2. System Configurations

The R10000 processor provides the capability for a wide range of computer systems; this chapter describes some of the uni- and multiprocessor alternatives.

2.1 Uniprocessor Systems

In a typical uniprocessor system, the System interface of the R10000 processor connects in a point-to-point fashion with an external agent. Such a system is shown in Figure 2-1. The external agent is typically an ASIC that provides a gateway to the memory and I/O subsystems; in fact, this ASIC may incorporate the memory controller itself.

If hardware I/O coherency is desired, the external agent may use the multiprocessor primitives provided by the processor to maintain cache coherency for interventions and invalidations. External duplicate tags can be used by the external agent to filter external coherency requests.

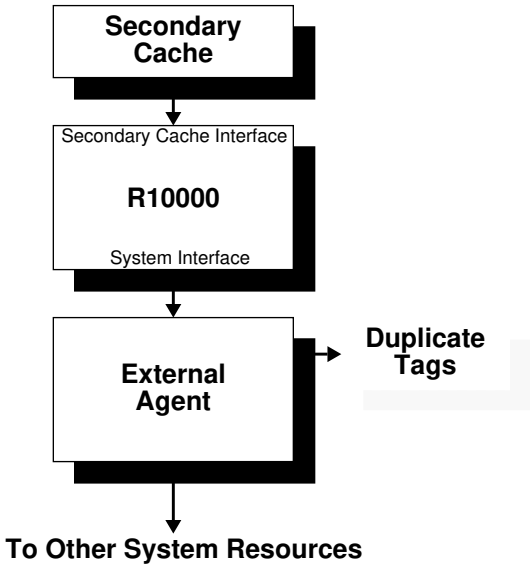

Figure 2-1 Uniprocessor System Organization

2.2 Multiprocessor Systems

Two types of multiprocessor systems can be implemented with R10000 processor:

- a dedicated external agent interfaces with each R10000 processor
- up to four R10000 processors and an external agent reside on a cluster bus

Multiprocessor Systems Using Dedicated External Agents

A multiprocessor system may be created with R10000 processors by providing a dedicated external agent for each processor; such a system is shown in Figure 2-2. The external agent provides a path between the processor System interface and some type of coherent interconnect. In such a system, the processor provides support for three coherency schemes:

- snoopy-based
- snoopy-based with external duplicate tags and control
- directory-based with external directory structure and control

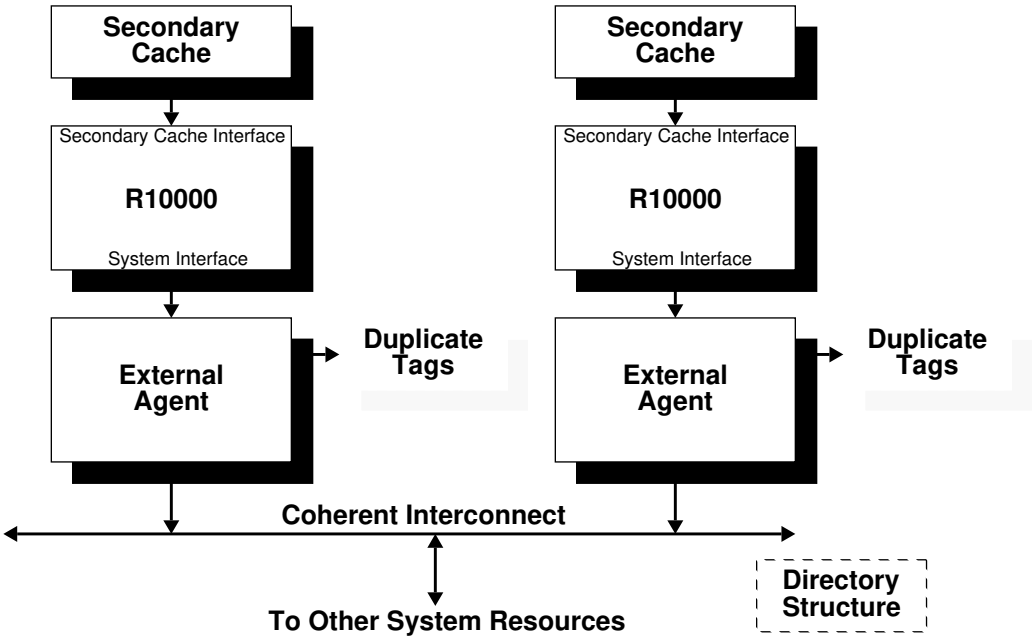

Figure 2-2 Multiprocessor System Organization using Dedicated External Agents

Multiprocessor Systems Using a Cluster Bus

A multiprocessor system may be created with R10000 processors by using a cluster bus configuration. Such a system is shown in Figure 2-3. A cluster bus is created by attaching the System interfaces of up to four R10000 processors with an external agent (the *cluster coordinator*). The cluster coordinator is responsible for managing the flow of data within the cluster.

This organization can reduce the number of ASICs and the pin count needed for a small multiprocessor systems.

The cluster bus protocol supports three coherency schemes:

- snoopy-based
- snoopy-based with external duplicate tags and control
- directory-based with external directory structure and control

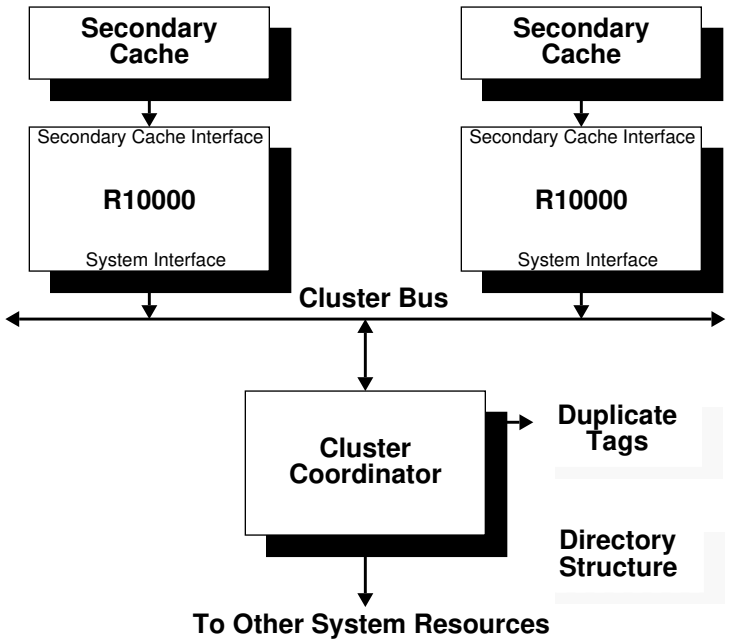

Figure 2-3 Multiprocessor System Organization Using the Cluster Bus