

Contrôle de connaissances SE207 “SystemC”

24 juin 2016

Instructions

Ce contrôle de connaissances est strictement individuel. Vous devez modifier ce fichier pour y inclure vos réponses. Seules les parties entre — sont à modifier.

Questions

Question 1

En SystemC, pour représenter une donnée signée sur 32bits vous pouvez utiliser l'un des types suivants :

- `int32_t` (unsigned int),
- `sc_int<32>`
- `sc_bv<32>`
- `sc_lv<32>`

Quels critères vous feraient choisir un type plutôt qu'un autre ?

Ceci est un exemple de réponse. Merci d'effacer ce paragraphe (mais de laisser les groupes de trois tirets et les lignes vides avant et après eux) lorsque vous y écrirez la vôtre.

Question 2

Pourquoi ne peut-on pas connecter *directement* la sortie d'une module (par exemple `sc_out<bool> o`) à l'entrée d'un autre module (par exemple `sc_in<bool> i`) ?

Question 3

- Pourquoi doit-on avoir une boucle infinie dans un `SC_THREAD` ?
- Que se passe-t-il s'il n'y a pas cette boucle ?

Question 4

Nous voulons modéliser un bloc matériel synchrone (à une horloge `clk`) dans lequel une étape de traitement doit attendre le passage à true d'un signal de validation `start`. Pour ce fait, on utilise un `SC_THREAD` sensible au seul front montant de l'horloge (`sensitive « clk.pos() ;`).

Nous proposons les deux implémentations suivantes pour cette attente (le reste du code n'est pas montré) :

Version 1

```
...  
// attente du passage à 1 de start  
wait(start.posedge_event());  
// passage à l'étape suivante  
...
```

Version 2

```
...  
// attente du passage à 1 de start  
while(!start)  
    wait();  
// passage à l'étape suivante  
...
```

- Expliquez brièvement la différence de comportement entre ces deux versions.
- Pourquoi choisiriez-vous l'une par rapport à l'autre ?

“

Question 5

- Pourquoi devons-nous prendre des précautions quand nous utilisons une `sc_fifo` dans une `SC_METHOD` ?
