DEVELOPMENT OF COMPUTER ARCHITECTURES FOR PROLOG
- Final Report

David H D Warren
Artificial Intelligence Center
SRI International
31 August 1983

1. Summary of Work Done

- Design of a byte-coded Prolog instruction set for a goal stacking model ("old Prolog Engine").

- Implementation of a prototype emulater for the Prolog instruction set in a macro language Progol which is mapped into VAX machine code by a preprocessor; this emulater is intended to serve as a model for a firmware implementation.

- Peformance testing of the prototype emulater on a VAX-780, showing a speed of 7,900 lips for determinate concatenate; some straightforward improvements would increase the speed to roughly 15,000 lips, which is about the limit we can forsee for a software bytecode emulator on the 780.

- Transliteration of the emulater into C.

- Implementation of a prototype compiler to translate Prolog into the Engine instructions.

- Redesign of the Prolog instruction set for an environment stacking model ("new Prolog Engine"), in order to overcome certain difficulties with the goal stacking model.

- Design of a pipelined Prolog processor for the new instruction set, with hand timings of sample execution traces.

- Design of a scheme to exploit or-parallelism in Prolog programs in the context of a multi-processor machine with shared random access memory.

- Transportation of C-Prolog to VMS, plus various bug fixes and improvements, as requested by DEC.

2. List of Documents Produced

- Notes: "Prolog Engine" [David Warren]
- Report: "An Abstract Prolog Instruction Set" [David Warren]
- Paper: "Towards a Pipelined Prolog Processor" [Evan Tick & David Warren]
- Report: "An Overlapped Prolog Engine" [Evan Tick]
- Notes: "Design of a Multi-Processor Prolog Machine Exploiting Or-Parallelism" [David Warren]