



Operation is subtraction, sign of  $(73)_{10}$  is changed.  $S = 0 \text{ XOR } 1 = 1$ .

The two's complement of  $(73)_{10}$  is  $(10110111)_2$ .

There is no overflow ... so  $(10101001)_2 = (-41)_{10}$

$A_S$	0		0	1	0	0	0	0	0
$B_S$	1	+	0	1	1	0	1	1	1
			1	0	1	0	1	1	1
$R_S$	1		0	1	0	1	0	0	1

- e. The floating-point operation  $(-12.625)_{10} + (451.00)_{10}$  in 32-bit IEEE format, and show the result in decimal. (6%)

$$(-12.625)_{10} + (451.00)_{10} = -(1100.101)_2 + (111000011)_2$$

$$= -2^3 * (1.100101)_2 + 2^8 * (1.11000011)_2$$

$$A \rightarrow 1\ 10000010\ 100101000000000000000000$$

$$B \rightarrow 0\ 10000111\ 110000110000000000000000$$

We denormalize the numbers by adding the hidden 1's to the mantissa and incrementing the exponent. Now both denormalized mantissas are 24 bits and include the hidden 1's.

$$A \rightarrow 1\ 10000011\ 110010100000000000000000$$

$$B \rightarrow 0\ 10001000\ 111000011000000000000000$$

We align the mantissas. We increment the first exponent by 5 and shift its mantissa to the right five times.

$$A \rightarrow 1\ 10001000\ 000001100101000000000000$$

$$B \rightarrow 0\ 10001000\ 111000011000000000000000$$

Now we do sign-and-magnitude addition treating the sign and the mantissa of each number as one integer stored in sign-and-magnitude representation.

$$R \rightarrow 0\ 10001000\ 110110110011000000000000$$

$$R \rightarrow 0\ 10000111\ 101101100110000000000000$$

$$E = (10000111)_2 = 135, M = 10110110011$$

$$(1.10110110011)_2 \times 2^{135-127} = (110110110.011)_2 = 438.375$$

3. We know that the operation of one's complement simply reverses (flips) each bit, and the each number must be 0 and 1 (not over two). The two's complement of number is to first take the one's complement and then add 1 to the result. So assuming three-digit allocation, find **the sixteen's complement** of the following hexadecimal numbers. (6%)

a.  $-1E2 \quad (E1E)_{16}$

b.  $+FE1 \quad (+FE1)_{16} \rightarrow$  Overflow occurs because it is not in the range  $(-800)_{16}$  to  $(7FF)_{16}$

4. In a positional number system with base  $b$ , the largest integer number that can be represented using  $k$  digits is  $b^k - 1$ . Find the largest number in hexadecimal systems with four digits. And find the minimum number of digits needed in the octal system for the seven-digit decimal number. (6%)

(1)  $16^4 - 1 = 65535$

(2) 8

5. Why is normalization necessary in the floating-point representation of real numbers? And what is the mantissa? (4%)

(1) Normalization is necessary to make calculations easier.

(2) Mantissa is the bit sequence to the right of the decimal point after normalization.

6. In some countries vehicle license plates have two decimal digits (0 to 9). How many distinct plates can we have? If the digit **0 is not allowed** on the license plate, how many distinct plates can we have? (4%)

(1)  $10^2 = 100$  if zero is allowed.

(2)  $9^2 = 81$  if zero is not allowed.

7. An imaginary computer has sixteen data register (R0 to R15), 1024 words in memory and 32 different instructions (add, subtract, and so on). What is the minimum size of an instruction in bits if a typical instruction uses the following format: **instruction M R2**? (5%)

We need 4 bits to determine the instruction ( $2^4 = 16$ ). We need 10 bits to address a word in memory ( $2^{10} = 1024$ ). We need 4 bits to address a register ( $2^5 = 32$ ). The size of the instruction is therefore  $(4 + 10 + 5)$  or 19 bits.

Since the size of the instruction is 19 bits, we must have 19-bit data registers.

8. A computer uses memory-mapped I/O addressing. The address bus uses 10 lines (10 bits). If memory made up of 1,000 words, how many **three-register** controllers can be accessed by the computer. (4%)

The address bus uses 10 lines which mean that it can address  $2^{10} = 1024$  words. Since the memory is made of 1000 words and the system uses shared. (memory-mapped I/O) addressing,  $1024 - 1000 = 24$  words are available for I/O controllers. If each controller has 3 registers, then  $24/3 = 8$  controllers can be accessed in this system.

9. Using the instruction set of the simple computer in following table, write the code a program that performs the following calculation:  $D \leftarrow A + B + C$ , in which A, B, C and D are integers in two's complement format. The user types the value of A, B, and C, and the value of D is displayed on the monitor. The keyboard is assumed to be memory location  $(FE)_{16}$ , and the monitor is assumed to be  $(FF)_{16}$ . (6%)

**Table 5.4** List of instructions for the simple computer

Instruction	Code				Action
	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>	
HALT	0				Stops the execution of the program
LOAD	1	R <sub>D</sub>	M <sub>S</sub>		R <sub>D</sub> ← M <sub>S</sub>
STORE	2	M <sub>D</sub>		R <sub>S</sub>	M <sub>D</sub> ← R <sub>S</sub>
ADDI	3	R <sub>D</sub>	R <sub>S1</sub>	R <sub>S2</sub>	R <sub>D</sub> ← R <sub>S1</sub> + R <sub>S2</sub>
ADDF	4	R <sub>D</sub>	R <sub>S1</sub>	R <sub>S2</sub>	R <sub>D</sub> ← R <sub>S1</sub> + R <sub>S2</sub>
MOVE	5	R <sub>D</sub>	R <sub>S</sub>		R <sub>D</sub> ← R <sub>S</sub>
NOT	6	R <sub>D</sub>	R <sub>S</sub>		R <sub>D</sub> ← $\bar{R}_S$
AND	7	R <sub>D</sub>	R <sub>S1</sub>	R <sub>S2</sub>	R <sub>D</sub> ← R <sub>S1</sub> AND R <sub>S2</sub>
OR	8	R <sub>D</sub>	R <sub>S1</sub>	R <sub>S2</sub>	R <sub>D</sub> ← R <sub>S1</sub> OR R <sub>S2</sub>
XOR	9	R <sub>D</sub>	R <sub>S1</sub>	R <sub>S2</sub>	R <sub>D</sub> ← R <sub>S1</sub> XOR R <sub>S2</sub>
INC	A	R			R ← R + 1
DEC	B	R			R ← R - 1
ROTATE	C	R	n	0 or 1	Rot <sub>n</sub> R
JUMP	D	R	n		IF R <sub>0</sub> ≠ R then PC = n, otherwise continue

Key: R<sub>S</sub>, R<sub>S1</sub>, R<sub>S2</sub>: Hexadecimal address of source registers  
R<sub>D</sub>: Hexadecimal address of destination register  
M<sub>S</sub>: Hexadecimal address of source memory location  
M<sub>D</sub>: Hexadecimal address of destination memory location  
n: hexadecimal number  
d<sub>1</sub>, d<sub>2</sub>, d<sub>3</sub>, d<sub>4</sub>: First, second, third, and fourth hexadecimal digits

(00) <sub>16</sub>	(1FFE) <sub>16</sub>	// RF ← MFE, Input A from keyboard to RF
(01) <sub>16</sub>	(240F) <sub>16</sub>	// M40 ← RF, Store A in M40
(02) <sub>16</sub>	(1FFE) <sub>16</sub>	// RF ← MFE, Input B from keyboard to RF
(03) <sub>16</sub>	(241F) <sub>16</sub>	// M41 ← RF, Store B in M41
(04) <sub>16</sub>	(1FFE) <sub>16</sub>	// RF ← MFE, Input C from keyboard to RF
(05) <sub>16</sub>	(242F) <sub>16</sub>	// M42 ← RF, Store C in M42
(06) <sub>16</sub>	(1040) <sub>16</sub>	// M40 ← R0, Load A from M40 to R0
(07) <sub>16</sub>	(1141) <sub>16</sub>	// M41 ← R1, Load B from M41 to R1
(08) <sub>16</sub>	(1242) <sub>16</sub>	// M42 ← R2, Load C from M42 to R2
(09) <sub>16</sub>	(3310) <sub>16</sub>	// R3 ← R1 + R0, Add A to B and store in R3
(0A) <sub>16</sub>	(3432) <sub>16</sub>	// R4 ← R3 + R2, Add C to (A + B) and store in R4
(0B) <sub>16</sub>	(2434) <sub>16</sub>	// M43 ← R4, Store The result in M43
(0C) <sub>16</sub>	(1F43) <sub>16</sub>	// RF ← M43, Load the result to RF
(0D) <sub>16</sub>	(2FFF) <sub>16</sub>	// MFF ← RF, Send the result to the monitor
(0E) <sub>16</sub>	(0000) <sub>16</sub>	// Halt

10. Match the following to one or more layers of the TCP/IP protocol suite. (4%)

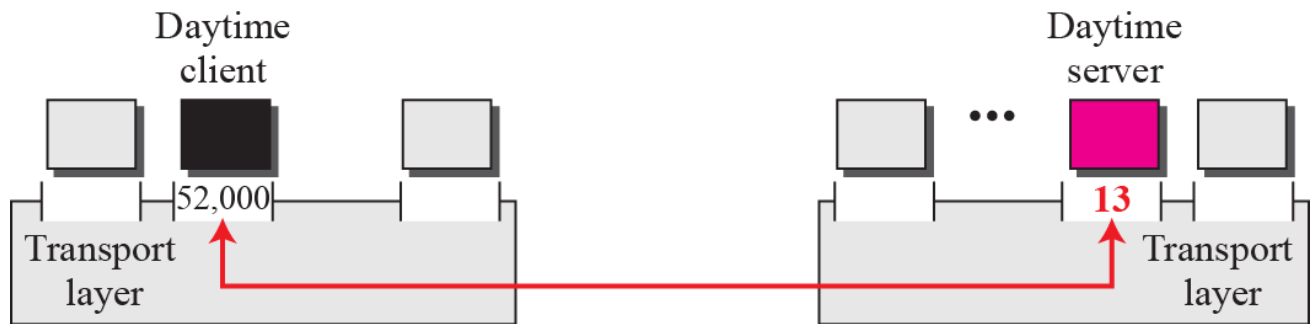
- Creating user datagrams **Transport layer**
- Deciding the routing of the packet **Network layer**
- Providing services for the end user **Application layer**
- Responsibility for handling frames between adjacent codes **Data link layer**

11. Define the protocol and describe why we need a protocol at each layer in the Internet. (3%)

**A protocol** defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.

12. Why do we use the **port numbers** for communication in transport-layer? (3%)

The client program defines itself with a port number, called the **ephemeral port Number**. The word ephemeral means *short-lived* and is used because the life of a client is normally short.



13. What is the difference between UDP and TCP in transport-layer protocols? (4%)

UDP is a connectionless, unreliable transport protocol and use a minimum of overhead.

TCP is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phase to provide a connection-oriented service.

14. What the function of DNS? What is the drawback of a central directory system for the huge Internet and how to solve this problem? (6%)

To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric address. Therefore, the Internet needs to have a dictionary system that cans a name to an address.

Since the Internet is so huge today, a central directory system cannot hold all the mapping.

In addition, if the central computer fails, the whole communication network will collapse.

Distribute the information among the many computers in the world.

The host that needs mapping can contact the closest computer holding the needed information.

15. During the weekend, Alice often needs to access files stored on her office desktop, from her home laptop. Last week, she installed a copy of the FTP server process on her desktop at her office and a copy of the FTP client process on her laptop at home. She was disappointed when she could not access her files during the weekend. What could have gone wrong? Assume the files are still integrity on her office desktop and the status of FTP system is fine. (5%)

Probably Alice turned off her desktop, which stopped the FTP server, when she left the office. A server process should be running all the time, waiting for clients to access it.