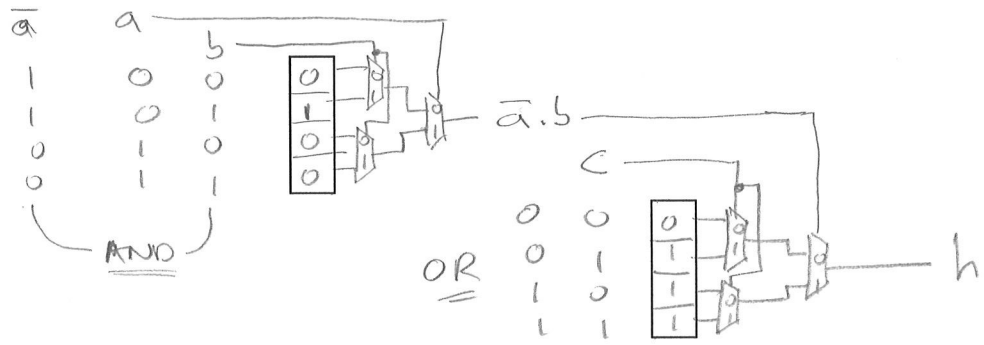
LAST NAME, First name SOLUTIONS ID#_____

Queen's University
Department of Electrical and Computer Engineering
ELEC 271 Digital Systems (Fall 2019)

/ 50

Practice Quiz – Reflects Planned Changes in Format/Style

1. Implement $h(a,b,c) = \overline{a} \cdot b + c$ for FPGA technology with 2-input LUTs. Show lookup tables *in detail*, including 0/1 multiplexer labels and table contents.

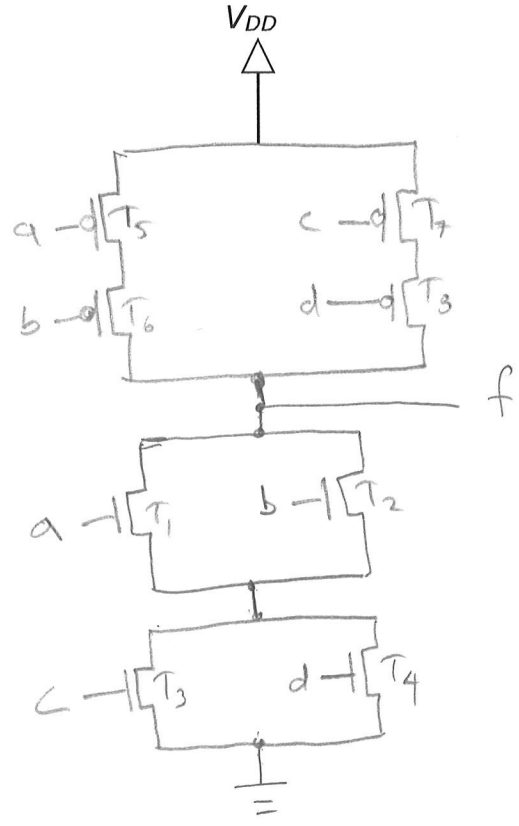


2. Write function h above as a VHDL signal assignment statement.

```
h <= (not a and b) or c;
```

3. For the function $f(a,b,c,d) = \overline{(a+b)(c+d)}$, implement a custom CMOS circuit for this function in the space provided. Label your transistors T_1, T_2, etc., *starting first with the NMOS transistors*. In the table below, consider just the *two* valuations of high/low voltages. For each valuation, fill in the table with the *on/off* state of each transistor and the resulting output voltage.

inputs a b c d	transistors T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8	output f
L L H H	off off on on (on) (on) off off	H
H L H L	(on) off (on) off off on off on	L



▷ Low inputs for T_5/T_6 create a path to V_{DD}

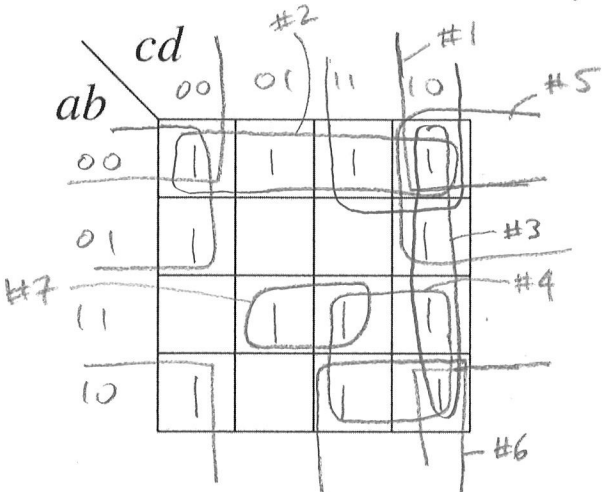
$\overline{(0+0)\cdot(1+1)} = \overline{0 \cdot 1} = \overline{0} = 1$

▷ High inputs for T_1/T_3 create a path to ground

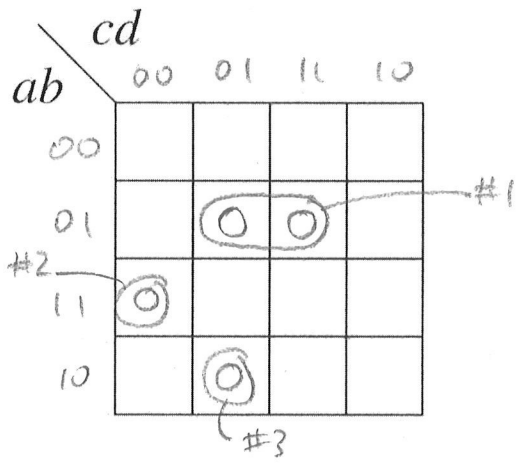
$\overline{(1+0)(1+0)} = \overline{1 \cdot 1} = \overline{1} = 0$

4. For $g(a,b,c,d) = \Sigma(m_0, m_1, m_2, m_3, m_4, m_6, m_8, m_{10}, m_{11}, m_{13}, m_{14}, m_{15})$, first write the equivalent product-of-sums specification (for your benefit). Then, generate two separate Karnaugh maps and use them to derive the lowest-cost expressions in *sum-of-products* and *product-of-sums* forms. Identify ***all*** of the prime implicants to ensure an optimal algebraic result.

$g(a,b,c,d) = \Pi$ (M_5, M_7, M_9, M_{12})



Prime implicants	Essential?
1. $\bar{b}\bar{d}$	Y
2. $\bar{a}\bar{b}$	Y
3. $c\bar{d}$	N
4. ac	N
5. $\bar{a}\bar{d}$	Y
6. $\bar{b}c$	N
7. abd	Y



Prime implicants	Essential?
1. $(a+\bar{b}+\bar{d})$	Y
2. $(\bar{a}+\bar{b}+c+d)$	Y
3. $(\bar{a}+b+c+\bar{d})$	Y

non-essential #4 (ac) covers the two remaining minterms (m_{11}, m_{14})

essential

Least-cost expressions in:

sum-of-products form $\bar{b}\bar{d} + \bar{a}\bar{b} + \bar{a}\bar{d} + abd + ac$

product-of-sums form $(a+\bar{b}+\bar{d}) \cdot (\bar{a}+\bar{b}+c+d) \cdot (\bar{a}+b+c+\bar{d})$

5. Briefly state the purpose of FPGA pin assignment/configuration.

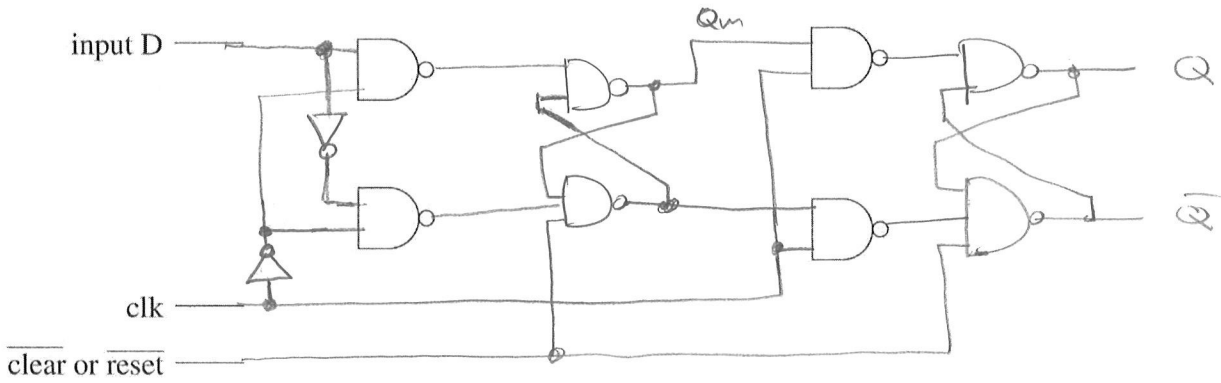
reflects the specific physical connections between FPGA chip and external components on printed circuit board to match design to hardware

6. Briefly describe the origin of the chip planner view in FPGA CAD software.

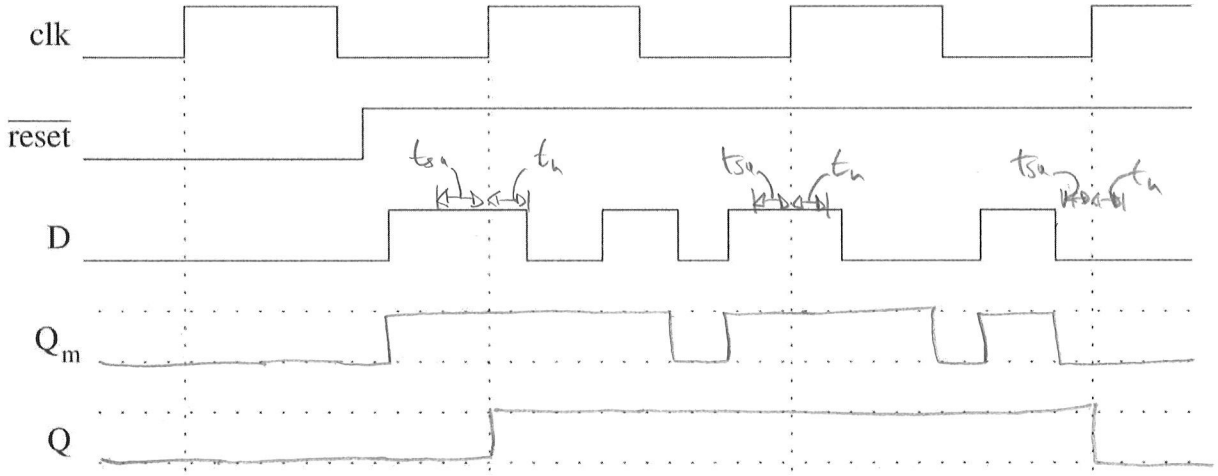
early CAD software required human to indicate which logic elements and which routing paths to use through graphical chip planner view

[after subsequent automation of these tasks, graphical view was retained to enable humans to understand and possibly adjust software decisions]

7. Draw the complete circuit for a ***positive***-edge-triggered D flip-flop.



8. For a *positive*-edge-trig. D flip-flop, complete the diagram, marking t_{su} & t_h.



9. Write VHDL process code for a *positive*-edge-triggered D flip-flop with *load-enable*, active-low *asynch. clear*, & active-low *asynch. preset* inputs.

```
the_dff: process (clk, reset_n, preset_n)
begin
    if (reset_n = '0') then
        q <= '0';
    elsif (preset_n = '0') then
        q <= '1';
    elsif (clk'event and clk = '1') then
        if (le = '1') then
            q <= d;
        end if;
    end if;
end process;
```