Designing and Implementing Field instruments Emulator for Supervisory Control and Data Acquistion (SCADA) System Testing

Undergraduate Final Project

Andi Nurrachmat
University of Hasanuddin – Indonesia

Abstract

Our industrial partner was looking for an easy way for conducting either a SCADA (Supervisory Control and Data Acquisition) system test or an interactive SCADA training for their engineers and customers so that the participants can absorb the material training as much as possible.

The goal of the project is to address that challenge above by developing a Field Instruments Simulator that able to demonstrate the behavior of field instruments connected to SCADA system. The field instruments that simulated consist of the collection of sensors (pressure, temperature, gas flow, open-closed valve, and salinity), actuator and alarm.

The simulator is designed upon software (SW) and hardware (HW) parts. The role of HW part is to interface between PC and SCADA components such as Remote Terminal Unit (RTU), Master Control, and Man Machine Itnerface (MMI). The design of interface is composed by some modules, that is 8-bits AD/DA converter, PPI-8255, conditional circuit based on OP-Amp 741 and power supply. Whereas the SW application part comprises 2 objects, namely: device driver which applies the principle of memory mapped I/O for driving the interface, and the GUI.

For the test and verification, we had adopted the ARCO's project as our case study. In the project it was installed the SCADA system to control and monitor the distribution of gas flow in the pipelines network from offshore crossing to the Java and Bali islands in Indonesia. As project's scenario is, the simulator was programmed to represent several part of real sensors, actuators and fire alarm that connected to the RTUs, and Master Control of SCADA.

Design and Implementation to Femto Java Processor on FPGA Xilinx XC4010 XL

Master in Microelectronic Thesis

Andi Nurrachmat
Bandung Institute of Technology – Indonesia

Abstract

With the philosophy "Write once and Run everywhere", Java becomes famous language because offers many advantages such as portability, simpler, safe object references, the notion of concurrency as a first-class language construct. However, either desktop or embedded application developed with Java technology has lack in performance during the execution time. The cause of lack is the need to build on-the-fly the Java Virtual Machine (JVM) on top of traditional processor. The consequence of the existence of virtual machine is time and memory consuming. The thesis proposes a radical way to overcome the lack by designing and implementing the virtual machine into a real machine called Femto Java on a dedicated FPGA hardware.

Femto Java directly executes Java instruction set (bytecode) without wasting the time of converting the bytecode into general processor-based native instruction set. Since JVM specification is stack-based machine, the Femto Java architecture has been designed to work as the stack-machine model instead of RISC or CISC machine. The benefit of stack architecture are to shorten the execution time of Java application by accelerating calls method process as well as the context switching and error handling when interrupt and exception occur. The Femto Java design is targeted to bring the JVM specification into reality as many as possible so that it can act as general-purpose processor for Java application.

The elements of JVM had been implemented are registers for method area register, operand stack, and local variable, execution environment and all of pointers (**OPTOP, PC, VARS**, and **FRAME**). For the JVM instruction set, Femto Java recognizes push constant type, load/store, stack manipulation, conditional branch, arithmetic and logic-categorized instruction set. The prototype of Femto Java was successfully implemented onto a FPGA-Xilinx XC4010XL.

As regard for the test and verification, Femto Java has passed 2 phases of test; that is functional test and timing simulation and then verifying the result of arithmetic-oriented Java class file that loaded into Femto Java's memory (RAM) and executed directly by Femto Java and then monitoring the result on the LED panel.

Implementation of Small Virtual Private Computer Network (based on IPSec)

Master in Embedded System Design Thesis

Andi Nurrachmat
ALaRI, University of Lugano – Switzerland

Abstract

In order to communicate securely, two network hosts must be able to create a secure channel over an initially insecure medium. This requires that a secure key exchange and negotiation mechanism have to be in place. This project is a part of ALaRI's smart card Master Project, which the main goal is to provide a security way of communication between two or more peers by establishing a Virtual Private Network (VPN) based on IPSec.

Pluto is one implementation of open-source IPSec in application layer and it does not support to work with the Smart Card System which relies on the AES and Elliptic Curve Cryptography algorithm. Due to that lack, Pluto should be modified in order to be used by the Smart Card system. The most important modification should be addressed is to define the steps for key exchanging and key management.

The proposed steps covers the establishing of a Security Association between two peers, the agreement for applying the compatible policy and cryptography algorithms and sharing of secure mechanism for determining key material exchange over an insecure tunnel.