Lösung 6 Arithmetik und Variablen

Jan Grapengeter

May 2, 2019

1 ieee.numeric_std.all Bibliothek

1.:
Vergleich abhängig von der Implementierung.
2.:
VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
            uhr : in std_logic;
            btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

Signal a,b,mult: std_logic_vector (7 downto 0);
Signal c: std_logic_vector (15 downto 0):="0000000000000000";
signal reset : std_logic :='0';

begin

a<=sw(15 downto 8);
b<=sw(7 downto 0);


process(uhr)
begin
reset<=btnd;
if(rising_edge(uhr)) then
if(reset = '1') then
reset <= '0';
c<="0000000000000000";
mult<=b;
elsif(mult /= "00000000") then
```

```
c<=c+a;
mult<=mult-1;
end if;
end if;
end process;

ld<=c;

end Behavioral;
```

Constraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports  sw[0]];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports  sw[1]];
set_property -dict { PACKAGE_PIN w16   IOSTANDARD LVCMOS33 } [get_ports  sw[2]];
set_property -dict { PACKAGE_PIN w17   IOSTANDARD LVCMOS33 } [get_ports  sw[3]];
set_property -dict { PACKAGE_PIN w15   IOSTANDARD LVCMOS33 } [get_ports  sw[4]];
set_property -dict { PACKAGE_PIN v15   IOSTANDARD LVCMOS33 } [get_ports  sw[5]];
set_property -dict { PACKAGE_PIN w14   IOSTANDARD LVCMOS33 } [get_ports  sw[6]];
set_property -dict { PACKAGE_PIN w13   IOSTANDARD LVCMOS33 } [get_ports  sw[7]];
set_property -dict { PACKAGE_PIN v2   IOSTANDARD LVCMOS33 } [get_ports  sw[8]];
set_property -dict { PACKAGE_PIN t3   IOSTANDARD LVCMOS33 } [get_ports  sw[9]];
set_property -dict { PACKAGE_PIN t2   IOSTANDARD LVCMOS33 } [get_ports  sw[10]];
set_property -dict { PACKAGE_PIN r3   IOSTANDARD LVCMOS33 } [get_ports  sw[11]];
set_property -dict { PACKAGE_PIN w2   IOSTANDARD LVCMOS33 } [get_ports  sw[12]];
set_property -dict { PACKAGE_PIN u1   IOSTANDARD LVCMOS33 } [get_ports  sw[13]];
set_property -dict { PACKAGE_PIN t1   IOSTANDARD LVCMOS33 } [get_ports  sw[14]];
set_property -dict { PACKAGE_PIN r2   IOSTANDARD LVCMOS33 } [get_ports  sw[15]];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports  ld[0]];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports  ld[1]];
set_property -dict { PACKAGE_PIN u19   IOSTANDARD LVCMOS33 } [get_ports  ld[2]];
set_property -dict { PACKAGE_PIN v19   IOSTANDARD LVCMOS33 } [get_ports  ld[3]];
set_property -dict { PACKAGE_PIN w18   IOSTANDARD LVCMOS33 } [get_ports  ld[4]];
set_property -dict { PACKAGE_PIN u15   IOSTANDARD LVCMOS33 } [get_ports  ld[5]];
set_property -dict { PACKAGE_PIN u14   IOSTANDARD LVCMOS33 } [get_ports  ld[6]];
set_property -dict { PACKAGE_PIN v14   IOSTANDARD LVCMOS33 } [get_ports  ld[7]];
set_property -dict { PACKAGE_PIN v13   IOSTANDARD LVCMOS33 } [get_ports  ld[8]];
set_property -dict { PACKAGE_PIN v3   IOSTANDARD LVCMOS33 } [get_ports  ld[9]];
set_property -dict { PACKAGE_PIN w3   IOSTANDARD LVCMOS33 } [get_ports  ld[10]];
set_property -dict { PACKAGE_PIN u3   IOSTANDARD LVCMOS33 } [get_ports  ld[11]];
set_property -dict { PACKAGE_PIN p3   IOSTANDARD LVCMOS33 } [get_ports  ld[12]];
set_property -dict { PACKAGE_PIN n3   IOSTANDARD LVCMOS33 } [get_ports  ld[13]];
set_property -dict { PACKAGE_PIN p1   IOSTANDARD LVCMOS33 } [get_ports  ld[14]];
set_property -dict { PACKAGE_PIN l1   IOSTANDARD LVCMOS33 } [get_ports  ld[15]];

set_property -dict { PACKAGE_PIN w5   IOSTANDARD LVCMOS33 } [get_ports  uhr];
set_property -dict { PACKAGE_PIN u17   IOSTANDARD LVCMOS33 } [get_ports  btnd];
```

2 Arithmetische Funktionen

1.:
VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
            uhr : in std_logic;
            btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

Signal a,b: std_logic_vector (15 downto 0):="0000000000000000";
signal g,h : unsigned (15 downto 0);

begin

a<=sw;
g<=unsigned(a);
h<=shift_left(g,1);
b<=std_logic_vector(h);
ld<=b;

end Behavioral;
```

2.:
VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
            uhr : in std_logic;
            btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

Signal a,b: std_logic_vector (7 downto 0);
Signal c,temp: std_logic_vector (15 downto 0):="0000000000000000";
```

```
signal reset : std_logic :='0';
signal f : unsigned (7 downto 0);
signal g : unsigned (15 downto 0);
signal d,e,g0,g1,g2,g3,g4,g5,g6,g7 : unsigned (15 downto 0);

begin

a<= sw(15 downto 8);
b<=sw(7 downto 0);

--d<=unsigned(a);
temp<= X"00" & a;
e<=unsigned(temp);
f<=unsigned(b);
d<="0000000000000000";

process
begin
if(f(0)='1') then
g0<=shift_left(e,0);
else
g0<=d;
end if;

if(f(1)='1') then
g1<=shift_left(e,1);
else
g1<=d;
end if;

if(f(2)='1') then
g2<=shift_left(e,2);
else
g2<=d;
end if;

if(f(3)='1') then
g3<=shift_left(e,3);
else
g3<=d;
end if;

if(f(4)='1') then
g4<=shift_left(e,4);
else
g4<=d;
end if;

if(f(5)='1') then
g5<=shift_left(e,5);
else
g5<=d;
```

```
end if;

if(f(6)='1') then
g6<=shift_left(e,6);
else
g6<=d;
end if;

if(f(7)='1') then
g7<=shift_left(e,7);
else
g7<=d;
end if;
end process;

ld<= std_logic_vector(g0+g1+g2+g3+g4+g5+g6+g7);

end Behavioral;
```

3.:
Shift-Module:

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

entity shift is
    Port ( vin : in Std_logic_vector(15 downto 0);
            shamt : in Std_logic_vector(3 downto 0);
           vout : out Std_logic_vector(15 downto 0));
end shift;

architecture Behavioral of shift is

signal vtemp1,vtemp2 : std_logic_vector (15 downto 0);

begin

process
begin
case (shamt) is
when "0000"=>
vtemp1<=vin;
when "0001"=>
vtemp1<=vin(14 downto 0) & '0';
when "0010"=>
vtemp1<=vin(13 downto 0) & "00";
when "0011"=>
vtemp1<=vin(12 downto 0) & "00";
when "0100"=>
vtemp1<=vin(11 downto 0) & "00";
when "0101"=>
vtemp1<=vin(10 downto 0) & "00";
```

```
when "0110"=>
vtemp1<=vin(9 downto 0) & "00";
when "0111"=>
vtemp1<=vin(8 downto 0) & "00";
when "1000"=>
vtemp1<=vin(7 downto 0) & "00";
when "1001"=>
vtemp1<=vin(6 downto 0) & "00";
when "1010"=>
vtemp1<=vin(5 downto 0) & "00";
when "1011"=>
vtemp1<=vin(4 downto 0) & "00";
when "1100"=>
vtemp1<=vin(3 downto 0) & "00";
when "1101"=>
vtemp1<=vin(2 downto 0) & "00";
when "1110"=>
vtemp1<=vin(1 downto 0) & "00";
when "1111"=>
vtemp1<="0000000000000000";
end case;
end process;

end Behavioral;
```

3 Arrays

VHDL_main.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

Signal temp: std_logic_vector (15 downto 0):="0000000000000000";
signal e,d : unsigned (15 downto 0);
signal i : natural:=7;
signal f : unsigned(7 downto 0);

type zahlvek is array (7 downto 0) of unsigned(15 downto 0);
signal gg : zahlvek;

begin
```

```
temp<= X"00" & sw(15 downto 8);
e<=unsigned(temp);
f<=unsigned(sw(7 downto 0));

process
begin

for i in 0 to 7 loop
if(f(i)='1') then
gg(i)<=shift_left(e,i);
else
gg(i)<=d;
end if;
end loop;
end process;

ld<= std_logic_vector(gg(0)+gg(1)+gg(2)+gg(3)+gg(4)+gg(5)+gg(6)+gg(7));

end Behavioral;
```

4 Variablen

VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
           uhr,btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

signal a,b,c : std_logic_vector(7 downto 0);

begin

process (btnd)
begin
if (rising_edge(btnd)) then
a <= sw(7 downto 0);
b <= a;
end if;
end process;

process (btnd)
variable var : std_logic_vector (7 downto 0);
```

```
begin
if (rising_edge(btnd)) then
var := sw(15 downto 8);
c <= var;
end if;
end process;

ld(15 downto 8)<=c;
ld(7 downto 0)<=b;

end Behavioral;
```

5 Vergleichsfunktionen

1.:
VHDL_main:

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

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
           uhr,btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

signal a,b,c : std_logic_vector(7 downto 0);

begin

process (uhr)
begin
if(rising_edge(uhr)) then
if(sw(15 downto 8)>sw(7 downto 0)) then
c<="00000000";
b<="11111111";
else
c<="11111111";
b<="00000000";
end if;
end if;
end process;

ld(15 downto 8)<=c;
ld(7 downto 0)<=b;

end Behavioral;
```

2.:

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

entity main is
    Port ( sw : in STD_LOGIC_vector(15 downto 0);
           uhr,btnd : in std_logic;
           ld :out STD_LOGIC_vector(15 downto 0));
end main;

architecture Behavioral of main is

signal a,b,c : std_logic_vector(7 downto 0);
signal vergleich : std_logic;

function vergleicher(vec1,vec2 : std_logic_vector(7 downto 0))
return std_logic is
variable temp : std_logic;
begin
for i in 7 downto 0 loop
if(vec1(i)='1' and vec2(i)='0') then
temp := '1';
exit;
elsif(vec1(i)='0' and vec2(i)='1') then
temp := '0';
exit;
else
temp := '0';
end if;
end loop;
return temp;
end vergleicher;


begin

a<=sw(15 downto 8);
b<=sw(7 downto 0);

ld(0)<=vergleicher(b,a);

end Behavioral;
```

Anmerkung: Funktionen sind nicht sequentiell, sequentielle Lösungen wären hier also falsch.