Monitoring and Analysis of Bus Data on a Prosthetic Limb System Using FPGA and Microcontroller Devices

Marcel Dombrowski, Kenneth B. Kent, Yves Losier,
Adam Wilson, Rainer Herpers
University of New Brunswick, Bonn-Rhein-Sieg University
Faculty of Computer Science, Institute of Biomedical Engineering,
Department of Mechanical Engineering
marcel.dombrowski@unb.ca, ken@unb.ca, ylosier@unb.ca,
awilson@unb.ca, rainer.herpers@h-brs.de

Outline

- Implementation of a bus monitoring tool using FPGA and microcontroller devices.
- Ability to interpret PDCP actions on ISO OSI layer 3.
- Output of generated data in a human readable format.

Motivation

UNB Institute of Biomedical Engineering (IBME) develops together with different research partners a prosthetic limb device, the UNB Hand. The different sensors and actuators inside the hand communicate via a CAN bus. On top of the CAN bus the IBME developed Prosthetic Device Communication Protocol (PDCP) is built. As the bus system inside the hand is running nearly at the bandwidth limit (1 Mbit/s) monitoring and analysis capabilities are needed in order to detect each module's individual load on the bus.

There are existing solutions available for monitoring the CAN bus, but they either lack realtime capabilities or the ability to monitor different modules at the same time. The goal is to develop a FPGA solution which monitors the CAN bus (ISO OSI layer 1 and 2) and a microcontroller solution which interacts with the FPGA and is able to interpret the messages on the bus according to the PDCP (ISO OSI layer 3).

Background

CAN bus

- Message-based protocol, 4 different message types: data frame, remote frame, error, frame, overload frame.
- Each data frame has either an 11 bit (SID) or a 29 bit (EID) identifier field and can support the transmission of up to 8 bytes of data.
- 15 bit CRC for error detection in the data field.

PDCP

- Serves as abstraction of the underlying bus system.
- Enables point-to-point connections of different nodes in the bus.
- Every device needs to register with the Bus Arbitrator.
- Different functions are specified in order for the devices to communicate.

Problem

Every node in a PDCP system gets its ID assigned by the Bus Arbitrator. Thus every message needs to be logged and interpreted in order to determine which nodes are communicating with each other.

Solution

- The microcontroller needs to be capable of logging and interpreting data from the CAN bus. Thus it needs to implement the full CAN specification.
- It needs to be able to generate usage statistics and save them in a human readable format on a SD card.
- A FPGA device should act as a helper and transmit bus load data to the microcontroller.
- To be able to interpret the PDCP data it needs knowledge about the devices in the system.

Results

A microcontroller with the ability to read and write from the CAN bus that registers itself with the Bus Arbitrator upon connection to the bus has been designed and implemented. It is capable of receiving all messages on the bus and interpreting them according to the PDCP. Based on the knowledge of the different modules in the system it can generate statistics on the usage of the module's communication on the bus and save these statistics as well as logging data onto a microSD card.