COMPS266F : Computer Architecture

Tutorial 0B : Introduction to LMC

Question 1 : Consider the following assembly language instructions with their descriptions.

|  |  |
| --- | --- |
| **Instruction** | **Description** |
| LDA | Load the content in memory location to accumulator |
| STO | Store the content of accumulator to a memory location |
| ADD | Add to the accumulator the content of memory location |
| SUB | Subtract the content in the memory location from accumulator |
| BRZ | Branch if the content of accumulator is zero |
| BRP | Branch if the content of accumulator is a positive value or zero |
| BR | Branch unconditionally |
| HLT | Program halt |

Below shows a section of memory inside a computer containing some data together with the assembly language instructions for a short program which uses the data.

On the left are the addresses of the memory locations and on the right the contents of each location.

|  |  |  |  |
| --- | --- | --- | --- |
| **Memory location** | **Content** | **Memory location** | **Content** |
| 601 | 030 | 740 | LDA 601 |
| 602 | 003 | 741 | ADD 602 |
| 603 | 009 | 742 | ADD 603 |
| : | : | 743 | STO 603 |
| : | : | 744 | HLT |

|  |  |  |
| --- | --- | --- |
| Task 1 – 1 : Find the value holds in accumulator (i.e. calculator, the value is 5 after execute location 739) after the location as below has been executed successively. | | |
|  | Location | Value in accumulator |
| Task 1 – 1 – 1 : | 740 | 30 |
| Task 1 – 1 – 2 : | 741 | 33 |
| Task 1 – 1 – 3 : | 742 | 42 |

|  |  |
| --- | --- |
| Task 1 – 2 : What is the value stored in location 603 after the program has halted? | 42 |

|  |
| --- |
| Task 1 – 3 : What is the function/purpose of the mailboxes 601-603? |
| They are data. |

Question 2 : The instruction used in little man computer (LMC) is listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| No need to specify mailbox location | | Need to specify the mailbox location | |
| Instruction | Three-digit code | Instruction | Three-digit code |
| Input (IN) | 901 (01 refers to the in basket, not the memory location) | Store (STO) | 3xx (xx is the memory address) |
| Output (OUT) | 902 (02 refers to the out basket, not the memory location) | Load (LDA) | 5xx (xx is the memory address) |
| Halt (HLT) | 000 (00 is the dummy) | Add (ADD) | 1xx |
| DATA (DAT) | 0yy (yy is the constant as specified) | Subtract (SUB) | 2xx |
|  |  | Branch if Zero (BRZ) | 7xx |
|  |  | Branch if Positive or Zero (BRP) | 8xx |
|  |  | Branch Unconditional (BR) | 6xx |

Table 1 : Instructions used in the Little Man Computer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task 2 – 1 : Given the following program, determine the output if the inputs are: | | | | |
| Mailbox | Assembly code | | Comment | |
| 00 | IN | | read the first number from the in-basket | |
| 01 | STO 20 | | put it into memory location 20 | |
| 02 | IN | | read the second number from the in-basket | |
| 03 | STO 21 | | put it into memory location 21 | |
| 04 | SUB 20 | | subtract with memory location 20 | |
| 05 | BRP 09 | | branch to location 9 if it is positive | |
| 06 | LDA 21 | | load the data in memory location 21 | |
| 07 | OUT | | put it into the out-basket | |
| 08 | BR 11 | | branch unconditionally to 11 | |
| 09 | LDA 20 | | load the data in memory location 20 | |
| 10 | OUT | | put it into the out-basket | |
| 11 | HLT | | program stops | |
|  | | Inputs | | Output |
| Task 2 – 1 – 1 : | | 23 and 44 | | 23 |
| Task 2 – 1 – 2 : | | 43 and 12 | | 12 |
| Task 2 – 1 – 3 : What is the main purpose of the program? (Within TEN words ONLY) | | | | |
| Compare two input values and show the smaller one. | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Self-Practice : The following program is to add three input numbers. What is wrong with this program? Rewrite the program in answer box so that it produces the correct result. | | | |
| Mailbox | Assembly code | Mailbox | Assembly code |
| 00 | IN | 00 | IN |
| 01 | STO 20 | 01 | STO 20 |
| 02 | IN | 02 | IN |
| 03 | ADD 20 | 03 | ADD 20 |
| 04 | IN | **04** | **STO 20** |
| 05 | ADD 20 | 05 | IN |
| 06 | OUT | 06 | ADD 20 |
| 07 | HLT | 07 | OUT |
|  |  | 08 | HLT |

Question 3 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task 3 – 1 : Consider the following program written in LMC language. Rewrite the program with mnemonic code (abbreviation for computer instructions of LMC language). You can refer the mnemonics and the machine code shown in previous tasks. | | | | |
| Mailbox | 3-Digit code | | Assembly Code | |
| 00 | 901 | | IN | |
| 01 | 310 | | STO 10 | |
| 02 | 901 | | IN | |
| 03 | 311 | | STO 11 | |
| 04 | 210 | | SUB 10 | |
| 05 | 808 | | BRP 08 | |
| 06 | 510 | | LDA 10 | |
| 07 | 111 | | ADD 11 | |
| 08 | 902 | | OUT | |
| 09 | 000 | | HLT (Generally first 000 is treat as halt) | |
| 10 | 000 | | DAT 00 (Data start to store after HLT) | |
| 11 | 000 | | DAT 00 | |
| Task 3 – 2 : What will be the output produced by this program in task 3 – 1 if below values entered respectively? | | | | |
|  | | Input values | | Output with explanation |
| Task 3 – 2 – 1 : | | 5 and 6 | | 1. Because 6 – 5 = 1 >= 0 so BRP operated then jump to location 08 directly to prompt output 1. |
| Task 3 – 2 – 2 : | | 6 and 5 | | 11. Because 5 – 6 < 0, so BRP is not operated, then run location 06 and 07 to perform 5 + 6 = 11 |

Question 4 : The following program is to perform addition and subtraction and output the result.

|  |  |  |  |
| --- | --- | --- | --- |
| Mailbox | Machine code | Mailbox | Machine code |
| 00 | IN | 05 | IN |
| 01 | STO 10 | 06 | SUB 10 |
| 02 | IN | 07 | OUT |
| 03 | ADD 10 | 08 | HLT |
| 04 | STO 10 |  |  |

|  |  |  |
| --- | --- | --- |
| Task 4 – 1 : Convert the Assembly code to machine code | | |
| Mailbox | Assembly code | Machine code |
| 00 | IN | 901 |
| 01 | STO 10 | 310 |
| 02 | IN | 901 |
| 03 | ADD 10 | 110 |
| 04 | STO 10 | 310 |
| 05 | IN | 901 |
| 06 | SUB 10 | 210 |
| 07 | OUT | 902 |
| 08 | HLT | 000 |

|  |
| --- |
| Task 4 – 2 : If the input values are input1, input2 and input3 and the output is output1, determine the relationship among input1, input2, input3, and output1. |
| output1 = input3 – (input1 + input2) |

|  |
| --- |
| Task 4 – 3 : In instructions 00 and 01, after reading the first input, explain why it needs to store the result in mailbox number 10. Explain why, in instructions 05 and 06, there is no need to store it after performing a subtraction. |
| Instructions 00 and 01 store the first input value, because the calculator needs to read the second input value from the in-basket to the calculator. If the first input value is not stored in mailbox 10, it (being in the calculator) will be overwritten by the second input value.  After performing instruction 06, there is no further need to use the calculator for other purposes, and the value in the calculator is the required result. That is why it can be output to the out-basket immediately. |