

EN: Please note the materials provided to you at the end of the test!

Aufgabe 1 von 3 (6 Punkte)

```
signal wort1 : std_logic_vector(3 downto 0) := "1010";  
signal wort2 : std_logic_vector(3 downto 0) := "1110";  
signal wort3 : std_logic_vector(3 downto 0) := "0011";  
signal wort4 : std_logic_vector(7 downto 0);  
signal wort5 : std_logic_vector(7 downto 0);  
signal wort6 : std_logic_vector(7 downto 0);
```

Vordefinierte Bit-Wörter / predefined bit-words

Durch geschickten Einsatz von Shift- und Verbindungsoperatoren und OHNE EINSATZ KONSTANTER WERTE WIE "00" oder '1', sollen aus den obigen 4-Bit Wörtern wort1, wort2 wort3 die folgenden 8-Bit Wörter wort4, wort5, wort6 gebildet werden:

wort4 soll sein / shall be: "10101110"
wort5 soll sein / shall be: "01100110"
wort6 soll sein / shall be: "00100010"

Bitte geben Sie also nachfolgend an, wie aus wort1,2,3 wort4,5,6 gebildet werden kann:

EN: So please write down below how word4,5,6 can be formed from word1,2,3:

wort4<=

//HINWEIS / HINT:

//Bitte alle Leerzeichen weglassen. Please leave off all blanks.

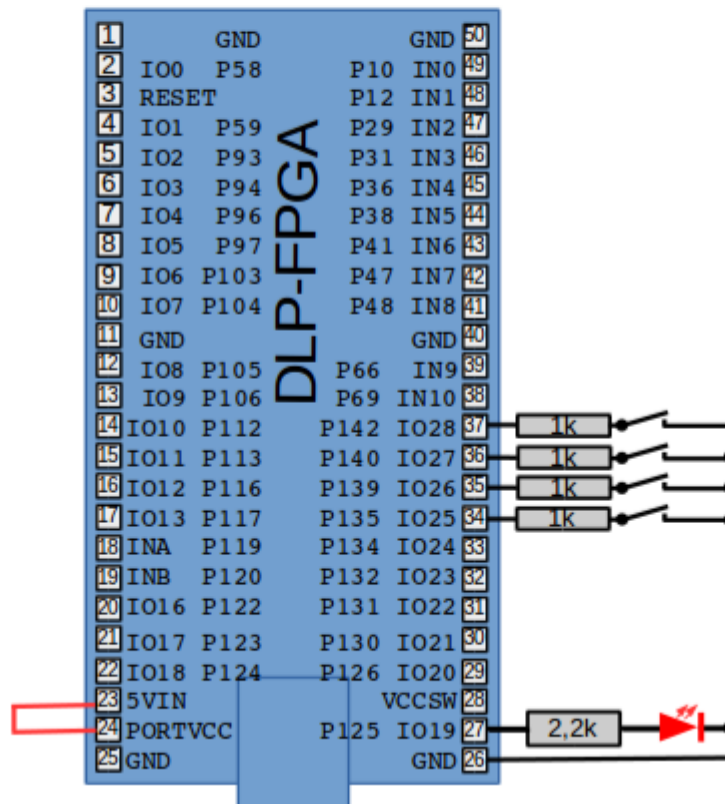
wort5<=

wort6<=

Aufgabe 2 von 3 (8 Punkte)

Gegeben seien der folgende Schaltplan und die dazu gehörige ucf-Datei:

EN: The following schematic and the corresponding ucf file are given:



```
NET "AA" LOC = "P135" | PULLUP | IOSTANDARD = LVCMOS33 ;
NET "BB" LOC = "P139" | PULLUP | IOSTANDARD = LVCMOS33 ;
NET "CC" LOC = "P140" | PULLUP | IOSTANDARD = LVCMOS33 ;
NET "DD" LOC = "P142" | PULLUP | IOSTANDARD = LVCMOS33 ;
NET "EE" LOC = "P125" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
```

Wird mindestens eine Taste gedrückt, soll die LED leuchten. Wird gar keine Taste gedrückt, soll die LED nicht leuchten.

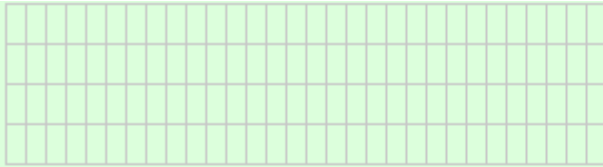
Vervollständigen Sie passend zu dieser Anforderung nachfolgende VHDL-Datei.

EN: If at least one key is pressed, the LED should light up. If no button is pressed at all, the LED should not light up.

Complete the following VHDL file in accordance with this requirement.

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

entity mustervergleich is
  Port (
```



```
end mustervergleich;

architecture Behavioral of mustervergleich is
begin

end Behavioral;
```

Aufgabe 3 von 3 (4 Punkte)

Schauen Sie sich die Materialien zu LUT4 weiter unten an. Es soll mit LUT4 ein UND-Gatter mit vier Eingängen realisiert werden. Wie muß dann INIT der generic map definiert werden? Tragen Sie dies nachfolgend ein:

```
-- Bitte Backslashes ignorieren.
-- Please ignore backslashes.

INIT=>    \"  \";
```

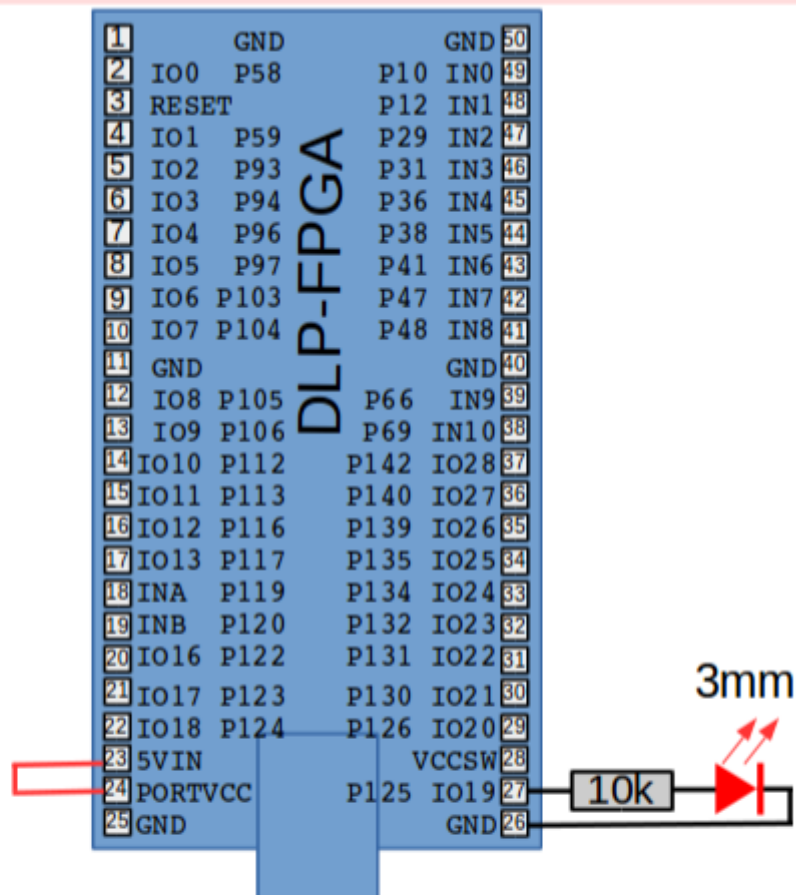
Hilfreiche Materialien aus der Vorlesung / Helpful materials from lectures

Operatoren

Operator	Bedeutung
and	Logisch UND
or	Logisch ODER
not	Logisch NICHT
xor	Logisch ANTIVALENZ (ungleich)
nxor	Logisch ÄQUIVALENZ (gleich)
nand	Logisch NICHT-UND
nor	Logisch NICHT-ODER

Tabelle 11-1: Logische Operatoren

BLINK



```
## GCLK6 ist auf FPGA-Chip Pin 56
NET "MEINECLOCK" LOC = "P56";
## IO L05P_0 ist auf FPGA-Chip Pin 125
NET "MEINELED" LOC = "P125";

NET "MEINECLOCK" IOSTANDARD = LVCMOS25;
NET "MEINELED" IOSTANDARD = LVCMOS33;
```

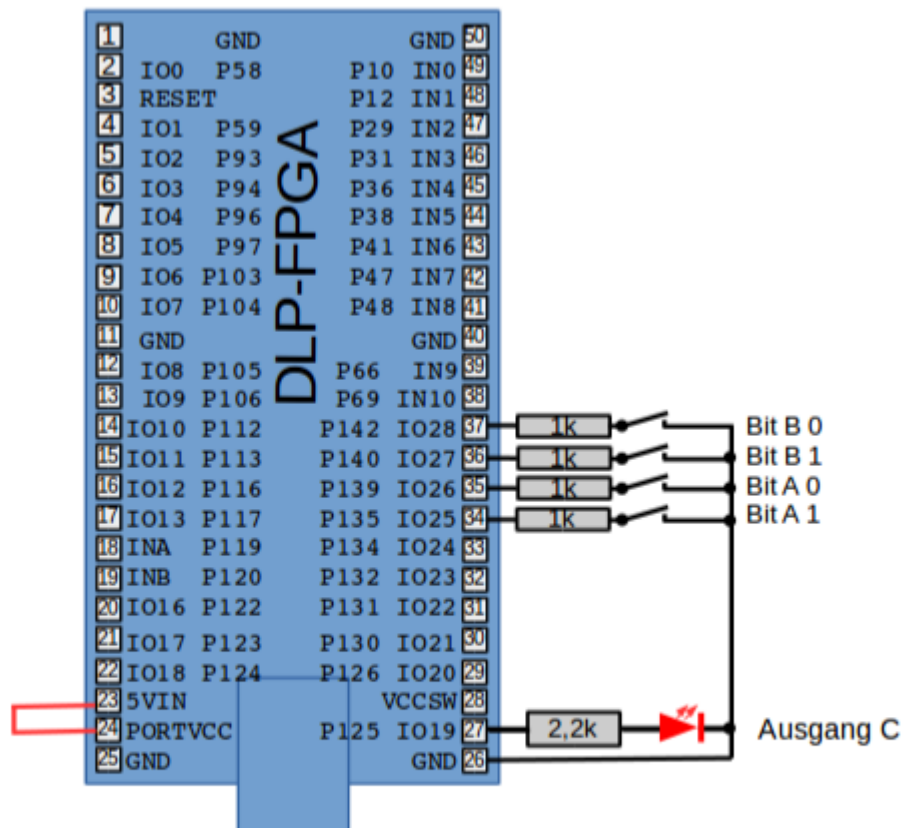
```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity blinkled is
  Port ( MEINECLOCK : in  STD_LOGIC;
        MEINELED   : out STD_LOGIC);
end blinkled;
-- Clock-In == 6MHz, Toggeln nach 3000000 Schritten ergibt 1Hz Blinkfrequenz.
architecture Behavioral of blinkled is
  signal zaehler      : integer range 0 to 2999999 := 0;
  signal logikpegel   : std_logic := '0';
begin
  process begin
    wait until rising_edge(MEINECLOCK);
    if (zaehler<2999999) then
      zaehler <= zaehler+1;
    else
      zaehler <= 0;
      logikpegel <= not logikpegel;
    end if;
  end process;
  MEINELED <= logikpegel;
end Behavioral;

```

MUSTERVERGLEICH



NET "A1"	LOC = "P135"	PULLUP	IOSTANDARD = LVCMOS33 ;
NET "A0"	LOC = "P139"	PULLUP	IOSTANDARD = LVCMOS33 ;
NET "B1"	LOC = "P140"	PULLUP	IOSTANDARD = LVCMOS33 ;
NET "B0"	LOC = "P142"	PULLUP	IOSTANDARD = LVCMOS33 ;
NET "C"	LOC = "P125"	IOSTANDARD = LVCMOS33	SLEW = SLOW ;

```

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

entity mustervergleich is
  Port ( A1 : in  STD_LOGIC;
        A0 : in  STD_LOGIC;
        B1 : in  STD_LOGIC;
        B0 : in  STD_LOGIC;
        C : out STD_LOGIC);
end mustervergleich;

architecture Behavioral of mustervergleich is
begin
  C <= (A1 xnor B1) and (A0 xnor B0);
end Behavioral;

```

LUT4

Logic Table

Inputs				Outputs
I3	I2	I1	I0	O
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]

```

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

library UNISIM;
use UNISIM.VComponents.all;

entity mustervergleich is
  Port ( A1 : in  STD_LOGIC;
        A0 : in  STD_LOGIC;
        B1 : in  STD_LOGIC;
        B0 : in  STD_LOGIC;
        C : out  STD_LOGIC);
end mustervergleich;

architecture Behavioral of mustervergleich is
begin
  LUT4_instanz : LUT4
    generic map ( INIT => "1000010000100001" )
    port map ( 0 => C, I0 => A0, I1 => A1, I2 => B0, I3 => B1);
end Behavioral;

```

ENDE