Reconfigurable Systems

UAlg

UNIVERSIDADE DO ALGARVE

Jorge Semião

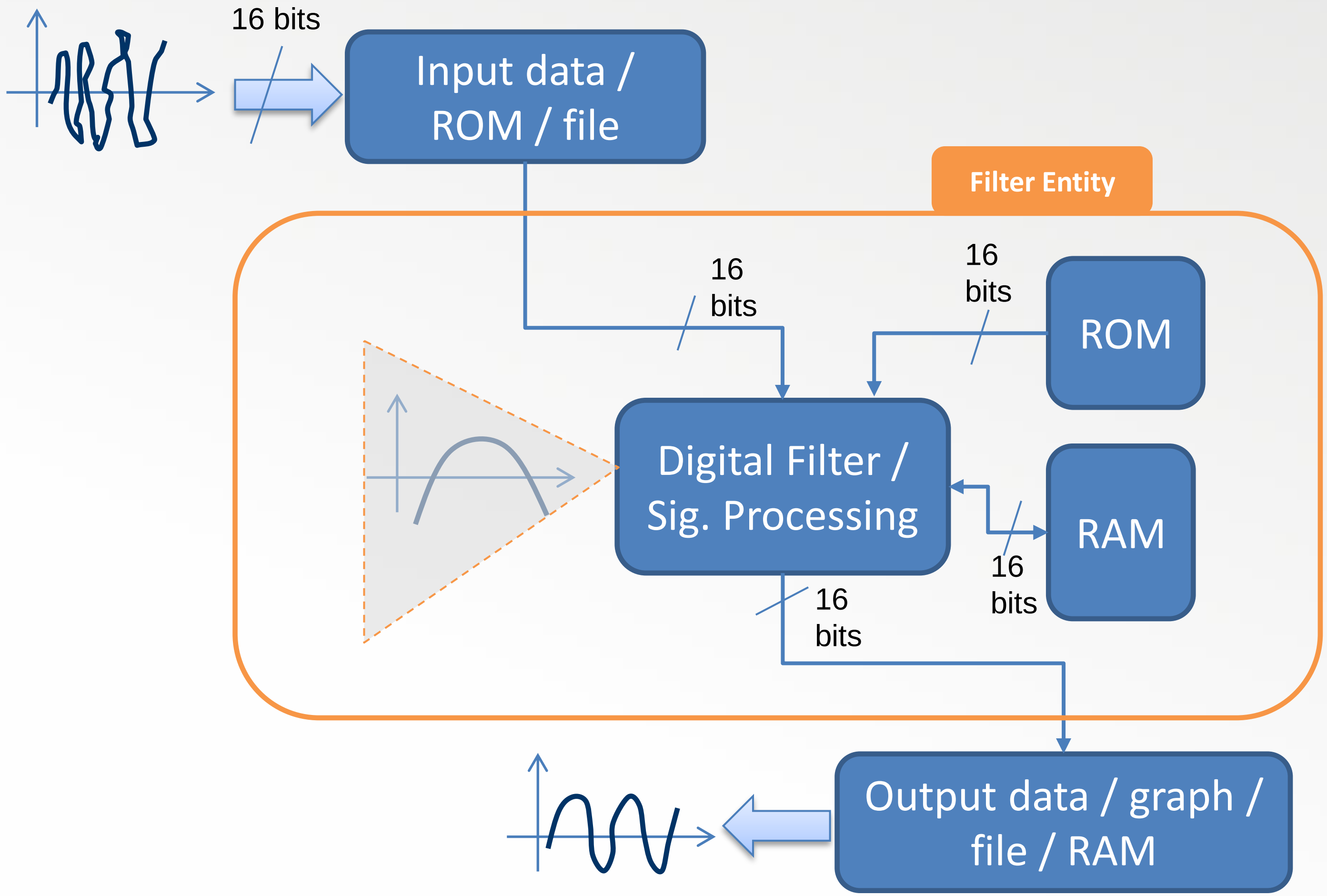
Instituto Superior de Engenharia

www.ualg.pt

Assignment 3

- The purpose of this project is to implement a digital FIR filter in VHDL and test it with digital data
- The digital filter should be either a Low-pass or a Band-pass filter
- The filter can be defined using a web application tool available (example: http://t-filter.engineerjs.com/)
- An entity for the filter should be designed, while the input and output data can be defined in the testbench

Assignment 3 - schematic



Assignment 3 – input and output data

- The input data can be provided to the filter either in a text file, or in a ROM memory/array
- The output data can be written/stored in a text file, or in a RAM memory/array, or both, however the output data stored in a text file will allow a graph generation to easily verify the filter correct operation
- Examples of a text file reading/writing is provided in the next slide
- An excel sheet should be used to help data conversion/verification

Example for Read and Write a file in Modelsim

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use STD.textio.all;
use ieee.std_logic_textio.all;

entity example_file_io_tb is
end example_file_io_tb;

architecture behave of example_file_io_tb is
file file_VECTORS : text;
file file_RESULTS : text;
signal r_ADD_TERM1 : std_logic_vector(3 downto 0) := (others => '0');
signal r_ADD_TERM2 : std_logic_vector(3 downto 0) := (others => '0');
signal w_SUM       : std_logic_vector(3 downto 0);
begin
process
    variable v_ILINE     : line;
    variable v_OLINE     : line;
    variable v_ADD_TERM1 : std_logic_vector(3 downto 0);
    variable v_ADD_TERM2 : std_logic_vector(3 downto 0);
    variable v_SPACE     : character;
 begin
    file_open(file_VECTORS, "input_vectors.txt",  read_mode);
    file_open(file_RESULTS, "output_results.txt", write_mode);
    while not endfile(file_VECTORS) loop
      readline(file_VECTORS, v_ILINE);
      read(v_ILINE, v_ADD_TERM1);
      read(v_ILINE, v_SPACE);           -- read in the space character
      read(v_ILINE, v_ADD_TERM2);
      -- Pass the variable to a signal to allow the ripple-carry to use it
      r_ADD_TERM1 <= v_ADD_TERM1;
      r_ADD_TERM2 <= v_ADD_TERM2;
      wait for 60 ns;
      write(v_OLINE, w_SUM, right, c_WIDTH);
      writeline(file_RESULTS, v_OLINE);
    end loop;
    file_close(file_VECTORS);
    file_close(file_RESULTS);

    wait;
  end process;
end behave;
```

input_vectors.txt:

```
0000 0000
0000 0001
1000 1000
1111 1111
```

output_results.txt:

```
00000
00001
10000
11110
```