Low cost external serial interface watchdog for SoCs and FPGAs automatic characterization tests

Low cost external serial interface watchdog for SoCs and FPGAs automatic characterization tests

Paolo Bernardi, Gabriele Filipponi, Tommaso Foscale, Giorgio Insinga
Politecnico di Torino, Italy name.surname@polito.it

***Abstract*—Manufacturers must characterize their design deeply when designing and producing devices like FPGAs and SoCs. Information collected through simulation and physical experiments is the primary data source for manufacturers that can then decide the optimal working ranges of multiple critical parameters such as operating voltage, frequency, temperatures, etc. With complex devices such as SoCs, and FPGAs with integrated PLLs and voltage regulators, each combination of voltage and frequency can be checked by communicating the desired parameters to the DUT, running a functional test, and observing the results. However once the ATE sends the desired parameters to the DUT through SPI or other serial interfaces, the DUT may freeze and stop to accept new commands entirely. This is particularly problematic for targeted characterization that may include a minimal number of boards and DUTs and where the ATE may simply be a simple laptop without any automatic DUT reset capabilities. This paper presents an external serial communication watchdog designed using an ESP32-based board. Our watchdog can detect the communications coming from the ATE, monitor the answers from the DUT, and restart it through power cycling in case of freezing.**

***Index Terms*—Automotive SoC, Reliability, Debugging, Serial Multiplexing**

I. Introduction

The complexity of modern System On Chip (SoC), and Field Programmable Gate Array (FPGA) devices is increasing every year. Manufacturers have to invest more and more in their tests to ensure that each commercialized device is correctly working and follows the declared specifications. To ensure these characteristics and compensate for the devices' infant mortality [1], manufacturers apply a complete set of tests [2] [3]. That can, in first approximation, be divided into structural and functional tests: functional tests evaluate the system's behavior based on its intended functionality. Meanwhile, structural tests operate on a hardware layer inserted to test the device as a sequential circuit. Functional test, in particular, can quickly check the behavior of a DUT by making it run a specific set of test programs and compare the output of the DUT with the expected values. These functional tests are executed in multiple conditions of supply voltage and working frequency. Through serial communication, the ATE can send these parameters to the DUT. The ATE then waits for the execution of the tests. The DUT then sends the results through the serial interface. However, with specific combinations of parameters, such as frequency or voltage, the device completely freezes, halting all of its functional operations. The device must be reset by performing a power cycling to continue the tests. However, if the tests are very targeted, with a few DUTs, the ATE may even be a laptop with limited capabilities and no way to perform power cycling.

In this paper, we will present a low-cost external serial interface watchdog able to analyze the communication between the DUT and the ATE and power cycle the DUT in case of freezing.

In this work, we will first talk in section II about the background, in section III about the proposed approach, in section IV about experimental results, terminate on section V with the conclusion of this work.

II. Background

In this section, we will discuss how devices are tested through functional tests and introduce the watchdog concept.

A. Functional test

Functional test is performed by executing a test program on the DUT and observing the resulting output. These outputs are then compared with the expected ones, determining if the DUT shows misbehavior. The misbehavior can be introduced by multiple conditions such as under or over-voltage operations and operating at an excessive frequency.

B. Watchdog

A watchdog is a hardware system that monitors a DUT in search of deadlock, loop, or, in general, a device that does not respond anymore. Watchdog integrated into SoC can trigger an interrupt or an exception to bring the DUT to a safe state.

III. The proposed approach

In this section, we will analyze our proposal from both the point of view of the designed hardware shield and its firmware.

A. Hardware

Our proposal is a board based on the ESP32 SoC WROOM-32. The main specifics of this SoC are:

- Dual-Core Tensilica Xtensa LX6 32-bit microprocessor
- 240MHz Clock
- SRAM: 520 Kb
- ROM: 448 Kb

Additionally, the board has a series of GPIOs and supports the most used communication protocol, such as I2C, SPI, CAN 2.0, and most notably, for our application, UART. The board is designed to provide the supply voltage for the DUT directly and includes a PMOSFET used as a switch to power cycle the DUT when needed. A MOSFET driver is also included to drive the PMOSFET faster and with a higher voltage that the microcontroller could not provide. The board is shown in Fig. 1.

B. Software

To achieve our goals, we developed custom firmware for the ESP32, running Free RTOS, with the help of Visual Studio Code plugin for the Arduino IDE. The code is structured as reported in Figure 2. Our application uses two threads to reach our purpose efficiently. These two threads collaborate to manage the communication from the ATE (acting as master of the system) and the DUT while also handling the watchdog logic. Each of these two threads is assigned to a specific core in the system. A command queue acts as a bridge between

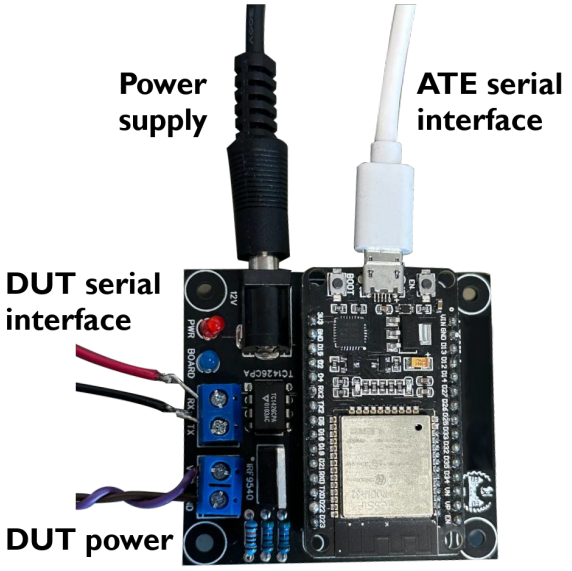

Fig. 1. Serial interface watchdog board based on the ESP32 microcontroller

the threads and collects the commands coming from the ATE. The queue also contains the result from the DUT after the execution of the functional tests.

Going deeper *Thread 0* is responsible for:

- serial manager for host communication: the ESP32 will receive commands and data from a serial USB interface from the host PC. *Thread 0* handles the incoming data from the serial interface. It will store the data and then process it;
- processing and storing the commands in a command queue. In this way, the data can pass from one thread to another.

Thread 1 has the following duties:

- serial manager for DUT communication: data from the host have to reach the DUT to allow the correct operation of the test. This thread extracts data from the command queue and reacts as needed: send a test vector to the DUT.
- Watchdog logic: it must recognize when the DUT is frozen and cannot respond, and in such case, reboot the DUT.

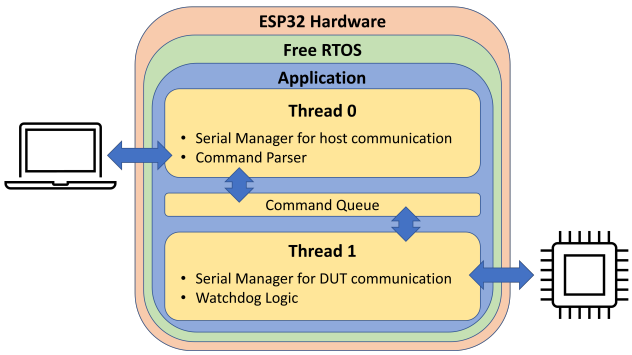

Fig. 2. Firmware Code Layering running on the ESP32 hardware

IV. Experimental Results

A. Hardware setup

To validate our system, we have considered, as a case study, an FPGA running a multicore SoC. The power supply does not directly reach the DUT as it passes through our system. In this way, our board can react to a freeze of the DUT and restart it by performing a power cycling. This behavior is obtained using the integrated MOSFET driver and PMOSFET transistor.

B. Automatic power cycling

To experimentally validate our board, we used a tool that we developed for flexible and targeted characterization using functional tests. This tool selects a given combination of parameters, such as voltage and frequency, and then sends them through a serial interface to our board. The board then forwards these commands to the DUT and waits for its answer, power cycling it in case of freezing. Fig. 3 shows the GUI of the program. On the one hand, the green box represents successfully completed tests. On the other hand, the blue box represents the freezing of our DUT that required the board to perform a power cycling.

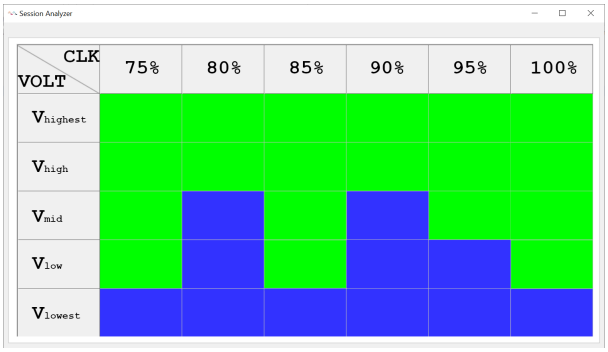

Fig. 3. Characterization device GUI. In green, passed the test. In blue, frozen device.

Without our device, human intervention would have been necessary to reset the devices and continue running the tests, a very time-consuming task.

V. Conclusion

This paper presents an external SPI watchdog that can be used to implement a characterization environment fast, flexible, and cheap. Our system, implemented in a compact PCB. In case the DUT freezes, reacts immediately by safely restarting it. This capability allows the execution exhaustive tests without requiring direct human intervention to reset the chip manually.

References

[1] G. Klutke, P. Kiessler, and M. Wortman, "A critical look at the bathtub curve," *IEEE Transactions on Reliability*, vol. 52, no. 1, pp. 125–129, 2003.

[2] P. Bernardi, R. Cantoro, A. Coyette, W. Dobbeleare, M. Fieback, A. Floridia, G. Gielenk, J. Gomez, M. Grosso, A. M. Guerriero, I. Guglielminetti, S. Hamdioui, G. Insinga, N. Mautone, N. Mirabella, S. Sartoni, M. S. Reorda, R. Ullmann, R. Vanhooren, N. Xamak, and L. Wu, "Recent trends and perspectives on defect-oriented testing," in *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2022, pp. 1–10.

[3] C. He et al., "Wafer level stress: Enabling zero defect quality for automotive microcontrollers without package burn-in," in *IEEE International Test Conference (ITC)*, 2020.