COMP9032 Lab 1

July 2018

1. Objectives

In this lab, you wi...

- AVR asse...
- basic asse...

2. Signing in

You will be working with your lab partner for all lab exercises. When you arrive at the laboratory for the first time, please sign in with your partner.

3. Examining the Host Environment in the Lab

The host environment is a standard personal computer running the Windows operating system. Take a note of the desktop icons. The icons include AVR Studio, which you will be using to edit your program, assemble it into the executable file and debug it. Another icon is the AVRDOC folder, which contains AVR Instruction Set Manual, AVR Instruction Set Summary and ATmega Reference Manual. Those documents are also available on the course website. You are encouraged to use these resources for your lab exercises.

4. Save Your Work

The computers in the lab are also used by other students. So when you finish your lab, please save all your work to your USB memory and DELETE all files associated with your lab if you do not want other students to use your files.

5. Programming Style

The general practice, when you write an assembly program, is to maintain the readability and consistency of your code. For this reason, you are encouraged to adopt the following rules, especially for this course:

- Starting each source code file with a heading that includes:
  - your name so that it is easy to see who is responsible for the file,
  - the date of last modification and a version number, and
  - **the description of what the program does, possibly with a pseudo-code for a high level abstraction.**

- Including appropriate comments that explain the "why", not just the "how" of the program throughout the source code.
- Using a sensible layout for your code -- to make it easy to see the code structures, instructions and any labels.

6. Tasks

6.1 Task 1 (for Week 2, due in Week 3)

Manually convert the program described by gcd.c below (i.e. calculating the greatest common divisor of a and b) into AVR assembly code (*gcd.asm)*. Assume the size of an integer is 1 byte. You need to produce two designs: 1) for minimum code size and 2) for minimum execution time.

```
int main(void)
{
        int a, b;                    /* Initialized elsewhere */

        while (a!=b)
        {                            /* Assume a, b > 0 */
                if (a>b)
                        a = a - b;
                else
                        b = b - a;
        }
        return 0;                    /* a and b both hold the result */
}
```

Figure 1: Program gcd.c

Assign register r16 to be the C code variable a, and r17 the variable b. **Note, do not initialize these registers within your code.** (You should be able to initialize these values in AVR Studio, refer to lab0).

Assemble and run your program in AVR Studio and show your working program to the lab tutor. Remember to save and delete your work before you leave the laboratory.

6.2 Task 2 (for Week 3, due in Week 4)

Write an AVR assembly program that performs the following function

$$sum = \sum_{i=1}^{n} a^i ,$$

where n and a are 8-bit unsigned integers and sum is 16 bits. *Try to minimize the number of registers used.*

Use macro to increase the readability of your program.

Assemble and run your program in AVR Studio and show your working program to the lab tutor.

NOTE:

Each task is wo[...]our programs should <u>be well commented and easy to read</u>. U[...]deducted for each program without proper and sufficient comm[...]

Please have your work marked in your lab class of the due week.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com