EE382V-SoC-ICS, Fall 2009 -- HW #1

Assigned: Aug 21st, 2009 Due: Sep 18th, 2009

Instructions:

- Please submit your solutions via Blackboard. Submissions should include a single PDF with the writeup and single Zip or Tar archive for source code.
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.

The SystemC environment is installed on the ECE LRC Linux servers, see:
http://www.ece.utexas.edu/~gerstl/ee382v_f09/docs/SystemC_setup.pdf
In short, you need to set the $SYSTEMC environment variable:
`setenv SYSTEMC /usr/local/packages/systemc-2.2.0` ([t]csh), or
`export SYSTEMC=/usr/local/packages/systemc-2.2.0` ([ba]sh)
You can then access the SystemC installation by referring to the '$SYSTEMC' variable.

Problem 1:

Get the attached *Hello* example running:

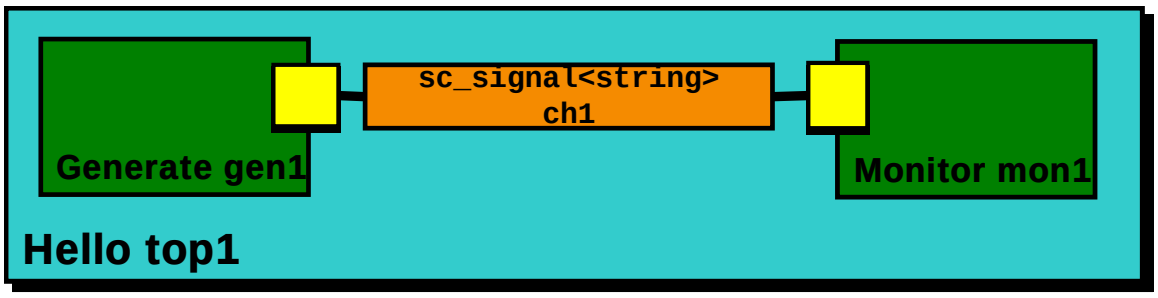
- Using your favorite debugger (`gdb` and `ddd` are good), walk through the behavior of the example.
- Generally describe the output and behavior of the design.

Problem 2:

Create a for-loop in the process to output the "Hello" message 10 times in bursts with a random delay between messages evenly distributed from 50 to 90 ns.

Problem 3:

Create two sub-modules, *Generate* and *Monitor*, connected by an `sc_signal<string>` channel. You will need an output port and an input port on each. Instantiate them inside *Hello*. Move the loop into the *Generate* module, but have it write to the output port. Have the *Monitor* display values that show up on the input port.



Appendix:

Sources for *Hello* example:

Hello.h

```
#ifndef Hello_h
#define Hello_h
#include <systemc>
SC_MODULE(Hello) {
  SC_CTOR(Hello);
  void end_of_elaboration(void);
  void Hello_thread(void);
  ~Hello(void);
};
#endif
```

main.h

```
#include "Hello.h"
#include <iostream>
using namespace std;
using namespace sc_core;
int sc_main(void) {
  Hello top_i("top_i");
  cout << "Starting" << endl;
  sc_start();
  cout << "Exiting" << endl;
  return 0;
}
```

Hello.cpp

```
#include "Hello.h"
#include <iostream>
using namespace std;
using namespace sc_core;
void Hello::Hello(sc_module_name nm)
: sc_module(nm) {
  cout << "Constructing "
       << name() << endl;
  SC_HAS_PROCESS(Hello);
  SC_THREAD(Hello_thread);
}
void Hello::end_of_elaboration(void) {
  cout << "End of elaboration" <<
endl;
}
void Hello::Hello_thread(void) {
  cout << "Hello World!" << endl;
}
Hello::~Hello(void) {
  cout << "Destroy " << name() <<
endl;
}
```

Available at

`/home/projects/courses/fall_09/ee382v-17248/hw/hw1.zip`