EECS 222C: System-on-Chip Software Synthesis Lecture 4

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

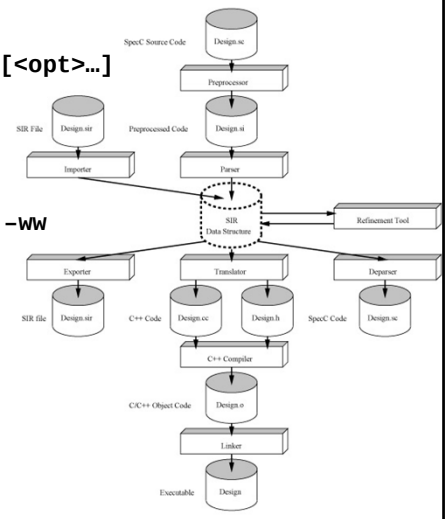
Lecture 4: Overview

- Assignment 2
  - SpecC Compiler, Simulator, Tools
  - Discussion
- Assignment 3
- Embedded Software
  - Execution time
  - Scheduling

The SpecC Compiler and Simulator

- SpecC Compiler
  - Command line interface
  - Usage: **`scc <design> [<cmd>] [<opt>…]`**
  - Help: **`scc –h`**
    `man scc`
  - Example:

```
% scc HelloWorld –sc2out –v –ww
scc: SpecC Compiler V 2.2.1
(c)2010 CECS, UC Irvine
Preprocessing...
Parsing...
Translating...
Compiling...
Linking…
Done.
```



The SpecC Compiler and Simulator

- SpecC Simulator
  - Execution as regular program
  - Example:

```
% ./HelloWorld
Hello World!
```

  - Simulation library
    - Access via inclusion of SpecC header files
    - Example: Print the current simulation time

```
#include <sim.sh>
...
sim_time t;
sim_delta d;
sim_time_string buffer;
...
t = now();  d = delta();
printf("Time is now %s pico seconds.\n", time2str(buffer, t));
printf("(delta count is %s)\n", time2str(buffer, d);
waitfor 10 NANO_SEC;
printf("Time is now %s pico seconds.\n", time2str(buffer, t));
...
```

The SpecC Compiler and Simulator

- SpecC Command Line Tools
  - Tools working with SpecC Internal Representation (SIR) files
  - Example:

```
% scc Adder -sc2sir -o Adder.sir
% sir_list -t Adder.sir
behavior ADD8
behavior AND2
behavior FA
behavior HA
behavior Main
behavior XOR2
% sir_tree -bt Adder.sir FA
behavior FA
|------ HA ha1
|       |------ AND2 and1
|       \------ XOR2 xor1
|------ HA ha2
|       |------ AND2 and1
|       \------ XOR2 xor1
\------ OR2 or1
```


Assignment 2

1. Practice the use of SpecC Command Line Tools
   - Setup
     - **source /opt/sce-20100908/bin/setup.csh**
   - Examine simple examples
     - **mkdir simple_tests**
     - **cd simple_tests**
     - **cp $SPECC/examples/simple/* .**
     - **ls**
     - **vi HelloWorld.sc**
   - Practice the compiler
     - **man scc**
     - **scc HelloWorld –sc2out –vv -ww**
   - Practice the simulator
     - **./HelloWorld**
   - Practice the tools
     - **man sir_tree**
     - **scc Adder -sc2sir -o Adder.sir**
     - **sir_tree -bt Adder.sir FA**

Assignment 3

1. Setup and simulate a SpecC model of the MP3 Decoder
   - Setup and unpack source code
     - **`source /opt/sce-20100908/bin/setup.csh`**
     - **`tar xvzpf ~/EECS222C/mad_SpecC.tar.gz`**
     - **`cd mad_SpecC`**
     - **`ls`**
   - Compile the SpecC model
     - **`make clean`**
     - **`make`**
   - Execute the SpecC model
     - **`testbench testStream/spot1.mp3 spot1.pcm`**
     - **`diff spot1.pcm ../mad_C/spot1.pcm`**
   - Use decoded PCM files from reference C code as “golden” reference
     - **`cp ../mad_C/spot1.pcm reference/`**
     - **`cp ../mad_C/spot1_3K.pcm reference/`**
     - **`cp ../mad_C/classic1.pcm reference/`**
   - Simulate the SpecC model (using the provided **`Makefile`**)
     - **`make test`** (or: **`make test1`** to run only the first test)

Assignment 3

2. Analyze the specification model of the MP3 Decoder
   - Setup (as in step 2)
     - **`cd mad_SpecC`**
   - Generate a top-level SIR design file
     - **`make`**
     - **`ls –l testbench.sir`**
   - View some statistics of the model
     - **`sir_stats testbench.sir`**
     - **`sir_stats -a testbench.sir`**
   - Generate a hierarchy tree of the model
     - **`sir_tree -blt testbench.sir`**
   - Generate a “clean” single-file SpecC model
     - **`scc testbench –sir2sc –vv –sn –sl –psi –o testbench_gen.sc`**
     - Or simply: **`make testbench_gen.sc`**
     - **`vi testbench_gen.sc`**

Assignment 3

3. Is there any parallelism specified in the model? If so, where?
   - Find all behaviors that execute in parallel
   - For each parallel behavior, note
     - the name of the parent behavior
     - the names of the parallel child behaviors

- Deliverables
  - Names of concurrent parent behaviors
  - Names of parallel executing child behaviors

- Due
  - by Friday, Oct 22, 2010, at noon
  - by email to **`doemer@uci.edu`** with subject "EECS222C HW3"

Embedded Software

- Chapter 4, part 1, of *"Embedded System Design"* by P. Marwedel (Univ. of Dortmund, Germany), Kluwer Academic Publishers, 2003.

  - Prediction of Execution Times
  - Scheduling in Real-Time Systems

  ➢ **`Lecture4-es-marw-4a-scheduling.ppt`**