

Laboratory 1

Programming in Assembly with CodeWarrior

Due Date: Beginning of Week 3 Lab Period

Files Needed:

- Week1Lab.zip from Blackboard

Introduction:

This lab addresses several topics. First and foremost is an introduction to programming for the S12 using CodeWarrior. We will use CodeWarrior to download machine code, monitor program execution, and inspect registers and memory. This week, we will only write code in assembly, and we'll add C programming in the next lab.

Assignment:

Open the project Week1Lab.mcp. Under the "Sources" folder in the file tree, there are two files. One is subs.asm, which is where your assembly code implementing the subroutine described below will be added. Main.asm contains the main program that calls your subroutine with 8 different inputs. Note that the expected output for each table is listed in the comments.

The subroutine will examine a list of 64 bits. The bits are grouped into bytes and uniquely numbered. The first byte has bit 0 (low or rightmost bit) to bit 7 (high or leftmost bit), the second byte has bit 8 (low or rightmost bit) to bit 15 (high or leftmost bit), and so on. The subroutine should return the number of the lowest bit which is set to 1.

A little context: Notice that in subs.asm, the name given to the subroutine is "priority". This list of 64 bits represents a set of flags corresponding to 64 different chunks of code. Each flag bit has a different number, and the number relates to the code's importance with lower values being more important than high values. Any piece of code that is ready to execute sets its corresponding flag to 1, while pieces of code that don't need to run have flags that are 0 (we'll see how that happens later). With the assumption that a system should always run the most important piece of code that is ready, your subroutine can be used to determine which piece of code that is.

Your subroutine must meet the following requirements:

1. The subroutine must accept one input, a 64-bit list, which is passed by reference (i.e. the address of the first eight bits) in register D.
2. The subroutine must return a one-byte unsigned integer in register B that is the value of the lowest bit set to 1 (assume there is always at least one bit set to 1).
3. Register values do not need to be preserved by the subroutine.

The subroutine should be written to minimize the bytes of code.

Deliverables

- Your CodeWarrior project in a zip file.
- Size of your subroutine in bytes.
- Running time of your subroutine in clock cycles for each of the 8 tables. This should be included in a text file attached to your project.