Programming Assignment 2:

Shantanu Mukhopadhyay[200541793]

Task 1: Instruction semantic

All changes were made to instruction.cc file.

Addition to power operation:

Running code: C[i] = A[i] + B[i]; // to test ADD on vectorAdd.cu

Under add_impl function:

The following changes are made to change the semantics of float(32-bits) point **add** instruction to **power** operation.

```
1061        case F32_TYPE:
1062          // data.f32 = src1_data.f32 + src2_data.f32;
1063          data.f32 = powf(src1_data.f32, src2_data.f32); // Changed the float 32 addition to a power operation
1064          break;
```

Output:

```
283   source vector A: 1.000 2.000 3.000 4.000 5.000 6.000 7.000 8.000 9.000 1.000
284   source vector B: 1.000 1.000 2.000 3.000 4.000 5.000 6.000 6.000 7.000 8.000
285   result vector C: 1.000 2.000 9.000 64.000 625.000 7776.000 117649.000 262144.000 4782969.000 1.000
```

Divide to subtract operation:

C[i] = A[i] / B[i]; // to test Divide

Under div_impl function:

The following changes are made to change the semantics of floating point(32 bits) **divide** to **subtract** operation.

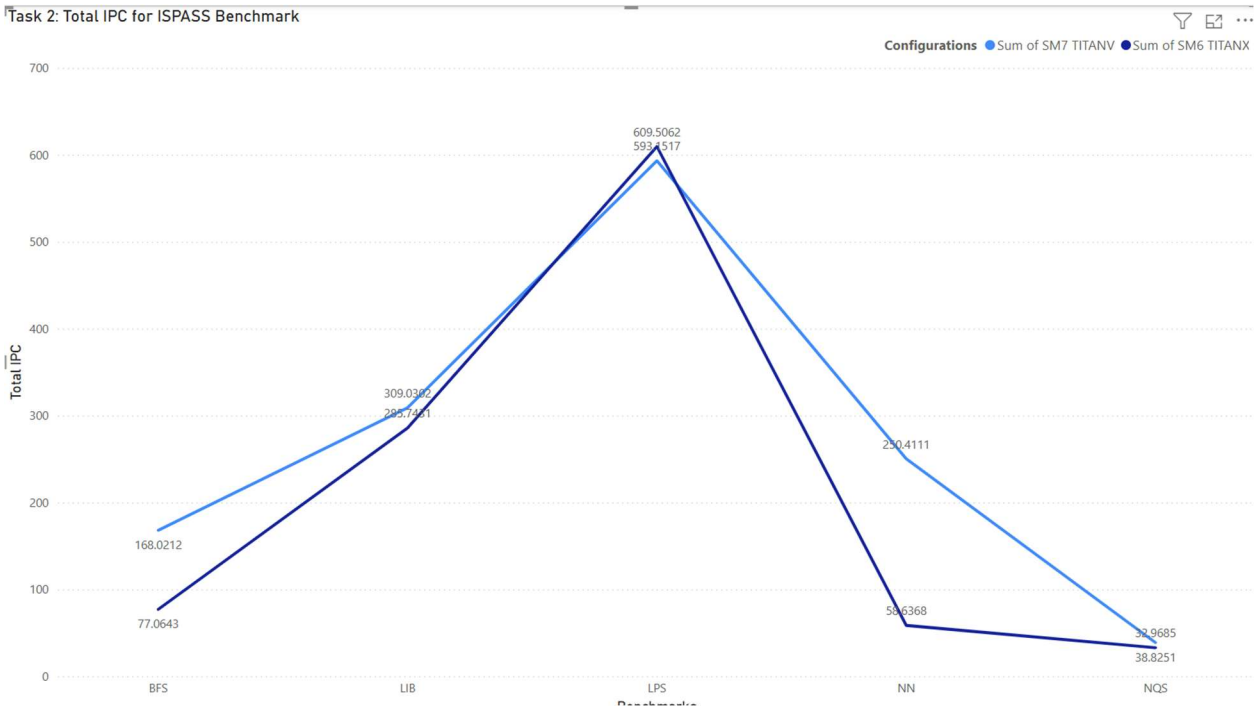
```
3193        case F32_TYPE:
3194          // data.f32 = src1_data.f32 / src2_data.f32;
3195          data.f32 = src1_data.f32 - src2_data.f32; // Changed the float-32 operation from division to subtract
3196          break;
```

Output:

```
283   source vector A: 1.000 2.000 3.000 4.000 5.000 6.000 7.000 8.000 9.000 1.000
284   source vector B: 1.000 1.000 2.000 3.000 4.000 5.000 6.000 6.000 7.000 8.000
285   result vector C: 0.000 1.000 1.000 1.000 1.000 1.000 1.000 2.000 2.000 -7.000
```

Task 2: Benchmark performance study

Benchmark	SM6_TITANX	SM7_TITANV
BFS	77.0643	168.0212
LIB	285.7431	309.0302
LPS	609.5062	593.1517
NN	58.6368	250.4111
NQS	32.9685	38.8251



- ➔ After running the benchmarks for both SM6 TITAN X and SM7 TITAN V we see that SM7 TITANV almost every time outperforms SM6 TITANX.
- ➔ The LPS benchmark has the highest IPC compared to the rest of the benchmarks, in order highest to lowest: LPS, LIB, NN, BFS and then NQS on average.
- ➔ This is due to the different architecture both of the configurations follow, the SM6 TITANX follows the Pascal (Pascal GP102) architecture and the SM7 TITANV follows the Volta (Volta Titan V)Architecture.

Task 3: Branch instructions and divergence

Under abstract_hardware_model.cc

Task 4: Memory access space

Changes in shader.h under shader_core_stats_pod for declaring variable for stats.

```
1682    //TASK 4 global and local memory accesses
1683    int gpgpu_n_mem_global_accesses;
1684    int gpgpu_n_mem_local_accesses;
```

Changes under shader.cc

Inside bool ldst_unit::memory_cycle function:

To track global memory accesses (added line 2049)

```
2043    const mem_access_t &access = inst.accessq_back();
2044
2045    bool bypassL1D = false;
2046    if (CACHE_GLOBAL == inst.cache_op || (m_L1D == NULL)) {
2047      bypassL1D = true;
2048    } else if (inst.space.is_global()) {  // global memory access
2049      m_stats->gpgpu_n_mem_global_accesses++;
2050      // skip L1 cache if the option is enabled
2051      if (m_core->get_config()->gmem_skip_L1D && (CACHE_L1 != inst.cache_op))
2052        bypassL1D = true;
2053    }
2054    if (bypassL1D) {
2055      // bypass L1 cache
```

To track local memory accesses

```
if (!inst.accessq_empty() && stall_cond == NO_RC_FAIL)
  stall_cond = COAL_STALL;
if (stall_cond != NO_RC_FAIL) {
  stall_reason = stall_cond;
  bool iswrite = inst.is_store();
  if (inst.space.is_local()){
    m_stats->gpgpu_n_mem_local_accesses++;
    access_type = (iswrite) ? L_MEM_ST : L_MEM_LD;
  }
    else
    access_type = (iswrite) ? G_MEM_ST : G_MEM_LD;
}
```

And print them from shader_core_stats::print function in shader.cc

```
601   //Print the global memory access count and local memory access count
602   fprintf(fout, "# of global memory access: %d\n", gpgpu_n_mem_global_accesses);
603   fprintf(fout, "# of local memory access: %d\n", gpgpu_n_mem_local_accesses);
```

Output:

```
2777   gpgpu_n_tot_thrd_icount = 83147904
2778   gpgpu_n_tot_w_icount = 2598372
2779   gpgpu_n_stall_shd_mem = 339200
2780   gpgpu_n_mem_read_local = 0
2781   gpgpu_n_mem_write_local = 0
2782   gpgpu_n_mem_read_global = 259000
2783   gpgpu_n_mem_write_global = 145000
2784   gpgpu_n_mem_texture = 0
2785   gpgpu_n_mem_const = 0
2786   # of global memory access: 448400
2787   # of local memory access: 0
2788   gpgpu_n_load_insn  = 1568800
2789   gpgpu_n_store_insn = 1000000
2790   gpgpu_n_shmem_insn = 13549960
2791   gpgpu_n_sstarr_insn = 0
```