**Instruction Set Architecture (ISA Design)**

|  |  |  |
| --- | --- | --- |
| **(a)** |  |  |
| **Opcode** |  | **Description** |
| **Binary** | **Mnemonic** |  |
| 0000 | Load x | Load the contents of address X into AC. |
| 0001 | Store x | Store the contents of AC to address X. |
| 0010 | Subt x | Subtract the contents of address X from AC and store the result  in AC. |
| 0011 | Add x | Add the contents of address X to AC and store the result in AC. |
| 0100 | Input | Input a value from the keyboard into AC. |
| 0101 | Output | Output the value in AC to the display. |
| 0110 | Halt | Terminate the program. |
| 0111 | Skipcond | Skip the next instruction on condition. |
| 1000 | Jump x | Load the value of X into PC. |
| **(b)** |  |  |
| 1001 | Mult x | Multiplies the value of address x with the value in the AC then loads resulting value into AC. |
| 1010 | SRL x | Shift the contents of AC to the right for the number of times entered in the address x, filling in incoming bits with 0’s. |
| 1011 | SLL x | Shift the contents of AC to the left for the number of times entered in the address x, filling in incoming bits with 0’s. |

**(c)**

Skip condition checks to see if value in AC is more than zero. If not will change nothing and continue executing program as normal.

If it is more than zero it will increase the program counter by 1. It will only increment it by one because the program will increase the program counter by one anyway at the end of every instruction, this means the next instruction in line is skipped.