Entrées et sorties par fichiers en VHDL

Entrées et sorties par fichiers en VHDL
Sujets de ce thème

- Banc d’essai et entrées et sorties par fichiers en VHDL
- Exemples de code

Banc d'essai et entrées et sorties par fichiers en VHDL

- On s'est concentrés à date sur la génération algorithmique de vecteurs de test à l'intérieur d'un banc d'essai codé en VHDL.
- Dans le processus de conception d'un système numérique, on passe souvent par une modélisation de haut niveau, par exemple avec Matlab.
- Lors de cette modélisation, on génère souvent une grande quantité de cas de test et de réponses attendues qui sont entreposés dans un fichier.
- Le banc d'essai peut lire ces cas de test et les réponses associées.
- Le banc d'essai peut aussi écrire ses résultats dans un fichier. Ceci est utile si:
  - la simulation dure plusieurs heures;
  - on désire obtenir des résultats progressifs;
  - on doit effectuer un traitement plus complexe des résultats dans un autre environnement que le simulateur VHDL. Par exemple, on pourrait vouloir afficher une image générée sous la forme d'un flux de pixels par un module.
- Les entrées et sorties de fichier en VHDL se font à l'aide d'objets de la catégorie `file`.
- Comme on traite du texte lors de ces opérations, on utilise aussi les types et les fonctions définis dans le package `textio` qui fait partie du langage.

Exemple
Module à vérifier et fichier de stimuli et réponses

```
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.all;
entity detecteurPremier is
   port (
      I : in unsigned(5 downto 0);
      F : out std_logic
   );
end detecteurPremier;
architecture flotdonnees of detecteurPremier is
begin
   with to_integer(I) select
      F <=
      '1' when 2 | 3 | 5 | 7 | 11 | 13 | 17 |
           19 | 23 | 29 | 31 | 37 | 41 | 43 |
           47 | 53 | 59 | 61 | 63, -- erreur!
        '0' when others;
end flotdonnees;
```

```
-- colonne1: entiers de 0 à 63
-- colonne2: P pour premier, N pour pas premier
0 N
1 N
2 P
3 P
4 N
5 P
...
```

Exemple:
banc d'essai avec lecture de stimuli et de réponses

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use std.textio.all;

entity detecteurPremierTB is
end detecteurPremierTB;

architecture arch2 of detecteurPremierTB is

signal I : unsigned(5 downto 0); -- signal pour les vecteurs de tests
signal F : std_logic; -- signal pour les réponses

constant filename : string := "premiers.txt";

file vecteurs : text open read_mode is filename;

begin

   UUT : entity detecteurPremier(flotdonnees) port map (I, F);
```

```
process

variable tampon : line;     -- pointeur vers un objet de type string
variable n : integer;
variable c : character;

begin
   while not endfile(vecteurs) loop
      readline(vecteurs, tampon);
      if tampon(1 to 2) /= "--" then -- passer les lignes de commentaires
         read(tampon, n); -- lecture de l'entier
         read(tampon, c); -- lecture du séparateur
         read(tampon, c); -- lecture de l'indication: premier ('P') ou non ('N')
         I <= to_unsigned(n, 6);
         wait for 10 ns;
         assert ((c = 'P') = (F = '1') and (c = 'N') = (F = '0'))
            report "erreur pour l'entrée " & integer'image(n) severity error;
      end if;
   end loop;
   deallocate(tampon); -- relâcher la mémoire du tampon
   report "simulation terminée" severity failure;

end process;

end arch2;
```

```
-- Fichier premier.txt
-- colonne1: entiers de 0 à 63
-- colonne2: P pour premier, N pour pas premier
0 N
1 N
2 P
3 P
4 N
5 P
...
```

Les entrées et sorties de fichier en VHDL se font à l'aide d'objets de la catégorie `file`. Comme on traite du texte lors de ces opérations, on utilise aussi les types et les fonctions définis dans le package `textio` qui fait partie du langage.

Exemple:
banc d'essai avec stimuli et écriture des réponses dans un fichier

```
process

variable tampon : line; -- pointeur vers objet de type string
variable tampon2 : line;

begin

-- La procédure writeline libère le pointeur quand elle a fini,
-- donc il faut construire une copie de l'objet si on veut l'afficher 2 fois.
-- À partir d'un pointeur, on va chercher le contenu avec '.all'.

write(tampon, string'(" ** sortie de simulation, detecteurPremierTB.vhd ** "));
write(tampon2, tampon.all); -- copier la chaîne de caractères
writeline(resultats, tampon); -- écriture dans le fichier
writeline(output, tampon2); -- écriture à la console

for k in 0 to 63 loop  -- application exhaustive des vecteurs de test
   I <= to_unsigned(k, 6);
   wait for 10 ns;
   write(tampon, string'("temps: ")); write(tampon, now, unit => ns);
   write(tampon, string'(", entier: ") & integer'image(k));
   write(tampon, string'(", sortie: ") & std_logic'image(F));
   write(tampon2, tampon.all);  -- copie la chaîne de caractères
   writeline(resultats, tampon); -- écriture dans le fichier
   writeline(output, tampon2); -- écriture à la console
end loop;

report "simulation terminée" severity failure;

end process;
```

Vous devriez maintenant être capable de …

- Utiliser les fonctions de lecture et d'écriture de fichiers en VHDL. (B3)

Code	Niveau (http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom)
B1	Connaissance – mémoriser de l'information.
B2	Compréhension – interpréter l'information.
B3	Application – confronter les connaissances à des cas pratiques simples.
B4	Analyse – décomposer un problème, cas pratiques plus complexes.
B5	Synthèse – expression personnelle, cas pratiques plus complexes.