Multi-Core Toy CPU Hardware Specification (Unified L1 Version)

1. Overview

- **Architecture:** 2-core, unified L1 per core, shared L2, 32-bit word size.
- **Pipeline:** 5 stages (IF, ID, EX, MEM, WB) with hazard detection and forwarding.
- **Caches:**
  - Each core has a **unified L1 cache** (instructions + data).
  - Shared **L2 cache** with round-robin arbiter.
- **ISA:** Minimal toy RISC-like ISA, 32-bit instructions.
- **Atomicity:** LL/SC with per-core reservations, enforced via MESI and invalidations.
- **Test Harness:** Two cores execute interleaved LL/SC increments/decrements of a shared counter.

2. System Architecture

- **Core count:** 2 (extendable).
- **Clock/reset:** Shared clock and async reset.
- **Program loading:**
  - Instructions and data preloaded into **L2 memory** from `program.hex`.
  - No separate instruction ROMs.
- **Core reset PCs:** Configurable per core (e.g., Core0 = `0x0000_0000`, Core1 = `0x0000_1000`).

3. Instruction Set Architecture (Toy ISA)

Opcode	Mnemonic	Semantics
`0000`	NOP	No operation
`0001`	LL rd, [addr]	Load-Linked (set reservation)
`0010`	SC rs, [addr]	Store-Conditional (commit if reservation valid)
`0011`	ADD rd, rs1, imm/rs2	Integer addition
`0100`	SUB rd, rs1, imm/rs2	Integer subtraction
`0101`	BRZ rs, offset	Branch if zero

Opcode	Mnemonic	Semantics
`0110`	JMP target	Unconditional jump
`1111`	HALT	Halt on retire

- **Registers:** 32 × 32-bit.
- **Encoding:** `[31:28]=opcode, [27:24]=rd, [23:20]=rs1, [19:16]=rs2, [15:0]=imm`.

---

4. Pipeline & Hazards

- **5 stages:** IF, ID, EX, MEM, WB.
- **Hazard resolution:**
    - Forwarding from EX/MEM/WB.
    - 1-cycle stall for load-use hazards.
    - Branch flush on taken branch.
- **HALT retires in WB stage** (realistic stop).

---

5. Cache Hierarchy

Unified L1 Cache (per core)

- Size: parametric.
- Ports: **IF and DATA**.
- Arbitration: **priority DATA > IF** (to reduce load stalls).
- MESI state per line.
- LL/SC reservation bits per line.
- IF fetches are always read-only (no reservations).

L2 Cache (shared)

- Central backing store (array).
- One request per cycle.
- **Two-master round-robin arbiter** selects between cores’ L1 requests.

---

6. Coherence Protocol (MESI)

- Instruction fetches → `S` or `E` states only.
- Data writes may trigger invalidation snoops.

- Reservation invalidated when snoop occurs or line evicted.

---

7. Atomic Operations

- **LL:** Read from memory, set reservation on cache line.
- **SC:** Write back only if reservation still valid.
  - Success → commit write, clear reservation.
  - Failure → return `sc_success=0`, no write.

---

8. Test & Verification Infrastructure

Testbench (`multicore_full_sim.sv`)

- Two cores, unified L1s, shared L2.
- `program.hex` loaded into L2.
- Self-check at HALT: final shared counter = 0.

Watchdog

- Terminates after `MAX_CYCLES` if cores don’t halt.
- Dumps:
  - PC of both cores
  - Last 5 retired instructions per core
  - SC success/fail counts
  - Shared memory dump around counter

Metrics

- SC success/fail counts per core always reported.
- Retire trace buffer for debugging deadlocks.

---

9. Debug Features

- **SC success/fail counters.**
- **Circular retire trace (last 5 instructions).**
- **Watchdog memory dump.**
- **Assertions:**

    - No LL/SC on IF fetches.
    - Counter bounded sanity checks.

---

10. Future Roadmap

- Exceptions & interrupts.
- Multiply/divide instructions.
- Associative L1, write-back buffers.
- Branch prediction.
- MMU / paging.
- Extend arbiter to N-core.
- FPGA prototyping.