

VHDL-esimerkit 6.2.13

```
--esim1
entity invertteri is
    port(A:in std_logic; Y:out std_logic);
end invertteri;

architecture toiminta of invertteri is
begin
    process(A)
    begin
        case A is
            --kohdistuu A-inputtiin
            when '0' => Y<='1' ; --kun A on 0 niin Y=1
            when '1' => Y<='0';
        end case;
    end process;
end toiminta;
```

```
--esim2 osa laskurista, port-määrittely ei sisällä kaikkea
port(CLK: in std_logic; Q:out std_logic_vector( 3 downto 0));
...
process(CLK)
    variable i:integer range 0 to 3; --oikea paikka muuttujille
begin
    if(CLK='0' and CLK'event) then --testaa laskevaa reunaa
        if (i<3) then i:=i+1; --sijoitus muuttujaan :=
        else i:=0;
        end if;
    end if;
    case i is
        when 0 => Q<="0000" ;
        when 1 => Q<="0001" ;
        when 2 => Q<="0010" ;
        when 3 => Q<="0011" ;
    end case;
end process;
```

```

--Esim 3, Harj 2 b)
entity dff_v2 is
    port(D, CLK:in std_logic; Q:out std_logic);
end dff_v2;

architecture toiminta of dff_v2 is
begin
    process(CLK, D)    --process(CLK) myös ok
        function nousureuna(signal s: std_logic) return boolean is
        begin
            if(s='1' and s'event) then return true;
            else return false;
            end if;
        end nousureuna;
    begin
        if (nousureuna(CLK)=true) then Q<=D;
        end if;
    end process;
end toiminta;

```

```

--Esim 4, Harj 10/2c
entity dff_v3 is
    port(CLK, D: in std_logic; Q:out std_logic);
end dff_v3;

architecture toiminta of dff_v3 is
begin
    process(CLK)    --riittää, voisi olla (D, CLK)
        procedure dkiikku(signal c,dd: in std_logic; qq:out std_logic) is
        begin
            if(c='1' and c'event) then qq<=dd;
            end if;
        end dkiikku;
    begin
        dkiikku(CLK, D, Q);
    end process;
end toiminta;

```