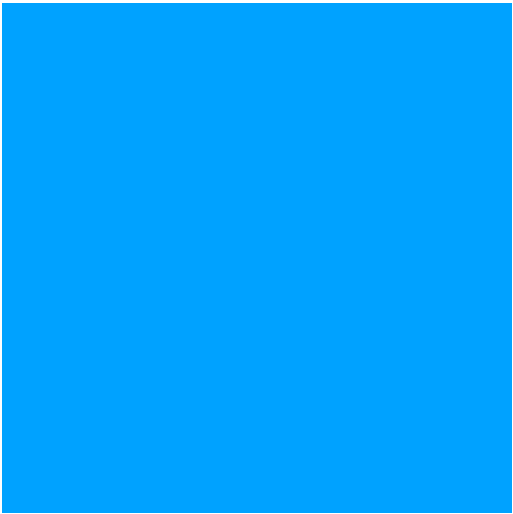
Memory Access

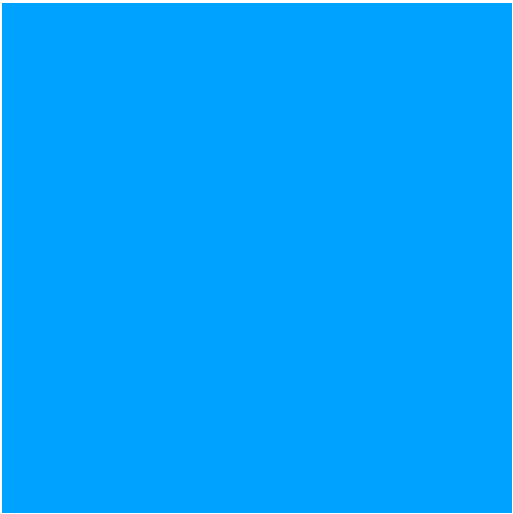

ADC


L1


L2


RAM





3 cycles

20 cycles

200+ cycles

20

Memory Access

Priority Inversion

```
// Shared state and mutex to synchronise access to it
std::vector<float> vec;
std::mutex mutex;


// Thread 1 - High priority
{
    std::scoped_lock<std::mutex> lock (mutex);      // Lock access to vec
    std::for_each (vec.begin(), vec.end(),          // Process vec
                   [] (auto& f) { f *= 0.1f; });
}

// Thread 2 - Low priority
{
    std::scoped_lock<std::mutex> lock (mutex);      // Lock access to vec
    vec.resize (500'000);                           // Perform expensive operation
}                                                   // Could be de-scheduled
```