IBM POWER line

Dezső Sima

January 2019

(Ver. 2.0)

© Dezső Sima, 2019

Contents

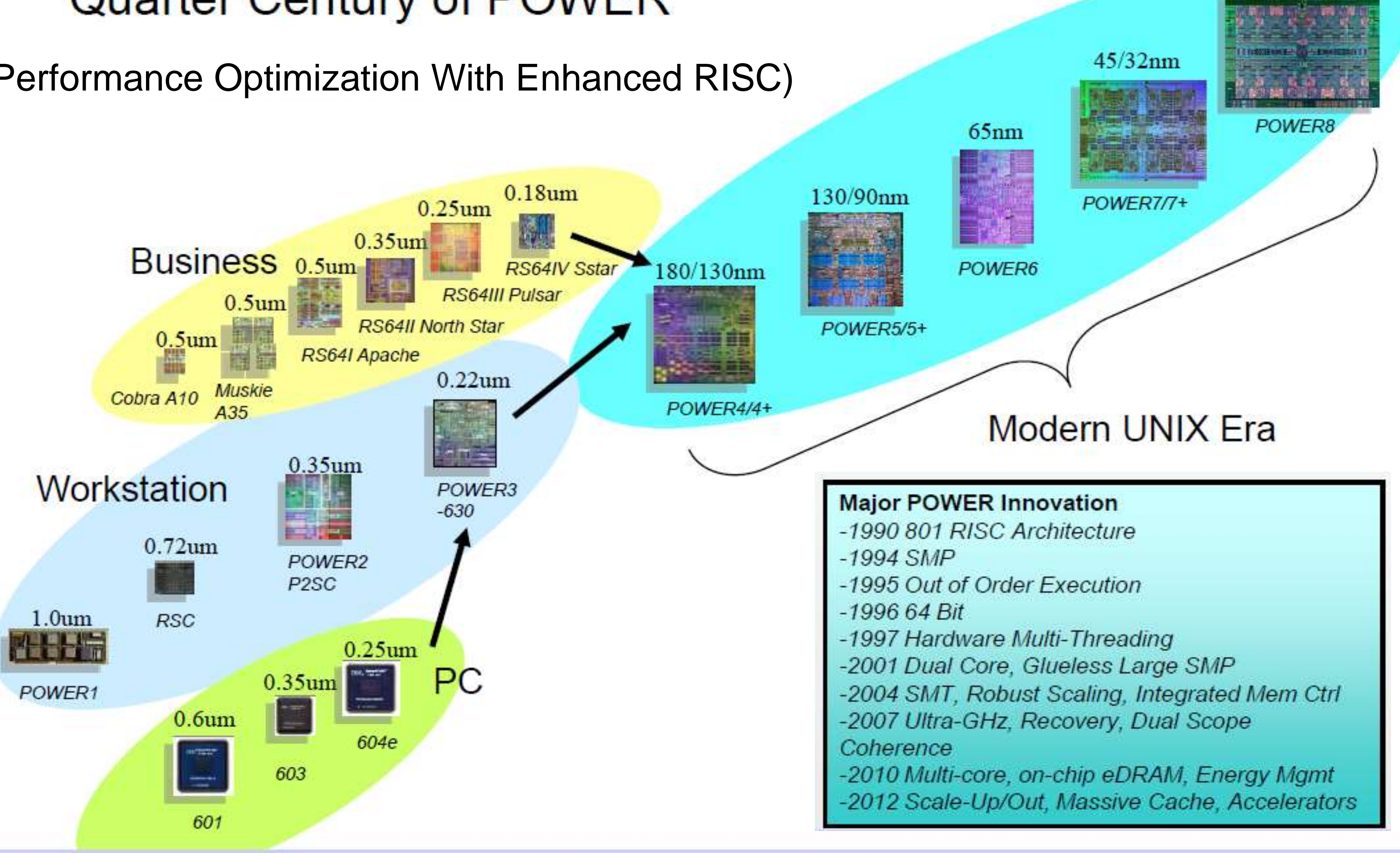
- 1. Overview of the POWER line
- 2. POWER4
- 3. POWER4+
- 4. POWER5
- 5. POWER5+
- 6. POWER6
- 7. POWER6+

Contents

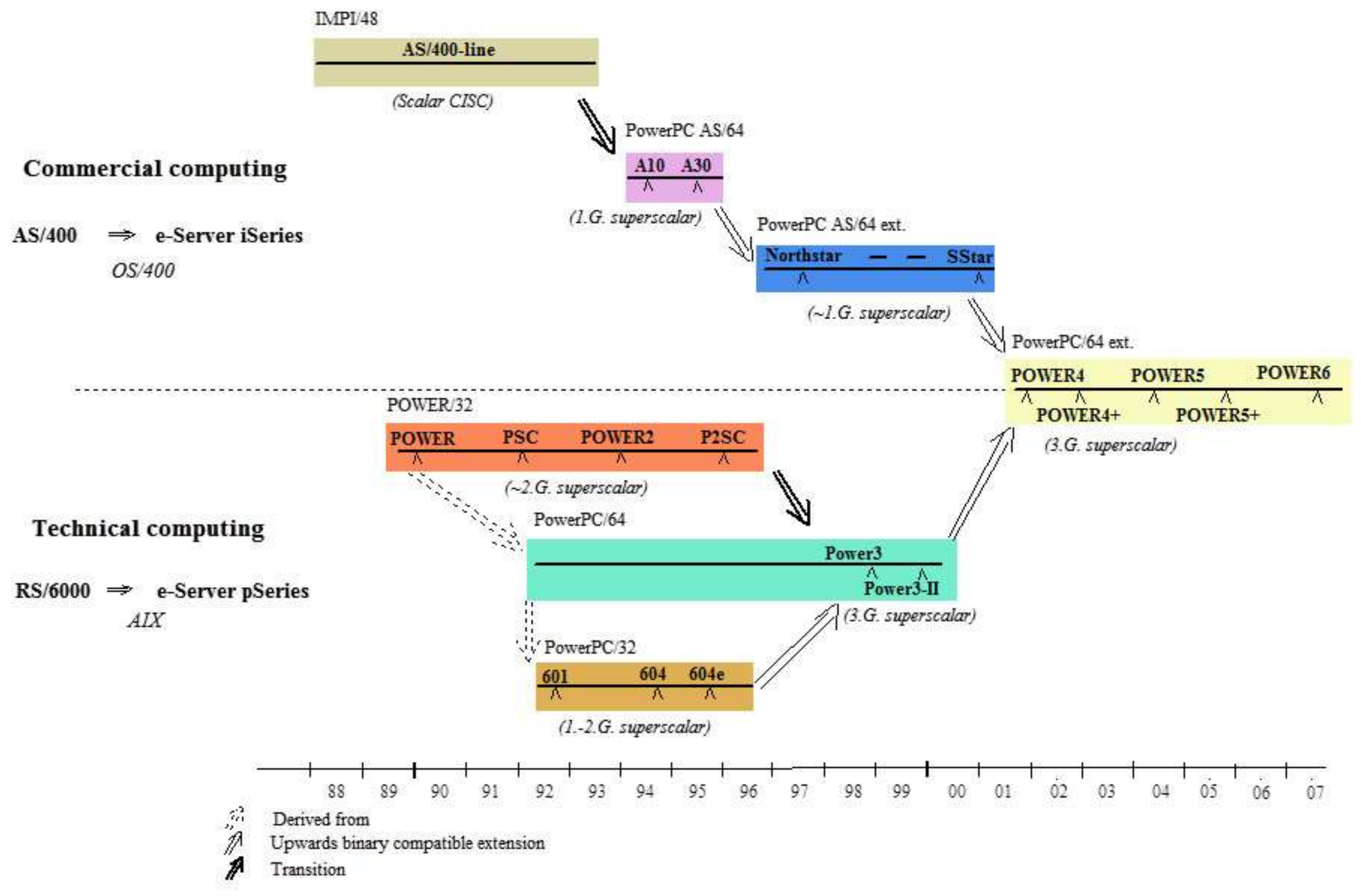
- 8. POWER7
- 9. POWER7+
- 10. POWER8
- 11. POWER8+
- 12. POWER9
- 13. References

1. Overview of the POWER line

Overview of IBM's product categories [1], [131]



The convergence of IBM's processor lines (except of the z line)

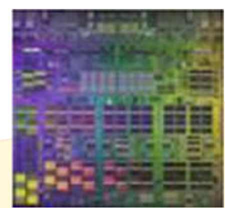
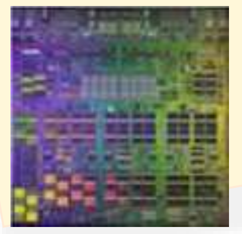


IBM's product naming around 2006 [2]

New Name	Old Names	Market	Processor
System i	iSeries, AS400	Commercial	RS64 POWER5
System p	RS6000 SP pSeries	Server, technical	POWER3 POWER4 POWER5
System x	xSeries IA-32	Server, technical	Intel AMD PowerPC
System z	zSeries ES9000	Mainframe	zSeries

1. Overview of the POWER line (4)

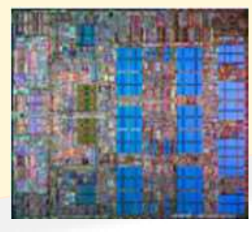
Key innovations of subsequent models of the POWER line -1 (Die photos from [3])

POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP

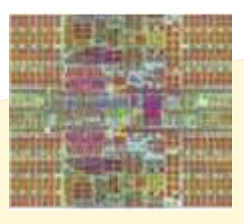
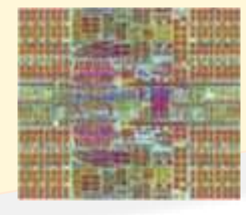
2001



POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating

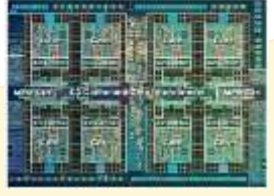
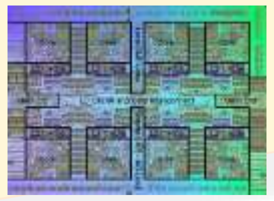
2004

POWER6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

2007

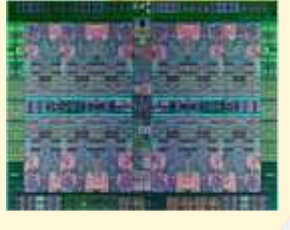



POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

*POWER7+

2010

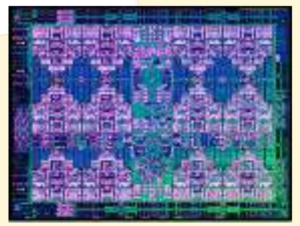


POWER8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
NVLink
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2014

Key innovations of subsequent models of the POWER line -2 (Die photos: [3])



POWER9
14 nm

24 cores
4/8-way SMT
PCIe 4.0 (aka G4)
CAPI 2.0
OpenCAPI (CAPI 3.0
NVLink 2.0

2018

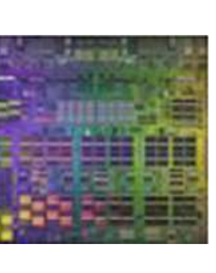
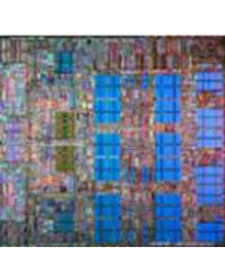
1. Overview of the POWER line (6)

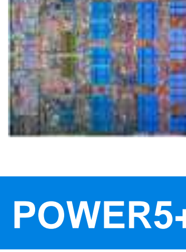
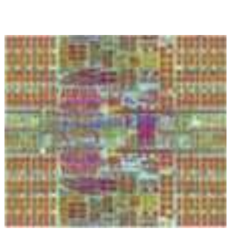
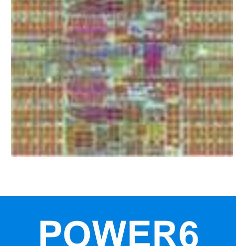
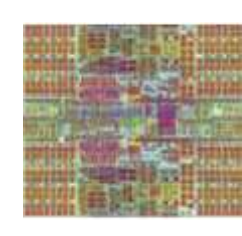
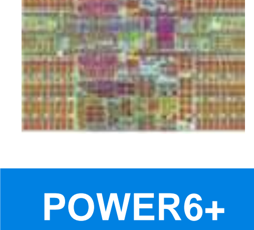
Remark to the subsequent tables

- Subsequent tables typically show maximum figures of the referenced features, e.g. of core counts or clock frequencies, in order to provide more concise comparisons.
- In fact, actual features of delivered server models usually differ from the maximum figures due to a number of reasons, e.g. due to power constraints or simply to keep prices low.

1. Overview of the POWER line (7)

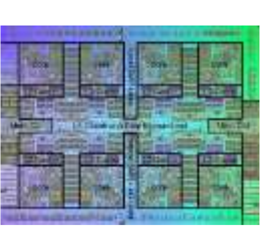
Key features of the POWER models -1

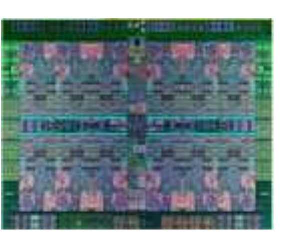



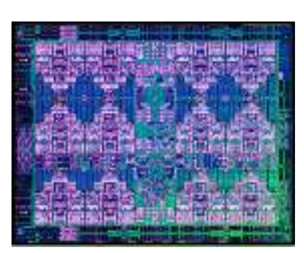






	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

Key features of the POWER models -2







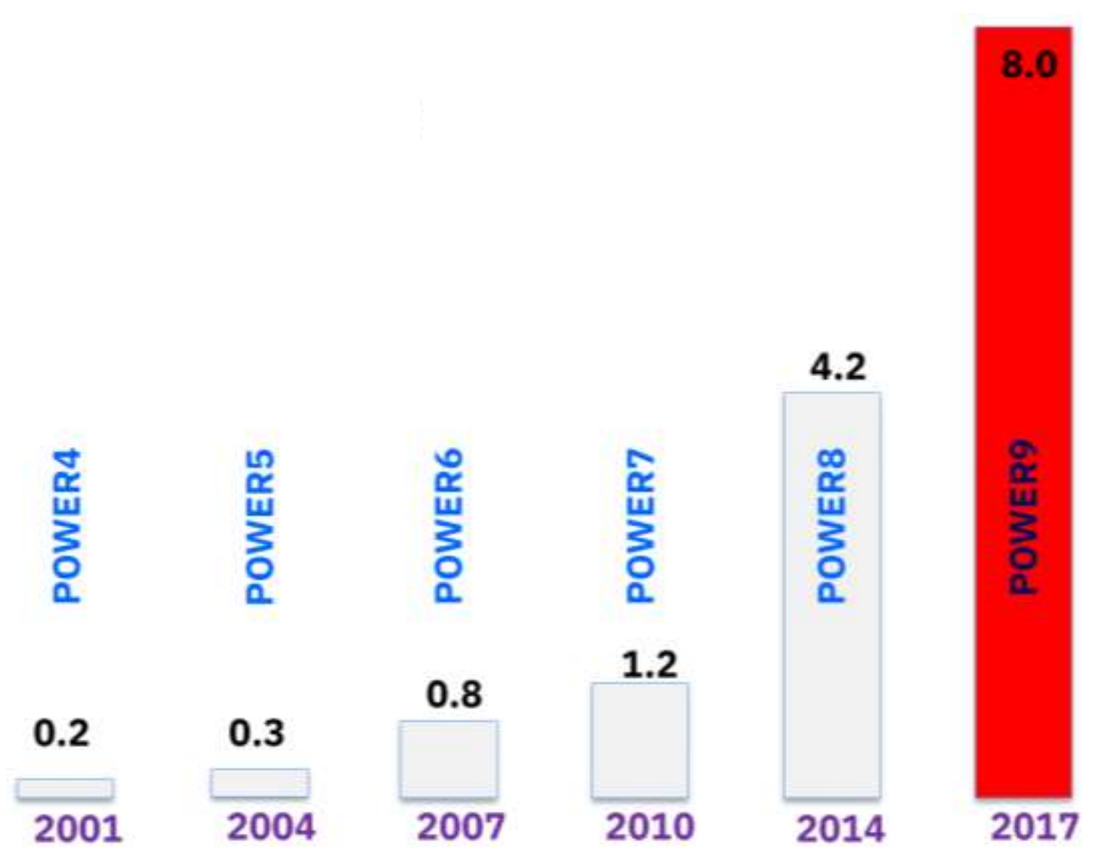
	POWER7	POWER7+	POWER8	POWER8+	POWER9
Launched	2/2010	10/2012	4/2014	Planned/cancelled	12/2017
Technology	45 nm	32 nm	22 nm		14 nm
Die size	567 mm2	567 mm2	650 mm2		693 mm2
Transistors	1.2 b	2.1 b	4.2 b		8.0 b
Cores (up to)	8	8	12		12 SMT8 cores 24 SMT4 cores
SMT	4-way	4-way	8-way		4-way/8-way
Typ. fc	3.72-4-42 GHz	3.1 -4.42 GHz	3.02-4.35 GHz		Up to 4 GHz
L2	256 KB/core	256 KB/core	512 KB/core		512KB/2 cores
L3	4 MB/core	10 MB/core	12 MB/core		10 MB/2 cores
Mem. contr.	2/1	2/1	8		8
Memory up to	DDR3-1066	DDR3-1066	DDR3-1600		DDR4-2666

POWER generations by CPU lithography [133]

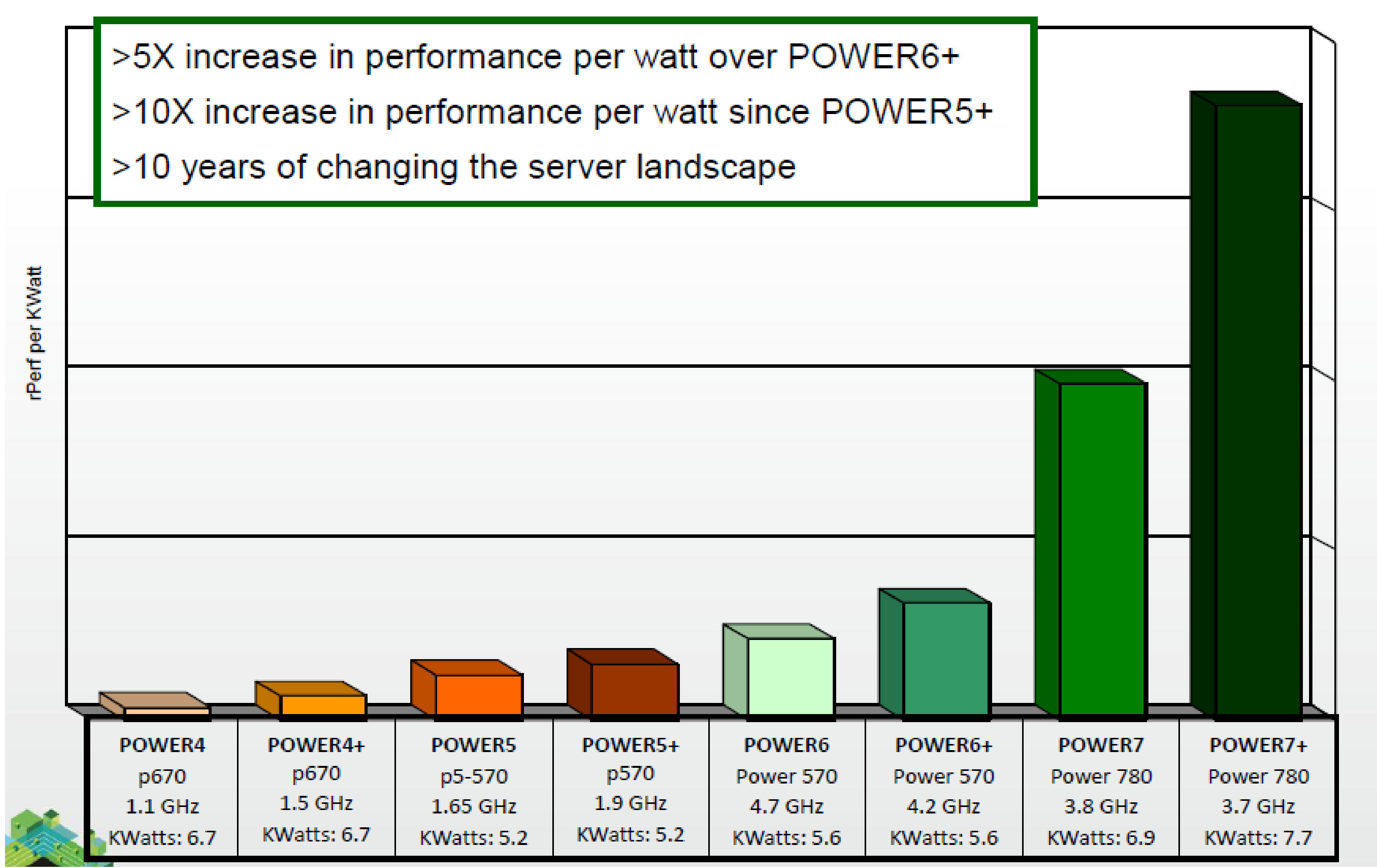
Enhancing the semiconductor technology in IBM's POWER line [169]

POWER4 180nm 2001	180nm	SOI, Copper	412mm^2	170 million transistors
POWER5 130nm 2004	130nm	SOI, Low-K	389mm^2	276 million transistors
POWER6 65nm 2007	65nm	SOI, 10L	341mm^2	790 million transistors
POWER7 45nm 2010	45nm	SOI, eDRAM, 11L	567mm^2	1.2 billion transistors (compare to 2.7 without eDRAM)
POWER8 22nm 2014	22nm	SOI, eDRAM, 15L	650mm^2	4.2 billion transistors (compare to 8.7 without eDRAM)

Transistor count (in billions) of the POWER processors [133]



Performance per kWatt figures of earlier POWER models [4]



Overview of the POWER ISA versions (Based on [134])

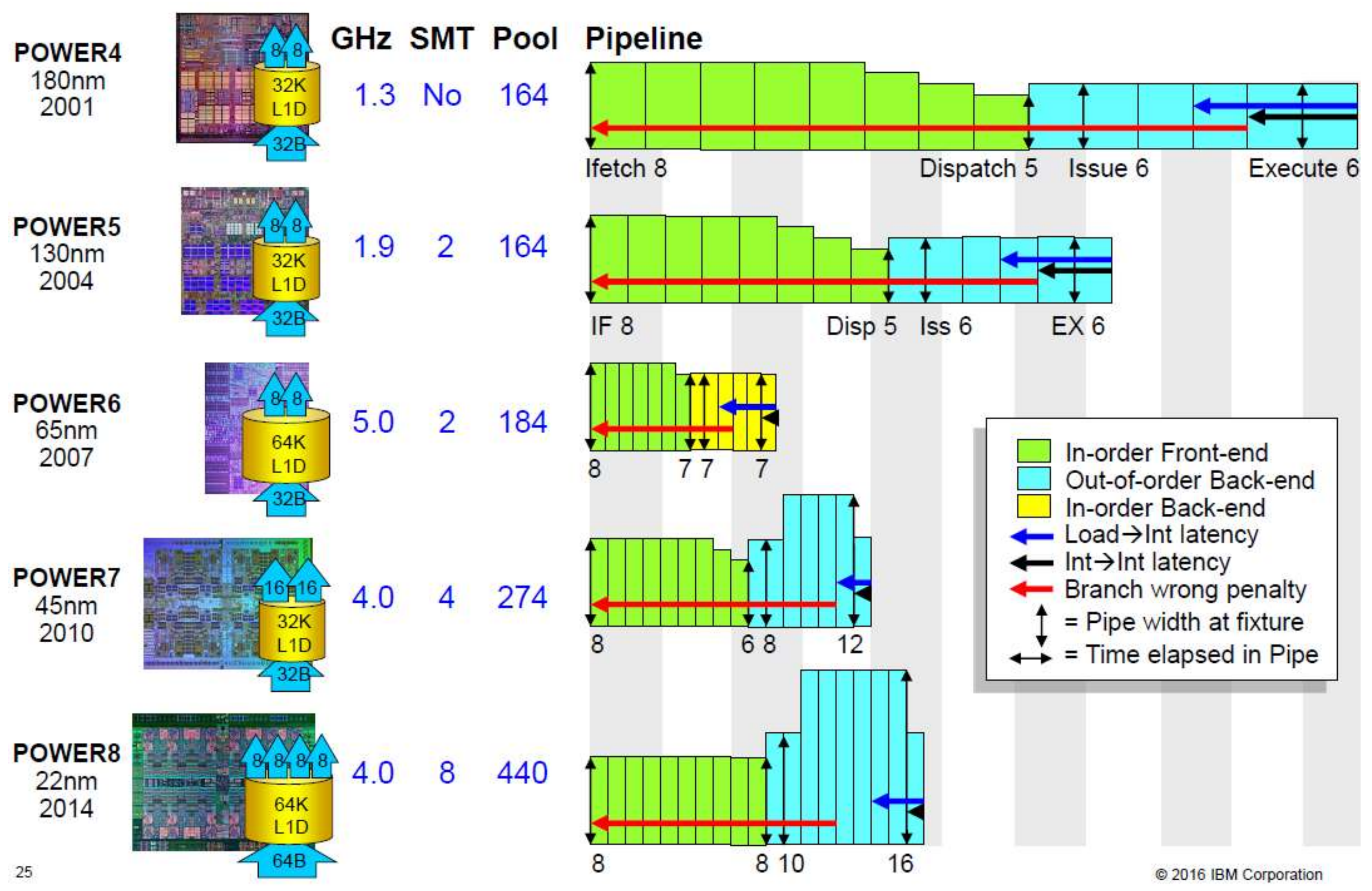
Power ISA version	Released	Main enhancements	Compliant POWER cores
PowerPC v. 2.01	9/2003		POWER4/4+
v. 2.02	01/2005		POWER5
Power ISA 2.03	06/2006	AltiVec	POWER5+
2.04	04/2007	Virtualization enhancements	POWER5
2.05	12/2007	Decimal arithmetic	POWER6/6+
2.06	02/2009	VSX (Vector Scalar)	POWER7
2.06 B	07/2010	Virtualization enhancements	POWER7/7+
2.07 with NVLink	05/2013	Transactional memory VMX, VSX2., crypto enhancements	POWER8
2.07 B	04/2015	Revised specification	POWER
3.0	11/2015	128-bit quad-precision FP operations, FP16 conversion, random number generator, hardware enforced trusted computing	POWER9 (Preliminary)
3.0 B	03/2017	Diverse instruction enhancements	POWER9

1. Overview of the POWER line (14)

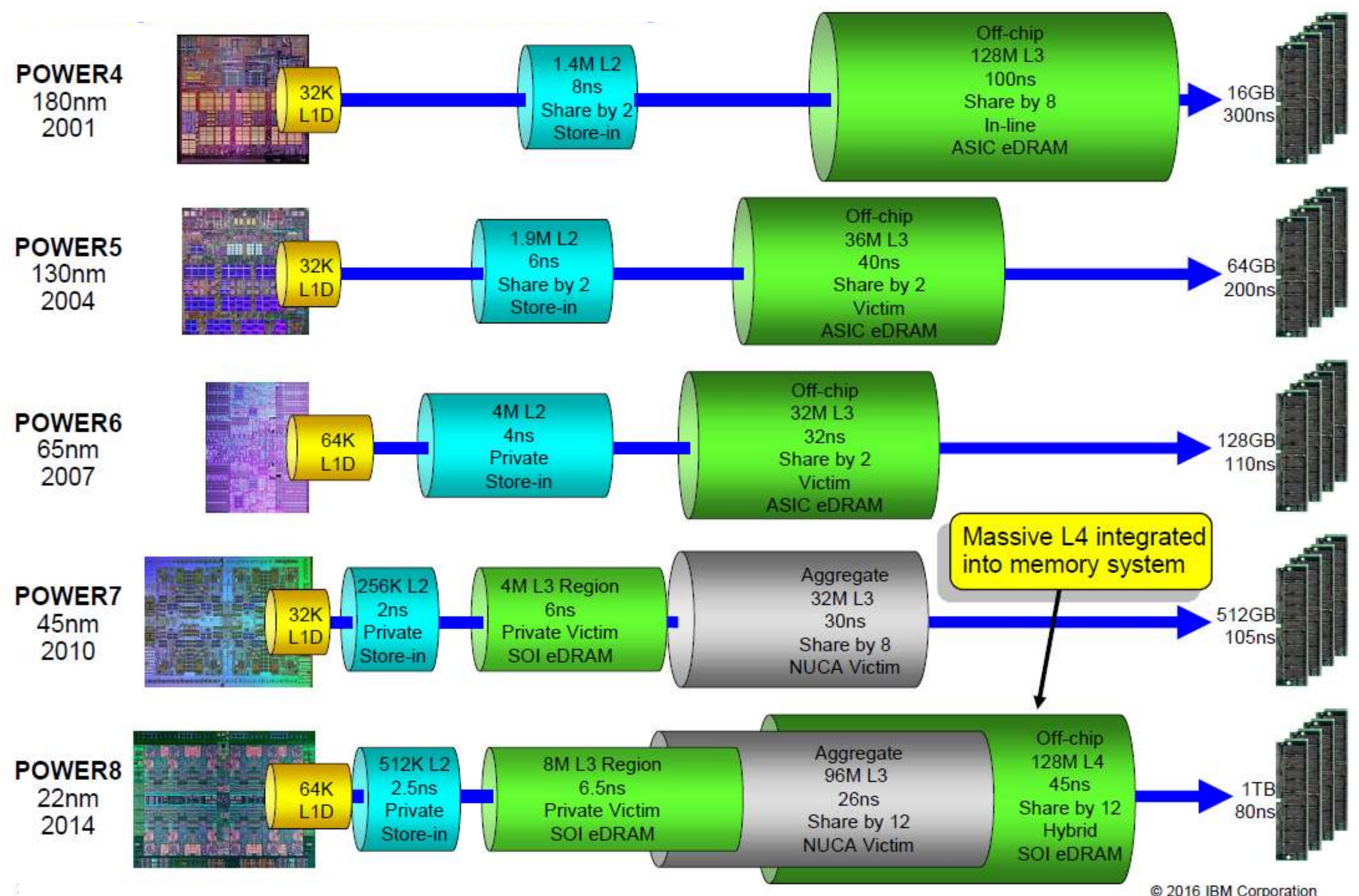
Remark [135]

- Power ISA version 2.03 was a joint development of Power.org members under the leadership of the Power Architecture Advisory Council led by IBM and Freescale (previously Motorola) from 2006.
- It includes the previous Power PC ISA version 2.2, the AltiVec vector extension and the PowerPC embedded extensions.
- It also adds features to the architecture to improve functionality and performance.

Evolution of the pipeline structure of the cores in IBM's POWER line [169]



Evolution of the cache architecture in IBM's POWER line [169] -1



Evolution of the cache architecture in IBM's POWER line -2

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-way
Typ. clock frequency	1.3 GHz	1.65- 1.90 GHz	4-5 GHz	3-4 GHz	4-4.35 GHz	3.5-4.0 GHz
L1 instr. cache per-core	64 kB	64 kB	64 kB	32 kB	32 kB	32 kB
L1 data cache per-core	32 kB	32 kB	64 kB	32 kB	64 kB	32 kB
L2 cache	1.44 MB shared	1.92 MB shared	4 MB per-core	256 KB per-core	512 KB per-core	512 KB per-core
L3 cache	32 MB shared off-chip dir. on-chip	36 MB shared off-chip	32 MB shared off-chip	8x4 MB tightly coupled on-chip eDRAM victim cache	12x8 MB tightly coupled on-chip eDRAM victim cache	12x10 MB Per 2 cores
L4 cache	--	--	--	--	up to 8x16 MB eDRAM on 8 mem.buffers (not snooped)	up to 8x16 MB eDRAM on 8 mem.buffers (not snooped)

Enhancing the core's execution resources in IBM's POWER line

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-ways
Width of the front-end	5	5	5	6	8	12
Dispatch rate	5	5	(In-order design)	6	8	12
Issue rate	8	8	7	8	10	16
No. of execution units per-core	8	8	9	12	16	20
No/type of execution units per-core	2 FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP 1BR/CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 2VMX, 1DFU, 2LU, 1 Crypto	8AGEN, 4VSU(128) 4LS(128) 2BRU DFU Cripto

Main features of the memory subsystems in IBM's POWER lines -1

Model	Year	No. of MCs[1] (up to)	No. of MBs/ MC (up to)	MC-MB link	Read/ write width of MC-MB link	Speed of MC-MB link	No. of DIMM chann./ MB or MC	DIMM type	DRAM speed (up to)
POWER4	2001	1	4	Bidir. P2P	4 B	2x DRAM speed	1	Propr. DIMM	DDR-200
POWER5	2004	1	2	Unidir. P2P	4 B read 2 B write	2x DRAM speed	2	Propr. DIMM	DDR2-533
POWER6	2007	1[1]	4	Unidir. P2P	2 B read 1 B write	4x DRAM speed	2	Commod. DIMM	DDR2-667
		FB-DIMM	4	Unidir. P2P	14+10 lanes	6x DRAM speed	1	FB-DIMM	DDR2-667
POWER7	2010	1[1]	4	Unidir. P2P	2 B read 1 B write	6.4 Gb/s	2	Commod. DIMM	DDR3-1066
		2	4	Unidir. P2P	20+13 lanes	6x DRAM speed (6.4 Gb/s)	1	Propr. FB-DIMM	DDR3-1066
POWER8	2014	2(8)[1]	4	Unidir. P2P	2 B read 1 B write	9.6 Gb/s	1	Propr. DIMM	DDR3-1600

MC: Memory controller MB: Memory buffer (POWER4-7: SMI buffer, POWER8-9: Centaur buffer)

Commod.: Commodity Prop.: Proprietary Bidir.: Bidirectional Unidir.: Unidirectional P2P: Point-to-Point

Main features of the memory subsystems in IBM's POWER lines -2

Model	Year	No. of MCs[1] (up to)	No. of MBs/ MC (up to)	MC-MB link	Read/ write width of MC-MB link	Speed of MC-MB link	No. of DIMM chann./ MB or MC	DIMM type	DRAM speed (up to)
POWER9 Scale-Out	2017	2	--	--	--	--	4	Commod. DIMM	DDR4-2666
POWER9 Scale-Up	2018	2	4	Unidir. P2P	2 B read 1 B write	9.6 Gb/s	4	Commod.. DIMM	DDR4-1600

MC: Memory controller MB: Memory buffer MB: Memory buffer (POWER4-7: SMI buffer, POWER8-9: Centaur buffer)

Commod.: Commodity Prop.: Proprietary Bidir.: Bidirectional Unidir.: Unidirectional P2P: Point-to-Point

1. Overview of the POWER line (21)

[1]: Here we note that actual implementations often do not make use of both memory controllers available on the chip.

MC-MB link limited memory bandwidth in IBM's POWER line -1

Model	Tech.	Intro.	No. of cores (up to)	fc up to	SMT	DIMM type	DRAM speed	No/speed/width of MC-MB links (up to)	MC-MB link limited BW/proc. (up to)	BW/fc/ /core (byte/cycle) (up to)
POWER 3-II	250 nm	1999	1	0.45 GHz	No	Propr. DIMM	SDRAM - 100	2@100 Kbits/s 8B R/W	1.6 GB/s	3.5
POWER 4	180 nm	2001	2	1.3 GHz	No	Propr. DIMM	DDR-200	8@400 Kbit/s 4B R/W	12.8 GB/s	4.9
POWER 4+	130 nm	2002	2	1.7 GHz	No	Propr. DIMM	DDR-200	8@400 Kbit/s 4B R/W	12.8 GB/s	3.5
POWER 5	130 nm	2004	2	1.9 GHz	2-way	Propr. DIMM	DDR2-533	4@1066 Kbit/s 4B R/2B W	25.6 GB/s	6.8
POWER 5+	90 nm	2005	2	2.3 GHz	2-way	Propr. DIMM.	DDR2-533	4@1066 Kbit/s 4B R/2B W	25.6 GB/s	5.5
POWER 6	65 nm	2007	2	5.0 GHz	2-way	Commod. DIMM	DDR2-667	4@2.67 Gbit/s 2B R/1B W	32.0 GB/s	3.2
						FB-DIMM	DDR2-667	8@4.0 Gb/s 12b R/6b W	72.0 GB/s	7.2
POWER 6+	65 nm	2008	2	5.0 GHz	2-way	Commod. DIMM	DDR2-667	4@2.67 Gbit/s 2B R/1B W	32.0 GB/s	3.2
						FB-DIMM	DDR2-667	8@4.0 Gb/s 12b R/6b W	72.0 GB/s	7.2

Commod.: Commodity Prop.: Proprietary MB: Memory buffer (POWER4-7: SMI buff.-POWER8-9: Centaur buff-)

MC-MB link limited memory bandwidth in IBM's POWER line -2

Model	Tech.	Intro.	No. of cores (up to)	fc (up to)	SMT	DIMM type	DRAM speed	No/speed/width of MC-MB links (up to)	MC-MB link limited BW/proc. (up to)	BW/fc/ /core (byte/cycle) (up to)
POWER7	45 nm	2010	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gbit/s 2B R/1B W	76.8 GB/s	2.2
						Propr. FB-DIMM	DDR3-1066	8@6.4 Gb/s 2B R/3B W	153.6 GB/s	4.4
POWER7+	32 nm	2013	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	153.6 GB/s	3.9
						Propr. FB-DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	76.8 GB/s	2.2
POWER8	22 nm	2014	12	4.35 GHz	8-way	Propr. CDIMM	DDR3-1600	2(8)[1]@9.6 Gbit/s 2B R/1B W	57.5 (230 GB/s)	1.1 (4.4)
POWER9 (Scale-Out)	14 nm	2017	12 24	4.00 GHz	8-way 4-way	Commod. DIMM	DDR4-2666	--	--	--
POWER9 (Scale-Up)		2018	12	4.00 GHz	8-way	Commod. DIMM	DDR4-1600	8@9.6 Gbit/s 2B R/1B W	230,4 GB/s	4.79

[1]: According to IBM's literature [] the POWER8 has up to eight memory channels.
Nevertheless, first servers delivered until 05/2015 makes use of only two of them.

Commod.: Commodity Prop.: Proprietary MB: Memory buffer (POWER4-7: SMI buff.-POWER8-9: Centaur buff-)

DIMM channels limited memory bandwidth in IBM's POWER line-1

Model	Tech.	Intro.	Max. no. of cores /SMT	fc (up to)	DIMM type	No. of MC-MB links (up to)	No. of DIMM chan./ MB	No./speed of DIMM chan. (up to)	DIMM chan. limited BW/proc. (up to)	BW/fc/ /core (B/cycle) (up to)
POWER3-II	250 nm	1999	1 core/ no SMT	0.45 GHz	Propr. DIMM	2	2	4xSDRAM-100	3.2 GB/s	7.1
POWER4	180 nm	2001	2 cores/ no SMT	1.3 GHz	Propr. DIMM	4	2	8xDDR-200	12.8 GB/s	4.9
POWER4+	130 nm	2002	2 cores/ no SMT	1.7 GHz	Propr. DIMM	4	2	8xDDR-200	12.8 GB/s	3.8
POWER5	130 nm	2004	2 cores/ 2-way	1.9 GHz	Propr. DIMM	4	2	8xDDR2-533	34.1 GB/s	9.0
POWER5+	90 nm	2005	2 cores/ 2-way	2.3 GHz	Propr. DIMM.	4	2	8xDDR2-533	34.1 GB/s	7.4
POWER6	65 nm	2007	2 cores/ 2-way	5.0 GHz	Commod. DIMM	4	2	8xDDR2-667	42.7 GB/s	4.3
					FB-DIMM	8	1	8xDDR2-667	42.7 GB/s	4.3
POWER6+	65 nm	2008	2 cores/ 2.way	5.0 GHz	Commod. DIMM	4	2	8xDDR2-667	42.7 GB/s	4.3
					FB-DIMM	8	1	8xDDR2-667	42.7 GB/s	4.3

Commod.: Commodity Prop.: Proprietary

DIMM channels limited memory bandwidth in IBM's POWER line-2

Model	Tech.	Intro.	Max. no. of cores /SMT	fc (up to)	DIMM type	No. of MC-MB links (up to)	No. of DIMM chan./ MB or MC	No./speed of DIMM chan./proc. (up to)	DIMM chan. limited BW/proc. (up to)	BW/fc/ /core B/cycle (up to)
POWER7	45 nm	2010	8 cores/ 4-way	4.42 GHz	Commod. DIMM	4	2	8xDDR3-1066	68.2 GB/s	1.9
					FB-DIMM	8	1	8xDDR3-1066	68.2 GB/s	1.9
POWER7+	32 nm	2013	8 cores/ 4.way	4.42 GHz	Commod. DIMM	4	2	8xDDR3-1066	68.2 GB/s	1.9
					FB-DIMM	8	1	8xDDR3-1066	68.2 GB/s	1.9
POWER8	22 nm	2014	12 cores/ 8-way	4.35 GHz	Propr. CDIMM	2[1]	1	2[1]xDDR3-1600	102.4 GB/s	2.9
						8[1]			409.6 GB/s	8.5
POWER9 (Scale-Out)	14 nm	2017	12 cores/ 8-way	4 GHz	Commod. DIMM	--	8	8xDDR4-2666	170.7	4.6
			24 cores/ 4-way							2.3
POWER9 (Scale-Up)		2018	12-cores 4-way	4 GHz	Commod. DIMM	8	4	32xDDR4-1600	409.6	8.5

Commod.: Commodity
Prop.: Proprietary

[1]: According to IBM's literature [] the POWER8 has up to eight memory channels. Nevertheless, first servers delivered until 05/2015 make use of only two of them.

Remark

[1]DDR2-533 DIMMs in POWER5+ based systems operate at DDR2-528 rate for a not specified reason.

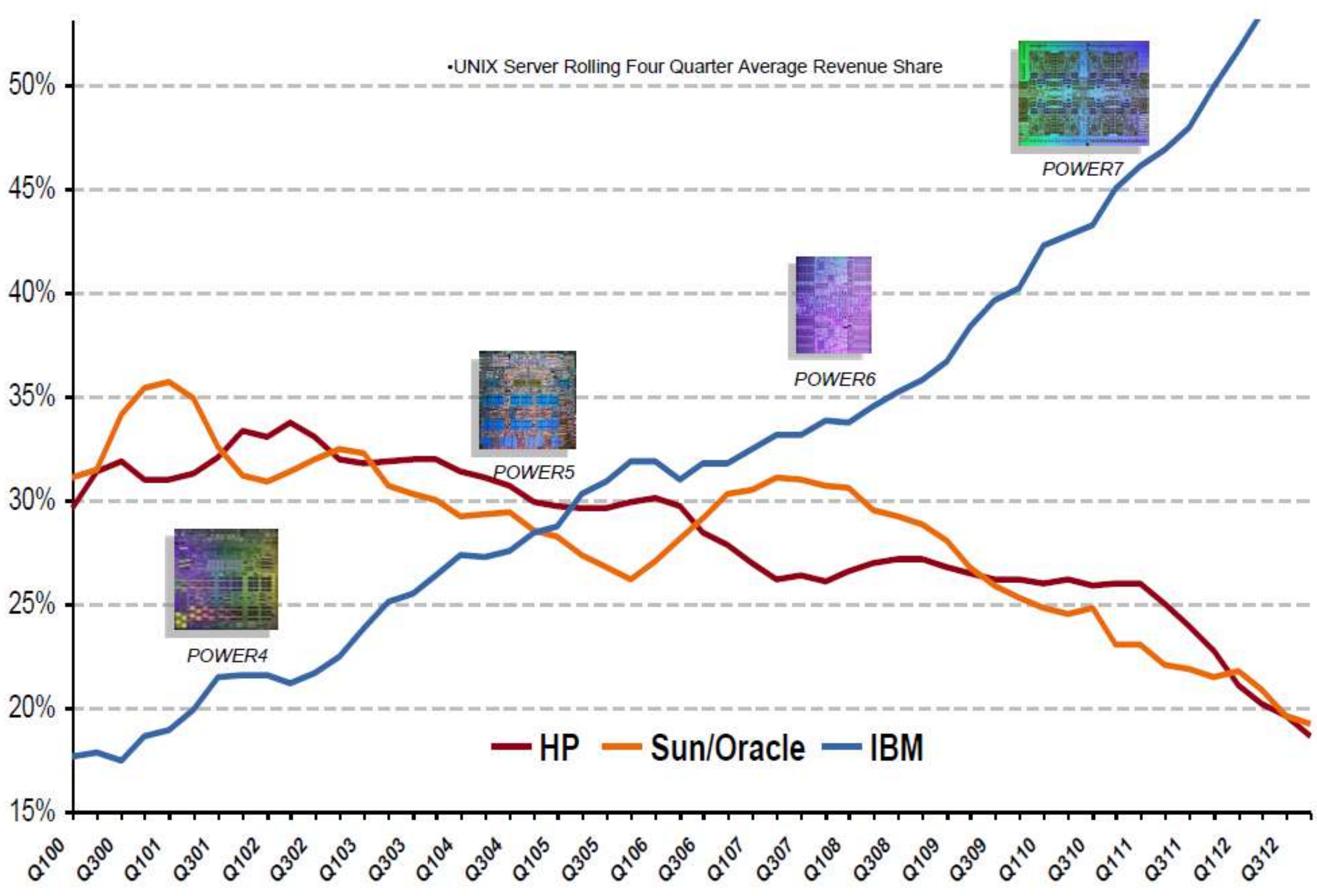
Evolution of the GX I/O bus in IBM's POWER line

Based on available IBM literature, the Table below illustrates the evolution of the GX bus.

Model	Designation of the GX bus	Speed	Exemplary total bandwidth per GX bus
POWER4	GX	1/3 fc e.g. 400 MHz for 1.2 GHz	3.2 GB/s
POWER5	GX+	1/3 fc e.g. 700 MHz for 2.1 GHz	5.6 GB/s
POWER6	GX++	¼ fc e.g. 1.05 GHz for 4.2 GHz	8.4 GB/s
POWER7	GX+/GX++	1.25 GHz 2.50 GHz	10 GB/s/ 20 GB/s
POWER8	--	--	--

Table: Evolution of the features of the GX I/O bus in IBM's POWER line

Market share of IBM's POWER line in the UNIX server market [169]



2. POWER4

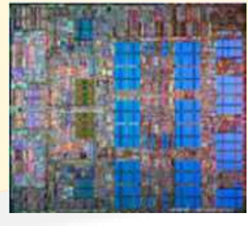
- 2.1 Introduction to the POWER4
- 2.2 Key features of the POWER4
- 2.3 POWER4 based SMP configurations

2. The POWER 4 (Die photos from [3])

POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
with SMI chips
GX I/O bus
Support for SMP

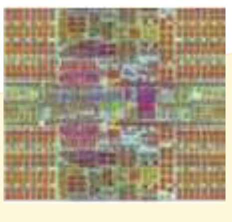
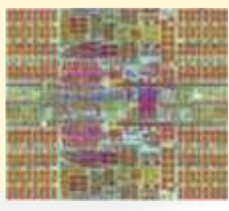
2001



POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
clock gating

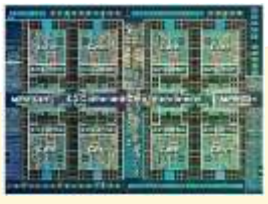
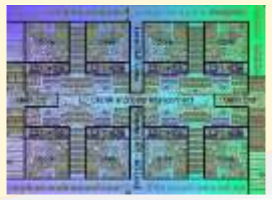
2004

POWER6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
Critical Path
Monitors
Nap idle mode

2007

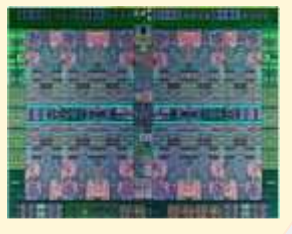



POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Energy Scale 2 with
Per core fc
Dyn. fan managm.
Sleep idle mode
*Accelerators for
cryptography
*Winkle idle mode

*POWER7+

2010



POWER8
22 nm

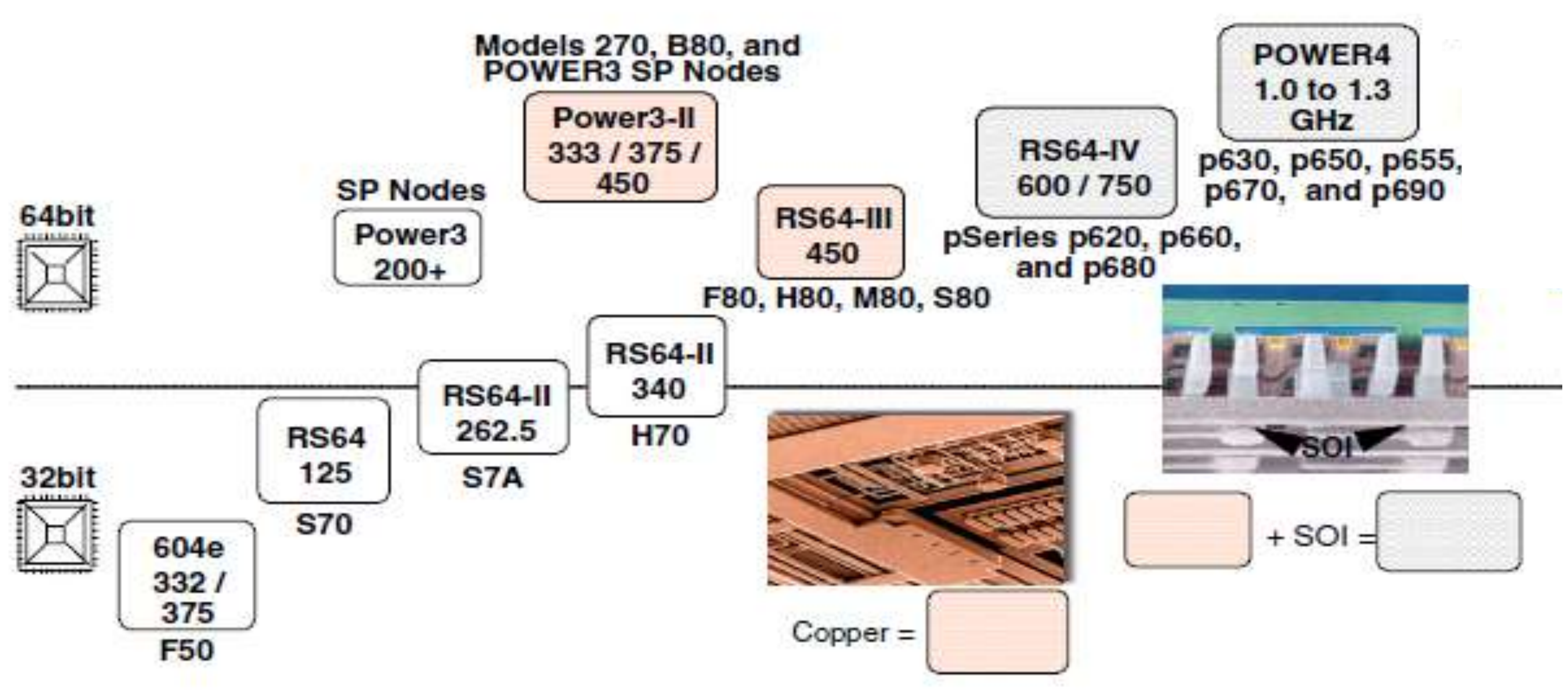
12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
with distributed L4
no FB-DIMM option
CAPI
NVLink
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2014

2.1 Introduction to the POWER4

2.1 Introduction to the Power4

Overview of IBM's server models up to the POWER4 [7]

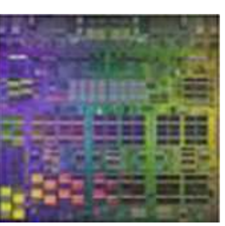
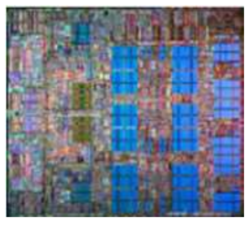


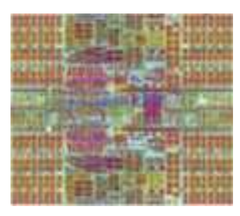
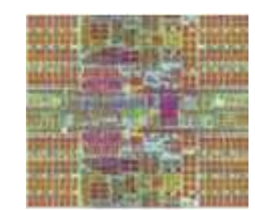
2.1 Introduction to the POWER4 (2)

Introduction to the Power4

- Design started in 1996 with 250 system and chip designers, software architects, researchers and semiconductor engineers.
- Introduced 10/2001, shipped in 12/2001

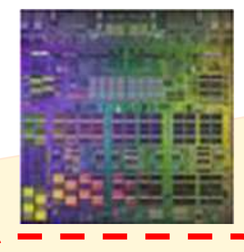
Key features of the POWER 4

	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

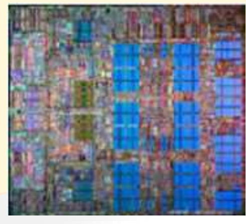
Key innovations of the POWER4 (Die photos from [3])



POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
with SMI chips
GX I/O bus
Support for SMP

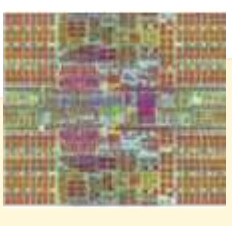
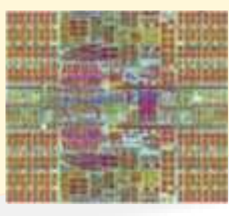
2001



POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
clock gating

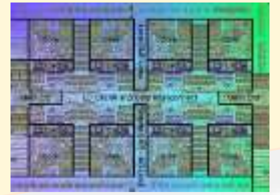
2004

POWER6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
Critical Path
Monitors
Nap idle mode

2007

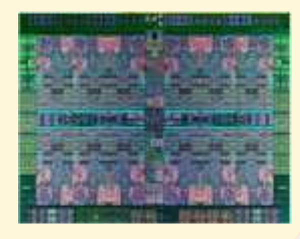


POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
Per core fc
Dyn. fan managm.
Sleep idle mode
*Accelerators for
cryptography
*Winkle idle mode

*POWER7+

2010

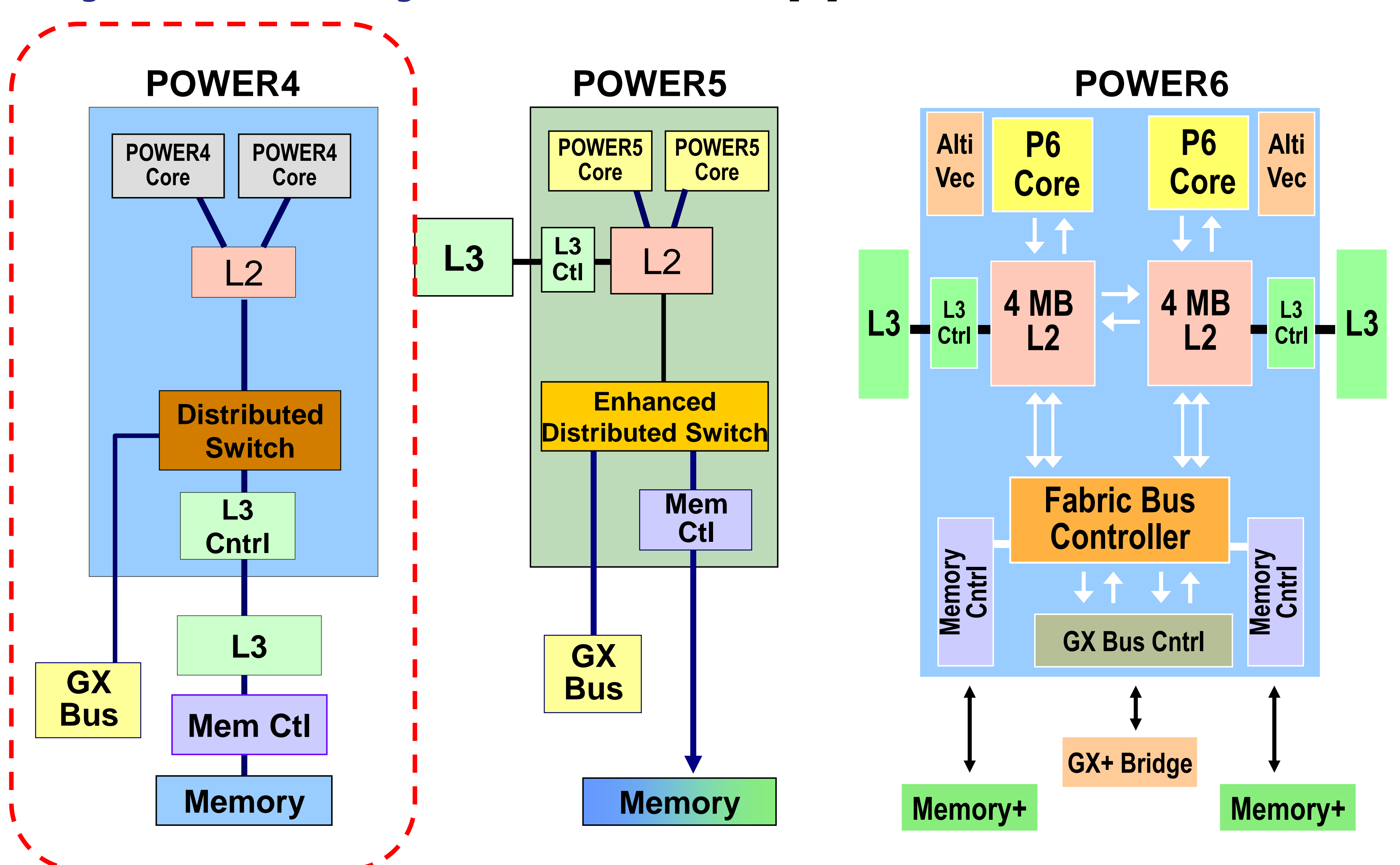


POWER8
22 nm

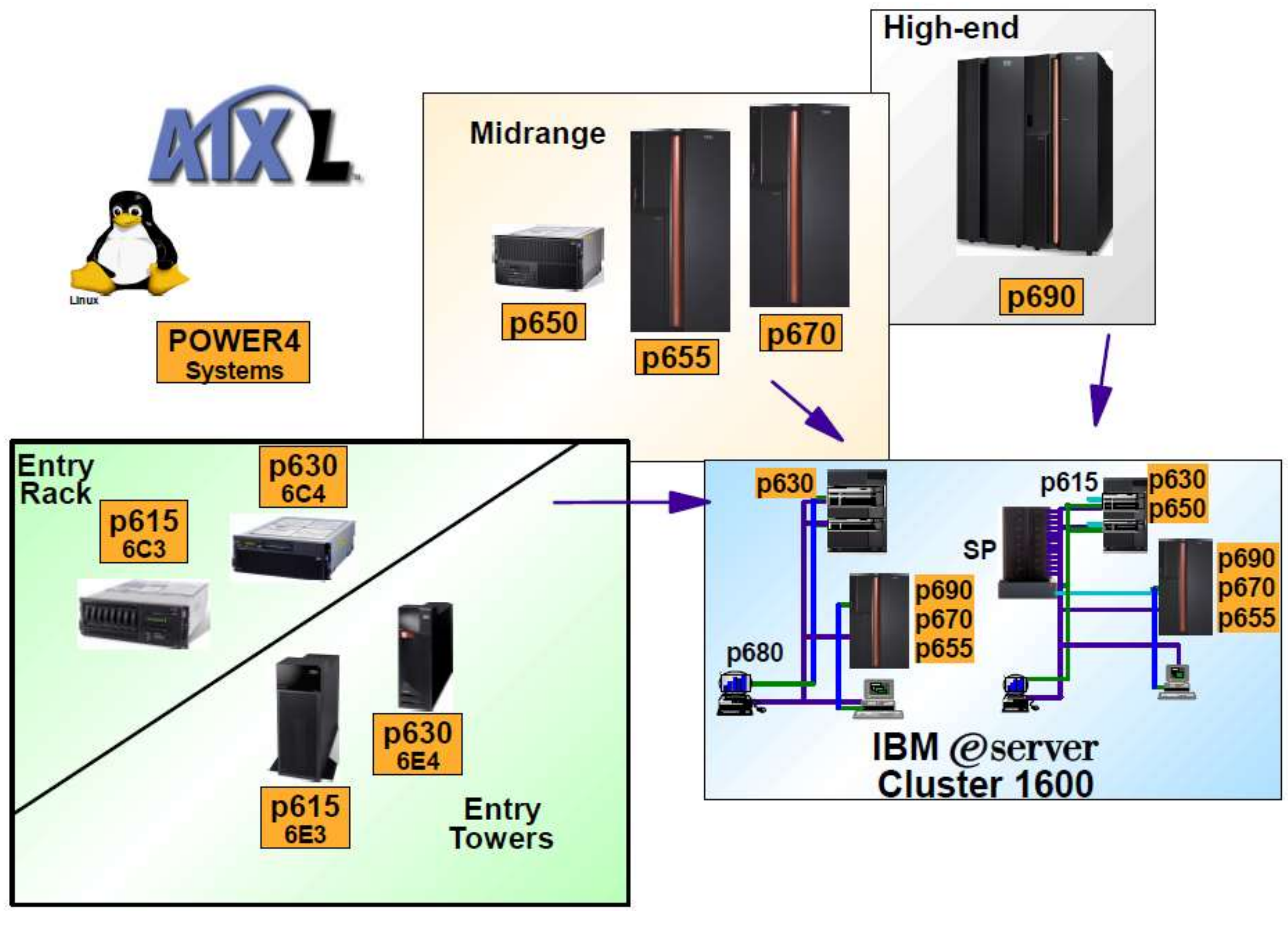
12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
with distributed L4
no FB-DIMM option
CAPI
NVLink
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2014

High level block diagram of the POWER4 [5]

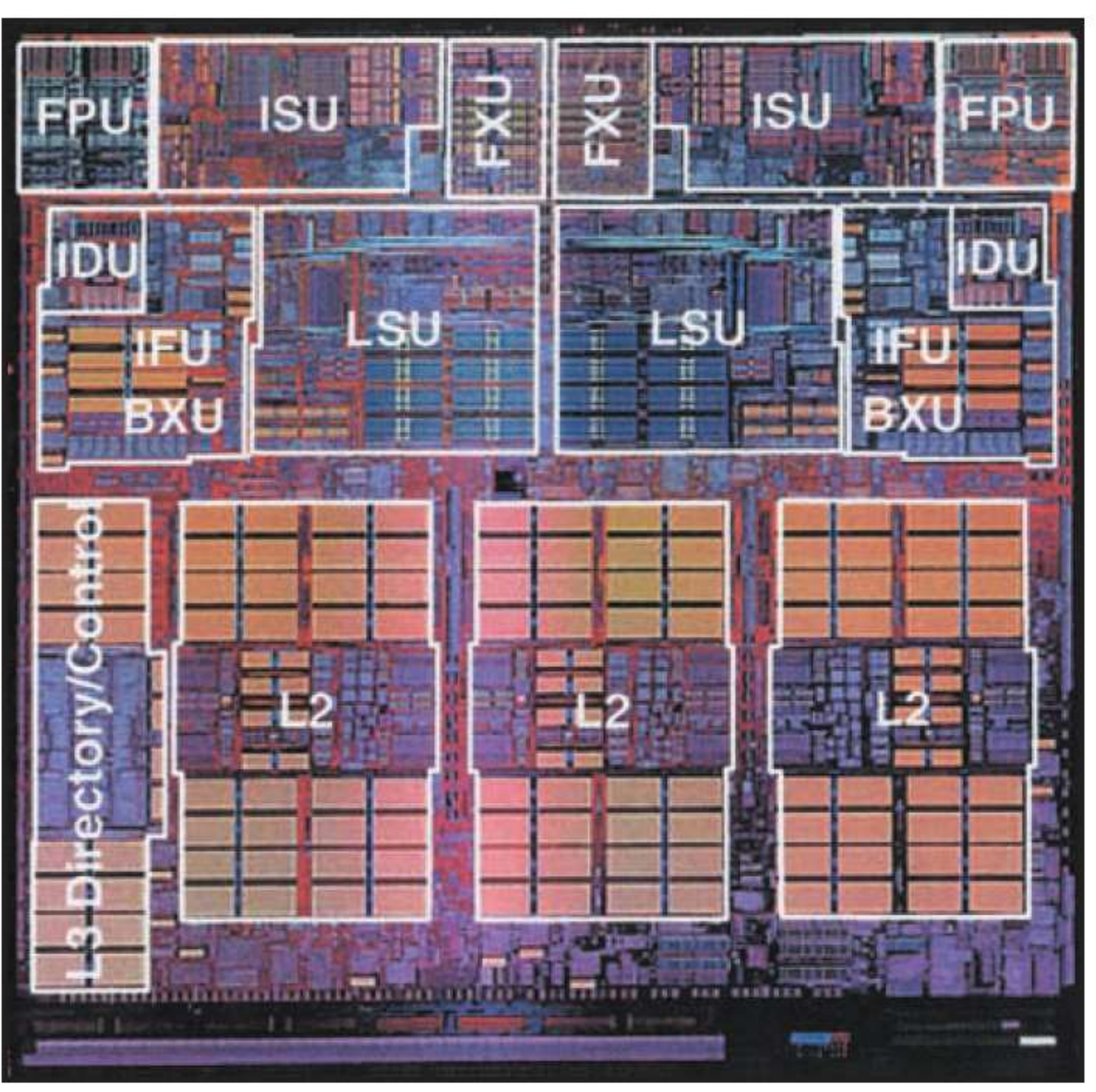


Overview of POWER4 based pSeries servers [8]



Floor plan of the POWER4 [6]

180 nm technology
414 mm^2
174 million transistors



2.2 Key features of the POWER4

- 2.2.1 Dual core processor
- 2.2.2 Superscalar design with instruction grouping
- 2.2.3 In-order dispatch at a rate of one group/cycle
- 2.2.4 Out-of-order issue at a rate of 8 instr./cycle
- 2.2.5 In-order completion at a rate of one group/cycle
- 2.2.6 16 stage pipeline design
- 2.2.7 Off-chip L3 cache
- 2.2.8 Linking memory via low pin count serial point-to-point buses and buffer chips (SMI chips)
- 2.2.9 Introduction of the GX bus
- 2.2.10 Vastly improved support for SMP

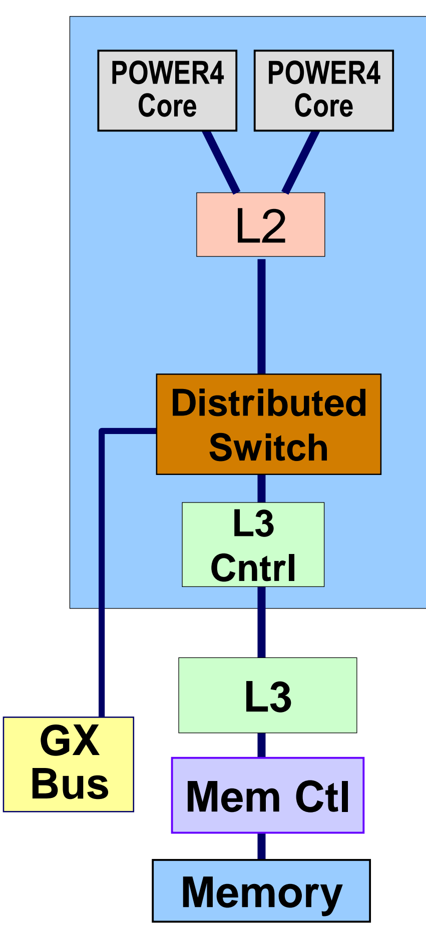
2.2.1 Dual core processor

IBM's POWER4 was industry's first dual core design preceding all other dual core core processors of other manufacturers, as indicated in the next Table.

Year of launching	Dual core design
10/2001	IBM launches dual core POWER4
11/2002	IBM launches dual core POWER4+
08/2003	ARM shows first synthesizable MP processor
05/2004	ARM demonstrates the ARM11 MPCore quad core processor
05/2004	IBM launches dual core POWER5
08/2004	AMD demonstrates first x86 dual core (Opteron) processor
04/2005	ARM launches the ARM11 MPCore quad core processor
04/2005	Intel launches dual core Pentium processors (Pentium D)
04/2005	AMD launches dual core Opteron server processors

Table: Emergence of dual core processors

High level simplified block diagram of the dual core POWER4 [5]



2.2.2 Superscalar design with instruction grouping -1

- To minimize the logic necessary for tracking a large number of in-flight instructions (> 200) IBM introduced instruction grouping in the POWER4.
- This means that after fetching, instructions are forwarded from the I-cache into an Instruction queue, they become decoded, some of them are split (cracked) into two or more instructions (like a load with update into a load and a register update instruction to simplify execution) and finally instructions are grouped, into groups of up to five instructions, as indicated in the next Figure.

2.2.2 Superscalar design with instruction grouping -2

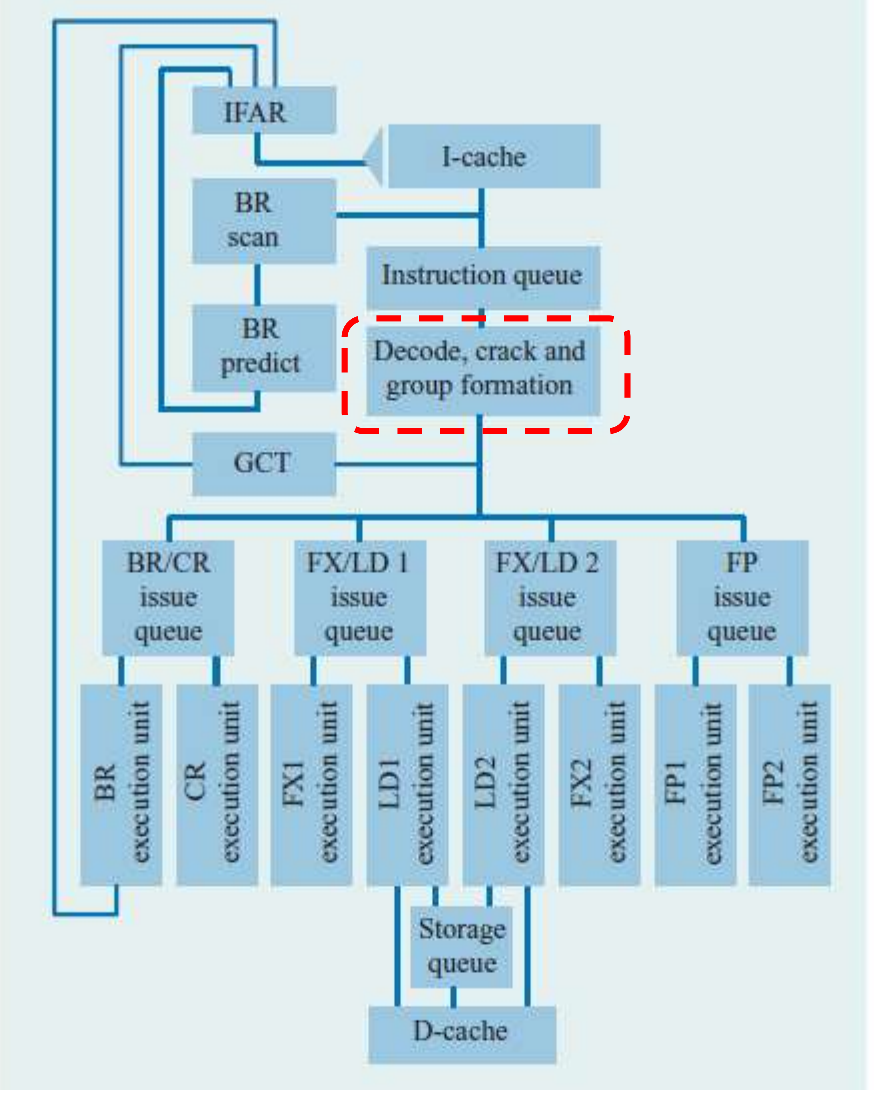


Figure: Instruction grouping within the flow of instruction processing [6]

2.2.2 Superscalar design with instruction grouping -3

Instruction decoding, cracking and group formation occurs in the D0 to GD pipeline cycles, as the next Figure shows.

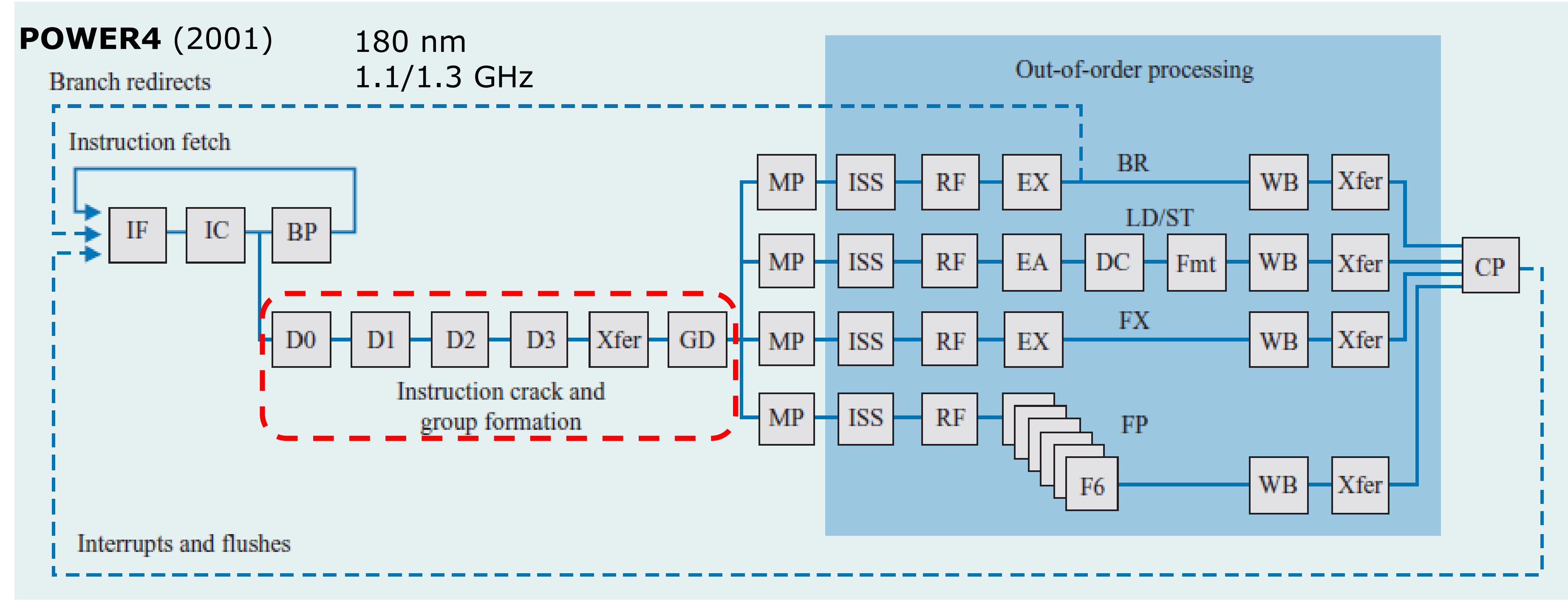


Figure: Decoding, cracking and grouping of instructions [6]

2.2.2 Superscalar design with instruction grouping -4

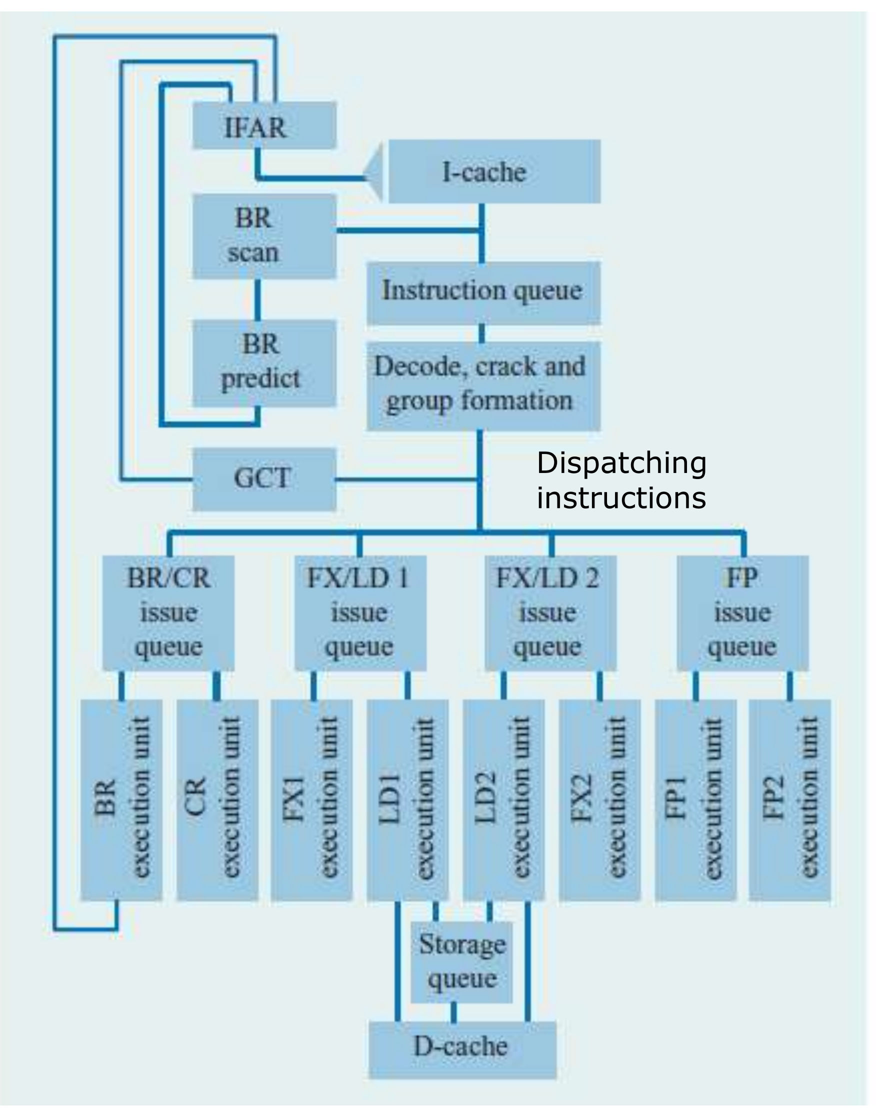
- Groups are formed by placing instructions sequentially in the five slots of a group—the oldest instruction is placed in slot 0, the next oldest one in slot 1, and so on.
- Slot 4 is reserved solely for branch instructions, if required, no-ops are inserted to force the branch instruction to be in slot four.

  If there is no branch instruction, slot 4 contains a no-op.
- Individual groups are tracked as entities through the system in a so called Group Completion Table (GCT).

  The state of the machine is preserved only at group boundaries, instead of instruction boundaries within a group.
- Any exception causes then the machine to restore its state to the state of the oldest group prior to the exception.

2.2.3 In-order dispatch at a rate of one group/cycle [6]

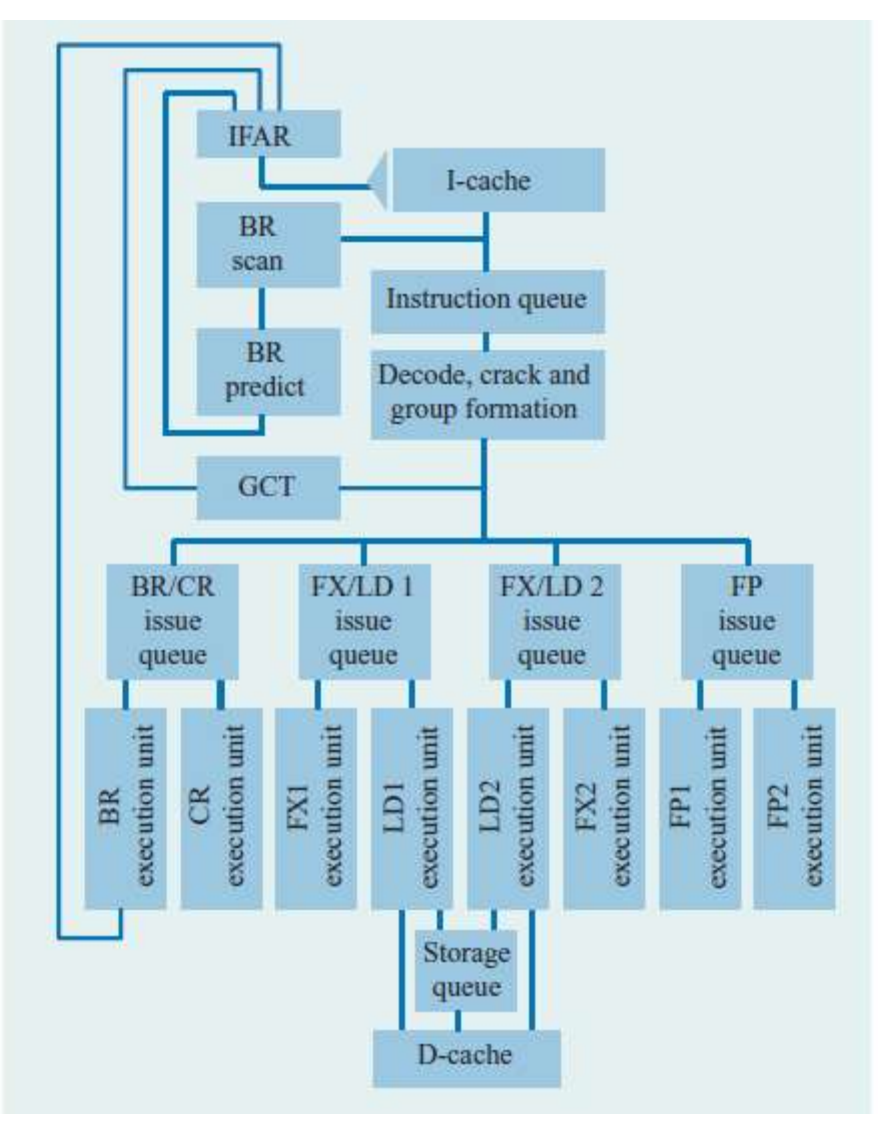
- After having placed into groups, instructions are renamed and dispatched, i.e. forwarded into the issue queues (in the Mapping cycle (MP)), as shown in the Figure on the right.
- Instructions are dispatched in groups, i.e. all instructions in a group are dispatched together.
- Groups are dispatched in program order, one group (i.e. up to five instructions) in a cycle.



2.2.4 Out-of-order issue at a rate of 8 instructions/cycle [6]

- From the issue queues individual instructions having no dependencies are issued to the 8 execution units out-of-program order, at a rate of 8 instructions/cycle.

Out-of-order instruction issue



2.2.5 In-order completion at a rate of one group/cycle-1

- Instructions with no dependencies are issued, access their operands and become executed in the related units.
- After finishing execution, instructions write their results back in the WB cycle, but instruction execution has not yet been completed.

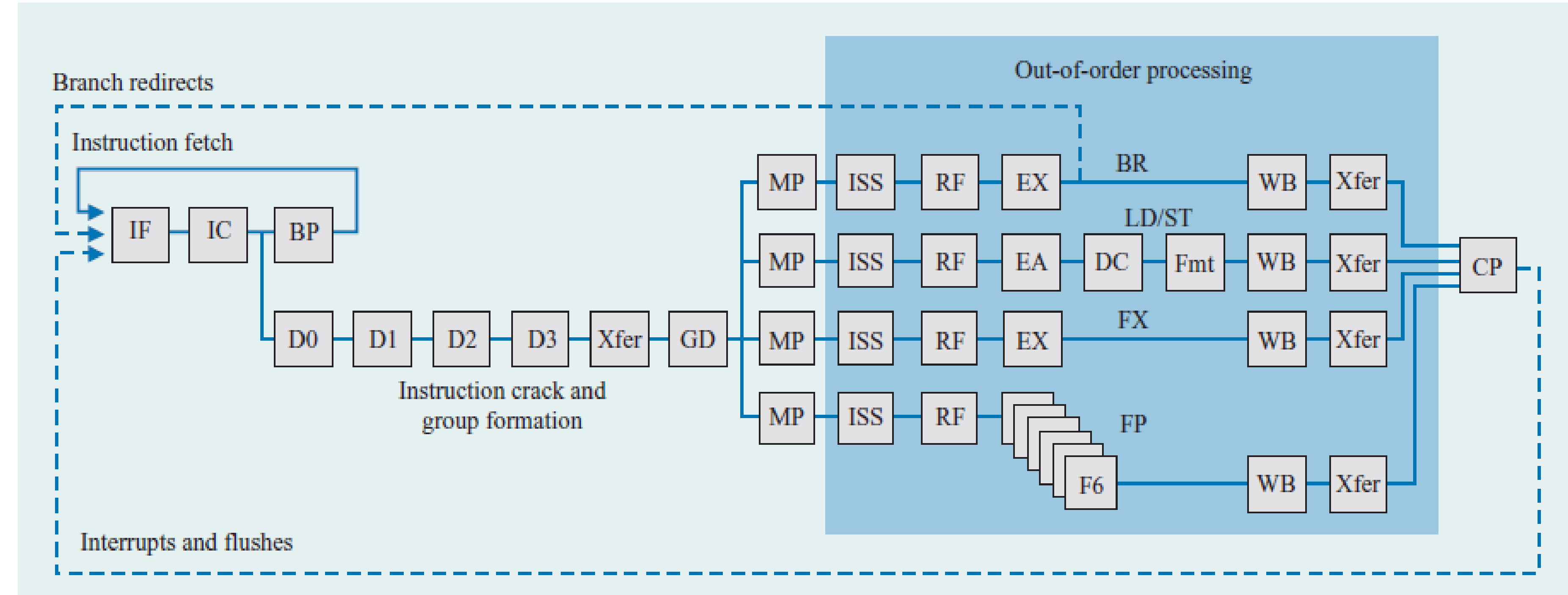


Figure: Pipelined instruction execution in the POWER4 [6]

2.2.5 In-order completion at a rate of one group/cycle -2

- Instructions complete on a group basis in program order in the Xfer and CP cycles.
- A group will complete when all older groups have completed and when all instructions in the group have finished execution.
- Only one group can complete in the same cycle.

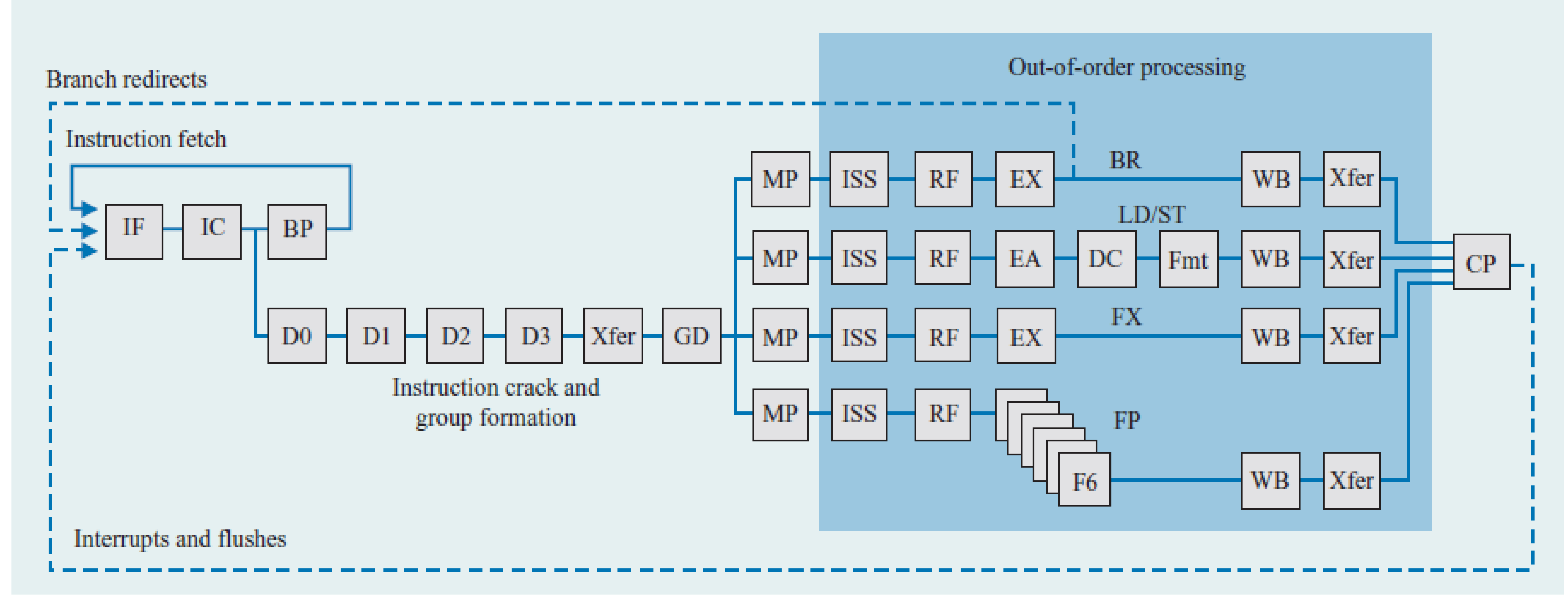
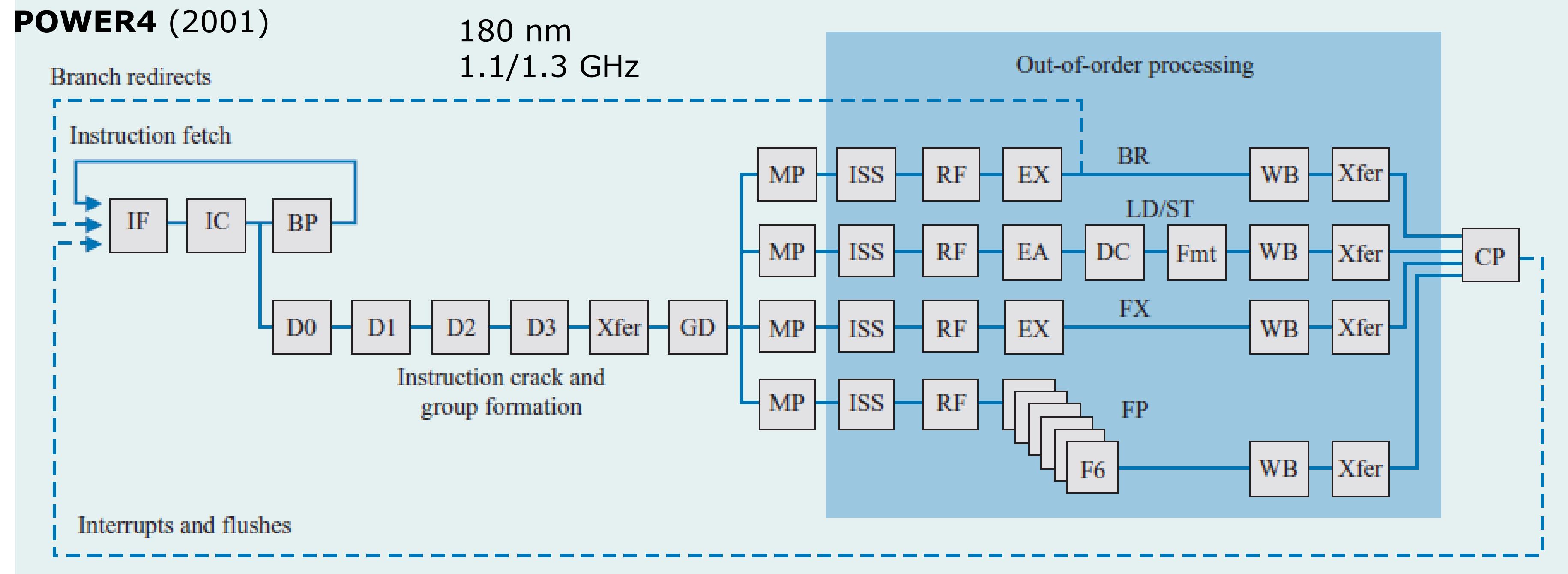


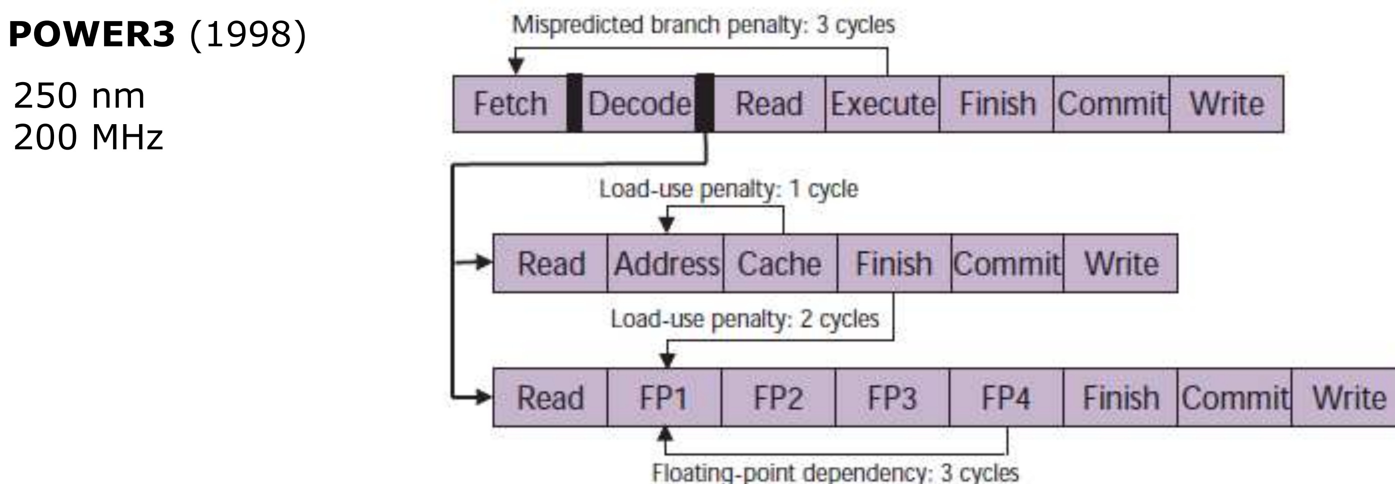
Figure: Pipelined instruction execution in the POWER4 [6]

2.2.6 16 stage pipeline design [6]



2.2.6 16 stage pipeline design (2)

Greatly increased pipeline length vs. POWER3 (from 7 to 16) to raise fc [9],



POWER4 (2001) 180 nm
1.1/1.3 GHz

Branch redirects

Out-of-order processing

Instruction fetch

IF IC BP

D0 D1 D2 D3 Xfer GD

Instruction crack and group formation

MP ISS RF EX BR WB Xfer

MP ISS RF EA DC Fmt LD/ST WB Xfer

MP ISS RF EX FX WB Xfer

MP ISS RF F6 FP WB Xfer

CP

Interrupts and flushes

2.2.7 Off-chip L3 cache

POWER4 introduced an external L3 cache of 32 MB for better supporting dual cores, whereas the L3 directory is on the processor die, as indicated below.

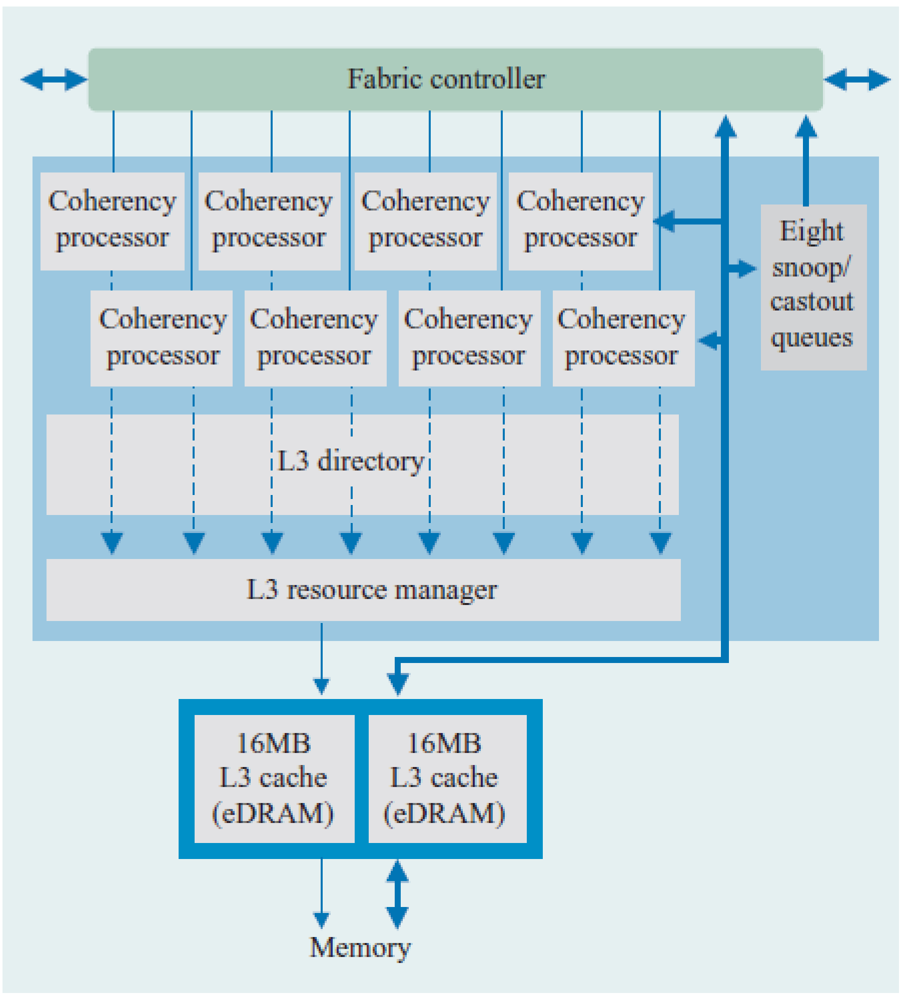


Figure: The external L3 cache of the POWER4 [6]

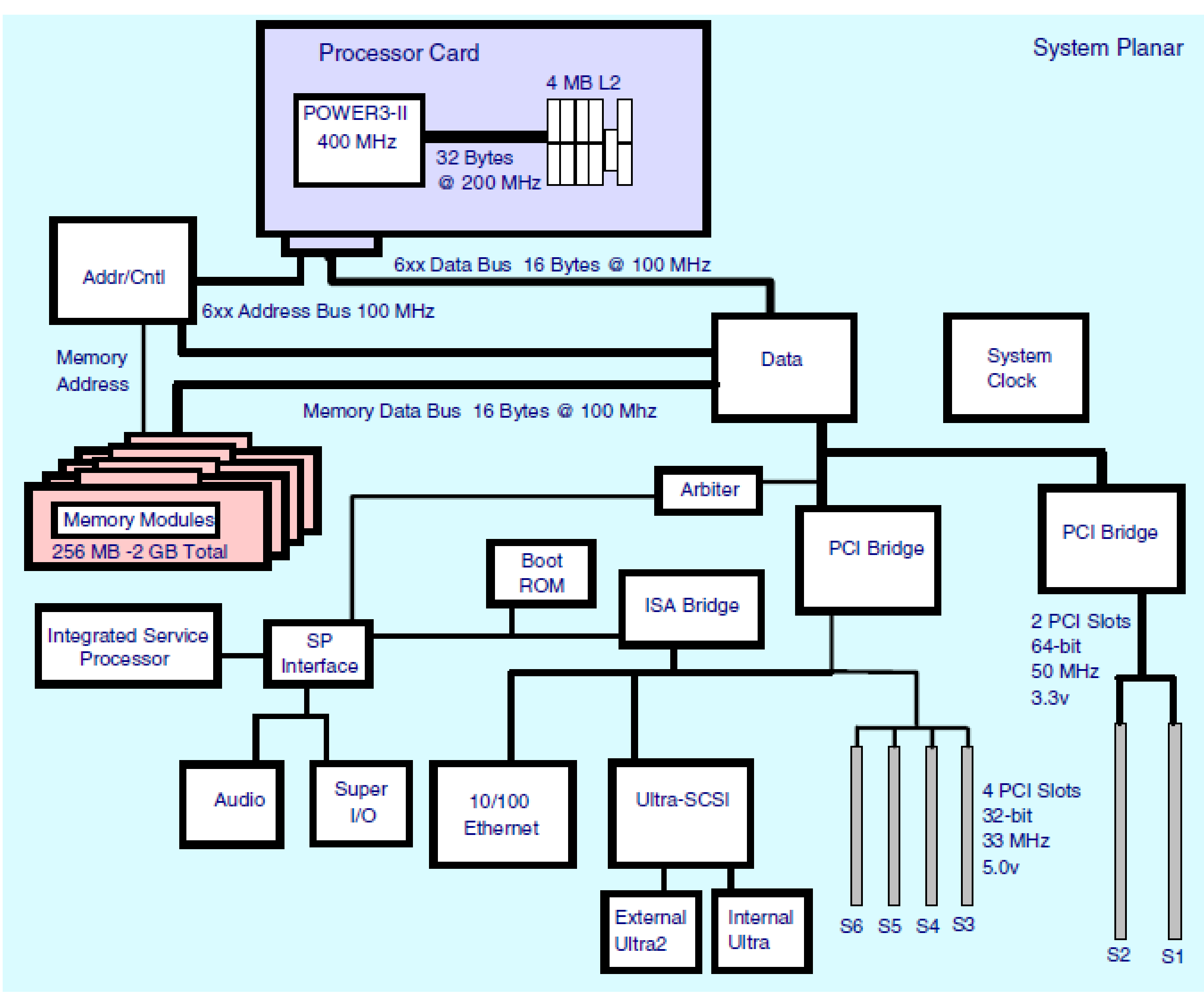
2.2.8 Linking memory via low pin count serial point-to-point buses and buffer chips (SMI chips) -1

The way of attaching memory in the previous POWER3-II based systems

- Previous POWER3-II based systems were clocked at 400 MHz or less.

  These systems were serviced by 168-pin SDRAM-100 memory via a 16-byte wide 100 Mbit/sec memory interface that is equivalent to two standard 8-byte wide SDRAM-100 memory channels, as indicated in the next Figure.

System architecture of a POWER3-II based system [10]



2.2.8 Attaching memory via low pin count serial point-to-point buses and buffer chips (SMI chips) -2

- By contrast, POWER4 based systems increased clock rate from 400 MHz to up to 1.3 GHz and introduced dual cores.
- By taking into account the roughly three times higher clock rate and dual cores POWER4 based systems can be expected to need about 6 times more memory bandwidth than POWER3-II based systems.
- Doubling the memory speed from 100 MT/s (POWER3-II) to 200 MT/s (POWER4) satisfies however only partly the increased bandwidth demand of POWER4.
- Thus a threefold deficit in bandwidth demand remains that in principle could be satisfied by tripling the number of standard parallel memory channels (from 2 to 6).

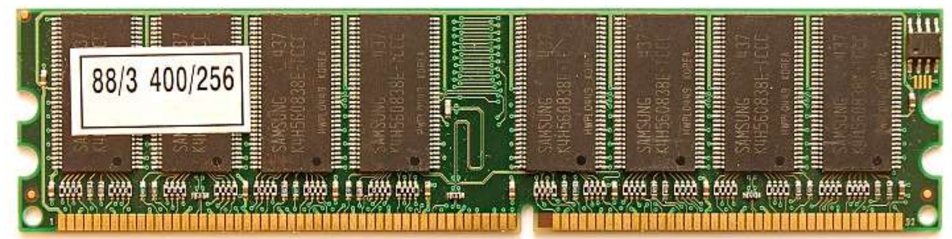
  Nevertheless, two standard DDR channels need already 2x184 copper lines between the memory controller and the memory DIMMs (see subsequent Figure) but placing three times more copper lines onto the same area is not more feasible due to electrical limitations (higher resistance of narrower copper lines and higher capacitive coupling of less line spacing).

SDRAM to DDR4 DIMMs

SDRAM (SDR)

168-pin

DDR

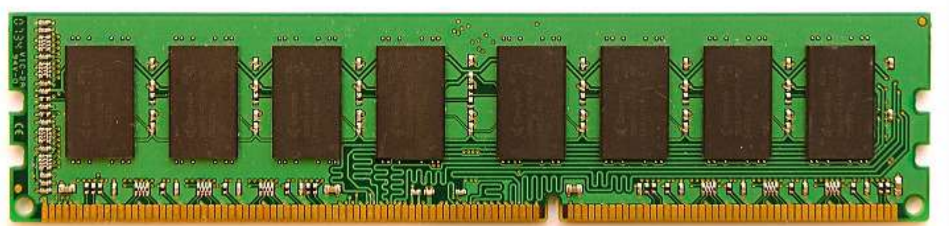


184-pin

DDR2

240- pin

DDR3



240-pin

DDR4

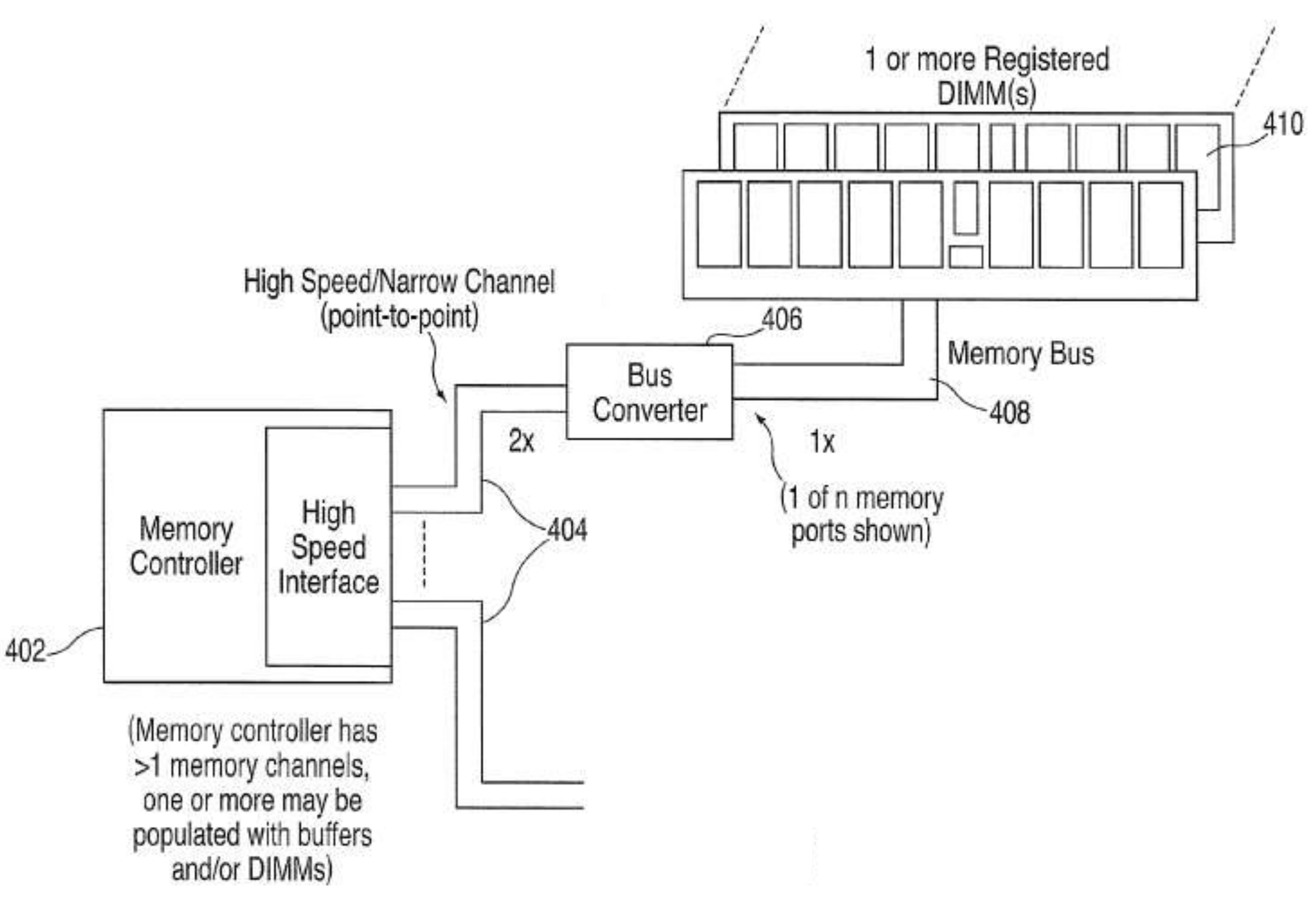
284-pin

All these DIMM modules are 64 bit (8-byte) wide

2.2.8 Attaching memory via low pin count serial point-to-point buses and buffer chips (SMI chips) -3

As a consequence, to implement a substantially higher number of memory channels on the POWER4 than on the previous POWER3-II IBM was forced to replace the large pin count (168 pins) standard parallel memory channels by low pin count (about 50 pins) serial point-to-point memory interconnection.

Principle of replacing standard parallel memory channels by low pin count serial point-to point channels as referred to as prior art in an IBM patent filed in 7/2004 [11]

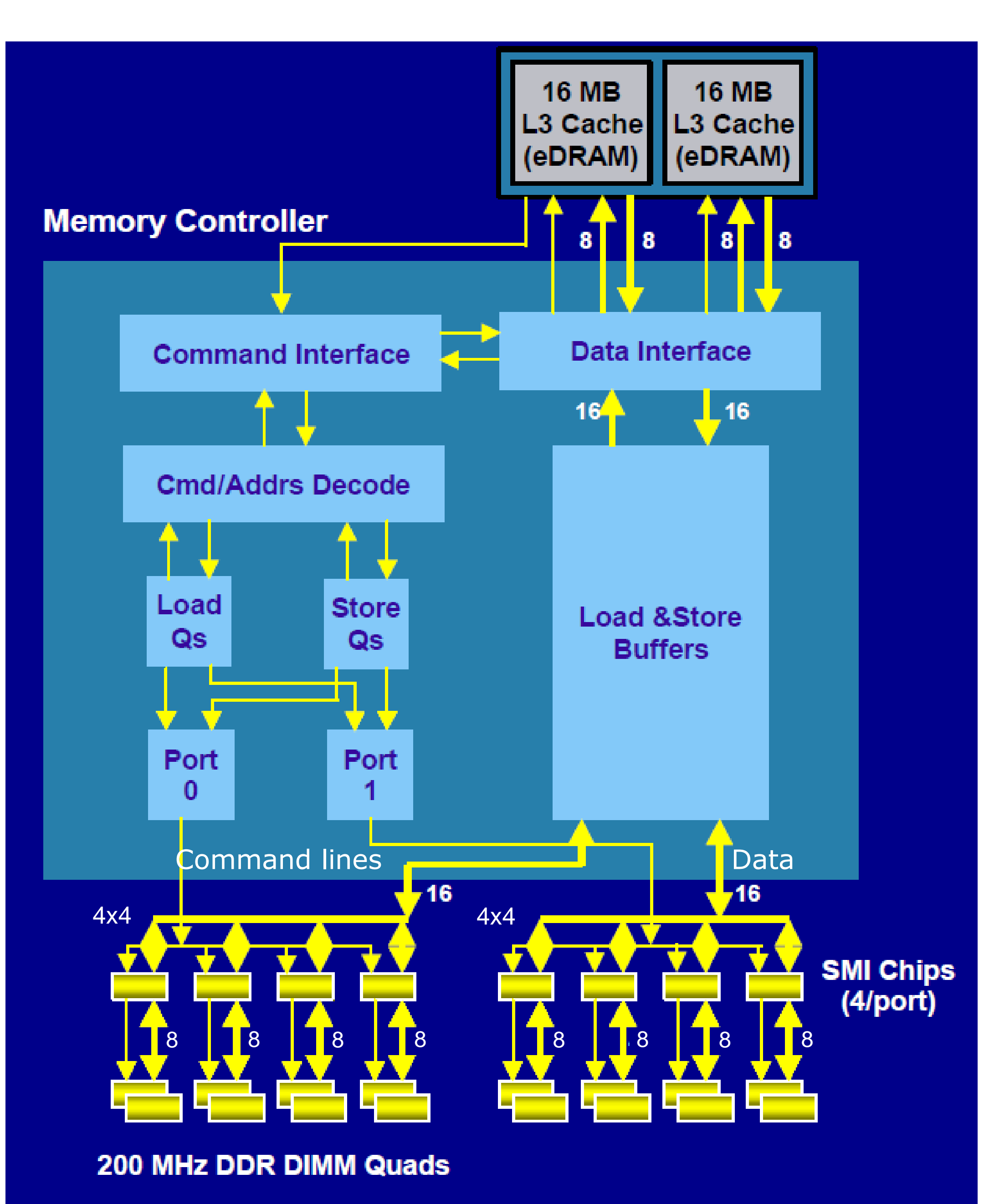


Layout of the memory subsystem of the POWER4 -1 [12]

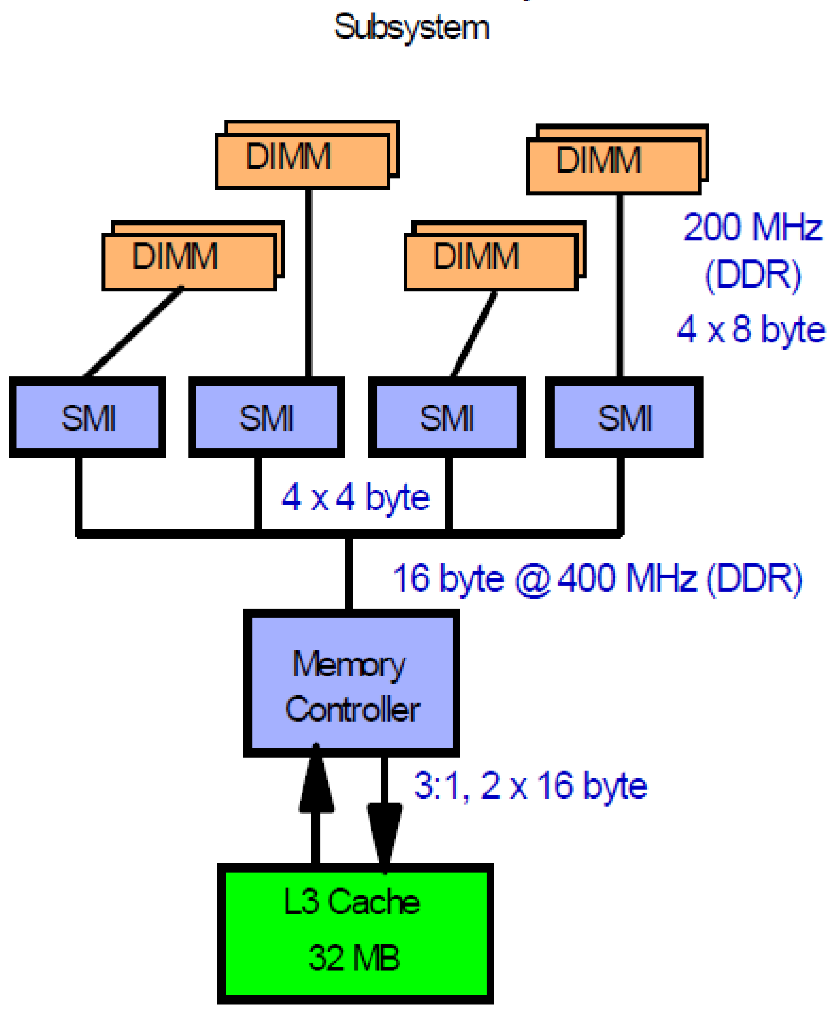
- The Figure shows two memory ports (Port 0 and Port 1).

  (Here we note that actual models implement one or two memory ports).
- Each of the ports provide four 4 byte wide serial high speed (400 MT/s) point-to-point low pin count buses to the SMI chips.

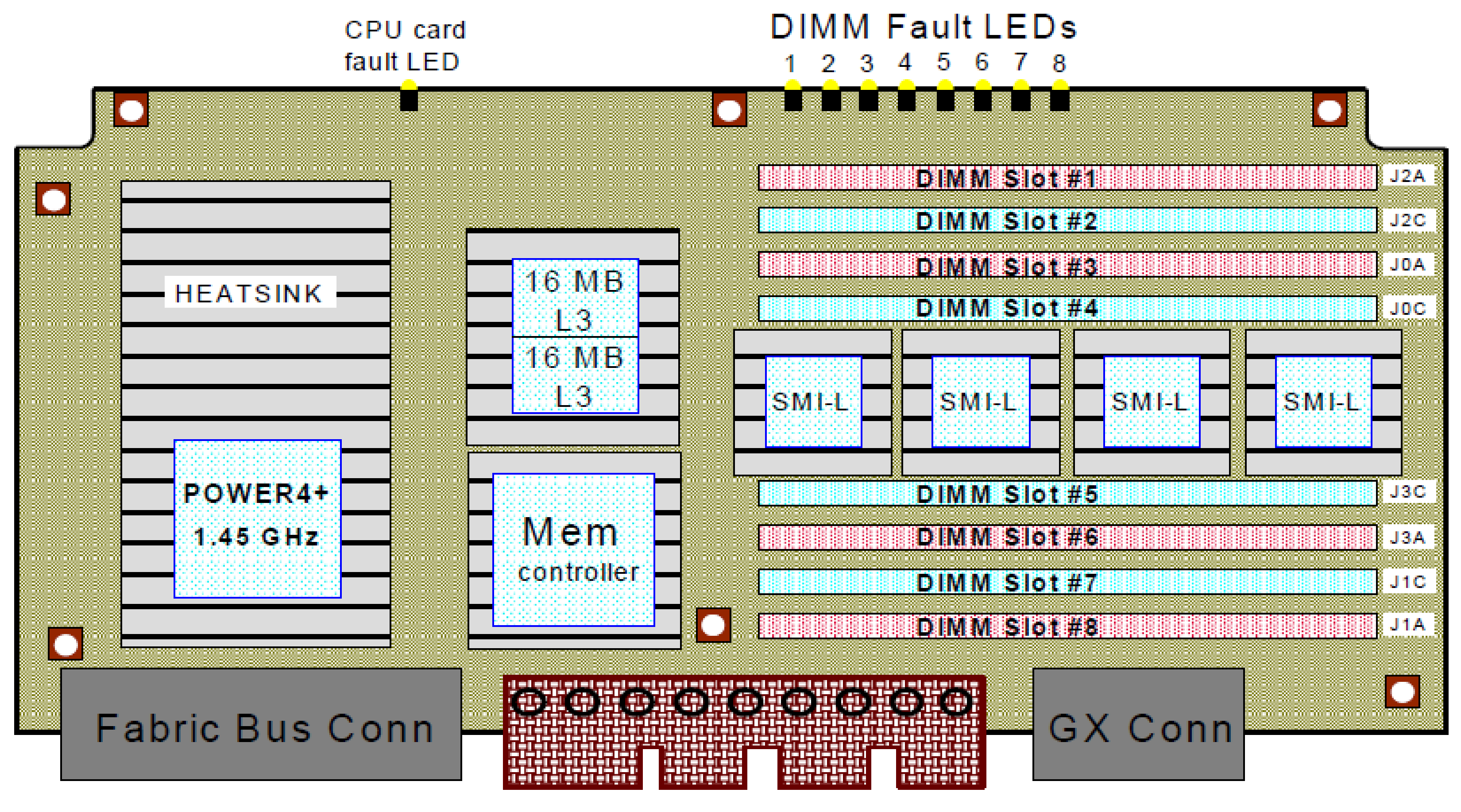
  SMI means Synchronous Memory Interface or System Memory Interface).
- Each SMI buffer chip serves two 8 byte wide DDR-200 DIMM channels with up to two DIMMs.



Implementation example: Conceptual view of the memory subsystem of the p650 server using a single port with 4 SMI chips [13]



Implementation example: Processor card of the p650 server implementing a one port memory subsystem with 4 SMI chips [13]



2.2.8 Connecting memory via low pin count serial P2P buses and buffer chips (10)

DIMMs used in the memory subsystem of the POWER4 [14]

- The DIMMs used in low and midrange POWER4 systems (such as the Power p630/p650 are IBM proprietary with typical sizes of 512 to 2048 MB and speed grades of up to DDR-200.
- They have 208 pins rather than 184 pins as industry standard DDR DIMMs.

Figure: IBM proprietary 208-pin DIMMs (FC 4452 4x512 MB) [14]

2.2.8 Connecting memory via low pin count serial P2P buses and buffer chips (11)

Remark

In the high-end p690 model the memory modules are soldered on the memory card for improved reliability [15].

Per socket bandwidth of the memory subsystem of the POWER4

- The maximum per socket memory bandwidth of the memory subsystems is limited by both the serial buses connecting the SMIs and the available DIMMs.
- The serial buses limited memory bandwidth of a 2 port memory configuration with DDR-200 memory DIMMs is:

  2 ports x 4 SMI buses x 4 B width x 2 x 200 MT/s = 12.8 GB/s

- The memory DIMMs limited memory bandwidth of a 2 port memory configuration with DDR2-200 memory DIMMs is:

  2 ports x 4 SMI chips x 8 B x 200 MT/s = 12.8 GB/s

- As the above calculations show, in the POWER4 processor both maximum per socket bandwidth limits are the same equaling 12.8 GB/s.
- By contrast foregoing POWER3-II based systems provided only

  2 x 8 x 0.1 = 1.6 GB/s.

- This means that the new memory subsystem of the POWER4 provides significantly (8 times) higher bandwidth than the previous POWER3-II system satisfying POWER4's vastly increased bandwidth demand.
- The peculiarity of IBM's solution is that IBM does not make use of standard 184-pin DDR DIMMs but employs proprietary 208-pin DDR-200 DIMMs.

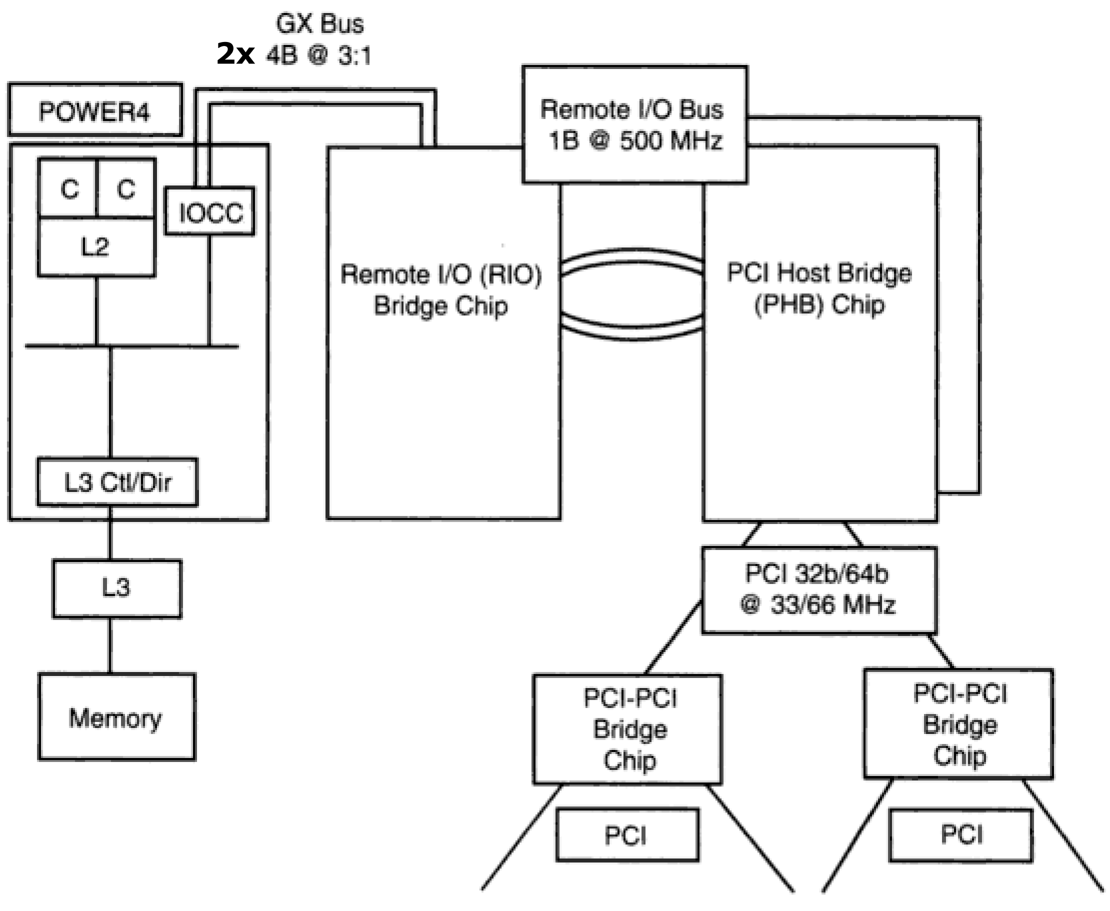
2.2.9 Introduction of the GX bus [13], [16]

- Along with the POWER4 IBM introduced also a new I/O bus, called the GX bus.
- The GX bus is a derivative of the 6XX bus of the PowerPC 620 (1997) processor, line, where 6 has been replaced with a G for Gigaprocessor, and one of the x's has been dropped [a].
- The GX bus is a high-frequency, single-ended, unidirectional, 4-byte wide point to-point bus.

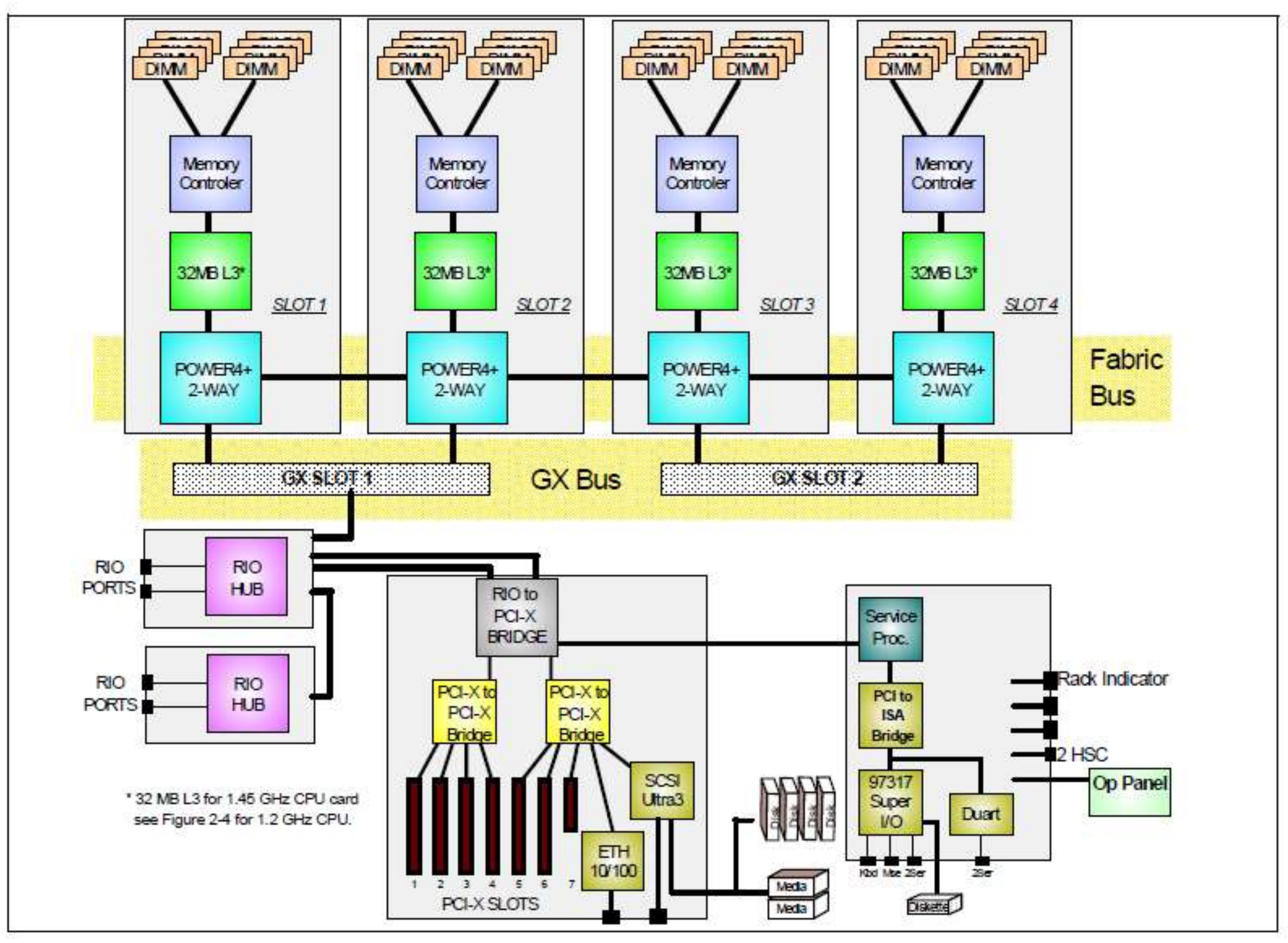
  Both data and address information are multiplexed onto the bus.

  The original GX bus of the POWER4 runs at 1/3 of the clock frequency, e.g. at 400 MHz (for a 1.2 GHz processor) to give an aggregate data rate of 3.2 GB/s in this case.
- The GX bus connects (through the GX slot) the processor card with the remote I/O Bridge chip (called RIO chip) that cares for attaching I/O, as shown in the next Figures.

Logical view of the I/O subsystem of the POWER4 [17]



Example I/O system: Block diagram of the POWER4+ based p650 server [13



Evolution of the GX bus in IBM's POWER line

Based on available IBM literature, the Table below illustrates the evolution of the GX bus.

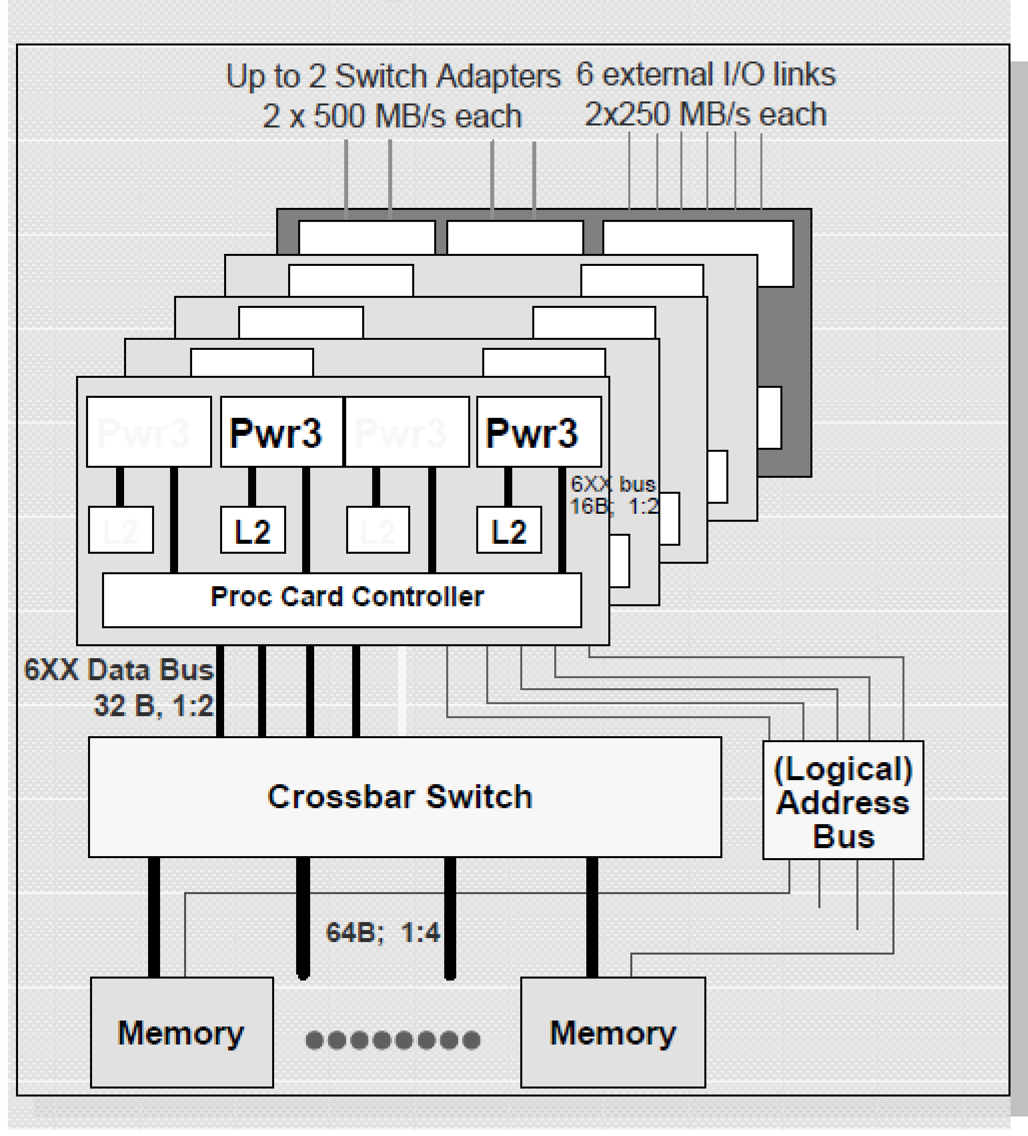
Model	Designation of the GX bus	GX speed	Exemplary total bandwidth per GX bus
POWER4	GX	1/3 fc e.g. 400 MHz for fc=1.2 GHz	3.2 GB/s
POWER5	GX+	1/3 fc e.g. 700 MHz for fc= 2.1 GHz	5.6 GB/s
POWER6	GX++	¼ fc e.g. 1.05 GHz for fc=4.2 GHz	8.4 GB/s
POWER7	GX+/GX++	1.25 GHz 2.50 GHz	10 GB/s/ 20 GB/s
POWER8	--	--	--

Table: Evolution of the features of the GX bus in IBM's POWER line

2.2.10 Vastly improved support for SMP

- Symmetrical Multiprocessor (SMP) configurations were already supported by the POWER3 processor within the POWER line.
- Nevertheless, with the POWER4 design IBM vastly improved and expanded the SMP capability of the POWER line (e.g. from 8-way to 32-way SMP), as indicated in the subsequent Figures.

Example of supporting SMP in POWER3 based systems [18]

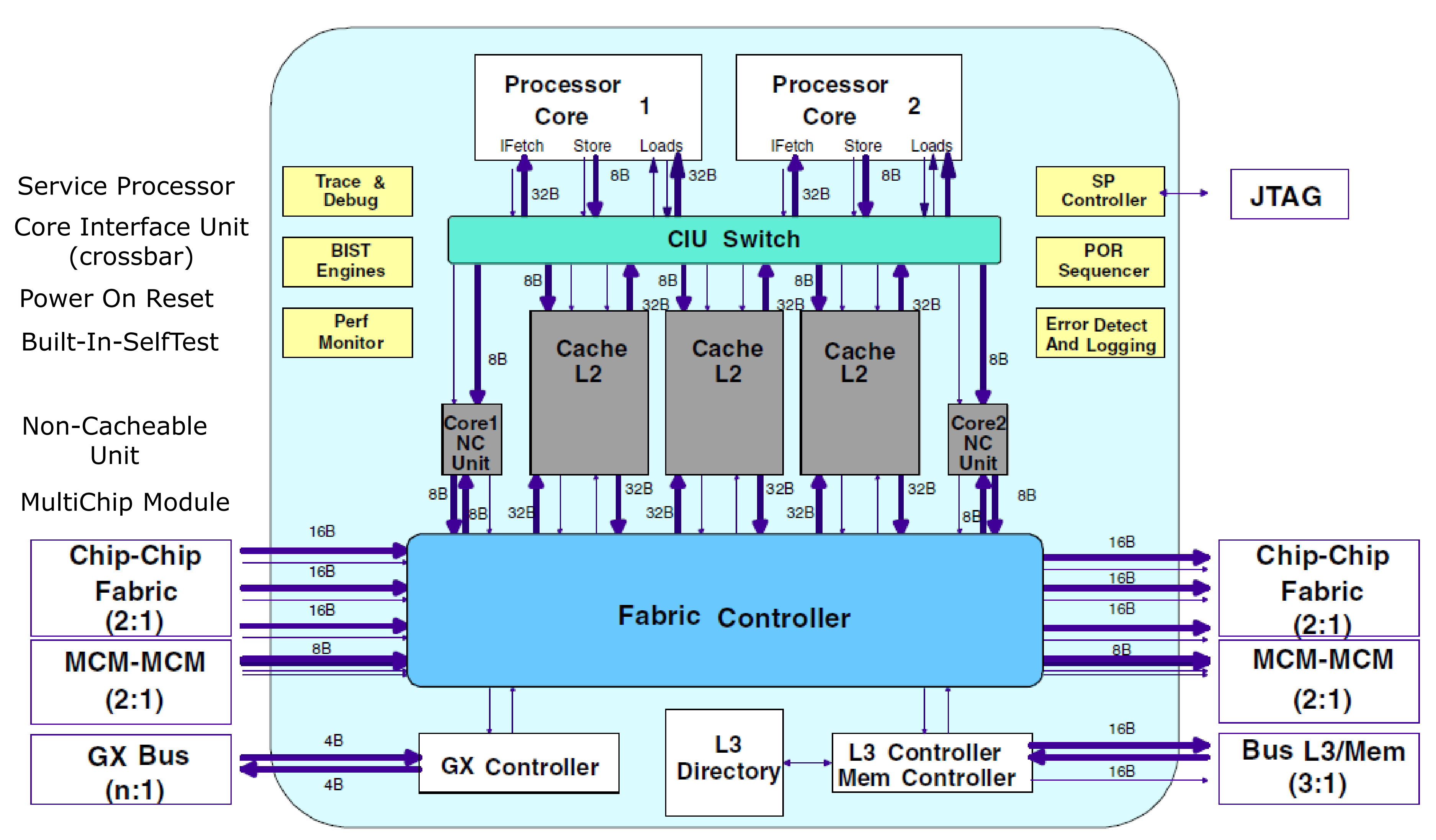


Nighthawk-1

- Up to 8-way SMP with POWER3
- 4 MB private L2/processor
- 1 GB to 16 GB memory
- 6 RIO (remote I/O) ports
- Crossbar data switch for BW
- Point-to-point buses
- Prefetch logic
- New SP Switch adapters directly into Node Crossbar
  - Up to 4 switch links at 500x2 MB/s each
- Or use current switch/adapter
- 7.1 GFlop/s peak

2.2.10 Vastly improved support for SMP (3)

Links provided by the POWER4 design to support SMP configurations -1 [19]



2.2.10 Vastly improved support for SMP (4)

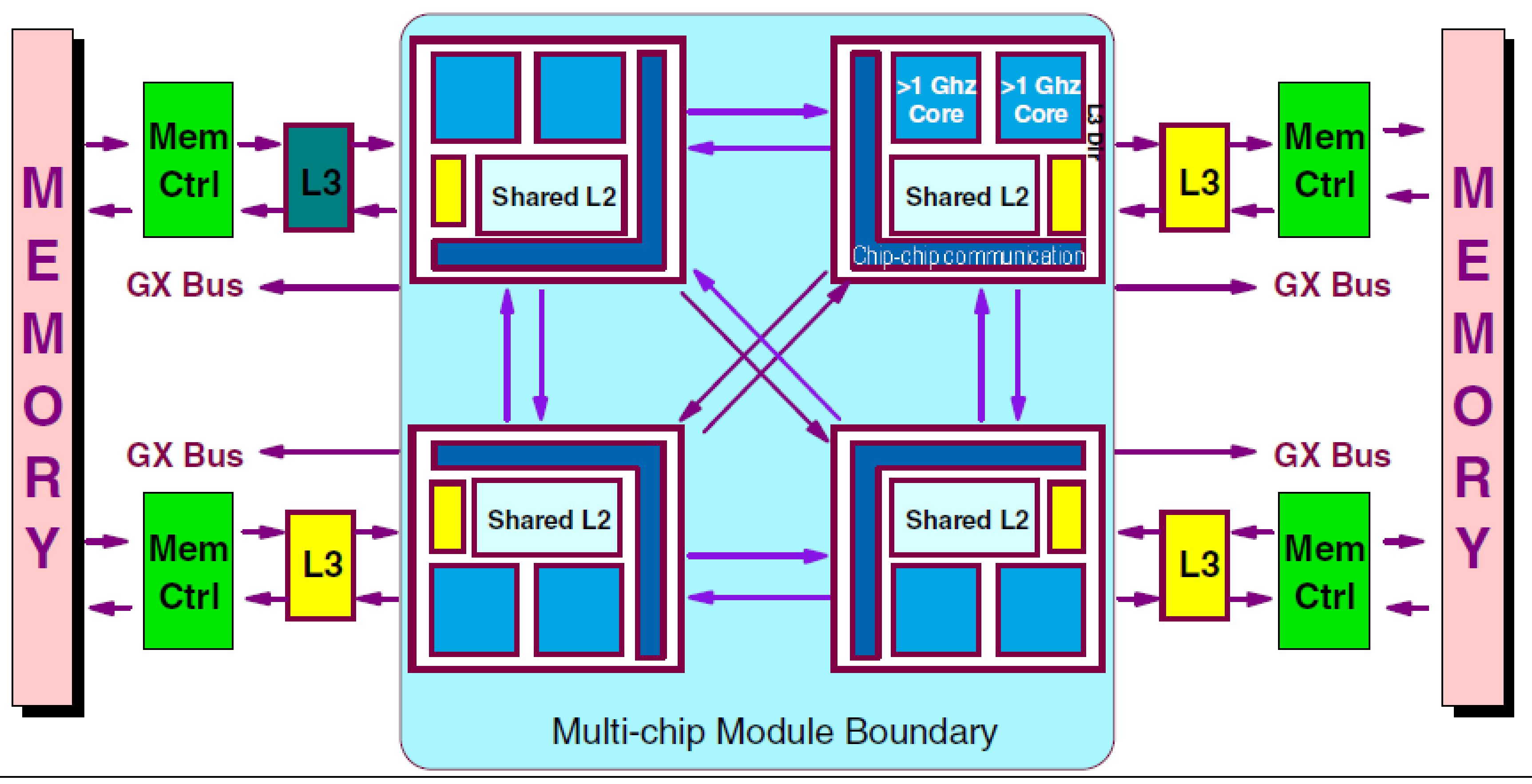
Links provided by the POWER4 design to support SMP configurations -2 [19]

The 16 Byte wide Chip-to-Chip links interconnect chips within a multi chip (MCM) module, allowing to implement 8-way SMP systems, as indicated next, whereas the 8 Byte wide MCM-to-MCM links are provided to interconnect up to four MCM modules, to implement up to 32-way SMP systems, as shown subsequently.

2.3 POWER4 based SMP configurations

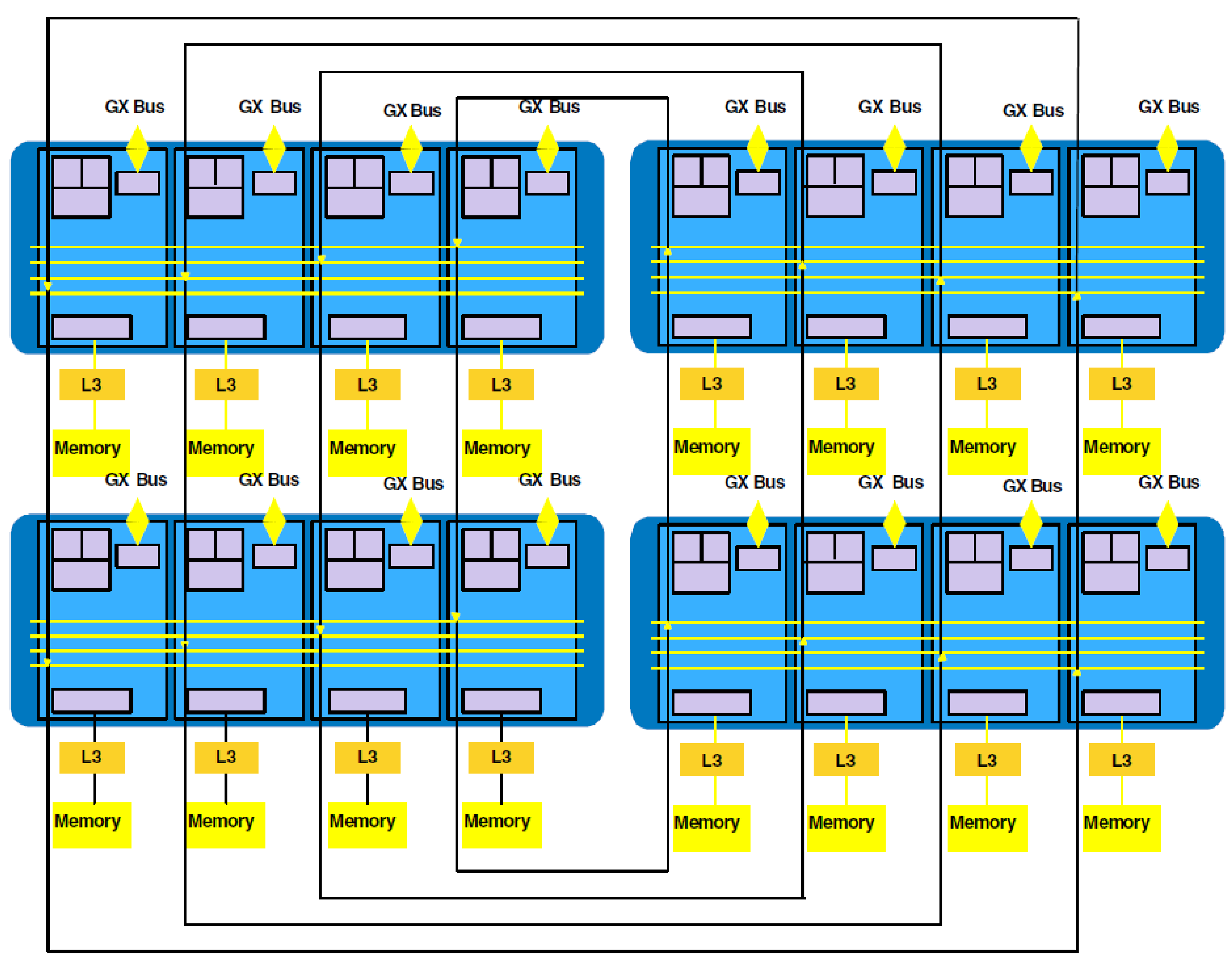
2.3 POWER4 based SMP configurations

POWER4 based 4-processor/8-core Multi-Chip Module (MCM) [19]



2.3 POWER4 based SMP configurations (2)

POWER4 based 16-processor/32-core SMP by interconnecting 4 MCMs [19]



Pictures of POWER4 based MCM and 16-processor/32-core SMP modules

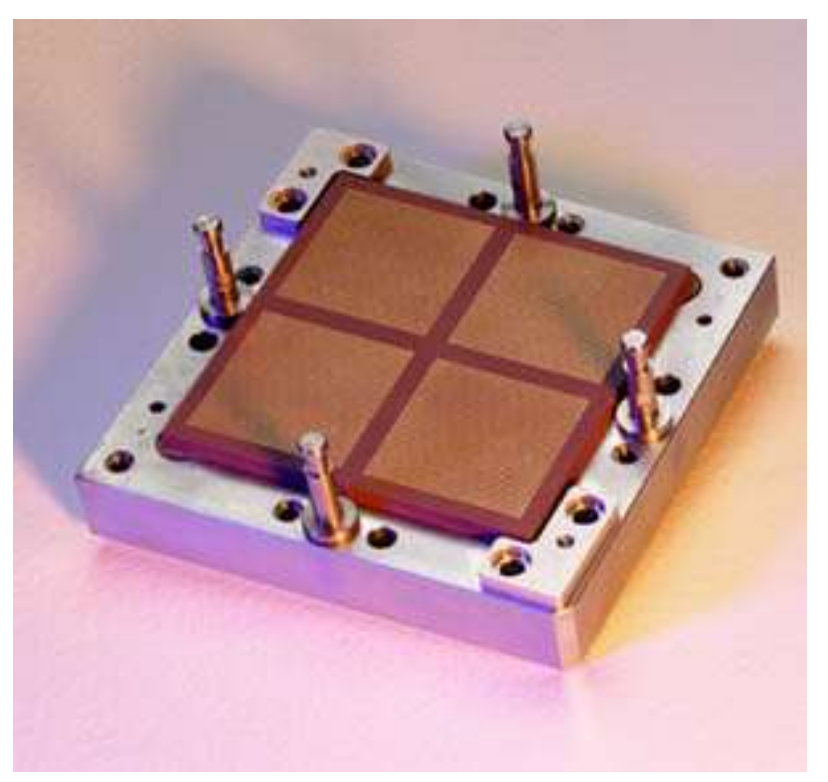


Figure: POWER4 based MCM, without L3 and memory controller chips [20]

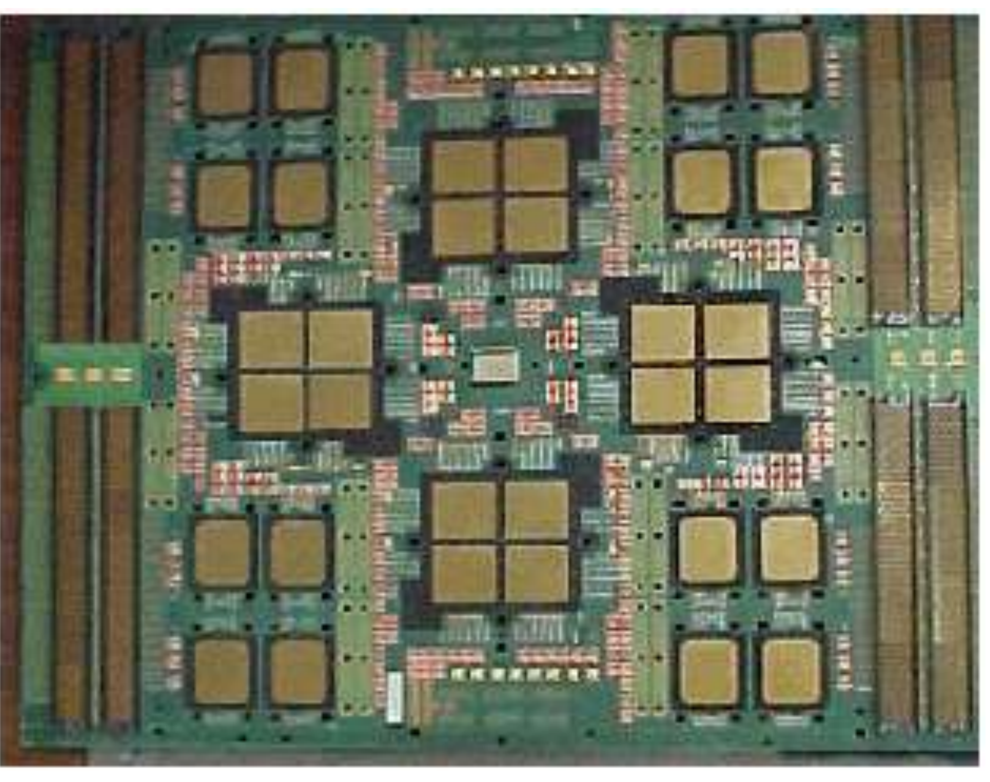


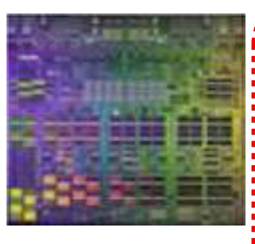
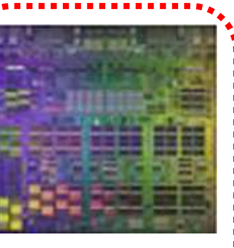
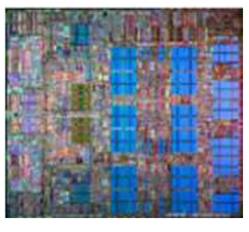
Figure: POWER4 based 32-way SMP built up of four MCMs [20]
(Only the memory controller chips are visible on this side)

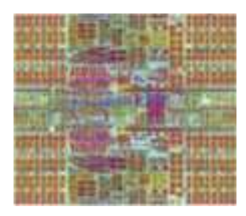
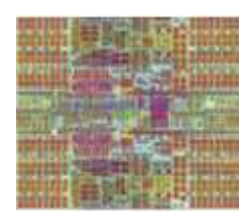
3. POWER4+

3. POWER4+

- Introduced 11/2002
- Main improvements vs. the POWER4 are
  - smaller feature size (130 nm vs. 180 nm) and consequently
  - higher clock rate of up to (1.45 GHz (at introduction) vs. 1.3 GHz) and
  - less power consumption (70 W at 1.2 GHz vs. 115 W).
- No new innovations were introduced vs. the POWER4.

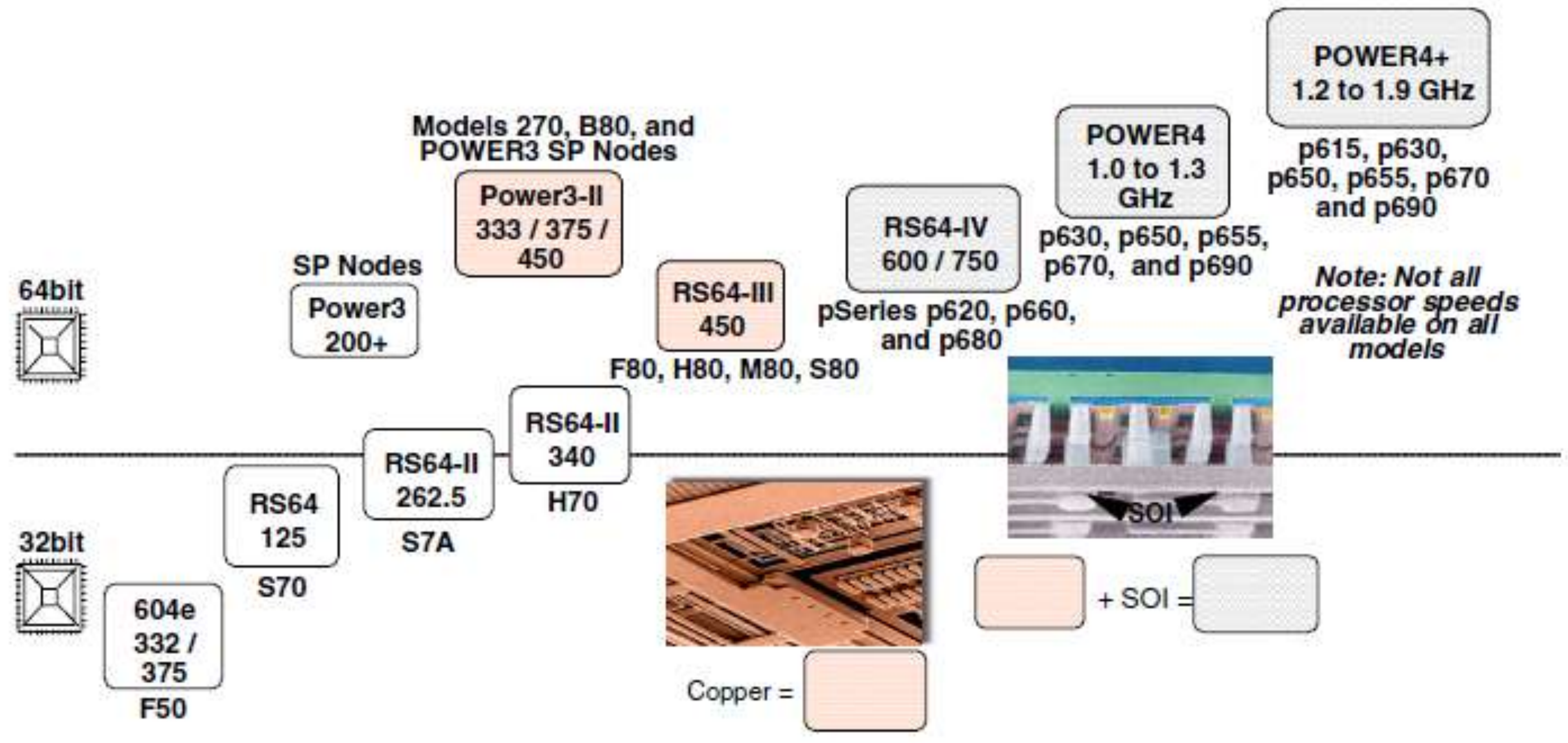
Key features of the POWER4+

	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

Overview of IBM's server models up to the POWER4+ [7]



4. POWER5

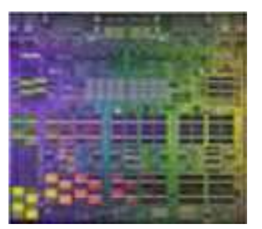
- 4.1 Introduction to the POWER5
- 4.2 Main enhancements of the POWER5 vs. the POWER4

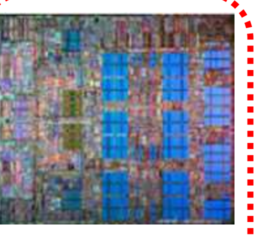
4.1 Introduction to the POWER5

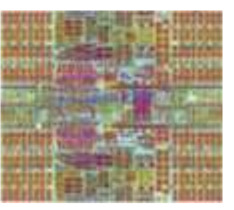
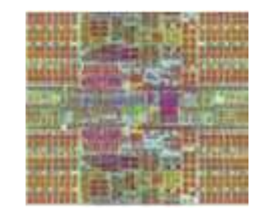
POWER5

- Introduced in 5/2004
- 130 nm technology
- 276 million transistors on 389 mm^2

Key features of the POWER5



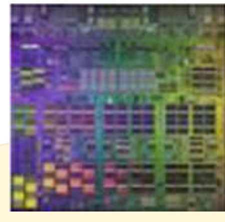


	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

4.1 Introduction to the POWER5 (3)

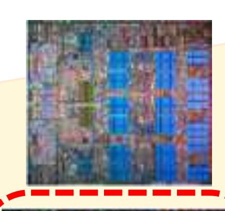
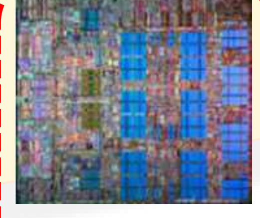
Key innovations of the POWER5 (Die photos from [3])



Power4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP

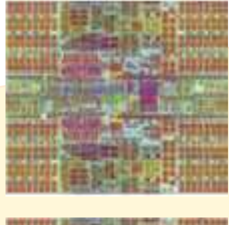
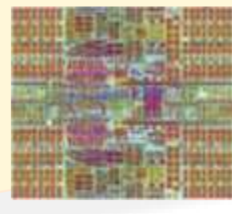
2001

Power5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating

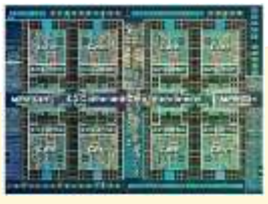
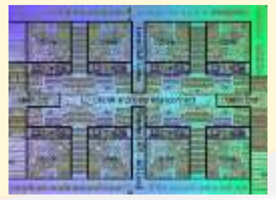
2004

Power6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

2007

Power7/7+*
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

*POWER7+

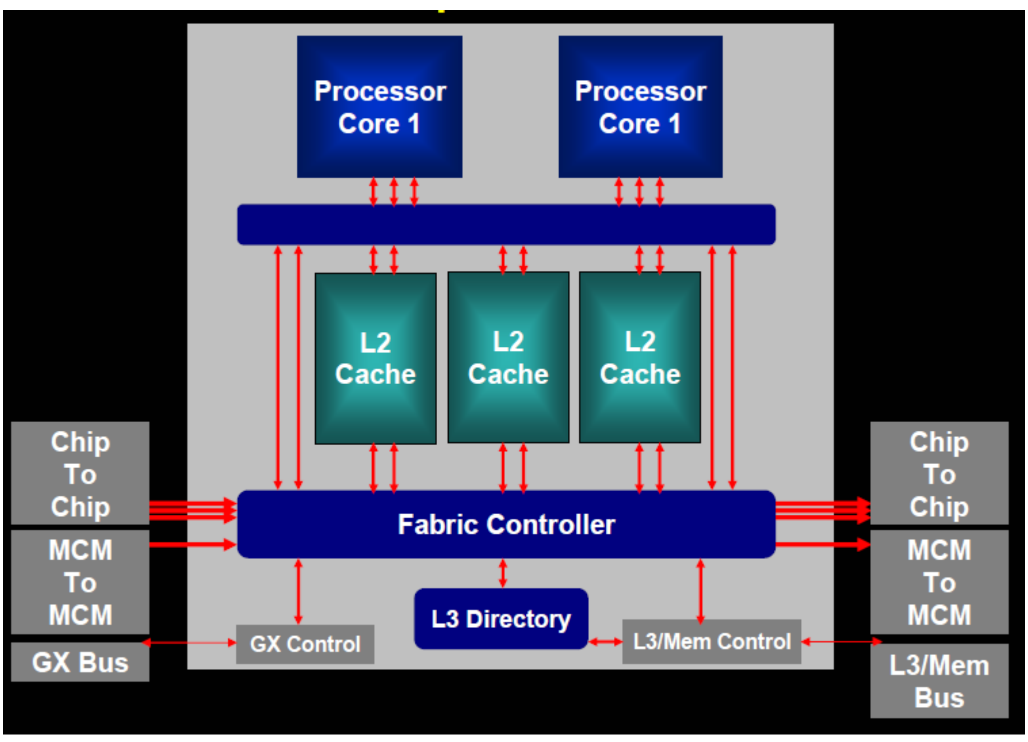
2010

Power8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip µc for PM
Per-core Vdd
Per-core VRMs

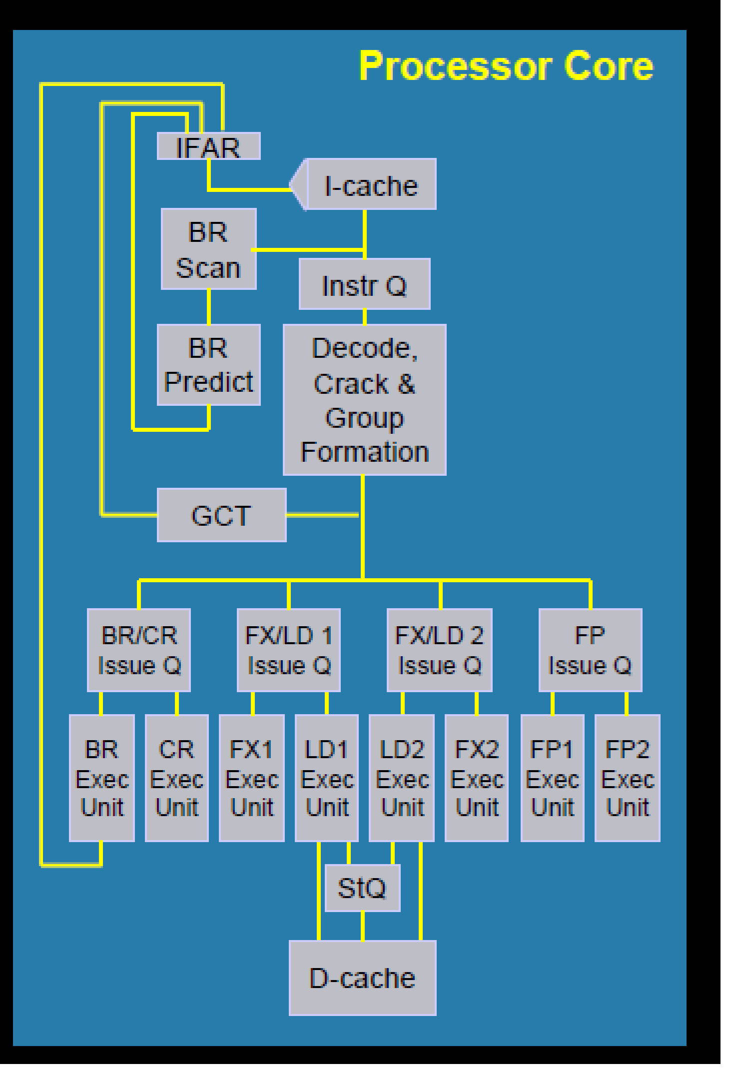
2014

High level block diagram of the POWER5 [21]



Block diagram of a POWER5 cores [21]

- Speculative superscalar organization
  - Out-of-Order execution
  - Large rename pools
  - 8 instruction issue, 5 instruction complete
  - Large instruction window for scheduling
- 8 Execution pipelines
  - 2 load / store units
  - 2 fixed point units
  - 2 DP multiply-add execution units
  - 1 branch resolution unit
  - 1 CR execution unit
- Aggressive branch prediction
  - Target address and outcome prediction
  - Static prediction / branch hints used
  - Fast, selective flush on branch mispredict

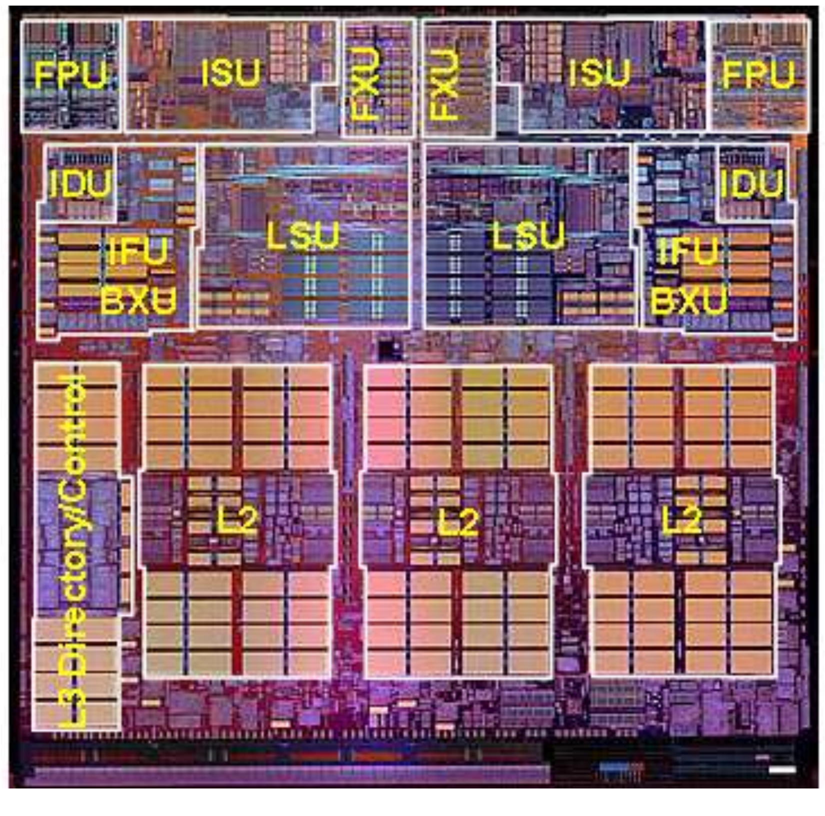


Execution resources of the POWER5 cores

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-ways
Width of the front-end	5	5	5	6	8	12
Dispatch rate	5	5	(In-order design)	6	8	12
Issue rate	8	8	7	8	10	16
No. of execution units per-core	8	8	9	12	16	20
No/type of execution units per-core	2 FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP 1BR/CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 2VMX, 1DFU, 2LU, 1 Crypto	8AGEN, 4VSU(128) 4LS(128) 2BRU DFU Cripto

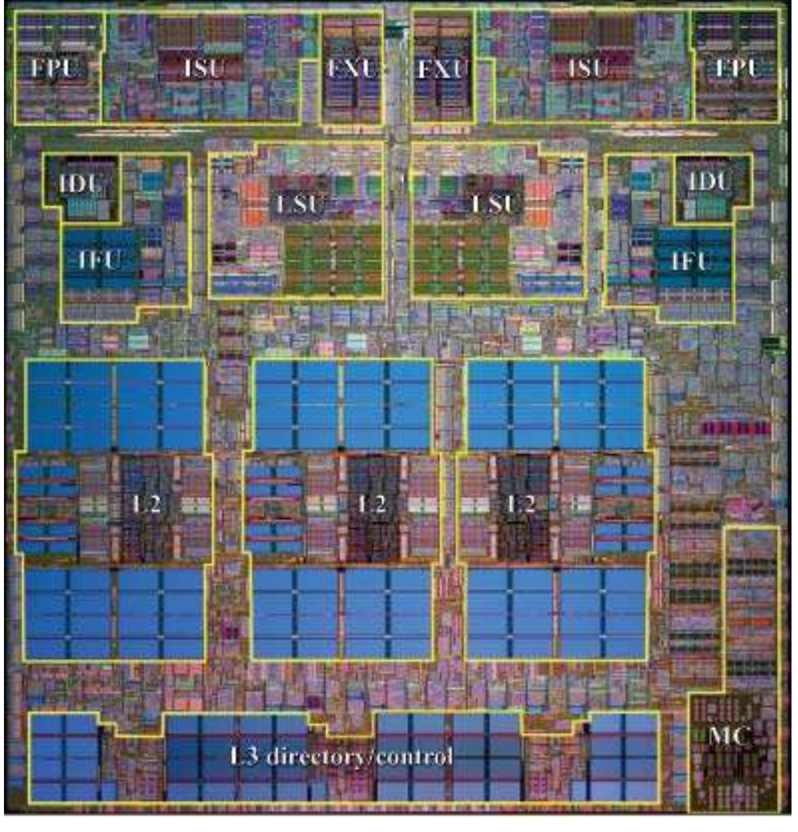
Contrasting the floor plans of the POWER4 and POWER5 dies

POWER4 [22]



180 nm, 174 mtrs, 412 mm^2

POWER5 [23]



130 nm, 276 mtrs, 389 mm^2 (enlarged)

Overview of the POWER5 and POWER5+ models [2]

Model	Processors	Clock Rate (GHz)	Max Memory (x 2^30 byte)
p5 595	16-64	1.65, 1.9	2000
p5 590	8-32	1.65	1000
p5 575	8-16	1.9, 2.2*	256
p5 570	2-16	1.9, 2.2*	512
p5 560Q	4-16	1.5*	128
p5 520	1,2	1.65, 1.9*	32
p5 505	1,2	1.5, 1.65*	32

* POWER5+

4.2 Main enhancements of the POWER5 vs. the POWER4

- 4.2.1 Modifications related to the L3 cache
- 4.2.2 Redesigned memory subsystem
- 4.2.3 Enhanced modularity for building SMPs
- 4.2.4 Modified interconnection links

4.2.1 Modifications related to the L3 cache

There are a couple of modifications concerning the L3 cache as follows.

a) Moving L3 from the memory side to the processor side of the fabric

b) Modifying L3's cache architecture from inclusive to exclusive

c) Enlarging the L3 cache from 32 MB to 36 MB

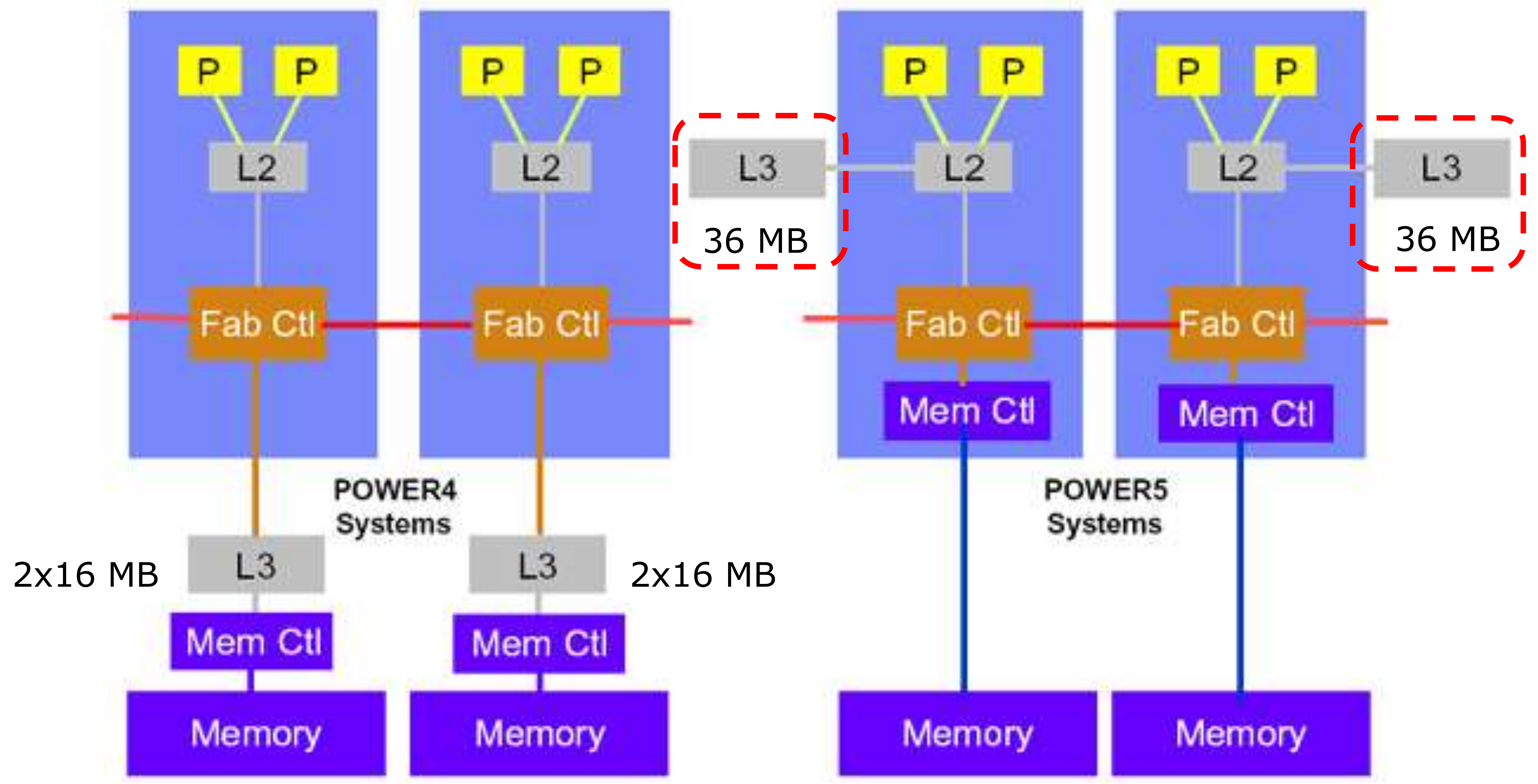
d) Implementing the L3 cache on a single external chip rather than on two chips, as in the previous POWER4.

4.2.1 Modifications related to the L3 cache (2)

a) Moving L3 from the memory side to the processor side of the fabric-1 [20]

- In POWER5 systems the L3 cache has been moved from the memory side to the processor side of the fabric, as shown in the next Figure.
- The L3 cache is interconnected with the processor chip via separate 16-byte-wide buses for reads and writes operating at half processor speed.

Moving L3 from the memory side to the processor side of the fabric-2 [20]



The new arrangement eliminates L3 hit traffic from the internal chip-to-chip bus.

4.2.1 Modifications related to the L3 cache (4)

b) Modifying L3's cache architecture from inclusive to exclusive [23]

In the POWER5 the L3 cache became an exclusive cache (victim cache) related to the L2 cache.

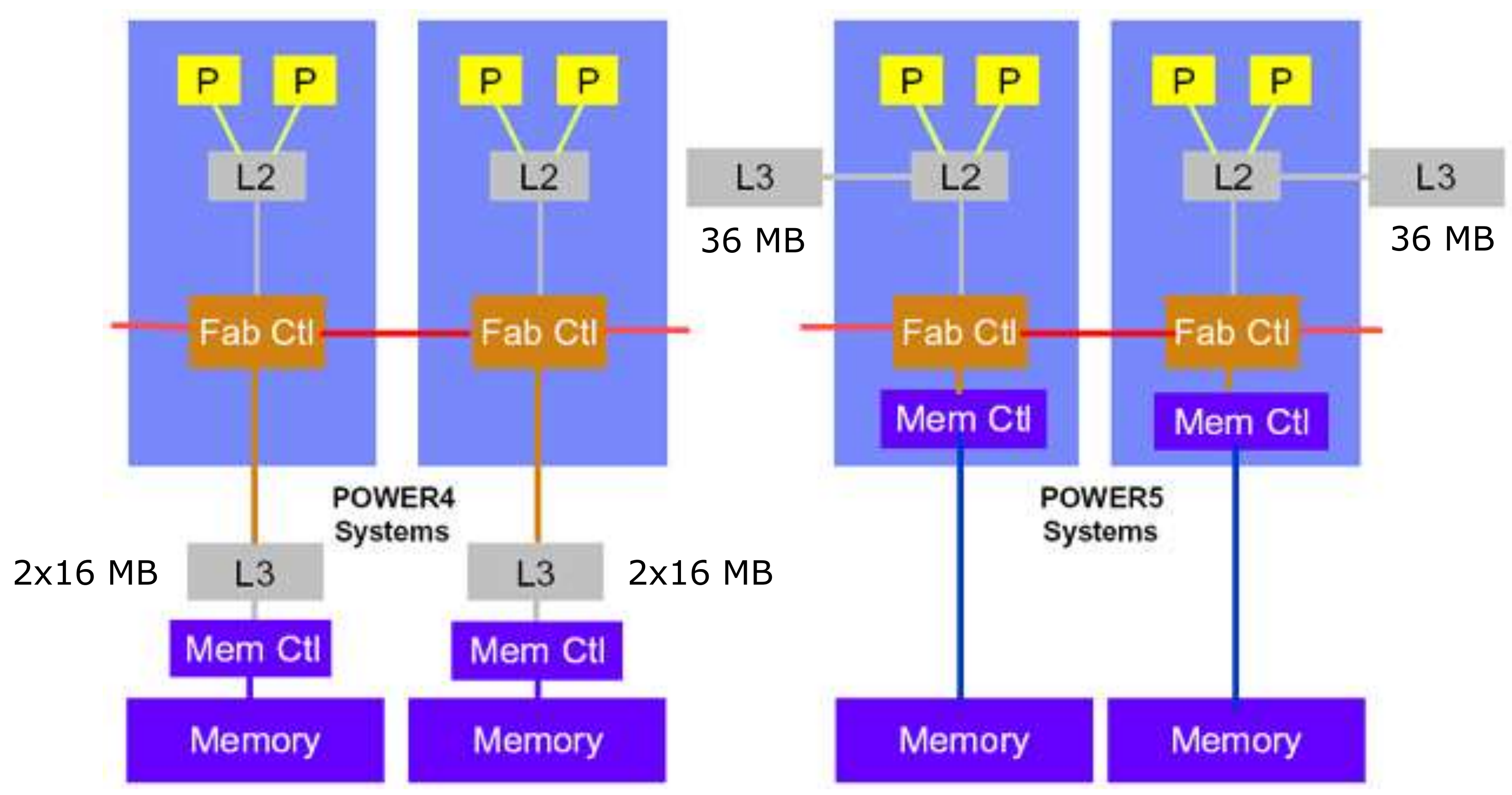
The associated principle of operation is as follows:

- When a cache line is to be written into the L2 cache but there is no more free line in it a line has to be evicted from the L2 cache according to a chosen replacement policy to free cache space.

  The evicted (victim) line is written then into the L3 cache.
- A reference to data kept in the L3 cache cause reloading the referenced cache line into the L2 cache.
- When a cache line needs to be evicted from the L3 cache and this line is in the modified state, it will be written back to the memory.

  An unmodified victim line on the other hand, is replaced simply by the new line in the L3 cache.

c) Enlarging the L3 cache from 32 MB to 36 MB [20]



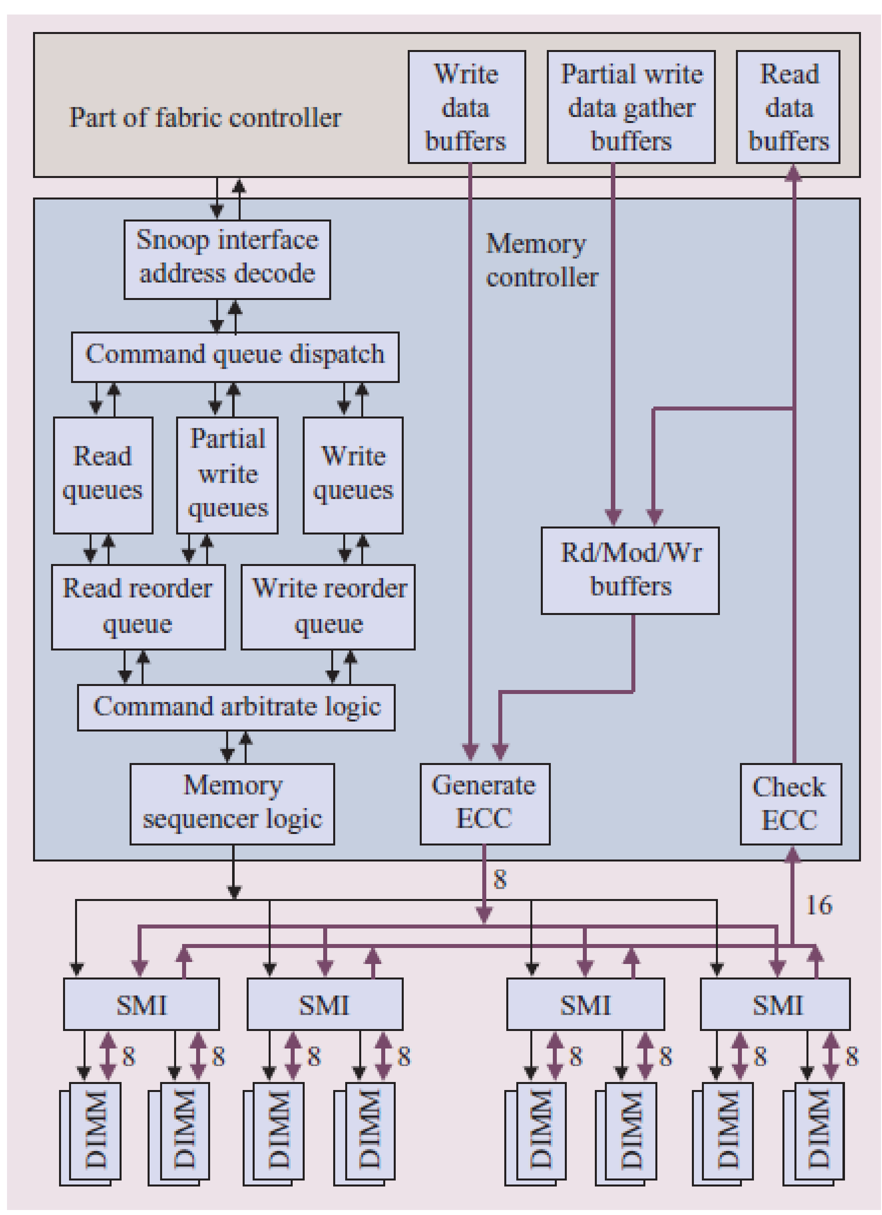
4.2.1 Modifications related to the L3 cache (6)

d) Implementing the L3 cache on a single external chip rather than on two chips, as in the previous POWER4.

(Not indicated on the previous Figure).

4.2.2 Redesigned memory subsystem [23]

- Model dependent there are one or two memory ports in the POWER5.
- Each port is connected to two buffer chips called SMI chip.
- Three point-to-point serial buses interconnect each SMI chip with one port of the Memory Controller;
  - an address/command bus,
  - a unidirectional 2-byte wide write data bus and
  - a unidirectional 4 or 8-byte wide read data bus (4-wide in two port configurations and 8-wide in single port configurations).
- All three buses operate at twice the DRAM speed.
- The DRAM chips used are DDR-266 or DDR2-533 chips mounted on IBM proprietary DIMMs with 208 pins.



4.2.2 Redesigned memory subsystem (2)

Remarks

In one port configurations with two SMI chips per POWER5 chip

- four bytes of the 8-byte write data bus remain unconnected and each of the two SMI chips is connected to the Memory controller by a two byte wide point-to-point write data bus, whereas
- all 16 read data bus bytes are used such that each of the two SMI chips is connected to the Memory Controller via an eight byte wide point-to-point read data bus.

4.2.2 Redesigned memory subsystem (3)

High level comparison of the memory subsystems of POWER4 and POWER5 [12], [23] -1

16 MB L3 Cache (eDRAM)
16 MB L3 Cache (eDRAM)
Memory Controller
8 8 8 8
Command Interface
Data Interface
16 16
Cmd/Addrs Decode
Load Qs
Store Qs
Load &Store Buffers
Port 0
Port 1
16 16
4x4
4x4
8 8 8 8 8 8 8 8
SMI Chips (4/port)
DDR--200 DIMMs

Part of fabric controller
Write data buffers
Partial write data gather buffers
Read data buffers
Snoop interface address decode
Memory controller
Command queue dispatch
Read queues
Partial write queues
Write queues
Rd/Mod/Wr buffers
Read reorder queue
Write reorder queue
Command arbitrate logic
Memory sequencer logic
Generate ECC
Check ECC
8 (4x2)
(4x4) 16
SMI SMI SMI SMI
8 8 8 8 8 8 8 8
DIMM DIMM DIMM DIMM DIMM DIMM DIMM DIMM
DDR-266 or 533 DIMMs

4.2.2 Redesigned memory subsystem (4)

High level comparison of the memory subsystems of POWER4 and POWER5 [12], [23] -2

A comparison (neglecting the internal architecture of the memory controllers) shows that

- the POWER5 has already an integrated memory controller whereas the POWER4 is using still an off-chip memory controller, as discussed in Section 4.3.2.
- Both the POWER 4 and the POWER5 can use model dependent one or two memory ports on the memory controller.
- Each port of the memory controller of the POWER5 serves two SMI chips, whereas each port of the POWER4 can support up to four SMI chips.
- The POWER5 has in the two port configuration 4 byte wide unidirectional read and 2-byte wide unidirectional write buses to the SMI chips whereas the POWER4 has four 4-byte wide bidirectional buses to the four SMI chips.
- In the POWER5 each SMI chip provides two 8-byte bus for the DIMMs whereas POWER4 SMI chips provide only a single 8-byte wide bus to the DIMMs.
- The buses operate in both processors at twice the DIMM speed.
- The POWER5 can use higher speed DIMMs than the POWER4 (DDR-266 or DDR2-533 DIMMs vs. DDR-200 DIMMs).

Example 1: A one port POWER5 configuration (e-server p5 550) [7]

- The referenced server has a single port memory with two SMI buffers (see below).

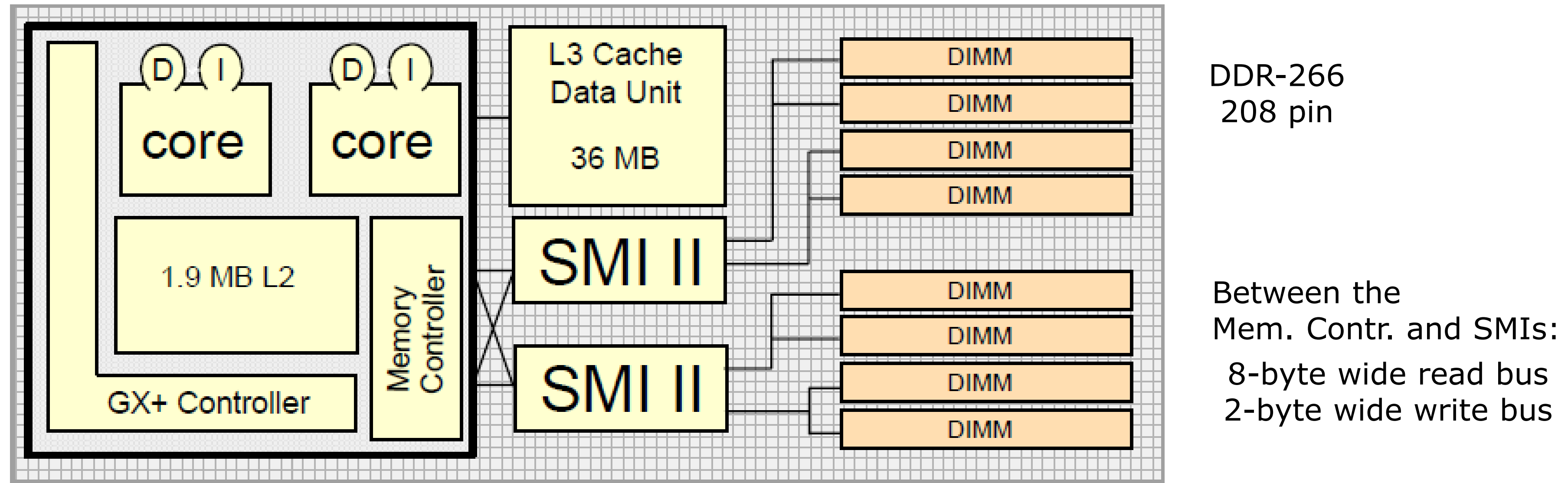


Figure: Attaching memory in the POWER5 based e-server p5 550 [7]

- The point-to-point read buses between the SMI chips and the Memory Controller are 8-byte wide whereas the point-to-point write buses between the SMI chips and the Memory Controller are 2-byte wide.
- Both buses operate at twice the speed of the DIMMs.
- The 208 pin DDR-266 DIMMs are IBM proprietary and have ½ to 16 GB capacity.

Example 2: A dual port POWER5 configuration (e-server p5 590) [24] -1

The referenced 4-processor server has 2 memory ports with 4 SMI buffers (see below)

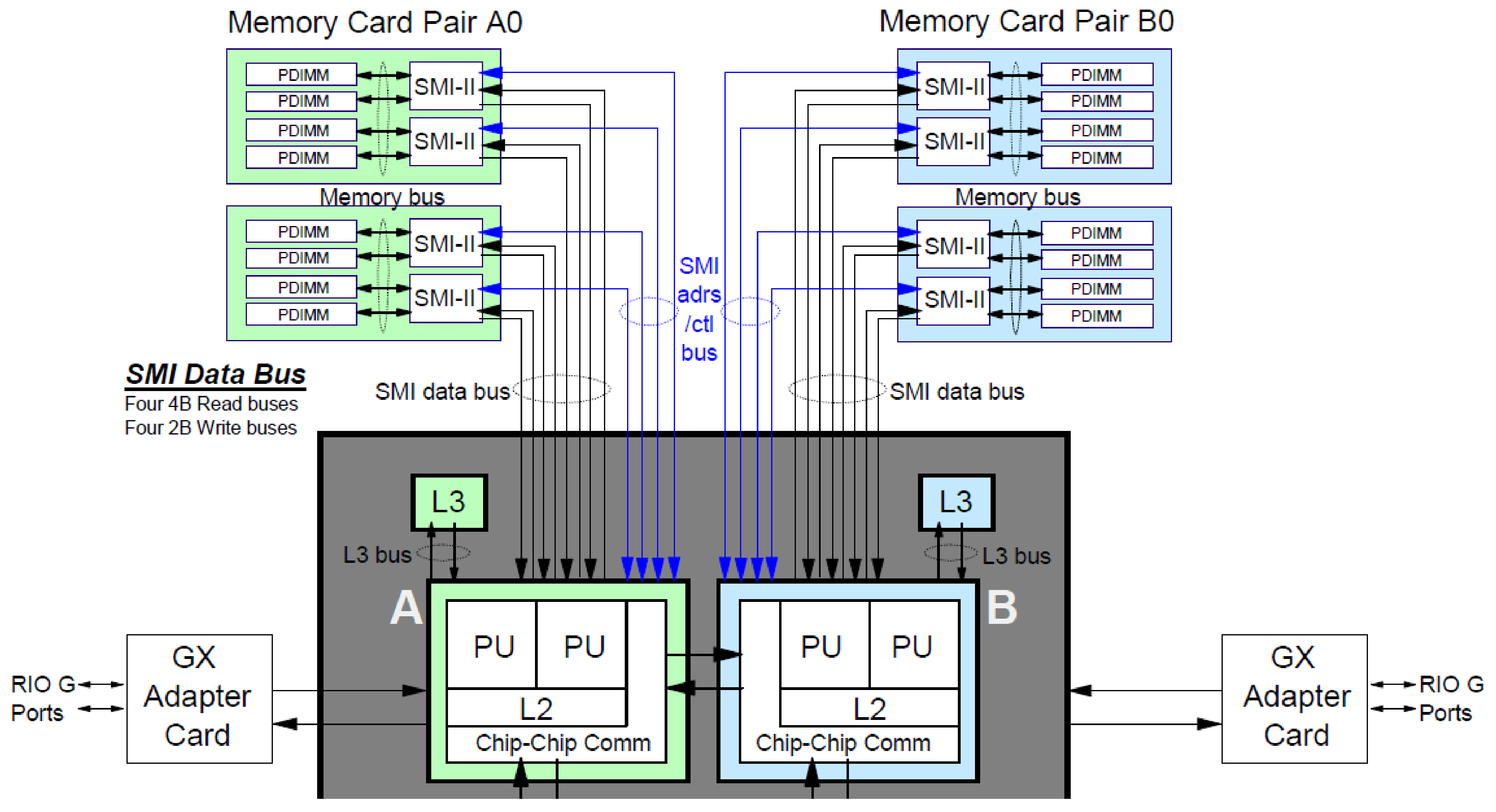


Figure: Attaching memory in the POWER5 based e-server p5 590 [24]

Example 2: A dual port POWER5 configuration (e-server p5 590) [24] -2

- The point-to-point read buses between the SMI chips and the Memory Controller are 4-byte wide whereas the point-to-point write buses between the SMI chips and the Memory Controller are 2-byte wide.
- Both buses operate at twice the speed of the DIMMs.
- Memory is mounted on memory cards shown below.
- Each memory card has four soldered DIMM cards and two SMI chips (called here SMI-II).
- The memory cards are IBM proprietary, mounted with DDR-200/266 or DDR2-533 memory and have 4 to 32 GB capacity.

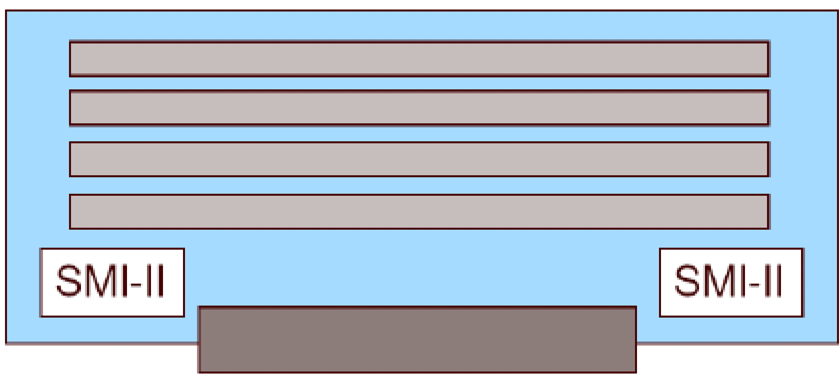


Figure: Memory card with four DIMM cards [24]

4.2.2 Redesigned memory subsystem (8)

Per socket bandwidth of the memory subsystem of the POWER5

- The maximum per socket memory bandwidth of the memory subsystems is constrained by both the serial buses connecting the SMIs and the available DIMMs.

- The serial buses limited memory bandwidth of a 2 port memory configuration with DDR2-533 memory DIMMs is:

  2 ports x 2 buses x (4 read B + 2 write B) x 2 x 533 MT/s = 25.6 GB/s

- The memory DIMMs limited memory bandwidth of a 2 port memory configuration with DDR2-533 memory DIMMs is:

  2 ports x 2 SMI chips x 2x DIMMs x 8 B x 533 MT/s = 34.1 GB/s

- As the above calculations show, in case of the POWER5 processor at last the serial buses constrained bandwidth is lower and limits the theoretical maximum bandwidth of the memory subsystem to 25.6 GB/s.

4.2.3 Enhanced modularity for building SMPs

There are three building blocks to implement larger POWER5 based SMP systems, as follows:

a) Dual Chip Modules (DCMs)

b) 8-wide Multi Chip Modules (MCMs)

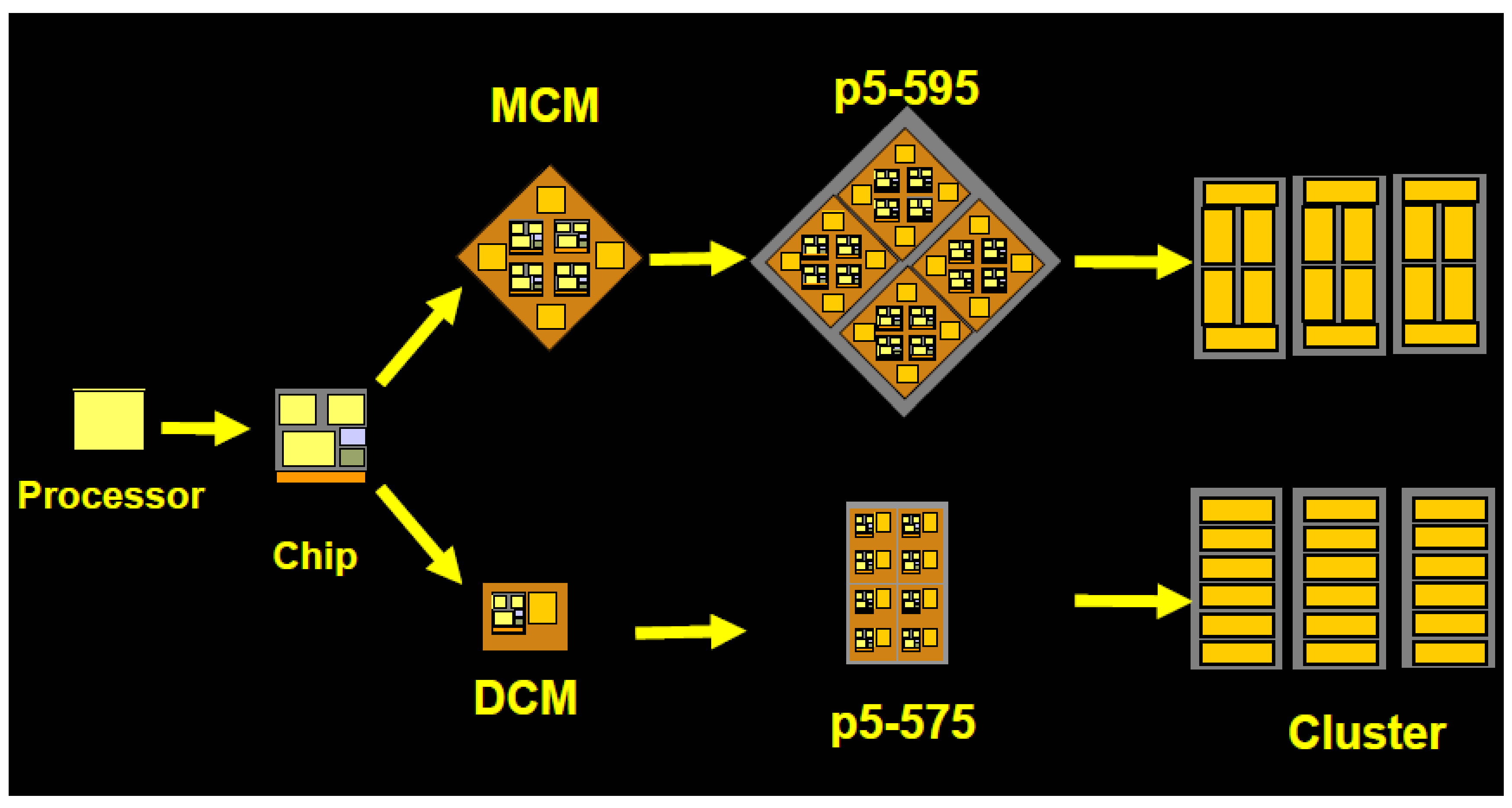
c) 16-wide Books

These building blocks will be briefly presented next.

Note that

- a Dual Chip Module refers to a single processor chip plus the associated L3 chip rather than two processor chips,
- an 8-wide Multi Chip Module (MCM) includes four processor chips with 8 cores
- and a 16-wide Book includes eight processor chips with 16 cores.

Principle of modularity in POWER5/POWER5+ based systems [2]



a) Dual Chip Modules (DCMs)

They are the smallest building blocks including a processor chip and an L3 chip in POWER5 based SMP systems, as shown below.

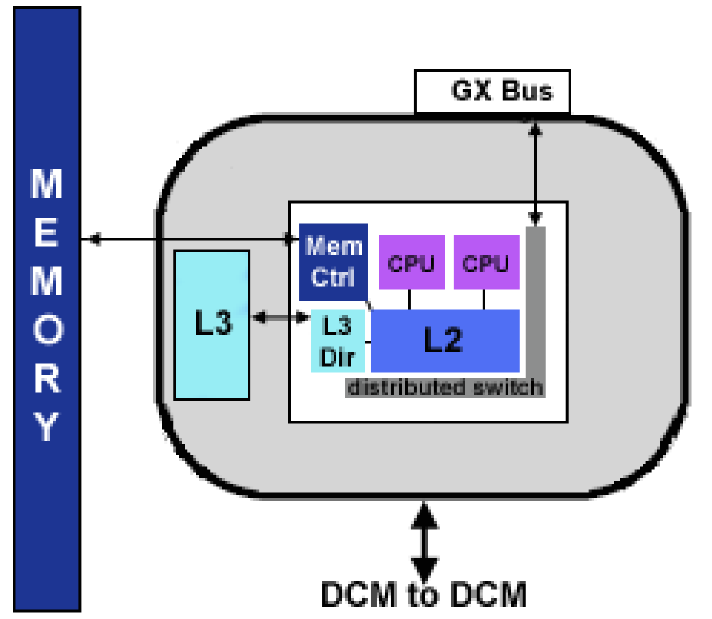


Figure: Layout of a DCM [20]

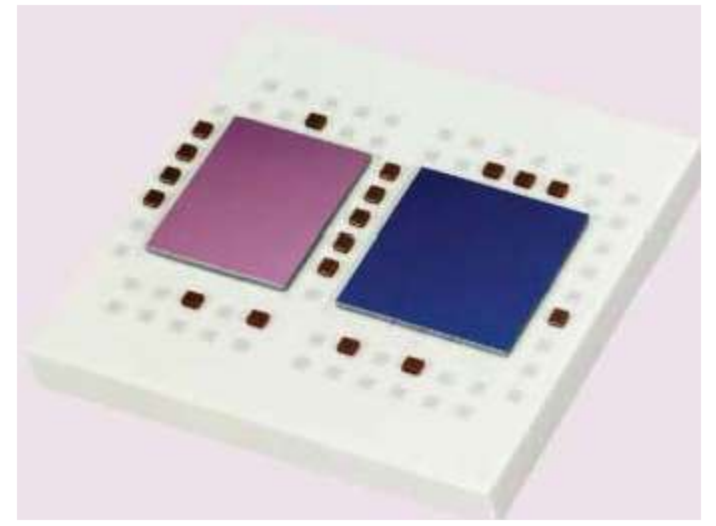


Figure: Implementation of a DCM [20]

Note that DCM modules were introduced by IBM first along with the POWER5 processor, no POWER4 based DCM modules were designed.

b) 4-processor/8-core Multi Chip Modules (MCMs)-1

They include four POWER5 chips with the associated L3 caches, linked together by two pairs of full speed 8-byte wide unidirectional buses where each processor chip is attached to the memory, as shown below.

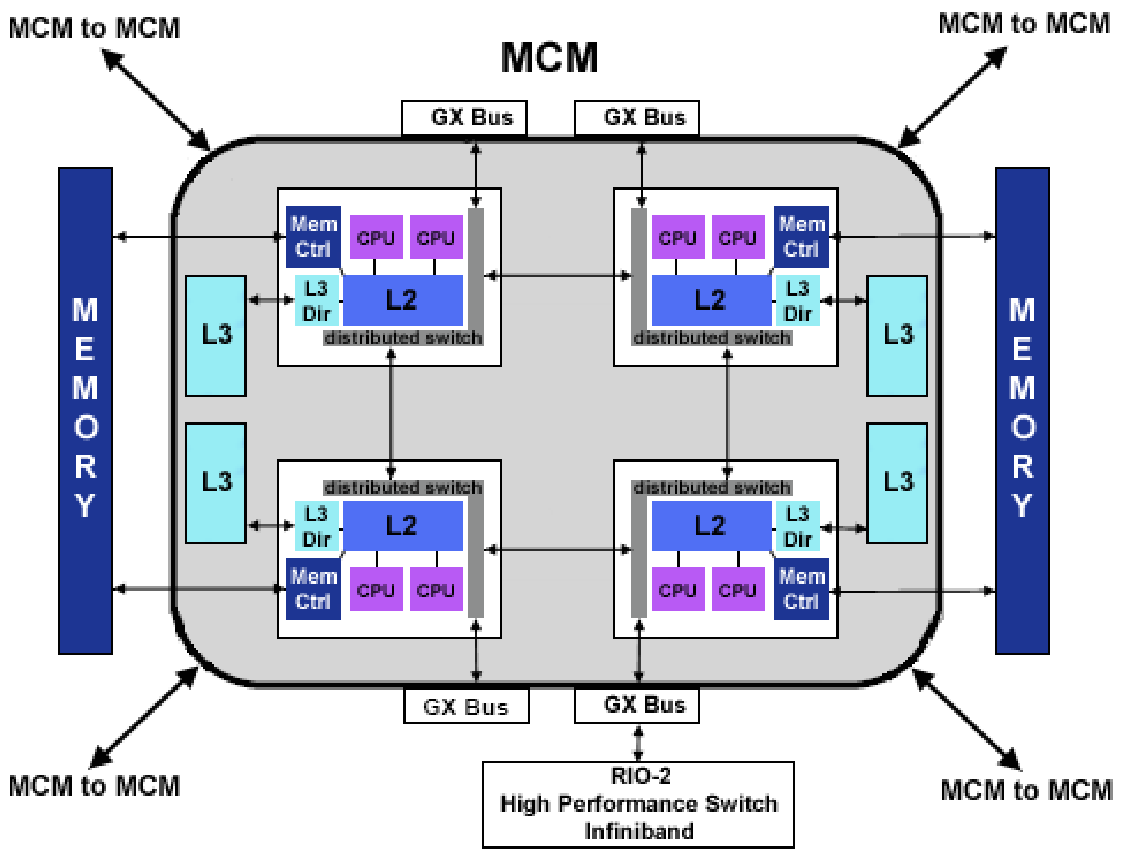
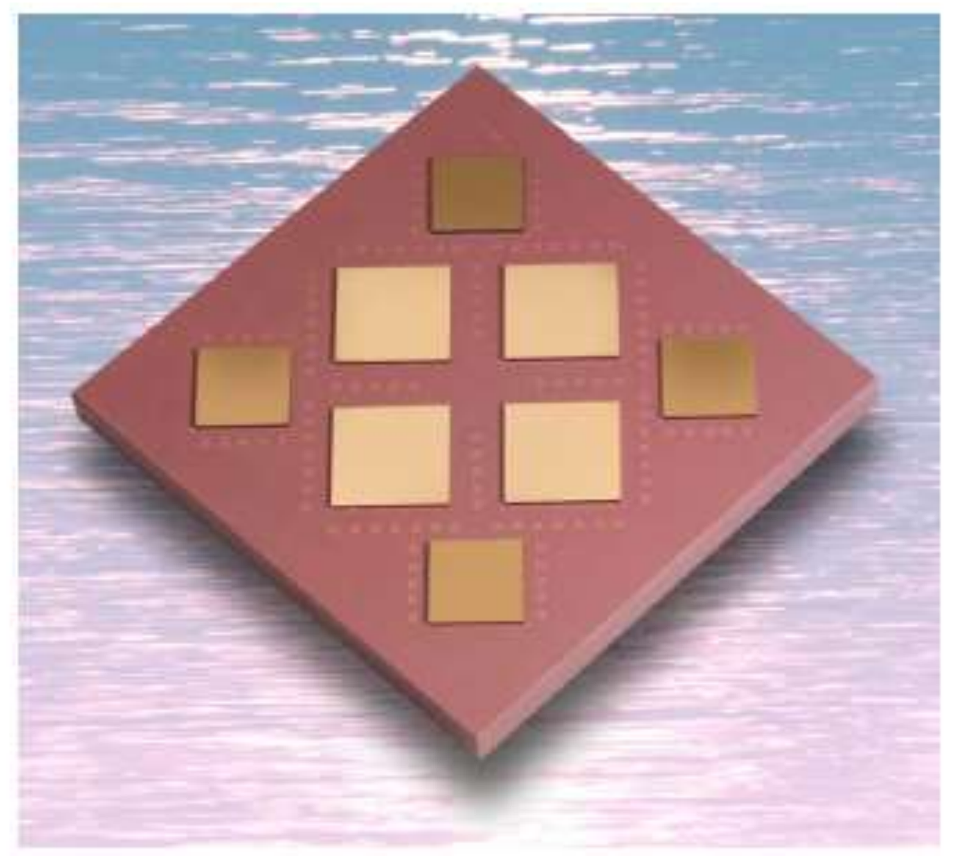


Figure: An 8-wide Multi Chip Module (MCM) [20]

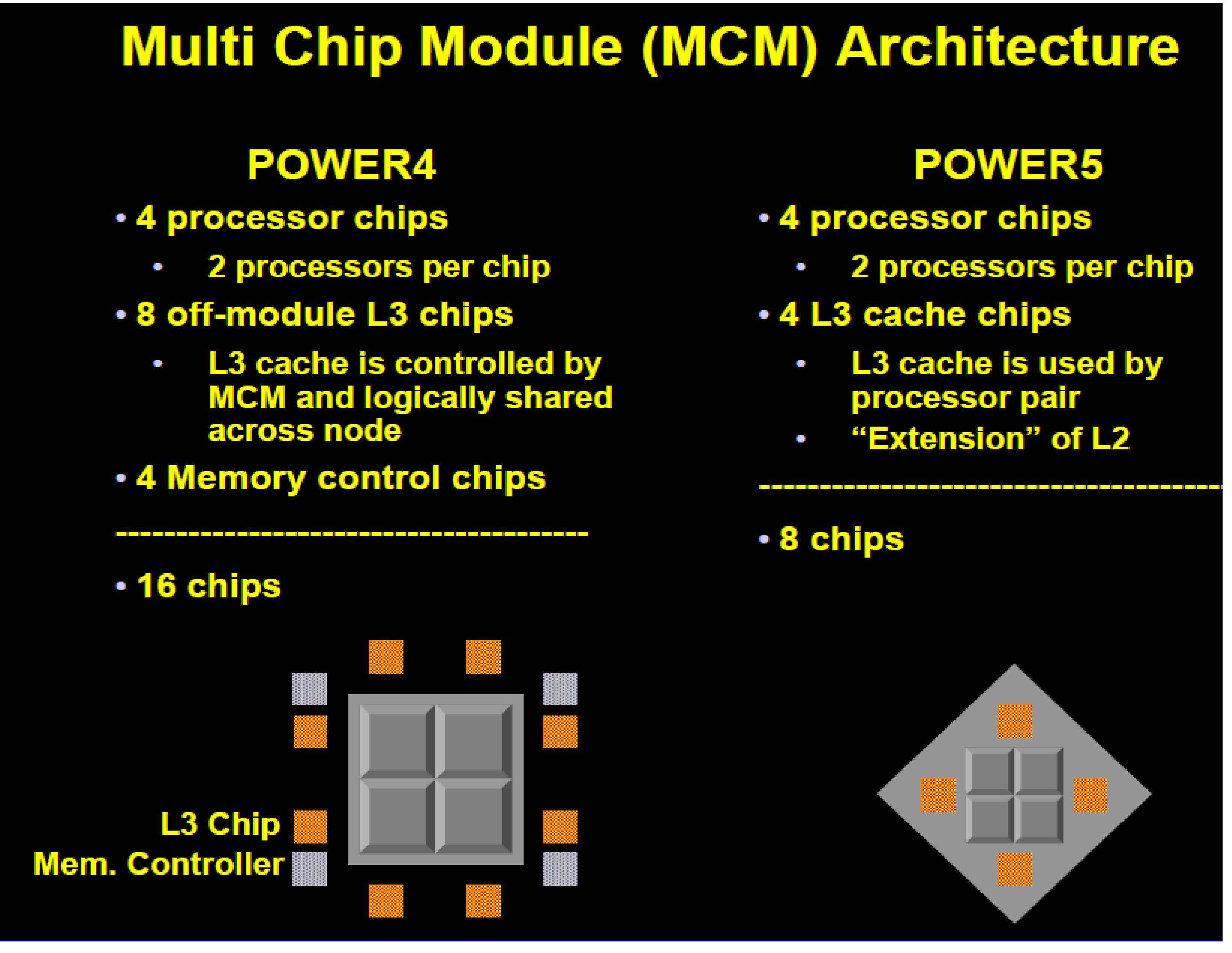
Note that two links allow only a partial interconnection of the four processor dies.

4-processor/8-core Multi Chip Modules (MCMs) -2 [25]

- 95mm × 95mm
- Four POWER5 chips
- Four cache chips
- 4,491 signal I/Os
- 89 layers of metal



Enhanced implementation of the Multi Chip Module (MCM) in POWER5 [21]



c) 8-processor/16-core Books

They include two MCMs appropriately interconnected through a pair of half speed 8-byte wide unidirectional buses and attached to memory, as shown below.

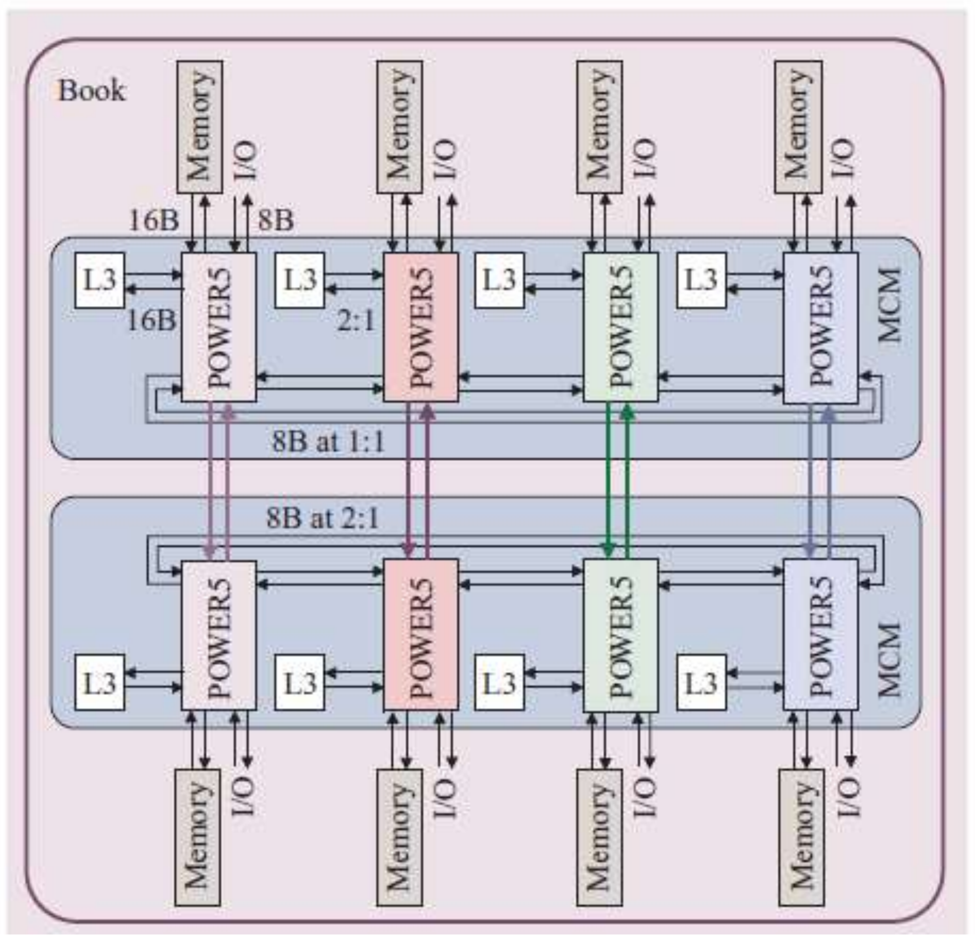
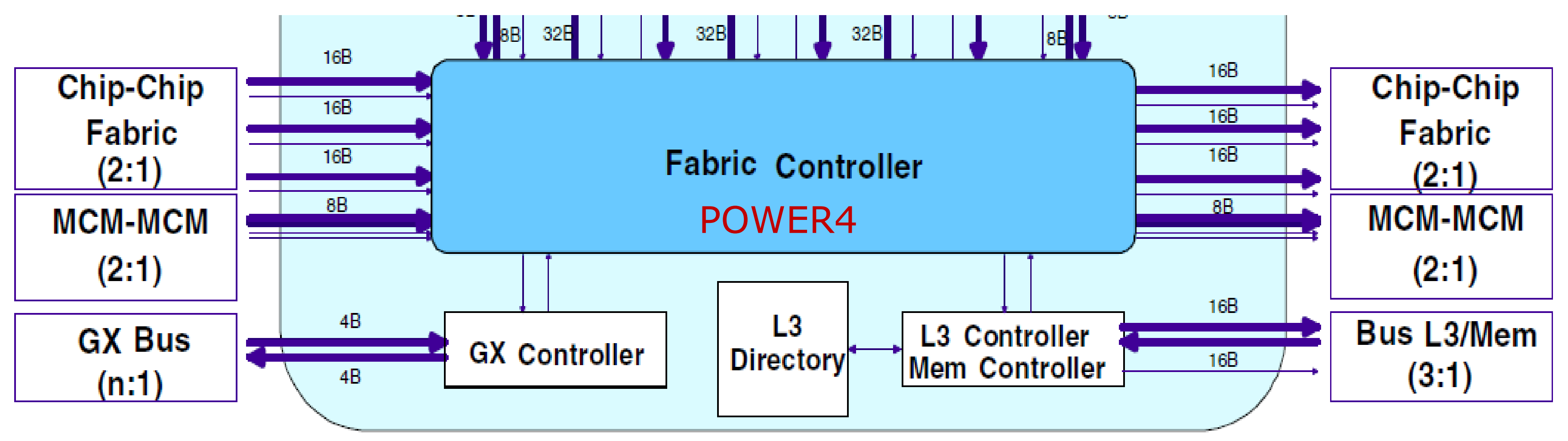
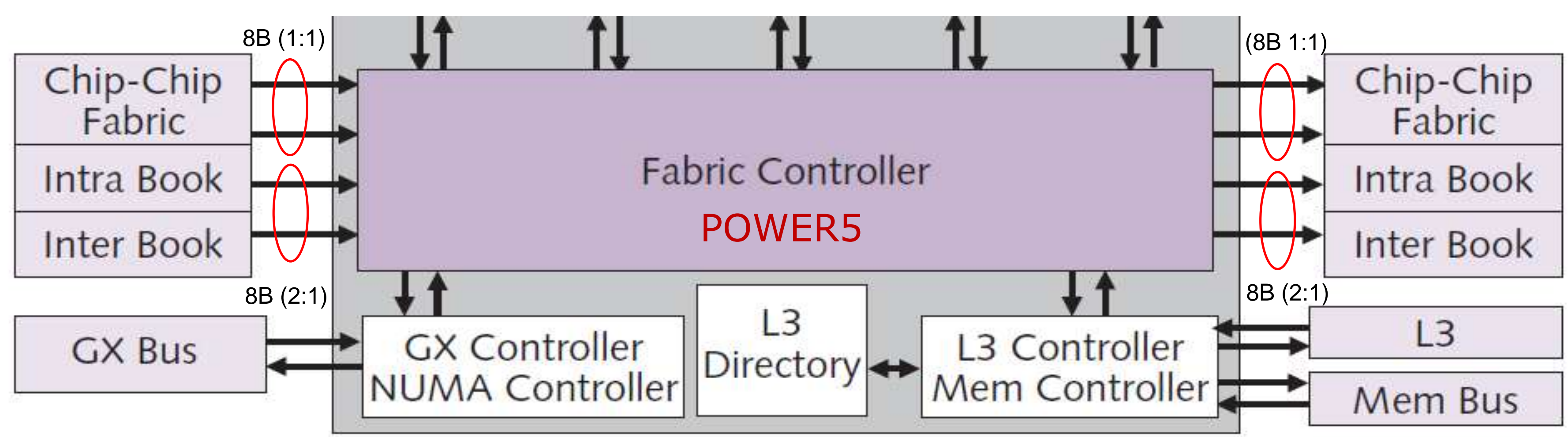


Figure: A 16-wide building block called a Book [23]

Photos of POWER5 based DCMs and MCMs [20]

4.2.4 Modified interconnection links [12], [26] -1





4.2.4 Modified interconnection links -2

A comparison reveals that

- the POWER5 provides only two Chip-to-Chip links, they are narrower but run at full speed, and
- on the other hand, the POWER5 has now two links to interconnect Books, they are 8 byte wide and run at half speed, by contrast the POWER4 provides only a single 8 byte wide half speed link.

32-processor/64-core SMP built of 4 Books [21]

The new interconnection link structure allows to build up to 64-way SMP systems, as indicated below.

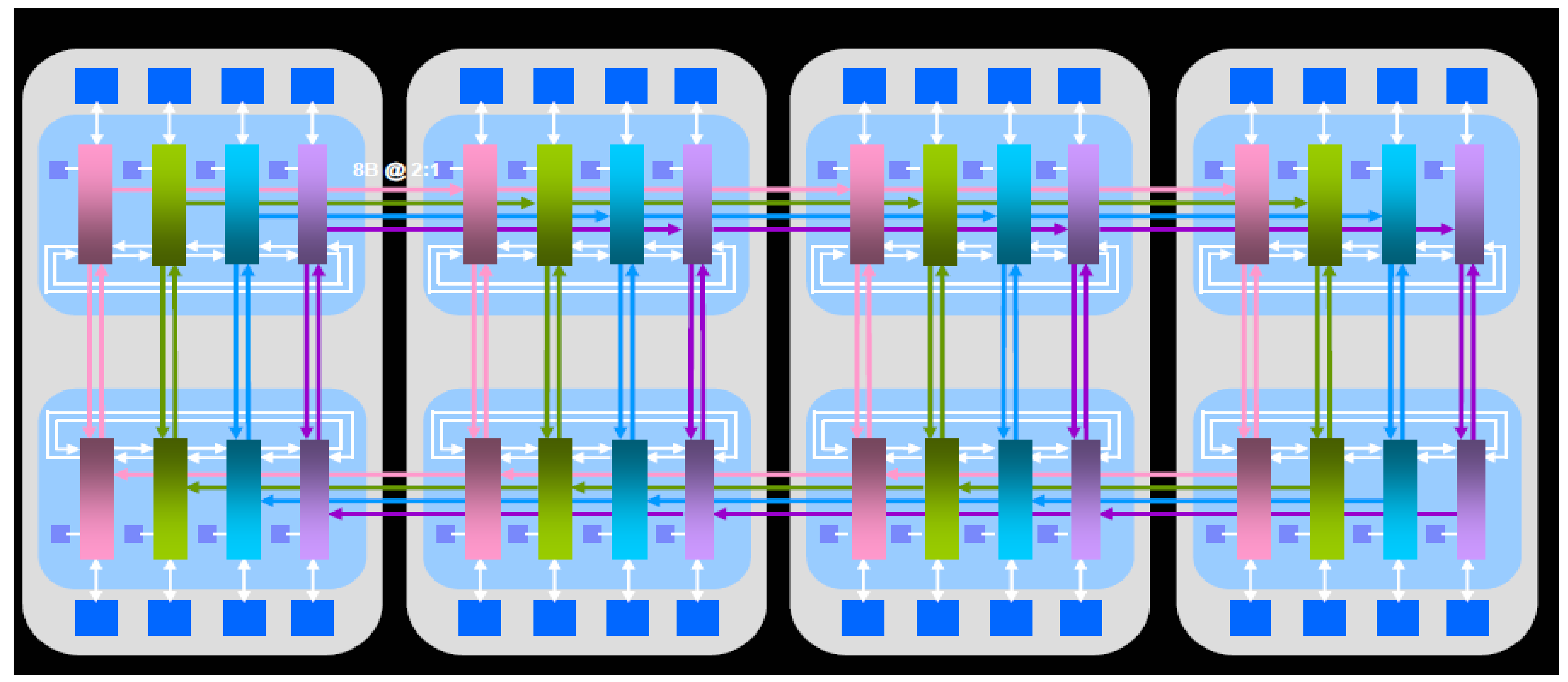


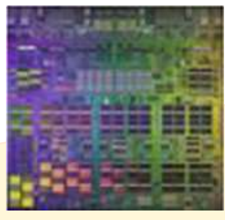
Figure: POWER5-based 64-way SMP [21]

Note that all intra-Book interconnects run at half processor frequency.

4.3 Key innovations of the POWER5

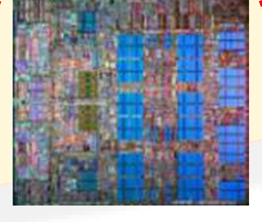
- 4.3.1 Introducing 2-way SMT
- 4.3.2 Integrating the memory controller to the processor die
- 4.3.3 Fine-grain clock gating to reduce switching power

4.3 Key innovations of the POWER5 (Die photos from [3])



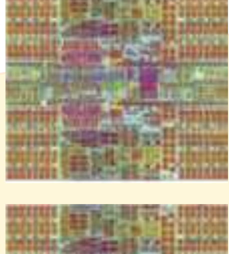
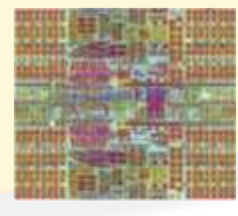
POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
with SMI chips
GX I/O bus
Support for SMP



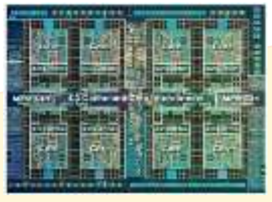
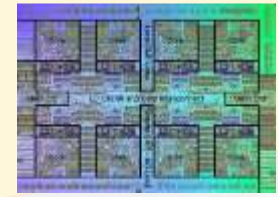
POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
clock gating

POWER6/6+
65/65 nm

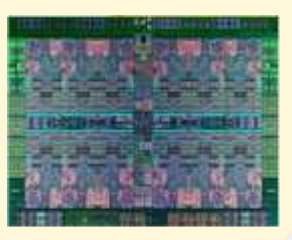
Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
Critical Path
Monitors
Nap idle mode

POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
Per core fc
Dyn. fan managm.
Sleep idle mode
*Accelerators for
cryptography
*Winkle idle mode

*POWER7+



POWER8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2001 | 2004 | 2007 | 2010 | 2014

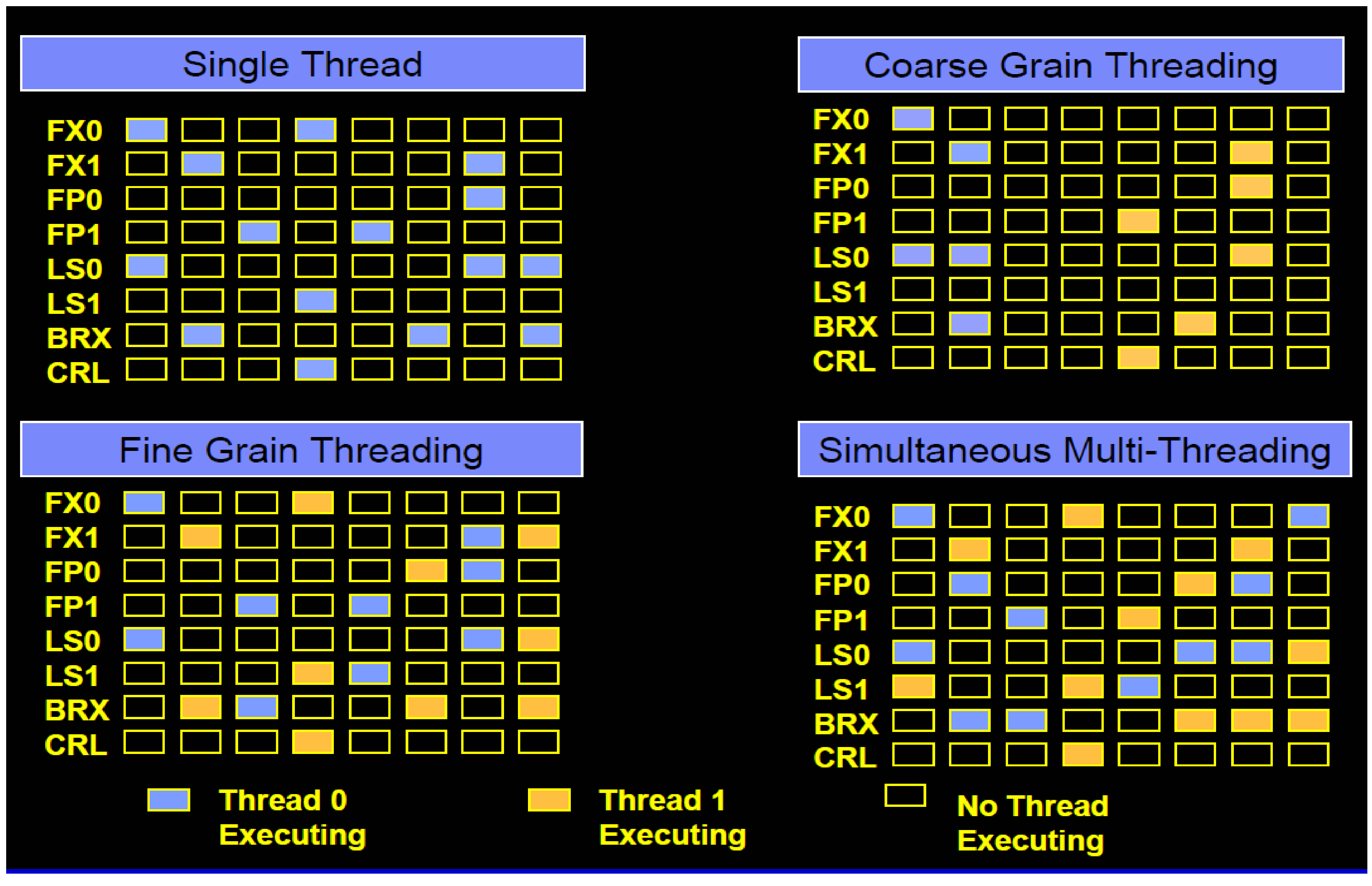
4.3.1 Introducing 2-way SMT

POWER5 introduces simultaneous multithreading (SMT) such that the processor can be operated

- either in single threaded (ST)
- or in symmetrical multithreaded (SMT)

mode.

Kinds of multithreading [21]



4.3.1 Introducing 2-way SMT (3)

Microarchitecture enhancements for the efficient implementation of SMT [23]

In order to run 2-way SMT efficiently, IBM enhanced execution resources of the microarchitecture

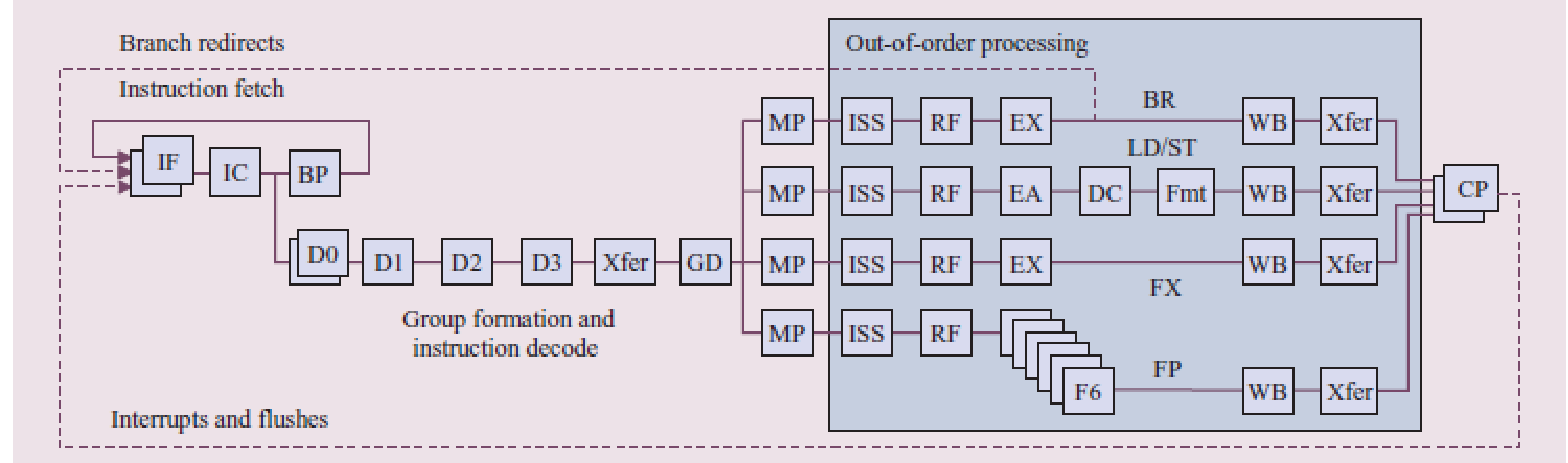
- partly by doubling resources (e.g. by providing two Program Counters) and
- partly by providing larger resources, e.g. by larger buffers etc.

To find out the best cost performance tradeoffs of the chosen solution designers carried out comprehensive simulations.

The next Figure shows IBM's solution for providing enhanced execution resources for SMT in the POWER5 processor.

Here we note that the POWER5 makes use of the same pipeline as the preceding POWER4 processor.

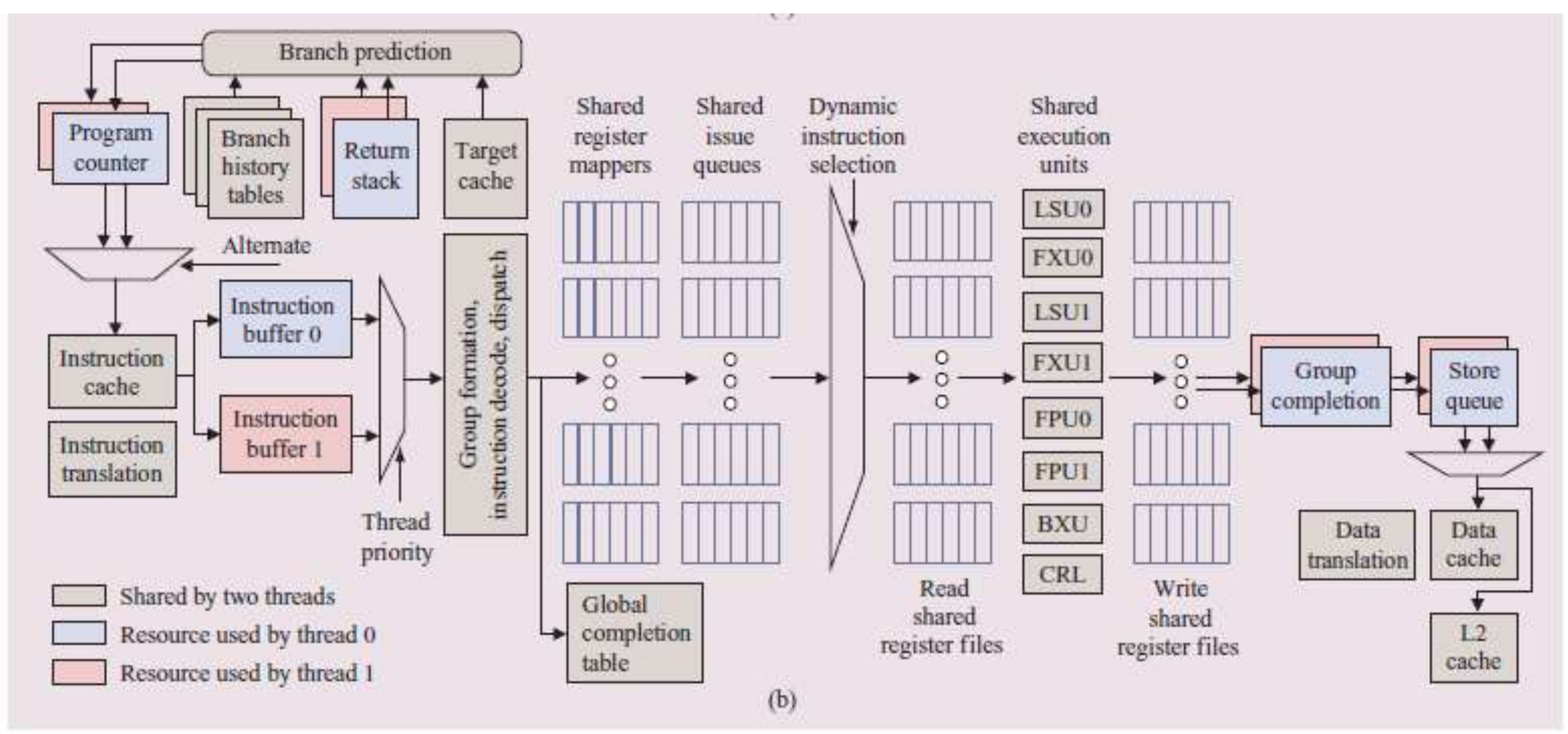
Instruction pipelines of the POWER5 (the same as that of the POWER4) [23]



IF: instruction fetch;
IC: instruction cache
BP: branch predict;
D0: decode stage 0
Xfer: transfer
GD : group dispatch
MP: mapping
ISS: instruction issue

RF: register file read
EX: execute
EA: compute address
DC: data caches
F6: six-cycle floating-point execution pipe
Fmt: data format;
WB: write back
CP: group commit;

Execution resources provided by the POWER5 for SMT processing [23]



BXU: branch execution unit;
CRL: condition register logical execution unit.

Contrasting the number of registers and issue queues in the POWER4 and POWER5 processors [23]

Resource type	*Logical size (per thread)*	*Physical size*	
		POWER4	*POWER5*
GPRs	32 (+4)	80	120
FPRs	32	72	120
CRs†	8 (+1) 4-bit fields	32	40
Link/count registers	2	16	16
FPSCR†	1	20	20
XER†	Four fields	24	32
Fixed-point and load/store issue queue	Shared by both threads	36	36
Floating-point issue queue	Shared by both threads	20	24
Branch execution issue queue	Shared by both threads	12	12
CR logical issue queue	Shared by both threads	10	10

†CR is an acronym for *condition register*. Architecturally, the CR consists of eight 4-bit fields that indicate how to resolve the direction of a branch instruction. In POWER4 and POWER5 systems, each field of the CR is treated as a separate register. FPSCR is an acronym for *floating-point status and control register*. XER is an acronym for *fixed-point exception register*. Four of the XER fields are renamed in POWER4 and POWER5 systems. See [1,6] for additional details.

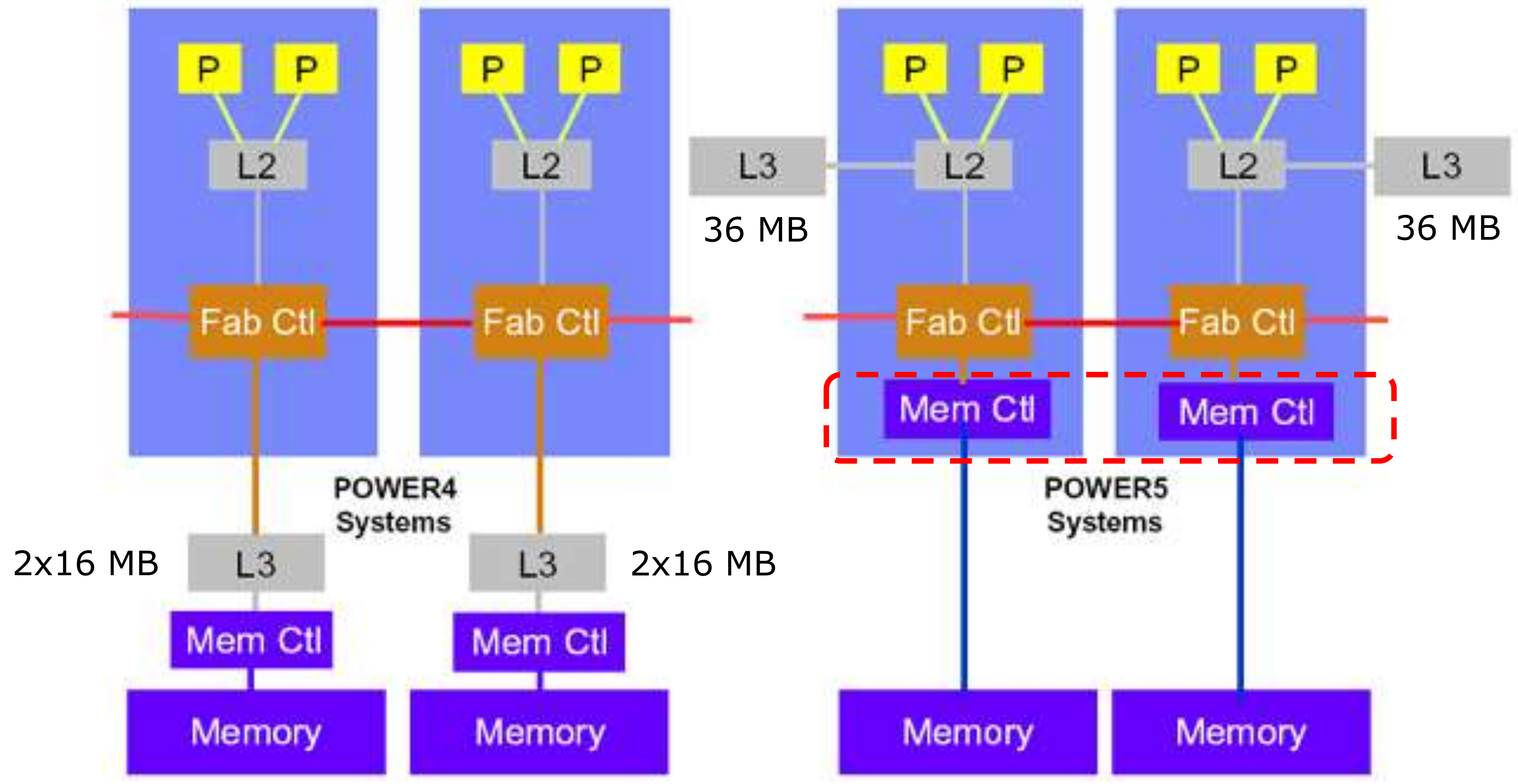
4.3.1 Introducing 2-way SMT (7)

Use of the single threaded (ST) mode [23]

- Some applications do not benefit really from SMT, e.g. such those run time is execution resource limited.
- This is the reason while POWER5 supports single threaded (ST) mode.
- In this mode the processor makes all available execution resources, like rename registers or execution queues, available for the single active thread.

  Such applications will run in ST mode faster than in SMT mode.

4.3.2 Integrating the memory controller to the processor die [20]



The on-die memory controller reduces latency to the SMI chips (Synchronous Memory Interface) connecting the memory vs. the previous outboard controller solution used on POWER4 systems [27].

4.3.3 Fine-grain clock gating to reduce switching power [23]

- Fine-grain clock gating switches off local clock buffers (LCB) if the set of latches it drives is expected not to be written in the next cycle.
- Power management logic determines whether or not a local clock buffer (LCB) has to be clock-gated in the next cycle.

  When an LCB is clock-gated, the set of latches it drives can still be read, but they cannot be written.
- Power-modeling tools were used to estimate the cost and benefit of using fine-grain clock gating in various parts of the circuit across a range of workloads and then it was decided whether or not it is beneficial to use fine-grain clock-gating in a specific part of the circuit.

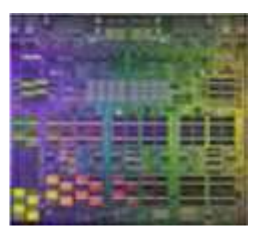
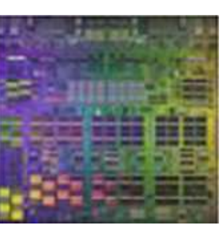
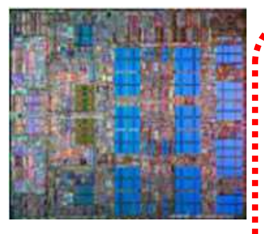
5. POWER5+

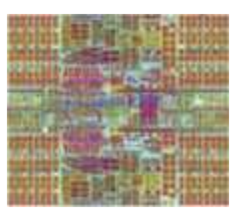
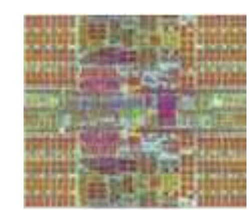
5. POWER5+

- Introduced 10/2005
- Main improvements vs. the POWER5 are
  - smaller feature size (90 nm vs. 130 nm) and consequently
  - higher clock rate of 1.9 to 2.3 GHz vs. 1.5 to 1.65 GHz,
  - a number of various enhancements in the microarchitecture, like increased number of TLB entries etc.,
  - increased memory speed from DDR-266 to DDR2-400 or DDR2-533.

    The related DIMMs have a proprietary 276 pin layout instead of the industry standard layout of 240 pins.
- Nevertheless, no noteworthy innovations were added in the POWER5+ vs. the previous model.

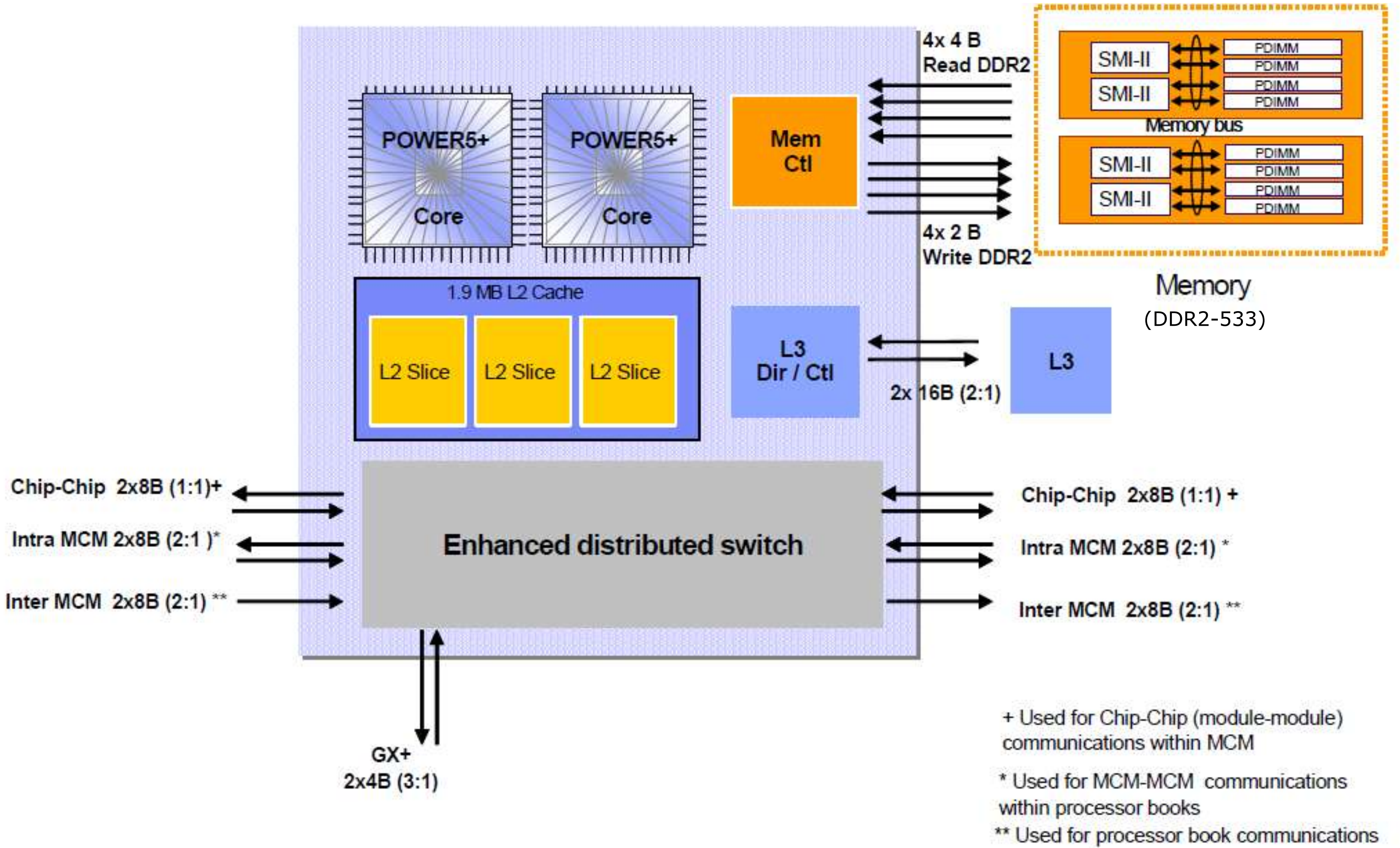
Key features of the POWER5+

	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

Block diagram of the POWER5+ [28]



Overview of the POWER5 and POWER5+ models [2]

Model	Processors	Clock Rate (GHz)	Max Memory (x 2^30 byte)
p5 595	16-64	1.65, 1.9	2000
p5 590	8-32	1.65	1000
p5 575	8-16	1.9, 2.2*	256
p5 570	2-16	1.9, 2.2*	512
p5 560Q	4-16	1.5*	128
p5 520	1,2	1.65, 1.9*	32
p5 505	1,2	1.5, 1.65*	32

* POWER5+

6. POWER6

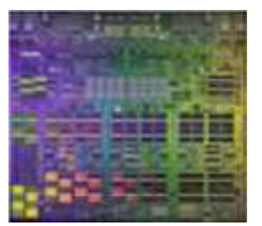
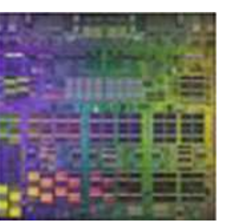
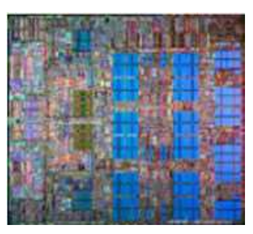
- 6.1 Introduction to the POWER6
- 6.2 Main enhancements of the POWER6 vs. the POWER5
- 6.3 Key innovations of the POWER6

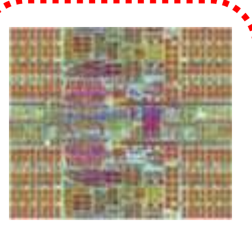
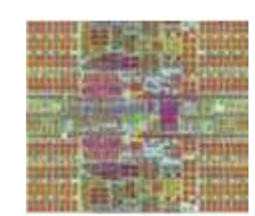
6.1 Introduction to the POWER6

6.1 Introduction to the POWER6

- Launched in 7/2007
- 65 nm technology
- 341 mm^2
- 790 million transistors
- High performance design achieved through high clock rates
  (13 FO4 design vs. the 22 FO4 POWER5 design)
- (Mostly) in-order core

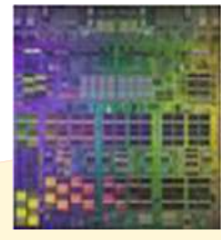
Key features of the POWER6

	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

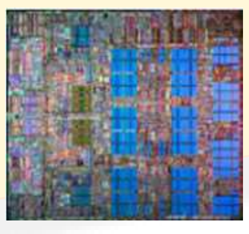
Key innovations of the POWER6 (Die photos from [3])



Power4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP

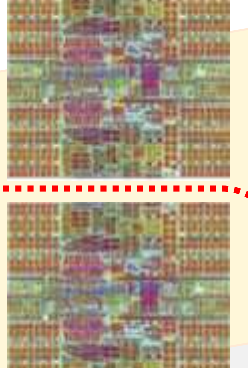
2001



Power5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating

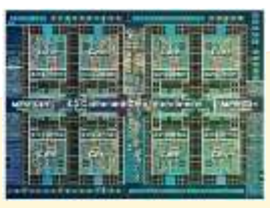
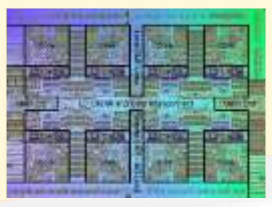
2004



Power6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

2007

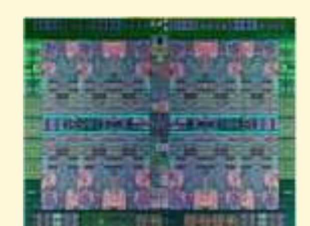



Power7/7+*
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

*POWER7+

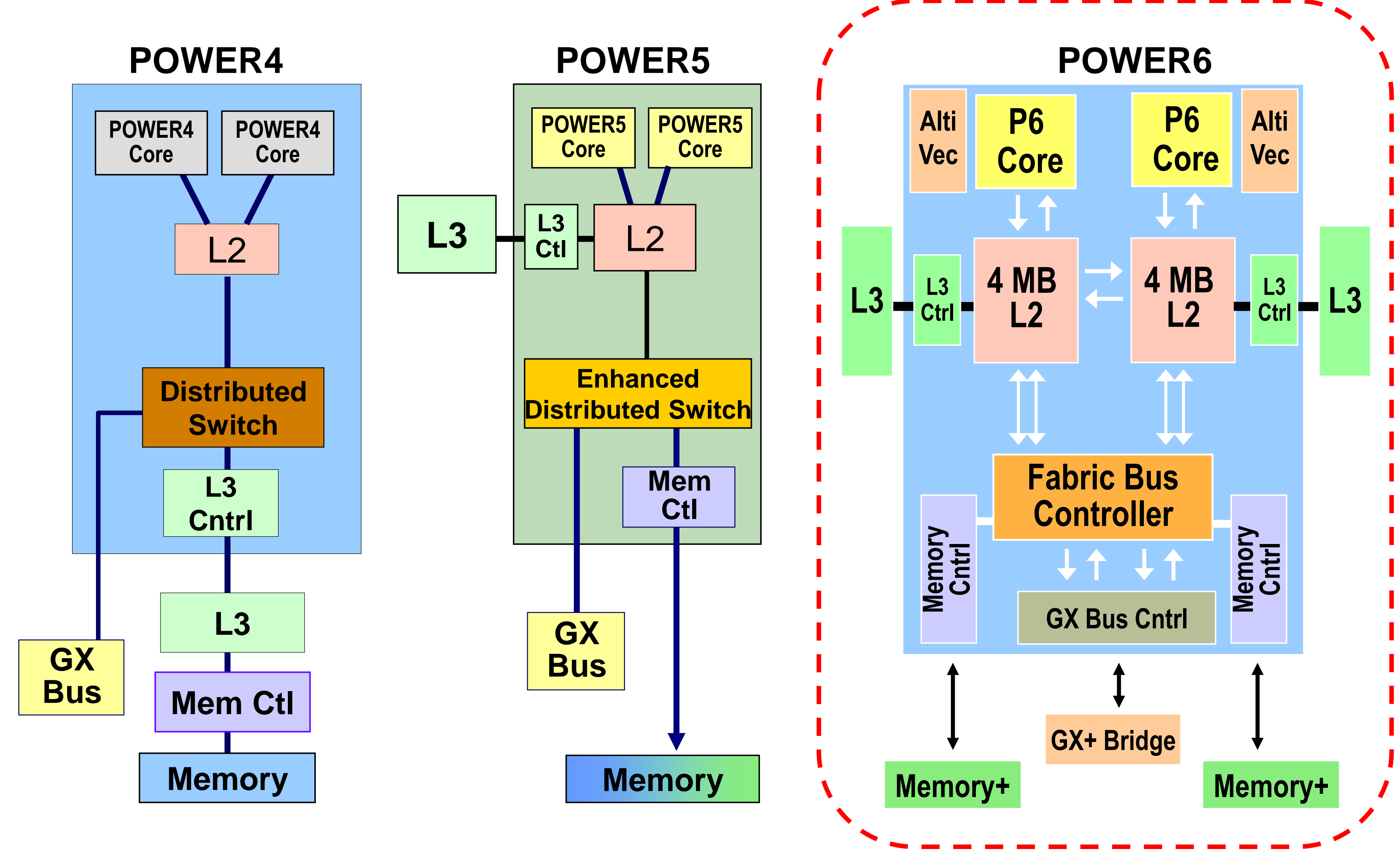
2010



12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

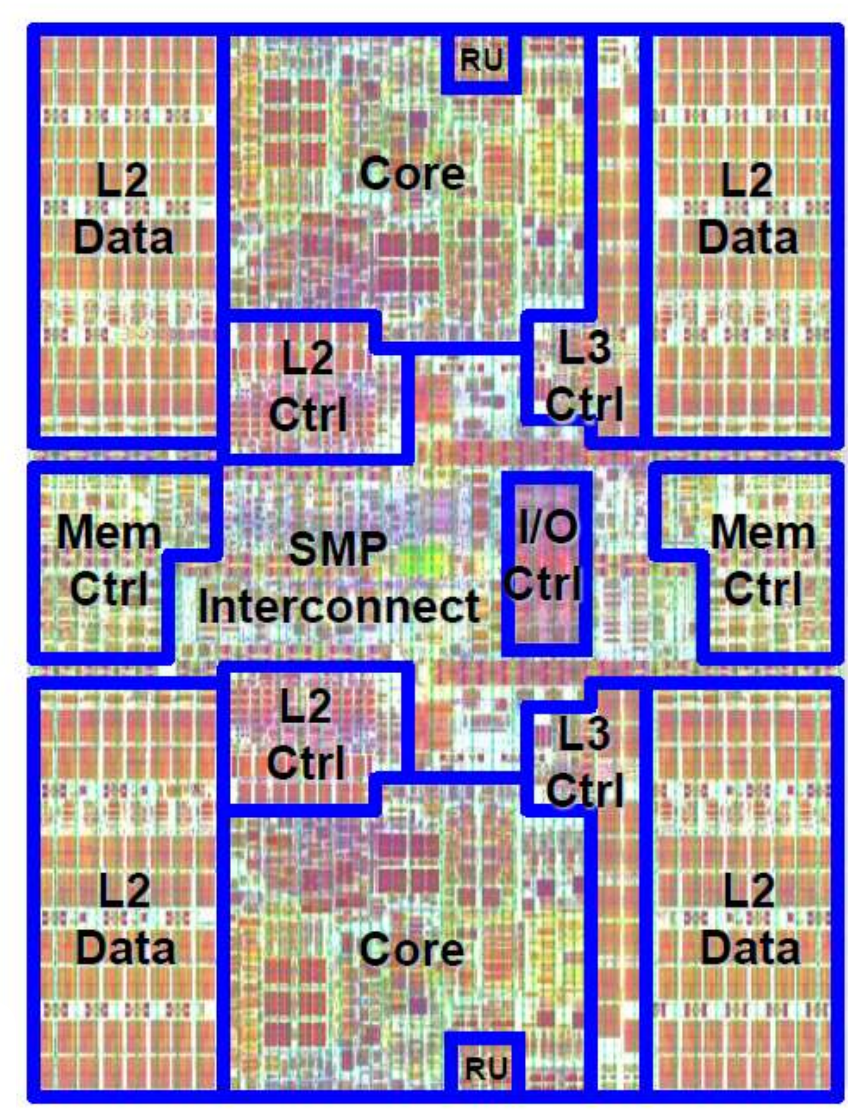
2014

High level block diagram of the POWER6 [5]



Layout of the POWER6 chip [29]

- **Ultra-high frequency dual-core chip**
  - 7-way superscalar, 2-way SMT core
  - 9 execution units
    - 2LS, 2FP, 2FX, 1BR, 1VMX,1DFU
  - 790M transistors
  - Up to 64-core SMP systems
  - 2x4MB on-chip L2
  - 32MB On-chip L3 directory and controller
  - Two memory controllers on-chip
  - Recovery Unit
- **Technology**
  - CMOS 65nm lithography, SOI
- **High-speed elastic bus interface at 2:1 freq**
  - I/Os: 1953 signal, 5399 Power/Gnd



IBM POWER5+ and POWER6 based servers [30]

IBM POWER6 based servers and their main features [30]

Footprint, Packaging	IBM JS22 Blade	IBM System p 520 19-inch 4U rack Deskside	IBM System p 550 19-inch 4U rack Deskside	IBM System p 570 19-inch 4U rack	IBM System p 575[2] 24-inch frame by node
Processor	POWER6	POWER6	POWER6	POWER6	POWER6
# of processors (# of cores)	4	1, 2, 4	2, 4, 6, 8	2, 4, 8, 12, 16	32
GHz clock	4.0	4.2	3.5, 4.2	3.5, 4.2, 4.7	4.7
DDR2 GB memory	2 to 32	2 to 64	2 to 256	2 to 768	4 to 256
Internal storage*	73GB – 146TB	73GB – 30.6TB	73GB – 30.6TB	73GB – 79.2TB	146.8GB – 5.1TB
Maximum rPerf	30	31.48	68.20	134.35	N/A
PCIe PCI-X slots PCI-X 266 slots GX bus slots	0 to 2 0 to 2 0 0	3 0 to 56 2 to 50 2	3 0 to 56 2 to 50 2	4 to 16 0 to 140 2 to 200 2 – 8	0 to 4 0 to 20 0 to 16 2
Max I/O drawers	N/A	8	8	32	1
Max micro-partitions	40	40[1]	80[1]	160[1]	254[1]
AIX® support	5.3, 6.1	5.3, 6.1	5.3, 6.1	5.2, 5.3, 6.1	5.3, 6.1
Linux® support	RHEL 4.5 / 5.1 SLES 10	RHEL 4.5 / 5.1 SLES 10	RHEL 4.5 / 5.1 SLES 10	RHEL 4.5 / 5.1 SLES 9 or 10	RHEL 4.5 / 5.1 SLES 10

1 Requires purchase of optional feature to support micro-partitions   *With maximum I/O drawers   Optiona

6.2 Main enhancements of the POWER6

- 6.2.1 High frequency core design
- 6.2.2 Providing more per-core execution units
- 6.2.3 Switching from shared to per-core L2 caches
- 6.2.4 Strongly enhanced L2 cache sizes
- 6.2.5 Increased cache and memory bandwidth figures
- 6.2.6 Dual memory controllers with redesigned high speed channels
- 6.2.7 Dual memory configurations

6.2.1 High frequency core design -1

The POWER6 achieves a remarkable high clock rate of 3.5 – 5.0 GHz, in comparison with the previous POWER processors, as the Figure below shows.

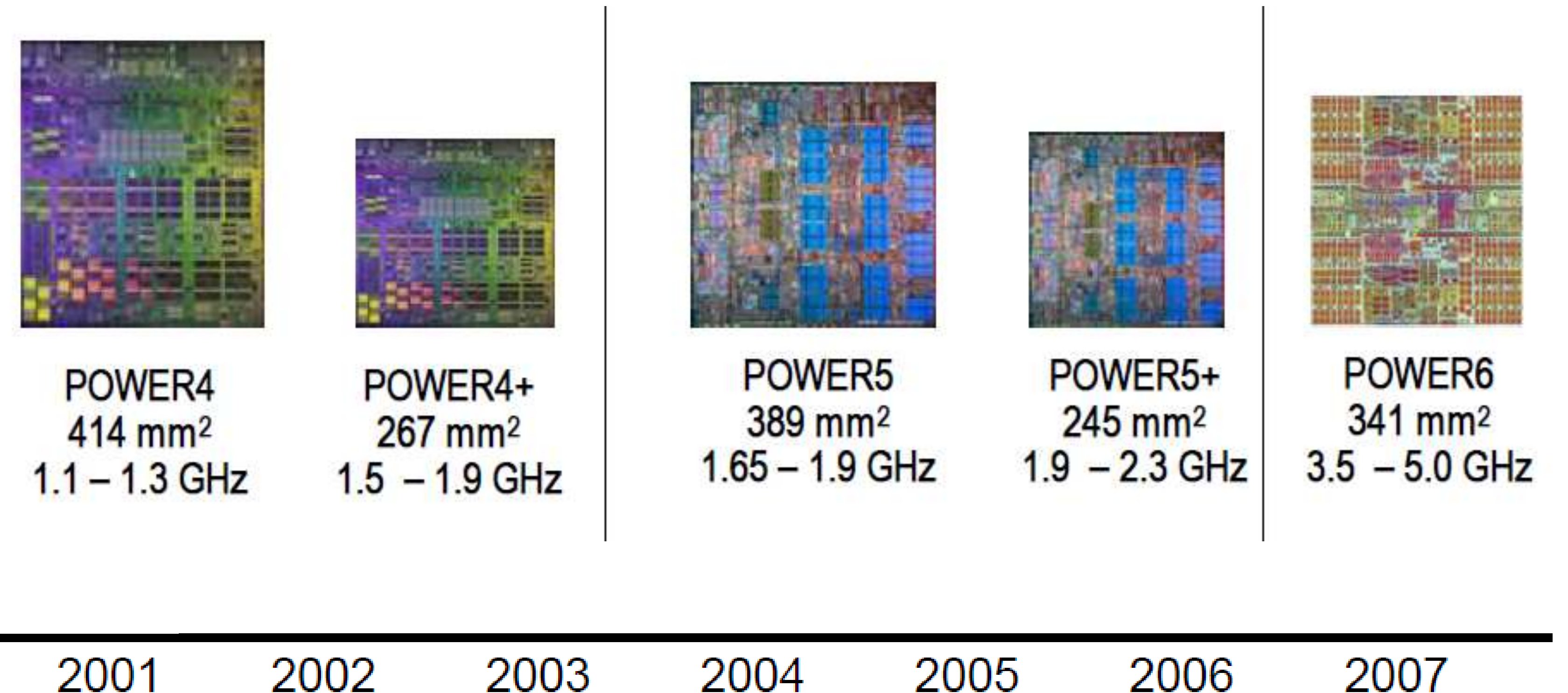


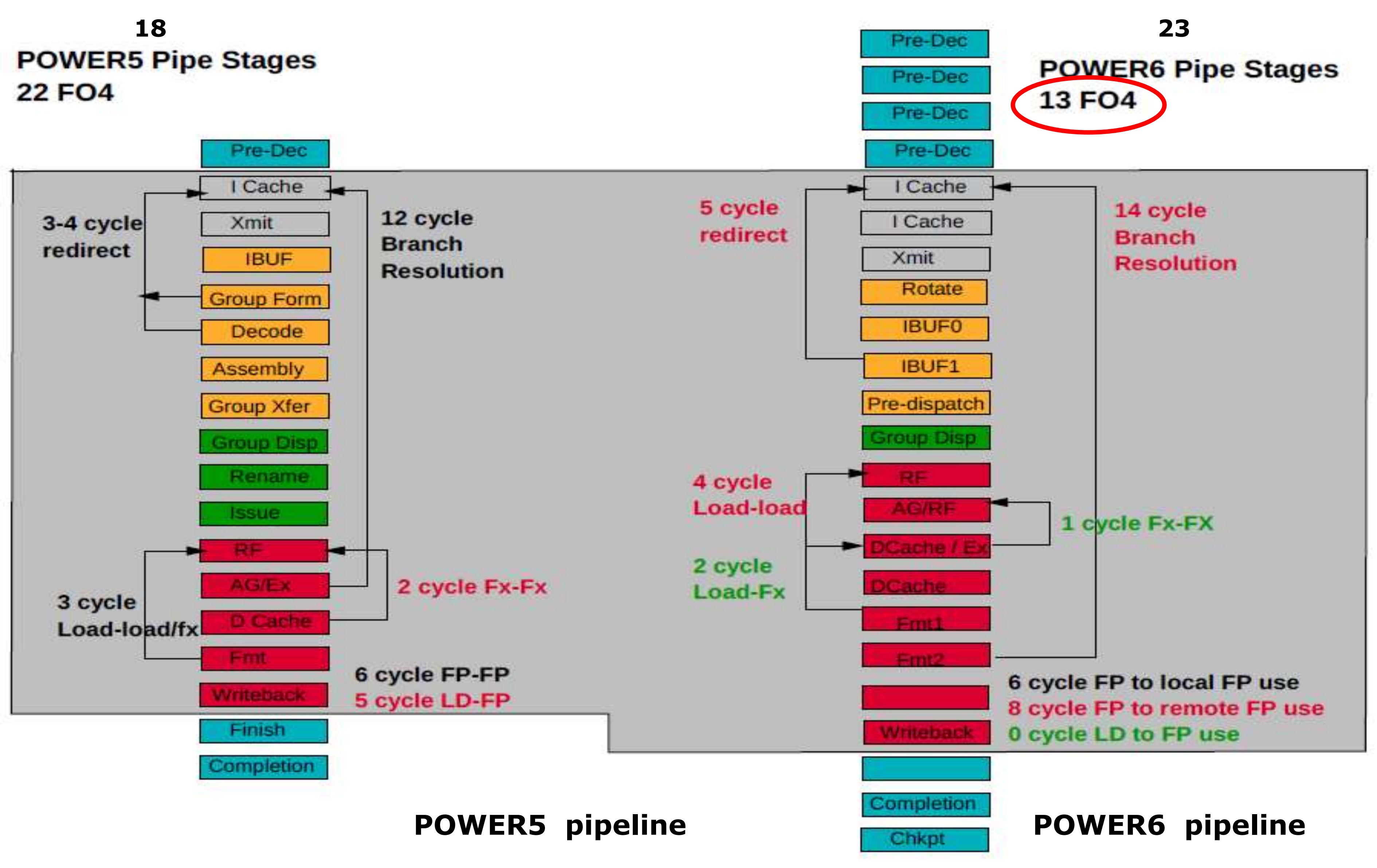
Figure: Die areas and clock frequencies of the POWER4 to POWER6 lines [30]

6.2.1 High frequency core design (2)

High frequency core design -2

- The main contributor for the high frequency core design is its 13 FO4 pipeline layout vs. the 23 FO4 pipeline implementation of the POWER5, as indicated in the next Figure.

Pipeline layout of the POWER6 vs. the POWER5 [5]

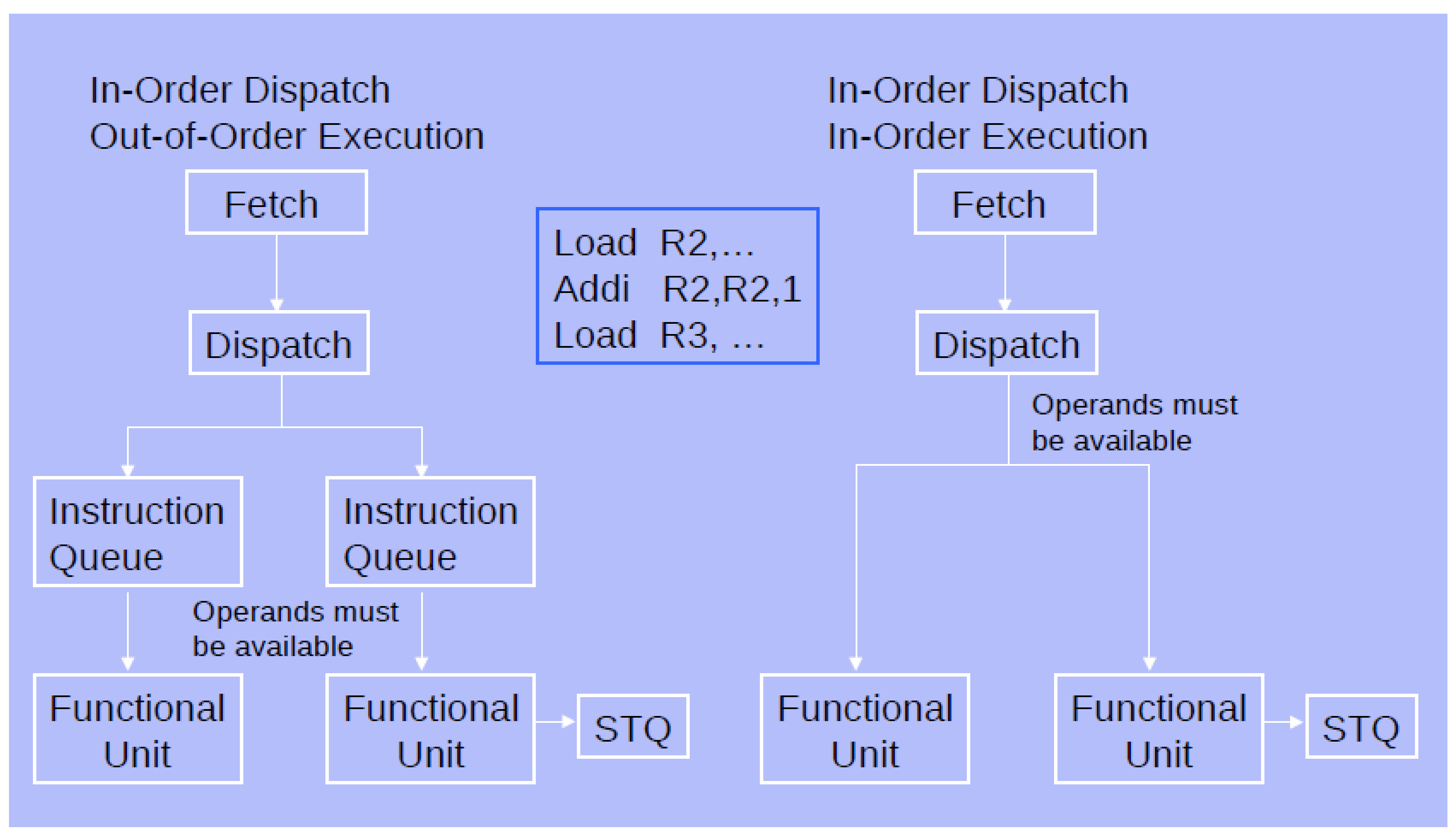


6.2.1 High frequency core design (4)

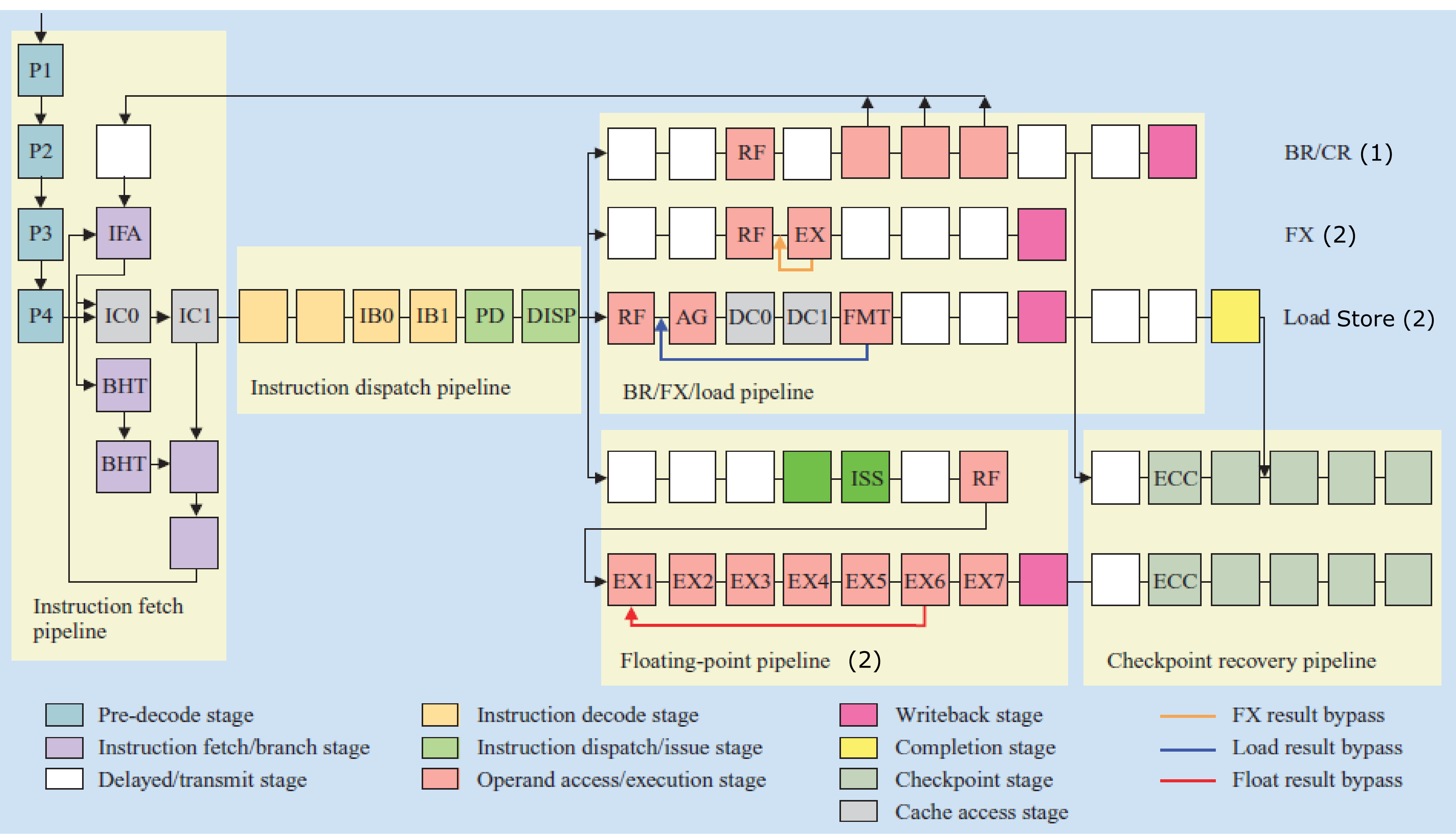
High frequency core design -3

- The 13 FO4 design resulted in almost doubling the clock frequency.
- In order to achieve a 13 FO4 pipeline implementation, designers abandoned register renaming and massive out-of-order execution, i.e. they implemented basically an in-order pipeline (see the next Figure).

In-order execution vs. out-of-order execution [5]



The in-order pipeline of the POWER6 [31]



AG: address generation; BHT: branch history table; BR: branch; DC: data-cache access; DISP: dispatch; ECC: error-correction code; EX: execute; FMT: formatting; IB: instruction buffer; IC0/IC1: instruction-cache access; IFA: instruction fetch address; ISS: issue; P1–P4: pre-decode; PD: post-decode; RF: register file access

6.2.1 High frequency core design (7)

Remarks [31]

- If needed, delay stages were inserted into the pipelines (designated by white boxes).
- The FP execution pipeline is decoupled from other pipelines and is allowed to run overlapped with the execution of load and FX instructions on the respective pipelines.

6.2.1 High frequency core design (8)

High frequency core design -4

- On the other hand, to compensate the impediments of in-order processing designers raised ILP, i.e. the efficiency of the microarchitecture as far as possible by various enhancements including
  - raising the number of the per-core available execution units from 8 to 9 (Section 6.2.2),
  - switching from shared to per-core L2 caches (Section 6.2.3),
  - raising massively the available L2 cache size (from 2 MB to 8 MB) (Section 6.2.4),
  - increased bandwidth figures (Section 6.2.5),
  - increasing the memory bandwidth by implementing a second memory controller as well (Section 6.2.6),
  - two possible memory configurations (Section 6.2.7).
- We note that memory and L2 cache bandwidth enhancements were also enforced by almost doubling the clock rate.

6.2.2 Providing more per-core execution units

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-ways
Width of the front-end	5	5	5	6	8	12
Dispatch rate	5	5	(In-order design)	6	8	12
Issue rate	8	8	7	8	10	16
No. of execution units per-core	8	8	9	12	16	20
No/type of execution units per-core	2 FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP 1BR/CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 2VMX, 1DFU, 2LU, 1 Crypto	8AGEN, 4VSU(128) 4LS(128) 2BRU DFU Cripto

6.2.3 Switching from shared to per-core L2 caches -1

POWER6 switched from a shared L2 cache used in the POWER5 to per-core L2 caches, as seen below.

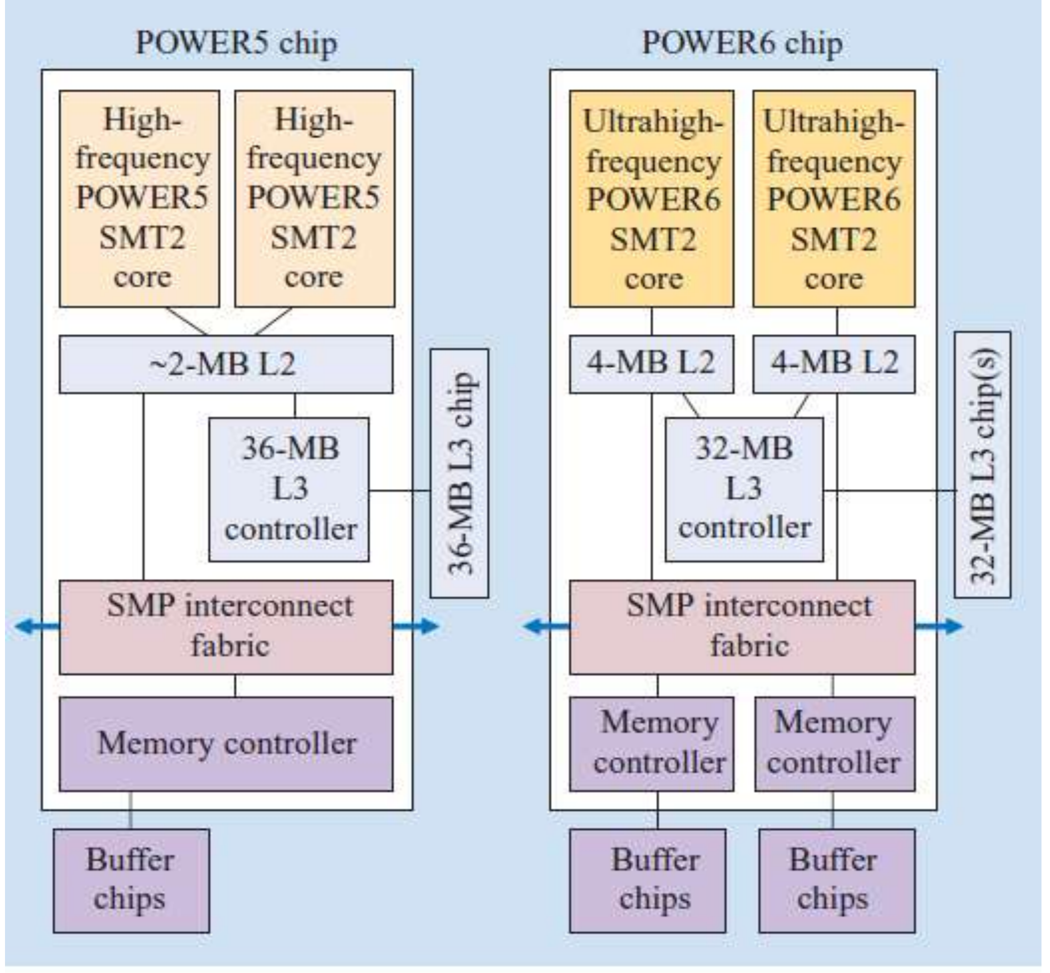
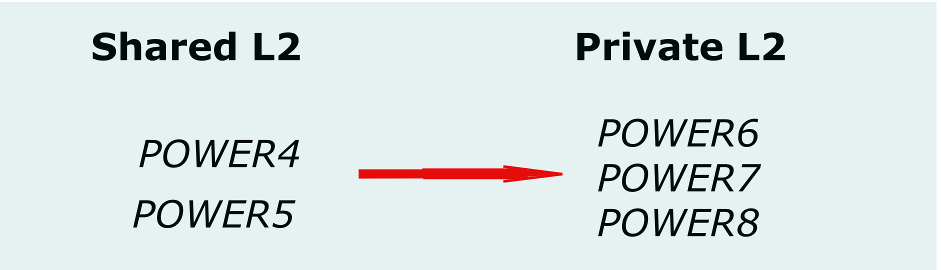


Figure: Switching to per-core L2 caches in the POWER6 [31]

6.2.3 Switching from shared to per-core L2 caches -2

As mentioned before, beginning with the POWER6 private (per-core) L2 caches displaced shared L2 caches in the POWER family.



The assumed reason for switching to the private scheme is as follows:

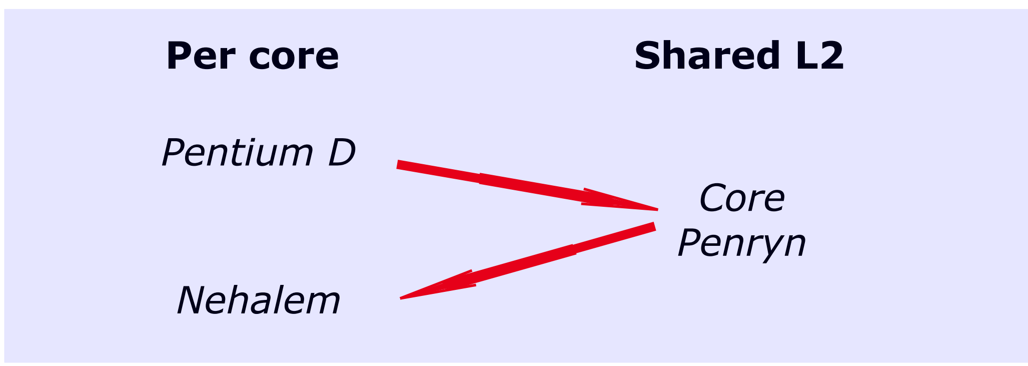
Private caches allow a more effective hardware prefetching than shared ones since

- hardware prefetchers track memory access patterns and
- private L2 caches have more easily detectable memory access patterns than shared L2 caches.

6.2.3 Switching from shared to per-core L2 caches (3)

Remarks

- A similar evolution from shared L2 caches to per-core L2 caches occurred in Intel's lines from Pentium 4 to Nehalem.
- Intel's first dual core Pentium D models included per-core L2 caches for a straightforward implementation.
- Subsequently, the Core 2 and Penryn changed to a shared L2 cache for a more efficient cache use (since unused parts of the L2 cache associated to one core may be used by the other core).
- Then the quad core Nehalem introduced a three level cache structure and came back to private L2 caches since now private L2 caches result presumable in a more performance gain than provided by a more efficient shared L2 cache.



6.2.4 Strongly enhanced L2 cache sizes

Thanks to the reduced feature size (65 nm) IBM was able to almost quadruple L2 cache size in the POWER6, from almost 2 MB to 8 MB, as seen below, to raise ILP.

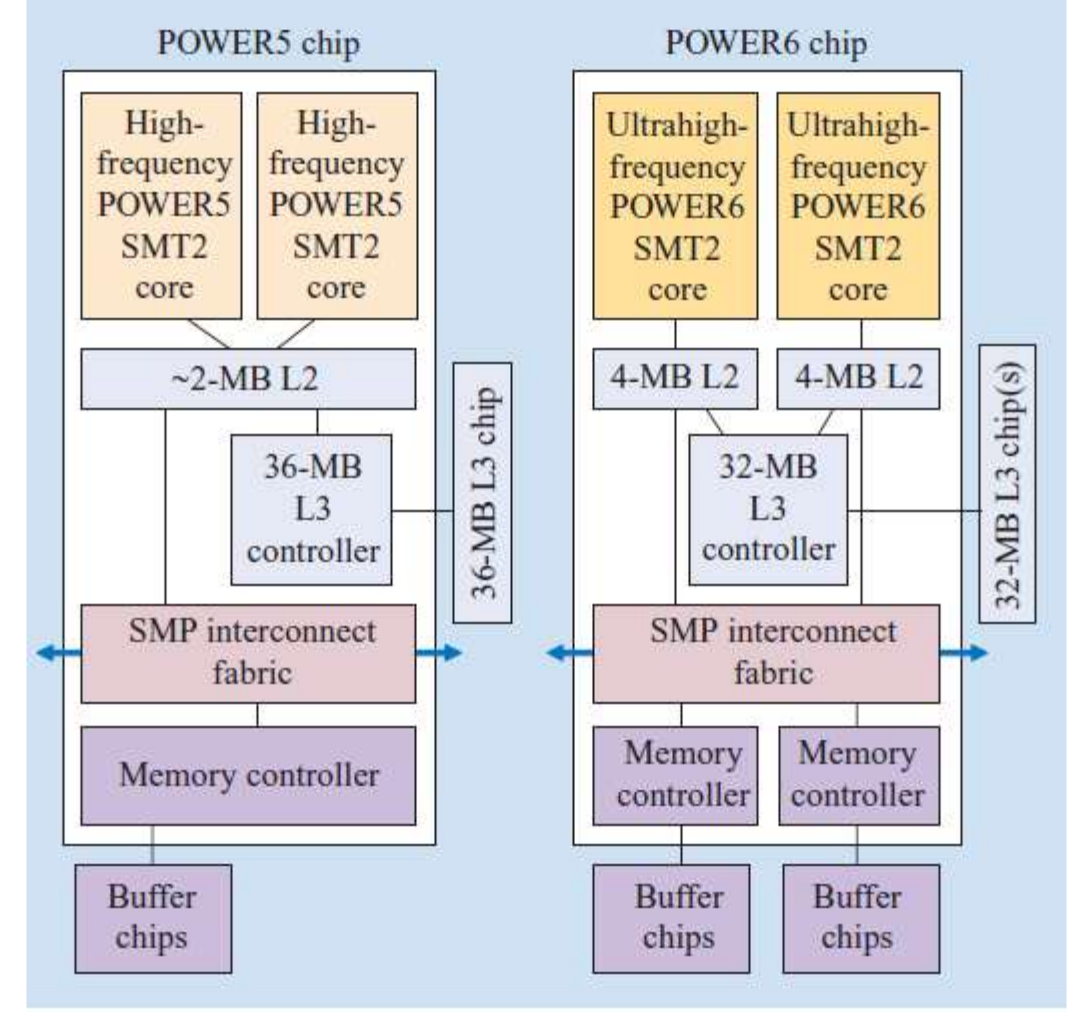


Figure: Enlarging the L2 cache size in IBM's POWER6 line vs. the previous POWER5 line [31].

6.2.5 Increased cache and memory bandwidth figures [31]

Throughput resource	*POWER5 processor*	*POWER6 processor*
Data from L2 to core	32 B per cycle to data cache	64 B per 2 cycles
Data from core to L2	8 B per cycle	16 B per 2 cycles
L2 busy time	2 cycles for 64 B	4 cycles for 128 B
Aggregate L2 bandwidth	64 B × 3 per 2 cycles (2 cores)	128 B per 4 cycles (1 core)
Data from L3 to core/L2	128 B per 16 cycles	128 B per 16 cycles
Data from L2 to L3	128 B per 16 cycles	128 B per 16 cycles
Memory data into chip	16 B per (2 × 533 MHz) peak	16 B per (4 × 800 MHz) peak
Chip data out to memory	8 B per (2 × 533 MHz) peak	8 B per (4 × 800 MHz) peak
SMP fabric data into chip	48 B per 2 cycles	67% of 40 B per 2 cycles
Chip data out to SMP fabric	48 B per 2 cycles	67% of 40 B per 2 cycles

6.2.6 Dual memory controllers with redesigned high speed channels -1

POWER6 models are clocked about twice as high as the previous POWER5 models, this doubles roughly the processors memory bandwidth need.

In fact the POWER6 already makes use of faster memory DIMMs (DDR-667 vs. DDR-533), this however compensates only partly for the nearly doubled clock rate, so IBM introduced a second memory controller as well to double the number of memory channels and hence also memory bandwidth.

The second controller is activated only in specific models.

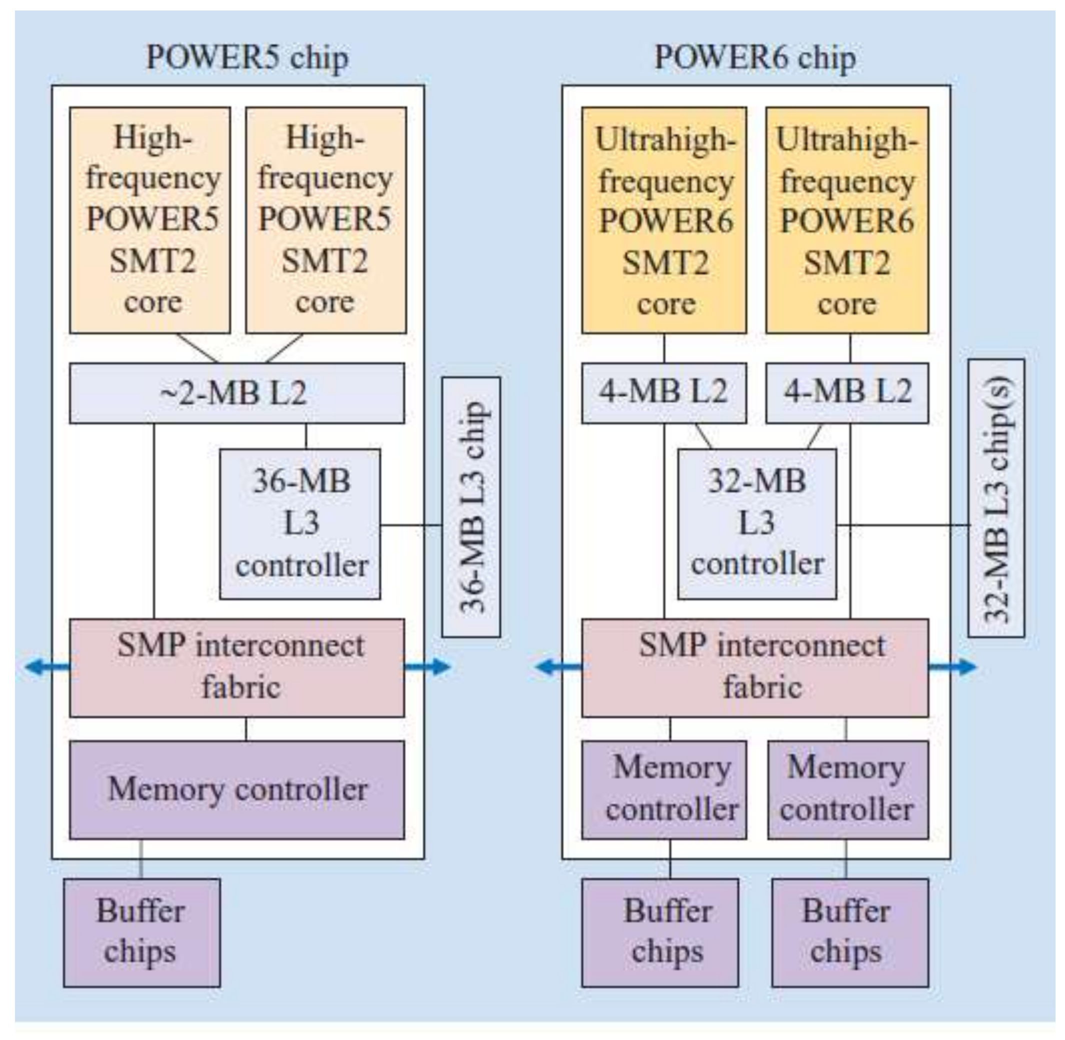


Figure: Introducing dual memory controllers in the POWER6 to double memory bandwidth [31].

Dual memory controllers with redesigned high speed channels -2

- Each of up to two memory controllers support up to four serial high speed channels, i.e. each controller has four ports.
- A channel supports a 2-byte read datapath, a 1-byte write datapath, and a command path, all operating at four times the DRAM rate, that amounts to up to DDR2-667.
- Each channel is linked to a buffer that is
  - either on the processor card (when DDR2 memory is used),
  - or on the DIMM (when FB-DIMM memory is used),

  as discussed next.

6.2.7 Dual memory configurations

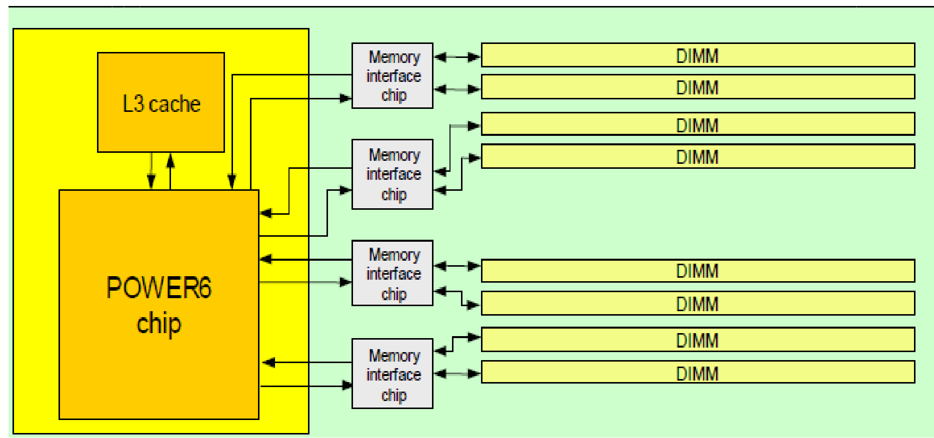
In their POWER6 models IBM implemented two types of memory configurations; one with commodity DDR2-DIMMs and another one with FBDIMMs, as shown below.

Memory configurations of the POWER6

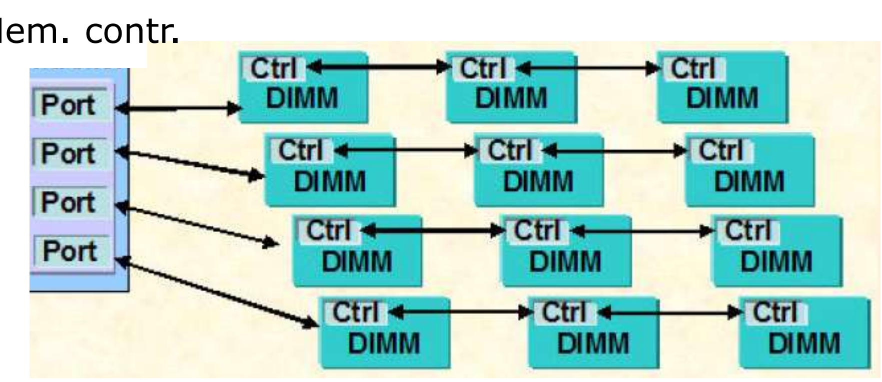
Commodity DIMM based configuration

FB-DIMM-based configuration

Both examples take for granted a single memory controller



Mem. contr.



DDR2-667 commodity DIMMs
(Replacing IBM proprietary DIMMs of previous models)

Power 520 (1xMC/2 ports)
Power 550 (1xMC/4 ports)
Power 570 (1xMC/4 ports) with the FC 5621 proc. card

DDR2-667 FBDIMMs

Power 570 (1xMC/4 ports)
with FC 5620/5622/7380 proc. cards
Power 595 (2xMC/4 ports each)

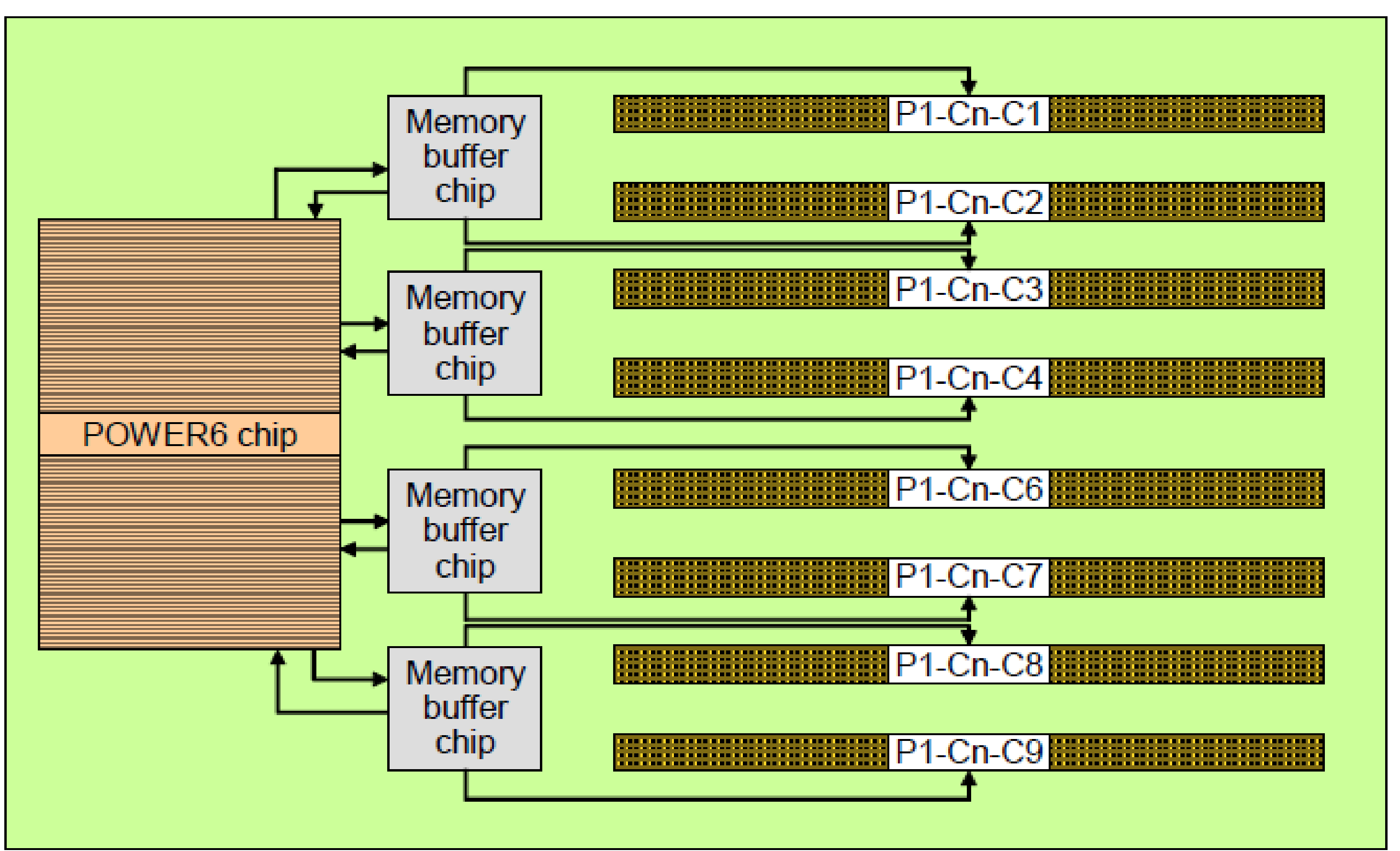
Figure: Implemented memory configurations of the POWER6 [32], [33]

6.2.7 Dual memory configurations (2)

a) Commodity DIMM based memory configuration

IBM equipped low end and midrange POWER6 models with memory buffer based memory subsystems including industry standard 240 pin DDR2-667 DIMMs in contrast to previous memory implementations using IBM specific 208 pin DDR DIMMs.

Example 2: POWER6 processor card with commodity DIMM based memory [3



- The Figure depicts a Power 570 based processor card with a POWER6 processor connected to four Memory Buffers via a memory controller and four ports.
- The ports are connected via high speed links to the Memory buffer chips that serve two DDR2-667 commodity RDIMMs each by standard 8-byte DDR2 buses.

6.2.7 Dual memory configurations (4)

Maximum per socket memory bandwidth of commodity DIMM based memory subsystems

Maximum per socket serial bus limited memory bandwidth of commodity DIMM based memory subsystems:

- Low end and midrange POWER6 models (like Power 520 to specific 570 models) makes use of commodity DDR2-667 based memory subsystems with a single memory controller and two to four ports per controller.
- Each port is connected to a Memory Buffer chip via a high speed serial bus that is clocked at 4 times the memory speed and transfers 2 read and 1 write bytes per clock cycle.
- Accordingly the per socket serial bus limited memory bandwidth of a single memory controller four port system is:

1 memory controller x 4 ports x (2 read B +1 write B) x 4 x 667 MT/s = 32.0 MB/s

6.2.7 Dual memory configurations (5)

The maximum per socket memory bandwidth of commodity DIMM based memory subsystems limited by the available DIMMs is:

- In case when a Buffer chip serves a single DDR2-667 DIMM:

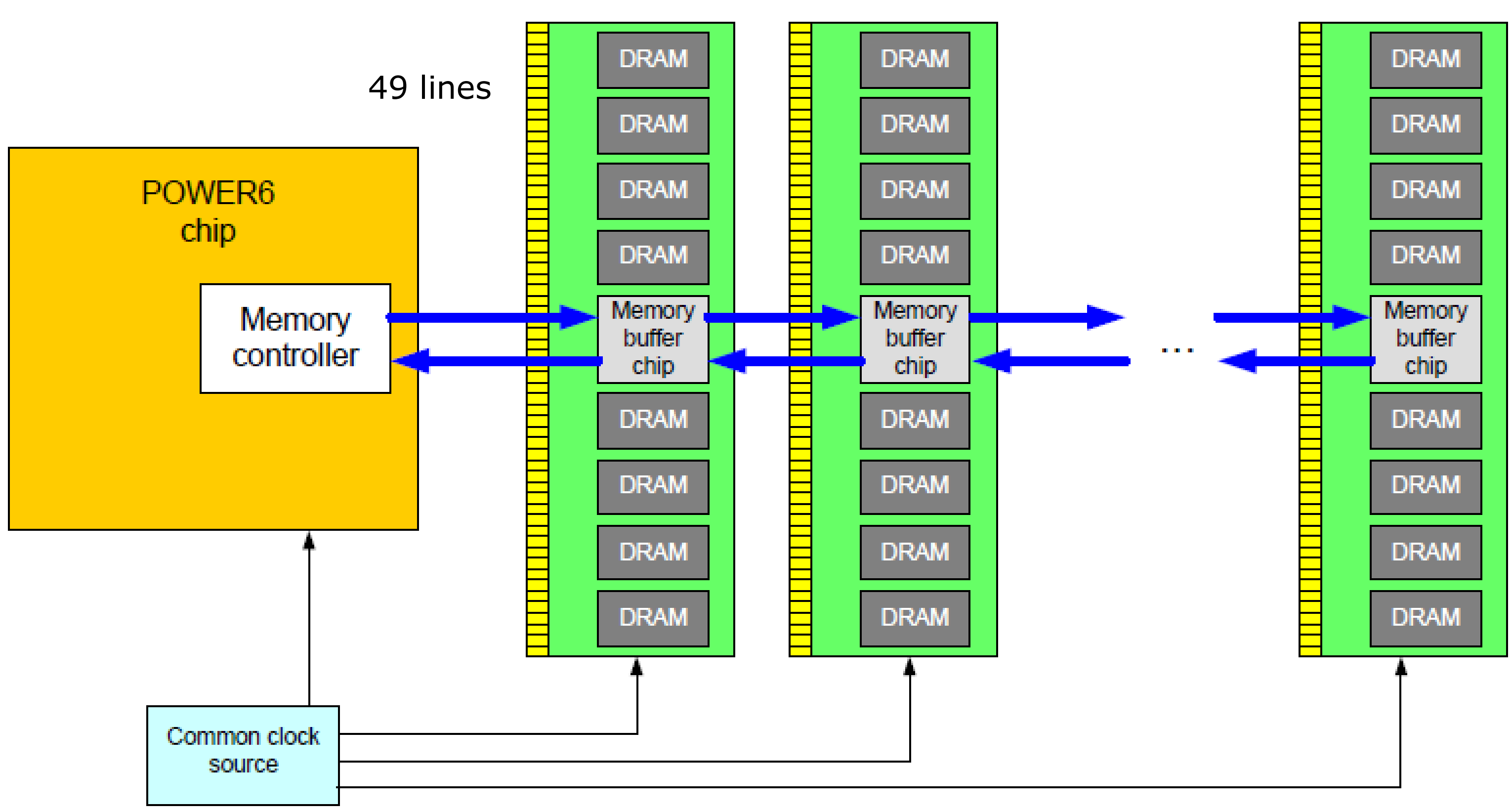
1 mem. contr. x 4 ports x 1 (DDR2-667 DIMM) x 8 B x 667 MT/s = 21.35 MB/s

- In case when a Buffer chip serves two DDR2-667 DIMMs:

1 mem. contr. x 4 ports x 2 (DDR2-667 DIMMs) x 8 B x 667 MT/s = 42.67 MB/s

b) FBDIMM based memory configuration

In contrast, high end POWER6 models are implemented with industry standard FBDIMM-667 based memory subsystems, as indicated in the next Figure.



Principle: Principle of FBDIMM based POWER6 memory implementation [34]

Industry standard FB-DIMM memory architecture

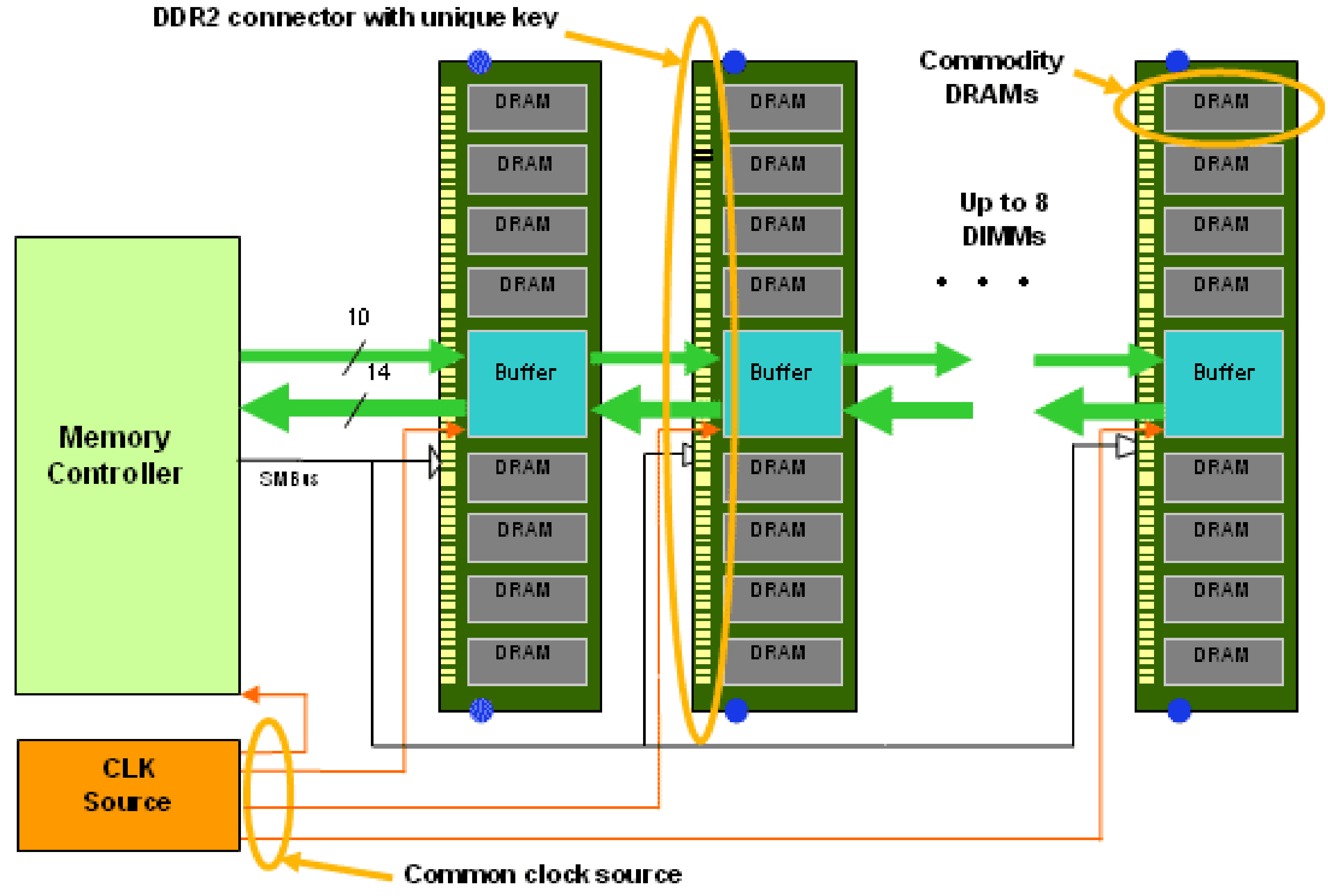


Figure: Industry standard FBDIMM memory architecture [35]

6.2.7 Dual memory configurations (8)

Implementation details of FB-DIMMs-1

- Serial transmission between the Memory controller and FBDIMM buffers (each bit needs a pair of wires)
- Number of serial links
  - 14 read lanes (2 wires each)
  - 10 write lanes (2 wires each)
- Clocked at 6 x memory data rate

  e.g. for a DDR2-667 DRAM the clock rate is: 6 x 667 MHz = 4 GHz
- Every 12 cycles (i.e. every two memory cycles) constitute a packet.
  - Read packets (frames, bursts): 168 bits (12 x 14 bits)
  - 144 data bits

    (equals the number of data bits produced by a 72 bit wide DDR2 module (64 data bits + 8 ECC bits) in two memory cycles)
  - 24 CRC bits.

6.2.7 Dual memory configurations (9)

Implementation details of FB-DIMMs-2

- Write packets (frames, bursts): 120 bits (12 x 10 bits)
  - 98 payload bits
  - 22 CRC bits.

98 payload bits

- 2 frame type bits,
- 24 bits of command,
- 72 bits for data and commands, according to the frame type,
  e.g. 72 bits of data, 36 bits of data + one command or two commands.

Commands

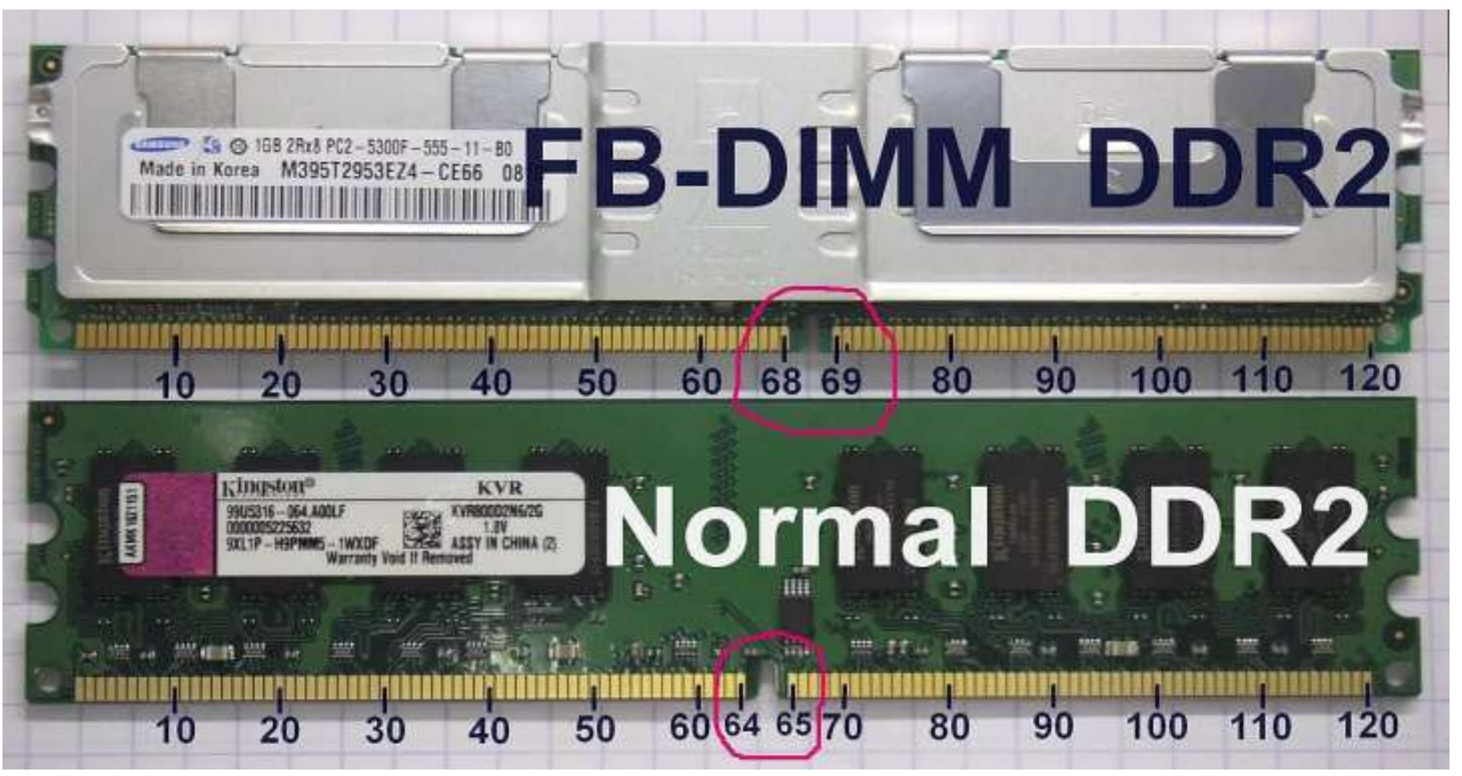
- row select, precharge, refresh, read, write etc.
- all commands include a 3-bit FBDIMM module address to select one of 8 modules.

6.2.7 Dual memory configurations (10)

Remark

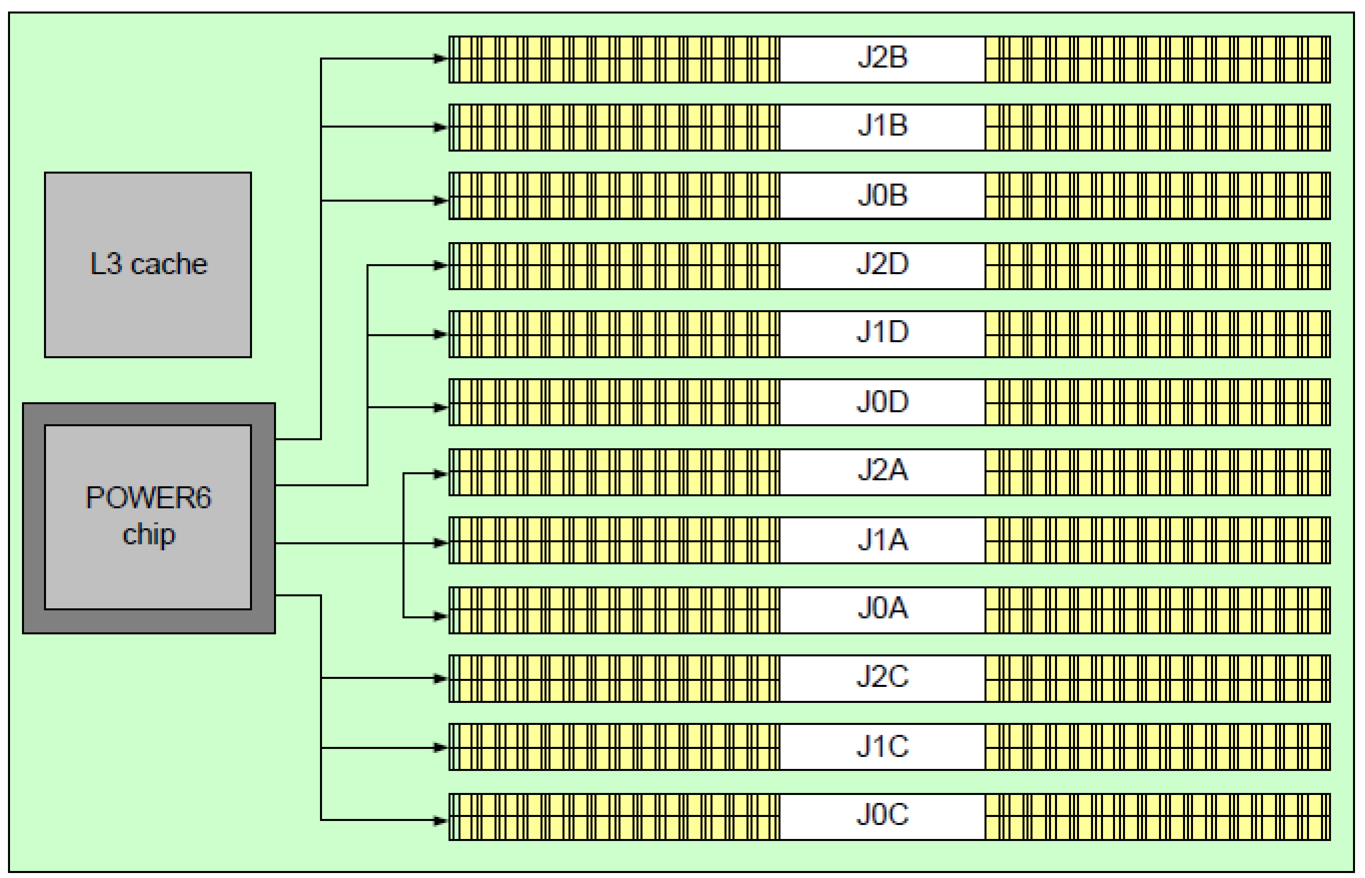
FB-DIMM memory appeared around 2006 with DDR2-667 memory chips mounted on them, but due to low demand FBDIMM manufacturers were reluctant to develop this kind of memory further on.

Physical implementation of FB-DIMM memory cards [36]



Note that FB-DIMM memory cards have the same number of pins (240) as traditional DDR2 DIMMs but they are differently notched, as indicated in the above Figure.

Layout of a POWER6 processor card with FBDIMM-based memory -1 [34]



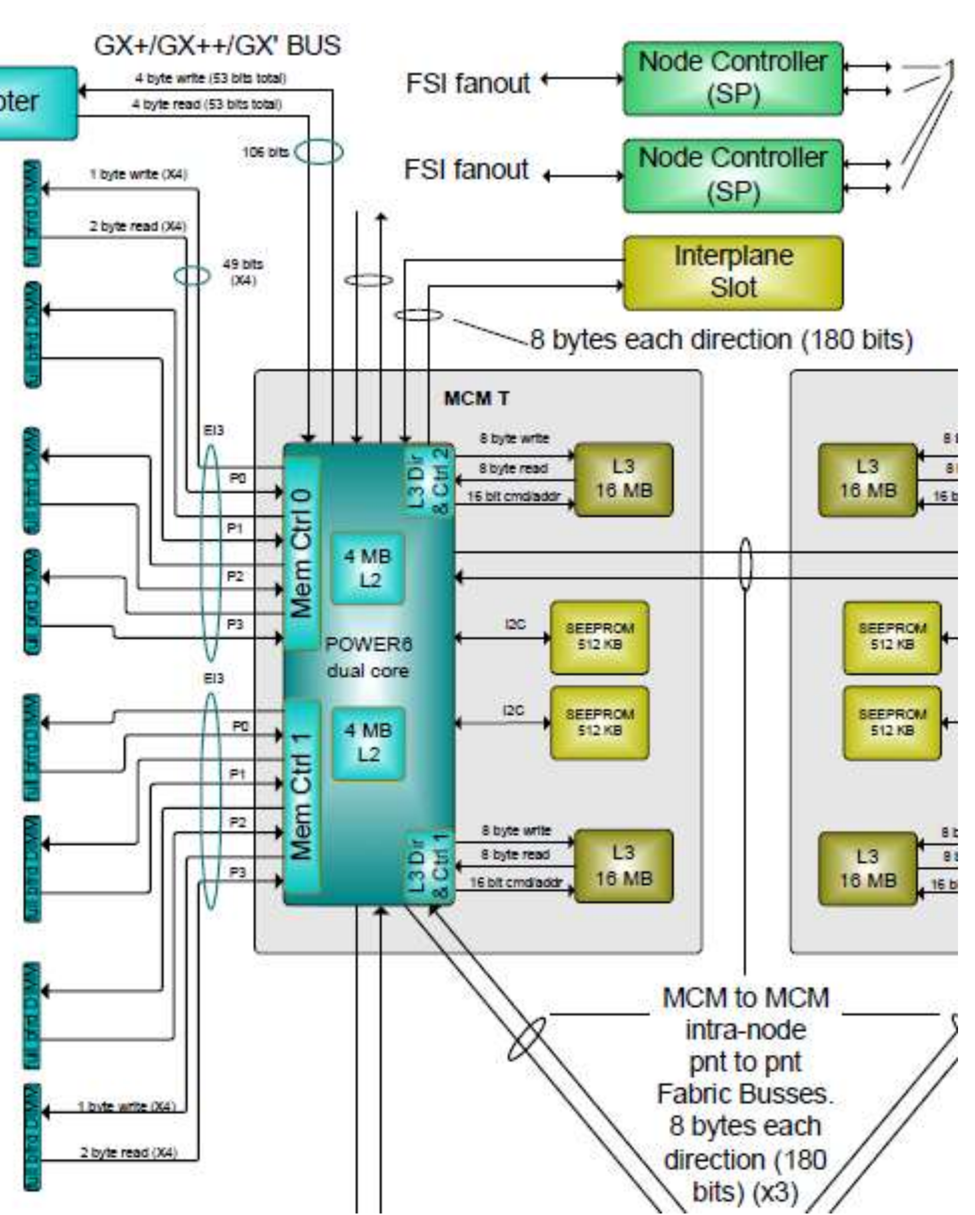
- The Figure shows the layout of an FBDIMM-667 based memory subsystem of the Power 570 server.
- The POWER6 processor is connected via a single memory controller and four ports to four Memory Buffers each serving three cascaded FBDIMM-667 memory cards.

Layout of a POWER6 processor card with FB DIMM-based memory -2 [34]

Note that the FBDIMM based memory subsystem allow 50% more memory capacity than the memory buffer based one (actually three DIMMs per port vs. two DIMMs per port), nevertheless FB-DIMM cards cost more than industry standard DDR2 modules.

Part of the block diagram of the POWER6 based Power 595 [34]

- Note that each port is connected to only a single FB-DIMM by 49 wires providing 1 byte write and 2 bytes read per cycle.
- The high speed memory interface runs at four times of the memory speed, i.e. 4 x 667 Mtransfers/s = 2.668 Gtransfers/s.



Physical implementation of the memory subsystem in the Power595

The Power595 is implemented on a number of processor Books such that a Book consists among others of 4 MCM modules (each including 4 processors) and 4x2 memory units, each with 4 FB-DIMM-667 memory cards, as seen in the Figure below.

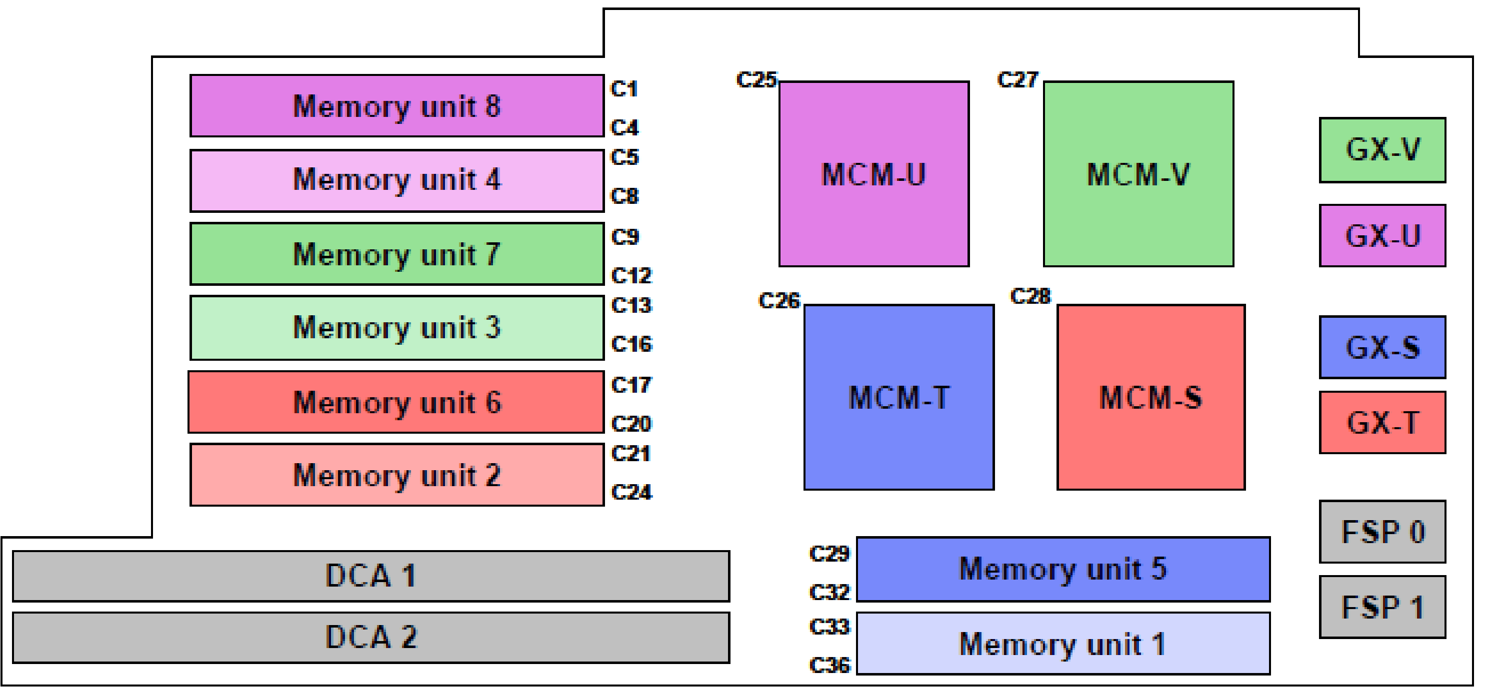
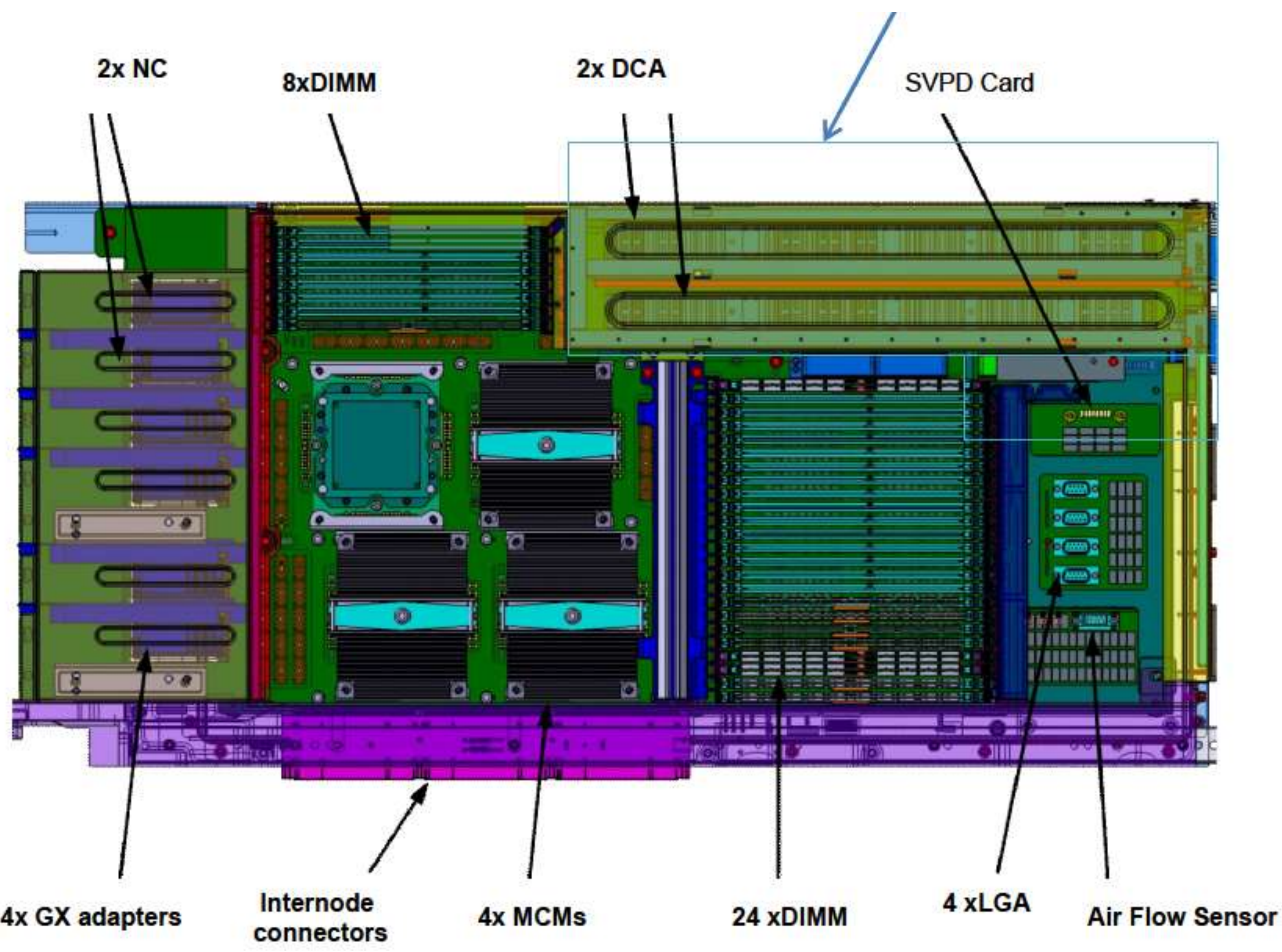


Figure: A processor Book of the Power595 [37]

Layout of a processor Book of the Power595 [37]



Example: 4x 2GB FB-DIMM-667 memory modules (#5694) used in the Power595

Picture source: ebay

Maximum per socket memory bandwidth of FB-DIMM based memory subsystems

Maximum per socket serial bus limited memory bandwidth of FBDIMM based memory subsystems:

- The memory subsystem of some midrange and high end POWER6 models (like specific Power 570 and all Power 595 models) is based on commodity, fully buffered FBDIMM-667 memory modules, as discussed before.
- A single or dual memory controllers with four ports per controller are used to attach the memory modules.
- Each port is connected to a single (Power 595) or three cascaded (specific Power 575 models) FB-DIMM-667 memory modules via a high speed serial bus that is clocked at 4 times the memory speed and transfers 2 read and 1 write bytes per clock cycle.
- Accordingly the per socket serial bus limited memory bandwidth is
  - in case of a single memory controller (specific Power 570 models):

1 memory controller x 4 ports x (2 read B +1 write B) x 4 x 667 MT/s = 32.0 MB/s

  - and in case of dual memory controllers (Power 595):

twice the single controller figure, that is 64.0 MB/s

6.2.7 Dual memory configurations (19)

Maximum per socket memory bandwidth of FBDIMM based memory subsystems limited by the available FB-DIMMs is:

- In case when there is a single memory controller with four ports and each port serves a single or three cascaded DDR2-667 FBDIMMs:

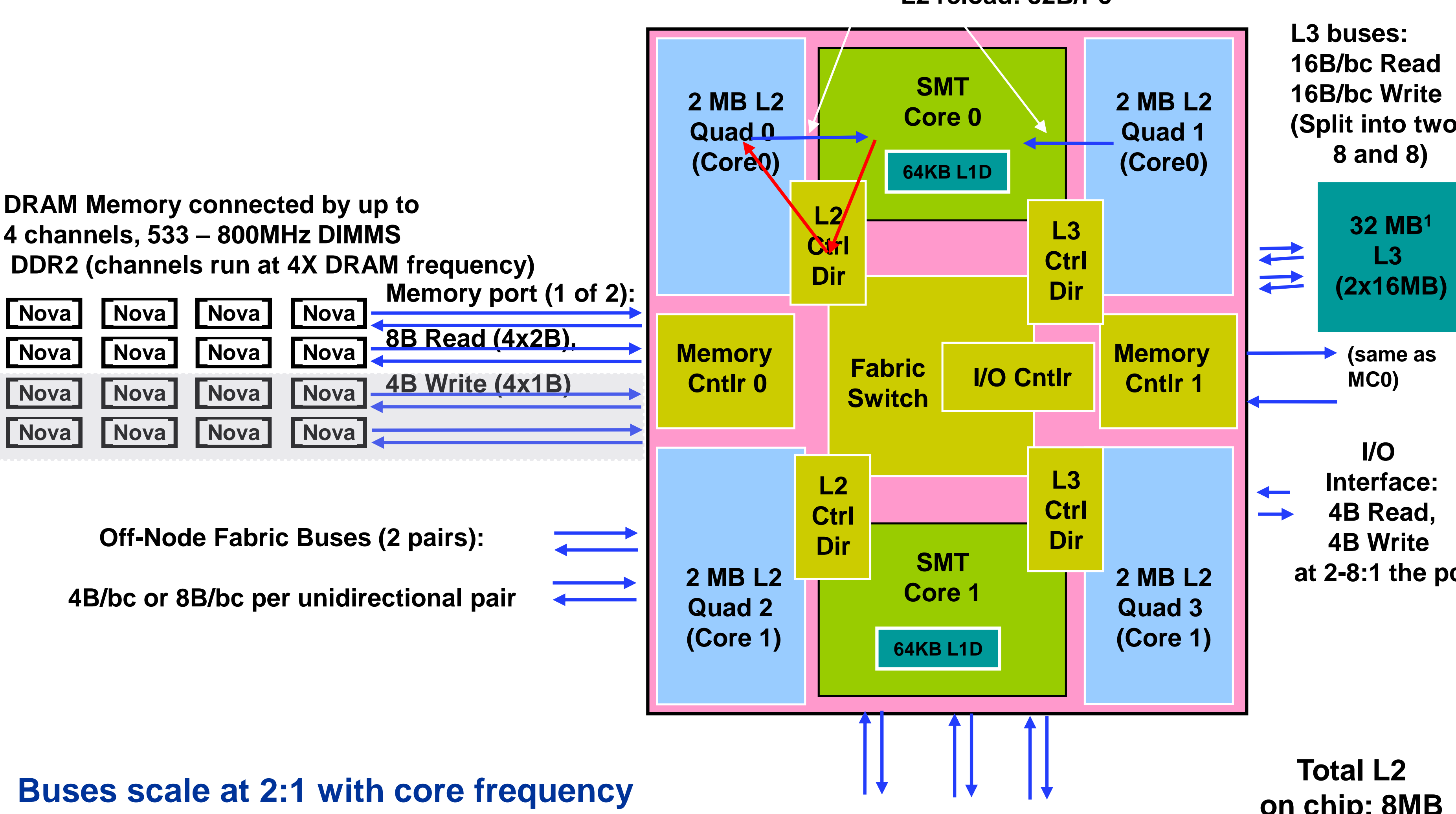
1 mem. contr. x 4 ports x 1 (DDR2-FB-DIMM) x 8 B x 667 MT/s = 21.3 MB/s

- In case when there are two memory controllers with four ports per controller and each port serves a single or three cascaded DDR2-667 FB-DIMMs:

2 mem. contr. x 4 ports x 1 (DDR2-667 FB-DIMM) x 8 B x 667 MT/s = 42.7 MB/s

These figures show that for the maximum configuration (two memory controllers, four ports) at last the serial bus limited bandwidth data are decisive for the resulting bandwidth of the memory subsystem.

I/O data widths and speeds [5]



Buses scale at 2:1 with core frequency

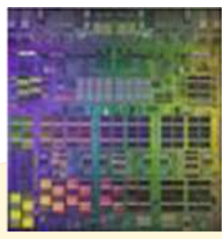
pc = processor clock
bc = bus clock
2 pc = 1 bc

[1] May be a single 32MB L3 chip with 8B buses

6.3 Key innovations of the POWER6

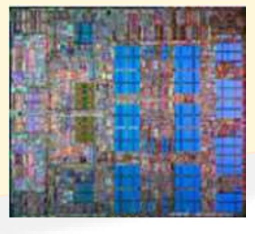
- 6.3.1 AltiVec SIMD extension
- 6.3.2 Hardware decimal floating point unit
- 6.3.3 EnergyScale architecture
- 6.3.4 Critical Path Monitors
- 6.3.5 Introduction of the Nap idle mode

6.3 Key innovations of the POWER6 (Die photos from [3])



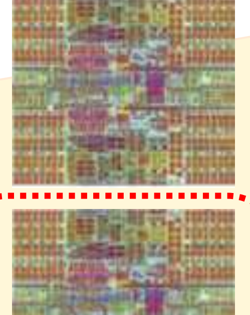
POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP



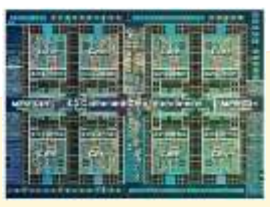
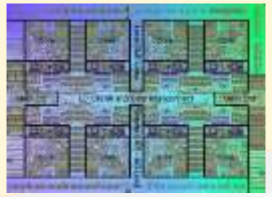
POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating



POWER6/6+
65/65 nm

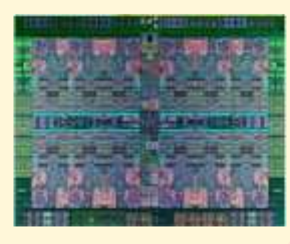
Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

POWER7/7+
*****
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

*POWER7+



POWER8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2001 | 2004 | 2007 | 2010 | 2014

6.3.1 AltiVec SIMD extension

6.3.1 AltiVec SIMD extension

Emerging of SIMD extensions -1

- In the middle of 1990's multimedia and graphics workloads running on desktops became widespread use also spurred by the proliferation of internet.
- Multimedia and graphics workloads make however, extensive use of unsigned or signed 8-bit or 16-bit integer data as well as single-precision FP data, as indicated in the next Table.

	Data type				
	8-bit integer		16-bit integer		Single-precision float
Task	**Unsigned**	**Signed**	**Unsigned**	**Signed**	**Signed**
Video		Low quality		High quality	
Audio				Low quality	High quality
Image processing		Low quality		High quality	
3D graphics				Low quality	High quality
Speech recognition				Low quality	High quality
Communication	Crypto		Crypto		
Media mining					High quality

Table: Data types often used in various media and graphics workloads [38]

Emerging of SIMD extensions -2

Consequently, there is a vast potential for speeding up multimedia processing by introducing vector processing, termed also as SIMD (Single Instruction Multiple Data) processing, meaning that the same operation is carried out on elements of two or three data vectors in parallel, like indicated below for the latter case.

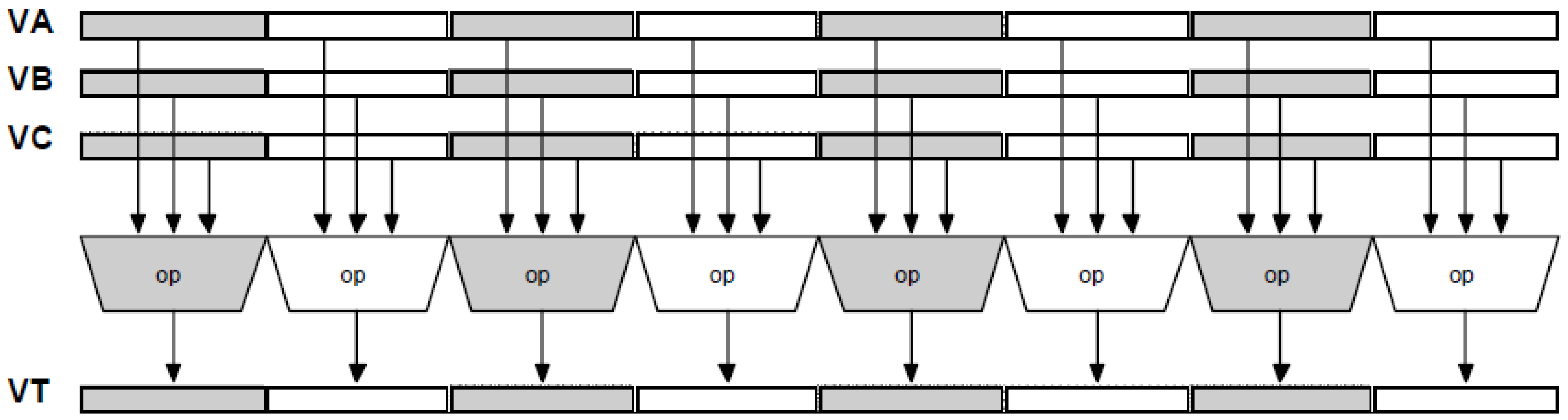


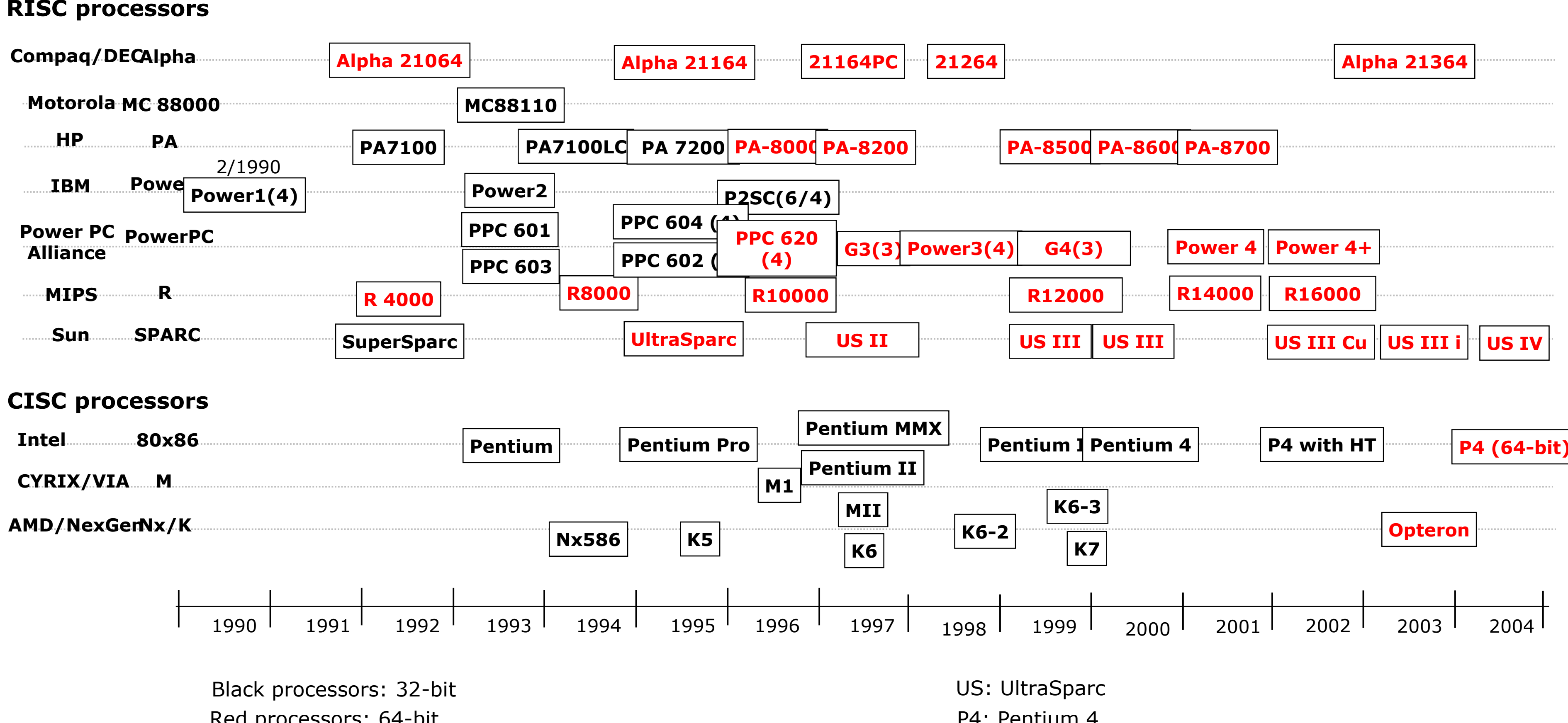
Figure: Performing the same operation on elements of three vectors in parallel by means of a SIMD (Single Instruction Multiple Data) instruction [39]

6.3.1 AltiVec SIMD extension (3)

Emerging of SIMD extensions -3

- The raising need for speeding up multimedia and graphics processing coincided with the 32 bit to 64 bit transition of RISC processors in the beginning of the 1990’s, as show in the next Figure.

Transition of desktop processors from 32 bit to 64 bit



Black processors: 32-bit

Red processors: 64-bit

US: UltraSparc

P4: Pentium 4

Emerging of SIMD extensions -4

- Obviously, 64 bit wide data offer already a straightforward opportunity to vector processing since 64 bit data words can be interpreted as vectors with multiple data elements, say of 8x8 or 4x16 or 2x32 bit FX data or 2x32 bit single precision FP data.
- The coincident need and possibility for vector processing lead to the introduction of SIMD ISA extensions first in RISC processors in the middle of the 1990 followed by CISC processors, as discussed next and indicated in the next Tables.

Emergence of SIMD extensions in RISC processors -1

Ven dor	SIMD extension	ISA	Processor	FX/FP	No. and type of available registers	Word length	Release date of SIMD extension
Sun	VIS	SPARC-V9	UltraSparc-1	FX	32 (shared with FP)	64	12/1994
hp	MAX-1	PA-RISC 1.1	PA7100-LC	FX	32 (shared with FX)	32	01/1994
hp	MAX-2	PA-RISC 2.0	PA-8000	FX	32 (shared with FX)	64	11/1995
MIPS	MDMX	MIPS V	R10000	FX/FP (FP in MIPS V)	32 (shared with FP)	64	10/1996
DEC	MVI (Minimalist impl.)		Alpha 21164PC	FX	32 (shared with FX)	64	10/2006

Emergence of SIMD extensions in RISC processors -2

- As the Table above indicates, SIMD extensions make typically use of already available FX or FP register space and provide 32 64-bit registers.

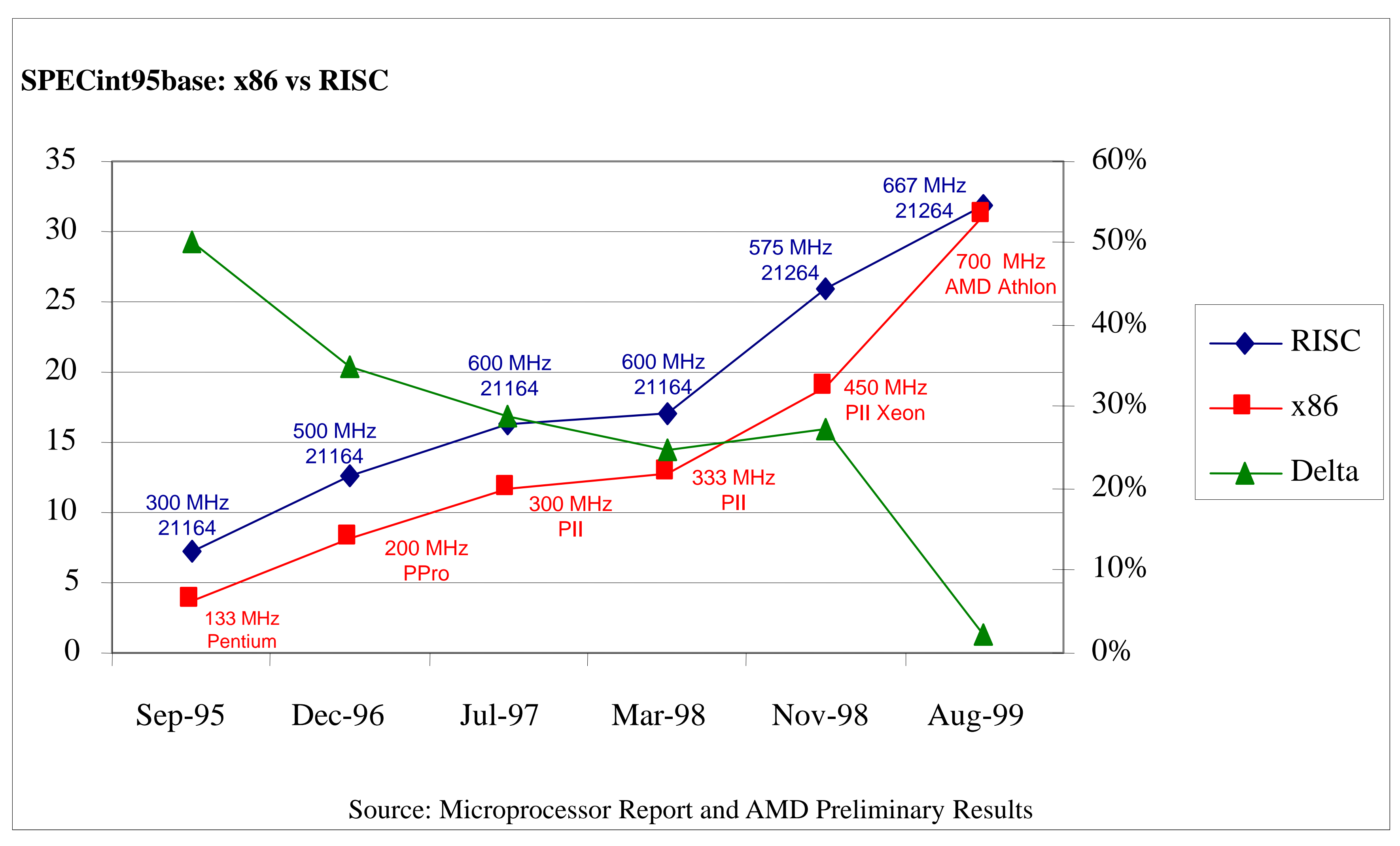
  It seems that it is only a particular design decision whether the FX or the FP register space is shared with SIMD processing.
- Introduced SIMD extensions provided usually only FX SIMD operations.
- Accordingly, these processors are called sometime 2.5G superscalars, indicating that FP SIMD processing is still missing.
- Here we note that MIPS already introduced single precision FP SIMD processing by enhancing the basic MIPS V ISA by a small set of instructions to provide dual single-precision FP operations, rather than implementing FP SIMD processing as part of the SIMD extension.

Emergence of SIMD extensions in RISC processors -3

Concerning the future of SIMD extensions in RISC processors we point out that in the end of the 1990's the performance of x86 CISC processors reached that of RISC processors, as demonstrated in the next Figure.

6.3.1 AltiVec SIMD extension (9)

CISC processors catching up with CISC processors in performance between 1995 and 2000 [40]



Emergence of SIMD extensions in RISC processors -4

- As a consequence, around 2000 processor manufacturers typically cancelled their RISC developments and most RISC processors became ousted from the market (except Sun's processors (at least for a few years) and IBM's POWER line).
- Thus, in a couple of years most RISC lines indicated in the above Table disappeared.

Emergence of SIMD extensions in x86 processors -1

The widespread use of multimedia forced also CISC processor vendors to sped up multimedia applications in their machines by introducing SIMD extensions to the x86 ISA, as indicated in the next Table.

Emergence of SIMD extensions in x86 processors -2

Vendor	SIMD extension	Introduced in the processor	FX/FP	No. and type of available registers	Word length	No. of instr.	Release date of SIMD extension
Intel	MMX	Pentium MMX	FX	8 (shared with FP)	64	56	01/1997
AMD	+3DNow!	K6-2	FX/FP	8 (shared with FP)	64	21	02/1998
Intel	+SSE (KNI)	Pentium III Katmai	FX/FP	8 XMM (dedicated to SIMD)	128	70	02/1999
AMD	+Enhanced 3DNow!	Athlon (K7)	FX/FP	8 (shared with FP)	64	24	06/1999
Intel	+SSE2	Pentium 4 Willamette	FX/FP	8 XMM (dedicated to SIMD)	128	144	11/2000
AMD	+SSE	Athlon-XP	FX/FP	8 XMM (dedicated to SIMD)	128	70	10/2001
Intel	+SSE3	Pentium4 Prescott	FX/FP	8 XMM (dedicated to SIMD)	128	13	02/2004

Emergence of SIMD extensions in x86 processors -3

- However, in the middle of the 1990's CISC processors had still a word length of 32 bit (and remained so until 2003 when AMD introduced their Opteron line and Intel followed suit with their 64-bit extension for the Pentium 4 in 2004).
- Thus when Intel introduced their x86 SIMD extension in 1/1997 it was plausible for their designers to use the 8x80 bit FP register space for processing 64 bit long multimedia data, as seen below.

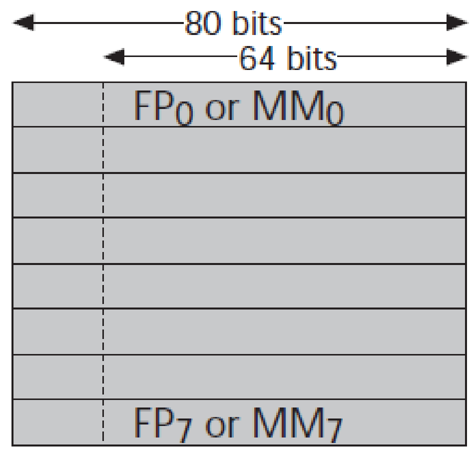


Figure: Use of the FP register space for 64 bit multimedia data in Intel's MMX ISA extension [41]

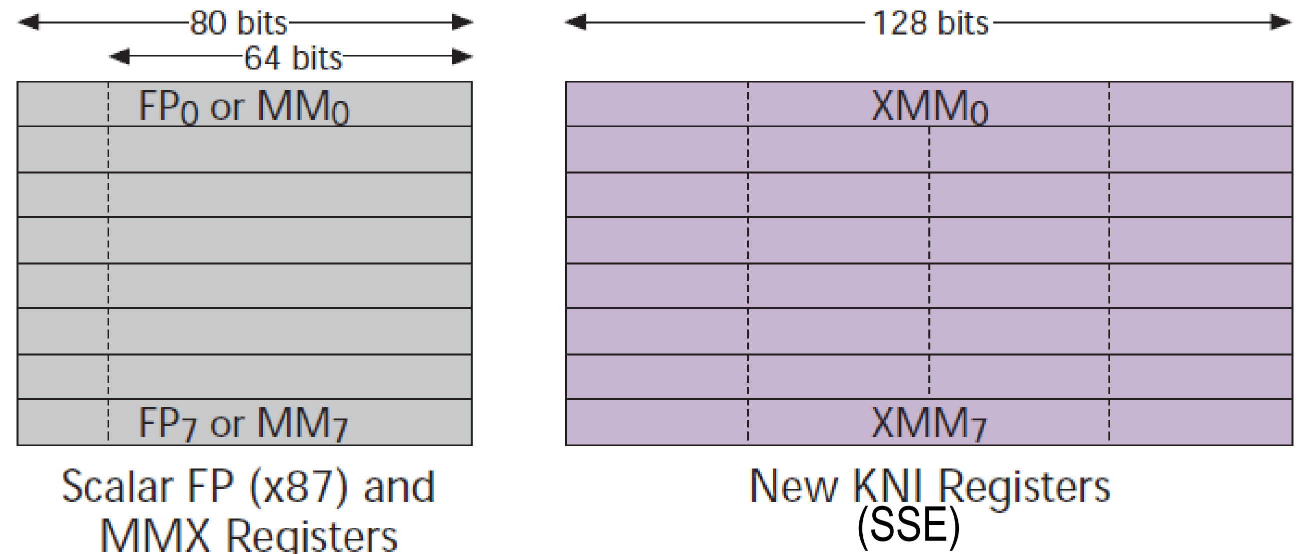
Here we point out that the MMX SIMD extension covered only FX data and associated instructions.

Emergence of SIMD extensions in x86 processors -4

- As the Table indicates, shortly after Intel's MMX introduction AMD enhanced this FX SIMD extension by FP SIMD capability (to process dual 32-bit single precision FP data in parallel) and designated the new SIMD extension including 21 instructions as 3DNow! to point out its support for graphics processing.
- Here we note that subsequently AMD enhanced further on their SIMD extension by 24 new instructions and called this enhanced SIMD extension as Enhanced 3DNow!.
- Meanwhile, in their first Pentium III core (termed Katmai) Intel introduced a new SIMD extension built from the scratch, termed as SSE (for Streaming SIMD Extension or KNI for Katmai New Instructions).

  SSE covers both FX and FP SIMD processing and defines a new data space of 8x128 bit dedicated for SIMD operations, as shown in the next Figure.

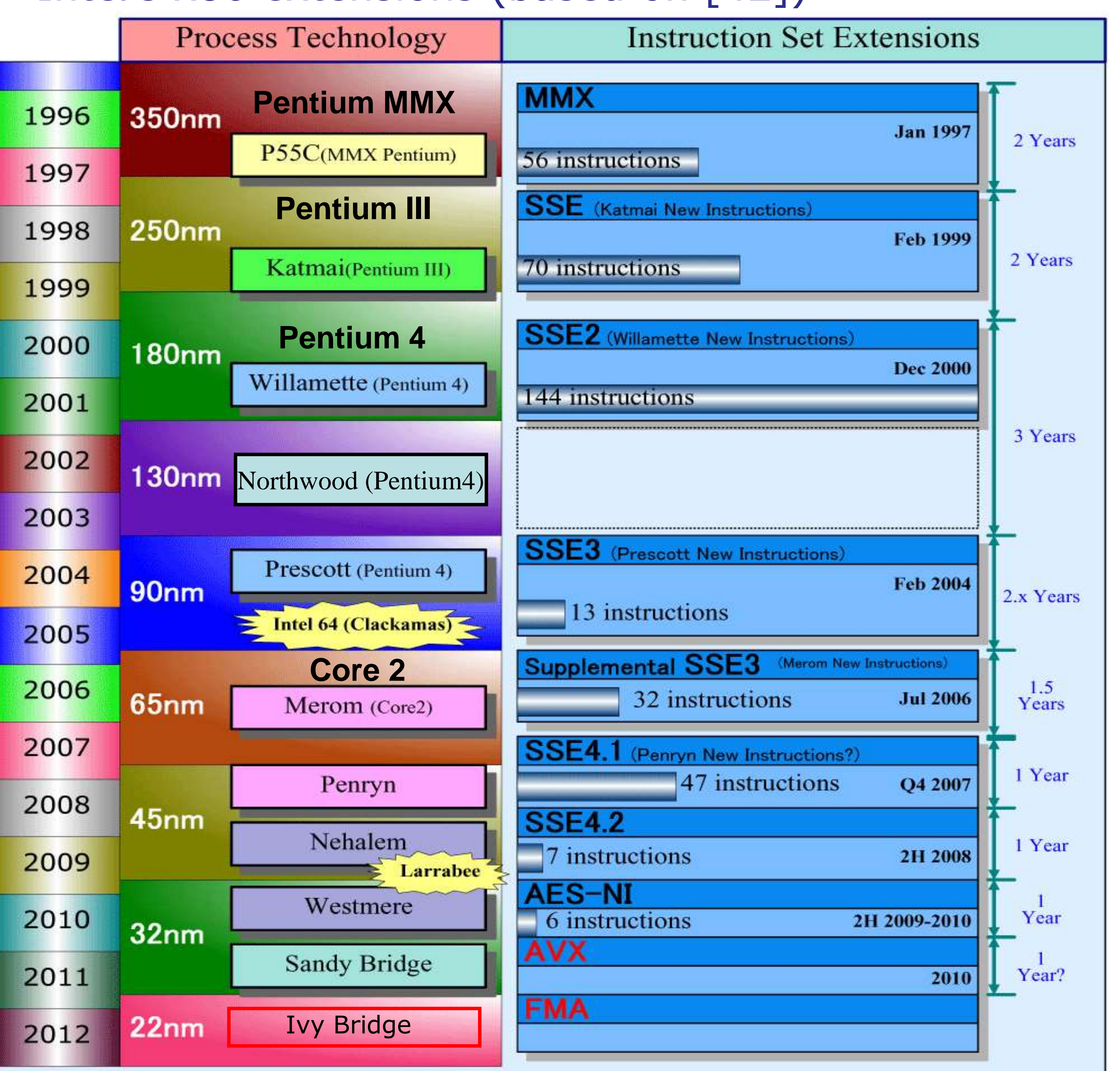
The register spaces used in MMX and SSE [41]



Emergence of SIMD extensions in x86 processors -5

- We note that subsequently Intel enhanced multiple times their x86 ISA extension, first to enhance SIMD processing, later on also for other purposes e.g. for supporting EAS (Enhanced Encryption Standard) processing.
- The next Figure shows this.

Intel's x86 extensions (based on [42])



For SIMD processing available register space

8 MM registers (64-bit), aliased on the FP Stack registers

8 XMM registers (128-bit)

16 XMM registers (128-bit) in 64-bit CPUs in 64-bit mode

16 YMM registers (256-bit)

6.3.1 AltiVec SIMD extension (18)

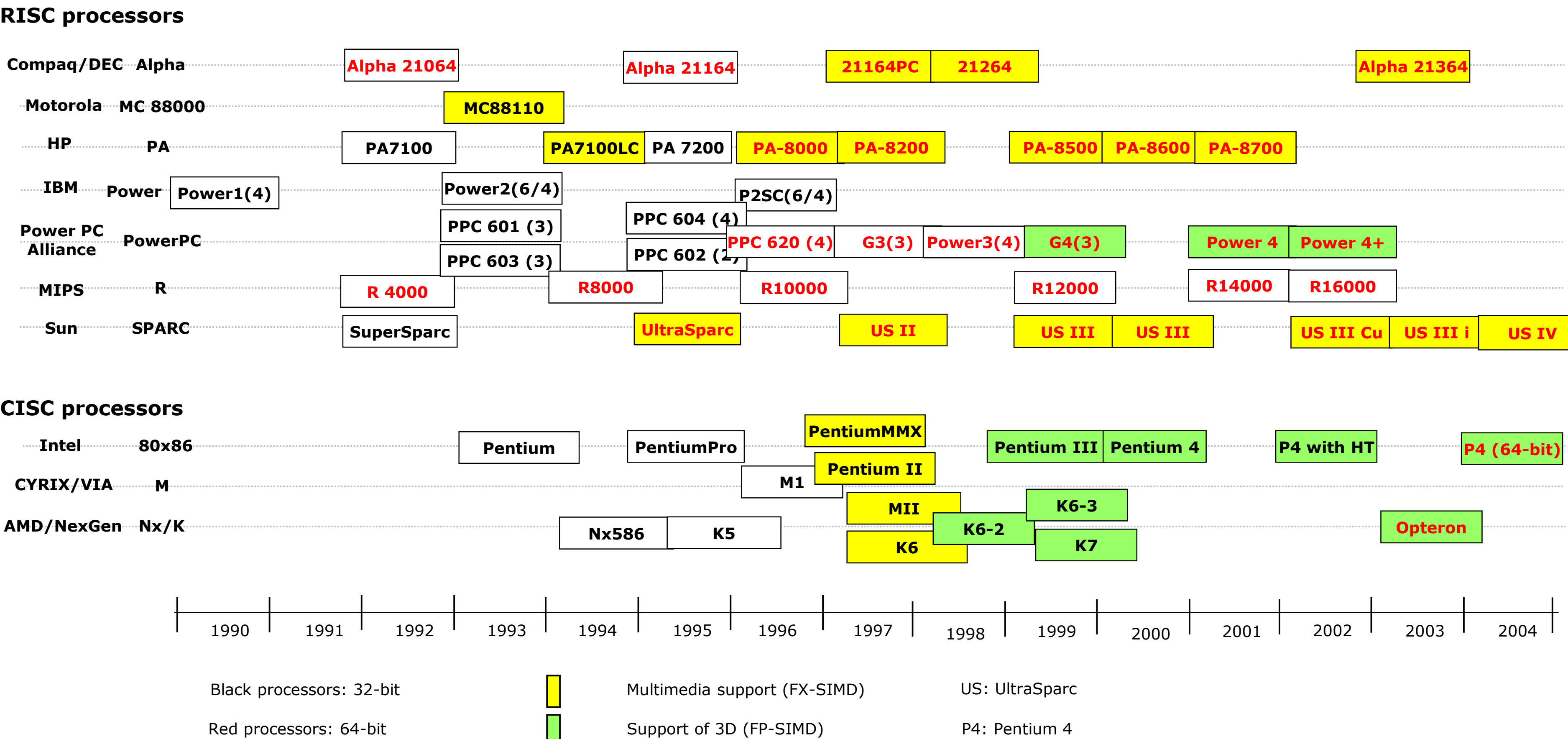
AMD's further involvement in SIMD extensions of their processor lines

- Due to Intel's dominance in the processor market AMD was forced to implement Intel's x86 ISA extensions as well.

  This happened first in AMD's Athlon-XP line in 2001, thus this processor supported beyond AMD's native SIMD extensions also Intel's SSE.

- AMD's subsequent processors tried to follow Intel's ISA extensions, nevertheless this occurred usually with a time lag.
- Finally, because of negligible interest, AMD cancelled supporting their own 3DNow! and further ISA extensions in 2010.

Overview of the emergence of FX and FP SIMD ISA extensions



6.3.1 AltiVec SIMD extension (20)

Emergence of the AltiVec extension in PowerPC and POWER processors -1

- AltiVec is a second generation SIMD extension, co-designed by Motorola, IBM and Apple in the middle of the 1990's.
- The name "AltiVec" was trademarked by Motorola (now by Freescale, the formerly Semiconductor Products Division of Motorola) so both IBM and Apple refer to AltiVec by their own designations (VMX and Velocity Engine, respectively).
- AltiVec was introduced relatively late (in 1998) into the PowerPC line by Motorola, followed by IBM and Apple a couple of years later.
- Finally IBM has decided to introduce the AltiVec SIMD extension also in their POWER line, relatively late, only starting with the POWER6 model in 2007.

Emergence of the AltiVec extension in PowerPC and POWER processors -2

Vendor	SIMD extension	Introduced in the processor	FX/FP	No. of avail. registers	Word length	No. of instr.	Release date of SIMD extension
Motorola (Freescale)	AltiVec	PowerPC MPC7400 (G4)	FX/FP	32 (dedicated to SIMD)	128	162	05/1998
Apple	Velocity Engine	PowerPC G5	FX/FP	32 (dedicated to SIMD)	128	162	06/2003
IBM	VMX	PowerPC 970	FX/FP	32 dedicated to SIMD)	128	162	10/2002
IBM	AltiVec	POWER6	FX/FP	32 (dedicated to SIMD)	128	162	05/2007

Main features of the AltiVec extension [38]

It provides

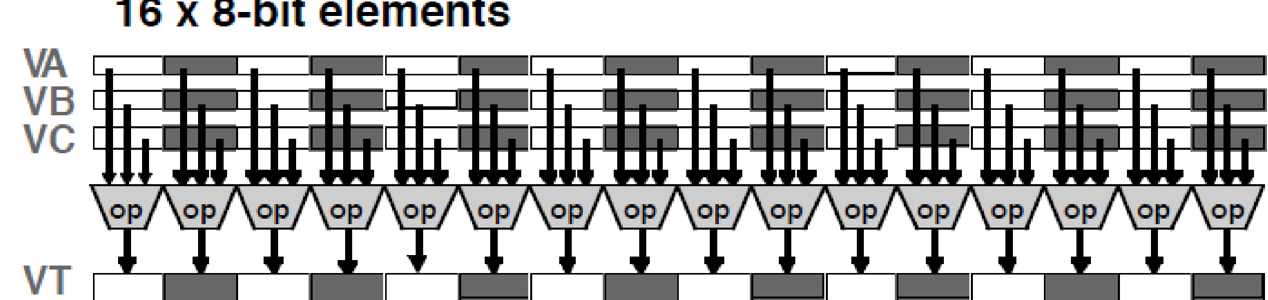
- fixed-length 128-bit vectors, each comprising four, eight, or 16 data elements,
- a separate vector register file with 32-registers, each capable of holding one 128-bit vector,
- vector-elements belong to the data types of 8-, 16-, and 32-bit signed or unsigned integers, as well as IEEE single-precision FP data,
- 162 new SIMD-style instructions optimized for digital signal processing, with saturation or modulo arithmetic,
- a four-operand, nondestructive instruction format (three sources, one destination), as seen below.

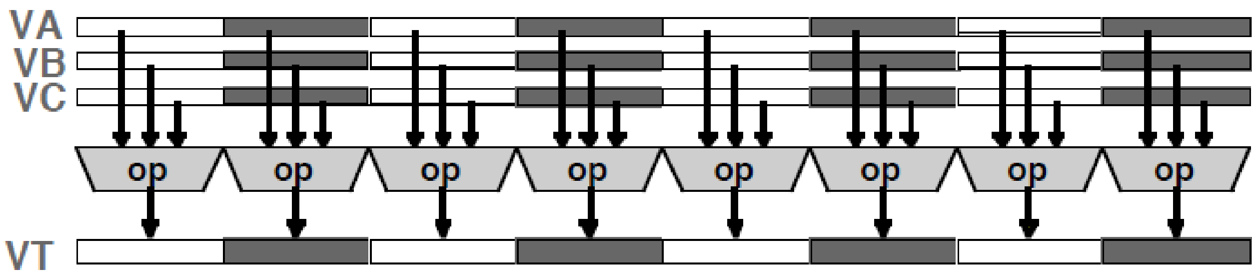
6.3.1 AltiVec SIMD extension (23)

Examples of three operand FX SIMD operations [129]

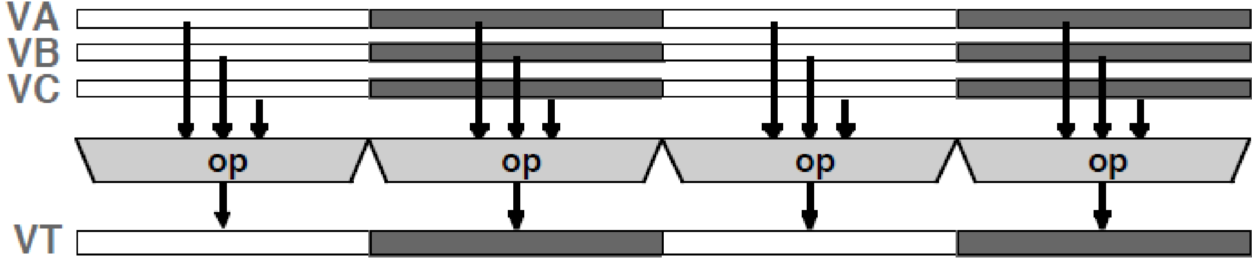
Operations include:

- Integer arithmetic
- FP arithmetic
- Memory access
- Conditional
- Logical
- Shift, Rotate
- Min, Max
- Saturation options

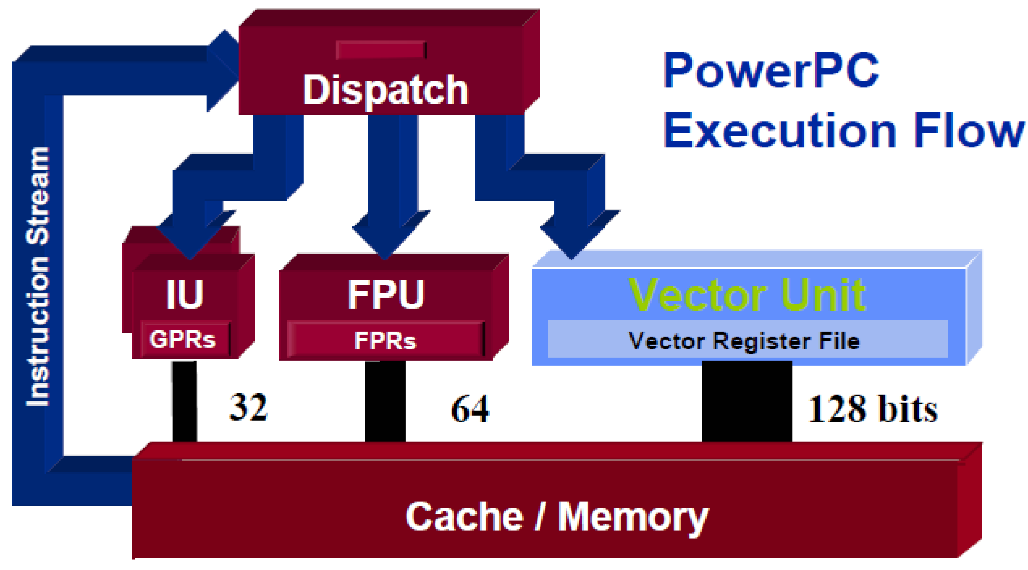




4 x 32-bit elements



Example AltiVec implementation using a dedicated Vector Unit [39]



- Concurrent with PowerPC integer and floating-point units
- Separate, dedicated 32 128-bit vector registers
- Approximately 11% of the silicon area

6.3.2 Hardware decimal floating-point unit

6.3.2 Hardware decimal floating-point unit

The need for introducing hardware decimal point processing -1

- According to [43] commercial databases have more than half of their numeric data in decimal format.
- Unfortunately, decimal calculations cannot be directly implemented using binary FP data representation because fractions such as 0.1 cannot be represented exactly.

  Instead, decimal FP operations are usually emulated by binary integer (fixed-point) processing.
- Binary integer implementations of decimal FP calculations keep the exponent and the coefficient of the decimal FP representation separately and operate on them independently.
- Binary integer data representation has its deficiencies due to its limited range and the difficulty in scaling and rounding.
- In addition, emulating decimal FP processing by binary FP processing (while using appropriate program libraries) slows down significantly decimal data calculations.

6.3.2 Hardware decimal floating-point unit (2)

The need for introducing hardware decimal point processing -2

As a consequence, in the course of the evolution of servers targeting also the financial sector it became imperative to introduce standardized decimal FP data representation and operations as well as to support decimal FP calculations directly by hardware.

6.3.2 Hardware decimal floating-point unit (3)

Standardization of binary and decimal FP data representations

- Binary floating-point (BFP) data representation became standardized in 1985 by the IEEE 754-1985 standard.
- More than 20 years later IEEE accepted the IEEE 754-2008 standard which is a major revision and extension of the IEEE-754-1985.

  The most significant extension of the new standard is that it covers both BFP and DFP (decimal floating-point) data representations.
- As far as the BFP data representation concerns, the 754-2008 supersedes the 754-1985 standard.
- Subsequently, we give a brief overview of the principles of the BFP and DFP data representations.

6.3.2 Hardware decimal floating-point unit (4)

BFP data representation according to the IEEE 754-1985 standard -1 [44]

According to the 754-1985 standard data will be represented in BFP (binary floating-point) format as follows:

Data will be first expressed in a binary floating-point notation by a triplet

$$\{s, b, e\}$$

in the form

$$(-1)^s (b_0\ b_1\ b_2 \ldots b_{p-1})\ 2^e$$

such that

$s \in \{0.1\}$ is the sign and
$b = (b_0\ b_1\ b_2 \ldots b_{p-1})$ is the binary mantissa with $b \in \{0,1\}$
e: binary exponent (any integer between e_{min} and e_{max}).

As an example the decimal number 7.25 can be expressed as

$$(-1)^2 \times 111.01 \times 2^0$$

BFP data representation according to the IEEE 754-1985 standard -2 [44]

Subsequently, the triplet {s, b, e} will be encoded in one the basic BDF formats

- single precision (32 bit) BFP or the
- double precision (64 bit) BFP

or into one of the extended BSD formats according to given rules, not detailed here.

For details we refer to the related literature [44].

DFP data representation according to the IEEE 754-2008 standard -1 [45]

The IEEE 754-2008 standard defines beyond the BFP formats also the decimal floating-point (DFP) formats.

The principle of the DFP data representation is as follows:

Let's express decimal floating-point data first by a triplet

$$\{s, d, e\}$$

in the form

$$(-1)^s \, (d_0 \; d_1 \; d_2 \ldots d_{p-1}) \; 10^e$$

such that

$s \in \{0.1\}$ is the sign
$d = (d_0 \; d_1 \; d_2 \ldots d_{p-1})$ is the decimal mantissa with $d \in \{0,1,2,3,4,5,6,7,8,9\}$
e: binary exponent (any integer between e_{min} and e_{max}).

As an example the number 7.25 can be expressed as

$$(-1)^2 \times 725 \times 10^{-3}$$

DFP data representation according to the IEEE 754-2008 standard -2 [45]

Subsequently, the triplet {s, b, e} will be encoded in one of three basic DDF formats:

- short (single precision) 32 bit DFP or the
- long (double precision) 64 bit DFP or
- extended (quad precision) 128 bit DFP.

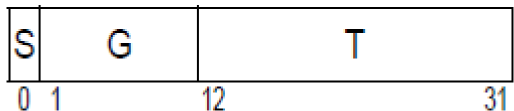
Here we note that the short (single precision) 32 bit DFP is supported only by the conversion operation to and from long DFP.

Encoding into one of the above DSD formats will be performed according to precise rules, detailed in the related literature (see e.g. [45]) but not discussed here.

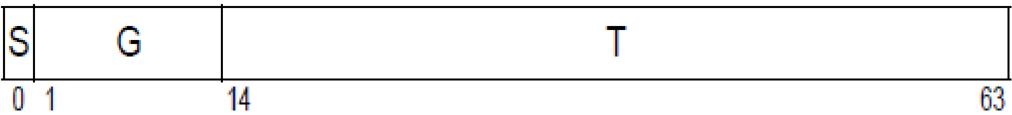
Nevertheless, in order to give a glimpse into the DFP data representation, the next Figure and Table show supported DFP data formats and related key parameters.

Supported DFP formats [45]

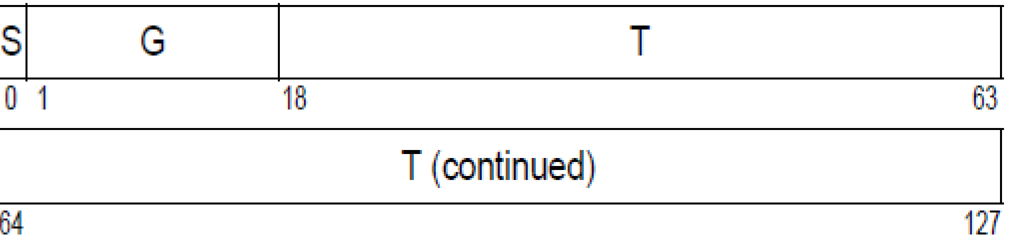
DFP short format



DFP long format



DFP extended format



Key parameters of the standardized DFP data representation [45]

	Format		
	DFP Short	DFP Long	DFP Extended
Widths (bits):			
Format	32	64	128
Sign (S)	1	1	1
Combination (G)	11	13	17
Trailing Significand (T)	20	50	110
Precision (p) (digits)	7	16	34
Magnitude:			
Maximum normal number (N_{max})	$(10^7 - 1) \times 10^{90}$	$(10^{16} - 1) \times 10^{369}$	$(10^{34} - 1) \times 10^{6111}$
Minimum normal number (N_{min})	1×10^{-95}	1×10^{-383}	1×10^{-6143}
Minimum subnormal number (D_{min})	1×10^{-101}	1×10^{-398}	1×10^{-6176}

6.3.2 Hardware decimal floating-point unit (10)

Hardware implementation of DFP processing

- 4/2007: IBM announced that their z9 line provides hardware support to IEEE 754R compliant DFP processing.

  The IEEE 754R became later the IEEE 754-2008 standard.

  This was actually an IEEE 754E compliant firmware, i.e. microcode controlled implementation accelerated by the BCD unit.

  (The BCD Unit (Binary Coded Decimal) Unit processes decimal integers stored in BCD form).
- 5/2007: IBM launched the POWER6 that included an IEEE 754-2008 compliant hardware DFU (Decimal FP Unit).
- 3/2008: IBM launched the z10 line that included also a 754-2008 compliant DFU unit.
- Subsequent models of both the POWER and z lines continued to support DFP processing in hardware.

6.3.2 Hardware decimal floating-point unit (11)

Register usage in DFP processing in the POWER6

The DFU (Decimal FP Unit) shares the 32 floating-point registers (FPRs) and the related FP status and control register (FPSCR) available per thread with the DFU (Binary FP Unit).

Remark

As the POWER line implements register renaming (except of the POWER6) per thread more FP unified architectural and rename registers are needed than FP architectural registers (32).

Actual per thread numbers are shown in the Table below.

Processor line	No. of Integer registers/thread	No. of FP registers/thread
POWER4 no SMT	80	72
POWER5	120	120
POWER6	120	120

Table: No. of unified architectural and rename registers provided per thread in select POWER models [21], [46]

6.3.3 EnergyScale Architecture

6.3.3 EnergyScale Architecture -1 [47]

EnergyScale has a two layer implementation covering both

- the system management level and
- the processor level.

At the system management level EnergyScale is embedded into IBM's system management concept that is implemented by means of a

- Hardware Management Console (HMC) and
- a Flexible Service (or Support) Processor (SSP),

as indicated below.

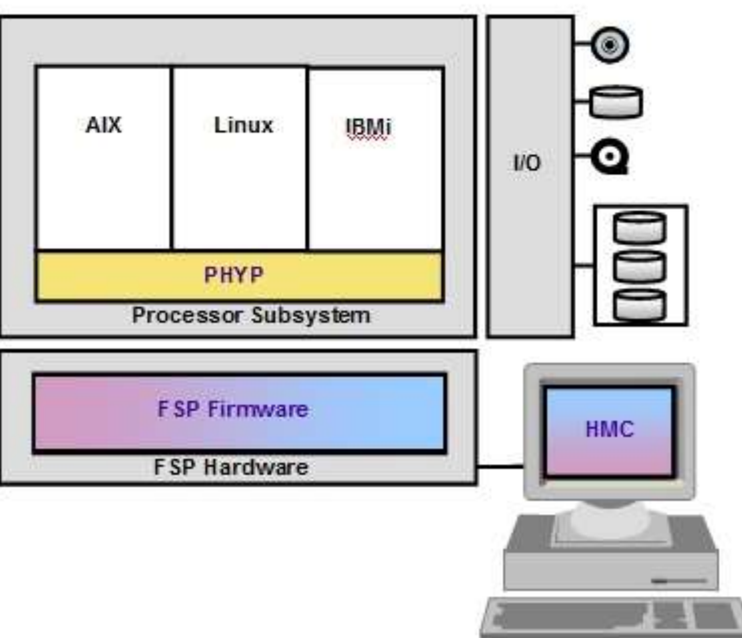


Figure: IBM's system management concept in the POWER line [47]

6.3.3 EnergyScale Architecture (2)

EnergyScale Architecture -2 [47], [48]

- The HMC provides a user interface that allows to configure and manage the platform, including the temperature and power management.
- The FSP is responsible for booting the system and monitoring the running system, including power and temperature control.
- As far as the power and temperature management is concerned the HMC/FSP subsystem has two basic functions:
  - it sends data-center and system policy (concerning also power and thermal management) directives downwards to the processor level, and
  - relays low-level (i.e. processor level) power and thermal information upward to the customer.

EnergyScale Architecture -3 [47], [48]

EnergyScale may cover either a multi processor configuration, such as a blade center or a tower configuration or a stand alone single processor server.

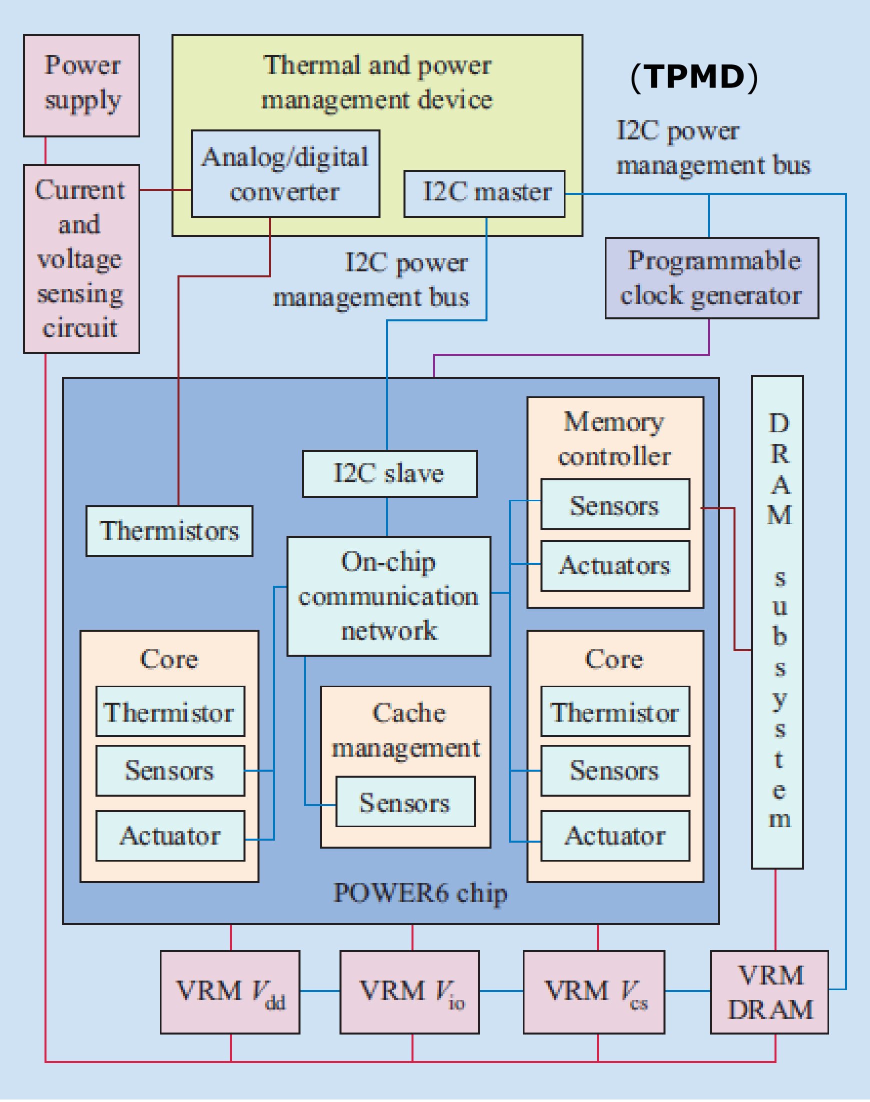
In both cases EnergyScale includes a plug-in card, called the Thermal and Power Management Device (TPMD), see below.

6.3.3 EnergyScale Architecture (4)

EnergyScale Architecture [48], [49]

(TPMD)
Power in
Microprocessor H8-2168
Debug
Filter logic
TPMD-provided clock; ROS
TPMD
Power measurement
FSP
Base clock gen
VRM array
Core 0
Core 1
POWER6 chip
Core 0
Core 1
POWER6 chip
VRM array
Rack-mounted reference implementation

Figure: Prototype HV8 board for EnergyScale covering two POWER6 processor chips



EnergyScale management for a Single chip POWER6 chip

Principle of operation

- In both the multiprocessor and single processor configurations the TPMD card includes a microcontroller e.g. a Hitachi's H8-2168) that runs the needed power and performance management firmware.
- In both cases the FSP directs the microcontroller of the TPDM card to perform power and thermal management according to the policy set by the customer.

  (Here we note that the Figure addressing the single chip configuration does not show the FSP unit.
- The TPMD card monitors the power consumption, temperature and performance related data of the processor or processors and the memory and sets the core clock and chip voltages (for each voltage domain) according to the given policy, as seen in the Figure.
- We note that voltage domains are parts of the processor chip that are supplied by the same voltage (like both cores, the DRAM or IO).
- Subsequently, we give some insight into the power and temperature management of the single processor configuration, based mainly on [48] and [49].

6.3.3 EnergyScale Architecture (6)

Collecting input data

Power and temperature management is based on a large set of real-time measurements.

The data captured includes

- temperature data
- critical path monitor data
- processor core and memory activity data
- power consumption data.

6.3.3 EnergyScale Architecture (7)

Capturing temperature data

- The POWER6 processor provides two types of on-chip temperature monitoring sensors;
  - on-chip thermal sensors (OCTSs) and
  - on-chip digital thermal sensors (DTSs).
- An on-chip thermal sensor (OCTS) is implemented as a thermistor (wire resistor), instead of the frequently used thermal diode.

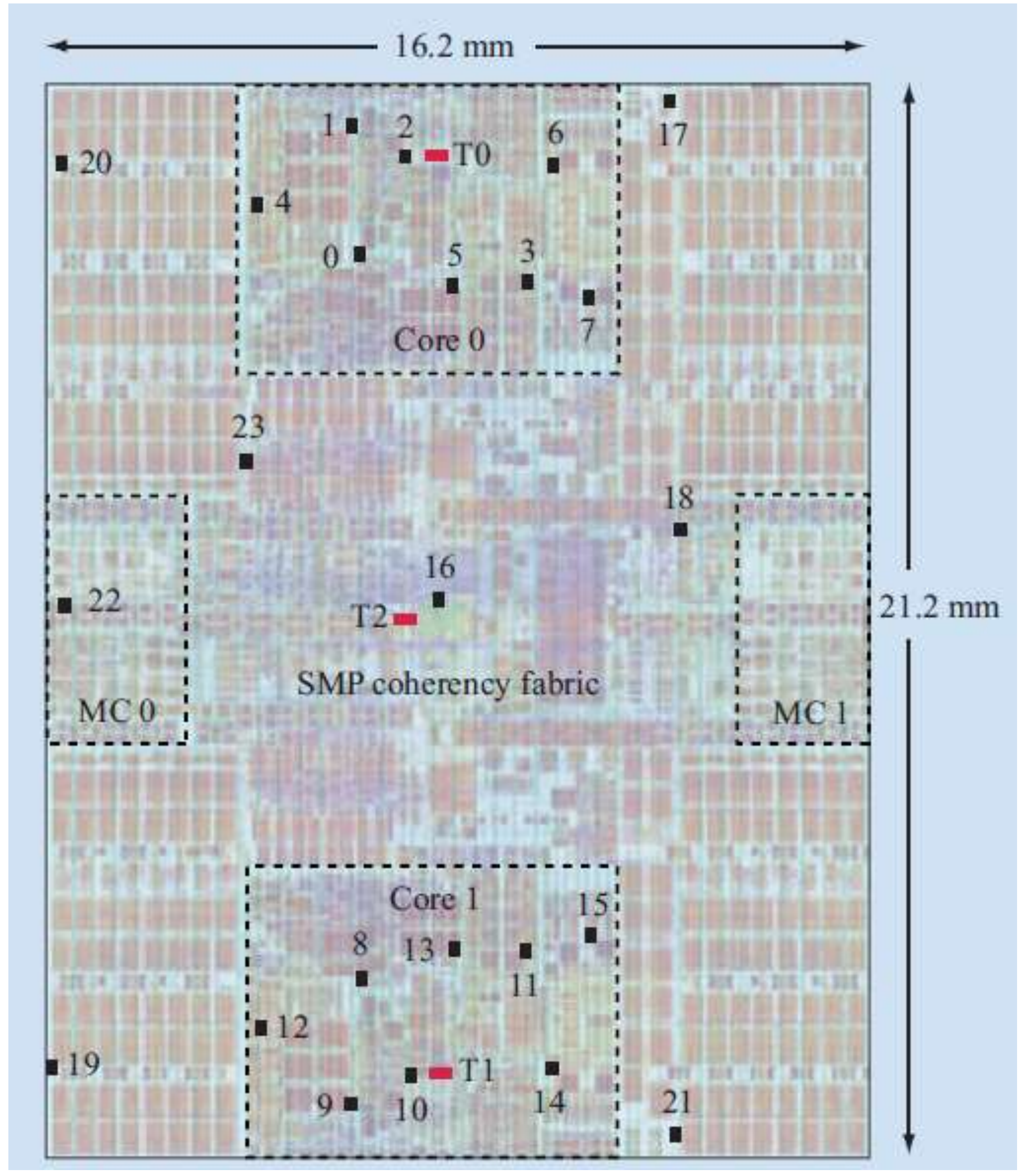
  The OCTS is calibrated during manufacturing tests by measuring its resistance at known temperatures.

  The disadvantage of the OCTS is that it requires off-chip module pins, external voltage connection and an A/D converter.
- The second type of thermal sensor is an on-chip digital thermal sensor (DTS) that can be read without an off-chip A/D converter.

  It is based on a temperature-sensitive ring oscillator.

  Since the ring oscillator's speed changes with the temperature, a count of the oscillations over a given time period characterizes the temperature.

Location of the on-chip thermal sensors OCTS (T0-T2) and digital thermal sensors (DTS0-DTS23) [48]



MC: Memory Controller

Introduction to Critical path monitors (CPMs) [48]

- CPMs are on-chip sensors to measure available timing margins and adjust the operating voltage according to a given policy.
- CPMs model the critical timing paths in various regions of the processors, as detailed in Section 6.3.4.
- Critical path monitors (CPMs) are placed next to the DTSs throughout the chip as close as possible to potential thermal hot spots and areas of high current draw.

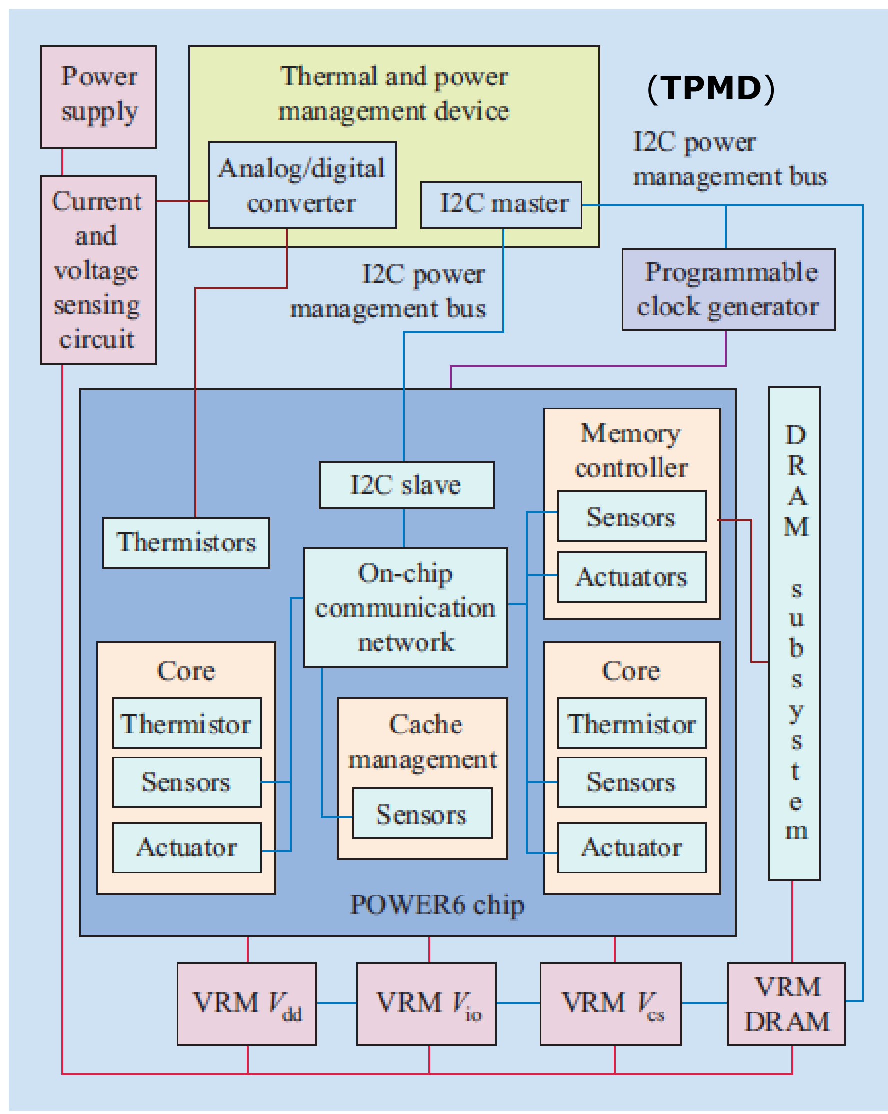
  These regions hold traditionally the speed limiting critical paths of the chip that tend to fail first.

Processor core and memory activity counters [48]

- Power/performance optimizations need to track core and memory activities.
- For each processor core there are three such registers available, as follows.
  - a finished (completed) instruction count register,
  - a dispatched instruction count register and
  - a processor cycle count register.
- These counters re-use signals already available to the performance monitoring unit (PMU) but are implemented independently from the traditional user-accessible performance counters and, thus, are always available for power management techniques.
- The three counts are cycle synchronized and are read in a single access.
- They can be used to obtain the finished instruction throughput (i.e., IPC) and the dispatch-to-finish rate, i.e. the extent of speculative activity in the core.

Power sensors [48]

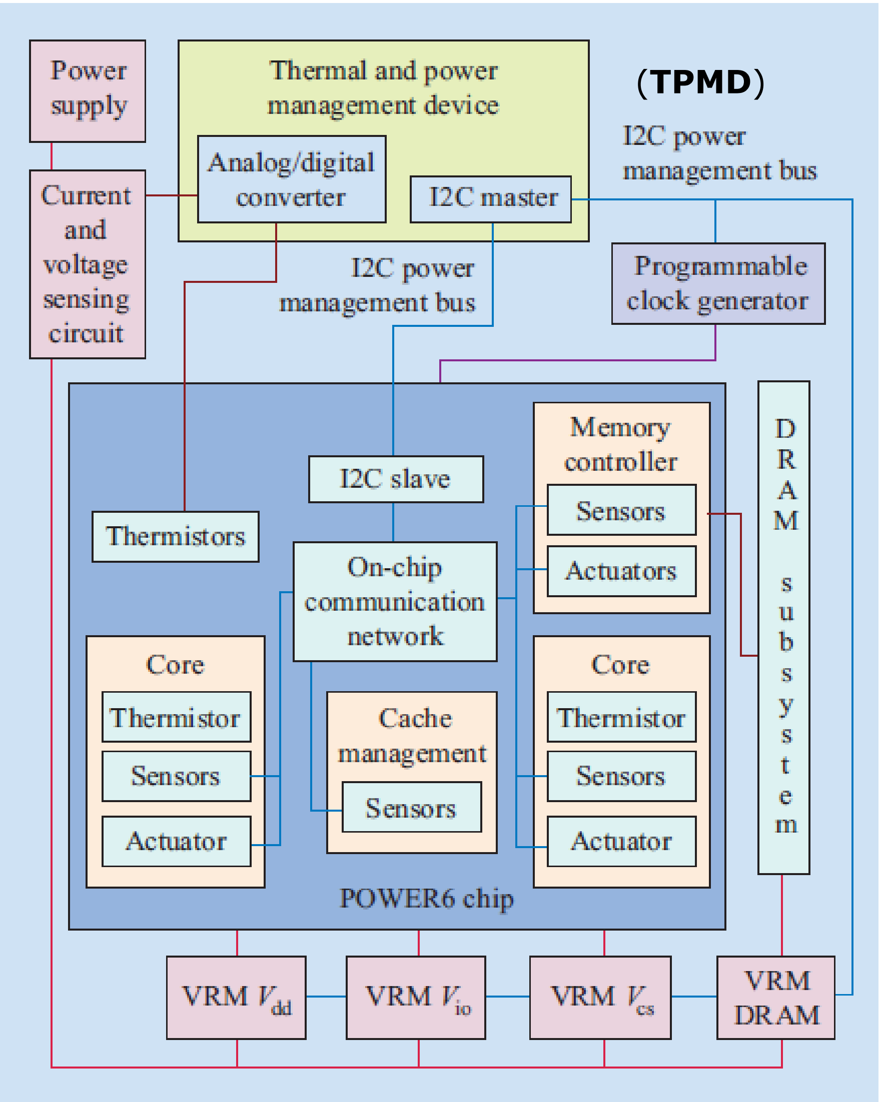
- Because it is difficult to design and implement accurate on-chip power sensors POWER6-based servers use high-precision external circuits for measuring the power consumption of select server subsystems and components, including the POWER6 chip, as well as the total system power consumption in order to enable power capping.
- In addition, individual VRMs provide reasonably accurate voltage and current measurements for both the POWER6 chip and memory (DRAMs).



EnergyScale management for a Single chip POWER6 chip [48]

On-chip communication network [48]

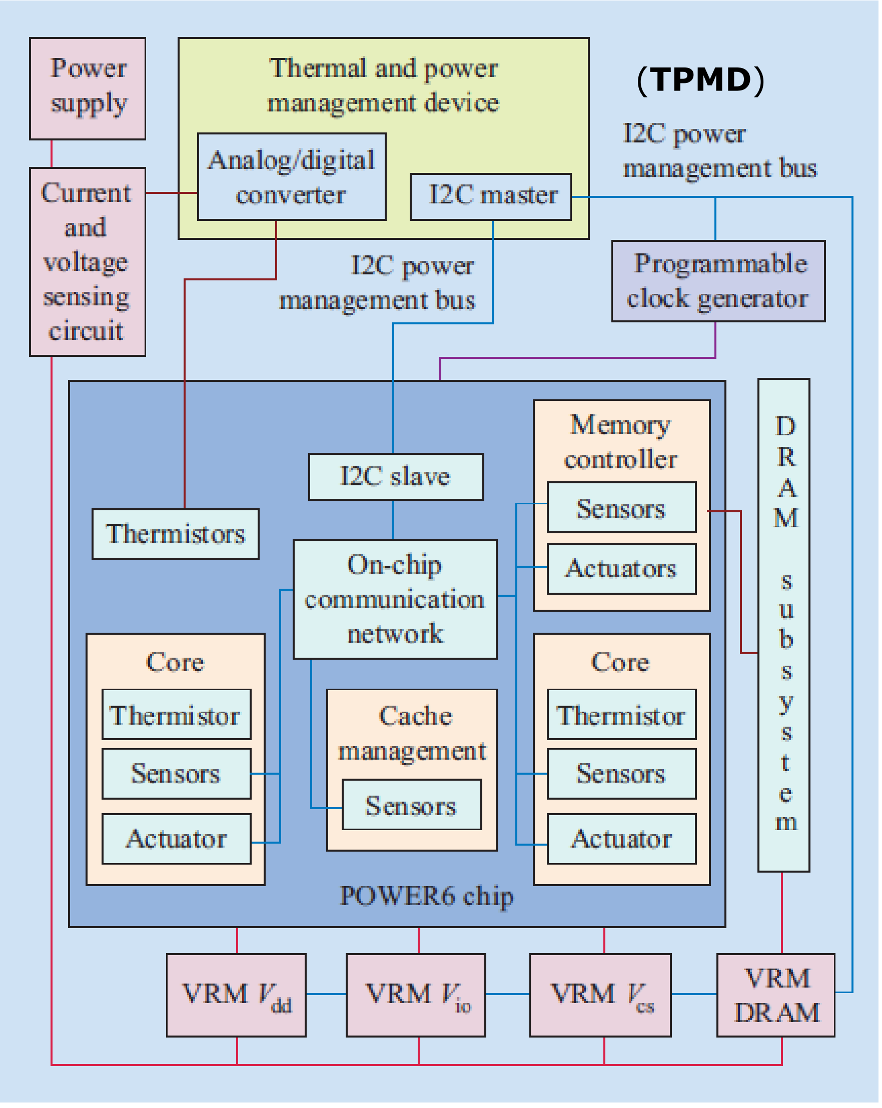
- All on-chip sensors and actuators have their own registers.
- These registers are connected to an on-chip network such that TPDM firmware can access these registers dynamically during runtime.
- The on-chip network is interconnected with the TPDM card via an I2C-based bus, called the Power management bus (see Figure).



EnergyScale management for a Single chip POWER6 chip

Actuators

- The available actuators control the core frequency through a Programmable clock generator as well as the supply voltages via the Voltage Regulator Modules (VRMs).
- There are four VRMs for four independent voltage domains, these are the
  - Vdd for logic
  - Vcs for SRAMs
  - Vio for the analog circuits used in off-chip interfaces
  - Avdd for the Phase-Locked Loops.



EnergyScale management for a Single chip POWER6 chip [48]

6.3.3 EnergyScale Architecture (14)

Power/performance optimization modes

- The customer may choose one of the power/performance optimization modes available with EnergyScale on Power6, including
  - Power/thermal capping

    It refers to the ability to limit the power consumption or temperature at a component, a subsystem, or the full system level.
  - Performance-sensitive power savings

    It refers to power management solutions that optimize for power savings while recognizing specific performance requirements.

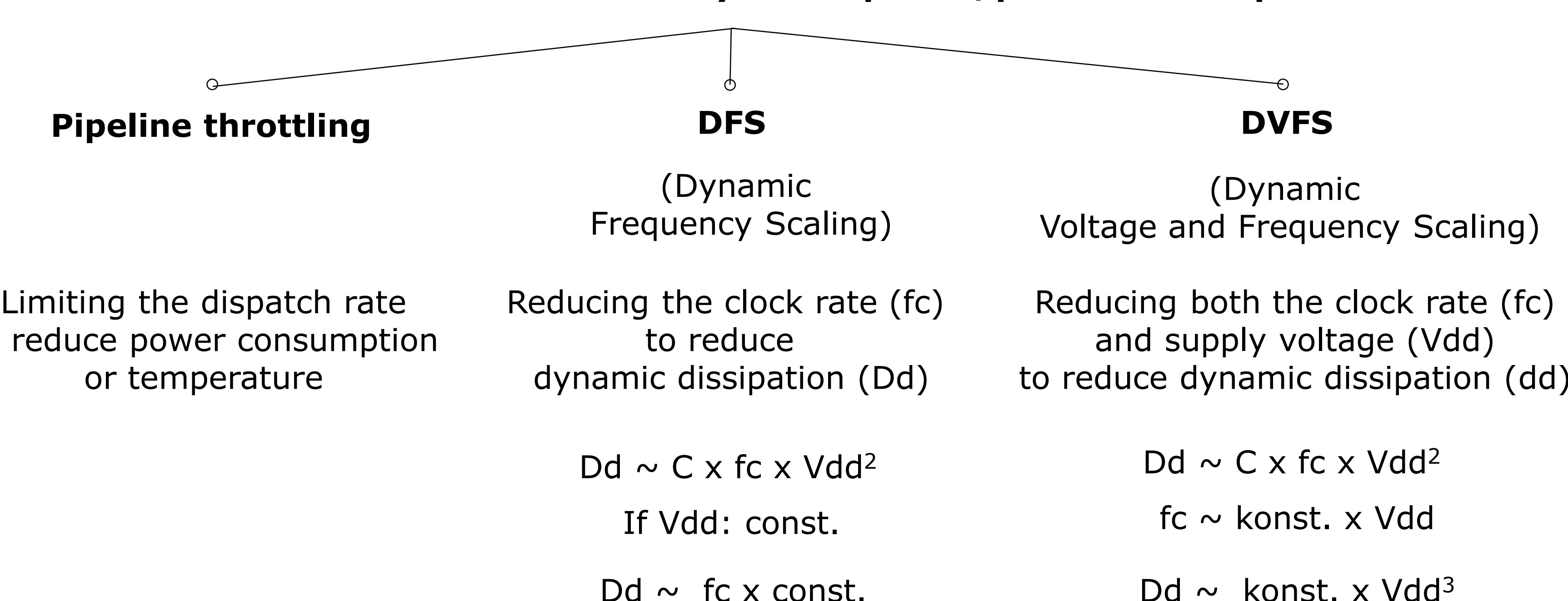
Mechanisms used for dynamic power/performance optimizations in the POWER6 [48]

Mechanisms used for dynamic power/performance optimizations in the POWER6

Processor related mechanisms for dynamic power/performance optimizations

Memory related mechanisms for dynamic power/performance optimizations

Processor related mechanisms for dynamic power/performance optimizations



Pipeline throttling

- Pipeline throttling saves a moderate amount of switching power (i.e. dynamic dissipation) by limiting the rate at which instructions can be dispatched.
- The disadvantage of this method is that the corresponding power savings is approximately linear with the performance loss (see Figure).
- On the other hand pipeline throttling allows a rapid-response mechanism to achieve a given power or thermal limit.

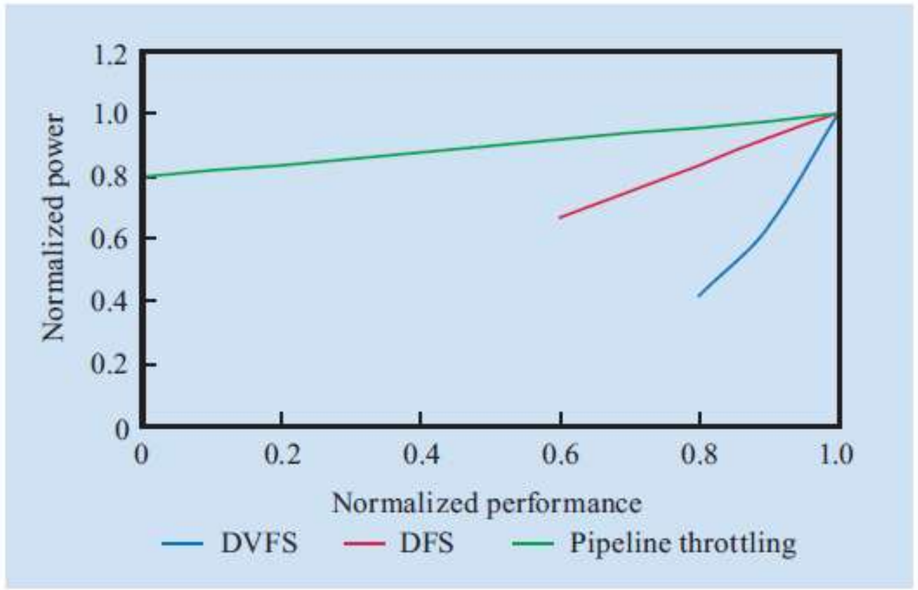


Figure: Power/performance graphs of processor related mechanisms for dynamic power/performance optimizations [48]

Dynamic frequency scaling (DFS)

- It is implemented by reducing the clock frequency at constant supply voltage to reduce power consumption or temperature.
- It is a more efficient but a bit slower method than pipeline throttling (see Figure).

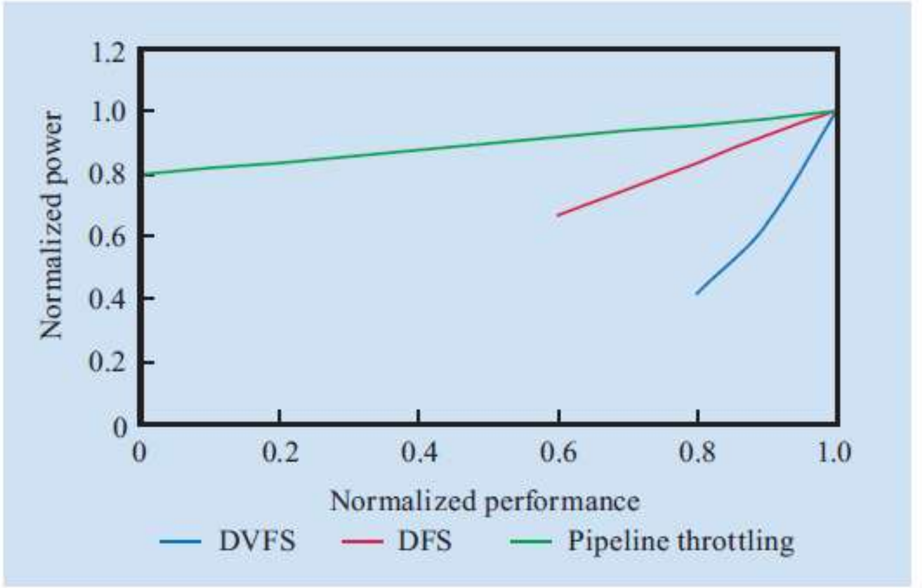


Figure: Power/performance graphs of processor related mechanisms for dynamic power/performance optimizations [48]

Dynamic voltage and frequency scaling (DVFS)

- By far the most efficient power-savings mechanism to reduce power consumption or temperature is to reduce both the supply voltage and the associated frequency, since the dynamic dissipation of CMOS circuits has a quadratic dependence on supply voltage and a linear dependence on frequency.
- In addition, static leakage current, which depends linearly on supply voltage, will also be reduced linearly by reducing the supply voltage.
- The drawback of DVFS is complexity and slower response time.

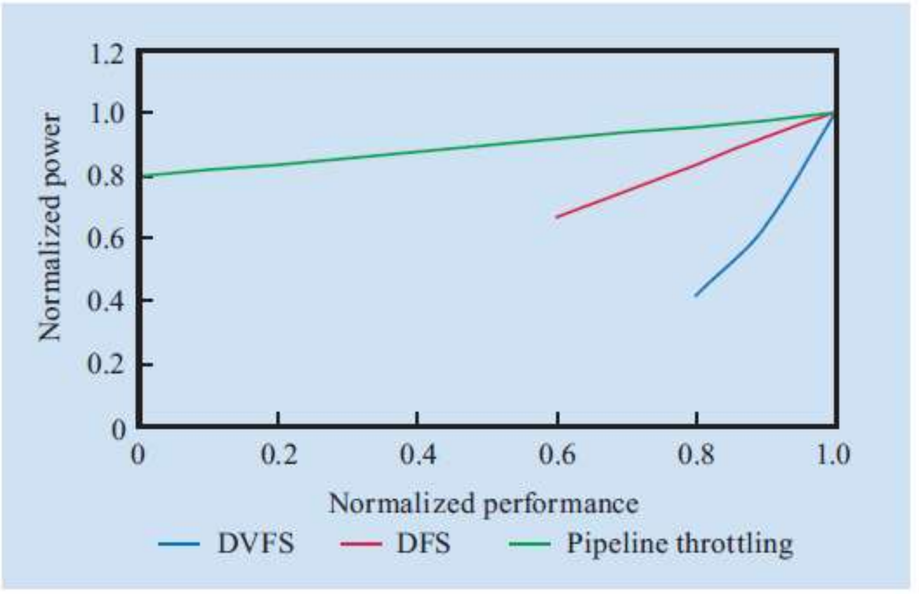


Figure: Power/performance graphs of processor related mechanisms for dynamic power/performance optimizations [48]

Memory related mechanisms for dynamic power/performance optimizations [48]

- POWER6 makes use of memory controller throttling to reduce power dissipation or temperature.
- This is a kind of activity regulations of the memory system.
- Since DRAM active power is proportional to the rate of memory requests being serviced, power savings result as a trade-off with memory subsystem performance (bandwidth).
- For details of the implementation of memory controller throttling we refer to [48].

6.3.4 Critical path monitors

6.3.4 Critical path monitors

Why to introduce critical path monitors? [50]

- Processor operating voltages are determined during the manufacturing test while running a self-test program to determine the voltage limit for correct operation at the chip's target maximum clock frequency and operating environment (e.g. temperature).
- Typically, an extra margin called the voltage guardband is added to the determined operating voltage in order to guarantee proper circuit timing even during worst-case conditions.

  Worst-case conditions relate to process variations, system power supply variations, workload induced thermal and voltage variations, aging, random uncertainty, and test inaccuracy.
- Taking into account a guardband into the voltage specification allows the microprocessor to operate correctly during worst-case conditions as well, but during typical conditions the guardband is substantially larger than necessary and wastes energy.
- Critical Path Monitors (CPM) are on-chip sensors that measure the actual available timing margin of circuits on the chip and thus allow to reduce the guardband to be added and thus to save power consumption.

Principle of implementation of CPM [51]

- Based on the Figure below let's imagine that at clock cycle n, a pulse is launched down the Representative Critical paths and then captured into a chain of say, 12 latches (flip-flops) of the edge detector on the following clock cycle, n + 1.
- The penetration of the edge into the chain of 12 latches gives an indication on the circuit's timing margin at the given operating conditions.
- The operating conditions include parameters such as voltage, temperature, workload, and age.
- In other words, the CPM indicates how the actual operating environment affects the current timing margin in the associated region of the chip.

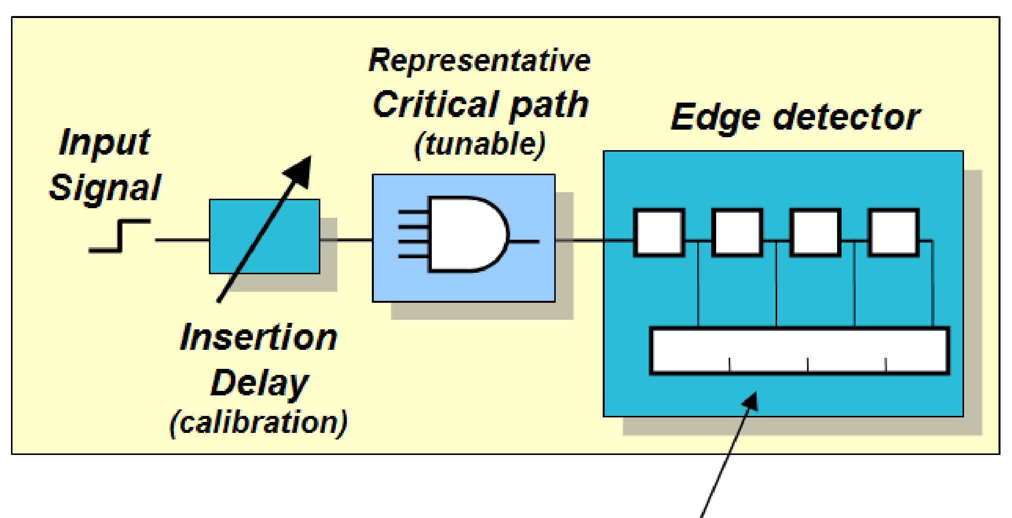


Figure: Principle of the implementation of the Critical Path Monitor (CPM) [51]

Implementation of CPMs -1 [52]

Here we do not want to go into details of the circuit details of the CPMs instead we will draw attention to key points of the implementation.

CPMs are built up of three main blocks as the Figure below shows:

- the Synchronizer
- the Critical path (designated as the Delay Path in the Figure)
- the Time-to-Digital Converter and
- the Data Analysis.

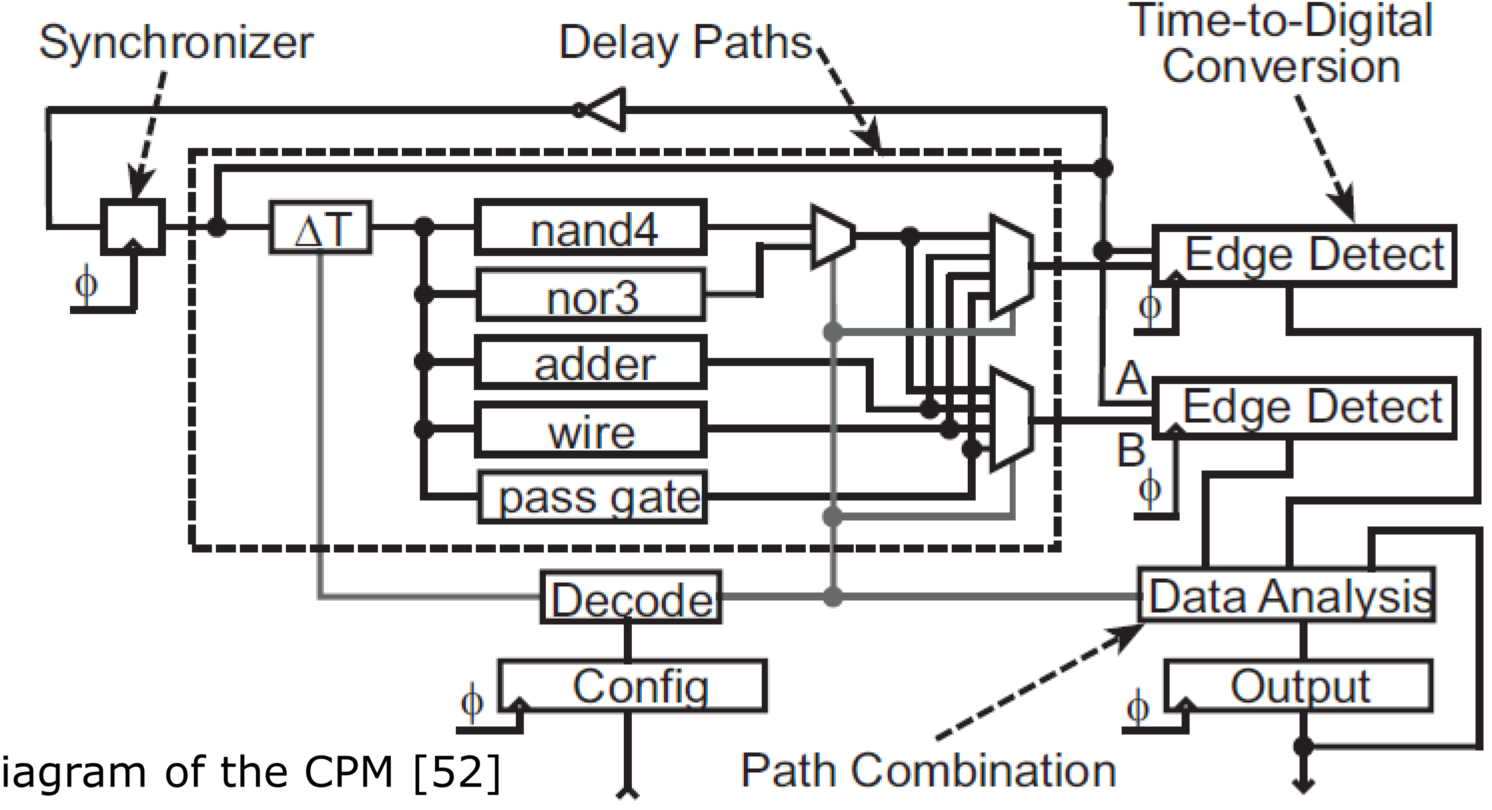


Figure: Block diagram of the CPM [52]

Implementation of CPMs -2 [52]

- The Synchronizer is in fact a flip-flop that operates as a clock divider such that the CPM samples a rising edge one cycle and a falling edge the following cycle.
- The Delay Path consists of five parallel paths that can be combined to one of 14 alternatives by means of the Data Analysis block to emulate a hybrid path.

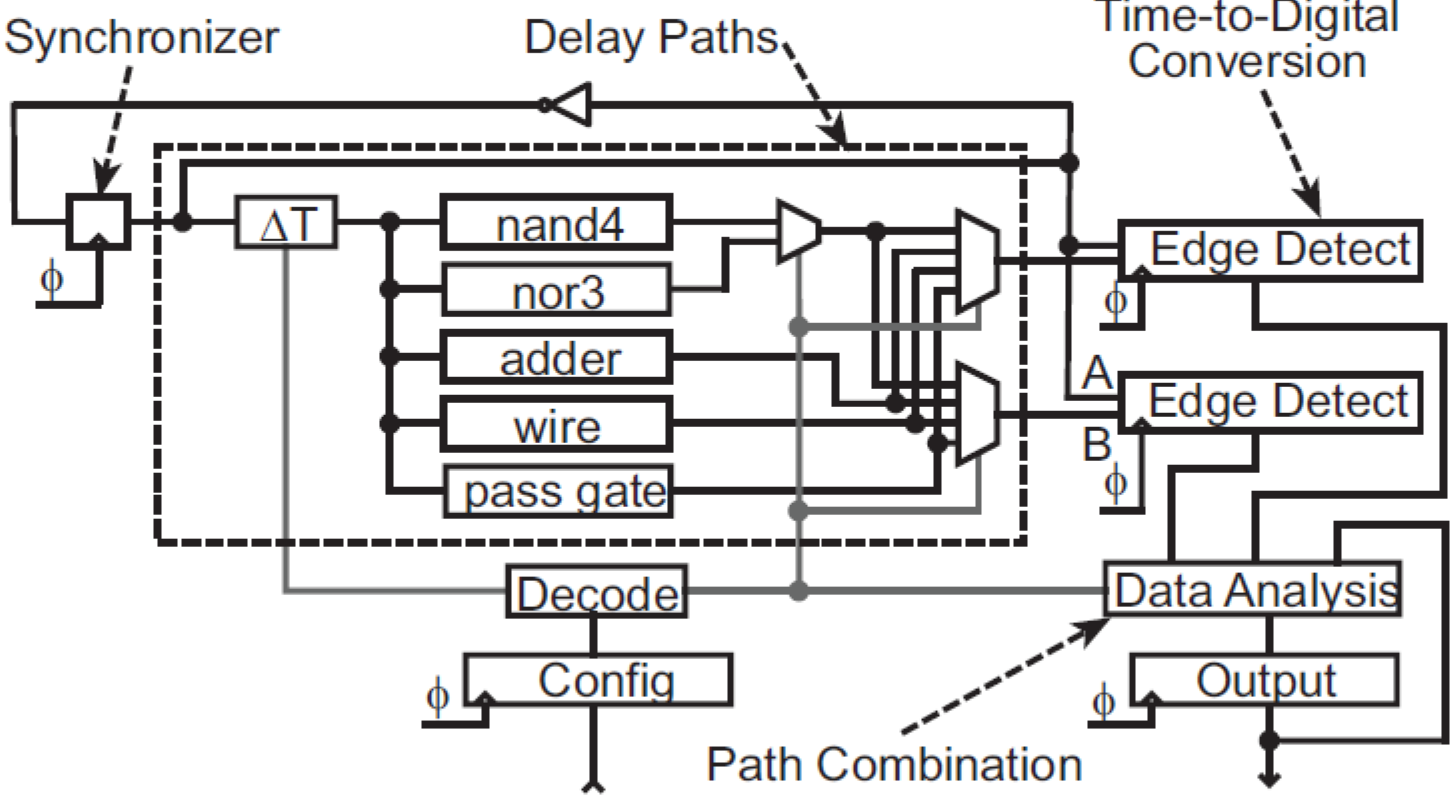


Figure. Block diagram of the CPM [52]

Principle of operation of the edge detector -1 [50]

- Basically, the edge detector is a delay line built up of a chain of buffers (e.g. of dual NAND gates).
- Clock edges generated by the synchronizer traverse first the Critical path and then enter the delay line and propagate along it, as seen in the Figure.
- The rising edge of the clock signal (ϕ) captures in the chain of latches how far the incoming edge has been propagated.
- The output of the edge detector is a string of 1s followed by 0s with the location of the 1 to 0 transition indicating the timing edge as a function of the clock frequency.

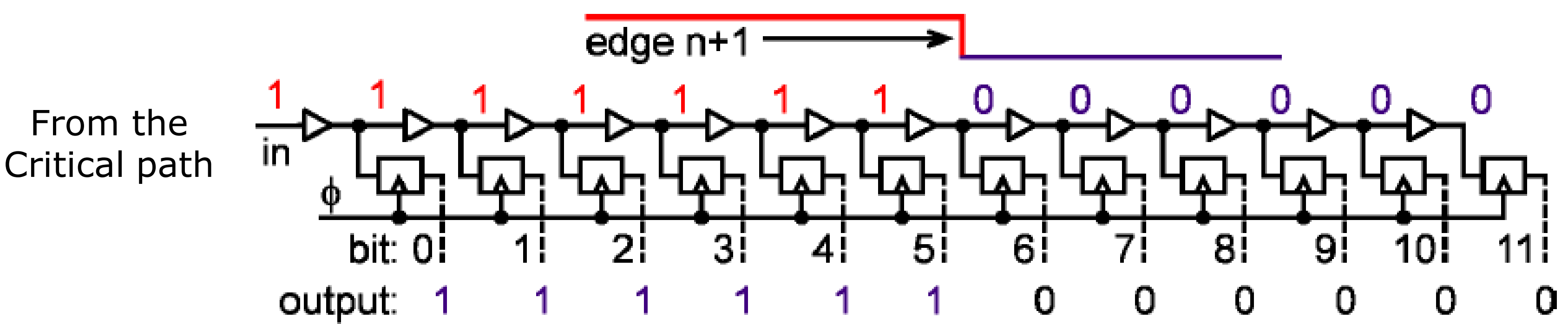


Figure: Principle of the operation of the Edge detector [50]

Principle of operation of the edge detector -2 [50]

- Ideally, after calibration, the edge transition is located in the middle of the edge detector for maximum sensor range.
- If the voltage drops or the temperature increases, the Critical path feeds the edge with a delay, so the edge will not propagate as far into the edge detector and the edge moves toward bit 0, as indicated in the Figure below for the edge n+2.

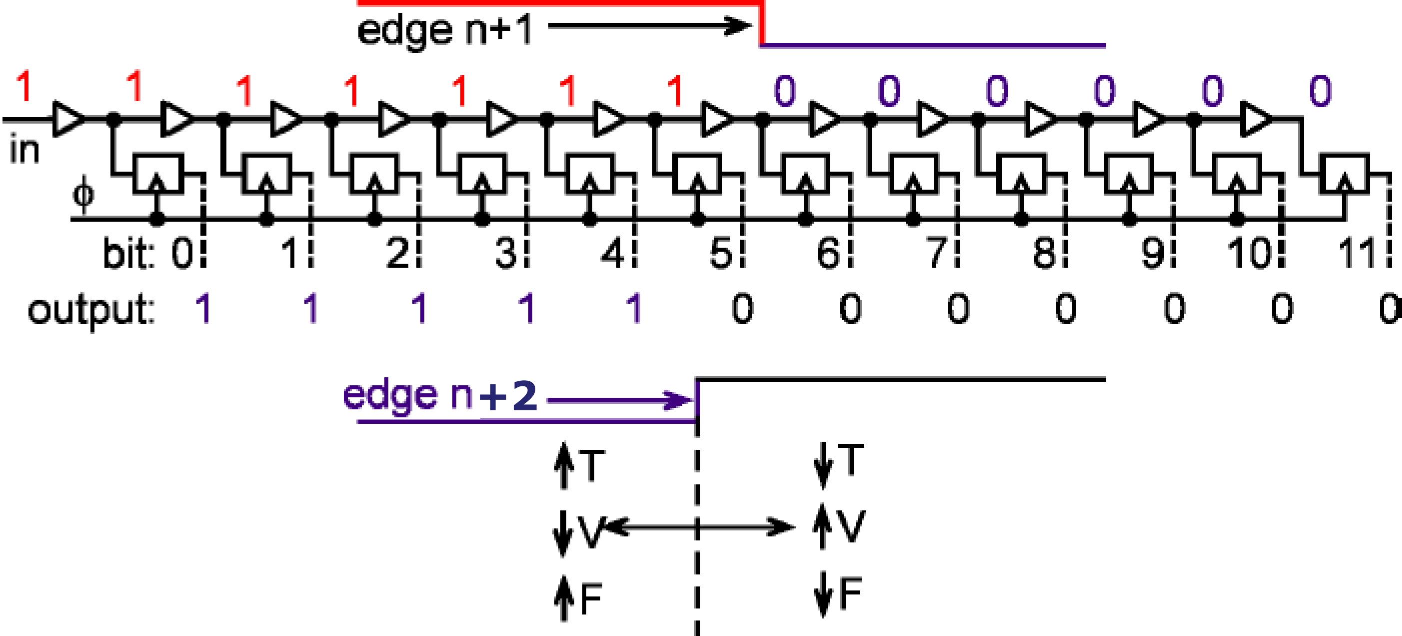


Figure: Principle of the operation of the Edge detector [50]

- If the frequency increases, there is less time for the edge to propagate, so the edge also moves toward bit 0.

6.3.4 Critical path monitors (7)

Principle of operation of the edge detector -3 [50]

- By contrast, increased voltage, reduced temperature, and reduced frequency will cause the edge to move toward bit 11.
- The shift of the edge in the edge detector is a measure of the change for any parameter modification or noise affecting timing (voltage, temperature, workload, clock jitter, skew, etc.) and can be used to adjust clock frequency and associated voltage according to the user given power management policy.

Placement of the CPMs on the POWER6 die [53]

There are 24 CPMs on one POWER6 chip; eight in each of the two cores which run at core frequency Fcore, and eight in the nest region which runs one-half Fcore, as seen in the Figure.

Figure: Placement of the CPMs on the POWER6 die [53]

Remarks

1) IBM has a patent for Critical Path Monitors (CPMs), (US 7576569, filed in 10/2006 and issued in 8/2009).

2) IBM's POWER6 (2006) is supposedly the first commercial computer implementing Critical Path Monitors.

Nevertheless, there are a few prior activities targeting the use of CPMs, here we pinpoint only two of them, both originating from about 2003.

a) The work done at the University of Michigan aimed at reducing the voltage margins by the Razor and Razor II designs [54].

A peculiarity of the work was adding a mechanism for rollback and replay in case of system failure.

The proposed Razor technique was implemented in a prototype Alpha processor.

b) The design proposal for an adaptive voltage Scaling system that is based on an on-chip critical path emulation at the University of Waterloo (Canada) [55].

For details of the cited developments we refer to the literature, e.g. [54], [55].

6.3.5 Introduction of the Nap idle mode

6.3.5 Introduction of the Nap idle mode [48]

- POWER6 introduced the Nap mode to reduce power consumption if a core or the processor is idle.
- Each hardware thread running on a POWER6 core can issue an instruction that puts it into the nap mode.

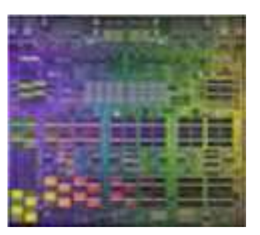
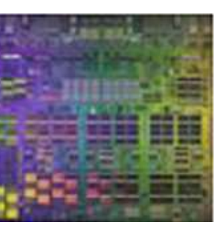
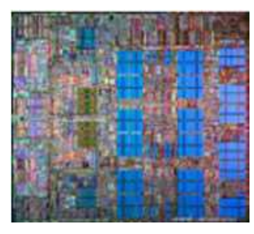
  When both hardware threads for that core are in Nap mode, the whole processor core then enters the Nap state.
- In the Nap state, the processor eliminates almost all of the switching power in the core by stopping the internal clocks and restricting operation of its functional units.
- The two cores of the POWER6 enter and exit Nap mode independently of each other.
- The Nap state is interruptible, so the OS or hypervisor can re-awaken a napping core either by issuing the appropriate type of interrupt or by configuring the core to wake up on the basis of external I/O or timer (decrementer) interrupts.
- As a result, by using the Nap mode 10% to 20% power savings can be achieved while the processor is running the idle loop.

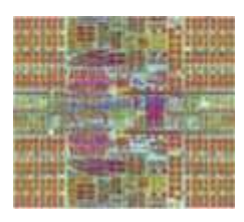
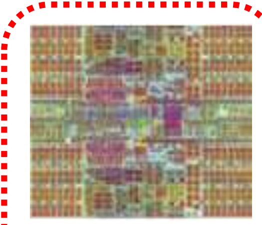
7. POWER6+

7. Introduction to the POWER6+

- It is a slightly enhanced version of the POWER6.
- Introduced in 4/2009, but shipped in Power systems already since 10/2008.
- Increased clock frequency up to 5.0 GHz.
- No relevant innovations were done in the POWER6+.vs. the foregoing POWER5.

Key features of the POWER6+

	POWER4	POWER4+	POWER5	POWER5+	POWER6	POWER6+
Launched	12/2001	11/2002	5/2004	10/2005	7/2007	4/2009
Technology	180 nm	130 nm	130 nm	90 nm	65 nm	65 nm
Die size	414 mm2	380 mm2	389 mm2	245 mm2	341 mm2	341 mm2
Transistors	174 M	184 M	276 M	276 M	790 M	790 M
Cores up to	2	2	2	2	2	2
SMT	-	-	2-way	2-way	2-way	2-way
Typ. fc	1.1-1.3 GHz	1.2-1.7 GHz	1.65 -1.9 GHz	1.9-2.3 GHz	3.5-5 GHz	4.7-5 GHz
L2	1.44 MB	1.5 MB	1.9 MB	1.9 MB	4 MB/core	4 MB/core
L3	32 MB	32 MB	36 MB	36 MB	32 MB	32 MB
Mem. contr.	1	1	1	1	2/1	2/1
Memory up to	DDR-200	DDR-200	8xDDR-533	8xDDR2-533	DDR2-667	DDR2-667

Overview of the POWER6+ based server models (12/2009) [56]

December 2009

POWER6™ BUILT ON Power™	BladeCenter® JS12 Express	BladeCenter JS22 Express	BladeCenter JS23 / JS43 Express	Power 520 Express	Power 550 Express	Power 560 Express	Power 570	Power 595
Machine type-model	7998-60X	7998-61X	JS23: 7778-23X JS43 = JS23 + FC 8446	8203-E4A	8204-E8A	8234-EMA	9117-MMA	9119-FHA
System package	Blade Server/ BladeCenter	Blade Server/ BladeCenter	Blade Server BladeCenter	4U, 19" rack or tower	4U, 19" rack or tower	4U / node, 19" rack	4U / node, 19" rack	42U, 24" CEC frame
# of cores (GHz & processor)	2 (3.8 GHz POWER6™)	4 (4.0 GHz POWER6)	JS23: 4 (4.2 GHz POWER6+) JS43: 8 (4.2 GHz POWER6+)	1, 2 (4.2 GHz POWER6) 2, 4 (4.7 GHz POWER6+)	2, 4, 6, 8 (3.5,/4.2 GHz POWER6) 2, 4, 6, 8 5.0 GHz POWER6+)	4, 8, 16 (3.6 GHz POWER6+)	2 to 16 (3.5 GHz POWER6, 4.4/5.0 GHz POWER6+)[1] 4 to 32 (4.2 GHz POWER6+)	8 to 64 (4.2 GHz POWER6) 16 to 64 (5.0 GHz POWER6)

8. POWER7

- 8.1 Introduction to the POWER7
- 8.2 Main enhancements of the POWER7 vs. the POWER6
- 8.3 Key innovations of the POWER7

8.1 Introduction to the POWER7

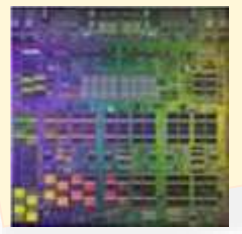
8.1 Introduction to the POWER7

- Launched: 2/2010
- 45 nm technology
- 567 mm2, 1.2 billion transistors
- Clock rate up to 4.42 GHz
- 6-wide out-of-order-superscalar with an issue rate of 8 and 12 execution units
- Uses the same socket as the POWER6

Key features of the POWER7

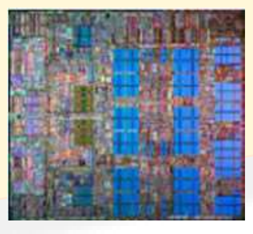
	POWER7	POWER7+	POWER8	POWER8+	POWER9
Launched	2/2010	10/2012	4/2014	Planned/cancelled	12/2017
Technology	45 nm	32 nm	22 nm		14 nm
Die size	567 mm2	567 mm2	650 mm2		693 mm2
Transistors	1.2 b	2.1 b	4.2 b		8.0 b
Cores (up to)	8	8	12		12 SMT8 cores 24 SMT4 cores
SMT	4-way	4-way	8-way		4-way/8-way
Typ. fc	3.72-4-42 GHz	3.1 -4.42 GHz	3.02-4.35 GHz		Up to 4 GHz
L2	256 KB/core	256 KB/core	512 KB/core		512KB/2 cores
L3	4 MB/core	10 MB/core	12 MB/core		10 MB/2 cores
Mem. contr.	2/1	2/1	8		8
Memory up to	DDR3-1066	DDR3-1066	DDR3-1600		DDR4-2666

Key innovations of the POWER7 (Die photos from [3])



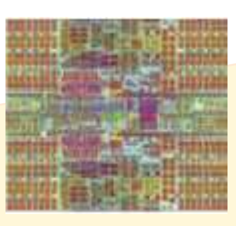
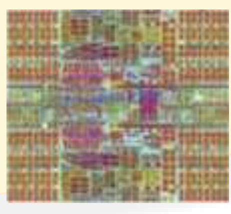
POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP



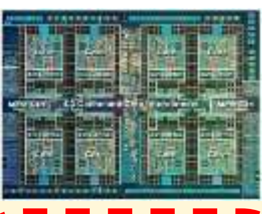
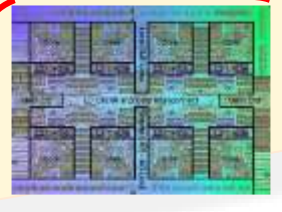
POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating

POWER6/6+
65/65 nm

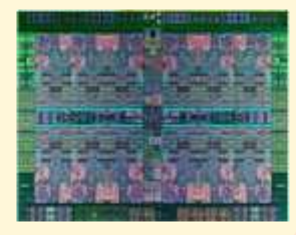
Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

*POWER7+

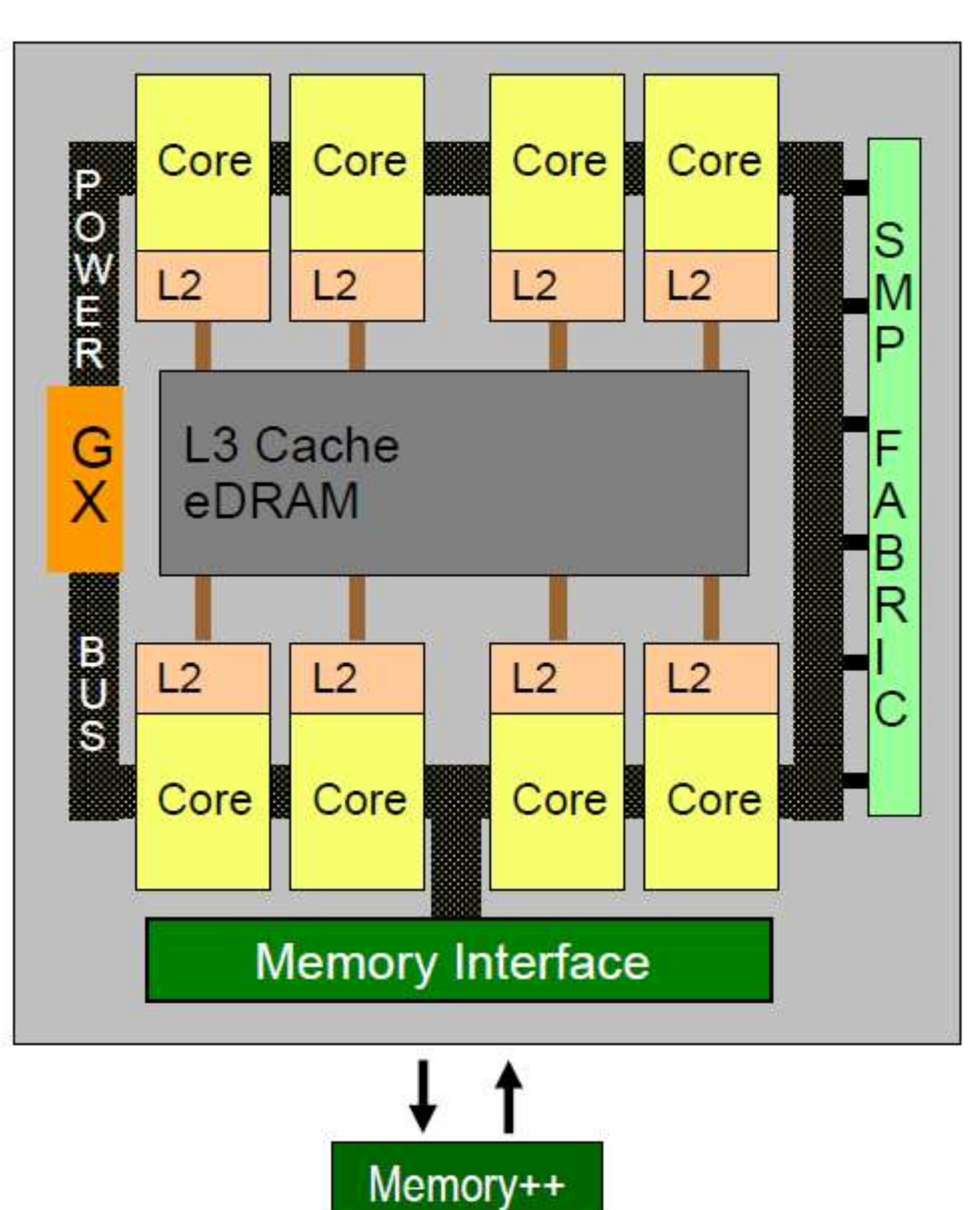


POWER8
22 nm

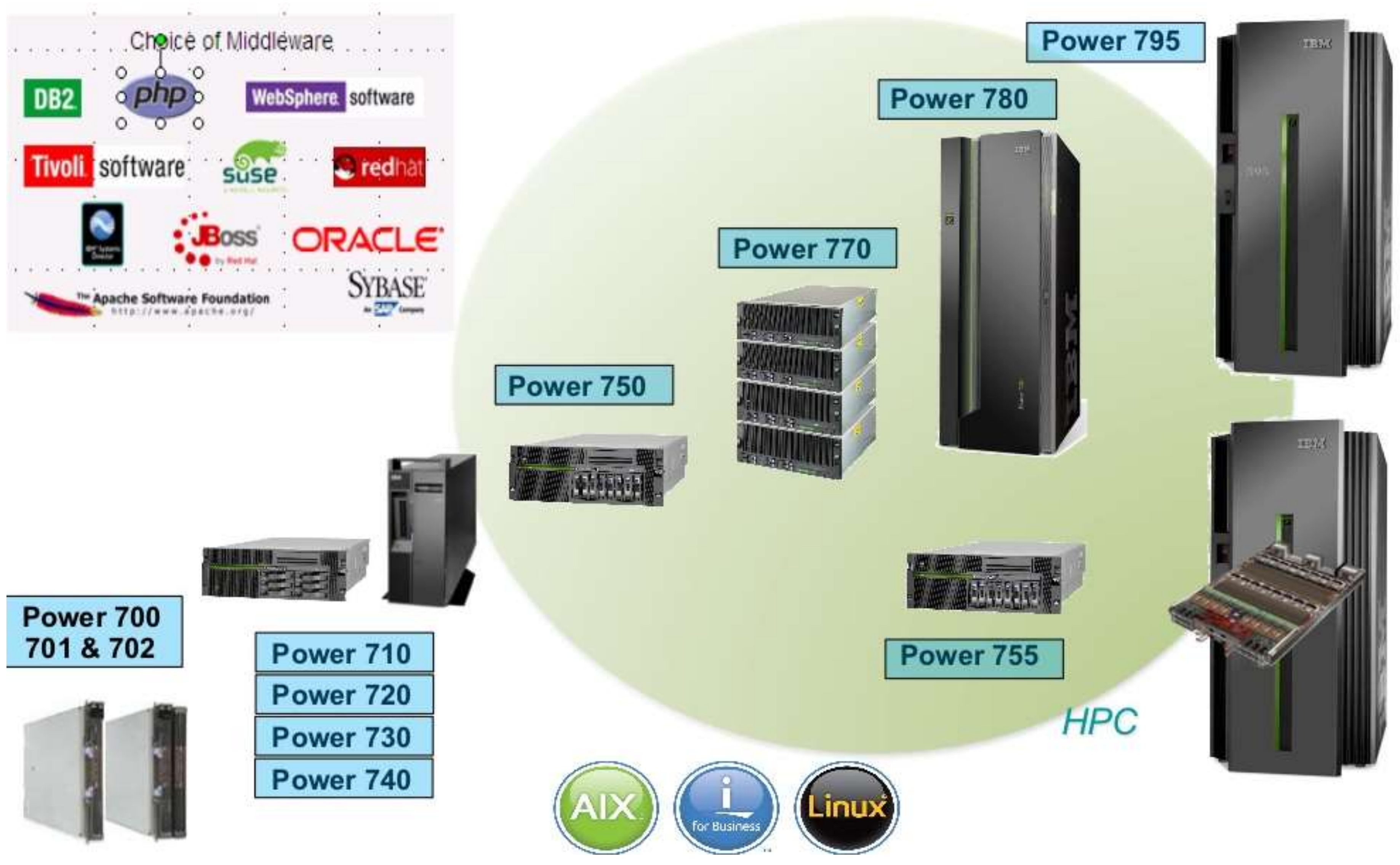
12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2001 2004 2007 2010 2014

High level block diagram of the POWER7 [57]



POWER7 based server models [58]



8.2 Main enhancements of the POWER7

- 8.2.1 Enhanced execution resources in the POWER7
- 8.2.2 Merged FPU and VMX units with a Unified Register File
- 8.2.3 Enhanced memory system

8.2 Main enhancements of the POWER7

The efficient implementation of an 8-core 4-way SMT design requires

- more execution resources in the microarchitecture,
- a more efficient cache system and
- higher memory bandwidth.

Subsequently, we will discuss how the above enhancements are implemented in the POWER7.

8.2.1 Enhanced execution resources in the microarchitecture

As the Table on the next slide below shows the POWER7 provides considerable more execution resources (6-wide front-end, 12 execution units) than preceding models to serve the increased number of cores and threads.

8.2.1 Enhanced execution resources in the POWER7

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-ways
Width of the front-end	5	5	5	6	8	12
Dispatch rate	5	5	(In-order design)	6	8	12
Issue rate	8	8	7	8	10	16
No. of execution units per-core	8	8	9	12	16	20
No/type of execution units per-core	2 FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP 1BR/CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 2VMX, 1DFU, 2LU, 1 Crypto	8AGEN, 4VSU(128) 4LS(128) 2BRU DFU Cripto

Block diagram of the Instruction-Sequencing Unit (ISU)

Without going into details we show that the front-end is 6-wide, the dispatch rate is 6 whereas the issue rate is 8 in the POWER7.

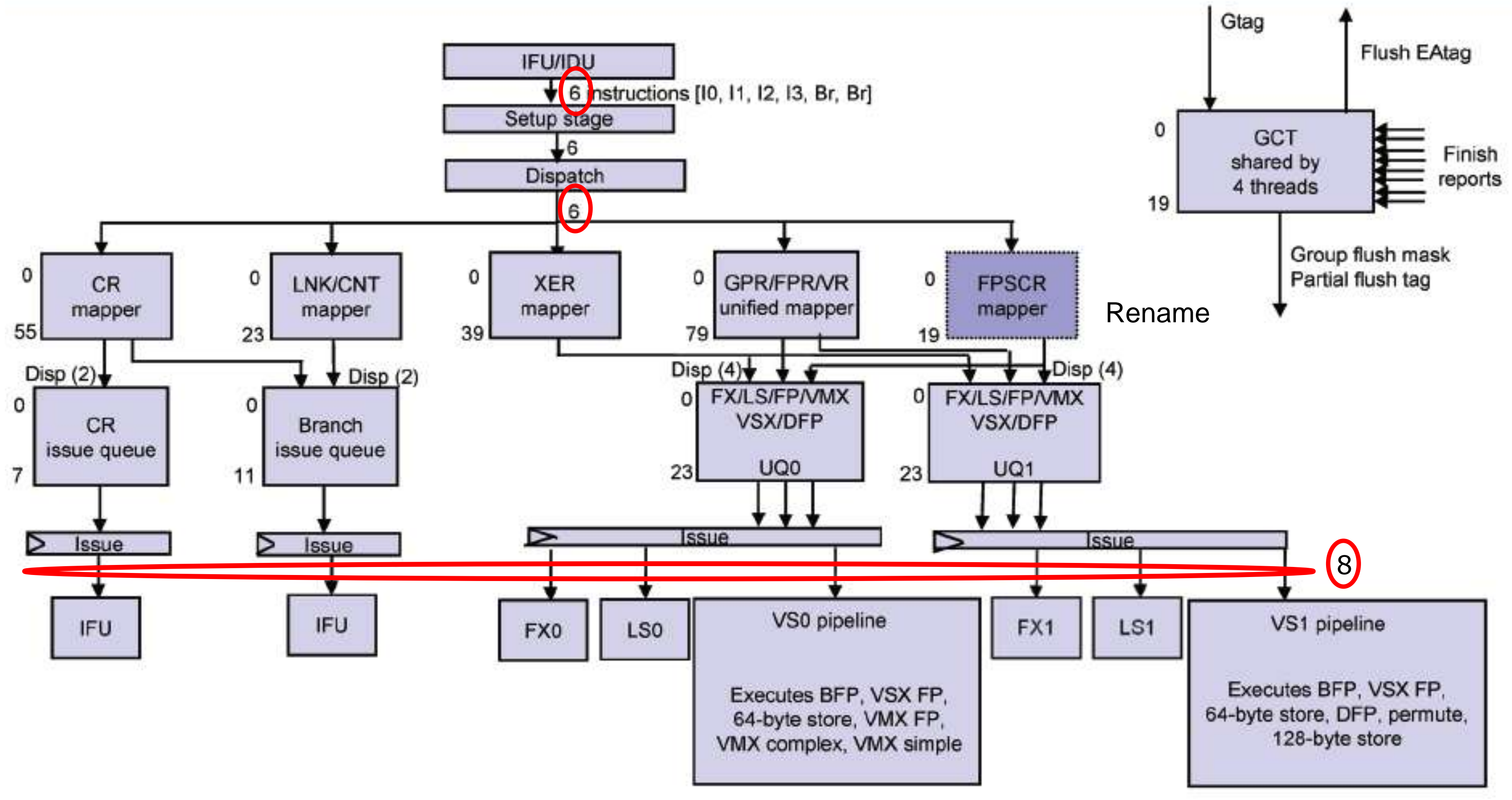


Figure: Block diagram of the Instruction-Sequencing Unit of the POWER7 [59]

8.2.1 Enhanced execution resources in the POWER7 (3)

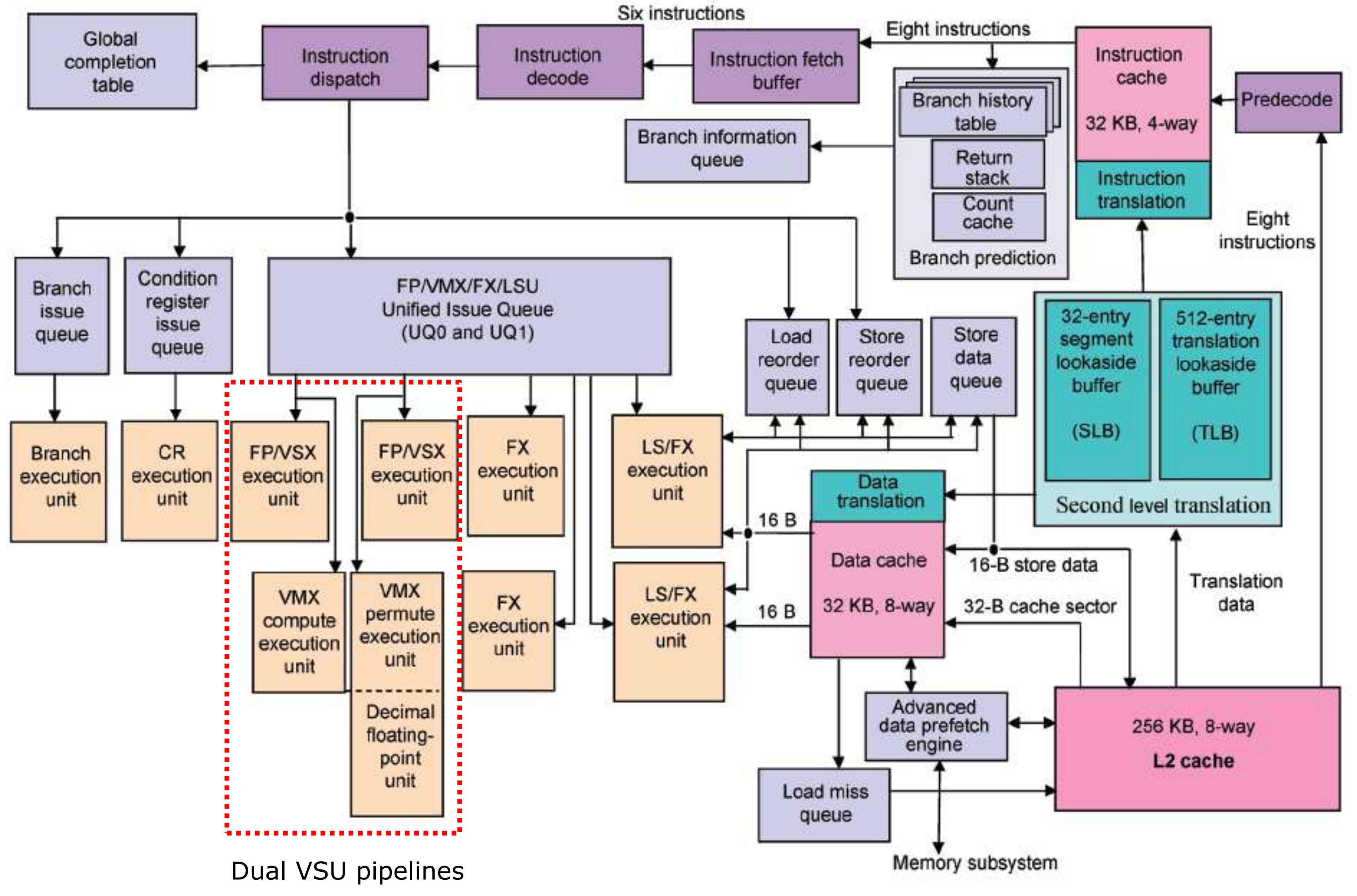
12 available execution units

- 2 FX pipelines
- 2 LS pipelines that can execute simple FX operations
- 4 double-precision (DP) FP pipelines, each able to execute DP MADD operations
- 1 vector pipeline
- 1 branch pipeline
- 1 condition register (CR) and logical and
- 1 decimal FP pipeline (first

Remark

- For achieving a significant improvement in HPC computing IBM doubled the number of DP FP pipelines.
- Thus a POWER7 chip executes per cycle approximately 8 times more FP operations than the POWER6.

Block diagram of a POWER7 core [59]



Dual VSU pipelines

8.2.2 Merged FPU and VMX units with a Unified Register File [60]

- In the POWER6 design the FPU (Floating-Point Unit) and VMX (Vector Media Extension Unit) execution units were separate, by contrast in the POWER7 these two units became merged into one unit, called the VSU (Vector Scalar Unit).
- The VSU implements also the new VSX extension (Vector and Scalar Extension) that allow two-way SIMD operations out of a unified 64 entry 128 bit register file, as discussed subsequently.
- The merged VSU unit is partitioned into two parts to support SMT4, as the next Figure shows.

The partitioned implementation of the merged VSU

As seen below the VSU is partitioned into the VS0 and VS1 units.

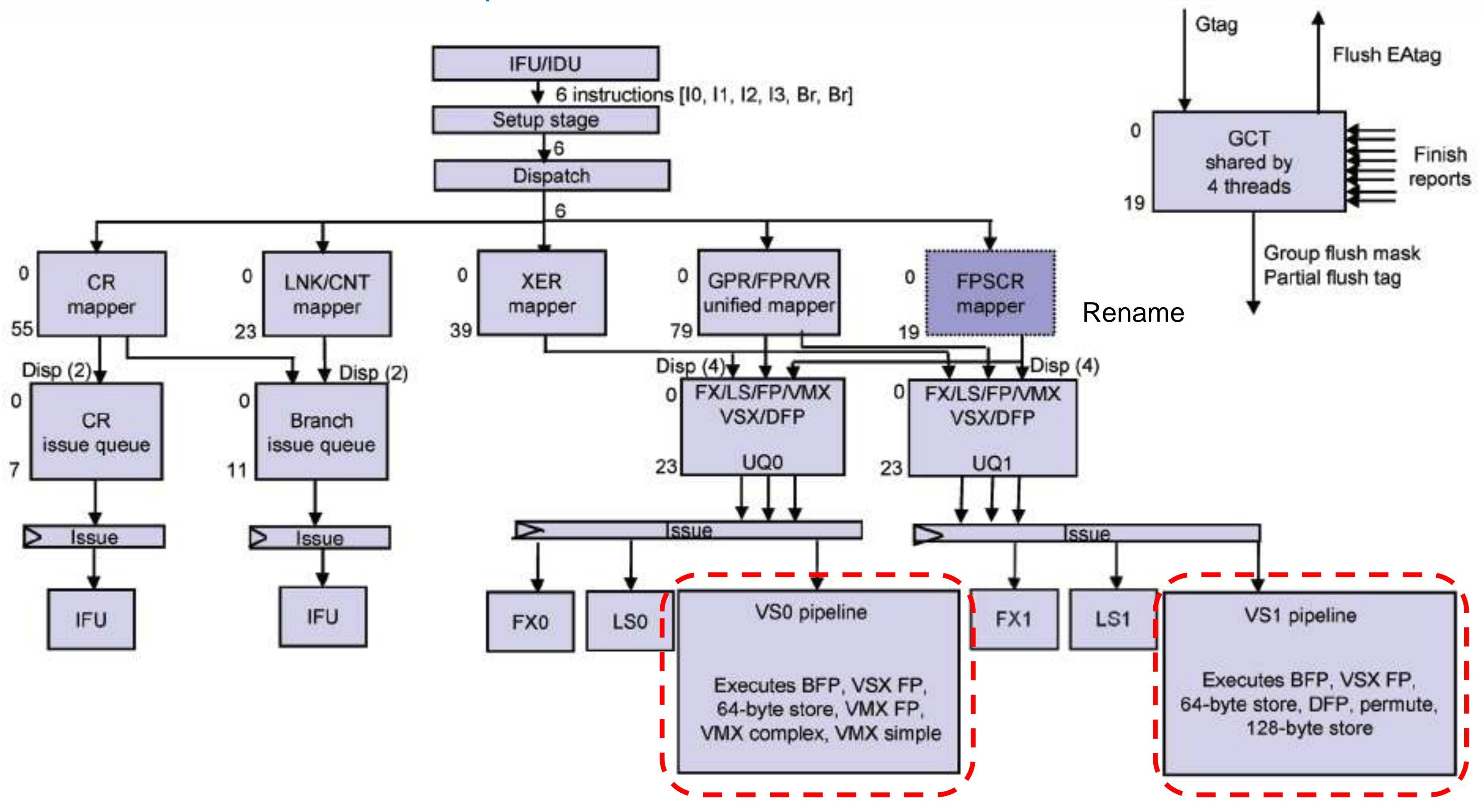
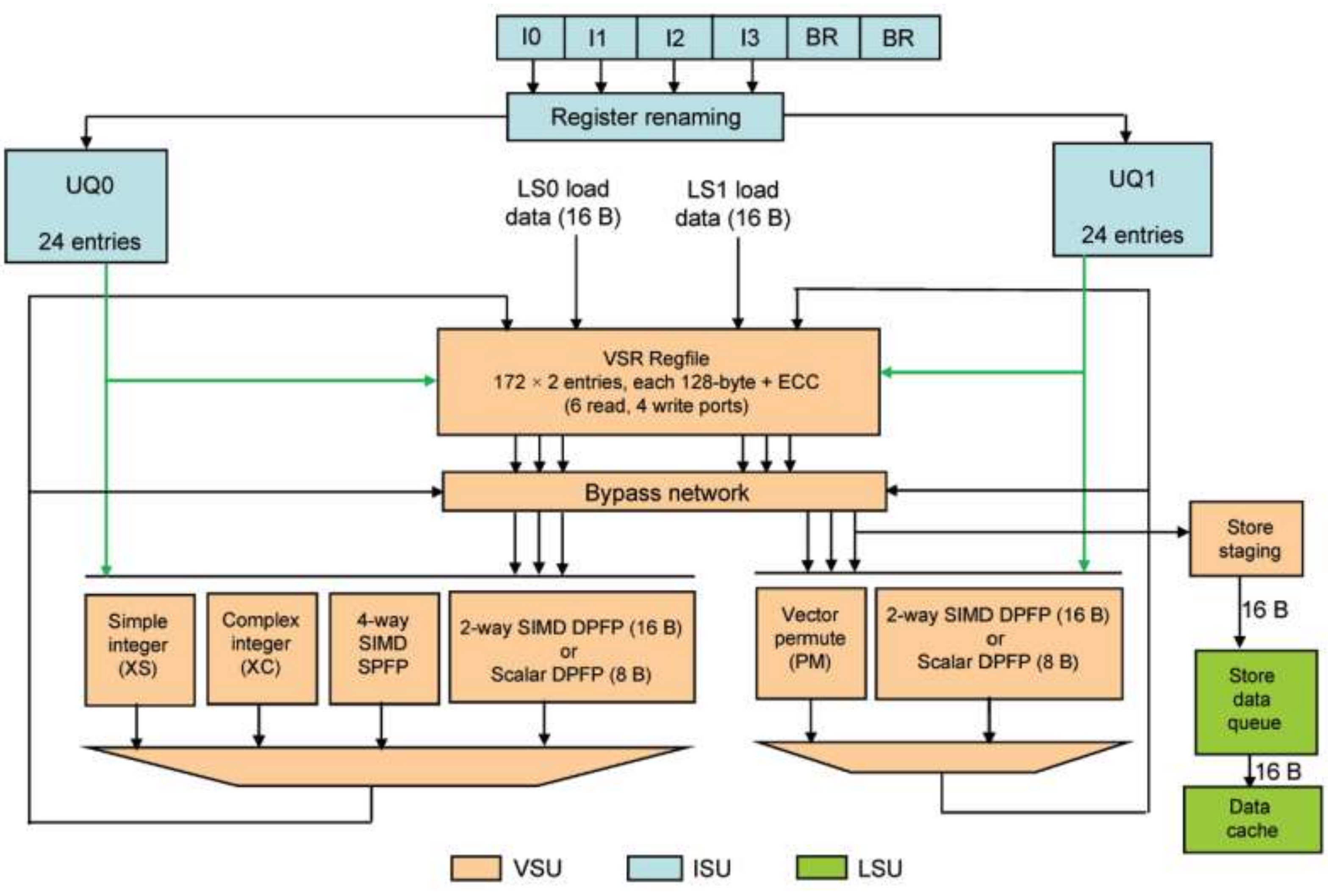


Figure: Block diagram of the Instruction-Sequencing Unit of the POWER7 [59]

The dual VSU pipelines of the POWER7 [59] -1



The dual VSU pipelines of the POWER7 [59] -2

Remark

In the last three Figures some details concerning the VSUs differ.
Presumably, the most detailed last Figure can be considered as the most authentic one.

Unified Register File [60]

With merging the FPU and VMX units IBM implemented a Unified Register File that is shared across the BFP (Binary Floating-Point), DFP (Decimal Floating-Point), VMX (Vector Multimedia Extension) and VSX (Vector Scalar Extension) operations, as indicated below.

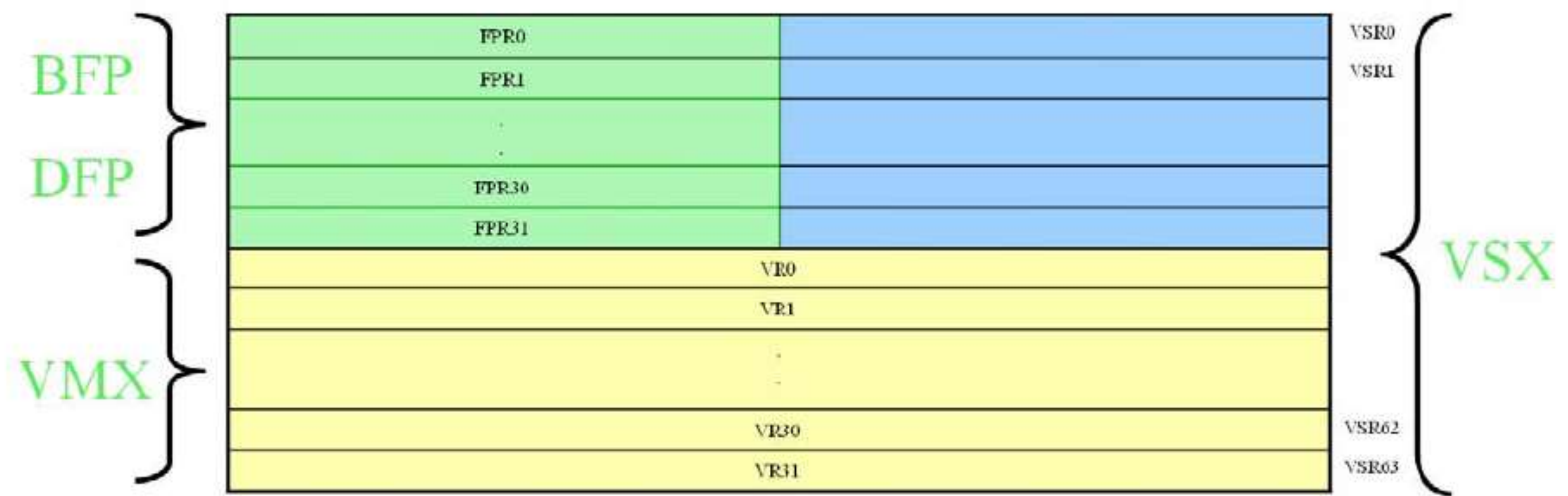


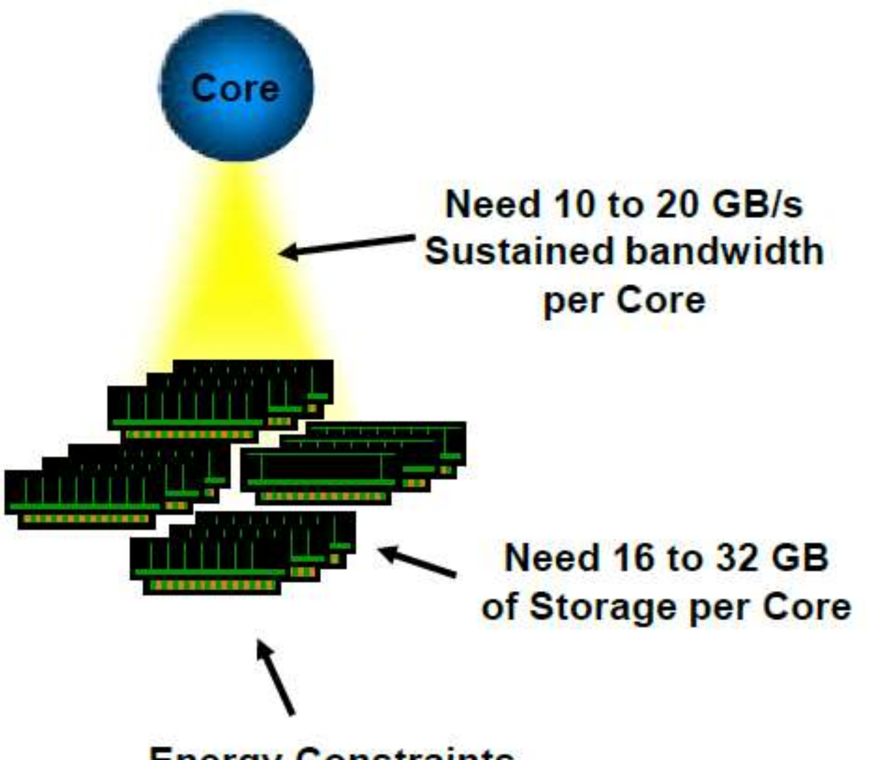
Figure: The unified Register File [60]

8.2.3 Enhanced memory system -1

Requirements imposed by multicores for the memory system:

An n-core multicore processors needs n-times more bandwidth and n-times larger memory, as the next Figure indicates.

Memory subsystem requirements for n-core multicore processors



Socket and system challenge vs. the POWER6

Socket Challenge:
4x growth in memory bandwidth and capacity needed per socket.

System Challenge:
Packaging more memory into similar volume with similar energy and cooling constraints.

Figure: Key memory system requirements for multicore processors [61]

Enhanced memory system -2

- Similarly to the previous POWER6 line also POWER7 based systems interconnect memory DIMMs and on-die memory controllers by serial, low pin count high speed channels, as indicated in the Figure below.

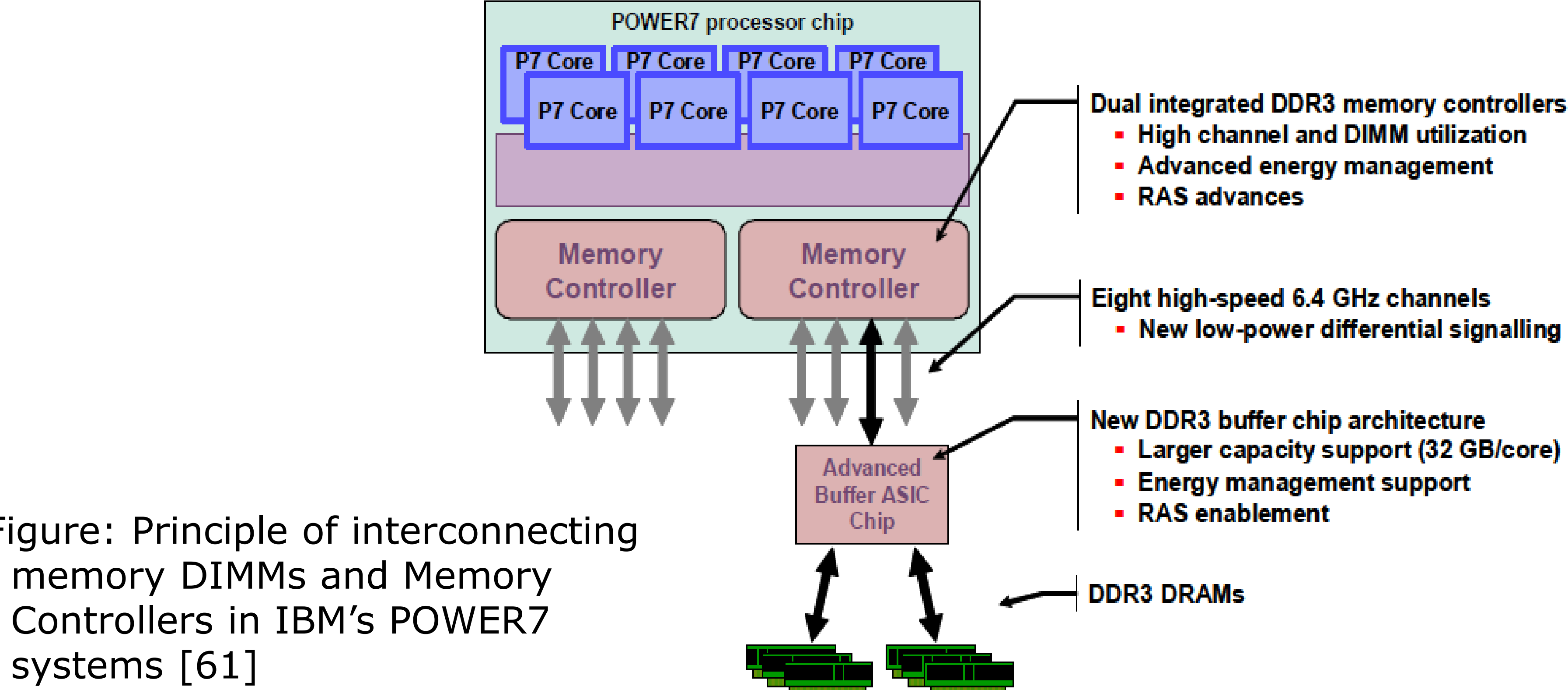


Figure: Principle of interconnecting memory DIMMs and Memory Controllers in IBM's POWER7 systems [61]

- As low pin count serial channels allow to implement obviously more memory channels than high pin count parallel channels in this way memory bandwidth and capacity can be raised.

Enhanced memory system -3

- With serial channels however serial/parallel conversion and a number of further tasks need to be performed at the DIMM side, an appropriate unit, called the Advanced Buffer Chips, need to be inserted onto the DIMMs, as indicated in the Figure above.
- The high speed channels use low power differential signaling with two lines per signal (called a lane) and operate at 6.4 GHz.
- Different models of the POWER7 line implement two different types of memory subsystems, in a similar way as in case of POWER6 systems, as follows:
  - low and midrange models (up to the Power 760) include two ported Advanced Buffer Chips to connect up to two commodity (industry standard) DDR3-1066 RDIMMs to the system, whereas
  - high-end models (Power 770/780/795) use the SuperNova FB-DIMM technology to provide more memory capacity and make use of IBM's proprietary 276 pin DDR3-1066 DIMMs,

as indicated in the next Figure.

8.2.3 Enhanced memory system (4)

Memory configurations of the POWER7

Memory configurations of the POWER7

Commodity DIMM based configurations

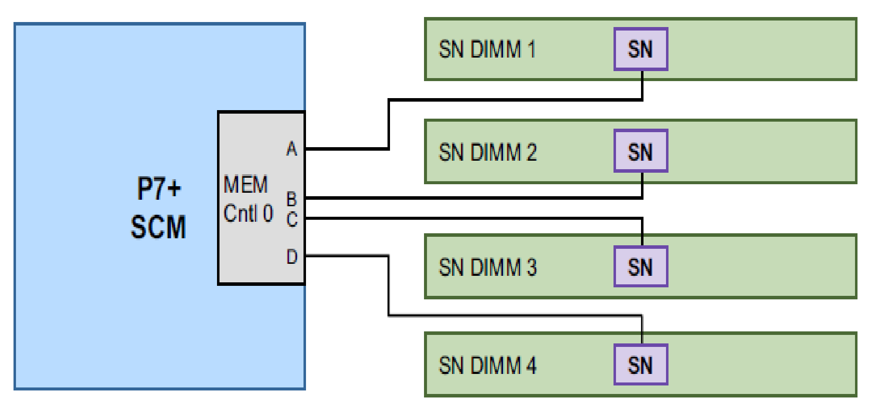
Advanced Buffer Chips are placed on the proc. or memory board, and standard 240 pin DDR3-1066 RDIMMs are used

FB-DIMM-based configurations

Advanced Buffer Chips are integrated to the proprietary 96 mm tall 276 pin DDR3-1066 DIMMs, SuperNova technology is used

POWER7
MC0 Port 0
MC0 Port 1
MC0 Port 2
MC0 Port 3
Port A SN-D Port B
DDR3 RDIMM Slot 3
DDR3 RDIMM Slot 1
Memory Card #2
Port A SN-C Port B
DDR3 RDIMM Slot 4
DDR3 RDIMM Slot 2
Port A SN-B Port B
DDR3 RDIMM Slot 3
DDR3 RDIMM Slot 1
Memory Card #1
Port A SN-A Port B
DDR3 RDIMM Slot 4
DDR3 RDIMM Slot 2

MC: Memory Controller
SN: Memory Buffer



(Strongly simplified)

Examples: (RDIMMs into slots)

Power 710/730 (1 MC/4 ports/ 2 DIMM channels)
Power 720/740 (1 MC/4 ports/ 2 DIMM channels)
Power 750/755 (1 MC/4 ports/2 DIMM channels)

Power 770/780 (2 MC/4 ports/1 FB-DIMM channel)
Power 795 (2 MC/4 ports/1 FB-DIMM channel)

The DIMMs used in POWER7 memory subsystems

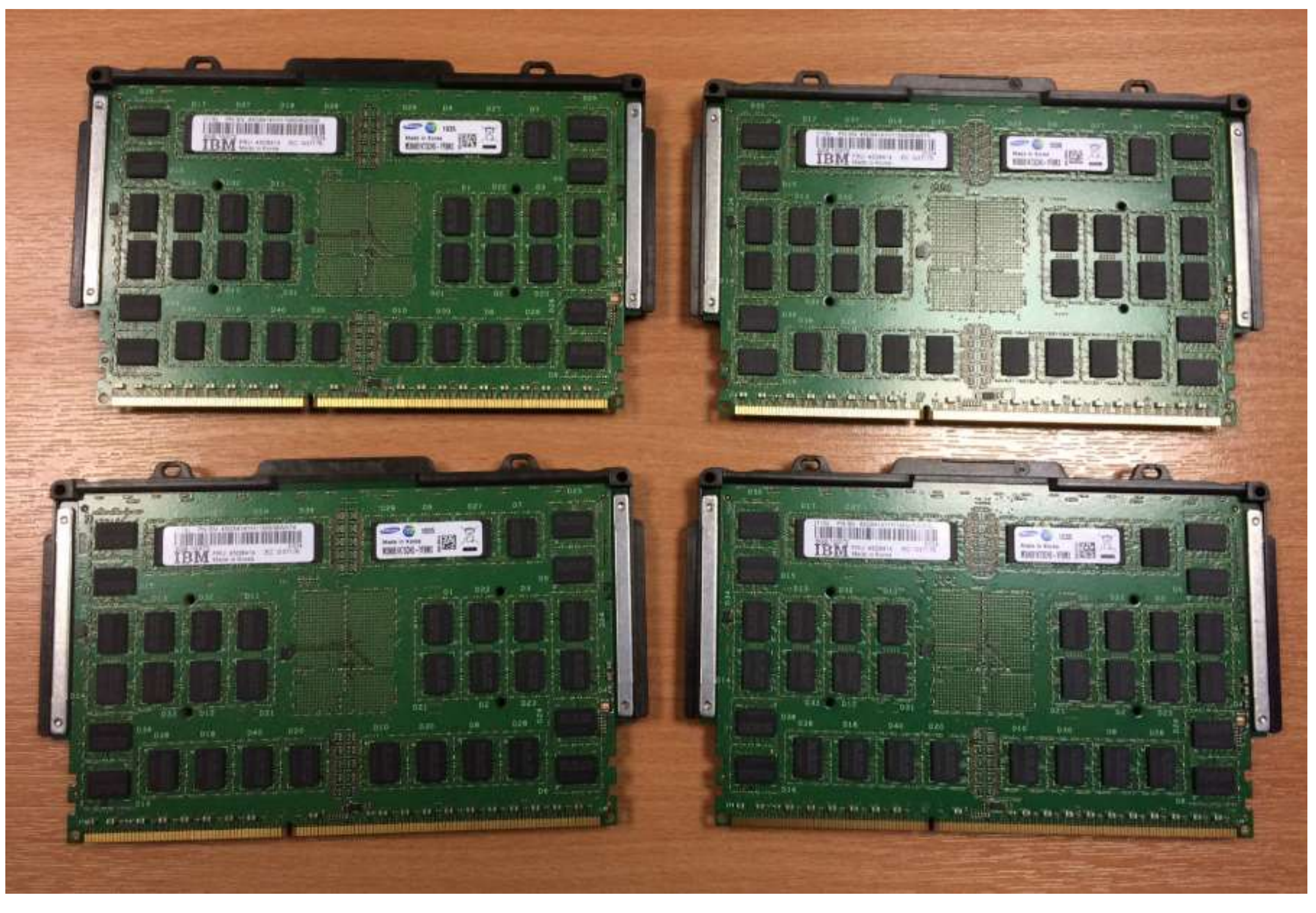
Memory configurations of the POWER7

	Commodity DIMM based configurations	**FB-DIMM-based configurations**
Examples	Power 710/730 Power 720/740 Power 750/755	Power 770/780 Power 795
DIMMs	-- 4 GB (2x 2 GB), 1066 MHz (EM04) -- 8 GB (2x 4 GB), 1066 MHz (EM08 or 4526) – 16 GB (2x 8 GB), 1066 MHz (EM16 or 4527) – 32 GB (2x 16 GB), 1066 MHz (EM32 or 4528) --16 GB (2x 8 GB), 1066 MHz (4529)	– 32 GB (4x 8 GB), 1066 MHz (#5600) – 64 GB (4x16 GB), 1066 MHz (#5601) – 128 GB (4x 32 GB), 1066 MHz (#5602) -- 256 GB (4x 64 GB), 1066 MHz (#5564)

E.g. 4526 8 GB (2x4GB) RDIMM, 240 pin

E.g. #5600 32 GB (4X8 GB) DDR3 DIMMs, 276 pin

Example proprietary 96 mm tall 276 pin DDR3-1066 DIMMs
(4 x 8 GB FC 5600) [Source: ebay]



The commodity DIMM based configuration

- The POWER7 has two memory controllers, nevertheless low and midrange models use typically only a single one.
- Each memory controller provides four high speed serial channels, as seen in the Figure below, showing a memory system with a single memory controller.
- Each high speed channel is connected to a Memory Buffer chip (designated as SN-A to SN-D in the Figure).
- Each Buffer chip has two ports, each one for a DDR3-1066 RDIMM.
- It follows that a server with a single memory controller providing four ports (like the Power 710 in the Figure) may have up to 8 RDIMMs.
- The high speed serial channel operates at 6.4 Gb/s speed.

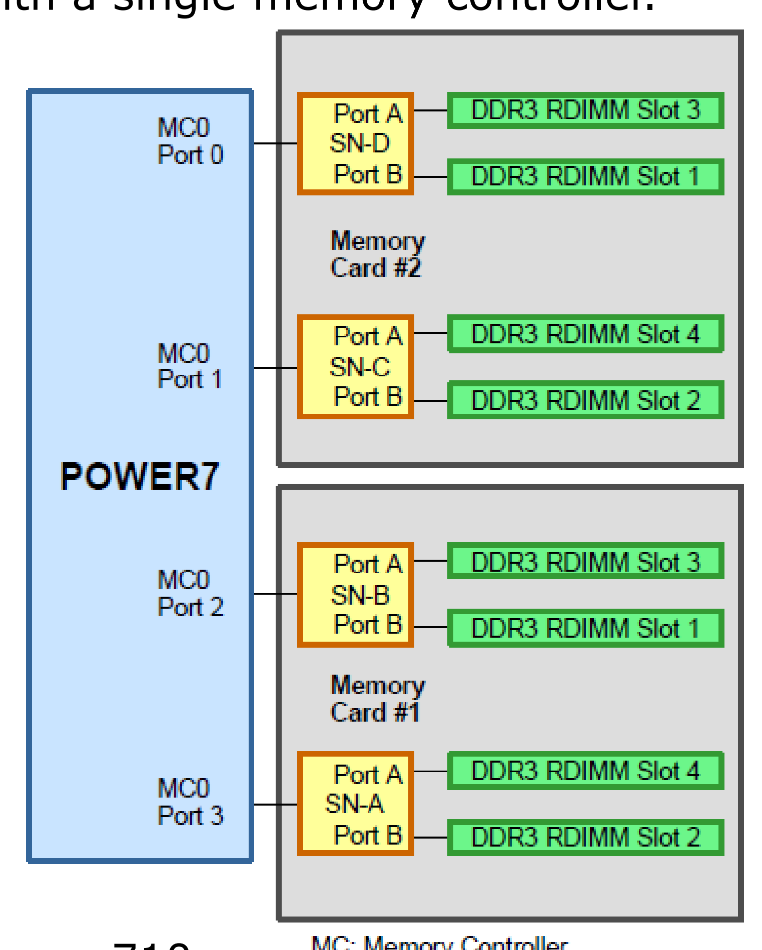


Figure: Layout of the memory subsystem of the Power 710

8.2.3 Enhanced memory system (8)

Per socket bandwidth of commodity DIMM based memory subsystems

The maximum per socket memory bandwidth of commodity DIMM based memory subsystems is constrained both by the serial bus and the available DIMMs.

The serial channel constrained bandwidth of commodity DIMM based memory subsystems (Power 710 – 755) with

- 1 memory controller and
- 4 ports per controller is:

1 memory controller x 4 ports x (1 B write + 2 B read) x 6.4 Gb/s = 76.8 GB/s

The DIMM constrained bandwidth of commodity memory subsystems (Power 710 – 755) with

- a single memory controller,
- 4 ports per controller and
- dual RDIMMs per Memory Buffer is:

1 memory controller x 4 ports x 2 RDIMMs x 8 B x 1066 Mtransfers/s = 68.25 GB/s

The FB-DIMM based configuration

- To provide more bandwidth and memory in POWER7 systems IBM redesigned the memory subsystem of POWER6 introducing SuperNova DIMMs.
- SuperNova DIMMs are proprietary fully buffered DIMMs with DDR3-1066 DRAM chips, replacing the previous industry standard DDR2-667 based FB-DIMMs used in the POWER6, as shown below.

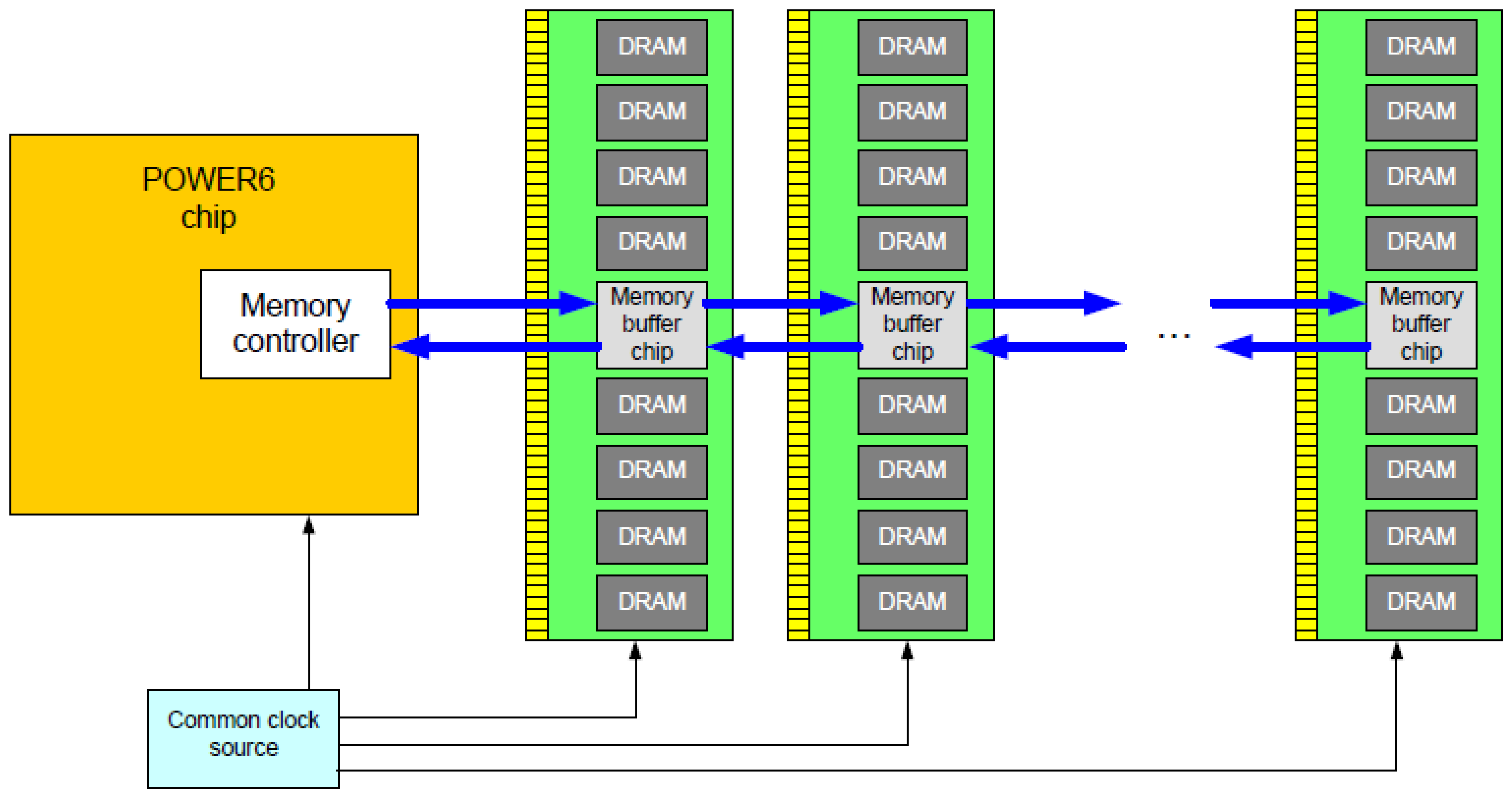


Figure: Layout of an FB-DIMM based POWER6 memory implementation [34]

Redesigned memory subsystem based on SuperNova DIMMs [62]

- SuperNova DIMMs include a SuperNova Buffer Chip (BC) on the DIMM that is interconnected with the memory controller by a high speed Memory Channel, as shown below.

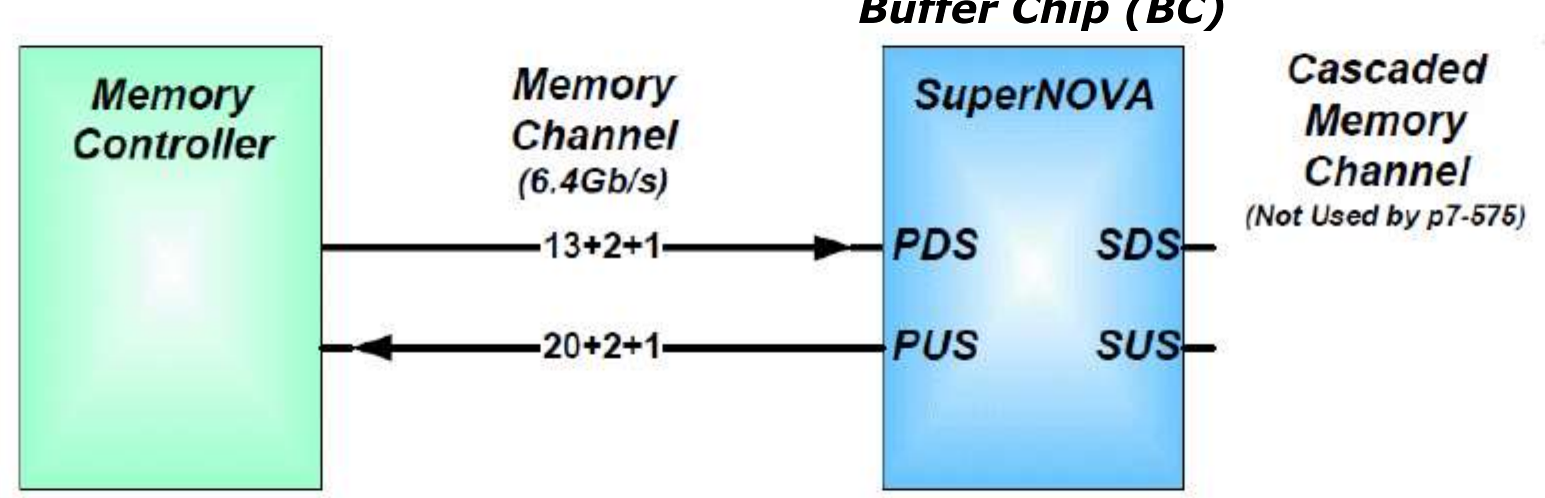


Figure: Principle of connecting SuperNova DIMMs to a Memory Controller [62]

- In the SuperNova technology multiple Buffer Chips may be cascaded, as shown above.

The high speed SuperNova memory channel -1 [62]

It is a point-to-point serial channel using differential lanes that consists of two unidirectional links, as follows:

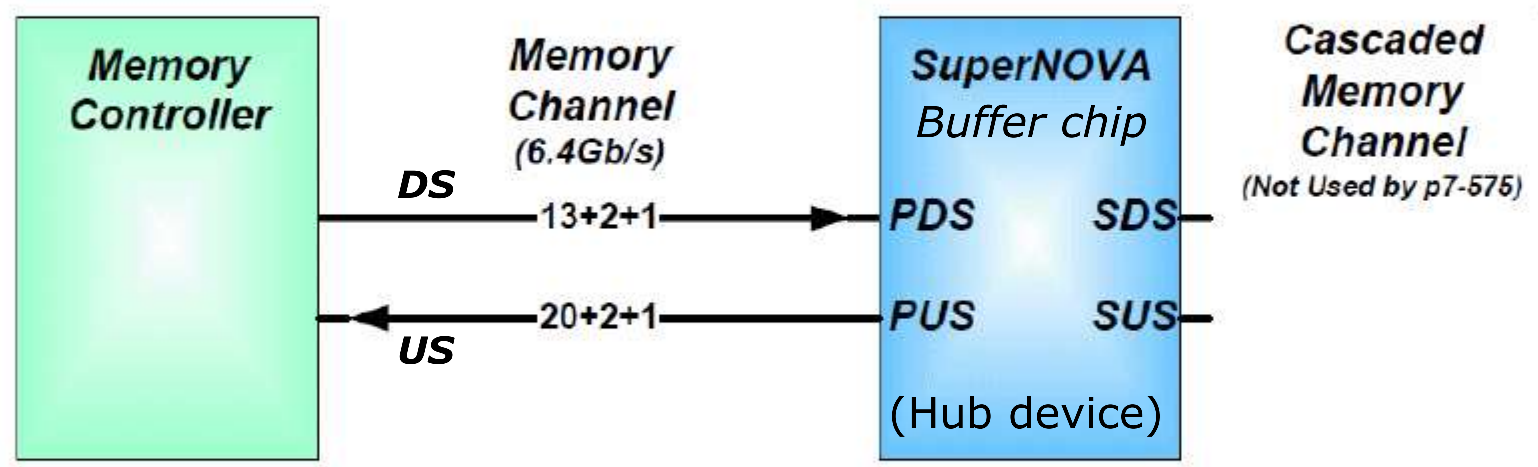


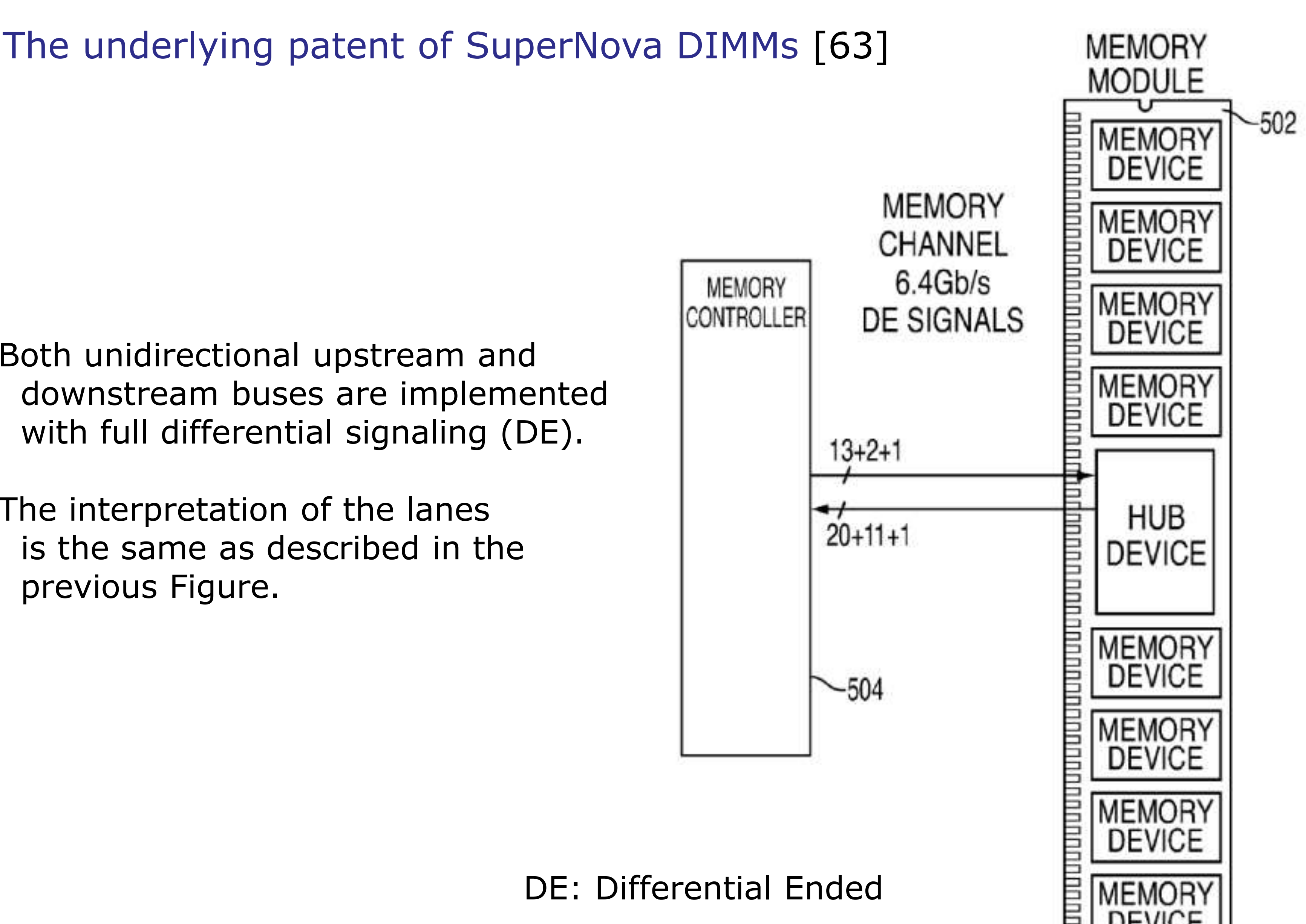
Figure: Principle of connecting SuperNova DIMMs to a Memory Controller [62]

- The downstream link (DS) transmits write data and commands from the memory Controller to the SuperNova DIMM.
- It includes 13 active lanes, two more spare lanes and a bus clock.
- The upstream (US) link forwards read data and responses from the SuperNova DIMMs back to the host.
- It includes 20 active lanes, two more spare lanes and a bus clock.

The underlying patent of SuperNova DIMMs [63]

Both unidirectional upstream and downstream buses are implemented with full differential signaling (DE).

The interpretation of the lanes is the same as described in the previous Figure.



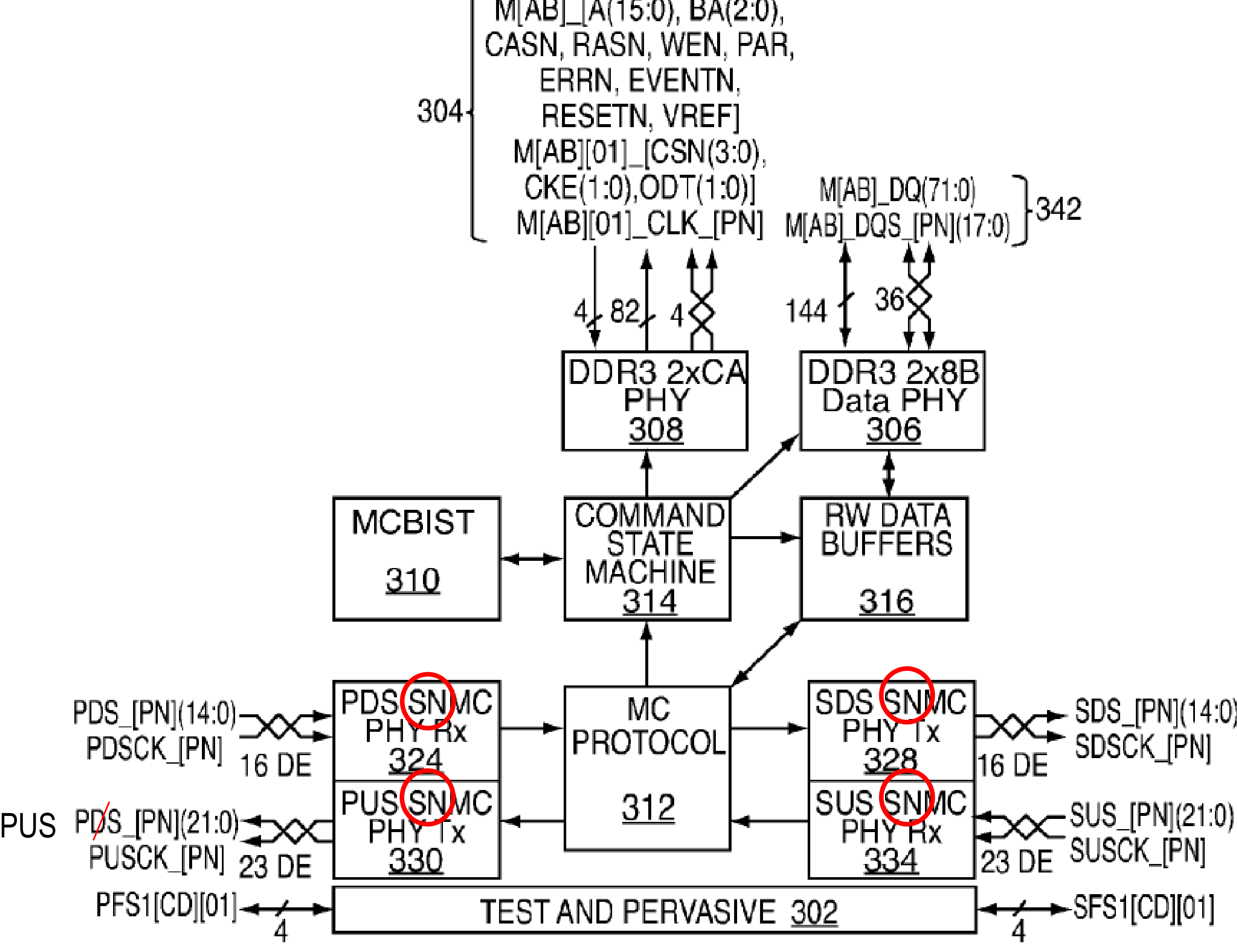
DE: Differential Ended

8.2.3 Enhanced memory system (13)

Principle of operation

- According to the patent the high speed data buses may transfer information at a rate that is four times faster than the data rate of the memory module.
- The signals transferred are in a packetized format and it may require several transfers (e.g. 4, 6 or 8) to receive a packet.
- For packets intended for a given memory module, the signals received in the packetized memory interface format are converted into a memory module interface format by the Hub device once sufficient information transfers are received to permit communication with the memory device.
- Assuming cascaded DIMMs the signals received in the packetized memory interface format are re-driven on the high speed memory buses.

Signal names on a SuperNova DIMM and indication of the SuperNova designation [63]

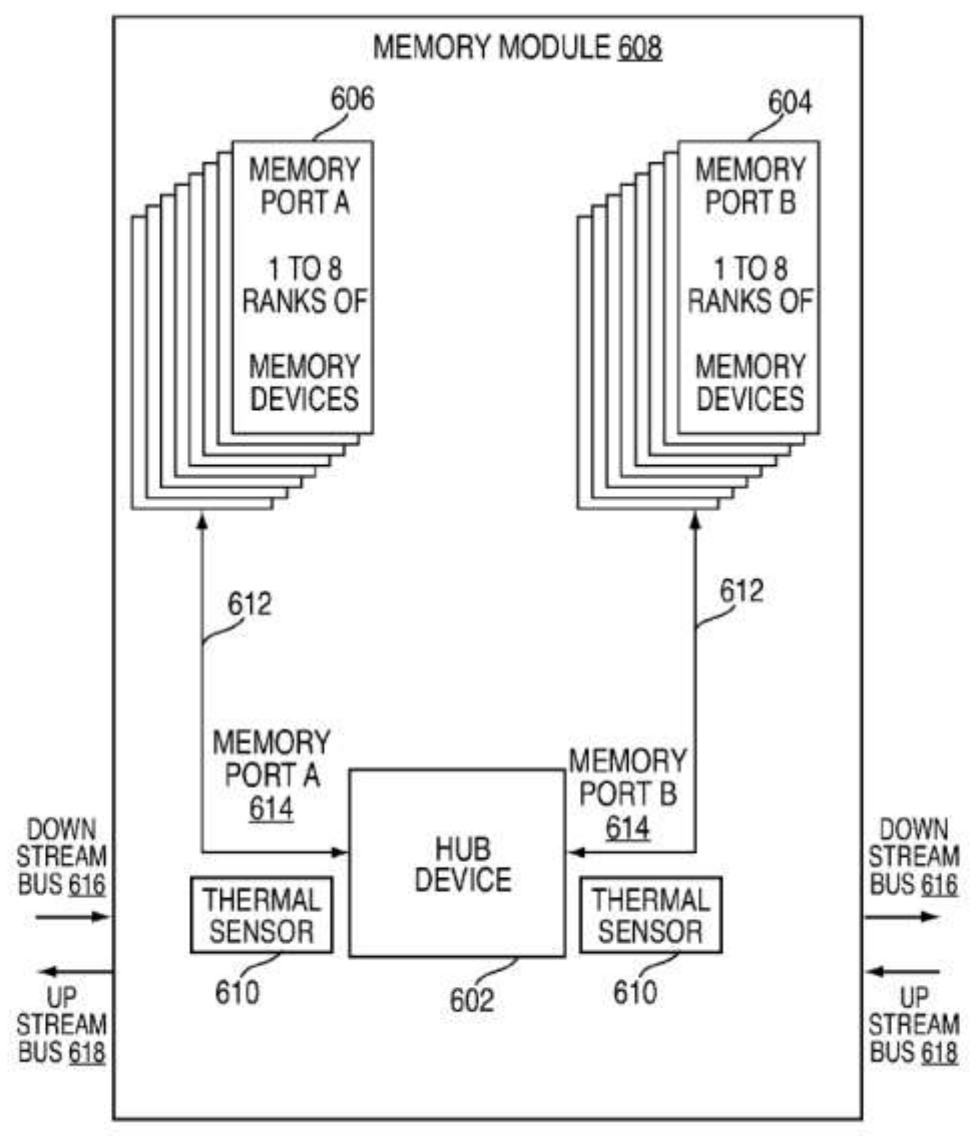


Part of the pin assignment on the DIMM, indicating why 276 pins are used [6

276 Pin Buffered Memory Module Pin Assignments (1 of 2)

TOP Pin Number	BOTTOM Pin Number	TOP Pin Function	BOTTOM Pin Function	Distance From Center Notch (or Key) left or right of notch	
1	139	VREFDQA_TST	1.5V	85.5	Distance To left of notch (front view)
2	140	GND	GND	84.5	
3	141	PUS0	SUS0	83.5	
4	142	/PUS0	/SUS0	82.5	
5	143	1.5V	1.5V	81.5	
6	144	PUS1	SUS1	80.5	
7	145	/PUS1	/SUS1	79.5	
8	146	GND	GND	78.5	
9	147	PUS2	SUS2	77.5	
10	148	/PUS2	/SUS2	76.5	
11	149	1.5V	1.5V	75.5	
12	150	PUS3	SUS3	74.5	
13	151	/PUS3	/SUS3	73.5	
14	152	GND	GND	72.5	
15	153	PUS4	SUS4	71.5	
16	154	/PUS4	/SUS4	70.5	
17	155	1.5V	1.5V	69.5	
18	156	PUS5	SUS5	68.5	
19	157	/PUS5	/SUS5	67.5	
20	158	GND	GND	66.5	
21	159	PUS6	SUS6	65.5	
22	160	/PUS6	/SUS6	64.5	
23	161	1.5V	1.5V	63.5	

Two port memory implementation [63] -1



8.2.3 Enhanced memory system (17)

Two port memory implementation [63] -2

- In an implementation alternative the Hub device can support two memory ports, both connecting to a group of up to eight ranks of memory devices.
- The Hub device has separate address, command, control and data connections with each memory port.
- Each rank includes eight bytes of data as well as eight ECC bits for error correction.
- According to the cited patent both memory ports can be operated simultaneously and independently of each other.

Remark

In contrast, standard FB-DIMMs have a different interconnection scheme than SuperNova DIMMs, as shown below.

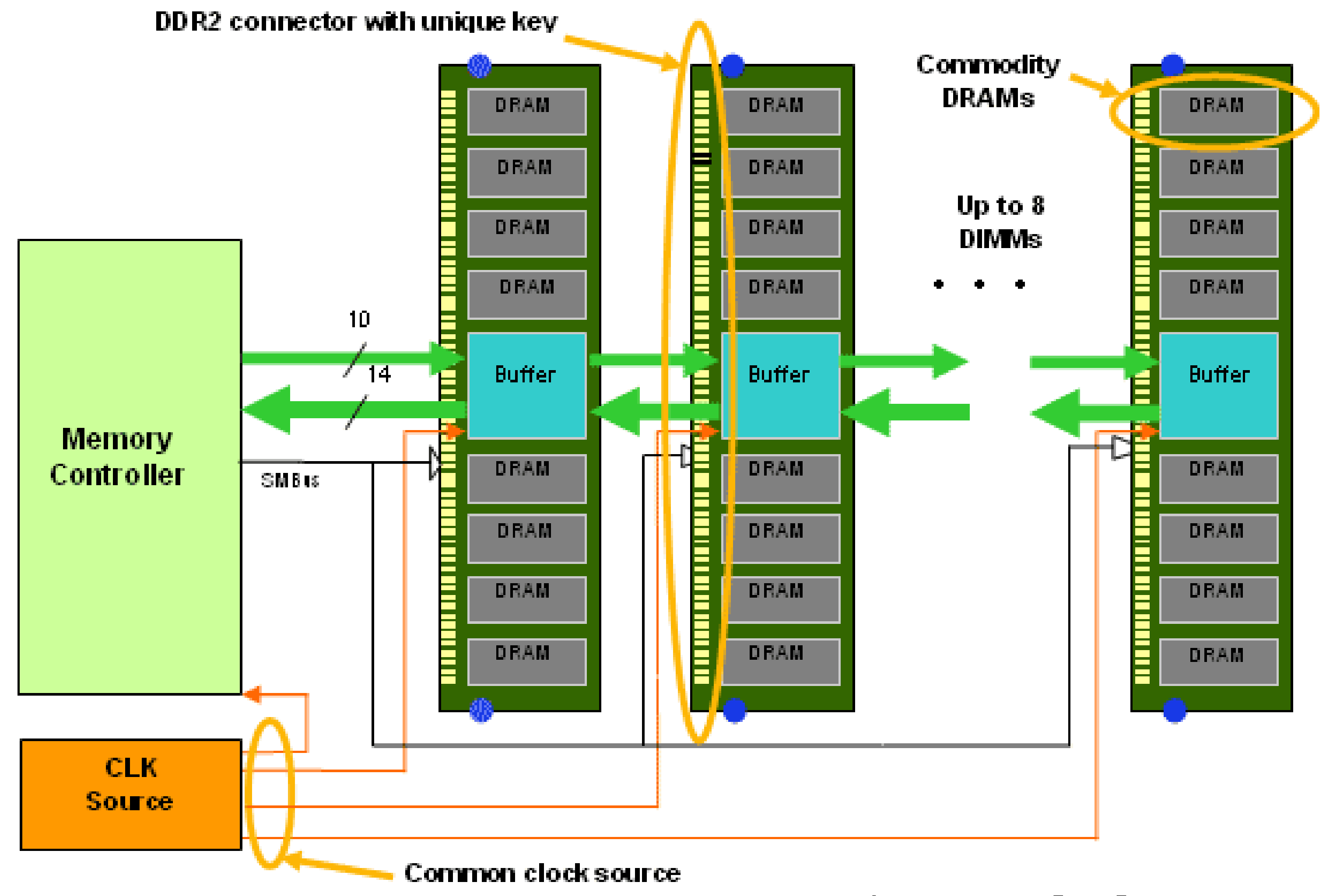


Figure: FB-DIMM memory architecture [35]

The high speed SuperNova memory channel -2 [35]

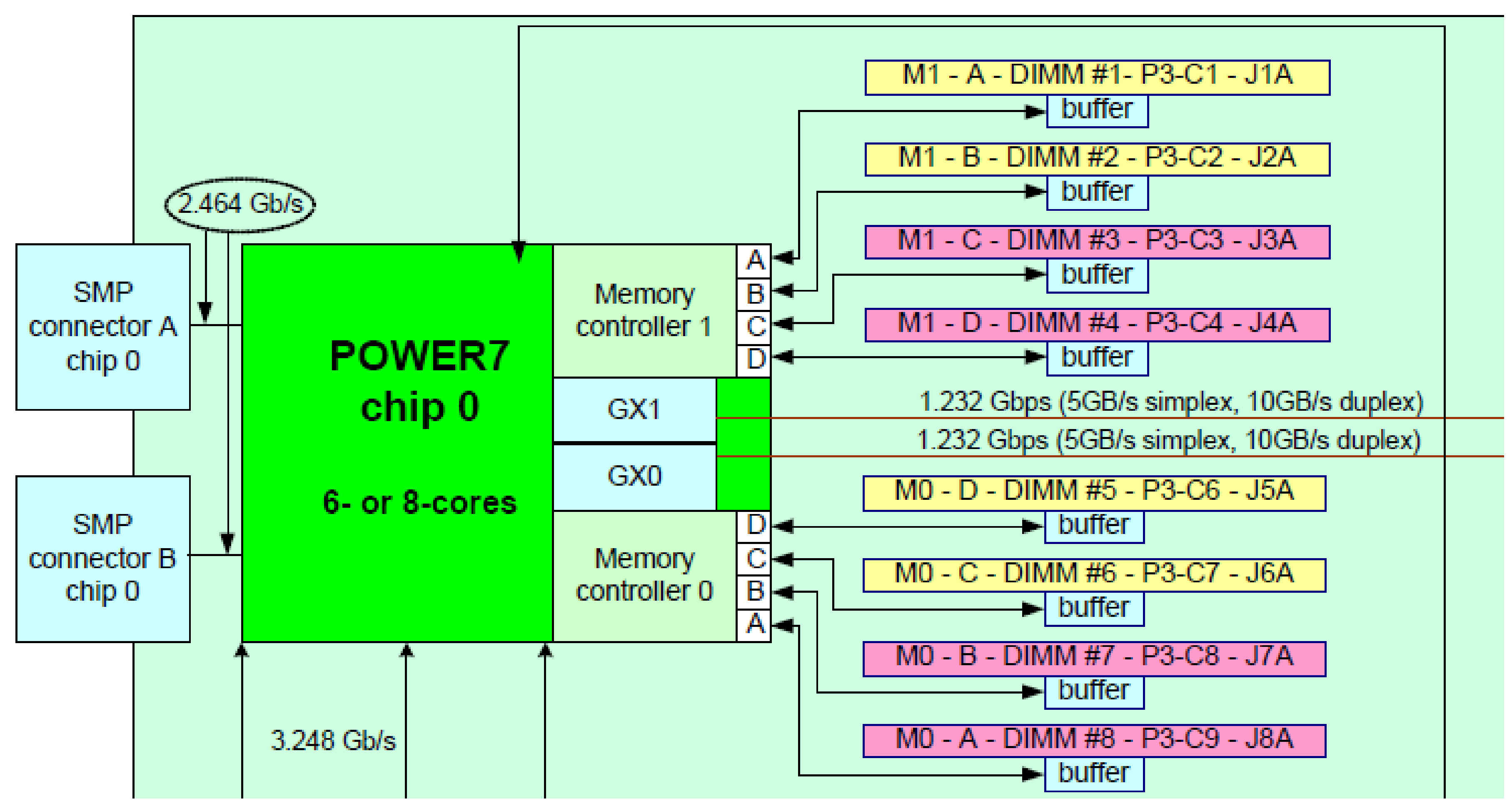
- The maximum speed of a memory channel is 6.4 GHz implementing 4:1 or 6:1 configurable data rate ratio between the Memory controller and DIMMs.
- As an example DDR-1066 RDIMMs with a data rate of 6:1 have a transfer rate of 6x 1066 = 4 Gb/s in the high speed memory channel.

The SuperNova Buffer Chip

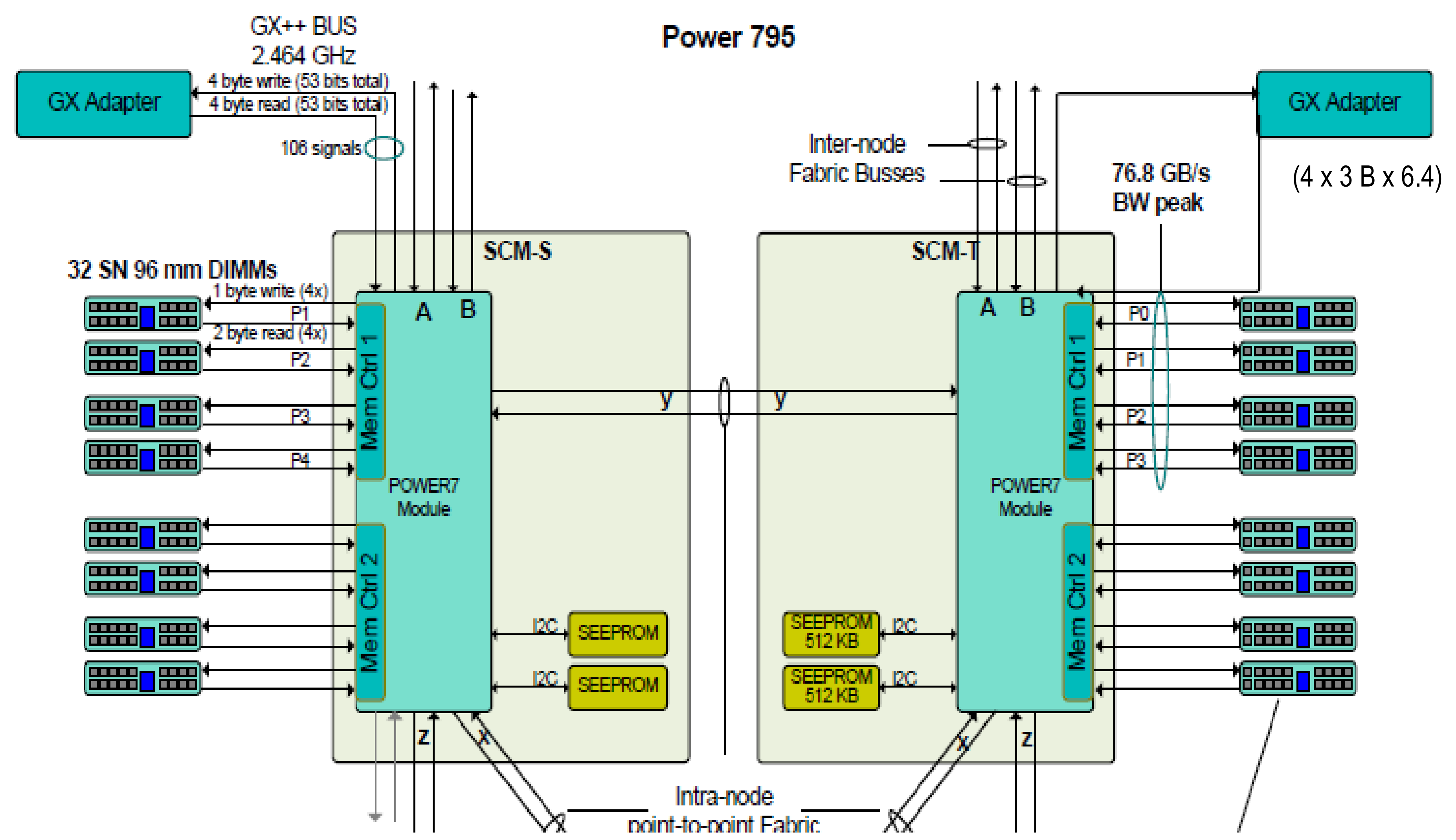
- SuperNova buffer chips provide two ports, each for connecting two cascaded industry standard DDR3-1066 DIMMs, nevertheless both high-speed models using Supernova Buffer Chips utilize only a single DDR3-1066 DIMM channel.
- In this way the high-end POWER7 models referred to with two memory controllers and four buffer chips per controller can service 8 DDR3-1066 memory channels.

  Nevertheless, a layout with two DDR3-1066 DIMM channels per Supernova Buffer chip had 16 memory channels.

Example 1: Part of the SuperNova FB-DIMM based memory subsystem of the two socket Power 770/780 servers [128]



Example 2: Part of the SuperNova FB-DIMM based memory subsystem of a 4 socket "processor book" of the Power 795 server [124]



Per socket bandwidth of FB-DIMM based memory subsystems

The maximum per socket memory bandwidth of FB-DIMM based memory subsystems is constrained both by the serial bus and the available DIMMs.

The serial channel constrained bandwidth of FB-DIMM based memory subsystems (Power 770/780/795) with

- 2 memory controllers and
- 4 ports per controller is:

2 memory controllers x 4 ports x (1 B write + 2 B read) x 6.4 GHz = 153.6 GB/s

The DIMM constrained bandwidth of FB-DIMM based memory subsystems (770/780/795) with

- dual memory controllers,
- 4 ports per controller and
- single RDIMM per SuperNova Memory Buffer is:

2 memory controllers x 4 ports x 1 RDIMM ch. x 8 B x 1066 Mtransfers/s = 68.3 GB/s

Memory bandwidth increase of POWER7 based systems vs. the previous POWER6 based systems

POWER6 based servers	Serial bus constrained BW	DIMM/FB-DIMM constrained BW
Commodity DIMM based memory subsystems	32.0 GB/s	42.67 MB/s
FB-DIMM based memory subsystem	64.0 GB/s	85.35 GB/s

POWER7 based servers	Serial bus constrained BW	DIMM/FB-DIMM constrained BW
Commodity DIMM based memory subsystems	76.8 GB/s	68.25 GB/s
FB-DIMM based memory subsystem	153.6 GB/s	68.25 GB/s

All in all IBM realized in the POWER7 an about 2.5-ofold memory bandwidth boost in the more limiting serial bandwidth figures to meet the fourfold increase of the core count compared to the POWER6.

MC-MB link limited memory bandwidth in IBM's POWER line -1

Model	Tech.	Intro.	No. of cores (up to)	fc up to	SMT	DIMM type	DRAM speed	No/speed/width of MC-MB links (up to)	MC-MB link limited BW/proc. (up to)	BW/fc/ /core (byte/cycle) (up to)
POWER 3-II	250 nm	1999	1	0.45 GHz	No	Propr. DIMM	SDRAM - 100	2@100 Kbits/s 8B R/W	1.6 GB/s	3.5
POWER 4	180 nm	2001	2	1.3 GHz	No	Propr. DIMM	DDR-200	8@400 Kbit/s 4B R/W	12.8 GB/s	4.9
POWER 4+	130 nm	2002	2	1.7 GHz	No	Propr. DIMM	DDR-200	8@400 Kbit/s 4B R/W	12.8 GB/s	3.5
POWER 5	130 nm	2004	2	1.9 GHz	2-way	Propr. DIMM	DDR2-533	4@1066 Kbit/s 4B R/2B W	25.6 GB/s	6.8
POWER 5+	90 nm	2005	2	2.3 GHz	2-way	Propr. DIMM.	DDR2-533	4@1066 Kbit/s 4B R/2B W	25.6 GB/s	5.5
POWER 6	65 nm	2007	2	5.0 GHz	2-way	Commod. DIMM	DDR2-667	4@2.67 Gbit/s 2B R/1B W	32.0 GB/s	3.2
						FB-DIMM	DDR2-667	8@4.0 Gb/s 12b R/6b W	72.0 GB/s	7.2
POWER 6+	65 nm	2008	2	5.0 GHz	2-way	Commod. DIMM	DDR2-667	4@2.67 Gbit/s 2B R/1B W	32.0 GB/s	3.2
						FB-DIMM	DDR2-667	8@4.0 Gb/s 12b R/6b W	72.0 GB/s	7.2

Commod.: Commodity Prop.: Proprietary MB: Memory buffer (POWER4-7: SMI buff.-POWER8-9: Centaur buff-)

MC-MB link limited memory bandwidth in IBM's POWER line -2

Model	Tech.	Intro.	No. of cores (up to)	fc (up to)	SMT	DIMM type	DRAM speed	No/speed/width of MC-MB links (up to)	MC-MB link limited BW/proc. (up to)	BW/fc//core (byte/cycle) (up to)
POWER7	45 nm	2010	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gbit/s 2B R/1B W	76.8 GB/s	2.2
						Propr. FB-DIMM	DDR3-1066	8@6.4 Gb/s 2B R/3B W	153.6 GB/s	4.4
POWER7+	32 nm	2013	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	153.6 GB/s	3.9
						Propr. FB-DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	76.8 GB/s	2.2
POWER8	22 nm	2014	12	4.35 GHz	8-way	Propr. CDIMM	DDR3-1600	2(8)[1]@9.6 Gbit/s 2B R/1B W	57.5 (230 GB/s)	1.1 (4.4)
POWER9 (Scale-Out)	14 nm	2017	12 24	4.00 GHz	8-way 4-way	Commod. DIMM	DDR4-2666	--	--	--
POWER9 (Scale-Up)		2018	12	4.00 GHz	8-way	Commod. DIMM	DDR4-1600	8@9.6 Gbit/s 2B R/1B W	230,4 GB/s	4.79

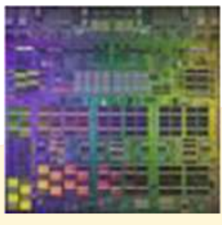
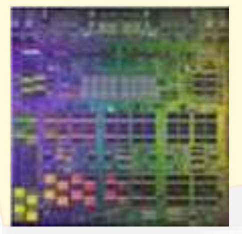
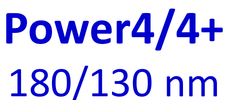
[1]: According to IBM's literature [] the POWER8 has up to eight memory channels.
Nevertheless, first servers delivered until 05/2015 makes use of only two of them.

Commod.: Commodity Prop.: Proprietary MB: Memory buffer (POWER4-7: SMI buff.-POWER8-9: Centaur buff-)

8.3 Key innovations of the POWER7

- 8.3.1 8-core design
- 8.3.2 4-way SMT
- 8.3.3 On-chip L3 cache
- 8.3.4 Ring bus based on-chip interconnect
- 8.3.5 Re-designed EnergyScale power management

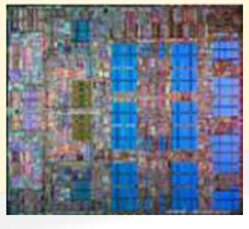
8.3 Key innovations of the POWER7 (Die photos from [3])

Power4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
  with SMI chips
GX I/O bus
Support for SMP

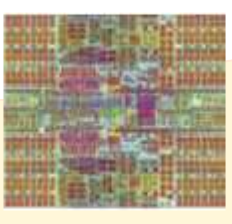
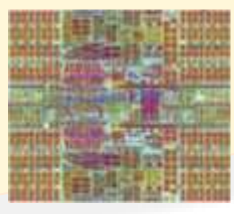
2001



Power5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
  clock gating

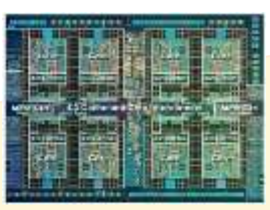
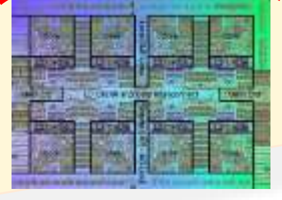
2004

Power6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
  Critical Path
    Monitors
  Nap idle mode

2007

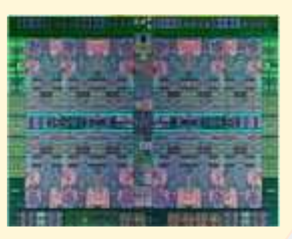



Power7/7+*
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
  Per core fc
  Dyn. fan managm.
  Sleep idle mode
*Accelerators for
  cryptography
*Winkle idle mode

* POWER7+

2010



Power8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
  with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2014

8.3.1 8-core design

8.3.1. 8-core design

The POWER7 incorporates 8 cores, as seen on the die plot of the chip.

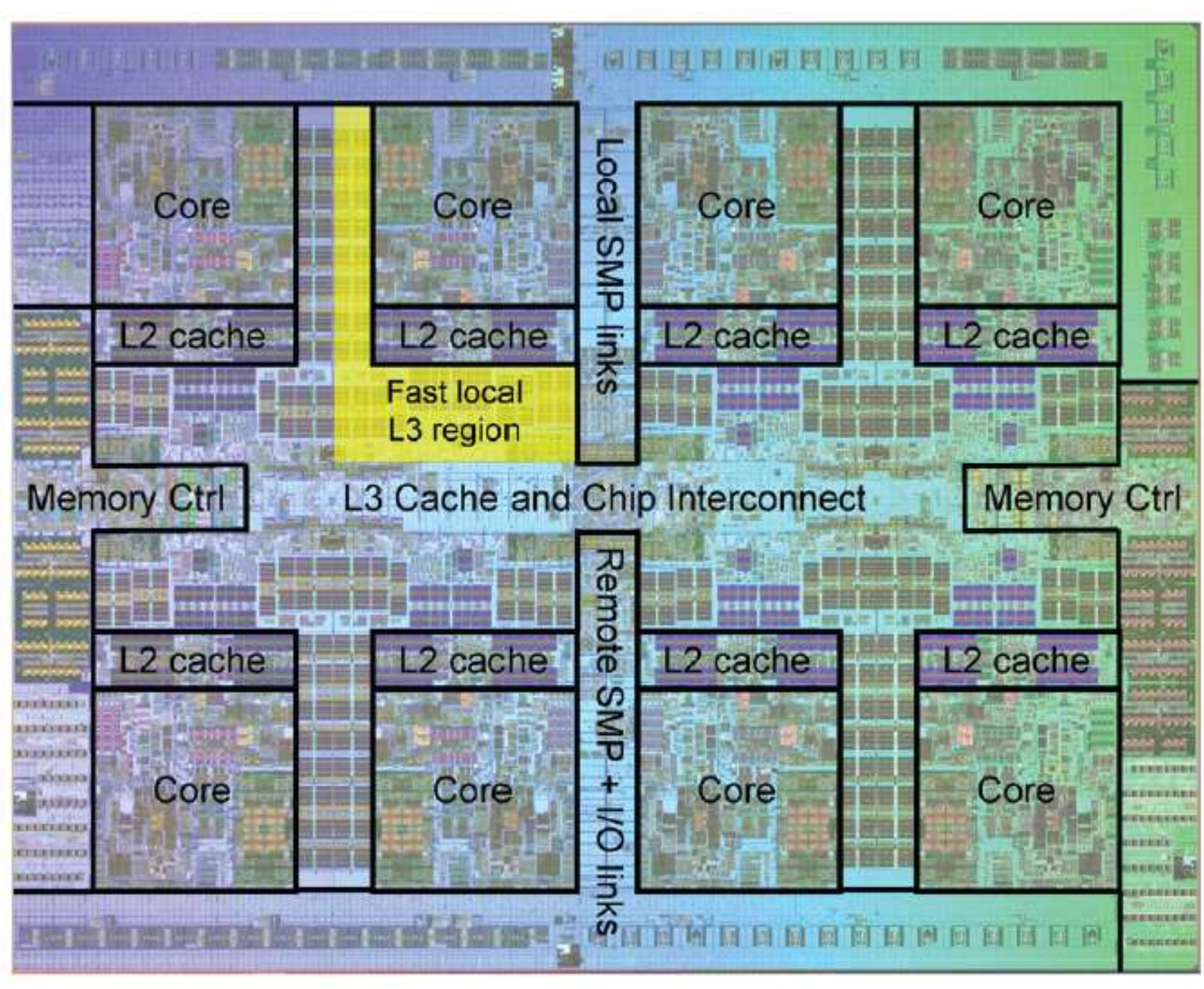


Figure Die plot
of the POWER7 [61]

8.3.2 4-way SMT

8.3.2 Four-way SMT (Symmetrical MultiThreading)

POWER7 adds four-way SMT as an option.

Accordingly, POWER7 provides three thread execution modes, as follows:

- SMT1: Single instruction execution thread per-core
- SMT2: Two instruction execution threads per-core (Two-way SMT)
- SMT4: Four instruction execution threads per-core (Four-way SMT),

as indicated in the next Figure.

Evolution of multithreading in IBM's processors (Based on [64])

IBM PowerPC 601

1993 Single thread out of order

FX0
FX1
FP0
FP1
LS0
LS1
BRX
CRL

IBM AS400 PPC Pulsar (RS64 III)

1997 Hardware multi-thread

FX0
FX1
FP0
FP1
LS0
LS1
BRX
CRL

2001 2 Way SMT

FX0
FX1
FP0
FP1
LS0
LS1
BRX
CRL

IBM POWER4

2010 4 Way SMT

FX0
FX1
FP0
FP1
LS0
LS1
BRX
CRL

IBM POWER7

8.3.2 4-way SMT (3)

Implementation of the four-way SMT in the POWER7 [59]

- In order to reduce power consumption and chip area IBM chose a partitioned SMT-4 design.
- This means that
  - as long as one pair of threads is assigned to one physical general-purpose register (GPR) file that feeds one FX pipeline and one load/store pipeline,
  - another pair of threads is assigned to a separate physical GPR file that feeds a separate FX-pipeline and load/store pipeline

as indicated in the next Figure.

Block diagram of the Instruction-Sequencing Unit (ISU)

Without going into details we show the partitioned implementation of execution resources supporting SMT4.

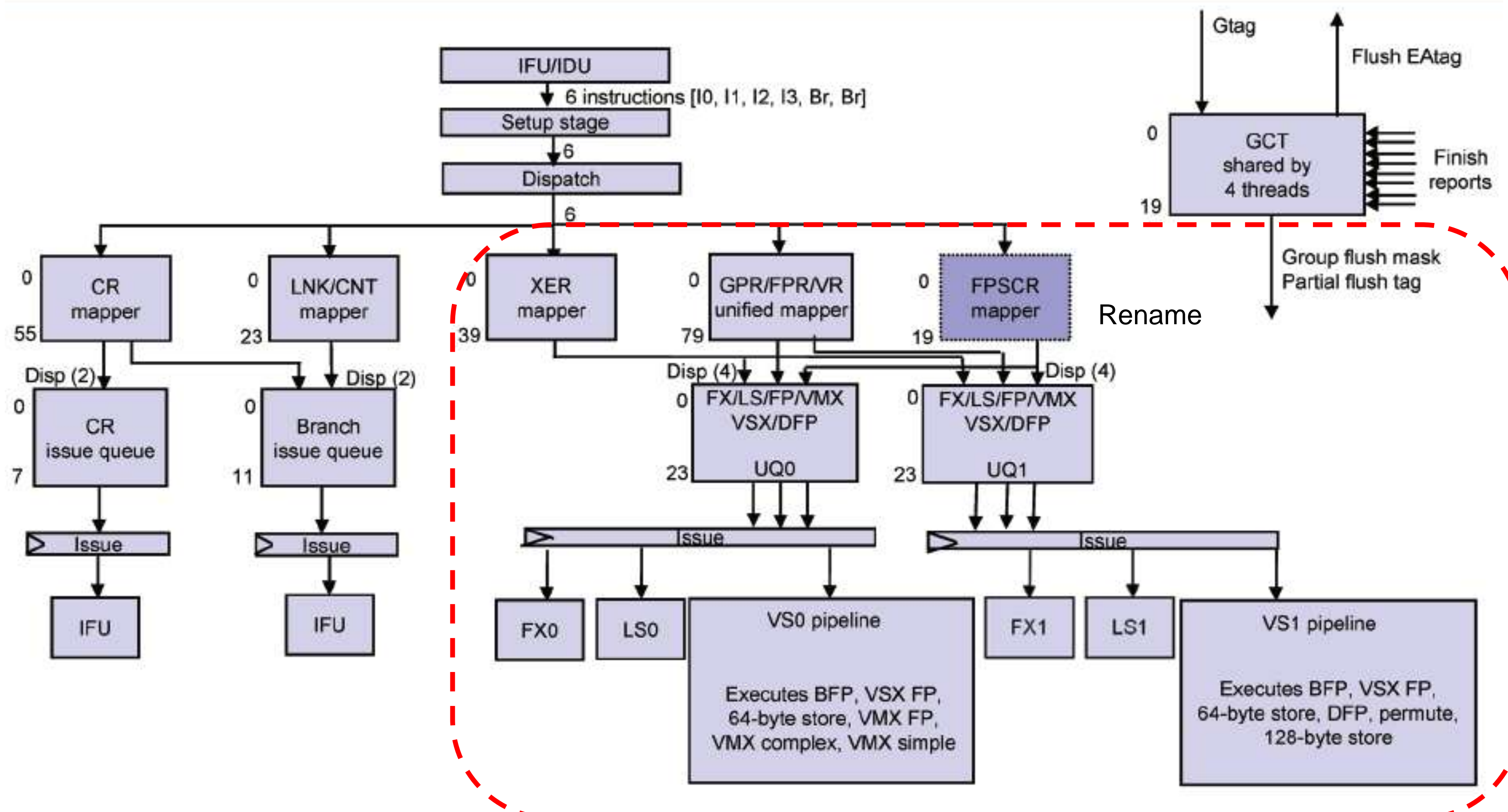


Figure: Block diagram of the Instruction-Sequencing Unit of the POWER7 [59]

8.3.3 On-chip L3 cache

8.3.3 On-chip L3 cache

The main improvements related to the on-die L3 cache are as follows:

8.3.3.1 Use of novel eDRAM cells for the L3 cache

8.3.3.2 Implementation of an on-chip L3 cache

8.3.3.3 Adaptive L3 cache management

8.3.3.4 Smaller per-core L2 caches

8.3.3.5 Resulting cache bandwidth figures

8.3.3 On-chip L3 cache (2)

8.3.3.1 Use of novel eDRAM cells for the L3 cache [59]

- The L3 cache is made up of novel eDRAM cells.
- IBM's novel eDRAM technology allows to implement each storage element by a single transistor instead of 6 required for conventional SRAM circuits.
- As a consequence, unlike conventional SRAMs the eDRAM technology needs only one third the area and dissipates only one fifth of the standby power.
- The resulting area and power reduction was the primary enabler for implementing eight cores on the same chip.

8.3.3.2 Implementation of an on-chip L3 cache

In contrast to the POWER6 chip the POWER7 provides an on-die L3 cache of 32 MB, as indicated below.

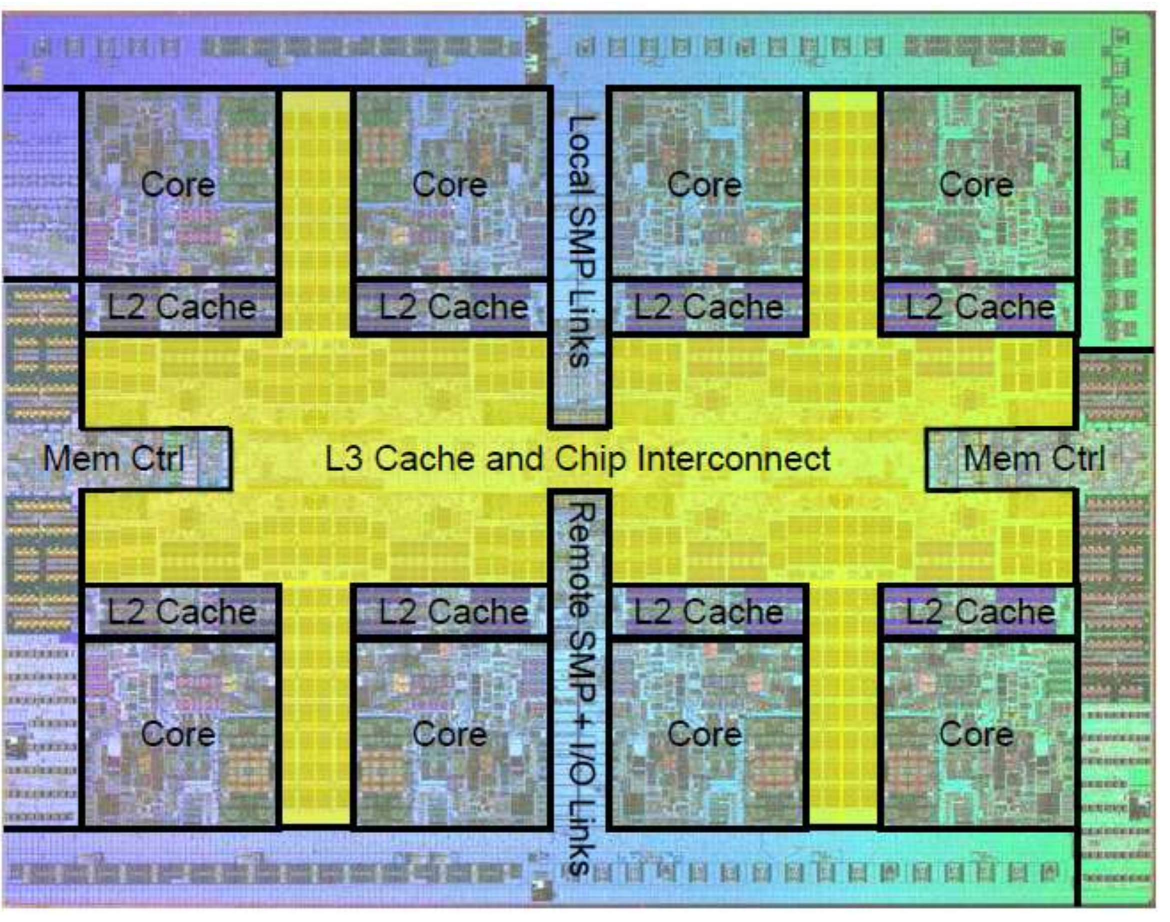


Figure: Die plot of the POWER7 chip [61]

8.3.3.3 Adaptive L3 cache management-1

The L3 cache consists of eight 4 MB L3 regions, termed as Fast Local L3 Regions, tightly coupled to the private L2 associated with each core, as the next Figure shows.

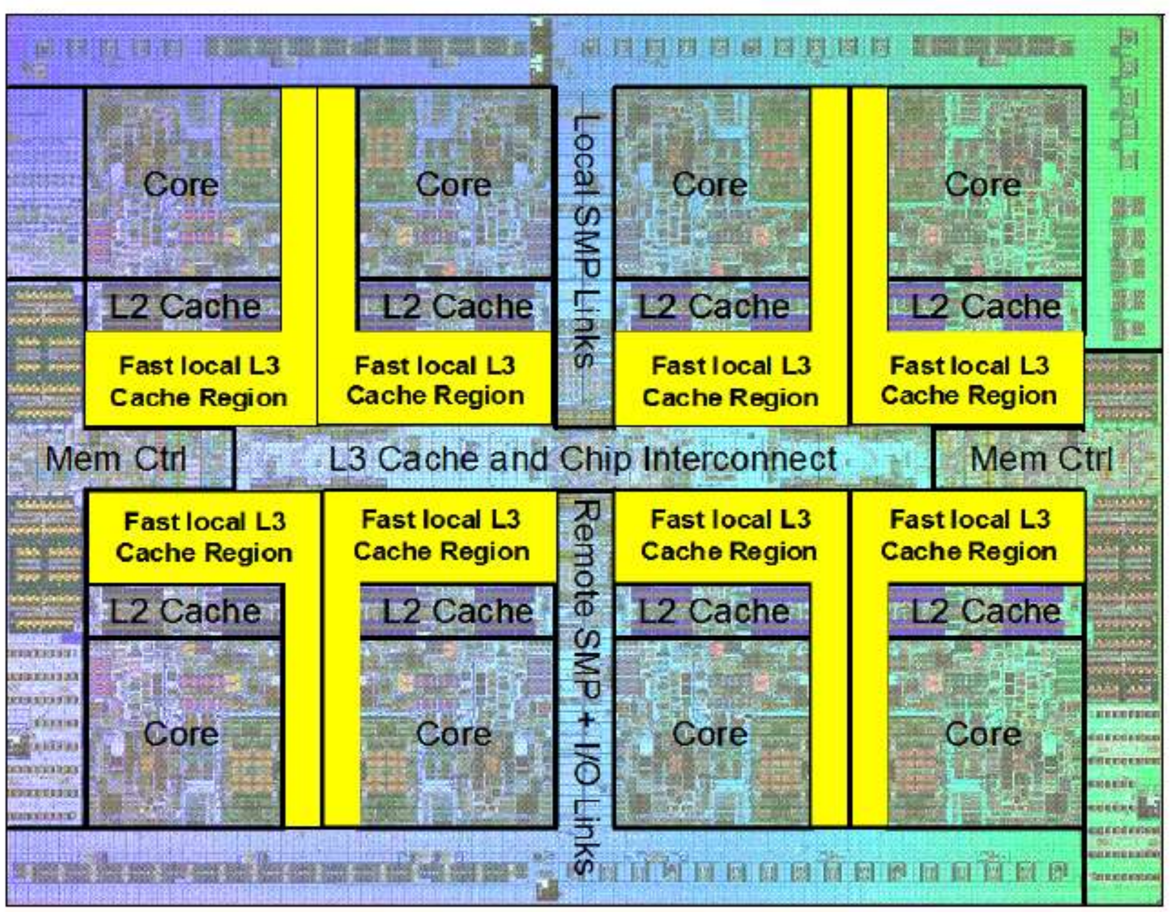


Figure: Fast Local L3 Regions on the POWER7 chip [64]

8.3.3.3 Adaptive L3 cache management-2 [59]

- The L2 cache utilizes the associated 4-MB local L3 region directly as a victim cache.
- Under certain circumstances the other seven 4-MB L3 regions can also serve an L2 cache as a victim cache (not discussed here, for details see the given reference).
- The adaptive L3 cache management policy means strongly simplified as follows:
  - Storage accesses that miss the L2 cache access the associated 4-MB local L3 region via the core/L2 dispatch port.

    Those that hit in the 4-MB local L3 region are serviced by the local L3 region.
  - Storage accesses that miss both the L2 cache and the 4-MB local L3 region are broadcasted to the coherence fabric to be serviced.
  - The coherence fabric is then responsible for performing the storage access.

    Here again we do not want to go into details, but refer to the referenced literature [59].
- The local L3 regions of the POWER7 have a strongly reduced latency of 6.0 ns vs. the 35.0 ns latency of the off-chip L3 cache in the POWER6.

8.3.3.4 Smaller, faster per-core L2 caches

- As long as the POWER6 has 4 MB large per-core L2 caches, in the POWER7 IBM reduced the per-core cache size to 256 kB.
- The reason is that a fast on-chip L3 cache implies a different trade off between cache latency and cache miss than a slow off-chip cache if maximum performance or maximum performance per power consumption is considered.
- A smaller L2 cache has then a lower latency, as the next table indicates.

	POWER6 (fc = 5 GHz) Off-chip L3 (175 cycles latency)	**POWER7 (fc = 4 GHz) On-chip L3 fast region (24 cycles latency)**
L2 size	4 MB	256 kB
L2 latency	25 cycles	8 cycles

Table: Contrasting the size and latency of L2 caches in the POWER6 and POWER7 [59]

8.3.3 On-chip L3 cache (7)

8.3.3.5 Resulting cache bandwidth figures

Compared to POWER6 the POWER7 provides substantially higher cache bandwidth figures to support eight cores and 4-way SMT, as the Table below indicates.

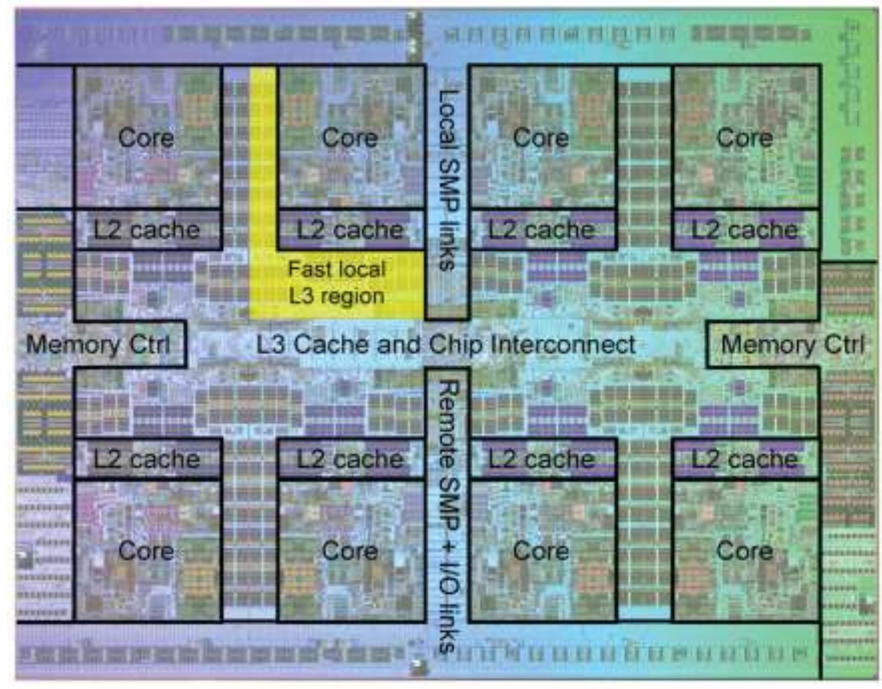
POWER6 (assuming 5-GHz core)	*POWER7 (assuming 4-GHz core)*
64 KB store-through L1 D-cache 0.8ns latency, 80 GB/s private	32 KB store-through L1 D-cache 0.5ns latency, 192 GB/s private
4 MB store-in L2 cache ~5.0-ns latency, 160 GB/s private	256 KB store-in L2 cache 2.0-ns latency, 256 GB/s private
	4 MB partial victim local L3 region ~6.0-ns latency, 128 GB/s private
32 MB victim L3 cache ~35-ns latency, 80 GB/s shared by 2	32 MB adaptive victim L3 cache ~30-ns latency, 512 GB/s shared by 8

Table: Contrasting cache bandwidth figures of the POWER6 and POWER7 [59].

Note that the POWER7 has 512 GB/s L3 bandwidth since the L3 is shared by 8 cores whereas the POWER6 provides 80 GB/s L3 bandwidth for 2 cores.

8.3.4 Ring bus-based on-chip interconnect

8.3.4 Ring bus-based on-chip interconnect [59]



- As the Figure on the right indicates, the POWER7 chip includes
  - eight cores, each of which is associated with an L2 cache and local L3 region,
  - two memory controllers,
  - two I/O controllers and
  - a multichip SMP interconnect that can be scaled up a single large SMP system consisting of 32 POWER7 chips.
- The on-chip interconnect consists of
  - an on-chip coherence interconnect and
  - an on-chip data interconnect.
- The on-chip coherence interface contains several coherence request, snoop, response, and notification buses and operates on a chosen coherence mechanism detailed in [59] but not described in this Section.

The on-chip data interconnect -1

- The on-chip data interconnect has a much more intricate task than previous interconnects since the POWER7 has 8 cores rather than just two.
- As long as previous interconnects were built up as crossbars, the large number of ports needed in this case gave rise to a new interconnect structure, actually, to a ring interconnect.
- Here we chose to describe the on-chip data interconnect of the POWER7 by citing a patent filed in 3/2005 [59] by employees of IBM and Sony.

The on-chip data interconnect -2

- “The on-chip data interconnect consists of eight 16-byte buses that span the horizontal trunk.
- Four flow from left to right, and the other four flow from right to left.

  These buses are bracketed by memory controllers found at the left and right edges of the chip.
- They are divided into multiple segments, such that multiple 16-byte data packets may be pipelined within the multiple segments of the same bus at any given time.
- The buses operate at the on-chip bus frequency.
- Each memory controller has two 16-byte on-ramps and two 16-byte off-ramps that provide access to the eight buses.
- Each core’s associated L2 cache and local L3 region share one 16-byte on-ramp/off-ramp pair, as does the pair of I/O controllers.
- The multichip data interconnect ports, found in the central vertical spines have a total of seven 16-byte on-ramp/off-ramp pairs.
- In total, there are twenty 16-byte on-ramps and twenty 16-byte off-ramps that provide access to and from the eight horizontal 16-byte trunk buses.
- Each ramp pair is associated with a bus segment.

8.3.4 Ring bus-based on-chip interconnect (4)

The on-chip data interconnect -3

- Note that a source-to-destination on-ramp/off-ramp route may consume only a subset of the segments in the horizontal trunk, depending upon the physical locations of the source and destination.
- Data transfers are managed by centralized arbitration logic (see next Figure) that takes into account source and destination locations, allocates available bus segments to plot one of several possible routes, allocates the on- and off-ramp resources, and manages destination data buffers.
- Since transfers may use only a subset of the segments in a given trunk bus, multiple noninterfering source-to-destination transfers may utilize the same horizontal trunk bus simultaneously.
- The arbitration logic must also account for the differing operating frequencies of the processor cores.

  For example, a source core operating at a lower frequency will send data via its on-ramp to the trunk buses at a slower rate.

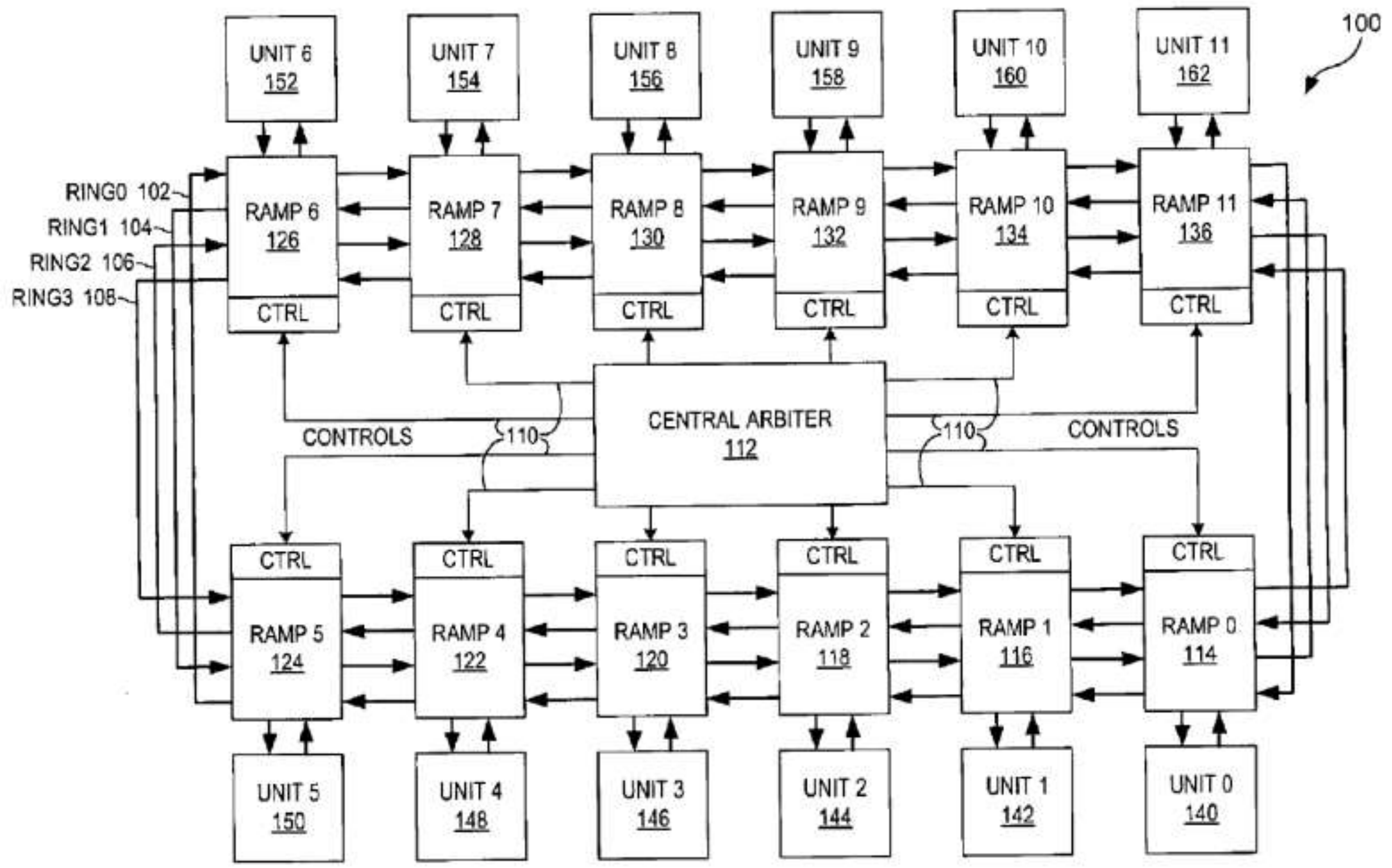
  Likewise, a destination core operating at a lower frequency will consume data via its off-ramp from the trunk buses at a slower rate.

  To manage this, the arbitration logic controls speed-matching buffers in all of the on-ramps/off-ramps."

Remark -1

The data ring operates as stated in the patent application US 2006/0206657 A1 (2005 [136].

A possible data ring implementation with 4 ring buses, 12 units and a central arbiter is shown below.



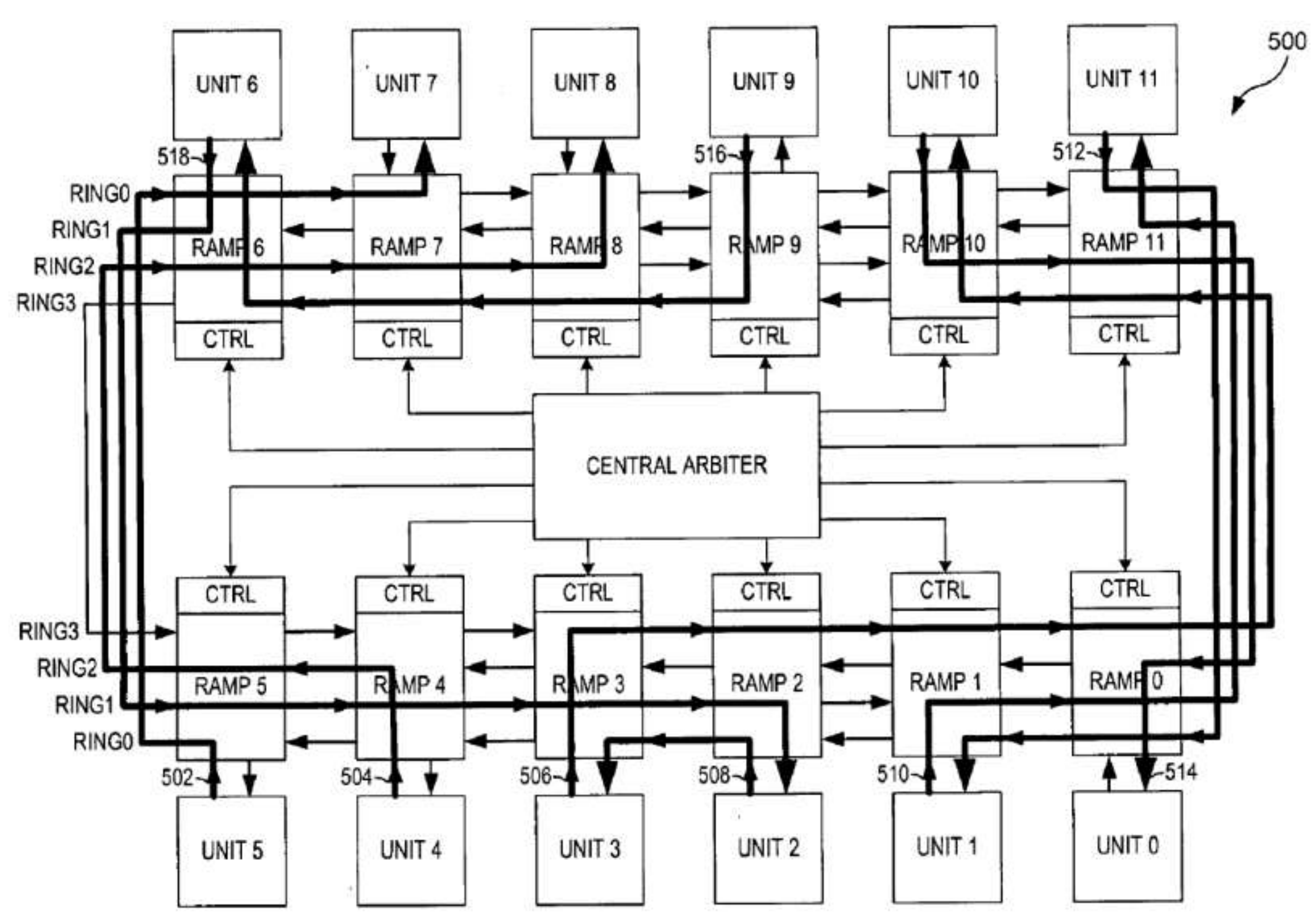
8.3.4 Ring bus-based on-chip interconnect (6)

Remark -2 [136]

According to the patent application the data ring operates as follows:

"For example, if Unit 10 160 wanted to send data to Unit 1/142 it would send out a request. The request reaches the central arbiter 112 and the arbiter 112 begins to send out control signals 110 to the necessary data ramp devices, Ramp 10/134, Ramp 11/136, Ramp 0/114 and Ramp 1/116. The central arbiter 112 also selects an available data ring, which can be Ring 0/102 for this operation. Unit 10/160 receives a grant from the central arbiter 112 and transmits the requested data to Ramp 10/134. Ramp 10/134 uses Ring 0/102 to transmit this data to Ramp 11/136. Ramp 11/136 allows this data to pass through to Ramp 0/114. Ramp 0/114 allows this data to pass through to Ramp 1/116. Ramp 1/116 accepts this data and transmits the requested data to Unit 1/142."

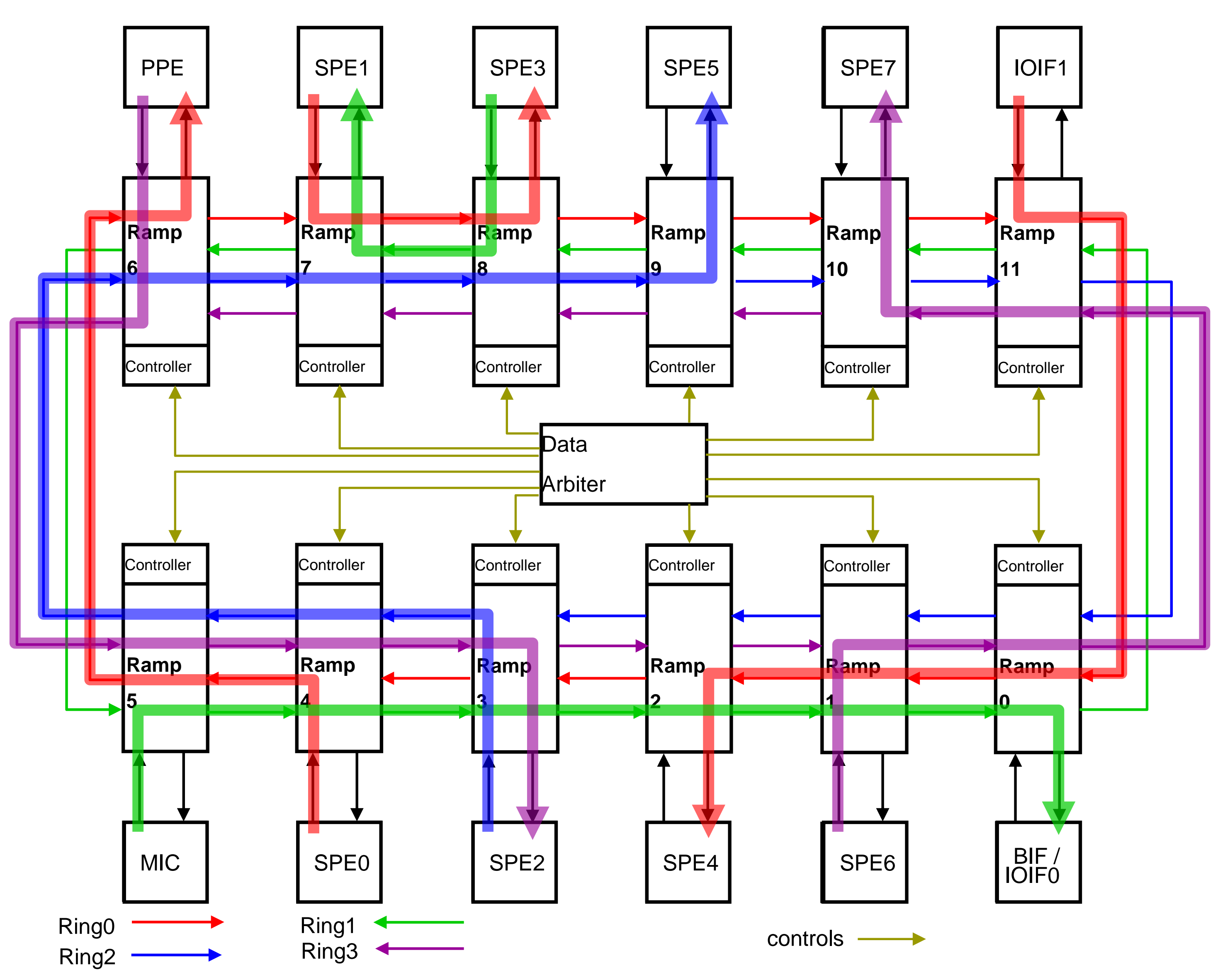
Multiple data transfers on the data rings according to the patent [136] (2005



Remark -1

The ring bus interconnect for data transactions was presumable conceived during the joint development of the Cell processor by engineers of IBM and Sony, as the names and affiliations of the inventors of the related patent application indicate and the similarity of the previous drawing, taken from the cited patent [136], filed in 3/2005, and the drawing taken from a Cell processor publication from 2008 (see next slide), reveals.

Concurrent transactions on the EIB bus of the Cell processor (2006) [137]



8.3.4 Ring bus-based on-chip interconnect (10)

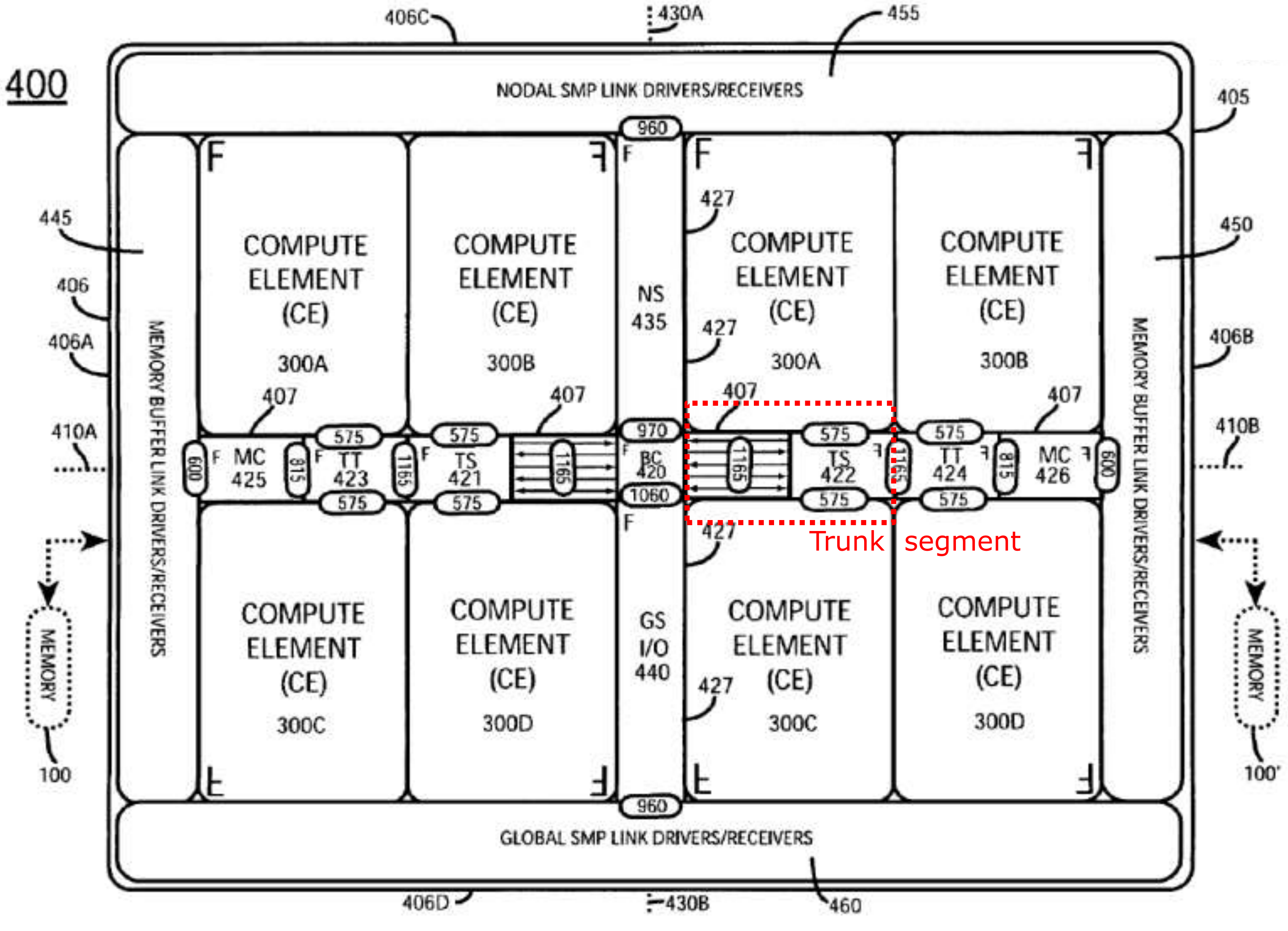
Remark -2

In a subsequent patent the data interconnect became expanded to a cache coherent interconnect by IBM's employees [138], filed in 4/2008.

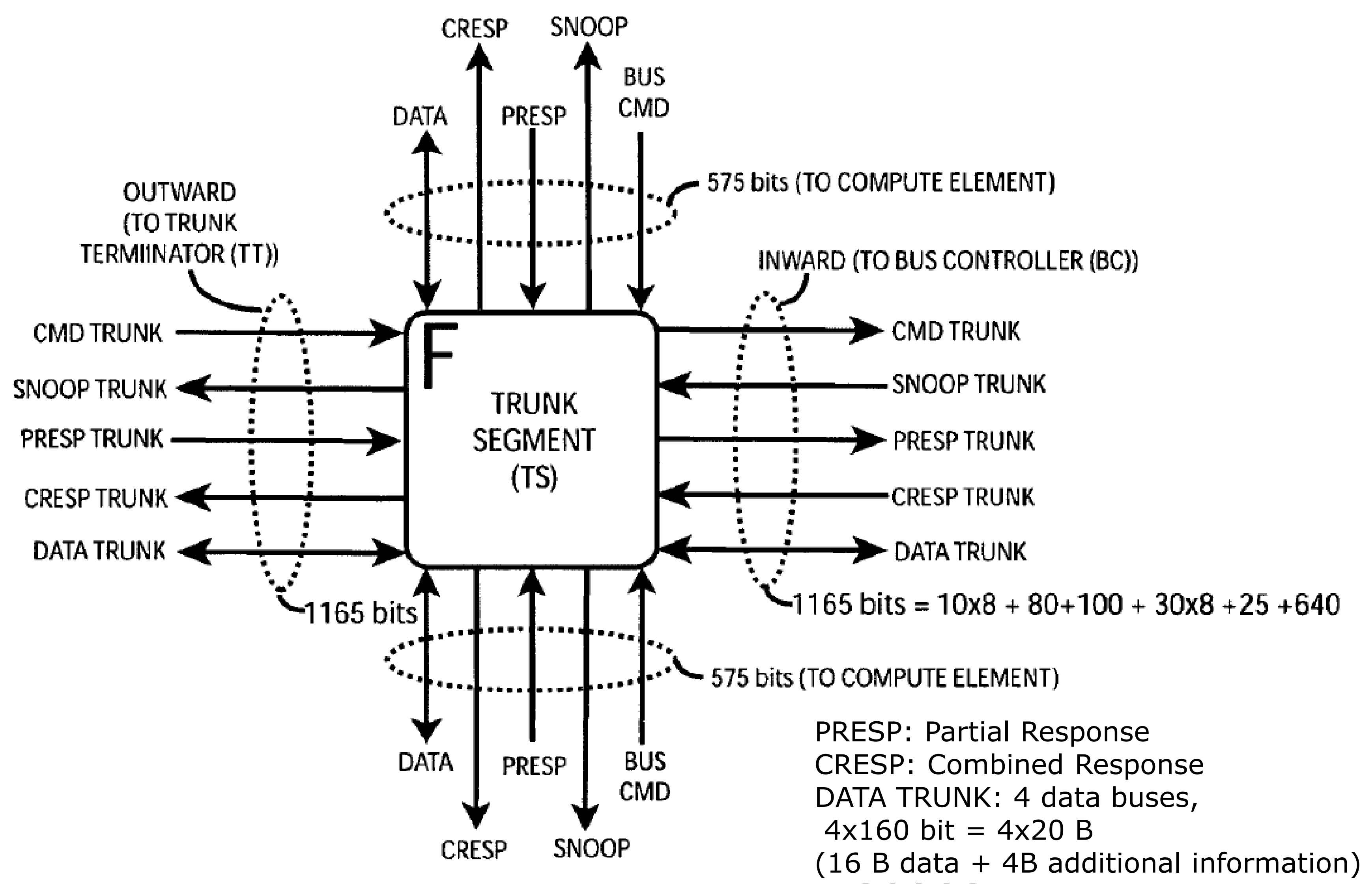
This patent provides also coherent links to interconnect 4 processors into a node (intra-node connections, X-buses) as well as to interconnect multiple nodes to an SMP installation (inter-node connections, A-buses).

As the POWER6 was delivered already in 7/2007, we can assume that the coherent ring bus interconnect could not be employed in the POWER6, only in the subsequent POWER7.

The data ring needs to be enhanced to an on-chip cache coherent interconnect, as seen below for 8 cores according to the patent [138], filed in 2008.

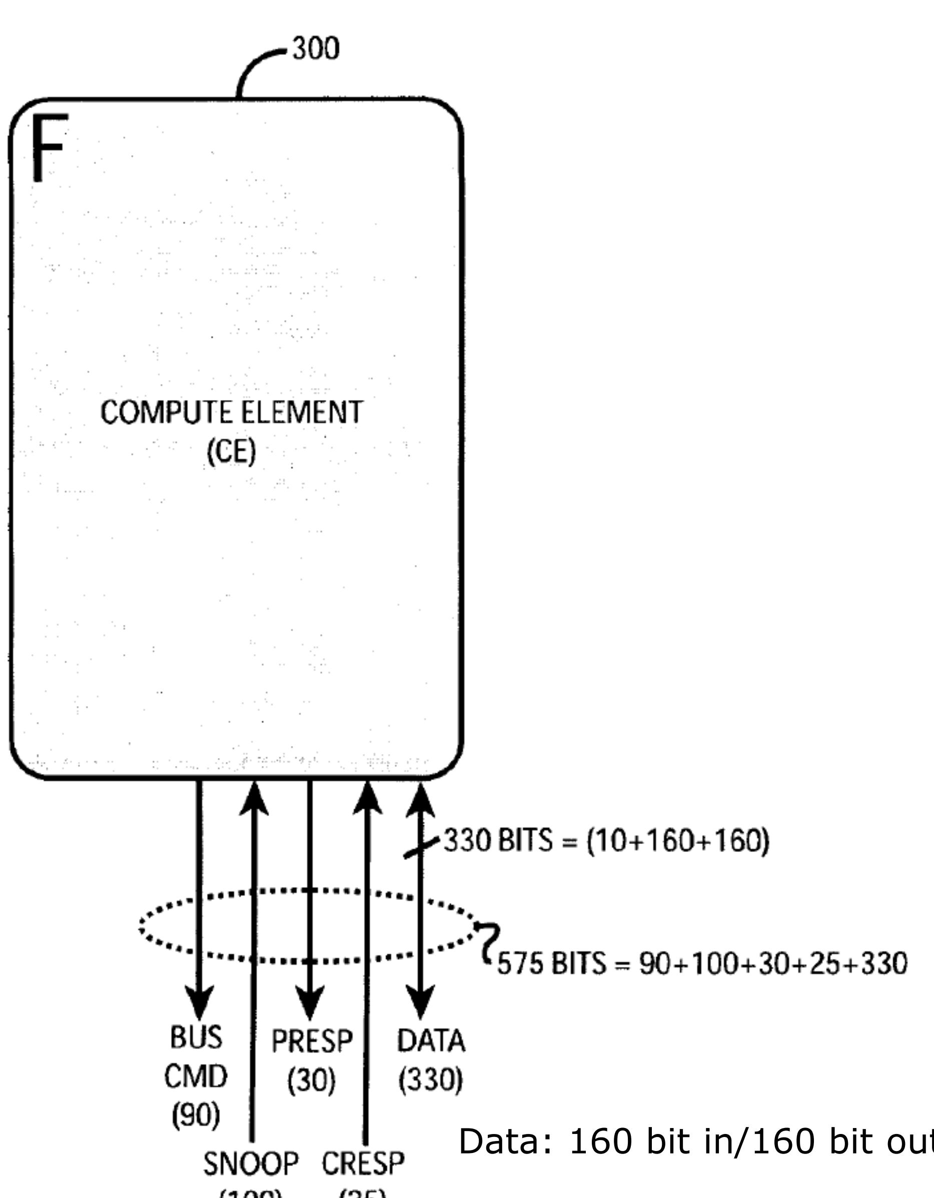


A trunk segment of the on-chip interconnect [138]



PRESP: Partial Response
CRESP: Combined Response
DATA TRUNK: 4 data buses,
4x160 bit = 4x20 B
(16 B data + 4B additional information)

The Compute Element interface
to a trunk segment
according to the patent [138]



PRESP: Partial Response
CRESP: Combined Response

Data: 160 bit in/160 bit out

8.3.5 Re-designed EnergyScale power management

8.3.5 Re-designed EnergyScale power management

The POWER7 introduced a large number of novel features addressing power management, including:

- 8.3.5.1 Novel sensors
- 8.3.5.2 Re-designed EnergyScale infrastructure
- 8.3.5.3 Per core frequency scaling
- 8.3.5.4 Dynamic fan management
- 8.3.5.5 Autonomous frequency control
- 8.3.5.6 Introducing the Sleep idle state

These points will be described next.

8.3.5 Re-designed EnergyScale power management (2)

8.3.5.1 Novel sensors

POWER7 provides:

a) Novel thermal sensors
b) Novel core and memory activity counters
c) Per chiplet power estimation (power proxy)

to be discussed next.

8.3.5 Re-designed EnergyScale power management (3)

a) Novel thermal sensors [65]

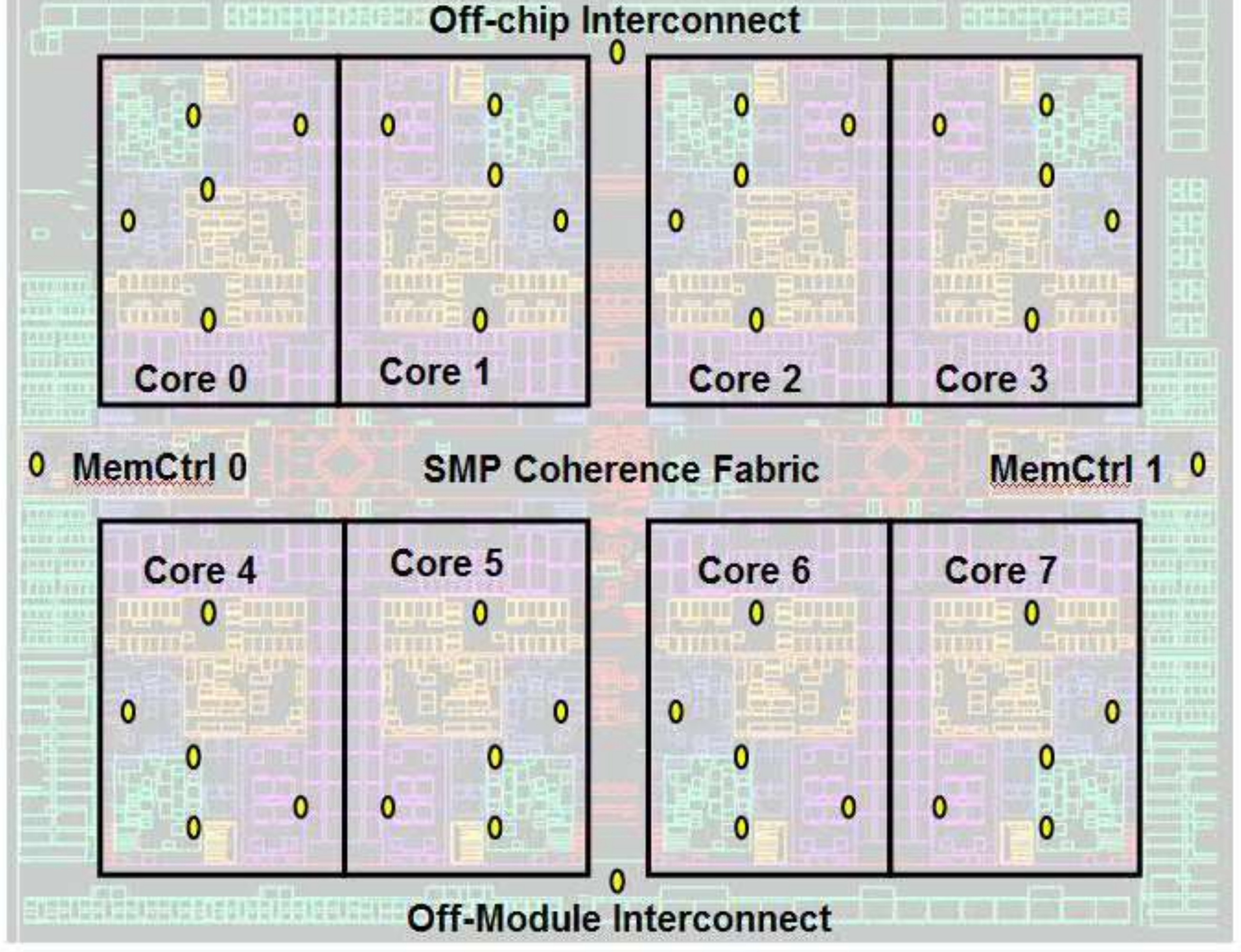
- POWER7 makes use of a new Digital Thermal Sensor (DTS) design that is based on a so called bandgap reference circuit.

  A bandgap reference circuit produces a constant voltage irrespective of variations of the supply voltage or temperature.
- The DTS includes a thermal sensor, like a diode, compares its output with the bandgap reference circuit and converts its value to a digital temperature reading in Celsius degrees by on chip logic.

  The conversion is performed via a polynomial curve fit (px2 + mx + b).

  The coefficients are derived during manufacturing test and calibrated per DTS using a traditional off-chip thermal diode as reference.
- Circuitry in the chip automatically detects when a DTS temperature exceeds a predefined threshold and notifies the microcontroller.
- The DTSs are placed near to areas of high activity (as seen in the next Figure) and temperature readings will be obtained in real-time.

Placement of the DTSs on the POWER7 chip [66]



As seen there are altogether 8x5 + 4 = 44 DTS sensors on the POWER7 chip.

8.3.5 Re-designed EnergyScale power management (5)

b) Novel core and memory activity counters [66]

- The POWER6 introduced three activity counters per-core.
- POWER7 vastly enriches the set of tracked events per-core to cover a larger number of instruction rates, cycle counts and memory event counts, rates, or stalls, not further detailed here but described in detail in related publications.

c) Per chiplet power estimation (power proxy)-1

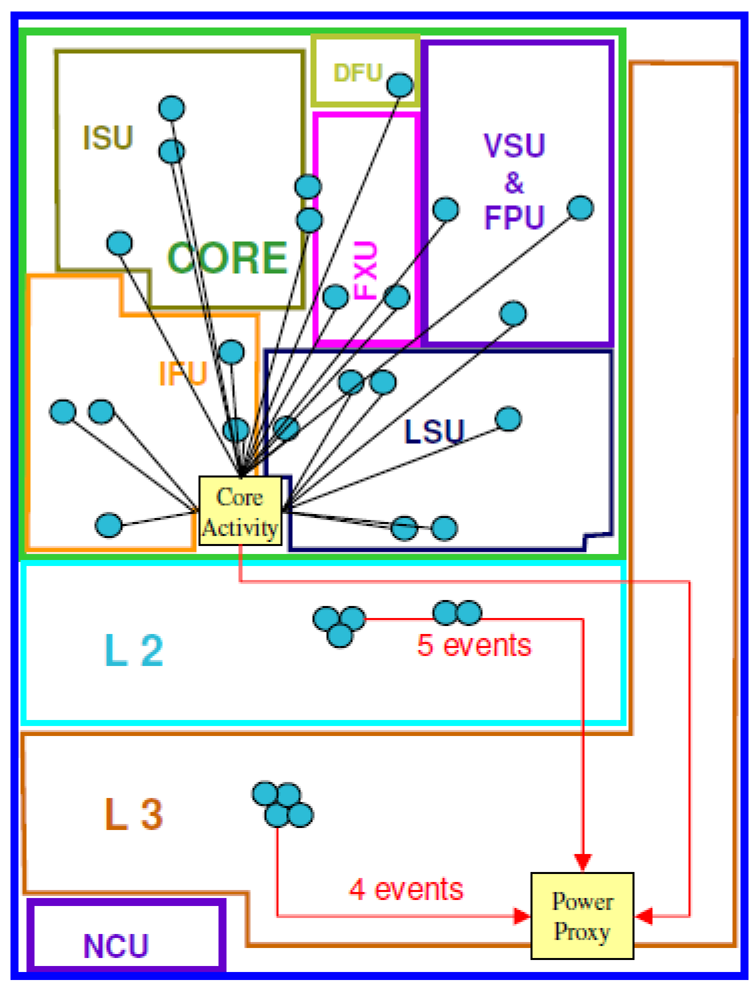
- IBM designates as chiplet the core, the associated L2, L3 segment and the NCU unit (see the next Figure).
- POWER7 estimates per-core power consumption by counter-based activity measurements (called also power proxy).
- To achieve this a minimal set of activities is monitored that correlates maximally with power consumption, such as cache and register file reads and writes, pipeline issues etc.
- Counter values are programmable weighted and combined into an aggregate value for the power consumption of the chiplet, as indicated below.

Active power of the core = ∑ (Wi *Ai) + C +K*fc

with Wi: Weight of the activity Ai
Ai: Activity i
C: Constant
K: Clock grid related power constant
fc: Clock frequency

For each functional unit, pick small subset of activities to infer power consumption (*e.g. cache & regfile reads & writes, execution pipeline issue)*

Locations of the activity sensors on a POWER7 chiplet [66]



The chiplet consists of

- the core,
- the L2 and L3 cache units and
- the NCU

IFU: Instruction Fetch Unit
ISU: Instruction Sequencing Unit
FX: Fixed-Point Unit (FXU)
DFU: Decimal FP Unit
FPU+VSU: Combined FP Unit and Vector-Scalar Unit
NCU: Non-Cacheable Unit .

8.3.5 Re-designed EnergyScale power management (8)

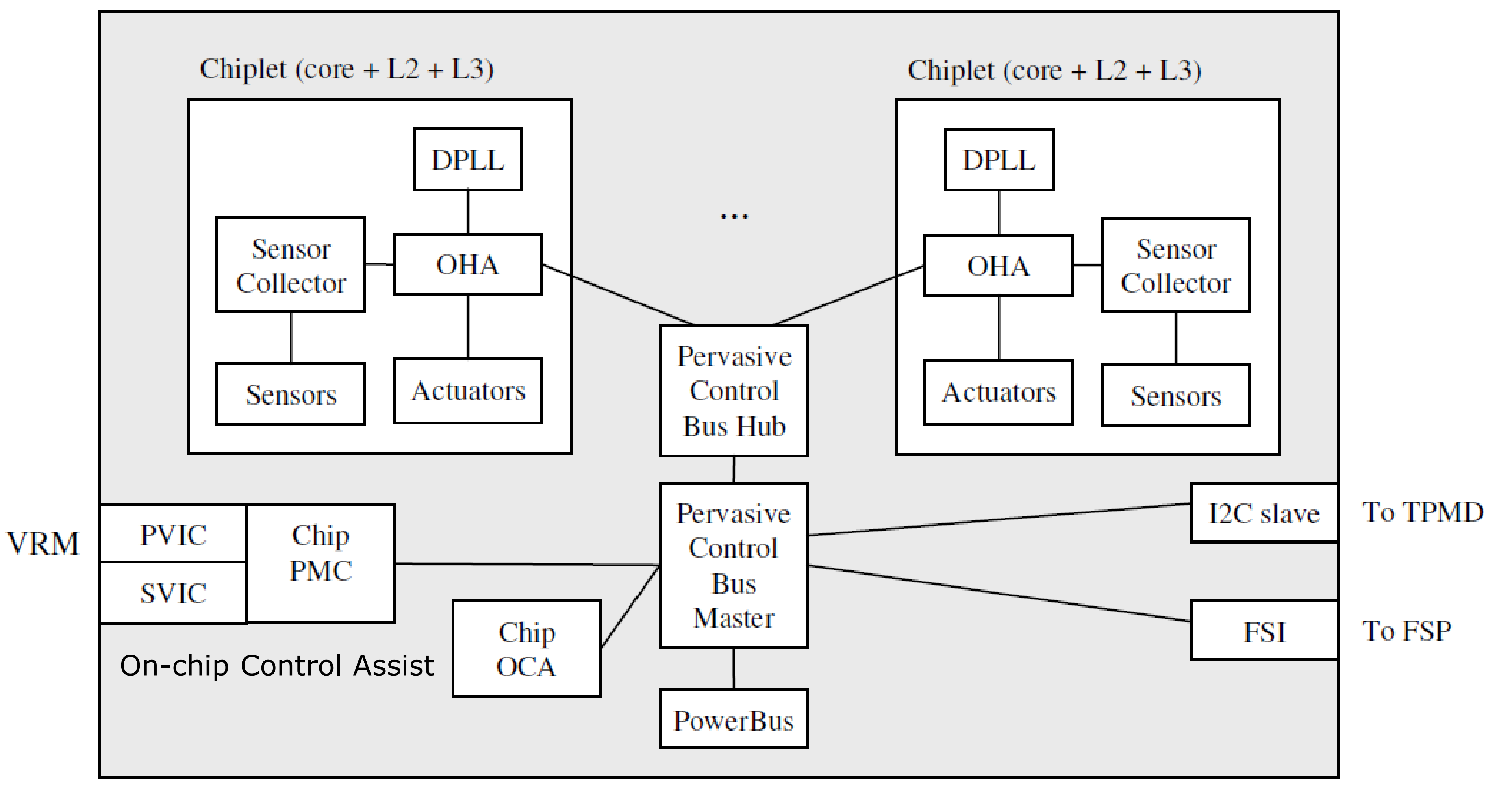
Per core power estimation (power proxy)-2

- The weights (Wi) are calibrated after fabrication (called post silicon) while running a set of benchmarks.
- Genetic algorithms are used to determine the weights for an optimal curve fit.
- The power estimate has an accuracy of a few percent.

8.3.5.2 Re-designed EnergyScale infrastructure [65]

The next Figure shows POWER7's re-designed EnergyScale infrastructure.

Key components of POWER7's re-designed EnergyScale infrastructure [65]



DPLL: Digital PLL
OHA: Chiplet Control Macro
VRM: Voltage Regulator Module

PVIC: Parallel Voltage IF control
SVIC: Serial Voltage IF Control
PMC: Power Management Control

TPMD: Thermal and PM Device
FSI: FRU Support Interface
FSP: Flexible Support Processor

8.3.5 Re-designed EnergyScale power management (11)

Operation of POWER7's re-designed EnergyScale infrastructure [65]

- Note that each chiplet (core + L2 + L3) has its own clock generator termed as DPLL (Digital Phase-Locked Loop).
- The OCA (On-Chip Control Assist) collects power, performance and temperature data from throughout the POWER7 chip and stores it into a central location.
- Up to 1024 bytes of sensor data can then be streamed out to the TPMD in a single high-level I2C access.
- Here we do not want to go into further details, but refer to the literature.

8.3.5 Re-designed EnergyScale power management (12)

8.3.5.3 Per core frequency scaling

- In previous POWER systems all cores run always at the same frequency.
- By contrast, POWER7 allows a per-core frequency scaling to achieve a more power efficient operation.
- Per core frequency scaling was implemented by
  - forming separate per-core clock domains and
  - using Digital Phased-Locked Loops (DPLLs),

  as outlined next.

Forming separate per clock frequency domains [67]

A prerequisite for implementing per-core frequency scaling is forming separate clock domains, as indicated for the POWER7 chip below.

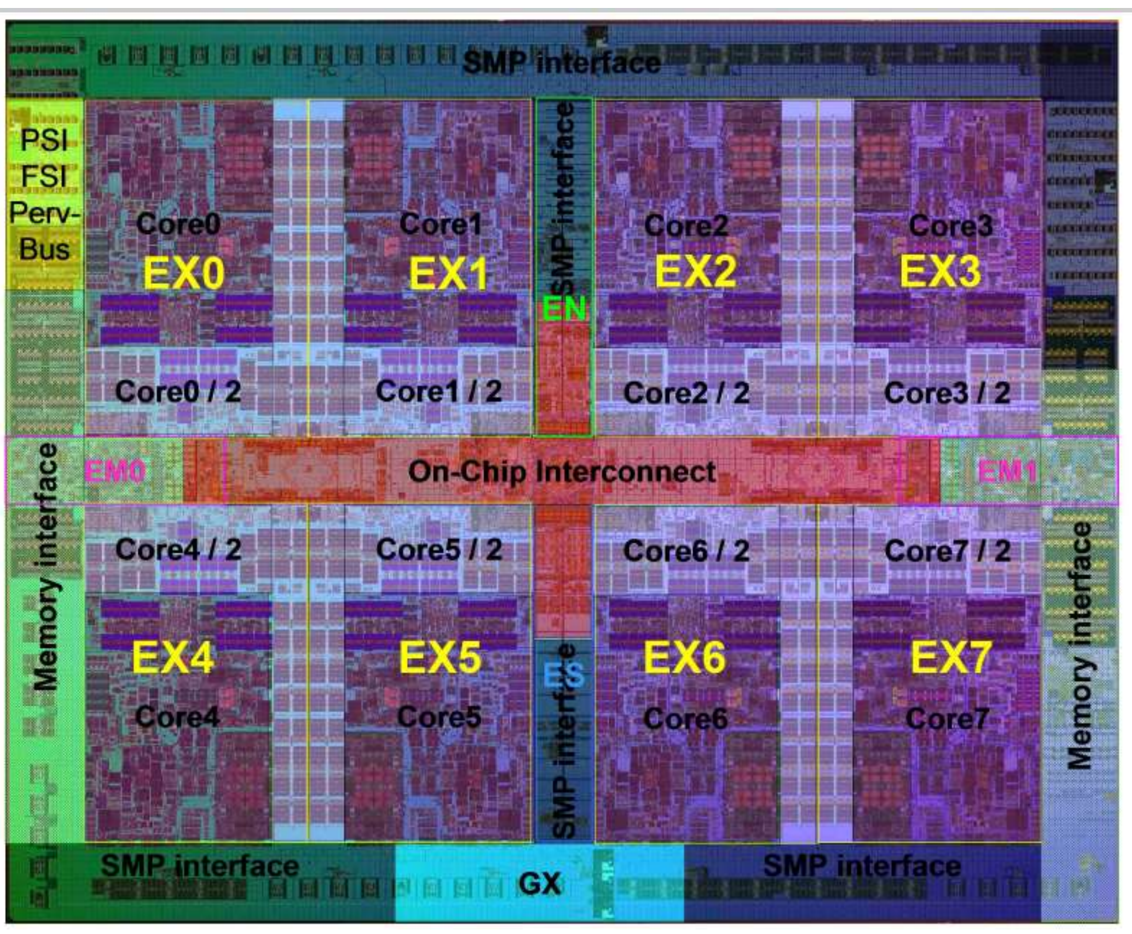


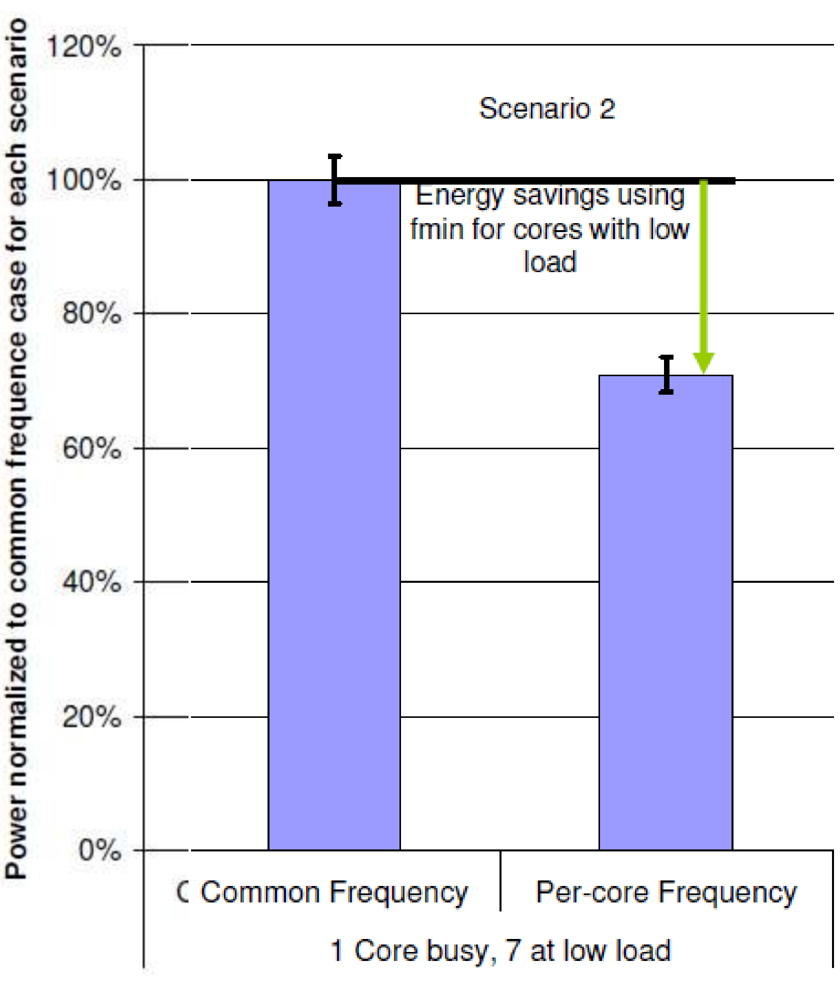
Figure: Separate clock domains on the POWER7 chip [67]

8.3.5 Re-designed EnergyScale power management (14)

Using Digital Phased-Locked Loops (DPLLs)

- The separate frequency domains include Digital Phased-Locked Loops (DPLLs) that are in fact digitally controlled oscillators (DCOs).
- The design allows to change the nominal frequency in the 50 % to 110 % range in excess of 50 MHz per μs.
- The DPLL based frequency generator adjust the core frequency while the core is fully operational in executing code.

Example for the achieved energy saving due to per-core frequency scaling [6



8.3.5.4 Dynamic fan management

POWER7 adds dynamic fan management in two options, as follows:

- SPS (Static Power Save)

  It uses a fixed clock frequency which is 70 % of the nominal one and includes dynamic fan management.

  This reduces power consumption by 24 % vs. no dynamic fan management.

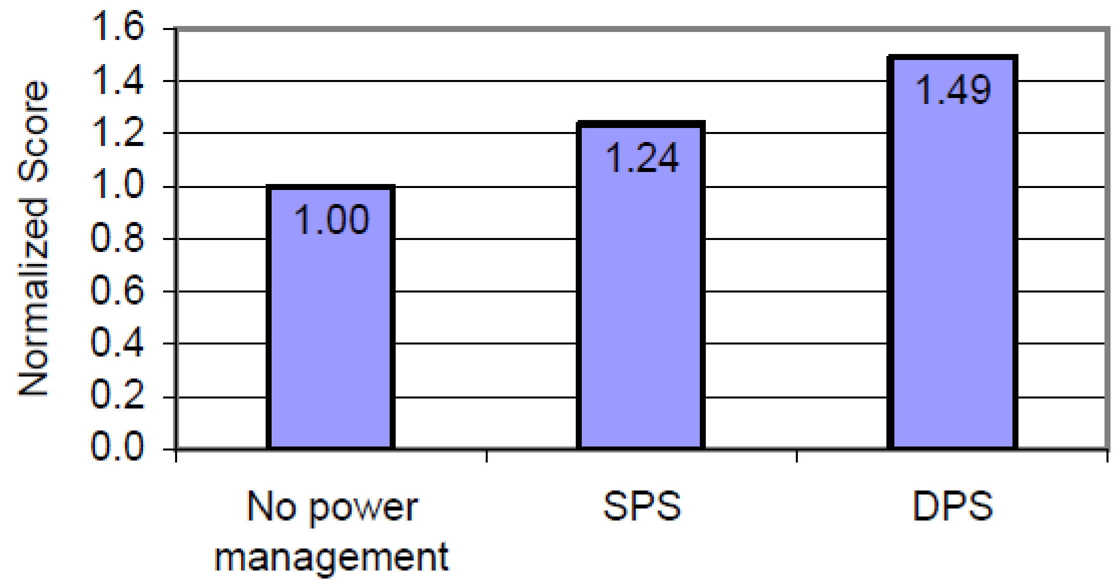
- DPS (Dynamic Power Save)

  It uses both DVFS and dynamic fan management.

  DVFS lets vary the core frequency from 50 % to 109 % of the nominal value.

  DPS results in 49 % power saving over no power management.

Power saving for the SPS or DPS EnergyScale policies [65]



8.3.5 Re-designed EnergyScale power management (18)

8.3.5.5 Autonomous frequency control

- The per-core asynchronous frequency scaling capability allows POWER7 to implement several new fine-grained, low-latency frequency control mechanisms, called autonomous frequency control.
- Autonomous frequency control adjusts chiplet frequencies within a defined range to quickly respond to operating conditions for reducing power or increasing performance.
- There are two such mechanisms:
  - autonomous frequency reduction during low-activity
  - reducing wasteful guard band to reduce power or increase performance,

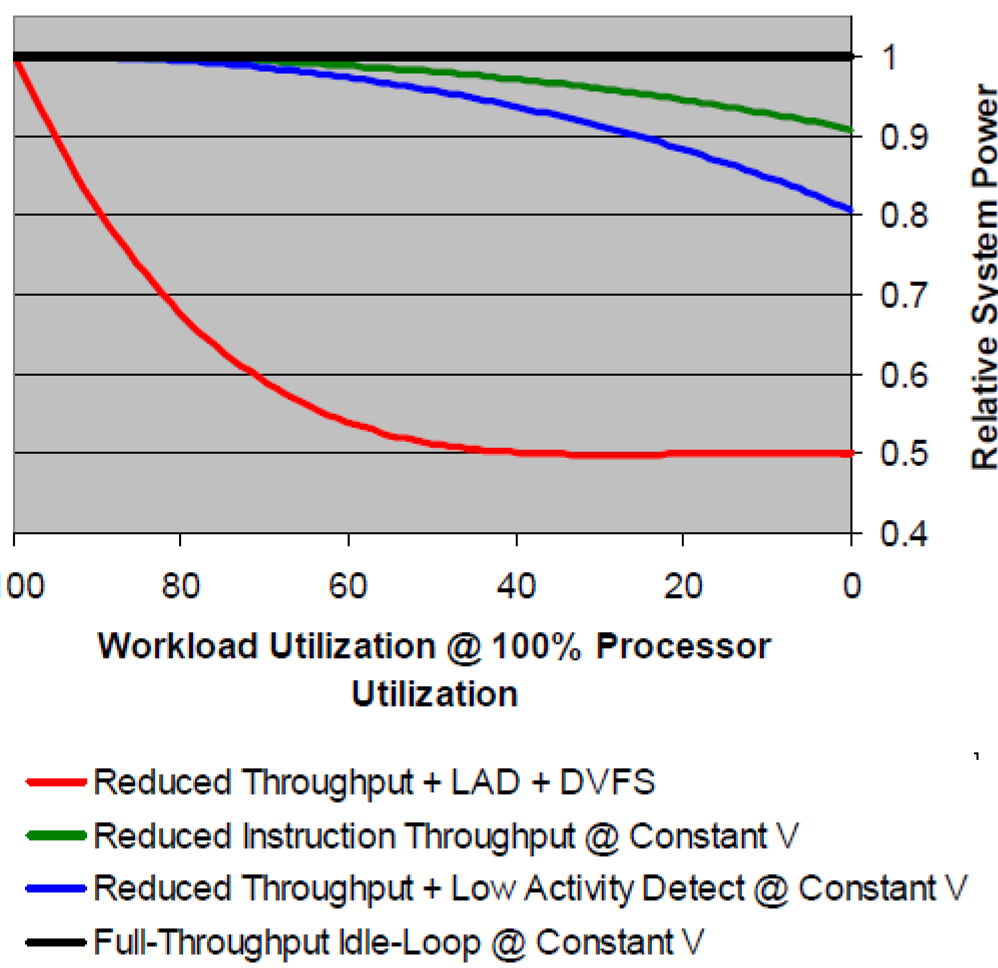
to be discussed next.

8.3.5 Re-designed EnergyScale power management (19)

Autonomous frequency reduction during low-activity [51]

- The Power7 implements a low-activity detection (LAD) mechanism that permits rapid frequency scaling to exploit low activity periods observed in intervals as short as a few microseconds that are too short for firmware or system software to detect.
- LAD is triggered by low per-core utilization, typically derived from IPC (instructions-per-cycle) measurements when the processor is executing a workload.
- In periods of low activity LAD hardware lowers core chiplet frequencies as long as the per-core utilization is below a programmable threshold.
- For example, if the LAD mechanism observes an average IPC of less than 0.25 over an interval of 32 μs the EnergyScale firmware may program this mechanism to drop core frequency to 50 % of nominal.

Benefit of the LAD mechanism [51]



8.3.5 Re-designed EnergyScale power management (21)

Reducing wasteful guard band to reduce power or increase performance [51]

- POWER6 introduced already critical path monitors (CPMs) to reduce guard band to lower power consumption.
- POWER7 makes use also of CPMs while utilizing the novel per-core frequency scaling feature of this processor in an enhanced way, as follows.
- The combined CPM outputs are used as a feedback for the frequency generator (DPLL) such that the DPLL will be slowed down if the timing margin is too small and speeded up if the timing margin is larger than necessary to achieve a comfortable guard band.
- Reducing the guard band can be utilized either to boost performance by overclocking the core or to reduce power consumption by undervolting the core, as indicated in the next Figure.

Using guard band reduction for increasing performance (overlocking) [51]

Example for the achieved performance increase.

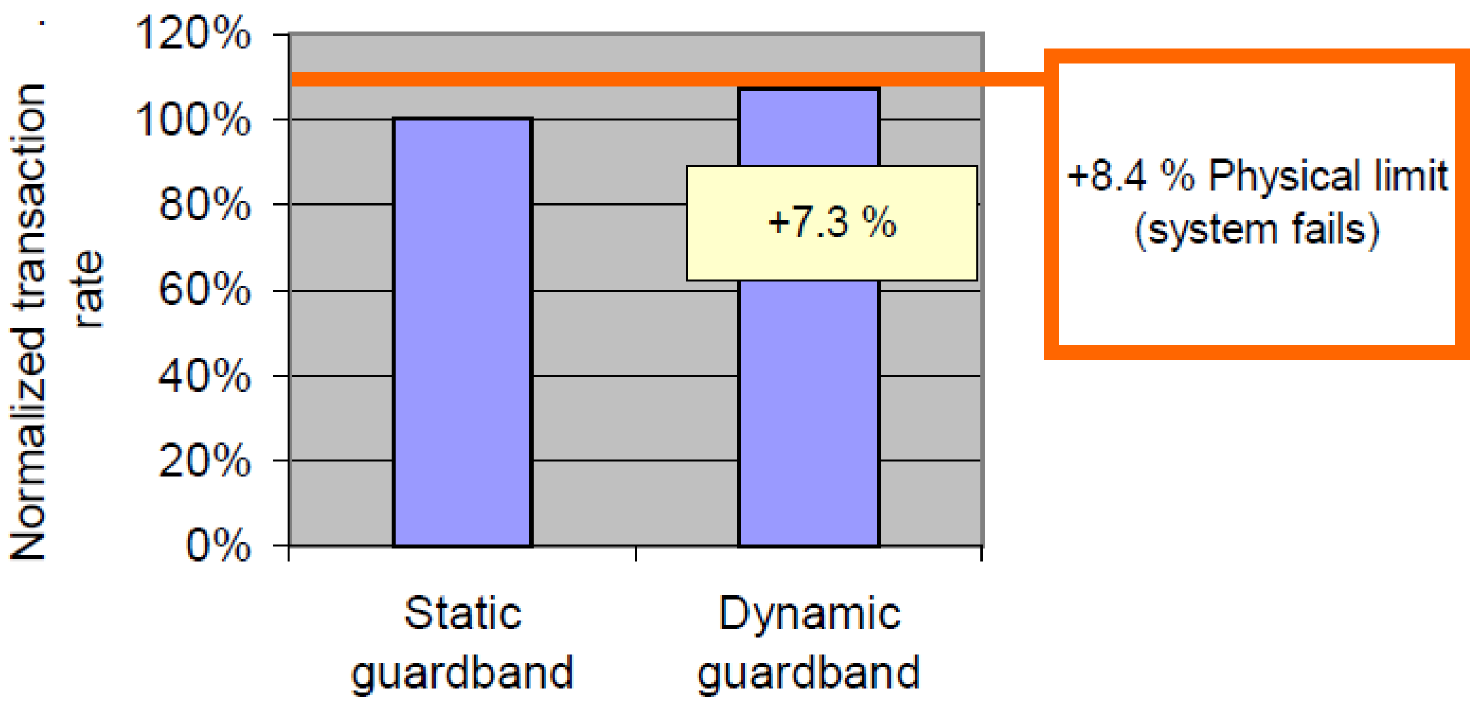


Figure: Performance boost for reducing the guard band while running the SPECpower ssj2008 benchmark on a POWER7 chip [51].

Using guard band reduction for lowering power consumption (undervolting)

Example for the achieved power reduction.

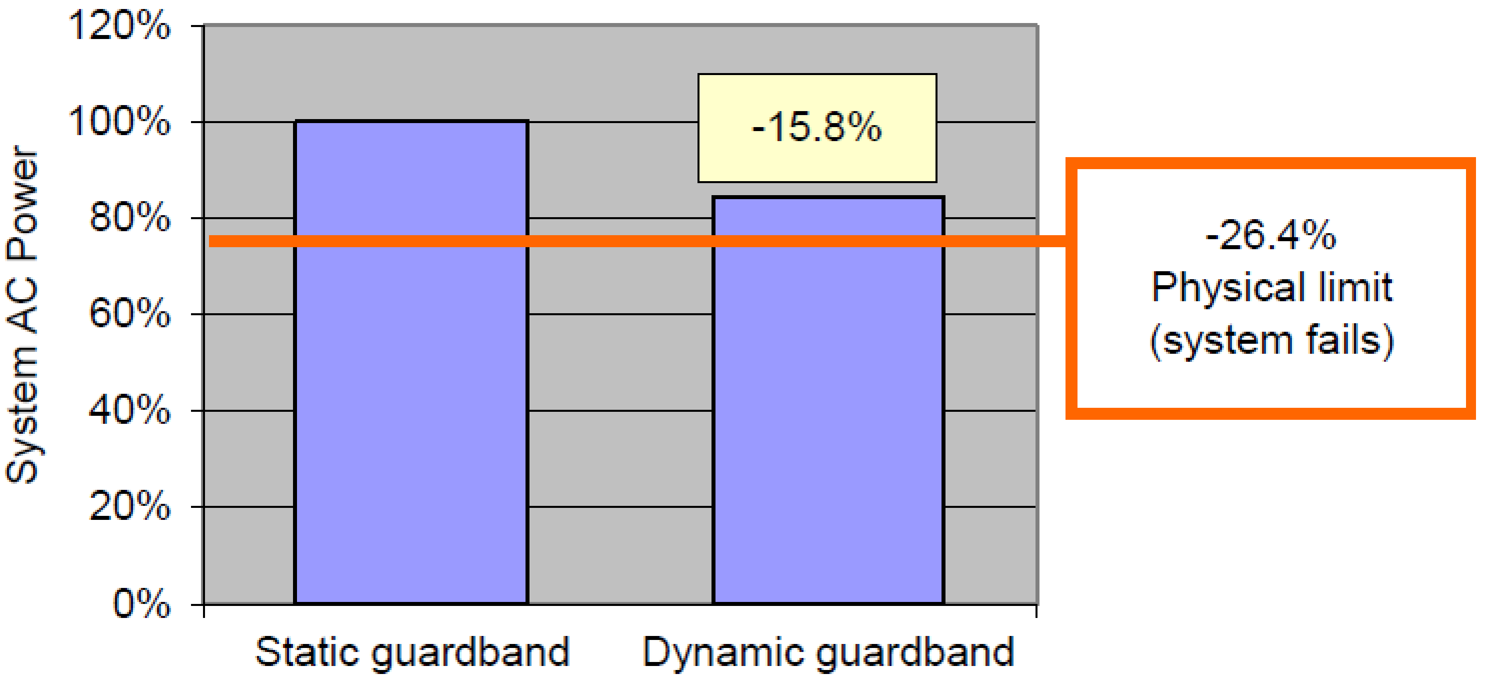


Figure: Power reduction for reducing the guard band while running the SPECpower ssj2008 benchmark on a POWER7 chip [51].

8.3.5.6 Introducing the Sleep idle state

- POWER6 introduced the Nap idle mode and Nap idle state previously to reduce power consumption.
- POWER7 enhanced idle states management by introducing a second, more aggressive idle state, called the Sleep state.
- accordingly, POWER7 differentiates between two idle states:
  - the Nap state and
  - the Sleep state.
- In the POWER architecture it is the task of the hypervisor to control the entry to and the exit from processor idle states through state-specific privileged instructions.
- Operating systems guide the hypervisor on which idle state to use based on expected idleness for the processor threads (and ultimately the core) through specific hypervisor calls.

8.3.5 Re-designed EnergyScale power management (25)

The Nap state introduced before in the POWER6 [65]

- The Nap state is entered whenever the hypervisor has executed a power-save instruction on all threads, and at least one thread executes a Nap instruction.
- In the Nap state, all of the execution units in a core and the L1 caches are clocked off; while the higher level caches and certain timing facilities remain functional, allowing low-latency workload resumption in the event of timer or external interrupts.
- In addition, on map entry hardware logic selectively supports scaling down frequency to a preprogrammed lower value (e.g. to 70 % of the operational value) and on nap exit it ramps up to the operational frequency value.

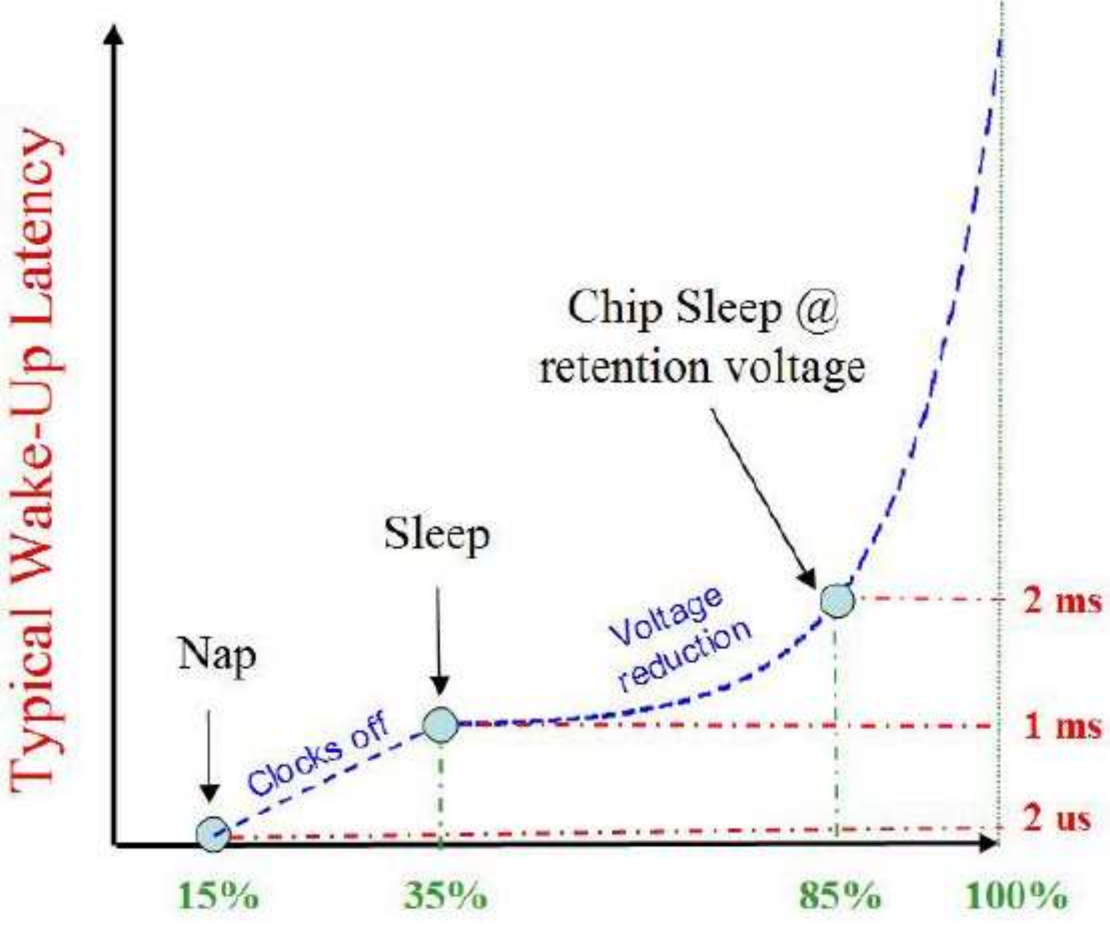
  This feature provides a significant reduction in power for napping core chiplets, as indicated in the related Figure, at the potential expense of increased access latency for shared data requested by a non-napping core from a napping cache (L2/L3).
- Upon wakeup instruction execution begins immediately regardless of whether the frequency was dropped while in nap.

The Sleep state of the POWER7 [65]

- The Sleep state is a new architectural feature introduced in the POWER7.
- It is a lower-power, higher-latency standby state than the Nap state intended for cores that the hypervisor/OS predicts unused for an extended period of time.
- The Sleep state is entered when every thread on a core executes a Sleep instruction.
- Upon entering Sleep, hardware state machines purge all data from the caches and invalidating all translation lookaside buffers before completely clocking off the entire chiplet.
- Nevertheless, a small logic associated with the core chiplet remains awake to handle external interrupts that wake-up the core out of sleep.
- When all the cores on a chip are in sleep state, the hardware logic can be configured to automatically lower the external voltage to the retention voltage, i.e. to a low-voltage at which latches and arrays continue to retain data but leakage current becomes reduced.

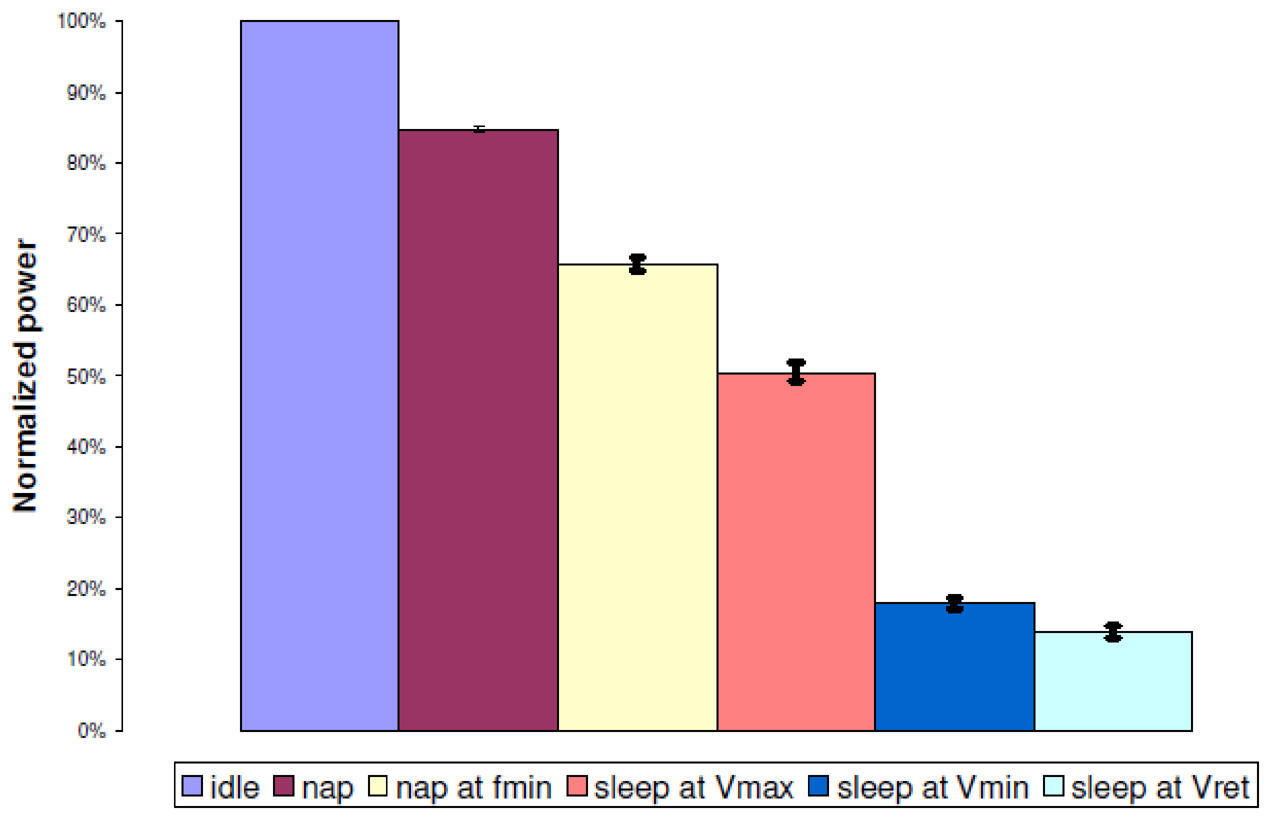
  This mode provides the lowest standby power for a POWER7 processor, as seen in the related Figure.
- Upon waking up from sleep, the hardware logic restarts the DPLL, powers up the L3 cache's eDRAM, and sequences the core chiplet back to operation.

Typical wake-up latencies and associated power reductions of idle states [51]



Power reduction values are given relative to idling the processor at the nominal operating point while the OS is polling for work.

Average power in idle modes [65]



For the above measurements the min. frequency (fmin) used was about 46 % of the max. frequency (fmax).

8.3.5 Re-designed EnergyScale power management (29)

Remark

- The idle states Nap and Sleep are not new.
- Originally they were introduced in the first low-power model of the 32-bit PowerPC family, called the PowerPC 603 in 1994.

9. POWER7+

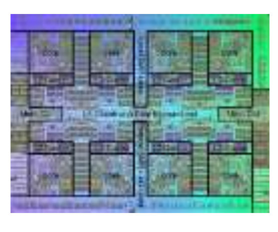
- 9.1 Introduction to the POWER7+
- 9.2 Main enhancements of the POWER7+
- 8.3 Key innovations of the POWER7

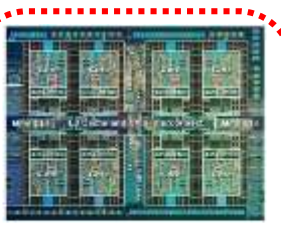
9.1 Introduction to the POWER7+

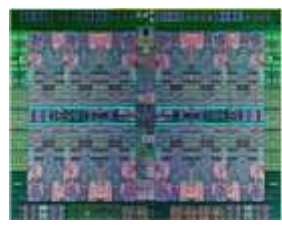
9.1 Introduction to the POWER7+

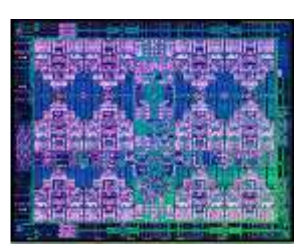
- Introduced: 10/2012
- 32 nm technology
- 567 mm2, 2.1 billion transistors
- 6-wide out-of-order-superscalar with an issue rate of 8 for 12 execution units
- Uses the same socket as the POWER7

Key features of the POWER7+ -1





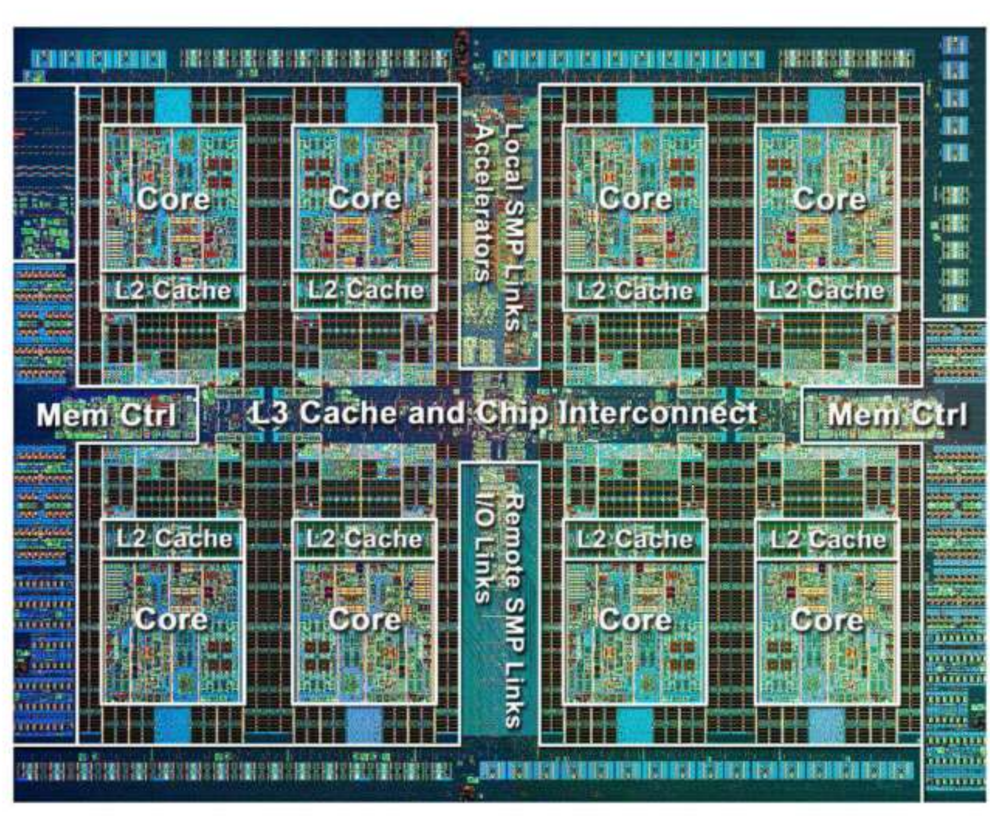




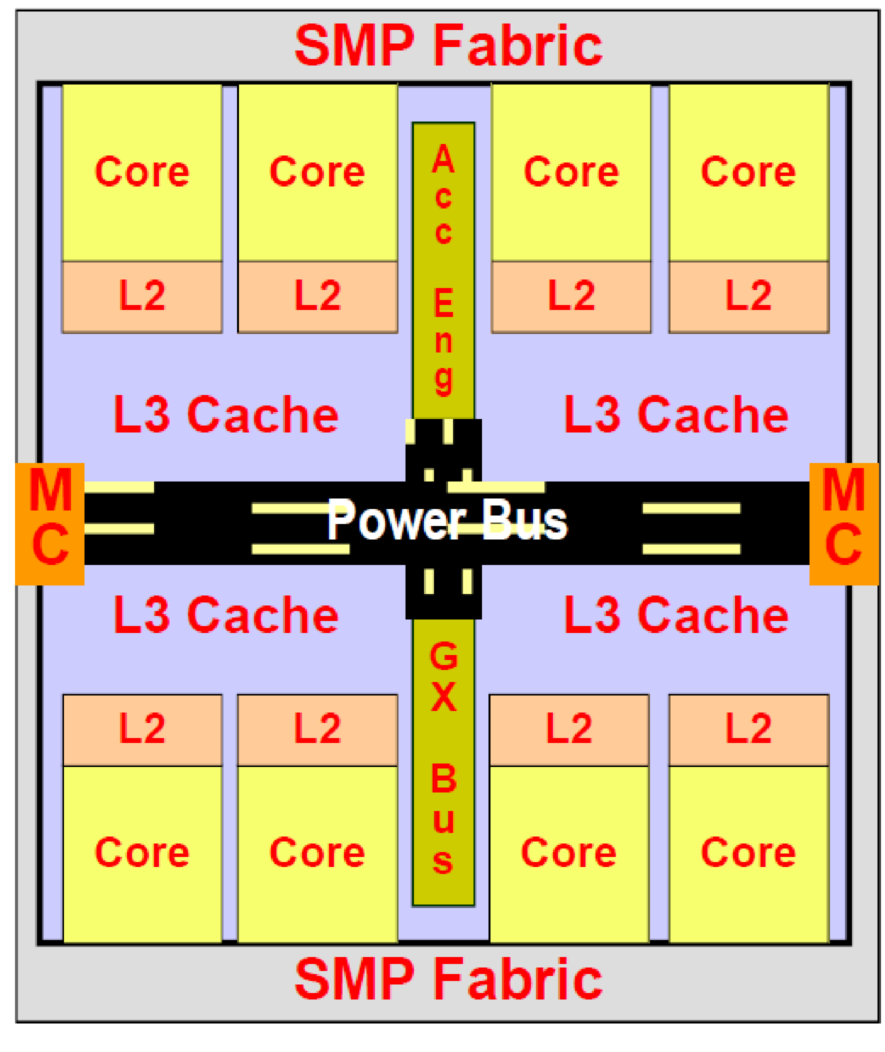
	POWER7	POWER7+	POWER8	POWER8+	POWER9
Launched	2/2010	10/2012	4/2014	Planned/cancelled	12/2017
Technology	45 nm	32 nm	22 nm		14 nm
Die size	567 mm2	567 mm2	650 mm2		693 mm2
Transistors	1.2 b	2.1 b	4.2 b		8.0 b
Cores (up to)	8	8	12		12 SMT8 cores 24 SMT4 cores
SMT	4-way	4-way	8-way		4-way/8-way
Typ. fc	3.72-4-42 GHz	3.1 -4.42 GHz	3.02-4.35 GHz		Up to 4 GHz
L2	256 KB/core	256 KB/core	512 KB/core		512KB/2 cores
L3	4 MB/core	10 MB/core	12 MB/core		10 MB/2 cores
Mem. contr.	2/1	2/1	8		8
Memory up to	DDR3-1066	DDR3-1066	DDR3-1600		DDR4-2666

Key features of the POWER7+ -2 [139]

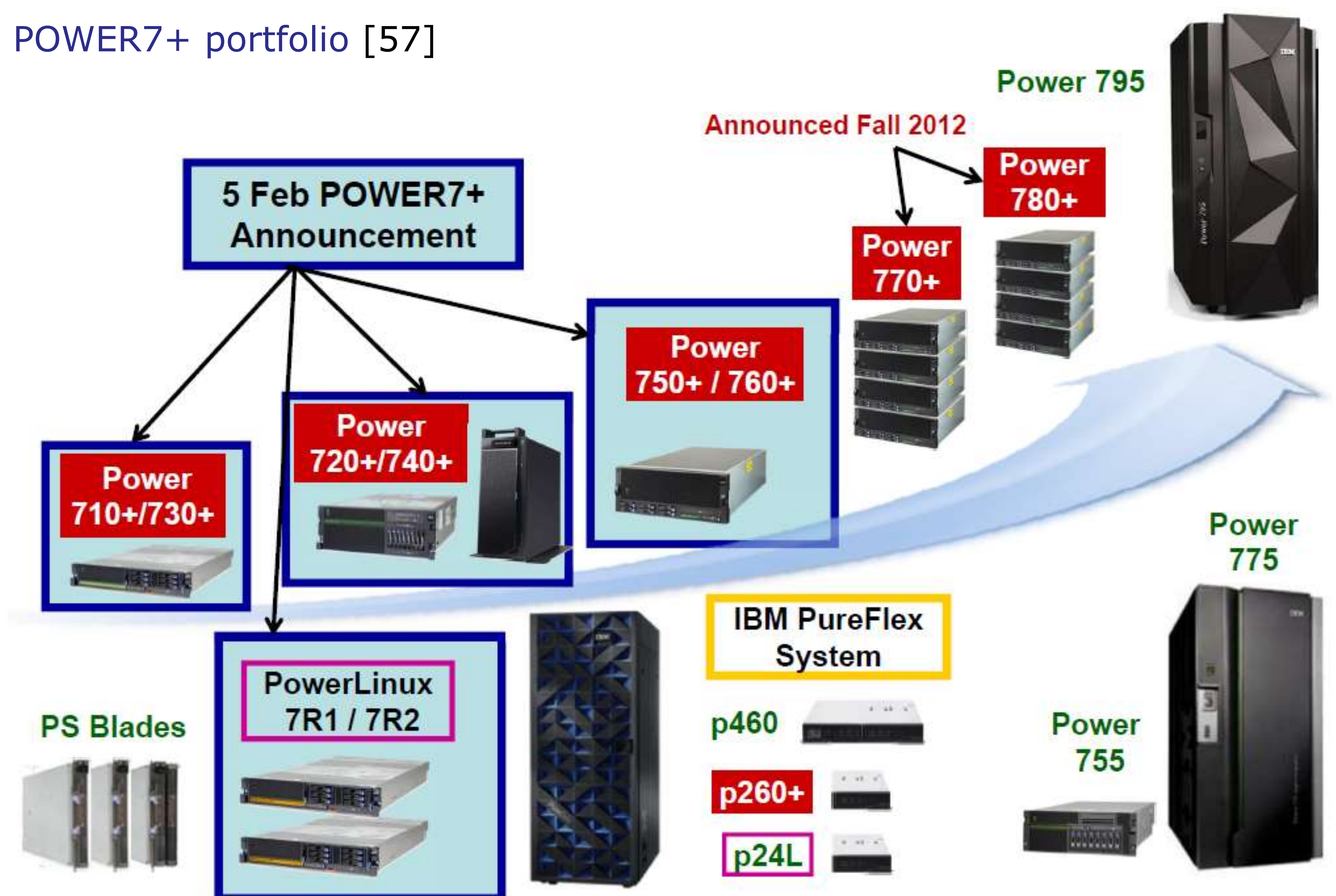
- Area: 567mm2
- Eight processor cores
  - 12 execution units per core
  - 4 Way SMT per core
  - 32 Threads per chip
  - 256KB L2 per core
- Scalability up to 32 Sockets
  - 360GB/s SMP bandwidth/chip
  - 20,000 coherent operations in flight
- Technology: 32nm lithography, Cu, SOI, eDRAM, 13 metal levels
- 2.1B transistors
  - Equivalent function of 5.4B
- 80MB on chip eDRAM shared L3
- Accelerators
- Enhanced Power management
- Binary Compatibility with POWER6/7



High level block diagram of the POWER7+ [68]



POWER7+ portfolio [57]



9.1 Introduction to the POWER7+ (6)

IBM Power systems family (Stand April 2014) [69]

PowerVM PowerVC
PowerKVM PowerVP
PowerHA PowerSC

IBM Systems Software

POWER8

POWER7+

POWER7

Power 710
Power 730

Power S822

Power 720
Power 740

Power S814

Power 750

Power S824

Power 760

Power 770

Power 780

Power 795

PowerLinux
7R1 / 7R2 / 7R4

Power S812L

Power S822L

IBM PureSystem

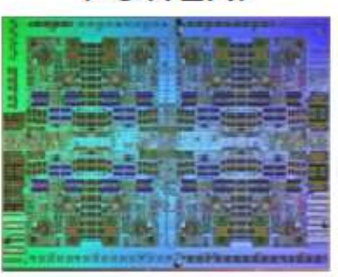
IBM Flex System p460

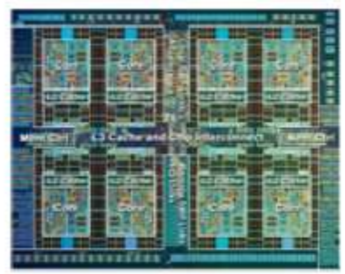
IBM Flex System p270

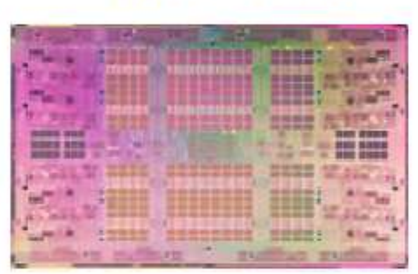
IBM Flex System p260

9.1 Introduction to the POWER7+ (7)

Contrasting key features of POWER7/POWER7+ and Intel's Poulson [57]







	POWER7	POWER7+	Intel Poulson
Cores	8	8	8
Threads per Core	4	4	2
Frequency	4.0 Ghz	4.5 GHz	2.53 GHz
Chip Size	567mm2	567mm2	544 mm2
Technology	45nm SOI 11 LM EDRAM	32nm SOI 13 LM Edram	32nm 9 LM
Max Socket support	32	32	32
Power	250 Watts	250 Watts	170 Watts
Spec_int Rate/Chip	340	390	180
Memory BW (70% utilization)	96GB/s (16 DDR3 channels)	96GB/s (16 DDR3 channels)	45 GB/s (4 DDR3 channels)
L3	32MB	80MB	32MB
Extras	Advanced Prefetch HPC Features Energy management Turbo Mode/Core	Need to add	QPI busses to IO interfaces

9.1 Introduction to the POWER7+ (8)

POWER7/POWER7+ module packaging [57]

Power 795
Single Chip Glass Ceramic

Power 775
Quad-chip MCM

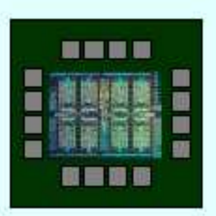
POWER7

Power 770 / 780
Single Chip Glass Ceramic

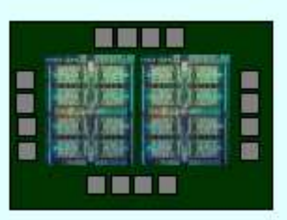
POWER7

Power 710 / 730
Single Chip Organic

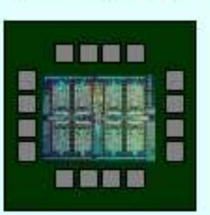
Power 720 / 740
Single Chip Organic



Power 750 / 760
Dual Chip Organic



Power 770 / 780
Single Chip Organic



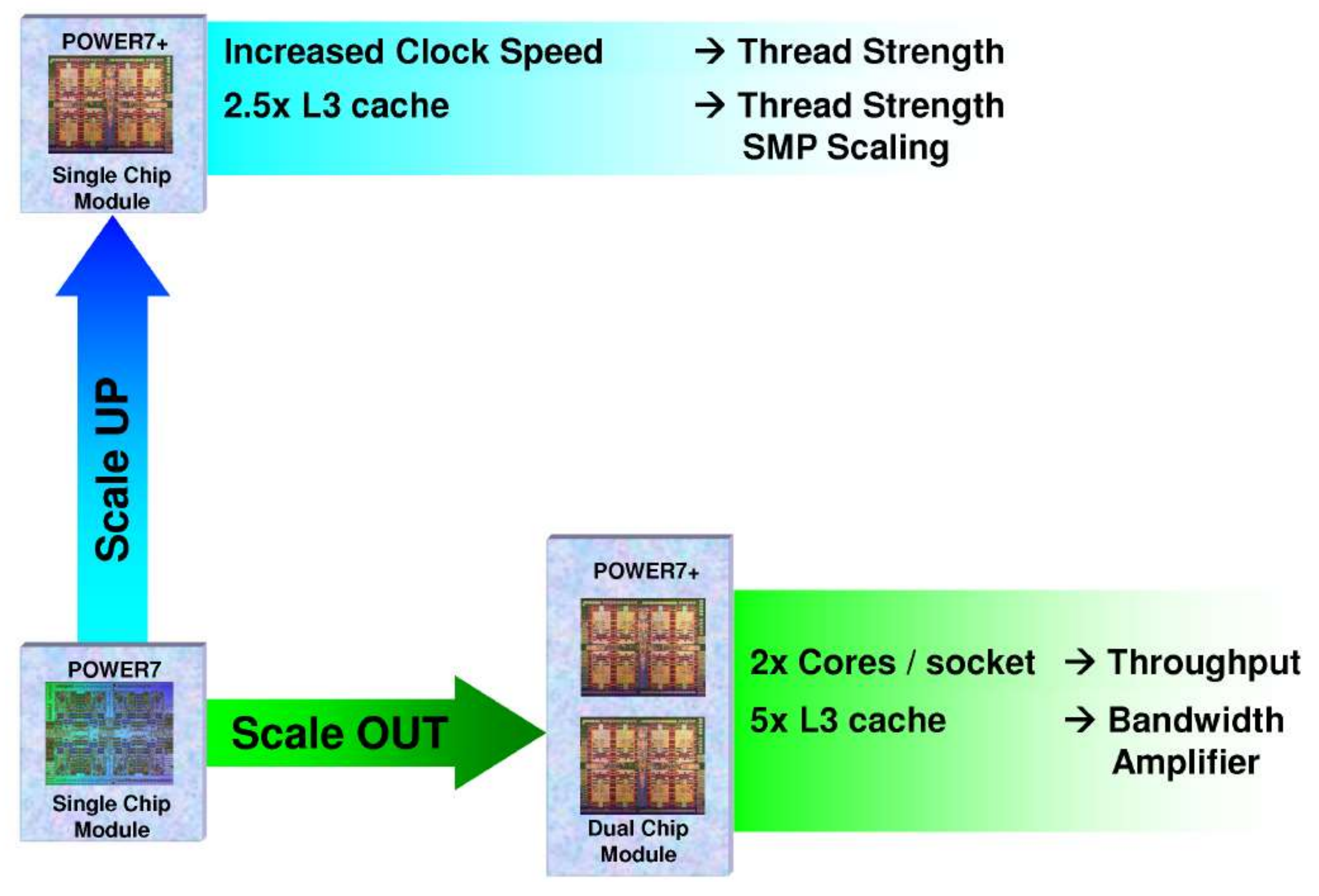
POWER7+

9.2 Main enhancements of the POWER7+

- 9.2.1 Increased clock frequency
- 9.2.2 Larger L3 caches per-core
- 9.2.3 Doubling the SP FP performance
- 9.2.4 Added power gating regions for cores and L2/L3 caches
- 9.2.5 Remarks to the memory subsystem of the POWER7+

9.2 Main enhancements of the POWER7+ [139]

Optimization in two directions

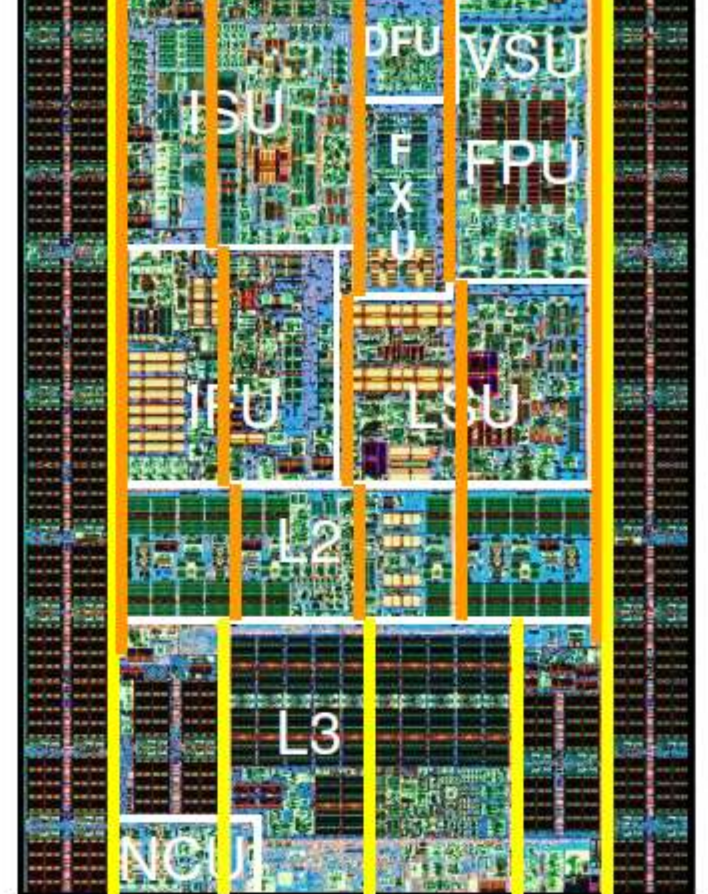


Main enhancements of the POWER7+ [70]

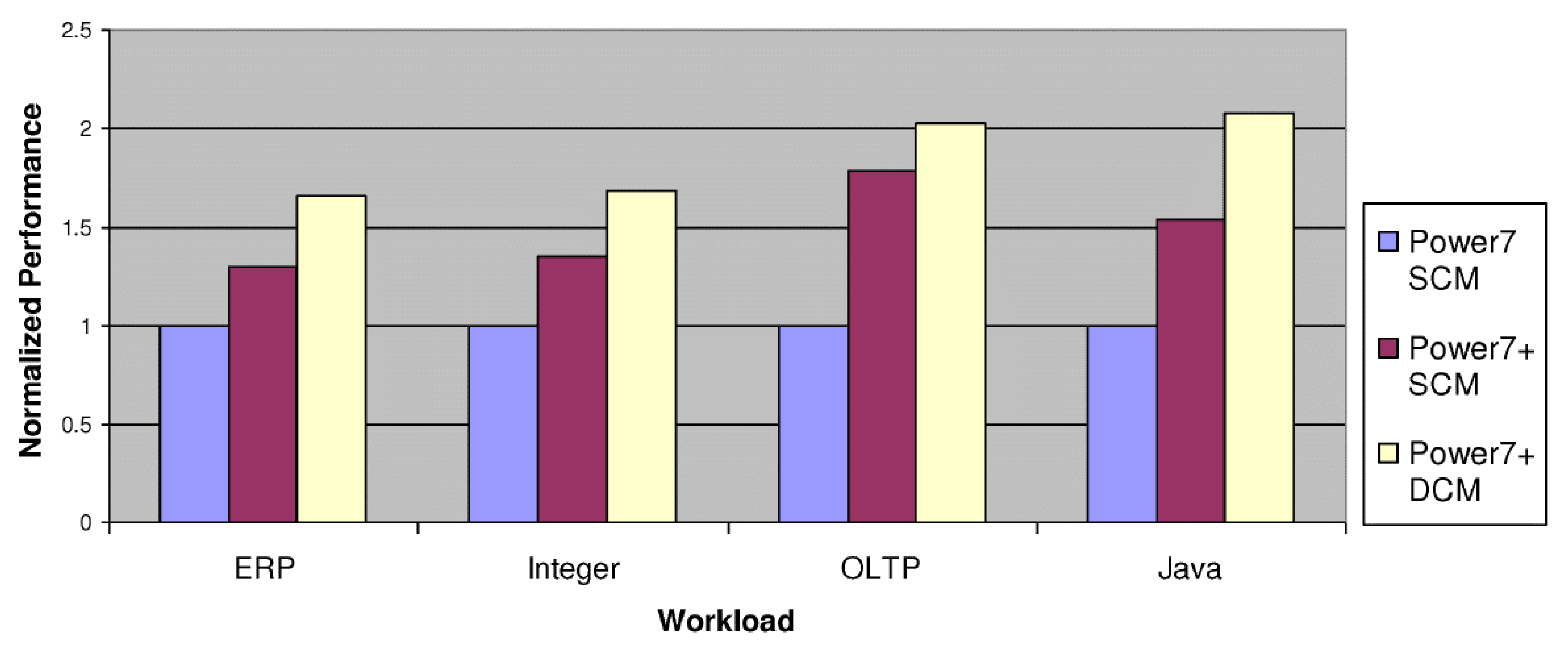
- Up to 25% frequency gain due to mapping into 32nm technology and power management improvements.
- Increase of L3 memory capacity by 2.5x
- Doubled single precision floating-point performance
- Added Power Gating regions for Core/L2 & L3 regions

Core/L2 Power-Gating

L3 Power-Gating



Results of the optimizations of the POWER7+ [139]



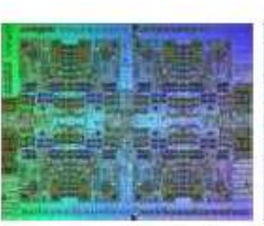
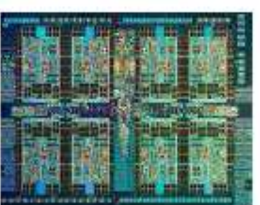
9.2.1 Increased clock frequency [123]

IBM reduced the cycle time through tuning the circuit design, e.g. by

- lowering the threshold of the devices on cycle time critical paths and
- enhanced buffering on long wires.

The next Table shows the clock frequencies of subsequent models for comparison.

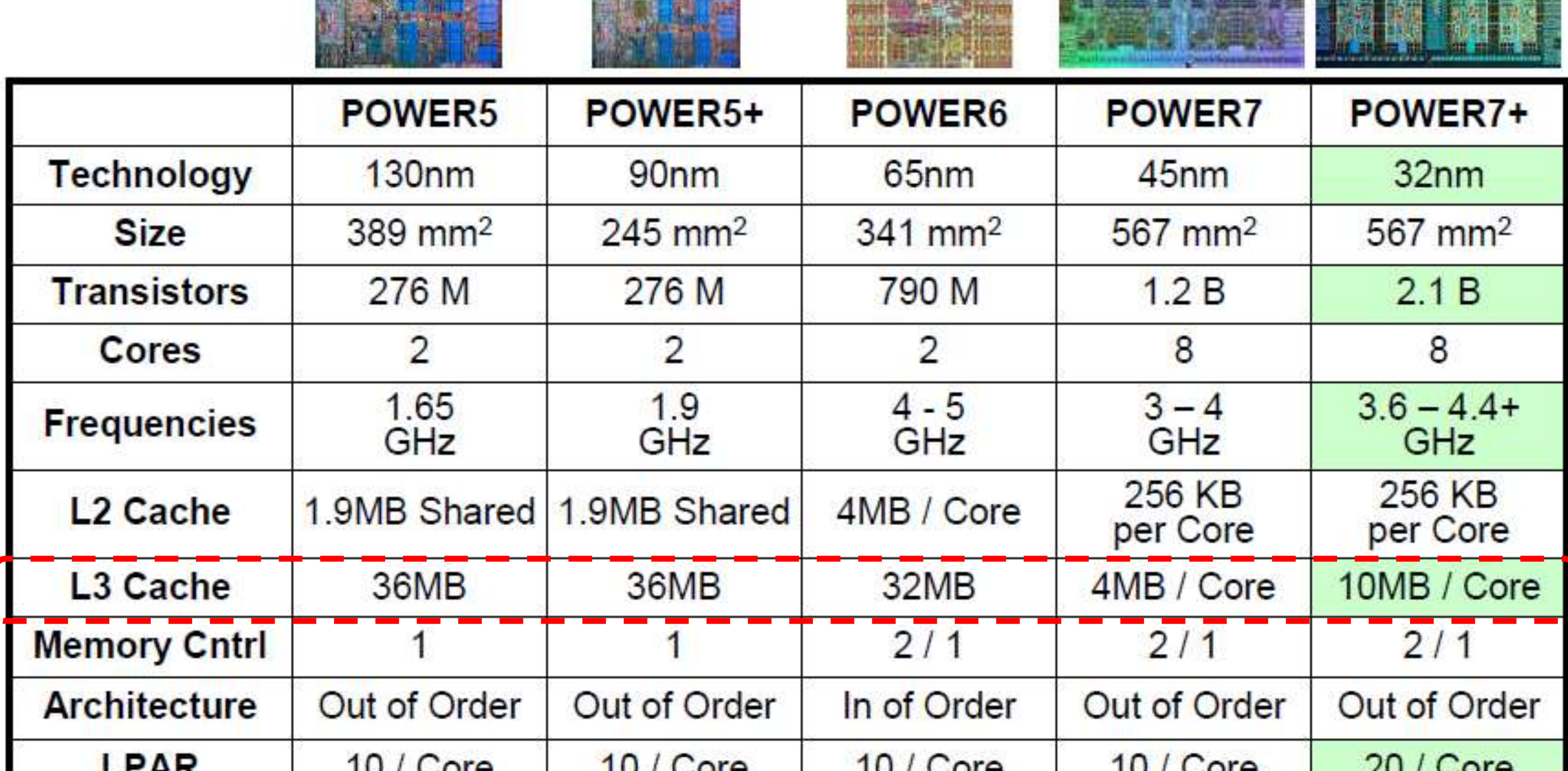
Clock frequencies of subsequent POWER models [57]

	POWER5	POWER5+	POWER6	POWER7	POWER7+
Technology	130nm	90nm	65nm	45nm	32nm
Size	389 mm^2	245 mm^2	341 mm^2	567 mm^2	567 mm^2
Transistors	276 M	276 M	790 M	1.2 B	2.1 B
Cores	2	2	2	8	8
Frequencies	1.65 GHz	1.9 GHz	4 - 5 GHz	3 – 4 GHz	3.6 – 4.4+ GHz
L2 Cache	1.9MB Shared	1.9MB Shared	4MB / Core	256 KB per Core	256 KB per Core
L3 Cache	36MB	36MB	32MB	4MB / Core	10MB / Core
Memory Cntrl	1	1	2 / 1	2 / 1	2 / 1
Architecture	Out of Order	Out of Order	In of Order	Out of Order	Out of Order
LPAR	10 / Core	10 / Core	10 / Core	10 / Core	20 / Core

9.2.2 Larger L3 cache

By utilizing the smaller feature size IBM enlarged the L3 cache size from 4 MB/core (in the POWER7) to 10 MB/core (in the POWER7+), as shown below.



	POWER5	POWER5+	POWER6	POWER7	POWER7+
Technology	130nm	90nm	65nm	45nm	32nm
Size	389 mm^2	245 mm^2	341 mm^2	567 mm^2	567 mm^2
Transistors	276 M	276 M	790 M	1.2 B	2.1 B
Cores	2	2	2	8	8
Frequencies	1.65 GHz	1.9 GHz	4 - 5 GHz	3 – 4 GHz	3.6 – 4.4+ GHz
L2 Cache	1.9MB Shared	1.9MB Shared	4MB / Core	256 KB per Core	256 KB per Core
L3 Cache	36MB	36MB	32MB	4MB / Core	10MB / Core
Memory Cntrl	1	1	2 / 1	2 / 1	2 / 1
Architecture	Out of Order	Out of Order	In of Order	Out of Order	Out of Order
LPAR	10 / Core	10 / Core	10 / Core	10 / Core	20 / Core

Figure: L3 cache sizes of the POWER models [57]

9.2.3 Doubling the SP FP performance

IBM redesigned the FP unit of the POWER7 as follows:

- Both the POWER7/7+ have four FP pipelines such that each can execute either two SP FP (Single Precision FP) or a single DP FP (Double Precision FP) operation.
- In the POWER7 two FP pipelines can execute together a 2-way SP FP SIMD instruction.
- By contrast, two POWER7+ FP pipelines are capable to execute together a 4-way SP FP SIMD instruction.
- In this way POWER7+ doubles the SP FP performance.
- As Multiply-Add instructions perform in parallel two operations, the four FP pipelines of the POWER7+ can execute altogether 16 SP FLOPs per cycle.

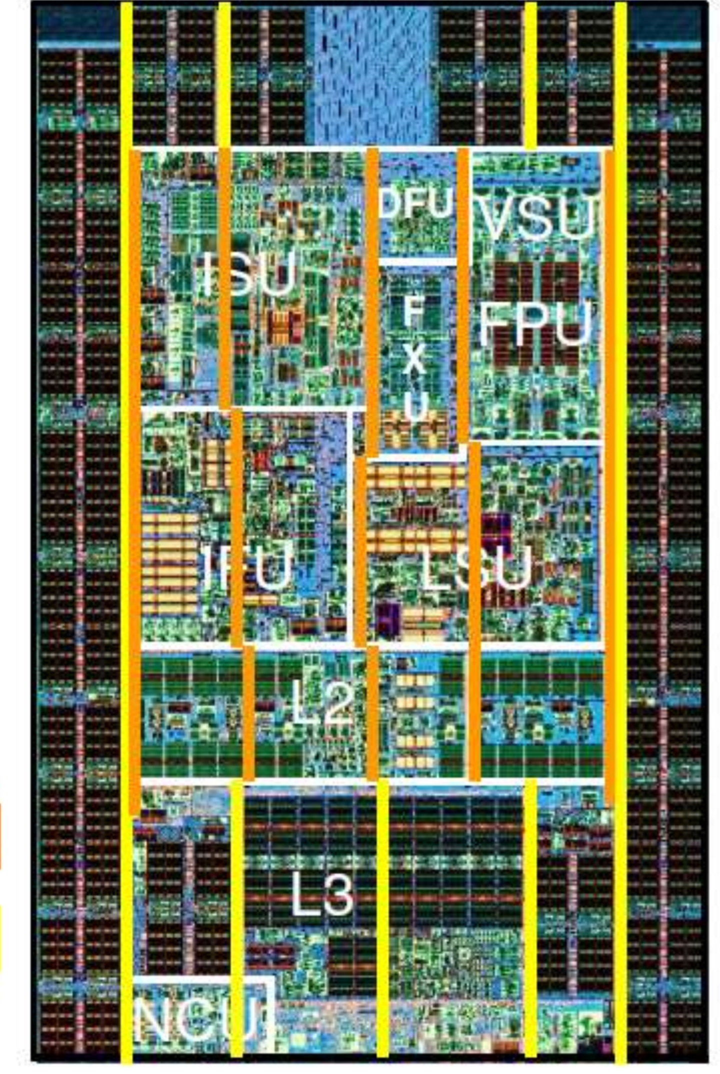
SPFP: Single Precision Floating-Point
DPFP: Double Precision Floating-Point

9.2.4 Added power gating regions for cores and L2/L3 caches [70]

- Added Power Gating regions for Core/L2 & L3 regions

Core/L2 Power-Gating

L3 Power-Gating



9.2.5 Remarks to the memory subsystem of the POWER7+

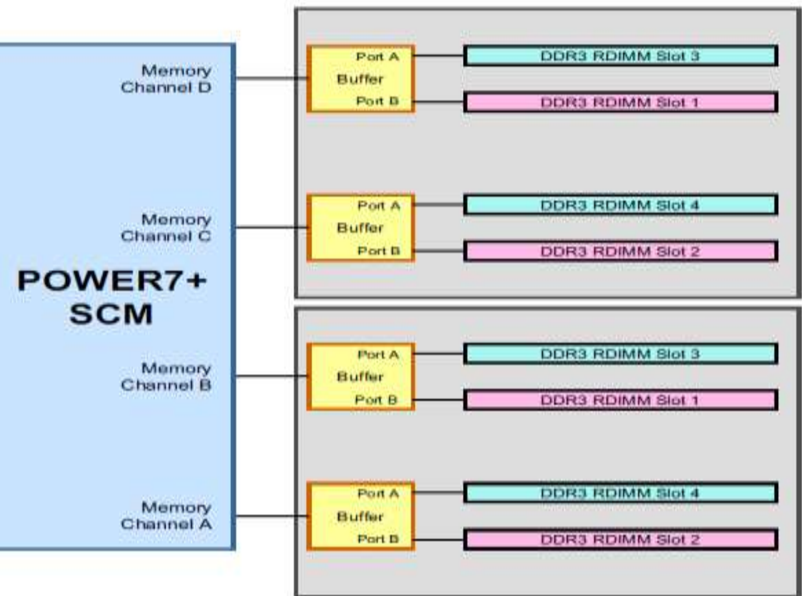
- With POWER7+ based servers IBM continued to implement both commodity DIMM and FB-DIMM based memory subsystems.
- Although the memory subsystem of the POWER7+ did not show improvements vs. the previous POWER7 based memory subsystems, subsequently we briefly discuss the memory configurations of POWER7+ systems as well, to round up the presentation given about the evolution of memory systems in IBM's POWER models, since the POWER7+ was the last model of this line with SuperNova based FB-DIMM memories.
- We note that the subsequent POWER8 made already use of Centaur Memory Buffers and commodity DDR3 DRAM chips.

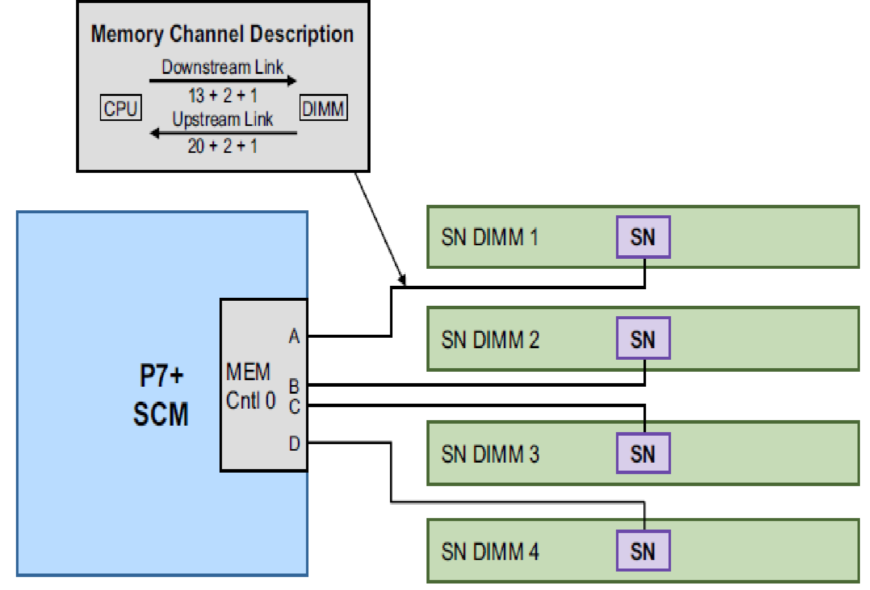
Memory configurations of the POWER7+ [71], [72] -1

Memory configurations of the POWER7+

Commodity DIMM based configurations

FB-DIMM-based configurations





Commodity 240 pin DDR3-1066 RDIMMs
Advanced Buffer Chips are on the processor card

Proprietary 96 mm tall, 276 pin SuperNova based
Fully buffered DDR3-1066 DIMMs
Buffers are integrated to the DIMM

Examples

- Power 710/730 (1 MC/4 ports/2 DIMM channels)
- Power 720/740 (1 MC/4 ports/2 DIMM channels)
- Power 750/760 (1 MC/4 ports/2 DIMM channels)

Power 770/780 (2 MC/2 ports/1 FB-DIMM channels)

Memory configurations of the POWER7+ -2

Memory configurations of the POWER7+

Commodity DIMM based configurations	**FB-DIMM-based configurations**
Commodity 240 pin DDR3-1066 RDIMMs Advanced Buffer Chips are on the system board	Proprietary, 96 mm tall 276 pin SuperNova based Fully buffered DDR3-1066 DIMMs Buffers are integrated to the DIMMs
Power 710/730 Power 720/740 Power 750/760	Power 770/780
-- 8 GB (2 x 4 GB), DDR3-1066 (FC EM08) – 16 GB (2 x 8 GB), DDR3-1066 (FC EM4B) – 32 GB (2 x 16 GB), DDR3-1066 (FC EM4C) – 64 GB (2 x 16 GB), DDR3-1066 (FC EM4D)	–32 GB (4 X 8 GB), DDR3-1066 (FC EM40) – 64 GB (4 X 16 GB), DDR3-1066 (FC EM41) – 128 GB (4 X 32 GB), DDR3-1066 (FC EM42) – 256 GB (4 X 64 GB), DDR3-1066 (FC EM44)

Examples

E.g. FC EM08 2x4GB RDIMM-1066, 240 pin

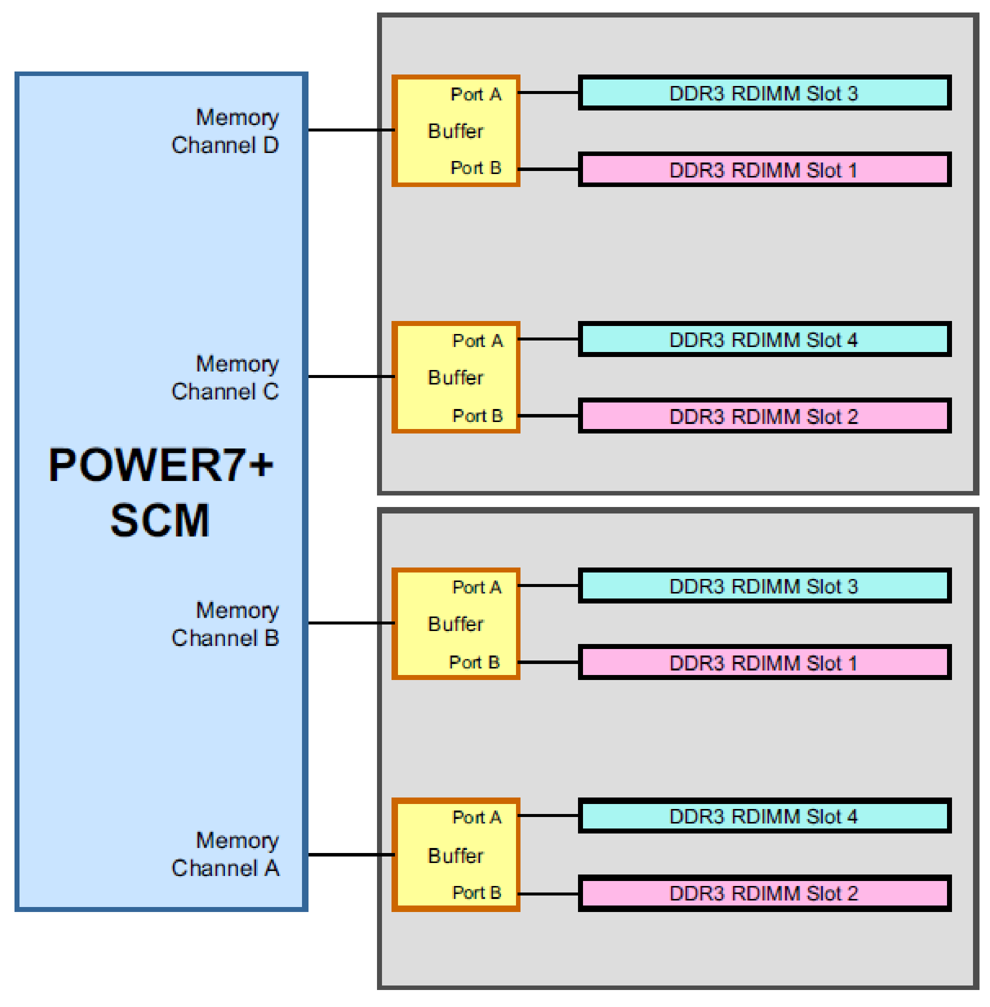
E.g. FC EM42 4X32GB DDR3-1066 DIMMS

Source of the pictures: eBay

Commodity DIMM based configurations -1

Low and midrange servers, such as the Power 710 to 760 models, have commodity DIMM based memories built of DDR3-1067 RDIMMs, as shown in the next two Figures.

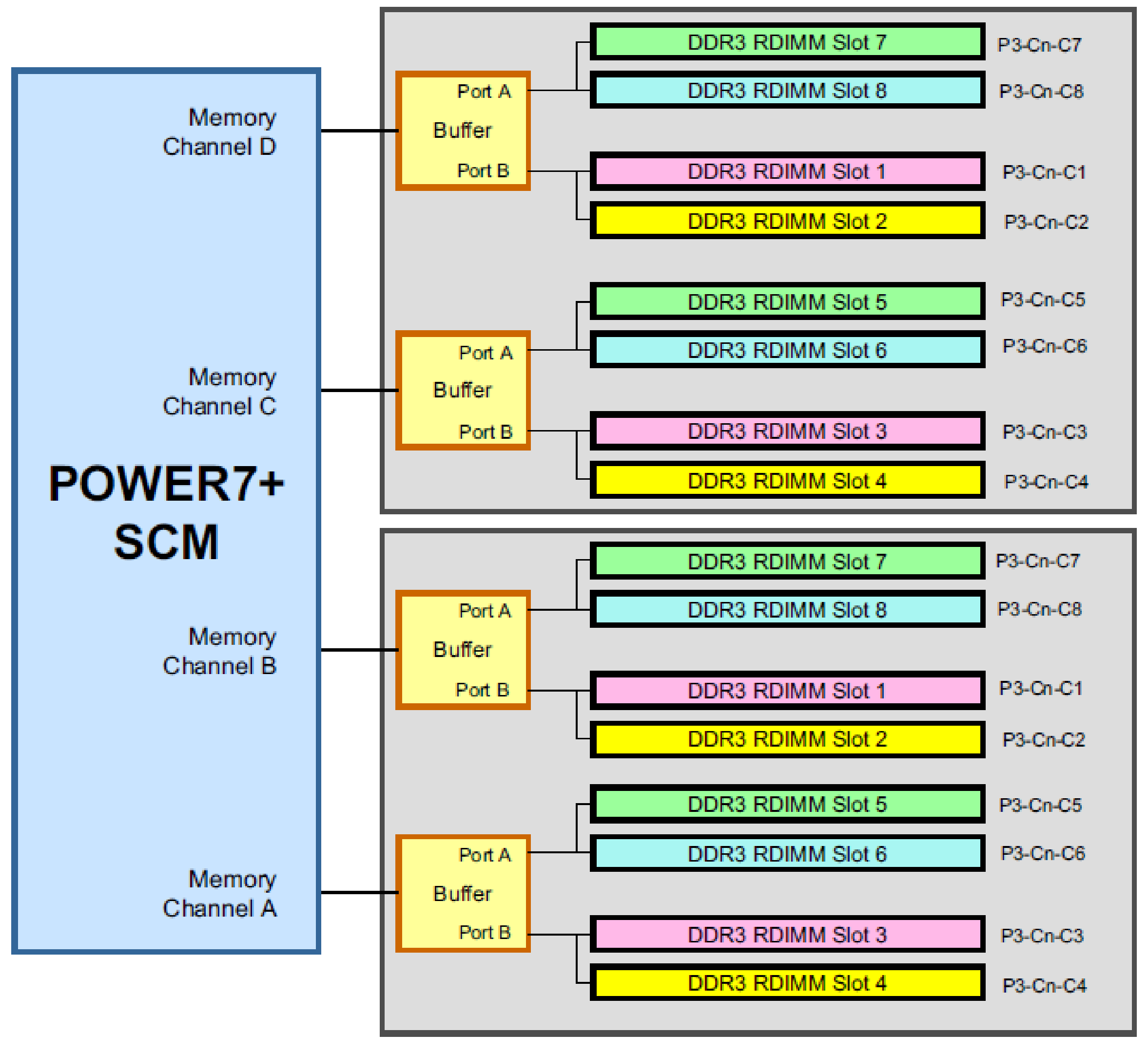
Example: The commodity DDR3-1067 based memory of the Power 710/730 servers [71]



4x2x8x1066

68.224 GB/s

Example: The commodity DDR3-1067 based memory of the Power 720-760 servers [73]



68.224 GB/s

9.2.5 Remarks to the memory subsystem of the POWER7+ (7)

Example: The 240 pin FC EM08 2x4GB DDR3-1066 commodity RDIMM

Source of the picture: ebay

Commodity DIMM based configurations -2

- As the above Figures show, both models makes use of a single memory controller that provides four ports.
- Each port is connected by a serial high speed bus to a Memory Buffer.
- Memory Buffers serve two (Power 710/730) or two pairs (Power 720/740/750/760) of DDR3 RDIMM-1067 modules, as indicated in the Figure before.

9.2.5 Remarks to the memory subsystem of the POWER7+ (9)

Per socket bandwidth of commodity DIMM based memory subsystems

The maximum per socket memory bandwidth of the commodity DIMM based memory subsystems is constrained both by the serial bus and the available DIMMs.

The serial bus constrained bandwidth of the commodity DIMM based memory subsystems (Power 710 – 760) with

- 1 memory controllers and
- 4 ports per controller is:

1 memory controller x 4 ports x (1 B write + 2 B read) x 6.4 GHz = 76.8 GB/s

The DIMM constrained bandwidth of the commodity memory subsystems (Power 710 – 760) with

- a single memory controller,
- 4 ports per controller and
- dual RDIMMs per Memory Buffer is:

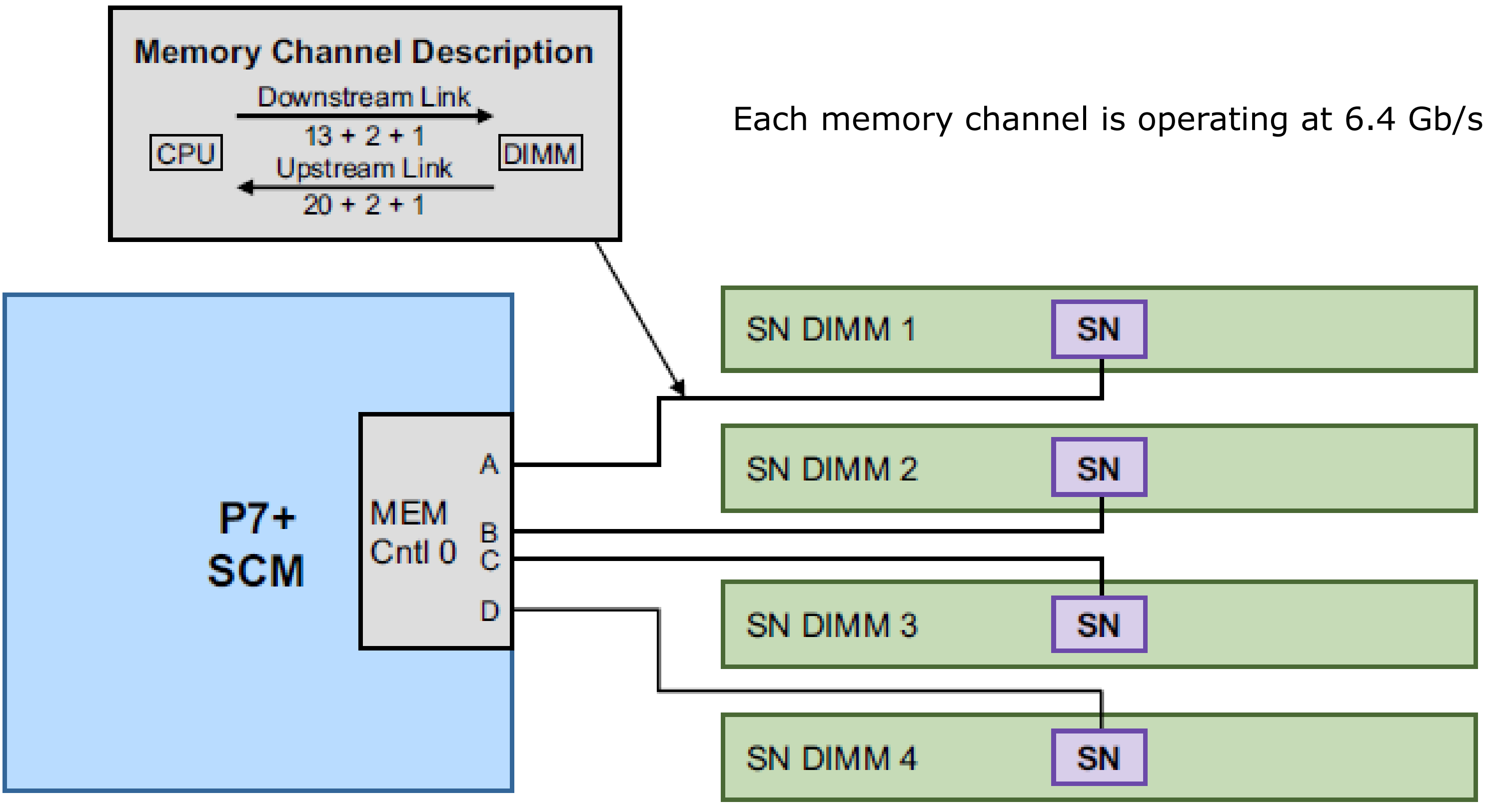
1 memory controller x 4 ports x 2 RDIMMs x 8 B x 1066 Mtransfers/s = 68.25 GB/s

As seen, at last the DIMM constrained bandwidth is lower and limits the theoretical maximum bandwidth of POWER7+ based memory subsystems built up of commodity DIMMs to 68.25 GB/s.

FB-DIMM based configurations

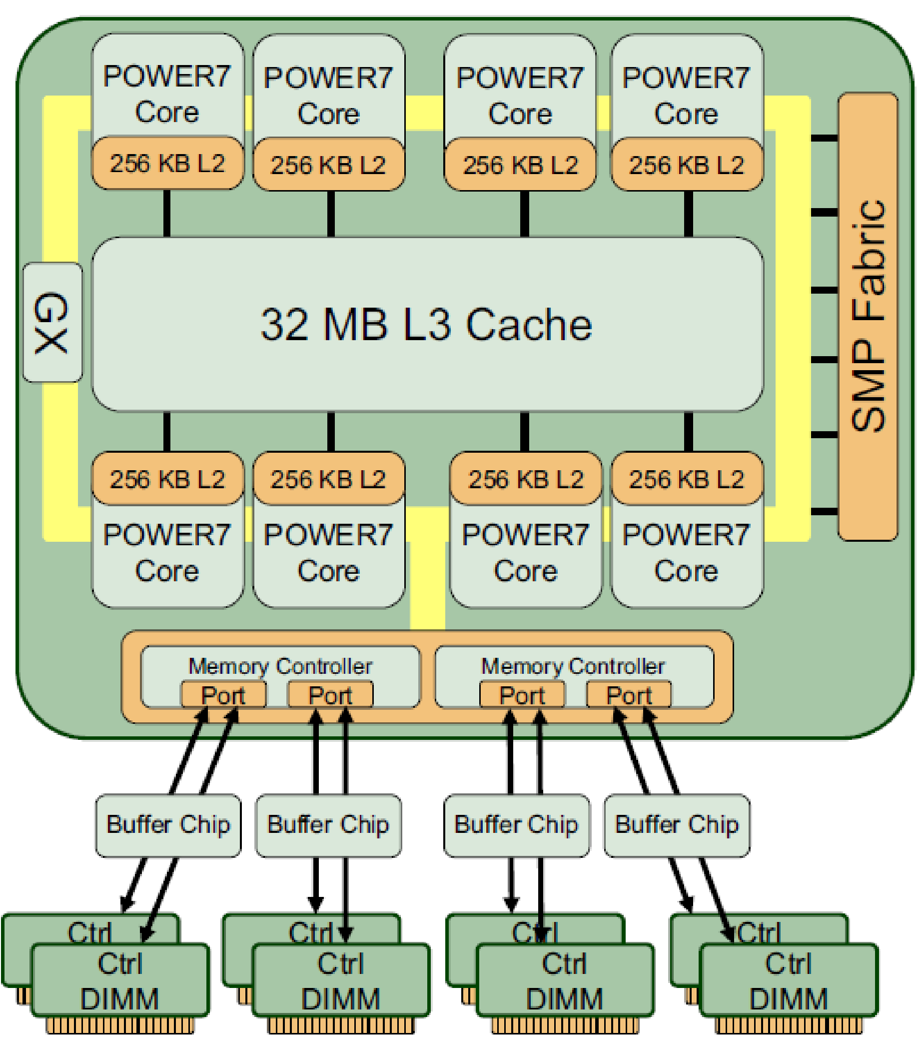
- The memory subsystem of the high end Power 770/780 server models is built of proprietary, 96 mm tall, 276 pin fully bufferedDDR3-1066 DIMMs with the Memory Buffers integrated to the DIMM according to the SuperNova technology, as indicated in the next Figure.
- Here we note that IBM did not design a POWER7+ based Power 795 server model.

Principle of the Supernova technology as used in the POWER7+ based Power 770/780 models [72]



SN: SuperNova

Example: Implementation of the SuperNova technology in the Power 770/780 models [72] -1



9.2.5 Remarks to the memory subsystem of the POWER7+ (13)

Remark

We note that both foregoing contradictory Figures originate from the same IBM document [72].

Nevertheless, no related document could be found to resolve the contradiction, so subsequently we took for granted the two memory controller version to remain in concert with the bandwidth figures given in the same document.

Principle of the Supernova technology as used in the Power 770/780 models [72] -2

As shown in the last Figure, memory is attached by dual memory controllers, with four ports per controller (here we take for granted the last but one Figure).

Each of the four ports is connected through a serial high speed (6.4 GT/s) channel to a proprietary FB-DIMM-1067 memory module pair

Consequently, a processor chip provides altogether eight high speed memory channels.

Example: FB-DIMM-1066 modules of the Power 770/780 servers [72]

These modules are IBM proprietary quad high (96 mm) 276 pin memory cards with DDR3-1066 DRAM chips, installed in 4 (termed as DIMM quads).

The available FB-DIMM module types are:

- 32 GB (4 X 8 GB), DDR3-1066 (FC EM40)
- 64 GB (4 X 16 GB), DDR3-1066 (FC EM41)
- 128 GB (4 X 32 GB), DDR3-1066 (FC EM42)
- 256 GB (4 X 64 GB), DDR3-1066 (FC EM44)

4x 32 GB
Four ranks
Each rank: 10 DRAMs
276 pins

Source of the picture: ebay

Figure: Example FB-DIMM module (the FC EM42)

Per socket bandwidth of FB-DIMM based memory subsystems

The maximum per socket memory bandwidth of FB-DIMM based memory subsystems is constrained both by the serial bus and the available FB-DIMMs.

The serial bus constrained bandwidth of the FB-DIMM based memory subsystems of the Power 770/780 servers with

- 2 memory controllers and
- 4 ports per controller is:

2 memory controller x 4 ports x (1 B write + 2 B read) x 6.4 GHz = 153.6 GB/s

The DIMM constrained bandwidth of the FB-DIMM based memory subsystems of the Power 770/780 servers with

- 2 memory controllers,
- 2 ports per controller and
- 2 FB-DIMMs per memory channel is:

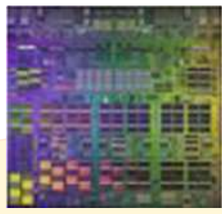
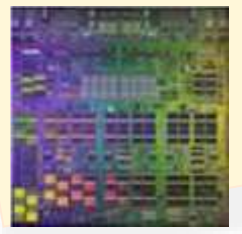
2 memory controller x 2 ports x 2 FB-DIMM channels x 8 B x 1066 Mtransfers/s =
= 68.25 GB/s

As seen, at last the FB-DIMM constrained bandwidth is lower and limits the theoretical maximum bandwidth of POWER7+ based memory subsystems built up of FB-DIMMs to 68.25 GB/s.

9.3 Key innovations of the POWER7+

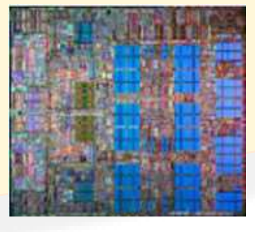
- 9.3.1 Add-on cryptographic accelerators
- 9.3.2 Introducing the Winkle idle state
- 9.3.3 Remarks to Using Critical Path Monitors (CPMs) in the POWER7+

9.3 Key innovations of the POWER7+ (Die photos from [3])

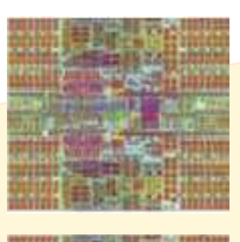
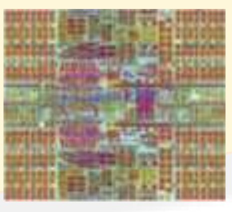
POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
with SMI chips
GX I/O bus
Support for SMP



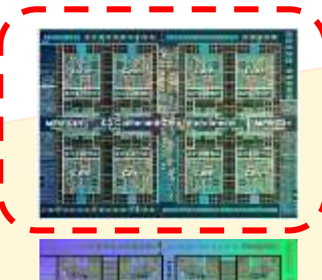
POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
clock gating

POWER6/6+
65/65 nm

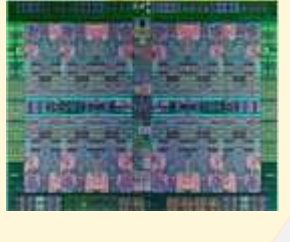
Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
Critical Path
Monitors
Nap idle mode



POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconn.
Energy Scale 2 with
Per core fc
Dyn. fan managm.
Sleep idle state
*Accelerators for
cryptography
*Winkle idle state

*POWER7+



POWER8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip μc for PM
Per-core Vdd
Per-core VRMs

2001 | 2004 | 2007 | 2010 | 2014

9.3.1 Add-on cryptographic accelerators

9.3.1 Add-on cryptographic accelerators

- POWER7+ accelerators support encryption according to the SSL standard by off-loading related tasks from the CPU speeding-up in this way the processing of encrypted files.
- POWER7+ provides the following accelerators [70]:
  - Random Number Generator (RNG)

    Needed in cryptography
  - Crypto offload accelerators

    They are cryptographic engines to relieve the processor from running calculation intensive cryptographic algorithms like AES, SHA and RSA.
- Subsequently, we briefly introduce the above notions.

9.3.1 Add-on cryptographic accelerators (2)

SSL [74]

- SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client — typically between a web server and a browser; or a mail server and a mail client (e.g. Outlook).
- SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely.

9.3.1 Add-on cryptographic accelerators (3)

AES

- AES (Advanced Encryption Standard): is a specification for the encryption of electronic data issued by NIST (U.S. National Institute of Standards and Technology) in 2001.
- It is currently one of the most popular algorithms used for data transmission according to the SSL standard.
- The AES algorithm uses 128, 192 or 256 bit long cryptographic keys to encrypt and decrypt data in blocks of 128 bits.

Remark

- To support encryption also Intel introduced an ISA extension called AES-NI (Enhanced Encryption Standard-New Instructions).
- AES-NI includes 7 instructions and became supported first in Intel's 32 nm Westmere line of processors in 2010.

9.3.1 Add-on cryptographic accelerators (4)

SHA [75]

- SHA-0-3 (Secure Hash Algorithm) a series of cryptographic hash function standards issued by NIST.
- SHA-0 (1993) is the original version of the 160-bit hash function, it was withdrawn shortly after publication.
- SHA-1 (1995) is very similar to SHA-0 but corrects alleged weaknesses of SHA-0.
- SHA-2 (2001) aims at improving SHA-1.
- SHA-1 became widely used, but due to some weaknesses published in 2005 [75] it is not more considered as a standard providing enough security.

  So many large software companies, like Microsoft, Google or Mozilla announced that they will stop accepting SHA-1 certificates in SSL in 2017 and will accept only SHA-2 certificates.

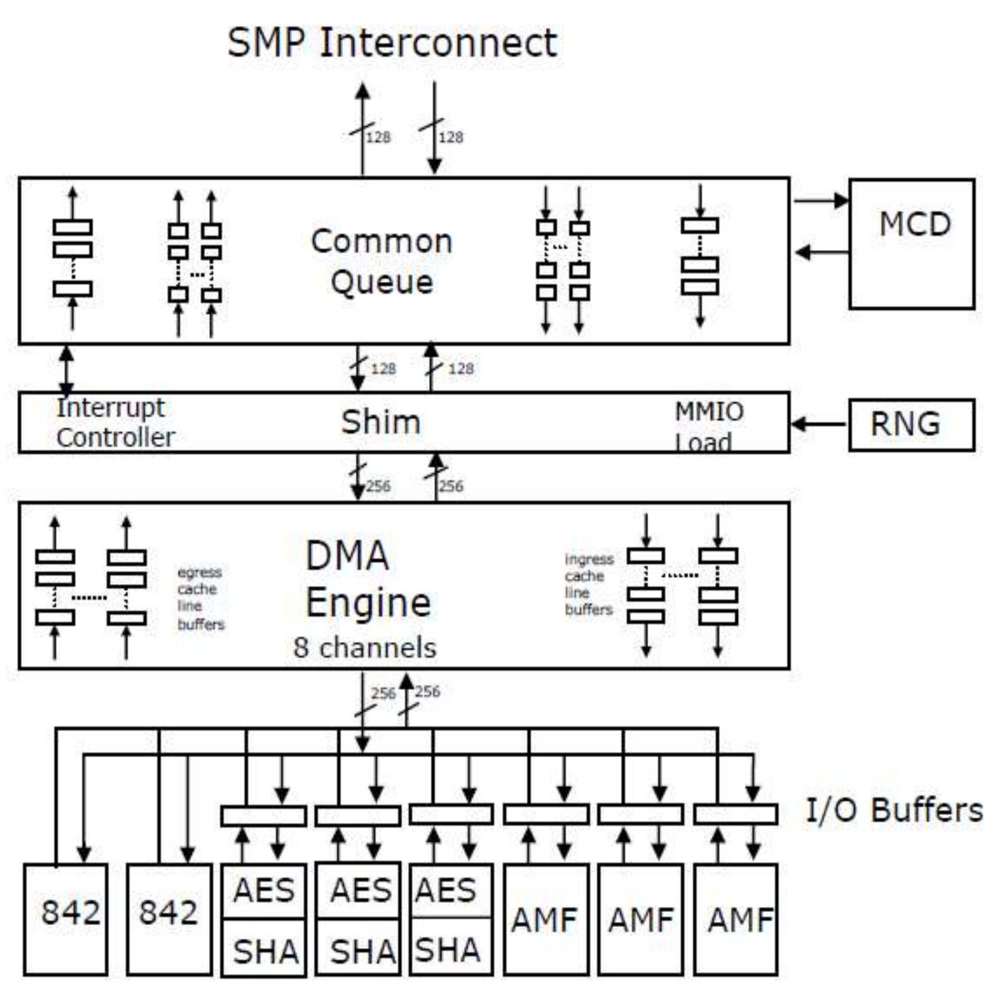
9.3.1 Add-on cryptographic accelerators (5)

RSA [76]

- It is an early algorithm publicly described by MIT scientists in 1977 for using in secure data transmissions.
- The name originates from the initials of the surnames of their inventors (Ron **R**ivest, Adi **S**hamir and Leonhard **A**dleman).
- With this algorithm a message is encrypted by using a public encryption key that can be decrypted by using a different decryption key which is secret.

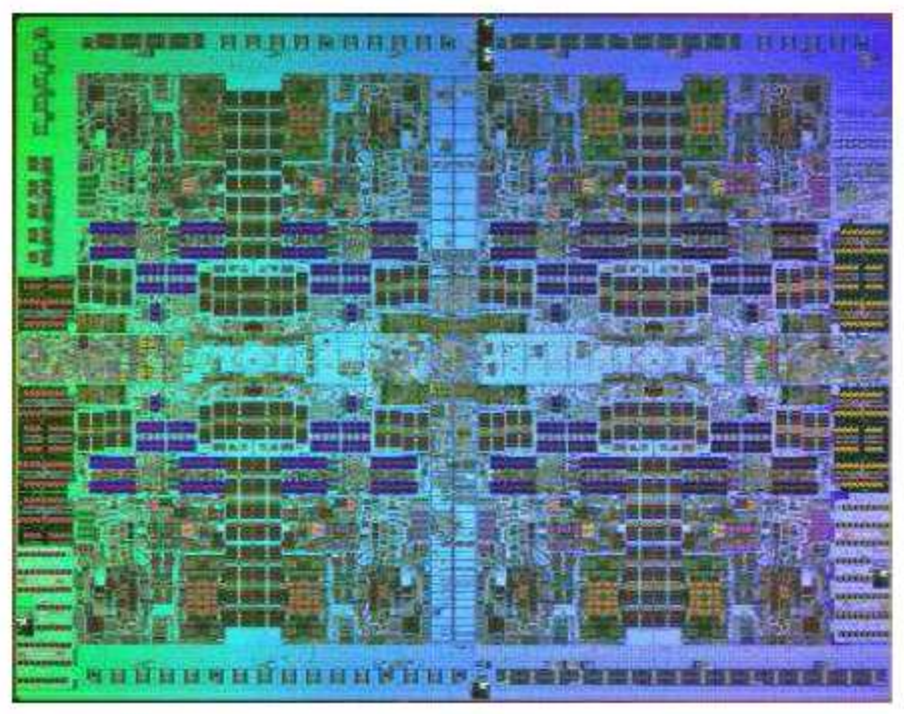
POWER7+ add-on accelerators [70]

- Advanced Encryption Standard engine
  - Modes: ECB, CBC, CTR, CCM, CCA, GCM, GCA, GMAC, CM, F8, XBC-MAC-96
  - Key lengths: 128b,192b, 256b
  - Three engines
- Secure Hash Algorithm engine
  - Modes: SHA-1, SHA-256, SHA-512, MD5
  - HMAC supported for SHA
  - Three engines
- Asymmetric Math Functions
  - Modular math functions for RSA (Rivest, Shamir, Adleman) and ECC (elliptic curve cryptography): mod add, mod subtract, mod inverse, mod reduction, mod multiplication, mod exponentiation, mod exponentiation CRT(integer only)
  - Point functions for ECC GF(p) and GF(2m): point add, point double, point multiply
  - RSA lengths: 512b,1024b, 2048b, 4096b
  - ECC GF(p) lengths: 192b, 224b, 256b, 384b, 521b (SuiteB)
  - ECC GF(2m) lengths: 163b, 233b,283b, 409b, 571b (SuiteB)
- Random Number Generator
  - All digital design which produces 64b random numbers accessible by MMIO load instructions
  - Correctness verified against the NIST Random Number Generator Test Suite
- Active Memory Expansion
  - IBM-proprietary algorithm with 8B-, 4B-, and 2B-phrase parsings
  - Throughput: Up to 8 bytes of compression or 8 bytes of decompression per bus cycle.
- MCD
  - Hardware to predict whether memory access is on-node or off-node.

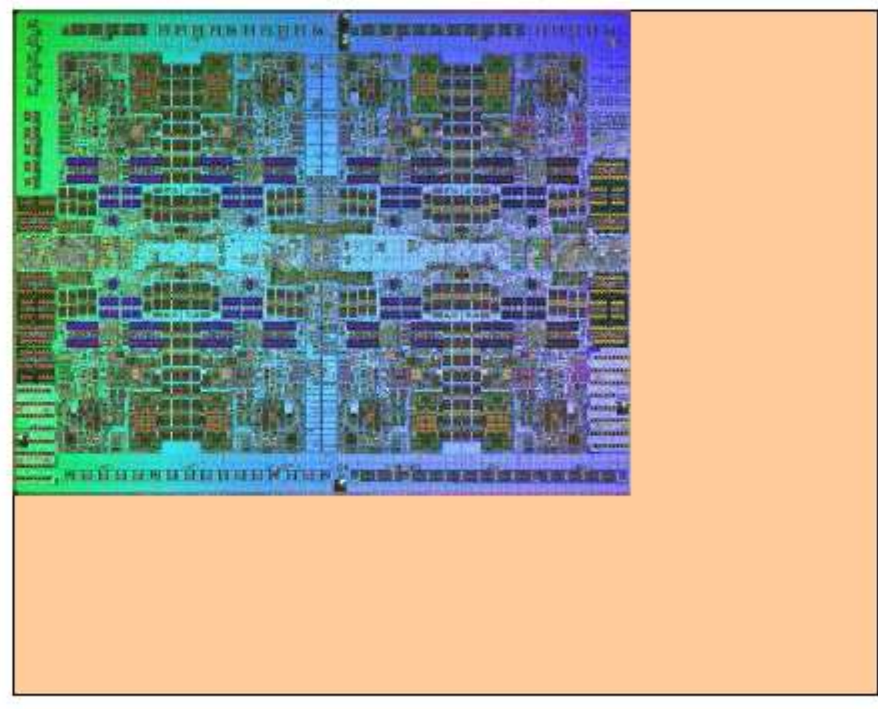


842:: Proprietary compression algorithm

Utilizing the chip area of the POWER7+ -1 [57]

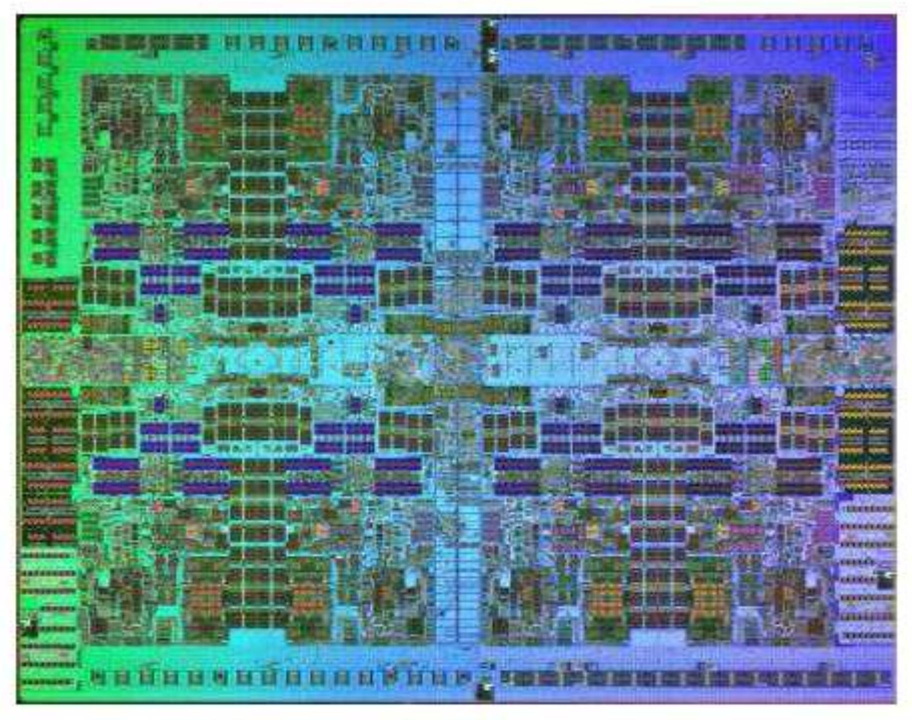


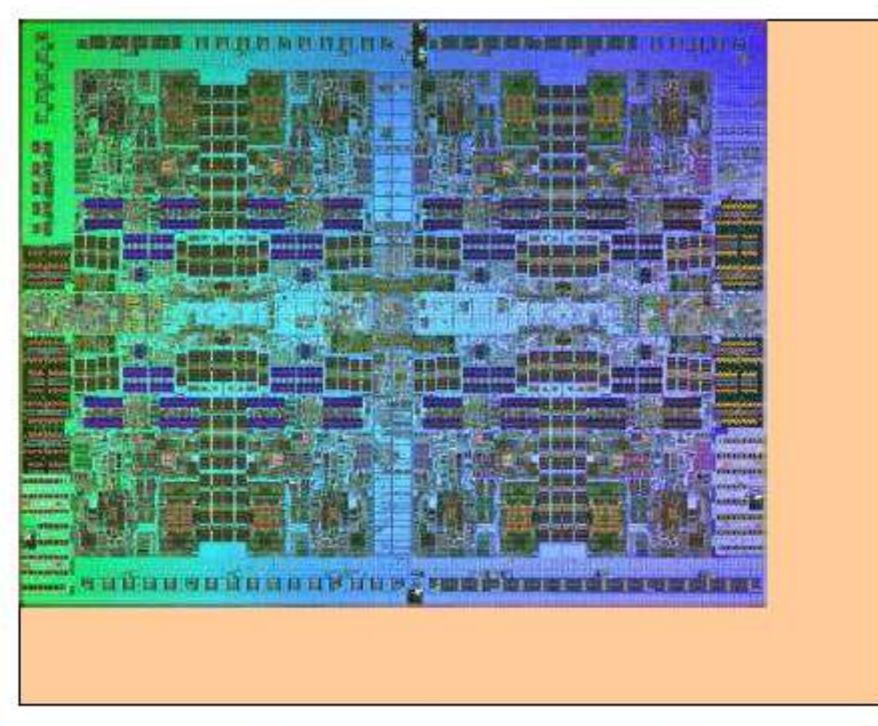
POWER7
45 nm



POWER7+
32 nm

Utilizing the chip area of the POWER7+ -2 [57]



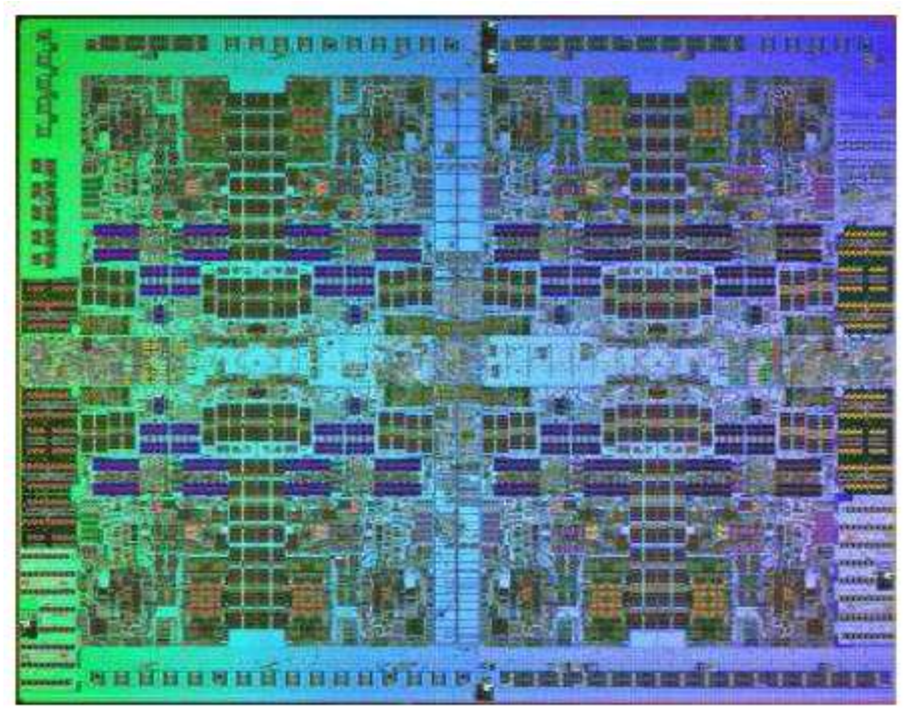


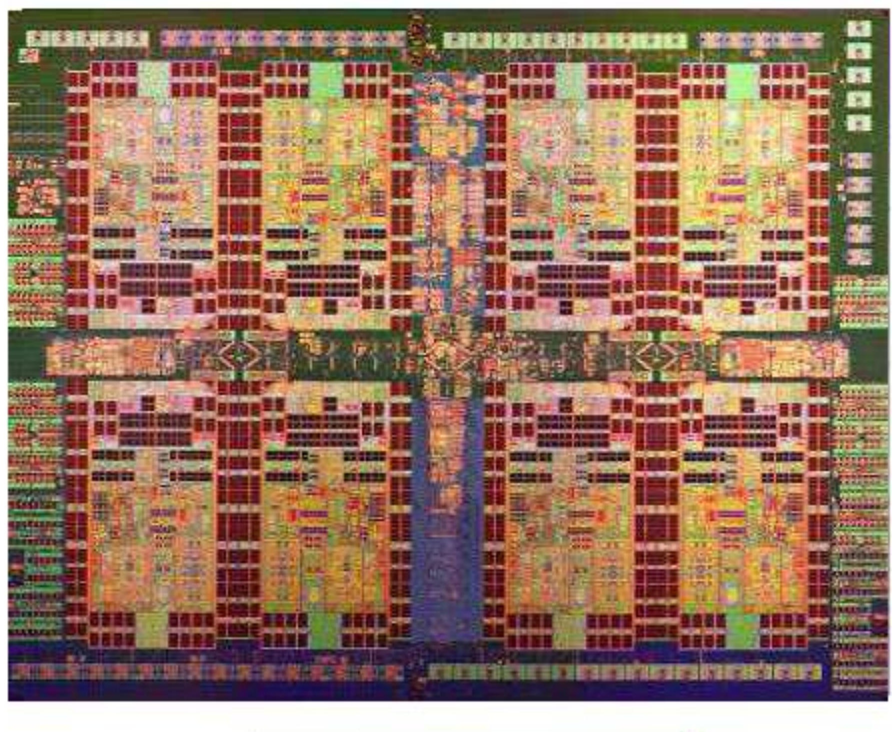
POWER7
45 nm

POWER7+
32 nm

Add additional Cache

Utilizing the chip area of the POWER7+ -3 [57]





POWER7
45 nm

POWER7+
32 nm

Add additional Cache

Add on Chip Accelerators

9.3.2 Introducing the Winkle idle state

9.3.2 Introducing the Winkle idle state

It is a more efficient idle state than Nap or Sleep, but has a longer latency, as shown below.

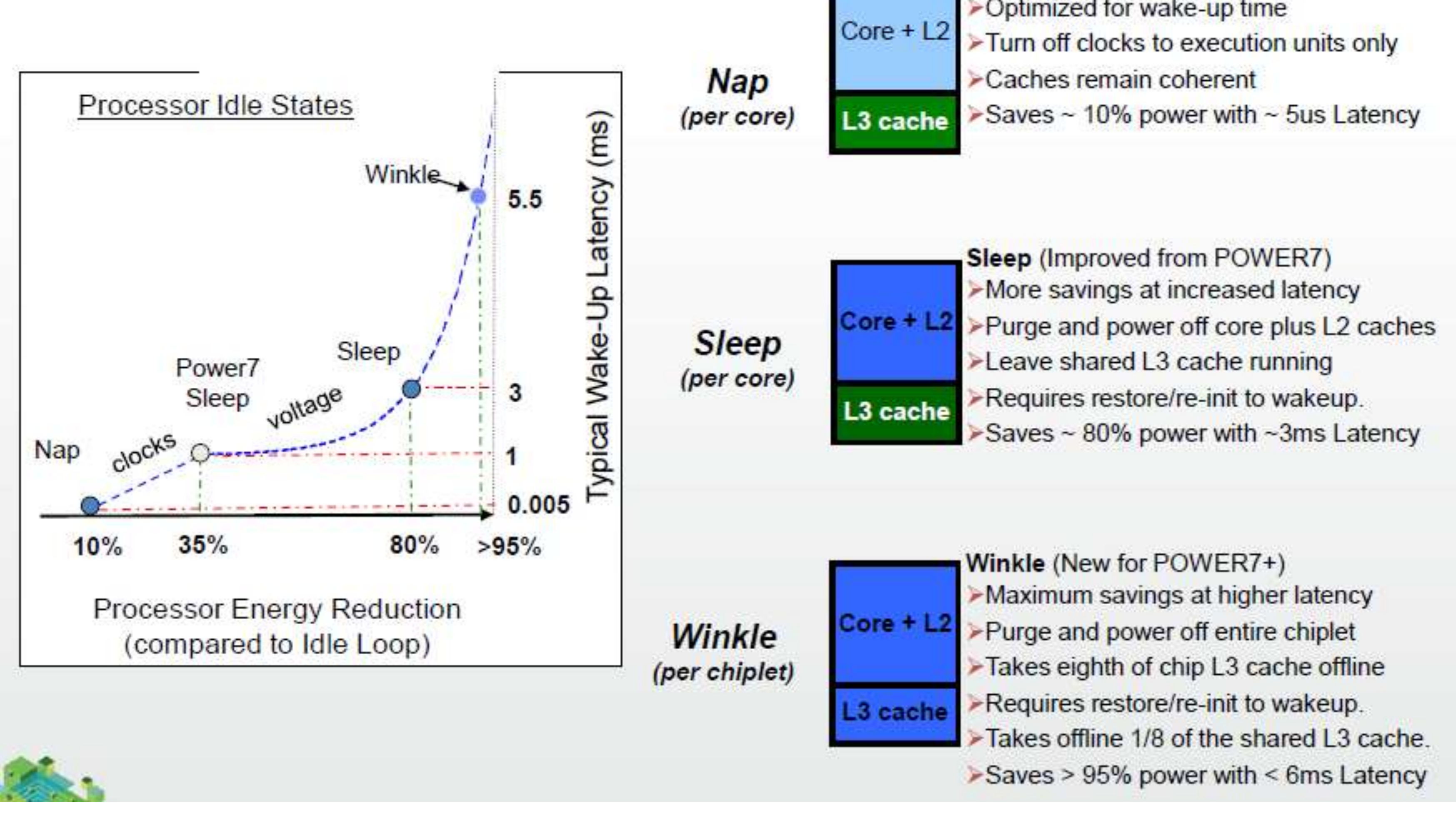


Figure: Idle states in the POWER7+ [77]

9.3.3 Remarks to Using Critical Path Monitors (CPMs) in the POWER7+ [139]

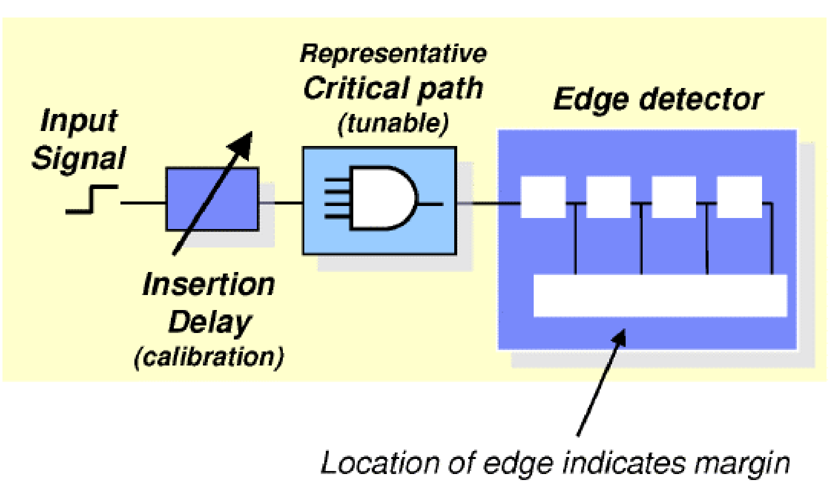
- CPMs were introduced in the POWER6.
- Here we give some details how CPMs were used in the POWER7+ [139].

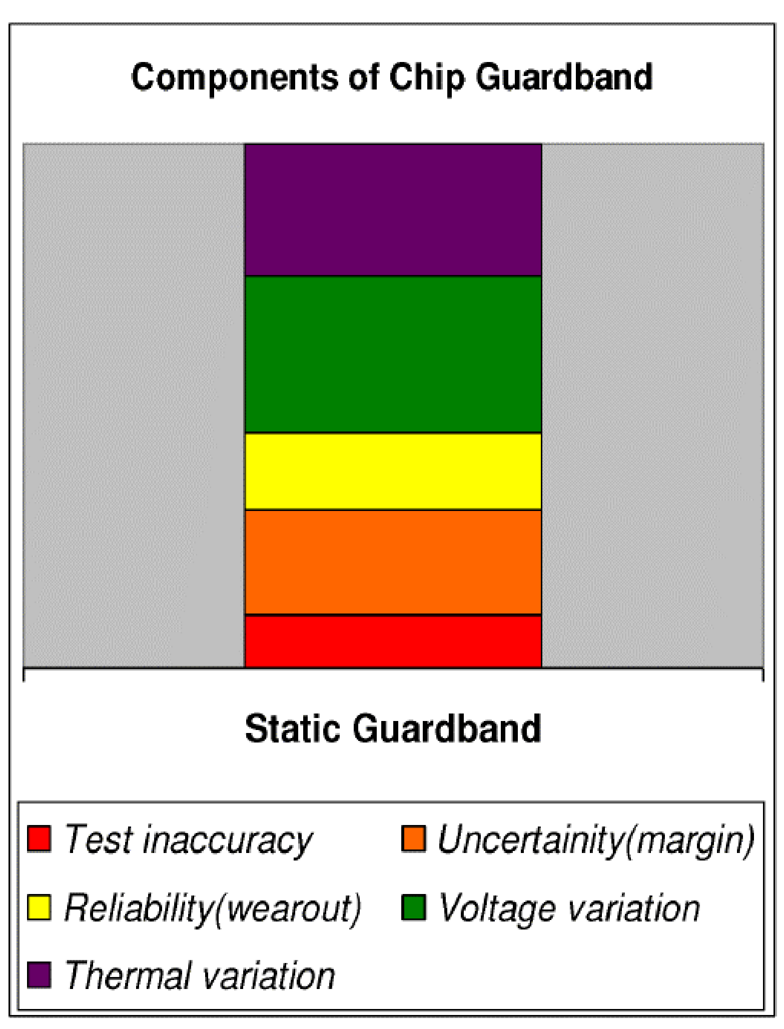
➢ **Conventional guardband**

- Static, conservative voltage margins for potential worst-case conditions
- Causes unnecessary loss of energy efficiency during typical server usage

➢ **Critical Path Monitor (CPM)**

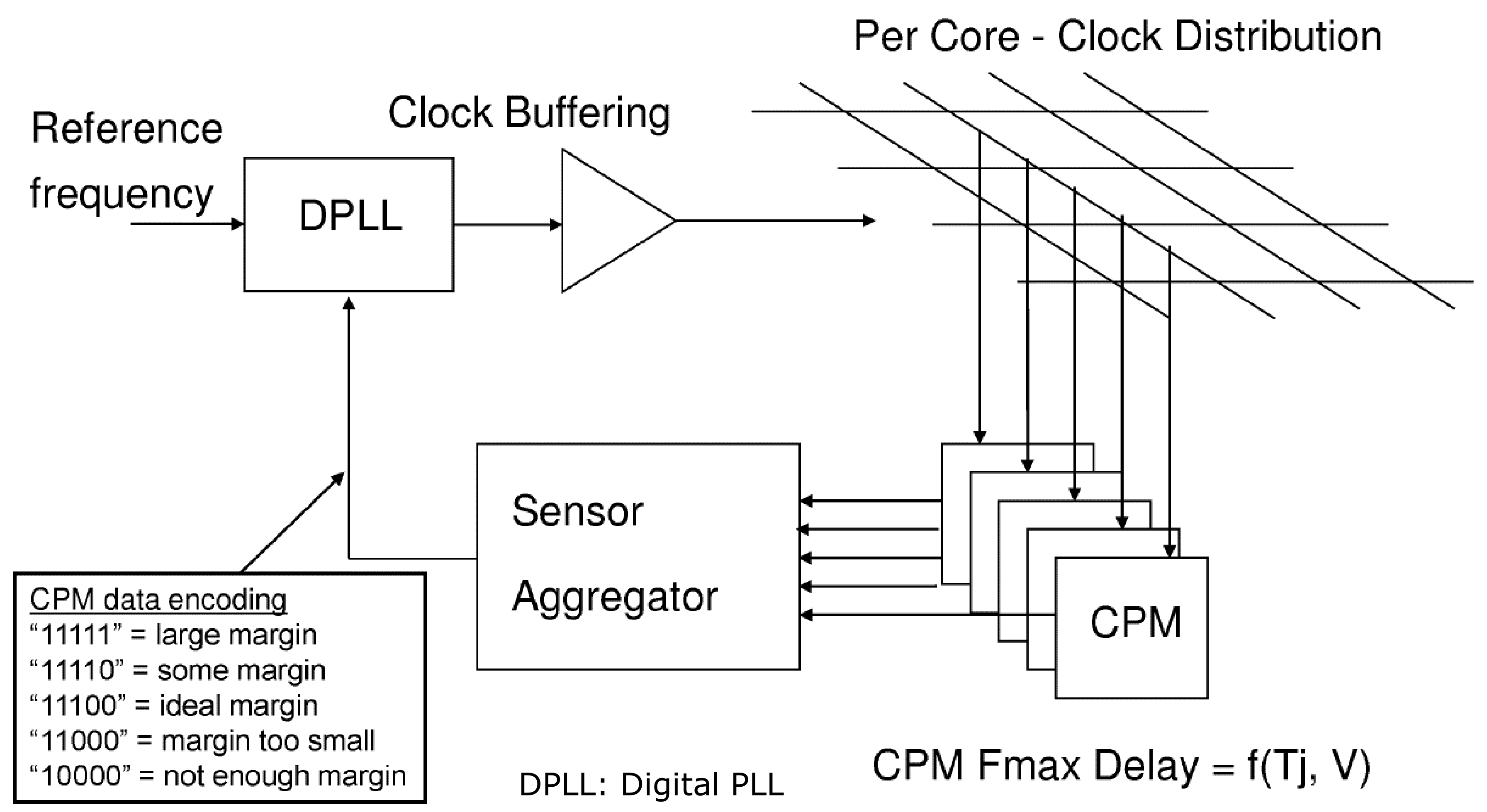
- Real Time detection of available circuit timing margin



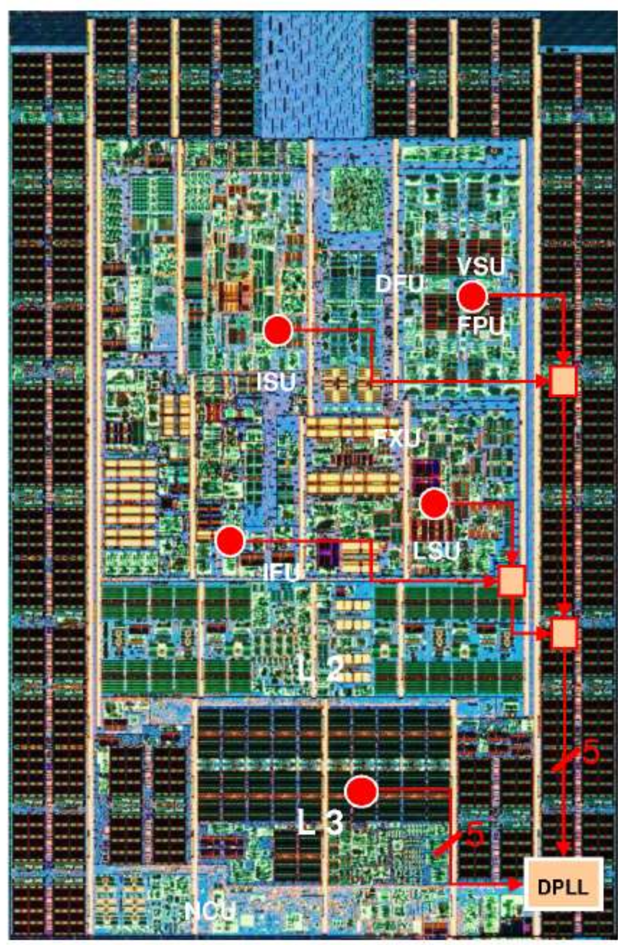


9.3.3 Remarks to Using Critical Path Monitors (CPMs) in the POWER7+ (2)

Real time guardbands – the DPLL/CPM feedback loop [139]



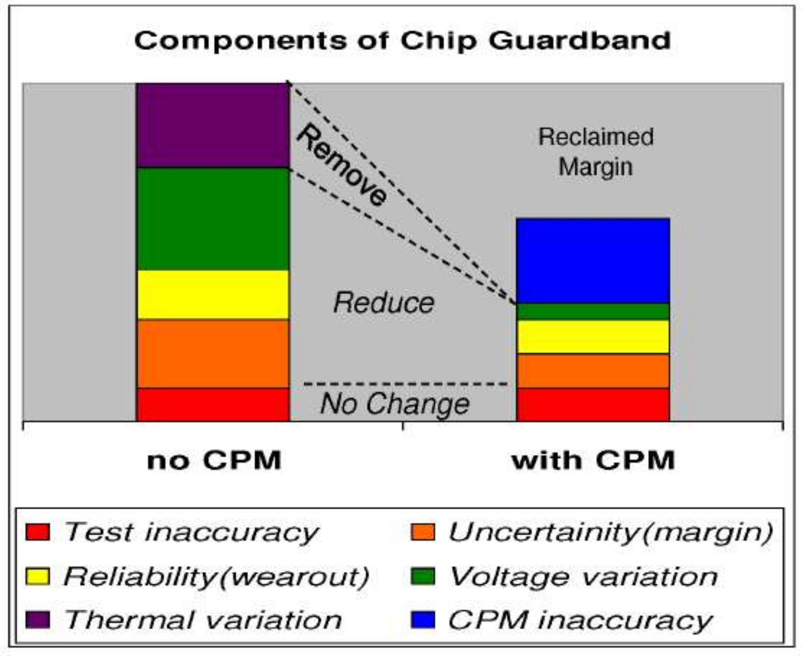
Placement an result of using CPMs in the POWER7+ [139]



CPMs are strategically placed in known hot spots typically near micro-architecture critical paths.

The real time feedback from CPMs can reduce how much margin is needed for various guardband components.

Real-time guardbanding will allow for greater energy efficiency.



10. POWER8

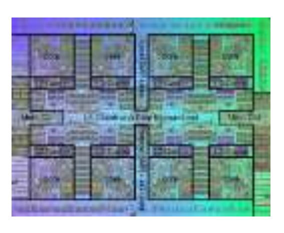
- 10.1 Introduction to the POWER8
- 10.2 Main enhancements of the POWER8
- 10.3 Key innovations of the POWER8
- 10.4 POWER8-based server lines

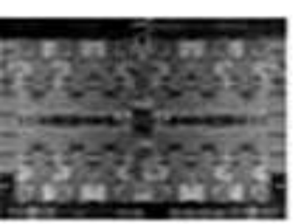
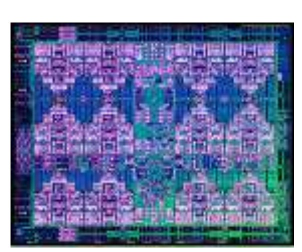
10.1 Introduction to the POWER8

10.1 Introduction to the POWER8

- Announced in 8/2013 at the Hot Chip Conference
- Introduced: 4/2014
- 22 nm technology
- 650 mm^2, 4.2 billion transistors

Key features of the POWER8



	POWER7	POWER7+	POWER8	POWER8+	POWER9
Launched	2/2010	10/2012	4/2014	Planned/cancelled	12/2017
Technology	45 nm	32 nm	22 nm		14 nm
Die size	567 mm2	567 mm2	650 mm2		693 mm2
Transistors	1.2 b	2.1 b	4.2 b		8.0 b
Cores (up to)	8	8	12		12 SMT8 cores 24 SMT4 cores
SMT	4-way	4-way	8-way		4-way/8-way
Typ. fc	3.72-4-42 GHz	3.1 -4.42 GHz	3.02-4.35 GHz		Up to 4 GHz
L2	256 KB/core	256 KB/core	512 KB/core		512KB/2 cores
L3	4 MB/core	10 MB/core	12 MB/core		10 MB/2 cores
Mem. contr.	2/1	2/1	8		8
Memory up to	DDR3-1066	DDR3-1066	DDR3-1600		DDR4-2666

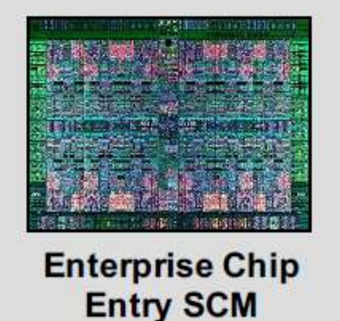
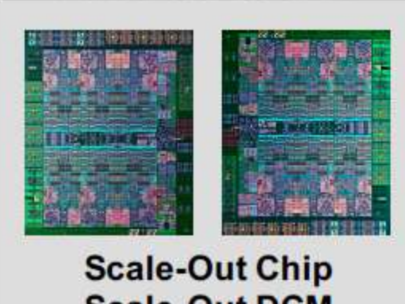
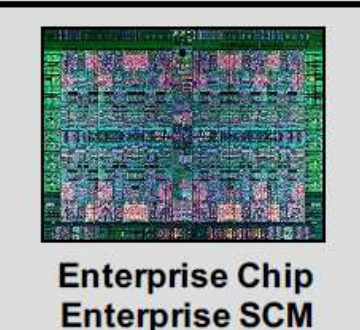
POWER8 chip alternatives and server lines [] -1

There were four different POWER8 chips,

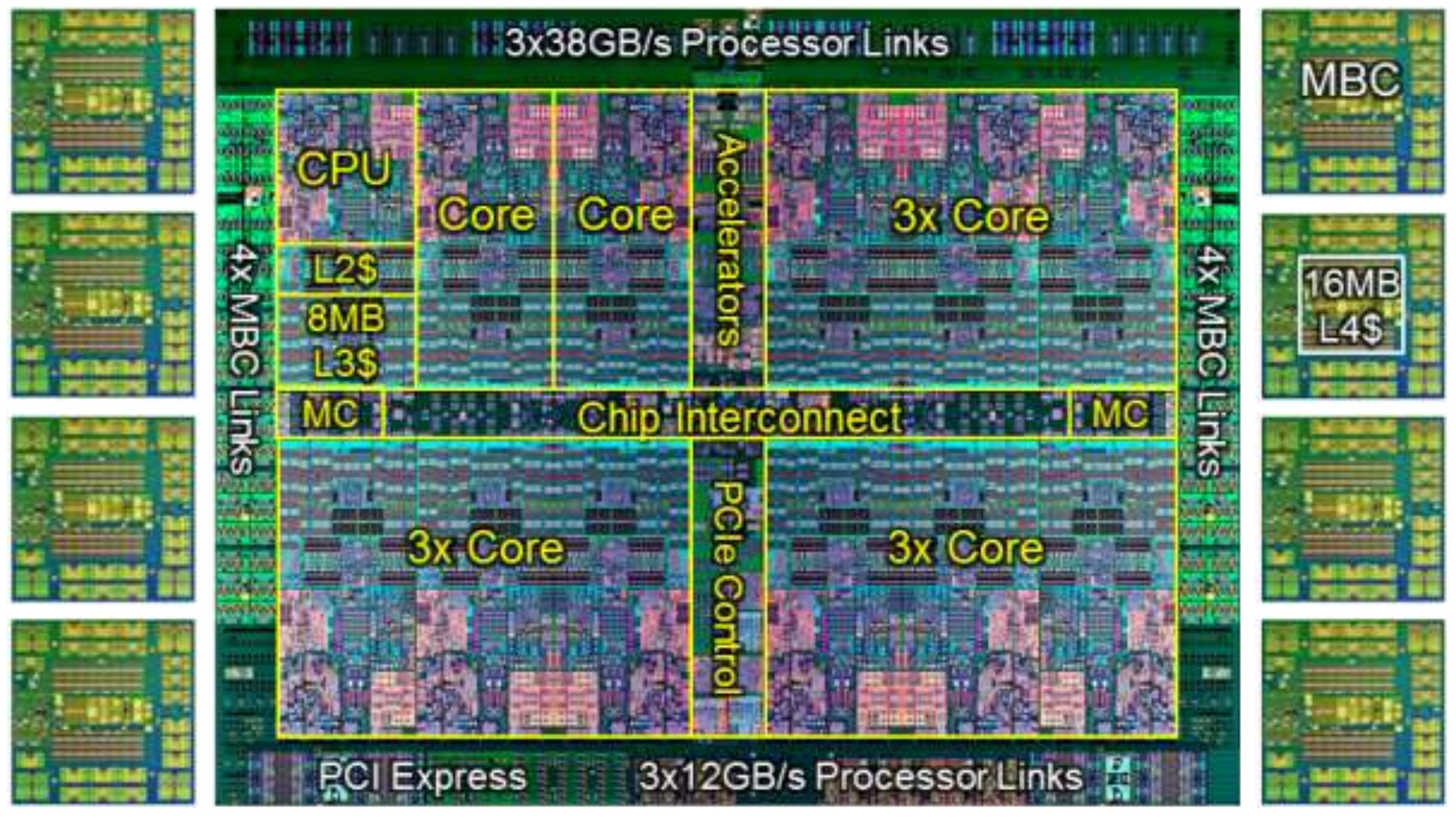
- one with six cores aimed at scale-out workloads and with two chips sharing a single package and
- one single-die, twelve-core chip aimed at bigger NUMA machines,
- one with twelve cores with the NVLink interconnect and
- one cost reduced processor also with twelve cores.

POWER8 chip alternatives and server lines -2 [140]

NVLINK GPU Enabled SuperCompute Node

Power S812LC	Power S822LC	Power S814/S22/S824	Power E870/E880
 Enterprise Chip Entry SCM	SuperCompute Chip SC SCM	 Scale-Out Chip Scale-Out DCM	 Enterprise Chip Enterprise SCM
Single Large Chip Up to 12 cores Up to 4 socket SMP Half memory Cost reduced	NVLINK GPU Attach	Dual Small Chips Up to 2 x 6 cores Up to 4 socket SMP Up to 48x PCI lanes Full Memory	Single Large Chip Up to 12 cores Up to 16 socket SMP Up to 32x PCI lanes Full Memory
		(For Scale-Out workloads)	(NUMA machines)

Die layout of the 12-core POWER8 die [130]



The POWER8 Scale-Out Dual-chip module (DLM) [141]

Technology

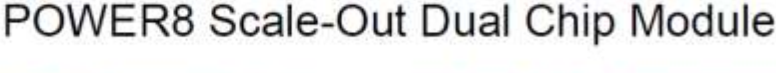
- 22 nm SOI, eDRAM, 15 ML 650 mm2

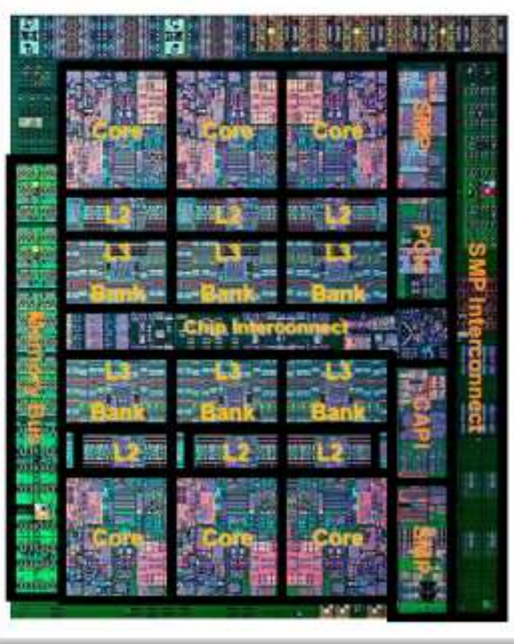
Cores

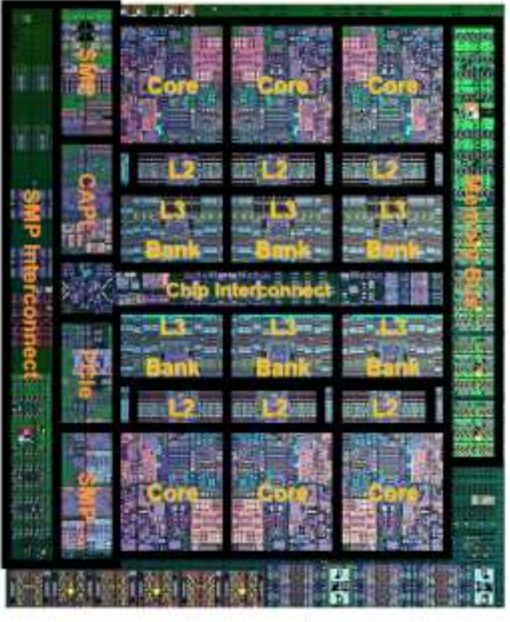
- 12 cores (SMT8)
- 8 dispatch, 10 issue, 16 execution pipes
- 2x internal data flows/queues
- Enhanced prefetching
- 64 KB data cache, 32 KB instruction cache

Accelerators

- Crypto and memory expansion
- Transactional memory
- VMM assist
- Data move/VM mobility







Caches

- 512 KB SRAM L2 / core
- 96 MB eDRAM shared L3

Memory

- Up to 230 GB/s sustained bandwidth

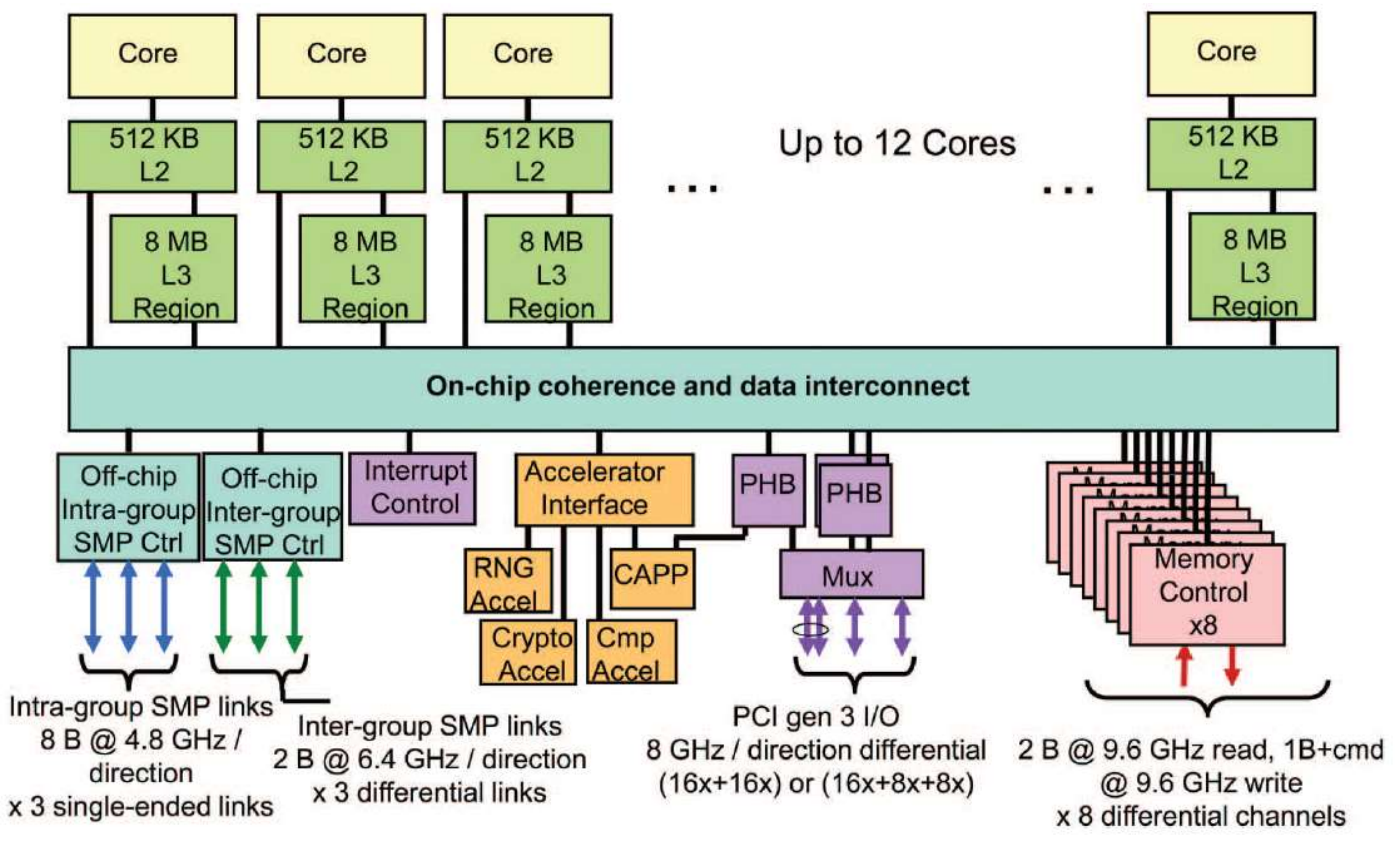
Bus Interfaces

- Durable open memory attach interface
- Integrated PCIe Gen3
- SMP interconnect
- CAPI

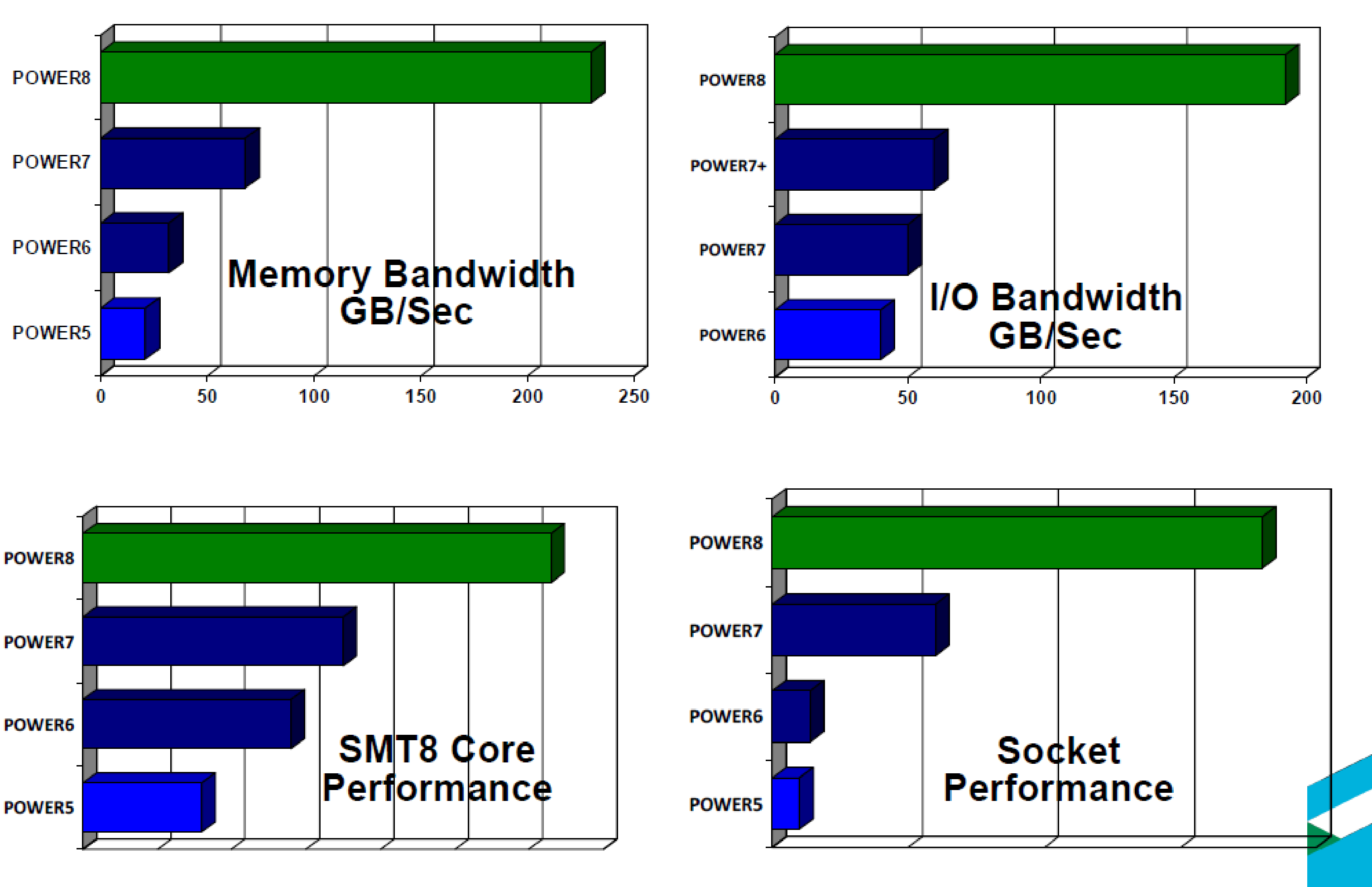
Energy Management

- On-chip power management microcontroller

Block diagram of a POWER8 processor [114]



POWER8 performance improvements [78]



Performance comparison of IBM's POWER8 with competing processors [130]

	IBM Power8	IBM Power7+	Oracle Sparc T5	Intel Xeon E7-4870
CPU Cores	12 cores	8 cores	16 cores	10 cores
Total Threads	96 threads	32 threads	128 threads	20 threads
Clock Speed	4.6GHz†	4.2GHz	3.6GHz	2.4GHz
L3 Cache	96MB	80MB	8MB	30MB
DRAM B/W	230GB/s	96GB/s	48GB/s	30GB/s
SPECint*	650†	308	436	226
SPECfp*	550†	242	350	187
IC Process	22nm SOI	32nm SOI	28nm HP	32nm HKMG
Die Area	650mm^2	567mm^2	478mm^2	513mm^2
Power (typ)	250W†	250W	300W†	130W TDP
Production	2Q14†	4Q12	1Q13	2Q11

IBM Power systems family (Stand April 2014) [69]

PowerVM PowerVC
PowerKVM PowerVP
PowerHA PowerSC

IBM Systems Software

POWER8

POWER7+

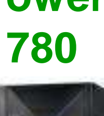
POWER7

Power 795

Power 780

Power 770

Power 760



Power 710
Power 730

Power S822

Power 720
Power 740

Power S814

Power 750

Power S824

PowerLinux 7R1 / 7R2 / 7R4

Power S812L

Power S822L

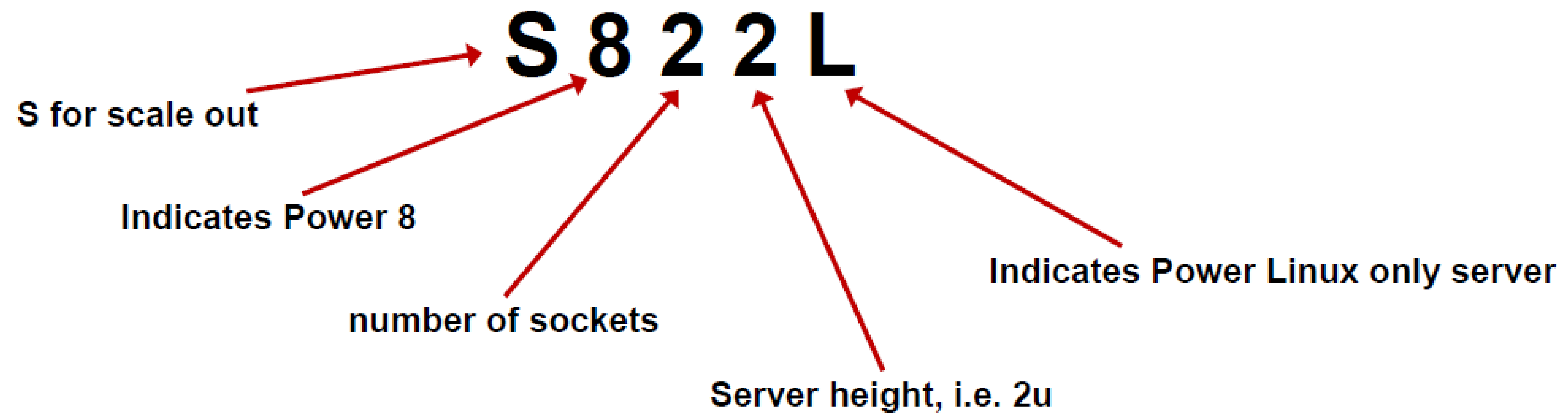
IBM PureSystem

IBM Flex System p460

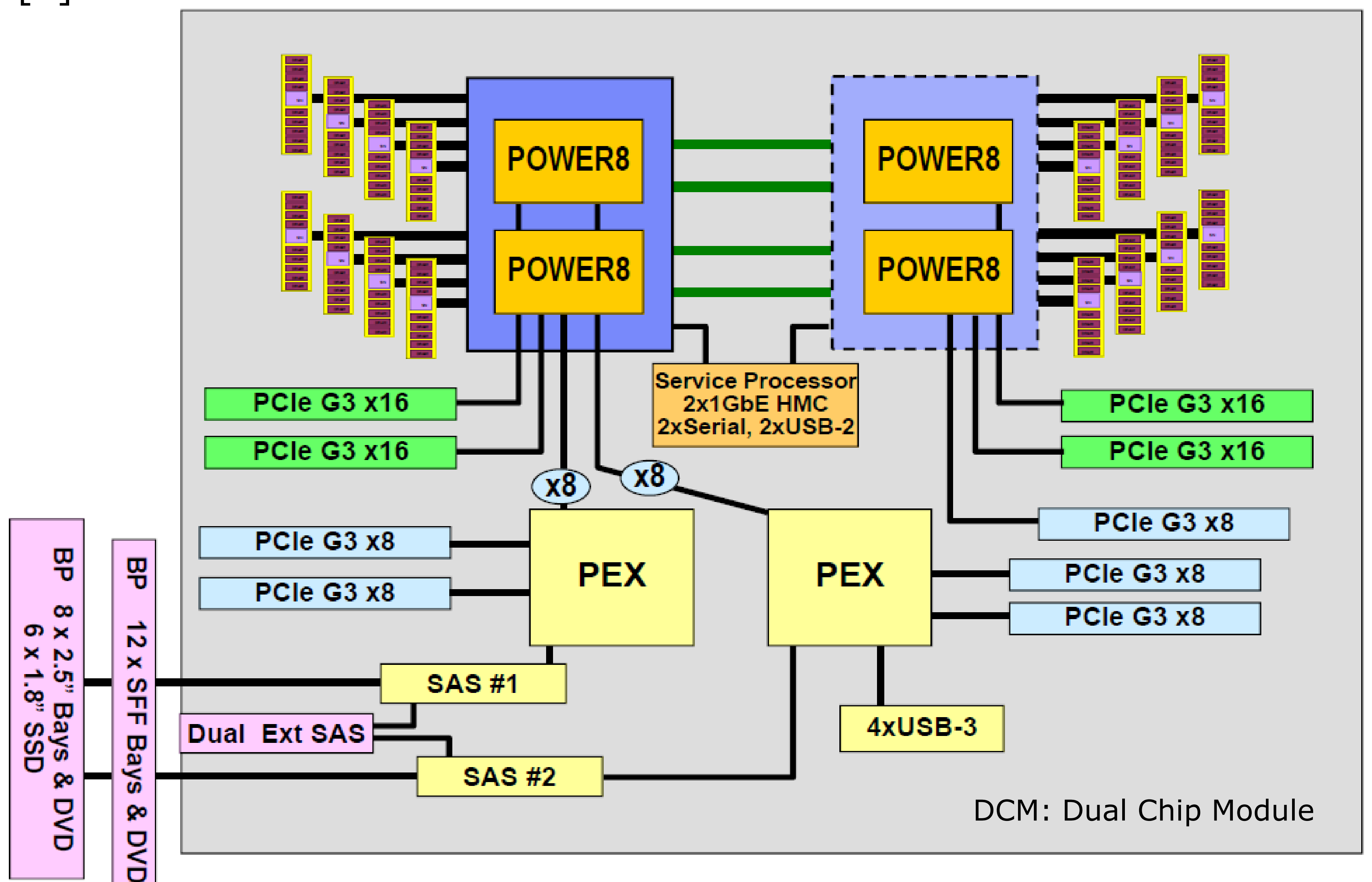
IBM Flex System p270

IBM Flex System p260

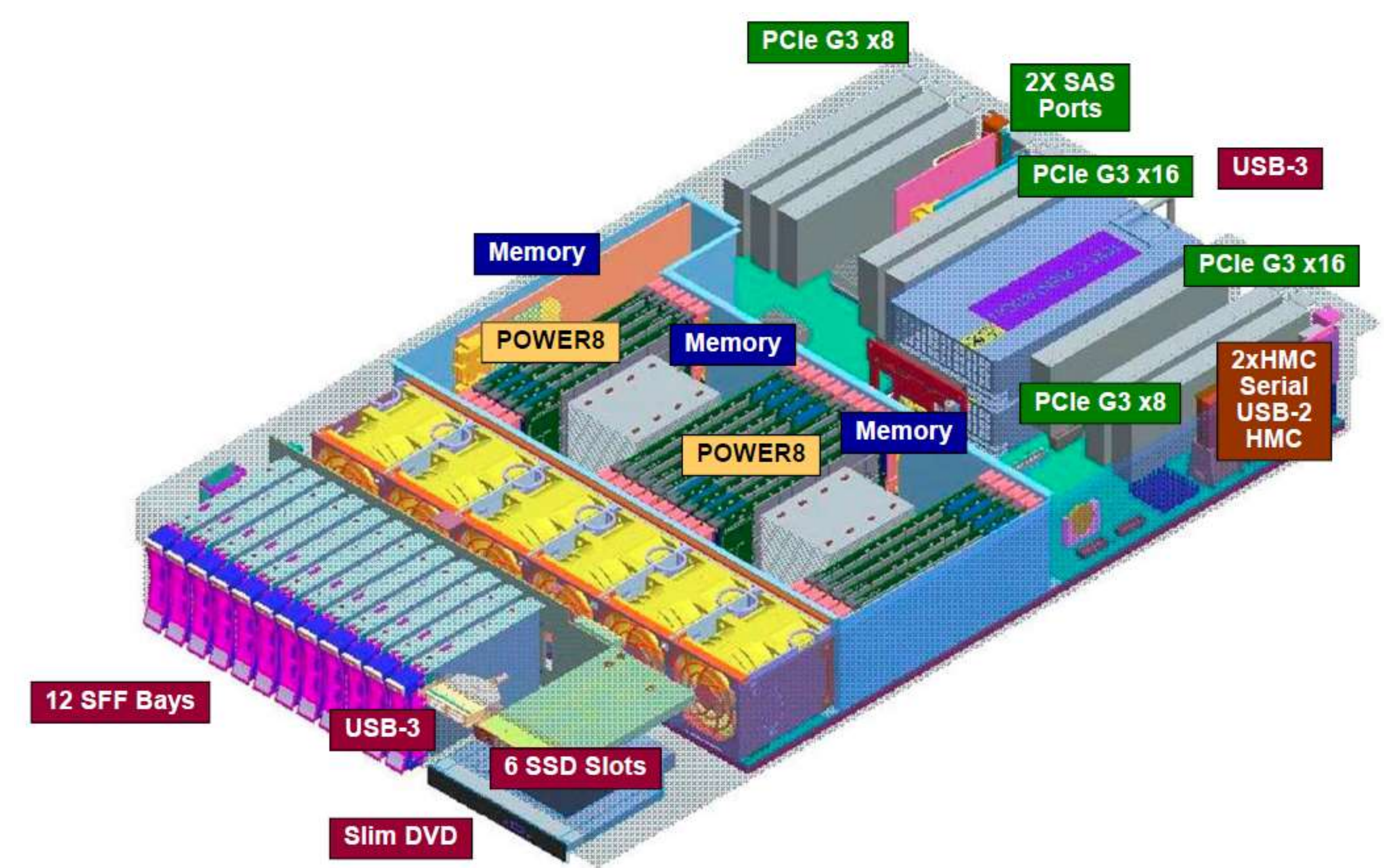
Naming conventions of POWER8 servers [3]



Example: Block diagram of a POWER8 based 2S2U server built up of 2 DCMs [3]



Example: Layout of the POWER8 based 2S2U server built up of 2 DCMs [3]



Opening up the IP (Intellectual Property) of the POWER architecture [79], [80]

- In 8/2013 IBM announced the formation of the OpenPOWER Consortium. Grounding members are: IBM, Google, Mellanox, NVIDIA, Tyan.
- This move made POWER IP licensable to other vendors allowing and fostering the open development of POWER hardware and software.
- The basic idea is that IBM will control the POWER instruction set, much as ARM Holdings does with the ARM instruction set allowing other firms to innovate even at the instruction set level.
- While the OpenPower Consortium is not restricting the licensing to any particular POWER chip, in fact licensing is focused on the Power8 and subsequent chips.
- Obviously, with establishing the OpenPOWER Consortium IBM intends to slow down or even reverse the declining use of the POWER architecture as seen in the past decade.
- The OpenPOWER Consortium is open to any firm that wants to innovate on the POWER platform and participate in an open, collaborative effort.

Remarks

- Following ARM's very successful IP licensing model many dominant processor vendors intend to open up their IP for licensing in order to promote spreading of their platforms, like IBM, but also NVIDIA and also Intel:

- In 06/2013 NVIDIA announced that it would begin licensing its Kepler GPU architecture to 3rd parties [81].
- In 09/2013 also Intel disclosed that they may license their new low power Quark SoC designs to be fabbed by other companies [82].

10.2 Main enhancements of the POWER8

10.2 Main enhancements of the POWER8

The efficient implementation of a 12-core 8-way SMT design requires in the first line

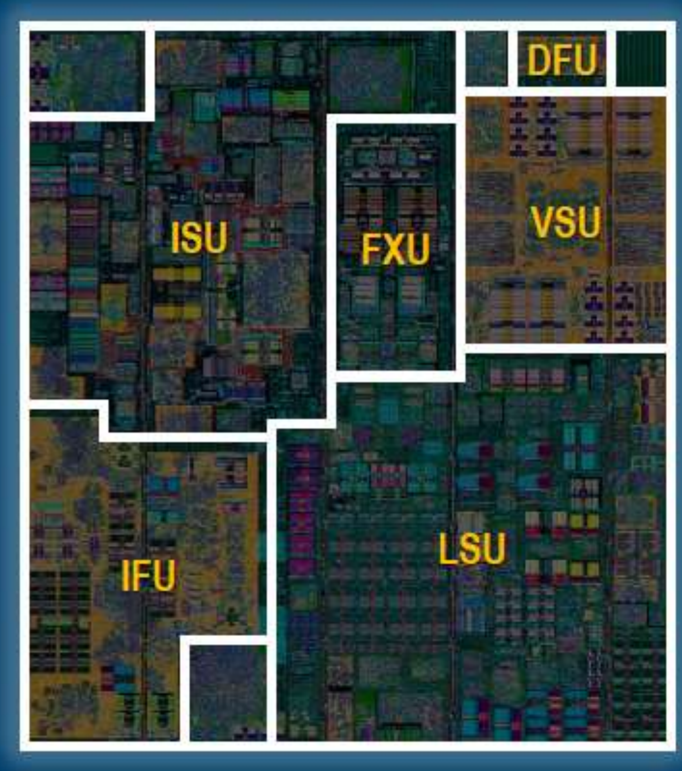
- more execution resources in the microarchitecture,
- a more efficient cache system and
- higher memory bandwidth,

as indicated in the subsequent discussion.

Main enhancements of the execution, cache and memory subsystems of POWER8 cores [83]

Execution Improvement vs. POWER7

- SMT4 → SMT8
- 8 dispatch
- 10 issue
- 16 execution pipes:
  - 2 FXU, 2 LSU, 2 LU, 4 FPU, 2 VMX, 1 Crypto, 1 DFU, 1 CR, 1 BR
- Larger Issue queues (4 x 16-entry)
- Larger global completion, Load/Store reorder
- Improved branch prediction
- Improved unaligned storage access



Larger Caching Structures vs. POWER7

- 2x L1 data cache (64 KB)
- 2x outstanding data cache misses
- 4x translation Cache

Wider Load/Store

- 32B → 64B L2 to L1 data bus
- 2x data cache to execution dataflow

Enhanced Prefetch

- Instruction speculation awareness
- Data prefetch depth awareness
- Adaptive bandwidth awareness
- Topology awareness

Core Performance vs . POWER7

~1.6x Single Thread

~2x Max SMT

10.2 Main enhancements of the POWER8 (3)

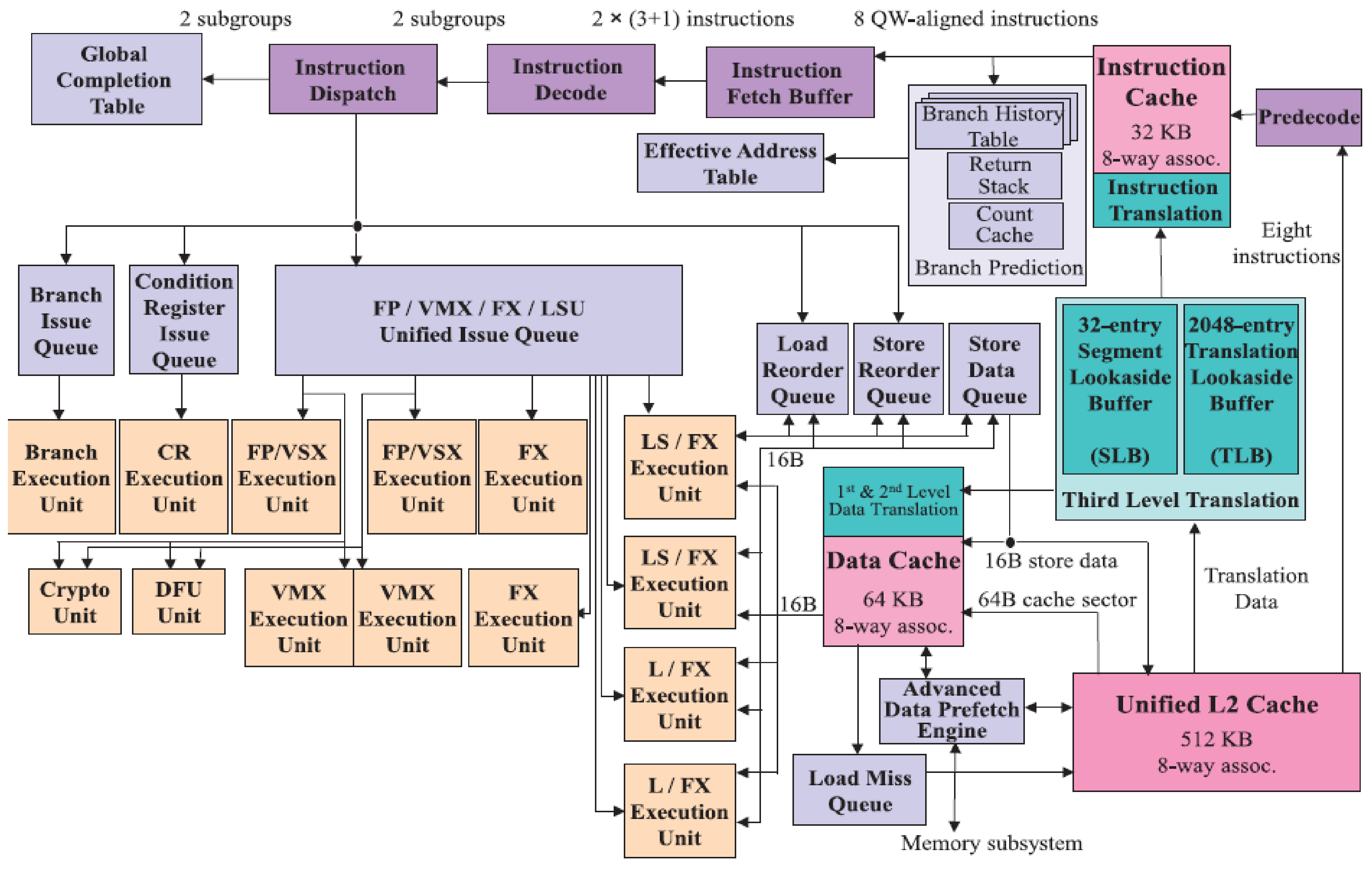
Quantitative enhancements of the microarchitecture and cache system of the POWER8 vs. the POWER7

Enhanced feature	POWER7	POWER8
No. of cores	8	Up to 12 cores (with 8-core and 6-core alternatives)
SMT	4-way SMT (SMT4)	8-way SMT (SMT8)
Width of the front-end	6-wide	8-wide
Issue rate	8	10
No. of execution units	12	16
Type of execution units	2 FXU, 2 LSU, 4 FPU, 1 VMX, 1 DFU, 1 CR, 1 BR	2 FXU, 2 LSU, 4 FPU, 2 VMX, 1 DFU, 1 CR, 1 BR, 2 LU (Logic Unit), 1 Crypto
L1 data cache	32 KB L1 data cache/core	64 KB L1 data cache/core
L2 cache	256 KB/core	512 KB L2/core
L3 (on-chip)	8 x 4 MB partially shared on-chip eDRAM	12 x 8 MB partially shared on-chip eDRAM
L4 (off-chip)	-	Up to 8x16 MB eDRAM in the memory chips

Evolution of the execution resources in the POWER8 cores

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-ways
Width of the front-end	5	5	5	6	8	12
Dispatch rate	5	5	(In-order design)	6	8	12
Issue rate	8	8	7	8	10	16
No. of execution units per-core	8	8	9	12	16	20
No/type of execution units per-core	2 FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP 1BR/CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 1VMX, 1DFU	2FX, 2LS, 4FP, 1BR, 1CR, 2VMX, 1DFU, 2LU, 1 Crypto	8AGEN, 4VSU(128) 4LS(128) 2BRU DFU Cripto

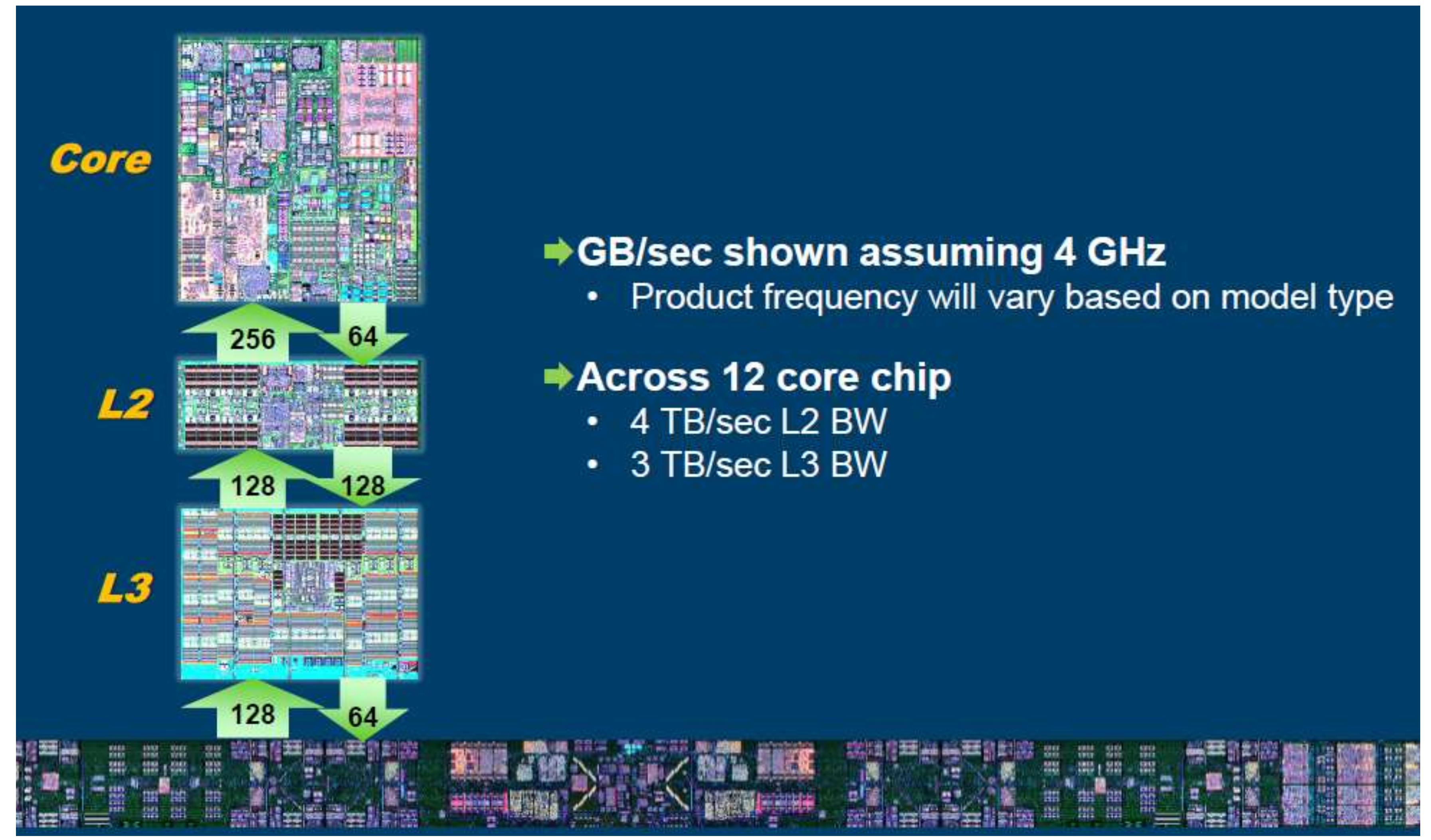
Block diagram of a POWER8 core [84]



Evolution of the cache architecture in subsequent POWER models

	POWER4 (2001)	POWER5 (2004)	POWER6 (2007)	POWER7 (2010)	POWER8 (2014)	POWER9 (2017)
No. of cores	2	2	2	8	12	24
SMT	No	2-way	2-way	4-way	8-way	4/8-way
Typ. clock frequency	1.3 GHz	1.65- 1.90 GHz	4-5 GHz	3-4 GHz	4-4.35 GHz	3.5-4.0 GHz
L1 instr. cache per-core	64 kB	64 kB	64 kB	32 kB	32 kB	32 kB
L1 data cache per-core	32 kB	32 kB	64 kB	32 kB	64 kB	32 kB
L2 cache	1.44 MB shared	1.92 MB shared	4 MB per-core	256 KB per-core	512 KB per-core	512 KB per-core
L3 cache	32 MB shared off-chip dir. on-chip	36 MB shared off-chip	32 MB shared off-chip	8x4 MB tightly coupled on-chip eDRAM victim cache	12x8 MB tightly coupled on-chip eDRAM victim cache	12x10 MB Per 2 cores
L4 cache	--	--	--	--	up to 8x16 MB eDRAM on 8 mem.buffers (not snooped)	up to 8x16 MB eDRAM on 8 mem.buffers (not snooped)

Bandwidth values relating the cache architecture of the POWER8 [83]



Evolution of memory features in IBM's POWER line

Model	Tech.	Intro.	No. of cores (up to)	fc (up to)	SMT	DIMM type	DRAM speed	No/speed/width of MC-MB links (up to)	MC-MB link limited BW/proc. (up to)	BW/fc/ /core (byte/cycle) (up to)
POWER7	45 nm	2010	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gbit/s 2B R/1B W	76.8 GB/s	2.2
						Propr. FB-DIMM	DDR3-1066	8@6.4 Gb/s 2B R/3B W	153.6 GB/s	4.4
POWER7+	32 nm	2013	8	4.42 GHz	4-way	Commod. DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	153.6 GB/s	3.9
						Propr. FB-DIMM	DDR3-1066	4@6.4 Gb/s 2B R/1B W	76.8 GB/s	2.2
POWER8	22 nm	2014	12	4.35 GHz	8-way	Propr. CDIMM	DDR3-1600	2(8)[1]@9.6 Gbit/s 2B R/1B W	57.5 (230 GB/s)	1.1 (4.4)
POWER9 (Scale-Out)	14 nm	2017	12 24	4.00 GHz	8-way 4-way	Commod. DIMM	DDR4-2666	--	--	--
POWER9 (Scale-Up)		2018	12	4.00 GHz	8-way	Commod. DIMM	DDR4-1600	8@9.6 Gbit/s 2B R/1B W	230 GB/s	4.79

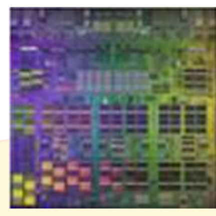
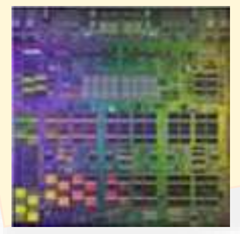
[1]: According to IBM's literature [] the POWER8 has up to eight memory channels.
Nevertheless, first servers delivered until 05/2015 makes use of only two of them.

Commod.: Commodity Prop.: Proprietary MB: Memory buffer (POWER4-7: SMI buff.-POWER8-9: Centaur buff-)

10.3 Key innovations of the POWER8

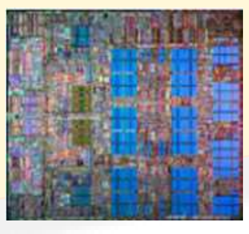
- 10.3.1 12-core design
- 10.3.2 8-way SMT (SMT)
- 10.3.3 Resonant clocking
- 10.3.4 Hardware transactional memory
- 10.3.5 Intelligent memory buffers
- 10.3.6 CAPI
- 10.3.7 Replacing the GX I/O bus by the PCIe Gen3 bus
- 10.3.9 On-chip microcontroller for power management
- 10.3.10 Integrated voltage regulator modules on the chiplets
- 10.3.11 Using charge pumps for per-core power gating

10.3 Key innovations of the POWER8 (Die photos from [3])

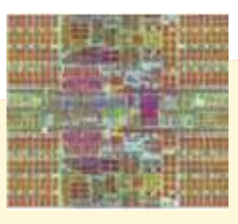
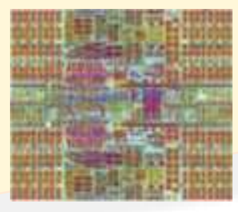
POWER4/4+
180/130 nm

2 cores
Inst. grouping
Shared L2
Off-chip L3
Serial P2P mem. buses
with SMI chips
GX I/O bus
Support for SMP



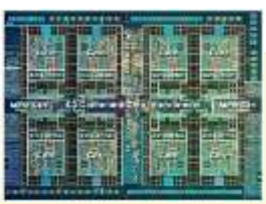
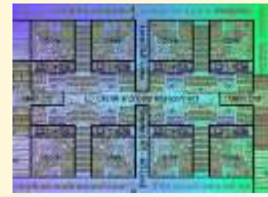
POWER5/5+
130/90 nm

2-way SMT
Integrated MC
Fine grained
clock gating

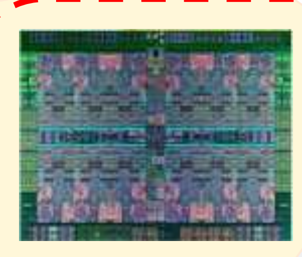
POWER6/6+
65/65 nm

Private L2
Dual MC
FB-DIMM option
AltiVec SIMD
Hardware DFP
EnergyScale with
Critical Path
Monitors
Nap idle mode

POWER7/7+
45/32 nm

8 cores
4-way SMT
On-chip L3
Ring bus interconnect
Energy Scale 2 with
Per core fc
Dyn. fan managm.
Sleep idle mode
*Accelerators for
cryptography
*Winkle idle mode
*POWER7+



POWER8
22 nm

12 cores
8-way SMT
Resonant clocking
Hardware TM
Intelligent mem. buffers
with distributed L4
no FB-DIMM option
CAPI
Replacing GX by PCIe G3
On-chip µc for PM
Per-core Vdd
Per-core VRMs

2001 | 2004 | 2007 | 2010 | 2014

10.3.1 12-core design

10.3.1 12-core design [83]

Technology

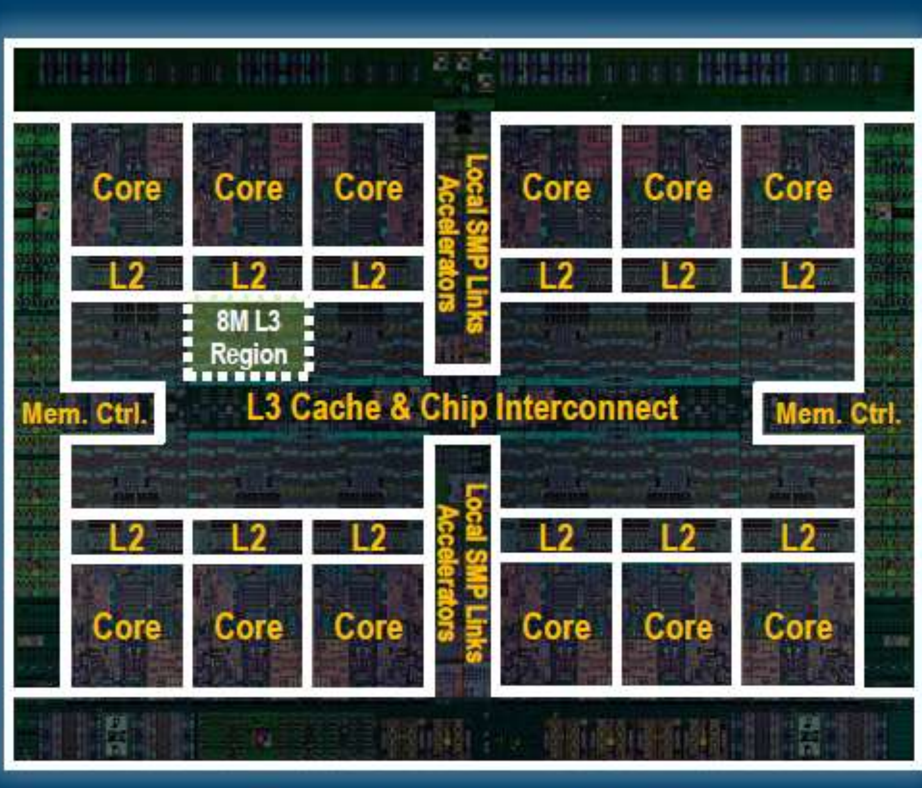
- 22nm SOI, eDRAM, 15 ML 650mm2

Cores

- 12 cores (SMT8)
- 8 dispatch, 10 issue, 16 exec pipe
- 2X internal data flows/queues
- Enhanced prefetching
- 64K data cache, 32K instruction cache

Accelerators

- Crypto & memory expansion
- Transactional Memory
- VMM assist
- Data Move / VM Mobility



Energy Management

- On-chip Power Management Micro-controller
- Integrated Per-core VRM
- Critical Path Monitors

Caches

- 512 KB SRAM L2 / core
- 96 MB eDRAM shared L3
- Up to 128 MB eDRAM L4 (off-chip)

Memory

- Up to 230 GB/s sustained bandwidth

Bus Interfaces

- Durable open memory attach interface
- Integrated PCIe Gen3
- SMP Interconnect
- CAPI (Coherent Accelerator Processor Interface)

10.3.2 8-way SMT (SMT8)

10.3.2 8-way SMT (SMT8)

- Many business applications with inherent parallelism, such as transaction systems or business analytics, can be speeded up if more hardware thread parallelism is available.

  To accommodate this, POWER8 doubled thread parallelism from 4 to 8, referred to as SMT8.
- To support twice as much threads the POWER8 doubled the sizes of the L1 data, L2 and L3 caches.
- It was a design requirement that at each multithreading level (ST, SMT2, SMT4 and SMT8) thread performance of a POWER8 core should be better than on a POWER7 core.
- The core can dynamically change the multithreading mode among ST, SMT2, SMT4 and SMT8 depending on the number of active threads.

10.3.3 Resonant clocking

10.3.3 Resonant clocking

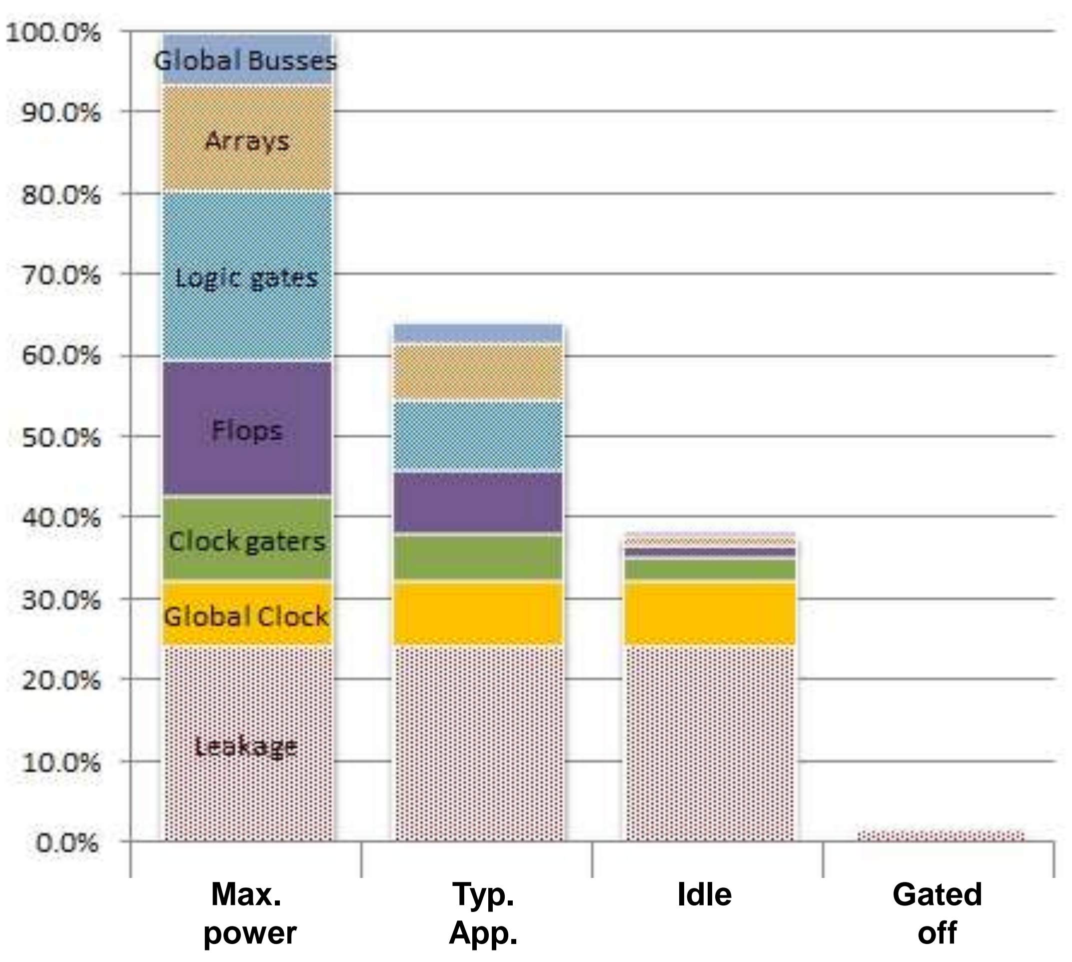
The motivation for introducing resonant clock meshes-1

Clock distribution networks (CDN) consumes a large portion of chip power ranging from 30 % to 70 % of the total chip power, mainly due to

- the ever increasing number of transistors to be clocked and
- the moderately increasing clock frequencies,

as pointed out e.g. in [85].

Example: Distribution of power consumption in a Bulldozer processor [86]



10.3.3 Resonant clocking (3)

The motivation to introduce resonant clock meshes-2

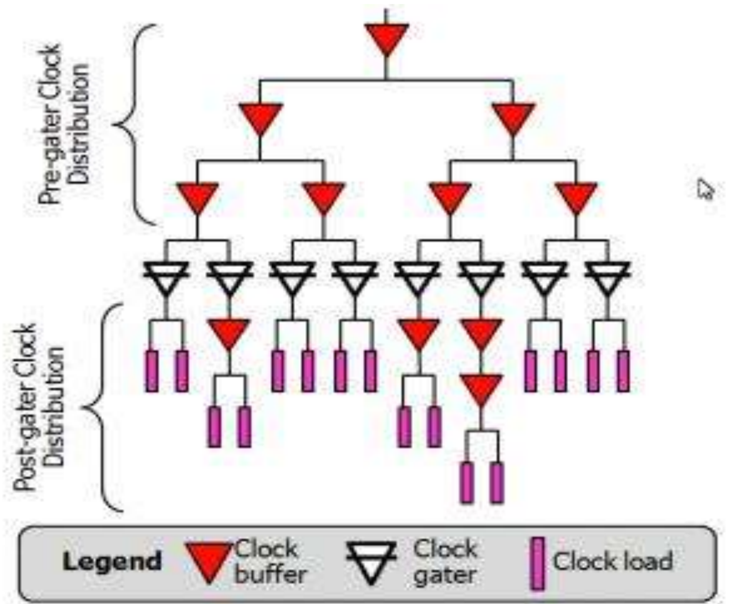
- Accordingly, CDNs became an important topic of research and development aiming at reducing power consumption and also propagation delay.
- Without going into details next we give an overview of the main steps of the evolution of CDNs.

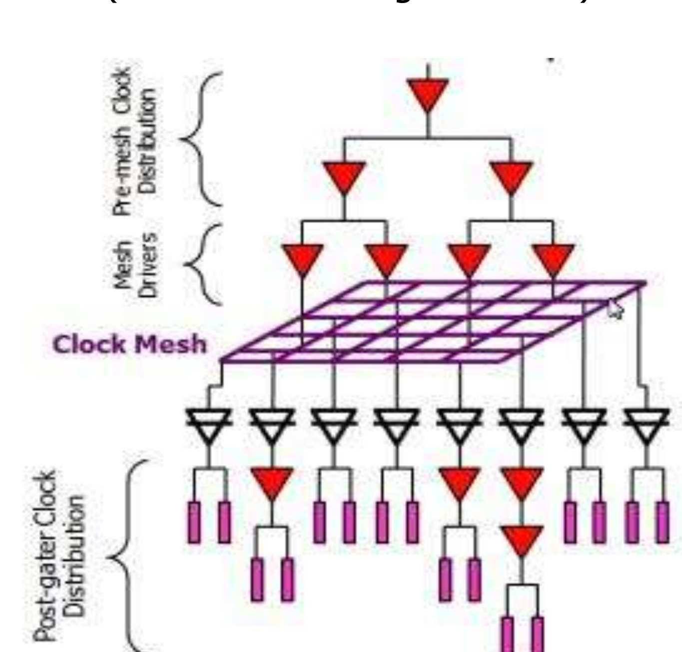
Main types of clock distribution networks [87]

Clock distribution

Tree-based clock distribution

Mesh-based clock distribution
(Termed also as grid-based)



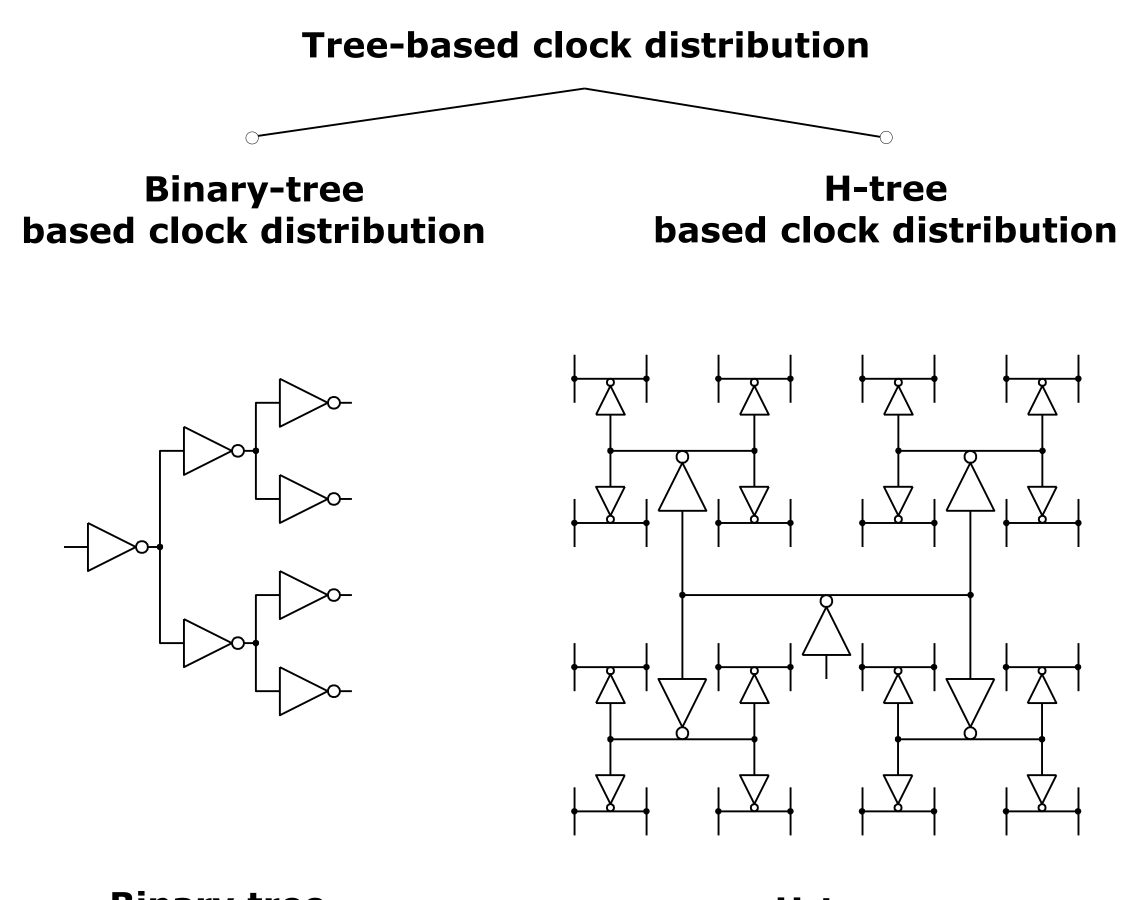


(The grid is actually a low-resistance metal grid fed by tree-based clock driving)

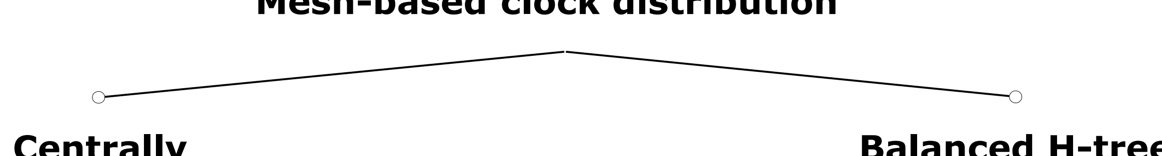
Remark

Both figures of clock distribution networks include clock gating to be discussed later.

Main types of tree-based clock distribution networks [88]



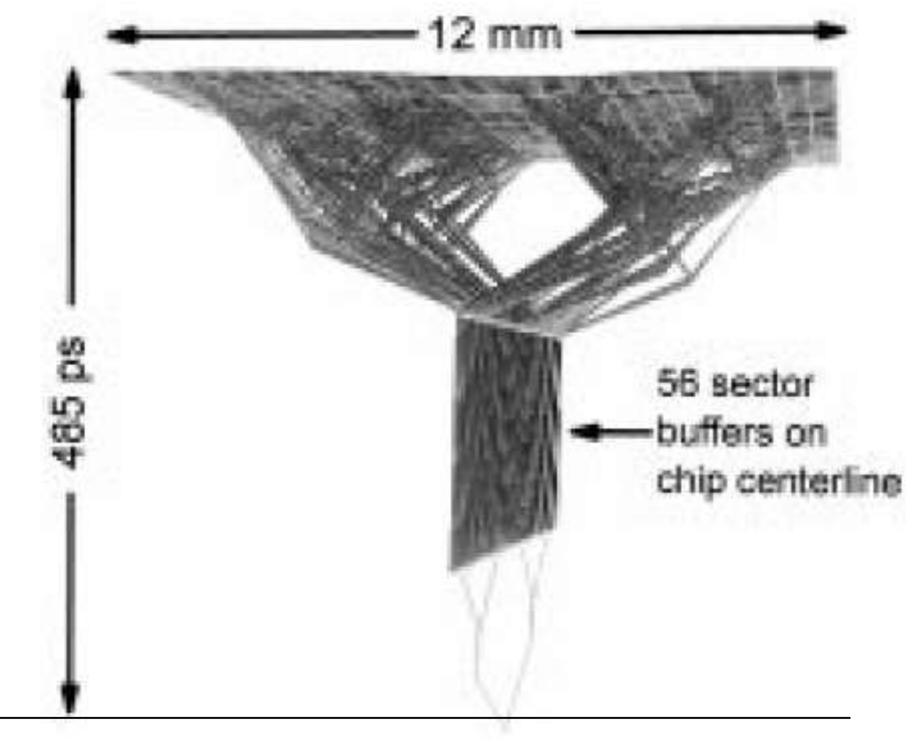
Main types of mesh-based clock distribution networks [89]



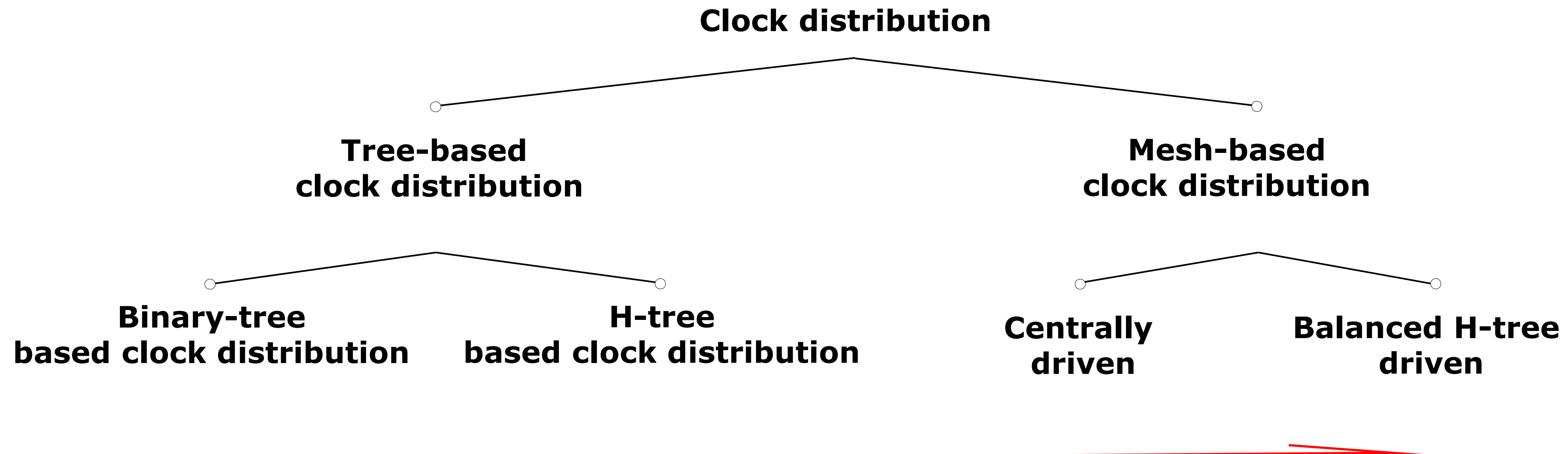
Early mesh-based clock distribution networks were centrally driven

Most modern mesh-based distribution networks use balanced H-trees to drive the mesh

Example: Illustration of an experimental mesh-based clock distribution network with H-tree driving [89]



Main steps of the evolution of clock distribution networks



Reduction of the power consumption of CDNs

Key techniques to reduce power consumption of CDNs

Clock gating

Currently not used clock buffers are switched off by gating the clock buffers

Resonant clocking

Through including inductors into the mesh resonant clock distribution networks are built up. These LC networks need less power to sustain the clock signal distribution than previous non-resonant networks by recycling the energy through transforming it between electrical and magnetic forms [85]

Principle of clock gating

- Clock gating is widely used to reduce power consumption by switching off clocking of temporarily not used circuits of the processor.
- Clock gaters are on/off switches on the clock buffers that are implemented simply by an AND function that enables or disables the clock buffer, as indicated below.
- As an example, Intel's Pentium 4 (2000) utilized already aggressive clock gating, whereas AMD made use of this technique later, presumable beginning with their K8 family (2003).

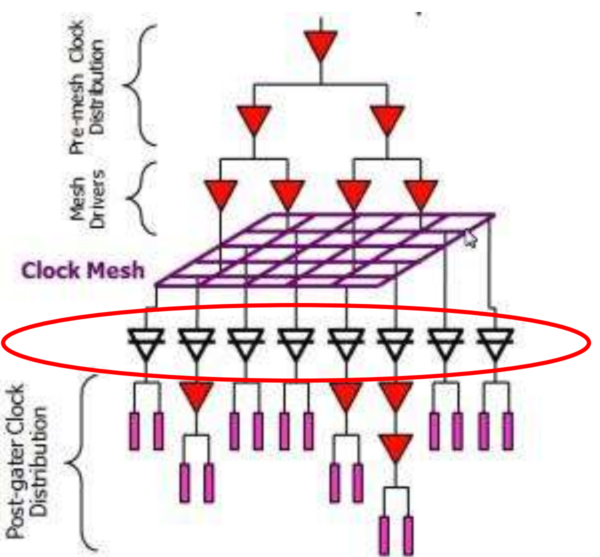


Figure: Use of clock gating to switch off temporarily not used circuits in a mesh-based clock distribution network [87]

Principle of resonant clocking-1

- Although resonant clocking may be implemented in many different ways, below we focus on the recently most often utilized alternative of resonant clocking, called resonant clock meshes.
- The basic idea is that the clock mesh represents a high input capacitance for the clock buffers that can be extended to an LC circuit (aka LC tank) by inserting an inductance into the clock distribution network, as indicated in the next Figure.

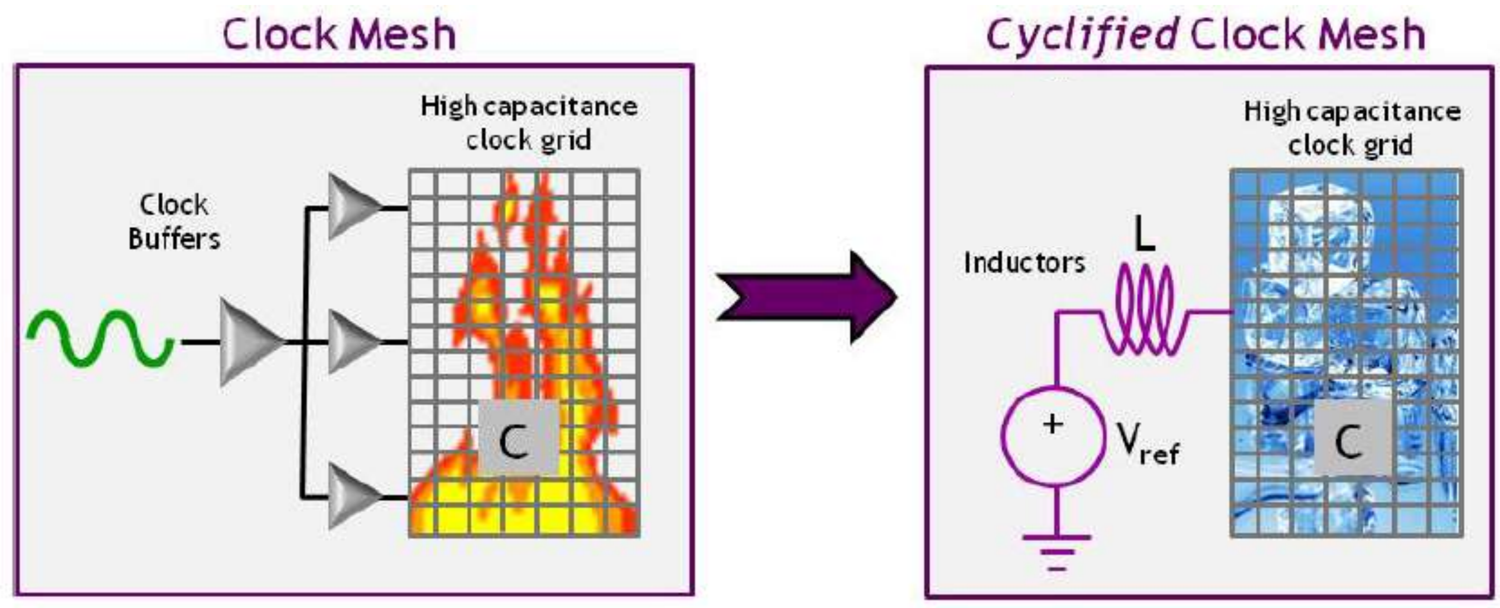


Figure: Insertion of an inductance into the clock mesh to form an LC circuit [125]

The resonant LC circuit needs less energy to sustain clock signals than before.

Principle of resonant clocking-2

- In the ideal case an LC circuit (LC tank) will be built, as indicated below.

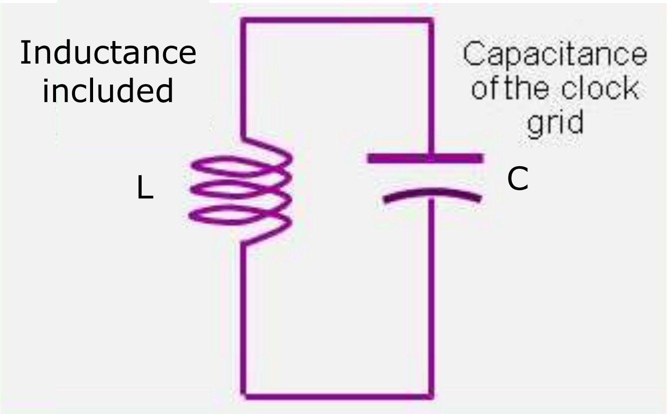


Figure: The LC circuit formed by the capacitance of the clock mesh (grid) (C) and the inductance (L) inserted into the clock distribution network [125]

- The LC circuit has a resonant frequency (fresonant) as follows:

$$f_{resonant} = \frac{1}{2\pi\sqrt{LC}}$$

Principle of resonant clocking-3

In the ideal case the LC circuit does not need any additional energy to oscillate, as indicated below.

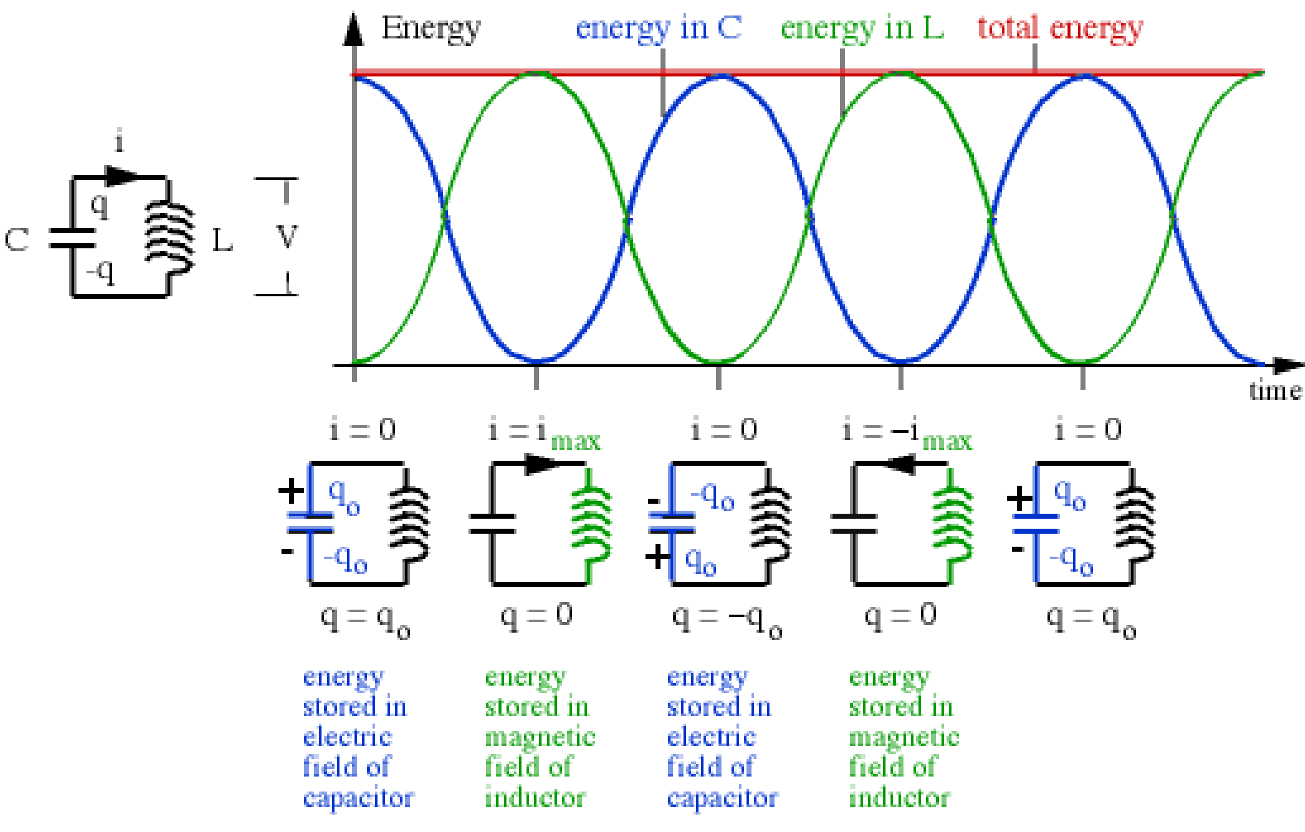


Figure: Principle of operation of an LC circuit [90]

10.3.3 Resonant clocking (14)

Principle of resonant clocking-4

- In the real case the LC circuit can recycle a part of the energy by transforming it between its electrical and magnetic forms, as indicated before.

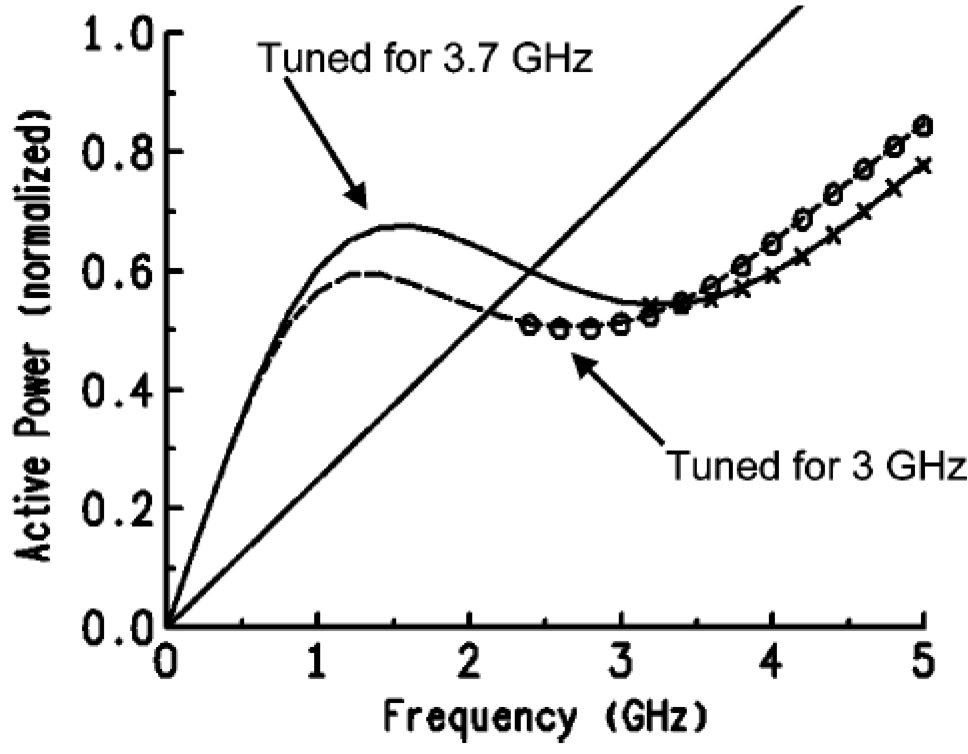
  In this approach at the resonant frequency the CDN needs only a part of the input power to sustain the oscillation, íe. to sustain the full swing clock signal, this results in a reduced overall power consumption vs. the non-resonant implementation.

- Another implementation issue relates to the scaling of the clock frequency.

  The clock frequency of recent processors is typically scaled by DVFS in a wide range (e.g. from 4 GHz to 2 GHz) to reduce power in case of less demanding workloads, in contrast, resonant clock distribution has a given resonant frequency.

  Accordingly, resonant clocking is less efficient for clock frequencies deviating from the resonant frequency of the LC circuit, the more the more the clock clock frequency differs from the resonant frequency of the resonant clock distribution network, as indicated in the next Figure.

Active power consumption of the resonant clock grid while tuned for 3.0 and 3.7 GHz [91]



10.3.3 Resonant clocking (16)

Overview of the implementation of resonant clock meshes (RCMs)

- RCMs was first proposed about 2003 by researchers at Stanford University [126] and IBM [127].
- First implementations in commercial processors followed a couple of years later, as overviewed below.
  - 2008 IBM: Experimental enhancement of the Cell processor for RCM
  - 2009 Cyclone: first experimental implementation on the ARM 926EJ-S and issuing patent applications
  - 2012 AMD: first commercial implementation in the Piledriver based Trinity A10).

    There were two switchable operating modes:
    - a resonant mode and
    - a direct mode.
  - 2014 IBM: POWER8 dual resonant modes design with on-the-fly mode changing
  - 2015 AMD: Excavator, it uses an improved implementation.

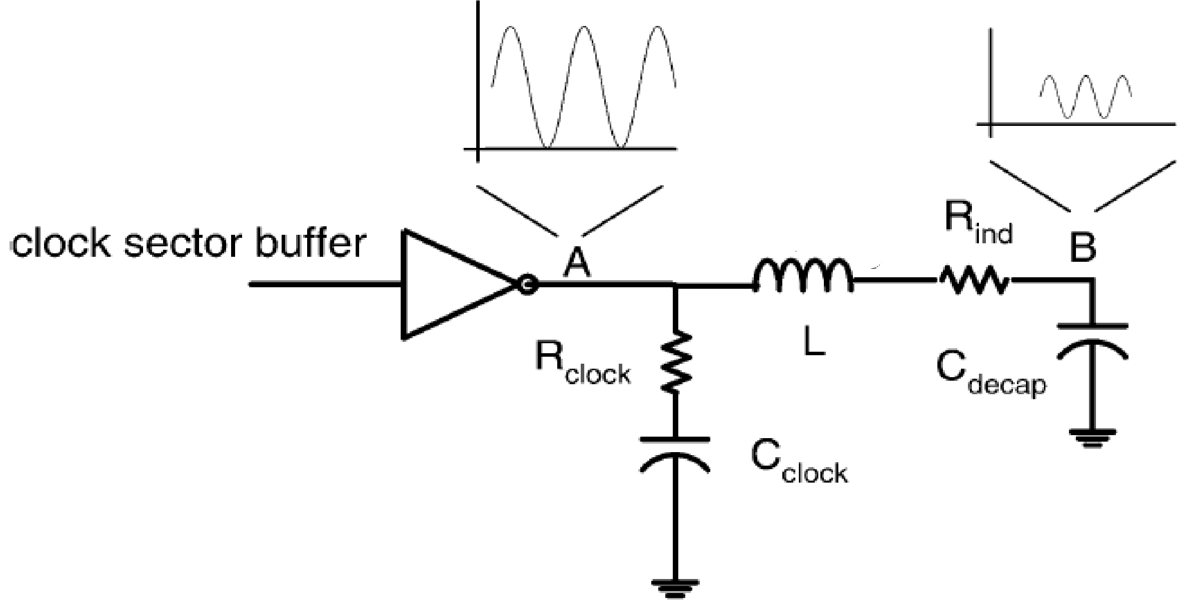
Remarks

- In the literature no reference was found whether AMD continued to use resonant clock meshing in their Excavator or Ryzen.
- Also no reference was found about Intel's plans to use RCMs.

Experimental implementation of RCM in a modified Cell BE chip

- The Cell BE processor has three clock meshes serving
  a) the cores and the master
  b) the memory interface and
  c) the I/O interface.
- Mesh a) is clocked at 3.2 GHz and occupies 85 % of the chip area, whereas meshes b) and c) operate at 1.6 GHz.
- Resonant clock distribution is implemented only for mesh a).
- This mesh is subdivided into 830 clock sectors, each driven by a sector clock buffer (implemented as a three-stage inverter).
- The mesh has a total capacitive load of about 2 nF.
- Resonant clocking was achieved by changing one wiring level and adding a new thick copper level for creating inductors and capacitors.

Simplified circuit model of the implemented resonant clock sectors [92]



- The indicated resistances R_{clock} and R_{ind} model inherent values of the physical implementation.
- A large decoupling capacitance C_{decap} connected in series with the inserted inductor is needed to eliminate the static leakage and provide the AC ground.
- C_{decap} needs to be sufficiently large such that the series resonance formed by L and C_{decap} remains much less than the parallel resonance of L and C_{clock}.
- C_{clock} is the capacitive load of the mesh and the sector clock buffers, it amounts to about 2 nF.

Key components of the implementation of a resonant clock sector [91]

- The L inductor is implemented as a spiral wire of 2.75 turns with a value of 1.2 nH.
- The C_{decap} capacitor is implemented as so called metal fringe capacitor of 12 nF, as seen below.

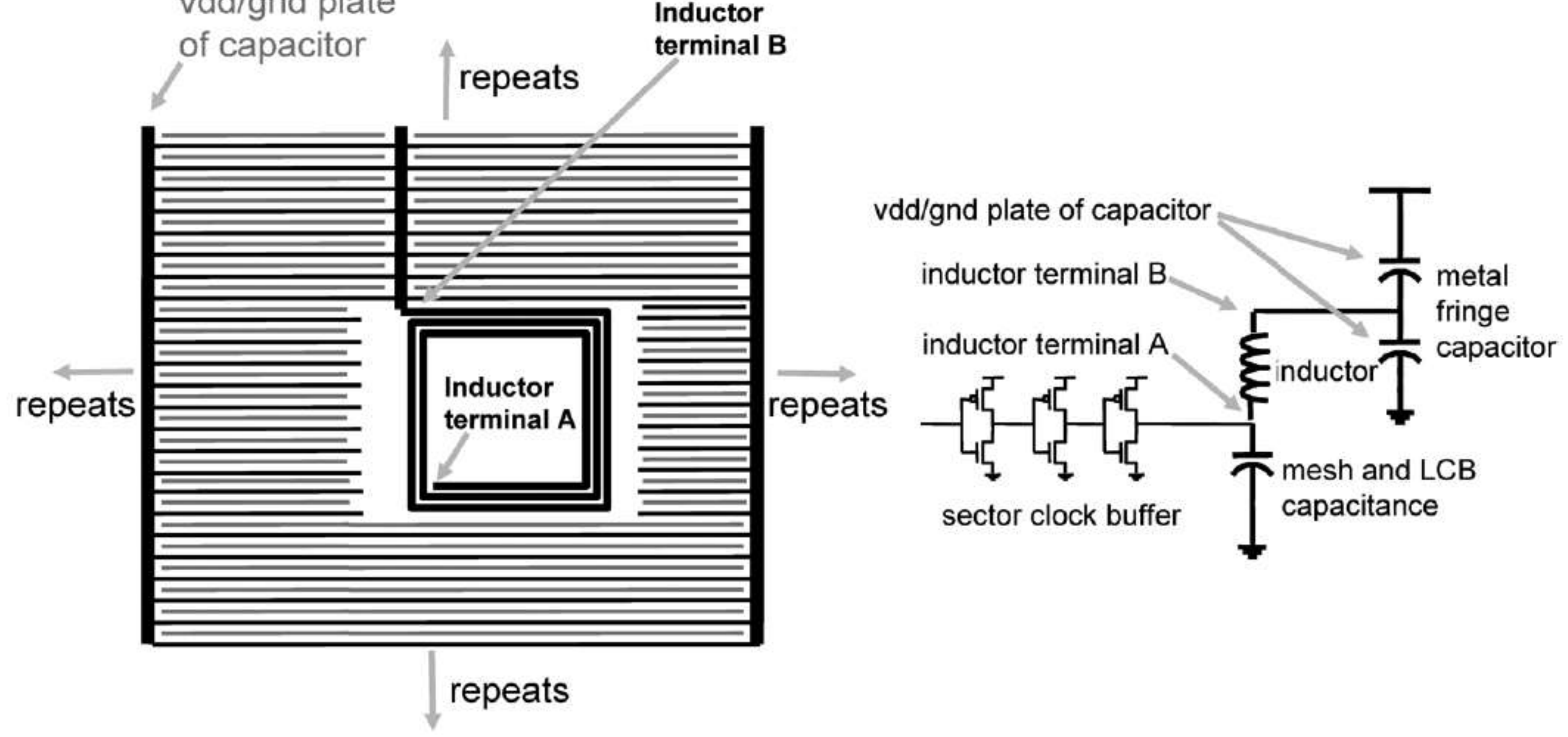
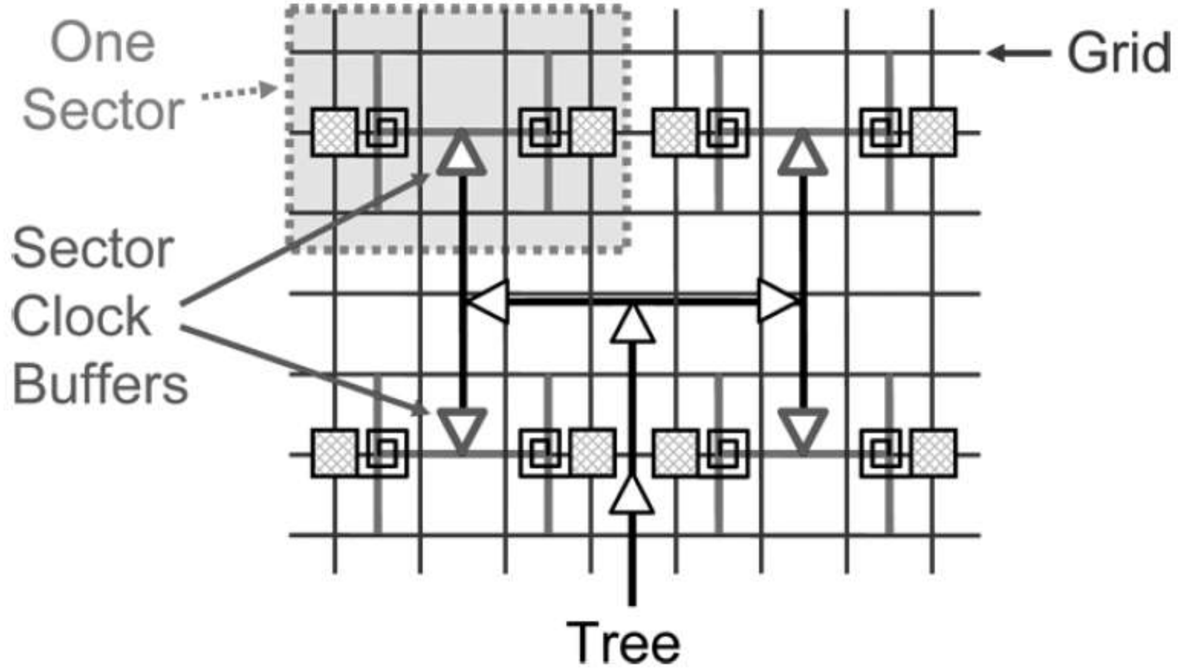


Figure: Key components of the implementation of a resonant clock sector [91]

Actual implementation of a single clock sector [91]



As seen, one resonant clock sector is actually implemented as two symmetrical halves including two inductors and the associated decoupling capacitors.

Achieved power savings-1

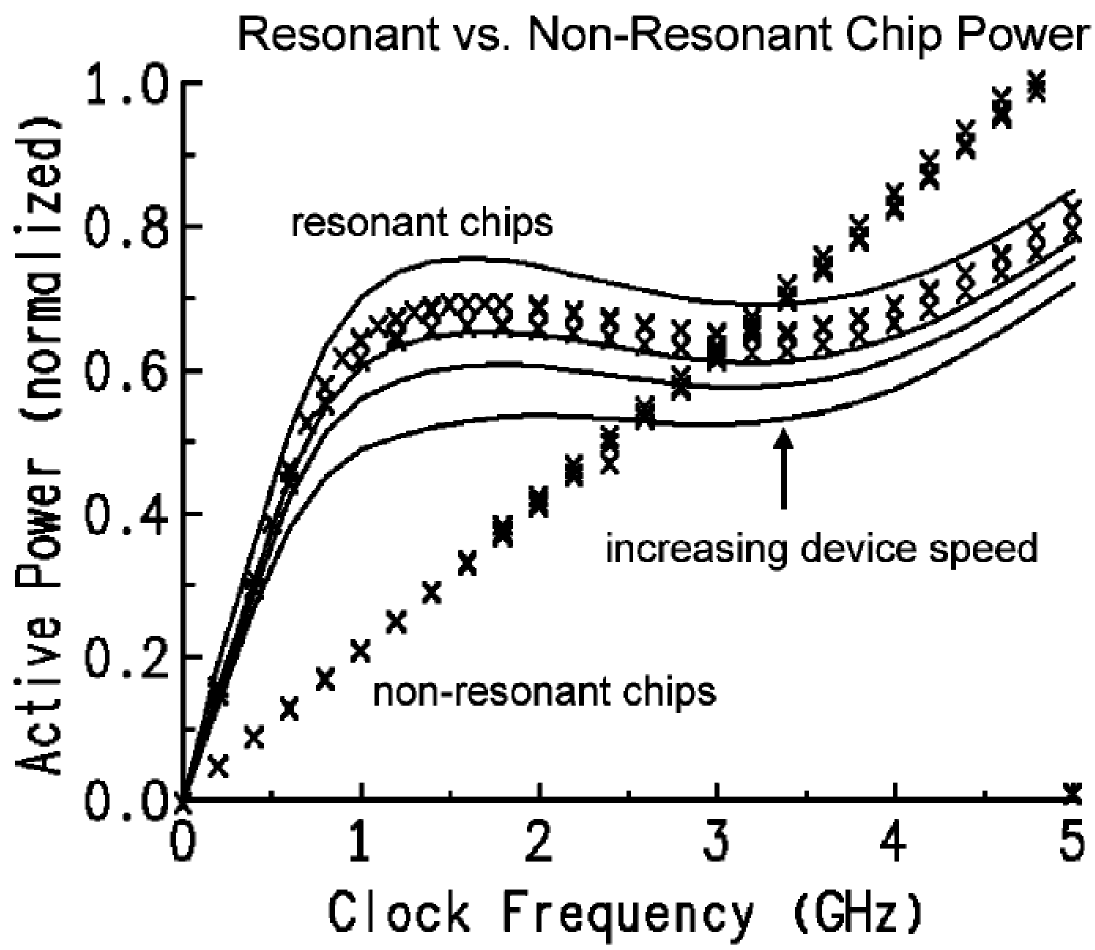


Figure: The Figure shows the measured power consumption (normalized) of the resonant and non resonant chip with only the global clock running and leakage current subtracted [91]

Achieved power savings-2

- As the Figure shows, above 1 GHz the resonant clock mesh achieves the minimum power consumption around the resonance frequency of 3.2 GHz.
- Between 4 and 5 GHz there is a power saving of about 5-25 % compared to the the non-resonant chip, whereas at clock frequencies below 3 GHz the non-resonant chip dissipates significantly less power than the resonant one.
- The results reveal the fact that the implementation of resonant clock meshes is an extremely intricate task including issues like
  - recent processors typically make use of DVFS with a wide range frequency scaling e.g. from 2 to 4 GHz, but resonant clock meshes operate efficiently only around a fixed frequency,
  - the load capacitance of the clock sectors can vary across the chip by as much as 4x, thus the same inductance will result in varying resonant frequencies this lessens efficiency,
  - possible power saving comes more from a possible reduction of the drive strength of the sector buffers than from the energy recirculation between inductors and capacitors.

Remark

Drive strength is the current delivery capability of a device, a device with higher drive strength will charge the input capacitance of the next stage in a shorter time but it consumes more power.

Achieved power savings-3

- All in all, implementing resonant clock meshes seems to be a very difficult engineering task, this can be the reason
  - why AMD discontinued implementing resonant clock meshes in their processors following the Piledriver (Steamroller, Excavator) or
  - why Intel did not implement this technique yet.
- Nevertheless, IBM continued their effort to implement resonant clock meshes and finally introduced this technique in their POWER8 processor in 2014, to be discussed next.

Implementation of RCM in IBM's POWER8 [93]

In the POWER8 the clock meshes of the chiplets (core + L2 cache) are subdivided into 57 clock sectors, as indicated below [93]

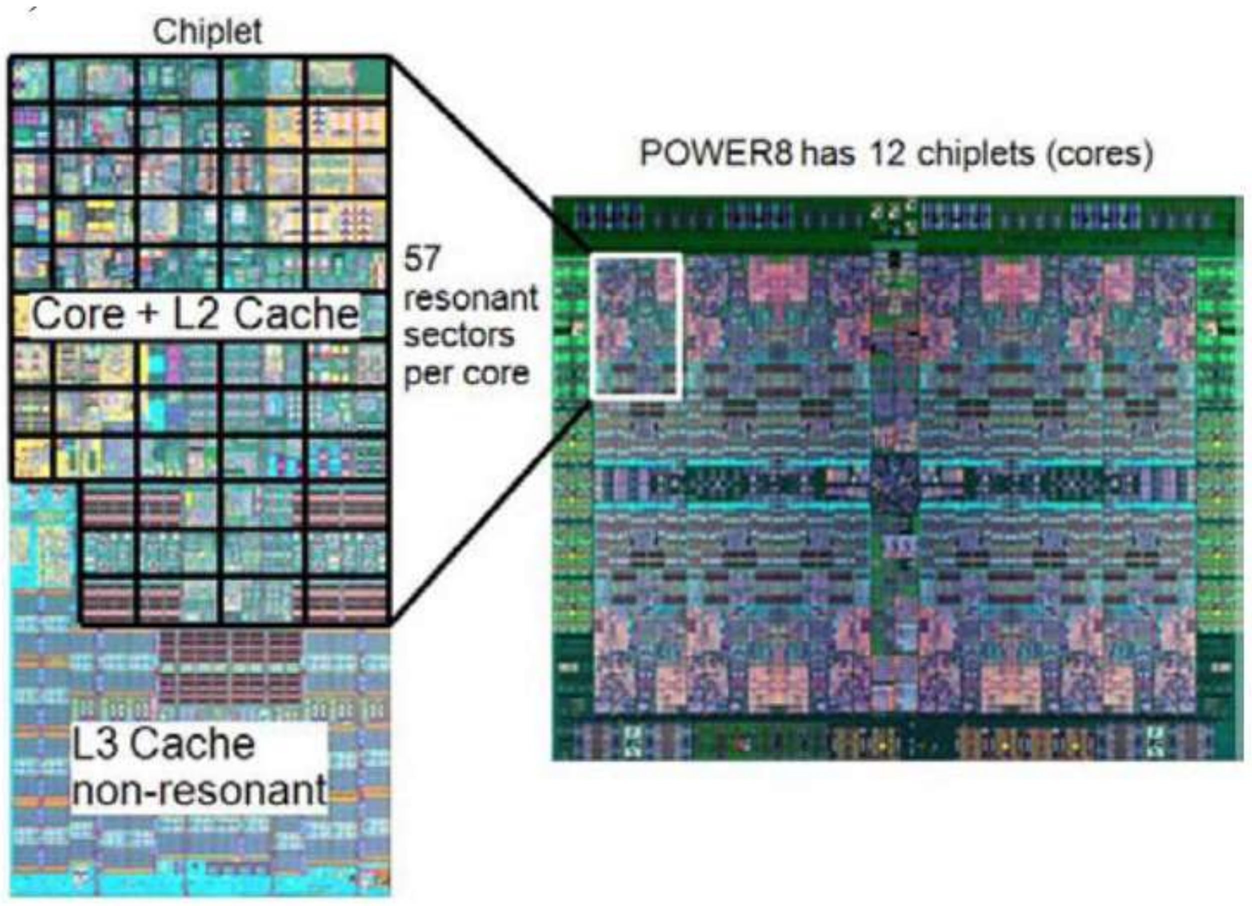
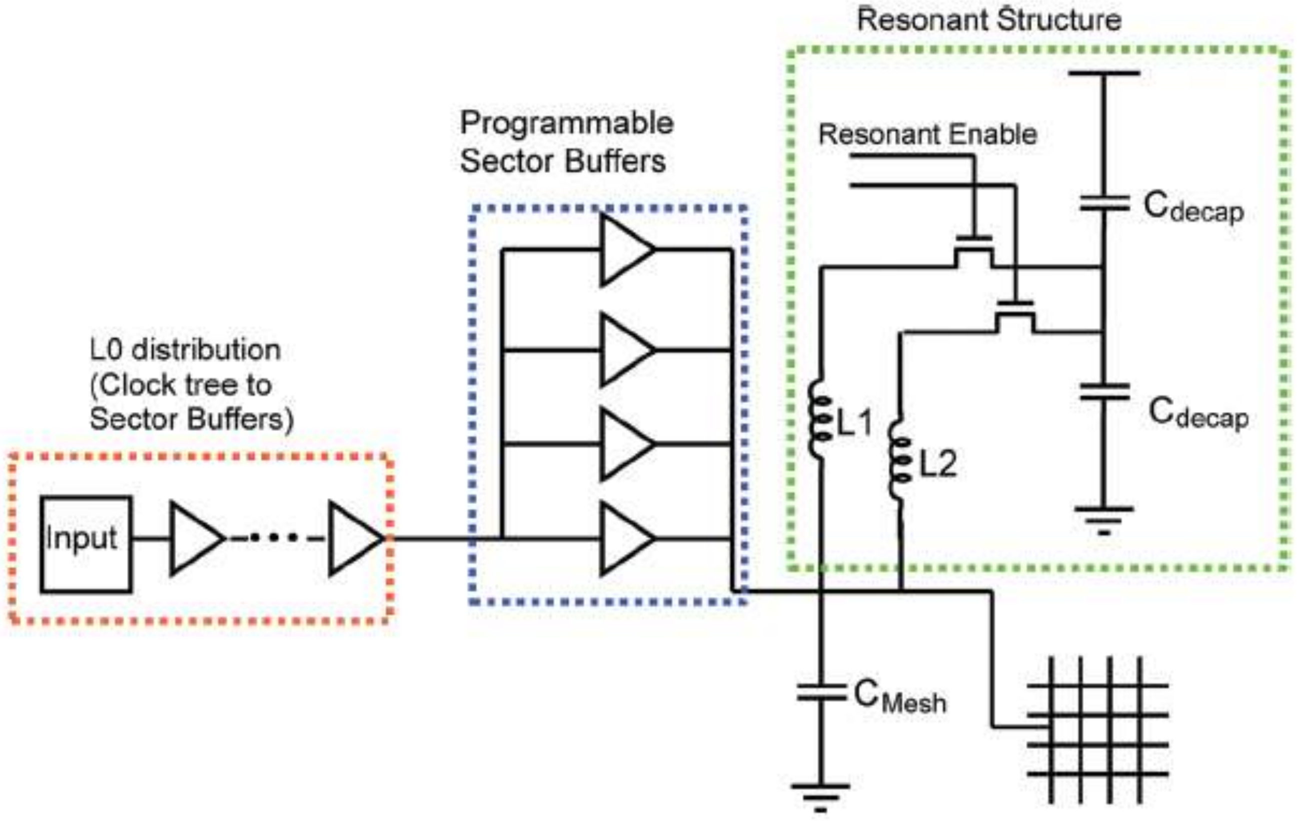


Figure: Subdivision of the clock mesh of each chiplet into 57 resonant clock sectors in the POWER8 [93]

Structure of a resonant clock sector-1 [94]

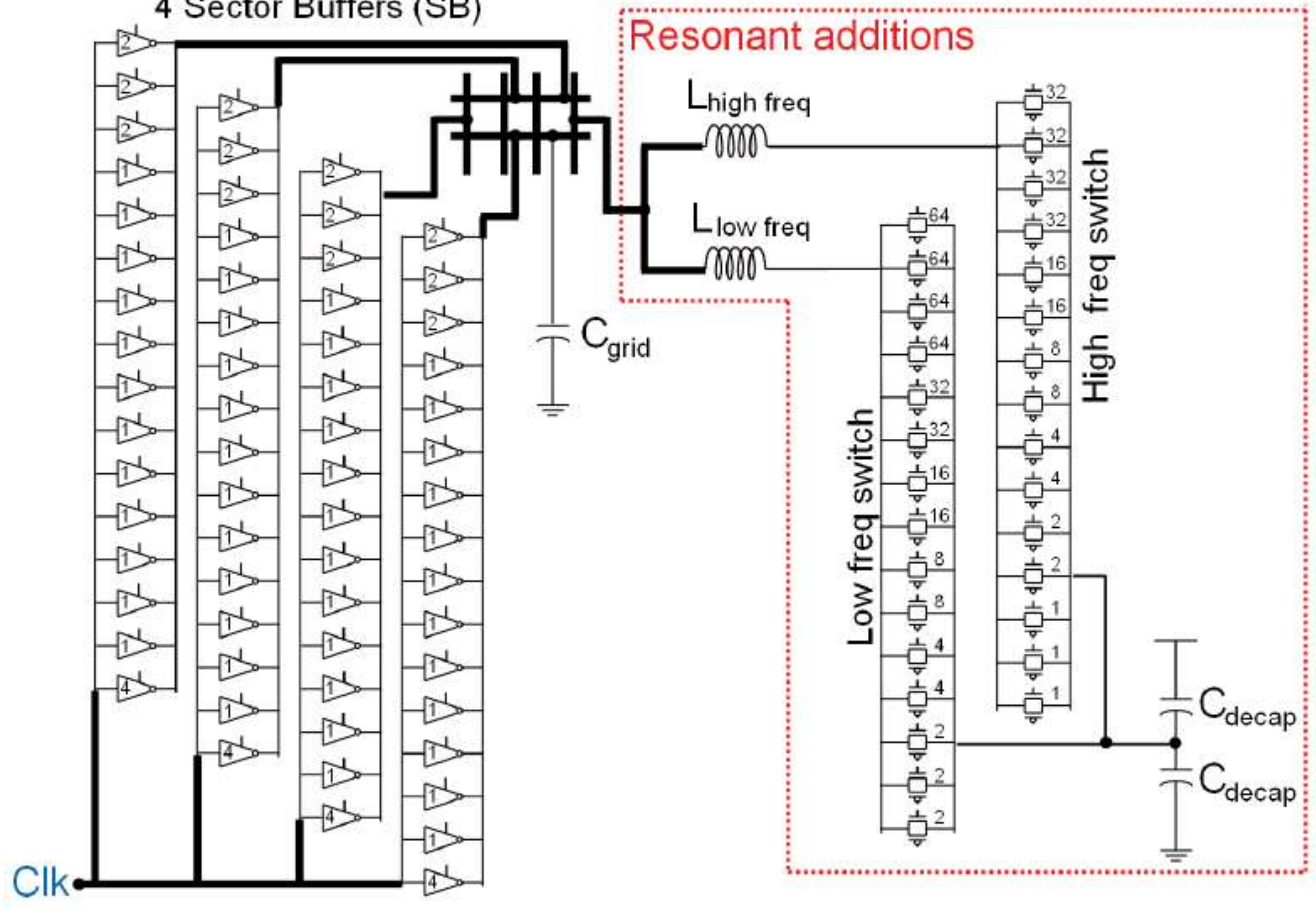


Subsequently, we discuss this resonant structure.

Programmable sector buffers [94]

- The resonant clock enhancement includes programmable sector buffers i.e. sector buffers with programmable drive strength, in order to set the drive strength of each sector buffer to its lowest possible level in order to save power.
- Actually, each four parallel connected sector buffer consists of 16 parallel connected gated inverters, as indicated in the next Figure.

Implementation of the programmable strength sector buffers [94]



Note, the relative device sizes are annotated, wide lines are Ultra Thick Metal lines.

Parallel switchable inductors [94]

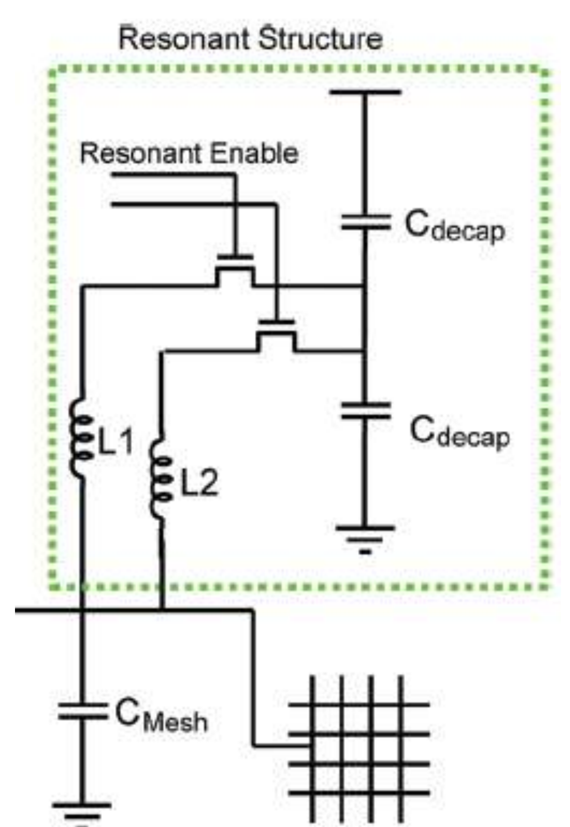
POWER8's resonant clock enhancement is designed to have two programmable resonant frequencies to cover a wide range of scalable clock frequencies.

- The principle of implementation is to have two switchable inductors, L1 and L2.
- When both associated switches are open the resonant enhancement is disabled.
- When the switch associated with L2 is open, only L1 is active and the lower resonant frequency mode is activated.
- Finally, when both switches are closed, L1 and L2 are parallel switched and the resulting inductance (Lr) becomes:

$$Lr = \frac{L1 \times L2}{L1 + L2}$$

Accordingly, the higher resonant frequency mode becomes active.

If both inductors have the same value, as in the case of the POWER8 processor, the lower resonant frequency is $1/\sqrt{2}$ x (~ 0.7 x) the higher resonant frequency.

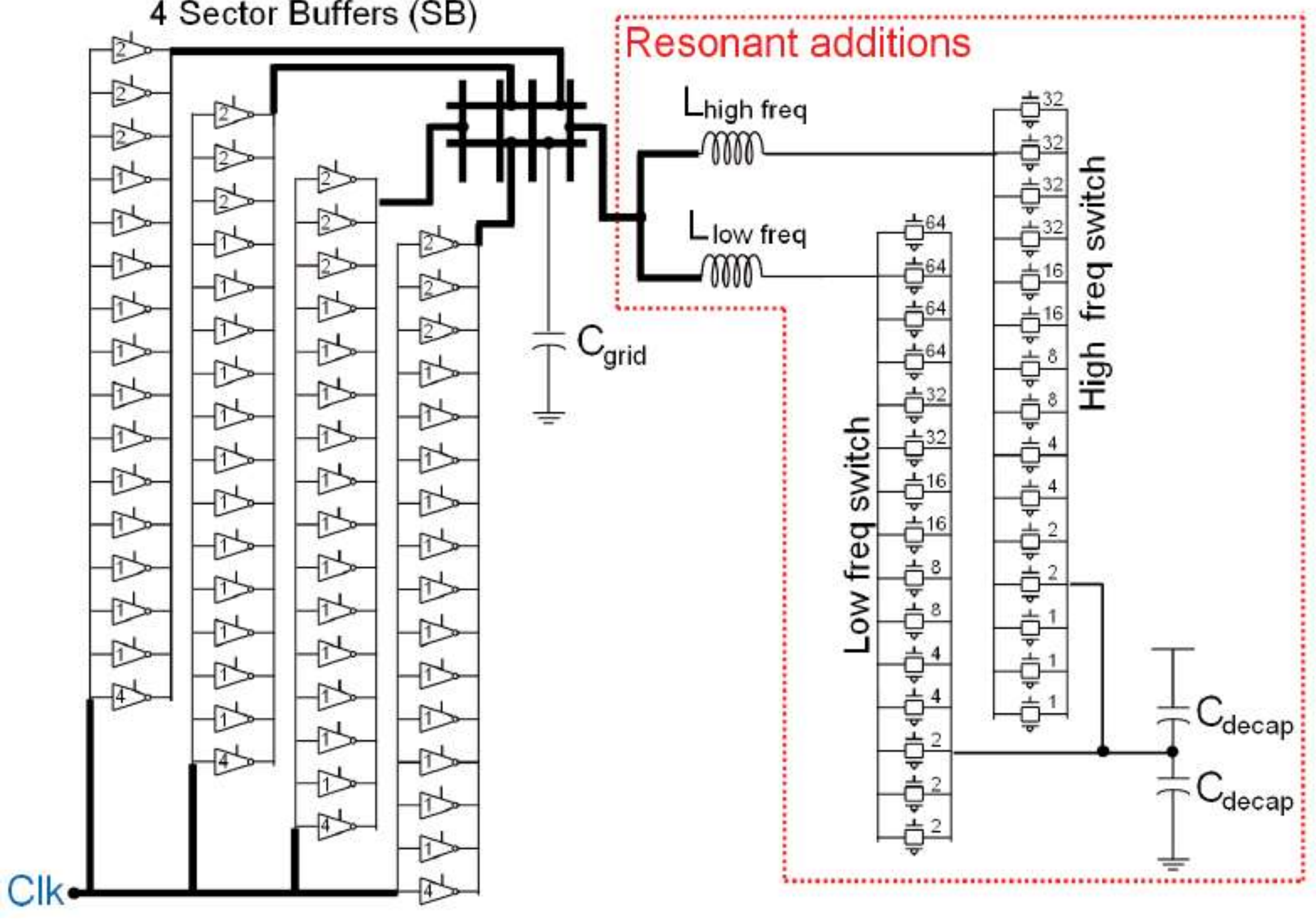


Part of a resonant clock sector [94]

On-the-fly mode changing [94]

In order to eliminate the need for inserting idle cycles between mode transitions, i.e. to achieve on-the-fly mode changing, each resonant mode nFET switch can be opened and closed in 16 small steps, to allow a gradual transition between modes under the control of the OCC (On-Chip-Controller), (see the next Figure).

Implementation of gradual mode changing [94]



Note, the relative device sizes are annotated, wide lines are Ultra Thick Metal lines. Both the Low and High frequency switches can be operated in 16 small steps.

Layout of the implementation of a resonant clock sector in the POWER8 [95]

Each resonant clock sector is built up symmetrically, as shown below.

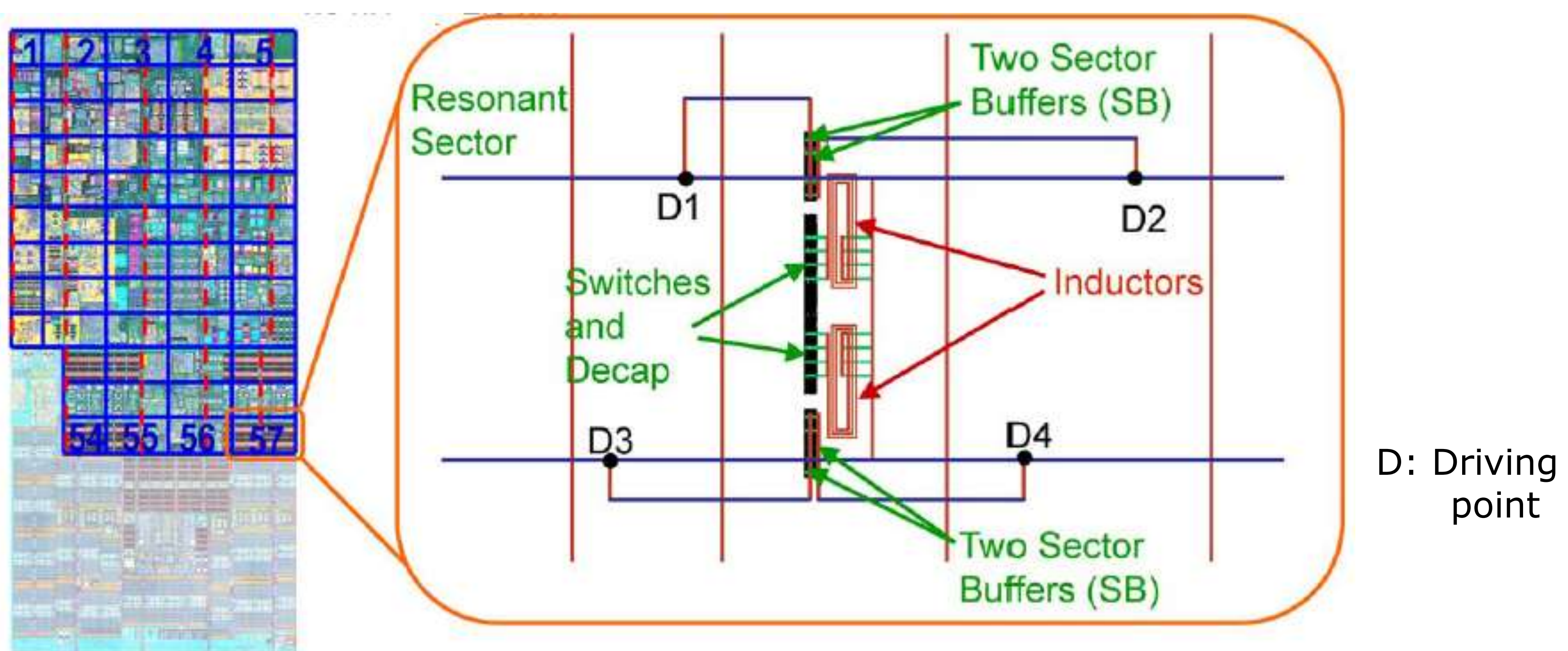


Figure: Layout of the implementation of a resonant clock sector in the POWER8 [95]

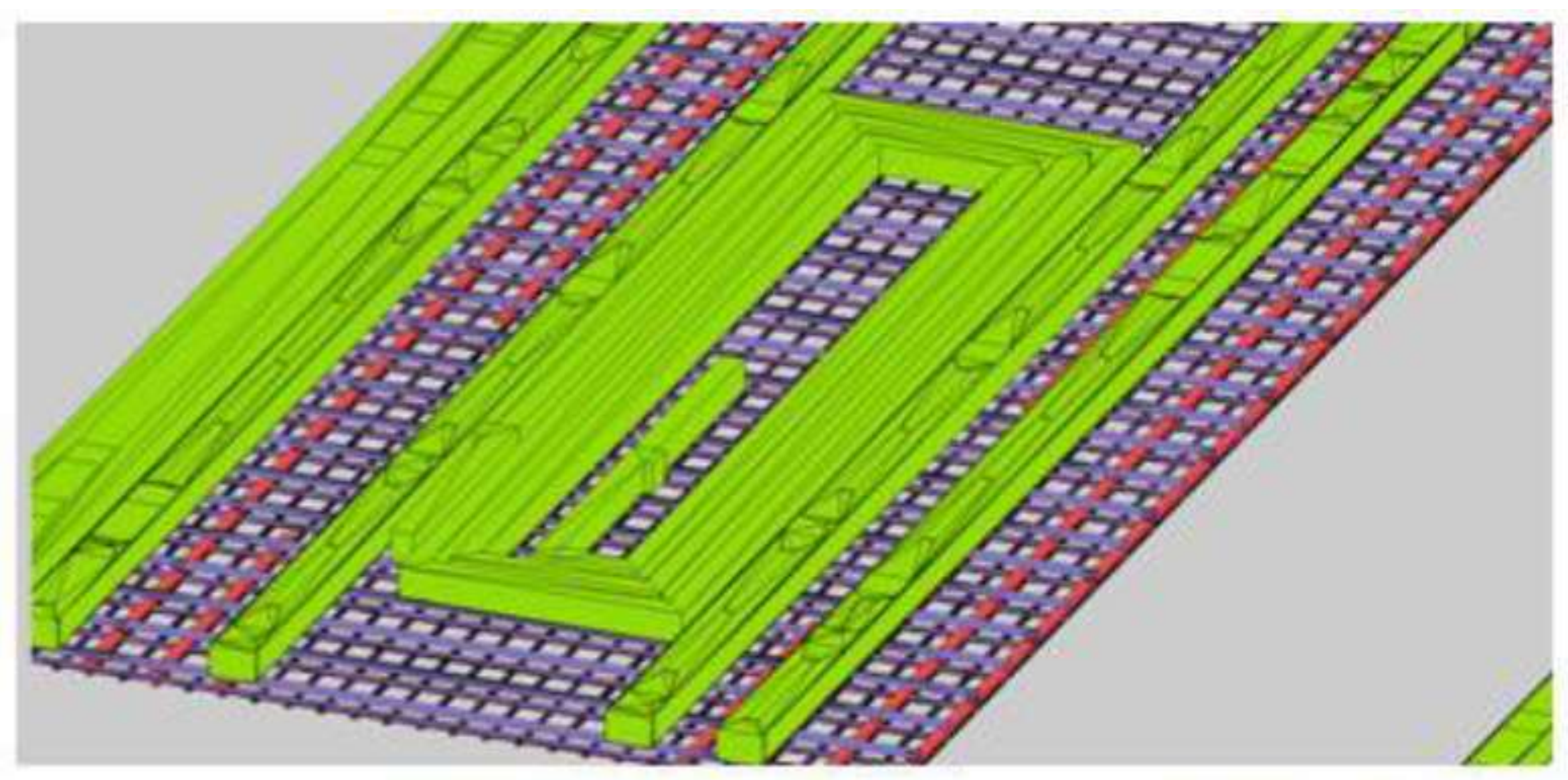
We note that both inductors have the same value but they are tuned according to the actual mesh capacity, as discussed next.

10.3.3 Resonant clocking (32)

Tuning inductors

- As the mesh capacitance varies greatly in different clock sectors across the chip each clock sector was tuned to the same two resonant clock frequencies by implementing inductors inversely proportional of the actual mesh capacitance.
- There were thirteen different on-chip inductors designed ranging from 0.3 nH to 2.5 nH, utilizing a new Ultra Thick Metal layer (UTM), as indicated in the next Figure.

Implementation of inductors on the Ultra Thick Metal (UTM) layer [93]

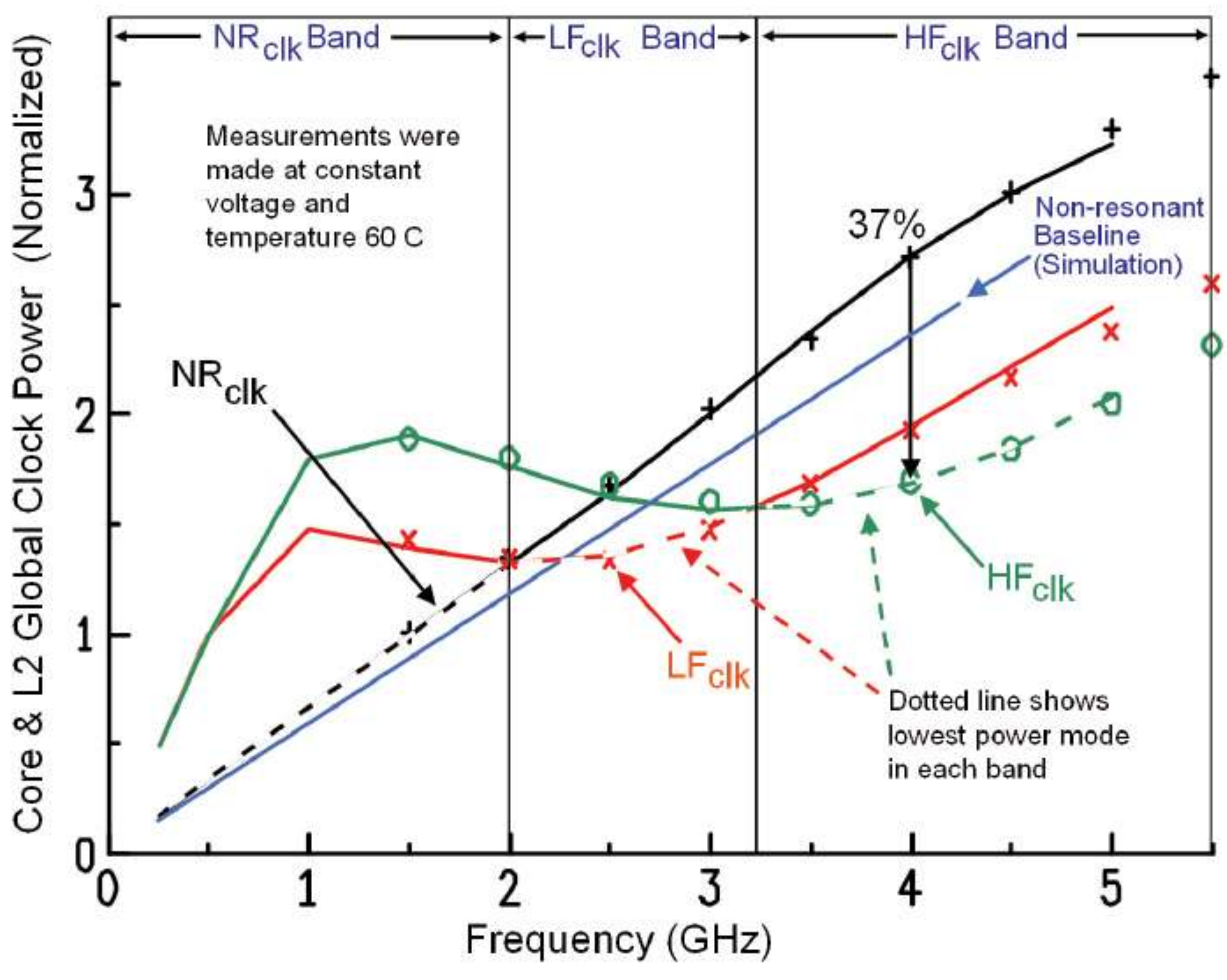


Achieved power reduction by resonant clocking in the POWER8-1 [96]

The next Figure shows

- the simulated and measured power consumption without using resonant clocking (NR_{clk})
- the measured power consumption in the high frequency mode (HF_{clk}) and
- the measured power consumption in the low frequency mode (LF_{clk}).

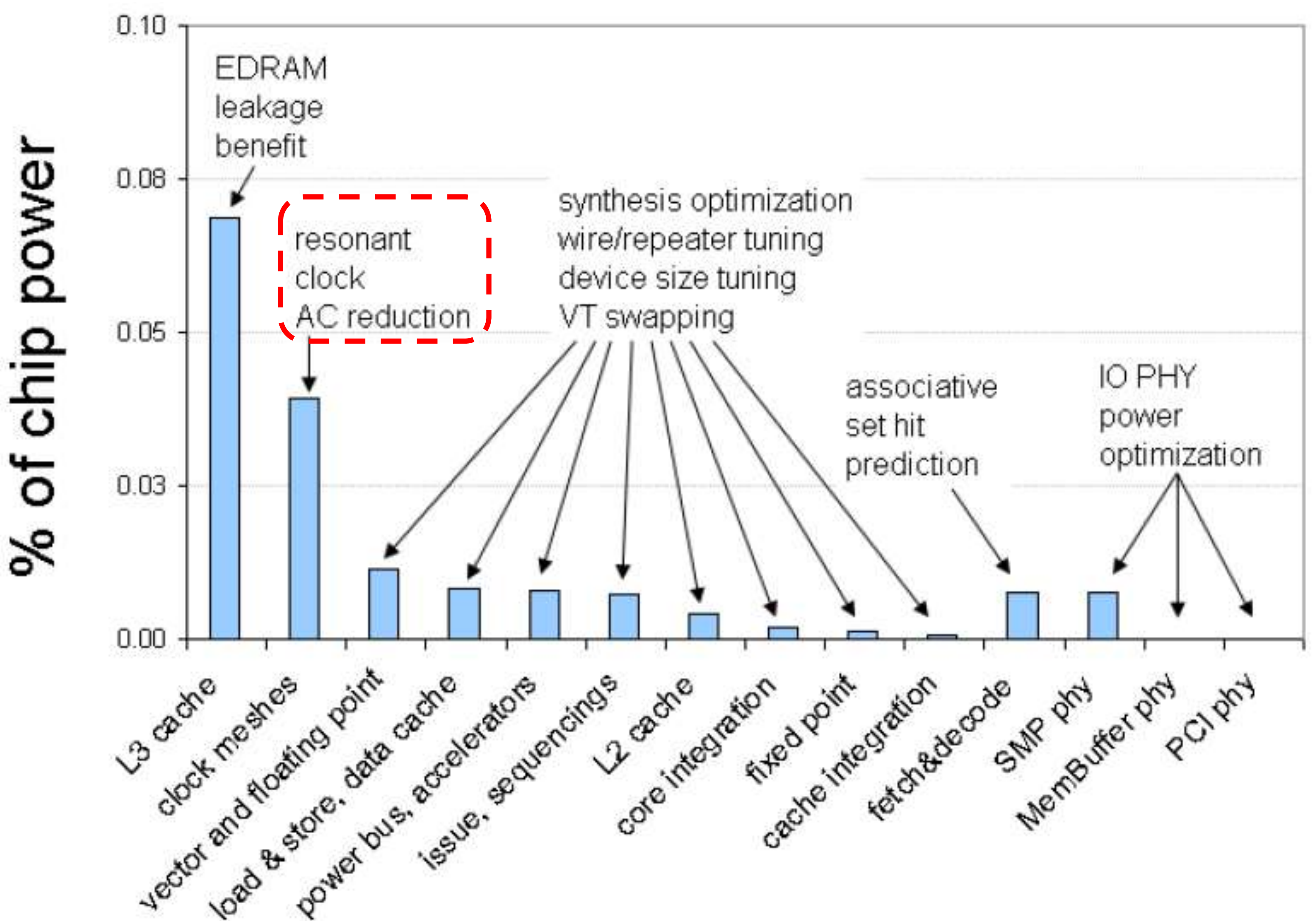
Power consumption by resonant clocking in the POWER8 [96]



Achieved power reduction by resonant clocking in the POWER8-2 [96]

- As the Figure indicates for low clock frequencies (up to 2 GHz) the non-resonant operation (NR_{clk}) yields the lowest power consumption.
- In the intermediate frequency range, between about 2 GHz and 3.2 GHz the low frequency mode (LF_{clk}) results in the minimum power consumption, whereas beyond above 3.2 GHz the high frequency mode (HF_{clk}) provides the lowest power consumption.
- The maximum power saving is achieved at the higher resonant frequency amounting to about 37 % compared to the non-resonant mode.

Components of the 17 % total power savings achieved in the POWER8 chip [97]



Concluding remarks to resonant clocking introduced into the POWER8

a) From proposing resonant clocking to its commercialization (in AMD's Piledriver (2012) and IBM's POWER8 (2014) about ten years have passed.

b) We point out that there was the same author who proposed resonant clocking (Chan S. C.) and also took part in the development of both the experimental implementation on a modified Cell chip (2008) and in the POWER8 (2014).

c) Resonant clocking is a sophisticated technique, it is questionable whether it becomes part of the mainline evolution of processors.

10.3.4 Hardware transactional memory

10.3.4 Hardware transactional memory (HTM)

10.3.4.1 Introduction

- Transactional memory is an efficient synchronization mechanism in concurrent programming used to effectively manage race conditions occurring when multiple threads access shared data.
- In order to make understandable the concept and implementation alternatives of transactional memory first we discuss race conditions and give a brief overview how address races can be addressed in multithreaded programs.

Race conditions-1 [98]

- As an example, let's consider a program for counting requests by a global counter, totalRequests that is incremented each time a request is completed.
- Obviously, a sequential program performing this task has a quite trivial code, as follows:

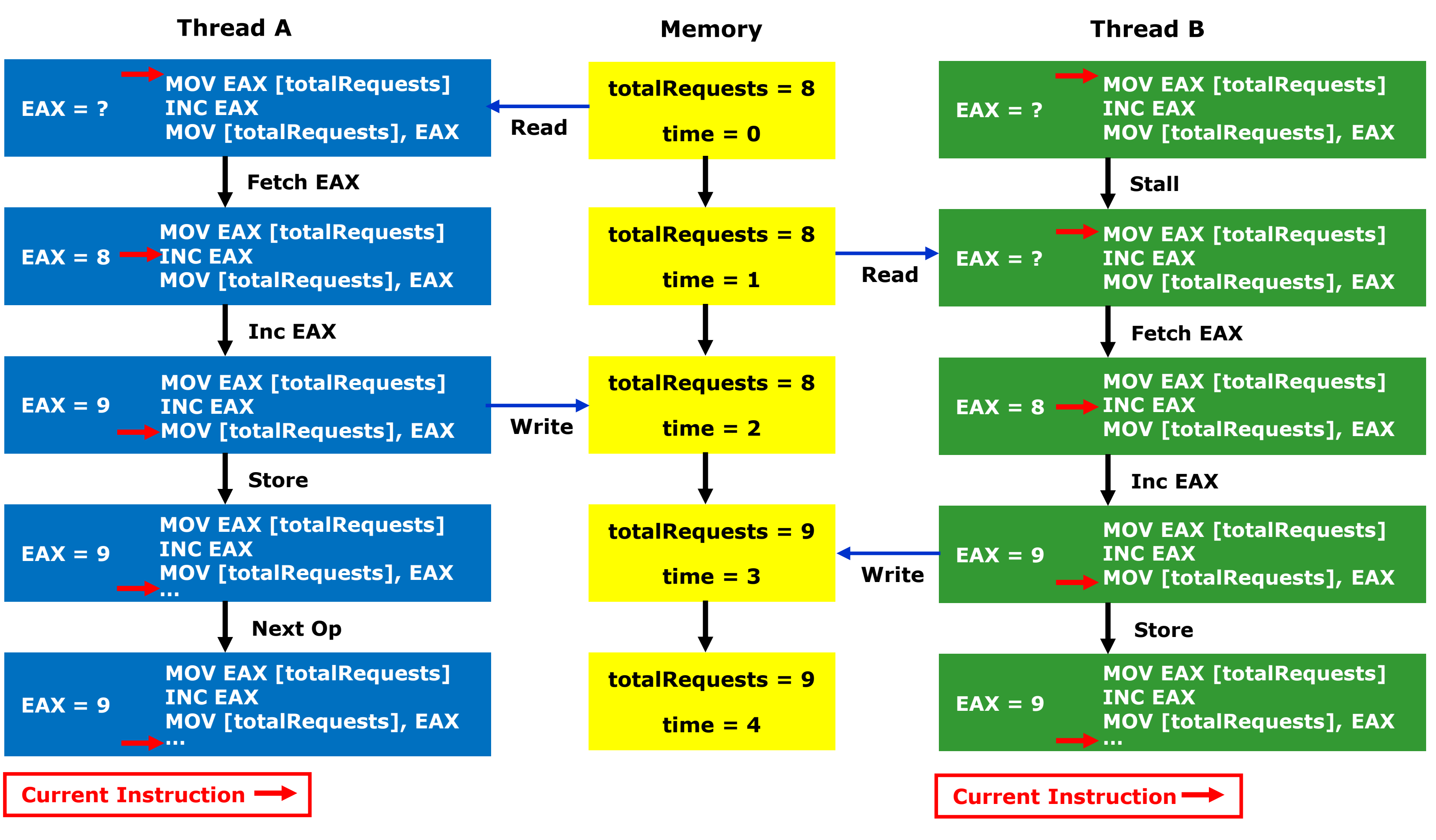
```
totalRequests = totalRequests + 1
```

- Assuming that the variable totalRequests is implemented in the memory, the compiler might translate the increment operation into the following assembly code:

```
MOV EAX, [totalRequests]     // load the value of totalRequests into register
INC EAX                      // update register
MOV [totalRequests], EAX     // store updated value back to memory
```

- In a concurrent program however, synchronization problems may arise when multiple threads service requests and update the global counter totalRequests independently from each other.
- We demonstrate this in the next Figure by showing what may happen if two threads will run the above code simultaneously.

Example for obtaining an incorrect result due to missing thread synchronization [98]



10.3.4 Hardware transactional memory (4)

Race conditions-2 [98]

- In the above example both threads will load the same value for totalRequests, then both will increment it and store it back to totalRequests.
- As seen in the above example, after both threads have executed their code sequences (requests) the actual value of totalRequests will be incremented only by one despite the fact that altogether two requests were processed and the value of totalRequest should have been incremented by two.

  Clearly, this kind of processing results in an error.
- Situations like this are called race conditions, their correct handling needs synchronization between the threads, to be discussed next.

Addressing race conditions

Basically there are two mechanisms to address race conditions in multithreaded programs, as indicated below:

Basic mechanisms to address races in multithreaded programs

Locks

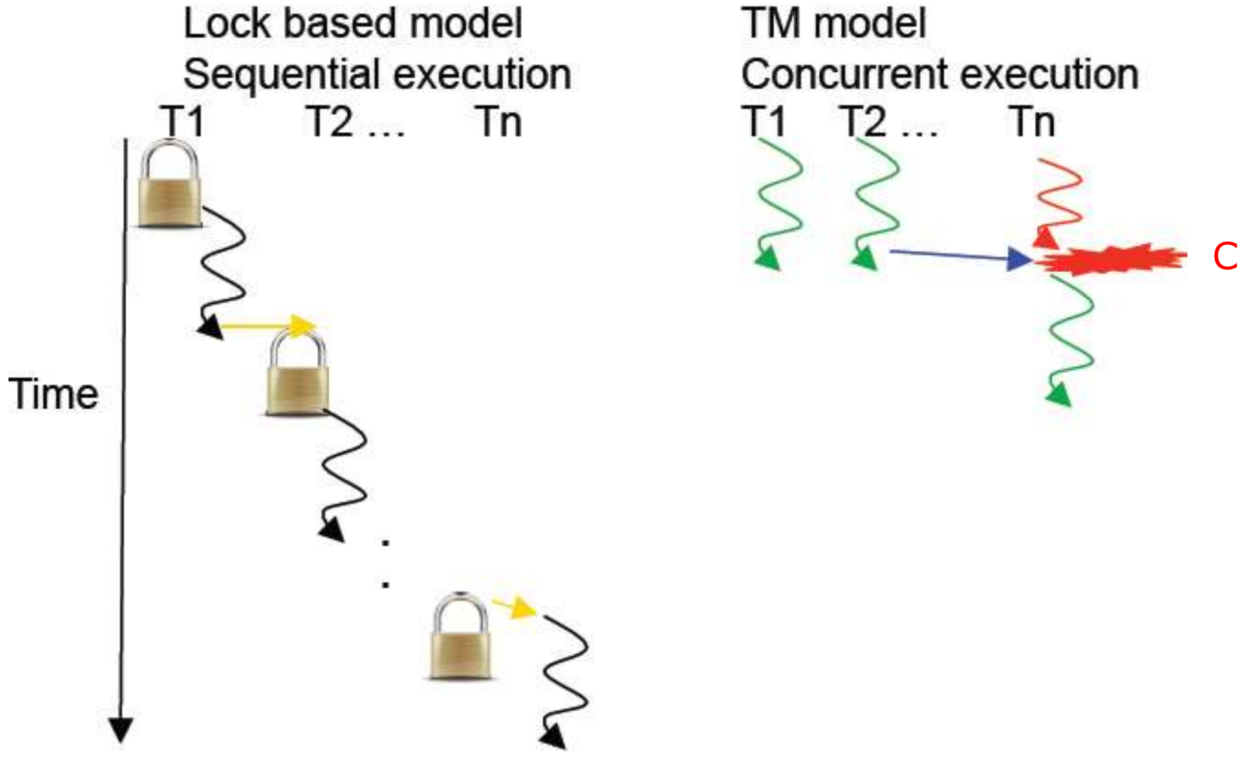
Pessimistic approach,
it intends to prevent possible conflicts
by enforcing serialization of transactions
through locks.

Transactional memory (TM)

Optimistic approach,
it allows access conflicts to occur
but provides a checking and repair mechanism
for managing these conflicts, i.e.
it allows all threads to access shared data simultaneously
but after completing a transaction,
it will be checked whether a conflict arose,
if yes, the transaction will be rolled back and
then replayed if feasible else
executed while using locks.

The next Figure illustrates these synchronization mechanisms.

Illustration of lock based and transaction memory (TM) based thread synchronization [78]



a) Locks-1 [98]

- The most common way of addressing race conditions is to use locks, called also monitors, mutexes or binary semaphores.
- Locks provide a mechanism for ensuring that at any time only one thread can execute a particular section of code (called critical section) that includes shared data.
- The principle of using locks is as follows:

  In a multithreaded program a thread that needs access to shared data must first acquire a data lock, then access the shared data and finally release the lock.
- As the lock mechanism is based on the suspicion that another thread may cause a program error, and this needs to be prevented, the lock mechanism is actually a pessimistic approach.
- From another point of view locks can be considered as a mechanism that enforces serialization of accessing shared data in concurrent programming.

10.3.4 Hardware transactional memory (8)

a) Locks-2 [98]

- On many systems however, acquiring locks may be time consuming since locks will be released only after completing the related transaction, thus locks make accessing shared data vastly more expensive than accessing non-shared data.
- Locking can be especially burdensome when the shared data has low contention between multiple threads.
- For this reason one of the proposals aiming at speeding up concurrent programs is an optimistic, speculative execution approach, designated as transactional memory, to be discussed next.

b) Transactional memory (TM)

The concept of the transactional memory (TM)-1

- Unlike the locking technique, which is a pessimistic approach and enforces serialization of accessing shared data in concurrent programming to avoid conflicts, TM is an optimistic approach allowing threads to access shared data in concurrent programs irrespective of other threads, i.e. it allows access conflicts to occur but provides a checking and repair mechanism for managing these conflicts.
- This approach replaces serialization and waiting for locks by the checking and repair mechanism of TM and has the potential for speeding up the execution of concurrent programs.

10.3.4.2 Implementation of transactional memory (TM)

There are however three different alternatives how TM can be implemented, these will be discussed next.

Implementation of the concept of transactional memory

Early proposals for hardware implementation of transactional memory	**Software transactional memory**	**Hardware transactional memory**
Proposals for purely hardware based implementation of transactional memory	The implementation of transactional memory is embedded into a programming language	Transactional memory is implemented typically as an ISA extension of a processor family with appropriate hardware support
Papers from 1986,1993	Mainly between 1990-2010	In mainstream processor families since about 2010

10.3.4 Hardware transactional memory (11)

a) Early proposals for hardware implementation of transactional memory

- The concept of transactional memory emerged already in the 1980's [99] and became widely discussed in the 1990's.
- First proposals however, aimed at pure hardware implementation, e.g. [100], and became soon over rolled by the idea of software transactional memory [101], to be discussed next.

b) Software transactional memory (STM)-1 [102]

- STM is implemented typically by providing a set of specific program constructs, such as APIs or program instructions, embedded into a programming platform, like the Java platform.
- With STM, programmers mark portions of their multithreaded programs that include shared data as being "atomic" by suitable language constructs provided.

  We note that these atomic blocks are also designated as "transactions".
- Instructions within an atomic block are encapsulated, i.e. either the whole block will be executed, or none of its instructions does.
- Within an atomic block, the thread can access shared data without locking it irrespective what other threads do, performs all the computations involved, and then writes the result back to the memory.

b) Software transactional memory (STM)-2 [102]

- At the end, the thread commits the transaction and checks whether shared data has been modified by other threads since the atomic operation was started.

  Here we take for granted that STM provides a proper mechanism for checking this, e.g. by maintaining transaction variables guarding memory regions.
- If checking results in no conflicts, all modifications become visible to all other threads and the considered thread will carry on with its work.
- If a conflict arose since the shared value has been accessed by another thread as well, the transaction becomes aborted and the work done is rolled back.
- Typically when this happens, the routine that manages data access conflicts first retries to execute the atomic block in concern, nevertheless if it proves not feasible, it cares for the execution of the transaction while using locks.
- STM is considered as a suitable mechanism to simplify concurrent programming but without proper hardware support it has significant overhead and often does not result in notable speed-up of concurrent programs [103].

brings well-known advantages:
freedom from deadlock and priority inversion

Examples for STM implementations

There are a large number of STM implementations, subsequently we point out only a few of them, as follows:

Haskell (since 1990), an open standard functional language.

Its STM extension was published in 2006.

Skala (since 2003), object-oriented with full support of functional programming, it runs on the Java platform, source code will be translated to Java byte code.

Clojure (since 2007).

It is a dialect of the LISP functional programming language and runs on the Java platform.

It is the only language, the core of which supports transactional memory [104].

C/C++ with GCC 4.7 (since 2012) or higher GCC version.

c) Hardware transactional memory (HTM)

Overview

- HTM is implemented typically by an appropriate ISA extension[1] and a suitable hardware enhancement of the processor.
- The ISA extension of the processor family, like the TSX x86 ISA extension from Intel, is used to declare critical sections of shared data in the thread, called transactions and usually also in order to simplify writing of the failure routine that decides what to do when the execution of the transaction fails.
- Main task of the hardware enhancement is to execute the new instructions, support the efficient saving and restore the program state as well as to detect and address access conflicts and also possible execution failures.

[1]Blue Gene/Q uses compiler directives instead of instructions

The programming model of HTM

- The programmer marks the begin and the end of critical sections of a thread, i.e. sections including data shared with other threads, in a similar way as done for STM, but now marking is typically done by new instructions of a dedicated ISA extension supporting HTM, such as TSX from Intel.
- Again these critical sections (termed also as atomic blocks) are termed "transactions".
- Transactions are interpreted in the same way as for STM; i.e. instructions within a transaction are considered encapsulated, that is either the whole block will be executed, or none of its instructions does.

  A thread can access shared data included in the transaction without locking the transaction irrespective what other threads do and performs all the computations involved.

10.3.4 Hardware transactional memory (17)

Principle of operation of HTM-1

- When a transaction is initiated, typically the program state of the thread, that is the content of the user programmable register files, needs to be saved, since after an access conflict or failure the program state needs to be restored to the state that has been at the beginning of the transaction.
- Saving and restoring the register space can be done differently, e.g.
  - by checkpointing, i.e. by saving the content of the programmable registers into an additional register space and reloading it if needed, or
  - by software, i.e. by storing the content of programmable registers into the memory and reloading these data if needed, or
  - partly by checkpointing specific registers and partly by software for other registers.
- The Table at the end of this section gives an overview how specific processors implement saving and restoration of the register space.
- Usually, also the starting address of the transaction is saved in preparation of failure handling to allow a possible retry of the execution of the transaction.

Principle of operation of HTM-2

- While a transaction is executed specific hardware checks for access conflicts and failures.

  This hardware is implemented mainly in the cache system such that the cache system maintains suitable status flags and thread identifier tags.
- When an access conflict or error arises hardware generates a jump to the failure routine.
- As the hardware transaction memory is in fact a best effort mechanism, there is no guarantee for a success, so it is the programmer's responsibility to write an appropriate failure handler to address occurring failures.
- The failure handler first restores the program state to the one that was at the beginning of the transaction.
- This requires a restoration of the register state and rolling back all stores done during the execution of the transaction.
- Subsequently the handler checks available condition codes to decide about the best way to address the failure.
- If feasible, the handler tries to re-execute the aborted transaction up to a given number of times.
- If it does not succeed or it is not feasible to handle the failure in this way, the handler passes control to a path where the transaction will be executed in a serial fashion while using locks.

Principle of operation of HTM-3

- By contrast, when the transaction succeeds, at the end of the transaction the updated program state becomes the new architectural state which is then made visible to all threads and the processing proceeds.

Example: A simple transaction's assembler code for the IBM's POWER8 [105]

- The following assembler code of a simple transaction writes the content of GPR 5 into a memory location whose address is given in GPR 4, into a memory location which is assumed to be shared among multiple threads.
- If the transaction fails due to a persistent cause, control is passed to an alternate code path at the label *lock based update* (its code is not shown).

```
trans_entry:
        tbegin.                      # Start transaction
        beq- failure_hdlr            # Handle transaction failure
# Transaction Body
        stw r5, 0(r4)                # Write the value of r5 to the memory pointed to by r4.
        tend.                        # End transaction
        b trans_exit
# Failure Handler
        failure_hdlr:                # Handle transaction failures:
        mfspr r4, TEXASRU            # Read high-order half of TEXASR
        andis. r5, r4, 0x0100        # Is the failure persistent?
        bne lock_based_update        # If persistent, acquire lock and then perform the write
        b trans_entry                # If transient, try again.
# Alternate path for obtaining a lock and performing memory updates (non-transactional path)
lock_based_update:
trans_exit:
```

Sample assembler code of a simple transaction written for IBM's POWER8 [105]

Overview of HTM implementations in commercial processors

	Sun Rock (2009)	IBM Blue Gene/Q (2011)	IBM Syst./z EC12 (2012)	Intel Haswell (2013)	IBM POWER8 (2014)
ISA	SPARC v9	Power ISA 2.06	z/architecture	x86	Power
Programming by	ISA extensions	Compiler directives	ISA extensions	ISA extensions	ISA extension
Initiate/commit transactions by	checkpoint/ commit	#pragma tm_atomic	TBEGIN/ TEND	xbegin/ xend	tbegin/ tend
Saving register state	by Checkpointing	by SW	GPR checkpointed, else by SW	by checkpointing	by checkpointing
Saving store data until commit	in the Store Queue	In the L2 cache	in the Store Queue	in the L1 cache	in the L2 cache
Conflict detection	L2 cache tracks conflicts mainly by maintaining appropriate flags	L2 cache tracks conflicts mainly by maintaining appropriate flags	L1 cache tracks conflicts mainly by maintaining appropriate flags	L1 cache tracks conflicts mainly by maintaining appropriate flags	L2 cache tracks conflicts mainly by maintaining appropriate flags
Hardware Lock Alision support	No	No	No	Yes	Yes
Multiproc. support	Yes(?).	No	Yes	Yes	Yes

10.3.4.3 Brief description of HTM implementations in commercial processors

Subsequently we briefly describe how HTM is implemented in commercial processors included in the Table above.

The processors covered are:

a) Sun's Rock (2009)

b) IBM Blue Gene/Q (2011)

c) IBM's System/z EC12 (2012)

d) Intel's Haswell (2013)

e) IBM's POWER8 (2014)

a) HTM in Sun's Rock (SPARC v9) (2009)-1 [106], [107]

- Sun cancelled the chip before its release about 6/2009 due to product delays and various glitches already in the shadow of the planned acquisition by Oracle.
- Rock is the first hardware implementation of transactional memory (TM).
- Rock supports TM by two new instructions (*checkpoint/commit*) to mark begin and end of transactions.
- When a *checkpoint* instruction initiates a transaction the register state will be saved in specific checkpoint registers to allow a rollback if the transaction aborts and also the PC value pointing to the beginning of the transaction is saved for a possible retry of executing the transaction.
- Loads within the transaction are executed speculatively and set flag-bits on the cache lines that are read.
- Stores within the transaction are placed in the Store Queue in program order, nevertheless the actual size of the Store Queue limits the extent of a possible transaction.
- Addresses of stores are sent to the L2 cache, which then tracks conflicts with loads and stores from other threads.
- If the L2 cache detects such a conflict, it reports it to the core, which then lets fail the transaction.

a) HTM in Sun's Rock (SPARC v9) (2009)-2 [106], [107]

- After committing the transaction, it will be checked whether the transaction succeeded or failed.
- If the transaction succeeds, the speculative register updates will become the architectural state as well as the speculative loads and stores are executed and the updated architectural state is made visible for all threads.
- If the transaction fails, the failure routine included into the thread, discards the speculative register updates performed within the transaction, restores the checkpointed state, and continues the execution from the saved PV value.
- We note that Rock support a kind of Lock Elision, designated as Transactional Lock Elision (TLE) which is a software implementation of Lock Elision.

b) HTM in IBM's Blue Gene/Q (Power ISA v. 2.06) (2011) [108]

- The supercomputer Sequoia is based on the Blue Gene/Q architecture, it is the first commercially available system with hardware support for TM.
- Blue Gene/Q does not supports TM by an ISA extension, instead transactions are marked by compiler directives.
- When a transaction is started no hardware checkpointing takes place to save register content, this remains the responsibility of the software.
- Blue Gene implements hardware support for transactional execution primarily in the L2 cache, which serves as the point of coherence.
- Stores within a transaction are buffered in the L2 cache such that the L2 cache will store multiple versions of the same physical cache line, while each version occupies a different cache way.
- The L2 cache controller detects access conflicts by tracking the thread IDs while the threads read or write cache lines.
- When during the transaction no access conflicts or other failures are detected the transaction succeeds and the transactional writes are made visible to other threads, else the hardware sends interrupts to the threads involved in the conflict.
- Conflicts need to be resolved by the failure routine of the thread.
- Blue Gene/Q didn't offer any kind of multiprocessor support for TM.

c) HTM in IBM's System z EC12 (z/Architecture) (2012) -1

- The programming model of TM is based on an ISA extension including six instructions.
- Transactions are initiated by the *TBEGIN* and committed by the *TEND* instruction.
- At transaction initiation a restricted checkpointing is done that is confined to the General Registers (GR) only.
- Actually, checkpointing consists of saving pairs of general registers (appointed by an 8-bit mask set in the instruction) into a special transaction-backup register file that is performed by a number of micro-operations, as preparation for the case when a transaction aborts and the content of the GRs needs to be restored.

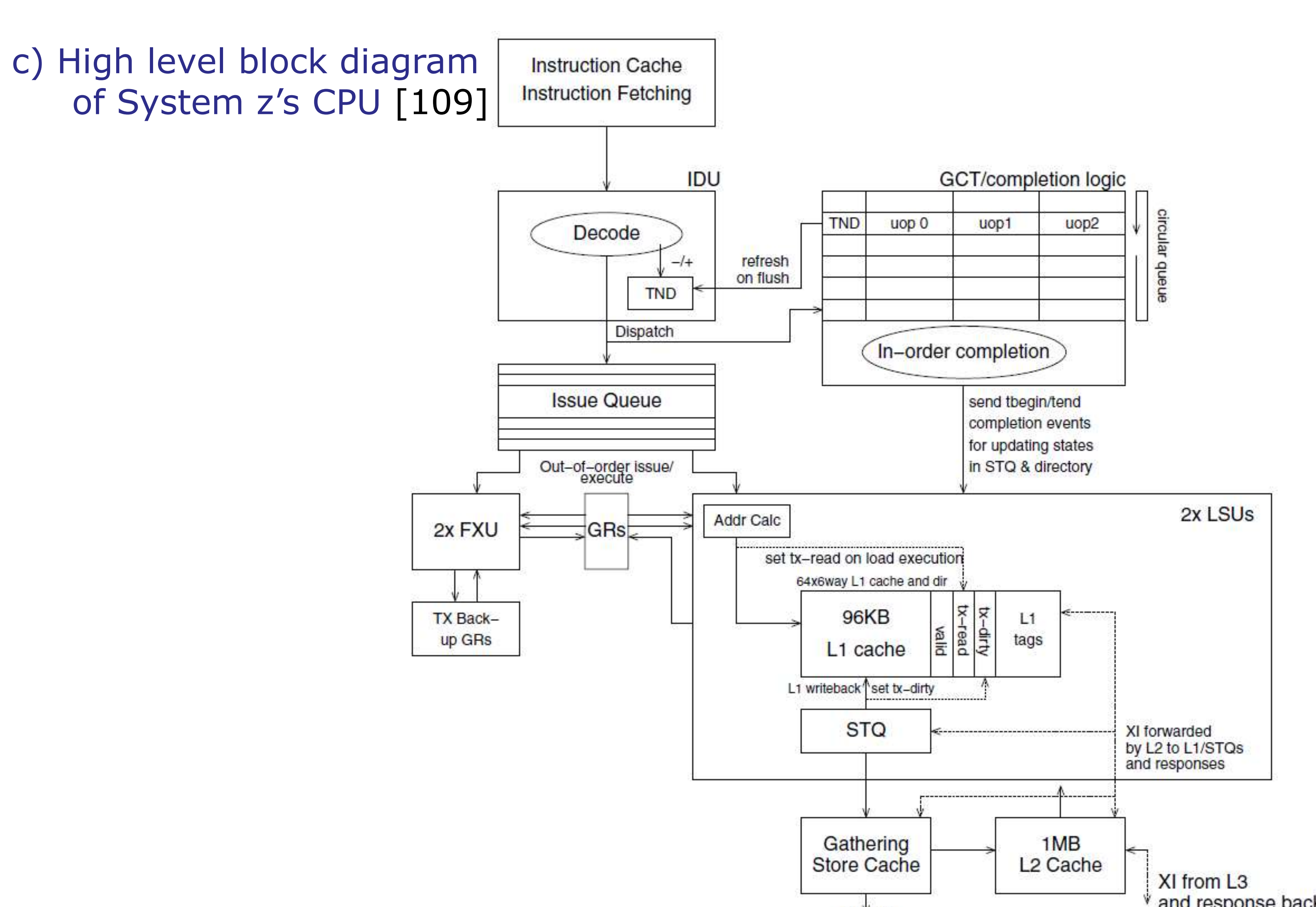
  At the same time also the value of the PC is saved to allow a later retry of the execution of the transaction if needed.

  For further register files of the processor, like Floating-Point Registers (FPRs) or Access Registers (AR), no hardware save/restore mechanism is provided, it is up to the abort handler to save necessary registers before entering a transaction and to restore those register in case of a transaction abort.
- Stores in transactions are kept in the Store Queue.
- Conflict detection is performed by the LSU (Load/Store Unit) such that the LSU tracks L1 cache line accesses during transactional execution, and triggers an abort if a conflicting memory access is detected.

c) HTM in IBM's System z EC12 (z/Architecture) (2012) -2

- The abort handler may retry the transaction, or it may perform a branch to a non-transient fallback path (that employs locks) depending on whether the conflict is considered by the CPU as transient or permanent (indicated by the condition code setting).
- A conflict is considered transient when it is originated e.g. from a conflicting memory access of another CPU and permanent when it is caused e.g. by a not eligible instruction in the transition.

c) High level block diagram of System z's CPU [109]



d) HTM in Intel's Haswell (x86) 2013 [110], [111]-1

- In 2012 Intel announced the Transactional Synchronization Extensions (TSX) to the x86 ISA for supporting TM.
- TSX provides two software interfaces:
  - the Hardware Lock Elision (HLE) and
  - the Restricted Transactional Memory (RTM)

  interfaces.
- The Hardware Lock Elision (HLE) interface is a legacy x86 compatible ISA extension (comprised of the XACQUIRE and XRELEASE prefixes used to mark transactional regions.
- HLE is compatible with the conventional lock-based programming model.

  Software written using the HLE hints can run on both legacy hardware without TSX and new hardware with TSX.
- With HLE there is still a lock, but ideally the lock is not used at all—it is elided.
- In fact, when the code is executed, the CPU starts an HTM transaction, but does not acquire the lock.

  Only when there is a conflict the transaction rolls back, acquires the lock, and will be then executed on the fallback path as a serial, lock controlled execution.
- So HLE provides a mechanism such that even in case of an abort due to a conflict or other reason, there is always a "slow path" available that guarantees successful execution.

d) HTM in Intel's Haswell (x86) 2013 [110], [111]-2

- The Restricted Transactional Memory (RTM) is a new instruction set interface (comprised of the XBEGIN, XEND, and XABORT instructions) that allows programmers to define transactional regions in a more flexible manner than is possible with HLE.
- Unlike the HLE extensions, but just like most new instruction set extensions, RTM instructions will generate an undefined instruction exception (#UD) on older processors that do not support RTM.
- When XBEGIN initiates a transaction, the programmable register space will be checkpointed.
- Stores in transactions are buffered in the L1 cache.
- An extension of the cache coherence protocol cares for tracking read and write collisions.
- In case of a successful transaction the updated register space and stores kept in the L1 cache are committed.

d) HTM in Intel's Haswell (x86) 2013 [110], [111]-3

- In the case of access conflicts or due to various further reasons, like exceeding cache associativity limits, certain interrupts, etc. the transaction fails.

  Then the CPU undoes all changes and reports an error that the application has to address.

- So, even though in most cases RTM will successfully commit transactions, there is no guarantee that a transaction will ever succeed, it remains always a best effort mechanism and the programmer has to care for an alternate code path for the case when the transaction fails.

Note

- In 8/2014 Intel admitted that there is a bug in the implementation of the TSX extension of the Haswell and Broadwell processors that can not be cured by a microcode patch so Intel issued a microcode update that deactivates TSX on these processors.
- Nevertheless, Intel promised to correct the TSX bug and enable this feature on future processors again [112].

e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-1

- The implementation of TM in POWER8 is closely related to the processor's cache system, so we first give an overview of it.
- Each POWER8 core is supported by a private three level cache hierarchy consisting of the following caches:
  - a write-through L1 cache (32 kB I-cache and 64 kB D-cache),
  - an 512 kB write back L2 cache that is inclusive of the L1 cache and is the system point of coherence, and
  - a 8 MB large semi-private L3 victim cache that can accept lateral evictions of data from other L3 caches as well.
- The next Figure gives an overview of the cache system indicating also the extensions supporting TM (green colored).
- Reference [] gives a very detailed description of the hardware implementation of POWER8's TM, by contrast here we restrict us to a strongly simplified discussion of it emphasizing only key points of the implementation.

e) Main components of the hardware implementation of POWER8's HTM [84]

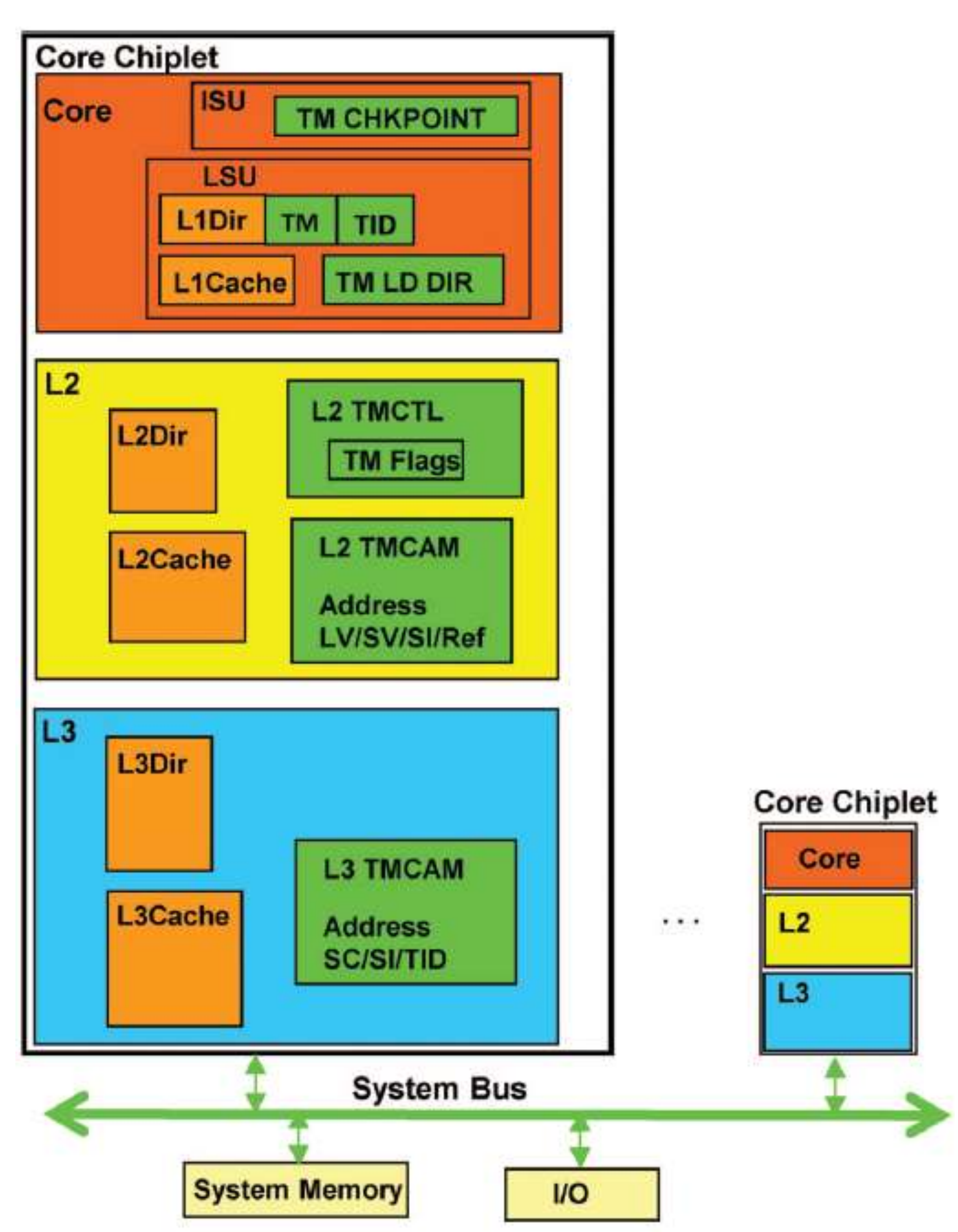
TM: TM bit
TID: Thread ID field
TM LD DIR: TM Load Directory

L2 TMCTL: L2 TM Control Logic
L2 TMCAM: 64-entry L2 TM directory
each entry including

- cache line address
- LV: Load Valid bit
- SV: Store Valid
- SI: Store Invalid
- Ref: 8 reference bits (one per thread)

L3 TMCAM: 64-entry L2 TM directory
each entry including

- cache line address
- SC bit
- SII bit
- TID: Thread ID field



e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-2

- Transactions are initiated by the *tbegin* and committed by the *tend* instruction.
- When a transaction is initiated the user level registers are checkpointed into the existing register renaming hardware of the core to allow a restoration of the user level registers in the event if the transaction fails.
- Checkpointing is performed on demand as the transaction progresses, this shortens the time until checkpointing completes.

  Checkpointing into the register renaming hardware eliminates the need for a dedicated set of backup registers.
- At transaction initiation also the so called Transaction Failure Handler Address Register (TFHAR) is set to point to the instruction that immediately follows the first instruction of the transaction which is typically a conditional branch to the software failure handler.

e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-3

- Because the L1 cache is write-through and not the point at which coherence is resolved, the main transactional tracking mechanism is in the L2 TM control logic (L2 TMCTL) rather than the L1 cache directory (L1Dir) or the Load Store Unit (LSU), and the ultimate decision as to whether a given transaction succeeds or fails is largely made at the L2 cache.
- However, the L1 cache (included in the core's LSU) must track the transactional stores for each thread in order to prevent other threads on the same core from reading or overwriting speculative transactional writes before the transaction is successfully committed.

  Therefore, the L1 cache directory (L1Dir) includes for each cache line a thread ID field (TID) and a "TM" bit (TM).
- When a transactional store occurs to a given cache line, the TM bit is set and the TID is set to indicate which thread has stored to the line.

  This makes the line "private" and therefore inaccessible to any thread other than the storing thread until the transaction commits or fails.
- We note also that in the POWER8 uncommitted transactional stores are held directly in the L2, rather than in the Li cache or the Store Queue, as in other TM implementations (see the Table indicating main features of TM implementations at the end of this Section).

e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-4

- During a transaction the L2 TMCTL logic monitors memory accesses for possible conflicts that may arise either from the local core or through snooping from other cores.
- If such a conflict occurs, the appropriate flag (TM Flag) is cleared, and the L2 TMCTL logic sends a failure indication back to the core, which redirects the transaction to enter the failure handler as early as is practical.

e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-5

- Finally, the transaction attempts to commit at the *tend* instruction.
- If the transaction succeeds, the transaction's stores are committed and the transactional state is set to Non-Transactional to allow other threads to access the updated register and memory state.
- If the transaction fails, the Condition Register is set to indicate the failure code and control is passed to the instruction pointed to by the TFHAR register, i.e. typically to the failure handler.
- The failure handler first determines whether it is reasonable to re-attempt the transaction.

  Reasons not to re-attempt the transaction include various failure indications in the Transaction Exception and Status Register (TEXASR), or the fact that the transaction has already been re-attempted a threshold number of times etc.
- A wide variety of algorithms to control transaction retry are possible, not further discussed here.

e) HTM in IBM's POWER8 (Power ISA) (2014) [84], [113]-6

- The transactional memory architecture and thus its implementation in the POWER8 provides no guarantees of forward progress but remains a "best effort" mechanism.
- Accordingly, if a transaction fails, the software failure handler may re-attempt the transaction some number of times based on indications in the Transaction Exception and Status Register (TEXASR), as described before.
- However, in order to ensure forward progress, if these repeated attempts also fail, the software failure handler typically must fall back to execute the transaction while using locks, i.e. perform the transaction in a critical section within the failure handler.

e) TLE (Transactional Lock Elision)

- TLE is a widely-applicable to utilize hardware TM in a straightforward way as introduced in Sun's Rock processor [106].
- The primary goal of TLE is to allow critical sections that would otherwise execute serially under the control of a lock to execute concurrently on multiple threads that do not acquire the lock.
- TLE, in its simplest form, consists of placing the function of a critical section into a transaction that has a failure handler that falls back to executing the function in a traditional lock-based critical section if necessary.
- When critical sections are replaced by transactions in this way, the inherent hardware conflict checking mechanism of HTM dynamically detects conflicts arising among transactions and cares for discarding and re-executing critical sections serially directed by the failure handler while using locks.
- When conflicts occur only rarely, significant performance gains can be achieved by using TLE compared to executing the same critical sections serially.

10.3.5 Intelligent memory buffers

10.3.5 Intelligent memory buffers

Layout of the memory subsystem in the previous POWER6/POWER7 systems

Different models of both previous POWER6/POWER7 systems implement one of two possible memory layouts, as follows:

Layout of the memory subsystem in the POWER6/7 systems

Commodity DIMM based configurations

Based on

- commodity DDR2-667 DIMMs in case of POWER6 systems and
- commodity DDR3-1067 DIMMs in case of POWER7 systems

FB-DIMM-based configurations

Based on

- proprietary IBM DDR2-667 FB-DIMMs in case of POWER6 systems and
- proprietary IBM DDR3-1067 FB-DIMMs in case of POWER7 systems

10.3.5 Intelligent memory buffers (2)

Example: Layout of the memory subsystem in the POWER7/POWER7+ [61], [124]

Memory configurations of the POWER7

Commodity DIMM based configurations

Dual integrated DDR3 memory controllers
- High channel and DIMM utilization
- Advanced energy management
- RAS advances

Eight high-speed 6.4 GHz channels
- New low-power differential signalling

New DDR3 buffer chip architecture
- Larger capacity support (32 GB/core)
- Energy management support
- RAS enablement

DDR3 DRAMs

Commodity 240-pin DDR3-1066 DIMMs

Low and midrange server models

E.g. Power 710-760 models

FB-DIMM-based configurations

IBM proprietary 276 pin DDR3-1066 FB-DIMMs

High end server models

E.g. Power 770/780
Power 795 (POWER7 only)

10.3.5 Intelligent memory buffers (3)

Layout of the memory subsystem in the POWER8-1 [114]

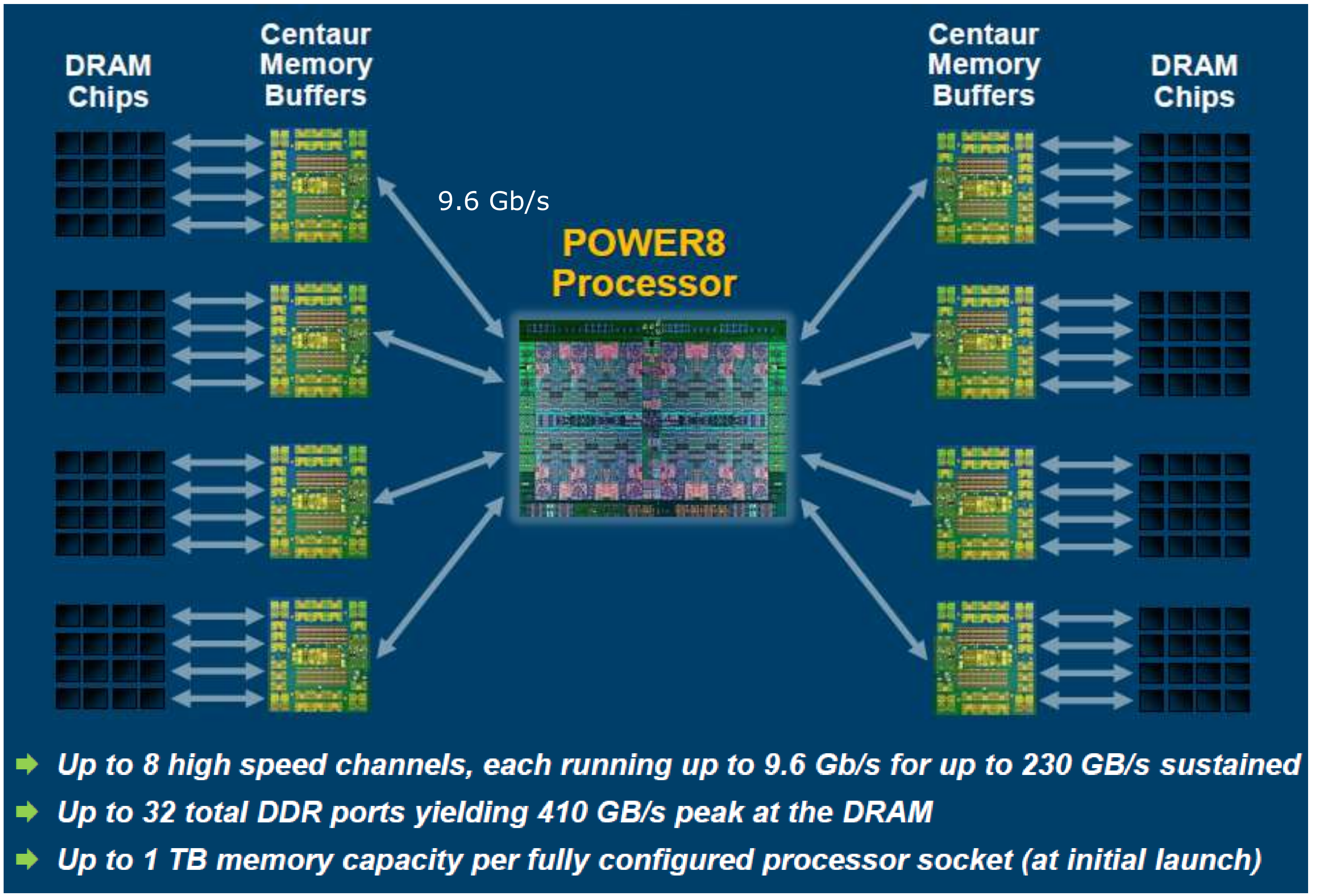
The POWER8 discontinued the FB-DIMM based alternative of the memory subsystem layout and implements only the commodity DIMM based one, nevertheless significantly enhanced vs. the previous POWER7 and POWER7+ to cope with the up to 50 % higher core count.

The enhancement is using an "intelligent" memory buffer, called the Centaur Memory Buffer and increasing the memory speed from DDR3-1066 to DDR3-1600.

The Centaur chip is manufactured in the same 22 nm SOI (Silicon On Insulator) process as the processor chip.

The memory subsystem is implemented with eight memory controllers such that each memory controller is interconnected to a Centaur Memory Buffer through a high speed (9.6 Gb/s) channel, as indicated below.

Layout of the memory subsystem in the POWER8-2 [83]



Implementation of the Centaur Memory Buffers-1 [115]

The Centaur Memory Buffers are implemented typically along with the commodity DDR3-1600 memory chips on tall or short Custom DIMMs (CDIMMs), as seen below.

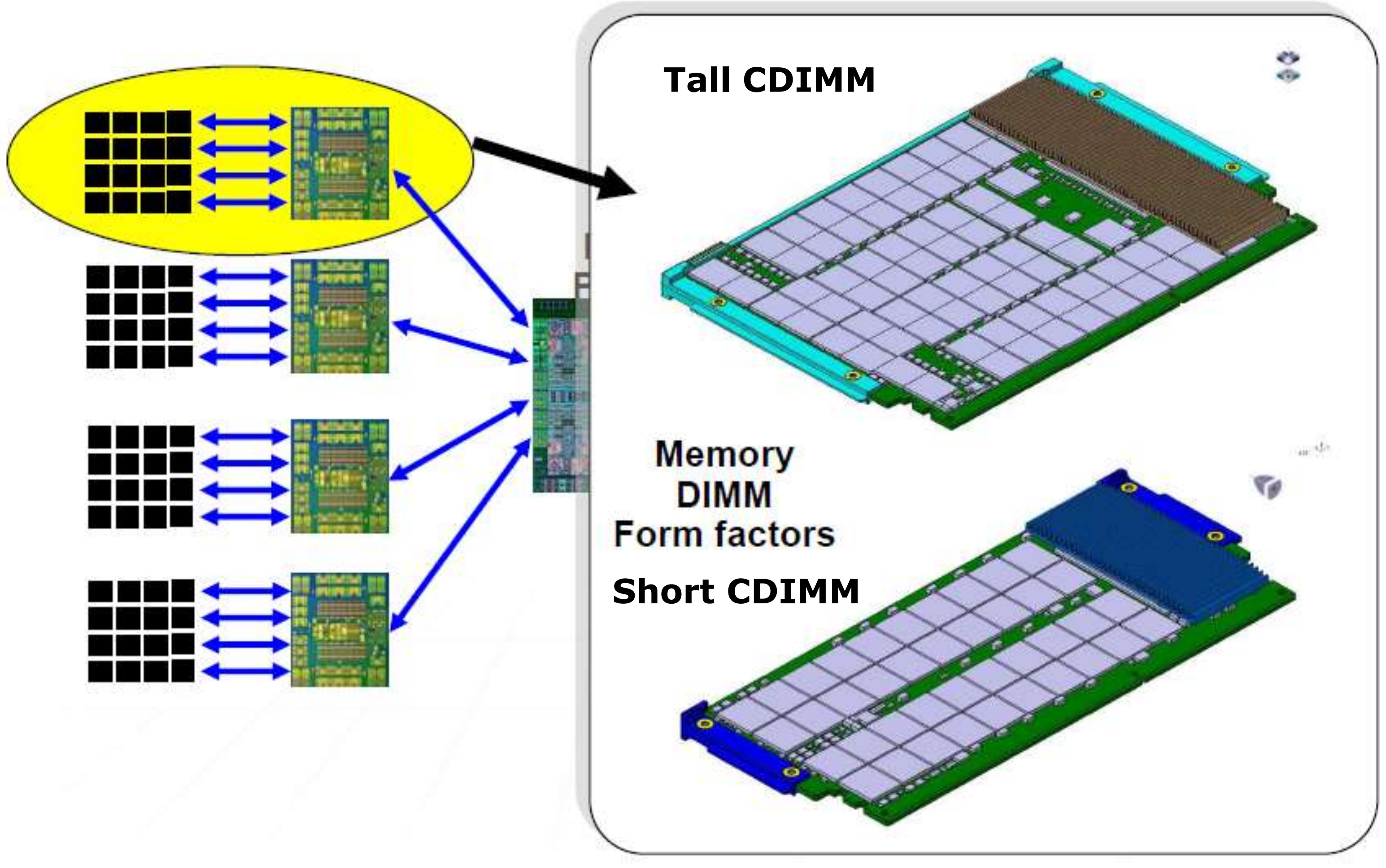
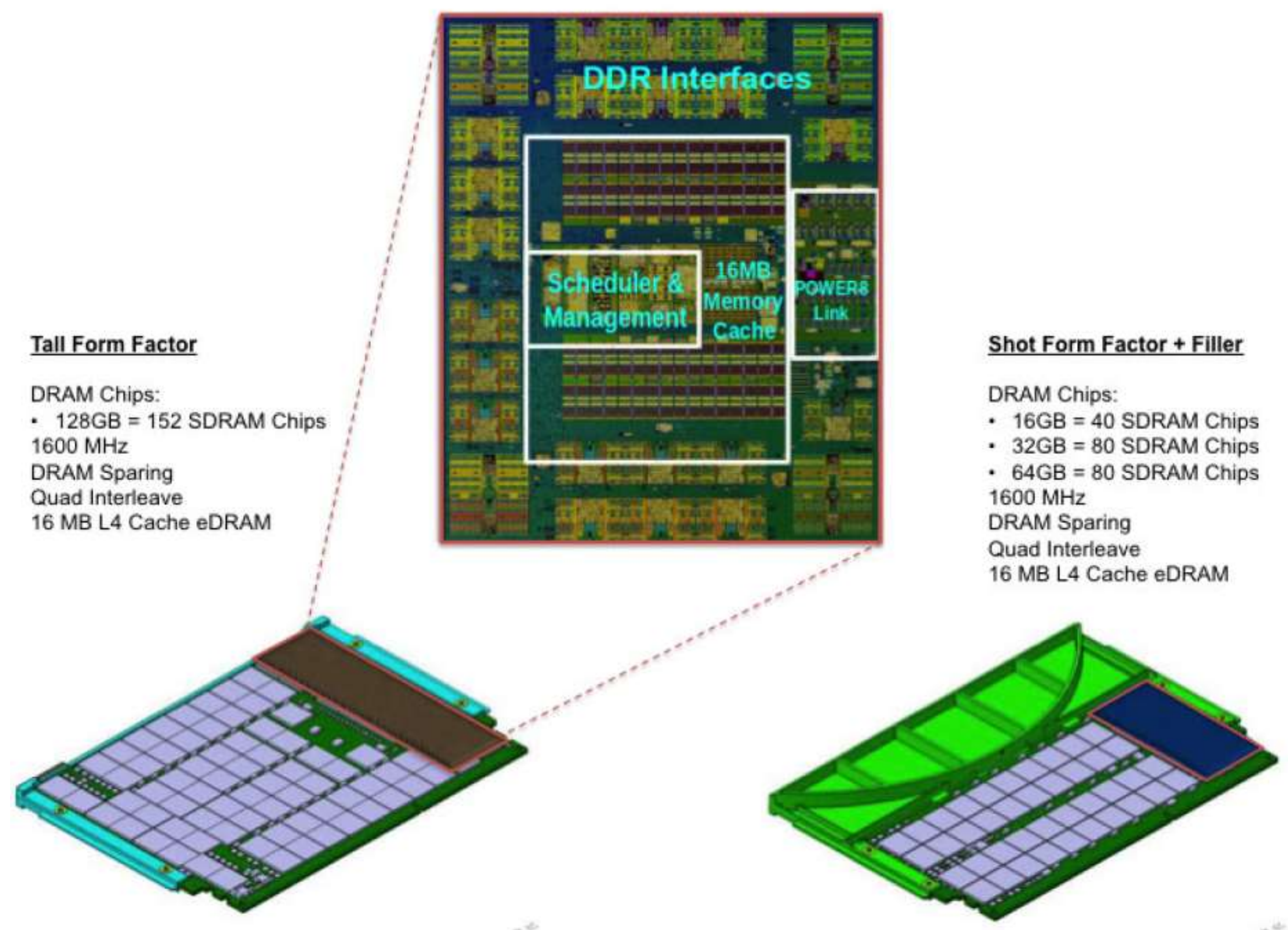


Figure: Implementation of the Centaur Memory Buffers [115]

Implementation of the Centaur Memory Buffers-2 [116]



10.3.5 Intelligent memory buffers (7)

Implementation of the Centaur Memory Buffers-3 [116]

As the Figure indicates CDIMMS exist in two different form factors:

- the tall CDIMMs incorporate 152 memory chips (e.g. 4 Gbit DDR3-1600 chips), whereas
- the short CDIMMs include 40 or 80 memory devices.

Implementation of the Centaur Memory Buffers-4 [3]

Key features of the Centaur Memory Buffer

- Memory technology agnostic high speed interconnect to the processor
- Intelligent memory controller functions
- 16 MB memory cache, termed as the L4 cache
- Industry standard interconnection to the DDR-1600 memory chips

These features will be briefly described next.

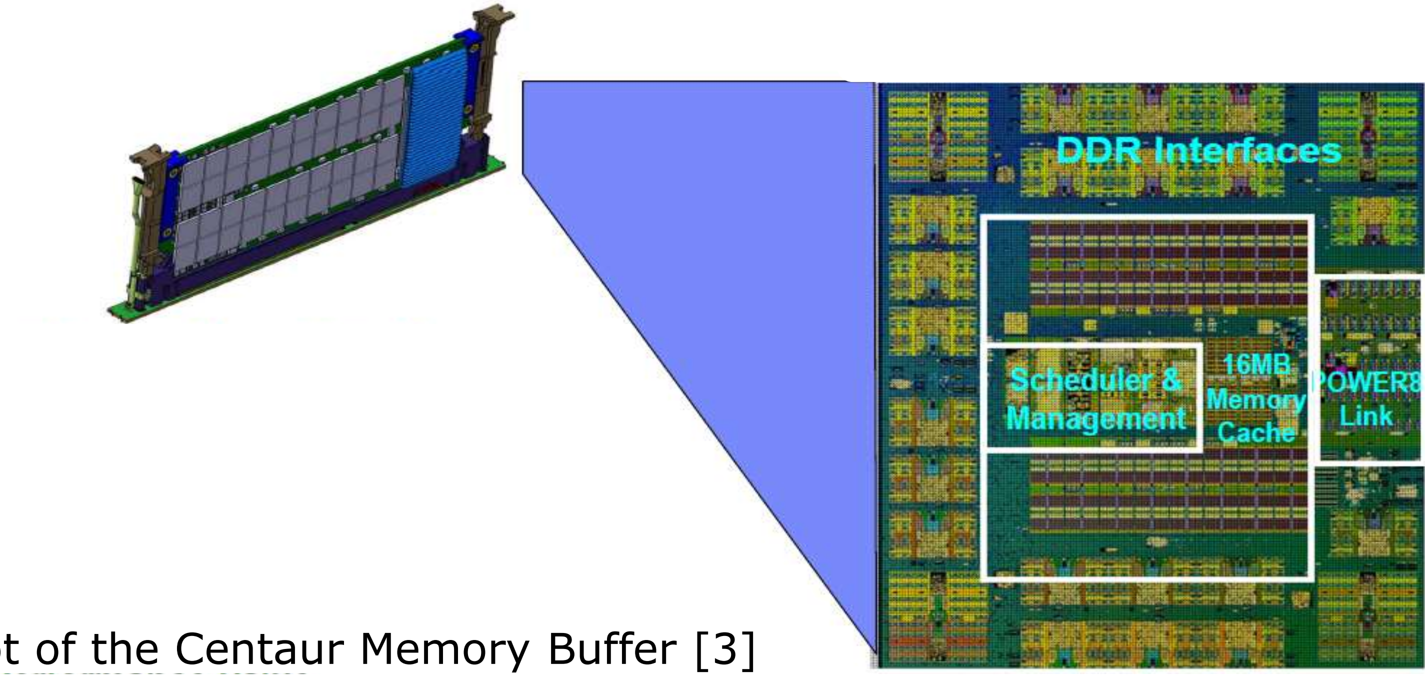


Figure: Die plot of the Centaur Memory Buffer [3]

10.3.5 Intelligent memory buffers (9)

Memory technology agnostic high speed interconnection to the processor [11

- There is a high speed, differential interconnection between the Centaur chip and the processor, providing a 2B wide read and a 1 B wide write sub-channels with a bit-rate of up to 9.6 Gbit/s.
- This results in a total interconnection bandwidth of 3x9.6=28.6 GB/s per interconnection.
- The interconnection channel is agnostic to the memory chip technology (e.g. to DDR3), i.e. it does not have features specific to a given memory technology.

  Requests sent to the Centaur chip are high-level commands, such as cache-line read, cache-line write commands that include also a tag to identify the request.

  This is in contrast to the low-level DDR3 commands sent across the POWER7 memory channel.
- The memory technology agnostic interconnection channel has the benefit that it allows updating the memory technology (e.g. to use DDR4 technology) without modifying the processor chip.

10.3.5 Intelligent memory buffers (10)

Intelligent memory controller functions [114]

- As a substantial innovation ”intelligent” memory controller functions, such as memory traffic scheduling were moved from the processor to the Centaur chip.

  This allows e.g. the reordering of memory requests such as younger memory requests hitting in the L4 cache may pass earlier accesses that do not hit L4.
- Intelligent scheduling also supports prefetching and write optimizations, such as “gathering” write operations that target the same cache line within the L4 cache before writing them to memory in a single write access.

16 MB Memory Cache, termed as the Level 4 (L4) cache [114]

- Each Centaur chip incorporates a 16 MB L4 cache, i.e. a POWER8 processor with 8 Centaur chips provide altogether 128 MB l4 cache.
- The L4 cache is built on eDRAM technology, in the same way as the L3 cache.
- Key benefits of having an L4 cache:
  - Reducing energy consumption by eliminating a number of memory requests.
  - Reducing the overall latency of memory accesses, as cached blocks being in the L4 cache can accessed with up to 55 % lower latency than non-cached blocks.

The resulting cache hierarchy of the POWER8 [117]

Cache	POWER7	POWER7+	POWER8
L1 instruction cache: Capacity/associativity	32 KB, 4-way	32 KB, 4-way	32 KB, 8-way
L1 data cache: Capacity/associativity bandwidth	32 KB, 8-way Two 16 B reads or one 16 B writes per cycle	32 KB, 8-way Two 16 B reads or one 16 B writes per cycle	64 KB, 8-way Two 16 B reads or one 16 B writes per cycle
L2 cache: Capacity/associativity bandwidth	256 KB, 8-way Private 32 B reads and 16 B writes per cycle	256 KB, 8-way Private 32 B reads and 16 B writes per cycle	512 KB, 8-way Private 32 B reads and 16 B writes per cycle
L3 cache: Capacity/associativity bandwidth	On-Chip 4 MB/core, 8-way 16 B reads and 16 B writes per cycle	On-Chip 10 MB/core, 8-way 16 B reads and 16 B writes per cycle	On-Chip 8 MB/core, 8-way 32 B reads and 32 B writes per cycle
L4 cache: Capacity/associativity bandwidth	N/A	N/A	Off-Chip 16 MB/buffer chip, 16-way Up to 8 buffer chips per socket

Industry standard interconnection to the DDR-1600 memory chips [115]

Each Centaur Memory Buffer supports four 10 Byte wide DRAM ports, consisting of 72 bits of data, 8 ECC bits as well as 8 bits of spare data to industry standard DDR3-1600 memory chips, as indicated below.

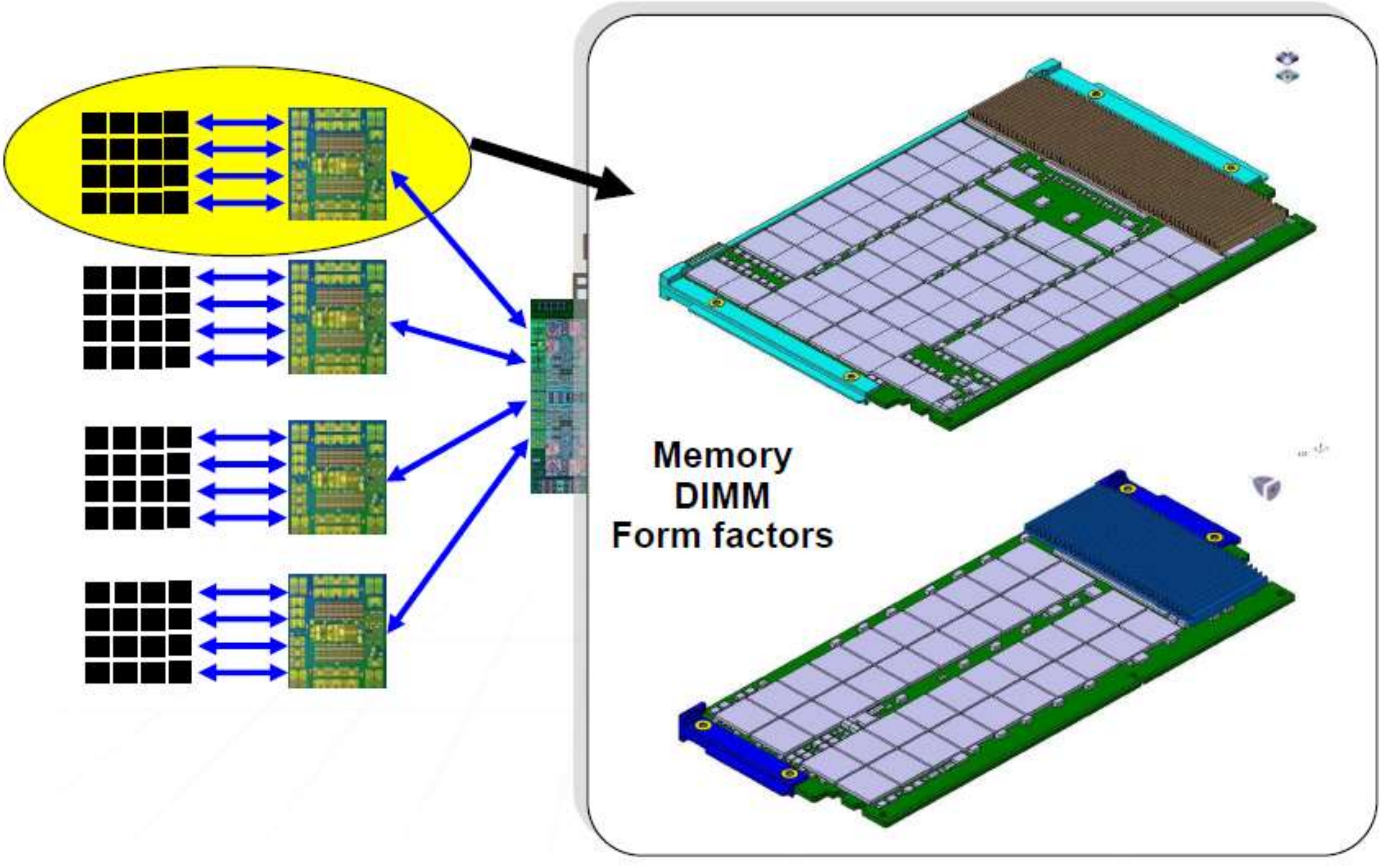
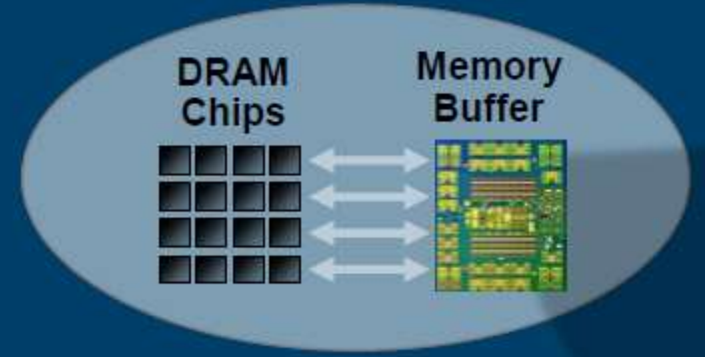
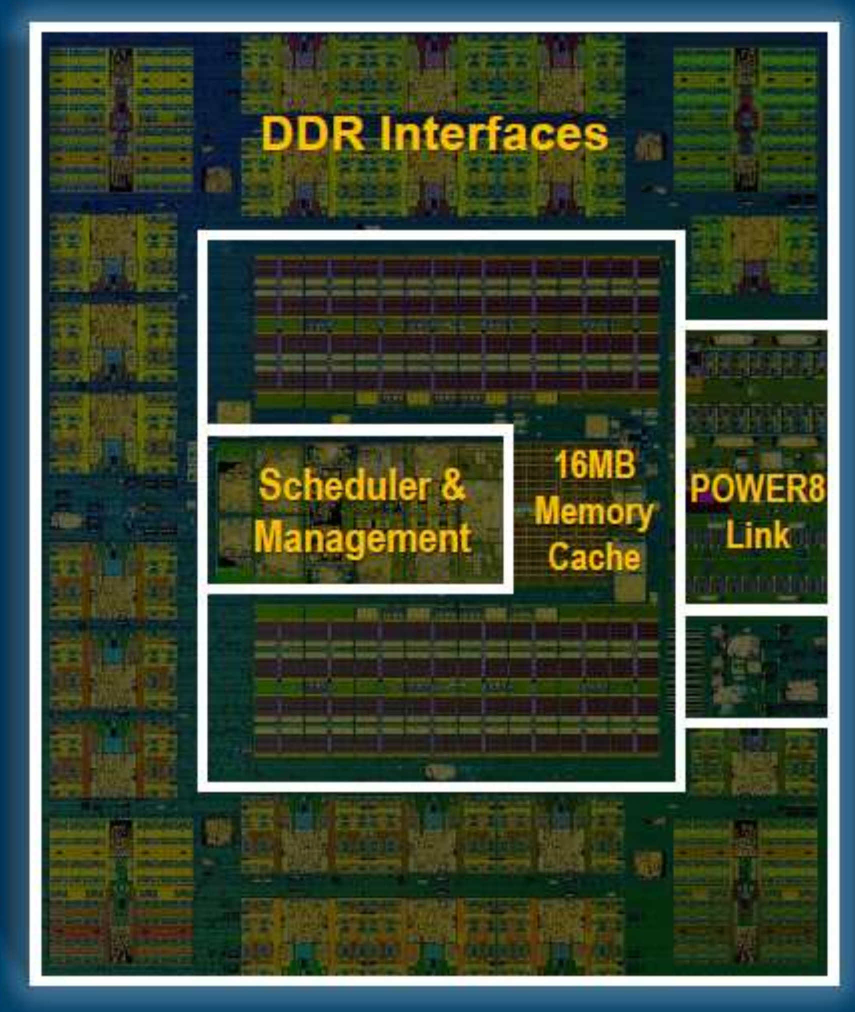


Figure: The Centaur chips providing four 10 Byte wide DDR3-1600 ports [115]

10.3.5 Intelligent memory buffers (14)

Die plot of the Centaur Memory Buffer [83]





Intelligence Moved into Memory

- Scheduling logic, caching structures
- Energy Mgmt, RAS decision point
  - Formerly on Processor
  - Moved to Memory Buffer

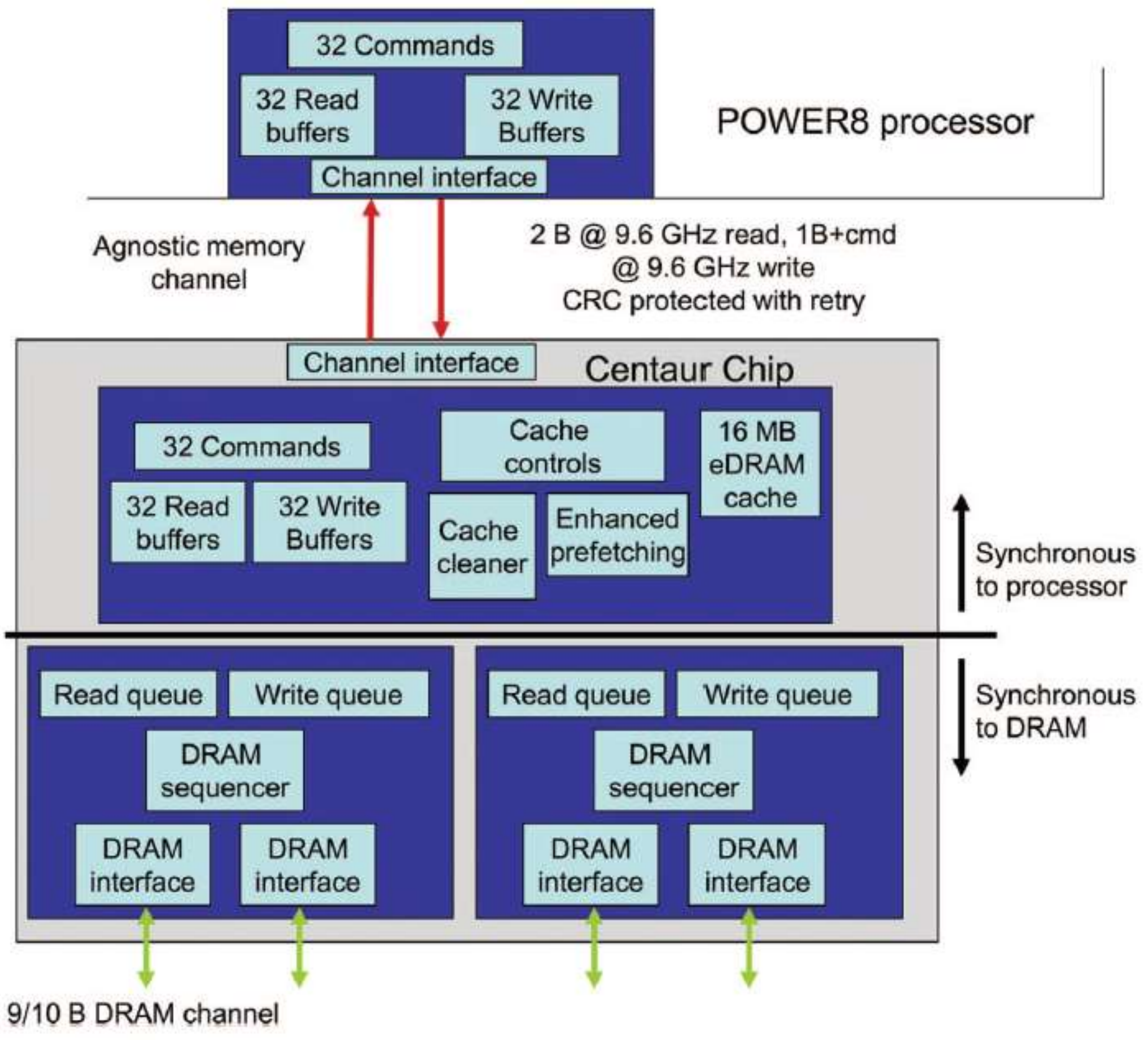
Processor Interface

- 9.6 GB/s high speed interface
- More robust RAS
- "On-the-fly" lane isolation/repair
- Extensible for innovation build-out

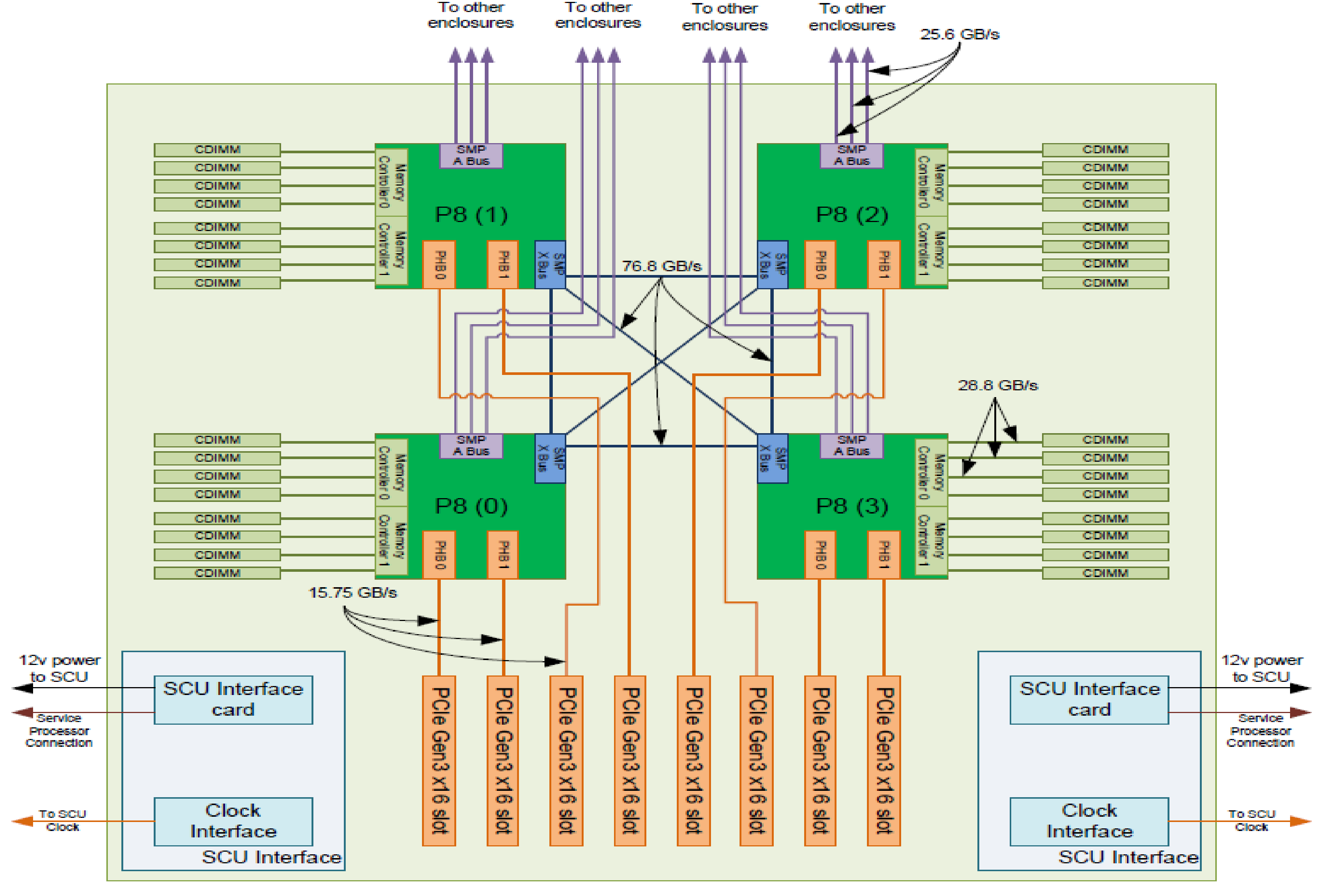
Performance Value

- End-to-end fastpath and data retry (latency)
- Cache → latency/bandwidth, partial updates
- Cache → write scheduling, prefetch, energy
- 22nm SOI for optimal performance / energy
- 15 metal levels (latency, bandwidth)

Block diagram of the Centaur Memory Buffer [114]



Example: Block diagram of a four processor node E870 server [116]



Memory feature evolution in IBM's POWER line

Model	Tech.	Intro.	No. of cores	fc up to	SMT	No., speed and width of high-speed channels	FB-DIMM option	Mem. ports up to	Max. chan. limited BW/ proc.	-Max. BW/fc/ /core (byte /cycle)
POWER3-II	250 nm	1999	1	0.45 GHz	No	2@100 Kbits/s 8B R/W	No	2xDDR-100	1.6 GB/s	3.5
POWER4	180 nm	2001	2	1.3 GHz	No	8@400 Kbit/s 4B R/W	No	8xDDR-200	12.8 GB/s	4.9
POWER5	130 nm	2004	2	1.9 GHz	2-way	4@1066 Kbit/s 4B R/2B W	No	8xDDR-533	25.6 GB/s	6.8
POWER5+	90 nm	2005	2	2.3 GHz	2-way	4@1066 Kbit/s 4B R/2B W	No	8xDDR2-533	25.6 GB/s	5.5
POWER6	65 nm	2007	2	5.0 GHz	2-way	8@2.67 Gbit/s 2B R/1B W	Y	8xDDR2-667	64.0 GB/s	6.4
POWER7	45 nm	2010	8	4.42 GHz	4-way	8@6.4 Gbit/s 2B R/1B W	Y	16xDDR3-1066	153.6 GB/s	4.4
POWER7+	32 nm	2013	8	4.42 GHz	4-way	8@6.4 Gb/s 2B R/1B W	Y	16xDDR3-1066	153.6 GB/s	3.9
POWER8	22 nm	2014	12	4.35 GHz	8-way	8@9.6 Gbit/s 2B R/1B W	No	32xDDR3-1600	230,4 GB/s	4.4

Note that clock rates and memory speeds are representative figures that vary in actual models in a wide range.

10.3.6 CAPI

10.3.6 The Coherent Accelerator Processor Interface (CAPI) -1

As its name indicates, CAPI provides a coherent interface for accelerators (see below).

It allows an FPGA or ASIC accelerator to connect coherently to the SMP interconnect via the PCIe bus.

Adapter in normal I/O mode

Core
Core
Mem Ctlr
Core
Core
PHB
Coherency Domain
PCIe

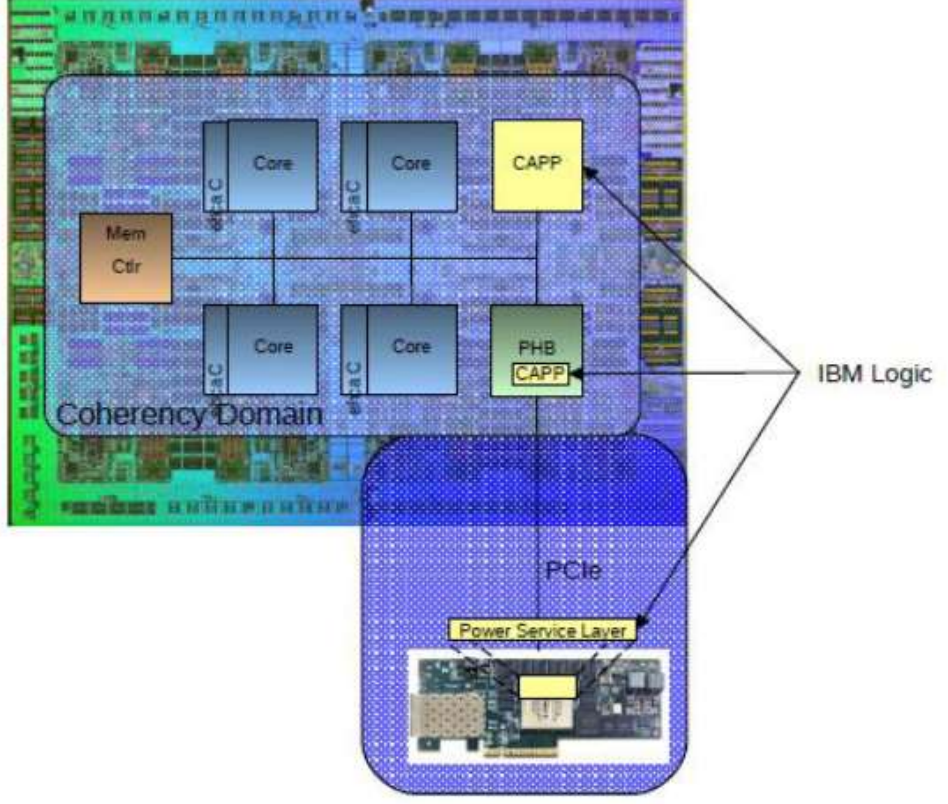
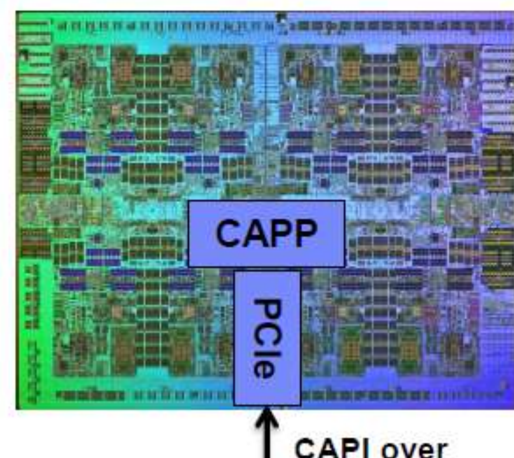


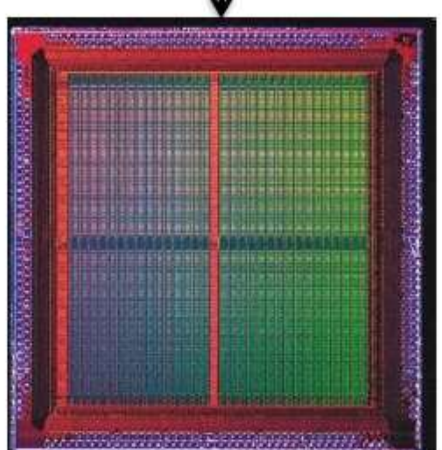
Figure: Concept of CAPI as a coherent accelerator interface [132]

Principle of attaching an accelerator over CAPI to a POWER8 processor [141]

Power Processor



CAPI over PCIe

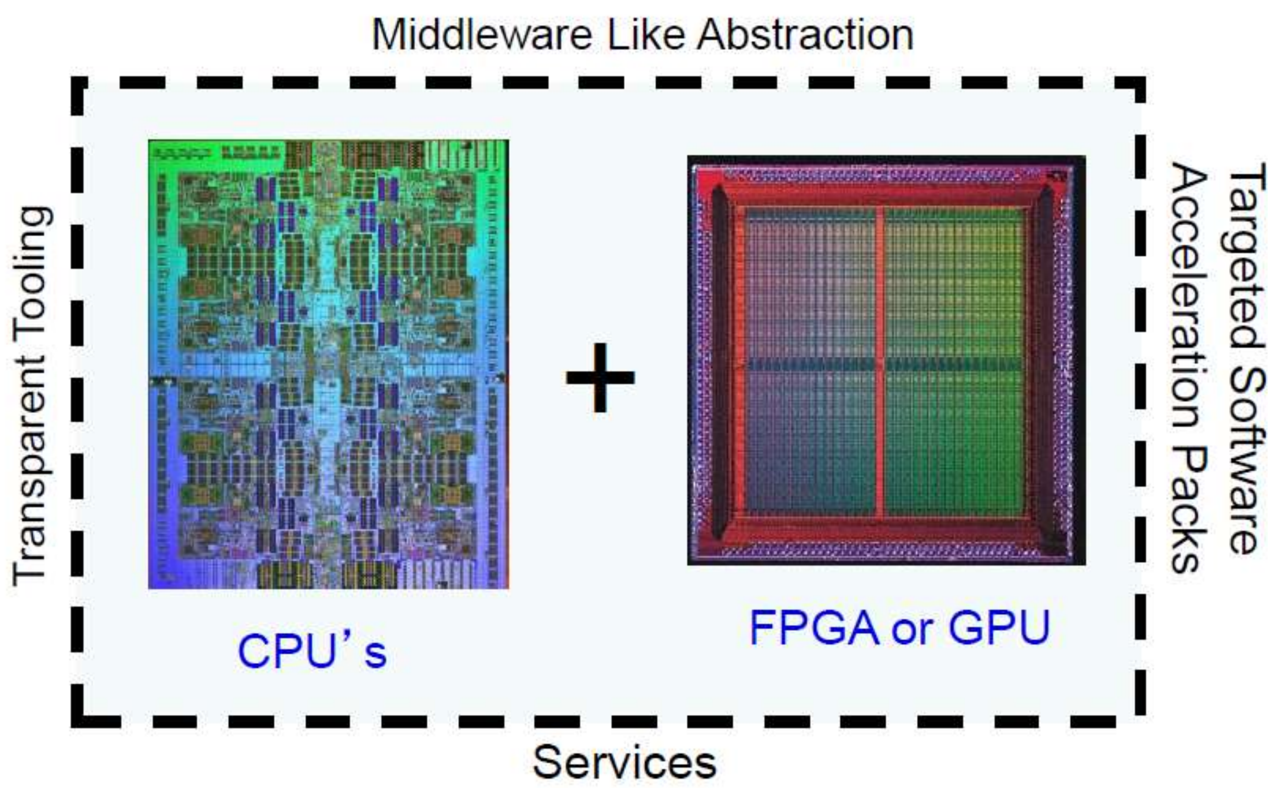


Coherently Attached Device

- **Coherent Attached Processor Proxy (CAPP) in processor**
  - Unit on processor that extends coherency to an attached device
  - On processor directory responds on behalf of off-chip device (Filtering snoops)
- **Coherency protocol tunneled over standard PCIe**
  - Eliminates the need for special I/Os and protocol logic
    CAPI utilizes standard Posted Write and Non-posted Reads
  - Reduces the complexity and bandwidth requirements of the attached device
- **Enables attached device to be a peer to the processor**
  - Simplifies programming model between application
  - Enables device to use same effective address as application running in processor
  - Eliminates the cumbersome I/O Device Driver requirements

    Pinned memory not required

Main options to use an accelerator [141]

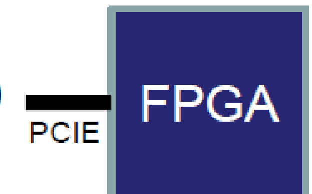


- Strong Cores for ***Serial*** Codes
- Runs Traditional & Legacy Software
- Runs OS (Security, Virtualization, etc)

- Extreme ***Parallelism*** available
- Targeted Software Accelerator packs
- IP Base Libraries
- Customer IP
- Reconfigurable Nature fights Commoditization

FPGA (Field Programmable Gate Array) [141]

- It's a re-programmable chip
- It can run fast (cycle times of 250 – 500 Mhz or more)
- It has Industry Standard Interfaces like PCI-E Gen3
- The Major FPGA Suppliers, Altera and Xilinx, are OpenPOWER Foundation members



Source code for FPGAs has traditionally been written in RTL* (VHDL** or Verilog). Now, we also have OpenCL, a more programmer friendly language.

gzip

Encrypt

Monte Carlo

FPGA Library

*RTL = Register Transfer Level
VHDL = VHSIC* Hardware Description Language
***VHSIC = Very High Speed Integrated Circuit

© 2015 IBM Corporation

- Strong Cores for ***Serial*** Codes
- Runs Traditional & Legacy Software
- Runs OS (Security, Virtualization, etc)

- Extreme ***Parallelism*** available
- Targeted Software Accelerator packs
- IP Base Libraries
- Customer IP
- Reconfigurable Nature fights Commoditization

10.3.6 CAPI (5)

When to Use FPGAs? [141]

- Extreme Parallelism
- Bit-level operations
- Variable-precision floating point
- Power-Performance Advantage
- FPGAs are not frequency or power limited yet
- 3D integration has great potential
- Dynamic reconfiguration
- Flexibility for application tuning at run-time vs. compile-time

When to Use GPGPUs? [141]

- Extreme FLOPS & Parallelism
- Double-precision floating point leadership
- Hundreds of GPGPU cores
- Programming Ease & Software Group Interest
- CUDA & extensive libraries
- OpenCL
- Start w/PCIe gen3 x16 and then move to NVLink
- Lots of existing use-cases to build on

10.3.6 CAPI (7)

The Coherent Accelerator Processor Interface (CAPI) -2 [119]

- Accelerators, like GPUs, FPGAs or ASICs can speed up applications significantly by implementing algorithms directly in hardware, but their integration into a multicore or SMP configuration implies notable software overhead to share data with threads running on CPUs.

  Such an overhead occurs for example, when a data set needs to be loaded from the main memory into the local memory of the GPU before processing or when the result needs to be reshuffled into the main memory.
- CAPI is a solution to this heterogeneous system inefficiency by providing a coherent low-overhead interface for the integration of PCIe-based accelerators into POWER8 systems.
- CAPI was announced along with the introduction of the POWER8 processor in 4/2014 and launched later, in 10/2014.

The Coherent Accelerator Processor Interface (CAPI) -3

- CAPI
  - makes use of virtual addressing and
  - provides hardware managed cache coherency.
- Virtual addressing allows
  - the accelerator to work with the same memory addresses as the processor,
  - pointers used by the accelerator to be de-referenced in the same way as in the host application.

  This removes OS and device driver overhead.
- Hardware managed cache coherency enables
  - the accelerator to participate in "locks" like a normal thread.

  This lowers latency in I/O communication.

When you want to access the data/value in the memory that the pointer points to - the contents of the address with that numerical index - then you ***dereference*** the pointer.

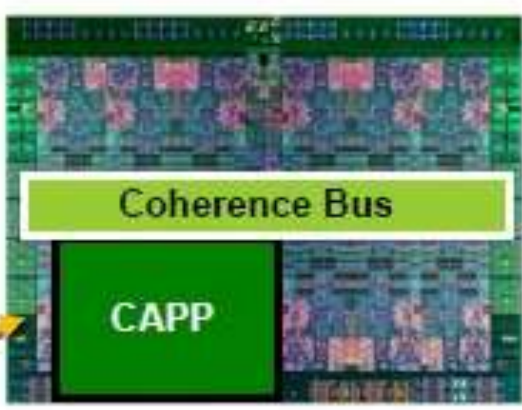
Main components of operating CAPI [142]

Virtual Addressing

- Accelerator can work with same memory addresses that the processors use
- Pointers de-referenced same as the host application
- Removes OS & device driver overhead

Hardware Managed Cache Coherence

- Enables the accelerator to participate in "Locks" as a normal thread Lowers Latency over IO communication model



PCIe Gen 3
Transport for encapsulated messages

Custom Hardware Application FPGA or ASIC

Processor Service Layer (PSL)

- Present robust, durable interfaces to applications
- Offload complexity / content from CAPP

Customizable Hardware Application Accelerator

- Specific system SW, middleware, or user application
- Written to durable interface provided by PSL

Components of CAPI-2

CAPI incorporates three essential components:

a) PSL: Power Service Layer
b) PHB: PCIe Host Bridge
c) CAPP: Coherent Accelerator Processor Proxy unit

as shown below.

These components will briefly outlined next.

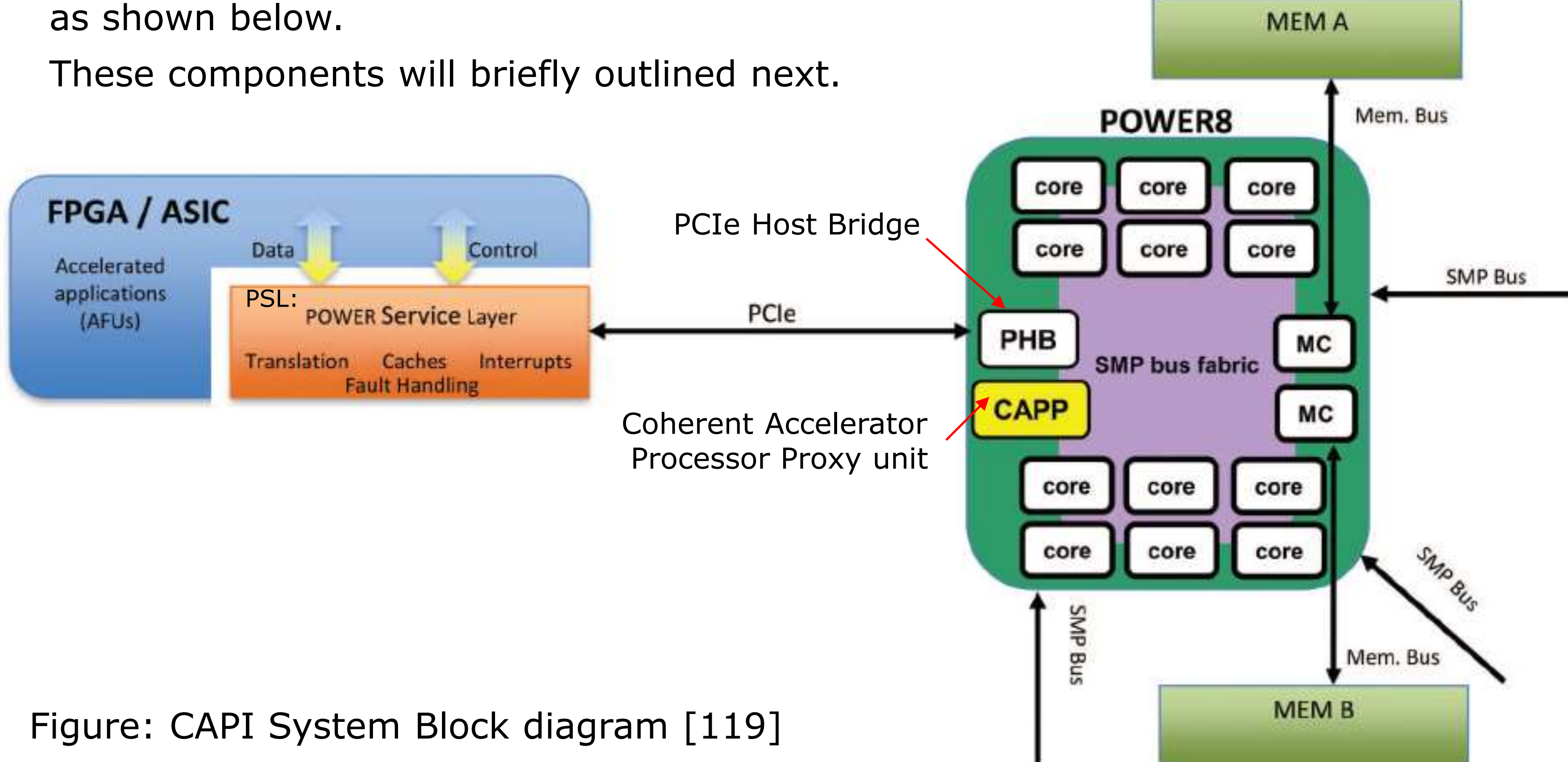
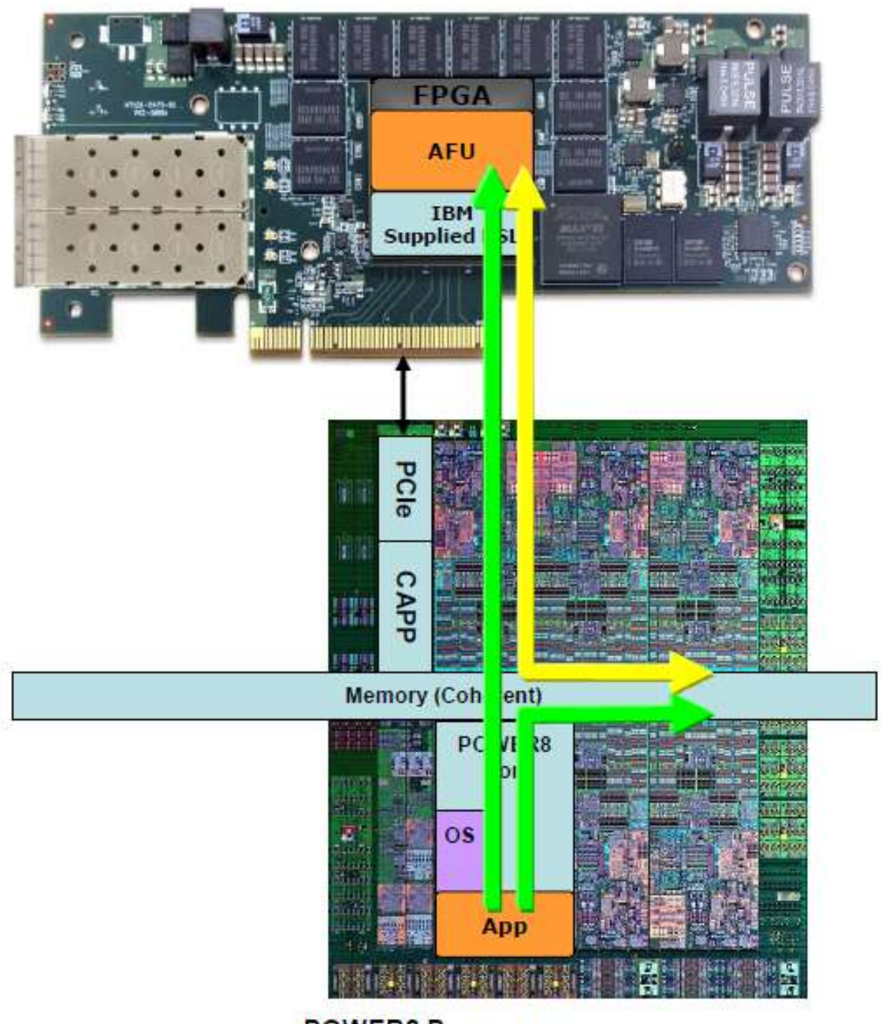


Figure: CAPI System Block diagram [119]

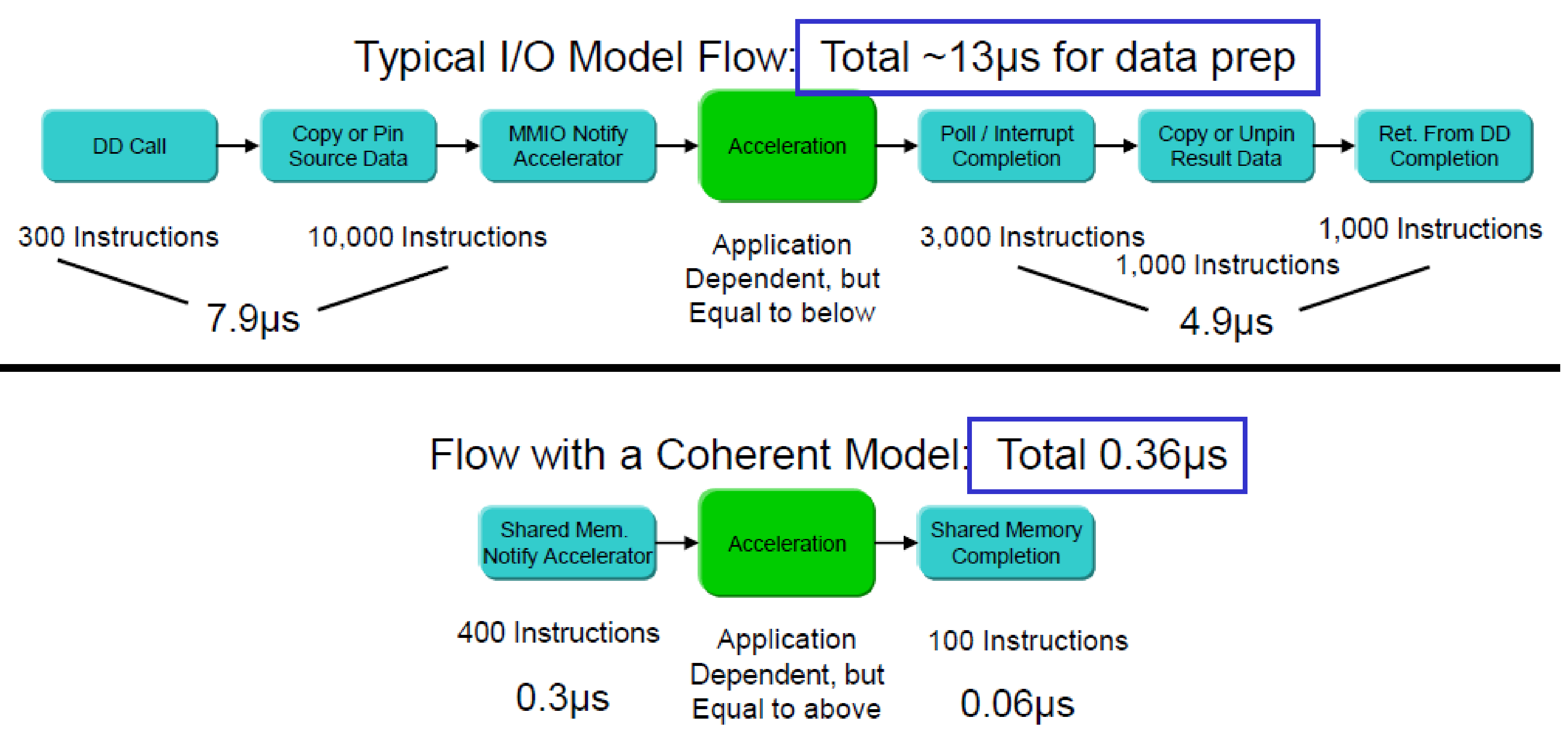
Principle of operation of CAPI [141]



- Proprietary hardware to enable coherent acceleration
- Operating system enablement
  - Ubuntu LE
  - Libcxl function calls
- Customer application and accelerator

- Application sets up data and calls the accelerator functional unit (AFU)
- AFU reads and writes coherent data across the PCIe and communicates with the application
  - PSL cache holds coherent data for quick AFU access

Connecting an accelerator traditionally or via CAPI [141]



DD: Device Driver

10.3.6 CAPI (13)

a) The Power Service Layer (PSL) [118], [119]

- It resides on the AFU (e.g. FPGA/ASIC) alongside of the acceleration engine.
- The PSL provides a straightforward command ⟹ data buffer ⇨ command response interface to the accelerator, which grants access to coherent memory.
- The PSL works in concert with the CAPP unit across a PCIe connection.
- AFUs use effective addresses to reference memory, with address translation provided by an MMU (Memory Management Unit) in the PSL.
- The PSL may also generate interrupts on behalf of AFUs to signal AFU completion, or to require a system service when a translation fault occurs.

AFU: Accelerator Function Unit

10.3.6 CAPI (14)

b) The PCIe Host Bridge (PHB)

- The PHB provides connectivity between the on-chip coherence and data interconnect and the PCIe Gen3 I/O links.
- The POWER8 processor chip integrates 3 PHBs per chip, with a maximum of 16 PCIe 3.0 lanes per PHB, resulting in a maximum bidirectional bandwidth of 32 GB/s per PHB.

c) The Coherent Accelerator Processor Proxy (CAPP) unit [114] -1

The CAPP unit, in conjunction with the PHB (PCIe Host Bridge) acts as a memory coherence, data transfer, interrupt and address translation agent on the on-chip coherence and data interconnect, as indicated in the Figure below.

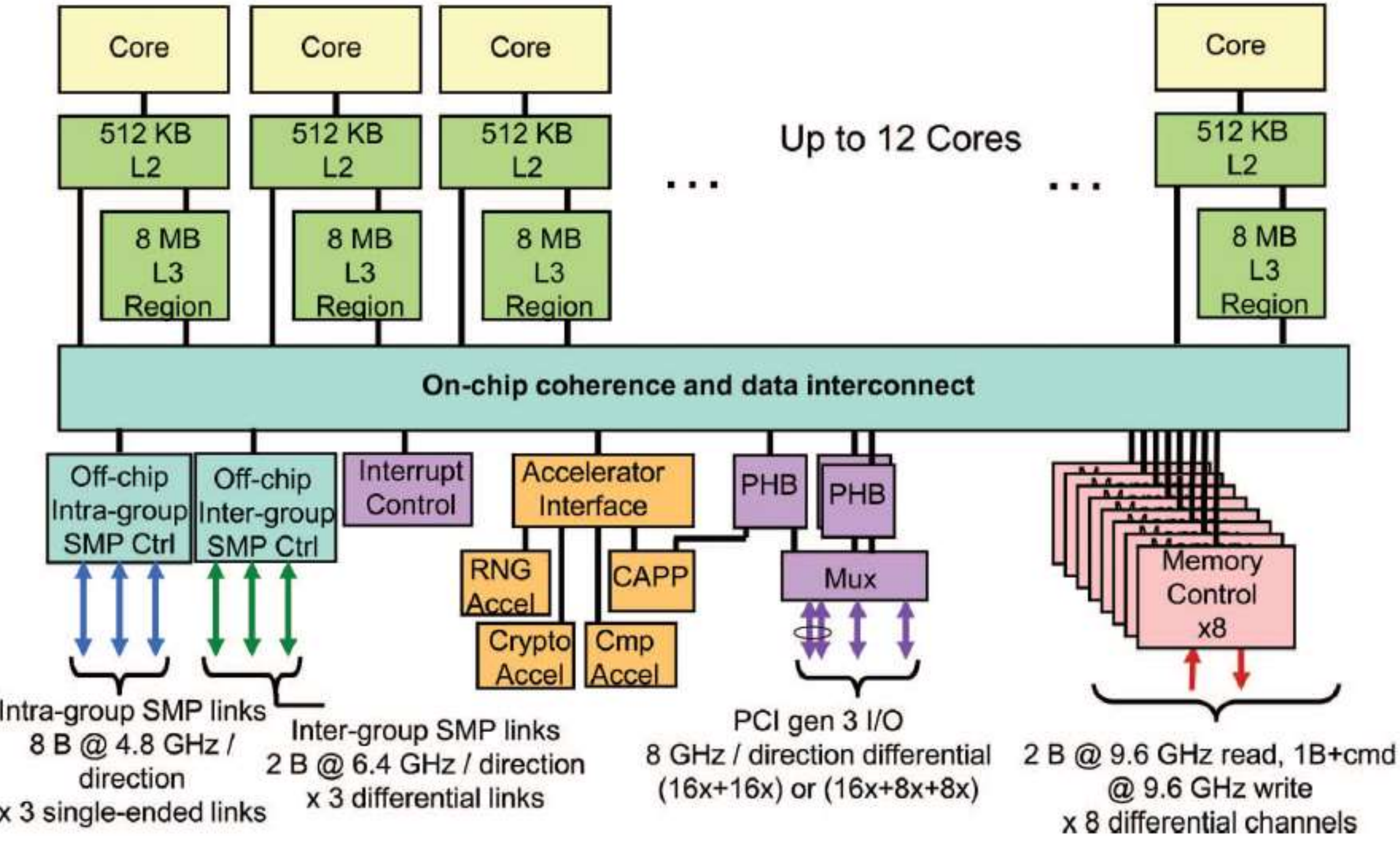


Figure: The on-chip coherence and data interconnect [114]

The Coherent Accelerator Processor Proxy (CAPP) unit [114] -2

The CAPP unit maintains a directory of all cache lines held by the off-chip accelerator, allowing it to act as the proxy that maintains architectural coherence for the accelerator across its virtual memory space.

10.3.6 CAPI (17)

Principle of operation of CAPI [118], [119]

- CAPI enables attaching an accelerator over the I/O interface, i.e. over the PCIe 3.0 bus, as a coherent CPU peer to the POWER8 system.
- In this case the accelerator accesses memory using the same virtual address space as the application that calls it.
- Accelerators access system memory through load and store requests to user space effective addresses, with address translation provided by an MMU (Memory Management Unit) in the PSL.
- In order to provide coherent access to system memory, CAPP and PSL each contain a directory of cache lines used by the AFUs.
- The CAPP snoops the interconnection fabric on behalf of the PSL, accesses its local directory, and responds to the fabric with latency that is the same as other caches on the chip.
- In this way, the insertion of an off-chip coherent accelerator does not affect critical system performance parameters such as cache snoop latency.

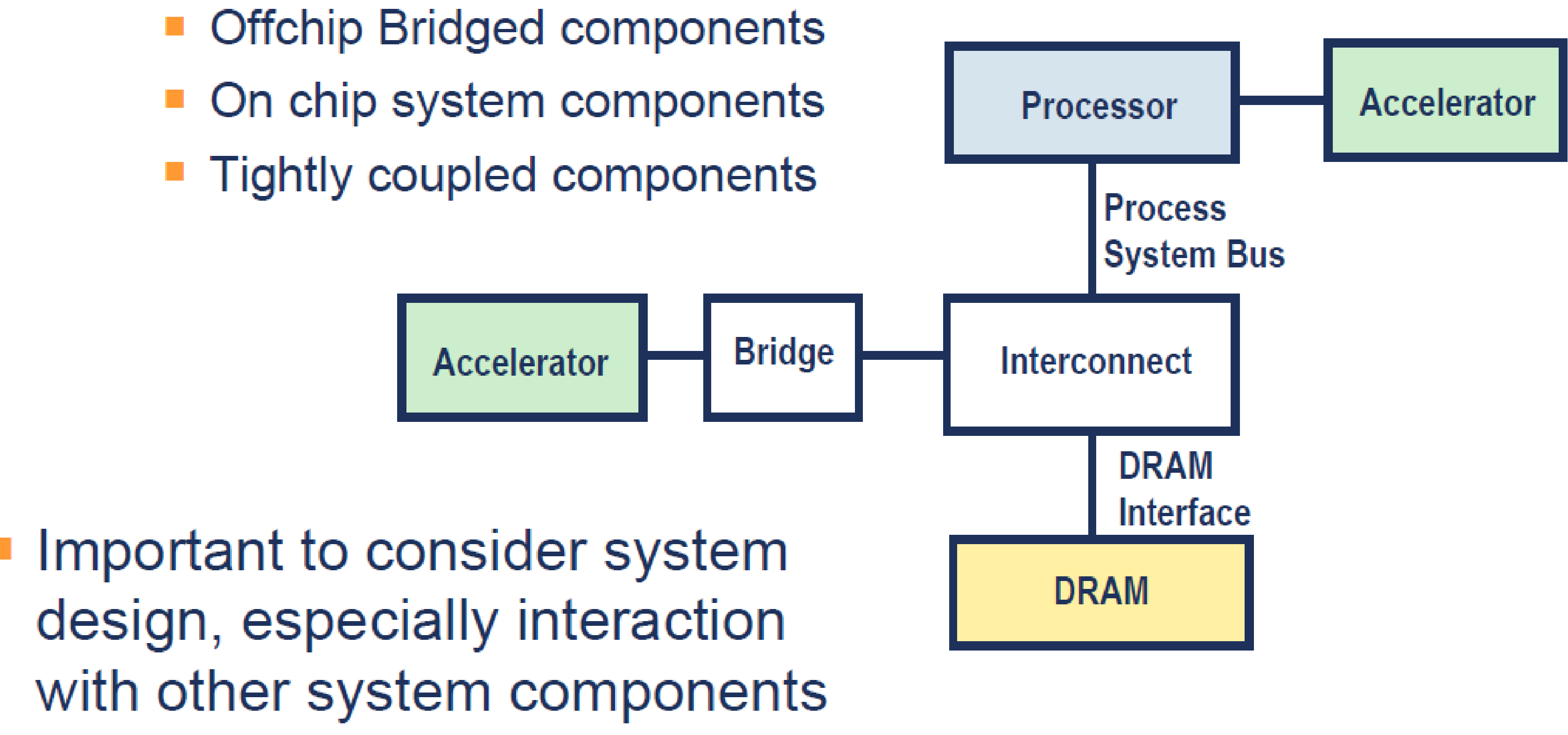
10.3.6 CAPI (18)

Software components of CAPI [118]

- CAPI needs operating-system kernel extensions and library functions created specifically for CAPI.
- The operating-system kernel extensions became available to all POWER-Linux distributions starting with Canonical's Ubuntu.
- A client application uses a CAPI library called **"libcxl**," which provides a set of functions for the application to connect, call, communicate, and disconnect with an available CAPI device.

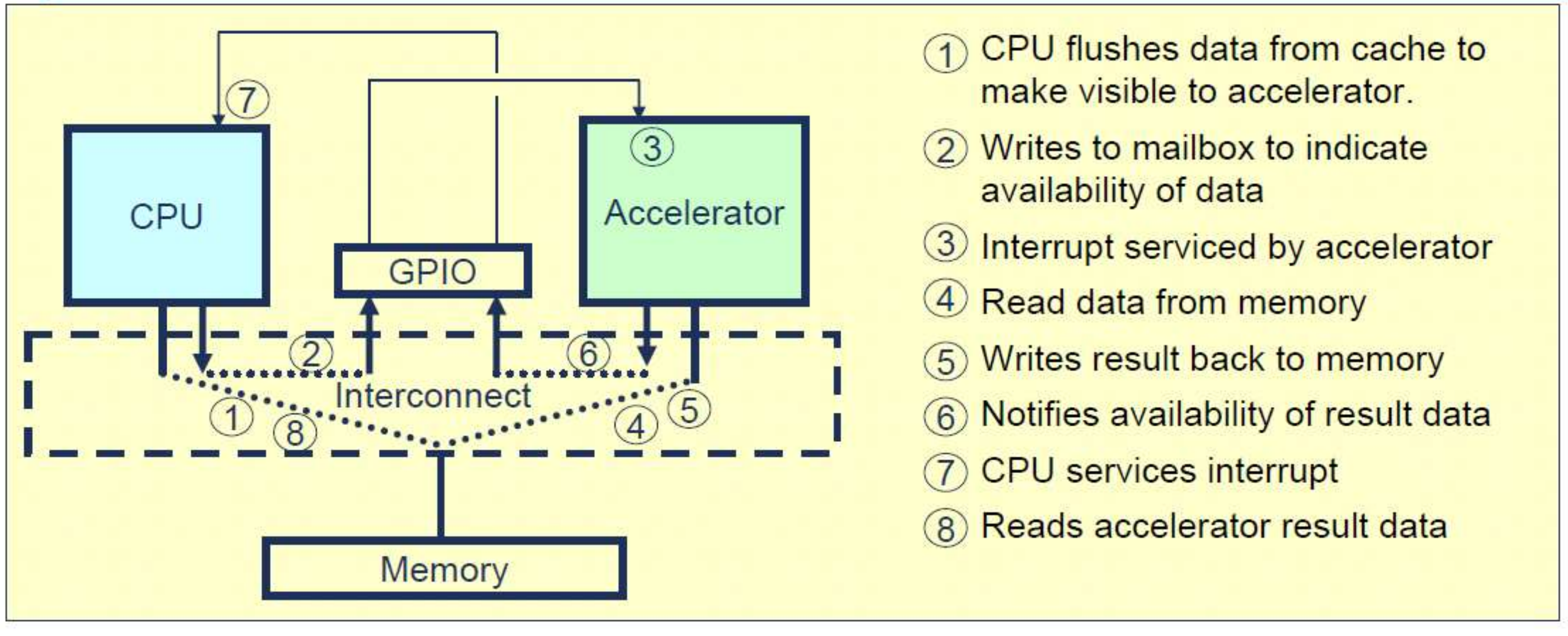
Remark – ARM's concept for attaching accelerators [143] -1

- System performance is not only defined by the processor
  - Speed/width and latency of DRAM and processor system interface
  - Connectivity and efficiency interacting with system components
    - Offchip Bridged components
    - On chip system components
    - Tightly coupled components



- Important to consider system design, especially interaction with other system components such as DMA and accelerators

ARM's concept for attaching accelerators [143] -2
Traditional integration of accelerators



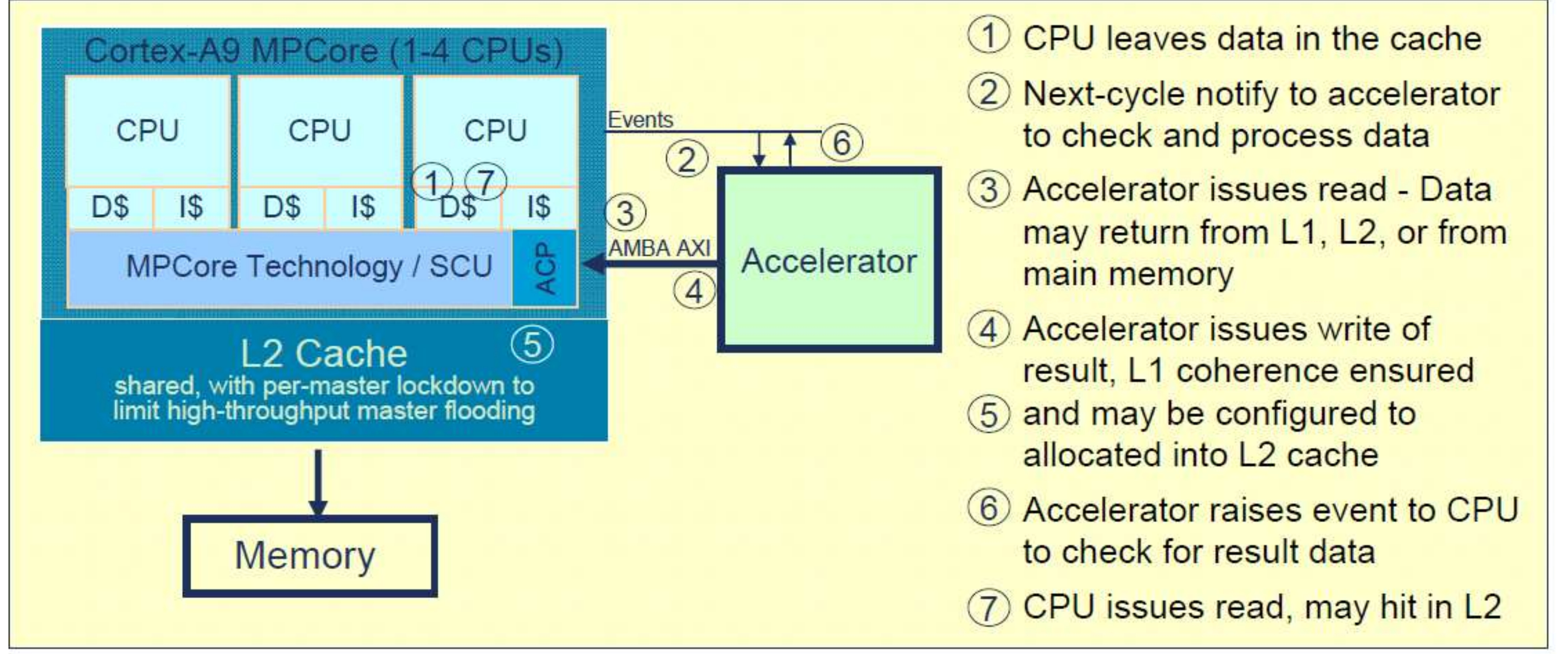
- Analysis of traditional SoC accelerator SoC integration
  - Inefficient usage of CPU cache
  - Significant performance and power implications from data movement
  - High signalling latencies due to mailbox access and interrupt latencies

ARM's concept for attaching accelerators [143] -3

Accelerator Coherence Port (ACP) as introduced into the Cortex-A9 MPCore

Announced in 10/2007, first devices about 2009/2010.

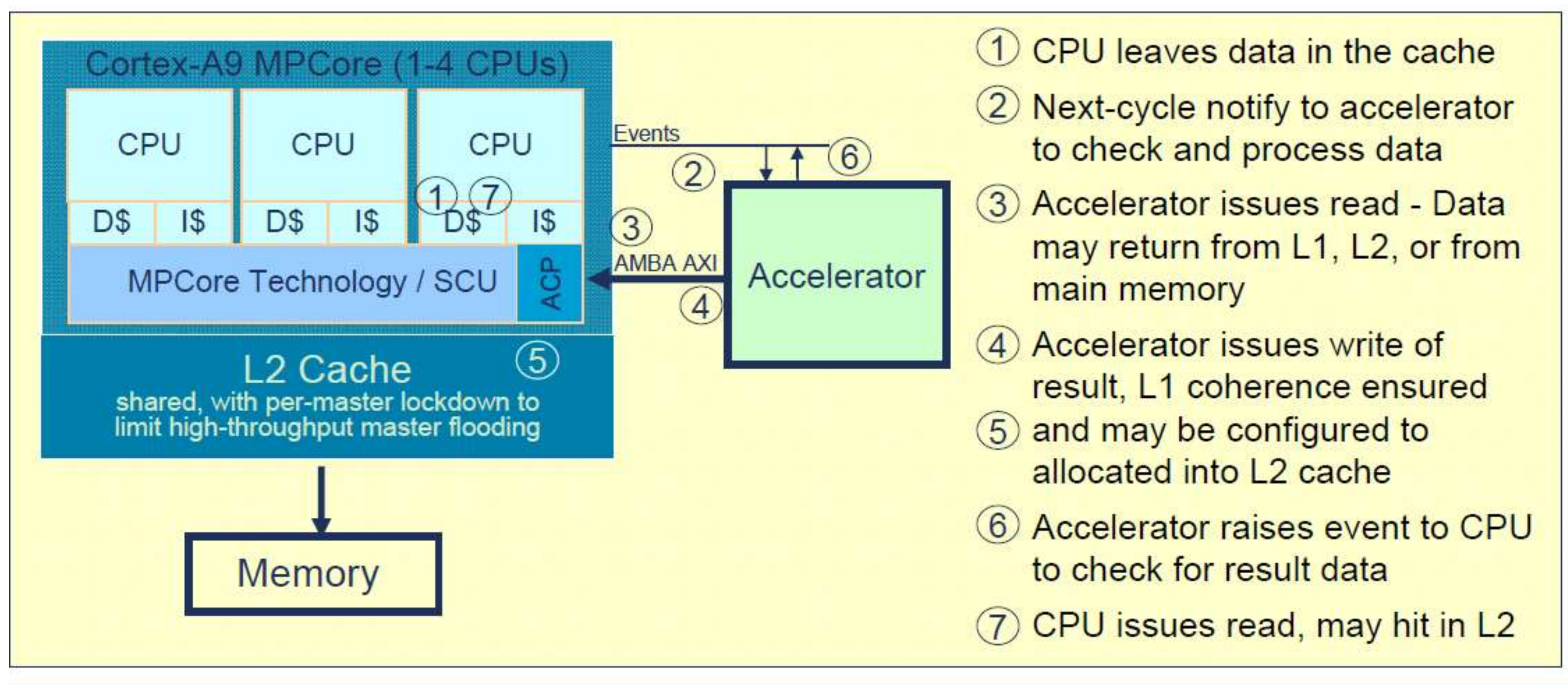
- Simplified software and reduces cache flush overheads
- Accelerators gain access to CPU cache hierarchy, increasing system performance and reducing overall power
- Uses AMBA® 3 AXI™ technology for compatibility with standard un-cached peripherals and accelerators



ARM's concept for attaching accelerators [143] -4

Accelerator Coherence Port (ACP) as introduced into the Cortex-A9 MPCore

Announced in 10/2007, first devices about 2009/2010.



10.3.7 Replacing the GX IO bus by the PCIe Gen3 bus

10.3.7 Replacing the GX system bus by the PCIe Gen3 bus

- IBM introduced the GX bus along with their POWER4 processor in 2001.
- The GX bus is a high-frequency, single-ended, unidirectional, 4-byte wide point to-point bus [13], [16], as briefly outlined in the Section 2.2.9.

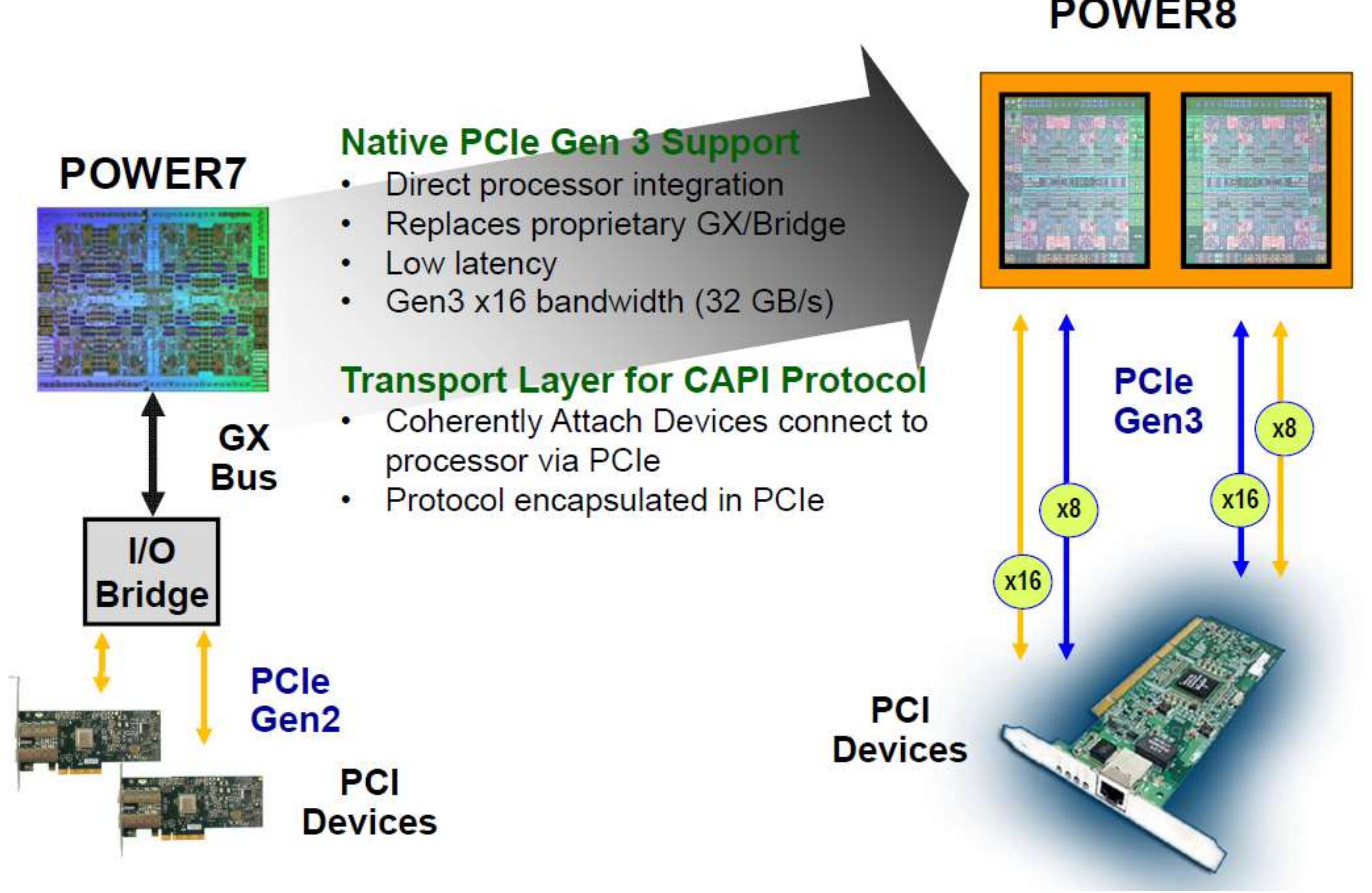
Evolution of the GX bus in IBM's POWER line

Based on available IBM literature, the Table below illustrates the evolution of the GX bus.

Model	Designation of the GX bus	Speed	Exemplary total bandwidth per GX bus
POWER4	GX	1/3 fc e.g. 400 MHz for 1.2 GHz	3.2 GB/s
POWER5	GX+	1/3 fc e.g. 700 MHz for 2.1 GHz	5.6 GB/s
POWER6	GX++	¼ fc e.g. 1.05 GHz for 4.2 GHz	8.4 GB/s
POWER7	GX+/GX++	1.25 GHz 2.50 GHz	10 GB/s/ 20 GB/s
POWER8	--	--	--

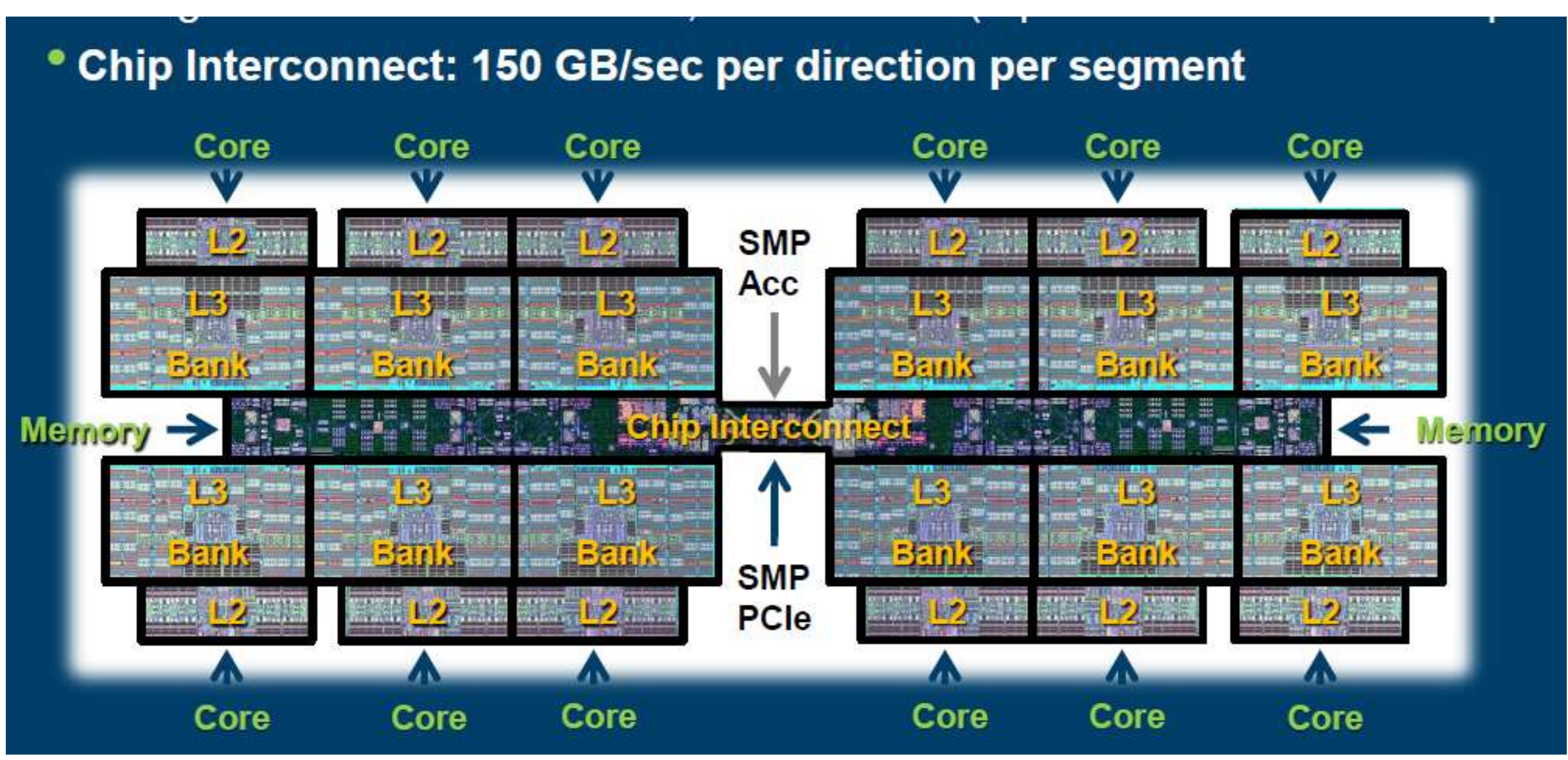
Table: Evolution of the features of the GX bus in IBM's POWER line

Replacing the GX bus in the POWER8 by the PCIe Gen3 bus [3]



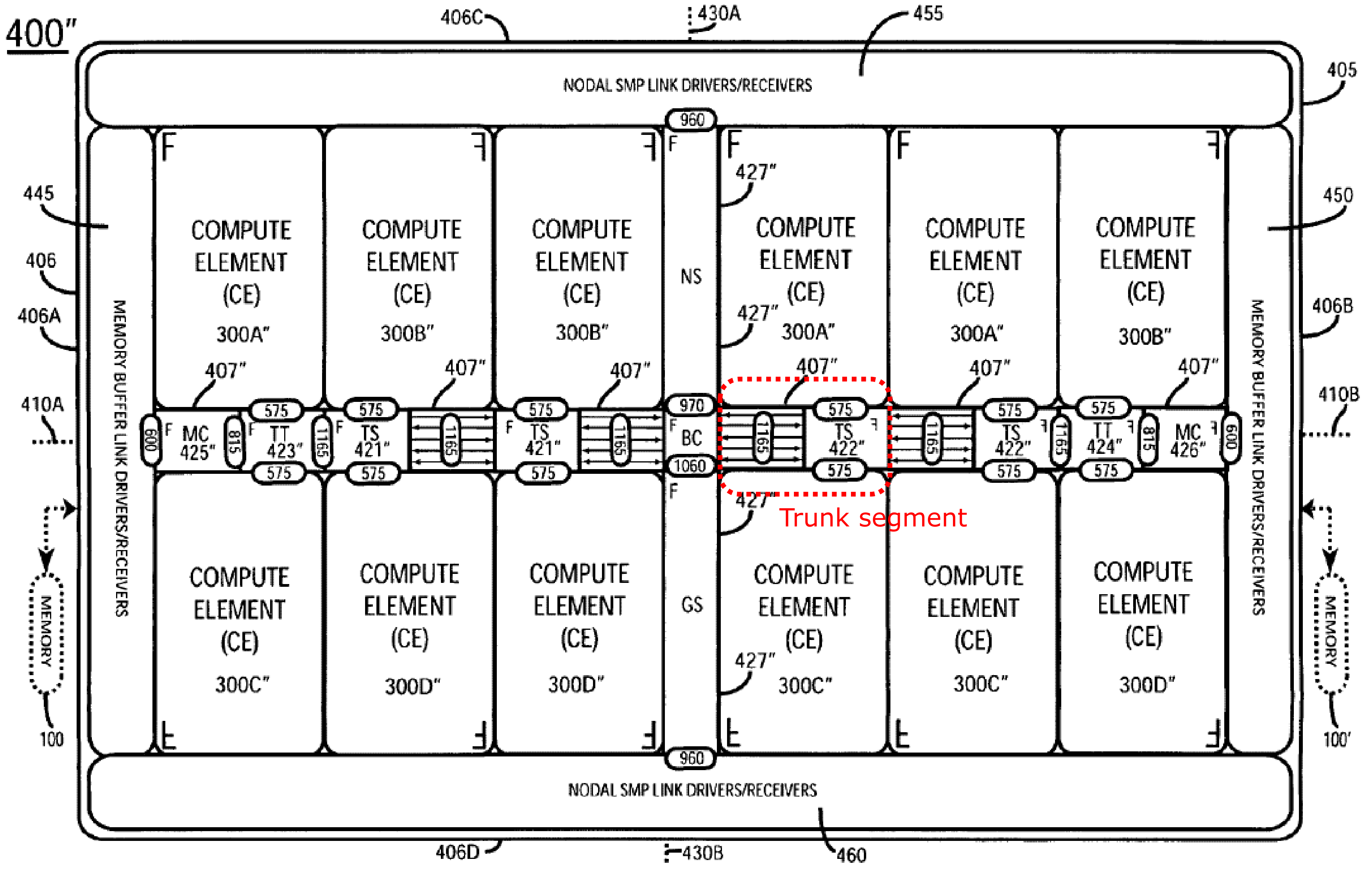
10.3.8 Enhanced on-chip interconnect

10.3.8 Enhanced on-chip interconnect []

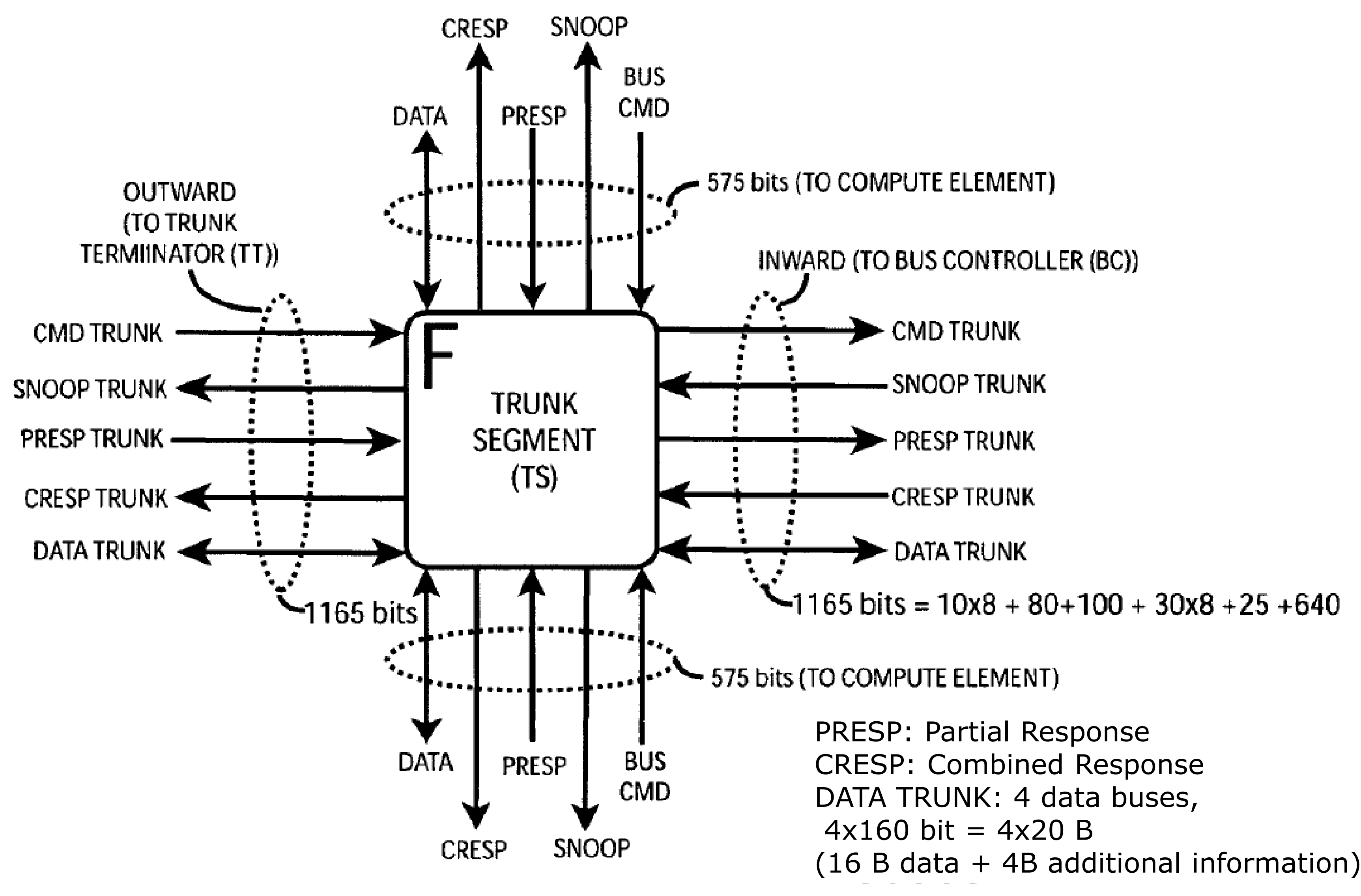


Per direction per segment bandwidth on the chip:
4 ring busses 16 B wide at 2.4 GT/s = 153.6 GB/s

The basic scheme of the on-chip interconnect for 12 cores as described along with the POWER7 [138]

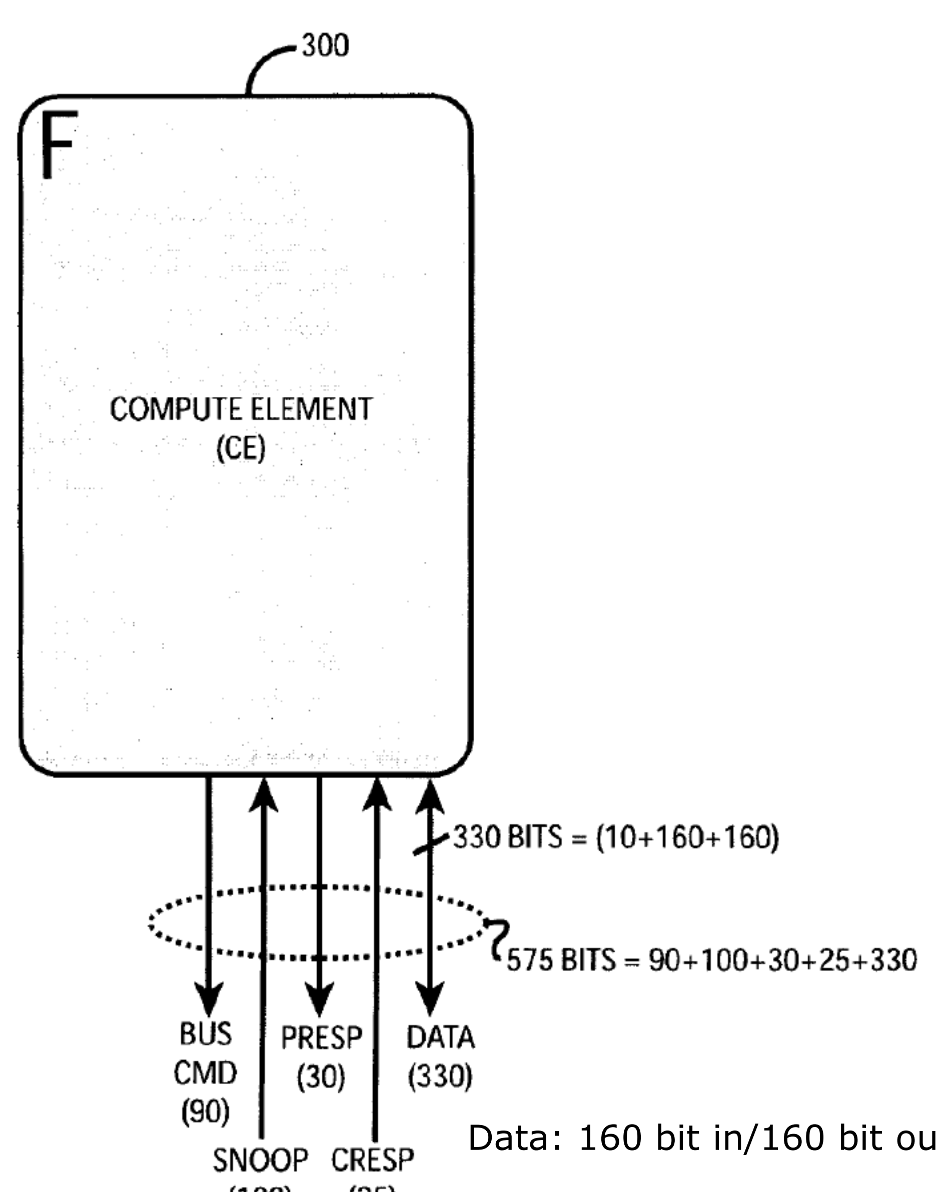


A trunk segment of the on-chip interconnect [138]



PRESP: Partial Response
CRESP: Combined Response
DATA TRUNK: 4 data buses,
4x160 bit = 4x20 B
(16 B data + 4B additional information)

Compute Element interface
to a trunk segment
according to the patent [138]



PRESP: Partial Response
CRESP: Combined Response

Data: 160 bit in/160 bit out

10.3.8 Enhanced on-chip interconnect (5)

The enhanced SMP interconnect of the POWER8 -1

The main enhancement of the coherent SMP interconnect of the POWER8 is a distributed arbitration scheme used rather than a central arbitration, as described below [114]:

." As in the POWER7 processor chip, the on-chip data connect consists of eight 16 B buses that span the chip horizontally, and are broken into multiple segments to handle the propagation delay across the chip, and allows the interconnect to be pipelined.

Four of the buses flow left to right, and four flow right to left, and the buses operate at up to 2.4 GHz.

Micro-architecture improvements were made to the internal data interconnect to reduce request latency by resolving on-chip data-routing decisions using a distributed-arbitration scheme instead of a centralized data arbiter in previous generations.

The on-chip interconnect also contains an adaptive-control mechanism to support independently controlled core frequencies while optimizing coherence-traffic bandwidth.

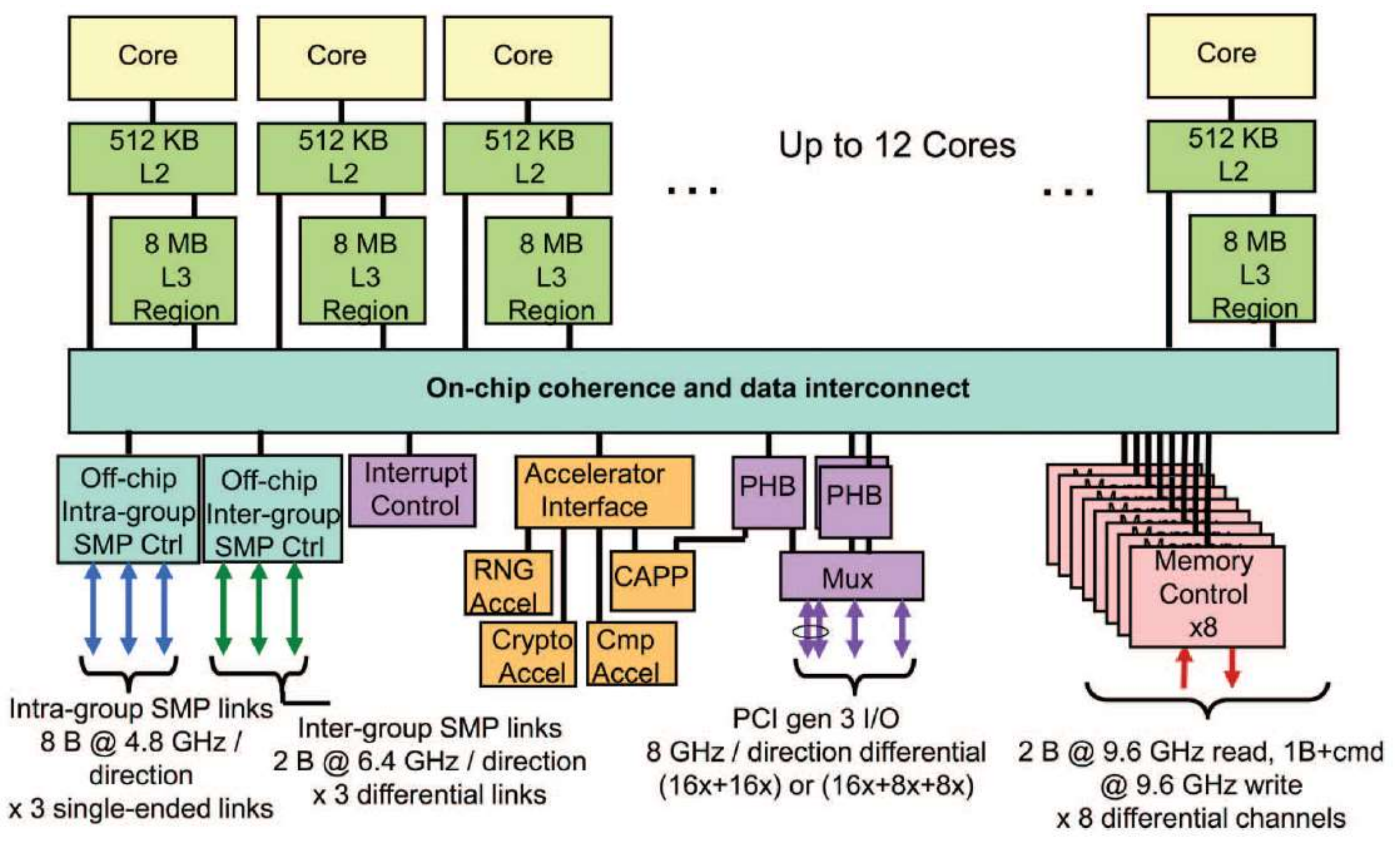
As core frequencies are adjusted upward or downward, the coherence-arbitration logic will throttle up or throttle down the rate at which commands are issued based on the frequency of the slowest core."

10.3.8 Enhanced on-chip interconnect (6)

The enhanced SMP interconnect of the POWER8 -2

- Further on, there are up to eight 9.6 GT/s differential memory interfaces (each with 2 B read and 1 B write) to the POWER8 memory system [144].
- Beyond the eight 16 B data buses (actually 4 ring buses) there are two address snoop buses.
- Altogether, there are 32 on/off ramps (instead of 20 along with the POWER7).
- There are three intra-group SMP links (X-buses) (8 B wide per direction, single ended, @ 4.8 GT/s) and
- three inter-group SMP links (A-buses for up to 48-way SMP) (2 B wide per direction, differential @ 6.4 GT/s), as indicated in the next slide.

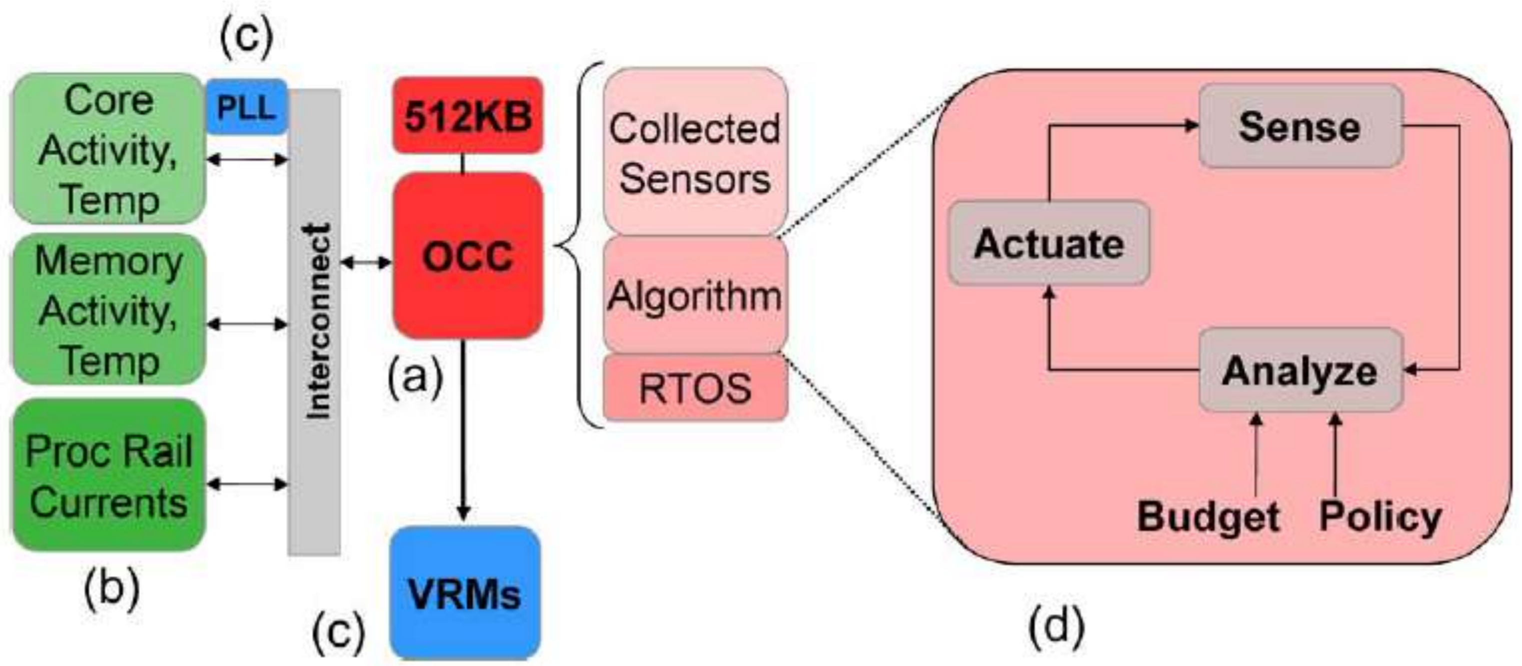
Block diagram of a POWER8 processor [114]



10.3.9 On-chip microcontroller based power management

10.3.9 On-chip Microcontroller (OCC) based power management-1 [95]

The POWER8 introduced an On-Chip Controller (OCC) to adjust the frequencies and voltages of the individual cores in response to the variations of the workload, based on measured core and cache activities, thermal sensor readings and current measurements, as indicated below.



(b): Sensor interfaces (c): Control points (actuators) (d): Algorithm

Figure: The On-Chip power management controller [95]

10.3.9 On-chip microcontroller based power management (2)

On-chip Controller (OCC) based power management-2 [95]

- The OCC is built up on a 32-bit PowerPC 405 core which runs real-time control software in a 512 kB SRAM, under the real-time OS RTOS.
- OCC fulfils three major tasks:
  - protecting the system in the event of fan or power supply failure,
  - optimizing the chip and further core frequencies according to the workloads and, if enabled,
  - managing the power-performance trade-off of cores by slowing down lightly used cores while speeding-up more heavier utilized ones.
- Among the inputs used by OCC are
  - the per-thread core and memory activity counters,
  - digital thermal temperature sensors, and
  - socket-based voltage and current readings, as well as
  - the thermal, power, and current constraints of the system.
- The key outputs of OCC are:
  - Core frequencies (PLL settings) and
  - Core voltages (VRM settings)

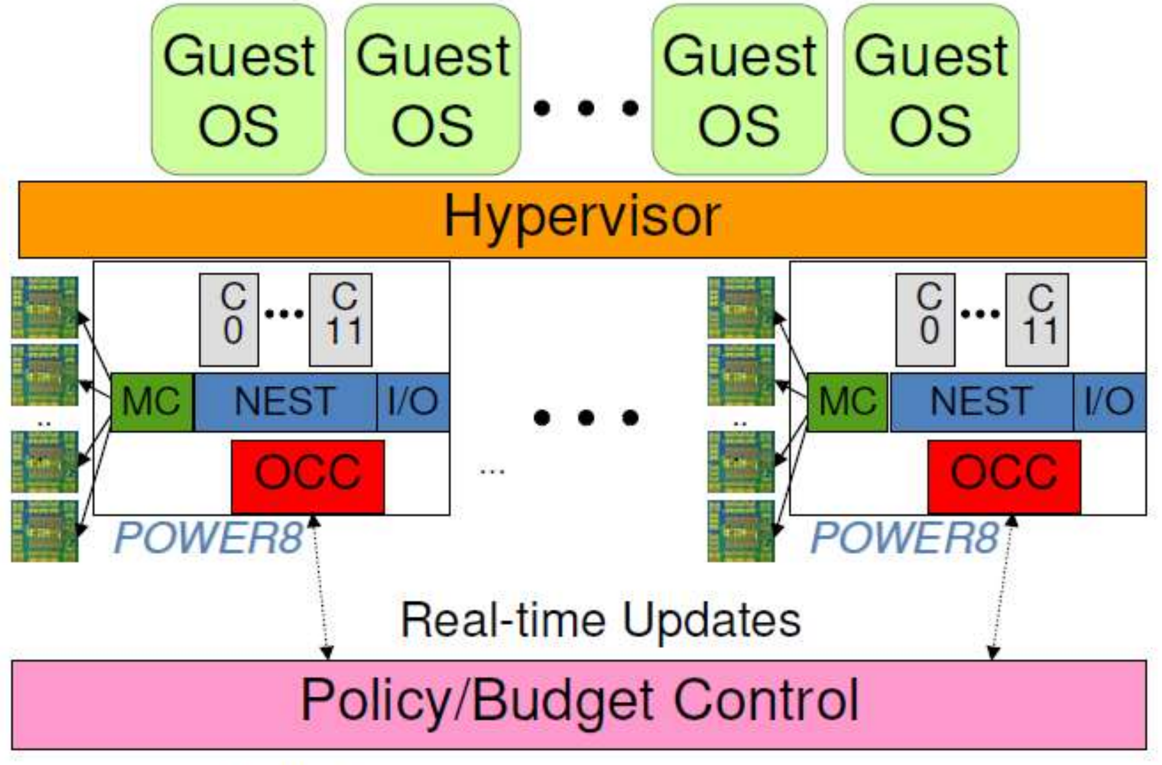
10.3.9 On-chip microcontroller based power management (3)

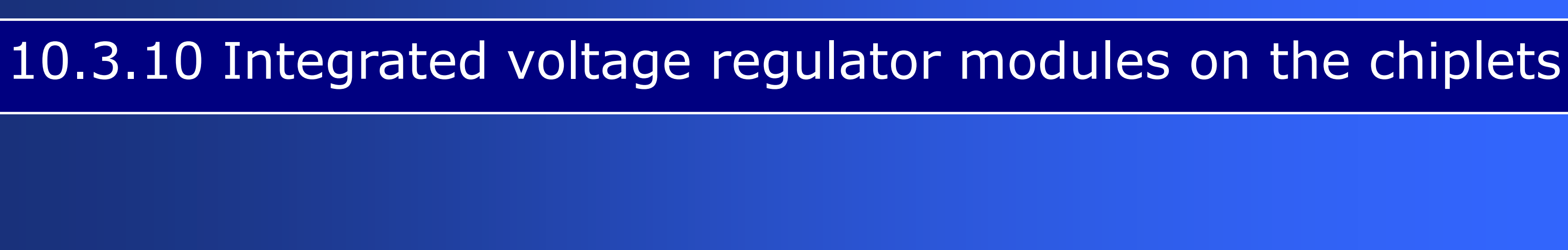
On-chip Controller (OCC) based power management-3 [95]

Benefit of the on-chip power management implementation vs. previous off-chip implementations

- There is a dramatic improvement of the system response time vs. the previous POWER7/POWER7+ processors, the control loop of the POWER8 runs at 0.25 ms vs. 32 ms in case of the previous models with off-chip controllers.
- As a consequence, OCC is much more suitable to support the power management of multi socket systems than the previous implementation.

Use of the On-Chip Controller (OCC) in POWER8-based multiprocessors [97]



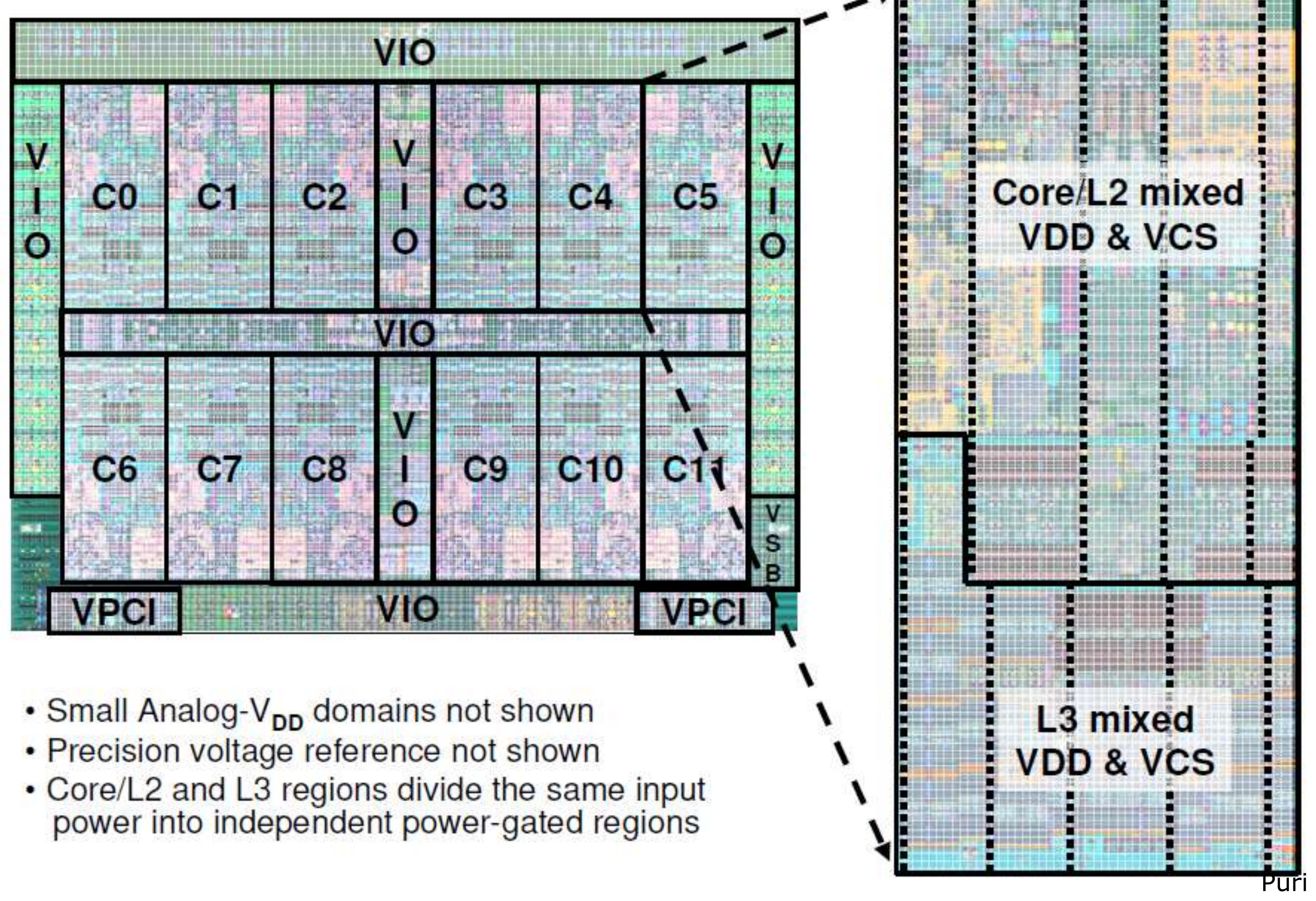

10.3.10 Integrated voltage regulator modules on the chiplets

10.3.10 Integrated per-chiplet VRMs

Before discussing integrated per-chiplet VRMs (Voltage Regulator Modules) introduced by IBM first we give an overview of the voltage domains of the POWER8.

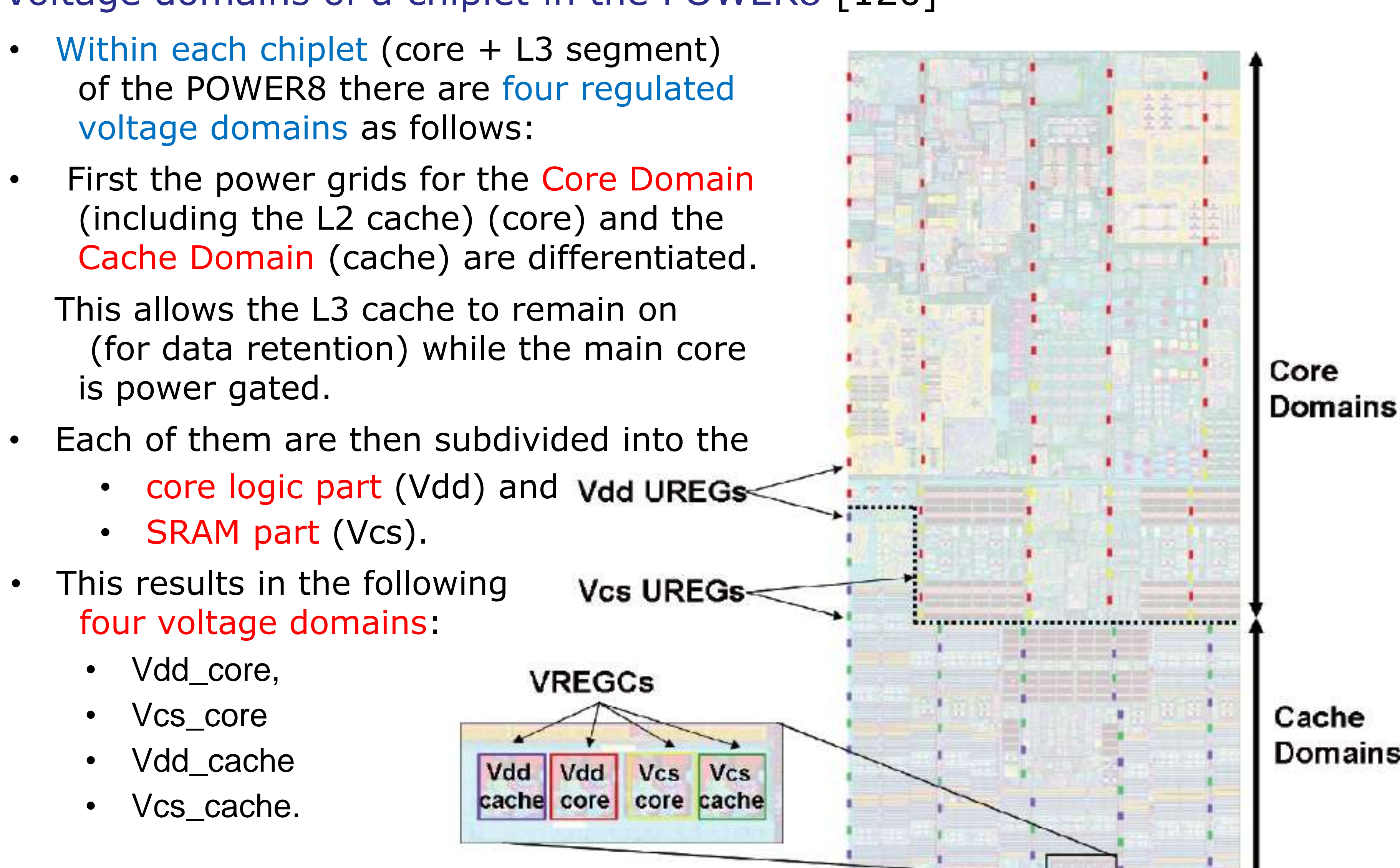
10.3.10 Integrated voltage regulator modules on the chiplets (2)

Voltage domains of the POWER8 [97]



10.3.10 Integrated voltage regulator modules on the chiplets (3)

Voltage domains of a chiplet in the POWER8 [120]



- Within each chiplet (core + L3 segment) of the POWER8 there are four regulated voltage domains as follows:
- First the power grids for the Core Domain (including the L2 cache) (core) and the Cache Domain (cache) are differentiated.

  This allows the L3 cache to remain on (for data retention) while the main core is power gated.
- Each of them are then subdivided into the
  - core logic part (Vdd) and
  - SRAM part (Vcs).
- This results in the following four voltage domains:
  - Vdd_core,
  - Vcs_core
  - Vdd_cache
  - Vcs_cache.

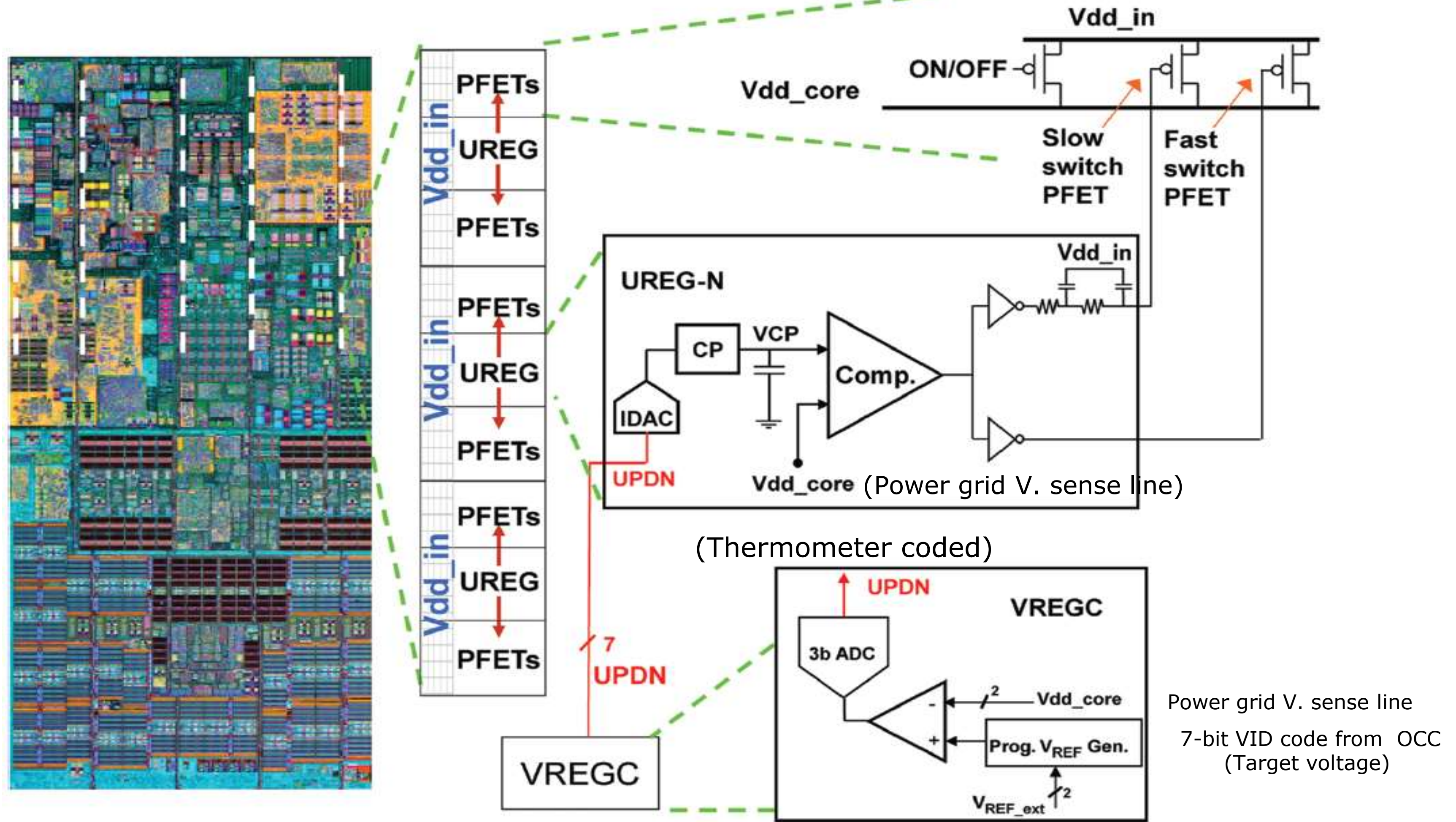
10.3.10 Integrated voltage regulator modules on the chiplets (4)

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-1 [120]

- On each chiplet there are four iVRMs, each for the four regulated voltage domains.
- Each iVRM is implemented as a distributed linear voltage regulator with
  - a single Voltage-Regulator Controller (VREGC) and
  - multiple (up to 64) micro-regulators (UREGs) distributed throughout the core

as shown for one of the four voltage domains, for the core logic domain (Vdd_core) of a chiplet in the next Figure.

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-2 [94]

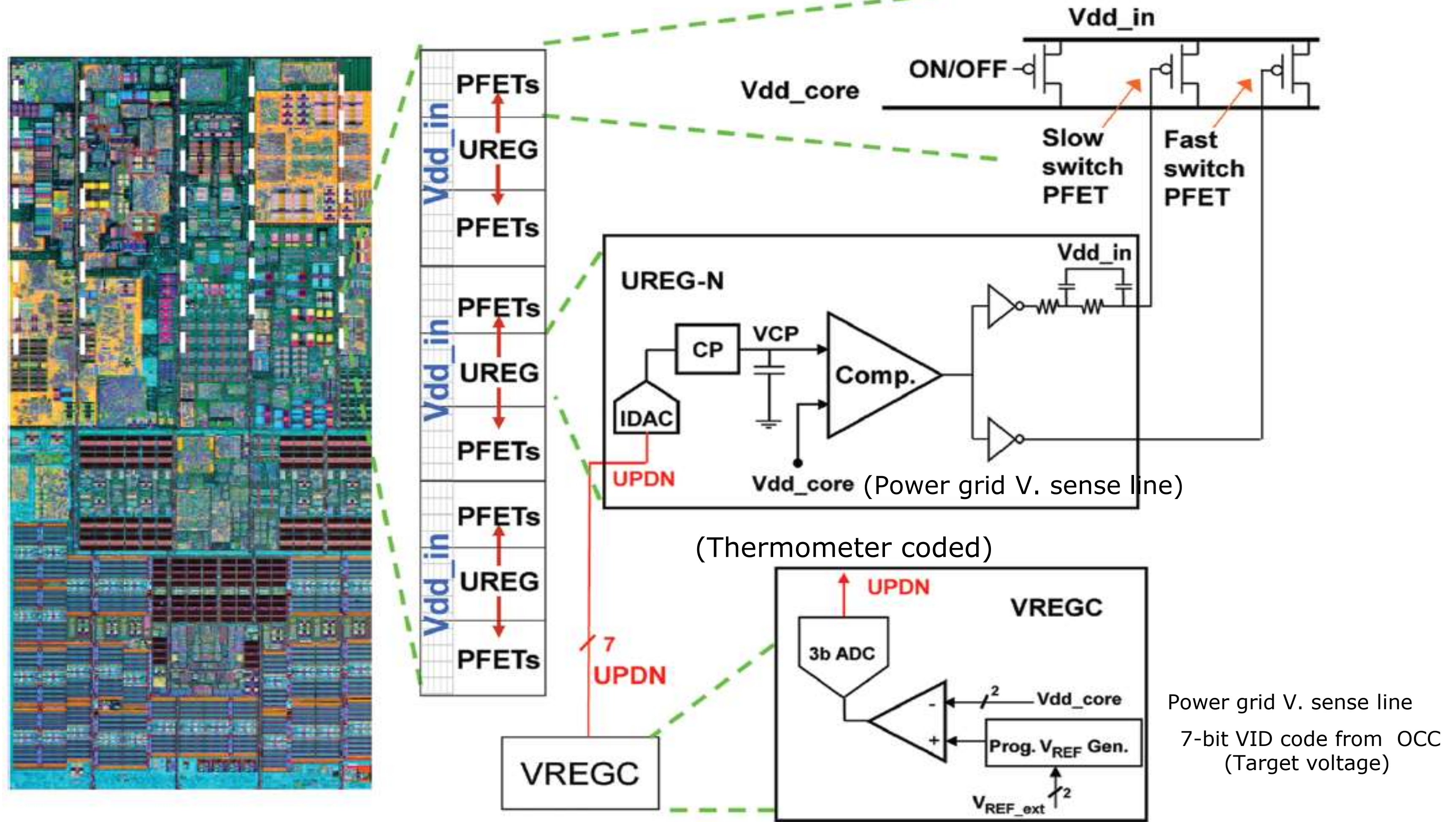


. UREG: microregulator; CP: charge pump; UPDN: 7-bit thermometer coded control; VREGC: central voltage regulator ; Vdd_core: core power grid voltage sense line; VCP: voltage at the output of the charge pump used as a reference point for UREG; IDAC: circuit decoding the UPDN code and controlling the charge pump.)

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-3 [94]

The UREG circuits are physically interleaved with the power switch transistors of the power gates (PFETS) and placed in five columns running through the core and caches, as shown in the next Figure.

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-4 [94]



. UREG: microregulator; CP: charge pump; UPDN: 7-bit thermometer coded control; VREGC: central voltage regulator ; Vdd_core: core power grid voltage sense line; VCP: voltage at the output of the charge pump used as a reference point for UREG; IDAC: circuit decoding the UPDN code and controlling the charge pump.)

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-5 [94]

- The Central Voltage Regulator (VREGC) unit senses the voltage at the core power supply grid (Vdd_core) and compares it to the internal target voltage.
- The internal target voltage is produced by the Programmed V_{REF} generator (Prog. VREF Gen.), based on the 7-bit digital VID code (Voltage IDentification code representing the target voltage value set by the power manager (OCC)) and an external high precision reference voltage (V_{REF_ext}).
- VREGC measures the difference between the sensed power grid voltage and the internal target voltage and generates a thermometer coded 7-bit code (UP-DOWN code) for controlling the UREGs.
- Based on the 7-bit UPDN thermometer code, the UREGs increase or decrease the pulse duty cycle of the slow and fast power switches (PFETs) in order to decrease or increase Vdd_core.

Remark

The pulse duty cycle is the ratio (e.g. expressed as percentage value) of the pulse period in which a signal is active compared to the total period of the signal.

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-6 [94]

- There are three groups of power switches (PFETs).
- The first group of PFETs, called fast PFETs, ensures a fast response to changes in the current consumption.

  The UREGs controls the duty cycle of the fast PFETs in response to any rapid changes in the core current consumption due to workload behavior.
- The second group of the power switch PFETs, is called slow PFETs.

  They are controlled by a slowly changing analog signal which is generated by the UREG via integrating the fast PFET control signal using a second-order low-pass filter and is a linear function of the duty cycle on the fast PFET.

  Thus, the Slow PFETs effectively supplies the static component of the core current.
- The transistors of the third group of PFETs are only turned on in the regulator bypass mode i.e. when the voltage regulator is switched off.

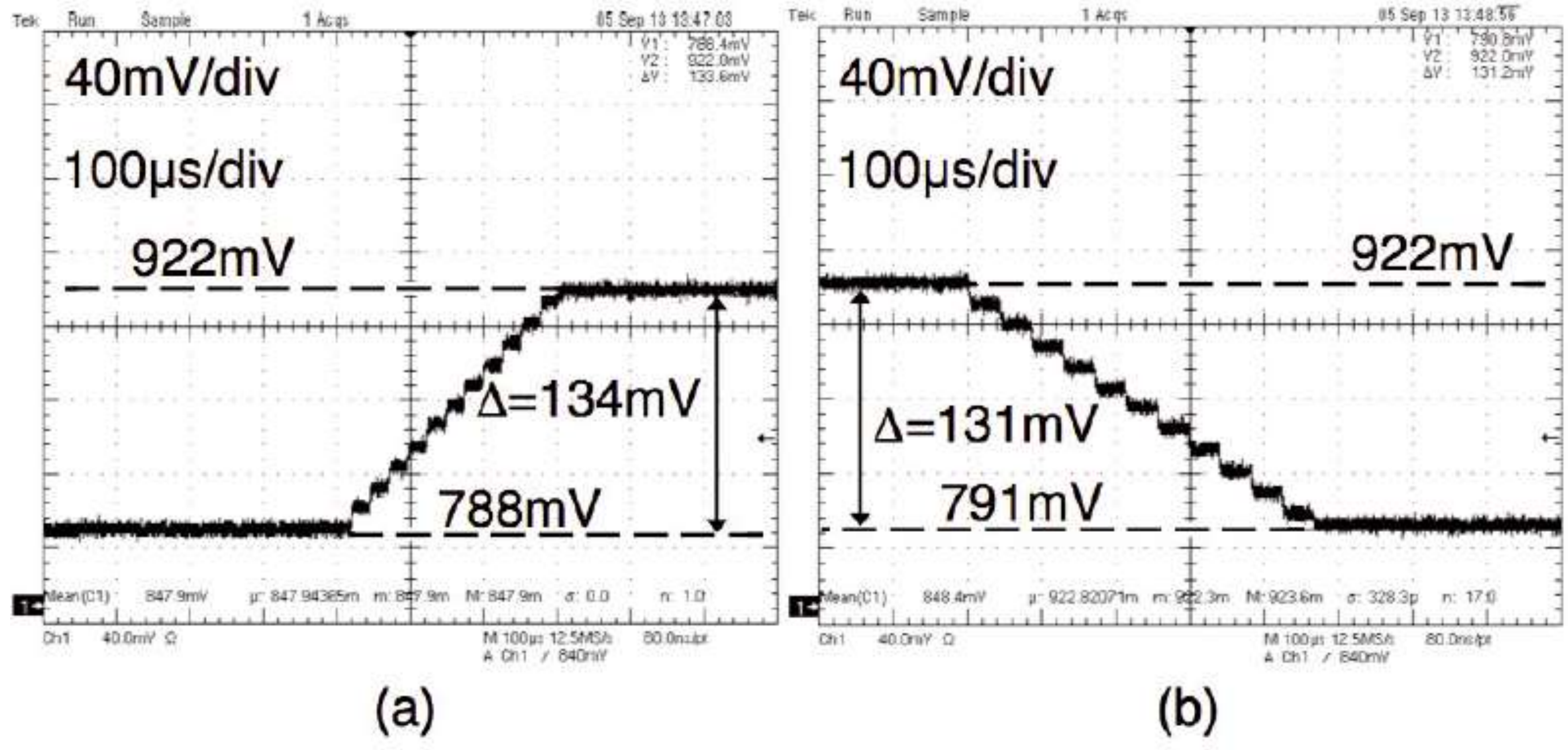
  They are turned 100% off when the voltage regulator is running.
- Note that the fast and slow power switches have two functions:
  - first, they operate as power gates, and
  - second, they operate as pulse duty cycle controlled linear voltage regulators.

10.3.10 Integrated voltage regulator modules on the chiplets (10)

Remark [94]

- While in the previous generation POWER7 the power gating switches could only operate in the 100% ON or OFF modes, the POWER8 processor re-uses the power switch transistors to implement the core-level linear voltage regulation.
- The effective resistance of the switch is controlled by the regulator circuit using a combination of modulating the voltage at the gates of a subset of header switch pFETs (Slow PFET) and applying pulses to another subset of the switch transistors (Fast PFET) with a controllable duty cycle.
- By rapidly adjusting the effective resistance of the header switch in response to workload-driven variations in core current consumptions the regulator can set the operating voltage of the core circuits to the level specified by the OCC on a per-core basis.

Stepping up and down Vdd-core voltage values [120]



Measured Vdd-core voltage values are shown in 12.5 mV steps for (a) upward and (b) downward directions.

Integrated Voltage Regulator Modules (iVRMs) on the chiplets-7 [120]

Key features of the iVRM of Vdd-core [120]

- The nominal resolution of Vdd-core is 6.25 mV.
- The iVRM of Vdd_core occupies about 1 % of the chiplet area.

Remark

Intel introduced per-core voltage regulators in their Haswell line of processors in 2013 termed Fully Integrated Voltage Regulator (FIVR).

Nevertheless, Intel's implementation is strongly different from the one used by IBM, as Intel preferred a Buck converter based concentrated, rather than distributed implementation, as indicated in the next Figures.

Principle of operation of Intel’s FIVR based on a Buck converter [121]

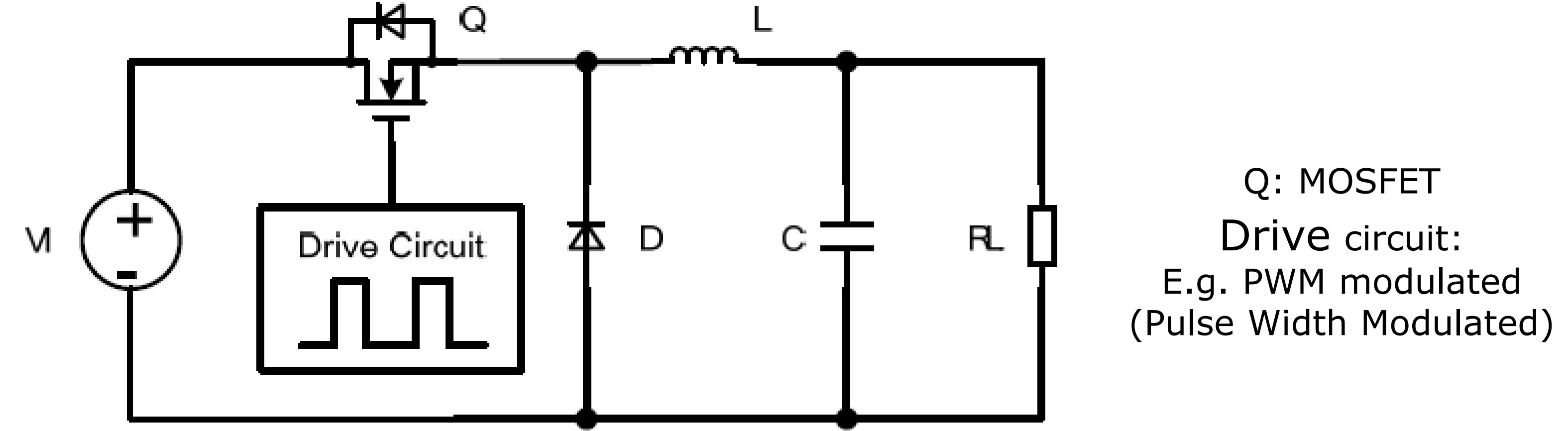


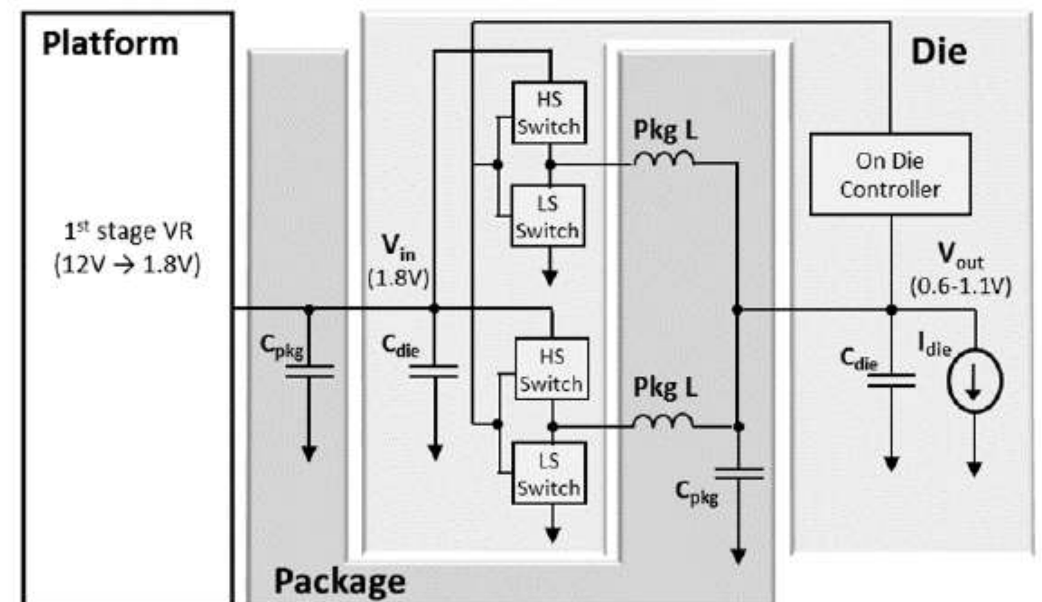
Figure: Block diagram of the Buck converter [121]

Figure: Operation of the Buck converter [121]

Partitioning of Intel's FIVR implementation [122]

- The first stage is on the motherboard.
- The inductors and the midfrequency decoupling capacitors are placed on the package.
- The power FETs, control circuitry and high frequency decoupling are on the die.
- Each FIVR is independently programmable to achieve optimal operation given the requirements of the domain it is powering.

- The settings are optimized by the Power Control Unit (PCU), which specifies the input voltage, output voltage, number of operating phases, and a variety of other settings to minimize the total power consumption of the die.

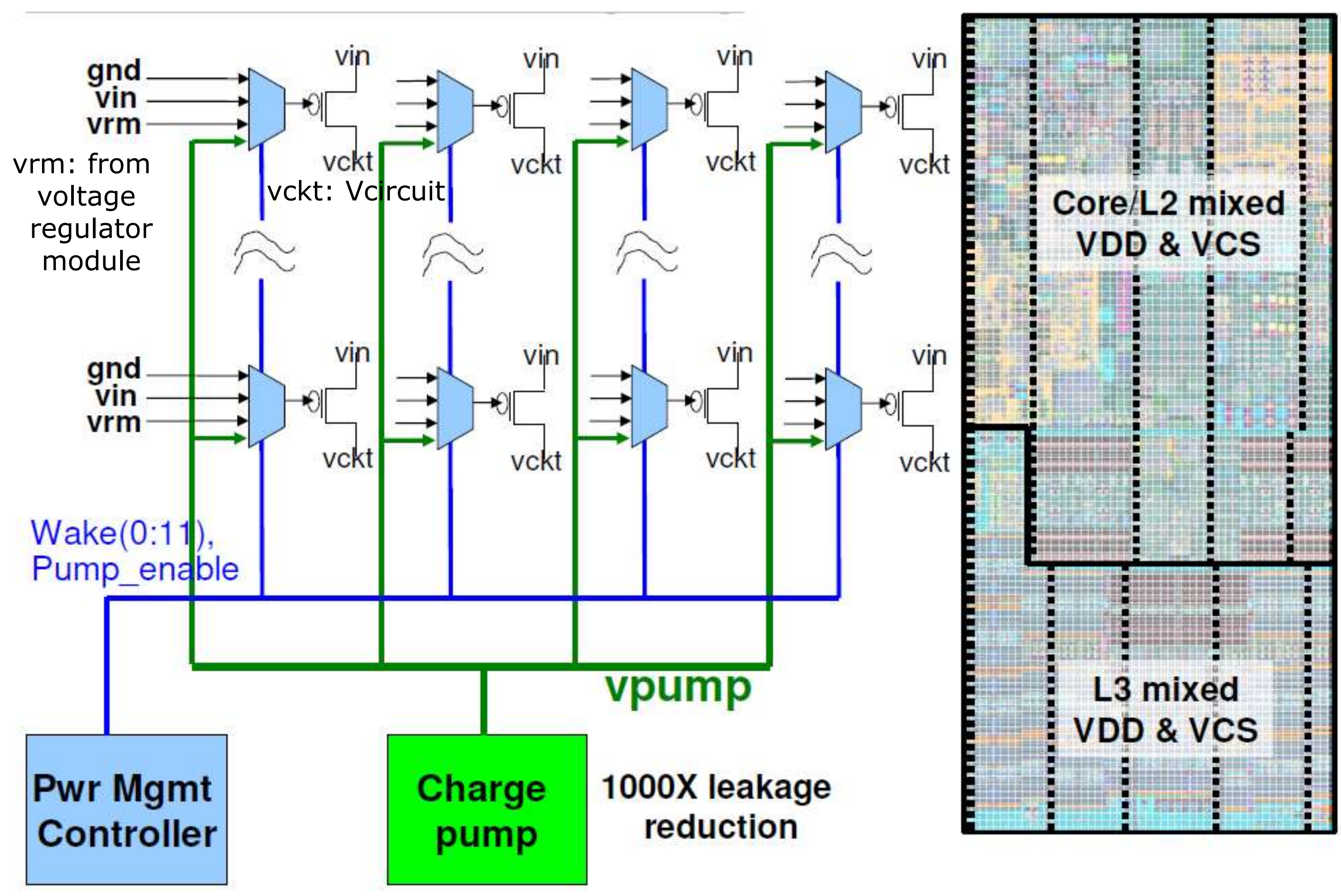


10.3.11 Using charge pumps for per-core power gating

10.3.11 Using charge pumps for per-core power gating -1

- IBM introduced per-core power gating in their POWER7 processors.
- In the POWER8 per-core power gating became improved by appropriate circuit enhancements (such as using charge pumps, not to be detailed here), only indicated in the next Figure.

Using charge pumps for per-core power gating [97] -2

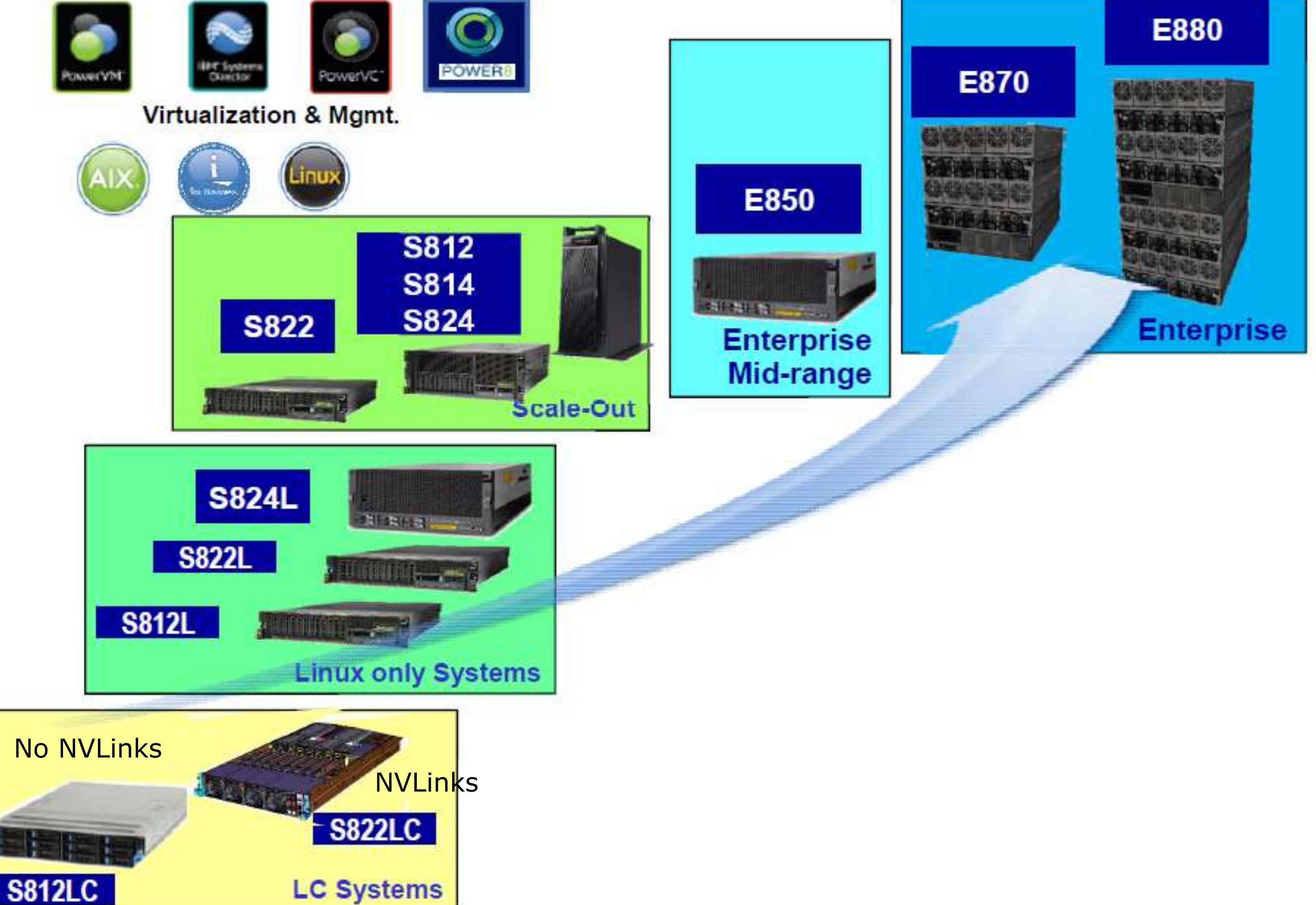


Using charge pumps for per-core power gating -3

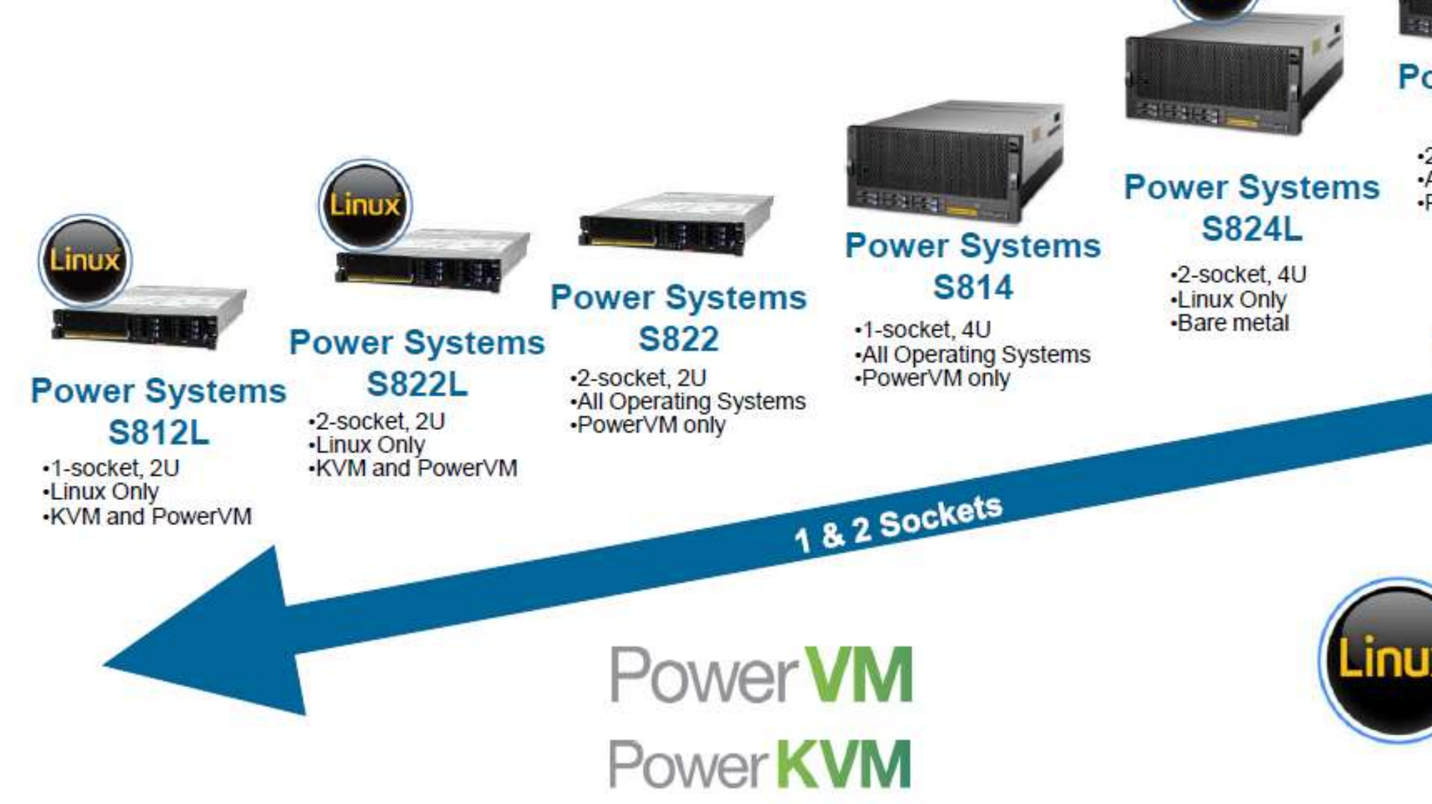
- In the POWER7+ processor IBM achieved a leakage reduction between 10 and 20, depending on the operating voltage, through per-core power gating.
- Subsequent circuit enhancements in POWER8's power gating (using charge pumps to raise the header gate voltage in the gated state), resulted in an over 1,000 times leakage reduction compared to the non-gated state.

10.4 POWER8-based server lines

The POWER8 family [133]



POWER8 systems Scale-Out portfolio [141]



11. POWER8+

11. POWER8+ [145], [146]

It was planned for 2016, as seen on an IBM roadmap (see below), but cancelled due to the too close launch date of the subsequent POWER9.

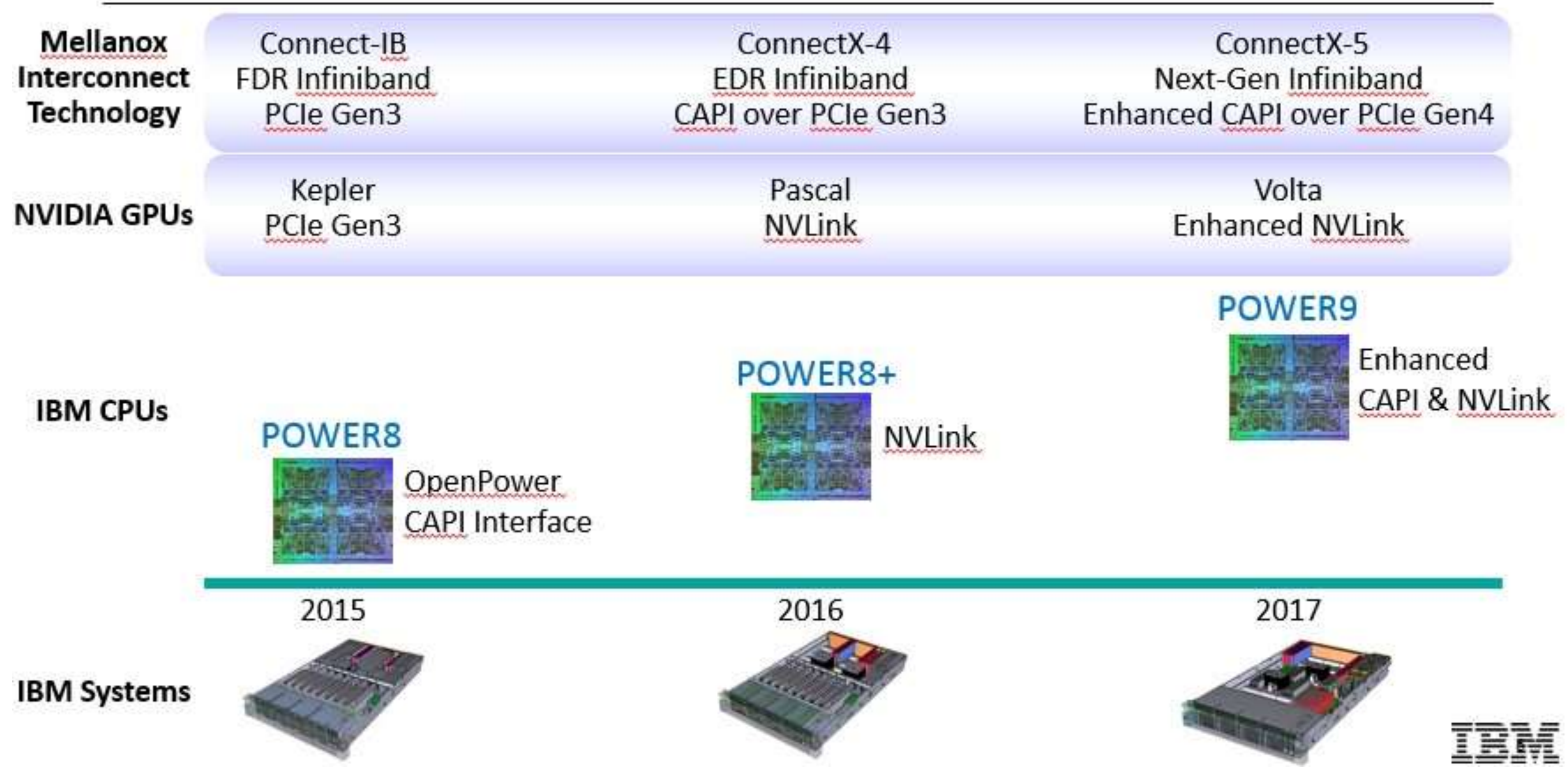


Figure: IBM's POWER roadmap from 3/2015 [146]

12. POWER9

- 12.1 Introduction to the POWER9
- 12.2 Main enhancements of the POWER9 ISA
- 12.3 Microarchitecture of the POWER9 processor
- 12.4 Enhancements in POWER9’s EnergyScale
- 12.5 POWER9 as a platform for accelerated computing
- 12.6 POWER9-based servers

12.1 Introduction to the POWER9

12.1 Introduction to the POWER9

- Announced in 4/2016 at the OpenPOWER Summit.
- It's Scale-Out version (with direct DDR4 memory channels) launched in 12/2017 (along with the 2S server AC922 targeting supercomputers).
- The Scale-Up version (with buffered memory channels) launched in 8/2018 (along with the 2S/4S server E950 and the 4S/node server (1 to 4 nodes) E980 enterprise servers).
- The POWER9 is IBM's first "platform for accelerated computing" i.e. a platform that is designed for heterogeneous computing with GPUs and dedicated FPGAs to offload tasks from the CPU and thus boost performance [147].
- 14 nm technology.
- 693 mm^2, 8.0 billion transistors.

Note

In their POWER8 family IBM interpreted the notion "scale-out" server differently, since there a scale-out server is an expandable server with a buffered memory subsystem rather than with direct memory channels as in POWER9 systems.

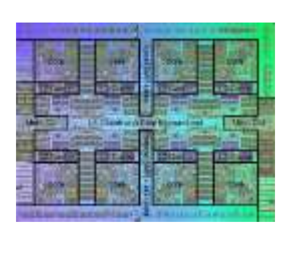
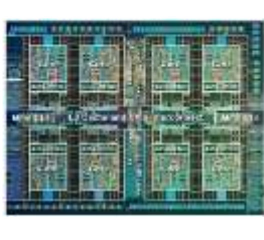
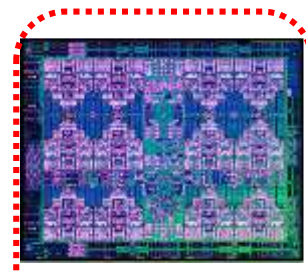
Remarks to the notions of scale-out and scale-up [148]

The terms "scale up" and "scale out" are commonly used in discussing different strategies for adding functionality to hardware systems. They are fundamentally different ways of addressing the need for more processor capacity, memory and other resources.

Scaling up generally refers to purchasing and installing a more capable central control or piece of hardware. For example, when a project's input/output demands start to push against the limits of an individual server, a scaling up approach would be to buy a more capable server with more processing capacity and RAM.

By contrast, scaling out means linking together other lower-performance machines to collectively do the work of a much more advanced one. With these types of distributed setups, it's easy to handle a larger workload by running data through different system trajectories.

Key features of the POWER9

	POWER7	POWER7+	POWER8	POWER8+	POWER9
Launched	2/2010	10/2012	4/2014	Planned/cancelled	12/2017
Technology	45 nm	32 nm	22 nm		14 nm
Die size	567 mm2	567 mm2	650 mm2		693 mm2
Transistors	1.2 b	2.1 b	4.2 b		8.0 b
Cores (up to)	8	8	12		12 SMT8 cores 24 SMT4 cores
SMT	4-way	4-way	8-way		4-way/8-way
Typ. fc	3.72-4-42 GHz	3.1 -4.42 GHz	3.02-4.35 GHz		Up to 4 GHz
L2	256 KB/core	256 KB/core	512 KB/core		512KB/2 cores
L3	4 MB/core	10 MB/core	12 MB/core		10 MB/2 cores
Mem. contr.	2/1	2/1	8		8
Memory up to	DDR3-1066	DDR3-1066	DDR3-1600		DDR4-2666

Main enhancements of the POWER9 microarchitecture [149]

New Core Microarchitecture

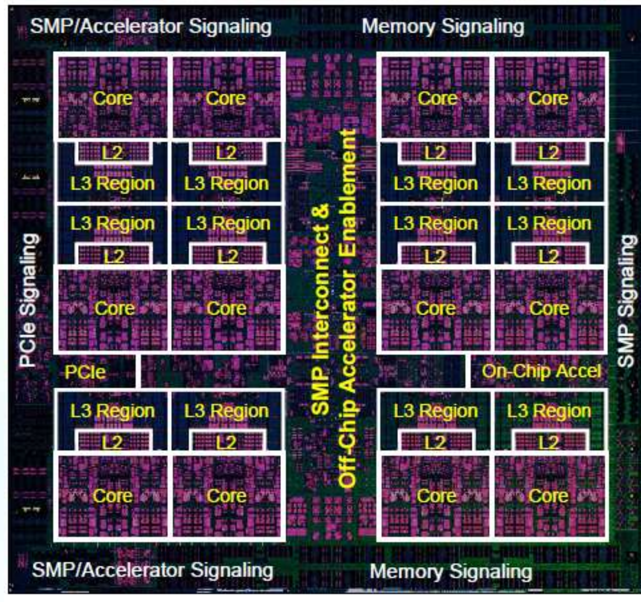
- Stronger thread performance
- Efficient agile pipeline
- POWER ISA v3.0

Enhanced Cache Hierarchy

- 120MB NUCA L3 architecture
- 12 x 20-way associative regions
- Advanced replacement policies
- Fed by 7 TB/s on-chip bandwidth

Cloud + Virtualization Innovation

- Quality of service assists
- New interrupt architecture
- Workload optimized frequency
- Hardware enforced trusted execution



14nm finFET Semiconductor Process

- Improved device performance and reduced energy
- 17 layer metal stack and eDRAM
- 8.0 billion transistors

Leadership Hardware Acceleration Platform

- Enhanced on-chip acceleration
- Nvidia NVLink 2.0: High bandwidth and advanced new features (25G)
- CAPI 2.0: Coherent accelerator and storage attach (PCIe G4)
- OpenCAPI: Improved latency and bandwidth, open interface (25G)

State of the Art I/O Subsystem

- PCIe Gen4 – 48 lanes

High Bandwidth Signaling Technology

- 16 Gb/s interface
  - Local SMP
- PowerAXON 25 GT/sec Link interface
  - Accelerator, remote SMP

SU: Buffered memory

2.1 Introduction to the POWER9 (5)

Introduction of different lines of the POWER9 family [149]

	POWER7 Architecture		POWER8 Architecture		POWER9 Architecture			POWER10
	2010 POWER7 8 cores **45nm** New Micro-Architecture New Process Technology	**2012 POWER7+** 8 cores **32nm** Enhanced Micro-Architecture New Process Technology	**2014 POWER8** 12 cores **22nm** New Micro-Architecture New Process Technology	**2016 POWER8 w/ NVLink** 12 cores **22nm** Enhanced Micro-Architecture With NVLink	**2017 P9 SO** 12/24 cores **14nm** New Micro-Architecture Direct attach memory New Process Technology	**2018 P9 SU** 12/24 cores **14nm** Enhanced Micro-Architecture Buffered Memory	**2019 P9 w/ Adv. I/O** 12/24 cores **14nm** Enhanced Micro-Architecture New Memory Subsystem	**2020+ P10** TBA cores New Micro-Architecture New Technology
Sustained Memory Bandwidth	65 GB/s	65 GB/s	210 GB/s	210 GB/s	150 GB/s	210 GB/s	350+ GB/s	435+ GB/s
Standard I/O Interconnect	PCIe Gen2	PCIe Gen2	PCIe Gen3	PCIe Gen3	PCIe Gen4 x48	PCIe Gen4 x48	PCIe Gen4 x48	PCIe Gen5
Advanced I/O Signaling	N/A	N/A	N/A	20 GT/s 160GB/s	25 GT/s 300GB/s	25 GT/s 300GB/s	25 GT/s 300GB/s	32 & 50 GT/s
Advanced I/O Architecture	N/A	N/A	CAPI 1.0	CAPI 1.0 , NVLink 1.0	CAPI 2.0, OpenCAPI3.0, NVLink2.0	CAPI 2.0, OpenCAPI3.0, NVLink2.0	CAPI 2.0, OpenCAPI4.0, NVLink3.0	TBA

© 2018 IBM Corporation

Statement of Direction, Subject to Change

SO: Scale-Out
SU: Scale-Up

12.2 Main enhancements of the POWER9 ISA

12.2 Main enhancements of the POWER ISA 3.0/3.0B versions (Based on [13

Power ISA version	Released	Main enhancements	Compliant POWER cores
PowerPC v. 2.01	9/2003		POWER4/4+
v. 2.02	01/2005		POWER5
Power ISA 2.03	06/2006	AltiVec	POWER5+
2.04	04/2007	Virtualization enhancements	POWER5
2.05	12/2007	Decimal arithmetic	POWER6/6+
2.06	02/2009	VSX (Vector Scalar)	POWER7
2.06 B	07/2010	Virtualization enhancements	POWER7/7+
2.07 with NVLink	05/2013	Transactional memory VMX, VSX2., crypto enhancements	POWER8
2.07 B	04/2015	Revised specification	POWER
3.0	11/2015	128-bit quad-precision FP operations, FP16 conversion, random number generator, hardware enforced trusted computing	POWER9 (Preliminary)
3.0 B	03/2017	Diverse instruction enhancements	POWER9

12.2 Main enhancements of the POWER9 ISA (2)

Remark [150]

- The POWER 3.0 ISA supports half-precision FP conversion but not half-precision FP computations.
- IBM’s argumentation is that half-precision computations should primarily be performed by GPUs or another accelerators.

More detailed list of the new instructions of the POWER ISA v3.0 [158]

Broader data type support

- 128-bit IEEE 754 Quad-Precision Float – Full width quad-precision for financial and security applications
- Expanded BCD and 128b Decimal Integer – For database and native analytics
- Half-Precision Float Conversion – Optimized for accelerator bandwidth and data exchange

Support Emerging Algorithms

- Enhanced Arithmetic and SIMD
- Random Number Generation Instruction

Accelerate Emerging Workloads

- Memory Atomics – For high scale data-centric applications

Cloud Optimization

- Enhanced Translation Architecture – Optimized for Linux
- New Interrupt Architecture – Automated partition routing for extreme virtualization
- Enhanced Accelerator Virtualization
- Hardware Enforced Trusted Execution

Energy & Frequency Management

- POWER9 Workload Optimized Frequency – Manage energy between threads and cores with reduced wakeup latency
  - Enables boost of frequency beyond the 3.1 Ghz base; Linux governors can also restrict / lower frequency to save power or boost other cores

© 2018 IBM Corporation

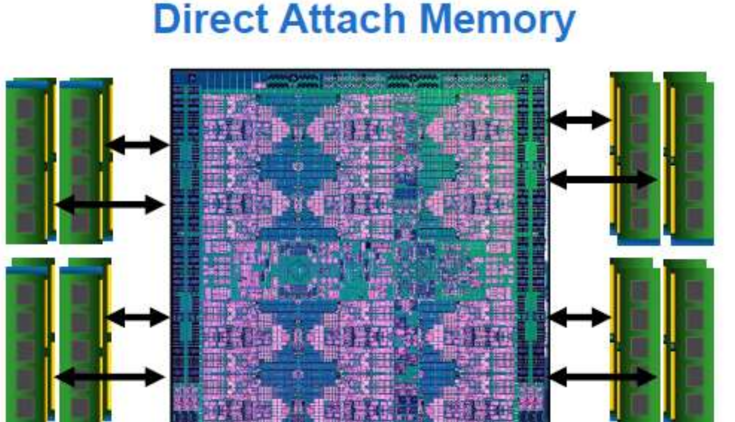
12.3 Microarchitecture of the POWER9 processor

12.3 Microarchitecture of the POWER9 processor

Kind of memory attachment of the POWER9 [149]

Scale Out option

Direct Attach Memory

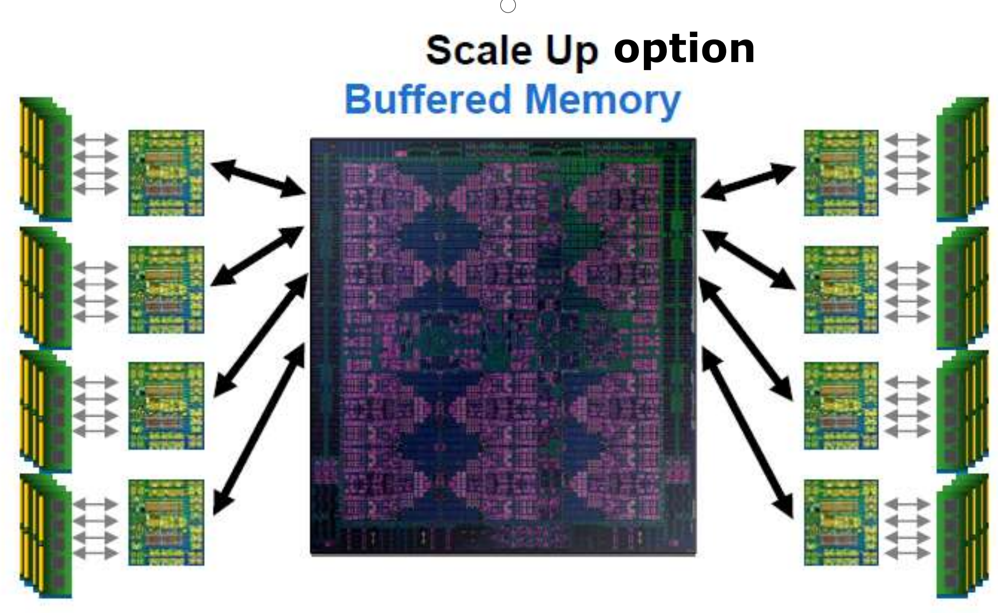


8 Direct DDR4 Ports

- Up to 120 GB/s of sustained bandwidth
- Low latency access
- Commodity packaging form factor
- Adaptive 64B / 128B reads
- 2 socket SMP optimized
- Up to 24 cores
- SMT4/SMT8

Scale Up option

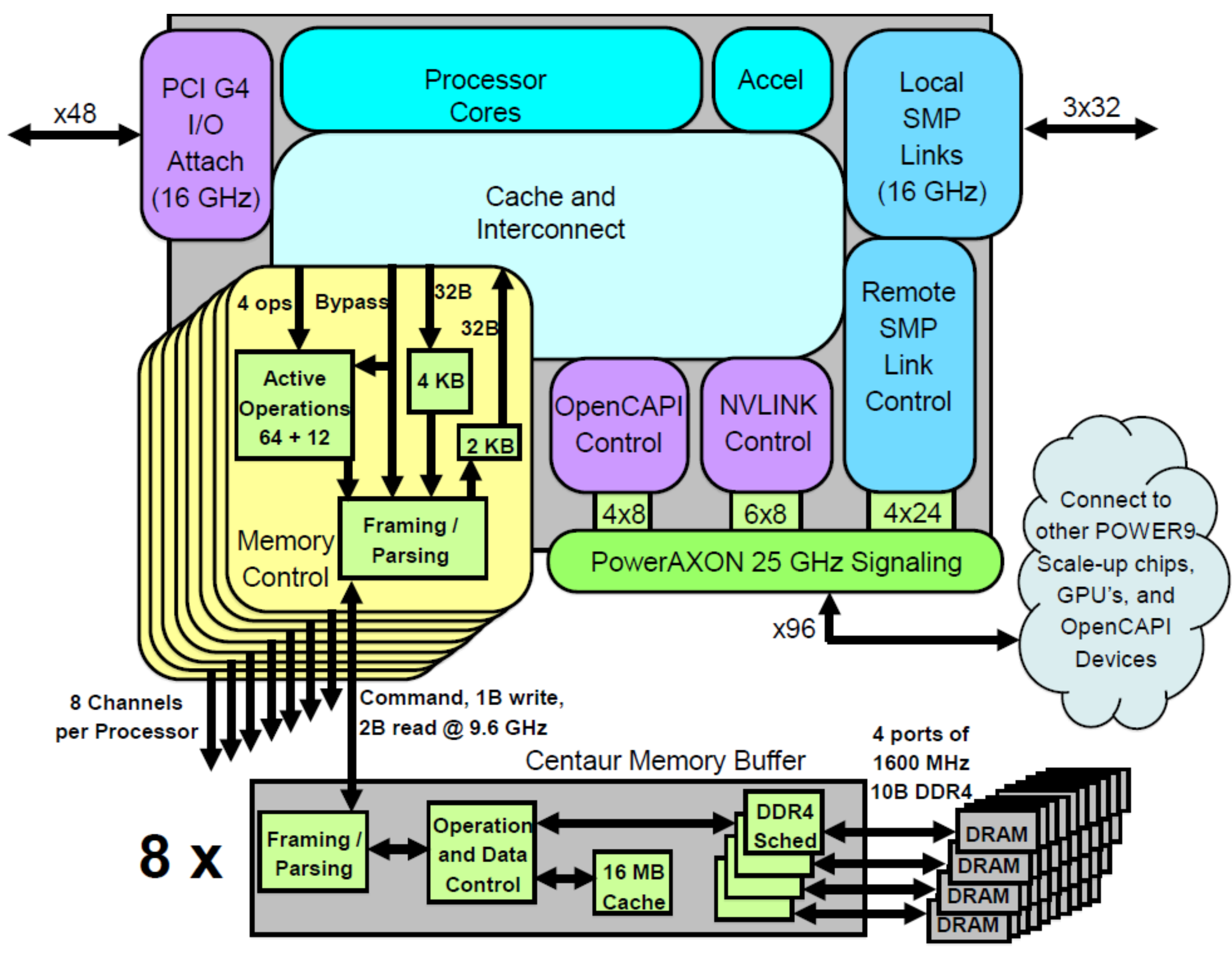
Buffered Memory



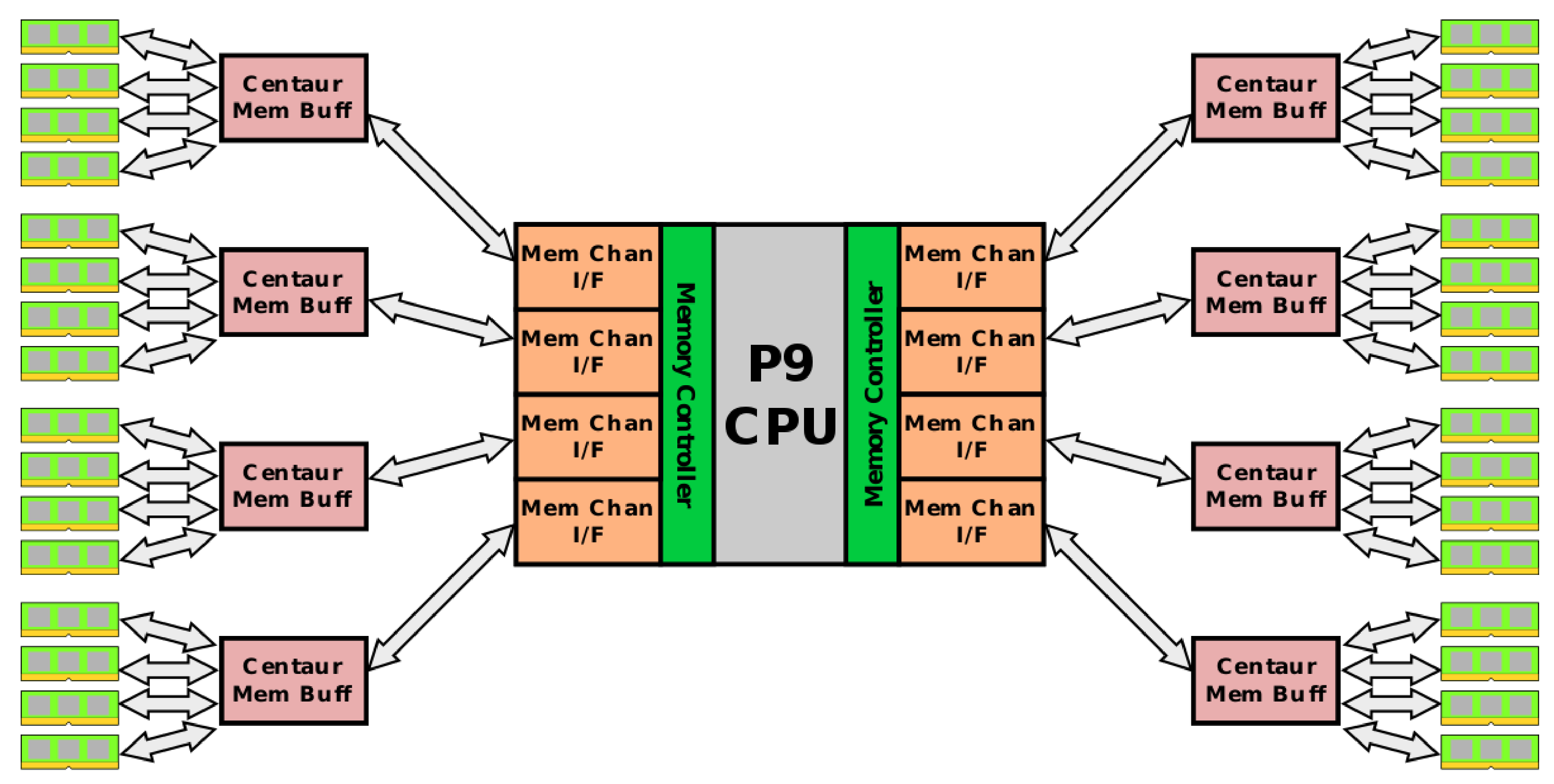
8 Buffered Channels

- Up to 230GB/s of sustained bandwidth
- Extreme capacity – up to 8TB / socket
- Superior RAS with chip kill and lane sparing
- Compatible with POWER8 system memory
- Agnostic interface for alternate memory innovations
- Multi-socket optimized (up to 16 sockets)
- Up to 24 cores
- SMT4/SMT8

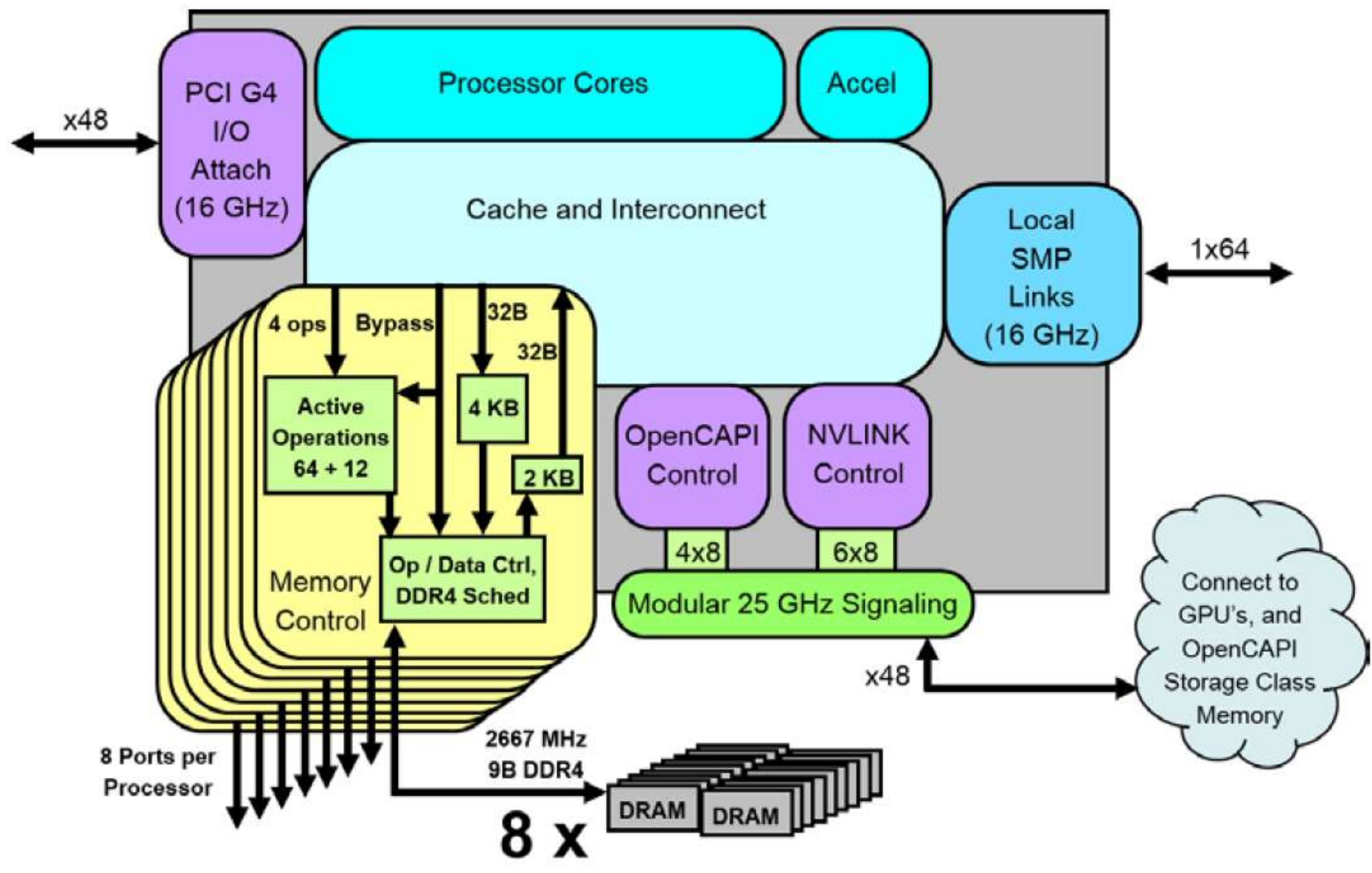
Memory controllers and Centaur Memory Buffers in the Scale-Up POWER9 system version [149]



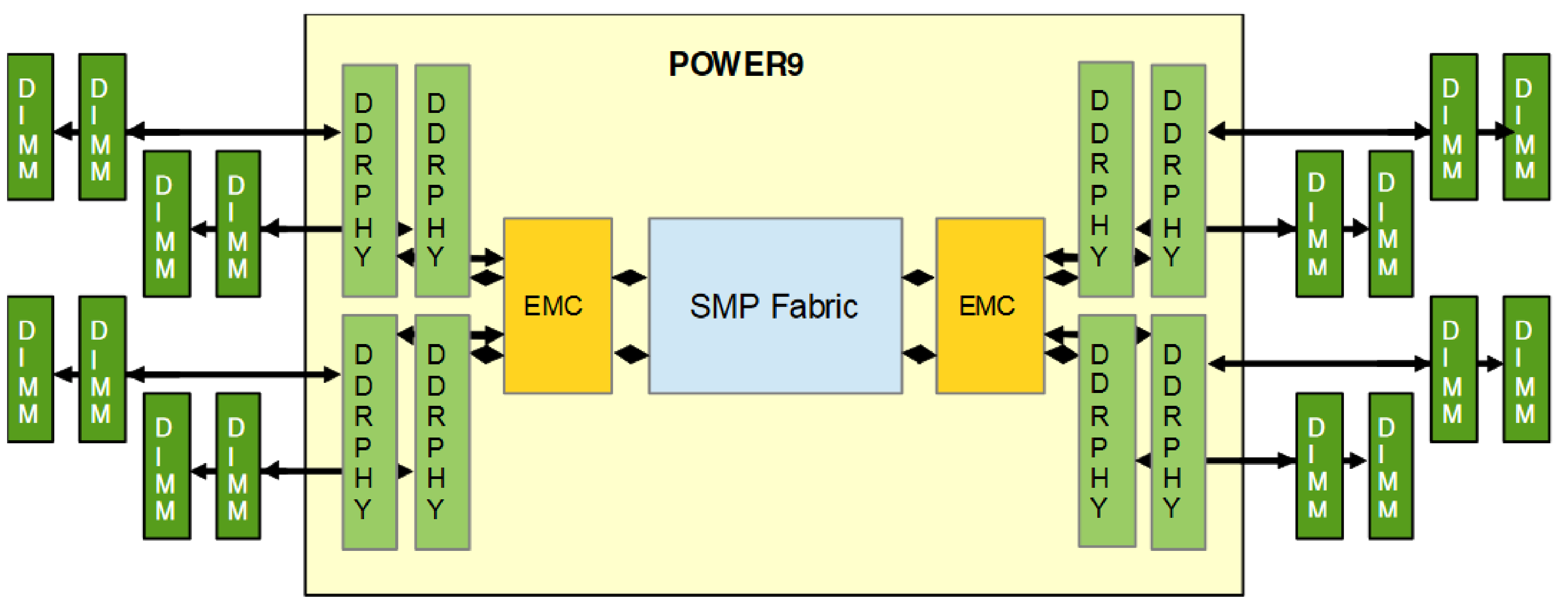
The memory subsystem of a POWER9-based scale-up system version [152]



Memory controllers and direct connected DDR4 DIMMs in the Scale-Out POWER9 system version [153]



The memory subsystem of a POWER9-based scale-out system version [154]



EMC: Extended Memory Controller

No. of ways of multithreading [155]

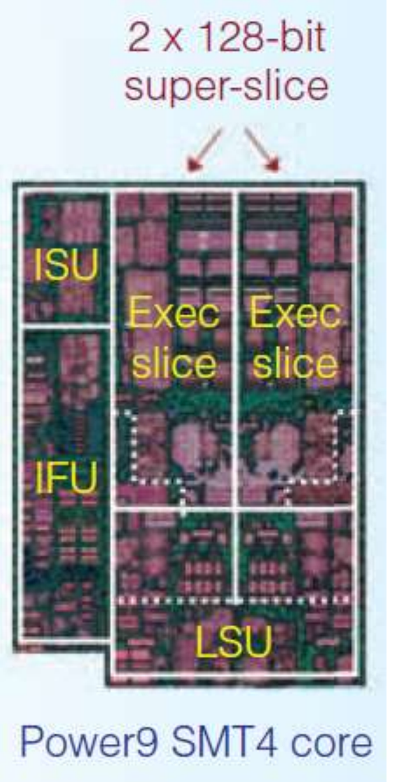
No. of ways of multithreading

SMT4
(4-way multithreaded)

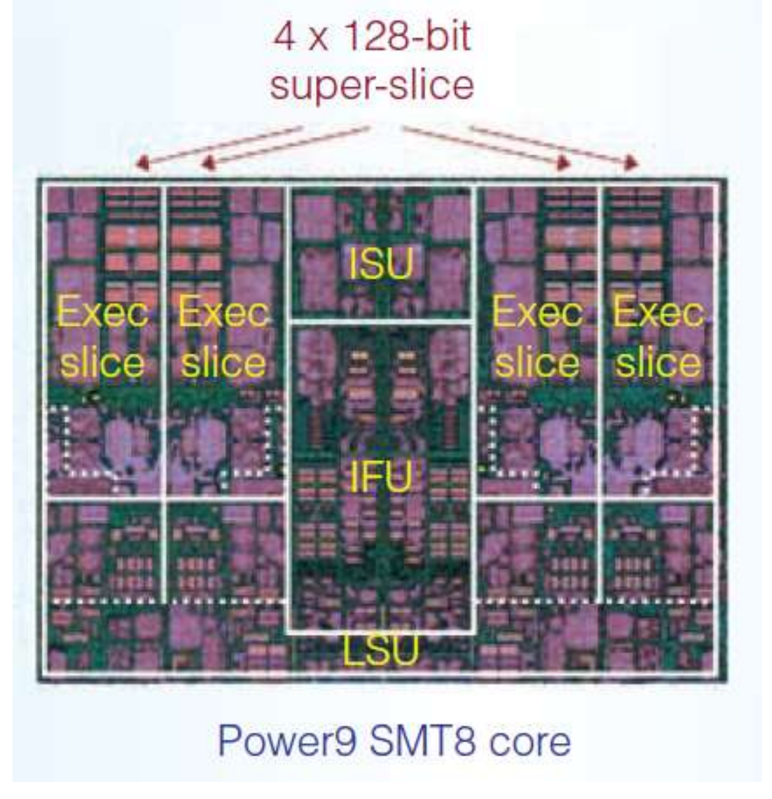
Thin core
(Up to 245 cores)

SMT8
(8-way multithreaded)

Wide core
(Up to 12 cores)

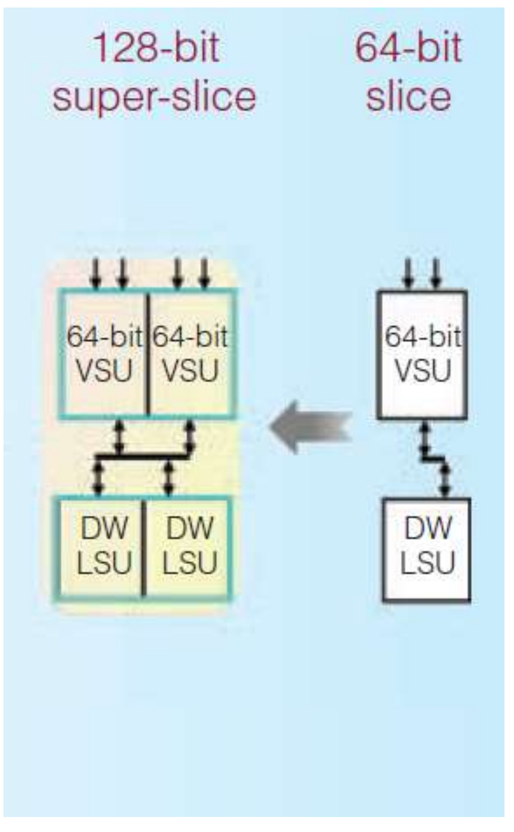


Power9 SMT4 core

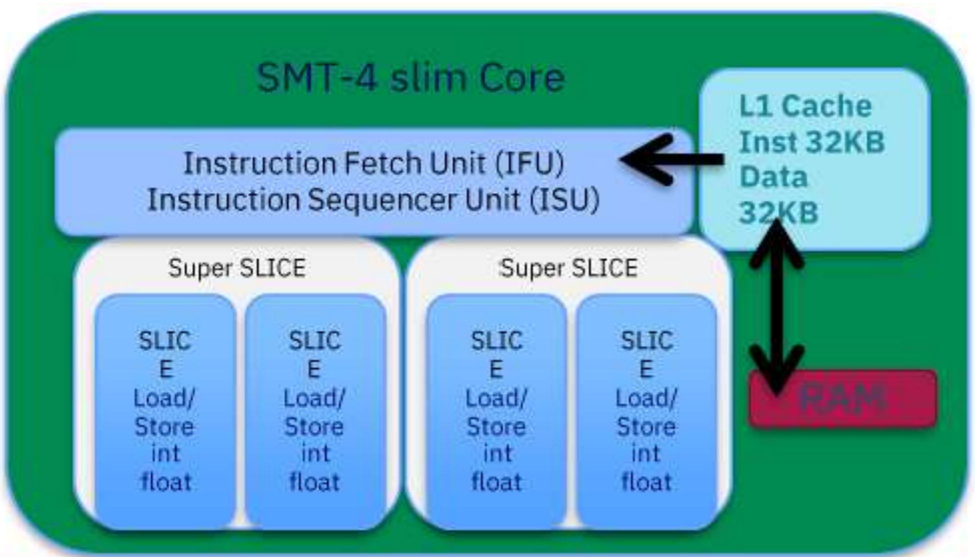


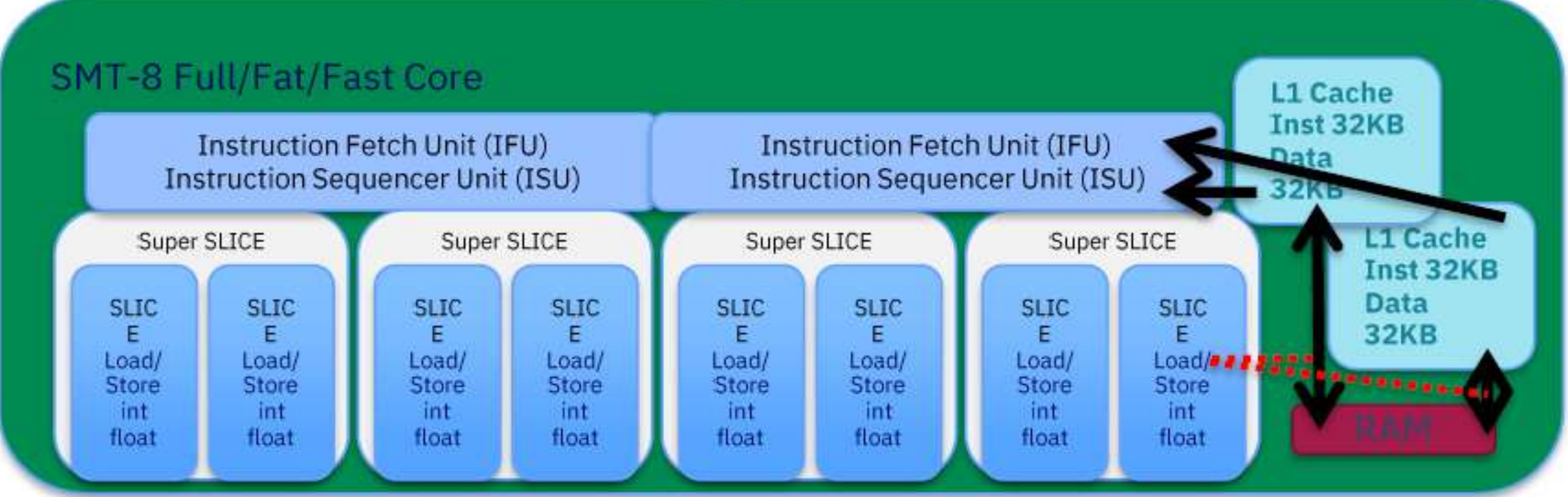
Power9 SMT8 core

Concept of a 64-bit slice and a 128-bit super-slice [155]

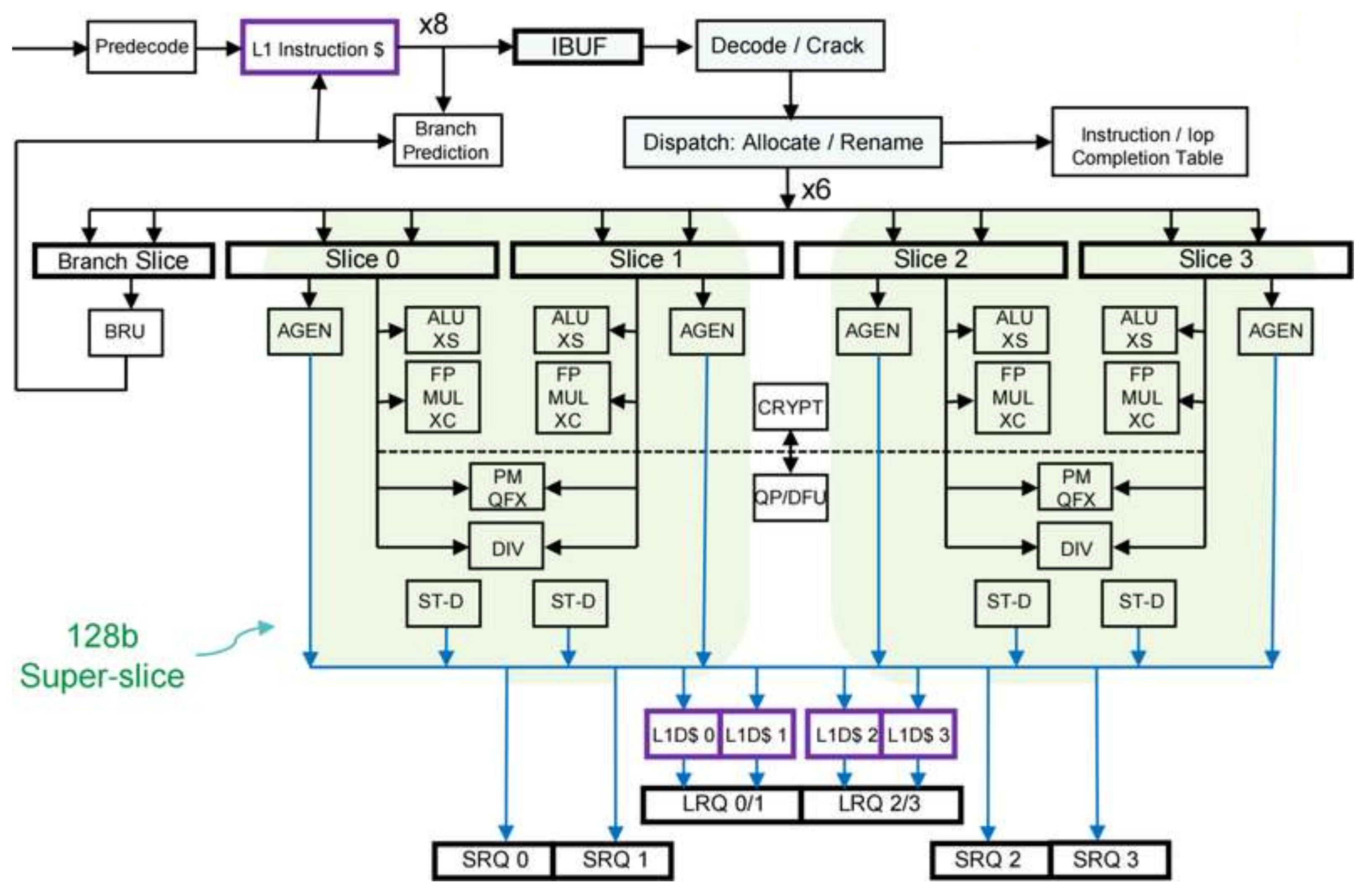


The SMT4 (slim) and SMT8 (wide) cores in more detail [133]

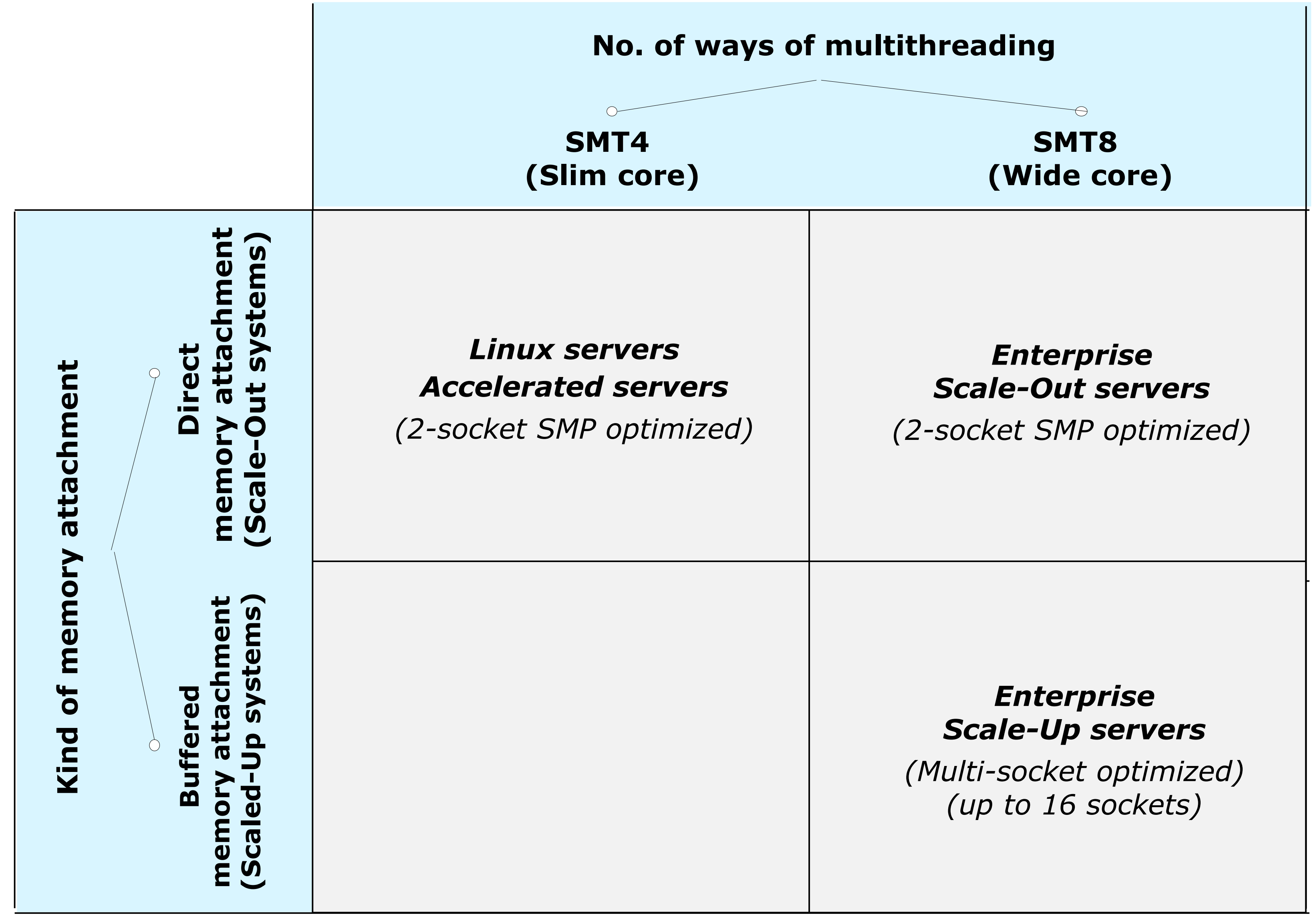




Detailed block diagram of an SMT4 core [155]



POWER9 server alternatives

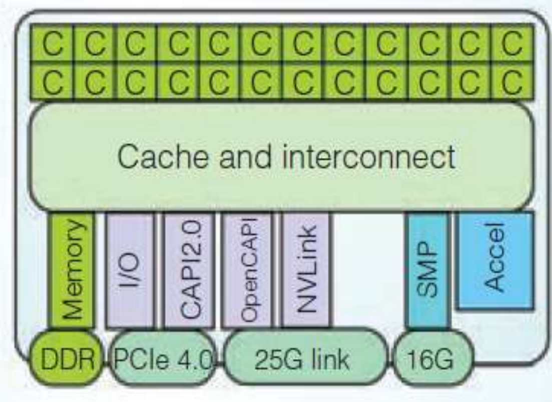


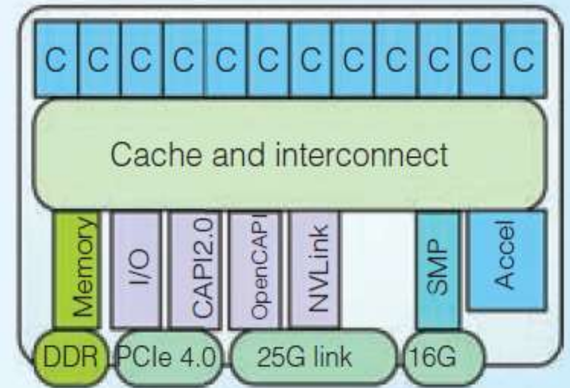
Layout of the four basic alternatives of POWER9 [155]

SMT4 core
24 SMT4 cores/chip
- Linux server optimized
- Accelerated servers optimized

SMT8 core
12 SMT cores/chip
Enterprise Scale-Out servers
Enterprise Scale-Up servers
(POWERVM optimized)

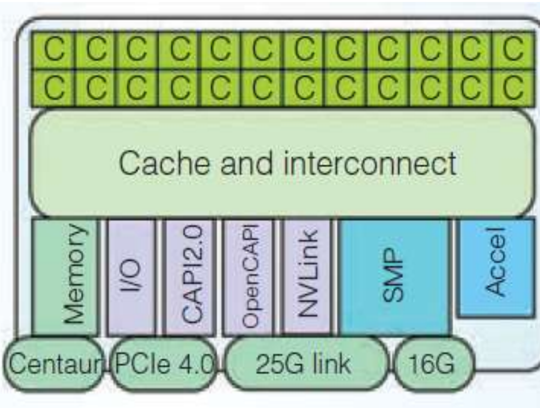
Scale-out–2 socket optimized
Robust two-socket SMP system
Direct memory attach
- Up to eight DDR4 ports
- Commodity packaging form factor

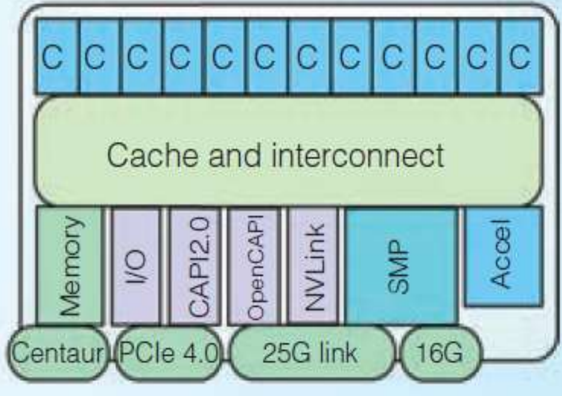




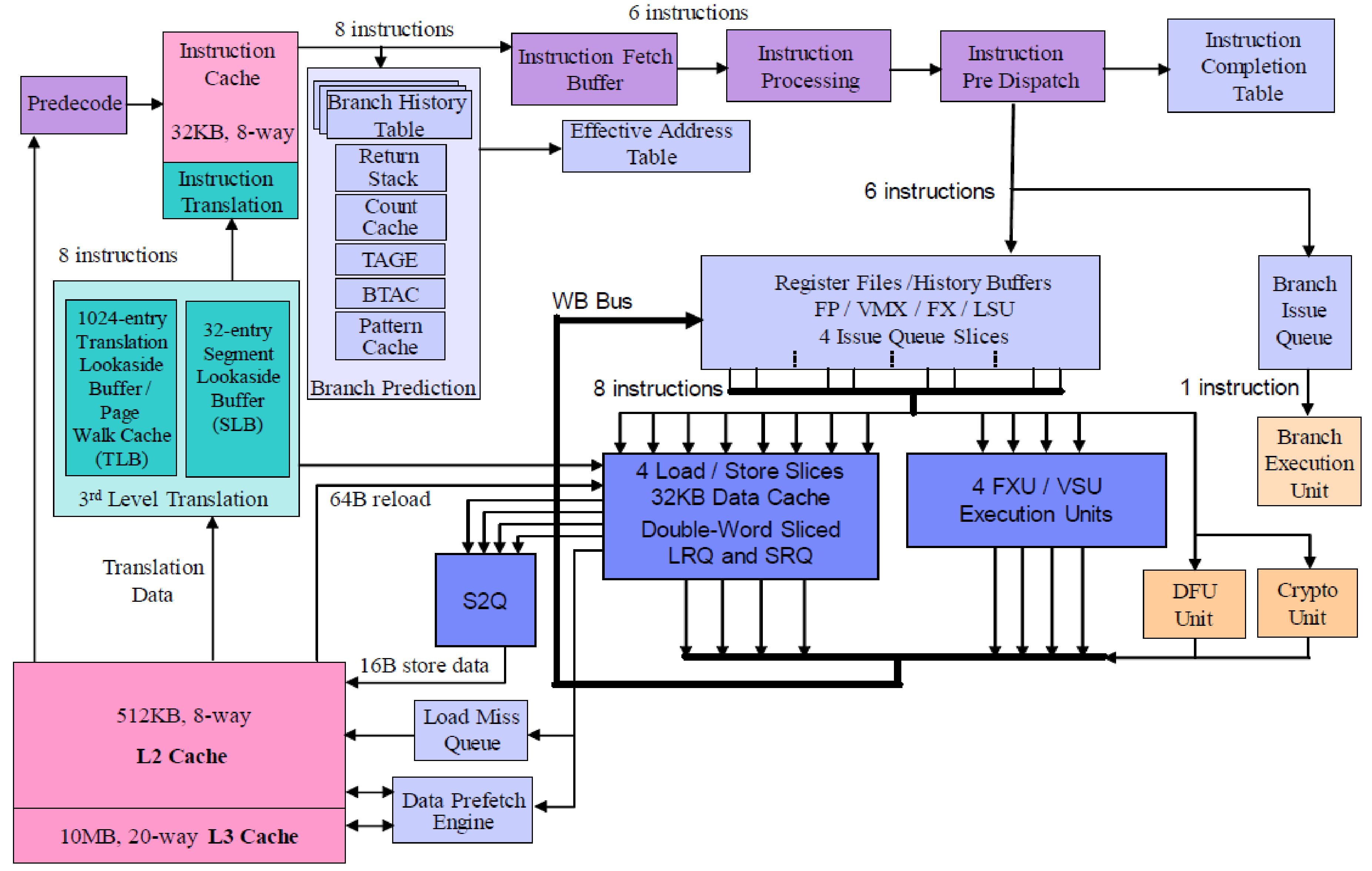
Scale-up–2 multisocket optimized
Scalable system topology/capacity
- Large multisocket
- Additional lanes of 25G link (96 total)

Buffered memory attach
- 8 buffered channels

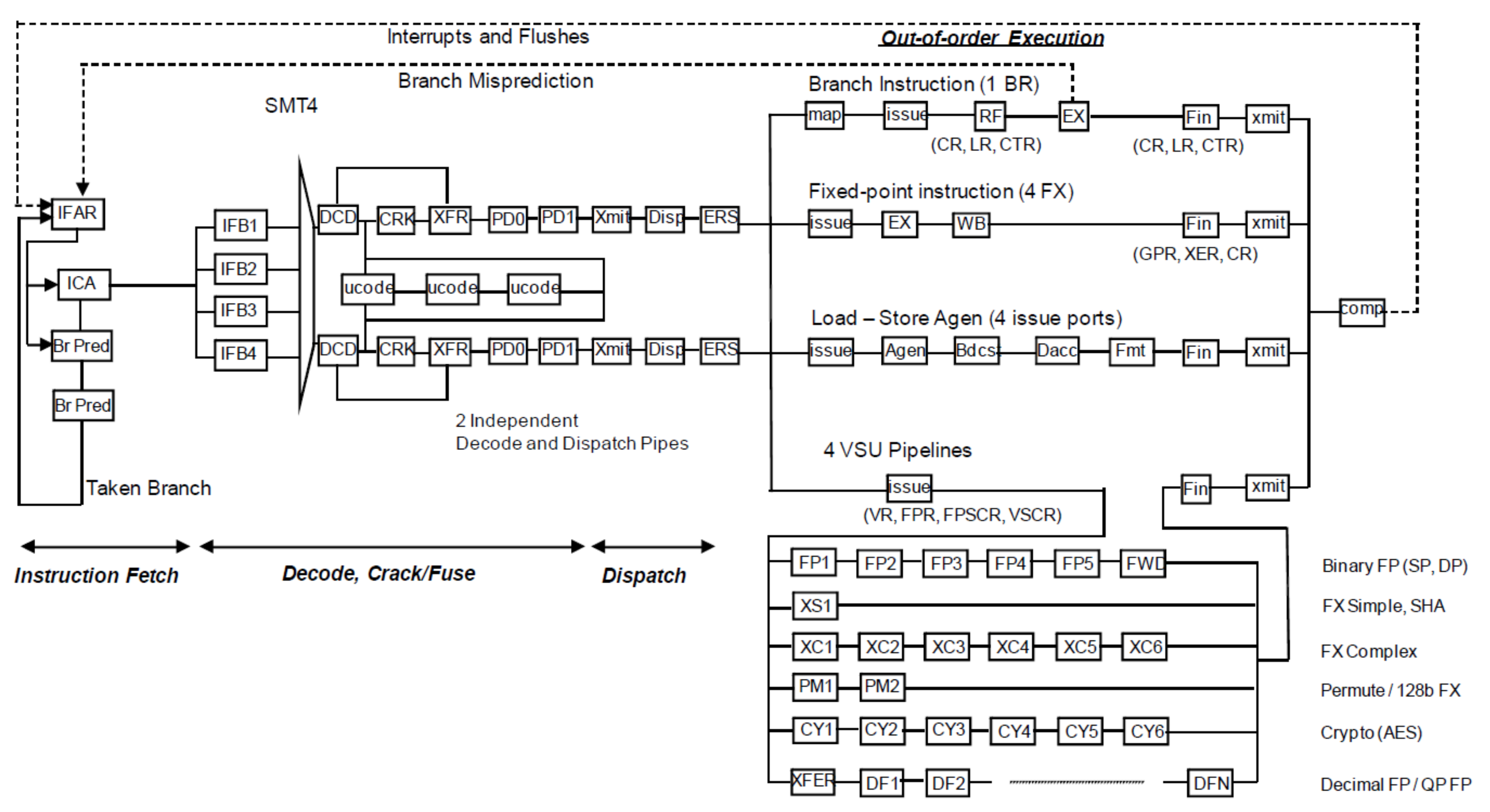




Block diagram of an SMT4 POWER9 core [154]

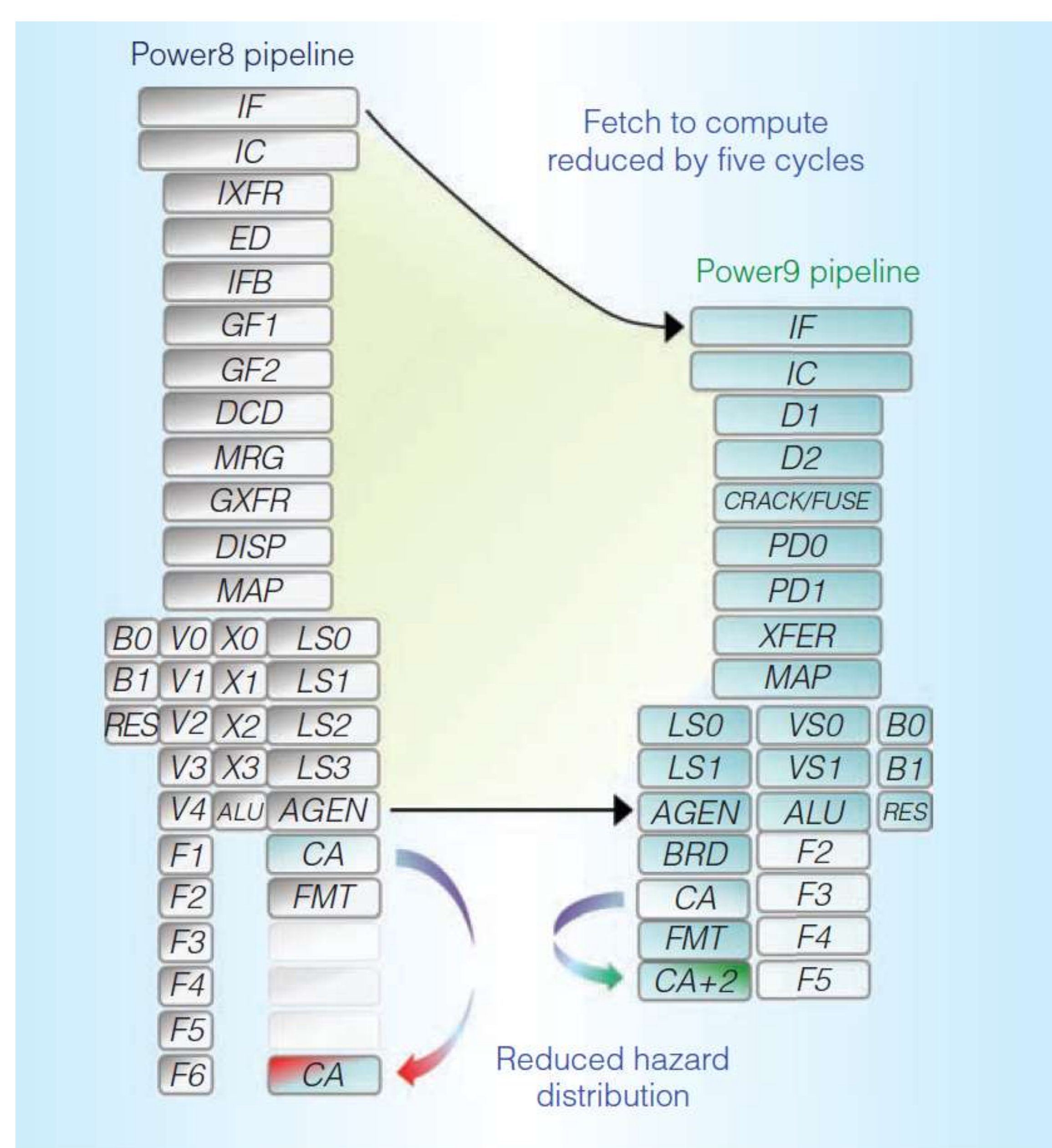


Pipeline structure of a POWER9 core [133]



12.3 Microarchitecture of the POWER9 processor (14)

Contrasting the pipeline stages of POWER8 and POWER9 [133]



On-die accelerators of the POWER9 [133]

- First add-on accelerators supporting security were introduced in the POWER7+.
- By contrast, POWER9 implements on-die accelerators for supporting security, as indicated below.

Virtualized: User mode invocation (No Hypervisor Calls)

Shared accelerators, accessible from each Thread

Accelerator Types

- Industry Standard GZIP Compression / Decompression
  - Up to 16GB/s of gzip / gunzip
- AES / SHA Cryptography Support
  - AES 128b
  - AES 256b
  - SHA 256
  - SHA 512
- Memory compression engine
- True Random Number Generation
- Data Mover

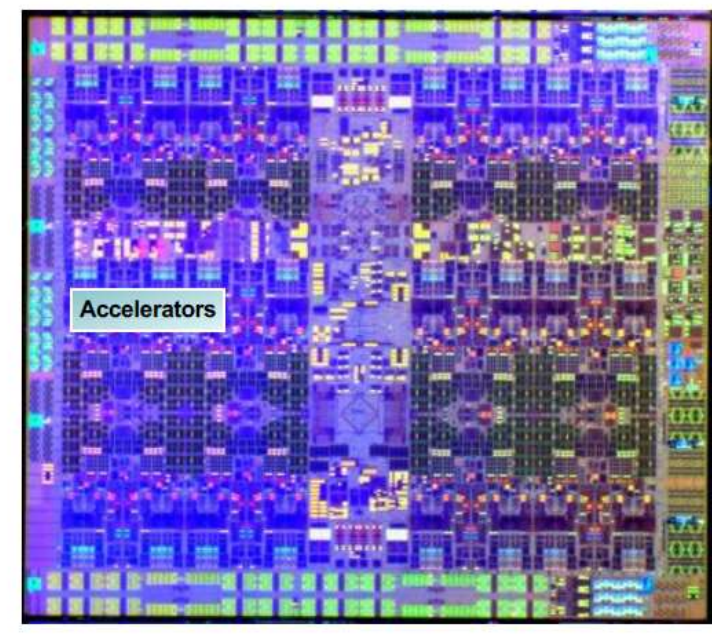
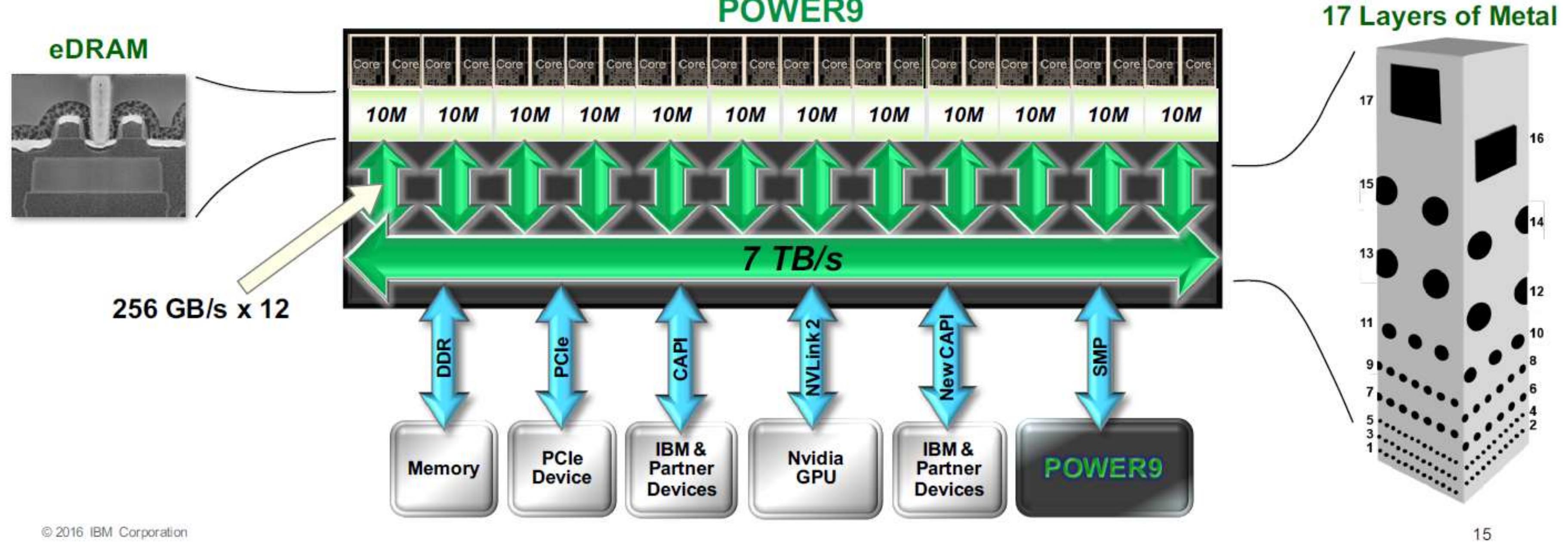


Figure: On-die accelerators of the POWER9 supporting security [133]

The on-chip fabric of the POWER9 [140] -1



horizontal buses?

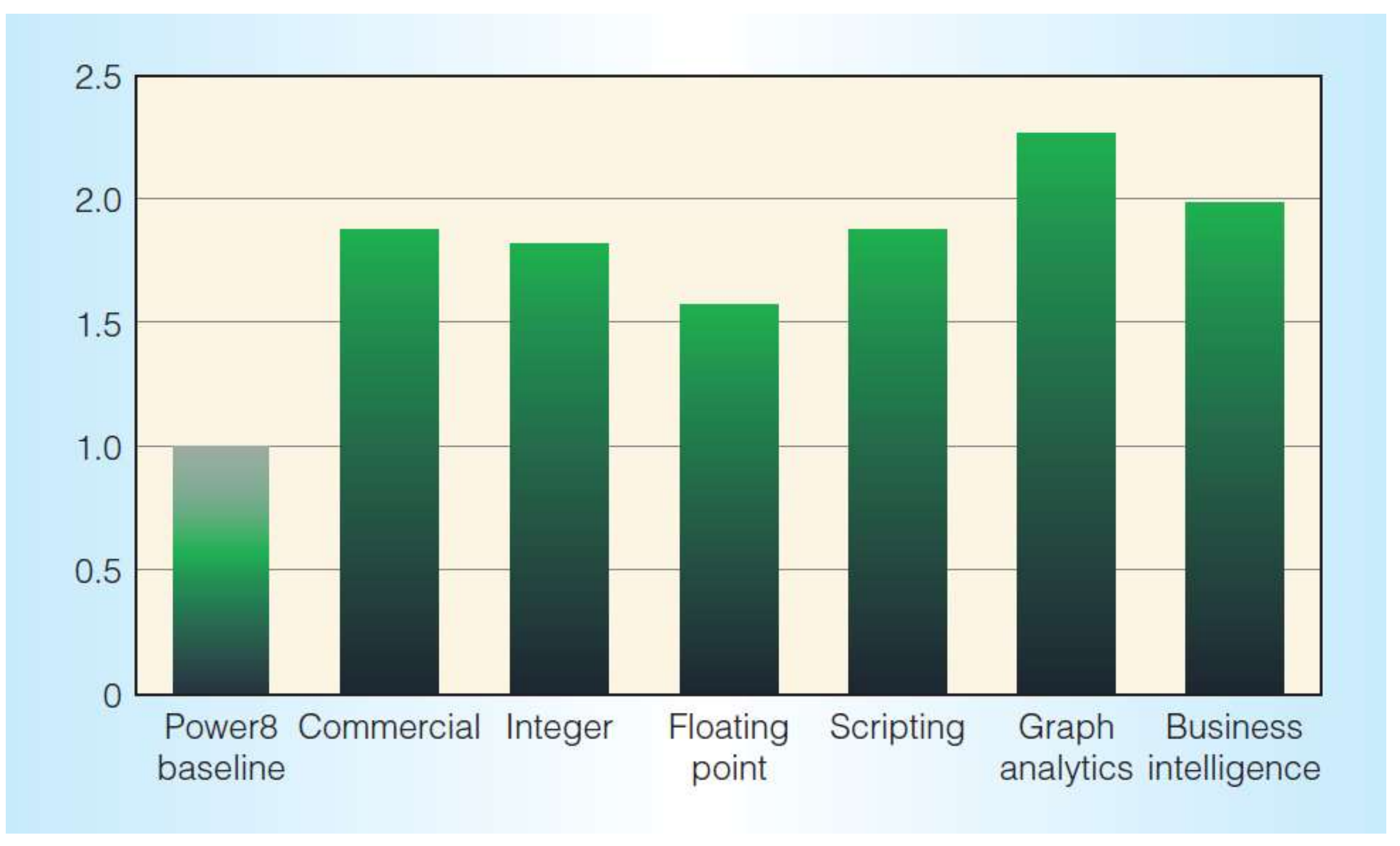
NUCA:Non-Uniform Cache Architecture (IBM patent **US9152569B2)**

The on-chip fabric of the POWER9 [154] -2

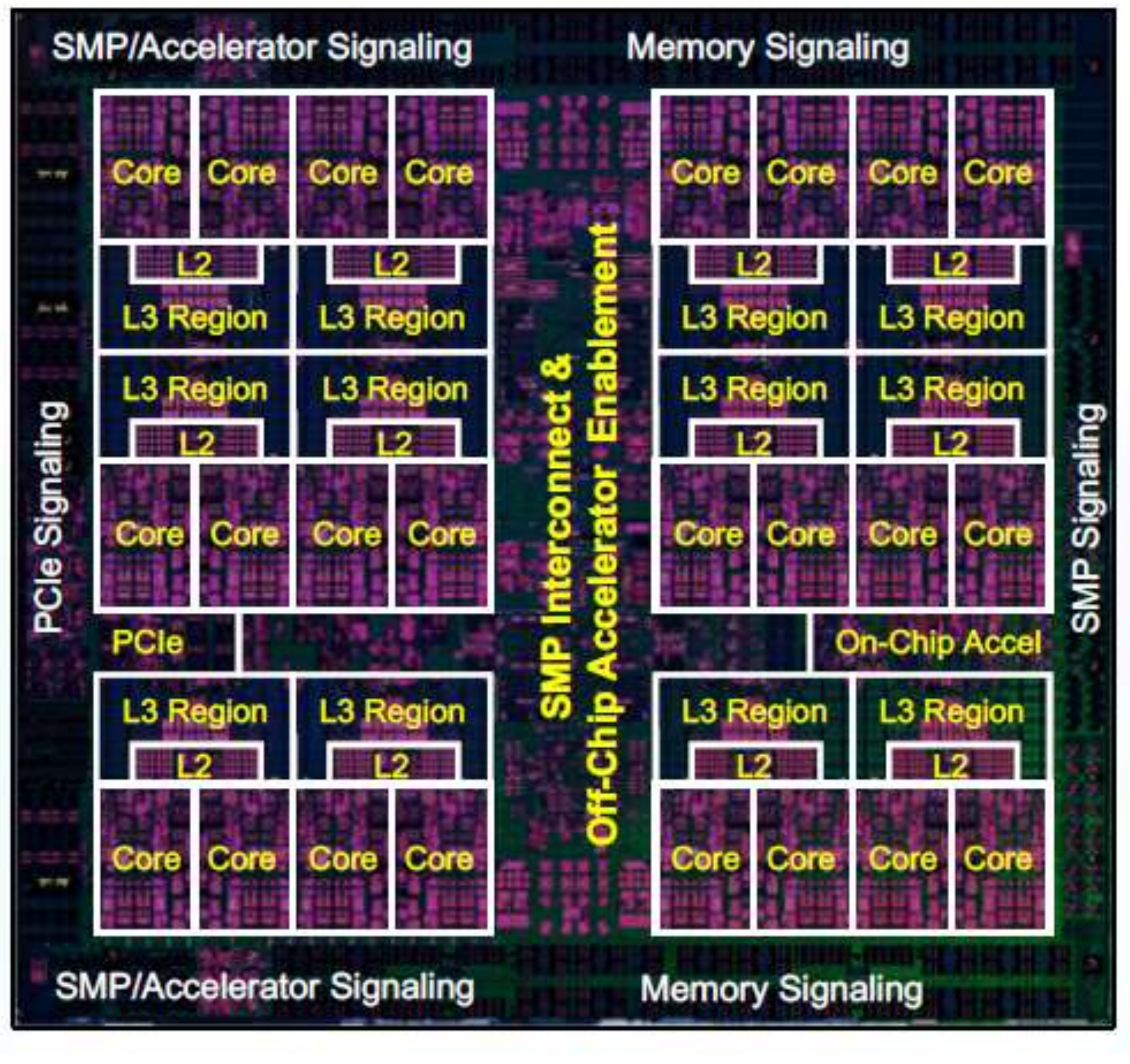
Main features

- 1600 - 2400 MHz frequency
- Eight 32-byte data buses (i.e. four ring buses)
- Four address snoop buses (rather than two as with the POWER8)
- 12 or 24 core ramps
- Fifteen nest ramps
- POWER9 SMP off-chip interconnect
  - Two 30-bit + 2 spare electrical X buses, differential, at 16 GT/s
  - Maximum two socket SMP

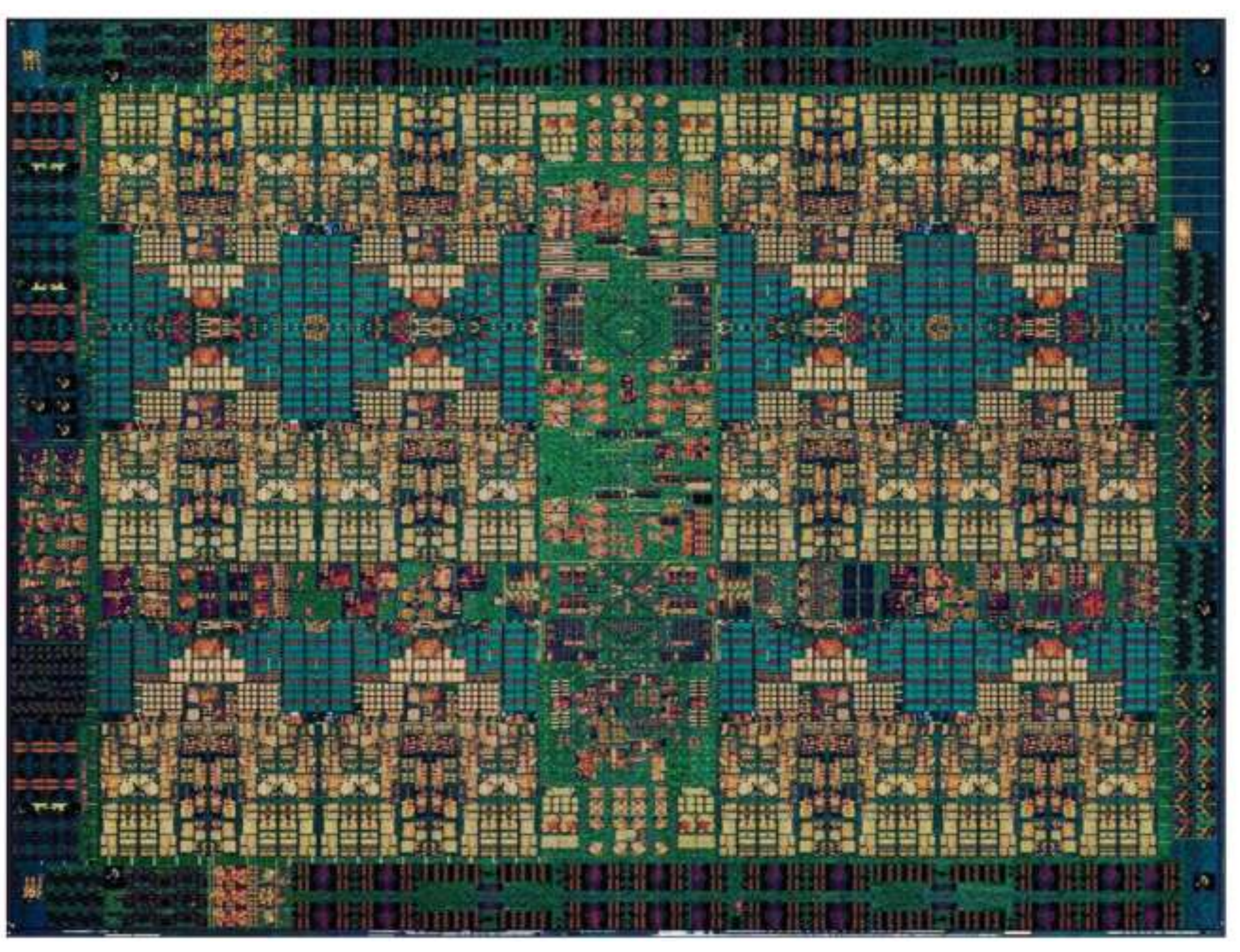
Performance increase of POWER9 vs. POWER8 [155]



Floorplan of a 24-core POWER9 die [155]



Micrograph of a 24-core POWER9 die [151]



12.4 Enhancements in POWER9’s EnergyScale

12.4 Enhancements in POWER9's EnergyScale [156]

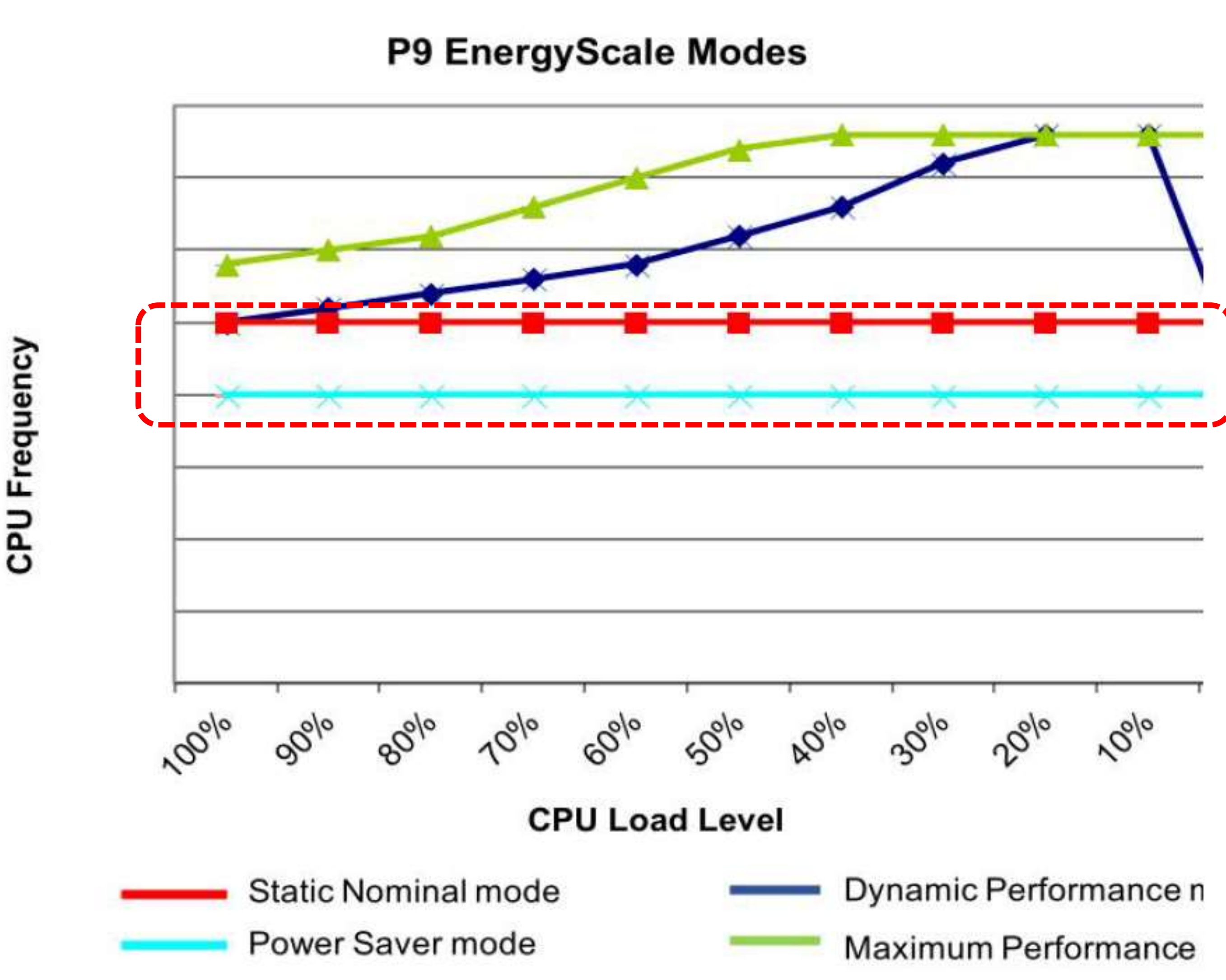
In POWER8 EnergyScale included two dynamic modes

- the Dynamic Power Saver – Favor Power, and
- the Dynamic Power Saver – Favor Performance.

modes.

- These modes changed in POWER9, as described below beyond the static EnergyScale modes.

Static EnergyScale modes of the POWER9 [156] -1



12.4 Enhancements in POWER9's EnergyScale (3)

Static EnergyScale modes of the POWER9 [156] -2

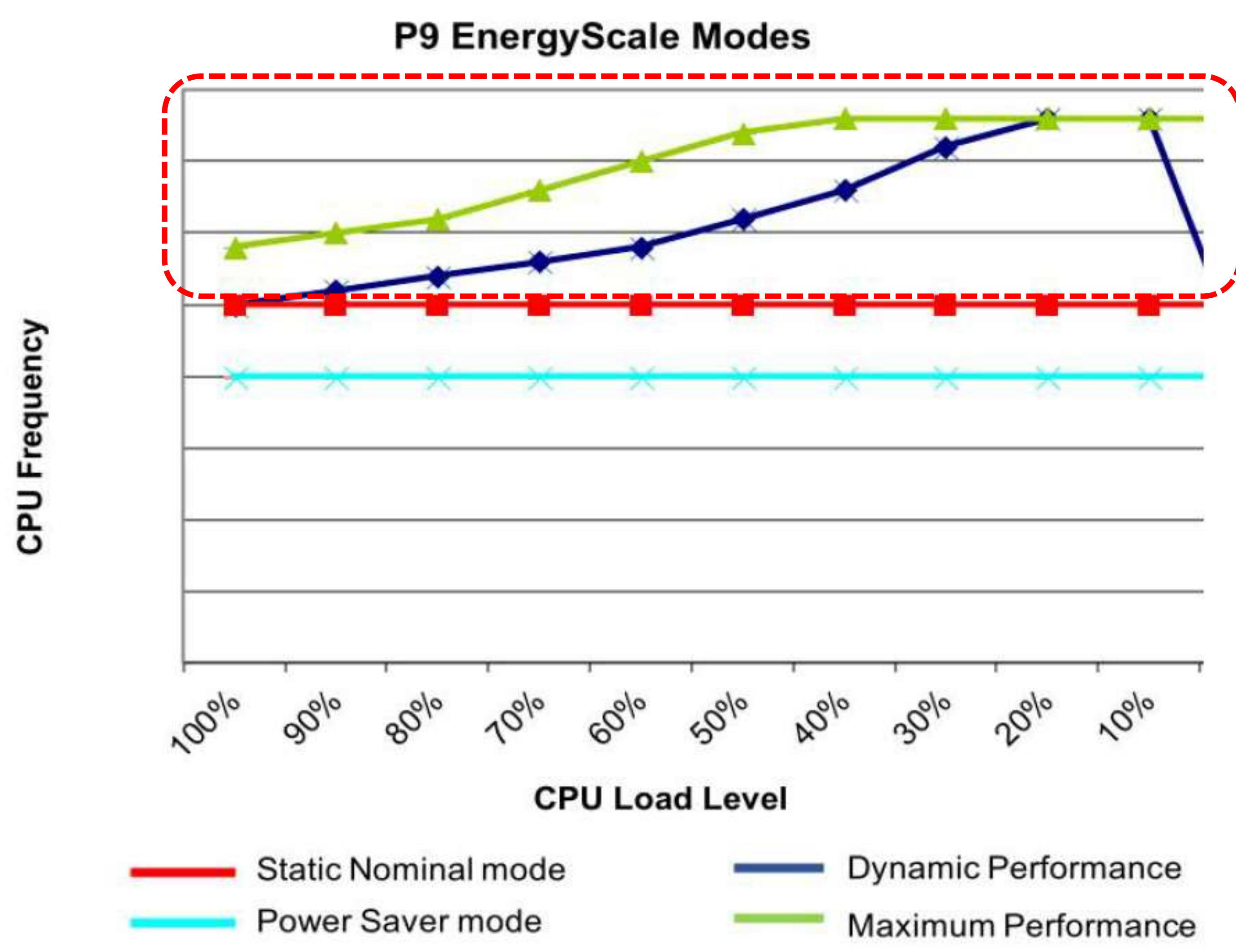
Fixed nominal frequency mode (called also as the Static Noinal mode)

- The nominal frequency is the guaranteed frequency the system will run at when operating within the specified environmental parameters (meaning under the max ambient temperature and elevation).
- This option was the default for all systems prior to POWER9.

Static Power Saver mode (termed also as the Power Saver mode)

- This mode might be used for running not critical workloads, meaning that a slower clock rate can be tolerated.
- In the Static Power Saver mode, the system will run at the minimal frequency all the time, regardless of workload.
- This mode is intended to reduce power consumption, i.e. electricity costs.

New dynamic EnergyScale modes of the POWER9 [156] -1



New dynamic EnergyScale modes [156] -2

Dynamic Performance mode -1

- In this mode the system usually will run above the nominal frequency and may even reach the maximum frequency if the workloads are light enough, or a large number of cores are idle.
- In the Dynamic Frequency mode the clock frequency will be determined by the power drawn at the socket and will vary above the static nominal value depending on the available power headroom.
- If some cores are idle, the system can run at much higher frequencies before the power limit is reached.

  If there are enough idle cores clock frequency may be as high as the maximum frequency.
- Note that the frequency is managed at the socket level, so different sockets may run at different frequencies.
- EnergyScale limits the socket power draw to a base consumption that varies somewhat by processor and system.

12.4 Enhancements in POWER9's EnergyScale (6)

Dynamic Performance mode -2

- Also, Dynamic Performance mode is the only mode that lowers the processor frequency if the entire processor socket is idle for 100s of milliseconds, thus, providing both performance boost and power savings when possible.
- Dynamic Performance mode is beneficial in cases when the best performance should be achieved that is possible across the full range of environmental conditions, but there are acoustic concerns.
- Dynamic Performance mode is the default mode for the S914 system.
- All other POWER9 scale-out systems default to the Maximum Performance mode.

Maximum Performance mode

- It takes advantage of lower active core counts and normal utilization workloads, like the Dynamic Performance mode) but it will allow the system to reach the maximum frequency under more relaxed conditions.
- There are the same constraints of power as before, but the system takes extra steps to extend the limit.
- In the Maximum Performance mode the voltage regulators will allow the socket to draw more power, e.g. 100 W more, than in the other modes.
- In order to provide adequate cooling, Maximum Performance mode will increase fan speeds, which can increase the associated acoustics by up to 15 decibels, and increase the power needed by the fans.
- If the datacenter's ambient environment is less than 25C, the frequency in Maximum Performance mode will consistently be in the upper range of the maximum frequency (roughly 10% to 20% better than nominal).
- Note that the increased noise and power consumption varies by system model, configuration, core count, and other factors.
- Additionally, there is no power reduction in Maximum Performance mode due to idleness – the system is always keeping the frequency at the maximum value possible for the running workload.
- Maximum Performance mode is best for customers who have no acoustic concerns, are in favorable ambient conditions, and want top performance.

Clock rate relations of the POWER9 EnergyScale modes [133]

Dynamic EnergyScale modes

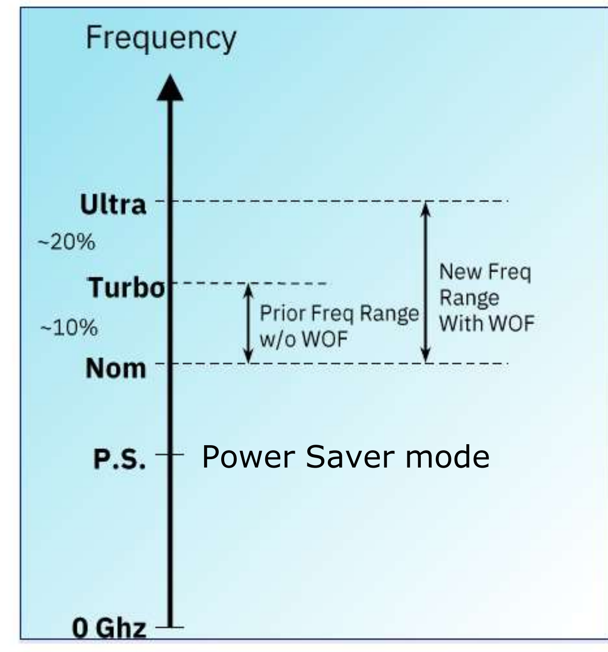
- Enable higher dynamic operational frequencies
  - For lighter workloads that do not fully utilize the core
  - For cases when all cores are not active
  - For systems in nominal operating conditions

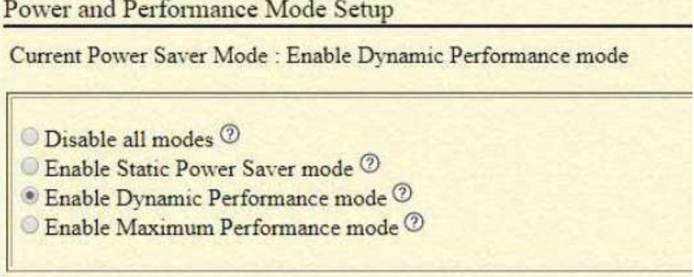
- Modes of Operation
  - Power Save Mode – Static frequency operation
  - Nominal Mode – Static frequency operation

  - Nominal Dynamic Performance Mode (Turbo)
    - CPU managed to Nominal power draw
    - Max Workload/Max Cores will run at least Nom Freq
    - Lighter workloads/Less cores will run at higher Freq

  - Maximum Dynamic Performance Mode (Ultra)
    - Same as Favor Perf Mode but with WOF enabled
    - Higher acoustics – CPU managed to Higher power draw
    - Max Workload/Max Cores runs at least Turbo Freq
    - Lighter workloads/Less cores will run at higher Freq

It does not take a reboot to change modes





12.4 Enhancements in POWER9's EnergyScale (9)

Example: Clock frequencies of POWER9 Scale-Out systems [156]

	Default Mode	Feature Code	Number of Cores	Static Nominal Frequency	Dynamic Performance Freq Range	Max Perfo Typical Ran
S924/H924	Max Performance	EP1G	12 cores	2.75 GHz	2.75 to 3.9 GHz (max)	3.4 to 3.9 G
		EP1F	10 cores	2.9 GHz	2.9 to 3.9 GHz (max)	3.5 to 3.9 G
		EP1E	8 cores	3.3 GHz	3.3 to 4.0 GHz (max)	3.8 to 4.0 G
S914	Dynamic Performance	EP12	8 cores	2.8 GHz	2.8 to 3.8 GHz (max)	3.15 to 3.8 C
		EP11	6 cores	2.3 GHz	2.3 to 3.8 GHz (max)	2.8 to 3.8 G
		EP10	4 cores	2.3 GHz	2.3 to 3.8 GHz (max)	2.8 to 3.8 G
S922/H922	Max Performance	EP19	10 cores	2.5 GHz	2.5 to 3.8 GHz (max)	2.9 to 3.8 G
		EP18	8 cores	3.0 GHz	3.0 to 3.9 GHz (max)	3.4 to 3.9 G
		EP16	4 cores	2.3 GHz	2.3 to 3.8 GHz (max)	2.8 to 3.8 G
L922	Max Performance	ELPX	12 cores	2.3 GHz	2.3 to 3.8 GHz (max)	2.7 to 3.8 G
		EPPW	10 cores	2.5 GHz	2.5 to 3.8 GHz (max)	2.9 to 3.8 G
		ELPV	8 cores	3.0 GHz	3.0 to 3.9 GHz (max)	3.4 to 3.9 G

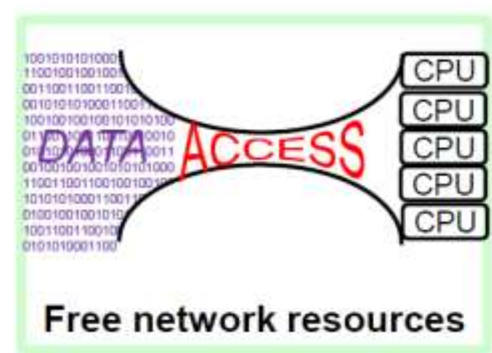
Note 1: Frequencies outlined in Red reflect the default mode (i.e. frequency range) for that particular syste

Note 2: In order to reach maximum frequency, some cores may need to be turned off

12.5 POWER9 as a platform for accelerated computing

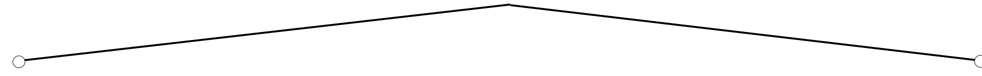
12.5 POWER9 as a platform for accelerated computing

Possible aims of using accelerators [157]



External accelerator support of processors

External accelerator support of processors



External accelerator agnostic processors

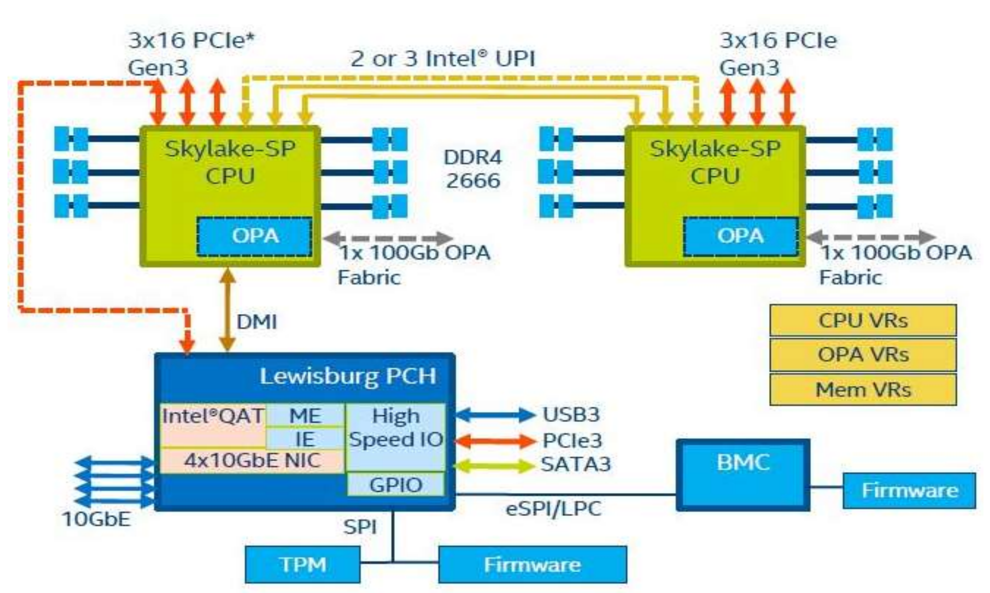
Moving work off the CPU
by using on-die or in-package accelerators

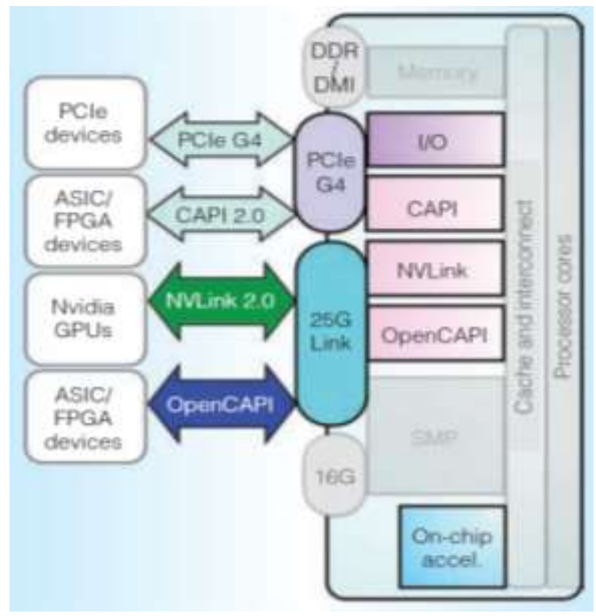
E.g.
- *Intel's recent servers*
- *IBMs POWER processors prior the PPWER9*

External accelerator oriented processors

Moving work off the processor
by using off-chip accelerators

IBM's POWER9

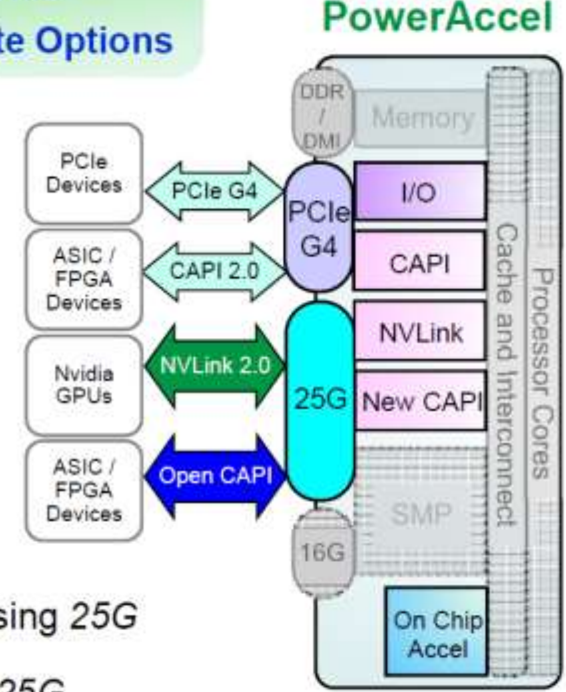




I/O links provided on the POWER9 [140]

- **Extreme Processor / Accelerator Bandwidth and Reduced Latency**
- **Coherent Memory and Virtual Addressing Capability for all Accelerators**
- **OpenPOWER Community Enablement – Robust Accelerated Compute Options**

- **State of the Art I/O and Acceleration Attachment Signaling**
  - **PCIe Gen 4** x 48 lanes – 192 GB/s duplex bandwidth
  - **25Gb/s Common Link** x 48 lanes – 300 GB/s duplex bandwidth
- **Robust Accelerated Compute Options with OPEN standards**
  - **On-Chip Acceleration** – Gzip x1, 842 Compression x2, AES/SHA x2
  - **CAPI 2.0** – 4x bandwidth of POWER8 using *PCIe Gen 4*
  - **NVLink 2.0** – Next generation of GPU/CPU bandwidth and integration using *25G*
  - **Open CAPI 3.0** – High bandwidth, low latency and open interface using *25G*



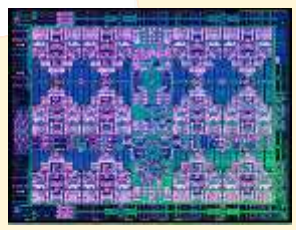
POWER9 as a platform for accelerated computing (4)

I/O links discussed in this Section

a) PCIe Gen. 4
b) 25 Gb/s BlueLink interface
c) CAPI 2.0
d) OpenCAPI (CAPI 3.0)
e) NVLink 2.0

These are the main enhancements of POWER9, as indicated in the next slide.

Key enhancements of the POWER9 (Die photo: [3])



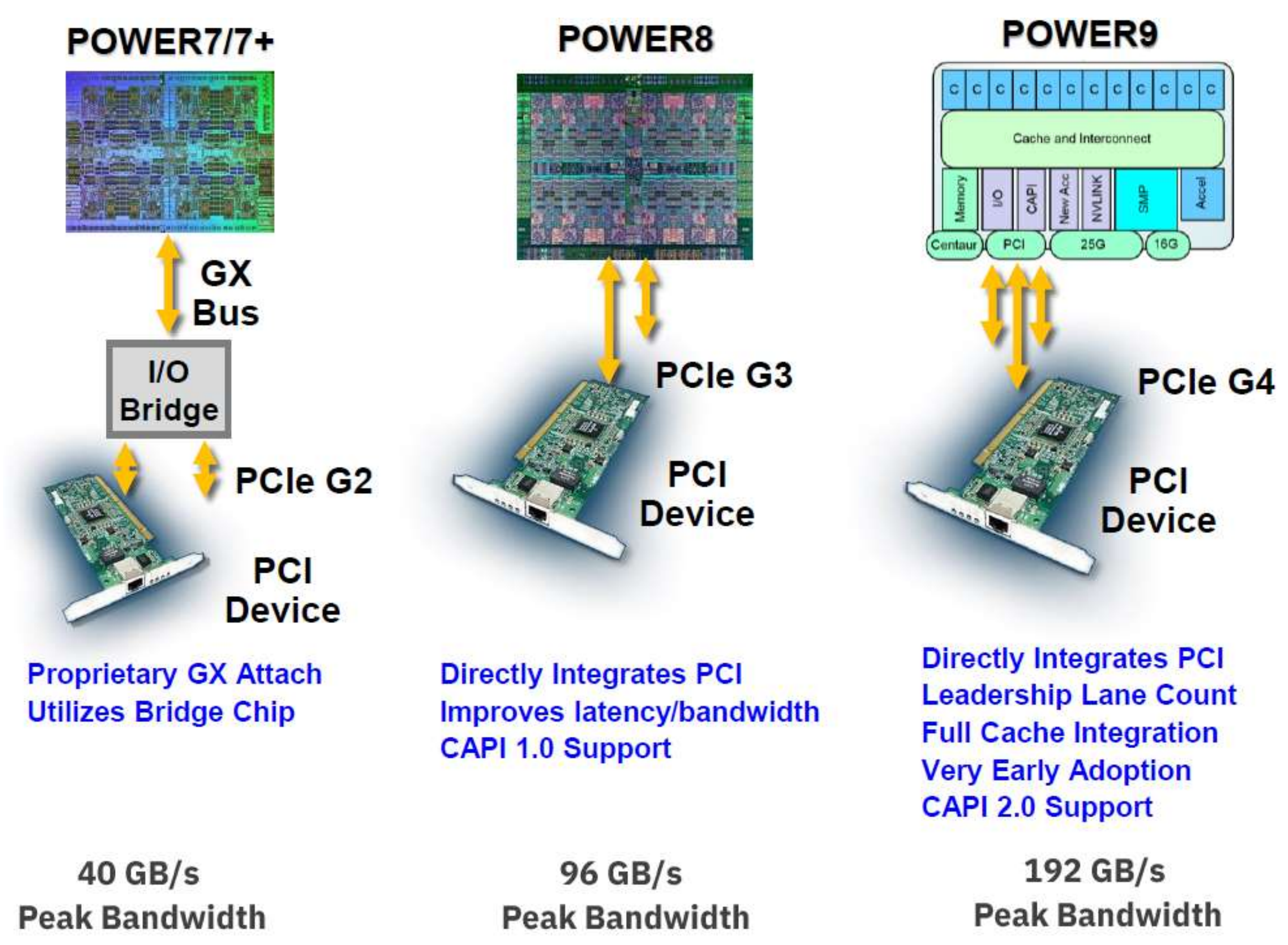
POWER9
14 nm

24 cores
4/8-way SMT
PCIe 4.0 (aka G4)
CAPI 2.0
OpenCAPI (CAPI 3.0
NVLink 2.0

2018

POWER9 as a platform for accelerated computing (6)

a) PCIe Gen. 4
   Evolution of connecting I/O from the GX-bus until PCIe 4.0 [133]



b) The 25 GT/s BlueLink interface [158]

- It has 8 signal lanes in each direction, operating at 200 GT/s.
- This results in 25 GB/s in each direction.
- IBM terms BlueLink also as AXON indicating its use for A-bus (inter-node bus), X-bus, (intra-node bus), OpenCAPI and NVLink).

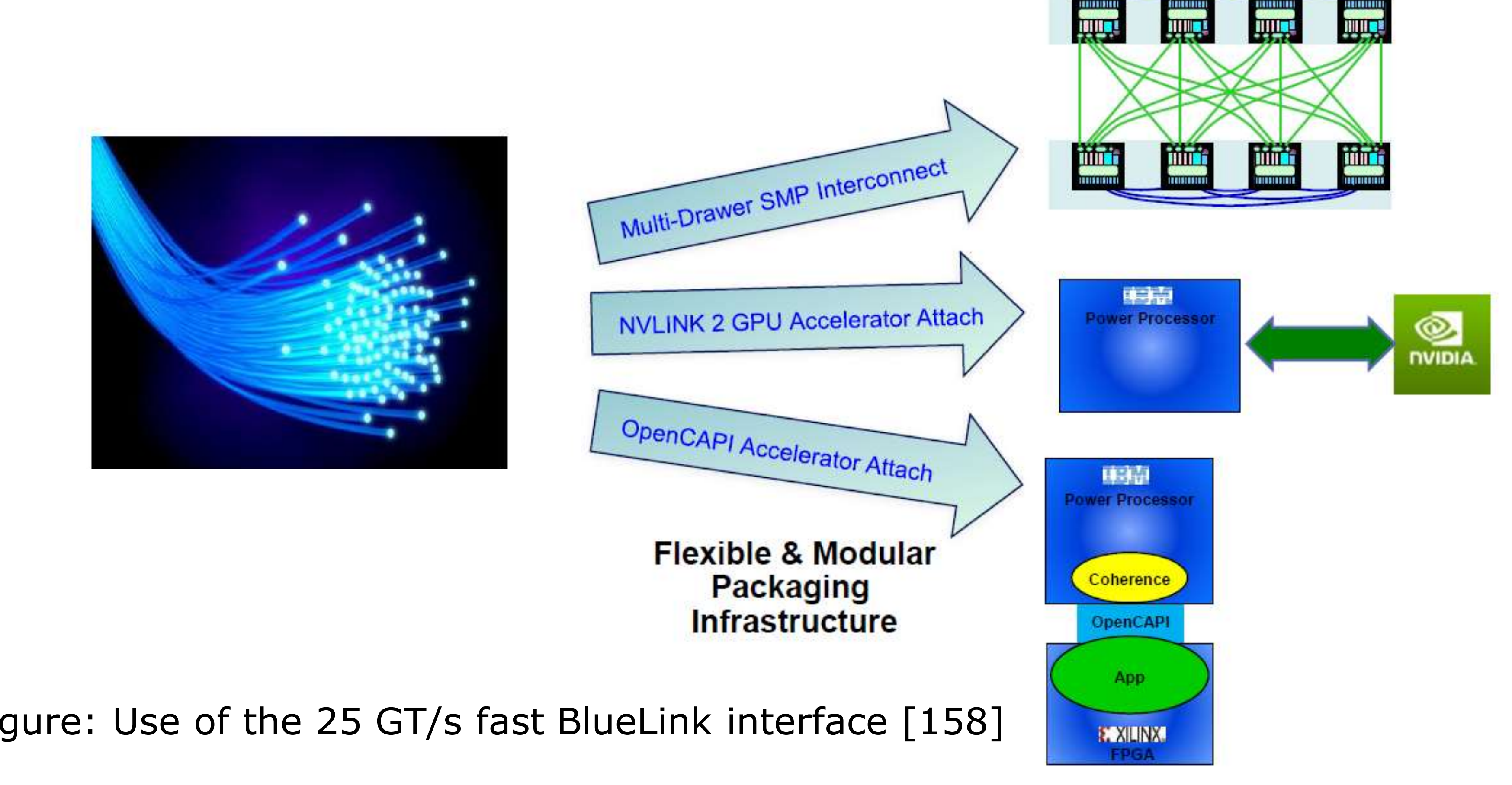
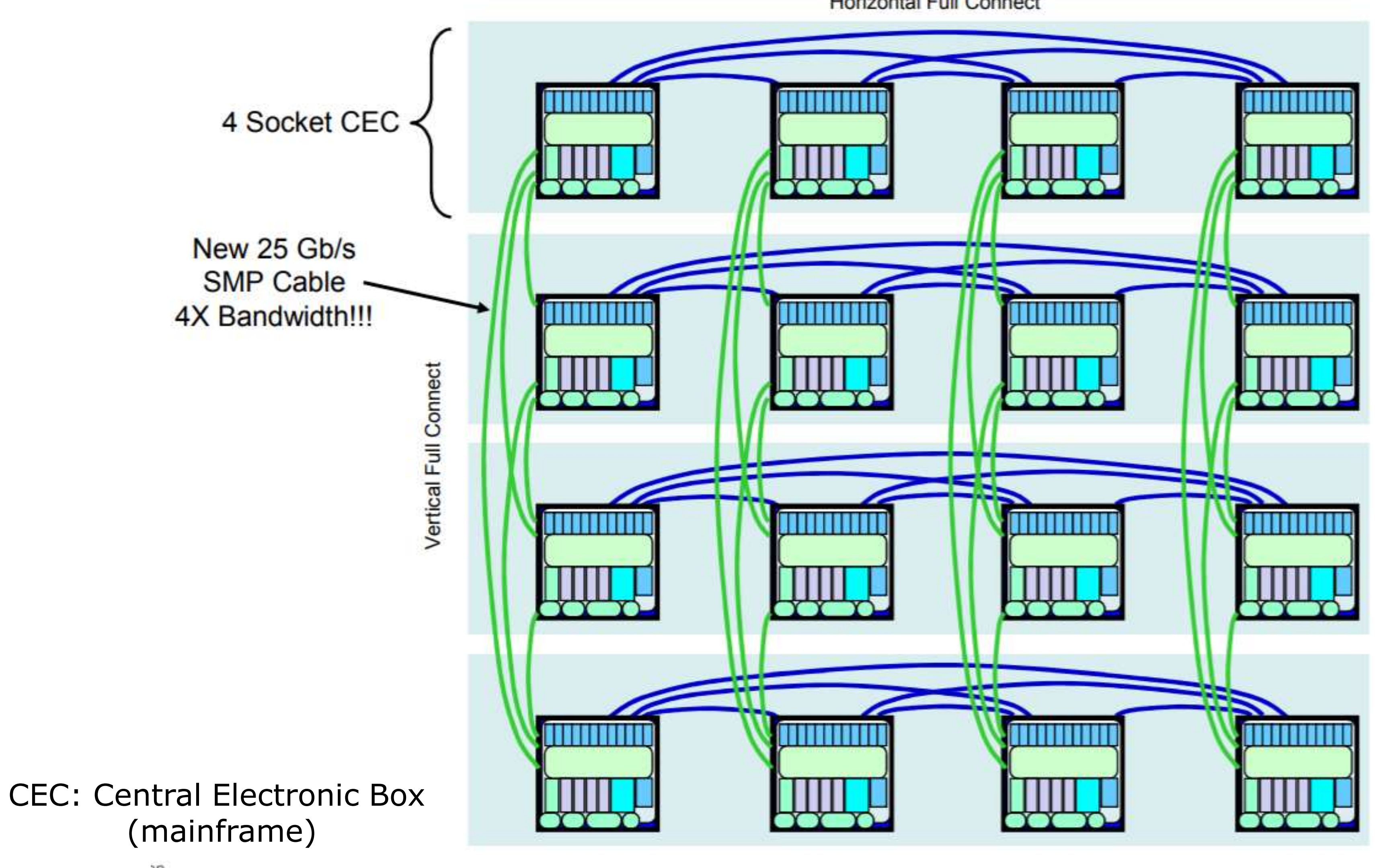


Figure: Use of the 25 GT/s fast BlueLink interface [158]

POWER9 as a platform for accelerated computing (8)

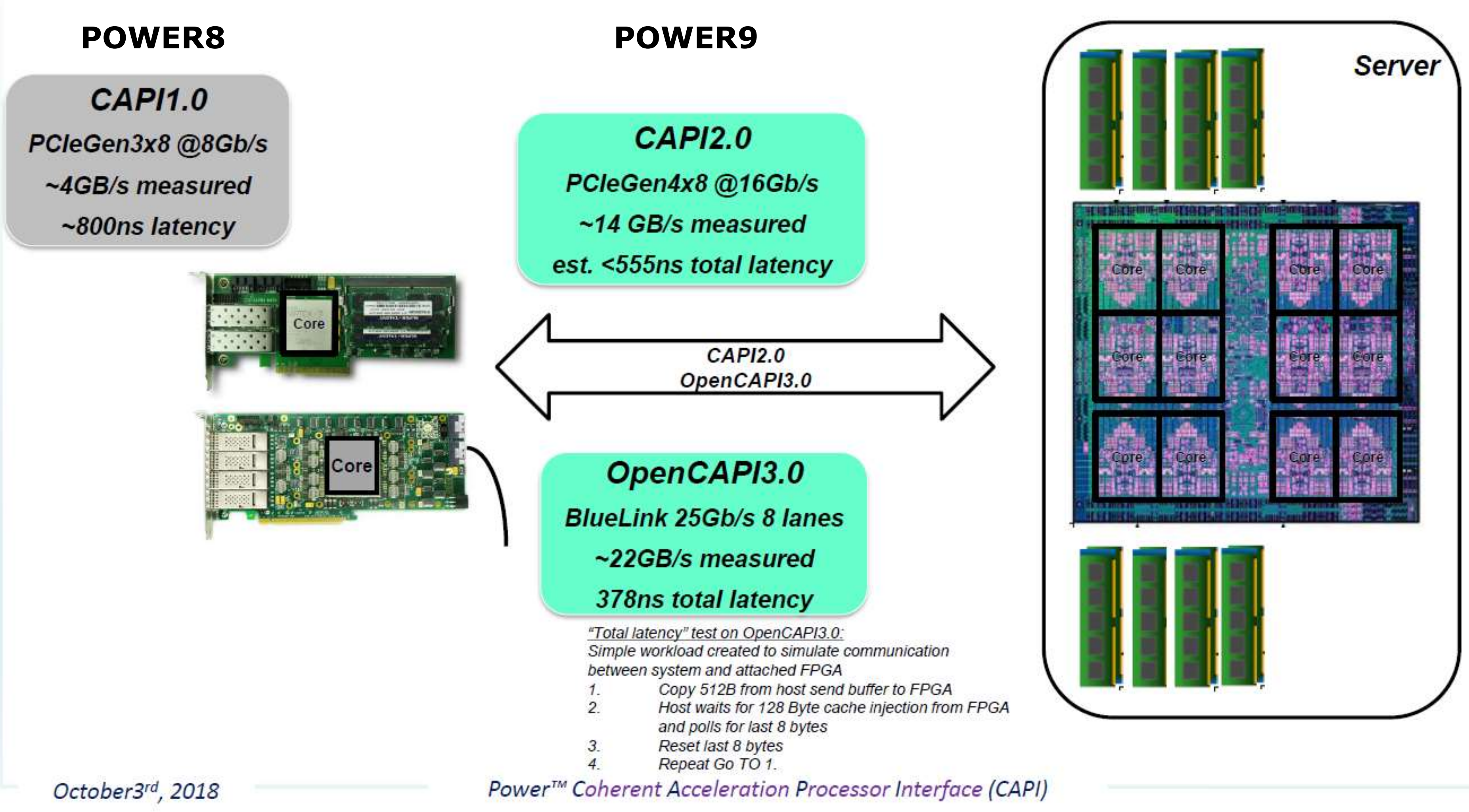
Use of BlueLink links for interconnecting a 16 socket POWER9 system by 2-hops [158]



CEC: Central Electronic Box (mainframe)

c) CAPI 2.0

Evolution of the Coherent Accelerator Processor Interface (CAPI) [157]



d) OpenCAPI (CAPI 3.0) [159]

POWER9 provides up to

- up to 48 PCIe 4.0 lanes (16 Gb/s) with up to 32 lanes supporting CAPI 2.0 and
- up to 48 BlueLink lanes (25 Gb/s) with up to 32 lanes supporting OpenCAPI 3.0 whereas all lanes can be used for NVLink 2.0, as indicated below.

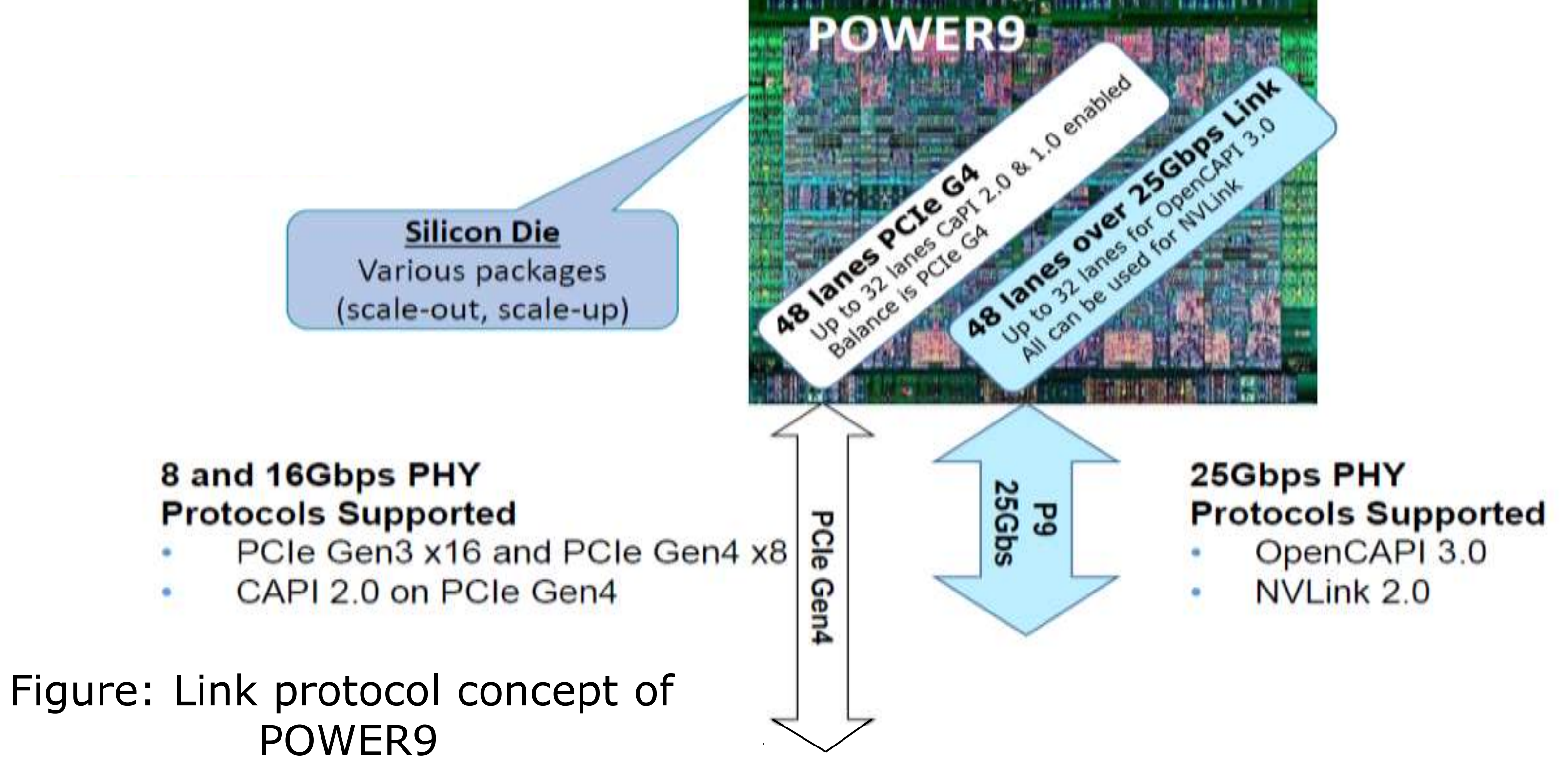


Figure: Link protocol concept of POWER9

POWER9 as a platform for accelerated computing (11)

OpenCAPI benefits [159]

- Architecture agnostic bus – applicable with any system/microprocessor architecture
- Optimized for high bandwidth and low latency
- High performance 25 Gbps design, called BlueLink, with zero ‘overhead’
- Coherency - Attached devices operate natively within application’s user space and coherently with host microprocessor
- Virtual addressing enables low overhead with no kernel, hypervisor or firmware involvement
- CPU coherent device memory (Home Agent Memory)
- Minimal OpenCAPI design overhead (FPGA less than 5%)

POWER9 as a platform for accelerated computing (12)

Remark

- The OpenCAPI Consortium was founded in 9/2016 by AMD, Google, IBM, Mellanox, and Micron.
- In 3/2018 it has over 35 members.

Hardware components of CAPI

CAPI incorporates three essential components:

- CAPP: Coherent Accelerator Processor Proxy unit
- PHB: PCIe Host Bridge
- PSL: Power Service Layer

as shown below.

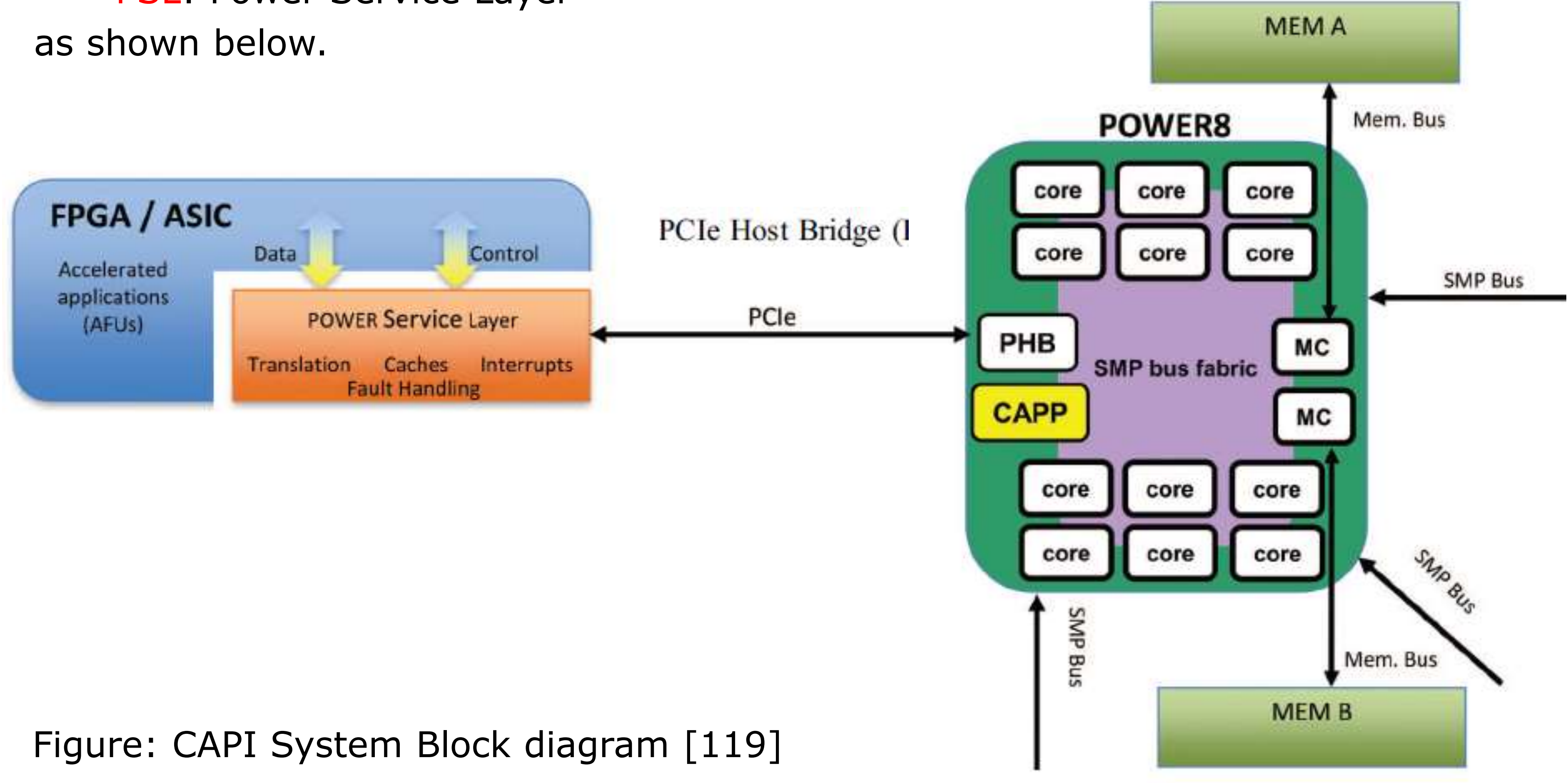
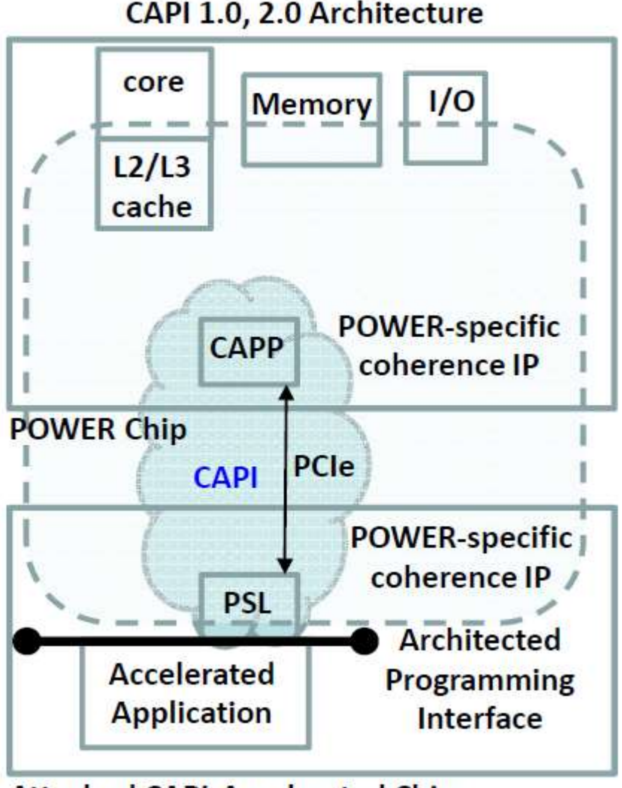
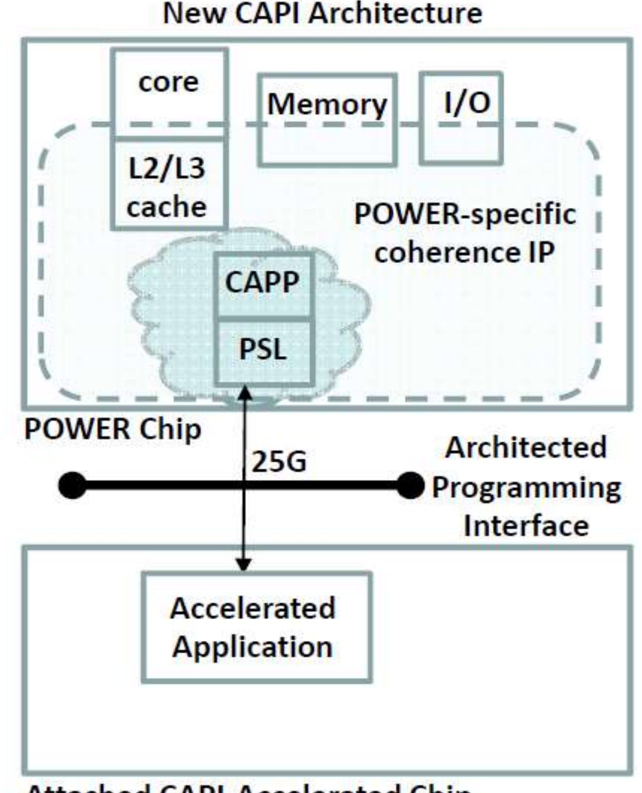


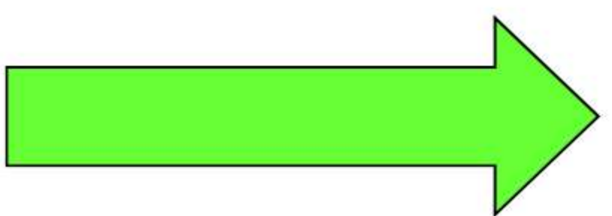
Figure: CAPI System Block diagram [119]

POWER9 as a platform for accelerated computing (14)

Evolution of the architecture of CAPI 1.0/2.0 to OpenCAPI (CAPI 3.0)) [160] -1

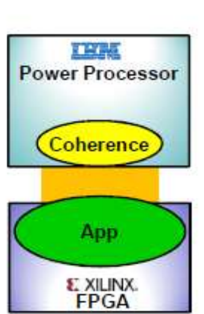






Open Industry Coherent Attach

- Latency / Bandwidth Improvement
- Removes Overhead from Attach Silicon
- Eliminates "Von-Neumann Bottleneck"
- FPGA / Parallel Compute Optimized
- Network/Memory/Storage Innovation



Contrasting the implementations of CAPI 1.0/2.0 and OpenCAPI (CAPI 3.0) (see previous Figure) [160] -2

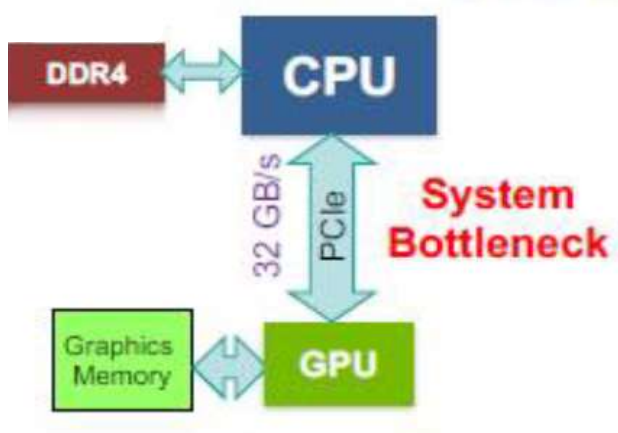
There are two key differences

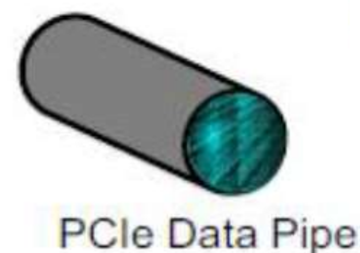
a) As long as CAPI 1.0 is connected to the processor by a PCIe 3.0 x8 link by 8 Gbps transfer rate and
CAPI 2.0 is connected to the processor by a PCIe 3.0 x8 link by 16 Gbpsx8,
OpenCAPI (aka CAPI 3.0) is connected to the processor by the Bluelink x8 interconnect by a transfer rate of 25 Gbps.

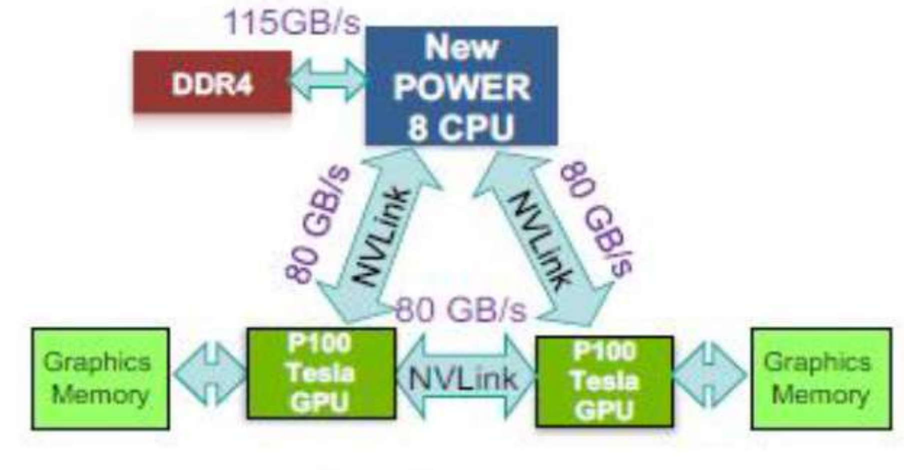
b) As long as in CAPI 1.0 and 2.0 the accelerator includes the POWER Service Layer (PSL), in OpenCAPI PSL is removed to the processor,
this eliminates the overhead to implement PSL in each accelerator and improves latency and bandwidth.

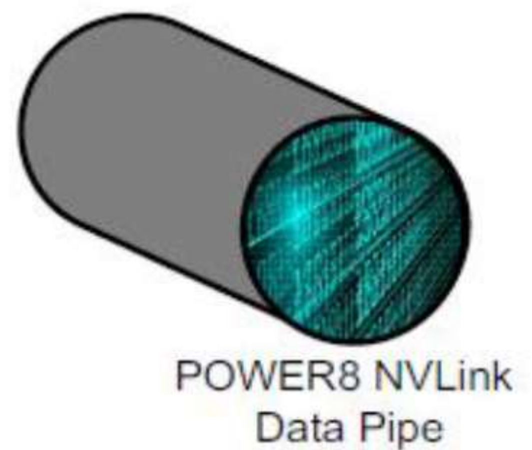
e) NVLink 2.0

Benefit of using NVLink 1.0 vs. PCIe 3.0 to connect GPUs [133]




PCIe Data Pipe



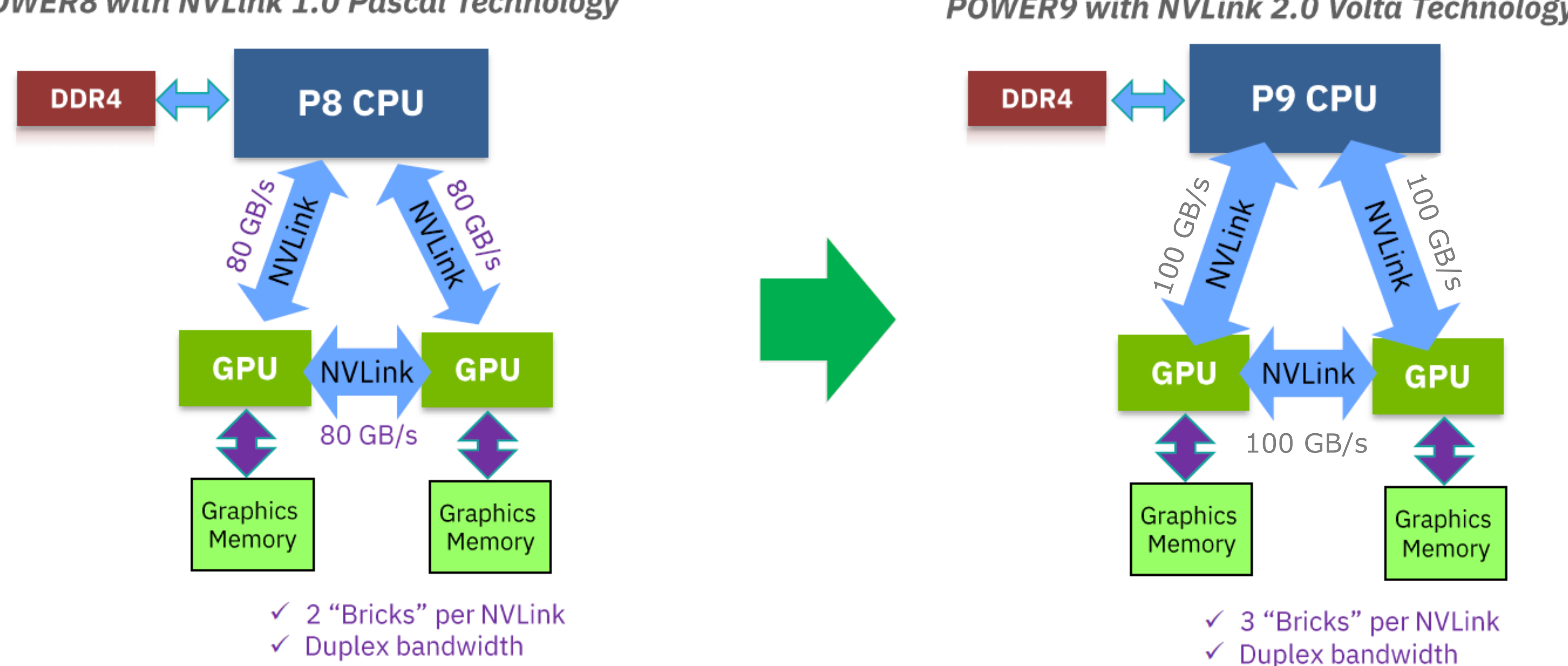

POWER8 NVLink Data Pipe

Remarks

PCIe 3.0 has a transfer rate of 8GT/s.
This yields a bandwidth for an x16 link in both directions:
2x 16x 8 GT/s = 256 Gbps = 32 GB/s

NVLink 1.0 has a data rate of 20 Gb/s.
This yields a 2.5 x higher bandwidth than PCIe 3.0.

Evolution of the bandwidth of NVLink from NVLink 1.0 to NVLink 2.0 [133]

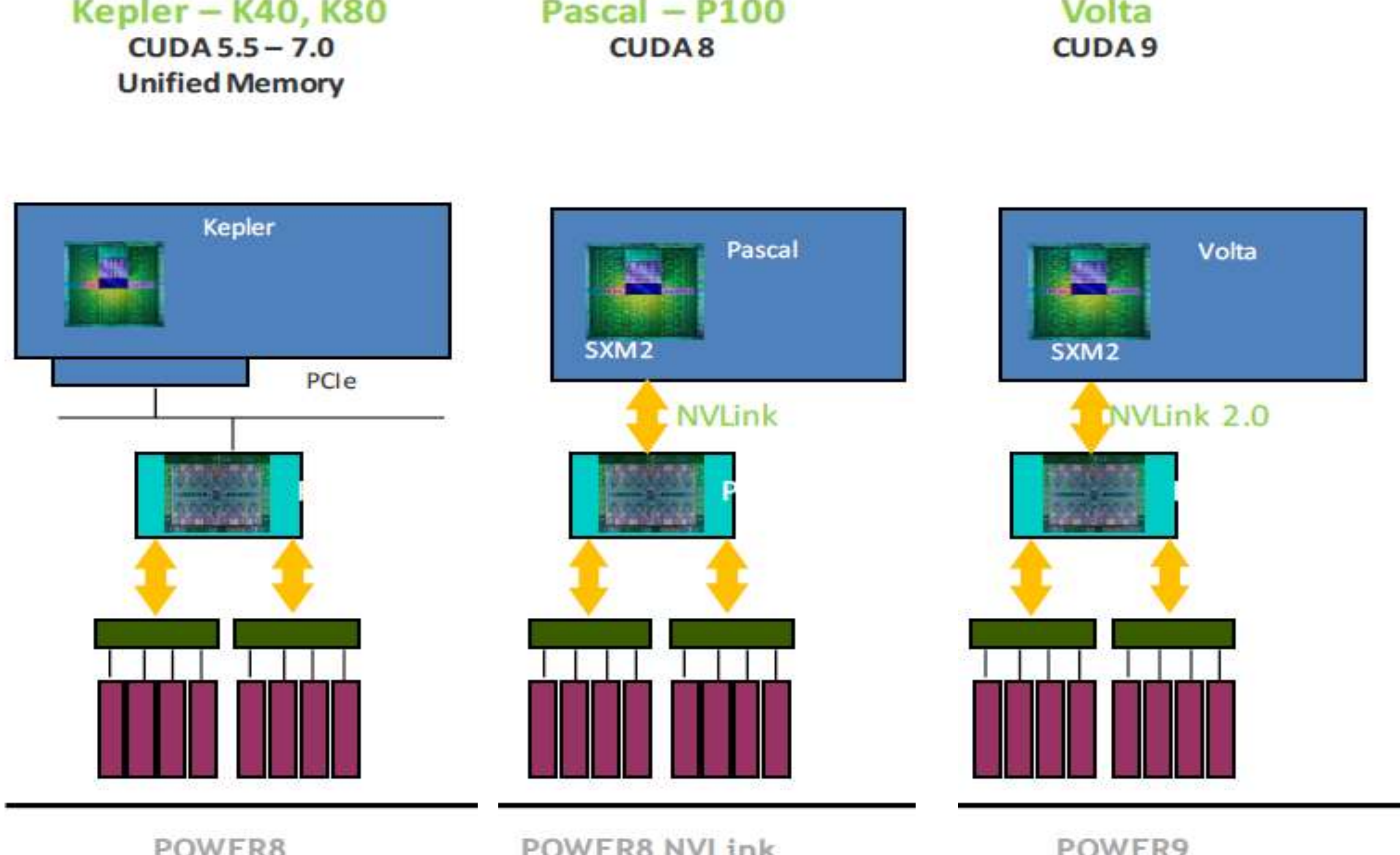


POWER9 with NVLink 2.0
delivers 25 % higher bandwidth vs. POWER8

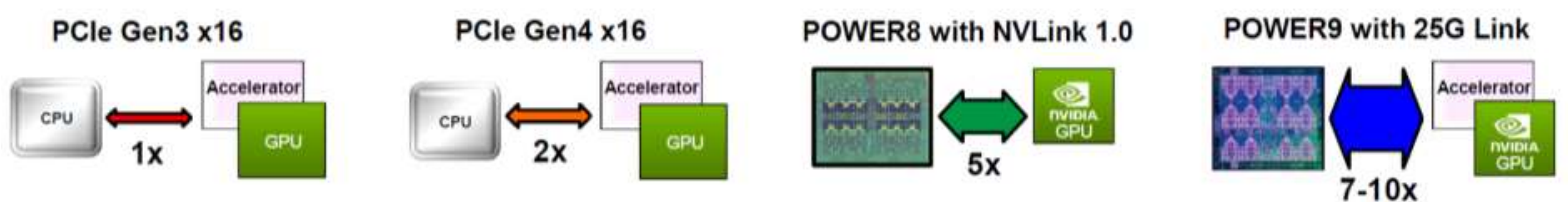
NVLink 1.0 has a data rate of 20 Gb/s.

NVLink 2.0 has a data rate of 25 Gb/s.
This provide 25 % higher bandwidth than NVLink 1.0.

Evolution of attaching GPUs in POWER8 and POWER9 processors [140]



Evolution of the CPU-Accelerator bandwidth from POWER8 to POWER9 [155]



Remark

NVMe support (on M.2) in the S922 and S924 processors in the POWER9 [133]

S922/S924 has two internal direct attached storage connectors

Connector support either:

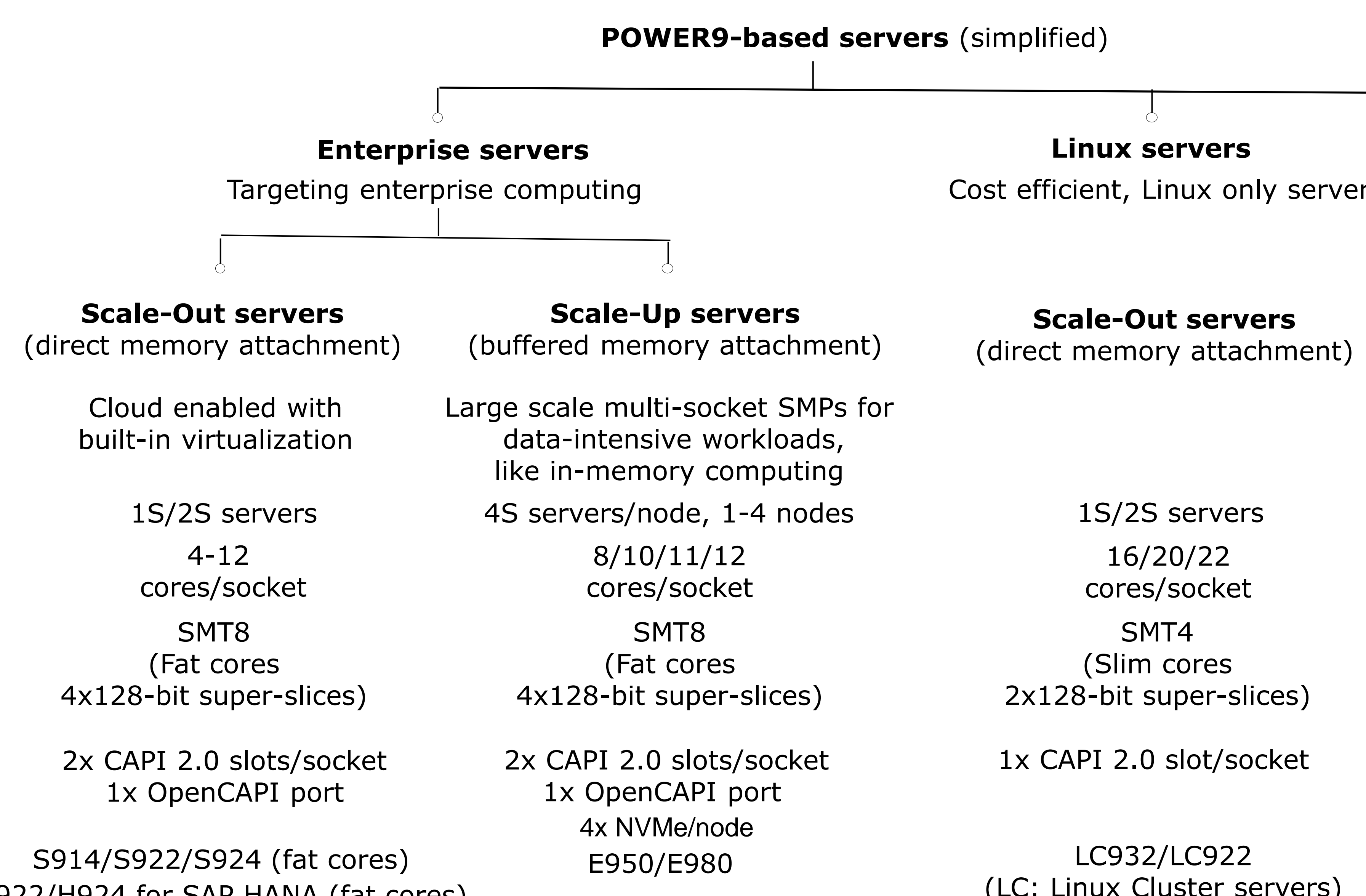
- NVMe carrier card & attaches two 400 GB M.2 NVMe drives
- SAS controller requiring DASD backplane (like POWER8)
- You can mix an NVMe card and a SAS card
- Not socket dependent

M.2 NVMe on POWER9 on S922/S924

- A maximum of four x M.2 NVMe drives
- Will be higher performance than SAS DASD in backplane
- Will not support concurrent maintenance (unlike SAS drives)
- Will have a write endurance of 1 drive write per day
- Intended primarily to store and boot OS (AIX / VIOS / Linux) images
- Each NVMe device→separate PCIe endpoint assign to different LPARs
- NVMe drives may be assigned to the VIOS and virtualized to client OS

12.6 POWER9-based servers

12.6 POWER9-based servers -1



POWER9-based servers -2

Accelerated servers
Targeting analytics, HPC, AI.
Accelerated by GPUs
(2or 3 GPUs /socket)

Scale-Out servers
(direct memory attachment)

2S servers

16/18/20
cores/socket

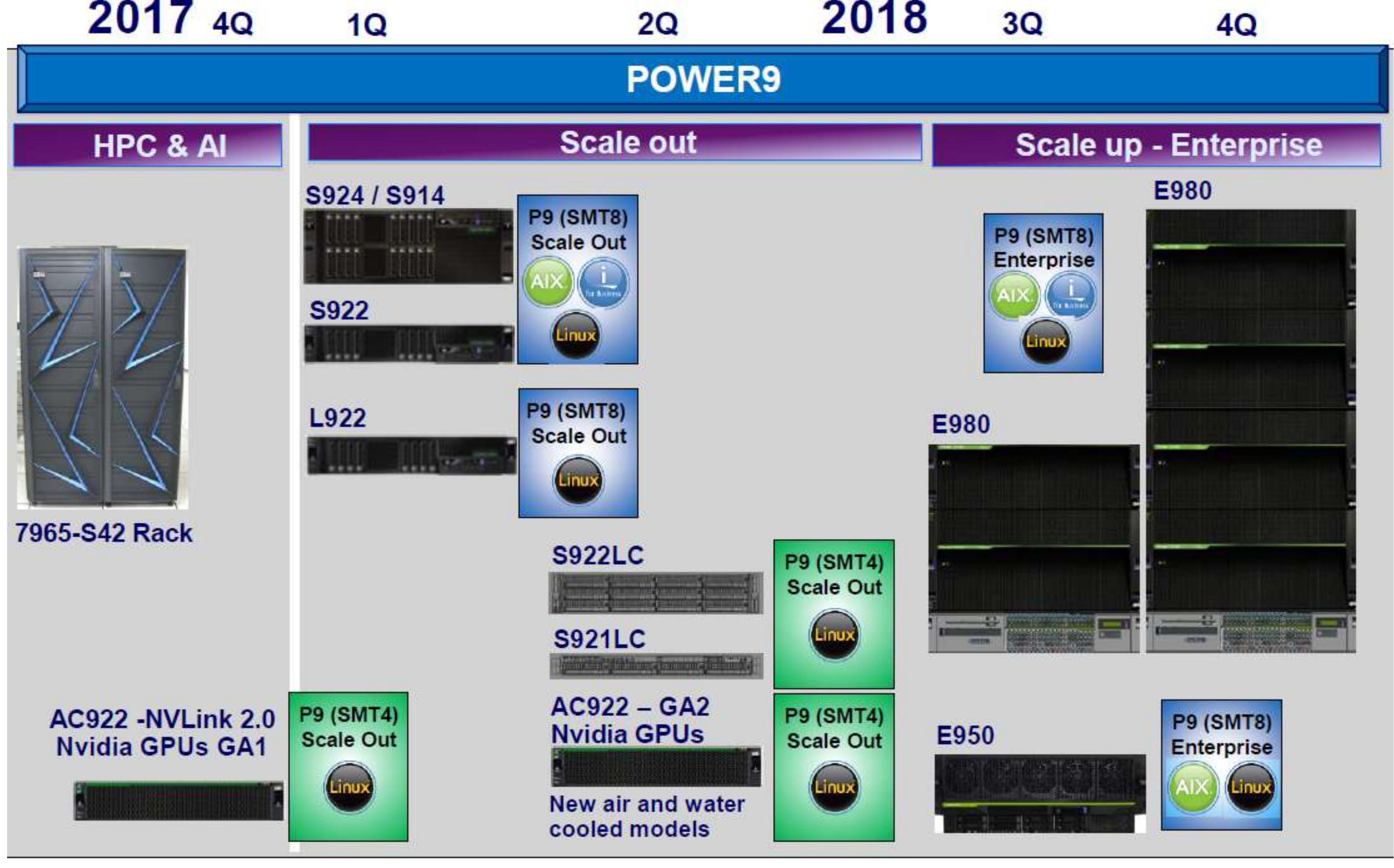
SMT4
(Slim cores
2x128-bit super-slices)

2or 3 GPUs/socket

1x CAPI slot/socket

AC922

POWER9 server roadmap from 2017 [161]



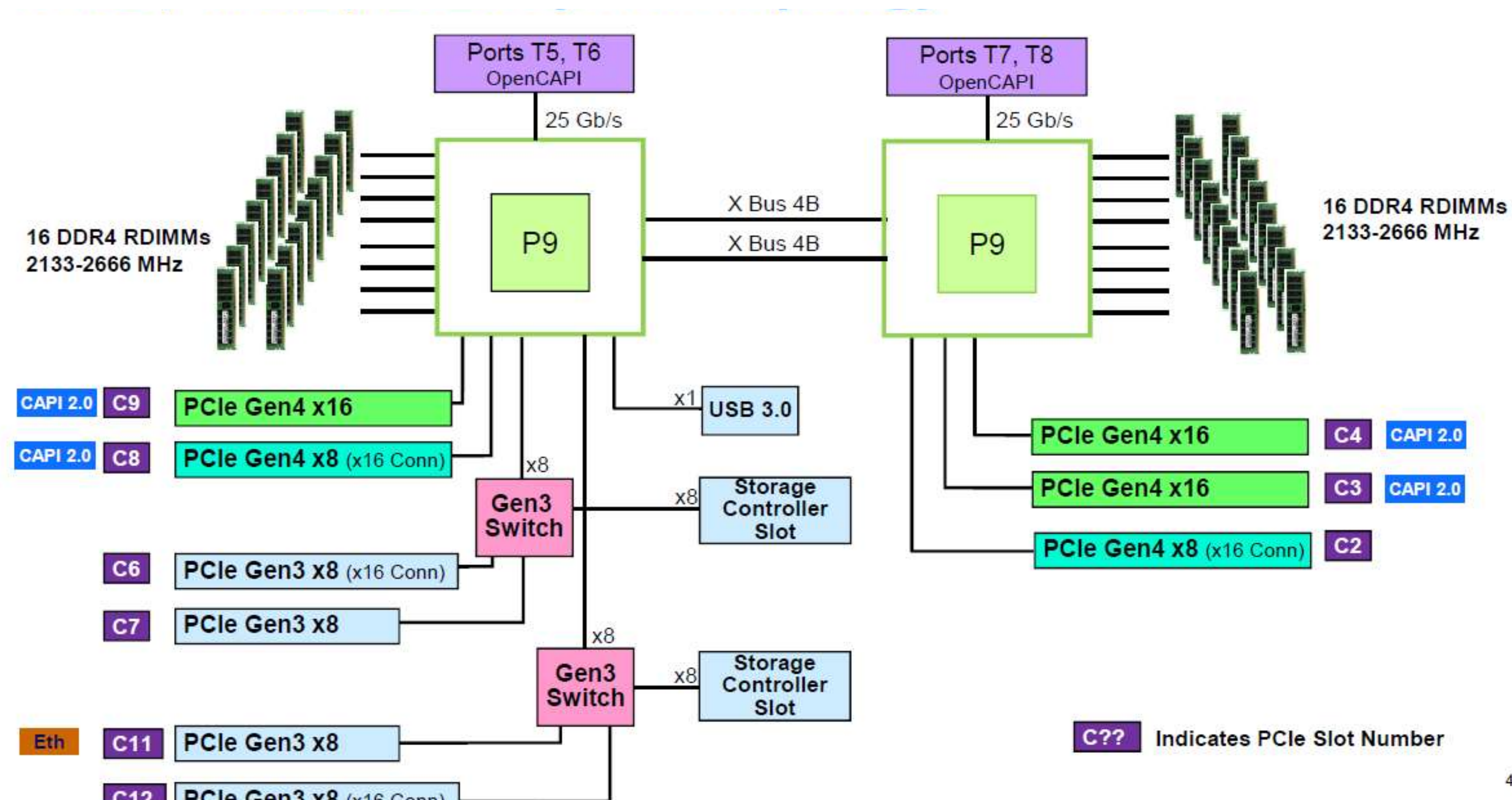
Example 1: The POWER9-based Scale-Out server family for enterprises [133]

L922 9008-22L	S922 9009-22A	S914 9009-41A	S924 9009-42A	H922 9223-22H	H924 9223-42H
• 1,2-socket, 2U • 8,10,12 cores/ socket • 32 IS RDIMM slots • 4TB memory • 4 CAPI 2.0 Slots • Linux only • PowerVM • KVM (GA2)	• 1,2-socket, 2U • 4, 8,10 cores/ socket • 32 IS RDIMM slots • 4TB memory • 4 CAPI 2.0 Slots • AIX, IBM i, & Linux • PowerVM	• 1-socket, 4U & Tower • 4,6,8 cores/ socket • 16 IS RDIMM slots • 1TB memory • 2 CAPI 2.0 Slots • Internal RDX Media • AIX, IBM i, Linux • PowerVM	• 2-socket, 4U • 8,10,12 cores/ socket • 32 IS RDIMM slots • 4TB memory • 4 CAPI 2.0 slots • Internal RDX Media • AIX, IBM i, Linux • PowerVM	• 1,2-socket, 2U • 4, 8,10 cores/ socket • 32 IS RDIMM slots • 4TB memory • 4 CAPI 2.0 Slots • AIX, IBM i up to 25% • Linux • PowerVM	• 2-socket, 4U • 8,10,12 cores/ socket • 32 IS RDIMM slots • 4TB memory • 4 CAPI 2.0 slots • Internal RDX Media • AIX, IBM i up to 25% • Linux • PowerVM

Technology Leadership

- Cloud enabled - Embedded virtualization capabilities with PowerVM
- Up to 4TB in 2 socket - DDR4 Industry Standard memory RDIMMs
- High Speed 25Gb/s external ports – one per socket
- 2 Internal NVMe Flash boot adapters
- Embedded Analytics and Algorithms on the chip help run POWER9 at an always optimized frequency

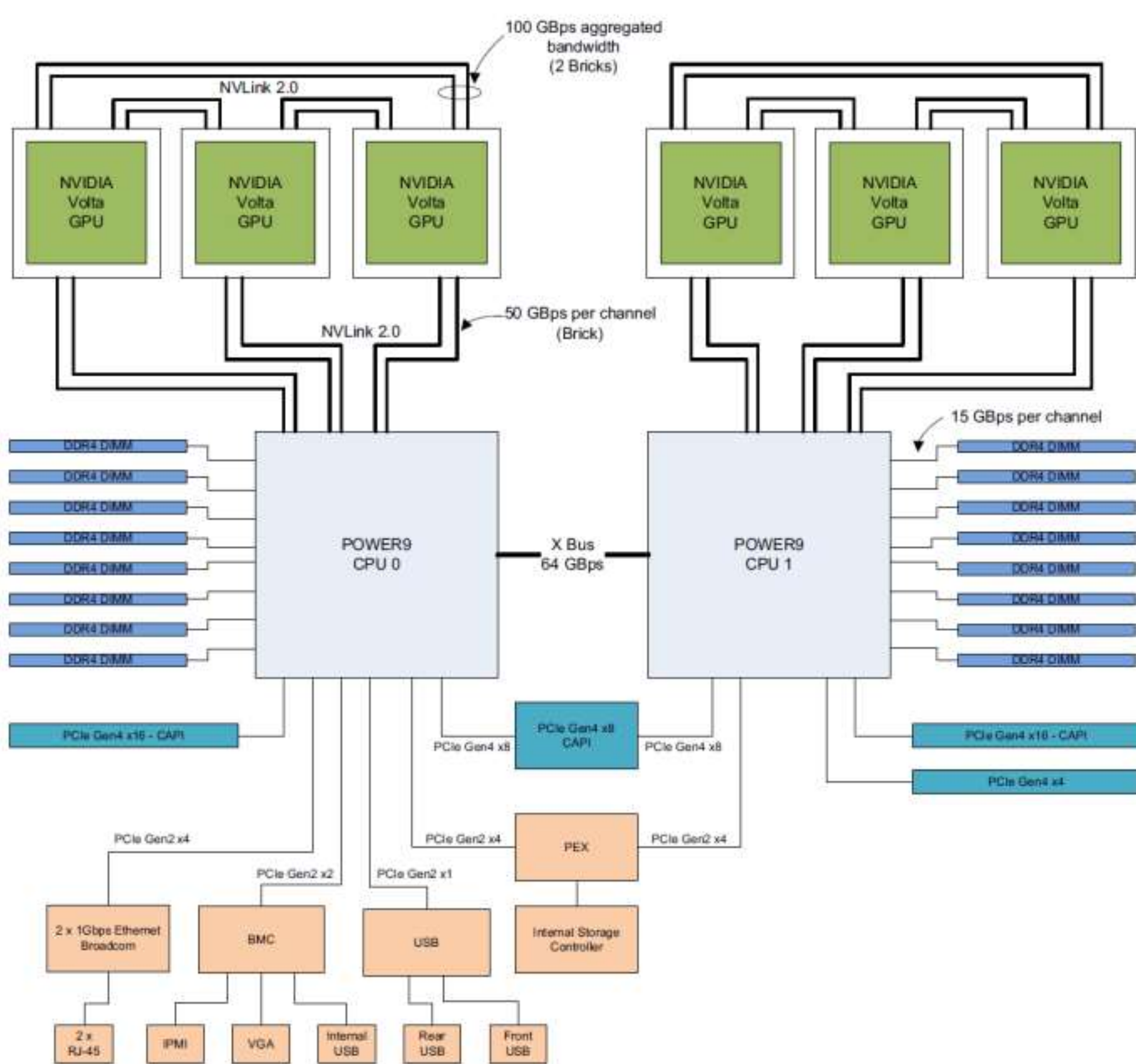
Block diagram of POWER9-based Scale-Out 2S servers for enterprises (S922/H922/L922) [133]



Example 2:
The POWER9-based 2S Accelerated server (AC922 Model 8335-GTG) [151]

It targets HPC or AI workloads.

2x PCIe Gen4 x8 – CAPI
2x PCIe Gen4 x16 - CAPI

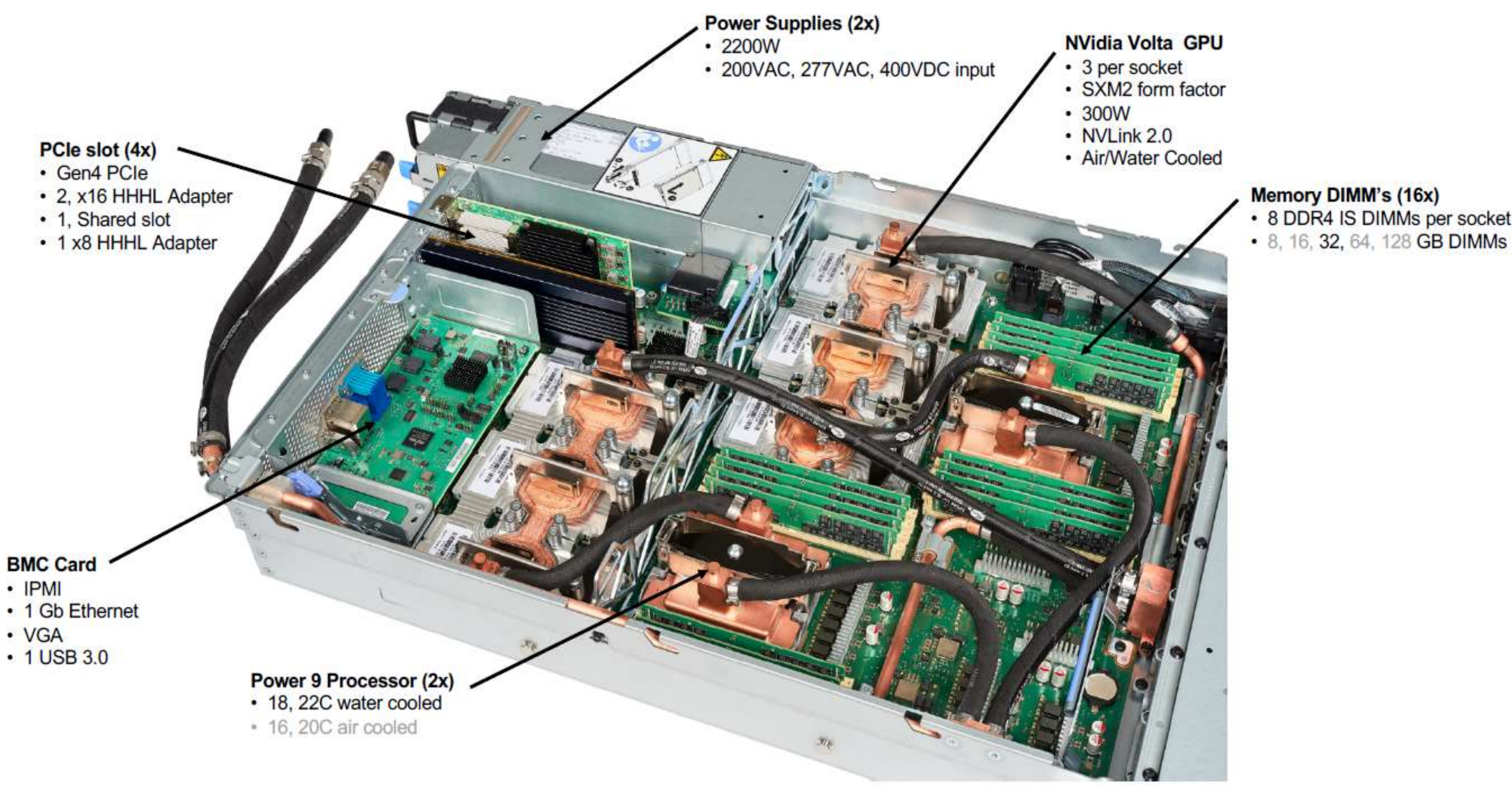


Layout of the POWER9-based AC922 server [151]

System planar with four GPUs

System planar with six GPUs

Picture of a POWER)-based AC922 server with 6 GPUs [162]



Remark to AC922 based supercomputers [162]

- A cooperation agreement was signed in 2014 between the three US national laboratories; Oak Ridge, Argonne and Livermore, called CORAL, to build state-of-the-art high performance supercomputers.
- In the framework of CORAL two procurements were issued in 2014,
  a) one to build the Summit supercomputer for the Oak Ridge National Laboratory (ORNL) that provides at least 5-times the performance of their Titan system, and
  b) another to build the Sierra supercomputer for the Livermore National Laboratory (LLNL) that is at least seven times more powerful then LLNL's current Sequoia machine.
- The contracts were won by IBM (POWER9 CPUs), NVIDIA (Volta GPUs) and Mellanox (Infiniband interconnection network).

  (IBM, NVIDIA and Mellanox are members of the OpenPOWER foundation, set up in 8/2013).
- Both supercomputers became operational in 2018,
  - Summit with a peak performance of 200 PFLOPS and
  - Sierra with a peak performance of 125 PFLOPS.

procurement: beszerzés

Summit: The world's fastest supercomputer at Oak Ridge NL (2018) [161]

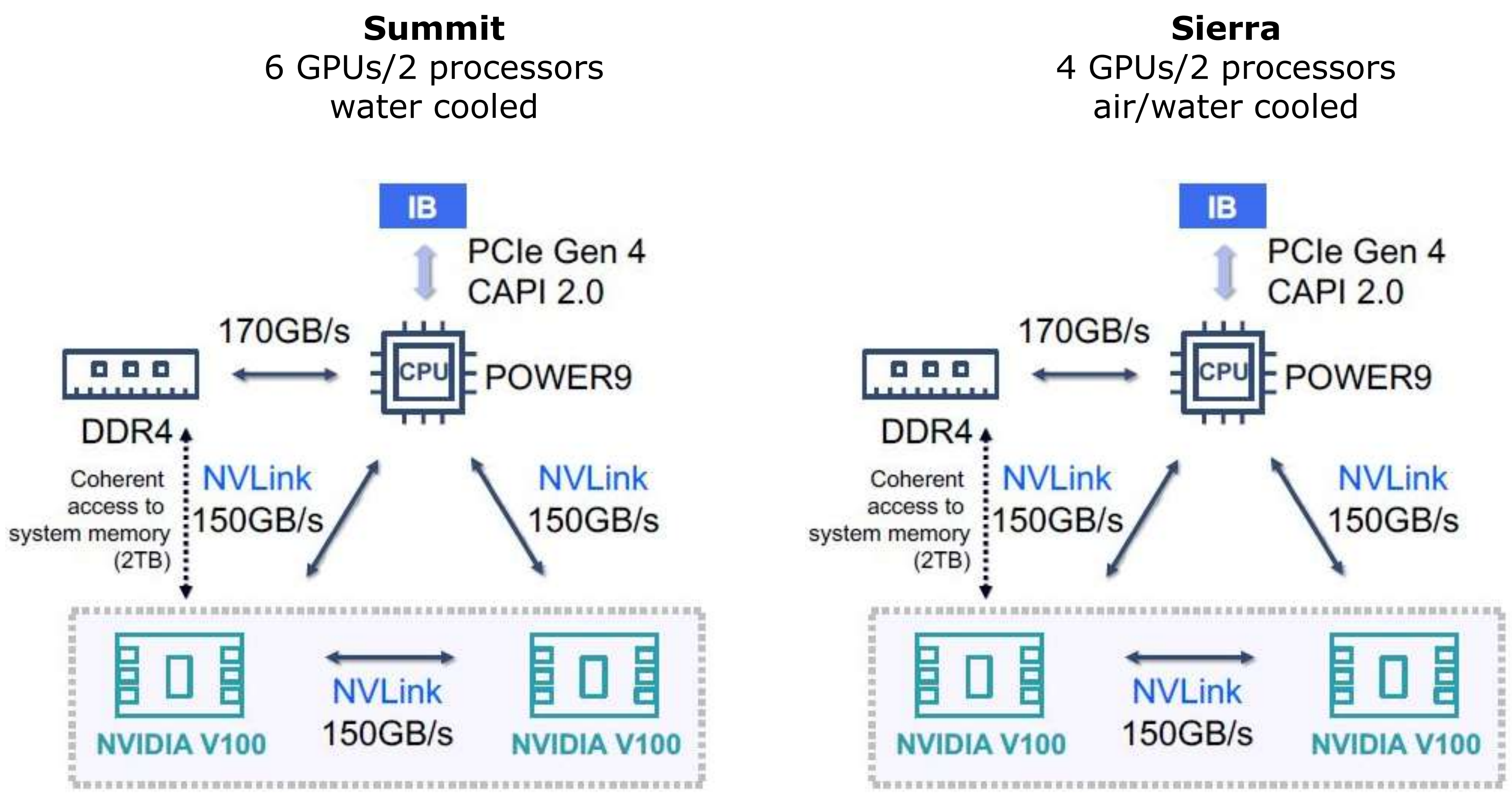
- **Summit** is made up of:
  - **4,608 compute nodes**
- Each node is made up of:
  - **Two 22-core IBM POWER9 CPUs** (AC922)
  - **6 NVIDIA Tesla V100 accelerators**
- Total Compute:
  - **9,216 IBM POWER9 CPUs**
  - **202,752 POWER9 cores**
  - **27,648 NVIDIA Volta GPUs**
  - **10 petabytes of Memory**
  - **250 petabytes of Storage**

12.6 POWER9-based servers (11)

The Sierra supercomputer at Lawrence Livermore National Laboratory (2018) [163],[164]

	Sierra
Nodes	4,320
POWER9 processors per node	2
GV100 (Volta) GPUs per node	4
Node Peak (TFLOP/s)	29.1
System Peak (PFLOP/s)	125
Node Memory (GiB)	320
System Memory (PiB)	1.29
Interconnect	2x IB EDR
Off-Node Aggregate b/w (GB/s)	45.5
Compute racks	240
Network and Infrastructure racks	13
Storage Racks	24

Contrasting the basic building blocks of Summit and Sierra [165]



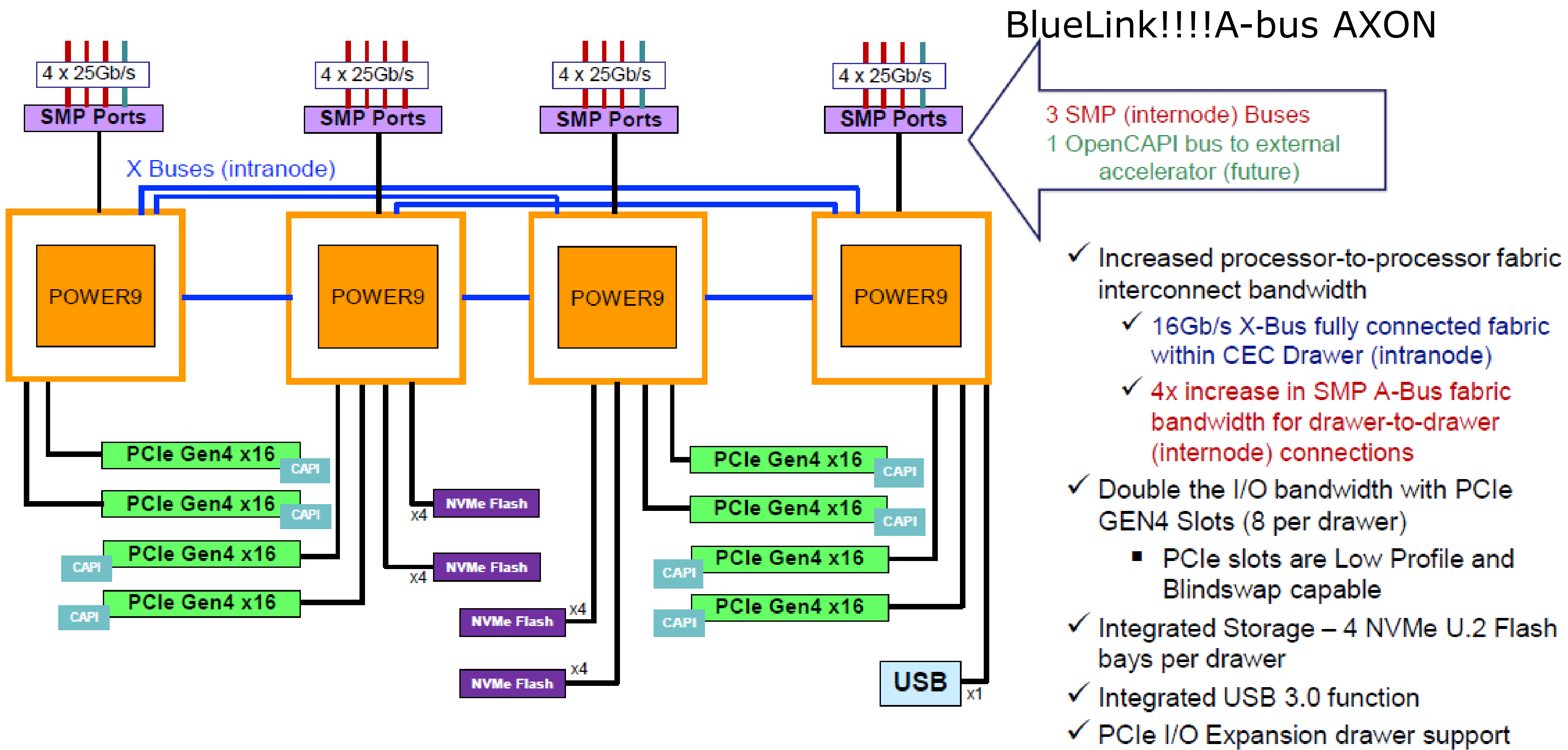
AC922, IBM POWER9 22C 3.07GHz , NVIDIA Volta GV100
Coherent access to system memory
PCI Gen.4 and CAPI 2.0 to Infiniband (IB)/Ethernet

Example 3: POWER9-based Scale-Up servers [166]

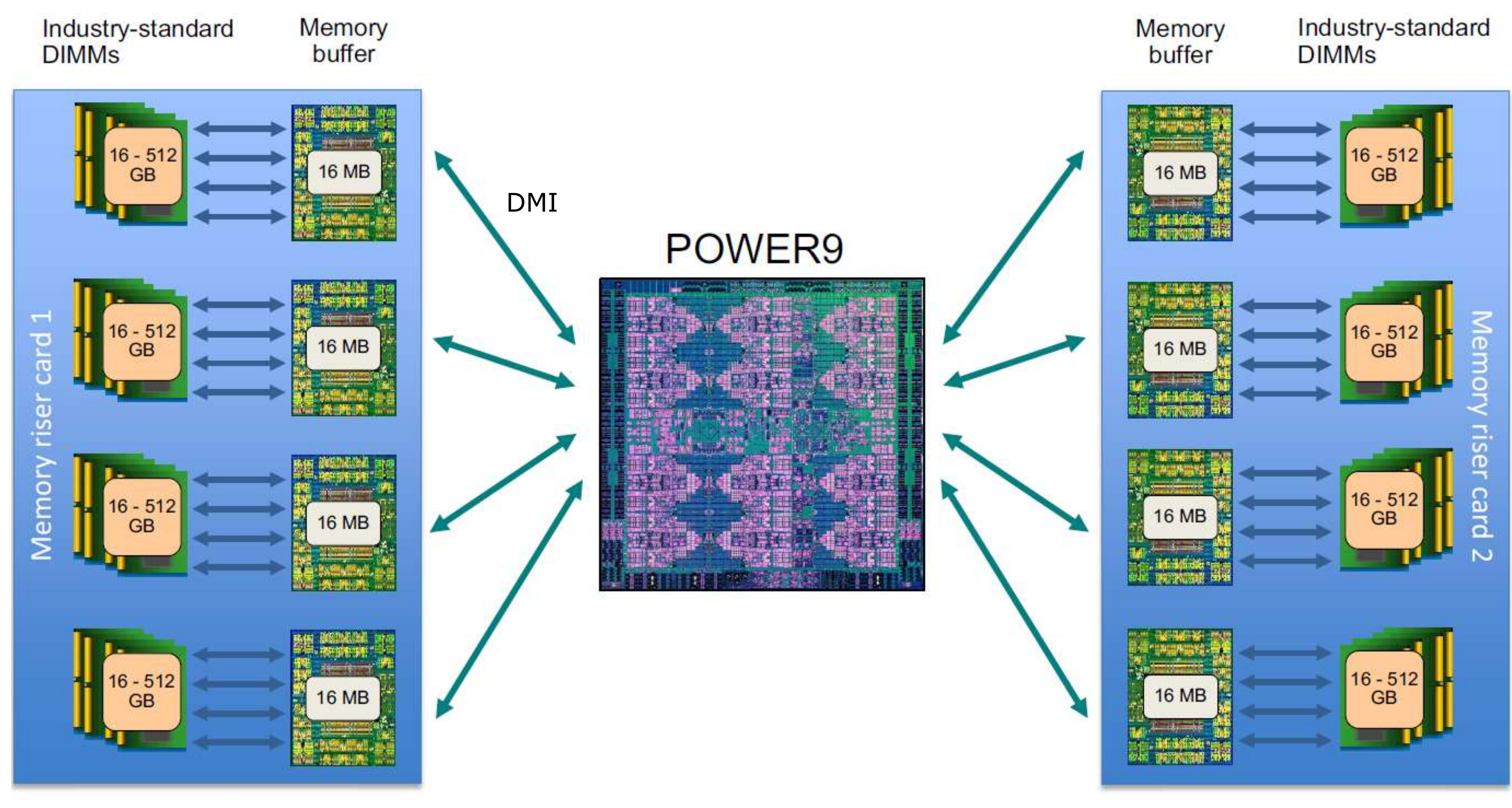
Designed for data intensive workloads (e.g. in-memory computing)

Feature	E950	E980 1-4 nodes
MTM	9040-MR9	9080-M9S
System Packaging	4U	5U system node & 2U system controller unit
Processor Socket	2S to 4S	4S per node
# of cores	32, 40, 44, or 48 cores	Up to 192 cores
Memory DIMM Slots -Max	128 DDR4 Industry Standard DIMMs	Up to 128 DDR4 CDIMMs
Memory – Max	16TB	64TB
Built-in virtualization	Yes	Yes
PCIe Gen4 Slots	10 slots	8 slots per node, up to 32 slots
Operating System	AIX, Linux	AIX, IBM i, Linux

Block diagram of a POWER9-based large scale Scale-Up server node (E980) (up to 4 nodes) [161]

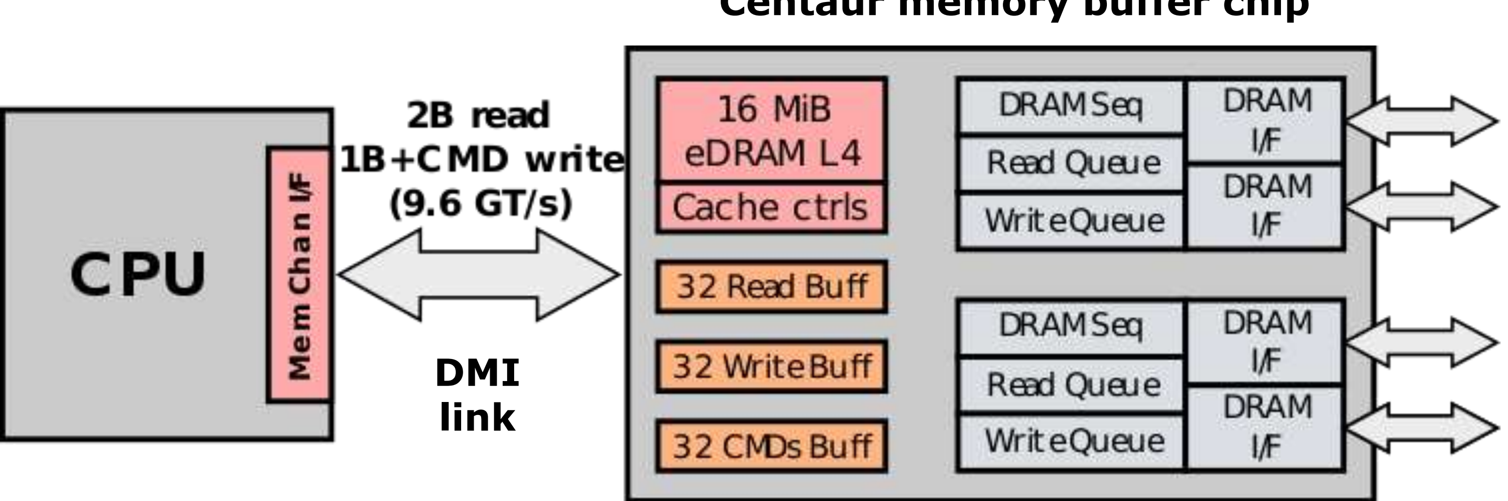


Buffered memory channels of a POWER 9-based Scale-Up server [167]



DMI: Differential Memory Interface

Block diagram of the Centaur buffer chip and its link to the CPU [168]



- DMI links have a max. signaling rate of 9.6 Gbit/s.
- Each link's read width is 2B and write width is 1B, this yields a bandwidth of 28.8 GB/s per link.
- Accordingly, 8 DMI links provide a total bandwidth of 230.4 GB/s.
- Each Centaur buffer chip services 4 DDR4 ports and has 16 MB L4 cache.
- One port connects to an industry standard DDR4 DIMM slot that is populated with a single industry standard DDR4-3200 DIMM.
- All in all, each processor has 32 DDR4 memory channels with a total bandwidth of 230,4 GB/s and 128 MB L4 buffer in the Centaur buffer chips.

13. References

13. References (1)

[1]: Starke W.J., Micro-Architecture, Oct. 12 2012,
http://regions.cmg.org/regions/ctcmg/Starke_CMG_charts.pdf

[2]: Hardware Overview, SCICOMP, IBM, July 2006, http://www.spscicomp.org/ScicomP12/Presentations/IBM/Tutorial_2.Hardware_AIX_Overview.pdf

[3]: Armstrong J., Sparks M., POWER8 Hardware Technical, April 14 2014

[4]: IBM Power Systems, Febr. 5 2013 Announcement, Hardware Deep Dive,
http://www-05.ibm.com/cz/events/febannouncement2012/pdf/power_architecture.pdf

[5]: McInnes J., POWER6 Processor and Systems, 2007

[6]: Tendler J.M., Dodson J.S., Fields Jr. J.S., Le H., Sinharoy B., POWER4 system microarchitecture
IBM J. Res. & Dev., Vol. 46, No. 1, Jan. 2002, http://www.ece.cmu.edu/~ece447/s14/lib/exe/fetch.php?media=tendler_et_al._-_2002_-_power4_system_microarchitecture.pdf

[7]: Anselmi G., Linzmeier G., IBM eServer p5 550 Technical Overview and Introduction, Oct. 2004
http://www.redbooks.ibm.com/redpapers/pdfs/redp9113.pdf

[8]: Vetter S., Cappelletti F., IBM eServer pSeries Systems Handbook, 2003 Edition, Dec. 2003,
http://www.redbooks.ibm.com/redbooks/pdfs/sg245120.pdf

[9]: Song P., IBM's Power3 to Replace P2SC, Power3 to Complete IBM's Transition to PowerPC Architecture, Microdesign Resources, Nov. 17 1997,
http://docencia.ac.upc.edu/ETSETB/SEGPAR/microprocessors/power3%20%28mpr%29.pdf

[10]: Haug V., Indest J., Vetter S., RS/6000 7044 Model 170 Technical Overview and Introduction, Febr. 24 2000, http://unixhq.com/websgt/7044-170_overview.pdf

[11]: Dreps D.M., Ferraiolo F.D., Gower K.C., Rippens R.A., 276-Pin buffered memory module with enhanced fault tolerance and a performance-optimized pin assignment, US 7529112 B2, May 5 2009, https://www.google.com.ar/patents/US7529112

[12]: Tendler J.M., Dodson S., Fields S., Le H., Sinharoy B., POWER4 System Microarchitecture, IBM Technical White Paper, Oct. 2001, ftp://ftp.software.ibm.com/software/mktsupport/techdocs/power4.pdf

[13]: Anselmi G. Lutz S., Okano M., IBM eServer pSeries 650 Model 6M2 Technical Overview and Introduction, May 2003, http://www.redbooks.ibm.com/redpapers/pdfs/redp0194.pdf

[14]: BM 4452 2048 MB (4X512MB) DIMMs 208-pin, 8NS DDR SDRAM, http://www.ebay.it/itm/IBM-4452-2048-MB-4X512MB-DIMMs-208-pin-8NS-DDR-SDRAM-/400853588766

[15]: Matsubara K., Hazelzet B., Kahle M-E., IBM eServer pSeries 670 and pSeries 690 System Handbook, Oct. 2002, http://www.hpcx.ac.uk/support/documentation/IBMdocuments/sg247040.pdf

[16]: Anselmi G., Linzmeier G., Vetter S., IBM eServer pSeries 615 Models 6C3 and 6E3 Technical Overview and Introduction, Oct. 2003, http://www.redbooks.ibm.com/redpapers/pdfs/redp0160.pdf

[17]: Hoskins J., Bluethman R., Exploring IBM eServer pSeries, 2004, https://books.google.hu/books?id=MLS6Ty8Vo80C&pg=PA94&dq=power4+i/o+gx&hl=hu&sa=X&ei=vN0iVdeQMcG7swGprIDYBQ&ved=0CCEQ6AEwAA#v=onepage&q&f=false

[18]: Holthoff H., A Hitchhiker's Guide to the IBM RS/6000 SP, Jan. 2000, http://wwwuser.gwdg.de/~applsw/Parallelrechner/sp_documentation/talks/ibm_sp_environment_jan2000.pdf

[19]: Behling S., Bell R., Farrell P., The POWER4 Processor Introduction and Tuning Guide, Nov. 2001, http://www.redbooks.ibm.com/redbooks/pdfs/sg247041.pdf

[20]: Barney B., „IBM POWER Systems Overview", Livermore Computing, 2006, http://www.llnl.gov/computing/tutorials/ibm_sp/

[21]: Grassl C., POWER5 Processor and System Evolution, May 2005, http://www.spscicomp.org/ScicomP11/Presentations/IBM/tutorial-power5.pdf

[22]: Kalla R., Sinharoy B., Tendler J., Simultaneous Multi-threading Implementation in Power5 – IBM's Next Generation POWER Microprocessor, 2003, http://www.hotchips.org/wp-content/uploads/hc_archives/hc15/3_Tue/11.ibm.pdf

[23]: Sinharoy B., Kalla R.N., Tendler J.M., Eickenmeyer R.J., Joyner J.B., POWER5 system microarchitecture, IBM J. R&D, Vol. 49, No. 4/5, 2005

[24]: Domberg P., Kelley N., Kim T., Wei D., IBM eServer p5 590 and 595 System Handbook, March 2005, http://www.redbooks.ibm.com/redbooks/pdfs/sg249119.pdf

[25]: Kalla R., IBM's POWER5 Microprocessor Design and Methodology, 2003, www-csl.csres.utexas.edu/users/billmark/teach/cs352-05-spring/lectures/Lecture22-RonKallaIBM.pdf

13. References (4)

[26]: Krewell K., POWER5 Tops On Bandwidth, Microprocessor Report, Dec. 22 2003, http://docencia.ac.upc.edu/ETSETB/SEGPAR/microprocessors/power5%20%282%29%20%28mpr%29.pdf

[27]: Anselmi G., Linzmeier G., Seiwald W., Vandamme P., IBM eServer p5 520 Technical Overview and Introduction, Oct. 2004, http://www.redbooks.ibm.com/redpapers/pdfs/redp9111.pdf

[28]: Constantini C., Cler C., Wood J., IBM System p5 590 and 595 Technical Overview and, Introduction, Sept. 2006, http://www.redbooks.ibm.com/redpapers/pdfs/redp4024.pdf

[29]: Reick K., Sanda P.N., Swaney S., Kellington J.W., Floyd M., Henderson D., Fault - Tolerant Design of the IBM POWER6 Microprocessor, Hot Chips 19, 2007, http://www.hotchips.org/wp-content/uploads/hc_archives/hc19/2_Mon/HC19.01/HC19.01.01.pdf

[30]: Suchy J., IBM System p Enterprise Technical Excellence, IBM Forum, 2008

[31]: Le H.Q., Starke W.J., Fields J.S., O'Connell F.P., IBM POWER6 microarchitecture, IBM J. R&D, Vol. 51, Issue 6, Nov. 2007

[32]: Anselmi G., Cho Y., Cook J., Linzmeier G., IBM Power 550 Technical Overview, May 2009, http://www.redbooks.ibm.com/redpapers/pdfs/redp4404.pdf

[33]: Henderson D., Warner B., Mitchell J., IBM POWER6 Processor-based Systems: Designed for Availability, June 11 2007, http://www.nasi.com/docs/pdfs/POWER6_availability.pdf

13. References (5)

[34]: Anselmi G., Cho Y., Linzmeier G., Quezada M., IBM Power 570 Technical Overview and Introduction, Oct. 2008, http://www.redbooks.ibm.com/redpieces/pdfs/redp4405.pdf

[35]: Haas J., Vogt P., Fully buffered DIMM Technology Moves Enterprise Platforms to the Next Level, Technology Intel Magazine, March 2005

[36]: Wikipedia, Fully Buffered DIMM, https://en.wikipedia.org/wiki/Fully_Buffered_DIMM

[37]: Cler C., Costantini C., IBM Power 595 Technical Overview and Introduction, Aug. 2008, http://www.redbooks.ibm.com/redpapers/pdfs/redp4440.pdf

[38]: Diefendorff K., Dubey P.K., Hochsprung R., Scales H., AltiVec Extension to PowerPC Accelerates Media Processing, IEEE Micro, March-April 2000, http://www.eecg.utoronto.ca/~moshovos/ACA06/readings/altivec.pdf

[39]: Larin S., H1119 – Introduction to AltiVec – Ten easy ways to Vectorize your code, Smart Networks Developer Forum, Apr. 26-29 2004, http://www.freescale.com/files/32bit/doc/reports_presentations/LNXDEVSYS_PPT1.pdf

[40]: Bovy C., AMD-K7 Processor Athlon, Dec. 24 1999, http://www.powershow.com/view1/265aca-ZDc1Z/AMDK7_PROCESSOR_Athlon_powerpoint_ppt_presentation

[41]: Diefendorff K., Katmai Enhances MMX, Microprocessor Report, Vol. 12, No. 13, Oct. 5 1998, http://studies.ac.upc.edu/ETSETB/SEGPAR/microprocessors/katmai%20%28mpr%29.pdf

[42]: Goto H., Larrrabee architecture can be integrated into CPU, PC Watch, Oct. 6 2008, http://pc.watch.impress.co.jp/docs/2008/1006/kaigai470.htm

[43]: Eisen L., Ward J.W., Tast H.-W., Mading N., Leenstra J. et al., IBM POWER6 accelerators: VMX and DFU, IBM J. R&D, Vol. 51, No. 6, Nov. 2007, http://www.christianjacobi.de/publications/ewt07_p6vmx.pdf

[44]: IEEE Standard for Binary Floating-Point Arithmetic, 1985, http://homepages.math.uic.edu/~jan/mcs471/Lec3/ieee754.pdf

[45]: Power ISA, Version 2.06 Revision B, July 23 2010, https://www.power.org/wp-content/uploads/2012/07/PowerISA_V2.06B_V2_PUBLIC.pdf

[46]: Dongarra J.J., van der Steen A.J., High-performance computing systems: Status and outlook, Cambridge University Press, 2012 http://www.netlib.org/utk/people/JackDongarra/PAPERS/acta-num-2012.pdf

[47]: Power Firmware - The Center of the Universe, Jan. 30 2013, https://www.ibm.com/developerworks/community/blogs/PowerFW/entry/power_firmware_the_center_of_the_universe57?maxresults=1&page=4&lang=en

[48]: Floyd M.S., System power management support in the IBM POWER6 microprocessor, IBM J. R&D, Vol. 51, No. 6, 2007

[49]: McCreary H.-Y., Broyles M.A., Floyd M.S., Geissler, A.J., EnergyScale for IBM POWER6 microprocessor-based systems, IBM J. R&D, Vol. 51, Issue 6, Nov. 2007

13. References (7)

[50]: Lefurgy C. R., Drake A.J., Floyd M.S., Allen-Ware M.S. et al., Active Management of Timing Guardband to Save Energy in POWER7, Micro 44, Dec. 3-7 2011, , http://researcher.watson.ibm.com/researcher/files/us-lefurgy/micro44_Charles_Lefurgy.pdf

[51]: Floyd M., Allen-Ware M., Rajamani K., Brock B., Introducing the Adaptive Energy Managemen Features of the Power7 Chip, IEEE Micro, Vol. 31, Issue 2, March-April 2011

[52]: Drake A.J., Senger R.M., Singh H., Carpenter G.D., James N.K., Dynamic Measurement of Critical-Path Timing, ICICDT 2008

[53]: Drake A., Senger r. Deogun H., Carpenter G., A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor, ISSCC 2007

[54]: Ernst D., Kim N.S., Das S., Pant S. et al., Razor: a low-power pipeline based on circuit-level timing speculation, Micro 36 2003

[55]: Elgebaly M., Sachdev M., Efficient Adaptive Voltage Scaling System Through On-Chip Critical Path Emulation, ISLPED'04, Aug. 9–11 2004, https://ece.uwaterloo.ca/~cdr/pubs/elgebaly_islped04

[56]: IBM Power Systems Family Quick Reference Guide, Dec. 2009, http://www.midlandinfosys.com/pdf/ibm-i-power6-model-cpw-systemi-performance-comparison.pdf

[57]: Armstrong J., IBM Power Systems, Febr. 5 2013 Announcement

13. References (8)

[58]: Kennewell P., Growth Powered by IBM and Oracle, Aug. 16 2011
http://www.slideshare.net/InSync2011/ebusiness-suite-1-peter-kennewell-ebs-growth-powered-by-ibm-and-oraclepdf

[59]: Sinharoy B., Kalla R., Starke W.J., Le H.Q., IBM POWER7 multicore server processor, IBM J. R&D, Vol. 55, Issue 3, May-June 2011

[60]: Sadasivam S.K., Kumar P., Mallikarjunan V., Power.org White Paper - What's new in the Server Environment of Power ISA v2.06?, Version 1.0, Jan. 18 2010,
https://www.power.org/wp-content/uploads/2012/06/Power.org_White_Paper_What_is_New_in_Server_Environment_of_Power_ISA_v2.06.pdf

[61]: Starke W.J., POWER7: IBM's Next Generation, Balanced POWER Server Chip, Hot Chips 21,
http://www.hotchips.org/wp-content/uploads/hc_archives/hc21/3_tues/HC21.25.800.ServerSystemsII-Epub/HC21.25.835.Starke-IBM-POWER7SystemBalancev13_display.pdf

[62]: Quintero D., Bosworth K., Chaudhary P. et al., IBM Power Systems 775 for AIX and Linux HPC Solution, Oct. 2012, http://www.redbooks.ibm.com/redbooks/pdfs/sg248003.pdf

[63]: Loughner K.D., Gower K.C., Kilmer C.A., Maule W.E., 276-pin buffered memory module with enhanced memory system interconnect and features, US 20100005220 A1, Jan. 7 2010, http://www.google.com/patents/US20100005220

[64]: Chen A.D., Cruickshank J., Costantini C., IBM Power 720 and 740 Technical Overview and Introduction, Nov. 2010, http://www.redbooks.ibm.com/redpapers/pdfs/redp4637.pdf

13. References (9)

[65]: Ware M., Rajamani K., Floyd M., Brock B., Architecting for Power Management: The IBM POWER7 Approach, IEEE HPCA, 2010

[66]: Floyd M., Adaptive Energy Management Features of the POWER7 Processor, Hot Chips 22, Aug. 23 2010, http://researcher.watson.ibm.com/researcher/files/us-lefurgy/hotchips22_power7.pdf

[67]: Wendel D., Kalla R., Cargoni R., Clables J., The Implementation of POWER7: A Highly Parallel and Scalable Multi-Core High-End Server Processor, IEEE ISSCC, 2010

[68]: Saykov M., Innovations in action with Power7+, 2012, http://www-05.ibm.com/bg/ibmforum/attachment/Mikhail_Saykov_Innovation%20in%20action%20with%20Power%207+.pdf

[69]: Haug V., Next generation of Power, April 29 2014, http://ibm.also.ch/fileadmin/Dateien/pdf/power8/POWER8_-_Next_Generation_of_Power_28.04.2014_Deutsch.pdf

[70]: POWER7+, IBM, Aug. 29 2012, http://www.hotchips.org/wp-content/uploads/hc_archives/hc24/HC24-8-DataCenter/HC24.29.815-Power7-Taylor-IBM-120828-Final.pdf

[71]: Cruickshank J., Hanganu S., Haug V., IBM Power 710 and 730 Technical Overview and Introduction, May 2013, http://www.redbooks.ibm.com/redpapers/pdfs/redp4983.pdf

[72]: Chen A.D., Freeman D., Leitao B.H., IBM Power 770 and 780 (9117-MMD, 9179-MHD) Technical Overview and Introduction, Febr. 2013, http://www.redbooks.ibm.com/redpapers/pdfs/redp4924.pdf

13. References (10)

[73]: Cruickshank J., Hanganu S., Haug V., IBM Power 750 and 760 Technical Overview and Introduction, May 2013, http://www.redbooks.ibm.com/redpapers/pdfs/redp4985.pdf

[74]: What Is SSL (Secure Sockets Layer) and What Are SSL Certificates?, DigiCert, https://www.digicert.com/ssl.htm

[75]: Wikipedia, SHA-1, https://en.wikipedia.org/wiki/SHA-1

[76]: Wikipedia, RSA (cryptosystem), https://en.wikipedia.org/wiki/RSA_%28cryptosystem%29

[77]: Christensen M.Z., POWER Processors Overview & Directions, 2013

[78]: Gao Y., HPC Workload Performance Tuning on POWER8 with IBM XL Compilers and Libraries, SPXXL/Scicomp Summer Workshop 2014, http://spscicomp.org/wordpress/wp-content/uploads/2014/05/gao-IBM-XL-compilers-for-POWER8-Scicomp-2014.pdf

[79]: Corrado M., Google, IBM, Mellanox, NVIDIA, Tyan Announce Development Group for Data Centers, http://www-03.ibm.com/press/us/en/pressrelease/41684.wss

[80]: Morgan T.P., IBM opens up Power chips, ARM-style, to take on Chipzilla, The Register, Aug. 6 2013, http://www.theregister.co.uk/2013/08/06/ibm_opens_up_power_chips_armstyle_to_take_on_chipzilla/

[81]: Shannon D., Visual Computing's Ascent Gives NVIDIA Room to Expand Its Business Model, Nvidia, June 18 2013, http://blogs.nvidia.com/blog/2013/06/18/visual-computings-ascent-gives-nvidia-room-to-expand-its-business-model/

[82]: Edwards C., Intel tips 14nm processor, Quark core and SoC licensing plans, Tech Design Forum, Sept. 11 2013, http://www.techdesignforums.com/blog/2013/09/11/intel-14nm-quark-soc/

[83]: Stuecheli J., POWER8, Hot Chips, 2013, http://www.hotchips.org/wp-content/uploads/hc_archives/hc25/HC25.20-Processors1-epub/HC25.26.210-POWER-Studecheli-IBM.pdf

[84]: Sinharoy B., van Norstrand J.A., Eickemeyer R.J., Le H.Q., IBM POWER8 processor core microarchitecture, IBM J. R&D, Vol. 59, Issue 1, Jan.-Febr. 2015

[85]: Liu W., Chen G., Wang Y., Yang H., Modeling and optimization of low power resonant clock mesh, ASP-DAC, 2015

[86]: AMD FX-8350: Vishera, a lánctalpas cölöpverő, Prohardver, Oct. 23 2010, http://prohardver.hu/teszt/amd_fx-8350_vishera_piledriver_teszt/piledriver_v2_bulldozer_kipofozva.html

[87]: Payne D., Clock Design for SOCs with Lower Power and Better Specs, SemiWiki, Dec. 15 2011, http://www.semiwiki.com/forum/content/917-clock-design-socs-lower-power-better-specs.html

[88]: Clock Distribution, Acsel-lab.com, July 28 2004, http://www.acsel-lab.com/Projects/clocking/clock_distribution.htm

[89]: Restle P.J., McNamara T.G., Webber D.A. et al., A Clock Distribution Network for Microprocessors, IEEE Journal of Solid-State Circuits, Vol. 36, No. 5, May 2001, http://weble.upc.es/ifsin/Block5/00918917.pdf

[90]: Wolfe J., LC Oscillations and Resonance, University of New South Wales, http://www.animations.physics.unsw.edu.au/jw/LCresonance.html

[91]: Chan S.C., Restle P.J., Bucelot T.J., Liberty J.S., A Resonant Global Clock Distribution for the Cell Broadband Engine Processor, IEEE Journal of Solid-State Circuits, Vol. 44, Issue 1, Jan. 2009

[92]: Chan S.C., Shepard K.L., Restle P.J., Uniform-Phase Uniform-Amplitude Resonant-Load Global Clock Distributions, IEEE Journal of Solid-State Circuits, Vol. 40, No. 1, Jan. 2005, http://bioee.ee.columbia.edu/downloads/clocking_3.pdf

[93]: Groves R. Restle P., Drake A., Shan D., Optimization and modeling of resonant clocking inductors for the POWER8 microprocessor, IEEE CICC, 2014

[94]: Zyuban V., Friedrich J. Dreps D.M., IBM POWER8 circuit design and energy optimization, IBM J. R&D, Vol. 59, Issue 1, Jan.-Febr. 2015

[95]: Fluhr E.J., Baumgartner S., Boerstler D., Bulzacchelli J.F., The 12-Core POWER8 Processor with 7.6 Tb/s IO Bandwidth, Integrated Voltage Regulation, and Resonant Clocking, IEEE Journal of Solid-State Circuits, Vol. 50, Issue 1, Jan. 2015

[96]: Restle P., Shan D., Hogenmiller D., 5.3 Wide-frequency-range resonant clock with on-the-fly mode changing for the POWER8 microprocessor, IEEE ISSCC, 2014

[97]: Puri R., Bridging High Performance and Low Power in the era of Big Data and Heterogeneous Computing, 2014, http://www.islped.org/2014/files/ISLPED%20invited%20talk%20final_Big_DATA_Heterogeneous.pdf

[98]: Morrison V., What Every Dev Must Know About Multithreaded Apps, MSDN Magazine, 2005, https://msdn.microsoft.com/en-us/magazine/cc163744.aspx

[99]: Knight T., An architecture for mostly functional languages, 1986, Proceedings of the 1986 ACM conference on LISP and functional programming

[100]: Herlihy M., Eliot J., Moss B., Transactional Memory: Architectural Support for Lock-Free Data Structures, 1993, Proceedings of the 20th Annual International Symposium on Computer Architecture

[101]: Shavit N., Touitou D., Software transactional memory, Distributed Computing, 1997, http://groups.csail.mit.edu/tds/papers/Shavit/ShavitTouitou.pdf

[102]: Bright P., IBM's new transactional memory: make-or-break time for multithreaded revolution, Ars Technica, Aug. 31 2011, http://arstechnica.com/gadgets/2011/08/ibms-new-transactional-memory-make-or-break-time-for-multithreaded-revolution/

[103]: What is transactional memory?, Stackoverflow, June 29 2012, http://stackoverflow.com/questions/11255640/what-is-transactional-memory

[104]: Transactional Memory: History and Development, May 15 2014, http://kukuruku.co/hub/cpp/transactional-memory-history-and-development

[105]: Performance Optimization and Tuning Techniques for IBM Processors, including IBM POWER8 IBM Redbooks, July 2014, http://www.redbooks.ibm.com/redbooks/pdfs/sg248171.pdf

[106]: Dice D., Lev Y., Moir M., Nussbaum D., Early Experience with a Commercial Hardware Transactional Memory Implementation, ASPLOS'09, March 7–11. 2009, http://vglab.cse.iitd.ac.in/~sbansal/csl862-os/readings/htm_experiences.pdf

[107]: Chaudhry S., Cypher R., Ekman M., Karlsson M., Landin A., Yip S., Zeffer H., Tremblay M., Rock: A High-Performance Sparc CMT Processor, IEEE Micro, Vol. 29, March/April 2009

[108]: Wang A., Gaudet M., Wu P., Ohmacht M., Software Support and Evaluation of Hardware Transactional Memory on Blue Gene/Q, IEEE Transactions on Computers, Vol. 64, 2013

[109]: Jacobi C., Slegel T., Greiner D., Transactional Memory Architecture and Implementation for IBM System z, 2012, IEEE/ACM 45th Annual International Symposium on Microarchitecture

[110]: Reinders J., Transactional Synchronization in Haswell, Intel Developer Zone, Febr. 7 2012, https://software.intel.com/en-us/blogs/2012/02/07/transactional-synchronization-in-hasw

[111]: Leis V., Kemper A., Neumann T., Exploiting Hardware Transactional Memory in Main-Memory Databases, http://www-db.in.tum.de/~leis/papers/HTM.pdf

[112]: Pirzada U., Intel TSX Bug Will Be Fixed By Broadwell-K – Impossible to Fix in Haswell, WCCF Tech, Oct. 22 2014, http://wccftech.com/intel-tsx-bug-fixed-broadwellk-impossible-fix-haswell-patch/#ixzz3UhaGAFHx

[113]: Le H.Q., Guthrie G.L., Williams D.E., Michael M.M., Transactional memory support in the IBM POWER8 processor, IBM Journal of Research and Development, Vol. 59, Issue 1, 2015

[114]: Starke W.J., Stuecheli J., Daly D.M., Dodson J.S., The cache and memory subsystems of the IBM POWER8 processor, IBM J. R&D, Vol. 59, Issue 1, Jan.-Febr. 2015

[115]: Stuecheli J., Power Technology For a Smarter Future, 2014

[116]: Caldeira A.B., Cho Y.H., Cruickshank J., IBM Power Systems E870 and E880 Technical Overview and Introduction, Dec. 2014, http://www.redbooks.ibm.com/redpapers/pdfs/redp5137.pdf

[117]: Caldeira A.B., Grabowski B., Haug V., IBM Power Systems S814 and S824 Technical Overview and Introduction, Aug. 2014, http://www.redbooks.ibm.com/redpapers/pdfs/redp5097.pdf

[118]: Wile B., Coherent Accelerator Processor Interface (CAPI) for POWER8 Systems, White Paper, Sept. 29 2014

[119]: Stuecheli J., Blaner B., Johns J.R., Siegel M.S., CAPI: A Coherent Accelerator Processor Interface, IBM J. R&D, Jan. 2015

[120]: Toprak-Deniz Z., Sperling M., Bulzacchelli J., Still G., Distributed system of digitally controlled microregulators enabling per-core DVFS for the POWER8TM microprocessor, IEEE ISSCC, 2014

[121]: Application Note: Atmel AT04204: Design a Buck Converter with XMEGA E, Atmel Corp., 2013, http://www.atmel.com/Images/Atmel-42183-Design-a-Buck-Converter-with-XMEGA-E_AP-Note_AT04204.pdf

[122]: Burton E.A. & al., FIVR – Fully Integrated Voltage Regulators on 4th Generation Intel Core™ SoCs, Proc. Twenty-Ninth Annual IEEE Applied Power Electronics Conference and Exposition (APEC), 2014

[123]: Ziegler M.M., Gristede G.D., Zyuban V.V., Power Reduction by Aggressive Synthesis Design Space Exploration, IEEE ISLPED, 2013

[124]: Chen A.D., Costantini C., Cruickshank J., IBM Power 795 (9119-FHB) Technical Overview and Introduction, Febr. 2013, http://www.redbooks.ibm.com/redpapers/pdfs/redp4640.pdf

[125]: Addressing the Power-Performance IC Design Conundrum, Cyclos Semiconductor, June 1 2012, http://www.cyclos-semi.com/pdfs/time_to_change_the_clocks.pdf

[126]: O'Mahony F., Yue C.P., Horowitz M.A., Wong S.S., A 10-GHz Global Clock Distribution Using Coupled Standing-Wave Oscillators, IEEE Journal of Solid-State Circuits, Vol. 38, No. 11, Nov. 2003, http://www.bioee.ee.columbia.edu/courses/ee6321/papers/omahony03.pdf

[127]: Drake A.J., Nowka K.J., Nguyen T.Y., Burns J.L., Brown R.B., Resonant clocking using distributed parasitic capacitance, IEEE Journal of Solid-State Circuits, Vol. 39, Issue 9, Sept. 2004

[128]: Anselmi G., Blanchard B., Cho Y., IBM Power 770 and 780 Technical Overview and Introduction, March 2010, http://www.redbooks.ibm.com/redpapers/pdfs/redp4639.pdf

[129]: Phillip M., A Second Generation SIMD Microprocessor Architecture, Aug. 17 1998, http://www.hotchips.org/wp-content/uploads/hc_archives/hc10/2_Mon/HC10.S5/HC10.5.3.pdf

[130]: Anthony S., IBM unveils Power8 and OpenPower pincer attack on Intel's x86 server monopoly, Extreme Tech, Apr. 23 2014, http://www.extremetech.com/computing/181102-ibm-power8-openpower-x86-server-monopoly

13. References (17)

[131]: McCredie B., OpenPOWER and the Roadmap Ahead, OpenPOWER Summit 2016, April 5-8, http://openpowerfoundation.org/wp-content/uploads/2016/04/5_Brad-McCredie.IBM_.pdf

[132]: Hurlimann D., POWER8 hardware, IBM, June 2014, http://docplayer.net/48005572-June-power8-hardware-dan-hurlimann-system-p-hardware-architect-2014-ibm-corporation.html

[133]: Bizon J., Power Systems POWER9 Scale Out Servers, IBM, 2018, http://dmctechgroup.com/wp-content/uploads/2018/03/POWER9-Announcements-John-Bizon-20180306.pdf

[134]: Power Architecture, WikiVisually, https://wikivisually.com/wiki/Power_Architecture

[135]: Power.org launches Power ISA Version 2.03, EETimes, June 6 2011, https://www.eetimes.com/document.asp?doc_id=1302231#

[136]: US 2006/0206657 A1, Clark et al. (43) Pub. Date: Sept. 14 2006

[137]: Wright C., Henning P., Bergen B., Roadrunner Tutorial, An Introduction to Roadrunner, and the Cell Processor, Febr. 7 2008, http://www.lanl.gov/orgs/hpc/roadrunner/pdfs/Roadrunner-tutorial-session-1-web1.pdf

[138]: US 7,917,730 B2 patent, Marino et al., March 29 2011

[139]: POWER7+™, IBM, Systems & Technology Group, Aug. 29 2012, https://pdfs.semanticscholar.org/c442/dce5b57c69d9911c029fa9929d252ddfab6e.pdf

13. References (18)

[140]: Stuecheli J., POWER8/9 Deep Dive , IBM, 2016, https://openpowerfoundation.org/wp-content/uploads/2016/11/Jeff-Stuecheli-POWER9-chip-technology.pdf

[141]: Kellington JT., POWER8 Scale Out, OpenPOWER and CAPI, Georgia IBM POWER User Group, April 16 2015

[142]: Rathle P., Ranganathan K., Neo4j on IBM POWER8, Slideshare, Oct. 27 2016, https://www.slideshare.net/neo4j/webinar-large-scale-graph-processing-with-ibm-power-systems-neo4j?from_action=save

[143]: Goodacre J., The Effect and Technique of System Coherence in ARM Multicore Technology, ARM, 2008, http://www.mpsoc-forum.org/previous/2008/slides/8-6%20Goodacre.pdf

[144]: POWER8 Processor User's Manual for the Single-Chip Module, IBM, March 16 2016

[145]: Wesseler B., IBM i 7.3 kommt bereits im Frühjahr, IT-Zoom, March 15 2016, https://www.it-zoom.de/dv-dialog/e/ibm-i-73-kommt-bereits-im-fruehjahr-12824/

[146]: Morgan T.P., IBM Readies POWER8+ For Openpower Push, The Next Platform, July 15 2015, https://www.nextplatform.com/2015/07/15/ibm-readies-power8-for-openpower-push/

[147]: Morris J., Chipmakers find new ways to go faster, ZDNet, Sept. 21 2016, https://www.zdnet.com/article/chipmakers-find-new-ways-to-go-faster/

[148]: What is the difference between scale-out versus scale-up (architecture, applications, etc.)?, Techopedia, https://www.techopedia.com/7/31151/technology-trends/what-is-the-difference-between-scale-out-versus-scale-up-architecture-applications-etc

13. References (19)

[149]: Stuecheli J., Starke W., The IBM POWER9 Scale Up Processor, Hot chips 30, 2018,
https://www.hotchips.org/hc30/2conf/2.12_IBM_POWER9_HC30.POWER9Cv7.pdf

[150]: Trader T., IBM Advances Against x86 with Power9, HPC Wire, Aug. 30 2016,
https://www.hpcwire.com/2016/08/30/ibm-unveils-power9-details/

[151]: IBM Power System AC922 Introduction and Technical Overview, 2018,
https://www.redbooks.ibm.com/redpapers/pdfs/redp5472.pdf

[152]: Power9 memory buff, Wikichip,
https://en.wikichip.org/wiki/File:power9_memory_buff.svg

[153]: Starke W.J., Dodson J.S., et al., IBM POWER9 memory architectures for optimized systems,
IBM Journal of Research and Development, Vol. 62, Issue 4/5, July-Sept. 2018

[154]: POWER9 Processor User's Manual, IBM, Version 2.0, April 9 2018,
https://www.setphaserstostun.org/power9/POWER9_um_OpenPOWER_v20GA_0
9APR2018_pub.pdf

[155]: Sadasivam S.K. et al., IBM Power9 Processor Architecture, IEEE Micro, Vol. 37, Issue 2,
March-April 2017

[156]: POWER9 EnergyScale Introduction, IBM Community, May 29 2018,
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/
Power%20Systems/page/POWER9%20EnergyScale%20Introduction

[157]: Mesnet B., Introduction to the SNAP framework, OperPOWER Summit Europe, Amsterdam,
Oct. 3-4, 2018, https://openpowerfoundation.org/wp-content/uploads/2018/10/
Bruno.Mesmet.OPF_Amsterdam_Introduction_to_SNAP.pdf

[158]: Stuecheli J., POWER9, IBM Power Systems, 2016,

[159]: Slota M., OpenCAPI Technology, OperPOWER Summit 2018, Las Vegas, March 19 2018, https://openpowerfoundation.org/wp-content/uploads/2018/04/Myron-Slota.pdf

[160]: Slota M., POWER Processor Technology Overview, IBM Power Systems, 2017, https://indico-jsc.fz-juelich.de/event/55/session/4/contribution/0/material/slides/0.pdf

[161]: Armstrong J., IBM POWER9 Family, IBM

[162]: Thompto B., IBM POWER9 Introduction Summit Training Workshop, IBM Power Systems, 2018, https://www.olcf.ornl.gov/wp-content/uploads/2018/12/summit_workshop_thompto.pd

[163]: Krieger L.M., Meet Sierra: Livermore's powerful new supercomputer, The Mercury News, Oct. 27 2018, https://www.mercurynews.com/2018/10/27/meet-sierra-livermores-powerful-new-supercomputer/

[164]: Morgan T.P., The Clever Machinations of Livermore's Sierra Supercomputer, Oct. 5 2017, The Next Platform, https://www.nextplatform.com/2017/10/05/clever-machinations-livermores-sierra-supercomputer/

[165]: Abazovic F., IBM AC922 Power 9 server has 6 Nvidia V100s, Fudzilla, May 8 2018, https://www.fudzilla.com/news/ai/46244-ibm-ac922-power-9-server-has-6-nvidia-v100

[166]: Sibley S., POWER9 Scale-Up servers designed to fuel innovation, IBM, Aug. 7 2018, https://www.ibm.com/blogs/systems/ibm-power9-enterprise-servers/

[167]: IBM Power System E950 Technical Overview and Introduction, Redbooks, Aug. 2018, http://www.redbooks.ibm.com/redpapers/pdfs/redp5509.pdf

13. References (21)

[168]: Centaur – IBM, Wikichip, https://en.wikichip.org/wiki/ibm/centaur

[169]: Starke W.J., Power Processor Micro-Architecture: History of POWER4 through POWER8, IBM, 2017