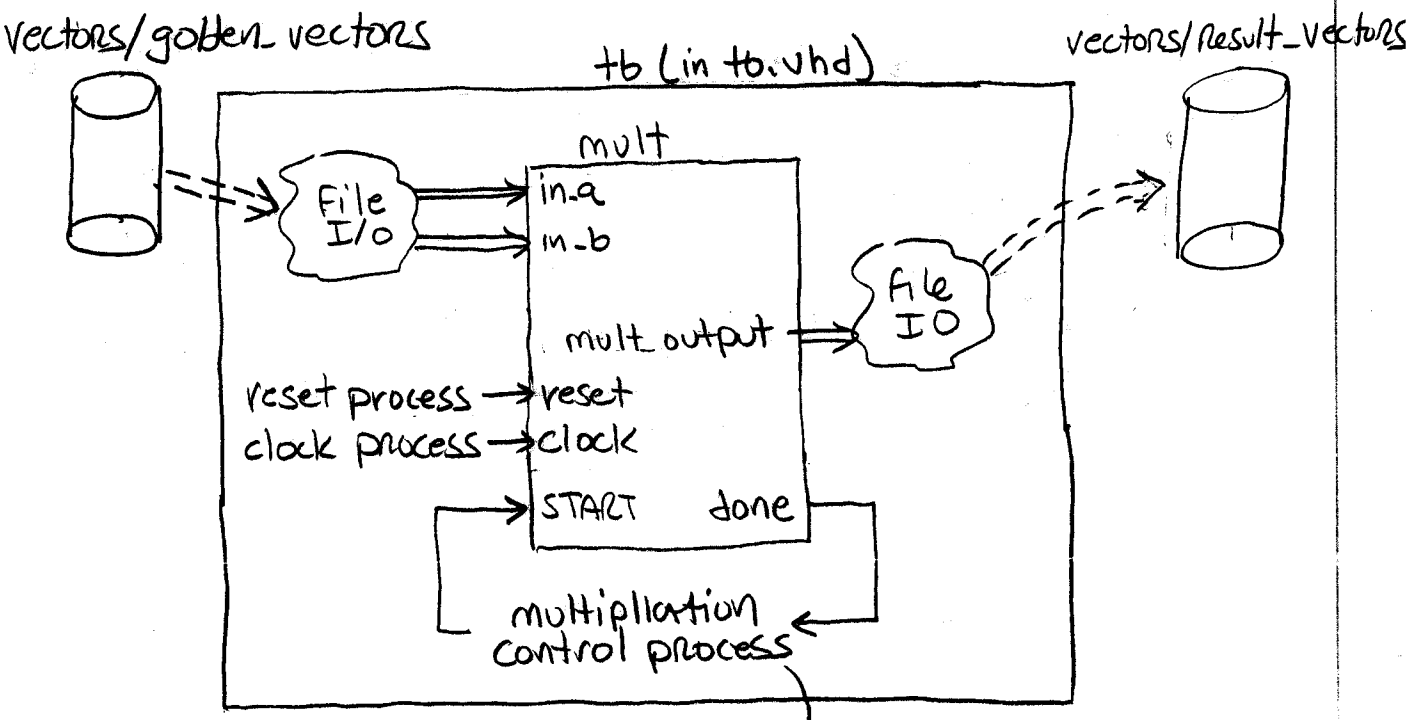
The Testbench For the Multiplier

- Applys external stimulus to multiplier (data, reset, clock)
- captures output data to multiplier (output data)
- does file I/O to read, apply & capture data




```
mult_ctl_process:
PROCESS (reset_n, clk, done) IS
 BEGIN
   IF (reset_n = '0') THEN  -- if reset is asserted
     step <= '0';
     start <= '0' AFTER 5 ns;
   ELSIF (clk'EVENT AND clk='1') THEN
     IF (step = '0') AND (start = '0') AND (done = '0') THEN  -- just kicking off
       start <= '1' AFTER 5 ns; --rising edge of start pulse
     ELSIF (step = '0') AND (start = '1') AND (done = '0') THEN   -- kicked off, end start pulse
       start <= '0'AFTER 1 ns; --create falling edge of start pulse
       step <= '1'; --keep track of the step
     ELSIF (step = '1') AND (done = '1') AND (start = '0') THEN  --beginning of done pulse
        step <= '0';
      END IF;
    END IF;
END PROCESS;
```

From tb.vhd (one possible way to do it)

- Clock: clk <= clk NOT AFTER (clock_period/2);

  Reset: reset_n <= '1' AFTER 15 NS;

* note that the entity for the testbench has no pins.

  ENTITY tb IS
  END tb;

* the same testbench + golden vectors are used to check RTL, GATE, POSTLAYOUT versions.

```
#!/bin/csh
./clean ---- { rm -R work
               rm vectors/*
               rm reports/*
               rm transcript
               rm vsim.wlf

echo "Build Golden Vectors"
perl ./bin/gen.pl ---- { #!/usr/bin/perl
                         open file for writing "open"
                         for ($i=0; $i<24; $i++) {
                            get 2 random numbers
                            compute their product
                            convert to bit strings
                            print to the file
                         }
                         Append corner case data

doit { #make "work" if necessary
       if (! -e work) then
          vlib work
       endif
       echo "compile source code"
       #compile behavorial multiplier
       ./bin/beh_comp ---- { vcom -93 beh_src/mult.vhd

       #compile testbench
       ./bin/tb_comp ---- { run 10000
                            quit -f

       echo "run simulation"
       vsim mult_tb -do bin/beh_tb.do -quiet

       #check results
       strip off results from golden vector file
       compare those results with multipliers results
       write differences to reports/gold2beh_diff

       if (-z reports/gold2beh_diff) then
          echo "regressed beh to golden"
       else
          echo "failure beh to golden
```