C2000™ “GANG” Programming

Flash Programming Multiple F28x Targets in a Single Operation

by Andrew Ferrari, EWA Technologies, Inc.

Overview

Programmable devices are growing in number every year, finding their way into more and more consumer products. Types of products that, in the past, have implemented application-specific integrated circuits (ASIC), are now being designed with reprogrammable cores. This is happening in part because of cost - the high NRE cost of producing the ASIC component is avoided, allowing affordable devices to be produced in smaller volumes - and flexibility - these new designs can be programmed or reprogrammed in development, in production and in the field (for servicing and upgrades).

Production programming is the process in which blank programmable devices are populated in production with their executable code. Gang programming is programming of more than one blank device using a single set of operations from one programming source. Gang programming is expected to be faster than programming single devices, so is typically associated with the production process.

This article is focused on gang programming of smaller production quantity F28x-based devices from Texas Instruments (TI).

The C2000 On-Chip Flash API

Before we get started on gang programming devices, let’s go over the basics of programming individual TI F28x controllers with on-chip flash. We’ll be covering the IEEE® 1149.1 JTAG interface for programming in this article. A serial RS232 is also available on these devices, but is not in the scope of this article.

The C2000 Flash API from TI exists in two parts. The first part is a program resident on the host machine (PC) that downloads the second part, the executable code running on the F28x CPU, to clear, program and lock the flash memory. The PC then sends data to the executable code on the CPU to program the flash memory. The two parts communicate via controller over the target JTAG interface to complete the task. Figure 1 illustrates that process. More details on the TI C2000 Flash API can be found on the TI web site at www.ti.com/c2000flashtools.

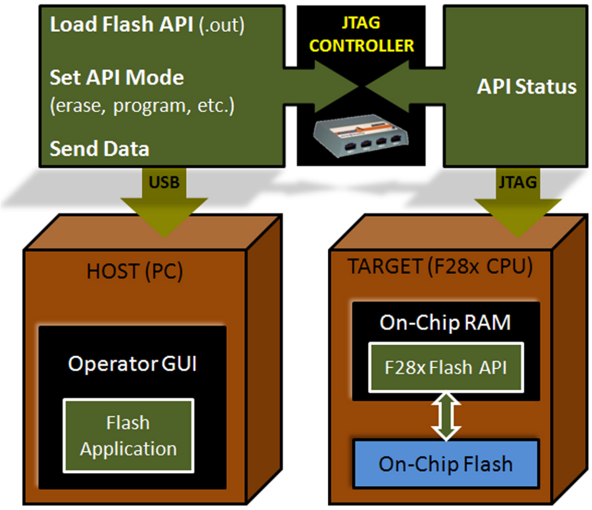


Figure 1 – C2000™ Flash API Implementation

Operator Interface

Common Integrated Development Environments (IDEs) include relatively sophisticated and complex user interfaces to empower source code development, compilation, testing and debug.

However, once development is completed, the job of production programming is commonly transferred to a straight-forward, easy-to-maintain operator station similar to that shown in Figure 2. The goal of the operator station is to perform the programming tasks on multiple boards as quickly as possible, while still maintaining high quality and control during execution.

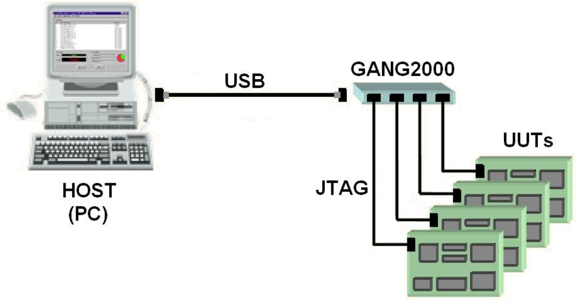
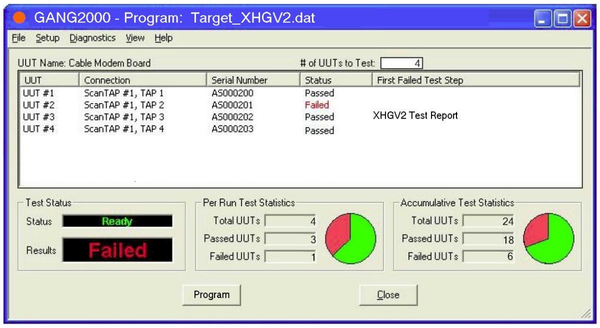


Figure 2 – Operator Station Layout

Clear, uncomplicated, GO and NO-GO indicators of programming status, as shown in Figure 3, are key to increased throughput and quality.


GANG2000 - Program: Target_XHGV2.dat
File Setup Diagnostics View Help
Program
Close

UUT Name: Cable Modem Board

# of UUTs to Test: 4

UUT	Connection	Serial Number	Status	First Failed Test Step
UUT #1	ScanTAP #1, TAP 1	AS000200	Passed	
UUT #2	ScanTAP #1, TAP 2	AS000201	Failed	
UUT #3	ScanTAP #1, TAP 3	AS000202	Passed	XHGV2 Test Report
UUT #4	ScanTAP #1, TAP 4	AS000203	Passed	

Test Status

Status: Ready

Results: Failed

Per Run Test Statistics

Total UUTs: 4

Passed UUTs: 3

Failed UUTs: 1

Accumulative Test Statistics

Total UUTs: 24

Passed UUTs: 18

Failed UUTs: 6

Figure 3 – Sample Operator Interface

In this environment, problematic UUTs (units under test) can be quickly identified and easily sorted into the *defective* bin for more comprehensive testing or analysis later. This allows more UUTs that pass the programming phase to make their way rapidly down the production line to assembly and packaging.

Multiple Target Considerations

So far we have discussed programmable devices, the C2000 flash API and the operator interface. Now I'd like to cover some of the lower level details related to programming multiple targets.

First, you want to be able to easily connect and disconnect the programming probes or cables from the controller's test access ports (TAP) to the UUTs. This connection should be long enough to reach each UUT's programming header and be able to sustain high usage without failure. As well, it must be easily replaced if damaged or worn out.

When more than one UUT is connected, JTAG signal distribution becomes important. Each UUT's scan chain (JTAG interface) requires several signals: test clock (TCK), Test Reset (nTRST), Target Mode Select (TMS) and Test Data Input and Output (TDI/TDO). The controller will need to manage each TAP's scan chain. This is typically accomplished using an FPGA or CPLD.

Voltage levels need attention. Today's devices' power requirements can range from 5 volts less than 1.0 volt, so the controller hardware needs to meet the levels required by each UUT. This can become critical when multiple UUTs are attached at different IO voltages.

UUTs implementing multiple programmable devices mean that device types and scan chain configuration need to be factored into the programming equation. The controller and application software must know not only what devices are connected to each TAP, but also which need to be programmed and which bypassed.

Finally, depending on the implementation, the connected devices can be programmed serially (one after another) or concurrently (at the same time). Concurrent methods, such as broadcasting the same command simultaneously to each device, will yield the fastest throughput. However, failure of one UUT will stop the entire process.

The next section discusses a current solution in which all of these issues are resolved.

A Gang Solution

The Blackhawk Gang2000 Programmer (Gang2000), Figure 4, is designed to program multiple F28x devices with on-chip flash via JTAG using the TI C2000 Flash API described earlier.

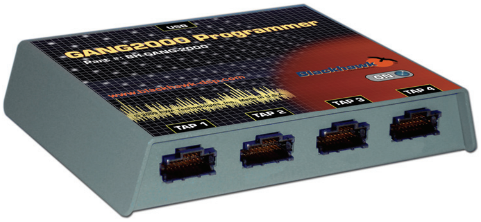


Figure 4 – Gang2000 Programmer

The Gang2000 interfaces to the host PC via high-speed USB 2.0 and includes support for up to four UUT connections. This means that the Gang2000 can program four F28x-based target boards in one step using the operator software graphical user interface (GUI).

To cope efficiently with UUT failures during programming, the Gang2000 balances speed with effective error detection by controlling the TI Flash API so that each device has executable code running independently. For example, each device can be given the erase command serially, but the actual operation is completed in parallel (concurrently).

Beyond On-Chip Flash Support

The Gang2000 meets programming needs for low-to-mid level production volumes at an affordable price, but what if you need more than just on-chip flash support? Maybe you want to perform interconnect testing using boundary scan or perform in-system programming of FPGAs and CPLDs. Well, we have that covered. You can easily step up from the Gang2000 to ScanExpress, a complete suite of boundary scan tools by Corelis, leaders in boundary scan www.corelis.com.