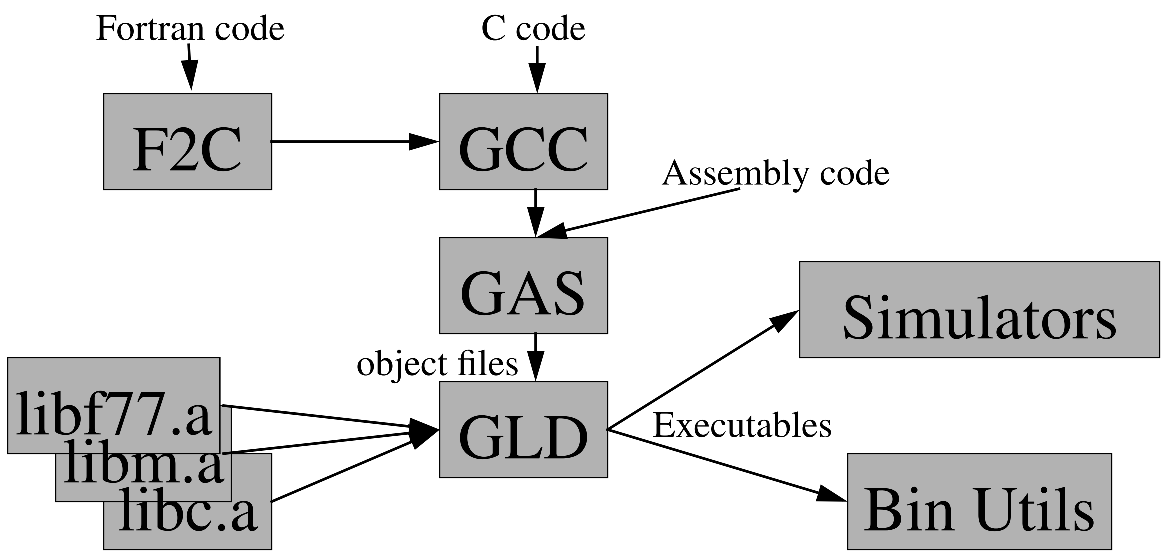
Tutorial Overview

- Overview and basics
- Using the tool suite
- How to use sim-outorder
- How to build your own simulators
- How to modify the ISA
- How to use the memory extensions
- SimpleScalar limitations and caveats
- Wrapup

SimpleScalar tool set overview



- Compiler chain is GNU tools ported to SS ISA
- Fortran codes are converted with AT&T *f2c*
- libraries are CLIBC ported to SimpleScalar

Getting up and running

- Simulators only
  - prebuilt binaries (SPEC95 and test)
  - prebuilt libraries
- Test benchmarks
  - compile SS gcc, binutils
- New ISA
  - modify ss.def
  - alter gcc, binutils

Sample commands

- running a simulator:

  sim-outorder [simulator opts] benchmark_name [bench opts]
- compiling a C program:

  ssbig-na-sstrix-gcc -g -O -o foo foo.c -lm
- compiling a Fortran program:

  ssbig-na-sstrix-f77 -g -O -o foo foo.f -lm
- compiling an SS assembly program:

  ssbig-na-sstrix-gcc -g -O -o foo.s -lm
- disassembling a program:

  ssbig-na-sstrix-objdump -x -d -l foo
- building a library:

  ssbig-na-sstrix-{ar,ranlib}

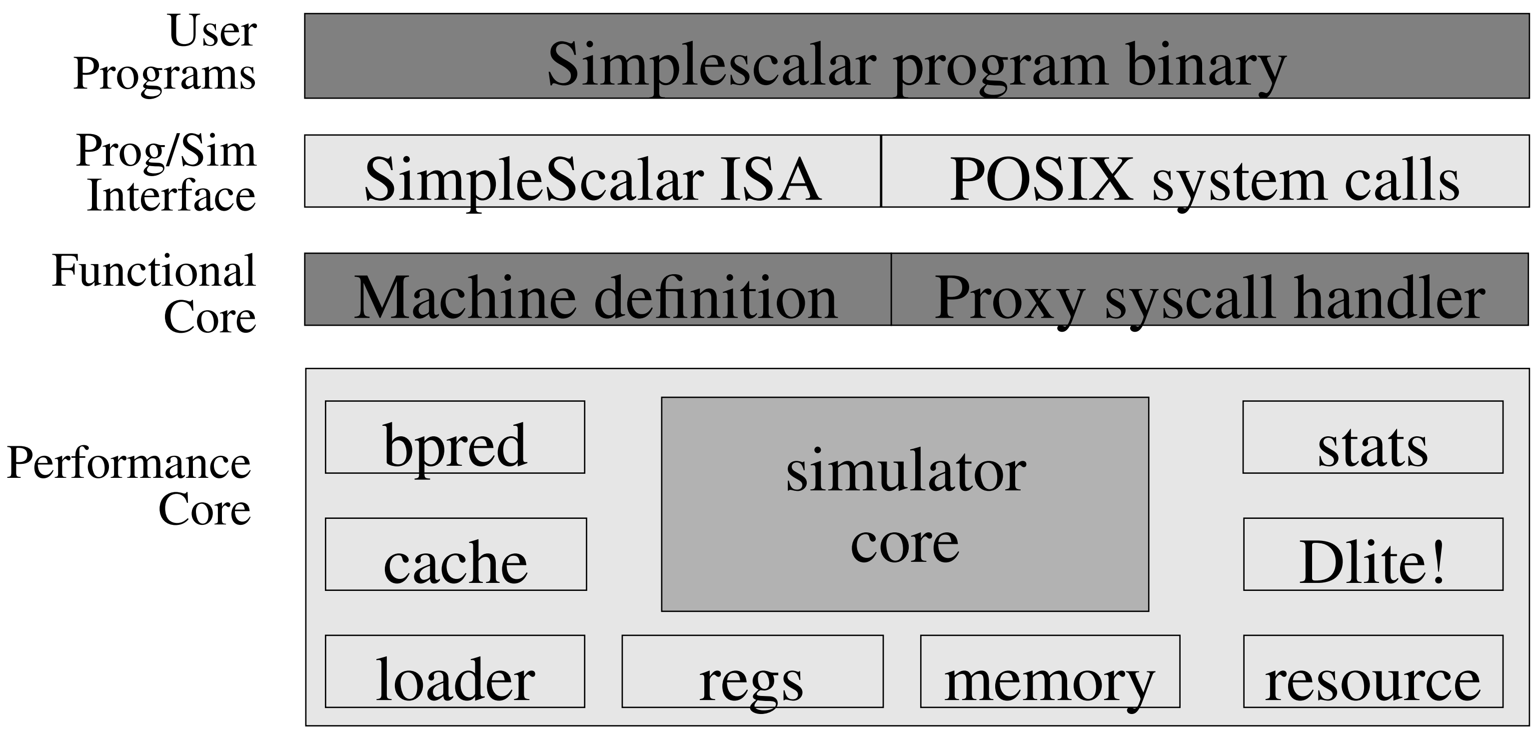
Global simulator options

- supported on all simulators:

`-h`	- print simulator help message
`-d`	- enable debug message
`-i`	- start up in DLite! debugger
`-q`	- quit immediately (use with `-dumpconfig`)
`-config <file>`	- read config parameters from `<file>`
`-dumpconfig <file>`	- save config params into `<file>`

- Comments allowed in configuration files ('#')

Simulator structure



- Most of performance core is optional
- Most projects will enhance “simulator core”

Simulation suite overview

sim-fast sim-safe	sim-profile	sim-cache sim-cheetah sim-bred	sim-eio	sim-outorder
300-400 lines functional 4+ MIPS	900 lines functional lots of stats	<1000 lines functional cache stats pred stats	func./timing reproducibility	~4000 lines performance OoO issue branch pred. mis-spec. ALUs cache TLB 200+ KIPS

performance ←

detail →