Implementing the MicroBlaster Configuration on the ColdFire Development Board

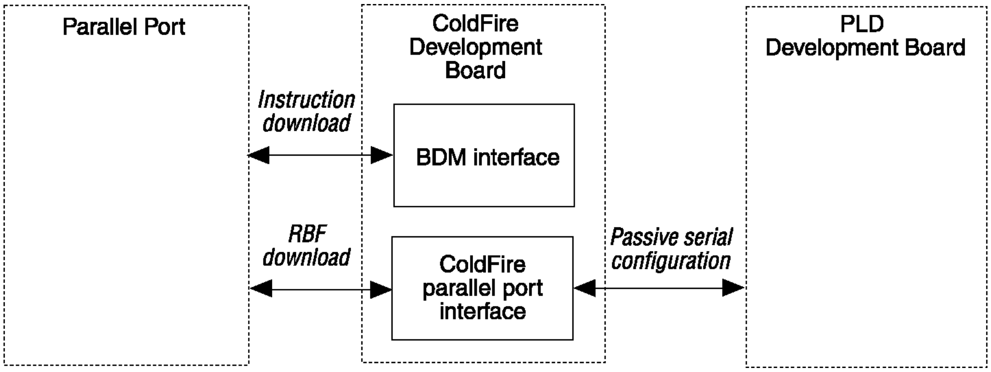
Introduction

The MicroBlaster software driver configures Altera® SRAM-based programmable logic devices (PLDs) in passive serial mode and targets embedded configurations. The source code is customized and implemented on a ColdFire development board to accomplish the embedded passive serial configuration. This document explains the customization and the implementation of the MicroBlaster on the ColdFire development board.

Implementation Block Diagram

Figure 1 shows the block diagram of the implementation of the MicroBlaster configuration on the ColdFire development board.

Figure 1. MicroBlaster Configuration on the ColdFire Development Board Block Diagram


Parallel Port
ColdFire Development Board
PLD Development Board
Instruction download
BDM interface
RBF download
ColdFire parallel port interface
Passive serial configuration

The three main blocks in the implementation process are the parallel port, the ColdFire development board, and the PLD test board.

The parallel port is the common printer port available on PCs. The ColdFire development board is a Motorola M5206EC3 ColdFire Matrix Design Evaluation System with 4Mbyte of ADRAM and supports background debug mode (BDM). The PLD test board is an Altera APEXE 20K200E PLD test board. The ColdFire development board block contains the BDM and ColdFire parallel port interface modules.

These main blocks are interconnected through three interfaces, which are involved at different stages of the implementation process. The three interfaces are the instruction download, the raw binary file (RBF) download, and the passive serial configuration. The role of each interface is explained in Table 1.

Table 1. MicroBlaster Configuration Interfaces

Interface	Role
Instruction download	Interface between the parallel port and the BDM module. Its main role is to download the instructions to the ColdFire board for execution by the ColdFire processor.
RBF download	Interface between the PC parallel port and the ColdFire parallel port (general purpose I/O). Two pins carry out the RBF download operation. The RBF file is downloaded from the PC and processed before being written to the on-board ADRAM
Passive serial configuration	Interface between the ColdFire parallel port and the PLD test board. Once the RBF download operation is successful, the ColdFire processor can instantiate the passive serial configuration for the PLD test board.

Implementation Process

The implementation of the MicroBlaster software driver on the ColdFire development board consists of three stages: instruction download, RBF download, and configuration. Table 2 summarizes these implementation stages.

Table 2. MicroBlaster Configuration Implementation Stages Summary

Stage	Objective
Instruction download	To download the instruction binary from the PC to the ColdFire development board. Once completed, the ColdFire processor starts to execute the instructions. The process then moves into the RBF download stage.
RBF download	To download the RBF bit stream into the ADRAM for PLD configuration. The ColdFire processor starts to read from its parallel port. For each bit read, the bit is processed and then written to the ADRAM. Once completed, the process moves into the configuration stage.
Configuration	To configure the APEXE 20K200E device on the PLD test board.

Instruction Download

The original and the customized MicroBlaster source code are written in the C language. The source code is compiled and linked using the Metrowerks CodeWarrior software. The binary file is downloaded into the ColdFire development board using the software through the P&E BDM Interface Cable (CABLECF). Once the instruction download is complete, the ColdFire processor starts executing the instructions and keeps reading its parallel port for data. This moves the implementation process into the RBF download stage.

RBF Download

While the ColdFire processor is reading at its parallel port, the PC starts to dump the contents of the RBF bit by bit to the parallel port. This process uses the `DATA` and `CLOCK` signals. Each bit dumped is associated with a bit `0` and a bit `1` at the `CLOCK` signal. This association gives a positive-edge-trigger signal to the ColdFire processor to indicate a valid bit at the `DATA` signal. Before dumping the RBF bit stream, the PC sends a 32-bit bit-stream indicating the size of the RBF file. Then the RBF download starts.

When catching a positive-edge-trigger signal at the `CLOCK`, the value of the `DATA` signal is processed and written to the on-board ADRAM. The bit data is processed by putting it into the appropriate parallel port register (`PP0-PP7`) before being written to the ADRAM. As shown in Figure 2, `PP5` holds the data to be dumped to the `DATA0` signal of the PLD development board during configuration. Therefore, the bit data read from the `PP2` (`DATA` signal from the PC parallel port) is shifted to PP5. Bit PP3 (mapped to the `DCLK` signal) is set to `0`. Then the byte (`PP0-PP7`) is written to the ADRAM. An example is illustrated below:

DATA Signal (PC Parallel Port)

```
    1
```

Parallel Port (ColdFire)

```
    PP0 PP1 PP2 PP3 PP4 PP5 PP6 PP7
     X   X   1   X   X   X   X   X
```

Parallel Port (ColdFire) After Rearrangement

```
    PP0 PP1 PP2 PP3 PP4 PP5 PP6 PP7
     X   X   X   0   X   1   X   X
```

This process is repeated until every bit in the RBF is downloaded. Once this stage ends, the process goes to the configuration stage. Figure 2 shows the pin mapping of the `DATA` and `CLOCK` signals to the ColdFire parallel port.

Configuration

The PLD configuration is in passive serial mode and involves the `DCLK`, `DATA0`, `NCONFIG`, `NSTATUS`, and `CONF_DONE` configuration signals. These signals map to five ColdFire parallel port pins. Of these pins, three are outputs and two are inputs with reference to the ColdFire development board. Figure 2 shows the pin mapping of the configuration signals to the ColdFire parallel port.

When the PLD configuration is instantiated, the first byte is loaded from the on-board ADRAM and then dumped to the ColdFire parallel port. The byte is then bit-wise OR'd with `0x08` to set the PP3 pin (mapped to the `DCLK` signal) to `1`. The modified byte is then dumped to the ColdFire parallel port. This process is to produce a positive transition (from `0` to `1`) for every `DATA0` bit sent to the PLD development board.

This process, illustrated in the following example, can speed up the configuration time but requires more memory space. Only the `DATA0` and `DCLK` signals are shown.

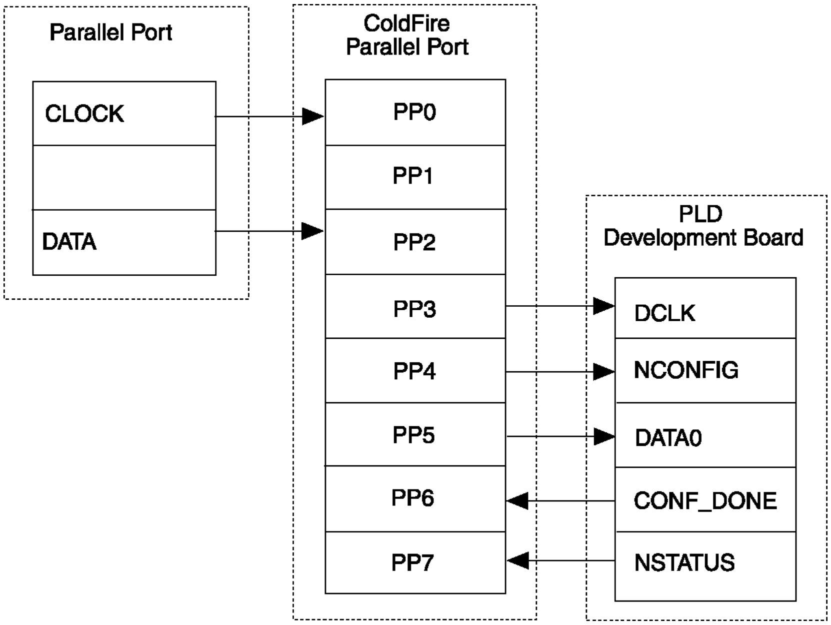
Bytes loaded from the ADRAM

Byte 1: `XX0X1XXX`
Byte 2: `XX0X0XXX`
Byte 3: `XX0X1XXX`
…

Sequential dumps as observed on the ColdFire Parallel Port

Dump 1: `XX0X1XXX`
Dump 2: `XX1X1XXX`
Dump 3: `XX0X0XXX`
Dump 4: `XX1X0XXX`
Dump 5: `XX0X1XXX`
Dump 6: `XX1X1XXX`
…

Figure 2. Mapping of DATA, CLOCK &Configuration Signals



Conclusion

The MicroBlaster is a software driver for the embedded passive serial configuration. You can easily customize it to fit in different embedded systems. The customized MicroBlaster driver allows fast and easy configuration and requires no extra boards or components between the embedded system and the PLD test board. The source code development or customization cycle is short. All of these have made MicroBlaster an embedded solution to the passive serial configuration.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com