RC-Filter im FPGA von Lothar Miller

15. Mai 2019

http://www.lothar-miller.de/s9y/categories/1-VHD

Hier die VHDL-Version des einfachsten Filters, das wie ein RC-Glied die Eingangswerte gewichtet aufsummiert. Der Implementierungsaufwand ist mit einem einzigen Summenregister überaus simpel im Vergleich zu einem Filter, das den gleitenden Mittelwert bildet und daher Speicher für n zurückliegende Eingangswerte braucht.

Der Filters funktioniert im Grunde wie ein RC-Glied: pro Takt wird vom Summenregister der Eingangswert der alte Mittelwert abgezogen und der neue Eingangswert aufaddiert. Das Summenregister entspricht dem Kondensator, der die Eingangsspannung ßammeltünd aufintegriert. Die Filterlänge stellt quasi den Widerstand des RC-Gliedes dar. Wenn die Filterlänge 1 ist, dann wird der Eingangswert "geradeausäuf den Ausgang durchgereicht. Der Widerstand hat in diesem Fall also 0 Ohm. Je länger der Filter ist, um so weniger wirksam ist der aktuelle Eingangswert, und um so hochohmiger ist der Widerstand im RC-Glied. Bei einer Filterlänge von 2**4 = 16 wirkt der neue Eingangswert nur zu 1/16 auf den Mittelwert.

Und genauso einfach wie ein RC-Glied ist dessen VHDL-Beschreibung:

```
0
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity filter is
6    generic(
7      data_width   : positive := 8;  -- Filterlaenge ist 2**
       filter_bits
8      filter_bits  : positive := 4); --> z.B. filter_bits = 4
9    port(                           --> Filterlaenge = 16
10     clk   : in  std_logic;         --> Gewichtung eines neuen
       Wertes = 1/16
11     inp   : in  std_logic_vector(data_width-1 downto 0);
12     outp  : out std_logic_vector(data_width-1 downto 0));
13 end entity filter;
14
15 architecture behavioral of filter is
16
17 -- das Summenregister entspricht dem Kondensator des RC-Glieds
18 signal  sum : unsigned(data_width+filter_bits-1 downto 0) := (
       others=>'0');
19
20 -- Mittelwert = hoechstwertige data_width Bits der Summe
21 alias   meanvalue : unsigned(data_width-1 downto 0) is sum(sum'left
       downto filter_bits);
```

```
22
23 begin
24
25   -- mit jedem Takt von der Summe den alten Mittelwert abziehen und
        neuen Wert aufaddieren
26   sum  <= sum - meanvalue + unsigned(inp) when rising_edge(clk);
27
28   -- Mittelwert (= linke data_width Bits) ausgeben
29   outp <= std_logic_vector(meanvalue);
30
31 end architecture behavioral;
```

Listing 1: FIFO instance from Xilinx macro in `buffer_fifo.vhd`.

ende