

MGPUSim

<https://gitlab.com/akita/gcn3>

Yifan Sun, Trinayan Baruah, Shi Dong, David Kaeli

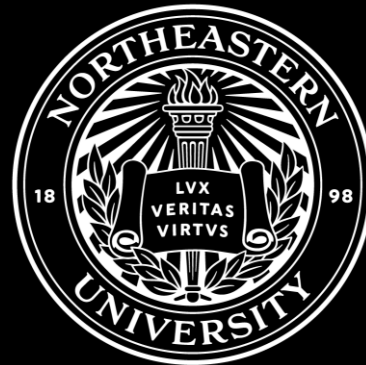
Northeastern University



yifansun@ece.neu.edu

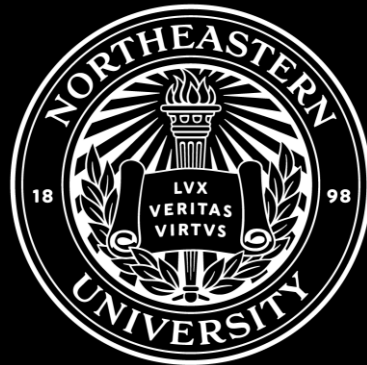


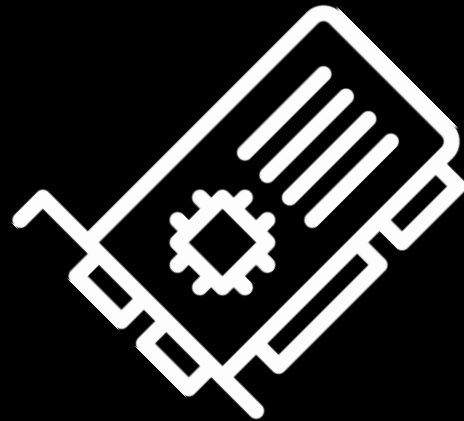
@_syifan_

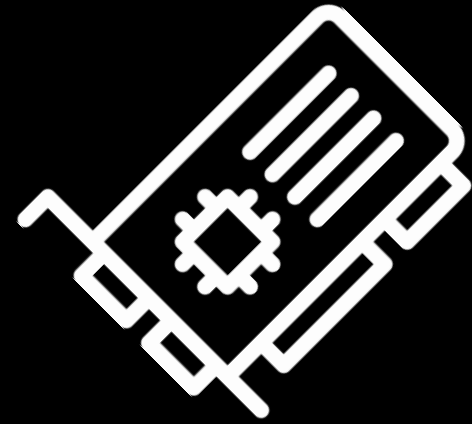
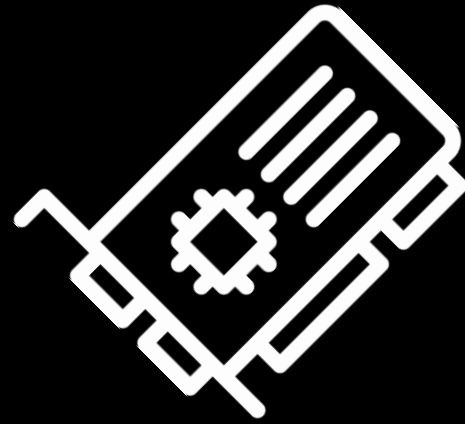
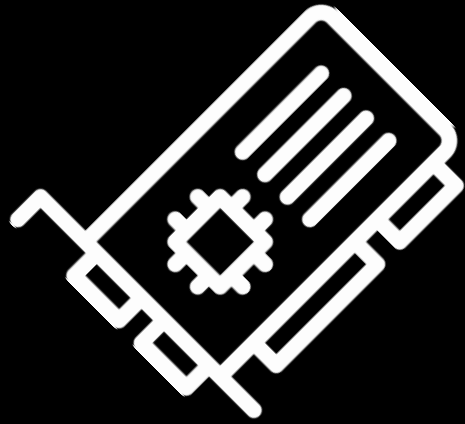
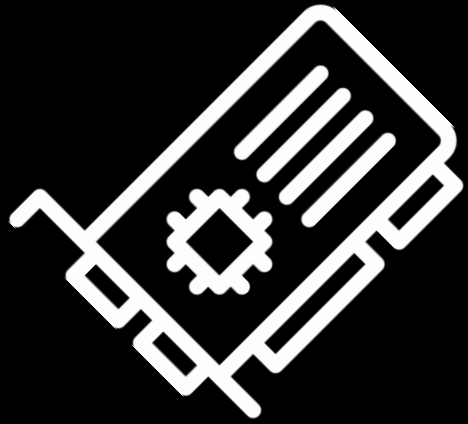


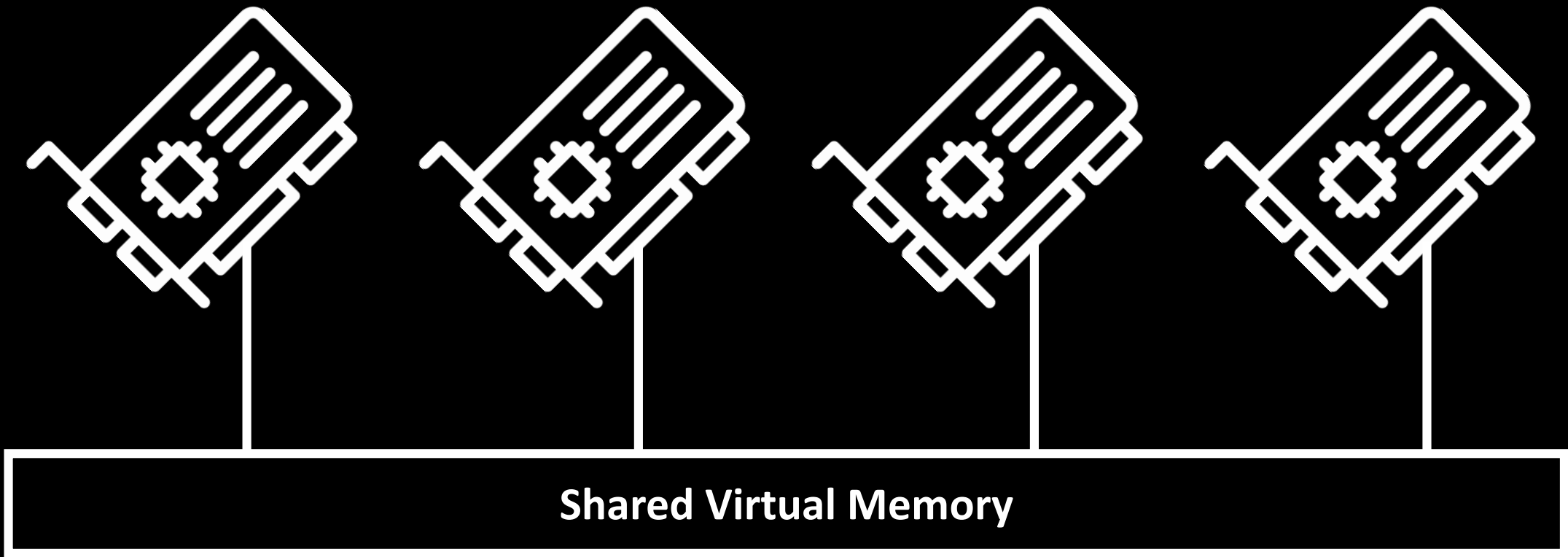
MGPUSim: a Flexible High-Performance Simulator for Multi-GPU Systems

Yifan Sun, Trinayan Baruah, Shi Dong, David Kaeli
Northeastern University











MGPUSim

OpenCL Kernels
AMD GCN3 ISA
Multi-GPU Simulation
Parallel Simulation

MGPUSim



AKITA

Go



Less is More

```
yifan@YIFANs-MBP ~$ git clone https://gitlab.com/akita/gcn3
Cloning into 'gcn3'...
warning: redirecting to https://gitlab.com/akita/gcn3.git/
remote: Enumerating objects: 10585, done.
remote: Counting objects: 100% (10585/10585), done.
remote: Compressing objects: 100% (3086/3086), done.
remote: Total 10585 (delta 7270), reused 10452 (delta 7193)iB/s
Receiving objects: 100% (10585/10585), 162.88 MiB | 9.79 MiB/s, done.
Resolving deltas: 100% (7270/7270), done.
yifan@YIFANs-MBP ~$ cd gcn3
yifan@YIFANs-MBP ~/gcn3$ git checkout master
```

**Do not modify my code,
extend it.**

```
func main() {  
    runner := new(runner.Runner).ParseFlag().Init()  
  
    benchmark := fir.NewBenchmark(runner.GPUDriver)  
  
    runner.AddBenchmark(benchmark)  
  
    runner.Run()  
}
```

```
func main() {  
    runner := new(runner.Runner).ParseFlag().Init()  
  
    benchmark := fir.NewBenchmark(runner.GPUDriver)  
  
    runner.AddBenchmark(benchmark)  
  
    runner.Run()  
}
```

```
func main() {  
    runner := new(runner.Runner).ParseFlag().Init()  
  
    benchmark := fir.NewBenchmark(runner.GPUDriver)  
  
    runner.AddBenchmark(benchmark)  
  
    runner.Run()  
}
```

```
func main() {  
    runner := new(runner.Runner).ParseFlag().Init()  
  
    benchmark := fir.NewBenchmark(runner.GPUDriver)  
  
    runner.AddBenchmark(benchmark)  
  
    runner.Run()  
}
```


Code > Configuration File

```
func (r *Runner) Init() *Runner {  
    if r.Timing {  
        r.Engine, r.GPUDriver = platform.BuildNR9NanoPlatform(4)  
    } else {  
        r.Engine, r.GPUDriver = platform.BuildEmuPlatform()  
    }  
    return r  
}
```

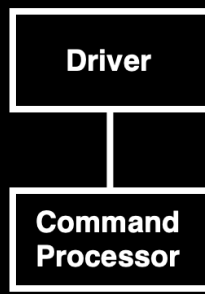
**Modular,
composable,
pluggable,
hookable**

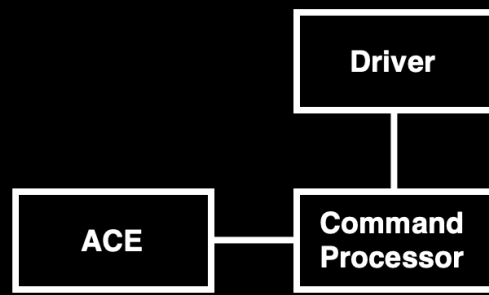
```
func (r *Runner) Init() *Runner {
    if r.Timing {
        r.Engine, r.GPUDriver = platform.BuildNR9NanoPlatform(4)
    } else {
        r.Engine, _, r.GPUDriver, _ = platform.BuildEmuPlatform()
    }

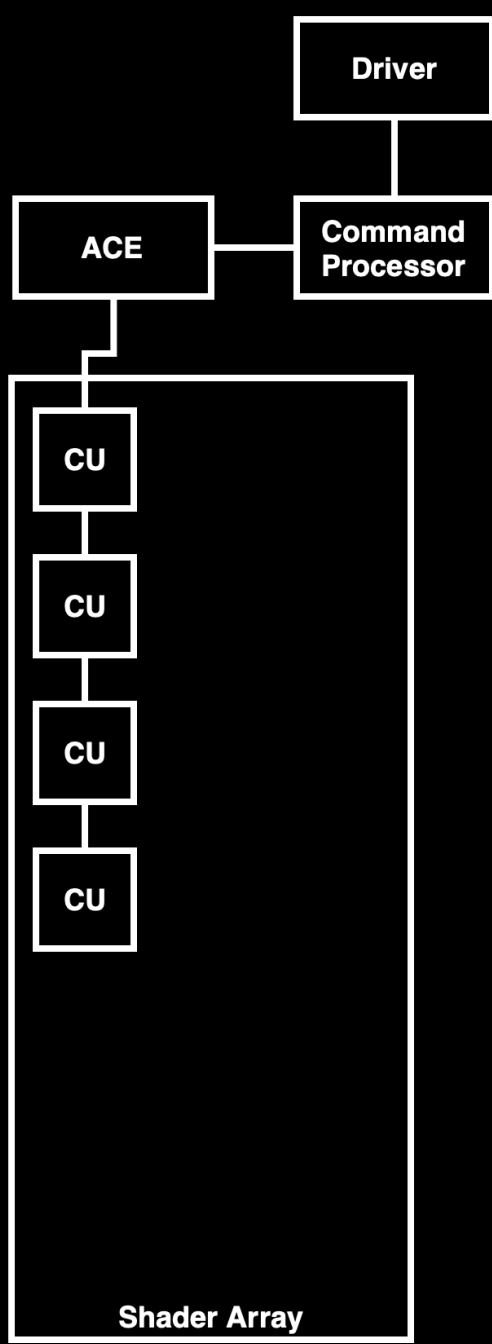
    r.KernelTimeCounter = driver.NewKernelTimeCounter()
    r.GPUDriver.AcceptHook(r.KernelTimeCounter)

    return r
}
```

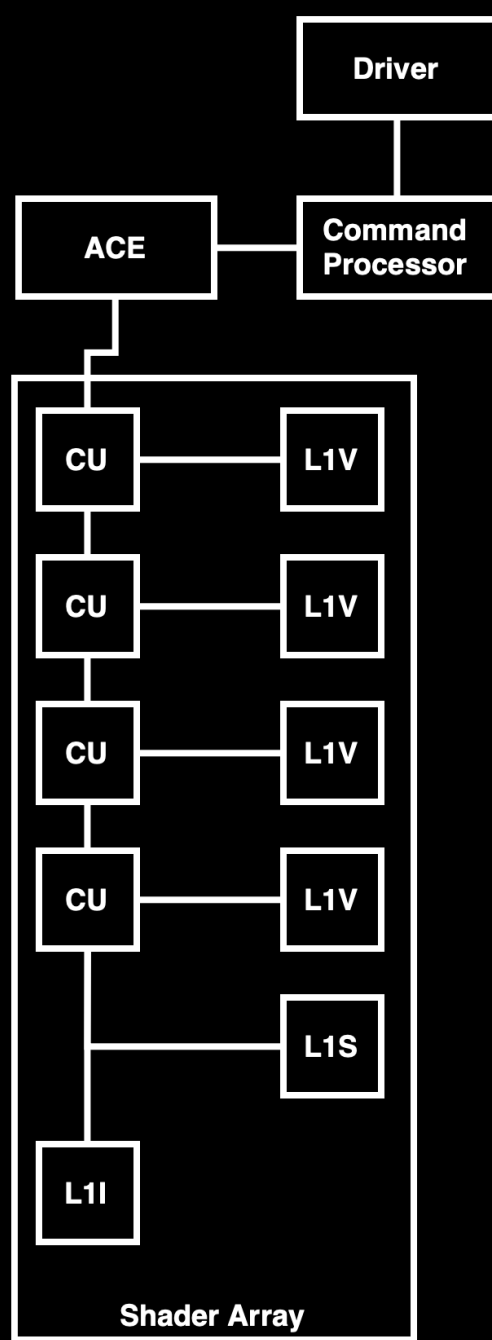
Driver



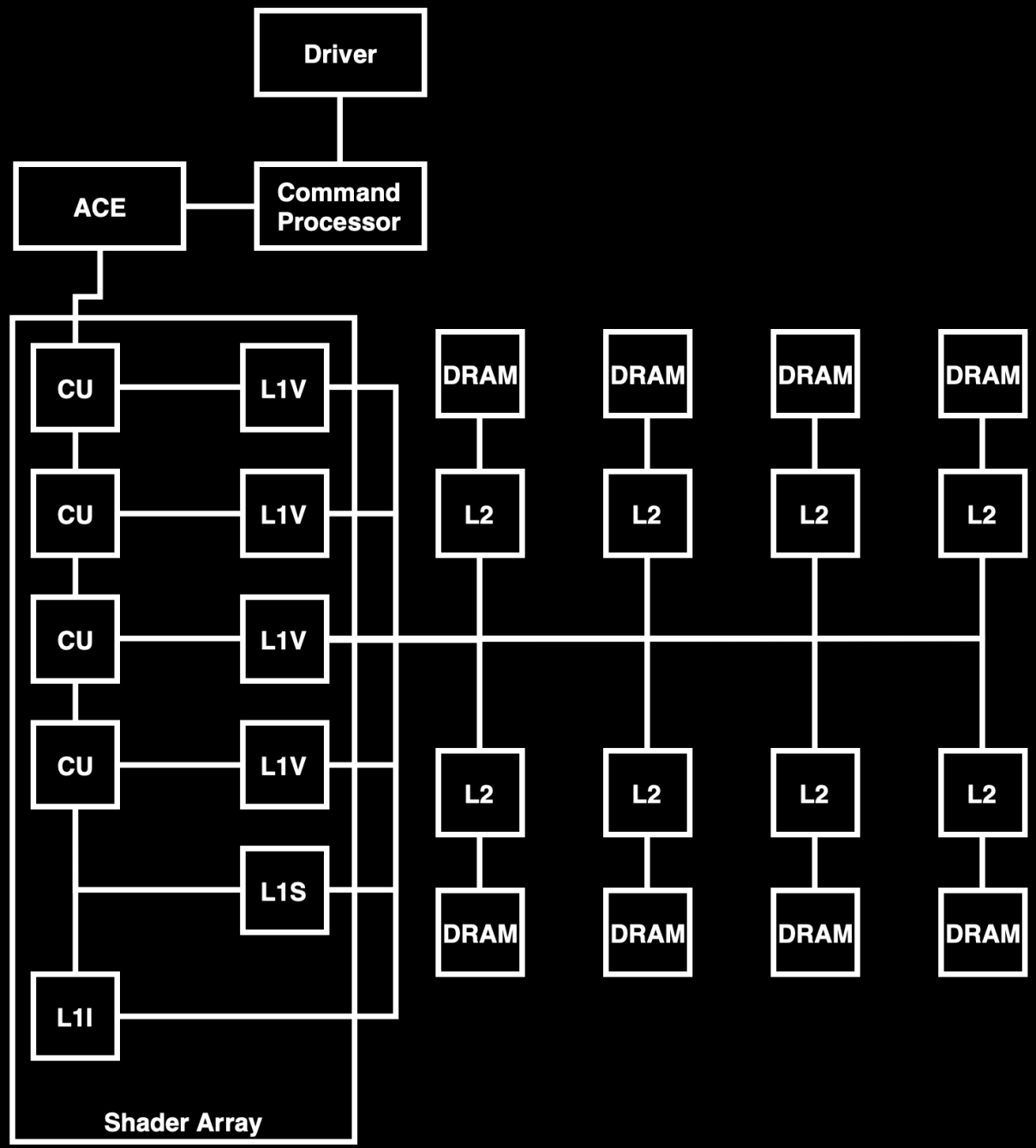




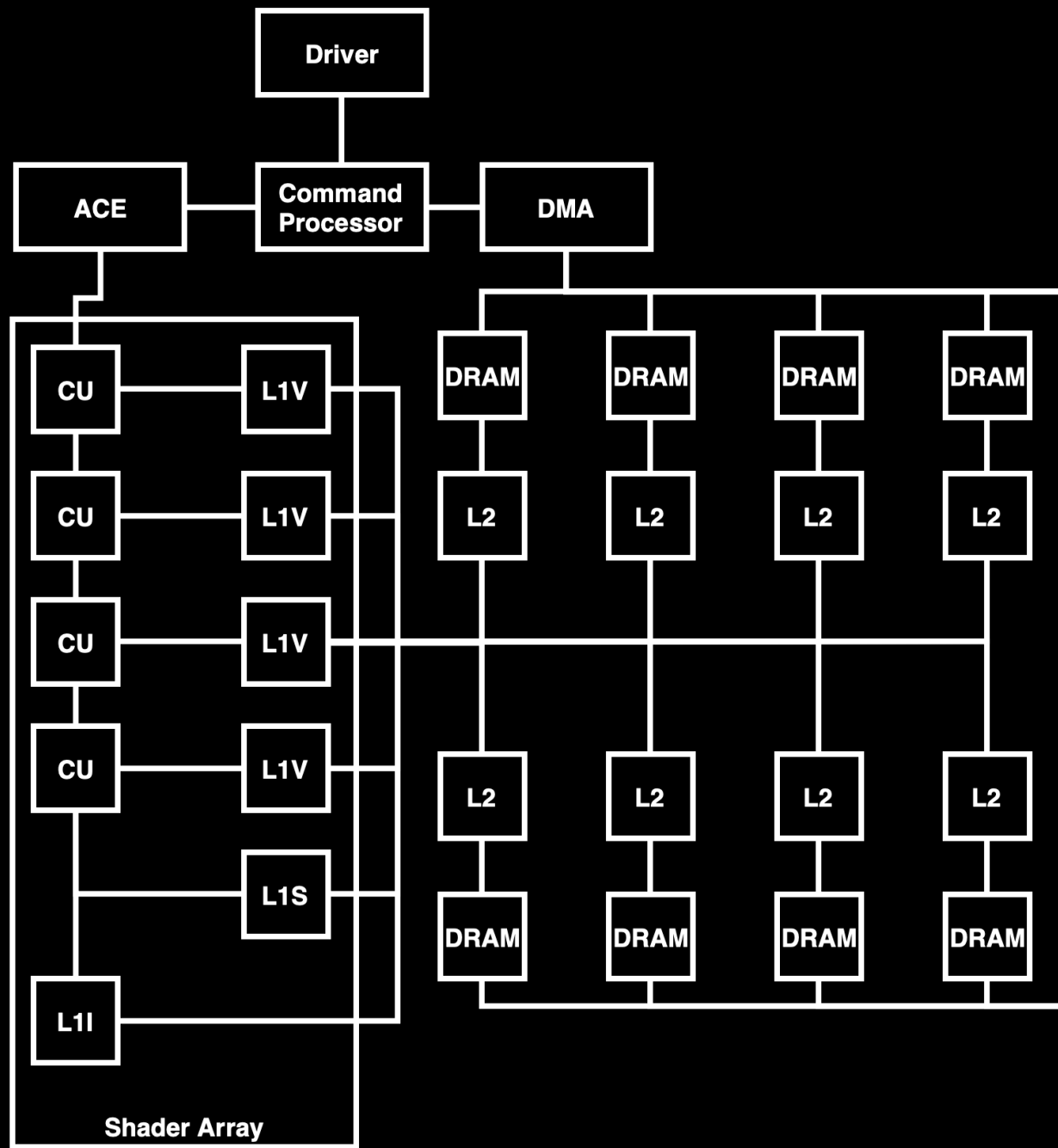
15 more ...



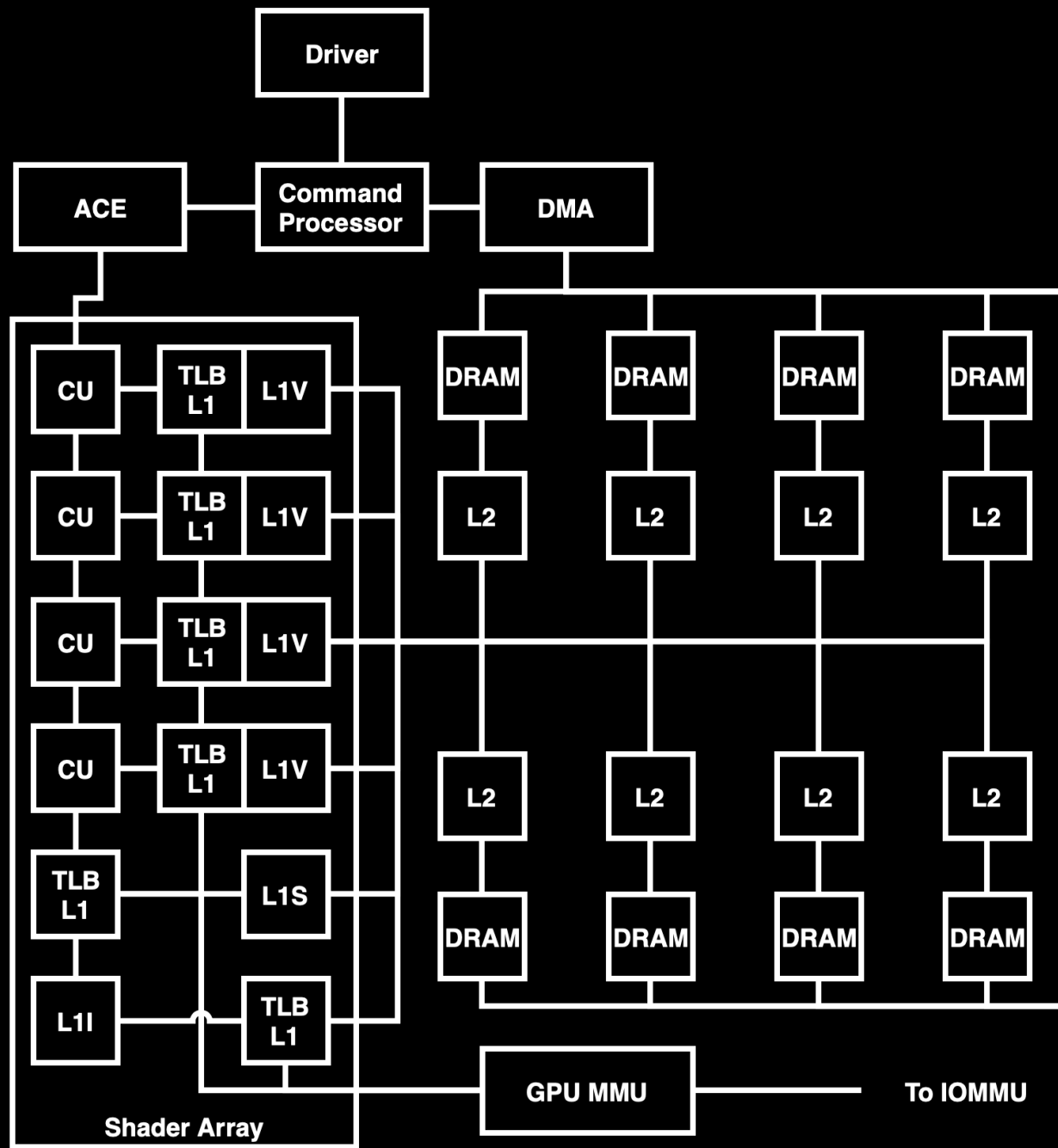
15 more ...



15 more ...



15 more ...

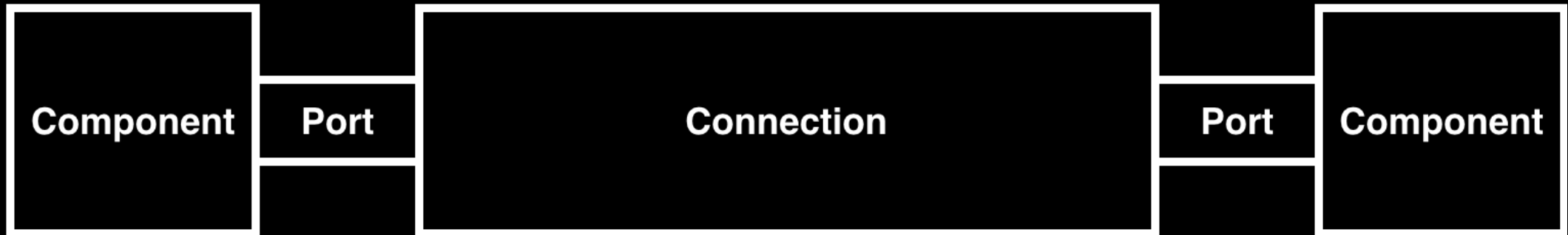


15 more ...

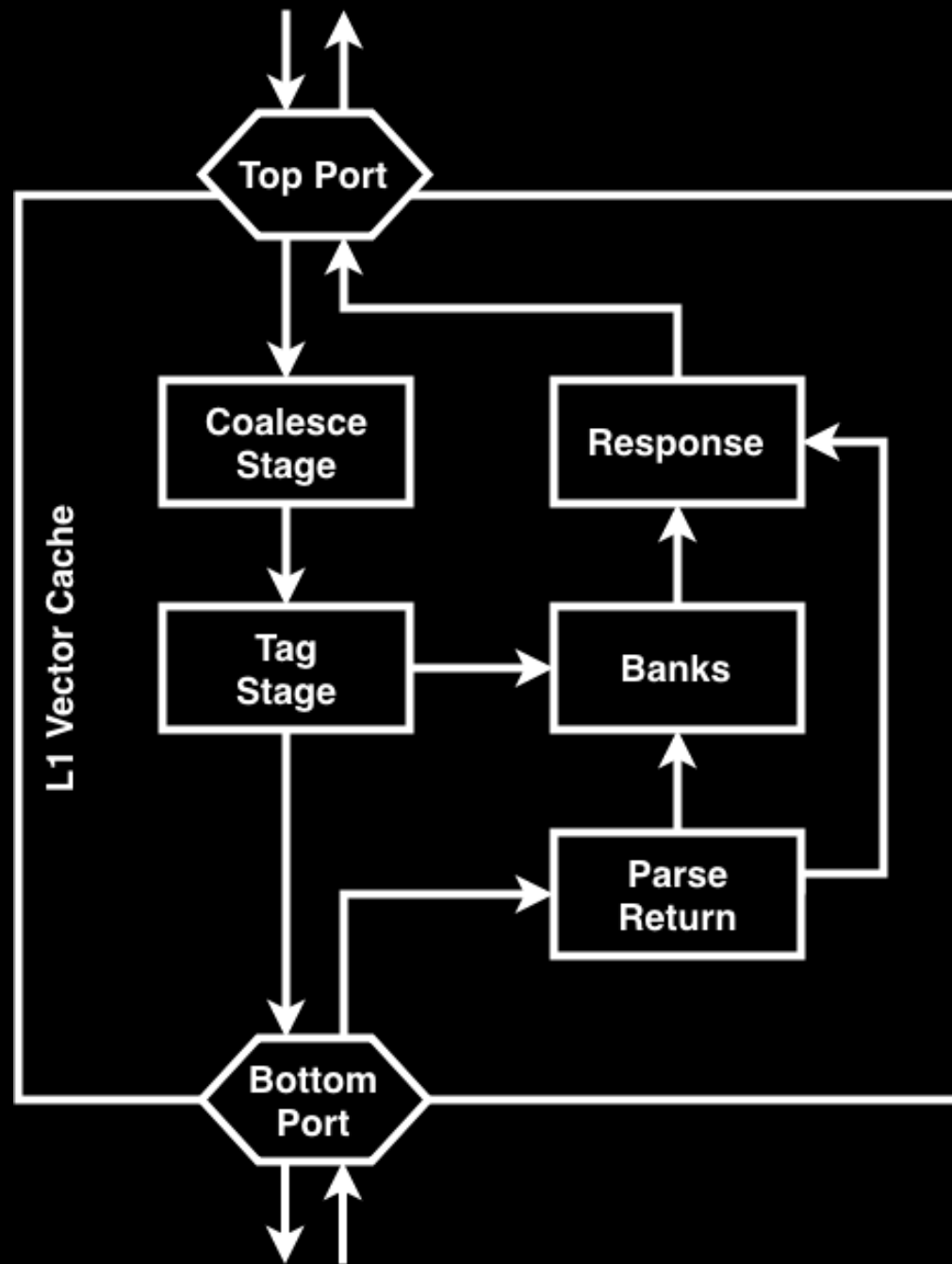
No Magic

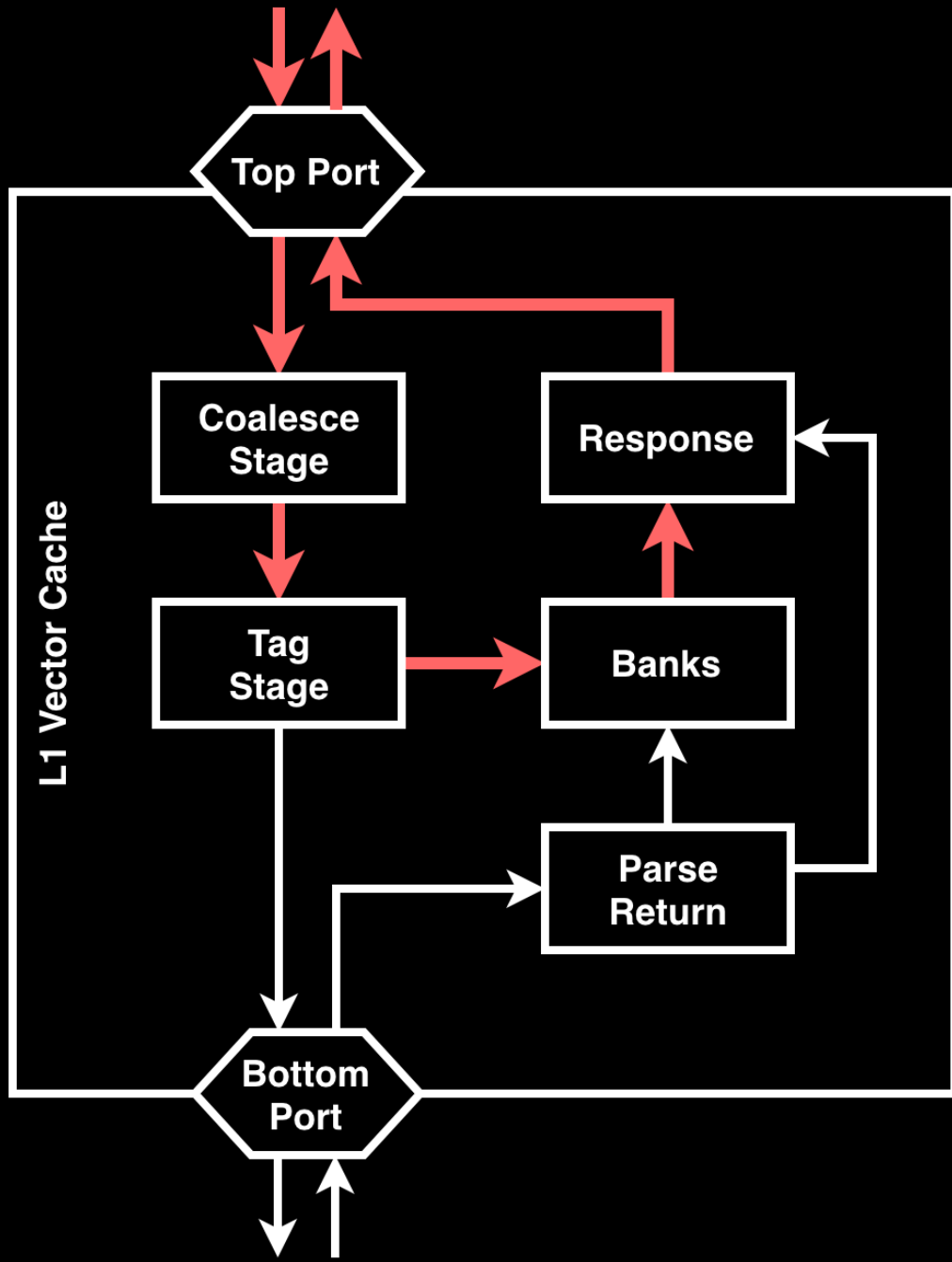
```
if !cache.isBusy {  
    send(req)  
}
```

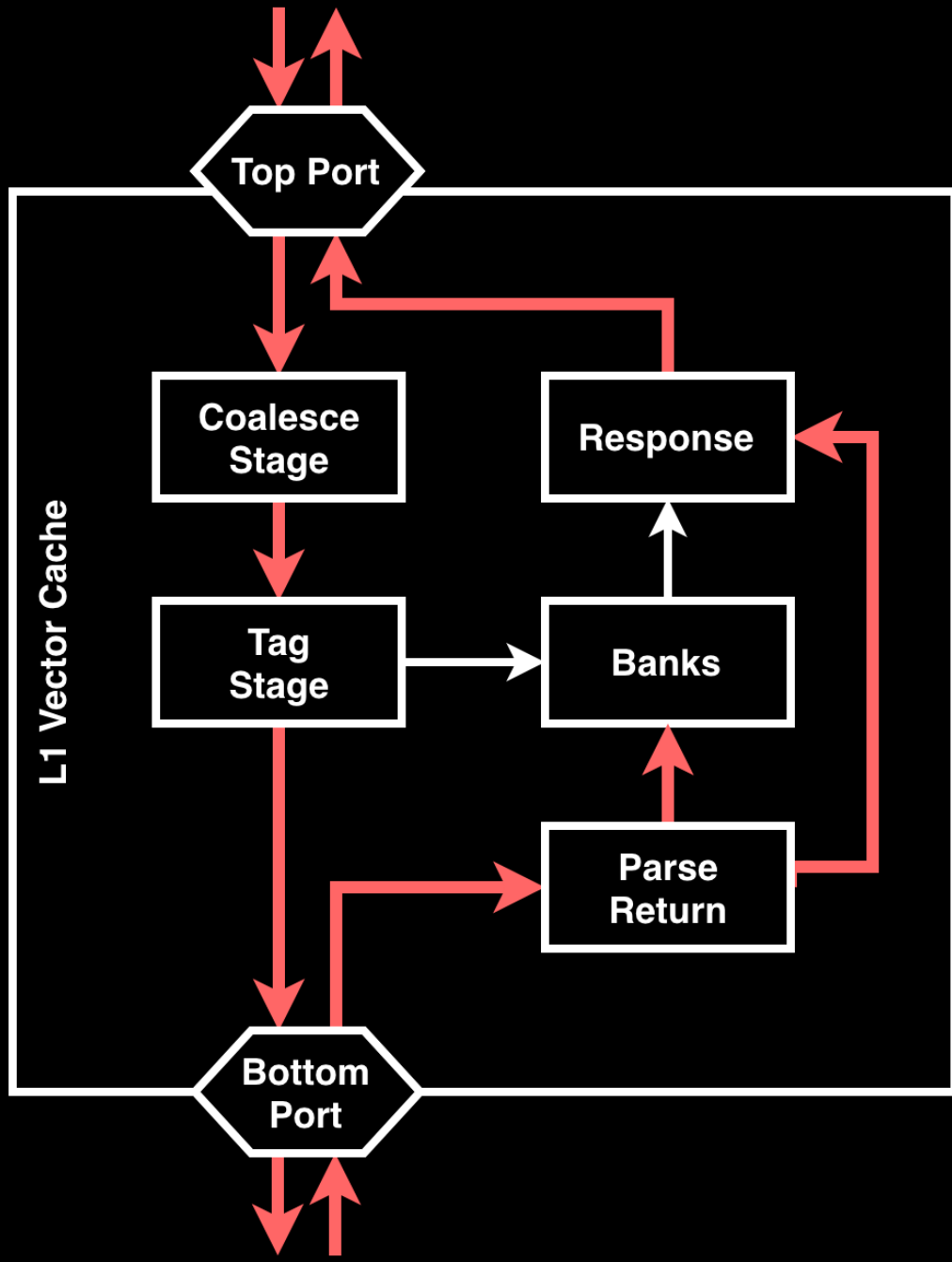
Component



Simulate with Data





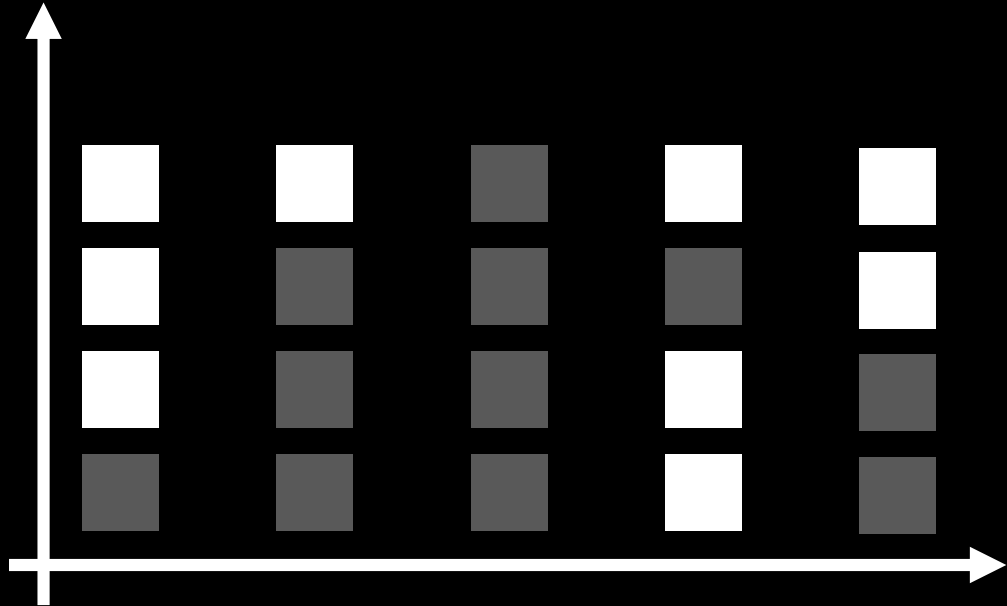


No Busy Ticking

■ Necessary State Update

■ Unnecessary State Update (Busy Ticking)

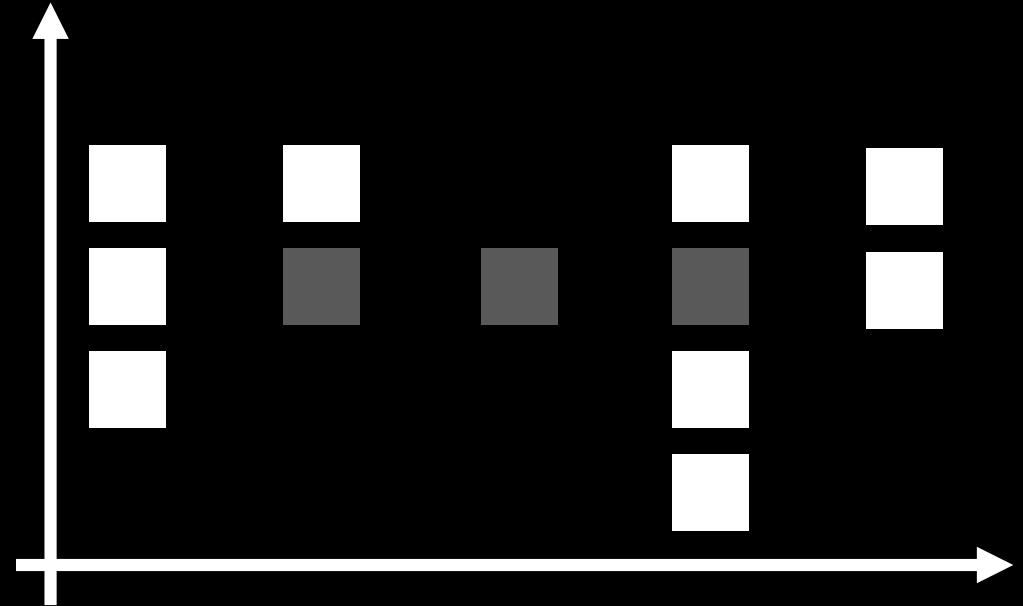
Components



Time

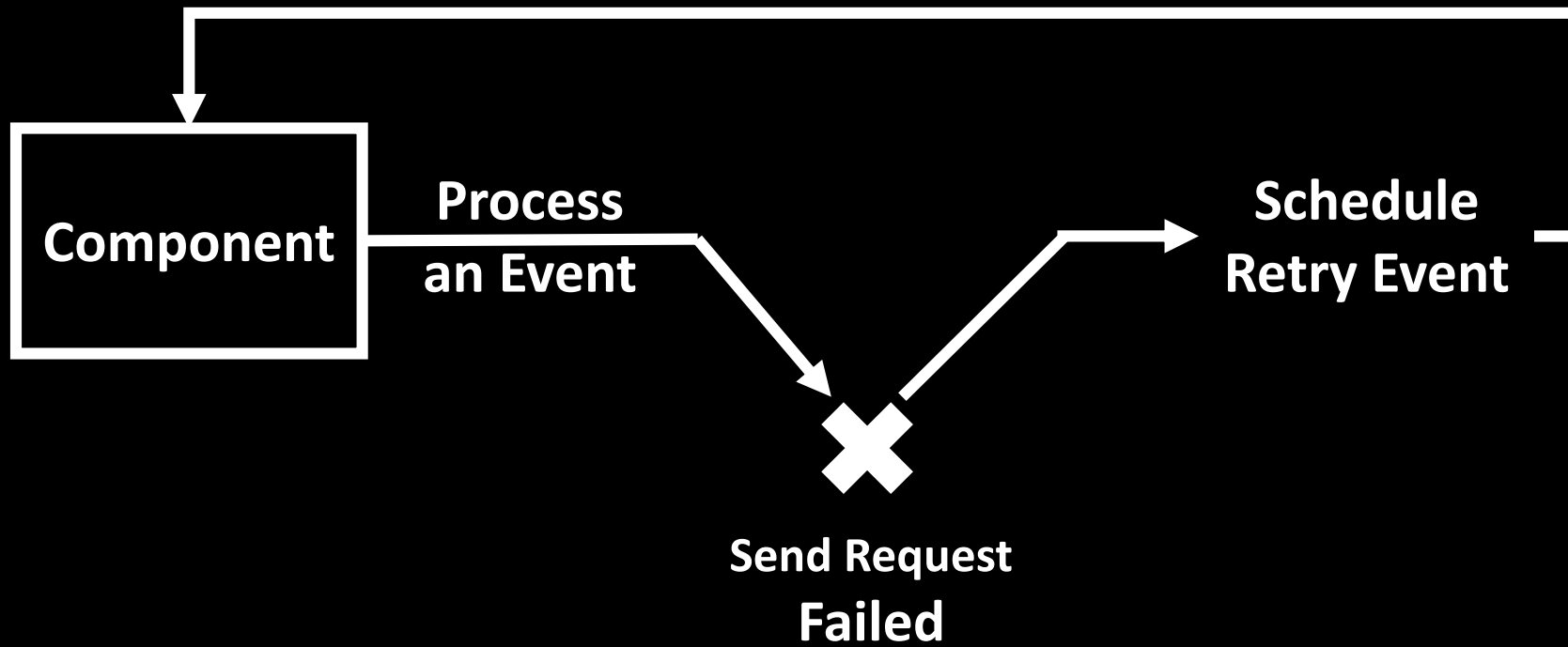
Cycle-Based

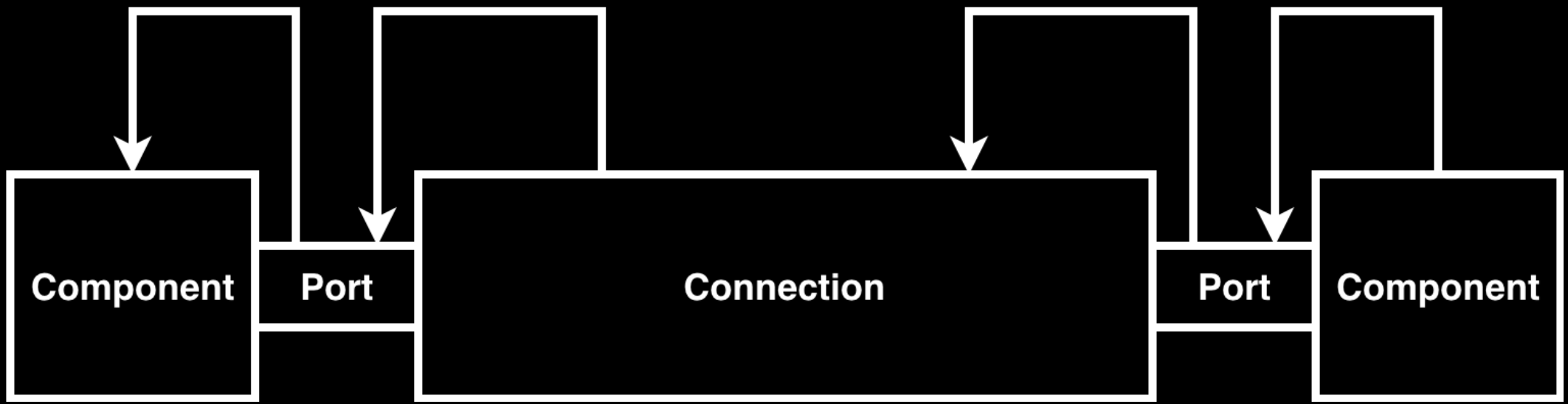
Components



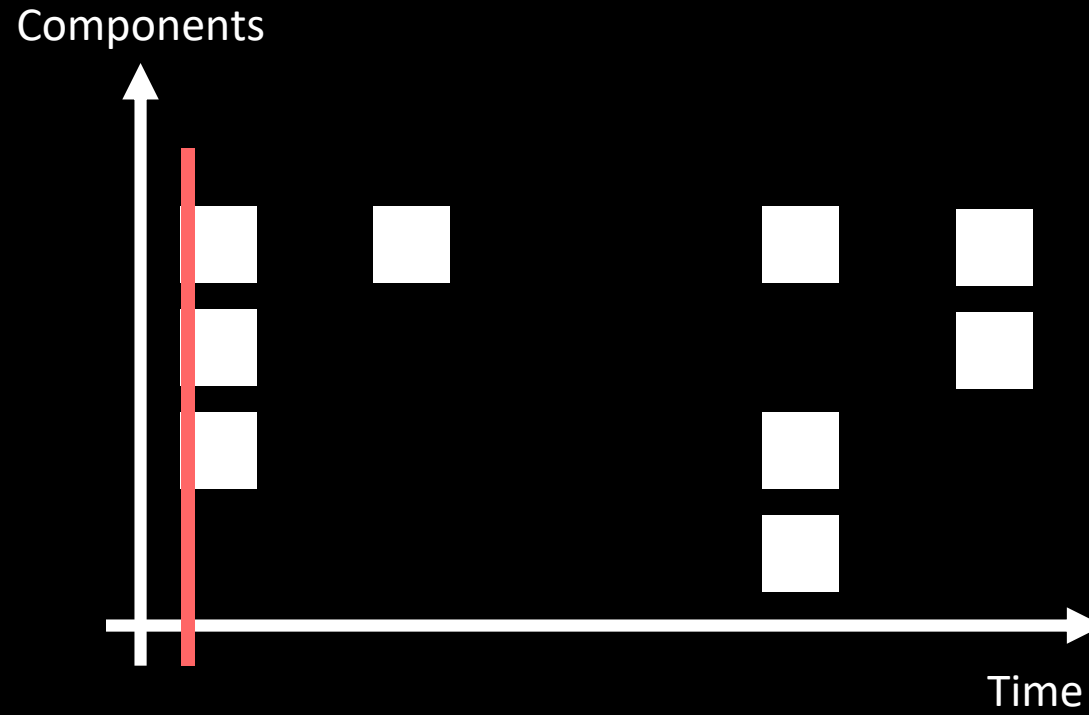
Time

Event-Driven



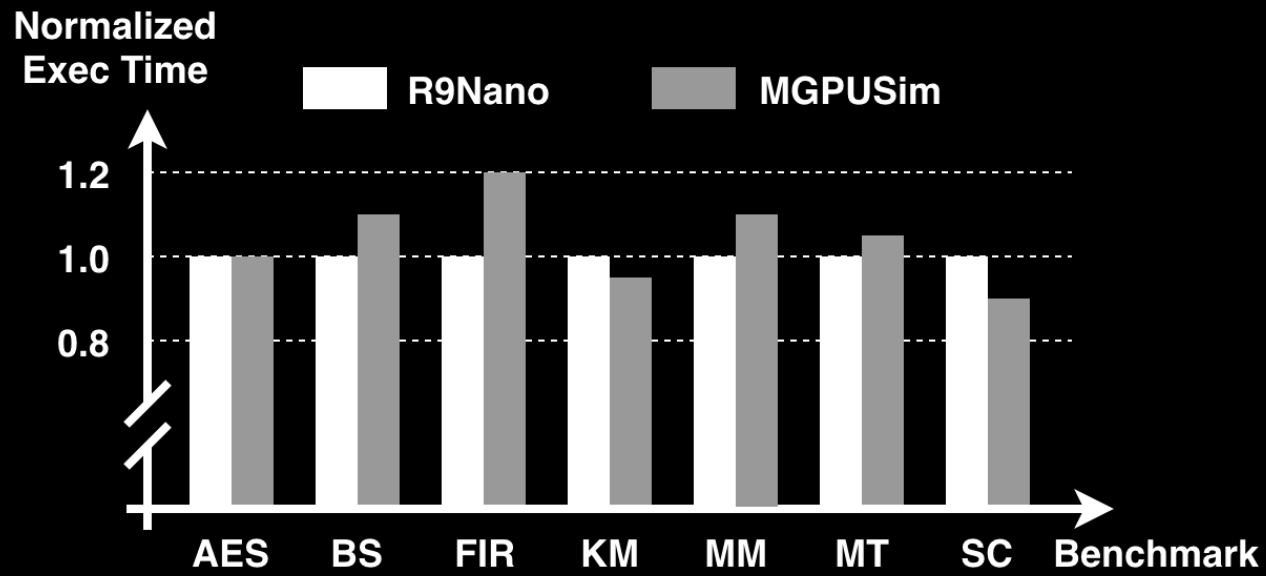


Simple Parallelization

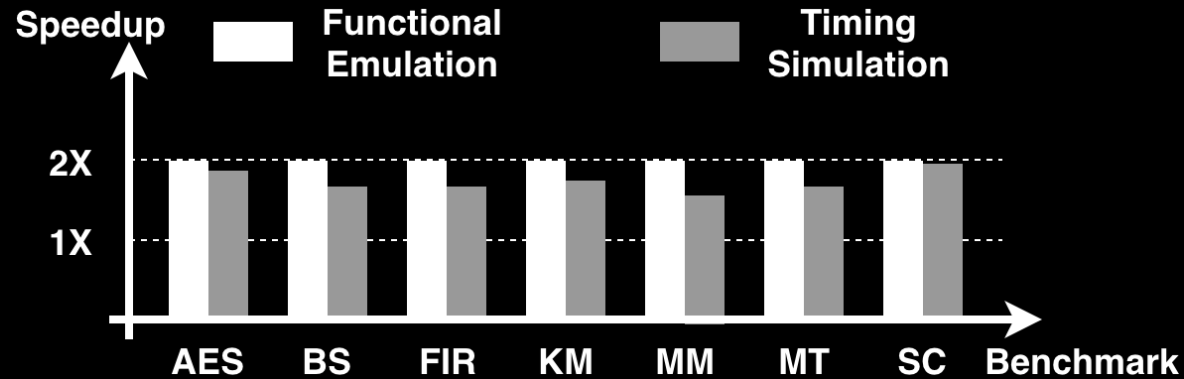


Events that are scheduled at the same do not depend on each other.

```
func (c *Cache) Tick(now akita.VTimeInSec) bool {  
    madeProgress := false  
  
    madeProgress = c.respondStage.Tick(now)    || madeProgress  
    madeProgress = c.parseReturnStage.Tick(now) || madeProgress  
    madeProgress = c.bank.Tick(now)           || madeProgress  
    madeProgress = c.tagStage.Tick(now)       || madeProgress  
    madeProgress = c.coalesceStage.Tick(now)  || madeProgress  
  
    return madeProgress  
}
```



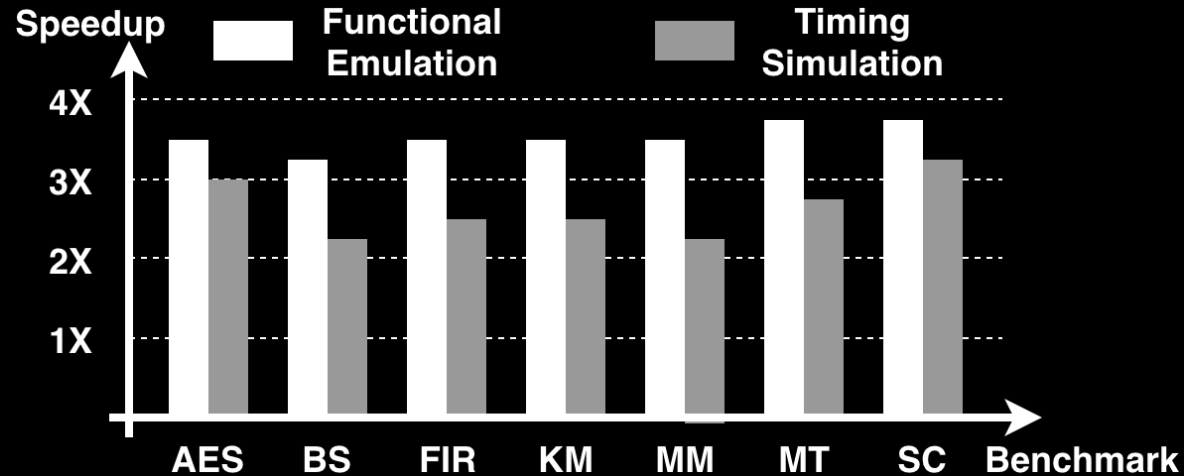
5.5% Average Error



2-Core

1.9X Functional Emulation

1.7X Timing Simulation



4-Core

3.5X Functional Emulation

2.5X Timing Simulation

GPGPUSim

0.8 KIPS

MGPUSim

27 KIPS

34X

Avoid busy ticking
Parallel simulation
Go programming Language

Takeaways

MGPUSim is a serious engineering effort that delivers a high performance, flexible multi-GPU simulator for OpenCL workloads.

Thanks!



AKITA

<https://gitlab.com/akita/akita>

MGPUSim

<https://gitlab.com/akita/gcn3>

Yifan Sun

yifansun@ece.neu.edu

 @_syifan_