SAND2008-2312P

TLBs, DRAMs, and Other Scary Things

and Impact of Multi-core

April 8, 2008
Software Engineering Seminar Series

Kevin Pedretti
Scalable System Software, Dept. 1423
ktpedre@sandia.gov

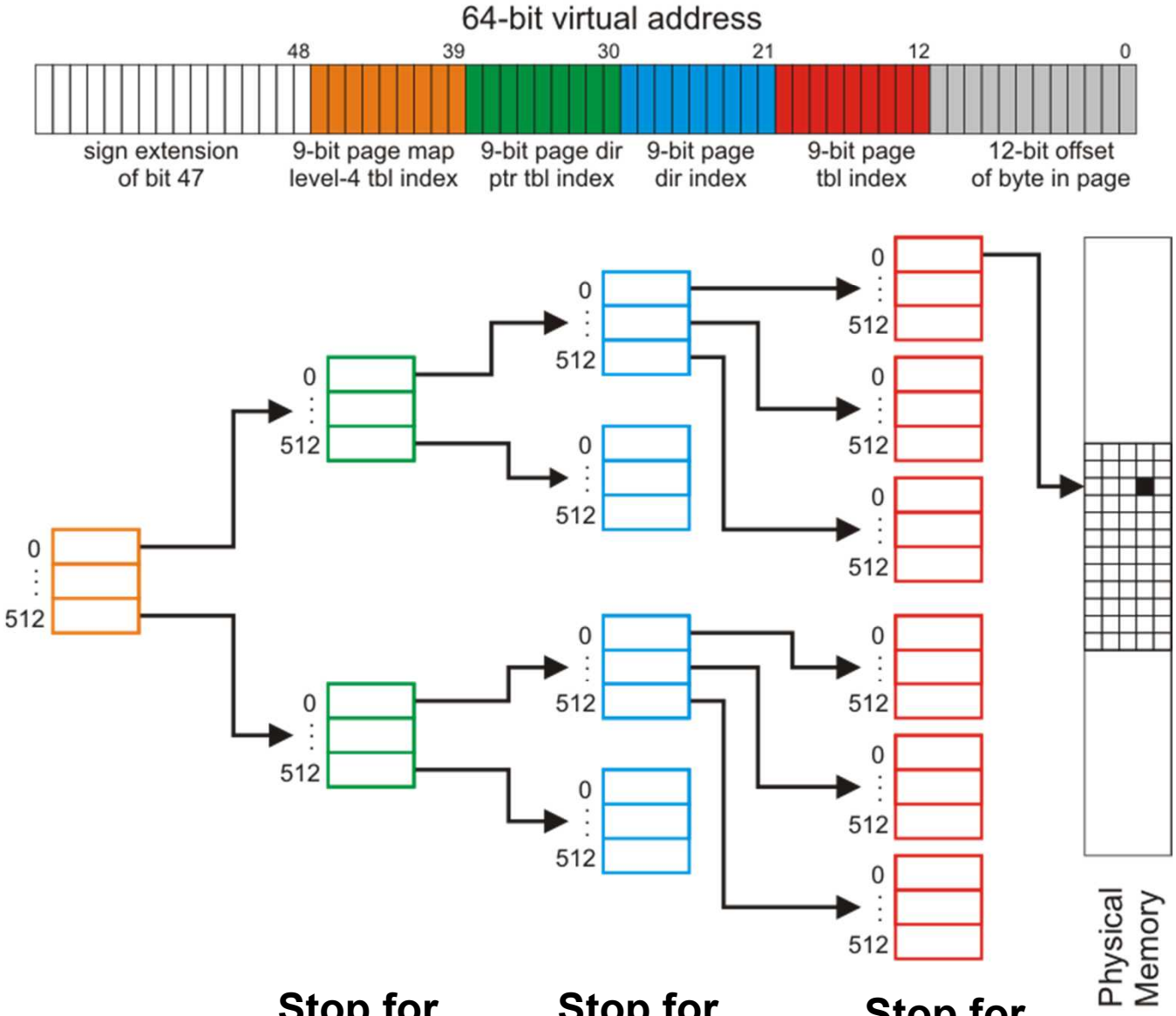
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Translation Lookaside Buffers (TLB)

- TLB purpose is to minimize performance impact of virtual memory
  - Given a virtual address, returns physical address
  - Size limited by speed requirements
- Yet another level of locality
  - Best thought of as a cache, like L1 & L2
  - Hit rates range from 0-100% depending on application

X86-64 Long PAE Page Table Lookup

(52-bit physical space, still 64-bit virtual space per process)

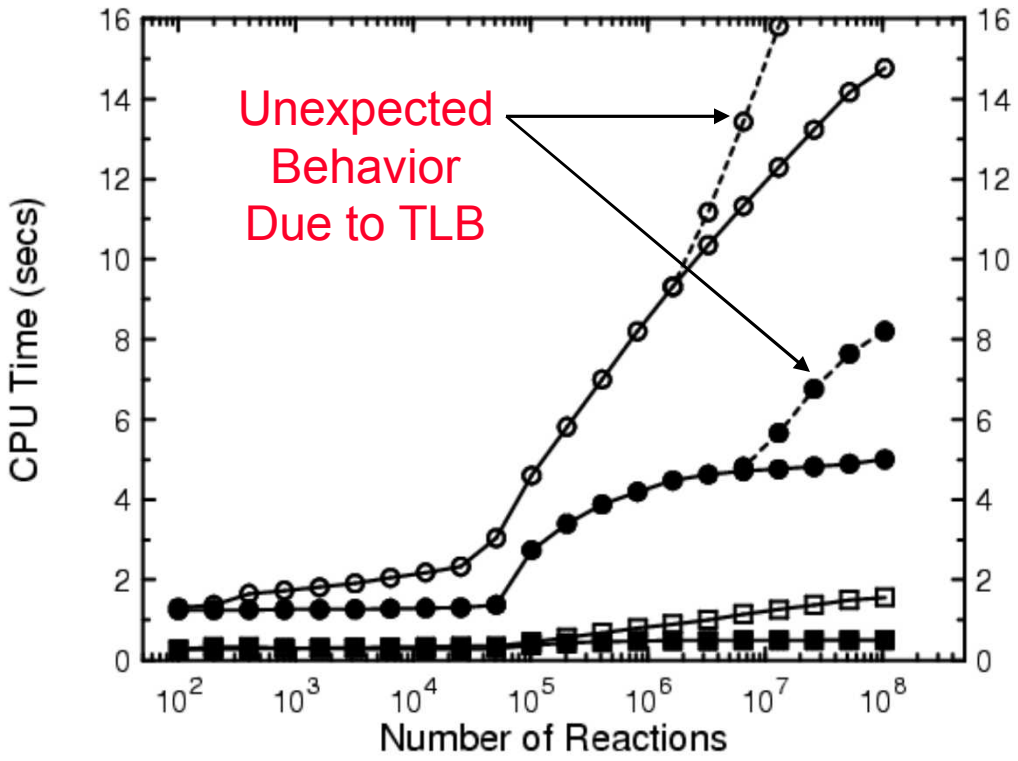


Red Storm Upgrade

- In past, small pages have usually been better
  - `yod -small_pages -sz 8192 app`
  - Not always (next slide)
- Red Storm being partially upgraded to quad-core
- For quad-core, large pages almost always better
  - Need '`-best_page_size`' option?

	Small Page TLB (4 KB)	Large Page TLB (2 MB)	Super Jumbo TLB (1 GB)
Dual-core	32+512	8+0	N/A
Quad-core	48+512	48+128	48+0

TLB Gets in Way of Algorithm Research



Open Shapes =
Existing Logarithmic Algorithm
(Gibson/Bruck)

Solid Shapes =
New Constant-Time Algorithm
(Slepoy, Thompson, Plimpton)

Dashed Line =
Linux with small pages

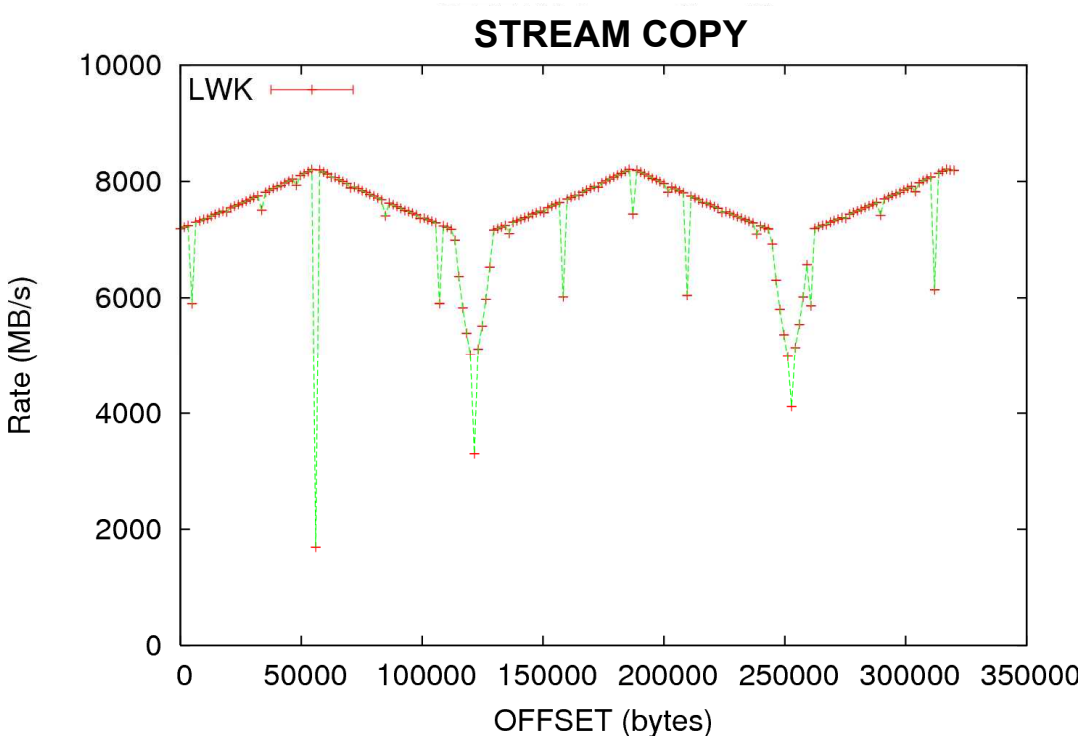
Solid Line
LWK with large pages
(Dual-core Opteron)

**TLB misses increased with large pages,
but time to service miss decreased dramatically (10x).
Page table fits in L1! (vs. 2MB per GB with small pages)**

Sandia National Laboratories

Mysterious STREAM Sawtooth (N=2000000, ~16MB Arrays)

First observed by Courtenay Vaughan running HPCC



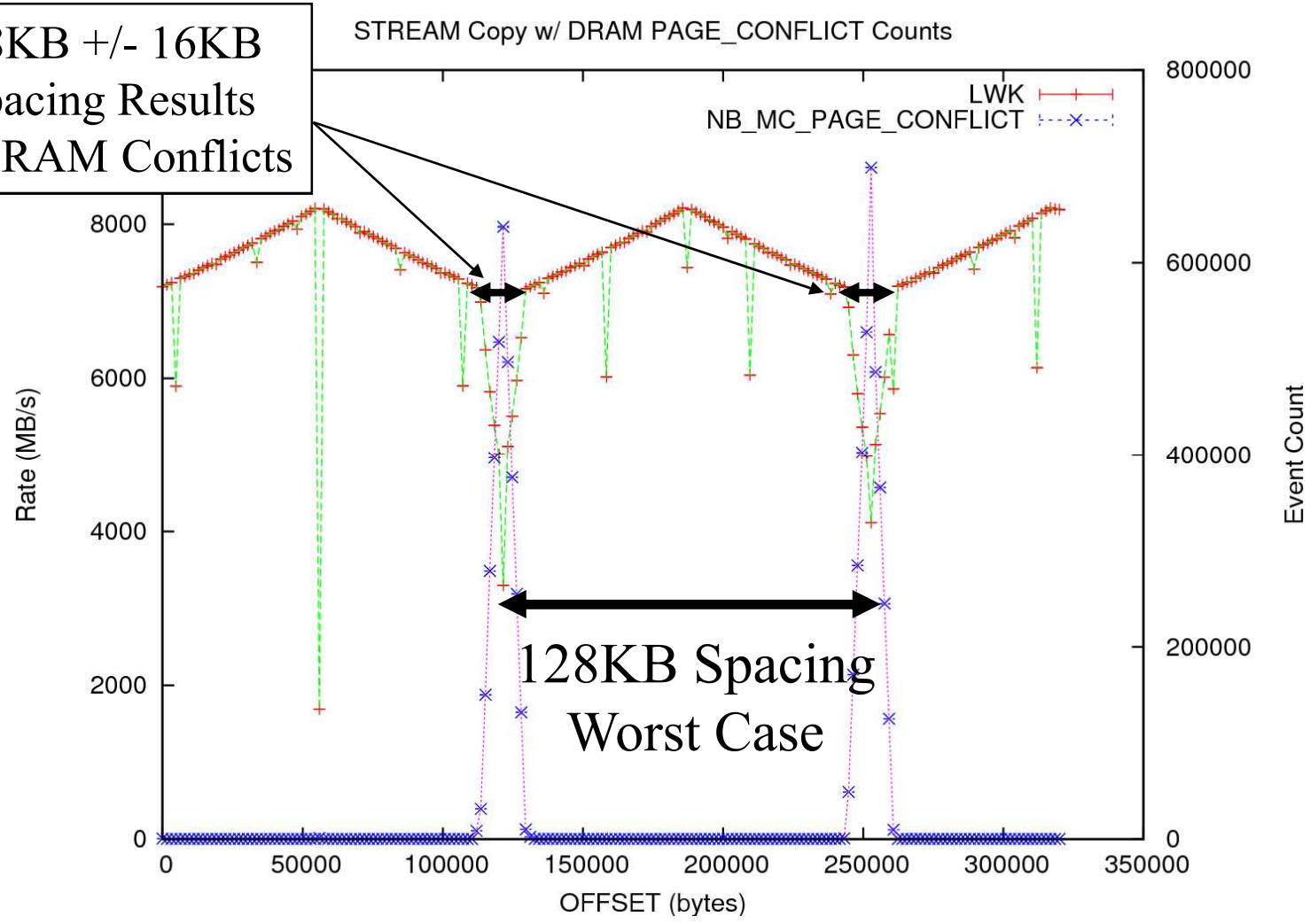
Double arrays
For i = 0 to 2000000 (out of cache)

COPY
a[i] = b[i]
SCALE
a[i] = scalar * b[i]
ADD
a[i] = b[i] + c[i]
TRIAD
a[i] = b[i] + scalar * c[i]

Ron Brightwell, Kurt Ferreira, Kevin Pedretti, Courtenay Vaughan

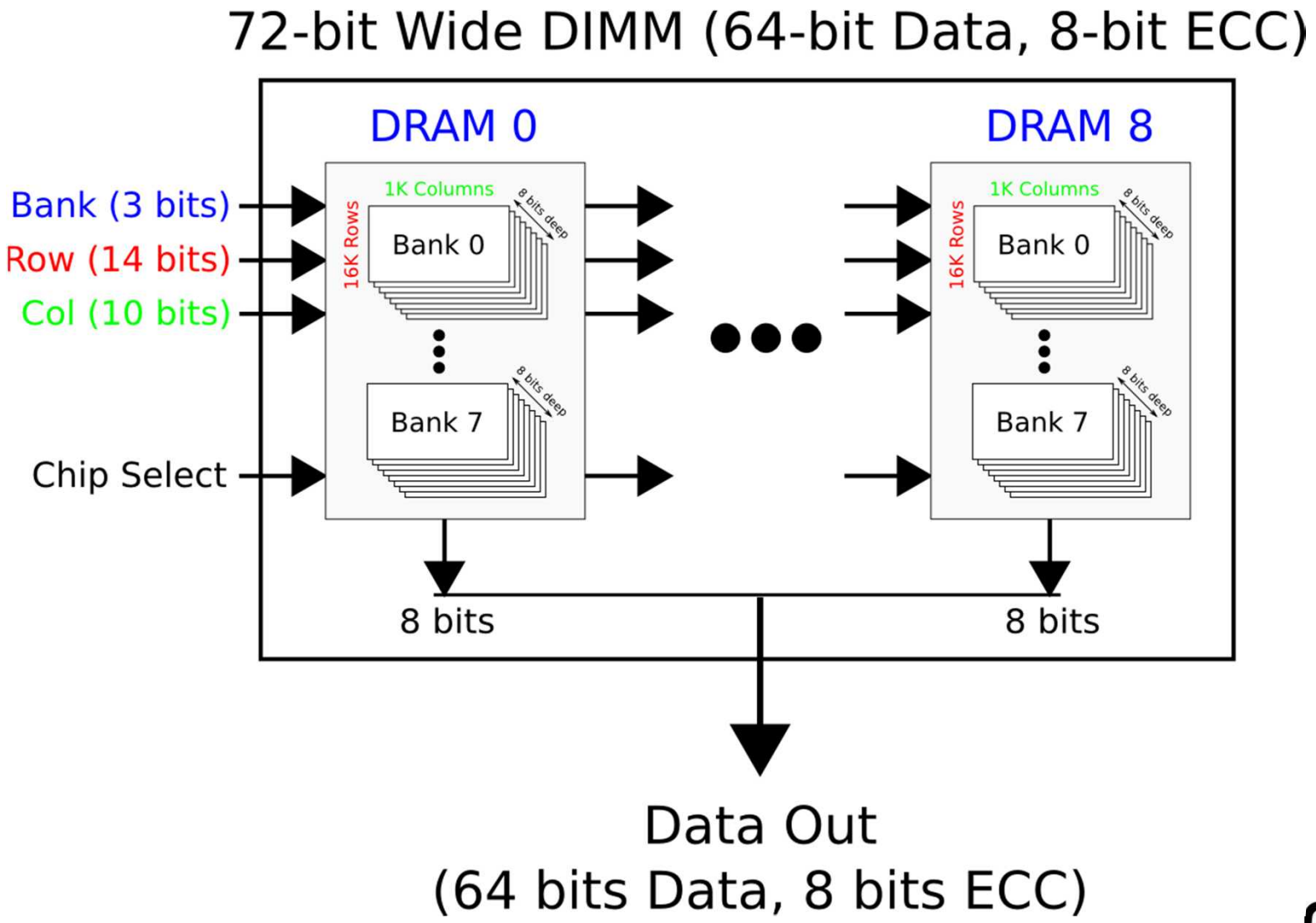
Mystery Solved

128KB +/- 16KB Spacing Results In DRAM Conflicts



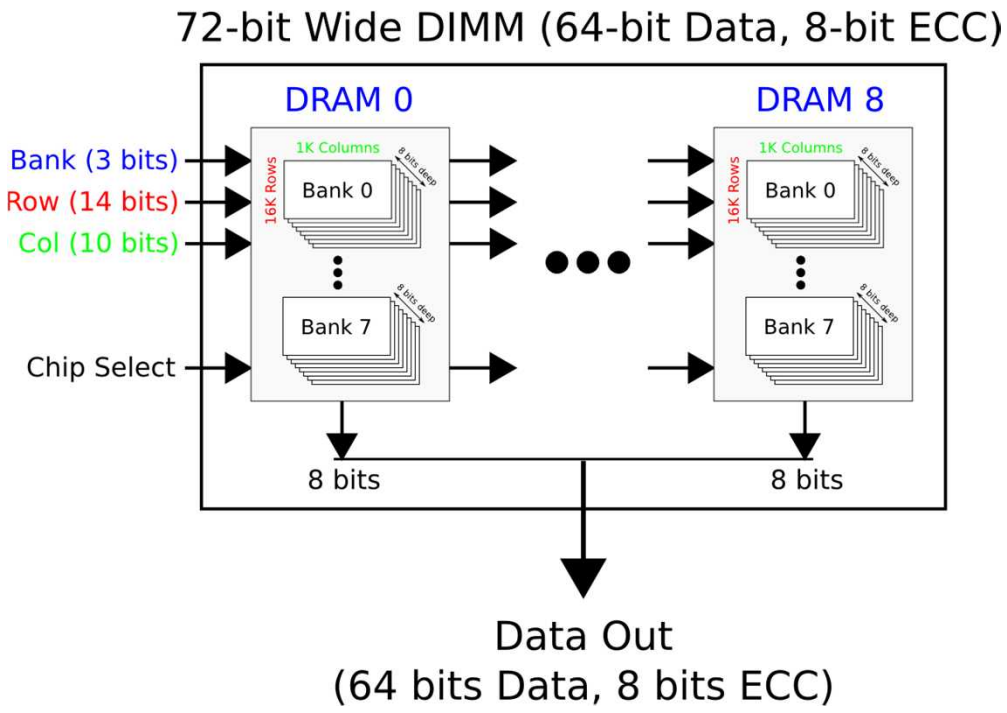
Sandia National Laboratories

DDR2 DIMM Architecture Example



Sandia National Laboratories

DDR2 DIMM Architecture Example



Each DRAM row is
1K columns * 8 bits = 1K bytes

Each DIMM Row is
1K bytes * 8 chips = 8K bytes

Each Memory "Page" is
8K bytes * 2 DIMMs = 16K bytes

Addresses that are
16K bytes * 8 banks = 128K bytes
apart will result in a Bank Conflict
(Simultaneous accesses to different rows in same bank, aka Row Conflict)

Sandia National Laboratories

Impact of Multi-core

- Main question: Does this matter for any apps?
- Lots of cores accessing same DRAM bank haphazardly decreases locality
  - HW memory controller can do better scheduling
  - OS can do something smarter
  - Application can do something smarter
    - Blocking, bringing in large unit-stride chunks into cache
    - Stream programming models