Contrôle de connaissances SE207 “SystemC”

23 juin 2021

Instructions

Ce contrôle de connaissances est strictement individuel. Vous devez modifier la fiche de réponse 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

Nous souhaitons modéliser en SystemC un opérateur numérique calculant la moyenne arithmétique de 8 nombres codés sur 8 bits.

1. Quels types pouvez-vous utiliser en SystemC pour représenter ces entiers ?
2. Quels critères vous feraient choisir un type plutôt qu’un autre et quelles implications sur les performances et la précision 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

Les notions d'affectations différées et de signal 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 générant des requêtes utilisant un rendez-vous (handshake) avec l'interface suivante :

— une horloge `clk` et un signal de remise à zéro asynchrone `nrst` actif sur niveau bas,
— une sortie sur 1 bit `valid` indiquant le début d'une requête (initialement à l'état bas),
— une entrée sur 1 bit `ready` indiquant que la requête a été acceptée,
— une sortie `Q` sur 4 bits indiquant le nombre de requêtes effectuées jusqu'ici (initialement à `0`).

Une requête est complétée si `valid` *et* `ready` *sont tous deux à l'état haut au front d'horloge*.

Après l'initialisation, nous voulons le comportement suivant :

— on génère 13 requêtes consécutives,
— on patiente 7 cycles sans requêtes et sans modifier `Q`,
— on refait passer le compteur `Q` à `0` et on recommence.

1. Complétez le code des deux module SystemC suivants :
   — 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_out<bool> valid;
  sc_in<bool>  ready;
...
  SC_CTOR(SeqThread){
    SC_THREAD(run);
  }
...
}
```

```
// En utilisant une ou plusieurs SC_METHOD
SC_MODULE(SeqMethod) {
  sc_in<bool>  clk;
  sc_in<bool>  nrst;
  sc_out<bool> valid;
  sc_in<bool>  ready;
...
  SC_CTOR(SeqThread){
    SC_METHOD(fsm);
  }
...
}
```

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 ce passe-t-il si une boucle infinie est présente dans une **SC_METHOD** ?
   — Si ceci pose problème, est-il possible de le détecter simplement à la compilation ?

...

Question 5

Soit les deux modules suivants servant à modéliser deux composants matériels synchrone (à une horloge `clk`) qui s'échangent des données (des entiers de 32 bits).

Pour modéliser le canal de communication, nous voulons utiliser une `sc_fifo`.

```
// La fonction produisant les valeurs à transmettre
// est définie ailleurs
int prod();

SC_MODULE(PROD){
  sc_in<bool> clk;

  void P1() {
    for(;;){
       out.write(prod());
       wait();
    }
  }
  SC_CTOR(PROD){
```

```
  }
};

// La fonction utilisant les données reçues
// est définie ailleurs
void use(int);

SC_MODULE(CONSO){
  sc_in<bool> clk;

  void P2() {
    for(;;){
       use(in.read());
       wait();
    }
  }
  SC_CTOR(CONSO){
  }
};
```

---

1. Quel type de processus SystemC devons-nous utiliser ici ?
2. Complétez le code en :
   — ajoutant les déclarations des ports manquants,
   — la déclaration des processus.
3. En conservant la `sc_fifo` et le type de processus, pouvons-nous garantir que le modèle se comporte correctement s'il est simulé avec d'autres modèles synchrones ?
   — Si ce n'est pas le cas, proposez une modification minimale permettant de le garantir.
4. Nous voulons une un modèle plus proche d'un modèle **RTL** pour le module **PROD**. Pour cela, on vous propose d'utiliser une **SC_METHOD**.
   — Quelles précautions faut-il prendre en utilisant la `sc_fifo` (justifiez votre réponse) ?
   — Expliquez brièvement ce qui change du point de vue du comportement et de la performance de la simulation par rapport à la version initiale.

...
