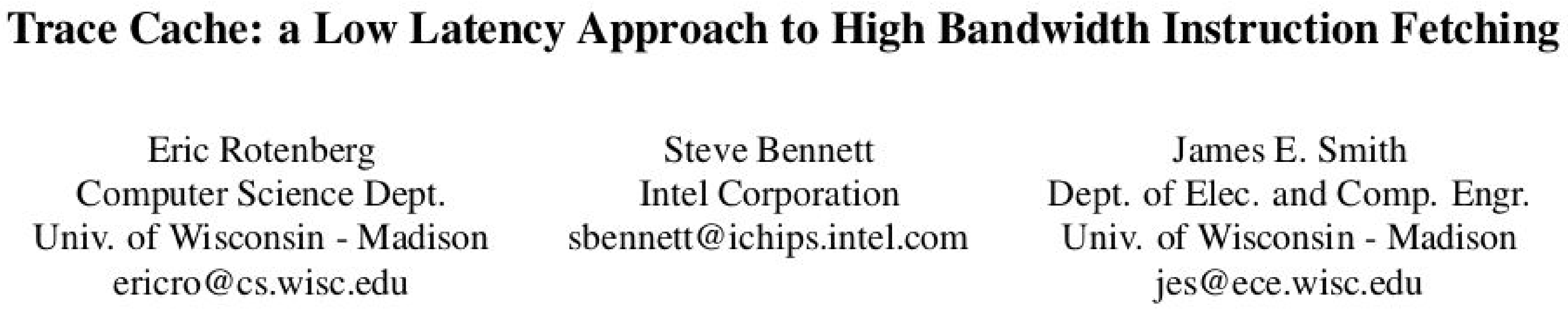
Projeto 4 - Apresentação

Unicamp - Universidade Estadual de Campinas
Paulo Henrique Silva Ribeiro
RA 181806
MO601

Artigo Escolhido



Trace Cache: a Low Latency Approach to High Bandwidth Instruction Fetching

Eric Rotenberg
Computer Science Dept.
Univ. of Wisconsin - Madison
ericro@cs.wisc.edu

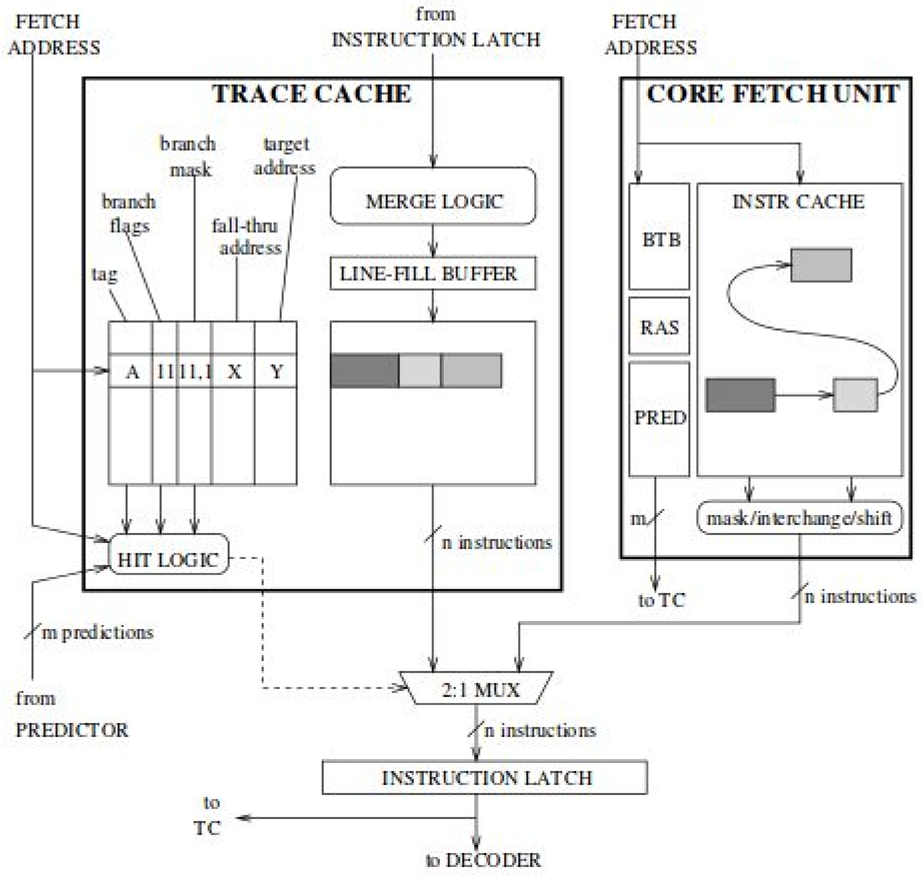
Steve Bennett
Intel Corporation
sbennett@ichips.intel.com

James E. Smith
Dept. of Elec. and Comp. Engr.
Univ. of Wisconsin - Madison
jes@ece.wisc.edu

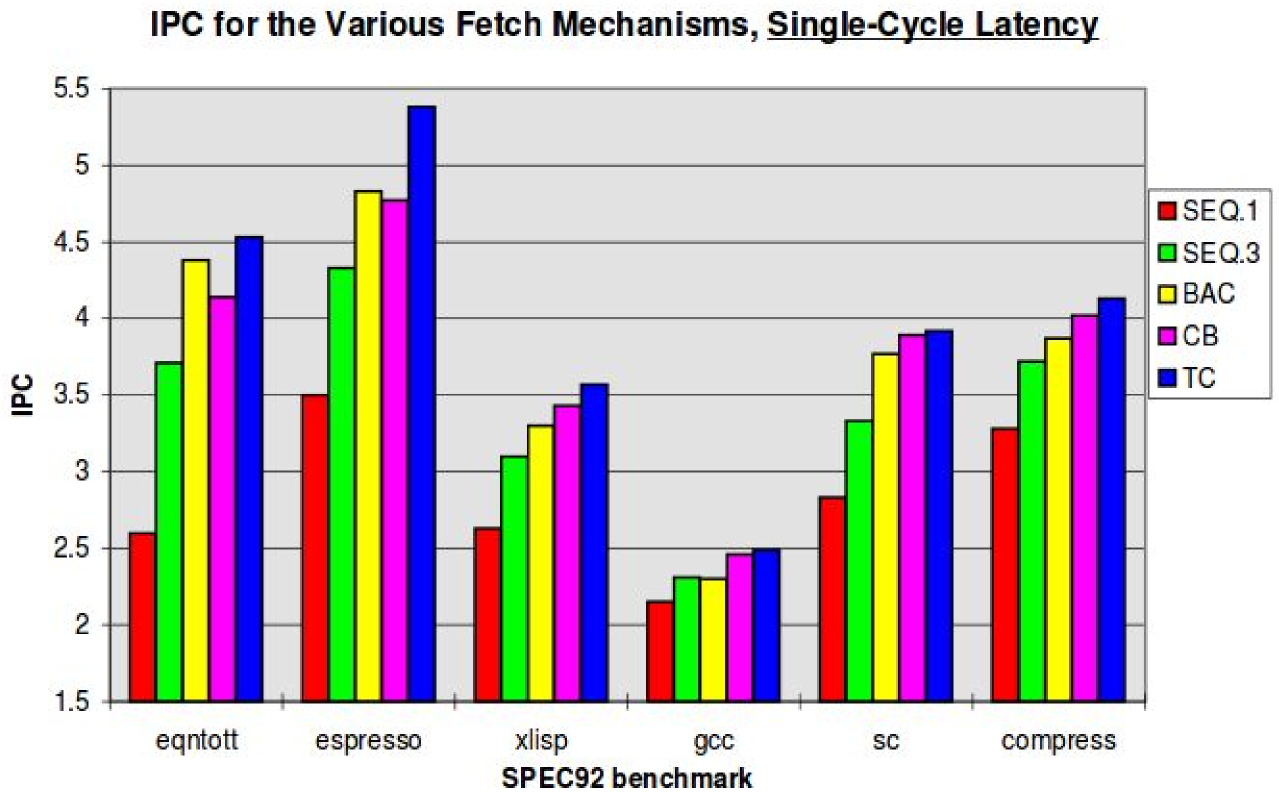
Trace Cache

- Não pretende replicar a cache de instrução convencional nem o hardware de fetch.

- New Fetch mechanism



Results



IPC for the Various Fetch Mechanisms, Single-Cycle Latency
IPC
5.5
5
4.5
4
3.5
3
2.5
2
1.5
eqntott
espresso
xlisp
gcc
sc
compress
SPEC92 benchmark
SEQ.1
SEQ.3
BAC
CB
TC

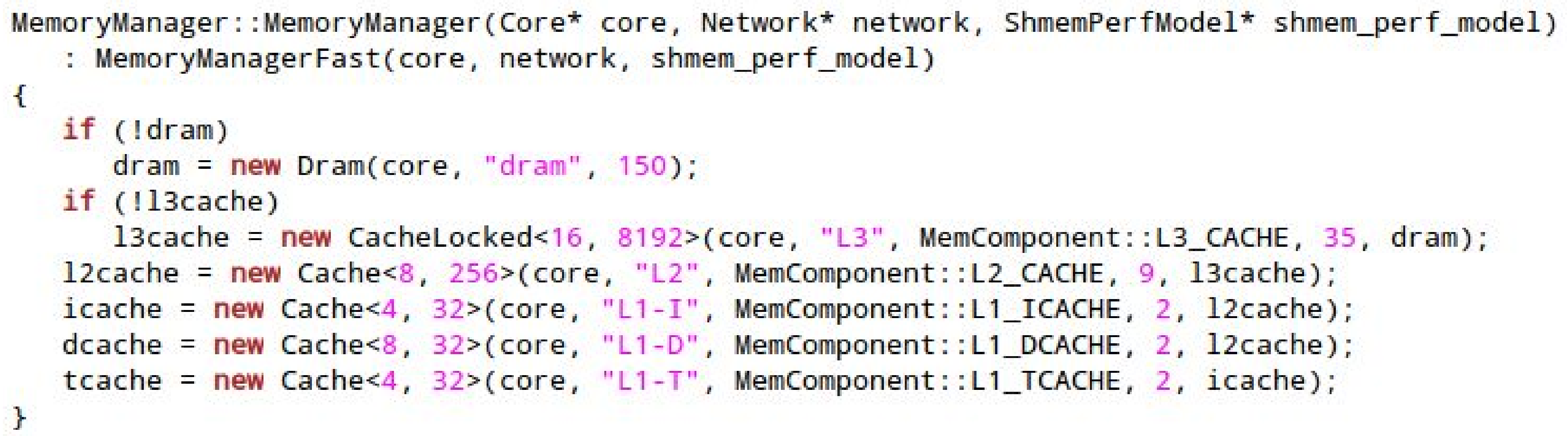
Implementação

- Calcular IPC com outros tipos de preditores, para comparar com o trace cache, fazendo a análise do “High Bandwidth Instruction Fetching”;

- Transformar L1-I no próximo nível de cache da Trace Cache;

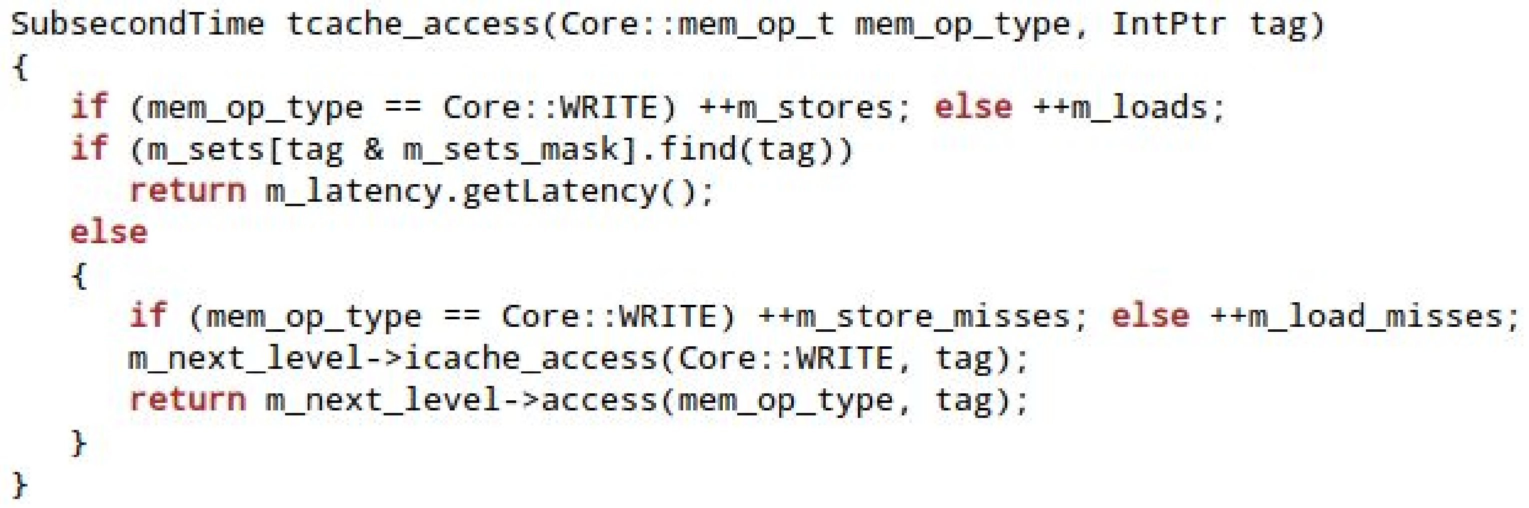
- Relatar e analisar a quantidade de hit/miss nas caches;

Implementação



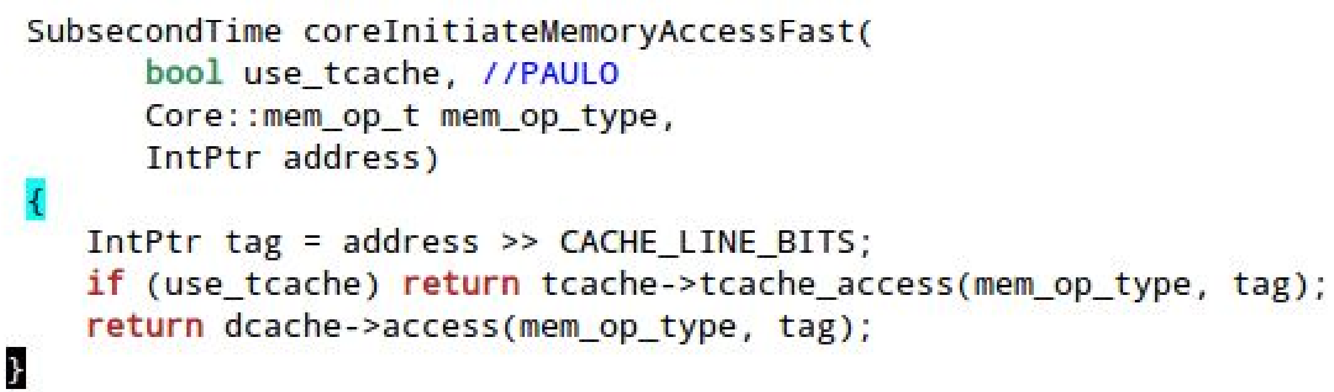
```
MemoryManager::MemoryManager(Core* core, Network* network, ShmemPerfModel* shmem_perf_model)
   : MemoryManagerFast(core, network, shmem_perf_model)
{
   if (!dram)
      dram = new Dram(core, "dram", 150);
   if (!l3cache)
      l3cache = new CacheLocked<16, 8192>(core, "L3", MemComponent::L3_CACHE, 35, dram);
   l2cache = new Cache<8, 256>(core, "L2", MemComponent::L2_CACHE, 9, l3cache);
   icache = new Cache<4, 32>(core, "L1-I", MemComponent::L1_ICACHE, 2, l2cache);
   dcache = new Cache<8, 32>(core, "L1-D", MemComponent::L1_DCACHE, 2, l2cache);
   tcache = new Cache<4, 32>(core, "L1-T", MemComponent::L1_TCACHE, 2, icache);
}
```

Implementação



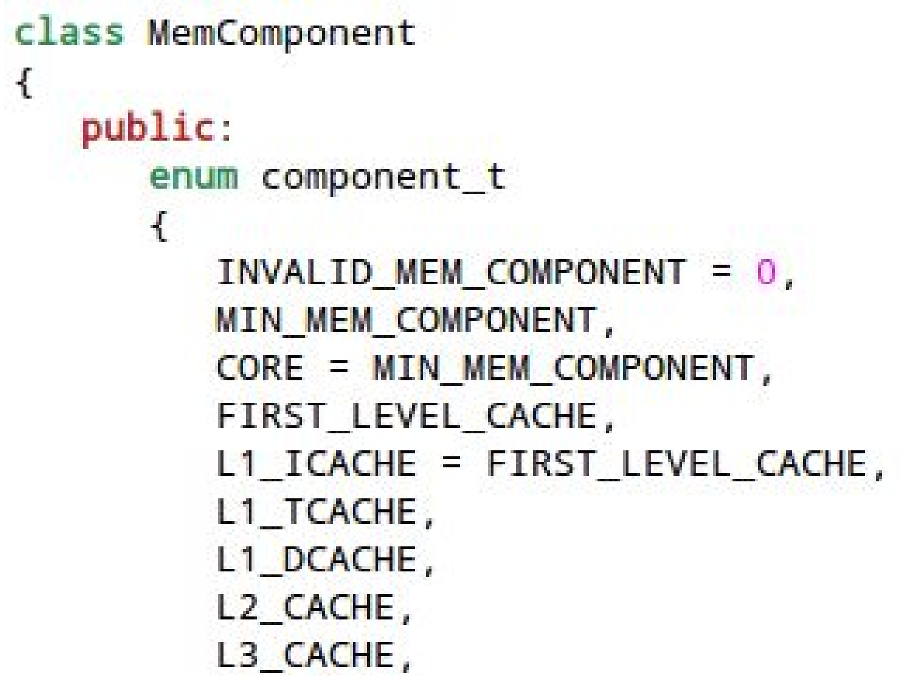
```
SubsecondTime tcache_access(Core::mem_op_t mem_op_type, IntPtr tag)
{
   if (mem_op_type == Core::WRITE) ++m_stores; else ++m_loads;
   if (m_sets[tag & m_sets_mask].find(tag))
      return m_latency.getLatency();
   else
   {
      if (mem_op_type == Core::WRITE) ++m_store_misses; else ++m_load_misses;
      m_next_level->icache_access(Core::WRITE, tag);
      return m_next_level->access(mem_op_type, tag);
   }
}
```

Implementação



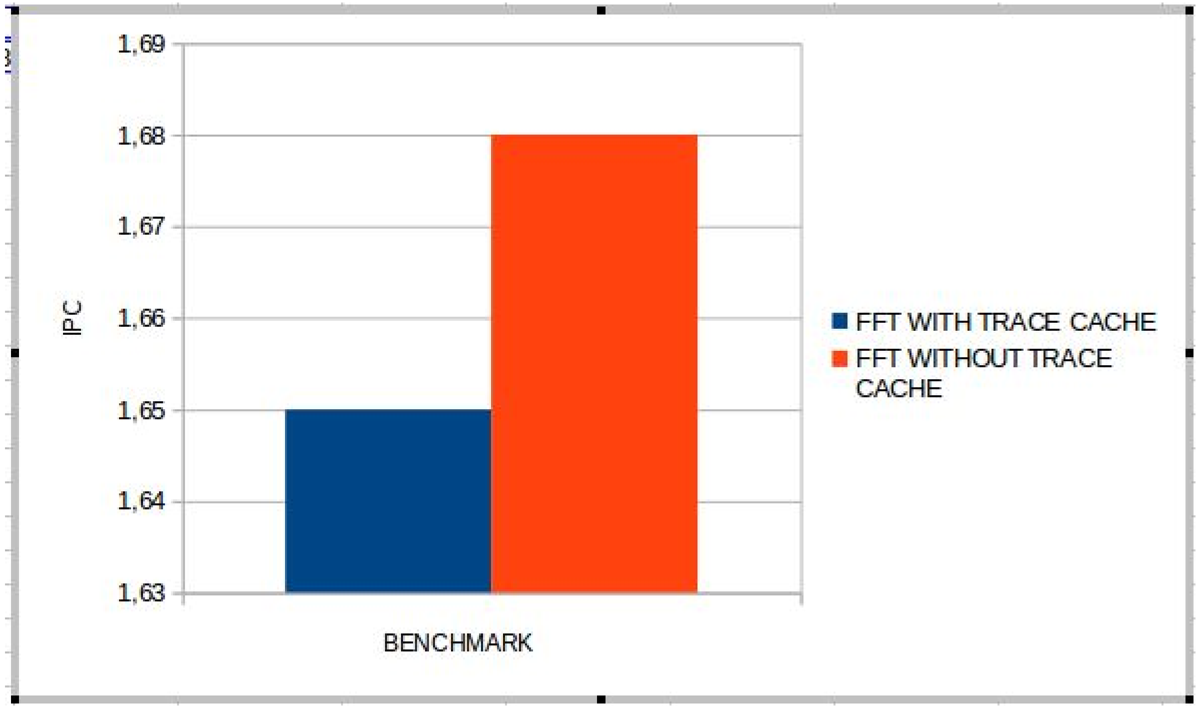
```
SubsecondTime coreInitiateMemoryAccessFast(
      bool use_tcache, //PAULO
      Core::mem_op_t mem_op_type,
      IntPtr address)
{
   IntPtr tag = address >> CACHE_LINE_BITS;
   if (use_tcache) return tcache->tcache_access(mem_op_type, tag);
   return dcache->access(mem_op_type, tag);
}
```

Implementação



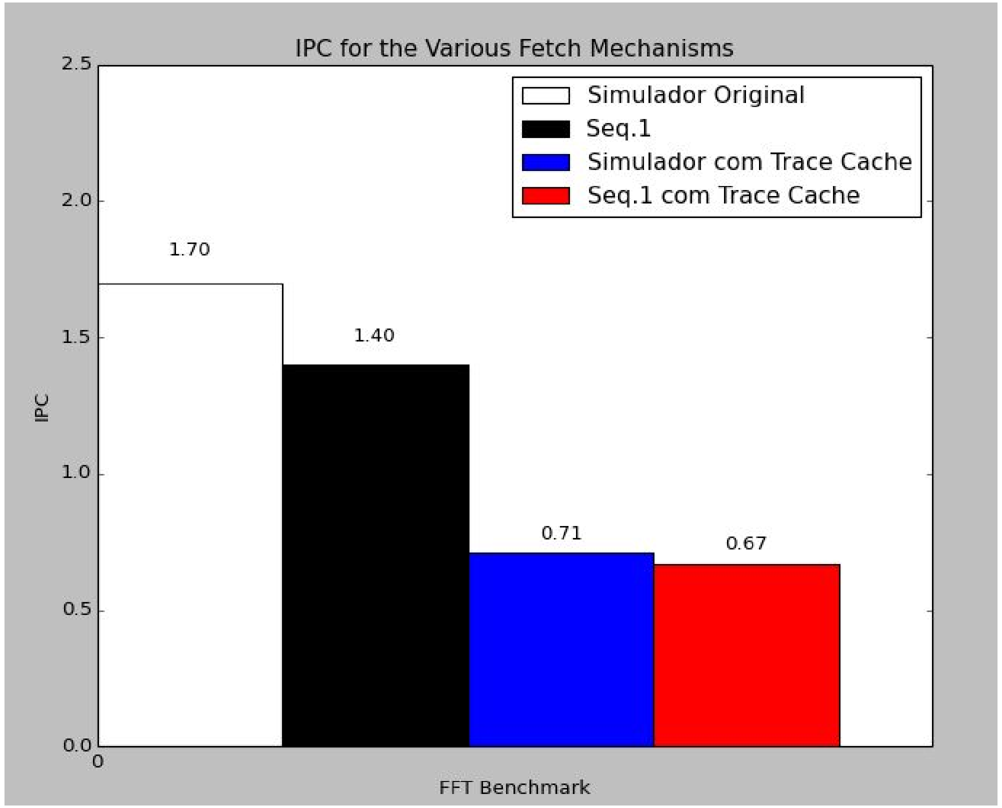
```
class MemComponent
{
   public:
      enum component_t
      {
         INVALID_MEM_COMPONENT = 0,
         MIN_MEM_COMPONENT,
         CORE = MIN_MEM_COMPONENT,
         FIRST_LEVEL_CACHE,
         L1_ICACHE = FIRST_LEVEL_CACHE,
         L1_TCACHE,
         L1_DCACHE,
         L2_CACHE,
         L3_CACHE,
```

Results

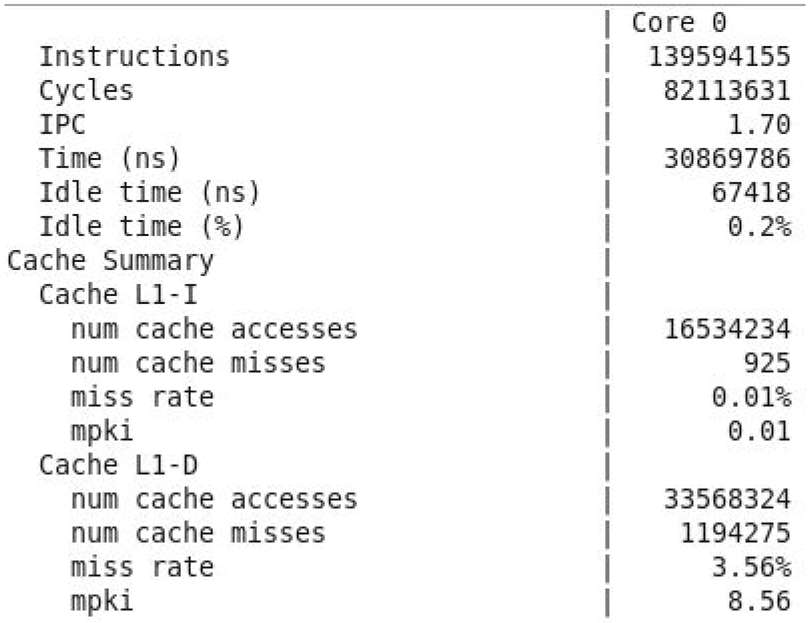


1,69
1,68
1,67
1,66
1,65
1,64
1,63
IPC
BENCHMARK
FFT WITH TRACE CACHE
FFT WITHOUT TRACE CACHE

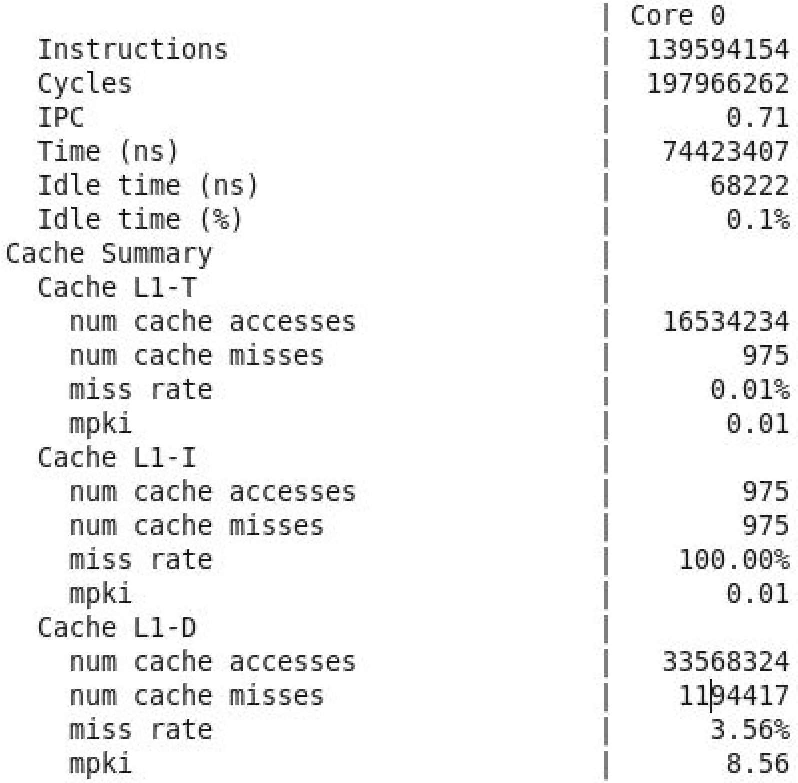
Results


IPC for the Various Fetch Mechanisms
Simulador Original
Seq.1
Simulador com Trace Cache
Seq.1 com Trace Cache
1.70
1.40
0.71
0.67
IPC
FFT Benchmark
0.0
0.5
1.0
1.5
2.0
2.5
0

Results



```
                                         | Core 0
  Instructions                           | 139594155
  Cycles                                 |  82113631
  IPC                                    |      1.70
  Time (ns)                              |  30869786
  Idle time (ns)                         |     67418
  Idle time (%)                          |      0.2%
Cache Summary                            |
  Cache L1-I                             |
    num cache accesses                   |  16534234
    num cache misses                     |       925
    miss rate                            |     0.01%
    mpki                                 |      0.01
  Cache L1-D                             |
    num cache accesses                   |  33568324
    num cache misses                     |   1194275
    miss rate                            |     3.56%
    mpki                                 |      8.56
```



```
                                         | Core 0
  Instructions                           | 139594154
  Cycles                                 | 197966262
  IPC                                    |      0.71
  Time (ns)                              |  74423407
  Idle time (ns)                         |     68222
  Idle time (%)                          |      0.1%
Cache Summary                            |
  Cache L1-T                             |
    num cache accesses                   |  16534234
    num cache misses                     |       975
    miss rate                            |     0.01%
    mpki                                 |      0.01
  Cache L1-I                             |
    num cache accesses                   |       975
    num cache misses                     |       975
    miss rate                            |   100.00%
    mpki                                 |      0.01
  Cache L1-D                             |
    num cache accesses                   |  33568324
    num cache misses                     |   1194417
    miss rate                            |     3.56%
    mpki                                 |      8.56
```

Conclusão