Application Note

Programming Actel FPGA's Using UniSite or the 3900 Programmer

Actel Field Programmable Gate Arrays (FPGAs) require special data processing during programming. You must use downloading and programming procedures that are unique to these devices.

Actel (ActI) devices 1010, 1010A, 1010B,1020, 1020A, and 1020B can be programmed one at a time with the following Data I/O programmers:

- UniSite with a PinSite module and either a PPI Base and adapter, or the PLCC base (depending on the device package)
- 3900 with a PPI base and adapter

This Application Note describes programming Actel FPGAs using the PinSite module.

UniSite can also program eight devices at a time using the USM-340 Actel Gang Programming Module. Operation of the Gang module is described in its User Manual.

The UniSite and 3900 can also program Actel (ActII) devices 1225, 1225A, 1240, 1240A, 1280, and 1280A one at a time, using the UniSite PinSite module (or the 3900) with a PPI Base and adapter or the PLCC base (depending on the package type). At the time of printing, the Actel Gang Module does not support ActII parts.

Theory of Operation

To program Actel devices, UniSite operates in conjunction with the Actel Action Logic Development System. Since a programmed device has no read mode, it cannot serve as a master device; therefore, the data to be programmed must be downloaded at the start of each programming session. Since the device fuse file is normally too large to be stored on the UniSite floppy disk (which can store a maximum of 730K), it must be downloaded to RAM from the host.

The high speed download feature (Remote Port only) of the UniSite can significantly reduce the file download time.

Equipment Requirements

Data I/O

- UniSite Programmer with 17 pin driver boards (68 pin drivers), or a 3900 Programmer
- UniSite PinSite module with PLCC Base or PPI Base and PPI adapter(s), or a 3900 PPI Base and PPI adapter(s)
- System software version that supports the desired part:
  UniSite: Currently V3.30 or greater for ACT1 devices and V4.0 or later for ACT2 devices
  3900: V2.40 or greater
- Second RS-232 serial cable (optional)

- Programmer Memory requirements (maximum*):

1010, 1010A, 1010B	2 MB (plus .5 MB for system code)
1020, 1020A, 1020B	2 MB (plus .5 MB for system code)
1225, 1225A	4 MB (plus .5 MB for system code)
1240, 1240A	6 MB (plus .5 MB for system code)
1280, 1280A	8 MB (plus .5 MB for system code)

Actel

- Action Logic System (ALS) development software running on a 80386-based DOS system, Sun-3, or Apollo workstation
- Output file translator software FUS2DIO

Setup

If you use one workstation to both control UniSite (using a terminal emulator program) and run the Actel development software, you need only one serial connection to UniSite, usually between the PC COM port and UniSite terminal port. However, if this workstation is a DOS platform and you are using the HiTerm terminal emulator program, connect the PC COM port to the UniSite remote port. HiTerm, which enables you to download device fuse files at up to 115.2K baud, is supplied on the UniSite Utility Disk. The *UniSite User Manual* describes HiTerm file transfers and high speed download.

If you run the Actel development software on a system other than the one controlling UniSite, connect that system to the UniSite remote port. By using the Remote port, you can access the high-speed download option.

Since the file to be downloaded is in 8-bit binary format (format **10**), set the serial ports used for file transfer to **8** data bits. Also, set up the host so there is no post-processing of the file during transfer. For example, there should be no control character reaction or filtering, and no end-of-line carriage return or line feed addition or deletion.

File Transfer

The fuse file is first generated using the Actel Action Logic System, then converted to a binary transfer format using the translation software (FUS2DIO), and finally downloaded.

Two sample sessions in the *UniSite User Manual* describe downloading a file from a remote host and from a controlling PC using the HiTerm terminal emulation program. Refer to these sessions for more information on performing these functions.

Following is a summary of the steps performed during the download procedure:

1. Select the Actel device to be programmed.
2. Press **M**, **T**, then **D** (More commands/Transfer data/Download) to bring up the Download Data From Host screen.
3. Select the source port **R** (remote) or **T** (terminal).
4. Select I/O translation format **10** (Formatted Binary).
5. Leave all other parameters in the default state.
6. Enter the appropriate Download host command to download the fuse file. (Refer to the sample sessions for more information.)
7. Press the Enter key to initiate the file transfer.

* *This is the maximum amount of RAM that may be required. Some designs may require less RAM than shown.*

Program Operation

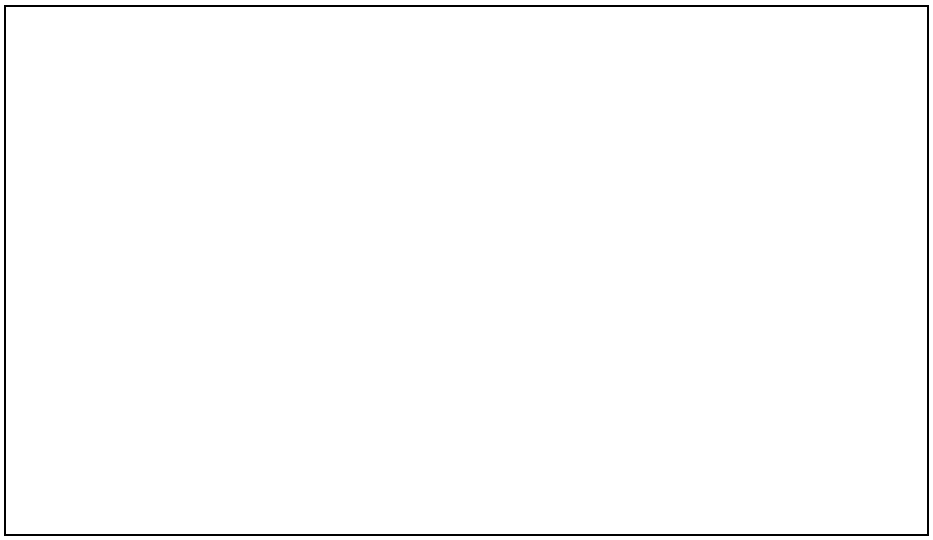
Although Actel FPGAs are logic devices, they are treated as memory devices in the UniSite screen menus. The binary data downloaded to memory is not just a bit map of data to be programmed into the part, but a complex set of instructions, addresses, and antifuse data used by the programming algorithm to program and verify the device. Attempting to edit this data using the programmer's editor is not recommended.

Program only blank devices; non-blank devices will be rejected. While viewing the Programming screen, press **PF4** to bring all parameters to view. To ensure that each device is properly tested before programming begins, set the blank check flag to **Y** . Also, disable the Verify Passes parameter on the Program device screen by setting Verify passes (0,1,2) to **0**.

Verify Operation

Since the verify operation is part of the programming sequence (fuse verification takes place during the programming operation) and the device does not have a read mode, running a separate verify operation would report a failure even though the device was programmed correctly. Therefore, you must disable Verify on the Programming screen by setting Verify passes (0,1,2) to **0**.

When all parameters are properly set, you should see the following screen. Once the parameters are properly defined, press the Enter key to program and verify the device.



Load Operation

The Load device function is not a valid operation for this type of device. Therefore, a programmed device cannot be used as a master. A load operation does not change any information in RAM and results in an "Operation complete" message. The following information is also displayed: the design sumcheck (calculated by the ALS development software), the User ID of the device, and the state of the security fuse (ActII devices only).

Security Fuse Programming

During the Programming Cycle

ActI Devices: Two security fuse modes are supported:

- SEC1 security bit data disables the device debug feature, Actel Action Probe
- SEC2 security bit data disables further device programming

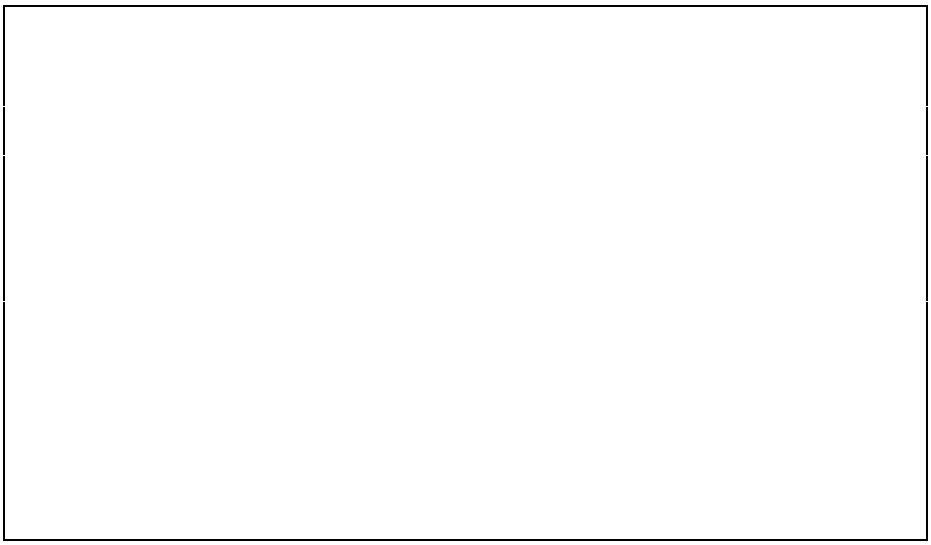
These security fuses can be programmed during the program and verify process for the main array. To enable this feature, set the desired security bit data to **1** and the Program security bits option to **Y**. The two security bits are stored at bits **6** and **7** of the second byte in user memory; therefore, changing these bits will affect the RAM data sumcheck.

ActII Devices: A single security fuse is supported. Set the Program security bit option to **Y**. Once the parameters are properly defined, press the Enter key to program and verify the device.

After the Programming Cycle (ActI devices only)

The security fuses can also be programmed in a separate operation for devices with previously programmed main arrays. To do this, change the first byte in user RAM (address 0000) to 80 (Hex). Then set the other parameters as described above. Also, disable the illegal bit and blank check. To speed up the programming process, change the user data size parameter to 100.

When all parameters are properly set, the following screen should appear. After the parameters are properly defined, press the Enter key to program and verify the security fuses.



Special Note for PROMlink and TaskLink Users

As previously mentioned, the Verify Passes parameter must be disabled (set to **0** in terminal mode) when you program Actel devices. When you use either PROMlink or TaskLink with the programmer, perform the following steps to set up the Task file so that it disables the Verify Passes parameter:

1. With PROMlink or TaskLink software running in non-production mode, access the "Setup" menu using the **ALT+S** key combination.
2. From the Setup menu, select "General Parameters."
3. On the General Parameter's menu enter "**023]**" as a "custom command". 023] is the CRC command that sets the programmer's Verify Passes to 0.

ACTI Error Messages

Generic non-fatal device-specific error
An invalid operation ode has been discovered within the programming data, indicating that the programming data has been corrupted.

Device programming error
A fuse failed to be programmed after the maximum number of programming pulses have been applied.

Invalid device ID
The ID of the device in the socket does not match the ID for the selected device.

Verify error due to non-blank device
The device in the socket is not blank.

Device verification error
The part has failed its own verification cycle.

Failed: programmer algorithm incompatible with DIO data file
The ID within the programming data does not match the expected ID for the selected part.

ACTII Error Messages

Unknown Opcode (Opcode, Select)
An invalid operation code has been discovered within the programming data, indicating that the programming data has been corrupted.

Illegal Header ID (Header ID)
The ID within the programming data does not match the expected ID for the selected part.

Calibration Failure
The programmer hardware was unable to calibrate its current drivers to an acceptable precision.

Programming Failure(Opcode, Select)
A fuse failed to be programmed after the maximum number of programming pulses were applied.

Preload Test Failure: (Opcode, Select)
The part has failed its own verification cycle.

Incorrect Device ID (Device ID)
The ID of the device in the socket does not match the ID for the selected device.