LAB3:

Sequential VHDL 1

**Introduction:**

In this lab, we have designed a register to divide clock frequency by two. This is done by two different techniques.

First, we designed a flip flop register with an inverted feedback. The output is single bit clock with half frequency.

In the second technique, we have designed a 3-bit register with counter as feedback which adds 1 to each clock.

In the register, we have two inputs. First one is clock and the other one is an active low reset (nrst).

The counter will increment its value on every rising edge of the clock.

**RTL VIEW:**

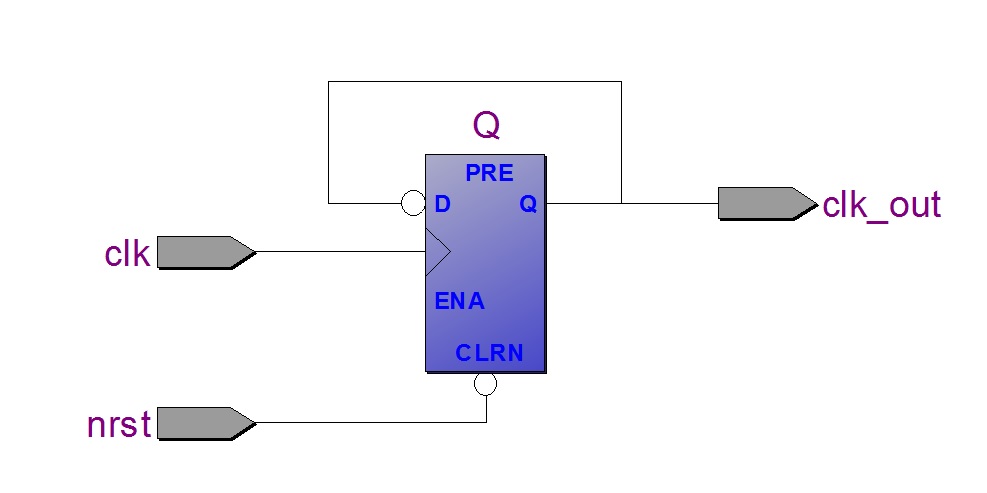


Figure 1: showing RTL view of flip flop register with an inverted feedback.

**Waveform:**

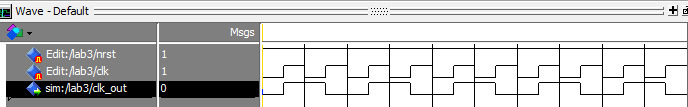


Figure 2: showing simulation of inverted feeedback flip flop register (output frequency divided by two)

**CODE:**

Library ieee;

Use ieee.std\_logic\_1164.ALL;

Entity lab3 is

port( nrst, clk: in std\_logic;

clk\_out : out std\_logic

);

End Entity;

Architecture rtl of lab3 is

signal Q: std\_logic := '0';

begin

freq\_Div\_2: process(nrst,clk) -- Asynchronous reset

--variable Q: std\_logic := '0';

begin

if nrst = '0' then -- Active low

Q <= '0';

elsif clk'event and clk='1' then

Q <= not Q;

End if;

End process;

clk\_out <= Q;

End Architecture;

**Counter Register:**

**RTL VIEW:**

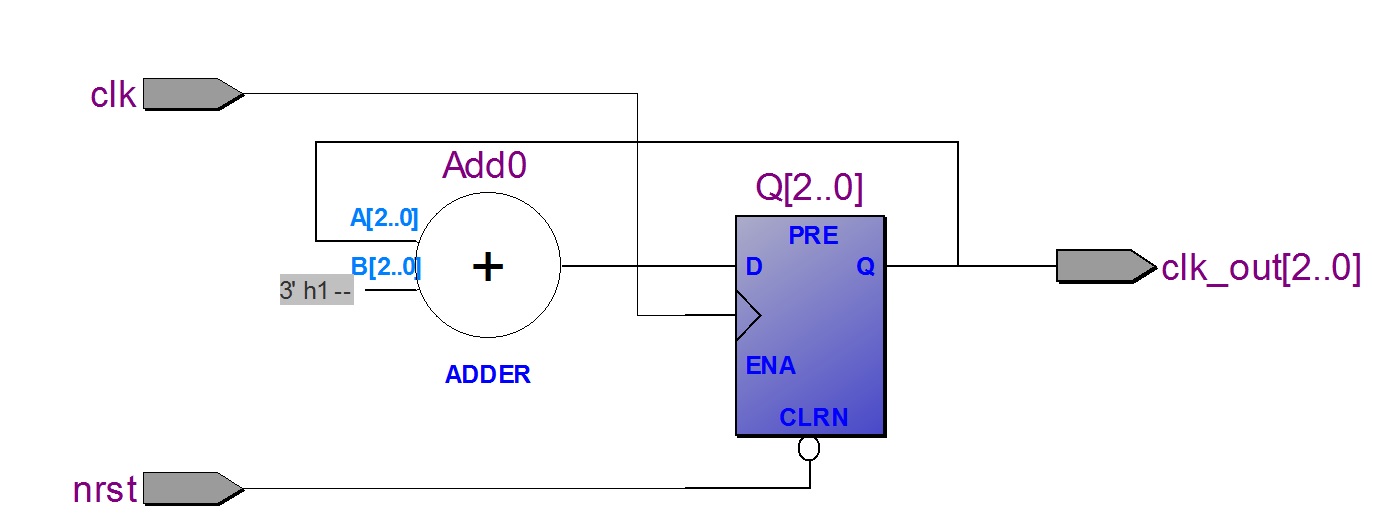


Figure 3: An RTL view of a 3-bit counter register.

**WAVEFORM:**

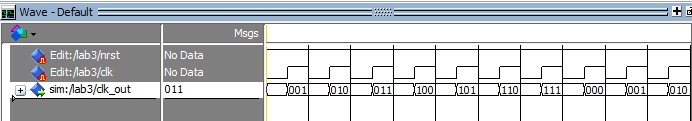


Figure 4: Simulation showing 3-bit output of a counter register (with clock frequency divided by two).

**CODE:**

Library ieee;

Use ieee.std\_logic\_1164.ALL;

Use ieee.std\_logic\_unsigned.ALL;

Entity lab3 is

port( nrst, clk: in std\_logic;

clk\_out : out std\_logic\_vector(2 downto 0)

);

End Entity;

Architecture rtl of lab3 is

signal Q: std\_logic\_vector(2 downto 0) := "000";

begin

freq\_Div\_2: process(nrst,clk) -- Asynchronous reset

--variable Q: std\_logic := '0';

begin

if nrst = '0' then -- Active low

Q <= "000";

elsif clk'event and clk='1' then

Q <= Q +1;

End if;

End process;

clk\_out <= Q;

End Architecture;

**Conclusion:**

Using VHDL, we can design a sequential feedback register to divide an input frequency which can be used further in different tasks.