Lösung 3 Unterfunktionen, Signale und std_logic_vector

Jan Grapengeter

May 2, 2019

1 Halbaddierer

VHDL_main:

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

entity main is
    Port ( sw0 : in STD_LOGIC;
           sw1 : in STD_LOGIC;

           ld0 : out STD_LOGIC;
           ld1 : out STD_LOGIC);
end main;

architecture Behavioral of main is

begin

ld0 <= sw0 and sw1;
ld1 <= sw0 xor sw1;

end Behavioral;
```

Constraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { sw1}];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { ld0}];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports { ld1}];
```

1.0.1 Volladdierer

VHDL_main.

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

entity main is
```

```
    Port ( sw0 : in STD_LOGIC;
           sw1 : in STD_LOGIC;
           sw15 : in STD_LOGIC;

           ld0 : out STD_LOGIC;
           ld1 : out STD_LOGIC);
end main;

architecture Behavioral of main is
begin

Summe <= (not cin and not A1 and A2) or (not cin and A1 and not A2)
or (cin and not A1 and not A2) or (cin and A1 and A2);
Ubertrag <= (not cin and A1 and A2) or (cin and not A1 and A2)
or (cin and A1 and not A2) or (cin and A1 and A2);

end Behavioral;
```

Contraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { sw1}];
set_property -dict { PACKAGE_PIN r2   IOSTANDARD LVCMOS33 } [get_ports { sw15}];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { ld0}];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports { ld1}];
```

2 Entity und Component

VHDL_main:

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

entity main is
    Port ( sw0 : in STD_LOGIC;
           sw1 : in STD_LOGIC;
           sw15 : in STD_LOGIC;

           ld0 : out STD_LOGIC;
           ld1 : out STD_LOGIC);
end main;

architecture Behavioral of main is

component volladdierer is
port(A1 : in STD_LOGIC;
     A2 : in STD_LOGIC;
     cin : in std_logic;
     Ubertrag : out STD_LOGIC;
     Summe : out STD_LOGIC);
```

```
end component;

begin

voll1:volladdierer port map(A1=>sw0,A2=>sw1,cin=>sw15,Ubertrag=>ld0,Summe=>ld1);

end Behavioral;
```

Contraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { sw1}];
set_property -dict { PACKAGE_PIN r2   IOSTANDARD LVCMOS33 } [get_ports { sw15}];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { ld0}];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports { ld1}];
```

Volladdierer:

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

entity volladdierer is
    Port ( A1 : in STD_LOGIC;
           A2 : in STD_LOGIC;
           cin : in STD_LOGIC;
           Ubertrag : out STD_LOGIC;
           Summe : out STD_LOGIC);
end volladdierer;

architecture Behavioral of volladdierer is

Summe <= (not cin and not A1 and A2) or (not cin and A1 and not A2)
or (cin and not A1 and not A2) or (cin and A1 and A2);
Ubertrag <= (not cin and A1 and A2) or (cin and not A1 and A2)
or (cin and A1 and not A2) or (cin and A1 and A2);

end Behavioral;
```

3 Dateiausgliederung

1., 2., 3.:
Die Lösung für den Code ist identisch mit dem aus der vorherigen Aufgabe, jedoch ist die Projektstruktur jetzt anders.
4.:
Die Definition eines Moduls muss im Code vor der Instanzierung des Moduls stehen.

3.0.1 Signale

VHDL_main:

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

entity main is
    Port ( sw0 : in STD_LOGIC;
           sw1 : in STD_LOGIC;
           sw15 : in STD_LOGIC;

           ld0 : out STD_LOGIC;
           ld1 : out STD_LOGIC);
end main;

architecture Behavioral of main is

component volladdierer is
port(A1 : in STD_LOGIC;
     A2 : in STD_LOGIC;
     cin : in std_logic;
     Ubertrag : out STD_LOGIC;
     Summe : out STD_LOGIC);
end component;

begin

voll1:volladdierer port map(A1=>sw0,A2=>sw1,cin=>sw15,Ubertrag=>ld0,Summe=>ld1);

end Behavioral;
```

Contraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { sw1}];
set_property -dict { PACKAGE_PIN r2   IOSTANDARD LVCMOS33 } [get_ports { sw15}];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { ld0}];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports { ld1}];
```

Volladdierer:

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

entity volladdierer is
    Port ( A1 : in STD_LOGIC;
           A2 : in STD_LOGIC;
           cin : in STD_LOGIC;
           Ubertrag : out STD_LOGIC;
           Summe : out STD_LOGIC);
end volladdierer;

architecture Behavioral of volladdierer is
```

```
signal wire1,wire2,wire3 : std_logic;

begin
wire1 <= A1 xor A2;
wire2 <= wire1 and cin;
wire3 <= A1 and A2;

Summe <= wire1 xor cin;
Ubertrag <= wire2 or wire3;

end Behavioral;
```

4 Verbindung von Komponenten

VHDL_main:

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

entity main is
    Port ( sw0 : in STD_LOGIC;
           sw1 : in STD_LOGIC;
           sw2 : in STD_LOGIC;
           sw3 : in STD_LOGIC;
           sw4 : in STD_LOGIC;
           sw5 : in STD_LOGIC;
           sw6 : in STD_LOGIC;
           sw7 : in STD_LOGIC;
           sw8 : in STD_LOGIC;
           sw9 : in STD_LOGIC;
           sw10 : in STD_LOGIC;
           sw11 : in STD_LOGIC;
           sw12 : in STD_LOGIC;
           sw13 : in STD_LOGIC;
           sw14 : in STD_LOGIC;
           sw15 : in STD_LOGIC;

           ld0 : out STD_LOGIC;
           ld1 : out STD_LOGIC;
           ld2 : out STD_LOGIC;
           ld3 : out STD_LOGIC;
           ld4 : out STD_LOGIC;
           ld5 : out STD_LOGIC;
           ld6 : out STD_LOGIC;
           ld7 : out STD_LOGIC;
           ld8 : out STD_LOGIC;
           ld9 : out STD_LOGIC;
           ld10 : out STD_LOGIC;
           ld11 : out STD_LOGIC;
           ld12 : out STD_LOGIC;
           ld13 : out STD_LOGIC;
           ld14 : out STD_LOGIC);
end main;



architecture Behavioral of main is

component volladdierer is
port(A1 : in STD_LOGIC;
     A2 : in STD_LOGIC;
     cin : in std_logic;
     Ubertrag : out STD_LOGIC;
     Summe : out STD_LOGIC);
end component;

signal a1,b1,carry,u1,s1 : std_logic;
signal a2,b2,u2,s2 : std_logic;
signal a3,b3,u3,s3 : std_logic;
signal a4,b4,u4,s4 : std_logic;

begin

carry<=sw0;
a1<=sw1;
a2<=sw2;
a3<=sw3;
a4<=sw4;
b1<=sw5;
b2<=sw6;
b3<=sw7;
b4<=sw8;

voll1:volladdierer port map(A1=>a1,A2=>b1,cin=>carry,Ubertrag=>u1,Summe=>s1);
voll2:volladdierer port map(A1=>a2,A2=>b2,cin=>u1,Ubertrag=>u2,Summe=>s2);
voll3:volladdierer port map(A1=>a3,A2=>b3,cin=>u2,Ubertrag=>u3,Summe=>s3);
voll4:volladdierer port map(A1=>a4,A2=>b4,cin=>u3,Ubertrag=>u4,Summe=>s4);

ld1<=s1;
ld2<=s2;
ld3<=s3;
ld4<=s4;
ld5<=u4;

end Behavioral;
```

Constraints:

```
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { sw1}];
set_property -dict { PACKAGE_PIN w16   IOSTANDARD LVCMOS33 } [get_ports { sw2}];
set_property -dict { PACKAGE_PIN w17   IOSTANDARD LVCMOS33 } [get_ports { sw3}];
set_property -dict { PACKAGE_PIN w15   IOSTANDARD LVCMOS33 } [get_ports { sw4}];
set_property -dict { PACKAGE_PIN v15   IOSTANDARD LVCMOS33 } [get_ports { sw5}];
set_property -dict { PACKAGE_PIN w14   IOSTANDARD LVCMOS33 } [get_ports { sw6}];
set_property -dict { PACKAGE_PIN w13   IOSTANDARD LVCMOS33 } [get_ports { sw7}];
set_property -dict { PACKAGE_PIN v2   IOSTANDARD LVCMOS33 } [get_ports { sw8}];
set_property -dict { PACKAGE_PIN t3   IOSTANDARD LVCMOS33 } [get_ports { sw9}];
set_property -dict { PACKAGE_PIN t2   IOSTANDARD LVCMOS33 } [get_ports { sw10}];
set_property -dict { PACKAGE_PIN r3   IOSTANDARD LVCMOS33 } [get_ports { sw11}];
set_property -dict { PACKAGE_PIN w2   IOSTANDARD LVCMOS33 } [get_ports { sw12}];
set_property -dict { PACKAGE_PIN u1   IOSTANDARD LVCMOS33 } [get_ports { sw13}];
set_property -dict { PACKAGE_PIN t1   IOSTANDARD LVCMOS33 } [get_ports { sw14}];
set_property -dict { PACKAGE_PIN r2   IOSTANDARD LVCMOS33 } [get_ports { sw15}];

set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { ld0}];
set_property -dict { PACKAGE_PIN e19   IOSTANDARD LVCMOS33 } [get_ports { ld1}];
set_property -dict { PACKAGE_PIN u19   IOSTANDARD LVCMOS33 } [get_ports { ld2}];
set_property -dict { PACKAGE_PIN v19   IOSTANDARD LVCMOS33 } [get_ports { ld3}];
set_property -dict { PACKAGE_PIN w18   IOSTANDARD LVCMOS33 } [get_ports { ld4}];
set_property -dict { PACKAGE_PIN u15   IOSTANDARD LVCMOS33 } [get_ports { ld5}];
set_property -dict { PACKAGE_PIN u14   IOSTANDARD LVCMOS33 } [get_ports { ld6}];
set_property -dict { PACKAGE_PIN v14   IOSTANDARD LVCMOS33 } [get_ports { ld7}];
set_property -dict { PACKAGE_PIN v13   IOSTANDARD LVCMOS33 } [get_ports { ld8}];
set_property -dict { PACKAGE_PIN v3   IOSTANDARD LVCMOS33 } [get_ports { ld9}];
set_property -dict { PACKAGE_PIN w3   IOSTANDARD LVCMOS33 } [get_ports { ld10}];
set_property -dict { PACKAGE_PIN u3   IOSTANDARD LVCMOS33 } [get_ports { ld11}];
set_property -dict { PACKAGE_PIN p3   IOSTANDARD LVCMOS33 } [get_ports { ld12}];
set_property -dict { PACKAGE_PIN n3   IOSTANDARD LVCMOS33 } [get_ports { ld13}];
set_property -dict { PACKAGE_PIN p1   IOSTANDARD LVCMOS33 } [get_ports { ld14}];
```

3.:
sw0,sw1,sw5,ld1 = 1
sw2,sw6,ld2 = 1
sw3,sw7,ld3 = 1
sw4,sw8,ld4 = 1
4.: Maximal 7 bit, limitierendes Kriterium ist die Anzahl der Schalter.

5 std_logic_vector

VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.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

component volladdierer is
port(A1 : in STD_LOGIC;
     A2 : in STD_LOGIC;
     cin : in std_logic;
     Ubertrag : out STD_LOGIC;
     Summe : out STD_LOGIC);
end component;

Signal a,b,cin,u,s : std_logic_vector (3 downto 0);

begin

a(0)<=sw(1);
a(1)<=sw(2);
a(2)<=sw(3);
a(3)<=sw(4);
b(0)<=sw(5);
b(1)<=sw(6);
b(2)<=sw(7);
b(3)<=sw(8);

voll1:volladdierer port map(A1=>a(0),A2=>b(0),cin=>sw(0),Ubertrag=>u(0),Summe=>s(0));
voll2:volladdierer port map(A1=>a(1),A2=>b(1),cin=>u(0),Ubertrag=>u(1),Summe=>s(1));
voll3:volladdierer port map(A1=>a(2),A2=>b(2),cin=>u(1),Ubertrag=>u(2),Summe=>s(2));
voll4:volladdierer port map(A1=>a(3),A2=>b(3),cin=>u(2),Ubertrag=>u(3),Summe=>s(3));

ld(1)<=s(0);
ld(2)<=s(1);
ld(3)<=s(2);
ld(4)<=s(3);
ld(5)<=u(3);
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]];
```

Die Contraints Datei wird erst wieder neu eingetragen, wenn sich am Code etwas ändert.

6 Generate Statement

VHDL_main:

```
library IEEE;
use IEEE.STD_LOGIC_1164.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

component volladdierer is
port(A1 : in STD_LOGIC;
     A2 : in STD_LOGIC;
     cin : in std_logic;
     Ubertrag : out STD_LOGIC;
     Summe : out STD_LOGIC);
end component;

Signal a,b,cin,u,s : std_logic_vector (3 downto 0);

begin

a(0)<=sw(1);
a(1)<=sw(2);
a(2)<=sw(3);
a(3)<=sw(4);
b(0)<=sw(5);
b(1)<=sw(6);
b(2)<=sw(7);
b(3)<=sw(8);
```

```
GENs_voll:
for i in 0 to 3 generate
vollex : volladdierer port map
(A1=>a(i),A2=>b(i),cin=>cin(i),Ubertrag=>u(i),Summe=>s(i));
end generate GENs_voll;

cin(3)<=u(2);
cin(2)<=u(1);
cin(1)<=u(0);
cin(0)<=sw(0);

ld(1)<=s(0);
ld(2)<=s(1);
ld(3)<=s(2);
ld(4)<=s(3);
ld(5)<=u(3);

end Behavioral;
```