Binome : Mathieu ESCOUTELOUP – Bastien MAHTAL

CSI 2A : Projet Informatique

Analyse syntaxique :

Fichier vide :

Vide.vhd	OK

Déclaration bibliothèque :

biblio.vhd	OK

Déclaration entité :

entite_seule.vhd	OK
entite_noarchi.vhd	OK
archivide.vhd	OK

Déclaration de signaux :

signaux.vhd	NOK
assign.vhd	NOK
typage.vhd	NOK

Syntaxe des process :

proc_vide.vhd	NOK
proc.vhd	NOK
case.vhd	NOK

Instanciation :

instances.vhd	NOK

Fichier avec des erreurs : erreur.vhd

Vous commenterez/décommenterez les erreurs au fur et à mesure de vos tests

Déclaration bibliothèque :

Ligne 2	erreur : manque y	NOK
Ligne 3	erreur : manque « . »	NOK
Ligne 4	erreur : manque espace	NOK
Ligne 5	espace ne pose pas de pb	OK
Ligne 6	manque « ; »	NOK
Ligne 7		OK

Déclaration entité :

Ligne 9	erreur : manque « is »	NOK
Ligne 10	erreur : manque « (»	NOK
Ligne 11	erreur : « bt » type invalide	NOK
Ligne 12	erreur : « , »	NOK
Ligne 13	erreur « : »	NOK
Ligne 14		OK
Ligne 15	erreur « ni »	NOK
Ligne 16	erreur manque « downto »	NOK
Ligne 17	erreur : manque «) »	NOK
Ligne 18	erreur « signaux » invalide	NOK

Déclaration architecture

Ligne 20	warning : étiquette	NOK
Ligne 23	erreur : « sigl »	NOK
Ligne 24		OK
Ligne 27	erreur manque « begin »	NOK
Ligne 30	erreur affectation	NOK
Ligne 33		OK
Ligne 39	erreur : « structe »	NOK

Ligne 17-18 : erreur « end » ne peut pas suivre le « ; » car il manque une «) » à la ligne d'avant
Ligne 18 : Etiquette « signaux » non valide, on veut « assign »

Ligne 3 : manque « . » entre « STD » et « textio »
Ligne 4 : manque espace entre « Library » et « ieee »
Ligne 9-10 : erreur « port» ne peut pas suivre le l'étiquette car il manque le mot clef « is » à la ligne d'avant
Ligne 13 : il y a « : » alors qu'on devrait avoir une autre étiquette de port
Ligne 15 : « ni » doit être remplacé par la direction « in »
Ligne 23 : lexeme « sigl » invalide, doit être « signal »
Ligne 30 : L'opérateur « >= » n'existe pas donc l'assignation n'est pas valide
Ligne 39 : L'étiquette « structe » n'est pas valide, on attend « struct »
Ligne 20 : L'étiquette « toto » ne correspond à aucune entité de ce fichier = erreur potentielle

Synthèse :

Erreurs de typage:

Base_bit.vhd	OK/NOK

Synthèse :

Base_bit.vhd	OK/NOK
Case.vhd	OK/NOK
Variables.vhd	OK/NOK
Process_clk.vhd	OK/NOK
Process_clk2.vhd	OK/NOK
Process_clk3.vhd	OK/NOK
Process_clk4.vhd	OK/NOK
Latch.vhd	OK/NOK
Sensibilite.vhd	OK/NOK

SYNTHESE NON TRAITEE

Bilan des tests :

Quels sont les points forts et faibles du code ?

POINTS FORTS :
- La vérification syntaxique détecte la plupart des erreurs (du moins dans les cas traités)
- Tous les blocs du VHDL sont bien séparés, ce qui rend le code plus « clair » :
    - -Vérification syntaxique plus simple à réaliser
    - -Cela rend plus simple l'ajout de nouvelles fonctionnalités
- La gestion d'erreur est très pratique

POINTS FAIBLES :
- Pas de synthèse
- La manière d'implémenter l'arbre n'est peut-être pas la plus optimale
- La vérification syntaxique aurait pu être codée en utilisant le principe exact d'une FSM en C (avec une structure qui contiendrait toutes les étapes)

Binome : Mathieu ESCOUTELOUP – Bastien MAHTAL

Que souhaitez vous corriger dans la matinée ?

- Faire une vérification syntaxique rapide pour une affectation (sans prendre tous les cas en compte)

- Pas d'étiquette à la fin d'un process n'est pas une erreur en soi, on va donc enlever cette erreur.

A la fin de la matinée, quels sont les tests qui passent ?

- Le fichier case.vhd n'est pas réellement traité car lorsque l'on croise un case, on crée un FLAG_CASE mais ensuite la vérification syntaxique du case n'est pas effectuée.

- Autrement tous les tests syntaxiques passent, les erreurs rencontrées sont toutes indiquées et peuvent donc être débuggées.
  - Le format du process avec étiquette en déclaration, par exemple « un : process(start) », mais sans étiquette en fin de process est maintenant accepté.
  - On gère l'affectation d'un signal dans un cas simple « done <= last_round_signal; » et si on a un affectation du type : « resultat <= test1 AND test2; » avec n'importe quelle porte (AND, OR, XOR, NAND, AND)

- Les tests de synthèse ne sont toujours pas traités.