Roll No.: _____

CS226 Quiz 1 (Spring 2018)

Max marks: 30 **Time: 90 mins**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- *If you need to make any assumptions, state them clearly.*
- ***IIT Bombay prohibits the use of communication devices and internet enabled devices during examinations. You will be debarred from taking the examination if you are found accessing the internet during the examination.***
- ***Please do not engage in unfair or dishonest practices during the examination. Anybody found indulging in such practices will be referred to the D-ADAC.***

1. *[10 marks]*A designer wishes to construct an AIG for a Boolean function f of three variables a, b, c. She constructs the required AIG by constructing AIGs for different sub-functions and combining them using appropriate operations ***in the following sequence***. Thus, t_1 is constructed first, then t_2 is constructed and so on.

   (a) $t_1 = a \cdot \overline{b}$

   (b) $t_2 = a + \overline{c}$

   (c) $t_3 = a \oplus b$

   (d) $t_4 = t_1 \oplus \overline{t_2}$

   (e) $t_5 = compose(t_3, b, t_1)$

   Show the final strash-ed AIG obtained by the designer. Your AIG must be structurally hashed (or strash-ed) and must have only one root corresponding to t_5. All AIG nodes that do not have any path to the root after strashing must be removed from the final AIG.

2. An ex-employee of *DontCares Inc.* designed the circuit shown in Fig. 1 fifteen years back. You, as the current lead designer of *DontCares Inc.*, have been tasked with the responsibility to figure out if the circuit can be simplified any further by making use of don't cares.

   Unfortunately, the original specification document from which the circuit in Fig. 1 was designed is no longer available. The only document available about the circuit says that that any implementation of the following 3-variable K-map can be used in place of the shaded logic block to meet the overall (input-output) specification, ***as long as the same implementation is used for all instances of the shaded logic block.*** Thus if the implementation for one instance of the shaded block uses 1 for a don't care entry, the implementation for the other instance of the shaded block must also use 1 for the same don't care entry.

   K-map for f_1

a,b→ / c↓	**00**	**01**	**11**	**10**
0	1	-	1	0
1	1	0	1	0

   It turns out that by analyzing the don't care combinations in the specification of the shaded block, we can derive some external don't cares for the overall circuit.

   (a) *[10 marks]* Give a Boolean function of $x_1, \ldots x_5$ that represents the set of external don't cares of the overall circuit. Express your Boolean function in terms of and $(\cdot)$, or $(+)$ and not () operators.

   (b) *[5 marks]* Give a Boolean function representing the observability don't cares of the first shaded block (block 1) in the context of the overall circuit. Express your Boolean function in terms of and $(\cdot)$, or $(+)$ and not () operators, and in terms of x_3, x_4, x_5.

   (c) *[5 marks]* Give a Boolean function representing the entire set of don't cares of the second shaded block (block 2) in the context of the overall circuit Express your Boolean function in terms of and $(\cdot)$, or $(+)$ and not () operators, and in terms of x_1, t_1, t_2.

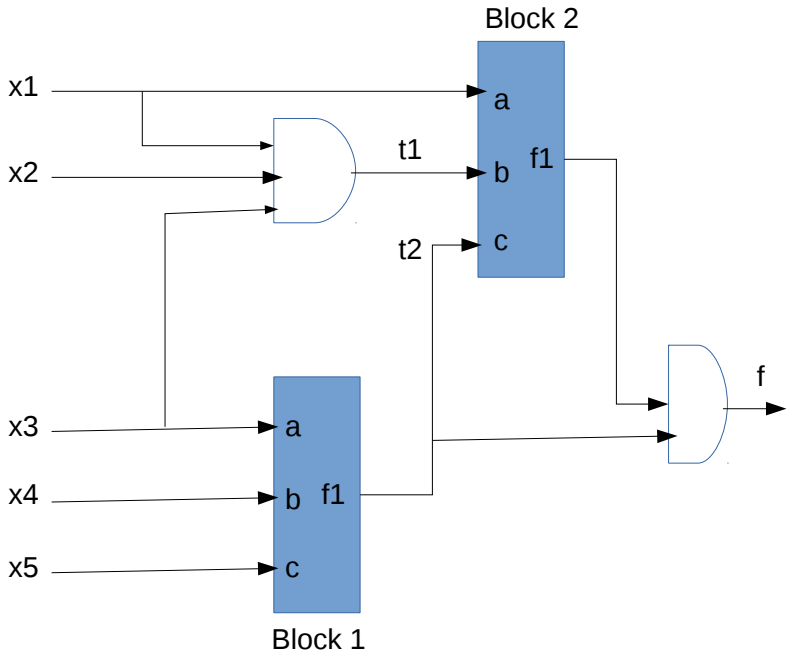

Figure 1: Circuit with two instantiations of logic block