Lab 03: Data and Instructions

### C:\Users\Donal_Heidenblad\Dropbox\Documents\Teaching\202101-Spring\INST346\Labs\Lab01-EncodingImages\bitmap_example_images\SMILE_FACE_20x20_PIXEL.png

### **Questions**

1. **How would you encode a black and white image to be shown on the toy’s display?**

**Black = 0 White = 1**

11111111111111111111

11111111111111111111

11111110000001111111

11111101111110111111

11110011111111001111

11110111111111101111

11101110011001110111

11011110011001111011

11011111111111111011

11011111111111111011

11011111111111111011

11011011111111011011

11011101111110111011

11101110000001110111

11110111111111101111

11110011111111001111

11111101111110111111

11111110000001111111

11111111111111111111

11111111111111111111

1. **How many *bytes* are required to represent one image that could be shown on this display?**

20 \* 20 = 400bit = 50byte

1. **Imagine that you are working on the next version of the game. This version will have a display that can show only the colors red, black, white. How would you encode an image for this display?**

**Black = 00, White = 11, Red = 01**

1. **How many bytes would be required to store an image with this new encoding?**

Each color has 2bit, then 20\*20 = 400 , 400\*2 = 800bit

800bit = 10byte

## Activity 01: Computer Instructions

To complete this activity, you will use the LMC simulator available here: <http://peterhigginson.co.uk/LMC/> . The LMC instruction set is included at the end of the document for your reference.

The LMC simulator will allow you to see what is happening during the instruction cycle. I have included two programs that you will answer questions about. You can load these programs into the LMC simulator by clicking in the left-most box under where it says, “Assembly Language Code” and hitting the “Submit” button. You can run the program by pressing the “Run” button or step through the program by hitting the “Step” button. You are also able to increase or decrease the speed of the simulation by hitting the “<<” or “>>” buttons while the program is running.

The simulator strays a little from the example in the lecture. The “Accumulator” is equivalent to the “Calculator” in the lecture. The “Program Counter” is equivalent to the “Instruction Counter” in the lecture. Red bubbles contain addresses. Blue bubbles contain values.

|  |  |
| --- | --- |
| **Program A** | **Program B** |
| INP  STA 99  INP  SUB 99  OUT | LDA 7  OUT  SUB 6  STA 7  BRP 0  HLT  DAT 1  DAT 5 |

### **Questions**

1. **Explain what Program A does.**
2. **How many instruction cycles does Program A take to complete?**
3. **Where is Program A storing the value from the first user input.**
4. **Explain what Program B does.**
5. **How many instruction cycles does it take Program B to complete?**
6. **Where does Program B store its data?**
7. **EXTRA CHALLENGE: Can you alter Program B to make it count up from zero to a number chosen by the user while the program is running?**

# Instruction Set

This is the instruction set for the Little Man Computer by Dr. Stuart Madnick (“Little Man Computer.”). **The opcode is 1 decimal digit; the operand is 2 decimal digits.**

Note: There is no instruction defined for the opcode 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | Mnemonic | Opcode | Operand | Description |
| Halt | HLT | 0 | 00 | Stop running |
| Add | ADD | 1 | XX \* | Add the contents of the memory address to the Accumulator |
| Subtract | SUB | 2 | XX \* | Subtract the contents of the memory address from the Accumulator |
| Store | STA | 3 | XX \* | Store the contents in the Accumulator in the memory address |
| Load | LDA | 5 | XX \* | Load Accumulator with contents of memory address |
| Branch always | BRA | 6 | XX \* | Set the PC to the value in the operand |
| Branch if zero | BRZ | 7 | XX \* | Set the PC to the operand if the Accumulator is zero |
| Branch if zero or positive | BRP | 8 | XX \* | Set the PC to the operand if the contents of the Accumulator is zero or positive |
| Input | INP | 9 | 01 | Retrieve user input and store in the Accumulator |
| Output | OUT | 9 | 02 | Output the contents of the Accumulator |
| Data storage | DAT |  |  | Label for a memory address; also can have a contents specified |

\* XX refers to a memory address.

“Little Man Computer.” In *Wikipedia*, January 27, 2020. <https://en.wikipedia.org/w/index.php?title=Little_man_computer&oldid=937810035>.