CS226 Quiz 2 (Spring 2017)

Max marks: 45 **Time: 60 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.*
- ***Please start writing your answer to each sub-question on a fresh page. DO NOT write answers to multiple sub-questions on the same page.***
- ***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. *[3 × 5 marks]*

   Consider the following K-maps, where a "-" means that the value in the square is a don't-care, i.e. you are free to choose 0 or 1.

   Function F:

$\frac{X,Y\rightarrow}{Z,W\downarrow}$	**00**	**01**	**11**	**10**
00	1	0	-	1
01	0	-	0	1
11	-	1	0	-
10	1	0	-	1

   Function G:

$\frac{X,Y\rightarrow}{Z,W\downarrow}$	**00**	**01**	**11**	**10**
00	1	0	0	0
01	0	-	-	0
11	0	0	-	1
10	0	-	1	1

   Give K-maps for each of the following functions, ensuring that your K-map in each case has the maximum number of don't cares (i.e. do not fill any square with 0 or 1 if it can be filled with a "-").

   (a) $G \oplus F$, where $\oplus$ signifies exclusive-or (exor).

   (b) $\mathsf{ite}(G, F_X, F)$, where F_X denotes the co-factor of F with respect to X.

   (c) $\mathsf{compose}(F, Y, G)$

2. *[15 marks]*

   In implementing a certain data-processing algorithm with a given datapath, a student has come up with the following controller table (similar to the tables we have seen in class when designing the controller for an integer divider).

CurrState	I1	I2	I3	NextState	O1	O2
S_1	1	-	-	S_2	0	-
S_2	0	1	-	S_2	1	1
S_2	0	0	1	S_3	-	-
S_3	-	-	0	S_4	0	1
S_4	1	-	1	S_5	-	1
S_5	0	0	-	S_2	0	1

   Note that the controller has 5 states, viz. S_1 through S_5, and three binary inputs, viz. I1, I2, I3. It generates two control outputs for the datapath, viz. O1 and O2. There are several entries in the table that are filled with "-". These represent don't cares.

   To illustrate how this table is to be read, let's look at the first row of the table. This row states that when the current state is S_1 and $I1$ is 1, then regardless of the values of the inputs I2 and I3, the next state becomes S_2, output O1 is set to 0 and the value of the output O2 doesn't matter. The interpretation for the other rows of the table are similar. The behaviour of the controller when the inputs and current state are not as listed in the above

table are irrelevant, and may be used with discretion to optimize the digital logic functions for the next state and output bits.

The student wishes to use three state variables x, y and z for encoding the states. Specifically, she wishes to use the encoding $S_1(xyz = 001)$, $S_2(xyz = 010)$, $S_3(xyz = 011)$, $S_4(xyz = 100)$ and $S_5(xyz = 101)$. Note that this means the controller has three current state variables, x, y and z, and three corresponding next state variables, say X, Y and Z.

Clearly, each of X, Y, Z, $O1$ and $O2$ can be expressed as Boolean functions of $x, y, z, I1, I2$ and $I3$. You are required to give as small ROBDDs as you can for each of the next state variables, X, Y, Z, and for $O1$ and $O2$ using the variable ordering $x < y < z < I1 < I2 < I3$. *None of your ROBDDs should have more than 10 nodes, including the leaf nodes.*

Instead of jumping straight ahead to draw ROBDDs, please take a few minutes to unearth the significant optimization opportunities available from the unspecified parts of the controller table, and from the don't cares in the table.

3. *[3 × 5 marks]* Consider the Boolean functions $f(x, y, z, w) = \overline{(x.y.\overline{z} + z.w + \overline{z}.\overline{w})}$ and $g(y, z) = z.(w + y)$.

   Using the variable order $x < y < z < w$, construct ROBDDs for the following:

   (a) f

   (b) g

   (c) $\mathsf{Simplify}(f, g)$, i.e. f simplified by considering g as the **care set**. You must indicate any assumptions regarding simplification that you are making.