

Contrôle de connaissances SE207 “SystemC”

25 juin 2019

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

En SystemC, pour représenter une donnée sur 13bits vous pouvez utiliser l'un des types suivants :

- `sc_bv<13>`
- `sc_lv<13>`
- `sc_uint<13>`
- `sc_int<13>`

1. Expliquez brièvement la différence entre ces types.
2. 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.*

- X ceci est X
- Y ceci est Y

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

Question 2

Les notions de signal et d'affectation différée sont des notions importantes pour un simulateur évènementiel.

1. Expliquez brièvement à quoi correspondent ces notions et dans quels cas il est indispensable de les utiliser.
2. Expliquez comment elles sont mises en œuvre en SystemC.

...

Question 3

Nous voulons modéliser un système synchrone dont la sortie **Q** produit cycliquement la séquence suivante :

0, 1, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7

Composée des phases suivants :

1. incrémente de 0 à 4,
2. décrémente de 4 à 0,
3. incrémente de 0 à 7...

Avec les caractéristiques suivantes :

- la sortie **Q** est sur 3 bits,
- elle évolue à chaque front montant d'une horloge **clk**,
- un signal de remise à zéro **nrst** actif sur niveau bas permet de réinitialiser la séquence,

1. Écrire le code de deux module SystemC reproduisant ce comportement :
 1. le premier contiendra un **SC_THREAD**,
 2. le second contiendra une ou plusieurs **SC_METHOD**.

```
// En utilisant un SC_THREAD
```

```
SC_MODULE(SeqThread) {  
    sc_in<bool> clk;  
    ...  
}
```

```
// En utilisant une ou plusieurs SC_METHOD
```

```
SC_MODULE(SeqMethod) {  
    sc_in<bool> clk;  
    ...  
}
```

Question 4

1. Expliquez pourquoi les **SC_THREAD** contiennent souvent une boucle infinie ?
— Que se passe-t-il si cette boucle n'existe pas ?
2. Que se passe-t-il par contre si une boucle infinie existe 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`. Pour les synchroniser nous utilisons un `sc_mutex`.

1. Donnez, **dans les grandes lignes**, un exemple de code illustrant ce fonctionnement.
2. Ce modèle est-il synchrone avec l'horloge `clk` ?
3. Donnez, **toujours dans les grandes lignes**, un exemple de code pour modéliser la même fonctionnalité au niveau RTL ?
4. Comparez ces deux modèles du point de vue des ressources nécessaires à la simulation.

```
// premier code (thred/mutex)
```

```
...
```

```
// second code (rtl)
```

```
...
```
