
Antonio Cerone

Software Verification Research Centre,
The University of Queensland, Brisbane QLD 4072, Australia
antonio@it.uq.edu.au,
http://svrc.it.uq.edu.au/pages/Antonio_Cerone.html

Abstract. In this paper we present two different approaches used in specifying a well-known audio control protocol with real-time characteristics. The first approach is based on Circal, a process algebra that permits a natural representation of timing properties and the analysis of interesting aspects of timing systems. The second approach is based on the Timed Interval Calculus, a set-theoretical notation for concisely expressing properties of timed intervals. The comparison between the two approaches shows that they are almost complementary: the former allows an easy modelling of the most procedural aspects of the protocol and provides a fully automatic proof but cannot catch all timing aspects; the latter allows easy modelling of all timing properties but the proof is quite hard and cannot be fully automated. This suggests a decomposition of the proof into subproofs to be performed in different proof environments.

1 Introduction

The case study presented in this paper is a time-sensitive audio control protocol developed by Philips. This protocol is intended to be used in future high fidelity systems where control signals pass between the various components of the system such as an amplifier, CD players, speakers, radio, etc., via a bus. The presence of such a control network adds additional features to the system, but market forces require that this extra functionality is achieved at minimal costs with a minimum of additional electronic components. Most of the modules in such an audio system contain a microprocessor and it is these microprocessors which will be used to realise a hopefully robust communication mechanism between them.

The protocol is an important benchmark case study which has challenged a number of formalisms. A version of the protocol with only one sender and one receiver was previously specified and analysed by Bosscher et al. [2] using linear hybrid systems. In subsequent papers by other authors, properties stated and proved by hand [2] have been automatically verified by using model checking based tools, such as Kronos [8], HyTech [11], Uppaal [13] and the Circal System [4,5], and theorem provers, such as the Larch Prover [10]. Recently the protocol has been modelled in a real-time process algebra by Chen, who has used a weak bisimulation to manually verify its implementation against its specification [7].
In the current paper two very different approaches used in specifying the audio control protocol are directly compared. In the first approach the protocol is modelled using the Circal process algebra. Although Circal is a discrete formalism, it permits a natural representation of timing properties and the analysis of interesting aspects of timing systems. The automated verification of the protocol is entirely performed within the process algebra framework using a model-checking technique without recourse to temporal logic.

In the second approach the protocol is modelled using the Timed Interval Calculus (TIC). The formal verification has been carried out by hand, but it is currently being automated through an implementation of the Timed Interval Calculus within the Ergo theorem prover.

We show that the two approaches are complementary in terms of complexity of the specification. The sender can be easily specified in an algorithmic fashion using the process algebra, but requires the definition of a complex infrastructure to build up the axiomatic specification in TIC. On the other hand, the time based decoding by the receiver can be immediately specified in TIC but is complex in Circal. Moreover, the process algebra cannot characterise a deadline for the decoding, due to the state explosion caused by the tick actions, whereas deadlines can be trivially defined in TIC. Finally, the correctness proof is simple and can be fully automated in the process algebra, whereas is quite complex in TIC.

2 Description of the Protocol

The protocol forms part of the physical layer of an interface bus that interconnects the components of stereo systems. Messages, which consist of finite sequences of 0 and 1 bits, are encoded by a sender into timed transitions of the voltage between two levels on the single wire bus connecting the components. Receivers in components attached to the bus interpret the voltage transitions and reconstruct the bitstream messages. The senders and receivers run on different microprocessors. Since the microprocessors run code in addition to the protocol software, sender and receiver clocks may not be synchronised in frequency.

We restrict our example to only two processors communicating over a bus, one sender and one receiver, each with its own independent clock. In this way problems of bus collisions due to different senders sending at the same time are avoided. We also suppose that the delay on the bus is negligible and that the sender transmits just one message.

Figure represents this view of the protocol. The sender processor of the audio control protocol accepts a message from a user which it then tries to send to the receiver processor over the bus. A message is encoded by the sender according to a Manchester encoding scheme which is an algorithm for constructing a bus signal equivalent of the message. The time flow is represented as a sequence of slots of the same length, \(4 \times Q_S\) (where \(Q_S = 2.22 \times 10^{-4}\) sec in the Philips protocol), and each bit of the sequence is sent by the sender as a voltage level transition in the middle of a slot: a bit 0 is a falling transition and a bit 1 is a rising transition (see Figure).