

P1581 Status and Technical update

Frans de Jong, Leon van de Logt
Philips Research
Prof.Holstlaan 4,
5656AA Eindhoven
The Netherlands

Frans.de.jong@philips.com, leon.van.de.logt@philips.com

ABSTRACT

Since 2001 the IEEE p1581 working group aims to provide a test standard for memories that cannot have boundary scan. Currently this p1581 activity focuses on test access and test circuitry. The latter came to a resolution and will be explained. Economic advantages for the user of p1581 devices and further work will be discussed.

INTRODUCTION

In the nineties, more digital parts were used in consumer designs. Typically discrete Flash and SDRAM devices were used more often. With growing complexity and speed of these devices, a need for easy testability during assembly became apparent.

This was the reason that a joint effort between Fujitsu and Philips was started that resulted in a test solution called 'Static Component Interconnection Test Technology' (SCITT) [1][2]. This result was implemented in an SDRAM. The solution covered two major questions:

1. How to get into the test mode without the need for additional pins?
2. What testability circuit will give the best results with respect to detection and diagnostics of the board level defects?

The first question was solved using free coding space in the state-machine of the SDRAM directly after power up. The drawback is that the device must be powered down (and therefore the whole PCB) before the test mode can be re-entered.

The second question was solved using a dedicated combinatorial circuit of exclusive-or (XOR) gates. From mathematical studies it is proven that full detection and guaranteed diagnosis are possible.

Now, the IEEE p1581 working group works on defining the exact needs for detection and diagnosis. The main reason for this action is that the SCITT circuit is set up for guaranteed diagnostic capabilities that are not always required. Another reason is the apparent resistance against SCITT being patented. The following chapter explains the wider range of applicable circuits possible for the p1581 standard.

Another aspect that is worked upon is the search for the proper test entry and test exit method. The power down scheme of SCITT was the only possible scheme at that time, but not an optimal one for the industry. To power-down a PCB during test in a production line is very inefficient. As a solution, internal registers of the device can be used for activating the test mode. However, this requires the use of the address and data bus, which are the actual subjects to be tested. We will elaborate more on test access later in this paper.

The last part of this paper describes an economic model for users of p1581 compliant devices. This model shows the costs-benefit analysis based on a variety of production settings: from single prototypes to a general volume production line.

FROM DETECTION TO DIAGNOSIS

This chapter describes the test circuit that will serve as basis for the IEEE p1581 standard draft standard. The test circuit will always guarantee the detection of the board level defects. Next to this, depending on which implementation is chosen the degree of guaranteed diagnostics may vary. Rules (circuit rules), required for this variable concept are given together with requirements for handling remaining I/O pins on the memory.

For a correct description of the method three situations need to be identified regarding the number of input and output pins:

- 1) $I = O$
- 2) $I < O$
- 3) $I > O$

Here "I" stands for Input pin of the device and "O" for output pin of the device.

The primary circuit rule for the test circuit will be that it comprises only combinatorial, static logic. This is to be able to deal with the restrictions for easy boundary scan testing.

Case 1: $I=O$

In case $I=O$ the number of inputs is equal to the number of outputs during testing. The test circuit is the so-called full inverter concept, i.e. every input of a device is connected to a single output of the device through an inverting buffer (see figure 1). This situation allows for guaranteed detection of stuck-at faults and bridging faults. The choice for inverters instead of buffers is easy to understand: This will detect input-output shorts, and inverters enhance diagnostics in case of a bridging fault: applying different test signals on two shorted inputs will result in two low signals on their respective outputs, or two high signals, depending on the technology. When the short is on the outputs, the result will be inverted and so diagnosable.

The circuit rules are:

- a) Every input pin is connected to an output pin using an inverter;
- b) No two outputs share the same input (uniqueness in connections);
- c) No two inputs are connected to the same output (uniqueness in connections).

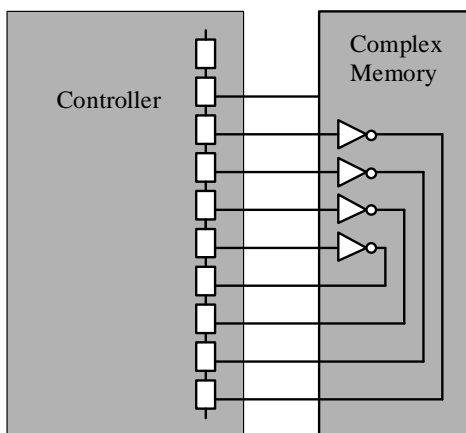


Figure 1:Complex memory in test mode, a simple scheme to test.

Case 2: $I < O$

During test, lesser inputs are available than outputs. For example lets take a typical device with a 16-bit output bus ($O=16$) and a 12-bits X-address bus ($I=12$, X means 'extended', including other inputs), so $I < O$.

Lets start by applying the rules for the case $I=O$ for this device: we will end up with 12 inverters leaving 4 outputs not connected yet. These 4 remaining outputs may be connected using simple gates of which the inputs are connected to the input pins of the device (figure 2).

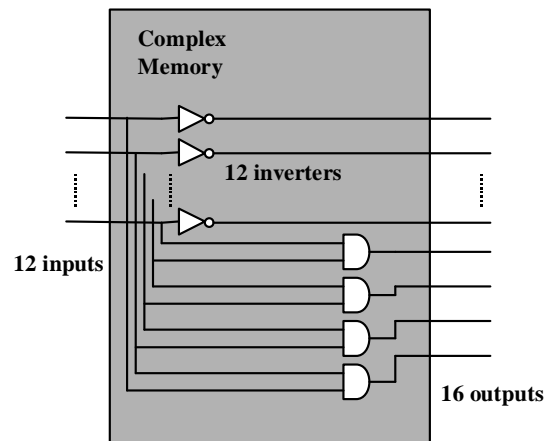


Figure 2: A valid implementation of case 2.

To guarantee full detection on these 4 remaining outputs the circuit rules are:

- a) Use gates with at least two inputs;
- b) The set of input pins used is different for every output.

Here also circuit permissions come in:

- c) The gate may consist of any combination of non-sequential logic;
- d) Every input used in a gate may also be re-used for another gate (implicit for case 2).

As an explanation to permission c): all input-shorts are covered already. So, although an AND-type short can not be detected by using an AND gate, this is not an issue. A simple AND or OR gate will suffice to produce an additional but different output signal.

Case 3: $I > O$

During test there are more inputs than outputs connected to the test circuitry. For example lets take a typical device with 16 bits output bus ($O=16$) and 22 bits X-address bus ($I=22$), so $I > O$.

First apply the rules for the case $I=O$ for this device: we will end up with 16 inverters leaving 6 inputs open.

Obviously, permission c) in case 2 will not hold for this case. The 6 remaining inputs shall be connected using XORs to ensure detection on the inputs. Because no unique outputs are remaining at least one of the initial 16 inverters must be removed. This results in a test circuit consisting of 15 inverters and an XOR circuit with 7 inputs and 1 output (figure 3).

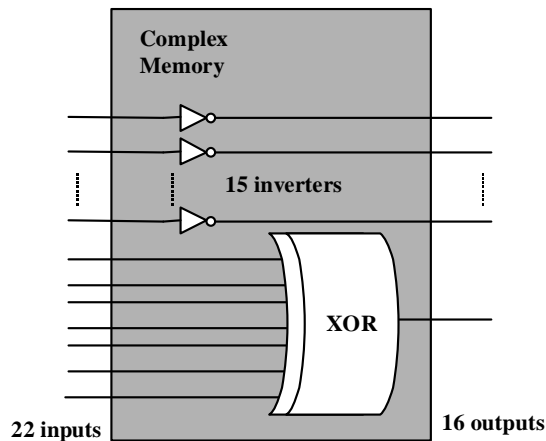


Figure 3: Case I>O, a valid implementation

In general, to obtain full detection on the outputs involved in these 7 (remaining) inputs the circuit rules are:

- Use XOR gates;
- The set of input pins is unique per output (so per XOR tree);
- Every XOR tree shall be connected with at least two inputs (implicit).

Again also some circuit permissions exist:

- Every input used in a XOR tree may also be re-used for another XOR tree.

Case 3 can be written in formulas:

If I = number of inputs and O = number of outputs, with x = number of XOR outputs (or XOR trees), then for the condition that $I > O$:

The number of inverters = $O - x$

and the number of inputs on the XORs = $I(x)$ where $I(x) = I - (O - x)$.

From a diagnostic point of view x can be regarded as a parameter that influences diagnosis.

If $x = O$ full guaranteed diagnosis can be obtained if the circuit complies to the additional three circuit rules:

- Each input is involved in at least two X(N)ORs and each X(N)OR has at least three inputs.
- Each XOR has an odd number of inputs.

- No two XORs have the same set of inputs.

Note: In this guaranteed full diagnosis setting, no inverters are present.

If $x = 0$, so if no XOR is used at all, the remaining outputs must be connected with additional gates. Obviously, the gates will not only show incomplete detection (for instance the and-short) but also detection and diagnosis become layout dependent.

In practice, normally case 2 or 3 apply.

TEST MODE SWITCHING

When the interconnections to a memory device are to be tested, the device must be set into a test mode. The solution as realized in the SCITT method [1] implies to hold a few pins at fixed levels during power up. As stated earlier this is not a preferred method because normally the PCB is powered before a (boundary scan) tester takes any action. Currently, activities are going on to find the best method to activate and stop the p1581 test mode. The most obvious, but hard to realize method is adding a dedicated test pin on the device. Another suggestion is to have the test mode controlled through a bit in a control register. However, accessing the register implies that pins are used that still need to be tested. For later (software based) service checking in a running system this could be a useful addition. Alternatively, especially for flash devices, a 'multi function' reset pin was suggested. The following paragraph discusses consequences of this option.

The IEEE 1149.1 standard states that the functional reset must be kept separated from the JTAG reset. However, there may also exist the situation where Boundary Scan is not present or just not used. Using the reset pin for test mode introduces test problems as we have encountered for a typical Flash. The situation can be divided in four cases.

Case A: Test mode controlled by boundary scan with separated reset for Flash.

See figure 4 for the board connections.

Generally a power-on-network (PON) is used on the PCB to generate the reset signal. While resetting the controller by keeping test point TP low, a 0-1-0 transition is driven via Boundary

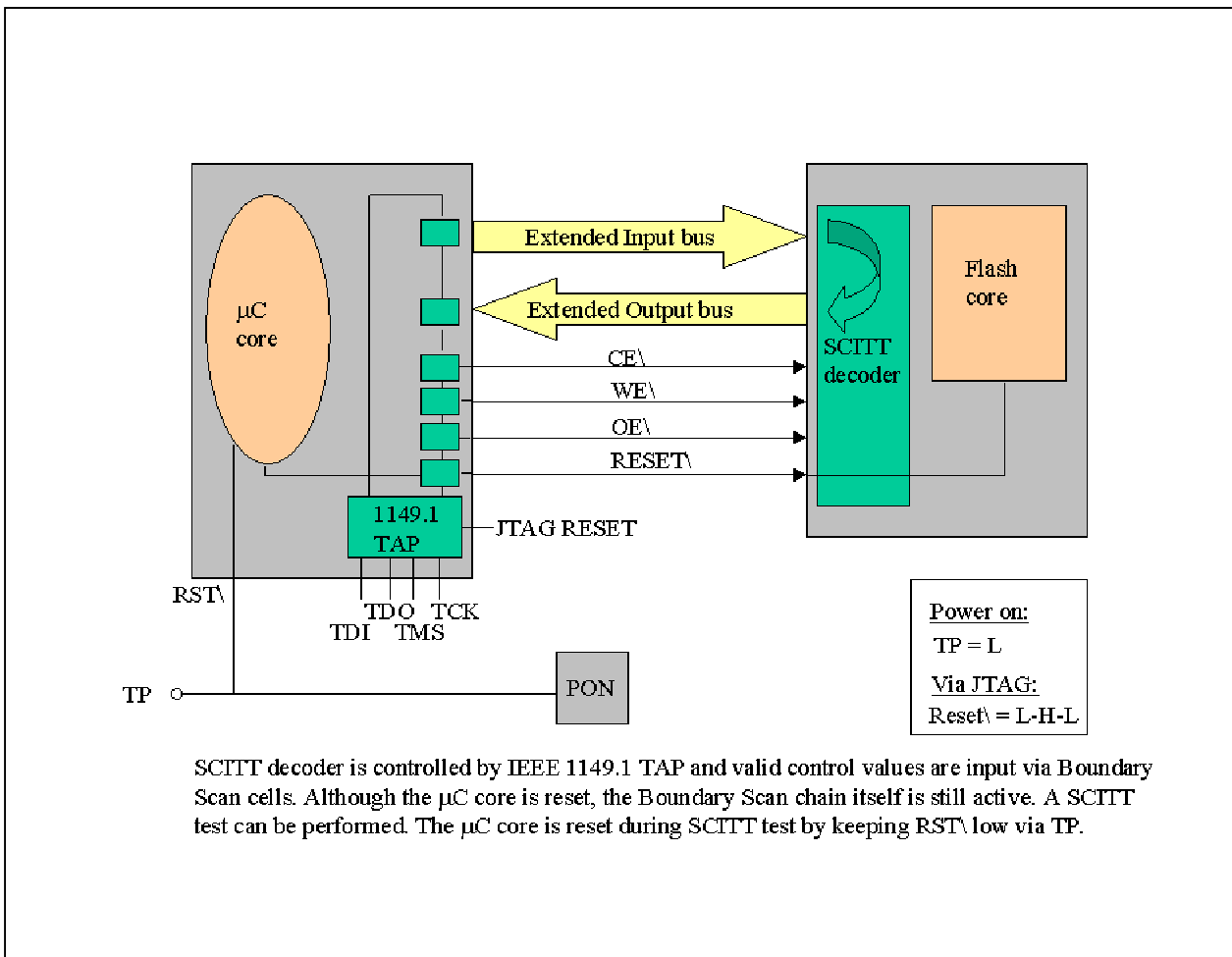


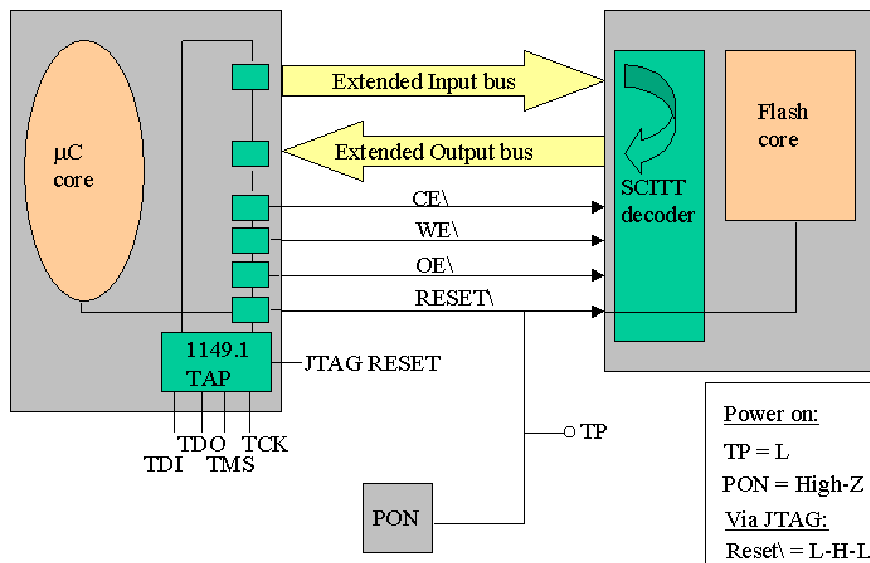
Figure 4: Boundary scan controlled test mode signal via reset.

Scan. The memory can still be brought into test mode. Using a reset pin in this design is no problem. This implementation introduces a small design-for-test step in the design by implementing a TP or switch on $\text{RST}\backslash$.

Case B: Test mode entry controlled by Boundary Scan via a common reset for Flash and μC . Here, the PON signal will reset the μC core and the Flash by a low-high transition. After this reset, the Boundary Scan chain is essentially able to output the entry commands for test mode to test the Flash (see figure 5). However, the Flash received a reset pulse without the corresponding test mode entry settings for CE and WE. Now, the device has entered the functional mode and test mode entry is not possible anymore (although this depends on the implementation). Alternatively, the PON activity can be suppressed. TP is kept low during power-on to prevent the controller from driving values on the $\text{RESET}\backslash$ pin.

When the BS-chain is active the μC core is disconnected from the external pins. Test mode entry commands via the chain are possible. The PON circuit must be protected against external driven values (high_Z output). Test mode entry is possible provided that an I/O Boundary Scan cell is implemented on the reset pin of the μC . Some μC don't have such a cell on the reset pin! Case B makes the design and the board more complex and in some cases test mode entry is even not possible.

Case C: Test mode entry controlled by a non Boundary Scan controller with separated reset for Flash. If a controller without Boundary Scan is used, the situation becomes problematic. In the μC a piece of assembler code taking care of test mode entry must be developed. Via a valid value on an external pin I (see figure 6) the controller recognizes that test mode must be activated. In this case the Flash RESET pin is only connected to the μC and the $\text{RST}\backslash$ of the μC is separately connected to the PON. Now, the μC is only reset during

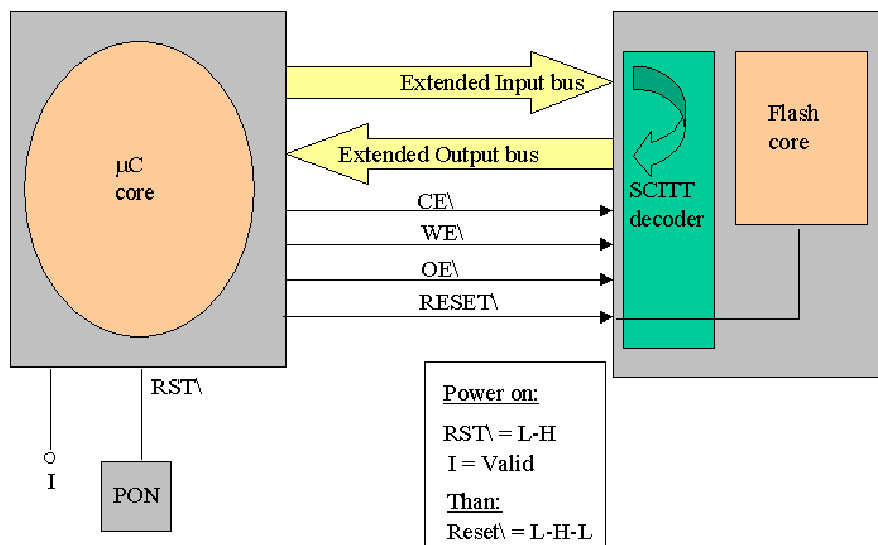


SCITT decoder is controlled by IEEE 1149.1 TAP and valid control values are input via Boundary Scan cells. Although the μC core and the Flash core are reset, the Boundary Scan chain itself is still active. A SCITT test can be performed. The PON must be protected against external driven data.

Note: not every μC has a Boundary Scan controlled I/O RESET pin. This has major impact on SCITT controllability.

Figure 5 (above): Boundary scan controlled test mode signal via reset\ which is also the system reset.

Figure 6 (below): Non-boundary scan controlled test mode using dedicated reset.



A different board configuration for the RESET. In this case the RESET for the Flash is controlled by the μC . The μC itself is connected to a power-on-network on the PCB. In this case a 0-1-0 RESET transition for SCITT control can still be performed. The PON will reset the μC core. Thereafter, the μC is functional. For a valid value on pin I the assembler code executes SCITT test.

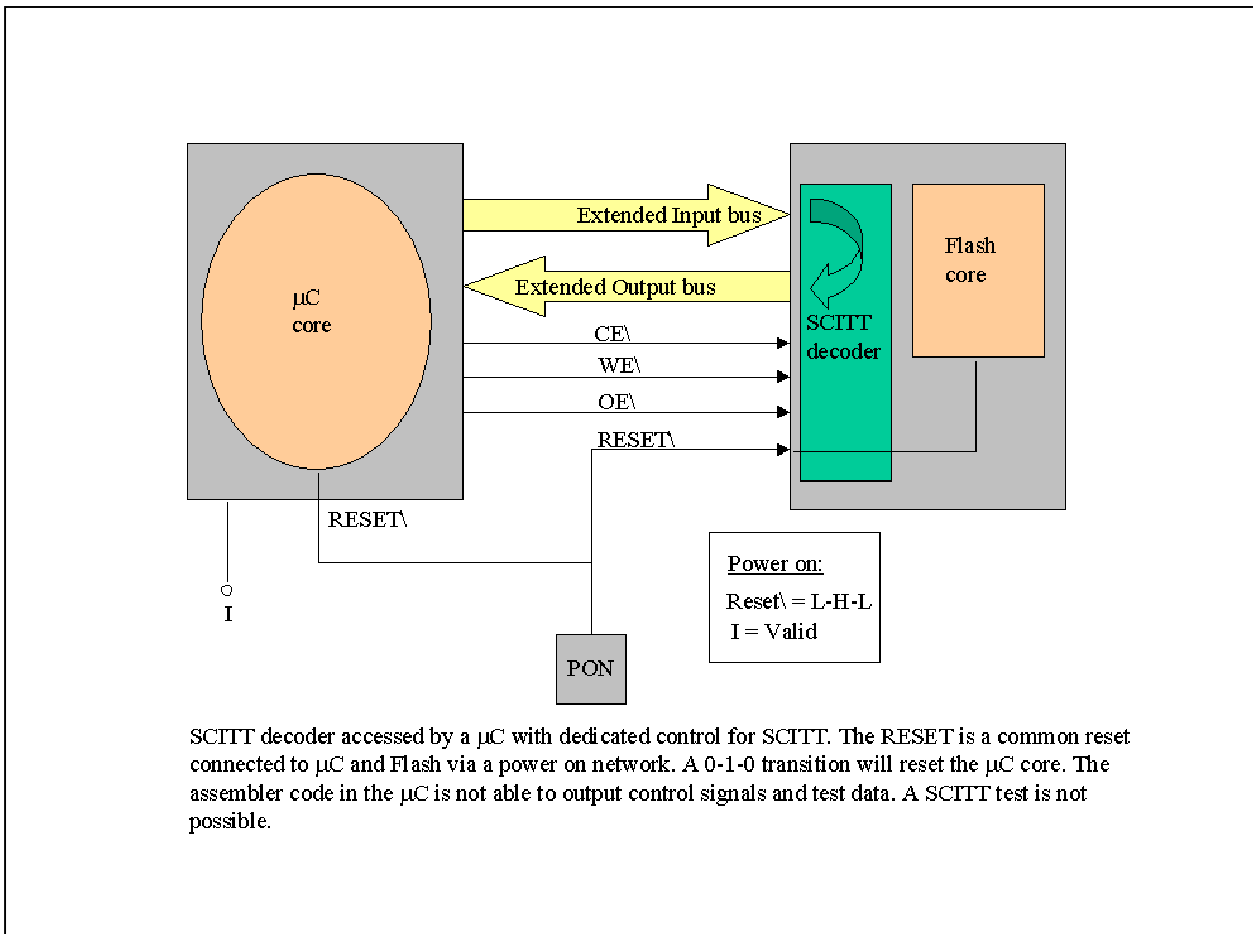


Figure 7: Non-boundary scan controlled test mode via common system reset.

power on. After reset (RST/) the uC is active and test mode entry is possible only if a valid value on pin I is present. During power on, the Flash reset pin must have a defined value in order to enter test mode after uC reset. For this Flash it needs to be low value i.e. by a pull down resistor. Test mode entry is possible with a small design-for-test implementation.

Case D: Test mode entry controlled by non Boundary Scan controller with common reset for Flash and uC. In the situation where the Flash reset is common with the uC reset, test mode entry is not possible (see figure 7). The uC must be reset before it can be active on the PCB. Test mode entry has to be done using dedicated assembly code in the uC in conjunction with a valid value on pin I. But, when the PON is active, the uC and the Flash are reset at the same time. To get the Flash into test mode, the RESET pulse is required but this pulse will at the same time reset the uC. Test mode entry and so test are not possible.

In conclusion:

CASE A: SCITT is possible.

CASE B: SCITT is possible but complicates the total design. For some uC not possible.

CASE C: SCITT is possible with a small design change.

CASE D: SCITT is not possible.

Using the reset pin for test mode of the P1581 device introduces a design dependent implementation. Test mode entry is only possible if the design and uC permits. This reduces the possibilities for P1581 as test method. Especially within an IEEE standard it is necessary to have a standard which is as much as possible design independent. Using reset as test control pin makes P1581 a more complicated and a less appealing test method for the designer.

WHAT IS IN IT FOR ME

In today's time-to-market driven economy, cost and efficiency are key issues for both users and manufacturers of P1581 devices. Working group discussions made clear that development of this

standard can not do without some cost-benefit analysis associated with this standard. A first attempt of such analysis has been made during the first experiments in 1998. Now a more complete analysis is set up for the users - the group that will actually use the P1581 devices. The model presented here is based on numbers from a typical factory setting. Numbers are included for low quantity to high volume production. Generally, a production stage is preceded by a prototyping stage of a few pieces. For a small batch of products (<500 pcs), the prototyping stage costs are not separately calculated. An important factor is, that prototyping is a less efficient stage with higher engineering costs per employee compared to fully optimized production stages. The model incorporates these effects, so with volume production the effects of (high cost) prototyping are added. Four categories are defined (from low to high volume production) each with the main parameters involved in the analysis. The table shows the relation:

Table 1: Main parameters used in the model.

Production (PCs)	Test development time	Test execution time	Repair time of defect PCBs	Savings on tester hardware
<500	Yes	No	Yes	No
10.000	Yes	Yes	Yes	No
250.000	Yes	Yes	Yes	Yes
1 Million	Yes	Yes	Yes	Yes

The parameters in the table will be shortly explained.

- Test development time**
 This parameter compares development time for functional tests of memories to development time with P1581 devices. The model takes the difference between the two.
- Test execution time**
 Test time is the actual time required for test execution per PCB. For this to calculate the Fall-off-rate (FOR) for memories is also used. Those PCBs that are defect (the FOR) will be timed out during testing consuming additional test time in production.
- Repair time**
 Defect statistics for memories (in Parts Per Million (PPM)) determine the number of PCBs to be repaired. The total repair time of all these defect PCBs are taken into account with this parameter.

- Consequences for tester hardware**
 For higher volume productions the use of P1581 could have impact on the total required tester hardware. In case memories can be rapidly tested using standard Boundary Scan tools, another test station in the line could be spared.

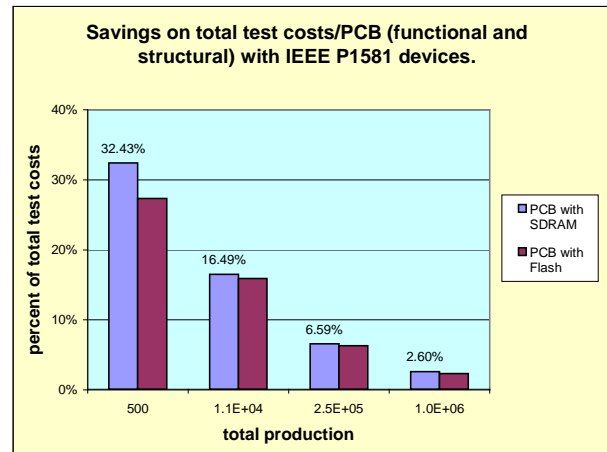


Figure 8: The graphic result of the model.

Taking appropriate numbers for the above mentioned parameters, savings in total test costs per PCB can be calculated. As targets a normal consumer application is used with four SDRAMs and a second application with two Flash devices. Figure 8 shows the results for the two PCBs for low to high volume production. Significant higher savings are received for low volume production and prototyping. Especially, in this production range we often see much higher engineering costs/hour. Besides that, during prototyping the test development time is relatively high per PCB and PCB repair takes much time for allocating the defect when P1581 is not used. P1581 saves time on testing and fault allocation and so it saves expensive engineering hours.

For high volumes, test engineering is must cheaper and the production process is highly optimized. Therefore, savings on test costs per PCB are smaller but still worth while because of the high volume produced. After all, a total saving of 2,6% in test costs per product times a million products is making sense. Savings in tester hardware are difficult to quantify but could well mean the difference between a total saving of 1% and 5%. This depends on the number and load of the equipment used and the depreciation method. This parameter is very interesting for the model because it determines largely the final number.

FINAL REMARKS

We see the following steps to be made from here: The writing of the P1581 draft standard based on the concepts discussed in this paper. This starts with chapters that will be thoroughly commented until the working group finds it sufficiently clear and correct. A second task is to define the preferred test entry method. Finally, another task is the description of the circuit in a language. Already proposed is a plain verilog version [1].

REFERENCES

- [1] De Jong, F. and Raaijmakers, R. Static component interconnection test technology in practice. Proceedings of the IEEE International Test Conference. 1999, p556-565.
- [2] Biewenga, A., de Jong, F., Hollmann, H. D. L., Lousberg, M. Static component interconnect test technology (SCITT) a new technology. Proceedings of the IEEE International Test Conference 1999, p439-448.
- [3] IEEE p1581 website:
[HTTP://grouper.ieee.org/groups/1581/](http://grouper.ieee.org/groups/1581/)