Project Title: Multicore Cache System

Implementing a simplified MOESI cache coherence mechanism to manage shared memory in a multicore system.

Technical:

Dual or four CPU cores (adjusted based on the gate usage) system, each core has an L1 cache and shares an L2 cache. All L1 caches and the L2 cache are connected to a common bus, using the MOESI protocol for cache coherence (potentially with fewer states if area is limited). Hardware will implement the L1 caches, the bus, and the L2 cache. Chip input ports will provide each L1 cache with core memory access instructions (idle/load/store). Caches will be very small to fit within the gate budget, with each cacheline containing the coherence state, tag and only one bit data to minimize gate count. The caches will be non-associative. If there is an L2 miss, it will pretend to have fetched the data from the main memory, and the data will be 0/1 in turn.

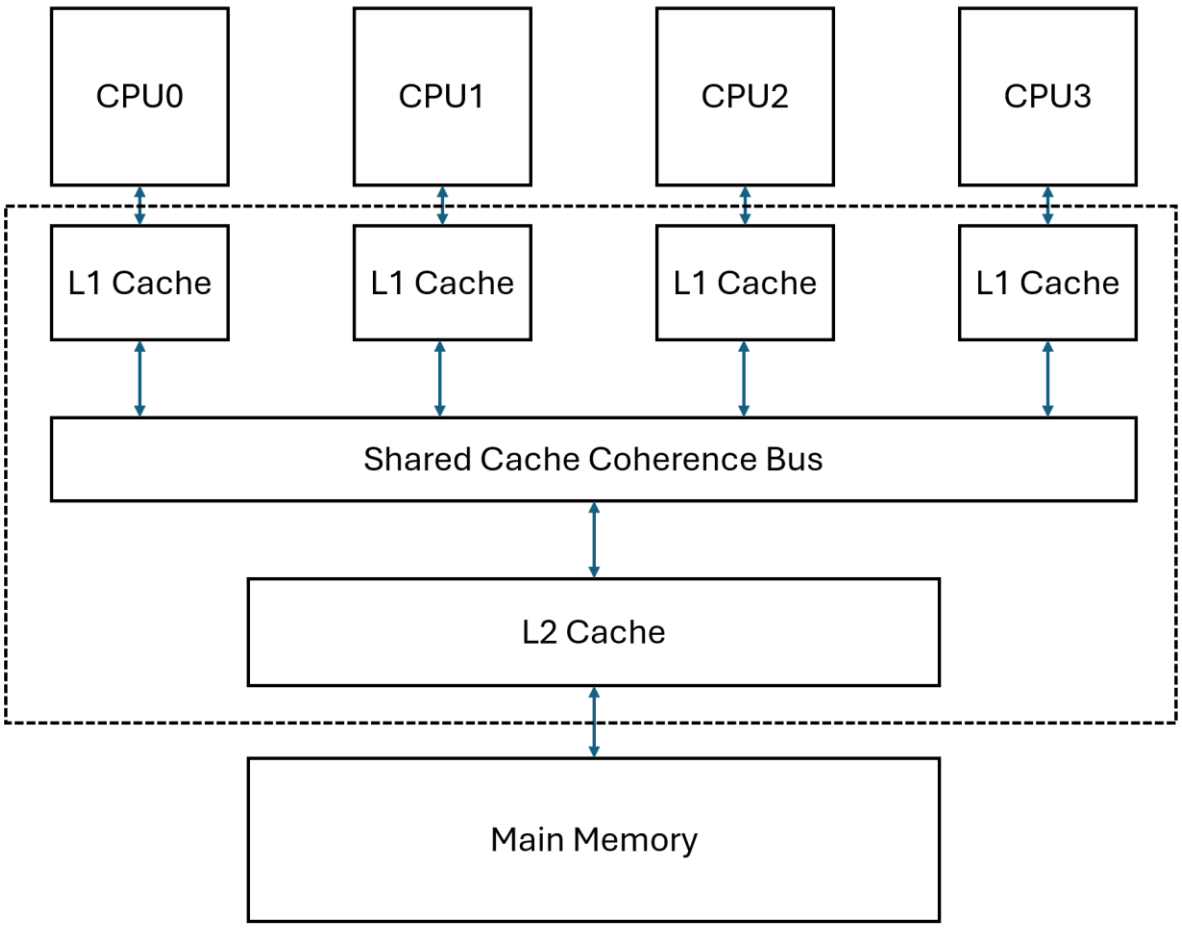
Parameters:

CPU counts	4
L1 cachelines	8
L2 cachelines	32
State bits	3 (MOESI)
Address bits	8

All of these parameters will be configurable, adjusted based on gate budget.

	Action and Next State						
Current State	*Processor Read*	*Processor Write*	*Eviction*		*Cache Read*	*Cache Read&M*	*Cache Upgrade*
I	*Cache Read* If no sharers: → E If sharers: → S	*Cache Read&M* → M			No Action → I	No Action → I	No Action → I
S	No Action → S	*Cache Upgrade* → M	No Action → I		Respond shared; → S	No Action → I	No Action → I
E	No Action → E	No Action → M	No Action → I		Respond shared; Supply data; → S	Respond shared; Supply data; → I	
O	No Action → O	*Cache Upgrade* → M	*Cache Write-back* → I		Respond shared; Supply data; → O	Respond shared; Supply data; → I	
M	No Action → M	No Action → M	*Cache Write-back* → I		Respond shared; Supply data; → O	Respond shared; Supply data; → I	

MOESI Protocol



Design Architecture

I/Os:

io_in[11:0]: For each L1 cache, providing memory access instructions (idle/load/store) to the L1 cache, requiring 4 clock cycles for complete input (one cycle for each core). So the cache system will run at a ¼ clock frequency.

io_in[0]: request_valid, io_in[1]: read/write, io_in[9:2]: request_address, io_in[10]: request data, io_in[11]: receive_ready.

io_out[2:0]: ready signals and data from L1 caches.

io_out[0]: request_ready, io_out[1]: return_data_valid, io_out[2]: return_data.

Hardware Peripherals:

None