

Development of Fuce Processor Emulator on Multiple FPGA Chips

Masaaki Izumi, Takanori Matsuzaki,
Satoshi Amamiya, Makoto Amamiya

Graduate School of Information Science and Electrical Engineering,
Kyushu University

We discuss the design policy of a Fuce processor emulator using multiple FPGA chips. The Fuce processor aims at efficient processing of external events and internal computations for the future communication networks. The Fuce processor unifies both external event processing and internal computation as a thread execution. Each thread is executed as a non-preemptive sequence of instructions. The Fuce processor has multiple Thread Execution Units (TEUs for short), and the multiple threads are executed in concurrent way. Thus, the Fuce processor exploits thread level parallelism (TLP).

The Fuce processor is a chip multiprocessor equipped with multiple TEUs. The Fuce processor is designed on the model of continuation-based multithread execution, which is an advanced version of the conventional dataflow computation model. Continuance is defined as computation continuation and data transfer between threads. Each thread execution becomes ready when all of continuances are notified from its preceding thread executions. Thread Activation Controller (TAC for short) is a hardware implementation of the continuation-based multithread execution model. TAC drastically reduces the overhead of multithreaded execution compared with the software implementation, and Fuce processor makes it possible to develop more feasible cost/performance TLP processor.

TEU is constructed with a pair of execution unit and preload unit. The execution unit is a simple RISC core, and executes register-to-register instructions. The preload unit supports only the load instructions that set up thread context. TEU has two register files; One is used by the execution unit for the current thread execution, and another is used by the preload unit to set up the context for next thread execution. The execution unit and preload unit run in parallel. When the thread execution switches, each register file alternates its role. This mechanism reduces the memory access latency.

In order to verify the behavior and to evaluate the performance of the Fuce processor, a Fuce processor emulator is designed using multiple FPGA chips. We can also use software simulators for gate level simulation and event level simulation, and we have developed such software simulators. However, if we want to perform more accurate simulation using a software simulator for practical processor in gate level, it consumes more time for simulation. Or, if we want to perform the simulation in shorter time, it will be less precise. Therefore, we have to develop the Fuce emulator by FPGA which will achieve high accurate design data with short measurement time. Furthermore, the FPGA emulator can be used as a test-bed for the custom-made chip development.

In the presentation, we show the design policy of the Fuce processor emulator on multiple FPGA chips. The emulator is constructed so that the design parameters of Fuce processor can be changed easily. The design parameters include the number of TEU, size of TAC, memory hierarchy and clock cycles of the memory access latency. We discuss the problem concerning the emulator design of chip multiprocessor on multiple FPGA chips; how to resolve the bus neck between FPGA chips and how to allocate the Fuce processor units on multiple FPGAs which are connected with narrow bandwidth buses.