1.6 Computer Architecture, Assembly and the LMC: 30-minute Pack

Total Marks

OCR H446 · Calculators not allowed · about 30 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A processor's assembly language supports both immediate and direct addressing. LDM #45 loads the value 45 directly into the accumulator. LDD 45 loads, into the accumulator, whatever value is currently stored at memory address 45.

(a) Explain what is meant by immediate addressing. **(2)**

(b) Explain what is meant by direct addressing. **(2)**

(c) Memory address 45 currently holds the value 12. State the value that would be loaded into the accumulator by (i) LDM #45 and (ii) LDD 45, and (iii) explain why these two answers are different. **(3)**

(Total for Question 1 is 7 marks)

© Revision Library | revisionlibrary.co.uk

2 The Little Man Computer (LMC) is a simplified model used to help understand how a real processor carries out the fetch-decode-execute cycle.

(a) State what the LMC is used to model. **(1)**

(b) Complete the table to match each part of the LMC model to its real computer equivalent. **(4)**

i) Calculator:

ii) In-tray:

iii) Out-tray:

iv) Mailboxes:

(c) State the number of mailboxes in the standard LMC model, and the range of values each mailbox can hold. **(2)**

(Total for Question 2 is 7 marks)

© Revision Library | revisionlibrary.co.uk

3 The table below sets out the LMC instruction set. Some cells have been left blank.

Opcode	Mnemonic	Meaning
1xx	(i)	Add the value in mailbox xx to the accumulator
(ii)	SUB	Subtract the value in mailbox xx from the accumulator
3xx	STA	(iii)
5xx	LDA	Load the value in mailbox xx into the accumulator
(iv)	BRA	Branch always: set the program counter to xx
7xx	(v)	If the accumulator is zero, set the program counter to xx
(vi)	BRP	If the accumulator is zero or positive, set the program counter to xx
901	(vii)	Input a value from the in-tray into the accumulator
(viii)	HLT	Stop the program executing

(a) Complete the table by giving the missing mnemonic or opcode for each of (i) to (viii). **(8)**

(b) Explain the purpose of a DAT instruction in an LMC program, giving an example. **(2)**

(Total for Question 3 is 10 marks)

4 Discuss the extent to which the Little Man Computer (LMC) is a realistic model of how a modern processor carries out the fetch-decode-execute cycle. Your answer should consider registers, addressing modes and instruction sets.

(Total for Question 4 is 6 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.6 Computer Architecture, Assembly and the LMC: 30-minute Pack

Question 1

(a) the operand given in the instruction is the actual data value to be used, not an address oe **B1**

(a) correct reference to the example, e.g. LDM #45 loads the literal value 45 into the accumulator **B1**

(a) Answer: In immediate addressing the operand is the data value itself, e.g. LDM #45 loads the value 45 directly into the accumulator.

(b) the operand given in the instruction is the address of the memory location holding the data to be used oe **B1**

(b) correct reference to the example, e.g. LDD 45 loads whatever value is currently stored at address 45 **B1**

(b) Answer: In direct addressing the operand is the address of the memory location holding the data, e.g. LDD 45 loads whatever value is stored at address 45.

(c) (i) 45, cao **B1**

(c) (ii) 12, cao **B1**

(c) (iii) LDM treats its operand as the data itself, whereas LDD treats its operand as an address and fetches the value stored there, so the two instructions use the number 45 in completely different ways oe **B1**

(c) Answer: (i) 45 (ii) 12 (iii) LDM uses 45 as the value; LDD uses 45 as an address, giving the different value 12 stored there.

Question 2

(a) a simplified model of how a (von Neumann) processor carries out the fetch-decode-execute cycle oe **B1**

(a) Answer: A simplified model of how a processor carries out the fetch-decode-execute cycle.

(b) i) Accumulator (ACC) **B1**

(b) ii) input device oe **B1**

(b) iii) output device oe **B1**

(b) iv) main memory/RAM oe **B1**

(b) Answer: Calculator = accumulator; in-tray = input device; out-tray = output device; mailboxes = main memory.

(c) 100 mailboxes, numbered 00 to 99 **B1**

(c) each mailbox holds a 3-digit number, from 000 to 999 oe **B1**

(c) Answer: 100 mailboxes (numbered 00-99), each holding a 3-digit number from 000 to 999.

Question 3

(a) (i) ADD **B1**

(a) (ii) 2xx **B1**

(a) (iii) Store the value in the accumulator into mailbox xx oe **B1**

(a) (iv) 6xx **B1**

(a) (v) BRZ **B1**

(a) (vi) 8xx **B1**

(a) (vii) INP **B1**

(a) (viii) 000 **B1**

(a) Answer: (i) ADD (ii) 2xx (iii) Store the value in the accumulator into mailbox xx (iv) 6xx (v) BRZ (vi) 8xx (vii) INP (viii) 000

(b) DAT reserves a mailbox to be used for storing a data value/variable, rather than as an executable instruction oe **B1**

(b) valid example given, e.g. total DAT 0 reserves a mailbox labelled 'total', initialised to the value 0 **B1**

(b) Answer: DAT reserves a mailbox for storing a data value rather than an instruction, e.g. total DAT 0 reserves a mailbox called total, initialised to 0.

© Revision Library | revisionlibrary.co.uk

Question 4

Level 3 A well-balanced discussion covering similarities and differences across registers, addressing modes and instruction sets, with a clear, justified overall conclusion on how realistic the model is. Technical terms are used accurately throughout.	**5-6**
Level 2 A discussion covering at least two of registers/addressing modes/instruction sets, with some valid comparison points, but a conclusion that is only partly justified, or an imbalance towards similarities or differences only.	**3-4**
Level 1 Limited comparison, largely a description of the LMC or of a real processor rather than a comparison between them; little or no reference to registers, addressing modes or instruction sets.	**1-2**
Level 0 No relevant content, or the response does not address the question.	**0**

Indicative content:

- The LMC uses a single general-purpose register (the accumulator/calculator), whereas a real processor has several distinct registers (PC, MAR, MDR, CIR, ACC, and sometimes an index register or general-purpose registers)
- The LMC's own cycle (fetch a mailbox, decode the opcode, execute it) mirrors the real fetch-decode-execute stages, so it is a valid simplified teaching model of the underlying process and of concepts such as branching and the role of the PC
- The LMC effectively only supports direct addressing of mailboxes for its ADD/SUB/STA/LDA instructions; it has no genuine immediate, indirect or indexed addressing, unlike a real processor which typically supports all of these
- The LMC's instruction set is very small (around ten instructions), compared with the much larger and more varied instruction sets of real processors
- LMC mailboxes store decimal numbers (000-999) directly as data or instructions, whereas real memory stores binary patterns, with instructions encoded in machine code as a binary opcode and operand
- The LMC has no concept of interrupts, pipelining, cache, or clock-driven timing, all of which significantly affect the performance of a real processor
- Overall the LMC is a useful, simplified model for understanding the sequence of fetch, decode, execute and basic branching, but it omits many features (multiple registers, full addressing modes, binary representation, performance features) present in a real processor, so it should be regarded as only a partial representation
- Credit either an overall conclusion (broadly realistic vs mostly unrealistic) provided it is supported by specific, accurate comparison points from at least two of the three areas named in the question

© Revision Library | revisionlibrary.co.uk