



LogicLoader Configuration Block Usage

Application Note 339

Hans Rempel

Logic Product Development

Published: April 2006

Abstract

This document presents information on usage of the LogicLoader configuration block, both within and outside of the LogicLoader operating environment.

This file contains source code, ideas, techniques, and information (the Information) which are Proprietary and Confidential Information of Logic Product Development, Inc. This information may not be used by or disclosed to any third party except under written license, and shall be subject to the limitations prescribed under license.

No warranties of any nature are extended by this document. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipments. The only warranties made by Logic Product Development, if any, with respect to the products described in this document are set forth in such license or agreement. Logic Product Development cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

Logic Product Development may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except as expressly provided in any written agreement from Logic Product Development, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

© Copyright 2006, Logic Product Development, Inc. All Rights Reserved.

REVISION HISTORY

REV	EDITOR	DESCRIPTION	LOGICLOADER VER.	APPROVAL	DATE
1	Hans Rempel	Preliminary	2.2.0	HAR	4/25/06
A	Jed Anderson	General editing and formatting for release; Added PN and App Note number	2.2.0	JCA	7/27/06

1 Introduction

LogicLoader™ version 2.0 and later supports the idea of an optional configuration block of flash memory on most Logic Product Development board-level products. This optional configuration block of memory can be used by LogicLoader to set default behavior of the system when in the LogicLoader environment (such as, default UART baud rate, start up scripts, etc.).

Customers sometimes require the ability to modify the configuration block data from an environment other than LogicLoader. This application note is being written so that customers understand how the configuration block is structured to enable them to write custom code that can safely modify, add, or delete data in the configuration block from software environments other than LogicLoader.

2 Reference Material

Please refer to the following documentation (available on Logic's download site—<http://www.logicpd.com/auth/>) as necessary, for more information on the terms, ideas, and functionality presented herein:

- *LogicLoader User's Manual*—provides an overview of LogicLoader's functionality and capabilities
- *LogicLoader Command Description Manual*—provides details on each command that is available from within the LogicLoader environment
- *[SOM Family]_LogicLoader_User's_Manual_Addendum*—provides specific System on Module (SOM) LogicLoader memory map and supported features information

3 Configuration Block Overview

3.1 Initialization

The use of the configuration block feature in LogicLoader is optional—it is only activated when the user specifically instructs LogicLoader to initialize a new configuration block with the 'config CREATE' command or uses the 'config C' command to load a valid configuration block. Please reference the *LogicLoader User's Manual* "Configuration Block" section for more information on creating and using a configuration block.

3.2 Size and Allocation

The configuration block memory section allocated in LogicLoader 2.0 and later is 64kBytes in size, and is typically located in flash memory directly following the allocated LogicLoader program storage flash blocks. The actual size of the configuration block data varies from engine to engine but is always guaranteed to be less than or equal to 64kBytes. On some board-level products, the available flash block(s) that contain the configuration block exceed 64kBytes of memory, leaving the remaining flash memory above 64kBytes unused. For example, the LH7A404-11 Card Engine product has 256kBytes of flash memory per block, so the upper 192kBytes are unused by the configuration block section.

3.3 Usage

The 64kBytes of configuration block memory is allocated to store information that defines default peripheral functionality and to store non-volatile scripts. LogicLoader manages all information in the configuration block through command-line commands that allow the user to add user data or change specific parameters in the configuration block. When data in the configuration block is

changed, LogicLoader reads the entire flash block into RAM (configuration section plus the unused section), modifies the necessary data in the configuration block, updates the applicable checksums, erases the flash block, and then writes the entire flash block back to flash memory. Any data in the unused section is preserved.

4 Configuration Block Structure

4.1 Entity Locations

Each item in the configuration block has its own section, and each section has its own checksum. Additionally, the entire configuration block has an associated checksum.

An example configuration block diagram with section and checksum positions is detailed below. Please see the *LogicLoader User's Manual Addendum* for specific information on addresses and valid sections for the board-level product in use.

LH7A404-11 Configuration Block Layout

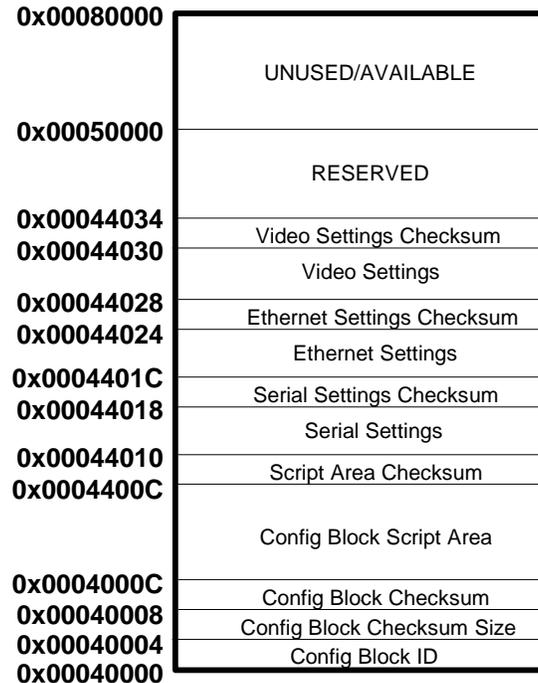


Figure 4.1: Example Configuration Block Diagram

As the figure above shows, every data section has its own checksum, and the entire “Configuration Block Checksum Size” is used to create the “Config Block Checksum.” Any part of the “Unused/Available” area may be used without destroying information in the configuration block as long as the application that is writing there preserves the first 64kBytes of flash—just as the configuration block write routines preserve the “Unused/Available” area on architectures where that area exists.

4.2 Checksum Calculation

Each checksum in the configuration block is created as an 8-bit checksum stored in a 32-bit location. The code snippet below shows how the checksums are calculated.

```
static unsigned int
dev_config_checksum(char *address, unsigned int len)
{
    u_int d_checksum = 0;
    int i;
    /* calculate & verify checksum */
    for (i = 0; i < len; i++)
        d_checksum += address[i];
    return d_checksum;
}
```

5 Sharing the Configuration Block with an Operating System

When using the configuration block in LogicLoader to store scripts, settings, or information, any access by a different software environment must take care to maintain the format of the configuration block, or risk losing data. The following steps outline the process that should be followed when adding, deleting, or modifying information in the configuration block from a non-LogicLoader environment.

- 1) Read entire contents of the configuration block into a RAM buffer
- 2) Modify/add/delete data in the RAM buffer
- 3) Calculate checksums as necessary and update in the RAM buffer
- 4) Erase the block of flash the configuration block resides in
- 5) Write the updated contents of the RAM buffer back to the block(s) of flash memory used for the configuration block

6 Summary

The flash block containing the optional 64kByte (or less) configuration block can be used to store custom data if care is taken not to corrupt existing information in the configuration block by paying attention to how data and checksums are updated in the flash memory.