**MOD-N Counter**

VHDL code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

use IEEE.STD\_LOGIC\_arith.ALL;

entity modcount is

--generic (n:natural :=8);

Port ( en : in STD\_LOGIC;

clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

y : out STD\_LOGIC\_VECTOR (7 downto 0));

end modcount;

architecture modarch of modcount is

signal count: unsigned (7 downto 0);

begin

process(clk,rst,en)

begin

if rst='1' then

count <= (others => '0');

elsif (clk ' event and clk = '1') then

if en='1' then

count <= count + '1';

end if;

end if;

end process;

y <= std\_logic\_vector(count);

end modarch;

**Testbench:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY modtb IS

END modtb;

ARCHITECTURE behavior OF modtb IS

COMPONENT modcount

PORT(

en : IN std\_logic;

clk : IN std\_logic;

rst : IN std\_logic;

y : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

--Inputs

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

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

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

--Outputs

signal y : std\_logic\_vector(7 downto 0);

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: modcount PORT MAP (

en => en,

clk => clk,

rst => rst,

y => y

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

rst <= '1';

wait for 100 ns;

rst <= '0';

en <= '1';

wait for 100 ns;

end process;

END;



