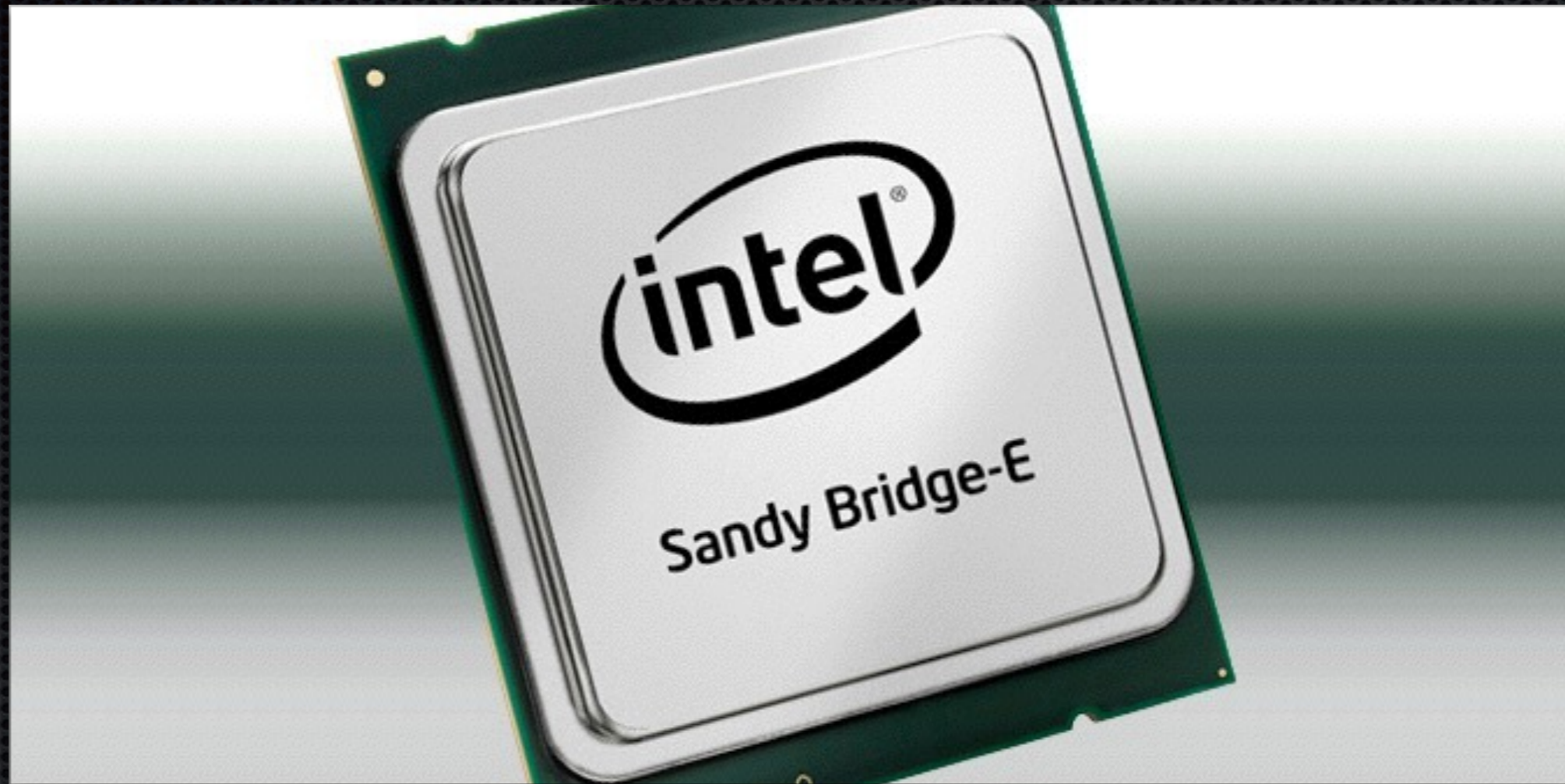


Intel Core i7-3960X Memory Cache Control



Corso di Architetture e Progetto di Sistemi e Servizi
Informatici

Anno Accademico 2011-2012

Stefano Cicero



45nm Process Technology

32nm Process Technology

22nm Process Technology

Penryn
Intel® Core™
Microarchitecture

Nehalem
Intel® Core™
Microarchitecture

Westmere
Intel® Core™
Microarchitecture
(Nehalem)

Sandy Bridge
Intel® Core™
Microarchitecture

Ivy Bridge
Intel® Core™
Microarchitecture
(Sandy Bridge)
FUTURE PLATFORM

TICK

TOCK

TICK

TOCK

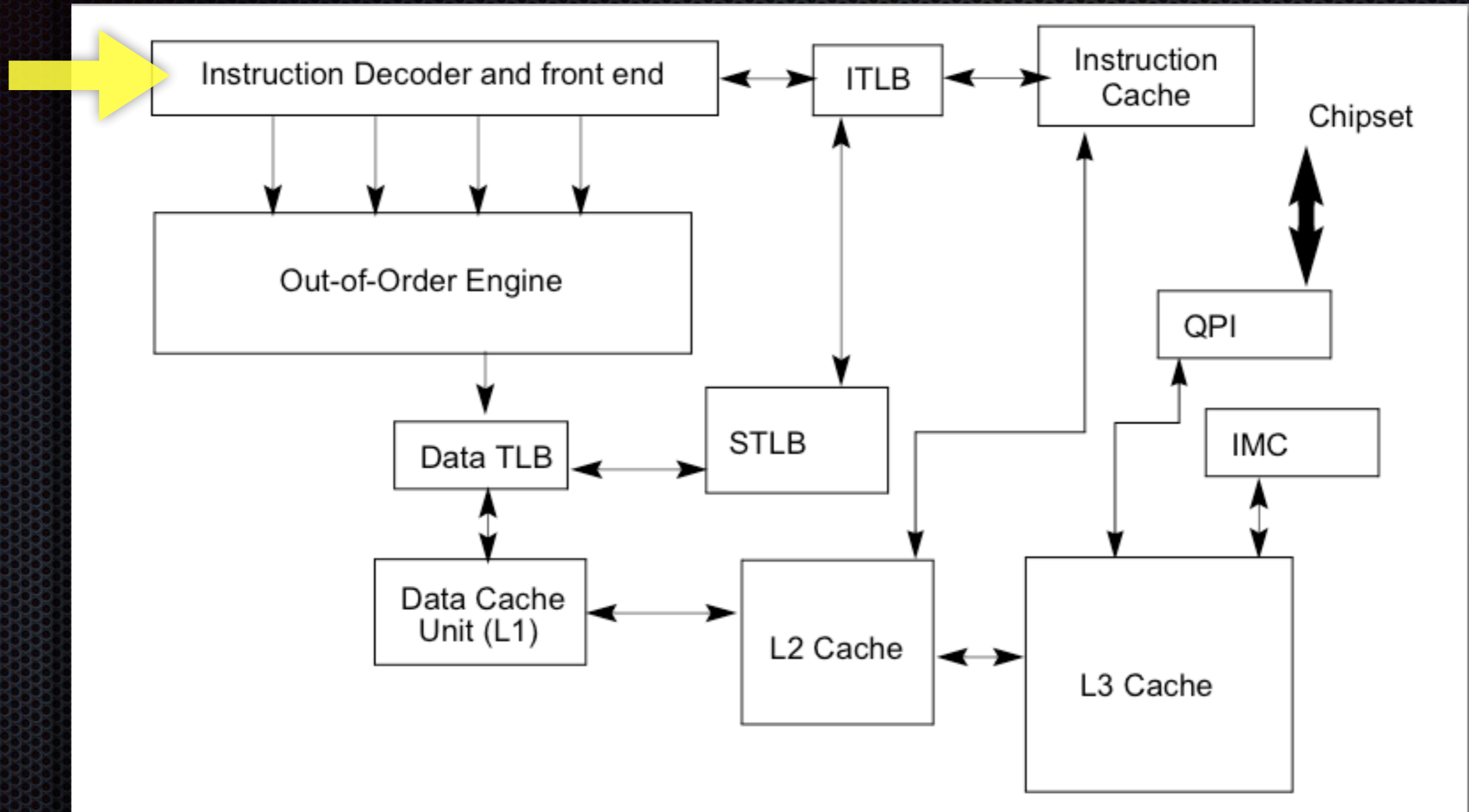
TICK

Sandy Bridge-E

First High End Desktop Platform
on the Sandy Bridge Microarchitecture



Cache Structure



Specifications (1/3)

- ✦ L1 Instruction Cache
 - ✦ 32-KByte, 4-way set associative
- ✦ L1 Data Cache
 - ✦ 32-KByte, 8-way set associative
- ✦ L2 Unified Cache
 - ✦ 256-KByte, 8-way set associative
- ✦ L3 Unified Cache
 - ✦ 15-MByte shared, 16-way set associative

Specifications (2/3)

- ✦ Instruction TLB (4-KByte pages)
 - ✦ 64-entries per thread (128-entries per core), 4-way set associative
- ✦ Data TLB (4-KByte pages)
 - ✦ DTLB0: 64-entries, 4-way set associative
- ✦ Instruction TLB (Large pages)
 - ✦ 7-entries per thread, fully associative
- ✦ Data TLB (Large pages)
 - ✦ DTLB0: 32-entries, 4-way set associative

Specifications (3/3)

- Second-level Unified TLB (4-KByte Pages)
 - STLB: 512-entries, 4-way set associative
- Store Buffer
 - 32-entries

Caching Terminology

- ✦ Cache coherency protocol
- ✦ Cache line fill
- ✦ Cache hit
- ✦ Cache miss
- ✦ Write hit
- ✦ Snooping

Methods of Caching

- ✦ Strong Uncacheable (UC)
- ✦ Uncacheable (UC-)
- ✦ Write Combining (WC)
- ✦ Write Through (WT)
- ✦ Write Back (WB)
- ✦ Write Protected (WP)

Cache Control Protocol

Cache Line State	M (Modified)	E (Exclusive)	S (Shared)	I (Invalid)
This cache line is valid?	Yes	Yes	Yes	No
The memory copy is...	Out of date	Valid	Valid	—
Copies exist in caches of other processors?	No	No	Maybe	Maybe
A write to this line ...	Does not go to the system bus.	Does not go to the system bus.	Causes the processor to gain exclusive ownership of the line.	Goes directly to the system bus.

MESI Protocol

- ✦ Upon loading:
 - ✦ A line is marked “E”
 - ✦ Subsequent read OK
 - ✦ Write marks “M”
- ✦ If another reads an “M” line
 - ✦ Write it back
 - ✦ Mark it “S”
- ✦ Write to an “S”, send “I” to all, mark “M”
- ✦ Read/write to an “I” misses

Cache Control

- ✦ Cache control registers and bits
- ✦ Cache management instructions

Registers and Bits

- ✦ NW, CD flag, bits 29, 30 of control register CR0
- ✦ PCD and PWT flags in paging-structure entries and control register CR3
- ✦ G (Global) flag in the page-directory and page-table entries
- ✦ PGE (Page Global Enable) flag in control register CR4
- ✦ Memory type range registers (MTRRs)
- ✦ Page Attribute Table (PAT)

Memory Type Range Registers (MTRRs)

- ✦ Provides a mechanism for associating the memory types with physical address ranges in system memory
- ✦ Allow the processor to optimize operations for different types of memory
- ✦ Simplify HW design
- ✦ Allows up to 96 memory ranges to be defined in physical memory
- ✦ In MP system, each processor MUST use the identical MTRR memory map

Page Attribute Table (PAT)

- ✦ Assigning memory type to the ranges of linear address space
- ✦ Checking PAT presence using CPUID
- ✦ MSR IA32_CR_PAT defines 8 types
- ✦ The type for a page is selected from IA32_CR_PAT by an index created from PAT(4), PCD(2), PWT(1) bits in page tables
- ✦ It is always switched on
- ✦ The initial setting after RESET is backward compatible with PCD and PWT (WB, WT, UC-, UC)

Memory Types Restrictions

- ✦ If CR0[CD]=1, then caching is disabled
- ✦ If CR0[CD]=0, the caching is restricted using PAT (or PCD and PWT) and MTRR
- ✦ Always selected the most restrictive type
 - ✦ WT “wins” over WB
 - ✦ WC “wins” over WT and WB

Management Instructions

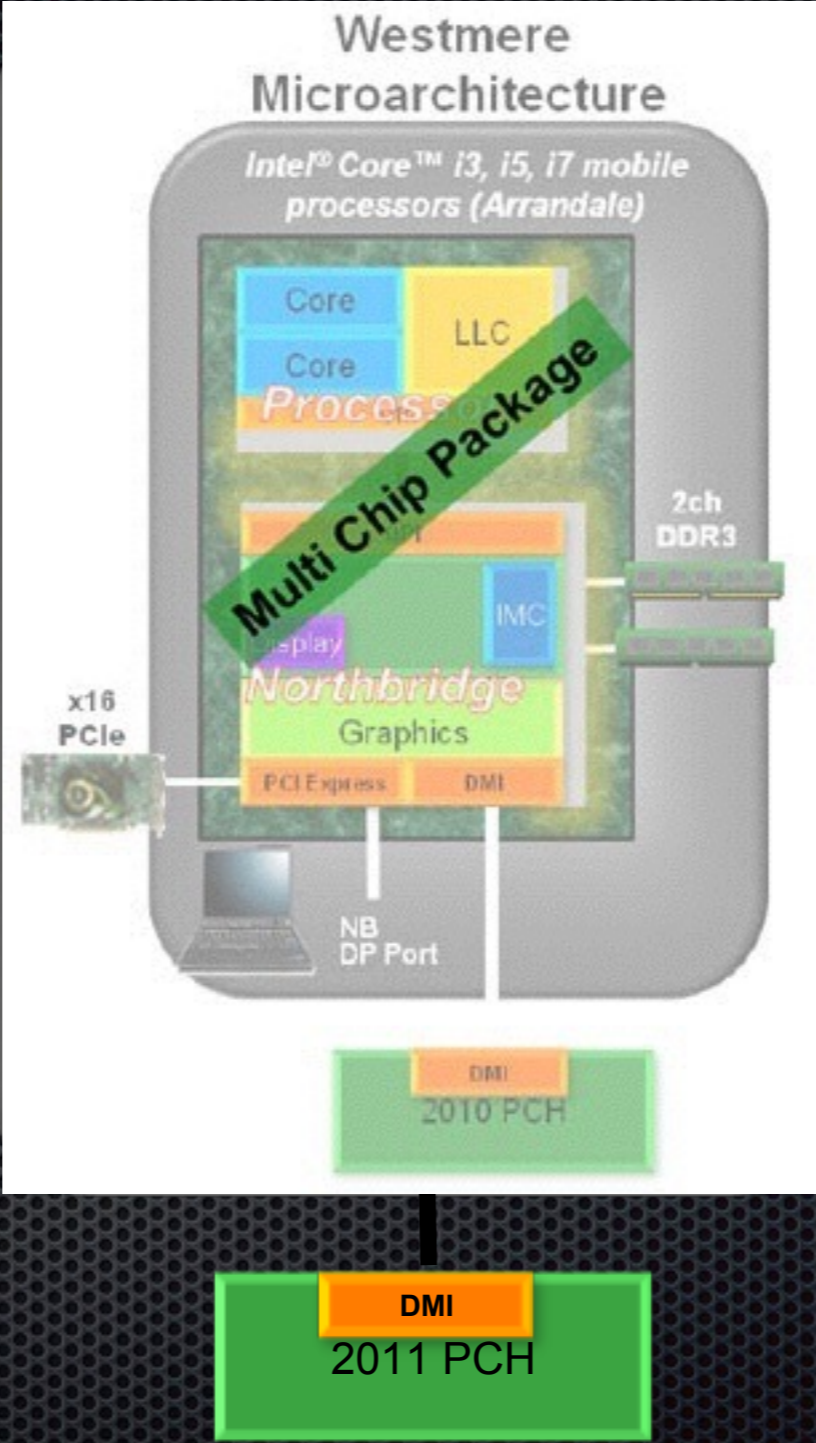
- ✦ INVD, WBINVD
- ✦ PREFETCHh, CFLUSH
- ✦ MOVNTI, MOVNTQ, MOVNTQD, MOVNTPS

Store Buffer

- ✦ Improve processor performance
- ✦ Contents are always drained to memory in the following situations:
 - ✦ When an exception or interrupt is generated
 - ✦ When a serializing instruction is executed
 - ✦ When an I/O instruction is executed
 - ✦ When a LOCK operation is performed
 - ✦ When a BINIT operation is performed
 - ✦ When using an SFENCE instruction to order stores

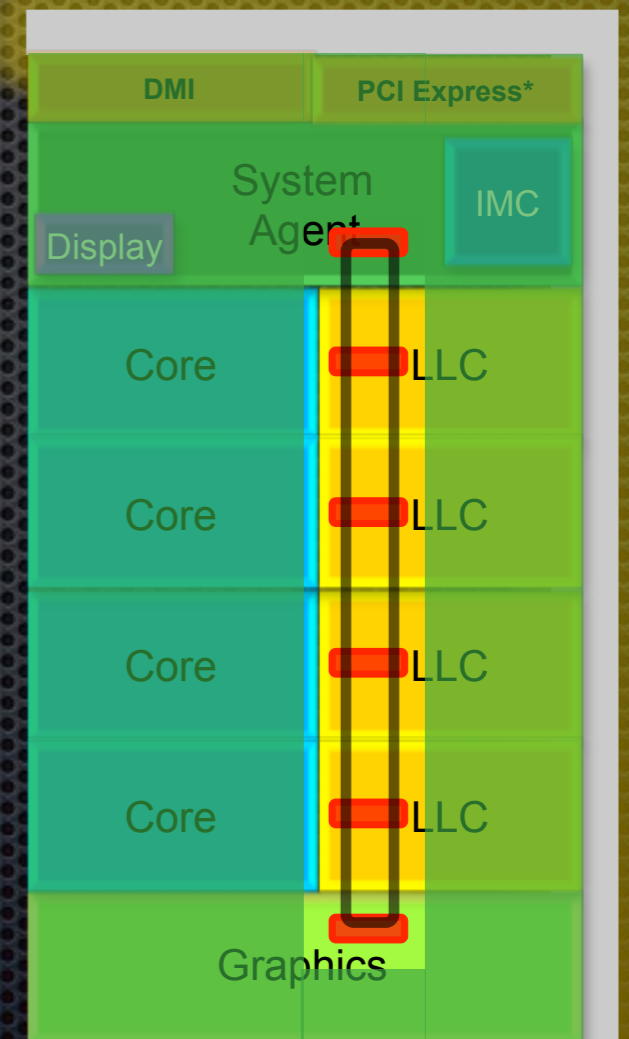
Ring Architecture Innovation in Sandy Bridge

2nd Gen Intel[®] Core[™] Microarchitecture



Scalable Ring On-die Interconnect

- Ring-based interconnect between Cores, Graphics, LLC and System Agent domain
- Composed of 4 rings
 - 32 Byte Data ring, Request Ring, Acknowledge ring and Snoop ring
 - Fully pipelined at core frequency/voltage: bandwidth, latency and power scale with cores
- Massive ring wire routing runs over the LLC with no area impact
- Access on ring always picks the shortest path - minimize latency
- Distribute arbitration, ring protocol handles coherency, ordering, and core interface
- **Scalable to servers** with large number of processors



High Bandwidth, Low Latency, Modular

Cache Box

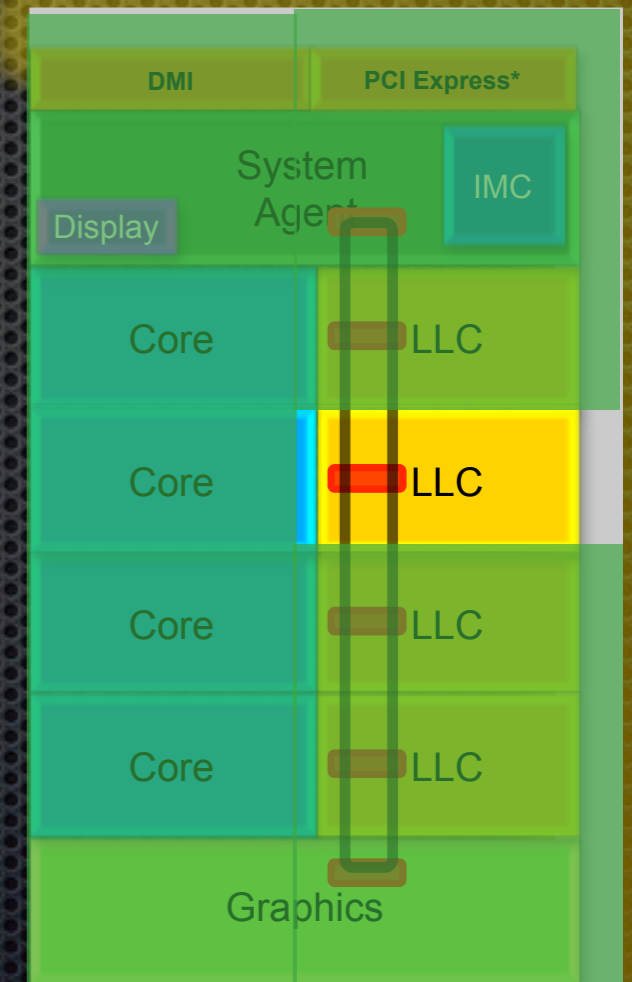
- **Interface block**

- Between **Core/Graphics/Media** and the **Ring**
- Between **Cache controller** and the **Ring**
- Implements the **ring logic, arbitration, cache controller**

- **Full cache pipeline** in each cache box

- Physical Addresses are **hashed at the source** to **prevent hot spots** and increase bandwidth
- Maintains **coherency and ordering** for the addresses **that are mapped to it**
- LLC is fully inclusive with “**Core Valid Bits**” - eliminates unnecessary snoops to cores

- Runs at **core voltage/frequency**, scales with Cores



*Distributed coherency & ordering; Scalable
Bandwidth, Latency & Power*

Thank you