

# IBM System z13 Overview

**Dr. Fadi Busaba**  
[busaba@us.ibm.com](mailto:busaba@us.ibm.com)  
**Adam Collura**  
[collura@us.ibm.com](mailto:collura@us.ibm.com)



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

\*BladeCenter®, CICS®, DataPower®, DB2®, e business(logo)®, ESCON, eServer, FICON®, IBM®, IBM (logo)®, IMS, MVS, OS/390®, POWER6®, POWER6+, POWER7®, Power Architecture®, PowerVM®, PureFlex, PureSystems, S/390®, ServerProven®, Sysplex Timer®, System p®, System p5, System x®, z Systems®, System z9®, System z10®, WebSphere®, X-Architecture®, z13™, z Systems™, z9®, z10, z/Architecture®, z/OS®, z/VM®, z/VSE®, zEnterprise®, 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 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 Sync 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 Sync 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.

## Glossary

<b>ASIC</b>	Application-specific integrated circuit
<b>BPH</b>	Bulk Power Hub
<b>CCA</b>	Common Cryptographic Architecture - IBM software that enables a consistent approach to cryptography on major IBM computing platforms
<b>CPC Drawer</b>	CPC drawer refers to the packaging of the PU and SC SCMs, Memory and PCIe Gen3, ICA-SR and IFB fanouts
<b>CS5</b>	Coupling Short Reach Generation 5 - CHPID type on z13 for ICA-SR short reach coupling links
<b>FPGA</b>	Field-programmable gate array
<b>IBM zAware</b>	IBM z Advanced Workload Analysis Reporter. Provides near real-time detection of anomalous situations in the system, based on the system's past behavior and continuous monitoring.
<b>ICA SR</b>	Integrated Coupling Adapter
<b>I/O Drawer</b>	I/O drawer connected to InfiniBand fanouts supporting the 6 GBps InfiniBand I/O interconnect. For z13, FICON Express8 is the only I/O feature supported in this drawer
<b>KVM</b>	Kernel-based Virtual Machine - Open source software providing a full virtualization solution for Linux
<b>Node</b>	A Node can be a z13 CPC or/and a standalone zBX Model 004 in an Ensemble. For prior generation systems it's a zEC12, zBC12, z196 or z114 and any optionally attached zBX. A node can be a member of only one ensemble
<b>PCIe I/O Drawer</b>	PCIe I/O drawer connected to PCI Express Generation 2 (PCIe Gen2) 8 GBps I/O interconnect infrastructure introduced with z196/z114 or PCI Express Generation 3 (PCIe Gen3) 16 GBps PCIe I/O interconnect infrastructure introduced with z13
<b>RAIM</b>	Redundant array of independent memory (RAIM). A new technology introduced with z196 designed to provide protection at the direct random access memory (DRAM), dual inline memory module (DIMM), and memory channel level
<b>RDMA</b>	Remote direct memory access
<b>RG</b>	Resource Group
<b>RoCE</b>	RDMA over Converged Enhanced Ethernet
<b>SCH</b>	System Control Hub
<b>SCM</b>	Single Chip Module. For z13, these can be either the Processor Unit (PU) or System Controller (SC) modules
<b>SIMD</b>	Single Instruction Multiple Data - Vector processing model providing instruction level parallelism, benefits workloads such as analytics and mathematical modeling
<b>SLC</b>	Separately licensed code. Internal zBX code that is licensed separately from the zBX's LIC
<b>SMT</b>	Simultaneous multithreading - Architectural concept of a core, which multithreading is enabled, comprises a group of CPUs (sometimes called threads)
<b>SMC-R</b>	Shared Memory Communications – Remote Direct Memory Access
<b>zEDC</b>	zEDC Express - Hardware feature for z13, zEC12 and zBC12. Integrated solution with software capability of zEDC in z/OS V2.1 for compression acceleration
<b>zHPF</b>	High Performance FICON for z Systems

# System z: Integrated by design



# IBM z Systems High End Generations

N-4



### z9 Enterprise Class

- Announced 7/2005
- Withdrawn 6/30/2010
- Chip: 2 core, 1.7 GHz
- Up to 54 client cores
- CP, IFL, ICF, zAAP, zIIP
- Single thread
- zIIP-zAAP to CP ratio 1x1
- Uni MIPS: 560
- Max MIPS: 18,505
- Max mem 512 GB - HSA
- Max/LPAR: 512 GB - HSA
- LCSS: 4, LPARs: 60
- Subchannel Sets: 2/LCSS
- Max I/O slots: 84
- Max FICON channels: 336
- Max FICON Express4 (GA2)
- Max OSA Ports: 48
- OSA-Express2
- Crypto Express2
- Coupling: ISC3, IFB, PSIFB:12x SDR

N-3



### z10 Enterprise Class

- Announced 2/2008
- Withdrawn 6/30/2012
- Chip: 4 core, 4.4 GHz
- Up to 64 client cores
- CP, IFL, ICF, zAAP, zIIP
- Single thread
- zIIP-zAAP to CP ratio 1x1
- Uni MIPS: 902
- Max MIPS: 31,826
- Max mem 1.5 TB
- Max per LPAR: 1 TB
- LCSS: 4, LPARs: 60
- Subchannel Sets: 2/LCSS
- Max I/O slots: 84
- Max FICON channels: 336
- FICON Express4
- Max OSA Ports: 96
- OSA-Express3
- Crypto Express3 (GA3)
- Coupling: ISC3, IFB
- PSIFB: 12x DDR, 1x DDR
- ASHRAE Class A1

N-2



### zEnterprise 196

- Announced 7/22/2010
- Withdrawn 6/30/2014
- Chip 4 core, 5.2 GHz
- Up to 80 client cores
- CP, IFL, ICF, zAAP, zIIP
- Single thread
- zIIP-zAAP to CP ratio 1x1
- Uni MIPS: 1,202
- Max MIPS: 52,286
- Max mem 3 TB (RAIM)
- Max per LPAR: 1 TB
- LCSS: 4, LPARs: 60
- Subchannel Sets: 3/LCSS
- Max I/O Slots: 160\*
- Max FICON channels: 320
- FICON Express8S (GA2)
- Max OSA Ports: 96
- OSA-Express4S (GA2)
- Crypto Express3
- Coupling: ISC3
- PSIFB: 12x DDR, 1x DDR
- ASHRAE Class A1

N-1



### zEnterprise EC12

- Announced 8/28/2012
- Chip: 6 core, 5.5 GHz
- Up to 101 client cores
- CP, IFL, ICF, zAAP, zIIP
- Single thread
- zIIP-zAAP to CP ratio 2x1
- Uni MIPS: 1,514
- Max MIPS: 78,426
- Max mem 3 TB (RAIM)
- Max per LPAR: 1 TB
- LCSS: 4, LPARs: 60
- Subchannel Sets: 3/LCSS
- Max I/O Slots: 160\*
- Max FICON channels: 320
- FICON Express8S
- Max OSA Ports: 96
- OSA-Express5S (GA2)
- Crypto Express4S
- Coupling: PSIFB: 12x DDR, 1x DDR
- ASHRAE Class A1
- Native PCIe: zEDC, Flash Express 10 GbE RoCE

N

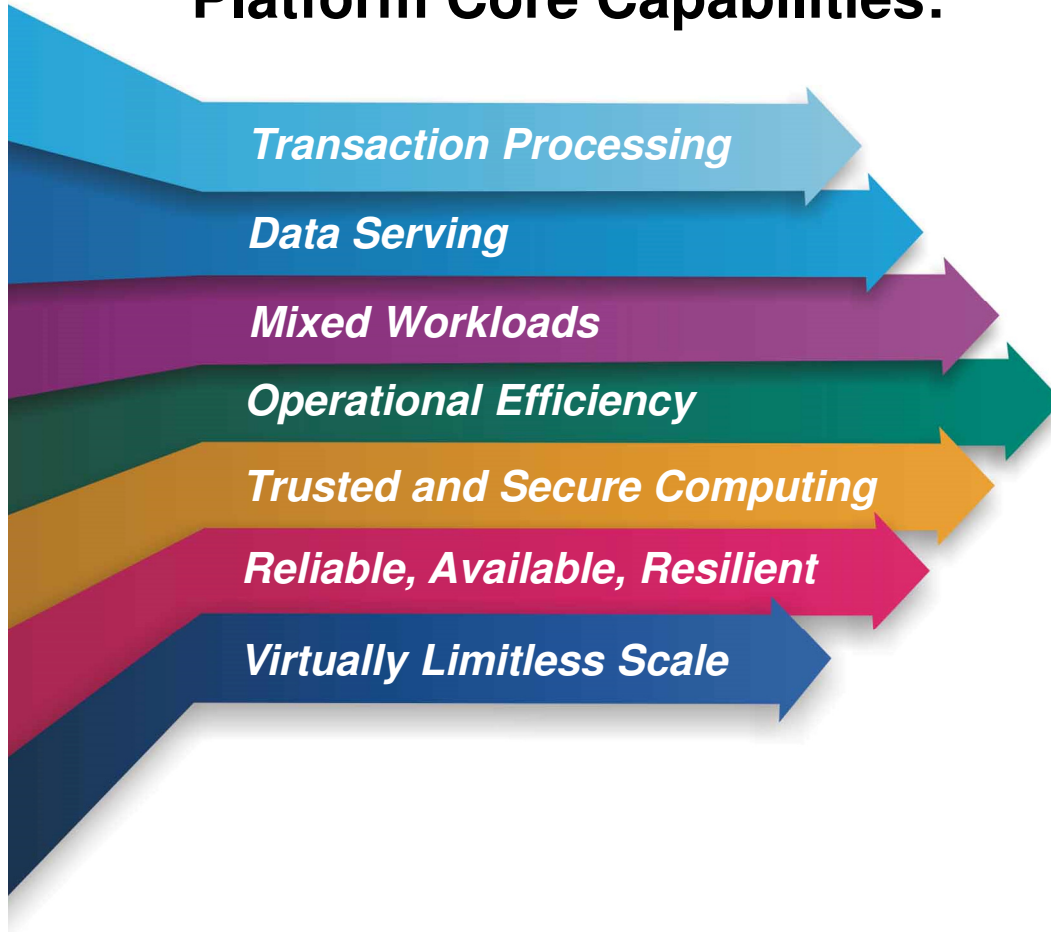


### IBM z13

- Announced 1Q2015
- Chip: 8 core, 5.0 GHz
- Up to 141 client cores
- CP, IFL, ICF, zIIP
- SMT: zIIP, IFL
- zIIP to CP ratio 2x1
- Uni MIPS: 1,695
- Max MIPS: 111,556
- Max mem: 10 TB (RAIM)
- Max per LPAR: 10 TB
- LCSS: 6, LPARs: 85
- Subchannel Sets: 4/LCSS
- Max I/O Slots: 160\*
- Max FICON Channels: 320
- FICON Express16S
- Max OSA Ports: 96
- OSA-Express5S
- Crypto Express5S
- Coupling: PSIFB: 12x DDR, 1x DDR
- ASHRAE Class A2
- PCIe: Gen3 16 Gbps
- Native PCIe: zEDC, Flash Express 10GbE RoCE with SR-IOV

## IBM z13 platform positioning

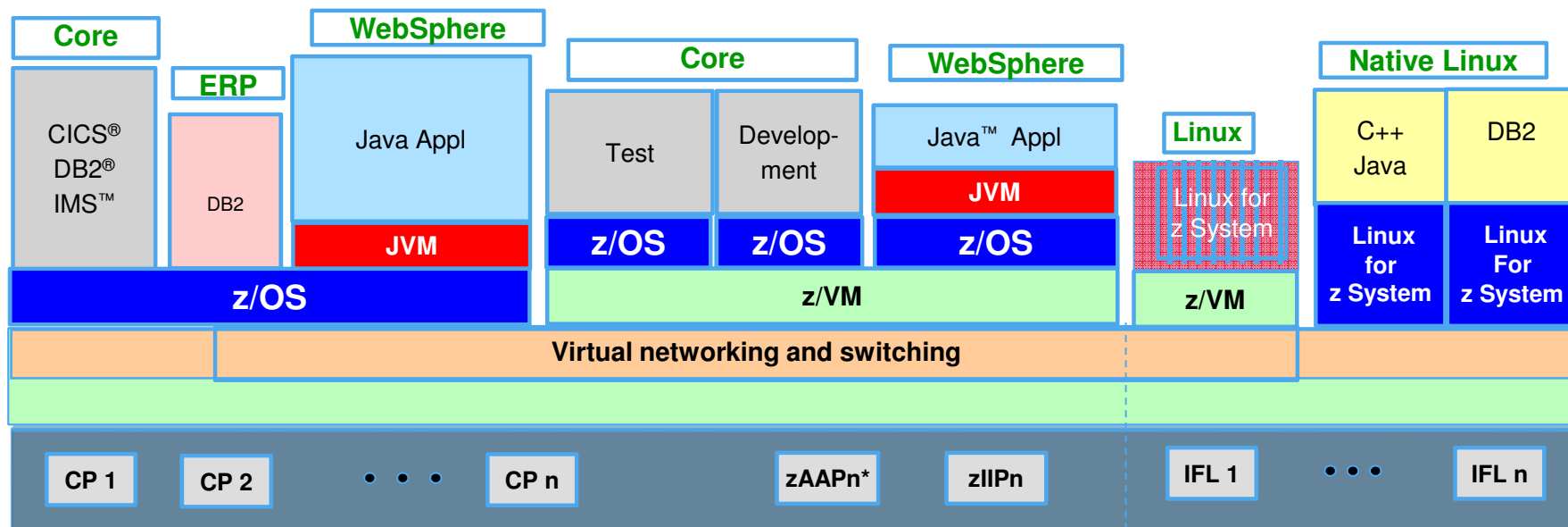
### Platform Core Capabilities:



- *The world's premier transaction and data engine now enabled for the **mobile** generation*
- *The integrated transaction and **analytics** system for right-time insights at the point of impact*
- *The world's most efficient and trusted **cloud** system that transforms the economics of IT*

# IBM z Systems

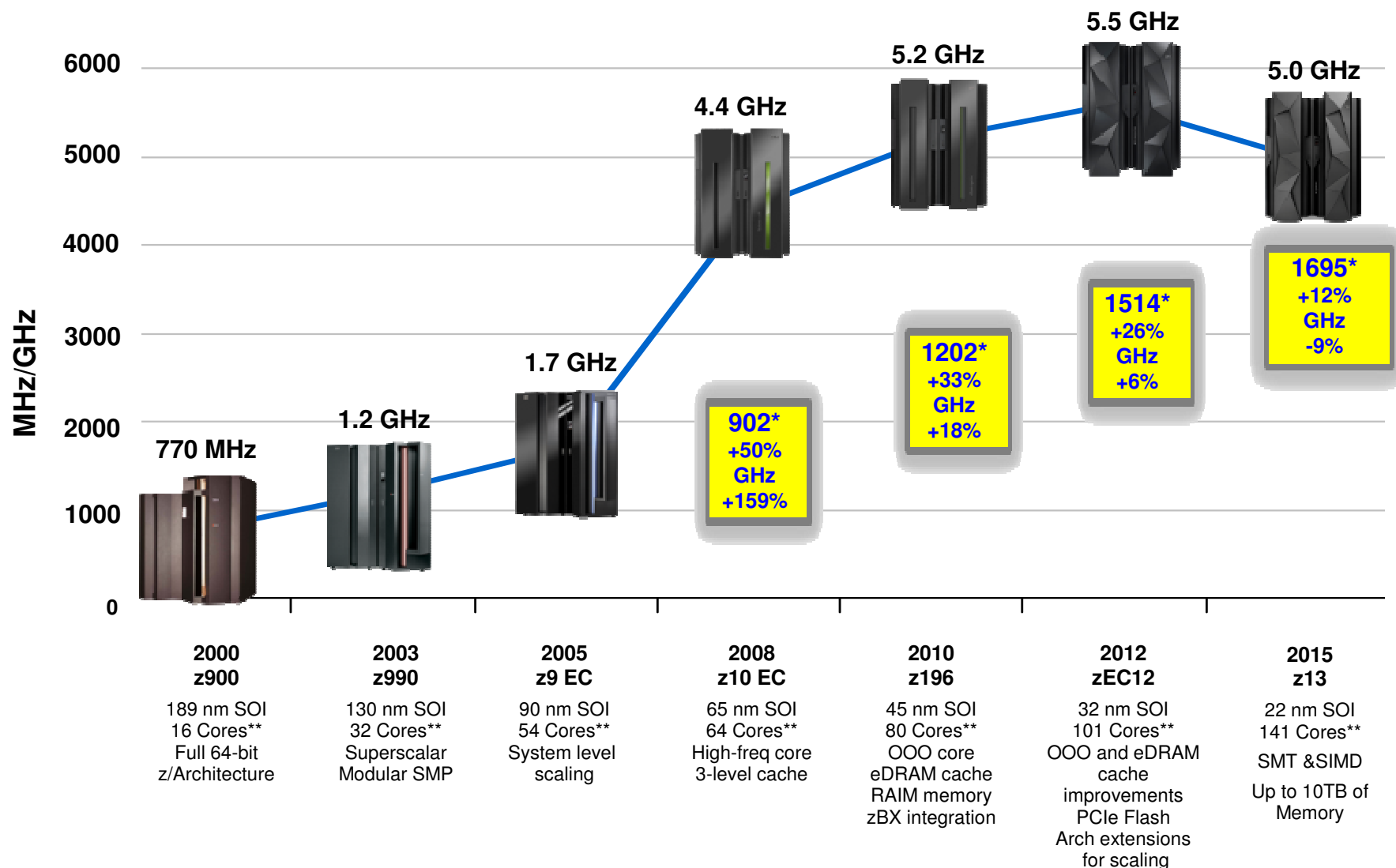
An integrated, highly scalable computer system that allows many different pieces of work to be handled at the same time, sharing the same information as needed with protection, handling very large amounts of information for many users with security, without users experiencing any failures in service



- Large scale, robust consolidation platform
- Built-in Virtualization
- 100's to 1000's of virtual servers on z/VM
- Intelligent and autonomic management of diverse workloads and system resources

\*zAAPs not available on z13

## z13 Continues the CMOS Mainframe Heritage Begun in 1994

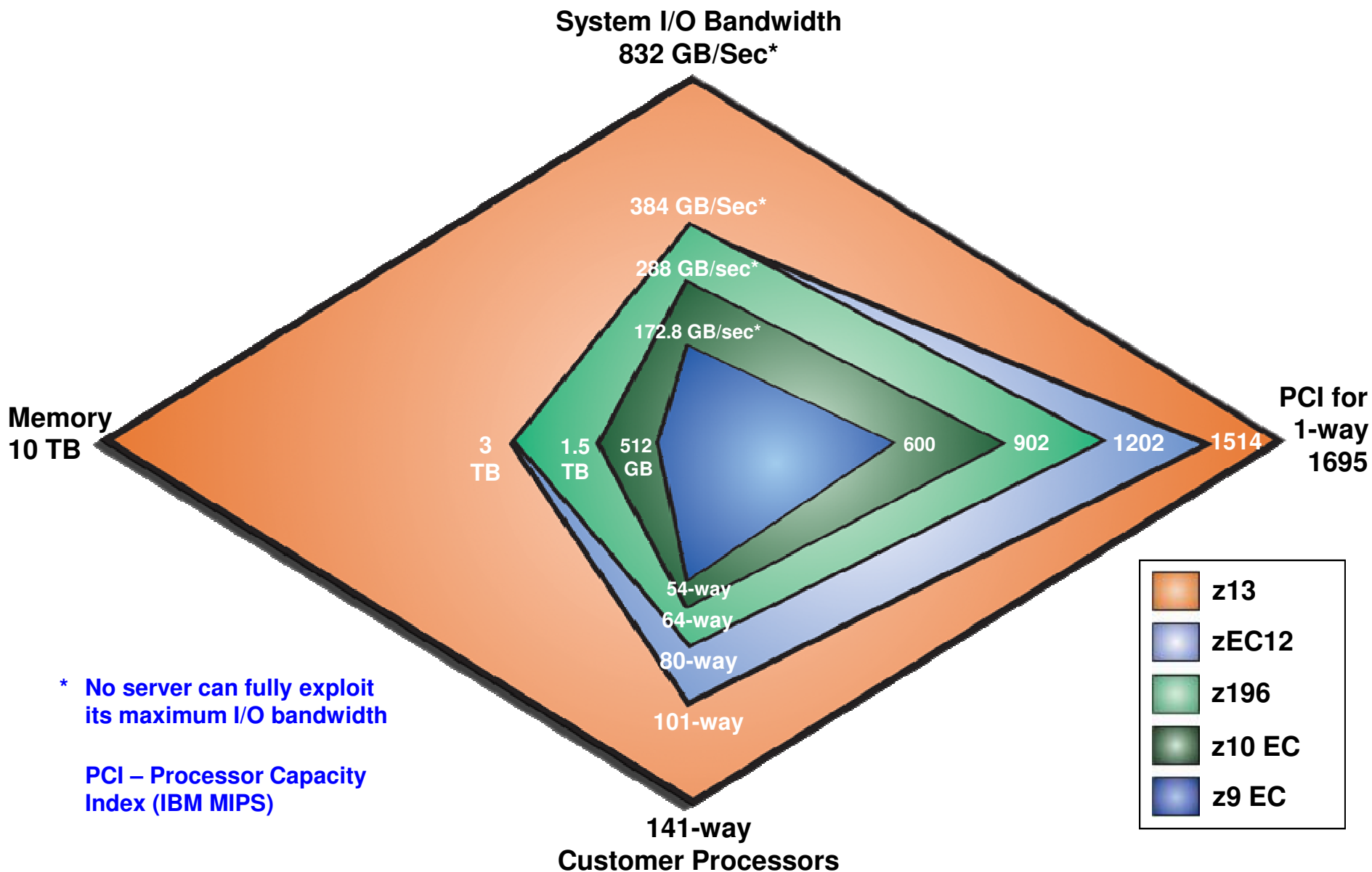


\* MIPS Tables are NOT adequate for making comparisons of z Systems processors. Additional capacity planning required

\*\* Number of PU cores for customer use



# IBM z13: Advanced system design optimized for digital business



\* No server can fully exploit its maximum I/O bandwidth

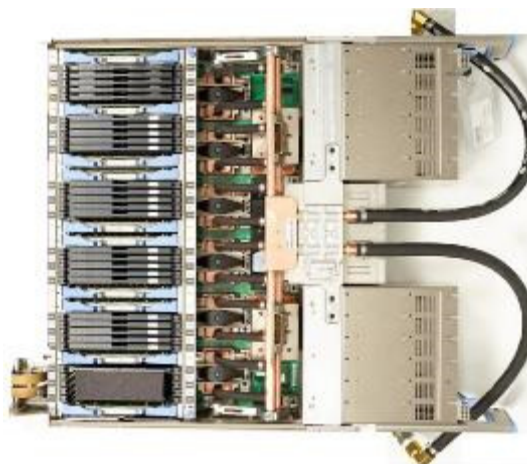
PCI – Processor Capacity Index (IBM MIPS)

## z13 System Design Changes

- 22nm Processor with SIMD, SMT
- Integrated I/O with PCIe Direct Attach – 16 GBPS
- Single Chip Modules
- Drawer-Based CPC Design
- Cable-Based SMP Fabric
- Oscillator Backplane
- Flexible Service Processor (FSP2)
- Integrated Sparing
- On-chip power/thermal monitor / control



- New Memory Controller
- Crypto Express5S
- FICON Express16S
- 1U Support Element
- Standalone zBX Node Hybrid Computing
- 2.7M lines of firmware changed
- Radiator Design improvements
- Expanded operating environment (Rear Doors)



# IBM z13: The New Possible

Mobile	Analytics	Cloud	Security
<p>Deliver <b>up to 36%</b> better response time, <b>up to 61%</b> better throughput, and <b>up to 17%</b> lower cost per mobile transaction</p>	<p>Deliver insights <b>up to 17X</b> faster and with <b>13X</b> better price performance than closest competitor</p>	<p>Enable superior Cloud services at <b>up to 32%*</b> lower cost than x86 Cloud and <b>up to 60%*</b> less than Public Cloud over three years</p>	<p>Accelerate speed of encryption <b>up to 2X</b> over the zEC12 to help protect the privacy of data throughout its life cycle</p>

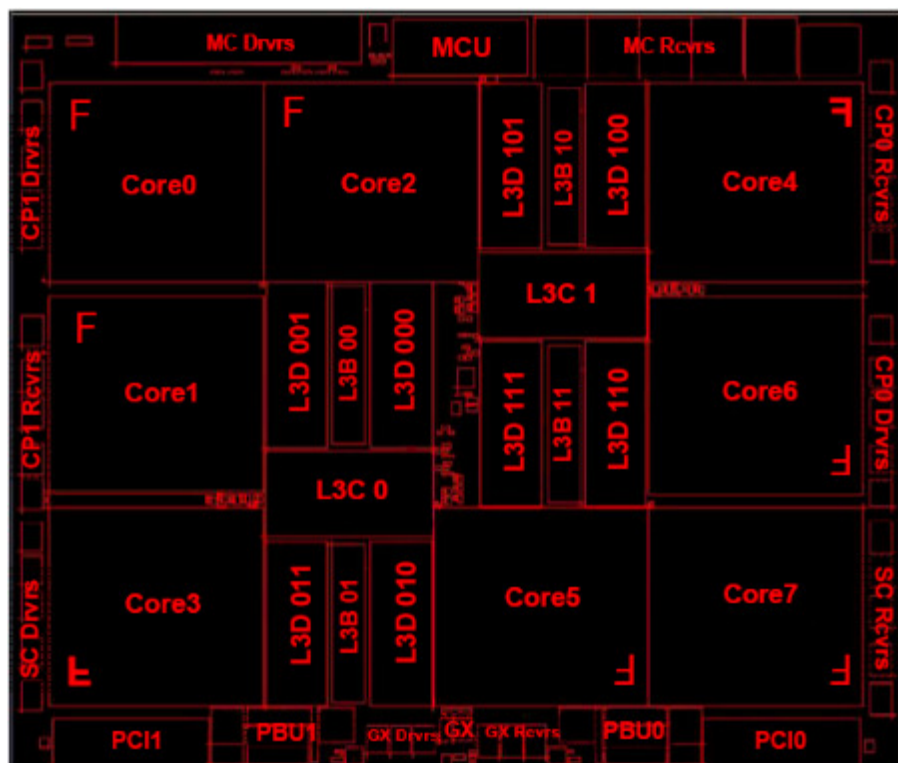
## z13 Details

## z13 z/Architecture / Micro-architecture Enhancements

- **Core micro-architecture radically altered to increase parallelism and to improve instruction execution.**
- **Simultaneous multithreading (SMT) operation**
- **Single Instruction Multiple Data (SIMD) instruction set and execution: Business Analytics Vector Processing**

**Single Thread Performance Equation= Code length \* Clock cycles per Instruction \* Cycle Time**

## z13 8-Core Processor Unit (PU) Chip Detail

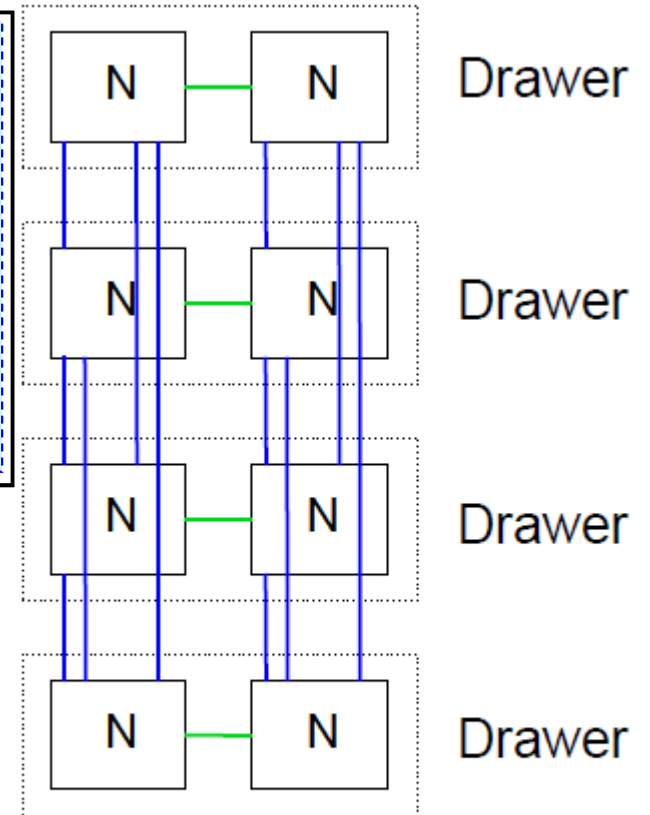
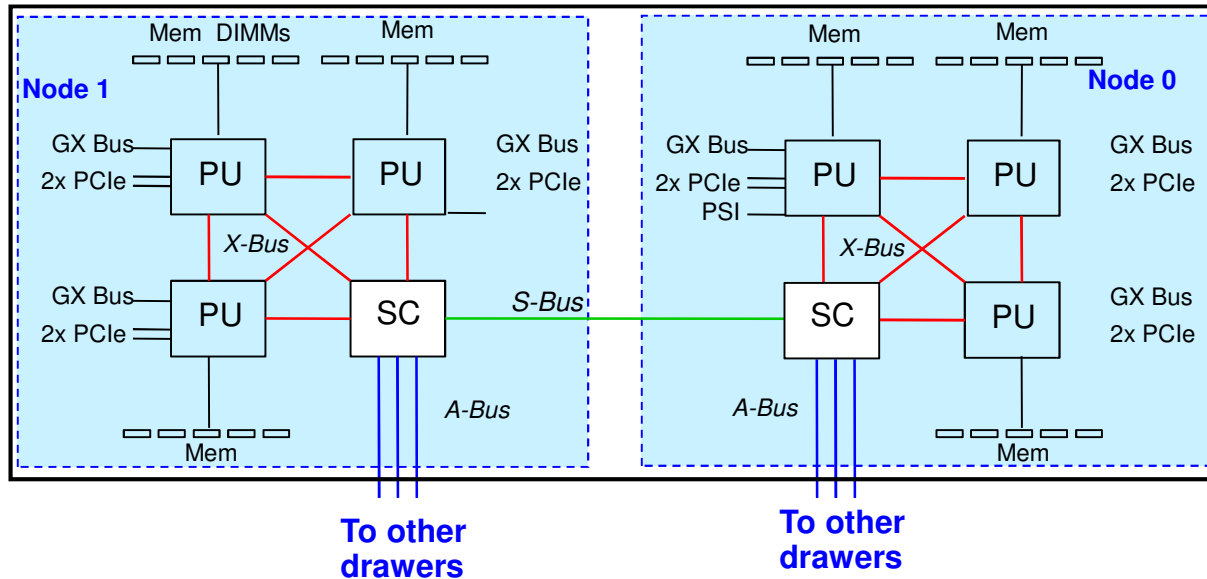


- **14S0 22nm SOI Technology**
  - 17 layers of metal
  - 3.99 Billion Transistors
  - 13.7 miles of copper wire
- **Chip Area**
  - 678.8 mm<sup>2</sup>
  - 28.4 x 23.9 mm
  - 17,773 power pins
  - 1,603 signal I/Os

- **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 multithreading (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 InfiniBand I/O bus
  - Two PCIe I/O buses
- **Memory Controller (MCU)**
  - Interface to controller on memory DIMMs
  - Supports RAIM design

# z13 Drawer Structure and Interconnect

Fully Populated Drawer

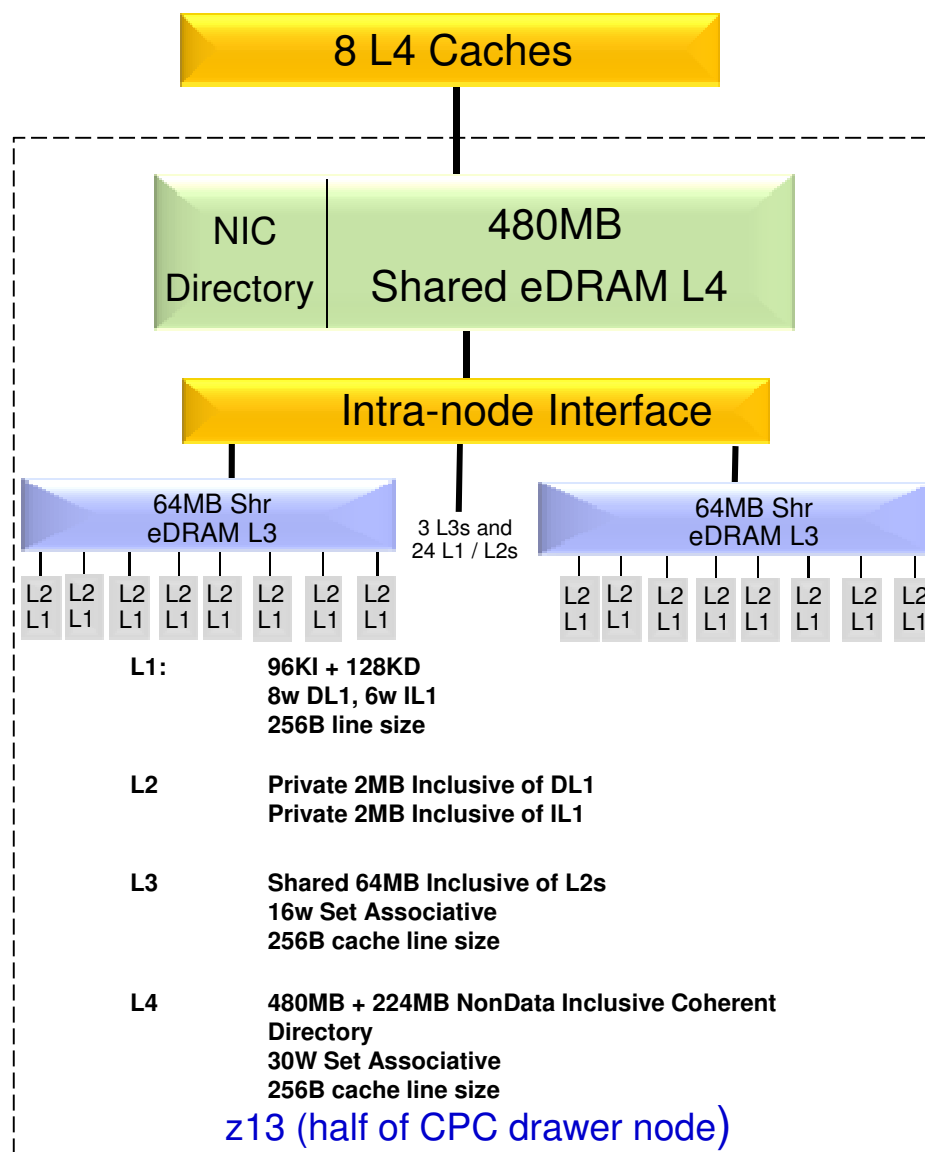
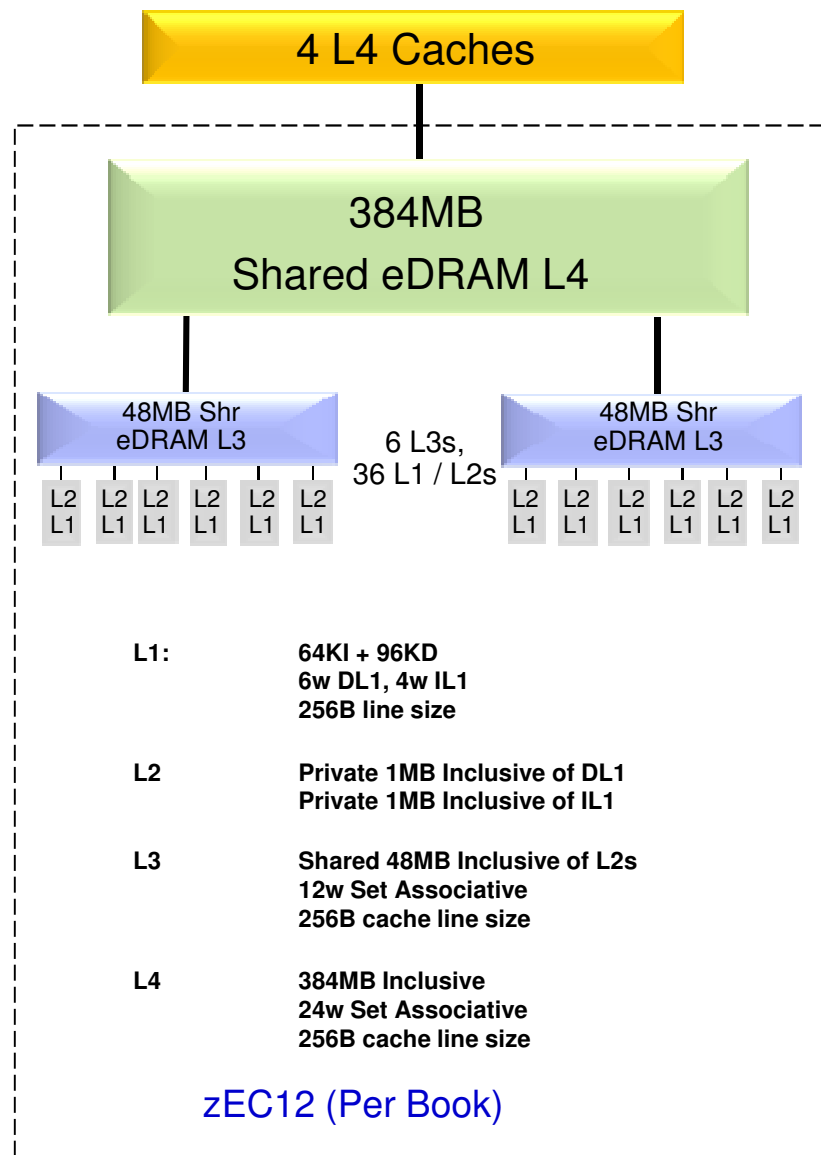


4 Drawer System Interconnect

**Physical node: (Two per drawer)**

- **Chips**
  - Three PU chips
  - One SC chip (480 MB L4 cache)
- **RAIM Memory (Redundant Array of Independent Memory)**
  - Three Memory Controllers: One per CP 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**

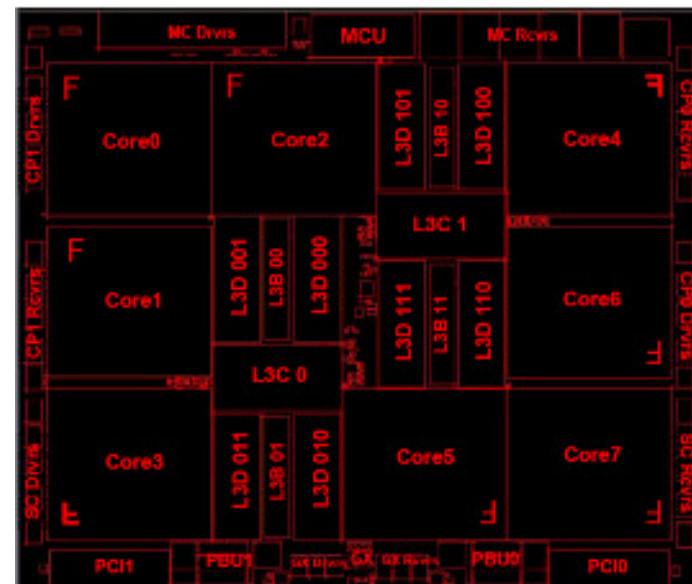
# z System Cache Topology – zEC12 vs. z13 Comparison



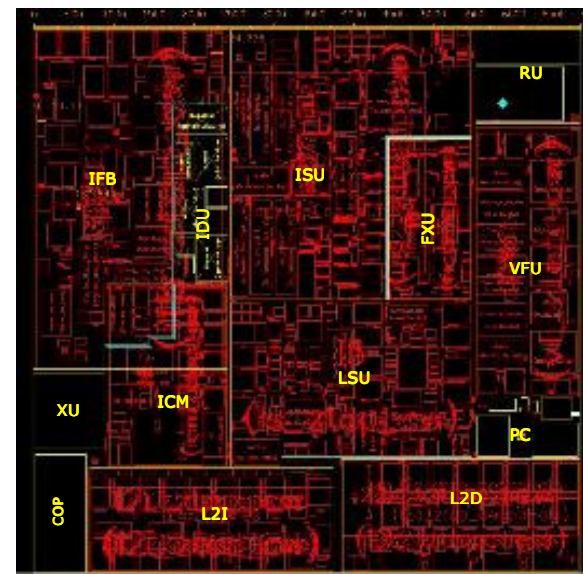


## z13 Processor Overview

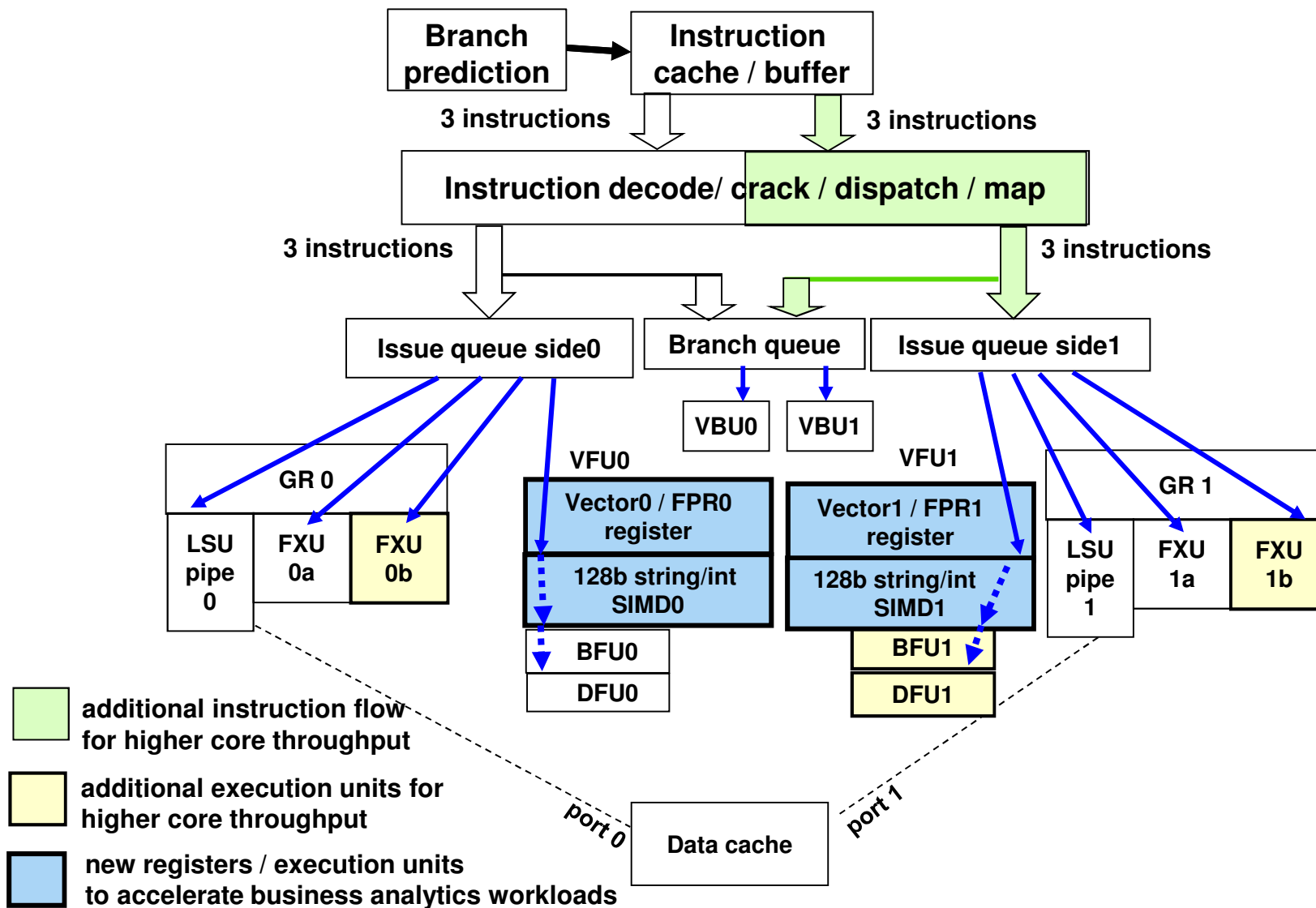
- **2X Instruction pipe width**
  - Improves IPC for all modes
  - Symmetry simplifies dispatch/issue rules
  - Required for effective SMT
- **Added FXU and BFU execution units**
  - 4 FXUs
  - 2 BFUs, DFUs
  - 2 new SIMD units
- **SIMD unit plus additional registers**
- **Pipe depth re-optimized for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



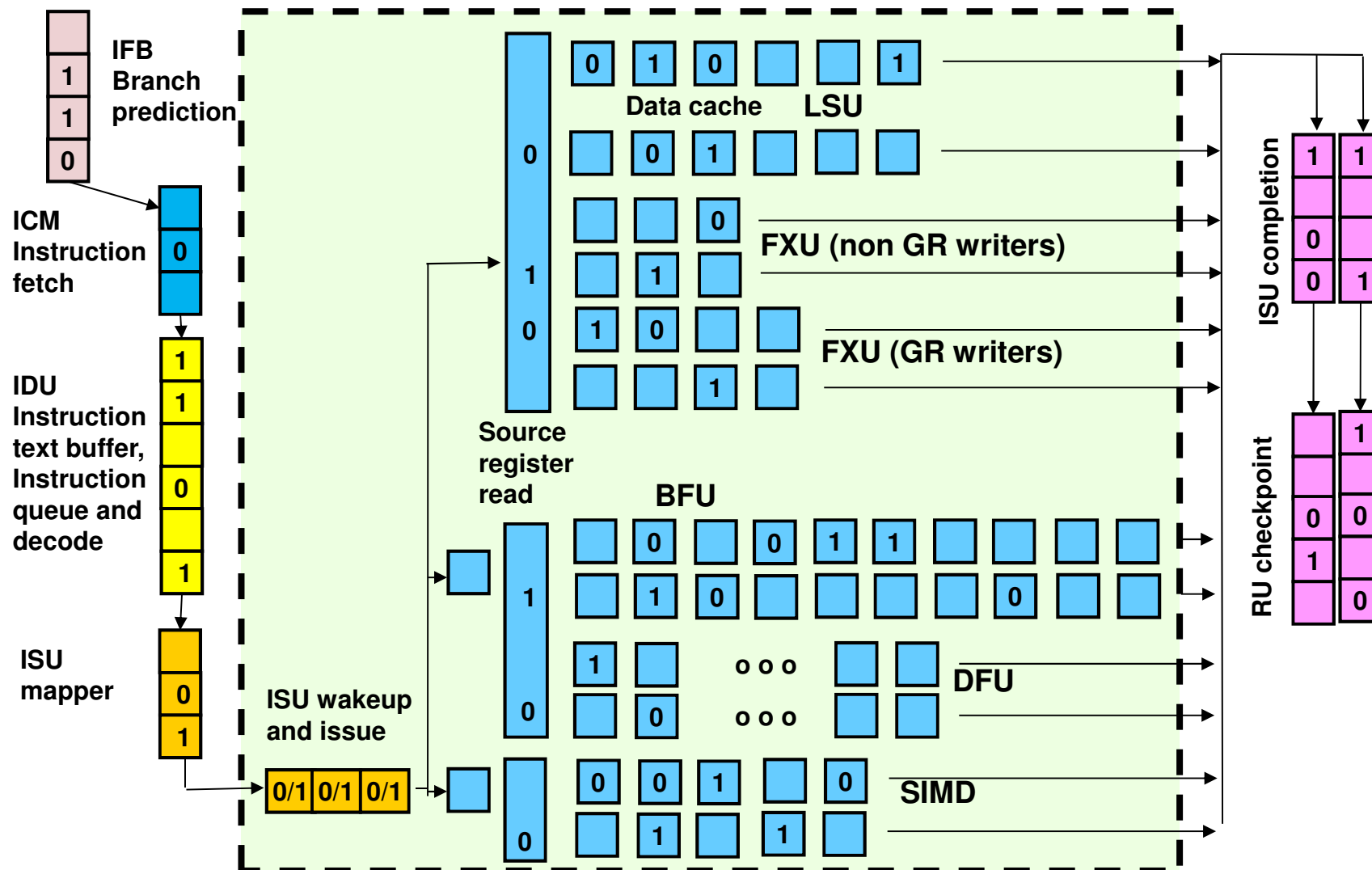
PU Chip Floorplan



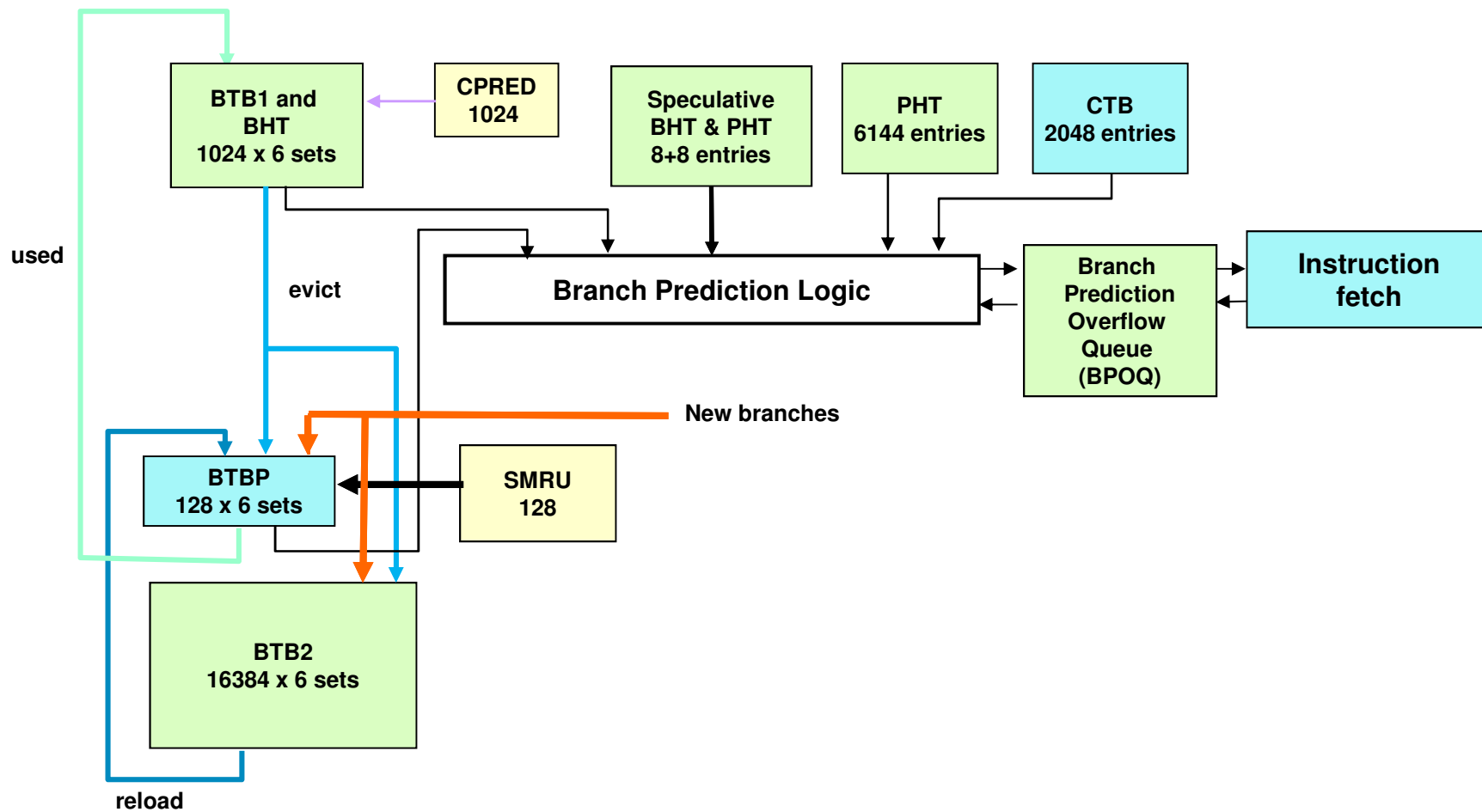
PU Core Floorplan



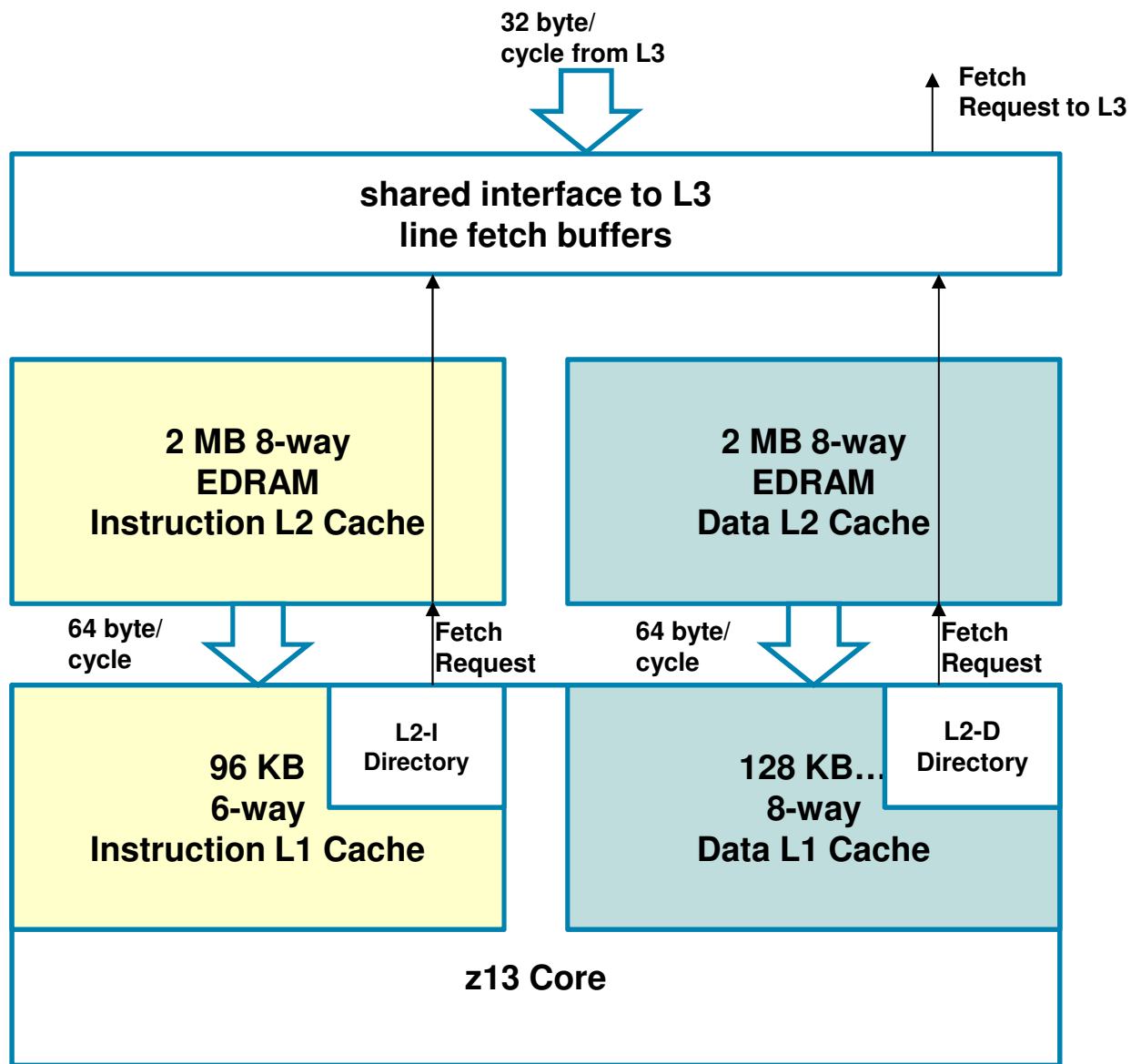
The z13 high-level instruction and execution flow.



**The z13 Microprocessor pipeline and SMT operation.**  
 Snapshot showing simultaneous execution of instructions from thread 0 and thread 1 in pipeline stages.



BTB2 increased from 4k 6-way to 16k 6-way.



Private L1 and L2 caches connected to shared L3 cache.

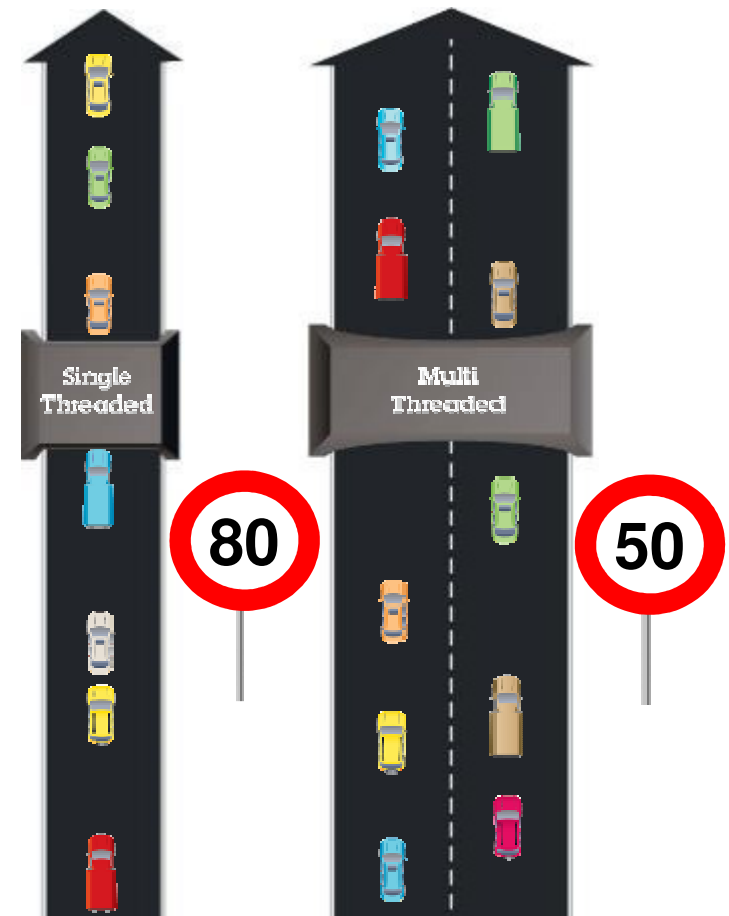
# Simultaneous Multithreading (SMT)

## 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
    - The OS(es) must actually issue instructions to switch into SMT mode
  - The SMT switch is uni-directional.
    - 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).

## Simultaneous Multithreading (SMT)

- Simultaneous multithreading allows instructions from one or two threads to execute on a Integrated Facility for Linux (IFL) or the IBM z Integrated Information Processor (zIIP) processor core.
- SMT helps to address memory latency, resulting in an overall capacity\* (throughput) improvement per core
- SMT can be turned on or off on an LPAR by LPAR basis by operating system parameters. z/OS can also do this dynamically with operator commands.



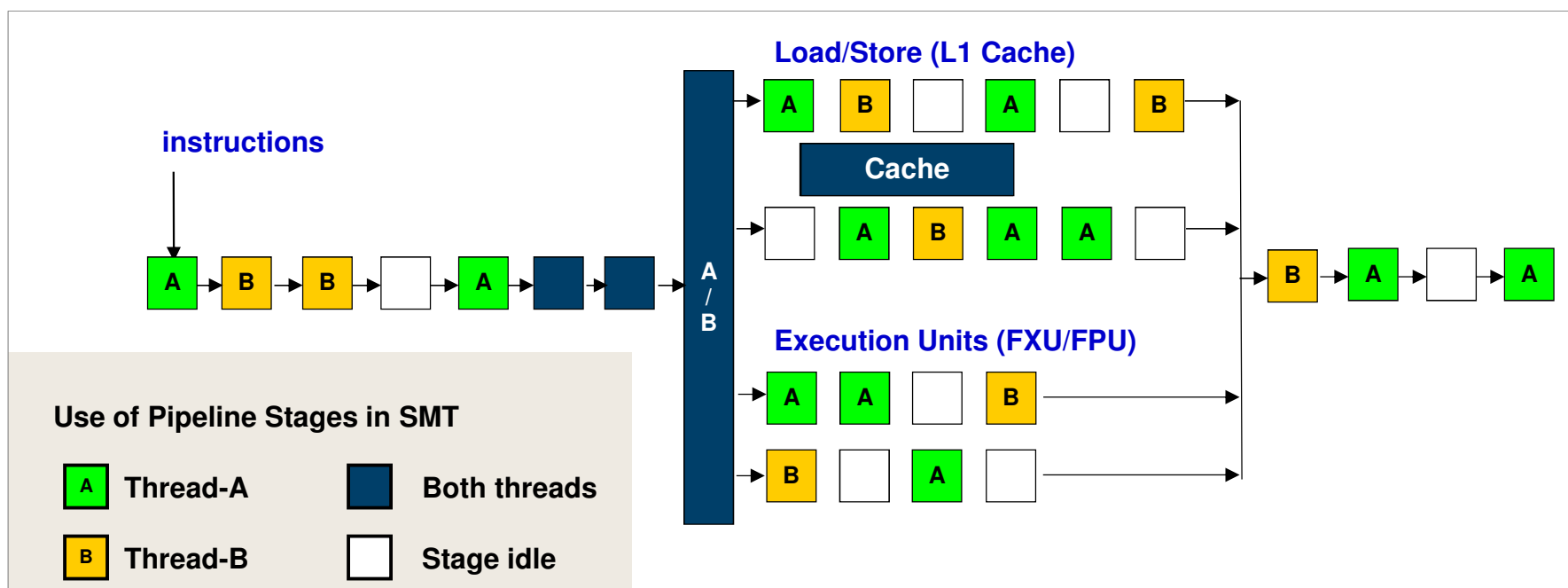
*Which approach is designed for the highest volume\*\* of traffic?  
Which road is faster?*

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



## Simultaneous Multithreading – The Technology

- **Simultaneous Multithreading (SMT) technology**
  - Multiple programs (software threads) run on the same processor core
  - More efficient use of the core hardware
- **Active threads share core resources**
  - In space: data and instruction caches, TLBs, branch history tables, etc.
  - In time: pipeline slots, execution units, address translator, etc.
- **Increases overall throughput per core when SMT is active**
  - Amount that increase, varies widely with workload – typically  $1.X-1.Y > 1$
  - Each thread runs more slowly than on a single-thread core



## z13 - Simultaneous Multithreading (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 Support Implementation

### ▪ CPU address expansion

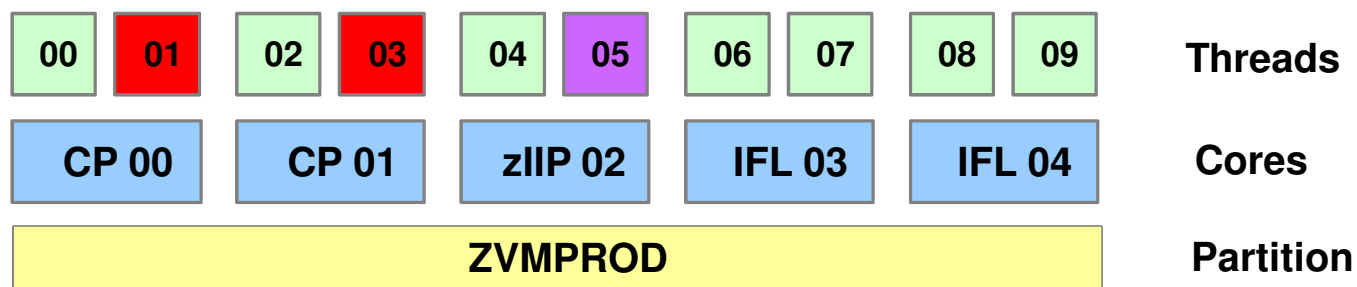
– Without SMT

- 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



## z13 Core Virtualization

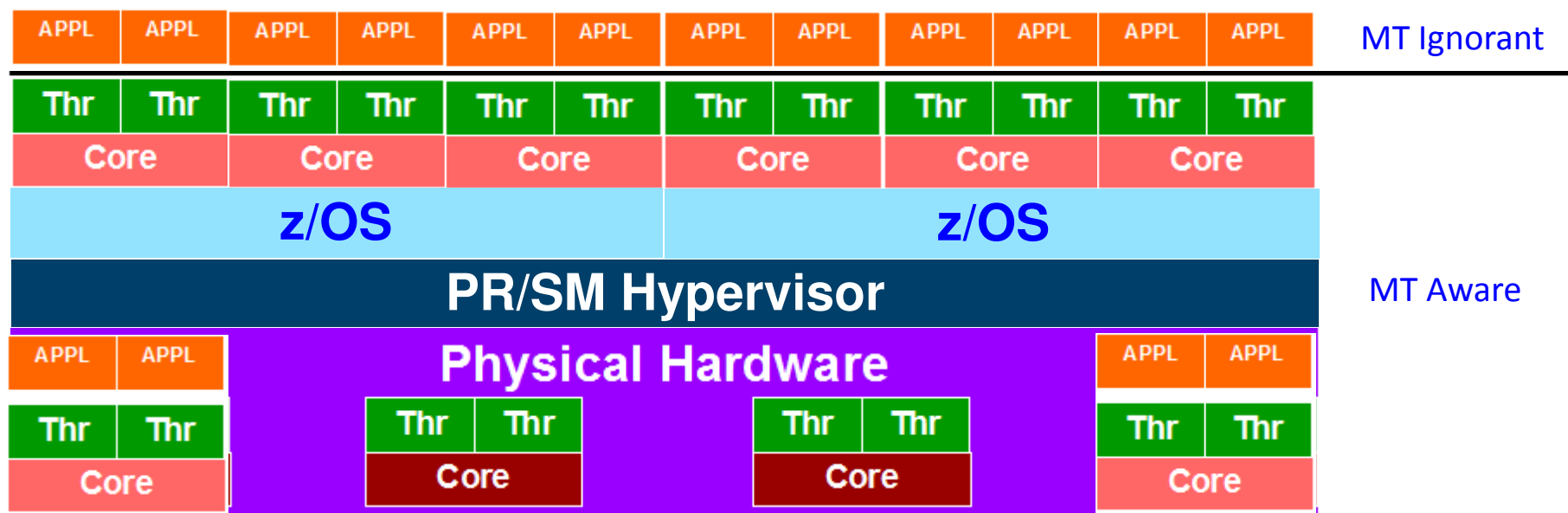
- **CPU Address changes with SMT**

- Sixteen bit CPU Id consists of a fifteen bit Core ID and one bit Thread ID



- CPU ID 6 (b'0000000000000110') means core 3 Thread 0
  - CPU ID 7 (b'0000000000000111') means core 3 Thread 1
- **On z13, z/OS will support SMT for zIIPs and z/VM will support SMT 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**
    - Hardware makes both threads usable on each core

## z System SMT Exploitation

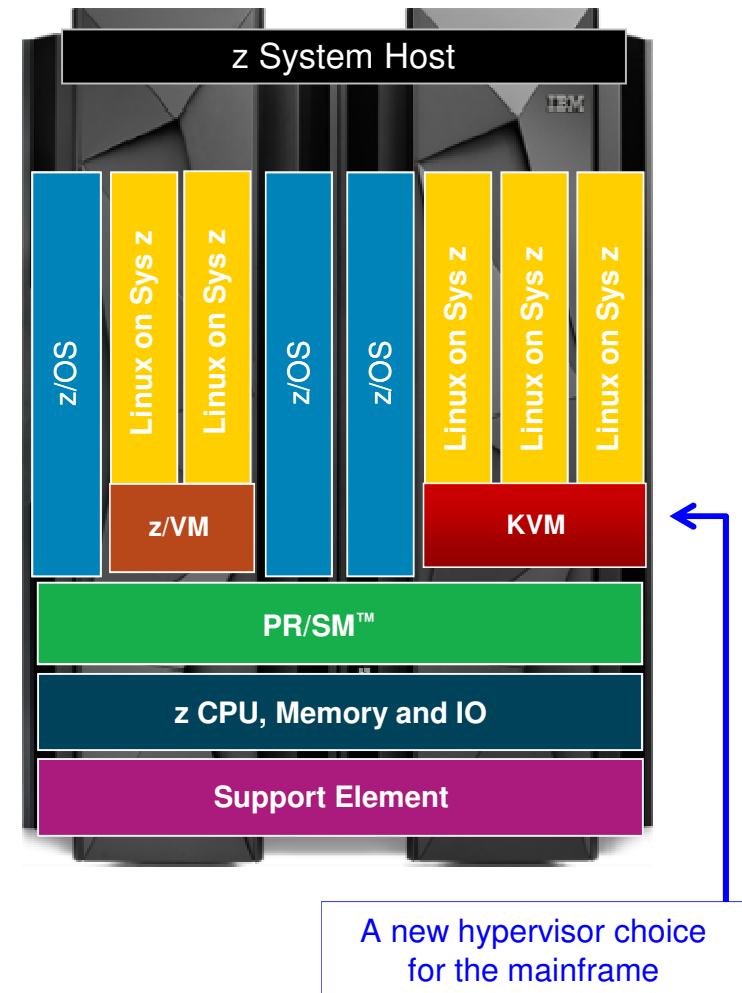


- **SMT Aware OS informs PR/SM that it intends to exploit SMT**
  - PR/SM can dispatch any OS core to any physical core
  - OS control the whole core – must follow rules
    - Maximize core throughput (Drive cores with high Thread Density [2] )
    - Maximize core availability (Meet workload goals using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

## Standardized virtualization for z System

*SOD at announcement for KVM optimized for z System*

- **Expanded audience** for Linux on z Systems
  - KVM on z System will co-exist with z/VM
  - Attracting new clients with in house KVM skills
  - Simplified startup with standard KVM interfaces
- Support of modernized **open source** KVM hypervisor for Linux
  - Provisioning, mobility, memory over-commit
  - Standard management and operational controls
  - Simplicity and familiarity for Intel Linux users
- **Optimized for z System** scalability, performance, security and resiliency
  - Standard software distribution from IBM
- Flexible **integration to cloud** offerings
  - Standard use of storage and networking drivers (including SCSI disk)
  - No proprietary agent management
  - Off-the-shelf OpenStack and cloud drivers
  - Standard enterprise monitoring and automation (i.e. GDPS)



A new hypervisor choice for the mainframe

All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice. Any reliance on these Statements of General Direction is at the relying party's sole risk and will not create liability or obligation for IBM.

# Single Instruction Multiple Data (SIMD)


<https://share.confex.com/share/124/webprogram/Session16897.html>  
IBM z Systems z13 Vector Extension Facility (SIMD)

# SIMD (Single Instruction Multiple Data) processing



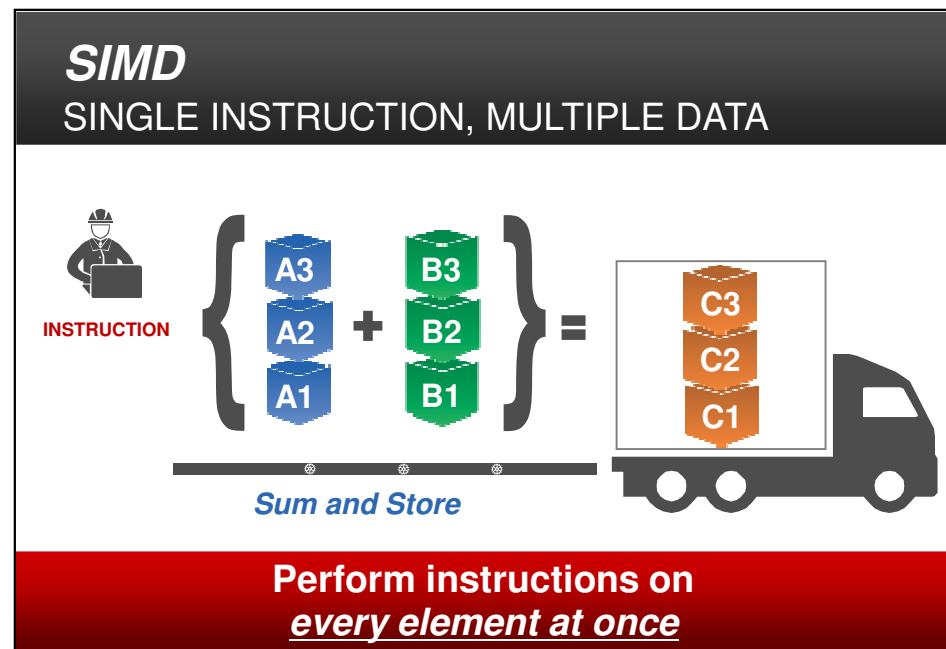
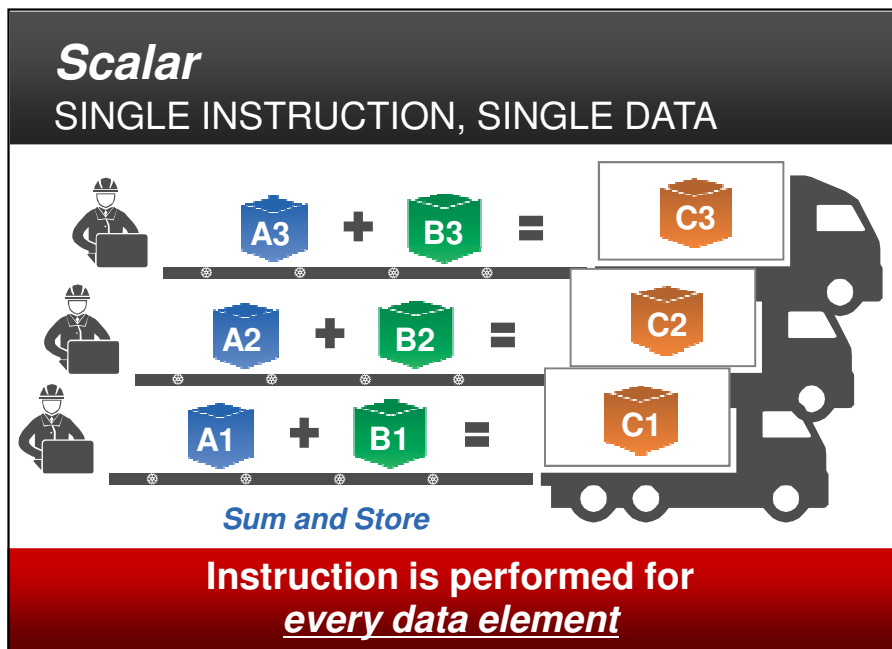
## Increased parallelism to enable analytics processing

- 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





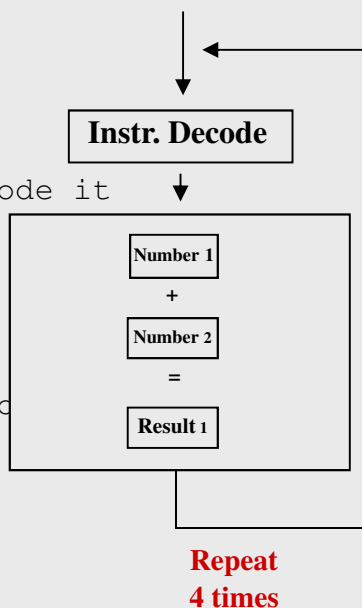
## SIMD (Single Instruction Multiple Data) Processing Example

### 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 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 there

```

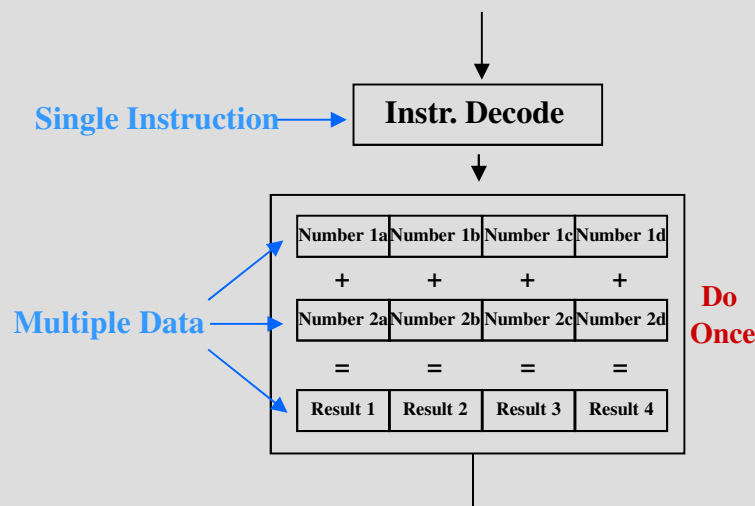


### SIMD code

```

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

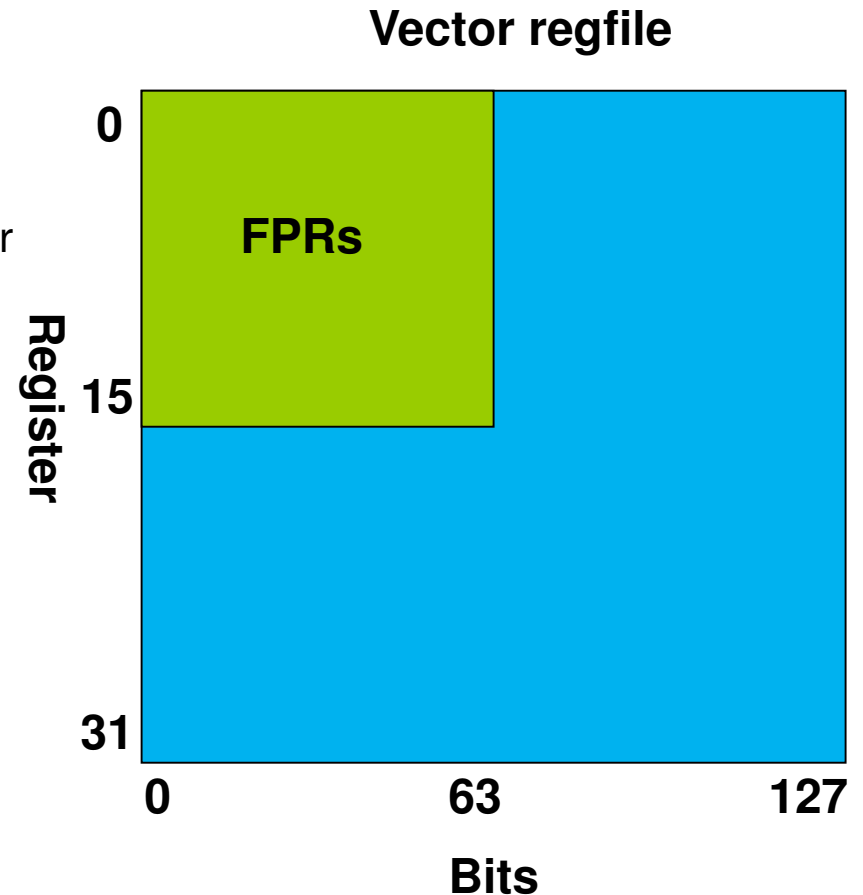
```



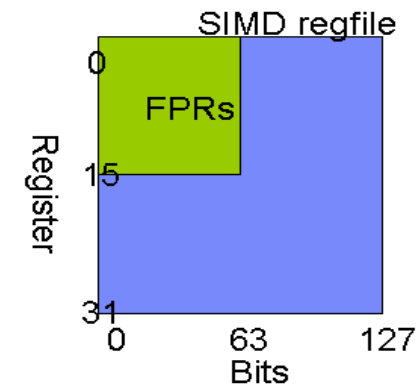
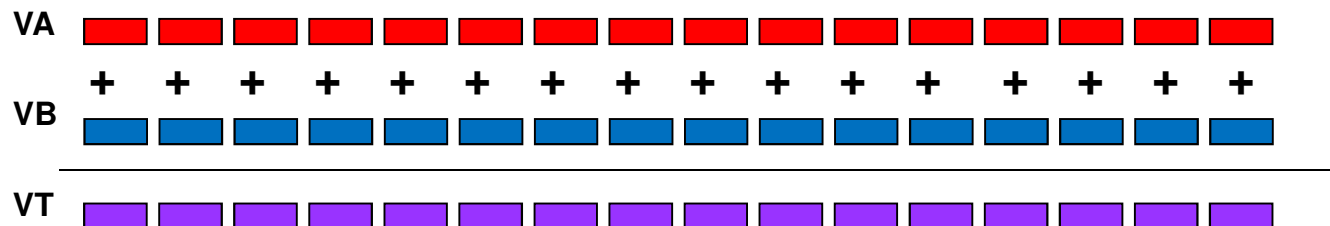
- **(Significantly) smaller amount of code => improved execution efficiency**
- **Number of elements processed in parallel = (size of SIMD / size of element)**

## Overlaid Vector/FPR register files

- **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



# z System SIMD Hardware Accelerator

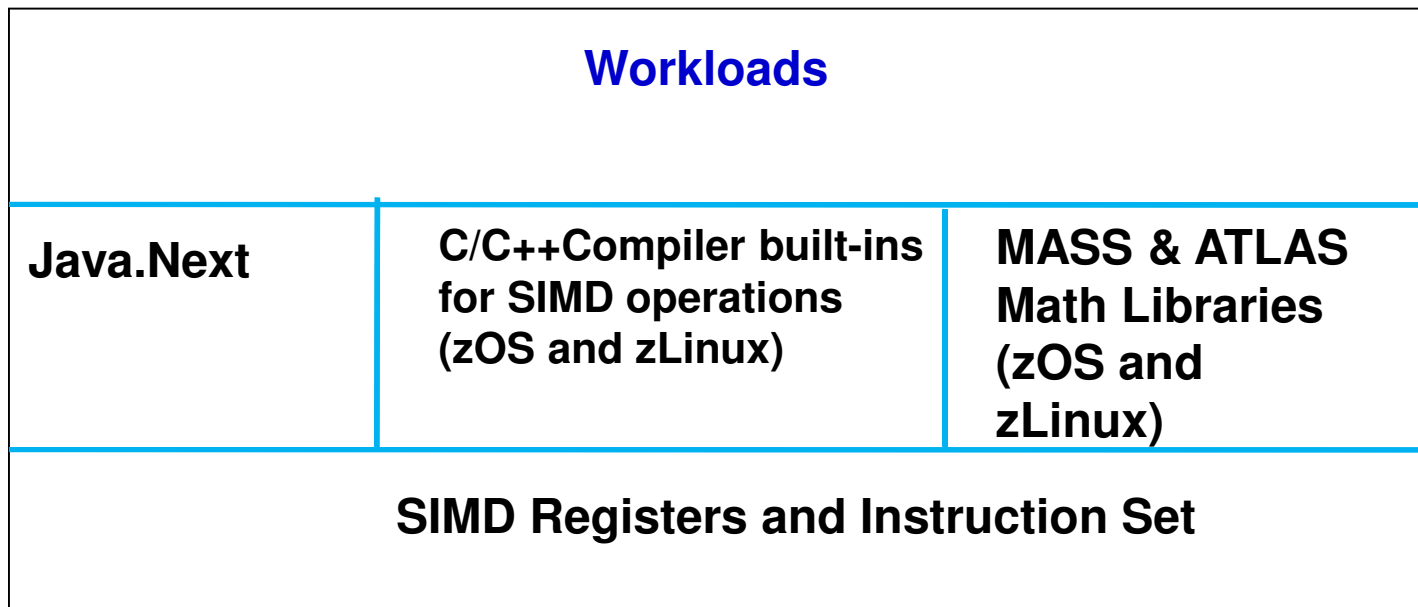


Operates on three distinct data types:

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

- 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)
- String Millicode Instructions (Translate, Compare Logical String, Compare Until Substring Equal)
- Java.Next
  - Accelerate string, converter, array operations etc
  - Idiomatic auto-vectorization (eg. simple loops)



## SIMD Instructions

- SUPPORT – Loads, Stores, Moves
- INTEGER ARITHMETIC
- FLOATING-POINT ARITHMETIC
- STRING ACCELERATION

## Vector Load Instructions

- VECTOR LOAD
  - VL  $VR_1, D_2(X_2, B_2)$
  - Load 16 bytes from storage into  $VR_1$ . **No alignment requirement**
- VECTOR LOAD AND REPLICATE
  - VL RP(B|H|F|G)  $VR_1, D_2(X_2, B_2), M_3$
  - Load 1-8 bytes and replicate across all elements of  $VR_1$
- VECTOR LOAD ELEMENT
  - VLE(B|H|W|D)  $VR_1, D_2(X_2, B_2), M_3$
  - The element sized second operand is placed into  $VR_1$  at index  $M_3$
- VECTOR LOAD ELEMENT IMMEDIATE
  - VLEI(B|H|F|G)  $VR_1, I_2, M_3$
  - Places  $I_2$  in  $VR_1$  at index  $M_3$ , leaves rest of vector unchanged
- VECTOR LOAD MULTIPLE
  - VLM  $VR_1, VR_3, D_2(B_2), M_4$
  - Up to 16 VRs loaded from storage
- **VECTOR LOAD TO BLOCK BOUNDARY**
  - VLBB  $VR_1, D_2(X_2, B_2), M_3$
  - Loads up to 16 bytes into  $VR_1$  without crossing block boundary specified by  $M_3$
- **LOAD COUNT TO BLOCK BOUNDARY**
  - LCBB  $R_1, D_2(X_2, B_2), M_3$
  - Loads  $R_1$  with number of bytes that can be loaded with specified block size
- **VECTOR LOAD WITH LENGTH**
  - VLL  $VR_1, D_2(B_2), R_3$
  - Loads the number of bytes specified in  $R_3$  from storage into  $VR_1$
- VECTOR LOAD LOGICAL ELEMENT AND ZERO
  - VLLEZ (B|H|F|G)  $VR_1, D_2(X_2, B_2), M_3$
  - Load element sized data from second operand address and place right justified in leftmost DW
- VECTOR GATHER ELEMENT
  - VGEF(VGEG)  $VR_1, D_2(V_2, B_2), M_3$
  - Loads element from memory addressed by  $B_2 + V_2(M_3) + D_2$

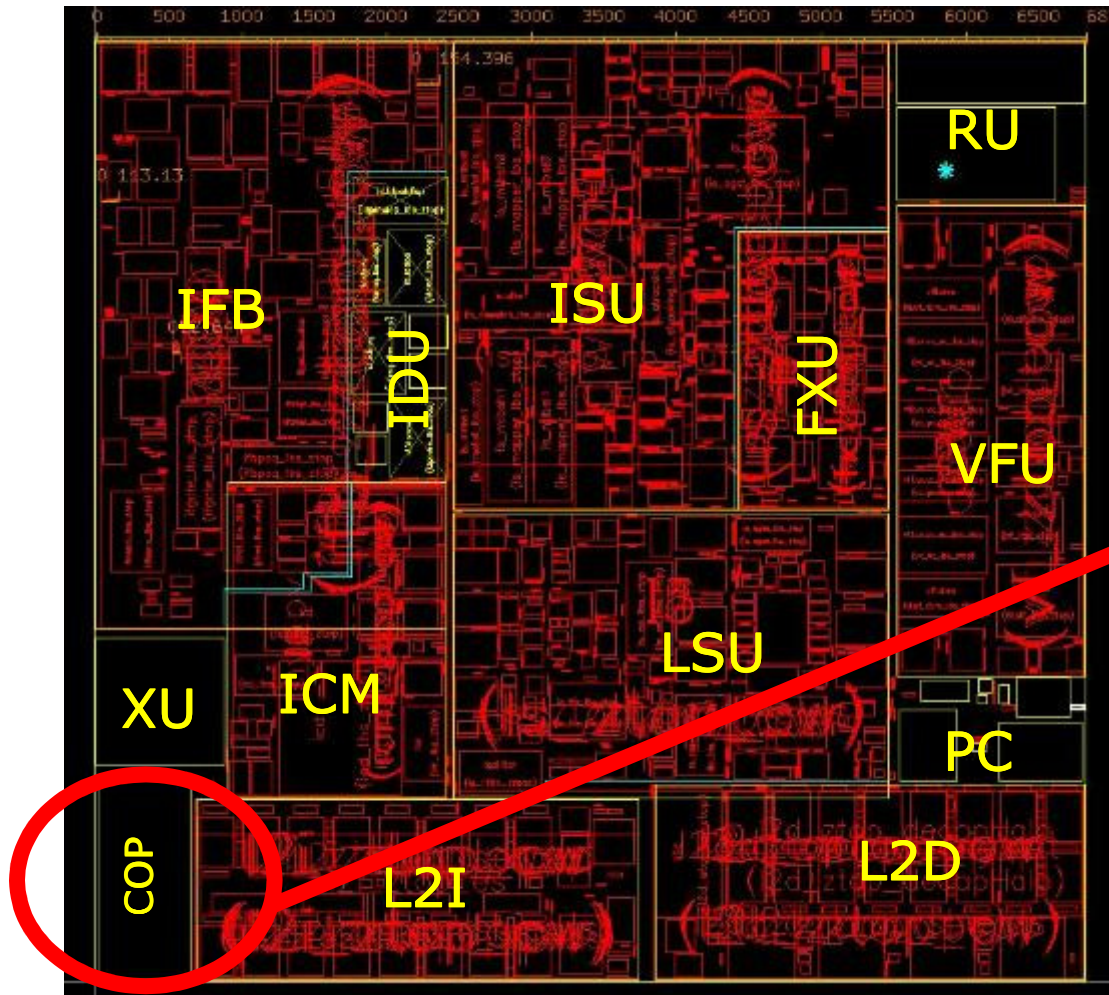
## Vector Store Instructions

- VECTOR STORE
  - VST  $VR_1, D_2(X_2, B_2)$
  - Stores 16 bytes on byte boundary, no alignment required
- VECTOR STORE ELEMENT
  - VSTE(B|H|F|G)  $VR_1, D_2(X_2, B_2), M_3$
  - Stores element of  $VR_1$  indexed by  $M_3$  to second operand
- VECTOR STORE MULTIPLE
  - VSTM  $VR_1, VR_3, D_2(B_2), M_4$
  - Stores range of up to 16 VRs to second operand location
- VECTOR STORE WITH LENGTH
  - VSTL  $VR_1, D_2(B_2), R_3$
  - Stores the number of bytes specified by  $R_3$  from  $VR_1$  into the second operand location
- VECTOR SCATTER ELEMENT
  - VSCEF(VSCEG)  $VR_1, D_2(V_2, B_2), M_3$
  - Stores element of  $VR_1$  indexed by  $M_3$  to memory addressed by  $B_2 + V_2(M_3) + D_2$

# z Systems Crypto



## Where is the Coprocessor located on the PU core?



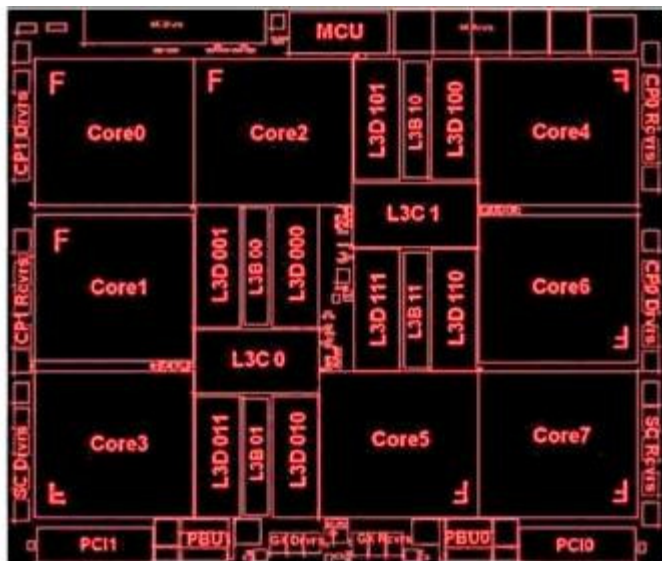
### 5 Engines

- CMPSC expansion
- CMPSC compression
- Cipher
- Hashing
- UTF

## 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 Systems
  - 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 or Crypto Express4S

# CPACF - CP Assist For Cryptographic Functions

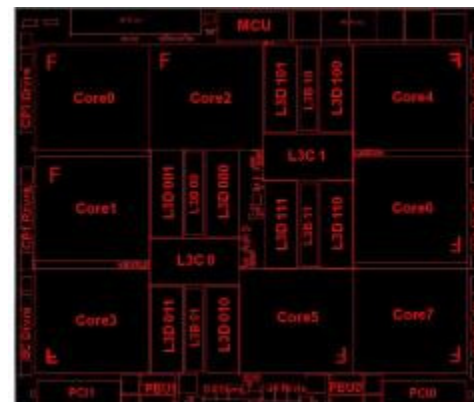


- **Provides a set of symmetric cryptographic functions and hashing functions for:**
  - Data privacy and confidentiality
  - Data integrity
  - Random Number generation
  - Message Authentication
- **Enhances the encryption/decryption performance of clear-key operations for**
  - SSL
  - VPN
  - Data storing applications
- **Available on every Processor Unit**
- **Supported by z/OS, z/VM, z/VSE, z/TPF and Linux on z Systems**
- **Must be explicitly enabled, using a no-charge enablement feature (#3863),**
  - SHA algorithms enabled with each server
- **Protected key support for additional security of cryptographic keys**
  - Crypto Express4s or Crypto Express5S required in CCA mode

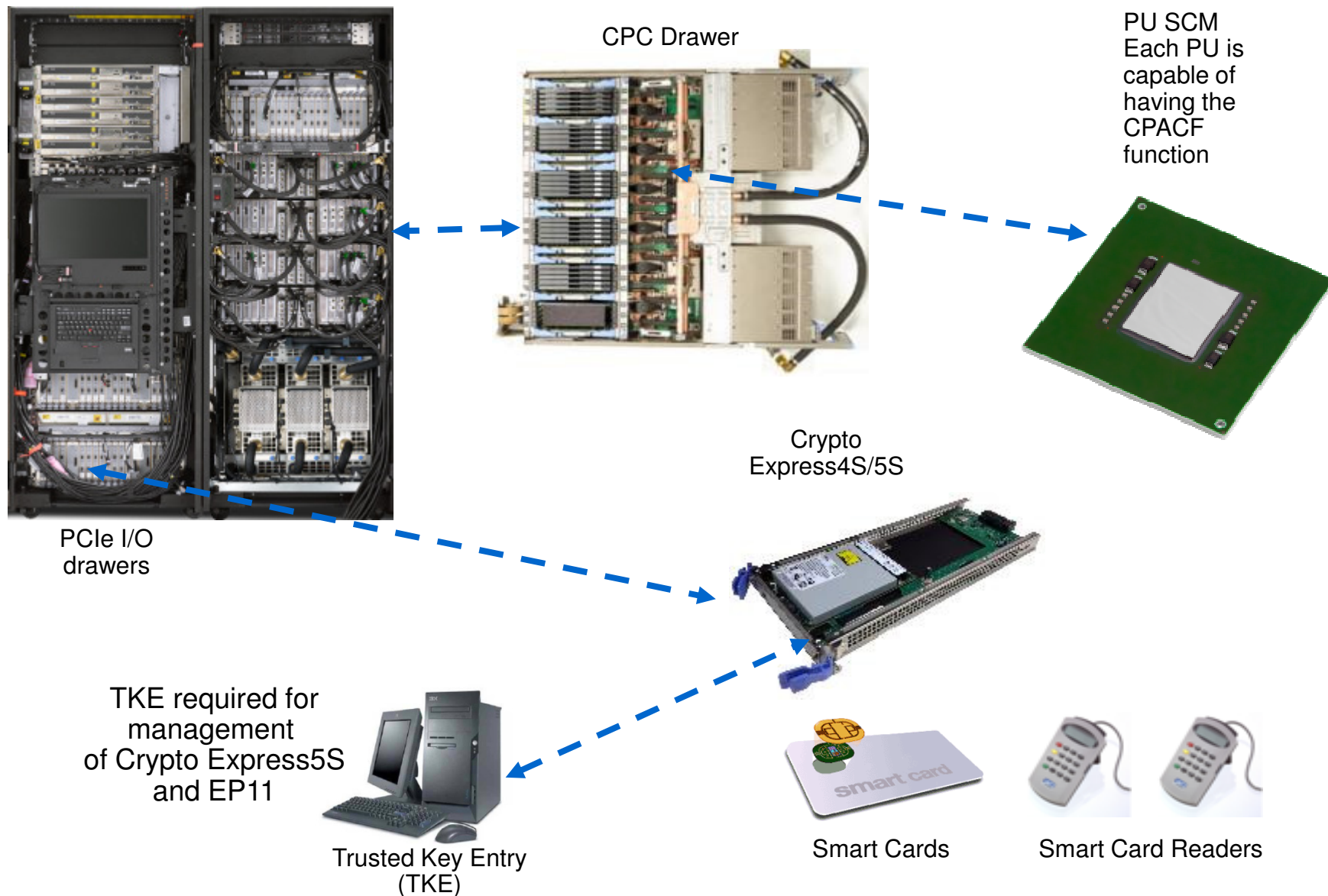
Supported Algorithms	Clear Key	Protect Key
DES, T-DES	Y	Y
AES128	Y	Y
AES192	Y	Y
AES256	Y	Y
SHA-1	Y	N/A
SHA-256	Y	N/A
SHA-384	Y	N/A
SHA-512	Y	N/A
PRNG	Y	N/A
DRNG	Y	N/A

## z13 CPACF

- **CP Assist for Cryptographic Function Co-processor redesigned from "ground up"**
- **Enhanced performance over zEC12**
  - Does not include overhead for COP start/end and cache effects
  - Enhanced performance for large blocks of data
    - AES: 2x throughput vs. zEC12
    - TDES: 2x throughput vs. zEC12
    - SHA: 3.5x throughput vs. zEC12
- **Exploiters of the CPACF benefit from exploited by the throughput improvements of z13's CPACF such as:**
  - DB2/IMS encryption tool
  - DB2® built in encryption
  - z/OS Communication Server: IPsec/IKE/AT-TLS
  - z/OS System SSL
  - z/OS Network Authentication Service (Kerberos)
  - DFDSS Volume encryption
  - z/OS Java SDK
  - z/OS Encryption Facility
  - Linux on z Systems; kernel, openssl, openCryptoki, GSKIT



# Overview – HW Crypto support in z Systems



## Crypto Express5S Standards supported

- **DES/TDES w DES/TDES MAC/CMAC**
- **AES, AESKW, AES GMAC, AES GCM, AES XTS mode, CMAC**
- **MD5, SHA-1, SHA-2 (224,256,384,512), HMAC**
- **VISA Format Preserving Encryption (VFPE)**
- **RSA (512, 1024, 2048, 4096) -> Performance improvement**
- **ECDSA (192, 224, 256, 384, 521 Prime/NIST)**
- **ECDSA (160, 192, 224, 256, 320, 384, 512 BrainPool)**
- **ECDH (192, 224, 256, 384, 521 Prime/NIST)**
- **ECDH (160, 192, 224, 256, 320, 384, 512 BrainPool)**
- **Montgomery Modular Math Engine**
- **RNG (Random Number Generator)**
- **PNG (Prime Number Generator) -> NEW**
- **Clear Key Fast Path (Symmetric and Asymmetric)**

## IBM z13 – Taking Java Performance to the Next Level

Continued aggressive investment in Java on Z

Significant set of new hardware features tailored and co-designed with Java

### Simultaneous Multi-Threading (SMT)

- 2x hardware threads/core for improved throughput
- Available on zIIPs and IFLs

### Single Instruction Multiple Data (SIMD)

- Vector processing unit
- Accelerates loops and string operations

### Cryptographic Function (CPACF)

- Improved performance of crypto co-processors

### New Instructions

New **5.0 GHz** 8-Core Processor Chip

**480Mb L4 cache** to optimize for data serving



Up to **50%**  
improvement for generic  
applications

Up to **2X** improvement in  
throughput per core for security  
enabled applications

## Accelerating using SIMD with IBM Java 8 and z13

### IBM z13 running Java 8 on z/OS Single Instruction Multiple Data (SIMD) vector engine exploitation

#### java.lang.String exploitation

- compareTo
- compareToIgnoreCase
- contains
- contentEquals
- equals
- indexOf
- lastIndexOf
- regionMatches
- toLowerCase
- toUpperCase
- getBytes

#### java.util.Arrays

- equals (primitive types)

#### String encoding converters

For ISO8859-1, ASCII, UTF8, and UTF16

- encode (char2byte)
- decode (byte2char)

#### Auto-SIMD

- Simple loops  
(eg. Matrix multiplication)

**Primitive operations are between 1.6x and 60x faster with IBM Java8**



धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Merci

French

Bedankt

Nederlands

Gracias!

Spanish

شكراً

Arabic

多谢

Simplified Chinese

Obrigado

Brazilian Portuguese

Danke

German

நன்றி

Tamil

ありがとうございました

Japanese

감사합니다