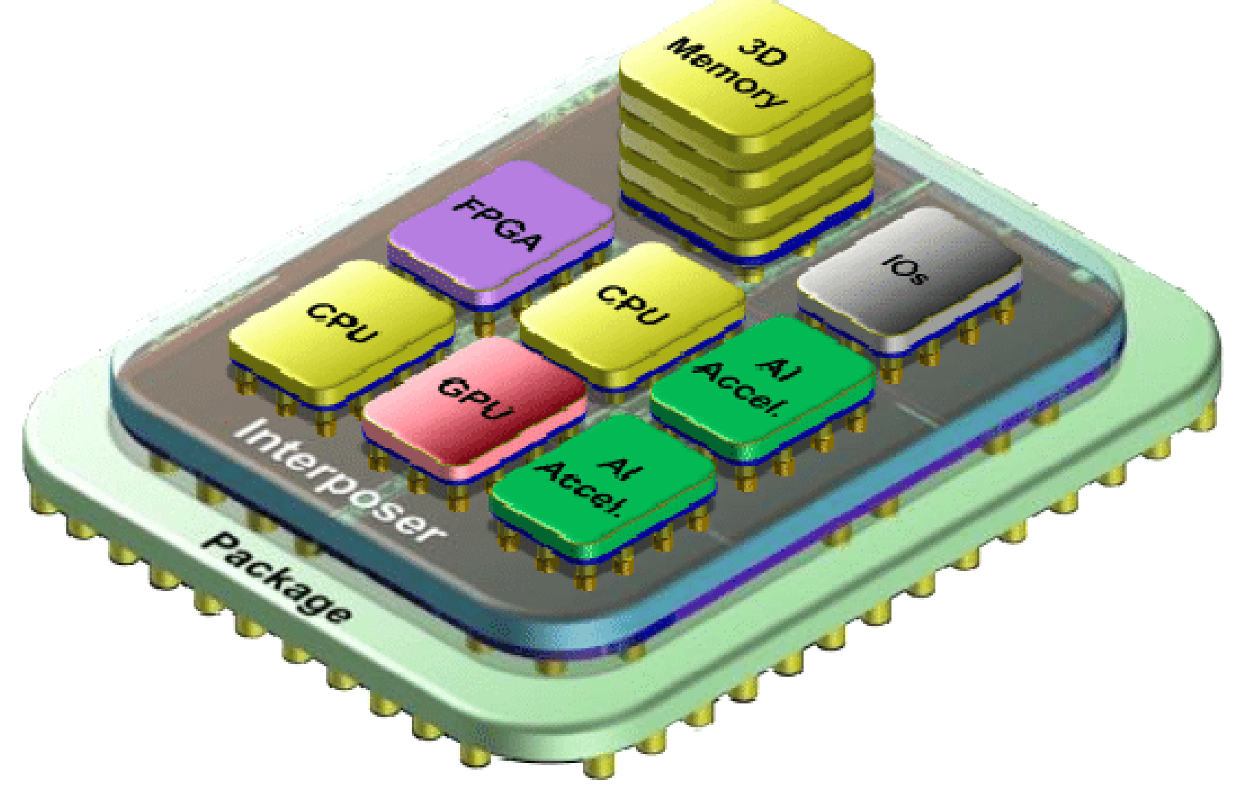
BOSTON UNIVERSITY

Chiplets Design

Jordan Crawford-O'Banner, Long Chen, Shizhe Yang

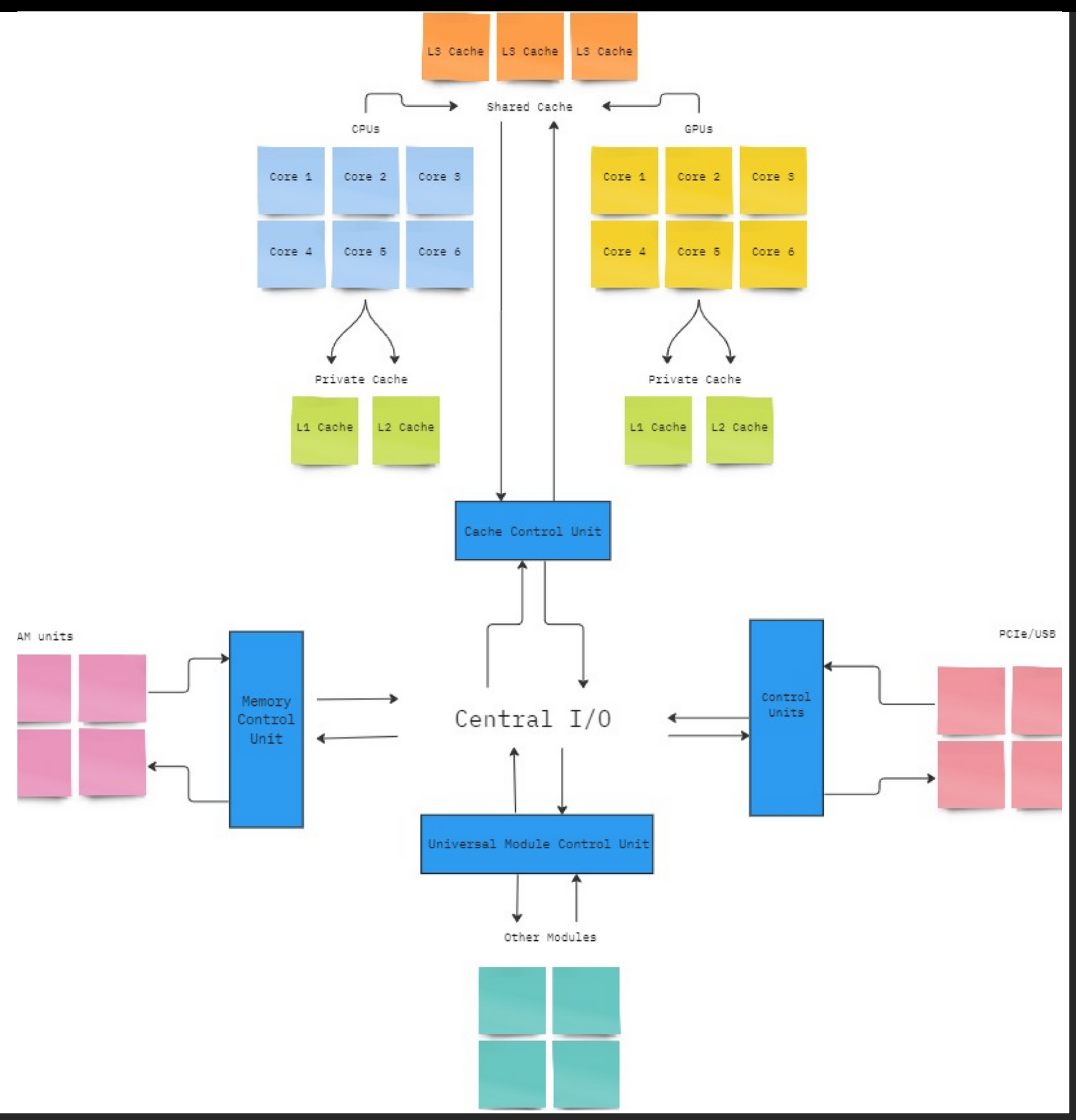
WHAT ARE CHIPLETS?

- Monolithic die divided into modular sub-units
- Multiple dies packaged on a single substrate
- Allows for the possibility of cheaper manufacturing and faster chip design and development



CHIPLET ARCHITECTURE

- Using central/IO to control the data flow and usage of sub units.
- Conventional designs of CPU only have 2 level of caches, but more advanced designs require more caches to optimize performance.
- Architecture to be tested and verified in a simulator.



RESULTS

- Single Core result
- Anticipated to improve performance

```
---------- Begin Simulation Statistics ----------
simSeconds                                  0.356250
simTicks                                 356200000000
finalTick                                356200000000
checkpoints and never reset) (Tick)
simFreq                                 1000000000000
hostSeconds                                    11.52
```

- Multi-core requires third-party peripheral plug-ins
- Cache coherency cause problems
- Workload distribution issue
- Reality reflection:AMD Chiplet also got high latency issues

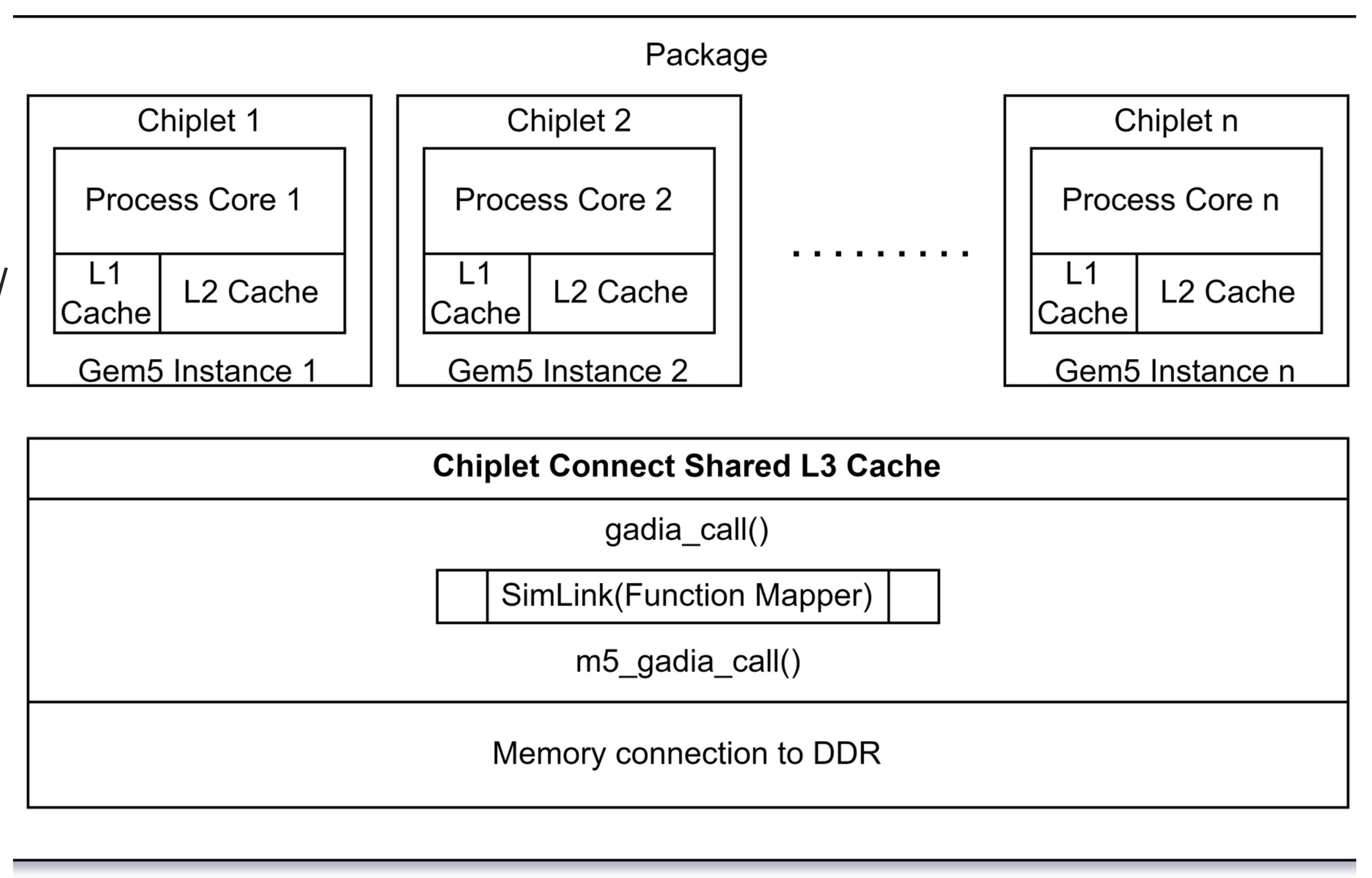
OUR GOALS

- Understand the design concepts of chiplets
- Using simulation tools to test chiplets
- Compare our own chiplet-based architecture to other known chiplet designs.
- Researching on challenges and future developments.
- Working on a potential conference paper.

CHIPLET SIMULATORS

Gem5 instance:
- Processor Cores (X86)
- Conventional L1 / L2 Caches
- Connected with Gem5 Cache

L3 Cache
- Using built-in Cache.py model
- No coherency
- Connected by L3Xbar

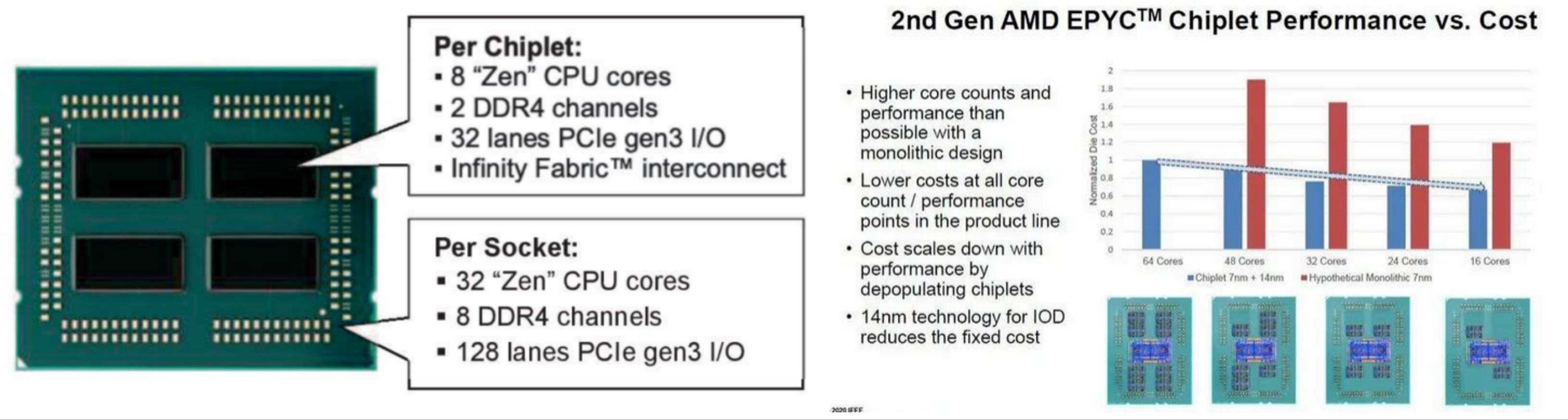


FUTURE

- Create a hardware simulator that is able to test chiplet designs
- Create an interface for users to interact with their own designs
- Reconfigurable interposer
- Real-world: AMD using a 3D stack chiplet for Zen3, physical layer emulations
- Packaging would be a huge challenge for manufacturing purposes.

CURRENT DEVELOPMENT

AMD is one of the most important players in chiplets deisgn design industry.Their famous zen structure extensively utilizes chiplet-based design to produce consumer grade products. Below is a typical commerical consumer-oriented chiplet design.



Per Chiplet:
- 8 "Zen" CPU cores
- 2 DDR4 channels
- 32 lanes PCIe gen3 I/O
- Infinity Fabric™ interconnect

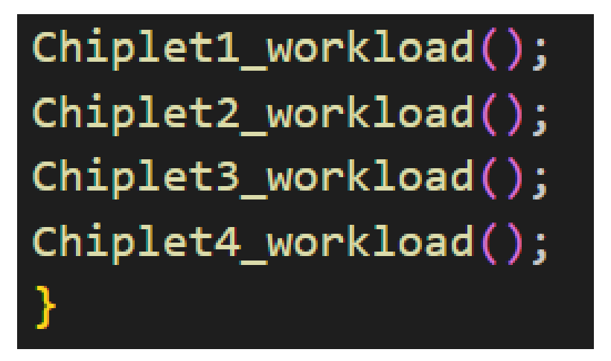
Per Socket:
- 32 "Zen" CPU cores
- 8 DDR4 channels
- 128 lanes PCIe gen3 I/O

2nd Gen AMD EPYC™ Chiplet Performance vs. Cost

- Higher core counts and performance than possible with a monolithic design
- Lower costs at all core count / performance points in the product line
- Cost scales down with performance by depopulating chiplets
- 14nm technology for IOD reduces the fixed cost

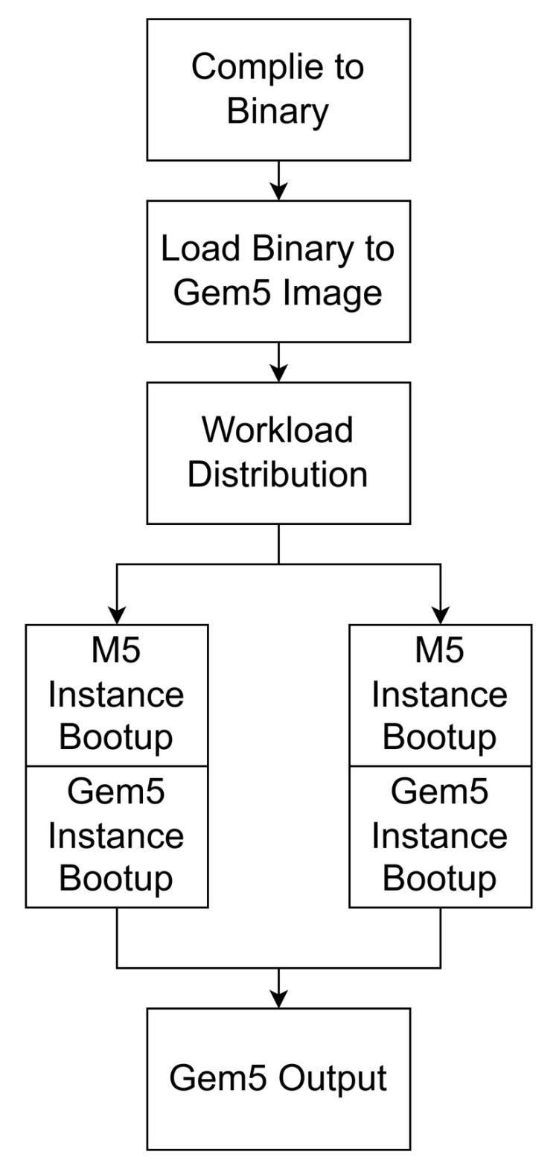
TESTING

- Compile x86 kernel
- Workload Distribute between cores
- Official "two_level.py"
- C Based test program
- Static complied with GCC
- 500 x 500 matrix computation
- Listen for connection on port



```
Chiplet1_workload();
Chiplet2_workload();
Chiplet3_workload();
Chiplet4_workload();
}
```

```
system.l3bus = L3XBar()
system.cpu.icache.connectBus(system.l3bus)
system.cpu.dcache.connectBus(system.l3bus)
### Add the L3 Cache, connect it to the l3bus.
system.l3cache = L3Cache()
system.l3cache.connectCPUSideBus(system.l3bus)
### Add the system cache, connect it to the l3Cache and memory.
system.membus = SystemXBar()
system.l2cache.connectMemSideBus(system.membus)
```



CONCLUSIONS

- Architecture assesment tools still lack integration with this new technology
- Memory control is an important factor in Chiplet analaysis tools
- Trade-off between latency and size cannot be neglected