Performance Evaluation: Intel Nehalem and AMD Shanghai Cache Architectures

Sai Dheeraj Polagani

Department of Electrical Engineering, San Jose State university, San Jose, California 95192.

Introduction

Chip-Multicore technology has now become a vital force driving the performance improvements in today's microprocessors. As the processing units gets faster, the Memory sub-system has been found out to be a bottle-neck to utilize the complete performance boost provided by the microprocessor cores. The introduction of the on-chip cache memory into the memory sub-system has provided a way to boost up the memory sub-system performance. In a multi-core scenario, cache architecture becomes crucial as the cores tries to share the available memory on the chip. This project deals with performance evaluation of two such cache architectures at RTL level. The project tries to come up with two RTL level designs of Cache Controllers, based on Intel Nehalem and AMD Opteron (Shanghai) Cache architectures. The paper picks up the cache architecture in a quad-core environment; the micro-processors being targeted for high performance Desktops. The Design includes two level private caches for each processor and a shared last level cache, with cache coherency maintained.

Cache Performance metrics, viz., Cache Hit / Cache Miss, Latencies and Run-Time would be compared for both the designs against a address space trace from a sample C application program.

Methodology

Cache Architecture Specifications

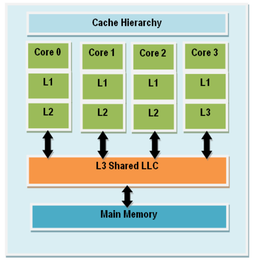
Intel Nehalem:

Processor: 4x INTEL Core i5
Architecture: x86-64
No. of Cores: 4
Codename: Nehalem-EP
Cache Line: 64Bytes
L1 Cache: 32KB/core, 8-Way Set Associative, non-inclusive, private to specific core.
L2 Cache: 256KB/core, 8-Way Set Associative, non-inclusive, private to specific core.
L3 cache: 8MB LLC, 16-Way Set Associative, inclusive of L1 and L2, shared among 4 cores.
Cache Coherency Protocol: MESIF
Cache Replacement Policy: pseudo LRU
Cache Write Policy: Write-Back Write-Allocate
On Chip Interconnects: Point-Point for Cache Coherency maintenance. Token Pass Ring for shared write-ports.

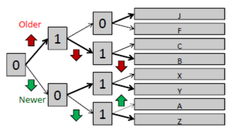
Methodology

AMD Opteron (Shanghai):

Processor: 4x AMD Opteron
Architecture: x86-64
No. of Cores: 4
Codename: Shanghai
Cache Line: 64Bytes
L1 Cache: 64KB/core, 2-Way Set Associative, non-inclusive, private to specific core.
L2 Cache: 512KB/core, 16-Way Set Associative, exclusive of L1, private to specific core.
L3 cache: 6MB LLC, 48-Way Set Associative, exclusive of L1 and L2, shared among 4 cores.
Cache Coherency Protocol: MOESI
Cache Replacement Policy: pseudo LRU
Cache Write Policy: Write-Back Write-Allocate
On Chip Interconnects: Point-Point for Cache Coherency maintenance. Token Pass Ring for shared write-ports.



Pseudo- Lease Recently Used (pLRU)



State	Replace
xxx0x00	J
xxx1x00	F
xx0xx10	C
xx1xx10	B
x0xx0x1	X
x1xx0x1	Y
0xxx1x1	A
1xxx1x1	Z

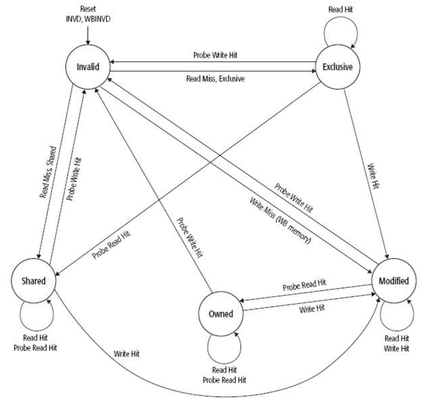
'x' mean don't care

Next State	Ref To
_ _ _ 1 _ 1 1	J
_ _ _ 0 _ 1 1	F
_ _ 1 _ _ 0 1	C
_ _ 0 _ _ 0 1	B
_ 1 _ _ 1 _ 0	X
_ 0 _ _ 1 _ 0	Y
1 _ _ _ 0 _ 0	A
0 _ _ _ 0 _ 0	Z

'_' means unchanged

MOESI/ MESIF Cache Coherency Protocol

(M) Modified: Cache Block written/updated by the core.
(O) Owned: Cache Block available for sharing among all shared cache lines.
(E) Exclusive: The core has the only copy of the Cache Block and is clean.
(S) Shared: Cache Block is shared with different cores.
(F) Forward: Cache Block to be forwarded to any other requesting core for sharing.
(I) Invalid: Cache Block is invalid and can be utilized for new set of data.



Cache Block possible states at any given time for MESIF and MOESI Protocol.

	M	O	E	S	I
M	✗	✗	✗	✗	✓
O	✗	✗	✗	✓	✓
E	✗	✗	✗	✗	✓
S	✗	✓	✗	✓	✓
I	✓	✓	✓	✓	✓

	M	E	S	I	F
M	✗	✗	✗	✓	✗
E	✗	✗	✗	✓	✗
S	✗	✗	✓	✓	✓
I	✓	✓	✓	✓	✓
F	✗	✗	✓	✓	✗

Results

AMD Shanghai Cache Performance Results:
Total Memory Access of (Core 0) = 1861782
Total Cache Hit = 1532936
Cache Hit Rate = Cache Hit / Total Memory Access
= 1532936/1861782 = 0.82 ~ 82 %
Total Memory Access of (Core 1) = 1890877
Total Cache Hit = 1562167
Cache Hit Rate = Cache Hit / Total Memory Access
= 1562167/1890877 = 0.82 ~ 82 %

Results

Cache Miss Rate = 18 %
Cache Hit Time = 2 Cycles
Cache Miss Penalty = 31 Cycles
Cache Consistency Cycles = 21 Cycles
Total Run Time = 12350912 Cycles

Intel Nehalem Cache Performance Results:
Total Memory Access of (Core 0) = 1867270
Total Cache Hit = 1546860
Cache Hit Rate = Cache Hit / Total Memory Access
= 1546860/1867270 = 0.83 ~ 83 %
Cache Miss Rate = 17 %
Total Memory Access of (Core 1) = 1890877
Total Cache Hit = 1571610
Cache Hit Rate = Cache Hit / Total Memory Access
= 1571610/1890877 = 0.83 ~ 83 %
Cache Miss Rate = 17 %
Cache Hit Time = 2 Cycles
Cache Miss Penalty = 31 Cycles
Cache Consistency Cycles = 19 Cycles
Total Run Time = 10364041 Cycles

Summary

Intel Nehalem and AMD Shanghai, though they differ in terms of cache sizes, associativity, cache coherence protocols, cache organization, etc., they had the same hit/miss ratio for the given application program. Intel though has a slight edge at 83%, over AMD architecture. However, if we examine the run-time, Intel clearly has a better result over the AMD architecture. AMD architecture increase in run-time is also due to its exclusive nature as the L3 has to be evicted often. Further the AMD architecture due to its non inclusive nature has to use a lot of decoders and additional hardware to handle cache coherency.

Key References

[1] Multi-Core Cache Hierarchies, Rajeev Balasubraman Norman P. Jouppi and Naveen Muralimanohar, 2011.
[2] Cache Organization of the Intel Nehalem Computer Architecture, Trent Rolf, University of Utah, Dec. 09.
[3] Intel® 64 Architectures Software Developer's Manual. February 2014.
[4] Architecture of the AMD Quad Core CPUs, Brian Waldecker, AMD, Austin. April 13, 2009.

Acknowledgements

Prof. Morris Jones, for his extensive help throughout the project. His teachings were very valuable to understand the cache and come up with an optimal design.