technische universität dortmund

computer science 12

Bachelor/Master Thesis

Reverse Engineering timing properties of TIs MSP430 FRAM Devices

Nils Hölscher
Prof. Dr. Jian-Jia Chen

Otto-Hahn Str. 16
Technische Universität Dortmund
Email: nils.hoelscher@udo.edu
February 7, 2025

Byte-addressable non-volatile memories (NVMs) recently have emerged to serve as main memory due to the features of low leakage power, high density, and low unit costs. However, only a few NVM technologies are mature and publicly available at the moment. The limited availability of NVMs leads to theoretical models not being validated on real hardware in most cases, especially timing models. Since Ferroelectric RAM (FRAM) [**?**] is a mature technology and is publicly available on the market, it is actually possible to create a timing model on a real device.

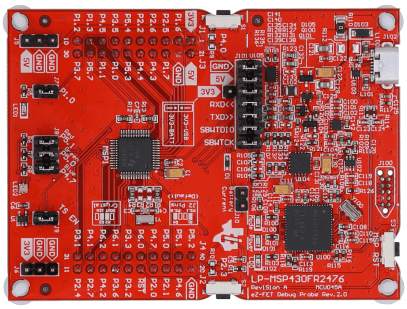

Figure 1: TI MSP430 FR2476 Launch Pad with FRAM and MSP430 CPUX.

The access latency of an MCU, like the MSP430 is typically influenced by two major factors: 1) The access latency of a single instruction in the CPU. 2) The timing latencies of the memory device, including read cache, and the memory bus connected to the CPU. We are currently developing a timing model for the MSP430 FR2xx and FR4xx families, based on TIs Documentation [1]. However, manufacturers documentations tend to be inaccurate or even wrong [2], which still needs to be evaluated and enriched with concrete experiments. For example, the replacement policy of the MSP430s FRAM controllers read cache is not stated in TIs documentation.

The best way to measure latencies is to use microbenchmarks, as Abel et al. [2] used them. Microbenchmarks execute many instructions in succession and measure the needed cycles, more complex methods may not be necessary, due to the MSP430s simple MCUs. However, the MSP430 does not have a Performance Measurement Unit (PMU) and the MSP430s clock has to be set, to count MCU cycles. Also, cache miss and hit counters are missing to reverse engineer the FRAM controllers read cache.

In this thesis,[1] students first should get familiar with the MSP430 microcontroller family, especially with the MSP430 FR5994 model. After being able to run simple programs on the real hardware, students should study the device specific Users Guide [1] and implement a cycle counting solution. With this the first latencies of single instructions can be measured with microbenchmarks. This should be automated with a script, that repeatedly flashes the devices and processes the MSP430s serial output. In the second part of this thesis a method to count hits and misses on the FRAM controller read cache has to be developed. Enabling the reverse engineering of the replacement policy, similar to Abel et al. [2].

Required Skills:

- Knowledgeable of C and C++ programming
- Understanding of embedded tool chains
- Willing to program microcontrollers

Acquired Skills after the thesis:

- Knowledge about modern memory technologies and access latency models.
- Knowledge about MSP430 microcontrollers and the integrated FRAM controller.
- Knowledge about reverse engineering.

References:

[1] Texas Instruments "MSP430FR4xx and MSP430FR2xx family Users Guide" Section 4.5.1.5 and 6.1

[2] Abel and Reineke "nanoBench: A Low-Overhead Tool for Running Microbenchmarks on x86 Systems"

[1] Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.