CS61C Spring 2026

Boolean Algebra, SDS

Discussion 7

1 Boolean Logic

1.1 Simplify the following Boolean expressions:

(a) $(A + B)\left(A + \overline{B}\right)C$

(b) $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,C + A\,\overline{B}\,C$

(c) $\overline{A\left(\overline{B}\,\overline{C} + BC\right)}$

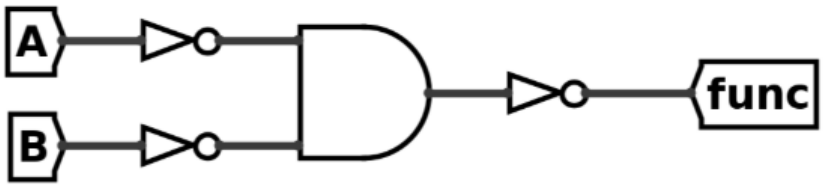
(d) $\overline{A}(A + B) + (B + AA)\left(A + \overline{B}\right)$

2 Digital Logic Simplification

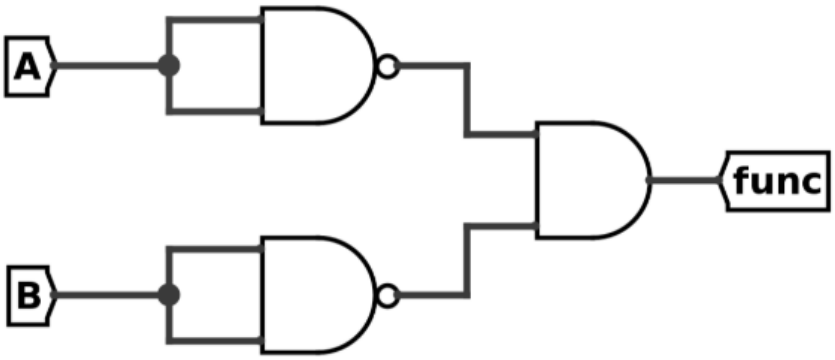
For the following digital logic circuits:

1. Write a boolean algebra expression that corresponds the physical circuit.
2. Simplify the expression and draw the simplified circuit.

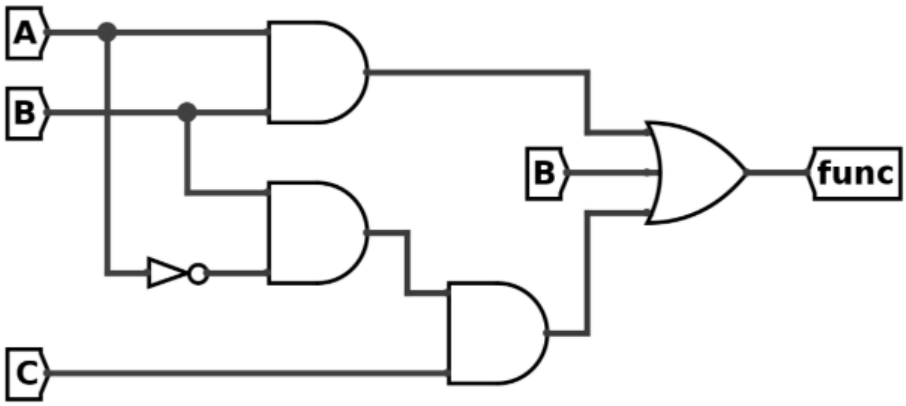
2.1



2.2



2.3



2.4 Why might it be useful to simplify logic circuits?

3 Combinational Logic from Truth Tables

For this question, we have a single 3-bit input and a single 4-bit output. We want to design a combinational logic circuit to achieve the desired output given the appropriate combinations of input bits (`Input=001` $\implies$ `Output=0011`, and so on...). Here is the truth table we wish to implement:

Input	Out
000	0001
001	0011
010	1111
011-111	xxxx

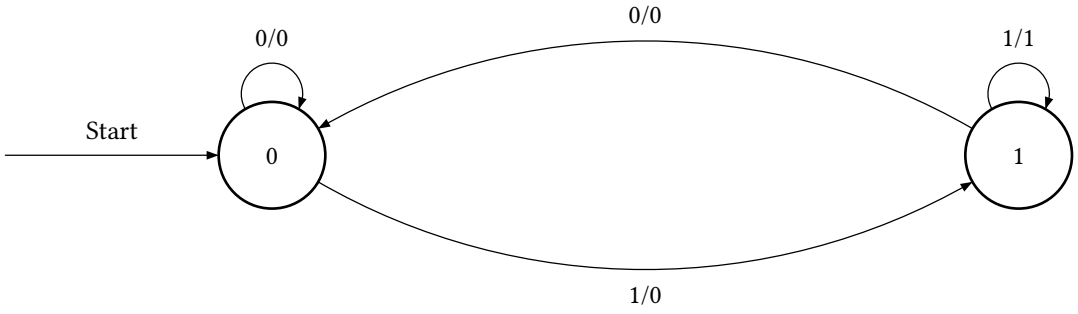
The `x`'s for the final entry of the table indicate that any output is valid for the case that `Input` is `011`, `100`, `101`, `110`, and `111`

3.1 Write out and simplify boolean expressions for each of the output bits `Out[3], Out[2], Out[1],` and `Out[0]` in terms of the input bits `In[2], In[1], In[0]`.

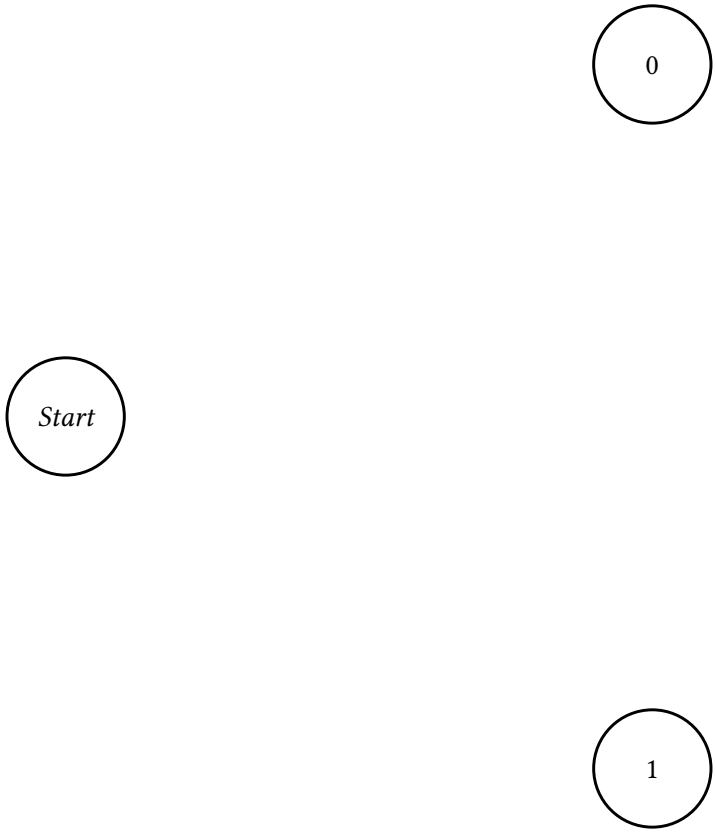
3.2 Draw out the boolean circuit based on your simplified expressions above. You may use constants 0 and 1, and the logic gates AND, OR, NOT.

4 FSM

4.1 What pattern in a bitstring does the FSM below detect? What would it output for the input bitstring `011001001110`?



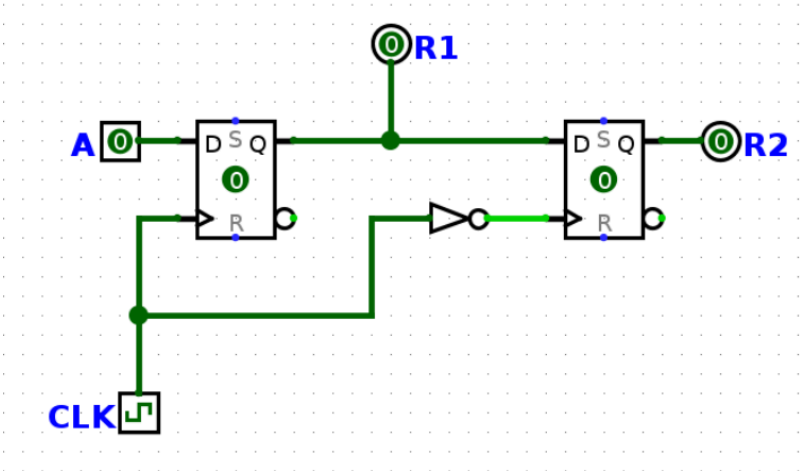
4.2 Fill in the following FSM for outputting a `1` whenever we have two repeating bits as the most recent bits, and a `0` otherwise. You may not need all states.

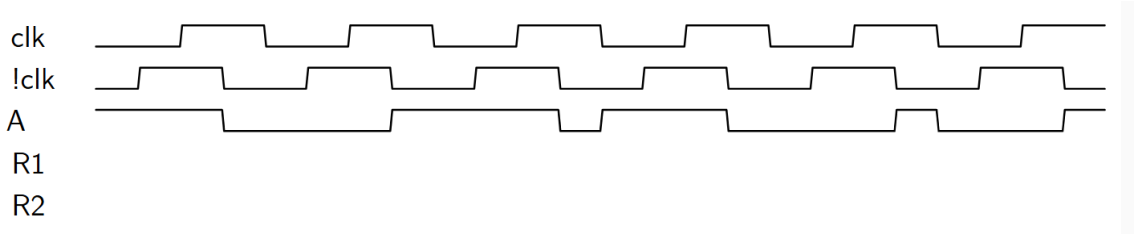


4.3 Draw an FSM that will output a `1` if it recognizes the regex pattern `{10+1}`. That is, if the input forms a pattern of a `1`, followed by one or more `0`s, followed by a `1`.

5 SDS Intro

5.1 Fill out the timing diagram. The clock period (rising edge to rising edge) is 8ps. For every register, clk-to-q delay is 2ps, setup time is 4ps, and hold time is 2ps. NOT gates have a 2ps propagation delay, which is already accounted for in the !clk signal given.





5.2 In the circuit below:

- RegA and RegB have setup, hold, and clk-to-q times of 4ns,
- All logic gates have a delay of 5ns
- RegC has a setup time of 6ns.

What is the maximum allowable hold time for RegC? What is the minimum acceptable clock cycle time for this circuit, and clock frequency does it correspond to?