1.6 Computer Architecture, Assembly and the LMC: Test Pack

Total Marks

OCR H446 · Calculators not allowed · about 25 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 Once an instruction has been fetched into the Current Instruction Register (CIR), it must be decoded before it can be executed.

(a) State what is meant by the 'opcode' part of a machine instruction, and what is meant by the 'operand' part. **(2)**

(b) Describe what happens during the decode stage of the fetch-decode-execute cycle. **(2)**

(c) State which CPU component is responsible for decoding the instruction. **(1)**

(Total for Question 1 is 5 marks)

© Revision Library | revisionlibrary.co.uk

2 The LMC program below inputs two numbers and outputs their product, calculated by repeated addition.

```
00 INP
01 STA 90
02 INP
03 STA 91
04 LDA 95
05 STA 92
06 LDA 91
07 BRZ 15
08 LDA 92
09 ADD 90
10 STA 92
11 LDA 91
12 SUB 96
13 STA 91
14 BRA 06
15 LDA 92
16 OUT
17 HLT
90 DAT 000
91 DAT 000
92 DAT 000
95 DAT 000
96 DAT 001
```

The program is run with the inputs 3, then 4, entered in that order.

(a) Complete a trace table showing the values in the accumulator (ACC) and in mailboxes 90, 91 and 92 after each pass through the loop (addresses 06 to 14), and the final value output. **(10)**

(b) Explain, in your own words, how this program uses repeated addition to calculate the product of the two input values. **(2)**

(c) State one limitation of using this method to multiply two large numbers, and explain your answer. **(2)**

(Total for Question 2 is 14 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.6 Computer Architecture, Assembly and the LMC: Test Pack

Question 1

(a) opcode: the part of the instruction that specifies which operation is to be carried out oe **B1**

(a) operand: the part of the instruction that specifies the data value, or the address of the data, to be used by that operation oe **B1**

(a) Answer: The opcode specifies which operation to perform; the operand specifies the data (or its address) that the operation acts on.

(b) the opcode held in the CIR is examined/interpreted oe **B1**

(b) this identifies which operation is being requested and which circuits/components must be activated to carry it out during the execute stage oe **B1**

(b) Answer: The opcode in the CIR is examined so that the CPU can identify which operation is being requested and which components need to be activated to carry it out.

(c) the Control Unit (CU) **B1**

(c) Answer: The Control Unit (CU).

Question 2

(a) initial setup correct after addresses 00-05: mailbox 90 = 3 (multiplicand), mailbox 91 = 4 (counter), mailbox 92 = 0 (result) **B1**

(a) pass 1 (counter=4): result updates from 0 to 3 (mailbox 92 = 3); counter decremented to 3 (mailbox 91 = 3) **B1**

(a) pass 2 (counter=3): result updates from 3 to 6 (mailbox 92 = 6); counter decremented to 2 **B1**

(a) pass 3 (counter=2): result updates from 6 to 9 (mailbox 92 = 9); counter decremented to 1 **B1**

(a) pass 4 (counter=1): result updates from 9 to 12 (mailbox 92 = 12); counter decremented to 0 **B1**

(a) on the fifth test at address 06/07, ACC (loaded from mailbox 91) = 0 **B1**

(a) the BRZ 15 condition is now met, so the program branches to address 15 rather than continuing the loop **B1**

(a) address 15 (LDA 92) loads the final result, 12, into the accumulator **B1**

(a) address 16 (OUT) outputs the value 12 **B1**

(a) final output correctly stated as 12, consistent with 3 x 4 **B1**

(a) Answer: Output = 12. Mailbox 92 (the running result) takes the values 0, 3, 6, 9, 12 across the four passes; mailbox 91 (the counter) counts down 4, 3, 2, 1, 0, ending the loop.

(b) the multiplicand (the first input) is added to a running total (the result) once on every pass through the loop oe **B1**

(b) a counter, initialised to the second input, is decremented by 1 on each pass, and the loop ends (via BRZ) once the counter reaches zero, by which point the multiplicand has been added exactly as many times as the value of the second input oe **B1**

(b) Answer: The multiplicand is added to a running result once per loop pass, while a counter (starting at the second input) is decremented each time; the loop stops once the counter reaches zero, by which point the multiplicand has been added exactly the right number of times to give the product.

(c) valid limitation identified, e.g. the number of loop passes (and so the number of instructions executed and the time taken) is directly proportional to the size of the second input oe **B1**

(c) explanation linking this to inefficiency for large multipliers, e.g. multiplying by a very large number would require a very large number of loop passes, making the program far slower than a processor's dedicated hardware multiplication instruction oe **B1**

(c) Answer: The method becomes slow for a large second input, because the number of loop passes (and instructions executed) grows in direct proportion to its value, unlike a single dedicated hardware multiply instruction on a real processor.

© Revision Library | revisionlibrary.co.uk