```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    14:53:11 01/23/2009
6  -- Design Name:
7  -- Module Name:    toplevel - Behavioral
8  -- Project Name:   FPGA-VHDL-Training
9  -- Target Devices: Spartan 3A DSP 1800 Board
10 -- Tool versions:
11 -- Description:    Hello World mit dem FPGA
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 ----------------------------------------------------------------------------------
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 --use IEEE.STD_LOGIC_ARITH.ALL;
23 --use IEEE.STD_LOGIC_UNSIGNED.ALL;
24 use IEEE.NUMERIC_STD.all;
25
26 ---- Uncomment the following library declaration if instantiating
27 ---- any Xilinx primitives in this code.
28 --library UNISIM;
29 --use UNISIM.VComponents.all;
30
31 entity toplevel is
32    Port (   LEDs_out             :        out   Std_Logic_Vector (7 downto 0);
33             DIP_Switches         :        in    Std_Logic_Vector (7 downto 0);
34
35             reset                :        in    Std_Logic;  --  (SW5)
36             RS232_RX             :        in    Std_Logic;  -- N21
37             RS232_TX             :        out   Std_Logic;  -- P22
38
39             CLK                  :        in    Std_Logic);
40 end toplevel;
41
42 architecture Behavioral of toplevel is
43
44 signal LEDs_in                  :        Std_Logic_Vector (7 downto 0);
45 signal count                    :        integer range 0 to (125000000/2-1);
46 signal frequenz_zaehler         :        integer range 0 to (125000000/2-1);
47
48 -- für die Schleife der Subtraktion
49 signal x                        :        integer range 0 to (125000000);
50 signal y                        :        integer range 0 to (125000000);
51 signal A                        :        integer range 0 to (125000000);
52 signal B                        :        integer range 0 to (125000000);
53
54 -- signalisiert ob die Schleife der Subtraktionen fertig ist
55 signal sub_fertig               :        STD_LOGIC := '0';
56
57 -- um  die DIP_Switches in INTEGER zu wandeln muss der umweg über SIGNED genommen werden
58 -- da eine direkte Umwandlung nicht möglich ist
59 signal int                      :        integer range 0 to 120000000;
60
61 COMPONENT Taktteiler
```

```
62          PORT(
63             CLK_tt               : IN std_logic;
64             Takt_in_tt           : IN integer range 0 to (125000000/2-1);
65             x_tt                 : IN integer range 0 to (125000000);
66             y_tt                 : IN integer range 0 to (125000000);
67             A_tt                 : IN integer range 0 to (125000000);
68             B_tt                 : IN integer range 0 to (125000000);
69             sub_fertig_tt        : INOUT std_logic;
70             frequenz_zaehler_tt  : OUT integer range 0 to (125000000/2-1));
71          END COMPONENT;
72
73          begin
74
75          Inst_Taktteiler: Taktteiler PORT MAP(
76                CLK_tt               => CLK,
77                Takt_in_tt           => B,
78                x_tt                 => x,
79                y_tt                 => y,
80                A_tt                 => A,
81                B_tt                 => B,
82                frequenz_zaehler_tt  => frequenz_zaehler,
83                sub_fertig_tt        => sub_fertig);
84
85          process(reset,CLK)
86          begin
87             if rising_edge(CLK) then
88                if sub_fertig = '0' then
89                   -- Aufruf von Taktteiler ???
90                elsif sub_fertig = '1' then
91                   if count < frequenz_zaehler then
92                      count <= count + 1;
93                   else
94                      LEDs_in <= LEDs_in(6 downto 0) & LEDs_in(7);     -- gegen den Uhrzeiger
95          --          LEDs_in <= LEDs_in(0) & LEDs_in(7 downto 1);     -- mit dem Uhrzeiger
96                      count <= 0;
97                   end if;
98                end if;
99                if reset = '1' then                                    -- RESET über Switch SW5
100                  count <= 0;
101                  LEDs_in <= X"18";
102                  sub_fertig <= '0';
103                  A <= 125000000;                                     --Clock 125MHz
104                  int <= to_integer(signed(DIP_Switches));            -- int ist der Wert der
          gewandelten DIP_Switches
105                  if int = 0 then                                     -- wenn alle DIP_Switches 0
          sind soll die Frequenz 256 ausgewählt sein
106                     int <= 256;
107                  end if;
108                  B <= int * 2;
109                  y <= 0;                                             --Schleifenzähler
110                  x <= A;                                             --Abbruchkriterium für die
          Schleife (x>=A)
111               end if;
112            end if;
113         end process;
114
115         LEDs_out <= LEDs_in;
116         end Behavioral;
```