Investigating Interrupts in the Xtratum Hypervisor using CSP

By Kevin Hennessy
Supervisor Dr, Andrew Butterfield

Extended Abstract

My thesis involves investigating interrupts on the Xtratum hypervisor and the Sparc v8 based LEON3 platform. To do this I build a simulation of the system using Communicating Sequential Processes (CSP) with the FDR3 model checker. This sytem contains elements of Xtratum, LEON3 and SPARC V8 and focuses mainly on functionality related to interrupts. I then run a series of experiments to demonstrate how high level properties of my simulation can be checked, and discuss how this could be used to verify properties the real Xtratum C code.

This work is done in the bigger context of a European Space Agency effort to certify the Xtratum LEON3 system for Time and Space Partitioning, a time and memory sharing system for space software.

- In Chapter 1 of my thesis I explore Integrated Modular Avionics and how that relates to the space focused Time and Space Partitioning.

- In Chapter 2 I discuss backgound and related work related to the verification of micro-kernels such as sel4 and VAMP.

- In Chapter 3 I give an overview of the modelling tool that I am using, Communicating Sequential Processes with the FDR3 model checker.

- In Chapter 4 I give an overview of my simulator and the modelling choices that I made as they relate to Leon3. Xtratum and Sparc v8. I explore the following@
    -

- In Chapter 5 I define four experiments which can be used to reason about high level properties of my simulator.
    - Experiment 1 was designed to demonstrate the use of a checker process that can be used to test for the presense of traces. This involved tracking memfault interrupt and deadlocking on its occurance.

    - Experiment 2 was designed to demonstrate how the ESA requirement PK-230 that all partitions must always be executing in user mode might be checked. This was done by tracking any interrupts that occur when a supervisor instruction is executed in user mode. CSP excels as a verification tool here, allowing us to quickly test a problem that would be prohibitively comples in the real system.

    - Experiment 3 was designed to further investigate the PK-230 requirement and show how CSP can be used to exhaustively check an aspect of a model without explosively growing the state space by clever use of the CSP choice operators. This experiment showed the advantages of using CSP for a project like this.

    - Experiment 4 was designed to demonstrate how high level conceptual models can be projected onto my simulator. I performed an experiment which linked partitions driven by an interrupting scheduler to a memory model via an intermediate link, the configurable Memory Management Unit. I used this to show how requirements sucj as PK-3 and PK-4 might be verified in the future using my simulation tool.

- In Chapter 6 I conclude and recap.

- In Chapter 7 I explore further work that could be done in this area to link the Xtratum C code to the ESA requirements using a formal proof.