ECE 786: Programming Assignment 2
Report

The changes are made to the following files in the simulator:

1) Abstract_hardware_model.h
2) Abstract_hardware_model.cc
3) Gpu-sim.h
4) Gpu-sim.cc
5) Shader.h
6) Shader.cc
7) Gpu-cache.h
8) Gpu-cache.cc

(1) abstract_hardware_model.h

- **Line 308**: added the counters to keep the count of all the branches encountered and the number of diverged branches(unsigned &all_branches and unsigned &diverged_branches) to argument list in the function declaration.
  The change is shown below:
  - void update(simt_mask_t &thread_done, addr_vector_t &next_pc, address_type recvg_pc, op_type next_inst_op,unsigned next_inst_size, address_type next_inst_pc, unsigned &all_branches, unsigned &diverged_branches);

- **Line 1011 and Line 1012**: added the counters all_branches and diverged_branches to class core_t This is the base class from which shader_core_ctx inherits. These counters will be used to keep the track of all the branch instructions encountered and the number of branch instructions which diverged control.
  The changes are shown below:
  - unsigned all_branches; // To keep track of all the branches//
  - unsigned diverged_branches; // To keep track of the branches which diverged control//
- **Line 1041 and Line 1042:** initialized the counters all_branches and diverged_branches to 0.
  The changes are shown below:
  - all_branches =0;
  - diverged_branches =0;

(2) abstract_hardware_model.cc

- **Line 650:** added the counters &all_branches and &diverged_branches to keep count of all the branch instructions encountered and the number of branch instructions that diverged control. These counters were added to the function simt_stack:: update().
  The original and the changed snippet in the code is shown below:

- Original: void simt_stack::update(simt_mask_t &thread_done, addr_vector_t &next_pc, address_type recvg_pc, op_type next_inst_op,unsigned next_inst_size, address_type next_inst_pc)
- Changed: void simt_stack::update(simt_mask_t &thread_done, addr_vector_t &next_pc, address_type recvg_pc, op_type next_inst_op,unsigned next_inst_size, address_type next_inst_pc,unsigned &all_branches, unsigned &diverged_branches).

- **Line 785 – Line 795**: added an 'if statement' in the simt_stack::update(). This if statement checks if the next instruction encountered is a branch instruction or not. If it is , then the counter to keep track of all the branches is incremented. Inside this 'if statement' another if statement is nested to check if each warp has diverged or not. If , the warp has diverged , the counter to keep track of the diverged branches is incremented. Thus with this code snippet added, we can keep track of the branches encountered and out of those branches the ones that diverged control.
  The change is shown below:
  - if (next_inst_op == BRANCH_OP){
    all_branches++;
    if(warp_diverged){
    diverged_branches++;
    }
    }
- **Line 835:** Passed the newly created counters to the function simt_stack::update()
  The original and the updated code is shown below:
  - Original: m_simt_stack[warpId]->update(thread_done, next_pc,inst->reconvergence_pc, inst->op,inst->isize,inst->pc);
  - Changed: m_simt_stack[warpId]->update(thread_done,next_pc,inst->reconvergence_pc, inst->op,inst->isize,inst->pc, all_branches, diverged_branches);

(3) gpu-sim.h

- **Line 422 and 441:** The scope of the class simt_core_cluster**m_cluster was changed from private to public.
  The change is shown below:
  - class simt_core_cluster **m_cluster;

(4) gpu-sim.cc

- **Line 898 - Line 909:** Declared two counters total_branches and total_divered_branches to keep track of the all branches and the branches that diverged control. The code snippet is added to compute the number of branches and diverged branches.
  The change is shown below:
  - unsigned total_branches = 0;
  - unsigned total_diverged_branches = 0;
  - for (unsigned i=0; i< m_shader_config->n_simt_clusters; i++)
    {

```
for(unsigned j=0; j< m_shader_config->n_simt_cores_per_cluster; j++)
{
    total_branches += m_cluster[i][0].m_core[j][0].all_branches;
    total_diverged_branches +=
m_cluster[i][0].m_core[j][0].diverged_branches;
    }
  }
  printf("gpu count of branches = %d\n", total_branches);
  printf("gpu count of diverged branches = %d\n", total_diverged_branches);
```

- **Line 972:** Added a print statement to display the total number of evictions.
  The change is shown below:
  - printf("L2 total cache evicts = %u\n", total_l2_css.evicts)

(5) shader.h

- **Line 1573:** Declared a function sum_nr_branch(). This function computes the total number of branches in all simt cores.
  The changes are shown below:
  - void sum_nr_branch();
- **Line 1841:** In class simt_core_cluster, a counter of total_branches_all is added to keep track of all the branches.
  The change is shown below:
  - unsigned total_branches_all;
- **Line 1885 and Line 1893:** The scope of the function shader_core_ctx **m_core was changed from public to private.
  The change is shown below:
  - shader_core_ctx **m_core;
  - //shader_core_ctx **m_core; - This access identifier of this method is changed to public//

6) shader.cc

- **Line 301:** A function called sum_nr_branch() of class shader_core_ctx is defined to compute the total number of branches in all the simt cores.
  The change is shown below:
  - void shader_core_ctx::sum_nr_branch()
    {
    m_cluster->total_branches_all += all_branches;
    }
- **Line 2041:** Added print statements to display the evictions.
  The original and changed code is shown below:
  - Original:
    fprintf(stdout, "\tL1D_cache_core[%d]: Access = %d, Miss = %d, Miss_rate = %.3lf, Pending_hits = %u, Reservation_fails = %u \n",i, css.accesses, css.misses, (double)css.misses / (double)css.accesses, css.pending_hits, css.res_fails);
  - Change:
  - fprintf(stdout, "\tL1D_cache_core[%d]: Access = %d, Miss = %d, Miss_rate = %.3lf, Pending_hits = %u, Reservation_fails = %u Evictions= %d\n", i,

css.accesses, css.misses, (double)css.misses / (double)css.accesses, css.pending_hits, css.res_fails, css.evicts);

- **Line 2056:** Added a print statement to display the total number of evictions.
  The change is shown below:
  - fprintf(fout, "\tL1D_total_evictions = %u\n", total_css.evicts);
- **Line 3184:** In simt_clore_cluster(), the total_branches_all counter is initialized to 0.
  The changes are shown below:
  - total_branches_all = 0;

(7) gpu-cache.h

- **Line 454:** In the structure cache_sub_stats which contains the general cache data such as the number of cache accesses, cache misses , reservation fails and the number of hits pending in the cache, another variable called evicts is added to keep track of the dirty evictions taking place.
  The changes are shown below:
  - unsigned evicts;
- **Line 473:** In the clear function, where the values of all the variables keeping track of the statistics of cache are reset, the newly created evicts variable is also cleared.
  The change are shown below:
  - evicts = 0;
- **Line 487:** In order to get the number of evictions for each core after each cycle, the value of css.evicts is added to evicts.
  The changes are shown below:
  - evicts += css.evicts;
- **Line 504:** To accumulate the value of evicts of each simt core after each cycle.
  The changes are shown below:
  - ret.evicts = evicts + cs.evicts;
- **Line 534:** A function is declared to copy the evictions and a variable called evictions is passed in the argument list.
  The changes are shown below:
  - void copy_evicts(int evictions);
- **Line 544:** A counter called total_evicts is declared to keep track of the evictions.
  The changes are shown below:
  - unsigned total_evicts;
- **Line 566 and Line 586:** A counter called evictions is declared in class baseline_cache and it is initialised to 0.
  The changes are shown below:
  - int evictions
  - evictions = 0;

3) **gpu-cache.cc**

- **Line 427:** The total evicts counter to count the number of evictions is initialised to 0 in cache_stats::cache_stats()
  The change is shown below:
  - total_evicts =0;
- **Line 439:** The total evicts counter to count the number of evictions is initialised to 0 in cache_stats::clear().
  The change is shown below:
  - evict_count =0;

- **Line 500:** In order to get the total number of evictions in all the cores, we accumulate the value of evictions in each core.
  The change is shown below:
  - ret.total_evicts = total_evicts + cs.total_evicts;
- **Line 517:** In order to get the total number of evictions in all the cores, we accumulate the value of evictions in each core.
  The change is shown below:
  - Total_evicts += cs.total_evicts;
- **Line 600**: t_css.evicts is assigned the value of the total number of evictions in all the cores.
  The change is shown below:
  - t_css.evicts = total_evicts;
- **Line 626 – Line 629:** A function is defined for copying the value of evictions to total_evicts.
  The changes are shown below:
  - void cache_stats::copy_evicts(int evictions)
    {
    total_evicts = evictions;
    }
- **Line 712:** Function call of the copy function.
  The change is shown below:
  - m_stats.copy_evicts(evictions);
- **Line 843:** incremented the counter evictions when there is a write hit and write evict.
  The change is shown below:
  - evictions++;
- **Line 916:** incremented the counter evictions when there is a write miss and write allocate.
  The change is shown below:
  - evictions++;
- **Line 1002:** incremented the counter evictions when there is read miss.
  The change is shown below:
  - evictions++;

OUTPUT AND STATISTICS:

gpu_sim_cycle = 224353

gpu_sim_insn = 12833777

gpu_ipc = 57.2035

gpu_tot_sim_cycle = 234520

gpu_tot_sim_insn = 12851163

gpu_tot_ipc = 54.7977

gpu_tot_issued_cta = 0

gpu count of branches = 49297

gpu count of diverged branches = 257

gpu_stall_dramfull = 139728

gpu_stall_icnt2sh = 522805

gpu_total_sim_rate=158656

========= Core cache stats =========

L1I_cache:

L1I_total_cache_accesses = 444228

L1I_total_cache_misses = 545

L1I_total_cache_miss_rate = 0.0012

L1I_total_cache_pending_hits = 0

L1I_total_cache_reservation_fails = 0

L1D_cache:

L1D_cache_core[0]: Access = 14155, Miss = 8669, Miss_rate = 0.612, Pending_hits = 85, Reservation_fails = 171991 Evictions= 3327

L1D_cache_core[1]: Access = 15212, Miss = 9768, Miss_rate = 0.642, Pending_hits = 98, Reservation_fails = 180580 Evictions= 3382

L1D_cache_core[2]: Access = 14282, Miss = 8713, Miss_rate = 0.610, Pending_hits = 81, Reservation_fails = 173224 Evictions= 3444

L1D_cache_core[3]: Access = 14062, Miss = 8650, Miss_rate = 0.615, Pending_hits = 106, Reservation_fails = 173136 Evictions= 3376

L1D_cache_core[4]: Access = 14042, Miss = 8601, Miss_rate = 0.613, Pending_hits = 131, Reservation_fails = 171616 Evictions= 3316

L1D_cache_core[5]: Access = 14207, Miss = 8717, Miss_rate = 0.614, Pending_hits = 81, Reservation_fails = 172056 Evictions= 3382

L1D_cache_core[6]: Access = 14366, Miss = 8794, Miss_rate = 0.612, Pending_hits = 103, Reservation_fails = 173091 Evictions= 3477

L1D_cache_core[7]: Access = 14489, Miss = 8794, Miss_rate = 0.607, Pending_hits = 75, Reservation_fails = 171761 Evictions= 3531

L1D_cache_core[8]: Access = 14159, Miss = 8700, Miss_rate = 0.614, Pending_hits = 104, Reservation_fails = 172421 Evictions= 3407

L1D_cache_core[9]: Access = 14227, Miss = 8604, Miss_rate = 0.605, Pending_hits = 122, Reservation_fails = 172479 Evictions= 3363

L1D_cache_core[10]: Access = 14198, Miss = 8702, Miss_rate = 0.613, Pending_hits = 87, Reservation_fails = 171867 Evictions= 3357

L1D_cache_core[11]: Access = 14320, Miss = 8706, Miss_rate = 0.608, Pending_hits = 87, Reservation_fails = 171516 Evictions= 3372

L1D_cache_core[12]: Access = 14184, Miss = 8670, Miss_rate = 0.611, Pending_hits = 103, Reservation_fails = 172155 Evictions= 3410

L1D_cache_core[13]: Access = 14431, Miss = 8680, Miss_rate = 0.601, Pending_hits = 96, Reservation_fails = 173120 Evictions= 3488

L1D_cache_core[14]: Access = 14237, Miss = 8671, Miss_rate = 0.609, Pending_hits = 69, Reservation_fails = 172548 Evictions= 3355

L1D_total_cache_accesses = 214571

L1D_total_cache_misses = 131439

L1D_total_cache_miss_rate = 0.6126

L1D_total_cache_pending_hits = 1428

L1D_total_evictions = 50987

L1D_total_cache_reservation_fails = 2593561

L1D_cache_data_port_util = 0.039

L1D_cache_fill_port_util = 0.039

RESULTS: -

1) The number of L1 data cache dirty evictions = 50987

2) The number of warps (NOT threads) that executed conditional branch instructions and had divergence = 257

3) The total number of warps (NOT threads) that executed conditional branch instructions (whether they had divergence or not) = 49297