

# Computer Design & Organization

## Solution for assignment 1

1. Set sim-outorder configuration parameters to reflect a computer that has the following configuration:
  - a pipeline that fetches, decodes, issues, executes and commits one instruction/cycle, no matter what the instruction type
    - # instruction fetch queue size (in insts) -**fetch:ifqsize 1**
    - # instruction decode B/W (insts/cycle) -**decode:width 1**
    - # instruction issue B/W (insts/cycle) -**issue:width 1**
    - # run pipeline with in-order issue -**issue:inorder true**
  - only one of each type of functional unit
    - # total number of integer ALU's available -**res:ialu 1**
    - # total number of integer multiplier/dividers available -**res:imult 1**
    - # total number of memory system ports available (to CPU) -**res:mempport 1 or 2**  
**1 memory port is correct, but 2 memory ports would allow misses in the instruction and data caches to proceed in parallel for a higher-performance implementation.**
    - # total number of floating point ALU's available -**res:fpalu 1**
    - # total number of floating point multiplier/dividers available -**res:fpmult 1**
  - an 8KB, two-way set-associative L1 instruction and data caches with 32 byte blocks
    - # I1 inst cache config -**cache:il1 il1:128:32:2:1**
    - # I1 data cache config -**cache:dl1 dl1:128:32:2:1**
  - a 256KB, direct-mapped L2 unified cache with 32 byte blocks
    - # I2 data cache config -**cache:dl2 ul2:8192:32:1:1**
    - # I2 instruction cache config -**cache:il2 dl2**
  - an 8-way, 128-entry data TLB
    - # data TLB config -**tlb:dtlb dtlb:16:4096:8:1**
  - a 4-way, 64-entry instruction TLB
    - # instruction TLB config -**tlb:itlb itlb:16:4096:4:1**
  - All caches and the TLB have an LRU block replacement policy.  
**see above**
  - The page size is 4KB.  
**see above**
  - This means that the rest of the parameters are either left with their default values or must be set to: `fetch:speed <1>` and `issue:inorder`.

1. Report the values for the following metrics:

- total number of instructions committed -- this is all the instructions that have completed all the phases of instruction execution, including writing their results to the register file.

**8185361**

- the total number of block replacements for the L1 data cache

**65027**

- the hit ratio for the L2 cache

**total number of hits/total number of accesses = .97**

**These were the metric values on Douglas's simulation run. Since they may vary slightly, depending on what data inputs you used, he graded according to the run output that you handed in.**

1. Answer the following questions:

- What is the rationale for the value for the number of cycles per instruction?

**CPI is higher than one because of cache misses (and hits if cache accesses take more than one cycle), mispredicted branches and data hazards. Structural hazards may also occur. CPI would otherwise be one if there were no processor stalls, as with a perfect pipeline.**

- Why is there a difference between the number of blocks replaced in the L1 data cache and the number of blocks written back from the L1 data cache?

**In a write-back cache, only the dirty blocks are written back to the next level of the cache hierarchy. Therefore there can be more block replacements than write-backs.**

- Justify the value of the total number of write-backs for the L1 instruction cache.

**Code is usually not written to, so there should be no writes to the instruction cache, and with no dirty blocks, no write-backs are possible. (This fact can be exploited to make the instruction cache circuits simpler – the downside is that so called self-modifying code has to bypass the read-only instruction cache and this slows down the processor).**