

5. File I/O in VHDL

In VHDL, we can perform file handling using the TextIO library. This feature is very useful for documenting the results of a program that has been created. To use the TextIO library, we need to add it at the beginning of our program as follows:

```
use std.textio.all;
use ieee.std_logic_textio.all; -- For std_logic types
```

5.1 Read File

When repeatedly simulating a design, changing the value of each input one by one can be time-consuming and inefficient. Therefore, we can use a feature from the TextIO library that can **read inputs from a file** to be used in the design simulation.

Here is how to read input from a file in VHDL. First, we can define the file to be read within a process statement and open it in `read_mode`:

```
process
  -- Define the file and its open mode
  file text_file : text open read_mode is "filename.txt";

  -- Variable to receive data from the file
  variable fileinp : integer;

  -- Line type variable to hold a row from the text file
  variable row : line;

  -- Variable to store file reading status
  variable ok : boolean
```

Then, we can read the file using the `readline` and `read` procedures from the TextIO library as follows:

```
begin
  while not endfile(text_file) loop --loop until the end of the text file
    readline(text_file, row); --reads the line from the file
```

```

-- Skip empty lines or comments that start with '#'
if row.all'length = 0 or row.all(1) = '#' then
    next;
end if;

-- Reads a variable from the line and puts it into fileinp
read(row, fileinp, ok);

assert ok
report "Read 'sel' failed for line: " & row.all
severity failure; --report if the file read fails

wait for delay; --delay for the read iteration
end loop;
end process;

```

5.2 Write File

Besides reading a set of inputs to be used in a testbench, we can also use the TextIO library to **save the results of our testbench to a file**. This allows us to analyze the results without having to use a simulator like ModelSim or Vivado.

Here are the steps required to write the results of a testbench to a file. First, we can define the file to be created and open it in `write_mode`:

```

process
    -- Open "filename.txt" in write mode
    file text_file      : text open write_mode is "filename.txt";

    -- Line type variable to build a row for the text file
    variable row        : line;

    -- Variable holding the data to be written to the file
    variable data_write : integer;

```

Then, we can use the `write` and `writeline` procedures from the TextIO library to write data to the file as follows:

```

begin
    -- writes data to the file, left-justified, 15 characters
    write(row, data_write, left, 15);

```

```
writeline(text_file, row); -- writes the line to the file  
end process;
```

Revision #3

Created 2025-09-15 16:24:25 UTC by RE

Updated 2025-09-16 09:57:15 UTC by RE