

Automatic Test Generation for PC Boards

Pre-requisites

It is assumed that the reader is familiar with the basic operation of our μ Master product, and has a reasonable understanding of processor board architecture.

Introduction

One of the unique features of the μ Master test and debug solution is its built-in Automatic Test Generator (ATG). This dramatically reduces the time required to develop a test program for a new board. Designed with manufacturing and field service repairs in mind, the ATG differentiates μ Master from other emulators used in board development, which normally require high levels of technical knowledge and skill to operate. μ Master's ATG makes CPU Emulation a viable and efficient tool for high volume board test and debug.

When generating test programs for PC-architecture boards, μ Master's ATG automatically produces a script within seconds. The generated script provides a comprehensive set of basic tests for the target board. **This Application Note deals with script generation for PC-architecture boards.**

The ATG can also be used to generate test programs for non-PC-architecture boards. However, the vast range of possible board architectures makes full automation of the test generation virtually impossible. Nevertheless, the ATG can still save much time in test generation. **Application Note #7 deals with script generation for non-PC-architecture boards.**

How does μ Master's ATG work?

When a board boots to its operating system, instructions for setting up devices on the board are run either from ROM memory (the BIOS in PC-architecture boards), or from hard disk

(OS-level device drivers). To be compatible with the wide variety of devices that could be present on a board, universal boot code includes a library of set-up instructions for most devices from different manufacturers.

However, μ Master normally tests a board **without booting it**. This saves time and also allows non-booting boards to be diagnosed. Since neither the ROM-based boot code nor the OS-level device drivers are being run, it is necessary for μ Master tests to contain similar device set-up code.

A μ Master test program or script for a specific board is broken down into individual sub-tests that set up and exercise each addressable device on the board. For example, there are individual tests for memory, USB, LAN, etc.

The ATG is supplied with a library of pre-written sub-tests for all devices commonly used on PC-architecture boards. This library also includes a large number of devices used on non-PC-architecture boards.

The ATG must always be run on a booted, fault-free board. Where possible, the ATG used PCI IDs to identify the types of devices that are on the board and then assembles an appropriate set of tests from the built-in device library (Fig. 1).



Fig. 1 – The ATG generates a test script for a known-good board using the device library.

This automatically generated set of tests will diagnose faults in the majority of the board. However, it will probably require some optimisation to maximise coverage. This is explained later in the document.

Running the ATG

The ATG is run in the following sequence:

1. Connect μ Master to a fault-free board that is configured with the same type of hardware (CPU, RAM, etc.) that will be used on the boards to be tested. The operating system installed on the board ideally should be Microsoft® Windows® XP Professional, and drivers should be present for all devices on the board. A different Windows version could be used but it is essential that all device drivers for the board’s components are available.
2. Connect a monitor to the board.
3. Power up the board and boot it to Windows. This can be done from the μ Master Toolbar (fig. 2).

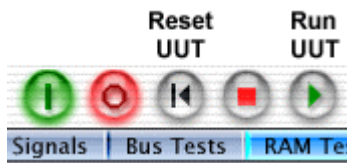


Fig. 2 – μ Master Toolbar buttons: Run UUT boots the board, after Power ON.

4. Use the Windows Device Manager to check that all device drivers are correctly installed (Right-click the “My Computer” icon on the desktop and select “Manage”, or select “Administrative Tools” from the “Control Panel”, then select “Computer Management” in which you will find the “Device Manager”. If any drivers have not been installed correctly, these will be indicated.
5. Click the “Developer” Tab on the μ Master Toolbar and then click the “Automatic Test Generation” button. If only the “Operator” tab is visible, the μ Master user interface options need to be changed by the system manager. If only the Developer tab is visible, but there is no Automatic Test

Generation button, μ Master has been started up in “Edit Only” mode.

6. In the Automatic Test Generation window (fig. 3), select the highest PCI Bus number that the ATG should scan for devices. If you don’t know how many buses are implemented on the board, this number can be set higher than the default of 2. However, additional time will be expended even though the buses may be absent.

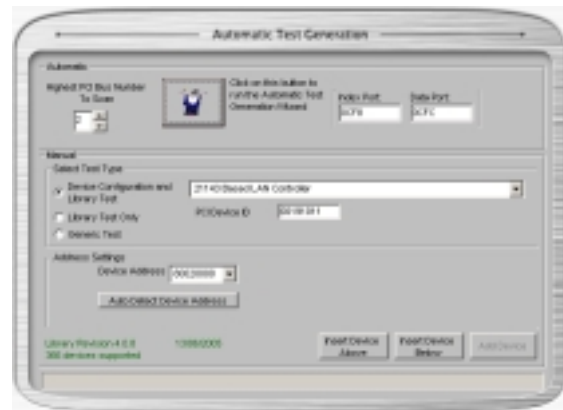


Fig. 3 – μ Master’s Automatic Test Generation window.

7. The PCI Index and Data port addresses are set to the defaults of 0CF8 and 0CFC. These are normally correct for PC-architecture boards.
8. Click the Run ATG Wizard button to start the ATG (it is vital that the board has been fully booted to Windows and that all drivers have installed correctly as in point 4).
9. Depending on the board type, questions about the configuration may be asked during the ATG sequence. For example, if the chipset is an Intel® BX series, you may be asked whether Normal DIMMs or SO-DIMMs are installed.
10. After some seconds, the ATG will have created a test script, and this will be displayed in the Test Script window (fig. 4).

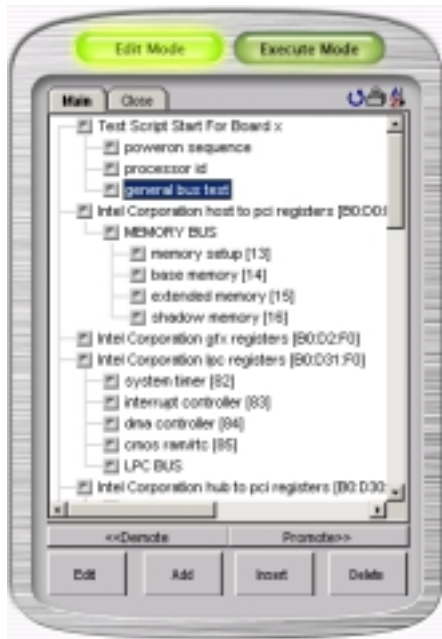


Fig. 4 – μMaster’s Test Script window.

Optimisation of the generated test script

When a test script has been generated by the ATG, it will be necessary to make some modifications to optimise coverage and to avoid non-existent components being tested, causing incorrect diagnostics. The following modifications may be required:

1. Addition of devices that do not identify themselves to the ATG, e.g. Super I/O device. If set-up code for these devices is available in the library, it can be manually added to the test script as follows:
 - a. Select a test in the Test Script window (fig. 4), above or below which the new test should be added.
 - b. In the “Manual” section of the ATG window (fig. 3), select the option “Device Configuration and Library Test”.
 - c. Select the device to be added from the drop-down list.
 - d. Click either the “Insert Device Above” or “Insert Device Below” button to add the test to the Test Script window in the desired location.

- e. Having added the test for the device, it may be necessary to delete sub-tests if some of the device’s functions are not implemented. For example, figure 5 shows that the newly added Super I/O test includes sub-tests for a parallel port, 2 serial ports, an IrDA SIR port and a floppy disk. If any of these are not present, the sub-tests must be removed (see point 2 below).

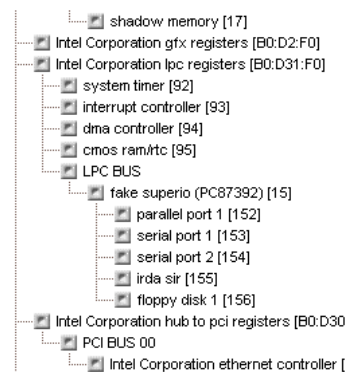


Fig. 5 – Delete unwanted sub-tests after adding a new test.

2. Removal of tests for devices that are not present on the board: the ATG assumes that the board is fully populated. For example, if one of the available RAM sockets will be empty when the board is tested, this test will need to be removed, otherwise the test will indicate that the RAM has failed. If the USB controller is capable of supporting 6 ports, but only 4 connectors have been implemented on the board, the 2 surplus tests need to be removed (fig. 6). Some of the IDE Primary and Secondary Master / Slave Tests are removed as follows:
 - a. Click the “Edit Mode” button at the top of the Test Script window, if this is not already selected (fig. 5).
 - b. Select the test(s) to be removed in the Test Script window (fig. 6).

- c. Click the “Delete” button at the bottom of the Test Script window.

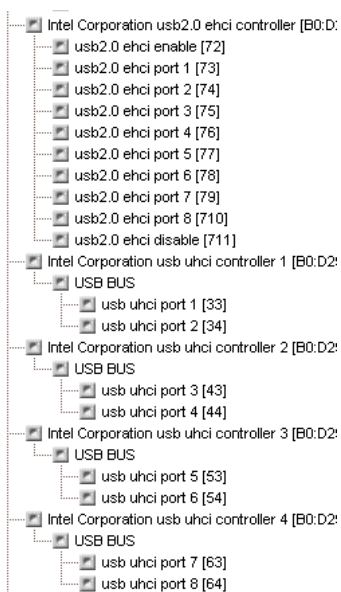


Fig. 6 – Possibly surplus USB tests in the Test Script window.

3. If the ATG device library does not contain set-up code for a specific device found on the board, a generic register test is added for that device. This test will confirm that the device is accessible but it will not verify its full functions. The test developer will need to expand the test if all functions must be verified. A description of how to do this is beyond the scope of this document. However, it will normally be necessary to have a device register mapping datasheet and a BIOS writer’s programming guide for the device in question, in order to be able to write a complete test script.
4. AC97 Codec IDs cannot be detected automatically by the ATG. However, the generated Codec Registers test script includes code to read the IDs and compare them with “dummy” values (fig. 7). This code is commented out (apostrophe at start of line), but if the comments are removed to allow the code to run, when the test is executed, it will probably* fail and

give an error message that contains the correct IDs. These can then be copied and pasted into the script to replace the dummy IDs.

*In Figure 7, the dummy IDs to be replaced are 0090, 4144 and 5372. These are in fact the real IDs for the Analog Devices 1981 device, so this test would pass if your board carries this device.

When the correct IDs have been pasted into the test script, the test will pass. Note: “Halt On Error” should be disabled in the µMaster “Options” menu under the “TSL/1 Settings” tab, otherwise the test-copy-paste process will need to be repeated 3 times to get the 3 real AC97 Codec IDs from the subsequent error messages.

```
'user can add codec device id check here if required
'remove comment below, and change xxxx to correct codec device id
GPrint - Check Codec Device ID's
'ReadIOWord 38B4
'LoopIOWord 38B4 0001 0001 2 #ERR_AcLinkReady
'ReadIOWord 3000 FFFF 0090 #ERR_AcLinkCodecID
'ReadIOWord 38B4
'LoopIOWord 38B4 0001 0001 2 #ERR_AcLinkReady
'ReadIOWord 307C FFFF 4144 #ERR_AcLinkCodecID
'ReadIOWord 38B4
'LoopIOWord 38B4 0001 0001 2 #ERR_AcLinkReady
'ReadIOWord 307E FFFF 5372 #ERR_AcLinkCodecID
```

Fig. 7 – Remove comments to obtain real AC97Codec IDs in the test error message.

5. The General Bus Test address range may need to be modified when the ROM area is not shadowed.

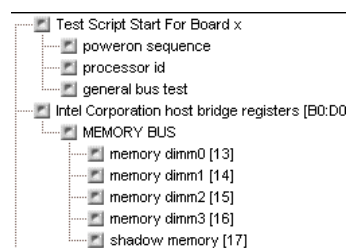


Fig. 6 – The General Bus Test address range may need to be altered.

6. If battery tests are required (e.g. for mobile devices), these need to be added manually from the ATG menu. This will provide a template test script that must be modified to match the

command codes used by the manufacturer of the board's embedded controller BIOS (ECBIOS). The required modifications are beyond the scope of this document. A good knowledge of ECBIOS routines will be needed to make these modifications. Customers may prefer our applications engineers to carry out this programming.

7. LCD tests are board specific. These need to be added manually via the ATG menu. An external file containing the board-specific pin mapping and levels is read by the ATG during the generation of the test script. This external file is either created by the test developer or by our application engineers. The way in which these pin mapping files are written is covered within our optional training courses.

Contacts for additional information

European Sales and Support:

International Test Technologies,
Larkin House, Oldtown Road,
Letterkenny, County Donegal, Ireland.
Tel: +353 (0)749 188 100
Fax: +353 (0)749 188 128
E-mail: sales@intertesttech.com
Web: www.intertesttech.com

N. American Sales and Support:

International Test Technologies,
2694 21st. Avenue,
San Francisco, CA 94116
Tel: 415 753 5376
Fax: 415 753 3635
E-mail: sales_usa@intertesttech.com

Asian Sales and Support:

International Test Technologies,
32 Maxwell Road,
#03-07 White House,
Singapore 069115.
Tel: +65 9642 3164
E-mail: sales_asia@intertesttech.com

For contact details of our worldwide reps.
please see our website:

http://www.intertesttech.com/ate/company_locations.htm