Chapter 6 The Little Man Computer

Consider this model of the LMC and answer the questions below.



1) The LOAD instruction copies data from the

a) in basket to a mailbox.

b) calculator to a mailbox.

c) in basket to the calculator.

**d) mailbox to the calculator.**

2) The STORE instruction copies data from the

a) in basket to a mailbox.

b) mailbox to the calculator.

**c) calculator to a mailbox.**

d) in basket to the calculator.

3) The ADD instruction adds data from

a) the in basket to a mailbox.

**b) a mailbox to the calculator.**

c) a mailbox to the in basket.

d) one mailbox to another mailbox.

4) The SUBTRACT instruction subtracts data in

a) the calculator from a mailbox.

b) the in basket from a mailbox.

**c) a mailbox from the calculator.**

d) one mailbox from another mailbox.

5) The INPUT instruction takes data from the

a) in basket and places it in a mailbox.

b) mailbox and places it in the in basket.

c) mailbox and places it in the calculator.

**d) in basket and places it in the calculator.**

6) The OUTPUT instruction takes data from the

a) out basket and places it in a mailbox.

b) mailbox and places it in the out basket.

c) out basket and places it in the calculator.

**d) calculator and places it in the out basket.**

7) The COFFEE BREAK (HALT) instruction

**a) pauses the program.**

b) clears all mailboxes.

c) empties the out basket.

d) ignores the address portion of the instruction.

8) A LOAD command will leave the original data in the mailbox

a) deleted.

b) corrupted.

**c) unchanged.**

d) overwritten.

9) A LOAD command will leave the original data in the calculator

a) deleted.

b) corrupted.

c) unchanged.

**d) overwritten.**

10) A STORE command will leave the original data in the mailbox

a) deleted.

b) corrupted.

c) unchanged.

**d) overwritten.**

11) A STORE command will leave the original data in the calculator

a) deleted.

b) corrupted.

**c) unchanged.**

d) overwritten.

12) An ADD command will leave the original data in the mailbox

a) deleted.

b) corrupted.

**c) unchanged.**

d) overwritten.

13) An INPUT command will leave the original data in the calculator

a) deleted.

b) corrupted.

c) unchanged.

**d) overwritten.**

14) An OUTPUT command will leave the original data in the calculator

a) deleted.

b) corrupted.

**c) unchanged.**

d) overwritten.

15) Which sequence of commands is needed to enter two numbers into the LMC (using the INPUT command)?

a) INPUT, ADD, INPUT

b) INPUT, LOAD, INPUT

**c) INPUT, STORE, INPUT**

d) INPUT, ENTER, INPUT

16) The BRANCH UNCONDITIONALLY instruction changes the value in the

a) mailbox.

b) calculator.

c) out basket.

**d) program counter (also called instruction location counter).**

17) The BRANCH ON ZERO instruction "jumps" if the value in the

a) mailbox is zero.

b) in basket is zero.

**c) calculator is zero.**

a) instruction location counter is zero.

18) The BRANCH ON POSITIVE instruction "jumps" if the value in the

a) mailbox is positive.

b) in basket is positive.

**c) calculator is positive.**

d) instruction location counter is positive.

19) The instruction cycle can be broken into these two parts

a) fetch and decode.

**b) fetch and execute.**

c) decode and execute.

d) execute and increment.

20) The LMC knows which mailbox contains the next task by looking at the

a) calculator.

b) in basket.

c) current mailbox.

**d) program counter (instruction location counter).**

*Please refer to this table of op codes for problems 21 through 33*

 Opcode Definition

0 Halt

1 ADD

2 SUBTRACT

3 STORE

5 LOAD

6 BRANCH UNCONDITIONALLY

7 BRANCH ON ZERO

8 BRANCH ON POSITIVE

901 INPUT

902 OUTPUT

*Please refer to this table of Mailboxes and Contents for problems 21 through 26*

Mailbox Contents

00 505

01 106

02 507

03 902

04 000

05 1 DAT

06 3 DAT

07 6 DAT

21) What is the value in the **calculator** after the first instruction (505) is completed?

**a) 1**

b) 2

c) 3

d) 6

22) What is the value in the **program counter (instruction location counter)** after the first instruction (505) is completed?

**a) 01**

b) 02

c) 03

d) 04

23) What is the value in the **calculator** after the fetch but before the execute portion of second instruction (106)?

**a) 1**

b) 2

c) 3

d) 6

24) What is the value in the **calculator** after the second (106) instruction is completed?

a) 1

b) 2

c) 3

**d) 4**

25) What is the value in the **calculator** after the third instruction (507) is completed?

a) 1

b) 2

c) 3

**d) 6**

26) What is the value in the **calculator** when the program is finished, *i.e.,* when the halt instruction has been executed?

a) 1

b) 2

c) 3

**d) 6**

*Please refer to this table of Mailboxes and Contents for problems 27 through 33*

Mailbox Contents

00 602

01 106

02 507

03 206

04 902

05 000

06 1 DAT

07 3 DAT

08 6 DAT

27) What is the value in the **calculator** after the first instruction (602) is completed?

a) 1

b) 2

c) 3

**d) unknown**

28) What is the value in the **program counter (instruction location counter)** after the first instruction (602) is completed?

a) 01

**b) 02**

c) 03

d) 06

29) What is the value in the **calculator** after the instruction 507 is completed?

a) 1

b) 2

**c) 3**

d) 6

30) What instruction is never executed?

a) 602

**b) 106**

c) 507

d) 902

31) What data (DAT) **value** is never used?

a) 1

b) 2

c) 3

**d) 6**

32) What is the value in the **program counter (instruction location counter)** when the program is finished?

a) 03

b) 04

c) 05

**d) 06**

33) What is the value in the **calculator** when the program is finished?

a) 1

**b) 2**

c) 3

d) 6

34) In the von Neumann architecture, memory holds

a) data only.

b) instructions only.

**c) data and instructions.**

d) neither data and instructions.

35) In the von Neumann architecture, memory is addressed

**a) by location number.**

b) by the value stored.

c) by contents of the memory location.

d) None of the above.

Discussion Questions

 *Please refer to this table of op codes for the discussion questions*

 Opcode Definition

0 Halt

1 ADD

2 SUBTRACT

3 STORE

5 LOAD

6 BRANCH UNCONDITIONALLY

7 BRANCH ON ZERO

8 BRANCH ON POSITIVE

901 INPUT

902 OUTPUT

1) Using the LMC program below, add comments to explain what the result (value in the calculator) is after the completion of each instruction. The first one is completed as an example.

**Mailbox Contents Result after completion**

00 901 *Read contents from in basket and store in calculator*

01 319

02 901

03 320

04 219

05 709

06 518

07 902

08 000

09 517

10 902

11 000

----------------

17 DAT

18 DAT

19 DAT

20 DAT

**Sol:**

**Mailbox Contents Result after completion**

**00 901 Read contents from in basket and store in calculator**

**01 319 Store value of calculator in mailbox 19**

**02 901 Read contents from in basket and store in calculator**

**03 320 Store value of calculator in mailbox 20**

**04 219 Subtract the value in calculator by value in mailbox 19**

**05 709 Branch to mailbox 09 if calculator is 0**

**06 518 Load value in mailbox 18 to calculator**

**07 902 Move value in calculator to out basket**

**08 000 Halt**

**09 517 Load value in mailbox 17 to calculator**

**10 902 Move value in calculator to out basket**

**11 000 Halt**

**----------------**

**17 DAT**

**18 DAT**

**19 DAT**

**20 DAT**

2) Refer to the LMC program in question (1). Suppose the contents of mailbox 17 = 5; contents of mailbox 18 = 1.

a) What is the final value in out basket if the first in basket is 56 and second in basket is 89?

b) What is the final value in out basket if the first in basket is 75 and second in basket is 75?

c) What is the final value in out basket if the first in basket is 89 and second in basket is 56?

**Sol: The order of input does not matter.**

**a) 5 (because 56 and 89 are different)**

**b) 1 (because 75 and 75 are the same)**

**c) 5 (because 56 and 89 are different)**

3) Describe what the LMC program in question (1) does. Suppose the contents of mailbox 17 = 5; contents of mailbox 18 = 1.

**Sol: This program displays to the out basket "5" if the inputs are different, and a “1” if they are same; the order of input does not matter.**

4) Refer to the LMC program below and the table of op codes given above.

a) What is the first number placed in the out basket?

b) What is the last number placed in the out basket?

**Mailbox Contents**

00 517

01 218

02 902

03 705

04 601

05 000

……………..

17 100 DAT

18 2 DAT

**Sol:**

**a) 98**

**b) 0**

5) Refer to the LMC program in question (4). Change DAT in mailbox 18 to 4.

a) What is the first number placed in the out basket?

b) What is the last number placed in the out basket?

**Sol:**

**a) 96**

**b) 0**

6) Describe what the LMC program in question (4) does.

**Sol: This program displays even numbers to the out basket counting backwards from 98. The first number displayed is 98, and the last number displayed is 0.**

7) Refer to the LMC program below. Write down what the calculator will hold after the instruction is complete in each loop. The first one is completed as an example.

**Mailbox Contents Calculator after instruction is complete**

 **Loop1 Loop2 Loop3 Loop4**

00 517 *1 2 3 4*

01 118

02 317

03 219

04 710

05 600

……………..

17 1 DAT

18 1 DAT

19 5 DAT

**Sol:**

**Mailbox Contents Calculator after instruction is complete**

 **Loop1 Loop2 Loop3 Loop4**

**00 517 1 2 3 4**

**01 118 2 3 4 5**

**02 317 2 3 4 5**

**03 219 -3 -2 -1 0**

**04 710 -3 -2 -1 0**

**05 600 -3 -2 -1 NA**

**……………..**

**17 1**  DAT

**18 1** DAT

**19 5**  DAT

8) Refer to the LMC program in question (7). How did the contents of mailboxes 17-19 change for each loop?

**Sol: The content of address 17 goes: 2-3-4-5. The others don’t change.**

9) What instruction should be placed in mailbox 02 so the program loops 4 times? Refer to the table of op codes above.

**Mailbox Contents**

**00 517**

**01 218**

**02 ???**

**03 317**

**04 902**

**05 600**

**06 000**

**……………..**

**17 10**

**18 2**

**Sol: The contents of address 02 should be 706 to loop 4 times.**

10) Describe what the following LMC program does. Refer to the table of op codes above.

Mailbox Contents

00 901

01 309

02 207

03 902

04 708

05 602

06 000

07 1

08 000

09 DAT

**Sol: The program gets a number from the user. It then loops by that number displaying the loop count to the out basket each time. Example: in basket 4: Display to out basket: 3 – 2 – 1– 0.**

11) The contents in memory occasionally have to be moved to another area of memory. When that happens, the mailbox references must be adjusted so that the program continues to function properly. Rewrite the LMC code in problem (10) so that it occupies mailboxes 05 through 14 only; mailboxes 01 through 04 will be used by another program, so they can't be used. Assume that instruction 605 remains in mailbox 00.

Mailbox Contents

00 605

01 used by other program

02 used by other program

03 used by other program

04 used by other program

05 ???

06 ???

07 ???

08 ???

09 ???

10 ???

11 ???

12 ???

13 ???

14 ???

**Sol:**

**Mailbox Contents**

**00 605**

**01 used by other program**

**02 used by other program**

**03 used by other program**

**04 used by other program**

**05 901**

**06 314**

**07 212**

**08 902**

**09 713**

**10 606**

**11 000**

**12 1**

**13 000**

**14 DAT**

12) Describe the LMC three-digit instruction format. How does the LMC know what part of the value is an instruction, and what part is an address?

**Sol: The format of an instruction takes the form XYY**

**Where X is the op code (0-9) and YY is the address (00-99).**

**There is no op code X=4.**

**The LMC only has to check the first digit for the opcode; the remaining 2 digits are an address.**

13) How does the LMC "know" if a particular mailbox contains data or instructions?

**Sol: LMC does not know if the value in a mailbox is an instruction or not. The first instruction location is 00 and the next is determined by the program counter. If the LMC happens upon a memory value that wasn’t intended to be an instruction, the LMC would try to execute it. Alternatively, if the program counter points to a particular mailbox, it is assumed to contain an instruction, not data.**

14) What happens if the LMC is executing a program and never encounters a "HALT" command?

**Sol: The program will continue to execute until it encounters a location that contains “000” (the HALT instruction, even it it’s data) or a location that begins with a 4, which is an invalid operation code. This assumes that incrementing the program counter past 99 returns it to zero.**

15) Describe how the LMC is von Neumann architecture.

**Sol: We need three things:**

**1) Memory holds both programs and data; this is known as the stored program concept. The stored program concept allows programs to be changed easily.**

**Yes, LMC memory holds programs and data**

**2) Memory is addressed linearly; that is, there is a single sequential numeric address for each and every memory location.**

**Yes, LMC memory is addressed linearly**

**3) Memory is addressed by the location number without regard to the data contained within.**

**Yes, LMC memory is addressed without regard for the contents.**

**Section 6.6 A Note Regarding Computer Architectures**

**Alternative answer:**

**1. The computer consists of a CPU and memory, with facility for input and output**

**2. The memory holds both instructions and data**

**3. The instructions are executed sequentially, that is, one at a time.**

Solutions

|  |  |  |
| --- | --- | --- |
| Problem | Answer | Section in text / comments |
| 1 | d | Section 6.2 Operation of the LMC |
| 2 | c | Section 6.2 Operation of the LMC |
| 3 | b | Section 6.2 Operation of the LMC |
| 4 | c | Section 6.2 Operation of the LMC |
| 5 | d | Section 6.2 Operation of the LMC |
| 6 | d | Section 6.2 Operation of the LMC |
| 7 | a | Section 6.2 Operation of the LMC |
| 8 | c | Section 6.2 Operation of the LMC |
| 9 | d | Section 6.2 Operation of the LMC |
| 10 | d | Section 6.2 Operation of the LMC |
| 11 | c | Section 6.2 Operation of the LMC |
| 12 | c | Section 6.2 Operation of the LMC |
| 13 | d | Section 6.2 Operation of the LMC |
| 14 | c | Section 6.2 Operation of the LMC |
| 15 | c | Section 6.4 An Extended Instruction Set  |
| 16 | d | Section 6.4 An Extended Instruction Set  |
| 17 | c | Section 6.4 An Extended Instruction Set  |
| 18 | c | Section 6.4 An Extended Instruction Set  |
| 19 | b | Section 6.5 The Instruction Cycle  |
| 20 | d | Section 6.5 The Instruction Cycle  |
| 21 | a | Multiple Sections 6.2, 6.4, 6.5 |
| 22 | a | Multiple Sections 6.2, 6.4, 6.5 |
| 23 | a | Multiple Sections 6.2, 6.4, 6.5 |
| 24 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 25 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 26 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 27 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 28 | b | Multiple Sections 6.2, 6.4, 6.5 |
| 29 | c | Multiple Sections 6.2, 6.4, 6.5 |
| 30 | b | Multiple Sections 6.2, 6.4, 6.5 |
| 31 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 32 | d | Multiple Sections 6.2, 6.4, 6.5 |
| 33 | b | Multiple Sections 6.2, 6.4, 6.5 |
| 34 | c | Section 6.6 A Note Regarding Computer Architectures  |
| 35 | a | Section 6.6 A Note Regarding Computer Architectures  |