Open2C: Open-source Generator for Coherent Cache Memory Subsystem

Anastasiia Butko, David Donofrio, John Leidel

Computer Architecture Group

Lawrence Berkeley National Lab

December, 2021

Design Automation Conference (DAC), San Francisco, CA

BERKELEY LAB

Outline

- Motivation
- Open2C Generator
- Validation and Verification
- OpenSoC System Architect Integration

OPEN CACHE COHERENCY Open2C

Motivation

Why?

Motivation

Cache Memory Hierarchy plays an important role in complex computational systems

- Maintains data movements hiding the memory latency to improve computational performance
- Mostly managed by hardware that facilitates programming

Motivation

Cache Memory Hierarchy plays an important role in complex computational systems

- Maintains data movements hiding the memory latency to improve computational performance
- Mostly managed by hardware that facilitates programming

- Multi-processor scalability is a big concern

Motivation

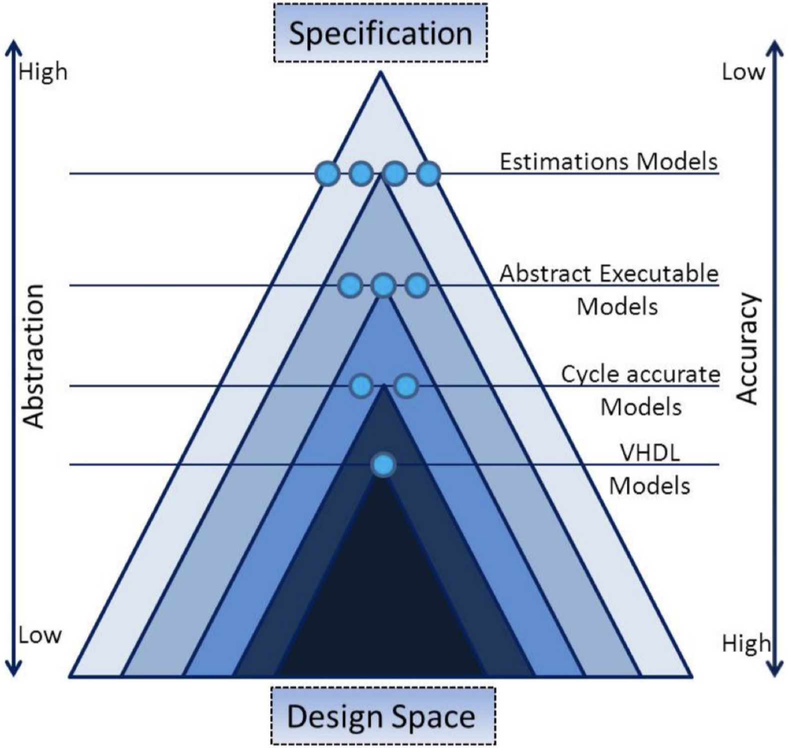
Cache Memory Hierarchy plays an important role in complex computational systems

- Maintains data movements hiding the memory latency to improve computational performance
- Mostly managed by hardware that facilitates programming

- Multi-processor scalability is a big concern
- Upcoming highly specialized architectures require memory subsystem to be less generic
  - Requires a ‘deep dive’ to get improved
  - Hard to **evaluate**

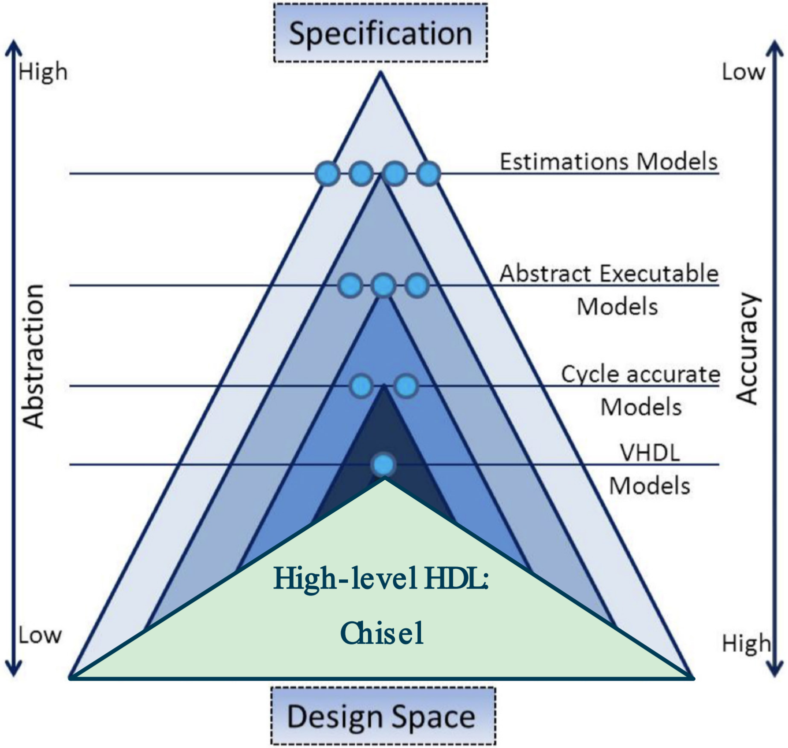
Evaluation issue

Pyramid of Abstraction levels



Evaluation solution

Pyramid of Abstraction levels

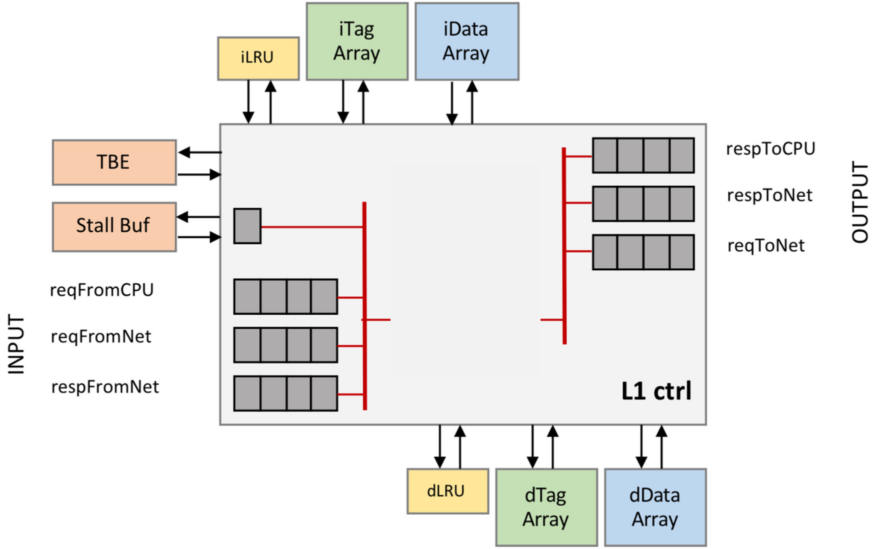


Open2C

How is it implemented?

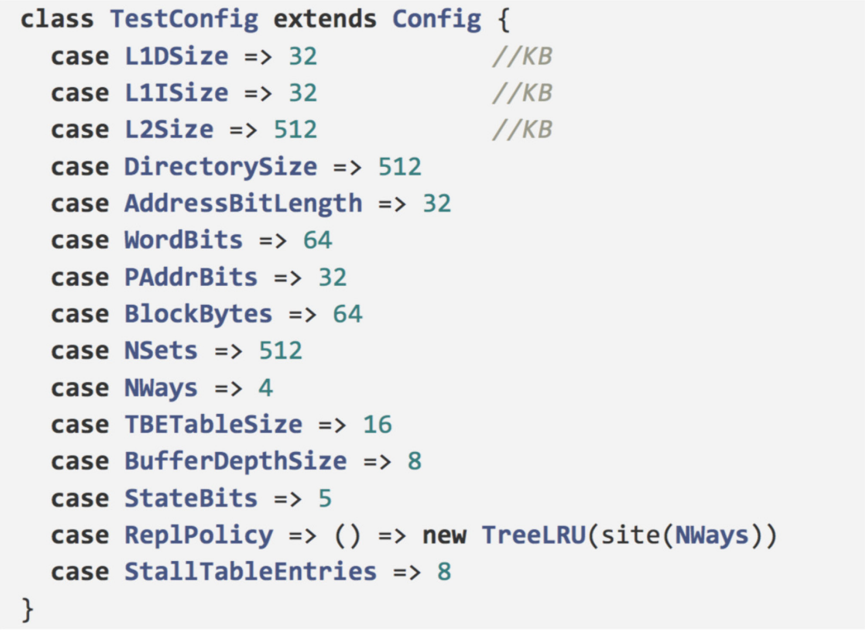
Open2C: Structure and Components

- Parameterizable hardware resource library (protocol independent)
  - Data arrays
  - Tag array
  - Miss Status Handling register
  - Stall table
  - Replacement policy unit
  - Message buffer



Open2C: Structure and Components

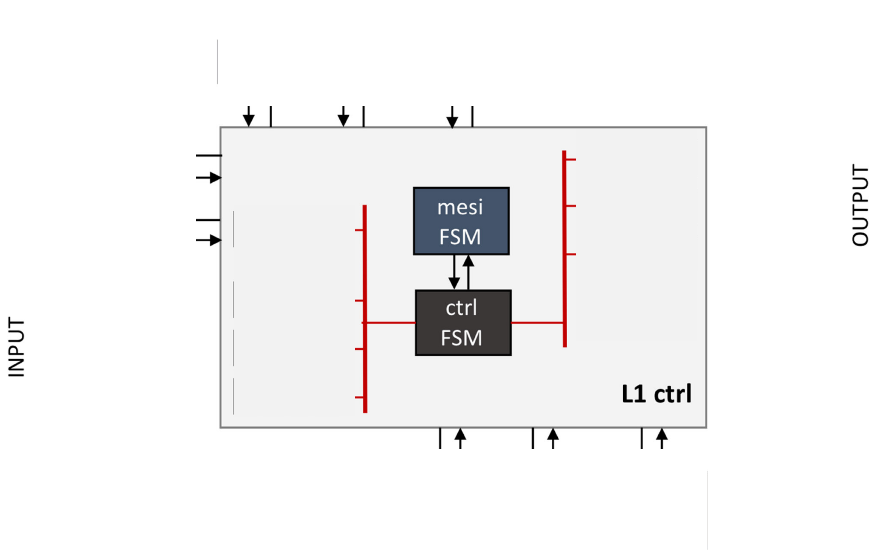
- Parameterizable hardware resource library (protocol independent)
  - Data arrays
  - Tag array
  - Miss Status Handling register
  - Stall table
  - Replacement policy unit
  - Message buffer



```
class TestConfig extends Config {
  case L1DSize => 32            //KB
  case L1ISize => 32            //KB
  case L2Size => 512            //KB
  case DirectorySize => 512
  case AddressBitLength => 32
  case WordBits => 64
  case PAddrBits => 32
  case BlockBytes => 64
  case NSets => 512
  case NWays => 4
  case TBETableSize => 16
  case BufferDepthSize => 8
  case StateBits => 5
  case ReplPolicy => () => new TreeLRU(site(NWays))
  case StallTableEntries => 8
}
```

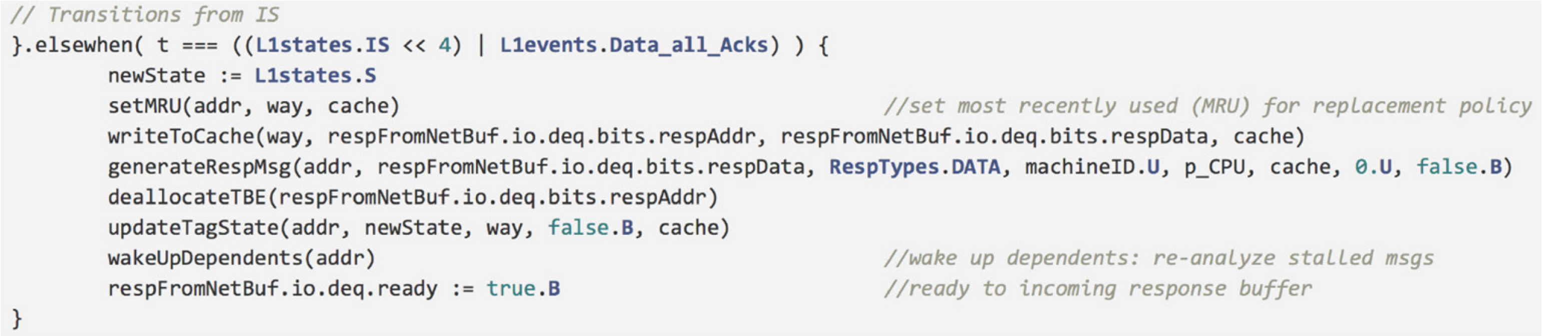
Open2C: Structure and Components

- Protocol-dependent coherence controllers
  - Protocol FSM
  - Controller resources FSM
  - Request and Response Messages



Open2C: Structure and Components

- Hardware resources access interface
  - Per each case Controller performs a specific set of actions
  - HR Library and access functions provide an efficient interface to combine multiple actions



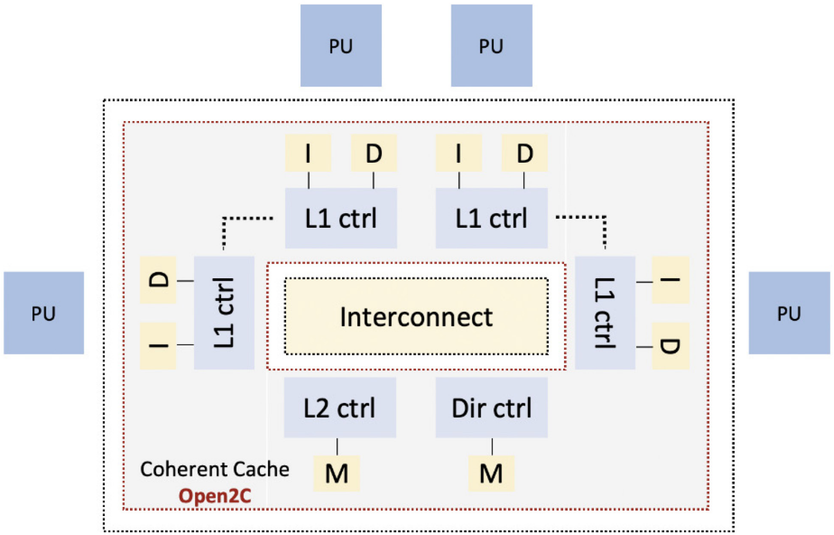
```
// Transitions from IS
}.elsewhen( t === ((L1states.IS << 4) | L1events.Data_all_Acks) ) {
        newState := L1states.S
        setMRU(addr, way, cache)                                       //set most recently used (MRU) for replacement policy
        writeToCache(way, respFromNetBuf.io.deq.bits.respAddr, respFromNetBuf.io.deq.bits.respData, cache)
        generateRespMsg(addr, respFromNetBuf.io.deq.bits.respData, RespTypes.DATA, machineID.U, p_CPU, cache, 0.U, false.B)
        deallocateTBE(respFromNetBuf.io.deq.bits.respAddr)
        updateTagState(addr, newState, way, false.B, cache)
        wakeUpDependents(addr)                                         //wake up dependents: re-analyze stalled msgs
        respFromNetBuf.io.deq.ready := true.B                          //ready to incoming response buffer
}
```

Open2C

System Level Verification

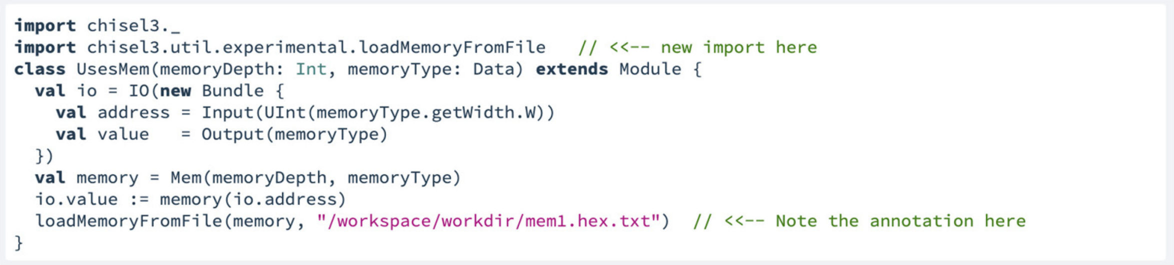
Open2C: System Level Integration

- Open2C hardware resources connected with other components
  - Processing Units (PUs) or other traffic generators
  - Interconnect
  - Memories



Memories

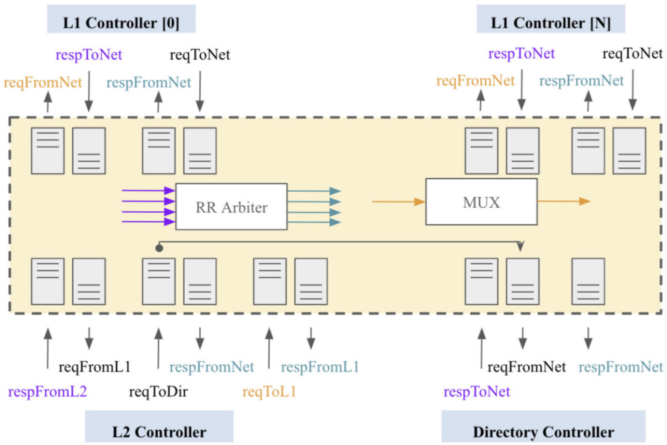
- Chisel provides facilities for creating memories:
  - Read-Only Memories (ROM)
  - Read-Write Memories
    - `SyncReadMem`: sequential/synchronous-read, sequential/synchronous-write memories
    - `Mem`: combinational/asynchronous-read, sequential/synchronous-write
    - Memory Initialization: from an external `binary` or `hex`
      - `loadMemoryFromFile`



```
import chisel3._
import chisel3.util.experimental.loadMemoryFromFile   // <<-- new import here
class UsesMem(memoryDepth: Int, memoryType: Data) extends Module {
  val io = IO(new Bundle {
    val address = Input(UInt(memoryType.getWidth.W))
    val value   = Output(memoryType)
  })
  val memory = Mem(memoryDepth, memoryType)
  io.value := memory(io.address)
  loadMemoryFromFile(memory, "/workspace/workdir/mem1.hex.txt")  // <<-- Note the annotation here
}
```

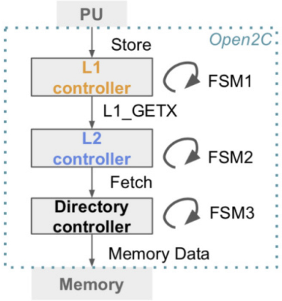
Interconnect

- Full System-on-Chip experiments
  - Realistic interconnect, e.g. OpenSoC Fabric
  - Dynamic behaviour, performance, coherency traffic, etc.
- **Verification purpose (Open2C)**
  - “Perfect” interconnect
  - Most of the packet transfers are done in 1-2 cycles



Processing Units or Traffic Generators

- Full System-on-Chip experiments
  - Realistic PUs, e.g. RISC-V cores
  - Execute real applications
- **Verification/exploration purpose (Open2C)**
  - Traffic Generators, e.g. **trace generators**, synthetic traffic generators, etc.
  - Varying traffic intensity, application pattern emulation, quick test and debug



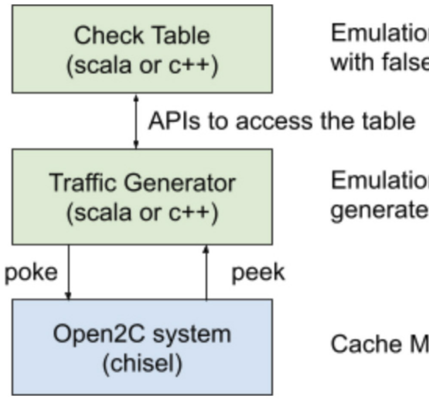
Open2C setup

Cycle	Module	Event	Transaction	Address
5	L1Cache	Store	NP>IM	[0x3fc0, line 0x3fc0]
348	L2Cache	L1_GETX	NP>IM	[0x3fc0, line 0x3fc0]
364	Directory	Fetch	I>IM	[0x3fc0, line 0x3fc0]
510	Directory	Memory_Data	IM>M	[0x3fc0, line 0x3fc0]
658	L2Cache	Mem_Data	IM>MT_MB	[0x3fc0, line 0x3fc0]
781	L1Cache	Data_all_Acks	IM>M	[0x3fc0, line 0x3fc0]
781	L1Cache	L1_Replacement	M>M_I	[0x3fc0, line 0x3fc0]
800	L2Cache	Exclusive_Unblock	MT_MB>MT	[0x3fc0, line 0x3fc0]
...				

gem5 reference system trace

Synthetic Traffic Tester

Tester Setup



Check Table (scala or c++)

APIs to access the table

Traffic Generator (scala or c++)

poke

peek

Open2C system (chisel)

Emulation of a data map with false sharing regions

Emulation of a core that generates Read/Write traffic

Cache Memory sub-system

Synthetic Traffic Parameters

```
val table_size = 10
val sim_cycle = 100
val total_req = 10
val traff_int = 5
val resp_int = 2
val deadlock_tracker = 20
```

Check Table Status after Simulation

```
The current status of the Check Table:
[info][37.111] Address: 1558070700, Pending: false, State: 1, Cycle: 18
[info][37.111] Address: 1270054682, Pending: false, State: 1, Cycle: 24
[info][37.112] There is a deadlock detected when sending message: 1
[info][37.112] Address: 2000587193, Pending: true, State: 1, Cycle: 12
[info][37.112] Address: 153470114, Pending: false, State: 1, Cycle: 37
[info][37.112] There is a deadlock detected when sending message: 3
[info][37.112] Address: 186176867, Pending: false, State: 1, Cycle: 43
[info][37.113] There is a deadlock detected when sending message: 4
[info][37.113] Address: 1781887705, Pending: false, State: 1, Cycle: 49
[info][37.113] There is a deadlock detected when sending message: 5
[info][37.113] Address: 650407566, Pending: true, State: 1, Cycle: 36
[info][37.113] Address: 1428028968, Pending: false, State: 1, Cycle: 61
[info][37.113] There is a deadlock detected when sending message: 7
[info][37.114] Address: 374187158, Pending: true, State: 1, Cycle: 48
[info][37.114] Address: 1947051239, Pending: true, State: 1, Cycle: 54
test CCSoC1C Success: 0 tests passed in 106 cycles taking 38.605787 seconds
```

OpenSoC
System Architect
Integration

Plugin

OpenSoC System Architect

- Open2C is configured through the `Features` nodes in the CoreGen IR
- To connect Open2C to your design define a `Cache` node and include the `Override` keyword

Configuration Parameters

```
Plugins:
    - Plugin: Open2CPlugin
      PluginName: Open2CPlugin_SysArch
      MajorVersion: 1
      MinorVersion: 0
      PatchVersion: 0
      Features:
        - FeatureName: L1DSize
          FeatureValue: 16384
          FeatureType: Unsigned
        - FeatureName: L1ISize
          FeatureValue: 8192
          FeatureType: Unsigned
        - FeatureName: L2Size
          FeatureValue: 32768
          FeatureType: Unsigned
        - FeatureName: DirectorySize
          FeatureValue: 512
          FeatureType: Unsigned
```

Connecting Open2C modules

```
# -----------------------------------------------------------
# Cache Section
# -----------------------------------------------------------
Caches:
  - Cache: Core0.L1.cache
    Sets: 512
    Ways: 4
    Override: Open2CPlugin

# -----------------------------------------------------------
# Core Section
# -----------------------------------------------------------
Cores:
  - Core: core0
    ThreadUnits: 1
    Cache: Core0.L1.cache
    ISA: BasicRISC.ISA
    RegisterClasses:
      - RegClass: GPR
      - RegClass: CTRL
```

Thank you for your attention.
Questions?

BERKELEY LAB

Case Study: resource utilization

MESI protocol implementation

- Single-chip 2-level cache system with strictly inclusive hierarchy
- Level 1 caches are private
- Level 2 cache is shared among all the cores

Case Study: resource utilization

MESI protocol implementation

- Single-chip 2-level cache system with strictly inclusive hierarchy
- Level 1 caches are private
- Level 2 cache is shared among all the cores

Experimental Setup

- Chisel hardware compilation: Verilog RTL implementation
- Yosys Open SYnthesis Suite: 14nm technology synthesis
  - area, number of cells, number of wires, etc.

BERKELEY LAB

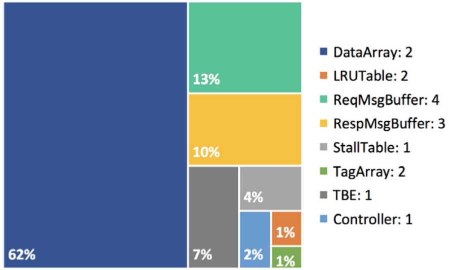
Case Study: resource utilization

MESI protocol implementation

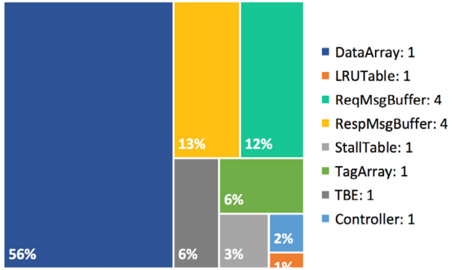
- Single-chip 2-level cache system with strictly inclusive hierarchy
- Level 1 caches are private
- Level 2 cache is shared among all the cores

Experimental Setup

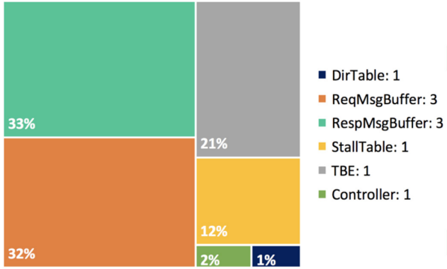
- Chisel hardware compilation: Verilog RTL implementation
- Yosys Open SYnthesis Suite: 14nm technology synthesis
  - area, number of cells, number of wires, etc.



L1 ctrl



L2 ctrl



Directory ctrl