AN1424
APPLICATION NOTE

Configuring FPGAs with FLASH+PSD

CONTENTS

There are several well-documented methods for configuring Field Programmable Gate Arrays (FPGAs). Each method involves transferring configuration data from some sort of Non-Volatile Memory (NVM) to the FPGA. There are usually many modes for doing the transfer, including stand alone (Master Mode), or in conjunction with an embedded microcontroller (Peripheral or Slave Mode). The method chosen depends on many factors specific to the particular application, such as:

- Speed of configuration
- Number of I/O pins available
- Microcontroller configuration
- Simplicity of design
- Multiple configuration files
- Board space
- Cost.

Invariably, cost is usually the main factor for determining the method chosen.

*Easy*FLASH™ PSD8XXF devices are members of a family of Flash memory-based peripherals for use with embedded systems. These programmable system devices (PSDs) consist of memory, logic, and I/O. When coupled with a low-cost microcontroller (MCU), the PSD forms a complete embedded Flash memory system that is 100% In-System Programmable (ISP).

This application note shows the benefits of using a PSD8XXF to provide the necessary functions for configuring an FPGA. The PSD8XXF devices are not only extremely low cost compared to other solutions, but also provide all of the inherent benefits associated with Flash-based PSDs, such as:

- In-system programmable and (re)configurable
- Additional memory (Flash, optional SRAM, and optional secondary Flash memory or EEPROM)
- Low power consumption
- Integration of many parts, including memory, programmable logic, decode logic, and security
- Flexibility
- Improved reliability
- Reduced number of components
- Board space savings

■ Reduced development time.

FPGA CONFIGURATION BASICS

Each manufacturer of FPGAs has their own unique configuration format. Usually, the configuration is made up of an internal data structure containing preamble bits, length count, data frame size, and so on. The configuration is normally generated by the manufacturer's development software. Although this note focuses specifically on the Xilinx families of FPGAs, the general concepts presented are applicable to all configurable FPGAs.

Xilinx Specific

There are at least six different programming modes available, which are user selectable via three mode pins on the FPGA: M0, M1, and M2. The programming modes include:

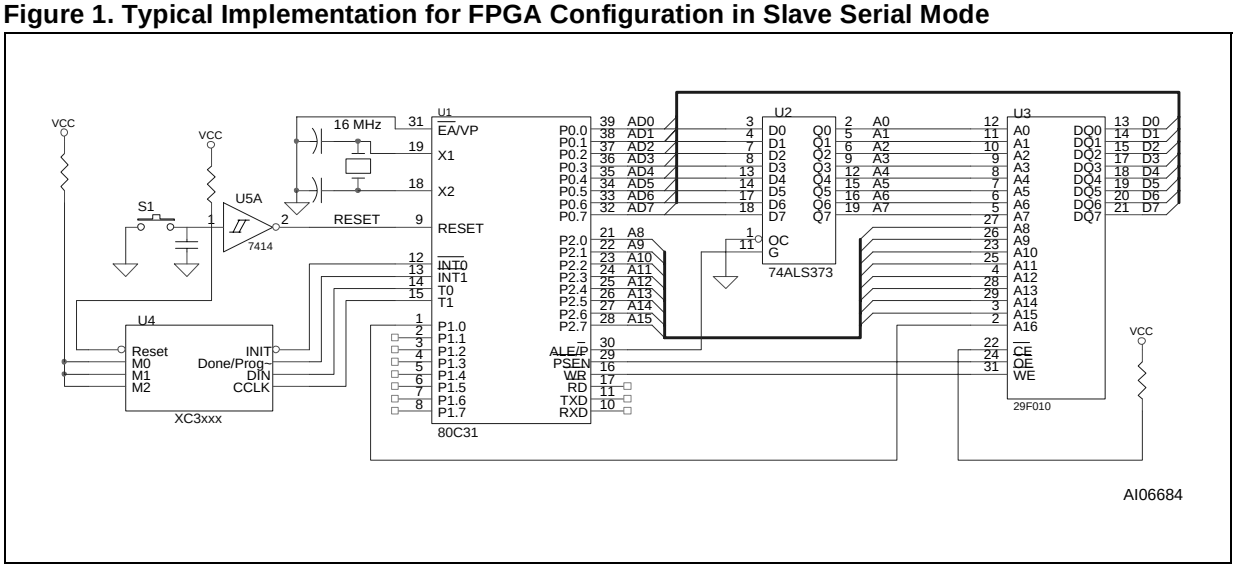
– Mode 0—Master Serial
– Mode 1—Master Parallel (Address = 0000 up)
– Mode 2—Slave Parallel (Express Mode—XC4000EX and XC5200 families only)
– Mode 3—Master Parallel (Address =FFFF down)
– Mode 4—Reserved
– Mode 5—Peripheral Parallel
– Mode 6—Reserved
– Mode 7—Slave Serial

For a complete description of these modes, refer to Xilinx's data sheets.

Typical Implementation

A typical application using an 8031 microcontroller connected to an external 128 Kbyte Flash, an octal latch, and an FPGA configured in Slave Serial Mode, is shown in Figure 1. Both the program code and the FPGA configuration code are stored in the Flash. This is the simplest method for configuring the FPGA since the MCU simply bit-bangs data into it.

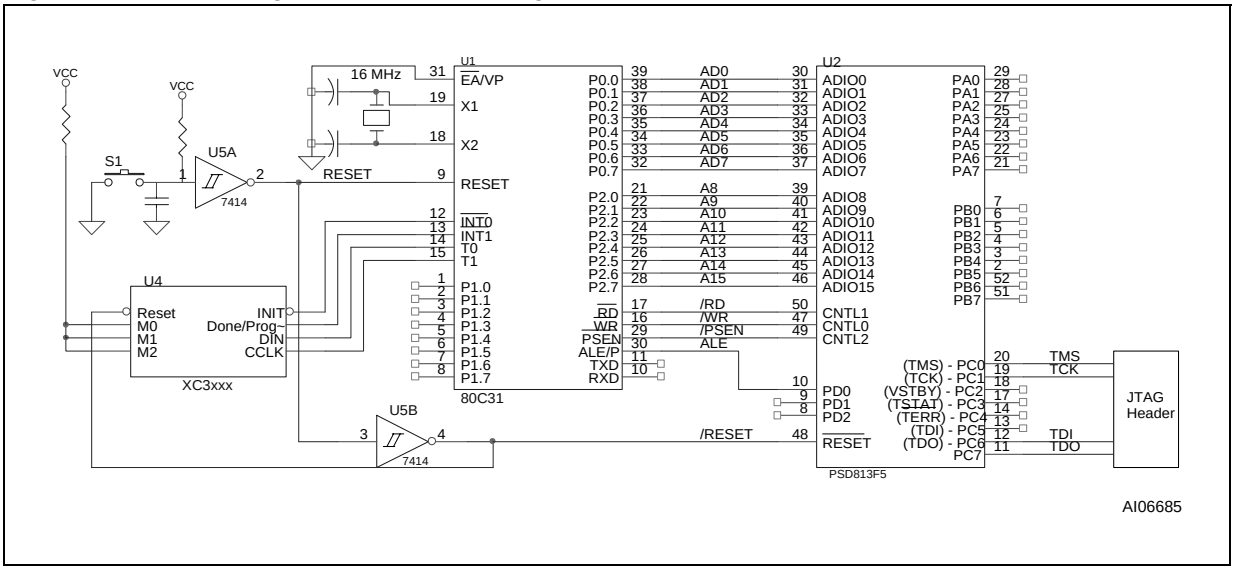
Figure 1. Typical Implementation for FPGA Configuration in Slave Serial Mode



PSD Solution

Although the solution above is simple, the Flash memory cannot be programmed or updated after it is soldered to the PC board. Using a low cost PSD8XXF device provides the optimal solution for configurable FPGA applications. The PSD8XXF contains the needed memory to store the system program code, the FPGA program code, and the FPGA configuration data. The memory of the PSD can be programmed/ updated over the JTAG channel at any time. Figure 2 shows one example of how a PSD8XXF might interface to an FPGA configured in Slave Serial Mode.

Figure 2. FPGA Configuration Circuit using a PSD8XXF



Chip selects for internal components and external devices are generated by the PSD's Decode PLD (DPLD). The address demultiplexing latch shown in Figure 1 (U2) is absorbed by the programmable MCU interface of the PSD. The PSD also replaces the two microcontroller ports lost when accessing external memory, which can be used for general purpose I/O or to provide the control signals needed to interface to the FPGA. Since the PSD provides plenty of memory, it could be used to store multiple configurations for the FPGA. These configurations could be changed (re-programmed) in-system when necessary using the JTAG port.

The flexibility of the PSD8XXF I/O ports allow them to be configured for many functions, including:

- Standard I/O
- Chip select outputs
- Latched address outputs
- Additional address inputs
- Registered I/O.

The circuit designer now has the freedom to select whichever FPGA mode is best suited to the particular application.

Another option for the designer would be to use Peripheral Mode or Express mode. In this case, one of the PSD8XXF ports would be configured as an 8-bit parallel data port. Some of the other port pins could then be used to provide any additional control signals necessary for that particular mode.

Also note that the Output Micro⇔Cells can be connected to form a loadable shift register that could be used to load FPGAs in parallel. See Application Note 55 for more information on CPLD usage.

SUMMARY

The PSD8XXF family is an ideal solution for embedded control applications that use configurable FPGAs because multiple configurations can be stored in the PSD and updated as necessary using the JTAG port. Also, with the PSD, you have four multi-purpose ports that can be used in conjunction with the MCU, FPGA, and other external devices to form a complete solution.

REFERENCES

For more information on the PSD8XXF family and its uses, visit our web site at www.st.com/psm and the following documents:

- PSD813F Family Data Sheet
- *Application Note AN1153— JTAG Information—PSD8XXF* for detailed use of the JTAG channel
- *Application Note AN1171—CPLD Primer—PSD8XXF* for details on the CPLD and I/O pins
- *Application Note AN1154—PSD813F1/80C31 Design Tutorial* for details on PSD I/O, GPLD, logic simulation, and PSDsoft features.

Table 1. Document Revision History

Date	Rev.	Description of Revision
Sep-1999	1.0	Document written (AN065) in the WSI format
03-Jan-2002	1.1	Front page, and back two pages, in ST format, added to the PDF file References to Waferscale, WSI and PSDsoft 2000 updated to ST, ST and PSDsoft Express
12-Apr-2002	1.2	Document converted to ST format

For current information on PSD products, please consult our pages on the world wide web:

www.st.com/psm

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.psd@st.com (for application support)

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.