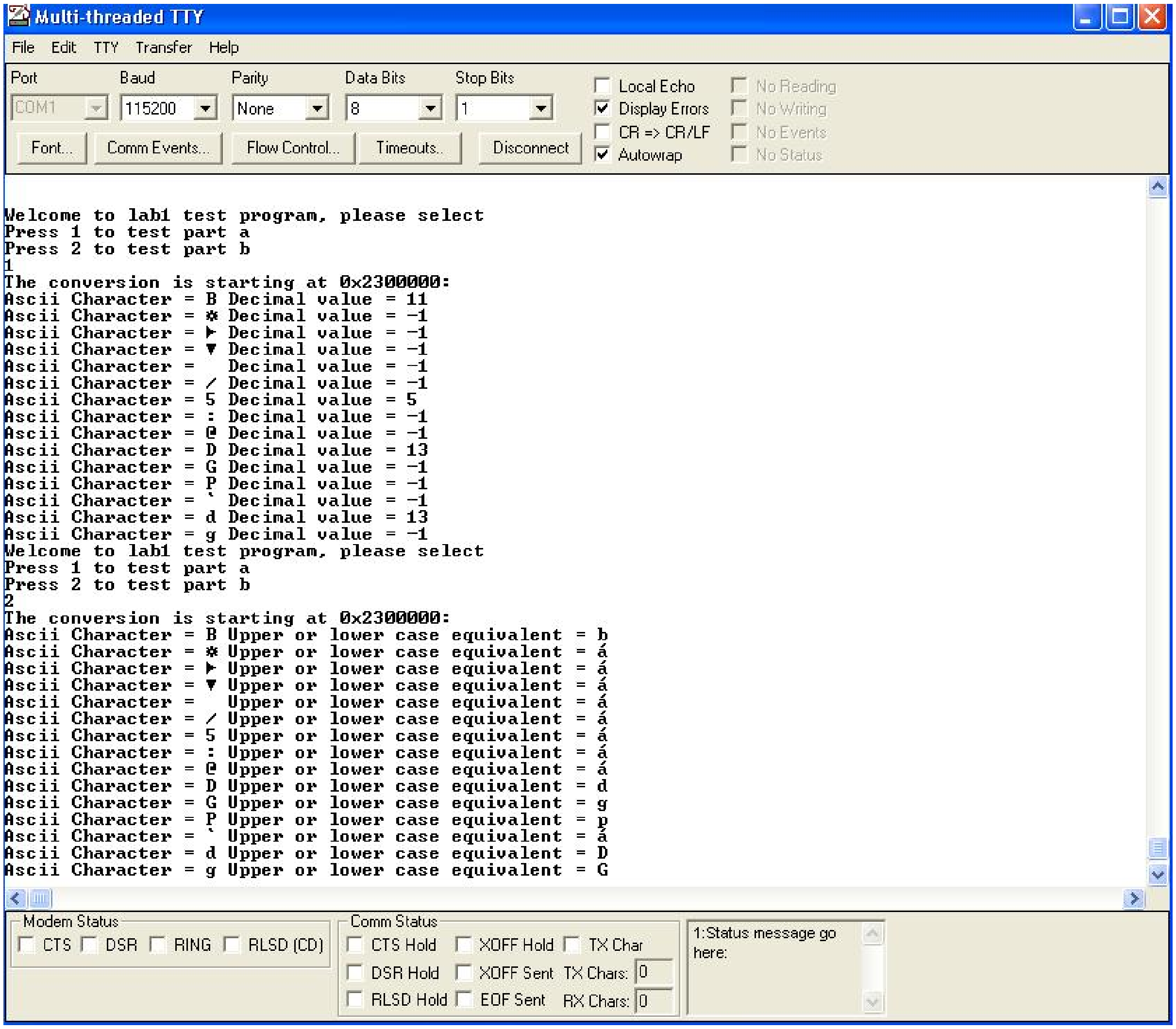
Lab 1: Introduction to Assembly language



Port: COM1 | Baud: 115200 | Parity: None | Data Bits: 8 | Stop Bits: 1

□ Local Echo
☑ Display Errors
□ CR => CR/LF
☑ Autowrap
□ No Reading
□ No Writing
□ No Events
□ No Status

```
Welcome to lab1 test program, please select
Press 1 to test part a
Press 2 to test part b
1
The conversion is starting at 0x2300000:
Ascii Character = B Decimal value = 11
Ascii Character = ✱ Decimal value = -1
Ascii Character = ► Decimal value = -1
Ascii Character = ▼ Decimal value = -1
Ascii Character =   Decimal value = -1
Ascii Character = / Decimal value = -1
Ascii Character = 5 Decimal value = 5
Ascii Character = : Decimal value = -1
Ascii Character = @ Decimal value = -1
Ascii Character = D Decimal value = 13
Ascii Character = G Decimal value = -1
Ascii Character = P Decimal value = -1
Ascii Character = ` Decimal value = -1
Ascii Character = d Decimal value = 13
Ascii Character = g Decimal value = -1
Welcome to lab1 test program, please select
Press 1 to test part a
Press 2 to test part b
2
The conversion is starting at 0x2300000:
Ascii Character = B Upper or lower case equivalent = b
Ascii Character = ✱ Upper or lower case equivalent = á
Ascii Character = ► Upper or lower case equivalent = á
Ascii Character = ▼ Upper or lower case equivalent = á
Ascii Character =   Upper or lower case equivalent = á
Ascii Character = / Upper or lower case equivalent = á
Ascii Character = 5 Upper or lower case equivalent = á
Ascii Character = : Upper or lower case equivalent = á
Ascii Character = @ Upper or lower case equivalent = á
Ascii Character = D Upper or lower case equivalent = d
Ascii Character = G Upper or lower case equivalent = g
Ascii Character = P Upper or lower case equivalent = p
Ascii Character = ` Upper or lower case equivalent = á
Ascii Character = d Upper or lower case equivalent = D
Ascii Character = g Upper or lower case equivalent = G
```

Modem Status: □ CTS □ DSR □ RING □ RLSD (CD)

Comm Status: □ CTS Hold □ XOFF Hold □ TX Char □ DSR Hold □ XOFF Sent TX Chars: 0 □ RLSD Hold □ EOF Sent RX Chars: 0

1:Status message go here:

Lab Dates

Refer to the ECE212 Moodle website for the latest schedule

Introduction

In this first lab the students will be introduced to some of the basic instruction set in assembly language programming.

Objectives:

1. To gain experience developing code in assembly language
2. To gain experience working with coldfire based microcontroller board manufactured by Netburner
3. To gain experience working with IDE eclipse development environment

Prelab and Preparation:

- Read the lab prior to coming to your lab section.
- Do the online prelab Quiz
- Provide flow chart for each part of the lab.

Lab Work and Specifications

1. Download the template files
   - main.cpp
   - DataStorage.s
   - SetZeros.s
   - lab1a.s (For part A)
   - lab1b.s (For part B)
2. Download and print off the marking sheet

If you recall from the tutorial, the 'main.cpp' is the standard project template that is used to initialize and call standard functions/subroutines including our 'AssemblyProgram'. This will always be provided and you need not worry too much about it. Do not modify any of the parameters in this file. 'SetZeros.s' and 'DataStorage' are also provided to initialize the memory contents. Do not modify any of the parameters in either file. 'lab1a.s' and 'lab1b.s' are the two files provided for you to write your assembly language program. There are no passed parameters and the solution is expected to reside in these two files. A more detail description is provided below.

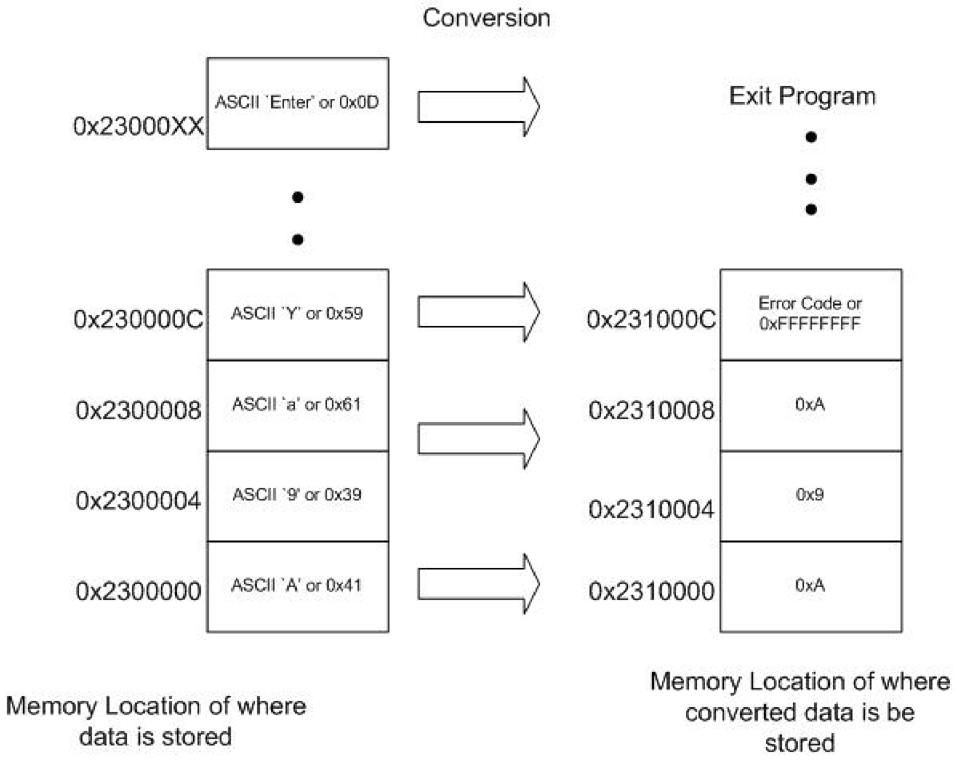
Specifications

Part A

Use the lab1a.s template for your coding

You are asked to write a program that will convert an Ascii character to its hex equivalent. Valid characters are 0 to 9, A-F and a-f. For example, the Ascii character ‘A’ when stored in memory will be represented by the hex value 0x41 or one byte (8bits) of data. Each memory location contains one byte of information. In other words, one memory location can store one Ascii character. However, for our lab, we are going to use 4 memory locations to store one Ascii character. A diagram is provided below to give you a visual interpretation. You are asked to convert this value to the hex value 0xA and store it in another location made up of 4 memory locations. Invalid Ascii characters should be rejected with an error code in the memory location. The error code has been chosen to be in hex ‘FFFFFFFF’. This error code is 4 bytes(32bit) long. If the memory location contains the Ascii ‘Enter’ code, your program should exit signaling the end and that no more conversions are required.

1. 0x2300000 - Start Address of where the data is stored
2. 0x2310000 - Start Address of where the converted values are to be stored



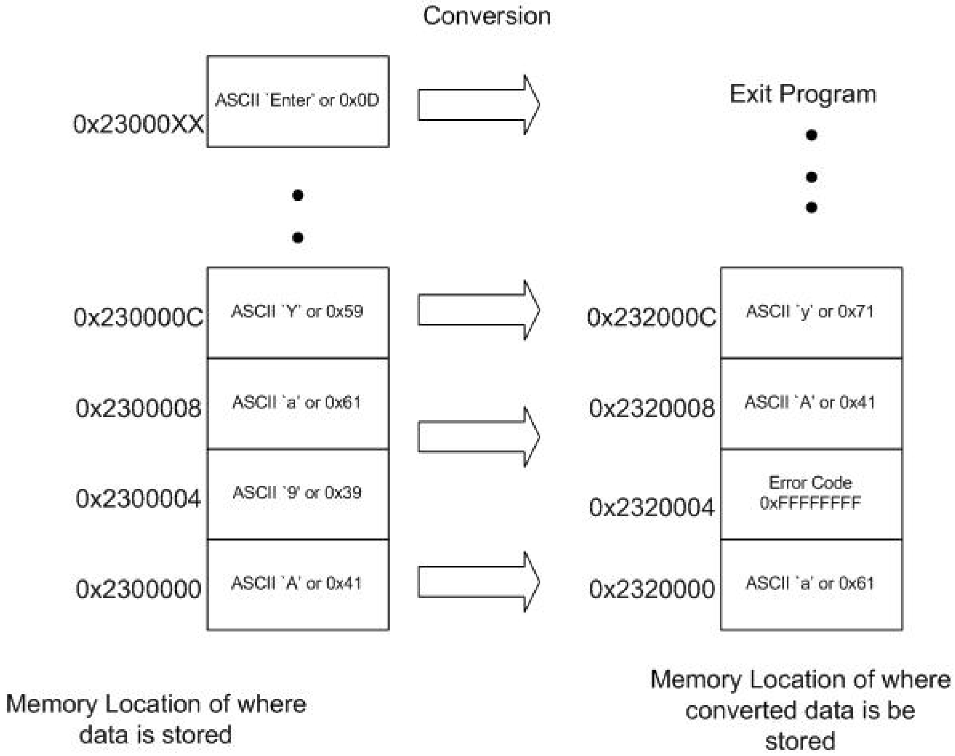
NOTE: Once you have completed part A, make sure you back up your lab1a.s by saving a copy of it somewhere. The best way is to back it up to your ECE home directory. If you proceed to part b without doing so, you may accidently delete your lab1a.s file and lose your solution.

Part B

Replace the lab1a.s template with lab1b.s for your coding

You are asked to write a program that will convert an Ascii letter to its upper or lower case equivalent. Valid characters are all upper or lower case letters. For example, The Ascii character 'A' when stored in memory will be represented by the hex value 0x41. You are asked to convert this value to its lower case equivalent 'a' which will be represented by the hex value 0x61 This value is then stored it in a memory location. Invalid Ascii characters should be rejected with an error code in the memory location. If the memory location contains the Ascii 'Enter' code, your program should exit signaling the end and that no more conversions are required.

1. 0x2300000 - Start Address of where the data is stored
2. 0x2320000 - Start Address of where the converted values are to be stored



Questions

1. What happens when there is no exit code '0x0D' provided in the initialization process? Would it cause a problem? Why or Why not?
2. How can your code be modified to provide a variable address range? For example, what if I only wanted to convert the first 10 data entries? You do not need to make these changes to your code. Just a detail description is sufficient

Marking Scheme

Lab 1 is worth 20 % of the final lab mark. Please view the Marking Sheet for this lab to ensure that you have completed all of the requirements of the lab. The Marking Sheet also provides a limited test suite in the demo section for you to think about. Make use of it! The report is to follow the Report Writing Guidelines

Demo and Report

The **report and demo due dates** are given on the ECE212 Laboratories page on Moodle. Note that you have one week from the dating of your prelab to complete the demo.

The reports must be handed in by **4 P.M. Do not be late**.

Late Submissions

Late submission penalties for the demo and report portions of the labs are given on the ECE212 Moodle website