

Full System Simulator

- Simulates different new IBM architectures like PERCS, PowerPC 970 and Cell
- SDK includes Full System Simulator preconfigured for Cell
- All functions of the simulator can be accessed using a command line or the GUI
- Can be booted with Linux as OS or without OS (standalone)

Full System Simulator

- Files can be transferred to the simulator
 - temporarily using callthru
 - permanently mounting the `sysroot_disk`
- SPU can be run in
 - *Instruction mode*, for checking and debugging the functionality, or in
 - *Pipeline mode*, for collecting performance statistics
- PPE can be run in *Fast Mode* disabling system analysis features

Full System Simulator

- Various debug-messages can be printed to the terminal
- Registers, DMA, Memory and Local Store can be watched while the simulator is running (changes are highlighted)

Debugging - GDB

- Modified version of GDB source-level debugger – The GNU Project Debugger
- gdb for PPU-code and spu-gdb for SPU-code; PPU and SPU can not be debugged at the same time
- Makefile: “CFLAGS= -g”
- Source-file has to be copied to the simulator



Profiling - OProfile

- Statistical, kernel-based profiler
- Not yet available for Cell
- Profiles hardware and software interrupt handlers, kernel modules, kernel, shared libraries, and applications

Profiling – static timing analysis

- SDK includes *spu-gcc_timing*, a static timing analyzer for the SPU
- Usage: *make sourcefile.s*
spu-gcc_timing sourcefile.s
sourcefile.s.timing
- Annotates machine code and shows
 - dependency stalls
 - dual and single-issue

Profiling – dynamic timing analysis

- In pipeline mode simulator collects performance characteristics of the SPUs
- Number of cycles, instructions, CPI, branch misses, dependency stalls and register use are collected
- Can be reset, started and stopped by the SPU program