### **Design of a TP-Jop co-processor**

Endrico Bester, Noah Miller, Diwakar Somu,

Department of Electrical and Computer Engineering

University of Auckland, Auckland, New Zealand

Abstract

The design of a highly integrated tandem processor “TP jop” is discussed in this report. TP-JOP is

# Introduction

With ever increasing technological advancement into smaller and faster computer systems, comes more application specific system designs. These application specific designs are more and more turning to the concepts of System-on-Chip (SoC) and Network-on-Chip (NoC) as their architecture model. Various types of these SoCs have been developed over the years, one of which is the Globally Asynchronous Locally Synchronous Tandem Processor – Java Optimised Processor (GALS TP-JOP). Due to the nature of this processor it falls in the category of Multi-Processor on Chip (MPSoC) because it contains two processors running in tandem with one another.

The aim of this project was to design and implement this processor architecture using a Fie

* Tools used
  + Talk about modelsim as tool for simulation
  + Quartus tool for design
  + SystemC
  + Assembly writer
* Talk about the processor being globally asynchronous locally synchronous (GALS).

The design of a complex multicomponent processor is described in the following sections. In this project The design of the Tandem processor (TP-JOP) was split into three distinct parts. The following sections will describe The individual phases in terms of their design goals the objectives and the progress achieved.

## Related Work

* Talk about previous implementation with NIOS JVM and reference the report found on uni database(Endrico has copy in dropbox)

# The Design Process

The project was divided into three phases, and after each of these phases the total progress made toward the end product was determined. The first of these phases was essentially the design stage in which the major components of the ReCop were to be designed.

SystemC was the desired designing language as it enables faster, higher level design capabilities.

## SystemC

* Designing of the datapath during this phase
* Control unit

## Multi-Cycle Datapath

* Talk about different components within the datapath the were designed

## Control Unit Design

* Structure of the control unit.
  + Using an FSM
  + Type of FSM
  + State progression
* Assigning the control signals to each state
* Add in a state diagram

## Simulation

# Creating a Synthesizable Design

## Converting SystemC to VHDL

* Progressing of the design of the datapath
* Control unit

## Addition of Essential Components

## Simulation

# Full Integration with JOP

## VHDL Finalisation

* Final Datapath design
* Changing of the control unit to fit the interface with the jop as well as the change of the processor top level entity to work with the jop and simcon interfaces

## SimpCon

## Using arbiter to connect two CPU’s together as both act as masters

## Testing

# Discussion

## Difficulties Faced

* Simpcon connection

# Conclusions

* Using SystemC as a design medium for ease of design concept validation

# Acknowledgements

# References

1. Smith, J. O. and Abel, J. S., ``Bark and ERB Bilinear Trans­forms'', IEEE Trans. Speech and Audio Proc., Vol. 7, No. 6, pp:697-708, 1999.
2. Lee, K.-F., Automatic Speech Recognition: The Development of the SPHINX SYSTEM, Kluwer Academic Publishers, Boston, 1989.
3. Rudnicky, A. I., Polifroni, Thayer, E H., and Brennan, R. A. "Interactive problem solving with speech", J. Acoust. Soc. Amer., Vol. 84, p S213(A), 1988.