Application of a Block-based Approach in Design of a Reconfigurable Virtual Instrumentation Platform

Gilbert Osei-Dadzie,
Department of Electrical & Electronic Engineering
Kwame Nkrumah University of Science and Technology, Ghana
gilbertoseid@theiet.org

Kwame Osei Boateng
Department of Computer Engineering
Kwame Nkrumah University of Science and Technology, Ghana
boat@ieee.org

Abstract

Scientists, particularly those in developing countries, cannot take full advantage of the availability of research equipment offered by the market due to high cost of commercial instruments. Through a Reconfigurable Virtual Instrumentation (RVI) platform, it is possible to emulate many standard general purpose instruments. It is also possible to implement sophisticated instrumentation for custom specific applications that cannot be accomplished by standard instruments. This paper presents the application of a hierarchical, block-based design approach in the implementation of a reconfigurable virtual instrumentation system, based on a Field Programmable Gate Array (FPGA). This work adopts a previously reported RVI architecture. Based on this architecture, implementation of an RVI system having a digital frequency meter (DFM), a function generator and a digital thermometer as the instrument cores is presented. Detailed implementation and testing of the DFM core is also presented. Test results on the DFM shows that it works within a reasonable margin of variance.

Keywords: Block-based design, reconfigurable hardware, virtual instrumentation, FPGA

Introduction

A high-level Reconfigurable Virtual Instrument (RVI) system architecture comprises both hardware and software sub-systems. The hardware includes a standard personal computer and the reconfigurable instrument (RI) connected to it through a physical connection. The connection allows the reconfiguration of the RI as well as the exchange of information between the PC and the RI [1]. The information exchanged can be data, commands, error messages, or acknowledgement messages. Interest in reconfigurable devices today is encouraged by the fact that manufacturers promote some of their products as dynamically reconfigurable [2]. The specification and design of such a system is done at the highest level of abstraction, register transfer level (RTL) using hardware description language (HDL). The commonly used design
style in such models is the ‘dataflow’ style. Here a larger number of concurrent HDL statements and small processes are used to implement the desired functionality. This makes code reading and understanding difficult since the concurrent statements and processes do not execute in the order they are written. To extract the functionality of dataflow code, a block diagram has to be drawn to identify the dataflow dependencies between statements [3].

In traditional model for ASIC development, the project transition is from phase-to-phase in a step function, never returning to the events of the previous phase. Such designs are handed over from one development team to the other without much interaction between them, and this poses a challenge [4]. In response to these challenges, designers have adopted a block-based design approach that emphasises design reuse (reusing pre-designed and verified blocks). In a typical example, to tackle team-based FPGA design, Troy Scott (Lattice Semiconductor Corporation) illustrates the application of modular FPGA design by a development team charged to implement a large communication design [5].

In this paper we propose the application of block-based approach to the design of RVI systems and present a particular RVI implementation based on this approach.

The paper is organized into the following sections. Section 2 presents a general concept on hierarchical, block-based design methodology and FPGA design methodology. The application of the block-based approach is presented in section 3. We highlight results of the approach in relation to the RVI in section 4 and conclude in section 5. We assume the reader is familiar with standard high-level design methodology including HDL design and synthesis, floor planning, physical synthesis and place and route. The terms block, core, macro, IP and module may be used interchangeably, and will refer to a design unit that can reasonably be viewed as a stand-alone subcomponent of a complete System-on-Chip (SoC) design.

Hierarchical block-based design

Hierarchical design is a methodology that divides a system recursively into small modules and then constructs each module independently. The major benefits of using this methodology is complexity management and design reuse. VHSIC Hardware Description Language (VHDL) provides powerful mechanisms and versatile language constructs to support hierarchical design methodology and to manage the design of large systems. To apply these features, the design hierarchy must be determined first and the system is divided into smaller parts. This is considered as design partitioning or block-based approach [6]. We refer to logical partition in this paper, as a means to make the design, development and verification of the RVI platform manageable, with respect to synthesis. Physical partitioning is another perspective which is not used in this work.

FPGA design methodology

FPGA design methodology supports both top-down and bottom-up design methodology. FPGA design flow support modular design approaches for bottom-up methodology, and hierarchical design partitioning for top-down design methodology,
similar to the process used for ASIC devices [7]. Software tools are important components in the application of this methodology.

**Block-partitioning of RVI hardware**

We discuss the implementation of a reconfigurable virtual instrumentation platform using a hierarchical, block-based approach. The systems target platform is an APA300 board, a ProASIC-Plus FPGA. In order to use the block-based approach effectively, the platform was partitioned into two parts, the virtual instruments (VIs) platform (reconfigurable part) and the communication interface (fixed part). Here each platform was implemented on a separate FPGA board, using a WISHBONE® bus [8] as the internal communication bridge. On the reconfigurable part, each virtual instrument (VI) was designed and implemented as a block, whiles the fixed platform has the various communication controllers as blocks. Each of these blocks in turn comprises sub-modules. The bus on each side of the platform was implemented as a block, together with the bus arbiter.

![FPGA Global Architecture for the RVI](image)

Figure 1: FPGA Global Architecture for the RVI

A block diagram of the global architecture for complex Reconfigurable Virtual Instrumentation system presented by Cicuttin et al in [1] is shown in figure 3.1. The

*Proceedings of The 2008 IAJC-IJME International Conference*
*ISBN 978-1-60643-379-9*
goal of this research is to develop an RVI system based on this architecture. The internal RVI bus allows exchanging data between the PC/FPGA communication block and each of the other blocks having a common interface. For critical communication between an instrument core and an external hardware a point-to-point fast dedicated connection is used. In Figure 3.2, we see the block implementation of a digital frequency meter, a function generator and a digital thermometer as a set of blocks (virtual instruments) on the reconfigurable platform. In the next section we show the application of the block-based design methodology in designing the digital frequency meter as an instrument core.

![Figure 2: RVI Hardware Platform](image)

**Results of application of the method**

The details of the application of the block-based approach to the implementation of the DFM instrument core are presented in this section. In this application, we use an Actel (Mountain View, CA, USA) ProASIC-Plus family FPGA, a flash based reconfigurable field programmable logic device. Like an ASIC, this device is non-volatile, “live” at power-up, offers a wide range of densities, and has very low power consumption. Its fine grained architecture enables designers to leverage existing ASIC or FPGA design flows and tools. This family is also supported by Actel’s Libero® Integrated Design Environment (IDE). Libero IDE provides an integrated design
manager that seamlessly integrates design tools while guiding the user through the
design flow, managing all design and log files, and passing necessary design data
among tools [9].

DFM design specification

The aim here is to design a digital frequency meter (DFM) to be integrated as a VI on
an RVI system. VHDL was used to specify the design at RTL level and a synthesis-
based coding to make the core of the design as technology-independent as possible.
The design is based on the signal-count method of determining an unknown signal
frequency. This method involves counting the number of rising edges (of the
unknown Clock or TTL signal being measured) that occur during a one second
window. The total of the number of rising edges counted is approximately the
frequency to be determined in Hertz (Hz). The range of frequencies to be measured is
from 2 kHz – 9 MHz. The signal to be tested must have an edge time \( t \) such that 10ns
\( \leq t \leq 25\text{ns} \).

Block-partitioning

The DFM block was partitioned into two main functional sub-blocks. The first sub-
block, ONESECOND_WINDOW (figure 4.1), generates a one second window flag
by counting the appropriate number of system clock cycles (onboard oscillator). The
second sub-block, COUNTER_FREQ (figure 4.1), counts the number of cycles the
TTL signal input (of unknown frequency) makes until the one second flag is asserted.
Counting is done in binary coded decimal (BCD). The BCD_7SEG_DECODER
(figure 4.1) and CLOCK_DIV (figure 4.1) sub-blocks operate by decoding the BCD
data to feed the 7-segment display on the I/O communication interface (figure 3.2).

Sub-block integration

VHDL was used to describe the various sub-blocks at the HDL entry level of the
Libero® IDE suite. Functional (pre-synthesis and post-synthesis) simulation was
performed on each sub-block as a measure of verification. A structural-style VHDL
was used to connect the various blocks after verification and re-synthesised as a
complete system. After successfully passing post-synthesis verification, layout (place
and route) and generation of the programming file (STAPL) was performed in that
order. The programming file was used by the device programmer to implement the
design onto the FPGA.

The resulting instrument core

The methodology and design flow described in section 2 was applied on each sub-
block used in this core. A bottom-up strategy was used during synthesis. Constraints
were set for the lowest block first and then, synthesis performed. Unlike previous
approaches where a block was tackled as one big unit, the division into sub-modules
provided a manageable approach to accomplishing the design task.
Using a standard function generator as a source of signal input, the DFM instrument was tested (based on the specifications in section 4.1). Table 4.1 shows the mean of readings during the test measurements. This was done for a range of randomly selected frequencies from the kilo-hertz (kHz) region to the mega-hertz (MHz) region. Figure 4.2 shows a picture of the test setup.

Table 1: Sample Data of Test Readings using External Function Generator

<table>
<thead>
<tr>
<th>Function Generator (kHz)</th>
<th>DFM Mean Measurement (Hz)</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4,001</td>
<td>2.72</td>
</tr>
<tr>
<td>10</td>
<td>10,001</td>
<td>2.46</td>
</tr>
<tr>
<td>15</td>
<td>15,001</td>
<td>4.18</td>
</tr>
<tr>
<td>256</td>
<td>256,002</td>
<td>3.12</td>
</tr>
<tr>
<td>1,000</td>
<td>1,000,001</td>
<td>4.77</td>
</tr>
<tr>
<td>2,500</td>
<td>2,500,001</td>
<td>3.51</td>
</tr>
<tr>
<td>3,000</td>
<td>3,000,000</td>
<td>3.07</td>
</tr>
<tr>
<td>8,000</td>
<td>8,000,000</td>
<td>1.73</td>
</tr>
</tbody>
</table>
Conclusion

Although the concepts of virtual instrumentation and reconfigurable computing are not new, implementation of a complete RVI system based on FPGA technology at a reasonable hardware cost/performance ratio is yet to be reported. This paper has shown that such implementation is achievable. The use of a hierarchical, block-based approach in the development of reconfigurable virtual instrumentation platform has been applied. The details of a particular implementation of a DFM core as part of an RVI hardware platform have been presented. The core has been tested and found to perform within a reasonable margin of variance.

Acknowledgement

We wish to acknowledge the contributions made by the members of MLAB of The Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy and Ms. Boadiwaa Ofori Amoateng (BSc. Computer Engineering, KNUST, 2007).

References


Biography

GILBERT OSEI-DADZIE received his BSc. (Physics, 2004) from KNUST, majoring in Opto-Electronics and Material Science. He is currently a post-graduate research student at KNUST (Electronic Engineering). His current work is on the application of Field Programmable Gate Array (FPGA) in reconfigurable virtual instrumentation. He also has interests in Digital Systems, Computer Networking and Programming. He is an associate member of the Institute of Engineering and Technology (IET).

KWAME OSEI BOATENG is a senior lecturer of Computer Engineering at Kwame Nkrumah University of Science and Technology (KNUST), Kumasi, Ghana. He earned his BSc. (Electrical/Electronic Engineering, 1991) from KNUST, Master’s degree (in Computer Science, 1997) from Ehime University, Japan and Doctoral degree (in Information Systems Engineering, 2000) from Ehime University, Japan. Dr. Boateng is a member of the Institution of Electrical and Electronic Engineers and Computer Society (IEEE). Dr. Boateng has worked as a researcher at Fujitsu Laboratories Ltd., Japan and is an ICT consultant for KNUST.