Parametric Frequency Divider

Tuesday, April 13, 2021 5:48 PM

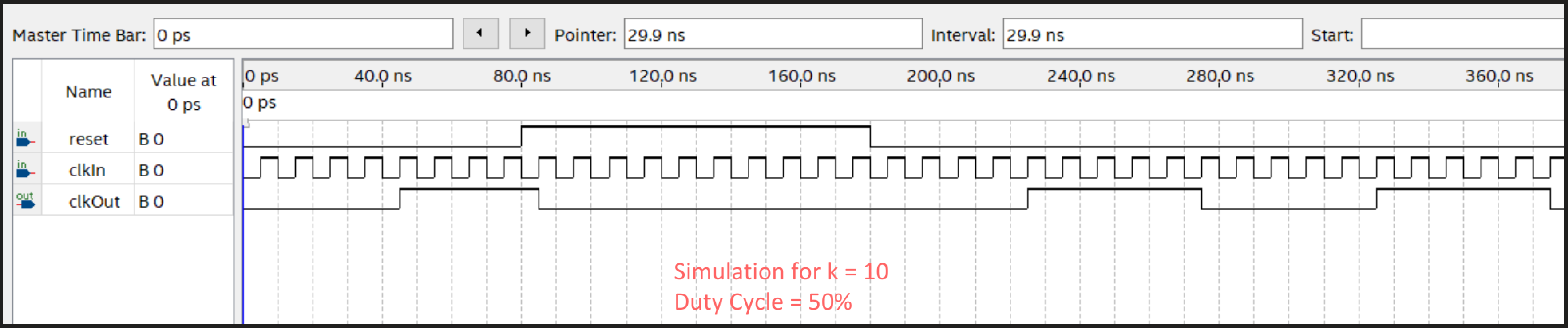
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;
entity FreqDiv is
      generic(K : positive :=10);
      port(reset : in std_logic;
           clkIn : in std_logic;
             clkOut : out std_logic);
end FreqDiv;
architecture Behavioral of FreqDiv is
signal s_counter : natural;
begin
      process(clkIn)
      begin
            if rising_edge(clkIn) then
                  if ((reset = '1') or (s_counter = K -
1)) then
                              clkOut <= '0';
                              s_counter <= 0;
                  else
                        if (s_counter = K/2 - 1) then
                              clkOut <= '1';
                        end if;
                        s_counter <= s_counter + 1;
                  end if;
            end if;
      end process;
end Behavioral;
```

Notice:

- K could be any positive
- This a parametic but programmable freq. divider

Notice:

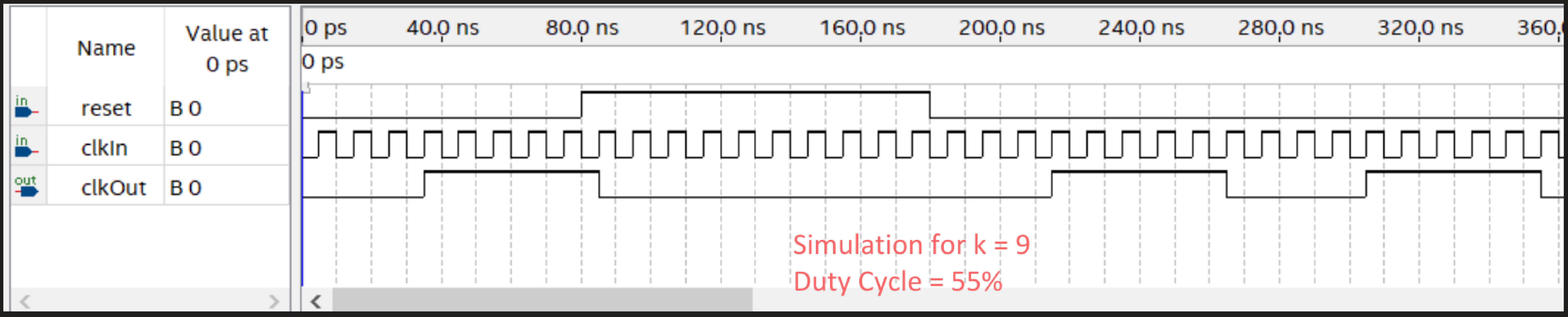
- Implies a 32 bit counting component because of the range of "natural" type.
- See the netlist



Master Time Bar: 0 ps
Pointer: 29.9 ns
Interval: 29.9 ns
Start:

Name	Value at 0 ps
reset	B 0
clkIn	B 0
clkOut	B 0

Simulation for k = 10
Duty Cycle = 50%



Name	Value at 0 ps
reset	B 0
clkIn	B 0
clkOut	B 0

Simulation for k = 9
Duty Cycle = 55%