Flexim Simulator User Guide (Pre-release)

Min Cai

September 2, 2010

1 About Flexim

Flexim is an open-source, modular and highly configurable architectural simulator for evaluating emerging multicore processors. It can run statically compiled MIPS32 LE programs.

For the latest Flexim code, please visit the project's website on Github: http://github.com/mcai/flexim.

2 Key Features

1. Architectural
   - Simulation of a classic five-stage superscalar pipeline with out-of-order execution.
   - Multi-level memory hierarchy with the directory-based MESI cache coherence protocol.
   - Support for Syscall-emulation mode simulation (i.e., application only, no need to boot an OS).
   - Correct execution of several state-of-the-art benchmark suites, e.g., wcet_bench, Olden and CPU2006.
2. Non-architectural
   - Developed from scratch in the object-oriented system programming language D 2.0. Great efforts are made to advocate software engineering practices in the simulator construction.
   - A powerful infrastructure that provides common functionalities such as eventing, logging and XML I/O.
   - Pervasive use of XML-based I/O for architectural, workload and experiment configurations and statistics.
   - Easy to use. No scripting. Only required are a statically compiled simulator executable and a few XML files.

3 Development Progress

Category	Current Progress	
Functional Simulation	Int. Inst. Decoding & Execution	OK for wcet-bench, mst, em3d, etc.
	Fp. Inst. Decoding & Execution	OK for wcet-bench, mst, em3d, etc.
	System Call Emulation	OK for wcet-bench, mst, em3d, etc.
	MIPS LE ELF Exe. Loader	Can run statically compiled programs
Performance Simulation	Five-stage OoO pipelining	RUU-based; to be written
	Set-associative cache structure	OK
	Cache coherence	Being rewritten; in good progress
	On-chip interconnect	Planned
	Interface to external DRAM simulators	To be planned
Common Infrastructure	Eventing and callback mechanism	OK, pervasive use in existing code
	Categorized logging mechanism	OK, limited use in existing code
	XML-based I/O for configs and stats	OK
	Plotting and table generation for experiments	Planned

4 System Requirements

1. Make sure that you have a Ubuntu 10.04 Linux machine. Other popular Linux distributions may work as well if you are lucky enough.
2. Make sure that you have the latest DMD 2.0 compiler installed. If not, go to this page and download "dmd D 2.0 compiler 1-click install for Ubuntu": http://www.digitalmars.com/d/download.html.

5 How to Build and Run Flexim

1. Unpack the zip or tar file containing the Flexim source.
2. In the main directory of the distribution, you can
   - build Flexim using the command: '`make`';
   - remove all the built files using the command: '`make clean`'.

   By default, the flexim binary is placed in the bin/ folder.
3. Download and unpack cross-compiler-mipsel.tar.bz2 from http://github.com/mcai/flexim/downloads/. Use it to compile MIPS32 LE programs to be simulated by Flexim.
4. In the subdirectory build/, you can start simulation with the default simulation configuration using the command: "`./flexim`" or "`./flexim --experiment=<experiment-name>`". Benchmarks and experiments are specified in the subdirectory configs/benchmarks/ and configs/experiments/, respectively.
5. You can find configuration and statistics files in the configs/ and stats/ subdirectories, respectively. Some sample XML files are provided for your reference.
6. Useful tip: As with all other open source projects, you can learn more by digging into the Flexim source code.

6 Contact Information

If you have any questions, please feel free to contact: Min Cai <itecgo@163.com>.