The Development of an Online Support Tool for the Teaching and Learning of the IEEE Standard 1500 for Embedded Core-based Integrated Circuits

http://dx.doi.org/10.3991/ijoe.v8i4.2146

Ian Grout[1] and Abu Khari Bin A'ain[2]
[1] University of Limerick, Limerick, Ireland
[2] Universiti Teknologi Malaysia, Johor, Malaysia

***Abstract*—In this paper, an online education tool for assisting the teaching and learning of the IEEE 1500 standard testability method, used to support the testing of complex system-on-a-chip (SoC) integrated circuits (ICs), is developed and presented. The tool is an Internet browser based tool that supports the ability to investigate key aspects of the standard and its application to embedded core-based IC designs. The tool allows the user to create VHDL descriptions of both the test circuitry and the function circuitry via the Internet browser interface. The key considerations for developing this tool were to provide a computer based learning tool to support the teaching and learning of the standard and its application. This paper is an extended version of a paper presented at the EDUCON 2012 conference in April 2012.**

***Index Terms*—Design for testability; testing; IEEE standards; system-on-a-chip**

I. INTRODUCTION

The IEEE 1500 *standard testability method for embedded core-based integrated circuits* [1] was created to address the increasing test complexity with embedded core-based system-on-a-chip (SoC) designs [2]-[4]. These cores may be designed by the overall SoC design team, or are existing (pre-designed) cores which may be acquired from a 3rd party. Where an existing core is acquired, this core would be referred to as an *IP (intellectual property) core*. The IEEE 1500 standard testability method provides for a standard interface (set of connections between the core and other circuitry) along with a set of *rules* (mandatory aspects), *recommendations* (preferred practice) and *permissions* (optional features) for the system designer to follow in order to create an isolation boundary for test purposes between each digital core and the circuitry the core is connected to. It is a *design for testability* (*DfT*) technique that allows for the effective testing of complex integrated circuits that are comprised of embedded IP (intellectual property) cores which are integrated at the IC level to form an overall IC system function. For a design to be compliant with the standard, it **shall** adhere to the *rules*, **should** adhere to the *recommendations* and **may** include the *permissions*. To achieve a successful implementation of the standard, an in-depth knowledge of the standard requirements and its implementation are needed

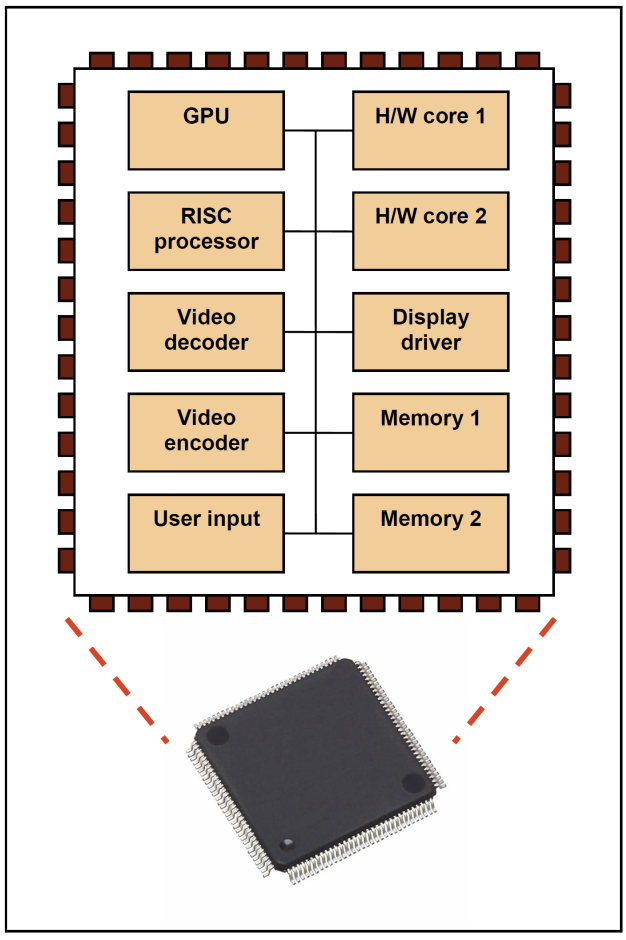

Figure 1. Example embedded core-based integrated circuit

which requires a seamless the integration of the design creation and design test processes. Fig.1 shows the architecture of an example IC comprised of IP blocks. Such designs are difficult to test due to their lack of controllability and observability at the overall system level. Here, the packaged device consists of a silicon die on which the system electronic circuitry is fabricated. The functional block diagram of this example design in Fig. 1 shows ten different cores which are connected with metal routing on the die and the die is then electrically bonded to the package. This is shown as an example of what can be

achieved, although typical SoC designs used in areas such as multimedia and communications would have a larger number of, and varied functionality, cores. These devices require to be designed, fabricated (manufactured) and tested. A fuller picture of design, manufacturing and test challenges can be found with reference to the *International Technology Roadmap for Semiconductors* (*ITRS*) [5]. For example, in the 2011 edition *Design* chapter [6], for SoC and SiP (system in a package) designs then:

> *"Integration of pre-existing design blocks into larger integrated devices produces nonlinear complexity growth for design tools, DFT, and manufacturing test, even when the blocks are homogeneous (such as all logic)."*

The range of existing test methods developed for smaller and lower operating frequency digital IC designs have become limited in their effectiveness and are expensive to undertake when attempting to apply these methods to the testing of SoC designs. SoC designs consist of cores ranging from digital logic through to memory, and part of the test requirement is to be able to effectively and cost efficiently access the cores and the interconnections between them during the test procedure. The access limitations encountered with the existing test methods can be overcome by the ability (through the use of the 1500 standard) to test the cores working together and to isolate each embedded core for test purposes.

The above considerations require an in-depth knowledge in the design and test of microelectronic circuits and systems. However, with the learning of the concepts and application of the standard, there can be difficulties in developing an understanding of the application of the standard to different IC design scenarios, and how the resulting test hardware is developed and applied. This can be particularly noticeable where students are introduced to the combination of design techniques, test methods, design for testability techniques and the use of industry standards – such a combination of concepts can be initially overwhelming, particularly where concepts need to be understood and applied in a practical design based project learning scenario.

The work described here considers the use of technology in the education environment, specifically targeting the IEEE 1500 standard testability method. The result is an Internet browser based tool that allows a user to interact with the design and analysis of a case study IC.

> *This paper is an extended version of the paper which was originally presented at the EDUCON 2012 conference [7] and provides an extended discussion into the motivation behind the work and the resulting online tutorial system.*

The basic idea in the 1500 standard is for each embedded core within the design to be accessed through a *wrapper serial port* (*WSP*) which provides access for both serial data (*WSI* (*wrapper serial input*) and *WSO* (*wrapper serial output*)) and *wrapper serial control* (*WSC*) connections. The IEEE 1500 standard testability method was created to address the increasing test complexity with embedded core-based system-on-a-chip (SoC) designs. It is one of a number of relevant standards that have been developed and adopted within the electronic/microelectronic industries. The key standards organizations are identified in Table I. The range of standards used cover aspects relating to design, manufacture (fabrication) and test. Whilst in the past much effort has been exerted towards the design area, it is now acknowledged that the manufacturing and test aspects as equally as important as the design; a good design must be able to be manufactured with a high quality and (cost) effectively tested.

TABLE I.
KEY STANDARDS BODIES

Organization	Acronym
The Institute of Electrical and Electronics Engineers	IEEE
International Electrotechnical Commission	IEC
Joint Electron Devices Engineering Council	JEDEC
US Department of Defense	DoD
American National Standards Institute	ANSI
British Standards Institute	BSI

The integration of design and test (both *DfT* and *built-in self-test (BIST)* hardware (and software)) have been focused on in the last number of years for analogue, digital and mixed-signal electronic circuits and systems. *DfT* and *BIST* are an integral part of many IC designs developed today and their inclusion is aimed at enhancing the IC test process and test program quality whilst keeping test costs as low as possible.

This section has provided an introduction to, and rationale for, the work undertaken; a background to the work and why the work was undertaken. The teaching and learning of microelectronic circuit test engineering and the increasing need to educate qualified test engineers to work in design and test activities are key requirements within the microelectronics field. The remainder of the paper is structured as follows: *section II* will provide a brief overview of the standard testability method. *Section III* will introduce the tools used in the development of the tutorial and *section IV* will describe the developed tutorial and its operation. *Section V* will conclude the paper and identify future work.

II. IEEE 1500 STANDARD TESTABILITY METHOD FOR EMBEDDED CORE-BASED INTEGRATED CIRCUITS

This section will introduce the IEEE 1500 standard testability method and its application; an overview of the testability method is presented, along with the operation of the test circuitry and the inclusion of relevant design for testability (DfT) into complex IC designs. It is a testability method for embedded core-based integrated circuits that was created to address the increasing test complexity with embedded core-based system-on-a-chip (SoC) designs. It provides for a standard interface and a set of *rules* (mandatory aspects), *recommendations* (preferred practice) and *permissions* (optional features) for the system designer to follow in order to create an isolation boundary between each digital core and the circuitry the core is connected to for test purposes.

With reference to Fig. 1, within a design then each identifiable core can be tested as both an isolated block of

circuitry and along with the other cores within the design. This then allows for:

1. Each core to be tested in isolation.
2. The interconnect between the cores to be tested.
3. The interaction between the cores to be tested.

The general structure of an embedded core with a *WSP* is shown in Fig. 2. Each core in the overall system design would be based on this architecture. The inclusion and use of such DfT electronic hardware requires a good knowledge of both circuit/system design and test methods. Each embedded core within the design to be accessed through a *wrapper serial port* (*WSP*) which provides access for both serial data (*WSI* (*wrapper serial input*) and *WSO* (*wrapper serial output*)) and *wrapper serial control* (*WSC*) connections.

Note that this testability method has many parallels to the IEEE Std 1149.1™-2001 - *IEEE Standard Test Access Port and Boundary-Scan Architecture* [8]. Test access (applying test signal inputs and receiving test signal outputs) is via the *wrapper serial port* and the design/operation of the *WSP* is set down in the standard.

Within the standard however, how the functionality is actually to be implemented (in digital logic) is not set by the standard and the actual implementation would be determined by the designer. However, some example circuit arrangements are provided for in the standard document, but in the main the functionality is described in words and block diagrams rather than circuit schematics. It is then for the designer to interpret the provided information in the standard to suit their own particular requirements.

Each core with its own wrapper is then integrated at the overall system level and the cores are connected together (with the *WSC* signals common to all cores and the data signals (*WSI* and *WSO*) arranged so that the cores are placed in a serial shift register arrangement).

Here, there are a number of key aspects to consider from a learner perspective, including:

1. Identifying relevant and linked standards. For example, the IEEE P1450.6 *Draft Standard for Standard Test Interface Language (STIL) for Digital Test Vector Data - Core Test Language (CTL)*, is used alongside the 1500 standard [9].
2. Understanding the structure of, and terminology used within, the standards. There are many acronyms to become familiar with and to use.
3. Understanding the role and application of the standard. Determining the mandatory, preferred and optional parts of the standard.
4. Understanding how to apply the standard to different design scenarios.
5. Developing an efficient and effective test strategy based on the use of this standard in conjunction with other standards (such as the IEEE Std 1149.1™-2001 - *IEEE Standard Test Access Port and Boundary-Scan Architecture* [8]).

III. TUTORIAL TOOL DEVELOPMENT

This section discusses the key technologies used in the development of the education tool; an introduction to, and summary of, the key software (and hardware) used in the creation of the education tool are presented. The target

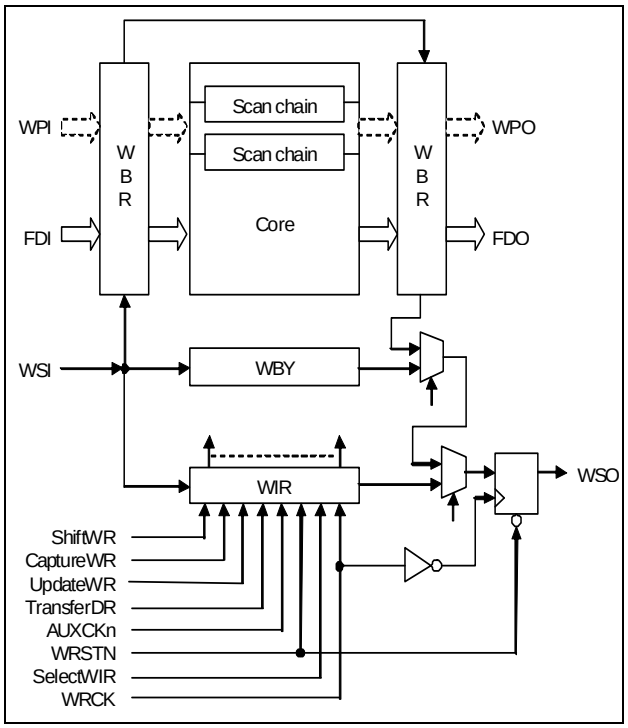

Figure 2. Core wrapper architecture

audience focus for this tool is primarily aimed at postgraduate student level, particularly those who would need to learn the concepts behind and application of this standard for use in research project work through a self-directed mode of learning. Given this focus, the tutorial is then considered for use by:

1. Postgraduate students undertaking a suitable course within a taught master's level programme where they are learning microelectronic circuit design and test engineering concepts. The tutorial would be used within laboratories and course project work.
2. Postgraduate students undertaking a research project and thesis within a taught masters level programme. The tutorial would be used for self-directed learning of the standard in the introductory stages of the project.
3. Postgraduate research students who would be required to utilize the standard as part of their personal research work. The tutorial would be used for self-directed learning of the standard in the introductory stages of their research project.

The content and structure of the tutorial was considered based on the learning requirements and the academic level of the target learner. For example, it was not considered necessary to provide features such as high levels of animation which were not considered central to the learning of the content. It was however necessary to consider that this tutorial is to act as an introduction rather than a single source of reference to the learner. Developing such a single source of reference site would have been extremely time consuming and with potentially little or no return to both the teacher and the learner. In addition, in order to be able to utilize the features of the standard in the future, the learner is required to obtain an insight into the standard which can only be achieved by undertaking practical experiments and not just an exercise in remembering

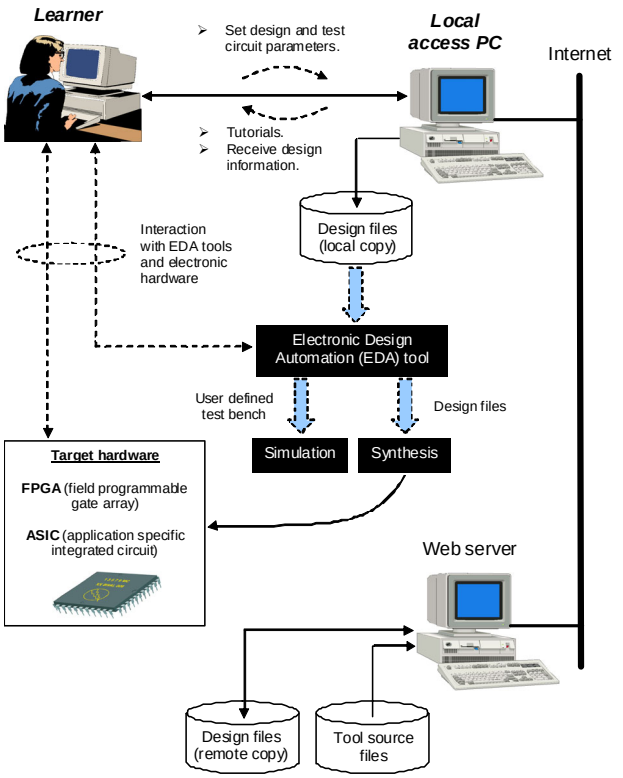

Figure 3. User interaction with the education system tools

material verbatim. Hence, the included case study exercise was considered to play a central role in the learner experience and would allow the learner to investigate their designs through logic simulation.

The user accesses the tutorial tool (based on HTML, PHP, CSS, Javascript, MySQL [10] and VHDL [11]) via the Internet browser from the local or remote web server. Fig. 3 shows how the tutorial pages are accessed by a learner from the main web server.

Fig. 3 also shows how the resulting design file (VHDL code) accessed via the tutorial can be used for design simulation and synthesis with an external EDA (electronic design automation) tool. Through the use of the resulting tutorial Internet browser pages, the user can then select aspects of the standard to investigate and can generate design data (in VHDL format) for both the test circuitry and the case study design. This design data can be investigated by the user by viewing and visually analyzing the design data. It is also provided in a form that can be both simulated and synthesized into logic (from the provided RTL (register transfer level) VHDL code) with a suitable off-line design tool (such as the Xilinx Inc. ISE (Integrated Software Environment) [12] for digital circuit and system design with field programmable gate arrays (FPGAs)). In the 1500 standard testability method, the use of a *core wrapper* is the fundamental building block for allowing access for test purposes Therefore, in order to apply the 1500 standard testability method, it is initially necessary to understand the design and operation of each part of the *core wrapper* circuit operation. This is undertaken with the use of VHDL based design descriptions. However, although VHDL is used here, there is no reason why the design data could not also be provided as Verilog-HDL [13] code if so required. The basic idea for using the tutorial is as follows:

1. The user logs onto the tutorial site and this gives access to all parts of the tutorial.
2. The user undertakes a self-directed learning approach in accessing each of the learning modules provided by the tutorial. They use the tutorial content and any other relevant references to become familiar with the standard and its application.
3. When the basic concepts are understood, if necessary through accessing other reference material (in particular the standard document itself), the user can access the case study design.
4. The case study design provides a means to access a template design (in VHDL code format) which does not include the testability circuitry. The template consists of an embedded core-based design which incorporates four embedded cores connected in a predefined manner (see Fig. 4). This template is not a complete code example in that the user must incorporate their own design details (by completing the VHDL *architecture* of each of the embedded cores).
5. When complete (including local simulation of the design using a suitable digital logic simulator), the modified VHDL files are uploaded to the tutorial site.
6. The wrapper architecture is automatically inserted and the user can then access their design with a particular implementation of the standard inserted. The user can then access the updated files and save these locally for analysis and evaluation using a suitable digital logic simulation tool.
7. The resulting VHDL code can be synthesized into logic and implemented in either FPGA or ASIC (application specific integrated circuit) forms using a suitable EDA tool. Whilst SoC designs are essentially ASIC designs, the FPGA provides for a low-cost and rapid design prototyping platform. In addition, various EDA tools provided by the FPGA vendors are free to access – accessing SoC design tools and manufacturing data would be out of the scope of many institutions due to the cost.

The case study design is based on VHDL code consisting of five VHDL files (each file consists of a VHDL *entity* and *architecture* pair) which describe the four embedded cores and the top level design (core integration and case study IC design primary inputs and outputs). The initial code accessed does not contain any *architecture* description. The user must therefore enter his or her own unique design details which can then be tested by using the *wrapper serial port*.

The tutorial is a custom designed tool based on the following software applications and programming languages:

1. **Apache web (HTTPD) server.** This provides the web server on which the tutorial is hosted and accessed from.
2. **HTML**. The Hypertext Markup Language used for creating the web pages to be viewed on the Internet browser of choice: it is important here to ensure that the generated HTML code is compatible with the main Internet browsers in use (Microsoft *Internet Explorer*, Opera Software *Opera*, Mozilla *Firefox* and Google *Chrome*).

3. **CSS**. Cascaded style sheets which are used to provide a consistent formatting and look of all pages within the tutorial.
4. **PHP**. The PHP Hypertext Processor scripting language to generate dynamic HTML pages and to interact with the MySQL database and operating system.
5. **MySQL**. The database to hold user information and design data information as it is generated and used.
6. **Javascript**. Used here to provide for web page error checking and providing user interaction with embedded figures within the HTML pages.
7. **VHDL**. The VHSIC hardware description language which describes the digital logic circuit design used as the case study design. The case study design architecture is shown in Fig. 4.

In VHDL code form, Fig. 5 shows the top level design *entity* prior to core wrapper insertion and Fig. 6 shows the top level design entity *after* core wrapper insertion. The additional input and output signals are defined to be of the VHDL `STD_LOGIC` / `STD_LOGIC_VECTOR` type as are all input and output port signals in this design.

IV. TECHNOLOGY TOOL PLANNING AND IMPLEMENTATION

This section discusses the education tool development and operation; the creation of the Internet browser based software tool is presented and the key features considered in this (first) version of the tool are discussed.

The basis for the tutorial is to provide an introduction to the standard and allow the user to investigate specific aspects of the standard through the case study design. It was not the intention to provide for a comprehensive source of material that would cover every aspect of the standard, rather to provide a reference source for an introduction to the standard with the target audience being postgraduate student level. Currently, the tutorial (version 1) has been developed and is being evaluated. Based on the results of the evaluation, further refinements of the structure, appearance and content of the tutorial would inevitably be necessary.

Fig. 7 shows the home page for the tutorial – structured in a "traditional" web page layout manner. The navigation bar across the top of the page is common to each of the *learning modules* (a *learning module* here being one of the linked pages providing specific aspects relating to the standard). The layout of the page consists of a simple menu system along the top of the page and the content below it. This is a common format for all tutorial pages.

Fig. 8 shows how the tutorial is structured and the different parts of the tutorial (and external links) can be accessed. The initial five pages (black boxes in Fig. 8) provide background information for the initial learning stage.

The next page (gray box in Fig. 8) provides the case study access. Although the case study is shown here as only one part of the tutorial, it would be anticipated that actually the user would spend time on this aspect of the learning as it is used to practice the implementation of the standard within a user created design (with the design based on the provided design template). The last two pages (white boxes in Fig. 8) provide additional references and information that the users can update.

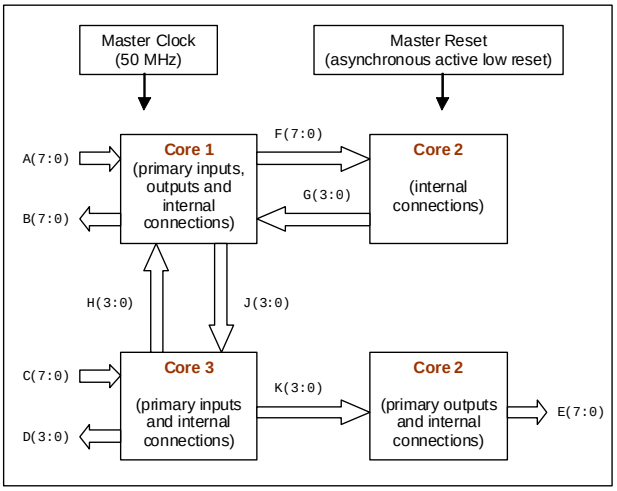

Figure 4. Case study design consisting of four embedded cores

```
ENTITY Top IS
   PORT( Master_Clock : IN   STD_LOGIC;
         Master_Reset : IN   STD_LOGIC;
            A : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
            B : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0);
            C : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
            D : OUT  STD_LOGIC_VECTOR (3 DOWNTO 0);
            E : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0));
END ENTITY Top;
```

Figure 5. Top level *entity* before core wrapper insertion

```
ENTITY Top_With_Wrapper IS
   PORT( Master_Clock : IN   STD_LOGIC;
         Master_Reset : IN   STD_LOGIC;
            A : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
            B : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0);
            C : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
            D : OUT  STD_LOGIC_VECTOR (3 DOWNTO 0);
            E : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0));
-- Insert core wrapper test control inputs
      ShiftWR : IN  STD_LOGIC;
    CaptureWR : IN  STD_LOGIC;
     UpdateWR : IN  STD_LOGIC;
   TransferDR : IN  STD_LOGIC;
       AUXclk : IN  STD_LOGIC;
        WRSTN : IN  STD_LOGIC;
    SelectWIR : IN  STD_LOGIC;
         WRCK : IN  STD_LOGIC;
-- Insert core wrapper test data I/O
          WSI : IN  STD_LOGIC;
          WSO : OUT STD_LOGIC);
END ENTITY Top_With_Wrapper;
```

Figure 6. Top level *entity* after core wrapper insertion

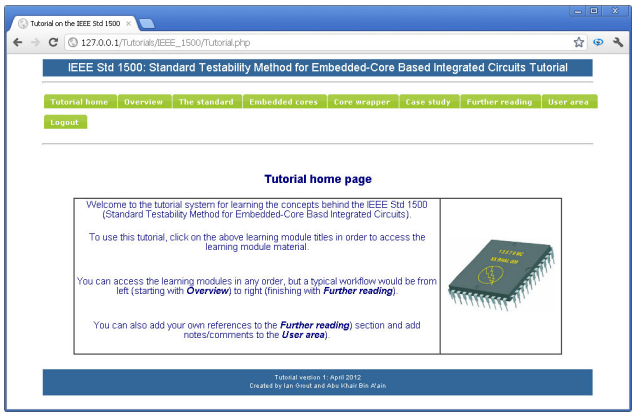

Figure 7. Tutorial home page

Although it is envisaged that a typical flow would be from top to bottom in Fig. 8, there would be no reason why the *learning modules* could not be accessed in any other order. However, care would need to be taken by the user to ensure that the important background theory is not simply ignored before proceeding to the case study design.

Where the material to be introduced is supported with the aid of images and figures, these images and figures are client-side image-mapped using the HTML `<map>` tag so that when the mouse is over part of the image (including) text, additional information relevant to that part of the image is displayed in the window. For example, Fig. 9 shows the page introducing the *wrapper serial port* (*WSP*). This image shows the additional text (in red) which is displayed when the mouse is over the *TransferDR* input text. The aim here is to allow the user to see information relevant to the particular part of the *WSP* that they are interested in.

With the access to, and generation of, the VHDL files for core wrapper insertion in the case study design, the user is provided with the *Case study* page and embedded form to complete. There are four main user steps involved in this part of the tutorial. These four steps are:

STEP 1: Add designer information

The user is prompted to enter the following details:

- Design name: the name of the circuit design.
- Designer name.
- Organization.
- Case study design summary: a short description.

STEP 2: View VHDL source code templates [prior to core wrapper insertion]

- A set of template VHDL files are provided which are to be completed with the user's own design descriptions (VHDL code *architectures* to be completed).
- The template files are to be copied to the user's own computer and then edited with the necessary architecture descriptions.

STEP 3: Set VHDL design files to upload [prior to core wrapper insertion]

- The modified VHDL template files from STEP 2 are to be selected for core wrapper circuitry insertion (in STEP 4).

STEP 4: Upload design files and run core wrapper insertion

- When this option is acted on, the core wrapper test circuitry is automatically inserted into the design VHDL files and additional test I/O and internal signals are created.
- The user is redirected to a new browser page from where the new design files can be accessed.

The user is allowed to upload design data and design files via the *Case Study* page. Fig. 10 shows the design submission area of the tutorial page and shows the four steps to this process as identified above.

All fields within the form require completion and a submission would only be possible if the required data has

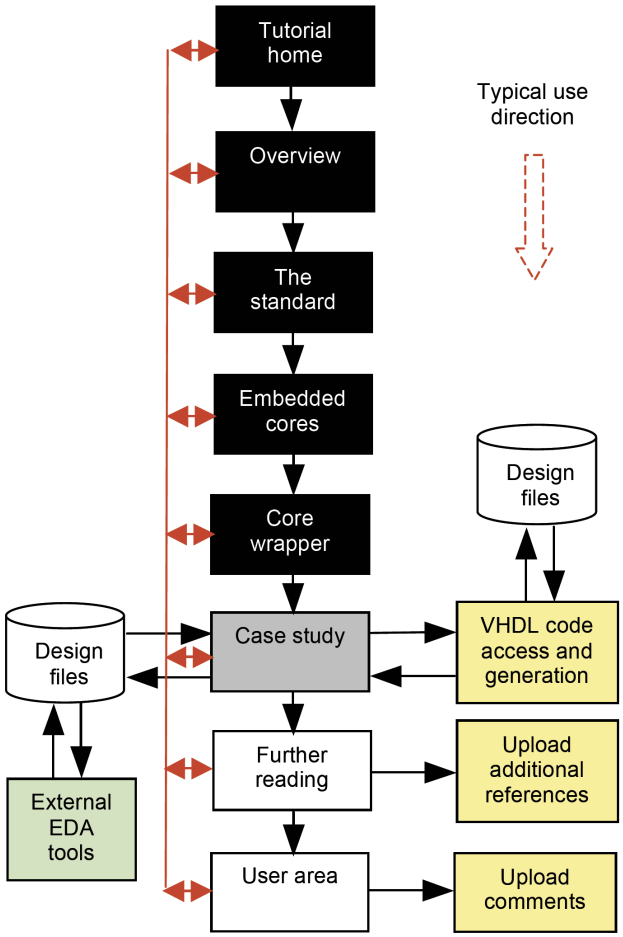

Figure 8. Navigating the tutorial

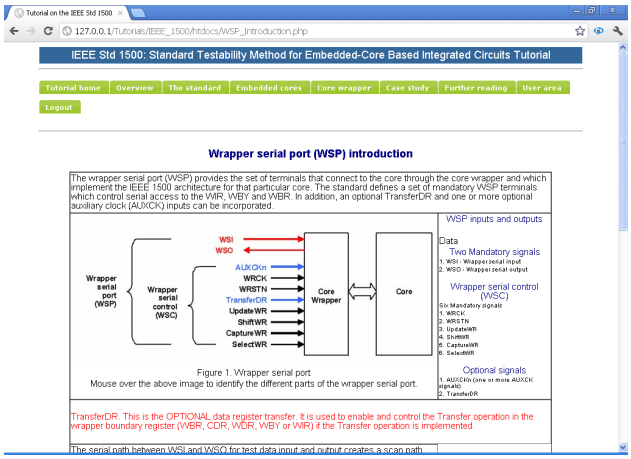

Figure 9. Navigating the tutorial

been entered into the form. The rationale for choosing this approach was to give a suitable level of user input without having the understanding of how to the use of the tutorial becoming the main focus of the work, so allowing the user to focus on the design and testability aspects.

Fig. 11 shows the VHDL code template for *Core 1*. The user accesses the code and completes the *architecture body* description (highlighted in red) with their unique design description. With this approach, each user would create an individual design which is tailored for their particular use. This provides each user with an individual case study functionality based on the single overall top

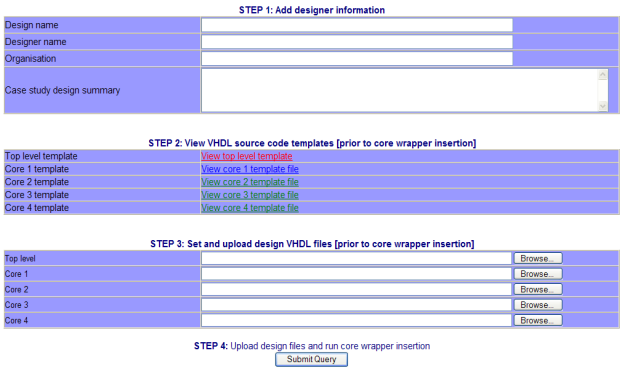

Figure 10. Submitting design data

```
------------------------------------------------------------
-
-- Case study design: Core1 design TEMPLATE
-- Version 1 (November 2011)
------------------------------------------------------------
-

------------------------------------------------------------
-
-- Libraries and packages
------------------------------------------------------------
-

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

------------------------------------------------------------
-
-- Core1 entity
------------------------------------------------------------
-

ENTITY Core1 IS
       PORT( Master_Clock : IN   STD_LOGIC;
             Master_Reset : IN   STD_LOGIC;
                 A : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
                 B : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0);
                 F : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0);
                 G : IN   STD_LOGIC_VECTOR (3 DOWNTO 0);
                 H : IN   STD_LOGIC_VECTOR (3 DOWNTO 0);
                 J : OUT  STD_LOGIC_VECTOR (3 DOWNTO 0));
END ENTITY Core1;

------------------------------------------------------------
-
-- Core1 architecture
------------------------------------------------------------
-

ARCHITECTURE Template OF Core1 IS

-----------------------------------------
-- Architecture main body
-----------------------------------------

BEGIN

END ARCHITECTURE Template;

------------------------------------------------------------
-
-- End of file
------------------------------------------------------------
-
```

Figure 11. Core 1 VHDL code template

level design. It would however be possible to make the process more flexible for the user by allowing a more customizable design (such as varying the number of cores and how the cores are interconnected) via the tutorial page. This however would be future work.

User interaction with the tutorial is an important consideration. Accessing the case study is only one way in which the user can interact with the tutorial. Two additional ways are:

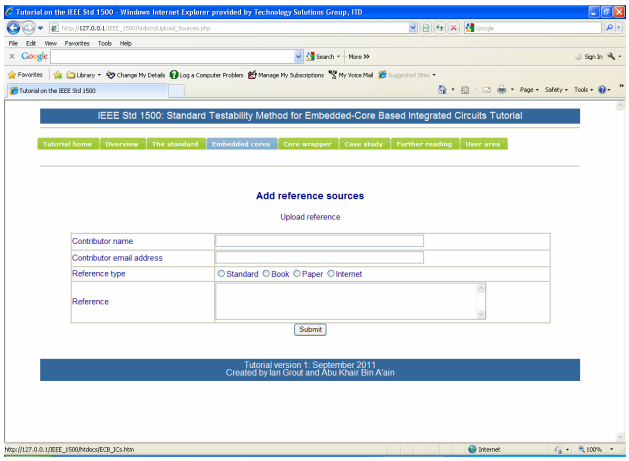

Figure 12. Additional reference upload form

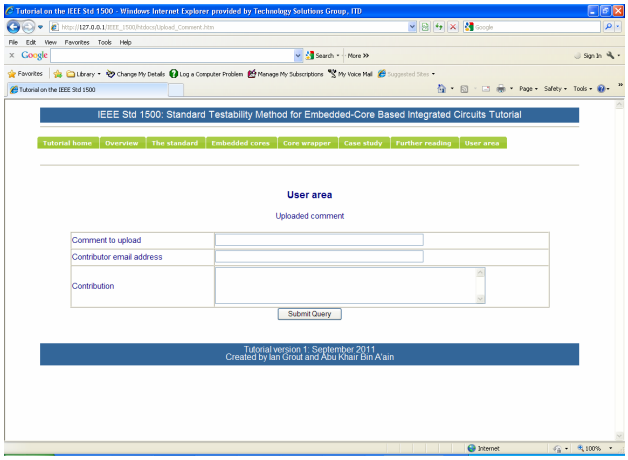

Figure 13. Comment upload form

1. Include additional references.

   A user can include their own references to supplement the provided reference sources. These additional references are also available to all other users. The form to allow for this action is shown in Fig. 12. However, care would need to be taken with these user contributions as they would not be peer reviewed for suitability and accuracy.

2. Upload comments.

   A user can upload contributions in the form of comments. The form to allow for this action is shown in Fig. 13. However, care would need to be taken with these user contributions as they would not be initially peer reviewed for suitability and accuracy. The tutorial administrator however has the option to review the comments and provide notes/clarifications alongside the contribution if necessary.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a tutorial tool for aiding the learning of a key standard used in the design and testing of microelectronic embedded core-based system-on-a-chip (SoC) designs was presented. Specifically, the tutorial is aimed at the IEEE 1500 standard testability method for embedded core-based integrated circuits. The target audience for the tutorial is the postgraduate student level who would need to learn the basics of this testability method through

a self-directed mode of learning in order to implement the standard within their own embedded core-based integrated circuit designs. The structure and operation of the tutorial were introduced and the manner in which the learner would use the tutorial was identified. The tutorial itself has been developed and evaluation is currently being undertaken with the aim to evaluate and enhance its use.

REFERENCES

[1] IEEE Std 1500™-2005: IEEE Standard Testability Method for Embedded Core-based Integrated Circuits, The Institute of Electrical and Electronics Engineers (IEEE)

[2] Erik Jan Marinissen and Yervant Zorian, "IEEE Std 1500 Enables Modular SoC Testing", IEEE Design & Test of Computers, Vol. 26, Issue 1, 2009, pp8-17 http://dx.doi.org/10.1109/MDT.2009.12

[3] Benoit Nadeau-Dostie, Saman M.I. Adham, and Russell Abbott, "Improved Core Isolation and Access for Hierarchical Embedded Test", IEEE Design & Test of Computers, Vol. 26, Issue 1, 2009, pp18-25 http://dx.doi.org/10.1109/MDT.2009.13

[4] Alfredo Benso, Stefano Di Carlo, Paolo Prinetto and Alberto Bosio, "Are IEEE-1500-Compliant Cores Really Compliant to the Standard?", IEEE Design & Test of Computers, Vol. 26, Issue 3, 2009, pp16-24 http://dx.doi.org/10.1109/MDT.2009.46

[5] International Technology Roadmap for Semiconductors (ITRS), 2011 Edition. [Online]. Available: http://www.itrs.net/

[6] International Technology Roadmap for Semiconductors (ITRS), 2011 Edition Design Chapter,. [Online]. Available: http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Design.pdf

[7] Ian Grout and Abu Khari Bin A'ain, "Internet based support tool for the teaching and learning of the IEEE standard 1500 for embedded core-based integrated circuits", Proceedings of the IEEE EDUCON 2012 conference, Marrakech, Morocco, 17-20 April 2012

[8] IEEE Std 1149.1™-2001 (R2008) (Revision of IEEE Std 1149.1-1990): IEEE Standard Test Access Port and Boundary-Scan Architecture, The Institute of Electrical and Electronics Engineers (IEEE)

[9] P1450.6 Draft Standard for Standard Test Interface Language (STIL) for Digital Test Vector Data - Core Test Language (CTL), The Institute of Electrical and Electronics Engineers (IEEE)

[10] Meloni J., SAMS Teach Yourself PHP, MySQLTM and Apache in 24 Hours, SAMS, 2003, ISBN 0-672-32489-X

[11] IEEE Std 1076™-2008 (Revision of IEEE Std 1076-2002): IEEE Standard VHDL Language Reference Manual, The Institute of Electrical and Electronics Engineers (IEEE)

[12] Xilinx Inc. [Online]. Available: http://www.xilinx.com

[13] 8EEE Std 1364TM-2005: IEEE Standard for Verilog Hardware Description Language, The Institute of Electrical and Electronics Engineers (IEEE)

AUTHORS

Ian Grout is with the University of Limerick, Limerick, Ireland (e-mail: Ian.Grout@ul.ie).

Abu Khari Bin A'ain is with the Universiti Teknologi Malaysia, Johor, Malaysia (e-mail: abu@fke.utm.my).

This article is a modified and extended version of a paper presented at the International Conference EDUCON2012, held April 2012, at University Mohammed V Souissi, Marrakesh, Morocco. Received 6 June 2012. Published as resubmitted by the authors 23 October 2012.