

# Test Infrastructure Design for the Nexperia™ Home Platform PNX8550 System Chip

Sandeep Kumar Goel<sup>1</sup> Kuoshu Chiu<sup>2</sup> Erik Jan Marinissen<sup>1</sup> Toan Nguyen<sup>3</sup> Steven Oostdijk<sup>2</sup>

<sup>1</sup> Philips Research Laboratories

Prof. Holstlaan 4, M/S WAY-41

5656 AA Eindhoven, The Netherlands

<sup>2</sup> Philips Semiconductors

1240 McKay Drive, M/S 11SJ

95131 San Jose, CA, USA

<sup>3</sup> Philips Semiconductors

811 E. Arques Avenue, M/S 80

94088 Sunnyvale, CA, USA

{SandeepKumar.Goel, Erik.Jan.Marinissen}@philips.com {Kuoshu.Chiu, Steven.Oostdijk}@philips.com

Toan.Nguyen@philips.com

## Abstract

*Philips has adopted a modular manufacturing test strategy for its SOC's that are part of the Nexperia™ Home Platform. The on-chip infrastructure that enables modular testing consists of wrappers and Test Access Mechanisms (TAMs). Optimizing that infrastructure minimizes the test application time and helps to fit the test data into the ATE vector memory. This paper presents the test architecture design for the chiplet-based PNX8550, the most complex Nexperia™ SOC designed to date. Significant savings in test time and TAM wires could be obtained with the help of TR-ARCHITECT, an in-house tool for automated design of SOC test architectures.*

## 1 Introduction

Digitalization allows an increasing number of functions to be added to traditional consumer electronics systems such as televisions. Fueled by the advances in semiconductor process technology, these systems are as much as possible integrated onto a single die, in order to fit tight cost and power budgets. The resulting ICs are referred to as *system chips*, or SOC's. To design these 'monster chips' in a timely manner, libraries of pre-designed modules (*cores*) are used, together with application-specific architecture templates, commonly referred to as *platforms*. The Nexperia™ Home Platform is the Philips-internal architecture template for consumer electronics.

The Nexperia™ Home Platform has adopted a modular approach to manufacturing test development. Non-logic modules (such as embedded memories) and black-boxed third-party IP cores demand stand-alone testing. However, even for the remaining logic for which full netlists are available, a modular test approach brings advantages in terms of better

manageable ("divide-n-conquer") ATPG runs and re-use of previous development efforts [1]. These advantages pay off even stronger for a family of chip derivatives, as is the case in a platform. A modular test approach requires an on-chip test access infrastructure, consisting of wrappers and Test Access Mechanisms (TAMs) [2].

Manufacturing test of large SOC's requires a large amount of test data and that test data volume is increasing dramatically over subsequent SOC generations. Modern process technologies suffer from defects which are not adequately detected by stuck-at-only tests, and hence require additional tests. Delay fault testing is an example of such an additional test method; it yields a test data volume several times larger than that of the conventional stuck-at-only tests. Also, the SOC content is growing faster than the SOC access width, i.e., the ratio of transistors per pin is growing. As a consequence, tests require increasingly deeper test vector memory per ATE channel, to store the test stimuli and expected responses [3]. One of the challenges while developing a modular test for an SOC is to design the on-chip test access infrastructure in such a way, that it enables effective scheduling of the various module tests, and fits the total amount of test data onto the given ATE vector memory.

This paper describes the design of the on-chip test access infrastructure for the PNX8550. This SOC is based on the Nexperia™ Home Platform and is the most complex CMOS device designed to date inside Philips. The remainder of this paper is organized as follows. Section 2 outlines the Nexperia™ Home Platform. Section 3 describes the main characteristics of the PNX8550 system chip, its test requirements, and its design-for-test strategy. Section 4 describes our in-house prototype tool TR-ARCHITECT, that was used to advise the test architecture for the PNX8550. Section 5 contains the details of the test architecture as implemented on the chip. Section 6 reports on the test time results. Section 7 concludes this paper.

## 2 Nexperia™ Home Platform

Platforms are architecture templates, that define for a family of ICs in a certain application domain the usage of embedded CPUs, bus architecture, common bus interfaces, etc. The Nexperia™ Home Platform (previously known as Digital Video Platform (DVP) [4]) is Philips' architecture template for the handling of digital video, audio, and interconnectivity in consumer electronics. Figure 1 depicts this platform. It uses one or more 32-bit MIPS CPUs (the PRxxxx series) for control processing and one or more 32-bit TriMedia VLIW processors (the TMxxxx series) for streaming data. Furthermore, the platform includes a flexible range of on-chip modules, such as an MPEG decoder, UART, PIC 2.2 Bus Interface module, etc. To connect the CPUs and other on-chip modules with each other and with the main external memory, a high-speed memory access network and two Device Control and Status (DCS) networks are used. The DCS networks enable each processor to control or observe on-chip the status of modules.

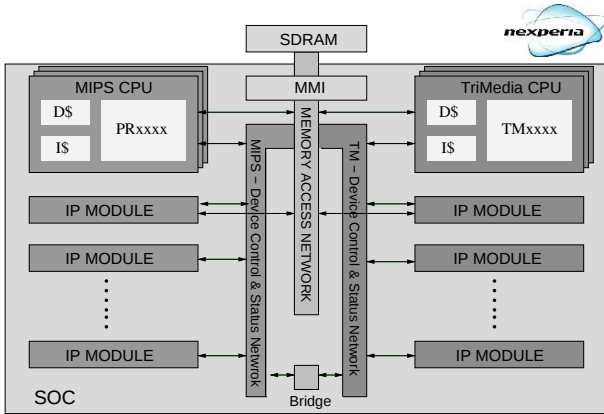


Figure 1: Nexperia™ Home Platform.

In early Home Platform instances, such as the one described in [4], only one MIPS CPU and one TriMedia CPU were used. However, application requirements have evolved over time and now typically a number of these processors are included in a single Home Platform device. This is actually one of the advantages of using a platform. Our Home Platform enables designers to seamlessly attach one or more CPUs, add lower-speed buses for peripheral expansion, and connect on-chip graphics, communication interfaces, or co-processing blocks as needed to address specific market or application requirements. Programmable CPU cores allow easy implementation of new capabilities and standards as they become available, without changing the silicon.

## 3 PNx8550 System Chip

The PNx8550 is the most complex device designed to date in 0.13μm technology inside Philips. It is based on the Nexperia™ Home Platform. It contains about ten million gates. In total, the PNx8550 contains 62 logic IP blocks,

out of which five are hard cores while the rest are soft cores. The five hard cores includes one MIPS RISC CPU and two VLIW TriMedia CPUs. Figure 2 shows its layout and characteristics.



0.13μm CMOS process  
6 metal layers  
1.2 V supply voltage  
PBGA564 package  
100 mm<sup>2</sup> die size  
10M logic gates  
40M logic transistors  
338,859 flip-flops  
62 logic cores  
212 memory cores  
94 clock domains  
140 TestRail wires  
full scan, BIST, and functional testability

Figure 2: PNx8550 chip layout and characteristics.

### 3.1 Chiplet-Based Design

A divide-and-conquer approach was used for the physical design of the PNx8550, in order to reduce the time required to obtain timing closure. The top-level netlist of the SOC was partitioned into manageable-sized blocks, called *chiplets*. A chiplet is a group of modules which are placed together because either they are synchronous to each other or they are not timing critical. The partitioning of the top-level netlist among chiplets followed these guidelines: (1) there should be as few synchronous signal crossings between chiplets as possible, (2) the clock module is placed into a separate chiplet because of its complexity, and (3) cross-chiplet scan chains are not allowed.

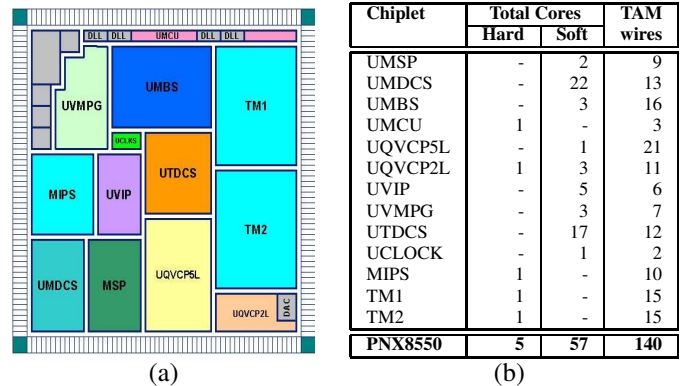


Figure 3: (a) PNx8550 floorplan, (b) distribution of 62 cores and 140 TAM wires over the 13 chiplets.

The PNx8550 design, consisting of 62 logic cores, was partitioned into 13 chiplets. Four out of the five hard cores, i.e., the two TriMedia CPUs, one MIPS CPU, and a Custom Analog Block (CAB) containing the PLLs and the DLLs, were placed into separate chiplets, called TM1, TM2, MIPS, and MCU respectively. The remaining 57 soft

cores and one hard core DAC were placed in the other nine chiplets. Apart from the logic cores, the 13 chiplets also contained a total of 212 memory cores. Figure 3 shows the chiplets in the floorplan of the PNX8550, and the number of cores per chiplet.

### 3.2 Test Requirements

The customer quality expectations for the PNX8550 were 100 ppm (parts per million). In order to obtain this in an environment with bridges, resistive opens, and increased leakage currents, advanced test methods such as gate and path delay testing were required.

From the Nexperia™ Home Platform, the PNX8550 inherited the requirement to have a modular test strategy, in order to allow tests to be re-used wherever possible [4]. A main motivation behind this requirement is the reduction of the creation effort of a full test program for subsequent derivatives in the platform family.

The target ATE for the PNX8550 was an Agilent 93000-P600 test system with 28 M vector memory per channel. Obviously, it was an important requirement to make sure that the test data would fit onto this tester.

### 3.3 Design-for-Test Strategy

The modular test strategy required DfT in the form of an on-chip test access infrastructure, i.e., wrappers and TAMs. The design of this infrastructure and its optimization in order to fit the Agilent test system are described later in this paper.

Full scan design was the default DfT strategy for the logic cores. The scan chains enabled ATPG tools to obtain 99% stuck-at fault coverage for all cores. The MIPS and Tri-Media processors also have BIST and functional tests. For clocks, DDR, DAC, and some speed-critical parts in the design, no scan design was implemented; a set of functional tests were used instead. Most of the small embedded memories were accessed through surrounding scan chains; larger memories were equipped with BIST. Additional built-in burn-in circuits were used in burn-in reliability tests.

More details about the test and debug strategies for the Nexperia™ Home Platform SOC's can be found in [5].

## 4 TR-Architect

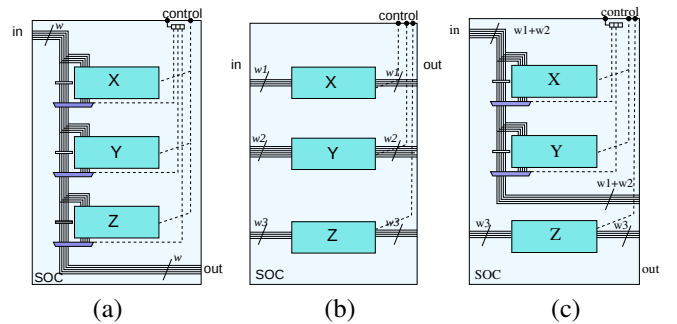
A modular test strategy requires that a module that will be tested as a stand-alone entity is isolated from its environment and equipped with an electrical test access mechanism (TAM) to the chip pins [2]. Isolation of a module is done by designing a wrapper around the module. Philips uses its so-called *TestShell* wrapper [6]; this wrapper is rather similar to the one of the IEEE P1500 SECT standard-under-development [7]. The test access to the module can be

provided by means of one or more dedicated TAM wires (termed *TestRail* in Philips [6]).

To design a modular test architecture for a SOC with a given number of modules and a given number of test pins, the SOC integrator has to determine the following: (1) the number of individual TAMs and (2) their widths, (3) the assignment of modules to TAMs, and (4) wrapper design for each module [8]. These parameters need to be chosen such that the total number of pins used for the TAM wires is less than or equal to the given test pins, while the overall test cost is minimized.

For a small SOC, having only a few modules and a few test pins, a good test architecture can be designed manually. However, the complexity of designing an architecture increases exponentially with the increase in the number of modules and test pins. Iyengar et al. [8] proved that the problem of designing an optimal test architecture is  $\mathcal{NP}$  hard, indicating that the required compute time increases exponentially with the problem instance size. Therefore, there is a need for a tool which can efficiently search the solution space of feasible architectures and yield a (near-)optimal test architecture. Inside Philips, we have developed such a tool, named TR-ARCHITECT, for which we have reported good results obtained in negligible compute times [1, 9].

TR-ARCHITECT has two inputs: a SOC data file and a list of user options. The SOC data file describes the relevant SOC parameters, such as the number of modules inside the SOC, and for each module, the number of inputs, outputs, bi-directionals, test patterns, and scan chains with their lengths. The SOC data file of TR-ARCHITECT is encoded in the so-called \*.soc format, introduced for the ITC'02 SOC Test Benchmarks [10].



**Figure 4:** (a) Daisychain, (b) Distribution, and (c) Hybrid Architectures.

The user options currently available for TR-ARCHITECT are the following [1, 9]: (1) total number of SOC test pins, (2) type of modules (hard/soft), (3) external bypass per module (yes/no), (4) test schedule type (serial/parallel), (5) TAM type (test bus/TestRail), (6) architecture type, and (7) test cost. TR-ARCHITECT supports the design of three types of architectures: (1) the Daisychain Architecture [11],

(2) the Distribution Architecture [11], and (3) the Hybrid Architecture [1, 9]. Example instances for the three architectures are shown in Figure 4.

In case of a Daisychain Architecture, all modules in the SOC are connected to a single TAM with full bandwidth. In a Distribution Architecture, all modules have disjunct TAMs; TR-ARCHITECT optimally distributes the total TAM width among all modules using the SCDP algorithm [11]. For the Hybrid Architecture, TR-ARCHITECT uses a four-step algorithm described in [1, 9] to determine the number of TAMs, their widths, and module-to-TAM assignments.

TR-ARCHITECT minimizes the SOC test time [1, 9], possibly in conjunction with the routing wire length [12] and/or the number of test control pins [13].

## 5 Test Architecture Design

For the design of the PNX8550 test architecture, each chiplet was considered as a stand-alone entity, to be connected to a dedicated set of TAM wires. Such a Distribution Architecture [11] allows testing of chiplets in parallel, and hence contributed to minimizing the overall test time for the SOC. This approach yielded two tasks: (1) distribution of the total number of available TAM wires over the chiplets, and (2) design of a test architecture per chiplet.

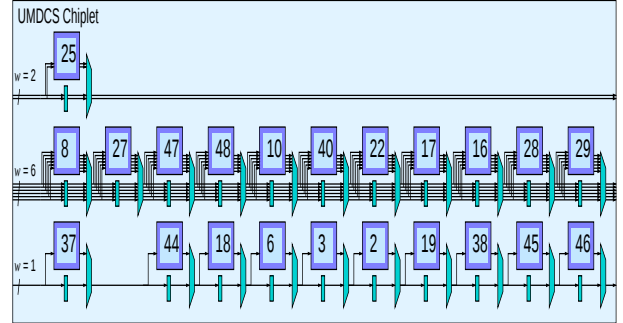
The chip design team could afford to spend 140 TAM wires, which, in test mode, connect to  $2 \times 140 = 280$  chip pins. These connections are time-multiplexed onto existing functional chip pins. The number 140 is the outcome of how many (reusable) synchronous digital chip pins were available and how much wiring the design team was willing to spend on TAMs. In the early design phase of the PNX8550, the tool TR-ARCHITECT was not available yet, and hence, the distribution of the 140 TAM wires over the 13 chiplets was based on extrapolation of test data for a predecessor SOC design and good engineering judgement. Figure 3(b) shows the distribution of the 140 TAM wires over the 13 chiplets.

Once the number of TAM wires per chiplet was decided, the remaining task was to design test architectures inside all chiplets. In principle, the Distribution Architecture [11] was chosen for all chiplets, in order to avoid the silicon area costs related to the dedicated bypass circuitry required by the Daisychain and Hybrid Architectures. For chiplets UMDCS and UTDCS, a Distribution Architecture was not possible; these chiplets have more cores than wires, while the Distribution Architecture requires that each core is assigned at least one TAM wire. For these two chiplets, a Hybrid Architecture [1, 9] was used.

For chiplets consisting of only one core, the test architec-

ture design was quite simple; all its TAM wires were assigned to the one core. However, for the chiplets containing multiple cores, test architecture design was done with the help of TR-ARCHITECT. For chiplets consisting of multiple cores and with the Distribution Architecture, TR-ARCHITECT determined the number of wires assigned to each individual core. For chiplets UMDCS and UTDCS, with a Hybrid Architecture, TR-ARCHITECT determined the number of TAMs, their widths, and the assignment of cores to TAMs.

Every core has a specific set of Pareto-Optimal TAM widths [8]. Assigning a core to a non-Pareto-Optimal TAM width leads to Type-2 idle bits [9], i.e., with less wires, this core still has the same test time). Ultimately, this can lead to unused (i.e. redundant) TAM wires. Next to optimizing the test architecture, TR-ARCHITECT reports on the amount of idle bits and unused TAM wires [9].



**Figure 5:** Hybrid Architecture advised for chiplet UMDCS.

Figure 5 shows the Hybrid Architecture advised by TR-ARCHITECT for chiplet UMDCS with 22 cores and 13 TAM wires. The numbers inside the cores represent the core ID. TR-ARCHITECT constructed three individual TAMs of widths 2, 6, and 1; in total four wires remained unused. This is due to the fact that with nine wires, the total test time of this chiplet is determined by Core 25. Core 25 has special characteristics, i.e., it has only one long internal scan chain of  $> 900$  flip-flops, a number of functional input and output terminals, and a relatively large number of test patterns ( $> 1000$ ). Therefore, if this core is assigned to a two-bit wide TAM, it reaches its minimum test time; assigning it more TAM wires does not further reduce its test time. With two wires, Core 25 dominates the overall test time, even while there are still four unused wires. This situation can only be resolved by re-designing the scan chains of this ‘bottleneck’ core.

The total SOC test time is determined by the maximum test time over all chiplets. If the chiplet that determines the overall test time has no unused wires (as was the case for PNX8550, see Section 6), re-designing scan chains of other chiplets in order to get rid of their unused wires does not bring any global test time reduction.

The test architectures as advised by TR-ARCHITECT were subject to some manual changes, in order to keep the scan chains from various clock domains separated, accommodate multiple-clock ATPG, and include access to small embedded memories. These changes are beyond the scope of this paper.

## 6 Results

In this section, we present test time results for five cases. In all five cases, the partitioning of the set of cores into chiplets is fixed, as described in Section 5. All test time numbers reported in this section are based on the assumption of only one clock domain per chiplet. In reality, most chiplets have multiple clock domains, for which the testing is handled by Philips-internal proprietary solutions; in this paper we abstract from that reality.

Case 1 is the original test architecture implemented on the PNX8550, in which the TAM width assignment to the chiplets was done manually, as described in Section 5. All chiplets have a Distribution Architecture, except for chiplets UMDCS and UTDCS, which have a Hybrid Architecture. The architectures for all chiplets were designed by TR-ARCHITECT without modifying the original number and

lengths of the core-internal scan chains.

In Cases 2a and 2b, the distribution of 140 TAM wires over the 13 chiplets was optimized by TR-ARCHITECT. Also in these cases, we did not modify the original number and lengths of the core-internal scan chains. In Case 2a, all chiplets had a Distribution Architecture. In Case 2b, all chiplets were allowed to have a Hybrid Architecture (which in some cases might end up becoming either a Distribution or Daisychain Architecture).

Cases 3a and 3b are equal to respectively Cases 2a and 2b, apart from the fact that we allowed TR-ARCHITECT to modify and optimize the number and lengths of the core-internal scan chains of all cores, except the Philips-external hard cores TriMedia (two instances) and MIPS.

For Cases 1, 2b and 3b, Table 1 lists the number of wires assigned to the chiplets and the resulting test time per chiplet. The maximum test time is printed in bold face. Figure 6 shows the corresponding test schedules. The horizontal axes display the test time, while the vertical axes show the TAM width; the latter is not to scale. The light numbered boxes depict the tests of the cores; the number in the box is

| Chiplet        | Case 1     |           |                         | Case 2b    |                         |             | Case 3b    |                         |             |
|----------------|------------|-----------|-------------------------|------------|-------------------------|-------------|------------|-------------------------|-------------|
|                | #TAM Wires |           | Test Time (clock cycle) | #TAM Wires | Test Time (clock cycle) | $\Delta T$  | #TAM Wires | Test Time (clock cycle) | $\Delta T$  |
|                | Assigned   | Unused    |                         |            |                         |             |            |                         |             |
| UMSP           | 9          | 1         | 1,503,479               | 6          | 1,886,040               | 25%         | 7          | 1,602,343               | 7%          |
| UMDCS          | 13         | 4         | 1,541,397               | 5          | 2,101,732               | 36%         | 6          | 1,689,467               | 10%         |
| UMBS           | 16         | 6         | 597,974                 | 3          | 1,817,149               | 203%        | 4          | 1,361,259               | 128%        |
| UMCU           | 3          | 0         | 2,494,687               | 3          | <b>2,494,687</b>        | 0%          | 4          | 1,434,985               | -42%        |
| UQVCP5L        | 21         | 2         | 1,301,162               | 15         | 2,279,801               | 75%         | 14         | 1,729,597               | 33%         |
| UQVCP2L        | 11         | 0         | 683,573                 | 3          | 2,025,224               | 196%        | 4          | 1,420,425               | 108%        |
| UVIP           | 6          | 0         | 787,814                 | 2          | 1,444,565               | 83%         | 2          | 1,372,879               | 74%         |
| UVMPG          | 7          | 4         | 1,403,060               | 1          | 2,349,895               | 67%         | 2          | 1,173,685               | -16%        |
| UTDCS          | 12         | 0         | <b>3,506,193</b>        | 17         | 2,485,886               | 29%         | 22         | 1,730,554               | -51%        |
| UCLOCK         | 2          | 0         | 223,097                 | 1          | 351,737                 | 57%         | 1          | 351,737                 | 58%         |
| MIPS           | 10         | 0         | 1,846,601               | 8          | 2,259,725               | 22%         | 11         | 1,653,533               | -10%        |
| TM1            | 15         | 0         | 2,953,559               | 18         | 2,428,079               | -18%        | 32         | 1,757,639               | -40%        |
| TM2            | 15         | 0         | 2,953,559               | 18         | 2,428,079               | -18%        | 31         | <b>1,766,095</b>        | -40%        |
| <b>PNX8550</b> | <b>140</b> | <b>17</b> | <b>3,506,193</b>        | <b>99</b>  | <b>2,494,687</b>        | <b>-29%</b> | <b>140</b> | <b>1,766,095</b>        | <b>-50%</b> |

Table 1: Test time results for all chiplets in PNX8550.

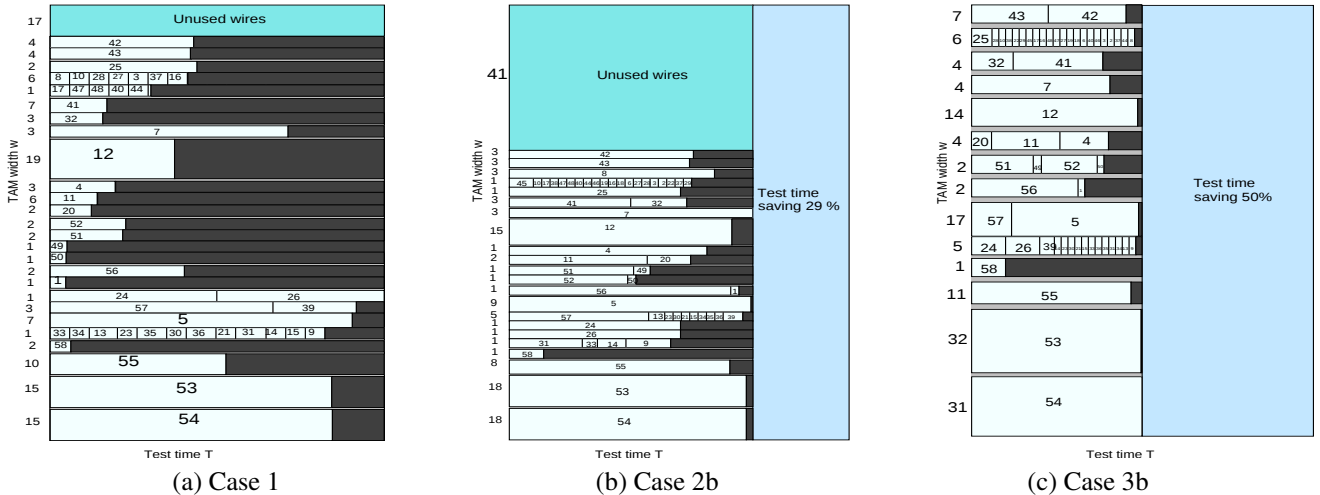


Figure 6: Test schedules of the various test architecture cases of PNX8550.

the core ID. The horizontal light-grey lines separate the test schedules of the individual chiplets.

For Case 1, the test time is determined by chiplet UTDCS with 3,506,193 clock cycles. After taking care of the testing with multiple clock domains, this number translated in a total test data volume that fitted onto the Agilent 93000-P600 test system with 28 M deep vector memories. The corresponding test schedule (cf. Figure 6(a)) contains a lot of Type-1 idle bits [9], indicated by the dark gray shading. The main reason for this is that the manual TAM width assignment to chiplets is not optimal. The bottleneck chiplet UTDCS was assigned 12 wires. Moving some TAM wires from chiplets with relatively short test times (such as UCLOCK, UMBS, UQVCP2L and UVIP) to UTDCS would have reduced the overall test time. Column 3 of Table 1 shows that some of the chiplet TAM width assignments are not Pareto-Optimal; in total, there are 17 unused wires.

For Case 2b, the TAM width assignment to chiplets was optimized by TR-ARCHITECT, and hence the test completion times of the various chiplets are much more balanced. This results in an overall test time reduction of 29%, compared to Case 1. In this case, the overall test time is dominated by chiplet UMCU which contains only one core (Core 7) with two very short and two long scan chains. If Core 7 is assigned three TAM wires, it reaches its minimum test time of 2,494,687 clock cycles and becomes the bottleneck core. TR-ARCHITECT can afford to assign three TAM wires to chiplet UMCU when it has 99 wires at SOC level; consequently, the other 41 TAM wires were left unused. Case 2a (not depicted) resulted in a test time of 2,494,687 clock cycles, obtained with 133 TAM wires.

In Cases 3a and 3b, the scan chains of most cores could be re-designed. This was also true for Core 7, and hence this was no longer a bottleneck core. In fact, we were able to use all 140 TAM wires effectively. In Case 3a (not depicted), TR-ARCHITECT obtained a test time of 2,337,317 clock cycles (33% reduction compared to Case 1). Case 3b yields a test time of 1,766,095 clock cycles (50% improvement over Case 1); chiplet TM2 gets assigned 31 TAM wires and determines the overall test time.

## 7 Conclusion

Philips' Nexperia™ Home Platform has adopted a modular test strategy. Such a test strategy requires an on-chip test access infrastructure, consisting of wrappers and TAMs. These wrappers and TAMs need to be carefully designed and optimized, in order to be able to fit the test data in the ATE vector memories and reduce the test application time.

This paper described the test architecture design for the PNX8550 SOC, part of Nexperia™ Home Platform series and the most complex SOC designed to date in Philips. As PNX8550 contains over 60 logic cores and 140 TAM

wires, design of its test access infrastructure was a big challenge; our tool TR-ARCHITECT helped to ease this task. A prototype version of TR-ARCHITECT became available halfway the design trajectory of PNX8550, when certain design decisions had already been frozen. Nevertheless, the tool helped to optimize the test architecture further within the given constraints and we successfully managed to fit the test onto the target ATE. Our paper also showed that, if TR-ARCHITECT would have been available from the project start onwards, further test time reductions up to 50% would have been possible.

## References

- [1] Sandeep Kumar Goel and Erik Jan Marinissen. Effective and Efficient Test Architecture Design for SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pages 529–538, Baltimore, MD, October 2002.
- [2] Yervant Zorian, Erik Jan Marinissen, and Sujit Dey. Testing Embedded-Core Based System Chips. In *Proceedings IEEE International Test Conference (ITC)*, pages 130–143, Washington, DC, October 1998.
- [3] Erik Jan Marinissen and Harald Vranken. On the Role of DFT in IC-ATE Matching. In *Digest of Papers of IEEE International Workshop on Test Resource Partitioning (TRP)*, Baltimore, MD, November 2001.
- [4] Santanu Dutta, Rune Jensen, and Alf Rieckmann. VIPER: A Multi-processor SOC for Advanced Set-Top Box and Digital TV Systems. *IEEE Design & Test of Computers*, 18(5):21–31, Sep-Oct 2001.
- [5] Bart Vermeulen, Steven Oostdijk, and Frank Bouwman. Test and Debug Strategy of the PNX8525 Nexperia™ Digital Video Platform System Chip. In *Proceedings IEEE International Test Conference (ITC)*, pages 121–130, Baltimore, MD, October 2001.
- [6] Erik Jan Marinissen et al. A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores. In *Proceedings IEEE International Test Conference (ITC)*, pages 284–293, Washington, DC, October 1998.
- [7] Erik Jan Marinissen et al. On IEEE P1500's Standard for Embedded Core Test. *Journal of Electronic Testing: Theory and Applications*, 18(4/5):365–383, August 2002.
- [8] Vikram Iyengar, Krishnendu Chakrabarty, and Erik Jan Marinissen. Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores. *Journal of Electronic Testing: Theory and Applications*, 18(2):213–230, April 2002.
- [9] Sandeep Kumar Goel and Erik Jan Marinissen. SOC Test Architecture Design for Efficient Utilization of Test Bandwidth. *ACM Transactions on Design Automation of Electronic Systems*, 8(4):399–429, October 2003.
- [10] Erik Jan Marinissen, Vikram Iyengar, and Krishnendu Chakrabarty. A Set of Benchmarks for Modular Testing of SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pages 519–528, Baltimore, MD, October 2002.
- [11] Joep Aerts and Erik Jan Marinissen. Scan Chain Design for Test Time Reduction in Core-Based ICs. In *Proceedings IEEE International Test Conference (ITC)*, pages 448–457, Washington, DC, October 1998.
- [12] Sandeep Kumar Goel and Erik Jan Marinissen. Layout-Driven SOC Test Architecture Design for Test Time and Wire Length Minimization. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 738–743, Munich, Germany, March 2003.
- [13] Sandeep Kumar Goel and Erik Jan Marinissen. Control-Aware Test Architecture Design for Modular SOC Testing. In *Proceedings IEEE European Test Workshop (ETW)*, pages 57–62, Maastricht, The Netherlands, May 2003.