Network Interface Buffer Elimination

checkpoint 2

Jiong Xue Qilu Guo Jing Ji

Project overview

The project aims to eliminate the network interface buffer to reduce area for a NoC. We propose a solution where the packet delivered is preserved on the cache instead of the network interface. Thus, only the head and tail flits of the packet need be stored in the network interface. We modified the communication scheme that cache would directly transmit the data to and from the router. Both transmissions require interference with network interface.

A high-level architecture design is shown in Figure 1 below.

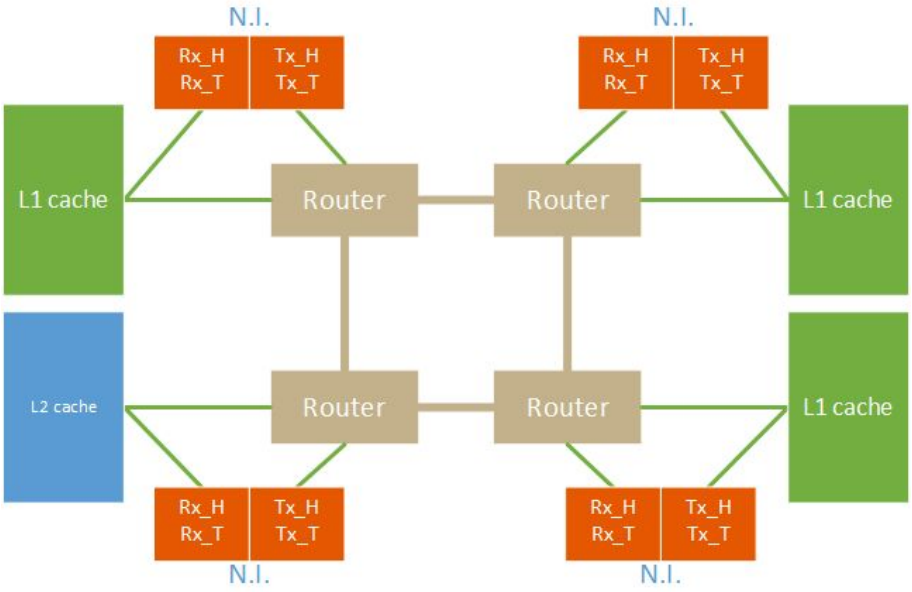

Figure 1 high-level architecture

Progress so far

1. Completed the design and coding for the router, the network interface and the cache.
2. Thoroughly tested both the network interface and the router for functional correctness
3. Assembled four routers with network interfaces and caches

Router

The flow control for the router is store and forward, which means that the head flit waits at the router until the entire packet is received. To support the new communication scheme between the cache and router, more states are added in the router.

Network Interface

We implement the network interface to only store two pairs of head and tail flits, corresponding to “send” and “receive” process respectively. The “send” process refers to sending data out of cache from this core to another, while the “receive” process is the opposite. The head flit containing adequate information to transmit one packet is composed of the (1) source and destination coordinates within the network, (2) the number of flits in one packet, and (3) the memory address of the leading data flit. The composition of head flit is shown in Figure 2. Data transmitted is assumed to be adjacent to each other and follow sequence order in address.

Figure 2. head flit

Cache

We implement the L1 cache as an N-way write-back, write-allocate cache. The L2 cache is implemented like a memory, meaning that it is capable of storing all the data coming from other caches and never needs eviction.

Data Transmission Protocol

When a cache miss happens on an invalid cache line, the cache will send an request to the processor to ask for the data from other caches. Since we do not implement the cache coherence protocol, the destination is determined by our simulation inputs. Then, the cache is blocked and it waits for the requested data to come back. When a cache miss happens on a valid cache line, eviction of that cache line is needed first. Hence, the cache sends an eviction request and the corresponding eviction address to the network interface. Then, the network interface will construct the head flit based on the eviction address. Once the head flit is sent to the router successfully, the network interface will signal the cache to start sending data flit by flit to the router. After the router receives all the data, it signals the network interface to send the tail to it. When the router gets the whole packet, it start to send the packet to the destination.

To receive a packet, the router first sends the head flit to the network interface. The network interface then deconstructs the head flit to obtain the memory address and signals the cache to receive the data from the router. Once all the data is received by the cache, the router will send the tail to the network interface to signal the end of the receiving.

Next steps:

1. Work on developing the testbench for cache
2. Start testing on entire system once cache is verified
3. Do experiments and evaluation on the system

Issue

What kind of results are expected in final deliverable?