Project 2 Report

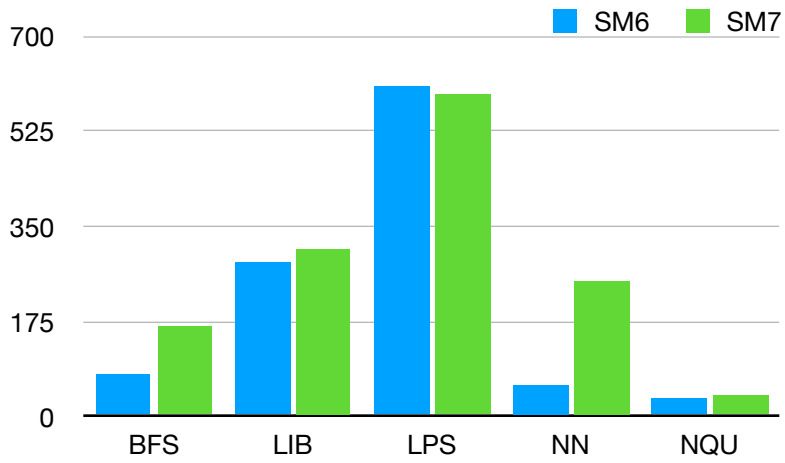
Task1:

Changes are made to where result of add instructions are computed. For the purpose of this task, simply change add to power, as shown below.

```
root@5264f688abda:~/gpgpu-sim_distribution/src/cuda-sim# git diff instructions.cc
diff --git a/src/cuda-sim/instructions.cc b/src/cuda-sim/instructions.cc
index 8936fa8..138a175 100644
--- a/src/cuda-sim/instructions.cc
+++ b/src/cuda-sim/instructions.cc
@@ -1342,19 +1342,19 @@ void atom_callback(const inst_t *inst, ptx_thread_info *thread) {
     case ATOMIC_ADD: {
       switch (to_type) {
         case U32_TYPE:
-          op_result.u32 = data.u32 + src2_data.u32;
+          op_result.u32 = pow(data.u32, src2_data.u32);
           data_ready = true;
           break;
         case S32_TYPE:
-          op_result.s32 = data.s32 + src2_data.s32;
+          op_result.s32 = pow(data.s32, src2_data.s32);
           data_ready = true;
           break;
         case U64_TYPE:
-          op_result.u64 = data.u64 + src2_data.u64;
+          op_result.u64 = pow(data.u64, src2_data.u64);
           data_ready = true;
           break;
         case F32_TYPE:
-          op_result.f32 = data.f32 + src2_data.f32;
+          op_result.f32 = pow(data.f32, src2_data.f32);
           data_ready = true;
           break;
         default:
```

Task2:

	SM6_TITANX	SM7_TITANV
BFS	77.06	168.02
LIB	285.74	309.03
LPS	609.51	593.15
NN	58.64	250.41
NQU	32.97	38.83



In general, SM7_TITANV outperform SM6_TITANX in 4 out of 5 benchmarks, presumably due to difference between memory bandwidth & # of cores.

Task3:

Changes are made to where entry is pushed into the SIMT stack. For entries that has a new re-convergence PC, increment count for divergence branches; Else increment count for total branches. Code changes are shown below.

```
root@5264f688abda:~/gpgpu-sim_distribution/src# git diff abstract_hardware_model.cc
diff --git a/src/abstract_hardware_model.cc b/src/abstract_hardware_model.cc
index 5ad6f10..3b45ea9 100644
--- a/src/abstract_hardware_model.cc
+++ b/src/abstract_hardware_model.cc
@@ -40,6 +40,9 @@
 #include "gpgpusim_entrypoint.h"
 #include "option_parser.h"

+int g_div_branch_count = 0;
+int g_branch_count = 0;
+
 void mem_access_t::init(gpgpu_context *ctx) {
   gpgpu_ctx = ctx;
   m_uid = ++(gpgpu_ctx->sm_next_access_uid);
@@ -1147,7 +1150,7 @@ void simt_stack::update(simt_mask_t &thread_done, addr_vector_t &next_pc,
         m_stack.back().m_pc = new_recvg_pc;
         m_stack.back().m_branch_div_cycle =
             m_gpu->gpu_sim_cycle + m_gpu->gpu_tot_sim_cycle;
-
+        g_div_branch_count++;                    // task3
         m_stack.push_back(simt_stack_entry());
       }
     }
@@ -1164,7 +1167,7 @@ void simt_stack::update(simt_mask_t &thread_done, addr_vector_t &next_pc,
     } else {
       m_stack.back().m_recvg_pc = top_recvg_pc;
     }
-
+    g_branch_count++;                          // task3
     m_stack.push_back(simt_stack_entry());
   }
   assert(m_stack.size() > 0);
```

```
diff --git a/src/abstract_hardware_model.h b/src/abstract_hardware_model.h
index 49f3e9f..fa72a7c 100644
--- a/src/abstract_hardware_model.h
+++ b/src/abstract_hardware_model.h
@@ -29,6 +29,9 @@
 #ifndef ABSTRACT_HARDWARE_MODEL_INCLUDED
 #define ABSTRACT_HARDWARE_MODEL_INCLUDED

+extern int g_div_branch_count;
+extern int g_branch_count;
+
 // Forward declarations
 class gpgpu_sim;
 class kernel_info_t;
```

Task4:

Changes are made to issue stage of load/store unit. Simply check the memory-space access type of the instruction every time an instruction is issued, and increment corresponding count. Code changes are shown below.

```
root@5264f688abda:~/gpgpu-sim_distribution/src# git diff gpgpu-sim/shader.cc
diff --git a/src/gpgpu-sim/shader.cc b/src/gpgpu-sim/shader.cc
index c6e7b8f..8b9d233 100644
--- a/src/gpgpu-sim/shader.cc
+++ b/src/gpgpu-sim/shader.cc
@@ -52,6 +52,9 @@
 #define MAX(a, b) (((a) > (b)) ? (a) : (b))
 #define MIN(a, b) (((a) < (b)) ? (a) : (b))

+int g_local_mem_access = 0;
+int g_global_mem_access = 0;
+
 mem_fetch *shader_core_mem_fetch_allocator::alloc(
     new_addr_type addr, mem_access_type type, unsigned size, bool wr,
     unsigned long long cycle) const {
@@ -2412,6 +2415,9 @@ void ldst_unit::issue(register_set &reg_set) {
     }
   }

+  if (inst->space.get_type() == global_space) g_global_mem_access++;       // task 4
+  else if (inst->space.get_type() == local_space) g_local_mem_access++;
+
   inst->op_pipe = MEM__OP;
   // stat collection
   m_core->mem_instruction_stats(*inst);
```

```
root@5264f688abda:~/gpgpu-sim_distribution/src# git diff gpgpu-sim/shader.h
diff --git a/src/gpgpu-sim/shader.h b/src/gpgpu-sim/shader.h
index 6481790..458138b 100644
--- a/src/gpgpu-sim/shader.h
+++ b/src/gpgpu-sim/shader.h
@@ -70,6 +70,9 @@

 #define WRITE_MASK_SIZE 8

+extern int g_local_mem_access;
+extern int g_global_mem_access;
+
 class gpgpu_context;

 enum exec_unit_type_t {
```