Programming Assignment 3a

Working with GPGPU Simulator Load Bypass Logic

Nirmal Kumar Sedhumadhavan

Added a counter(load_bypasses) that all the load instructions in the address range of 0xc0000000 - 0xc00fffff bypass the L1 data cache in shader.h.

shader_core_stats_pod (line 1826 modified)

```
// memory access classification
int load_bypasses = 0; // Variables for PA3a
int gpgpu_n_mem_read_local;
int gpgpu_n_mem_write_local;
```

Added a boolean variable (load_in_range) to check if the instruction is a load type and falls in the address range of 0xc0000000 - 0xc00fffff in shader.cc. A bool variable 'bypassL1D' gets set to true if the above condition satisfies and the counter(load_bypasses) value is incremented.

ldst_unit::memory_cycle (line 2363 modified)

```
bool bypassL1D = false;
bool load_in_range = (inst.is_load() && access.get_addr() <= 0xc00fffff && access.get_addr() >= 0xc0000000); // PA3a load bypass mechanism

if (CACHE_GLOBAL == inst.cache_op || (m_L1D == NULL) || load_in_range)
{
      if(load_in_range) m_stats->load_bypasses++;
      bypassL1D = true;
}
else if (inst.space.is_global())
{ // global memory access
  // skip L1 cache if the option is enabled
  if (m_core->get_config()->gmem_skip_L1D && (CACHE_L1 != inst.cache_op))
    bypassL1D = true;
}
```

ldst_unit::cycle (line 3002 modified)

```
bool bypassL1D = false;
  bool load_in_range = (mf->get_inst().is_load() && mf->get_addr() <= 0xc00fffff && mf->get_addr() >= 0xc0000000); // PA3a load bypass mechanism

if (CACHE_GLOBAL == mf->get_inst().cache_op || (m_L1D == NULL) || load_in_range)
{
  bypassL1D = true;
}
else if (mf->get_access_type() == GLOBAL_ACC_R ||
         mf->get_access_type() ==
             GLOBAL_ACC_W)
{ // global memory access
  if (m_core->get_config()->gmem_skip_L1D)
    bypassL1D = true;
}
```

Added a print statement as highlighted in the snippet below to dump values of the counter

shader_core_stats::print (line665 modified)

```
//PA3a Print the bypassed load instructions
fprintf(fout, "bypassed load instructions: %d\n", load_bypasses);

fprintf(fout, "gpgpu_n_tot_thrd_icount = %lld\n", thread_icount_uarch);
fprintf(fout, "gpgpu_n_tot_w_icount = %lld\n", warp_icount_uarch);
```

Output (SM7_TITANV)

```
Total_core_cache_fail_stats:
ctas_completed 256, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
210, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,
bypassed load instructions: 26
```

```
Total_core_cache_fail_stats:
ctas_completed 512, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
230, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 20, 20, 20, 20, 20, 20, 20, 20,
bypassed load instructions: 163
```

```
Total_core_cache_fail_stats:
ctas_completed 768, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
250, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 345, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 20, 20, 20, 20, 20, 20, 20, 20,
bypassed load instructions: 963
```

```
Total_core_cache_fail_stats:
ctas_completed 1024, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
270, 80, 80, 80, 80, 80, 80, 80, 218, 323, 80, 302, 80, 650, 80, 80, 80, 80, 80, 80, 80, 80, 80, 80, 20, 20, 20, 20, 20, 20, 20, 20,
bypassed load instructions: 5884
```

```
Total_core_cache_fail_stats:
ctas_completed 1280, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
290, 100, 301, 100, 100, 100, 238, 100, 238, 343, 301, 481, 100, 808, 100, 100, 238, 100, 100, 280, 228, 332, 100, 100, 20, 20, 20, 20, 20, 20, 20, 20,
bypassed load instructions: 34108
```

```
Total_core_cache_fail_stats:
        Total_core_cache_fail_stats_breakdown[GLOBAL_ACC_W][MISS_QUEUE_FULL] = 1722
ctas_completed 1536, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
500, 342, 574, 373, 352, 342, 501, 321, 396, 574, 596, 786, 426, 1039, 289, 342, 386, 373, 436, 522, 512, 679, 426, 331, 283, 294, 304, 40, 241, 241, 272, 2
94,
bypassed load instructions: 181692
```

```
Total_core_cache_fail_stats:
        Total_core_cache_fail_stats_breakdown[GLOBAL_ACC_R][MISS_QUEUE_FULL] = 49654
        Total_core_cache_fail_stats_breakdown[GLOBAL_ACC_W][MISS_QUEUE_FULL] = 16238
ctas_completed 1792, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
794, 563, 827, 615, 627, 616, 776, 584, 648, 774, 838, 997, 732, 1335, 479, 606, 617, 679, 615, 764, 796, 911, 647, 638, 283, 294, 304, 40, 241, 241, 272, 2
94,
bypassed load instructions: 609201
```

```
Total_core_cache_fail_stats:
        Total_core_cache_fail_stats_breakdown[GLOBAL_ACC_R][MISS_QUEUE_FULL] = 49654
        Total_core_cache_fail_stats_breakdown[GLOBAL_ACC_W][MISS_QUEUE_FULL] = 16238
ctas_completed 2048, Shader 0 warp_id issue ditsribution:
warp_id:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
distro:
955, 681, 999, 755, 766, 767, 895, 755, 787, 871, 957, 1116, 883, 1465, 608, 758, 767, 840, 788, 882, 958, 1029, 808, 789, 283, 294, 304, 40, 241, 241, 272,
 294,
bypassed load instructions: 671956
```

Kernels	# of loads bypassed
Kernel 1	26
Kernel 2	163
Kernel 3	963
Kernel 4	5884
Kernel 5	34108
Kernel 6	181692
Kernel 7	609201
Kernel 8	671956
Kernel 9	674391
Kernel 10	674406