CS226 Practice Problem Set 3 (Spring 2016)

Date posted: Feb 23, 2017 **Expected Solving Time: 2 hours**

- *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.*
- ***These are ungraded practice questions. You are strongly encouraged to solve these on your own to ensure you understand the material being taught in class.***
- ***Mutual discussion is allowed, but copying is not. Please read the guidelines on the course webpage if you don't understand the distinction between the two.***

1. You are given three combinational circuits, each with 6 Boolean inputs $x_1, \ldots x_6$, as shown in Fig. 1.

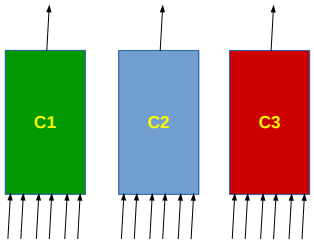

Figure 1: Three faulty circuits

You are told that each of the three circuits is a faulty implementation of an unknown Boolean function $F(x_1, \ldots x_{10})$. Furthermore, you are told that

- Circuit C_1 implements F correctly iff either exactly 1 or all 3 of x_1, x_2 and x_3 are 1.
- Circuit C_2 implements F correctly iff either none or exactly 2 of x_2, x_3 and x_4 are 1, and
- Circuit C_3 implements F correctly iff either exactly 1 or all 3 of x_4, x_5 and x_6 are 1.

- Design a circuit using at most 5 XOR gates, 3 two-to-one multiplexers and C_1, C_2, C_3 that implements the unknown function F as accurately as possible. In other words, ensure that the output of your circuit matches that of F for as many input combinations as possible. You must show the block diagram of your design.

- What is the number of input combinations for which it is not possible to get the correct value of F using your design?

2. We have seen in class how to convert an ROBDD for a Boolean function F into a circuit realizing F using 2-to-1 multiplexers. Note that a multiplexer is a specific instance of a 3-input, 1 output circuit implementing the Boolean function $f(c, i_0, i_1) = c.i_1 + \overline{c}.i_0$. In this question, we wish to ask what Boolean function, say G, would be realized if a designer took an ROBDD for F and replaced each node with a 3-input, 1-output circuit element that she mistakenly thought was implementing a multiplexer. Specifically, suppose the Boolean function implemented by the circuit element used by the designer is $g(c, i_0, i_1) = c.\overline{i_1} + \overline{c}.\overline{i_0}$.

   Given the ROBDD for F shown in Fig. 2, construct the ROBDD for G that would be implemented if the designer mistakenly uses the circuit element for $g(c, i_0, i_1)$ instead of that for a multiplexer (i.e. $f(c, i_0, i_1)$). You may use the same variable ordering for the ROBDD for G as that for F.

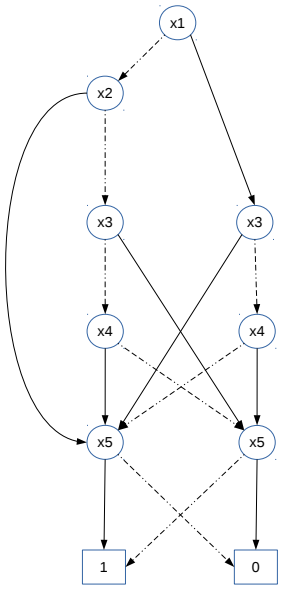

Figure 2: Example ROBDD

3. You are the chief digital logic designer in your company, and a client wants you to implement a Boolean function $F(x_1, x_2, \ldots x_{10})$ such that all of the following properties hold:

   - $F_{x_i} = \overline{F_{\overline{x_i}}}$ for all $i \in \{1, \ldots 3\}$.
   - $F_{x_i} = F_{\overline{x_i}}$ for all $i \in \{4, \ldots 6\}$.
   - $F_{x_i} \rightarrow F_{\overline{x_i}}$ for all $i \in \{7, \ldots 10\}$.

   Indicate whether such a function F exists. If your answer is in the affirmative, indicate with justification if F is uniquely defined by the above properties. If your answer is in the negative, indicate with justification why you think such a function cannot exist.

4. We want to implement a data-processing algorithm, written in a C-like language using the datapath shown below (Fig. 3). Note that this is the same datapath used in Quiz 1.

```
T = 0; Q = 0; R = 0; A = 0; B = 0;
read A and B;
while (A >= B) {
  temp = FindSmallestIndex(A, B);
  A = A - B * 2^temp;
  temp = FindSmallestIndex(A, B);
  A = A - B * 2^temp;
  temp = FindSmallestIndex(A, B);
  T = T + 2^temp;
}
R = T+2^temp; output R;
```

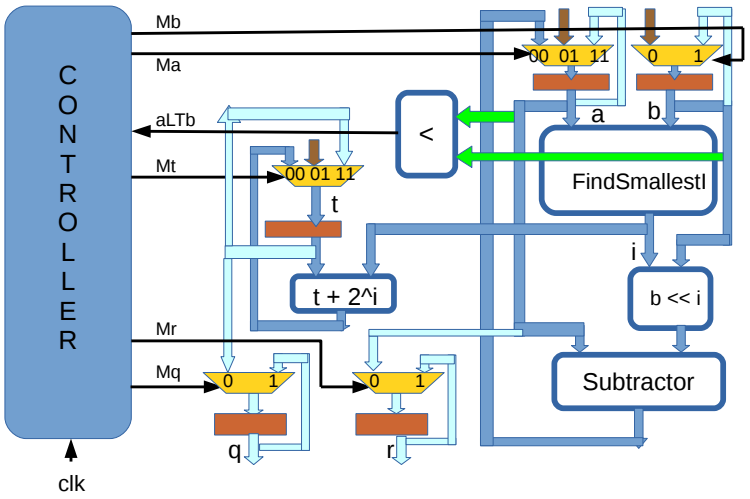

Figure 3: Datapath studied in class

For the controller, you must give the state transition table in the format given below. You may assume that the values of A and B do not change until the entire computation is over.

You MUST indicate through brief comments what each row of the controller table achieves, e.g. resets registers, or updates T with such and such expression, etc.

CurrState	$aLTB$	NextState	M_a	M_b	M_t	M_q	M_r	Reset	Comment
...	...	...	...	...	...	...	...	...	...
⋮									