ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

# *Scoot:* A Tool for the Analysis of SystemC Models

Nicolas Blanc, Daniel Kroening, and Natasha Sharygina

www.verify.ethz.ch/scoot

- *Scoot* is a tool for static analysis of SystemC

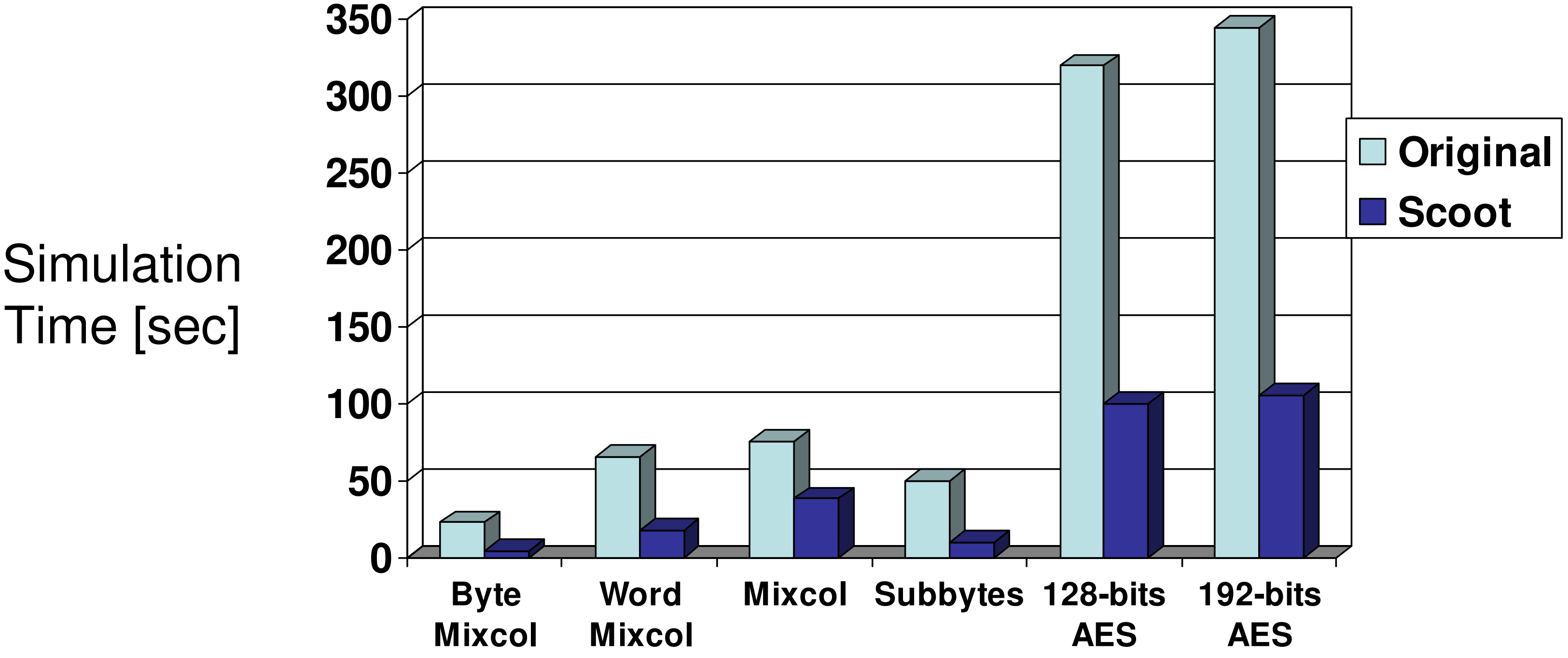
- This talk:

  *The benefits of static scheduling for SystemC*

# Outline

- Motivation: Simulation speedup
- Overview of SystemC
- Static Scheduling
- Demo
- Conclusion

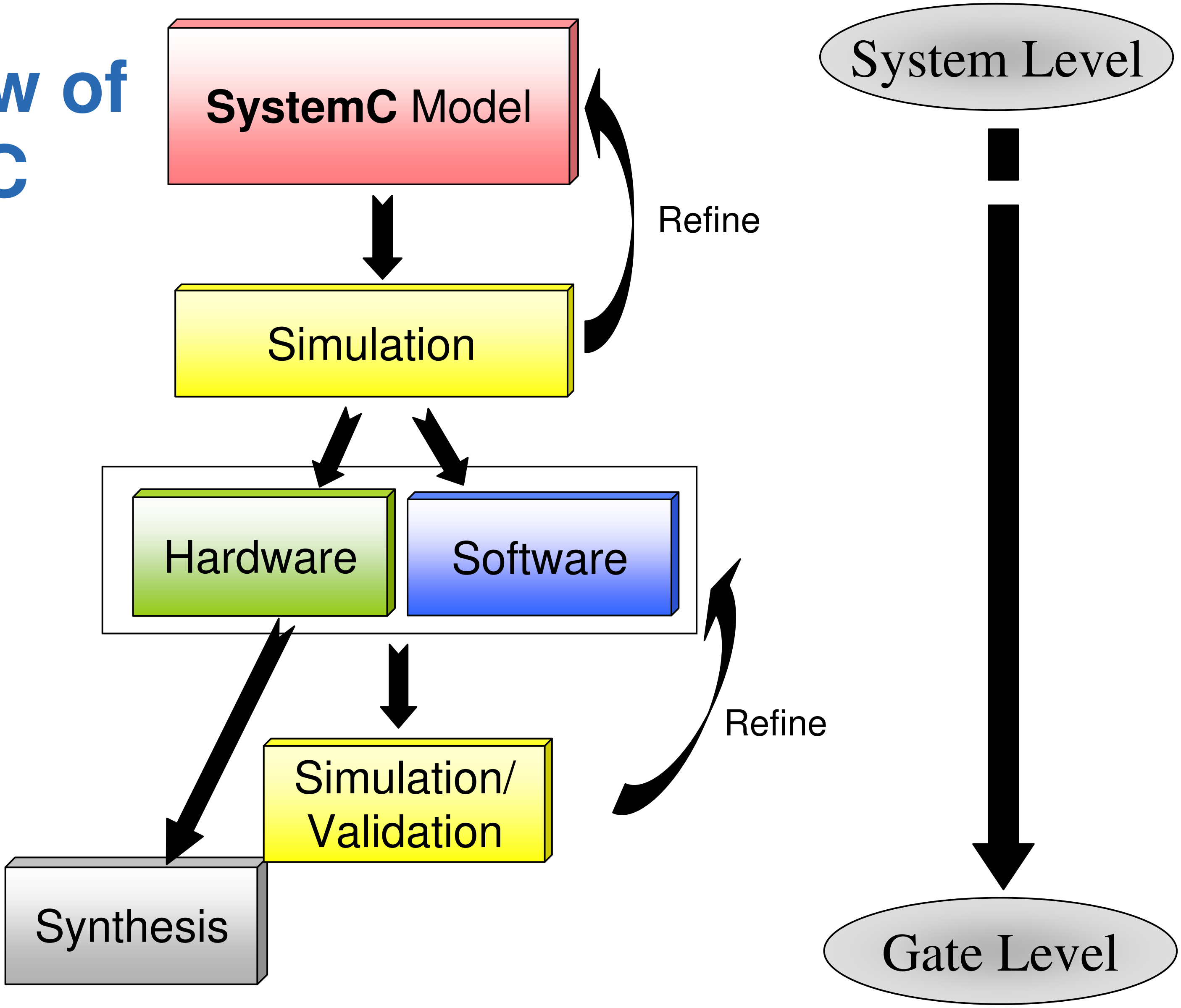
# Motivation: Benefits of Static Scheduling for SystemC



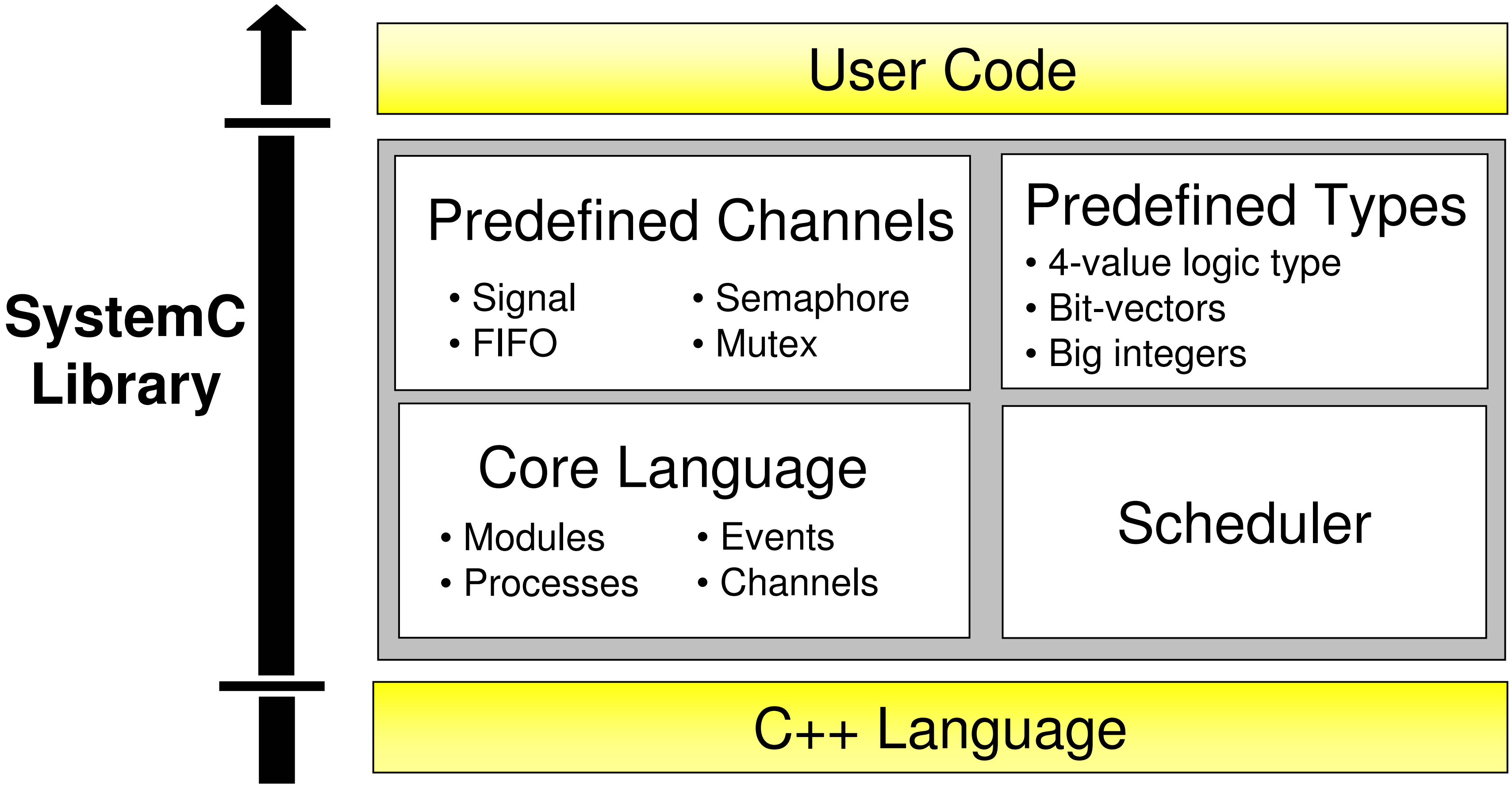
Oct. 2007, 3Ghz Linux, *g*++ 4.0.3

Benchmarks available at www.verify.ethz.ch/scoot

# Overview of SystemC



# Overview of SystemC



User Code

SystemC Library

Predefined Channels

- Signal
- FIFO
- Semaphore
- Mutex

Predefined Types

- 4-value logic type
- Bit-vectors
- Big integers

Core Language

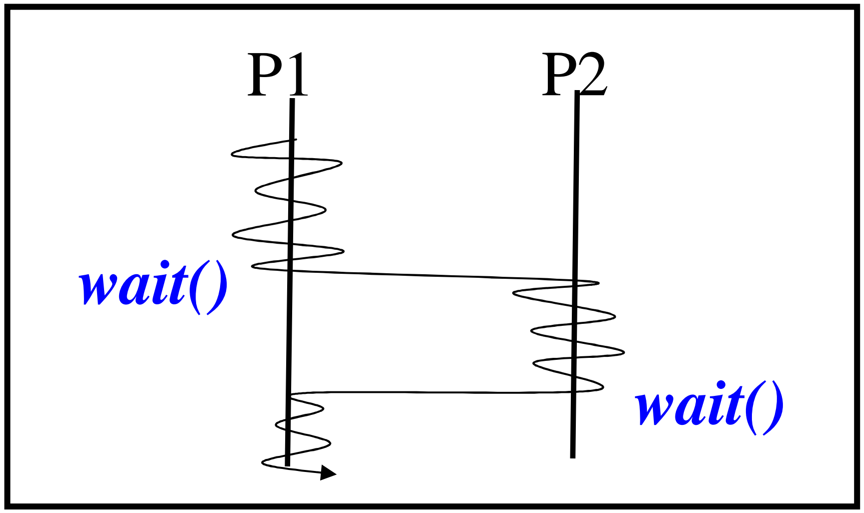
- Modules
- Processes
- Events
- Channels

Scheduler

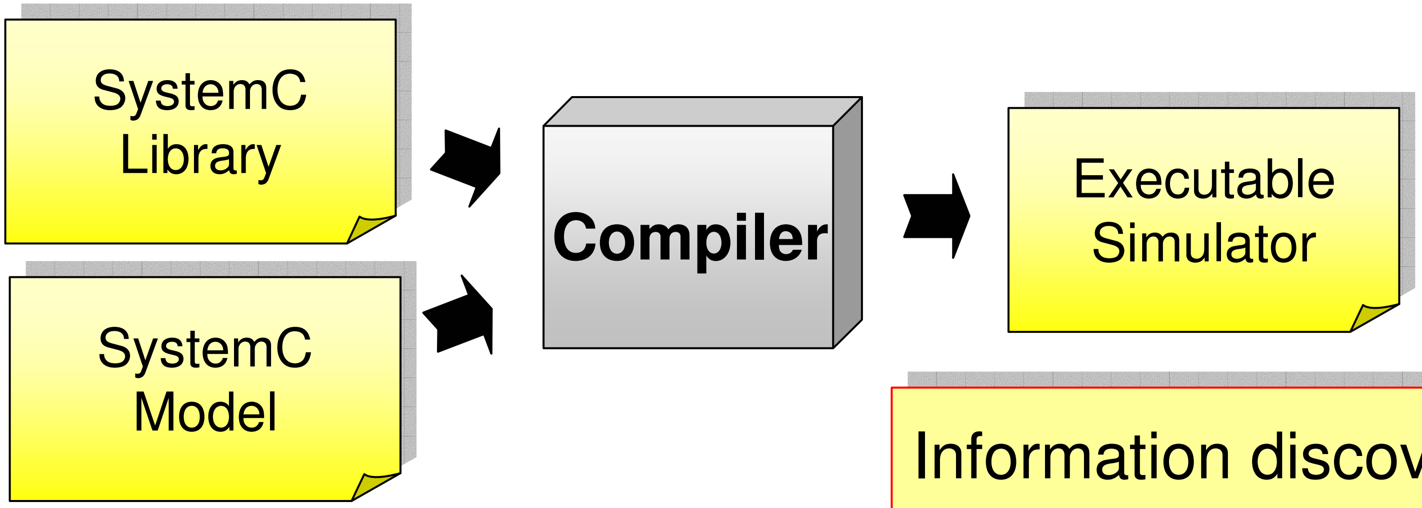
C++ Language

# The Concurrency Model of SystemC

- Execution driven by *events*
- Cooperative Multitasking Model:
  - One process running at a time
  - **No preemption!**



# SystemC Simulation: Standard Approach



**Simulator:**

- Fast execution of processes ✓
- Not fast enough. Compiler is missing opportunities ✗

Information discovered at **simulation time**:

- Module Hierarchy,
- Processes,
- and Port binding

# Scoot

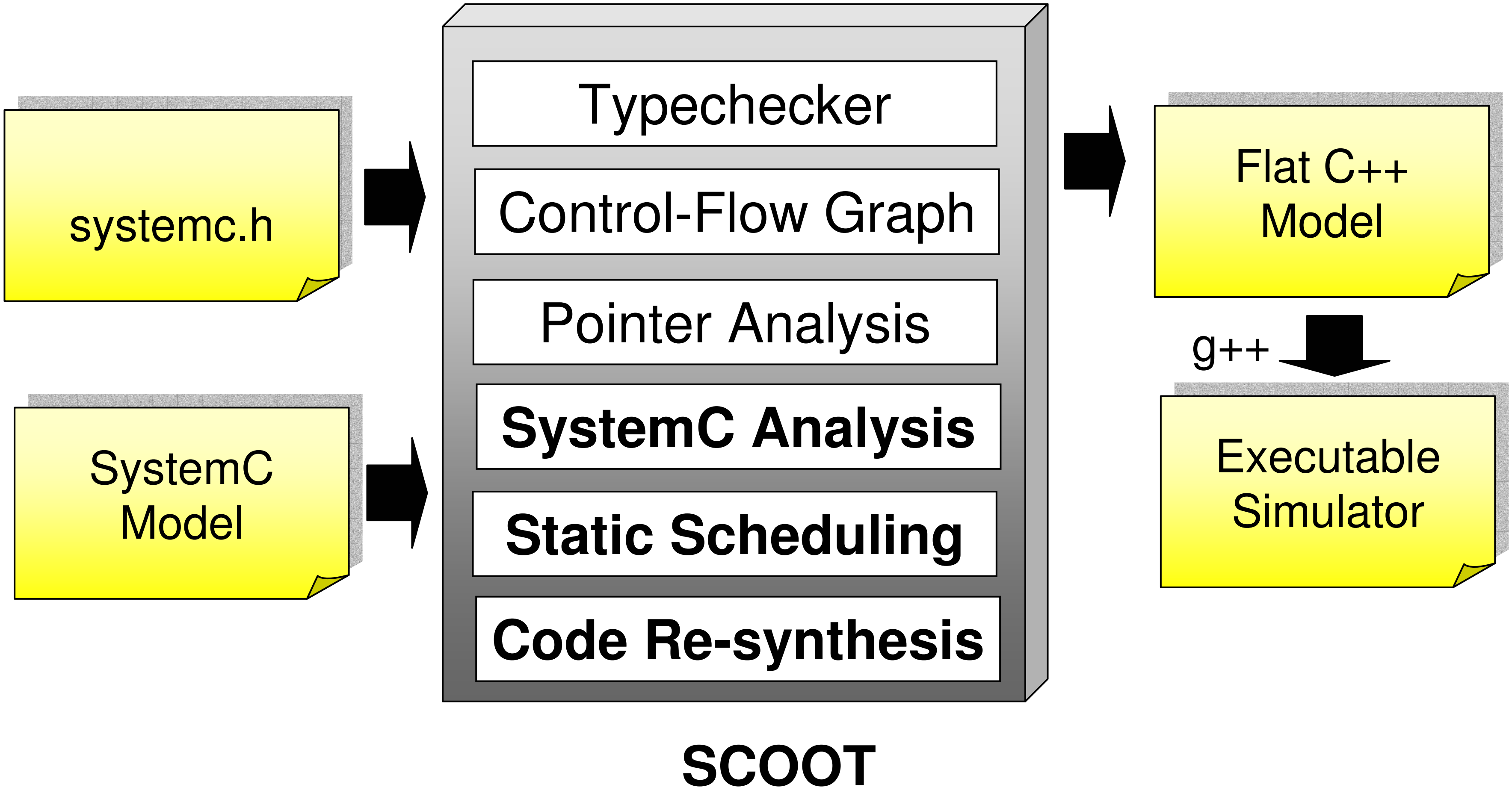
- Scoot *statically* discovers:
  - module hierarchy,
  - port bindings,
  - processes,
  - and sensitivity of the processes
- Simulation benefits from Static-Scheduling:
  - Resolution of dynamic calls
  - Static scheduling → no dynamic data structures.
  - Fast context switches using goto statements.

# Static Scheduling

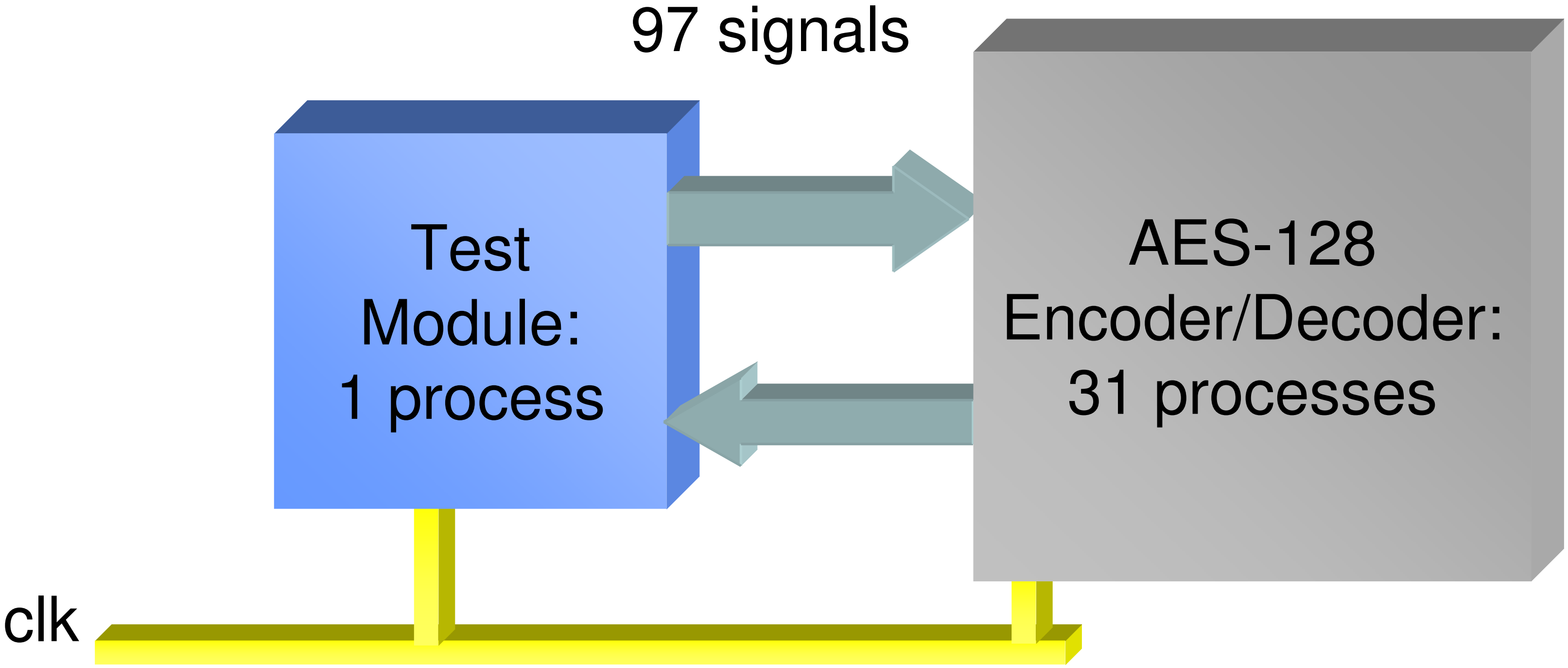
```
SC_MODULE(modelt){
  sc_clock clk;
  sc_signal<bool> i, o;

  void  inv(){ o = ! i.read(); }

  void  test() {
   while(true)
   {
     i = ! o.read();
     wait();
     ;   // skip
     wait();
   }
  }

  SC_CTOR(modelt) {
   SC_METHOD(inv); sensitive << i;
   SC_THREAD(test); sensitive << clk;
  }
};
```

```
if (i.value_changed()) inv();

if(!clk.value_changed()) goto TEST_END;
if(pc == 1 ) goto PC1;
if(pc == 2 ) goto PC2;
TEST_BEGIN:
WHILE:
        i = !o.read();
        pc = 1;
        goto TEST_END;
PC1:  ;   // skip
      pc = 2
      goto TEST_END;
PC2:
      goto WHILE;
TEST_END:
 ...
```

# Overview of Scoot



# Demo: AES-128



*Encryption/Decryption of 700 messages of 128 bits*

# Conclusion

- Formal analysis of SystemC is difficult.
- *Scoot* translates SystemC/C++ code into a simpler representation.
- Applications: Verification, Simulation

- http://www.verify.ethz.ch/scoot