Räknare med synkron Reset

```vhdl
-- en 4-bitars binärräknare som räknar modulo-16 med synkron reset
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity cnt4b_r is
   port(clock, reset:in std_logic;
      q:out std_logic_vector(3 downto 0));
end entity cnt4b_r;

architecture beteende of cnt4b_r is
subtype state_type is integer range 0 to 15;
signal present_state, next_state:state_type;
begin
   process(present_state,reset)
      begin
         if reset='1' then next_state<=0;
         elsif present_state>=15 then next_state<=0;
         else
             next_state<=present_state+1;
         end if;
   end process;
   q<=conv_std_logic_vector(present_state,4);

   process(clock)
   begin
      if rising_edge(clock) then
         present_state<=next_state;
      end if;
   end process;
end architecture beteende;
```

Räknare med asynkron reset

```vhdl
-- en 4-bitars binärräknare som räknar modulo-16 och med asynkron reset!!!

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity cnt4b is
   port(clock,reset:in std_logic;
      q:out std_logic_vector(3 downto 0));
end entity cnt4b;

architecture beteende of cnt4b is
subtype state_type is integer range 0 to 15;
signal present_state, next_state:state_type;
begin
   process(present_state)
   begin
         if present_state=15 then next_state<=0;
            else
               next_state<=present_state+1;
         end if;
   end process;
   q<=conv_std_logic_vector(present_state,4); -- omvandlar en integer till en 4-bitars vektor.

   process(clock,reset)
   begin
      if reset='1' then
         present_state<=0;
         else
            if rising_edge(clock) then
               present_state<=next_state;
            end if;
      end if;
   end process;
end architecture beteende;
```