Flash programming using boundary scan and an FPGA

Dominic Plunkett describes how an alternative flash programming method achieves near theoretical programming times.

KEY POINTS

- Flash memories enable engineers to reprogram in situ, streamlining design
- Testing boards via JTAG has advantages, but JTAG flash programming can be slow
- FPGA-based programming overcomes the speed limitation of JTAG programming

Relatively inexpensive, high-density flash memories are good news for product developers under pressure to implement more and better functions in next-generation designs. Reprogramming in situ not only streamlines design and prototyping, but also delivers several advantages when used in the end product.

For example, several different variants can share a common hardware platform, to be loaded with the correct program at the end of the line. Manufacturers can also streamline purchasing and inventory by using fewer different types of memory ICs across a wide variety of products.

Later in the product lifecycle, bug fixes and new features can be applied in the field. Moreover, depending on the type of product, such updates can be applied remotely.

To gain maximum benefit from flash, engineers need methods for programming without removing memories from the board

Flash programming sequence

To gain the maximum benefit from flash, engineers need convenient and effective methods for programming without removing memories from the board.

Chip designers are working hard to simplify read and write operations, but depending on the type of device, a program sequence may require four or more bus cycles to issue the required programming commands. A dedicated programmer may be used to manage this sequence, or the device may be programmed via a PC.

In either case, the designer must consider the programming strategy near the beginning of the project, and arrange the necessary access to address and data pins, as well as control pins such as write-enable (WE) and ready/busy (RDY/BSY).

When a board is being tested via JTAG, it is relatively easy to incorporate flash programming without adding any extra equipment. Using the JTAG scan chain may also simplify the PCB and produce a smaller final product.

However, relative to the speed of the flash device, JTAG flash programming can be slow, since the control, address and data bits must be shifted serially through the boundary scan chain. This extra time taken to program the memory can offset the advantages of programming via the JTAG port and as memories become larger, the time penalty becomes more acute.

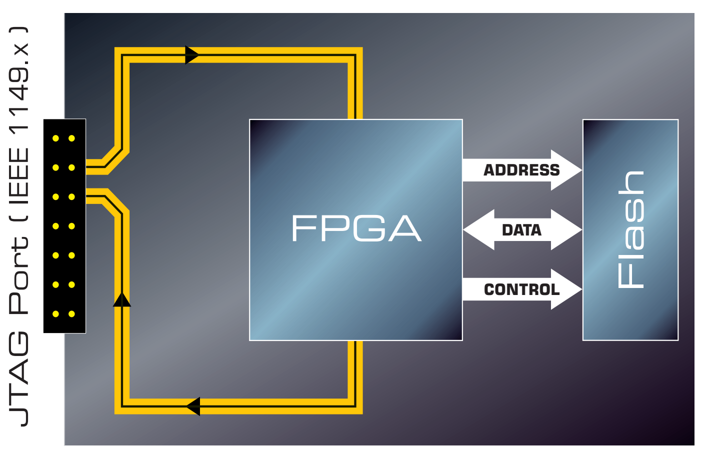
Faster JTAG programming

This can be overcome by implementing a method for programming flash memories that uses an FPGA, which

Reprogramming in situ streamlines design and gains maximum benefit from flash is often already onboard and connected to the flash device. This relieves the JTAG equipment of repetitive tasks such as shifting in control and address data, thereby speeding up programming by removing traffic from the JTAG chain. The chain is used almost exclusively for shifting program data into the memory, which makes greater use of its limited bandwidth.

The FPGA is used to generate the control and address signals that are required. In effect, a simple flash programmer is created using the FPGA's internal circuitry (*see figure 1*).

FIGURE 1: FPGA PERFORMING FLASH PROGRAMMING



Upon receiving initial commands, the FPGA then co-ordinates each cycle as the program data is streamed via the JTAG chain. Depending on the type of memory IC and the number of other devices in the JTAG chain, programming time can be close to the theoretical time for a given flash device.

The ability to program flash through the JTAG chain at close to the maximum speed possible allows engineers to use the same tool to perform boundary scan tests as well as to apply program data.

The JTAG chain is used for shifting program data into the memory, making greater use of its limited bandwidth

Time and cost savings

This saves engineers from giving special consideration as to how best to program the on board flash, and also saves the time and costs associated with using extra equipment purely for programming. The boundary scan tests and programming script can also be re-used throughout development, production and field maintenance.

With increasing use of boundary scan as a main test strategy to overcome the test coverage challenges arising from poor physical access for conventional test equipment, using the same equipment to program devices on the board can potentially save time and promote ease of use.

By overcoming the traditional speed limitation of JTAG programming, for flash memories, FPGA-based programming allows for quick and easy programming on the developer's bench, on the factory floor, and in the field. ●

Dominic Plunkett is chief technology officer at XJTAG

XJTAG website
www.xjtag.com