Programmable Frequency Divider

Tuesday, April 13, 2021 6:43 PM

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;
entity FreqDivProg is
generic(N : positive := 8);
      port(reset : in std_logic;
      divFactor : in std_logic_vector(N-1 downto 0);
      clkIn : in std_logic;
      clkOut : out std_logic);
End FreqDivProg;
architecture Behavioral of FreqDivProg is
      signal s_divCounter : unsigned(N-1 downto 0);
begin
      process(clkIn)
      begin
            if (rising_edge(clkIn)) then
                  if ((reset = '1') or (s_divCounter >= unsigned(divFactor) - 1)) then
                        clkOut <= '0';
                        s_divCounter <= (others => '0');
                  else
                        if (s_divCounter = (unsigned(divFactor)/2 - 1)) then
                              clkOut <= '1';
                        end if;
                        s_divCounter <= s_divCounter + 1;
                  end if;
            end if;
      end process;
end Behavioral;
```

Notice:

- N bits counting internal component