Defining Test Input Stimuli (Forces)

A *"force value"* is a value-delay pair that defines a stimulus to be applied to a signal and the time at which it is to be applied. The time is expressed as a delay from the current simulation time. Multiple time-delay pairs may be specified with a single *"force"* command, and a pattern may be repeated if desired.

• Force commands may be typed in the main window or read from a command file. The format of a force command is as follows.

force signal-name value1 time1, value2 time2, ... [-rep]

If the optional *-rep* n is specified, the pattern is repeated with a period of n time units.

• Forces may also be specified via the "Force" button in the Signals Window.

Examples: (Assume current time is T)

Force signal "abc" to logic value '1' at time T+20:

force abc 1 20

0

Force "abc" to '0' at time T, '1' at T+20 and then '0' at T+40:

force xyz 0 0, 1 20, 0 40

Create a clock waveform on signal "clk" with period 40, with value '0' for the first half and '1' for the last half:

force clk 0 0, 1 20 -rep 40

• Numeric force values can be expressed in any radix by prepending the prefix *radix#* to the number.

Example: The decimal number 123 may be expressed as:

- Decimal: 123 or 10#123
- Hexadecimal: 16#7B
- Binary: 2#1111011

• Logic states "Z" and "X" may only be written as part of a binary string.

Example: To float a 32-bit data bus at time 25:

• A signal within a lower-level components of a hierarchical design is specified via its full "path name", which is similar to the path name of a UNIX file.

Examples:

```
clk - signal clk in the current environment
/clk - signal clk in the top-level model
/blk1/clk - signal clk within block blk1
/blk1/blk2/clk - signal clk within block blk2 of block blk1
```

• Force values are delayed from the "current time", allowing a simulation to be run in multiple steps.

Example Procedure:

- 1. Define force values for step 1, relative to time 0.
- 2. Run the simulation for T1 time units (time is now T1).
- 3. Define force values for step 2, relative to time T1.
- 4. Run the simulation for T2 time units (time is now T1+T2). etc.

ModelSim EE Simulator Command Summary

ABORT

```
Halt execution of a macro file C_{1}^{(1)}
```

```
BD <filename> line>
```

Delete breakpoint at indicated line of VHDL source file

```
BP [<filename>] [<line>] [<command> {;<command>}]
```

Set breakpoint at indicated line of VHDL source file and execute one or more simulation commands. List all breakpoints if no filename/line given.

```
CHANGE <variable> <value>
```

Change the value of a variable or signal. (Simulator must be paused at a breakpoint or after STEP)

CONTINUE

Continue execution after a STEP command or breakpoint.

DESCRIBE <variable_name>

Display info about the type of a variable or signal

DO <filename> {cparameter_value>...}

Execute commands from a macro file. Can pass up to nine parameters, designated in the macro file as \$1 ... \$9.

DRIVERS <signal_name>

Display the current value and scheduled future values for all drivers of the specified signal.

ECHO <text_string>

Display a message in the Transcript window.

ENVIRONMENT [<pathname>]

Display or change the current signal environment.

EXAMINE <name>

Display current value of named signal. (Simulator must be paused at a breakpoint or after STEP)

EXIT [-FORCE]

Exit simulator. If -FORCE specified, then quit without asking if you want to save data.

FIND [-RECURSIVE] [-IN] [-OUT] [-INOUT] [-INTERNAL] [-PORT] <signal_name> Display full pathnames of all signals whose names match the signal name specification.

FORCE <signal_name> <value> [<time>] {,<value> <time>} [-REPEAT period] Force signal to indicated values at indicated times. Optionally repeat the pattern. (eg. for clocks)

Example

```
FORCE x1 0 -- set x1<=0 now FORCE x1 0 100
set x1<="1" 100 time units hence FORCE x1 0 10, 0 20 Rep 40 set
x1<="0" at time 10, 1 at time 20 then repeat every 40 time units
FORMAT LIST <filename> and
```

FORMAT WAVE <filename>

Record list of signal names currently in List/Wave window to a file. This file may be invoked with a DO command to recreate the window.

IF <expression> {<command>} {;<command>} {ELSE {<command>} {;<command>} ENDIF

Execute selected commands if certain conditions are true.

LIST [-STROBE [<first_strobe>,]<period>] [-COLLAPSE] [-DELTA | -NODELTA] [{-RECURSIVE -IN -OUT -INOUT -INTERNAL -PORT -<radix> -TRIGGER -NOTRIGGER -WIDTH <n> {<signal_name> ...}]

List signals and their values in the *List Window*.

LOG [-RECURSIVE] [-IN] [-OUT] [-INOUT] [-PORT] [-INTERNAL] <signal_name> {<signal_name>...}

Create a log file containing simulation data for all listed signals. All signals in List/Wave Windows are automatically included. (The log file is the source of data for List/Wave windows.)

NOFORCE <signal_name> {<signal_name>...}

Cancel a repeating force on a signal.

```
NOLIST <signal_name> {<signal_name>...}
```

NOWAVE <signal_name> {<signal_name>...} Remove signals from List/Wave Window. ONBREAK [<command>; {<command>}]

Specify command to be executed when a breakooint is encountered during simulation.

QUIT [-FORCE]

Exit simulator without saving data, after confirmation. (Confirmation not requested if -FORCE specified.)

RESTART [-FORCE] [-NOBREAKPOINT] [-NOLIST] [-NOLOG] [-NOWAVE] Reload design and reset simulation time to 0.

-FORCE => without confirmation

-NOBREAKPOINT => discard breakpoints (otherwise keep)
-NOLIST => do not keep List window (otherwise keep)
-NOLOG => do not keep log file (otherwise keep)

-NOWAVE => do not keep Wave window (otherwise keep)

RESUME

Resume execution of a macro file after a breakpoint.

RUN [<time_units>...] [-all]

Execute the simulation for the specified number of time units. If no time specified - run for default time. If -all specified, run until there are no events scheduled.

SET [<variable_name> <value>]

Create a user-defined variable and set it to the indicated value. If no argument - display all current user-defined variables.

SHIFT

Shift macro parameter values down one place. (Parameter \$2 is assigned to \$1, \$3 to \$2, etc.)

SHOW [<pathname>]

List signals and subregions visible from the current signal environment.

SOURCE [<filename>]

Display VHDL source file.

STEP [-OVER]

Step to the next VHDL statement. If -OVER specified, treat procedures/functions as single statements (do not step through them).

SYSTEM <unix_command>

Execute one UNIX shell command.

Ex. >SYSTEM cp vector2.dat vectors.dat

TΒ

Display stack trace for the current process.

TRANSCRIPT [ON | OFF]

Echo to Transcript window commands being executed from a macro file. VIEW [LIST] [PROCESS] [SIGNAL] [SOURCE] [STRUCTURE] [VARIABLE] [WAVE] Open one or more designated simulation windows. "VIEW *" will open ALL of the above windows.

WAVE {[-RECURSIVE] [-IN] [-OUT] [-INOUT] [-INTERNAL] [-PORT] [-<radix> - <format> [-OFFSET <n>] [-SCALE <n>] [-COLOR <color_name>] {<signal name>...}}

Add signals to a Wave Window.

- -<radix> specifies signal display radix default is DECIMAL (BINARY, OCTAL, DECIMAL, HEX)
- -< format> specifies whether to use "logic", "literal" or "analog"
- -OFFSET <n> => offset and scale values for analog signals
- -SCALE <n>
- -COLOR => display color of the signal. Ex. >WAVE -logic -color gold signal1

WHERE

Display info about system environment (current directory, project file, and work library.)

WRITE LIST <filename>

Record contents of List Window in an output file.

WRITE REPORT <filename>

Print summary of design being simulated to output file.

WRITE WAVE <filename>

Record contents of Wave Window in an output file in PostScript format.