SAND2014-3090P

Exceptional service in the national interest

Sandia National Laboratories



SST Cassini Component

S.D. Hammond
Scalable Computer Architectures
Sandia National Laboratories, NM

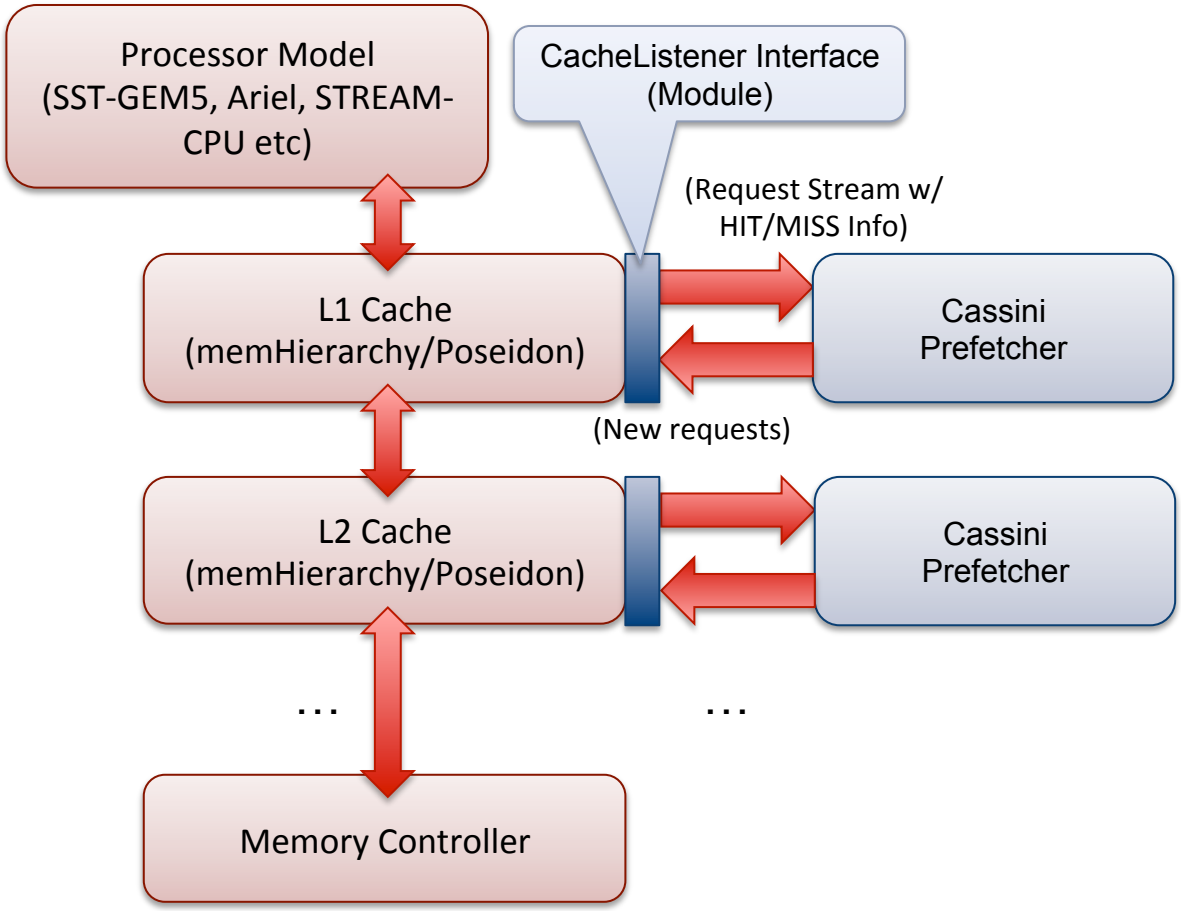
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Cassini Overview

Sandia National Laboratories

- **Purpose:** provide (simple?) configurable "un-core" processor components for use across a wide component portfolio
  - Prefetchers
  - TLB Engines
  - Possibly an "OS" component or memory manager
- Still in development but an important area of simulation as vendors begin to invest more time in uncore design
- Increase sophistication as the simulations/experimental demand rises
- Can be used to record cache activity/trace etc

Cassini Prefetcher Overview



Cassini Prefetcher

Sandia National Laboratories

- Cassini implements two basic prefetching **modules**:
  - **NextBlockPrefetcher** – prefetches the next *k*-cache line on a miss
    - Very simple logic, fast to execute
    - Used in HP RISC processors effectively
    - Downside is that processor has to miss in order to generate a prefetch
  - **StridePrefetcher** – detects stride groups and then prefetches (hopefully) future cache lines
    - Very common in contemporary processors (e.g. Knights Corner L2)
    - Slower to execute as prefetch maintains a window of addresses
    - Advantage is that processor can continue to prefetch ahead easily
- Both prefetcher modules:
  - Halt at a page line boundary
  - Operate entirely in physical address space, no virtual mapping applied

Cassini Prefetcher Discussion

Sandia National Laboratories

- Use Cassini if you want:
  - Lightweight/simple prefetchers
  - Prefetching with memHierarchy or Poseidon caches
  - Prefetching in physical address space
- Don't use Cassini if you want:
  - Aggressive prefetching connected to a processor core (e.g. SST-GEM5)
  - Prefetch logic to use virtual address maps
- Future additions:
  - StreamPrefetcher – detects independent streams (not just strides)
  - IndirectAddressPrefetcher – detect indirect load patterns

Cassini TLB Discussion

Sandia National Laboratories

Not Implemented – In Development

- Currently being developed
  - Simple TLB behavior, provides a mapping operation against a module
  - Responds to commands to flush TLB
  - Support multiple address pools with varying page size per pool
- Design is as a module **not a component**
- Intention to allow TLBs to be added to memory, processor models *etc*

Cassini Virtual OS Component

Sandia National Laboratories

Not Implemented – In Development

- Isolate memory management and some OS functions into a component
- Very early stages of development
- Designed to allow multiple levels of memory to be mapped
  - Allocation policy isolated from components
  - Permits parallel simulation (many processor cores/components)
- Will be a full component with events
  - TLB flush
  - Possibly I/O

Cassini Prefetcher Deck

```
<component name="cpu" type="memHierarchy.streamCPU">
            <params>
                        <workPerCycle> 1000 </workPerCycle>
                        <commFreq> 100 </commFreq>
                        <memSize> 0x1000 </memSize>
                        <do_write> 1 </do_write>
              <num_loadstore> 100000 </num_loadstore>
            </params>
            <link name=cpu_cache_link port=mem_link latency=$lat />
</component>

<component name="l1cache" type="memHierarchy.Cache">
            <params>
                        <num_ways> 4 </num_ways>
                        <num_rows> 16 </num_rows>
                        <blocksize> 64 </blocksize>
                        <prefetcher> cassini.NextBlockPrefetcher </prefetcher>
                        <prefetcher:blocksize> 64 </prefetcher:blocksize>
                        <access_time> 2 ns</access_time>
                        <num_upstream> 1 </num_upstream>
            </params>
            <link name=cpu_cache_link port=upstream0 latency=$lat />
            <link name=cache_bus_link port=snoop_link latency=$buslat />
</component>
```

Assign a CacheListener to a memHierarchy cache

Set the cache line width for the NextBlockPrefetcher

Sandia
National
Laboratories

Exceptional service in the national interest