CECS 285: Lab 4

OBJECTIVES:

- Gain familiarity with register banks
- Gain familiarity with using jumps and condition jumps for value coparisons

ACTIVITY 1

Write a program to fill the first 32 RAM locations with the value 0xAA. Accomplish this task by writing a program which performs the following set of operations:

1. With the default register bank selected, move the value 0xAA to each of the general purpose registers
2. Select register bank 1
3. Move the value 0xAA to each of the general purpose registers
4. Select register bank 2
5. Move the value 0xAA to each of the general purpose registers
6. Select register bank 3
7. Move the value 0xAA to each of the general purpose registers

Single step through a debug simulation and observe the value 0xAA being moved into each general purpose register in each register bank.

ACTIVITY 1 Deliverables:

- Your commented assembly language program source code
- A screenshot of Memory 1 window which shows that each of the first 32 RAM locations contains the value 0xAA.

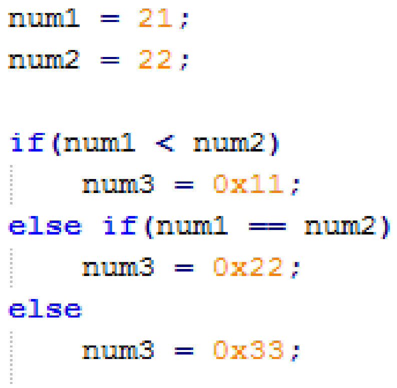
Proceed to Activity 2

ACTIVITY 2

Create the following variables using equate directives:

- **num1** which shall be mapped to RAM location **0x30**
- **num2** which shall be mapped to RAM location **0x31**
- **num3** which shall be mapped to RAM location **0x32**

Now write an assembly program according to the following C pseudocode



```
num1 = 21;
num2 = 22;

if(num1 < num2)
    num3 = 0x11;
else if(num1 == num2)
    num3 = 0x22;
else
    num3 = 0x33;
```

Test your program by assigning different values to variables **num1** and **num2** to ensure that your relational operator logic is working correctly.

ACTIVITY 2 Deliverables:

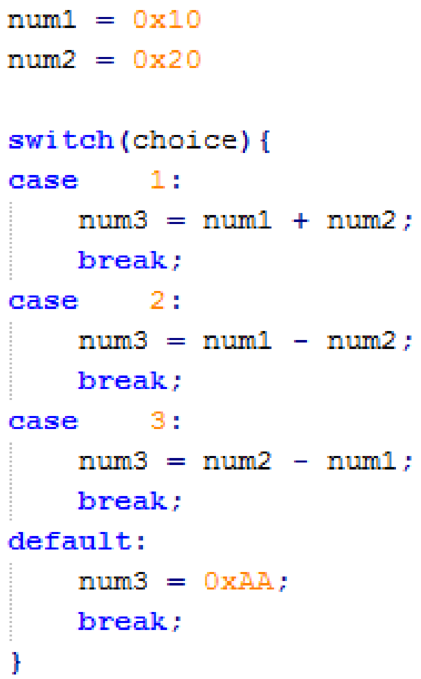
- Your assembly language program with comments to identify the instructions that correspond to each conditional statement
- A debug simulation screenshot showing the outcome of your program after a test of alternate values assigned to the num1 and num2 variables

ACTIVITY 3

Create the following variables using equate directives:

- **num1** which shall be mapped to RAM location **0x30**
- **num2** which shall be mapped to RAM location **0x31**
- **num3** which shall be mapped to RAM location **0x32**
- **choice** which shall be mapped to RAM location **0x40**

Now write an assembly program according to the following C pseudocode



```
num1 = 0x10
num2 = 0x20

switch(choice){
case    1:
        num3 = num1 + num2;
        break;
case    2:
        num3 = num1 - num2;
        break;
case    3:
        num3 = num2 - num1;
        break;
default:
        num3 = 0xAA;
        break;
}
```

Perform debug simulations to ensure your program's logic matches the operation of the pseudocode given for this activity.

ACTIVITY 3 Deliverables:

- Your assembly language program with comments to identify the instructions that correspond to conditional statement
- A debug screenshot showing the outcome of your program when **choice** is equal to **3**
- A debug screenshot showing the outcome of your program when **choice** is equal to **7**