In-System Programming of Atmel ATF1500AS Devices on the HP3070

Introduction

In-System Programming (ISP) support of Programmable Logic Devices (PLD) is becoming a requirement for customers using Automated Test Equipment (ATE) for board-level programming, testing and verification in a production environment. The advantages of ISP-compatible PLDs allow these customers to program the parts directly on their system board before they are tested. This saves customers money by simplifying their production test flow and allowing them to release their products to market sooner. To address this need, Atmel is offering programming support for the ATF1500AS family of devices in the ATE environment. This application note describes programming support for the Hewlett-Packard HP3070 Series of ATE testers.

Device Support

There are no device restrictions within the Atmel ATF1500AS family for programming support on HP3070 testers. All devices in the ATF1500AS family have a fixed algorithm, which is automatically compatible with HP testers. The Atmel programming algorithm offers the added benefit of a fast and consistent programming time for all devices within the family. Table 1 below shows the Atmel devices supported by the HP3070 tester. Support for new devices is planned. Contact Atmel EPLD Applications for further information.

Contact information for PLD Applications:

e-mail: pld@atmel.com
hotline: (408) 436-4333
url: www.atmel.com

Table 1. ATF1500AS Device Support on HP3070

ATF1502AS/ASL/ASV/ASVL
ATF1504AS/ASL/ASV/ASVL
ATF1508AS/ASL/ASV/ASVL

ATF1500AS Device Family

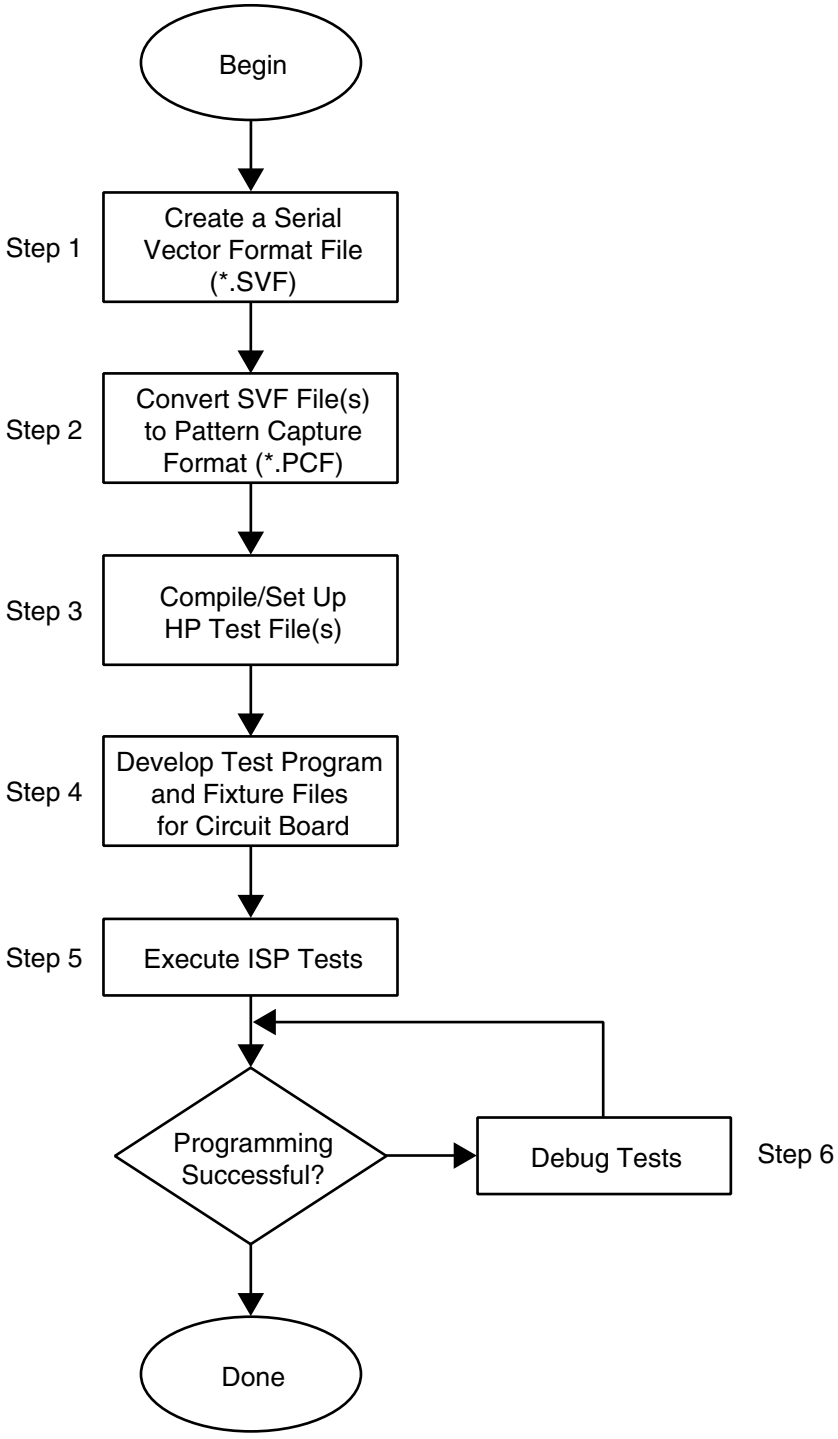
Application Note

Rev. 1651A–04/00

HP3070 Programming Flow

The programming flow for Atmel ATF1500AS devices on the HP3070 tester is described below in Figure 1. Further information for each step is detailed in later sections of this application note.

Figure 1. ATF1500AS Devices Programming Flow



Creating the HP Test Fixture

The HP3070 Tester is a “bed-of-nails” tester. This means that the tester’s pin array (i.e. “bed-of-nails”) makes direct contact with the circuit board traces at precise access points. Signal probing is done through a test fixture. The circuit board mounts into this fixture and probe pins make contact with the board via holes drilled at precise grid points on the fixture. The tester is a noisy environment where many high-speed signals can be in close proximity to each other. Hence, crosstalk is a problem. To eliminate crosstalk and other tester problems, certain guidelines, listed below, are recommended for the test fixture. This will help reduce hardware debugging time after tests are completed.

- **Use a bi-level test fixture**
  This fixture allows for two types of probe pins: short and long pins. It is recommended that long pins be used for critical length such as the VCC, GND and all the JTAG signals. Short pins should be used for all other signals to test. Minimizing signal lengths for the VCC, GND and JTAG signals is important to prevent ISP programming and test problems. If you plan to use a bi-level fixture, you will need a fixture “gate”. This can be purchased from HP. The gate allows the probe pins to be applied sequentially to the circuit board – long pins first, then short pins. A regular test fixture supports only short pins. If you must use a regular test fixture, make sure to minimize the lengths of the VCC, GND and JTAG signals to reduce any inductive noise that may occur.
- **Pull down TMS and TCK pins through external resistors**
  Both TMS and TCK inputs must be pulled to GND through an external resistor on the fixture. These signals should be held low between PCF downloads on the tester. Otherwise, the tester may tri-state these pins and this can disrupt the JTAG programming operations of the devices on the circuit board. Atmel recommends that you use resistor values equal to 200 ohms maximum.
- **Disable oscillators**
  Oscillators should be disabled during the JTAG programming process. Oscillators can create additional system noise that may affect JTAG programming. It is recommended that your circuit board design include logic to disable oscillators when necessary.
- **Signal edge rate**
  Atmel recommends that the JTAG input (TMS, TDI and TCK) signals should have an ideal edge rate (both rise and fall time) of 0.1V/ns (0.1V per nanosecond) when applied to the board. The V_{CC} ramp rate should ideally be 1V/us (1V per microsecond) and monotonic. Slower edge rates on either the JTAG inputs or the V_{CC} supply are likely to create noise that may affect either programming or functional testing of the circuit board.
- **Circuit board layout of JTAG signals**
  It is important to optimize your circuit board layout so that all your JTAG inputs, especially the TCK and TDI signals, are placed away from any other high-frequency signals on your circuit board. This may interfere with JTAG programming operations in your JTAG device chain.
- **Ground plane on test fixture**
  It is recommended that the test fixture have a ground plane to reduce ground bounce effects that might create noise on JTAG signals during programming operations.
- **Bus contention on bi-directional I/Os**
  To prevent bus contention on bi-directional output pins on your circuit board that interface with Atmel devices, it is recommended that you modify your PLD logic to include an externally-controlled disable feature so that outputs may be tri-stated during the testing process.

Creating the SVF Files

Serial Vector Format (SVF) files for the HP tester are created by using the Atmel-ISP software. This software has an option to create either Rev. C or Rev. D SVF files. *The PCF translator requires that you generate Rev. D SVF files.* These are preferred over Rev. C because they implement programming delays as fixed values instead of TCK's delays. TCK delays are proportional to the tester's TCK cycle time. Following is a brief outline of the procedure to create SVF files. Additionally, some programming guidelines are included.

For more information, download the "Creating JAM™/JBC™ Files for Atmel Devices" application note from the Atmel web site or browse the Help file included with the software. To view the Help file, select **Help.. Contents..** from the ISP software main menu. Then select the **Getting Started** topic.

Procedure to generate SVF files:

1. Create the JTAG device chain using the Atmel-ISP software. This involves specifying the device type, the ISP operation and the JEDEC programming file (when necessary) for each Atmel device in your JTAG hardware chain. Remember that non-Atmel ISP devices must be placed in the bypass mode. Make sure the software's chain configuration in software matches the JTAG hardware chain on the circuit board.
2. When you are done, go to **Process.. Options..**
3. At the notice "Write SVF file instead of LPT port", select **Yes**.
4. At the notice "Does the Target System Support SVF Rev. D?", select **Yes**.
5. The Create SVF File dialog box will open. Enter a file name and select **OK**.
6. At the notice "Use STATE-RESET's in SVF File", select **No** if you are programming a multi-vendor JTAG device chain (both Atmel and non-Atmel ISP devices). Otherwise, select **Yes**.
7. Go to **Process.. Run..** to execute the ISP operations.
8. Exit the Atmel-ISP software by selecting **File.. Exit..** from the main menu. The SVF file will be written only after you exit the software.
9. If you attempt to create a new SVF file while the software is still running, the notice "Close Existing SVF file Create Another?" will appear. Select **No**.

Notes:
1. It is *not* recommended that you execute more than one type of ISP operation in an SVF file. Create a new SVF file for each different ISP operation you want to perform on the Atmel device(s) in the JTAG chain. Make sure that all non-Atmel devices in the chain are in the bypass mode.
2. It is recommended that you create at least one SVF file that executes a Program/Verify operation on the Atmel device(s) in your chain. This SVF file automatically includes verify vectors and performs a bulk-erase of the device. If the Atmel devices program and verify successfully, then you can streamline your test flow to include just a verify operation during further production tests.
3. If you intend to reprogram the Atmel devices through JTAG ISP, make sure that the JEDEC file does not use the JTAG ports pin for logic I/O. The Atmel-ISP software will warn you of this. Otherwise, the devices can be programmed only once on the tester. A third-party programmer will be required to erase the devices for reprogramming.

Creating the HP Pattern Capture Format (PCF) Files

The Atmel SVF2PCF utility converts SVF files generated by the Atmel ISP software to PCF files compatible for the HP3070 tester. To download the utility, go the Atmel web site and select **Products.. Programmable Logic Devices.. Software** and download the self-extracting *svf2pcf10.exe* file.

The SVF2PCF translator supports Rev. D SVF files only. PCF files can be quite large and are limited to the amount of memory available on the HP3070 tester – typically 7M bytes. The size of the PCF file created by the SVF2PCF translator depends on both the JEDEC file to be programmed and the type of ISP operations to be performed. The SVF2PCF translator will automatically partition large PCF files to suit the memory requirements of the tester.

The SVF2PCF utility is easy to use. Below is a procedure for using it and some guidelines:

1. In the DOS command prompt type:

   ```
   Svf2pcf <input_filename>(.svf) <output_filename>
   ```

   You must type an output file name. The translator will always create an output file with the *.v0x extension, where x = number. Depending on the order the PCF file is to be run on, the input file name(s) must be 8 characters maximum for the file name and 3 for the suffix.

2. If you want to include comments in the PCF output file, then use the –c option. Command line syntax is below:

   ```
   Svf2pcf -c <input_filename>(.svf)
   <output_filename>
   ```

The translator includes two files: a *head.pcf* and *tail.pcf* file. These are appended by the translator to the output PCF files. These files must exist in the same directory where the output file is written to. Otherwise, the translator will not run.

The default vector cycle time on the output PCF file is 170 ns, which corresponds to a 3 MHz TCK clock rate. If you need to run at a different rate, you will need to manually edit either the PCF file and re-run the test or use the HP Debug Mode to change the TCK period “on-the-fly”. If you plan to use the latter option, please refer to the “Debug Mode” topic in the “Debugging the Tests” section of this application note.

In step 1 above it was noted that the translator uses the *.v0x suffix to label the output file name it created. This is to assure that the output files are run in the correct order on the tester. For example, if the translator created two PCF files, then <output_file>.v01 and <output_file>.v02 are generated. <output_file>.v01 must be loaded and executed before <output_file>.v02 is run.

It is recommended that you create PCF files without comments for your production test. This will reduce the size of the PCF files generated and shorten the time to load and compile them on the HP3070. Comments are useful for debugging purposes if your tests fail to run correctly.

Setup and Execution of Programming Tests on the HP3070

Before JTAG programming can be performed on the HP3070, each PCF file must be compiled and the digital tests set up for the tester. This is a multi-step process, outlined below:

1. Copy the PCF files to the board library directory.
2. Compile the PCF files.
3. Run Board Consultant.
4. Run Test Consultant.
5. Load/Verify the Testplan.
6. Execute the tests.

Copy the PCF Files

The first step in the setup process is to copy all PCF files created by the translator to your *custom_lib* subdirectory. Specify in the name of the PCF file to indicate the type of test you want to perform and include all of them in your board library directory. For example:

```
U1.pgm    // PCF Programming File
U1.ID     // PCF ID-Check File
U1.vfy    // PCF Verify File
U1.byp    // PCF Bypass File
Etc
```

Compile the PCF Files

The next step requires you to compile your PCF source file to create the tested bitstream (*.o) object files for the test. Make sure the node names for the JTAG signals in your board file match the JTAG signal node names in the PCF files created by the translator. If these node names do not match, your files will not compile correctly on the tester.

Note: Instead of editing your board file, you can edit the JTAG signal node names in the *head.pcf* file to match the JTAG signal node names in the board file.

To compile your PCF files, perform the following steps:

1. Go to the *custom_lib* subdirectory on the command prompt. Type:

   ```
   compile "PCF filename" ; library. <return>
   ```

2. View **Error/Warnings** from compiler's outputs and correct accordingly.
3. Open the board file.
4. Review the **Connections** section.
5. Find the node names associated with the JTAG port pins on the ATF1500AS device.
6. Change the JTAG node names to "TCK", "TMS", "TDI" and "TDO" for each JTAG signal, respectively. If you are unsure what pin corresponds to the JTAG port on the ATF1500AS device, please download the device's datasheet from the Atmel web site.

Running Board Consultant

The Board Consultant program allows you to edit or modify the information in your board files. The *Board* and *Board_xy* files contain all the information the test needs to execute. For example, the pin/node names and numbers, the devices on the board to test, the boundary-scan chain, the testing coordinates for the short probes, etc.

7. Type:

   ```
   board consultant <return>
   ```

8. Select **View**, **Edit**, **Board Description**.
9. Select **Enter Node Library**.
10. Under "Designator" and "Part Numbers" type PCF file name. The library and part number names will be the PCF file names.
11. Select **Add/Replace Devices**.
12. Repeat steps 6 through 9 above for each PCF file.
13. When you are done, select **File**, **Save Board Information**.
14. Select **Final Compile and Verify**.

Running Test Consultant

Test Consultant schedules and generates the tests on the HP3070. The Testplan executes the binary tests on the HP3070. The Testplan applies power to the board and downloads the bitstream stimulus file (HP object files) to the tester to execute the test. The Testplan will also report any errors that occur during the testing process.

15. At the command prompt, type:

    ```
    test_consultant <return>
    ```

    If the PCF files were compiled correctly, the TestPlan should be generated automatically.

16. Load the Testplan and verify the test order sequence. Modify accordingly if the files are in the wrong order.
17. Run the Testplan. The tests will execute in the order specified.

Debugging the Tests

There are many reasons why tests could fail. Both hardware and software configuration problems can create problems on the tester. The following is a list of troubleshooting guidelines you can use to help debug your tests.

PCF File Ordering

If the TestPlan has the PCF files listed in the wrong order it will cause programming to fail because the devices are not receiving the correct programming data. In some cases the tester will partition the testing into several PCF files. These files must be executed in the exact order they were created. The translator organizes this for you by labeling the file suffix to indicate the order a file needs to be tested.

No Pull-down Resistors on Fixture

For multiple-partitioned files, a pull-down resistor is essential on the TCK and TMS pins to keep them in a stable state during test file loads. Make sure the resistor is equal to 200 ohms. Otherwise, you might receive an overpower error from the tester.

Debug Mode

Adjusting the vector cycle time in the HP Debug Mode to a faster rate may cause the tester to incorrectly model internal programming delays that are necessary to program the part. This can create programming problems. If you need to adjust the vector cycle time, change it in the PCF file, re-compile and test without using the debug mode. A warning also appears about this in the header of each PCF file generated by the translator.

Check Your JTAG Chain Configuration

Make sure your JTAG chain configuration matches the hardware configuration on circuit board under test. All devices in the software chain must match the circuit board both in order and device type. Any mismatch between the two will cause a programming error.

Using State-Reset's in SVF File

Do not use State-Reset's in the SVF file if you are attempting to program a multi-vendor JTAG chain that contains Atmel devices. Doing so may cause other non-Atmel parts to reset internally. This can affect the JTAG bitstream if these devices reset without a bypass register between TDI and TDO pins of the devices. ISP programming will be successful only if non-Atmel devices are placed in the bypass mode during all ISP operations.

Isolate Problems to a Single Device

Try to isolate the programming problems to a specific device in the JTAG chain. Create an SVF file that programs and verifies one Atmel device at a time while putting others in the bypass mode. Repeat for each Atmel device in the chain. This process allows verify errors to be debugged to a specific device in the chain.

Cross-verify on Another Programmer

Try programming the device chain using the Atmel-ISP software. This will verify whether the devices can be programmed and will help isolate whether the problem is device- or circuit board-related.

Check Your BSDL Files

The BSDL files for the device indicate the number of bits for the instruction and data registers of various ISP operations for a device. If this information is incorrect in the board file, the tester will have an incorrect model of the JTAG bitstream and the test will fail.

IDCODE Compatibility

Some JTAG devices do not power up with a 32-bit MFGR-ID register between their TDI and TDO pins. The 1-bit bypass register is placed instead. If your chain configuration assumes a 32-bit MFGR-ID register, then the software and circuit board's JTAG chain will not match and programming will fail. In the Atmel-ISP software there is an option to enable or disable IDCODE compatibility. IDCODE compatibility is enabled by default for Atmel devices, but needs to be disabled for non-Atmel devices without an MFGR-ID register. Check with the device manufacturer for details.

Do Not Mix Flow Files

There are two kinds of tests the HP3070 can perform: boundary-scan tests and JTAG programming tests. Boundary-scan tests create different PCF files from JTAG programming tests. Separate your boundary-scan test files in a different directory from your JTAG programming files. If your boundary-scan test files include a pin library, it is recommended that you do not use it for your JTAG programming tests. The pin library may contain conflicting and/or incorrect information about JTAG pins on the Atmel device. This may prevent the tests from executing correctly. Modify your board file accordingly to make sure it has all the correct pin information for the JTAG programming tests.

Programming Problems

Isolate whether the problem is test fixture-related or caused by the device. There are two steps to try:

1. Use the HP3070 to power-on the board and disconnect the JTAG signals. Then use the Atmel-ISP software and Atmel-ISP download cable from the PC to program the JTAG device chain. If this test passes, the problem is likely the test fixture. In particular, the JTAG signals. If this test fails, then the problem is either device- or computer-related. Contact Atmel EPLD Applications if you need help using the Atmel-ISP software to program devices on your board.
2. Connect both JTAG and power to the board but disconnect all other signals. If programming is successful, then the problem may be caused by noise and/or crosstalk between signals in the fixture. If this test fails, then the problem can be either the tests were not being performed correctly or there is a fixture problem. Check for noise or crosstalk on the JTAG signal pins on the fixture or change your test fixture to a bi-level test fixture. A bi-level test fixture is the preferred approach to reducing and/or eliminating most noise or crosstalk problems on the HP3070.

Programming Times

The HP3070 is able to create very accurate programming times. The TCK frequency is the main variable that will affect the programming time. Due to the simple algorithm Atmel uses to program all devices in the ATF1500AS family, the program time will not vary significantly between devices in the family. Table 2 describes the programming time measured for one ATF1508AS-15JC84 device, tested for three vector cycle times.

Table 2. Programming Time for ATF1508AS-15JC84 Device

	TCK = 500 kHz	**TCK = 2 MHz**	**TCK = 3 MHz**
Vector Cycle Time	1000 ns	500 ns	170 ns
Programming Time	4.3 sec	4.0 sec	3.5 sec
	PCF with Comments	**PCF without Comments**	
PCF File Size	5.5M bytes	2.0M bytes	
Compile Time	7 minutes	2 minutes	
Number of PCF Files	1		
Number of Vectors per File	About 300,000		
Controller Type	725/100		

The programming time did vary with TCK frequency, but not to a significant degree. Compile time decreased dramatically when the PCF file did not include comments. Programming time for device chains of more than one device should not be much longer because of the concurrent programming feature available with JTAG programming. The number of vectors on Atmel PCF files are much smaller due to a simpler Atmel programming algorithm that extends to all members of the family. Smaller devices of the family (for example, the ATF1504AS) should have fewer vectors so programming time may be slightly shorter on the HP3070.

Total compile and setup time for the test depends on the number of PCF file(s) that need to be loaded. However, the combination of a simplified programming algorithm, PCF file size and removing comments from the PCF file can all help to further reduce programming time.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1651A–04/00/xM