GDB-Based Fault Injection on Raspberry Pi Pico

Donovan Perry

North Carolina Agricultural and Technical State University

Introduction & Motivation

- Fault Injection
  - Research technique to assess a device's resilience and vulnerability
  - Helps us understand how these systems react to unexpected faults/errors
- Raspberry Pi Pico
  - Used in critical applications
  - Ensuring it's reliability is vital
  - Lead to improved system robustness

Defining the Problem and Objectives

- Problem?
  - Evaluate Raspberry Pi Pico's unexpected faults or errors
  - Helps us understand how these systems react to unexpected faults/errors
- Objective
  - Conduct GDB-based fault injection testing
  - Assess the device's behavior
  - Identify potential vulnerabilities

Related Work

- Two Approaches
  - Hardware Implemented Fault Injection
  - Software Implemented Fault Injection
- Experiment Details
  - Control Software programmed to inject faults using breakpoint trigger
  - GDB assures easy access to properties
    - Approach can be easily adopted for many different platforms (Pico)

Fault injection in embedded systems using GNU Debugger

Mgr inż. Michał MOSDORF

PHD student at Institute of Computer Science of Faculty of Electronics and Information Technology. Graduate of Computer Science at Faculty of Electronics and Information Technology (2009). Conducts research in field of software reliability in embedded systems environment.

e-mail: m.mosdorf@ii.pw.edu.pl

Prof. dr hab. inż. Janusz SOSNOWSKI

Graduated from the Faculty of Electronics and Information Technology at Warsaw University of Technology. Received professor title in 2006. Currently employed in professor position at Computer Science Institute of Warsaw University of Technology. He is author and coauthor of 200 publications. His scientific interests concern computer systems reliability, architecture and interfaces design.

e-mail: jss@ii.pw.edu.pl

Abstract

The paper presents the technique of simulating faults in embedded systems. It consists of PC software that performs fault injection through the JTAG interface controlled by GNU Debugger (GDB) server for a chosen platform. This approach can be easily adopted to various platforms due to a wide support of GDB project for many architectures. The experimental results for ARM architecture show high controllability of the fault injection process and measured time overhead in the implemented injector.

Keywords: fault injection, JTAG interface, GDB project, microcontrollers.

Symulacja błędów w systemach wbudowanych z wykorzystaniem GDB

Streszczenie

Praca przedstawia technikę symulacji błędów dla systemów wbudowanych wykorzystującą interfejs JTAG sterowany za pomocą oprogramowania „GNU Debugger" przygotowanego dla danej platformy mikroprocesorowej. Opracowana architektura symulatora błędów została przedstawiona na rys. 1. Zaprezentowane rozwiązanie umożliwia symulację błędów typu bit-flip oraz błędów trwałych za pomocą mechanizmów breakpoint oraz watchpoint. Obserwacja wyników symulacji została zrealizowana za pomocą programowego mechanizmu breakpoint. W ramach pracy zweryfikowano koncepcję dla współczesnych mikroprocesorów z rdzeniem ARM7TDMI oraz zaprezentowano rezultaty symulacji błędów dla wybranych obszarów pamięci SRAM oraz rejestru PC procesora. Podejście to może być łatwo dostosowane do różnych platform systemów wbudowanych wspieranych przez projekt GDB. Przeprowadzone eksperymenty symulacyjne potwierdziły ich dużą sterowalność. W pracy przedyskutowano również efektywność opracowanej metody symulacji błędów oraz przedstawiono wyniki pomiarów opóźnień związanych z symulacją błędów oraz obserwacją wykonywania programu wynoszące odpowiednio 52ms i 42ms.

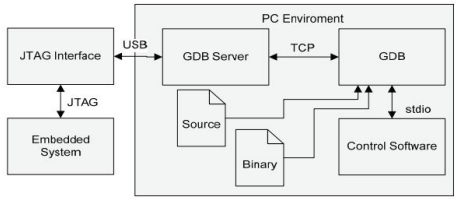
Słowa kluczowe: symulacja błędów, interfejs JTAG, projekt GDB, mikrokontrolery.

these systems create some implementation problems with SWIFI. There are only a few reports in the literature, so further research and new experience are still needed in this issue. The authors particularly concentrate on very popular ARM7 microcontrollers family [10]. In the presented solution fault injection is performed through JTAG interface available in the ARM platform. Fault injection is controlled by PC software that interfaces with GNU Debugger (GDB) [5] for ARM devices. This approach can be widely adopted to different hardware platforms supported by GDB project. This usually requires some modifications within the processor JTAG interface and the controlling GDB server.

The paper is organized as follows: Section 2 outlines architecture of the designed fault injector, Section 3 presents experimental results related to embedded program testing in ARM7 microcontroller environment. The final conclusions and future work perspectives are presented in Section 5.

2. GDB-Based Fault Injector

The developed Fault Injector architecture is presented in Fig. 1. Direct access to the tested embedded system is realized by JTAG probe compatible with a given architecture. In our studies we used J-Link pro device manufactured by SEGGER company [11].



Rys. 1. Architektura systemu do symulacji błędów opartego na GDB
Fig. 1. Architecture of GDB-Based fault injector system

Methodology and Progress

- Approach
  - Use GDB to simulate these faults on the Raspberry Pi Pico
  - Create
- Continuing Progress
  - Conducting extensive fault scenarios
  - Improve the device's resilience
  - Identification of areas where the device needs improvement to handle faults

Next Steps

- Research
  - Begin Implementing faults
  - What have others done? (Cont)
  - Enhancing fault tolerance and documenting best practices…