White Paper

Reduce Manufacturing Costs by Integrating Flash Device Programming

Introduction

Standard flash memory devices are used in many applications to store configuration, program, or memory data. Before these flash devices can be used in a system, they must be programmed. Using traditional methods, this programming can absorb large amounts of time in the manufacturing process. These methods provide little flexibility for last-minute changes or in-field programming updates, an increasingly necessary ability needed for design changes that add features or address bugs. The Altera® Parallel Flash Loader (PFL) solution addresses these issues, thereby simplifying manufacturing processes, lowering costs, and providing other benefits.

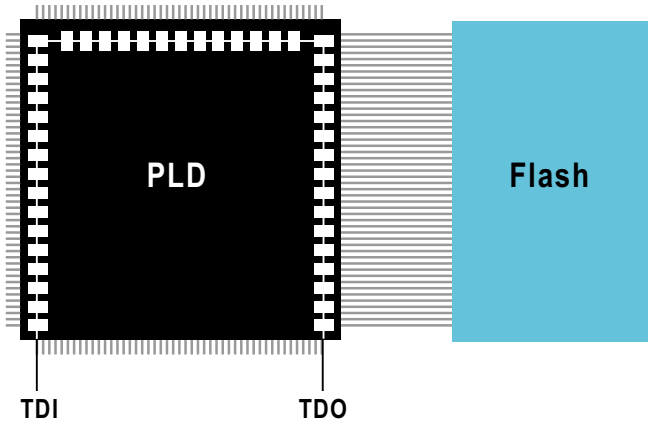
Traditional Options for Programming Flash Memory Devices

There are three commonly-used options to program flash memory devices. The first is to program the device before mounting it onto the printed circuit board (PCB). This approach requires extra programming fixtures, increases manufacturing cost, and does not allow for last-minute changes, enhancements, or bug fixes that may be necessary after the flash device is mounted onto the PCB.

The second option is to use a processor on the same PCB to program the flash memory device via in-system programability (ISP). The processor accesses the programming code either in-system or through in-circuit emulation hardware, which adds equipment and an extra step to the manufacturing process. This method is inefficient because the processor must first access the data from a different source, store it in available RAM, and finally program the flash memory device.

The third option uses ISP with a JTAG boundary scan chain to control the pins connected to the flash memory device. This option is used oftenbecause many flash memory devices do not support the JTAG interface. In this approach, the flash memory device is connected to a JTAG-compliant device on the PCB, which acts as a programming host, as shown in Figure 1. This method requires shifting hundreds of bits of data through the entire JTAG boundary scan chain to write a few bits of data to the flash memory device, also an inefficient process. This option also requires the programming host device to suspend normal operation, which impacts any devices connected to it

Figure 1. Programming a Flash Device Through the JTAG Boundary Scan Chain.



Integrated PFL Shortens Programming Times

Altera has developed a PFL solution that provides an easy, cost-effective way to program flash devices through the JTAG interface. The solution uses a MAX® II CPLD to bridge the JTAG interface and the flash device's parallel address/data interface. Instead of shifting data through all of the pins on the MAX II device, this solution quickly retrieves data from the JTAG scan chain and generates data formatted for the receiving target flash device.

WP-92005-1.2

This solution significantly reduces the flash device programming time. Using the example of programming a single vector into a 48-pin common flash interface (CFI) flash device, Table 1 shows the amount of time savings possible using the PFL solution. The example compares using the PFL solution to the traditional method of programming via the JTAG boundary scan chain using a JTAG-compatible PLD or ASIC that has approximately 200 pins.

Table 1. Comparison of JTAG Boundary Scan Programming Method to PFL Method

Programming Method	Clock Cycles Required
JTAG Boundary Scan Chain	1,200
Altera PFL	48

In addition to shorter programming times, the PFL solution can configure Altera FPGAs on the same PCB, using FPGA configuration data stored in a flash device. The PFL logic determines when to start the configuration process, reads the data from the flash memory device, and configures the Altera FPGA accordingly. Using this method, hardware developers avoid using dedicated FPGA configuration devices, thereby reducing component costs, shrinking board size, and simplifying board design.

PFL Solution Offers Ease-of-Use and Low Cost of Adoption

The PFL can be easily integrated into a MAX II CPLD by using a simple GUI within Altera's Quartus® II development software. The GUI enables the user to set the clock frequency, flash memory type, byte address of the option bits, and desired supplemental files. The Quartus II software automatically generates the needed logic for implementation in the Altera device. Quartus II Web Edition software supports design development for MAX II CPLDs and the PFL solution, and is available free of charge. Additionally, the PFL function can fit into the smallest MAX II CPLD, resulting in a very low component cost. Table 2 compares these advantages and others resulting from using the PFL solution with those of traditional flash device programming methods.

Table 2. Advantages and Disadvantages of Various Flash Device Programming Solutions

Flash Programming Method	Programming Time	Advantages	Disadvantages
Program before mounting on PCB	Short	• Easy to implement • No PCB layout considerations	• Inflexible • Flash devices cannot be reprogrammed after mounted on PCB
ISP programming via microprocessor	Medium	• Generally leverages existing on-board processor	• Requires additional hardware and RAM • Adds to manufacturing complexity
ISP programming via JTAG Boundary Scan Chain	Long	• Generally leverages existing JTAG boundary scan chain	• Requires suspension of programming host operation during programming process
Altera PFL	Short	• Easy to implement • Allows last-minute programming changes and reprogramming in the field • Enables programming without disrupting system • Low cost of adoption • Provides FPGA configuration	None

Ease RoHS Transition With Altera Products

Altera maintains one of the most extensive RoHS-compliant product offerings in the industry, with over 1200 products in lead-free packages. As a preeminent supplier of environmentally friendly programmable logic solutions, Altera has shipped over 25 million RoHS-compliant products since 2002. Altera's devices comply with the maximum concentration restrictions, as required in the EU Directive on the Restriction of Hazardous Substances ("RoHS Directive") No.2002/95 with respect to lead (Pb), mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB), and polybrominated diphenyl ethers (PBDE). Help ease your RoHS transition by integrating non-compliant ASSPs with Altera's PLDs.

Conclusion

While utilizing ISP with a JTAG boundary scan chain is often the method of choice for programming today's flash memory devices, this approach has limitations that must be considered. These include requiring considerable data shifting through the entire JTAG boundary scan chain to write just a few bits of data. Engineers can take advantage of a PFL solution for an easy, cost-effective way to program flash memory devices through the JTAG interface. Such a solution offers many benefits, including shorter flash memory device programming times, ease of use, and low adoption cost.

Additional Resources

- Parallel Flash Loader for MAX II CPLDs:
  www.altera.com/products/devices/cpld/max2/applications/bridge/mx2-jtag_translator.html
- MAX II CPLD Family: The Lowest-Power, Lowest-Cost CPLDs Ever:
  www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- *AN 386: Using the MAX II Parallel Flash Loader With the Quartus II Software*:
  www.altera.com/literature/an/an386.pdf
- Quartus II Web Edition Software Download Page:
  https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp

101 Innovation Drive
San Jose, CA 95134
www.altera.com