Assigned: *03/09/2018* **Due:** *03/29/2018* **Total Points:** *200*

1. (c) For ease of readability, let $'x'$ denote grant0 and $'y'$ denote grant1.
Also, let rx denote req0 and ry denote grant1.
State Variables: $V = \{x, y\}$
Let $V' = \{x', y'\}$ denote the next state
Initial State: $I(V) = \overline{x}y$
Transition Functions:
$x' = rx(\overline{ry} + \overline{x}y)$
$y' = ry(\overline{rx} + \overline{y}x)$
Bad State : $Bad(V) = xy$
Let $\widetilde{T}(V, rx, ry, V')$ denote the transition relation.
$\widetilde{T}(x, y, rx, ry, x', y') = (x' \leftrightarrow rx(\overline{ry} + \overline{x}y)(y' \leftrightarrow ry(\overline{rx} + \overline{y}x)$
Since we just want to look at all possible state transitions, we will existentially quantify $\widetilde{T}(V, rx, ry, V')$ over rx and ry to get $T(V, V')$.
$T(V, V') = \exists rx, ry\ \widetilde{T}(V, rx, ry, V')$
$T(V, V') = \bar{x}'\bar{y}' + \overline{x}'y' + x'\overline{y}' + (x' \leftrightarrow \overline{x}y)(y' \leftrightarrow x\overline{y})$
The above transition relation ($T(V, V')$) can now be used to compute the reachable states from the initial state.

Let $R(V)$ denote the set of all states reachable. We will initialize $R(V)$ with $I(V)$ and use the image computation algorithm with the transition relation $T(V, V')$.
Before we start the algorithm, we check that $I(V) \wedge Bad(V) = 0$.
For each iteration, let $F(V)$ denote the set reachable from $R(V)$ in one step.

Step 1: $R(V) = \overline{x}y$
$F(V') = \exists V\ R(V).T(V, V')$
$F(V') = \bar{x}'\bar{y}' + \overline{x}'y' + x'\overline{y}'$
$F(V) = \bar{x}\bar{y} + \overline{x}y + x\overline{y}$
New states were discovered, so we will continue to the next iteration.
$R(V) = R(V) \vee F(V)$
$R(V) \wedge Bad(V) = 0$. Therefore, no bad state reached.

Step 2: $R(V) = \bar{x}\bar{y} + \overline{x}y + x\overline{y}$
$F(V') = \exists V\ R(V).T(V, V')$
$F(V') = \bar{x}'\bar{y}' + \overline{x}'y' + x'\overline{y}'$
$F(V) = \bar{x}\bar{y} + \overline{x}y + x\overline{y}$
No new states were discovered. Therefore, the algorithm terminates. Since no bad states were encountered, the algorithm returns "PASS".

The bad state is not reachable from the initial state. Therefore, the property will pass.

Extra: There is a possible bug in the design which might violate some liveness property that you have written. In the case when both requests become low for some cycles (making $grant0 = 0$ and $grant1 = 0$) and then both requests go high together, neither would be granted. This might lead to a possible deadlock.

2. (a) Rule R1: $a \vee b \rightarrow \neg 2NOR(a, b)$
   Rule R2: $\neg(\neg a) \rightarrow a$
   Rule R3: $\neg a \rightarrow 2NOR(a, a)$
   Rule R4: $a \wedge b \rightarrow \neg(\neg a \vee \neg b)$
   Rule R5: $a \wedge b \wedge c \rightarrow (a \wedge b) \wedge c$

   (b) $(\neg(A \wedge B \wedge C))$
   $\Longrightarrow$ (R5) $(\neg((A \wedge B) \wedge C))$
   $\Longrightarrow$ (R4) $\neg(\neg(\neg A \vee \neg B) \wedge C)$
   $\Longrightarrow$ (R1) $\neg(\neg(\neg 2NOR(\neg A, \neg B)) \wedge C)$
   $\Longrightarrow$ (R2) $\neg(2NOR(\neg A, \neg B) \wedge C)$
   $\Longrightarrow$ (R4) $\neg(\neg(\neg 2NOR(\neg A, \neg B) \vee \neg C))$
   $\Longrightarrow$ (R2) $(\neg 2NOR(\neg A, \neg B) \vee \neg C)$
   $\Longrightarrow$ (R1) $(\neg 2NOR(\neg 2NOR(\neg A, \neg B), \neg C)$
   $\Longrightarrow$ (R2) $(\neg 2NOR(\neg 2NOR(\neg A, \neg B), \neg C)$
   $\Longrightarrow$ (R3) $(\neg 2NOR(\neg 2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C))$
   $\Longrightarrow$ (R3) $(\neg 2NOR(2NOR(2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C),$
   $2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C)))$
   $\Longrightarrow$ (R3) $2NOR(2NOR(2NOR(2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C),$
   $2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C))),$
   $2NOR(2NOR(2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C),$
   $2NOR(2NOR(A, A), 2NOR(B, B)), 2NOR(C, C)))$

3. Liveness Property

$$\bigwedge_{0\le k<n}\left(\bigvee_{0\le l<n-1} p_{k,l}\right)$$

Safety Property

$$\bigwedge_{k}\bigwedge_{i}\bigwedge_{j\neq i}\neg p_{ik}\vee\neg p_{jk}$$

Symmetry is the reason why SAT Solvers find it difficult to computer satisifiability.

4. $(x1'\bigvee x2')\bigwedge(\text{x1'}\bigvee x3')\bigwedge(\text{x1'}\bigvee x4')\bigwedge(\text{x1'}\bigvee x5')\bigwedge(\text{x2'}\bigvee x3')\bigwedge(\text{x2'}\bigvee x4')\bigwedge(\text{x2'}\bigvee x5')\bigwedge(\text{x3'}\bigvee x4')\bigwedge(\text{x3'}\bigvee x5')\bigwedge(\text{x4'}\bigvee x5')\bigwedge(\text{x1}\bigvee x2\bigvee x3\bigvee x4\bigvee x5)$

5. (a) Transition Functions:
   $s1' = s1\overline{s3} + \overline{s3}s1$
   $s2' = s1$
   $s3' = s2$

   (b) Let $\widetilde{T}(V,V')$ denote the transition relation.
   $\widetilde{T}(V,V') = (s1' \leftrightarrow s1\overline{s3} + \overline{s3}s1)(s2' \leftrightarrow s1)(s3' \leftrightarrow s2)$

   (c) The state $\overline{s1}.\overline{s2}.\overline{s3}$ is unreachable and it is proved by forward reachability algorithm as below. The above transition relation ($T(V,V')$) can now be used to compute the reachable states from the initial state. Given $I(V) = s1.\overline{s2}.\overline{s3}$, let $R(V)$ denote the set of all states reachable. We will initialize $R(V)$ with $I(V)$ and use the image computation algorithm with the transition relation $T(V,V')$.
   For each iteration, let $F(V)$ denote the set reachable from $R(V)$ in one step.

   Step 1: $R(V) = s1.\overline{s2}.\overline{s3}$
   $F(V') = \exists V\ R(V).T(V,V')$
   $F(V') = s1'.s2'.\overline{s3'}$
   $F(V) = s1.s2.\overline{s3}$
   New state is discovered, so we will continue to the next iteration.
   $R(V) = R(V) \vee F(V)$

   Step 2: $R(V) = s1\overline{s2}.\overline{s3} + s1.s2.\overline{s3}$
   $F(V') = \exists V\ R(V).T(V,V')$
   $F(V') = s1'.s2'.\overline{s3'} + s1'.s2'.s3'$
   $F(V) = s1.s2.\overline{s3} + s1.s2.s3$
   New state is discovered, so we will have to continue to the next iteration.
   (Iteration 3 to 6 will be similar)

Step 7: $R(V) = s1\overline{s2}.\overline{s3} + s1.s2.\overline{s3} + s1.s2.s3 + \overline{s1}.s2.s3 + s1.\overline{s2}.s3 + \overline{s1}.s2.\overline{s3} + \overline{s1}.\overline{s2}.s3$
$F(V') = \exists V\ R(V).T(V,V')$
$F(V') = s1'\overline{s2'}.\overline{s3'} + s1'.s2'.\overline{s3'} + s1'.s2'.s3' + \overline{s1'}.s2'.s3' + s1'.\overline{s2'}.s3' + \overline{s1'}.s2'.\overline{s3'} + \overline{s1'}.\overline{s2'}.s3'$
$F(V) = s1\overline{s2}.\overline{s3} + s1.s2.\overline{s3} + s1.s2.s3 + \overline{s1}.s2.s3 + s1.\overline{s2}.s3 + \overline{s1}.s2.\overline{s3} + \overline{s1}.\overline{s2}.s3$
No new state is discovered, that is, $R(V) = F(V)$. Hence, the unreachable state is $\overline{s1}.\overline{s2}.\overline{s3}$.

(d) If the circuit enters the unreachable state, it will be struck in that state forever. The assertion to check the same is given below.
assert property(@(posedge clk) s1+s2+s3);