Digital design lab 10

Assembly programming

Task 1

- We continue practicing Assembly programming
- If you haven’t finished on the last laboratory, try to multiple the upper and lower 4-bit number on the switches, and indicate the result on the LEDs.
- LD = SW[3:0]*SW[7:4]
- Hint: there is no instruction for multiplication in the MiniRISC system. Store the upper and lower 4-bits in two registers (e.g. r0 and r1). Initialize r2 with #0.
- Add the value of r0 to r2 and subtract #1 from r1 until r1=0.

Task 2

- Write an assembly program that looks for the greatest element in an array stored in the data memory.
- Initialize the data memory by adding the DATA and DB directives:

```
DATA
numbers:
DB 0x10, 0x20, 0x15, 0x25, 0x30, 0x01
```

- Hints:
  - Instruction mov r0, numbers will move the first element into r0
  - You can store the address also: mov r0, #numbers
  - (continued on next slide)

Task 2

- Hints (continued):
  - Now r0 stores the address of the first element.
  - If you add #1 to r0, it will store the address of the second element, etc.
  - Use the brackets in the mov instruction to read from a memory address stored in a register: mov r2, (r0)
  - If you have found the greatest element, write it to the LEDs.

Task 3

- Implement the greatest common divisor calculator algorithm
- Specification:
  - Set the first operand (A) on the switches, and confirm it with BT0.
  - Set the second operand (B) on the switches, and confirm it with BT1.
  - Launch the calculation of the GCD after BT2 is pressed.
  - Algorithm: if (A=B), then GCD=A. Else if A>B, then A = A-B. Else B = B-A.
  - When done, indicate the result on the LEDs.

Task 4

- Fibonacci sequence generator:
  - General expression for the nth element: $F_n = F_{n-1} + F_{n-2}$
  - The first two elements: $F_1 = F_2 = 1$
  - Put the elements into the data memory
  - What is the highest element that we can represent on 8 bits?

Task 5

- Fibonacci sequence generator on 16 bits
  - The number representation can be increased to 16 bits
  - Let r1r0 represent the first, and r3r2 the second 16-bit number.
  - The sum of the numbers can be calculated as:

```
add r2, r0
adc r3, r1
```

  - The adc (“add with carry”) instruction performs the addition with the carry bit: r3 = r3 + r1 + 8’b0000000C where C represents the carry bit.
  - Implement the Fibonacci sequence generator using 16-bit arithmetics.