

TEST AUTOMATION IN DESIGNWARE HDMI RX

SENTHIL KUMAR PANNEERSELVAM

julho de 2020

TEST AUTOMATION IN DESIGNWARE HDMI RX

Senthil Kumar Panneerselvam



Tese/Dissertação

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2020

This document satisfies the requirements present on the Syllabus of
Tese/Dissertação, of the Master Degree in Electrical and Computer
Engineering - Telecommunications

Candidate: Senthil Kumar Panneerselvam, No 1151929, 1151929@isep.ipp.pt

Scientific Orientation: Professor Manuel Gericota, mgg@isep.ipp.pt

Company: Synopsys

Supervision: Eng. Rui Manuel de Sousa Ferreira, rui.ferreir@synopsys.com



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

July 3, 2020

Acknowledgments

For this thesis, and throughout my academic journey, many were the people who helped and contributed to my work and to improve my person. For this reason, I would like to thank each of the people who were present along this route and who in one way or another on the other they left their mark. The people who guided me, I would like to first thank the Professor Manuel Gericota, not only for the time spent with the thesis but also for his teachings in the classes and for the willingness and availability that you have always shown me. The patience for a long time and pushing me to write the thesis even it took long days. Orlando, Terra Rui and Rui Ferreira, from Synopsys, who guided me during my internship, they were always available to help me and that they believed in my work, my thanks. I also want to thank the entire team of prototyping kits, who welcomed me and helped me at all times. A special thanks to Pedro Rodrigues, my "buddy" within the company, who has always been present to ask questions, advise and guide. Pedro Moreira of the software and automation thank you very much also at all times and for all the help! Special thanks to my parents who have always been encouraging all throughout my career and always be a part of all my studies and taught me how education is important in this society.

This page was intentionally left blank.

Abstract

The first rule of any technology used in a business is that automation applied to any operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency. HDMI (High-Definition Multimedia Interface) is a leading technology in audio/video interface for transmitting uncompressed digital multimedia data. The limelight of this project is to automate the test procedure involved for testing HDMI-RX by using developed or customized software tools (Jenkins, python scripts,..etc.) targeting to reduce engineering manpower effort. The focus is not only on reducing the testing time but also to maintain an effective record of the final design parameters and ensuring exact result reproducibility. The testing of hardware components involves proceeding through several steps based on design parameters, evaluating results, and storing records to a File.

Keywords

HDMI,automate,testing,Jenkins, Python HDMI-RX-software tools-testing time-result reproducibility-design parameters, evaluating results-File

This page was intentionally left blank.

Contents

1	Introduction	1
1.1	Thesis Context	1
1.2	Objectives	3
1.3	Document Structure	3
2	Technology overview	5
2.1	HDMI	6
2.1.1	Audio ,Video and Image inputs of HDMI	7
2.1.2	HDMI Versions	8
2.1.3	Testing HDMI	14
2.1.4	Quantum Manager	15
2.1.5	980 User Interface	16
2.1.6	External 980 GUI Manager	17
2.2	Quantum Manager for compliance testing	17
3	Specifications	19
3.1	JENKINS	19
3.1.1	Continuous Integration with Jenkins	22
3.2	PYTHON	23
3.2.1	Python in test automation	24
3.2.2	PYWINAUTO	26
3.2.3	SWAPY	27
3.2.4	AUTOIt	27

3.2.5	PERFORCE	29
4	Project Flow	31
4.1	Synopsys Design Ware HDMI Rx IP Prototyping Kit	31
4.1.1	Earlier Software Validation	33
4.1.2	Complete reference design	33
4.1.3	HAPS-DX Prototyping System	34
4.2	EDID	35
4.2.1	Importance of EDID in HDMI	35
5	Project Development	39
5.1	JENKINS MULTIJOB	39
5.1.1	DWIPK-HDMIQP-RX-multijob	41
5.1.2	DWIPK-HDMIQP-RX-multijob-p4sync	41
5.1.3	DWIPK-HDMIQP-RX-multijob-HW-intel	42
5.1.4	DWIPK-HDMIQP-RX-multijob-ADVANTECH	42
5.1.5	DWIPK-HDMIQP-RX-multijob-haps	43
5.1.6	DWIPK-HDMIQP-RX-multijob-quantum	43
5.2	Running compliance test	44
5.2.1	Quantum GUI control	44
5.2.2	Audio Comparison	46
5.2.3	Image Comparison	47
6	Conclusion and Future Work	49

List of Figures

2.1	HDMI IP core.	7
2.2	HDMI Evolution's.	10
2.3	External GUI Manager.	17
3.1	Specifications.	20
3.2	Jenkins user interface.	21
3.3	Companies using Jenkins.	21
3.4	Continuous Integration with Jenkins.	23
3.5	Python Language.	25
3.6	Swapy.	28
3.7	AutoIt Gui	29
5.1	Jenkins multijob phases.	40
5.2	Jenkins multijob phases flow.	40
5.3	Slaves Assignment.	41
5.4	Continuous Integration.	43
5.5	Process flow diagram.	45
5.6	E.g. Code to open a folder in compliance test.	45
5.7	Audio comparison using FFT.	47
5.8	Image conversion using PIL and numpy.	48
6.1	Time taken for testing.	50
6.2	Manual vs Automatic testing.	50

This page was intentionally left blank.

List of Acronyms

- API** *Application Programming Interface*
- ARC** *Audio Return Channel*
- ASIC** *Application Specific Integrated Circuit*
- ATC** *Advanced Transfer Cache*
- CEC** *Consumer Electronics Control*
- CRT** *Cathode Ray Tube*
- CTS** *Clear to Send*
- DDC** *Display Data Channel*
- DSD** *Data Structure Diagram*
- DTS-HD** *Digital Theater Systems - High Definition*
- DVD** *Digital Versatile Disc*
- DVI** *Digital Visual Interface*
- eARC** *Enhanced Audio Return Channel*
- EDID** *Extended Display Identification Data*
- EQ** *Electronic Quartet*

ESD	<i>Electrostatic Discharge</i>
FFT	<i>Fast Fourier Transform</i>
FPGA	<i>Field Programmable Gate Array</i>
FMC	<i>Flexible Metallic Conduit</i>
GUI	<i>Graphical User Interface</i>
HAPS	<i>High-Performance ASIC Prototyping System</i>
HDCP	<i>High-bandwidth Digital Content Protection</i>
HDMI-RX	<i>High Definition Multimedia Interface Receiver Driver</i>
HDL	<i>Hardware Description Languages</i>
HDR	<i>High-Dynamic-Range</i>
HD-SDI	<i>High-Definition Serial Digital Interface</i>
HEC	<i>Hollerith Electronic Computer</i>
I2C	<i>Inter-Integrated Circuit</i>
IP	<i>Intellectual Protocol</i>
IPK	<i>IP Prototyping Kit</i>
ISEP	<i>Instituto Superior de Engenharia do Porto</i>
LCD	<i>Liquid Crystal Display</i>
MSV	<i>Mean Square Value</i>
PCI	<i>Peripheral Component Interconnect</i>
PCM	<i>Pulse-Code Modulation</i>
PHY	<i>Physical Layer</i>

PLL *Phase-Locked Loop*

SACD *Super Audio CD*

SDA *Serial Data Signal*

SOC *System On Chip*

sRGB *standard Red Green Blue*

RTL *Register Transfer Level*

TMDS *Transition Minimized Differential Signaling*

TV *Television*

UMRBus *Universal Multi-Resource Bus*

UXGA *Ultra Extended Graphics Array*

VGA *Video Graphics Array*

VPF *Vector Product Format*

VRR *Voltage Rectifier and Regulator*

xyYCC *Extended YCC Colorimetry for Video Applications*

Ycbcr *Green (Y), Blue (Cb), Red (Cr) (digital video color space)*

This page was intentionally left blank.

Chapter 1

Introduction

The major semiconductor industry with most evident growth are audio and video devices starting from mobile phones to Television. Advances in semiconductor technology are extremely fast, and with new development periodically, along with constant growth the companies focused on lower production cost with minimal time consumption between development and start of production. This will initiate new protocols, more complex designs and adopting faster to technological changes. HDMI is one of the important interface for modern digital media with new updates ranging from High Definition to 8K video handling capability and it will still update with more resolutions. The crucial process to adopt these fast growth and changes is test automation. Synopsys is one of the leading provider of high-quality, silicon-proven IP solutions for SoC designs. The broad DesignWare IP portfolio includes logic libraries, embedded memories, embedded test, analog IP, wired and wireless interface IP, security IP, embedded processors and subsystems.

1.1 Thesis Context

The projects reported in this document were developed in full collaboration with Synopsys as part of the Tese/Disseration (TEDI) course of the

Electrical and Computer Engineering Masters Degree lectured at Instituto Superior de Engenharia do Porto (ISEP). The work proposed by Synopsys is to automate the total test process for HDMI IP solutions. As the digital designs get more and more complex, the possibility of the existence of hardware bugs or defects increases. In this way, and because the production of the designs to real silicon is a very expensive process, companies like Synopsys prototype and test the circuits in Field Programmable Gate Arrays (FPGA) systems before producing or selling the design to a client. To perform these, Synopsys has a team (IP Prototyping Kits [IPK] team) that integrate the IP in larger designs and tests it in real world applications. Synopsys HDMI Transmitter (TX) and Receiver (RX) IP is one of the solutions and provide the necessary logic to implement and verify designs for various consumer electronic applications. The complete HDMI IP solutions consisting of digital controllers, PHYs and verification IP as well as IP Prototyping Kits with associated software and drivers, enable SoC designers to accelerate time-to-market and lower IP integration risk. The aim is to avoid manual stress and lengthy process to verify images and audio before releasing new HDMI IP solutions.

1.2 Objectives

Synopsys is developing various interface IP protocols and the focus of this project is to make test easier for the complex protocol which is HDMI because in order to automate it needs video and image processing, as it is the cutting edge technology in terms of Displays the automation must be done in an most efficient way. There are several steps taken to progress

- Study various automation techniques using Jenkins
- Document the changes needed to the existing design
- Methods to implement audio and video comparison
- Implement and Run tests
- Log and save results

1.3 Document Structure

To better understand the structure of this document, this section is used as a guideline to the content of the following chapters:

- **Technology Overview** - States the existing Display and video technologies. Informs about questions such as: What is HDMI? What does it stand for? What are the new features from HDMI? How tests have been done for HDMI? what is Quantum Manager? Peripherals? In short, it provides a state of the art for the subjects covered in this thesis;
- **Specifications** - although the HDMI protocol is mentioned on the previous chapter (Technology Overview) this chapter details the justification various software's and tools selected for this automation process and how it is used in a low level implementation;

- **Project Flow** - Informs how an HDMI Rx IP Prototyping Kit is structured in Synopsys and explained about HDMI IP prototyping using FPGA ?
- **Project Development** - this chapter states the project developments regarding this thesis in detail, keeping Synopsys, their clients and HDMI Alliance confidential issues protected;
- **Conclusion and Future Work** - Sums up the work done, what we achieved from this automation? How to improve this process further? and refers to future work.

Chapter 2

Technology overview

Display technology is evolving at an impressive rate with LCD and flat panel technologies gaining an increasing market share over traditional CRT display application. Prior to new modern interfaces display technology was more straightforward where VGA was the authoritative standard in big business PCs. And keeping in mind that screens were smaller, CRT were massive and the quality was not on a par with it is today, the interface choices were substantially more direct. Today, interface alternatives incorporate DVI, HDMI and DisplayPort, and also the waiting accessibility of VGA to empower similarity with more established, inheritance equipment. The quick spreading of 4K-screens requests more noteworthy video transfer speed for greatest execution than that which can be provided through contemporary HDMI. Accomplishing 4K resolutions at anything past a 60-Hz refreshing rate now requires DisplayPort. Worth noting, however, is that becoming tied up with the innovation is a long haul venture, in light of the fact that DisplayPort takes into consideration for numerous 4K showcases, or future building for 5K and 8K resolutions. HDMI has been specified to consent to an advancing arrangement in detail , each released as an alternate form and appointed a variant number. This section clarifies the development of HDMI innovation and the testing of HDMI.

In everything electronic, progress advancement conveys the appreciated advantages of endless change. DisplayPort was relatively revolutionary and confronted firm rivalry from HDMI in top notch digital gadgets. Display interfaces are no exception. A little more than ten years back the incorporation of sound and video, following on rapid move from simple to advanced, upset the nature of video yield. Also, alongside the clarity of top notch screens and the simplicity of incorporated sound came new interface models and a large group of new connectors and links [1].

2.1 HDMI

The High Definition Multimedia Interface (HDMI) is a wired advanced interconnect that replaces the simple TV out or VGA out. There are many advantages to utilize HDMI for displays. Fundamentally, it runs quick at 1080p, with a rate of around 150 Megapixels/second. Consider, by differentiate, what the communicate world did when it expected to course advanced video from point to point. The outcome was HD-SDI, top notch serial computerized interface. One coaxial link can route a HD-SDI flag several feet without blunders, with no repeater equipment or EQs in the line. Had the consumer business decided on a coaxial-based standard, we had have the capacity to do likewise in our homes. As a matter of fact, few of us have to make 300-foot runs; yet the capacity to run 300 feet without issues would be joined by strong conviction of having the capacity to do 50, or 75, in a HDMI link with no stress at all. The following section portrays the advancement and refresh of HDMI technology[2].

HDMI transmits scrambled uncompressed advanced video and sound information (utilizing TMDS like DVI), it bolsters DDC for display distinguishing proof and ability promotion. Additionally, it presents various new advances, which are possibly developed from a security perspective[3]. These include:

- CEC – Consumer Electronics Control
- HDCP - High-bandwidth Digital Content Protection
- HEC – HDMI Ethernet Channel

2.1.1 Audio ,Video and Image inputs of HDMI

Many A/V receivers contain digital processors that can take analog video signals, from a VHS or DVD player, and convert them to HDMI. This makes it possible to use just one HDMI cable, connected from the receiver to a TV, for all video processing. Figure 2.1 explains the IP core inside HDMI.

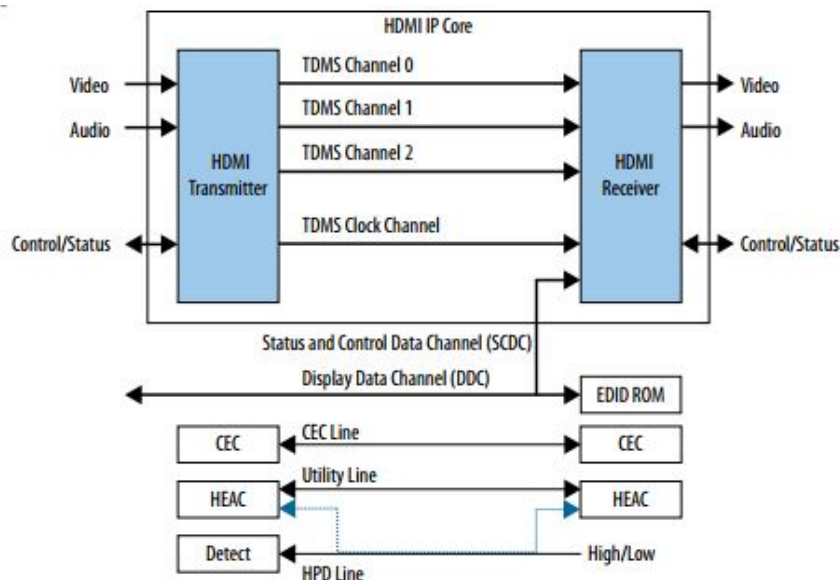


Figure 2.1: HDMI IP core [3]

- Pin 1 - TMDS Data2+
- Pin 2 - TMDS Data2 Shield
- Pin 3 - TMDS Data2-
- Pin 4 - TMDS Data1+

- Pin 5 - TMDS Data1 Shield
- Pin 6 - TMDS Data1-
- Pin 7 - TMDS Data0+
- Pin 8 - TMDS Data0 Shield
- Pin 9 - TMDS Data0-
- Pin 10 - TMDS Clock+
- Pin 11 - TMDS Clock Shield
- Pin 12 - TMDS Clock-
- Pin 13 - CEC
- Pin 14 - Reserved (HDMI 1.0–1.3c) Utility/HEAC+ (HDMI 1.4+, optional, HDMI Ethernet Channel and Audio Return Channel)
- Pin 15 - SCL (I2C serial clock for DDC)
- Pin 16 - SDA (I2C serial data for DDC)
- Pin 17 - Ground (for DDC, CEC, ARC, and HEC)
- Pin 18 - +5 V (minimum 0.055 A)[3]
- Pin 19 - Hot Plug Detect (all versions) HEAC- (HDMI 1.4+, optional, HDMI Ethernet Channel and Audio Return Channel)

2.1.2 HDMI Versions

. HDMI versions and specifications [4] are stated and explained in the following subsections and in Figure 2.2.

HDMI 1.0

Release date: December 2002

Specs:

- Single-cable digital audio/video connection with a maximum bitrate of 4.9Gbps
- Supports up to 165Mpixels/sec video (1080p at 60Hz or UXGA)
- 8-channels of 192kHz/24-bit audio (PCM)
- Added support for DVD Audio
- Slight mechanical and electrical spec changes

HDMI 1.1

Release date: May 2004

Specs:

- Single-cable digital audio/video connection with a maximum bitrate of 4.9Gbps
- Supports up to 165Mpixels/sec video (1080p at 60Hz or UXGA)
- 8-channels of 192kHz/24-bit audio (PCM)
- Added support for DVD Audio
- Slight mechanical and electrical spec changes

HDMI 1.2

Release date: August 2005

Specs:

HDMI Version	1	1.1	1.2	1.3	1.4	2.0	Importance
sRGB	Yes	Yes	Yes	Yes	Yes	Yes	High
YCbCr	Yes	Yes	Yes	Yes	Yes	Yes	High
8 channel LPCM/192 kHz/24-bit Audio	Yes	Yes	Yes	Yes	Yes	Yes	High
1080p Video	Yes	Yes	Yes	Yes	Yes	Yes	High
Consumer Electronic Control (CEC)	Yes	Yes	Yes	Yes	Yes	Yes	Low
DVD Audio	No	Yes	Yes	Yes	Yes	Yes	Low
Super Audio CD (DSD)	No	No	Yes	Yes	Yes	Yes	Low
Deep Color	No	No	No	Yes	Yes	Yes	Low
xvYCC	No	No	No	Yes	Yes	Yes	Low
Auto Lip-sync	No	No	No	Yes	Yes	Yes	High
Dolby TrueHD/DTS-HD Bitstream	No	No	No	Yes	Yes	Yes	Medium
Audio Return Channel	No	No	No	No	Yes	Yes	Medium
Automotive Connection System	No	No	No	No	Yes	Yes	Low
3D Over HDMI	No	No	No	No	Yes	Yes	Medium
4K 30 fps Resolution	No	No	No	No	Yes	Yes	Medium
Ethernet Channel	No	No	No	No	Yes	Yes	High
Micro HDMI Connector	No	No	No	No	Yes	Yes	Low
Expanded Support for Color Spaces	No	No	No	No	Yes	Yes	Low
4K 60 fps Resolution	No	No	No	No	No	Yes	Medium
YCbCr 4:2:0	No	No	No	No	No	Yes	Low
32-channel Audio	No	No	No	No	No	Yes	Low
1536 kHz Audio	No	No	No	No	No	Yes	Low
4 audio streams	No	No	No	No	No	Yes	Low
2 video streams	No	No	No	No	No	Yes	Medium
21:9 aspect ratio	No	No	No	No	No	Yes	Low

Figure 2.2: HDMI Evolution's [4]

- Added DSD (Direct Stream Digital) support, allowing native transmission of Super Audio CD (SACD) content at up to 8 channels
- Enabled and acknowledged an HDMI Type A connector for PC-based sources
- Permitted PC sources to use native RGB color-space with the optional ability to also support the YCbCr color space for consumer electronics applications
- Mandated that HDMI 1.2 and later displays support low-voltage sources such as those found with PCI Express technology (the current display interface standard for PC video cards)

HDMI 1.2a

Release date: December 2005

Specs:

Fully specified Consumer Electronic Control (CEC) features, command sets, and compliance tests Minor changes to CEC (Consumer Electronic Control) spec

HDMI 1.3

Release date: June 2006

Specs:

- Increased single-link bandwidth to 340 MHz (10.2 Gbps)
- Optionally supports 10-bit, 12-bit, and 16-bit "Deep Color" per channel (over one billion colors) up from 8-bit
- Allowed the use of xvYCC color space (previously just sRGB or YCbCr)
- Incorporated automatic audio "lip" syncing capability
- Supported output of native Dolby TrueHD and DTS-HD Master Audio streams for external decoding by AV receivers
- Made available a new Type C "mini" connector for devices such as camcorders
- Added gamut Metadata transmission capability
- Added Reference Cable Equalizer mandate to high frequency displays to recapture degraded copper cable signal

HDMI 1.4

Release Date: May 2009

Specs:

- Ethernet over HDMI: Adds support for ethernet over HDMI for connected devices
- Audio Return Channel: Allows an HDMI-connected TV with a built-in tuner to send audio data "upstream" to a surround audio system, eliminating the need for a separate audio cable
- 3D Support: Defines input/output protocols for major 3D video formats, paving the way for true 3D gaming and 3D home theater applications
- Support for 4K x 2K resolution (3840 x 2160) at 24Hz, 25Hz, and 30Hz and 4096 x 2160 at 24Hz
- Real-time signaling of content types between display and source devices enables a TV to optimize picture settings based on content type
- Additional Color Spaces: Adds support for sYCC601, AdobeRGB, and AdobeYCC601, which are used in digital photography and computer graphics
- HDMI Micro Connector: A new, smaller connector for phones and other portable devices, supporting video resolutions up to 1080p.
- Automotive Connection System: New cables and connectors for automotive video systems, designed to meet the unique demands of the motoring environment

HDMI 2.0

Release Date: September 2013

Specs:

- Increases bandwidth to 18Gbps

- Resolutions up to 4K@50/60 (2160p), (4 times the clarity of 1080p/60 video resolution)
- Up to 32 audio channels
- Up to 1536kHz audio sample frequency
- Simultaneous delivery of dual video streams to multiple users on the same screen
- Simultaneous delivery of multi-stream audio to multiple users (Up to 4)
- Support for the wide angle theatrical 21:9 video aspect ratio
- Dynamic synchronization of video and audio streams
- Updated CEC extensions for more expanded command and control of consumer electronics devices through a single control point
- Support for 4:2:0 chroma subsampling
- Support for 25 fps 3D formats
- Backward compatible with high speed (category 2) HDMI cables

HDMI 2.1

Release Date: Q2 2017

Specs:

- Higher Video Resolutions support a range of higher resolutions and faster refresh rates including 8K60Hz and 4K120Hz for immersive viewing and smooth fast-action detail
- Dynamic HDR ensures every moment of a video is displayed at its ideal values for depth, detail, brightness, contrast, and wider color gamuts — on a scene-by-scene or even a frame-by-frame basis

- 48G cables enable up to 48Gbps bandwidth for uncompressed HDMI 2.1 feature support including 8K video with HDR. The cable is backwards compatible with earlier versions of the HDMI Specification and can be used with existing HDMI devices
- eARC supports the most advanced audio formats such as object-based audio, and enables advanced audio signal control capabilities including device auto-detect
- Game Mode VRR features variable refresh rate, which enables a 3D graphics processor to display the image at the moment it is rendered for more fluid and better detailed gameplay, and for reducing lag, stutter, and frame tearing

2.1.3 Testing HDMI

Testing HDMI assumes a critical part in the Electronics Industry. Before mass delivering or conveying any Licensed Product or segment that cases compliance with the HDMI Specification, every IP must undergo a test for HDMI compliance. Regardless of the many points of interest offered by new releases, it is additionally critical to stretch that HDMI 1.4b is a long way from outdated. HDMI later forms are completely down gradable with HDMI 1.4b and utilizes a similar 19-stick connectors. Higher velocities, for example, 4K at 50/60Hz require a Cat2 HDMI link. Furthermore, to guarantee in backward compatibility, the HDMI Forum is requiring that HDMI 2.0 items must pass HDMI 1.4b compliance testing. First, the IP must be individually tested as indicated in the then-current HDMI Compliance Test Specification. The HDMI Compliance Test Specification gives a suite of testing strategies, and builds up certain base necessities determining how each HDMI IP manufacturer should test Licensed Products for conformance to the HDMI Specification. Such compliance testing is constrained to as-

assessment of an item's congruity with the HDMI Specification and is not intended to demonstrate the general execution of any Licensed Product[5].

An sample test of first generation display under classification of authorized item subject to test here under(i.e., Source,sink, Repeater and link) must be tried at an ATC(Authorized Testing Centers)prior to large scale manufacturing or dispersion if such item asserts conformance to the High Definition Multimedia Interfaces or bears any IP trademarks. Once an IP result of each sort has passed testing, it is not expected to agree to the ATC consequent results of that sort. For instance, once an IP's TV has passed ATC testing, it is not required to present another TV or other show to ATC testing. By and by, to help guarantee HDMI interoperability, we urge IPs to utilize the ATCs.

In Synopsys such IP is tested on their own facility before prior submission to ATC so that the IP can compare its own results with the ones from ATC. This will assist in evaluating its ability to perform proper self-testing of subsequent production Models. Synopsys use Quantum Manager to test HDMI internally[6]. The Quantum Manager was used in HDMI in order to compare the data and analyse the results. The detail description of Quantum Manager is explained in next section.

2.1.4 Quantum Manager

Quantum Manager gives full visibility into the HDMI protocol, timing, control and auxiliary data up to 300MHz (HDMI).It can provide compliance test support from HDMI1.4 to advanced HDMI versions(of HDMI 2.0 4K Ultra HD sources at 50/60Hz).The Quantum 980 analyzer can be connect through GUI from any PC with ssh connection support. The Quantum 980 test instruments are not expected to provide solutions for the protocol it give only the test analysis and results[7]. The Analyser may be controlled through its own Graphical User Interface (GUI) or remotely, through a PC-

based GUI. The 980's worked in shading touch screen offers a graphical UI (GUI) to order the module and run the compliance tests.

The 980 Protocol Analyzer module operates in one of two modes: Capture mode or Real Time mode. Capture Mode - In Capture mode, it is able to capture and decode either unencrypted or HDCP-encrypted HDMI protocol. It supports and displays timing, control and subordinate test data of HDMI protocol or HDMI IP. The captured data can be shown directly in GUI interface to analyse further. The captured data also have information of protocol and control, with exact time stamps. The timestamps can be used to determine interoperability issues ahead of schedule in the lifetime cycle. The data to be viewed can be readily available in manufactures web-site.

Real Time Mode – As the name indicates it provide the information on real time basis. The information like control, timing data , the HDCP and EDID exchanges would all be able to be seen in real time. It watch changes in the source and display it on GUI. The 980's worked in 10.4-inch shading touch screen offers a graphical UI (GUI) to ace the device and to see approaching video in real time mode. In test automation the image analysis can be done by using these modes. The captured image can be stored in local and send for image processing. The picture caught can be looked at utilizing python by comparing each pixel by using specific python tool.

2.1.5 980 User Interface

The 980 Protocol Analyzer module gives a graphical User Interface to operation. The GUI can be controlled either by internal embedded GU Manager or by External GU Interface. The latest version of 980 have full support to external GUI interface. For automation reason the External GUI is exceptionally appropriate to run the compliance test as it is controlled effortlessly by windows automation.

2.1.6 External 980 GUI Manager

The external 980 GUI Manager gives full control of 980 analyser. Additionally the external 980 GUI Manager empowers to work the 980 HDMI Protocol Analyzer through a large interface which enables to utilize different boards at the same time[8]. Figure 2.3 demonstrates the External GUI Manager

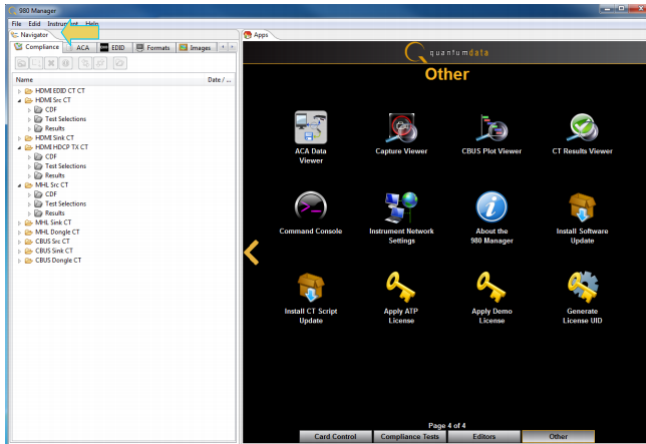


Figure 2.3: External GUI Manager.[8]

2.2 Quantum Manager for compliance testing

In order to verify HDMI protocol the compliance tests before sending to ATC it can be tested internally by 980 HDMI Protocol Analyzer module. It gives further examination and compliance test highlights which is important to process number of tests rapidly and with reduced cost[8]. In HDMI the compliance test is picked with predefined information to execute the compliance test prepare. This incorporates HDMI 2.0 compliance tests for EDIDs on sink and 4:2:0 pixel encoding of source and sink gadgets. The compliance tests are perfect for pre-testing your item before sending to an Authorized Test Centers (ATC) for endorsement. Pre-testing gives included

certainty that your item will delay on the principal passage to the ATC. Where allowed, the 980 HDMI Protocol Analyzer can perform individual test for HDMI item. Self-testing offers much greater advantages from time to market and cost decrease than pre-testing. The apparatus gives all the fundamental information test results to demonstrate similarity. The Final outcomes can be logged in HTML file.

This chapter described about the technology used for the project and also about the specification for proceeding test automation.

Chapter 3

Specifications

This chapter presents the test automation specification whose different levels are shown in figure 3.1. The software were selected based on recent analysis of automating windows GUI. This chapter presents a detailed description of the software developed to automate the test flow.

3.1 JENKINS

Jenkins is a self-contained, open source automation server, which can be used to automate all sorts of tasks. It can do building, testing, and deploying software. Jenkins can be installed to any system with the Java Runtime Environment (JRE) installed. Jenkins can control the test automation flow from start to end in multijob format. The multijob plugin is the process to run the job on flexible in demand. Jenkins is an open source automation server with a series plugin ecosystem to support practically every tool as part of delivery pipelines. It helps to automate the whole software development process with common things like continuous integration and by empowering teams to implement the technical aspects of continuous delivery. It is a server-based system running in a servlet container such as apache tomcat. It also adopt version tools like git, Perforce as an online platform as well as

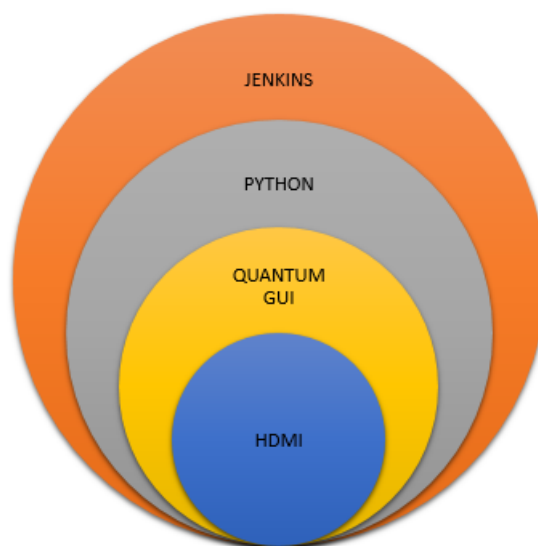


Figure 3.1: Specifications

arbitrary shell scripts and Windows batch commands. Each build can be triggered by means of online through server commands and several offline services. Builders can be triggered by various means for example by commit in versions control system, by scheduling a cron-like mechanism and by requesting a specific build URL.

Jenkins have plugins used to increase the functionality inside the server. It is used for extending its use for the projects in which languages other than java can be used. Many build tools are supported via their respective plugins. It can also change the way Jenkins looks or add new functionality. Plugins and Jenkins can display the reports and generate trends and render them in GUI. It is a server application with continuous server and continuous integration support. Its use-cases are building application from version control system, running acceptance tests on test, developing and production

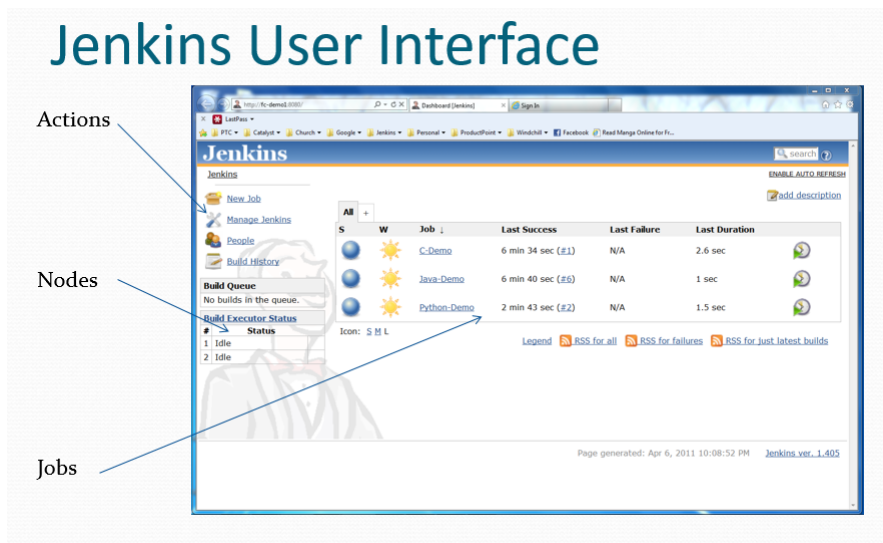


Figure 3.2: Jenkins user interface [9].

environment. Jenkins security depends upon two factors: access control and protection from external threats. Access control can be customized via two ways, user authentication and authorization.



Figure 3.3: Companies using Jenkins [10].

3.1.1 Continuous Integration with Jenkins

To update the process continuously and verify the results obtained as a cyclic process continuous integration plays an important role also for maintaining balance for automating, continuous integration plays a vital role in which all development work is integrated as early as possible. The resulting artifacts are automatically created and tested. This process should identify errors as very early in the process. Jenkins as an open source environment also support continuous integration to build automation and to execute a predefined list of steps. It has specific trigger option which can be initiated internally or externally or event based. For example, every 20 minutes or after a new commit in a Git repository[11]. In this process the Jenkins trigger can be build while updating software or hardware inside the perforce software library and Jenkins initialize it and start to build the multijob. The list of steps can, for example, include:

- Perform a software build
- Run a shell script
- Archive the build result
- Afterwards start the integration tests

The multijob plugin in Jenkins is used to monitor the execution of the steps and allows to stop the process if one of the steps fails. Jenkins also displays the final result about the build process or failure. Additional plugins can extend Jenkins, e.g., for building and testing Android applications or to support the Git version control system.

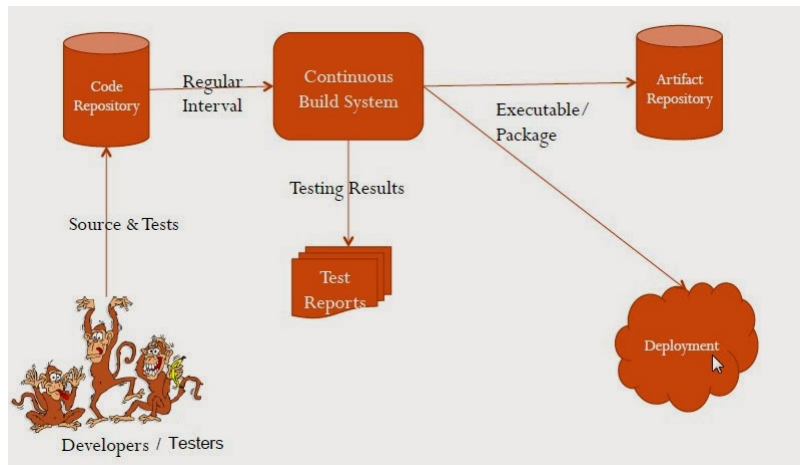


Figure 3.4: Continuous Integration with Jenkins[12].

3.2 PYTHON

This session explains what are the python wrappers used to make this windows automation work. It starts with pywin auto as the interface with python and windows with all libraries needed for accessing windows.

If a GUI some pop up a window to move to next step. For eg. in our case when a GUI pop up to click the image comparison is passed or failed depends on the data running on background. It needs to make a automation to click and do the mouse control.

The mouse control can be done by two tools Swapy and AutoIt. Swapy is used to locate where to click and AutoIt to make a click. Both can be exported as a python script from the software itself. Combination of all these tools make the automation process more accurate.

Python is a programming language that gives a chance to work more rapidly than other programming language. Experienced developers in some other language can get Python rapidly, and beginners find the clean syntax and indentation structure simple to learn[13]. For hybrid problems which involves software and hardware python predict the risk and make the task easier as

it is flexible for both hardware and software coding. Professionally, Python is suitable for back end web advancement, information examination, data analysis, artificial intelligence, and scientific computing. Many developers have also used Python to build productivity tools, games, and desktop apps, so there are plenty of resources to help IT developers to learn how to do those as well. Being an abnormal state dialect, Python uses like English, which takes a great deal of language structure learning worry off coding beginners. Python handles a great deal of many-sided quality for IT developers, so it is exceptionally beginner-friendly in that it enables amateurs to concentrate on picking up programming ideas and not need to stress over an excess of detailed elements. As a powerfully wrote dialect, Python is truly adaptable. This implies there are no hard standards on the most proficient method to construct highlights, and it has greater adaptability taking care of issues utilizing distinctive strategies (however the Python logic energizes utilizing the obvious approach to illuminate things). Moreover Python is also more forgiving of errors, so regardless it has the capacity to assemble and run program until the point when IT developers hit the risky part. Python is widely used all over the world as it is easy to adopt and learn even for complex problems. Figure 3.5 describes the usage of popular tools all over the world.

3.2.1 Python in test automation

The project implements various automation like windows GUI, hardware, signal processing networking. Python implements all these automation process. Knowing different Python modules for altering spreadsheets, downloading documents, and launching projects is valuable, however once in a while there simply are not any modules for the applications IT developers should work with. Python contains various packages for individual tasks like image and audio comparison, telnet call, GUI control which can be imple-

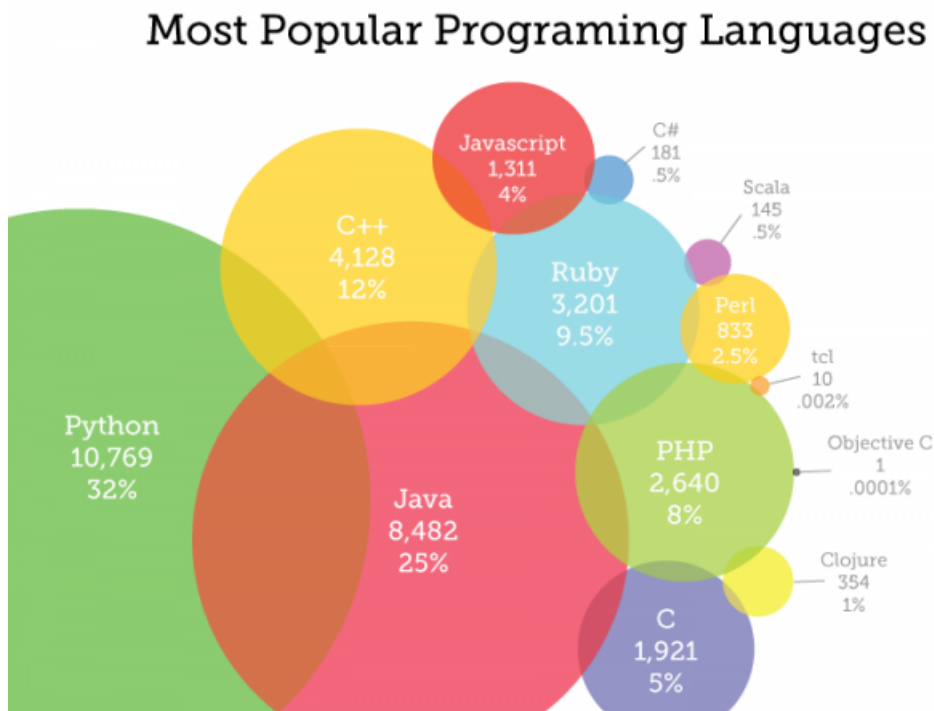


Figure 3.5: Python Language [14].

mented in SWAPY, NUMPY, PIP, PYWINAUTO packages. A definitive device for automating on PC are programs that straightforwardly control the console and mouse. These projects can control different applications by sending them virtual keystrokes and mouse clicks, just as if someone was sitting at PC and connecting with the applications self. This strategy is known as graphical UI automation, or GUI automation for short. With GUI automation, projects can do anything that a human client sitting at the PC can do, aside from spill coffee on the console. Think of GUI automation as programming an automated arm. We can program the automated arm to sort at console and move mouse. This strategy is especially valuable for assignments that include numerous careless clicking or rounding out of forms. The pyautogui module has capacities for simulate mouse developments, catch clicks, and looking over the mouse wheel. Before bouncing in

to a GUI mechanization, it is important to know how to escape issues that may emerge. Python can move mouse and sort keystrokes at an incredible speed. In fact, it may be too quick for different projects to stay aware of. Additionally, if something turns out badly yet program continues moving the mouse around, it will be difficult to tell what precisely the program is doing or how to resolve from the issue. Like the enchanted brooms from Disney's *The Sorcerer's Apprentice*, which kept filling—and then overflowing—Mickey's tub with water, program will gain out of power despite the fact that it is following directions superbly. Ceasing the program can be troublesome if the mouse is moving around without anyone else, keeping it from tapping the idle window to close it. Luckily, there are a few approaches to keep or recover from GUI computerization issues.

There are tremendous stuff Python can help in automating tasks like

1. Setting up environments for tests;
2. Preparing test data;
3. Analyzing data;
4. Extracting performance data.

The quantum manager was used in HDMI in order to compare the data and analyze the results. There are numerous list of python test automation frameworks, tools, libraries and software to help software engineers easily bootstrap test automation on python, like JUNIT XML which can output the xml file automatically, a unit test runner that can save test results to XML files that can be consumed by a wide range of tools, such as build systems, IDEs and continuous integration server.

3.2.2 PYWINAUTO

Pywinauto is a python application package used for GUI automation library written in Python and very much created for Windows GUI[15]. It is

a python module to automate windows GUI applications. Python modules like 'SendKeys' and 'ctypes' acts as an interface and pywinauto utilize these tools to construct GUI automation. At its most straightforward it enables it to send mouse and console activities to discourses and controls on the two Windows and Linux, while more complex content construct activities are used in Windows just up until now (Linux AT-SPI support is under development). Upheld advances in the engine: Win32 API, MS UI Automation (backend="uia"). Client input copying modules mouse and console chip away at the two Windows and Linux. In GUI computerization pywinauto is utilized broadly as frequently it should simply guarantee that UI carries on and is right as for the Source UI. This is a less demanding activity then to test the first source UI. In any case, most computerization instruments are based of directions or content of the controls and these can change in the limited programming. It can automate the windows click inputs and keyboard functions like robot in front of the system but not spilling the coffee on the keyboard. The pywin-auto application renders to run windows application consequently in python code.

3.2.3 SWAPY

Swapy is an another python application packaged combined with pywinauto for windows automation. It generates code used for native python code for pywin module. It is also a pywinauto Inspector to describe the application on windows also in dynamic process. The figure 3.6 explains the swapy window and its code generating process.

3.2.4 AUTOIt

AutoIt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation to

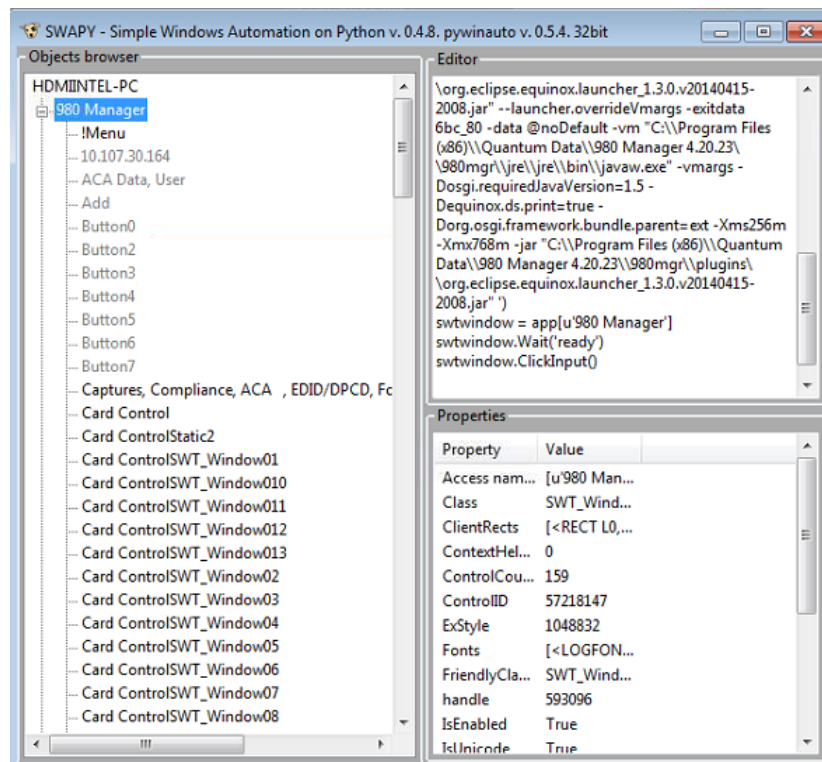


Figure 3.6: Swapy [16].

automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys). AutoIt is also very small, self-contained and will run on all versions of Windows out-of-the-box with no annoying “runtimes” required. It manipulates windows and processes, also it simulates keystrokes and mouse movements which can be used for getting mouse coordinates and injecting that into the python file. AutoIt was initially designed for PC “roll out” situations to reliably automate and configure thousands of PCs. Over time it has become a powerful language that supports complex expressions, user functions, loops, and everything else that veteran scripters would expect.

- `coords = (410,63)`
- `appWindow[u'SWTWindow022'].ClickInput(coords=self.coords)`

The above code explains the use of AutoIt to use directly in python for finding mouse points and clicks. The Finder tool used to locate the mouse and in the control tab the ControlClick Coords denotes the cursor click location which is the input coords of the program.

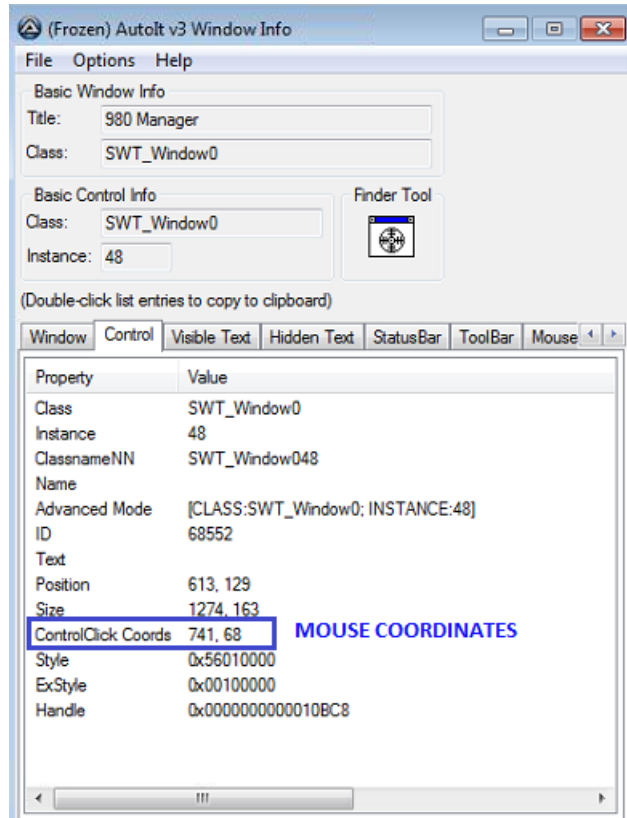


Figure 3.7: AutoIt Gui [17].

3.2.5 PERFORCE

Perforce is a software company that specializes in software configuration management, version control and collaboration products. In the project all the necessary software and hardware files are stored in Perforce which can be updated and released. The Jenkins trigger and grasp the file from perforce to start the automation process. Perforce keep track of versions for software

databases and more[18]. The Helix platform facilitates collaborative development and protection of any type of intellectual property (IP). The Helix platform includes:

- Helix Versioning Engine, which includes distributed version control capabilities for files of any type or size.
- Helix GitSwarm, a complete ecosystem for Git-based development.
- Helix Swarm, a comprehensive review application.

This chapter described the various software tools used for the project, the various platforms used, their necessity and their actual need to perform test automation technology. The software tools and API are selected by carefully reviewing the python user-friendly software tools which can perform the task easier.

Chapter 4

Project Flow

This chapter presents how the projects flow at Synopsys and explains the details of the project developed in the context of this thesis. The details of HAPS and IPK kit from Synopsys, the need of testing HDMI and EDID function are explained in this chapter. The information presented in this chapter should allow the preparation of a data sheet of the system (*datasheet*).

4.1 Synopsys Design Ware HDMI Rx IP Prototyping Kit

The DesignWare IP Prototyping Kit in which the complete test platform is initiated consists of validated IP configuration and necessary SoC integration logic for a specific IP protocol, implemented on a Synopsys HAPS Developer express (HAPS-DX) FPGA-based prototyping system[19]. The HDMI Rx IP Prototyping Kit is designed to be the primary mechanism within the Consumer Electronics Audio/video industry for ensuring EDIDs do what they are supposed to do by going through compliance testing. HDMI publishes on a general Specification and a Compliance Test Specification (CTS) associated with the general specification for each release. There are several

“Test IDs” associated with EDID verification in the HDMI CTS. Each is designed to test a particular aspect of EDID compliance.

The HDMI Rx IP Prototyping Kit is designed to

- accelerate the integration of the IP with an SoC,
- optimize the IP configuration, and
- develop drivers and software applications.

The HDMI Rx IP Prototyping Kit consists of ready-to-use hardware and software solutions, including design files and scripts required to reproduce the project, for FPGA design and software reference initialization samples and application examples. The HDMI Rx IP Prototyping Kit centers around a complete, out-of-the-box reference design that consists of a validated HDMI Rx configuration, necessary SoC integration, and test logic—implemented on an HAPS Developer eXpress (HAPS-DX) FPGA prototyping system. The kit also includes a Design Ware Advantec processor-based software development platform (running Linux), reference drivers, and application examples.

HDMI Rx IP Prototyping Kit provides the following benefits:

- Hardware engineers can validate IP configurations and explore design trade-offs for the target application. The kit accelerates the availability of a prototype for software development through quick modification of verified reference designs using IP configuration and prototyping tools.
- Software developers can speed up debugging and system integration of firmware and device drivers using the provided sample drivers and applications, a software development platform with debugging capabilities, and hardware signal observability. Moreover, developers do not need an in-depth understanding of the hardware when developing applications or drivers.

- SoC integration teams can reduce their integration risk by using a proven controller and PHY with interoperability and reference design.

4.1.1 Earlier Software Validation

The IP Prototyping Kits accelerate software integration and validation of Synopsys DesignWare IP by enabling real-time software development, validation, code porting, software debug, and analysis. The IP Prototyping Kits provide embedded software engineers with an ideal, ready-to-use solution for efficient coding. IP validation and system integration engineers can use IP Prototyping Kits to confirm compatible operation of interfaces, IP, and software. Users will have a choice of targets for their software development, including a fast connection to Linux or Windows-based host PC, or an embedded Linux-based ARC Software Development Platform.

4.1.2 Complete reference design

The FPGA reference design includes an instantiation of a DesignWare IP controller and all necessary additional blocks for:

- Clock and reset management;
- Power management;
- Traffic/test pattern generator;
- PHY interface;
- AXI system bus interface;
- Deep-trace debug for most relevant interfaces;
- Registers and internal state machine bring-up;
- Analysis and debug;
- Rebuilding the subsystem with provided scripts and source code.

4.1.3 HAPS-DX Prototyping System

The IP Prototyping Kits[20] take advantage of the Synopsys HAPS Developer eXpress (HAPS-DX) system to provide best-in-class prototyping hardware and automation software tools in a package that includes:

- Xilinx Virtex-7 690T FPGA device with up to 4 million ASIC gates of capacity with Configurable Logic Blocks (CLBs), RAM, and DSP resources;
- I/O interfaces compatible with both industry-standard FPGA Mezzanine Card (FMC) and HAPS HapsTrak 3 formats provide designers with a wide selection of daughter boards;
- Compatible design flow and hardware interfaces with HAPS-70 Series systems expand prototype capacity and make full SoC validation feasible;
- ProtoCompiler DX, which recognizes ASIC synthesis coding styles, DesignWare IP, SDC, and UPF. In fast HDL compiler modes, it provides up to 4 times faster throughput than traditional FPGA synthesis tools;
- RTL debug and high-capacity storage options provide up to 8 gigabytes of storage and a simulatorlike RTL debug;
- Interface for design troubleshooting and protocol compliance checks;
- Integrated Universal Multi-Resource Bus (UMRBus) hardware interface and C/ C++/Tcl APIs provide intimate control and visibility of the prototype from a host workstation.

4.2 EDID

The automation process on testing on Quantum Manager can be run by using EDID module. It is a set of standardized test data to verify the audio and video quality[21]. The EDID, which stands for Extended Display Identification Data, contains vendor and product id, serial number, manufacture date, size of the display, resolution and frequencies supported, and detailed signal timings for native resolutions. Plus, often when we speak of EDID we are actually referring to E-EDID, or Enhanced Extended Display Identification Data. Today's HDMI devices most often use E-EDID and include information to also specify audio capabilities including codecs, sampling rates, and channels.

4.2.1 Importance of EDID in HDMI

[22] When a video player or set top box is switched on and it has something connected to the HDMI port the source will request information about the display. The display will send reply with some manufacturer identifier like company name, model etc and their detailed information. These details declare or define the handling of screen resolution like 720p or 1800p and audio signals which are dependent upon the information specified. These capability information always set the video and audio signal up accordingly. The manufacturer and capability information is know as the EDID.

The previous explanation will work fine for single display connection and a complication occurs if HDMI networks are connected to multiple displays, e.g., if a Video signal is sent to two locations using a HDMI splitter. The complication happens if two lots of manufacturer and capability information going back to the source are different. The question here is Which set of information should the source use to set up its signal? You may well have a 1080p capable screen in the main lounge but only a 720p capable screen in the second location. If there is a capability problem like when a 1080p

signal is sent but the display will not be able to handle this, it will show a messaging saying "Out of Range" or something similar.

Another example may be when you want to send your HD signal to multiple locations and one location is a surround sound zone with a 5.1 surround sound amplifier and the other zones are TVs that can only handle 2 channel stereo. If the source sets up the audio format based on the EDID from the surround sound zone then the 2 channel TV displays will typically have a picture but a nasty buzzing noise for the audio as they cannot handle the digital 5.1 channel audio signal.

These two examples show how you easily get into trouble when trying to build HDMI networks. It stresses the point about building HDMI networks as when operating point to point between source and screen you will probably never hit any issues. It is only when you start to build HDMI networks that some of these issues come to play. This is not a limitation of the splitter or the HDMI protocol, this is an example of what happens when you try to build a network. No matter what technology you use, when you start building networks of devices the complexity always increases compared to a simple point to point system. As specialists in our field we need to understand how the protocols work so that we are aware of the behaviour and can engineer the system accordingly.

So how do we do that? Now that we know what is going on it is simply a matter of taking control of the EDID information that is being fed to the source device so it configures itself in a manner that meets our requirements. Clearly in the first example above it needs the source to set the video output at a resolution which is the lowest common denominator of the two screens, i.e. 720p in our example and we will want 2 channel audio. If we can ensure this happens then we will get video and audio on both our displays. Sometime this can be achieved by making sure the screen with the lowest capability is plugged into output port 1 of the splitter as some splitters

use the EDID from the display plugged into output port 1 when the source requests the EDID. Some splitters may have an "Internal/External" switch on them. When switched to "Internal" the splitter will use a generic EDID that has been designed by the manufacturer rather than actually going out to the screen to get the screen EDID. This is typically a generic EDID that has a capability information stating it can do everything. Other splitters may have physical switches on them allowing you to set the EDID up manually by selecting the right switch locations. More advanced HDMI networking equipment have an HTML user interface where you can log into the device and inspect the EDID coming back from the display and override it with something different and even create your own EDIDs.

All the above techniques are valid and may well allow you to engineer your HDMI solution successfully. However, only the last option above gives you total control of the situation. An example of HDMI networking equipment that has full EDID management built in and configured via an HTML user interface is the lightware range of products. However, these products are commercial grade equipment and may well be outside your project budget. An alternative solution for EDID management that puts you in full control and is a great deal more affordable is the use of an external hardware device that is connected to the HDMI output of the source such as the DVD player and has an HDMI output which carries on into the HDMI network. It is therefore usually used as the first item in the HDMI chain after the source. This device responds to the EDID request from the source and replies with what ever you want it to provide in terms of manufacturer and capability information. Therefore the EDID request never actually makes it beyond this device and the screen is never actually interrogated. This puts you back in control of the EDID and allows you to make sure the source sets up the video and audio format in a manner that meets your requirements. To program this magic box it is first put in learning mode and then you

plug it into a screen of your choice which provides the EDID you want to use. The device then stores this EDID in non-volatile memory. The device is then located on the back of the source as described above and the source will then always be given this EDID no matter what screen or screens(s) are being used to display the source output. Alternatively there are some factory preset EDIDs that can be selected by dip switches to alleviate the need to learn the EDID from a real screen. An example of such a device is the gefen HDMI Detective. The EDID can be differentiated for 3 types of HDMI. It starts with HDMI 1.4, HDMI 2.0 and HDCP RX 2.2. In this tests each HDMI versions have specific EDID format to display and run the test in quantum manager.

This chapter details about the specific test platforms used in Synopsys and the necessity of using EDID for testing HDMI. The HDMI IPK kit from Synopsys designed exclusively for prototyping or emulating the HDMI and other interface protocols and with EDID it performs test solutions for HDMI.

Chapter 5

Project Development

The test automation developed in cooperation with Synopsys are described in detail in this chapter

5.1 JENKINS MULTIJOB

In this process the multijob plugin from Jenkins was used. When creating new Jenkins jobs you have an option to create multiJob project. In the build section, this job can define phases that contain one or more jobs. All jobs that belong to one phase are executed in parallel. All jobs in phase 2 are executed only after the jobs in phase 1 are completed, and so on. In total six multijob were created for this project, everything controlled by the Master job which acts as a top module. The job created is shown in Figure 5.1)

These jobs were operated by 3 machines: one Master Machine, which was the central server and two slave machines. The Master job controls the slave machines to run job phases in parallel and also the flow of each phase. The flow of each job which was running in parallel is described in Figure 5.2)

The Advantech machine is connected through HAPS. All software pro-

S	W	Job
		DWIPK_HDMIQP_RX_multijob
		<i>P4 synchronization</i>
		DWIPK_HDMIQP_RX_multijob_p4sync
		<i>Setup intel - update software</i>
		DWIPK_HDMIQP_RX_multijob_HW_intel
		<i>updating software in advantec</i>
		DWIPK_HDMIQP_RX_multijob_ADVANTEC
		<i>Setup HAPS - bitfile upload</i>
		DWIPK_HDMIQP_RX_multijob_HW_haps
		<i>Compliance Tests - quantum 980</i>
		DWIPK_HDMIQP_RX_multijob_HW_quantum

Figure 5.1: Jenkins multijob phases.

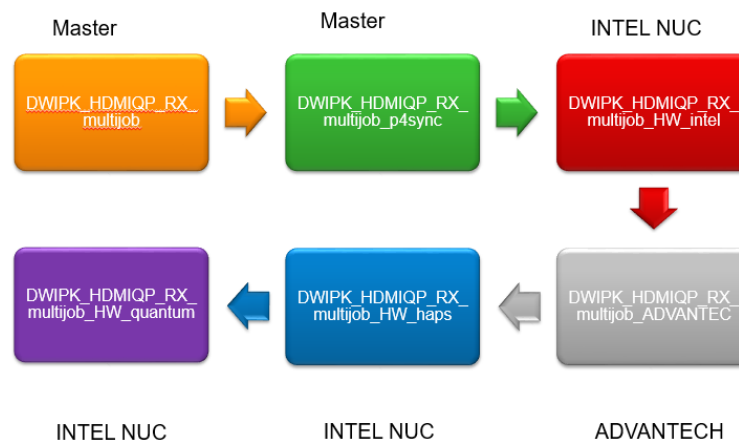


Figure 5.2: Jenkins multijob phases flow.

tools and processes are running on this slave. The master control this Advantech machine to update the software and run the compliance test. All other processes start with the compliance test and until the report process is running on Intel Machine. Figure 5.3 shows the slave assignment and their workflow.

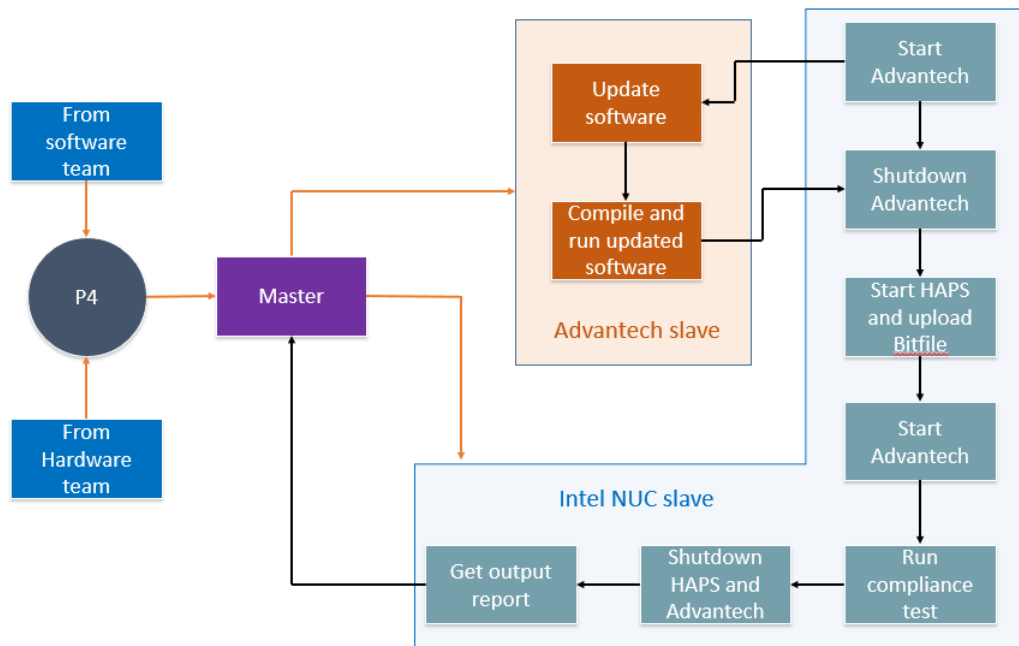


Figure 5.3: Slaves Assignment.

The session follows the task perform from each jenkins multijob. These jobs runs in parallel to make the flow expedite.

5.1.1 DWIPK-HDMIQP-RX-multijob

- To control the flow of jobs and to run all phase jobs in parallel.
- To synchronize and archive all file needed.
- To archive the artifacts and save in master.

5.1.2 DWIPK-HDMIQP-RX-multijob-p4sync

- For Perforce synchronization.
- To grasp the concern files from P4 and to save in a specific folder.
- To create a final report.

- To communicate (mail) to the concern author the final result.

5.1.3 DWIPK-HDMIQP-RX-multijob-HW-intel

- To start the Advantech Machine through WLAN
- With Wake-on-LAN (WOL) one can wake up a computer over the network.
- The prerequisite is that the computer has a network card with WOL support and it is activated in the BIOS or in the operating system.
- To wake up the computer, a so-called Magic Packet is sent over the network to its MAC address or as a broadcast message.
- The Magic Packet contains six times in succession the hexadecimal value 0xFF and immediately thereafter 16 times the MAC address of the system which is to be awakened.

5.1.4 DWIPK-HDMIQP-RX-multijob-ADVANTECH

- Copy the updated software from Perforce through Master Machine.
- Compile and run the updated software.
- This node can be modified or changed if the slave(Advantech) changes.
- To maintain continuous integration nevertheless the slave machine can be used from any location. It needs changes only on IP, user name and password and hardware address.
- Whenever the softwares are updated in Perforce, this job grab the updated software from Perforce to Master and from Master to Slave. Finally it compile and run the updated software in slave machine. Figure 5.4) describes this process.

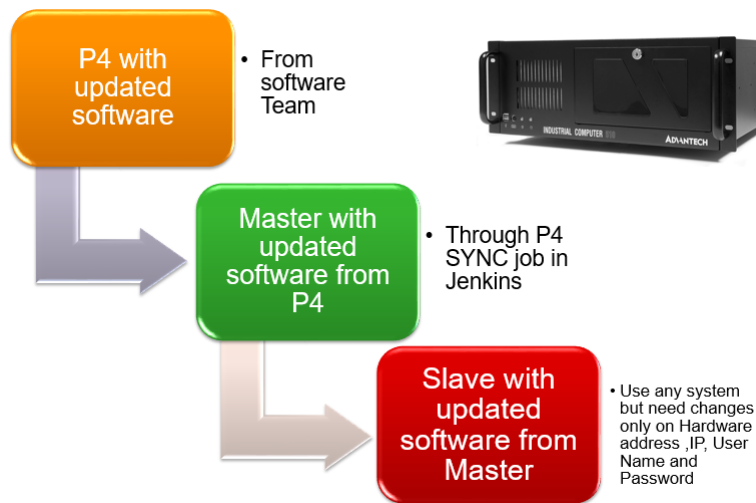


Figure 5.4: Continuous Integration.

5.1.5 DWIPK-HDMIQP-RX-multijob-haps

- This Job performs the following steps :
 - Step 1 : Shutdown Advantech slave through telnet call in python program
 - Step 2 : To power up HAPS through Ethernet switch.
 - Step 3 : Upload Bitfile into the HAPS and get board configuration.
 - Step 4 : Start the Advantech Machine through WLAN.

5.1.6 DWIPK-HDMIQP-RX-multijob-quantum

- This job controls the whole process of Testing in Quantum through GUI in Intel NUC slave.
- The pywinauto in python controls and process the windows GUI automation.

- Compliance test have been executed based upon order HDMI 1.4, 2.0 and HDCP 2.2. As the three tests are independent to each other anyone can be called whenever needed.
- After all compliance tests, this job make a directory and store all information, like HTML report and summary, in a folder and make a ZIP file.

5.2 Running compliance test

The last job of Jenkins DWIPK-HDMIQP-RX-multijob-quantum runs the compliance test. There are three crucial tasks to perform which are mainly performed manually:

1. Quantum GUI control;
2. Audio comparison;
3. Image comparison.

5.2.1 Quantum GUI control

The Quantum Manager is controlled by the windows GUI which is installed in the slave machine. The GUI can perform compliance test but in-order to automate the keyboard and mouse control they have to be driven to the GUI by pywinauto from python. The process is explained in Figure 5.5)

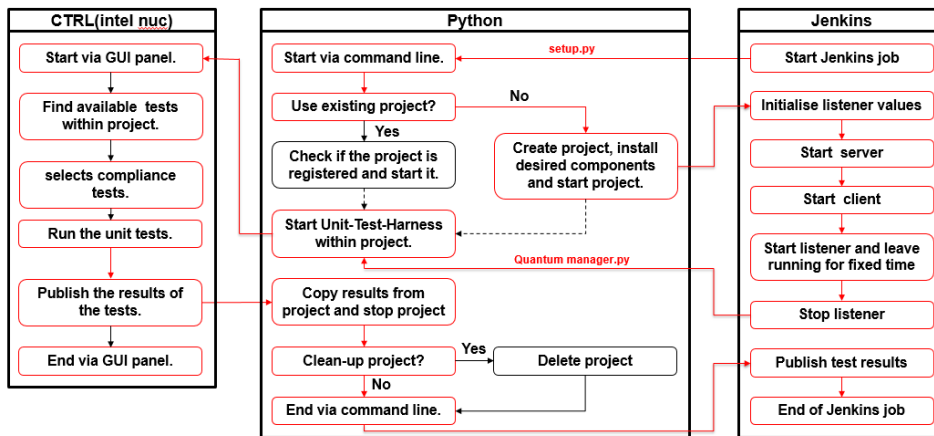


Figure 5.5: Process flow diagram.

How Pywinauto works in windows automation? It uses two software :

- Swapy – To open a GUI and to control clicks.
- AutoIT – To find mouse coordinates of the window.

Figure 5.6) represent a sample code to open a folder in compliance test. It uses Swapy and AutoIt to click the folder and open it.

```

appWindow = app.Window_(title=u'980 Manager', class_name='SWT_Window0')
appWindow[u'SWT_Window027'].ClickInput()
appWindow[u'SWT_Window022'].ClickInput(coords=self.coordsHDMI14)
appWindow[u'SWT_Window022'].ClickInput(coords=self.coordsEdid)
  
```

Figure 5.6: E.g. Code to open a folder in compliance test.

5.2.2 Audio Comparison

The HDMI signal contains video and audio data. The audio from the input must be compared in order to check if the protocol performs perfectly. The Fourier transform is used here to compare the input audio signal with the reference audio. The fast Fourier transform (FFT) is an algorithm for converting a time-domain signal into a frequency-domain representation of the relative amplitude of different frequency regions in the signal. The FFT is an implementation of the Fourier transform. The Fourier transform is one of the most useful mathematical tools for many fields of science and engineering. The Fourier transform displays the frequency components within a time series of data. The result of the FFT contains the frequency data and the complex transformed result. The FFT works perfectly when analyzing exactly one tone (an audio tone as single tone is considered as single cycle). The FFT takes a chunk of time called a frame (number of samples) and considers that chunk to be a single period of a repeating waveform. Most sounds are locally stationary, which means that the sound does look like a regularly repeating function in a short period of time. The FFT is the one-dimensional Fourier transform. If assuming a signal is saved as an array in the variable X , then performing $\text{fft}(X)$ will return the Fourier transform of X as a vector of the same size as X . However, the values returned by $\text{fft}(X)$ are in the frequency domain.

The following steps describe in short how the audio analysis had done using FFT:

- Step 1: Load audio files
- Step 2: Read in two audio files into the workspace.
- Step 3: Truncate both signals so that their durations are equivalent.
- Step 4: Perform FFT

- Step 5: Compute normalized energy spectral density (ESD) from DFT's two signals
- Step 6: Compute mean-square-Value (MSV)
- Step 7: Compute MSV between normalized ESD's of the two signals
- Step 8: Two perfectly identical signals will obviously have MSV of zero.
- Step 9: Display error message, if MSV value is not zero.

The flowchart in Figure 5.7) explains the process of audio comparison using FFT.

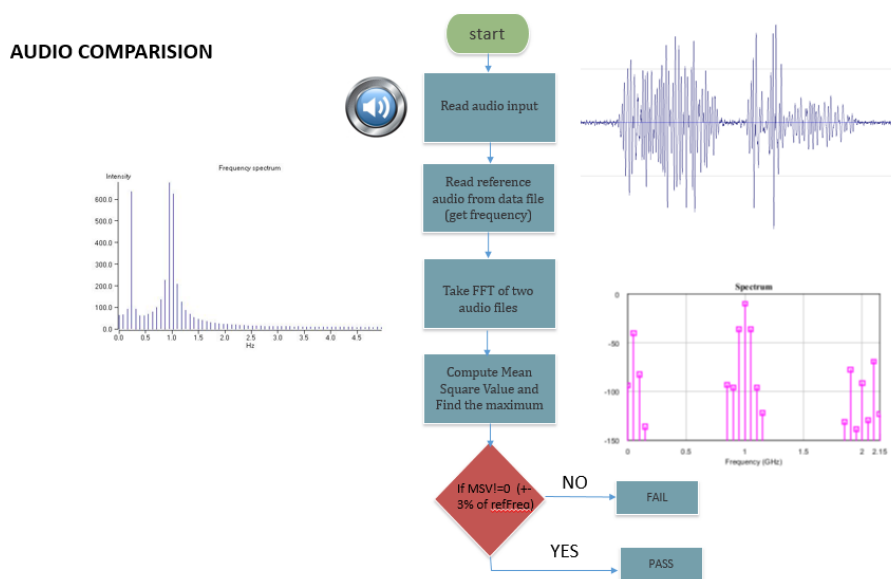


Figure 5.7: Audio comparison using FFT.

5.2.3 Image Comparison

The video input during compliance test can be converted to image frame for comparing the file. The reference images are stored by default and the input images from HDMI protocol can be compared with these reference images to

check the input data performance. These tasks can be performed by specific software called Numpy. Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. Load both images as arrays and calculate an element-wise (pixel-by-pixel) difference. Calculate the norm of the difference. For the video sequence, where frames are likely to be almost the same, you look for something unusual. These process still needs some specific tasks like:

- background subtraction and segmentation (to detect foreground objects);
- sparse optical flow (to detect motion);
- comparing histograms or some other statistics instead of images.

The Python Imaging Library, or PIL for short, is one of the core libraries for image manipulation in Python. For the image comparison the PIL image can be converted into a Numpy array. These arrays can be compared. The output is compared pixel by pixel and a normalized value can be obtained from that result. The code in Figure 5.8) is an example of converting single channel PIL image to a Numpy array and back.

```
1 | import numpy
2 | import PIL
3 |
4 | # Convert Image to array
5 | img = PIL.Image.open("foo.jpg").convert("L")
6 | arr = numpy.array(img)
7 |
8 | # Convert array to Image
9 | img = PIL.Image.fromarray(arr)
```

Figure 5.8: Image conversion using PIL and numpy.

This chapter explained about the jobs created in Jenkins and how it works.

Chapter 6

Conclusion and Future Work

This chapter explains the overview of the work developed during the internship, presenting what makes it better along with the future work to correct/improve them. Synopsys, as one of the leading companies in the silicon industry, is in constant improvement to present to its clients better products and new solutions. The prototyping kits are an enhancement, made on the IP production, that accelerate time to market and facilitate the controllers production. These provide consistent examples on IP use cases allowing the client to modify and adapt it to its systems and designs. During the internship time at Synopsys, topics and subjects were studied in order to successfully accomplish the proposed projects.

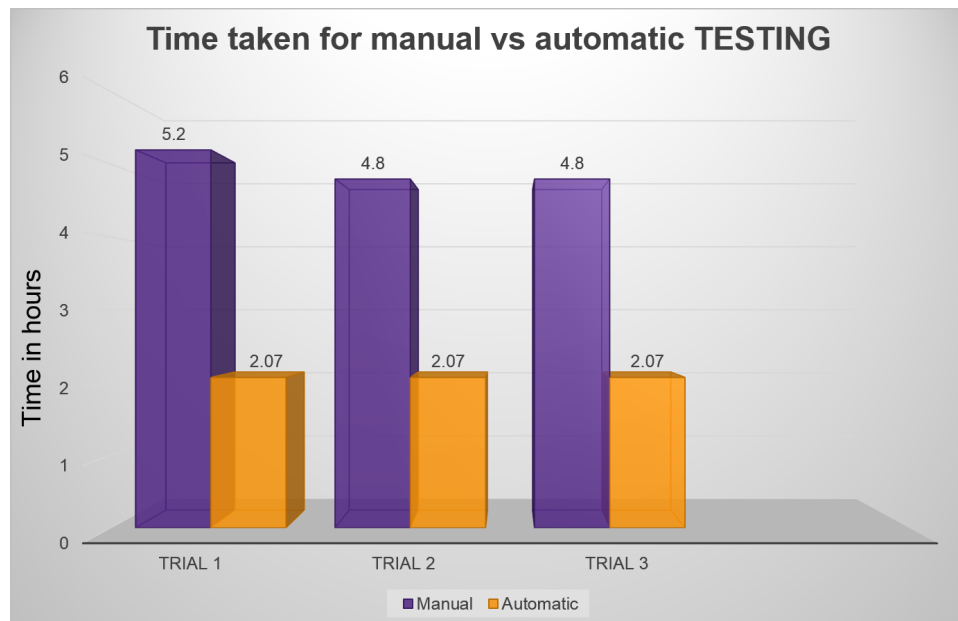


Figure 6.1: Time taken for testing.

Manual testing	automated
Less thorough than automation testing	Find more bugs than a human test
Testing fatigue	Get the speed and efficiency of computers
Not reusable.	Reusable tests for code that gets frequent updates
More clocking needed.	Gives your team a break and Consolidate the testing process

Figure 6.2: Manual vs Automatic testing.

The following results from the graph describe the advantages of using test automation. The required number of manual work completely eliminated. The development of the future analysis predict that the process involved in the determination of risk factors that manual work can have in cognitive envi-

ronment. This automation process saved more than 50 percent compliance test time and also on top of that it achieved 100 percent result in High Definition Video (4K) comparison and 95 percent in audio comparison.

This page was intentionally left blank.

Bibliography

- [1] Jason Knott. Hdmi uncensored. 2012.
- [2] INTEL. Ug hdmi. <https://www.altera.com/content/dam/altera-www/global>, 2017.
- [3] EEWEB. The displayport display interface: Robust and future-proof. <https://www.altera.com/content/dam/altera-www/global>, 2016.
- [4] Audio Holice,. Hdmi-versions. <http://www.audioholics.com/hdtv-formats/understanding-difference-hdmi-versions>, 2013.
- [5] U N Vasudev,. Compliance testing and debug tips for HDMI 2.0. 2014.
- [6] HDMI ORG,. Testing Policies. <http://www.hdmi.org/manufacturer/testingpolicies>, 2017.
- [7] TELEDYNE LECROY,. Testing. <http://www.quantumdata.com/980series>, 2017.
- [8] TELEDYNE,. 980 Guide. <http://www.quantumdata.com/assets/980hdmipaug.pdf>, 2017.
- [9] JENKINS,. Jenkins. <https://wiki.jenkins-ci.org/pages/viewpage.action?pageId=58001258>, 2019.
- [10] JENKINS,. Jenkins User Documentation. <https://jenkins.io/doc/>, 2019.

-
- [11] JENKINS,. UI Refresh. <https://www.jenkins.io/blog/2014/08/11/user-interface-refresh/>, 2014.
- [12] Aroraglobalservices,. Continuous Integration Tool .
<http://aroraglobalservices.blogspot.pt/2014/11/continuous-integration-tool-jenkins.html>, 2017.
- [13] PYHTON,. PYTHON. <https://www.python.org>, 2019.
- [14] Codeeval. Most popular coding languages.
- [15] The pywinauto contributors community,. Pywinauto.
<https://buildmedia.readthedocs.org/media/pdf/pywinauto/uia/pywinauto.pdf>, 2016.
- [16] GitHub,. Swapy. <https://pywinauto.github.io/pages/swapy-is-a-pywinauto-inspector-and-code-generator.html>, 2015.
- [17] AUTOIT,. AutoItLanguage. <https://www.autoitscript.com/site/autoit>, 2016.
- [18] PERFORCE,. P4Connect in Unity. <https://www.perforce.com/video-tutorials/installing-and-configuring-p4connect-unity>, 2016.
- [19] Synopsys,. DesignWare IP Prototyping Kits.
<https://www.synopsys.com/dw/doc.php/ds/o/ipprototypingkitsds.pdf>, 2017.
- [20] Synopsys,. Hdmi2 to arc. <https://www.synopsys.com/dw/doc.php>, 2017.
- [21] EXTRON,. EDID Understanding.
<http://www.extron.com/download/files/articles/understandingedid.pdf>, 2015.

- [22] Aldoussystems,. EDID in HDMI.
<http://aldoussystems.blogspot.pt/2010/07/why-edid-management-in-hdmi-is-so.html>, 2015.