CPU INSTRUCTION:

• AC: The accumulator, which holds data values that the CPU needs to process. - Claudia

• MAR: The memory address register, which holds the memory address of the data being referenced. -Claudia

• MBR: The memory buffer register, which holds either the data just read from memory or the data ready to be written to memory. - Anthony

• PC: The program counter, which holds the address of the next instruction to be executed in the program. -Anthony

• IR: The instruction register, which holds the next instruction to be executed. - James

• InREG: The input register, which holds data from the input device. - Rokas

• OutREG: The output register, which holds data for the output device. - Rokas

**MACHINE INSTRUCTIONS**

Opcode Description Binary Mnemonic

0010 Load X Load the contents of address X into AC. - Anthony

0011 Store X Store the contents of AC to address X. - Anthony

0111 Skipcond Skip the next instruction on condition. - Anthony

* Skip if zero

1000 Subt X Subtract the contents of address X from AC and store the result in AC. - James

1001 Add X Add the contents of address X to AC and store the result in AC. -James

1111 Jump X Load the value of X into PC. - James

0100 Input Input a value from the keyboard into AC. - Claudia

0101 Output Output the value in AC to the display. - Claudia

0000 Halt Terminate the program. - Claudia

1100 Jump if greater than - Rokas

1101 Jump if less than - Rokas

1010 Jump if not zero – Rokas

9/11/2017

C FUNCTIONS

• Convert an integer from decimal representation to binary. - James

• Convert an integer from binary representation to decimal. - James

• Load some default content into the memory. -Claudia

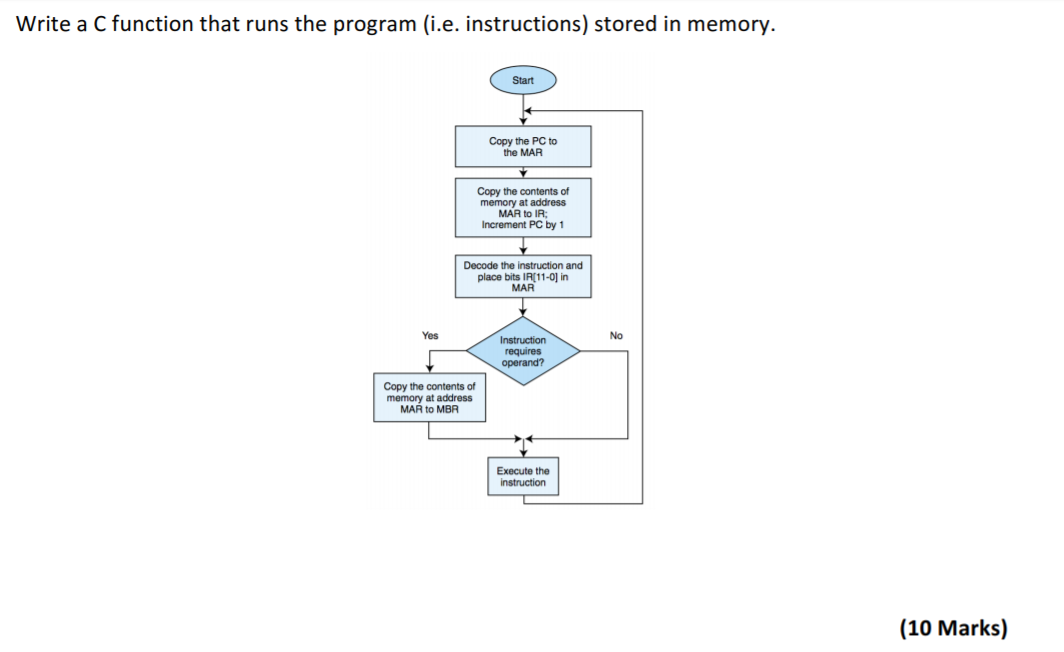
• Display the memory contents on the screen. Your function must also display the respective memory locations. -Claudia

• Convert the memory contents to assembly language and display the assembly language on the screen. Your function must display each assembly instruction on a new line. -Rokas

• Prompt the user to input the memory contents from the console (i.e. command line), read the input in and store it into the memory. Your function must be able to stop reading user input in, e.g. you may require that the user types "stop" or "exit" etc. - Anthony

• Load the instructions from a file (i.e. rather than the console) and store its contents into the memory. – Anthony

FETCH-DECODE-EXECUTE CYCLE



16/11/2017