ITB Journal

Issue Number 13, May 2006

Contents

The academic journal of the Institute of Technology Blanchardstown

http://www.itb.ie/site/researchinnovation/itbjournal.htm

http://www.itb.ie/site/researchinnovation/itbjournal.htm

Papers for submission to the next ITB Journal should be sent to the editor at the address below. Alternatively, papers can be submitted in MS-Word format via email to brian.nolan@itb.ie

Dr. Brian Nolan
Editor
ITB Journal
Institute of Technology Blanchardstown
Blanchardstown Road North
Blanchardstown
Dublin 15

Editorial

I am delighted to introduce the **13th edition of the ITB Journal**, the academic journal of the Institute of Technology Blanchardstown.

Again, we have the usual diverse, but interesting, mix of papers. Software and hardware engineering is, however, the major theme within this issue.

Specifically, in this issue of the ITB Journal we have papers that present recent research work on virtual learning environments for Astronomy as part of the Virtual Telescopes in Education as undertaken by researchers from ITB and NUI Maynooth, leading edge engineering in the service of wheelchair navigation at UL, a critical review (also from UL) of System-On-Chip (SoC) designs containing multiple IP (Intellectual Property) cores and why it is not yet possible to apply generic test architectures to an IP core within a SoC, and ongoing work in ITB on a rapid prototyping system using SystemC. In particular, it looks at the suitability of using SystemC for rapid prototyping of embedded systems.

We hope that you enjoy the papers in this issue of the ITB Journal.

Dr. Brian Nolan
Editor
ITB Journal
Institute of Technology Blanchardstown
Blanchardstown Road North
Blanchardstown
Dublin 15
Email: brian.nolan@itb.ie

Integration and Communication of Process Support Tools in an online Virtual Learning Environment

L. Raeside
Department of Informatics and Enginneering,
Institute of Technology Blanchardstown,
Blanchardstown Road North, Blanchardstown,
Dublin 15, Ireland

B. Busschots, Shen Fei, J. G. Keating
Department of Computer Science,
National University of Ireland, Maynooth,
Maynooth, Co. Kildare, Ireland
May 7, 2006

Abstract

The Education through Virtual Experience (EVE) research group based at the National University of Ireland Maynooth have successfully developed an online Virtual Learning Environment (VLE) for Astronomy as part of the Virtual Telescopes in Education (VTIE) project. The VTIE VLE has been developed to provide online software support for the engagement of schoolchildren and their teachers in a scientific process. The development of an online VLE to support a scientific process has brought about complex interactions from the user perspective. This has lead to complex integration and communication challenges from the software implementation perspective. This paper contains a high level user-centered summary of the integration and communication challenges that have arisen during the VLE development and summarizes the mechanisms used to meet these challenges.

1. Introduction

The VTIE VLE, the virtual learning environment discussed in this paper, has already been used successfully to engage schoolchildren in a scientific process through Astronomy based experimentation [1]. The VLE has been constructed using a rapid prototyping approach combined with formative evaluations [2]. The VLE provides a software environment through which schoolchildren and their teachers can create project proposals, investigate and analyze data and images associated with that proposal and engage in team based collaborative writing to summarize all of the work carried out during the course of a project [3]. The completion of the VTIE scientific process culminates in the production of a project research paper. Following the completion of the formative evaluations of the VLE, the requirement for a set of fully integrated process support tools was identified. The aim of these tools is to fully support of all the above phases of the VTIE scientific process from project proposal to the generation of the project research paper. These tools are required to work online and be consistent in terms of look-and-feel, interoperability and usability within the VLE. The EVE research team have developed a number of tools to support the scientific process including the Collaborative Writing Environment (the CWE includes the Project Design and Management

Interfaces) [4], the Scrapbook Tool, the Imaging Tool [5] and the Graphing Tool. The CWE allows students to produce a collaboratively written document within the VTIE environment that amalgamates all of the experiences of a group of students working on a particular project into a research paper. The Scrapbook Tool provides the basis for sharing of information resources between students within the VLE. The Imaging Tool provides a mechanism for students to analyze images and produce data about the images being analyzed. Finally, the Graphing Tool ensures that student data can be shown graphically via plots and graphs, these graphs are added to the group research paper. This paper summarizes the software design constraints applied to the VLE in order to ensure consistent interoperability between the process support tools.

2. VTIE VLE Architecture

The VTIE VLE is a Java-based solution that has been constructed to operate within the Firefox web browser. The Firefox browser is part of the Mozilla project, is available online and may be downloaded and installed by participating schools at no cost [6]. There are a number of features that make the Firefox browser a suitable front-end for the VTIE VLE including the provision for rich customizability through software extensions, the provision of separate browser tabs which enhance the usability while navigating the VLE, and a high level of compatibility with the W3C standards [7]. In order to maximize the ease of integration of tools within VTIE, the VLE has been implemented using the Apache Struts framework [8]. The Apache Struts framework is based on the Model View Controller Model 2 (MVC2) design pattern and offers a well structured architecture through which new tools can be seamlessly introduced [9]. Each new software tool introduced into the VLE must follow the Struts framework specification, i.e., JSP is used for the view and the model is implemented using Java classes (namely Action classes). The Struts Controller is implemented as a servlet that maps events to particular classes using an XML based configuration file. The integration of new tools involves the editing of this configuration file and the addition of server classes to implement program logic [10]. Access to the VTIE database is carried out in a unified manner using Java utility classes designed at the architectural level. This ensures that each of the software tools can achieve database access without having to implement the low-level code statements to store and share information. The VTIE utility packages also provide unified access to various session-related information such as direct access to information relating to the current users and user groups. Communication between the various tools within the VLE is achieved using XML. The front-end communicates with the server via AJAX [11] and the server based components use the JDOM package [12] to produce and parse XML schema. The XML schema are used to represent data passed between tools and between the client and

server. Figure 1 shows a diagram of the VTIE VLE Architecture and the communications mechanisms between the client and server components.

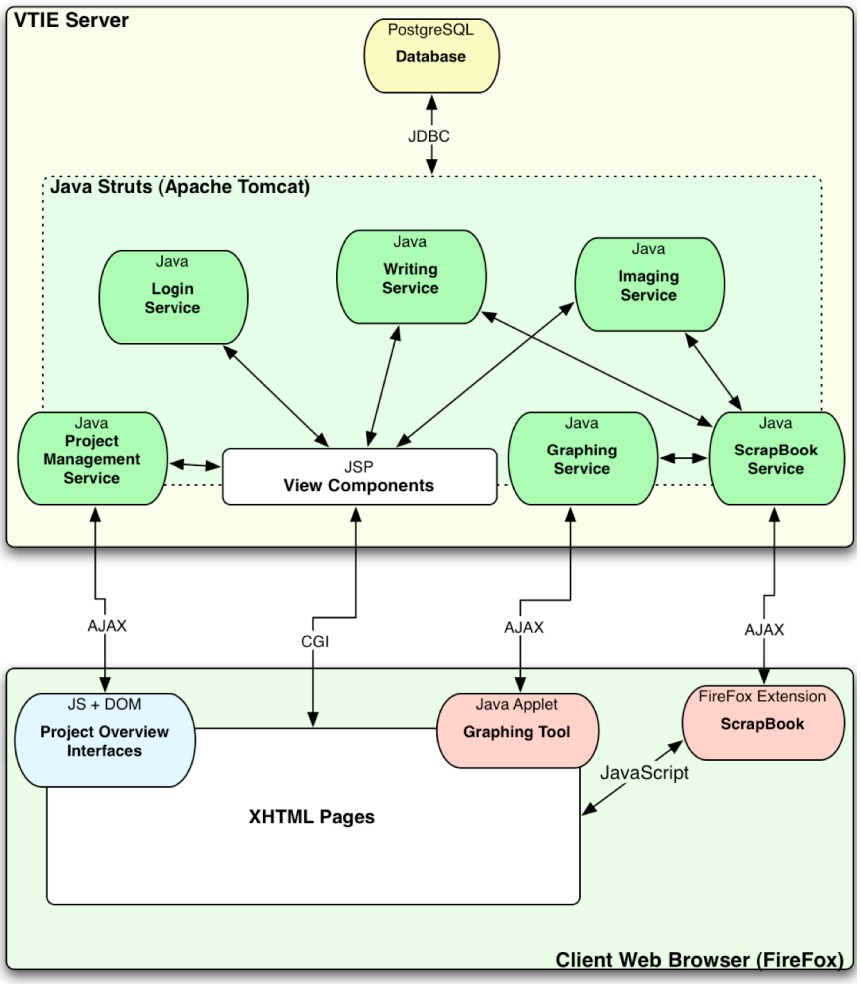

Figure 1: Architectural Diagram of the VTIE VLE

3. Examples of Tool Integration and Communication in VTIE

The requirement to support the VTIE scientific process has been the determining factor with respect to the communications between the various software tools. The communication links between the various tools have been formed directly from the user requirements and the educational goals of the VLE. This section provides examples of some of the communication that takes place between the process support tools within VTIE.

The Project Design and Management interface (a component within the CWE) is responsible for initializing projects, setting up assignments, assigning students into various groups and the assignment of members of each group to particular aspects of each project. The Project Design and Management interface is used primarily by the students' mentor, the mentor may be a teacher or any other person acting as the coordinator of student groups. The mentor can use this interface to assign particular sections of a document to particular members of each student group. The students then work in their respective teams while

following the VTIE scientific process described above. The UML Use Case diagram in Figure 2 represents the main functions that each student carries out while completing a project. The Write Paper Use Case shown in Figure 2 involves the use of the Scrapbook Tool to gather various resources and include them in the project research paper, it also involves the use of the Collaborative Writing interface within the CWE and may also require the use of the Graphing Tool if the student wishes to produce a graph of project data. The Analyze Images Use Case shown in Figure 2 involves the launching of the Imaging Tool. The student may have stored images relating to the project and may wish to carry out measurements on these images. The Browse Web Use Case involves the use of the Firefox browser to search other online resources for information relating to the project, this phase of the project uses the Scrapbook Tool to store URLs, text, and any other information relating to the project. The Scrapbook Tool is capable of persistently storing any online resource while students browse the Web. This information is stored automatically in the VTIE database and can be shared within the students' group and may also be viewed by the mentor.

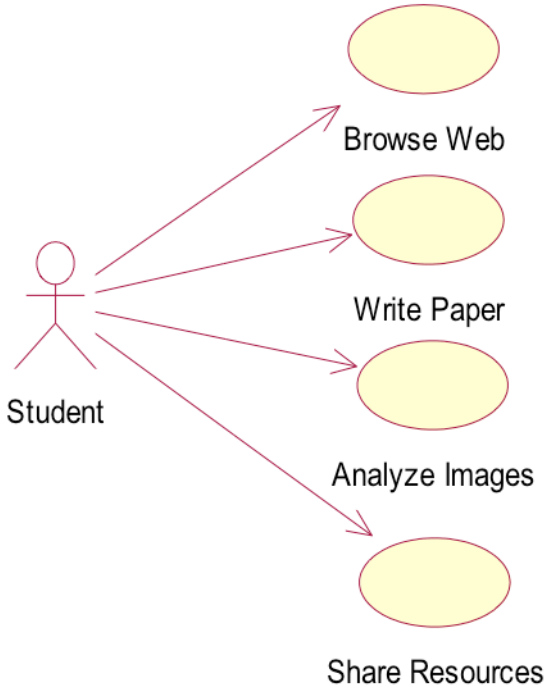

Figure 2: UML Use Case diagram of main activities for the student within the VTIE Process

The Scrapbook Tool is implemented as a Firefox extension. This tool operates within the browser window and can be used within the VTIE environment. However, this tool can also be used as a general purpose tool for the storage and retrieval of online resources within Firefox. The UML Use Case diagram in Figure 3 represents a student-centered view of the main functions performed through the Scrapbook Tool. The Scrapbook Tool integrates and

communicates with all of the other tools within the VTIE VLE and provides a universal linking mechanism to the project research paper. The Scrapbook Tool represents an example of a communication intermediary between other tools within the VTIE VLE. The Scrapbook Tool communicates with other tools in the VLE through direct interaction with the students and is constantly available in the left pane of the browser window. Students simply carry out click-and-drag operations on online resources from the main browser window into the Scrapbook Tool as required. The student resources are automatically saved to the database and are associated with either the individual student or the student's group. The Scrapbook Tool is also used as an intermediate communication link between the Imaging Tool and the Graphing Tool and the research paper created using the CWE. Images that have been modified within the Imaging Tool and graphs that have been generated within the Graphing Tool can be saved directly to the Scrapbook Tool as with all other online resources. The Add Scrap To Paper Use Case shown in Figure 3 allows students to add any of the resources collected into the project research paper including those images and plots generated within the VLE.

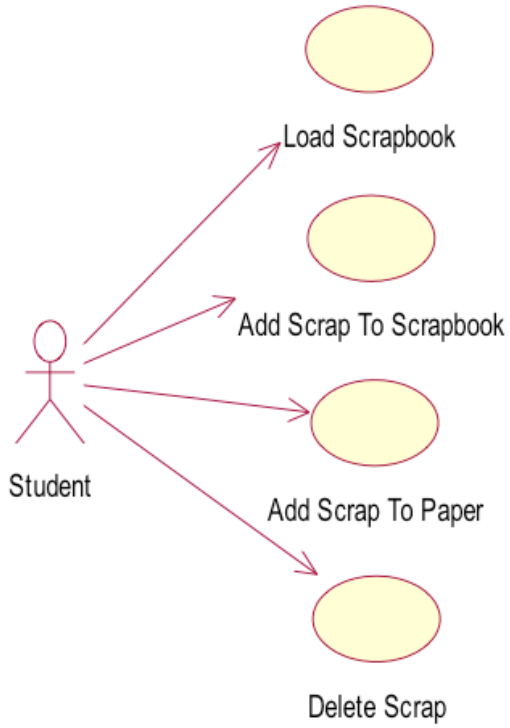

Figure 3: UML Use Case diagram of main functionality provided by the VTIE Scrapbook Tool

The analysis of images using the Imaging Tool and the Graphing Tool represents a direct tool-to-tool communication link within the VTIE VLE using the VTIE database and XML. The UML Use Case diagrams in Figures 4 and 5 represent a student-centered view of the functionality provided by the Imaging and Graphing Tools respectively. As part of the

analysis phase of the process the student must be able to gather data about images and then plot that data when required. Students may choose to gather data as one brief activity or they may choose to accumulate data over many days, weeks and perhaps even longer periods of time. In order to cater for the various time lapsed experiments that students may perform using VTIE it is important that data associated with images be stored persistently and that this data be associated with the student, the project and the image being analyzed. The Imaging Tool provides the functionality to store data associated with the student images in the database using an XML representation.

Figure 6 shows an example of the type of information stored in the database, the XML instance shown represents an angular measure carried out using the Imaging Tool (an angle measure requires the storage of three points in the image and the angle calculated between those points using the Imaging Tool). It is this XML database representation that forms the communication link between the Imaging Tool and Graphing Tool. When the student collects the necessary data from the image being analyzed this data is stored in XML format into the database, subsequently, the Graphing Tool may be launched by the student via the Imaging Tool or from elsewhere within the VTIE VLE. When the student chooses to launch the Graphing Tool directly from within the Imaging Tool an XML representation of all of the data saved for the current image is retrieved by the Graphing Tool from the database.

The Graphing Tool retrieves the XML representation of the image being analyzed and uses JDOM to extract the image data for display within the data interface component of the Graphing Tool. Finally, the options interface of the Graphing Tool allows the student to select and modify the data they wish to display so that the graph is of the desired format to be added to the project research paper. The Graphing Tool provides the functionality to store the current plot as an image representation that can be retrieved, stored, and shared via the VTIE Scrapbook Tool as described above.

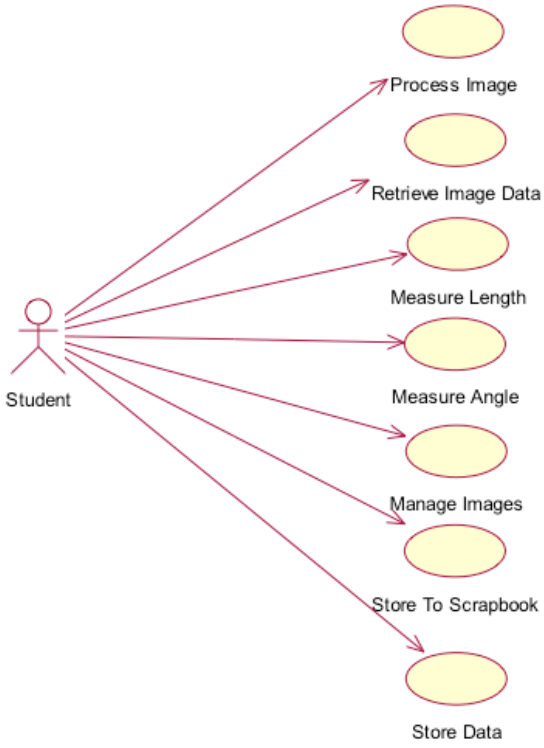

Figure 4: UML Use Case diagram of main functionality provided by the VTIE Imaging Tool

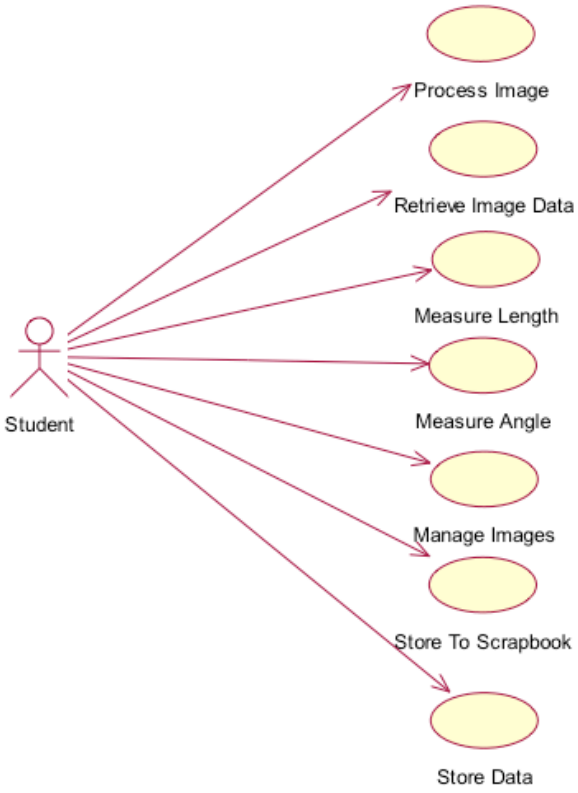

Figure 5: UML Use Case diagram of main functionality provided by the VTIE Graphing Tool

```
<angleMeasure>
<innerAngle> 134 </angleMeasure>
<outerAngle> 226 </outerAngle>
<centerX> 200 </centerX>
<centerY> 198 </centerY>
<x1> 157 </x1>
<x2> 117 </x2>
<y1> 278 </y1>
<y2> 219 </y2>
</angleMeasure>
```

Figure 6: XML Representation of and Angle measurement using the VTIE Imaging Tool

4. Conclusions

The formative evaluations of the VTIE VLE have demonstrated that it is possible to successfully engage schoolchildren and their teachers in a scientific process using an online environment. The software required to support this scientific process has been continuously evolving in response to user testing. The complexity of the user interactions during the engagement within the scientific process has necessitated the implementation of complex integration and communication links between the various process support tools in the VLE. This paper has provided a high level user-centered model of the integration and communication necessary to support the various stages of the VTIE process from which more detailed models will emerge. The modeling of the integration and communication links between the tools within the VTIE VLE will provide the EVE research team with a valuable representation from which to base future development. Finally, the integration of the VTIE process support tools has served to illustrate the importance of applying sound software engineering practices to the development of a complex learning environment.

References

[1.] Hoban, S., desJardins, M., Farrell, N., Rathod, P., Sachs, J., Sansare, S., Yesha, Y., Keating, J. G., Busschots, B., Means, J., Clark, G., Mayo, L., & Smith, W. (2002). Virtual Telescopes in Education. *Journal of Digital Information*, 2(4), Article No. 104, 2002-05-23.

[2.] B. Busschots, L. Raeside, J. G. Keating (2005). Formative evaluations of a virtual learning environment for Astronomy education. *Accepted Journal Paper: Computers and Education*

[3.] Hoban, S., Keating, J. G., Sachs, J., Laughlin, D., & Yesha, Y. (2003). Science Investigation System for Telescopes in Education Research. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Track 1 - Volume 1, p35.1.

[4.] B. Busschots, L. Raeside, S. Waddington, J. G. Keating (2006). The VTIE collaborative writing environment. *Proceedings of the 2nd International WEBIST Conference on Web Information Systems and Technologies,* Volume 2, P. 221-228

[5.] L. Raeside, B. Busschots, E. O'Cinneide, S. Foy, J. G. Keating (2005). Empowering schoolchildren to do astronomical science with images. *Proceedings SPIE* Vol. 5827, P. 669-678

[6.] getfirefox.com (2005). The Firefox web browser, http://www.getfirefox.com. Available online. Accessed on the 26th of March 2006

[7.] w3.org (2006). World Wide Web Consortium, http://www.w3.org. Available online. Accessed on the 5th of May 2006

[8.] apache.org (2005). The Struts Framework, http://www.apache.org/struts. Available online. Accessed on the 26th March 2006

[9.] javasoft.com (2005). The Model View Controller Model 2 Design Pattern, http://www.javasoft.com. Accessed on the 26th of March 2006

[10.] S. Speilman (2003). The Struts Framework. *Morgan Kaufmann*

[11.] xulplanet.com (2006). Advanced Javascript and XML, http://www.xulplanet.com. Accessed on the 26th of March 2006

[12.] jdom.org (2006). The JDOM XML solution, http://www.jdom.org. Accessed on the 26th of March 2006

Review of Assistive Devices for Electric Powered Wheelchairs Navigation

K.Arshak, D.Buckley, K. Kaneswaran
Department of Electronic Engineering
University of Limerick, Limerick, Ireland
Khalil.Arshak@ul.ie, Denis.Buckley@ul.ie, Kelly.Kaneswaran@ul.ie

ABSTRACT

The decreasing costs of microprocessor systems and increasing range of "Smart Sensors" have led to a boom in Assistive Device Technology. The annual rate of expenditure for mobility related devices has reached $1 billion dollars in the United States alone. The industries current focus is to develop a wider range of Independent Mobility Controllers to allow, even the most severely disabled person, the ability to control an Electric Powered Wheelchair (EPW). Advances in Autonomous Robot Design have led to corresponding improvements in EPW technology. This paper outlines user interfaces and input device technologies used at present to navigate an EPW.

Keywords: Autonomous, Input Devices, Electric Powered Wheelchairs, Rehabilitative, Assistive Devices, Robot Navigation.

1. INTRODUCTION

Over the years advances in medicine have led to an increase in life expectancy and changed the focus of medical care from acute treatment to long-term rehabilitation of patients. This has generated a greater demand for more advanced wheelchairs, in specific, powered wheelchairs, encouraging developments in Assistive Device Technology (ADT). In 1994 there were over 54 million people with disabilities in the United States alone representing nearly one fifth of the American population. Out of this figure almost 1.8 million required the use of a wheelchair [1].

A clinical survey in 2000 [2] of 200 practicing clinicians in the U.S. showed that a vast amount of their patients rely on joystick, sip-and-puff, chin or head control devices for navigation. However, 40% of these patients found steering and maneuvering tasks difficult or impossible. This paper outlined the requirement for more "independent mobility controllers"; controllers which allow even the most severely immobile person the ability to navigate a powered wheelchair. This enables them to implement day to day tasks without the intervention of carers, hence increasing their autonomy and experience of life. Over the last decade there have been many attempts to design an Autonomous EPW System [3]-[15]. These attempts have benefited from the decreasing cost of microcontrollers and increasing range of readily available smart sensors. The introduction of new ADT's such as Global Positioning Systems (GPS) [16], Electroencephalogram (EEG) [17] and Gesture control systems [18], in the last five years, gives new hope to those users who previously could not use an EPW. At present, many of these new ADT's are in the experimental stages of design. Therefore this

review is intended to highlight the currently available ADT's and provide an insight into future designs.

2. DISABILITIES

The three main types of disabilities are Physical, Sensory and Emotional. A disabled person may have one or more of these disabilities but it is mainly physical disabilities which cause motor function impairment and necessitate the need for ADT equipped wheelchairs. A physical disability limits basic physical activities such as walking, climbing stairs, reaching, lifting or carrying. There are several physical disabilities/conditions which require the use of EPW including brain injury, stroke, fractures, amputation, pulmonary disease, neurological disorders, musculoskeletal diseases/injuries and spinal cord injuries.

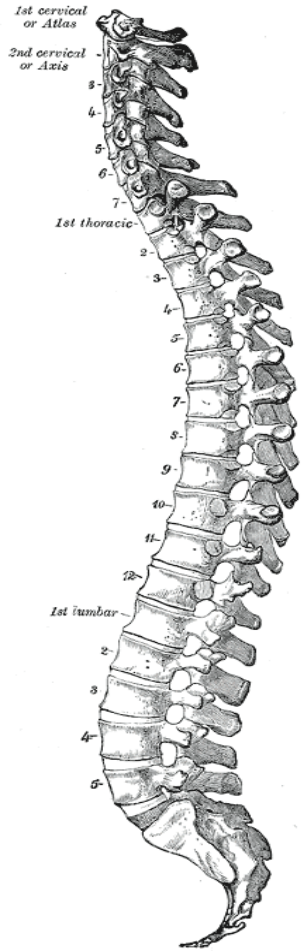

Figure 1 Human Spinal

Complete spinal cord injury branch levels and candidate input devices

C1	C2	C3	C4	C5	C6	C7	C8	T1
Eye tracker								
Eye wink switch								
Brain wave								
Facial movement								
Bite switch								
Tongue joystick								
Tongue switches								
Head switches – limited movement directions								
Head 6DOF position– limited movement directions								
Chin switch								
Speech								
Muscle EMG (available muscles)								
		Sip-puff switch (diaphragmatic)						
		Chin joystick						
		Head mouse						
		Mouth stick						
		Shoulder switches						
		Shoulder 6DOF position						
				Elbow flexion switch (assisted return)				
				Shoulder joint – upper arm switches				
				Shoulder joint – upper arm 6DOF position				
					Wrist extension switch (assisted return)			
					Mouse			
					Trackball			
					Joystick			
					Glidepoint			
					Touchscreen			
					Alternative keyboards			
						Wrist switches		
						Wrist position		
						Hand 6DOF position		
						Thumb switch		
						Thumb position		
						Standard keyboard		
						Finger switch		
						Finger position		
						Tablet / Pen		
						Spaceball		
						Space glove		

Figure 2 Spinal cord injury and candidate input devices [20]

column [19]

The Spinal Cord is the largest nerve in the body and is responsible for carrying motor communication to the muscles. The spinal cord runs through the spinal column and consists of 8 cervical (neck), 12 thoracic (upper back), 5 lumbar (lower back), and 4 sacral (base of the spine) branches (Figure 1). The number of the branch, running from cervical to sacral, and a letter indicating the region are used to denote each nerve branch throughout this paper. Figure 2 depicts a map of the spinal cord injury branches associated with possible candidate ADT's. Spinal cord injury patients require expert analysis from occupational therapists and rehabilitation engineers when deciding which ADT will be most suitable for controlling an EPW.

3. ASSISTIVE DEVICE TECHNOLOGIES

The main components of an Electric Powered Wheelchair can be categorised as Inputs, Control System and Outputs as illustrated in Figure 3 below. Assistive Device Technologies are categorised as inputs to Electric Powered Wheelchairs control systems.

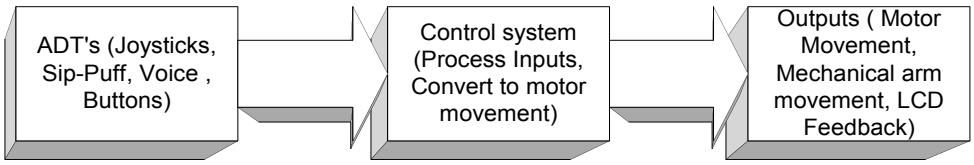

Figure 3 Components of Wheelchair

Figure 4 illustrates the various areas of the human body which can be used to control an EPW. A joystick is the most common control device for an EPW and unless specifically requested is normally supplied with all wheelchairs. The most commonly used joysticks are tongue, chin, finger and hand .The main types of joystick are switched and proportional. The Switched Joysticks act in a unidirectional manner converting electrical contact signals into discrete positioning information. They usually implement four switches giving a total of eight discrete positions, four by activating a single switch and four by combined activation of switches.

Proportional Joysticks get their name from the (proportional) movement of the joystick resulting in a proportional output signal, due to resistive or inductive change, which indicates the position of the joystick. Another type of proportional joystick is the Force Feedback Joystick [22]. Sensors are used to detect obstacles in the wheelchairs path, a force is applied to the joystick lever in opposition to the wheelchairs direction so that the user may avoid the obstacle. Results have shown that these Joysticks enable the user to improve manoeuvrability with fewer collisions in their environment [23].

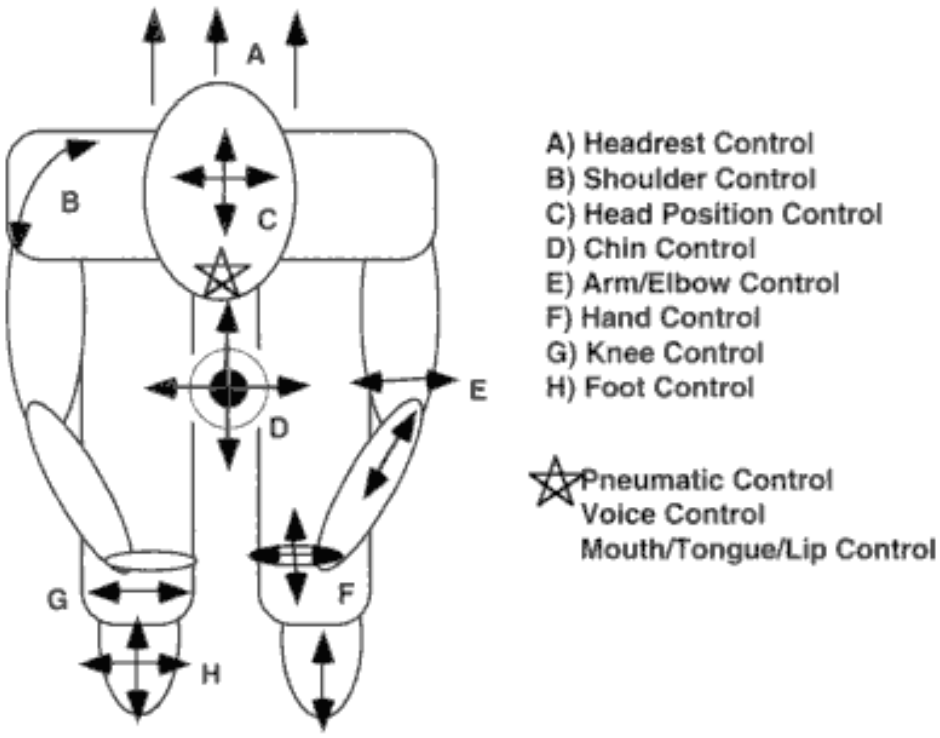

Figure 4 Possible anatomic sites on the body which can be used to control an EPW [21], © DEMOS Publishing.

Arthritis is one of the major reasons for wheelchair use in elderly patients [24]. Constant use of Switched/ Proportional Joysticks and repeated wrist movements can be very painful for an arthritic patient, and may prove difficult for elderly patients. Isometric Joysticks [25] use Force Sensing Resistors (FSR) to sense the force of the joystick in any given direction. Isometric Joysticks do not utilise spherical or orthogonal pivots on the base of the stick so there is minimal movement of the wrist and upper body. This has the advantage of reducing muscle movement normally encountered with the operation of Proportional Joysticks while also reducing motion triggered reflexes, such as spasticity. Many of the switched, proportional and isometric joysticks are implemented digitally which allows seamless integration with digital controller systems often used in EPW's. This improves immunity to interference from phones, citizen band (CB) radio and even electric motors of the wheelchair.

For motor function impairments that limit upper and lower body movement, a combination of devices are used. Headwands (Figure 5) usually consist of a pointer or extension device that is mounted to a head piece and extends from the centre of the forehead angling downwards In a similarly fashion, a stick which is controlled by the mouth is used and is known as a Mouthstick. Headsticks and Mouthsticks are usually used in direct selection of an object such

as a key on a keyboard or a symbol/word on a board. They are primarily for use by persons with good head control.

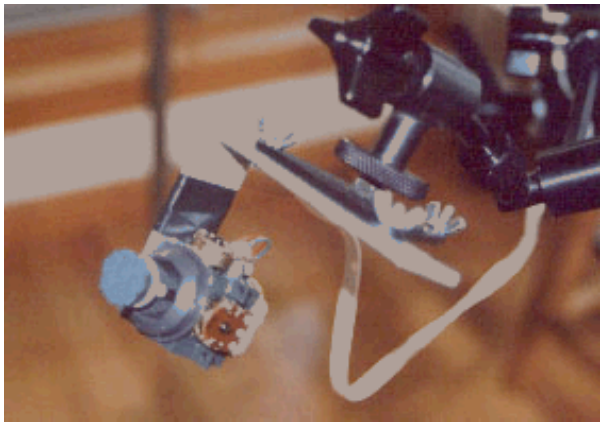

Figure 5 Headstick (Left) [26].

They are useful when it comes to everyday tasks like turning on a light or typing a message but they are not so useful for controlling an EPW's directional movement, because while the user is concentrating on position of the stick they are distracted from the path in front of them leading to increased collisions. Therefore Headsticks/Mouthsticks are primarily used as an accessibility device for seating position and computer input.

Chinsticks (Figure 6) can be used to directly control the navigation of a wheelchair, they consist of a small joystick mounted on a mechanical arm under the users chin. Sip/Puff devices (Figure 7) are widely used for controlling EPW's. A Sip/Puff switch is head mounted (Figure 8) and used to actuate a two-position switch by a simple sip or puff (pulling and pushing of air through a tube). It consists of a head frame with a switch box and replaceable plastic mouth tube attached. Sips and puffs are converted to switch closures inside the Switch Box.

Figure 7 A Sip/Puff system [28].

Figure 8 Head Mounted Sip/Puff device [29]

A serious of sips can determine the direction the operator wants the chair to travel while the puff confirms the choice. They are commonly used in conjunction with a Liquid Crystal Display (LCD) so that the user can navigate using a computerized menu system and obtain feedback. This maybe the only device which is operated without movement of the upper and lower limbs but again it has its pitfalls as the user must have strong respitory control.

Switches are also used to control directional movement of EPW's. The control scheme is similar to that of the Switched Joystick and is nearly always used with a Switch Box Controller (Figure 9). Nearly every limb on the body can be used to activate a switch (see Figure 4). There are different types of switches namely non-contact, contact, hit, and target. Non contact switches work by detecting movement; accelerometers and mercury switches are used extensively for detection of limb movement. Contact switches work by touching, pressing or squeezing the switch and can be capacitive, inductive or resistive. Hit switches are usually the type which need to be depressed to work, are possibility mounted on a spring and are butted or smacked to operate and have a higher tolerability for the force placed on them. Most contact switches are target switches were the user has to reach out and touch the switch.

Pressure type switches are also used to determine pressure in certain areas of the patient's seating to determine directional movement of EPW. There are several examples of two way switches/transducers used as partial control in EPW and they are usually placed according to the anatomic site which the patient has most precise control over.

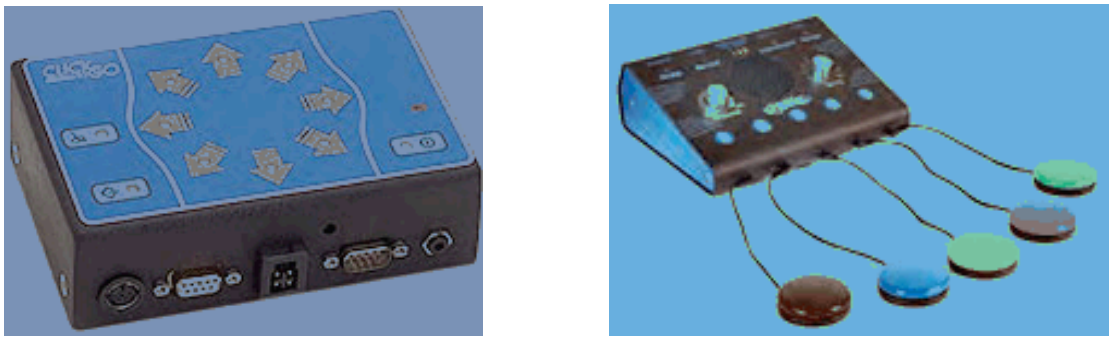

Figure 9 Switch Box Controllers for EPW Navigation [30]

For example, if the operator has stronger control over an anatomic site like the foot then switches can be mounted around the foot to control movement of the chair. The foot can also be used to control a pedal similar to a pedal on a bicycle or an accelerator in a car. This can be used to control speed or direction. Accelerometers have successfully been used on C5/C6-Level quadriplegic patients, on the elbow, as an access device on wheelchairs [31]. The shoulder has been used as a control device since the early seventies, by using dual-axis transducers researchers have successfully tested the use of shoulder movement as a control device in C4 and C5 Level patients [32]. The head can be used in several ways as a control device. With C5/C6 patients switches or Ultrasonics [33], mounted in the headrest can be used to track left and right head movement. While a joystick attached to the headrest detects forwards and reverse movement. Head tilt on a two dimensional axis can be monitored using accelerometers to control direction. There are also examples of wrist, blink and bite switches commercially available on the market.

Electropalatography [34] [35] is an instrumental technique for determining tongue/palate contact during speech. The technique utilizes an artificial palate with electrodes embedded in its tongue-facing surface. Each palate is made to fit the subject and normally requires a simple dental impression and subsequent fitting. The electrodes on the palate are connected to an electronics unit, which collects contact data from the palate and passes it on to a computer. The computer displays the contact patterns, either in real time, or off-line for analysis. This same technique was being used to control wheelchairs from as early as the nineties. The user

can adapt any number of electrodes on the palette to control the direction and speed of the wheelchair. This type of interface is advantageous over others as the tongue does not suffer from fatigue like other limbs and can be controlled very accurately.

Speech control [36] is another useful way of controlling automation of EPW's. Speech systems are usually used with higher level spinal injury patients C1 – C4. The EPW's is configured with a built in speech recognition system which recognizes a small vocabulary of words used to control directional movement e.g. forward, back , left, right etc [37][38]. Some more advanced systems use speech recognition to recognize all spoken words so that the user may use it for navigation and communication through computer systems e.g. email, documents [39]. Another method is to use Electropalatography to determine tongue movements on spoken words, such as directions, and process the movements with a pattern recognition system so that the system will recognize the pattern and implement the directional movement.

Facial expression (Gesture) tracking systems [40] are a relatively new way of controlling EPW. The chair is mounted with a camera and computer system. The camera monitors the patients face and hands for gestures, it uses the face to recognize the patient, and locate the patient hands so not to mix them up with other patient's hands. Other examples include, face and gaze tracking systems [18], which monitors the position of the head and the area around the eye to see if the patient is staring in a certain direction. The chair will then go in that direction after confirmation from the user by nod or shake of the head. An advantage of this system is that the chair can monitor their surroundings and by using a nod of the head the chair can automatically be called to the patient's side. Electrooculography [41] is a novel method which uses electrodes placed around the eye to detect the position of the eyeball in its ocular cavity giving the user the ability to control the chair by eyeball movement. Fiber-optic [42] technology has also been used for monitoring the movement of the patients cheek, a headset around the ears which monitors the movement of the patients cheek checking for a wink like movement, this can simulate the left of right clicking of a mouse or be used for directional control.

4. FUTURE TECHNOLOGY

There have been several new advances in access device technology although the trend is in designing of autonomous system which requires little patient input. Researchers have had significant results in using Electroencephalogram (EEG) [43] to control wheelchair direction. This involves monitoring of the brains electrical signals and a neural network to learn patterns resulting from moving of a joystick. This is an area which shows great potential as to be able

to control a chair by thought would reduce all levels of muscle fatigue and could be used widely across all levels of spinal injury patients. But as yet this is still in the experimental stage.

Capacitive and Magnetic technologies are being realised at the University of Limerick, Ireland, as possible replacements for the two dimensional switched and proportional joysticks. They require little movement from the user and act similar to that of a touch pad system on a laptop. They are inexpensive methods of tracking limb movement for use in control of EPW Navigation. They require low power consumption and can readily be adapted to any analogue or digital controller systems.

There has been significant work on integrated controllers for EPW which has led to a new ISO standard being adopted (ISO/TC-173/SC-1/WG-7). The Multiple Master Multiple Slave (M3S) [44] initiative is a European Union (EU) effort to develop a standard interface between the various assistive devices so as to make all devices plug and play and give the user increased independence and control over the configuration of the EPW.

5. CONCLUSION

As electronic devices continue to drop in price the use of Switched/Proportional Joysticks will decline with the trend lending towards Isometric Joystick Systems. Since the inclusion of micro-controller hardware, the capabilities of EPW have increased rapidly, extending the range of assistive devices on the market today. New technology has made it possible for users to fine tune the assistive devices used to control their EPW's without the need for carer intervention. Future developments will see inclusion of remote monitoring of EPW systems through onboard modems, reducing servicing time for the manufacturer and patient. Furthermore the advances in rehabilitive technology will give more flexible control over Electric Powered Wheelchair Systems.

6. REFERENCES

1. McNeil J.M., Americans with Disabilities: 1994-1995, *Data From the Survey of Income and Program Participation,* P70-61.
2. M. W. E. L. Linda Fehr, Phd;Steve B.Skaar,Phd, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey," Journal of Rehabilitation Research and Development, vol. 37, pp. 353 - 360, 2000
3. R. C. Simpson and S. P. Levine, "Automatic adaptation in the NavChair Assistive Wheelchair Navigation System," *IEEE Transactions on Rehabilitation Engineering*, vol. 7, pp. 452-463, 1999.
4. M. I. Bresler, "- The Tamara system for control of mobility, communication, and Enviorment.," pp. 121-123, 1992.
5. R. Madarasz, L. Heiny, R. Cromp, and N. Mazur, "- The design of an autonomous vehicle for the disabled," vol. - 2, pp. 117 -126, 1986

6. K. B. Stanton, P. R. Sherman, M. L. Rohwedder, C. P. Fleskes, D. R. Gray, D. T. Minh, C. Espinoza, D. Mayui, M. Ishaque, and M. A. Perkowski, "- PSUBOT-a voice-controlled wheelchair for the handicapped," vol. -, pp. 669 - 672 vol.2, 1990.
7. G. Pires, N. Honorio, C. Lopes, U. Nunes, and A. T. Almeida, "- Autonomous wheelchair for disabled people," , pp.797 - 801 vol.3, 1997
8. P. Beattie, "- SENARIO: SENsor Aided intelligent wheelchaiR navigatIOn," vol. -, pp. 2/1-2/4, 1995.
9. T. Gomi and K. Ide, "- The development of an intelligent wheelchair," vol. -, pp.70 - 75, 1996.
10. N. I. Katevas, N. M. Sgouros, S. G. Tzafestas, G. Papakonstantinou, P. Beattie, J. M. Bishop, P. Tsanakas, and D. Koutsouris, "- The autonomous mobile robot SENARIO: a sensor aided intelligent," vol. - 4, pp. 60 - 70, 1997.
11. G. Pires, R. Araujo, U. Nunes, and A. T. de Almeida, "- RobChair-a powered wheelchair using a behaviour-based navigation," , pp. 536 - 541, 1998.
12. S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, and J. Borenstein, "- The NavChair Assistive Wheelchair Navigation System," vol. - 7, pp. 443 - 451, 1999.
13. A. Roberts, W. Pruehsner, and J. D. Enderle, "- Vocal, motorized, and environmentally controlled chair", pp. 33 - 34, 1999.
14. S. Vicente Diaz, C. Amaya Rodriguez, F. Diaz Del Rio, A. Civit Balcells, and D. Cagigas Muniz, "- TetraNauta: an intelligent wheelchair for users with very severe mobility restrictions," vol. - 2, pp.778 - 783, 2002.
15. G. Bourhis and P. Pino, "- Mobile robotics and mobility assistance for people with motor," vol. - 4, pp. 7 - 12, 1996.
16. M. Imamura, R. Tomitaka, Y. Miyazaki, K. Kobayashi, and K. Watanabe, "- Outdoor waypoint navigation for an intelligent wheelchair using differential GPS and INS," vol. - 3, pp.2193 - 2196, 2004.
17. K. Tanaka, K. Matsunaga, and H. O. Wang, "Electroencephalogram-Based Control of an Electric Wheelchair," Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on], vol. 21, pp. 762-766, 2005.
18. Y. Matsumotot, T. Ino, and T. Ogsawara, "- Development of intelligent wheelchair system with face and gaze based interface", pp.262 - 267, 2001
19. Image of Human spinal Column from wikimedia, http://upload.wikimedia.org/wikipedia/commons/8/83/Gray_111_-_Vertebral_column.png
20. Bates, R., Computer Input Device Selection Methodology for Users with High-Level Spinal Cord Injuries - Proceedings of the 1st Cambridge Workshop on Universal Access and Assistive Technology (CWUAAT), Trinity Hall, University of Cambridge 25th-27th March 2002"
21. R. A. Cooper, "-Wheelchair Selection and Configuration," pp. 121 -126, 1998.
22. A. Fattouh, M. Sahnoun, and G. Bourhis, "- Force feedback joystick control of a powered wheelchair: preliminary study", pp. 2640 - 2645, vol.3, 2004.
23. eJames L. Protho, M. S., B. S. Edmund F. LoPresti, et al. (2000). "An Evaluation of an Obstacle Avoidance Force Feedback Joystick." The Proceedings of the Annual RESNAConference, Orlando, FL: 447-449.
24. Pickles B., Topping A., "Community Care for Canadian Seniors: An Exercise in Educational Planning", Disability and Rehabilitation, vol. – 3, pp. 181-189, 1994.
25. R. A. Cooper, D. M. Spaeth, D. K. Jones, M. L. Boninger, S. G. Fitzgerald, and S. Guo, "Comparison of virtual and real electric powered wheelchair driving using a position sensing joystick and an isometric joystick" Medical Engineering & Physics, vol. - 24, pp. 703 - 708, 2002.
26. HeadStick image from AT Soloutions, http://www.atsolutions.org/images/headwand_small.jpg
27. Chinstick image from Pulsar, http://www.pulsar.org/2k/images/widgits/chinstick1.jpg
28. Sip/puff control system image from M.I.M.E. Senior Design Clinic Spring 2002, http://www.mime.eng.utoledo.edu/ design_clinic/design_ Expo/ Spring02Pages/2002-01-08/sip_puff2.html
29. Image of Head mounted Sip/puff device from Origin Instruments Corporation, 2002-2004, http://www.orin.com/ dealer_press_page/sip-puf_head.jpg.
30. Images of Switchbox controllers from Click2Go, http://www.click2go.ie
31. P. E. Crago, W. D. Memberg, M. K. Usey, M. W. Keith, R. F. Kirsch, G. J. Chapman, M. A. Katorgi, and E. J. Perreault, "An elbow extension neuroprosthesis for individuals with tetraplegia," Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation], vol. 6, pp. 1- 6, 1998.

32. J. J. Sarver, B. T. Smith, R. Seliktar, M. J. Mulcahey, and R. R. Betz, "A study of shoulder motions as a control source for adolescents with C4 level SCI," Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation], vol. 7, pp. 27-34, 1999.
33. D. L. Jaffe, "An ultrasonic head position interface for wheelchair control," Journal of Medical Systems, vol. 6, pp. 337-342, 1982.
34. Wolfgang Nutt, Christian Arlanch, S. Nigg, and G. Staufert, "Tongue-mouse for quadriplegics," J. Micromech. Microeng, vol. 8, pp. 155-157, 1998.
35. M. E. Tyler, D. Kim, and D. J. Beebe, "Development of a Tongue-Operated Switch Array as an Alternative Input Device", International Journal of Human-Computer Interaction, vol. 18, p. 19, 2005.
36. K. B. Stanton, P. R. Sherman, M. L. Rohwedder, C. P. Fleskes, D. R. Gray, D. T. Minh, C. Espinoza, D. Mayui, M. Ishaque, and M. A. Perkowski, "- PSUBOT-a voice-controlled wheelchair for the handicapped,", pp. 669 - 672 , vol.2, 1990.
37. K. Komiya, K. Morita, K. Kagekawa, and K. Kurosu, "- Guidance of a wheelchair by voice," vol. - 1, pp. 102 - 107, 2000.
38. R. C. Simpson and S. P. Levine, "- Adaptive shared control of a smart wheelchair operated by voice control,", pp. 622 – 626, 1997.
39. H. R. Singh, A. Mobin, S. Kumar, S. Chauhan, and S. S. Agrawal, "- Design and development of voice/joystick operated microcontroller," vol. - 2, pp. 1573 - 1576, 1999.
40. Y. Kuno, T. Murashima, N. Shimada, and Y. Shirai, "- Interactive gesture interface for intelligent wheelchairs," vol. - 2, pp. 789 - 792, 2000.
41. R. Barea, L. Boquete, M. Mazo, and E. Lopez, "System for assisted mobility using eye movements based on electrooculography," Neural Systems and Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Rehabilitation Engineering], vol. 10, pp. 209-218, 2002.
42. R. Marsden, "100% Hands-free Computer Access- Madentec's 2000 Series," California State University, Northridge Center on Disabilities' 20th Annual International Technology and Persons with Disabilities Conference, 2000.
43. H. Lakany, "- Steering a wheelchair by thought," The IEE International Workshop on Intelligent Environments, pp. 199 - 202, 2005.
44. S. Linnman, "- M3S: the local network for electric wheelchairs and rehabilitation," vol. - 4, pp. 188 - 192, 1996.

SoC Test: Trends And Recent Standards.

Michael Higgins and Ciaran MacNamee.
Circuits and Systems Research Centre (CSRC),
University of Limerick.
Email: *Michael.Higgins@ul.ie*

Abstract

The well-known approaching test cost crisis, where semiconductor test costs begin to approach or exceed manufacturing costs has led test engineers to apply new solutions to the problem of testing System-On-Chip (SoC) designs containing multiple IP (Intellectual Property) cores. While it is not yet possible to apply generic test architectures to an IP core within a SoC, the emergence of a number of similar approaches, and the release of new industry standards, such as IEEE 1500 and IEEE 1450.6, may begin to change this situation. This paper looks at these standards and at some techniques currently used by SoC test engineers. An extensive reference list is included, reflecting the purpose of this publication as a review paper.

1. INTRODUCTION.

With the recent approval and acceptance of the IEEE 1500 standard for the test of core-based integrated circuits and the IEEE 1450.6 standard for CTL (core test language), it is an appropriate time to review what techniques and standards are currently being used by industry including some that are compatible with these two IEEE standards. Section two of this paper documents a selection of the different types of ATE (Automated Test Equipment) that are available and how these ATE approaches are influenced by cost and functionality. Section three describes some SoC test architectures that have been used by industry in advance of the publication of the IEEE 1500 standard. TESTRAIL and AMBA focus on TAM (Test Access Mechanism) development, whereas TESTSHELL and TESTCOLLAR focus on core test wrapper development. It is also shown how the ETM10 by ARM has had the IEEE 1500 wrapper built around its core. Software tools and test vector compression techniques that have been used are presented in section four, which include schemes such as IBMs STUMPS, Philips TR-Architect, SmartBIST and IEEE 1450.6. Section five gives a general overview of embedded memory test. Embedded memory test is in itself a challenge and some approaches to its testing are described depicting its significance in the overall SoC test area. The embedded memory test section includes techniques such as: Fault Modelling, BIST (Built In Self Test), BISR (Built In Self Repair) and image processing techniques. Section six briefly outlines examples of how industry has already adopted the IEEE 1500 and IEEE 1450.6 standards.

2. CURRENT ATE APPROACHES.

ATE systems can be broadly categorised into three types: Conventional '*Big Iron*' testers (typically costs above $1,000,000), '*Middle Iron*' testers (costs range between $399,000 and

$1,000,000 [1]), and Low cost DFT focused testers (Teseda's V500 approximate cost is $60000 [2]). To understand the role of an ATE it is useful to understand the difference between structural and functional testing. Structural testing involves developing a set of test vectors to detect specific faults that may have been introduced in the design by errors such as DSM (Deep Sub Micron) effects and processing defects. By applying specific test vectors to a circuit, and then capturing and comparing the actual responses against the expected responses is what is known as functional testing**.** '*Big Iron*' testers are designed to test the device functionally; at speed functional testing is performed at every pin. An example of a '*Big Iron*' ATE is the Teradyne Catalyst [3]. '*Middle Iron*' testers combine structural and functional testing. Some tradeoffs in speed, performance or flexibility are made with '*Middle Iron*' testers in comparison to the '*Big Iron*' testers such that overall ATE cost is reduced. An example of a '*Middle Iron*' tester is the Agilent 93000 [4]. Low cost DFT (Design For Test) focused testers have limited or no functional test capabilities, they are designed to support test methodologies such as boundary scan or BIST. This type or architecture does not have as high a level of flexibility and accuracy as the '*Big Iron*' and '*Middle Iron*' testers but can have a lower cost due to its reduced flexibility and accuracy. The software of a DFT focused tester can play a more pertinent role than the hardware itself. In the case of Teseda's V500 [5], IEEE 1450 STIL (Standard Test Interface Language) is used which provides an interface between digital test generation tools and test equipment.

Multi-site testing is an effective and popular way to increase throughput and reduce the cost of test [6]. Multi-site testing is described as testing multiple instances of the same SoC in parallel on a single ATE. In addition to the reduction of test cost by multi-site testing it has also been noted that increasing the vector memory depth is more cost effective than increasing the number of ATE channels [6]. The Tiger test system by Teradyne takes advantage of the benefits of multi-site testing by incorporating multi-site test for complex devices. The economic advantages of combining multi-site testing with reduced pin-count test, low channel cost ATE and bandwidth matching are explored by [7] where it is found that the multi-site test approach had more benefits than the single-site test approach.

A method is described by [8] where ATE and EDA (Electronic Design Automation) tools are linked to identify and diagnose failures at layout level. In this approach the ATE and EDA tools share the EDA IC design database and ATE failure datalog to determine a failure and then identify the location and cause of the failure.

STEPS (Software-based Test Environment for P1500 compliant SoCs) [9] is based on the concept that the ATE is not considered as an initiator applying vectors to SoC test pins but a target comprising of a bank of 32 bit test data and control commands. Using STEPS the SoC

can act as the active component and the ATE as the passive component. STEPS has the advantage that the test program is executed at the system speed and is not limited by the ATE frequency.

3. SOC TEST ARCHITECTURES.

a) IEEE 1500

The IEEE 1500 standard was approved in late March 2005 after the IEEE P1500 SECT was started ten years earlier in 1995. The scope of the standard defines a mechanism for the test of core designs within a SoC. This mechanism constitutes a hardware architecture and leverages the CTL to facilitate communication between core designers and core integrators [10]. The IEEE 1500 has two levels of compliance: a wrapped and unwrapped compliance. The wrapped compliance caters for a core that comes with an IEEE 1500 wrapper function and a CTL program. The unwrapped compliance refers to a core that does not have an (complete) IEEE 1500 wrapper but does have a CTL description. The IEEE 1500 is independent of the functionality of the IC or the embedded cores. The IEEE 1500 is a new standard, but it has been widely anticipated and discussed for a number of years, so this paper attempts to look at some other industrial SoC test techniques and attempts to outline their levels of compatibility with the IEEE 1500.

b) Philips TESTRAIL

The scalable and flexible TAM TESTRAIL developed by Philips can provide access to one or more cores. A SoC may contain more than one TESTRAIL each of varying bandwidth determined by the width of the TESTRAIL [11]. The TESTRAIL approach attempts to combine both the strengths of TESTBUS and BST (Boundary Scan Test) [12]. BST is the IEEE 1149.1 standard for test access port and boundary scan architecture. The TESTBUS architecture allows the cores under test to be directly accessed from the pins of the IC. The TESTBUS approach can have one or more TESTBUSs per SoC similar to TESTRAIL so that tradeoffs can be made between silicon area and test time [12]. Similar to BST, multiple cores can be daisy-chained into one TESTRAIL. The TESTRAIL architecture is therefore a combination of daisychain and distribution architectures. A daisychain architecture can be achieved using only one TESTRAIL, whereas a distribution architecture can be implemented using more than one TESTRAIL where each TESTRAIL acts independently [13]. A TESTRAIL example is shown in figure 1. Core A has a private TESTRAIL of 16 bits, while the TESTRAILs both 16 bits of core B and C is concatenated. The two TESTRAILs from core A and from core B and C are then multiplexed back onto one single 16-bit TESTRAIL. Figure 1 is an example of the flexibility of TESTRAIL.

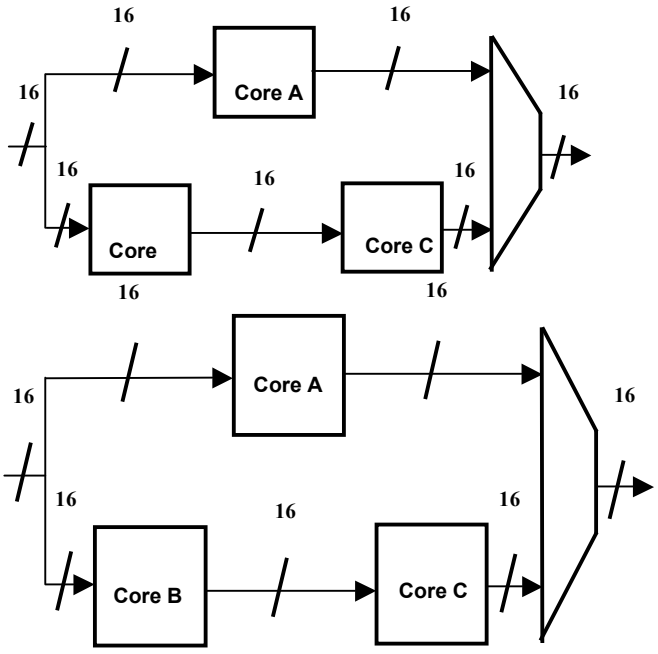

Figure 6: TESTRAIL Example Architecture [11, 14]

c) TESTSHELL & TESTCOLLAR

Other hardware approaches that have been used to test SoCs are TESTSHELL [14] and TESTCOLLAR [15]. These two approaches are considered similar to the recently approved IEEE 1500 due to the scalability of their TAMs [12]. A TESTSHELL example used in conjunction with TESTRAIL is shown in figure 2. TESTSHELL is an interface layer between the piece of IP (Intellectual Property) and the host environment. This interface layer wraps the piece of IP and has four modes of operation: function, IP test, interconnect test and bypass [14].

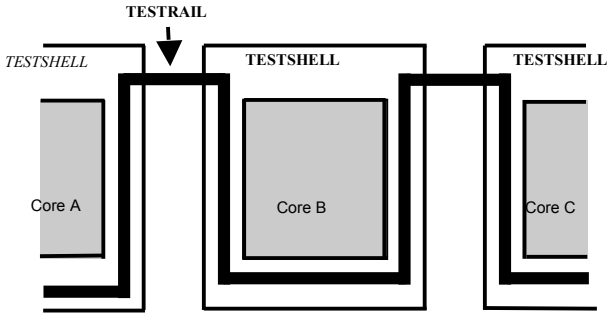

Figure 7: TESTSHELL used with TESTRAIL[12]

TESTCOLLAR cells are added to pieces of IP for the primary purpose of isolation; isolating the IP from the rest of the system to enable test or isolating the UDL (User Defined Logic) from the rest of the system so that it and its interconnects can be tested [15]. The structure of the TESTCOLLAR cell is shown in figure 3. There are three basic types of TESTCOLLAR

cells: combinational, latched and registered. Full implementation of TESTCOLLAR at the input and output of each component provides a double level test structure. Providing a "full featured" TESTCOLLAR around each component can result in a double-level test structure, i.e. observability and controllability. This double level test structure can lead to a higher cost for the test structure implementation [15].

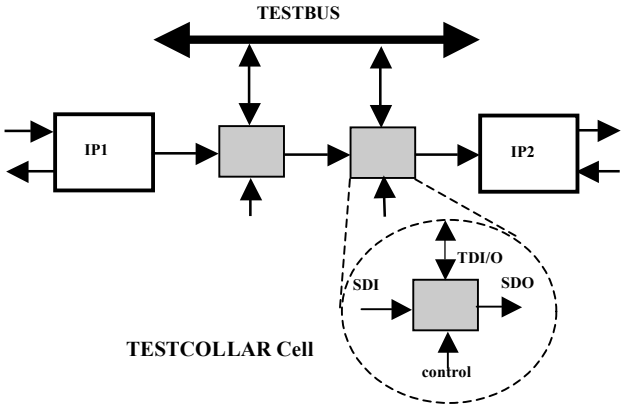

Figure 8: TESTCOLLAR [15]

d) ARM (Advanced RISC Machines)

Given that many SoC test applications require access to individual cores as well as the isolation of these cores, it would appear that the SoCs functional bus structure might be used to realise a workable TAM. Some test applications of the well-known AMBA (Advanced Micro-controller Bus Architecture) bus structure attempt to do this. An example [12] of ARM's AMBA system is shown in figure 4. 32 bit test vectors are passed from the IC pins to the core under test using the EBI (External Bus Interface).

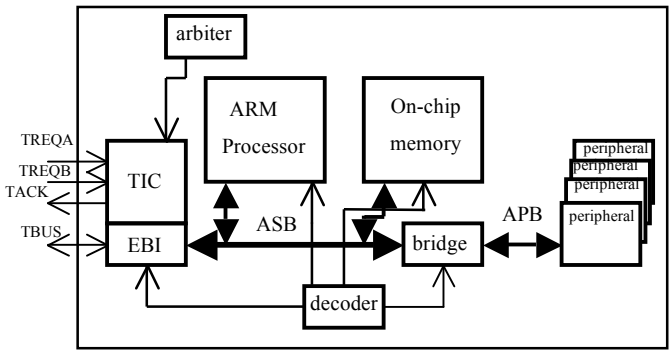

Figure 9: ARMs AMBA System [12]

A more recent approach by ARM to tackle the SoC test issue is applied to their ETM10 (Embedded Trace Macro) [16], ETM10 is the embedded trace macro for the ARM10 processor. This is a real time module that is capable of instruction and data tracing. During experimentation in 2002 by [16], the IEEE 1500 was applied to the ETM10 to implement it as

a full scan core. Figure 5 shows how the IEEE 1500 wrapper was built around the ETM10. A two-step approach was used to check the functionality of the IEEE 1500 wrapper. Firstly, test patterns were generated using an ATPG tool and then verified with a Verilog test-bench on the ETM10 without the WIR or WBY. Secondly the same patterns were applied with the WIR and WBY included in the design with the ETM10. The ETM10s test coverage was the same in each instance, showing that the IEEE 1500 test wrapper can be used without any degradation in test coverage.

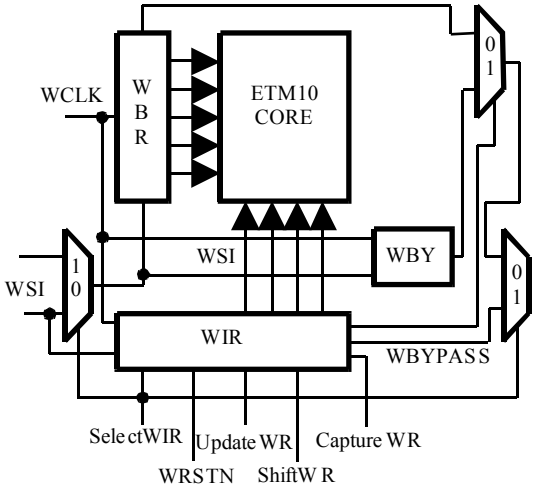

Figure 10: ETM10 with IEEE 1500 Wrapper [16]

4. SOFTWARE TOOLS AND TEST VECTOR COMPRESSION

Functional testing represented the first generation (1G) of IC test. The shift from functional test to scan test represented the second generation (2G). The multimillion gate SOC provides new challenges for the third generation (3G) of digital test [17].

Two critical challenges that test planning for SOC must address are: Handling the increase in test suite sizes ("*can we fit a new test suite on an existing ATE?*") and transporting test data to cores embedded deep within the system ("*Can we get test data to where we want it on chip and can we do it on time?*") [18]. The pin count is one of the main causes of speed degradation for test data transfer across the chip [19]. To reduce the number of test pins and memory size required for ATE, the test data that is transferred between the ATE and the SOC needs to be reduced. To reduce the test data, compaction and loss-less compression schemes can be used. The techniques that can be used in these compaction and compression schemes can be divided into three categories [20]:Vertical Compression (minimize the amount of test data per ATE pin), Horizontal Compression (reduce the number of ATE channels) and schemes that incorporate both vertical and horizontal compression. Horizontal compression

can be achieved by serialising the test data. Data could be loaded serially using only one test pin but this requires an increased memory depth and longer test time [20].

TAM design and test data compression offer promising solutions to the problem of ballooning test data volume, more complex ATE requirements and the challenge of transporting test data to the cores. In work conducted by [18] the use of data compression and TAM design is presented as an integrated approach to modular SoC test.

a) IBMs STUMPS

On-Product signature generation techniques are well known from Logic BIST. IBM's pioneering scan-based logic BIST is called STUMPS (Self Test Using a MISR (Multiple Input Signature Register) and Parallel SRSG (Shift Register Sequence Generator)), which uses a MISR at the outputs of product scan chains [21]. The OPMISR (On-Product MISR) solution intends to reduce the required number of ATE pins as well as the amount of test responses to send back to the ATE. Initially, the input and output circuit ports are merged into bi-directional ports. Additionally, an MISR is inserted on scan chain outputs. The scan vector signatures (compacted responses) are transmitted back to the ATE trough I/O ports instead of bit-by-bit responses [20].

b) SmartBIST

SmartBIST [22] is the name for a family of streaming scan test pattern decoders that are suitable for on-chip integration. SmartBIST is the second phase of a technology roadmap that combines the benefits of ATPG (Automatic Test Pattern Generation) and Logic BIST techniques for the cost effective testing of 100M+ gate chips. The first phase, called On-Product MISR or OPMISR, has already been implemented in the DFT and ATPG tools for selective use on very large ASIC chips. The use of an OPMISR essentially eliminates most of the data volume and solves some of the logic test throughput issues related to the test response data. SmartBIST is intended for very large and complex designs.

c) Linear Compression Schemes

Test vector compression schemes, as described by [23], that use only linear operations to decompress the test vectors are called linear decompression schemes. Linear decompression techniques exploit the unspecified (don't care) bit positions in scan test cubes (i.e. deterministic scan test vectors where the unassigned bit positions are left as don't cares) to achieve large amount of compression. Continuous flow linear decompressors are those that receive data from the tester in a continuous-flow manner i.e. every cycle. These operate very efficiently since they can be directly connected to the tester and they simply receive the data as fast as the tester can transfer it.

d) Philips TR-Architect

The TR-Architect tool has been developed by Philips, which is designed to generate a test architecture for SoCs that are more complex than just a handful of cores. (In the case of a SoC with relatively few cores, it may be simpler to develop the test architecture manually.) TR-Architect accepts two inputs: a SoC data file and a list of user options. The SoC data file consists of information about the SoC itself such as: the numbers of modules embedded in the SoC, the number of inputs, outputs, bi-directionals, test patterns and the number of scan chains and their lengths [24]. The user options list contains more information about the SoC and its properties. These can be categorised as follows: Total number of SoC test pins, Types of modules (hard/soft), External bypass per module, Test schedule type (serial/parallel), TAM type (test bus/test rail), Architecture type and Test cost [24]. The TR-Architect tool supports three types of architectures: Daisychain, Distributed and a Hybrid of the previous two as shown in figure 6.

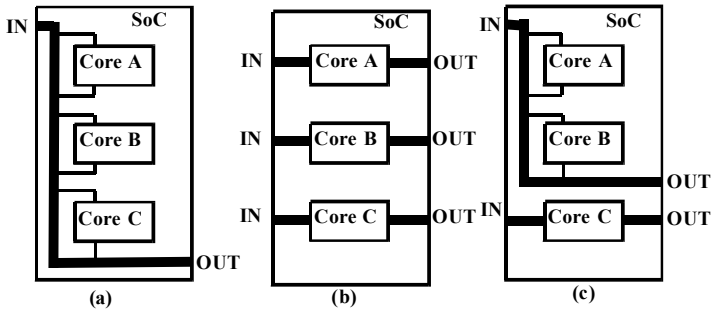

Figure 11: (a) Daisychain, (b) Distribution, (c) Hybrid [24]

e) IEEE 1450.6 CTL

The IEEE 1450.6 standard for Standard Test Interface Language (STIL) for digital test vector data – Core Test Language (CTL) has recently been approved. IEEE 1450.6 describes CTL, which has a close connection with the recently published IEEE 1500 standard for embedded core based test. CTL is a language for capturing and expressing test-related information for reusable cores, which is meant to co-exist with and complement information expressed as a netlist. CTL is an extension of IEEE 1450 STIL and is a software language that is targeted to SoC DFT. IEEE 1450.6 is used to describe IEEE 1500 wrappers. An appropriate TAM and wrapper can be designed using the CTL description of a core. The system integrator can test an embedded core and UDL around a core in a SoC using information that is supplied by the CTL description of the core provided by the designer. The bulk of the data in CTL is reusable without modification by using protocol statements from the traditional STIL. CTL is machine and human readable therefore allowing the CTL program to be used for documentation processes. This language is broad enough to describe 1500, VSIA (Virtual Socket Interface Alliance) and even IEEE 1149.1

[25]. It has been speculated by [26] that the IEEE 1450.6 could result in new and more powerful test optimisation capability and it has been noted that some IEEE 1450.6 tools have become available.

5. EMBEDDED MEMORY TEST

The ITRS (International Technology Roadmap for Semiconductors) 2001 speculates that embedded memories will dominate the majority of silicon area of a SoC (approximately 94%) by the year 2014. If this trend is to continue it is likely that embedded memory yield will worsen. The cost of memory testing increases with every new generation of memory chips [27]. Embedded memories have several advantages that include: improved performance, lower power consumption and overall cost. These advantages do have complications such as yield limitations, higher mask cost and an increased development complexity. Some of the strategies that are used to test embedded memories are introduced below.

a) Fault Modelling

Fault modelling is the translation of physical defects to a mathematical construct that can be operated upon algorithmically and understood by a software simulator to provide a quality measurement. One of the most common fault models is the Stuck At Fault (SAF) but there are many more. Static faults such as SAF and address decoder faults are sensitised by applying at most one operation. Dynamic faults take place in the absence of static faults, which require more than one operation to be performed sequentially in time so that they are sensitised. The majority of tests used in industry target specific faults and therefore may not detect dynamic faults [28].

b) BIST

BIST is considered to be one of the most cost effective solutions for embedded memory test [29]. The philosophy behind BIST is to let the hardware test itself. Although BIST is considered to be one of the more cost effective methods to test embedded memory, it will face challenges including: minimising the BIST overhead in both silicon area and routing, adhering to power budget constraints and support of different types of memory [28]. A new MBIST (Memory BIST) architecture is described by [30] which attempts to address some of the above challenges. PBIST (Programmable BIST) is described by [31] that targets specific faults in memory according to the user defined algorithm used. It is suggested by [32] that it is possible to programme the BIST circuit using an on-chip microprocessor that almost any SoC has incorporated into its design. This on-chip processor core can also be used to test other cores on the same chip.

c) BISR

BISR is used to enhance memory yield. Depending on redundancy and the BISR method used it is possible to increase yield by between 5% and 20% [33]. Repair is essential for present

and future memory technologies. The traditional way to perform memory repair is usually external test and repair. All known repair algorithms are not optimal and future schemes must consider practical issues including [28]: low hardware cost, test time reduction and 'on the fly' repair.

d) Hough Transform

Another strategy that has been investigated by [34] for the diagnosis of faults in embedded memories, is the use of an image processing technique; the Hough transform. The Hough transform is used to identify the most probable failure pattern among the set of possible ones provided in a fault dictionary.

6. INDUSTRY ADOPTION OF CURRENT STANDARDS AND TEST STRATEGIES.

Three examples of tools incorporating the new test strategies are described below.

The DFT compiler, SoCBIST by Synopsys [35] automates the creation and integration of IP cores, optimised for test reuse. This tool is based on the IEEE 1450.6 standard. First, the DFT Compiler automatically synthesizes test-reuse IP cores and creates CTL test models for them. Synopsys' TetraMAX automatic test-pattern generation tool then creates reusable test patterns for those cores. Finally, the SoCBIST tool reads the CTL models of these cores and automatically integrates the cores into the overall SoC, reusing pre-supplied core test patterns referenced from the SoC-level pattern set.[36]

Logic Vision has developed a test architecture for cores that are embedded within a SoC called ELT (Embedded Logic Test) Core. ELT core operates by placing an ELT controller in each logic block in the system. Each of these controllers supports random pattern testing and external scan test. Each of the logic blocks within the system can be isolated using the ELT controllers for multi-clock domain, at-speed testing. One of the isolation approaches, dedicated isolation, supports the requirements of the IEEE 1500 standard for test ready cores. Access to the ELT controllers is provided through a hierarchical TAP architecture compliant with the IEEE 1500 standard. One of the advantages of using this IEEE 1500 standard compliant approach is reduced global test signal routing [37].

The Standard for Embedded Core Test (SECT) *e*VC (Verification Components) by Globetech Solutions [38] can verify a chain of one or more IEEE 1500 compliant core wrappers. The *e*VCs are fully compliant with the IEEE 1500. The SECT *e*VC will also provide a feature in the future that will support CTL based auto configuration.

7. CONCLUSION.

In this paper, we have looked at a number of important new techniques, which have been used to test multi-core System-on-Chip designs. It is particularly useful to observe how some of the analyses and proposals of the last decade or so have come to fruition in the form of

implementation of some practical solutions. Many of these solutions are similar to or compatible with the proposals outlined in the recently adopted IEEE standards 1500 and 1450.6. In addition we have noted the release of some tools that incorporate elements of these standards. These developments mean that the next few years will allow researchers to make a realistic assessment of how well their efforts have succeeded in making real progress in overcoming the many challenges of System-on-Chip testing.

8. ACKNOWLEDGEMENT

The authors acknowledge the support of the CSRC at the University of Limerick.

9. REFERENCES

[1] Electronic_Design, "Penny-A-Second SoC DFT Tester Saves Time," 2003.
[2] http://www.ikonix.com/, "The Final Test Report (FTR)," vol. 14, pp. 12, 2003.
[3] Teradyne, "http://www.teradyne.com/catalyst/," ND.
[4] Agilent, "http://www.home.agilent.com/USeng/nav/-536886626.0/pc.html," ND.
[5] www.teseda.com/pdfs, "Transporting STIL between the Teseda V500 and the Agilent 93000 SOC Tester," 2004.
[6] S. K. Goel and E. J. Marinissen, "On-chip test infrastructure design for optimal multi-site testing of system chips," presented at Design, Automation and Test in Europe, 2005. Proceedings, 2005.
[7] E. H. Volkerink, A. Khoche, J. Rivoir, and K. D. Hilliges, "Test economics for multi-site test with modern cost reduction techniques," presented at VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE, 2002.
[8] K. Nagano, "CAD navigation and diagnostics by linking ATE and EDA," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, pp. 1699-1707, 2005.
[9] M. Benabdenbi, A. Greiner, F. Pecheux, E. Viaud, and M. Tuna, "STEPS: experimenting a new software-based strategy for testing SoCs containing P1500-compliant IP cores," presented at Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings, 2004.
[10] IEEE, "IEEE Standard Testability Method for Embedded Core-based Integrated Circuits," in *IEEE Std 1500-2005*, 2005, pp. 0_1-117.
[11] K. Chakrabarty, "Design of system-on-a-chip test access architectures using integer linear programming," presented at VLSI Test Symposium, 2000. Proceedings. 18th IEEE, 2000.
[12] Y. Zorian, E. J. Marinissen, and S. Dey, "Testing embedded-core based system chips," presented at Test Conference, 1998. Proceedings. International, 1998.
[13] S. K. Goel and E. J. Marinissen, "A novel test time reduction algorithm for test architecture design for core-based system chips," presented at European Test Workshop, 2002. Proceedings. The Seventh IEEE, 2002.
[14] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, and C. Wouters, "A structured and scalable mechanism for test access to embedded reusable cores," presented at Test Conference, 1998. Proceedings. International, 1998.
[15] P. Varma and B. Bhatia, "A structured test re-use methodology for core-based system chips," presented at Test Conference, 1998. Proceedings. International, 1998.
[16] T. McLaurin and S. Ghosh, "ETM10 incorporates hardware segment of IEEE P1500," *Design & Test of Computers, IEEE*, vol. 19, pp. 6-11, 2002.
[17] M. Chandramouli, "DBIST answers advanced SoC test challenges," *http://www.synopsys.com*,ND.
[18] V. Iyengar, A. Chandra, S. Schweizer, and K. Chakrabarty, "A unified approach for SoC testing using test data compression and TAM optimization," presented at Design, Automation and Test in Europe Conference and Exhibition, 2003, 2003.
[19] F. Karimi, Z. Navabi, W. M. Meleis, and F. Lombardi, "Using data compression in automatic test equipment for system-on-chip testing," *Instrumentation and Measurement, IEEE Transactions on*, vol. 53, pp. 308-317, 2004.
[20] M.-L. Flottes, R. Poirier, and B. Rouzeyre, "On using test vector differences for reducing test pin numbers," presented at Electronic Design, Test and Applications, 2004. DELTA 2004. Second IEEE International Workshop on, 2004.
[21] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the foundation for compressed ATPG vectors," presented at Test Conference, 2001. Proceedings. International, 2001.

[22] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, "A SmartBIST variant with guaranteed encoding," presented at Test Symposium, 2001. Proceedings. 10th Asian, 2001.
[23] K. J. Balakrishnan and N. A. Touba, "Reconfigurable linear decompressors using symbolic Gaussian elimination," presented at Design, Automation and Test in Europe, 2005. Proceedings, 2005.
[24] S. K. Goel, K. Chiu, E. J. Marinissen, T. Nguyen, and S. Oostdijk, "Test infrastructure design for the Nexperia/spl trade/ home platform PNX8550 system chip," presented at Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings, 2004.
[25] R. Kapur, M. Lousberg, T. Taylor, B. Keller, P. Reuter, and D. Kay, "CTL the language for describing core-based test," presented at Test Conference, 2001. Proceedings. International, 2001.
[26] K. R. Miller, W. J. Lonowski, R. Kapur, P. Harrod, and D. Appello, "CTL: The New Language of DFT," in *Evaluation Engineering*, 2003.
[27] S. Shoukourian, V. Vardanian, and Y. Zorian, "SoC yield optimization via an embedded-memory test and repair infrastructure," *Design & Test of Computers, IEEE*, vol. 21, pp. 200-207, 2004.
[28] S. Hamdioui, G. Gaydadjiev, and A. J. van de Goor, "The state-of-art and future trends in testing embedded memories," presented at Memory Technology, Design and Testing, 2004. Records of the 2004 International Workshop on, 2004.
[29] S. Banerjee, D. Mukhopadhyay, and D. R. Chowdhury, "Automatic generated built-in-self-test for embedded memory," presented at India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004. First, 2004.
[30] B. H. Fang, Q. Xu, and N. Nicolici, "Hardware/software co-testing of embedded memories in complex SOCs," presented at Computer Aided Design, 2003. ICCAD-2003. International Conference on, 2003.
[31] F. Karimi and F. Lombardi, "A scan-BIST environment for testing embedded memories," presented at Memory Technology, Design and Testing, 2002. (MTDT 2002). Proceedings of the 2002 IEEE International Workshop on, 2002.
[32] C.-H. Tsai and C.-W. Wu, "Processor-programmable memory BIST for bus-connected embedded memories," presented at Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific, 2001.
[33] R. Rajsuman, *System-on-a-Chip: Design and Test*. Boston, London: Artech House Publishers, 2000.
[34] D. Appello, A. Fudoli, V. Tancorre, F. Corno, M. Rebaudengo, and M. Sonza Reorda, "A BIST-based solution for the diagnosis of embedded memories adopting image processing techniques," presented at On-Line Testing Workshop, 2002. Proceedings of the Eighth IEEE International, 2002.
[35] Synopsys, "DFT Compiler SoCBIST- Deterministic Logic BIST," 2002.
[36] Electronic_Design, "Standards Boost SoC Test," 2003.
[37] LogicVision, "Core Embedded Logic Test (ELT Core)," 2004.
[38] Globetech_Solutions, "IEEE 1500 (SECT) *e*VC," 2005.

Rapid-Prototyping Emulation System for Embedded System Hardware Verification, using a SystemC Control System Environment and Reconfigurable Multimedia Hardware Development Platform

Dave Carroll, Richard Gallery
dave.carroll@itb.ie , richard.gallery@itb.ie
School of Informatics and Engineering,
Institute of Technology Blanchardstown, Dublin 15, Ireland

Abstract

This paper describes research into the suitability of using SystemC for rapid prototyping of embedded systems. SystemC[1][2][3] communication interface protocols [4][5] are interfaced with a reconfigurable hardware system platform to provide a real-time emulation environment, allowing SystemC simulations to be directly translated into real-time solutions. The consequent Rapid Prototyping Emulation System Platform[1], suitable for the implementation of consumer level multimedia systems, is described, including the system architecture, SystemC Controller model, the FPGA configured MicroBlaze CPU system and additional logic devices implemented on the Multimedia development board used for the hardware in the PESP, illustrated in the context of a typical application.

Introduction

The paper describes research into the development for SystemC of an emulation environment in which SystemC based specifications can be translated directly into real-time hardware/software implementations. This would allow SystemC based designs to be implemented directly as real-time solutions, and would provide a rapid prototyping route (improved time-to-market) for complex hardware/software systems.

This paper describes a prototyping platform, incorporating hardware acceleration under software control, suitable for the real-time implementation of SystemC designs,. The development of

this system includes research into the architecture required to enable SystemC to be used directly for real-time systems, and the implementation of a prototyping system, with an application focus on consumer level multimedia systems, to investigate the viability of the approach.

The SystemC section of the prototyping system provides a tool for the specification, implementation and evaluation of different embedded system architectures and facilitates selection and evaluation of different communication interface approaches between the embedded SystemC functional models within the design.

The hardware acceleration platform, oriented towards consumer level multimedia applications, is based on an FPGA multimedia development platform, which provides a reconfigurable hardware real-time emulation environment, for prototyping real-time application designs, in conjunction with the SystemC system model. The Hardware subsystem is designed around a Xilinx MicroBaze and Multimedia Development Board[11][2], incorporating a MicroBlaze[3] processor[16] architecture implementation, on an Xilinx Virtex II[9] FPGA based platform. Hardware functional blocks developed as part of the system are written as VHDL[4] models[5] [18][19][20] in the FPGA.

Real-time implementation of SystemC based designs

Research questions with regard to the practicality of developing real-time emulation systems for SystemC designs include:

- Whether the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model
  - ⇨ If not, is there some subset of, or enhancement to, the SystemC approach, which is suitable for real-time implementation.
  - ⇨ Which modelling approaches, when deployed in SystemC based designs, are suitable for (real-time) emulation.
- The interaction of SystemC with real-time OS. This is an active development area within SystemC, and also in hardware-software co-verification systems (e.g. [6][7][8]).
- What hardware/software architecture(s) for emulation platforms are required to allow real time implementation of SystemC designs?
- What hardware components are required within such real-time platforms to facilitate SystemC designs?
- What interfaces are required between the software and hardware aspects of such systems?

This paper discusses the PESP and in doing so presents a suitable SystemC based design, that when interfaced with a reconfigurable hardware system platform, using a suitable communication interface protocol, demonstrates that the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model. The paper discusses a modular modelling approach, which when deployed in the PESP SystemC based design, is suitable for (real-time) emulation. The author presents an hardware/software architecture for

[1] Here after referred to as PESP
[2] Here after referred to as MMDB
[3] Harvard architecture, 32bit CPU, defined and supplied by Xilinx
[4] VLSI Hardware Descriptive Language
[5] Refers to a functional block configured within the FPGA, written in VHDL or Verilog.

an emulation platform that facilitates the real time implementation of the SystemC design and defines the hardware components used in that design.

A Co-Verification Platform

There is already active research into the development of architectures and platforms suitable for the emulation of systems described with system level description languages (such as SystemC). To date the research is largely targeted towards co-verification platforms[6], where fast verification of a design is required, rather than real-time emulation [6]. Nevertheless there is a significant correspondence in the research required to develop both fast verification systems and real-time systems.

For example in [4] the functional requirements for a co-emulation modelling interface are discussed and an architecture and API is provided (SCE-MI) to achieve these requirements. SCE-MI has been developed to meet growing industry demands for verification platforms for Systems-on-Chip (SoC) designs. It provides a mechanism via which the software aspects of an SoC design may be simulated on a workstation, the hardware aspects may be simulated on a hardware verification platform, and an efficient interface mechanism (hardware-software transactors) between the two, represented in Figure 12. SCE-MI facilitates the use of a system design language such as SystemC, as it offers the potential for fast verification (as the hardware aspects of the system may be emulated using real hardware, as opposed to using a workstation based software simulator).

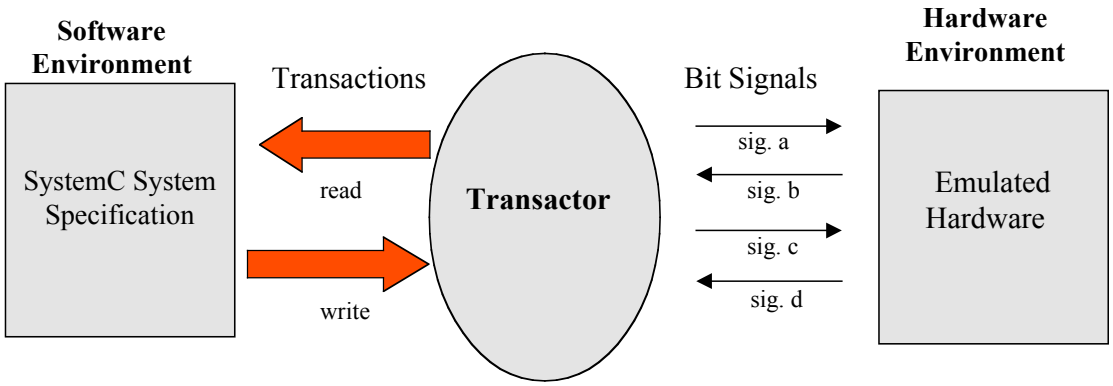

Figure 12: co-emulation modelling interface

In a SystemC design it would be normal to initially produce a high-level abstract model of the entire system, with the process of refinement then being used to redesign the system, such that

[6] Sometimes termed hardware/software co-emulation or co-verification platforms. In general the software aspects of the design under test (DUT) are run on a workstation, the hardware aspects are run on the hardware emulation platform, with a mechanism provided for communication and synchronisation between the two.

software components of the system modelled at a high level of abstraction, and hardware components modelled at an appropriate lower level of abstraction. Once this process of refinement has taken place, the hardware components must be simulated and verified. The Zebu system[21] provides a software/hardware co-verification platform using an implementation of the SCE-MI protocol. Using Zebu the software and hardware components of the design can be verified together, with the hardware emulation system allowing much more rapid verification time than would be the case using a software simulation of the hardware components.

The hardware-software transactor is a form of abstract gasket, which forms part of the SCE-MI infrastructure. The transactor communicates at the transactional level (e.g. Read and Write commands) with the software side of the system model, decomposing untimed messages into a series of cycle-accurate clocked signals. These clocked signals form the communication interface between the transactor and the hardware, on the hardware side of the system model. Similarly the hardware communicates with the transactor at the signal level, where the cycle accurate signals are recomposed into transactional level messages, for transfer to the software interface. Figure 13 shows an example of a message being translated from the software side of a system model, through the transactor, to the hardware signal side of the transactor.

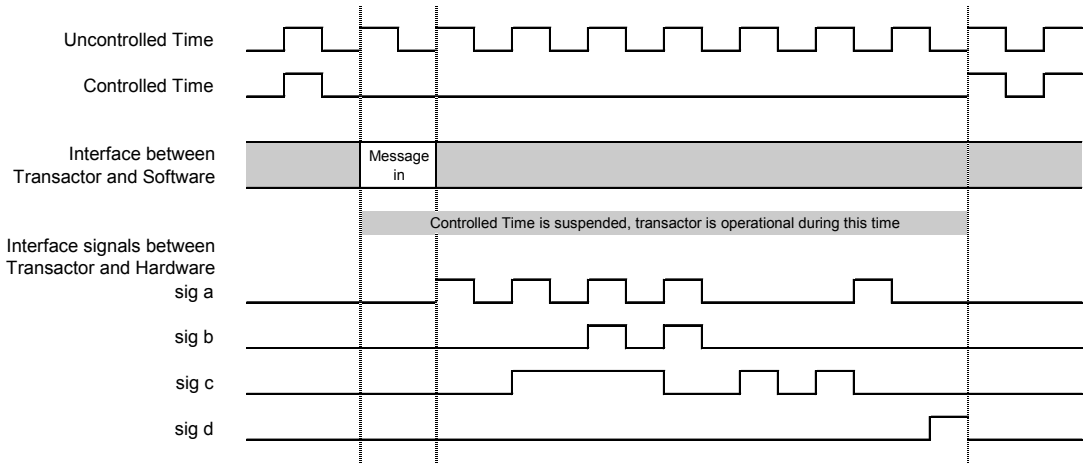

Figure 13: message timing through transactor

During the period from when the message is received by the transactor, controlled time will be suspended within the software environment via handshaking between the transactor and the SCE-MI infrastructure. The transactor will decompose the incoming message and generate required cycle accurate bit level signals to drive the hardware. Once the transactor actions have been completed, controlled time will be resumed via the transactor handshaking and the SCE-MI.

PESP Description

The PESP has been developed with a focus on investigating the application of SystemC based rapid prototyping in consumer level multimedia applications. Its is based around the concept of a SystemC based control system, supported by application specific hardware accelerators and other components, with communication between the software and hardware managed by transactors based upon the SCE-MI protocol.

Currently this system is based on a SystemC control system using set-up on a host PC[7] and a MMDB[11] hardware development platform, containing a Xilinx, MicroBlaze softcore processor[16] implemented on a XilinxVirtex II FPGA fabric, along with other ASIC devices. The SystemC control system and the MMDB communicate through transactors, utilising a serial or other interface (such as an Ethernet link) at the physical layer. It should be apparent that the use of a PC in the development system is for convenience of development purposes only, and that, outside of the research environment, the System C control system would naturally be implemented, for example, on the MicroBlaze softcore processor.

Platform Architecture

The architecture of PESP can be described using a multi-layered model to describe the overall system in an abstract form, de-coupling the system architecture from the implementation details. The multi-layered architecture consists of an Application Layer (AL), Presentation Layer (PL) and Driver Layer (DL), as shown in Figure 14.

The PL is responsible for the management of the data-flow, which implements the various functions of the platform and is controlled by the AL. The PL converts information received from the AL into a defined order and communicates with the DL, sending requests to and receiving status from the DL. The DL controls each of the hardware components (e.g. in the case of a Digital camera application, a task may consist of image capture) directly, passing required parameters and enabling the hardware components. The DL will receive status information from the hardware components and pass appropriate status response to the PL. Where several tasks are required to complete one single user command (e.g. capture an image and save it to memory), the PL will define the sequence of execution and control the execution, through the DL.

[7] Running a Microsoft Windows operating system

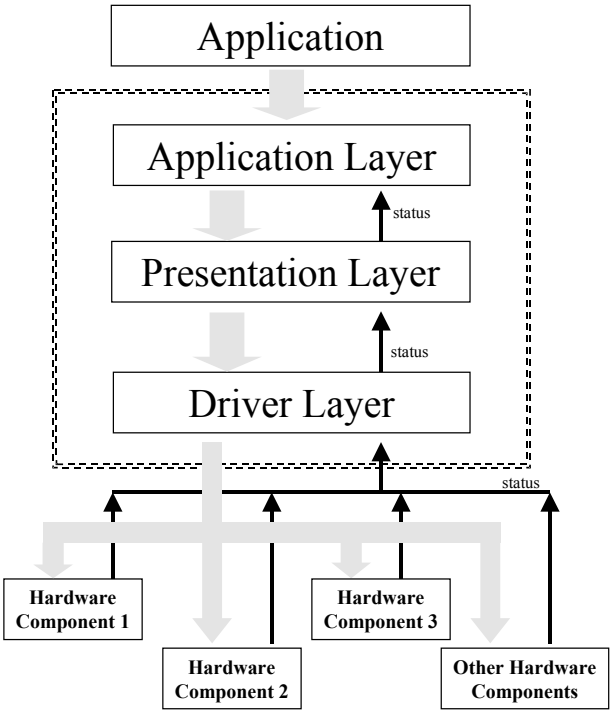

Figure 14: Multi-layered Architecture Model

The DL is the only layer that interfaces directly with the Hardware Components of the system. Thus if any changes are required to the hardware, only the DL will be required to be modified to accommodate the change. The Hardware Component blocks are under the control of the DL and may be implemented directly in hardware, software, or a combination of both hardware and software. For example an image compression feature may be implemented in hardware, in software or in a combination of both, depending on the optimal implementation for the particular system, which will be determined by the design engineer based on system design goals and constraints. For example, hardware implementation of the compression feature might result in a faster execution of this feature than the software implementation of the same feature, however this may result in an unacceptable increase in hardware costs.

SystemC and MMDB Communications Interface

The overall platform architecture described above is now mapped to an implementation incorporating SystemC, a re-configurable hardware board. The communications interface between the SystemC model and the re-configurable hardware is modelled on the Standard Co-Emulation API: Modelling Interface[4]. This specification describes a modelling interface based on a multi-channel abstract bridge, providing multiple communication channels that allow software models describing system behaviour, such as the PESP SystemC controller

models, to connect to structural models describing implementation of a hardware emulation system.

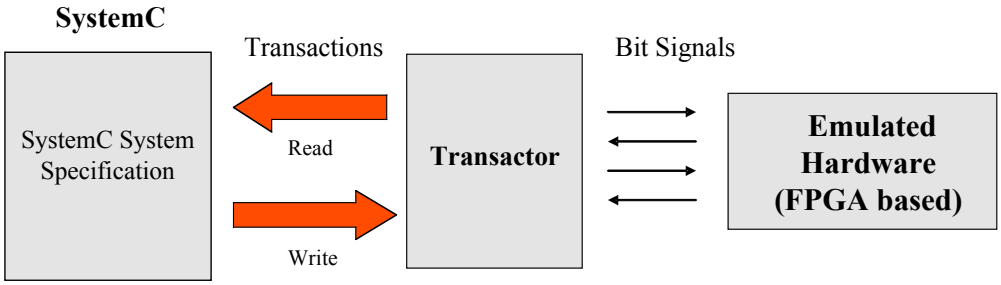

Figure 15: Interface between the SystemC controller and MMDB platform

Figure 15 shows a high level view of the communications interface between the SystemC controller system and the MMDB based hardware emulation system. The SystemC system is on the left of the figure and consists of several SystemC models describing the controller functions, the Xilinx MMDB platform is on the right of the figure. The Transactor performs the interface mechanism linking the high level SystemC controller description and the hardware implementation of functions within the FPGA MMDB development platform. The Transactor implementation consists of several layers of software and hardware, from the SystemC model down to the PC hardware drivers, across the physical serial interface to the MMDB development platform and onto the MicroBlaze system implemented on the FPGA fabric.

SystemC Controller System Description

The PESP SystemC is constructed using a modular approach to provide partitioning between the different functional elements of the overall controller. Some of the benefits obtained from using this approach include:

1. The ability to break the complex system to smaller more manageable pieces, which proved useful when defining the overall SystemC system. A list of sub-functions required by the system was initially generated and a SystemC model created for each sub-function. Each model was then tested individually to ensure functionality. When the overall system was built, errors were diagnosed quickly by focusing on the interconnections between the system sub-elements.

2. The movement of functionality between different models, which proved useful during the Control model design. For example, moving all of the message display functions to the message display model, which has previously been contained in the system control

model. The modular nature of the system made the movement of the functionality a relatively straightforward process, with only two SystemC models affected.

3. The simplification of addition and/or removal of models from the system. This was particularly useful when moving functionality from the SystemC Controller System to hardware, as the hardware platform was being built. Additional SystemC models were required in order to implement the SCE-MI protocol. It was simply a matter of defining the functionality and inserting the models into the existing SystemC system. The ability to change or refine the communication interfacing between the different models is relatively straightforward when the system is designed with a modular construction. This proved very useful when deciding which interface channel type to use to connect the SystemC models that make up the overall SystemC Controller system. Different interface channel and port types can affect the flow of data from one SystemC model to another, communication refinement is described in detail in [2][3] and [5]. Primitive channels (sc_fifo) were chosen for communications between SystemC models in this system. The sc_fifo primitive channel provides the ability to communicate between SystemC models using a blocking and non-blocking read/write instruction; which ensures that data does not get lost between data transfers. Primitive channels also suspend the operation of a Thread model if the channel is full or empty, depending on the requested command. Figure 16 shows the SystemC models designed for this system and the interconnections used.

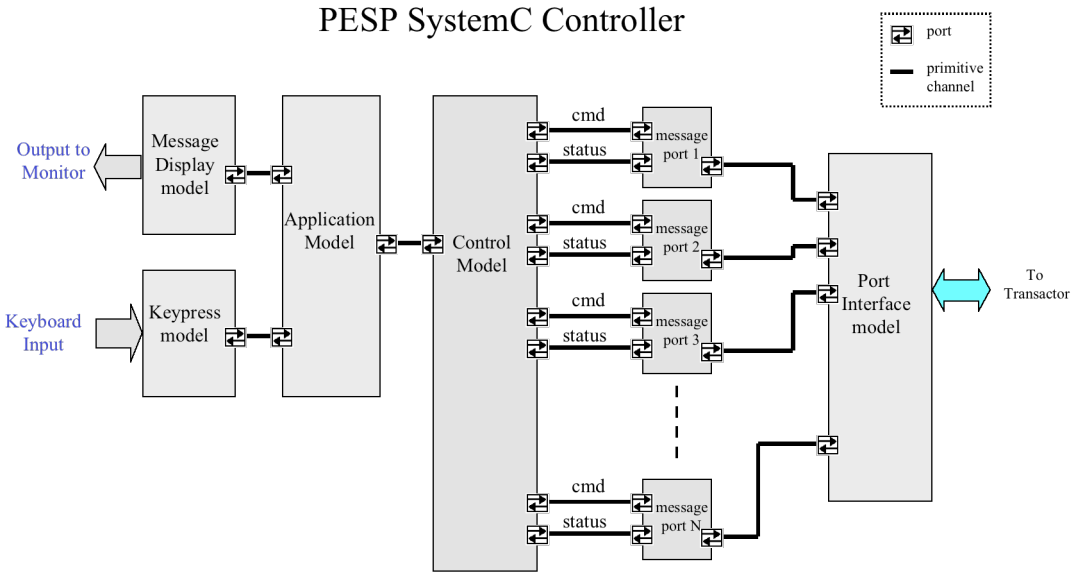

Figure 16: Block diagram for PESP SystemC Controller

The controller system architecture can be represented as a multi-layered architecture model as shown in Figure 17, where the AL of the system consists of the SystemC application model. The PL function of the controller system architecture is defined in the control model. The PL puts commands received from the AL in a defined order and communicates with the DL, sending requests and receiving status. The PL defines the order of execution and controls the hardware component activities through the DL. The PL will interpret commands and ensure that only predefined and acceptable user commands will be responded to and passed on to the DL. The PL will also communicate command execution status to the user, via the AL. In this system the DL is spread across the Port Interface model (Figure 16) and on to the MMDB platform. The DL contains the transactors, resident on the MMDB, and is responsible for all communications with the hardware.

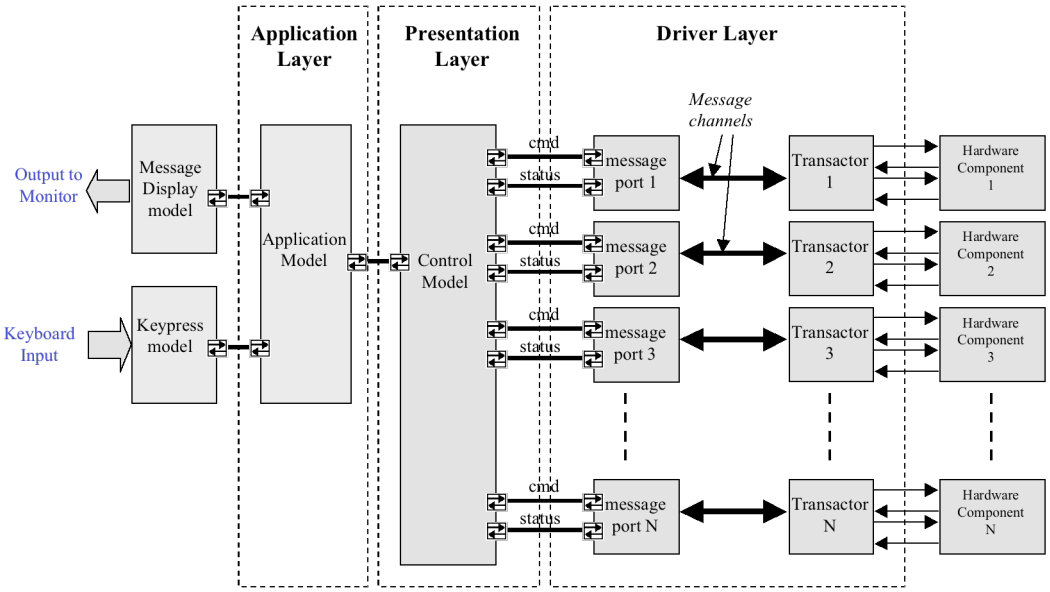

Figure 17: multi-layered architecture of SystemC Controller model

Reconfigurable Hardware

The MMDB was chosen to provide the reconfigurable hardware as it provided a range of multimedia oriented features (such as video and audio codecs), combined with an FPGA to allow additional hardware functionality to be developed. This system is built around the Xilinx MicroBlaze 32-bit RISC Soft processor, which, as currently configured, acts as part of the transactor, interpreting commands from the SystemC controller, controlling hardware components on the MMDB, and managing dataflow between those hardware components themselves, and the higher level SystemC modules. The transactor is implemented through software (written in C programming language[17]) and hardware, which instructs the MicroBlaze hardware to initialise the image grabber hardware using clock cycle accurate bit signals through the MicroBlaze Peripheral Bus interface registers Figure 18.

As has been eluded to earlier, the higher level SystemC models are currently implemented on a PC, but it would be the intention, in a later stage of development, to transfer all software functionality to the MicroBlaze itself, in which case it would implement the entire system architecture described in Figure 14.

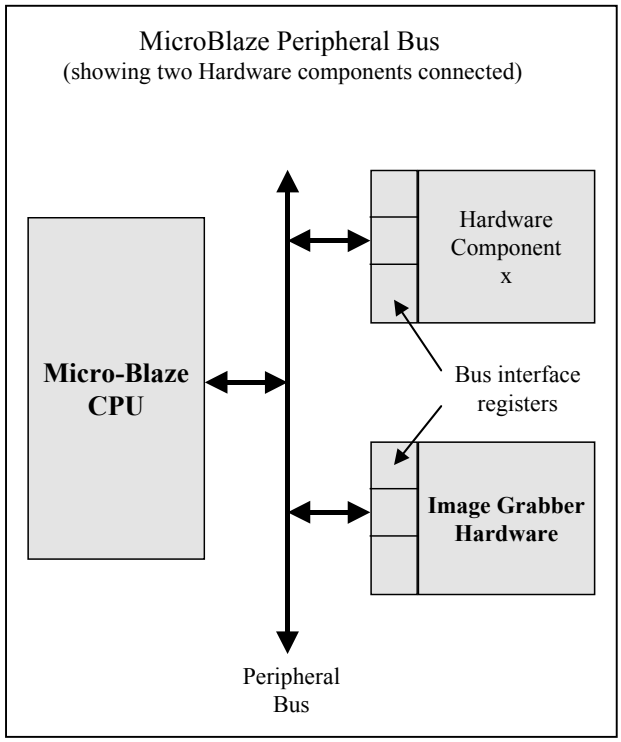

Figure 18: simplified MicroBlaze Peripheral Bus connection

Figure 19 shows the hardware components that were developed as part of the research and are controlled by the transactor. The Display Engine Module, Image Grabber, Workarea Control Module and Resync Module are all VHDL modules, designed, constructed and tested using the Xilinx ISE design environment [12]. The peripherals are attached to the MicroBlaze CPU via the OPB[8] Interface Bus, which is a 32Bits data bus and is configured to run at 81 MHz. All devices are accessed and controlled by software running on the MicroBlaze processor. Video images are decoder external to the FPGA and enter the FPGA via the Video Resync module[9] in 10-bit YCbCr PAL data format [13][14][15]. Here the data is synchronised with the output from a 27 MHz-clock generator model, contained in the FPGA (not shown in the figure). Data exiting the Video Resync model is passed directly to the Video Encoder, a single chip, outside the FPGA boundary, on the development board.

[8] MicroBlaze On-Chip Peripheral Bus
[9] This is a VHDL model, which is used to synchronise the input data with the 27 MHz input clock. The model implements a data FIFO function, using a Dual Port Block RAM to temporarily store a portion of a line of video data.

Image display switching between the live video input data stream and the Display Engine output (containing a saved image) is controlled by the Display Engine under the control of the software running on the MicroBlaze.

The Image Grabber VHDL model is used to capture and store images, and is under the control of the MicroBlaze software. Communications to and from, the software takes place through Control, Status and Data models connected to the OPB interface bus. Images are stored in the Image Memory[10] device, located outside the FPGA. The SRAM is controlled by the Image Grabber VHDL model.

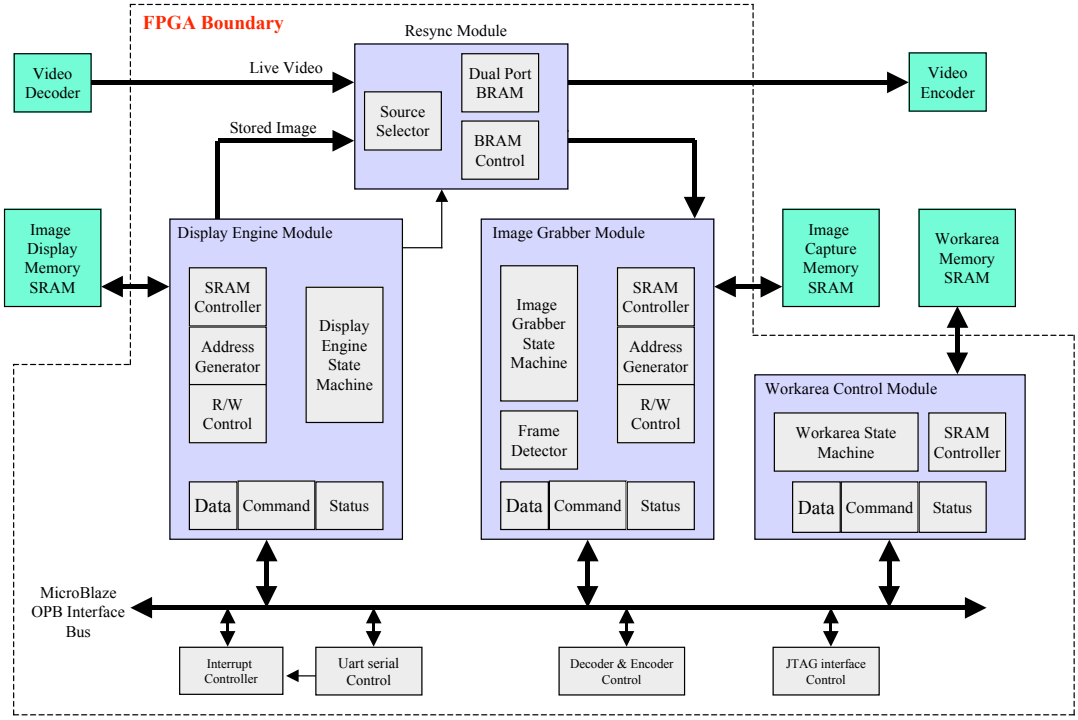

Figure 19: FPGA based Video System Hardware Architecture

The Display Engine model is responsible for the display of captured images. It provides a continuous stream of 10-bit video data to the Video Resync module, synchronised with the system 27 MHz clock. The Display Engine also controls the contents of the Image Display Memory device, which is located on the MMDB, external to the FPGA. The Display Engine is under the control of the MicroBlaze via a Control, Status and Data models connected to the OPB interface bus.

The Interrupt Controller is used to interrupt the CPU upon request from the Uart Serial Control model, which is connected to the RS232 interface and the OPB interface bus.

[10] A 512K x32 SRAM

Commands to and from the SystemC Controller are sent through the Uart model. The JTAG Interface Control model is used for debug interface access for the EDK system.

The Workspace Memory consists of a single SRAM device on the MMDB and is controlled by the Workarea Controller connected to the OPB bus. This storage facility is used for temporary storage of images as they are being manipulated.

Sample PESP Application Implementation

A digital camera was used as the focus application for functional testing of the PESP platform. The digital camera application is an example of a widely available digital system, which is sufficiently complex to provide scope for different architecture and technology configurations. The digital camera functions also suited the application devices, which are integrated into the Xilinx MMDB hardware development board (however PESP is not board specific). A digital camera is an example of a system that uses several data processing functions, which are relatively self-contained and as such simplify the implementation of functions using a modular approach. For example the image grab function is not dependent on image display function and as such both functions can be implemented in separate hardware components.

Conclusion & Future work

This paper discussed the PESP, which consists of an integration of both a SystemC controller and a reconfigurable multi-media hardware development platform.

This paper presented a suitable SystemC based design, that when interfaced with a reconfigurable hardware system platform, using a suitable communication interface protocol, demonstrates that the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model.

The paper discussed a modular modelling approach, which when deployed in the PESP SystemC based design, is suitable for (real-time) emulation. The paper also discussed an hardware/software architecture for an emulation platform that facilitates the real time implementation of the SystemC design and defines the hardware components used in that design.

The higher level SystemC models used in PESP are currently implemented on a PC, however models could be transferred from the PC platform to the MicroBlaze itself, which would implement the entire system architecture described in Figure 14 on the MMDB. The interaction of SystemC with real-time OS was not considered as part of this paper and could

be an area of focus for future work. This is an active development area within SystemC, and also in hardware-software co-verification systems (e.g. [6][7][8]).

References

[1] OSCI; www.systemc.org
[2] OSCI; "Functional Specification for SystemC 2.0", available at www.systemc.org
[3] T. Grotker, S Liao, G. Martin, S Swan; "System Design with SystemC"; Kluer Academic Publishers, ISBN 1-4020-7072-1
[4] Standard Co-Emulation Modelling Interface (SCE-MI) Reference Manual, Version 1.0, Published by Accellera, available from www.eda.org/itc
[5] J. Stickley, Duaine Pryor; "Functional Requirements Specification: Standard Co-Emulation Modeling Interface (SCE-MI)"; www.systemc.org
[6] Emulation and Verification Engineering; "Universal Design Verification Platform for IP, FPGA, ASIC Design and Embedded Software Developers"; www.eve-team.com
[7] Jon Connell, Bruce Johnson , Early Hardware/Software Integration Using SystemC 2.0, ESC San Francisco 2002.
[8] L. Benini et al., SystemC Cosimulation and Emulation of Multiprocessor SoC Designs, IEEE Computer, April 2003, pp. 53-59.
[9] Virtex II data sheet (Xilinx doc. no. DS031-1), www.xilinx.com
[10] Embedded Systems Tools Guide (EDK 3.2.2), www.xilinx.com
[11] Xilinx Virtex II MicroBlaze and Multimedia Development Board, www.xilinx.com
[12] ISE user guide, www.xilinx.com
[13] A Technical Introduction to Digital Video, Charles Poynton, John Wiley & Sons, ISBN 0-471-12253-X
[14] Video & Television Engineering, J. Whitaker and B. Benson, McGraw Hill, ISBN 0-07-069627-6
[15] ADV7185 Video NTSC/PAL Decoder data sheet, www.analog.com
[16] MicroBlaze RISC 32-Bit Soft Processor data sheet, www.xilinx.com
[17] The C Programming Language, Kernighan & Ritchie, Prentice Hall, ISBN 0-13-110362-8
[18] VHDL made Easy, D. Pellerin & D. Taylor, Prentice Hall, ISBN 0-13-650763-8
[19] VHDL for Designers, S. Sjoholm & L. Lindh, Prentice Hall, ISBN 0-13-473414-9
[20] Digital System Design with VHDL, M.Zwolinski, Prentice Hall. ISBN 0-201-36063-2
[21] Co-Simulation Between SystemC and a New Generation Emulator, C. Alquier, S. Guerinneau, L.Rizzatti, L. Burgun, DesignCon 2003

http://www.itb.ie

http://www.itb.ie/site/researchinnovation/itbjournal.htm