

Feasibility of Testing Functional Interfaces using Boundary Scan

CHENGWEN LIU

MASTER'S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY



Feasibility of Testing Functional Interfaces using Boundary Scan

Chengwen Liu
ch53121i-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Erik Larsson, Noroz Akhlagi, Niklas Gustafson

Examiner: Christian Nyberg

November 12, 2024

Abstract

Before electronic products reach the market, Functional Circuit Testing (FCT) is an essential process to ensure performance and reliability. Traditional FCT requires developing dedicated testing firmware for each batch of products, which increases both development costs and time. This thesis proposes a testing method based on Boundary Scan, provides an alternative that eliminates the need for firmware development.

To validate the feasibility of Boundary Scan, this thesis selects the Serial Peripheral Interface (SPI) protocol as case study. As SPI is a widely used interface in electronic products, testing the SPI protocol through Boundary Scan provides an initial assessment of the potential for applying Boundary Scan in real FCT scenarios.

This thesis uses two methods to implement Boundary Scan: one based on Open on-chip debugger (OpenOCD) software, and the other using a Field Programmable Gate Array (FPGA). Boundary Scan is implemented on a NUCLEO-F722ZE development board to facilitate SPI communication with another identical board. A comparison is made between the communication speeds and overall test times. The results demonstrate that Boundary Scan can successfully perform SPI communication testing, and it achieves shorter total test time under smaller communication content size than the firmware-based method.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Erik Larsson, for his invaluable guidance and insightful critiques, which have greatly enriched my work. I extend my deep appreciation to Noroz Akhlagi and Niklas Gustafson for their extensive support and assistance in the practical aspects of my research. I am also grateful to the team at Mikrodust AB for providing the necessary resources and environment that facilitated my research. Their expertise and cooperation have been crucial to the success of this thesis. Thank you all for your dedication and support.

Popular Science Summary

Have you ever used electronic products? Today in 2024, electronic products are almost everywhere, from mobile phones, computers to TVs, mice and keyboards that we use every day. All these devices need to undergo rigorous testing before leaving the factory to ensure that there are no faults when you receive them. These testing processes are to ensure the reliability of the products.

One testing process is called functional circuit testing (FCT). In FCT, engineers must develop a dedicated testing firmware for each batch of products. This firmware is like an "examiner" of the device to check whether the product is working properly. But as the types and number of electronic products continue to increase, it becomes time-consuming and costly to develop testing firmware for each product.

So, is there a way to save the steps of firmware development? Can the test be completed without using test firmware? This is exactly this thesis is about: using Boundary Scan to optimize the testing process.

The focus of this thesis is the feasibility of Boundary Scan in Serial Peripheral Interface (SPI) communication protocol testing. In this thesis, the SPI protocol was tested using Boundary Scan and compared it with traditional firmware testing methods. The results shows that in test scenarios with low data size transfer, Boundary Scan can not only complete the test reliably, but also reduce the test time.

Table of Contents

1	Introduction	1
1.1	Research Context	1
1.2	Problem Statement	1
1.3	Thesis Objectives	2
1.4	Thesis Structure	2
1.5	Related Work	3
2	Background	5
2.1	Printed Circuit Board	5
2.2	Common Testing Methods	6
2.3	Common Communication Protocol	7
2.4	Boundary Scan	10
2.5	Boundary Scan Description Language	12
2.6	Open On-Chip Debugger	12
2.7	Field Programmable Gate Array	12
2.8	Firmware	12
2.9	Evaluation of Protocols	13
3	Methodology	15
3.1	Interface Test Method	15
3.2	Theoretical Communication Speed	16
3.3	SPI Data Receiving	18
3.4	Boundary Scan Implementation	18
4	Experiment and Result Analysis	23
4.1	Experimental Equipment Selection	23
4.2	Experimental Platform Configuration	27
4.3	Result Analysis: Communication Feasibility	29
4.4	Result Analysis: Communication Speed	30
4.5	Result Analysis: Total Testing Time	32
5	Conclusion and Future Work	35
5.1	Conclusion	35
5.2	Evaluation of Boundary Scan and OpenOCD in FCT	35

5.3 Future Work	36
References _____	39

List of Figures

2.1	The PCB of a sound card.	5
2.2	Test instrument produced by Mikrodust.	7
2.3	UART protocol architecture.	8
2.4	I2C protocol architecture.	9
2.5	SPI communication architecture.	10
2.6	Boundary Scan architecture.	10
2.7	Boundary Scan FSM.	11
3.1	Waveform of the letter "o" transmitted on MOSI and MISO.	18
3.2	Boundary Scan process for SPI.	19
3.3	Boundary Scan FSM process.	22
4.1	NUCLEO-F722ZE development board.	24
4.2	FT2232HL board by Mikrodust.	25
4.3	FT2232HL development board schematic.	25
4.4	Nexys 4 development board.	26
4.5	RIGOL DS1054 oscilloscope.	26
4.6	Firmware-based method configuration.	27
4.7	OpenOCD-based Boundary Scan method configuration.	28
4.8	FPGA-based Boundary Scan method configuration.	28
4.9	Firmware waveform of SLCK and MISO.	29
4.10	OpenOCD-based waveform of SLCK and MISO.	29
4.11	One bit transmission time of OpenOCD-based Boundary Scan method.	30
4.12	One bit transmission time of the FPGA-based method.	31
4.13	Transmission speed comparison for the three methods.	31
4.14	Total test time for the three methods.	33
4.15	Test time comparison for different data sizes across the three methods.	34

List of Tables

2.1	Overview of Testing methods	8
3.1	Partial Boundary Scan cell description	20
3.2	Boundary Scan data configuration for a pin	21

List of Acronyms

Acronym	Definition
AOI	Automated Optical Inspection
BGA	Ball Grid Array
BSDL	Boundary Scan Description Language
CS	Chip Select
DUT	Device Under Test
FCT	Functional Circuit Testing
FSM	Finite State Machine
FPGA	Field Programmable Gate Array
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICT	In-circuit Testing
IR	Instruction Register
JTAG	Joint Test Action Group
kbps	Kilobits Per Second
MISO	Manager Input Subordinate Output
MOSI	Manager Output Subordinate Input
OpenOCD	Open On-chip Debugger
SCLK	Serial Clock
SPI	Serial Peripheral Interface
TAP	Test Access Port
TCK	Test Clock
TMS	Test Mode Select
UART	Universal Asynchronous Receiver/Transmitter

Introduction

1.1 Research Context

In the electronics manufacturing industry, the quality of electronic products are important. Ensuring that each product meets quality standards is a critical aspect of the production process. Testing is an indispensable step, which is used to identify potential defects early in the manufacturing process and ensuring each product meets design and functional requirements before reaching the market.

One essential testing process is Functional Circuit Test (FCT). FCT verifies whether an electronic product operates as intended under simulated real-world conditions. FCT usually involves defining test requirements, developing testing firmware, setting up testing environment, and connecting the device under test (DUT) to equipment such as power supplies, signal generators, and oscilloscopes, these equipment evaluate the product's performance under various operational conditions.

1.2 Problem Statement

The Global consumer electronics market is expected to grow at a compound annual growth rate of 6.6% from 2023 to 2030 [1]. As the variety and complexity of electronic products increase, FCT needs to adapt to a wider range of designs. One significant challenge is the time and resources needed for developing dedicated testing firmware for each new product. Additionally, downloading the testing firmware onto the DUT increases testing time. Recognizing these issues, this thesis aims to explore more efficient and flexible methods that not only reduce or eliminate the need for testing firmware but also shorten overall testing times.

In traditional FCT, the behavior of the DUT is controlled by testing firmware. The Firmware instructs the DUT to perform specific testing tasks. It can be tasks

like simply pulling up the pins or enabling the DUT to communicate with other devices. These tasks are essentially controlling the DUT's pins to output "0" or "1". This thesis proposes to operate the pins directly at hardware level. To achieve this, Boundary Scan is introduced. Boundary Scan is an integrated method for testing interconnects on PCBs that are implemented at the integrated circuit (IC) level [8]. By sending commands through the Joint Test Action Group (JTAG) interface, the pins of the DUT can be externally controlled and monitored.

1.3 Thesis Objectives

The objectives is to explore the feasibility of Boundary Scan to handle testing of functional interface, with a focus on Serial Peripheral Interface (SPI) protocol as a case study. In addition, this thesis will compare the communication speed of Boundary Scan with the method using firmware. Furthermore, the overall testing time will be analyzed to assess whether Boundary Scan can reduce the total testing time.

To achieve these objectives, two implementations of Boundary Scan will be used: one utilizing the Open On-Chip Debugger (OpenOCD) with the specified FT2232HL chip and another one with the Nexy4 Field-Programmable Gate Array (FPGA) board [2]. Both implementations will be tested on the NUCLEO-F722ZE STM32 development board [3], which will be configured as the SPI manager device, communicating with another identical board configured as the SPI subordinate device.

1.4 Thesis Structure

This thesis is structured as follows:

Chapter 1: Introduction, introduces the research content and objectives of this thesis.

Chapter 2: Background, introduces the techniques required for this thesis.

Chapter 3: Methodology, describes the implementation methods and the theoretical calculations.

Chapter 4: Experiment and result analysis, describes the experimental configuration, analyzes the experimental results, and discusses its significance.

Chapter 5: Conclusion and future work, summarizes the results and proposes future research directions.

1.5 Related Work

Keysight Technologies has applied Boundary Scan extensively in their printed circuit board (PCB) testing systems [4]. Keysight uses Boundary Scan to test interconnections between components, including ball grid arrays (BGAs) and other surface-mounted devices, where physical probing access is restricted. Boundary Scan allows Keysight to check for defects in the board's interconnects and ensure that components communicate as expected. Boundary Scan reduces the need for customized test fixtures, which can be costly and complex in dense designs. Keysight's use of Boundary Scan significantly improves fault detection in both the development and production stages.

XJTAG has integrated Boundary Scan into its testing solutions for prototype debugging, production testing, and board bring-up [5]. XJTAG's tools allow engineers to access and control JTAG-enabled pins on various ICs, helping them verify connectivity and functionality without requiring extensive test fixtures. One of the key advantages of XJTAG's Boundary Scan tools is their ability to automate the testing process, reducing manual intervention and improving the speed of the test cycles.

A recent study from the University of Electronic Science and Technology of China (2023) presented an advanced method for optimizing test vector generation in Boundary Scan-based testing [6]. Their approach focused on enhancing the efficiency of fault detection in complex PCB designs by improving the generation of test patterns, which are used to identify defects in the interconnections of the board. The researchers demonstrated that by refining the test vectors, they were able to increase the fault coverage while reducing the time required for testing.

2.1 Printed Circuit Board

A PCB serves as a platform for mechanically supporting and electrically connecting various electronic components. It realizes the connection by carving conductive paths on the insulating substrate. With the development of technology, PCBs have evolved from single-layer to multi-layer [7], which allows more components to be integrated in a small space to meet the higher complexity and functional requirements of electronic device. Figure 2.1 shows the PCB of a sound card.

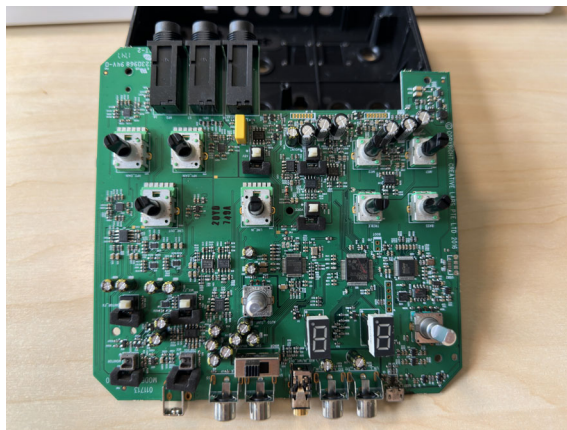


Figure 2.1: The PCB of a sound card.

2.2 Common Testing Methods

Several methods are implemented for testing during electronic production. These methods include precise examinations conducted with equipment, such as automated optical inspection (AOI) and X-ray inspection. In-circuit testing (ICT) is used to perform electrical tests such as checking for short circuits or open circuits. In order to simulate the actual conditions under the final working environment, FCT is also employed to verify the performance.

An AOI system typically integrates both hardware and software. The hardware consists of image sensors and lighting systems: captures images of the PCB's surface, such as solder joints and component placements. The software analyzes these images to detect defects such as component misalignment or soldering issues [9]. By processing features extracted from images, the system can classify products as defective or non-defective according to predefined criteria. AOI is an efficient inspection method, especially suitable for quickly identifying surface-level defects on the production line.

X-ray inspection is an inspection method that can reveal defects in the internal structure of PCBs, such as soldering problems and short circuits between layers. By absorbing X-rays, heavier materials such as tin and lead in solder are clearly visible, while lighter materials, like the PCB substrate, appear more transparent [10]. This enables the detection of defects like solder bubbles that may remain hidden from other inspection methods. X-ray inspection is particularly important for multi-layer PCBs and complex assemblies such as BGA packages [11], as many aspects of these structures are not easily assessed through traditional visual inspection methods.

ICT measures electrical properties such as resistance, capacitance, and inductance by directly contacting test points on the PCB. ICT can be carried out using either a bed-of-nails fixture, where spring-loaded pins make contact with hundreds of points on the board, or a flying probe that moves across the PCB, making contact as needed. The bed-of-nails setup is more rigid and costly, making it difficult to adapt to changes, especially on densely populated boards. The flying probe, on the other hand, offers more flexibility, as its movements are controlled by software, allowing for easy updates. While ICT is effective for verifying component placement and electrical connections, it does not evaluate the functionality of the board itself [10].

FCT simulates the working environment of the PCB to verify whether its functions and performance meet the design requirements. This step is usually the last step in the testing process. At Mikrodust, test instrument produced is used in FCT [12], which is shown in Figure 2.2. Test instrument includes test fixtures and probes customized for PCB, computers and instrument boards which perform various test functions [13]. The probe establishes a connection via the predetermined test point on the PCB and downloads the test firmware to the PCB. Boards with test functions, such as detecting voltage, current, etc., monitor and record

the response of the PCB under the control of the test firmware and computer in real time. These data are used to evaluate the performance of the PCB.

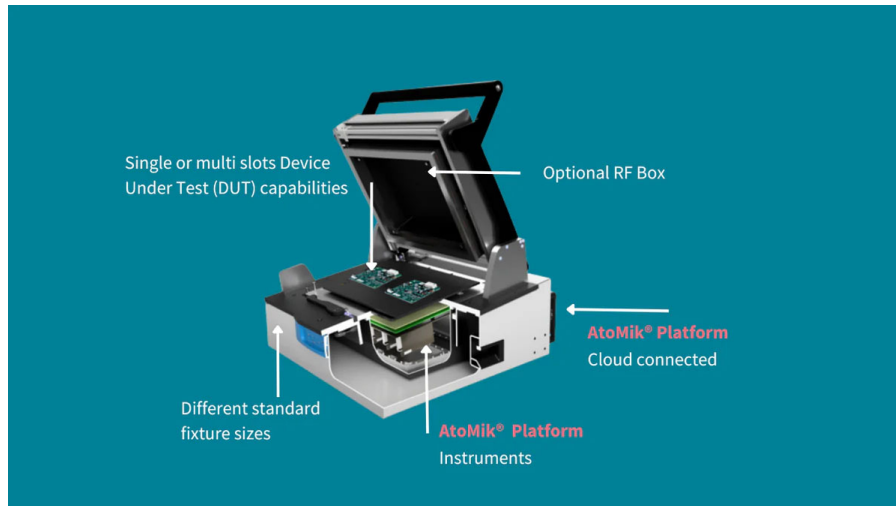


Figure 2.2: Test instrument produced by Mikrodust.

AOI and X-ray inspection primarily ensure production quality of the PCB. ICT and FCT are focused on performance evaluation. ICT primarily assesses basic parameters such as resistance, component polarity, voltage, and current. On the other hand, FCT is more advanced, evaluating aspects like amplification, transfer function, and signal levels in both analog and digital circuits. The following table summarizes the features of these PCB test methods.

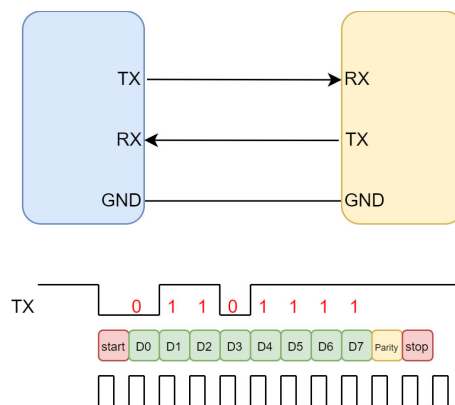
2.3 Common Communication Protocol

Universal Asynchronous Receiver/Transmitter (UART) is a protocol for asynchronous communication that allows devices to transmit data without the need of a clock signal [14]. Instead of relying on an external clock, synchronization between the transmitting and receiving devices is achieved by both maintaining a consistent baud rate. The transmitter encodes the data into a serial format, while the receiver samples the incoming data using its own internal clock. This avoids the need for simultaneous transmission of clock and data signals. UART typically utilizes two communication lines: a transmit line (TX) and a receive line (RX), where data is sent through the TX line and received via the RX line.

Figure 2.3 shows the architecture of the UART protocol between two devices. The data transmission rate of UART is usually no more than 1 Mbps, and it supports full-duplex communication, data can be transmitted in both directions simultaneously. The UART data format includes a start bit, 5 to 9 data bits, an optional parity bit, and one or two stop bits. Figure 2.3 shows the signal

Table 2.1: Overview of Testing methods

Testing Method	Main Focus	Advantages	Disadvantages
AOI	Surface Quality	Fast inspection speeds, ideal for production lines.	Limited to visible areas, not suitable for hidden connections under components.
X-ray	Internal Structures	Provides detailed views of internal layers, essential for complex PCBs.	High investment costs; less effective for simpler boards.
ICT	Circuit Verification	Direct measurement of components, accurate electrical testing.	Requires expensive fixtures or detailed programming; may miss connector faults.
FCT	Overall Functionality	Comprehensive system-level evaluation, ensures final product reliability.	High cost due to complex programming and specialized equipment.

**Figure 2.3:** UART protocol architecture.

waveform process of transmitting the ASCII character 'o' (01101111) through the UART protocol, including the start bit, 8 bits of data, parity bit and stop bit.

Inter-Integrated Circuit (I2C) is a half-duplex communication protocol that allows multiple devices to exchange data over a single bus using two lines: a serial data line for data transmission and a serial clock line to synchronize communication [15]. Devices connected to the I2C bus are identified by unique addresses, and data transmission is structured into frames that include a start bit, eight data bits, an optional check bit, and a stop bit. The clock speed for I2C communication can be adjusted depending on the application, with standard speeds starting from 100 Kbps and going up to 3.4 Mbps in high-speed mode.

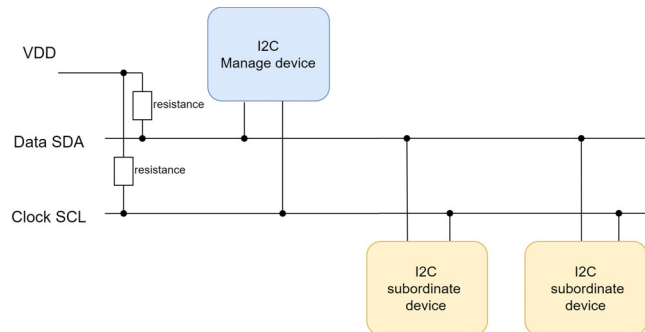


Figure 2.4: I2C protocol architecture.

In I2C communication, pull-up resistors are essential. The I2C bus is an open-drain output design, which requires pull-up resistors to ensure that the line remains high when no data is transmitted. This design allows multiple devices to be safely connected to the same data line and pull the line down for data transmission only when it is absolutely necessary, thereby effectively avoiding conflicts and damage on the data line.

The SPI protocol enables high-speed, synchronous, full-duplex communication between microcontrollers and peripheral devices [15]. The SPI protocol follows a manager-subordinate architecture where the manager device, typically a microcontroller, manages communication with one or more subordinate devices. The protocol uses four key signal lines: the serial clock (SCLK), manager output subordinate input (MOSI), manager input subordinate output (MISO), and chip select (CS).

In the SPI protocol, the manager device synchronizes the entire communication process by generating a clock signal, and data is transmitted bidirectionally between the manager device and the subordinate device through MOSI and MISO lines [14]. The CS line is used by the manager device to select a specific subordinate device to communicate with.

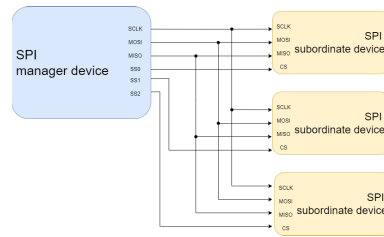


Figure 2.5: SPI communication architecture.

2.4 Boundary Scan

Boundary Scan is an integrated method for testing interconnects on PCBs that are implemented at the integrated circuit (IC) level [8]. This standard embeds a standardized set of test logic into an IC, allowing the pin status of the IC to be accessed and controlled through JTAG interface. Figure 2.6 shows the basic hardware structure of Boundary Scan.

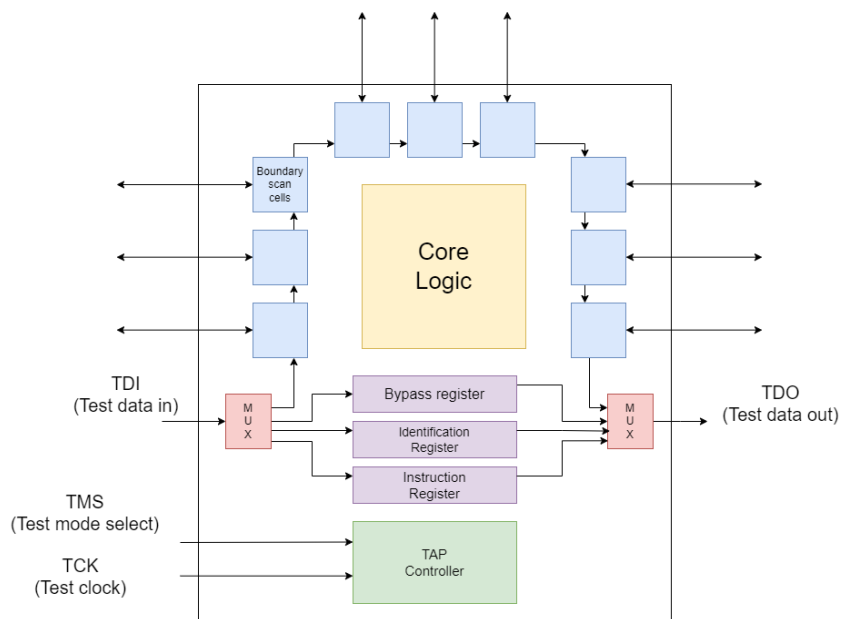


Figure 2.6: Boundary Scan architecture.

Boundary Scan cells inside the IC are directly connected to each pin. Each Boundary Scan cell typically consists of a capture register and an update register. The Capture register is responsible for capturing the current state of the associated pin. The Update register is responsible for driving the state of the associated pin, which can set the pin to a specific state. The Test access port (TAP) Controller controls state transitions based on signals from test clock (TCK) and test

mode select (TMS). Instruction Register (IR) holds the current Boundary Scan instruction which dictates the operation of the TAP controller and the behavior of the test. Bypass register provides a direct path for test data to pass through the cells when Boundary Scan is not needed. Identification Register stores information about the IC, such as the manufacturer ID and part number.

The TAP finite state machine (FSM) controls the behavior of Boundary Scan and ensures that Boundary Scan process can be performed in the specified order. The TAP FSM switches states by following the TMS signal. The TAP FSM contains multiple states, such as "Test-Logic-Reset", "Run-Test/Idle", "Select-DR-Scan", "Capture-DR", "Shift-DR", "Exit1-DR", "Pause-DR", "Exit2-DR", "Update-DR", and the corresponding IR status as shown in the figure below, each status corresponds to a Boundary Scan specific operation, such as loading test data (Shift-DR), executing test instructions (Update-IR), etc. Through the FSM, the behavior of Boundary Scan can be precisely manipulated.

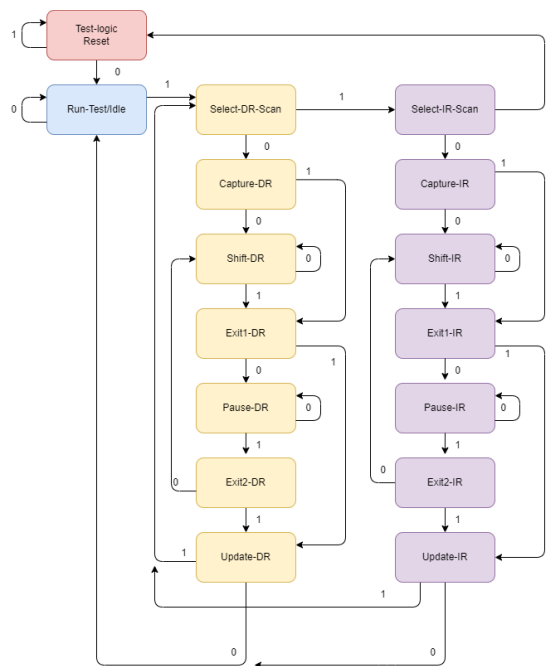


Figure 2.7: Boundary Scan FSM.

The Boundary Scan processes are loading Boundary Scan commands and data to the registers and reading the data out through JTAG port. Commands and data are transferred from test data in (TDI) port into the register chain, and at the same time, data inside the register chain is transferred out from test data out (TDO) port, both following the clock signals from the TCK port. By analyzing the output data, the pin status can be known.

2.5 Boundary Scan Description Language

The Boundary Scan Description Language (BSDL) as outlined by the IEEE 1149.1 standard provides a detailed description of the Boundary Scan architecture within an IC [16]. BSDL specifies the configuration of Boundary Scan cells, their connection to the IC's physical pins, and the scan chain's structure. This standardized language allows engineers to define the testability features and ensure accurate testing and diagnostics of ICs.

2.6 Open On-Chip Debugger

The Open On-Chip Debugger (OpenOCD) is an open-source software tool designed for debugging, in-system programming, and testing of embedded devices. OpenOCD is an ideal complement for the GNU GCC toolchain for ARM processors [17]. OpenOCD operates with the help of a debug adapter a hardware module that ensures appropriate electrical signaling to the target device. OpenOCD supports various debug adapters and target devices and offers the flexibility to use scripts for automation complex debugging tasks. OpenOCD provides support for Boundary Scan testing, making it a comprehensive tool for embedded system development and testing.

2.7 Field Programmable Gate Array

FPGAs are specialized hardware devices that can be reprogrammed to perform different tasks after manufacturing. FPGAs consist of a grid of programmable logic blocks, which can be configured to implement digital circuits and interconnections that link these blocks. This flexibility allows FPGAs to execute a wide range of complex digital logic operations.

FPGAs are commonly used in areas like signal processing, embedded systems, and communication protocols because they can handle tasks like data manipulation, algorithm acceleration, and high-speed input/output management. This makes them ideal for prototyping and testing, as well as for applications that may require adjustments over time.

2.8 Firmware

Firmware is a type of software embedded directly into a hardware device to control its specific functions [18]. Firmware operates at the closest level to the hardware, managing communication, input/output processes, and controlling basic system

tasks. It is stored in non-volatile memory, which ensures that it persists even when the device is powered off. Firmware plays an essential role in electronic devices such as microcontrollers, sensors, and communication modules where it manages basic operations and ensures that hardware components interact correctly. It is typically updated less frequently than regular software, as it controls fundamental aspects of the device's behavior.

2.9 Evaluation of Protocols

UART is simple to implement, but it has high requirements on signal stability. Any instability in the signal may cause communication failure. When using Boundary Scan to simulate UART communication, requirements are placed on the Boundary Scan device. Some Boundary Scan devices cannot generate stable signals, which will cause UART communication failure.

In I2C, the reliance on pull-up resistors to ensure that the data and clock lines remain high when inactive poses a challenge to Boundary Scan simulation. The hardware architecture of Boundary Scan is not designed to support this type of continuous resistor drive, which limits its effectiveness in simulating I2C communication. Therefore, simulating I2C communication based on Boundary Scan may encounter some practical problems.

In SPI, the clock signal is provided by the manager device, which allows the clock signal to have a certain error and does not have to maintain a fixed frequency, providing flexibility for the communication process. When using Boundary Scan to implement SPI, the frequency fluctuation of the clock signal will not have a significant impact on the communication results. Therefore, this clock signal can be generated with some simple Boundary Scan hardware. In addition, SPI does not rely on pull-up resistors like I2C, and the signal line is driven directly by the device. These features simplify the hardware requirements, making SPI an ideal choice for simulation using Boundary Scan.

Methodology

3.1 Interface Test Method

To evaluate the feasibility of Boundary Scan in testing functional interfaces and its performance, this thesis explores three testing methods for SPI communication: a firmware-based method, an OpenOCD-based Boundary Scan method and a FPGA-based Boundary Scan method.

In the firmware-based method, a testing firmware is developed for the DUT. The testing firmware is first loaded onto the DUT which is configured as the SPI manager device. The manager device then initiates communication with another identical DUT configured as the SPI subordinate device. The manager sends clock signals to the subordinate to request one byte of data, namely "o". The entire time from when the command is sent by the manager to when the response byte is received is measured and recorded with an oscilloscope to calculate the communication speed.

In the OpenOCD-based Boundary Scan method, OpenOCD software is used to control the DUT's pins directly via the JTAG interface. Boundary Scan manipulates the DUT's pins externally to initiate SPI communication. The process is similar to the firmware-based method: the manager device sends clock signals to the subordinate device, requesting one byte of data, namely "o". The total communication time is measured and recorded using an oscilloscope.

In the FPGA-based Boundary Scan method, a custom Boundary Scan controller is implemented on an FPGA to directly control the DUT's pins via the JTAG interface. Similar to the OpenOCD-based method, the FPGA manipulates the pins externally to initiate SPI communication. The manager device sends clock signals to the subordinate device, requesting one byte of data, namely "o". The communication time is measured and recorded using an oscilloscope.

Through the above three methods, the feasibility of using Boundary Scan in

testing SPI interfaces can be verified. Additionally, there is a comparison of communication speed and total test time.

3.2 Theoretical Communication Speed

In FCT, test time is one of the important features. Less test times means reduction of the cost. This section will explore the theoretical communication speeds of the three methods.

3.2.1 Firmware-based Method Theoretical Communication Speed

In the firmware-based method, when the DUT is placed onto the test instrument, the test instrument downloads the firmware to the DUT and then starts the SPI test. Therefore, the total test time is the firmware downloading time plus SPI communication time.

$$T_{total} = kx + t \quad (3.1)$$

Here x is the SPI communication content length measured in bits. k is the transmission time for 1 bit, and t is the firmware download time.

For example, assuming the transmission time for one bit is 1ms and the firmware download time is 1.5s, the formula can be written as:

$$T_{total} = 1ms \times x + 1.5s \quad (3.2)$$

If the length of the SPI communication content x is 1000 bits, the total test time would be:

$$T_{total} = 1ms \times 1000 + 1.5s = 2.5s \quad (3.3)$$

Thus, the total test time T_{total} is 2.5 seconds. This example illustrates how the total test time depends on both the SPI communication content and the firmware download time.

3.2.2 OpenOCD-based Boundary Scan Method Theoretical Communication Speed

In the OpenOCD-based Boundary Scan method, when the DUT is placed onto the test instrument, the instrument can directly use the JTAG interface to perform

Boundary Scan on the DUT and start the SPI test. Therefore, the total test time is SPI communication time. Each SPI clock cycle requires two Boundary Scan operations: one to generate a "1" and another to generate a "0" for a complete clock period. The time for one Boundary Scan operation is determined by the number of Boundary Scan cells (denoted as m) and the Boundary Scan clock frequency (denoted as f_b). The more Boundary Scan cells there are, the longer it takes to shift in the data. The Boundary Scan clock frequency determines how fast the data can be shifted in. Additionally, there is a delay D between each Boundary Scan operation when using OpenOCD. Therefore, the total test time is:

$$T_{total} = 2 \times \left(\frac{m}{f_b} + D \right) \times x \quad (3.4)$$

The number of Boundary Scan cells is described in the BSDL file from the chip manufacturers. In this example, it is assumed to be 406. The Boundary Scan clock frequency is assumed to be 1 kHz, and the delay time per operation is 1 ms. The total test time can be calculated as follows:

$$T_{total} = 2 \times \left(\frac{406}{1000} + 0.001 \right) x = 0.814 s \times x \quad (3.5)$$

If the length of the SPI communication content x is 1000 bits, the total test time would be:

$$T_{total} = 0.814 s \times 1000 = 814 s \quad (3.6)$$

3.2.3 FPGA-based Boundary Scan Method Theoretical Communication Speed

The FPGA-based Boundary Scan method is similar to the OpenOCD-based method, because FPGA can continuously send signals and there is no delay between each Boundary Scan operations. Therefore, the time expression of FPGA is:

$$T_{total} = \frac{2m}{f_b} x \quad (3.7)$$

Here x is the length of SPI communication content measured in bits, m is the the number of Boundary Scan cells, and f_b is the Boundary Scan clock frequency.

The number of Boundary Scan cells is described in the BSDL file from the chip manufacturers. In this example, it is assumed to be 406. The Boundary Scan clock frequency is assumed to be 1 kHz. The total test time can be calculated as follows:

$$T_{total} = \left(\frac{2 \times 406}{1000} \right) x = 0.812 s \times x \quad (3.8)$$

If the length of the SPI communication content x is 1000 bits, the total test time would be:

$$T_{total} = 0.812 s \times 1000 = 812 ms = 0.812 s \quad (3.9)$$

3.3 SPI Data Receiving

During SPI communication, the manager device generates a clock signal to synchronize the data transfer from the subordinate device via the MISO line. In this thesis, the objective is to read the letter "o" from the subordinate, which corresponds to the ASCII binary value "01101111" (8 bits). Therefore, the manager needs to produce 8 clock signals to receive the complete data.

Figure 3.1: Waveform of the letter "o" transmitted on MOSI and MISO.

As the manager sends the clock signals over the SCLK line, the subordinate simultaneously transmits each bit of data on the MISO line. During each clock period the manager reads one bit of data from the subordinate. After 8 clock periods, the manager will receive the full 8-bit ASCII code for the letter "o".

3.4 Boundary Scan Implementation

The overall process consists of three core steps: executing Boundary Scan commands, handling the test data, and the finite state machine (FSM) operation that drives the Boundary Scan functionality.

3.4.1 Overview of Boundary Scan Process

In this SPI process, Boundary Scan allows precise manipulation of the SCLK pin while monitoring the MISO pin for the subordinate's response.

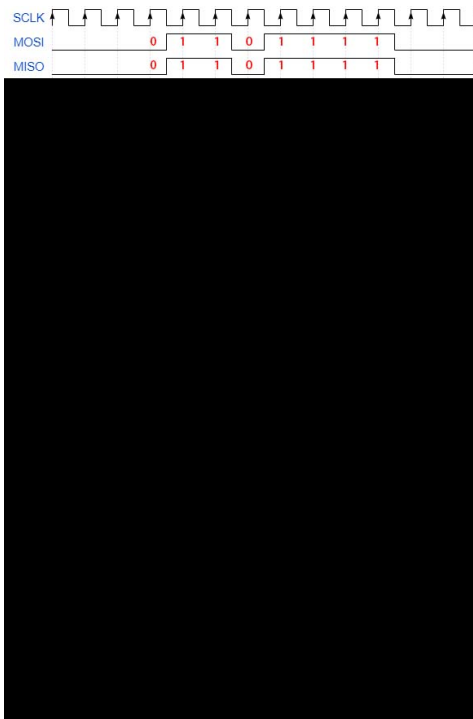


Figure 3.2: Boundaries Scan process for SPI.

The Boundary Scan method for SPI communication can be divided into three key steps: loading instructions, shifting data, and analyzing data. The process is illustrated in Figure 3.2.

Loading Instructions: The first step of Boundary Scan is to load Boundary Scan instructions into the IR. These instructions are used to control the behavior of Boundary Scan.

Shifting Data: Data is shifted into the Boundary Scan cell. The data shifted in is used to change the state of the pin. At the same time, the existing data is shifted out.

Analyzing Data: The data shifted out is further analyzed to determine the actual state of the pin. The data in the Boundary Scan cell reflects the actual state of the pin, such as high level, low level or high impedance.

3.4.2 Boundary Scan Data

Each bit of the input data is assigned to a specific Boundary Scan cell, establishing a direct one-to-one mapping between data bits and the Boundary Scan cells. This mapping is defined in the BSDL file, which specifies the purpose and configuration of each cell, ensuring that data is correctly assigned and managed. A partial description of the Boundary Scan cells is shown in the following table:

Table 3.1: Partial Boundary Scan cell description

Num of bits	Cell	Port	Function
297	BC_1	*	CONTROL
296	BC_1	PA3	OUTPUT3
295	BC_4	PA3	INPUT
294	BC_1	*	CONTROL
293	BC_1	PA4	OUTPUT3
292	BC_4	PA4	INPUT
291	BC_1	*	CONTROL
290	BC_1	PA5	OUTPUT3
289	BC_4	PA5	INPUT

Each pin is controlled by multiple Boundary Scan cells. For example, the PA3 pin is controlled by three distinct cells: 297, 296, and 295. Each cell's state dictates the behavior of the pin, such as whether it is outputting a signal, receiving an input, or in a high-impedance state. The data Configuration for a pin is shown in Table 3.2. By combining the states of these cells, the precise behavior of the pin can be effectively managed and monitored. This allows for the determination of each

pin's state, such as whether it is at a high level, low level, or in high impedance, thereby enabling effective control and management of the DUT's pins.

Table 3.2: Boundary Scan data configuration for a pin

Control	Output	Input	State
1	1	X	Output High (1)
1	0	X	Output Low (0)
0	X	1	Capture state
0	X	0	High Impedance

To generate the SPI clock signal, the SCLK pin is controlled to output alternating "0" and "1" values, while the MISO pin is monitored. Other pins are kept at their default values. So the related cell's data cycles between "110" and "100" . During this process, the data shifted out from the Boundary Scan cells is analyzed to determine the MISO pin state, thus obtaining the SPI communication result.

3.4.3 Boundary Scan TAP FSM

In Boundary Scan operations, the TAP FSM is used to control the execution of Boundary Scan behaviors. The FSM consists of a series of states, State transition is controlled by by interpreting the TMS signal. To correctly execute Boundary Scan instructions and data shifting, the FSM must follow a specific state transitions sequence.

In Figure 3.3, the numbers represent the sequential order of FSM states in the Boundary Scan process. The FSM follows a specific sequence of transitions to control the behavior of the Boundary Scan cells. The process starts with selecting the IR. In this phase, the FSM transitions to the appropriate states to shift in the instruction data. Once the shifting process is complete, the FSM moves to the Update-IR state, where the newly shifted instruction is written into the IR. After the IR phase is complete, the FSM transitions to the data register (DR) phase. During this phase, new data is shifted in through the TDI port, the data representing the intended states of the pins. Once the shifting is complete, the FSM moves to the Update-DR state, where the shifted data is written into the data register and then applied to the chip's pins.

Through these state transitions, the TAP FSM accurately controls the behavior of the Boundary Scan cells, enabling monitoring and manipulation of pin states to facilitate automated SPI communication. Each FSM state plays a specific role in processing instructions and data. Both OpenOCD and FPGA implementations follow this sequence of state transitions to perform Boundary Scan operations, ensuring that each instruction is properly executed and data is accurately handled throughout the Boundary Scan process.

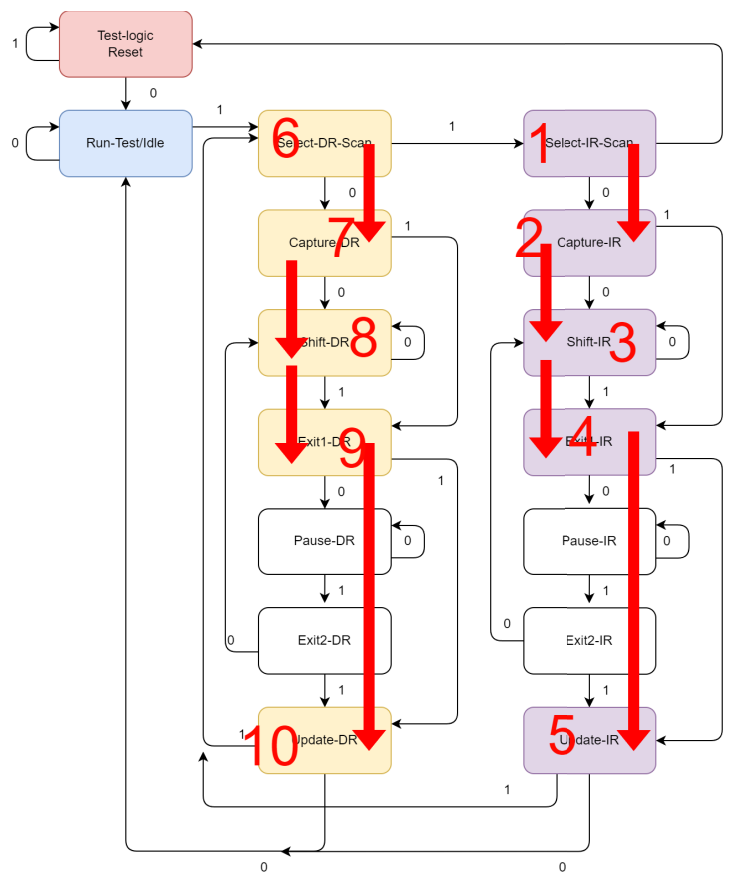


Figure 3.3: Boundaries Scan FSM process.

Experiment and Result Analysis

This chapter provides an overview of the selection of experimental equipment. Then describes the configuration of the testing platforms for both firmware-based and Boundary Scan-based methods. Finally, the chapter presents a detailed analysis of the experimental results, comparing the SPI communication feasibility, transmission speed, and total test time of each method.

4.1 Experimental Equipment Selection

4.1.1 Device Under Test

STM32F722 series NUCLEO-F722ZE development board [3] was selected as the DUT as shown in Figure 4.1,. The DUT is equipped with an STM32F722ZE microcontroller.

Choosing the STM32F722 series NUCLEO-F722ZE development board as the DUT is based on several considerations: STM32 series is a digital IC and it is used in various electronic products. The built-in SPI controller makes it very suitable for SPI communication test comparison. STM32F722 also supports Boundary Scan.

The experimental design uses two identical NUCLEO-F722ZE development boards. One is configured as the manager device, and the other is configured as the subordinate device in SPI communication.

4.1.2 OpenOCD-based Boundary Scan Hardware

A computer with Linux is used to run OpenOCD. OpenOCD with FT2232HL was used to implement Boundary Scan, FT2232HL is a dual-channel USB to serial/par-

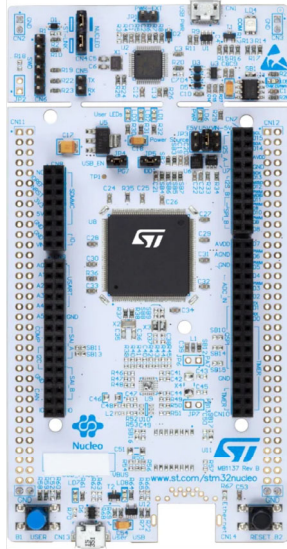


Figure 4.1: NUCLEO-F722ZE development board.

allel interface converter chip produced by FTDI [19]. FT2232HL supports multiple communication interfaces, including UART, JTAG, SPI and I2C. By utilizing its USB to JTAG interface function, FT2232HL allows the computer to communicate with the DUT to implement Boundary Scan. Figure 4.2 and Figure 4.3 shows the circuit and PCB of FT2232HL board.

4.1.3 FPGA-based Boundary Scan Hardware

The FPGA can generate high-speed and stable signals. By directly controlling signals at the hardware level, FPGA ensures more precise and efficient signal control than the OpenOCD. For this thesis, the Nexys 4 development board featured with a Xilinx Artix-7 FPGA [2] was chosen to implement Boundary Scan as shown in Figure 4.4.

4.1.4 Oscilloscope

An oscilloscope was used to observe the waveform of the SPI communication. The oscilloscope model is RIGOL DS1054. It is a 50 MHz Digital Oscilloscope with 4 channels plus 12 Mpt memory and 1 GSa/sec sampling.

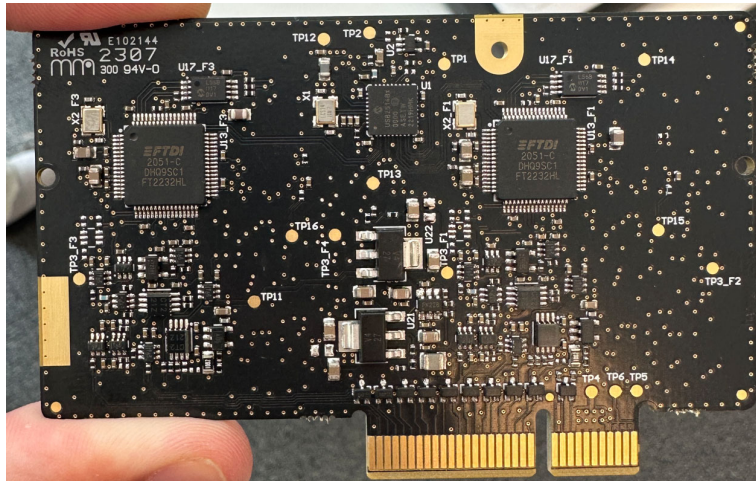


Figure 4.2: FT2232HL board by Mikrodust.

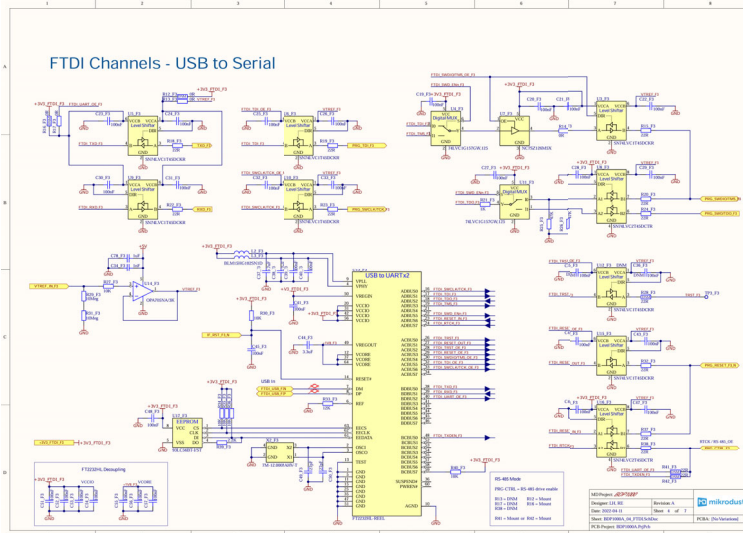


Figure 4.3: FT2232HL development board schematic.

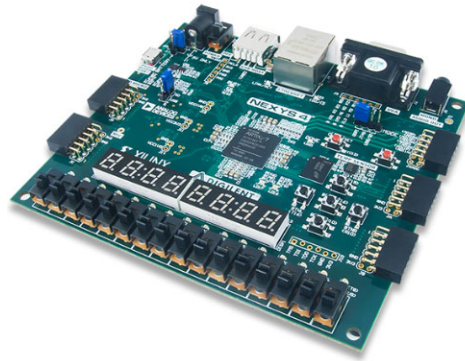


Figure 4.4: Nexys 4 development board.

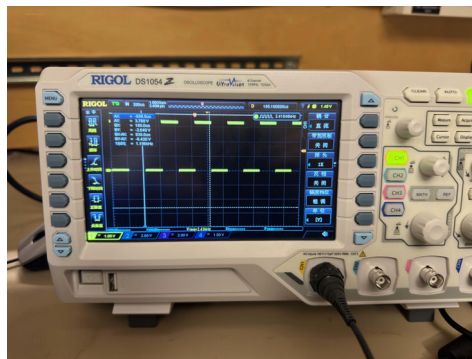


Figure 4.5: RIGOL DS1054 oscilloscope.

4.2 Experimental Platform Configuration

4.2.1 Firmware-based Method Configuration

Figure 4.6 shows the configuration of the firmware-based method. The configuration consisted of a computer, two NUCLEO-F722ZE development boards and an oscilloscope. Two boards were connected through the SPI interface on the board. The computer downloaded the test firmware to the development board through the ST-Link interface. The oscilloscope was connected to the SPI interface pins to monitor and display the waveform of each pin during the SPI communication process in real time.

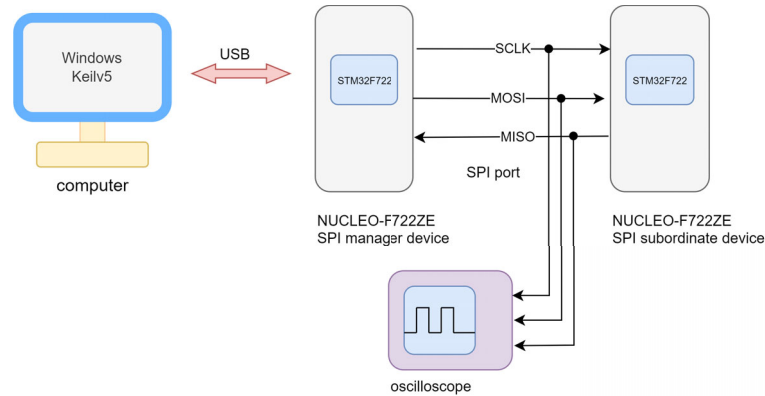


Figure 4.6: Firmware-based method configuration.

4.2.2 OpenOCD-based Boundary Scan Method Configuration

Figure 4.7 shows the configuration of the OpenOCD-based method. The configuration included a computer, two NUCLEO-F722ZE development boards, an FT2232HL board, and an oscilloscope. The FT2232 chip acted as a USB to JTAG bridge to connect OpenOCD and the manager device. The role of the oscilloscope was the same as in the firmware-based method.

4.2.3 FPGA-based Boundary Scan Method Configuration

Figure 4.8 shows the configuration of the FPGA-based method. The FPGA-based Boundary Scan method configuration included a Nexy 4 development board, two NUCLEO-F722ZE development boards, and an oscilloscope. Four pins of the Nexy 4 development board were connected to the JTAG interface of the manager device to implement Boundary Scan.

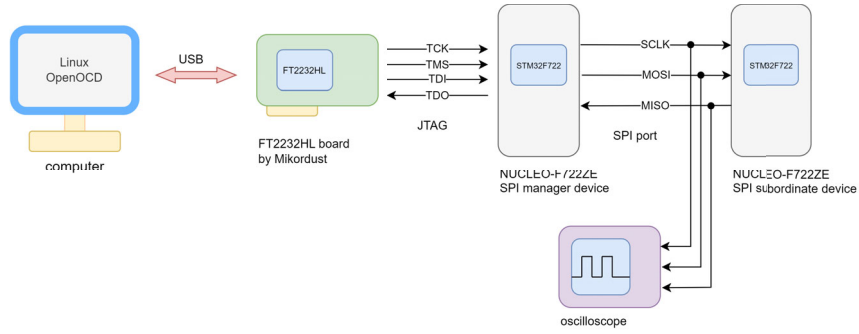


Figure 4.7: OpenOCD-based Boundary Scan method configuration.

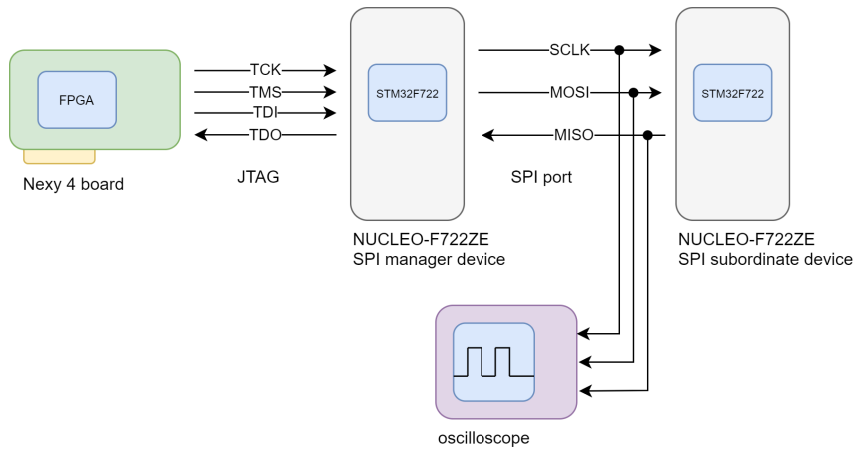


Figure 4.8: FPGA-based Boundary Scan method configuration.

4.3 Result Analysis: Communication Feasibility

In this experiment, firstly the firmware-based method was tested to receive the letter "o", which ASCII code is "01101111" from the subordinate device. The waveform measured by the oscilloscope is as shown in Figure 4.9. The data was sampled on the rising edge of the clock.

Figure 4.9: Firmware waveform of SCLK and MISO.

Next, the OpenOCD-based Boundary Scan Method was implemented. The waveform measured by the oscilloscope is shown in Figure 4.10:

Figure 4.10: OpenOCD-based waveform of SCLK and MISO.

The SPI clock signals were generated correctly by the manager device. At the same time, the subordinate device returned data accurately on the MISO line following the clock signals. Although the SCLK waveform is not stable, this shows that the regularity and stability of the CLK signal meet the requirements of the SPI communication. The signal on the MISO line is completely synchronized with the clock signals. The data content was the same as in the firmware-based method, which was "01101111". The result shows that Boundary Scan is feasible for SPI communication.

4.4 Result Analysis: Communication Speed

For the firmware-based method, the time to transfer one bit can be known from Figure 4.9, which is 300 nanoseconds. The communication speed can be calculated as:

$$Speed = \frac{1}{300 \text{ ns}} = 3333 \text{ kbps} \quad (4.1)$$

Thus, the communication speed is 3333 kilobits per second (kbps).

For the OpenOCD-based Boundary Scan method, the time to transfer one byte is illustrated from Figure 4.11, which was 576 microseconds. The communication speed can be calculated, which is 1.736 kbps.

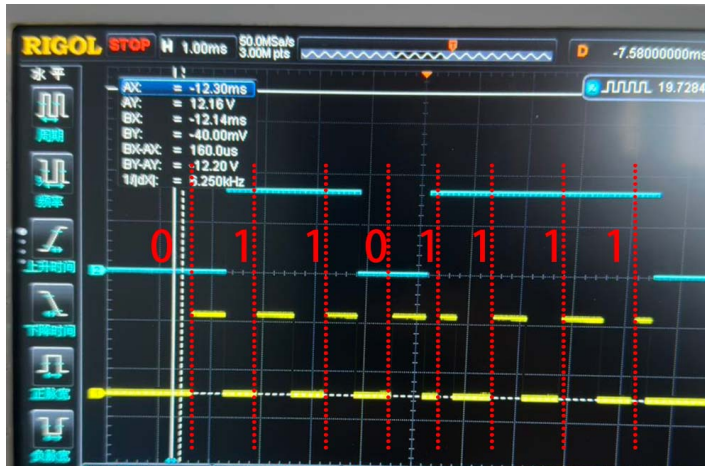


Figure 4.11: One bit transmission time of OpenOCD-based Boundary Scan method.

For the FPGA-based Boundary Scan method, the time to transfer one byte can be known from Figure 4.12, which was 41.08 microseconds. The communication speed can be calculated, which is 24.34 kbps.

Figure 4.13 shows the SPI transmission speed with these three methods. It can be observed that the firmware-based method is much faster than the Boundary Scan method. Compared with the OpenOCD-based method, the FPGA-based method is faster.



Figure 4.12: One bit transmission time of the FPGA-based method.

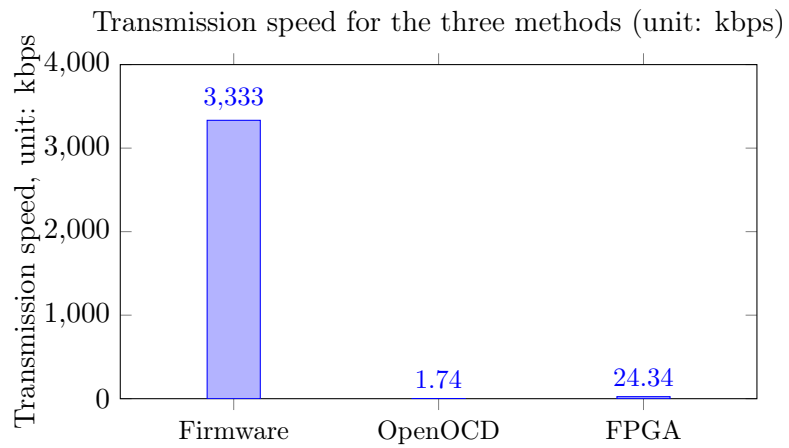


Figure 4.13: Transmission speed comparison for the three methods.

4.5 Result Analysis: Total Testing Time

Based on the discussion in Chapter 3, the formula for the total test time can be derived.

For the firmware-based method, the transmission time for one bit is 300 nanoseconds and the firmware download time obtained from the computer is 1.2 seconds. Therefore, the formula for the total test time is:

$$T_{total} = 0.0000003x + 1.2 \quad (4.2)$$

For the OpenOCD-based method, the number of Boundary Scan cells is determined by the BSDL file for the STM32F722, which is 406. According to the OpenOCD configuration, the Boundary Scan clock frequency is 1.5 MHz and the delay time per operation is 0.5 microseconds. The theoretical total test time can be calculated as follows:

$$T_{total} = 2 \times \left(\frac{406}{1500000} + 0.0000005 \right) x = 0.000542x \quad (4.3)$$

The actual transmission time for one bit measured by the oscilloscope was 576 microseconds. Therefore, the formula for the total test time is:

$$T_{total} = 0.000576x \quad (4.4)$$

The calculated theoretical time and the measured actual time are very close.

For the FPGA-based method, the number of Boundary Scan cells is determined by the BSDL file for the STM32F722, which is 406. According to the FPGA configuration, the Boundary Scan clock frequency is 20 MHz. The theoretical total test time can be calculated as follows:

$$T_{total} = 2 \times \left(\frac{406}{20000000} \right) x = 0.0000406x \quad (4.5)$$

The actual transmission time for one bit measured by the oscilloscope is 41.06 microseconds. Therefore, the formula for the total test time is:

$$f(x) = 0.00004106x \quad (4.6)$$

In Figure 4.14, the X-axis is the SPI transmission content in bits, and the Y-axis is the total test time, red represent for the OpenOCD-based method, green represent for the FPGA-based method, and the blue represent for the firmware-based method. It can be observed that the function graphs have two intersection

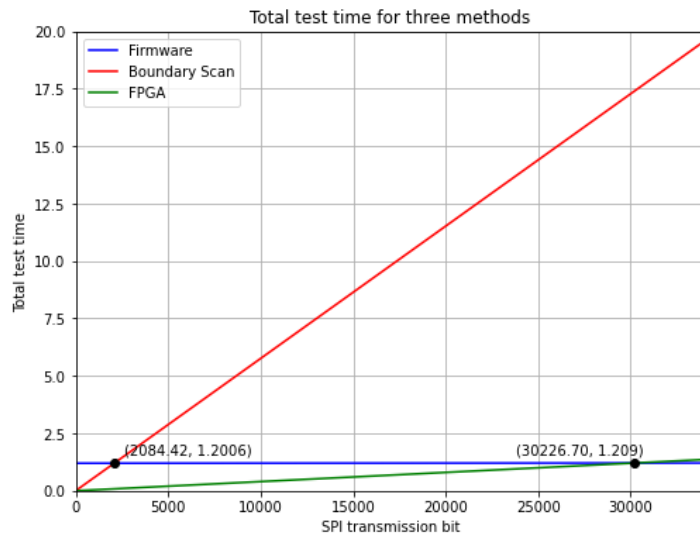


Figure 4.14: Total test time for the three methods.

points, which are at (2084.42, 1.2006) and at (30226.7, 1.209). The meaning of these two intersection points is, when the SPI transmission content is lower than 2048 bits, the OpenOCD-based method is faster than the firmware-based method, and when the SPI transmission content is lower than 30226 bits, the FPGA-based method is faster than the firmware-based method.

Figure 4.15 shows the total test time for the three methods with different test data content. It can be known that when the test content is small, the Boundary Scan-based method takes less time. When the test content grows higher, the firmware-based method is faster. The FPGA-based method is always faster of the two Boundary Scan-based methods.

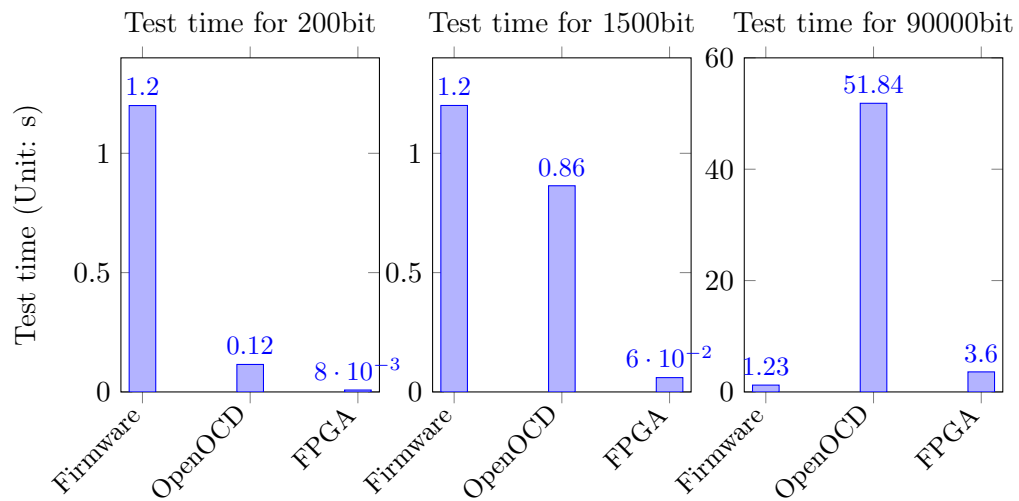


Figure 4.15: Test time comparison for different data sizes across the three methods.

Conclusion and Future Work

5.1 Conclusion

This thesis studies the feasibility of testing functional interfaces using Boundary Scan, with a focus on SPI protocol as a case study. The results show that Boundary Scan can successfully implement SPI communication. The SPI communication speed using Boundary Scan is slower than the firmware-based method. The total test time depends on the size of the test content data. For smaller sizes of data, Boundary Scan method achieves a shorter total test time compared to the firmware-based method. For larger sizes of data, the firmware-based method achieves a shorter total test time. The FPGA-based method consistently achieves less time than the OpenOCD-based method.

5.2 Evaluation of Boundary Scan and OpenOCD in FCT

5.2.1 Applicability of Boundary Scan in FCT

Boundary Scan is a testing method that directly controls the logical states of the pins, which are purely digital operations. This makes it suitable for certain FCT scenarios where only digital processes are required.

For testing tasks like pull up or pull down the pins, Boundary Scan can fully cover the tasks and provide accurate results, making it a reliable method for these scenarios.

For testing tasks involves communication protocols such as SPI and UART, Boundary Scan is able to complete the tasks. However, Boundary Scan achieves slower communication speed compared to firmware-based method, which limits its effectiveness in high-speed communication tests.

In contrast, Boundary Scan is not applicable for analog testing, such as measuring current or power consumption, as Boundary Scan is limited to digital operations.

5.2.2 Advantages and Disadvantages of Boundary Scan

In FCT, Boundary Scan has certain advantages and limitations:

Boundary Scan saves development time by not requiring testing firmware for FCT, and Boundary Scan is able to run directly onto the DUTs which significantly reduces test time. In small-scale testing, Boundary Scan method is faster than firmware-based methods.

However, for large-scale testing, Boundary Scan is significantly slower than firmware methods. Also, the application of Boundary Scan is highly dependent on chip support. If the chip does not have Boundary Scan capabilities, Boundary Scan cannot be implemented. Boundary Scan cannot handle the measurement and verification of analog signals.

5.2.3 Advantages and Disadvantages of OpenOCD

OpenOCD is an open source software that is free for individuals and companies to use and modify its source code to meet personalized needs. OpenOCD can be operated using scripts, which has a clear structure. OpenOCD provides a variety of built-in commands, which is easy to program.

However, OpenOCD and FT2232HL showed some shortcomings in terms of generating stable signals and achieving fast communication. It is observed that there was a significant delay between OpenOCD instructions, and the stability of the signals generated by the FT2232HL is low. In high-speed communication testing, OpenOCD is slower than other Boundary Scan methods such as FPGA.

5.3 Future Work

There is still potential to be explored in the application and optimization of Boundary Scan, some work is able to be done in the following areas:

First, improving the test speed. Currently, Boundary Scan may face speed bottlenecks when performing complex functional tests. To this end, we plan to research and develop new technologies, which includes exploring new hardware designs and improving existing software algorithms to speed up Boundary Scan testing.

Second, developing an integrated test platform. By combining Boundary Scan with other test methods such as AOI and ICT, we aim to create a more comprehensive test system.

Finally, we can focus on the optimization of PCB design and explore how to integrate Boundary Scan functions in the early stages of design to improve its performance. This involves early planning of PCB layout and circuit design to ensure that Boundary Scan nodes are effectively arranged, thereby achieving more efficient testing and maintenance throughout the product life cycle.

References

- [1] Grandview Research, "Consumer Electronics Market Size And Share Report, 2030," <https://www.grandviewresearch.com/press-release/global-consumer-electronics-market>
- [2] Digilent. Nexys 4 Reference Manual, Digilent Reference, 2024. <https://digilent.com/reference/programmable-logic/nexys-4/start>
- [3] STMicroelectronics. Nucleo-F722ZE Evaluation Board, STMicroelectronics, 2024. <https://www.st.com/en/evaluation-tools/nucleo-f722ze.html>
- [4] Keysight Technologies. Boundary Scan for PCB Testing, 2023. <https://www.keysight.com/us/en/library/demos/how-to-video/boundary-scan-basic-tutorial.html/>
- [5] XJTAG Technologies. Boundary Scan Testing by XJTAG, XJTAG Blog, 2023. <https://www.xjtag.com/jtag-applications/>.
- [6] Su, Yuanzhang, Xinfeng Guo, Hang Luo, Jingyuan Wang, and Zhen Liu. 2024. "A Boundary Scan Test Vectors Optimization Method Based on Improved GA-AO* Approach Considering Fault Probability Model" *Applied Sciences* 14, no. 6: 2410. <https://doi.org/10.3390/app14062410>
- [7] Khandpur, Raghbir Singh. "Printed Circuit Boards Design, Fabrication, and Assembly." 2006.
- [8] IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001), 2013, pp. 1-444. <https://standards.ieee.org/ieee/1149.1/4484/>
- [9] A. A. R. M. A. Ebayyeh and A. Mousavi, "A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry," *IEEE Access*, vol. 8, pp. 183192-183271, 2020. <https://doi.org/10.1109/ACCESS.2020.3029127>
- [10] Houdek, Cal, and Caltronics Design. "Inspection and testing methods for PCBs: An overview." *Engineer/OwnerCaltronics Design & Assembly*, no. 401, 2016.

-
- [11] Sumimoto, T., Maruyama, T., Azuma, Y., Goto, S., Mondou, M., Furukawa, N., Okada, S. "Development of image analysis for detection of defects of BGA by using X-ray images", Proceedings of the 20th IEEE Instrumentation Technology Conference (Cat. No.03CH37412), 2003, vol. 2, pp. 1131-1136.
- [12] Mikrodust. "Functional Circuit Test (FCT) Overview", Mikrodust Suite, 2024. <https://www.mikrodust.com/suite/>
- [13] NextPCB. "FCT, or Functional Circuit Test", NextPCB Blog, 2024. <https://www.nextpcb.com/blog/fct-test#:~:text=FCT%2C%20or%20Functional%20Circuit%20Test,normally%20after%20the%20assembly%20process.>
- [14] Zhuang, Chenyu. "Comparison and Selection of Commonly Used Communication Protocols in Measurement and Control Instruments" Highlights in Science, Engineering and Technology, vol. 81, Jan. 2024, pp. 540-6.
- [15] Leens, F. "An introduction to I2C and SPI protocols", IEEE Instrumentation & Measurement Magazine, 2009, vol. 12, no. 1, pp. 8-13.
- [16] Parker, K.P., Oresjo, S. "A language for describing boundary scan devices", Journal of Electronic Testing, 1991, vol. 2, pp. 43-75.
- [17] Högl, H., Rath, D. "Open on-chip debugger–openocd", Fakultät für Informatik, Tech. Rep., 2006.
- [18] Tan, Chu Jay, et al. "Review on firmware", Proceedings of the International Conference on Imaging, Signal Processing and Communication, 2017.
- [19] FTDI Chip. "FT2232H Dual High-Speed USB to Multipurpose UART/FIFO IC", FTDI Chip, 2024. <https://ftdichip.com/products/ft2232h1/>



LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2024-1030
<http://www.eit.lth.se>