



onTAP BOUNDARY SCAN SOFTWARE
APPLICATION NOTES FOR
appXam

FLYNN SYSTEMS CORP.

PROVIDING BOUNDARY SCAN SOLUTIONS SINCE 2000

Introduction

appXam creates boundary scan tests and diagnostics based on qualified application mode signals such as those found in a user’s functional tests.

appXam uses application mode signal activity to quickly extend boundary scan fault coverage and diagnostics. The signal activity on a target board may simply be idle mode activity available from system clocks, or it may be purposeful activity created by a system program, for example to read DDR or flash memory. appXam provides the tools to capture this activity and then automatically create test models and add them along with diagnostic messages to a test suite.

Capturing application signals may also be qualified by prescribing a mix of specific signal values that must be true in order to create valid tests. For example, if the interest is in reading data from a DDR memory, then qualifiers can be set to validate data bus capture when a memory device’s chip select pins are active, but write enable is at an inactive level.

Application mode signals run at normal system speeds whereas appXam is capturing this data at boundary scan speeds. As a result sufficient application activity has to be present in order for the sampling to be effective. To assist along these lines qualifier conditions, in the form of prescribed logic levels at designated pins, may be set within appXam to capture data only when it is meaningful.

appXam is easily accessed from the ProScan screen, as shown in *Figure 1*.

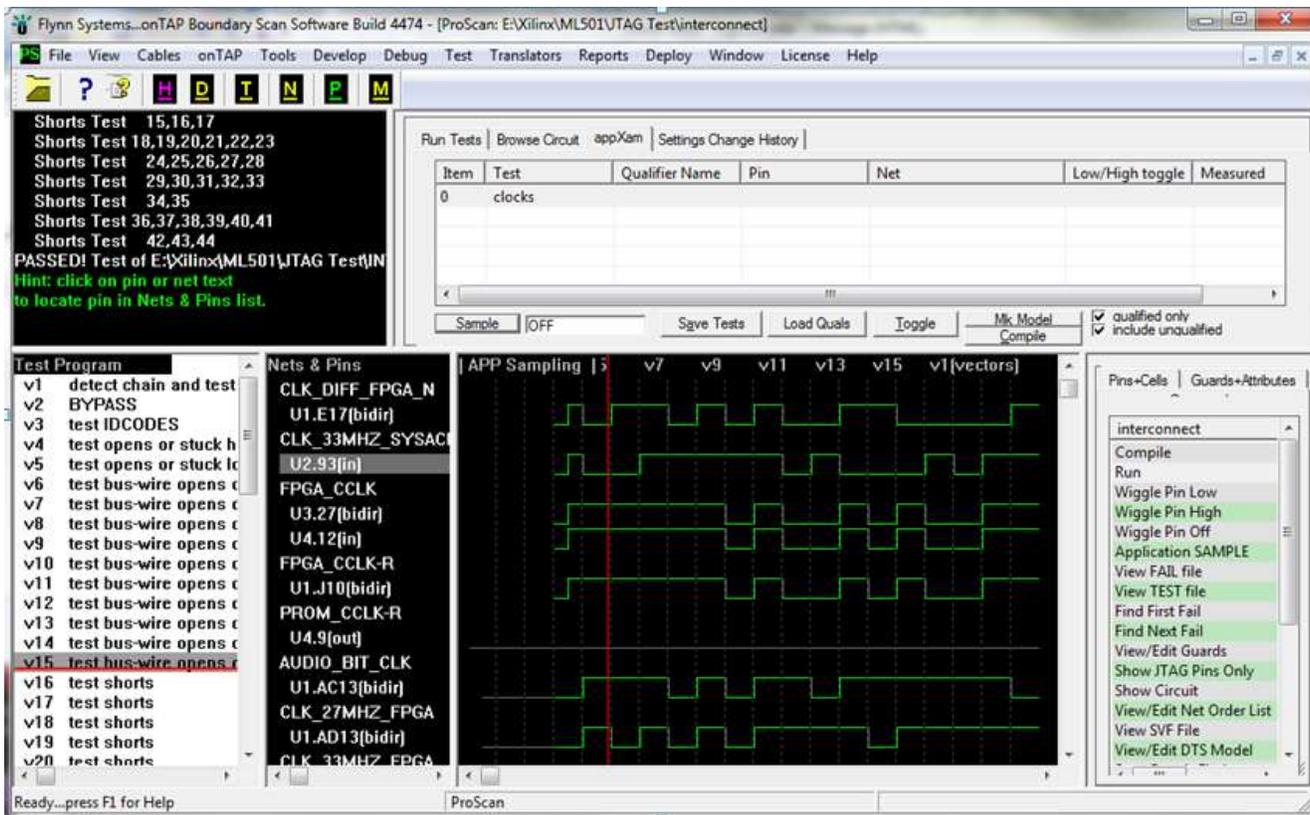


Figure 1: ProScan screen with appXam selected.

Quick steps to create an appXam test

1. Create an interconnect test. It does not need to be fully debugged.
2. With the Development screen closed, open ProScan and then load and run the interconnect test.
3. Select the appXam notebook tab near the top of the ProScan screen.
4. Enter a name, for example **clocks**, in the Test column.
5. Click on the Sample button and let the sampling run for a few seconds to capture application mode signals and then toggle Sample off.
6. Click on save Tests.
7. Click the Mk Model and the Compile buttons to create a cluster model of the sampling and an executable SVF file. appXam will automatically run through the steps on the Development screen to create a cluster test for sampling and then return to ProScan.
8. Go to the Test screen and run the *clocks* test, which verifies the presence of idle mode system signal activity.



Figure 2: A step by step process to create an appXam test in the ProScan environment.

Quick Steps to create an appXam test (cont'd)

After Sample is toggled off, signal activity that is present may be viewed by vertically scrolling through the Nets & Pins plus Vectors display. To facilitate relating signals to one another, right click on a pin in the Nets & Pins list and select *Move Net + Pins to top of List* (Figure 3).

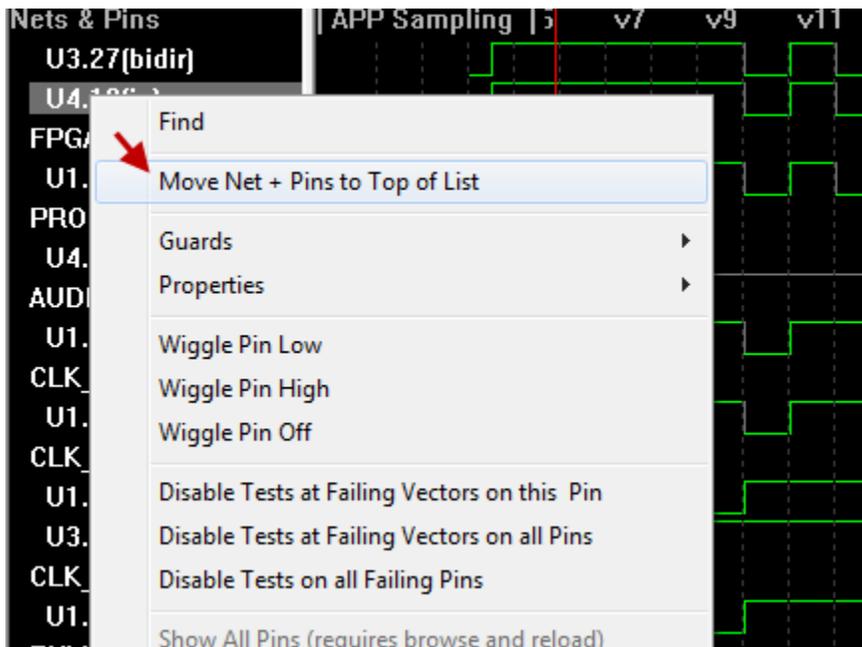


Figure 3: Move nets name to top of Nets & Pins list.

In addition a vertical red marker can help show concurrent levels of related signals.

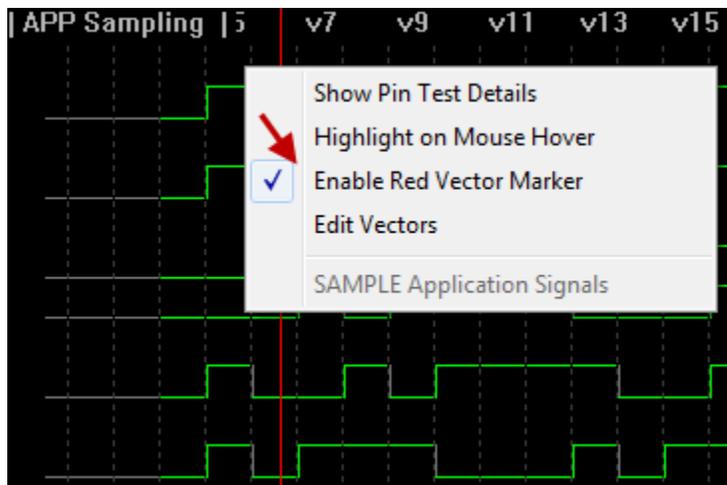


Figure 4: Enable red marker as visual aid. Click on top row showing V numbers to move marker.

Qualifying appXam data

Figure 5 shows an example of how qualifying conditions can be included in a sampling test. In this case we will capture the number of instances when the AUDIO_BIT_CLK signal is low and when the FPGA_CLK signal is low.

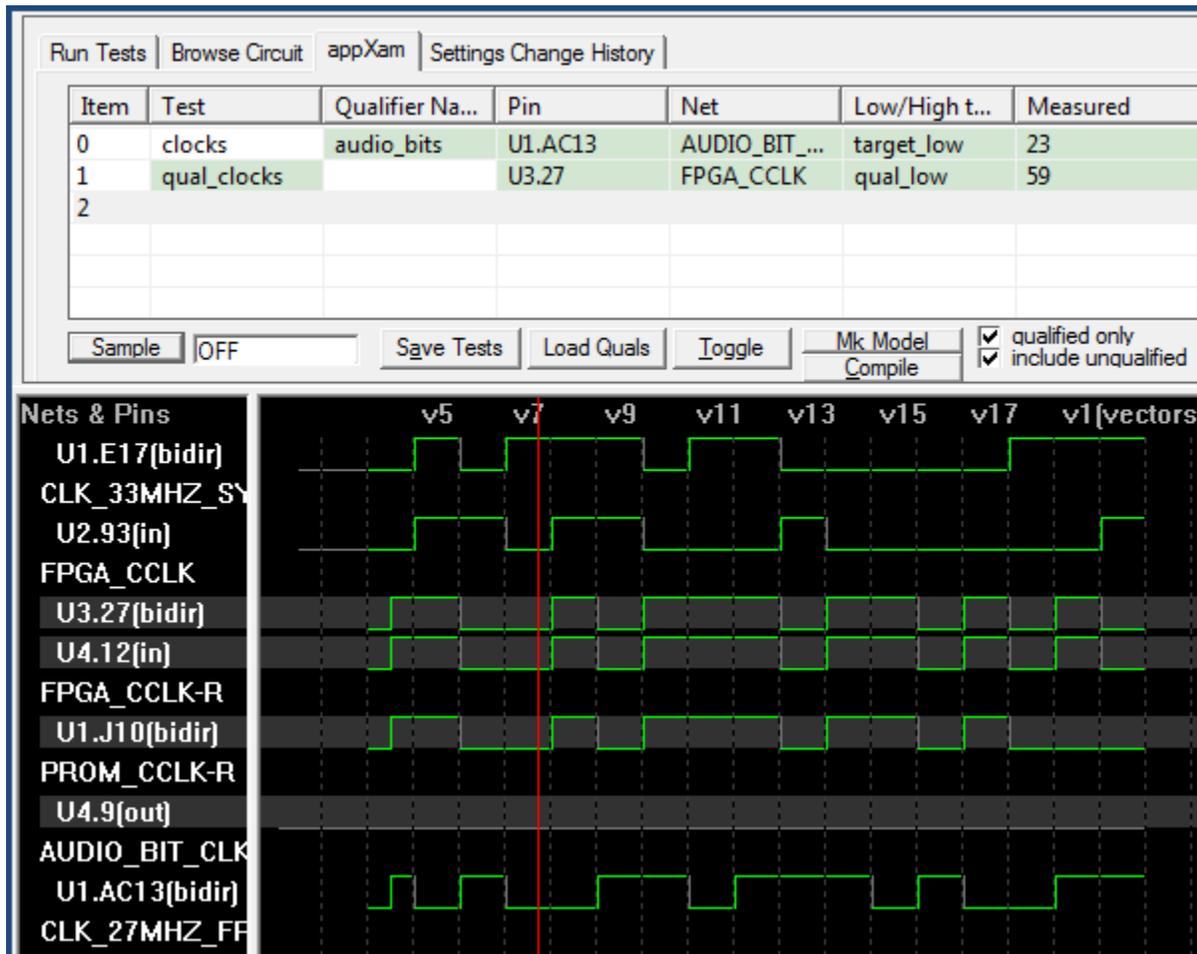


Figure 5: Adding qualifying conditions to appXam sampling tests.

Qualifying appXam data (cont'd)

After some preliminary procedure to examine sampling data and bring the related nets near each other in the Nets & Pins list as described above, the procedure to setup this test is as follow:

1. Enter a new test name **qual_clocks** in the *Test* column.
2. Enter a name for the qualifier, in this case **audio_bits**, in the *Qualifier Name* column.
3. To enter the pins related to the net signals that we wish to qualify, click on the first empty element in the *Pin* column and then click on the respective pin in the *Nets & Pins* list.
4. To set target value and qualifier levels, click in the row element to the right of the *Net* column and use the *Toggle* button to set a target or qualifier condition. Multiple qualifiers are permitted.
5. When finished entering targets and qualifiers, select *Save Tests*.
6. Click *Mk Model* and then *Compile* to create a DTS cluster test model and executable SVF file representing the sampling data. **Note:** the default conditions *qualified_only* and *include_unqualified* are selected. These settings can be changed and control whether or not the compiled test includes all sampling activity or qualified sampling tests only.

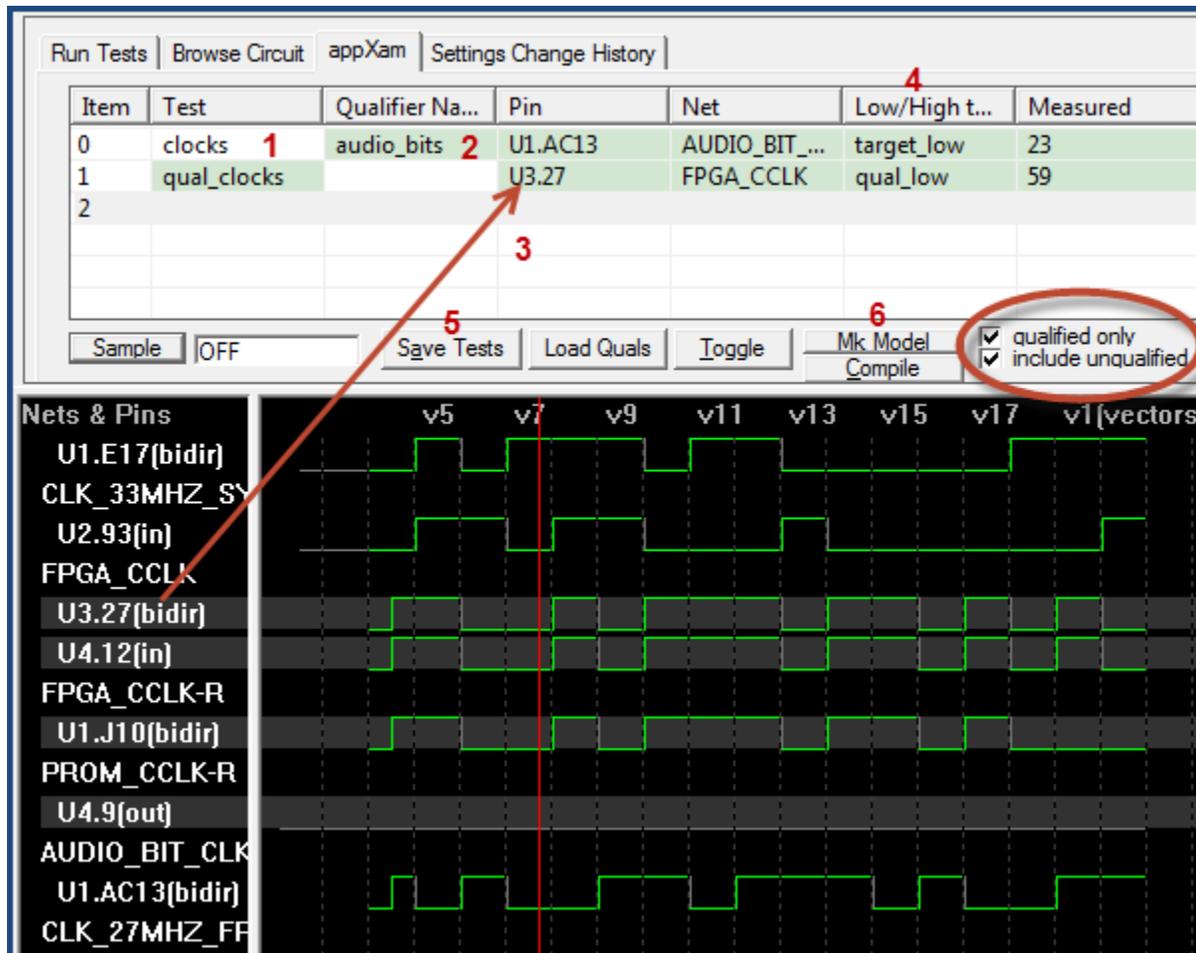


Figure 6: Procedure to set up qualifying conditions in a sampling test.