Project Title: Multicore Cache System

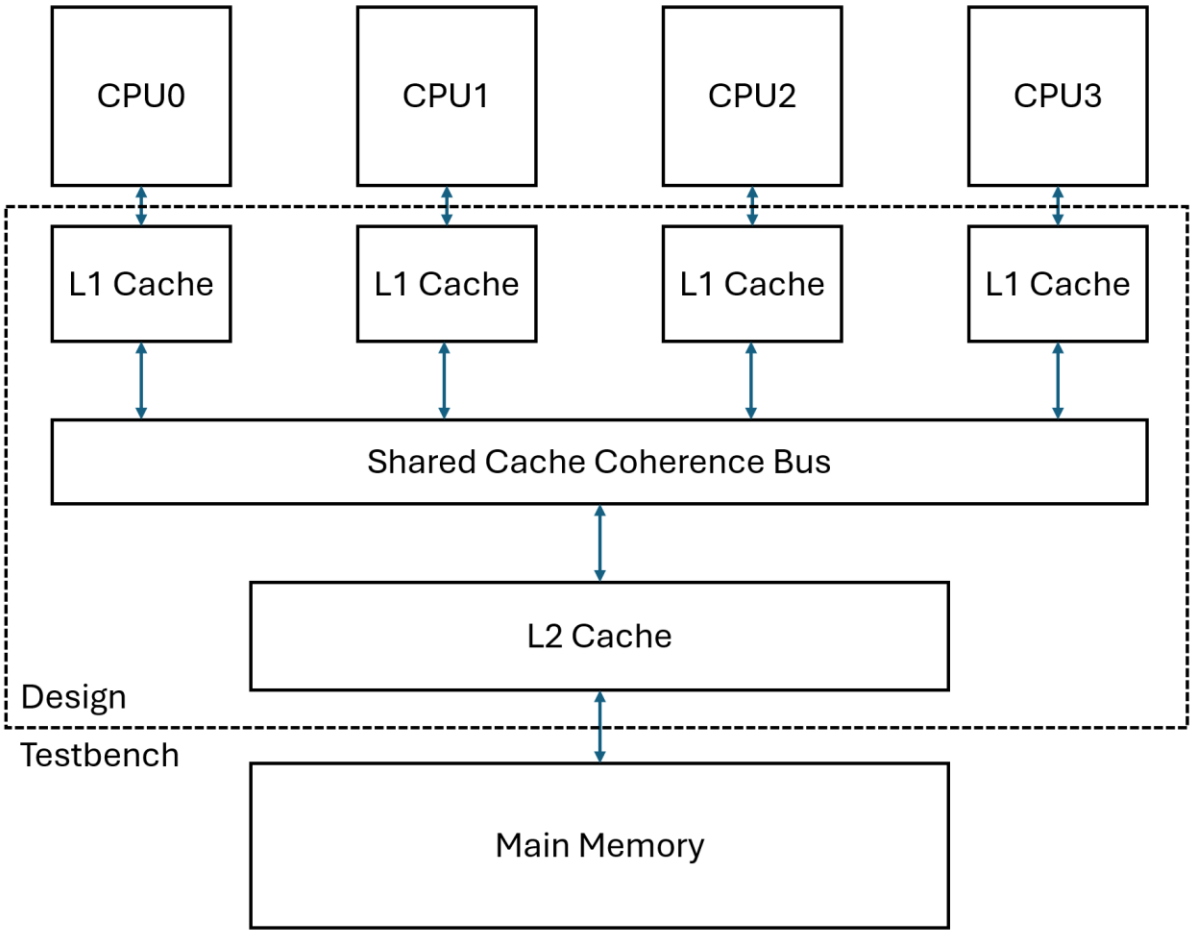
Implementing a MOESI cache coherence mechanism to manage shared memory in a multicore system.

Parameters:

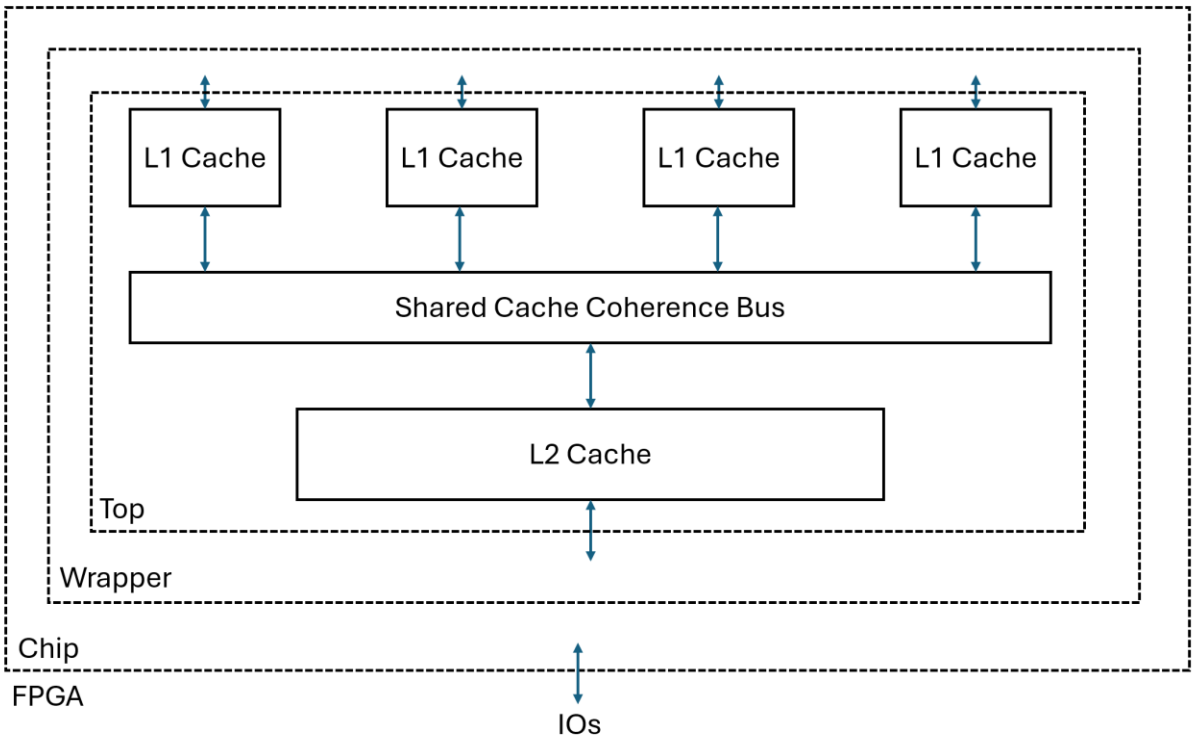
CPU counts	4
L1 cachelines	4
L2 cachelines	16
Coherence protocol	MOESI
Address bits	8
Offset bits	2
Cacheline bits	1

These parameters are designed to minimize area usage and the number of I/Os. All parameters are configurable in the *cache.svh* file.

Schematic:



Schematic for testbench



Schematic for tape-out

How to test:

Pre-silicon Test:

Run the following command:

```
vcs -sverilog ./src/design.v ./testbench/tb.sv ./testbench/tb_mm.sv
```

This test includes:

- A serialized test, where each CPU writes and reads to the same address in turn, verifying the correctness of each operation.
- A parallelized test, where all CPUs simultaneously write and read to randomly selected addresses in the cache system. Addresses are then exchanged among CPUs, continuing the write/read operations, checking all operations are correct and coherent.

Post-silicon Test:

Due to limited chip IOs, a wrapper module is used to serialize the input and output of the design. Each cycle of the design operation requires 6 clock cycles to complete:

- The first 4 cycles are used to accept input bits and feed them to the corresponding ports of the design.
- A rising edge is given to the design when transitioning to the fifth cycle.
- The last 2 cycles are used to serialize the output data and transmit it to an external FPGA.
- When transitioning from the last cycle to the first cycle of the next round, the FPGA determines the next round inputs and outputs them to the DUT.