Objets des catégories `signal` et `variable` en VHDL

Utilisation d'objets des catégories `signal` et `variable` en VHDL
Sujets de ce thème

- Cette présentation fait suite à la présentation sur la simulation de modèles VHDL

- Assignation de valeurs à des `variables` et des `signals`
- Exemples
- Formes d'un processus

Assignations à des objets des catégories `signal` et `variable` dans un processus

- À l'intérieur d'un processus, les énoncés sont exécutés de façon séquentielle.
- Les assignations à des objets des catégories `variable` et `signal` sont traitées différemment.
- Quand la cible est une `variable`, celle-ci prend immédiatement la valeur qui lui est assignée.
- Pour un `signal`, l'assignation est placée sur la liste des événements:
  - au temps courant + Δ (clause implicite `after 0 ns`); ou,
  - au moment spécifié par une clause `after` explicite.
- L'évaluation des énoncés du processus se poursuit ensuite.
- Les assignations de valeurs à des `signals` ne sont effectuées que lorsqu'elles sont tirées de la liste des événements:
  - après la fin de l'exécution du processus; ou bien,
  - pendant la suspension de l'exécution du processus par un énoncé `wait`.

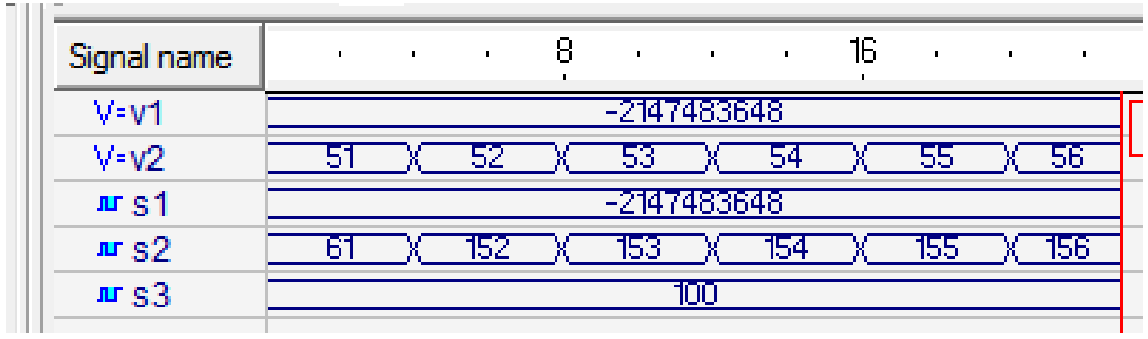
Expressions contenant des objets des catégories `signal` et `variable` dans un processus

Quand	Variable	Signal
Au début de la première exécution du processus	Valeur d'initialisation dans la déclaration, ou valeur par défaut correspondant au type.	
Au début des autres exécutions du processus	Valeur à la fin de la dernière exécution	Valeur assignée après la fin de la dernière exécution
Pendant l'exécution d'un processus qui ne contient pas d'énoncé `wait`	Dernière valeur assignée	Valeur au lancement du processus
Pendant l'exécution d'un processus qui contient un ou des énoncés `wait`	Dernière valeur assignée	Valeur au lancement du processus ou bien dernière valeur assignée avant l'énoncé `wait` qui précède l'expression

Exemple #1

```
entity demoprocessus1 is
end demoprocessus1;

architecture arch of demoprocessus1 is
signal s1 : integer;
signal s2 : integer := 5;
signal s3 : integer := 10;
begin
   process
   variable v1 : integer;
   variable v2 : integer := 50;
   begin
      v2 := v2 + 1;
      s3 <= 100;
      s2 <= s3 + v2;
      wait for 4 ns;
   end process;
end arch;
```

Quand	Variable	Signal
Au début de la première exécution du processus	Valeur d'initialisation dans la déclaration ou valeur par défaut correspondant au type.	
Au début des autres exécutions du processus	Valeur à la fin de la dernière exécution	Valeur assignée après la fin de la dernière exécution
Pendant l'exécution d'un processus qui ne contient pas d'énoncé wait	Dernière valeur assignée	Valeur au lancement du processus
Pendant l'exécution d'un processus qui contient un ou des énoncés wait	Dernière valeur assignée	Valeur au lancement du processus ou bien dernière valeur assignée avant l'énoncé wait qui précède l'expression



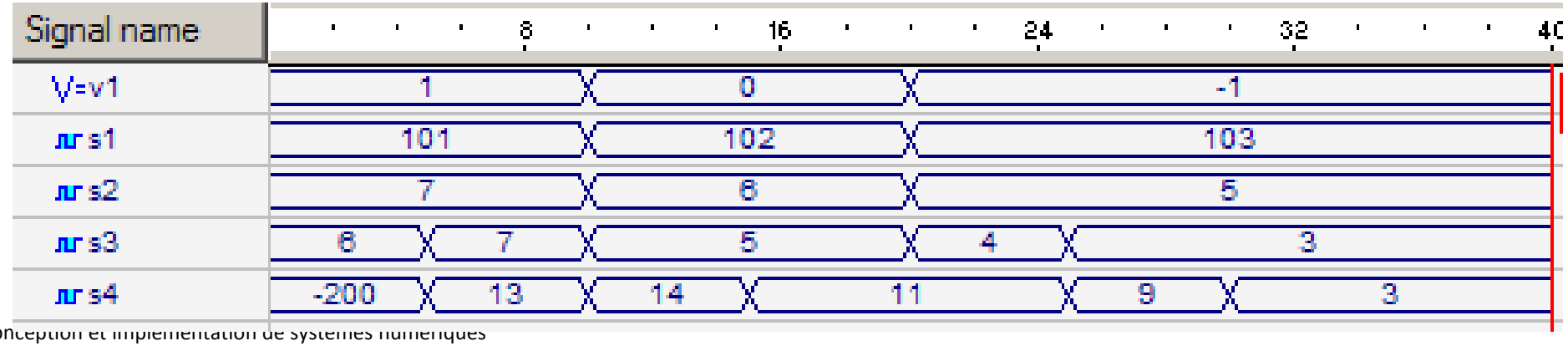
Time	Delta	V= line__12/v1	V= line__12/v2	s1	s2	s3
0 ps	0	-2147483648	51	-2147483648	5	10
0 ps	1	-2147483648	51	-2147483648	61	100
4000 ps	0	-2147483648	52	-2147483648	61	100
4000 ps	1	-2147483648	52	-2147483648	152	100
8000 ps	0	-2147483648	53	-2147483648	152	100
8000 ps	1	-2147483648	53	-2147483648	153	100
12000 ps	0	-2147483648	54	-2147483648	153	100
12000 ps	1	-2147483648	54	-2147483648	154	100
16000 ps	0	-2147483648	55	-2147483648	154	100
16000 ps	1	-2147483648	55	-2147483648	155	100
20000 ps	0	-2147483648	56	-2147483648	155	100
20000 ps	1	-2147483648	56	-2147483648	156	100

Exemple #2

```
entity demoprocessus is
end demoprocessus;

architecture arch of demoprocessus is

signal s1 : integer := 100;
signal s2, s3, s4 : integer := -100;

begin
```

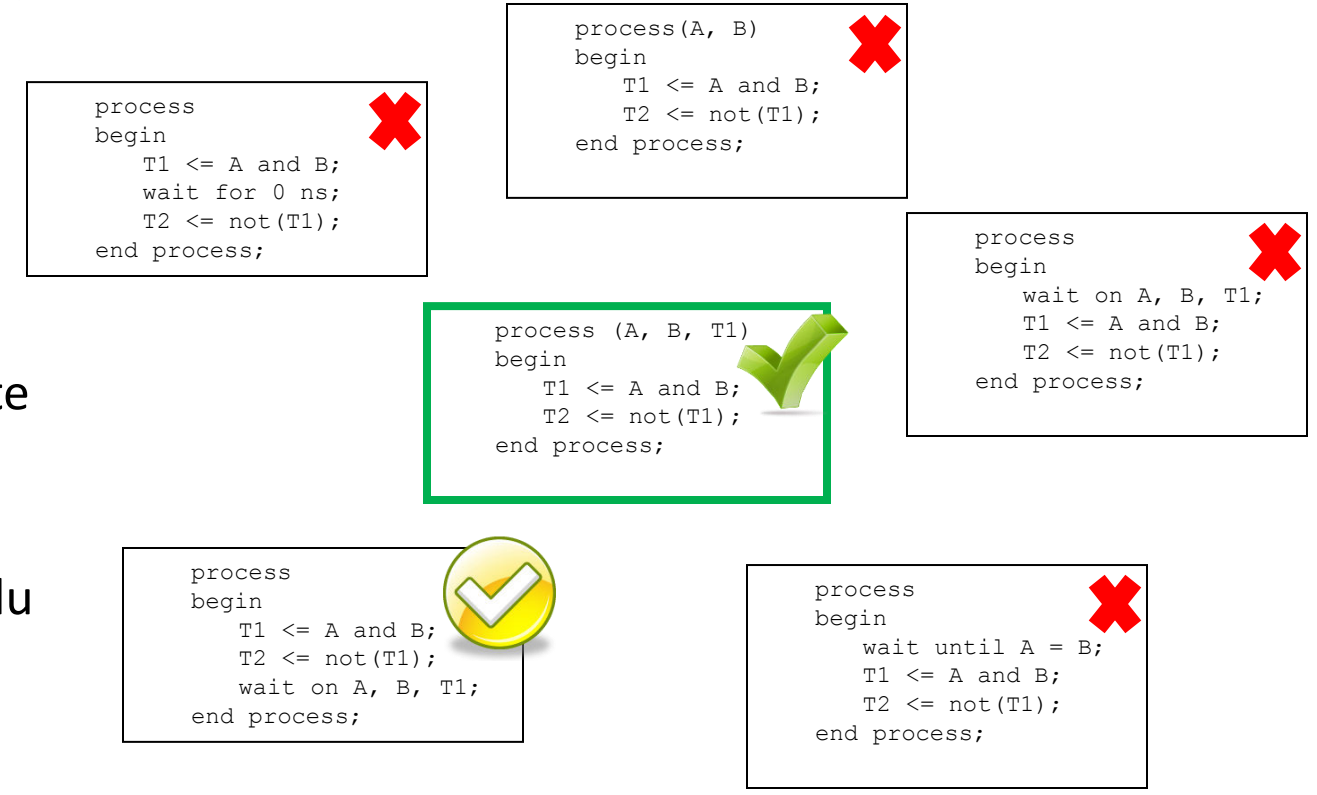
```
   process
   variable v1 : integer := 2;
   begin
      while (v1 >= 0) loop
            s2 <= v1 + 5;
            v1 := v1 - 1;
            s3 <= v1 + 5;
            s1 <= s1 + 1;
            s4 <= s3 + s2;
            wait for 5 ns;
            s3 <= s3 + v1;
            s4 <= s3 + s2;
            wait for 5 ns;
            s4 <= s3;
      end loop;
      wait for 100 ns;
   end process;
end arch;
```

Quand	Variable	Signal
Au début de la première exécution du processus	Valeur d'initialisation dans la déclaration ou valeur par défaut correspondant au type.	
Au début des autres exécutions du processus	Valeur à la fin de la dernière exécution	Valeur assignée après la fin de la dernière exécution
Pendant l'exécution d'un processus qui ne contient pas d'énoncé wait	Dernière valeur assignée	Valeur au lancement du processus
Pendant l'exécution d'un processus qui contient un ou des énoncés wait	Dernière valeur assignée	Valeur au lancement du processus ou bien dernière valeur assignée avant l'énoncé wait qui précède l'expression



Quel format de processus choisir?

- Il y a plusieurs façons de décrire un comportement donné en VHDL.
- On peut remplacer la liste de sensitivité d'un processus par un énoncé `wait` (avec trois variations: `wait for`, `wait until`, `wait on`).
- On peut avoir une liste de sensitivité non complète (qui n'inclut pas tous les signaux qui font partie d'expressions dans le processus).
- Ces approches ne sont pas recommandées pour du code qui doit être synthétisé.



Vous devriez maintenant être capable de …

- Interpréter le modèle de simulation pour expliquer le traitement différent des objets des catégories signal et variable dans un modèle. (B2, B3)
- Étant donné un modèle, donner la valeur des objets dans le temps. (B2, 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.