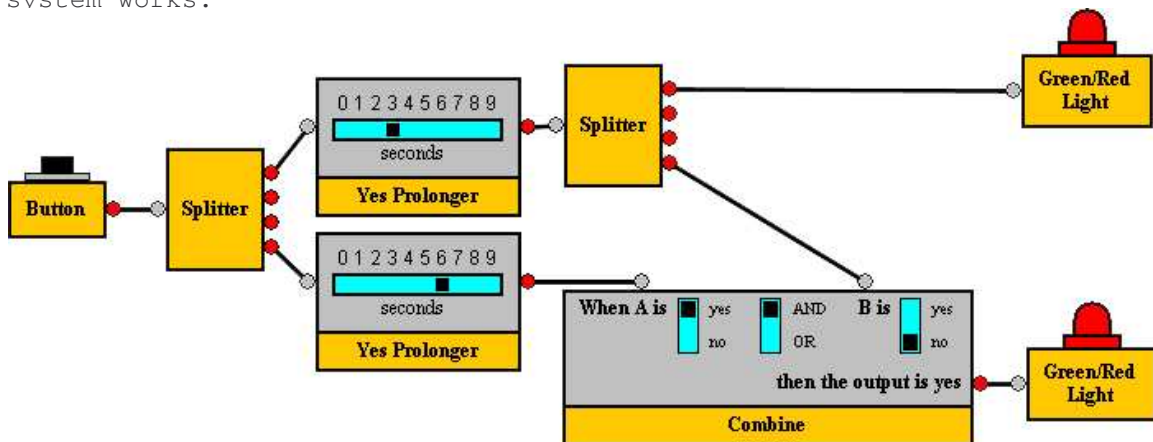


Note: there are three problems - don't forget problem 3 on the back.

1. You have implemented a PID controller in C code for a water heater. You find that the controller quickly gets near the desired reference temperature, but then stays a few degrees off, approaching the desired reference temperature very very slowly. Which constant term, P, I or D, would you change, and how would you change it (increase or decrease)? EXPLAIN the intuition behind your decision - explain how that term works and why changing it will help.

Increase I. I is the integral of the error, meaning the sum of the previous errors. An increase of this sum causes a change in the PID equation. If the change is too slow, increasing I will magnify the sum and thus cause change more quickly.

2. Design an eBlock system to implement a race starting light system. The system has two LEDs, labeled "Ready" and "GO". Initially, both LEDs are off. Pressing a button causes the "Ready" LED to light for 3 seconds, then the "GO" LED for 3 seconds. Then both lights turn off again until the next button press. You may use one or more of each of the following blocks: button, combine (logic), toggle, once-yes-stays-yes, prolonger (adjustable from 1-10 seconds), pulse generator (adjustable 1-10 seconds high, 1-10 seconds low), splitter, opposite, and LED. EXPLAIN how your system works.



The top (Ready) LED is trivially on for 3 seconds. The Combine block is configured such that the bottom (GO) LED is on only when the top LED is off and the bottom Yes Prolonger is on. Since we want the GO LED to be on for 3 seconds and since the Ready LED is on for 3 seconds, the bottom Yes Prolonger is set to $3 + 3 = 6$ seconds so that its on time extends past the top Yes Prolonger's on time.

(Over for problem 3)

3. Write VHDL code for an FPGA to generate an output pulse with a 1 second high time and 0.5 second low time. Assume the FPGA has a 10MHz clock input.

10MHz clock = 1/10000000s clock period

```
entity quiz4_p3 is
  port(
    reset: in STD_LOGIC;
    clock: in STD_LOGIC;
    pulse: out STD_LOGIC
  );
end quiz4_p3;

architecture bhv of quiz4_p3 is
begin
  process(reset, clock)
    variable count: integer := 0;
  begin
    if (reset = '1') then
      count := 0;
      pulse <= '1';
    elsif (clock = '1' and clock'event) then
      count := count + 1;

      -- count to 1s for a 1s high time
      if (count < 10000000) then
        pulse <= '1';

        -- count to 1.5s for a 0.5 low time
        elsif (count < 15000000) then
          pulse <= '0';

        else
          count := 0;
          pulse <= '1';
        end if;
      end if;
    end process;
  end bhv;
```