Extension: Two-Level Cache

Specifications

Extension Description (1/2)

- Main Goal
  - Add 2 separate L2 caches for I-cache and D-cache
  - Unified L2 cache (1 L2 cache for both L1 I-cache & L1 D-cache) will be considered as better design (but not required)
- Total size of L1 caches is suggested to be 32 words
  - 8 blocks & each 4 words
- Total size of L2 caches suggested to be 256 words
  - Block size and number of blocks are up to your considerations

Extension Description (2/2)

- System Clock Speed
  - L1 Caches, L2 Caches, Slow Memory : Same clock speed as MIPS
- I/O Specification
  - Refer to the same top module as baseline: add ‘module L2_Cache’
- “UnifiedL2 cache” is better than “Separate L2 caches”

Test Program Generation

- Based on condition “hasHazard”, long version.
  - +1+2+3 sequence generate while Fibonacci
- In file “generate”:
  - L2Cache_generate.py/ipynb(jupyter)
    - Python (version = 3.x)
    - argv[1] = nb_Fibonacci
    - I_mem_L2Cache_ref & TestBed_L2Cache_ref should be placed in the same folder
    - **I_mem_L2Cache** & **TestBed_L2Cache** will be generated
      - Provieded files nb_Fibonacci = 20
- **+define+L2Cache** in ncverilog simulation command

Comparison Metrics

- Base on the test program: "I_mem_L2Cache"
- Score 1 (L2C_S1): Avg. memory access time (ns)
  - L2C_S1 = HT1 + MR1*(HT2+MR2*MP2)
  - HT: hit time
  - MR: miss rate
  - MP: miss penalty
  - ~1: of level 1 cache
  - ~2: of level 2 cache
- Score 2 (L2C_S2): Total execution time (ns)
  - L2C_S2 = total execution time of the test program

Some Possible Discussion

- Design methodology for good score (before/after)
- The size ratio of L2 Cache to L1 Cache
- Sequence length for demonstrating the effectiveness of L2 Cache