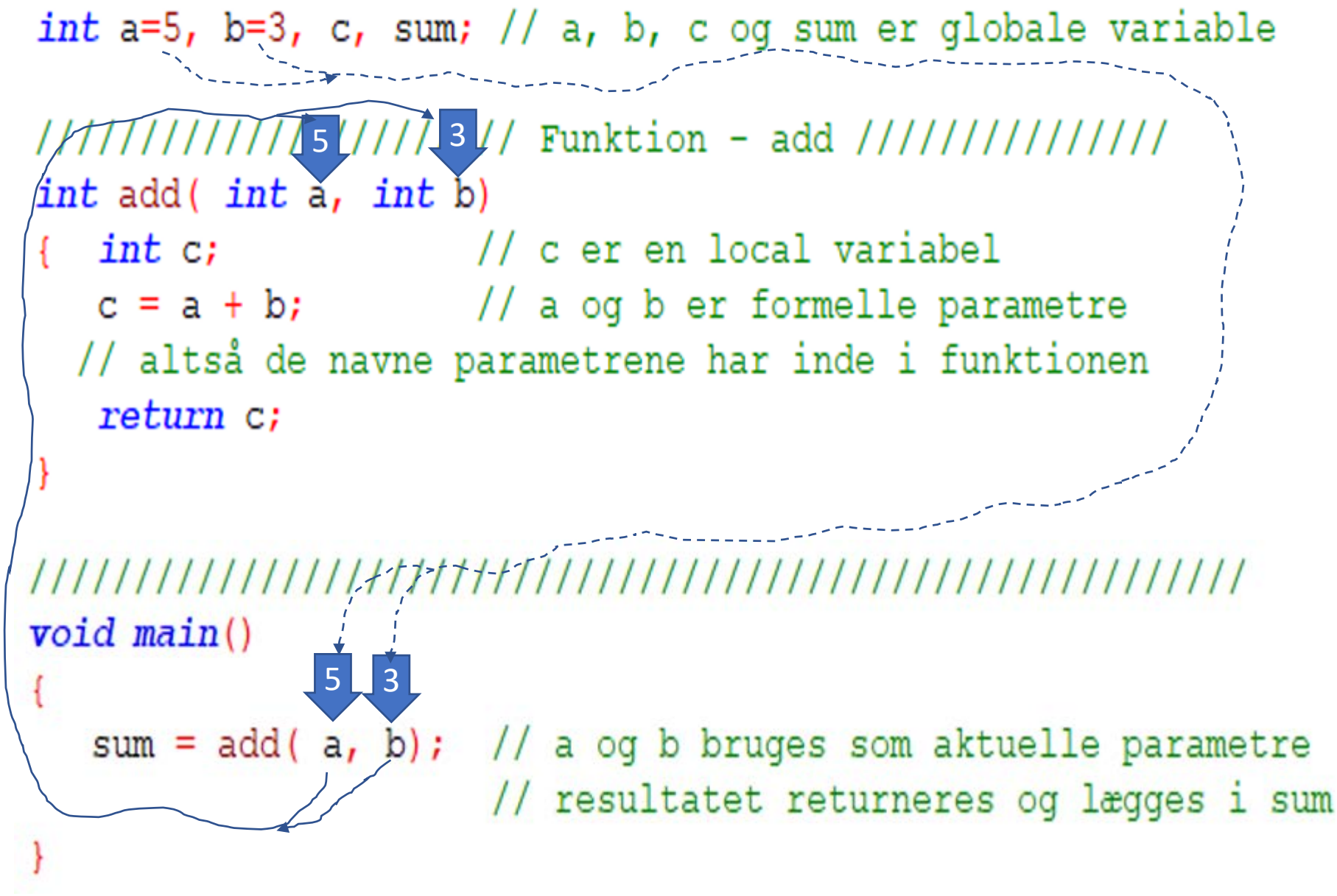
C-program og add funktion

```
//////////////////////// Funktion - add ///////////////
int add( int a, int b)
{   int c;              // c er en local variabel
    c = a + b;          // a og b er formelle parametre
   // altså de navne parametrene har inde i funktionen
    return c;
}

int a=5, b=3, c, sum; // a, b, c og sum er globale variable

//////////////////////////////////////////////////////////
void main()
{
    sum = add( a, b);  // a og b bruges som aktuelle parametre
                       // resultatet returneres og lægges i sum
}
```

C-program og add funktion



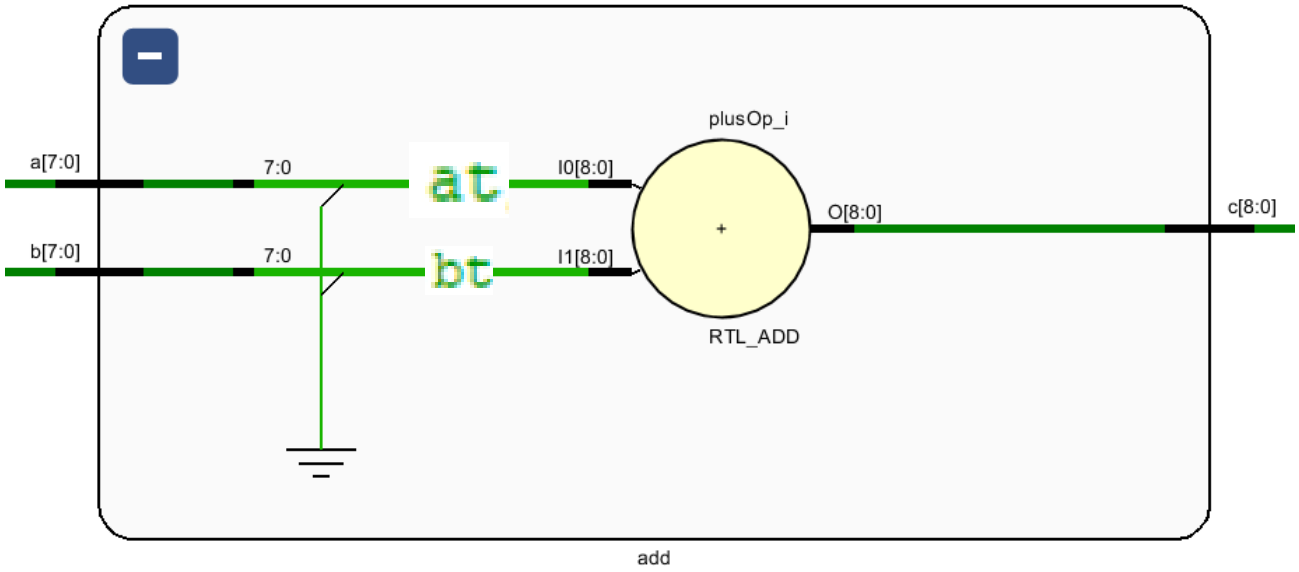
Add - komponent

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity add is
    Port ( a : in STD_LOGIC_VECTOR (7 downto 0);
           b : in STD_LOGIC_VECTOR (7 downto 0);
           c : out STD_LOGIC_VECTOR (8 downto 0));
end add;

architecture Behavioral of add is
   signal at,bt: STD_LOGIC_VECTOR (8 downto 0);
begin
    at <= '0'& a;
    bt <= '0'& b;
    c <= at + bt;
end Behavioral;
```

Bemærk! – når man lægger tal sammen med VHDL skal antallet af bit i alle tal og resultat være det samme.

Hvis man vil kunne rumme a+b skal resultatet være 1 bit større.

at (a-temp) og bt (b-temp) er lavet et bit større og værdierne overført med '0' &



Toplevel version 1, 2 og 3

```
ENTITY Standard_toplevel IS
    PORT ( -- clk: in  STD_LOGIC;
        --sw : in  STD_LOGIC_VECTOR (15 downto 0);
          a:   in  STD_LOGIC_VECTOR (7 downto 0);
          b:   in  STD_LOGIC_VECTOR (7 downto 0);
       --led:  out STD_LOGIC_VECTOR (15 downto 0);
          sum:  out STD_LOGIC_VECTOR (8 downto 0));
end Standard_toplevel;
```

```
ARCHITECTURE Version1 of Standard_toplevel is
   component add
       port (a : in std_logic_vector (7 downto 0);
             b : in std_logic_vector (7 downto 0);
             c : out std_logic_vector (8 downto 0));
   end component;

    signal anet : std_logic_vector (7 downto 0);
    signal bnet : std_logic_vector (7 downto 0);
BEGIN
   anet <= a;
   bnet <= b;

   U1 : add
        port map (a => anet,
                  b => bnet,
                  c => sum);
END Version1;
```

Ekstern label = lokalt net

Intern label (navn) i komponenten

Ekstern label

Intern label kobles til ekstern label

```
ARCHITECTURE Version2 of Standard_toplevel is
   component add
       port (a : in std_logic_vector (7 downto 0);
             b : in std_logic_vector (7 downto 0);
             c : out std_logic_vector (8 downto 0));
   end component;
BEGIN
   U2 : add
        port map (a => a,
                  b => b,
                  c => sum);
END Version2;
```

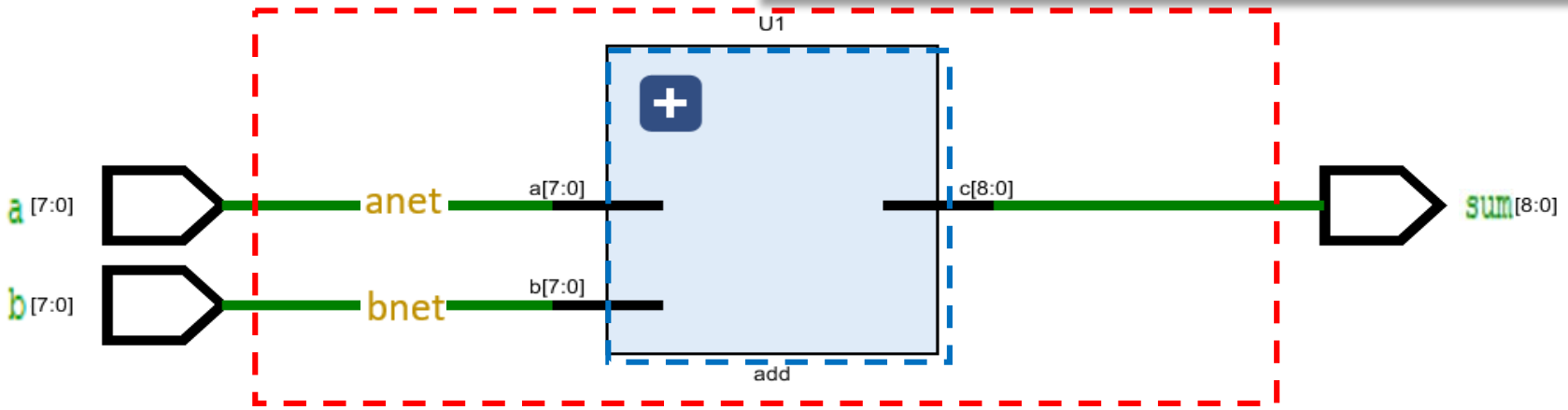
```
ARCHITECTURE Version3 of Standard_toplevel is
   component add
       port (a : in std_logic_vector (7 downto 0);
             b : in std_logic_vector (7 downto 0);
             c : out std_logic_vector (8 downto 0));
   end component;

    signal c : std_logic_vector (8 downto 0);
BEGIN
   sum <= c;
   U3 : add port map ( a, b, c);
END Version3;
```

Positionen er vigtig – frarådes at bruge

Version1

```
ENTITY Standard_toplevel IS
    PORT ( -- clk: in  STD_LOGIC;
        --sw : in  STD_LOGIC_VECTOR (15 downto 0);
          a:   in  STD_LOGIC_VECTOR (7 downto 0);
          b:   in  STD_LOGIC_VECTOR (7 downto 0);
        --led:  out STD_LOGIC_VECTOR (15 downto 0);
          sum:  out STD_LOGIC_VECTOR (8 downto 0));
end Standard_toplevel;
```

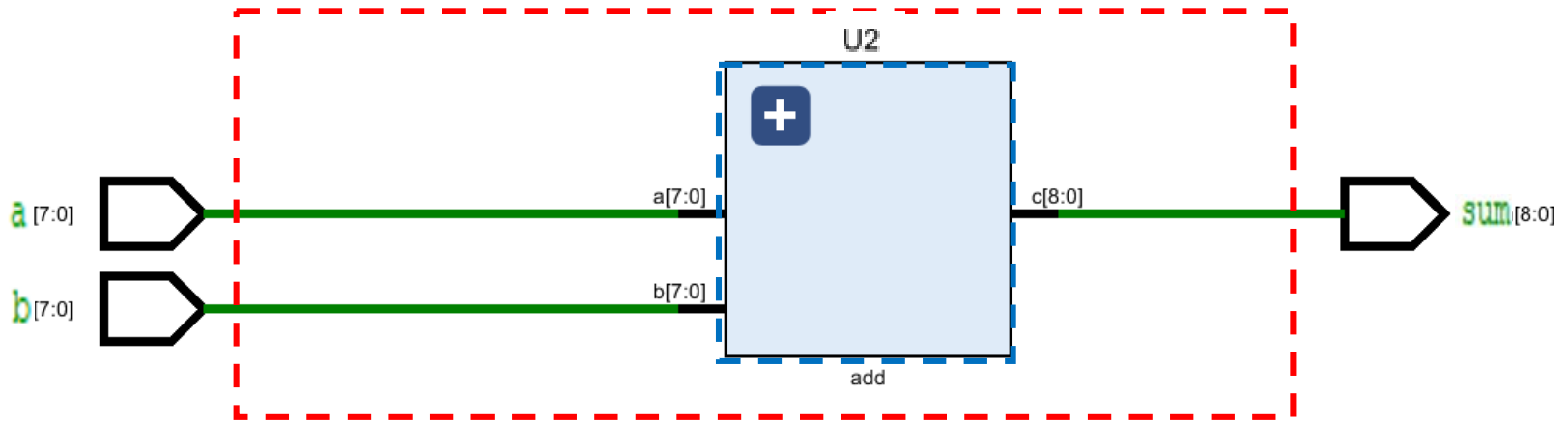
```
ARCHITECTURE Version1 of Standard_toplevel is
    component add
        port (a : in std_logic_vector (7 downto 0);
              b : in std_logic_vector (7 downto 0);
              c : out std_logic_vector (8 downto 0));
    end component;

    signal anet : std_logic_vector (7 downto 0);
    signal bnet : std_logic_vector (7 downto 0);
BEGIN
   anet <= a;
   bnet<= b;

   U1 : add
        port map (a =>anet,
                  b => bnet,
                  c => sum);
END Version1;
```



Version2

```
ENTITY Standard_toplevel IS
   PORT ( -- clk: in  STD_LOGIC;
        --sw : in  STD_LOGIC_VECTOR (15 downto 0);
          a:   in  STD_LOGIC_VECTOR (7 downto 0);
          b:   in  STD_LOGIC_VECTOR (7 downto 0);
        --led:  out STD_LOGIC_VECTOR (15 downto 0);
          sum:  out STD_LOGIC_VECTOR (8 downto 0));
end Standard_toplevel;
```

```
ARCHITECTURE Version2 of Standard_toplevel is
   component add
       port (a : in std_logic_vector (7 downto 0);
             b : in std_logic_vector (7 downto 0);
             c : out std_logic_vector (8 downto 0));
   end component;
BEGIN
   U2 : add
        port map (a => a,
                  b => b,
                  c => sum);
END Version2;
```



Version3

```
ENTITY Standard_toplevel IS
    PORT ( -- clk: in  STD_LOGIC;
        --sw : in  STD_LOGIC_VECTOR (15 downto 0);
          a:   in  STD_LOGIC_VECTOR (7 downto 0);
          b:   in  STD_LOGIC_VECTOR (7 downto 0);
        --led:  out STD_LOGIC_VECTOR (15 downto 0);
          sum:  out STD_LOGIC_VECTOR (8 downto 0));
end Standard_toplevel;
```

```
ARCHITECTURE Version3 of Standard_toplevel is
    component add
        port (a : in std_logic_vector (7 downto 0);
              b : in std_logic_vector (7 downto 0);
              c : out std_logic_vector (8 downto 0));
    end component;

    signal c : std_logic_vector (8 downto 0);
BEGIN
    sum <= c;

    U3 : add port map ( a, b, c);
END Version3;
```