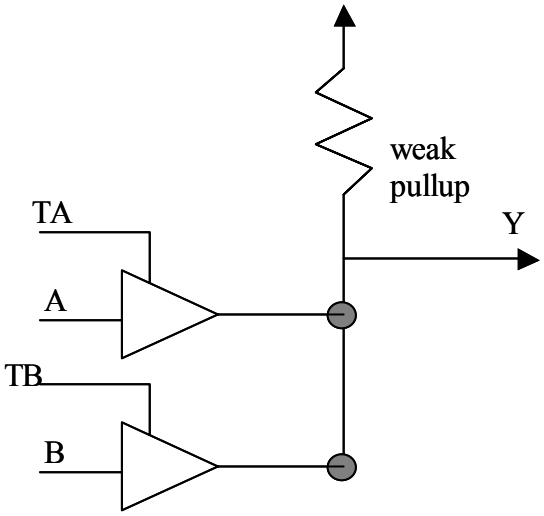
Summer 2001 – Test 1 Solutions

1. (10 pts) Write a single process that will function as a clock divide-by-2 function. You can use any approach that you desire (with or without a sensitivity list). The new clock signal should be called *clkhalf*" and the input clock signal is called *clk*. Use *std_logic* for the signal types and assume that the clock only has values of '1' and '0'. You CANNOT assume that the duty cycle is 50%.

```
signal  clkhalf : std_logic := '0';
process (clk)
begin
        if (clk = '1') then clkhalf <= not (clkhalf);
        end if;
end process;
```

2. Write a VHDL code segment that models the situation shown below. The signal declarations are as shown. When neither tri-state buffer is enabled, the Y value should be a weak '1' condition, and if both tri state buffers are enabled then Y should be a strong unknown.

```
Signal  TA, A, TB, B, Y :  std_logic;
```




```
-- use multiple driver approach

Y <= 'H';     -- pullup
Y <= A when (TA = '1') else 'Z';
Y <= B when (TB = '1') else 'Z';
```

When tristate control is '1', then output follows input
else high impedance output.

3. (15 pts) a. Draw the waveform generated by the following code:

SEE ATTACHED FIGURE ('a3' waveform)

```
architecture A  of E
    begin
    signal  a: std_logic = 'Z';

        a <=  transport 'H'  after 3 ns;
         a <=   transport 'L'  after  5 ns;
        a  <=   transport  '0 ' after   7ns;

end  E;
```

4. (10 pts) Draw the waveform generated by the following process for at least 20 ns. ('wait on' waits for an event on the specified signal).

SEE ATTACHED FIGURE ('a4' waveform)

```
signal   A: std_logic:= '0';
process
begin

    A <= transport '1' after 5 ns;
   wait on  A;
   A <=  transport '0' after 5 ns;
   Wait on A;
End process;
```

5. (10 pts) Draw the waveform generated by the following process for at least 20 ns.

SEE ATTACHED FIGURE ('a5' waveform)

```
signal   A: std_logic:= '0';
process
begin

    A <= transport '1' after 5 ns;
   wait on  A;
   A <=  transport '0' after 5 ns;
   Wait;
End process;
```

6. (10 pts) Draw the waveform generated by the following process for at least 20 ns.

SEE ATTACHED FIGURE ('a6' waveform)

```
signal   A: std_logic:= '0';
process
begin

    A <= transport '1' after 5 ns;
   wait on  A;
   A <=  transport '0' after 5 ns;
End process;
```

7. (10 pts) Draw the waveform generated by the following process for at least 20 ns.

SEE ATTACHED FIGURE ('a7' waveform)

```
signal   A: std_logic:= '0';
process
begin

    A <= transport '1' after 5 ns;
   wait for 2 ns;
   A <=  transport '0' after 2 ns;
   Wait;
End process;
```

8. (15 pts) Draw the waveforms generated by the following VHDL code fragment for at least 20 ns. (Hint: recall the difference between inertial and transport delay models).

SEE ATTACHED FIGURE ('a8', 'b8', 'c8' waveforms)

```
signal   A, B, C: std_logic:= '0';

    A <= transport '1' after 4 ns, '0' after 6 ns;
   B <= transport A after 3 ns;
   C <= A after 3 ns;
```

9. (10 pts) In the process below, what is the value of 'cnt' at time = 25 ns? EXPLAIN your answer to get partial credit.

SEE ATTACHED FIGURE ('a9', 'b9') CNT = 4 because the process is triggered twice on assignments to A, B.

```
Signal   A, B : std_logic := '0';


Process (A, B)
 Variable cnt: = 0;
 Variable ll :line;
 Variable init :boolean

  Begin

    If (init = FALSE) then
      Init := TRUE;
    Else
      Cnt := cnt + 1;    -- do not increment CNT for initial triggering of process
    End if;
    A <=  transport not(A) after 10 ns;
    B <= transport A;
  End process;
```

10. Write a VHDL process that will generate a fixed number of clock cycles based upon a generic called 'CLKNUM'. The clock is a 50% duty cycle clock whose period is a generic called 'CLKPER'. The initial value of the clock signal should '0' and the clock signal name is 'clk'.

```
Signal   clk:   std_logic;


process

  begin

    for i in 0 to CLKNUM-1 loop

      wait for CLKPER/2;
      clk <= not(clk);
      wait for CLKPER/2;
      clk <= not(clk);
    end loop;
    wait;
 end process;
```

11. (10 pts) Assume that 'clk' is a 50% duty cycle clock with a 10 ns period.

    a. In the process below, what is the value of 'atime' ? **0 because process triggers on clk.**

```
Process (clk)
Variable atime: time;

   Atime := clk'last_event;
End process;
```

    b. In the process below, what is the value of 'atime' ? **5 ns (trigger on each clk change)**

```
Process (clk)
Variable atime: time;

   Atime := clk'delayed'last_event;
End process;
```

12. (5 pts) Given a signal 'A', what would need to be in the sensitivity list of a process if I wanted the process to be triggered each time an ASSIGNMENT was made to 'A'?

    A'TRANSACTION

13. (5 pts) What is difference between *std_logic* and *std_ulogic* data types? When would I use one over the other?

    Std_logic is the resolved version of std_ulogic. You would use std_logic for signals that need multiple drivers, and either std_logic or std_ulogic if the signal only has one driver.