

CCLOCK and SETCLOCK tutorial

rdp

May 16, 1992

1 Introduction

This document describes *cclock*, an alternative to the standard clock function of the ldvsim simulator. Its principal features are

- provides facilities for reading input files and writing output files.
- runs either a single or two phase clock.
- generates test vector files.

The command *cclock* can be configured using the command *setclock*.

2 Usage

2.1 *cclock*

cclock MaxCycles EndCycles

If the argument *MaxCycles* is provided the simulator will execute *at most* MaxCycles. The default value for this parameter is 1.

If the argument *EndCycles* is provided the simulator will execute *exactly* EndCycles after all modules that read from files have finished. The default value for this parameter is 0.

Synchronization between *cclock* and the modules is achieved through the external integer variable ALLDONE. The modules should increment the variable ALLDONE when they start reading and should decrement ALLDONE when they finish reading the file.

2.2 *setclock*

setclock [standard options] [test vector generation options]

2.2.1 Standard Options

Standard options relate to the output produced by the simulator during execution.

REPORT: This variable controls the mode of reporting nodes and vectors selected by the watch (w) function. Possible values are: 0 (no report), 1 (report at the end of each cycle, Phi2H is high), 2 (report at the time *clock* returns), 3 (report twice each cycle, at the end of Phi1H and at the end of Phi2H). The default value for this variable is 3.

WALK: This variable controls the way in which particular operators in the modules are executed. This allows the module to produce inputs and sample outputs at specific points in the clock cycle. Possible values are: 0 (no execution), 1 (produce/sample HS1 and HS2 type signals by executing all modules with a Stable1 or Stable2 operator), 2 (execute all signal type operators). The default value for this variable is 2.

EVENTS: This flag controls reporting of pending events. If it has nonzero value, at the end of the clock cycle, the simulator prints a list of the nodes which have pending events. The default value for this flag is 0.

STATE: outputs a vector to a file during the simulation. It may be sometimes useful to look at particular nodes during the simulation.

The options described above can be modified with *setclock* by specifying the:

[-R *integer*] changes the value of the report variable to *integer*.

[-W *integer*] changes the value of the walk variable to *integer*.

[-E *integer*] changes the value of the pending events flag to *integer*.

[-S *State Vector File*] specifies the *State Vector* and the *File* where it should be written.

2.2.2 Test Vector Generation Options

These options relate to the specification of the test vector file. A vector may consist of a single node but MUST be declared as a vector.

Phi: specifies the clock node.

Clear: specifies the reset node.

Test: specifies the node that controls the test function (scan path).

Input: specifies a *vector* of *input nodes*. Multiple definitions are allowed. The test vector file will contain a column for each *vector*.

Output: specifies a *vector* of *output nodes*. Multiple definitions are allowed. The test vector file will contain a column for each *vector*.

Input/Output: specifies a *vector* of *input/output nodes* and the *node* that enables the *vector* for output. It is assumed that the output is enabled when *node* is high. Multiple definitions are allowed. The test vector file will contain a column for each *vector*.

Test Name: specifies the name of the particular test.

Test Vector File: specifies the test vector file name.

Last Flag: initiates test vector generation. No change to any option is allowed after this option is executed.

The options described above can be modified with *setclock* by specifying:

[-P *node*] specifies the clock (Phi) node.

[-C *node*] specifies the reset (Clear) node.

[-H *node*] specifies the test enable node.

[-I *vector*] specifies *vector* as a group of input nodes.

[-O *vector*] specifies *vector* as a group of output nodes.

[-T *vector node*] specifies *vector* as a group of input/output (tri-stated) nodes that are enabled as outputs when *node* is high.

[-N *name*] specifies the name of the test.

[-V *name*] specifies the name of the test vector file.

3 Example

The following command file is an example of how to use the commands *clock* and *setclock*.

```
| Set initial values for the inputs
h ResethSR PCWrLLR PCrDLLR TestHSR Phi
l ScanInHSR PCCSLLR PCAdr0{0:9}HLR Hs4HLR Hs3HLR Hs2HLR Hs1HLR

| Define vectors.
vector inputs_control PCWrLLR PCCSLLR PCrDLLR
```

```

vector inputs_addr  PCAdr0{9:0}HLR
vector inputs_h1    Hs1HLR
vector inputs_h2    Hs2HLR
vector inputs_h3    Hs3HLR
vector inputs_h4    Hs4HLR
vector output_done  PCDoneLSR
vector output_1     LdOut1 TrOut1 TgOut1
vector output_2     LdOut2 TrOut2 TgOut2
vector output_3     LdOut3 TrOut3 TgOut3
vector output_4     LdOut4 TrOut4 TgOut4
vector datapins     PCDat{23:10}HLR PCDat0{9:0}HLR

| Watch and Trace
w io7_0/enable
t output_1 Hs1HLR

| Set the clock
setclock -R 0 -W 0 | Disable reports and module execution
cclock 2 0 | Do two cycles with Reset enabled
l ResetHSR TestHSR
cclock 2
setclock -R 3 -W 2 | Enable reports and module execution
setclock -C ResetHSR -P Phi | Define clock and reset nodes
setclock -I inputs_control -I inputs_addr
setclock -I inputs_h1 -I inputs_h2 -I inputs_h3 -I inputs_h4
setclock -N this_is_a_test | specify test name
setclock -O output_done
setclock -O output_1 -O output_2 -O output_3 -O output_4
setclock -T datapins io7_0/enable
setclock -V vectors -Z
cclock 640 200 | run simulation for 640 cycles or for
                | 200 cycles after all modules finished,
                | whichever happens first.

exit

```