1. **PUF RING OSCILLATOR Design**
   1. We have designed a 64-bit RNG PUF
   2. There are totally 64 ring oscillators.
   3. Each ring oscillators have fixed 7 inverters (inv), but variable feedback buffers (FB)
   4. Feedback buffers (FB) are added on the feedback path to introduce different amount of propagation delay for different ring oscillators to generate variety of frequencies.
   5. We have split 64 ring oscillators into two groups/sets. Group1(1-32) RNG outputs are connected to the mux1, group2(64-33) RNG outputs are connected to the mux2.
   6. Both Mux1 & Mux2 are 32:1. The challenge input **challenge** [9:0] is serves as a select input to the mux1 and mux2 to select on out of 32 RNG output as a clock to the counters.
   7. LED6-7 are used to display the comparator results
      1. LED6 will glow if counter1 > counter2
      2. LED7 will glow if counter2 > counter1
      3. Both the LDEs cannot glow for a given trial. Only one will glow
   8. LED4-5 are used to display the state of the counters
      1. LED4 will glow if counter1 reaches the maximum value (2^41)
      2. LED5 will glow if counter2 reaches the maximum value (2^41)



1. **Test steps**
   1. load the bit file
   2. observe these LEDs
      1. LED0 red --> RNG disabled
      2. LED1 green --> code is alive  
         2.3 LED2 off (it will be on only when btn3 is pressed)  
         2.4 LED3 is green --> system is out of reset (btn0 presse will put system in reset)  
         2.5 LED4-LED7 are all off  
           
         3. now it is time to set the challenge inputs  
         3.1 set sw[3:0] to any value you want say "0010". it will be used as a mux select for the first 16:1 mux  
         3.2 press btn[2] --> it will latch the sw[3:0] to challenge[3:0]  
         3.3 set sw[3:0] to any value you want which is different from 3.1, say "0001". it will be used as a mux select for the second 16:1 mux  
         3.4 press btn[3] --> it will latch the sw[3:0] to challenge[7:4]  
         3.5 check led4-7, they all should be off, since we haven't yet enabled the rng oscillator  
           
         4. now it is time to enable the RNG oscillator and capture the results  
         4.1 press btn[1], it will enable the RNG oscillator  
         4.2 led0 should turn green when you press btn[1]

# Verification Methodology

Simple testbench is developed using SystemVerilog. Since this is a very simple design complex verification methodology like UVM is not used. The testbench computes the expected clock frequency based on parameter values i.e. Inverter stage and Inverter delay using the following equation,

Where, n = number of inverter stage and t= delay value of single inverter.

This frequency value is compared with that of the measured clock frequency and test passes if both are equal.