MISSIONE 4
ISTRUZIONE
RICERCA

# Development and Applications of a SystemC-AMS Virtual Testing Framework for Industrial Automatic Test Solutions

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di / Department of

Fisica "Giuseppe Occhialini"

Dottorato di Ricerca in / PhD program Fisica e Astronomia Ciclo / Cycle 38°

Curriculum in (se presente / if it is) Fisica Applicata ed Elettronica

# Development and Applications of a SystemC-AMS Virtual Testing Framework for Industrial Automatic Test Solutions

Cognome / Surname Turossi Nome / Name Davide

Matricola / Registration number 814416

Tutore / Tutor: Andrea Baschirotto

Cotutore / Co-tutor:
(se presente / if there is one)

Supervisor: Daniela Barge
(se presente / if there is one)

Coordinatore / Coordinator: Laura D'Alfonso

**ANNO ACCADEMICO / ACADEMIC YEAR 2024-2025**

# Contents

**List of Figures** **iv**

**Acronyms** **vii**

**Abstract** **ix**

**1 Introduction** **1**

1.1 Automatic Testing Fundamentals . . . . . . . . . . . . . . . . . . 2

1.1.1 Wafer-level testing . . . . . . . . . . . . . . . . . . . . . . 4

1.1.2 Package-level testing . . . . . . . . . . . . . . . . . . . . . 6

1.1.3 Test programs . . . . . . . . . . . . . . . . . . . . . . . . 7

1.1.4 Industrial workflow . . . . . . . . . . . . . . . . . . . . . . 7

1.2 Gate Drivers . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

1.2.1 Testing of gate drivers . . . . . . . . . . . . . . . . . . . . 11

1.3 Virtual Testing . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

1.3.1 State of the art . . . . . . . . . . . . . . . . . . . . . . . . 13

1.4 Activity Overview . . . . . . . . . . . . . . . . . . . . . . . . . 14

**2 Basic Methodology** **16**

2.1 Software and Tools . . . . . . . . . . . . . . . . . . . . . . . . . 16

2.1.1 SystemC-AMS . . . . . . . . . . . . . . . . . . . . . . . . 17

2.1.2 COSIDE modelling environment . . . . . . . . . . . . . 19
2.2 Models Basic Structure . . . . . . . . . . . . . . . . . . . . . . 19
2.2.1 Input parameter file . . . . . . . . . . . . . . . . . . . . 21
2.3 Framework Usage . . . . . . . . . . . . . . . . . . . . . . . . . 23

**3 Components Modelling 24**

3.1 Schematic Components . . . . . . . . . . . . . . . . . . . . . . 25
3.1.1 Non-linear modelling . . . . . . . . . . . . . . . . . . . . 26
3.2 Abstraction Levels . . . . . . . . . . . . . . . . . . . . . . . . . 28

**4 ATE Models Development 29**

4.1 Top-Level . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
4.2 Smart Pin Unit 112 . . . . . . . . . . . . . . . . . . . . . . . . 31
4.3 Analog Pin Unit 12 . . . . . . . . . . . . . . . . . . . . . . . . 34
4.4 Digital Pin Unit 16 . . . . . . . . . . . . . . . . . . . . . . . . 36
4.5 Control Bits . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

**5 Application - Gate Driver Test Solution Development 40**

5.1 General Aspects . . . . . . . . . . . . . . . . . . . . . . . . . . 40
5.1.1 Probe-card design . . . . . . . . . . . . . . . . . . . . . . 41
5.1.2 Test program . . . . . . . . . . . . . . . . . . . . . . . . 43
5.2 Case-Studies . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
5.2.1 Clamp voltage . . . . . . . . . . . . . . . . . . . . . . . . 44
5.2.2 Current pulse generator . . . . . . . . . . . . . . . . . . . 46
5.2.3 Wake-up time . . . . . . . . . . . . . . . . . . . . . . . . 48
5.3 Data Analysis Results . . . . . . . . . . . . . . . . . . . . . . . 50

**6 Application - Peak Current Test Concept Development 54**

6.1 Motivation and Concept . . . . . . . . . . . . . . . . . . . . . . 55

6.1.1 Board concept . . . . . . 56
6.1.2 Test program . . . . . . 56
6.2 Test-Board Design . . . . . . 57
6.2.1 Power booster . . . . . . 57
6.2.2 Current peak detector . . . . . . 59
6.2.3 Test bench . . . . . . 61
6.3 Eagle PCB Conversion . . . . . . 63
6.3.1 Test-board design . . . . . . 63
6.3.2 Test-board layout . . . . . . 63
6.4 Laboratory Measurement . . . . . . 67
6.4.1 Calibration process . . . . . . 67
6.4.2 Repeatability tests . . . . . . 69
6.4.3 Looping tests . . . . . . 69
6.4.4 Bench correlation . . . . . . 71

**7 Conclusion** **74**

**8 Other Projects** **75**
8.1 Automatic Test Laboratory . . . . . . 75
8.2 IonoTRACK ADC . . . . . . 77

**Acknowledgement** **78**

**Bibliography** **79**

# List of Figures


1.1 Propagation of single-device defects to larger systems . . . . . 2

1.2 Teradyne ETS-88 automatic test equipment . . . . . . . . . . . 3

1.3 Elements of wafer-level testing . . . . . . . . . . . . . . . . . . . . 4

a Industrial silicon wafer . . . . . . . . . . . . . . . . . . . . 4

b Probe-card for an industrial test solution . . . . . . . . 4

1.4 Accretech UF200 fully automatic wafer prober . . . . . . . . . . 5

1.5 Elements of package-level testing . . . . . . . . . . . . . . . . . . 6

a Packaged industrial devices . . . . . . . . . . . . . . . . 6

b Application board for package-level testing . . . . . . . 6

1.6 MST test program development interface . . . . . . . . . . . . 8

1.7 Industrial automatic test development process . . . . . . . . . . 9

1.8 Low and high-side gate driver . . . . . . . . . . . . . . . . . . . . 10

1.9 Simplified flowchart of virtual testing . . . . . . . . . . . . . . 12

2.1 SystemC-AMS hierarchical structure . . . . . . . . . . . . . . . 17

2.2 COSIDE graphical user interface . . . . . . . . . . . . . . . . . 20

2.3 Framework hierarchical library structure . . . . . . . . . . . . . 21

2.4 Input parameter file section . . . . . . . . . . . . . . . . . . . . . 22

3.1 Sample collection of modelled components . . . . . . . . . . . 25

3.2 TDF PN diode behavioural model . . . . . . . . . . . . . . . . . 27

3.3 PN diode multiple abstraction levels . . . . . . . . . . . . . . 28

4.1 Teradyne ETS-88 top-level view . . . . . . . . . . . . . . . . . . 30
4.2 Smart Pin Unit 112 top-level view . . . . . . . . . . . . . . . . . . 31
4.3 Smart Pin Unit 112 single-channel view . . . . . . . . . . . . . 32
4.4 Analog Pin Unit 12 single-channel view . . . . . . . . . . . . . 34
4.5 Analog Pin Unit 12 top-level view . . . . . . . . . . . . . . . . . 35
4.6 Digital Pin Unit 16 single-channel view . . . . . . . . . . . . . 36
4.7 Digital Pin Unit 16 top-level view . . . . . . . . . . . . . . . . . 37
4.8 Control Bits single-channel view . . . . . . . . . . . . . . . . . . 38
4.9 Control Bits top-level view . . . . . . . . . . . . . . . . . . . . . 39

5.1 Gate driver wafer-level probe-card schematic . . . . . . . . . . 42
5.2 Clamp voltage - SPU-112 measurement . . . . . . . . . . . . . 45
5.3 Clamp voltage - QTMU measurement . . . . . . . . . . . . . . . 45
5.4 Current pulse generation - time diagram . . . . . . . . . . . . 47
5.5 Current pulse - simulation and measurement comparison . . . 47
5.6 Wake-up time - APU-12 supply . . . . . . . . . . . . . . . . . . 49
5.7 Wake-up time - gate driver supply . . . . . . . . . . . . . . . . . 49
5.8 Failure map for a wafer test . . . . . . . . . . . . . . . . . . . . 51
5.9 Summarised yield data for a wafer test . . . . . . . . . . . . . 52
5.10 Clamp voltage distribution . . . . . . . . . . . . . . . . . . . . . 52
5.11 Trip current distribution . . . . . . . . . . . . . . . . . . . . . . 53
5.12 Wake-up time distribution . . . . . . . . . . . . . . . . . . . . . 53

6.1 Isolated gate driver output current . . . . . . . . . . . . . . . . 55
6.2 Peak current measurement circuit block diagram . . . . . . . . 56
6.3 SPU-112 current boosting stage . . . . . . . . . . . . . . . . . . 58
6.4 OpAmp-based current peak detector . . . . . . . . . . . . . . . 60

6.5 Sampling capacitor dimensioning process . . . . . . . . . . . . 60
6.6 Peak current measurement circuit - test bench . . . . . . . . . 62
6.7 Peak current measurement circuit - Eagle PCB design . . . . . 64
6.8 Peak current measurement circuit - Eagle PCB layout . . . . . 65
6.9 Manufactured board for the peak current measurement . . . . 66
6.10 Peak current measurement oscilloscope reading . . . . . . . . . 67
6.11 Peak detector calibration oscilloscope reading . . . . . . . . . 68
6.12 Voltage-current relationship . . . . . . . . . . . . . . . . . . . . 68
6.13 Looping test results . . . . . . . . . . . . . . . . . . . . . . . . . 70
6.14 Application board for bench correlation . . . . . . . . . . . . . 72
6.15 Schematic of the measurement setups . . . . . . . . . . . . . . 72
6.16 Shunt-based bench correlation oscilloscope reading . . . . . . . 73
6.17 Probe-based bench correlation oscilloscope reading . . . . . . . 73

8.1 ATLab panoramic view . . . . . . . . . . . . . . . . . . . . . . . . 76
8.2 IonoTRACK ADC schematic view . . . . . . . . . . . . . . . . . . 77

# Acronyms

**APU-12** Analog Pin Unit 12

**ATE** Automatic Test Equipment

**AWG** Arbitrary Waveform Generator

**CBITS** Control Bits

**DFT** Design For Testability

**DPU-16** Digital Pin Unit 16

**DUT** Device Under Test

**ELN** Electrical Linear Network

**HDL** Hardware Description Language

**PWL** Piece Wise Linear

**QTMU** Quad Time Measurement Unit

**SPU-112** Smart Pin Unit 112

**TDF** Timed Data Flow

**TF** Test Function

**TP** Test Program

**Abstract**

This dissertation presents a methodology based on SystemC-AMS and supported by the COSIDE graphical design environment for the development of automatic test solutions for industrial electronic devices. The proposed approach optimises the design of test hardware and the development of test programs by allowing efficient capture of test hardware schematics and rapid mixed-signal simulation of test program sections. This offers detailed insight into the behaviour of the test setup and into the mutual interactions between test hardware, automatic test equipment, and device under test, enabling comprehensive pre-silicon troubleshooting of the test solution.

The development of the framework from the ground-up is presented in detail, emphasising the selection of appropriate software and programming methodologies, along with the modelling principles for automatic test equipment resources and electronic components for test hardware development. The validity of the proposed methodology is demonstrated through its application in two innovative industrial test scenarios.

In the first case study, the framework is employed in the design of the probe-card for the wafer-level testing of an innovative industrial gate driver with built-in over-current protection. The adoption of the framework enables the investigation of several critical aspects of the test hardware, including schematic robustness, analog and digital signal integrity, measurement solutions for key electrical quantities, and the generation of a variable-amplitude current pulse in which active components modulate the native automatic test equipment capabilities. A comparison between simulation results and bench measurements performed during test program execution on the fabricated probe-card demonstrates the capability of the framework to provide an effective representation of the modelled components and to properly support the development of automatic test solutions.

In the second case study, a test concept to enable the measurement of the high level output peak current of isolated gate drivers is developed from the initial concept to the physical implementation. Alternative circuit topologies and component dimensioning are evaluated within the framework and subsequently validated through measurements performed on the manufactured hardware. A final sequence of qualification measurements is performed to finalise the test concept and confirm its suitability for integration into future standard automatic test solutions.

# Chapter 1

# Introduction

Testing an electronic device consists of applying specific electrical stimuli and evaluating the device responses, comparing the observed and expected behaviours. Electrical testing is an extremely transversal activity and represents a fundamental step in the development of any electronic device, independently of its complexity and scale of production.

The methods and good practices used to implement electrical testing during the development of electronic devices are highly diverse. For simplicity, the contribution of testing to the overall development process can be summarised in two main applications:

Firstly, faults, weaknesses, and deviations between the observed behaviour and the expected one must be properly identified, evaluated, and quantified. These observations support improvements in subsequent design iterations of the device, and implementation of appropriate post-production corrections.

Secondly, a quantitative description of the device performance must be produced to better understand its present and future functionality under different operating conditions. A proper amount of data must therefore be collected from a sufficiently large number of sample devices, and statistically analysed.

The testing process assumes further importance in the case of industrial electronic devices, which are typically manufactured at very large scales. These devices are often embedded within larger systems, in which faults and malfunctions affecting a single unit (even if statistically unlikely) may propagate through the larger system, as intuitively illustrated in Fig.1.1. Depending on the specific application, this propagation effect can result in

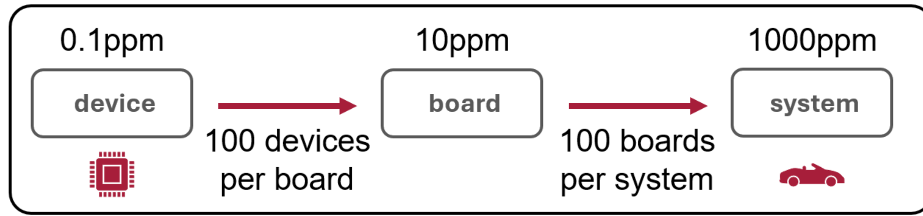



Figure 1.1: Quantitative example of the propagation of single-device defects to larger systems in which they are embedded (e.g., a car)

significant safety hazards and customer dissatisfaction. It is therefore essential to properly identify, evaluate, and quantify these operational defects, to minimise their impact before the devices are released to the market.

In industrial environments, the testing process accounts for a significant portion of the total budget allocated to the development of electronic devices. Although exact figures are generally treated as restricted data and therefore not publicly disclosed, it is acknowledged within the sector that approximately 20%-30% of the manufacturing budget is allocated for testing related activities [1, 2]. This figure corresponds to approximately 5% of the total budget allocated for the development of semiconductor devices [3, 4][1].

As a result, the electronic industry has, since its early stages, attempted to streamline, automate, and improve the testing process to reduce the associated costs while maintaining a proper level of accuracy on the largest possible number of devices.

The collection of methods and techniques developed to achieve these objectives is referred to as *automatic testing*, and constitute a fundamental field of industrial research and engineering.

## 1.1 Automatic Testing Fundamentals

In industrial environments, to manage the large number of devices that must be tested, signal sourcing and acquisition are typically performed by specialised machinery referred to as Automatic Test Equipment (ATE). These complex systems can be described as a collection of signal sources (voltage and current levels, arbitrary waveforms, digital signals, etc.) and measurement instruments (voltage, current, frequency, propagation time, etc.), integrated into a comprehensive system that provides synchronisation and

[1]This figure is largely dependant on the complexity and application of the electronic devices under consideration.

Figure 1.2: Teradyne ETS-88, highly versatile automatic test equipment widely used for the automatic testing of semiconductor devices

interconnection between the various resources, along with a unified control software. As an example, the Teradyne ETS-88 is shown in Fig.1.2.

The test flow is managed by the Test Program (TP), a computer program written in a system-specific programming language, which encodes all the necessary steps to properly test the Device Under Test (DUT) through the subsequent execution of the test functions. A Test Function (TF) contains specific instructions for the ATE resources to configure the test setup correctly, source and acquire all the required signals, and generate the output data-log for further data analysis.

Given the continuous increase in device complexity and associated test requirements, several additional techniques have been developed by the in-

(a) Industrial silicon wafer

(b) Probe-card

Figure 1.3: Elements of wafer-level testing:
(a): Industrial silicon wafer with approximately 12,000 embedded devices
(b): Probe-card to interface the silicon wafers with the ATE

dustry to optimise the overall efficiency of the testing process [5, 6].

For example, Design For Testability (DFT) aims to embed self-test functionalities into electronic devices to reduce test time and resource usage, while multi-site testing significantly shortens the test time by running parallel simultaneous testing on multiple devices[2].

As previously mentioned, industrial automatic testing is a field of engineering with an extremely broad range of methodologies and approaches. In general, two different testing environments can be distinguished, according to the specific phase of production of the DUT they are targeted to characterise.

### 1.1.1 Wafer-level testing

Factory-produced devices are initially embedded in silicon wafers, which can host approximately 1,000 to 10,000 individual devices, depending on the device complexity. A silicon wafer is shown in Fig.1.3a. The main objectives of wafer-level testing are to collect statistical data for the initial definition of the device data-sheet, to characterise sample engineering wafers and provide valuable feedback for subsequent device iterations, and to eliminate defective devices before the packaging process.

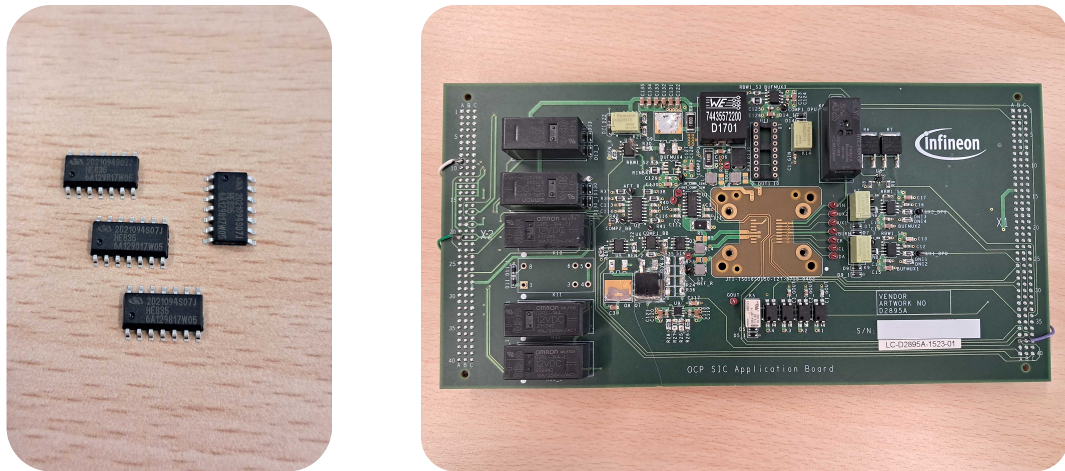
[2] Increasing the number of devices that can be simultaneously tested is one of the main current challenges in the automatic testing industry.

Figure 1.4: Accretech UF200, fully-automatic wafer prober specialised for productive environments

The defining piece of equipment for wafer-level testing is the wafer prober, which is capable of handling silicon wafers, and establishing electrical contact with one or multiple DUTs through the probing needles, effectively serving as the interface between the DUT and the test hardware during the TP execution by the ATE.

Advanced features of wafer probers include the automatic handling of multiple wafers through self-managed loading and unloading processes (fully automatic wafer probers), and the control of the DUT temperature by a thermal chuck, enabling low and high temperature validation. As an example of a fully automatic wafer prober used in production environments, the Accretech UF200 is shown in Fig.1.4.

In wafer-level testing, the test hardware is represented by the probe-card, a device-specific interface hardware mounted on the wafer prober, that establishes all the required connections between the ATE (through specialised cables) and the on-wafer DUT (through the probing needles). An example of a probe-card is shown in Fig.1.3b.




(a) Packaged devices (b) Application board

Figure 1.5: Elements of package-level testing:
(a): Packaged devices for device debug and test solution development
(b): Application board to interface the packaged devices with the ATE

### 1.1.2 Package-level testing

At the package-level, devices are cut from silicon wafers and embedded in appropriate packages, made from different materials and associated with different footprints and pinouts according to the specific application.

The main purpose of package-level testing is the pre-market validation of the produced device lots[3], although the characterisation of sample packaged devices for debug and improvement across different device iterations can also be performed. As an example, packaged industrial devices are shown in Fig.1.5a.

The defining piece of equipment for package-level testing is the die handler, capable of mechanically feeding the DUTs to the test sockets and establishing the required electrical connection to execute the TP.

For device and test solution debug, application-boards can serve as temporary interface hardware between the ATE and the DUT. An example of an engineering application board is shown in Fig.1.5b.

Given the dynamic nature of package-level testing, additional components can be deployed during the debug phase. For instance, thermal control devices can establish thermal contact with the DUT, and regulate its temperature for low and high temperature validation.

[3] In this phase, leakage, breakdown, and other tests specific to the package are included into the test plan.

### 1.1.3 Test programs

As previously explained, the flow of operation of the ATE is controlled by the test program, which is a collection of instructions written in a dedicated programming language, and organised into multiple test functions. These instructions are interpreted by the ATE and translated into actions performed on the test setup[4].

Engineering TPs are primarily used during the debug and validation phase. Once the development is complete, they are converted into production test programs, which are targeted towards efficient data collection and test time optimisation for the final test.

The execution of a generic TF designed to perform a specific measurement includes setting the optimal test setup configuration, performing the measurement, restoring the test setup to its original state, and collecting the test data. The main challenges in the development of TFs and TPs are the management of code complexity and the optimisation of the test time.

For the Teradyne ETS-88, previously used as an example of ATE, the TPs are managed using the proprietary MST software, and written in the C++ programming language making use of a collection of dedicated libraries and macros. The development interface for an ETS-88 TP is shown in Fig.1.6.

### 1.1.4 Industrial workflow

A variety of different approaches and methodologies are adopted across different industries for automatic testing. A general structure of the industrial workflow that covers the main mandatory steps to implement a successful test solution can however be identified, and is schematically shown in Fig.1.7.

In detail, the following steps are typically involved:

- **Test plan definition** - The general aspects of the DUT test requirements are defined, including the main parameters and features to be characterised, pass and fail criteria, the necessary wafer and package-level test coverages, and the overall yield target. DFT functionalities can also be proposed and implemented at this stage to enable or improve the testability of specific parameters.

- **Test program development** - Based on the defined test plan, the most suitable ATE is selected, and the test sequence is specified and implemented within the development environment. The TP is typically updated and optimised throughout the subsequent phases.

[4] In this context, the test setup includes the DUT, embedded into the test hardware.

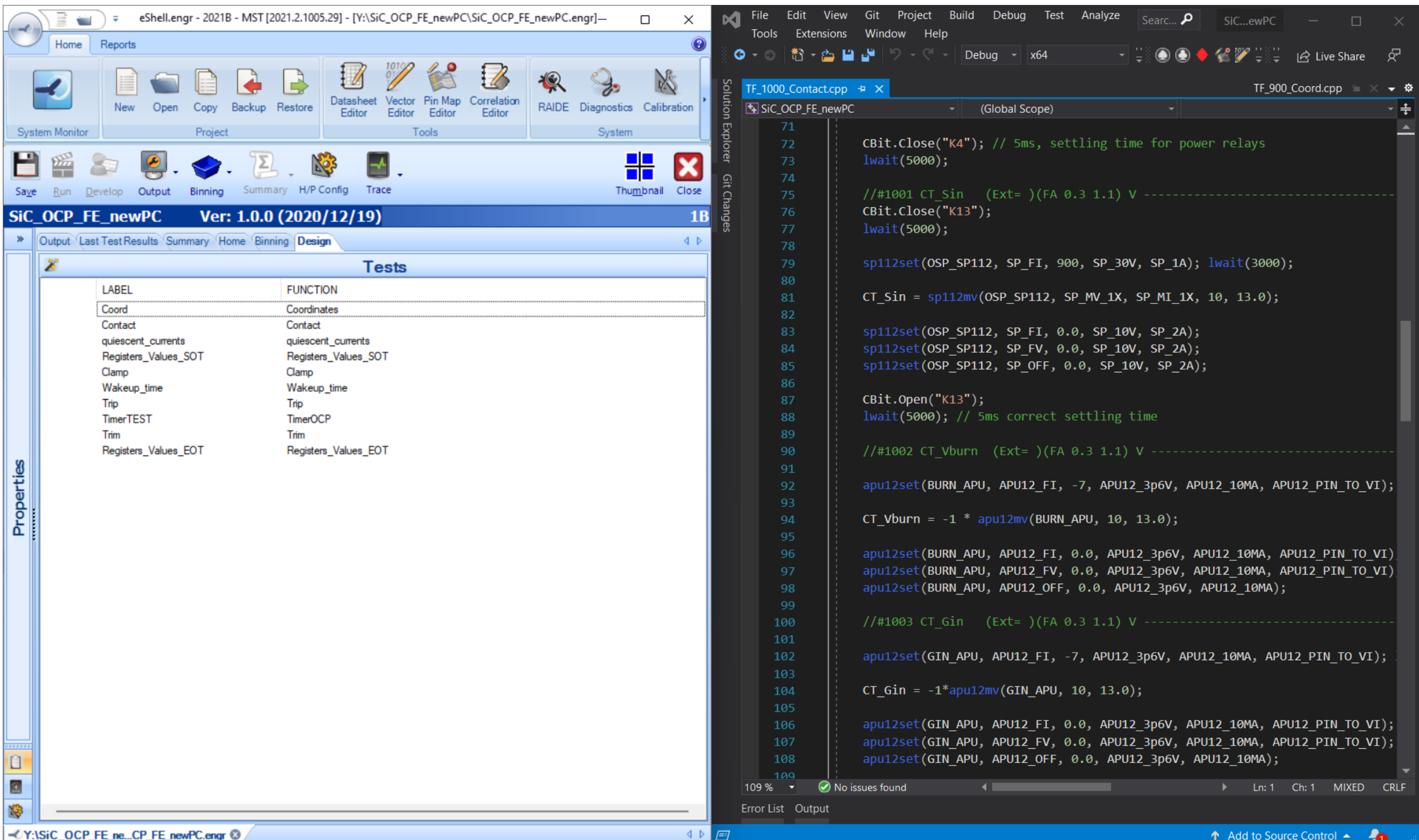



Figure 1.6: Interface for the development of an ETS-88 test program. The MST main control window is on the left, while the visual studio solution is on the right.

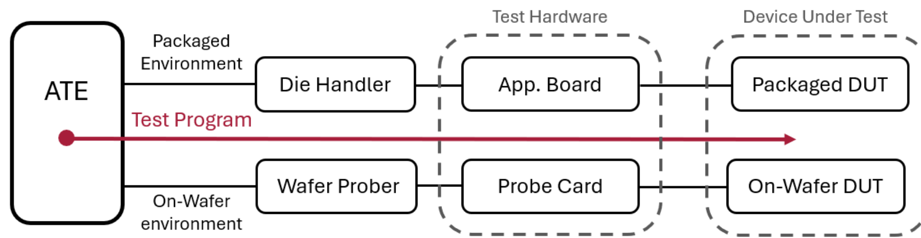



Figure 1.7: Simplified flowchart of the standard automatic test development process in industries

- **Test hardware development** - The specialised hardware required to interface the ATE with the DUTs is designed and manufactured. After the manufacturing, the test hardware (together with all other elements of the test solution) is debugged and validated in terms of functionality, accuracy, and statistical significance.

- **Data analysis** - Statistical test data are gathered on sample wafers and packaged devices to support the device characterisation and improvement, and the optimisation of all elements of the test solution.

- **Final test** - Following the validation phase, the definitive test solution is deployed in production environments for pre-market product validation. Test data are collected and archived for future analysis, and comprehensive test reports are produced.

## 1.2 Gate Drivers

Gate drivers are included in the broad category of interface devices within power electronic systems, and represent one of the most important and widespread branches of modern electronics across a wide range of applications, from industrial to automotive systems.

A gate driver can generally be defined as a power amplifier that accepts a low-current, low-voltage input signal from a microcontroller or a similar digital source, and translates it to a high-current output signal (driver) at the proper voltage level, which is capable of quickly charging or discharging the gate of the driven power device, efficiently switching between its on-state and off-state operations.

Gate drivers are generally used to drive high power transistors such as power MOSFETs (Metal-Oxide-Semiconductor Field Effect Transistors) or IGBTs (Insulated Gate Bipolar Transistors), providing fast switching times

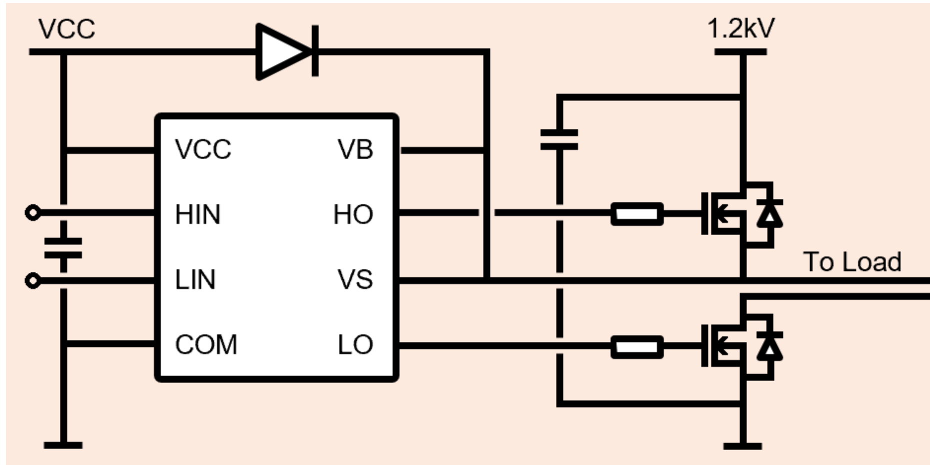



Figure 1.8: Example of a low and high-side gate driver employed in a high-voltage application

and ensuring the safety of downstream components through gate insulation and additional safety mechanisms, such as under and over-voltage protection, over-current protection, and automatic thermal shutdown[5].

Gate drivers can generally be classified according to their topology and method of operation. According to the scope of this dissertation, two main relevant distinctions are considered:

Firstly, **low-side** and **high-side** gate drivers can be distinguished based on the position of the driven switch. For low-side gate drivers, the switch is connected between the load and ground, while for high-side gate drivers, the switch is connected between the supply voltage and the load. These two architectures implement distinct driving modes, and can be employed in different applications. The switching element is typically a n-channel MOSFET, as it offers inherently lower on-resistance and superior current capability compared to p-channel devices. Gate drivers with combined low and high-side outputs can be employed together with n-channel MOSFETs to drive high-voltage inductive loads (e.g., motors), as schematically shown in Fig.1.8.

Secondly, **isolated** gate drivers provide galvanic separation between low and high-power domains through coupling mechanisms (e.g., capacitive or inductive), improving safety and noise immunity [7]. On the other hand,

[5]Protection mechanisms can be integrated into the gate driver IC, or embedded at the system level.

**non-isolated** gate drivers allow simpler architectures when direct control is sufficient.

### 1.2.1 Testing of gate drivers

The testing process for gate drivers is centred on the characterisation of their static and dynamic behaviour, with particular focus on the state switching action. Key parameters are typically tested across a wide range of electrical, ambient, and load conditions for both off-to-on and on-to-off transitions.

Industrial practices for gate driver testing are well-established, although a formal industrial standard (IEEE-P2964) is still under development [8].

The main parameters that are typically measured during gate driver testing include:

- **Switching time** - defined as the total time required for a state transition. It affects switching losses and the overall efficiency of the driven active component.
- **Peak current** - defined as the maximum output current achievable when actively driving a load. It directly affects the switching time and the gate charge and discharge capability.
- **Threshold voltage** - defined as the voltage at which the gate driver initiates a state transition. It is crucial as a characterisation parameter and to enable timing measurements.
- **Quiescent current** - defined as the consumed current in idle condition when the device is not actively driving a load. The characterisation must be done between different device pins.
- **Leakage current** - defined as the current dissipated through the device when it is actively driving a load. It directly impacts the device efficiency.

Other device-specific parameters, related to the functionalities, protection mechanisms, and applications of the tested device often require dedicated testing procedures.

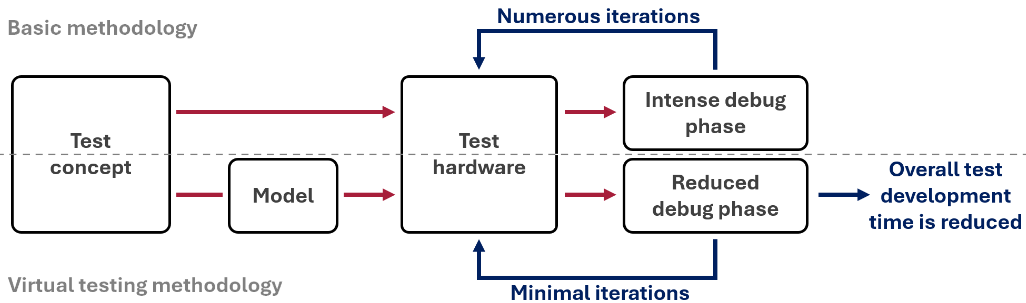



Figure 1.9: Simplified flowchart highlighting the advantage of virtual testing within the industrial workflow

## 1.3 Virtual Testing

Given the rise in circuit complexity, test hardware is becoming more complex and is therefore frequently subjected to errors and malfunctions of various nature, which are often related to the analog electrical behaviour of the test hardware when it is stimulated by the ATE.

The development of test hardware results therefore time-consuming and resource-intensive, often requiring multiple revisions, entire redesigns, and intensive physical troubleshooting. In this scenario, the adoption of an electrical virtual testing framework to support the industrial workflow by running preliminary mixed-signal simulations, allows early fault detection in the test hardware, and brings significant improvement to the overall efficiency of the test development process.

To be well-suited to accomplish this task, and properly account for the mutual interactions between all the different components of the test solution, an electrical design framework must include: models of the ATE resources which account for their limitations and analog electrical characteristics, a complete library of electrical components to allow versatile schematic design, and the ability to effectively model the DUT functionalities.

Virtual testing aims to apply techniques commonly used in electronic design to the field of automatic testing. The core idea is to operate within a development framework that allows design and validation of comprehensive test setups, reducing the reliance on physical debugging and significantly improving the overall efficiency of the process. A conceptual visualisation of the virtual testing approach is shown in Fig.1.9.

Developing a virtual testing framework requires the definition of several variables, such as the choice of the proper software and tools, modelling methodologies, and level of abstraction. Due to the broad range of possible

implementations and approaches, virtual testing remains an interesting and promising field of industrial and academic research.

### 1.3.1 State of the art

Virtual testing is a well-known concept in academic and industrial research, and attempts to introduce frameworks into the industrial workflow can be traced back to the late 1980s and early 2000s. These early efforts are focused on digital [9, 10], purely analog [11], and mixed-signal virtual testing [12].

The diversity across proposed virtual testing approaches is significant. As a result, a wide range of applications and methodologies has emerged over the years, without the consolidation of an effective industrial standard. In recent years, an extensive collection of literature works and commercially available solutions has become available. For clarity, a limited number of representative examples are referenced for each broad category.

A significant portion of the available solutions focuses on the digital debug, providing capabilities such as syntax checking and debugging of TP code, and assisted generation of digital patterns for scan-based tests [13, 14].

Mixed-signal options extend these concepts by implementing proper analog modelling. Such approaches typically rely on the availability of models for both ATE and DUT [15], adopt interface-based modelling integrated within black-box simulation tools [17, 18], or propose direct integration within the native ATE programming environment [16].

Alternative solutions propose different approaches, such as integration within the pre-silicon verification environment [19], or methodologies requiring the external simulation of test hardware [20].

In this work, a different approach is proposed and evaluated in its principle, functionality, and usefulness. The proposed SystemC-AMS virtual testing framework is completely independent of the ATE control software and focuses on a schematic-based approach, ease of setup, efficiency of simulation, and the ability to support versatile experimentation for the user. The framework allows the use of fully customisable ATE models, schematic design of test hardware, evaluation of TP sections, and simulation of the complete test setup within a user-friendly graphical design environment.

This approach is specifically targeted at addressing the limitations of existing solutions related to the efficiency of test hardware design and test hardware early validation.

## 1.4 Activity Overview

The specific focus of the presented work is on methodologies for automatic testing applied to electronic devices operating in high-power applications. For these devices, mixed-domain tests are becoming increasingly relevant given the growing number of integrated digital functionalities, but a strong emphasis remains on the purely analog behaviour.

The work presented in this dissertation aims to tackle the intrinsic inefficiency encountered during the development of versatile test hardware by exploring the concept of virtual testing, and developing a virtual testing framework based on SystemC-AMS and supported by the COSIDE graphical design environment. The mentioned working environments are chosen following proper research, considering their relevant features for the target application, their observed advantages with respect to similar alternatives in terms of modelling flexibility and simulation efficiency, and their potential for future improvement and expansion.

The developed framework is aimed at the overall improvement of the automatic test solutions development process, by coherently optimising the schematic design of test hardware and the coding of test programs, allowing the shortening of test time, and the tuning of instruction sequencing.

The proposed framework focuses on the schematic design of test hardware within a comprehensive environment that includes developed models of the ATE resources and of various electrical components, characterised by a varying level of abstraction. Sections of the test program can be optimised through mixed-domain simulations that provide insight into both the analog behaviour of the test setup, and into the mutual interaction between test hardware, test equipment, and DUT. This improves the effectiveness of the pre-physical troubleshooting of the test solution, and of the overall process.

The first part of the work involves the development from the ground-up of models for the main resources of the Teradyne ETS-88 ATE, accounting for their features and data-sheet specifications. Electrical components for the development of wafer and packaged-level test hardware are subsequently modelled and organised in comprehensive component libraries for versatile schematic design.

The developed framework is validated within different industrial test sce-

narios, and the main results are presented in this dissertation as case studies to demonstrate the validity and potential for improvement of the proposed methodology for the target application.

The first presented industrial case study is the design of the probe-card for testing silicon wafers of an innovative industrial gate driver, that features built-in over-current protection, and several digitally configurable operational parameters. In this application, the framework is employed to investigate several key aspects of the test hardware functionality, including the general robustness of the circuitry, the integrity of the analog and digital signals, and solutions for the measurement of notable electrical quantities.

An important sub-application which is effectively tackled within the framework is the investigation of solutions for generating a variable-amplitude current pulse ($t_{pulse} < 1.5\mu s$, $I_{max} > 4A$), where active electrical components are employed to modulate the native capabilities of the ATE.

The comparison between simulation results and bench measurements performed within the test program execution on the manufactured probe-card, highlights the capability of the proposed framework to provide an effective analog representation of the modelled components and to effectively support the development of automatic test solutions.

The second presented industrial case study is the development of a circuit configuration to enable the measurement of the high-level output peak current of isolated gate drivers ($t_{event} < 100ns$, $I_{peak} \approx 15A$). An application board is developed with the assistance of the framework from the conceptual design to the physical implementation. A sequence of formal qualification measurements is subsequently performed to validate the test concept and make it viable for integration into future standard test solutions.

This dissertation incorporates activities and results previously published by the authors in [21] and [22].

# Chapter 2

# Basic Methodology

As previously introduced, the proposed virtual testing framework is intended as a tool to assist in the development of automatic test solutions by allowing design and pre-silicon validation of test hardware, mainly in scenarios in which the focus is on the analog behaviour, and when the electrical capabilities of the ATE are modulated by active and passive components on the test hardware.

In this chapter, the methodology behind the development and implementation of the framework is introduced, focusing on the selected software and tools, the adopted principles for hardware modelling, and the main steps for designing and simulating a test hardware instance.

## 2.1 Software and Tools

The choice of the proper software and tools to develop and implement the models and functionalities of the framework is a fundamental preliminary step. The selected modelling language must offer the possibility to develop hardware models with the proper level of abstraction for the target application, and the development environment must be equipped with all the necessary features for versatile modelling and end-user experience.

Following a comparison with similar alternatives, the SystemC-AMS hardware modelling language, implemented within the COSIDE development environment is selected for the development of the proposed framework.

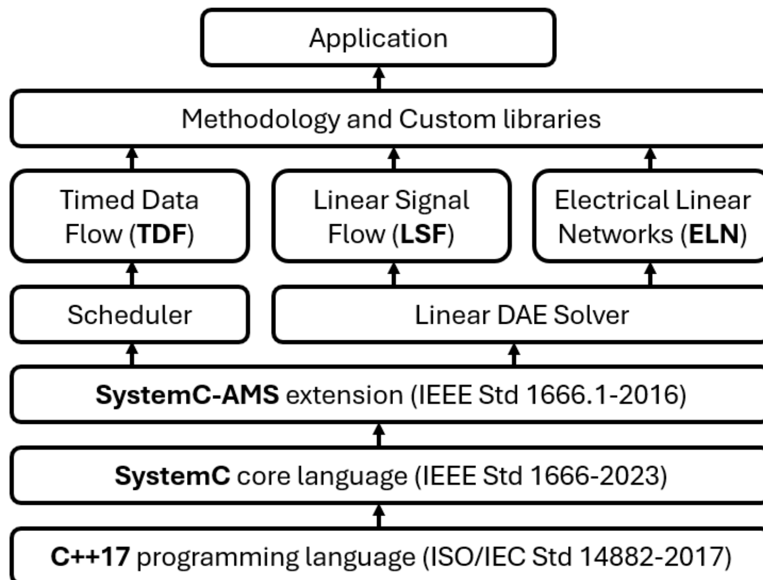



Figure 2.1: Hierarchical structure of the SystemC-AMS hardware modelling language

### 2.1.1 SystemC-AMS

SystemC-AMS is a hardware modelling language that is extension of the standard SystemC hardware description language [23], built on the C++ programming language. It offers various functionalities and modelling formalisms that allow the design of mixed-domain systems at various levels of abstraction, and the setup of efficient simulations.

For mixed-level systems, SystemC-AMS has been shown to achieve higher simulation performance compared to low-level SPICE simulators, while maintaining a sufficient level of analog accuracy for the target application, along with the versatility and abstraction capabilities of typical system-oriented Hardware Description Language (HDL)s, such as Verilog-AMS [24]. In purely analog systems, similar performance to SPICE have been observed, with improved versatility and efficiency in specific scenarios [25, 26]. Moreover, formal methods have been developed and employed to confirm the behavioural equivalence between SystemC-AMS and SPICE [27, 28].

SystemC-AMS is a relatively recent modelling language with promising potential for further development [29]. Several studies have demonstrated its effectiveness as a powerful modelling tool for both strictly electrical systems [30, 31] and heterogeneous systems [32, 33], opening the possibility for future integration within the framework of models representing physical and

mechanical aspects of automatic testing, such as temperature effects and the mechanical handling of wafers and packaged devices.

Considering these factors, SystemC-AMS is selected as the ideal modelling language for the target application.

The native SystemC-AMS modelling environment, whose hierarchical structure is shown in Fig.2.1, offers the following modelling formalisms, which contribute to a versatile and efficient implementation of the required models:

- **Electrical Linear Network (ELN)** - used for modelling continuous-time conservative electrical behaviour. Kirchhoff laws are solved at each node and port of the circuit, to compute the correct voltage and current values at each time step. The library of built-in ELN components includes passive electrical elements and different types of voltage and current sources, which can be combined to model more complex electrical functionality.

- **Linear Signal Flow (LSF)** - used for modelling continuous-time non-conservative behaviour by allowing the implementation of user-defined differential equations based on two key quantities, referred to as *flux* and *potential*. LSF networks are useful to efficiently implement linearised electrical behaviours and signal processing functionalities, other than physical and mechanical models.

- **Timed Data Flow (TDF)** - used for modelling discrete-time non-conservative behaviour. The definition of custom equations and modules is allowed, and the module activation and signal sampling scheduling is statically calculated at the beginning of the program execution, to achieve an important increase in simulation speed. TDF networks are useful to implement complex functionalities while maximising the simulation performance.

Models developed in SystemC-AMS present intrinsic modularity and allow a high degree of versatility and expandability. In the specific application of the presented work, the developed models of ATE, DUT, electrical components, and laboratory equipment, can be updated with increasing complexity according to the fidelity required by the specific application, without altering the underlying model structure.

Built-in SystemC-AMS features, such as test-benches, stimuli files and various fault injection methods, allow in-depth examination of the system behaviour in ideal and non-ideal conditions, and subsequent fine tuning of the developed hardware solutions [34].

### 2.1.2 COSIDE modelling environment

The proposed SystemC-AMS framework is fully implemented and intended to run on the COSIDE software [35], a comprehensive mixed-signal design and simulation environment, recognised as the first commercially available development environment entirely based on SystemC and SystemC-AMS.

COSIDE supports the modelling of complex systems that integrate analog and digital functionalities at various levels of abstraction, and natively includes extensive built-in libraries, a user-friendly schematic editor, efficient hierarchy management, and a capable waveform analyser. Various literature works have demonstrated the versatility and user-friendly nature of COSIDE for SystemC-AMS development [36, 37].

By employing the features offered by the development environment, the versatility and efficiency of the framework in assisting test hardware development are further improved. For completeness, a snapshot of the COSIDE interface is shown in Fig.2.2.

## 2.2 Models Basic Structure

To achieve optimal versatility within the modelling environment, the modelled components are organised into distinct libraries, which can be referenced and used for the schematic design of test hardware instances. Dependency relationships between libraries are automatically managed by the library manager, and sub-libraries are automatically imported when the corresponding master libraries are referenced. A conceptual scheme of the implemented library structure is shown in Fig.2.3. Three main library categories are included into the first iteration of the framework:

The **ATE library** includes all the models of the specific ATE chosen for the test solution. At the top-level, each library symbol represents a specific resource of the ATE. This guarantees increased flexibility in the choice of the specific ATE configuration, and a reduction of the simulation time by arbitrary exclusion of all the non-necessary resources from the schematic.

The internal hierarchy of the modelled resources allows extension of this concept to the single channels, which can be singularly imported for particularly small designs to maximise simulation speed. It is important to notice the applicability of this principle is subjected to the structure of the specific resource, and the full top-level structure may be necessarily used[1].

[1] As shown in Chapter 4, an example for the Teradyne ETS-88 is the APU-12.

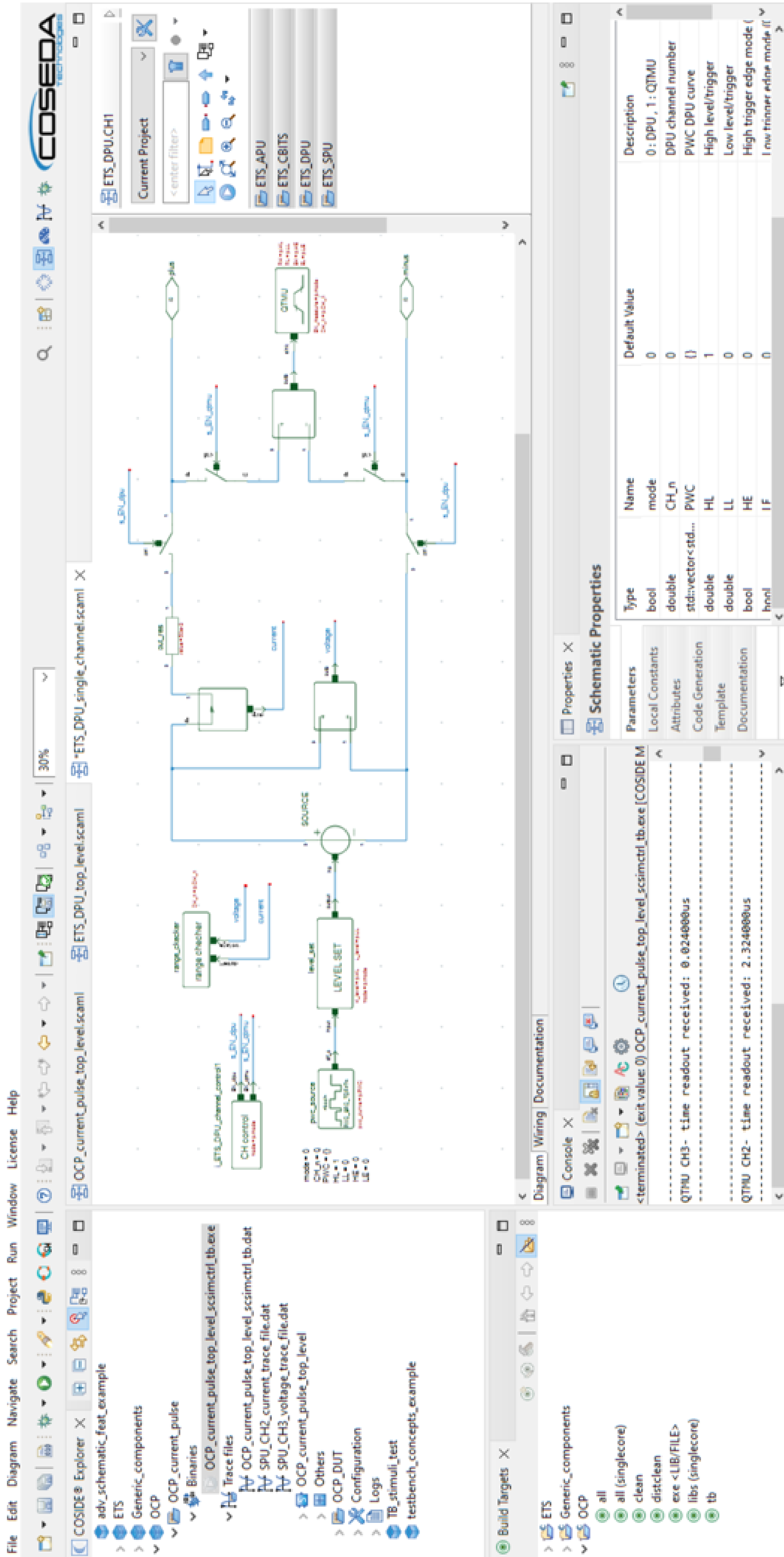


Figure 2.2: Graphical user interface of the COSIDE modelling environment

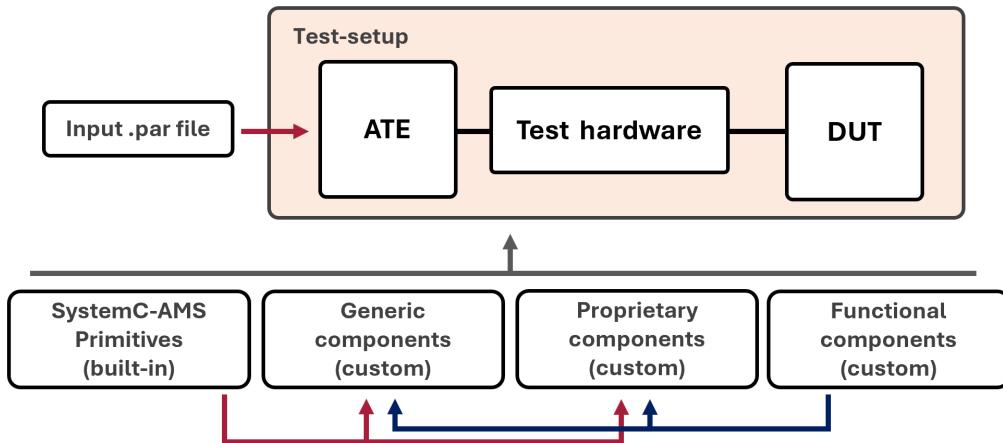



Figure 2.3: Conceptual chart of the framework SystemC-AMS hierarchical library structure.

As shown in the next section, to efficiently assign parameters and control the top-level symbols of the ATE resources, a text-file based approach is chosen.

The **component libraries** contain the components necessary to model electrical systems and design test hardware instances. Other than purely electrical components, non-electrical elements that implement numerical models of complex effects and functionalities (slew rate, signal sampling, etc.) are also available. These libraries can be referenced individually according to the requirements of the specific model or test hardware under development, and are often subjected to dependency relationships with other libraries.

The **test hardware library** is the main working library, which includes the schematic of the test hardware under development and, the models of the DUT (if required). Components from other libraries are easily imported and connected as top-level symbols.

### 2.2.1 Input parameter file

The behaviour and characteristic values of the components used in test hardware schematics is specified by top-level parameters, assigned by the user before the start of the simulation.

Top-level parameters for standard schematic components (e.g., passive components, diodes, and transistors) are assigned directly within the schematic,

```
//  --- SPU-112 -------------------------------------------------

    // CHANNEL ENABLING LIST
    // CHANNEL MODE LIST
    // 0:FI, 1:MI, 2:FV, 3:MV
    // format: array{bool<4>}
    // default: {0,0,0,0}
    SPU112_CH_EN = {0,0,0,0}
    SPU112_CH_MODE = {0,0,0,0}
    // CHANNEL VOLTAGE RANGE LIST
    // format: array{double<4>[V]}
    // default: {30,30,30,30}
    SPU112_CH_VRANGES = {30,30,30,30}
    // CHANNEL CURRENT RANGE LIST
    // format: array{double<4>[mA]}
    // default: {1,1,1,1}
    SPU112_CH_IRANGES = {1,1,1,1}
    // PWL CURRENT FORCING CURVE (4)
    // format: array{pair{double[us],double[A]}}
    // default: {{0.0,0.0}}
    SPU112_CURR_PWL_1 = {{0.0,0.0}}
    SPU112_CURR_PWL_2 = {{0.0,0.0}}
    SPU112_CURR_PWL_3 = {{0.0,0.0}}
    SPU112_CURR_PWL_4 = {{0.0,0.0}}
    // PWL VOLTAGE FORCING CURVE (4)
    // format: array{pair{double[us],double[V]}}
    // default: {{0.0,0.0}}
    SPU112_VOLT_PWL_1 = {{0.0,0.0}}
    SPU112_VOLT_PWL_2 = {{0.0,0.0}}
    SPU112_VOLT_PWL_3 = {{0.0,0.0}}
    SPU112_VOLT_PWL_4 = {{0.0,0.0}}
    // CHANNEL TRACE START LIST
    // format: array{double<4>[us]}
    // default: {0.0,0.0,0.0,0.0}
    SPU112_TR_START = {0.0,0.0,0.0,0.0}
    // CHANNEL TRACE STOP LIST
    // format: array{double<4>[us]}
    // default: {1000,1000,1000,1000}
    SPU112_TR_STOP = {1000,1000,1000,1000}
```

Channels setup

I/V ranges configuration

I/V sourcing control

Tracing control

Figure 2.4: Section of the input parameter file to control the SPU-112 operation in a generic test-function. Controls for general operation, signal sourcing, and tracing are available.

while the parameters associated with ATE control are assigned to the top-level symbols of the ATE resources by compiling an input parameter file, whose basic structure for a specific ETS-88 resource is shown in Fig.2.4.

Different TFs can be represented by different parameter files and executed independently, enabling versatile debugging and rapid switching between different test setup configurations.

## 2.3 Framework Usage

This section describes the configuration of SystemC-AMS and of the COSIDE environment required to operate within the proposed framework and to instantiate a new test hardware configuration. The main purposes of this section are to support the reproducibility of the presented results, and to facilitate extension of the methodology to additional test scenarios. The main steps to configure the development environment are summarised as follows:

- A new project is defined within the COSIDE environment. Within this project, separate libraries are allocated to the test hardware schematic and to the models of the DUT.
  The library configuration includes a blank parameter file, representing a blank TF, which can be subsequently modified according to the target test configuration.

- The required framework libraries are referenced within the project configuration files and in the schematic editor to enable interaction and graphical integration of external components. These libraries include the ATE models and the needed electrical components.
  If required, built-in libraries for added analog fidelity[2] are referenced within the configuration files.

- Instances of the required ATE resources and the DUT model are instantiated within the test hardware schematic. A dedicated module is included to map the parameters defined in the TF parameter file to the top-level schematics of the ATE resources.

- The test hardware schematic is designed within the newly created schematic view, by defining all the necessary connections with the ATE and the DUT. During this process, rapid mixed-signal simulations can be executed to support debug and validation of the configuration.

[2]The built-in NgSpice and PWL libraries are introduced in Chapter 3.

# Chapter 3

# Components Modelling

As previously introduced, basic libraries of components to enable the implementation of all the necessary models within the framework must be implemented. SystemC-AMS natively supports a variety of standard electrical and non-electrical components, that can be used as building blocks to implement more complex models and functionalities.

The built-in components which result particularly useful for the target application are listed below:

- **ELN components** - standard electrical components that can be connected together to form an ELN network, prioritising analog fidelity with a compromise on simulation speed. ELN components include resistors, capacitors, inductors, voltage and current sources, simple measurement instruments, etc.
- **Piece Wise Linear (PWL) components** - part of the PWL library. They offer linearised models of non-linear and complex electrical components (diodes, transistors, etc.), balancing between simulation performance and analog accuracy.
- **SPICE components** - part of the NgSpice library. They can be imported and linked to SystemC-AMS schematics to significantly increase the level of analog fidelity at the expense of simulation speed.

In this chapter, an overview of the implemented component libraries for schematic design and model implementation is given. The model of a PN diode is presented as an example of a non-linear model built from the ground-up. Finally, an overview of the different available abstraction levels for equivalent components is given.

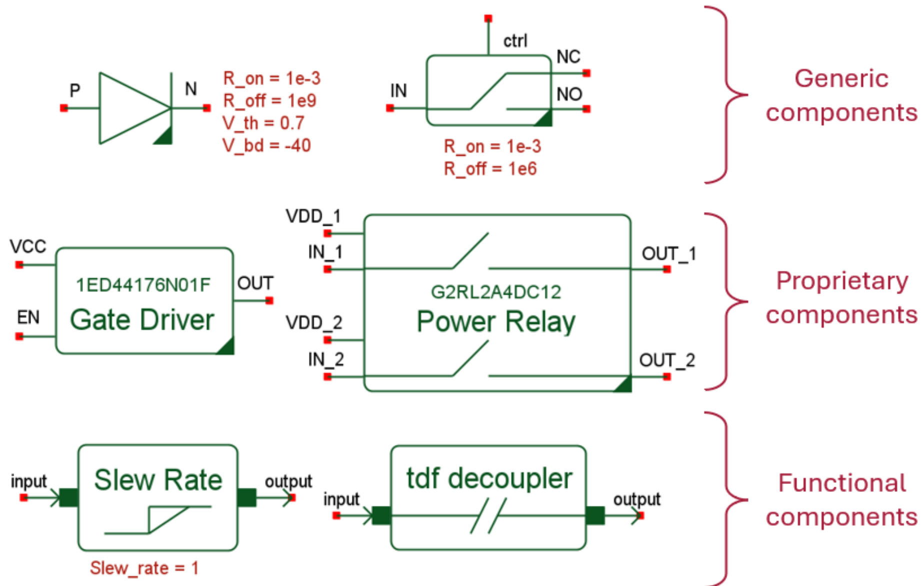



Figure 3.1: Sample collection of modelled schematic components. Two components are shown for each implemented type.

## 3.1 Schematic Components

The built-in SystemC-AMS components are limited in both complexity and selection, especially considering the scope of the target application. In the presented work, three types of components, distinguished by the nature of their functionality, have been modelled and added to comprehensive libraries:

**Generic components** include behavioural, general-purpose models of widely used electrical components, such as various types of diodes and relays. Given the general-purpose nature of these components, they are customisable in their operation and specifications by editing the top-level parameters. These components are useful during schematic design to identify the target component specifications for a specific application.

**Proprietary components** consist of behavioural models of commercially available electrical components. These models follow data-sheet specifications and include passive elements with fixed values within the internal hierarchies. Given the black-box nature of these components, they are not externally customisable.
These components are useful during schematic design to compare similar

available options of a set component and finalise the design.

**Functional components** are non-electrical, and are used to standardise the implementation of complex signal-related features, such as slew-rates, signal digitisation, TDF cluster decoupling, and placeholder connections.

As an example, a collection of modelled components is shown in Fig.3.1.

### 3.1.1 Non-linear modelling

Non-linear component models are natively available within the PWL and NgSpice libraries. To better suit the application, additional models of non-linear components, that prioritise simulation speed, can be purposely implemented by employing specific features of TDF networks. As an example, the functionality of a modelled PN diode is presented.

The diode schematic is shown in Fig.3.2. The voltage difference across the terminals ($\Delta V$) is measured, converted to a TDF signal, and provided as input to the TDF network. The *diode control* module regulates the value of the variable resistance ($R_D$) and performs the subtraction of the threshold voltage ($V_{TH}$), distinguishing between the ON and OFF states. The reverse breakdown phenomenon is also modelled through the breakdown resistance value ($R_{BD}$). To summarise:

$$\begin{array}{ll} \Delta V < V_{BD} & \rightarrow \; R_D = R_{BD} \\ V_{BD} \leq \Delta V \leq V_{TH} & \rightarrow \; R_D = R_{OFF} \\ \Delta V > V_{TH} & \rightarrow \; R_D = R_{ON} \end{array}$$

As visible in the schematic, the diode functionality is based on the existence of an internal feedback loop. $\Delta V$ sets $R_D$, but $R_D$ has a direct influence on the value of $\Delta V$. This kind of circular dependency within TDF networks cannot be analytically resolved by the SystemC-AMS kernel solver, and a delay of at least one time-step must be applied to the update of the TDF signal which sets the value of $R_D$.

Depending on the current size of the time-step, this can lead to instabilities and contradictory states. If, for example, at a specific instant $t_1$ the diode encounters a off-to-on state transition, the resistance value will not be updated until the next time-step is calculated.

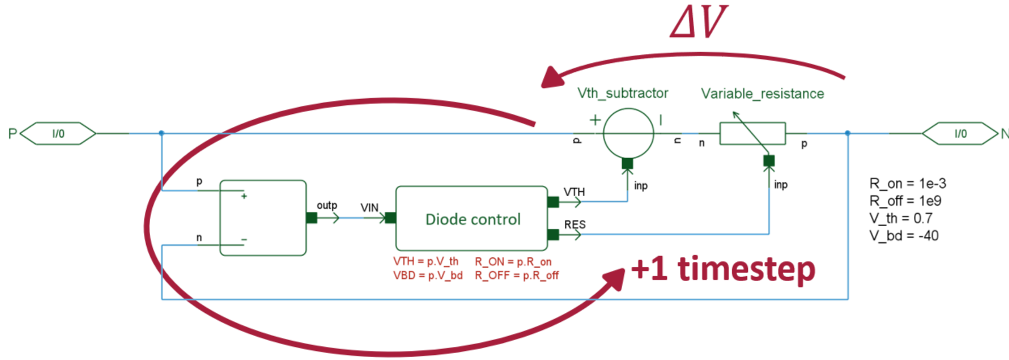



Figure 3.2: Top-level schematic of the TDF based behavioural model of a PN diode. TDF modelling constructs are highlighted.

Considering subsequent time-steps, the following situation is configured:

$$t_0 : \Delta V < V_{TH}\ ,\ R_D = R_{OFF}$$
$$t_1 : \Delta V > V_{TH}\ ,\ R_D = R_{OFF}$$
$$t_2 : \Delta V > V_{TH}\ ,\ R_D = R_{ON}$$

The time step $t_1$ is contradictory, and depending on the behaviour of the networks surrounding the diode, can generate convergence errors. Within the SystemC-AMS environment, this can be solved by implementing a time-step recalculation instruction, which is executed every time a state transition condition is detected. The new system evolution is given by:

$$t_0 : \Delta V < V_{TH}\ ,\ R_D = R_{OFF}$$
$$t_1 : \Delta V > V_{TH}\ ,\ R_D = R_{OFF}$$
$$t_1* : \Delta V > V_{TH}\ ,\ R_D = R_{ON}$$
$$t_2 : \Delta V > V_{TH}\ ,\ R_D = R_{ON}$$

At the expenses of a single time-step, which increases the total simulation time by a negligible amount, feedback-generated convergence errors can be avoided as shown.

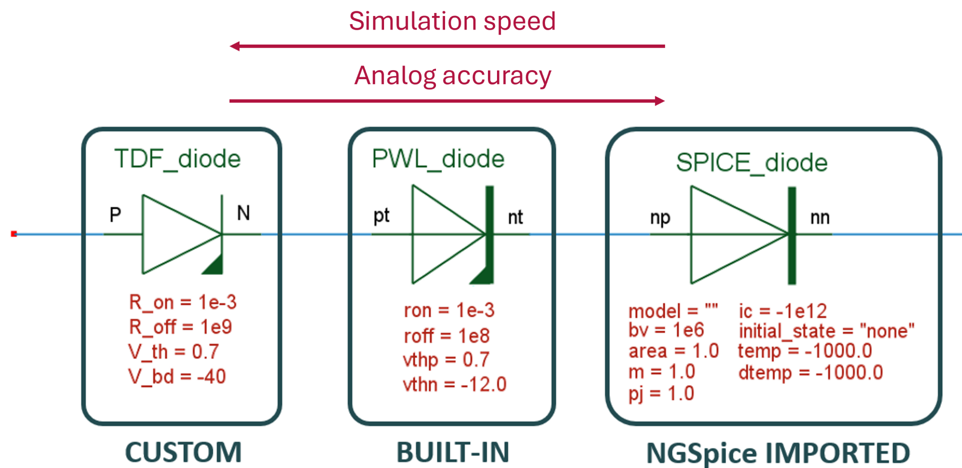



Figure 3.3: Example of the multiple abstraction levels available for a PN diode, ordered for increasing analog accuracy (and decreasing simulation speed)

## 3.2 Abstraction Levels

As previously introduced, NgSpice components can be imported into SystemC-AMS schematics by enabling the necessary settings. If a simulation is started on a schematic that includes a NgSpice component, the built-in NgSpice simulator solves the ELN network associated with the NgSpice component, prioritising analog accuracy at the expense of simulations speed.

Considering the built-in, and newly implemented models, a wide range of different abstraction levels is available for different specific components, as shown as an example in Fig.3.3 for the PN diode.

Electrical components with different abstraction levels can coexist within the same schematic, and can be selected according to the level of analog fidelity required for the specific circuit section[1].

[1]This concept is applied for the test concept shown in Chapter 6.

# Chapter 4

# ATE Models Development

A fundamental step in the implementation of the framework is the ground-up development of models for the selected ATE resources, accounting for their specifications and main features [38]. Following the previously introduced modelling principles, the primary requirement for the developed models is to be ready to be imported and connected within a test hardware schematic, and to be controllable by the user from the top-level, without the need to access or modify their internal hierarchy.

It is important to notice the developed methodology can be applied to several models of ATE, given the typically modular nature of ATE resources.

As previously mentioned, the chosen ATE is the Teradyne ETS-88, one of the most widely used instruments for the automatic mixed-signal testing of semiconductor devices involved in power management. It is characterised by a variety of different application-specific resources packaged within a highly customisable and configurable environment, with a strong emphasis on cost-efficiency, test parallelisation, and ease of control through a powerful, unified control software.

The selection of the ATE resources to be modelled follows the specific configuration of the ETS-88 chosen for the development of the test solutions presented as case studies[1]. This configuration is highly versatile and allows comprehensive testing of a wide range of semiconductor devices.

In this chapter, the models of the selected ETS-88 resources are presented, highlighting their key features and functionalities.

[1] Case studies are presented in Chapters 5 and 6.

## 4.1 Top-Level

In Fig.4.1, the ready-to-connect symbols of the modelled resources are shown. The modelled resources include: Smart Pin Unit 112 (SPU-112), Analog Pin Unit 12 (APU-12), Digital Pin Unit 16 (DPU-16) with integrated Quad Time Measurement Unit (QTMU), and Control Bits (CBITS).

The ETS-88 provides power supply units with various standard voltage levels ($\pm 5V, \pm 12V, 15V$, etc.). These power supplies are used to power components on test hardware, and are modelled within the framework as current-limited ideal voltage sources. No dedicated symbol is therefore implemented for their representation.

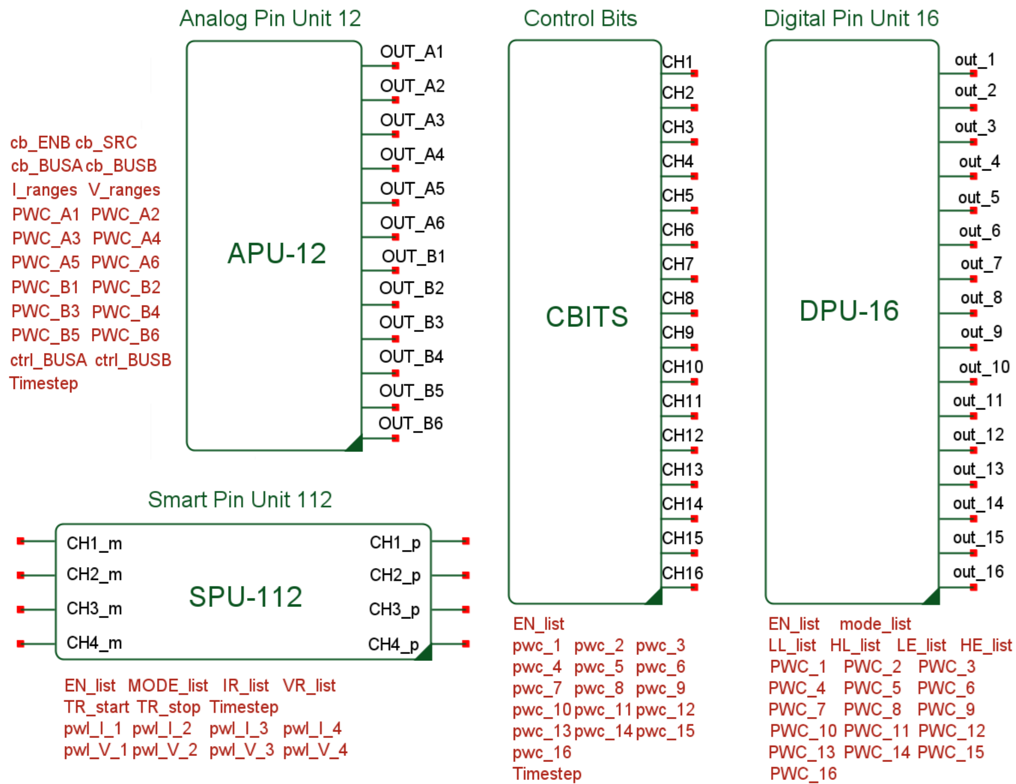



Figure 4.1: Top-level view of the models of the selected Teradyne ETS-88 resources. For clarity, the parameters associated to each resource are listed omitting their default values.

## 4.2 Smart Pin Unit 112

The SPU-112 is a quad-stackable-channel high-power voltage and current source, featuring selectable voltage and current ranges, ability to operate in Arbitrary Waveform Generator (AWG) mode, and capability to capture input signals with a 500$kSPS$ signal digitiser.

The top-level schematic of the SPU-112 model is shown in Fig.4.2. The schematic of the SPU-112 single channel model, shown in Fig.4.3, includes the following modules and components:

- Ideal PWL sources generate the TDF signals for voltage and current according to the resource specifications. Random offset and noise are added to account for the source non-ideality. The resulting TDF signal

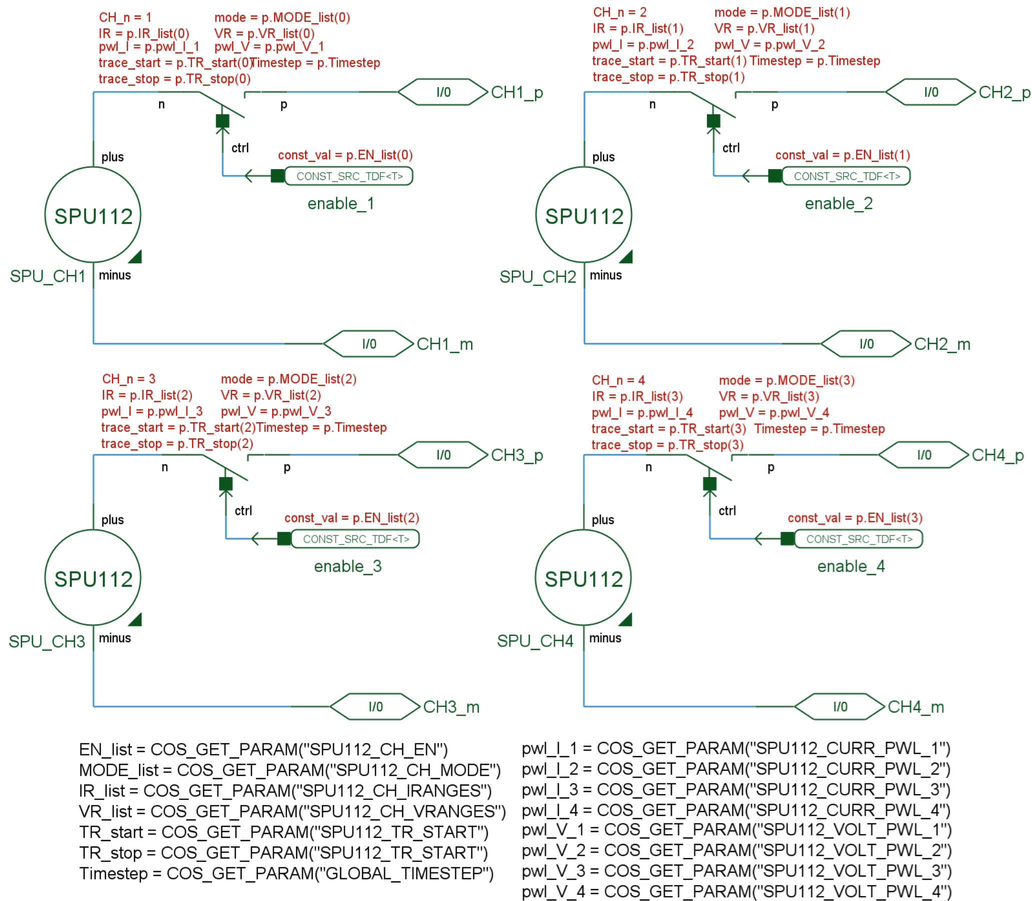



Figure 4.2: Smart Pin Unit 112 model top-level schematic view

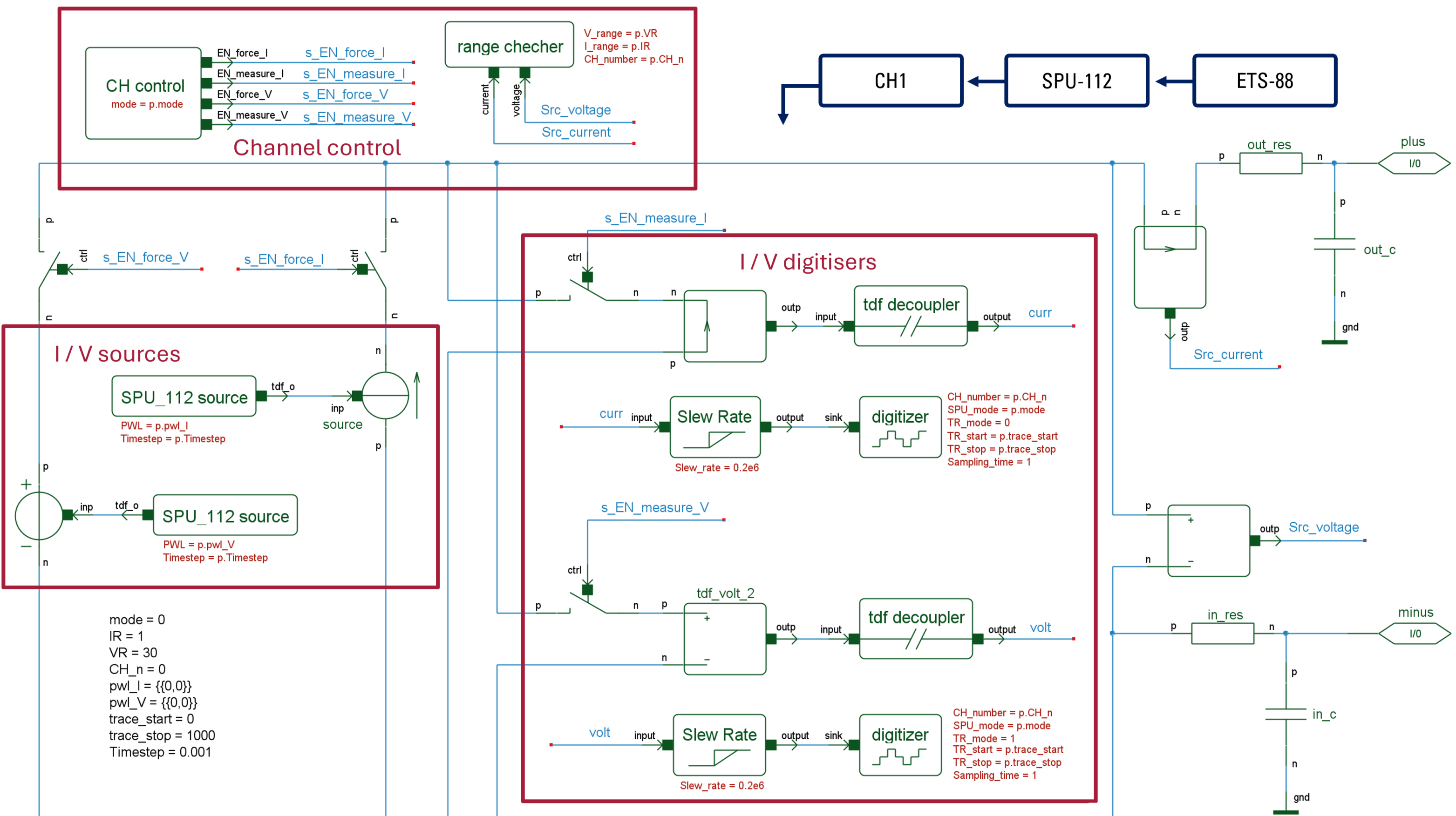



Figure 4.3: Smart Pin Unit 112 model single channel schematic view. Different circuit blocks and simplified ETS-88 hierarchy are highlighted.

is sampled by a TDF-to-ELN converter module and provided as input to the main ELN network.

- Signal tracers for voltage and current are available for each channel. Input and output signals passing through the SPU-112 main ELN network are converted back to TDF signals, and a dedicated *TDF decoupler* module samples them at the proper sampling rate ($500kSPS$).
A *slew-rate* module accounts for the maximum slew-rate of the digitiser (measured within a targeted experiment at approximately $0.2V/\mu s$).
The resulting TDF signals are saved into a trace file, which can be produced on-demand with selectable start and stop times.

- The *range_checker* module monitors the values of input and output voltage and current passing through the SPU-112 main ELN network and produces specific warning messages to the user terminal, halting the running simulation, if the resource capabilities are exceeded for the selected settings.

- The *channel_control* module configures the various operational modes of each SPU-112 channel (force current, force voltage, measure), by opening and closing ideal switches along the main ELN network.

- Specific passive components account for the SPU-112 input and output resistance and capacitance.

## 4.3 Analog Pin Unit 12

The APU-12 is an intermediate power 12-channel analog control unit, capable of voltage and current sourcing with different selectable ranges, signal digitalisation, AWG capability, and fully controllable multiplexing between the different channels.

The single channel and top-level schematics of the APU-12 model are shown in Fig.4.4 and Fig.4.5. The main modelled features include:

- The multiplexing between the different channels is controlled by ideal switches that enable or disable signal paths. The configuration is performed by the *mux_control* module, that translates the top-level parameters selected by the user into a resource configuration.
- Within the single channel, a dedicated *APU_source* module generates voltage curves according to the source specifications in terms of rise and fall times, and maximum current ratings.
- The *range_checker* module monitors the signal values along the ELN network and produces warning messages if the capabilities are exceeded.
- Specific passive components account for the APU-12 output resistance and capacitance.

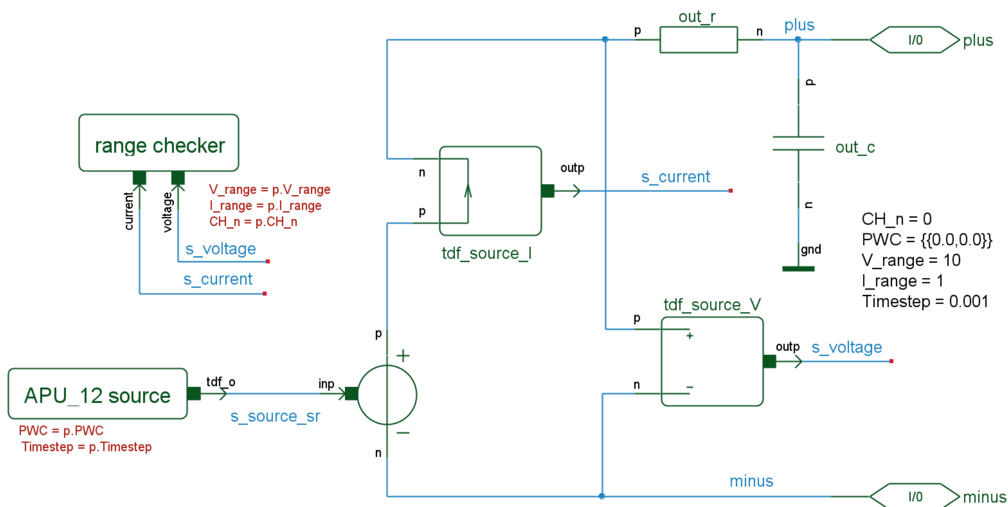



Figure 4.4: Analog Pin Unit 12 model single channel schematic view

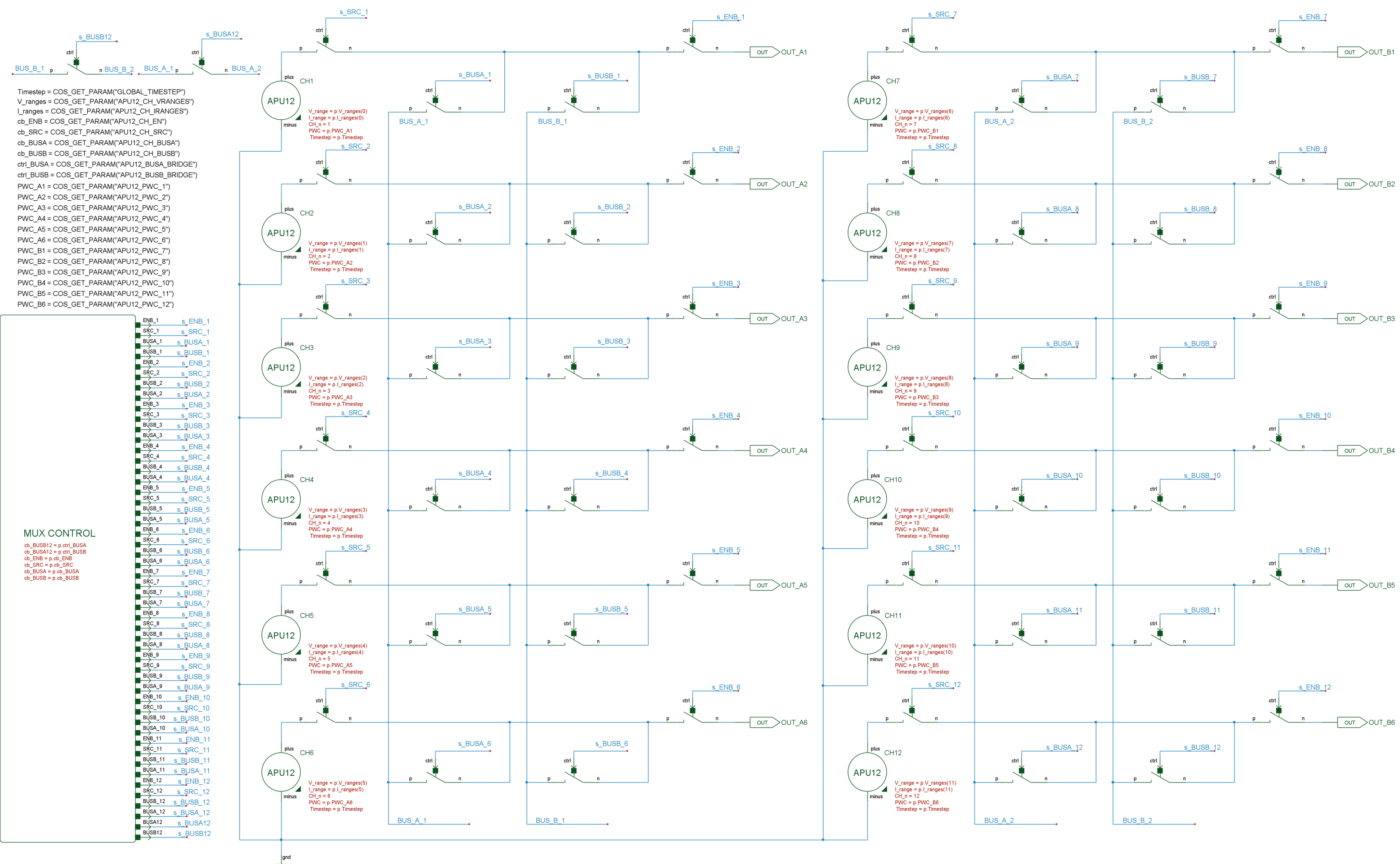



Figure 4.5: Analog Pin Unit 12 model top-level schematic view

## 4.4 Digital Pin Unit 16

The DPU-16 is a 16-channel high speed I/O digital control unit with selectable high and low voltage levels, and an integrated Quad-Time-Measurement-Unit (QTMU).

The single channel and top-level schematics of the DPU-16 model are shown in Fig.4.6 and Fig.4.7. The main modelled features include:

- Within the single channel, a dedicated *DPU_source* module generates digital signals according to the source specifications in terms of rise and fall times, and maximum current ratings.
- The *range_checker* module monitors the signal values along the ELN network and outputs warning messages if the capabilities are exceeded.
- Specific passive components account for the DPU-16 output resistance.

The QTMU is a time measurement resource capable of detecting crossing events of trigger voltages, with selectable levels and edge types. The QTMU is natively multiplexed to the DPU-16 channels and, in the presented model, is directly integrated into the DPU-16.

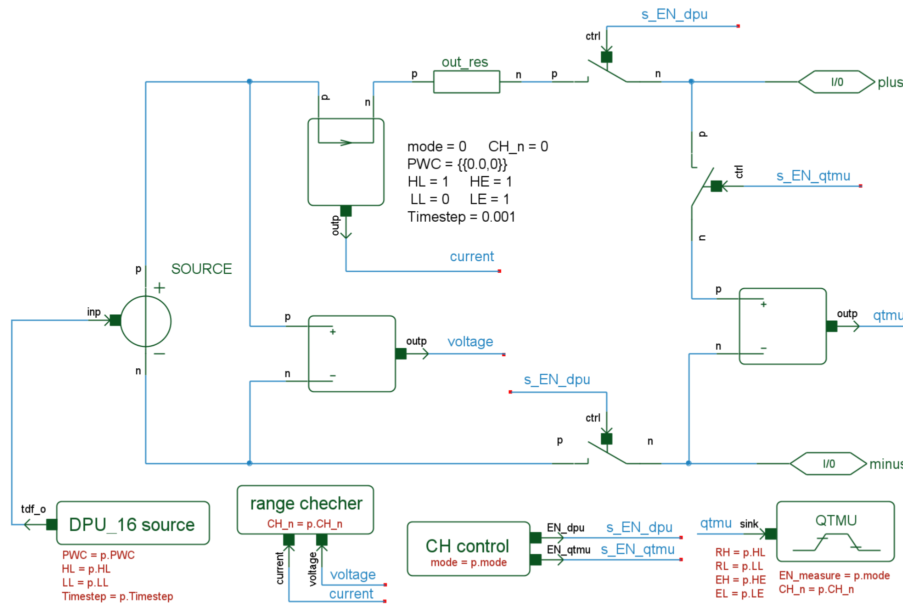



Figure 4.6: Digital Pin Unit 16 model single channel schematic view

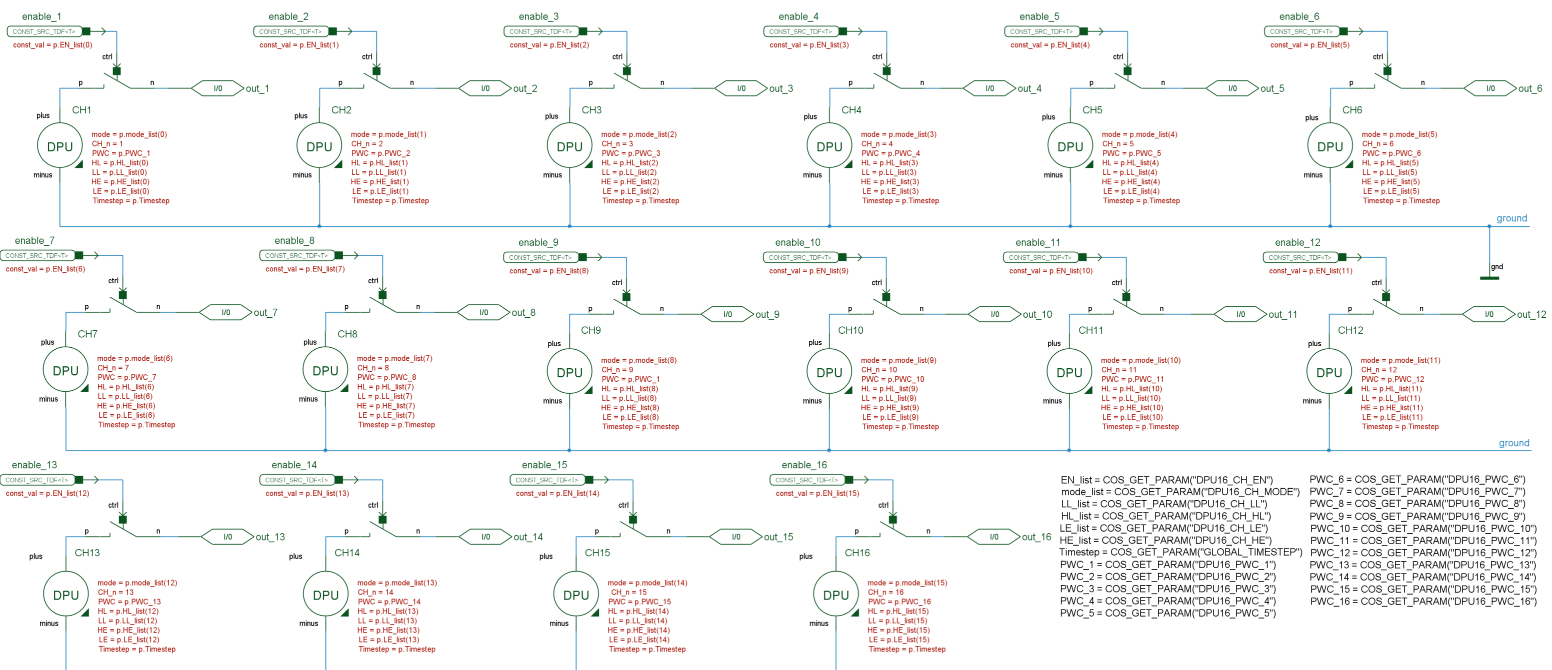



Figure 4.7: Digital Pin Unit 16 model top-level schematic view

## 4.5 Control Bits

The CBITS are a bank of 16 low-side switches dedicated to relay control on test hardware, capable of driving coils with operating voltages up to $24V$.

The single channel and top-level schematics of the CBITS models are shown in Fig.4.8 and Fig.4.9. The main modelled features include:

- The maximum voltage and current ratings are enforced by a *limit_checker* module, that produces specific warning messages if the limits are exceeded. This limits the number of relays controllable by a single CBIT.
- Specific active and passive components account for the CBITS impedance characteristics.

Being low-side switches, CBITS work in complementary logic and must be connected to the negative terminal of the relay coil to properly control it. The positive coil terminal must be powered through a built-in power supply.

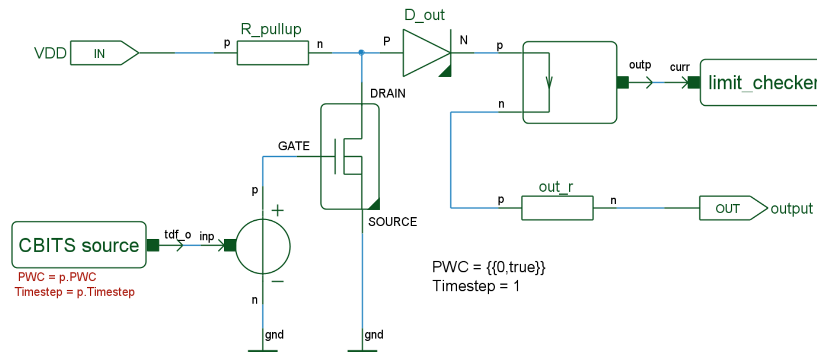



Figure 4.8: Control Bits model single channel schematic view

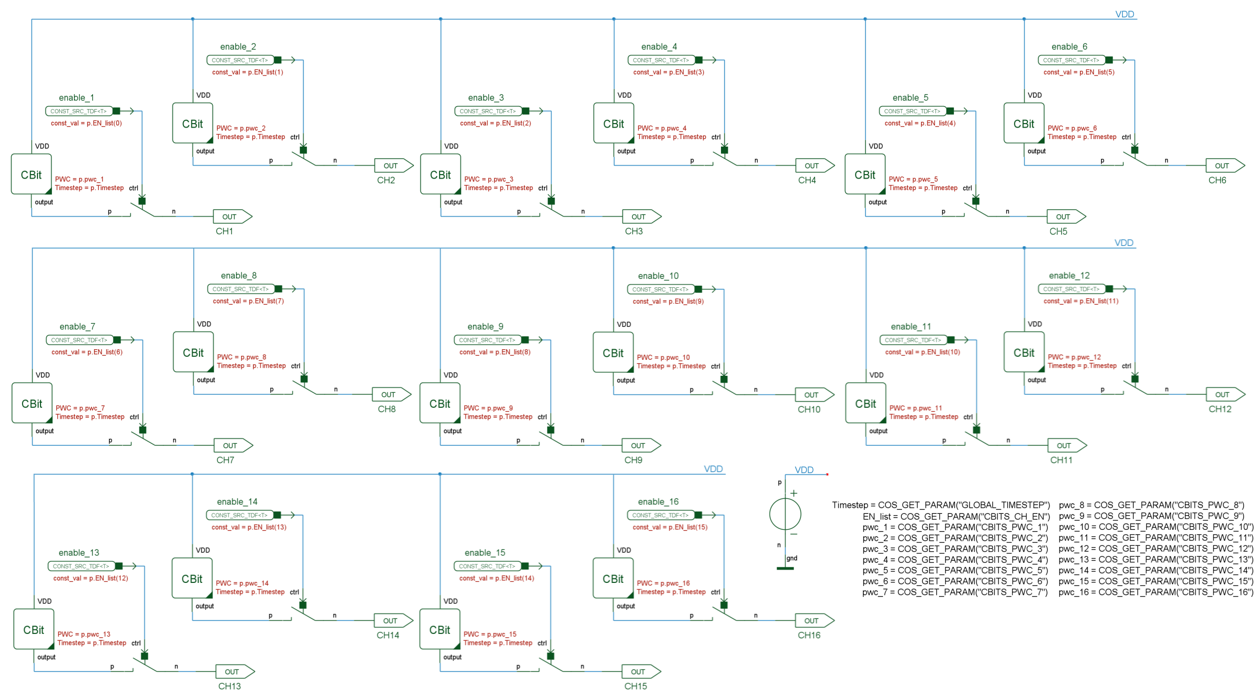



Figure 4.9: Control Bits model top-level schematic view

# Chapter 5

# Application - Gate Driver Test Solution Development

The first main application of the framework to an industrial test scenario is the development of the full test solution for an industrial innovation device.

The device to be tested is a gate driver for silicon carbide (SiC) transistors equipped with a built-in over-current protection mechanism based on an IC with current readout capability, which can switch off the transistor and trigger the clamping of the output voltage. Configuration and trimming of several operational parameters and DFT functionalities is available through I2C communication with the device.

In the scope of the presented work, the development and validation of the test solution serves as a powerful case study, to validate and improve all the main functionalities of the framework after the development of the first operational iteration.

In the following sections, after a general introduction and an overview of the designed probe-card, solutions for the measurements of the main operational parameters of the gate driver are presented. A brief overview of the statistical analysis of the produced data are presented as conclusion.

## 5.1 General Aspects

The test solution development, which has been followed from the beginning to the closure, is comprehensive of the following steps:

- The test-plan is developed according to the standard industrial practices and all the key parameters and functionalities to be tested are

defined, including their lower and upper limits, and allowed measurement tolerances. In this phase, the first blueprint of the TP is coded and debugged offline[1].

- The first iteration of the application board for the test of packaged sample devices is designed. After the physical manufacturing, the test board is debugged together with the TP and the associated DUTs.
- The wafer probe-card is designed. After the physical manufacturing, the probe-card is interfaced with the wafer prober through a driver software and debugged together with the first sample wafers. The TP is finalised for the automatic wafer testing phase.
- Data are collected on the automatically tested wafers, and statistical analysis is performed as the final product of automatic test development. The produced analysed data are employed for further debug and improvement of the device, and to prepare the final production test.

### 5.1.1 Probe-card design

The principal applications of the framework within the development of the test solution are the design and validation of the test hardware and the tuning of specific sections of the TP. Given the redundant nature of the test hardware for package-level and wafer-level testing, the probe-card is used as a test hardware representative example in the following sections.

The schematic of the designed probe-card is shown in Fig.5.1. A purposely developed behavioural model of the DUT describes its interaction with the test setup (ATE and test hardware) by numerical models and electrical components included in its hierarchy.

The general robustness of the test hardware, in terms of correctness of the implemented connections and avoidance of accidental short circuits, is often object of oversight and can lead to lengthy debug sessions and hardware damage. As previously introduced, the single top-level schematic approach allows efficient preliminary inspection of the test setup.

The possibility to inject elementary signals from the different channels of the ATE resources, in conjunction with the warning terminal messages implemented for the various components, allows easy detection and fix of improper connections.

[1]During this phase the TP debug is done from the point of view of the code syntax and structure, detached from the electrical behaviour of the ATE.

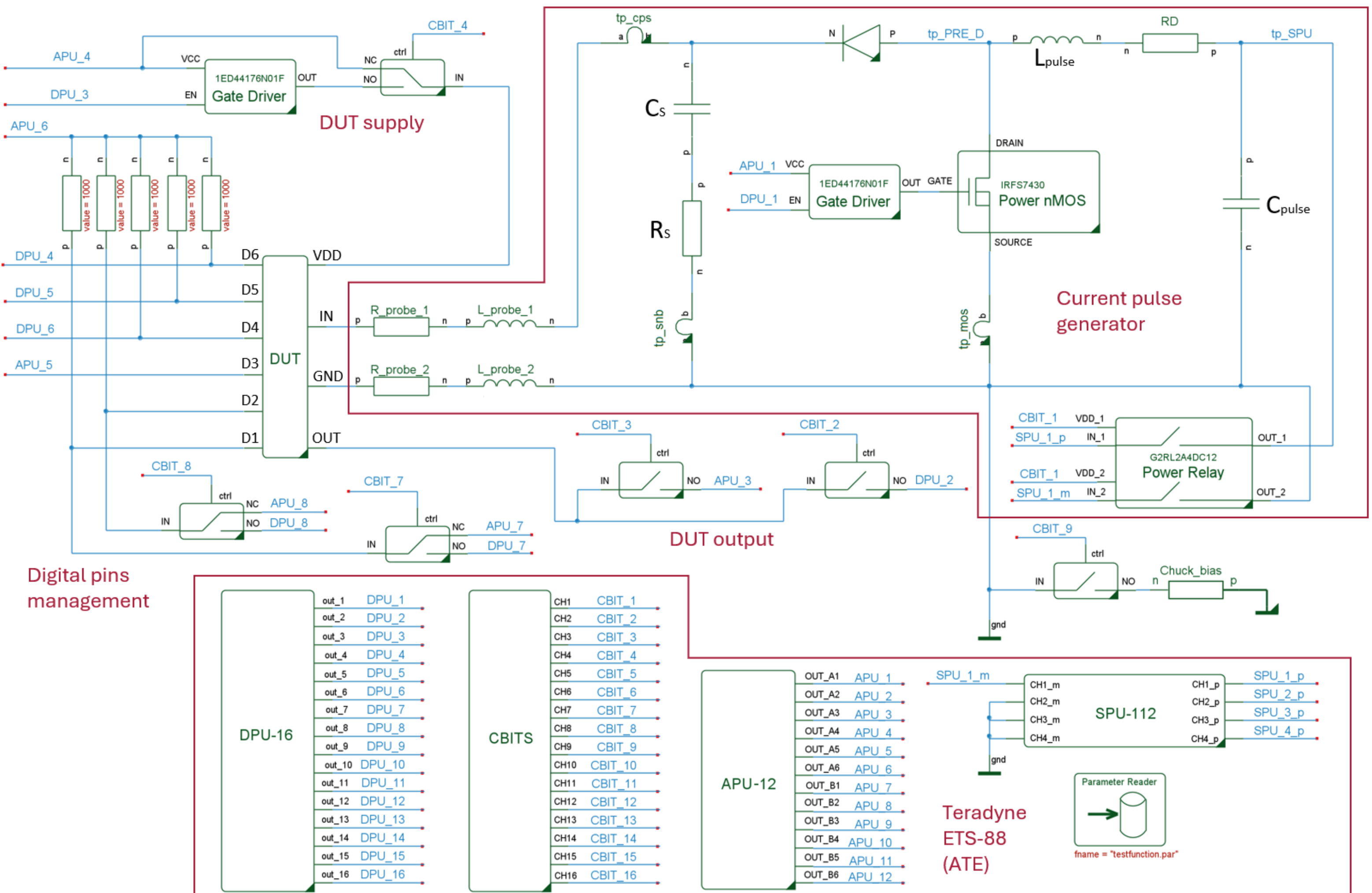



Figure 5.1: Top-level schematic of the probe-card for the test of the gate driver silicon wafers. Models of the ETS-88 resources are included within the schematic to highlight all the implemented connections.

The shape and general quality of the transmitted analog and digital signals is verifiable by generating signal transitions at the ATE channel outputs, and observing the resulting signals at the DUT pins, after their interaction with the test hardware components.

### 5.1.2 Test program

The TP follows an industry-standard structure, typically implemented for similar devices, and is comprehensive of the following principal TFs:

- *Coordinates* data-logs the position on the wafer for each tested device. During the data analysis, the coordinates are used to reconstruct the wafer maps and perform pattern-based analysis of the failures.
- *Contact* checks the contact robustness between device and socket (or probing needles). Known currents are injected at the device pads and the corresponding voltages, resulting from the presence of Electrostatic Discharge (ESD) protection diodes, are measured and characterised.
- *Quiescent_curr* measures and checks the device static current consumption between different device pins.
- *I2C* checks the robustness of the I2C communication between ATE and device. Test words are written and read back from all the digital registers of the device.
- *Clamp* measures and characterises the voltage clamping curve at the gate driver output. A digital signal triggers the voltage clamping action, and digital registers are used for configuration.
- *Wu_time* measures the device wake-up time, defined as the device response time when turned on in fault condition, with different loading configurations. Proper synchronisation of the involved resources is a crucial aspect of this test.
- *Trip* measures the value of the trip current, defined as the necessary current value to trigger the over-current protection mechanism. Current pulses with known amplitude are injected and the device response is evaluated. Digital registers are used for configuration.
- *Timer* measures different propagation delays related to the device functionality. The trends of the propagation delays with respect to the register settings are tracked and data-logged.

- *Burn* handles the e-fuses burning process for the device trimming. The positive result of the operation is checked, and a final comprehensive test on the burned device is performed.

# 5.2 Case-Studies

In the presented case studies the application of the framework allowed avoidance of faulty topologies and measurement solutions that would have forced troubleshooting and rework on the test hardware. The presented alternative solutions for the measurement of clamp voltage, trip current, and wake-up time are investigated within the framework and integrated into the manufactured probe-card.

## 5.2.1 Clamp voltage

The target of this measurement is the intermediate value of the rapidly changing voltage clamping curve, shown in Fig.5.2, referred to as *clamp voltage*. Circuit solutions to perform the measurement have been investigated within the framework, allowing preparation of all the necessary connections and resources within the probe-card schematic.

The original setup involved the measurement of the clamp voltage by employing one of the SPU-112 channels in voltage measurement mode. The slew-rate performance of the SPU-112 digitiser[2] is however not sufficient, and the resource is therefore unable to properly resolve the clamp voltage.

As shown in Fig.5.2, the behaviour of the SPU-112 in voltage measurement mode is effectively predicted by the test setup simulation, and a purposely implemented warning terminal message signals the exceeding of the resource slew-rate capability to the user, allowing early investigation and preparation of alternative solutions.

As shown in the figure, the chosen solution consists of indirectly measuring the clamp voltage by digitally triggering a clamp event and measuring the propagation delay between two known values of the voltage clamping curve via QTMU. The *start* trigger ($V_{start}$) is fixed at $11V$, while the *stop* trigger ($V_{stop}$) is gradually risen from $8V$ after each repeated clamp event.

When a significant decay of the propagation delay between two subsequent measurements is observed, the real clamp voltage is assumed crossed

[2]measured within a targeted experiment at $dV/dt_{max} \approx 0.2V/\mu s$

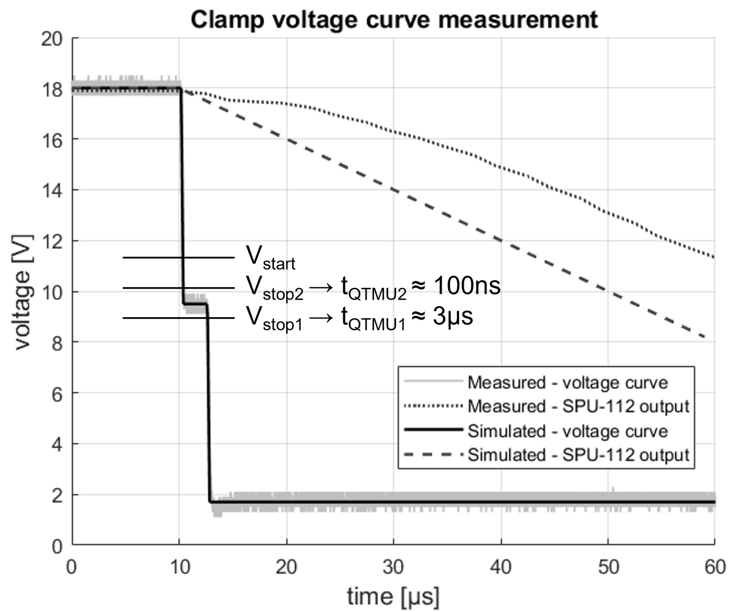



Figure 5.2: Clamp voltage measurement through SPU-112, comparison between framework prediction and bench measurement

```
-----------------------------------------
QTMU CH3- time readout received: 0.024000us
-----------------------------------------
-----------------------------------------
QTMU CH2- time readout received: 2.324000us
-----------------------------------------
```

Figure 5.3: Simulated QTMU output for the indirect clamp voltage measurement. For presentation, the measurement is performed with two different QTMU channels, and both results are shown simultaneously.

by $V_{stop}$, and the average between the last two values of $V_{stop}$ is accepted as the measured value of the clamp voltage:

$$t_{QTMU1} \approx 3\mu s \ , \ t_{QTMU2} \approx 100ns \quad \rightarrow \quad V_{clamp} = \frac{V_{stop1} + V_{stop2}}{2}$$

The QTMU output obtained within the framework during the test setup simulation is shown in Fig.5.3.

### 5.2.2 Current pulse generator

To trigger the over-current protection of the gate driver while preserving its integrity and functionality, a current pulse with specific characteristics ($t_{pulse} < 2\mu s$, $I_{peak} > 4A$) is required[3]. Given the limitations of the dedicated ATE resources, an application specific current pulse generator circuit capable of generating the target pulse by modulating the native ATE capabilities is designed and simulated within the framework.

The main criticality of the measurement is represented by the SPU-112, well-suited for high current applications and therefore selected for the pulse generation. Despite being able to supply the target current ($I_{max} = 12A$), the SPU-112 is not capable of fast enough switching, and would expose the DUT to high currents for a critical amount of time. An application-specific current pulse generator circuit is therefore designed.

The chosen architecture, shown within the probe-card schematic in Fig.5.1, is based on the charging of a large inductor-capacitor pair ($L_{pulse} = 470\mu H$, $C_{pulse} = 100\mu F$) and the well-timed switching of a n-channel power MOSFET for a brief period of time ($t_{pulse} = 0.75\mu s$).

When the MOSFET is turned off, the inductor $L_{pulse}$ acts a current generator, and discharges a semi-constant current through the DUT. The capacitor $C_{pulse}$, together with the snubber branch ($R_S$ - $C_S$), dampens unwanted voltage and current oscillations.

The amplitude of the current pulse is controllable by varying the SPU-112 current used to charge the $L_{pulse}$ inductor, and the pulse duration $t_{pulse}$ is controlled by the width of the digital pulse used to switch the MOSFET gate driver. The simulated time diagram of the circuit is shown in Fig.5.4.

To validate the robustness of the proposed methodology against the standard approach, a similar circuit has been designed with ideal components and

[3] The $t_{pulse}$ quantity is in this scenario defined as the length of time the current through the DUT is greater than $4A$.

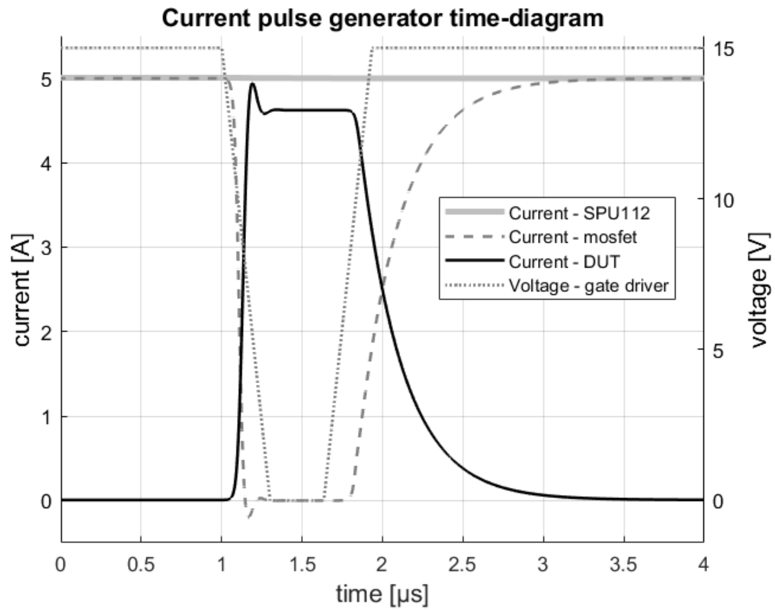



Figure 5.4: Time diagram of the current pulse generator circuit. Simulations are performed within the framework.

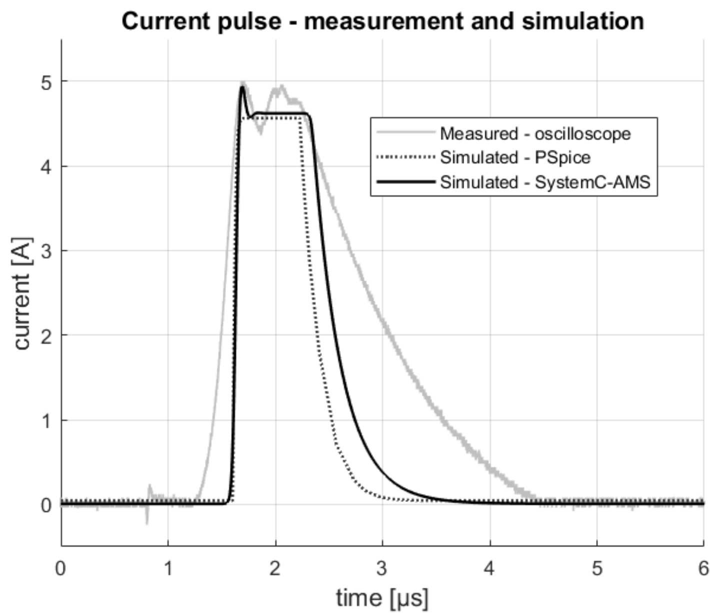



Figure 5.5: Current pulse generation - comparison between framework prediction, SPICE simulation, and bench measurement

by emulating the ATE with ideal sources. Preliminary simulations have been performed in the Cadence PSpice simulator, confirming shape and general size of the predicted current pulse.

The comparison between simulations and measurement of the current pulse generation is shown in Fig.5.5. The current pulse predicted by SystemC-AMS has $I_{peak} \approx 4.61A$ and $t_{pulse} \approx 0.72\mu s$.

After the integration within the physical probe-card, oscilloscope measurements of the current pulse have been performed, and the following can be observed:

Firstly, the measured current pulse presents an evident oscillatory behaviour during the peak current phase. Despite being underestimated in duration, this behaviour is predicted by SystemC-AMS and oversighted by PSpice. This effect results from the better fidelity of the modelled setup to the physical one, with respect to the simplified version simulated in PSpice. The SystemC-AMS setup results in more accurate predictions despite the greater analog accuracy offered by PSpice.

Secondly, the measured current pulse presents longer rise and fall times with respect to the ones predicted by SystemC-AMS and PSpice. This is likely due to parasitic passive components along the network that are not accounted for in the simplified models.

In conclusion, the framework offered valuable insight into the current pulse generation, and allowed preparation of a proper setup. Results for similar applications are expected to improve as the models of ATE and electrical components are furtherly developed.

### 5.2.3 Wake-up time

The target of this measurement is the is the DUT wake-up time, defined as the time required by the DUT to react when it is turned on while the fault condition is verified. Specifically, the DUT wake-up time is defined as the time between the rise of the supply pin VDD (turn-on), and the fall of the digital pin D1 (response to the fault condition).

The chosen resource to supply power to the DUT is the APU-12, well suited to supply intermediate voltages with good accuracy. Similarly to the previous case, the slew-rate performance[4] of the APU-12 is not sufficient and

[4]Measured within a targeted experiment at $dV/dt_{max} \approx 1.1V/\mu s$

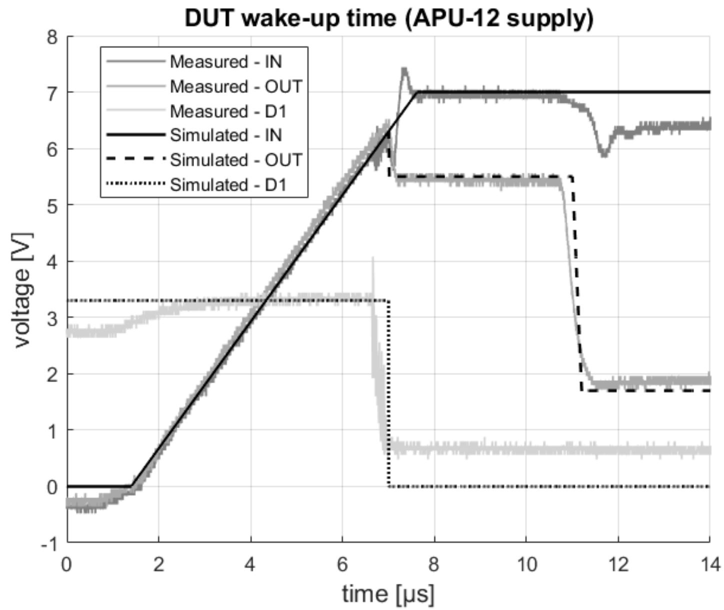



Figure 5.6: DUT wake-up time observed when the DUT is supplied directly by an APU-12 channel

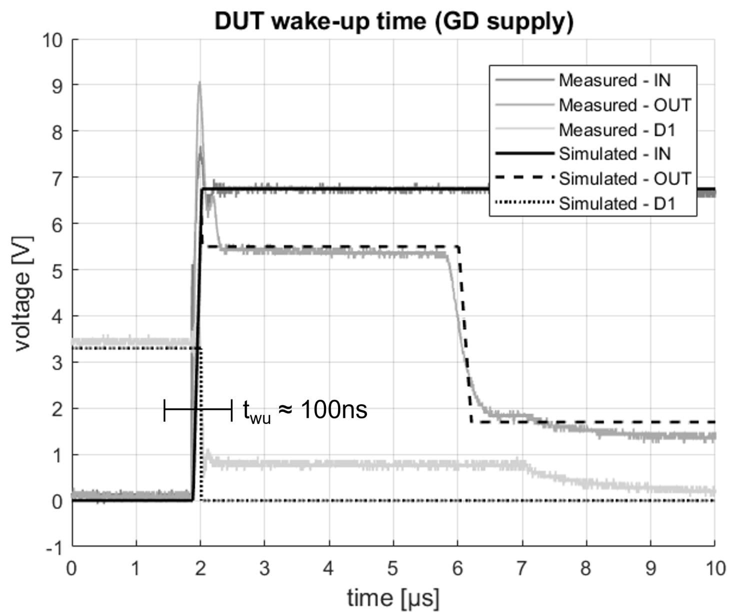



Figure 5.7: DUT wake-up time observed when the DUT is supplied by a gate driver controlled through a DPU-16 digital channel

leads to a slowly rising VDD, a slowly falling D1, and a general uncertainty and added error to the wake-up time measurement, as shown in Fig.5.6.

The investigated alternative solution involves the use of a gate driver to supply power to the DUT. As shown in Fig.5.7, this significantly decreases the rise time of VDD and the fall time of D1, and helps better defining the wake-up time measurement. In this case:

$$t_{wu} = t_{VDD=2V} - t_{D1=2V} \approx 100ns$$

To preserve the versatility of the APU-12 direct connection to the DUT supply pin, the possibility of alternatively connecting the gate driver is implemented within the probe-card through CBITS-controlled relays. As visible from the figures, the behaviour of the DUT under different supply conditions is effectively predicted by the framework, which is able to offer valuable insight and allow preliminary investigation of the available possibilities.

## 5.3 Data Analysis Results

The test program is executed on silicon wafers containing approximately 12,000 devices, and approximately 100 measured parameters are saved for each device. The produced raw data results therefore of relevant size, and a streamlined way to manage and analyse these data is mandatory.

The raw data are processed with a proprietary program that formats the acquired measurements into standardised tables, that can be opened in standard data analysis software. For this test solution, a Microsoft Excel installation equipped with purposely made proprietary tools, suited for test-data analysis, is employed to generate wafer maps, graphical distributions, and summary reports. An example of a complete wafer map is shown in Fig.5.8, and the relative yield summary is shown in Fig.5.9.

Comprehensive maps of the tested wafers can be generated by mapping the test data to the device coordinates indexed by the wafer prober, to visualise and investigate the emergence of failure patterns related to the position of the dies on the wafer. Wafer maps are especially useful when multiple wafers from a lot are tested, and the emergence of production-related failure regions can be observed on subsequently tested wafers, and used to provide useful input to diagnose and improve the wafer manufacturing process.

The data collected on the measurements presented as case studies ($V_{clamp}$, $I_{trip}$, $t_{wu}$) is shown in Fig.5.10, Fig.5.11, and Fig.5.12. The wafer maps show the clear presence of position-based deviations in the considered parameters.

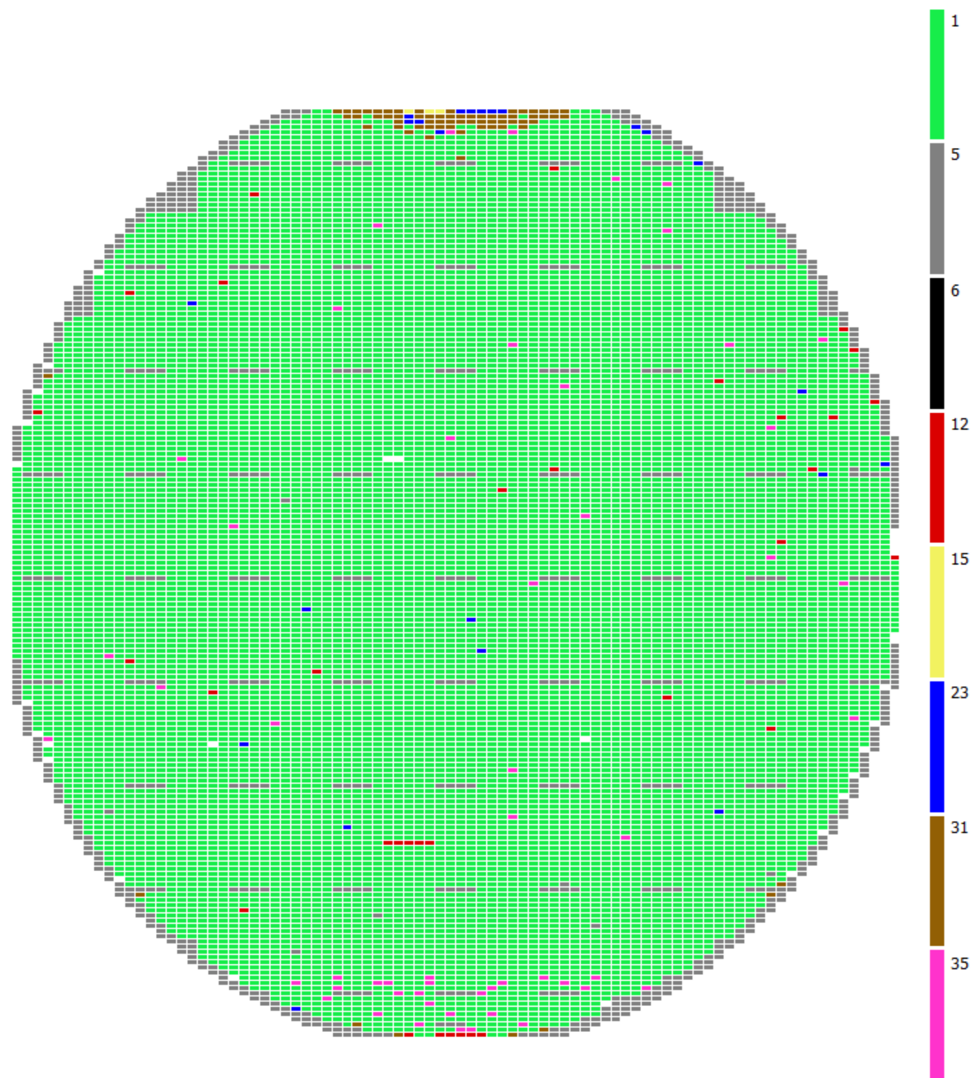



Figure 5.8: Wafer map from the test of a complete silicon wafer. Different failure mechanisms are highlighted in different colours to facilitate the recognition of failure patterns. Deviation from the ideal behaviour are noticeable near the top and the bottom of the wafer.

| Bin# | Parameters | Qty Failed | %Yield Loss |
|---|---|---|---|
| | Dataset | DATA | |
| | Test Program Name | PCV2_TP | |
| | Lot # | \ | |
| | Temperature Tested | 30°C | |
| | Total Tested | 12816 | |
| | Total Passed | 11917 | |
| | Total Failed | 899 | |
| | Yield % | 93.0% | |
| [1] | Pass | 11917 | 93.0% |
| [5] | Contact | 734 | 5.7% |
| [31] | ITrip | 93 | 0.7% |
| [6] | Contact | 21 | 0.2% |
| [12] | I2C | 21 | 0.2% |
| [35] | Burn check | 19 | 0.1% |
| [23] | Tclamp | 6 | 0.0% |
| [9] | IQCC | 3 | 0.0% |
| [24] | Iclamp | 2 | 0.0% |

Figure 5.9: Yield data summary from the test of a complete silicon wafer

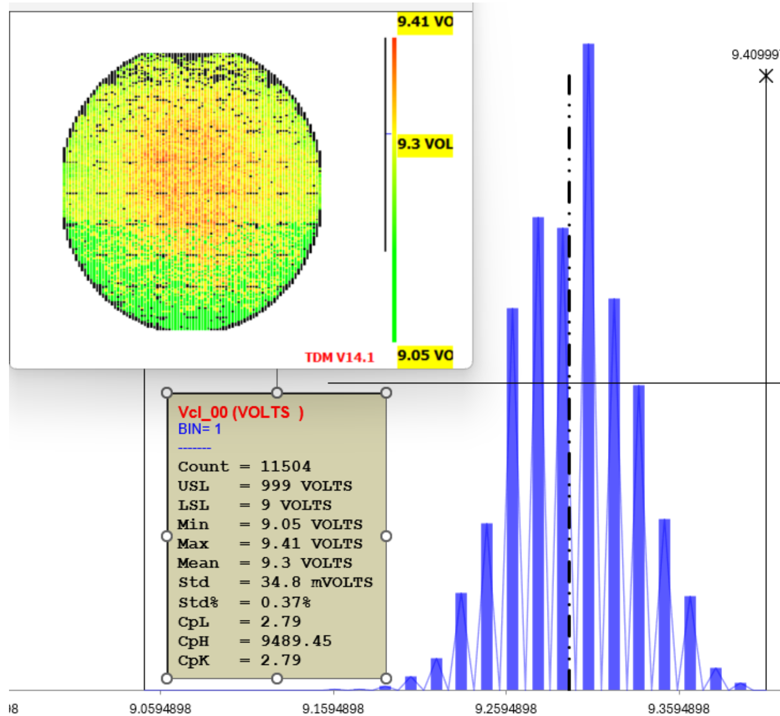



Figure 5.10: Distribution and wafer map for the clamp voltage measurement on a full silicon wafer

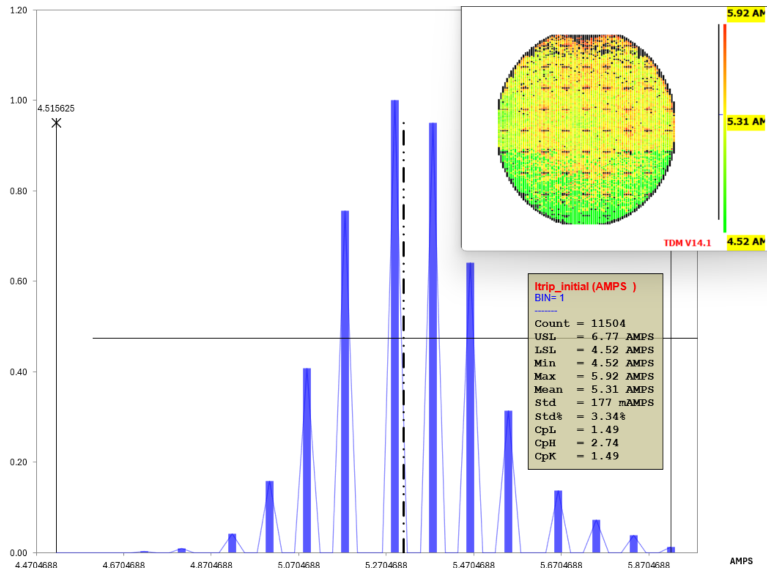



Figure 5.11: Distribution and wafer map for the trip current measurement on a full silicon wafer

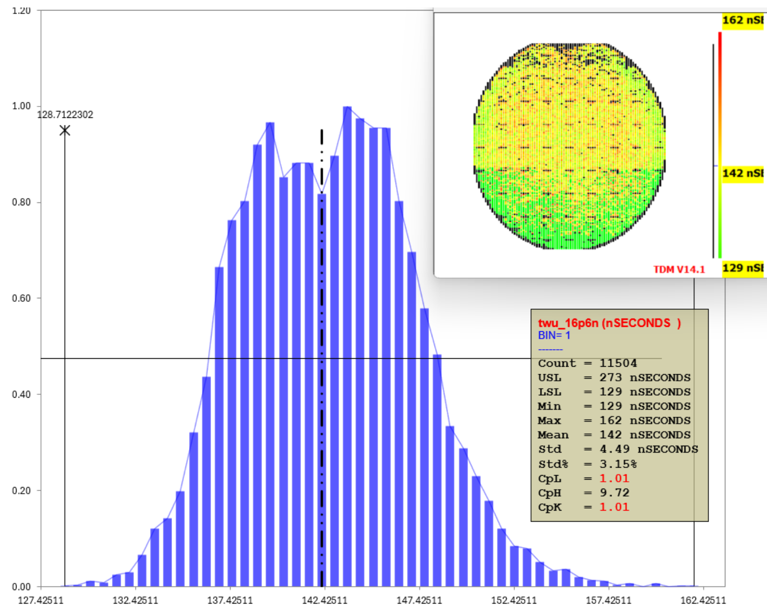



Figure 5.12: Distribution and wafer map for the wake-up time measurement on a full silicon wafer

# Chapter 6

# Application - Peak Current Test Concept Development

Isolated gate drivers provide the same core functionality as standard gate drivers while incorporating galvanic isolation between the low-voltage control circuitry and the high-voltage power circuitry, enhancing protection against noise, voltage spikes and interference effects. Multiple technologies are available to achieve the necessary isolation, including capacitive coupling and coupling based on transformers.

A key parameter for the characterisation of isolated gate drivers is the *high-level output peak current*, defined as the value of the DUT output current when the digital input pin is triggered, measured after the settling time.

Within the current industrial practice at the host company, the high-level output peak current is reported in data-sheets as a bench-measured parameter, characterised on a limited number of samples [39]. In the following case-study, test circuitry to perform the measurement at the ATE level is investigated and validated. Specifically, an application board to interface the packaged devices with the ATE and perform the peak current measurement is designed, manufactured and subsequently validated through targeted laboratory measurements.

This case-study serves as a powerful application of the proposed framework, in which all the principal implemented features are employed to assist the full development of an industrial test concept.

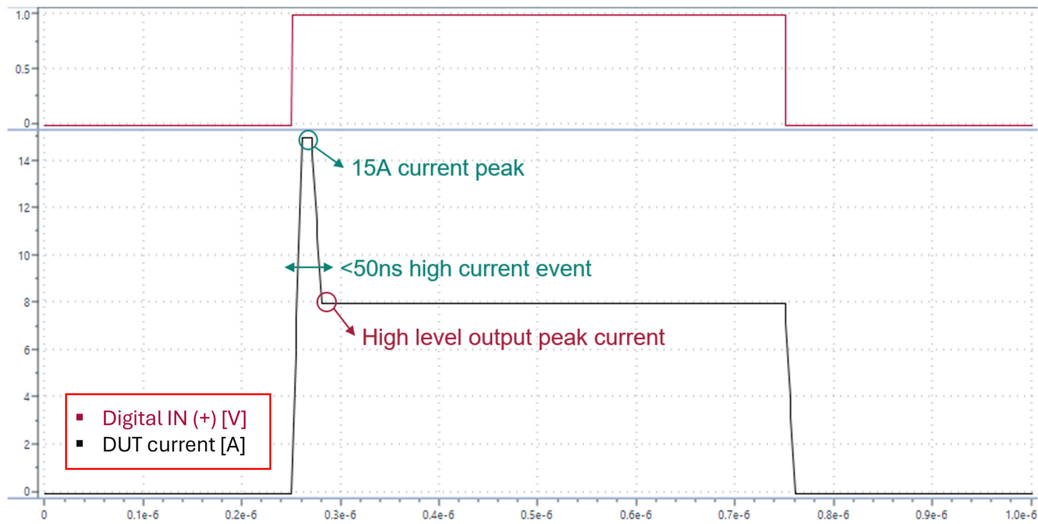



Figure 6.1: Behavioural model of the isolated gate driver output current

## 6.1 Motivation and Concept

A behavioural time-diagram of the DUT output current is shown in Fig.6.1. Measuring the high-level output peak current (referred to as *peak current* for the remainder of the chapter) at the ATE level presents two primary challenges that must be addressed and overcome.

The first critical aspects is related to the circuit supply. The current drawn by the DUT must be provided by the ATE and, given the limitations of the dedicated resource[1], a custom supply stage is required.

The constrains of the available ATE resources also impact the measurement of the peak current. As shown in the figure, the current transient event occurs within less than $50ns$, and a sufficiently high bandwidth is required for the acquisition. A dedicated stage that provides fast enough acquisition must therefore be designed.

Overall, an application-specific circuit that satisfies all the project specifications needs to be developed and validated. Particular focus must be placed on the long-term reliability of the measurement stage, to allow integration into the standard test hardware for isolated gate drivers.

[1]The power output must be maximised. The SPU-112 is therefore employed.

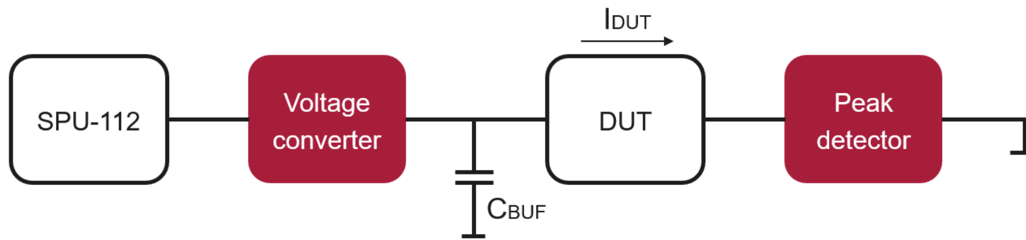



Figure 6.2: Conceptual block diagram of the peak current measurement circuit

### 6.1.1 Board concept

In Fig.6.2 the conceptual schematic of the test hardware is presented. Excluding the ATE and the DUT, it is comprehensive of three main components:

A **buffer capacitor** placed at the DUT supply pin is pre-charged before the test and used to provide the current drawn by the DUT. The capacitor is dimensioned optimising voltage drop, charge, and discharge times.

A **voltage converter** stage placed before the DUT can be used to boost the output voltage of the SPU-112, effectively creating an intermediate I-V range to optimise the buffer capacitor charging time. The voltage converter incorporates a bypass mechanism to allow direct charging of the buffer capacitor if required.

A **peak detector** stage performs the peak current measurement indirectly. The DUT output current is sensed on a shunt resistor and converted into a voltage. Following proper amplification and sampling on a peak detector circuit, the voltage measurement can be performed on the sampled voltage value using an APU-12 channel without time constraints.

### 6.1.2 Test program

The TP is developed within the standard ETS-88 MST software, and is comprehensive of three main TFs:

- *HW_check* implements the hardware verification functionality, which ensures long-term circuit reliability. The DUT circuitry is completely bypassed by relays, and the SPU-112 is operated in current mode to directly supply known current values to the measurement stage. The

measured voltages are compared with the expected values to detect drifts and anomalous behaviours in circuit operation.

- *Peak_meas* manages the DUT power-up sequence and the triggering of the digital signal, and performs the formal measurement on the peak current. This TF implements a conversion function, that translates the measured voltages into the corresponding current values.
- The *HW_debug* TF contains code for the test of various circuit functionalities through code sections that can be individually activated.

## 6.2 Test-Board Design

The test board has been designed within the SystemC-AMS framework, with occasional use of Cadence PCB to run PSpice double-check simulations on the most critical aspects of the circuit functionality.

Subsequently, the board schematic is imported and converted to the Eagle PCB software to finalise the board design and prepare the layout for the manufacturing process.

### 6.2.1 Power booster

The SPU-112 is capable of delivering up to $12A$ in its highest power setting. The current rise time of the resource is however insufficient to support the rapid current spike.

The straightforward solution to this limitation is the adoption of a buffer capacitor to store charge before the test, and supply the required current during the event, at the expense of the additional charging time.

The minimum value of the buffer capacitor can be found by integrating the current curve shown in Fig.6.1, according to the following relation:

$$I = C \cdot \frac{dV}{dt} \rightarrow C_{min} = I \cdot \frac{\Delta t_{max}}{\Delta V_{max}} \tag{6.1}$$

$\Delta t_{max}$ is the maximum total on-time triggered by the digital signal. Given the test requirements, a maximum total on-time of $100\mu s$ is assumed[2]. $\Delta V_{max}$ represents the maximum allowed voltage drop due to the presence of the contact resistance. Considering the DUT on and off-voltages:

$$\Delta V_{max} = VDD_{max} - VDD_{min} = 40V - 15V = 25V$$

[2]The brief duration of the initial current peak relative to the total event time justifies the assumption of constant current.

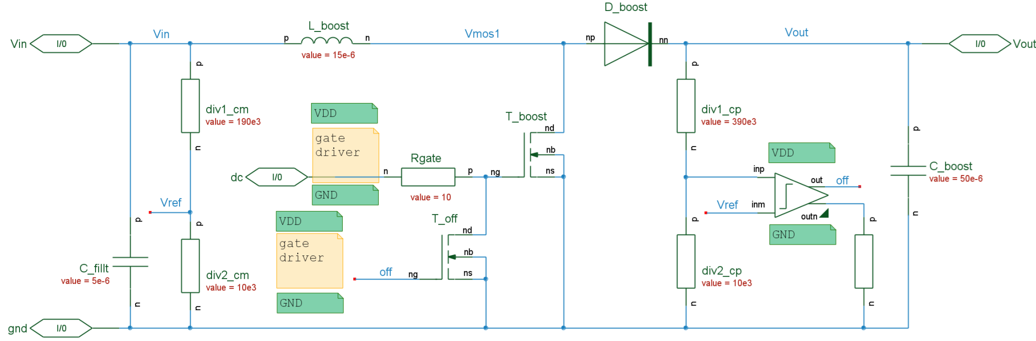



Figure 6.3: Schematic of the booster converter to improve the SPU-112 current capability

From Eq.6.1, results:

$$C_{min} \approx 48uF \rightarrow C_{buffer} = 50uF$$

If the buffer capacitor is properly charged during the circuit start-up, it is capable of supplying all the necessary current during the high current event, while maintaining the voltage drop within acceptable limits.

The I-V setting of the SPU-112 with the highest current output ($12A - 30V$) is chosen to rapidly charge the buffer capacitor. Considering a maximum predicted socket contact resistance of $1.5\Omega$, the initial supply voltage required at the DUT supply pin to avoid turn-off during the high current event is:

$$VDD_{init} = VDD_{min} + I_{max} \cdot R_{cont} = 37.5V$$

Consequently, the highest current setting is insufficient to provide the target voltage. A first possible solution is to employ the next available I-V setting for the SPU-112, $0.5A - 100V$, at the expense of an increased capacitor charging time, which can be estimated as:

$$t_{charge} \approx 5 \cdot \tau = 5 \cdot \frac{C \cdot V_{src}}{I_{max}} = 5 \cdot \frac{50\mu F \cdot 37.5V}{0.5A} \approx 20ms$$

This charging time is considered acceptable given the single-measurement nature of this test.

For the future scalability of the methodology, an option to reduce the buffer capacitor charging time by at least an order of magnitude is however investigated.

A voltage booster, implemented as a duty-cycle controlled booster converter with a comparator-based control loop, is integrated into the test hardware[3]. The voltage booster effectively functions as a SPU-112 range converter, that transforms the $12A, 30V$ I-V setting into an intermediate $8A, 40V$ I-V setting using a duty cycle $D = 0.5$.

This range transformation significantly reduces the buffer capacitor charging time:

$$t_{charge} = 5 \cdot \frac{50\mu F \cdot 37.5V}{8A} \approx 0.25ms$$

The schematic of the designed voltage converter is shown in Fig.6.3.

### 6.2.2 Current peak detector

For the initial iteration of the test concept, two version of the peak detector circuit have been designed and integrated into the test hardware. The two versions are based on the same topology, but differ in the nature of their output stage.

The current waveform produced by the DUT ($I_{peak} \approx 15A$) is firstly converted into a voltage waveform ($V_{peak} \sim 1.5V$) via a current sensing stage, comprehensive of a $10m\Omega$ shunt resistor and a $40dB$ ($G = 100$) low-noise operational amplifier[4].

A sample-and-hold circuit follows this stage. The amplified voltage charges a sampling capacitor that is specifically dimensioned to capture the voltage waveform at the appropriate instant. The sampled voltage can be subsequently measured using an APU-12 channel[5], and the value of the peak current can be derived by applying a specific mathematical relationship, calculated during the circuit calibration phase.

The first version of the current peak detector, shown in Fig.6.4, uses a high-output-current ($I_{out} = 250mA$) operational amplifier as a current-boosting stage. The first sizing of the sampling capacitor is done considering as a target the sampling of the initial high current peak itself, occurring after approximately $20ns$ after the digital trigger:

[3]The voltage booster is not used for the validation phase presented in this dissertation.

[4]The gain figure is presented considering the OpAmp-based peak detector.

[5]The voltage is acquired without time constraints, aside from the capacitor leakage.

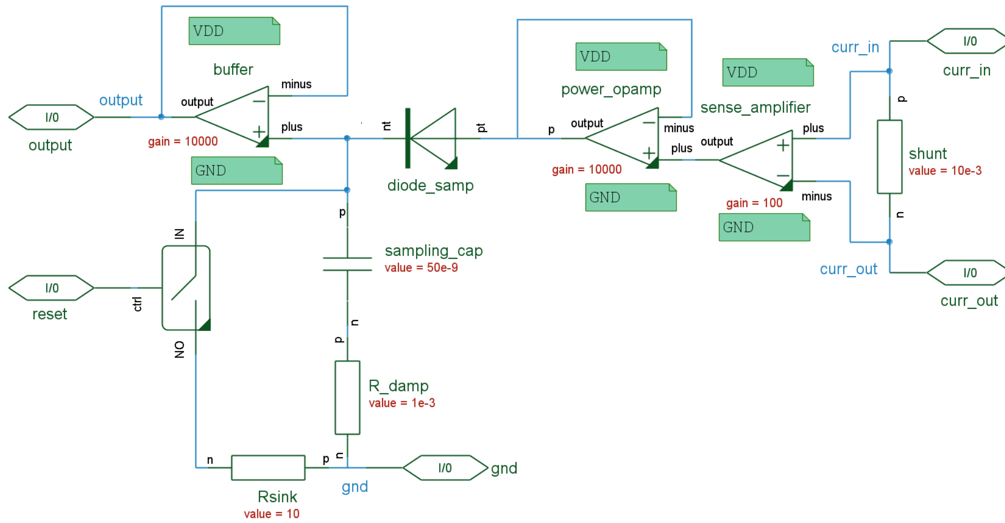



Figure 6.4: Schematic of the peak detector circuit based on a shunt reading stage and a current-booster amplifier

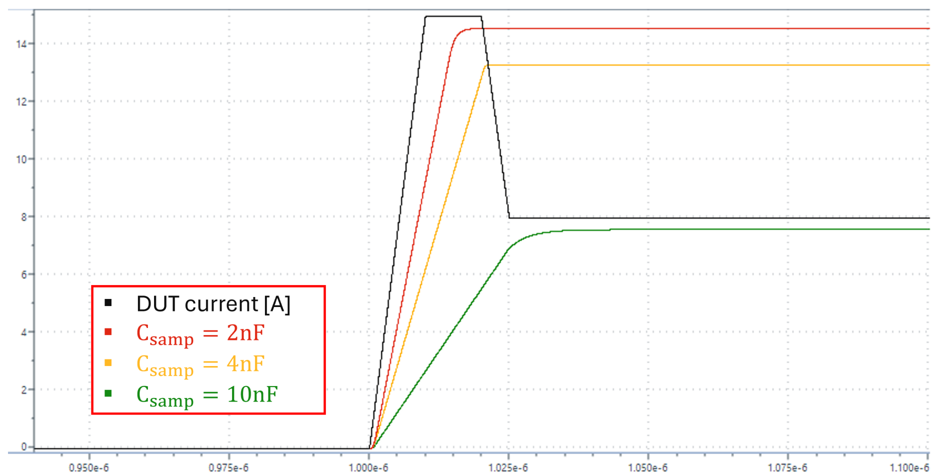



Figure 6.5: Simulation of the current peak sampling with different values of $C_{samp}$. With the right value of $C_{samp}$ the current value immediately after the initial peak is intercepted and sampled.

$$C_{samp} = \frac{I \cdot \Delta t}{\Delta V} = \frac{250mA \cdot 20ns}{1.5V} \approx 3nF$$

As shown in Fig.6.5, the value of the sampling capacitor is subsequently increased until the voltage rise across $C_{samp}$ is sufficiently slowed down to intercept the current waveform at the right instant.

The peak current is defined as the current value immediately following the initial peak, to account for oscillations and losses in the real circuit. In simulation, a final theoretical value of $10nF$ is determined for the sampling capacitor[6], and included into the initial iteration of the test hardware.

The second version of the current peak detector uses a high output current BJT ($I_{out} = 3A$) for the current-boosting stage, allowing the use of an intrinsically larger sampling capacitor ($35nF$).

### 6.2.3 Test bench

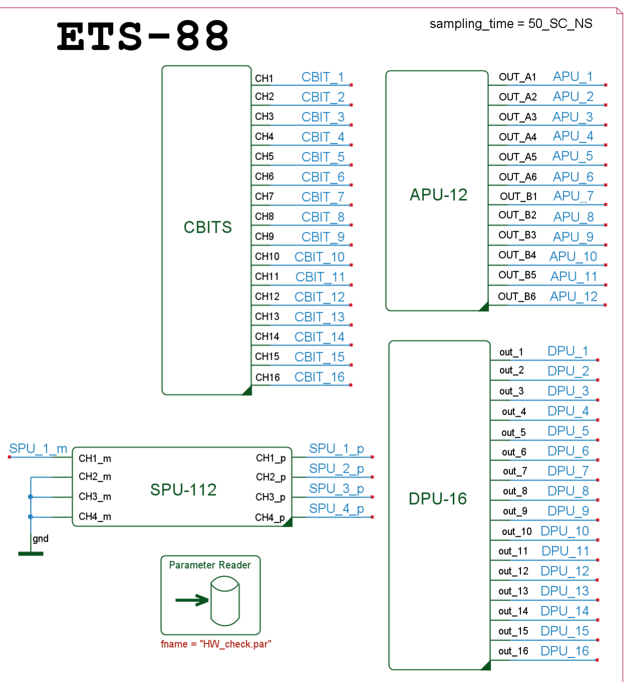
The COSIDE SystemC-AMS test bench for the developed hardware is shown in Fig.6.6. The simulation setup includes: Models of the Teradyne ETS-88 employed to generate all the required signals and perform the relevant measurements, test hardware comprehensive of the power supply unit and the two peak detector circuits, and a functional model of the DUT to emulate the high-current event. The three TFs are associated to three different parameter files, enabling versatile debug of the test concept.

The different circuit configurations (calibration and measurement with the two peak detectors) are selected on the test hardware via CBIT-controlled relays. Within the simulation environment, ELN dummy connectors are toggled on and off[7] to replicate the same functionality.

This case study is an example of a lower-level investigation focused on the analog behaviour of the involved components. The NgSpice and PWL libraries are therefore employed.

[6]This value is later adjusted on the physical hardware following measurements

[7]The *ignore* feature is used in COSIDE to exclude components from the netlist.



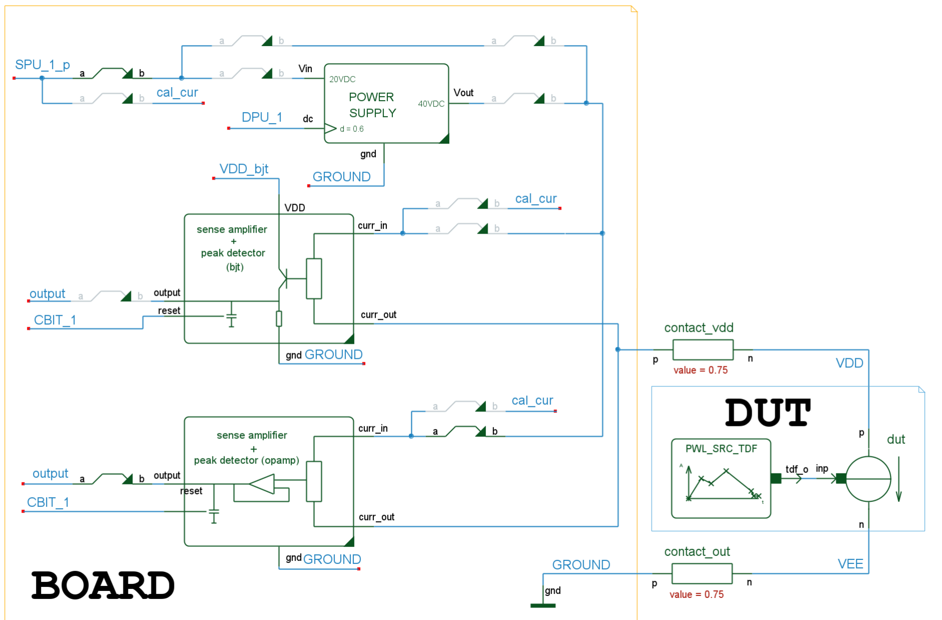




Figure 6.6: Test bench for the peak current measurement circuit. The ETS-88 models are included to highlight the implemented connections.

## 6.3 Eagle PCB Conversion

The test hardware schematic is converted to the Eagle PCB software for the manufacturing process. Eagle PCB facilitates efficient parallel design and layout, by tracking schematic modifications and mirroring them in the layout view in real time.

### 6.3.1 Test-board design

In this phase, the ideal or modelled components must be replaced with suitable commercially available options. Particular attention must be given to the components selection, ensuring compliance with all specified requirements. Additional supply paths, connections, and electrical parts must be incorporated as needed for specific components, in accordance with recomendations provided in their data-sheet application sections.

The final board design, as implemented in the Eagle PCB software, is shown in Fig.6.7.

### 6.3.2 Test-board layout

The layout of the test hardware is realised following established industrial PCB layout practices [40]. The final layout-view is shown in Fig.6.8.

The board layout follows a standard template for test concept application boards, featuring specialised connectors on both sides of the board that carry all the necessary signals to a bottom board for routing to the ETS-88.

A total of five routing layers is used, which are split into three layers for signal routing, one layer dedicated to power paths[8], and one layer fully reserved to the ground plane to minimise ground voltage swinging and ensure a direct ground-return path for every circuit node.

To minimise heat generation and mitigate damage risk, the high-current paths are implemented using larger traces and vias. Large capacitors are placed as close as possible to their interacting components to minimise trace length and reduce the risk of interference.

An extensive set of test points is included in the schematic to facilitate the validation process.

The final manufactured board is shown in Fig.6.9.

[8] A standard supply voltage of $12V$ is used.

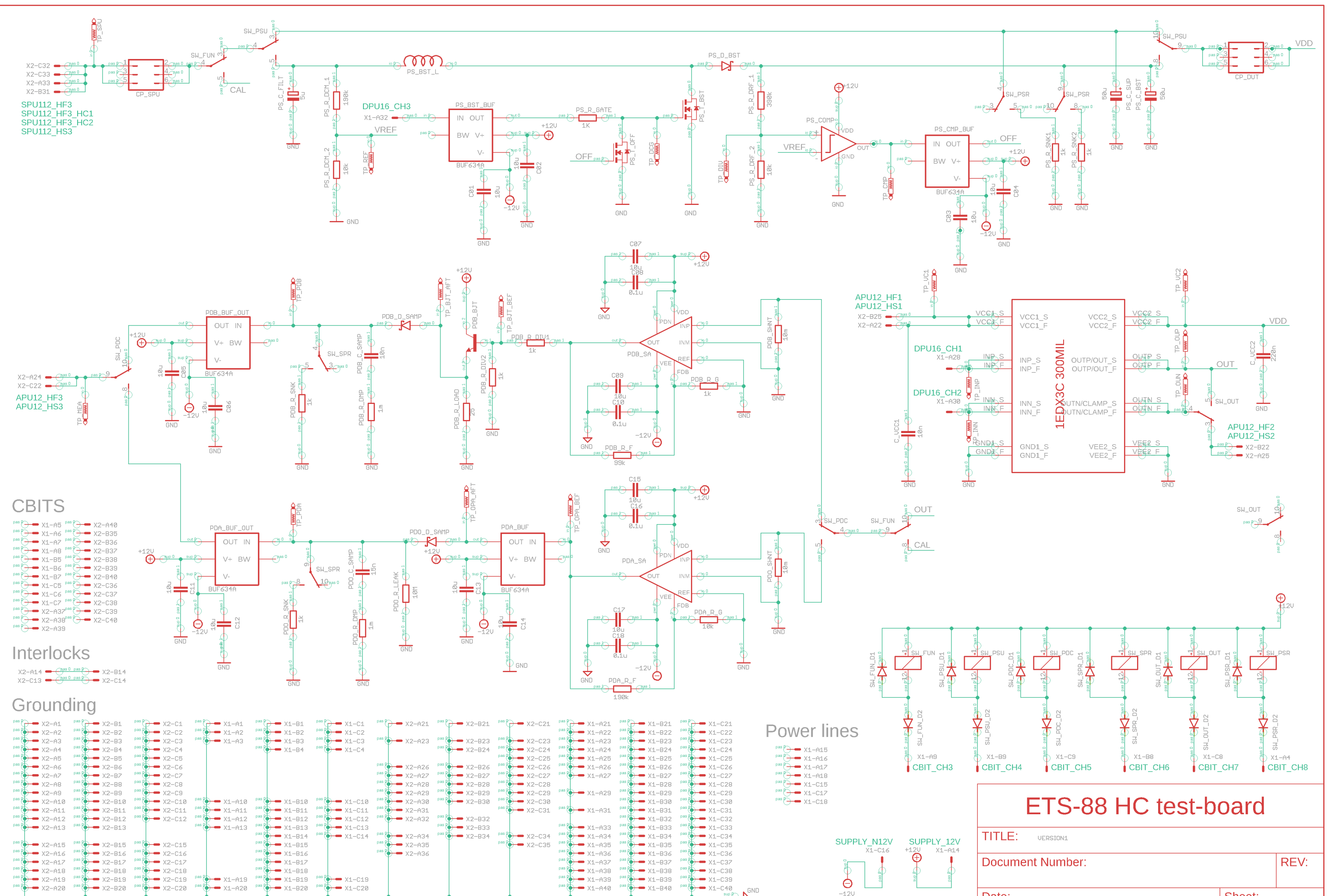



Figure 6.7: EAGLE PCB design view of the peak current measurement circuit

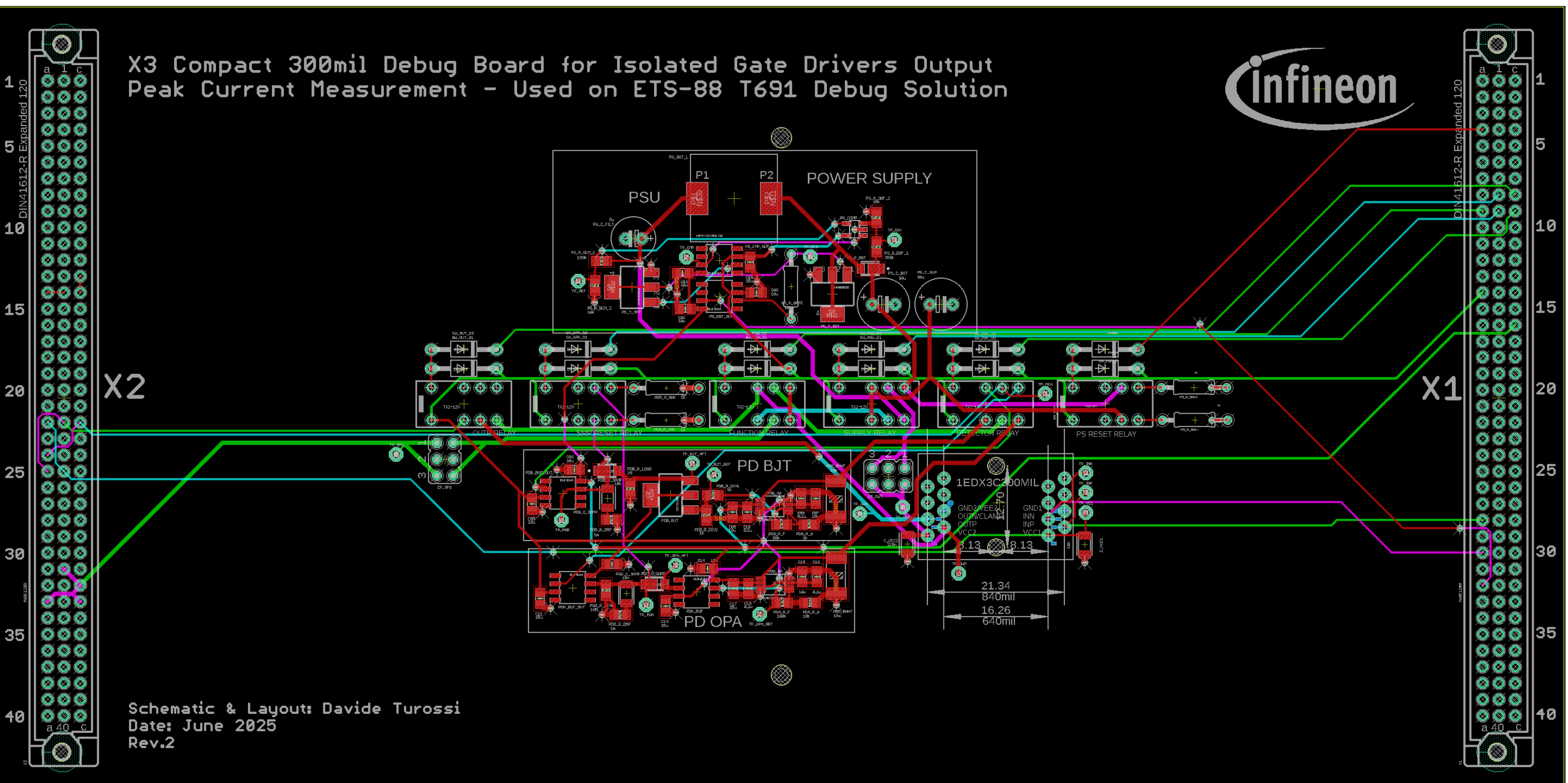



Figure 6.8: EAGLE PCB layout view of the peak current measurement circuit

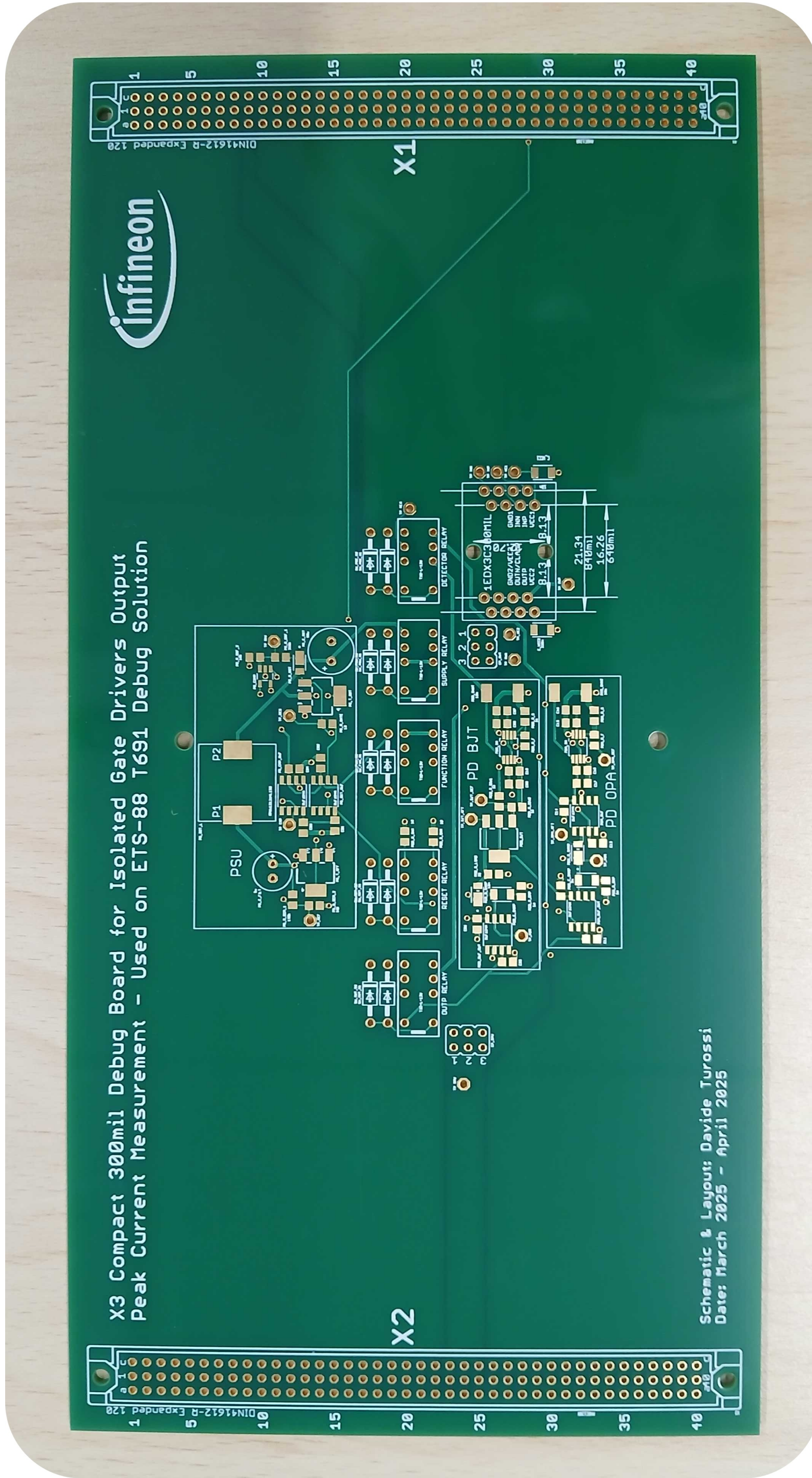



Figure 6.9: Manufactured test board for the output peak current measurement (the board is shown unpopulated from components for clearer visibility)

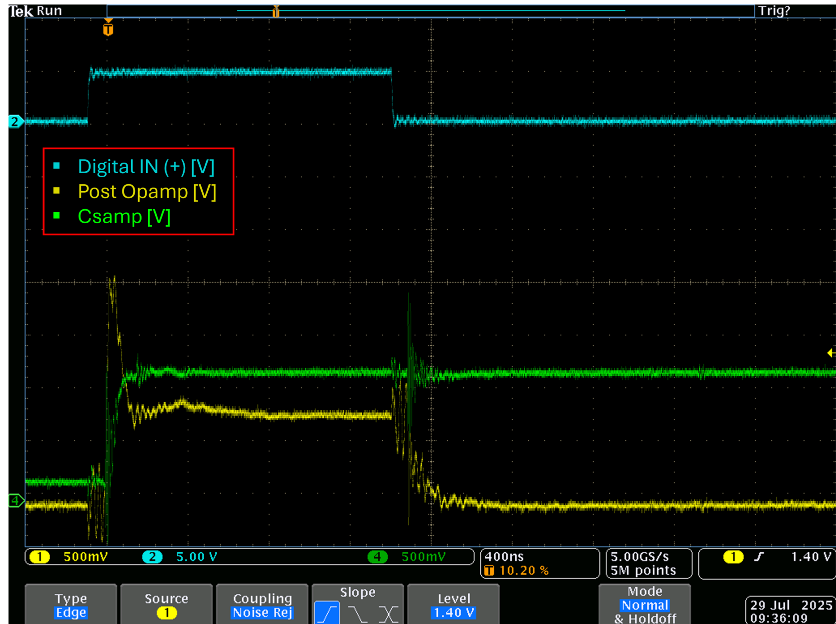



Figure 6.10: Oscilloscope reading of the peak current measurement event on the manufactured test hardware

## 6.4 Laboratory Measurement

The standard validation process defined at the host company involves a series of formal measurements targeted at assessing the accuracy, repeatability and reliability of the peak current measurements performed by the test circuit.

As an example, the oscilloscope reading of the measurement of a current peak event is shown in Fig.6.10.

### 6.4.1 Calibration process

Following an initial assessment of the overall functionality and robustness of the hardware, the calibration process is carried out.

As previously mentioned, in calibration mode the relays on the test board configure the circuit to completely bypass the DUT related circuitry, and the SPU-112 is operated in current mode to supply a known current level directly to the peak detector shunt. The APU-12 measurement is subsequently performed, and the resulting voltage is correlated to the sourced current value.

This process is iterated for current levels ranging from 0 to 12A, and the resulting data are interpolated to construct a mathematical relationship

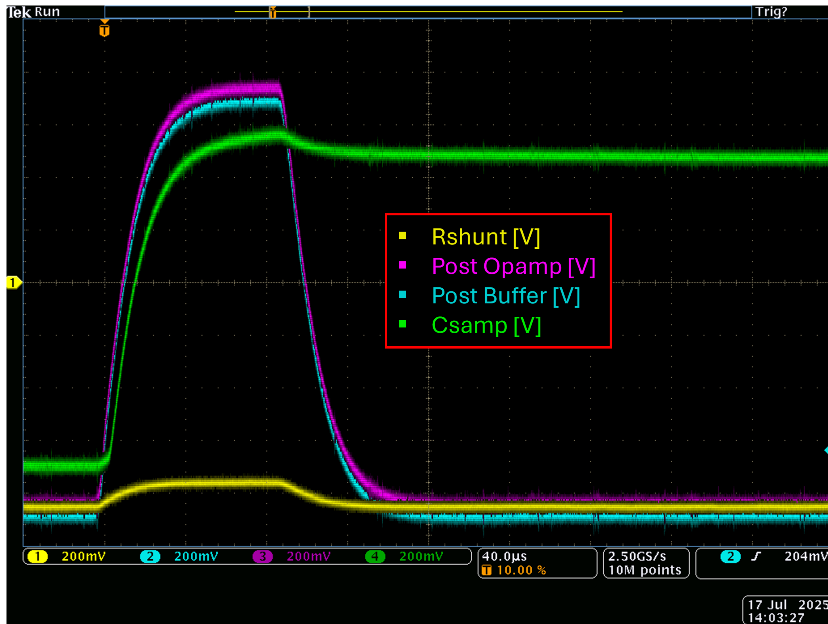



Figure 6.11: Oscilloscope reading of the peak current measurement circuit calibration

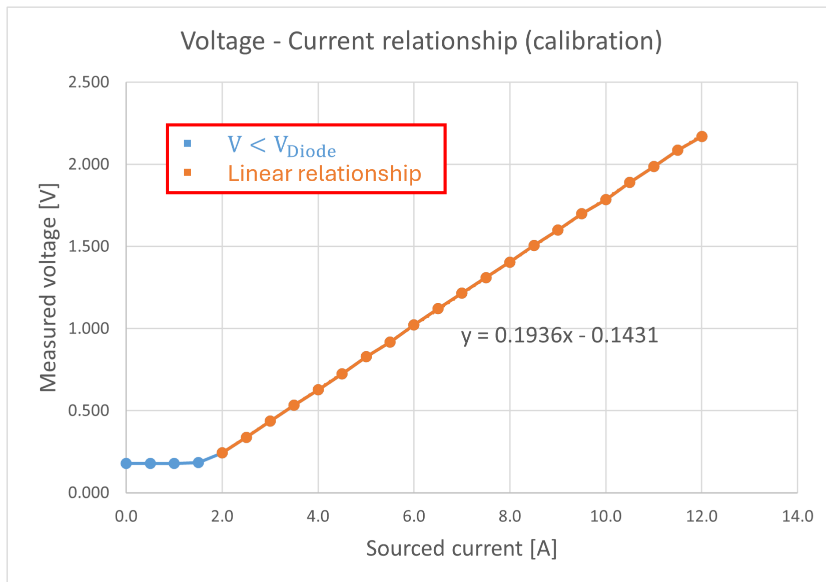



Figure 6.12: Relationship between measured voltage and sourced current in calibration mode. A current threshold of approximately $2A$ is necessary to activate the peak detector diode.

relating any measured voltage to the corresponding current value[9]. The derived relationship is:

$$V_{meas} = 0.197 \cdot I_{source} - 0.143 \;\rightarrow\; I_{peak} = \frac{V_{samp} + 0.143}{0.197} \tag{6.2}$$

The obtained relationship is inverted and applied during peak current measurements to derive the current value from the sampled voltage. The oscilloscope reading from the circuit calibration is shown in Fig.6.11, while the voltage-current relationship data are reported in Fig.6.12.

For the remainder of the chapter, the peak current values derived from Eq.6.2 are directly given as results.

### 6.4.2 Repeatability tests

To evaluate the repeatability of the hardware measurements, two sets of time-delayed measurements are carried out and subsequently correlated.

A total of 40 devices, classified as low-current and high-current devices, are characterised in sequence. The hardware is powered down and, following an appropriate cool-down time, the measurement sequence is repeated.

The resulting data are analysed in terms of Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE), to screen for systematic drift in the hardware performance and to asses the overall repeatability. The obtained results are reported in the table below:

| Low-current devices | | | High-current devices | | |
|---|---|---|---|---|---|
| N | MPE | MAPE | N | MPE | MAPE |
| 20 | 0.37% | 0.76% | 20 | 0.01% | 0.61% |

For both the low-current and high-current devices, the data do not indicate any significant systematic drift in the measurements, and for both low-current and high-current devices, the absolute deviation remains within the accepted limit (conventionally set at MAPE$\leq$ 1%).

### 6.4.3 Looping tests

Looping tests represent a crucial evaluation for measurement setups in ATE environments, as they verify the hardware reliability under large quantities of measurements with minimal delay between the events.

[9] The calibration current is limited by the 12A SPU-112 current limit.

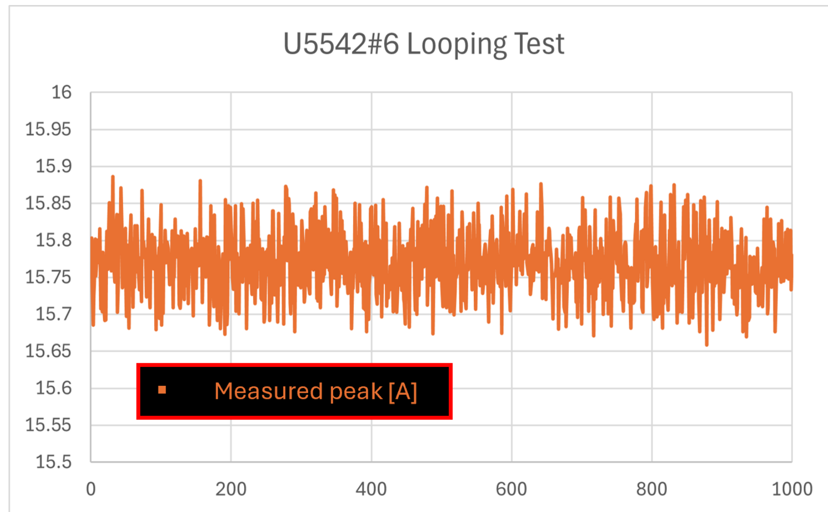



Figure 6.13: Results of the 1000-tries looping test performed on high-current devices

In this specific application, 1,000 looped measurements have been performed on four distinct devices to assess drift in the circuit performance, and the possible onset of hardware degradation. It is important to observe that careful attention must be paid to preserve environmental conditions (temperature, humidity, etc.) during looping tests, which often span several hours of run-time, and may be affected by gradual drifts in the environmental conditions (i.e. local temperature rise caused by the test setup power dissipation).

The results of the loping tests are presented in Fig.6.13, and summarised in the table below:

| Results [A] | Low-current devices | High-current devices |
|---|---|---|
| Avg | 4.68 | 15.77 |
| Min | 4.63 | 15.66 |
| Max | 4.73 | 15.89 |
| Stdev | 0.02 | 0.04 |

Looping tests are also performed with the contact between device and socket broken after each test, to evaluate the sensitivity of the measurement to the contact resistance. An average increase in the standard deviation of approximately $0.045A$ is observed across the different devices. This increase is expected and remains within the accepted limit.

### 6.4.4 Bench correlation

Bench correlation tests are performed to verify the validity of the ATE measurement against the bench measurements performed on the minimum possible hardware.

A purposely designed application board, shown in Fig.6.14, is employed for bench measurements. It incorporates all the required signal paths and accessory components to manage the DUT operation and perform the target measurements. The interaction with the DUT is managed using standard laboratory equipment, including function generators, power supplies, and oscilloscopes.

Due to the sensitive nature of the measurement, two different setups have been adopted to acquire the current curve, with their schematics schematics shown in Fig.6.15.

In the first approach, analogously to the automatic test hardware, the current peak is read on a $10m\Omega$ shunt resistor. The oscilloscope reading is shown in Fig.6.16. Given the small voltage values involved, this bench test shows limited reliability and significant sensitivity to external conditions. The second approach is therefore preferred.

In the second approach, a high-bandwidth current probe is used to directly measure the current peak on a looping wire[10]. This method is characterised by a limited bandwidth which suppresses spikes and high frequency behaviour. The oscilloscope reading is shown in Fig.6.17.

Results for the Percentage Error (PE) obtained from the bench correlation tests on two sample devices are reported in the table below as an example:

| Device | ATE [A] | Probe [A] | PE |
|---|---|---|---|
| Low-curr. | 4.52 | 4.50 | -0.45% |
| High-curr. | 14.99 | 15.48 | +3.19% |

The results indicate, as anticipated, a greater error in high-current devices. In all conducted trials, the PE remains below the established limit (5%), and the results are therefore considered acceptable.

[10]The wire resistance must be minimised to reduce heating and prevent over-voltages.

Figure 6.14: Application board for the peak current measurement circuit bench correlation

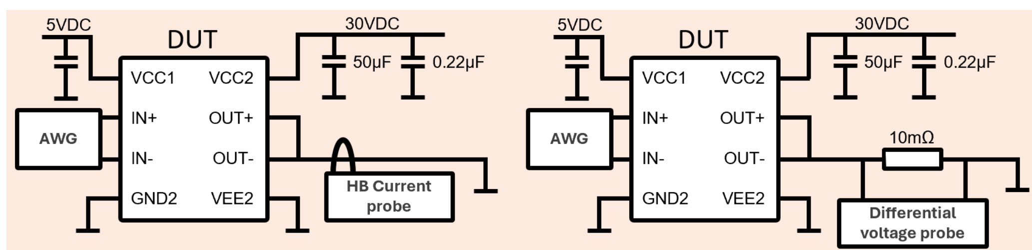



Figure 6.15: Schematic view of the measurement setup for high-current peak output current bench correlation

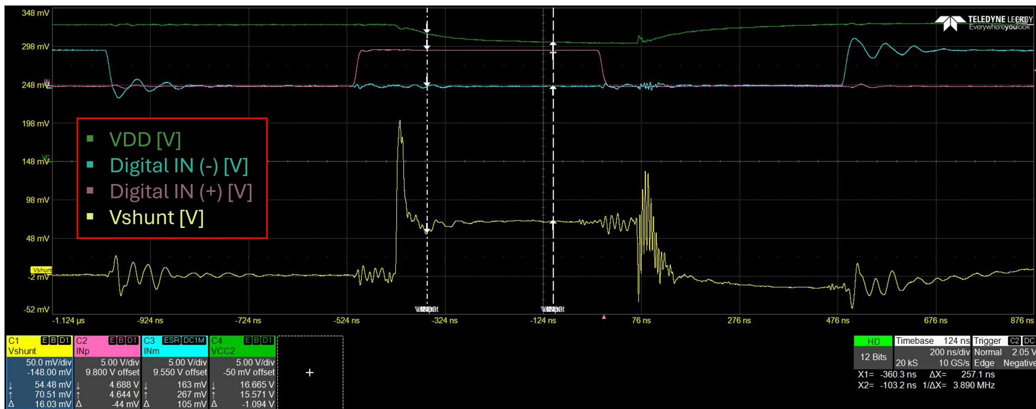



Figure 6.16: Oscilloscope reading of the shunt-based bench correlation test

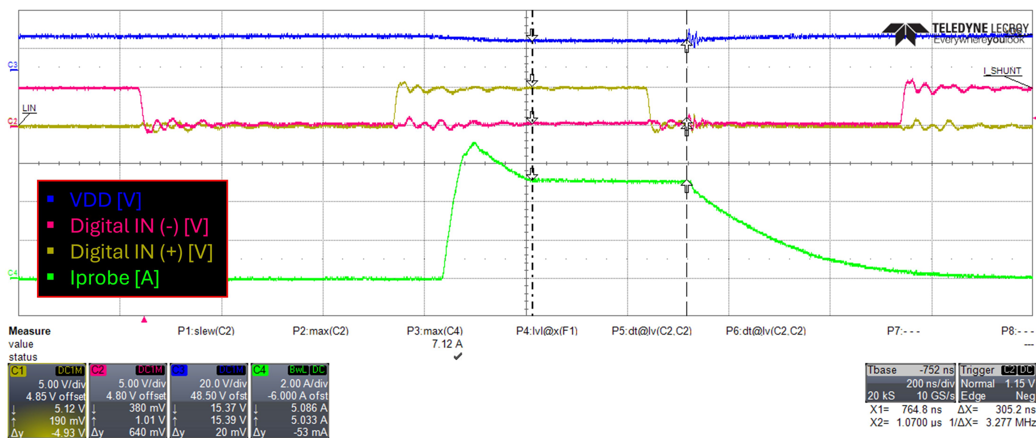



Figure 6.17: Oscilloscope reading of the probe-based bench correlation test

# Chapter 7

# Conclusion

In this dissertation, a methodology based on a SystemC-AMS virtual testing framework for the development of test hardware has been proposed.

The proposed framework has proven, during its development and within two innovative industrial test scenarios, to be an effective tool for supporting automatic test development activities.

In detail, during the development of a test solution for an innovative industrial device, the framework demonstrated the capability to predict the behaviour of the test setup (ATE, test hardware, DUT) under different operating conditions and to allow preparation of specific circuit configurations for measuring crucial electrical quantities, potentially saving days of debugging on the physical test hardware and allowing proper optimisation of the test program in terms of test time and resources usage.

During the development of an innovative test concept, the framework demonstrated the capability to simulate test setups with greater focus on the analog fidelity, especially through the integration of NGSpice components, leading to a successful measurement campaign and to the finalisation and integration of the test concept into future standard test solutions.

The observed capabilities of the framework show appreciable potential for further expansion and improvement, and for employment in more complex test scenarios involving larger test hardware and more sophisticated test functions.

The first functional iteration of the framework is currently in use within the test development teams of the host company to gather additional usage data and feedback, and to conduct a formal performance assessments under a variety of different working conditions.

# Chapter 8

# Other Projects

In the following chapter, the main side-projects followed during the Ph.D. research period are briefly introduced, and the main produced results are outlined. The two presented projects include:

- Development of an industry-level automatic testing laboratory, named ATLab. The development has been followed from the identification of the proper laboratory room and machinery, to the definitive inauguration to operational state.
- Design of a Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) for next-generation hadron therapy imaging devices. The converter has been designed and validated at the schematic level through an extensive simulation process.

## 8.1 Automatic Test Laboratory

The Automatic Test Laboratory (ATLab) is a joint-lab established in collaboration with Infineon Technologies, designed and realised as an industrial-standard facility fully dedicated to research and development in the field of automatic testing. As part of the MUSA project, ATLab has been established starting from the selection of the proper room within the university buildings, followed by an extensive market research to identify the most appropriate pieces of equipment. The setup phase included the installation and configuration of all the machinery.

Currently, ATLab is fully operational and actively employed in the development of industrial test solutions and innovative test concepts in collaboration with the industrial partner.

Figure 8.1: Automatic Test Laboratory (ATLab) panoramic view.

The collection of testing equipment present within ATLab, shown in a panoramic view in Fig.8.1, is currently comprehensive of:

- Teradyne ETS-88 Automatic Test Equipment in a custom configuration, associated with a Talon5 workstation
- Cascade Tesla200 semi-automatic wafer prober, fitted with a custom ETS-88 interface for wafer-level testing
- Mechanical Devices MAXTCpp thermal control device, allowing accurate temperature regulation for package-level testing
- Tektronix MSO66B 6-channel 1GHz oscilloscope, allowing proper test hardware characterisation and troubleshooting

The presented instrumentation, in combination with a complete set of standard laboratory equipment, provides a comprehensive environment for the full development of test solutions for power management devices, and means to conduct research in all the major branches of automatic test devleopmet.

## 8.2 IonoTRACK ADC

Ionoacoustics aims to investigate and characterise several quantities of the thermoacoustic signal generated after the deposition of the energy of a particle beam into an absorber material.

Ionoacoustic detectors present several advantages with respect to traditional nuclear imaging devices. Among them, they achieve sub-$mm$ precision and sub-$ms$ delay with an intrinsically simpler detector architecture based on acoustic sensors. They are therefore ideal for innovative radiotherapy techniques, where nuclear imaging falls short in all cited aspects.

State-of-the-art ionoacoustic detectors are limited by off-the-shelf components, whose non-optimal performance preclude proper application of ionoacoustic techniques in clinical scenarios, characterised by signal amplitudes as low as tens of $mPa$. It is therefore necessary to design specialised electronics to work in these experimental conditions.

In this part of the work[1], a SAR ADC for new generation ionoacoustic detectors is presented and validated at the schematic level. The ADC target specifications are defined based on the clinical scenarios and on the requirements of the comprehensive acquisition chain, and all of the detector building blocks are consequently optimised.

The proposed ADC, shown in Fig.8.2, achieves an Equivalent Number Of Bits (ENOB) of 7.69 and a total power consumption of $160.98\mu W$ in nominal conditions on the full frequency range. The ADC is also adequately robust to Process-Voltage-Temperature (PVT) and mismatch variations to properly suit the proton sound detector application.

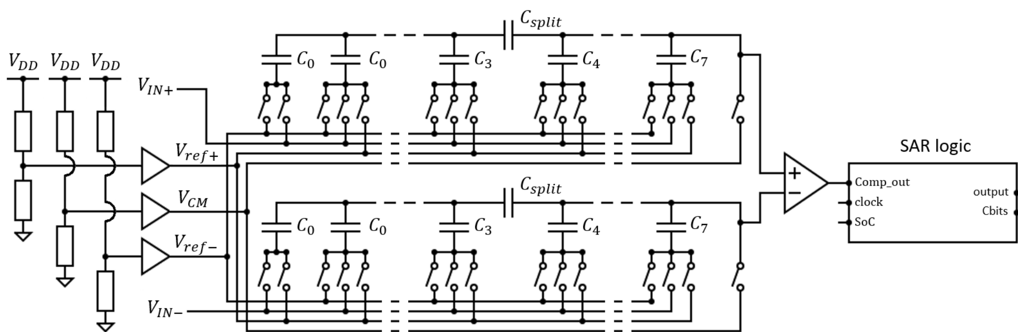



Figure 8.2: IonoTRACK SAR ADC top-level schematic

[1]Turossi, Davide, et al. "A 7.69 ENOB, $161\mu W$ SAR ADC in 28nm CMOS for Proton Sound Detectors." 2024 19th Conference on Ph.D Research in Microelectronics and Electronics (PRIME). IEEE, 2024.

# Acknowledgement

The SAR ADC design activity has been partially supported by the iono-TRACK project, funded by the Italian Institute for Nuclear Physics (INFN) CSN5, and by the Brain28 project, funded by the Italian Ministry for Universities and Research (MUR).

The Department of Physics of the University of Milan-Bicocca is gratefully acknowledged for supporting my overall Ph.D. activity.

Special gratitude is personally expressed to the Infineon Technologies Italia (Pavia) test development team and the Infineon Technologies Austria (Villach) test development team for offering warm hospitality during my Ph.D. activity and sharing valuable knowledge that supported my research and professional growth.

# Bibliography

[1] J. Sawal, EmergenResearch, Production Testing Market, https://www.emergenresearch.com/industry-report/production-testing-market (accessed 24/01/2026)

[2] PW Consulting, Worldwide Semiconductor Test Services Market Research Report 2026, Forecast to 2032. https://pmarketresearch.com/worldwide-semiconductor-test-services-market-research-2024-by-type-application-participants-and-countries-forecast-to-2030/ (accessed 28/01/2026)

[3] Keysight Technologies, Reduce Cost of Test, https://www.keysight.com/zz/en/assets/7018-04600/application-notes/5992-0195.pdf (accessed 22/01/2026)

[4] IEEE EPS, 2024 Heterogeneous Integration Roadmap - Chapter 17: Test Technology, https://eps.ieee.org/images/files/HIR_2024/HIR_2024_ch17_Test_Technology.pdf (accessed 24/01/2026)

[5] Vermeulen, Bart, et al. "Trends in testing integrated circuits." 2004 International Conferce on Test. IEEE, 2004.

[6] Ali, Liakot, et al. "Challenges and directions for testing IC." Integration 37.1 (2004): 17-28.

[7] S.Sapre, Analog Devices, Isolated Gate Drivers-What, Why, and How?, https://www.analog.com/en/resources/analog-dialogue/articles/isolated-gate-drivers-what-why-and-how.html (accessed 25/01/2026)

[8] IEEE, Standard for Datasheet Parameters and Tests for Integrated Gate Drivers, https://standards.ieee.org/ieee/2964/10393/ (accessed 26/01/2026)

[9] Lee, S. Daniel, and Tom Middleton. "Behavioral Simulation of VLSI Test System Aids Debugging and Analysis of Test Programs." Proceedings of the International Test Conference, IEEE, 1984, pp. 614–620.

[10] Krampl, Gunter, Marco Rona, and Hermann Tauber. "Test Setup Simulation: A High-Performance VHDL-Based Virtual Test Solution Meeting Industrial Requirements." Proceedings of the International Test Conference, IEEE, 2002, pp. 870–878.

[11] Webster, Bruce A. "An Integrated Analog Test Simulation Environment." Proceedings of the International Test Conference, IEEE, 1989.

[12] Austin, T. "Creating a Mixed-Signal Simulation Capability for Concurrent IC Design and Test Program Development." Proceedings of the International Test Conference, IEEE, 1993, pp. 125–132.

[13] Pointner, Sebastian, et al. "Test Your Test Programs Pre-Silicon: A Virtual Test Methodology for Industrial Design Flows." 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). IEEE, 2019.

[14] Zivkovic, Vladimir A., et al. "AMS Test Vector Generation using AMS Verification and IEEE P1687. 2." 2022 IEEE European Test Symposium (ETS). IEEE, 2022.

[15] AMS-VT Analog/Mixed-Signal Virtual Tester, `https://www.testinsight.com/ams-vt-mixed-signal-virtual-ate` (accessed 26/01/2026).

[16] Lu, Ping, et al. "Bridge the gap between simulation and test: An OSA-compliant Virtual Test Environment." 2009 IEEE AUTOTESTCON. IEEE, 2009.

[17] van de Logt, Leon MA, Vladimir A. Zivkovic, and Ingrid HA van Baast. "Model-driven AMS test setup validation tool prepared for IEEE P1687. 2." 2019 IEEE European Test Symposium (ETS). IEEE, 2019.

[18] Sunter, Stephen, Vladimir Zivkovic, and Bartlomiej Praselski. "A method for simulating mixed-signal ATE tests." 2024 IEEE 42nd VLSI Test Symposium (VTS). IEEE, 2024.

[19] Aderholz, E., et al. "Virtual Test Development Using Pre-Silicon Verification Environment." 2024 IEEE International Test Conference (ITC). IEEE, 2024.

[20] Singh, Gursimran, et al. "An Integrated Environment for Simulation of Mixed Signal Tests Using Open-Source Tools." 2025 IEEE 9th International Test Conference India (ITC India). IEEE, 2025.

[21] D. Turossi and A. Baschirotto, "A SystemC-AMS Development Framework for High Power IC Test-Hardware", 2024 IEEE European Test Symposium (ETS), The Hague, Netherlands, 2024.

[22] D. Turossi and A. Baschirotto, "Versatile Development of Test-Hardware in a SystemC-AMS Virtual Testing Framework", 2025 IEEE International Conference on Integrated Circuit Design and Technology (ICICDT), Lecce, Italy, 2025.

[23] Standard SystemC-AMS extensions 2.0 Language Reference Manual.

[24] Barnasconi, Martin. "SystemC AMS extensions: Solving the need for speed." DAC Knowledge center 6 (2010).

[25] Fey, Dietmar, Lukas Riedersberger, and Marc Reichenbach. "Simulating memristive networks in systemc-ams." Memristor and Memristive Neural Networks (2018): 147.

[26] Rizzi, Tommaso, et al. "Comparative analysis and optimization of the SystemC-AMS analog simulation efficiency of resistive crossbar arrays." 2021 XXXVI Conference on Design of Circuits and Integrated Systems (DCIS). IEEE, 2021.

[27] Coşkun, Kemal Çağlar, Muhammad Hassan, and Rolf Drechsler. "Equivalence Checking of System-Level and SPICE-Level Models of Static Nonlinear Circuits." 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2023.

[28] Coşkun, Kemal Çağlar, Muhammad Hassan, and Rolf Drechsler. "Equivalence Checking of System-Level and SPICE-Level Models of Linear Circuits." Chips 1.1 (2022): 54-71.

[29] Vinco, Sara, Michele Lora, and Mark Zwolinski. "Conservative behavioural modelling in SystemC-AMS." 2015 Forum on Specification and Design Languages (FDL). IEEE, 2015.

[30] Ma, Kezheng, et al. "A precise and high speed charge-pump PLL model based on systemC/systemC-AMS." International Journal of Electronics and Telecommunications 58 (2012): 225-232.

[31] Hassan, Muhammad, Daniel Große, and Rolf Drechsler. "System-level verification of linear and non-linear behaviors of RF amplifiers using metamorphic relations." Proceedings of the 26th Asia and South Pacific Design Automation Conference. 2021.

[32] Caluwaerts, Ken, and Dimitri Galayko. "SystemC-AMS modeling of an electromechanical harvester of vibration energy." 2008 Forum on Specification, Verification and Design Languages. IEEE, 2008.

[33] A SystemC AMS Framework for the Design and Simulation of Energy Management in Electric Vehicles" (2019) — Chen et al.

[34] Markwirth, Thomas, Paul Ehrlich, and Dominik Matter. "Dynamic fault injection library approach for SystemC AMS." (2016).

[35] COSIDE Overveiw, `https://www.coseda-tech.com/coside-overview` (accessed 25/01/2026)

[36] Pêcheux, François, et al. "SystemC AMS based frameworks for virtual prototyping of heterogeneous systems." 2018 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2018.

[37] Bhadani, Rahul, et al. "Modeling and real-time simulation of microgrid components using systemc-ams." 2023 Winter Simulation Conference (WSC). IEEE, 2023.

[38] EV-MST™ Basic Programming ETS-88, MAN0152 Rev 1.5, Eagle Test Systems, 2009

[39] EiceDRIVER™ 1ED31xxMC12H Compact Datasheet, `https://www.infineon.com/row/public/documents/24/49/infineon-1ed31xxmc12h-1ed-x3-compact-datasheet-en.pdf` (accessed 24/01/2026)

[40] E. Lee, er al., Analog Devices, PCB Layout Design Guide for Analog Applications, `https://www.nxp.com/docs/en/application-note/AN3962.pdf` (accessed 24/01/2026)

Tesi di dottorato realizzata nell'ambito del progetto MUSA finanziato dal PNRR Missione 4 Componente 2 Investimento 1.5, finanziato dall'Unione Europea - NextGenerationEU - CUP H43C22000550001