Contrôle de connaissances SE207 “SystemC”

17 juin 2020

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

1. Quels types pouvez-vous utiliser en SystemC pour représenter des entiers ?
2. Quels critères vous feraient choisir un type plutôt qu'un autre ?
3. Quelles implications sur les performances de la simulation ?

---

*Ceci est un exemple de réponse. **Merci d'effacer ce paragraphe**

LAISSEZ les groupes de trois tirets précédés et suivis d'une ligne vide

- X ceci est X
- Y ceci est Y

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

Question 2

1. Pourquoi, contrairement aux autres langages HDL, SystemC vous impose l'utilisation d'un type particulier pour pouvoir faire des affectations différées ?
2. Quel est ce type ? Et expliquez brièvement comment ces affectations différées sont mises en œuvre dans la bibliothèque.
3. Dans quels cas l'utilisation d'affectations différées est-elle nécessaire ?

…

Question 3

Nous voulons modéliser un système de temporisation synchrone ayant les caractéristiques suivantes :

- une horloge `clk` et un signal de remise à zéro asynchrone `nrst` actif sur niveau bas,
- une entrée sur 1 bit go,
- une sortie sur 4 bits initialement à 0.

À chaque fois que go passe à 1 la sortie effectue la séquence suivante :

- incrémente de 0 à 11,
- patiente 7 cycles,
- décrémente de 11 à 0 puis s'arrête.

1. Écrire le code de deux module SystemC reproduisant ce comportement :
    - le premier contiendra un **`SC_THREAD`** pour une description "cycle accurate",
    - le second contiendra une ou plusieurs **`SC_METHOD`** pour une description RTL.

```
// En utilisant un SC_THREAD
SC_MODULE(SeqThread) {
  sc_in<bool> clk;
  sc_in<bool> nrst;
  sc_in<bool> go;
...
}

// En utilisant une ou plusieurs SC_METHOD
SC_MODULE(SeqMethod) {
  sc_in<bool> clk;
  sc_in<bool> nrst;
  sc_in<bool> go;
...
}
```

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. Donnez un exemple d'utilisation pratique où cette boucle n'est pas nécessaire.

...

Question 5

Deux processus servent à modéliser deux composants matériels qui fonctionnent à tour de rôle. Pour modéliser ce comportement, nous pouvons utiliser des `sc_fifo`.

1. Donnez, **dans les grandes lignes**, un exemple de code illustrant le fonctionnement **cyclique** suivant :
    - le premier module effectue une action,
    - il passe la main au second et attend,

- le second module prend la main et effectue une action,
- il repasse la main au premier et attend,

Précisez le type des processus utilisés

```
// premier processus
void P1() {
}

// second processus
void P2() {
}
```

Nous voulons un modèle plus précis temporellement et nous ajoutons un signal d'horloge `clk` dont le front montant est ajouté à la liste de sensibilité des deux processus.

1. Pensez-vous que la synchronisation des deux processus garantisse que le modèle reste synchrone (*justifiez votre réponse*) ?
2. Proposez une autre méthode de synchronisation qui garantisse le synchronisme.
   - Quelles implications sur la vitesse de la simulation par rapport à l'utilisation de la `sc_fifo` ?

…
