Computing 101 Introduction to Computing
Summer 2001 - Solutions to Homework Assignment #2

Department of Computing Sciences
University of Alberta

Chapter 4:

Question 1.

1a. $103_4 = 1 \cdot 4^2 + 0 \cdot 4^1 + 3 \cdot 4^0 = 16 + 0 + 3 = 19$

1c. $F1F_{16} = 15 \cdot 16^2 + 1 \cdot 16^1 + 15 \cdot 16^0 = 15 \cdot 256 + 16 + 15 = 3840 + 31 = 3871$

Question 2.

3c. $1\,1\,1\,1\,1\,1\,1 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2^7 - 1 = 128 - 1 = 127$

3d. $1\,0\,0\,0\,0\,0\,0\,0\,0\,0_2 = 1 \cdot 2^9 = 512$

Question 3.

If a computer uses 10 bits to represent an integer value, then the largest unsigned integer that can be represented is

$$1\,1\,1\,1\,1\,1\,1\,1\,1\,1_2 = 2^{10} - 1 = 1023$$

Question 4.

5c. $1\,0\,0\,0\,0\,0\,0\,0\,0\,1 = -1$

5d. $1\,0\,0\,0\,0\,0\,0\,0\,0\,0 = -0$, which is the reason we don't use the sign/magnitude form to represent signed integers, the representation of the integer 0 is not unique.

Question 5.

7a. $+9.250 = +1001.01 = +.100101 \cdot 2^4$, in binary, which is represented as

0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0

7b. $-10.125 = -1010.001 = -.1010001 \cdot 2^4$ in binary, which is represented as

1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0

7c. $-\frac{1}{32} = -.00001 = -.1 \cdot 2^{-4}$, which is represented as

1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0

Question 6.

8a. $AbC = 0\,1\,0\,0\,0\,0\,0\,1 \quad 0\,1\,1\,0\,0\,0\,0\,1 \quad 0\,1\,0\,0\,0\,0\,1\,1$

8d. $(a + b) = 0\,0\,1\,0\,1\,0\,0\,0 \quad 0\,1\,1\,0\,0\,0\,0\,1 \quad 0\,0\,1\,0\,1\,0\,1\,1 \quad 0\,1\,1\,0\,0\,0\,1\,0 \quad 0\,0\,1\,0\,1\,0\,0\,1$

Question 7.

The Truth Is Out There requires 22 characters (including spaces), or $8 \cdot 22 = 176$ bits.

Question 8. $a = 1,\ b = 0,\ c = 2$

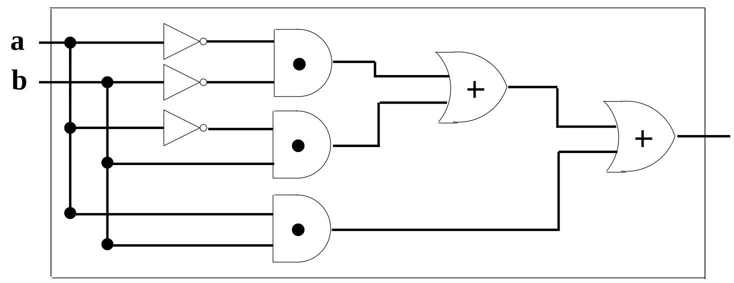
11b. $((a + b) > c)$ AND $(b \leq c)$ = **False**

11d. NOT $[(a = b)$ OR $(b = c)]$ = **True**

Question 9.

(2 marks for the expression, 2 marks for the diagram).
(It is okay to use 3-input gates to draw the circuit diagrams).

Boolean expression: $\overline{a} \cdot \overline{b} + \overline{a} \cdot b + a \cdot b$



Question 10. (2 marks for the truth table, 2 for the expressions, 1 for the diagram).
(it is okay to use 3-input gates to draw the circuit diagram).

In the truth table below, **old** represents the borrow bit from the previous column, while **new** represents the new borrow bit that propagates to the new column.

old	a	b	a - b	new
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

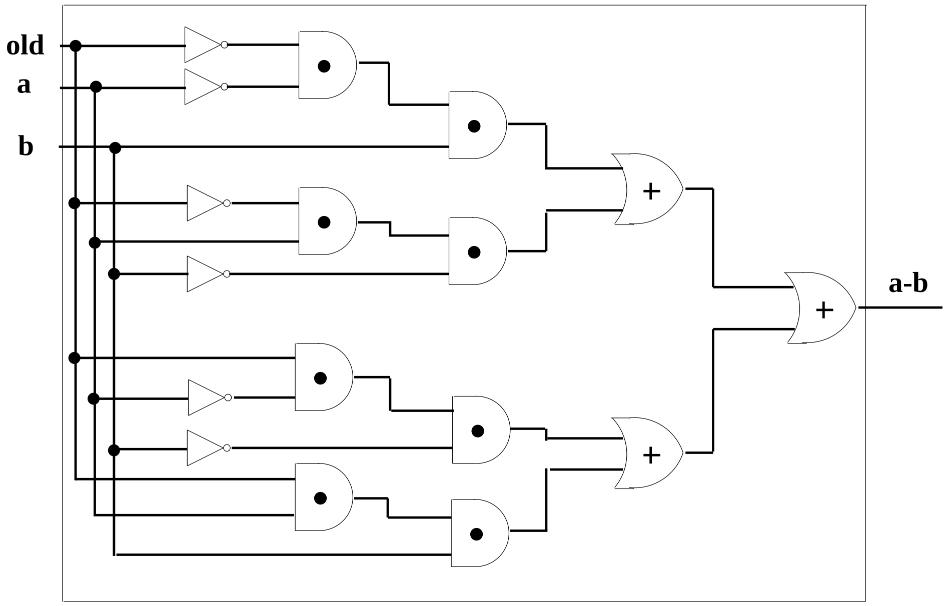
The boolean expressions are

$$\text{a-b} = \overline{\text{old}} \cdot \overline{a} \cdot b + \overline{\text{old}} \cdot a \cdot \overline{b} + \text{old} \cdot \overline{a} \cdot \overline{b} + \text{old} \cdot a \cdot b$$

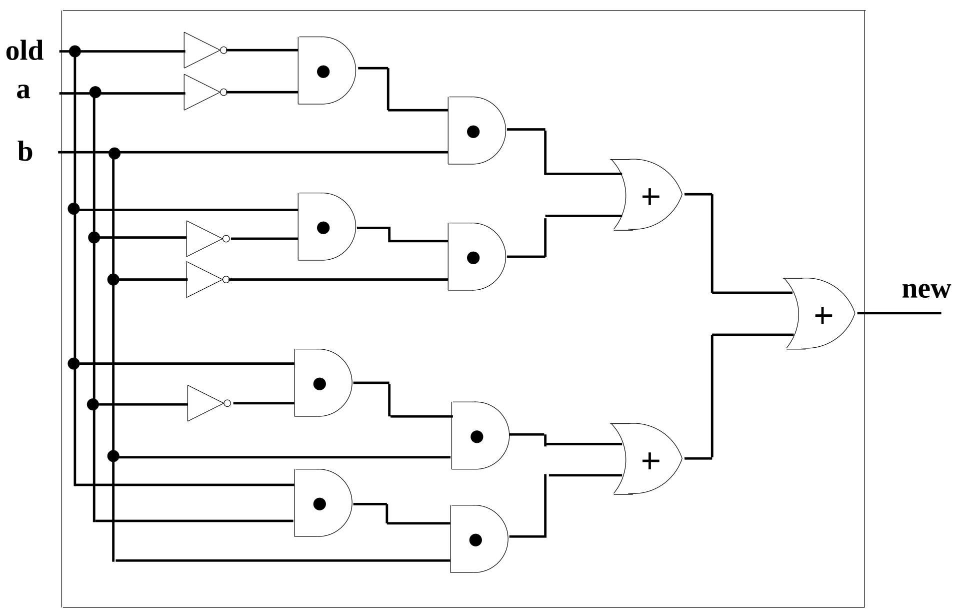
and

$$\text{new} = \overline{\text{old}} \cdot \overline{a} \cdot b + \text{old} \cdot \overline{a} \cdot \overline{b} + \text{old} \cdot \overline{a} \cdot b + \text{old} \cdot a \cdot b$$

The circuit diagram for the output $a - b$ is,



while the circuit diagram for the output new is



Question 11. A 32-input multiplexor has $2^5 = 32$ input lines and requires 5 selector lines.

Chapter 5.

Question 1.

(a) Since 1MB = 2^{20}, then 128MB would contain

$$\mathbf{2^7 \cdot 2^{20} = 2^{27} = 134,217,728}$$

memory cells.

(b) **The time it takes to carry out either a fetch or a store operation.**

(c) If the MAR has 24 bits, then the maximum memory size of the computer is $\mathbf{2^{24} = 16,777,216}$ memory cells.

(d) The MAR holds the address in memory from which a data value is to fetched, or the address in memory at which a data item is to be stored.

The MDR hold the value which is to be fetched from memory, or the value which is to be stored in memory.

Question 2.

Only information that you do not ever want to change should be stored in ROM, for example, the computer's start-up routines. This information would usually be stored in memory at the factory when the computer is manufactured.

Question 3.

(a) The number of characters that can be stored on the disk is

$$\begin{aligned}\#\text{Chars} &= 512 \text{ characters/sector} \times 120 \text{ sectors/track} \times 100 \text{ tracks/surface} \times 2 \text{ surfaces/disk} \\ &= 12,288,000 \text{ characters/disk}\end{aligned}$$

(b) The seek time in the worst case occurs when the read/write arm has to move from the first track to the last track, a total of 99 tracks, and since it moves at a rate of 0.1 msec per track crossed, the seek time in the worst case is 9.9 msec.

On the average, the read/write head moves 50 tracks, so the average seek time is $50 \times 0.1 = 5$ msec.

Each revolution takes $60/7200 = 1/120$ sec or 8.33 msec, so the latency in the worst case occurs when we have to wait a full revolution for the correct sector to arrive at the read/write head. Thus, the latency in the worst case is 8.33 msec.

In the average case, we have to wait one half a revolution, and the average latency is 4.17 msec.

The transfer time is the same in the best, worst and average cases. It takes 8.33 msec to make one complete revolution, and there are 120 sectors per track, so the transfer time is $8.33/120 = 0.07$ msec.

Time in msec.	**Best Case**	**Worst Case**	**Average Case**
Seek Time	0	9.90	5.00
Latency	0	8.33	4.17
Transfer Time	0.07	0.07	0.07
Disk Access Time	**0.07**	**18.30**	**9.24**

Question 4.

13. Since there are 20 sectors per track and 512 bytes/sector, then there are 10,240 bytes/track. Therefore, it will take **two** tracks to hold the entire file.

In order to store the file in a manner that minimizes the time it takes to retrieve the information, we should store the information in consecutive sectors of one track until the track is full, and then move to the same track on the other surface of the disk.

The read/write arm holds the read/write heads for both tracks, so there will not be any seek time, only an electronic switch from one head to the other.

Question 5.

The registers in the ALU hold the operands of upcoming arithmetic operations, or hold intermediate results in a computation.

Question 6.

(a) The instruction register holds the machine instruction that is currently being executed.

(b) During the store phase of an instruction execution the control unit sends the address IR_{addr} in the address field of the machine instruction to the MAR, and sends a store (S) signal to the fetch/store controller. The control unit then sends the value from the register R to the MDR. The memory controller then sends that value from the MDR to the address in the MAR.

(c) The control unit sends the address X to the program counter (PC).