Programming Assignment 3: Load Bypassing

- Shantanu Mukhopadhyay [200541793]

To implement the load bypassing mechanism in the gpgpu-sim for all load instructions in the range of 0xc0000000-0xc00fffff bypass the L1 data cache:

Changes made in src/gpgpu-sim/ shader.h

- Added the counter variable under **struct shader_core_stats_pod** so that it can be accessed in shader.cc and incremented using m_stats pointer.

root > gpgpu-sim_distribution > src > gpgpu-sim > C shader.h

```
1604    struct shader_core_stats_pod {
1680      int gpgpu_n_mem_read_inst;
1681
1682      //assignment 3: Address based cache load bypass
1683      int bypass_loads_count; // Number of loads bypassed
1684
1685      int gpgpu_n_mem_l2_writeback;
1686      int gpgpu_n_mem_l1_write_allocate;
1687      int gpgpu_n_mem_l2_write_allocate;
1688
```

Changes made in src/gpgpu-sim/ shader.cc

Under **bool ldst_unit::memory_cycle(warp_inst_t &inst, mem_stage_stall_type &stall_reason, mem_stage_access_type &access_type)**

```
2031  bool ldst_unit::memory_cycle(warp_inst_t &inst,
2032                               mem_stage_stall_type &stall_reason,
2033                               mem_stage_access_type &access_type) {
2034    if (inst.empty() || ((inst.space.get_type() != global_space) &&
2035                         (inst.space.get_type() != local_space) &&
2036                         (inst.space.get_type() != param_space_local)))
2037      return true;
2038    if (inst.active_count() == 0) return true;
2039    if (inst.accessq_empty()) return true;
2040
2041    mem_stage_stall_type stall_cond = NO_RC_FAIL;
2042    const mem_access_t &access = inst.accessq_back();
2043
2044    bool bypassL1D = false;
2045    if (CACHE_GLOBAL == inst.cache_op || (m_L1D == NULL)) {
2046      bypassL1D = true;
2047    } else if (inst.space.is_global()) {  // global memory access
2048
2049      //assignment 3: cache L1D bypass for address range: 0xc0000000-0xc00fffff
2050      if(access.get_addr() >= 0xc0000000 && access.get_addr() <= 0xc00fffff) {
2051        // Bypass L1D cache for this address range
2052        bypassL1D = true;
2053        m_stats->bypass_loads_count++;
2054      }
2055      // skip L1 cache if the option is enabled
2056      if (m_core->get_config()->gmem_skip_L1D && (CACHE_L1 != inst.cache_op))
2057        {bypassL1D = true;}
```

The address range is set under the inst.space.is_global as the address bypass is under global memory access and the address is set using access.get_addr() and set the bypassL1D variable as *true*. Also incremented the bypassed loads counter under this condition.

Also under **void ldst_unit::cycle(),** have included the bypass address condition for memory fetches.

```
2597        if (CACHE_GLOBAL == mf->get_inst().cache_op || (m_L1D == NULL)) {
2598          bypassL1D = true;
2599        } else if (mf->get_access_type() == GLOBAL_ACC_R ||
2600                   mf->get_access_type() ==
2601                       GLOBAL_ACC_W) {  // global memory access
2602          // Assignment 3: Bypass L1D cache for address range: 0xc0000000-0xc00fffff
2603          if(mf->get_addr() >= 0xc0000000 && mf->get_addr() <= 0xc00fffff) {
2604            // Bypass L1D cache for this address range
2605            bypassL1D = true;
2606          }
2607          if (m_core->get_config()->gmem_skip_L1D) bypassL1D = true;
2608        }
2609        if (bypassL1D) {
```

And printed the per kernel counter for the bypassed loads under void **shader_core_stats::print(FILE *fout) const**

```
601   //assignment 3:
602   fprintf(fout, "bypassed load instructions = %d\n",bypass_loads_count);
603
```

Results of the counters before and after bypass of the load instructions for BFS Benchmark with SM7 TITANV configurations:

SM7_TITANV - BFS Benchmark			
Kernel	L1D_total_cache_accesses(original)	L1D_total_cache_accesses(bypass)	bypassed load instructions
1	2116	2090	26
2	4596	4433	163
3	9307	8344	963
4	27763	21879	5884
5	120908	86800	34108
6	508961	353055	181692
7	1230577	875477	609201
8	1478462	1060607	671956
9	1488124	1067834	674391
10	1490210	1069905	674406