ECPS 203 Discussion Week3

TA: Emad Arasteh
emalekza@uci.edu
ecps203@eecs.uci.edu

Office Hours: Fri, 10:00-11:00am
EH 3404 Zoom 989 2181 4881

Center for Embedded and Cyber-Physical Systems
University of California, Irvine

Outline

- Review assignment 2
- Assignment 3
- UVM test bench architecture (optional)
  - Industrial application
- Questions

Common mistakes

- Overall many good deliveries!
- Missed removing command-line argument support
- Missed removing dynamic allocation related codes
  - Forgot to remove calls to `free()` after removing `malloc/calloc()` -> causes undefined behavior
  - Forgot to remove **ALL** `malloc/calloc` function calls
- Missed to apply off-by-one error in `non_max_supp()`

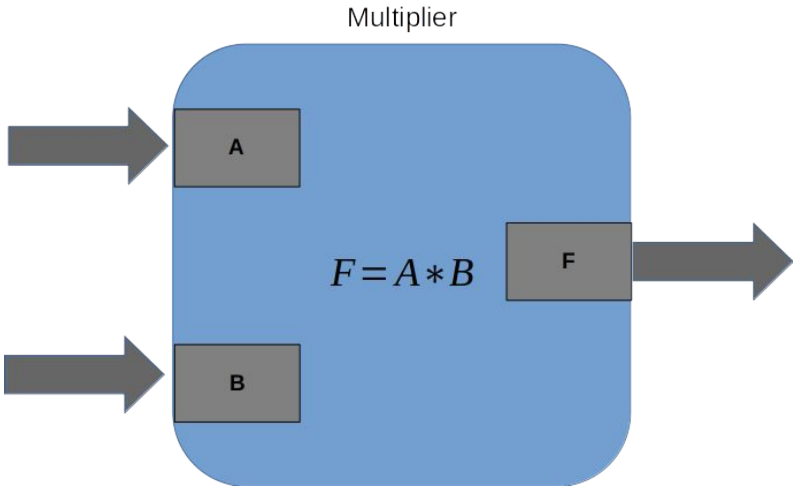
Assignment 3

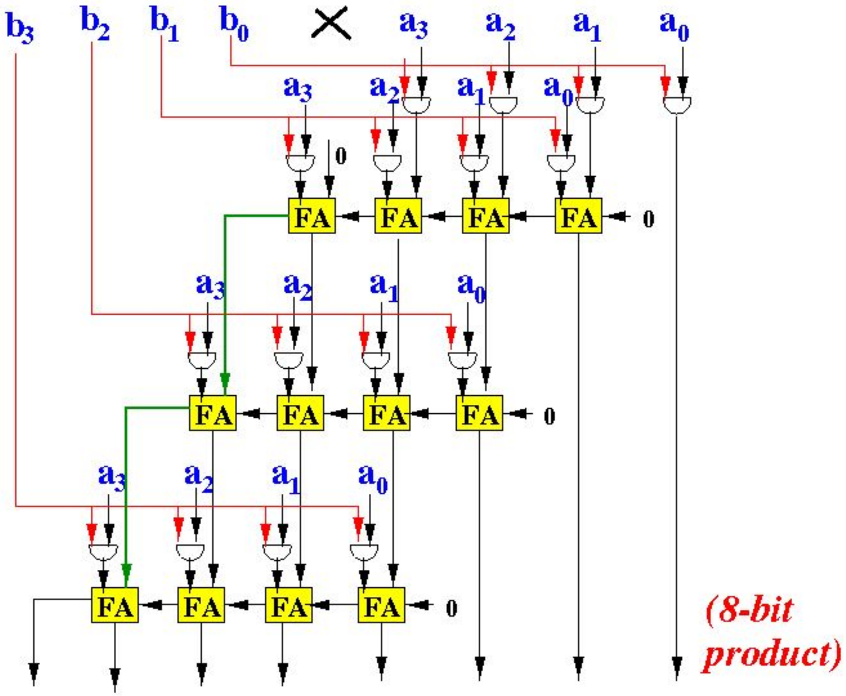
- Build a SystemC model of an integer multiplier
- Build a test bench to verify functionality of design under test (DUT)
- Simulate the model and test bench using Accellera SystemC library
- Create a package for submission
- Makefile to compile and simulate is already provided:

```
○ > cp ~ecps203/public/MakefileA3 Makefile
○ > make all
○ > make test
```

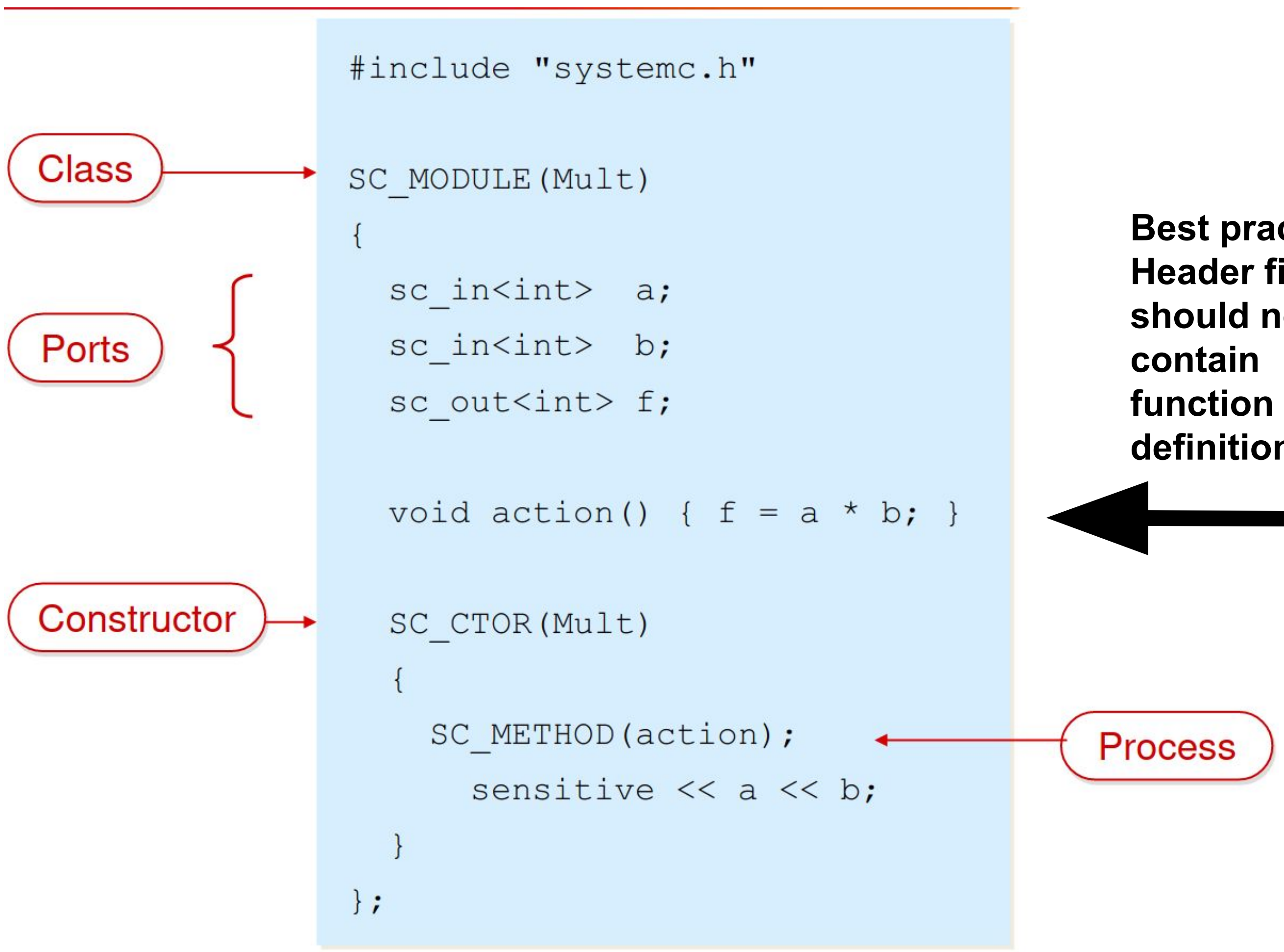
Multiplier

- Two input ports, one output port and one process
- System-level modeling abstracts implementation details of a multiplier





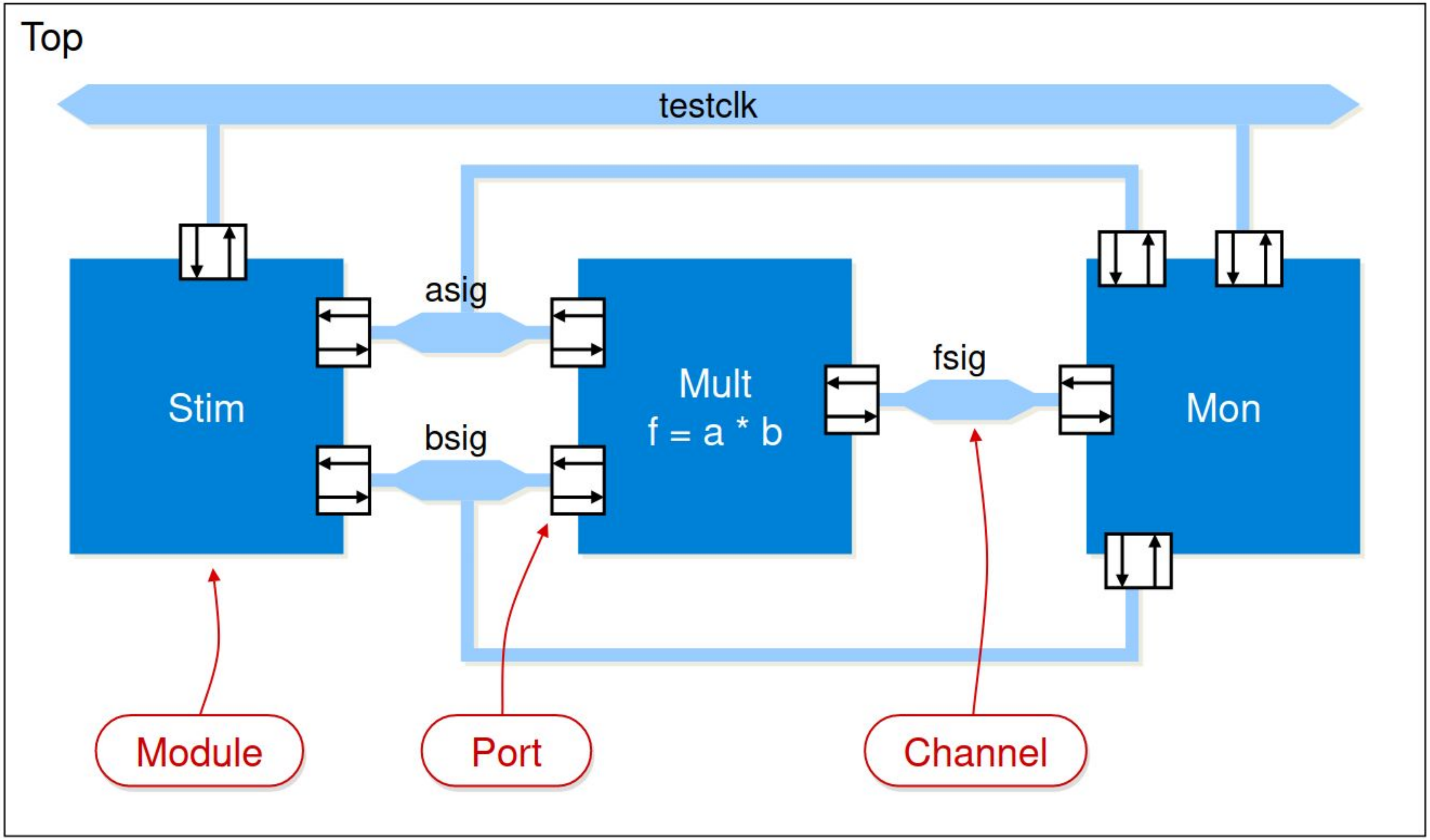
SystemC Multiplier Module



```
#include "systemc.h"

SC_MODULE(Mult)
{
  sc_in<int>  a;
  sc_in<int>  b;
  sc_out<int> f;

  void action() { f = a * b; }

  SC_CTOR(Mult)
  {
    SC_METHOD(action);
      sensitive << a << b;
  }
};
```

Best practise: Header file should not contain function definition!

SystemC Test bench

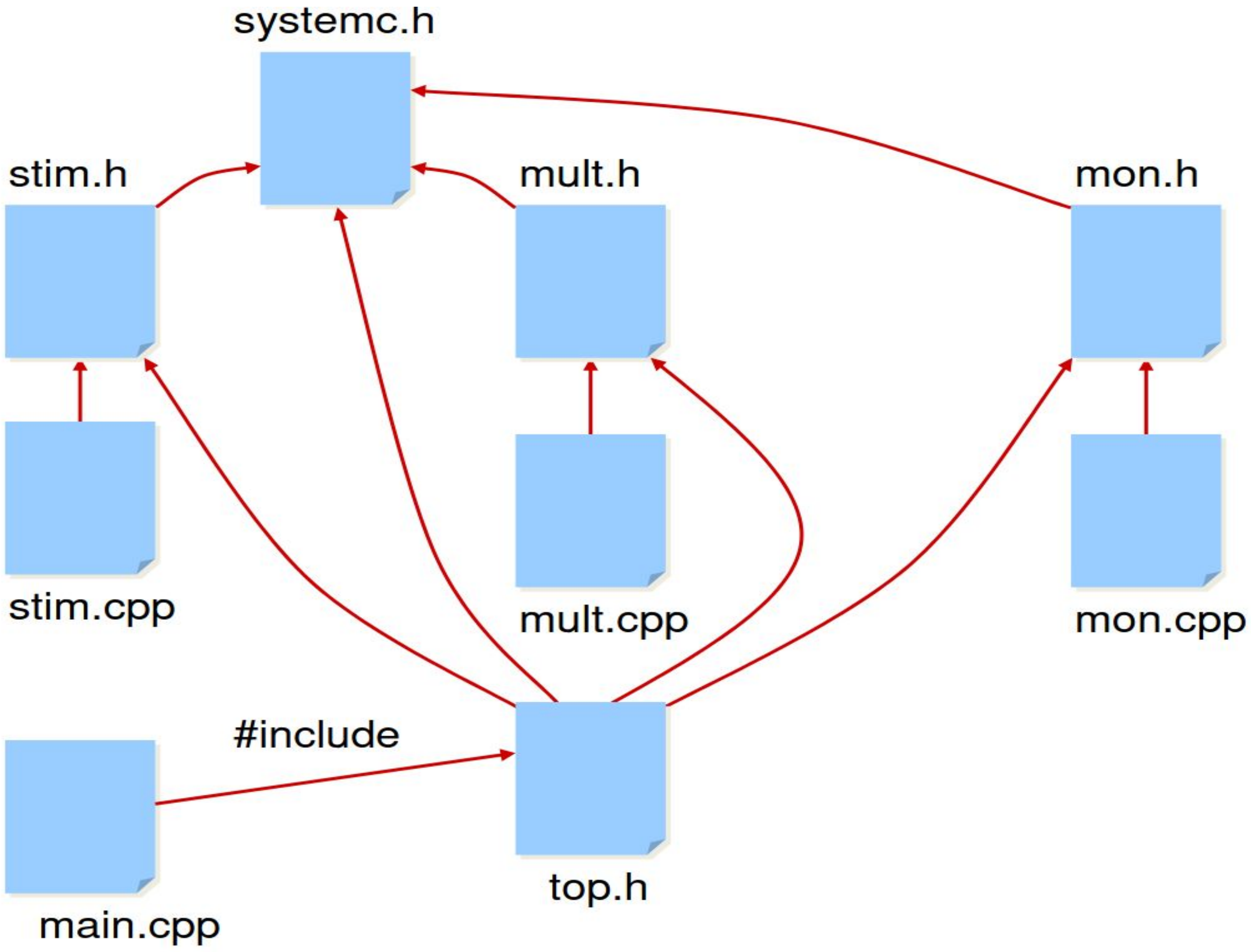


Stimulus and Monitor

- Stimulus module **generates stimulus vectors** and monitor module **samples DUT output** for design validation
- Apply stimulus on the **rising** edge of the clock and sample the output on **falling** edge of the clock

- Similar test bench structure applies to complex SystemC DUTs or even SystemVerilog DUTs such as CPU/GPU, memory controller, etc

File Structure

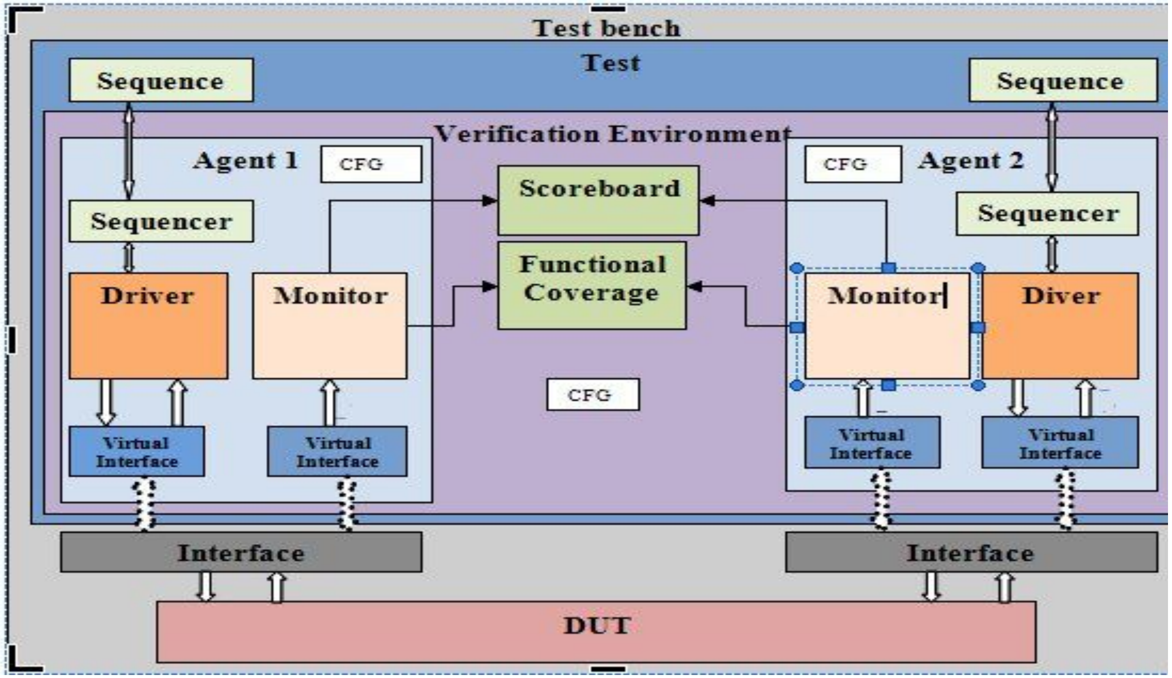


Deliveries

- Make a tarball of all source files, README and Makefile
  - `> gtar cvzf hw3.tar.gz README.txt Makefile *.cpp *.h`
- README: description of the model and any issues you encountered
- To submit, type:
  - ~ecps203/bin/turnin.sh (tilde key)
- To verify your submission, type:
  - ~ecps203/bin/listfiles.py

UVM test bench architecture (optional)

- The Universal Verification Methodology (UVM) is a standardized methodology for verifying integrated circuit designs
- UVM-SystemC is a new standard to develop structured verification environments following the UVM



Questions?