

Isto Kukkohovi

## **BASEBAND TEST AUTOMATION AND STANDARDIZATION IN R&D**

# **BASEBAND TEST AUTOMATION AND STANDARDIZATION IN R&D**

Isto Kukkohovi  
Master's thesis  
Fall 2014  
Information Technology  
Oulu University of Applied Sciences

# TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikka

---

Tekijä: Isto Kukkohovi

Opinnäytetyön nimi: BaseBand Test Automation and Standardization in R&D

Työn ohjaaja: Timo Vainio

Työn valmistumislukukausi ja -vuosi: Syksy 2014

Sivumäärä:41

---

Verkko ja mobiilituotteiden rakenne voidaan karkeasti jakaa kolmeen eri osaan: Radio (RF), Signaalinkäsittely(DFE) ja Baseband(BB). Eri tuotteiden välillä RF ja DFE voivat vaihdella paljonkin, mutta BB osio on monesti hyvin samankaltainen. Tästä johtuen myös BB tuotekehityksen aikainen testaus on paljolti samanlaista eri tuotteiden ja versioiden välillä.

Uusien tuoteversioiden välinen aika on tarkoituksenmukaista minimoida, jotta valmis tuote saataisiin markkinoille ajoissa. Tämä fakta tuo aikataulullisia haasteita tuotekehityksen aikaiseen testaamiseen. Tuotteet pitäisi pystyä testaamaan nopeasti samalla säilyttäen riittävä testauskattavuus.

Työ tehtiin Elektrobit Wireless Communications Oy:lle. Heidän portfolioonsa kuuluu niin verkkolaite- kuin mobiililaitesuunnittelu. Työn tavoitteena on nopeuttaa ja helpottaa testaus suunnittelijoiden ja testaajien tehtäviä rakentamalla valmis testikirjasto tuotekehityksen aikaista testausta varten. Kirjasto sisältää perustietoja ja esimerkkejä erilaisista tuotteille tehtävistä testeistä. Sama kirjasto toimii myös apuna mittalaitehankinnoissa ja resurssien varauksissa. Kirjaston teossa hyödynnettiin SpiraTest testauksenhallintatyökalua.

Automatisoinnin tarkoituksena on tutkia miten tuotekehityksen Baseband testausta voitaisiin nopeuttaa. LabVIEW ohjelma valittiin automatisointityökaluksi johtuen lähinnä siihen valmiiksi tarjolla olevista ajureista. Automatisointi tehtiin kellosignaalien mittausta varten, jolloin oskilloskoopin säädöt eri signaaleille ja tulosten tallennus saatiin automatisoitua.

Avainsanat: Testaus, Automaatio, Standardisointi

# ABSTRACT

Oulu University of Applied Sciences  
Information Technology

---

Author: Isto Kukkohovi

Title of Bachelor's thesis: BaseBand Test Automation and Standardization

Supervisor: Timo Vainio

Term and year of completion: Fall 2014

Number of pages:41

---

Infrastructure and mobile radio devices hardware can be roughly divided in three different sections: RF (Radio Frequency), DFE (Digital Front End) and BB (Baseband). RF and DFE sections can vary between different products, but usually the BB section is very similar. Because of this the testing of the Baseband section is also similar between different products and versions. The time between different product versions is rational to minimize to keep the product schedule of a product announcement on time. This fact brings a schedule pressure to a Research and Development testing. Products should be tested faster and the test coverage should be sufficient.

This Master's thesis was commissioned by Elektrobit Wireless Communications Oy. Infrastructure and mobile radio device planning belongs to their portfolio. The aim of this Thesis was to speed up and to ease the work of test designers and test operators. It was done by creating a ready test library for the research and development phase testing. The library contains basic information and examples about different tests done to a device under test. The same library will also help in measurement instruments acquisition and resource planning. SpiraTest test management software was used in test library generation.

The purpose of the Automation is to investigate how Research and Development phase Baseband testing can be speed up. LabVIEW programming language was chosen to be used as automation tool. The main reason for its selection was the available device drivers. Clock signal quality measurement test was chosen to be automated. The oscilloscope setup for different signals and results saving was automated.

---

Keywords: Testing, Automation, Standardization

## **PREFACE**

This Master's Thesis was commissioned by Elektrobit Wireless Communications Ltd. Oulu office. Tutoring teacher was Timo Vainio from Oulu University of Applied Sciences. Mikko Koutonen was acting as a tutoring colleague.

## **TABLE OF CONTENTS**

<b>TIIVISTELMÄ</b>	<b>3</b>
<b>ABSTRACT</b>	<b>4</b>
<b>PREFACE</b>	<b>5</b>
<b>ABBREVIATIONS</b>	<b>9</b>
<b>1 INTRODUCTION</b>	<b>10</b>
<b>2 THE WORK ENVIRONMENT</b>	<b>12</b>
<b>3 THEORY AND TERMS</b>	<b>13</b>
<b>3.1 Test definitions</b>	<b>13</b>
3.1.1 Production tests	13
3.1.1 Type approval tests	13
3.1.2 Research and Development tests	14
3.1.3 Measurement uncertainty	15
<b>3.2 Documents</b>	<b>15</b>
<b>3.3 Measurement instruments automation</b>	<b>16</b>
<b>3.4 Clock lines</b>	<b>17</b>
<b>3.5 Rise and Fall time</b>	<b>18</b>
<b>3.6 Jitter</b>	<b>19</b>
<b>3.7 Serial Data</b>	<b>20</b>
<b>3.8 Eye Diagram Measurements</b>	<b>21</b>

3.9	Compliance Tests	22
3.10	Test Points	23
3.11	Self-tests	23
4	STANDARDIZATION	23
4.1	SpiraTest	24
4.2	Test library	25
4.3	Standardization Implementation	25
5	AUTOMATION	26
5.1	Test Case for Automation	26
5.2	Clock signal Quality	26
6	AUTOMATION IMPLEMENTATION	27
6.1	Driver Setup	28
6.2	Front Panel	29
6.3	Block Diagram	30
6.3.1	Setup	31
6.3.2	First Measurement and vertical adjust	31
6.3.3	Horizontal adjust	32
6.3.4	Measurements	33
6.3.5	Saving of data	34
6.3.6	Save waveform	34
6.3.7	Testing	35
7	POSSIBILITIES OF FURTHER DEVELOPMENT	39

<b>8</b>	<b>CONCLUSION</b>	<b>40</b>
<b>9</b>	<b>LIST OF REFERENCES</b>	<b>41</b>



## ABBREVIATIONS

BB	Baseband
BIST	Build In Self-Test
CLK, CK	Clock
DDR	Double Data Rate
DFE	Digital Front End
DFT	Design For Test
DUT	Device Under Test
FMEA	Failure Modes and Effects Analysis
GPIB	General Purpose Interface Bus
GPIB	General Purpose Interface Bus
I2c	Inter-Integrated Circuit
IVI	Interchangeable Virtual Instruments
LAN	Local Area Network
LXI	LAN eXtensions for Instrumentation
MAX	Measurement & Automation Explorer
R&D	Research and Development
RF	Radio Frequency
SGMII	Serial Gigabit Media Independent Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VI	Virtual Instrument

# 1 INTRODUCTION

Radio devices (Infrastructure and mobile) hardware can be usually divided in three different sections: RF (Radio Frequency), DFE (Digital Front End) and BB (Baseband). Baseband is the section that contains all the basic functionality that will allow RF and DFE to work. Usually this means that the Baseband section is very similar to different products and devices.

The main aim of the research and development testing is to secure that the product fulfills standards and specifications needed. Compared to a Production testing, where only the quality of production process is verified, a test operator must have a good understanding of what he/she is doing. The aim of this Master's thesis was to study how the R&D phase baseband testing could be automated and standardised. The thesis will explain what test cases should be done and how and can they be automated at some level. Automation was focused on test cases that are laborious and simple at the same time. The thesis can also be used as an Introduction to the R&D HW testing for new test Engineers.

The standardization part of the thesis was to gather a library about the test cases and information on how the testing should be done. The test cases included basic functional tests and a signal quality test. Some of the quality tests were done using ready compliance test packages from measurement device manufacturers.

Baseband testing standardization and automation could improve test coverage, test time and test yield. This is a very important issue because usually the test time is very limited between new proto versions. Possible design errors and device malfunctions should be found as soon as possible to minimize costs. A ready library for test cases makes testing of a new product and test planning a lot easier. Measurement devices and their acquisition are easier to control if all the test cases are known.

The automation of measurement devices and result gathering was done using a LabVIEW graphical programming language. A DUT internal testing was done using scripts that can be run in its operating system. Depending on the product capabilities, it could include basic internal bus

testing (for example i2c, SGMII, SPI etc.), voltage testing, external communication testing (for example UART, USB and Ethernet).

## 2 THE WORK ENVIRONMENT

The work was done for Elektrobit Wireless Communication research and development department. The Wireless Business Segment product portfolio includes products and product platforms for defense and public safety markets and also for industrial use. Product development services and customized solutions in wireless communications segment were also provided by EB. (Elektrobit Corporation, 2014. Date of retrieval 16.3.2014).

The thesis work was done in close relationship between BB HW design and BB HW testing teams. The testing in R&D department was mainly done manually using e.g. a low and high end oscilloscope, DMM, counter, network analyzers.

SpiraTest was chosen for a tool to build a test library. SpiraTest provides a complete Quality Assurance solution that manages requirements, tests, bugs and issues in one environment, with a complete traceability from inception to completion. (Inflectra,2014. Date of retrieval 10.04.2014)

LabVIEW is a system-design platform and development environment for a visual programming language from National Instruments. It was chosen as an automation implementation tool, because it has good support for different measurement instruments. (National Instruments, 2013. Date of retrieval 10.12.2013).

### **3 THEORY AND TERMS**

When talking about test terms and abbreviations the testing field is full of them. Some of the most important ones are explained in this chapter. Many times there is not only one right way to define things, but one possible way is explained.

#### **3.1 Test definitions**

Testing has different goals when a product is in a different life cycle. At the early stage R&D tests are done to verify the design of the product. Type approval tests are done to ensure that the product meets a minimum set of regulatory, technical and safety requirements. Production tests are done to verify manufacturing quality and the functionality of the product.

##### **3.1.1 Production tests**

The main aim of the production testing is to make sure that the assembly is correctly fabricated. All the needed components must have right values and they are located in the right place at the right order. Basically the correlation between design and the manufactured device is confirmed.

Tests must be done using a minimum test time and with necessary test coverage. At the early stage of the production the test coverage must be as wide as possible. After the manufacturing process has been verified then the unnecessary tests can be removed to speed up the test time.

##### **3.1.1 Type approval tests**

Before a product is released, appropriate approvals must be obtained. This procedure is called type approval tests. The approval requirements may vary from country to country.

Type approval tests compound an approving process and a technical testing. When considering telecommunication equipment, the TA system must guarantee the following things:

- Protection of the consumer against any other effects dangerous to health.
- Opening of frequency bands for concurrent access according to international agreements to provide the basis for technological progress.
- Guarantee of the basic quality of service by limiting the potential of interferences.

- Overall control of the telecommunication market.
- Assurance of a minimum product quality and functionality.
- Promotion of an industry self-regulation by setting the legal boundaries

(Hassan M., Hassan Y., Maryam F. 2007. Date of retrieval 18.3.2014)

### **3.1.2 Research and Development tests**

R&D Tests and measurements are done to verify that product functions exactly as designed in different conditions. Test cases were in this thesis roughly divided into two different groups: Functional and Quality. These tests can be done in different environment conditions to confirm the functionality of the product also in its life time.

In the Functional test DUT, usually the power is connected and possibly also the software is running. Usually an input is given to the DUT and the output is measured. This can e.g. be a test where a device is powered up and voltage rails are measured or an IO is set to change its state and it is verified by measuring the IO. The basic outcome of the tests is to confirm that the functions of the product are working but it does not give information on how well the product is working.

Quality tests are tests that confirm that the signal properties match with the standards and specifications needed. These can e.g. be eye patterns, rise and fall times, setup and hold times and jitter. Quality tests include a compliance or conformance test that basically verifies that a device meets needed standards. Measurement instrument suppliers have done ready applications for them, e.g. DDR, Ethernet, jitter and USB. Some of the Quality tests can be simple but laborious to do especially if they need statistical information for the test case to pass. These tests are e.g. clock signal, timing and synchronization measurements.

Reliability tests are done to ensure the quality of the product in its expected life time. Reliability test include High/Low Temperature, Temperature Cycling, Vibration, Drop, Shock, Crush, Humidity and Altitude tests. Different reliability test cases can be found when using Failure Modes and Effects Analysis (FMEA) process in the reliability test planning.

(Mike Silverman. 2010. Date of retrieval 20.3.2014.)

### **3.1.3 Measurement uncertainty**

When the result of a measurement is reported the Quality of the measurement should be given so that those who use it can assess its reliability. If this information is not given the comparison between different measurement occasions or referenced values cannot be done.

In a measurement there are many possible sources of uncertainty including the following ones: incomplete definition of the measurand, an imperfect realization of the definition of the measurand, an nonrepresentative sampling — the sample measured may not represent the defined measurand, an inadequate knowledge of the effects of environmental conditions on the measurement or an imperfect measurement of environmental conditions, personal bias in reading analogue instruments, a finite instrument resolution or a discrimination threshold, inexact values of measurement standards and reference materials, inexact values of constants and other parameters obtained from external sources and used in the data-reduction algorithm, approximations and assumptions incorporated in the measurement method and procedure, variations in repeated observations of the measurand under apparently identical conditions.

(GUM:1995. ISO/CEI GUIDE 98-3)

## **3.2 Documents**

One of the most important things of the testing is the Documentation. All of the tests that are designed to be done to DUT must be clearly explained in the documents. Outputs of the test cases must also be documented so that the traceability between different test operators, HW and SW versions and the measurement environment can easily be done.

A test specification defines specific test cases, which are executed in the validation process. It should consist of the following parts: Revision, Feature Description and detailed test cases. It may also include Test Case Priorities.

A test Plan is usually derived from the Product Description, which includes basic information on the product. It is used as the basis when testing any product or software. The test Plan will include at least the following information: DUT Identification information, Features to be tested,

Features not to be tested, Schedule, Test Environment, who will do the testing and Test deliverables. The test plan defines how the tests are done. (Elektrobit Corporation, 2013. Date of retrieval 1.11.2013).

A detailed procedure that fully tests a feature or an aspect of a feature is called a test case. The test case should include the following information:

- Test ID: The test case unique ID
- Node List: The list of the actual hardware being tested.
- Test Description: The test case description.
- Test Suite: If applicable, include the feature that this test case will be used to verify.
- Test Setup: The test setup clearly describes the topology, hardware, logical configurations, test tools, applications, or other prerequisites that must be in place before the test can be executed.
- Test Steps: Detailed step-by-step instructions on how to carry out the test.
- Expected Results: Describe what the system must give as output or how the system must react based on the test steps.
- Observed Results.
- Pass/Fail information.

(OneStopTesting.com, 2013. Date of retrieval 1.11.2013)

### **3.3 Measurement instruments automation**

Measurement instruments usually have a communication bus that can be used to control an instrument and acquire data from it. The bus can be a serial port, a parallel port, GPIB, USB or LAN/LXI depending on the instrument properties.

LabVIEW was chosen to implement automation. It has a support for all of the needed buses and also a support for IVI (Interchangeable Virtual Instruments) drivers which is very convenient when different models of instruments are used. (National Instruments, 2013. Date of retrieval 10.12.2013).



LabVIEW is a graphical programming platform that enables to build of applications in less time than traditional programming languages. It is commonly used in instrumentation automation and it has good support in instrumentation drivers. See FIGURE 1. Simple LabVIEW program.

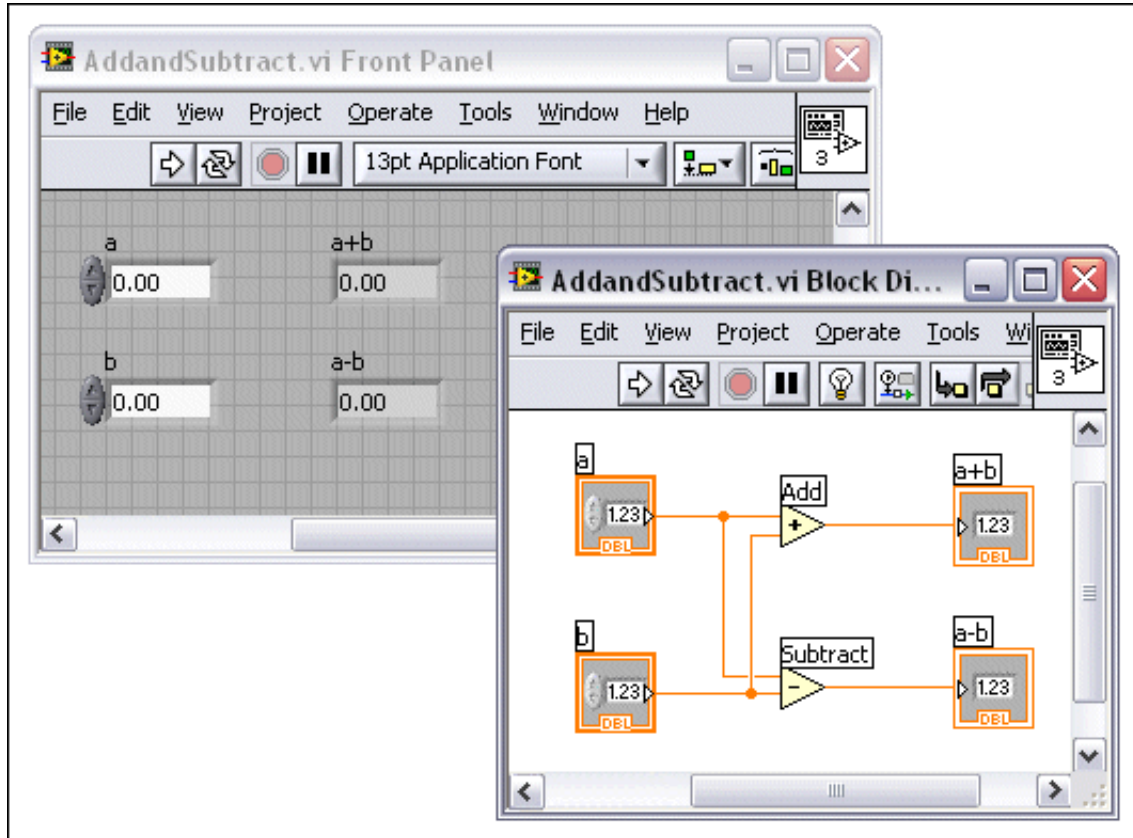


FIGURE 1. Simple LabVIEW program (National Instruments, 2014. Date of retrieval 20.10.2014)

### 3.4 Clock lines

Clock signals are needed in almost every integrated circuit or electrical system.

In today's world digital data transmission is done at higher and higher speeds and at the same time a higher resolution and data rate in an analog to digital conversion is used. These facts mean that the clock signal quality must have a special attention when design and testing is done.

(Maxim Integrated, 2013. Date of retrieval 10.12.2013).

Clock signals keep the internal timing and synchronization in shape. Clock signal quality is usually described as jitter measurements, which basically means timing error measurements. Clock signals can be very fast (see TABLE 1. DDR Standards and CLK speeds) and this leads to the fact that measurement points and measurement instruments must be carefully chosen.

TABLE 1. DDR Standards and CLK speeds (Poikola, 2012.)

Standard name	Memory clock	Cycle time	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR3-800	100 MHz	10 ns	400 MHz	800 Million	PC3-6400	<a href="#">6400 MB/s</a>
DDR3-1066	133 MHz	7.5 ns	533 MHz	1066 Million	PC3-8500	8533 MB/s
DDR3-1333	166 MHz	6 ns	667 MHz	1333 Million	PC3-10600	<a href="#">10667 MB/s[1]</a>
DDR3-1600	200 MHz	5 ns	800 MHz	1600 Million	PC3-12800	12800 MB/s

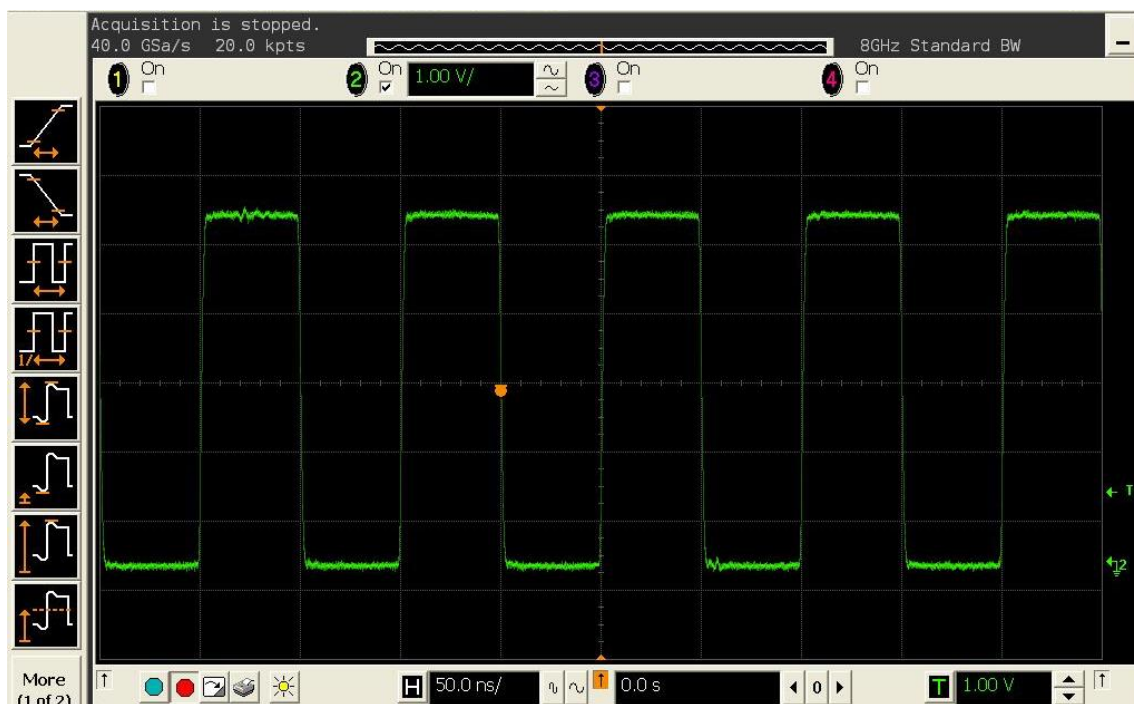


FIGURE 2. 10MHz CLK signal

### 3.5 Rise and Fall time

A Rise and Fall time analysis needs to have statistical data to make correct measurements. A histogram of a waveform that transitions in two states contains two peaks. Two clusters that contain the largest data density will be identified in the analysis. Top and Base reference levels will be computed from this data.

(Teledyne Technologies, 2013. Date of retrieval 10.12.2013).

After the Top and Base levels are estimated, the Rise and Fall times can be calculated. Base and Top levels are subdivided into a percentile scale where Base is zero and Top is 100 percent. The wanted threshold levels are then used to determine the vertical position of the crossing points. The estimated time interval between these points (see FIGURE 3. Rise- and Fall time of a pulse pattern) on the rising or falling edges is the Rise or Fall time. (Zsolt, P. 2004. Date of retrieval 12.12.2013)

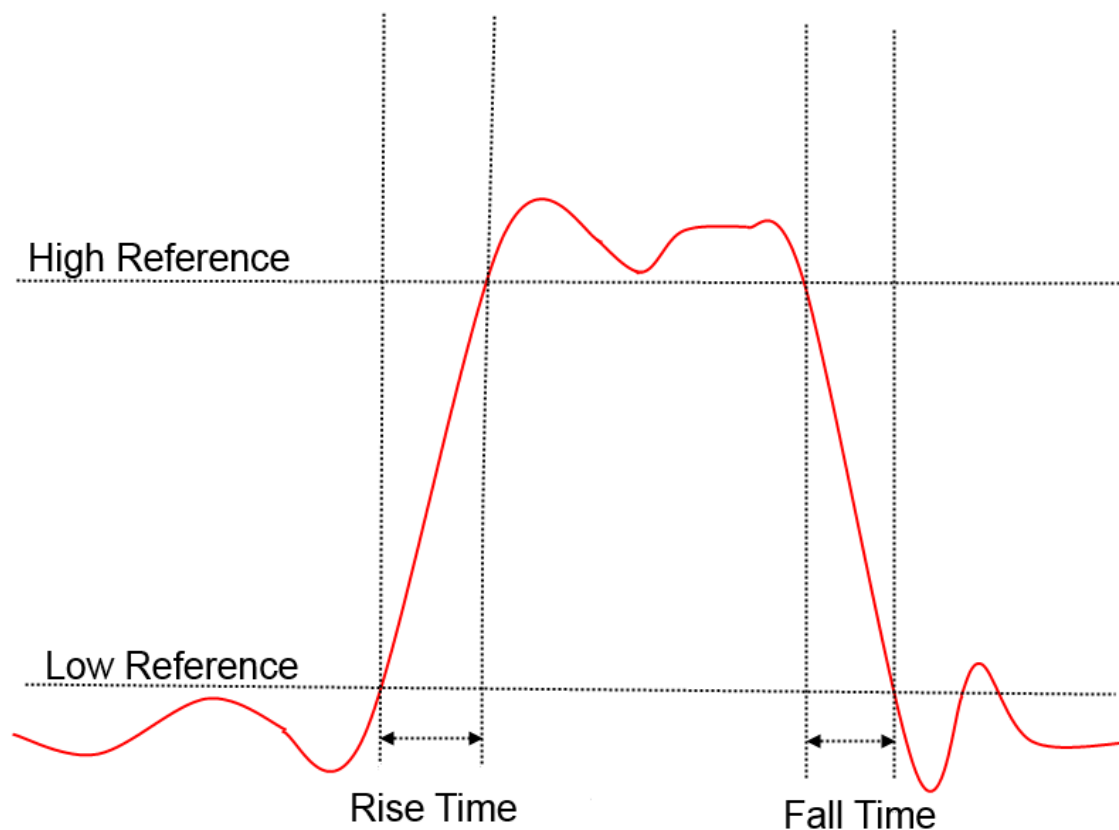


FIGURE 3. Rise- and Fall time of a pulse pattern

### 3.6 Jitter

A jitter can be defined as an undesired deviation from a true periodicity of an assumed periodic signal and often in relation to a reference clock source. A clock jitter can be separated in three different classes, see FIGURE 4. Jitter Classes.

(Wikipedia, 2013a. Date of retrieval 12.12.2013).

When talking about serial data communication the jitter has a key role in the clock extraction and network timing. Usually a data clock is not transmitted explicitly with the data. (Kossel, M.A, - Schmatz, M.L. 2004. 536 – 543).

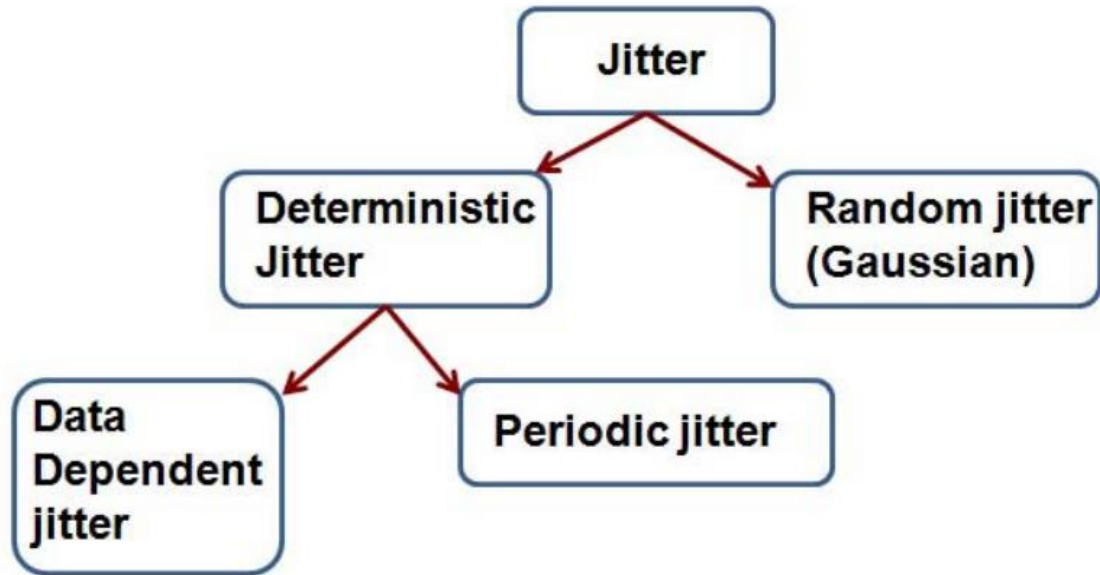


FIGURE 4. Jitter Classes (SILICON LABS, 2012. Date of retrieval 10.12.2013)

### 3.7 Serial Data

A serial transmission is increasingly used for the data transmission. It is used in almost every up-to-date communication networks. Applications like office communication computer networks, fieldbus systems in process, building and manufacturing automation, Internet and ISDN use Serial Data transmission.

A single transmission line is used to send one bit after another and this is called a serial data transmission. Microprocessors process data in a bit-parallel mode and it means that the data must be converted from a parallel to a serial mode for data transmission and vice versa. See FIGURE 5. Example of Serial Data.

The conversion between serial data and parallel data is done by special transmitter and receiver modules that are commercially available for different network types.

The upsides of this technology are extremely high data rates, reduction in installation costs and effort and user-friendliness. The downside is that it needs some time for coding and decoding the data. (SAMSON AG, 2013. Date of retrieval 10.12.2013).

The quality of the serial data can be measured using eye-diagram plots and a bit-error rate (BER) measurements.

(Kossel, M.A and Schmatz, M.L. 2004, 536 – 543)

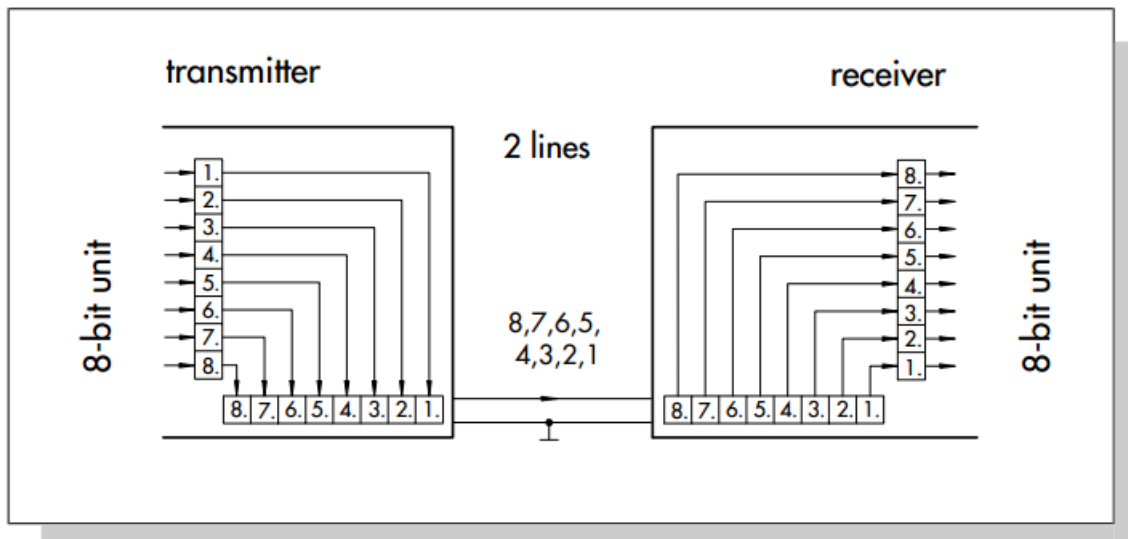


FIGURE 5. Example of Serial Data conversion (SAMSON AG, 2013. Date of retrieval 10.12.2013)

### 3.8 Eye Diagram Measurements

” An eye diagram is a common indicator of the quality of signals in high-speed digital transmissions. An oscilloscope generates an eye diagram by overlaying sweeps of different segments of a long data stream driven by a master clock. The triggering edge may be positive or negative, but the displayed pulse that appears after a delay period may go either way; there is no way of knowing beforehand the value of an arbitrary bit. Therefore, when many such transitions have been overlaid, positive and negative pulses are superimposed on each other. Overlaying

many bits produces an eye diagram, called so because the resulting image looks like the opening of an eye". (Behera, D. Varshney, D. Srivastava, S. and Tiwari, S. 2011. Date of retrieval 20.5.2014.)

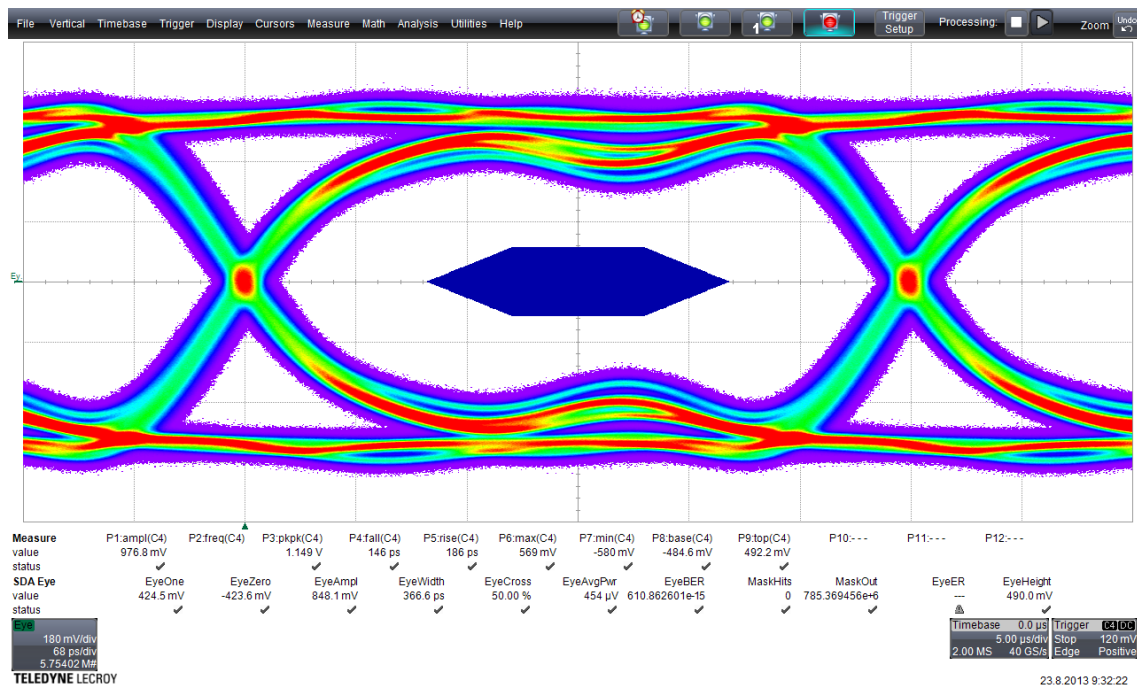


FIGURE 6. Eye Diagram

### 3.9 Compliance Tests

Conformance or compliance testing is done to determine whether a product or system meets a specified standard that it has been developed for. In this thesis compliance tests are understood as separate tests done to verify different technologies against standards.

Many test instrument manufacturers have done applications to automatically test the DUT against different standards. To manually verify this would be very time consuming because of the number of the measured parameters.

### **3.10 Test Points**

A test point selection is a very crucial matter especially in quality testing. Fast signals can have very different signal properties between a driver and a receiver end. The optimal test point is usually as close as possible in the receiver end. Sometimes this can be hard to accomplish because of the DUT layout or a poor DFT.

Sometimes the test point at the receiver side chip pin is not close enough. In this case a comparison with simulation results must be taken into use. A simulation should give us information what the signal shape should look like in the measured test point and it should match the measured shape. This is the only way we can say that the signal quality is OK in the receiver chip die.

### **3.11 Self-tests**

BIST (Built In Self-Test) is a more and more common way to do a fast functional testing with complex devices. BIST basically means that the device tests its functions by itself, for example writes a data pattern to memory and then verifies it. The benefits in using BIST are speed, repeatability, no need for external measurement instruments or connectivity to test points. The test has no external hardware connected to the DUT that could interfere with the test results.

On the down side the self-test needs some coding and maintenance. Code needs some place to store and it will grow the needed storage space. It is also a possible security threat; if the coding is done poorly it can leave some way to hack the device.

## **4 STANDARDIZATION**

Usually, when a new project is started, a huge work must be done in the test planning. Test tools, resources, documentation, etc. must be done. A ready test case library would help this laborious work. The goal of standardization is also to help a test engineer in testing and interpreting the results. Spiratest was an application that was already available in EB and it was chosen as a tool for building a test library.

## 4.1 SpiraTest

SpiraTest is a complete quality assurance and test management system and it can also be used for requirements management, see FIGURE 7. SpiraTest requirements management. Requirements can be mapped to one or more test cases that can be used to validate that your Product functionality works as expected.

The screenshot shows the SpiraTest Requirements Management interface. The left sidebar contains a tree view with categories like 'My Filters', 'Shared Filters', 'Components', and 'Releases'. The main table displays 15 requirements for the 'Library Information System' project. The table columns include Name, Test Coverage, Importance, Status, Release, Type, ID, and Edit. The requirements are organized into hierarchical groups such as 'Functional System Requirements', 'Book Management', 'Edition Management', and 'Author Management'. Each requirement has a corresponding test coverage bar and a status indicator.

Name	Test Coverage	Importance	Status	Release	Type	ID	Edit
Functional System Requirements	100%	1 - Critical	In Progress		Package	RQ:000001	Edit
Online Library Management System	100%	1 - Critical	In Progress		Package	RQ:000002	Edit
Book Management	100%	1 - Critical	Developed		Package	RQ:000003	Edit
Ability to add new books to the system	100%	1 - Critical	Developed	1.0.0.0.0001	Feature	RQ:000004	Edit
Ability to edit existing books in the system	100%	1 - Critical	Developed	1.0.0.0.0001	Feature	RQ:000005	Edit
Ability to delete existing books in the system	100%	1 - Critical	Developed	1.0.0.0.0002	Feature	RQ:000006	Edit
Ability to associate books with different subjects	100%	1 - Critical	Developed	1.1.0.0.0001	Feature	RQ:000007	Edit
Ability to associate books with different authors	100%	1 - Critical	Developed	1.1.0.0.0001	Feature	RQ:000008	Edit
Ability to associate books with different editions	100%	1 - Critical	Developed	1.1.0.0.0002	Feature	RQ:000009	Edit
Ability to completely erase all books stored in the system	100%	1 - Critical	Developed	1.2.0.0	Feature	RQ:000010	Edit
Edition Management	Not Covered	1 - Critical	In Progress		Package	RQ:000011	Edit
Ability to create different editions	Not Covered	1 - Critical	In Progress	1.0.0.0.0003	Feature	RQ:000012	Edit
Author Management	100%	2 - High	In Progress		Package	RQ:000013	Edit
Ability to add new authors to the system	100%	2 - High	Planned	1.0.0.0	Package	RQ:000014	Edit
Ability to edit existing authors in the system	100%	2 - High	Planned	1.0.0.0.0002	Feature	RQ:000015	Edit

FIGURE 7. SpiraTest requirements management (Inflectra Corporation, 2014. Date of retrieval 20.04.2014)

The focus of the software is in software testing but it can also be used in HW the verification. It has many features but in this thesis it is only used to create a test library, see FIGURE 8. SpiraTest Test Case Management. More information about the software can be found from Inflectra web pages. (Inflectra Corporation, 2014. Date of retrieval 20.04.2014).

The screenshot shows the SpiraTest Test Case Management interface. The left sidebar contains a tree view with categories like 'Test Cases', 'Test Sets', 'Test Runs', and 'Automation Hosts'. The main table displays 10 test cases for the 'Library Information System' project. The table columns include Name, Execution Status, Owner, Last Executed, Author, Active, Last Updated, ID, and Edit. The test cases are organized into hierarchical groups such as 'Functional Tests (5)', 'Regression Tests (2)', 'Scenario Tests (2)', and 'Exception Scenario Tests (1)'. Each test case has a corresponding execution status bar and a last executed date.

Name	Execution Status	Owner	Last Executed	Author	Active	Last Updated	ID	Edit
Functional Tests (5)	Failed	Fred Bloggs	30-Nov-2003	Fred Bloggs	Yes	30-Nov-2003	TC000001	Edit
Ability to create new book	Failed	Fred Bloggs	8-Jan-2013	Fred Bloggs	Yes	10-Jan-2013	TC000002	Edit
Ability to edit existing book	Failed	Fred Bloggs	16-Jan-2013	Fred Bloggs	Yes	16-Jan-2013	TC000003	Edit
Ability to create new author	Passed	Joe P Smith	28-Jan-2013	Fred Bloggs	Yes	30-Nov-2003	TC000004	Edit
Ability to edit existing author	Blocked	Joe P Smith	30-Nov-2003	Fred Bloggs	Yes	30-Nov-2003	TC000005	Edit
Ability to reassign book to different author	Passed	Joe P Smith	30-Nov-2003	Fred Bloggs	Yes	30-Nov-2003	TC000006	Edit
Regression Tests (2)	Passed	Joe P Smith	30-Nov-2003	Joe P Smith	Yes	30-Nov-2003	TC000007	Edit
Book management	Passed	Joe P Smith	30-Nov-2003	Joe P Smith	Yes	30-Nov-2003	TC000008	Edit
Author management	Passed	Joe P Smith	30-Nov-2003	Joe P Smith	Yes	30-Nov-2003	TC000009	Edit
Scenario Tests (2)	Failed	Joe P Smith	28-Jan-2013	Joe P Smith	Yes	30-Nov-2003	TC000010	Edit
Exception Scenario Tests (1)	Failed	Joe P Smith	28-Jan-2013	Joe P Smith	Yes	30-Nov-2003	TC000011	Edit
Person loses book and needs to report loss	Failed	Joe P Smith	28-Jan-2013	Joe P Smith	Yes	30-Nov-2003	TC000012	Edit
Adding new book and author to library	Not Run	Joe P Smith	30-Nov-2003	Joe P Smith	Yes	30-Nov-2003	TC000013	Edit
Common Tests (2)	Passed	Joe P Smith	30-Nov-2003	Fred Bloggs	Yes	30-Nov-2003	TC000015	Edit
Open Up Web Browser	Passed	Joe P Smith	30-Nov-2003	Fred Bloggs	Yes	30-Nov-2003	TC000016	Edit

FIGURE 8. SpiraTest Test Case Management (Inflectra Corporation, 2014. Date of retrieval 20.04.2014)

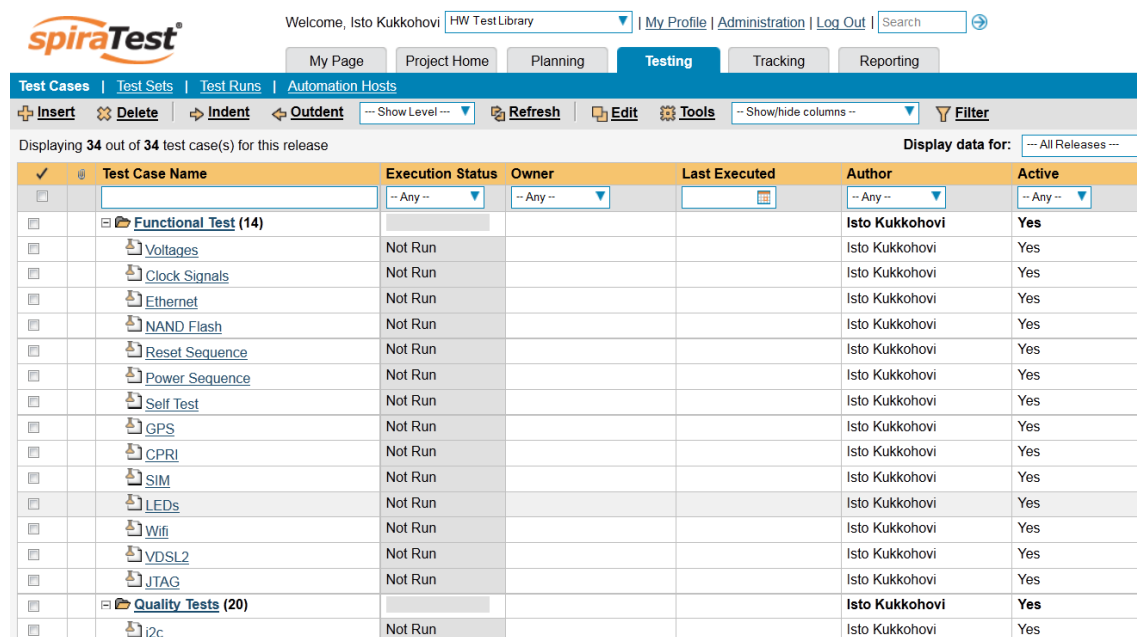


## 4.2 Test library

A test Library was done for the company's internal HW testing use for different projects. It contains information about the test cases that could be done for the DUT and basic information on how the tests should be done. An example test case is done for most common buses / protocols and it will include information on how the test limits should be decided for each device.

## 4.3 Standardization Implementation

The test library was done to an Elektrobit server, see FIGURE 9. SpiraTest library. It includes 14 functional tests and 20 quality tests. Some of the test cases include a template for test steps and test limits.



The screenshot shows the SpiraTest web application interface. At the top, there is a navigation bar with the SpiraTest logo and user information: "Welcome, Isto Kukkohovi" and "HW Test Library". Below this, there are tabs for "My Page", "Project Home", "Planning", "Testing" (selected), "Tracking", and "Reporting". The "Testing" tab is active, showing a list of test cases. The interface includes a toolbar with buttons for "Insert", "Delete", "Indent", "Outdent", "Refresh", "Edit", and "Tools". A dropdown menu for "Show Level" is also present. The main content area displays a table of test cases, with a header row indicating "Displaying 34 out of 34 test case(s) for this release". The table has columns for "Test Case Name", "Execution Status", "Owner", "Last Executed", "Author", and "Active". The test cases are grouped into "Functional Test (14)" and "Quality Tests (20)".

✓	Test Case Name	Execution Status	Owner	Last Executed	Author	Active
<input type="checkbox"/>	<input type="text"/>	-- Any --	-- Any --		-- Any --	-- Any --
<input type="checkbox"/>	<input type="checkbox"/> Functional Test (14)				Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Voltages	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Clock Signals	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Ethernet	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> NAND Flash	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Reset Sequence	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Power Sequence	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Self Test	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> GPS	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> CPRI	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> SIM	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> LEDs	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Wifi	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> VDSL2	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> JTAG	Not Run			Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> Quality Tests (20)				Isto Kukkohovi	Yes
<input type="checkbox"/>	<input type="checkbox"/> i2c	Not Run			Isto Kukkohovi	Yes

FIGURE 9. SpiraTest library

Complete information about the library cannot be published in this thesis. FIGURE 10. i2c measurement shows an example of what is expected to be measure from an i2c line according to the library. The library also includes information on what kind of test procedure must be explained in the test specification, for example "Connect a PC via Ethernet cable to the DUT, use TeraTERM in the test PC and product SW to communicate, Check that I2C peripherals can communicate by using self-tests and test commands/methods listed in table 12".

Bus name	Signal	Test point	Vmin	Req.	Vlow	Req.	Vhigh	Req.	Vmax	Req.	Trise	Req.	Tfall	Req.	Setup time	Req.	Hold time	Req.	PASS/FAIL
I2C_D123_1V8	DATA	R2223																	
	CLK	R2233																	

FIGURE 10. i2c measurement

## 5 AUTOMATION

When a new product is developed, it usually means that many prototypes are done before the final product is ready. This leads to a fact that many of the laborious measurements are done over and over again. Automation improves the measurement repeatability, minimizes the testing time and the measurement uncertainty caused by a test operator.

### 5.1 Test Case for Automation

The chosen test case was a Clock signal measurement using an Oscilloscope. The test case is quite easy to perform, but automation has some additional value to add to this case in the form of reliability and reproducibility.

Different Clock signals exist in almost all of the modern day products. A Clock signal is a very crucial signal of the proper functionality of the product in its full life time in different climate conditions. Clock signal quality measurements are laborious because of the amount of signals and properties. Automaton could help the tester to setup oscilloscope properties for each clock signal and save the test results. This will lead to more robust measurements and an easier result interpretation.

### 5.2 Clock signal Quality

An Oscilloscope Auto Setup function is used to receiver signal parameters. These parameters are used to configure an oscilloscope setup for a clock quality measurement. For example, reference levels, number of measurements, channel, probe attenuation parameters are given in front panel.

After the setup is done, measurements are automatically done as many times as the number of measurements defines. The measurements are Frequency, Rise time, Fall time, Voltage min,

Voltage max, Voltage low, Voltage high, Duty Cycle Positive, Duty Cycle Negative, Overshoot and Preshoot (see FIGURE 11. Clock signal measurements). The measurements can easily be added because a state machine is used in the measurement VI design.

After the measurements defining the signal parameters are done, a new measurement is automatically done to get a waveform of the signal. The waveform must be acquired to check the signal pattern and to make sure that the oscilloscope setup is proper. Finally, the waveform and the measurement results are saved to a computer hard drive.

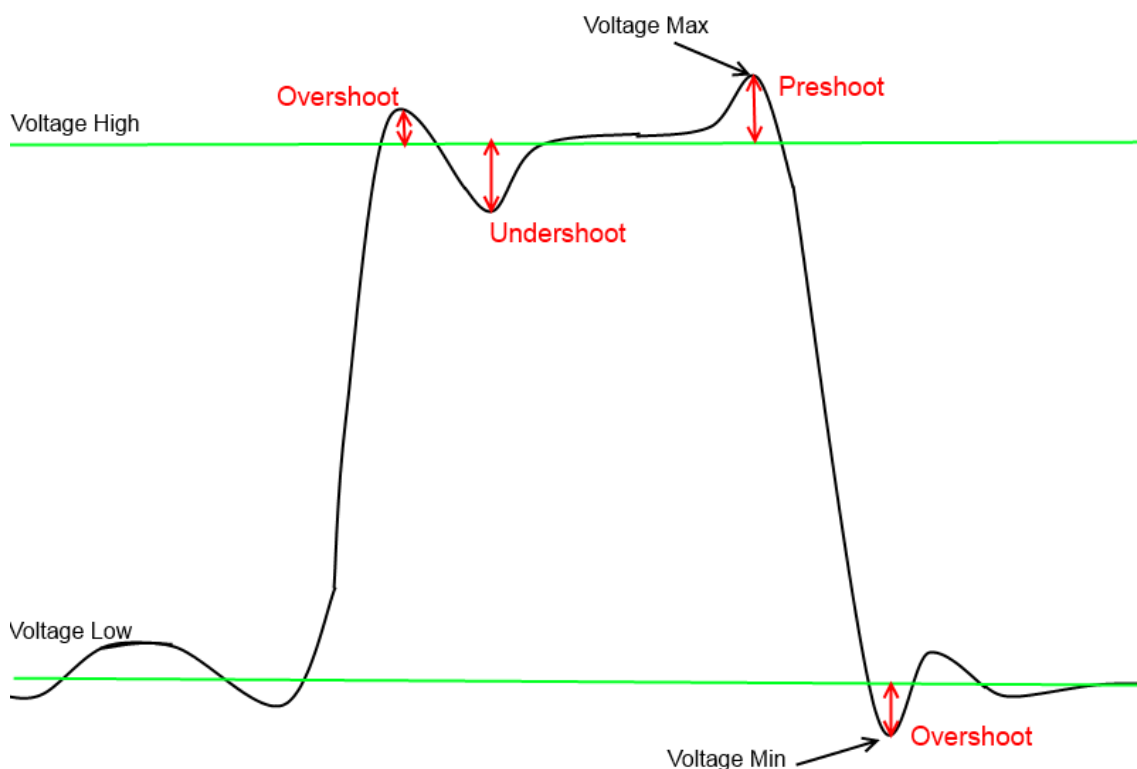


FIGURE 11. Clock signal measurements

## 6 AUTOMATION IMPLEMENTATION

In this chapter the detailed information about the automation implementation will be explained. The Oscilloscope was automated to do clock signal quality measurements and also to save the result of the measurements to a text file.

The test environment included the DUT, PSU, power & Ethernet cable, NI GPIB-USB-HS GPIB Controller, PC and Agilent Infiniium 9000 Series oscilloscope. IVI drivers were chosen for the Oscilloscope control and LabVIEW is the programming environment.

## 6.1 Driver Setup

IVI and GPIB Controller drivers needed to be installed to the computer and a proper setting had to be done in MAX (Measurement & Automation Explorer) before the oscilloscope automation could be done. MAX is a software provided by National Instruments that allows configuring and controlling National Instruments' hardware and software.

After the IVI drivers were installed for the used oscilloscope a logical name had to be given for it. In this case a logical name was Skooppi. LabVIEW uses an oscilloscope's logical name when controlling the oscilloscope. The link between drivers and logical name is done in MAX, see FIGURE 12. IVI driver logical name. In Driver session different Initialization options can be configured and also instrument simulation can be activated if needed. See FIGURE 13. Driver session.

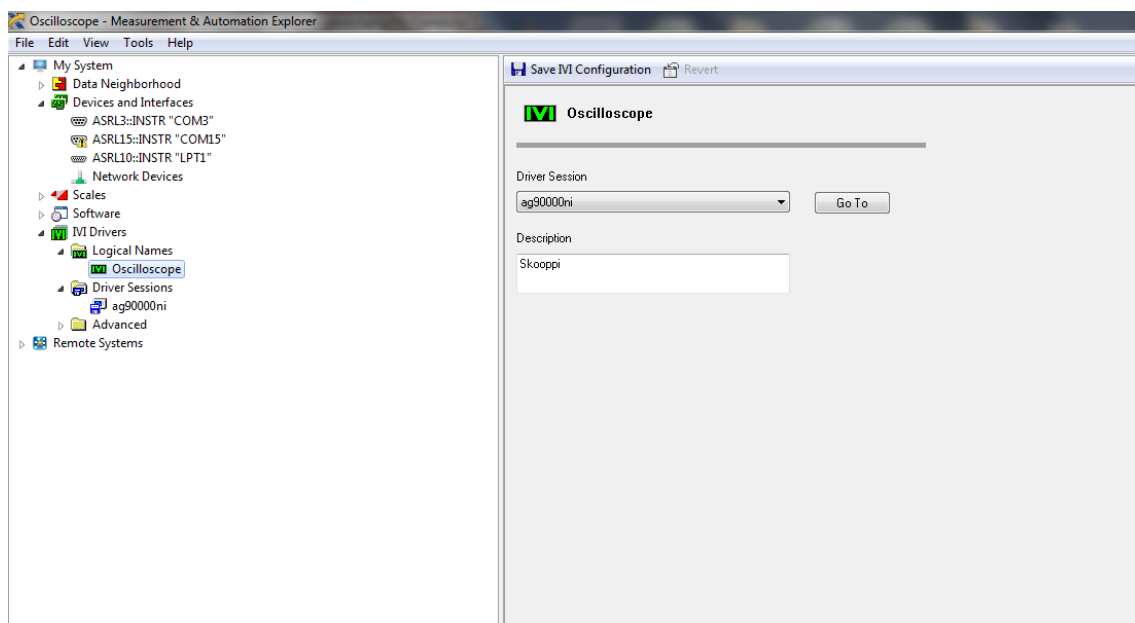


FIGURE 12. IVI driver logical name

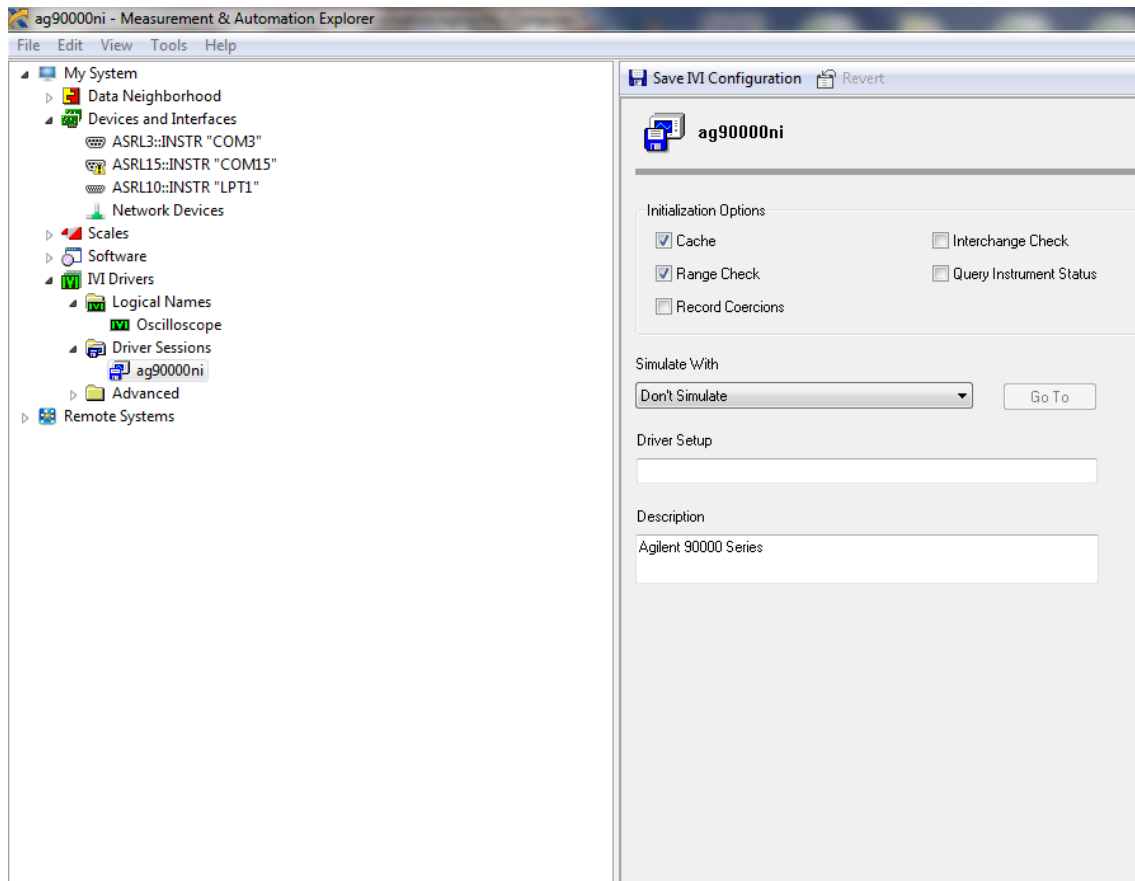


FIGURE 13. Driver session

## 6.2 Front Panel

Front Panel is the view that the User of the program will see, look FIGURE 14. Front Panel. Here, all of the needed configurations are given and also the measurement results will be shown.

A channel name is the channel which will be used in the measurement. A probe attenuation depends on the Probe properties, but usually this value could be 10. The maximum time is the time that will be given to an oscilloscope to perform its functions. If this time is exceed an error message will be created and the execution will be stopped. Reference values will be used when for example a rise time of the signal is calculated. The number of measurements enables to do the same measurements multiple times.

A path defines where the measurement result will be saved. A test point name and serial number information will be added to the name of the measurement results. Pressing a Run button will start the measurements and pressing a Stop will stop the LabVIEW application.

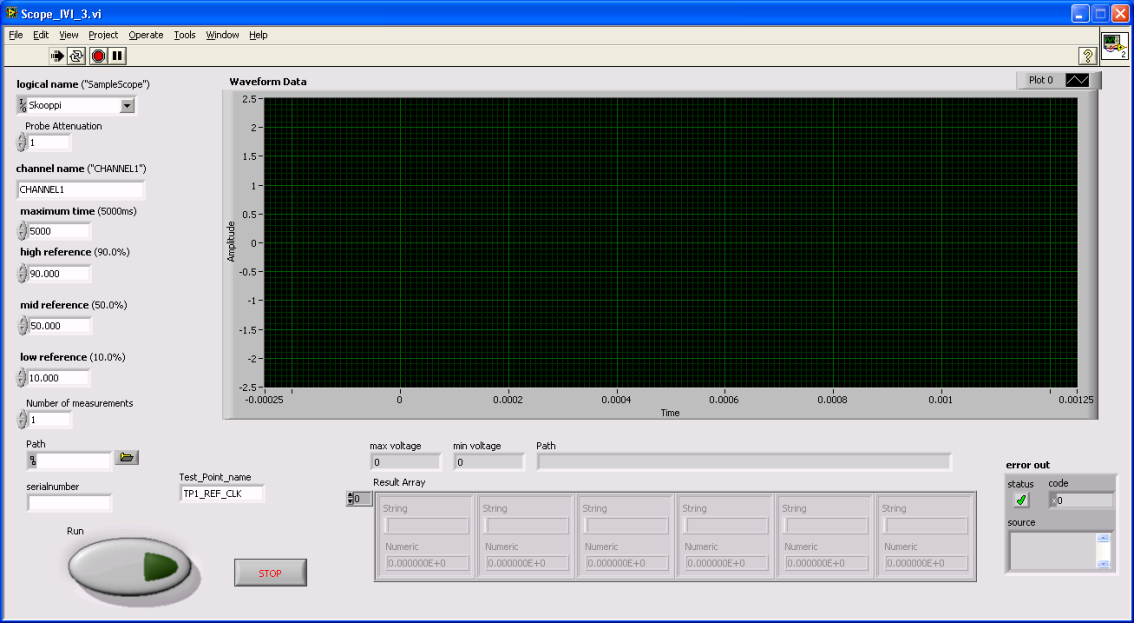


FIGURE 14. Front Panel

### 6.3 Block Diagram

LabVIEW uses the name Block Diagram from the code view of the program. LabVIEW is a graphical programming language and the execution is determined by the structure of a graphical block diagram. A programmer connects different function nodes by just drawing wires which propagate variables and any node can execute as soon as all its input data become available. (Wikipedia, 2013b. Date of retrieval 12.12.2013)

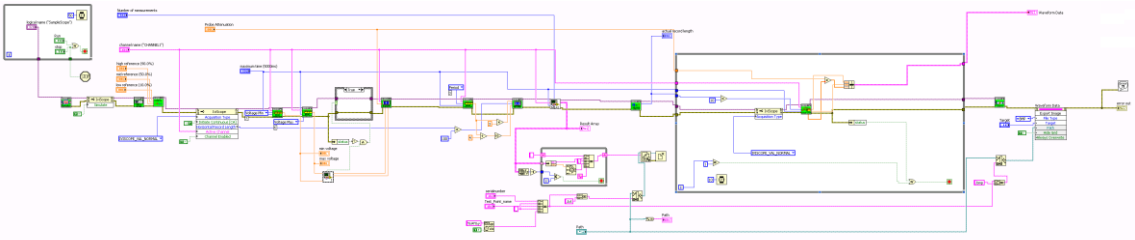
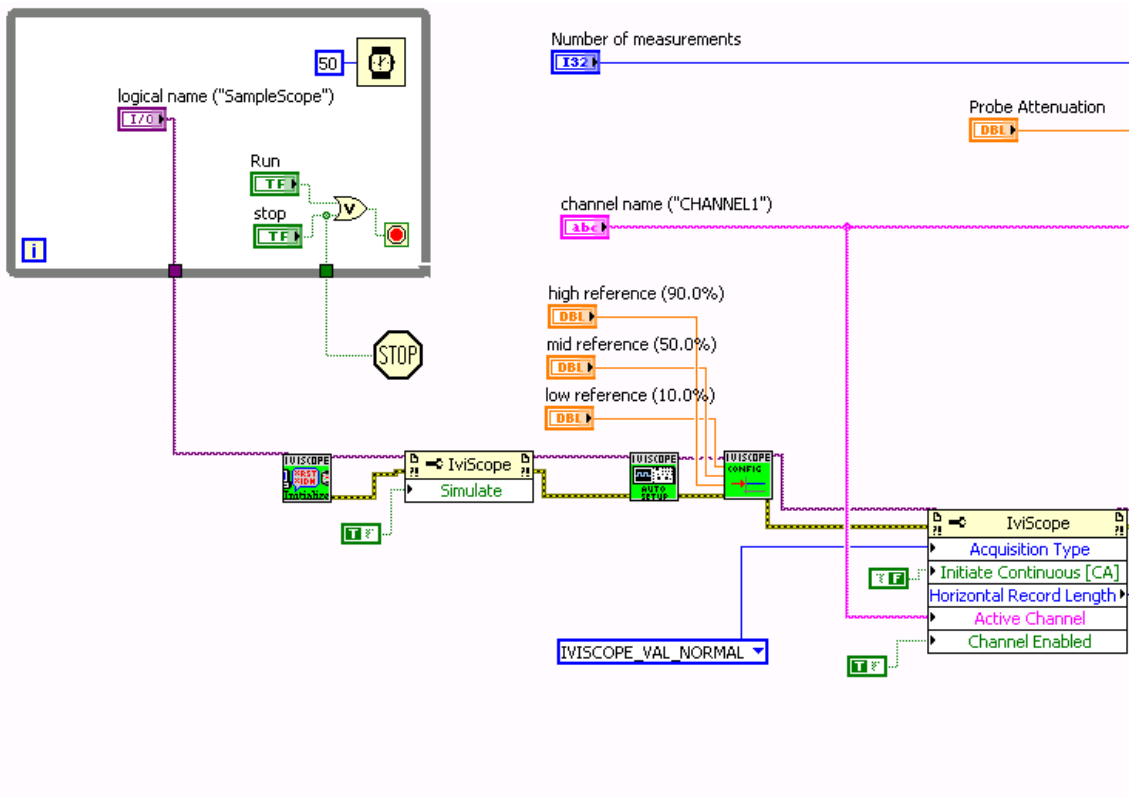


FIGURE 15. Block diagram

### 6.3.1 Setup

When the program is run it will stop in while loop where a logical name for a scope is given. In this case the logical name should be Skooppi, which is defined in MAX. Wait 50ms function is added to decrease valuable processor load. After the Run button is set to a TRUE value the actual code will start.

IviScope Initialize.vi creates a new IVI instrument session. After this, an auto setup command is given to the Oscilloscope to acquire signal properties approximately.



**FIGURE 16. Setup**

### 6.3.2 First Measurement and vertical adjust

The first measurement is taken with the parameters that an oscilloscope auto setup has defined. An acquisition type must be defined to normal to make a read Waveform VI to work. A IviScope Read Waveform[WM].vi is used to receive measurement results from the oscilloscope. Voltage Maximum and Minimum values are measured and they will be used to adjust an Oscilloscope

channel's vertical range and offset. The maximum voltage value is checked to be different from zero, if the Maximum value is zero, then error is generated.

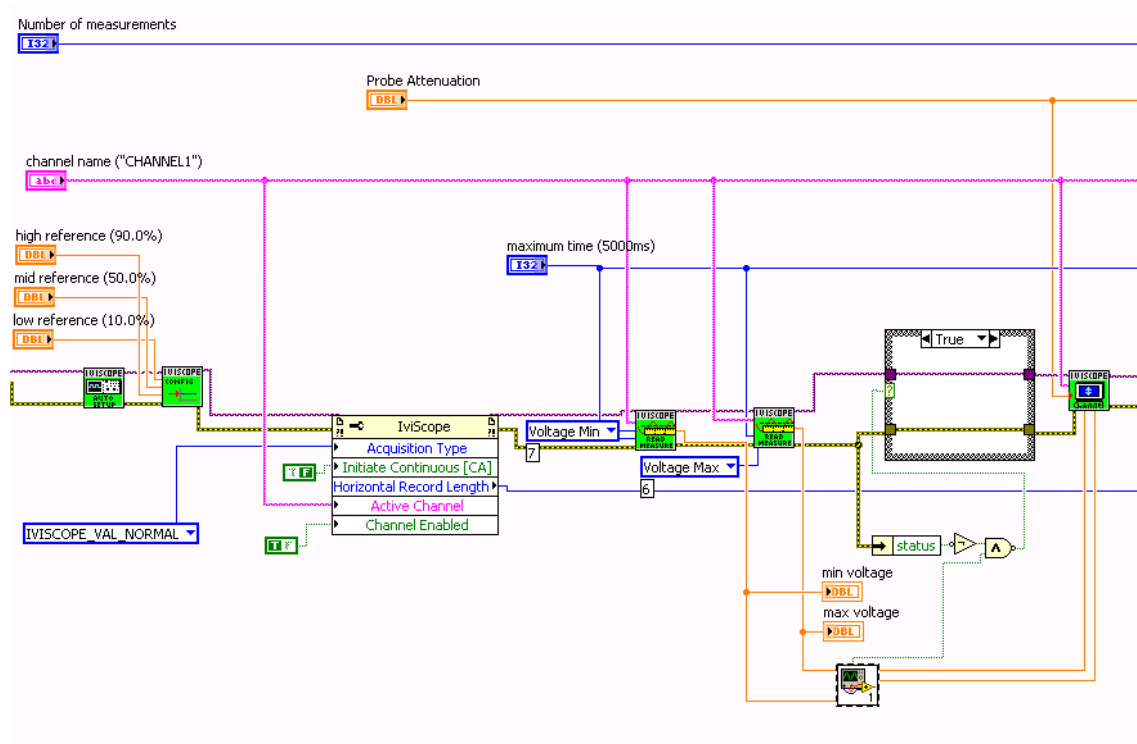


FIGURE 17. First Measurement

### 6.3.3 Horizontal adjust

A horizontal adjust is done using a IviScope Configure Acquisition Record virtual vi. A Signal period is measured and it is multiplied by three and it is used as time per record. This way a signal pattern will be shown three times per a taken record. An acquisition start time is delayed by adding one fourth of a period time. The minimum number of measurement points is given to a IviScope Configure Acquisition Record virtual vi to receive a good measurement accuracy. Then number of measurement points are taken from the Oscilloscope auto setup and multiplied by one hundred. See FIGURE 18. Record Length



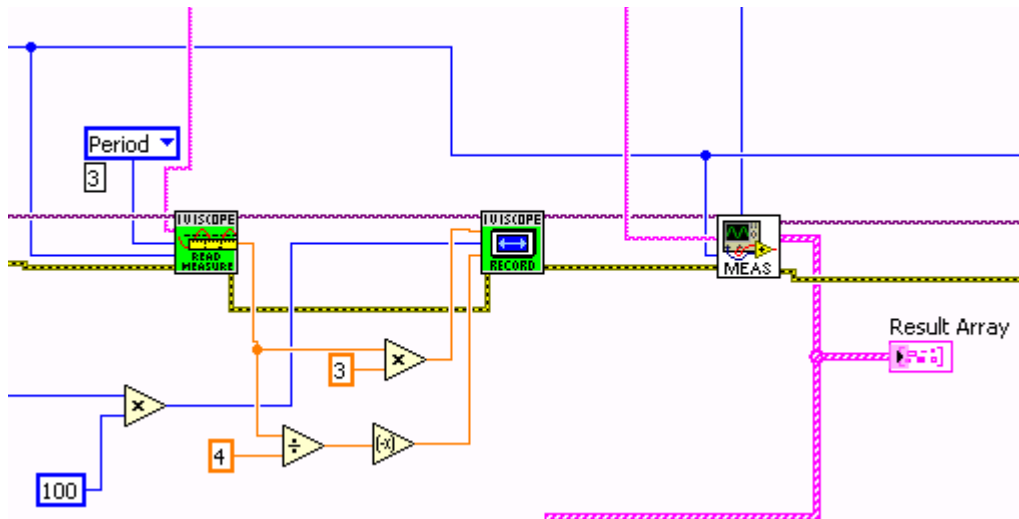


FIGURE 18. Record Length

### 6.3.4 Measurements

After the Horizontal and Vertical record lengths have been adjusted, the actual measurements can be done in sub-vi called Scope\_Meas.vi. They are done using a State Machine structure to achieve scalable measurements. Measurements can be easily added and removed. The number of measurement data is controlling how many times all the measurements are done. All the measurement results are saved in an Array variable including a cluster that includes a measurement name and a result as elements.

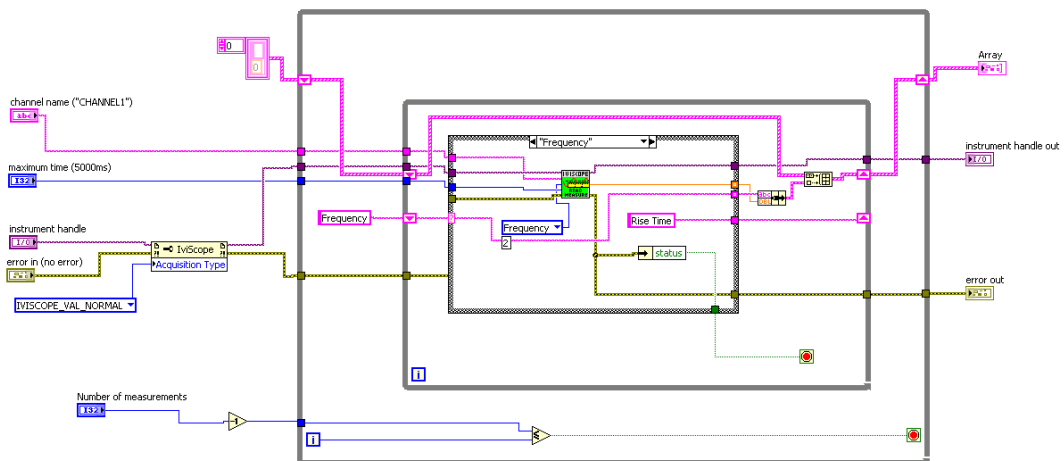


FIGURE 19. Scope\_Meas.vi





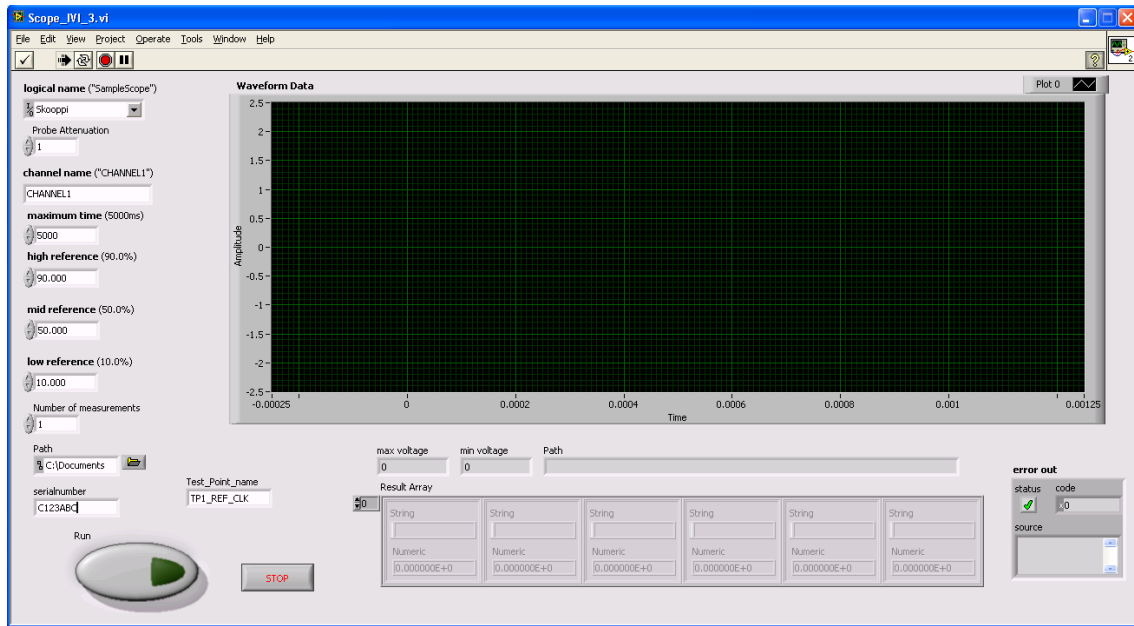


FIGURE 22. Testing-Setup

After the Run button is pressed the next phase is to determine simulated signal parameters. This of course is done in the simulation mode only. In this case a square wave with a 10MHz frequency and 5V amplitude was chosen. See FIGURE 23. Testing-Simulation.

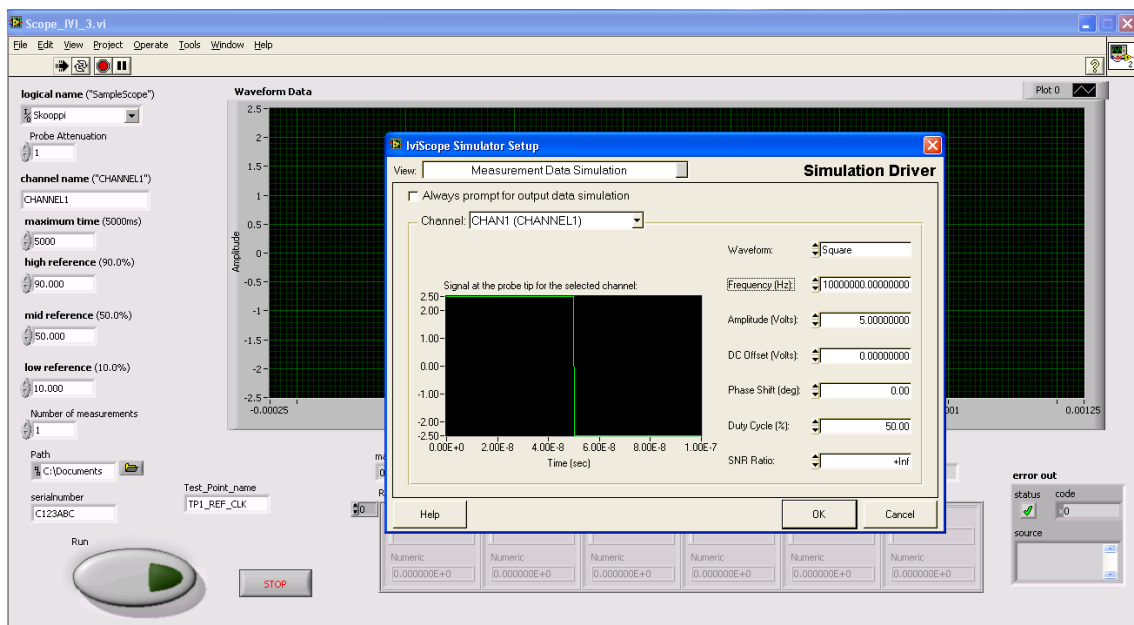
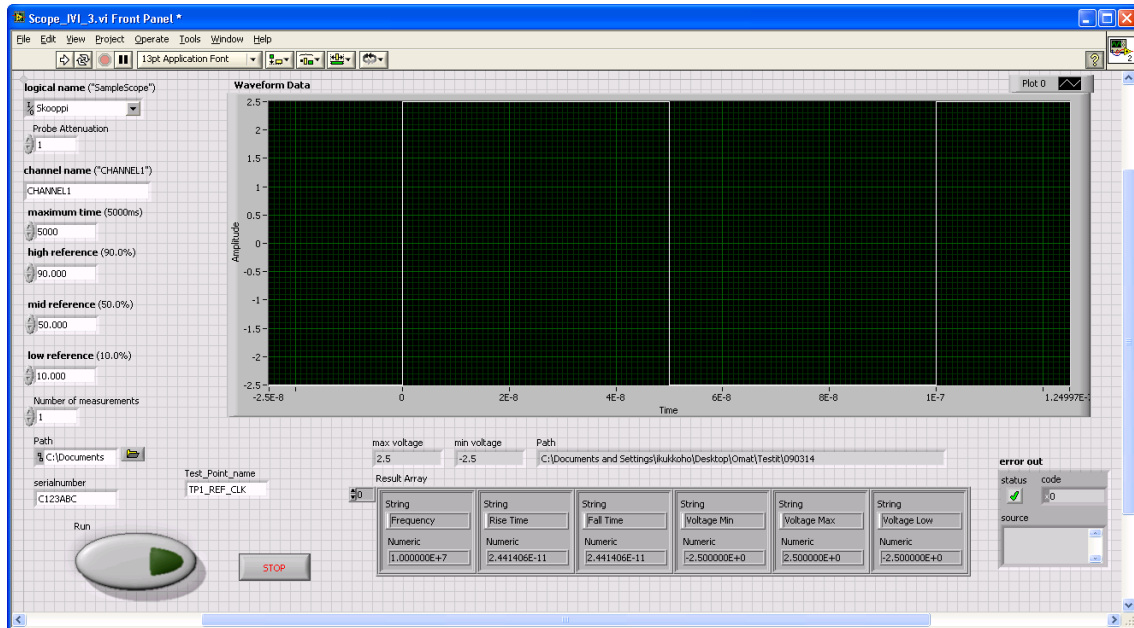


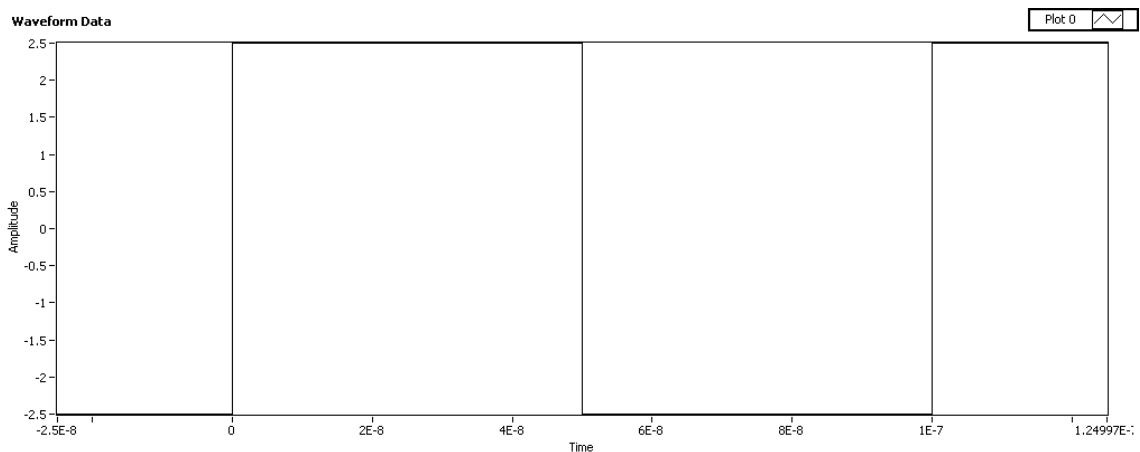
FIGURE 23. Testing-Simulation

After simulation parameters have been given, the measurements and result saving are done automatically. Waveform data is shown in the Front panel and also the Result array including the measurement data of the signal is shown.



**FIGURE 24. Testing-Result**

The waveform is saved in the path that has been defined in the front panel. The name of the waveform was C123ABC\_TP1\_REF\_CLK\_0914.bmp. As seen in FIGURE 25. Testing-Waveform, the simulated signal is ideal and it does not include e.g. overshoots or undershoots.



**FIGURE 25. Testing-Waveform**

The measurement data is saved in the path that has been defined in the front panel. The name of the data text file was C123ABC\_TP1\_REF\_CLK\_0914.txt, see FIGURE 26. Testing-Data. All of the data match the simulated data properties.

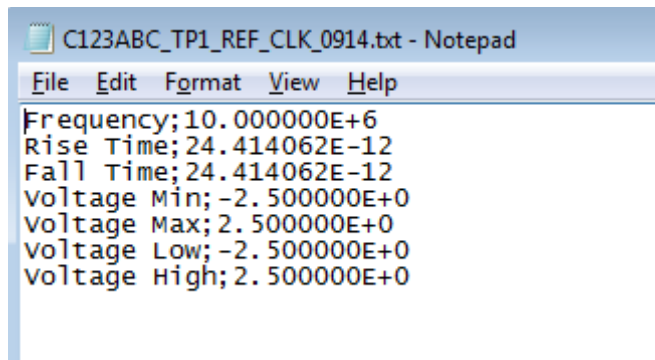


FIGURE 26. Testing-Data

## 7 POSSIBILITIES OF FURTHER DEVELOPMENT

Many of the test cases can be automated at least at some level. It only needs time and effort from test engineers to design and implement the test cases. The focus should be on long lasting tests and tests that need many repeats. For example, GPS measurements are one of the cases that should be automated.

One possible development sector is to start using a JTAG in R&D testing. JTAG nowadays includes much more than only a traditional Boundary Scan testing. If a product has a chip that contains Boundary Scan cells, a traditional boundary scan testing can be done. Devices that do not have boundary scan cells can possibly be tested using a debug port interface. JTAG interface provides a “backdoor” for emulating the processor and enables a test vector shifting without Boundary Scan cells. A Test Coverage rises when the DUT has more than one JTAG testing supported chip. A JTAG testing has also one big advantage: no need for a working SW in DUT which can be quite an advantage when dealing with new product concepts.

Using a SpiraTest library use requires a continuous updating and maintenance. At least one person should be given responsibility for taking care of it. A user guide should be done for the appropriate use of the library.

## 8 CONCLUSION

The test Automation can be a tool to speed up the testing process also in R&D. At the same time tests can have better repeatability and reproducibility parameters. This means more reliable results which will cause a more straight full feedback line from test results to HW designers. Changes to hardware design can be done faster and even some unnecessary proto design rounds can be avoided.

An automated clock signal test was created to speed up testing that must be done multiple times to different signals. One major issue was to discover a way to make the test software as versatile as possible. The use of IVI drivers were an answer for it, because it gives a way to use different measurement instruments without modifying the LabVIEW code.

The test standardization was done by creating a test library using SpiraTest. Also, the testing documentation and different signal parameters were defined in the theory section of the thesis. These things should help test engineers to do tests and documents in a more standardized form.



## 9 LIST OF REFERENCES

Behera, D. Varshney, D. Srivastava, S. and Tiwari, S. 2011. Eye Diagram Basics. . Date of retrieval 20.5.2014.

<http://www.edn.com/design/test-and-measurement/4389368/Eye-Diagram-Basics-Reading-and-applying-eye-diagrams>

Elektrobit Corporation, 2013, Internal source. Date of retrieval 1.11.2013

Elektrobit Corporation, 2014, What We Deliver / Wireless. Date of retrieval 16.3.2014

[http://www.elektrobit.com/what we deliver/wireless](http://www.elektrobit.com/what_we_deliver/wireless)

GUM:1995 ISO/CEI GUIDE 98-3.Uncertainty of measurement. Part 3: Guide to the expression of uncertainty in measurement

Hassan, M. Hassan, Y. Maryam, F. 2007.Type Approving the Signaling Systems. Date of retrieval 18.3.2014.

<http://ieeexplore.ieee.org.ezp.oamk.fi:2048/stamp/stamp.jsp?tp=&arnumber=4215529>

Inflectra Corporation, 2014. SpiraTest. Date of retrieval 20.04.2014

<https://www.inflectra.com/SpiraTest/>

Kossel, M.A - Schmatz, M.L. 2004. Jitter Measurements of High-Speed Serial Links. Design & Test of Computers, IEEE, 536 - 543.

Maxim Integrated, 2013. Clock Jitter. Date of retrieval 10.12.2013,

<http://www.maximintegrated.com/app-notes/index.mvp/id/3359>

Mike Silverman. 2010. How to Design a Better Reliability Test Program. Date of retrieval 20.3.2014.

<http://ieeexplore.ieee.org.ezp.oamk.fi:2048/stamp/stamp.jsp?tp=&arnumber=5447979&tag=1>

National Instruments, 2013. IVI drivers. Date of retrieval 10.12.2013,  
<https://www.ni.com/ivi/what.htm>

National Instruments, 2014. Simple labview program. Date of retrieval 20.10.2014,  
<http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/blockdiagram/>

OneStopTesting.com, 2013, Test Cases, Date of retrieval 1.11.2013  
<http://www.onestoptesting.com/test-cases/designing.asp>

Poikola V. 2012. DDR3-muisti ympäristön verifiointi singaalianalysaattorilla. Oulu: Oulu University of Applied Sciences, Degree Programme in Information Technology.

SAMSON AG, 2013. Serial Data Transmission. Date of retrieval 10.12.2013  
[http://www.samson.de/pdf\\_en/l153en.pdf](http://www.samson.de/pdf_en/l153en.pdf)

SILICON LABS, 2012. A PRIMER ON JITTER, JITTER MEASUREMENT AND PHASE-LOCKED LOOPS. Date of retrieval 10.12.2013  
<http://www.silabs.com/support%20documents/technicaldocs/AN687.pdf>

Teledyne Technologies, 2013, Operator's Manual. Date of retrieval 10.12.2013  
<http://cdn.teledynelecroy.com/files/manuals/sda-om-e.pdf>

Wikipedia, 2013a. Jitter Testing. Date of retrieval 12.12.2013,  
<http://en.wikipedia.org/wiki/Jitter#Testing>

Wikipedia, 2013b. LabVIEW. Date of retrieval 12.12.2013,  
<http://en.wikipedia.org/wiki/LabVIEW>

Zsolt, P. 2004. Rise Time. Date of retrieval 12.12.2013  
<http://www.hit.bme.hu/~papay/edu/Lab/RiseTime.pdf>