

6.004 Computation Structures
Spring 2001

Bsim Documentation

bsim is a simulator for *Beta*, the 6.004 Instruction Set Architecture. Bsim may be invoked by typing "bsim" at the Athena command prompt if you've added the 6.004 locker. This causes bsim to begin in a reset state with its memory clear. More commonly, bsim will be used to debug a program generated from a file with a name like foo.uasm by the command lines

```
uasm foo          (produces foo.sym, foo.bin and foo.map)
bsim foo
```

The main screen of bsim displays processor state (registers, PC) as well as a region of the stack near its top. As many as 16 additional memory locations may be displayed as well. All displayed data is in *hexadecimal*, although decimal, binary, and symbolic user input is allowed. The main screen also includes the current line of the source program being executed, if this information is available.

Single-character commands are accepted while the main screen is showing. These include:

- ? or h Help. This is the only command you really need to remember. It displays a single-screen command summary, and waits for a character to be typed before returning to the main screen.
- q or ^C Quit. Returns to UNIX, abandons simulation.
- SPACE Single step: executes one simulated instruction.
- g Go: steps through the program, updating the display as successive instructions are simulated. You may stop by typing ^G or SPACE. Execution also stops when an error, HALT() instruction, or breakpoint (below) is encountered.
- G Go faster. Like g, but updates the display infrequently. This allows progress to be monitored imprecisely, along with a substantial speed improvement.
- ^G Halts execution.
- B Set a breakpoint. Asks for an address; you respond with a typed value (or symbolic expression) to establish a breakpoint, or return to clear any existing breakpoint. When a breakpoint is set, the simulator stops g or G mode execution immediately *prior* to each execution of the instruction at the specified address.
- b Run silently to breakpoint.
- ^N or ^P Move cursor downward (^N) and upward (^P) in the display of monitored memory locations.
- w Watch a specified memory location. Requests an address to be monitored, using the current screen position, which is marked by the = symbol.
- d Don't watch location; stops monitoring location at the current position, which is marked by the = symbol.

- n Watch Next location. Moves to next screen location, which gets becomes devoted to watching memory location following that of the current screen location.
- R Reset simulator to power-up configuration. Clears registers and DRAM; reloads binary file into DRAM.
- ` Switch to and from Terminal mode. Terminal mode is used for debugging programs which perform console I/O; it allows the single screen or window of your workstation to be multiplexed between the main bsim screen (showing simulated state) and the simulated console used by the program being executed. When in terminal mode, console user input (with the exception of the “” character) goes to the simulated program rather than to the simulator. To return to the simulator, type another “”. Console output from the simulator always goes to the terminal-mode screen; you must enter terminal mode to see output from the program.
- c Interact with cache simulator. See below.
- ^L Refresh display.
- L Load a file into bsim; prompts for primary name of file. Not needed if you specify a filename on the bsim command line, the usual UNIX usage; provided for PC/Mac version compatibility.
- e Evaluate an expression, and report its value in hexadecimal and decimal. Useful for finding symbol values, radix conversion, etc.

Expressions read by bsim's w, B, and e commands can contain any of the following forms:

- 12345 Decimal constants. **Warning:** Although data is displayed in hexadecimal, numbers typed in without the 0x prefix are interpreted as decimal.
- 0xBAD1 Hexadecimal constants, with C-style 0x prefix.
- 0b010 Binary constants.
- foo Symbols from .uasm assembly.
- + - * / & ! Operators.
- () Parentheses.

Cache simulation

Bsim allows the simulation of a single cache between the processor and main memory, and keeps a simple summary of hit/miss statistics to allow performance evaluation of various cache configurations during simulated execution. The “c” command from the main screen enters a cache configuration screen which displays the current cache parameters and performance history, and which accepts a number of single-character commands allowing the user to vary the parameters. These include:

- q Quit. Return to main simulator screen.
- ^N or ^P Move cursor downward (^N) and upward (^P) in the display of cache parameters. The selected parameter is noted by an equal sign on the display.
- SPACE Change the selected parameter. Cycles through legal values.
- r Reset cache statistics.

- R Reset and invalidate cache entries. Sets all valid bits to 0, thereby discarding cache contents. Note that this may discard important information in the case of a write-back cache.
- s Synchronize memory with cache. Causes any dirty cache lines to be written back into main memory. Note that until such a writeback is performed, the memory display on the main simulator screen may contain stale values.
- t Enable/disable trace file output. When enabled, a voluminous ASCII file containing a line for each cache transaction is output during execution. This feature is primarily for debugging of the cache simulator itself; its use otherwise is not recommended.

Cache configuration parameters which may be adjusted include:

ON/OFF	Whether the cache is in use.
Set size	Selections include non-associative direct-mapped configuration, or N-way set-associative for N from 2 to 8.
Number of lines	Varies the number of cache lines. Note that the total number of lines for an N-way cache is N times the number entered here.
Line size	Varies the data block size, i.e. the number of data words stored in each cache line.
Replacement	varies the replacement strategy for set-associative configurations. Choices include {\tt cycle}, using a single cyclic counter incremented at each replacement; Least Recently Replaced; Least Recently Used; and Random.
Write strategy	may be varied between write-through and write-back. Note that the latter choice allows stale values to be displayed as contents of main memory. Dirty bits are kept in each cache line to eliminate redundant write-backs.

Reconfiguration of the cache causes its contents to be lost, and all lines to be initialized in an empty state. Reinitialization is also implied by the R (reset) command to the main simulator.