



AUTOMATED HARDWARE VERIFICATION SYSTEMS

By José Pérez
R&D Senior Hardware Engineer *SALICRU*

1.- INTRODUCTION

One of the most important, if not the most important, goals of a new product is to be the best on the market, especially as far as the quality and features it offers are concerned. A good standard of quality is not only achieved by the R&D department creating a good design but also by the product being correctly manufactured to avoid defects occurring that could affect its functioning and therefore its quality. Because of this, continuous control of production is necessary to ensure that each and every one of the steps involved in executing the product is carried out properly.

This responsibility rests with the production department. But it is a task that is difficult to manage because comprehensive control requires considerable time and resources, both of which are difficult to achieve because of today's associated economic costs and the fact that they have a direct impact on the final price of the product. It is quite pointless having the best product if no one can access it or if the competition can simply offer one similar with a more affordable price.

The main goal of the production department, therefore, is to create a reliable manufacturing process to achieve the highest quality in the fastest possible time in order to reduce costs. And that is the reason for writing this document which will look at ways of achieving higher quality electronic cards and performing complete functional tests in order to considerably reduce time and minimise expenditure.

Different solutions exist to achieve the goals described in the preceding paragraph involving the use of companies that specialise in electronic assembly which not only assemble the cards but also offer the possibility of verifying them functionally and in an automated manner. But at what price? What cost should be charged to each product to recover the initial expenditure?

Obviously, if there is a high volume of products, the cost to charge would not be significant, but if the volume is not high, this cost could be quite high, making the expenditure impossible to justify.

This is the situation that many companies or production departments find themselves in: they cannot invest heavily in automated production systems to enable them to achieve greater product quality and competitiveness, but they are required to do so in order to not be left out of the market.

This usually occurs in companies that operate with medium-sized production volumes, not in those with large volumes which, as previously explained, can spend more on verification systems or in companies where production volumes are very low but with value-added costs that are so high that the cost of thorough manual verification does not represent a significant increase in the final price of the product.

To solve this problem, the idea arises to create one's own automated electronic card verification system. And although the system would initially be designed for verifying electronic cards, it could also be applied to finished electronics products, those consisting of different electronic cards, wiring, etc. Because of this, as indicated by the title of this document, the system is for verifying hardware in an automated manner.

2.- BENEFITS OF AUTOMATED VERIFICATION

The application of an automated hardware verification system provides the following improvements:

- **Reduced verification time:** The change from manual to automated verification inherently reduces verification time.

- **Higher test quality:** The manual instruments used to check electrical signals, such as oscilloscopes and multimeters, although very accurate, produce readings that have to be finally approved by the operator. With the automated test, measurements are taken by acquisition cards and approval is carried out by software in which admissible measurement tolerances, the number of measurements to be executed and statistical formulas are entered to analyse the measurements captured before a decision is given.

This all results in improved measurement quality during verification.

In addition, due to the speed of microcontrollers and computer systems, the number of verification points can be significantly increased, thereby slightly improving verification time.

- **Greater traceability:** All tests on the electronic card are recorded in the computer. This means that each card can be assigned a serial number that it will share with the saved file. It also enables the report document to be formatted in terms of internal structure and type (e.g. txt, pdf, log, etc.).

- **System migration:** Once some knowledge of the system has been gained, without any great effort, similar solutions can be applied to all kinds of products, whether they be other types of individual electronic cards or finished products with different cards, since the system sees the finished product as a black box to verify where there are different inputs and outputs regardless of whether it is a single card or a set.

3.- ELEMENTS FOR THE IMPLEMENTATION OF AN AUTOMATED VERIFICATION SYSTEM

To implement a custom verification system, the following material is required:

- **Electronic card with microcontroller:** This card connects the computer's test software to the electronic card to be verified. The test software takes control of the microcontroller's peripherals to enable the loaded sequence to be executed, meaning that the microcontroller becomes the slave and executor of the commands received from the computer.

The peripherals can enable and disable actuators, acquire analogue and digital signals, use timers and counters, etc. Using all of these actions, it is possible, for example, to measure the amplitude and duty cycle of a transistor's excitation signal.

- **Software architecture (ESSA):** Embedded system software architecture (ESSA) enables easy application of the software on any type of system. This is because of the layer structure with which it is constructed. Fig. 1 shows an illustrative example of an ESSA.

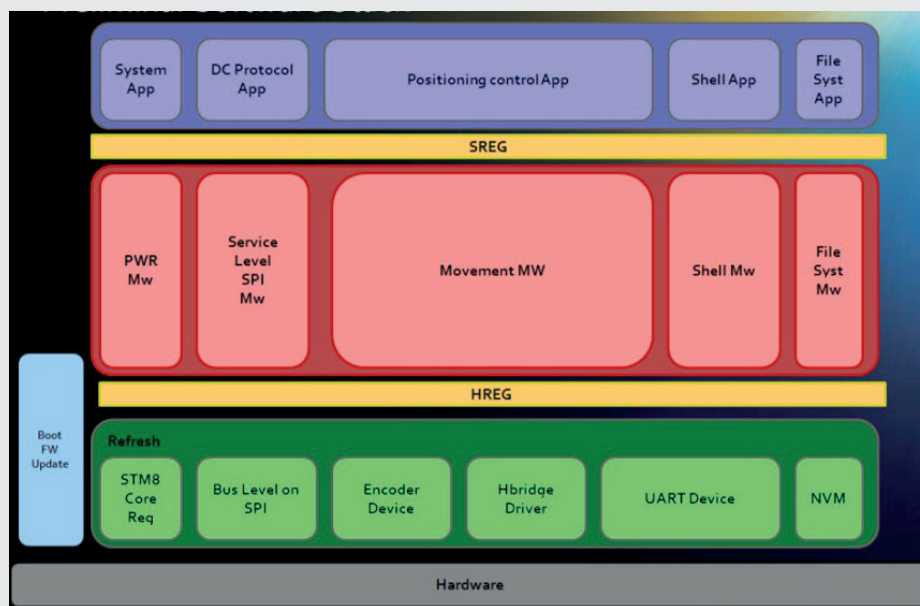


Fig. 1 Example of layers of an ESSA

Broadly speaking, the structure can be divided into three main layers: application layer, middleware layer and driver layer; and two secondary layers: SREG layer and HREG layer.

Application layer: This layer contains high-level modules that have control of the processes. Communication between the application modules and middleware modules is made through the SREG layer.

The SREG layer provides access to the different communication registers between the application modules and the middleware modules. This is one of the key points in the abstraction of software from hardware.

Thanks to the accessibility of the SREG layer, it is possible to change the registers, for example through a test sequencer, to establish, without the need for hardware, whether the application and middleware layers function properly when the register values are changed.

Middleware layer: This layer contains the modules for actions that are performed regularly. Communication with applications, as already explained above, is performed through the SREG layer and communication with the driver layer is performed through the HREG layer. Communication between different middleware modules is not permitted directly without passing through the layers.

The HREG layer, like the SREG layer, enables access to the different registers involved in the communication process between application modules and middleware modules. Like the SREG layer, this layer is one of the key points in the abstraction of software from hardware, and can simulate, without the need for hardware, the functioning of the middleware modules.

In the case of automated hardware verification, as described later, access to the HREG layer is the most important point in the system, as it is the layer which controls the microcontroller's drivers to enable it to function as a slave of the test sequencer.

Driver layer: The driver layer depends on the hardware where the ESSA is installed. In our system, it is the microcontroller. This layer is the only layer of the ESSA that cannot be transferred to any other system because it depends directly on the hardware. For example, if you have two completely identical electronic cards, except the microcontroller, this part of the ESSA software would be the only one that could not be transferred directly from one microcontroller to another since the functioning of the drivers in each microcontroller is different, but everything else could be transferred directly.

SREG and HREG abstraction layers: As noted above, the SREG and HREG abstraction layers are two of the most important points of an ESSA. These layers enable interaction with the application, middleware and driver modules.

In the case of automated verification, the driver layer and the HREG abstraction layer are mainly used. This is through the registers of this layer which manages the drivers. So, knowing the control registers of the drivers, a test pattern is executed in a test sequencer and the microcontroller becomes a slave of the sequencer, regardless of the functionality of its own firmware.

· SHELL communication interface: The SHELL is the communications port that enables external interaction with the ESSA. For automated verification, it is the port through which the microcontroller and test sequencer communicate.

· Test sequencer: The test sequencer executes shell commands in the hardware layer to verify proper functioning.

To perform the test, applications can be used which are designed to execute test sequences. Software exists that is open source, meaning that no licences need to be purchased. Open-source software also allows tests to be customised, for example, by adding one's own libraries which closely match the requirements of the product.

Many of these applications create reports once the test has been completed, which show the result on widely used formats (XML, HTML, XLS, etc.). This is very important for ensuring the verified product's traceability and continuous improvement.

Fig. 2 shows two examples of reports from a test sequence execution application. In the first image, the test result is positive and, in the second, it is negative.

Auto Test Test Report (Generated: 20121223 14:41:48 GMT +08:00, 29 seconds ago)

Summary Information

Status: 1 critical test failed
 Start Time: 20121223 14:41:40.197
 End Time: 20121223 14:41:48.661
 Elapsed Time: 00:00:08.464
 Log File: log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	P
Critical Tests	2	1	1	00:00:08	50%
All Tests	2	1	1	00:00:08	50%

Statistics by Tag

Tag	Total	Pass	Fail	Elapsed	P
TT case P1	1	0	1	00:00:00	0%
basic data structure	1	1	0	00:00:08	100%

Statistics by Suite

Suite	Total	Pass	Fail	Elapsed	P
Auto Test	2	1	1	00:00:08	50%
Auto Test: Rbtree	1	1	0	00:00:08	100%
Auto Test: Rbtree: Test Rbtree	1	1	0	00:00:08	100%
Auto Test: Test	1	0	1	00:00:00	0%
Auto Test: Test: Test Hello	1	0	1	00:00:00	0%
Auto Test: Test: Test Hello: Test Hello	1	0	1	00:00:00	0%

Test Details

Type: Critical Tests All Tests

Acceptance Test Report (Generated: 20130915 18:22:50 GMT +11:00, 13 minutes, 36 seconds ago)

Summary Information

Status: All tests passed
 Start Time: 20130915 18:22:26.566
 End Time: 20130915 18:22:54.630
 Elapsed Time: 00:00:28.044
 Log File: log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:28	100%
All Tests	1	1	0	00:00:28	100%

Statistics by Tag

Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags	0	0	0	00:00:00	0%

Statistics by Suite

Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Acceptance	1	1	0	00:00:28	100%
Acceptance: Basic Test	1	1	0	00:00:27	100%

Test Details

Type: Critical Tests All Tests

Fig. 2 Test reports performed with open source software

Adapted hardware or signal adapter cards: To improve test quality, the number of points to be verified needs to be increased. This is why, when starting the design of an electronic card with the intention that it will be verified automatically, it is necessary to include circuitry specifically dedicated to automated verification, that is to say, circuits that are not part of the core functionality of the card, but are only used at the time of verification. This increased product cost is negligible compared to its contribution.

In addition, when it is not possible to incorporate these verification circuits, signal conditioning cards can be designed. These cards are the interface between the microcontroller and the card to be verified. Their purpose is to adapt signals bidirectionally, for example, adapting an analogue signal from the card being verified to the analogue input of the microcontroller or elevating a digital microcontroller output signal to excite a relay. It is important for these acquisition and control cards to be designed with great configurability so that they can be adapted as much as possible to the different electronic cards that require automated verification.

4.- AUTOMATED VERIFIER STRUCTURES

Two types of automated verifier can be defined depending on the structure used to perform the verification test. These structures are:

- Cards that include microcontrollers (auto-verification)
- Cards that do not include microcontrollers (PLC structure)

· **Cards that include microcontrollers (auto-verification structure):** This structure is the simplest to implement as it requires very little additional circuitry to implement the system. This is because if the electronic card requiring verification has a microcontroller, it already serves as a slave to the test sequencer and, with very little external auxiliary circuitry, it is possible to perform auto-verification.

However, and this is the major strength of this system, if the design of the product's firmware is based on an ESSA, there is no need to reprogram the microcontroller with special software to perform the auto-verification test because the ESSA, as explained above, enables abstraction of the microcontroller's hardware and disconnection of the upper layers or tasks, freeing up the external control of the hardware. Thus, with the abstraction, the test sequencer uses the microcontroller as a slave and executor of the necessary actions to carry out the auto-verification process.

Another advantage is that products usually consist of a set of electronic cards and one of them is usually the control, which incorporates a microcontroller, and a set of additional cards such as drivers, screens, interfaces, etc. Almost all additional cards are connected directly to the control card, so, by taking advantage of these connections, it is also possible to perform automated verification of any electronic card that complements the control card.

Fig. 3 shows a possible structure for checking different electronic cards of the same product (2, 3) through one of the electronic cards of the same product mounted to a microcontroller (1) with an ESSA software structure.

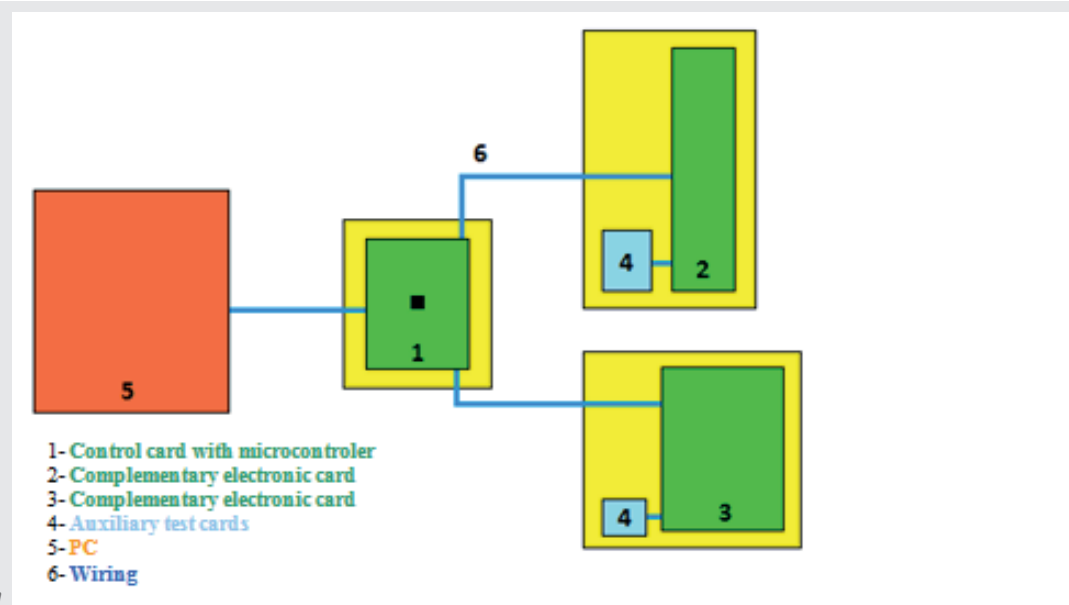


Fig. 3 Wiring diagram

· **Cards that do not include microcontrollers (PLC structure):** This structure is more complex than the previous one, but is more powerful and versatile. Broadly speaking, it involves creating a simplified PLC structure consisting of a CPU (Central Processing Unit), input signal conditioning modules and output signal conditioning modules.

The test sequencer sends commands to the CPU module, which, through the input and output modules, executes the card verification process.

Fig. 4 shows an example structure for checking the different electronic cards of a product (2, 3) through a PLC-type structure (1, 4).

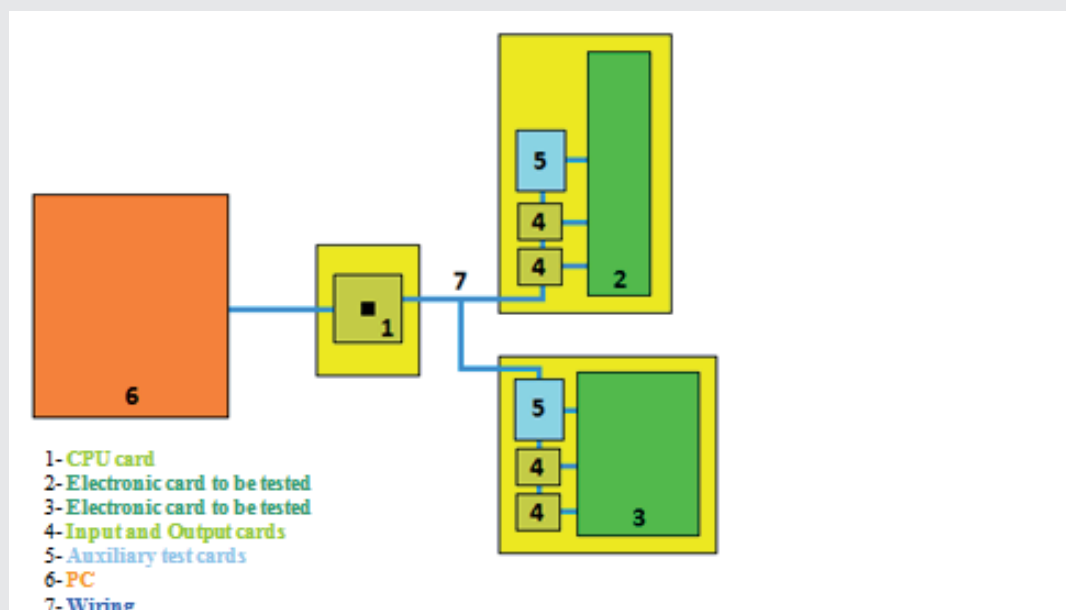


Fig. 4 Wiring diagram