A Local and Remote Laboratory User Experimentation Access Arrangement using Digital Programmable Logic

I.A. Grout, J. Walsh, T. O’Shea and M. Canavan

Department of Electronic and Computer Engineering, University of Limerick, Ireland

Key words: Remote Laboratory, microelectronics design and test education

Abstract:

This paper will discuss the set-up and role of a programmable logic based experimentation arrangement that is suitable for both local and remote teaching and learning scenarios targeting electronic and microelectronic circuit design and test principles. With this experimentation arrangement, the ability to provide both local and Internet based “remote” access for the student and the teacher can provide a number of advantages where physical laboratory accessibility is limited and/or the learning experience must be undertaken with one or more of the parties remotely based. The paper concentrates on the design and example use of the system that is to be considered for use in the next academic year.

1. Introduction

Remote learning[1][2] has matured over the last number of years to provide a realistic and important support mechanism in which practical laboratory based experimentation work can be undertaken by remote learners, who are provided with access to facilities that they would not otherwise necessarily be able to utilise. The emphasis has been on providing an enhancement to the student learning experience by adopting the Internet as a suitable means in which to provide a communications channel between the learner, teacher and hardware/software experimentation facilities for remote learners. This is particularly important in the engineering and science disciplines. The ability to design, develop and utilise experimentation based on remote learner requirements is an exciting and important step in the evolution of, and increasing access to, teaching and learning resources. The ability then to design, develop and utilise a core experimentation arrangement which is utilisable by both local and remote users, providing flexibility in the location of both the teacher and learner, is a step in the direction to consider the provision for cost-efficient experimentation.

Whilst the technological ability to support remote learning exists in a number of forms, and has been widely demonstrated, the economics of providing such a learning support mechanism cannot be underestimated and must be realistic in the context of course provision. Experimentation set-ups with multiple modes of use can aid a reduction in the cost of such laboratory equipment. This paper will discuss an experimentation arrangement aimed to investigate and demonstrate a common experimentation arrangement supporting both local and remote users. Sections 2 and 3 introduce the experimentation arrangement and Internet access equipment developed. Section 4 will place the work into context in the support of local and remote users, with a case study design aimed to highlight the operation provided in Section 5. Section 6 will conclude the paper and discuss future work to be undertaken. The
paper discussions will concentrate on the hardware experiment arrangements and a web based user interface design and support.

2. Programmable Logic Based Laboratory Experimentation

The basic system consists of a circuit hardware prototyping system, which once configured (via a PC), can be run, in conjunction with, or independent to, the PC. The core of the system is based on the Lattice Inc.[3] MACH4A CPLD (Complex Programmable Logic Device). This is configured via the PC parallel port and the JTAG (Joint Test Action Group) boundary scan interface on the CPLD. The system hardware contains support circuitry including a visual display and 8-bit resolution data converters (Analogue to Digital (A/D) and Digital to Analogue (D/A)). When the user is ready, a new or a previously designed circuit would be downloaded on to the CPLD. The user can produce a test vector set for the circuit using software program with a suitable Graphical User Interface (GUI) in order to determine the functionality of the design via suitable test vectors for the particular design. This system could provide a new fast and effective way of teaching microelectronic design and test engineering concepts. The system arrangement for this microelectronic circuit engineering laboratory environment is shown in figure 1.

![Figure 1: System Arrangement](image)

The system is referred to as the University of Limerick Test Engineering Development Box (ULTEDB)[4] environment, see figure 2. The basic principle of operation is as follows:-

1. A design is downloaded into the CPLD (typical digital designs intended to highlight specific aspects relating to both design, test and Design for Testability (DfT)). The circuit may be configured to be a normal fault-free circuit or to include specific logical circuit faults.
2. The student is then required to analyse the design, develop and provide test vectors in order to undertake test procedures on the design. The test vectors are either applied manually (via switches/external logic) or automatically via the PC.

![Figure 2: ULTEDB Environment](image)
3. Web Server Arrangement

The web server arrangement[5] is based on the Apache Web Server, with CGI scripting (PERL) and Visual Basic (VB) application programs. This arrangement allows for Internet access (web page I/O for the user) and experimentation hardware access (for this arrangement, via the RS-232 serial port). The system arrangement is shown in figure 3.

With this arrangement, all the necessary hardware and software interfacing exists (i.e. in a modular, template form) such that extensions to the experimentation arrangement can be readily undertaken in order to extend the functionality of the system. Figure 4 shows the arrangement now considered in this work.
4. Remote and local access modes

A requirement for the work undertaken is to design, develop and utilise the hardware/software solution by developing as much of the hardware and software as practical. This has cost implications and the ability to re-use as much of the system as possible in order to extend the functionality is a benefit. In order to consider the re-use of the experimentation, it is possible to envisage the following modes of operation for both the teacher and learner, see figure 5.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Remote Location</th>
<th>Local Laboratory Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LEARNER</td>
<td>LEARNER</td>
</tr>
<tr>
<td>2</td>
<td>LEARNER</td>
<td>LEARNER</td>
</tr>
<tr>
<td>3</td>
<td>LEARNER</td>
<td>LEARNER</td>
</tr>
<tr>
<td>4</td>
<td>LEARNER</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>LEARNER</td>
</tr>
</tbody>
</table>

Users access the Laboratory via a PC and Web Page

Location of experimentation hardware and web server

**Figure 5: Modes of operation**

*Figure 5* shows 5 modes of operation which could be undertaken in order to access the hardware experiments:-

- In **Mode 1**, this is seen as the “traditional” laboratory mode in that both the teacher and learner are locally based and have a “hands-on” utilisation of the experimentation.
- In **Mode 2**, the learner is remotely based, accessing the experimentation via the Internet and the teacher provides the necessary support in a tutorial style of interaction.
- In **Mode 3**, both the teacher and learner are remotely based and access the experimentation via the Internet in different modes (a teacher and a learner mode).
- In **Mode 4**, the learner is remotely based, but there is no direct teacher support.
- In **Mode 5**, the learner is locally based, but there is no direct teacher support.

The system arrangement identified and discussed in *Sections 2 and 3* provides the necessary base hardware/software to facilitate these potential modes of operation. A number of important considerations exist, include:-

- Flexible use.
- Support local and remote teaching and learning scenarios.
- Custom solution – both hardware and software in order to minimise the need for specific hardware/software licensing agreements.
- Minimal set-up and maintenance requirements.
- A library of case-study circuits is created – including circuit operation in both fault-free and specific circuit fault conditions.
- Ability to include an automatic self-test routine and system “health” reporting to the teacher/technical support staff.
5. Case study laboratory experiment

The experimentation set-up has the potential to be utilised in a number of modes and for a range of electronic hardware design and test related scenarios. For this work, the main area of interest is microelectronic circuit design and test engineering education. An example circuit that could be considered in the teaching of digital logic design and test is the priority encoder. Table 1 identifies the truth table for an 8-input encoder, with an example VHDL description in figure 5.

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A7</td>
<td>B2</td>
</tr>
<tr>
<td>A6</td>
<td>B1</td>
</tr>
<tr>
<td>A5</td>
<td>B0</td>
</tr>
<tr>
<td>A4</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>A0</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Priority Encoder Truth Table

Such a design is a useful combinational logic design which can be considered for learning as follows:-

- The learning of digital combinational logic design (truth-tables, Boolean logic).
- The learning of Hardware Description Languages (HDLs) – Verilog-HDL or VHDL - types of HDLs available, design development, code syntax, coding styles and HDL based design flows.
- The use of programmable logic for implementing digital designs (concepts of programmable logic, schematic entry and HDL synthesis, simulation)
- Digital logic test (test stimulus generation, fault modelling, functional and structural test, test program creation).

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity encoder is
    Port (A : in std_logic_vector(7 downto 0);
          B : out std_logic_vector(2 downto 0);
          Valid : out std_logic);
end encoder;

architecture simple of encoder is
begin
    B <= "000" when A = "00000001" else
         "001" when A = "00000010" else
         "010" when A = "00000100" else
         "011" when A = "00001000" else
         "100" when A = "00010000" else
         "101" when A = "00100000" else
         "110" when A = "01000000" else
         "111" when A = "10000000" else
         "000";
    Valid <= '1' when A = "00000001" or A = "00000010" else
              "001" when A = "00000100" or A = "00001000" else
              "010" when A = "00100000" or A = "10000000" else
              '0';
end simple;
```

Figure 6: VHDL description of design
6. Conclusions and future work

A extension to a remote learning experimentation arrangement has been discussed in which a CPLD based programmable logic experiment has been added in order to provide both local and remote access modes of operation. This extends the functionality of the developed experiments and provides a demonstrator for a flexible experimentation arrangement may be developed to support the teaching and learning of electronic and microelectronic circuit design and test engineering concepts.

7. Acknowledgements

The authors would like to thank the University of Limerick and the University of Limerick Foundation for the funding to support the initial support of the Remote Test Laboratory access project, along the TRIP initiative managed by the Centre for Teaching and Learning at the University of Limerick.

8. References


Author(s):

Ian Grout, Dr
Joseph Walsh, Mr
Thomas O’Shea, Mr
Michael Canavan, Mr

University of Limerick, Department of Electronic and Computer Engineering, Limerick, Ireland

Email: Ian.Grout@ul.ie, Joseph.Walsh@ul.ie, Thomas.O’Shea@ul.ie and Michael.Canavan@ul.ie