```vhdl
-----------------------------------------------------------------------
-- This component divides the 50 MHz clock signal down to a clk signal
-- of 762Hz and a clk_3 signal of 3Hz.
-----------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity clock_manager is
    port(clk_50  : in  std_logic;
         clk_man : in  std_logic;
         sel_man : in  std_logic;
         clk     : out std_logic;
         clk_3   : out std_logic);
end clock_manager;

architecture structure of clock_manager is
    signal count, count_next   : unsigned(15 downto 0) := (others => '0');
    signal count3, count3_next : unsigned(23 downto 0) := (others => '0');
    attribute clock_signal : string;
    attribute clock_signal of clk : signal is "yes";

begin
    count_next  <= count + 1;
    count3_next <= count3 + 1;

-------------------------------------------------------------------------
-- this process generates the clk and clk3 signal, which are derived from
-- the 50MHz clock. When MSB of count = '1' clk will go high, when MSB of
-- count = '0' clk will go low. Same with count3 and clk_3. If sel_man is
-- set to 1, clk will be set to the input signal clk_man.
-------------------------------------------------------------------------

    process(sel_man, clk_man, count, count3)
    begin
        if sel_man = '1' then
            clk <= clk_man;
        else
            clk <= count(15);
        end if;
        clk_3 <= count3(23);
    end process;

--------------------------------------------
-- on the rising edge of the 50MHz clock
-- count and count3 will be increased by 1
--------------------------------------------
```

```

    process(clk_50, count, count3)
    begin
        if rising_edge(clk_50) then
            count  <= count_next;
            count3 <= count3_next;
        end if;
    end process;

end structure;

```