

Georgia Institute of Technology

CS4290/CS6290/ECE4100/ECE6100

- Do we have to worry about memory disambiguation problem in this assignment?

No, We assume that there is a perfect memory disambiguation predictor.

- What should I use to find out memory instructions? Is checking mem_type sufficient or do I have to check opcode?

if opcode is OP_ST mem_type is MEM_ST and if opcode is OP_LD mem_type is MEM_LD. so just checking mem_type itself is sufficient.

- What will be the cache miss penalty? Is this KNOB_DCACHE_HIT_LATENCY+KNOB_MEM_LATENCY_ROW_HIT or just KNOB_MEM_LATENCY_ROW_HIT?

It is KNOB_DCACHE_HIT_LATENCY+ KNOB_MEM_LATENCY_ROW_HIT (or KNOB_MEM_LATENCY_ROW_MISS) + additional queuing delay.

- Do we need to implement store-load forwarding?

Yes, it will be done though MSHR. Be careful of memory sizes. e.g.) store writes location 0x001 with the memory size 1B, then load address 0x003 cannot get data from the store because both two addresses really do not overlap.

- I see op->mem_read_size and op->mem_write_size , what are for?

You need those information to check store-load forwarding.

- What if load/store addresses are mapped into two different cache blocks (unaligned accesses)?

Let's just access the first cache block

- Do we need to translate virtual addresses to physical addresses?

No, we just assume that the addresses that are provided in Pin are physical addresses.

- Is the DCACHE pipelined ? No, we assume that the DCACHE is not pipelined. So if one instruction is accessing the D-cache, the MEM_stage is busy.
- When an instruction can retire? Can it retire out of order?

Yes, while cache miss load instructions are waiting for DRAM requests, older

instructions that are not dependent on the load instruction can be executed and can be retired.

- Do non load/store instructions go through the MEM stage?

Yes.

- While the MEM stage is busy for dcache accesses do non load/store instructions still wait in the EXE stage?

Yes. Although a modern processor would be optimized for this case, for our simulator, we simply stall the pipeline if the MEM stage is busy for dcache accesses. However, after `KNOB_DCACHE_HIT_LATENCY`, regardless of whether a load/store instruction hits cache, the pipeline is free. The instruction is moved to the MSHR if there is a cache miss.

- A load instruction is in the MSHR. A new store instruction has the same memory address as the load instruction. Does the store instruction need to go to the MSHR?

In principle, yes, the store instruction (actually the store data) also needs to be stored in the MSHR. However, since we do not actually simulate memory contents, the simulator can just simply send the store op into the WB stage after `KNOB_DCACHE_HIT_LATENCY`.

- Now multiple ops can be sent to the WB stage when memory requests are merged and later the memory request is serviced. However, the simulator frame can store only one op into the MEM stage latch. Can I (or should I) modify the MEM stage latch?

Yes, you should modify the MEM stage latch to store multiple ops. For `print_pipeline()`, you should also change it to print out multiple ops. Since we might look at `print_pipeline` for partial grading, please print out the ops as following.

```
FE: ##### ID: ##### EX: ##### MEM: 1 (1 op is sent to the MEM stage latch)
FE: ##### ID: ##### EX: ##### MEM: 2 3 4 // (3 ops are sent to the MEM Stage latch)
FE: ##### ID: ##### EX: ##### MEM: 2 3 5 6 7 8 // (6 ops are sent to the MEM stage latch)
```

- How to set up `m_state` in `mem_req_s` class ?

- `insert_mshr --> m_state = MEM_NEW;`
- `send_bus_in_queue --> m_state = MEM_DRAM_IN;`
- `dram_schedule`, after DRAM is scheduled `m_state = MEM_DRAM_SCH;`
- `send_bus_out_queue --> m_state MEM_DRAM_OUT;`

- How to set up m_tye in mem_req_s class?

- DRT_DFETCH // dcache load miss
 - DRT_DSTORE // dcache store miss
- You can ignore other types for now.

- Do I still need to maintain data hazard and control hazard correctly?

No, we will not grade data hazard and control hazard any more.

- A memory instruction generates a cache miss but because of MSHR full, the instruction cannot be stored inside the MSHR. How many times do we have to increment the cache miss counter?

Only once at the first time access.