```
end counter;
architecture dataflow of counter is
signal latch_in, latch_out, incr_out : std_ulogic_vector(0 to 12);
begin
  sync: process(clock)
    begin
      if (clock = '1' and not(clock'stable)) then
         latch_out <= latch_in;
      end if;
    end process;
  latch_in <= "0000000000000" when (start_timer = '1') else incr_out;
  incr_out <= latch_out + '1';
  short <= '1' when (incr_out = "0100000000000") else
           '0';
  long <= '1' when (incr_out = "1000000000000") else
          '0';
end dataflow;
```

```
          when "10" =>
            hl1 <= '0';
            hl0 <= '1';
            fl1 <= '0';
            fl0 <= '0';
            if (short = '1') then
              next_state <= "00";
              start_timer <= '1';
            else
              next_state <= "10";
              start_timer <= '0';
            end if;
          when others =>
            next_state <= "XX";
            start_timer <= 'X';
            hl0 <= 'X';
            hl1 <= 'X';
            fl0 <= 'X';
            fl1 <= 'X';
          end case;
       end if;
    end process;
end behavior;
```

VHDL for the counter

```
-- Counter is set up for a 1kHz clock.  Short
-- will occur after 4 seconds, and long after
-- 40 seconds.
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity counter is
  port(start_timer: in std_ulogic;
       clock: in std_ulogic;
       short, long: out std_ulogic);
```

```
  -- set lights
  hl1 <= '0';
  hl0 <= '0';
  fl1 <= '0';
  fl0 <= '1';
  if (cars and long) = '1' then
    next_state <= "01";
    start_timer <= '1';
  else
    next_state <= "00";
    start_timer <= '0';
  end if;
when "01" =>
  -- set lights
  hl1 <= '1';
  hl0 <= '0';
  fl1 <= '0';
  fl0 <= '1';
  if short = '1' then
    next_state <= "11";
    start_timer <= '1';
  else
    next_state <= "01";
    start_timer <= '0';
  end if;
when "11" =>
  hl1 <= '0';
  hl0 <= '1';
  fl1 <= '0';
  fl0 <= '0';
  if (not(cars) or long) = '1' then
    next_state <= "10";
    start_timer <= '1';
  else
    next_state <= "11";
    start_timer <= '0';
  end if;
```

Columbia University
Department of Electrical Engineering
EE E4340. VHDL for the traffic light controller.

VHDL for the controller block

```
library ieee;
use ieee.std_logic_1164.all;

entity traffic_controller is
  port(clock: in std_ulogic;
       reset: in std_ulogic;
       cars: in std_ulogic;
       short, long: in std_ulogic;
       hl1, hl0: out std_ulogic;
       fl1, fl0: out std_ulogic;
       start_timer: out std_ulogic);
end traffic_controller;

architecture behavior of traffic_controller is

  signal current_state: std_ulogic_vector(0 to 1);
  begin

    sync: process(clock)
      if ((clock = '1') and not(clock'stable)) then
        current_state <= next_state;
      end if;
    end process sync;

    fsm_comb: process(current_state, short, long, cars)
    begin
      if (reset = '1') then -- asynchronous reset
        next_state <= "00";
      else
        case current_state is
          when "00" =>
```