Control Mesh Network

MIRABILIS
design

Product
Control mesh network in the Desktop Processor

Business
X86 Mobile and Desktop Processor manufacturer

Product Overview
The control plane mesh is the communication network that handles the coordination, control, power request and all not synchronization across all the masters and slaves on the processor. This was new generation of the processor, and several new features were integrated into the mesh. This includes a new Network-on-Chip, new types of requests, multi-core distributed processing, concurrent operations and larger number of request variations. There are new rate limiters, arbitration requests and scheduling schemes in the new mesh.

Project Team Background
Chief Architect and verification engineers

Project Challenges
- Mesh controls the processor throughput and depends on timing deadline of the request response.
- Understand bandwidth requirements on wires based on priorities, virtual circuits, and scheduler.
- Define the scheduling algorithm and priority of different tasks based on the timing requirements.

Results
- Defined Network-on-Chip configuration- flit size, wire throughput, clock domains, buffers, schedulers, rate limiters and routing table.
- Ran traces to validate the configurations for different use cases and traffic rates.
- Complete the entire modeling effort in three months and start the development.

VisualSim Solution
- Simulated the control mesh with trace input, traffic generators and statistics output.
- VisualSim Libraries used are NoC, distribution-based traffic, trace file reader, memory, power library, System Resources and Queues.
- Supported a NoC with 30 routers connecting 45 masters and slaves at different clock rate.
- Used trace files to stress test the design for complete system validation.
- The main reason for VisualSim was immediately availability of models, ease-of-customization, ability to run distributed simulation, and the accuracy of the design.
- Exploration used the reports and statistics generated for latency for each request, the buffer usage at each Router, and throughput on the wires.
- Detected throughput bottlenecks and buffer overflow for various use-cases and traces