**CIS 350 – INFRASTRUCTURE TECHNOLOGIES**

**HOMEWORK # 3**

NAME(S):        Louis Ries

(You may do this homework in groups of 2 students maximum.)

**Topics**: Data formats and standards, representing numerical data, and LMC assembly language (Chapters 4-6)

1. Ex. 4.12, p. 134. (Assume that a typical page of text holds, say roughly 2500 characters.)

136,314 pages.

2. Ex. 4.22, p. 135. Using the Web as a resource, create a table that compares the features and capabilities of .PNG, .GIF, and .JPEG.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Colors | Size | use | quality |
| PNG | 16581375 colors + Alpha(Transparency) | 1-2mb | Online, editing, storage | good |
| GIF | 256 | <1mb | animaton | bad |
| JPEG | 16581375 colors | <1mb | Online, email | okay |

3. Find the 16-bit (2-byte) 2’s complementary binary representation for the decimal number

(-178)10. (Note that when you convert the 1’s complement to 2’s complement a carry may be generated.)

1111111101001110

4. Below is a Little Man program that solves exercise 6.9, p. 192. The program is very similar to the LMC program which you will find in the lecture notes on chapter 6 posted on Blackboard. The difference is that the program below is somewhat simpler as it uses only 2 branches (BRZ 09 and BR 01), whereas the program in the lecture notes uses 3 branches (BRP 05, BR 10, and BR 01). Assume that the following items in this order will be placed in the In-basket: 4, 34, 17, 19, and 20, one at a time. (The 4 is the count of numbers that follow.) What will the Out-basket contain after the program is executed? First try to understand each instruction thoroughly and then trace the execution of each instruction. Next write brief and precise comments which describe what each instruction does. Note that we initialized memory location 51 and 59 with 1 and 0, respectively. Memory location 59 will eventually store the sum (total) of four input values (34, 17, 19, and 20). Memory location 51 stores 1, a constant, by which memory location 50 will be decreased each time the loop is executed. Memory location 50 stores the current count, which is initially set to 4. Note that the program starts at address 00.

Address Instruction Comments start with // (for you to fill in)

(Mnemonics)

00 IN //Store the contents of in-basket in the calculator

01 STO 50 //Store the contents of the calculator in memory location 50

02 BRZ 09 //If calculator equals zero go to the instruction at address 09 else execute next instruction

03 IN // Store the contents of in-basket in the calculator

04 ADD 59 //Calculator equals calculator plus contents of memory location 59

05 STO 59 // Store the contents of the calculator in memory location 59

06 LDA 50 //Load the contents of memory location 50 to the calculator

07 SUB 51 //Calculator equals calculator minus contents of memory location 51

08 BR 01 //Go to the instruction at address 01 and execite it, unconditional branch

09 LDA 59 //Load the contents of memory location 59 to the calculator

10 OUT //output

11 HLT //halt

Address Contents Comments

50 DAT ? // data, contents initially unknown, stores current count

51 DAT 1 // data, used to decrement the count

…

59 DAT 0 // data, stores the sum, initially 0, at end stores

5. Assume now that the program from problem 4 will only read 3 numbers. That is, the following numbers in the following order will be placed, one at a time, in the In-basket: 2, 35, and 25, where 2 is the count of numbers that follow, and 35 and 25 are the numbers that are to be added. The first column in the table below shows the order in which the instructions from the program will be executed. Trace the execution of these instructions and determine the contents of the PC **before** and **after** each instruction is executed. Also, write down in the table the contents of the In-basket, Out-basket, Accumulator, and Memory locations 50, 51, and 59 **after** each instruction is executed. Memory locations 51 and 59 are initialized with 1 and 0, respectively. The entry 00 → 01 in the PC column means that the PC is 00 when the instruction IN started and is changed to 01 when the instruction IN is finished.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| The sequence in which instructions  are executed | PC  before → after | In-basket | Out-basket | Accumulator | Memory location 50 | Memory location 51 | Memory location 59 |
| IN | 00 → 01 | 2 | ? | 2 | ? | 1 | 0 |
| STO 50 | 01 → 02 | 2 | ? | 2 | 2 | 1 | 0 |
| BRZ 09 | 02 → 03 | 2 | ? | 2 | 2 | 1 | 0 |
| IN | 03 → 04 | 35 | ? | 35 | 2 | 1 | 0 |
| ADD 59 | 04 → 05 | 35 | ? | 35 | 2 | 1 | 0 |
| STO 59 | 05 → 06 | 35 | ? | 35 | 2 | 1 | 35 |
| LDA 50 | 06 → 07 | 35 | ? | 2 | 2 | 1 | 35 |
| SUB 51 | 07 → 08 | 35 | ? | 1 | 2 | 1 | 35 |
| BR 01 | 08 → 09 | 35 | ? | 2 | 2 | 1 | 35 |
| STO 50 | 09 → 10 | 35 | ? | 2 | 2 | 1 | 35 |
| BRZ 09 | 10 → 11 | 35 | ? | 2 | 2 | 1 | 35 |
| IN | 11 → 12 | 25 | ? | 25 | 2 | 1 | 35 |
| ADD 59 | 12 → 13 | 25 | ? | 60 | 2 | 1 | 35 |
| STO 59 | 13 → 14 | 25 | ? | 60 | 2 | 1 | 60 |
| LDA 50 | 14 → 15 | 25 | ? | 2 | 2 | 1 | 60 |
| SUB 51 | 15 → 16 | 25 | ? | 1 | 2 | 1 | 60 |
| BR 01 | 16 → 17 | 25 | ? | 2 | 2 | 1 | 60 |
| STO 50 | 17 → 18 | 25 | ? | 2 | 2 | 1 | 60 |
| BRZ 09 | 18 → 19 | 25 | ? | 2 | 2 | 1 | 60 |
| LDA 59 | 19 → 20 | 25 | ? | 60 | 2 | 1 | 60 |
| OUT | 20 → 21 | 25 | 60 | 60 | 2 | 1 | 60 |
| HLT | 21 → 21  Or  21 → 0 | 25 or 0 | 60 or 0 | 60 or 0 | 2 or 0 | 1 or 0 | 60 or 0 |