Computers and Electrical Engineering 35 (2009) 790–802

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A cycle-accurate transaction level SystemC model for a serial communication bus

Syed Mahfuzul Aziz *

School of Electrical and Information Engineering, University of South Australia, Mawson Lakes Campus, SA 5095, Australia

ARTICLE INFO

Article history:
Received 29 March 2006
Accepted 19 November 2008
Available online 25 April 2009

Keywords:
Transaction level modelling
Cycle-accurate model
SystemC
Serial communication
1553 data bus

ABSTRACT

This paper presents a transaction level SystemC model of an avionics mission system data bus that provides cycle-accurate simulation of the bus. The mission system is a complex distributed computer network consisting of a mission control computer, radars, an array of subsystems and sensors. The data bus plays a critical role in the system as it carries all the information between the system components. Therefore modelling the bus at an appropriate level of abstraction using appropriate technology is important for evaluation of the performance of both the bus and the entire system. While models based on traditional hardware description languages (HDL) provide cycle-accurate performance estimates they are very slow and have high code complexity. In order to enhance model performance this paper presents a transaction level model (TLM) utilizing the enhanced SystemC features and levels of abstraction. The TLM model incorporates a clock-based synchronisation strategy thereby providing cycle-accurate performance estimates like the HDL models. The developed model has been validated for various payloads and system sizes. Simulation results show that the proposed SystemC transaction level model is much more efficient than models developed using conventional hardware description languages.

1. Introduction

SystemC is a high level Hardware Description Language (HDL) that comprises of C++ class libraries and a simulation kernel [1]. SystemC inherits the features of the C++ programming language. The Object-Oriented features of the C++ language helps in modelling a complex system in an easier way and allows the designer to analyse the performance of the model [2]. Also the concept of design reuse in SystemC reduces the modelling time compared to that of traditional hardware languages like VHDL [3] and Verilog [4]. It was reported in [1] that a SystemC model was able to achieve a speed ten times faster than a VHDL model at the same level of abstraction. SystemC is also considered to be more powerful for design validation when compared to VHDL and Verilog [5].

The ability of this language to incorporate different levels of abstraction within the same model provides flexibility in separating the functional specification of a system from the communication specification [6]. For the same system implemented in VHDL, the level of abstraction within a model is restricted to the structural domain [7]. In SystemC abstraction is supported by the C++ features like polymorphism, inheritance, template, class and objects. In VHDL the lack of separation between communication and functional specification makes the system design more complex while the reuse of designs is restricted to component level [1].

This paper presents a transaction level model (TLM) of a serial data bus in SystemC incorporating a clock-based synchronisation strategy to ensure cycle-accurate performance estimates. The model conforms to the military standard MIL-STD-

* Tel.: +61 8 83023643; fax: +61 8 83023384.
E-mail address: mahfuz.aziz@unisa.edu.au.

0045-7906/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2008.11.029

1553. The data bus is the main communication component in an Avionics Mission System (AMS). An AMS is considered to be a real-time embedded system which includes standard and custom software and hardware [8]. In this paper we first briefly describe the communication protocol and the associated standard in Section 2. The modelling approach and some important features of SystemC supporting this approach are discussed in Section 3. The transaction level SystemC model of the serial data bus and the state diagrams are presented in Sections 4 and 5 respectively. Section 6 presents the testbench developed for simulation of the model and the specific SystemC features used for measurement of bus performance. Section 7 presents the simulation results and Section 8 concludes the paper.

2. The 1553 serial data bus

MIL-STD-1553 is a Digital Internal Time Division Command/Response Multiplex Data Bus [8] used in the AMS of many aircrafts. It is a standard developed by the US Department of Defence. The main functionality of the serial data bus is to communicate, store and manage the aircraft avionics [9]. The main components of the serial data bus are the Bus Controller (BC) and Remote Terminals (RT). These components communicate at a data rate of 1 Mbps as specified by the standard. The function of the BC is to provide data flow control for all communications taking place on the bus. The RTs do not have any data flow control capability. Their function is to provide the interface and data transfer to and from the various subsystems under the control of the BC. Each RT can have a maximum of 30 subsystems attached to it. There can be a maximum of 31 RTs connected to a single data bus [9].

The communications between these components are based on three different message types: BC to RT, RT to BC and RT to RT. The format of only the BC–RT message type is shown in Fig. 1 [9] for the purpose of illustration. Each message is composed of command, data and status words, each being twenty bits long. Each word includes a parity bit for error control and three sync bits to differentiate between the various words [9]. The command word is issued by the BC to initiate communication. The status word is used for acknowledgement, i.e., to obtain information on whether the RT is transmitting data or has received data. The number of data words that can be transmitted in one message cycle is restricted to thirty-two. As mentioned previously, the communications between the BC and the RTs follow a command/response protocol, similar to the master/slave polling arrangement [10]. For example in a BC to RT message type, the BC first sends a poll message (command) to a particular RT to begin receiving data. The BC then continues by transmitting the data words onto the bus. While all the RTs listen to the command, only the RT that is polled processes the command and receives the data. Once the RT receives all the data (assuming an error-free channel) it responds to the BC with a status message. The details of the other message types are similar to the above description but with some distinct differences, and can be found in [9]. The serial data bus is classified into two topologies: single and multiple levels [9]. Fig. 2 presents the single level topology of the bus.

3. Modelling approach

The aim of the modelling approach is to develop a model, for critical serial communication busses in distributed systems such as the Avionics data bus, that provides a high simulation performance and accurate estimation of the bus performance (e.g. transaction time). We therefore aim to develop the model at the Transaction Level, which provides a higher level of abstraction than the RTL level. The model will focus on communications at the message (transaction) level rather than at the bit or word level. The details of the bit or word level transactions will therefore be hidden allowing higher efficiency in coding and in the performance of the resulting model. Nevertheless we want our model to provide *cycle-accurate* estimate of the bus performance. We therefore use a clock to synchronise the transactions within our Transaction Level Modelling (TLM) framework. This unique combination of transaction level modelling with clock-based synchronisation is similar to the TLM model defined in [11] as the bus-functional model. In [11] four alternative transaction level models have been proposed based on the concept of a channel defined in [12]. The bus-functional model has been defined to contain cycle-accurate communication and approximate-timed computation. It is still expected to provide a much higher level of abstraction than the RTL level because it separates the communication among the computation components from the details of the computation [11]. Examples of Bus-Cycle-Accurate (BCA) transaction level models of the AMBA bus architecture based on similar concepts have been presented in [13]. These TLMs developed in SystemC provide much higher simulation performance than SystemC RTL models and TLM models written in C [13]. Although our transaction level model of the Avionics data bus uses clock-based synchronisation the synchronisation details are abstracted away by using *blocking* conditions [14]. These are conditions that wait for a time-dependent event to restart the block of code, typically a transaction. The proposed modelling

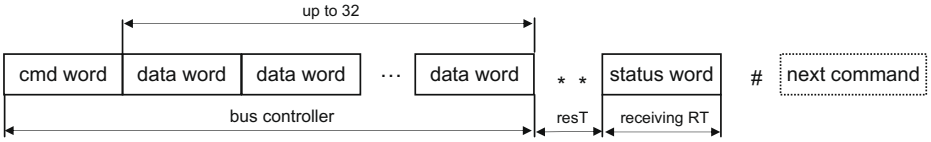

Fig. 1. Data exchange format for the BC to RT message.