

## **The Semantics of Power and ARM Multiprocessor Machine Code**

Multiprocessors are now dominant, but real multiprocessors do not provide the sequentially consistent memory that is assumed by most work on semantics and verification. Instead, they have subtle relaxed (or weak) memory models, usually described only in ambiguous prose, leading to widespread confusion. Authors develop a rigorous and accurate semantics for ARM multiprocessor programs, from instruction decoding to relaxed memory model, mechanised in HOL. Authors test the semantics against actual processors and the vendor litmus-test examples, and give an equivalent abstract-machine characterisation of our axiomatic memory model. For programs that are (in some precise sense) data-race free, we prove in HOL that their behaviour is sequentially consistent. This provides a solid intuition for low-level programming, and a sound foundation for future work on verification, static analysis, and compilation of low-level concurrent code.

### **Reference:**

[The Semantics of Power and ARM Multiprocessor Machine Code](#)