**Laboratory 02.2.- “K-Induction - Tricky Timer”**

Authors: Diego Hernandez Ramirez  
**Date: 01/12/19**

# Objective

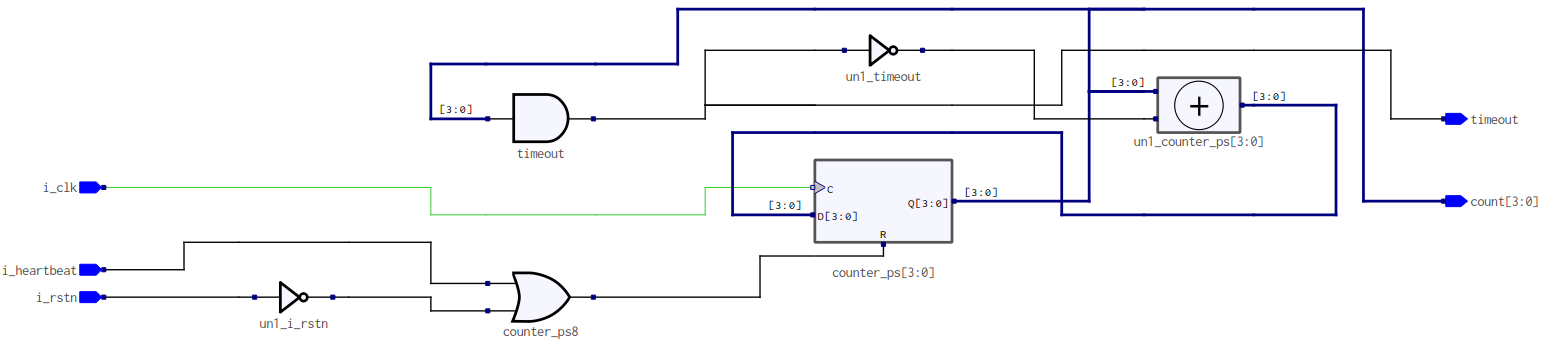
Understand the fundamentals of BMC by using a toy example.

# Material

1. PC
2. SEDA tool, FPV App

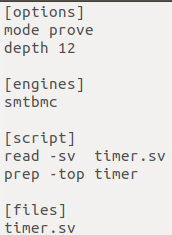
# Practical development

* **Description:** You succeeded with BMC, but let me tell you something: BMC is not complete. That means, there might be bugs that escaped from your past test. Since you need this design to be complete verified, and because you learned that K-induction completes BMC since it introduces induction to the basecase (BMC) process, then we will try it in this tricky timer.
  + Let’s go back and remember the specification of the timer:



* You asked a friend for a design of a timer, whose max value (or the value of the register counter to assert the timeout pin) must be set to 4’d10 (i.e., 11 clock cycles). If counter reaches 4’d10, or more, the **timeout** pin should be asserted.
* If **timeout** happens, timer should stop counting.

You need to reuse your previous working model from the BMC test. To change the mode from BMC to K-induction, you need to change the mode to **prove** in the SBY file, as shown below:



* Run again the SEDA tool.
* If you don’t see any error, congratulations!, you are grasping the concepts just fine.
  + If you do have an error, congratulations!, you will learn something new today.
* If you got an error, look at the terminal output. Remember the theory, there are two process running now: basecase (BMC) and induction (K-induction). BMC passed in last lab, therefore induction is the failing one (you can see that in the output log):

SBY 18:49:22 [timer] engine\_0.induction: finished (returncode=1)

**SBY 18:49:22 [timer] engine\_0: Status returned by engine for induction: FAIL**

SBY 18:49:22 [timer] engine\_0.basecase: ## 0:00:00 Status: passed

SBY 18:49:22 [timer] engine\_0.basecase: finished (returncode=0)

SBY 18:49:22 [timer] engine\_0: Status returned by engine for basecase: pass

SBY 18:49:22 [timer] summary: Elapsed clock time [H:MM:SS (secs)]: 0:00:00 (0)

SBY 18:49:22 [timer] summary: Elapsed process time [H:MM:SS (secs)]: 0:00:00 (0)

**SBY 18:49:22 [timer] summary: engine\_0 (smtbmc) returned FAIL for induction**

SBY 18:49:22 [timer] summary: engine\_0 (smtbmc) returned pass for basecase

**SBY 18:49:22 [timer] DONE (UNKNOWN, rc=4)**

* The solver is in unknown state because one test have a model, and the other a CEX.

***Grade yourself:*** Open the CEX with GTKWave and try to answer this questions:

* Why induction failed?
* Why do you think that, the exact same property that passed in BMC, failed in induction?
* Can you explain the errors you saw so far to your colleagues?, they might have different ideas of how to solve these problems.
* If you learn to write properties, do you think they will help you to design RTL better, by enabling yourself to check all possible bugs before releasing your work?
* How about as a verification engineer?

# Conclusions

After completing this lab, you will be able to understand how to interface with Symbiotic EDA tool for Formal Property Verification, and had a glance of how those flows looks like. Hopefully you can understand some of it usefulness, not only when verifying circuits, but also when modelling them.