```
library ieee;
use ieee.std_logic_1164.all;

entity ls165 is
	port(PL, CP, CE, DS: in std_logic;
	D: in std_logic_vector(0 to 7);
Q7, Q7INV: out std_logic);
end ls165;

architecture toiminta of ls165 is
begin
	process(CP, CE, PL)		--CP on kello
	variable apu, result: std_logic_vector(0 to 7);
	begin
		apu:=result;   --   := muuttujaan sijoitus
		if(PL='0') then result:=D;
		elsif ((CE='0') and (CP='1' and CP'event)) then
			for index in 0 to 6 loop
				result(index+1):=apu(index);
			end loop;
			result(0):=DS;
		end if;
		Q7<=result(7);			--kirjoitetaan ulostulo
		Q7INV<= (not result(7));
	end process;
end toiminta;
```

```
--esim2, ls163
library ieee;
use ieee.std_logic_1164.all;
entity ls163 is
	port(CLR, LOAD, ENT, ENP, CLK: in std_logic;
	D: in std_logic_vector(3 downto 0);   --   A-D
	RCO: out std_logic;
Q: out std_logic_vector(3 downto 0));
end ls163;

architecture toiminta of ls163 is
	function inc_counter(input :std_logic_vector)
	return std_logic_vector is
		variable result: std_logic_vector(3 downto 0):=input;
		variable carry : std_logic:='1';
		begin
			for index in 0 to 3 loop
				result(index):=result(index) xor carry;
				carry:=input(index) and carry;
				exit when carry='0';
			end loop;
		return result;
	end inc_counter;
begin
	process(CLK, LOAD, ENT, ENP)
		variable tulos: std_logic_vector(3 downto 0);
	begin
		if(CLK'event and CLK='0') then
    if (CLR='0') then tulos:="0000";
			elsif(LOAD='0') then tulos:=D;
			elsif (ENT='1' and ENP='1') then   --rivi jatkuu
				tulos:=inc_counter(tulos);
			end if;
		end if;
		Q<=tulos;
		if (tulos="1111") then RCO<='1';
		else RCO<='0';
		end if;
	end process;
end toiminta;
```