12. Real-Time ISP and ISP Clamp for MAX II Devices

MII51019-1.6

Introduction

During in-system programming, most CPLDs automatically tri-state their input/output (I/O) pins to prevent contention issues on a board. After successful programming, the device enters user mode and the new design begins to function. Apart from this normal programming mode, MAX® II devices also support real-time in-system programmability (ISP) and ISP Clamp programming modes, which allow control of I/O and device behavior during ISP. This chapter describes the following two features and how to use them in the Quartus® II software, as well as the Jam™ Standard Test and Programming Language (STAPL) and Jam STAPL Byte-Code Players:

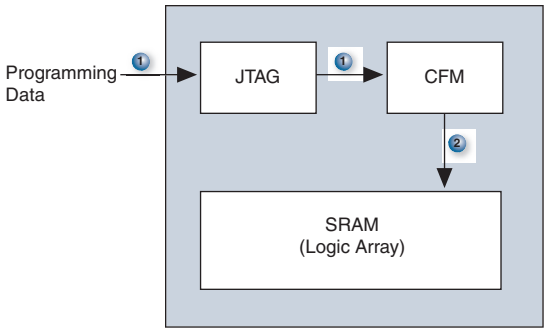
Real-Time ISP

Real-time ISP allows you to program a MAX II device while the device is still in operation. The new design only replaces the existing design when there is a power cycle to the device (i.e., powering down and powering up again). This feature enables you to perform in-field updates to the MAX II device at any time without affecting the operation of the whole system.

How Real-Time ISP Works

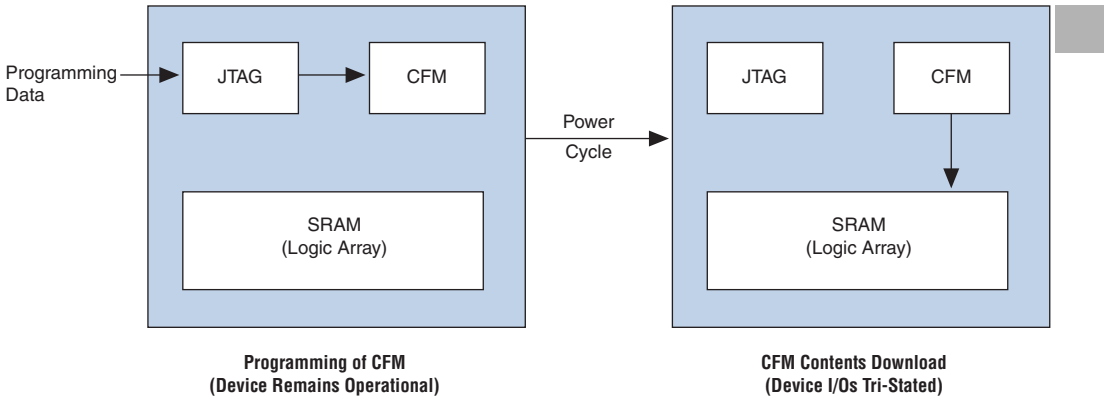
For normal ISP operation, downloading the new design data from the configuration flash memory (CFM) to the SRAM begins after the completion of CFM programming. During the process of CFM programming and subsequent downloading of CFM data to SRAM, I/O pins will remain tri-stated. After the CFM download to the SRAM, the device resets and enters user mode operation. Figure 12–1 shows the flow of normal programming.

Figure 12–1. MAX II Device with Normal ISP Operation



In real-time ISP mode, the user flash memory (UFM), programmable logic, and I/O pins remain operational while programming of the CFM is in progress. The contents of the CFM will not download into the SRAM after the successful programming of the CFM. Instead, the device waits for a power cycle to occur. The normal power-up sequence occurs (CFM downloads to SRAM at power-up) and the device enters user mode after t_{CONFIG} time. Figure 12–2 shows the flow of real-time ISP.

Figure 12–2. Real-Time ISP Operation



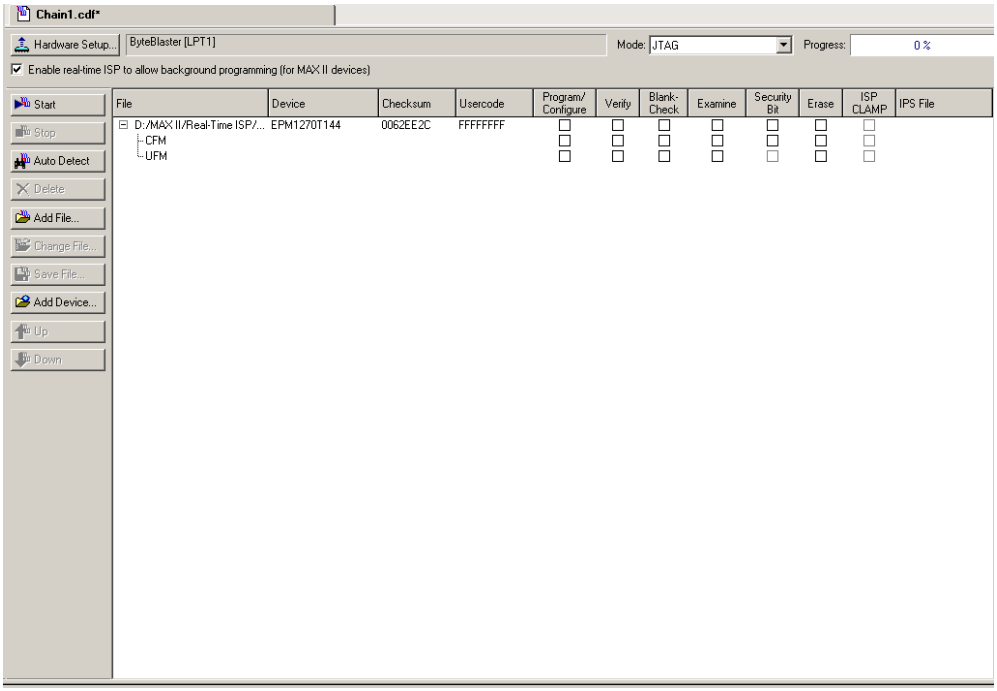
For the t_{CONFIG} value for a specific MAX II device, refer to the *DC and Switching Characteristics* chapter in the *MAX II Device Handbook*.

Real-Time ISP with the Quartus II Software

The programming file formats generated by the Quartus II software that support these two features are the Programmer Object File (**.pof**) that is used with the Quartus II programmer, and the Jam File (**.jam)** and Jam Byte-Code File (**.jbc**) that are used with either the Quartus II programmer or other programming tools.

Ensure that you enable this feature before programming a MAX II device through the Quartus II programmer. You can enable the real-time ISP feature by selecting the **Enable real-time ISP to allow background programming (for MAX II devices)** option from the Quartus II programmer window. See Figure 12–3.

Figure 12–3. Real-Time ISP Option in the Quartus II Programmer Window

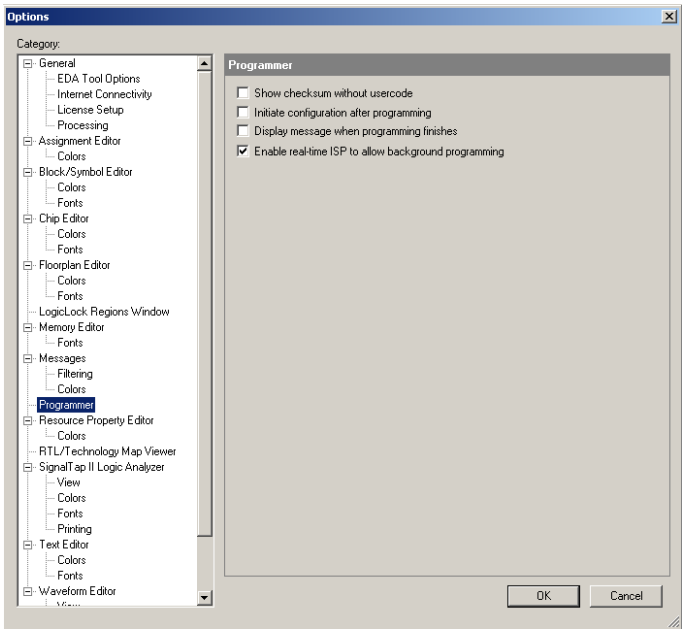


You can also enable the real-time ISP feature in the Quartus II software through the following steps:

1. On the Tools menu, click **Options**.
2. Under **Category**, select **Programmer**.
3. Turn on **Enable real-time ISP to allow background programming (for MAX II devices)** and click **OK**. The MAX II device will go into real-time ISP mode when the Quartus II programmer starts programming it with any one of the three types of programming files.

Figure 12–4 shows the Programmer options in the Options menu.

Figure 12–4. Programmer Options in the Options Menu



Real-Time ISP with Jam and JBC Players

You can use the Jam or JBC file created from the POF to program a MAX II device in real-time ISP mode with the Jam or JBC Player.

For real-time ISP with the Jam File and Jam Player, type the following at the command-line prompt:

```
jp_23 -aprogram -ddo_real_time_isp=1 <file_name.jam>
```

For Real-Time ISP with the JBC File and JBC Player, type the following at the command-line prompt:

```
jbi_22 -aprogram -ddo_real_time_isp=1 <file_name.jbc>
```

The names of the executable files for the players are different, depending on the version of the players. Download the latest version of the Jam and JBC Player from the Altera® web site at www.altera.com.

ISP Clamp

When a MAX II device enters normal ISP operation, all the I/O pins tri-state and are weakly pulled up to V_{CCIO} with internal pull-up resistors. However, there are situations when the I/O pins of the device should not be tri-stated when the device is in ISP operation. For instance, in a running system, some signals (e.g., output enable or chip enable signals) might use some of the I/O pins and require those I/O pins to assume a high or low logic level, or even maintain their current state when the device is in ISP mode.

With the ISP clamp feature in MAX II devices, you can hold each I/O pin of a device to a specified static state when programming the device. You can set the state in the Quartus II software. After successfully programming the device in ISP clamp mode, those I/O pins will be released and function according to the new design.

This feature can be used to indicate when the device is being programmed and when the programming is done by setting a particular pin to a specific state (different from the state when the device is in user mode) when the device enters ISP clamp mode.

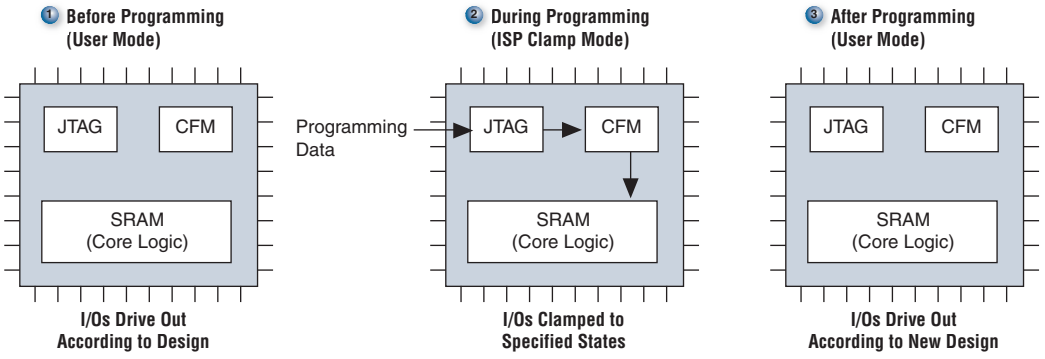
How ISP Clamp Works

When the ISP clamp feature is used, you can set the I/O pins to tri-state (default), high, low, or even sample the existing state of a pin and hold the pin to that state when the device is in ISP clamp operation. The software determines the values to be scanned into the boundary-scan registers of each I/O pin, based on your settings. This will determine the state of the pins to be clamped to when the device programming is in progress. The weak I/O pull-up resistors are disabled during programming when the ISP clamp feature is used, even if the I/O is clamped to a tri-state value.

Before clamping the I/O pins, the SAMPLE/PRELOAD JTAG instruction is first executed to load the appropriate values to the boundary-scan registers. After loading the boundary-scan registers with the appropriate values, the EXTEST instruction is executed to clamp the I/O pins to the specific values loaded into the boundary-scan registers during SAMPLE/PRELOAD.

If you choose to sample the existing state of a pin and hold the pin to that state when the device enters ISP clamp mode, you must make sure that the signal is in steady state. You need a steady state signal because you cannot control the sample set-up time as it depends on the TCK frequency as well as the download cable and software. You might not capture the correct value when sampling a signal that toggles or is not static for long periods of time. Figure 12–5 shows the ISP clamp operation.

Figure 12–5. ISP Clamp Operation



Using ISP Clamp in the Quartus II Software

You have to define the states of the I/O pins to use the ISP clamp feature. There are two ways to define the pin states in the Quartus II software. You can either:

- Use an I/O Pin State file (**.ips**), or
- Use the Assignment Editor to set the clamp states of the pins

Using the IPS File

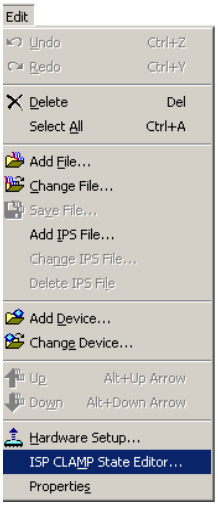
Creating an IPS File

You can specify the clamp states of the pins when the device is in ISP clamp operation without configuring the settings in the Assignment Editor and recompiling the design. You must first create a new I/O pin state file (**.ips**) file and define the states of the pins in the file, or use an existing IPS file. The IPS file defines the states for all the pins of the device when the device is in ISP clamp operation. The file created is usable for programming the device with any designs, as long as it targets the same device and package. An IPS file must be used together with a POF file, which contains the programming data to program the device.

To create an IPS file, perform the following:

1. Click **Programmer** on the toolbar, or on the Tools menu, click **Programmer** to open the Quartus II Programmer window.
2. Click **Add File** in the programmer to add the programming file (POF, Jam, or JBC) into the programmer window.
3. Click on the programming file in the programmer (the entire row will be highlighted) and on the Edit menu, click **ISP CLAMP State Editor**. See Figure 12–6.

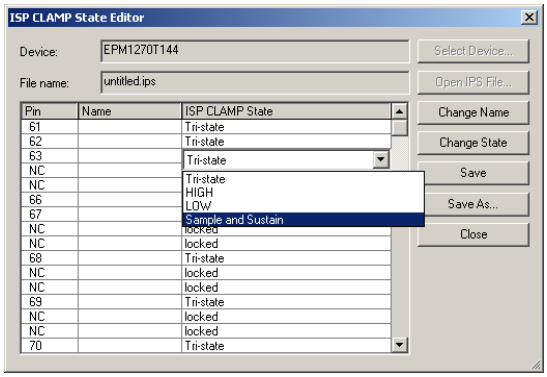
Figure 12–6. Edit Menu



4. Specify the states of the pins in your design in the ISP Clamp State Editor. There are four clamp state choices: tri-state, high, low, and sample and sustain. By default, all pins are set to tri-state.
5. Save the IPS file after making the modifications.

Figure 12–7 shows the ISP Clamp State Editor. On the File menu, you can also click **Create/Update** > **Create/Update IPS File** to open the ISP Clamp State Editor and create a new IPS file.

Figure 12–7. ISP Clamp State Editor



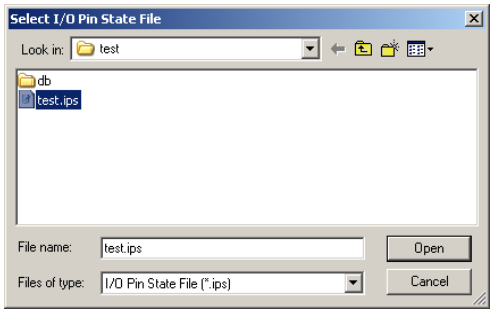
Using the IPS File

In the Quartus II Programmer, you must specify the IPS file you want to use by performing the following steps:

1. Double-click on the cell under the IPS File column. The **Select I/O Pin State File** menu appears.
2. Choose the IPS file for your project and click **Open**.

You can also left-click on the programming file (this will highlight the entire row) and on the Edit menu, click **Add IPS File** to open the Select I/O Pin State File dialog box as shown in Figure 12–8.

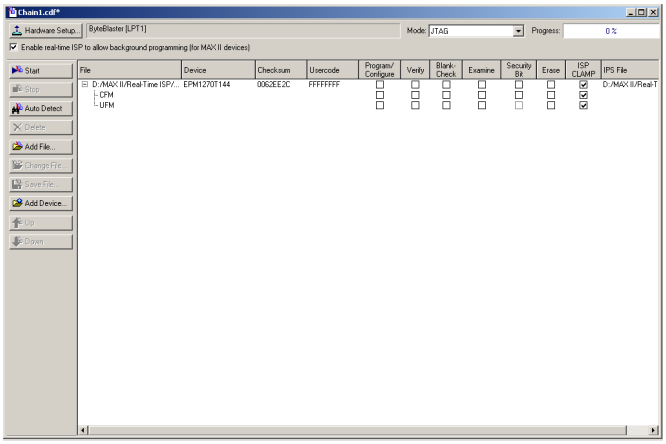
Figure 12–8. Select I/O Pin State File Menu



1. The IPS file you have selected will be listed in the Quartus II Programmer window, as shown in Figure 12–9.

Make sure the ISP CLAMP check box is checked before you start programming your device.

Figure 12–9. The Quartus II Programmer Window with the Specific IPS File



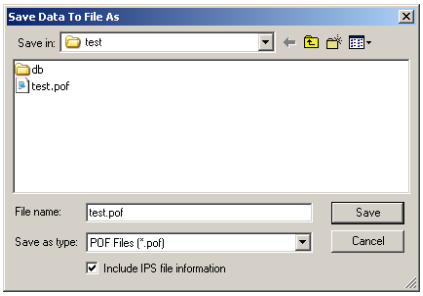
Saving the IPS File Information to the Programming File

The pin state information in the IPS file can be saved into the POF to avoid requiring two files. You will only need the programming file to program a device in ISP clamp mode. This programming file is also used for creating the Jam and JBC files for the ISP clamp so that the Jam or JBC files will contain the pin state information. The following are the steps to save the pin state information from the IPS file to the programming files.

1. Add in the programming file in the programmer window.
2. Add in the IPS file to the programmer.
3. Click **Save File** in the programmer window or on the Edit menu, and the **Save Data To File** As dialog box will appear. See Figure 12–10.
4. Enter the file name, check the **Include IPS file information** box, and click **Save**.

The POF with saved IPS information only supports ISP clamp operation in the Quartus II software and not with third-party programming tools. For third-party tools, Jam or JBC files should be used if ISP clamp is required.

Figure 12–10. Save Data To File as Menu



When programming a device with the ISP Clamp box checked, the Quartus II Programmer will first look for the IPS file. If the IPS file is not found, only then it will look into the POF for the pin state information.

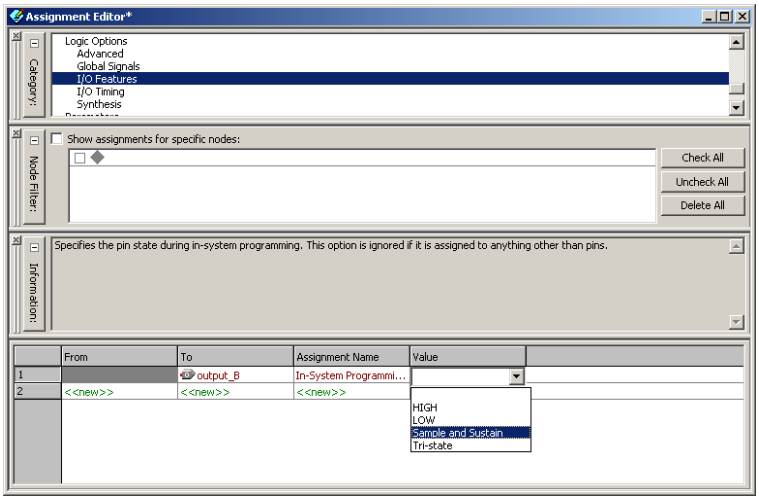
Defining the Pin States in Assignment Editor

Another way to define the pin states is through the Assignment Editor. After you have defined the pin states in the Assignment Editor and compile the design, the programming file generated will have all the pin state information in it. The following are the assignment editor states:

1. Click **Start Analysis and Synthesis** on the toolbar.
2. On the Assignments menu, click **Assignment Editor**.
3. In the Assignment Editor, under **Category**, select **I/O Features**.
4. List down all the pins you wish to clamp when the device is in ISP clamp mode under the **To** column. You can use the Node Finder to help you select the pins.
5. Select **In-System Programming Clamp State** for all the pins under Assignment Name after you have listed down the pins you wish to set state values.
6. Define the states for each of the pins under Value. You can also choose to clamp the pins to high, low, tri-state, or sample and sustain the pin state. By default, the pins are tri-stated when the device enters ISP clamp mode.

Figure 12–11 shows how to define the states of the pins in the Assignment Editor.

Figure 12–11. Assignment Editor



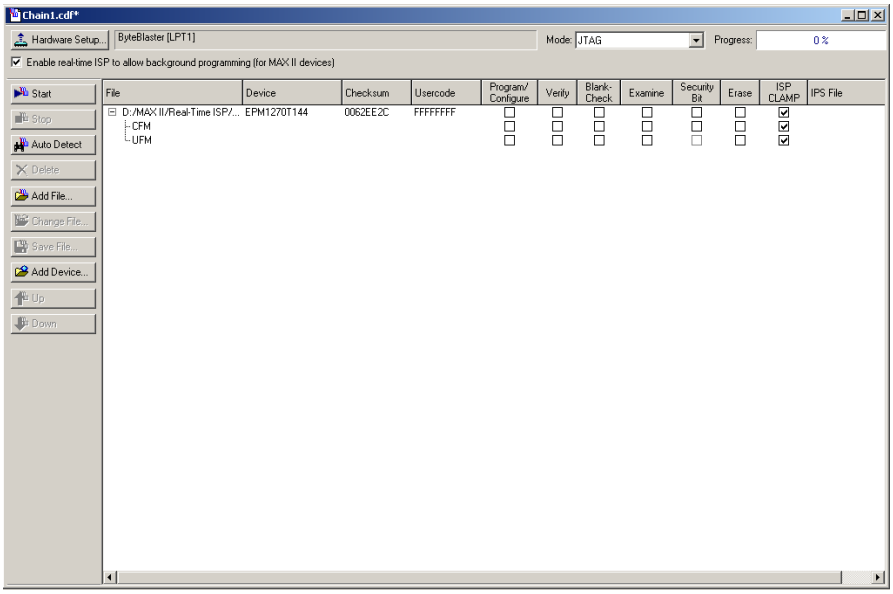
1. Save the assignments and recompile your design.

After you have recompiled the design, the ISP clamp state information will be stored in the POF. You can also view the settings in the Quartus II Settings File (**.qsf**).

Running ISP Clamp in the Quartus II Programmer

In the Quartus II Programmer window, make sure that the ISP Clamp check box is checked before programming the device. Do not add any IPS file in the programmer as the programmer will use the values specified in the IPS file instead of the values you set in the Assignment Editor (which is stored in the POF). Figure 12–12 shows the **Quartus II Programmer** window with ISP Clamp checkbox. Jam and JBC files created using the POF will have the pin state information in them.

Figure 12–12. Quartus II Programmer Window with ISP Clamp Checkbox



ISP Clamp with Jam/JBC Files

The Jam or JBC files used for ISP clamp should contain all the pin state information and do not need any IPS file. Always use the POF file with pin state information to create the Jam or JBC files. The pin state information can be stored into the POF through the Assignment Editor or saving the pin state information to the POF as mentions earlier. The Jam or JBC files can be used with the Quartus II programmer, or with the Jam or JBC player, respectively.

Conclusion

With the real-time ISP and ISP clamp features in MAX II devices, you can set the I/O pins of a device to certain states while programming the device. Through real-time ISP, you can program a MAX II device at any time without affecting the functionality of your system. The ISP clamp feature allows you to hold the I/O pins of a device to specific states when programming the device.

Referenced Documents

This chapter references the following document:

- *DC and Switching Characteristics* chapter in the *MAX II Device Handbook*

Document Revision History

Table 12–1 shows the revision history for this chapter.

Table 12–1. Document Revision History

Date and Revision	Changes Made	Summary of Changes
October 2008, version 1.6	■ Updated New Document Format.	—
December 2007, version 1.5	■ Added "Referenced Documents" section.	—
December 2006, version 1.4	■ Added document revision history.	—
February 2006, version 1.3	■ Updated the "Real-Time ISP with the Quartus II Software" section. ■ Added Figure 12–3. ■ Updated Figure 12–9 and 12–10.	—
June 2005, version 1.2	■ Updated the first paragraph of the How ISP Clamp Works section.	—
January 2005, version 1.1	■ Previously published as Chapter 13. No changes to content.	—