



**Ingénieur 3ème année GE option: SMART**

**Mastères 2 : Signaux, Systèmes et Données (SSD)**

**Information System techniques (IST)**

**MANIP N°1 SOC : Conception Haut niveau**

**Enseignante : Dr. Faten BEN ABDALLAH**

**Objectif :**

La manipulation a pour objectifs de former l'étudiant à la gestion de SystemC, et entre autres la manipulation des systèmes combinatoires décrits au niveau cycle-prés (Cycle-Accurate level) en langage SystemC. L'évaluation se fait par simulation assistée par le logiciel GTKwave.

Tout au long de la manip, nous adoptons la version SystemC-2.3.3 standardisée sous la norme IEEE 1666 – 2005.

**1. Installation de SystemC :**

*Sous windows (visual studio) :*

Décompresser le fichier systemc-2.3.3.zip à la racine de C:\

0. Lancer Visual Studio

1. Installation des bibliothèques SystemC :

Menu : Fichier -> ouvrir -> Projet/Solution et aller chercher

C:\systemc2.3.3\msvc10\SystemC.sln

Compiler la bibliothèque SystemC

2. Créer un nouveau projet vide de type Win32 Console Application (à nommer HelloWorld)

3. Menu : **Projet -> Propriétés -> Propriétés de configuration**

3.1. Choisir **Répertoires VC++**

-> Cliquer sur **Répertoires Include** (Include Directories) et aller chercher le répertoire "C:\ systemc2.3.3\src"

-> Cliquer sur **Répertoires de bibliothèques** (Library Directories) et aller chercher le répertoire "C:\systemc2.3.3\msvc10\SystemC\debug"



### 3.2. Choisir C/C++ -> Ligne de commande

Sous « options supplémentaires» écrire `/vmg` utilisée pour déclarer un pointeur vers un membre d'une classe avant de définir la classe. Ce besoin peut survenir si vous définissez des membres dans deux classes différentes qui se réfèrent mutuellement. Pour les classes qui réfèrent mutuellement, une classe doit être référencée avant d'être définie.

### 3.3. Choisir Editeurs de lien -> entrée

Ajouter `systemc.lib` dans « Dépendances supplémentaires »

4. Cliquer sur Appliquer et ok

## 2. Projet 0 : Hello world

1-Créer un fichier d'entête `HelloWorld.h` avec les lignes du code suivantes :

```
#include <systemc.h>
SC_MODULE(Hello_world) {
    SC_CTOR(Hello_world) {}
    void say_hello() {
        cout << "hello world \n";
    }
};
```

2-Créer un fichier source `main.cpp` avec les lignes du code suivantes :

```
#include <systemc.h>
#include "HelloWorld.h"

int sc_main (int argc, char* argv[])
{
    HelloWorld hello("Hello");
    hello.say_hello();
    return(0);
}
```

## 3. Projet 1 : Bascule D

1. Donner le code systemC décrivant et simulant le fonctionnement d'une bascule D.

Rappelons que pour simuler un concept en SystemC dans un système synchrone on dispose généralement de 3 principaux modules (Dont le principe est expliqué par la figure 1) :

Un module dédié pour envoyer les requêtes de la liste de sensibilité d'entrée du concept (dans la figure nommé "stimulus generator"), l'autre est réservé pour visualiser les résultats de la simulation (dans la figure nommé "Response Monitor"). Le troisième contient le module conçu et qui se place entre les deux autres modules (dans la figure nommé "algo").

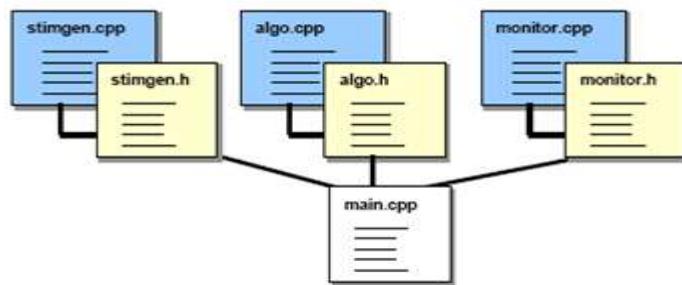
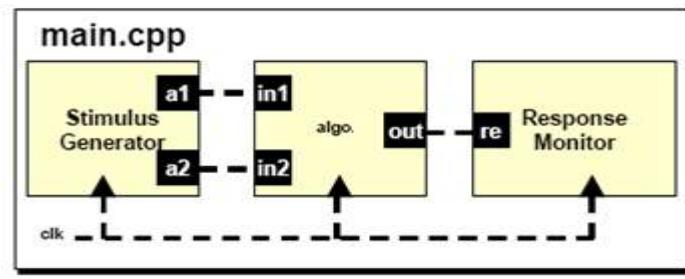


Figure 1 Concept systemC pour un système synchrone

2. Compiler l'exemple et donner le résultat d'exécution de l'application.
3. Générer un fichier fichier de trace ".vcd". Afficher le fichier à l'aide de gtkwave.