above questions asked, lab reports must address the following issues
  - New concepts learnt in the lab session
  - Difficulties faced
  - Possible solutions to the problems faced
  - The number of clock cycles required
  - The program size
  - Further possible improvements (if applicable)
- Apart from the above criteria, points will be based on the efficiency of the code (area/clock cycles). The highest grade will be given to the most efficient design.
- Identical codes/lab reports will get 0 points and are liable for serious penalties.