Faculty of Science, Engineering and Technology



Computer Systems

Week 8

A computer screen shot of a blue screen

Description automatically generated***Part 8.1: Matchsticks***

***Part 8.2: Matchsticks***

A computer screen with a white screen

Description automatically generated

**Question 8.2.2(a): What is the condition that needs to be satisfied in order for this loop to occur?  Write this as a comparison using an inequality (ie., less than, greater than, less than or equal, greater than or equal)**

The condition that needs to be satisfied for this loop to occur is the input must be from 1 to 3, and not smaller than 0 and greater than 3.

**Question 8.2.2(b): What two ARM assembly instructions could be used to create a branch that only occurs under this condition?**

Two ARM assembly instructions could be used to create a branch that only occurs under this condition is BLT (Branch Less Than) and BGT (Branch Greater Than)

**Question 8.2.2(c): Based on the instructions you outlined in 8.2.2(b), what status bit would be set to 1 if the loop was to repeat?**

If the loop is to repeat, it means that the comparison did not find the two values to be equal. Therefore, the Z (Zero) status bit would not be set to 1 (i.e., Z = 0). In this case, the loop continues until the values are equal, at which point the Z bit will be set to 1, causing the program to exit the loop

**Question 8.2.2(d): What are all the modifications needed to the current program to implement this feature? Make the required modifications to your program to perform the task.**

The modifications needed to the current program to implement this feature is add:

+ “CMP R3,#0” which means that compare the value of R0 after “SUB R0,R0,R3” to 0.

A screenshot of a computer

Description automatically generated + “BEQ stop” indicates that branch equal to 0 stop the program.

***Part 8.3: Implementing computer player***

**Question 8.3.1(a): What bit-wise operation can we perform on the register holding the 32-bit pattern to set all bits in the register to zero except the least signficant 2 bits? Write this as a single line of code.**

A screenshot of a computer

Description automatically generated

To keep only the least significant 2 bits of a 32-bit value in a register (for example register R0), we can use a bitwise AND operation with a mask that has the least significant 2 bits set to 1 and all other bits set to 0. The mask for this operation is ‘0b00000000000000000000000000000011’, which is 3 decimals.

A screenshot of a computer

Description automatically generated **Question 8.3.1(b): Using a label named "select:". Write the code needed to repeatedly sample a random number (from. Random) until the value is in the range 1-3.  For now, just write this as a seperate program and test it.**

A computer screen shot of a blue and white computer

Description automatically generated **Question 8.3.2(a): Write the ARM assembly code that implements the algorithm expressed in the psuedo code above. Implement this as a seperate stand alone program and initialise R0 with a number at the start of your program to allow you to test the functionality.  You wil want to test it using different values in R0.**

A screenshot of a computer

Description automatically generated***Part 8.4: Implementing the full game***

A screenshot of a computer

Description automatically generated