DOM Hub Communications Testing Procedure

Document version 1.4

Use with `dor-driver` release V02-05-02 or higher.

December 20, 2004

John Jacobsen
NPX Designs, Inc.
jacobsen@npxdesigns.com

NPX DESIGNS, INC.
1420 W. Edgewater Ave.
Chicago, IL 60660
773 769 0522 (o)
775 254 5992 (f)
www.npxdesigns.com
info@npxdesigns.com

Table of Contents

Objective

The goal of the procedure described below is to have a self-contained set of tests which provide sufficient confidence that the interface between the DOMs and the DOM Hub works as required. By "interface" here we mean the DOR card driver and firmware ("dor-driver") and the DOM firmware and software ("DOM-MB").

Test Framework

Several test and utility programs (written in C or Perl) are provided as part of the dor-driver package. While many of these perform simple actions, such as changing the state of a DOM or performing an echo test, some of these provide aggregate test functions, with higher level programs calling simpler ones. The highest level test program is called **moat** ("mother of all tests").

Moat provides a framework for repeatedly cycling through a set of tests and storing the results. Results are stored in a time-stamped directory, with a subdirectory for each test. A log file for the test session is also created. The global test pass/fail is recorded in the log file, and is also indicated in an empty file named "SUCCESS" or "FAIL."

The advantages of such a framework are:

1) Test suite can be run unattended;
2) Test can be run multiple times with results retrievable according to when the test was started;
3) No tools other than the Unix command line interface are required to examine the result;
4) New tests can be added to the framework relatively easily.

Moat and its supporting tests are installed in /usr/local/bin when the DOR card Linux driver is installed (either manually or by following the DOM Hub Software Installation Procedure; see References, below).

Individual Tests

Moat currently supports three tests:

1. **configboot stagedtests** (default duration: 10 min): put DOMs into echo mode ("echo-mode-cb") using the firmware used for configboot. Verify communications on all attached DOMs;

2. **"release" stagedtests** (default duration: 8 hrs): put DOMs into echo mode using the latest firmware release. Verify communications and time calibration functionality on all attached DOMs;

3. **"jumping bean tests"** (**MJB)** (6 hrs): run a battery of short tests on all attached DOMs in random order. The main idea is to look for edge effects and interactions between differenct kinds of operations in the DOR firmware and driver. The short tests are currently described elsewhere [3].

4. **OPTIONAL: "release" stagedtests, saving time calibration data**: using the -t switch to moat, one can opt to save time calibration data for a certain time interval; details are provided below.

The tests are implemented in moat by executing the "stagedtests" and "run-mjb" programs.

Stagedtests

Stagedtests is a Perl script which puts the DOMs in the proper state and then does continuous communications and repeated time calibrations using the "readwrite" and "tcaltest" programs:

readwrite: echo test program for a single DOM. DOM must be in a state where it will echo back all data packets that are sent by the DOM Hub. Readwrite creates random messages with variable length (1 to 4092 bytes) and writes them into the DOR driver as quickly as possible. It then collects the replies from the DOM and verifies that no messages have been lost or corrupted.

tcaltest: time calibration test program for a single DOM. Performs time calibration operation once per second. Verifies that time calibration operation completes. Verifies that the time stamps are in the correct order. Verifies that the DOR time delay ($t_3 - t_0$) exceeds the DOM time delay ($t_2 - t_1$) (when corrected for their relative clock frequencies). Performs simple waveform test to verify that a minimum threshold is passed.

While it may seem a bit complicated at first, the modular, heirarchical organization of the test programs is very helpful for diagnosing problems with a release. When one test fails, the more basic programs can be run by hand in order to determine exactly where the problem might lie.

As new DOM firmware versions become available (e.g., "real FPGA" as opposed to the current release of the "test" FPGA), each will have its own test assigned to it and added to moat.

Moat Options

Current options to moat (dor-driver release V02-02-01 or higher) are:

Option	***Argument***	***Default***	***Purpose***
-n	number of runs	1	Number of times to run each test
-d, -dorfreq	Clock speed (MHz)	(prompt)	Specify DOR clock frequency (required for time calibration test)
-c, -cbsecs	duration (seconds)	600	Number of seconds for each run of configboot stagedtests
-r, -relsecs	duration (seconds)	28800 (8 hrs)	Number of seconds for each run of "release" stagedtests

Option	*Argument*	*Default*	*Purpose*
-t, -savetsecs	duration (seconds)	0	Number of seconds for each run of "release" stagedtests where tcalib data is saved
-s, -skipmjb	-	-	Skip MJB phase of tests
-h, -help	-	-	Show help message
-g, -testgps	-	-	Test GPS/DOR time sync. functionality
-k, -kill	-	-	Kill existing test jobs and exit (may have to wait for currently running MJB test to finish)

Running Moat

First, determine your DOM Hub clock frequency. The standard configuration is 10 MHz (DSB installed) but if you are running a stand-alone DOM Hub with no DSB for testing purposes it may be 20 MHz. If you don't know, ask whoever installed the DOM Hub.

Log in as a normal user and verify that nobody else is using the DOMs attached to the DOM Hub. Verify that it is safe to apply power to the DOMs (moat applies power automatically when started!). Make sure the freezer temperature is set correctly.

Then, start the tests by running moat, e.g.:

```
% /usr/local/bin/moat -d 10 -n 8
```

The above command runs all tests on a 10 MHz DOM Hub, eight times or until one of the tests fails.

Moat cycles through its tests and stores the result in a directory (relative to where you started moat) beginning with `MOAT_....` The file MOAT.out in that directory contains a log of the session. Inspection of this log will indicate whether the suite of tests passed successfully or not. You can also inspect supporting files for each test, stored in subdirectories named

```
test???/<testname>.
```

(where `???` is `000`, `001`, `002`, ...). The files in these directories can provide helpful information in case of a test failure.

For convenience, the directory `latest_moat` is a symbolic link to the most recent output directory (`MOAT_....`).

Moat powers off all DOMs when it is finished. A file called `SUCCESS` or `FAIL` is created in the top level testing directory (the same directory where `MOAT.out` lives), depending on the result of the test.

Only one instance of moat can be run at a time on a given DOM Hub.

Saving Time Calibration Data (Cable Test)

For the purposes of evaluating string performance where a full complement of DOMs are attached to a real cable, and for other situations, it may be helpful to save time calibration data collected during the test. Moat provides the "-t" option which allows one to save all the time calibration data collected during a stagedtests run. For example,

```
% moat -d 10 -c 600 -r 2400 -t 600 -n 2
```

runs moat on a DSB (10 MHz) DOM Hub for 2 runs of 7 hours each (1 hr stagedtests + 6 hours mjb). The time calibration data (1/sec per channel per run = $N_{\mathrm{DOMs}} \times 1200$ records in the above example) will appear in the files named

```
MOAT__YYYY-MM-DD__HH:MM:SS/test???/save_tcal_stagedtests/tcal_data_c?w?d?.out
```

Recommended Test Parameters

The current requirement for new driver and firmware releases is that moat run on 16 or more DOMs for 14.2 hours (8 hrs, 10 min. stagedtests + 6 hrs MJB) at cold temperatures (-50 C) with no errors[4]. This is one test cycle with the default moat settings.

For a cable test where it is desired to save time calibration data, the recommended test is as follows:

```
moat -d 10 -c 600 -t 3600 -r 25200 -g
```

(10 minutes of configboot stagedtests, 7 hrs of release stagedtests, 1 hr of release stagedtests saving tcalib information, 6 hrs MJB; 10 MHz DSB card installed).

Short Tests

It can be helpful to run shorter tests to be sure that your DOM Hub configaration is "basically working." A good short test to try is

```
moat -d 10 -r 60 -c 0 -s -g
```

The test skips the MJB and configboot test phases. It should take about 2 minutes to finish. As always, check for `SUCCESS` or `FAIL` (and the other result files) in the `latest_moat` directory.

Stopping Moat

Normally, moat stops automatically when all runs have completed or an error condition has been detected. The files `SUCCESS` or `FAIL` in the directory `MOAT__YYYY-MM-DD__HH:MM:SS` indicate that moat has finished. If for some reason you need to stop moat before it completes, you can issue the following commands (from the same account you ran moat from):

```
% killall moat stagedtests.pl readwrite tcaltest run-mjb
% off all
```

It's best to wait several minutes to allow for any test processes spawned by run-mjb to finish (one can use the "ps ax" command to look for these; unfortunately there is currently no other way to guarantee that no other "mjb" processes are still attached to DOMs). Moat can then be run again; a new results directory will be created for the new run.

References

1. *J. Jacobsen*, The DOM Hub Device Driver – Function and Design. http://docushare.icecube.wisc.edu/docushare/dsweb/Get/Document-3743/domhub_driver_spec_2_18.pdf

2. *J. Jacobsen and M. Krasberg*, DOM Hub Installation Procedure. http://docushare.icecube.wisc.edu/docushare/dsweb/View/Collection-979

3. *A. Jones*, Communications Tests, file `dom-ws/testing/comm-tests` in the IceCube CVS repository.

4. *C. McParland*, private communication.