Technical Note

Executing Code Faster with the eZ80F91 MCU's Copy to RAM Feature

TN003401–0804

Introduction

This Technical Note discusses how eZ80F91 MCU code is executed faster by running the code from RAM memory instead of Flash memory, while at the same time being able to store the code in Flash. With ZiLOG's feature-rich ZDSII code development tools, speedy code execution is easily performed by making a few simple changes to the project settings. The `COPY2RAM` project, available in the `TN0034-SC01.zip` source code file accompanying this Technical Note, is used as an example to highlight the difference in execution speeds when the code is executed from Flash and RAM, respectively.

Copying eZ80F91 Code from Flash to RAM Using ZDSII

Working with the eZ80F91 device, you arrive at a stage of development in which you have debugged your code in RAM using the ZDSII development tool, experimented with the code in Flash, and found that it does not run fast enough. The solution to this problem is to copy the code from Flash to RAM and execute it from RAM memory, which is faster than Flash.

In this document, you will learn how to copy the code from Flash to RAM using the ZDSII development tool. In the ZDSII development environment, there are project settings that allow the user to select the type of configuration that the code stores and executes. These configurations include the *Standard* configuration, the *All RAM* configuration, the *Custom* configuration, and the *Copy to RAM* configuration. For details about these four configurations, please refer to the ZiLOG Developer Studio II—eZ80Acclaim!™ User Manual (UM0144) and the technical article titled How Code and Data are Placed in Memory Using ZDSII (TA0004).

The objective of the Copy to RAM configuration is to:

- Program Flash memory with a HEX file containing application code and data
- Direct the application program on startup to copy all of the code except start-up from Flash into RAM
- Continue execution from RAM

The result is an increase in code execution speed.

A flowchart detailing the Copy to RAM function is illustrated in Figure 1.

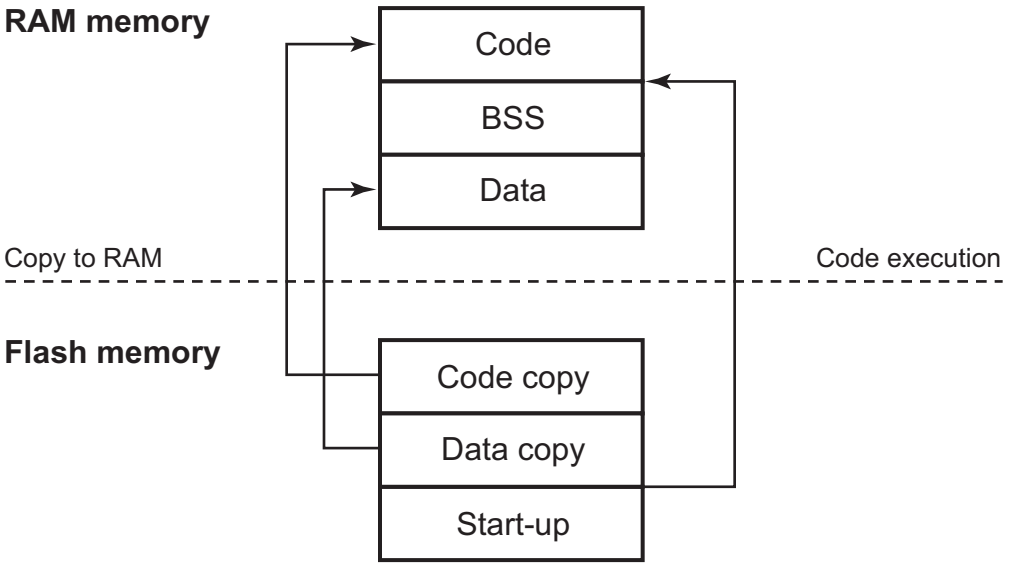

Figure 1. Execution Flow of Copy to RAM

Figure 1 depicts the start-up code, the Data Copy, and the Code Copy sections that are programmed into Flash.

Flash memory can be programmed using the ZDSII internal Flash Loader by selecting the appropriate Flash setting (internal, internal and external, or external) according to the memory usage. After Flash memory is programmed, pushing the Reset button starts the execution of the code. The start-up code performs the following sequence of operations.

1. Initializes the required processor registers.
2. Copies the Data Copy section in Flash to the Data section in RAM.
3. Creates the BSS section in RAM.
4. Copies the Code Copy section in Flash to the Code section in RAM.
5. The program then starts executing from the first line of code in the code section.

Note: The start-up code is not copied into RAM; only the code section in the main portion of the code that runs after startup is copied.

Using Copy To RAM Feature of the eZ80F91 MCU

The `COPY2RAM` project, available in the `TN0034-SC01.zip` source code file that accompanies this Technical Note, is an example that demonstrates the Copy to RAM feature. The `COPY2RAM` project, a simple project to generate a square wave at Port B, pin 0 (PB0) of the eZ80F91 Development Kit (ZiLOG part number Z80F910100KIT), highlights the difference in execution speed for code execution from Flash and RAM, respectively.

The project settings of ZiLOG's ZDSII for eZ80Acclaim!™ IDE are modified such that the code can be copied from Flash to RAM and can be executed from RAM. The ZDSII build generates a HEX file that can be programmed into Flash. The `COPY2RAM` project uses the standard configuration and runs with a standard start-up.

Table 1 lists the steps involved to make a new project that copies the code section from Flash and executes it from RAM.

Table 1. Copying Code from Flash and Executing in RAM

Step #	Step	Comments
1.	Launch ZDS II_eZ80Acclaim!_4.8.0.	Ensure that you use the latest version of ZDSII for eZ80®, or a more recent version.
2.	Open the `COPY2RAM.pro` project.	The source code file, `TN0034-SC01.zip`, accompanies this Technical Note.
3.	Flash the project. Follow the instructions provided in steps 8–11 to flash the project.	Observe the output at Port B, pin 0 of the eZ80F91 MCU, using CRO. This observation indicates the difference in program execution, with and without the Copy to RAM feature.
4.	Click the **Linker** tab in the **Project Settings** dialog box.	Figure 2 illustrates the **Linker** pop-up window.
5.	Ensure that sufficient RAM is available for storing the complete code so that the code can be executed from RAM.	Ensure that the RAM settings in the **Target** tab of the **Project Settings** dialog box are set appropriately.
6.	In the **General** category of the **Linker** tab in the **Project Settings** dialog box, click **Link Configuration** and select **Copy to RAM**. Click **OK**.	This step sets the configuration to **Copy to RAM** and places the Copy to RAM linker commands into the Link Control file. Figure 2 illustrates the **Link Configuration** pop-up window.
7.	Click **OK** in the **Project Settings** dialog box. A pop-up dialog window is displayed, which prompts the user with: `If you want to rebuild the files, click Yes.` Click **Yes**.	A Link Control file named `COPY2RAM` is created. This file builds the project and also generates the `COPY2RAM.hex` file which is programmed into Flash memory using the ZDSII internal Flash Loader.
8.	To program Flash memory using the ZDSII internal Flash Loader, select the **ZDSII Flash Loader** command from the **Tools** menu to display the **Flash Loader Processor** dialog box.	It is assumed that the eZ80® Development Platform is set up and the `COPY2RAM` project has been successfully demonstrated.
9.	Using the browser in the **File** menu, find and select the `COPY2RAM.hex` file.	
10.	Click **Burn** and **Verify** in the **Flash Controls** pane.	Flash memory is now programmed with the code that includes the Copy to RAM feature.

Table 1. Copying Code from Flash and Executing in RAM

Step #	Step	Comments
11.	After Flash memory is programmed, push the **Reset** button on the eZ80® Development Platform.	Notice that the output waveform generated by the CRO and measured at Port B pin 0 of the eZ80F91 device is different from that generated when the Copy to RAM feature is not used.

Figures 2 and 3 illustrate the linker setting and the linker configuration settings used to copy the code from Flash to RAM.

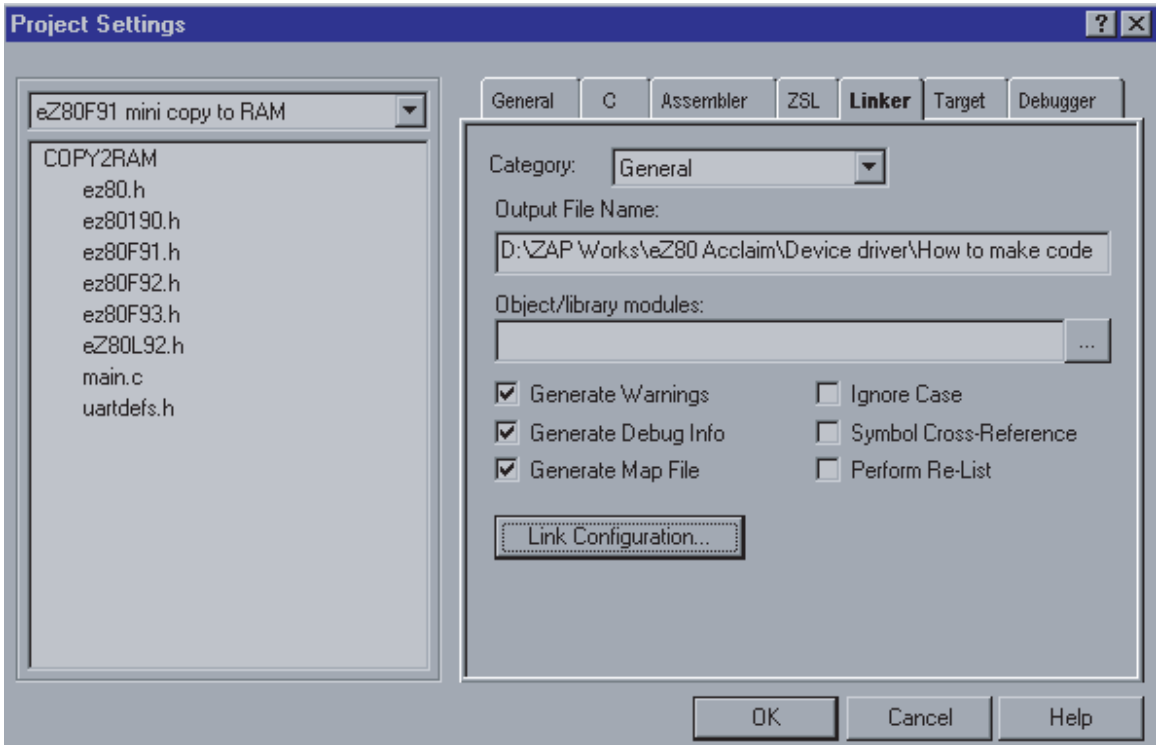

Figure 2. Linker Settings for Copying Flash into RAM

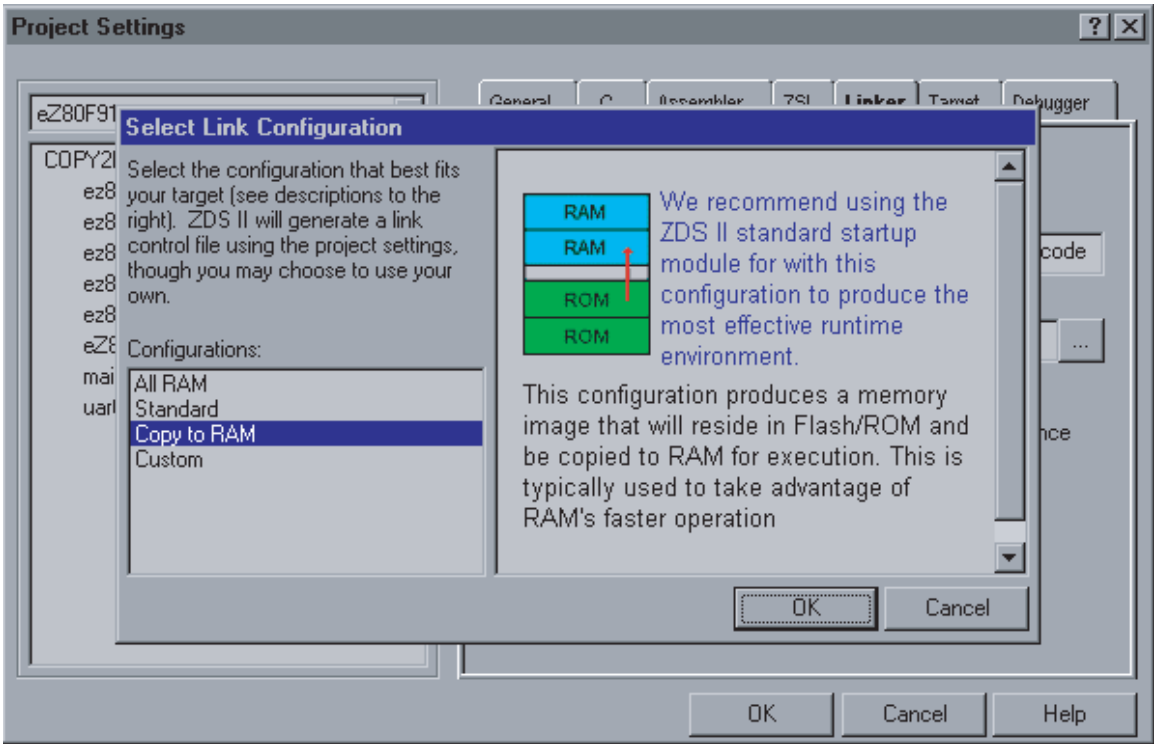

Figure 3. Linker Configuration Settings for Copying Flash into RAM

The waveform diagrams shown in Figures 4 and 5 illustrate the differences in execution time with and without the Copy to RAM feature. Figure 4 shows the execution time using the linker's standard setting. In this instance, the code executes from Flash memory.

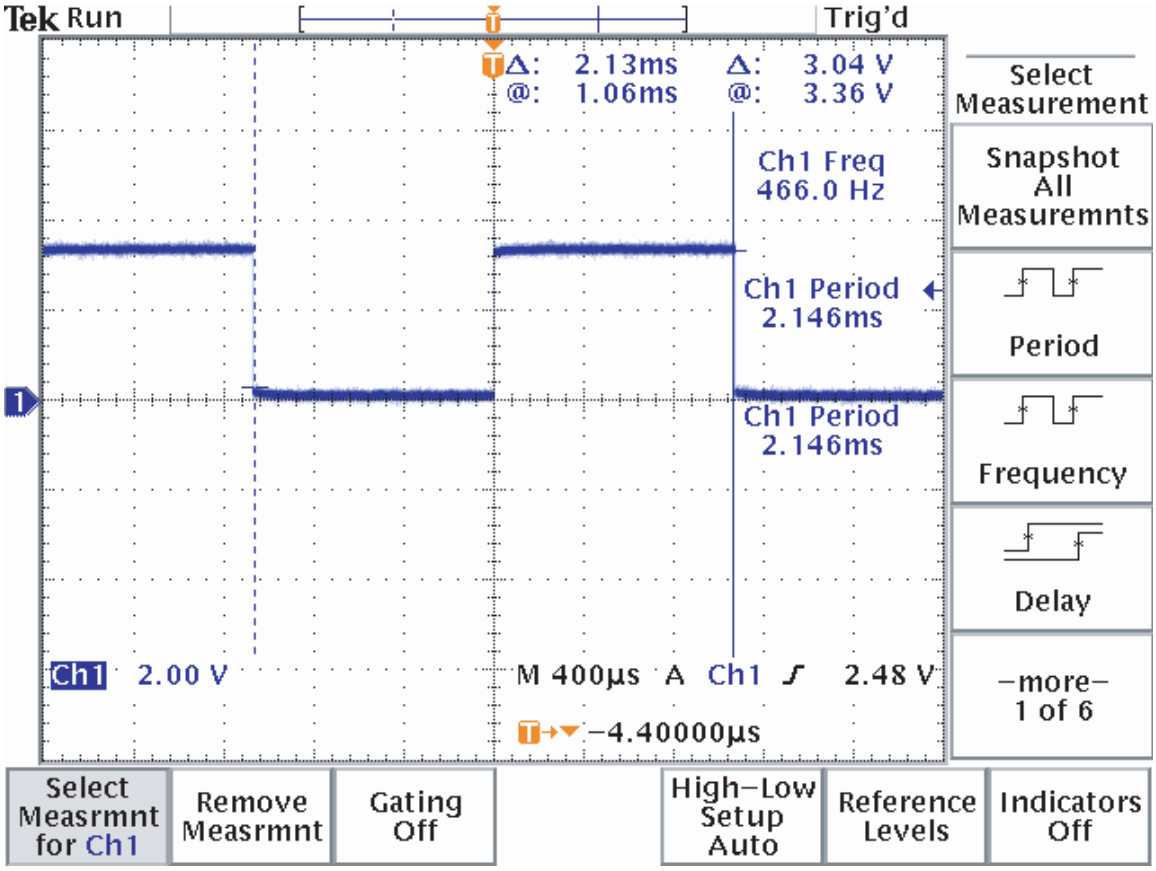

Figure 4. Output Waveform with the Standard Linker Setting

Figure 5 shows the execution time using the linker's Copy to RAM setting. In this instance, after resetting the linker, the code is copied into RAM from Flash and runs from RAM. A comparison between Figures 4 and 5 indicates that code execution speed with the Copy to RAM functionality is nearly two times greater, compared to that without the Copy to RAM functionality.

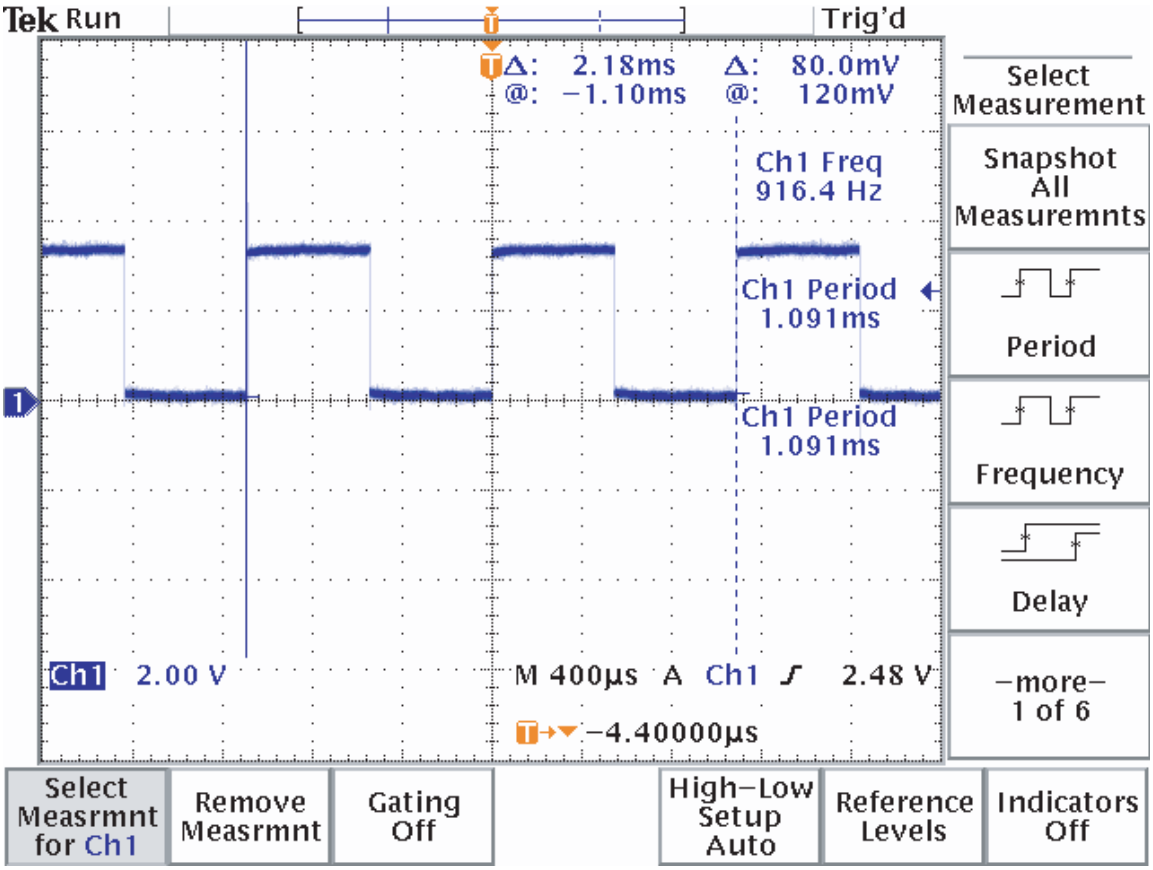

Figure 5. Output Waveform with the Copy to RAM Linker Setting

Summary

This Technical Note discusses how code execution speed increases when code is executed from RAM instead of Flash using the Copy to RAM feature available with ZDSII. The difference in the speed of code execution is determined using a simple code example to generate two waveforms, each representing the execution of code from Flash and RAM, respectively.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.zilog.com

Information Integrity

The information contained within this document has been verified according to the general principles of electrical and mechanical engineering. Any applicable source code illustrated in the document was either written by an authorized ZiLOG employee or licensed consultant. Permission to use these codes in any form, besides the intended application, must be approved through a license agreement between both parties. ZiLOG will not be responsible for any code(s) used beyond the intended application. Contact the local ZiLOG Sales Office to obtain necessary license agreements.

Document Disclaimer