



**IBM Inside Sales**

International Technical Support Organization Global Content Services

# ITSO – z System Hardware Workshop

[www.ibm.com/redbooks](http://www.ibm.com/redbooks)

Part 2 – CPC Details, Capacity and Performance



# Trademarks

**The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.**

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) I:

BladeCenter®, DB2®, e business(logo)®, DataPower®, ESCON, eServer, FICON, IBM®, IBM (logo)®, MVS, OS/390®, POWER6®, POWER6+, POWER7®, Power Architecture®, PowerVM®, S/390®, System p®, System p5, System x®, z Systems®, System z9®, System z10®, WebSphere®, X-Architecture®, zEnterprise®, z9®, z10®, z196®, z114®, zEnterprise System z196®, zEnterprise System z114®, zEnterprise System zEC12®, zEnterprise System zBC12®, z13®, z/Architecture®, z/OS®, z/VM®, z/VSE®, zSeries®

**The following are trademarks or registered trademarks of other companies.**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

\* All other products may be trademarks or registered trademarks of their respective companies.

## Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

## Important information about today's workshop

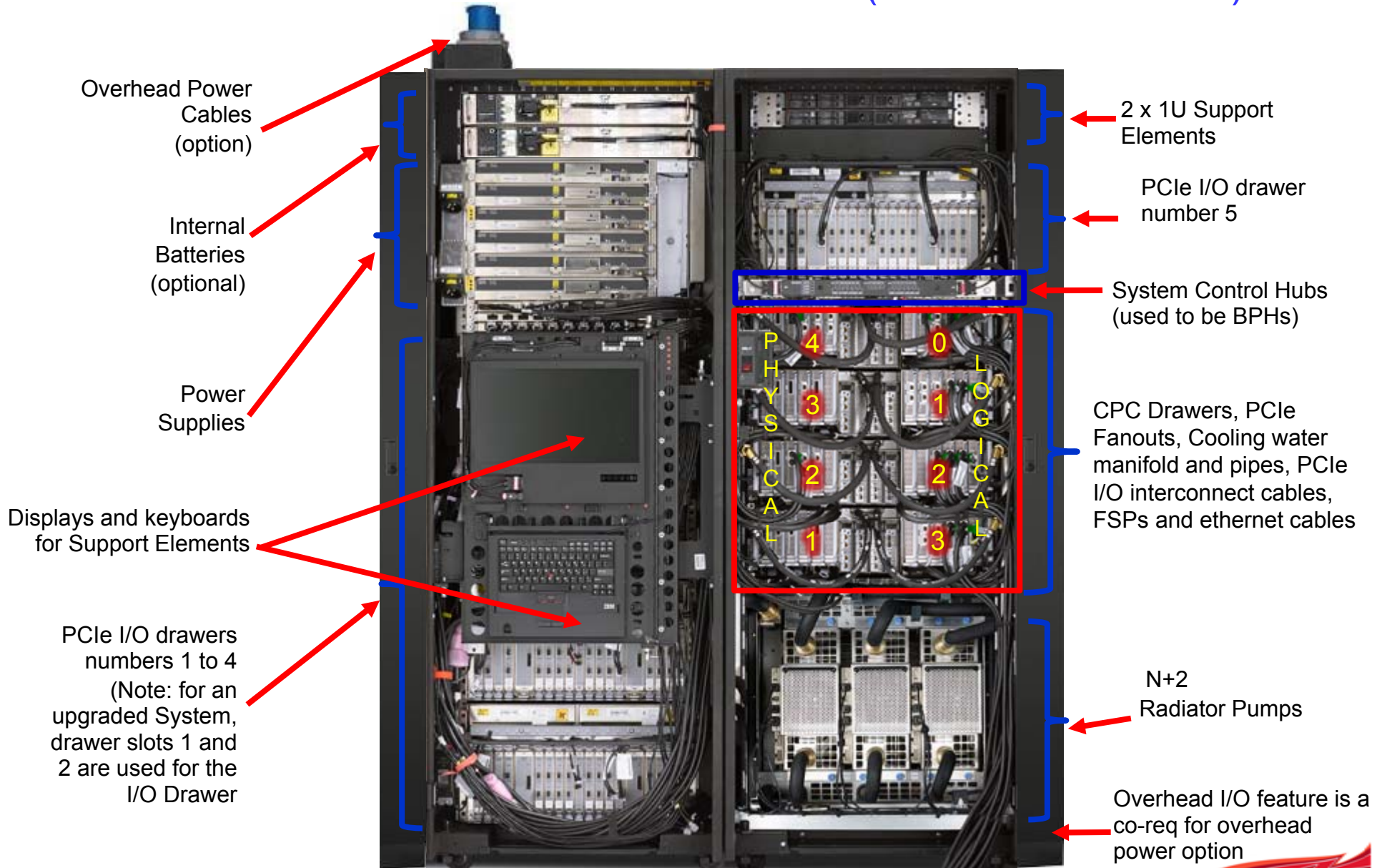
- The ITSO z hardware team created 7 IBM z13 presentations to be delivered today
  - Part 1 – IBM z13 – Positioning / introduction
  - **Part 2 – z13 CPC Details Capacity and Performance**
  - Part 3 – z13 I/O Subsystem
  - Part 4 – Native PCIe Adapters – zEDC and RoCE (what's new with z13)
  - Part 5 – HMC, CoD and RAS and zAware
  - Part 6 – Installation Planning
  - Part 7 – Software Support
  
- The main references for the presentations today are:
  - IBM z13 Technical Guide – Redbook – SG24-8251
  - IBM z13 Technical Introduction – Redbook - SG24-8250
  
- **Part of the available material may not be presented..** 😞
  - Even if we don't cover the presentations entirely,
    - The material can be download from:
      - <http://www.redbooks.ibm.com/Redbooks.nsf/pages/addmats>
  
- **The material being presented may not fully match the copied version you have**
  
- **You can always get the latest version .. If you want it, just ask !** 😊
  
- **Please ask questions, make comments and share your own experiences at any time**
  
- **Thank You !**



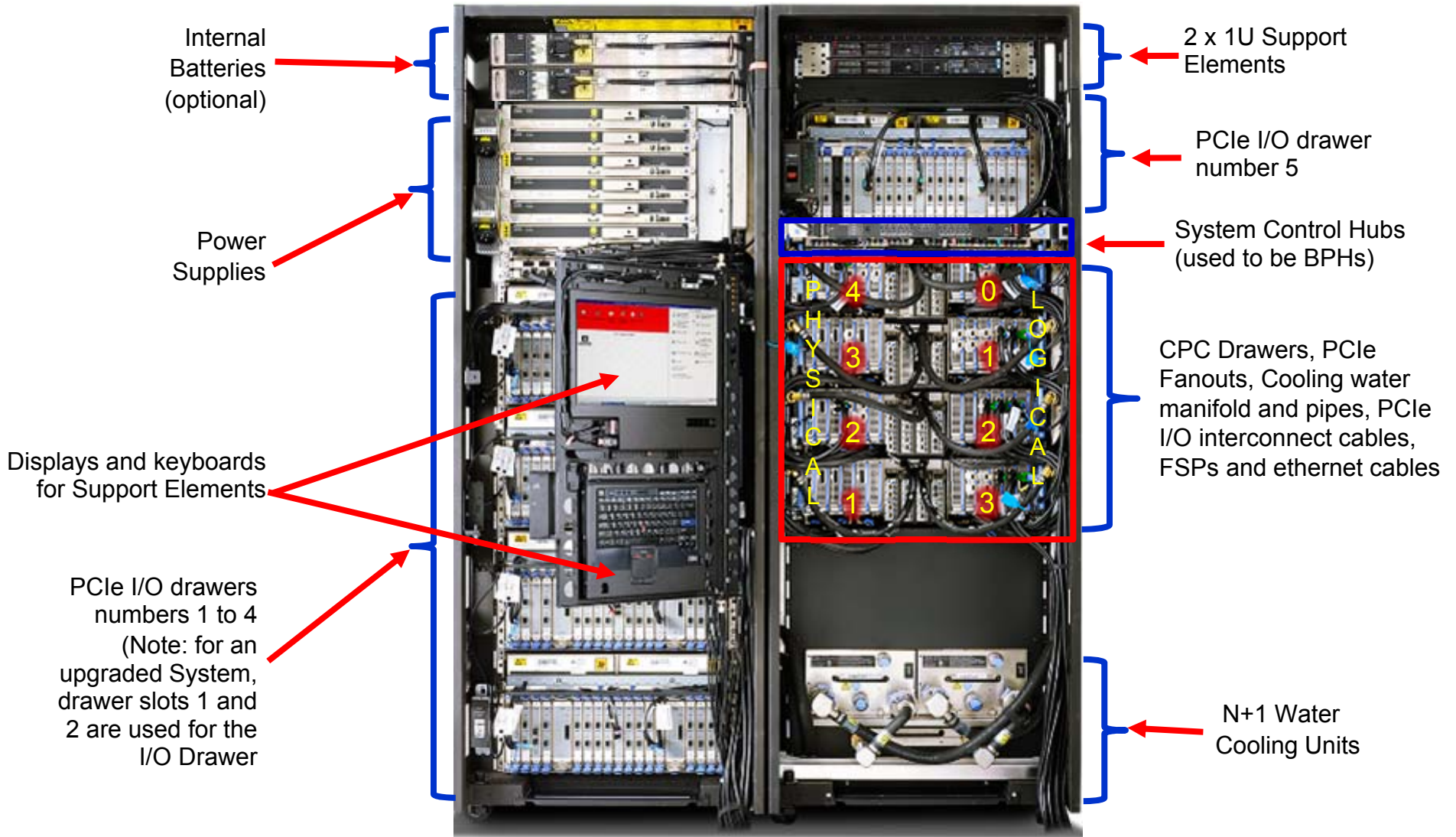
## The IBM z13 under the covers



# z13 Radiator-based Air cooled – Front View (Model NC9 or NE1)

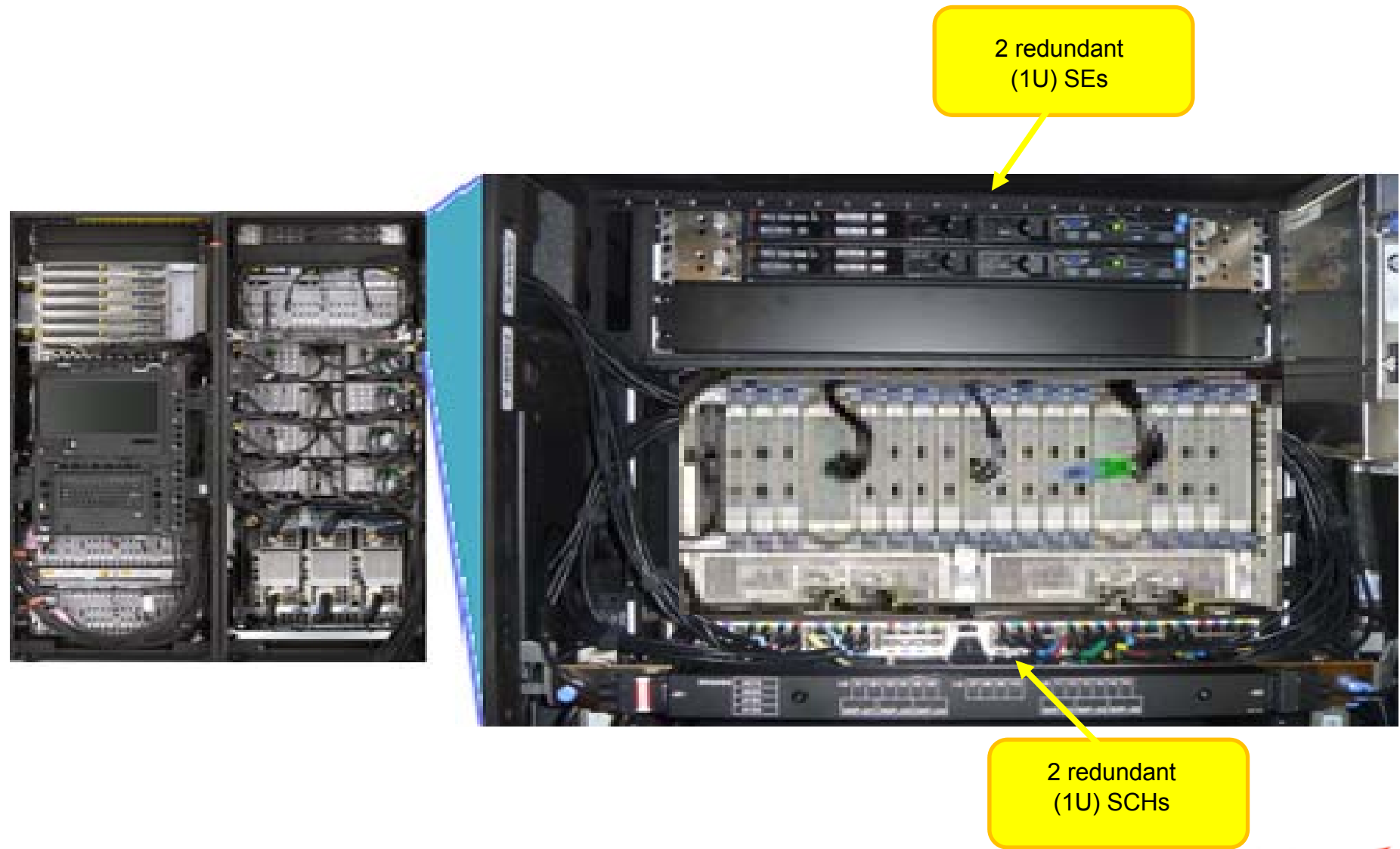


# z13 Water cooled – Front View (Model NC9 or NE1)



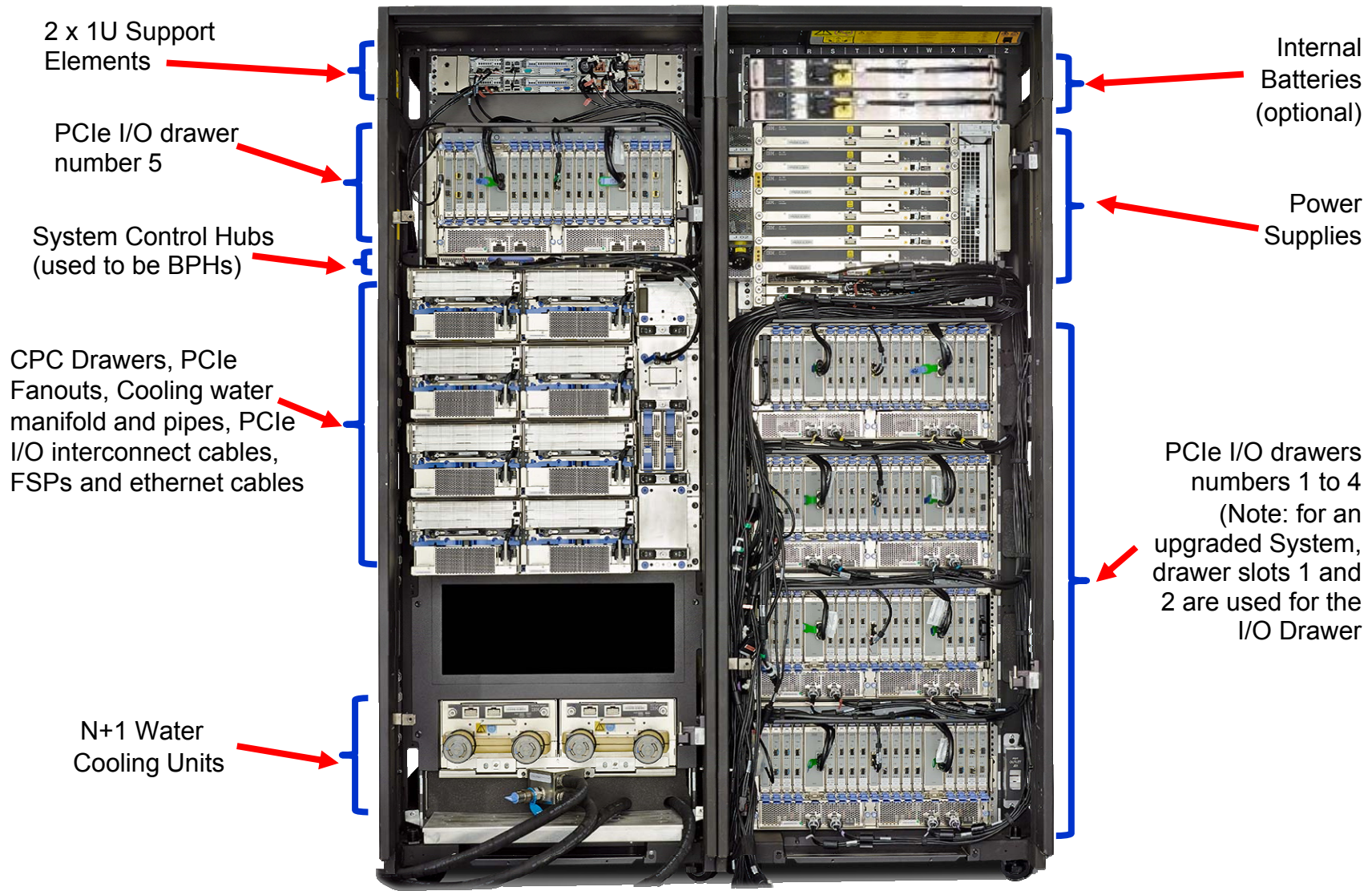
\* Overhead Power and I/O options not shown. Same as for the Air Cooled System

# z13 SE's and SCHs detailed view (top of "A" Frame A front)

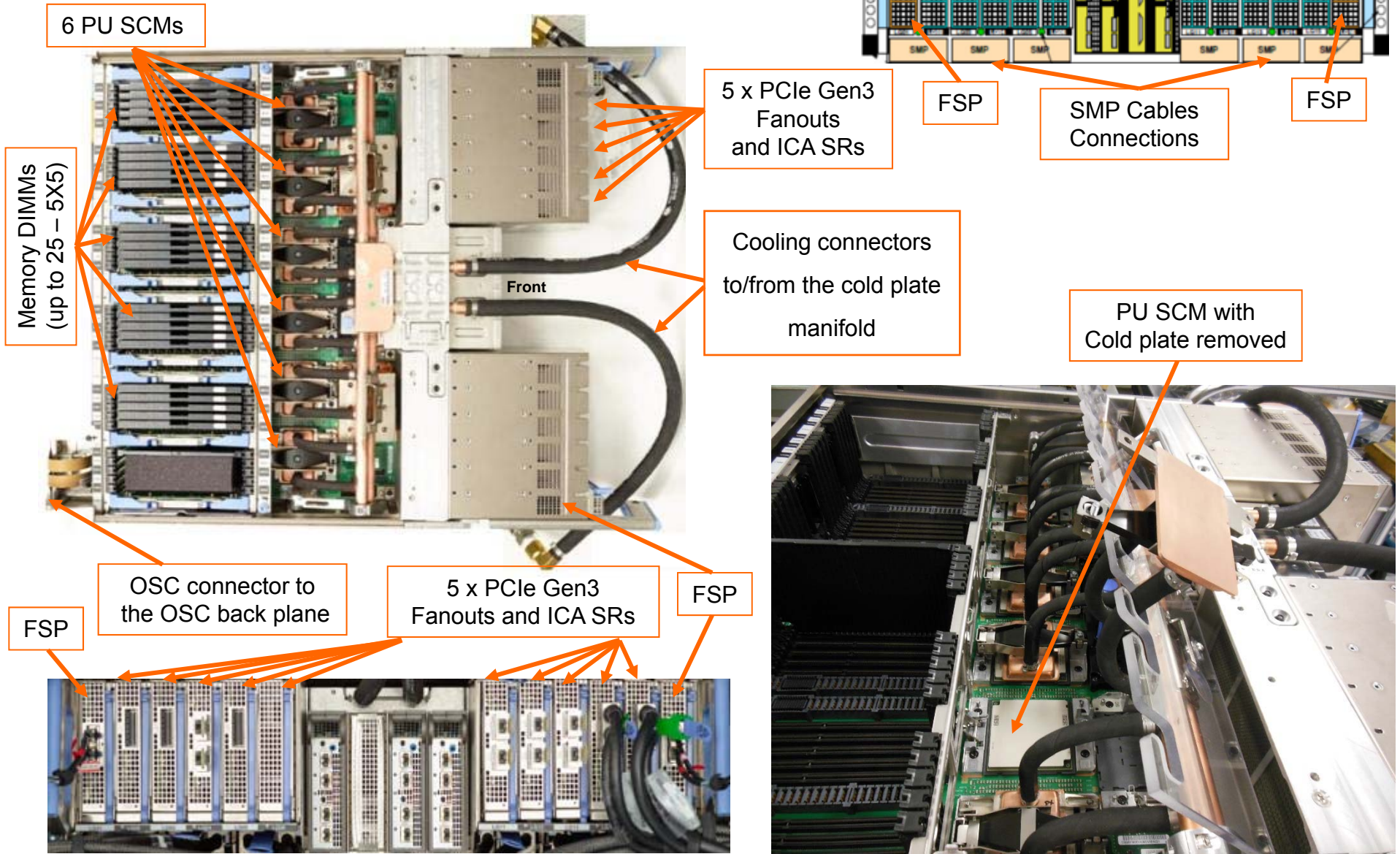




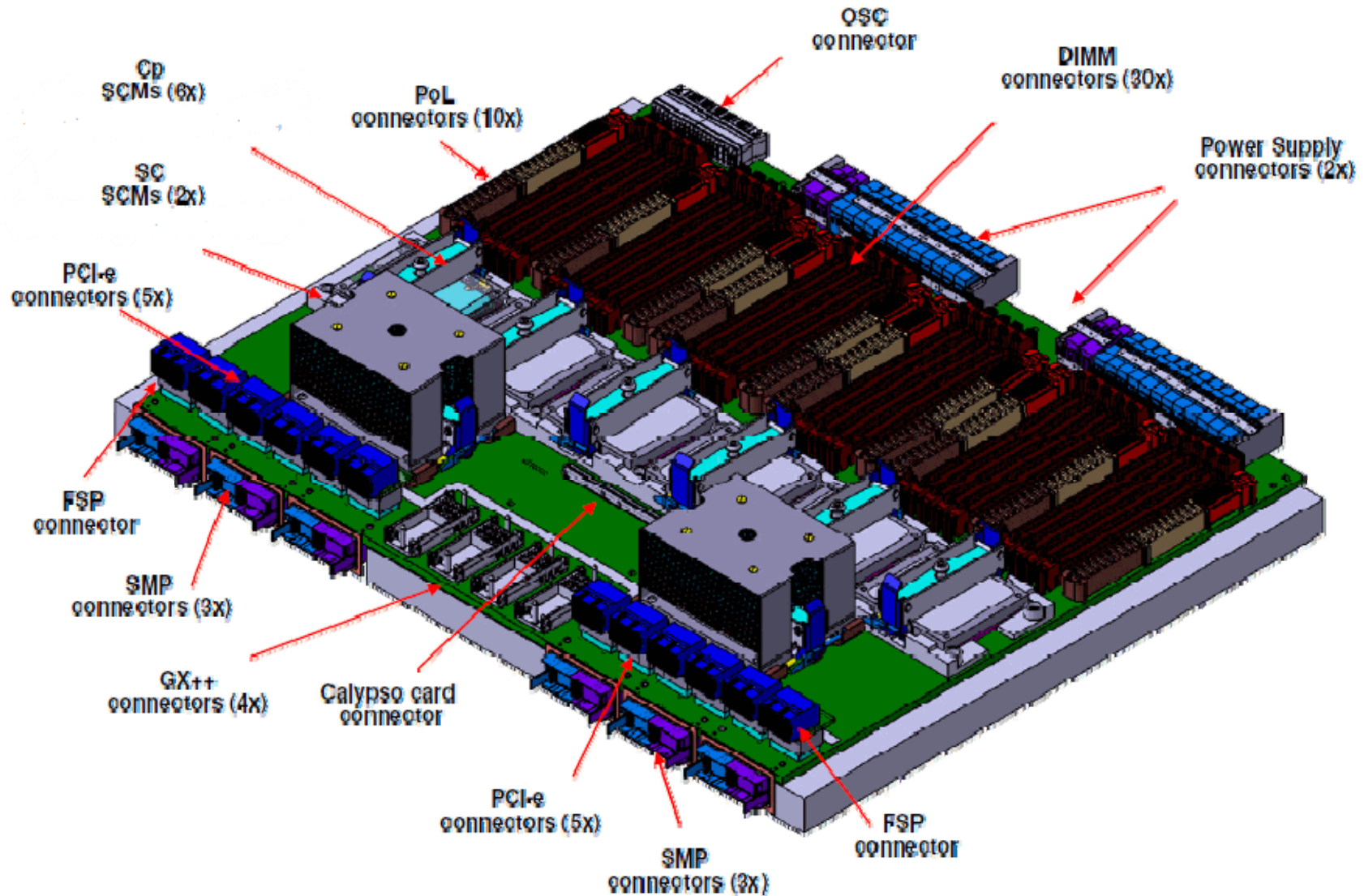
# z13 Water cooled – Rear View (Model NC9 or NE1)



# z13 CPC Drawer Layout



# z13 CPC Drawer Layout



# z13 PU and SC SCM assembly

Capped PU

6x PU SCMs

PU Chip

2x SC SCMs (Air Cooled)

SC SCM with Thermal Module

Capped SC

SC Chip

6x PU SCMs under the cold-plates

Front of CPC Drawer for Fanouts/FSPs

Fully assembled CPC Drawer with the chilled water supply manifold lifted to the left

Drawer 1

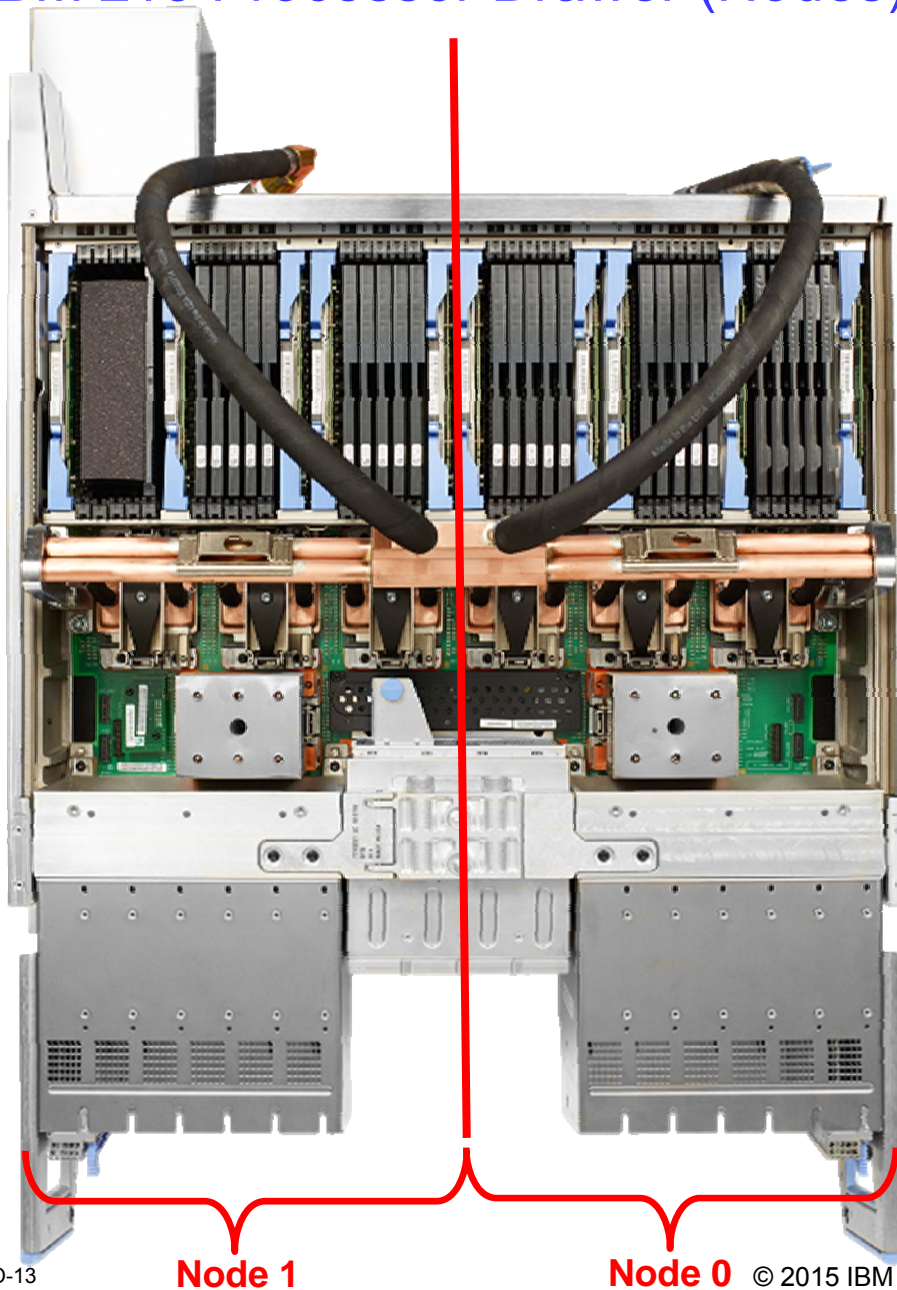
Drawer 2

Drawer 3

Drawer 4

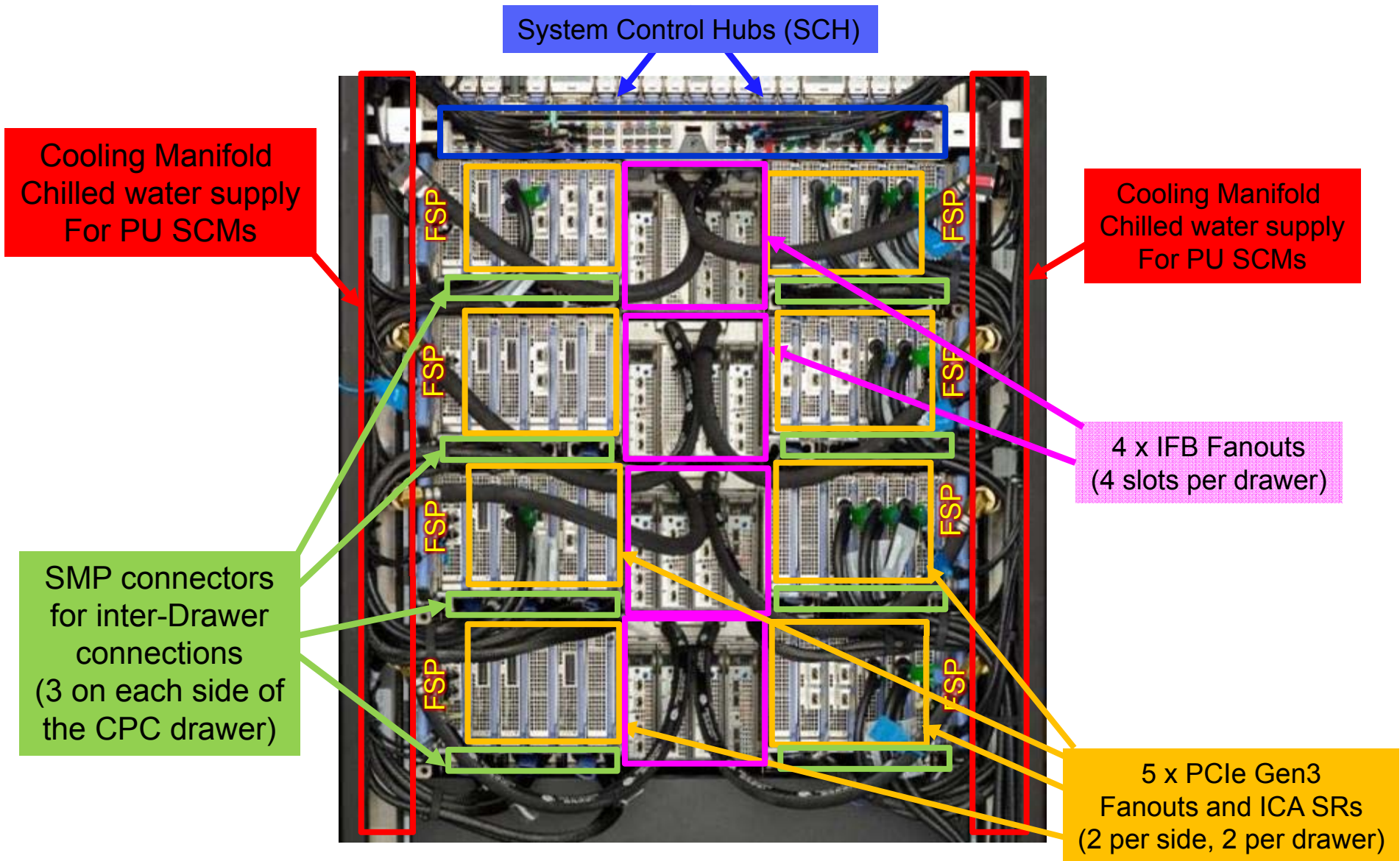
4 CPC Drawer System connectivity

## IBM z13 Processor Drawer (Nodes)

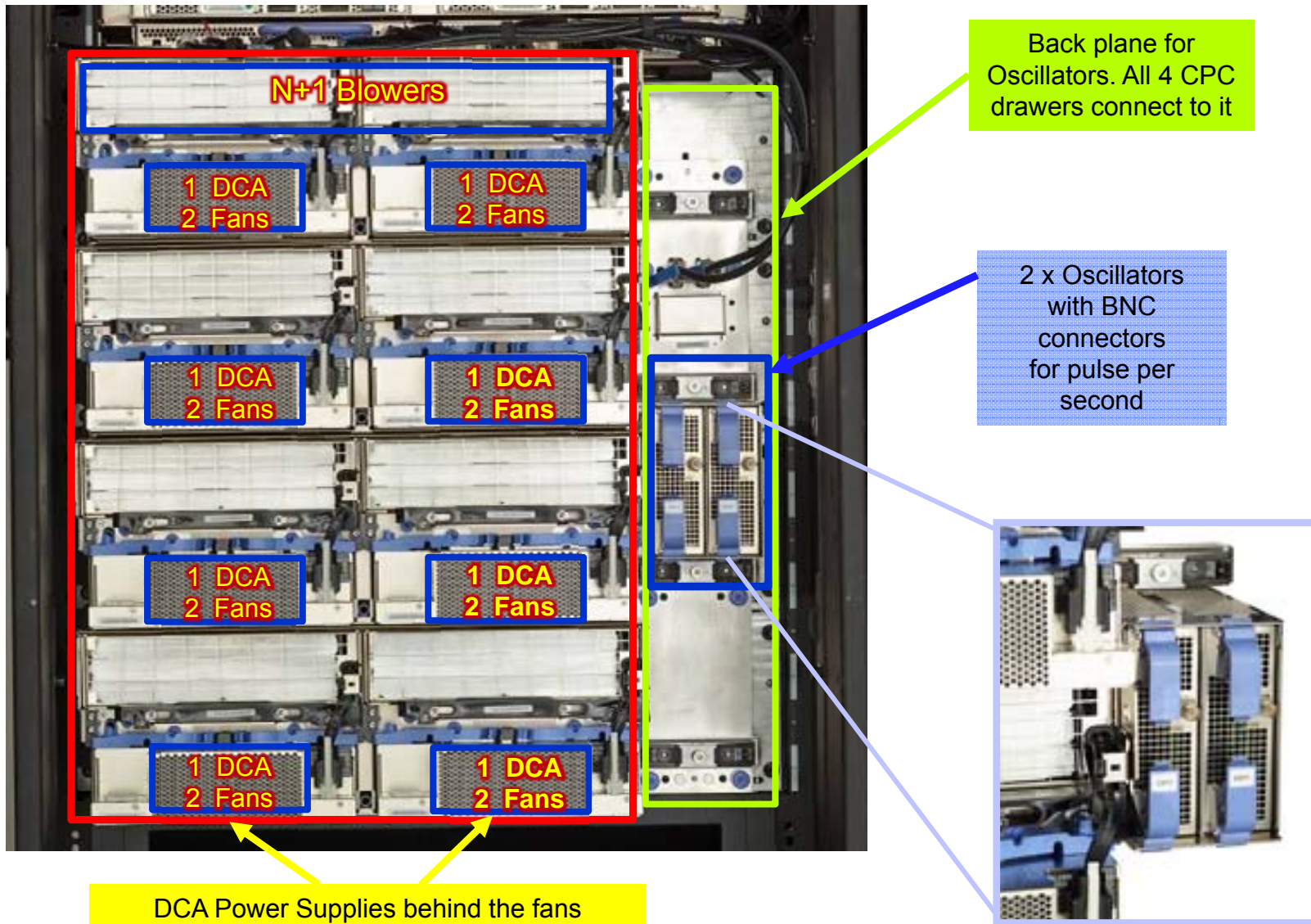


- **Two physical nodes, left and right**
- **Each logical node:**
  - Three PU chips
  - One SC chip (480 MB L4 cache)
  - Three Memory Controllers:
    - One per PU Chip
  - Five DDR3 DIMM slots per Memory Controller: 15 total per logical node
- **Each drawer:**
  - Six PU Chips: 39 active PUs (42 in z13 Model NE1)
  - Two SC Chips (960 MB L4 cache)
  - Populated DIMM slots: 20 or 25 DIMMs to support up to 2,368 GB of addressable memory (3,200 GB RAIM)
  - Water cooling for PU chips
  - Two Flexible Support Processors
  - Ten fanout slots for PCIe I/O drawer fanouts or PCIe coupling fanouts
  - Four fanout slots for IFB I/O drawer fanouts or PSIFB coupling link fanouts

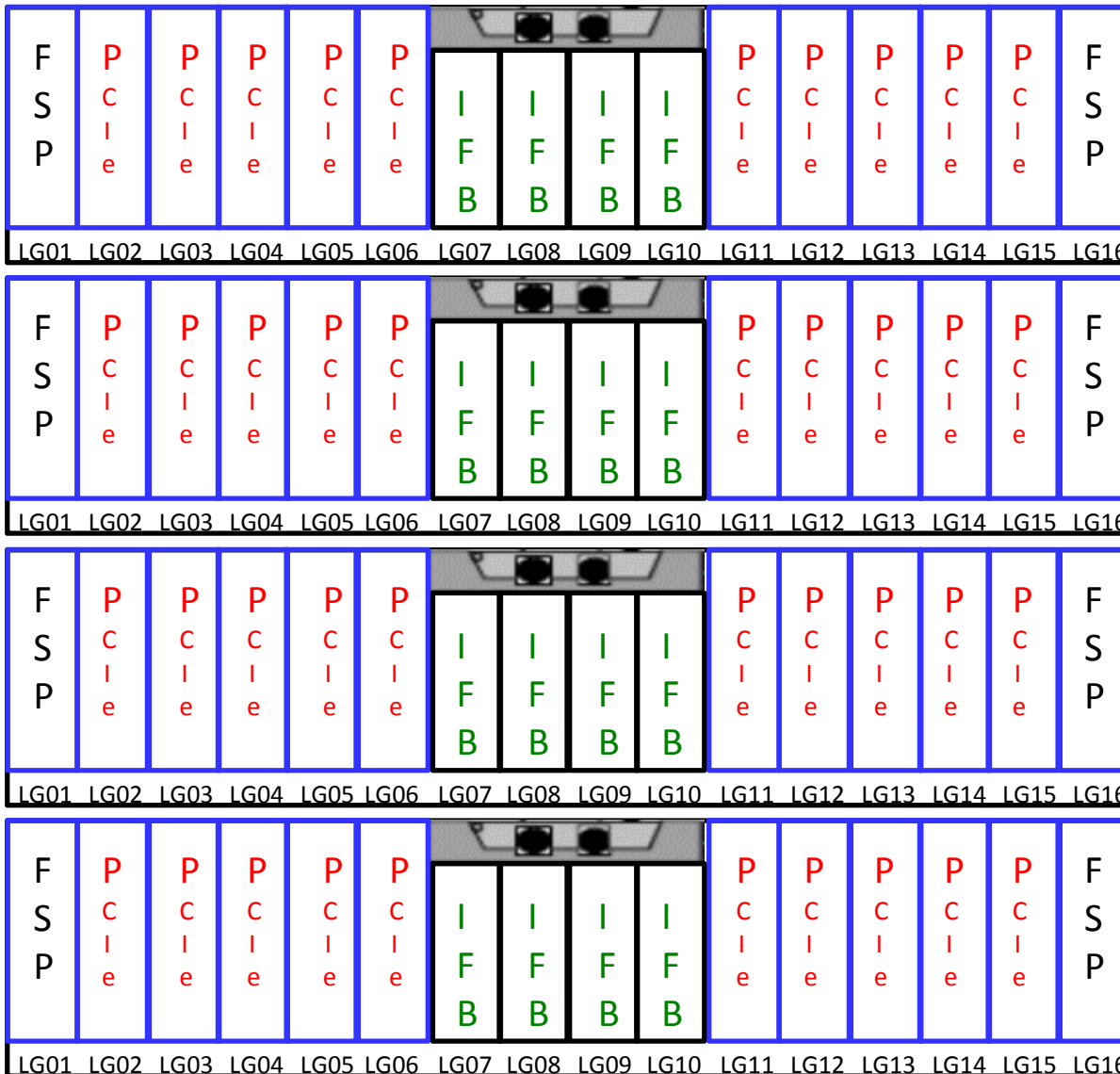
# z13 – 4 CPC Drawer System Details – Front View



# z13 – 4 CPC Drawer System Details – Rear View



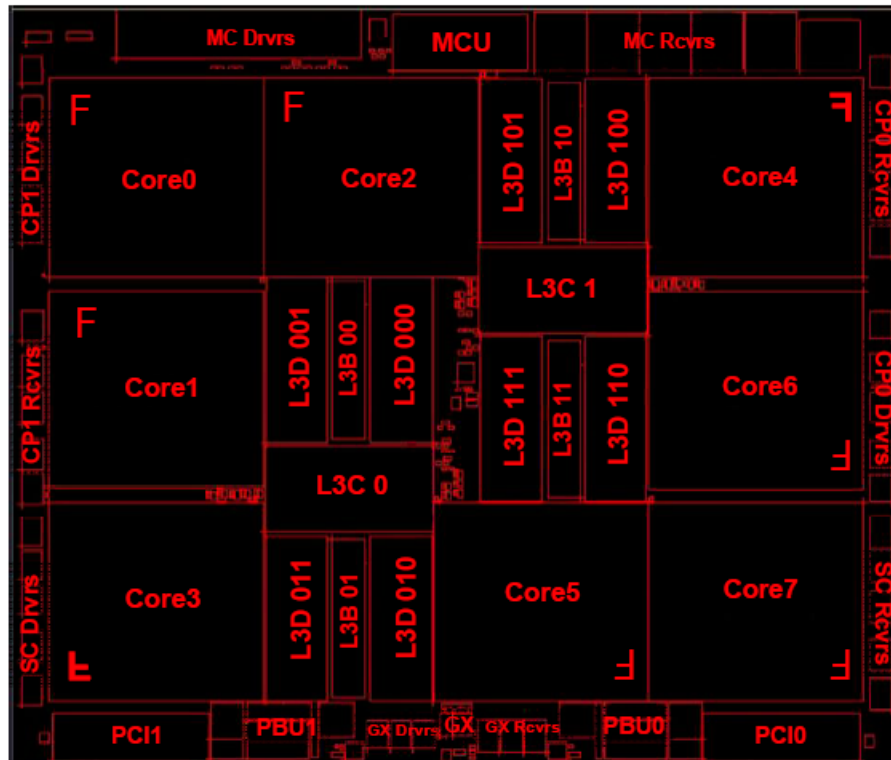
# z13 – 4 CPC Drawer - additional details – Front View



- Flexible Support Processor (FSP)
- PCIe Fanout Slots:
  - Total of 10 per CPC Drawer
  - Used for either connectivity of cPC Drawer to PCIe I/O Drawer or ICA SR coupling links
- IFB Fanout Slots:
  - Total of 4 per CPC Drawer
  - Used for HCA2-C fanout for connectivity to I/O Drawer. Maximum of ONLY 2 HCA2-Cs fanouts supported per CPC
  - Used for HCA3-O (12x) or HCA2-O LR (1x) PSIFB coupling links



# IBM z13 8-Core Processor Chip Detail



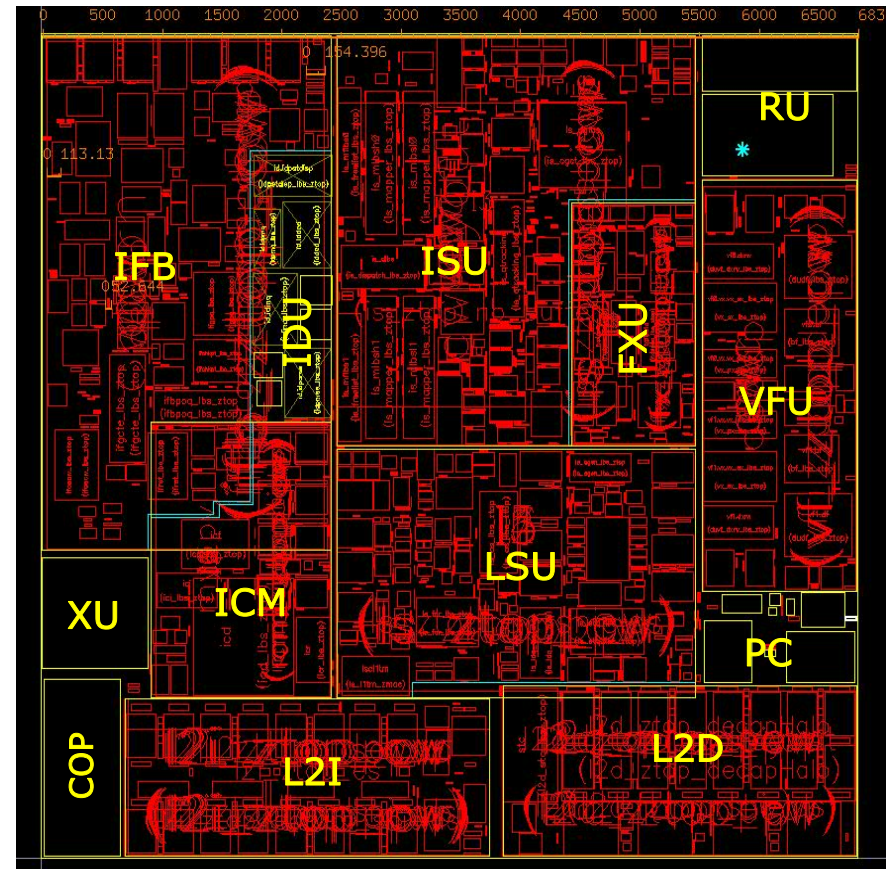
- 14S0 22nm SOI Technology
  - 17 layers of metal
  - 3.99 Billion Transistors
  - 13.7 miles of copper wire
- Chip Area
  - 678.8 mm<sup>2</sup>
  - 28.4 x 23.9 mm
  - 17,773 power pins
  - 1,603 signal I/Os

- Up to eight active cores (PUs) per chip
  - **5.0 GHz** (v5.5 GHz zEC12)
  - L1 cache/ core
    - 96 KB I-cache
    - 128 KB D-cache
  - L2 cache/ core
    - 2M+2M Byte eDRAM split private L2 cache
- Single Instruction/Multiple Data (SIMD)
- Single thread or 2-way simultaneous multithreaded (SMT) operation
- Improved instruction execution bandwidth:
  - Greatly improved branch prediction and instruction fetch to support SMT
  - Instruction decode, dispatch, complete increased to 6 instructions per cycle\*
  - Issue up to 10 instructions per cycle\*
  - Integer and floating point execution units
- On chip 64 MB eDRAM L3 Cache
  - Shared by all cores
- I/O buses
  - One GX++ I/O bus
  - Two PCIe I/O buses
- Memory Controller (MCU)
  - Interface to controller on memory DIMMs
  - Supports RAIM design

\* zEC12 decodes 3 instructions and executes 7

## z13 Processor Overview

- **2X Instruction pipe width**
  - Improves IPC for all modes
  - Symmetry simplifies dispatch/issue rules
  - Required for effective SMT
- **Added FXU and BFU execution units**
  - 4 FXUs
  - 2 BFUs, DFUs
  - 2 new SIMD units
- **SIMD unit plus additional registers**
- **Pipe depth re-optimized for power/performance**
  - Product frequency reduced
  - Processor **performance** increased
- **SMT support**
  - Wide, symmetric pipeline
  - Full architected state per thread
  - SMT-adjusted CPU usage metering

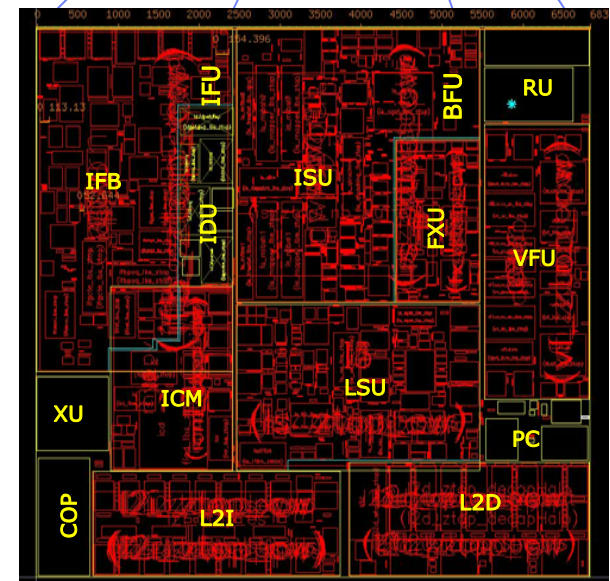
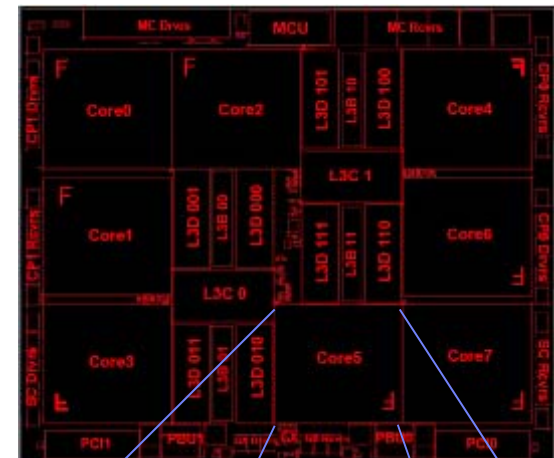


CP Chip Floorplan

# PU Chip/Core Execution Unit Details

- Each processor unit, or core, is a superscalar and out-of-order processor that has ten execution units:
  - **Fixed-point unit (FXU):** The FXU handles fixed-point arithmetic
  - **Load-store unit (LSU):** The LSU contains the data cache. It is responsible for handling all types of operand accesses of all lengths, modes, and formats as defined in the z/Architecture
  - **Binary floating-point unit (BFU):** The BFU handles all binary and hexadecimal floating-point and fixed-point multiplication operations
  - **Decimal floating-point unit (DFU):** The DU runs both floating-point and decimal fixed-point operations and fixed-point division operations.
  - **Instruction fetch and branch (IFB) (prediction) and Instruction cache & merge (ICM).** These two sub units (IFB and ICM) contain the instruction cache, branch prediction logic, instruction fetching controls, and buffers. Its relative size is the result of the elaborate branch prediction
  - **Instruction decode unit (IDU):** The IDU is fed from the IFU buffers, and is responsible for parsing and decoding of all z/Architecture operation codes
  - **Translation unit (XU):** The XU has a large **translation lookaside buffer (TLB)** and the Dynamic Address Translation (DAT) function that handles the dynamic translation of logical to physical addresses
  - **Instruction sequence unit (ISU):** This unit enables the out-of-order (OoO) pipeline. It tracks register names, OoO instruction dependency, and handling of instruction resource dispatch
  - **Instruction fetching unit (IFU) (prediction):** These units contain the instruction cache, branch prediction logic, instruction fetching controls, and buffers. Its relative size is the result of the elaborate branch prediction design.
  - **Recovery unit (RU):** The RU keeps a copy of the complete state of the system that includes all registers, collects hardware fault signals, and manages the hardware recovery actions.
  - **Dedicated Co-Processor (COP):** The dedicated coprocessor is responsible for data compression and encryption functions for each core
  - **Core pervasive unit (PC)** for instrumentation, error collection
  - **Vector and Floating point Units (VFU)**
    - Fixed-point unit (FXU): The FXU handles fixed-point arithmetic
    - Binary floating-point unit (BFU): The BFU handles all binary and hexadecimal floating-point and fixed-point multiplication operations.
    - Decimal floating-point unit (DFU): The DU runs both floating-point and decimal fixed-point operations and fixed-point division operations.
    - Vector execution unit (VXU)

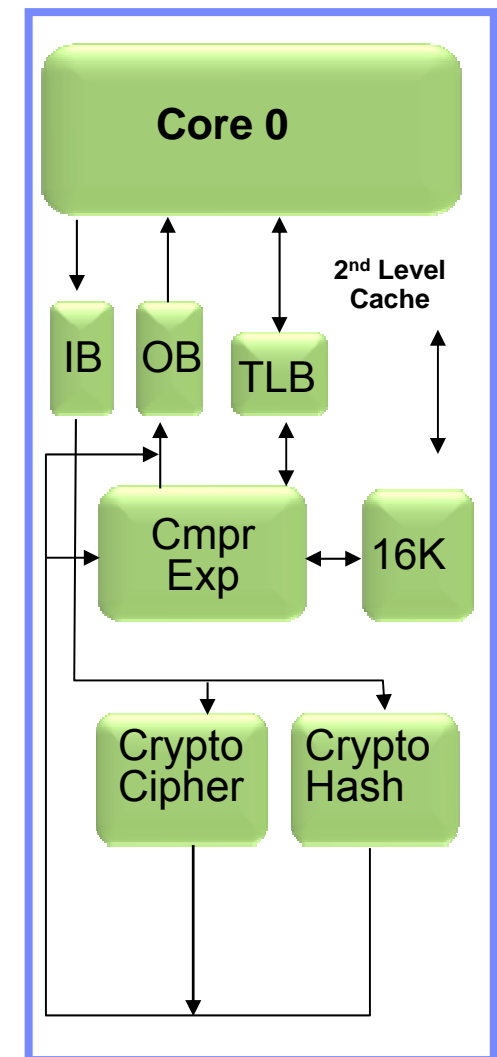
PU *Chip* Floorplan



PU *Core* Floorplan

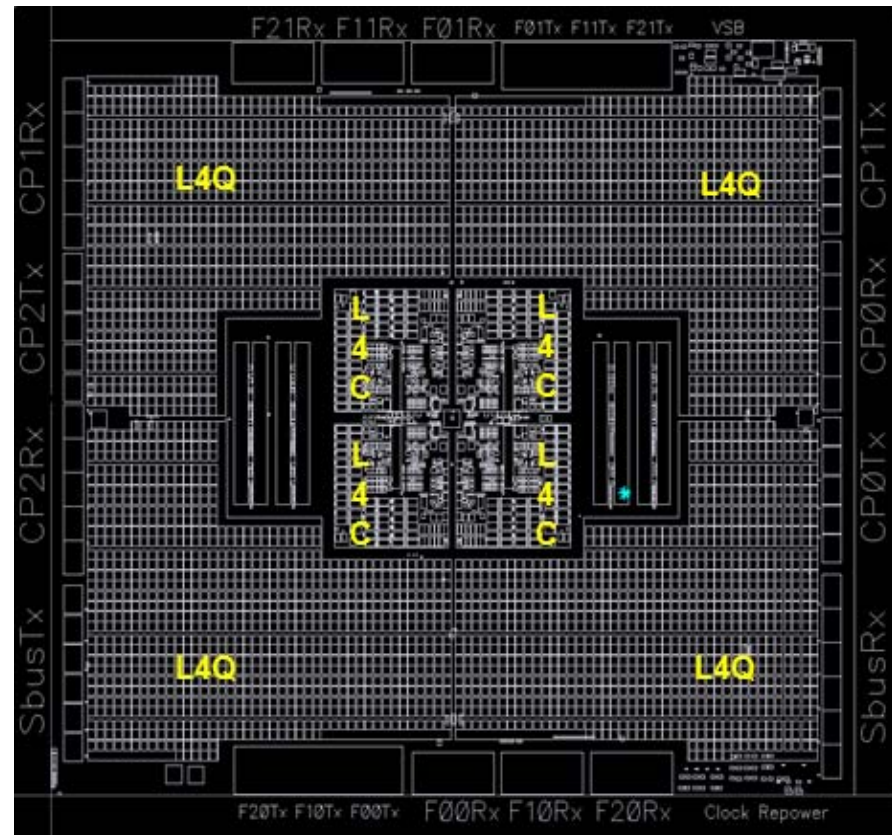
## z13 Compression and Cryptography Accelerator

- **Coprocessor dedicated to each core**  
(Was shared by two cores on z196)
  - Independent compression engine
  - Independent cryptographic engine
  - Available to any processor type (CP, zIIP, IFL)
  - Owning processor is busy when its coprocessor is busy
  - Instructions available to any processor type
- **Data compression/expansion engine**
  - Static dictionary compression and expansion
- **CP Assist for Cryptographic Function**
  - Supported by z/OS, z/VM, z/VSE, z/TPF, and Linux on z System
    - DES, TDES – Clear and Protected Key
    - AES128, 192, 256 – Clear and Protected Key
    - SHA-1 (160 bit) – Clear Key
    - SHA-256, -384, -512 – Clear Key
    - PRNG – Clear Key
    - DRNG – Clear Key
    - CPACF FC 3863 (No Charge – Export Control) is required to enable some functions and to support Crypto Express5S



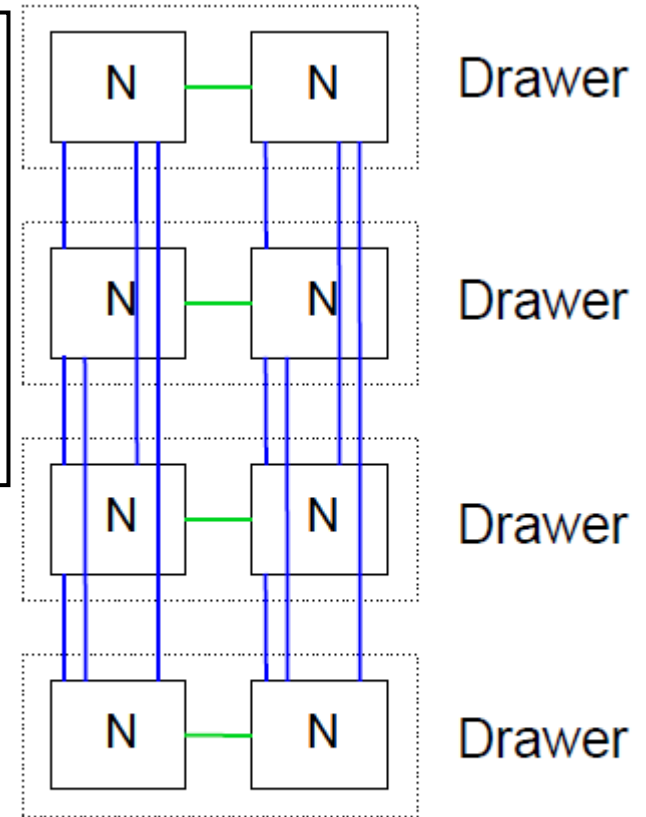
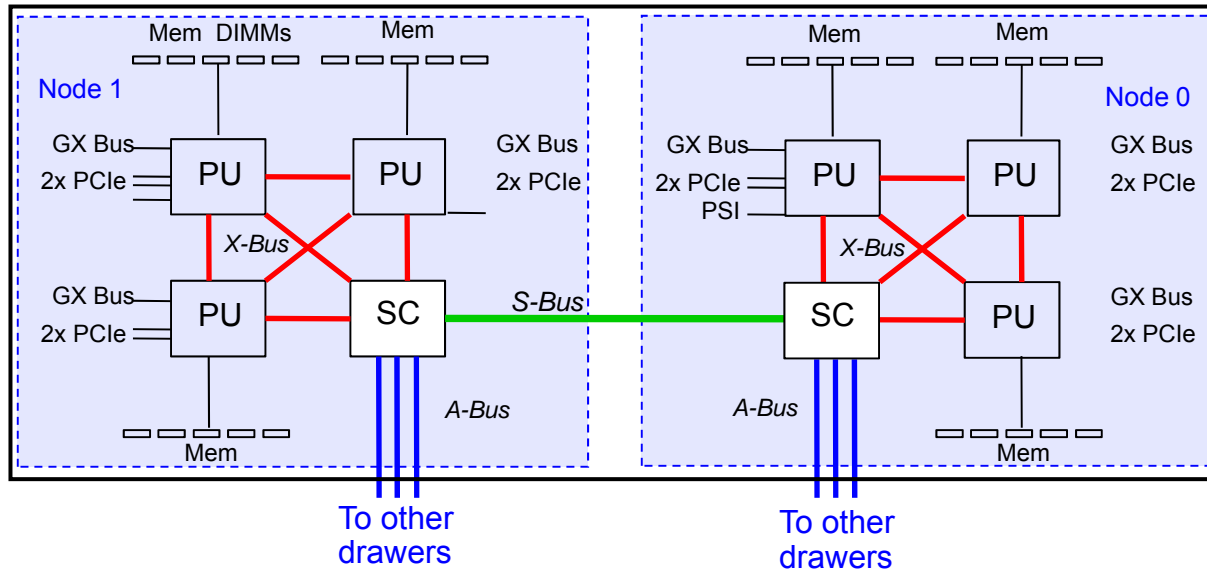
## z13 Storage Control (SC) Chip Detail

- **CMOS 14S0 22nm SOI Technology**
  - 15 Layers of metal
  - 7.1 Billion transistors
  - 12.4 Miles of copper wire
- **Chip Area –**
  - 28.4 x 23.9 mm
  - 678 mm<sup>2</sup>
  - 11,950 power pins
  - 1,707 Signal Connectors
- **eDRAM Shared L4 Cache**
  - 480 MB per SC chip (Non-inclusive)
  - 224 MB L3 NIC Directory
  - 2 SCs = 960 MB L4 per z13 drawer
- **Interconnects (L4 – L4)**
  - 3 to CPs in node
  - 1 to SC (node – node) in drawer
  - 3 to SC nodes in remote drawers
- **6 Clock domains**



# z13 Drawer Structure and Interconnect

Fully Populated Drawer

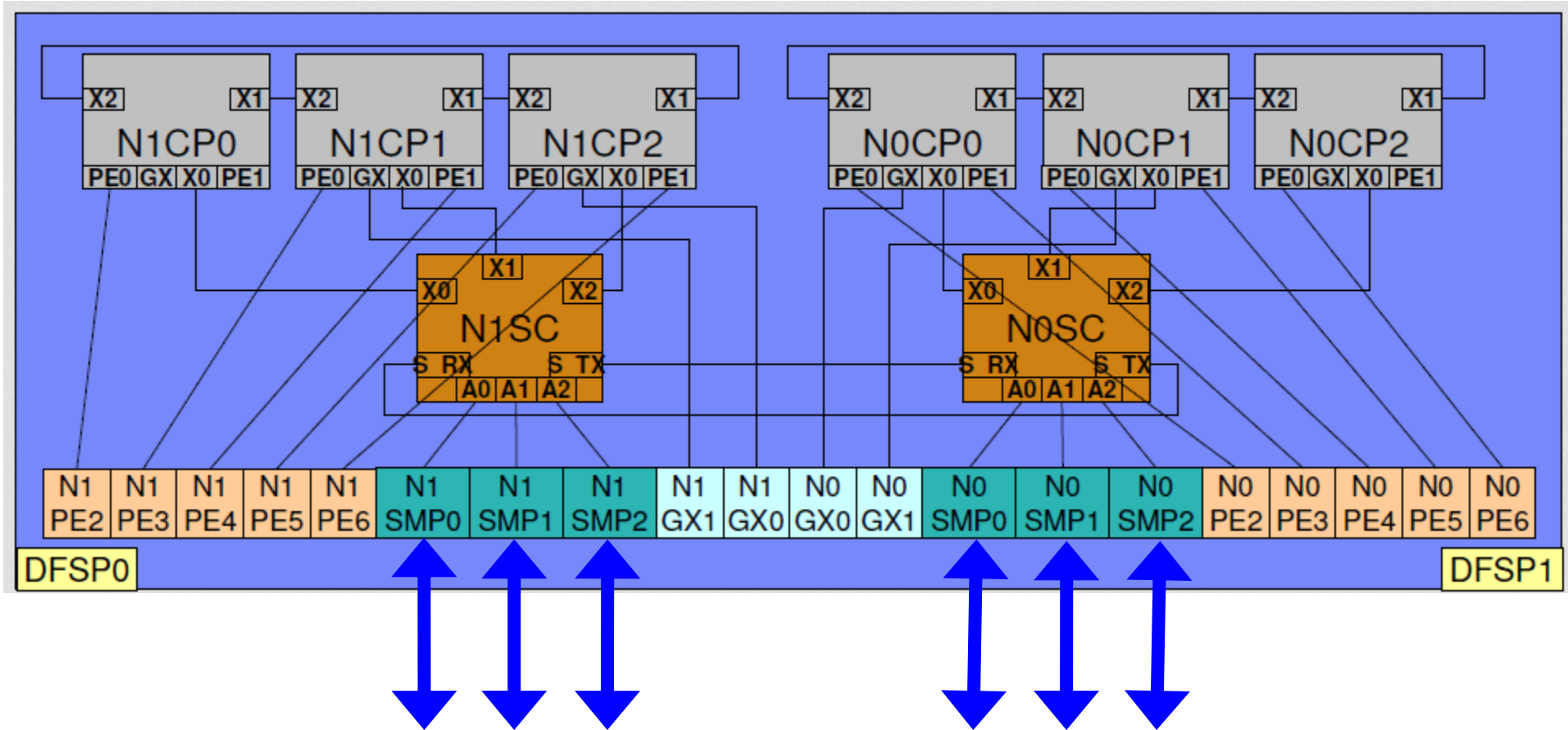


4 Drawer System Interconnect

## Physical nodes: (Two per drawer)

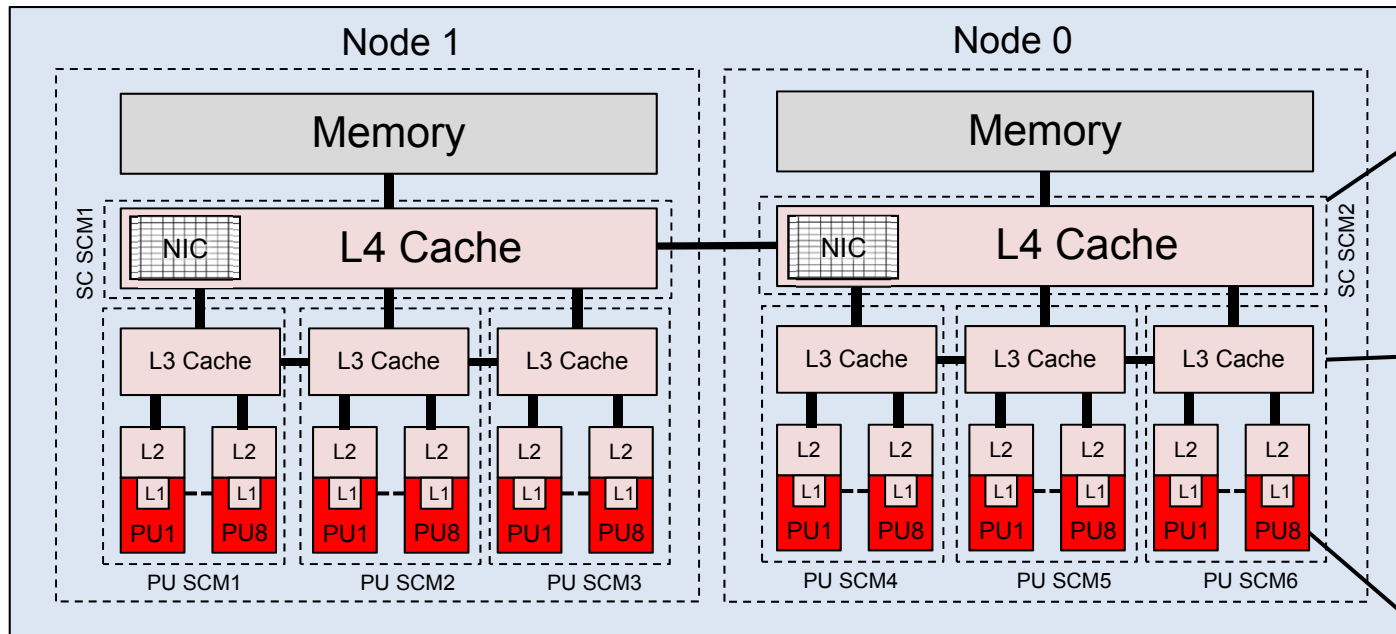
- **Chips**
  - Three PU chips
  - One SC chip (480 MB L4 cache)
- **RAIM Memory**
  - Three Memory Controllers: One per PU Chip
  - Five DDR3 DIMM slots per Controller: 15 total per logical node
  - Populated DIMM slots: 20 or 25 per drawer
- **SC and CP Chip Interconnects**
  - **X-bus: SC and CPs to each other (same node)**
  - **S-bus: SC to SC chip in the same drawer**
  - **A-bus: SC to SC chips in the remote drawers**

# z13 Drawer Logic Interconnect



# z13 CPC Drawer Cache Hierarchy Detail

Single CPC Drawer View (N30 Model) – 2 Nodes



2 SC SCM Chips  
(1 per node)



6 PU SCM Chips  
(3 per node)



\* Up to 8 PU cores per chip



Single PU core

\* Not all PU's active

### Node 1 - Caches

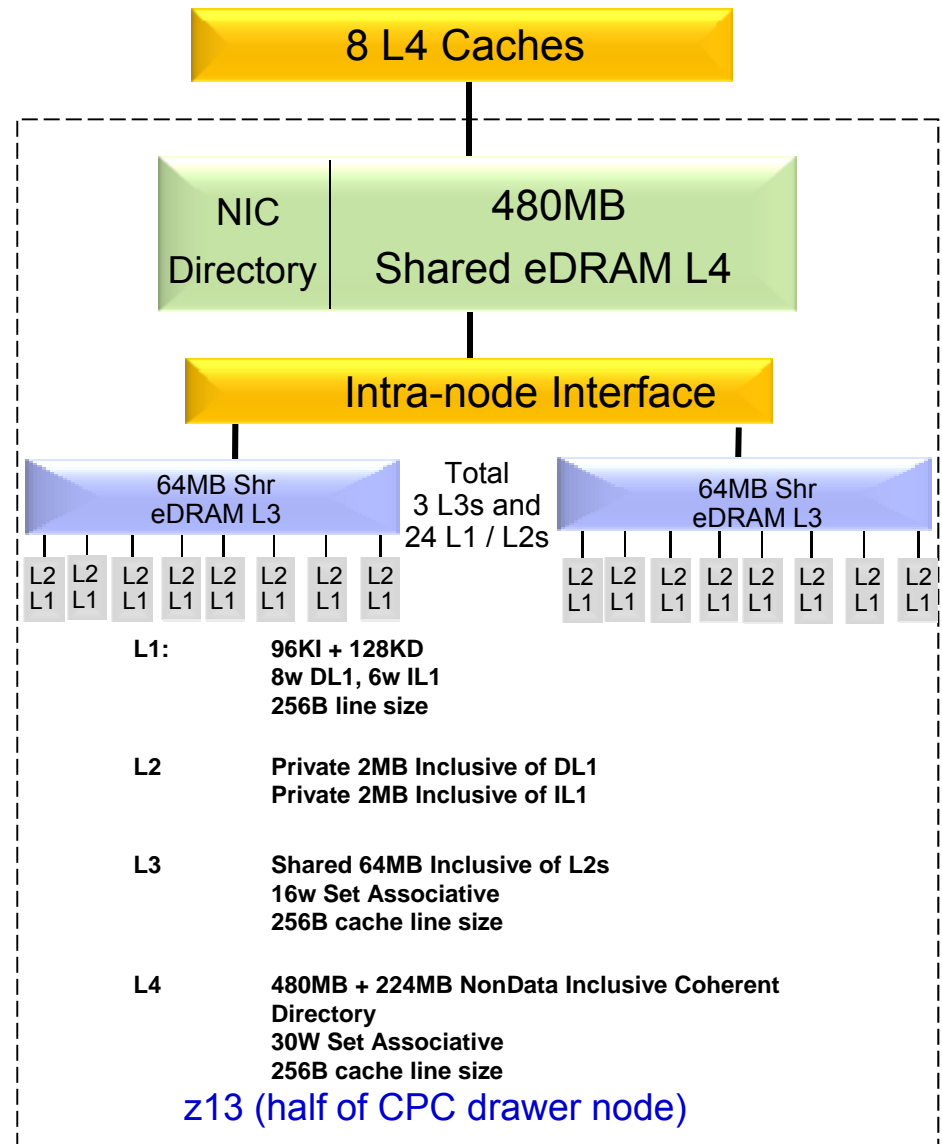
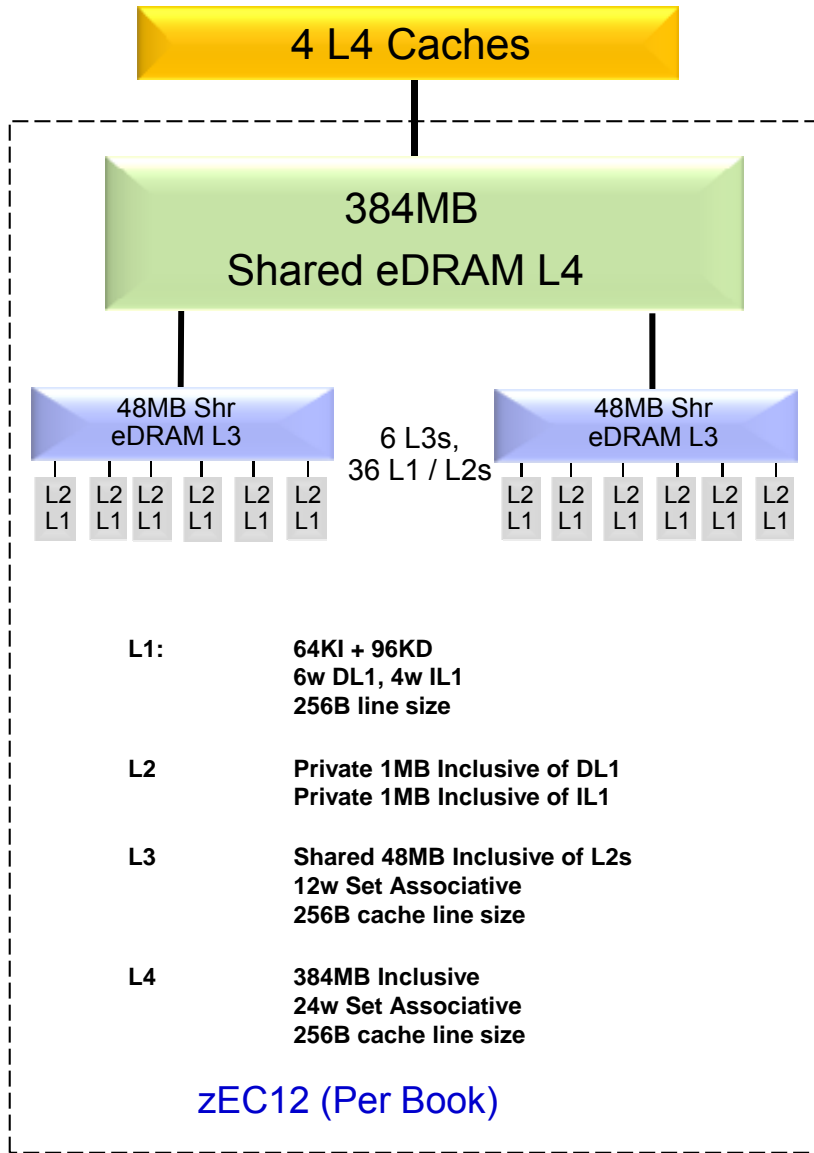
- L1 private 96k i, 128k d
- L2 private 2 MB i + 2 MB d
- L3 shared 64 MB / chip
- L4 shared 480 MB / node  
- plus 224 MB NIC

### Node 0 - Caches

- L1 private 96k i, 128k d
- L2 private 2 MB i + 2 MB d
- L3 shared 64 MB / chip
- L4 shared 480 MB / node  
- plus 224 MB NIC

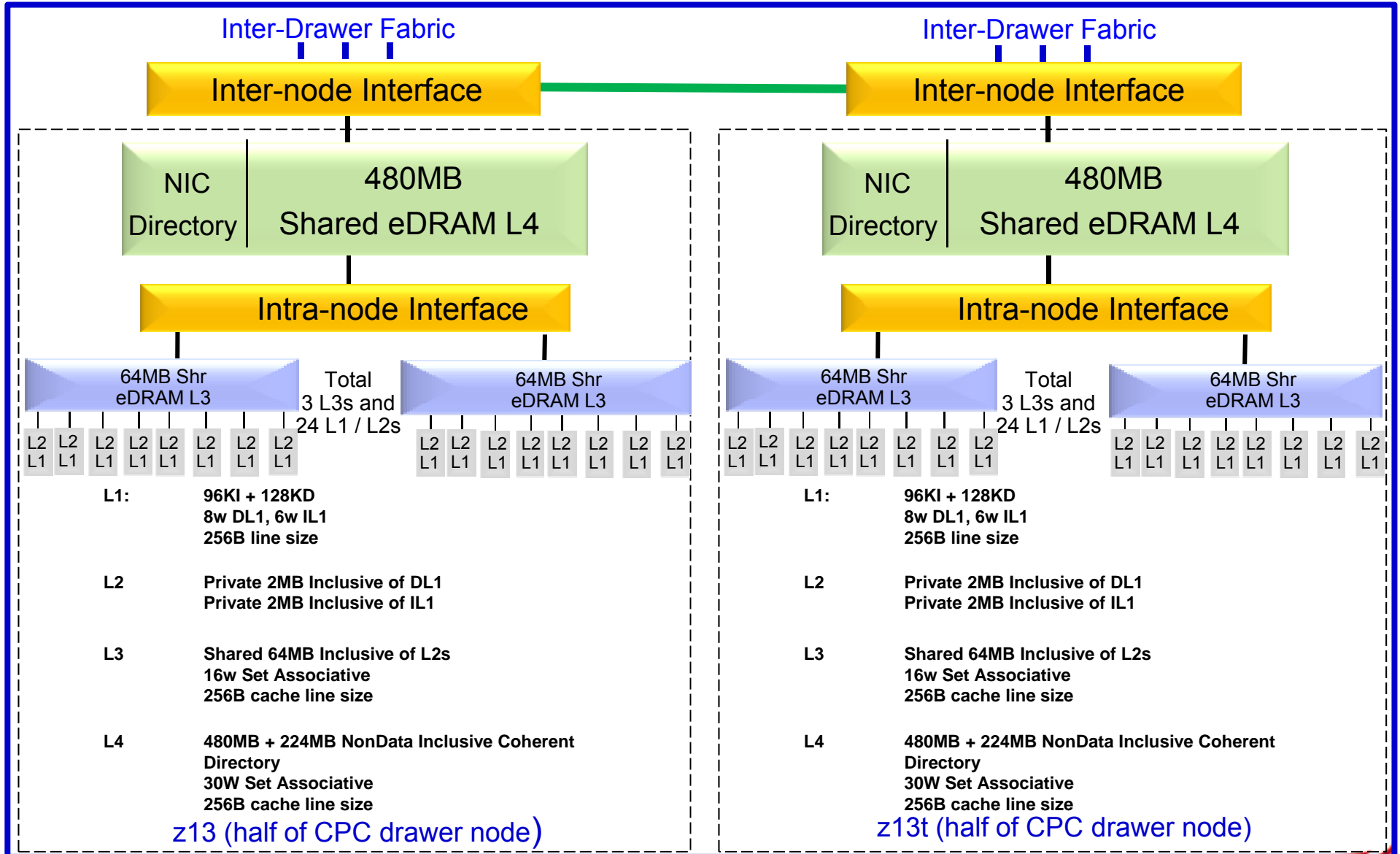


# z System Cache Topology – zEC12 vs. z13 Comparison



# z13 Drawer Nodes

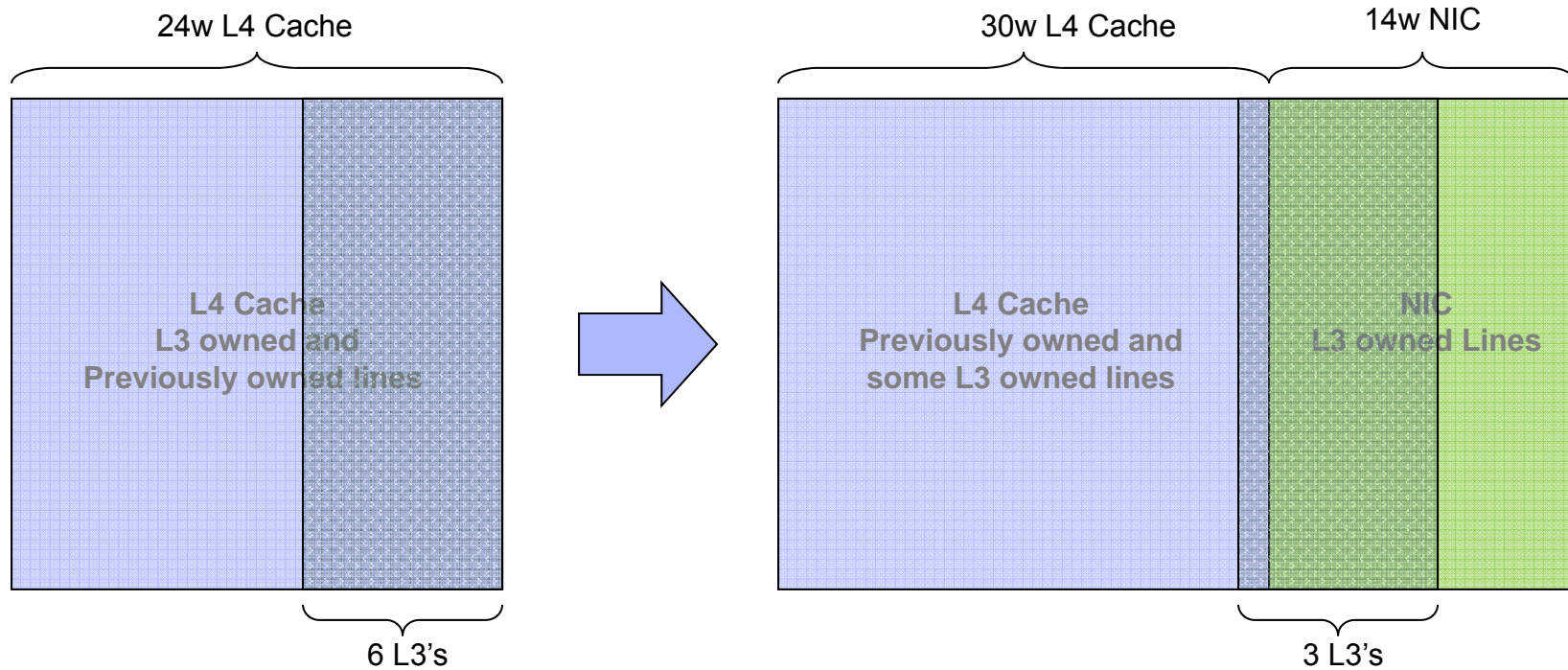
## Intra-Node Snoop Interface and Inter-Node Snoop Interface



## z13 L4 Cache Design with Non-Data Inclusive Coherent (NIC) Directory

**zEC12 Inclusive L4 Design**  
**192 MB + 192 MB per Book**

**z13 New Inclusive L4 Design**  
**480 MB L4 with 224 MB NIC Directory**  
**(Two nodes per drawer)**

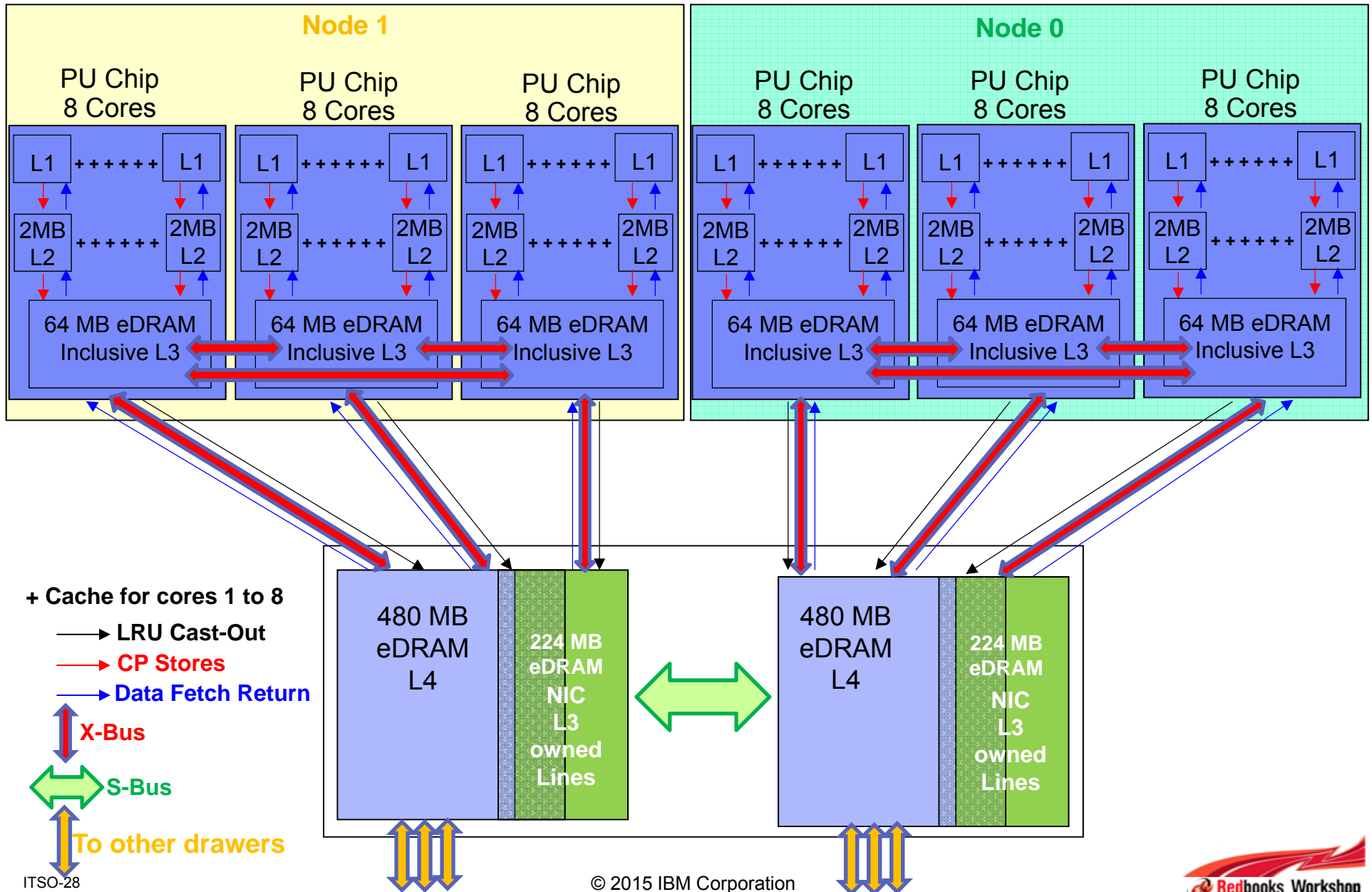


L3 locally owned lines can be accessed over the X-bus L3 – L3 using the **Intra-Node** Snoop Interface without being included in L4. **Inter-Node** snoop traffic to L4 can still be handled effectively.

Traditional inclusive approach yields nest cache size at L4's 480MB

With NIC, nest cache size can potentially be sum of the 3 L3's and L4 (3X64MB + 480MB)

# z13 Cache Topology



## Cache Comparison against POWER & Intel

	<b>Intel IvyBridgeEP 22nm (3Q'13) 3.3GHz Core 10-core Die</b>	<b>POWER p7+ 32nm 4.4GHz Core 8-core Die</b>	<b>zEC12 32nm 5.5GHz Core 6-core Die</b>	<b>z13 22nm 5.0GHz Core 8-core Die</b>
<b>L1</b>	Private Split I+D 32KB I\$, 32KB D\$	Private Split I+D 32KB I\$, 32KB D\$	Private Split I+D 64KB I\$, 96KB D\$	Private Split I+D 64KB I\$, 96KB D\$
<b>L2</b>	Private 256KB	Private 256KB	Private Split I+D 1MB L1+, 1MB D\$	Private Split I+D 2MB I\$, 2MB D\$
<b>L3</b>	Distributed & Shared NUCA 30MB	Private with ECO NUCA 80MB	Shared UCA 48MB	Shared UCA 64MB
<b>L4</b>	-	-	Shared UCA 384MB	Shared with NIC UCA L4 480MB NIC 224 MB

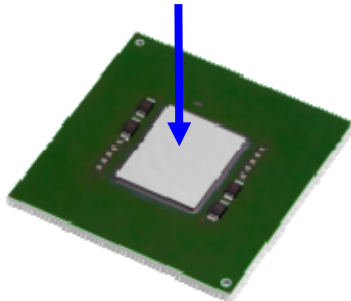
(N)UCA = (Non-) Uniform Cache Access

# z13 SCM Vs zEC12 MCM Comparison

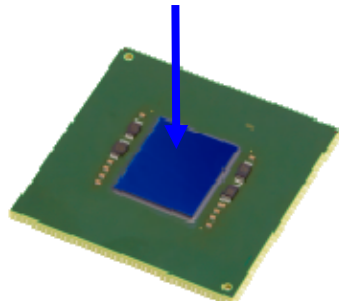
## z13 Single Chip Modules (SCMs)

- Processor Unit (PU) SCM
  - 68.5mm x 68.5mm – fully assembled
  - PU Chip area 678 mm<sup>2</sup>
  - Eight core chip with 6, 7 or 8 active cores
- System Controller (SC) SCM
  - 68.5mm x 68.5mm – fully assembled
  - SC Chip area 678 mm<sup>2</sup>
  - 480 MB on-inclusive L4 cache per SCM
  - [Non-Data Integrated Coherent \(NIC\) Directory for L3](#)
- Processor Drawer – Two Nodes
  - Six PU SCMs for 39 PUs (42 PUs in Model NE1)
  - Two SC SCMs (960 MB L4)
  - N30: One Drawer, N63: Two Drawers, N96: Three Drawers, NC9 or NE1: Four Drawers

Single PU Chip without Module Thermal Cap

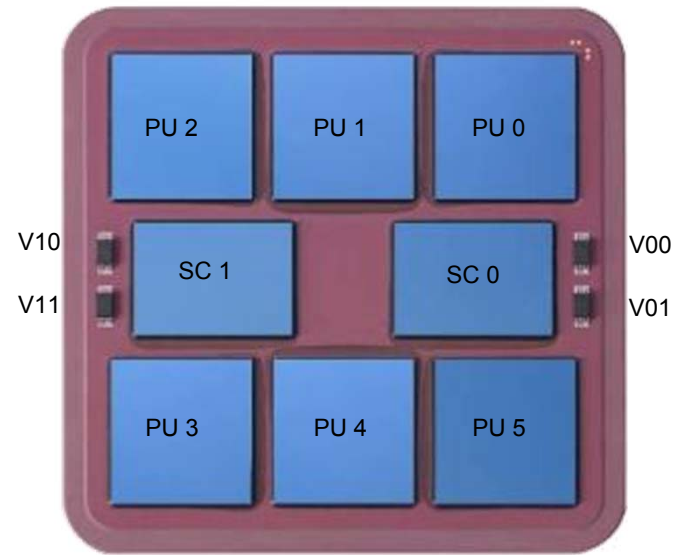


Single SC Chip without Module Thermal Cap



## zEC12 Multi Chip Module (MCM)

- Technology
  - 96mm x 96mm with 102 glass ceramic layers
  - 7,356 LGA connections to 8 chip sites
- Six 6-core Processor (PU) chips
  - Each with 4, 5 or 6 active cores
  - 27 active processors per MCM (30 in Model HA1)
  - PU Chip size 23.7 mm x 25.2 mm
- Two System Controller (SC) chips per MCM
  - 192 MB L4 cache per SC, 384 MB per MCM
  - SC Chip size 26.72 mm x 19.67 mm
- One MCM per book, up to 4 books per System



## z13 Processor Unit Allocation/Usage – zIIP to CP 2:1 ratio

Model	Drawers /PUs	CPs	IFLs uIFLs	zIIPs	ICFs	Std SAPs	Optional SAPs	Std. Spares	IFP
<b>N30</b>	1/39	0-30	0-30 0-29	0-20	0-30	6	0-4	2	1
<b>N63</b>	2/78	0-63	0-63 0-62	0-42	0-63	12	0-8	2	1
<b>N96</b>	3/117	0-96	0-96 0-95	0-64	0-96	18	0-12	2	1
<b>NC9</b>	4/156	0-129	0-128 0-127	0-86	0-129	24	0-16	2	1
<b>NE1</b>	4/168	0-141	0-141 0-140	0-94	0-141	24	0-16	2	1

- z13 Models N30 to NC9 use drawers with 39 cores. The Model NE1 has 4 drawers with 42 cores.
- The maximum number of logical ICFs or logical CPs supported in a CF logical partition is 16
- The integrated firmware processor (IFP) is used for PCIe I/O support functions
- Concurrent Drawer Add is available to upgrade in steps from model N30 to model NC9
  1. At least one CP, IFL, or ICF must be purchased in every machine
  2. Two zIIPs may be purchased for each CP purchased if PUs are available. This remains true for sub-capacity CPs and for “banked” CPs.
  3. On an upgrade from z196 or zEC12, installed zAAPs are converted to zIIPs by default. (Option: Convert to another engine type)
  4. “uIFL” stands for Unassigned IFL
  5. The IFP is conceptually an additional, special purpose SAP

## Processor Unit (Core) Locations: Customer, SAP, IFP and Spare

z13		1 <sup>st</sup> Drawer				2 <sup>nd</sup> Drawer				3 <sup>rd</sup> Drawer				4 <sup>th</sup> Drawer			
Model	Cust PUs	Cust PUs	SAPs	IFP	Spare	Cust PUs	SAPs	IFP	Spare	Cust PUs	SAPs	IFP	Spare	Cust PUs	SAPs	IFP	Spare
NE1	141	34	6	1	1	35	6	0	1	36	6	0	0	36	6	0	0
NC9	129	31	6	1	1	32	6	0	1	33	6	0	0	33	6	0	0
N96	96	31	6	1	1	32	6	0	1	33	6	0	0				
N63	63	31	6	1	1	32	6	0	1								
N30	30	30	6	1	2												

- ▶ PUs can be purchased as CPs, IFLs, Unassigned IFLs, zIIPs, ICFs or Additional SAPs
  - zAAPs no longer available fulfilling the statement of direction
  - zIIP to CP purchase ratio is 2:1
  - Additional SAPs + Permanent SAPs may not exceed 32
  - Any un-configured PU can act as an additional Spare PU
  - CPs and zIIPs initial placement in 1<sup>st</sup> drawer working up
  - IFLs and ICFs initial placement in highest drawer working down
  
- ▶ Upgrades available from any lower model to any higher any models
  - Achieved via [Concurrent Drawer Add](#) from model N30 to model NC9
  - Achieved via combination of [Concurrent Drawer Add](#) and drawer replacement to model NE1



## IBM z Integrated Information Processor (zIIP) on the z13

- The IBM z13 continues to support the z Integrated Information Processor (zIIP) which can take advantage of the optional simultaneous multithreading (SMT) technology capability. SMT allows up to two active instruction streams per core, each dynamically sharing the core's execution resources
  - With the multithreading function enabled, the performance capacity of the zIIP processor is expected to be up to 1.4 times the capacity of these processors on the zEC12
- The rule for the CP to zIIP purchase ratio is that for every CP purchased, up to two zIIPs may be purchased
- zAAP eligible workloads such as Java and XML, can run on zIIPs using zAAP on zIIP processing
- zAAPs are no longer supported on the z13

## Reminder - What workloads are eligible to run on zIIPs?

### Enabled technologies

- Centralized data serving eligible for zIIP: Portions of BI, ERP, and CRM remote connectivity to DB2 V8, as well as portions of long running parallel queries. and select utilities
- Network encryption on zIIP – IPsec network encryption/ decryption (with z/OS V1.8)
- XML parsing – z/OS XML System Services eligible on zAAP or zIIP (with z/OS V1.9, V1.8 and V1.7 with maintenance)
- Remote mirror – zIIP-assisted z/OS Global Mirror function (with z/OS V1.9)
- HiperSockets™ – HiperSockets Multiple Write operation for outbound large messages (with z/OS V1.9)
- Business Intelligence – IBM Scalable Architecture for Financial Reporting provides a high-volume, high performance reporting – can be eligible for zIIP processing.
- Intra-server communications – z/OS CIM Server processing eligible for zIIP (with z/OS V1.11).
- DB2 sort utility – DB2 utilities sorting fixed-length records using IBM's memory object sorting technique
- “zAAP on zIIP” capability – Optimize the purchase of a new zIIP or maximize your investment in existing zIIPs
- Select Tivoli® products – for DASD scans and Performance Expert/ Performance Monitor
- Select RMF™ processing – (z/OS V2.1) small portion of RMF monitoring eligible for zIIP
- Java – for WebSphere® Application Server and Java technology-based applications
- Select XML System services workloads
- Select ISV applications

Note: Levels shown represent the required minimum levels

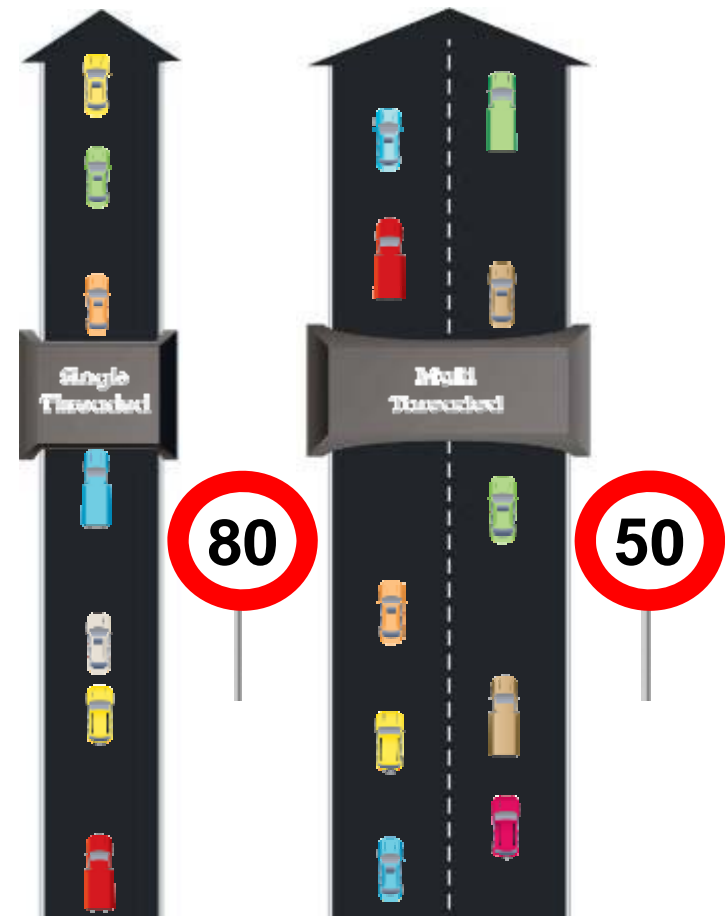
## IBM z13 z/Architecture Extensions

- **Simultaneous multithreaded (SMT) operation**
  - Up to two active execution threads per core can dynamically share the caches, TLBs and execution resources of each IFL and zIIP core. SMT is designed to improve both core capacity and single thread performance significantly.
  - PR/SM online logical processors dispatches physical cores; but, an operating system with SMT support can be configured to dispatch work to a thread on an IFL or zIIP core in single thread or SMT mode so that HiperDispatch cache optimization is considered. (Zero, one or two threads can be active in SMT mode). Enhanced hardware monitoring support will measure thread usage and capacity.
- **Core micro-architecture radically altered to increase parallelism**
  - New branch prediction and instruction fetch front end to support SMT and to improve branch prediction throughput.
  - Wider instruction decode, dispatch and completion bandwidth:  
Increased to six instructions per cycle compared to three on zEC12
    - Decodes 6, executes 10 (zEC12 decodes 3, executes 7)
  - Larger instruction issue bandwidth: Increased to up to 10 instructions issued per cycle (2 branch, 4 FXU, 2 LSU, 2 BFU/DFU/SIMD) compared to 7 on zEC12
  - Greater integer execution bandwidth: Four FXU execution units
  - Greater floating point execution bandwidth: Two BFUs and two DFUs; improved fixed point and floating point divide
- **Single Instruction Multiple Data (SIMD) instruction set and execution: Business Analytics Vector Processing**
  - Data types: Integer: byte to quad-word; String: 8, 16, 32 bit; binary floating point
  - New instructions (139) include string operations, vector integer and vector floating point operations: two 64-bit, four 32-bit, eight 16-bit and sixteen 8-bit operations.
  - Floating Point Instructions operate on newly architected vector registers (32 new 128-bit registers). Existing FPRs overlay these vector registers.

# Simultaneous Multithreading (SMT)

- Simultaneous multithreading allows instructions from one or two threads to execute on a zIIP or IFL processor core
- SMT helps to address memory latency, resulting in an overall **capacity\*** (throughput) improvement per core
- Capacity improvement is variable depending on workload. For **AVERAGE** workloads the **estimated capacity\*** of a z13 zIIP/IFL with exploitation of the SMT option is:
  - zIIP is **38%** greater than a zEC12 zIIP
  - IFL is **32%** greater than a zEC12 IFL
  - zIIP is **72%** greater than a z196 zIIP
  - IFL is **65%** greater than a z196 IFL
- SMT exploitation: z/VM V6.3 + PTFs for IFLs and z/OS V2.1 + PTFs in an LPAR for zIIPs
- The use of SMT mode can be enabled on an LPAR by LPAR basis via operating system parameters.
  - When enabled, z/OS can transition dynamically between MT-1 (multi thread) and MT-2 modes with operator commands.
- Notes:
  1. SMT is designed to deliver better overall capacity (throughput) for many workloads. Thread **performance** (instruction execution rate for an individual thread) may be faster running in single thread mode.
  2. Because SMT is not available for CPs, LSPR ratings do not include it

\*Capacity and performance ratios are based on measurements and projections using standard IBM benchmarks in a controlled environment. Actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload .



*Which approach is designed for the highest volume\*\* of traffic?  
Which road is faster?*


\*\* Two lanes at 50 carry 25% more volume if traffic density per lane is equal

# SIMD (Single Instruction Multiple Data) processing

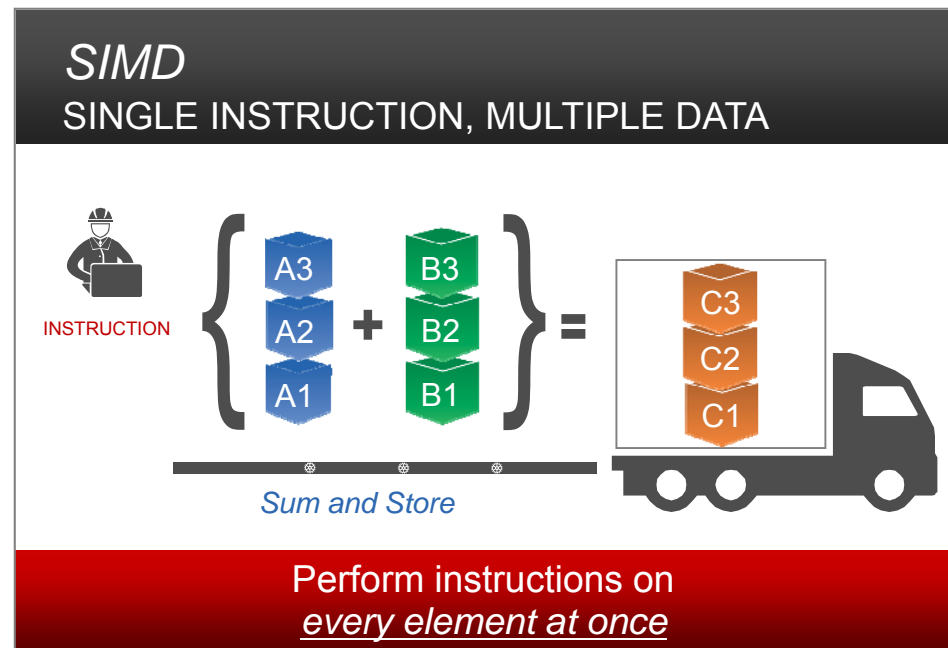
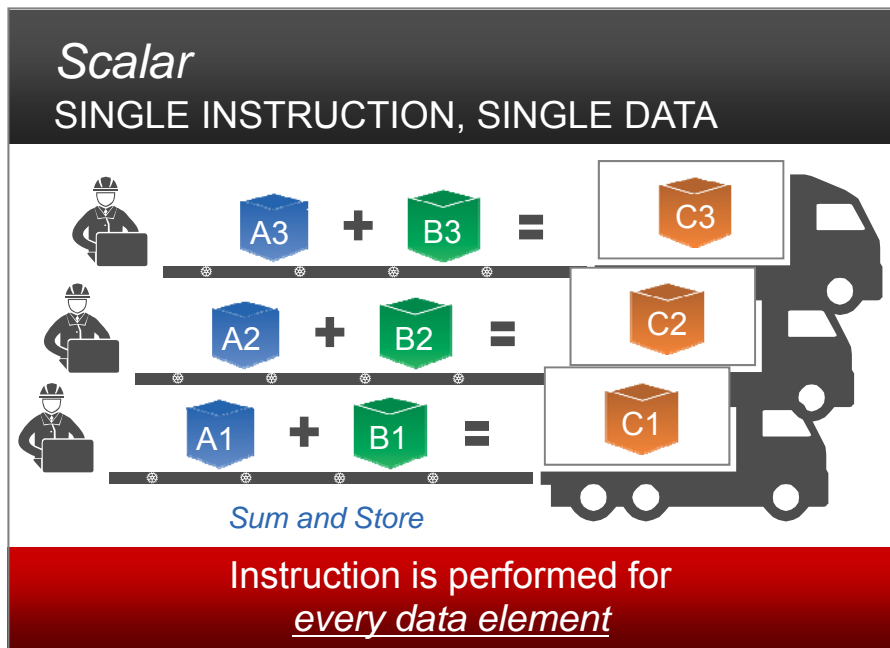


Increased parallelism to enable analytics processing

- Smaller amount of code helps improve execution efficiency
- Process elements in parallel enabling more iterations
- Supports analytics, compression, cryptography, video/imaging processing

 **Value**

- ✓ Enable new applications
- ✓ Offload CPU
- ✓ Simplify coding



# z13 Memory Design / Structure

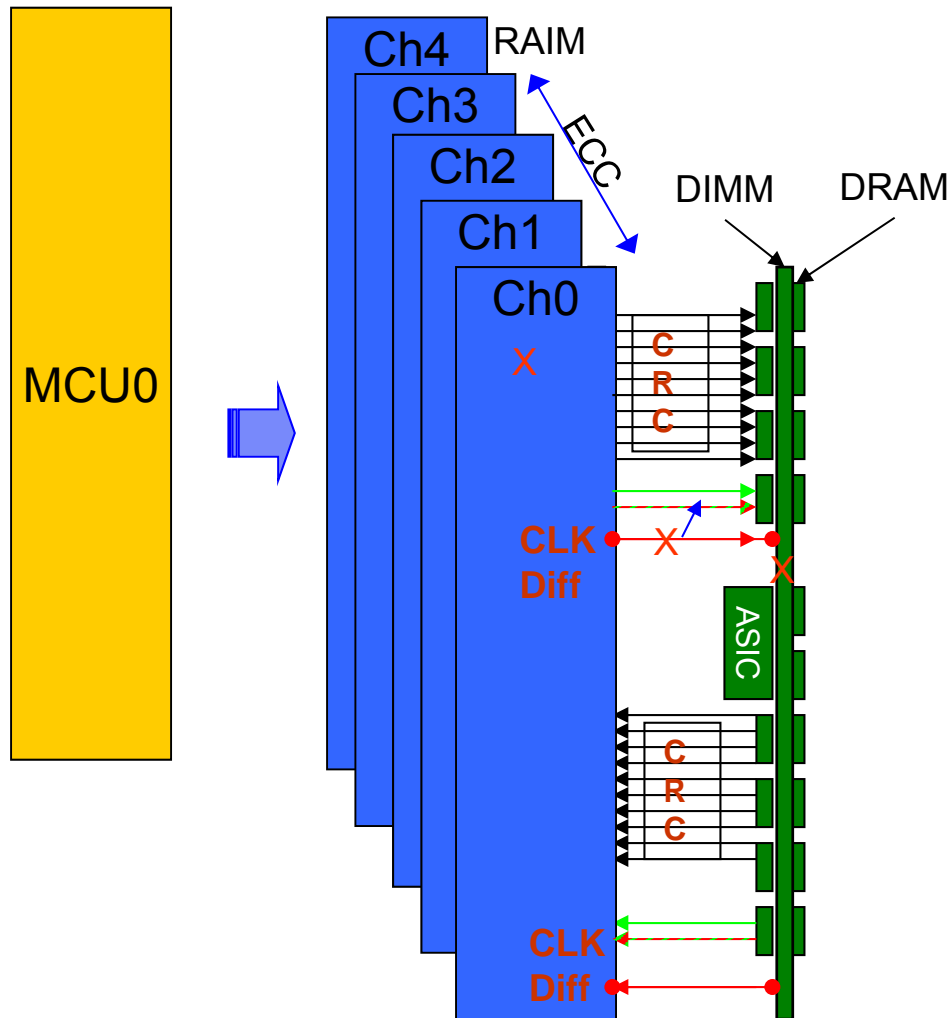


## z13 Memory Design

- One Memory Controller (MCU) per processor chip with *five memory channels*, one DIMM per channel. (zEC12 and z196 have a two DIMM cascade)
- The fifth channel in each MCU enables memory to be implemented as a **Redundant Array of Independent Memory (RAIM)**. *This technology has significant error detection and correction capabilities. Bit, lane, DRAM, DIMM, socket, and complete memory channel failures can be detected and corrected, including many types of multiple failures. So, RAIM takes 20% of DIMM capacity. (There is no non-RAIM option)*
- z13 uses new **DIMM** technology (“DIMM carry-forward” is impossible)
- DIMM sizes used are **16, 32, 64 or 128 GB** with **five DIMMs of the same size included in a memory feature. (80, 160, 320, 640 GB RAIM array size respectively)**
- Four or five features (**20 or 25 DIMMs**) are plugged in each drawer
- Features with different DIMMs sizes can be mixed in the same drawer
- **Eight different configurations of memory features are supported in a drawer (320 to 3200 GB RAIM which equates to 256 to 2560 GB addressable memory)**
- Addressable memory is required by *partitions and Hardware System Area (HSA)*

# z13 5-Channel RAIM Memory Controller Overview

*RAIM = Redundant Array of Independent Memory)*



**z13: Each memory channel supports only one DIMM**

## Layers of Memory Recovery

### ECC

- Powerful 90B/64B Reed Solomon code

### DRAM Failure

- Marking technology; no half sparing needed
- 2 DRAM can be marked
- Call for replacement on third DRAM

### Lane Failure

- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

### DIMM Failure (discrete components, VTT Reg.)

- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

### DIMM Controller ASIC Failure

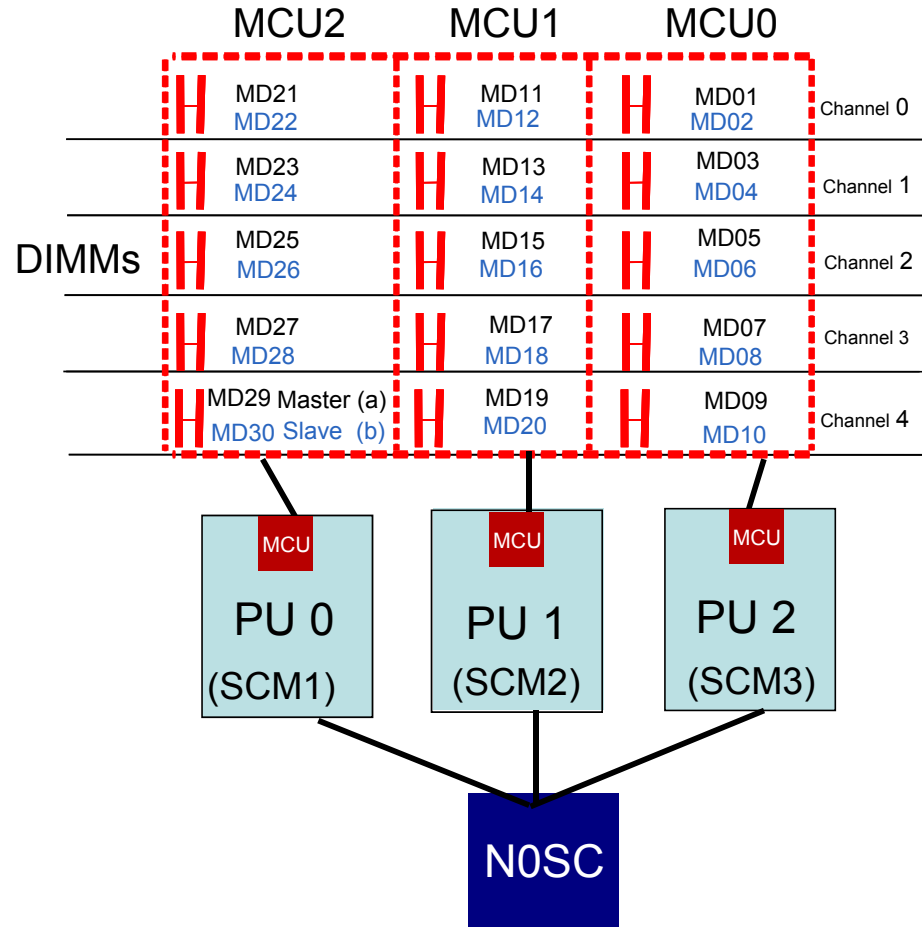
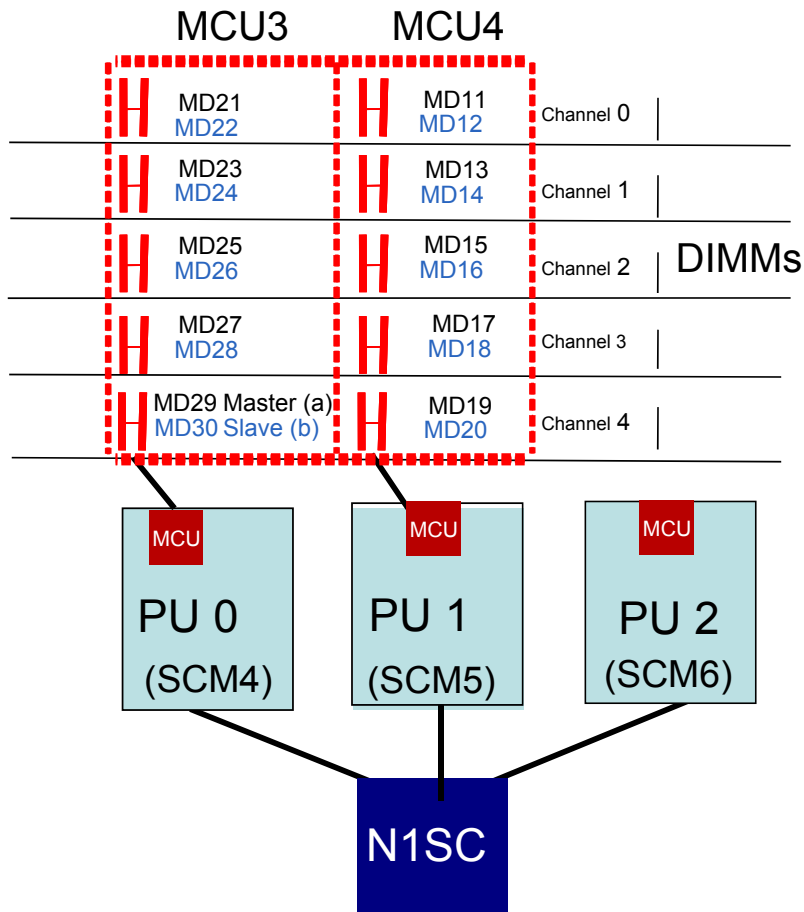
- RAIM Recovery

### Channel Failure

- RAIM Recovery



# Processor Drawer Memory Topology



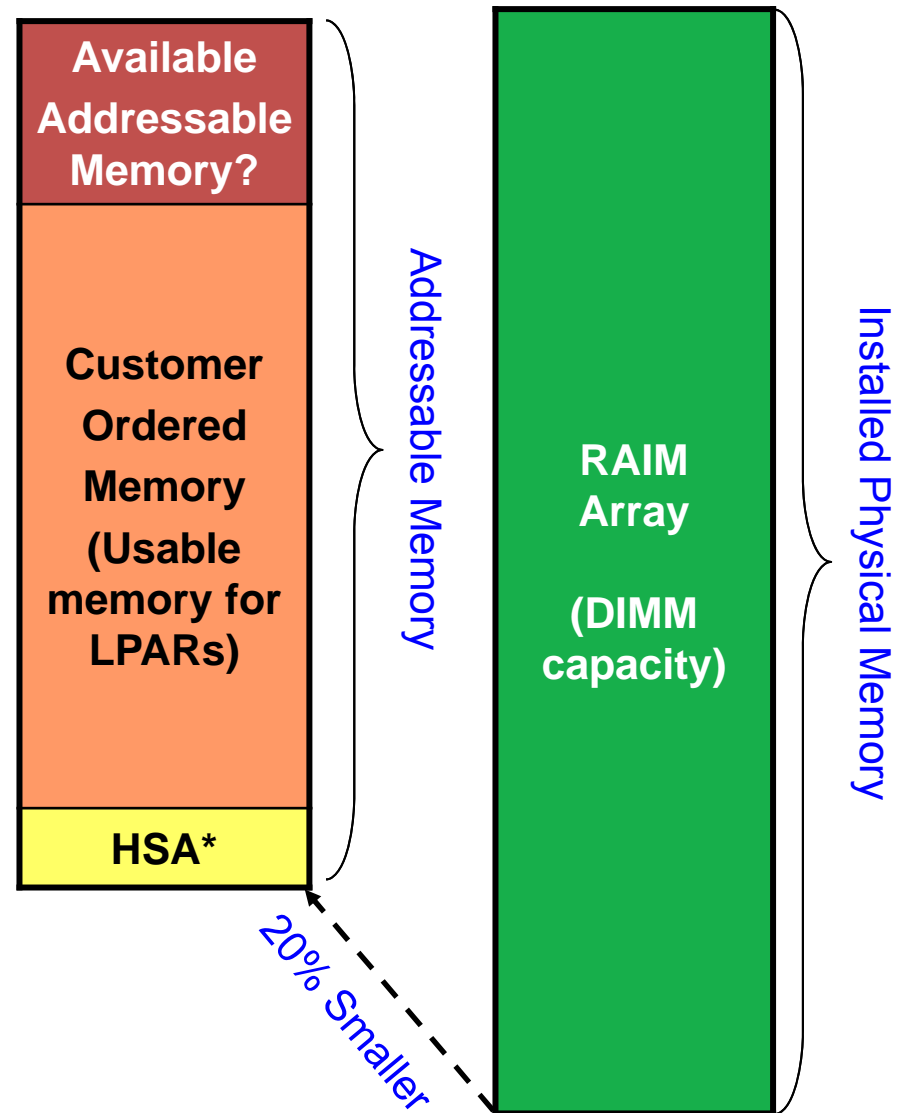
# z13 Memory Usage and Allocation

- Installed Physical Memory (DIMM capacity) in configuration reports is RAIM Array size. **Addressable Memory** for customer partitions and HSA is 20 percent smaller.
- Servers are configured with the most efficient configuration of memory DIMMs that can support **Addressable Memory required for Customer Ordered Memory plus HSA**. In some cases, there will be **Available Addressable Memory** that might support one or more concurrent LIC CC Customer Memory upgrades with no DIMM changes.

Note: DIMM changes require a disruptive POR on z13 Model N30. They are always done without a POR on z13 models with multiple drawers using **Enhanced Drawer Availability (EDA)**. On those models, some or all LPARs can continue to run with one drawer out of service to have DIMMs changed or added. Probably all LPARs, if Flexible Memory is selected.

- IBMer's: eConfig user TIP: To determine the size of the largest LIC CC Customer Memory upgrade possible, examine the configurator default "Memory Plan Ahead Capacity" field. If the customer requires a LIC CC upgrade larger than that, configure Plan Ahead Memory by selecting a larger "Memory Plan Ahead Capacity" target value.

\*HSA size is 96 GB on z13



## z13 Memory DIMMs and Plugging

- **z13 Memory Plugging**

- Six memory controllers per drawer, one per PU chip, three per node
- Each memory controller supports five DIMM slots
- Four or five memory controllers per drawer will be populated (20 or 25 DIMMs)
- Different memory controllers may have different size DIMMs

- **Maximum Client Memory Available**

- Remember RAIM – 20% of DIMM memory is used only for error recovery
- Minimum memory per drawer: 320 GB RAIM = 256 GB addressable
- Maximum memory per drawer: 3200 GB RAIM = 2560 GB addressable
- To determine maximum possible customer memory from the DIMM configuration: Calculate addressable memory, subtract 96 GB, and round down if necessary to an offered memory size

DIMM Size	z13 Feature (5 DIMMs) RAIM and Addressable Size
16 GB	#1610 = 80 GB RAIM, 64 GB Addressable Memory
32 GB	#1611 = 160 GB RAIM, 128 GB Addressable Memory
64 GB	#1612 = 320 GB RAIM, 256 GB Addressable Memory
128 GB	#1613 = 640 GB RAIM, 512 GB Addressable Memory

## z13 Drawer Memory Configurations

16 GB DIMMs/ Features	32 GB DIMMs/ Features	64 GB DIMMs/ Features	128 GB DIMMs/ Features	Total DIMMs	RAIM Array GB (Physical)	Customer Memory GB (If no HSA*)
<b>20/4</b>				<b>20</b>	<b>320</b>	<b>256</b>
<b>20/4</b>	<b>5/1</b>			<b>25</b>	<b>480</b>	<b>384</b>
<b>10/2</b>	<b>15/3</b>			<b>25</b>	<b>640</b>	<b>512</b>
	<b>20/4</b>	<b>5/1</b>		<b>25</b>	<b>960</b>	<b>768</b>
	<b>10/2</b>	<b>15/3</b>		<b>25</b>	<b>1,280</b>	<b>1,024</b>
		<b>20/4</b>	<b>5/1</b>	<b>25</b>	<b>1,920</b>	<b>1,536</b>
			<b>20/4</b>	<b>20</b>	<b>2,560</b>	<b>2,048</b>
			<b>25/5</b>	<b>25</b>	<b>3,200</b>	<b>2,560</b>

\*In the first book, available addressable memory is reduced 96 GB taken by the Hardware System Area (HSA). HSA size is fixed, standard and is included in the base price. It is not taken from customer purchased memory.

## z13 Purchased Memory Offering Ranges

Model	Standard Memory GB	Flexible Memory GB
N30	64 - 2464	NA
N63	64 - 5024	64 - 2464
N96	64 - 7584	64 - 5024
NC9	64 - 10144	64 - 7584
NE1	64 - 10144	64 - 7584

- **Purchased Memory** - Memory available for assignment to LPARs
- **Hardware System Area** – Standard 96 GB of addressable memory for system use outside customer memory
- **Standard Memory** - Provides minimum physical memory required to hold customer purchase memory plus 96 GB HSA
- **Flexible Memory** - Provides additional physical memory needed to support activation base customer memory and HSA on a multiple CPC drawer z13 with one drawer out of service.
- **Plan Ahead Memory** – Provides additional physical memory needed for a concurrent upgrade (LIC CC change only) to a preplanned target customer memory

## z13 Plan Ahead Memory

- Provides the capability for concurrent memory upgrades changing Licensed Internal Code (LIC CC) without exploitation of [Enhanced Drawer Availability](#)
  - Memory DIMMs are pre-installed to support a specified **target** preplanned memory size
  - Available on all z13 models
  - Can be ordered with **Standard** memory on all z13 models  
(**Standard plus Preplanned will NOT be Flexible in most cases.**)
  - Can be ordered with **Flexible** memory on a multiple drawer z13 model
- Preplanned memory features are chargeable
  - The charge is calculated as 50% of the cost to upgrade to the **MAXIMUM** standard memory **ALLOWED** by any **LARGER DIMM HARDWARE** that must be added to enable the selected Plan Ahead **target**, **NOT on 50% of the cost of upgrade to the target the client wanted unless that happens to be the maximum supported by the larger DIMM hardware that has to be added.** (“Additional hardware” means larger hardware than needed for base memory.)
  - Features: **FC #1996 16GB** Preplanned Memory, **FC #1990 32GB** Preplanned Memory
- Preplanned memory activation is chargeable
  - Subsequent memory upgrade orders will “consume” Preplanned Memory first
  - Charged when Preplanned Memory is enabled by concurrent LIC upgrade at 50% of the cost if the Preplanned Features were not present.
- Note: Preplanned Memory is NOT temporary, On Demand Memory  
Why? [Because Memory LIC downgrade is disruptive.](#)

## z13 Flexible Memory

- Provides additional physical memory DIMMs needed to support activation of all customer purchased memory and HSA on a multiple drawer z13 with one drawer down for:
  - [Scheduled concurrent drawer upgrade](#) (e.g. memory add)
  - [Scheduled concurrent drawer maintenance](#) (N+1 repair)
  - Concurrent repair of an out of service book “fenced” during Activation (POR)
  - [Note: All of the above can be done without Flexible Memory; but, all customer purchased memory will not be available for use in most cases. Some work may have to be shut down or not restarted.](#)
- Offered on N63, N96, NC9 and NE1 in:
  - 32 GB increments from 64 GB to 192 GB
  - 64 GB increments from 192 GB to 448 GB
  - 96 GB increments from 448 GB to 928 GB
  - 128 GB increments from 928 GB to 1440 GB
  - 256 GB increment to 1440 GB to 6048 GB ([2464 GB max for N63, 5024 GB max for N96](#))
  - 512 GB increments from 6048 GB to 7584 GB ([NC9 and NE1](#))
- IBMers: [Check the “Flexible” box](#) in eConfig when configuring memory
  - eConfig adds additional physical memory to the configuration.
  - eConfig adds “Preplanned Memory” features for pricing.

## z13 Standard and Flexible Purchase Memory Offerings

Memory Increment (GB)	Offered Memory Sizes (GB)	Memory Maximum Notes (GB)
<b>32</b>	64, 96, 128, 160, 192	
<b>64</b>	256, 320, 384, 448	
<b>96</b>	544, 640, 736, 832, 928	
<b>128</b>	1056, 1184, 1312, 1440	
<b>256</b>	1696, 1952, 2208, <b>2464</b> , 2720, 2976, 3232, 3488, 3744, 4000, 4256, 4512, 4768, <b>5024</b> , 5280, 5536, 5792, 6048	<b>2464</b> – N30 Standard, N63 Flexible  <b>5024</b> – N63 Standard, N96 Flexible
<b>512</b>	6560, 7072, <b>7584</b> , 8096, 8608, 9120, 9632, <b>10144</b>	<b>7584</b> – N96 Standard, NC9 and NE1 Flexible <b>10144</b> – NC9 and NE1 Standard



## IBM z13 Memory Granularity (Increment Size)

- Memory Granularity for Assignment to an LPAR and for Configuration ON and OFF is the LPAR's **Increment Size**
  - z13 physical increment size is fixed at 512 MB (Increased from 256 MB on zEC12)
  - Central memory granularity is virtualized as a multiple of 512 MB for each LPAR based on the size of the **larger** of its two Elements: Initial Central and Reserved Central. Increment Size increases with Element size because operating systems that support memory reconfiguration (z/OS and z/VM) support only up to 512 Increments per Element.
  - **Expanded Memory** granularity on z13 is always 512 MB (Increased from 256 MB on zEC12)  
 Note: Expanded Memory is simulated using Central Memory  
**SoD: z13 is the last z System to support Expanded Memory**
  
- **Action:** Review MVS™ RSU parameter. Change RSU (if non-zero) to MB or GB values, not as a number of Increments to avoid problems if LPAR memory grows. A MB or GB value will be rounded up if not equal to the partition's Increment Size.

Larger Central Memory Element Size	Increment Size (Granularity)
512 MB to 256 GB	512 MB
>256 GB to 512 GB	1 GB
>512 GB to 1 TB	2 GB
>1 TB to 2 TB	4 GB
>2 TB to 4 TB	8 GB
>4 TB to 8 TB	16 GB
>8 TB to 10 TB	32 GB

# IBM z13 Memory Addressability

- **PR/SM assigns more than just memory to a partition when it is activated:**
  - Initial Central Memory
  - Initial Expanded Memory (Simulated using Central for z/VM and Linux on z System only)
  - **Contiguous Central Memory Addressability** for **both** Initial and Reserved Central
  - **Contiguous Expanded Memory Addressability** for **both** Initial and Reserved Expanded
  - **Contiguous Initial and Reserved memory addressability** is required to configure the Reserved Element ON
  - Neither central nor expanded memory is required to be physically contiguous. Operating systems “see” it that way because of the contiguous addressability assigned to it. In fact, **PR/SM can use Dynamic Memory Relocation (DMR) to move a running partition’s memory to different physical memory**
  - To change a partition’s Central or Expanded Memory definitions requires the partition to be deactivated.
  
- **Memory Addressability Notes:**
  - Every z13 has **64 TB** of addressability, HSA requires **118 GB**, leaving **65,418 GB for partitions**.
  - Addressability is assigned working from high to low in **2GB chunks aligned on 2 GB boundaries** to support 2GB fixed pages. (New on zEC12 and zBC12). A small partitions may “waste addressability”. For example, a partition with 1 GB of Initial Central and no Reserved Central “wastes” 1 GB. This is reported as a **“Memory Gap”** on the HMC. (**NOT a problem with 65,418 GB available**)

Larger Central Memory Element Size	Addressability Increment Size
512 MB to 256 GB	2 GB
>256 GB to 512 GB	2 GB
>512 GB to 1 TB	2 GB
>1 TB to 2 TB	4 GB
>2 TB to 4 TB	8 GB
>4 TB to 8 TB	16 GB
>8 TB to 10 TB	32 GB

## z System Hardware System Area (HSA) Comparisons

- The size of HSA on z System servers before System z10 was dependent on configuration, taken from customer purchased memory, and required a POR to increase if a configuration change required it to grow.
- On System z10 and later HSA size is fixed and not taken from customer purchased memory. The increase in size correlates with increased memory size and enhanced function. (More partitions, more subchannel sets, etc.)

– z13	96 GB
– zEC12	32 GB
– zBC12	16 GB
– z196	16 GB
– z114	8 GB
– z10 EC	16 GB
– z10 BC	8 GB



## Definitions

- **Processor core (PU core)**
  - Hardware capable of executing one or more instruction streams
  - Includes controls for fetching, decoding, and executing instructions
- **Multi-core**
  - A design in which multiple processor cores are fabricated in a single processor chip
- **Software thread**
  - A linear sequence of instructions executed for one program
- **Hardware thread**
  - The set of processor resources used to execute a single software thread
- **Multi-thread**
  - A design in which a single processor core can be actively executing multiple software threads
- **Thread speed**
  - The rate at which the instructions of a given software thread are executed
- **Instruction Throughput**
  - The aggregate rate at which instructions for all software threads are executed across a given set of processor cores / hardware threads
- **Cache Miss**
  - When a program accesses a memory location that is not in the cache, it is called a *cache miss*. Since the processor then has to wait for the data to be fetched from the next cache level or from main memory before it can continue to execute, cache misses directly influence the performance of the application

## Simultaneous Multithreading - Background

- SMT enables to run multiple threads on a single core
  - Other processor families (i.e. x86, etc.) already have similar support
  - Each thread runs slower than a non-SMT core, but the combined ‘threads’ throughput is higher. The overall throughput benefit depends on the workload
- Hardware support
  - Single thread (ST) operation
  - SMT operation with seamless transition between ST and SMT
  - Precise metering of SMT utilization => Monitors Dashboard
- Software must actually enable the use of SMT operation
  - You must have software at levels that can exploit SMT
  - Use of SMT is on a per-LPAR basis
  - The support is present in the z13
    - The OS(es) must actually issue instructions to switch into SMT mode
  - Enabling the use of SMT is unidirectional
    - Once the OS switches, the only way back to ST mode is via a disruptive action (re-activate the partition or to re-IPL it).
    - With the SMT enabled mode it is possible to dynamically switch between MT-1 (multi thread) and MT-2 mode for the processor types that support MT-2

## What is Simultaneous MultiThreading (SMT)

- Today each z System CPU supports a single instruction stream
  - z System workloads tend to receive a nontrivial # of cache misses
  - CPU generally unproductive while resolving cache miss
- z13 SMT makes CPU (Core) productive during cache misses
  - z13 supports two way SMT (Two instruction streams [Threads]) per core
    - Each Thread has its own unique architecture / state information (Registers, PSW, etc.)
    - Cannot necessarily execute instructions instantly and must compete and win the use of desired core resources shared between threads
    - z13 insures that one thread can't lock out the other
  - z13 allows following engine types to run in SMT2 mode
    - zIIPs under z/OS
    - IFLs under z/VM
- READY TO RUN Threads share core
  - Threads NOT READY TO RUN still unproductive while resolving cache miss
  - Core resources are productive when either READY TO RUN Thread is executing

## What is SMT ...

- SMT is architected in System z13
  - The facility provides the means by which more efficient utilization of a configuration's resources may be realized
  
- MultiThreading introduces the terminology of a core
  - Comprise of CPUs (often called Hardware Threads)
  - The Core comprises all the architected resources available to the threads such as the program-status word, registers, timing facilities, and so forth
  
- When the SMT is not enabled, a core executes a single Hardware Thread
  
- When the SMT facility is enabled, the Hardware Threads within a core share certain hardware resources such as execution units and caches
  
- When one Hardware Thread in a core is waiting for other hardware resources (typically, while waiting for a storage access), second Hardware Thread in the core can utilize the shared resources in the core rather than remain idle
  
- Threads can share resources that are not experiencing competition



## Simultaneous Multi-threading (SMT)

- **z13 is the first z System Processor to support SMT**
  - Enable continued scaling of per-processor capacity
  - z13 supports 2 threads per core on IFLs and zIIPs *only*
- **Increases per-core and system throughput versus single thread design**
  - More work done per unit hardware
  - Aligns with industry direction of multi-thread
  - Improves **per-core** performance comparisons vs. X86, POWER
  - Improves efficiency of IFL for Linux consolidation
- **Designed to preserve unique z System values and attributes**
  - Full support for 2-level processor virtualization
  - Full z/Architecture capability for each thread
- **Design will allow independent enablement of SMT by LPAR**
  - Operating systems must be explicitly enabled for SMT
  - Operating system may opt to run in single-thread mode
- **Processors can run in single-thread operation for workloads needing maximum thread speed**
- **Functionally transparent to middleware and applications**
  - No changes required to run in SMT partition
- **Operating System / Hypervisor Support**
  - z/OS (for zIIPs) at GA
  - zVM (for IFLs) at GA
  - Linux: IBM is working with its Linux Distribution partners to support new functions / features

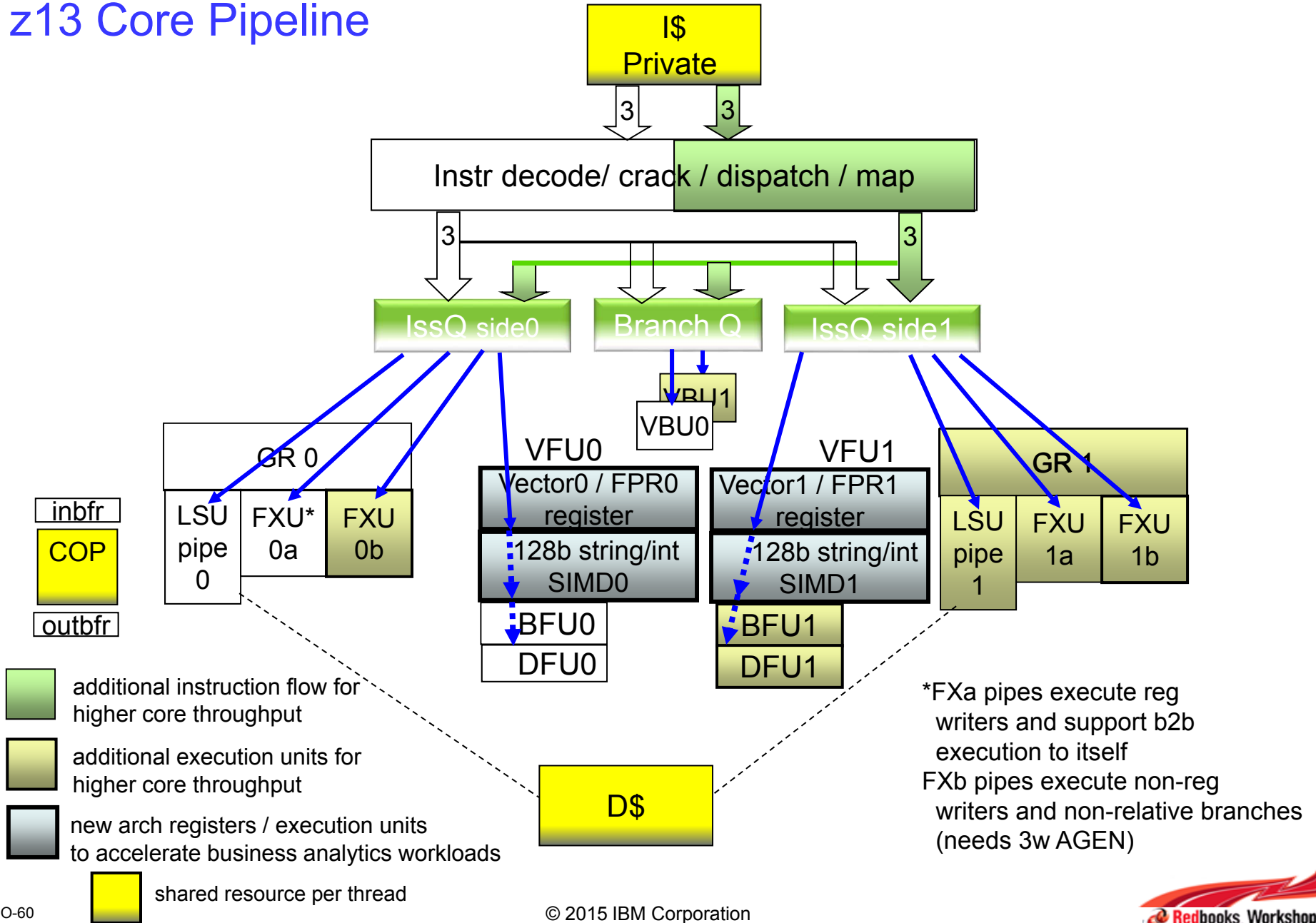
## SMT Hardware Implementation

- Support two instruction streams (threads) per core
  - Active threads share core resources
    - **In time:** Address translator, instruction execution units, pipeline slots, ...
      - Cache misses provide opportunities for other thread
      - Machine ensures fairness between threads
    - **In space:** Data and instruction caches, branch history tables, TLBs, ...
  - Each core thread has its set of architectural resources
    - E.g., take interruptions, start I/O, load wait PSW, signal other threads
  - Increases overall throughput per core when SMT is active
    - Amount increased varies widely with the workload

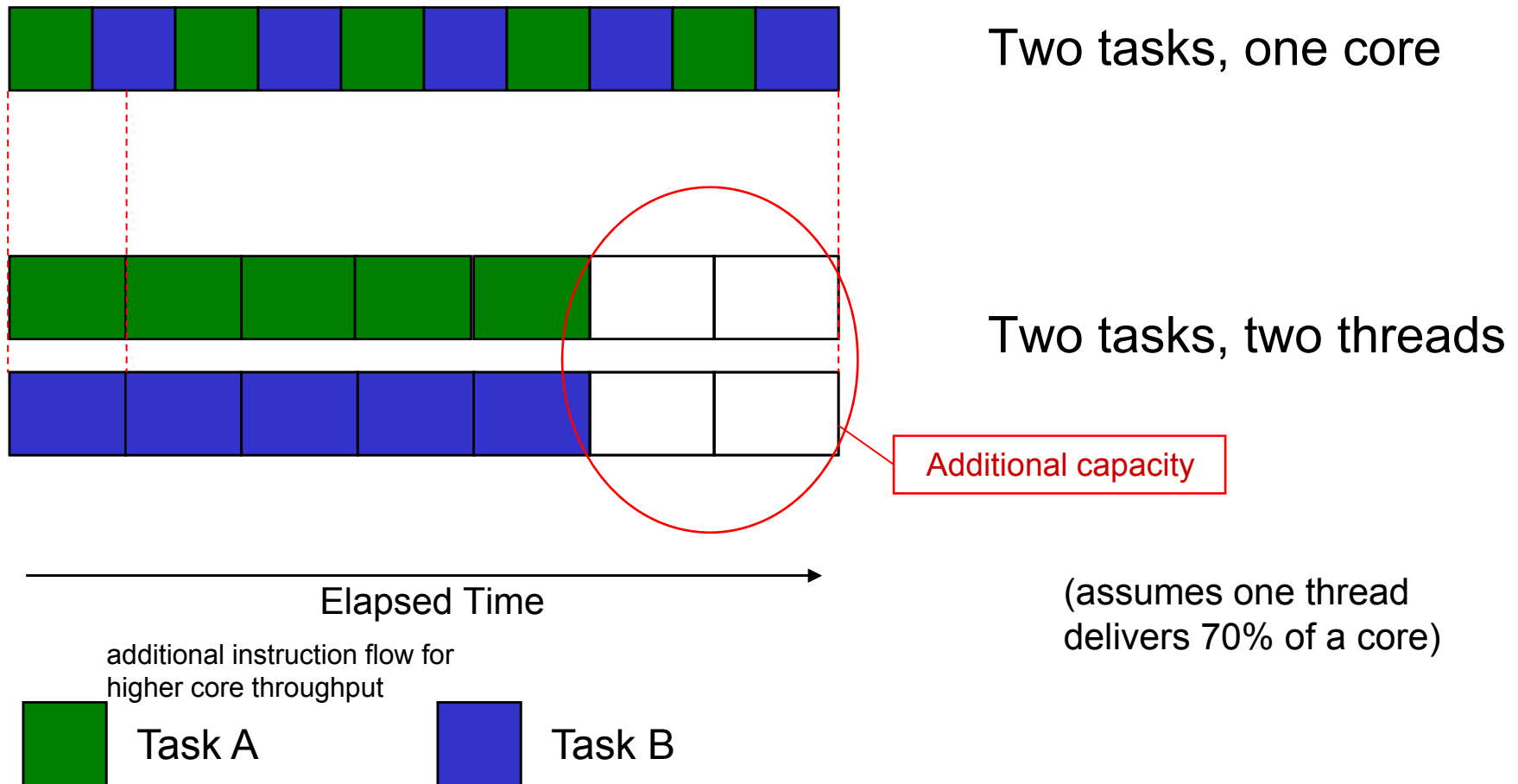
## z13 Core Highlights to support SMT2

- New instruction front-end
- Significant branch prediction changes
- Shorter latency instruction fetch pipeline
- Double instruction fetch/ decode/ dispatch bandwidth
  - Up to six instructions per cycle
- I-side L2
- Larger ISU structures (OoO) window
  - Mapper, issue Q..
  - Increased execution bandwidth
  - 4 FXUs
  - 2 BFUs, 2 DFUs
  - 2 new SIMD units
- New store forwarding cache
- Double completion bandwidth
  - Up to two triplets (groups) per cycle
- New COP design

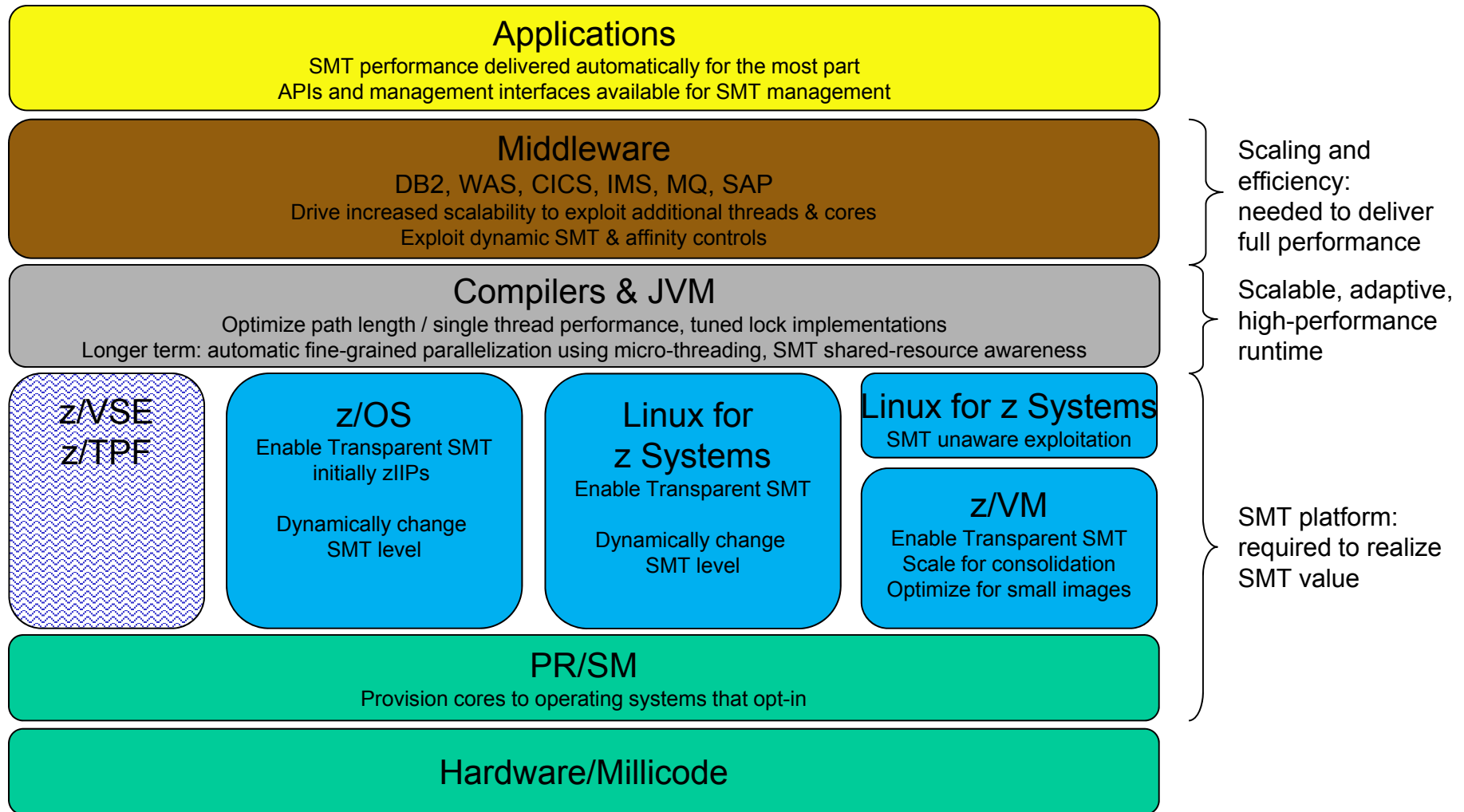
# z13 Core Pipeline



# Simultaneous Multithreading Value Example



# Layered SMT Exploitation on z System



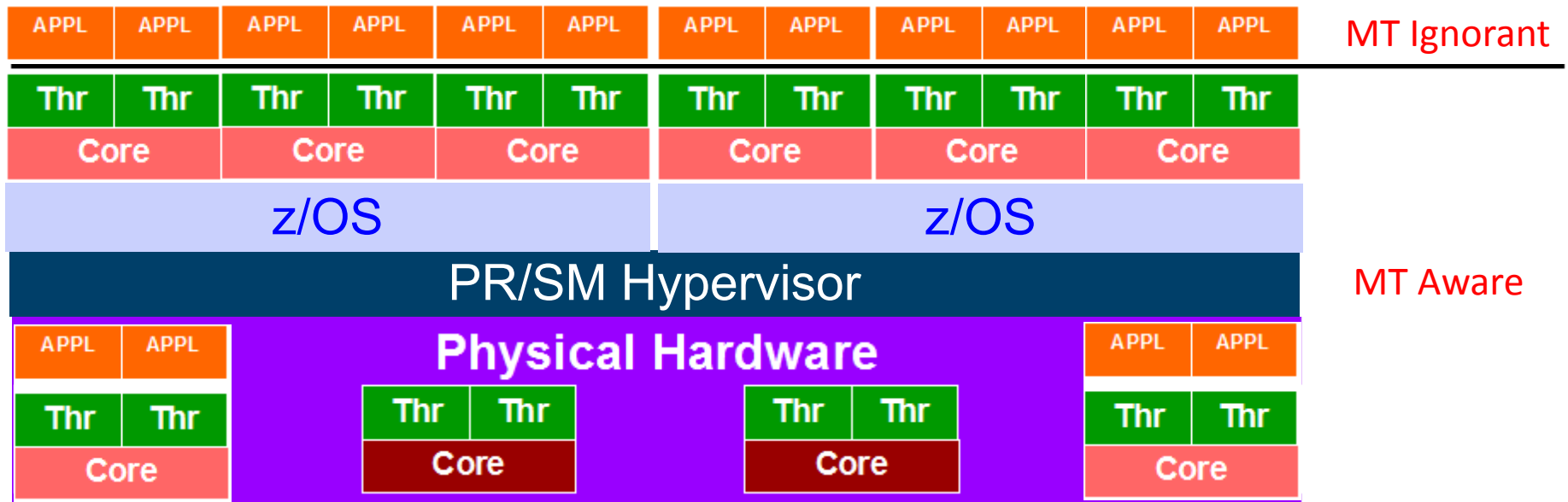
## z13 Core Hardware

- **SMT Core on z13**
  - **Thread Density = 1** Core (One Thread Executing, one waiting)
    - No capacity gain, no speed degradation. Same performance as without MT
    - Core using some capacity and has some free capacity (workload dependent)
  
  - **Thread Density = 2** Core (Two Threads Executing)
    - Workload dependent core capacity gain (0-xx%)<sup>1</sup> versus single thread
    - Workload dependent thread speed degradation (0-yy% slower)<sup>1</sup> versus single thread
    - Hardware provides the time the work is occupying (executing, competing to execute, resolving cache misses) a thread
  
  - **Core in wait** (all threads waiting), all core capacity is free

<sup>1</sup> Final numbers will become available once the measurements are completed

## z System SMT Exploitation

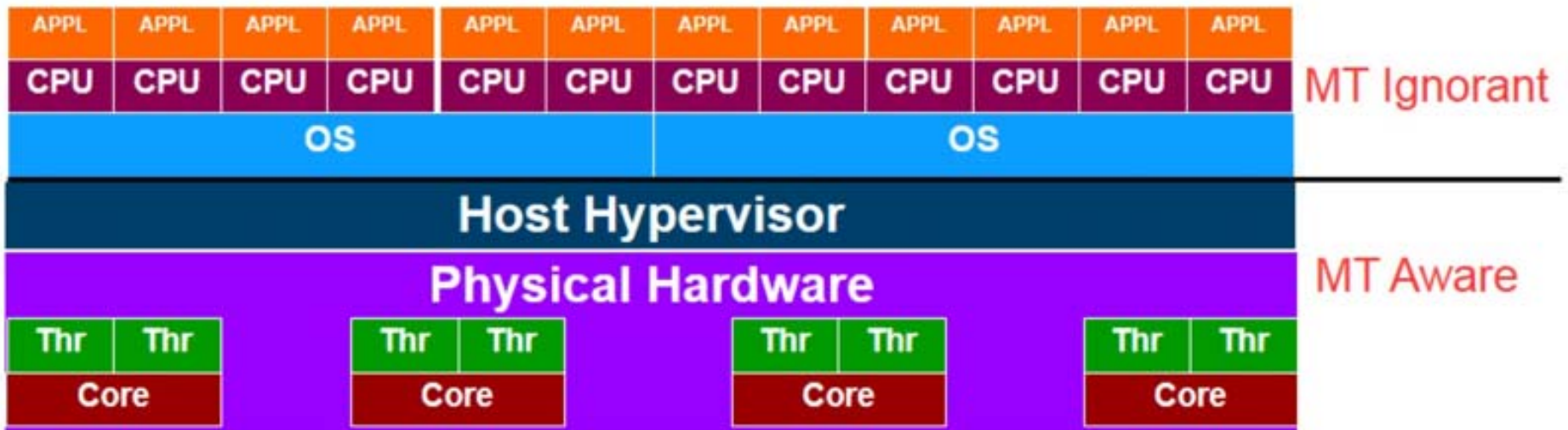
- SMT Aware OS informs PR/SM that it intends to exploit SMT
  - PR/SM can dispatch any OS core to any physical core
  - OS control the whole core – must follow rules
    - Maximize core throughput (Drive cores with high Thread Density [2] )
    - Maximize core availability (Meet workload goals using fewest cores )
  
- SMT is transparent to applications





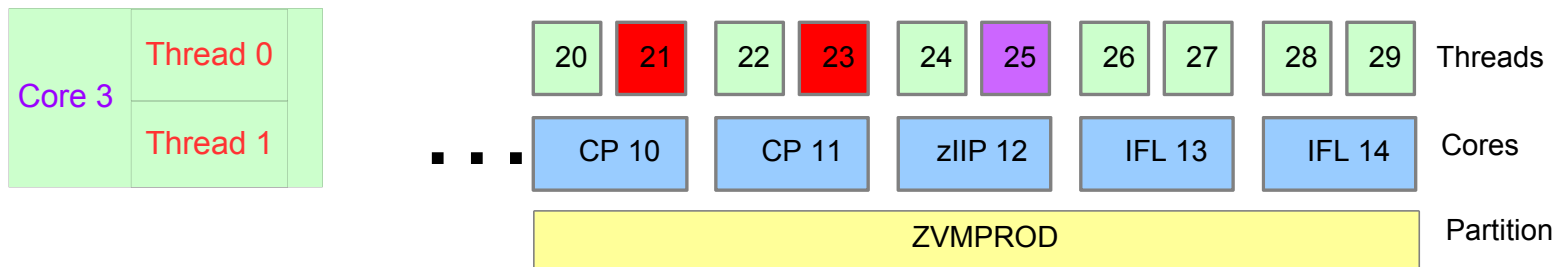
## Industry SMT Exploitation

- Host Hypervisor provides MT transparently to the OS
  - Hypervisor can dispatch any OS Logical CPU to any thread
  - OS receives variability in the form of:
    - Unpredictable capacity (core floats between TD=1, TD=2 randomly)
    - Meaningless charge back (time executing and competing to execute)
    - Uncontrolable latency, response time
  - Industry MT unacceptable for z System virtualization scalability requirements



# z13 Core Virtualization

- CPU Address changes with SMT2
  - Sixteen bit CPU Id consists of a fifteen bit Core ID and one bit Thread ID



– Without SMT

- CPU x0014 = 0000 0000 0001 0100

– With SMT

- Core x0014 thread 0 = 0000 0000 0010 1000 (CPU x0028)
- Core x0014 thread 1 = 0000 0000 0010 1001 (CPU x0029)
- Non-IFL processor odd address unavailable or unused
- Hardware makes both threads usable on each core

- On z13, z/OS will support SMT2 for zIIPs and z/VM will support SMT2 for IFLs
- For CPs only **Thread 0** usable on each core
- SMT aware Hypervisors (z/VM) or Operating Systems (z/OS) must Opt-in at IPL to exploit SMT over the life of IPL

**Views**

Groups

Exceptions

Active Tasks

Console Actions

Task List

Help

**A01 CPs Work Area**

000 Online Operating

002 Online Operating

004 Online Operating

006 Online Operating

008 Online Operating

009 Online Operating

00A Standby Stopped

00C Standby Stopped

00E Standby Stopped

010 Standby Stopped

012 Standby Stopped

013 Standby Stopped

```

2015045 15.11.36 SC76      IEE174I 15.11.36 DISPLAY M 954
CORE STATUS: HD=Y      MT=2  MT_MODE: CP=1  zIIP=2
ID   ST   ID RANGE  VP  ISCM  CPU  THREAD STATUS
0000 +   0000-0001  M   FC00  +N
0001 +   0002-0003  L   0000  +N
0002 +   0004-0005  LP  0000  +N
0003 +   0006-0007  LP  0000  +N
0004 +I  0008-0009  M   0200  ++
0005 -   000A-000B
0006 -   000C-000D
0007 -   000E-000F
0008 -   0010-0011
0009 -I  0012-0013

CPC ND = 002964.N63.IBM.02.000000008DA87
CPC SI = 2964.735.IBM.02.0000000000008DA87
      Model: N63
CPC ID = 00
CPC NAME = SCZP501
LP NAME = A01          LP ID = 1
CSS ID = 0
MIF ID = 1
    
```

- Daily**
- Hardware Messages
  - Operating System Messages
  - Activate
  - Reset Normal
  - Deactivate

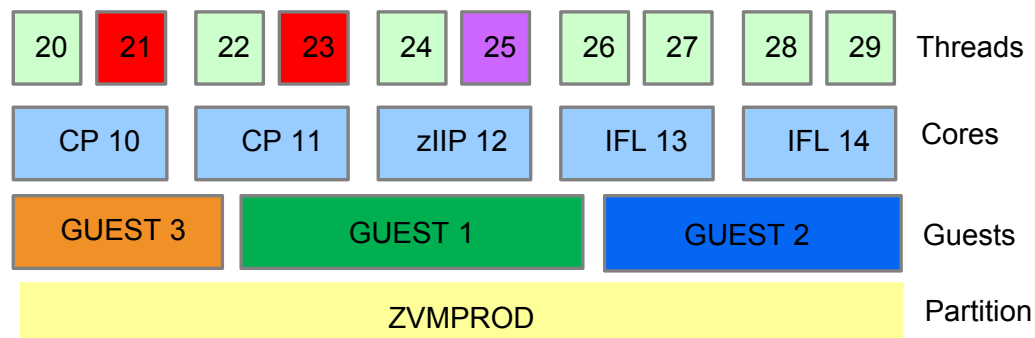
## Exploiting SMT on z13 with zIIP

- z/OS V2R1 SMT APARs must be applied
  - OA43366 (BCP), OA43622 (WLM), OA44439 (XCF)
    - z/OS manages Threads according to the SMT Mode
- New LOADxx and IEAOPTxx controls ONLY available on z/OS V2R1 and higher
  - Requires a separate LOADxx for z/OS V1.13 or z/OS V1.12 if SMT is exploited
- Define a LOADxx PROCessor VIEW (PROCVIEW) CORE|CPU for the life of the IPL
  - PROCVIEW CORE in z/OS on z13 enables SMT2 support
  - Fall back from PROVIEW CORE to PROCVIEW CPU requires IPL
  - PROCVIEW CORE,CPU\_OK causes z/OS to treat CPU as an acceptable alias for CORE
- New IEAOPTxx parameter to control zIIP SMT mode
  - MT\_ZIIP\_MODE=2
    - MT\_ZIIP\_MODE=2 for 2 active threads (the default is 1)
    - When PROCVIEW CPU is specified the processor class MT Mode is always 1
  - Without an IPL you can change the zIIP processor class MT Mode (the number of active threads per online zIIP) using IEAOPTxx. SET OPT=xx
  - Requires HiperDispatch = YES

Note: z13 doesn't support zAAPs - zAAP on zIIP makes zAAPs obsolete

## z/VM SMT Support Implementation

- Enable SMT for IFLs
- Support up to 32 multithreaded cores (64 threads)
- Treat each thread as an independent ‘processor’
- Dispatch virtual IFLs on the Hardware Threads
  - Same or different guests can share core
  - Adds to variability
- Exploit topology awareness
  - Single Dispatch Vector per core
  - Topologically-aware steal
  - Slight bias towards placing virtual MT sibling CPUs on same Dispatch Vector
- Reduced overhead
  - Improved handling of guest IPTE and similar instructions



## SMT Limitations in z/VM

- No dynamic switching of SMT mode
- Do not require (do not support) guest awareness or exploitation of SMT
- No mechanism to give guest whole core (by leaving one thread idle)
- HiperDispatch must be enabled
  - No support for dedicating processors
- Thread-aware CPU Pooling planned to be a later deliverable
- No drawer awareness in HiperDispatch support



## Single Instruction Multiple Data (SIMD) Introduction

### ▪ Background

- The amount of data is increasing exponentially  
IT shops need to respond to the diversity of data
- Enterprises use traditional integer, floating point, string, and XML character-based data
- It's becoming more important for customers to do computations, analytics closer to the data

### ▪ Customer Perception of Analytics and z System

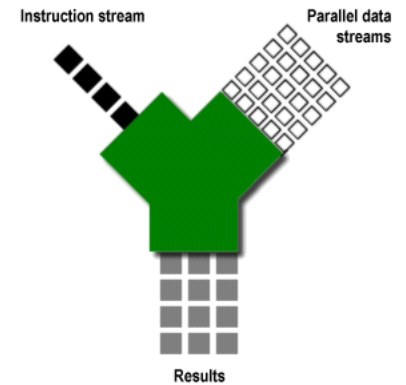
- z System handles OLTP and Batch; math and data intensive operation can lead to unaffordable MIPS usage

### ▪ Reality of Analytics and z System

- For the last 2-3 generations, z has changed its capabilities in compute processing (analytics)
  - SuperScalar, Out of Order (OoO), compiler improvements, floating point
  - Existing Capabilities: quad precision floating point, fuse/multiply/add
- SIMD provides next phase of enhancements for analytics and compute-intensive competitiveness on z System

### ▪ Single Instruction Multiple Data (SIMD)

- When used with provided libraries and compilers, creates a platform for numeric and data intensive computing, minimizing the effort on the part of middleware/application developers for exploitation



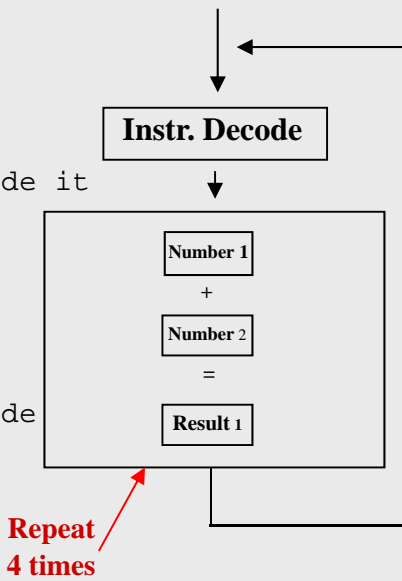


# SIMD (Single Instruction Multiple Data) Processing Example

- **Significantly smaller amount of code – improved execution efficiency**
  - Number of elements processed in parallel = (size of SIMD / size of element)

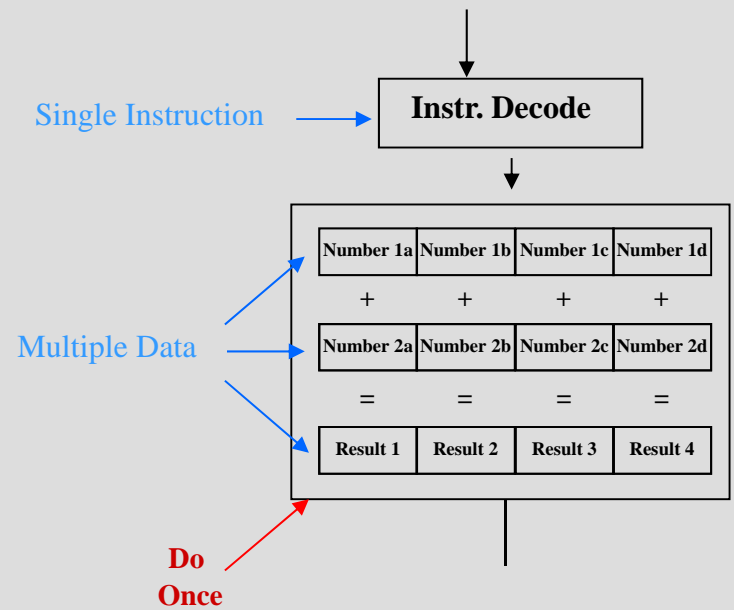
## Scalar code

```
read the next instruction and decode it
get this number
get that number
add them
put the result here
read the next instruction and decode it
get this number
get that number
add them
put the result here
read the next instruction and decode
get this number
get that number
add them
put the result here.
read the next instruction and decode it
get this number
get that number
add them
put the result there
```



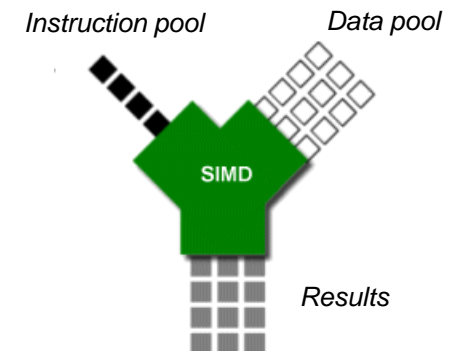
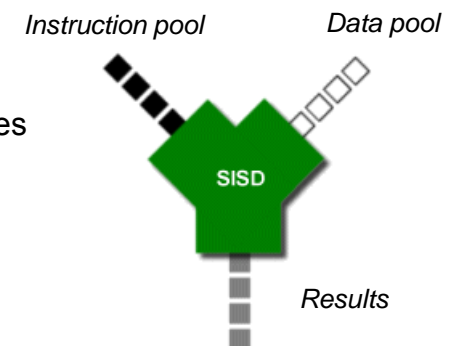
## SIMD code

```
read instruction and decode it
get these 4 numbers
get those 4 numbers
add them
put the results here
```



# Single Instruction Multiple Data (SIMD) Vector Processing

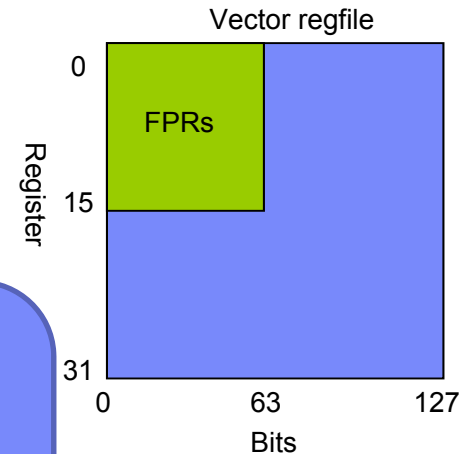
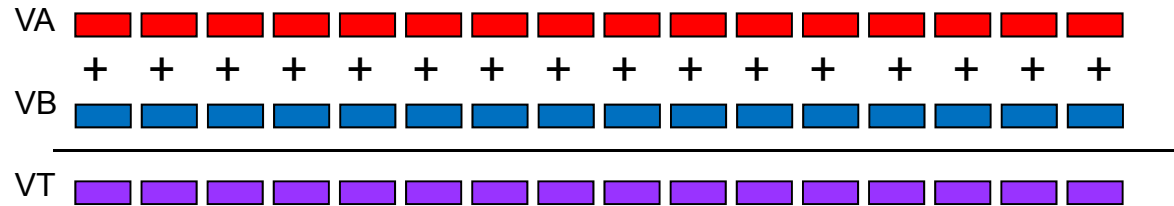
- Single Instruction Multiple Data (SIMD)
  - A type of data parallel computing that can accelerate code with integer, string, character, and floating point data types
- Provide optimized SIMD math & linear algebra libraries that will minimize the effort on the part of middleware/application developers
- Provide compiler built-in functions for SIMD that software applications can leverage as needed (e.g. for use of string instructions)
- OS/Hypervisor Support:
  - z/OS: 2.1 SPE available at GA
  - Linux: IBM is working with its Linux Distribution partners to support new functions/features
  - No z/VM Support for SIMD
  - Compiler exploitation
    - IBM Java => 1Q2015
    - XL C/C++ on zOS => 1Q2015
    - XL C/C++ on Linux on z => 2Q2015
    - Enterprise COBOL => 1Q2015
    - Enterprise PL/I => 1Q2015



Workloads		
Java.Next	C/C++Compiler built-ins for SIMD operations (z/OS and Linux on z System)	MASS & ATLAS Math Libraries (z/OS and Linux on System z)
SIMD Registers and Instruction Set		

MASS - Mathematical Acceleration Sub-System  
 ATLAS - Automatically Tuned Linear Algebra Software

# z System SIMD Hardware Accelerator



- Each register contains multiple data elements of a fixed size: Byte, Halfword, Word, Doubleword, Quadword
  - Instruction specifies which data format to use
- The collection of elements in a register is also called a vector
- A single instruction will operate on all of the elements in the register
- Instructions have a non-destructive operand encoding (T=A+B vs A=A+B)
- For most operations the CC is not set
  - For a few instructions a summary *condition code* is used

- Initial implementation: 32 x 128b Vector Registers  
Both dimensions may grow in future
- Vector register file overlays the FPRs  
FPRs 0-15 == Bits 0:63 of SIMD regs 0-15  
Update to FPR <x> alters **entire** SIMD register <x>
- Why overlay?  
Saves hardware area / power  
Easier mixing of scalar / SIMD code  
Less copying of values between registers  
Effectively get 64 FPRs  
Can improve FP code efficiency

## Operates on three distinct data types:

Integer	String	Floating-point
<p><b>16 x Byte, 8 x HW, 4xW, 2xDW, 1xQW</b></p> <ul style="list-style-type: none"> <li>• Byte to QuadWord add, sub, compare</li> <li>• Byte to DoubleWord min, max, ave.</li> <li>• Byte to Word multiply, multiply/add 4 - 32 x 32 multiply/adds</li> <li>• Logical ops, shifts,</li> <li>• CRC (GF multiply up to 64b), Checksum (32b),</li> <li>• Loads efficient with 8B alignment though minor penalties for byte alignment</li> <li>• Gather by Step</li> </ul>	<ul style="list-style-type: none"> <li>• Find 8b, 16b, 32b, equal or not equal with zero character end</li> <li>• Range compare</li> <li>• Find any equal</li> <li>• Load to block boundary, load/store with length</li> </ul>	<p><b>BFP DP only 32 x 2 x 64b</b></p> <ul style="list-style-type: none"> <li>• 2 BFUs with an increase in architected registers</li> <li>• Exceptions suppressed</li> </ul>

## SIMD Exploitation Considerations

- MASS (Mathematical Acceleration Sub-System) Libraries
  - Libraries require SIMD instruction support for the SIMD and vector version of MASS
  - Library provides highly optimized primitive math functions (log, inv, sqrt, exp etc.) that are the foundation of Business Analytics
  - The scalar version of the library does not use SIMD instructions
  - The zEC12 libraries (vector and scalar) do not use SIMD instructions
- ATLAS (Automatically Tuned Linear Algebra Software) Libraries
  - ATLAS has a version for z13 and another version for zEC12
- Either Library
  - Any library function calls will abend if run on a lower hardware level system
  - Any z13 library function calls will abend if run on pre-z13 generation server
  - Therefore don't use the z13 version of the MASS and ATLAS libraries until the compiled application is targeted to run on an z13 server
- XL C/C++ Performance Improvements can be measured using Performance Analyzer

## SIMD Exploitation Considerations

- To use the MASS library functions instead of the standard math library functions, the MASS library must be put first in the library concatenation
  - The xlc.cfg stanza's can be modified to put the MASS library first in the sysobj attribute
  - The compiler procs can be modified to put the MASS library first in the SYSOBJ DD concatenation
- To use the libraries alongside the standard math library, the libraries can be added to the end of the library concatenation
  - This means that the MASS functions that are common with the C standard library will not be used
- POSIX(ON) is required for ATLAS to run in multi-threaded mode
  - For maximum performance


## SIMD instruction set and execution: - Business Analytics Vector Processing

### **z/OS Support includes:**

- Enablement of Vector Registers (VR)
- Use of VR when using XL C/C++ ARCH(11) and TUNE(11)
- MASS – Mathematical Acceleration Sub-System
  - A math library with optimized and tuned math functions
  - Has SIMD, vectorized, and non-vectorized version
  - Can be used in place of some of the C Standard math functions
- ATLAS (Automatically Tuned Linear Algebra Software)
  - A specialized math library that is optimized for the hardware
- LE enablement for ATLAS (for C runtime functions)
- DBX to support disassemble the new vector instructions, and to display and set vector registers
- XML SS Exploitation to use new vector processing instructions to improve performance

## SIMD Migration, and Fallback Considerations

- This is new functionality and code will have to be developed to take advantage of it
- Some mathematical function replacement can be done without code changes by inclusion of the scalar MASS library before the standard math library
  - Different accuracy for MASS vs. the standard MATH library
  - IEEE is the only mode allowed for MASS
  - **Migration Action:** Assess the accuracy of the functions in the context of the user application when deciding whether to use the MASS and ATLAS libraries
- LOADxx MACHMIG can be used to disable SIMD at IPL time
  - The MACHMIG statement of the LOADxx parmlib member was extended to allow the specification of VEF, which indicates that the Vector Extension Facility is not to be exploited, even if it is available



# IBM z13 Capacity and Performance Planning

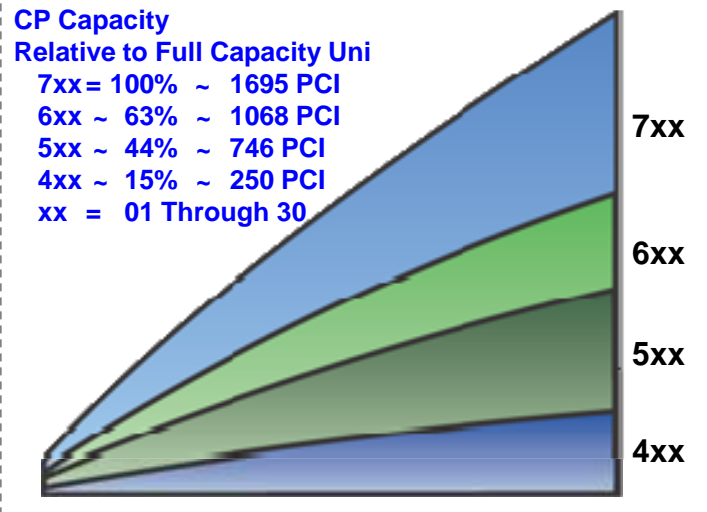


# z13 Full and Sub-Capacity CP Offerings

**CP Capacity**

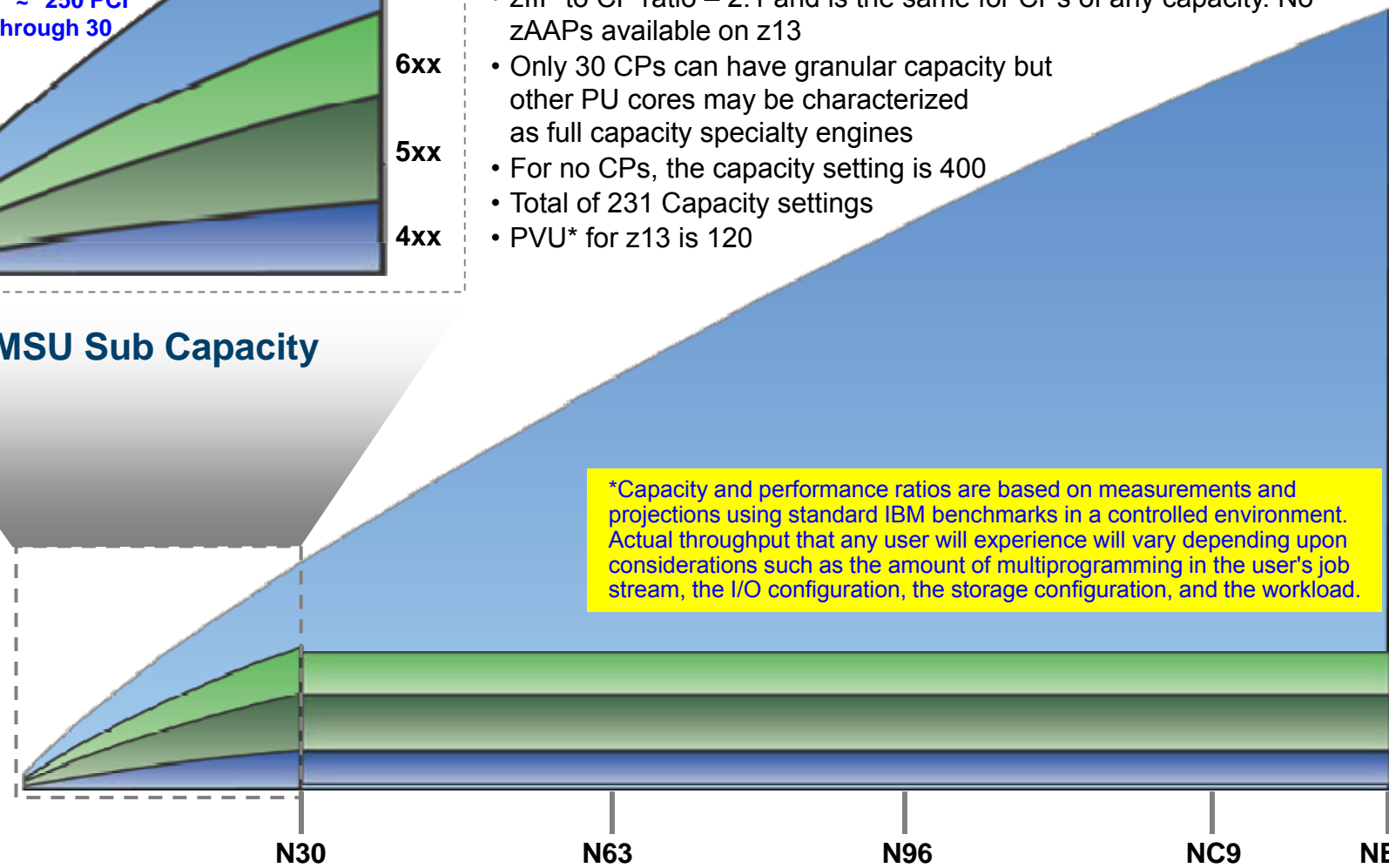
**Relative to Full Capacity Uni**

- 7xx = 100% ~ 1695 PCI
- 6xx ~ 63% ~ 1068 PCI
- 5xx ~ 44% ~ 746 PCI
- 4xx ~ 15% ~ 250 PCI
- xx = 01 Through 30



- Subcapacity CPs, up to 30, may be ordered on ANY z13 model. If 31 or more CPs are ordered all must be full 7xx capacity
- All CPs on a z13 CPC must be the same capacity
- All specialty engines are full capacity.
- zIIP to CP ratio – 2:1 and is the same for CPs of any capacity. No zAAPs available on z13
- Only 30 CPs can have granular capacity but other PU cores may be characterized as full capacity specialty engines
- For no CPs, the capacity setting is 400
- Total of 231 Capacity settings
- PVU\* for z13 is 120

**MSU Sub Capacity**



\*Capacity and performance ratios are based on measurements and projections using standard IBM benchmarks in a controlled environment. Actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload.

PVU: Processor Value Unit - [http://www-01.ibm.com/software/passportadvantage/pvu\\_licensing\\_for\\_customers.html](http://www-01.ibm.com/software/passportadvantage/pvu_licensing_for_customers.html)

## z13 Model Capacity Identifier

HW Model	Model Capacity Identifier	Comments
<b>N30</b>	700 – 730, 6nn, 5nn, 4nn	Where nn = 1 to 30
<b>N63</b>	700 – 763, 6nn, 5nn, 4nn	Where nn = 1 to 30
<b>N96</b>	700 – 796, 6nn, 5nn, 4nn	Where nn = 1 to 30
<b>NC9</b>	700 – 7C9, 6nn, 5nn, 4nn	Where nn = 1 to 30
<b>NE1</b>	700 – 7E1, 6nn, 5nn, 4nn	Where nn = 1 to 30

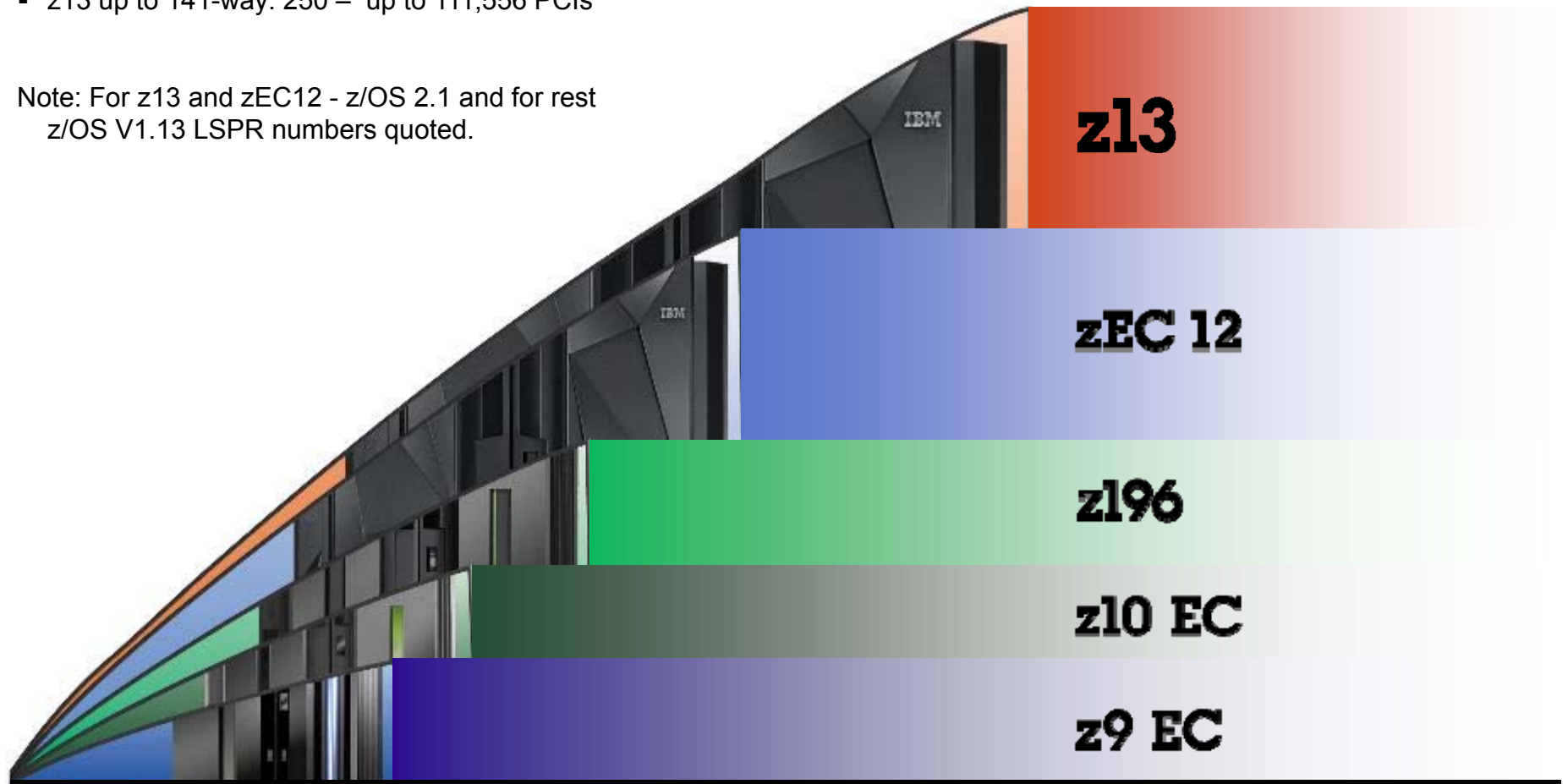
- The Model Capacity Identifier (MCI) is the value used by Independent Software Vendors (ISVs) to set the price for software billing
- It is used to indicate the number of active CPs rather than total physical PUs delivered (purchased)
- The 4/5/6/7xx MCI along with the real hardware model number and the MSU value is returned when the STSI instruction is executed

Note: For no CPs, the capacity setting is 400.

## z13 Vs zEC12 Vs z196 Vs z10 EC Vs z9 EC capacity comparison

- z9 EC up to 54-way: 193 – 18,505 PCIs.
- z10 EC up to 64-way: 214 – 31,826 PCIs
- z196 up to 80-way: 240 – 52,286 PCIs
- zEC12 up to 101-way: 240 – 78,426 PCIs
- z13 up to 141-way: 250 – up to 111,556 PCIs

Note: For z13 and zEC12 - z/OS 2.1 and for rest z/OS V1.13 LSPR numbers quoted.



# Performance Drivers with z13

## ▪ Hardware

- Memory subsystem
  - Focus on keeping data "closer" to the processor unit
    - Larger L1, L2, L3 and L4 caches
    - Improved IPC (Instructions Per Cycle)
  - 3 X configurable memory
- Processor
  - 2 X instruction pipe width, re-optimized pipe depth for power/performance
    - Improved IPC
  - SMT for zIIPs and IFLs
    - Includes metering for capacity, utilization and adjusted chargeback (zIIPs, IFLs)
  - SIMD unit for analytics
  - Up to 8 processor units (cores) per chip
- Up to 141 configurable processor units (cores)
- 3 sub-capacity settings

## ▪ HiperDispatch

- Exploits new chip configuration
- Required for SMT

## ▪ PR/SM

- 85 customer partitions (up from 60)
- Memory affinity
  - Automatically keep partition's memory and CPs on same drawer as much as possible

## PR/SM Partition Logical Processor and Memory Assignment

### ■ System z9 EC to zEnterprise EC12

- **Memory Allocation Goal:** Stripe across all the available books in the machine  
**Advantage:** Exploit fast book interconnection; spread the memory controller work; smooth performance variability
- **Processor Allocation Goal:** Assign all logical processors to one book; packed into chips of that book. Cooperate with operating system use of HiperDispatch  
**Advantage:** Optimal shared cache usage

### ■ z13

- **Memory Allocation Goal:** Assign all memory in one drawer striped across the two nodes.  
**Advantage:** Lower latency memory access in drawer; smooth performance variability across nodes in the drawer
- **Processor Allocation Goal:** Assign all logical processors to one drawer; packed into chips of that drawer. Cooperate with operating system use of HiperDispatch
- **Reality:**
  - Easy for any given partition. Complex optimization for multiple logical partitions because some need to be split among drawers

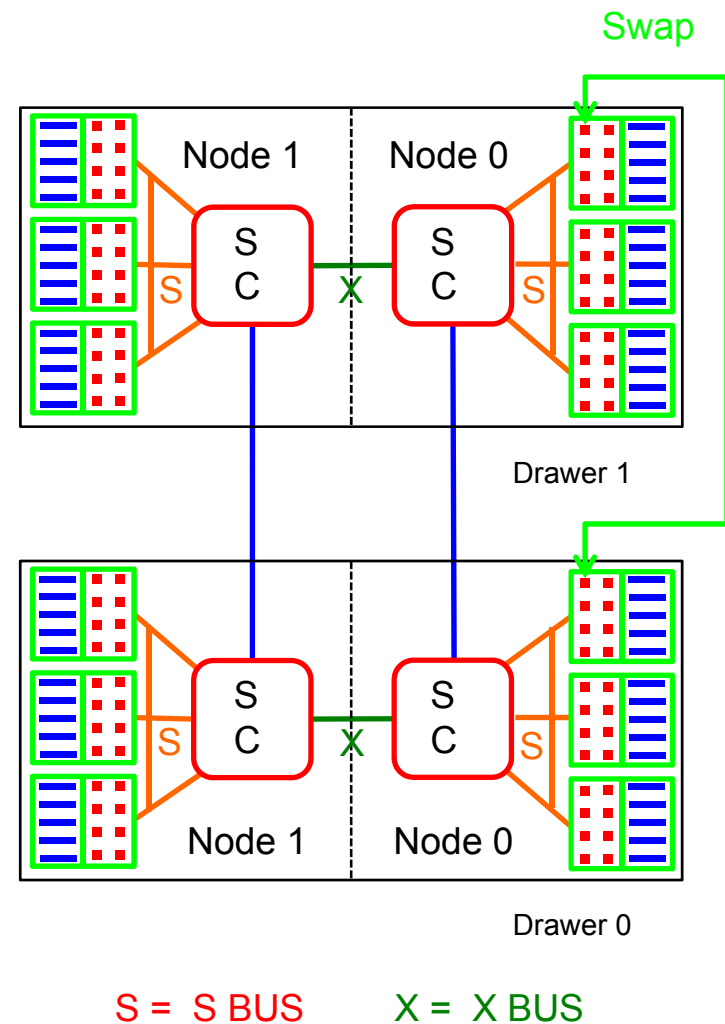
## z13 Processor and Memory Assignment and Optimization

- **Default processor assignments by POR, MES adds, and On Demand activation:**
  - Assign CPs and zIIP in low drawers working up
  - Assign IFLs and ICFs to cores on chips in “high” drawers working down
  - **Objective:** Keep “Linux Only”, “IBM zAware” and “Coupling Facility” using IFLs and ICFs “away” from “ESA/390” partitions running z/OS on CPs and zIIPs and in different drawers if possible.
- **PR/SM makes optimum available memory and logical processor assignment at activation**
  - Logical Processors specified in the Image Profile, are assigned a **core** if Dedicated or a **“home” drawer, node and chip** if Shared. Later, if it becomes a HiperDispatch “Vertical High”, a Shared Logical Processor is assigned a specific core
  - **Ideally assign all memory in one drawer with the processors** if everything “fits”
  - If memory striped across drawers, processors must also be split and vice-versa (DMR + DPR)
- **PR/SM re-optimizes resource assignment when triggered**
  - **Triggers:** Available resources changes: partition activation or deactivation or significant processor entitlement changes, dynamic memory increases or processor increases or decreases (e.g. by CBU) or MES change
  - **Resulting Actions:**
    - Examines partitions in priority order by the size of their “**processor entitlement**” (dedicated processor count or shared processor pool allocation by weight) to determine priority for optimization
    - Changes logical processor “home” drawer / node / chip assignment
    - Moves processors to different chips, nodes, drawers (**LPAR Dynamic PU Reassignment**)
    - Relocates partition memory to active memory in a different drawer or drawers using the newly optimized **Dynamic Memory Relocation (DMR)**, also exploited by **Enhanced Drawer Availability (EDA)**.
    - **If available but inactive memory hardware is present** (e.g. hardware driven by Flexible or Plan Ahead) in a drawer where more active memory would help: activate it, reassign active partition memory to it, and deactivate the source memory hardware, again using DMR.

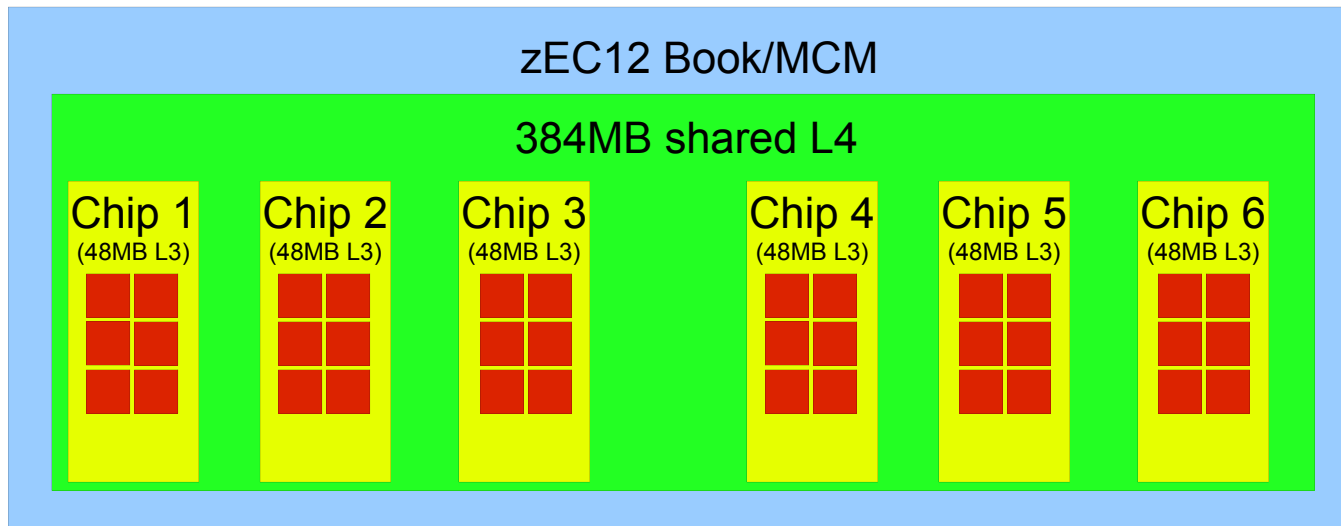
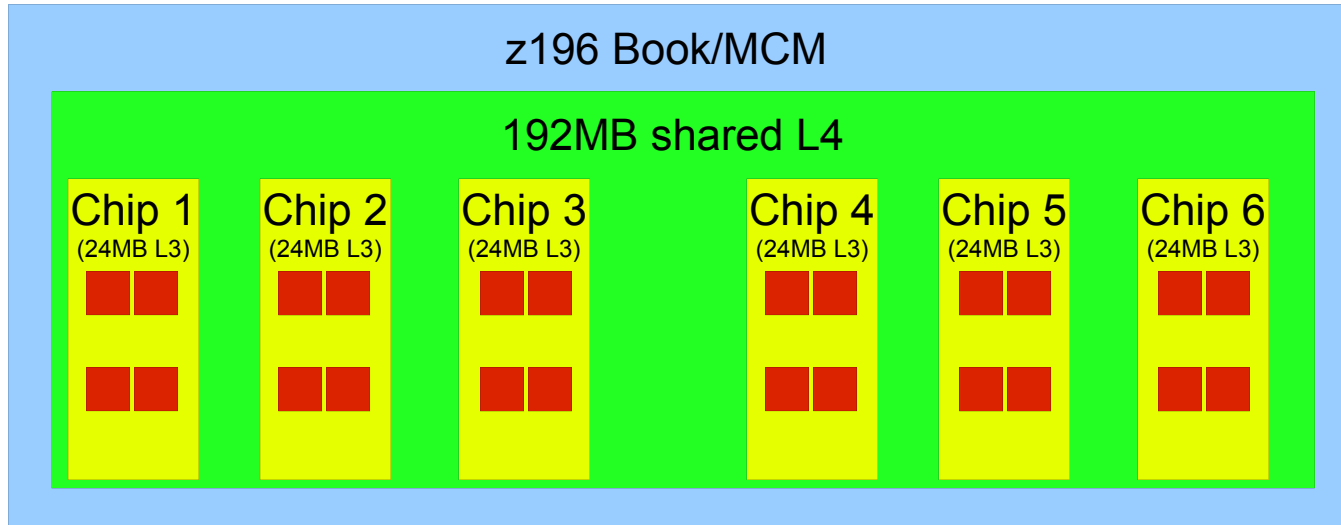
Note: (PR/SM can use all memory hardware but concurrently enables no more memory than the client has paid to use.)

## z13 LPAR Dynamic PU Reassignment

- PR/SM dynamic relocation of running **processor cores** to different physical core locations
  - CP, zIIP, IFL and ICF supported
  - Swap an active core to a core in a different PU chip in a different drawer or node
- Designed to optimize physical processor location for the current LPAR's logical processor configuration:
  - Better L3 and L4 cache reuse
  - Move processor to partition memory
- **Triggers:** Partition activation/deactivation, machine upgrades/downgrades, logical processors configured on/off
- Designed to provide the most benefit for:
  - Multiple drawer machines
  - Dedicated partitions and wide partitions with HiperDispatch active

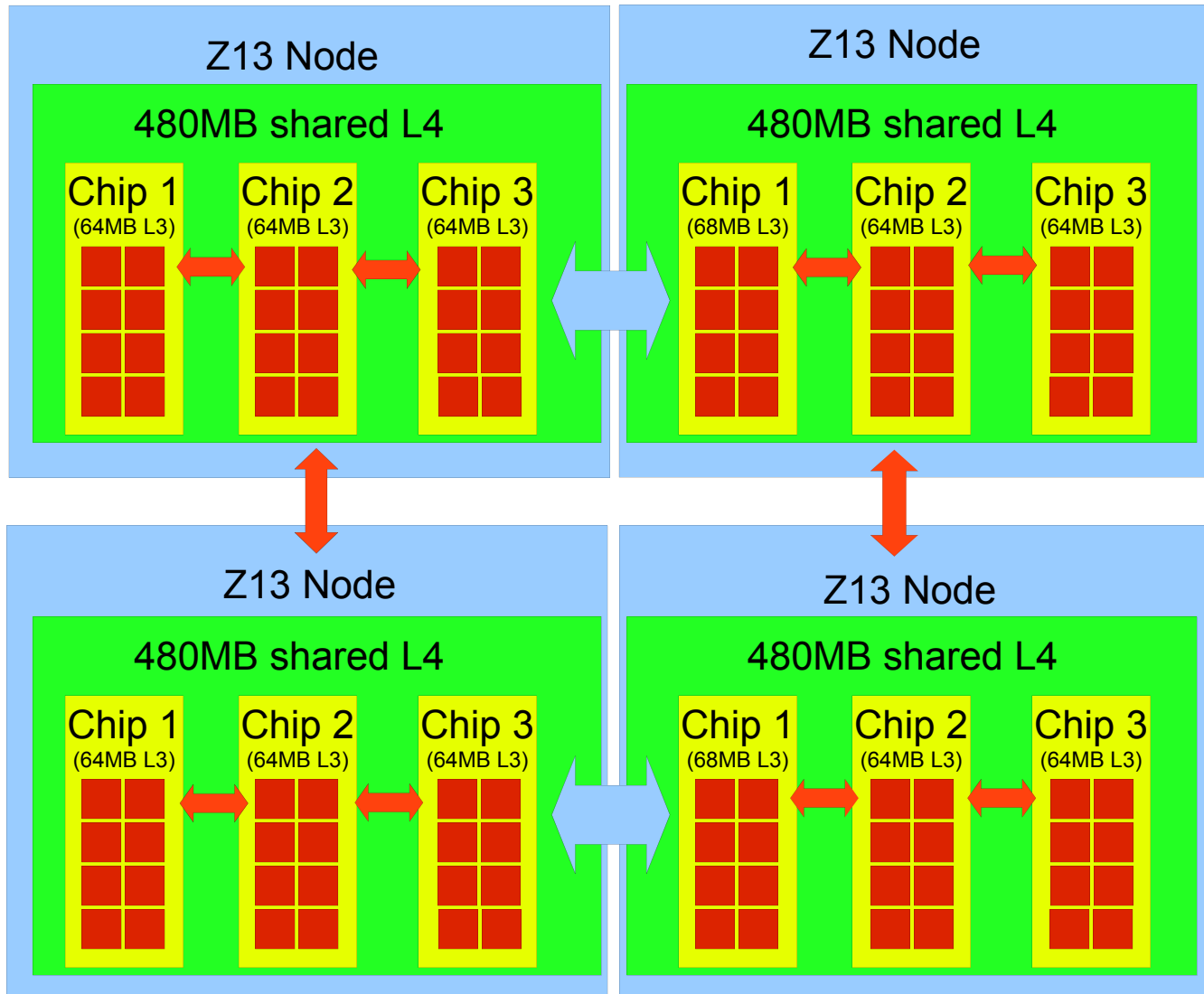


# Cache alignment – previous systems z





# Cache alignment – z13



## PR/SM Dispatching on z13

- Dispatch an “on and ready” Dedicated Logical Processor
  - To its **dedicated home physical processor core**
  
- Dispatch the “most deserving” on and ready Shared Logical Processor
  - To the best choice **shared physical processor core of the same type**
  - **Favor**: Home core, chip, node, and drawer to maximize cache reuse
  - **Goals**: Allocate shared processor resource according shares calculated from weights. Work with HiperDispatch (e.g. “Vertical High” or “Parked” logical processors). Minimize performance variability. Maximize capacity.
  - **Dispatching work to SMT threads is done by z/OS for zIIPs or z/VM for IFLs if enabled by PTFs and activated with operating system parameters.**  
Note: z/OS messages may refer to logical processors as “CPUs” or “Cores”
  
- PR/SM cooperates but does NOT dispatch SMT threads

## LPAR Dispatcher Selection Algorithm

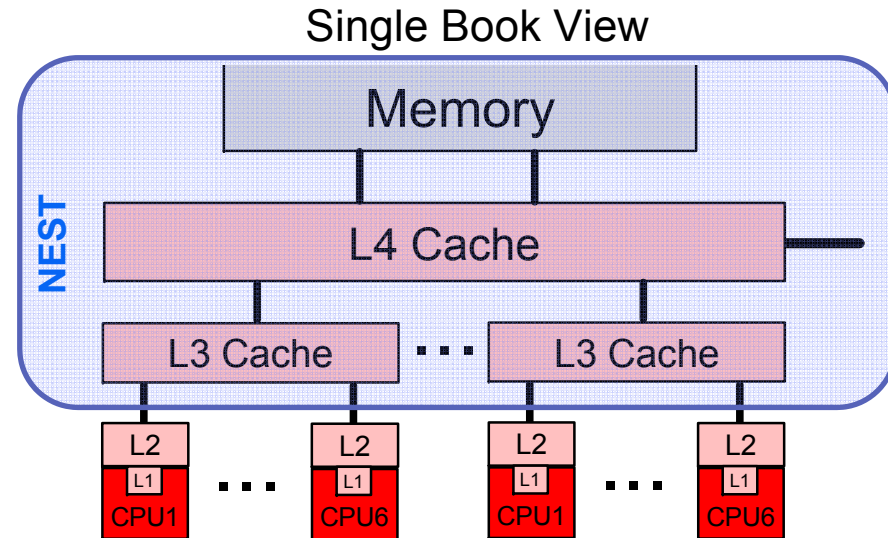
- Attempt assignment to an available physical CP (in wait)
  - CP last run on.. is available on home chip
  - Choose available CP on home chip
  - Choose available CP on home node
  - CP last run on is available on home drawer
  - Choose available CP on home drawer
  - Choose available CP on “sister” drawer(s)
    - “sisters” is the set of drawers where memory is allocated for the partition
  
- Search for lowest priority dispatched work to displace in this order:
  - CPs on home chip
  - CPs on home node
  - CPs on home drawer
  - CPs on “sister” drawer(s)

The entire CPC (non-sisters) is not immediately searched. After some delay, the above searches are then expanded to include everything, attempting to trade off higher cache hit rate for transient expansion. Synergy here with z/OS HiperDispatch decisions.

# z13 versus zEC12 Hardware Comparison

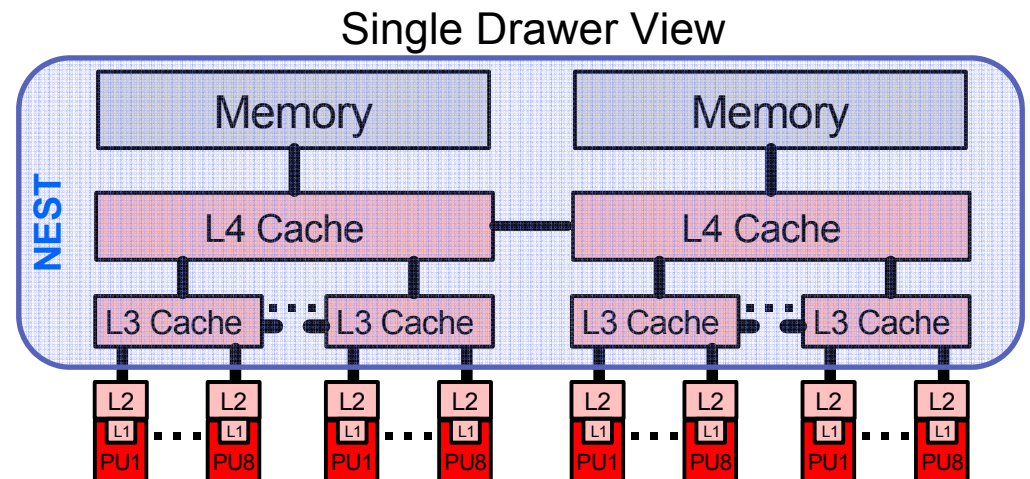
- **zEC12**

- CPU
  - 5.5 GHz (**1514 PCI**)
  - Enhanced Out-Of-Order
- Caches
  - L1 private 64k i, 96k d
  - L2 private 1 MB i + 1 MB d
  - L3 shared 48 MB / chip
  - L4 shared 384 MB / book



- **z13**

- CPU
  - 5.0 GHz (~**1695 PCI**)
  - Major pipeline enhancements
- Caches
  - L1 private 96k i, 128k d
  - L2 private 2 MB i + 2 MB d
  - L3 shared 64 MB / chip
  - L4 shared 480 MB / node
  - plus 224 MB NIC



## z13 Performance Capacity Highlights

- **Full speed capacity models (7xx) capacity ratio to zEC12**
  - Up to 40% more total system capacity compared to the zEC12 (141w z13 versus 101w zEC12)
- **Capacity / models**
  - 401: 0.15x (250 PCI)
  - 501: 0.44x (746 PCI)
  - 601: 0.63x (1068 PCI)
  - 701: 1.00x (1695 PCI) - (full speed)
- **SMT capacity option**
  - IFLs and zIIPs can choose to run 2 HW threads per core
    - Controlled by OS parm at the LPAR level
    - Added HW threads appear as additional processors to the OS's
    - Default is single thread
  - Likely wide range in capacity improvement per core over single thread: 10% to 40%
- **Variability amongst workloads**
  - Moving to z13 expected to be more variable than last few migrations
  - Potential Sources of Variability
    - Workload interaction with Processor Design
    - PR/SM placement of CPs and memory for an LPAR
    - LPAR placement to minimize spread over chips, nodes and drawers – dependent on PR/SM optimization
    - A “Very Large” single LPAR that exceeds a drawer (>30 CPs) – very few worldwide and predictable
    - LPAR Memory larger than currently configured on drawer (LPAR > Total Memory / Drawers ) – predictable

## LSPR: Performance Showcase for z Processors

- IBM z System provides capacity comparisons among processors based on a variety of measured workloads which are published in the [Large System Performance Reference \(LSPR\)](#)
  - <https://www-304.ibm.com/servers/resourceink/lib03060.nsf/pages/lspindex>
- Old and new processors are measured in the same environment with the same workloads at high utilizations
- Over time, workloads and environment are updated to stay current with customer profiles
  - Old processors measured with new workloads / environment may have different average capacity ratios compared to when they were originally measured
- LSPR presents capacity ratios among processors
- Single number metrics include PCI/MIPS, MSUs, and SRM Constants
  - MIPS is based on the ratios for the “Average” workload category with a median customer LPAR configuration

## What's New in the LSPR for z13

- **Workload updates**
  - Up leveled software - **z/OS 2.1**, subsystems, compilers
  - Minor tweaks to three hardware-characteristic-based workload categories
    - Based on CPU MF data from customers' z196 to zEC12 migrations
- **HiperDispatch** continues to be turned on for all measurements
  - Particularly valuable on smaller n-way configurations starting with z196 due to sensitivity to L3 chip-level cache
- LSPR will only publish single thread capacity in the multi-image table
  - Multi-Image (MI) table
    - Median LPAR configuration for each model based on customer profile
      - Including effect of average number of ICFs and IFLs
    - Most representative for vast majority of customers
    - Basis for single-number metrics PCI/MIPS, MSUs, SRM constants
- **zPCR Used to Model Any Reasonable LPAR Configuration**
  - Customized LPAR configurations and workloads (as always)
  - SMT capacity effect will be included via a user controlled dial by partition
    - Default effect established for zIIPs and IFLs
    - Set dial to reflect the estimated capacity increase when using two threads over one thread
    - Pre-install guidance in setting dial to be provided based on internal testing and eventual field experience
    - Post-install guidance in setting dial from metering data available in RMF and the z/VM Performance Report

# LSPR RNI\*-based Workload Categories

## Validated and Now Default in zPCR and zCP3000

- Historically, LSPR workload capacity curves (primitives and mixes) had application names or been identified by a "software" captured characteristic
  - for example, CICS, IMS, OLTP-T, CB-L, LoIO-mix, TI-mix, etc
- However, capacity performance is more closely associated with how a workload is using and interacting with a processor "hardware" design
- With the availability of CPU MF (SMF 113 or VM Monitor) data starting with z10, the ability to gain insight into the interaction of workload and hardware exists
- The LPSR for z196 introduced three new workload categories which replaced all prior primitives and mixes
  - LOW, AVERAGE, HIGH Relative Nest Intensity (RNI)
- RNI reflects the distribution and latency of sourcing from shared caches & memory
- Migrations to z196, z114, zEC12, and zBC12 have validated this approach
  - Analyzed customer LPARs before and after migration
- RNI-based methodology for workload matching is now the default in zPCR and zCP3000

RNI\*=Relative Nest Intensity



## z System - CPC - Processor Subsystem - NEST

- CPC – Central Electronic Complex
  - Processors, data storage, memory...
  - Processor Subsystem
    - Multiple Nodes of multiple processors

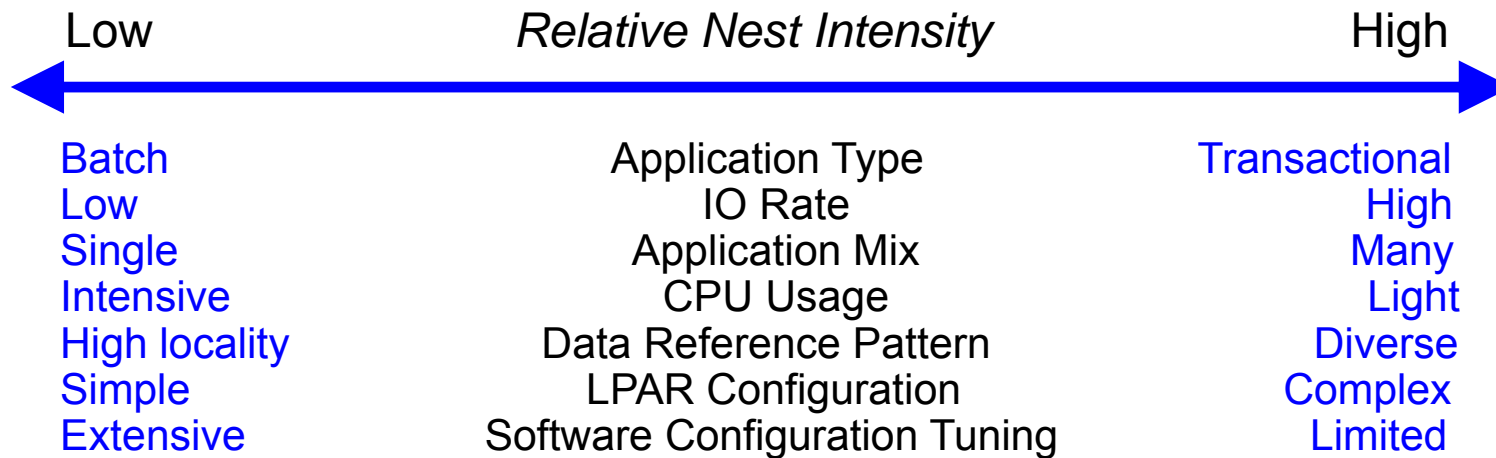
- **What is the “NEST”..**

- ✓ Multi-level Cache Hierarchy
  - L1, L2, L3, L4+NIC
  - Inclusive Cache Policy
- ✓ System Coherency Management
  - Strongly Ordered System Architecture
  - MESI (Modified, Exclusive, Shared, Invalid)
- ✓ Series of interconnect networks
  - X, S, and A Fabric Buses
  - Address/Command, Responses and Data
- ✓ Memory Subsystem interface functions
  - MCU, DIMMs, RAIM
- ✓ IO Hub interface functions
  - GX+, support for legacy IO and PCIe Gen3

**NEST**

# Most Influential Factor Underlying Workload Capacity Curves is *Relative Nest Intensity*

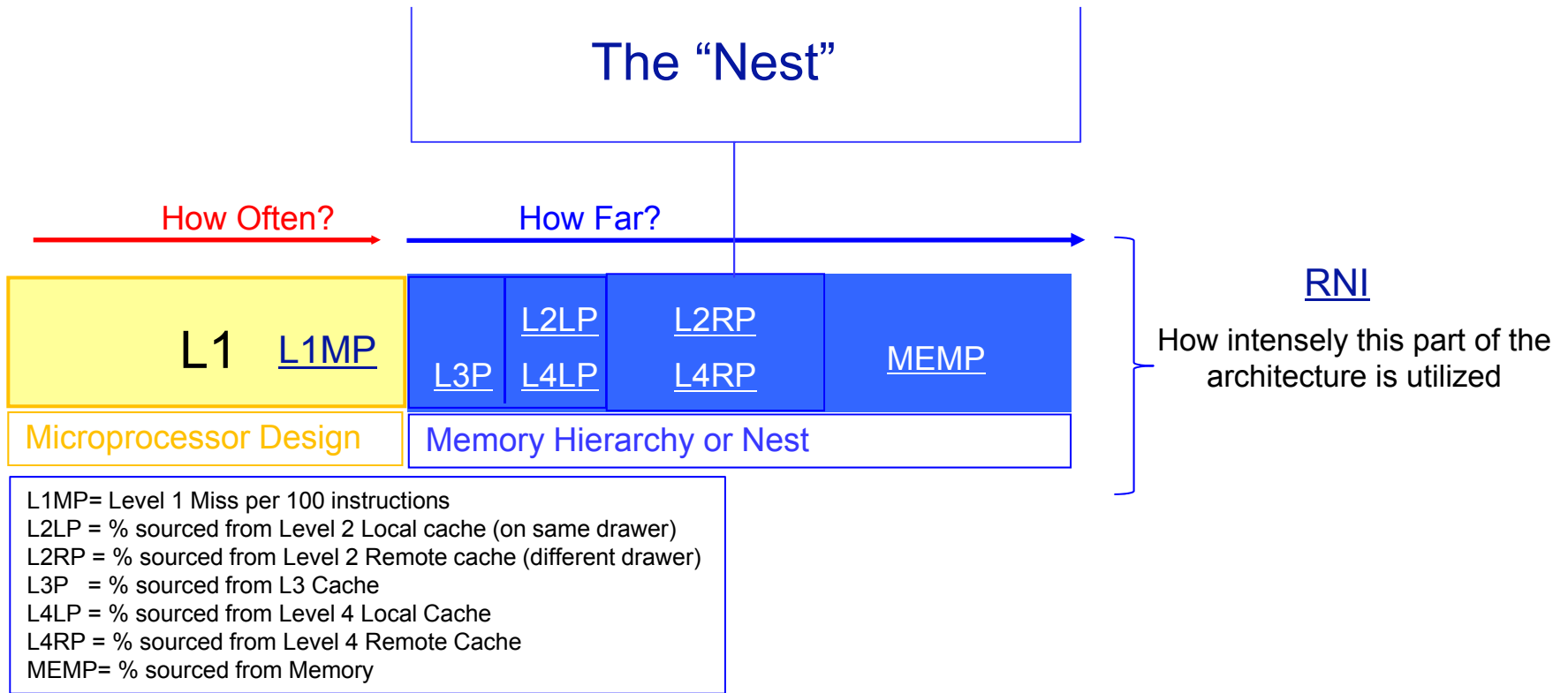
- Many factors influence a workload’s capacity curve
- What they are actually affecting is the workload’s Relative Nest Intensity (RNI)
- The net effect of the interaction of all these factors determines the capacity curve
- The chart below indicates the trend of the effect of each factor but is not an absolute
  - For example, some batch will have high RNI while some transactional workloads will have low
  - For example, some low IO rate wklds will have high RNI, while some high IO rates have low



# Relative Nest Intensity (RNI) Metrics

- Reflects the distribution and latency of sourcing from shared caches and memory

## Relative Nest Intensity



Customer	SYSID	MON	DAY	CPI	PRBSTATE	Est Instr Cmplx	Est Finite CPI	Est SCPL1M	L1MP	L15P	L2LP	L2RP	MEMP	Rel Nest Intensity	LPARCPU	Eff GHz
		Minimum		3.1	1.1	2.1	0.9	50.6	1.3	48.6	5.6	0.0	2.2	0.4	14.4	
		Average		72	312	3.2	3.9	101.4	3.9	68.9	21.2	1.6	8.3	0.9	3763	
		Maximum		120	67.1	5.6	8.6	104.9	6.9	82.8	32.9	6.9	20.2	1.8	14423	4.40

## RNI-based LSPR Workload Decision Table

L1MP	RNI	LSPR Workload Match
< 3	≥ 0.75 < 0.75	AVERAGE LOW
3 to 6	1.0 0.6 to 1.0 < 0.6	HIGH AVERAGE LOW
> 6	≥ 0.75 < 0.75	HIGH AVERAGE

Current table applies to z10 EC, z10 BC, z196, z114, zEC12, zBC12 and z13 CPU MF data

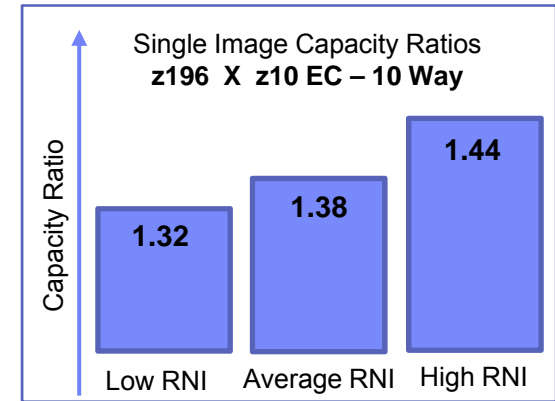
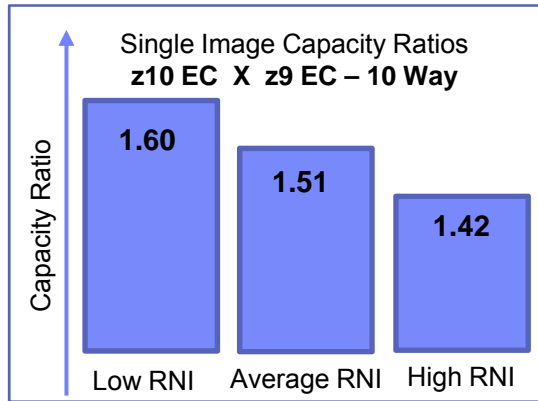
L1MP= Level 1 Miss per 100 instructions

RNI value is calculated using a specific formula for each CPU family

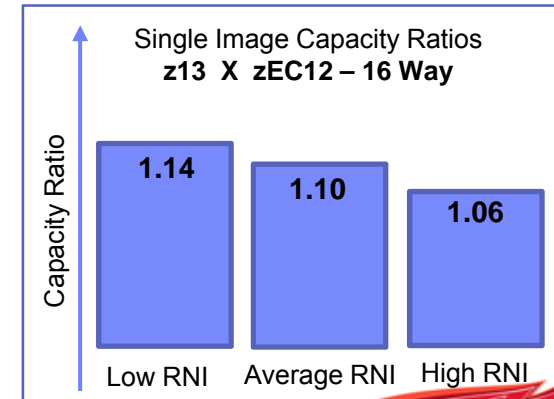
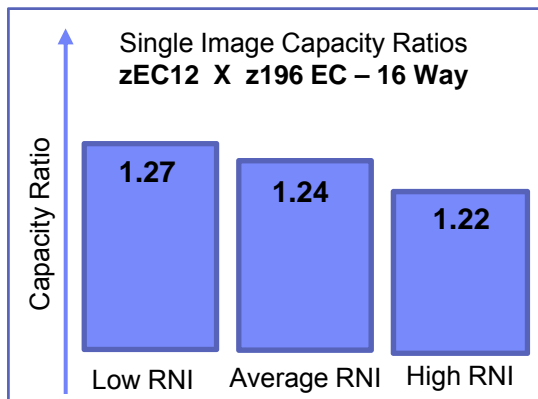
## LSPR Workload Categories

- Various combinations of prior workload primitives are measured on which the new workload categories are based
  - Applications include CICS, DB2, IMS, OSAM, VSAM, WebSphere, COBOL, utilities
- **Low** (relative nest intensity)
  - Workload curve representing light use of the memory hierarchy
  - Similar to past high N-way scaling workload primitives
- **Average** (relative nest intensity)
  - Workload curve expected to represent the majority of customer workloads
  - Similar to the past LoIO-mix curve
- **High** (relative nest intensity)
  - Workload curve representing heavy use of the memory hierarchy
  - Similar to the past DI-mix curve
- zPCR and zCP3000 extend published categories to add granularity to LSPR workloads
  - **Low-Avg**
    - 50% Low and 50% Average
  - **Avg-High**
    - 50% Average and 50% High

# LSPR Workloads – Single Image Capacity Ratios

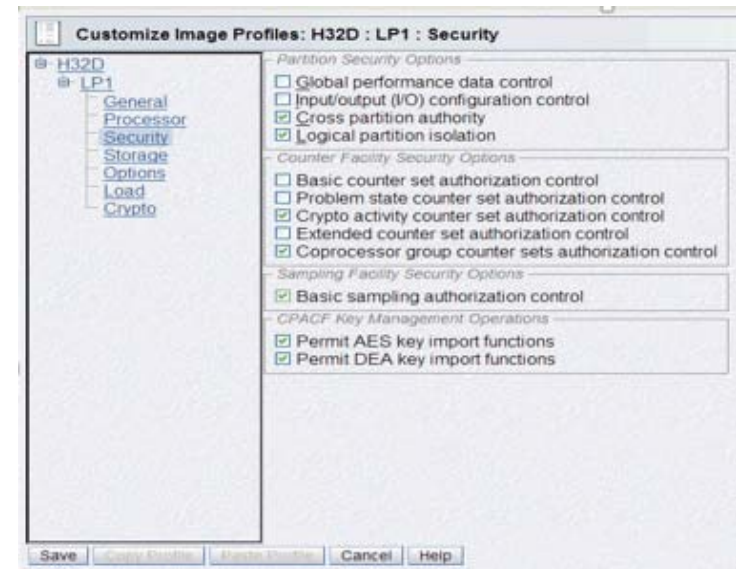


Family	GHz	L1	L 1.5	L2	L3	L4
z9	1.7	256Ki + 256Kd	-	40 MB	-	-
z10	4.4	64Ki + 128Kd	3 MB	48 MB	-	-
z196	5.2	64Ki + 128Kd	-	1.5 MB (i+d)	24 MB	192 MB
zEC12	5.5	64Ki + 96Kd	-	1 MB (d) 1 MB (i)	48 MB	384 MB
z13	5.0	96Ki + 128Kd	-	2 MB (d) 2 MB (i)	64 MB	480 MB (224 MB NIC)



## - CPUMF – CPU Measurement Facility

- Introduced with z10 and available in later processors starting with z/OS V1R10
- Facility to provide hardware instrumentation data for production systems
- Two Major components
  - Counters
    - Cache and memory hierarchy information
    - SCPs supported include z/OS and z/VM
    - Sampling - Instruction time-in-CSECT
- z/OS HIS Component writes SMF 113 records
  - Gathered on an LPAR basis
  - Writes SMF113 records e
- New z/VM Monitor Records
  - Gathered on an LPAR basis – all guests are aggregated
  - Writes new Domain 5 (Processor) Record 13 (CPU MF Counters) records
- **Minimal overhead**
- **Additional Information**
  - ***CPU MF – 2012 Update and WSC Experiences for z/OS***
    - <http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TC000066>
  - ***How to Collect CPU Measurement Facility data for z/VM***
    - <http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TD105949>



## z/OS Steps to Enable CPU MF Counters

- **1 - Configure the processor to collect CPU MF**  
Update the LPAR Security Tabs, can be done dynamically
  
- **2 - Set up HIS and z/OS to collect CPU MF**  
Set up HIS Proc  
Set up OMVS Directory - required  
Collect SMF 113s via SMFPRMxx
  
- **3 - Collect CPU MF COUNTERs**  
Start HIS  
Modify HIS:  
“F HIS,B,TT='Text',PATH='/his/',CTRONLY,CTR=(B,E),SI=SYNC”  
  
– Recommend to start HIS, Modify for Counters, and continuously run



## Additional Customer Value with CPU MF Counters data

- **Counters can be used as a secondary source to:**
  - Supplement current performance data from SMF, RMF, DB2, CICS, etc.
  - Help understand why performance may have changed
  - Supported by many software products including Tivoli TDSz
  
- **Some examples of usage include:**
  - HiperDispatch Impact
  - Configuration changes (Additional LPARs)
  - 1 MB Page implementation
  - Application Changes (e.g. CICS Threadsafe Vs QR)
  - Estimating Utilization Effect for capacity planning
  - GHz change in Power Saving Mode
  - Crypto CPACF usage

## Recommendations for All Customers

- **Before and After CPU MF Counters data will be critical to determine the source of variation for workloads that do encounter it**
  - Ensure the CPU MF data is captured and kept for analysis
  - For z/VM “Before” and “After” peak hour data must be written to disk
- **Critical Migration Action for every z13 candidate (z/OS and z/VM)**
  - CPU MF Counters must be enabled on their **current** processor
  - CPU MF Counters must be enabled on their z13
- **Take Action to Validate CPU MF is implemented or get a plan started to implement CPU MF**
  - See CPU MF Counters Enablement Resources for detailed step by step instructions
  
  - Note: for z/OS z13 “After” partitions, SMF 99 subtype 14 (HiperDispatch Topology) may also be required

## CPU MF Counters Enablement Resources

- CPU MF Webinar Replays and Presentations
  - <http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/PRS4922>
- z/OS CPU MF - “Detailed Instructions” Step by Step Guide
  - <http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TC000066>
- z/VM Using CPU Measurement Facility Host Counters
  - <http://www.vm.ibm.com/perf/tips/cpumf.html>

## Capacity Planning Tools

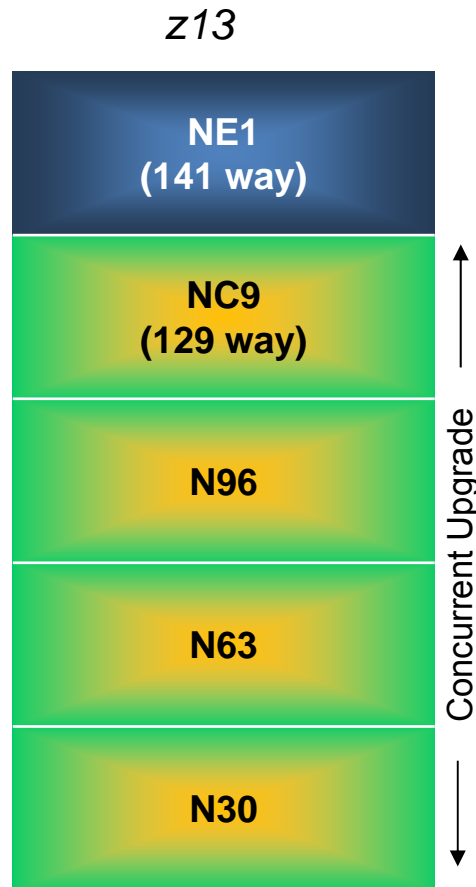
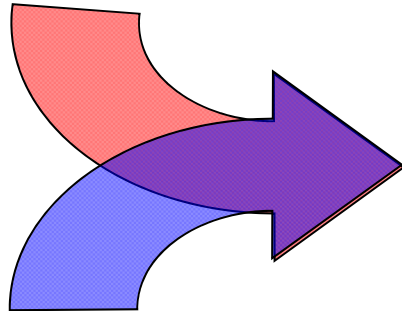
- **zPCR\*** Processor Capacity Reference for IBM z System
- **zCP3000** Performance Analysis and Capacity Planning for IBM z System
- **zTPM\*** Tivoli Performance Modeler
- **zBNA\*** IBM z System Batch Network Analyzer

\*Note: Available to customers

# IBM z13 Upgrades



# z13 System Upgrades



- z13 to z13 model upgrades
  - Upgrade of z13 Models N30, N63, N96 and NC9 to NE1 is disruptive
  - When upgrading to z13 Model NE1, all the CPC Drawers are replaced
  - Conversion\* from Radiator-based air to Water cooled or Water to Radiator-based air cooling not available
- Any\* z196 to any z13
- Any\* zEC12 to any z13
- Feature conversion of installed zAAPs to zIIPs (default) or another processor type
- For installed On Demand Records, change temporary zAAPs to zIIPs. Stage the record
- When a z196 with a zBX Model 002 or zEC12 with a zBX Model 003 is upgraded to z13, the zBX is detached from the CPC and converted to a Model 004. The zBX becomes a Node without a CPC. Additional planning required and conditions apply

**\*Note:**  
 Air to Water Conversions

- Conversions from z196 & zEC12 Air to z13 Water will be supported with a frame roll as was done on z196 & zEC12.
- Upgrading from a z13 Radiator based air to a z13 Water will only be offered via a Migration offering (8P2979). RPQ 8P2979 is ONLY available on initial orders and not available as an MES.

Water to Air conversions

- Conversions from water to air are NOT supported for either z196, zEC12 or z13 to z13.

## z13 Upgrade Paths

- z196 and z12EC systems will be upgradeable to z13
- The table below identifies vertical upgrade paths within the z13 (CPC Drawer adds)
- Within the table:
  - X indicates a path that is announced and configurable through eConfig
  - Model upgrades are handled as feature conversions

z13 to z13 (2964 to 2964) Upgrade Paths

	N30	N63	N96	NC9	NE1
N30	-	X	X	X	X
N63		-	X	X	X
N96			-	X	X
NC9				-	X
NE1					-

## z13 Model and Memory Upgrade Summary

- Model upgrades allowed within z13, from zEC12 and z196
- Model downgrades will NOT be allowed
  - Cross machine downgrades will be allowed during the conversion only from zEC12 and z196
- System capacity upgrades and downgrades allowed within z13, from zEC12 and z196
- Frame roll MES for upgrades from zEC12 and z196 to z13
  - New frames are shipped
  - MES will ship with any I/O drawers being carried forward
    - New PCIe I/O drawers will be supplied with the MES
- Memory:
  - z13 to z13 – Upgrades = Yes, Downgrades = No
  - zEC12 to z13 – Upgrades = Yes, Downgrades = No
  - z196 to z13 – Upgrades = Yes, Downgrades = No



## Comparisons, z196, zEC12, and z13 entry models

	<b>z196 Model H20</b>	<b>zEC12 Model H20</b>	<b>z13 Model N30</b>
<b>Uniprocessor Performance</b>	1514 MIPs	1514 MIPs	1695 MIPs
<b>z/OS Capacity</b>	240 – 21380 MIPs	240 – 21380 MIPs	250 – 23929 MIPs
<b>Total System Memory</b>	704 GB	704 GB	2560 GB
<b>Configurable Engines</b>	20	20	30
<b>Configurable CPs</b>	0 - 20	0 - 20	0 - 30
<b>LPARS/CSS</b>	60/4	60/4	85/6
<b>HiperSockets</b>	32	32	32
<b>I/O Cages/ I/O drawers/ PCIe I/O drawers</b>	1/2/5	1/2/5	0/2/5
<b>I/O slots per Cage/ I/O drawers/ PCIe I/O drawers</b>	28/8/32	28/8/32	0/8/32
<b>FICON® Channels</b>	256	256	256
<b>OSA Ports (10GbE/1GbE/1000BASE-T)</b>	48/96/96	48/96/96	48/96/96
<b>ESCON® Channels</b>	0	0	0
<b>IFB host bus Bandwidth PCIe GenX Bandwidth</b>	6.0 GB/sec (IFB) 8.0 GB/sec (PCIe-Gen2)	6.0 GB/sec (IFB) 8.0 GB/sec (PCIe-Gen2)	6.0 GB/sec (IFB) 16.0 GB/sec (PCIe-Gen3)
<b>ICB-4/ISC-3<sup>(8)</sup>/PSIFB/ICA-SR</b>	0 <sup>(5)</sup> /48/16 - 32	0 <sup>(5)</sup> /48/16 - 32	0 <sup>(5)</sup> /48/16 - 32
<b>zIIP/zAAP Max Qty (MIPs)</b>	13 (with Max of 7 CPs)	13 (with Max of 7 CPs)	20 (with Max of 10 CPs)
<b>IFL / ICF Maximum Qty</b>	20 (21380 MIPs)	20 (21380 MIPs)	30 (23929 MIPs)
<b>Capacity Settings</b>	80	80	120
<b>Upgradeable</b>	zEC12 H43, H66, H89, HA1 Radiator and Water Cooled	zEC12 H43, H66, H89, HA1 Radiator and Water Cooled	Z196 and zEC12 Radiator and Water Cooled
<b>Power Consumption</b>	7.7 – 13.3 KW	7.7 – 13.3 KW	5.4 – 13.1 KW



**■ Questions ?**

– *Ewerson Palacio*

*bird@br.ibm.com*

– *Frank Packheiser*

*F.Packheiser@de.ibm.com*

– *Parwez Hamid*

*pnh@us.ibm.com*

**IBM z13**

Reinventing enterprise IT  
for digital business

## zIIP to CP Ratio Summary:

	<b>CP Ratio</b>	<b>Conversions</b>	<b>Override default</b>	<b>Comments</b>
<b>New Build</b>	2:1	N/A	N/A	
<b>Upgrade</b>	Can exceed 2:1	zAAPs to zIIPs automatically by default	Yes. Override default to other PU types permitted	
<b>Upgrade CBU</b>	Can exceed 2:1	zAAPs to zIIPs automatically by default	No. Override default to other PUs not permitted	New record (same record id) is “staged” and not “installed”
<b>Upgrade On/Off CoD</b>	2:1	Record not migrated	N/A	Order new On/Off CoD record(s)