

(a) Convert the 6-bit 2-complement binary number 101010 to a decimal number. First give your final answer (the decimal number):

Briefly explain your answer.

Word count: 0, character count: 0

(b) Convert the hexadecimal number ABC to binary; first give the final answer:

Briefly explain your answer.

Word count: 0, character count: 0

(c) Convert the hexadecimal number 208 to decimal; first give the final answer:

Briefly explain your answer.

Word count: 0, character count: 0

(d) Suppose we make a new number system for 4-digit binary numbers, by giving the bits, from left to right, the following weights: 8, -4, -2, -1. What can we say about this?

- This is equivalent to the normal 2-complement notation.
- When a number is written in binary according to this system and then interpreted as if it is a normal 2-complement number, only the plus/minus sign is flipped.
- This is equivalent to the normal 1-complement system.
- This is useless because there cannot be multiple negative weights.
- When a number is written in binary according to this system and then interpreted as if it is a normal 2-complement number, the plus/minus sign is flipped, and the result is off by 1.

(e) Suppose we take a negative, 8-bit, 2-complement binary number, and shift all bits one position to the left. The left-most bit (which falls out) shifted in at the right, to fill the empty spot there. (This is also called 'rotating' the bits.) Is this a correct way to double the number? (You may assume the number that we started out with, is such that also its double can still be represented in the 8 bits 2-complement notation.)

- Yes it is.
- No, the resulting number will be 1 too high.
- No, the resulting number will be 1 too low.
- No, this will halve rather than double the number.
- No, it will make the number positive since the sign bit is moved.
- Shift to the left by 1 position, drop the left-most bit, and insert the dropped left-most bit at the right.

☆ Flag this question

🧮 Calculator

➔ Save and continue

(a) Give the truth table of a 3-input "uniqueness check": the output D indicates (by being 1) whether exactly one of the inputs is 1. (You may not need all rows in the table; leave any unused rows blank.)

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

The editable cells are marked in light yellow.

(b) In the left-most column of the below table, we give a derivation in Boolean algebra, starting with $\overline{AB} + \overline{BC} + A(B + C)$, making one step per row.

Indicate in each row, which rule of Boolean algebra is used to make that step. Choose "wrong" if you think that that step is not correct. (It is possible that a rule is used multiple times, or not at all, in this derivation; however, each step uses only a single rule.)

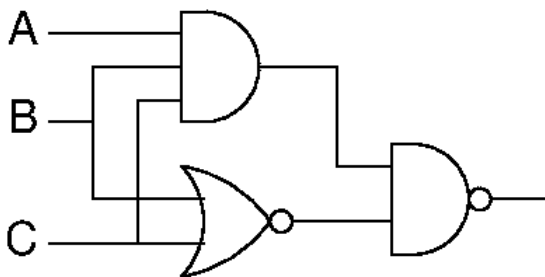
To clarify this, in the first row you should indicate which Boolean algebra rule is used to demonstrate that the formula on that row, namely $\overline{AB} + \overline{BC} + AB + AC$, follows from our starting formula, $\overline{AB} + \overline{BC} + A(B + C)$. The second row should indicate how the formula on the second row follows from the formula on the first row, and so on.

(Note: the table is quite wide; depending on the screen settings, you may need to use the scrollbar to reach the right-most columns.)

	complement	Idempotence	commutative	distributive	associative	identity	w
$= \overline{AB} + \overline{BC} + AB + AC$	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
$= \overline{AB} + \overline{BC} + ABC$	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
$= \overline{ABBC} + ABC$	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
$= \overline{ABC} + ABC$	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
$= 1$	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

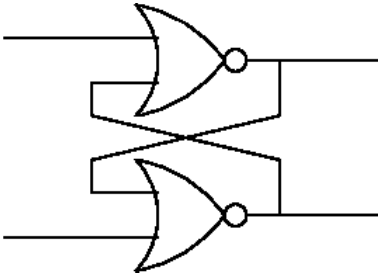
(c) Describe the operation of the following logic circuit as a formula in Boolean logic.

Note: Boolean formulas often have a bar above part of the equation. If this is the case in your formula, you can either use the built-in equation editor to type this (accessible via the Σ button), or simply type something like not(ABC) or !(ABC), as long as it's clear what you mean.



Rich text editor toolbar with options for bold, italic, underline, subscript, superscript, bulleted list, numbered list, indent, outdent, link, unlink, undo, redo, table, table border, text color, background color, link, unlink, insert link, insert image, and a summation symbol (Σ).

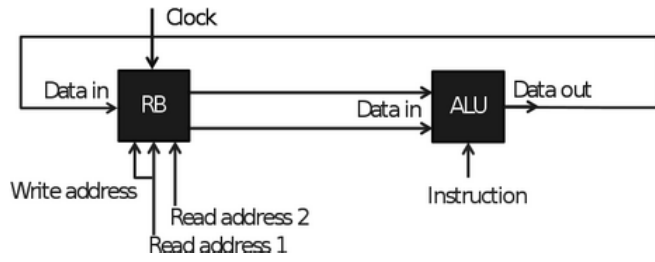
(d) Consider the following circuit. Note that it is exactly the same as the simple flip-flop circuit with two NAND gates that we saw in the tutorial and the lecture, except that the NAND gates have been replaced by NOR gates. What does this circuit do?



- It's a flipflop that remembers which of the inputs was most recently '1', rather than most recently '0' as in the original NAND flip-flop.
- It behaves exactly like the original flip-flop built of NAND gates.
- It has two stable states like a regular flip-flop, but the input signals cannot flip it to the other state.
- It is an unstable circuit: the output will constantly alternate between 0 and 1 as fast as the physical hardware allows, and the inputs have no influence on this.

☆ Flag this question

→ Save and continue



The ALU of the processor above has two instructions: 0 = 'ADD' and 1 = 'MUL'.
 The register bank (RB) contains four 8-bit registers, numbered from 0 to 3 (decimal).
 Initially R0, R1 and R2 contain some numbers x , y and z , respectively.
 Give for this processor the program for computing $x^2y + y^2z$ and storing the result in R1.
 (You may not need all timeslots.)

	A	B	C	D
1		read address 1 / write address	read address 2	instruction
2	Timeslot 0			
3	Timeslot 1			
4	Timeslot 2			
5	Timeslot 3			
6	Timeslot 4			

The editable cells are marked in light yellow.

☆ Flag this question

→ Save and continue

Consider the following program for the Arduino's 8 bit 'AVR' processor; the numbers at the left are memory address at which that instruction is stored:

```

0000 LDI R17, $D8
0001 LDI R18, $08
0002 LDI R19, $10
0003 ADD R17, R18
0004 ADD R17, R19
0005 SUB R19, R18
0006 BRNE -3
0007 ADD R17, R18
0008 SUBI R18,$08
0009 BREQ -4
    
```

Write in the below table the contents of the program counter, the registers and the Z(ero) flag after each instruction, one instruction per line. If a register doesn't change from one line to the next, you may leave it blank. If the instruction is a jump or branch, use the "branch/comment" column to write down whether a jump is performed, and if so, to where. You may not need all the lines in the table. (Note that you have a copy of the AVR instruction set document available in another browser tab, in case some of the instructions are unknown to you. In case you accidentally closed that tab, you can reopen it using this link: <https://www.utwente.nl/en/ces/sal/exams/intranet/books/Hardware-Diamond/>)

A	B	C	D	E	F	G
1	Program counter	R17	R18	R19	Z	branch/comment
1						
2	2					
3	3					
4	4					
5	5					
6	6					
7	7					
8	8					
9	9					
10	10					
11	11					
12	12					
13	13					
14	14					
15	15					
16	16					
17	17					
18	18					
19	19					
20	20					
21						

The editable cells are marked in light yellow.

How many clock cycles does the program take?

First give the numerical answer:

Explain your answer:

Word count: 0, character count: 0

☆ Flag this question

🧮 Calculator

➔ Save and continue

Question 5

15 points

What is the mathematical function that is computed by the code below?
Assume that X is positive; the final result is in R19.

(Recall that you have the AVR instruction set document available for reference in another browser tab.)

```
LDI R17, $X
LDI R18, $01
LDI R19, $FF
repeat:
SUB R17, R18
BRCS finish
ADD R17, R18
ADD R18, R18
INC R19
RJMP repeat
finish:
```

Write as a function of X:

Explain your answer:

Word count: 0, character count: 0

☆ Flag this question

→ Save and continue