User Manual

Kevin Wittlinger & Jaran Binning
I pledge my honor that I have abided by the Stevens Honor System

CPU name: Todd

Job description:
Kevin: worked on the construction of the circuit would eventually be remodeled to fit the design of py and assembly work, worked on designing the user manual, and helped with adding extra credit digit display

Jaran: Remodeled the circuit to be more in tune with design, worked on designing the user manual, designed py file to fit the circuit, made txt files including image and instructions, added extra credit digit display.

Our architecture has 4 general purpose registers (referred to as X0 - X3).
The CPU is built to handle two functions: addition and subtraction. The descriptions of each is as follows:

Addition -

- ADD Rd, Rn, Rm - 00000000 - 8 bits
  - Rd = Rn + Rm
  - Bit 1: Opcode - 0
    - 1 bit to determine operation
  - Bits 2-3: first register, Rn
    - Since there are 4 registers (2^2) we 2 bits to select a register
  - Bits 4-5: second register, Rm
    - 2 bits to determine register
  - Bits 6-7: destination register, Rd
    - 2 bits to determine register
  - Bit 8: fixed 0
    - 1 filler bit to make an even byte, the value of this one doesn't matter

Subtraction -

- SUB Rd, Rn, Rm - 10000000 - 8 bits
  - Rd = Rn - Rm
  - Bit 1: Opcode - 1
    - 1 bit to determine operation
  - Bits 2-3: first register, Rn

        - 2 bits to determine register
    - Bits 4-5: second register, Rm
        - 2 bits to determine register
    - Bits 6-7: destination register, Rd
        - 2 bits to determine register
    - Bit 8: fixed 0
        - 1 filler bit to make an even byte

To use these functions:

- Write all instructions in a text file titled “instructions.txt”. Each instruction should be in the above format exactly. (ADD Rd, Rn, Rm / SUB Rd, Rn, Rm)
- Run the “project2.py” script by running the module in an IDE shell It will assemble the instructions into machine code.
- The output will be sent to the “image.txt” file, which can then be imported into the instruction memory of the CPU through loading the image into memory.
- The CPU is then functional upon starting the clock

Some additional notes:

- Make sure the following inputs are set to the correct values:
    - RegWrite = 1
    - PC WE pin = 1
    - Instruction Mem M2 pin = 1
    - Instruction Mem M3 pin = 0
    - Addition to PC input = 00000001
    - PC = ff
    - Set each register X0-X3 with desired values
- Load “image.txt” into instruction memory

Extra credit:
Led display showing calculations of the hexadecimal.