

Midterm Exam

600.233 Computer Systems Fundamentals

Fall 2016

Johns Hopkins University

Instructor: Prof. Philipp Koehn

14 October 2016

Complete all questions.

Use additional paper if needed.

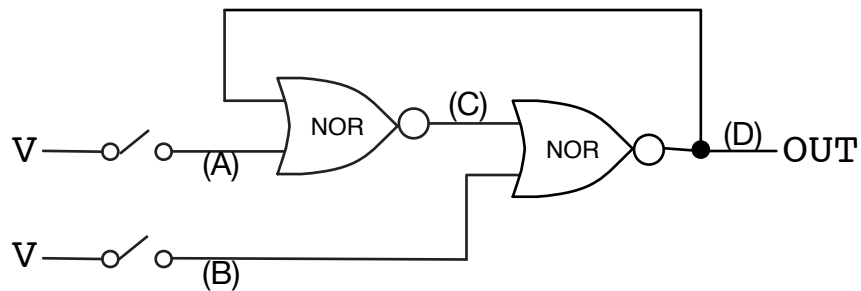
Time: 50 minutes.

Name of student:

Q1. Analysis of Circuit

25 points

Consider the following circuit:



Assume that at the onset, the wires are activated as follows: $A=0$, $B=0$, $C=1$, $D=0$.

Moreover, assume that it takes 1ms for a NOR gate to change its output value.

The following actions are performed:

- At time 10ms, the key connected to wire (A) is closed.
- At time 20ms, the key connected to wire (A) is opened.
- At time 30ms, the key connected to wire (B) is closed.
- At time 40ms, the key connected to wire (B) is opened.

Trace the activation levels for each wire (A)–(D) at each time step when wire activations change values.

Q2. Design a Circuit

25 points

You are given the following truth table for a function.

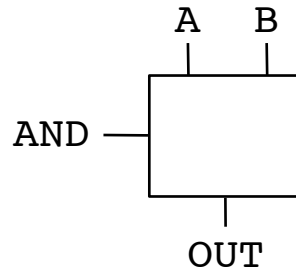
A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Design a circuit with AND, OR, and NOT gates that implements this function (hint: use CNF or DNF).

Q3. Boolean Logic Unit

25 points

Design the following component.



Inputs are A and B.

Flag AND indicates the operation.

- If the AND input flag is 1, then the OUT value is A AND B.
- If the AND input flag is 0, then the OUT value is A OR B.

Design this component with AND, OR, and NOT gates.

Q4. SCRAM

25 points

Consider the following code of a SCRAM-like program.

Address	Operation	Data
0	PRINT	11
1	PRINT	12
2	LDA	11
3	ADD	12
4	PRINT	AC
5	STA	13
6	LDA	12
7	STA	11
8	LDA	13
9	STA	12
10	JMP	2
11	DAT	0
12	DAT	1
13	DAT	0

Operations:

- STA: store accumulator
- LDA: load accumulator
- ADD: add to accumulator
- JMP: jump to memory address
- DAT: dummy instruction (data value used only)
- PRINT: print accumulator or memory value

Note that the program runs in an infinite loop.

What are the first 6 numbers printed by the program?

Show your work (for instance by reporting values in relevant memory locations).

