Computer
ECE 001

Benha University

Computer Systems Engineering
Electrical Engineering Department

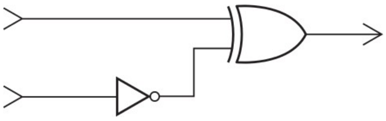
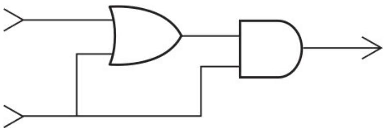
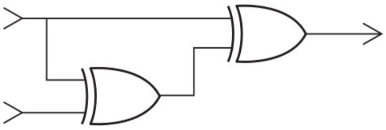
Faculty of Engineering
(at Shoubra)

Sheet 3

I Solve the following *Review Problems* from *Computer Science: An Overview*:

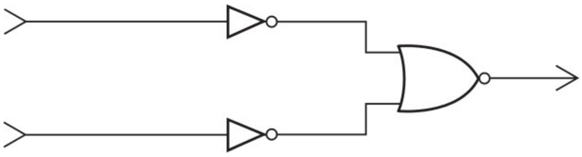
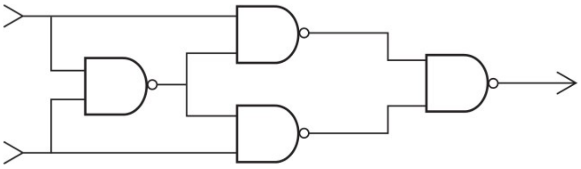
- **1.1**

  Determine the output of each of the following logical circuits

a)	b)	c)
		

- **1.2**

  What Boolean operation does each of the following logical circuits compute?

a)	b)
	

- **1.3**

  How many cells can be in a computer's main memory if each cell's address can be represented by two hexadecimal digits? What if four hexadecimal digits are used?

II Answer the following questions:

1. If your Internet connection speed is 4 Mbit/s, how long does it take to download a 1 GB movie?
2. How many bits do you need to address 12 drawers?
3. Give a definition/an example for each of the following:
   a) *Bit*
   b) *Byte*
   c) *Boolean Operation*

4. a) In what way are general-purpose registers and main memory cells similar?
   b) In what way do general-purpose registers and main memory cells differ?

5. Suppose a block of data is stored in the memory cells of the machine described in Appendix C from address 98 to A2, inclusive. How many memory cells are in this block? List their addresses.

6. What is the value of the program counter in the machine described in Appendix C immediately after executing the instruction B0CD?

7. Suppose the memory cells at addresses 00 through 05 in the machine described in Appendix C contain the following bit patterns:

Address	Contents
00	22
01	11
02	32
03	02
04	C0
05	00

Assuming that the program counter initially contains 00, record the contents of the program counter, instruction register, and memory cell at address 02 at the end of each fetch phase of the machine cycle until the machine halts.

8. Suppose three values x, y, and z are stored in a machine's memory. Describe the sequence of events (loading registers from memory, saving values in memory, and so on) that leads to the computation of x + y + z. How about (2x) + y?

9. The following are instructions written in the machine language described in Appendix C. Translate them into English.

a) 7123 b) 40E1 c) A304 d) B100 e) 2BCD

10. Suppose a machine language is designed with an op-code field of 4 bits. How many different instruction types can the language contain? What if the op-code field is increased to 6 bits?

11. Translate the following instructions from English into the machine language described in Appendix C. a) LOAD register 6 with the hexadecimal value 77.

b) LOAD register 6 with the contents of memory cell 77.
c) JUMP to the instruction at memory location 24 if the contents of register 0 equals the value in register A.
d) ROTATE register 4 three bits to the right.
e) AND the contents of registers E and 2 leaving the result in register 1.

12. Summarize the steps involved when the machine described in Appendix C performs an instruction with op-code B. Express your answer as a set of directions as though you were telling the CPU what to do.

13. Perform the indicated operations:

a)	111001	b)	000101	c)	001110	d)	111011	e)	111001	f)	010100
AND	101001	AND	101010	AND	010101	AND	110111	OR	101001	OR	101010
g)	010100	h)	101010	i)	111001	j)	000111	k)	010000	l)	111111
OR	101010	OR	110101	XOR	101010	XOR	101010	XOR	010101	XOR	110101

14. What would be the result of performing a 4-bit left circular shift on the following bit patterns?

a) 10101 b) 11110000 c) 001 d) 101000 e) 00001

Appendix C

The machine has 16 general-purpose registers numbered 0 through F (in hexadecimal). Each register is one byte (eight bits) long. For identifying registers within instructions, each register is assigned the unique four-bit pattern that represents its register number. Thus register 0 is identified by 0000 (hexadecimal 0), and register 4 is identified by 0100 (hexadecimal 4).

There are 256 cells in the machine's main memory. Each cell is assigned a unique address consisting of an integer in the range of 0 to 255. An address can therefore be represented by a pattern of eight bits ranging from 00000000 to 11111111 (or a hexadecimal value in the range of 00 to FF).

The Machine's Language

Each machine instruction is two bytes long. The first 4 bits provide the op-code; the last 12 bits make up the operand field. The table that follows lists the instructions in hexadecimal notation together with a short description of each. The letters R, S, and T are used in place of hexadecimal digits in those fields representing a register identifier that varies depending on the particular application of the instruction. The letters X and Y are used in lieu of hexadecimal digits in variable fields not representing a register.

Op-code	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY. Example: 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.
2	RXY	LOAD the register R with the bit pattern XY. Example: 20A3 would cause the value A3 to be placed in register 0.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY. Example: 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.
4	0RS	MOVE the bit pattern found in register R to register S. Example: 40A4 would cause the contents of register A to be copied into register 4.
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R. Example: 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating-point result in register R. Example: 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.
7	RST	OR the bit patterns in registers S and T and place the result in register R. Example: 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.
8	RST	AND the bit patterns in registers S and T and place the result in register R. Example: 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R. Example: 95F3 would cause the result of EXCLUSIVE ORing the contents of
A	R0X	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end. Example: A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
B	RXY	JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. (The jump is implemented by copying XY into the program counter during the execute phase.) Example: B43C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the pattern 3C would be placed in the program counter so that the next instruction executed would be the one located at that memory address. Otherwise, nothing would be done and program execution would continue in its normal sequence.
C	000	HALT execution. Example: C000 would cause program execution to stop.