Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Prof. Archana Chidanandan

Peer evaluation of the Instruction Set Architecture and the associated documentation for a "miniscule" general purpose processor

Due date: 6th February 2004 for Sections 1 and 2
Turn-ins: 1. Assembly language program.
2. Machine language equivalent for procedure.
3. Evaluation.

You are provided with the Instruction Set Architecture(ISA) specifications for a "miniscule" general purpose processor. You are required to use these specifications to write an assembly language program that can run on this architecture. You will also convert a portion of this program to machine language. In the process of writing the program and it's machine language equivalent, you will also evaluate the documentation and the ISA provided to you. The criteria for evaluation is being provided to you.

Requirements that the ISA satisfies:

a. Support a minimum of two interrupts.
b. Accept at least one 4-bit input value.
c. Display a 16-bit value to an output device.
d. Support a variety of arithmetic and logical operations, data movement operations and flow control operations.
e. Support procedures.

Part 1:
Write a documented assembly language program to multiply an unsigned 8-bit number with the unsigned value 255(0xFF). The resulting output will be a value no larger than 16 bits. The value to be multiplied is fed as input to the processor through the use of an interrupt. As seen in the RelativePrimes program, the 8-bit input is fed in 4-bits at a time. The input value must also be displayed through the output device. Let us assume that the users are obedient or at least intelligent, and will not enter an input value larger than 8-bits. The second interrupt will cause the program to calculate the required product and display the result on the screen. If the output value is determined to be too large to be represented by a 16-bit value, then the value displayed must be "0x0000". The Java code fragment for this program is provided below. The applet demonstrating this program is also available on your class website, under Projects. To determine the product, you must use a procedure.

```
//JAVA code fragment that calls the method "product".
            .
            .
            .
      int input = (some input from the user);
      result = product ( input );
      if (result > 65535) // (If result > 0xFFFF)
            result = 0;
            .
            .
            .
```

```
//Method to calculate the product of input and 255.
public static int product ( int input ){

      int result = 0;

      for (int i = 0; i < input; i++){
            result = result + 255;
      }

      return result;
}
```

Part 2:
Convert the procedure "product" to the machine language equivalent. For any address calculations, assume that the first instruction of the procedure is at address 0x0000 and so on.

Part 3:
Use the evaluation criteria provided to evaluate the ISA and the associated documentation.

How to turn-in?
Create a folder in your turn-in folder and name it "PeerEval". Within this folder, place a copy of the assembly language program, the machine language program fragment and the evaluation criteria. The evaluation criteria is in Word format. After you have completed the evaluation, note your observations and comments on the Word document. Convert the same to pdf format and place it in the PeerEval folder. On your website, do not add links to this folder or any files in this folder. Do not add this folder to your CVS folder.