TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

The University of Texas at Austin

Mapping Cores, CHAs, and Addresses in the Xeon Platinum 8380

Ice Lake Xeon

PRESENTED BY:

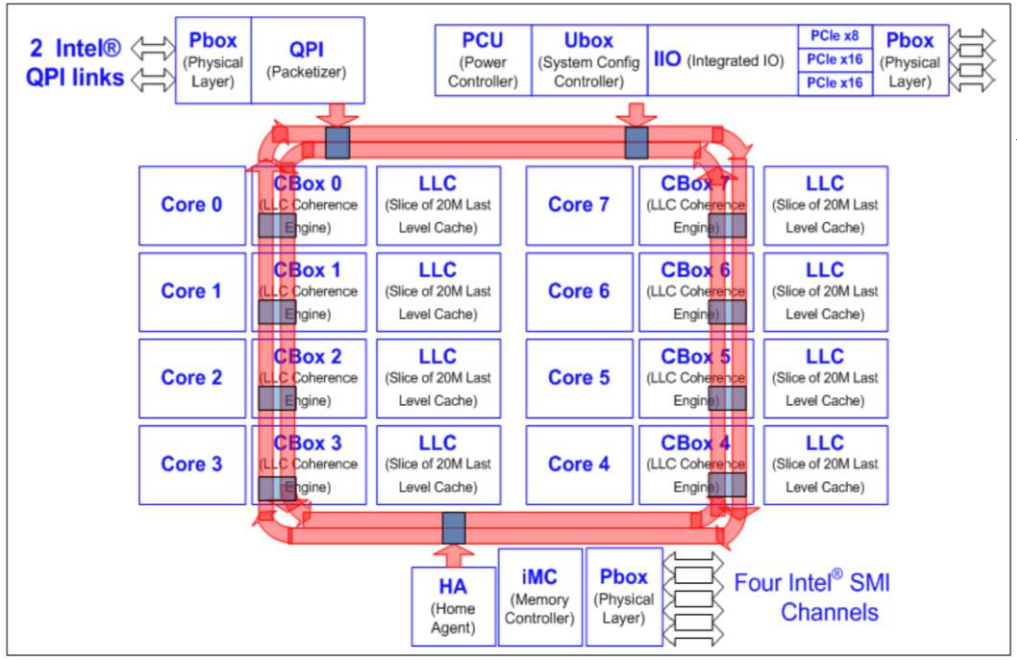
John D. McCalpin, Ph.D.

mccalpin@tacc.utexas.edu

Review: Mapping Intel Xeon Processors

- Starting with “Sandy Bridge EP” (Xeon E5-2xxx, 2Q2012), Intel’s multicore processors used a bidirectional ring topology.
  - The L3 cache was distributed around the chip, with cache line addresses mapped to L3 “slices” using an undocumented hash function.
  - Most models had one L3 slice for each core, so with uniform distribution of accesses each L3 slice would only need to handle the average traffic of one core – while still allowing any core to access the entire aggregate L3 cache.
  - Intel provided performance counters to measure mesh traffic at each L3 slice but provided no hints on the *mapping* of core and L3 slice numbers to *locations* on the die.

Figure 1-1. Uncore Sub-system Block Diagram of Intel Xeon Processor E5-2600 Family



← Single Bidirectional Ring (SNB-EP, IVB-EP, 1Q2012+)

Bridged Bidirectional Rings (IVB-EX, HSW/BDW-EP, 1Q2014+) →

Figure 1-2. Intel® Xeon® Processor E5 v3-1600/2600/4600 Family -12C Block Diagram

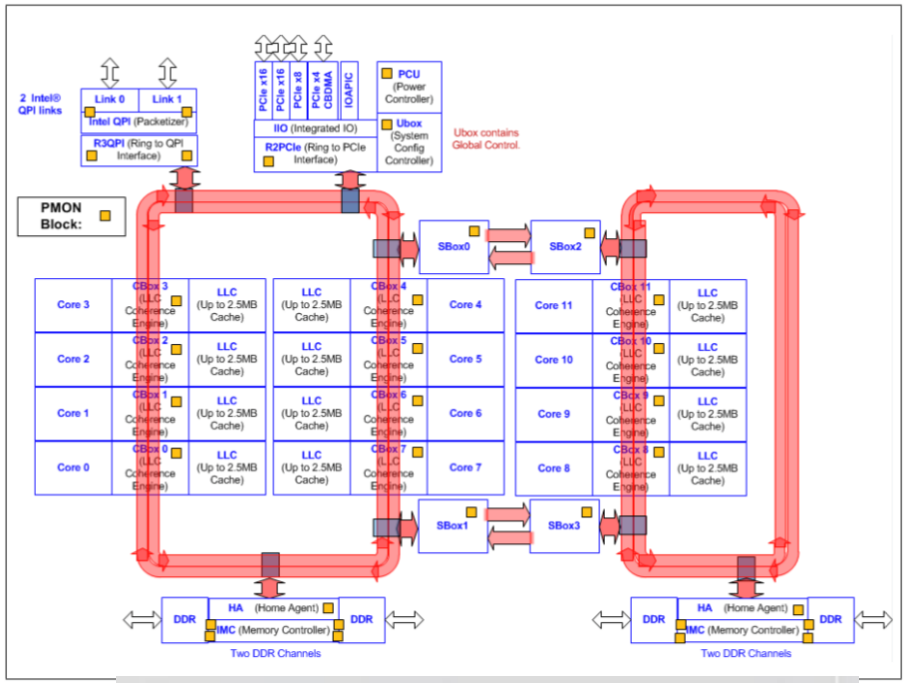
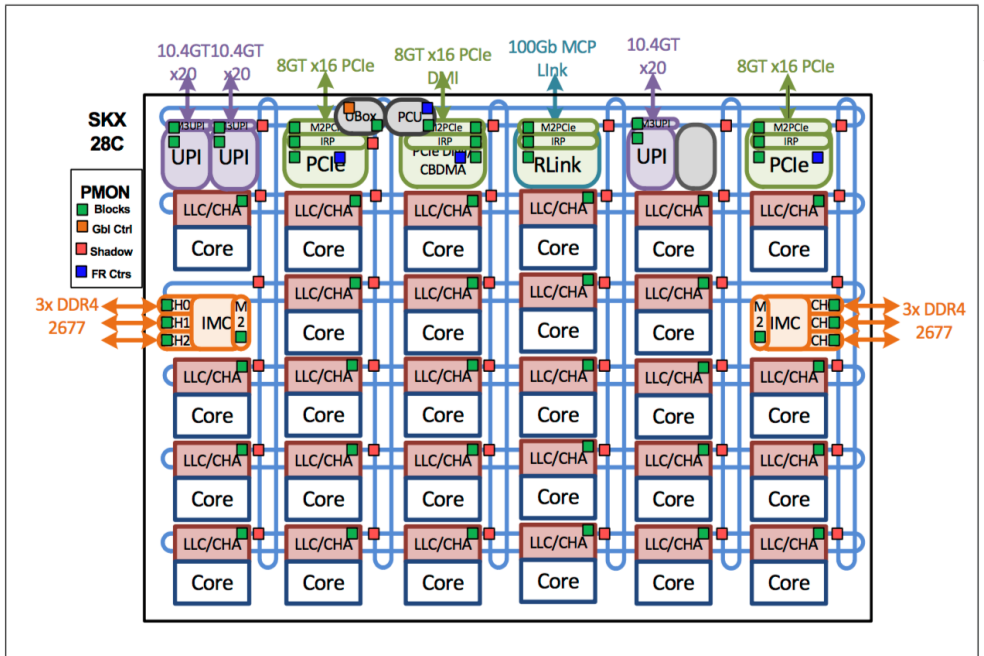
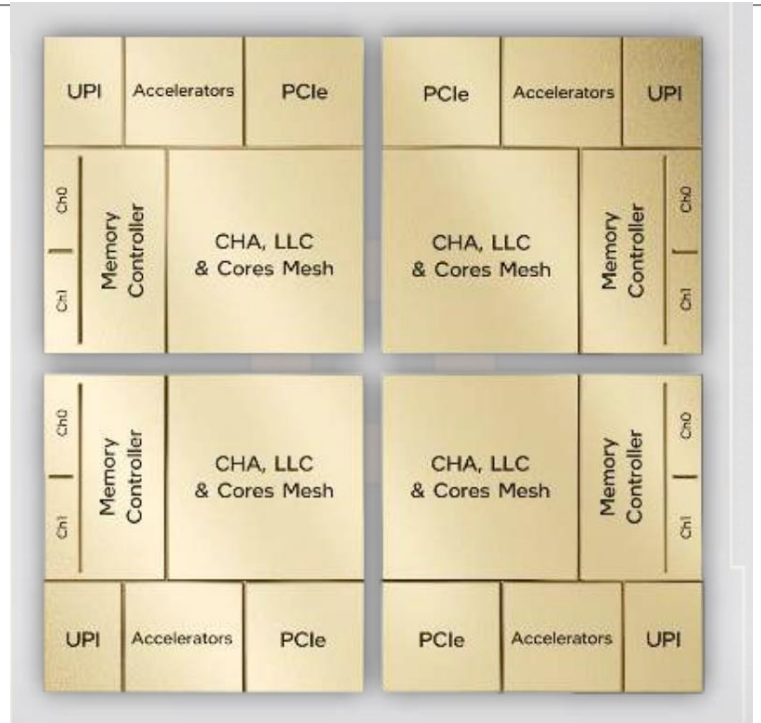


Figure 1-1. Intel® Xeon® Processor Scalable Memory Family - Block diagram for a 28C part



← 2D Bidirectional Mesh (KNL, SKX, CLX, ICX, 2Q2016 - present)

Multiple interconnected die, each with Bidirectional Mesh (Sapphire Rapids, TBD) →



Prior Work on Intel Processor layout and numbering

- "Topology and Cache Coherence in Knights Landing and Skylake Xeon Processors"
  - IXPUG Working Group Presentation 2018-04-12, http://dx.doi.org/10.26153/tsw/13160
- "Address Hashing in Intel Processors"
  - IXPUG Fall Conference 2018-09-25, http://dx.doi.org/10.26153/tsw/13161
- "Observations on Core Numbering and Core ID's in Intel Processors"
  - TR-2020-01, 2020-11-30, http://dx.doi.org/10.26153/tsw/10858
- "Mapping Core and L3 Slice Numbering to Die Locations in Intel Xeon Scalable Processors"
  - TR-2021-01b, 2021-02-28, http://dx.doi.org/10.26153/tsw/13119
- "Mapping Core, CHA, and Memory Controller Numbers to Die Locations in Intel Xeon Phi x200 (Knights Landing, KNL) Processors"
  - TR-2021-02, 2021-05-20, http://dx.doi.org/10.26153/tsw/13120
- "Mapping Addresses to L3/CHA Slices in Intel Processors"
  - TR-2021-03, 2021-09-10, http://dx.doi.org/10.26153/tsw/14539
- "Disabled Core Patterns and Core Defect Rates in Xeon Phi x200 (Knights Landing) Processors"
  - TR-2021-04, 2021-10-18, https://hdl.handle.net/2152/89580

Today's Topic: Changes in Ice Lake Xeon

- The mesh is similar, but
  - ICX has 4 Memory Controllers (vs 2 on SKX & CLX)
  - There are cores in the "IO" row at the bottom
  - No longer the same number of cores in left and right halves
- Numbering conventions for CHAs and Cores have changed
  - These were the same for SNB/IVB, HSW/BDW, SKX/CLX
- Changes in address to CHA mapping functions
  - These were the same (for the same core count) for SNB/IVB, HSW/BDW, SKX/CLX

Figure 1-1. Intel® Xeon® Processor Scalable Memory Family - Block diagram for a 28C part

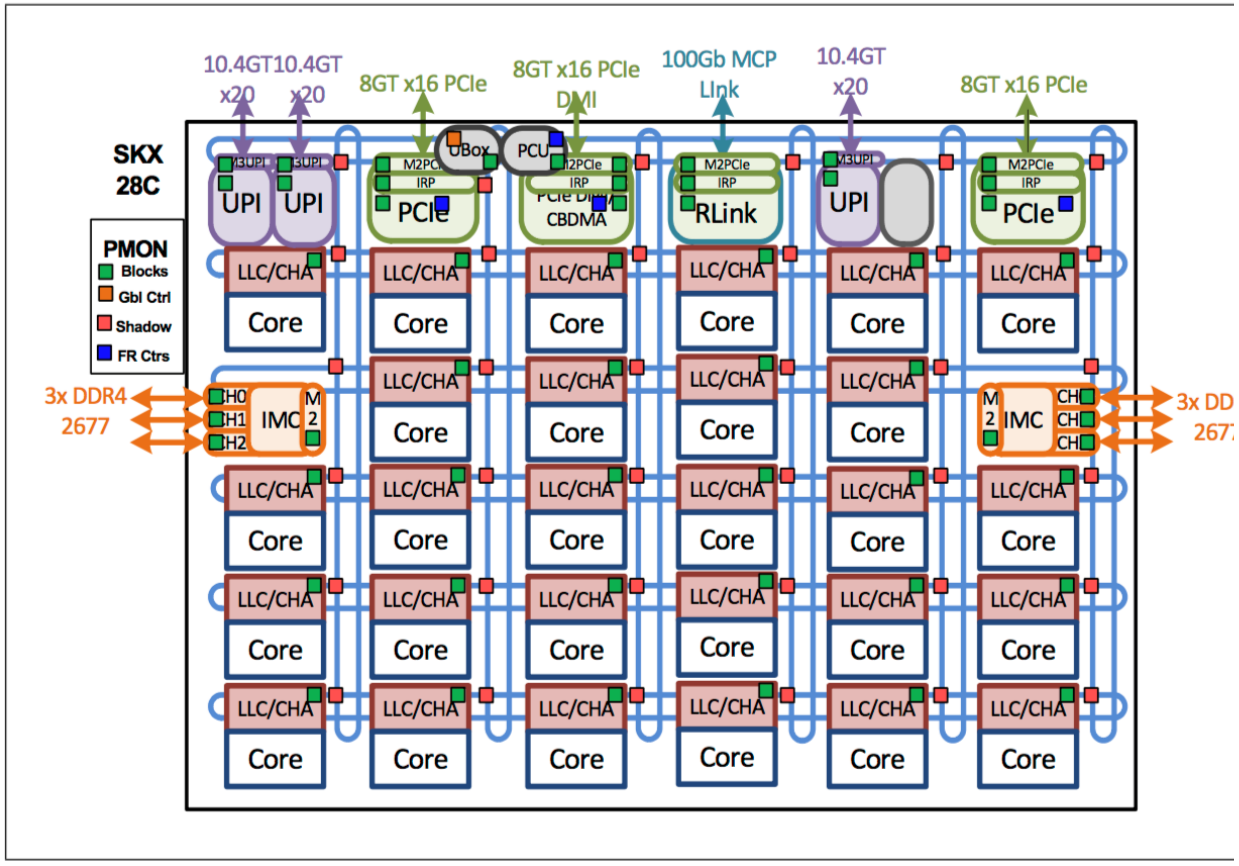
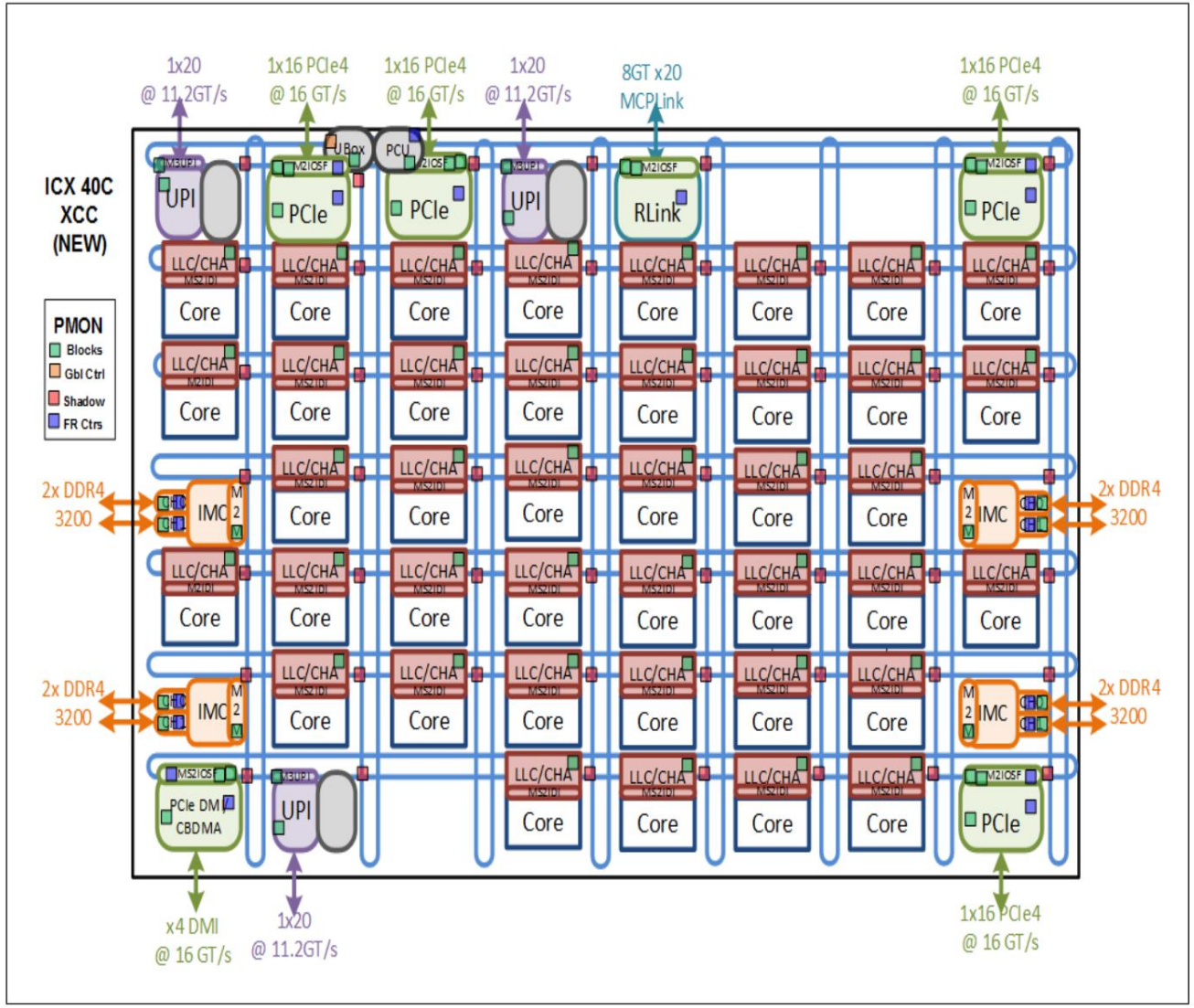


Figure 1-1. 3rd Gen Intel Xeon Processor Scalable Family Server-40C XCC Block Diagram



Numbering Conventions

- Registers in PCI configuration space hold bit maps of the enabled L3 slices in HSW/BDW, SKX/CLX, and ICX
  - No hints about mapping of bit positions to physical positions
- An analogous register holds a bit map of the enabled cores
  - Only documented in HSW/BDW, but easily found for SKX/CLX/ICX
- After determining the locations of enabled LLC slices in many processors, the interpretation of the bit map is now clear – at least for the cases reviewed so far…

For SKX/CLX:

CAPID6 bits 27:0 map directly to tile location

UPI 0/1	PCIe	PCIe	Rsvd	UPI 2	PCIe
Bit 0	Bit 4	Bit 9	Bit 14	Bit 19	Bit 24
IMC0	Bit 5	Bit 10	Bit 15	Bit 20	IMC1
Bit 1	Bit 6	Bit 11	Bit 16	Bit 21	Bit 25
Bit 2	Bit 7	Bit 12	Bit 17	Bit 22	Bit 26
Bit 3	Bit 8	Bit 13	Bit 18	Bit 23	Bit 27

Example: assume CAPID6 bits 4, 10, 17, 23 are zero.
Logical CHA numbers skip over disabled tiles

UPI 0/1	PCIe	PCIe	Rsvd	UPI 2	PCIe
CHA 0	disabled	CHA 8	CHA 12	CHA 16	CHA 20
IMC0	CHA 4	disabled	CHA 13	CHA 17	IMC1
CHA 1	CHA 5	CHA 9	CHA 14	CHA 18	CHA 21
CHA 2	CHA 6	CHA 10	disabled	CHA 19	CHA 22
CHA 3	CHA 7	CHA 11	CHA 15	disabled	CHA 23

Sample Xeon Platinum 8160 Core & Tile (CHA) numbering

UPI 0/1	PCIe	PCIe	*RLink*	*UPI 2*	PCIe
c0 t0	disabled	c2 t8	c3 t12	c4 t16	c5 t20
IMC0	c1 t4	disabled	c15 t13	c16 t17	IMC1
c12 t1	c13 t5	c14 t9	c9 t14	c10 t18	c17 t21
c6 t2	c7 t6	c8 t10	disabled	c22 t19	c11 t22
c18 t3	c19 t7	c20 t11	c21 t15	disabled	c23 t23

Xeon Platinum 8380 (Ice Lake Xeon) XCC die (40-core), Core and CHA numbering

UPI	PCIe x16	PCIe x16	UPI	Rlink (unused)	(unused)	(unused)	PCIe x16
Core 0 CHA 0	Core 4 CHA 1	Core 8 CHA 2	Core 12 CHA 3	Core 76 CHA 19	Core 2 CHA 20	Core 6 CHA 21	Core 10 CHA 22
Core 16 CHA 4	Core 20 CHA 5	Core 24 CHA 6	Core 28 CHA 7	Core 14 CHA 23	Core 18 CHA 24	Core 22 CHA 25	Core 26 CHA 26
IMC 0	Core 32 CHA 8	Core 36 CHA 9	Core 40 CHA 10	Core 30 CHA 27	Core 34 CHA 28	Core 38 CHA 29	IMC 2
Core 44 CHA 11	Core 48 CHA 12	Core 52 CHA 13	Core 56 CHA 14	Core 42 CHA 30	Core 46 CHA 31	Core 50 CHA 32	Core 54 CHA 33
IMC 1	Core 60 CHA 15	Core 64 CHA 16	Core 68 CHA 17	Core 58 CHA 34	Core 62 CHA 35	Core 66 CHA 36	IMC 3
PCIe / DMI x4	UPI	(unused)	Core 72 CHA 18	Core 70 CHA 37	Core 74 CHA 38	Core 78 CHA 39	PCIe x16

Xeon Platinum 8380: Scaled Mesh Traffic Counts: Core 24 reading from IMC 2

UPI	PCIe	PCIe	UPI				PCIe
0.02 0.00 0.15 0.09	0.00 0.02 0.18 0.00	0.00 0.02 0.10 0.20	0.00 0.11 0.10 0.00	0.00 0.09 0.02 0.06	0.00 0.02 0.00 0.02	0.00 0.01 0.00 0.00	0.00 0.01 0.00 0.00
0.20 0.00 0.05 0.07	0.00 0.04 0.08 0.00	0.00 0.05 core 24 1.10 0.04	0.00 0.09 1.10 0.01	0.02 0.06 1.04 0.04	0.00 0.02 1.01 0.05	0.00 0.01 1.01 0.01	0.00 0.01 0.00 1.00
IMC 0 IMC 0 IMC 0 IMC 0 IMC 0 IMC 0 IMC 0 IMC 0 IMC 0	0.00 0.02 0.03 0.00	0.46 0.02 0.00 0.04	0.00 0.09 0.00 0.00	0.00 0.06 0.00 0.04	0.00 0.05 0.00 0.01	0.00 0.01 0.00 0.01	**IMC 2 IMC 2 IMC 2** **IMC 2 IMC 2 IMC 2** **IMC 2 IMC 2 IMC 2**
0.14 0.00 0.03 0.07	0.00 0.02 0.06 0.00	0.33 0.02 0.00 0.04	0.00 0.07 0.00 0.00	0.00 0.05 0.00 0.00	0.00 0.02 0.00 0.00	0.00 0.01 0.00 0.00	0.00 0.01 0.00 0.00
IMC 1 IMC 1 IMC 1 IMC 1 IMC 1 IMC 1 IMC 1 IMC 1 IMC 1	0.00 0.02 0.03 0.00	0.18 0.02 0.00 0.00	0.00 0.04 0.00 0.00	0.00 0.02 0.00 0.00	0.00 0.01 0.00 0.00	0.00 0.01 0.00 0.00	IMC 3 IMC 3 IMC 3 IMC 3 IMC 3 IMC 3 IMC 3 IMC 3 IMC 3
PCIe/DMI	UPI		0.00 0.04 0.00 0.00	0.00 0.02 0.00 0.00	0.00 0.01 0.00 0.00	0.00 0.01 0.00 0.00	PCIe

Mapping Addresses to CHAs

- 28-core ICX hash is different than 28-core SKX/CLX hash – that has not happened at fixed core-count before…
- The ICX 28-core hash has slightly better conflict properties than the SKX/CLX 28-core hash, but not by very much.
- The ICX 40-core hash has the same fundamental weaknesses as the hashes for all prior systems – each 2MiB page maps multiple addresses to the same (global) L3 congruence classes.

		L3/CHA/SF Slice Number					
		0	1	2	3	4	5
Local cache set number	15	90	91	92	93	94	95
	14	84	85	86	87	88	89
	13	78	79	80	81	82	83
	12	72	73	74	75	76	77
	11	66	67	68	69	70	71
	10	60	61	62	63	64	65
	9	54	55	56	57	58	59
	8	48	49	50	51	52	53
	7	42	43	44	45	46	47
	6	36	37	38	39	40	41
	5	30	31	32	33	34	35
	4	24	25	26	27	28	29
	3	18	19	20	21	22	23
	2	12	13	14	15	16	17
	1	6	7	8	9	10	11
	0	0	1	2	3	4	5

Consider a 6-slice cache with 16 sets per slice, where each set is a set-associative entry.

To use the cache effectively, a simple mapping scheme might map the addresses as shown at left – guaranteeing that contiguous addresses allocate once into every set before allocating twice into any set.

Intel Xeon processors do not do this….

SF (& L3) Conflicts for 2MiB pages in Intel Processors

Generation	Slices	SF/L3 sets	sets/2MiB page	Aggr L2 MiB	2MiB pages in L2	SF Associativity	Max SF Occupancy per 2MiB page	Worst-case L2 capacity MiB	Worst-case Fraction of actual L2
KNL	38	77824	2.375	34	17	12	3	8	23.5%
SKX/CLX	14	28672	0.875	14	7	12	2	12	85.7%
	16	32768	1.000	16	8	12	1	24	150.0%
	18	36864	1.125	18	9	12	3	8	44.4%
	20	40960	1.250	20	10	12	2	12	60.0%
	22	45056	1.375	22	11	12	2	12	54.5%
	24	49152	1.500	24	12	12	3	8	33.3%
	26	53248	1.625	26	13	12	2	12	46.2%
	28	57344	1.750	28	14	12	2	12	42.9%
ICX	28	57344	1.750	35	17.5	16	2	16	45.7%
	40	81920	2.500	50	25	16	2	16	32.0%

		Sets with 0 Allocations		Sets with 1 Allocation		Sets with 2 Allocations		Sets with 3 Allocations		
Family	Slices	# of sets	% of lines	# of sets	% of lines	# of sets	% of lines	# of sets	% of lines	%lines over-subscribed
KNL	38	55680	71.55%	11712	15.05%	10240	13.16%	192	0.25%	13.41%
SKX/CLX	14	0	0.00%	24576	85.71%	4096	14.29%			14.29%
	16	0	0.00%	32768	100.00%					0.00%
	18	15872	43.06%	10112	27.43%	9984	27.08%	896	2.43%	29.51%
	20	18432	45.00%	12288	30.00%	10240	25.00%			25.00%
	22	17216	38.21%	22912	50.85%	4928	10.94%			10.94%
	24	31744	64.58%	5120	10.42%	9216	18.75%	3072	6.25%	25.00%
	26	23552	44.23%	26624	50.00%	3072	5.77%			5.77%
	28	26624	46.43%	28672	50.00%	2048	3.57%			3.57%
ICX	28	25920	45.20%	30080	52.46%	1344	2.34%			2.34%
	40	59392	72.50%	12288	15.00%	10240	12.50%			12.50%

Allocation of lines to Snoop Filter & L3 sets for a single 2MiB page

Summary

- The layout and unit numbering of the Ice Lake Xeon is very similar to the previous Xeon Scalable Processors
  - but still required a significant exercise of measurement & analysis
- The conventions for numbering CHAs and cores differ substantially from previous processors
  - More data is needed for processor models with disabled cores and CHA/L3 slices.
- Improvements in Snoop Filter and L3 don't keep up with scaling
  - SF increased from 12x → 16x, but L2 capacity increased from 14 → 25 2MiB pages
  - L3 increased from 11x → 12x, but L3 capacity increased from 19 → 30 2MiB pages
- Improvements (?) in Address Hash don't eliminate fundamental problem of over-allocation to a subset of the sets when using 2MiB pages
  - A simple 1-bit change to one of the address hash masks eliminates over-allocation for the 40-slice system!

Abstract

Intel Xeon processors from the Xeon E5 ("Sandy Bridge EP") through the 2nd generation Intel Xeon Scalable Processors ("Cascade Lake") employed the same patterns for x2APIC core ID numbering, for the numbering of cores and CHA slices on the die, and for mapping physical addresses to CHA slices. The 3rd generation Intel Xeon Scalable Processors ("Ice Lake Xeon") include some small, but significant changes in the all three of these areas. We review prior results, then show how the 3rd generation processors differ. The changes in numbering of cores and CHAs do not directly influence performance, but the mapping of addresses to CHAs does differ. New results show that the new hash function used by Intel is better than the previous version but is not good enough to avoid the pathological snoop filter conflicts seen in previous generations.