**Laboratory 02.2.- “Bounded Model Checking - Tricky Timer”**

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

# Objective

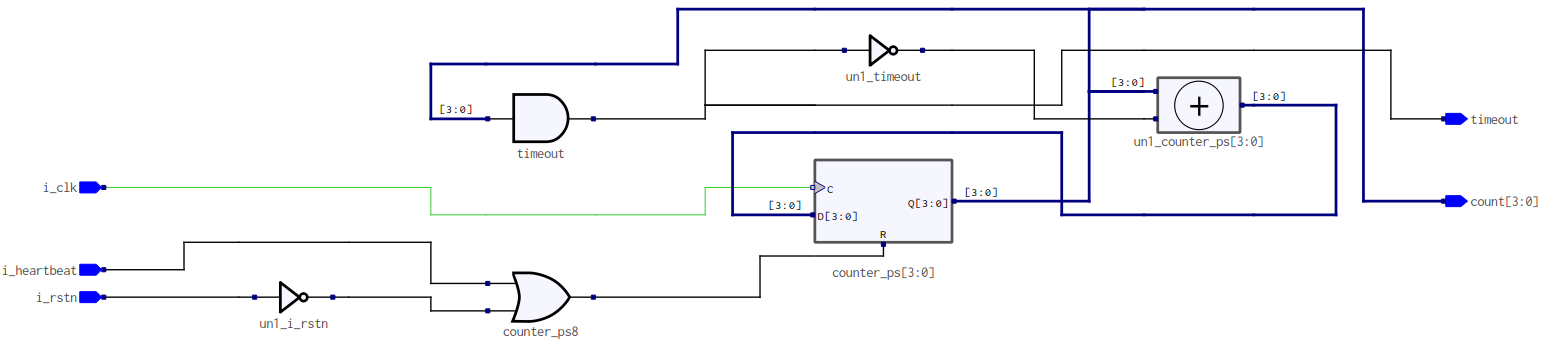
Understand the fundamentals of BMC by using a toy example.

# Material

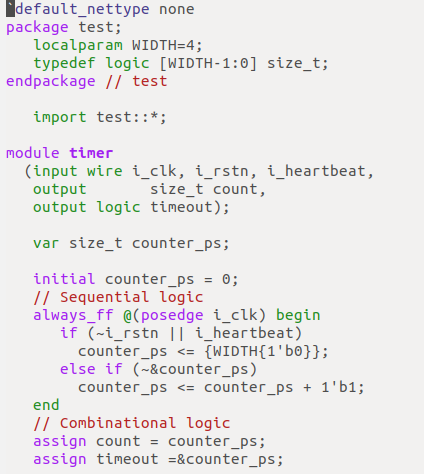
1. PC
2. SEDA tool, FPV App

# Practical development

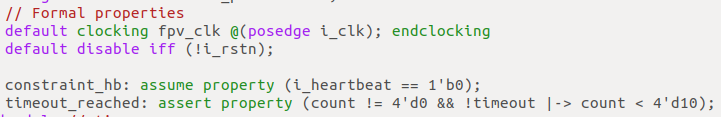
* **Description:** You just learned about BMC, and its capabilities to run proofs of bounded length, therefore, having a limited CEX if exists. Since you had the idea of developing a timer for your realtime processor, on which having more that **exact** N clock cycles within instructions must violate the design, you thought it would be a good idea to develop a timer and check that the timeout pin should not be asserted for N clock cycles strictly needed for instruction completion.



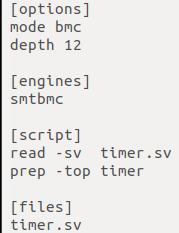
* You asked a friend for a design of a timer, whose max value (or the value of the register counter to assert the timeout pin) must be set to 4’d10 (i.e., 11 clock cycles). If counter reaches 4’d10, or more, the **timeout** pin should be asserted.
* If **timeout** happens, timer should stop counting.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



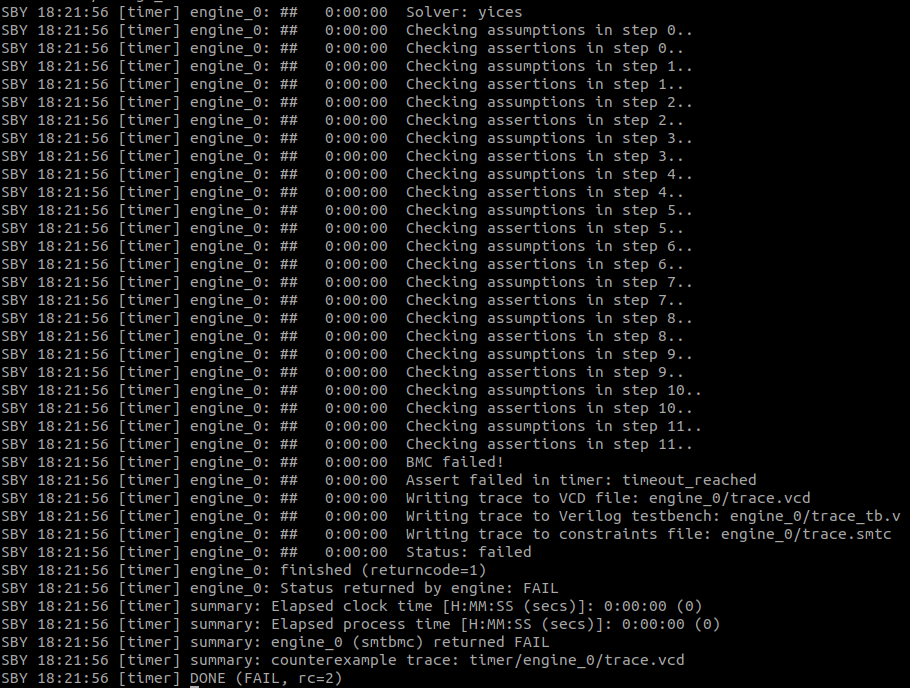
* Since you just learned model checking, you wrote the following assertions and assumptions to your model:
  + The heartbeat signal is never deasserted.
  + The timeout pin is asserted when the counter value is more than or equal to 10d.



* You understand that, for the timer to work correctly, you need 11 clock cycles to test: from counter starting at time t=0, to counter reaching time t=10. You also want to check that the counter will be stopped if **timeout** happens. That means, your assertion should also hold on clock tick 12, because again, if the counter is full at t=11, should remain in that state from t=12 until i\_heartbeat is asserted.
* You developed this SBY file, where you defined the mode as bmc, and depth of 12:



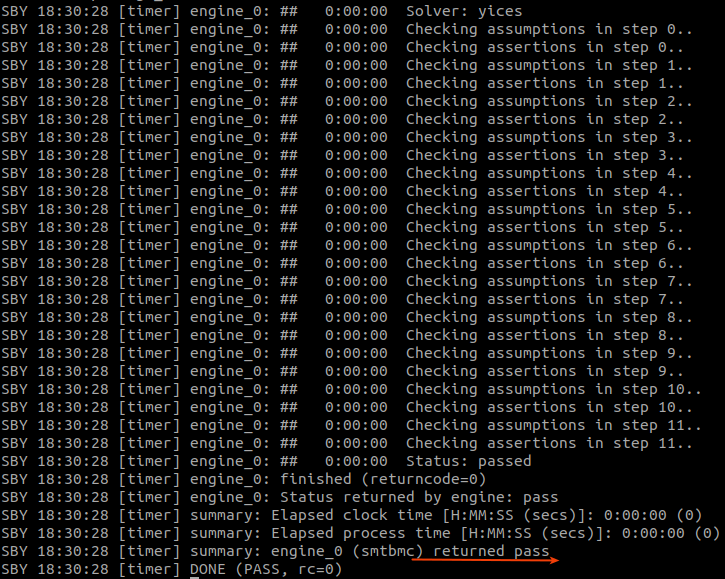
* You ran the formal tool. Analyzing the output you saw that the BMC actually executed 12 steps, but it seems that there is an error in the design.



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

* Is the counter reaching the expecting value?
* Can you spot the problem?
* Usually, when running BMC, you need to increase the **depth** so you can reach more transactions. Do you think that will help here?, why?

***Grade yourself:*** Fix the problem by making that assertion passes. Show that your design is working. Here is a proof of the BMC passing:



# 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.