

## ModelSim® - VHDL

### 1. Zakres tematyczny ćwiczenia:

- budowa testbench'a VHDL z wykorzystaniem plików wejścia/wyjścia,
- modelowanie automatów skończonych w języku VHDL.

### 2. Wykorzystanie plików w środowisku testowym VHDL

Pakiety potrzebne przy wykorzystaniu wejścia/wyjścia plikowego w modelach VHDL to `textio` (biblioteka `std`) oraz `std_logic_textio` (biblioteka `ieee`). Pakiet `std_logic_textio` definiuje funkcje `read` i `write` przeciążone dla typów `std_logic`. Typy otwarcia plików (`file_open_kind`) i statusu plików (`file_open_status`) zdefiniowane zostały w pakiecie `standard` (biblioteka `std`).

Otwarcie pliku do odczytu:

```
-- 1076-1987
file <handle>: TEXT is in "<file>";
file wejście: TEXT is in „xyz.txt”;
-- 1076-1993
file <handle>: TEXT open <file_open_kind> is "<file>";
file wejście: TEXT open READ_MODE is „xyz.txt”;
```

Otwarcie pliku do zapisu:

```
-- 1076-1987
file <handle>: TEXT is out "<file>";
file wyjście: TEXT is out „xyz.txt”;
-- 1076-1993
file <handle>: TEXT open <file_open_kind> is "<file>";
file wyjście: TEXT open WRITE_MODE is „xyz.txt”;
```

### Listing 1. Model badany

```
entity bin2onehot is
  Port ( bin : in std_logic_vector(3 downto 0);
        onehot : out std_logic_vector(9 downto 0));
end bin2onehot;

architecture Behavioral of bin2onehot is

begin
  with bin select
  onehot<="0000000010" when "0001",--1
        "0000000100" when "0010", --2
        "0000001000" when "0011", --3
        "0000100000" when "0100", --4
        "0001000000" when "0101", --5
        "0001000000" when "0110", --6
        "0010000000" when "0111", --7
        "0100000000" when "1000", --8
        "1000000000" when "1001", --9
        "0000000001" when others; --0
end Behavioral;
```

## Listing 2. Testbench z wykorzystaniem textio

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE STD.TEXTIO.ALL;

ENTITY bin2onehot_tb IS
END bin2onehot_tb;

ARCHITECTURE testbench_arch OF bin2onehot_tb IS
FILE infile: TEXT is in "test.txt";      -- plik wymuszen
FILE outfile : TEXT is out "wyniki.txt"; -- plik wynikow

-- deklaracja komponentu -----
COMPONENT bin2onehot
  PORT (
    bin : in  std_logic_vector (3 DOWNTO 0);
    onehot : out std_logic_vector (9 DOWNTO 0)
  );
END COMPONENT;

SIGNAL bin_internal : std_logic_vector (3 DOWNTO 0);
SIGNAL onehot_internal : std_logic_vector (9 DOWNTO 0);

-- architektura test bench'a-----
BEGIN
  UUT : bin2onehot          -- unit under test
  PORT MAP (               -- mapowanie portow
    bin => bin_internal,
    onehot => onehot_internal);

  PROCESS                  -- process bez listy wrażliwości
    VARIABLE text_in, text_out : LINE;
    VARIABLE in_data : integer := 0;
  BEGIN
    write(text_out,string'("bin") & HT & string'("one_hot"));
    writeline(outfile,text_out);    -- nałówek pliku wynikowego
    czytaj:
    while not Endfile(infile) loop
      readline(infile,text_in); -- czytaj linie pliku wejściowego
      read(text_in, in_data);    -- pobierz zmienną in_data
      bin_internal <= conv_std_logic_vector(in_data,4);
      WAIT FOR 100 ns;
      write(text_out,bin_internal);
      write(text_out,HT);
      write(text_out,onehot_internal);
      writeline(outfile,text_out); -- zapis kolejnej linii pliku wynikowego
    end loop czytaj;
    WAIT;
  END PROCESS;
END testbench_arch;
```

## 3. Przygotowanie projektu

[!] **Polecenie:** utwórz nowy projekt o nazwie <nr\_indeksu\_4> w katalogu CADHDL na dysku wskazanym przez prowadzącego (jeśli katalog nie istnieje utwórz go); pozostaw domyślną nazwę dla biblioteki roboczej;

[!] **Polecenie:** pobierz z serwera kursu pliki 'bin2onehot.vhd' (model badany), 'bin2onehot\_tb.vhd' (testbench). Zapisz pliki w dowolnym katalogu tymczasowym; dodaj powyższe pliki do projektu z opcją 'Copy to project directory' i domyślnym typem źródła;

#### 4. Symulacja z wykorzystaniem textio

**[!] Polecenie:** napisz makro .do kompilacji i symulacji; uruchom symulację modelu korzystając z makra .do (testbench VHDL);

**[!Z] Polecenie:** dokonaj modyfikacji testbench'a tak, aby można było wskazywać tekstowe pliki wejściowe i wyjściowe z linii poleceń ModelSim'a; przy definicjach plików używaj składni vhd'93;

**[!Z] Polecenie:** Stwórz testbench VHDL dla pamięci statycznej ram20d [dostępna na serwerze kursu]. Wymagania dla środowiska symulacji:

- generator adresu i sygnałów sterujących jako oddzielne procesy zawarte w testbench'u;
- symulacja zapisu całej pamięci zawartością pliku tekstowego;
- symulacja odczytu zawartości całej pamięci do pliku tekstowego;

**wyniki symulacji przedstaw prowadzącemu!**

ram20d to model pamięci jednoportowej z zapisem synchronicznym i odczytem asynchronicznym. Czas propagacji ustalany jest przez wartość generyczną  $T_{pd}$ . Pamięć posiada dwa wejścia sterujące: we – write enable (sterowanie zapisem) aktywne stanem wysokim oraz oe – output enable (sterowanie szyną danych) aktywne stanem wysokim.

**[!Z] Zadanie:**

Umieść w stworzonym powyżej środowisku testowym dla pamięci ram20d proces automatycznej weryfikacji poprawności działania urządzenia.

Napisz makro kompilacji/symulacji.

[1] ModelSim®SE Reference Manual, Software Version 6.4a, Mentor Graphics 2008.