

Contrôle de connaissances SE207 “SystemC”

27 juin 2018

Instructions

Ce contrôle de connaissances est strictement individuel. Vous devez modifier ce fichier pour y inclure vos réponses puis l'ajouter à votre dépôt dans un dossier CC à la racine de ce dernier.

Seules les parties entre --- sont à modifier. Laissez une ligne vide avant et après chaque ---.

Questions

Question 1

Dans quels cas est-il obligatoire d'utiliser des signaux (`sc_signal`) à l'intérieur d'un module SystemC (`SC_MODULE`) ?

*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.*

- X ceci est X
- Y ceci est Y

```
// ceci est un exemple de code
int main() {
    return 0;
}
```

Question 2

Pourquoi peut-on connecter *directement* la sortie (`sc_out<>`) d'un module à la sortie d'un autre module mais pas à une entrée (`sc_in<>`) ?

Question 3

Écrire le modèle **RTL** d'un compteur module 33 avec remise à zéro asynchrone en utilisant une `SC_METHOD` puis un `SC_THREAD` (seuls les codes des processus sont demandés).

```
// Code du compteur
```

```
...
```

Question 4

- Que se passe-t-il s'il existe une boucle infinie dans une SC_METHOD ?
- Que se passe-t-il si la fonction wait() est appelée dans cette boucle infinie ?

Question 5

Pour modéliser deux calculs effectués par des blocs matériels synchrones nous utilisons deux SC_THREAD sensibles au front d'un signal d'horloge clk. Comme les deux calculs sont interdépendants, nous utilisons un sc_mutex pour les synchroniser.

- Donnez un exemple de code montrant, dans les grandes lignes, ce fonctionnement.
- Donnez une implémentation RTL modélisant la même fonctionnalité en utilisant des SC_METHOD.
- Pour ces deux versions :
 - Quelles sont les différences éventuelles du point de vue des ressources nécessaires à la simulation ?
 - Quelles sont les différences éventuelles du point de vue de la précision temporelle de la simulation.

```
// premier code
```

```
...
```

```
// second code
```

```
...
```
