XILINX®

XAPP330 (v1.1) October 9, 2000

XPLA1 Programming Times

Summary

This document provides a description of the of JTAG (Joint Test Action Group) Programming structure of XPLA1 devices, and lists the programming times required to fully configure the 32 through 128 Macrocell XPLA1 CPLDs.

Introduction

Traditional configuration of Programmable Logic Devices (PLDs) usually included some sort of stand alone programming hardware. Historically, a manufacturing person would reside at a piece of programming equipment and place a blank device into a socket (or sockets, in the case of a gang programmer) and invoke some algorithm to begin the electrical configuration of the PLD. As is the case with any process (especially those involving a human interface) there are risks involved with this procedure. The operator must have the correct device, programming file, and algorithm, and be trained in the proper handling of sensitive electronic devices. Component damage due to mishandling and ESD burdened the overall cost of manufacture, and uniformity of the process could vary depending on whoever was programming at that time. Automated programming equipment became available to program loose components; this equipment was (and is) very costly. Manufacturers can contract programming houses to program their PLDs, but this still requires that the PLD be subjected to handling risk and places additional cost on system assembly.

With the introduction of In-System Programming (ISP), manufacturers found a way to decrease handling related damage and to expedite the manufacturing of their systems. "Blank" devices are assembled onto PCBs and later programmed via ISP. This allows for the removal of the risk of handling damage associated with the programming process, but still necessitated seperate steps to program the board and then test.

ATE systems have existed as "bed of nails" testers for quite some time. These systems are quite costly and require expensive mechanical custom interfaces to a PCB. They depend on direct contact to a PCB node, and can inflict damage to a board if mechanical misalignment occurs. As PC boards increased in size and density, the costs associated with setup and maintenance became large factors. As devices for PCB designs became larger and more complex, a new technology was introduced to facilitate cost effective and time effective board testing. Test circuitry was designed into each integrated circuit, which allowed for electrical vector tests to determine solder and assembly integrity (on the PCB), and also allows for some system operation tests. This is implemented using a JTAG port, which is the same port that is used for ISP operations.

Manufacturing engineers began to evaluate device programming capability on these platforms, and it was realized that the same equipment that was performing boundary scan and other test operations could also be used to perform device programming on the board at time of test. This removed the seperate programming step and provided manufacturers with a more integrated assembly, program, and test solution.

Operation time on these test systems is costly however, with tester time being billed out at $0.10 to $1.00 per minute of operation. With greater and greater focus on reducing manufacturing costs, time spent on these test fixtures has come under scrutiny. Companies are recognizing the need for devices that program and test quickly in order to acheive optimized return on manufacturing investment. For this reason it is crucial that programming times of

PLDs and other programmable elements be as short as possible. This document will detail the theoretical programming times of XPLA1 devices.

Programming Times

Programming times are calculated from the ISP specification for each component. The maximum TCK specification is listed as 10 MHz; for each component a 10 MHz programming time has been calculated, and an algorithm is provided for the end user to calculate programming times based upon other TCK frequencies. These calculations are valid for dedicated programming controllers (such as ATE), and do not take into account miscellaneous overhead delays such as interrupts or wait states that might be incurred by non-dedicated ISP host machines (such as PCs).

Basic XPLA ISP Programming Description

XPLA devices have memory storage in the form of EE arrays, which store the configuration of the programmed CPLD. For the XPLA1 devices, these arrays range in size from 21,930 bits (32-macrocell) to 88,408 bits (128-macrocell). It is important to note that not all of these bits are user configured; some of these cells are used exclusively by the manufacturer.

In all XPLA1 devices, data is partitioned in the EE array in blocks, rows and columns. There are two blocks of EE storage in XPLA1 devices. Column size will vary with device size, but the number of rows (43) remains consistent throughout the family. Only 41 of these columns are required to be programmed to fully configure an XPLA1 CPLD. The programming procedure (in its most basic description) is comprised of shifting in a column of data and seven address bits, and then programming the addressed column of the EE array. A programming pulse (delay) of 10 ms is required to program the EE cells. Since there are 41 columns being programmed into two blocks, a total program pulse delay of 820 ms is incurred. There is an additional 100 ms pause during the bulk erase (actually a 100 ms delay plus 103 TCKs), and a 100 µs configuration delay which sums to a fixed overhead total (for all XPLA1 ISP devices) of 920.1 ms.

The XPLA1 devices support 10 MHz JTAG operations, so the amount of time required to shift in the actual configuration data is very small compared to the amount of time required by the actual programming pulse delays. For this reason, there is only a slight difference in programming times between the 32-macrocell device and the 128-macrocell device.

Theoretical Times

The ISP programming of the three XPLA1 devices primarily differs in the number of bits shifted into the device during programming. Refer to Table 1 for the total amount of TCKs and the total time required for programming the different devices with a 10 MHz TCK.

The XPLA1 128 macrocell original (non-enhanced) device has one additional difference in the basic programming algorithm. This device supports boundary scan, and the programming specification details an additional 280 bits of data to be shifted into the device to place the BSC's into a benign state. At 10 MHz, this additional requirement only adds approximately 29 us to the total programming time. This 29 µs has been omitted from the programming time in Table 1.

Table 1: **TCK Requirement and 10 MHz Programming Times**

Device	Number of TCKs	Time to Program (sec) @ 10 MHz TCK
32 MC	21968	0.922
64 MC	43042	0.924
128 MC	85461	0.929

The programming times were calculated using the following formula:

Time = C x L + 0.920s

Where:

C = total number of TCKs

L = TCK clock period

0.920s = approximate erase, program, and initialization delay.

Actual Programming Time

The measured amount of programming time for a 128MC device being programmed by a 200 MHz Pentium PC (via the parallel port) is approximately 5.6 seconds. Version 4.05 of XPLA ISP was used to perform this test. Individual results will be machine speed dependant.

Conclusion

The evolution of the manufacturing process now includes programming of configurable devices at time of test using Automated Test Equipment. Because of the expenses associated with the use of this equipment, designers must now also consider the time spent by manufacturing to program a configurable device as part of the overall system assembly burden. Device programming times vary from manufacturer to manufacturer (and even family to family) so intimate knowledge of architecture and device technology is critical to the person who must succeed in today's "design to cost" oriented environment.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
02/18/00	1.0	Initial release.
10/09/00	1.1	Added Discontinuation Notice.